FortiGate 中文一本通

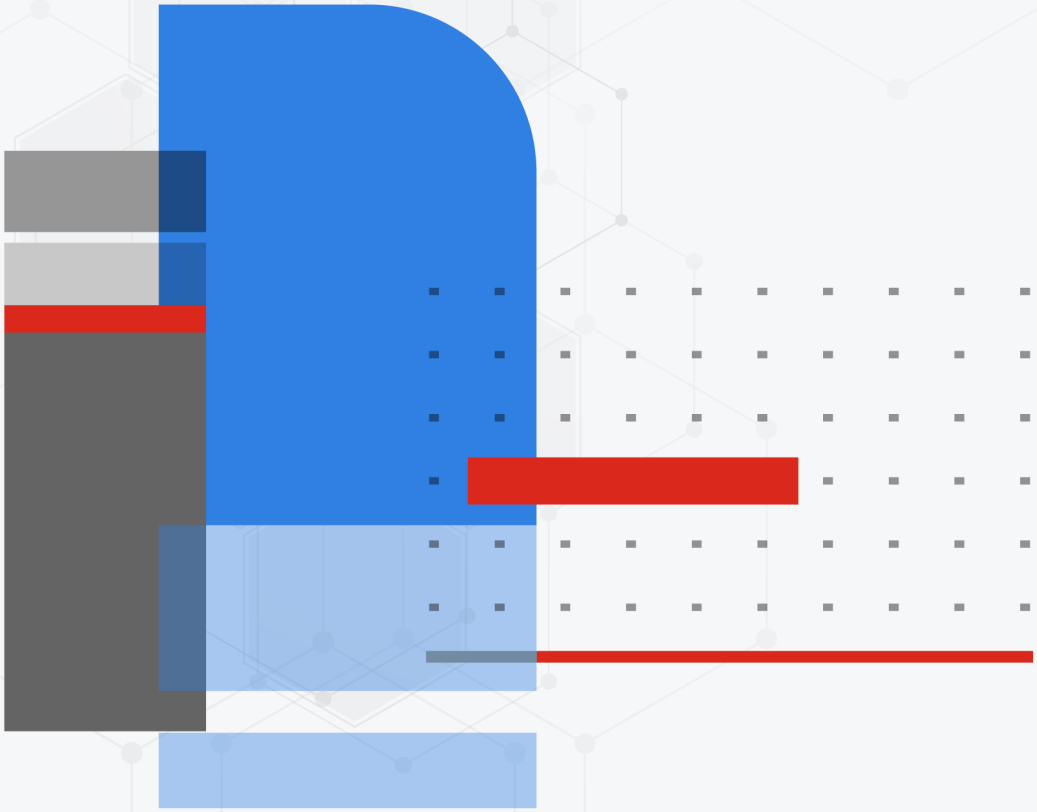
FortiOS 7

40th

Fortinet TAC Team

提示：

通过左侧 PDF 书签浏览文档目录



产品简介

产品介绍

FortiGate 是全新的下一代防火墙，在整个硬件架构和系统上面都有新的设计，在性能和功能上面都有了很大提升，具有性能高、接口丰富、功能齐全、安全路由交换一体化、性价比高等优势。

FortiGate 全新下一代防火墙是面向云计算、数据中心和园区及企业网出口用户开发的新一代高性能防火墙设备。它的功能特性如下：

- 采用多核 CPU + 芯片结构，内置 CP 芯片和 NP 芯片，CP 芯片用于处理内容检测加速，NP 芯片用于处理网络层数据转发，具有很高的数据转发性能，大小包线速，最高可达 80Gps 吞吐。
- 具有稳定的防火墙功能，同时具备全面安全防护功能，入侵防御、反病毒、反垃圾邮件、Web 过滤、敏感数据防泄漏等。
- 支持基于用户，五元组、应用层的安全的访问控制。
- 支持 IPSec、SSL 等多种 VPN 业务，可以构建多种形式的 VPN。
- 提供强大的路由能力，静态/RIP/OSPF/BGP 路由、策略路由，路由负载均衡。
  - 高可靠性，支持 VRRP、同时支持更高靠性的 HA 技术，可实现连接状态的同步，支持主主模式，主备模式；支持多心跳冗余，同时可以利用通讯线路做后备心跳线。充分满足各种环境的可靠性需求。

产品型号

桌面设备

桌面设备是保护小型办公、远程办公、零售网络的理想的安全部署选择，同时还可以做为客户端设备（CPE）。Fortinet 所提供的桌面级的 FortiGate 网络安全设备同样具有全面的网络与安全功能，保护远程的用户、应用以及数据。远离 APT 攻击的威胁。利用 AI/ML 驱动的安全和创新融合（包括安全 SD-WAN）保护和连接您的分布式边缘。

FortiWiFi 设备还提供了一个额外的优势，集成了无线功能，即做为无线胖客户端。FortiWiFi 设备使您具备整合无线与有线流量的能力，消除潜在的安全盲点、简化了无线网络部署与维护的复杂性。

```
FortiGate/FortiWiFi-30G
FortiGate/FortiWiFi-40F
FortiGate/FortiWiFi-50G
FortiGate/FortiWiFi-60F
FortiGate-70F
FortiGate/FortiWiFi-70G
FortiGate/FortiWiFi-80F
FortiGate-90G
```

中端设备

FortiGate 中端型号安全设备是经过用户验证的高性价比的解决方案。Fortinet 能够提供同类厂商中最广的产品线与最先进的安全技术，在不断变化的网络环境中，以有限的部署，提供无限的安全防护。为企业站点获得无与伦比的可见性和保护，以便您在单个仪表板中管理应用程序、用户、设备和访问。

```
FortiGate-100F
FortiGate-120G
FortiGate-200F
FortiGate-200G
FortiGate-400F
FortiGate-600F
FortiGate-900G
```

高端设备

Fortinet 业界领先的网络安全平台可提供下一代防火墙安全，同时具有非凡的吞吐量、超低的延迟率与多重威胁防御机制。您可以依赖 FortiGate 高端设备的扩展性，灵活性，可靠性来实现您网络的需要。部署超大规模安全性，具有一致、协调的保护、丰富的接口和可扩展到任何环境的解密。

FortiGate-6K/7K 系列机架和刀片可对高速服务提供商，数据中心和电信运营商的网络提供优越的性能以及可扩展的灵活性。自带的 10-GbE 支架和高度灵活 ATCA 兼容的架构使 FortiGate-6K/7K 系列可服务于下一代网络苛求的环境以及多租户基于云端的 Saas 与 IaaS。

FortiGate-3K/4K 系列设备可对大型企业和服务管理提供商提供高性能且集成 NGFW 的安全解决方案。基于 FortiASIC 专有芯片加速、高密度端口以及高度的部署灵活性是这一系列产品的显著特点。

```
FortiGate-1000F
FortiGate-1800F
FortiGate-2600F
FortiGate-3000F
FortiGate-3200F
FortiGate-3500F
FortiGate-3700F
FortiGate-4200F
FortiGate-4400F
FortiGate-4800F
FortiGate-6001F
FortiGate-6300F
FortiGate-6500F
FortiGate-7081F
FortiGate-7121F
```

虚拟机 VM

通过 FortiGate 虚拟机， 可实现虚拟架构下威胁的控制与安全防护，同时还可以避免安全盲区以及满足网络的合规性要求。 FortiGate 虚拟机具有如同 FortiGate 物理设备所具有相同的安全与网络服务，自由灵活的集中管控物理设备与虚拟设备混合部署。

```
FortiGate-VM00
Throughput
12 Gbps
vCPU
1x vCPU core, (up to) 2 GB RAM

FortiGate-VM01, -VM01V
Throughput
12 Gbps
vCPU
1x vCPU core, (up to) 2 GB RAM

FortiGate-VM02, -VM02V
Throughput
15 Gbps
vCPU
2x vCPU cores, (up to) 4 GB RAM

FortiGate-VM04, -VM04V
Throughput
28 Gbps
vCPU
4x vCPU cores, (up to) 6 GB RAM

FortiGate-VM08, -VM08V
Throughput
33 Gbps
vCPU
8x vCPU cores, (up to) 12 GB RAM

FortiGate-VM16, -VM16V
Throughput
36 Gbps
vCPU
16x vCPU cores, (up to) 24 GB RAM

FortiGate-VM32, -VM32V
Throughput
50 Gbps
vCPU
32x vCPU cores, (up to) 48 GB RAM

FortiGate-VMUL, -VMULV
vCPU
Unlimited vCPU cores and RAM
```

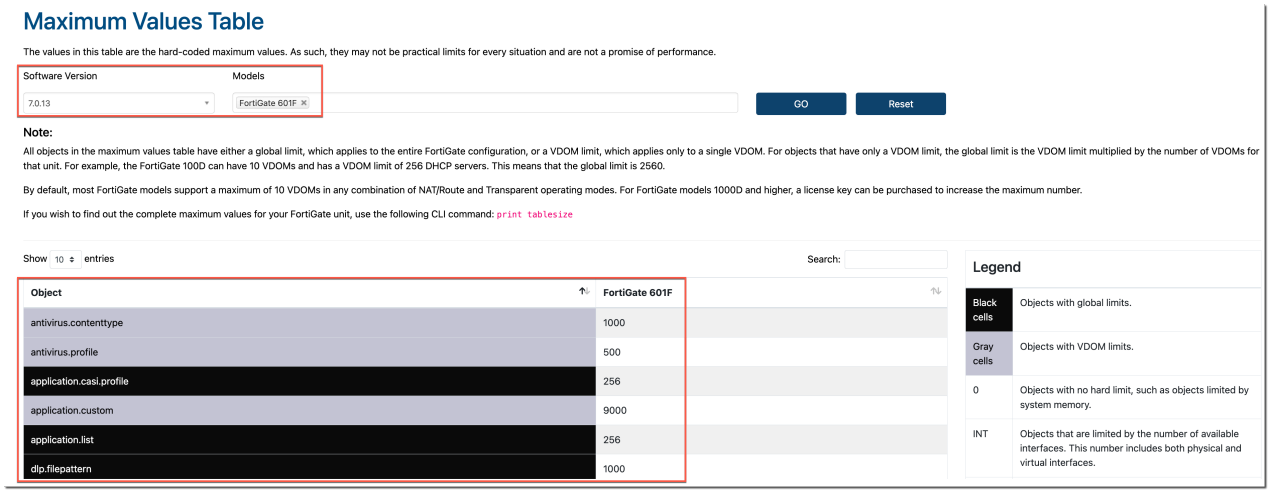
详细产品参数参考官网：https://www.fortinet.com/cn/products/next-generation-firewall.html#models-specs ↗

产品性能规格

不同产品型号的性能规格请参考Fortinet Product Matrix ↗。

产品功能规格

1. 不同产品型号的功能规格请参考 https://docs.fortinet.com/max-value-table↗ 查询。



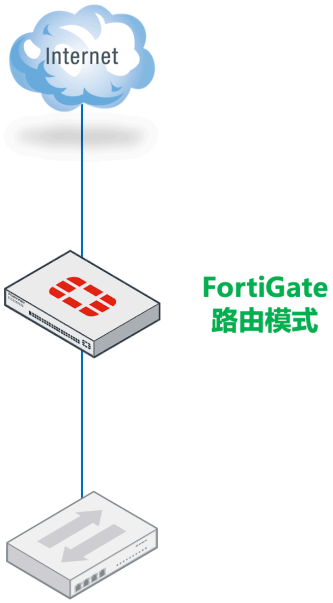
2. 也可以在对应型号设备的 CLI 下执行“print tablesize”命令得到规格显示。

```bash
FortiGate # print tablesize
system.vdom: 0 0 10
system.datasource: 0 0 0
system.accprofile: 0 0 18
system.np6xlite: 0 256 512
system.vdom-link: 0 0 0
system.switch-interface: 0 0 0
system.switch-interface:span-source-port: 0 0 0
system.switch-interface:member: 0 0 0
system.object-tagging: 0 1024 1024
system.object-tagging:tags: 0 0 0
system.interface: 0 0 4096
system.interface:client-options: 0 0 0
system.interface:fail-alert-interfaces: 0 0 0
system.interface:member: 0 0 0
system.interface:security-groups: 0 0 0
system.interface:vrrp: 0 0 0
system.interface:vrrp:proxy-arp: 0 0 0
system.interface:secondaryip: 256 0 0
system.interface:dhcp-snooping-server-list: 255 0 2048
......
```

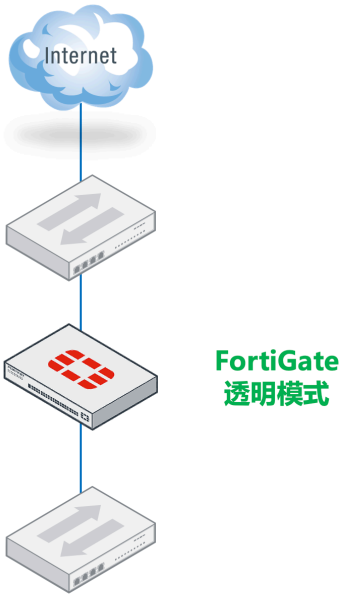
产品布署模式

FortiGate 有四种工作模式：路由模式、透明模式、旁路模式及混合模式：

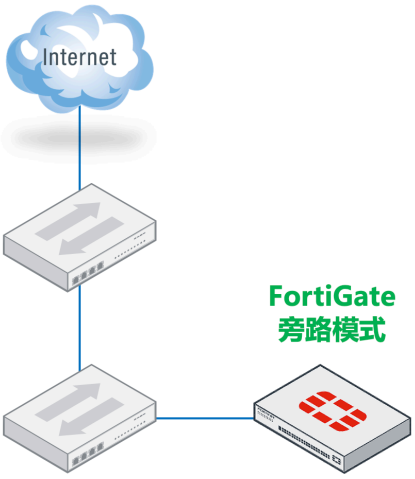
1. NAT（路由）模式：把设备当作网络出口，充分使用设备的 NAT、路由选路、行为控制、VPN 等功能的部署方式。如下图；



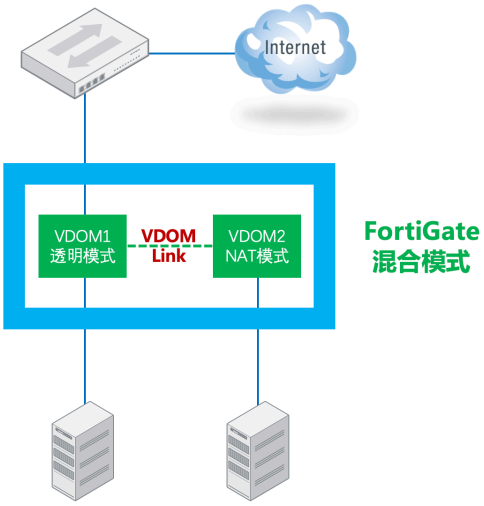
2. 透明模式：当用户网络已具备高性能的网络出口，又想使用 NGFW 的功能，则可以把 NGFW 串接在内网核心交换机和网络出口设备之间，用户将不必变更网络拓扑和修改路由，就 可使用 NGFW 的行为控制、VPN 等功能。如下图；



3. 旁路模式：当用户网络已具备高性能的网络出口，也不想把设备串接在内网核心交换机和出口设备之间，又想使用 NGFW 的 VPN、DHCP 等功能，则可将设备像一台服务器一样旁挂在核心交换机上，如下图；



4. 混合模式（vdom）：指设备同时工作于是路由模式和透明模式，将一个设备分成两个 vdom 域，一个 vdom 是透明模式，一个 vdom 是路由模式，将多个接口划入不同的 vdom 实现混合模式，如下图。



产品生命周期

Fortinet 各个硬件与软件的产品生命周期（Product Life Cycle），请访问 https://support.fortinet.com/support/#/lifecycle ↗ ，使用 Support 账户登录后查看。

产品选型

产品选型请联系 Fortinet 售前部门：se_china@fortinet.com ↗

部署前注意事项

版本推荐

公网接口访问限制

不要将 10G 与 1G 接口混用

120G/121G HA 心跳口问题

PPPoE 接口配置

TLS 探测配置

开启 COMLog

云部署注意事项

运维注意事项

公网接口访问限制

为了提升网络安全性，限制来自互联网对 FortiGate 的管理访问是至关重要的。通过限制管理接口的访问范围，可以有效防止未经授权的远程登录，降低潜在的安全风险。

为什么要限制 FortiGate 的互联网访问?

- **防止未经授权的访问**：公开的管理接口可能被恶意用户扫描和利用，导致未经授权的访问。
- **减少攻击面**：限制管理接口的访问范围，可以减少暴露在互联网中的服务，降低被攻击的可能性。
- **遵循安全最佳实践**：网络安全的基本原则之一是最小化暴露面，限制管理接口的访问是实现这一原则的重要措施。

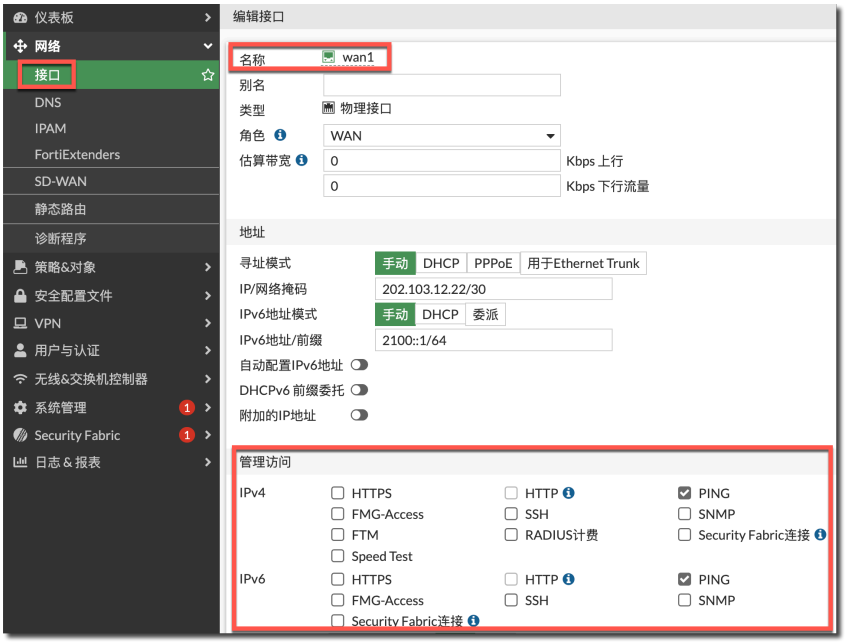
请根据实际网络环境和安全需求，合理应用以下方法，以确保 FortiGate 的安全管理。

禁用互联网接口管理协议

> **重要**
>
> 通过禁用外部接口（如 WAN 口）的管理协议（如 HTTP、HTTPS、SSH、Telnet、FMG-Access 等）访问，仅保留必要的管理协议（如 Ping），可以减少暴露在互联网的管理服务，降低被攻击的风险。

1. 进入“网络 → 接口”页面，编辑 FortiGate 的互联网网接口（这里以 wan1 为例），在管理访问中，只开启 PING 访问，关闭其他协议的管理访问（如果 FortiGate 启用了互联网接口的 IPv6，则需要在 IPv6 的管理访问中执行相同操作）。

   > **重要**
   >
   > FortiManager 通过 Internet 管理 FortiGate 时，如果是 FortiGate 主动注册到 FortiManager 的模式，不需要开启 FMG-Access 管理。




```bash
config system interface
    edit "wan1"
        set allowaccess ping
        config ipv6
            set ip6-allowaccess ping
        end
    next
end
```

2. 在内网接口上开启管理协议（如 HTTPS、SSH 等）通过内网接口或 VPN 远程连接的方式管理 FortiGate。

配置 Local-in 策略

重要

Local-in 策略用于控制目标是 FortiGate 自身地址的流量，可用于限制对接口本身的访问。

关于 Local-in 策略的更多介绍可参考策略与对象 → 本地策略 → Local-in 策略 ↗章节。

1. 如果由于实际需要，FortiGate 的外网接口必须开启 HTTPS/SSH 远程访问（或内网接口限定使用特定跳板机/VPN 客户端网段访问），可以通过 Local-in 策略指定可信任的主机 IP 地址范围，确保只有来自这些地址的连接才能进行管理访问，从而提高安全性。

```bash
config system interface
    edit "wan1"
        set allowaccess ping https ssh
        config ipv6
            set ip6-allowaccess ping https ssh
        end
    next
end
```

2. 创建允许访问 FortiGate 的公网接口的源地址对象（请替换内容为您真实环境中的信任公网地址），并使用地址组引用这些地址对象。

```bash
config firewall address
    edit "60.247.121.226/32"
        set subnet 60.247.121.226 255.255.255.255
    next
    edit "123.112.245.11/32"
        set subnet 123.112.245.11 255.255.255.255
    next
    edit "39.105.115.224/29"
        set subnet 39.105.115.224 255.255.255.248
    next
end

config firewall addrgrp
    edit "Internet_Admin_Trust_List4"
        set member "39.105.115.224/29" "60.247.121.226/32" "123.112.245.11/32"
    next
end
```

3. （可选）如果 FortiGate 启用了互联网接口的 IPv6，则需要同时创建 IPv6 的地址对象/地址组。

```bash
config firewall address6
    edit "240e:88::3/128"
        set ip6 240e:88::3/128
    next
    edit "2001::/96"
        set ip6 2001::/96
    next
end

config firewall addrgrp6
    edit "Internet_Admin_Trust_List6"
        set member "2001::/96" "240e:88::3/128"
    next
end
```

4. 在 FortiGate 的互联网接口（这里以 wan1 为例）上配置 Local-in 策略，禁止除源地址组外的其他地址访问 FortiGate 互联网接口的 HTTP/SSH 协议。

重要

- Local-in 策略只会影响目标是 FortiGate 自身地址的流量，不会影响 DNAT 策略流量（穿越流量）。
- 此例中的 FortiGate 使用默认的 HTTPS 端口 443、SSH 端口 22 作为管理端口，如果您修改了默认 HTTPS/SSH 管理端口，请创建对应端口的服务对象（如下所示），然后在 Local-in 策略中的 `service` 中引用。

```bash
config firewall service custom
    edit GUI_HTTPS
        set tcp-portrange 4443
    next
    edit Custom_SSH
        set tcp-portrange 2233
    next
end
```

```bash
config firewall local-in-policy
    edit 1
        set intf "wan1"
        set srcaddr "Internet_Admin_Trust_List4"
        set dstaddr "all"
        set action accept
        set service "HTTPS" "SSH"
        set schedule "always"
    next
    edit 2
        set intf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action deny
        set service "HTTPS" "SSH"
        set schedule "always"
    next
end
```

重要

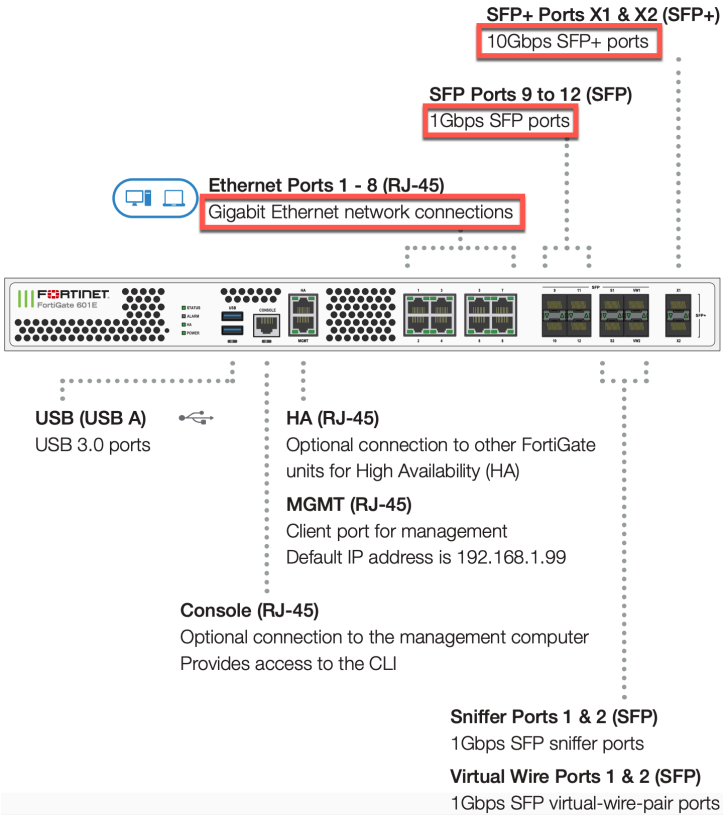
Local-in Policy 默认隐含策略的动作为 permit all。所以以上配置只会影响 HTTPS、SSH 的访问，不会影响其他访问 FortiGate 自身的流量（如 IPSec、SSL VPN、BGP、OSPF 等）。

5. （可选）如果 FortiGate 启用了互联网接口的 IPv6，则需要对应创建 IPv6 的 Local-in 策略。

```bash
config firewall local-in-policy6
    edit 1
        set intf "wan1"
        set srcaddr "Internet_Admin_Trust_List6"
        set dstaddr "all"
        set action accept
        set service "HTTPS" "SSH"
        set schedule "always"
    next
    edit 2
        set intf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action deny
        set service "HTTPS" "SSH"
        set schedule "always"
    next
end
```

不要将 10G 与 1G 接口混用

1. 对于使用 NP6 芯片（不包含 NP6xlite）的设备（如 FortiGate 600E），同时具备 QSGMII 的 1G 接口和 XAUI 的 10G 接口：



2. 如果存在以下情况，容易出现 NP 层面丢包，这对基于 TCP 的服务应用很可能造成测速下降的问题现象：

   - 在 10G 接口 → 1G 接口的方向上存在较高带宽的数据传输
   - 数据传输过程存在突发流量 micro bursts
   - 流量可以被 NP6 加速

3.**实际部署时，尽量避免同时使用 1G 接口和 10G 接口进行流量转发。**

4. 如果设备已经部署无法更改接口连线，需要同时保持 10G 和 1G 接口在生产环境中使用，可以考虑修改以下配置让上述流量不被 NPU 加速，以规避这个问题。

```bash
config system npu
  set host-shortcut-mode host-shortcut    //该命令效果是10G→1G方向的流量关闭NPU加速
end
```

120G/121G HA 心跳口问题

问题现象

在使用 FortiOS 7.2.9/7.2.10 版本的 120G/121G 设备上配置 HA，且 HA 心跳接口使用“ha”或“mgmt”接口时，会出现如下问题：

1. “系统管理 → 高可靠性”页面无法正常打开，一直处于加载状态。
2. 两台 120G/121G 设备的 HA 状态一直无法成为“Synchronized”（同步）状态，如下所示（在主机 FG120GTKXXYYZZ50 查看 HA 状态），备机 FortiGate (FG120GTKXXYYZZ59) 的状态在 1721194954 秒前更新，checksum dump 为 00。这表明 FortiGate 设备之间存在心跳连接问题。

```bash
FG120GTKXXYYZZ59 # get system ha status
HA Health Status: OK
Model: FortiGate-120G
......
 Configuration Status:
    FG120GTK24001650(updated 1 seconds ago): in-sync
    FG120GTKXXYYZZ50 chksum dump: 29 e0 e4 32 6c 76 99 68 2b ed 8b bc c1 2d 2c 37
    //FG120GTKXXYYZZ59(updated 1721194954 seconds ago): out-of-sync//<———备机无法同步
    //FG120GTKXXYYZZ59 chksum dump: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00//<———checksum dump为00
......
HBDEV stats:
    FG120GTK24001650 (updated 1 seconds ago):
        ha: physical/1000auto, up, rx-bytes/packets/dropped/errors=121750988/234138/0/0, tx=120280738/235391/0/0
    //FG120GTKXXYYZZ59 (updated 1721194954 seconds ago)://<———更新时间异常
Primary : FortiGate-120G , FG120GTK24001650, HA cluster index = 1
Secondary : , FG120GTKXXYYZZ59, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.2
Primary: FG120GTK24001650, HA operating index = 0
Secondary: FG120GTKXXYYZZ59, HA operating index = 1
```

问题原因

当使用“ha”或“mgmt”接口用作 HA 心跳接口时，FortiGate 无法正常更新 HA 逻辑接口“port_ha”的 MAC 地址。这是一个 FortiOS 7.2.9/7.2.10 版本的 bug（参考：https://docs.fortinet.com/document/fortigate/7.2.9/fortios-release-notes/236526/known-issues，Bug ID 1056138）。

解决方法

1. Workaround：暂时使用“ha”和“mgmt”以外的接口作为 HA 心跳接口。
2. 该问题已在 FortiOS 7.2.11/7.4.5/7.6.1 版本解决。

PPPoE 接口配置

配置说明

需要使用 FortiGate PPPoE 拨号时，建议通过 CLI 的方式新建 PPPoE 虚拟接口，在该接口下配置用户名与密码，并绑定公网物理口。

> **重要**
>
> 在 7.4.2 及以后版本的硬件设备上，可以在 GUI 创建 PPPoE 虚拟接口。

使用该种配置方法有如下优势：

- 传统方式（将普通接口的模式修改为 PPPoE），修改接口下与 PPPoE 拨号无关的参数（如“DNS 覆盖”、接口管理协议、MTU、inbandwidth/outbandwidth 等），PPPoE 会重新拨号，影响业务。而手动创建的 PPPoE 虚拟接口修改与 PPPoE 拨号无关的参数，PPPoE 不会重新拨号，不会影响业务。
- 传统方式（将普通接口的模式修改为 PPPoE）拨号成功后，会动态创建 ppp 接口，无法指定动态创建的 ppp 接口的 `tcp-mss`。PPPoE 虚拟接口可以针对 PPPoE 虚拟接口设置 `tcp-mss` 参数（传统方式），这在 PPPoE 环境下产生的 TCP 分片丢包问题非常有用。
- PPPoE 接口配置更灵活，可以在一个物理口上配置多个 PPPoE 接口。可独立于物理口配置每个 PPPoE 接口的属性。在其他模块（如防火墙策略、SD-WAN）引用接口时更加直观和灵活。
- 由于 PPPoE 接口为动态获取网关，与策略路由配合使用时，策略路由可以正常获取到 PPPoE 的网关，不需要手动填写网关。
- 可配置按需拨号。

配置步骤

1. 例如配置 port2 的 PPPoE，在 CLI（`config system pppoe-interface`）下新建虚拟 PPPoE 接口，并绑定在 port2 接口上，配置 PPPoE 用户名和密码。

```bash
config system pppoe-interface
    edit "pppoe"
        set device "port2"
        set username "bing"
        set password xxxxxxxx
    next
end
```

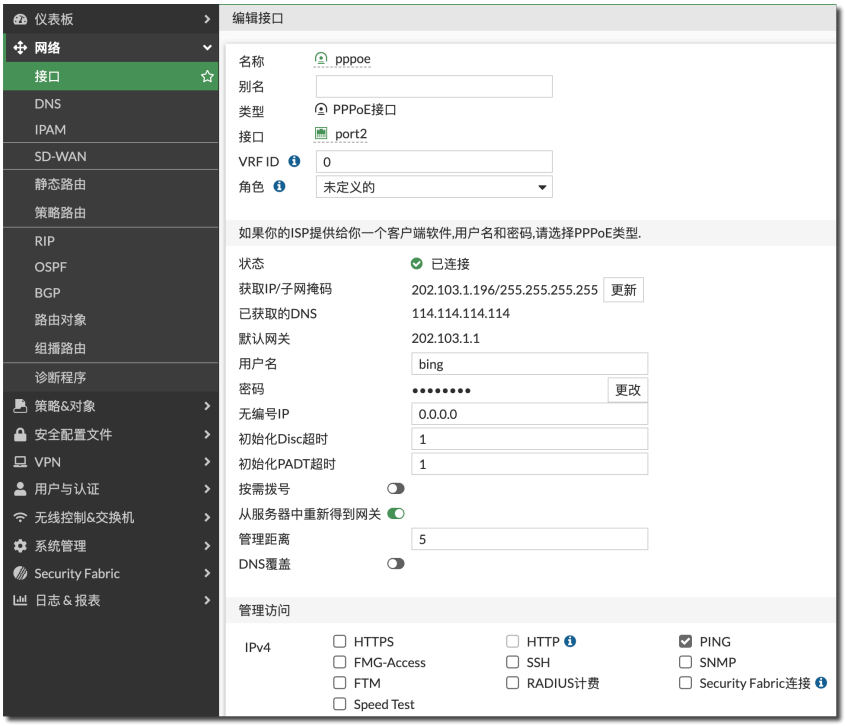
2. 以上命令执行后，会自动在 `config system interface` 下创建同名接口，其中包含“从服务器中重新得到网关”的配置。

```bash
FortiGate # show full sys int pppoe
config system interface
    edit "pppoe"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set defaultgw enable
        set interface "port2"
    next
end
```

3. 在此接口下，可以配置 PPPoE 虚拟接口的相关参数，如管理协议、TCP - MSS、DNS 覆盖等，在这里修改的参数不会导致 PPPoE 重新拨号。

```bash
config system interface
    edit "pppoe"
        set allowaccess ping
        set tcp-mss 1400
        set dns-server-override disable
    next
end
```

4. 以上配置完成后，可以在 GUI 上看到 port2 下多了一个 PPPoE 接口，并获取到了运营商分配的 IP。


仪表板
网络
接口
DNS
IPAM
SD-WAN
静态路由
策略路由
RIP
OSPF
BGP
路由对象
组播路由
诊断程序
策略&对象
安全配置文件
VPN
用户与认证
无线控制&交换机
系统管理
Security Fabric
日志 & 报表
编辑接口
名称 pppoe
别名
类型 PPPoE接口
接口 port2
VRF ID 0
角色 未定义的
如果你的ISP提供给你一个客户端软件,用户名和密码,请选择PPPoE类型.
状态 已连接
获取IP/子网掩码 202.103.1.196/255.255.255.255 更新
已获取的DNS 114.114.114.114
默认网关 202.103.1.1
用户名 bing
密码 ••••••••
更改
无编号IP 0.0.0.0
初始化Disc超时 1
初始化PADT超时 1
按需拨号
从服务器中重新得到网关
管理距离 5
DNS覆盖
管理访问
IPv4
HTTPS
HTTP
PING
FMG-Access
SSH
SNMP
FTM
RADIUS计费
Security Fabric连接
Speed Test

TLS 探测配置

TLS 探测介绍

当 FortiGate 在检查基于 TLS 的连接（如开启 WebFilter 并检测 HTTPS 连接）时，客户端经过 FortiGate 的业务 VDOM 访问某个 HTTPS 网站，FortiGate 会提取客户端 PC 发送的 TLS Client Hello 中的 SNI，从 SNI 中提取网站的域名，然后先于客户端去探测网站证书的合法性，这是为了保证后续 TLS 协商以及业务报文的安全性。TLS 探测报文的源地址只会使用 FortiGate 自身的 IP 地址。

在多出口、多 VDOM、透明模式、SD-WAN、接口配置第二 IP 等场景下，TLS 探测可能会因路由或线路问题导致超时失败。

拓扑信息

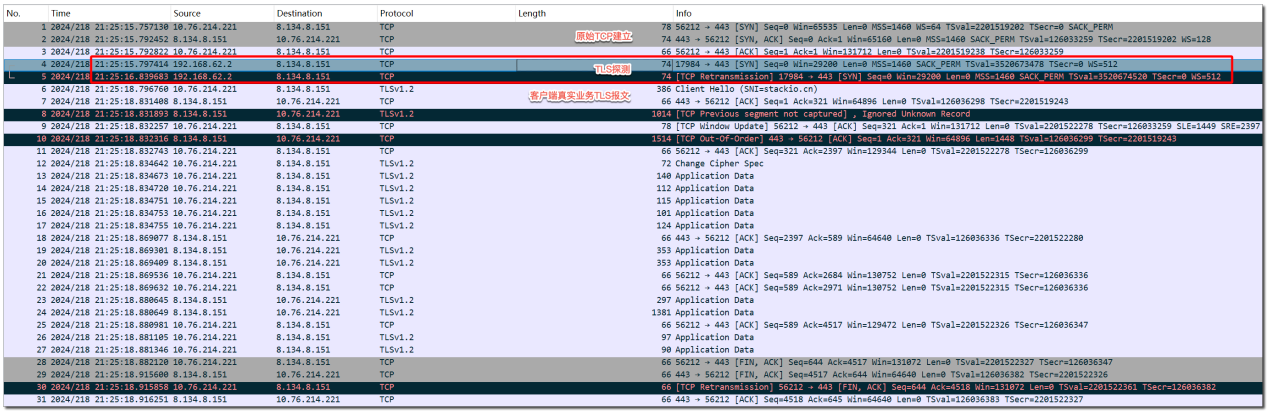
FortiGate 如果开启了网页过滤（Web Filter）功能，需要注意如下拓扑环境下的配置。

- FortiGate 划分了多个 VDOM：
  - 管理 VDOM 接口（mgmt）可以访问 Internet 与内网其他设备，用于登录、FortiGuard 分类查询、Syslog、SNMP 等管理业务。
  - 业务 VDOM 的所有接口只做业务转发，接口 IP 无法访问 Internet。
  - 通过业务 VDOM 接口转发的流量可以正常访问 Internet。
- FortiGate 未开启 VDOM：
  - 默认路由中包含多个出接口。
  - 其中一些出接口的 IP 无法访问 Internet。
  - 接口配置了第二地址，只有第二地址可以访问 Internet。

问题现象

1. 以 FortiGate 多 VDOM 的场景为例。默认配置下，从 FortiGate 发送的 TLS 探测报文，只能以防火墙业务 VDOM 自己的接口 IP 来发出。

2. 如果业务 VDOM 的接口 IP 本身无法访问 Internet 的，FortiGate 去探测网站 HTTPS 时，TCP 是无法建立的，超时 3s 后，会放弃 TLS 探测，继续真实客户端的 TLS 报文的转发。如下所示在 FortiGate 业务 VDOM 的外网口抓包。



No.	Time	Source	Destination	Protocol	Length	Info
1	2024/218 21:25:15.757130	10.76.214.221	8.134.8.151	TCP		78 56212 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2201519202 TSecr=0 SACK_PERM
2	2024/218 21:25:15.792452	8.134.8.151	10.76.214.221	TCP		74 443 → 56212 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=126033259 TSecr=2201519202 WS=128
3	2024/218 21:25:15.792822	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=2201519238 TSecr=126033259
4	2024/218 21:25:15.797414	192.168.62.2	8.134.8.151	TCP		74 17984 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=3520673478 TSecr=0 WS=512
5	2024/218 21:25:16.839683	192.168.62.2	8.134.8.151	TCP		74 [TCP Retransmission] 17984 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=3520674520 TSecr=0 WS=512
6	2024/218 21:25:18.796760	10.76.214.221	8.134.8.151	TLSv1.2		386 Client Hello (SNI=stackio.cn)
7	2024/218 21:25:18.831408	8.134.8.151	10.76.214.221	TCP		66 443 → 56212 [ACK] Seq=1 Ack=321 Win=64896 Len=0 TSval=126036298 TSecr=2201519243
8	2024/218 21:25:18.831893	8.134.8.151	10.76.214.221	TLSv1.2		1014 [TCP Previous segment not captured] , Ignored Unknown Record
9	2024/218 21:25:18.832257	10.76.214.221	8.134.8.151	TCP		78 [TCP Window Update] 56212 → 443 [ACK] Seq=321 Ack=1 Win=131712 Len=0 TSval=2201522278 TSecr=126033259 SLE=1449 SRE=2397
10	2024/218 21:25:18.832316	8.134.8.151	10.76.214.221	TCP		1514 [TCP Out-Of-Order] 443 → 56212 [ACK] Seq=1 Ack=321 Win=64896 Len=1448 TSval=126036299 TSecr=2201519243
11	2024/218 21:25:18.832743	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [ACK] Seq=321 Ack=2397 Win=129344 Len=0 TSval=2201522278 TSecr=126036299
12	2024/218 21:25:18.834642	10.76.214.221	8.134.8.151	TLSv1.2		72 Change Cipher Spec
13	2024/218 21:25:18.834673	10.76.214.221	8.134.8.151	TLSv1.2		140 Application Data
14	2024/218 21:25:18.834720	10.76.214.221	8.134.8.151	TLSv1.2		112 Application Data
15	2024/218 21:25:18.834751	10.76.214.221	8.134.8.151	TLSv1.2		115 Application Data
16	2024/218 21:25:18.834753	10.76.214.221	8.134.8.151	TLSv1.2		101 Application Data
17	2024/218 21:25:18.834755	10.76.214.221	8.134.8.151	TLSv1.2		124 Application Data
18	2024/218 21:25:18.869077	8.134.8.151	10.76.214.221	TCP		66 443 → 56212 [ACK] Seq=2397 Ack=589 Win=64640 Len=0 TSval=126036336 TSecr=2201522280
19	2024/218 21:25:18.869301	8.134.8.151	10.76.214.221	TLSv1.2		353 Application Data
20	2024/218 21:25:18.869409	8.134.8.151	10.76.214.221	TLSv1.2		353 Application Data
21	2024/218 21:25:18.869536	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [ACK] Seq=589 Ack=2684 Win=130752 Len=0 TSval=2201522315 TSecr=126036336
22	2024/218 21:25:18.869632	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [ACK] Seq=589 Ack=2971 Win=130752 Len=0 TSval=2201522315 TSecr=126036336
23	2024/218 21:25:18.880645	8.134.8.151	10.76.214.221	TLSv1.2		297 Application Data
24	2024/218 21:25:18.880649	8.134.8.151	10.76.214.221	TLSv1.2		1381 Application Data
25	2024/218 21:25:18.880981	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [ACK] Seq=589 Ack=4517 Win=129472 Len=0 TSval=2201522326 TSecr=126036347
26	2024/218 21:25:18.881105	10.76.214.221	8.134.8.151	TLSv1.2		97 Application Data
27	2024/218 21:25:18.881346	10.76.214.221	8.134.8.151	TLSv1.2		90 Application Data
28	2024/218 21:25:18.882120	10.76.214.221	8.134.8.151	TCP		66 56212 → 443 [FIN, ACK] Seq=644 Ack=4517 Win=131072 Len=0 TSval=2201522327 TSecr=126036347
29	2024/218 21:25:18.915600	8.134.8.151	10.76.214.221	TCP		66 443 → 56212 [FIN, ACK] Seq=4517 Ack=644 Win=64640 Len=0 TSval=126036382 TSecr=2201522326
30	2024/218 21:25:18.915858	10.76.214.221	8.134.8.151	TCP		66 [TCP Retransmission] 56212 → 443 [FIN, ACK] Seq=644 Ack=4518 Win=131072 Len=0 TSval=2201522361 TSecr=126036382
31	2024/218 21:25:18.916251	8.134.8.151	10.76.214.221	TCP		66 443 → 56212 [ACK] Seq=4518 Ack=645 Win=64640 Len=0 TSval=126036383 TSecr=2201522327

3. 相关 IPS Debug 日志显示 TLS 探测失败。

```bash
diagnose ips filter set "(host x.x.x.x or host x.x.x.x) and port 443"
diagnose ips debug enable all
diagnose debug console timestamp enable
diagnose debug enable

2024-12-27 15:32:49 [10600@-1]ssl_resume_sess: sess 4311: ssl resume
2024-12-27 15:32:49 [10600@-1]ips_urlf_del_query: id:2229, queue:0
2024-12-27 15:32:49 [10600@-1]eng_debug_log: Probe info:
2024-12-27 15:32:49 [10600@-1]eng_debug_log:   Server: 8.134.8.151:443
2024-12-27 15:32:49 [10600@-1]eng_debug_log:   Server name: stackio.cn
2024-12-27 15:32:49 [10600@-1]eng_debug_log:   STARTTLS: no
2024-12-27 15:32:49 [10600@-1]eng_debug_log:   Probe failed: unable to connect    ←——TLS探测失败
2024-12-27 15:32:49 [10600@-1]eng_debug_log: parallel probes: 1
2024-12-27 15:32:49 [10600@-1]eng_debug_log: Memory usage: 276 KiB, errors: 2046
2024-12-27 15:32:49 [10601@-1]probe_finish: probe is finished. id: 1194, sess: 4218
2024-12-27 15:32:49 [10601@-1]ssl_resume_sess: sess 4218: ssl resume
2024-12-27 15:32:49 [10601@-1]ips_eng_log_ssl: ssl log host stackio.cn, CN , type 12
```

4. 对应的 SSL 安全事件日志（日志子类型为 `certificate-probe-failed`）。

```
date=2025-10-16 time=14:20:41 eventtime=1760595641888725921 tz="+0800" logid="1700062305" type="utm" subtype="ssl"
eventtype="ssl-anomaly" level="warning" vd="root" action="allow" policyid=1 poluuid="04d852a8-7f02-51f0-65cd-
4c50ca8f3632" policytype="policy" sessionid=6753380 service="HTTPS" profile="test-example" srcip=192.168.100.177
srcport=53784 srccountry="Reserved" dstip=173.243.143.6 dstport=443 dstcountry="United States" srcintf="lan"
srcintfrole="lan" dstintf="wan1" dstintfrole="wan" srcuuid="e2f430d6-7f00-51f0-8d55-e762fbba6612" dstuuid="e2f430d6-
7f00-51f0-8d55-e762fbba6612" proto=6 eventsubtype="certificate-probe-failed" hostname="173.243.143.6"
```
bash

5. 这是由于默认配置下，TLS 探测报文使用当前业务 VDOM 下的默认路由出接口。但只有管理 VDOM 的接口可以正常访问 Internet，这样就会造成访问延迟。

解决方法 1

1. 修改 TLS 探测使用的 VDOM 和接口。

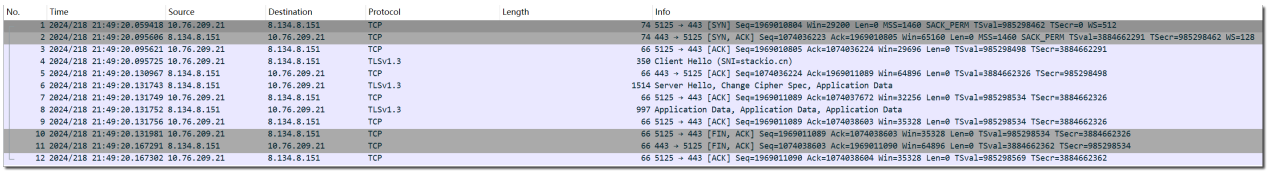
```
config ips global
config tls-active-probe
    set interface-selection-method <auto|sdwan|specify>
    set interface <intf name>
    set vdom <vdom name>
    set source-ip <source_ipv4>
    set source-ip6 <source_ipv6>
  end
end
```
bash

- `interface-selection-method`：TLS 探测报文选择接口的方式。
  - `auto`：默认配置，根据当前 VDOM 下的路由表来进行发送（有多个接口负载分担时，可能会发送到错误的接口）；
  - `sdwan`：匹配 SD-WAN 规则进行发送。
  - `specify`：手工选择需要发送的方式。
- `interface`：指定发送 TLS 探测报文的接口（当 `interface-selection-method` 设置为 `specify` 时可选）。
- `vdom`：指定发送 TLS 探测报文的 VDOM（当 `interface-selection-method` 设置为 `specify` 或 `sdwan` 时可选）。
- `source-ip` / `source-ip6`：指定发送 TLS 探测报文的源 IPv4/IPv6（当 `interface-selection-method` 设置为 `specify` 或 `sdwan` 时可选）。

2. 此例中，在 Global 下将 TLS 探测的接口改为 root VDOM 的 mgmt 口，可以正常进行 TLS 探测，不会产生超时现象。

```
config global
    config ips global
        config tls-active-probe
            set interface-select-method specify
            set vdom root
            set interface mgmt
        end
    end
end
```
bash

3. 在 mgmt 接口上抓取的完整的 TLS 探测报文如下。



No.	Time	Source	Destination	Protocol	Length	Info
1	2024/218 21:49:20.059418	10.76.209.21	8.134.8.151	TCP		74 5125 → 443 [SYN] Seq=1969010804 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=985298462 TSecr=0 WS=512
2	2024/218 21:49:20.095606	8.134.8.151	10.76.209.21	TCP		74 443 → 5125 [SYN, ACK] Seq=1074036223 Ack=1969010805 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3884662291 TSecr=985298462 WS=128
3	2024/218 21:49:20.095621	10.76.209.21	8.134.8.151	TCP		66 5125 → 443 [ACK] Seq=1969010805 Ack=1074036224 Win=29696 Len=0 TSval=985298498 TSecr=3884662291
4	2024/218 21:49:20.095725	10.76.209.21	8.134.8.151	TLSv1.3		350 Client Hello (SNI=stackio.cn)
5	2024/218 21:49:20.130967	8.134.8.151	10.76.209.21	TCP		66 443 → 5125 [ACK] Seq=1074036224 Ack=1969011089 Win=64896 Len=0 TSval=3884662326 TSecr=985298498
6	2024/218 21:49:20.131743	8.134.8.151	10.76.209.21	TLSv1.3		1514 Server Hello, Change Cipher Spec, Application Data
7	2024/218 21:49:20.131749	10.76.209.21	8.134.8.151	TCP		66 5125 → 443 [ACK] Seq=1969011089 Ack=1074037672 Win=32256 Len=0 TSval=985298534 TSecr=3884662326
8	2024/218 21:49:20.131752	8.134.8.151	10.76.209.21	TLSv1.3		997 Application Data, Application Data, Application Data
9	2024/218 21:49:20.131756	10.76.209.21	8.134.8.151	TCP		66 5125 → 443 [ACK] Seq=1969011089 Ack=1074038603 Win=35328 Len=0 TSval=985298534 TSecr=3884662326
10	2024/218 21:49:20.131981	10.76.209.21	8.134.8.151	TCP		66 5125 → 443 [FIN, ACK] Seq=1969011089 Ack=1074038603 Win=35328 Len=0 TSval=985298534 TSecr=3884662326
11	2024/218 21:49:20.167291	8.134.8.151	10.76.209.21	TCP		66 443 → 5125 [FIN, ACK] Seq=1074038603 Ack=1969011090 Win=64896 Len=0 TSval=3884662362 TSecr=985298534
12	2024/218 21:49:20.167302	10.76.209.21	8.134.8.151	TCP		66 5125 → 443 [ACK] Seq=1969011090 Ack=1074038604 Win=35328 Len=0 TSval=985298569 TSecr=3884662362

解决方法 2

如果是 FortiGate 所有接口都不能访问 Internet 的封闭环境，可以关闭 TLS 探测，需要配置自定义 SSL/SSH 配置文件，然后在防火墙策略中引用（出于安全性考虑，不推荐此方法）。

```bash
config firewall ssl-ssh-profile
    edit "test-example"
        config https
            set sni-server-cert-check disable
        end
    next
end

config firewall policy
    edit 1
        set name "to_Internet"
        set ssl-ssh-profile "test-example"
    next
end
```

TLS 探测失败默认动作

上文介绍过，在检查基于 TLS 的连接（如 HTTPS）时，FortiGate 需要执行 TLS 证书探测，以便获取服务器的证书并确定用户试图访问的目标域名/地址。若因某些原因导致此 TLS 探测失败（例如 FortiGate 的探测请求无法到达目标 Web 服务器），则 FortiGate 需判断该连接应被阻断还是允许通过。

SSL/SSH 配置文件中的 `cert-probe-failure` 选项可以控制此行为：

> **重要**
>
> - 在 FortiOS 7.6.0 之前版本中，`cert-probe-failure` 的默认动作为 `block`。
> - 在 FortiOS 7.6.0 及之后版本中，`cert-probe-failure` 的默认动作为 `allow`。

```bash
config firewall ssl-ssh-profile
    edit <profile_name>
        config https
            set cert-probe-failure <allow|block>     ←——在7.6.0之前，默认配置为block，在7.6.0及之后，默认动作为allow
        end
    next
end
```

但在不同的 FortiOS 版本/防火墙策略检测模式（flow/proxy）上，该配置产生的影响存在区别。

- **7.2.11/7.4.5/7.6.1 之前版本（动作 block）：**
  - **代理（proxy）模式：** 若 TLS 探测失败，则阻断连接。该选项仅对采用代理（proxy）模式的防火墙策略生效。
  - **流（flow）模式：** `cert-probe-failure` 选项会被忽略，此时即使探测失败，系统仍会始终允许连接建立（但可能会产生上文提到的访问延迟）。
- **7.2.11/7.4.5/7.6.1 及之后版本（动作 block）：**
  - **代理（proxy）模式：** 若 TLS 探测失败，则阻断连接。
  - **流（flow）模式：** `cert-probe-failure` 选项会被考虑，若 TLS 探测失败，则阻断连接，与 proxy 模式保持一致。

产生的问题

如上所述，在 7.2.11/7.4.5/7.6.1 及之后版本开始，`cert-probe-failure` 设置已经对基于流（flow）模式的防火墙策略生效。在常规部署中，通常会使用 flow 模式（默认配置）作为防火墙的检测模式。

如果防火墙之前已存在 TLS 探测失败的的情况，在升级到 7.2.11/7.4.5 后可能会突然出现 TLS 连接问题。此时，转发流量日志将显示 Web 会话被 UTM 阻断。安全事件中的相关 SSL 日志将记录以下信息（`certificate-probe-failed`、`SSL connection is blocked due to unable to retrieve server's certificate`）：

```bash
↓↓↓TLS探测在初始阶段失败，没有具体原因↓↓↓
date=2025-10-16 time=14:20:41 eventtime=1760595641888725921 tz="+0800" logid="1700062305" type="utm" subtype="ssl"
eventtype="ssl-anomaly" level="warning" vd="root" action="block" policyid=1 poluuid="04d852a8-7f02-51f0-65cd-4c50ca8f3632"
policytype="policy" sessionid=6753380 service="HTTPS" profile="test-example" srcip=192.168.100.177 srcport=53784
srccountry="Reserved" dstip=173.243.143.6 dstport=443 dstcountry="United States" srcintf="lan" srcintfrole="lan"
dstintf="wan1" dstintfrole="wan" srcuuid="e2f430d6-7f00-51f0-8d55-e762fbba6612" dstuuid="e2f430d6-7f00-51f0-8d55-
e762fbba6612" proto=6 eventsubtype="certificate-probe-failed" hostname="173.243.143.6"

↓↓↓TLS探测在与Server交换证书阶段失败，原因是无法获取服务器证书，可能是因为线路MTU等问题↓↓↓
date=2024-10-03 time=16:00:05 id=7421548890156962227 itime="2024-10-03 16:00:05" euid=3 epid=3 dsteuid=3 dstepid=101
logflag=0 logver=704052702 devid="F2K60FTK21900878" vd="root" type="utm" subtype="ssl" level="warning"
sessionid=1749042461 policyid=1735 srcip=172.16.11.37 dstip=54.177.212.176 srcport=53508 dstport=443 proto=6
logid=1700062305 service="SSL" action="blocked" eventtime=1727964004535181139 srcintfrole="wan" dstintfrole="wan"
```

```
srcintf="port33" dstintf="port19" eventtype="ssl-anomaly" profile="certificate-inspection" hostname="fortinet.com"
msg="SSL connection is blocked due to unable to retrieve server's certificate" tz="+0200" eventsubtype="certificate-probe-
failed" srcuuid="7aaabd7a-7b3b-51ef-043f-65b562658a6d" dstuuid="83807c12-a3df-51ed-6ed9-2e2613693fe8" sni="fortinet.com"
policytype="policy" srccountry="Reserved" dstcountry="United States" poluuid="2df1b1e4-7a4d-51ef-078f-d42cf99bc294"
dtime="2024-10-03 16:00:05" itime_t=1727964005 devname="FortiGate-100F"
```

解决方法 1

解决底层 TLS 探测失败问题。如上个章节所示，在大多数情况下，通过检查 FortiGate 用于连接远程 Web 服务器的路由和网络路径，可修复 TLS 探测失败问题。

解决方法 2

如果无法解决 TLS 探测失败问题，则可以创建自定义 SSL/SSH 配置文件，修改 `cert-probe-failure` 的动作为 allow（但仍可能会产生上个章节提到的访问延迟），并在相应的防火墙策略中引用该自定义 SSL/SSH 配置文件。

重要

在 FortiOS 7.6.0 及之后版本中，`cert-probe-failure` 的默认动作已经被设置为 `allow`。

bash

```
config firewall ssl-ssh-profile
    edit "test-example"
        config https
            set cert-probe-failure allow
        end
    next
end
config firewall policy
    edit 1
        set name "to_Internet"
        set ssl-ssh-profile "test-example"
    next
end
```

重要

注意：如果在 `config ssl` 中配置了 `inspect-all`（如下所示），则 `config https` 中的 `cert-probe-failure` 选项会被禁用。

bash

```
config firewall ssl-ssh-profile
edit "test-example"
config ssl
set inspect-all certificate-inspection
end
next
end
```

解决方法 3

创建自定义 SSL/SSH 配置文件，关闭 `sni-server-cert-check`，禁用 SNI 检查，同时可以禁用 TLS 探测。并在相应的防火墙策略中引用该自定义 SSL/SSH 配置文件。出于安全性考虑，不推荐此方法。

bash

```
config firewall ssl-ssh-profile
    edit "test-example"
        config https
            set sni-server-cert-check disable
        end
    next
end

config firewall policy
    edit 1
        set name "to_Internet"
        set ssl-ssh-profile "test-example"
    next
end
```

开启 COMLog

COMLog 介绍

该功能会将 Console 下的所有输出记录到 Flash 卡的一个日志文件（4 MB）中，设备断电、重启、从 BIOS 格式化 Flash 或升级 FortiOS 后，该日志文件不会被擦除。

启用 COMLog 后，FortiGate 会将启用后所有 Console 输出记录到 Flash 中，还会每小时写入一个时间戳。

如果设备遇到突然重启或无响应等意外行为，系统日志可能无法记录到有价值的信息，而 Console 下会产生一些系统日志无法记录的日志信息，COMLog 的记录可能会提供一些线索，从而帮助工程师找到问题发生的原因。在设备正式上线前，建议在支持 COMLog 的设备上开启 COMLog 记录功能。

COMLog 支持列表

- B 系列：5001B Gen 2。
- C 系列：1000，3240，5001C，5101C。
- D 系列：300，400，500，600，800，900，1000，1200，1500，3000，3100，3200，3700，3800，3810，3815，3960，3980，5001。
- E 系列：300，500，1100，2000，2200，2500，3300，3400，3600，3960，3980，5001（也包含带硬盘型号）。
- F 系列：400，600，1000，1800，2200，2600，3000，3500，3700，4200，4400，4800，6300，6500，7121（也包含带硬盘型号）。
- G 系列：700，900（也包含带硬盘型号）。
- VM 与其他型号：不支持。

开启 COMLog

1. 开启/关闭 COMLog。

```
diagnose debug comlog <enable|disable>
```
bash

2. 查看 COMLog 开启状态。

```
diagnose debug comlog info
```
bash

3. 清除记录在 Flash 中的所有 COMLog 日志。

```
diagnose debug comlog clear
```
bash

查看 COMLog

CLI 查看

1. 配置 Console 输出方式为 Standard，一次显示所有输出。

```
config system console
    set output standard
end
```
bash

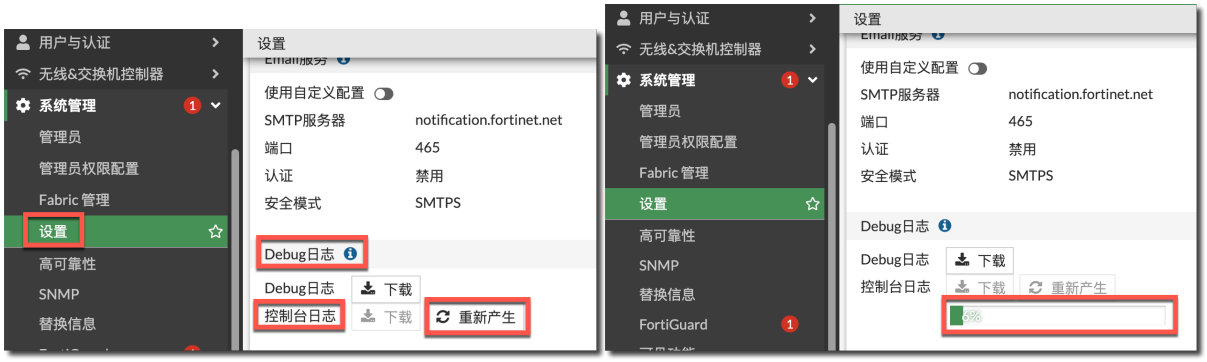
2. 在 CLI 下查看已记录的 COMLog，并复制到文本文件中。

> **重要**
>
> 由于 Console 的输出速率有限，建议使用 SSH 或 GUI 来收集 COMLog 输出，不要使用 Console 来输出 COMLog。

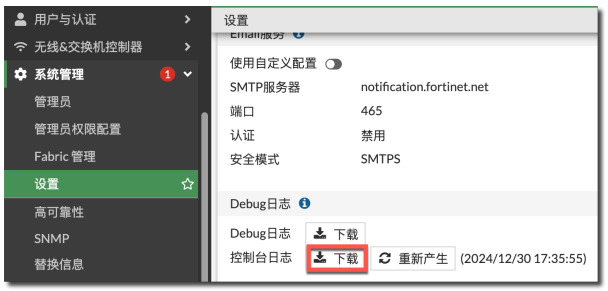
```
diagnose debug comlog read
```
bash

GUI 下载

1. 开启 COMLog 后，进入“系统管理 → 设置”的“Debug 日志”部分，点击“控制台日志”右侧的“重新产生”按钮。FortiGate 会开始将 Flash 中的 COMLog 复制到 GUI 缓存中以供下载。



2. 等待 COMLog 生成进度完成，点击“下载”按钮。



3. 通过 GUI 下载到的 COMLog 文件后缀名为.log，内容为 Zlib 压缩格式，必须使用 Zlib 解压工具（如 zlib-flate）进行解压，才能看到明文的 COMLog。

注意事项

1. 启用 COMLog 时，请避免配置密码或密钥等机密配置，因为这些密码和密钥会以明文方式记录在 COMLog 中。可以在配置密码和密钥等配置前禁用 COMLog，配置完成后再启用 COMLog。
2. 由于 Console 的输出速率有限，建议使用 SSH 或 GUI 来收集 COMLog 输出，不要使用 Console 来输出 COMLog。

运维注意事项

请参考运维注意事项↗。

系统管理

管理方式

- 语言/时区切换
- Web 管理
- 消除 Web 告警或提示

  消除 Web 证书告警　关闭 Web 提示信息　文件系统检查
- Console 管理
- SSH/Telnet 管理
- CLI 输出模式
- GUI/CLI 连接问题

固件与配置管理

- 固件版本管理

  固件版本　产品生命周期　升级路径　Web 界面升级　FTP/TFTP/USB 方式升级　通过 FortiGuard 升级　固件切换

  Flash 刷机　软件签名增强　G 系列修改安全级别　过期 License 升级限制　有效 License 自动升级　过期 License 强制升级
- 配置管理

  配置备份及恢复　备份配置文件到 Flash　备份不包含密码的配置文件　通过 SCP 备份配置文件　恢复出厂　配置回退

管理员配置

- 管理员设置
- 远程认证管理员
- 远程认证与本地认证的优先级
- 管理员登录锁定
- 管理员密码策略
- 通过 FortiCloud 账户登录 FortiGate
- 限制管理员 SSH/Telnet 跳板
- 密码恢复
- API 用户

  API 文档获取　API 用户配置　API 常见问题

SNMP

- SNMP 介绍
- SNMP 公共 MIB 和私有 MIB
- SNMP 建议监控指标
- SNMP 的常用工具
- SNMP 的设置
- SNMP 与管理员可信主机
- SNMP 的 HA 独立管理
- 基于 VDOM 的 SNMP

FortiGuard 管理

- FortiGuard 服务介绍
- License 到期的影响
- 服务指南

  FortiCare 服务介绍　设备注册步骤　设备强制注册　服务注册步骤　PRMA 高级服务注册步骤　产品服务续费　RMA 服务迁移

  RMA/DOA 硬件保修政策　产品注册信息变更　远程技术支持　FortiEDR Cloud 注册指南
- 在线更新特征库
- 离线更新特征库
- 离线导入设备 License
- FortiGuard 更新服务器位置
- 查看合同/更新状态
- FortiGuard 连接问题
- 禁用云通信
- 特征库误报/缺失上报

FortiManager 管理

- FortiManger 功能简介
- FortiManger 产品介绍
- FortiManger 版本推荐
- FortiManger 网管登陆
- FortiManger ADOM
  - ADOM 简介
  - 开启 ADOM
  - ADOM 的高级特性
- FortiManger 管理员
  - 管理员账号管理
  - 配置远程管理员
  - 管理员和 ADOM 关联
  - ADOM 的 Workspace 模式

系统设置

- 证书管理
- Email 服务
- 页面替换
- 登录提示信息

常用基础命令

常用基础命令

命令结构

```
config      Configure object.                          对策略，对象等进行配置
get         Get dynamic and system information.        查看相关关对象的参数信息
show        Show configuration.                        查看配置文件
diagnose    Diagnose facility.                         诊断命令
execute     Execute static commands.                   常用的工具命令，如 ping
exit        Exit the CLI.                              退出
```

常用命令

常用配置命令

1. 配置接口地址

```
FortiGate # config system interface
FortiGate (interface) # edit lan
FortiGate (lan) # set ip 192.168.100.99/24
FortiGate (lan) # end
```

2. 配置静态路由

```
FortiGate # config router static
FortiGate (static) # edit 0                   //“ 0 ”系统自动分配路由条目的序号
FortiGate (0) # set device wan1               //出接口
FortiGate (0) # set dst 10.0.0.0 255.0.0.0    //目的网段
FortiGate (0) # set gateway 192.168.57.1      //网关IP
FortiGate (0) # end
```

3. 配置默认路由

```
FortiGate # config router static
FortiGate (static) # edit 0
FortiGate (0) # set gateway 192.168.57.1
FortiGate (0) # set device wan1
FortiGate (0) # end
```

4. 添加 IP 地址对象

```
FortiGate # config firewall address
FortiGate (address) # edit clientnet
new entry 'clientnet' added
FortiGate (clientnet) # set subnet 192.168.1.0 255.255.255.0
FortiGate (clientnet) # end
```

5. 添加 IP 池

```
FortiGate# config firewall ippool
FortiGate (ippool) # edit nat-pool
new entry 'nat-pool' added
FortiGate (nat-pool) # set startip 100.100.100.1
FortiGate (nat-pool) # set endip 100.100.100.100
FortiGate (nat-pool) # end
```

6. 添加虚拟 IP

```
FortiGate # config firewall vip
FortiGate (vip) # edit webserver
new entry 'webserver' added
FortiGate (webserver) # set extip 202.0.0.167
FortiGate (webserver) # set extintf wan1
```

```
FortiGate (webserver) # set mappedip 192.168.0.168
FortiGate (webserver) # end
```

7. 配置上网策略

```
FortiGate # config firewall policy
FortiGate (policy) # edit 1
FortiGate (1)#set srcintf internal     //源接口
FortiGate (1)#set dstintf wan1         //目的接口
FortiGate (1)#set srcaddr all          //源地址
FortiGate (1)#set dstaddr all          //目的地址
FortiGate (1)#set action accept        //动作
FortiGate (1)#set schedule always      //时间
FortiGate (1)#set service ALL          //服务
FortiGate (1)#set logtraffic disable   //日志开关
FortiGate (1)#set nat enable           //开启nat
FortiGate (1)#end
```

8. 配置 VIP 映射策略

```
FortiGate # config firewall policy
FortiGate (policy) #edit 2
FortiGate (2)#set srcintf wan1      //源接口
FortiGate (2)#set dstintf internal  //目的接口
FortiGate (2)#set srcaddr all       //源地址
FortiGate (2)#set dstaddr ngfw1     //目的地址，虚拟VIP映射，需要先配置好VIP条目才可以被调用
FortiGate (2)#set action accept     //动作
FortiGate (2)#set schedule always   //时间
FortiGate (2)#set service ALL       //服务
FortiGate (2)#set logtraffic all    //日志开关
end
```

常用 show 命令

1. 查看设备信息、主机名，管理端口

```
FortiGate # show system global
```

2. 查看系统状态信息和当前资源信息

```
FortiGate # get system status
FortiGate # get system performance status
```

3. 查看应用流量统计

```
FortiGate # get system performance firewall statistics
```

4. 查看 ARP 表

```
FortiGate # get system arp
```

5. 查看 arp 丰富信息

```
FortiGate # diagnose ip arp list
```

6. 清除 ARP 缓存

```
FortiGate # execute clear system arp table
```

7. 查看当前会话表

```
FortiGate # diagnose sys session stat
FortiGate # diagnose sys session full-stat
```

8. 查看会话列表

```
FortiGate # diagnose sys session list
```

9. 查看物理接口状态

```
FortiGate # get system interface physical
```

10. 查看默认路由配置

```
FortiGate # show router static
```

11. 查看路由表中的静态路由

```
FortiGate # get router info routing-table static
```

12. 查看 OSPF 相关配置

```
FortiGate # show router ospf
```

13. 查看全局路由表

```
FortiGate # get router info routing-table all
```

14. 查看 HA 状态

```
FortiGate # get system ha status
```

15. 查看主备机是否同步

```
FortiGate # diagnose sys ha checksum
```

常用诊断命令

1. IPSec IKE

```
FortiGate # diagnose debug application ike -1
FortiGate # diagnose debug enable
```

常用 execute 命令

```
FortiGate #execute  ping  8.8.8.8                          //常规ping操作
FortiGate #execute  ping-options source  192.168.1.200    //指定ping数据包的源地址 192.168.1.200
FortiGate #execute  ping  8.8.8.8                          //继续输入ping的目标地址，即可通过192.168.1.200的源地址执行ping操作
FortiGate #execute  traceroute   8.8.8.8
FortiGate #execute  telnet 2.2.2.2      //进行telnet访问
FortiGate #execute  ssh  2.2.2.2        //进行ssh 访问
FortiGate #execute  factoryreset        //恢复出厂设置
FortiGate #execute  reboot              //重启设备
FortiGate #execute  shutdown            //关闭设备
```

管理方式

语言/时区切换

Web 管理

消除 Web 告警或提示

- 消除 Web 证书告警
- 关闭 Web 提示信息
- 文件系统检查

Console 管理

SSH/Telnet 管理

CLI 输出模式

GUI/CLI 连接问题

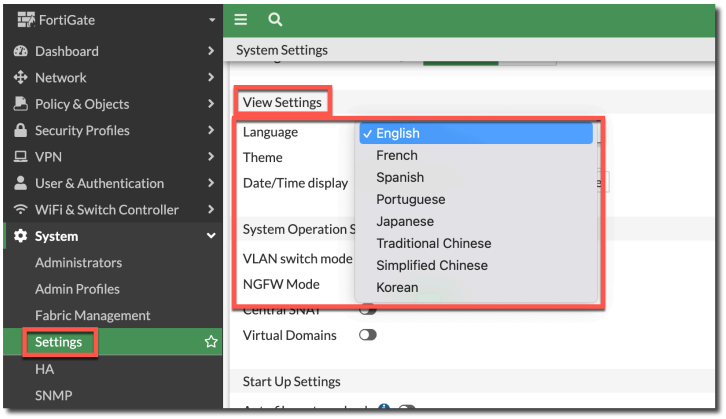
语言/时区切换

简介

- 设备 Web 界面初始语言为英文，可更改为中文（或其他语言）界面。
- 设备默认时区为 `(UTC-12:00) Etc/GMT+12`，可更改为 +8 区（或其他区）时间。

语言切换

登录 Web 界面后，进入 System → Settings，找到 View Settings，修改 Language 为 Chinese 即可：

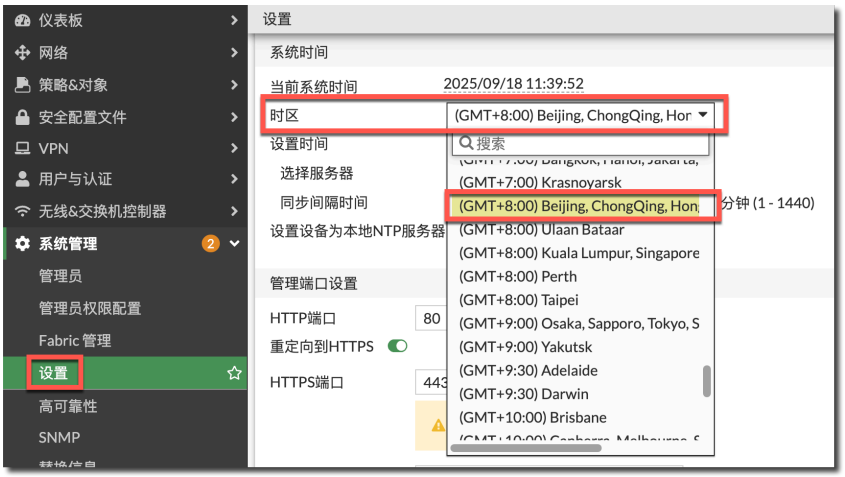



```
FortiGate # config system global
FortiGate (global) # set language ?

english       English.
french        French.
spanish       Spanish.
portuguese    Portuguese.
japanese      Japanese.
trach         Traditional Chinese.
simch         Simplified Chinese.
korean        Korean.
FortiGate (global) # set language simch
FortiGate (global) # end
```

时区切换

FortiOS 7.4 之前版本

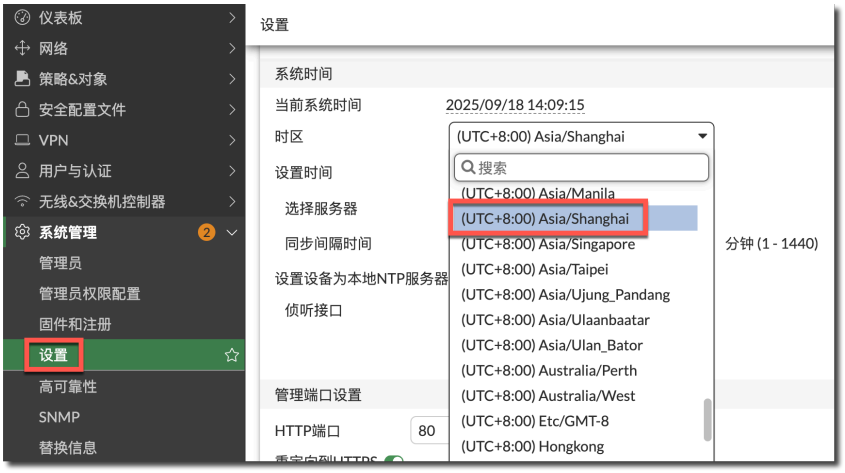
进入“系统管理 → 设置”页面，在“系统时间”处修改。



```
config system global
    set timezone 55
end
```

FortiOS 7.4 及之后版本

进入“系统管理 → 设置”页面，在“系统时间”处修改。



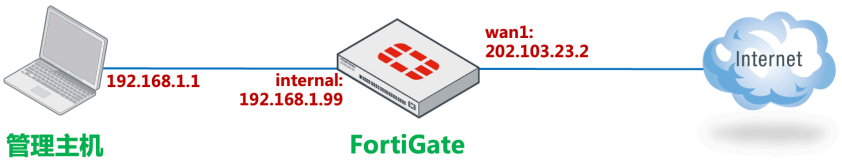

```
config system global
    set timezone "Asia/Shanghai"
end
```

Web 管理

需求

通过 web 可视化操作界面对防火墙进行配置管理，配置 wan1 口的管理功能。

组网拓扑

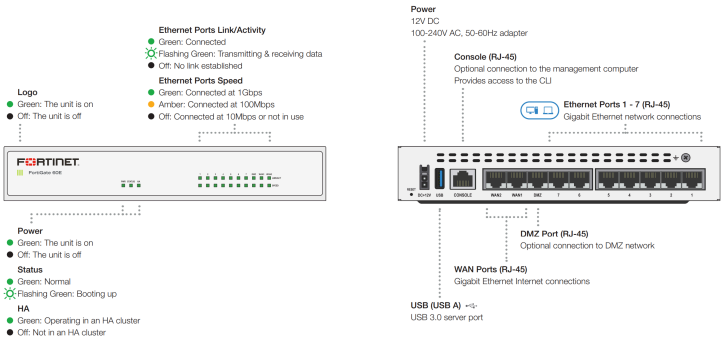


配置要点

1. FortiGate 出厂配置默认地址为 192.168.1.99，可以通过 https 的方式进行 web 管理（默认用户名 admin，密码为空），不同型号设备用于管理的接口略有不同。

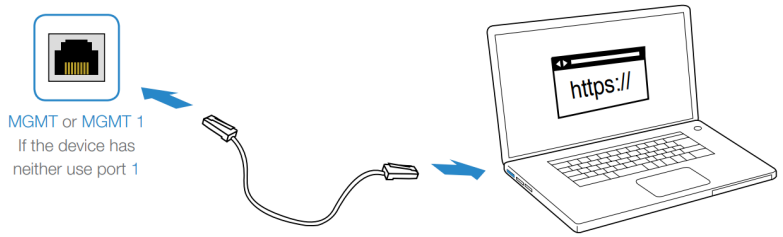
> **重要**
>
> FortiGate-60F、FortiGate-40F 等所有的交换口隶属于三层口 internal 下，默认配置下，仅有 internal 接口可以进行三层配置，例如 IP 配置。
>
> 如需对交换接口（如 internal/lan）下的成员接口配置 IP，需要将其从交换接口中拿出（取消成员关系）后再单独配置。
>
> 具体的接口管理 IP 分配需要参考设备出厂包装盒里面的：QuickStart Guide 和 FortiGate Supplement 文档。或者通过 docs 网站文档搜索获取具体型号的接口信息和登陆管理信息链接：
>
> https://docs.fortinet.com/product/fortigate/hardware ↗
>
> https://docs.fortinet.com/document/fortigate/hardware/fortigate-fortiwifi-60e-poe-qsg-supplement?model=all ↗
>
> https://docs.fortinet.com/document/fortigate/hardware/fortigate-quickstart-guide-high-end?model=all ↗

2. 举例：FortiGate-60E，默认配置下，1、2、3、4、5、6、7 均属于 internal 交换接口。



1 FortiGate Setup

A Web Browser with Ethernet Cable



To Connect to the GUI

1. Connect the Ethernet cable.
2. Configure the management computer to be on the same subnet as the internal interface as the FortiGate unit:

   IP address: 192.168.1.2
   Netmask: 255.255.255.0

3. Visit 192.168.1.99 in your web browser.
4. Login using username "admin" and no password.
5. Configure your device and save your settings.
6. Register your device from the dashboard page.

3. 将电脑 IP 地址设置为 192.168.1.2/24（或同网段除 192.168.1.99 的地址），并连到 internal 口（举例：FortiGate-60E，默认配置下，1、2、3、4、5、6、7 均属于 internal 交换接口）或者 MGMT 口，管理主机打开浏览器，输入 `https://192.168.1.99`，用户名：admin，密码：空，进入 FortiGate 设备页。

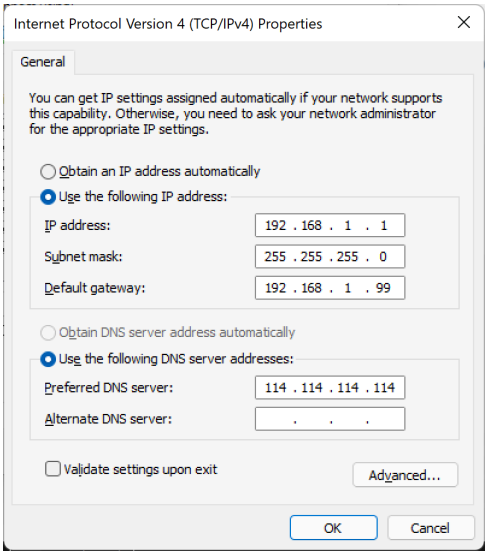
> **重要**
>
> 若密码修改后忘记请参照系统管理 → 管理员配置 → 密码恢复 ↗章节进行恢复初始密码。

4. 登陆设备后按需配置其他接口的 IP 和管理功能。默认其他接口是没有 IP 地址，也未开启 HTTPS 等其他管理功能。若防火墙接口地址修改过，但忘记相应地址，可进命令行查看当前配置，参考系统管理 → 常用基础命令 ↗" 章节。

> **重要**
>
> 浏览器建议采用 Firefox、Google Chrome、Microsoft Edge。

配置步骤

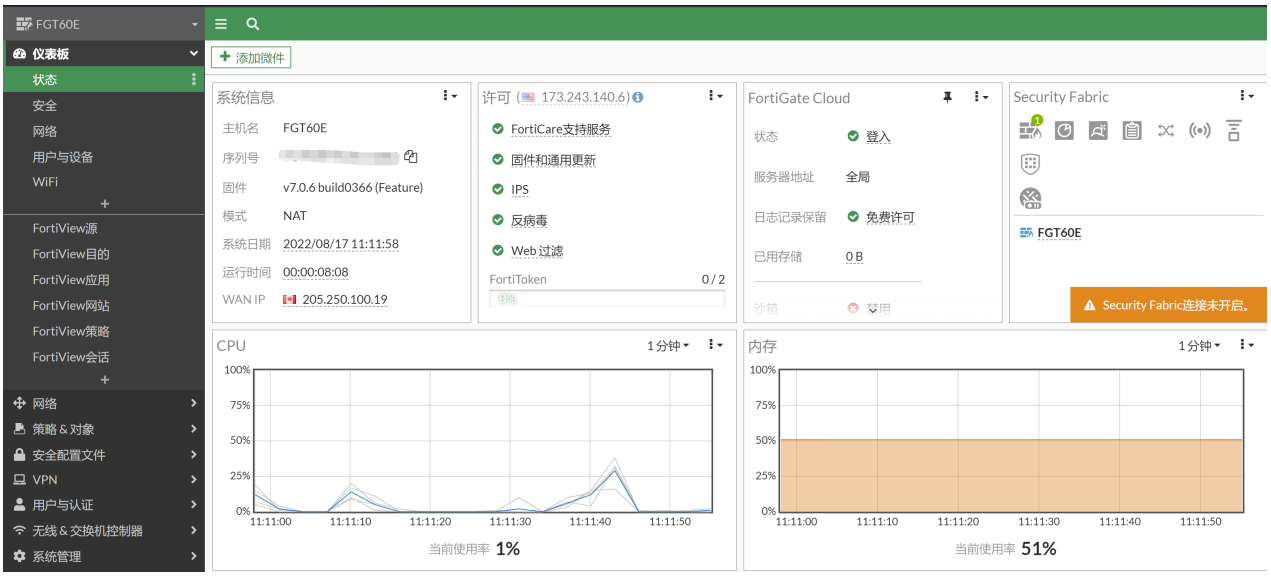
1. 在 FortiGate 出厂配置情况下，将电脑设置为 192.168.1.1，网关设置为 192.168.1.99，如下图：



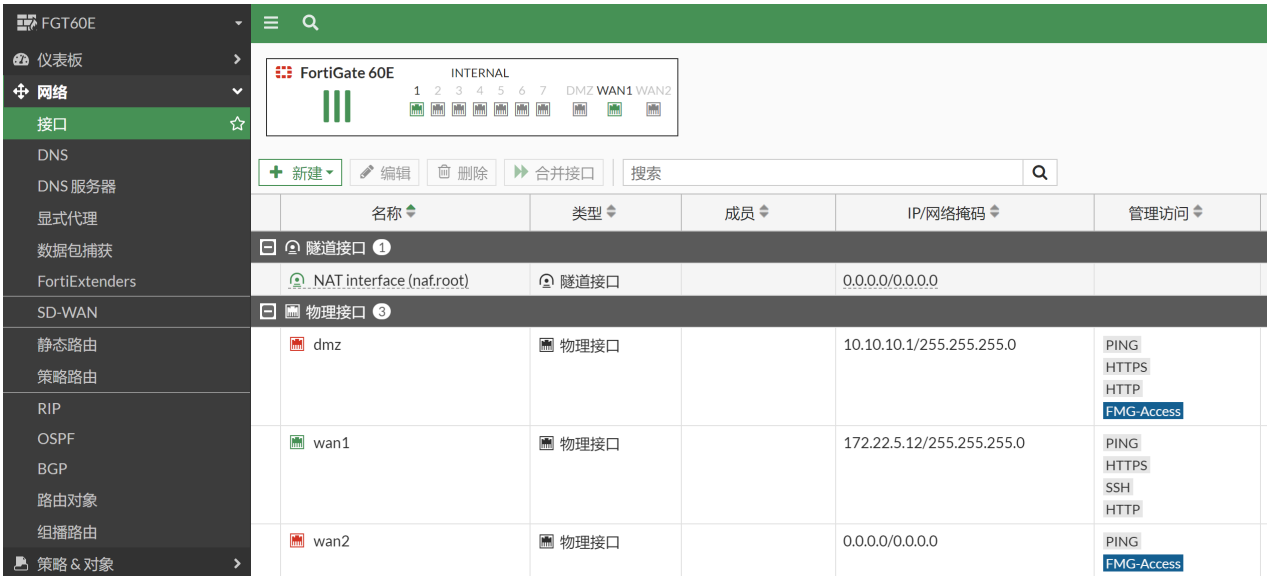
2. 在 IE 浏览器里输入 `https://192.168.1.99` 将出现防火墙登陆的页面。

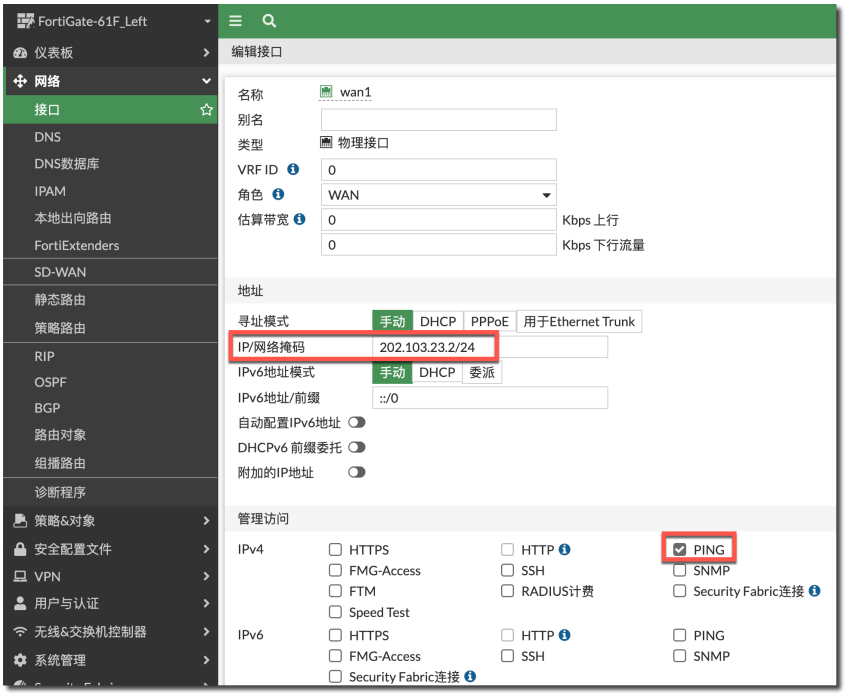


3. 输入用户名 admin 密码默认空，按照提示设置新的管理员密码后，即可进入防火墙首页。



4. 进入“网络 → 接口”页面，双击 wan1 接口进行编辑，配置接口地址为：202.103.23.2/24，管理访问：勾选 PING。





具体各种方式含义如下：

- HTTPS：允许用户用 `https://202.103.23.2`，进行管理。
- PING：允许用户 ping 此接口地址，如果不勾选，在路由可达的情况下也 Ping 不通。
- HTTP：允许用户用 `http://202.103.23.2`，进行管理。
- SSH：允许用户用用 SSH 192.168.0.99 方式管理设备。
- SNMP：允许用户通过该接口对设备进行 SNMP 管理。

验证效果

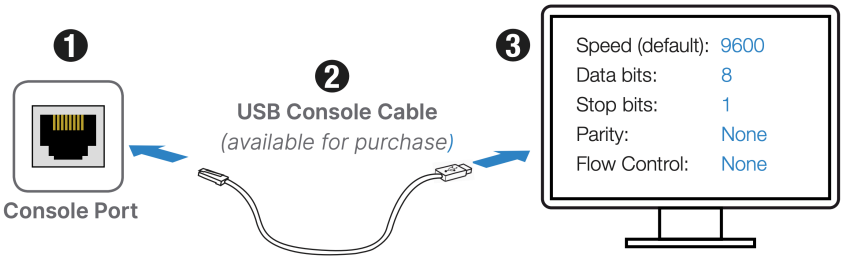
1. 电脑直连 Internal 接口的成员接口，电脑网卡 IP 修改为 192.168.1.2，在浏览器里输入：`https://192.168.1.99`，即可通过 internal 成员接口登陆管理防火墙。
2. 开启 wan1 的 PING 管理协议后，也可以 Ping 通 wan1 接口。

Console 管理

功能需求

若需要进入命令行进行配置管理，可通过 Concole 线，使用 Putty/SecureCRT/XSHELL 等终端软件进入 CLI 界面，防火墙默认允许 Concole 管理。

组网拓扑

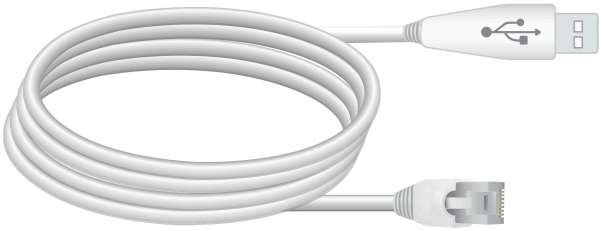


配置要点

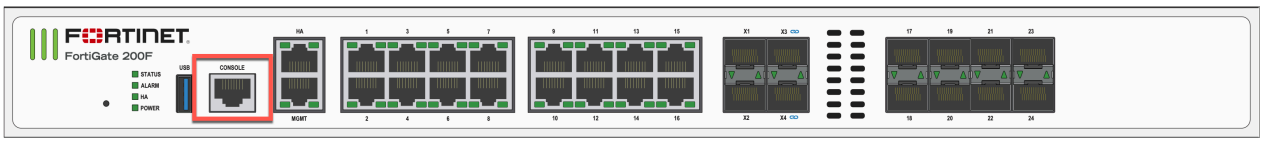
1. 准备配置线及可连 Console 线的 PC（根据不同的 Console 线种类，可能需要自行安装 Console 线驱动，部分型号的 FortiGate 会随箱包含一条 USB → Console 线缆）。
2. 连接 Console 线。Console 线的水晶头端连到设备的 concole 口上，另一头连电脑的 USB 接口。
3. 配置终端软件，如 Putty、SecureCRT、XSHELL。

配置步骤

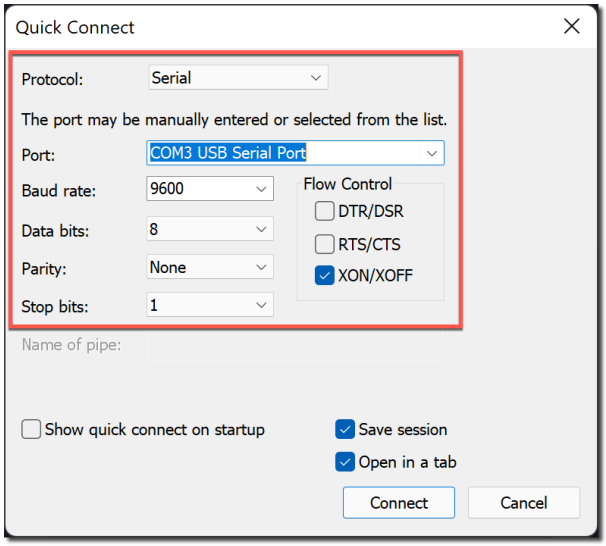
1. 准备 Console 线及 PC。Console 线一头类似网线水晶头（接在 FortiGate 的 Console 口上），另一头为 USB 接口，如下图所示。



2. 连接配置线，将 Console 线水晶头一端插入网络设备的 Console 接口（Console 口一般位于网络设备的网络接口旁，其上都会标有字母 CONSOLE），然后将 Console 线的 USB 接口端连接 PC。

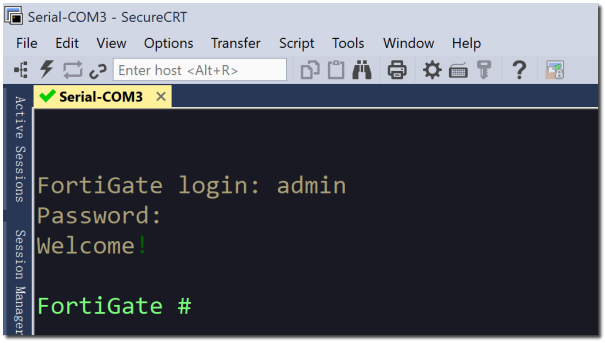


3. 连好配置线后，配置终端软件，Protocol 选择“Serial”，Baud rate 设置为 9600，Data bits 设置为 8，Stop bits 设置为 1，这里以 SecureCRT 为例。



结果验证

按回车键，则会打印出“FortiGate login：”提示输入用户名及密码（设备出厂默认用户名为 admin 密码为空，如果密码有改过，且忘记了，可参考"系统管理 → 管理员配置 → 密码恢复"恢复密码。



注意事项

若设置后无法进入命令行，请检查：

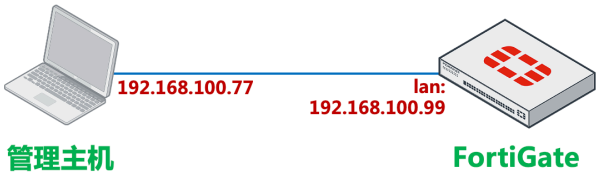
- Console 线是否连接在 concole 口。
- 终端软件的波特率是否配置为 9600，Data bits 设置为 8，Stop bits 设置为 1。
- 在“硬件管理器”中检查 COM 接口是否为“未识别设备”，如果是，可能是缺少驱动程序，需要安装 Console 线驱动程序（Windows 10/Windows 11 一般不需要安装，驱动程序可从 Console 线经销商处获取）。
- 如果以上操作都对还是无法进入，请尝试更换电脑、Console 线及终端软件，或联系 Fortinet 技术支持。

SSH/Telnet 管理

功能需求

当需要进设备的命令行进行配置或收集信息，但没有配置线，或不在设备旁边时，可通过远程 SSH 或 Telnet 对设备进行管理。

组网拓扑



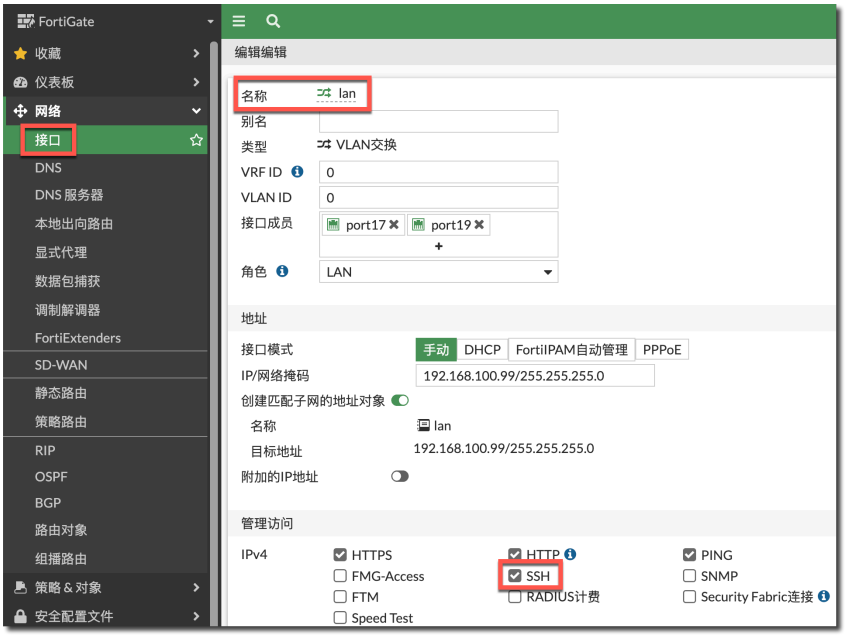
配置要点

使用 SSH/Telnet 方式管理首先要保证管理主机到设备的接口地址的连通性，可把接口的 PING 功能勾选，若能 PING 通管理接口则说明连通性没问题：

1. 开启接口 SSH、TELNET 功能。
2. SSH 管理设备。
3. TELNET 管理设备。

配置步骤

1. 开启接口 SSH 管理功能，进入“系统管理 → 网络 → 接口”，编辑 lan（双击 lan 接口）接口，在“管理访问”中，勾选 SSH。



2. 在 7.0 以后版本中，Telnet 管理必须通过 CLI 开启，由于 Telnet 为明文传输，从安全性考虑，建议只开启 SSH 即可。

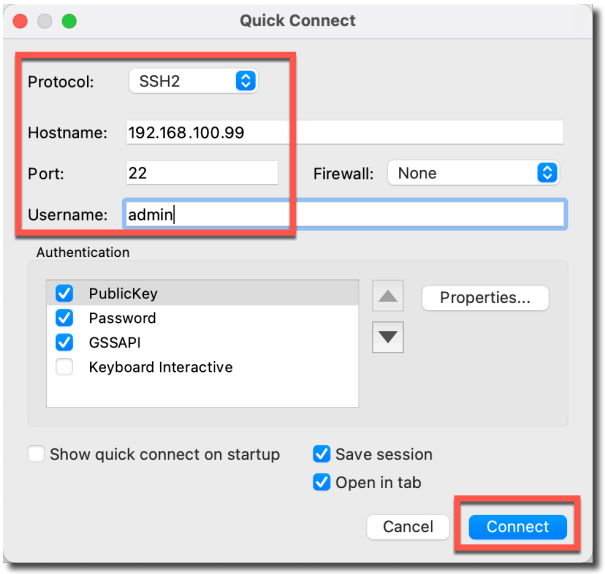
> **重要**
>
> 添加 Telnet 管理协议时，使用 append 命令将 telnet 协议附加上去即可（不会改变原有的管理协议），不要使用 set 命令，使用“set allowaccess telnet”会将原有允许的管理协议清空。

```
FortiGate # config system interface
FortiGate (interface) # edit lan
FortiGate (lan) # append allowaccess telnet
FortiGate (lan) # end
```

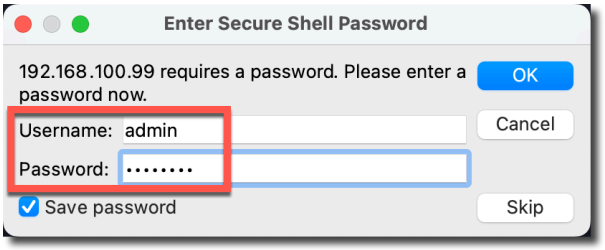
结果验证

SSH 管理

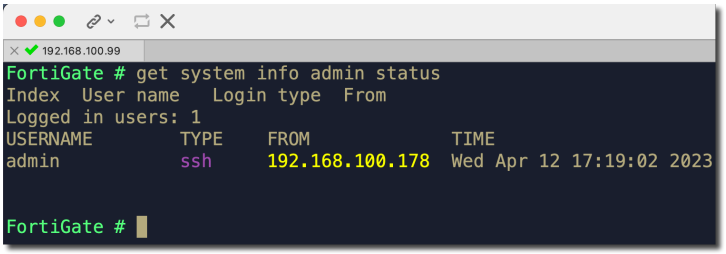
1. 使用终端软件（如 SecureCRT）新建一个 SSH 连接，以下以 Secure CRT 软件为例。Protocol 选择 SSH2，Hostname 输入设备的管理地址 192.168.100.99（即 lan 口对应的 IP 地址），端口号 22，其他保持默认值，点击 Connect。



2. 输入管理员密码（用户名密码与 Web 登录的一致），点击 OK。

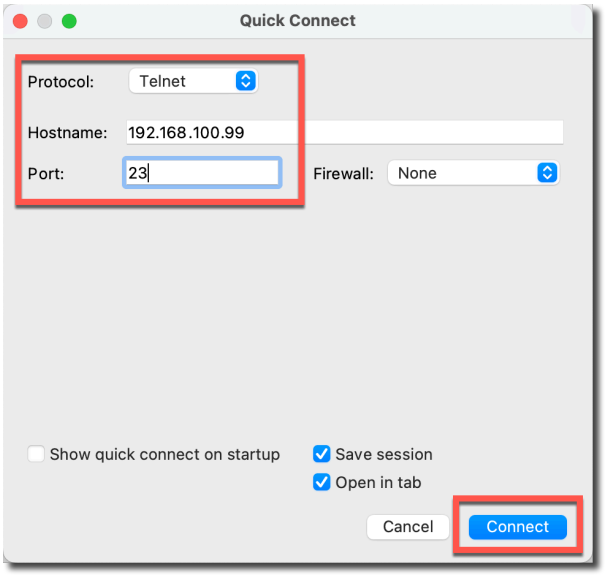


3. 登录设备进行 CLI 管理。

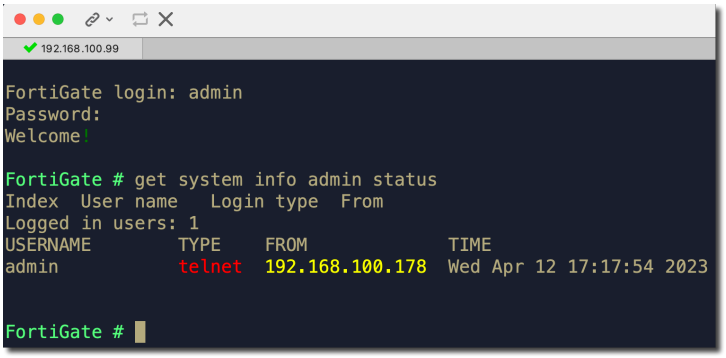


Telnet 管理

1. 使用终端软件（如 SecureCRT）新建一个 SSH 连接，以下以 Secure CRT 软件为例。Protocol 选择 Telnet，Hostname 输入设备的管理地址 192.168.100.99（即 lan 口对应的 IP 地址），端口号 23，其他保持默认值，点击 Connect。



2. 输入用户名和密码（用户名密码与 Web 登录的一致），登录设备进行 CLI 管理。



CLI 输出模式

功能简介

在 FortiGate 的 CLI 中，默认的输出行为为分页显示，即在输出内容达到屏幕上限时暂停显示，直到按下键盘上的某个键才能继续显示。这种分页模式对于 CLI 的输出非常常见，但如果希望查看完整的命令输出，或者将输出保存到日志文件中，则分页模式会频繁显示 `--More--`，影响阅读体验。

FortiGate 的 CLI 输出模式可以设置为 standard 模式（无暂停）或 more 模式（在屏幕满时暂停，按任意键继续）。此设置仅适用于 CLI 命令（Console、Web CLI、SSH/Telnet）的 `show`、`?` 或 `get` 命令，而不影响 Debug 或 Sniffer 输出。

配置方法

```
config system console
    set output {standard | more}
end
```

- `standard`：移除输出中的暂停，直接输出全部内容。
  - 当需要通过 CLI 一次收集完整输出（如通过自动化收集），不希望输出受到分页干扰时，可以启用标准模式。
  - 适用于通过 SSH 或其他 CLI 操作来执行配置文件导出等任务时，确保无间断输出。

```bash
FortiGate # show
......
config system global
    set admintimeout 60
    set alias "FortiGate-101F"
    set gui-auto-upgrade-setup-warning disable
    set gui-ipv6 enable
    set hostname "FortiGate-101F"
    set language simch
    set switch-controller enable
    set timezone "Asia/Shanghai"
    set virtual-switch-vlan enable
end
config system accprofile
    edit "prof_admin"
        set secfabgrp read-write
        set ftviewgrp read-write
        set authgrp read-write
        set sysgrp read-write
        set netgrp read-write
        set loggrp read-write
        set fwgrp read-write
        set vpngrp read-write
        set utmgrp read-write
        set wanoptgrp read-write
        set wifi read-write
        set cli-get enable
        set cli-show enable
        set cli-exec enable
        set cli-config enable
    next
......
```

- `more`（默认）：分页模式，逐页查看输出结果，按任意键继续显示后续内容。

```bash
FortiGate # show
....
config system global
   set admin-scp enable
   set admin-sport 4443
   set admin-telnet disable
   set autorun-log-fsck enable
   set cfg-save manual
   set device-idle-timeout 60
   set gui-local-out enable
   set gui-theme onyx
```

```
    set management-port-use-admin-sport disable
    set proxy-auth-timeout 5
    set switch-controller enable
    set sys-perf-log-interval 1
    set timezone "Europe/Amsterdam"
end
config system accprofile
    edit "prof_admin"
        set secfabgrp read-write
        set ftviewgrp read-write
        set authgrp read-write
        set sysgrp read-write
        set netgrp read-write
        set loggrp read-write
 --More--
```

总结

- 此设置是全局生效的，因此可能会对其他管理员产生影响，特别是在 Console 上使用时，由于 Console 打印速度较慢，如果没有中断的输出，执行如 `show` 的命令时，将会显示完整配置，直到输出完成为止。
- 该设置对 FortiManager 的 CLI 脚本特别有用，尤其是在通过 FGFM 隧道应用配置更改时。使用此设置可以确保配置不会因为分页而被中断。
- 此设置不影响 Debug 或 Sniffer 输出。

GUI/CLI 连接问题

说明

当 FortiGate 出现 Web GUI（HTTP/HTTPS）或 CLI（SSH/Telnet）无法登录/连接时，可按本文进行基础的检查与抓包定位。

排查步骤

1. 确认管理 IP/接口是否正确：先通过 Console 登录 FortiGate，确认你访问的管理 IP 属于哪个接口、接口当前地址是否正确。

```bash
show system interface mgmt

config system interface
    edit "mgmt"
        set vdom "root"
        set ip 11.10.10.253 255.255.255.0
        set allowaccess ping https ssh
        set type physical
        set dedicated-to management
        set role lan
        set snmp-index 2
    next
end
```

2. 确认管理服务端口是否被改过：很多“连不上”的问题其实是端口不对（例如 HTTPS 不再是 443）。常见默认值示例：

- `set admin-port 80`（HTTP）
- `set admin-sport 443`（HTTPS）
- `set admin-ssh-port 22`（SSH）
- `set admin-telnet-port 23`（Telnet）

```bash
show full-configuration system global | grep 'set admin-\(port\|sport\|ssh-port\|telnet-port\)'

    set admin-port 8081
    set admin-sport 4443
    set admin-ssh-port 2222
    set admin-telnet-port 2323
```

3. 确认接口层是否放行管理访问（`allowaccess`）：即使端口正确，如果接口没开启 HTTPS/SSH，也会无法访问。

```bash
show system interface mgmt | grep allowaccess

set allowaccess ping https ssh
```

4. 检查是否配置了管理员可信主机（trusthost）：如果管理员账号设置了可信主机，只有在可信名单内的源 IP 才能登录（详情参考：系统管理 → 管理员配置 → 管理员设置↗章节。

- 客户端到 FortiGate 的路径上是否发生 SNAT。
- 若发生 SNAT，需要把 NAT 后的源 IP 加入 trusthost，否则会被拒绝。

```bash
show system admin

config system admin
    edit "admin"
        set vdom "root"
        set trusthost1 192.168.100.178 255.255.255.255
        set accprofile "super_admin"
        set password ENC
PB2BUdKgOSS1ILNihB8s0TQOeY3mp/Pny1J0j6eQhSx1xODelJYY7SJv8b7pnZ0v9wUGUgXlSx7m1GkNttonTpQjUQQ6g4JqkjJoWuw6ynGKow=
    next
end
```

5. 检查是否有 Local-in Policy 限制管理访问：如果配置了 Local-in Policy，可能会把管理流量挡掉（详情参考：策略与对象 → 本地流量策略 → Local-in 策略↗章节。

```bash
show firewall local-in-policy

config firewall local-in-policy
    edit 1
        set uuid 6f31b814-e943-51f0-e47b-b2c1eb16e4ad
        set intf "mgmt"
        set srcaddr "192.168.100.0/24"
        set dstaddr "all"
        set action accept
        set service "ALL"
        set schedule "always"
    next
    edit 2
        set uuid 056c756c-e944-51f0-318c-5c49729dd5e3
        set intf "mgmt"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set schedule "always"
    next
end
```

6. 若仍无法解决：用 Debug Flow/Sniffer 定位：用于判断：流量是否到达、防火墙是否丢弃、服务进程是否报错等：
   - Debug 与 Debug Flow：

     ```bash
     diagnose debug reset
     diagnose debug console timestamp enable

     diagnose debug app httpsd -1     ←—— Debug GUI
     diagnose debug app sshd -1       ←—— Debug SSH

     diagnose debug flow filter addr <Client_Source_IP>
     diagnose debug flow show function-name enable
     diagnose debug flow trace start 100
     diagnose debug enable

     Debug结束后执行:
     diagnose debug disable
     diagnose debug reset
     ```

   - Sniffer（抓包确认是否有回包/是否到达）：

     ```bash
     diagnose sniffer packet any "host <Client_Source_IP>" 6 0 a
     ```

7. 尝试重启 `httpsd` 进程。

```bash
fnsysctl killall -11 httpsd
```

8. 若怀疑 443 端口冲突，可将 管理员 HTTPS 端口改为自定义端口进行验证与规避。

```bash
config system global
    set admin-sport 8443
end
```

9. GUI 异常，可按 Fortinet 建议收集浏览器调试信息（使用 Fortinet Support Tool 插件，详情参考：故障排查 → FortiGate Support Tool ↗ 章节）。

消除 Web 告警或提示

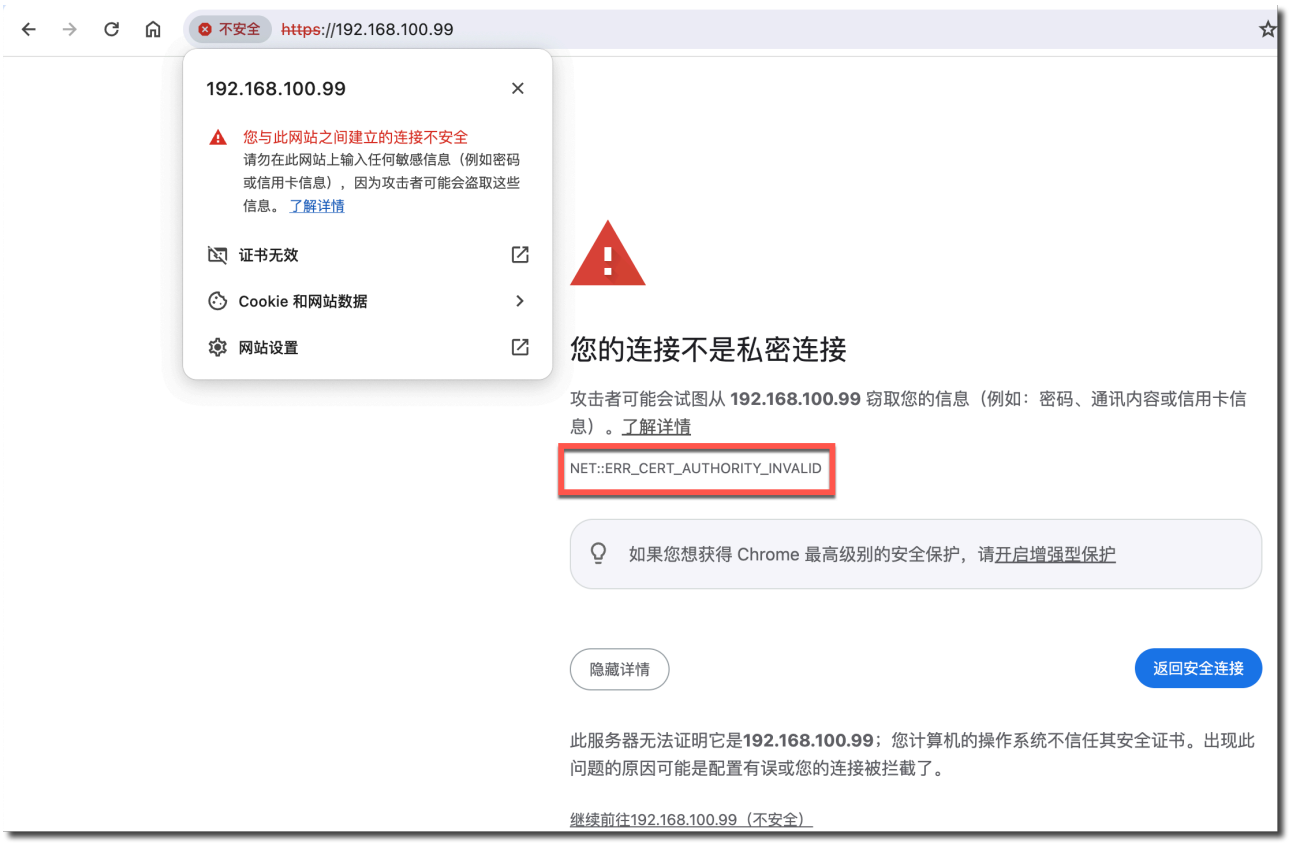
- 消除 Web 证书告警
- 关闭 Web 提示信息
- 文件系统检查

消除 Web 证书告警

问题现象

默认配置下，访问 FortiGate 的 Web 管理页（GUI）面时，浏览器会弹出告警提示，这是由于 FortiGate 默认使用的是自签名证书，CA 为 FortiGate 自身，浏览器默认情况下只信任公网权威 CA 机构颁发的证书，所以自签名的证书会被浏览器认为是不被信任的证书。

> **重要**
>
> 即使用户在管理 PC 信任了 FortiGate Web 管理页面证书对应的 CA，但由于该证书可能没有 SAN（Subject Alternative Name）字段（也可能是 SAN 属性中没有包含 FortiGate 的管理 IP 或域名），访问 FortiGate Web 管理页面时，浏览器依然会提示告警信息。



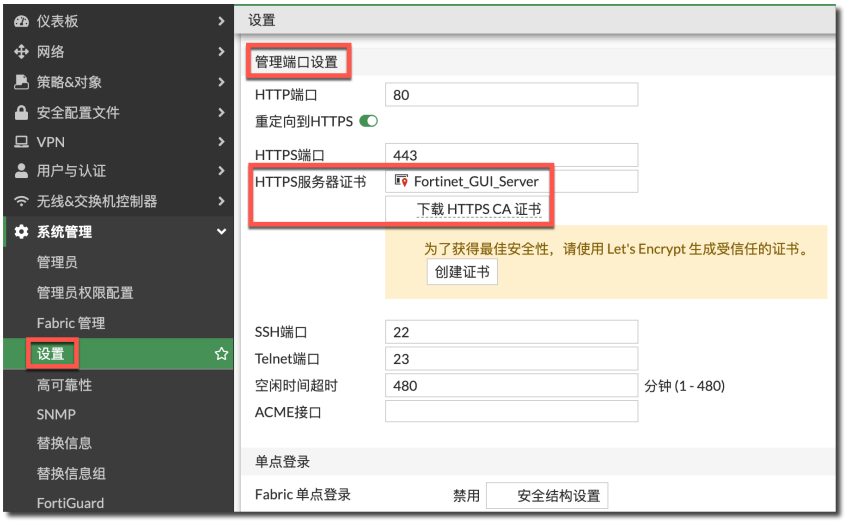
解决方法 1 - 使用 Fortinet_GUI_Server 证书

> **重要**
>
> 该方法在 FortiOS 7.2.1 及以上版本支持。

下载 CA 证书

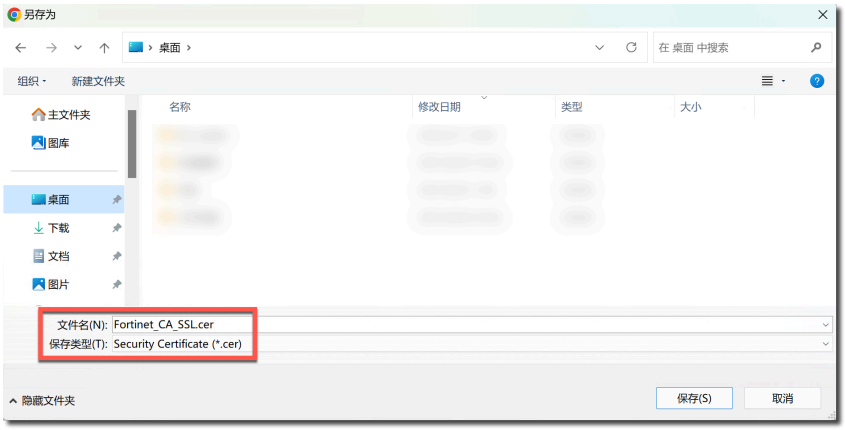
1. 在 FortiGate 的“系统管理 → 设置”页面的“管理端口设置”部分中，确认“HTTPS 服务器证书”选择为“Fortinet_GUI_Server”证书（默认配置会选择此证书）。

   > **重要**
   >
   > - 如果“HTTPS 服务器证书”选择的不是“Fortinet_GUI_Server”证书，请先选择“Fortinet_GUI_Server”证书后点击页面下方的“应用”按钮下发配置。
   > - 如果是 HA 环境，从所有 HA 成员下载的 CA 证书都是一样的，CA 证书名称均为第一次成为 HA 集群主设备的序列号。




```bash
config system global
    set admin-server-cert Fortinet_GUI_Server
end
```

2. 点击“下载 HTTPS CA 证书”按钮，下载“Fortinet_GUI_Server”证书的 CA 证书，将 CA 证书保存在本地磁盘，该 CA 文件名称一般为“Fortinet_CA_SSL.cer”。



重要

- 当“Fortinet_GUI_Server”证书到期前 2 天时，FortiGate 会重新生成“Fortinet_GUI_Server”证书（客户端 PC 不需要重新下载并信任 Fortinet_CA_SSL）。
- 如果“Fortinet_CA_SSL”证书在 FortiGate 上过期并重新生成， Fortinet_GUI_Server 也会重新生成（客户端 PC 需要重新下载并信任新的 Fortinet_CA_SSL）。

导入和信任 CA 证书

Windows PC

1. 双击下载的 CA 证书文件，在安全警告框中点击“打开”按钮。



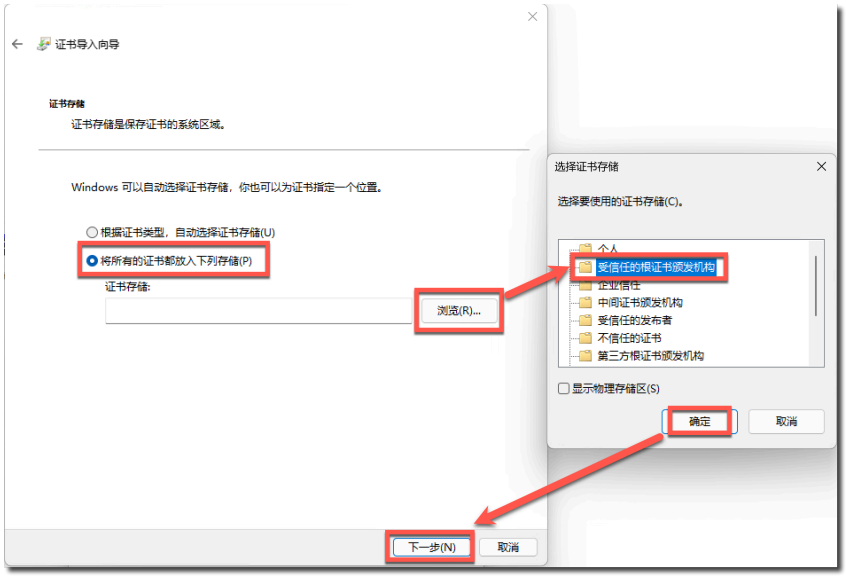
2. 在打开的证书界面点击“安装证书”按钮。



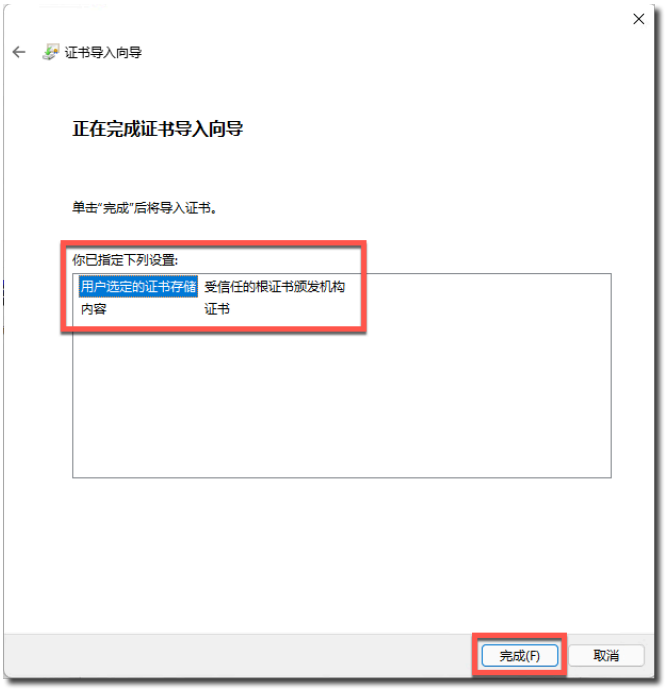
3. 在弹出的“证书导入向导”中选择存储位置（“当前用户仅对此用户生效”，“本地计算机”对所有用户生效），点击下一步按钮。



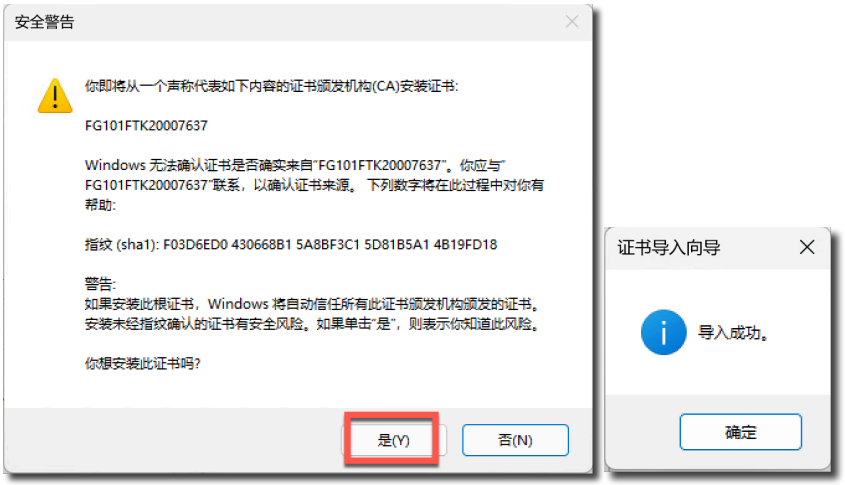
4. “证书存储”选择“将所有的证书都放入下列存储 (P)”，点击“浏览”按钮，选择存储到“受信任的根证书颁发机构”中，点击“确定”按钮，随后点击“证书导入向导”的“下一步按钮”。



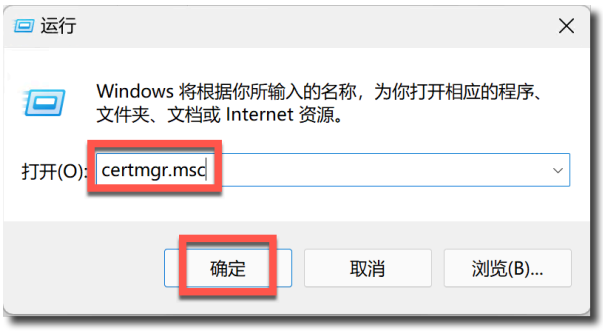
5. 点击“完成”按钮，完成 CA 证书的导入向导。



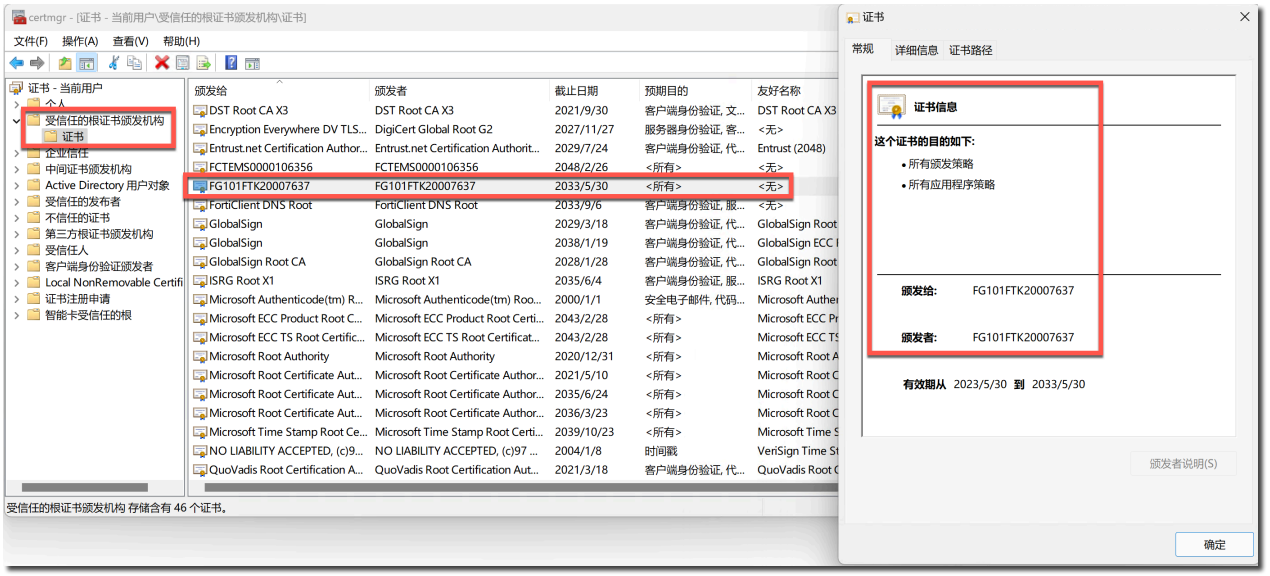
6. 随后可能会弹出如下的提示信息，点击“是 (Y)”按钮即可成功导入。



7. 使用 Win + R 快捷键打开运行窗口，输入“certmgr.msc”，点击确定按钮。

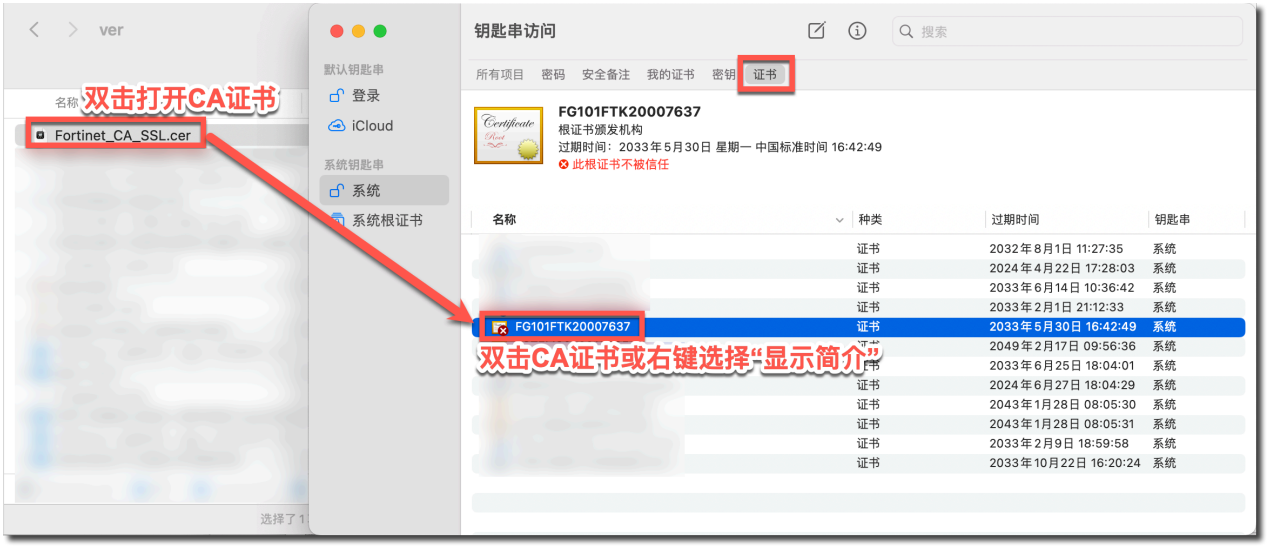


8. 在“受信任的根证书颁发机构”中可以看到以上导入的 CA 证书，证书的名称为设备的序列号。



macOS

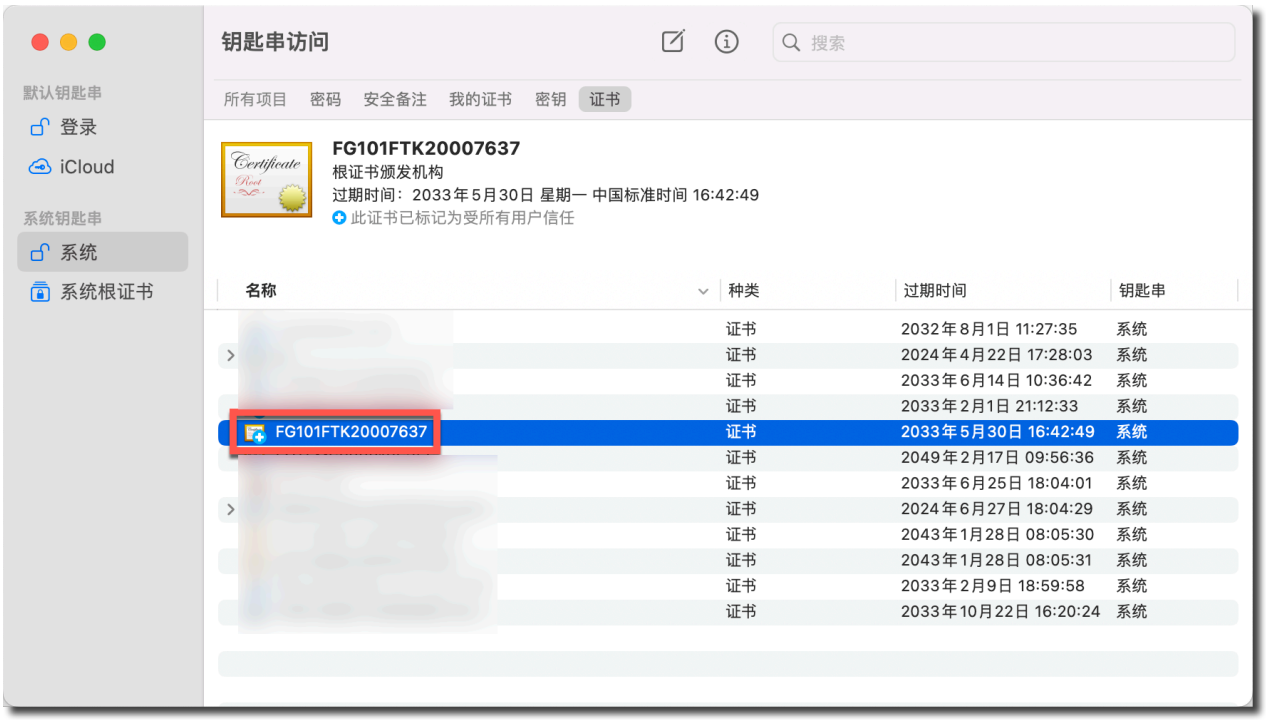
1. 双击下载的 CA 证书文件，会弹出“钥匙串访问”窗口（需要系统密码验证），进入“证书”选项页，双击 FortiGate 的 CA 证书（或右键点击该 CA 证书并选择“显示简介”）。



2. 打开该 CA 证书后，展开“信任”选项卡，修改“安全套接字层 (SSL)”的选项为“始终信任”，然后关闭该窗口（关闭时需要系统密码验证）。



3. 此时证书列表中该 CA 证书前的✖符号消失，变为✚。



iOS 与 iPad OS

请参考：https://support.apple.com/zh-cn/102390 ↗

Android

请参考：https://juejin.cn/s/安卓安装根证书 ↗

结果验证

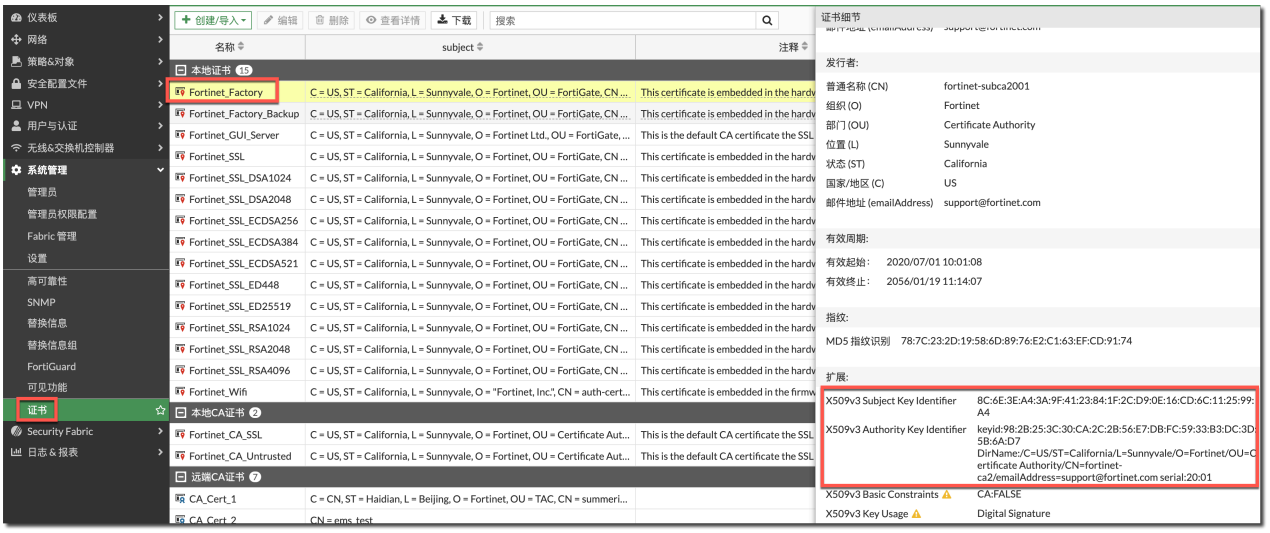
再次使用 Chrome/Firefox 访问 FortiGate 的 Web 管理页面（需要清除该站点的 Cookie），不会再产生证书告警，浏览器显示“连接是安全的”。



实现原理

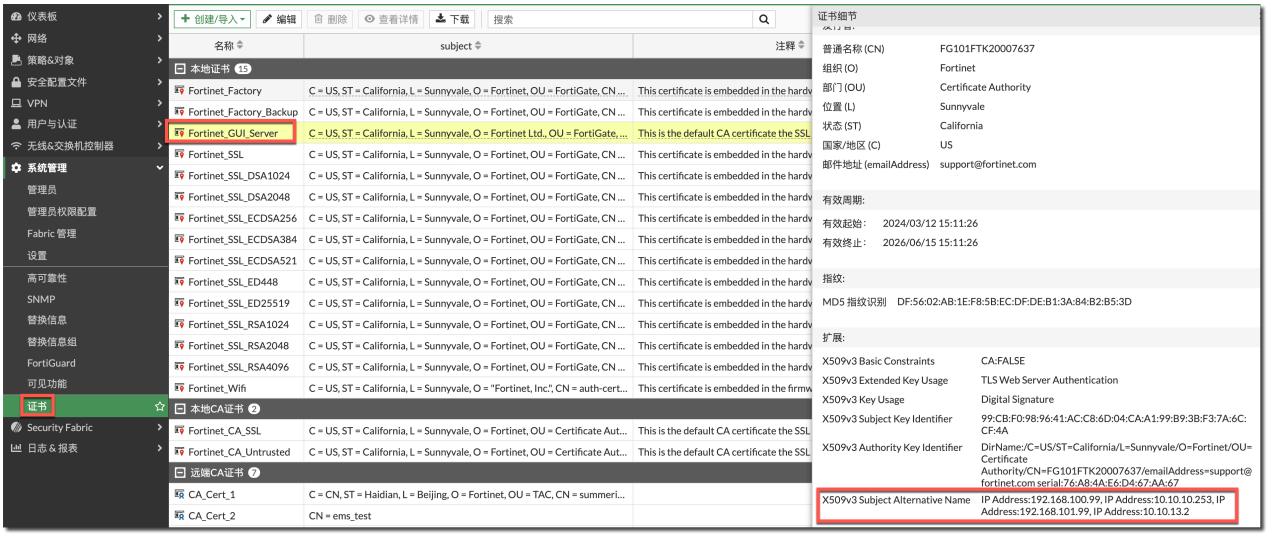
1. 现在的主流浏览器在验证网站证书时，不仅会检查网站证书的 CA 证书是否受信任，也会检查网站证书的 SAN（Subject Alternative Name）字段是否包含访问该网站时使用的 IP 或域名。

2. 使用其他非“Fortinet_GUI_Server”的内置证书（如 Fortinet_Factory）作为 Web 管理页面的证书时，即使用户在管理 PC 导入了对应的 CA 作为“受信任的根证书颁发机构”中的一员，但由于该证书没有 SAN 字段（如下图所示），访问 FortiGate Web 管理页面时，浏览器依然会提示告警信息。



3. 当使用“Fortinet_GUI_Server”作为 Web 管理页面的证书，FortiGate 会自动将设备的所有管理 IP 添加到“Fortinet_GUI_Server”证书的 SAN（Subject Alternative Name）属性中（如下图所示），当管理 PC 导入了对应的 CA 作为“受信任的根证书颁发机构”中的一员，浏览器检查 Web 管理页面中的 SAN 包含页面 IP，就认为该证书是受信任的。

> **重要**
>
> 只有开启了 HTTPS 访问的接口 IP 会被添加到“Fortinet_GUI_Server”证书的 SAN 属性中。当接口关闭 HTTPS 访问后，对应的 IP 也会从 SAN 字段中删除。



解决方法 2 - 导入第三方证书

1. 通过其他证书颁发机构（CA）为 FortiGate 颁发证书，颁发的证书中的 SAN（Subject Alternative Name）字段必须包含 FortiGate 的管理 IP 或域名（颁发和导入过程请参考：系统管理 → 系统设置 → 证书管理 章节）。

> **重要**
>
> 如果 FortiGate 的管理 IP 或域名发生变化，则需要重新为 FortiGate 颁发包含对应 SAN 的证书。

> **重要**
>
> 通过 CSR 的方式配置 SAN 时，注意 CSR 配置中请求的 SAN 的格式，如下举例：
>
> 域名：`DNS:www.fortinet.com.cn`
>
> IP 地址：`IP:10.10.12.1`

2. 在 FortiGate 的“系统管理 → 设置”页面的“管理端口设置”部分中，将“HTTPS 服务器证书”选择为上步导入的证书。

3. 管理 PC 信任该证书所属的 CA（参考上述方法，如果 CA 为公网公认的权威 CA，则不需要手动信任）。

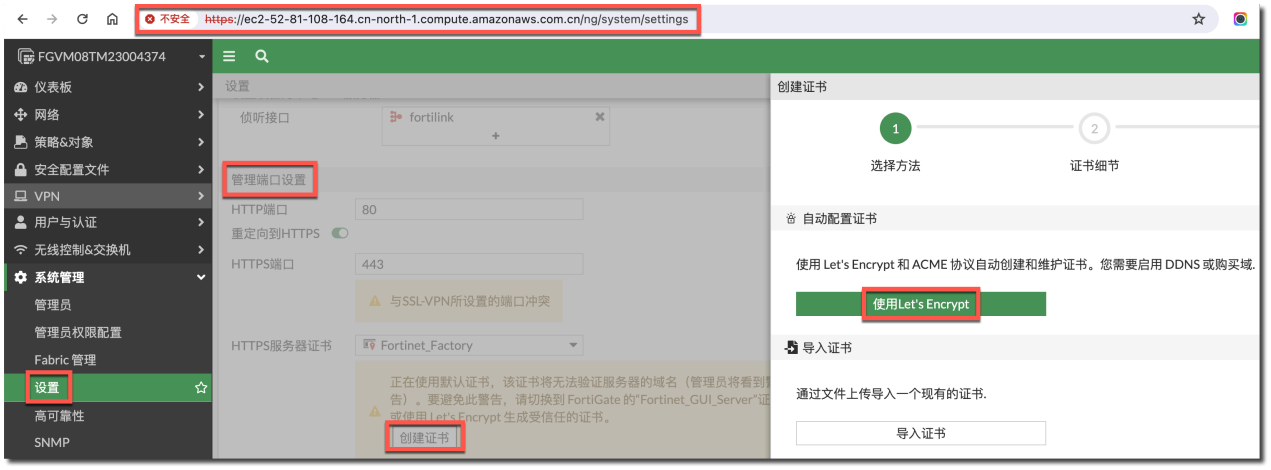
解决方法 3 - 使用 Let's Encrypt 颁发证书

> **重要**
>
> 使用该方法有以下要求：
>
> - FortiGate 必须有公网 IP 和对应的公网域名（或是在 Internet 设备上将公网 IP 映射到 FortiGate）。
> - FortiGate 的公网接口必须面向 Internet 开放 TCP 80 和 443 端口，以便 FortiGate 能够侦听 ACME 更新请求。 该接口不能配置任何 VIP，也不能在端口 80 或 443 上进行端口转发。
> - 颁发的证书中的 SAN（Subject Alternative Name）属性会自动填充 FortiGate 的 DNS 主机名，无法编辑，不能使用通配符，不能添加多个 SAN 字段。
> - FortiGate 必须能够访问 `acme-v02.api.letsencrypt.org` ， 这个域名是证书服务的提供商 Let's Encrypt 的地址（默认）。
> - 此方法创建的 FortiGate 证书不需要管理 PC 将 CA 证书导入本地的“受信任的根证书颁发机构”。
> - FortiGate 注册的域名需要在国内备案，否则可能出现注册过程被 ISP 或云运营商阻断的情况。

配置步骤

1. 如下图所示，FortiGate 的管理域名为 `ec2-52-81-108-164.cn-north-1.compute.amazonaws.com.cn` （注意该域名必须在 Internet 的 DNS 服务器上可以解析）。

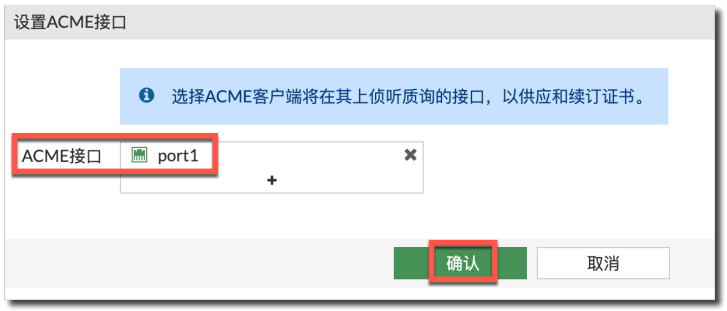
2. 现在使用浏览器访问时，会有“不安全”的标识。在 FortiGate 的“系统管理 → 设置”的“管理端口设置”中，点击“创建证书”按钮，在右侧弹出的“创建证书”窗口中，点击“使用 Let's Encrypt”，通过向导为 FortiGate 创建公有 CA 颁发的证书（该功能免费）。



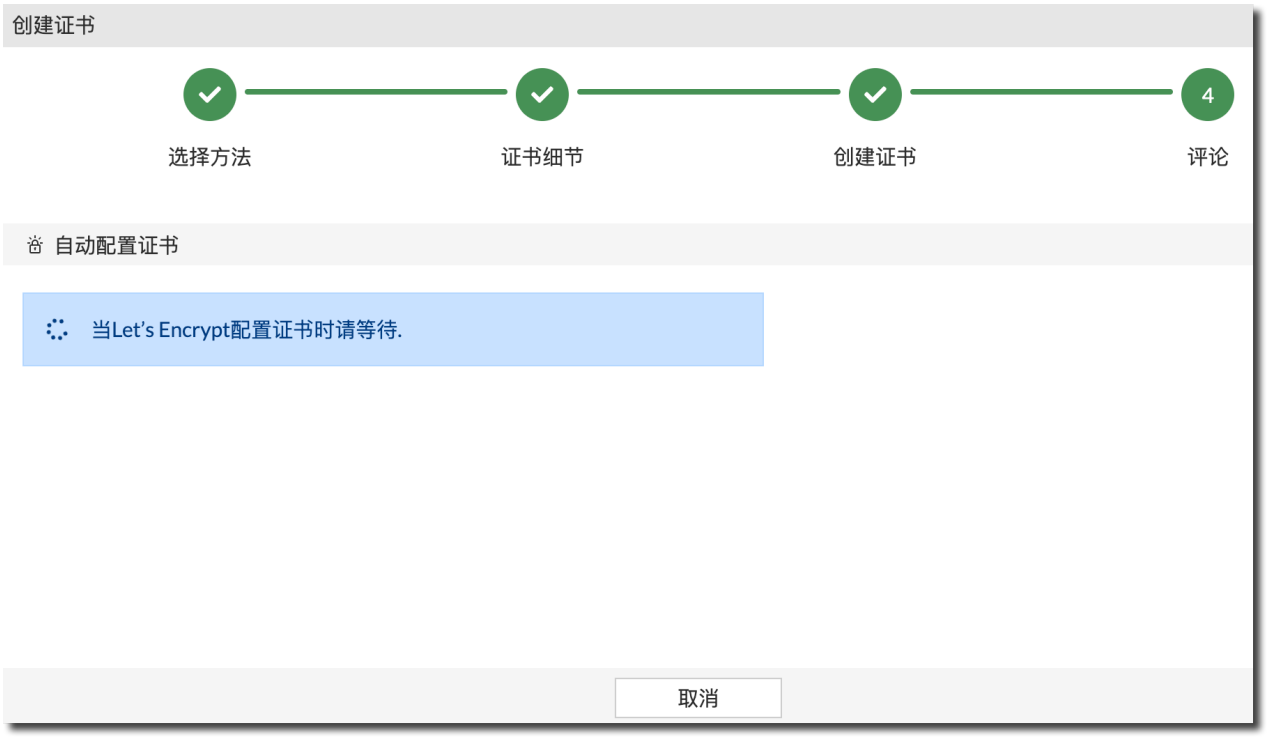
3. 在右侧弹出的“创建证书”窗口中，填写证书名称，“域”填写 FortiGate 的管理域名，填写 Email 信息，点击“创建”按钮。



4. 在弹出的“设置 ACME”窗口中，选择“ACME 接口”为 FortiGate 的公网接口，点击“确认”按钮。



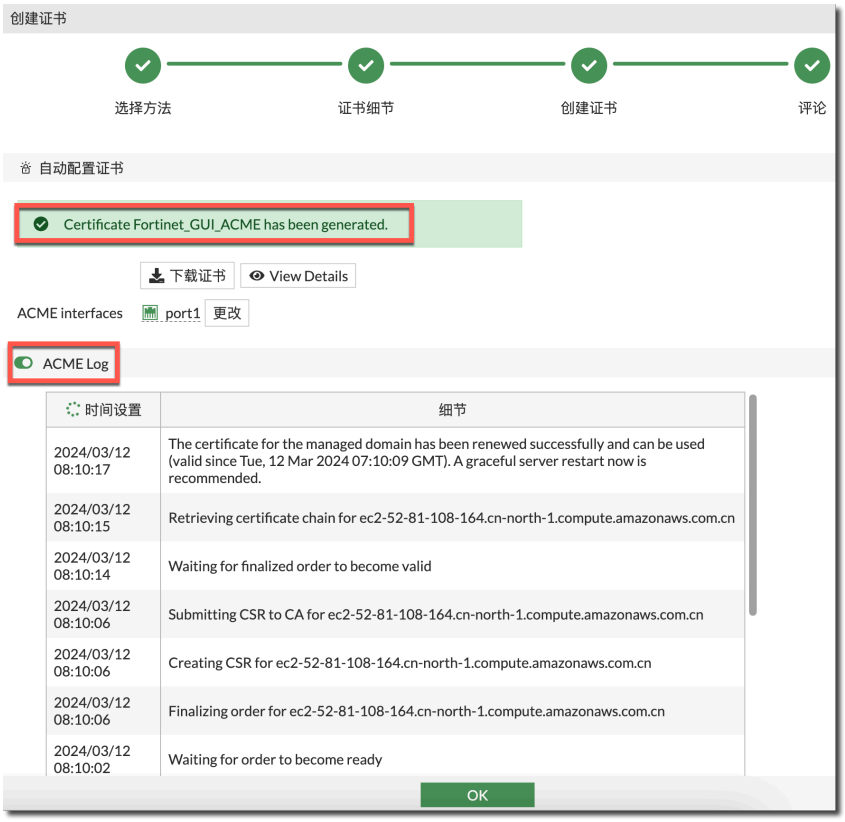
5. 耐心等待证书配置完成。



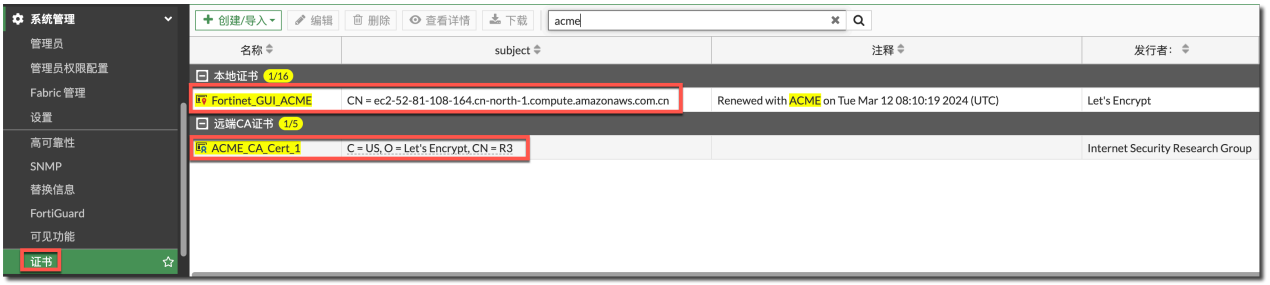
6. Let's Encrypt 证书创建完成。

重要

如果创建失败，可以点击“ACME Log”检查创建过程中的报错信息。

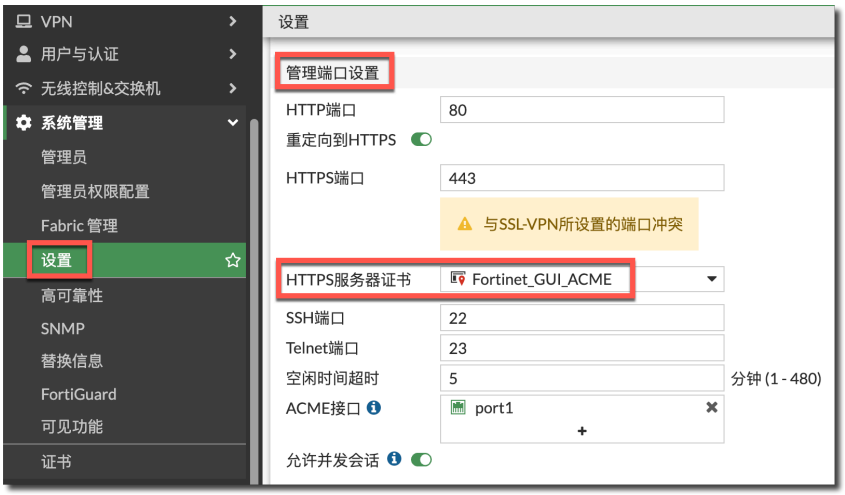


7. 证书的管理界面中除了生成了申请的“Fortinet_GUI_ACME”证书外，还有一张中间 CA 证书。




```bash
config vpn certificate ca
    edit "ACME_CA_Cert_1"
        set range global
    next
end

config vpn certificate local
    edit "Fortinet_GUI_ACME"
        set password ENC
jWTgiiU41fD+Mi94XyBXAY2ZY2wc+syGDVfIAEXKYRSwX2s30XJuR4sPybF+SNgKrWKDSp4tBxWunSNTMAjloZxtYZY1ieXOPPWQqkLtrZHuPZbRWxTBzl8h
Jm4ZeYxnh5xcN0hLjX4PXELjgIcQWv9fDE2B8YyJPzZDeL5FgSYA+tlM9670nfzlO9CbGIpQXqRGHw==
        set comments "Renewed with ACME on Tue Mar 12 08:10:19 2024 (UTC)"
        set range global
        set enroll-protocol acme2
        set acme-domain "ec2-52-81-108-164.cn-north-1.compute.amazonaws.com.cn"
        set acme-email "bbai@fortinet.com"
    next
end
```

8. 在“系统管理 → 设置”页面的“管理端口设置”部分中，将“HTTPS 服务器证书”选择为 Let's Encrypt 创建的证书“Fortinet_GUI_ACME”。



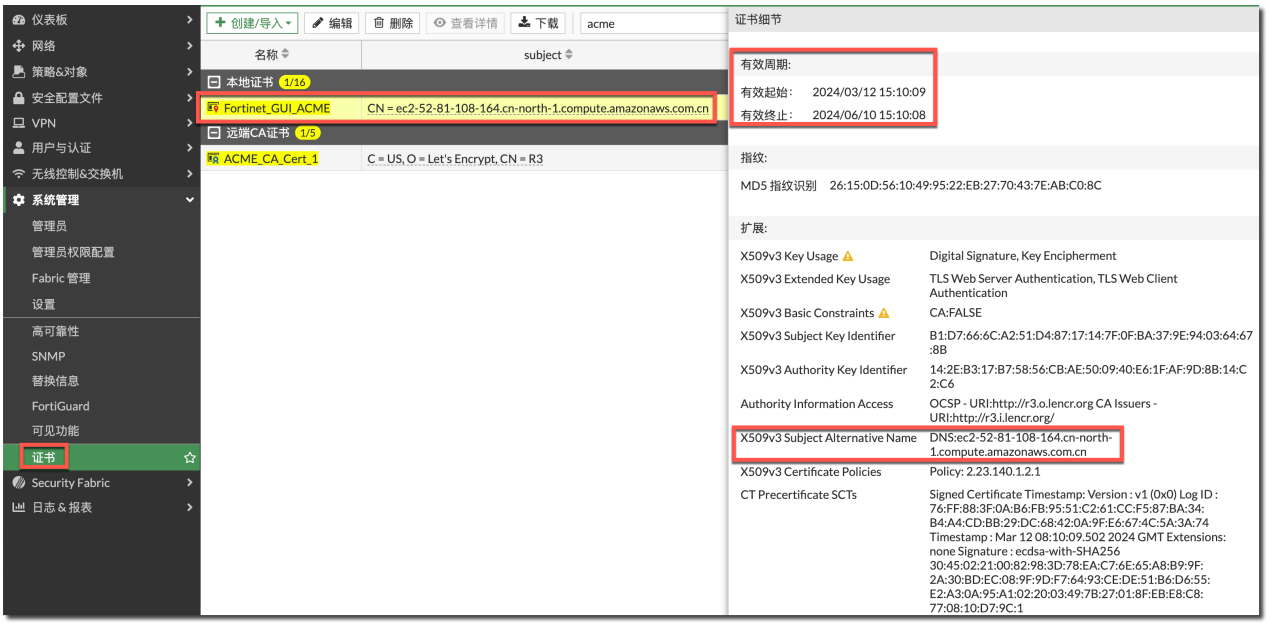

```bash
config system global
    set admin-server-cert Fortinet_GUI_ACME
end
```

结果验证

1. 再次使用 Chrome/Firefox 访问 FortiGate 的 Web 管理页面（需要清除该站点的 Cookie），不会再产生证书告警，浏览器显示“连接是安全的”。



2. 申请到的 Let's Encrypt 证书中的 SAN 字段包含 FortiGate 的管理域名，有效期为 90 天，到期时间 2 天前会自动续期。



相关调试 CLI

1. 查看 FortiGate ACME 状态。

> **重要**
>
> ACME hallenge type 是根据 FortiGate 的当前配置自动选择的。 当端口 443 是可用状态（不被 GUI、SSL VPN、Portal 等使用）时，它更喜欢使用 tls-alpn-01 挑战，否则使用 TCP 端口 80 上的 http-01 挑战。 如果 GUI 在端口 80 上运行，并且在所选的接口上设置了 HTTP 访问权限，那么 http-01 挑战将由节点代理转发，否则 Apache 将直接监听端口 80，只监听 ACME。

```bash
FortiGate # get system acme status
Listening on interfaces:        port1
ACME certificates registered:   1    //注册的证书个数//
ACME challenge type:            http-01    //监听80端口//
```

```
FortiGate # diagnose sys tcpsock | grep :80-
172.31.1.91:80→0.0.0.0:0→state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=59778 process=152/httpsd
```

2. 查看 ACME 申请记录日志。

```
FortiGate # diagnose sys acme status-full ec2-52-81-108-164.cn-north-1.compute.amazonaws.com.cn
```
bash
```
{
  "name": "ec2-52-81-108-164.cn-north-1.compute.amazonaws.com.cn",
  "finished": true,
  "notified": false,
  "last-run": "Tue, 12 Mar 2024 08:09:20 GMT",
  "valid-from": "Tue, 12 Mar 2024 07:10:09 GMT",
  "errors": 0,
  "last": {
    "status": 0,
    "detail": "The certificate for the managed domain has been renewed successfully and can be used (valid since Tue, 12 Mar 2024 07:10:09 GMT). A graceful server restart now is recommended.",
    "valid-from": "Tue, 12 Mar 2024 07:10:09 GMT"
  }
......
```

3. 重启与 Let's Encrypt 通信的 Apache 服务器。

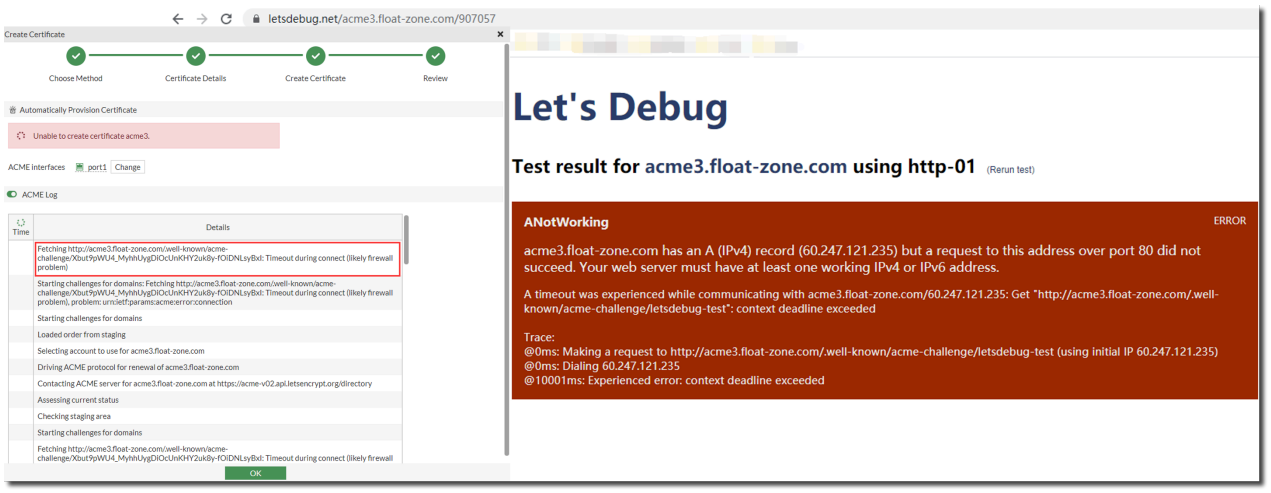
```
FortiGate # diagnose sys acme restart
```
bash

Let's Encrypt 申请域名失败举例

检查申请 Let's Encrypt 证书时页面打印的“ACME Log”。

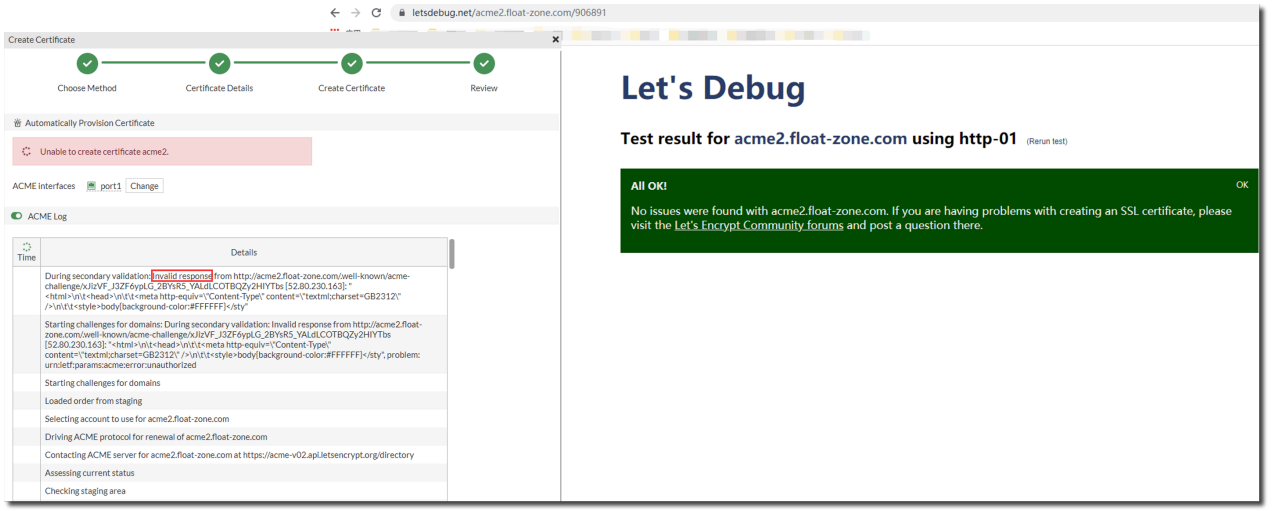
Let's Encrypt 无法主动访问到 FortiGate 的 TCP 80 或 443 端口

1. 例如使用位于内网的 FortiGate（也无无公网 IP 映射到 FortiGate），Let's Encrypt 访问不了其 TCP 80 或 443 端口，日志提示 Timeout during connect（likely firewall problem），通过抓包查看未有 Let's Encrypt acme-challenge 消息到达 FortiGate。
2. Let's Encrypt 有一个测试网站 https://letsdebug.net/ ↗ ，可以先对 FortiGate 进行检查。



FortiGate 的域名使用了未在国内备案的域名

1. 例如，DDNS 域名使用 acme2.float-zone.com ↗，由于 float-zone.com ↗ 未在国内备案，因此运营商可能会拦截对该域名的请求。FortiGate 的 ACME Log 显示 invalid response。
2. 但 https://letsdebug.net/检测是正常的。 ↗

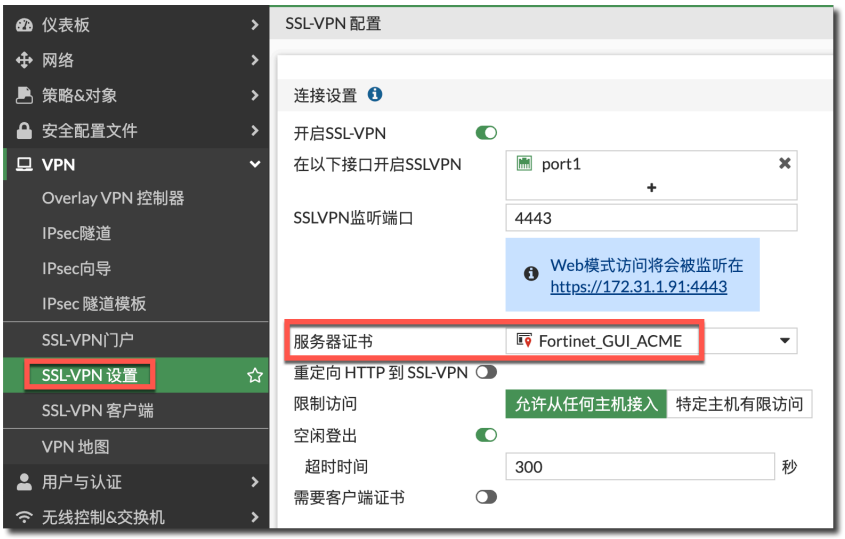


3. 本地使用游览器主动访问 FortiGate ACME 日志中的域名被拦截。因此 ACME 日志显示 invalid response。



SSL VPN 证书

参考方法 2 或方法 3 导入设备证书，并在客户端信任对应的 CA 证书后，在“VPN → SSL VPN → SSL - VPN 设置”中选择该证书即可。



```bash
config vpn ssl settings
    set servercert "Fortinet_GUI_ACME"
end
```

关闭 Web 提示信息

关闭升级提示

当 FortiGate 有新的固件可以升级时，每次登录页面都会弹出升级固件的提示信息，必须点击“下次再说”才能正常进入管理页面。



通过如下 CLI 可关闭“升级固件”的提示信息。

```bash
config sys global
    set gui-firmware-upgrade-warning disable
end
```

关闭注册提示

当 FortiGate 未注册时，每次登录页面都会弹出升级固件的提示信息，必须点击“下次再说”才能正常进入管理页面。



通过如下 CLI 可关闭“注册 FortiCare”的提示信息。

```
config sys global
    set gui-forticare-registration-setup-warning disable
end
```

关闭 FortiConverter 提示

从 FortiOS 7.4 开始，登录 FortiGate 时会提示使用 FortiConverter 进行配置转换。



通过如下 CLI 可关闭 FortiConverter 的提示信息。

```bash
diagnose sys forticonverter set-prompt-visibility hidden
```

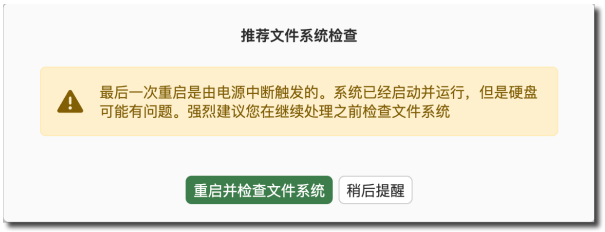
文件系统检查

功能介绍

设备启动或登录 CLI 时，如果出现如下告警信息：

```bash
WARNING: File System Check Recommended! An unsafe reboot may have caused an inconsistency in the disk drive.
It is strongly recommended that you check the file system consistency before proceeding.
Please run 'execute disk scan 259'
Note: The device will reboot and scan the disk during startup. This may take up to an hour.
```

GUI 中也会能弹出“推荐文件系统检查”提示：



该提示表示设备检测到上一次关机过程不安全。通常发生在未使用 `execute shutdown` 正常关机的情况下，例如突然断电、强制重启或异常掉电。

在此类场景中，系统无法确认磁盘文件系统是否被完整、正确地卸载，因此在启动过程中检测到潜在的不一致状态。

由于掉电或异常重启时，磁盘写入操作可能尚未完成，数据有可能仅被部分写入，从而导致文件系统状态不一致。为防止隐藏的磁盘或文件系统错误进一步扩大，系统通过该提示提醒管理员执行文件系统检查。

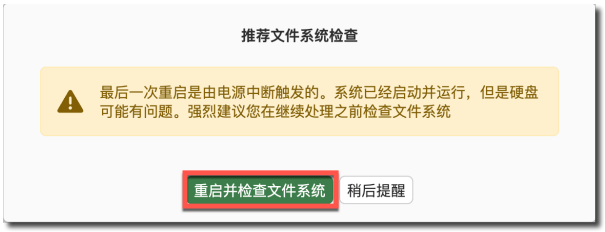
该行为属于 FortiGate 设备的磁盘与文件系统完整性保护机制，并非设备故障或系统异常。如果不进行文件系统检查，可能在后续运行中引发日志异常、存储错误或系统稳定性风险。

执行磁盘检查

> **注意**
>
> 整个检查过程通常耗时 5~15 分钟，大容量磁盘或部分型号可能更久（最多可达 1 小时），请在业务窗口期执行。

GUI

1. 在登陆 FortiGate GUI 后弹出的“推荐文件系统检查”提示窗口中点击“重启并检查文件系统”。



2. 推荐开启“自动文件系统检查”，点击“确认”按钮，将重启设备并执行磁盘检查。



3. 确认后设备将：

- 卸载相关分区
- 执行磁盘扫描
- 扫描完成后自动重启

4. 随后磁盘检查告警信息将不再出现。

CLI

1. 查看磁盘的分区编号，如下所示，磁盘 `partition ref` 为 259。

```bash
FortiGate # exe disk list

Disk Internal        ref: 258 447.1GiB    type: SSD [ATA Micron_5200_MTFD] dev: /dev/sda
  partition ref: 259 440.1GiB, 439.0GiB free  mounted: Y  label: LOGUSEDX2E9CDAE8 dev: /dev/sda1 start: 2048
```

2. 执行磁盘扫描，键入 `y` 后会自动重启并执行磁盘扫描。

```bash
FortiGate # execute disk scan 259
scan requested for:  device=/dev/sda1 259/Internal status=enable media-status=enable
This action requires the unit to reboot.
Do you want to continue? (y/n)
```

启用自动文件系统检查

当发生非正常关机后，下次启动自动执行文件系统检查，避免反复弹窗提醒（也可以在上文提到的“推荐文件系统检查”提示窗口中直接开启）：




```bash
config system global
    set autorun-log-fsck enable
end
```

附 FortiManager 的开启方法：

```bash
config fmupdate fwm-setting
    set auto-scan-fgt-disk enable
end
```

HA 场景注意事项

- 建议先在备机（Secondary）执行磁盘扫描。
- 备机扫描完成、HA 状态恢复后，进行一次主备切换，再对另一台设备执行相同操作。

固件与配置管理

固件版本管理

- 固件版本
- 产品生命周期
- 升级路径
- Web 界面升级
- FTP/TFTP/USB 方式升级
- 通过 FortiGuard 升级
- 固件切换
- Flash 刷机
- 软件签名增强
- G 系列修改安全级别
- 过期 License 升级限制
- 有效 License 自动升级
- 过期 License 强制升级

配置管理

- 配置备份及恢复
- 备份配置文件到 Flash
- 备份不包含密码的配置文件
- 通过 SCP 备份配置文件
- 恢复出厂
- 配置回退

固件版本管理

- 固件版本
- 产品生命周期
- 升级路径
- Web 界面升级
- FTP/TFTP/USB 方式升级
- 通过 FortiGuard 升级
- 固件切换
- Flash 刷机
- 软件签名增强
- G 系列修改安全级别
- 过期 License 升级限制
- 有效 License 自动升级
- 过期 License 强制升级

固件版本

版本推荐

相关信息

FortiGate 版本与 FortiManager、FortiAnalyzer 产品版本的适应关系可通过如下网站查询：https://docs.fortinet.com/compatibility-tool/fortimanager ↗

产品类型	产品名称	推荐版本	是否在支持周期内
入门级	FortiGate-30E	6.2.17	✖
	FortiWiFi-30E	6.2.17	✖
	FortiGate-30G	7.4.12	✅
	FortiWiFi-30G	7.4.12	✅
	FortiGateRugged-35D	6.2.17	✖
	FortiGate-40F/40F-3G4G	7.4.12	✅
	FortiWiFi-40F/40F-3G4G	7.4.12	✅
	FortiGate-50E/51E/52E	6.2.17	✖
	FortiWiFi-50E/51E	6.2.17	✖
	FortiGate-60E	7.4.12	✅
	FortiWifi-60E	7.4.12	✅
	FortiGate-60F/61F	7.4.12	✅
	FortiWiFi-60F/61F	7.4.12	✅
	FortiGateRugged-60F/60F-3G4G	7.4.12	✅
	FortiGate-70F/71F	7.4.12	✅
	FortiGateRugged-70F/70F-3G4G	7.4.12	✅
	FortiGate 70G/71G	7.4.12	✅
	FortiGate-80E/81E	7.4.12	✅
	FortiGate-80F/81F	7.4.12	✅
	FortiGate-90E/91E	7.4.12	✅
	FortiGate-90G/91G	7.4.12	✅
	FortiGate-94D	6.0.18	✖
	FortiGate-98D	6.0.18	✖
	FortiGate-100E/101E	7.2.13	✅
	FortiGate-100F/101F	7.4.12	✅
	FortiGate-120G/121G	7.4.12	✅
	FortiGate-140E	7.4.12	✅
	FortiGate-200E/201E	7.4.12	✅
	FortiGate-200F/201F	7.4.12	✅
	FortiGate 200G/201G	7.4.12	✅
	FortiGate-240D	6.0.18	✖
	FortiGate-280D	6.0.18	✖

中端	FortiGate-300E/301E	7.4.12	✅
	FortiGate-400E/401E	7.4.12	✅
	FortiGate-400E-BYPASS	7.4.12	✅
	FortiGate-400F/401F	7.4.12	✅
	FortiGate-500E/501E	7.4.12	✅
	FortiGate-600D	6.4.16	✖
	FortiGate-600E/601E	7.4.12	✅
	FortiGate-600F/601F	7.4.12	✅
	FortiGate-800D	7.4.12	✅
	FortiGate-900D	7.4.12	✅
	FortiGate-900G/901G	7.4.12	✅
高端	FortiGate-1000D	7.4.12	✅
	FortiGate-1100E/1101E	7.4.12	✅
	FortiGate-1000F/1001F	7.4.12	✅
	FortiGate-1200D	7.0.19	✖
	FortiGate-1500D	7.2.13	✅
	FortiGate-1500DT	7.2.13	✅
	FortiGate-1800F/1801F	7.4.12	✅
	FortiGate-2000E	7.4.12	✅
	FortiGate-2200E/2201E	7.4.12	✅
	FortiGate-2500E	7.4.12	✅
	FortiGate-2600F/2601F	7.4.12	✅
	FortiGate-3000D	7.4.12	✅
	FortiGate-3000F/3001F	7.4.12	✅
	FortiGate-3100D	7.4.12	✅
	FortiGate-3200D	7.4.12	✅
	FortiGate-3200F/3201F	7.4.12	✅
	FortiGate-3300E/3301E	7.4.12	✅
	FortiGate-3400E/3401E	7.4.12	✅
	FortiGate-3500F/3501F	7.4.12	✅
	FortiGate-3600E/3601E	7.4.12	✅
	FortiGate-3700D	7.4.12	✅
	FortiGate-3700F/3701F	7.4.12	✅
	FortiGate-3800D	7.2.13	✅

	FortiGate-3810D	6.4.16	✖
	FortiGate-3815D	6.4.16	✖
	FortiGate-3960E	7.4.12	✅
	FortiGate-3980E	7.4.12	✅
	FortiGate-4200F/4201F	7.4.12	✅
	FortiGate-4400F/4401F	7.4.12	✅
	FortiGate-4800F/4801F	7.4.12	✅
	FortiGate-5001D	6.4.16	✖
	FortiGate-5001E/5001E1	7.4.12	✅
	FortiGate-6K/7K	7.4.12	✅
虚机 VM	FortiGate-VM64	7.4.12	✅

固件命名规则

FGT_**101F**-**v7.0.13.M**-**build0566**-FORTINET.out

其中：

- `101F` 部分表示 FortiGate 型号，不同型号有不同的固件。
- `v7.0.13` 代表版本，这里使用 x.y.z（如 7.0.13）来详细说明：



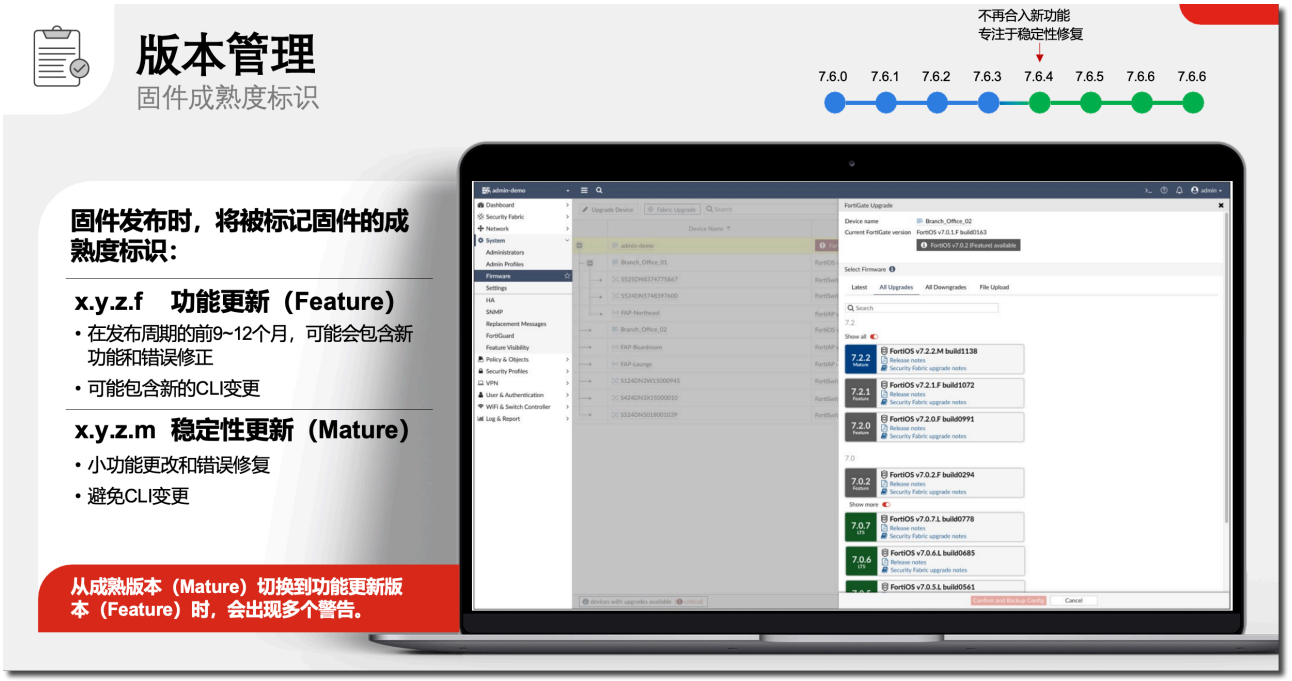
  - x.y 表示大版本号，我们已经发布的大版本有 v6.0、v6.2、v6.4、v7.0、v7.2、v7.4 等。
  - z 表示大版本下持续发布的小版本，对于成熟版本，主要是修复 bug，对于新功能版本，主要是推出新功能或者改进原有功能。
- `build0566` 表示内部代码 build 编号。
- 正在运行的设备版本号，可以在 CLI 下使用 get system status 观察到，如下所示。

```
FortiGate # get sys status | grep Version:
Version: FortiGate-101F v7.0.13,build0566,231024 (GA.M)
```

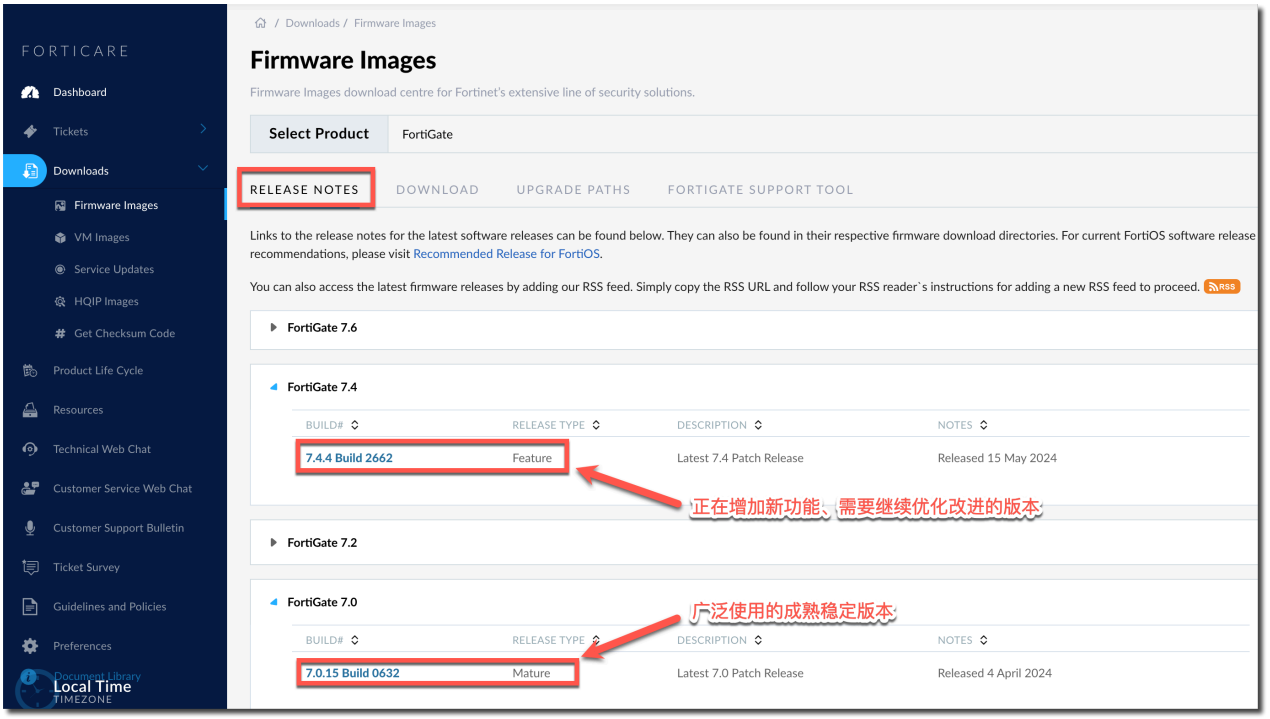
选择版本的原则

重要

- 带 M 标识的版本，B (x.y.z) 越大越稳定可靠。
- 在版本选择上，尽量使用版本数字后为 M（Mature）的版本，如“FGT_101F-v7.0.13.M-build0566-FORTINET.out”，该种类型的固件版本较为成熟稳定，尽量不要使用版本数字后为 F（Feature）的版本，如“FGT_101F-v7.4.1.F-build2463-FORTINET.out”。



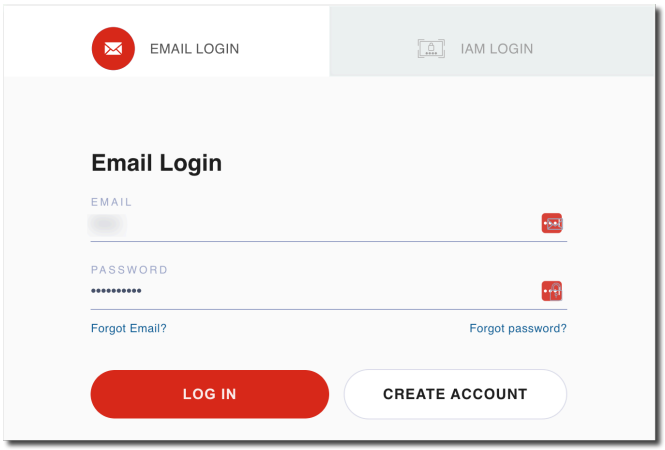
- Mature 版本是成熟的版本，是我们主要推荐的稳定版本。该种版本的更新几乎不会引入新功能，以修复 Bug 为主。对于承载敏感和重要业务的 FortiGate，应该部署 Mature 类型的版本。
- Feature 版本属于持续增加新功能、不停优化改进的版本。该种版本的更新以推出新功能为主，只有必须要用到该版本包含的新功能时才应该部署。
- 随着功能的增加改进，Feature 版本到一定阶段后会发展成为 Mature 成熟版本，您在 https://support.fortinet.com/support/#/downloads/firmware ↗ 下载版本时，一定要参考菜单 Support → Firmware Download → Release Notes 的说明。这部分的说明是经常被更新的。



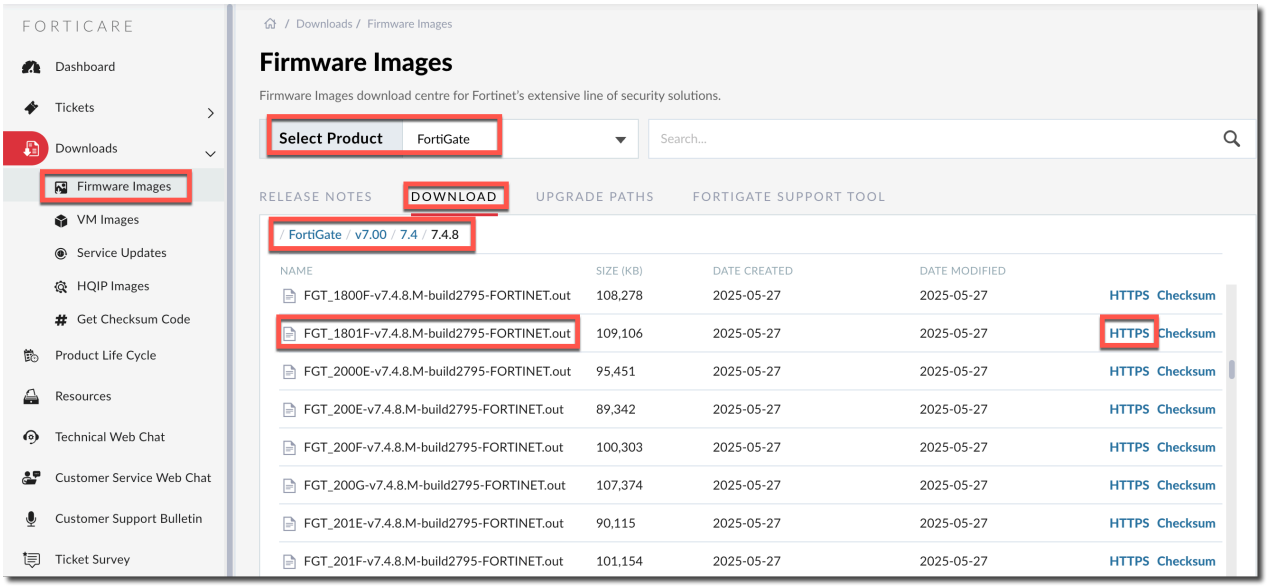
固件版本下载

FortiOS 固件可以通过 Support 网站，登录您的 FortiCloud 账户后进行下载。

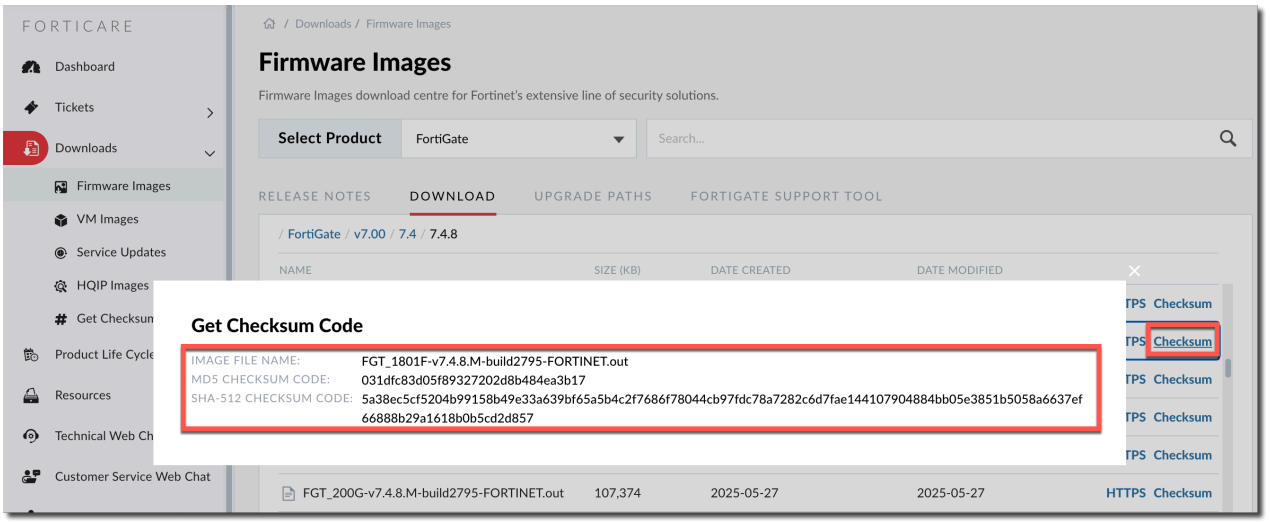
1. 使用浏览器直接打开 https://support.fortinet.com/support/#/downloads/firmware ↗ ， 登录您的 FortiCloud 账户（如您忘记了自己的 FortiCloud 账户或密码或账户被锁定，可联系 support_cn@fortinet.com ↗ 申请恢复）。



2. 登录后，选择对应的产品：FortiGate，选择 Download，进入要下载的版本目录（如下所示为 FortiOS 7.4.8），选择您的产品对应的版本（如下图所示为 FortiGate-1801F），点击右侧的 HTTPS 按钮进行下载。



3. 点击固件名称右侧的 Checksum 按钮，可以看到该版本文件的 Checksum 信息，请您下载文件后进行验证。



4. 附 Checksum 验证工具：Checksum_Tool ↗，也可通过在线 Checksum 校验（如 https://www.md5ma.com/md5-calculator ↗）。

产品生命周期

产品生命周期简介

Fortinet 会定期淘汰旧产品，以支持创新与新技术的发展。本政策适用于以下类别：

- 硬件产品及其相关服务（如 FortiCare、FortiGuard）
- 软件产品（包括 FortiOS、FortiClient 等）
- 设备型订阅服务（如 FortiGuard）
- 独立服务（Stand-alone Services），例如高级技术支持（Premium Technical Support）

生命周期阶段

Fortinet 的产品生命周期由多个关键节点组成，每个节点定义了产品销售与支持状态的变化：

阶段	英文缩写	含义与说明
产品正式发布	GA（General Availability）	产品向公众开放销售，可购买支持合同
终止声明	EOLN（End of Life Notification）	Fortinet 发布停止销售与支持的公告，通常包含替代产品建议 对于固件，EOL 日期通常在 GA 发布时即给出
停止订购	EOO（End of Order）	产品最后可下单日期，通常在公告前至少 90 天通知 对于独立服务，EOO 即 Last Extension Date
最后服务续约日期	LSED（Last Service Extension Date）	软件/硬件：合同续期不得超过 EOS LSED 为 EOS 前 12 个月 独立服务：LSED 为 EOL 公告后 90 天
终止工程支持	EOES（End of Engineering Support）	Fortinet 不再提供常规软件工程支持 仅对行业级关键问题与 PSIRT 漏洞进行必要修复（must-fix） 通常为 GA 后 36 个月
最终固件版本	FFR（Final Firmware Release）	指定硬件所支持的最后固件版本，维持 must-fix 状态直到 EOS
支持终止	EOS（End of Support）	生命周期最后阶段，Fortinet 不再销售、制造或提供支持服务

生命周期时长

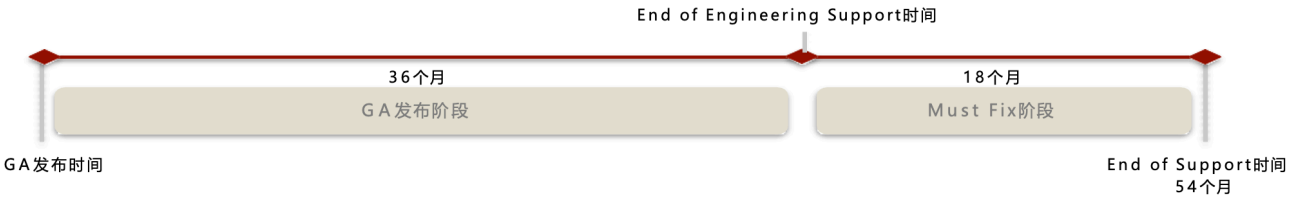
产品类型	生命周期主要阶段	持续时间
硬件产品	从 EOO 起至 EOS	60 个月（约 5 年）
软件产品	从 GA 起至 EOS	54 个月（约 4.5 年）
独立服务（Stand-alone Services）	到服务合同终止	按合同期限
工程支持终止（EOES）	从 GA 起	36 个月
服务续约截止（LSED）	EOS 前	12 个月（独立服务为 EOL 公告后 90 天）

相关信息

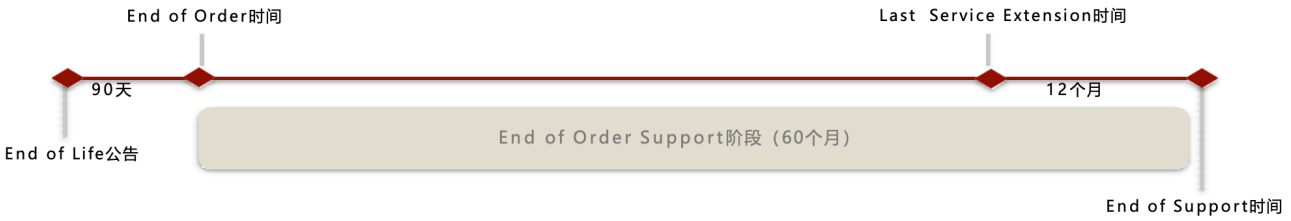
2026-03-31，Fortinet 发布公告 **CSB-260330-1**：FortiOS、FortiManager、FortiAnalyzer 的 7.4 与 7.6 版本支持生命周期延长 12 个月，EOES 与 EOS 日期整体顺延 1 年。

因此，涉及 7.4.x/7.6.x 版本的升级与维护计划（例如基于 EOES 的版本替换窗口）应按顺延后的日期执行；请以 https://support.fortinet.com/support/#/lifecycle ↗ 页面显示时间为准。

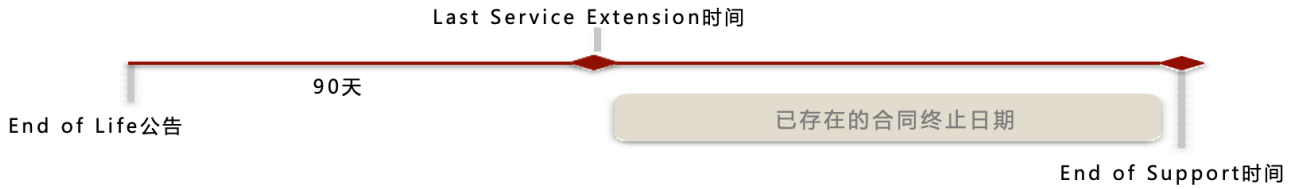
软件与固件



硬件与 FortiCare 服务



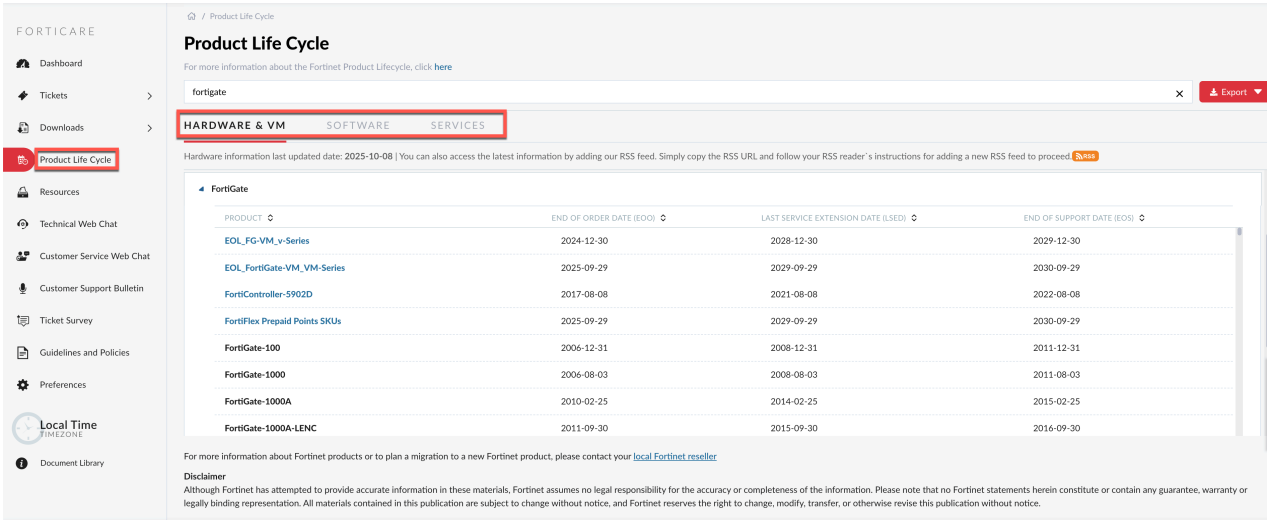
独立服务



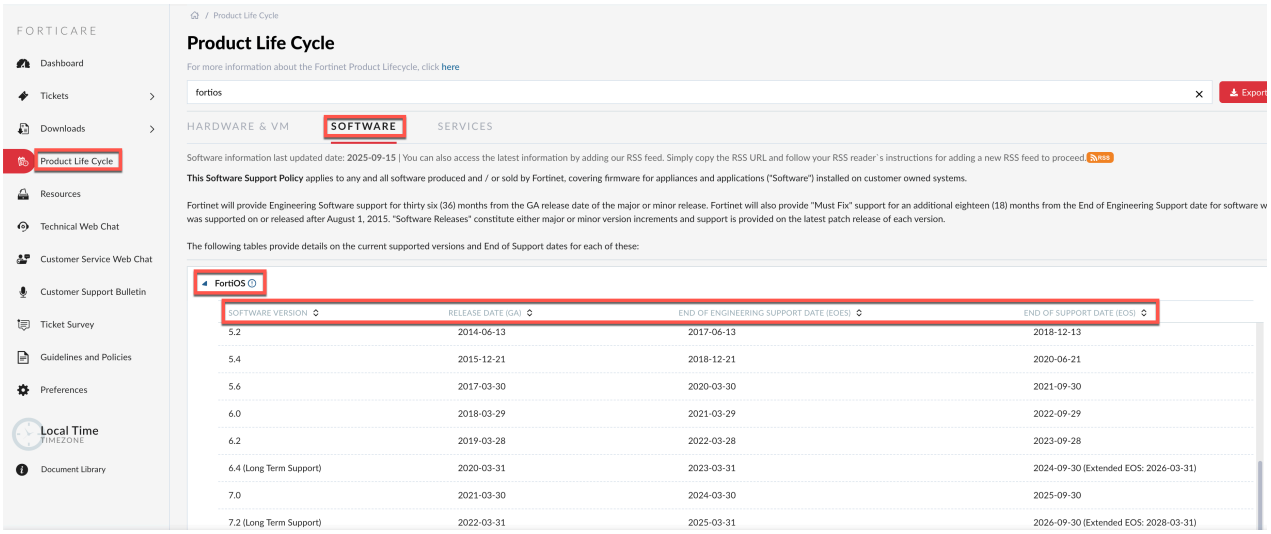
查询生命周期

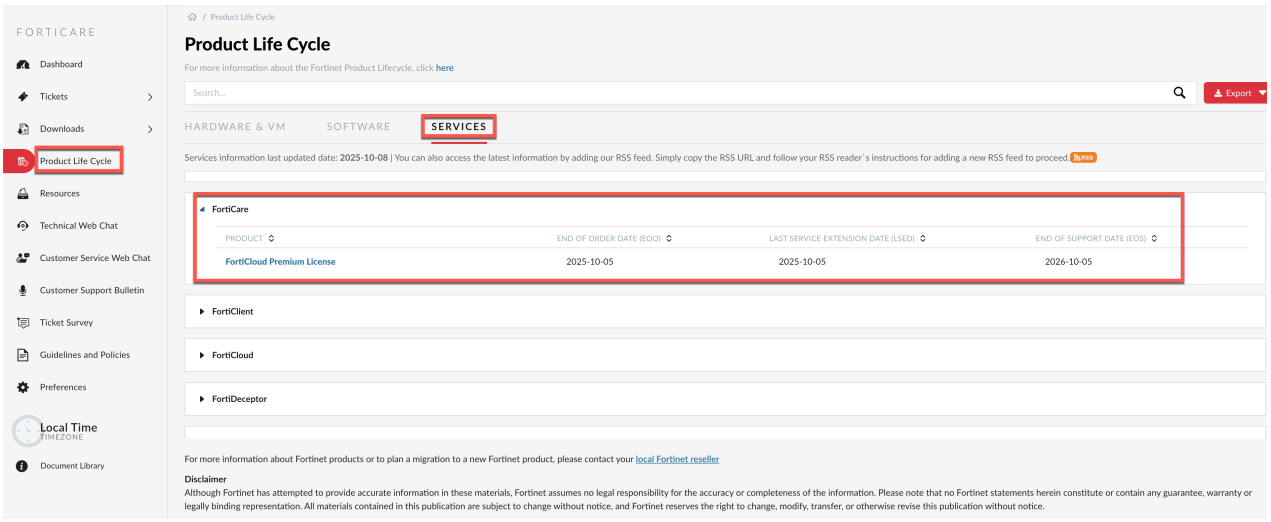
1. 使用设备注册的 FortiCloud 账户登录 https://support.fortinet.com/support/#/lifecycle ↗ 。

2. 登录后，可以选择不同的分类来查看硬件、软件、服务器的生命周期。



3. 如下所示可以查看 FortiOS 各个软件版本的生命周期（GA、EOES、EOS）、FortiCare 服务的生命周期（EOO、LSED、EOS）。





总结与建议

- 提前规划升级：建议在 EOES 阶段前（约 GA 后 3 年内）规划替换方案。
- 合同管理：关注 FortiCare/FortiGuard 到期，避免在 EOS 后尝试续约。
- 风险评估：EOS 后继续运行存在安全与合规风险，建议迁移到下一代产品。
- 关注公告：定期检查官方 EOL 通知，纳入资产与补丁管理流程。

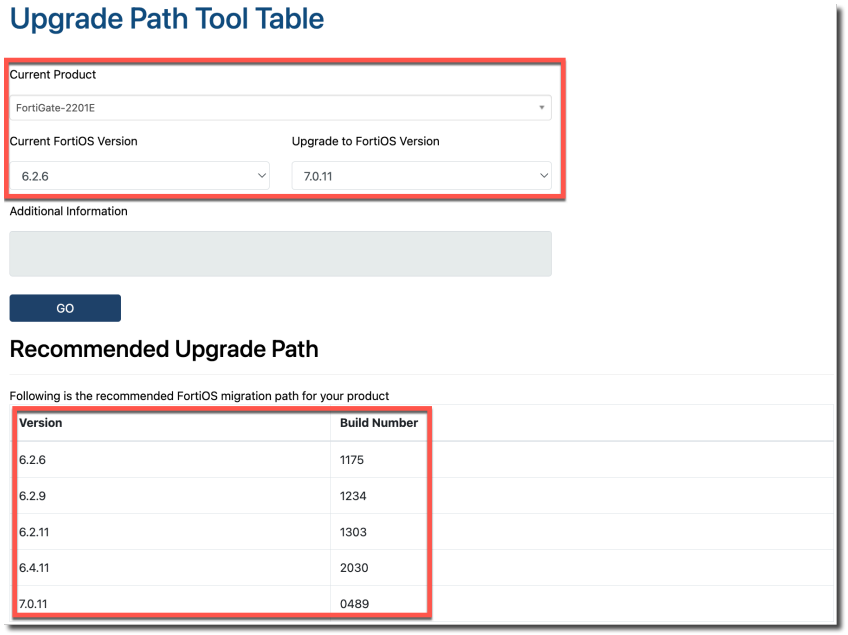
升级路径

由于不同固件版本之间数据库的差异，FortiGate 在升级固件版本时请 **务必遵循**升级路径的顺序进行升级，以保证数据库的正确性。

> **重要**
>
> 如果不按照升级路径而跨版本升级，可能会导致 FortiGate 数据库损坏，影响 FortiGate 的正常运行和功能使用。
>
> FortiGate 版本与 FortiManager、FortiAnalyzer 产品版本的适应关系可通过如下网站查询：https://docs.fortinet.com/compatibility-tool/fortimanager

查询步骤

1. 进入 https://docs.fortinet.com/upgrade-tool ，选择要升级的 FortiGate 型号，选择当前固件版本，和要升级到的目标版本，可以查询到中间的过渡升级版本。



2. 按照查询得出的升级路径，分别下载除当前版本的其他固件版本文件（参考系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 章节）。
3. 按照查询的升级路径依次升级固件版本，直至升级到目标版本。

Web 界面升级

需求

由于当前的软件版本老旧，想要通过 web 界面升级设备的系统软件版本。

升级要点

重要

- 升级前，务必做好配置备份，参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。
- 升级固件必须遵循升级路径，参考系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗，前往 https://docs.fortinet.com/upgrade-tool ↗ 查询升级路径，否则可能造成配置丢失。

1. FortiGate 防火墙的每款型号都有单独的版本文件，升级前务必确认下当前的设备型号。
2. 升级包的后缀名必须为.out，前缀任意。
3. 升级前建议准备配置线，以防在升级失败后能及时处理。
4. 升级过程中请不要切换到其他界面，更不能断电或重启设备，升级过程一般在 5 分钟以内完成。
5. 版本导入后设备将自动重启，重启才能生效。

升级步骤

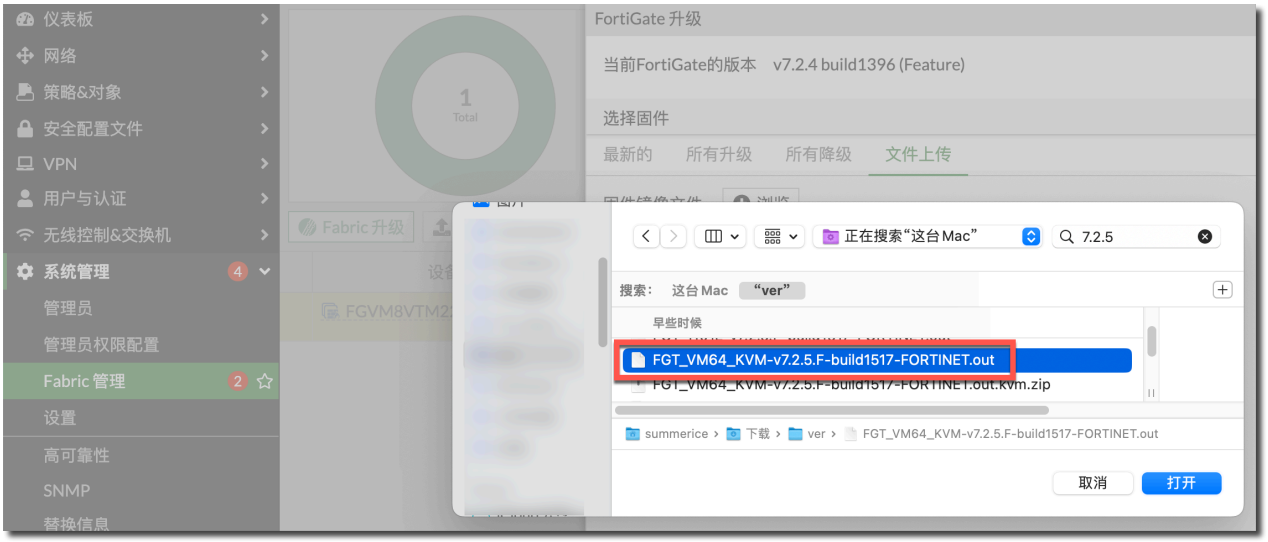
1. 登入 FortiGate 的 Web 界面，进入系统管理 → Fabric 管理（7.4 为 Firmware & Registration 页面），右键点击要升级的设备，选择升级按钮。



2. 在弹出的页面中，选择文件上传。



3. 点击浏览选择本地固件文件。



4. 点击“确认和备份配置”，系统会备份配置文件并自动重启升级。



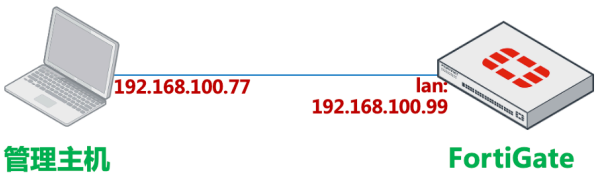
FTP/TFTP/USB 方式升级

需求

当不方便通过 Web 页面管理和升级 FortiGate 时，可以考虑通过 FTP/TFTP/USB 等 CLI 下的方式升级。使用该方法升级同样支持 HA 不间断升级↗。

> **重要**
>
> 升级前，务必做好配置备份，备份步骤请参考：系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。

拓扑图



配置要点

> **重要**
>
> - 升级前，务必做好配置备份，参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。
> - 升级固件必须遵循升级路径，参考系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗，前往 https://docs.fortinet.com/upgrade-tool ↗ 查询升级路径，否则可能造成配置丢失。

1. 连接网线，保证通讯正常。
2. 搭建好 FTP/TFTP 服务器，如果使用 USB 方式升级，需要将包含固件的 U 盘插入 FortiGate 的 USB 接口。
3. 开始升级。

升级方法

FTP/TFTP 方式

1. CLI 下通过 FTP/TFTP 的方式升级/导入固件。

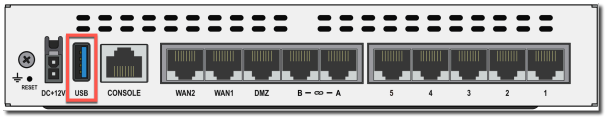
```bash
FTP:
exe restore image ftp <固件文件名称> <FTP服务器IP或域名>[:FTP端口号，标准端口21无需填写] <FTP用户名> <FTP密码>

TFTP:
exe restore image tftp <固件文件名称> <TFTP服务器IP或域名>[:TFTP端口号，标准端口21无需填写]
```

2. 举例如下。

```bash
FTP:
exe restore image ftp FGT_61F-v7.0.11.M-build0489-FORTINET.out 192.168.1.100 user1 Fortinet123#

TFTP:
exe restore image tftp FGT_61F-v7.0.11.M-build0489-FORTINET.out 192.168.1.100
```

USB 方式

1. 通过 USB 方式升级，需要将包含固件的 U 盘插入 FortiGate 的 USB 接口，如下图所示。



2. 升级命令。

```bash
exe restore image usb <固件文件名称>
```

3. 举例如下。

```bash
exe restore image usb FGT_61F-v7.0.11.M-build0489-FORTINET.out
```

通过 FortiGuard 升级

功能介绍

FortiGate 可以通过 `execute restore image management-station <version_name>` 命令在线下载版本并升级，该功能有以下特点：

- 相比较通过 Web 方式在线升级，可以看到固件的下载进度，更直观。
- 适用于暂时无法登录 FortiCloud 下载固件版本文件进行离线升级。
- FortiGate 需要可以通过 Internet 访问 FortiGuard。
- FortiGate 需要有 Firmware & General Updates License。
- 使用该方法升级同样支持 HA 不间断升级↗。

> **重要**
>
> - 升级前，务必做好配置备份，参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。
> - 升级固件必须遵循升级路径，参考系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗，前往 https://docs.fortinet.com/upgrade-tool ↗ 查询升级路径，否则可能造成配置丢失。

升级要点

> **重要**
>
> 升级固件必须遵循升级路径，参考系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗，前往 https://docs.fortinet.com/upgrade-tool ↗ 查询升级路径，否则可能造成配置丢失。

1. 升级前建议准备配置线，以防在升级失败后能及时处理。

2. 升级过程中请不要断电或重启设备，升级过程一般在 5 分钟以内完成。

3. 版本下载后设备将自动重启，重启才能生效。

升级步骤

举例：FortiGate-VM，7.0.17 GA 版本，需要升级至 7.4.7 版本，查询升级路径为 7.0.17 → 7.2.11 → 7.4.7。

1. 登入 FortiGate 的 CLI 界面，输入命令 `execute restore image management-station ?`（最后输入 `?`）。

```bash
FortiGate-VM # execute restore image management-station ?
Image-ID    Version
07006000FIMG0013706003  v7.06 MR6-GA-F P3 b3510 (upgrade)
07006000FIMG0013706002  v7.06 MR6-GA-F P2 b3462 (upgrade)
07006000FIMG0013706001  v7.06 MR6-GA-F P1 b3457 (upgrade)
07006000FIMG0013706000  v7.06 MR6-GA-F b3401 (upgrade)
07004000FIMG0013704007  v7.04 MR4-GA-M P7 b2731 (upgrade)
07004000FIMG0013704006  v7.04 MR4-GA-M P6 b2726 (upgrade)
07004000FIMG0013704005  v7.04 MR4-GA-M P5 b2702 (upgrade)
07004000FIMG0013704004  v7.04 MR4-GA-F P4 b2662 (upgrade)
07004000FIMG0013704003  v7.04 MR4-GA-F P3 b2573 (upgrade)
07004000FIMG0013704002  v7.04 MR4-GA-F P2 b2571 (upgrade)
07004000FIMG0013704001  v7.04 MR4-GA-F P1 b2463 (upgrade)
07004000FIMG0013704000  v7.04 MR4-GA-F b2360 (upgrade)
07002000FIMG0013702011  v7.02 MR2-GA-M P11 b1740 (upgrade)
07002000FIMG0013702010  v7.02 MR2-GA-M P10 b1706 (upgrade)
07002000FIMG0013702009  v7.02 MR2-GA-M P9 b1688 (upgrade)
07002000FIMG0013702008  v7.02 MR2-GA-M P8 b1639 (upgrade)
07002000FIMG0013702007  v7.02 MR2-GA-M P7 b1577 (upgrade)
07002000FIMG0013702006  v7.02 MR2-GA-F P6 b1575 (upgrade)
07002000FIMG0013702005  v7.02 MR2-GA-F P5 b1517 (upgrade)
07002000FIMG0013702004  v7.02 MR2-GA-F P4 b1396 (upgrade)
07002000FIMG0013702003  v7.02 MR2-GA-F P3 b1262 (upgrade)
07002000FIMG0013702002  v7.02 MR2-GA-F P2 b1255 (upgrade)
07000000FIMG0013700016  v7.00 GA-M P16 b0667 (downgrade)
07000000FIMG0013700015  v7.00 GA-M P15 b0632 (downgrade)
07000000FIMG0013700014  v7.00 GA-M P14 b0601 (downgrade)
07000000FIMG0013700013  v7.00 GA-M P13 b0566 (downgrade)
```

```
07000000FIMG0013700012  v7.00 GA-M P12 b0523 (downgrade)
07000000FIMG0013700011  v7.00 GA-M P11 b0489 (downgrade)
07000000FIMG0013700010  v7.00 GA-M P10 b0450 (downgrade)
07000000FIMG0013700009  v7.00 GA-M P9 b0444 (downgrade)
07000000FIMG0013700008  v7.00 GA-F P8 b0418 (downgrade)
07000000FIMG0013700007  v7.00 GA-F P7 b0367 (downgrade)
06004000FIMG0013704016  v6.04 MR4-GA-M P16 b2098 (downgrade)
......
```

2. 输出信息中可以看到该设备型号可以升级/降级的所有固件版本，其中“Version”列可以标识具体的版本号，如 `v7.04 MR4-GA-M P7 b2731` 表示 7.4.7，对应的“Image-ID”为 `07004000FIMG0013704007`，`v7.02 MR2-GA-M P11 b1740 (upgrade)` 表示 7.2.11，对应的“Image-ID”为 `07002000FIMG0013702011`。

> **重要**
>
> - `v7.02`：表示版本 7.2。
> - `MR2-GA-F`：F 表示功能版本，M 表示成熟版本。
> - `P11`：表示 Patch 11（7.2.11）。
> - `b1740`：表示 build 号。

```
Image-ID    Version
07004000FIMG0013704007  v7.04 MR4-GA-M P7 b2731 (upgrade)
07002000FIMG0013702011  v7.02 MR2-GA-M P11 b1740 (upgrade)
```

3. 首先需要升级到 7.2.11 版本，执行如下 CLI 命令，`07002000FIMG0013702011` 为 7.2.11 版本的“Image-ID”，如果 FortiGate 可以正常通过 Internet 访问到 FortiGuard，会下载对应的版本并显示下载进度，下载完成后，键入 `y` 确认，随后设备将重启并升级至目标系统版本 7.2.11。

```
FortiGate-VM # execute restore image management-station 07002000FIMG0013702011
Please wait...

Getting image 07002000FIMG0013702011 from Management station...
################################################################################################################
This operation will replace the current firmware version!
Do you want to continue? (y/n)y

Verifying the signature of the firmware image.
Warning: Upgrading to an image with Mature maturity notation.

Checking new firmware integrity ... pass

Please wait for system to restart.
```

4. 使用相同的方法升级到 7.4.7 版本。

```
FortiGate-VM # execute restore image management-station 07004000FIMG0013704007
Please wait...

Getting image 07004000FIMG0013704007 from Management station...
################################################################################################################
This operation will replace the current firmware version!
Do you want to continue? (y/n)y

Verifying the signature of the firmware image.
Image verification OK!
Warning: Upgrading to an image with Mature maturity notation.

Checking new firmware integrity ... pass

Please wait for system to restart.
```

5. 如果升级的目标版本是 F（Feature）类型，会有额外的警告信息，并需要多次确认。

```
Verifying the signature of the firmware image.
Warning: Upgrading to an image with Feature maturity notation.
Image file uploaded is marked as a Feature image, are you sure you want to upgrade?
Do you want to continue? (y/n)y

Please confirm again. Are you sure you want to upgrade using uploaded file?
Do you want to continue? (y/n)
```

6. 如果是降级或不按照升级路径升级（不推荐），会有额外的警告信息，并需要多次确认。

```bash
Verifying the signature of the firmware image.
Warning: Installing image v7.2.11 from v7.4.7 is not a recommended upgrade path. Continuing with the upgrade may result in loss of configuration. Do you want to proceed?
Do you want to continue? (y/n)y

Warning: Downgrading to an image with Mature maturity notation.
Image file uploaded is marked as a Mature image, are you sure you want to downgrade?
Do you want to continue? (y/n)y

Please confirm again. Are you sure you want to downgrade using uploaded file?
Do you want to continue? (y/n)y

This operation will downgrade the current firmware version.
Do you want to continue? (y/n)
```

固件切换

功能介绍

1. 在 FortiGate 物理设备的 Flash 闪存中，可以保存两个固件版本及其对应的配置文件，一个为默认运行的固件，另一个为备份固件。
2. 通过如下命令可以看到当前 Flash 闪存中存在的两个固件版本分区。如下输出中，Active 为 Yes 的分区（2）是当前正在运行的固件版本分区，另外一个是升级到此版本前的上个固件版本分区。

```
FortiGate # diagnose sys flash list
Partition  Image                                  TotalSize(KB)  Used(KB)  Use%  Active
1          FG101F-7.00-FW-build0450-230221                253920    111052   44%  No
2          FG101F-7.00-FW-build0489-230314                253920    110816   44%  Yes
3          ETDB-91.03066                                 3021708    270904    9%  No
Image build at Mar 15 2023 00:12:07 for b0489
```

3. 如果在升级后遇到问题需要回退（降级），我们可以将当前的运行版本切换回升级前的上个版本。

> **注意**
>
> - 配置文件也会恢复到升级前上个版本运行的配置文件，如果您在升级后又配置了新的内容，请注意切换后配置上的差异。
> - 请尽量使用此方法（Flash 固件切换）进行无损降级。不要直接在较高版本中上传较低版本的固件进行降级，即使是空配置的状态，也可能导致 FortiOS 数据库损坏，进而导致降级后部分功能无法正常使用，甚至是无法正常启动！

切换步骤

1. 要恢复到 Flash 分区中的另一个固件，请通过 CLI 使用以下命令（该命令只能由具有 super_admin 权限的管理员执行），并选择下次重新启动时应使用的固件。

```
FortiGate # execute set-next-reboot {primary | secondary}
```

> **重要**
>
> Primary 和 Secondary 分别指分区 1 和分区 2，可以忽略分区 3。

2. 在此例中，我们正在运行的是分区 2，想要切换的是分区 1，所以要执行以下命令。

```
FortiGate # execute set-next-reboot primary
Default image is changed to image# 1.
```

3. 重启 FortiGate，FortiGate 将会加载分区 1 中的固件版本及其对应的配置文件。

```
FortiGate # execute reboot
```

4. 重启后，再次查看 Flash 分区信息，Active 的分区已更改为分区 1。

```
FortiGate # diagnose sys flash list
Partition  Image                                  TotalSize(KB)  Used(KB)  Use%  Active
1          FG101F-7.00-FW-build0450-230221                253920    111052   44%  Yes
2          FG101F-7.00-FW-build0489-230314                253920    110816   44%  No
3          ETDB-91.03066                                 3021708    270904    9%  No
Image build at Feb 13 2023 00:11:57 for b0450
```

注意事项

1. 每次执行升级后，Flash 分区会将较旧的固件版本分区替换为新升级的固件版本。
2. 在 HA 环境下，该命令不会在集群成员中同步，必须在每个 HA 成员中执行。

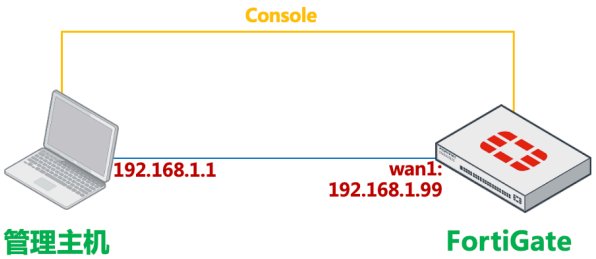
Flash 刷机

需求

防火墙无法正常启动时，可以考虑格式化 Flash 闪存盘，传入新的固件，防火墙会使用新的固件与配置文件启动。

> **重要**
>
> 通过此方式将固件刷入 Flash 卡后，原有固件与配置文件会被清空。
>
> 刷机前，务必做好配置备份，参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。
>
> 如果 FortiGate 设备运行的 BIOS 版本为 5000100 或 6000100（或更新版本），用户将无法降级到 6.0、6.2 或 6.4.13、7.0.12、7.2.5、7.4.0 以下的 FortiOS 版本，如果需要降级到这些版本，请参考这里 ↗修改 BIOS 安全级别。

拓扑图



配置要点

1. 准备好工具，连接好 Console 线。
2. 连接网线，保证通讯正常。
3. 搭建好 TFTP 服务器。
4. 开始升级。

配置步骤

1. 准备好工具，准备好 Console 线，网线，升级文件，TFTP 工具，若 PC 没有 com 口，需要购买转 USB 的线缆，并且装好驱动。

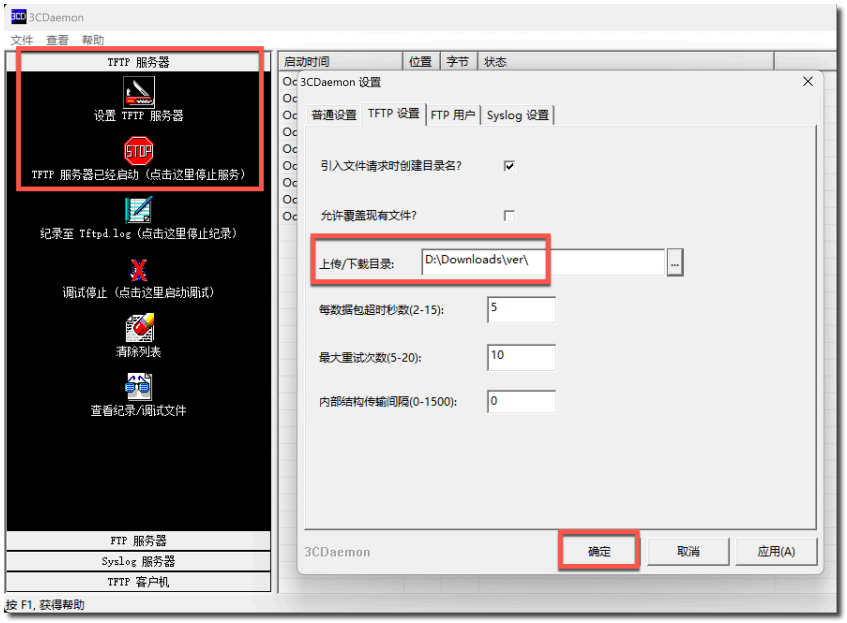
2. 连接网线，保证通讯正常。

3. 搭建好 TFTP 服务器，可下载如下附件里的 TFTP 服务器软件（也可使用其他 TFTP Server 软件）。

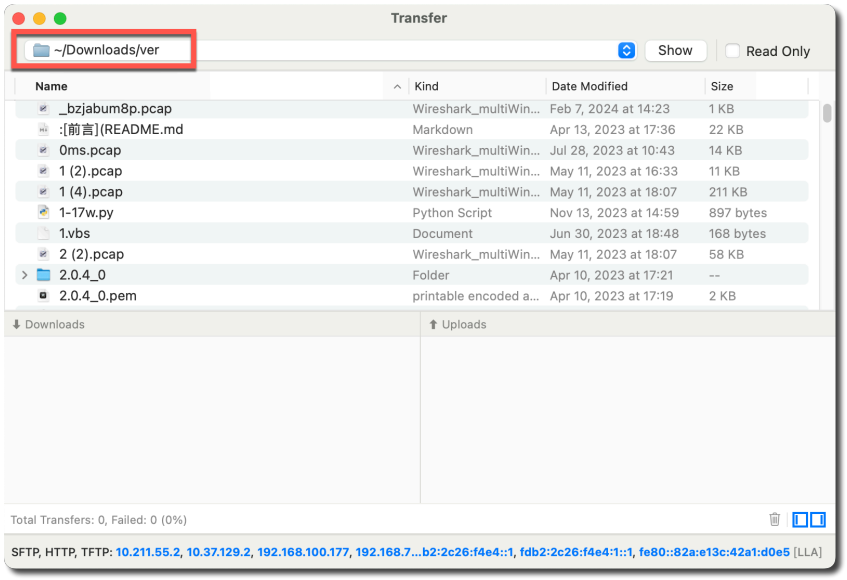
   - Windows：3CDaemon ↗
   - macOS：Transfer ↗

4. 启动 TFTP 服务器软件（确保 TFTP 服务器处于开启状态），修改 TFTP 服务器的文件路径到固件包所在路径，如下所示。

   - Windows（如下图路径为 `D:\Downloads\ver\`）：



- macOS（如下图路径为 `~/Downloads/ver`）：



5. 开始升级，重新启动设备。
6. 重启开始时将出现 "Press any key to display configuration menu..."，此时按任意键进入 BIOS 菜单（有些型号的 BIOS 可能需要敲入 Ctrl + B 进入，请根据提示信息操作）。

```bash
Please wait for OS to boot, or press any key to display configuration menu....

[C]: Configure TFTP parameters.
[R]: Review TFTP parameters.
[T]: Initiate TFTP firmware transfer.
[F]: Format boot device.
[I]: System information.
[B]: Boot with backup firmware and set as default.
[Q]: Quit menu and continue to boot.
[H]: Display this list of options.
```

7. 键入 `F`，格式化 Flash 卡，提示是否确认格式化，键入 `yes` 后回车。

> **重要**
>
> 键入 F，将 flash 卡格式化，原有配置文件及 OS 将会被删除掉。

```
Enter C,R,T,F,I,B,Q,or H:f
It will erase data in boot device. Continue? [yes/no]:yes
Formatting.......... Done.
```
bash

8. 格式化完毕后，键入 `C`，配置 TFTP 服务器，看到菜单如下。

```
[P]: Set firmware download port.
[D]: Set DHCP mode.
[I]: Set local IP address.
[S]: Set local subnet mask.
[G]: Set local gateway.
[V]: Set local VLAN ID.
[T]: Set remote TFTP server IP address.
[F]: Set firmware file name.
[E]: Reset TFTP parameters to factory defaults.
[R]: Review TFTP parameters.
[N]: Diagnose networking(ping).
[Q]: Quit this menu.
[H]: Display this list of options.
```
bash

9. 键入 `P` 可以选择 FortiGate 上传固件使用的接口，这里我们使用 WAN1 接口为例，键入 WAN1 的序号 8，然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:8
   [0]:  port 1
   [1]:  port 2
   [2]:  port 3
   [3]:  port 4
   [4]:  port 5
   [5]:  A
   [6]:  B
   [7]:  DMZ
   [8]:  WAN1
   [9]:  WAN2
```

0. 键入 `I`，输入防火墙本地 IP（中括号 [] 内的 IP 为上次进入 BIOS 配置的本地 IP），然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:i

Enter local IP address [192.168.91.180]: 192.168.90.176
.done
```
bash

1. 键入 `S`，输入防火墙本地 IP 的掩码（中括号 [] 内的掩码为上次进入 BIOS 配置的本地 IP 掩码），然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:s

Enter local subnet mask [255.255.255.0]: 255.255.255.0
.done
```
bash

2. 如果 FortiGate 与 TFTP 服务器不在同一网段，键入 `G`，输入 FortiGate 的默认网关，否则跳过此步（中括号 `[]` 内的网关为上次进入 BIOS 配置的网关），然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:g

Enter local gateway IP address [192.168.91.254]: 192.168.90.254
.done
```
bash

3. 键入 `T`，输入 TFTP 服务器的 IP 地址（中括号 `[]` 内的 IP 为上次进入 BIOS 配置的 TFTP 服务器 IP），然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:t

Enter remote TFTP server IP address [192.168.118.26]: 192.168.90.178
.done
```
bash

4. 键入 `F`，输入传输的版本镜像名称（中括号 `[]` 内的文件名称为上次进入 BIOS 配置的镜像文件名称），然后键入回车键。

```
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:f

Enter firmware file name [FGT_61F-v6-build0866-FORTINET.out]: FGT_61F-v7.2.8.M-build1639-FORTINET.out
.done
```
bash

5. 键入 `R`，检查 TFTP 配置是否正确，检查无误后，键入 `Q` 返回上一级菜单。

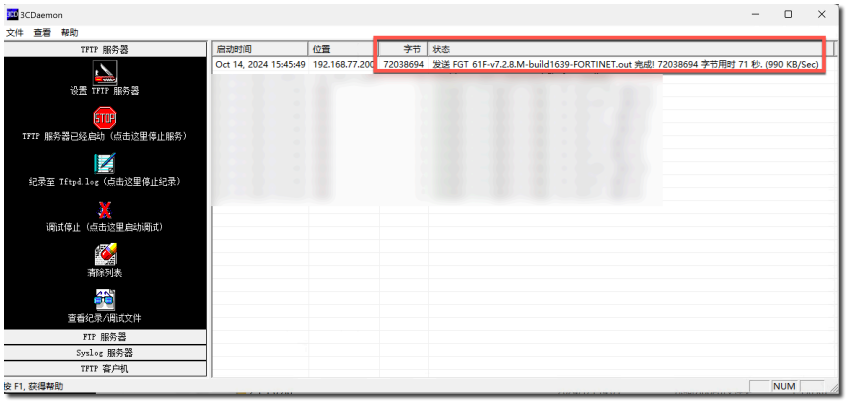
```bash
Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:r
Image download port:    WAN1
DHCP status:            Disabled
Local VLAN ID:          <NULL>
Local IP address:       192.168.90.176
Local subnet mask:      255.255.255.0
Local gateway:          192.168.90.254
TFTP server IP address: 192.168.90.178
Firmware file name:     FGT_61F-v7.2.8.M-build1639-FORTINET.out
```

6. 键入 `T`，开始从 TFTP 服务器传输版本镜像文件，传输完成后，键入 `D` 将该镜像设置为默认启动的主系统版本（也可键入 `B` 设置为备份系统的镜像）。

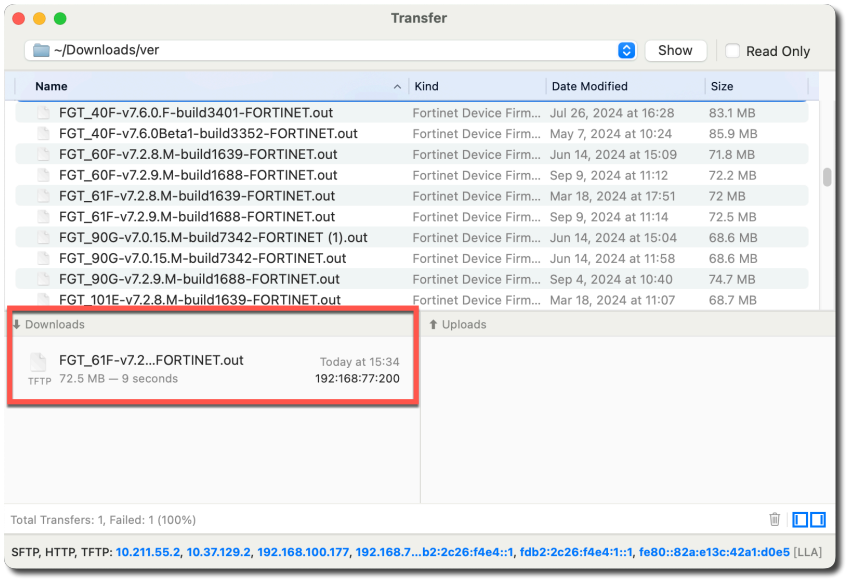
```bash
Enter C,R,T,F,I,B,Q,or H:t

Please connect TFTP server to Ethernet port 'WAN1'.
MAC: 90:6c:ac:f9:18:e8
Connect to tftp server 192.168.90.178 ...

##########################################################################################################################
#########################
Image Received.
Checking image ... OK
Uncompressed size: 268435968 = 0×10000200
Save as Default firmware/Backup firmware/Run image without saving:[D/B/R]?D
```

7. 同时 TFTP 服务器也会提示下载成功。

- Windows：



- macOS：



8. 随后 FortiGate 可能会提示重新部署启动分区，键入 `Y`（如果没有出现此提示，请忽略此步）。设备将重新格式化启动分区并使用以上步骤刷入的镜像版本启动。

```bash
Programming the boot device now.
The system must re-layout the boot device to install this firmware.
The default and backup firmware will be lost.
Continue:[Y/N]?y
..
Booting OS...
Reading boot image... 2967212 bytes.
Initializing firewall...
System is starting...
Resizing shared data partition...done
Formatting shared data partition  ...  done!
Starting system maintenance...
Scanning /dev/mmcblk0p1... (100%)
Scanning /dev/mmcblk0p3... (100%)

FortiGate-101E login:
```

9. 设备已加载 TFTP 导入的镜像作为主版本启动。

```bash
FortiGate-61F # get sys status | grep Version
Version: FortiGate-61F v7.2.8,build1639,240313 (GA.M)
Release Version Information: GA
```

软件签名增强

功能简介

“软件签名增强”功能是 FortiOS 7 推出的新功能，对重要的系统文件和可执行文件实现了 BIOS 级别签名和文件完整性检查。首先要保证在引导系统时加载的内核版本是经过签名的。系统一旦启动，将在活动系统上运行定期检查，以确认系统内核和进程以及 AV/IPS 引擎是经过签名的。此功能从 6.4.13、7.0.12、7.2.5、7.4.0、7.6.0 开始生效。

“软件签名增强”功能主要有三个模块。

启动保护

在设备启动时，会检查版本固件以及 AV/IPS 的签名合法性，在 BIOS 中支持 3 个“启动保护安全级别”设置（FortiGate 硬件产品），分别为 Level 0、Level 1、Level 2。：

- Level 1：当检测到系统内核文件的错误签名时（如非 GA 的临时版本），产生告警信息。





- Level 2（大多数设备默认）：当检测到系统内核文件的错误签名时（如非 GA 的临时版本），将 FortiGate 关机，FortiOS 固件和每个版本的 AV/IPS 引擎文件将由 Fortinet CA 和第三方权威 CA 双重签名。

```bash
Initializing firewall...
System is starting...
System file integrity init check failed!
The system is going down NOW !!
The system is halted.
```

- Level 0：静默检查，当检测到系统内核文件的错误签名时（如非 GA 的临时版本），不产生告警信息，也不会将设备关闭。

系统安全保障

篡改文件系统或文件的尝试将立即被检测到，会触发设备重新启动，并生成事件日志。

定期检查系统

系统将定期（每 6 小时）检查进程和文件名、权限，并对它们执行签名检查。这种定期检查每 6 小时进行一次。如果检测到异常，将触发设备重启，并生成事件日志，相关日志如下所示：

20230 - LOG_ID_SYS_SECURITY_WRITE_VIOLATION 432
Root filesystem is read only. Any modification triggers this log
20231 - LOG_ID_SYS_SECURITY_HARDLINK_VIOLATION 432
Attacker trying to replace symlink would trigger this log
20232 - LOG_ID_SYS_SECURITY_LOAD_MODULE_VIOLATION 433
Only kernel can load modules any unusual loading of modules triggers this log
20233 - LOG_ID_SYS_SECURITY_FILE_HASH_MISSING 434
File hashes are generated for legitimate files during bootup. If hash cannot be found, file may be suspicious as it could be a new routine inserted by attacker
20234 - LOG_ID_SYS_SECURITY_FILE_HASH_MISMATCH 434
File hashes are generated for legitimate files during bootup. If hash does not match when the file is exercised its an indication that it could have been modified by an attacker

IPS/AV 特征库签名

1. AV 和 IPS 包现在由 Fortinet CA 签名，以确保包的真实性。FortiGate 更新时将进行如下检查：

   - 自动更新时，只接受经过签名和验证的 IPS/AV 包。
   - 手动更新时，将接受签名和验证的 IPS/AV 包。如果 IPS/AV 包未签名，则有以下情况：
     - BIOS Security Level 0：接受新的 AV 和 IPS 包，即使它没有签名。
     - BIOS Security Level 1：显示警告并要求用户确认接受。
     - BIOS Security Level 2：显示错误并拒绝 AV 和 IPS 包。

2. 对于 HA，备设备将在保存签名包的同时从主设备同步签名文件。

3. 在 CLI 下，可以查看当前的 IPS/AV 特征库是否通过签名验证。

```bash
FortiGate # diagnose autoupdate signature check-all
aven(7,33) signature is valid.
virdb(2,2) signature is valid.
etdb(2,7) signature is valid.
exdb(2,4) signature is valid.
fcni(9,0) signature N/A.
contract(10,0) signature N/A.
idsen(30,80) signature is valid.
ipscfgscr(30,50) signature is valid.
fldb(34,2) signature is valid.
idsdb(4,24) signature is valid.
idsetdb(4,27) signature is valid.
idsurldb(5,1) signature is valid.
appdb(38,1) signature is valid.
isdb(39,1) signature is valid.
iot_detec(40,1) signature is valid.
geoip(28,0) signature N/A.
certdb(33,0) signature N/A.
update objects signature check finished.

FortiGate # diagnose autoupdate versions
AV Engine
————————
Version: 6.00288 signed
Contract Expiry Date: n/a
Last Updated using manual update on Mon May 15 18:35:00 2023
Last Update Attempt: n/a
Result: Updates Installed

Virus Definitions
————————
Version: 1.00000 signed
Contract Expiry Date: n/a
Last Updated using manual update on Mon Apr  9 19:07:00 2018
Last Update Attempt: n/a
Result: Updates Installed
```

```
Extended set
————————
Version: 1.00000 signed
Contract Expiry Date: n/a
Last Updated using manual update on Mon Apr  9 19:07:00 2018
Last Update Attempt: n/a
Result: Updates Installed
```

4. FortiGate 向 FortiGuard 自动更新时的验证签名 Debug 如下。

```bash
...
installUpdateObject[321]-Signature verified for obj 9, ret=0, data_len=90, obj_len=90, sig_len=0.
installUpdateObject[358]-Step 2:Prepare temp file for obj 9
installUpdObjRest[827]-Step 3:Signal parent not to respawn
installUpdObjRest[837]-Step 4:Kill daemon(s)
installUpdObjRest[871]-Step 5:Backup /etc/fcni.dat→/tmp/update.backup
installUpdObjRest[899]-Step 6:Copy new object /tmp/updrbmfhS→/etc/fcni.dat
installUpdObjRest[984]-Step 7:Validate object
...
```

功能影响

1. 当版本签名在启动检查失败时（如“临时发布”或“特殊分支”等固件），BIOS 安全级别为 2 时将强制 FortiGate 关机，不会重新启动。 2.**当 BIOS 安全级别为 Level 2 时，将不能降级到一些较旧的版本（6.0、6.2 或是 6.4.13、7.0.12、7.2.5、7.4.0 以下的版本）。解决方法是将 BIOS 安全级别从 2 更改为 1 或者 0。**
2. 在 RMA 期间，没有 BIOS 安全级别的设备（6.4.13，7.0.12，7.2.5，7.4.0 以下版本）可能被替换为具有安全级别支持的新硬件（默认安全级别为 2）。如果客户正在运行非 GA 版本（如 TOP 3 版本），则客户应在更换设备之前需将 BIOS 上的安全级别设置为 0。
3. 当 FortiGate 被攻击时，作为一种保护措施，试图篡改系统文件会导致 FortiGate 立即重启。修改 BIOS 安全级别不会影响此行为。
4. 如果系统文件校验和不匹配，在文件系统定期检查期间，FortiGate 也会重新启动。修改 BIOS 安全级别不会影响此行为。

查看 BIOS 安全级别

```
FortiGate # get system status
...
Version: FortiGate-601F v7.2.6,build1559,230822 (interim)
Security Level: 1
Firmware Signature: un-certified
...
```

提示

从 7.0.16、7.2.11、7.4.6、7.6.1 版本开始，更新了安全级别的命名规则。以前的数字级别 0、1 和 2 现在分别表示为 low、low 和 high。

修改 BIOS 安全级别

G 系列机型说明

对于 50G/51G、70G/71G、90G/91G Gen2、120G/121G Gen2、200G/201G、700G/701G 等机型，BIOS 菜单通常不提供 `[U] Set security level` 选项，请参考系统管理 → 固件与配置管理 → 固件版本管理 → FortiGate G 系列修改安全级别 ↗。

警告

请注意，调低该参数将影响设备或网络的整体安全状态。可能允许访问设备的用户在系统中安装或运行修改过的恶意代码。 强烈建议将 FortiOS 升级到支持 BIOS 安全检查的版本，最大限度地提高设备的安全等级。

如果 FortiGate 设备运行的 BIOS 版本为 5000100 或 6000100（或更新版本），用户将无法降级到 6.0、6.2 或这些：6.4.13、7.0.12、7.2.5、7.4.0 以下的 FortiOS 版本。

设备将无法正常启动，启动时将显示以下错误信息：

```
Booting OS ...
Fatal error: Loading FOS fails!
Please power cycle. System halted.
```

如果因为特殊要求，需要在已经具备 BIOS 安全级别功能的 FortiOS 版本上降级到以上 FortiOS 版本，请按照以下步骤更改 BIOS 安全级别。

1. 使用 PC 已连接设备的 Console 接口，重启 FortiGate。在重启过程中，FortiGate 会在 Console 中打印一条信息“press any key to display configuration menu”，然后按一个任意键进入 BIOS。

```bash
FortiGate-60F
Ver:05000009
Serial number: FGT60XXXXXXXXXXXX
CPU: 1200MHz
Total RAM: 2 GB
Initializing boot device ...
Initializing MAC ... NP6XLITE#0
Please wait for OS to boot, or press any key to display configuration menu. ⟵ Press any key.
```

2. 键入 `I`，进入 `System information`。

```bash
[C]: Configure TFTP parameters.
[R]: Review TFTP parameters.
[T]: Initiate TFTP firmware transfer.
[F]: Format boot device.
[I]: System information.    ⟵键入I
[B]: Boot with backup firmware and set as default.
[Q]: Quit menu and continue to boot.
[H]: Display this list of options.
```

3. 键入 `U`，进入 `Set security level`。

```bash
Enter C,R,T,F,I,B,Q,or H:
[S]: Set serial port baudrate.
[R]: Set restricted mode.
[T]: Set menu timeout.
[U]: Set security level.    ⟵键入U
[I]: Display system information.
[E]: Reset system configuration.
[P]: Normal POST test.
[Q]: Quit this menu.
[H]: Display this list of options.
```

4. 键入安全级别的数字 （0、1 或 2）。

```bash
Enter S,R,T,U,I,E,P,Q,or H:
[0]: Level 0 - Check image silently
[1]: Level 1 - Check image with result only
[2]: Level 2 - Check image and reinforce validity
Enter security level setting [2]:    ⟵键入安全级别的数字（0、1或2）
```

5. 随后按照说明退出 BIOS 菜单并启动设备（通常是键入 2 次 `Q`）。

常见问题

1. 哪些硬件支持 BIOS 安全级别设置?

```
FGT_40F, FGT_41F, FGT_41F_3G4G
FGT_60F, FGT_61F, FGR_60F, FGR60F_3G4G
FGT_70F, FGT_71F
FGT_80F, FGT_80F_BP, FGT_80F_POE
FGT_81F, FGT_81F_POE
FGT_100F, FGT_101F
FGT_200F, FGT_201F
FGT_400E_BP, FGT_400E, FGT_401E
FGT_1100E, FGT_1101E
FGT_1800F, FGT_1801F
FGT_2200E, FGT_201E
FGT_2600F, FGT_2601F
FGT_3300E, FGT_3301E
FGT_3400E, FGT_3401E
FGT_3500F, FGT_3501F FGT_3600E, FGT_3601E
FGT_4200F, FGT_4201F
FGT_4400F, FGT_4401F
FGT_600E, FGT601E
```

2. 是否对 VM 有影响?

- 虚拟机 VM 上无法修改“BIOS 安全级别”。从 `get system status` 中看到的 BIOS 安全级别取决于固件版本：
  - 6.4.16、7.0.16、7.2.9、7.4.4、7.6.0 及之后版本：Level 2。
  - 6.4.13 ~ 6.4.15，7.0.12 ~ 7.0.15，7.2.5 ~ 7.2.8，7.4.0 ~ 7.4.3：Level 1。
- 所以在 BIOS 安全级别为 2 的 VM 镜像中上传非 GA 的临时版本时，是无法上传的，必须先降级到 BIOS 安全级别为 1 的 VM 版本后，再导入非 GA 版本。

3. 如果 BIOS 设置为 Level 2，但启动的固件不支持软件签名增强，预期的行为是什么?

- 在引导时和系统启动时都不执行检查。

4. 如果 BIOS 设置安全级别为 0，在启动时是否还会执行检查?

- 会执行检查，它是静默地完成的，结果可以在 `get system status` 中看到。

5. 如果 BIOS 安全级别设置为 0，系统安全保障和定期检查是否会激活。

- 是的，如果加载的固件支持该特性，仍然会执行系统安全保障和定期检查。

6. 如何查看一个系统文件的 MD5 校验和?

```
FortiGate # diagnose sys csum /bin/init    //该命令需要指定具体的文件//
/bin/init:      df c1 b2 8f d9 a6 82 c5 d2 1e 45 87 e5 ec 98 f6

FortiGate # diagnose sys filesystem hash    //列出所有敏感文件的hash，7.4支持//
12b9377cbe7e5c94e8a70d9d23929523d14afa954793130f8a3959c7b849aca8        /data/etc/fdsm_mgmt_id.dat
00664fbc0fdafced6c4e0f77570f371cf0d0b85e0429c10d1376a81941f17289        /data/etc/cloudinit.log
48bd618e47b0dc185ff512b75c9fc6f7c5437d8aec6d7f3527c41d6687375d59        /data/etc/ips.mudb.rules.x
3af43771ffc8990ed38c7bd10ad37e60391e565c51a133e0acdc63539661ef11        /data/etc/applications.csv
423720df1a74eca781eaa37a388d86f438541d7f0624cd8f0c5781c843a5bded        /data/etc/sample.pac
db057221b788f6ed5c8b629ad83d6f8fca8dc35274bce13be6520595c09f1f65        /data/etc/modem_list_fex.conf
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data/etc/virexdb
0ceb39183d6ce36e7bd6bb5edfe6beacac702be9bfeac3bf0e2066df3643af1d        /data/etc/ips.iot.rules
a38bba93dc8691ca7367d134f575df31ba51c5c591309447338d10d08812910c        /data/etc/geoip_db.gz
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data/etc/virext
0cbf59ea97ced14b3dd844cb4dcd784ddcae7d5ef38c2a6081c9e369c25d5ac6        /data/etc/protocols.csv
896b71b3d9b209d339213f9d4af4088d3addd891cd292e93b5168eddb36b599a        /data/etc/virfldb.x
dbe703c06735797ac86570442b39827ca1844fea6312e48fee8c2d80647392d0        /data/etc/fgt.crt
3e36debe496618432707dc680ef78fc59c2d9ecc38b3f65a2f055dc826492492        /data/etc/fcni.dat
```

7. 如果遇到系统文件被篡改或是误报，我应该怎么办?

- 如下所示，设备 Console 打印信息：

```
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/libcrashpad.so) is missing hash
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/flatkc_info) is missing hash
```

- 如果您看到文件缺失散列值，如以下（但不限于），可能是攻击造成，可以访问此链接联系 PSIRT 团队：https://www.fortiguard.com/faq/psirt-contact ↗ ，会有 Fortinet 相关人员与您取得联系，或联系 Fortinet 技术支持，并请提前收集以下信息：
  - 系统分区文件内容：

```bash
cofig global (如果开启了多VDOM)

config system console
    set output standard
end

↓↓↓请分行执行，不要一次性粘贴↓↓↓
get system status
fnsysctl ls -la /
fnsysctl ls -la /bin
fnsysctl ls -la /sbin
fnsysctl ls -la /lib
fnsysctl ls -la /tmp
fnsysctl ls -la /usr
fnsysctl ls -la /usr/bin
fnsysctl ls -la /var
fnsysctl ls -la /data
fnsysctl ls -la /data2
fnsysctl ls -la /data/lib
fnsysctl ls -la /data/etc
fnsysctl ls -la /data/bin
fnsysctl ls -la /data/cmdb
fnsysctl ls -la /data/config
diagnose sys csum /data/rootfs.gz
diagnose sys csum /data/flatkc
diagnose sys csum /data/lib
diagnose sys csum /bin
diagnose sys csum /bin/sysctl
diagnose sys csum /bin/smit
diagnose sys csum /bin/init
diagnose sys csum /bin/smartctl
diagnose sys csum /bin/lspci
diagnose sys csum /sbin/init
fnsysctl ps
execute tac report
```

- 文件系统 Hash 信息：

```bash
cofig global (如果开启了多VDOM)
get system status
diagnose sys filesystem hash
```

- 系统配置文件。
- 系统日志文件。

G 系列修改安全级别

功能简介

本文说明如何在 FortiGate G 系列机型上调整 BIOS 安全级别（Security Level）。

在多数 FortiGate 机型上，可以通过开机 BIOS 菜单中的 `[U] Set security level` 修改安全级别；但在部分 G 系列机型上，该选项不再提供，需改用设备硬件上的 `Signed Firmware` 物理开关进行切换。

适用范围

- FortiGate/FortiWiFi 50G/51G 系列。
- FortiGate/FortiWiFi 70G/71G 系列。
- FortiGate 90G/91G Gen2 及以上（Part Number `Pxxxxx-11-01` 及以上）。
- FortiGate 120G/121G Gen2 及以上（Part Number `Pxxxxx-2x-01` 及以上）。
- FortiGate/FortiWiFi 200G/201G 系列。
- FortiGate 700G/701G 系列。

> **提示**
>
> 对于 30G/31G、90G Gen1、120G Gen1 等仍支持 BIOS 菜单修改的机型，可参考系统管理 → 固件与配置管理 → 固件版本管理 → 软件签名增强 ↗ 中的 BIOS 修改方法。

安全级别说明

- `High`（默认）：阻止未签名固件。
- `Low`：允许未签名固件安装，但会告警。

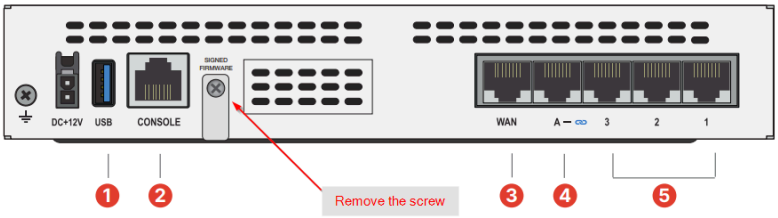
> **相关信息**
>
> 从 FortiOS `7.0.16`、`7.2.11`、`7.4.6`、`7.6.1` 开始，安全级别名称统一为 `Low/High`（早期文档中的 `0/1/2` 对应为 `Low/Low/High`）。

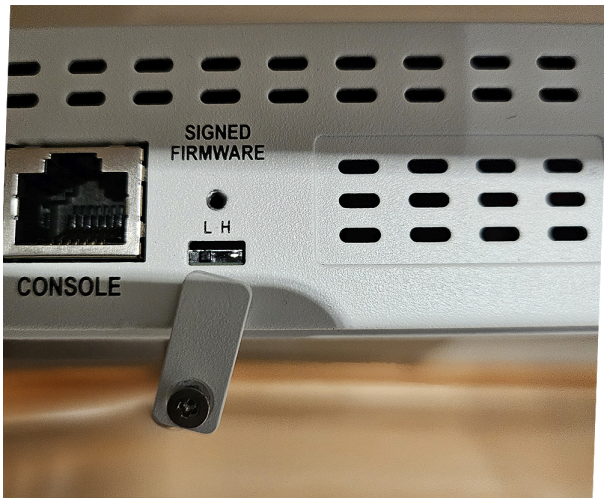
开关位置

- `50G/70G/90G Gen2`：开关位于设备后面板（状态 LED 的对侧）。
- `120G Gen2/200G/700G`：开关位于设备前面板。

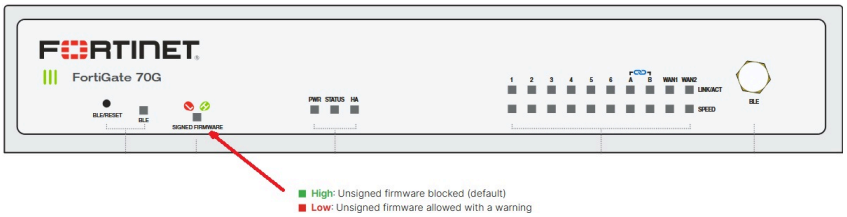
原文图片示意

50G 系列（后面板盖板与开关）

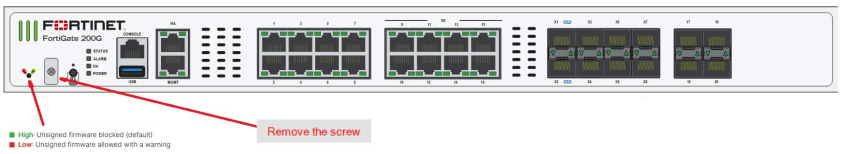


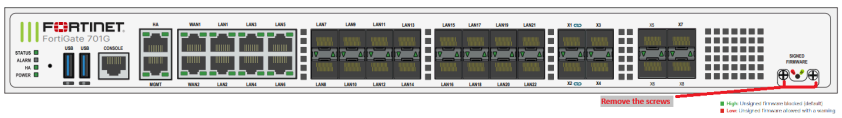


70G 系列（前面板 LED）

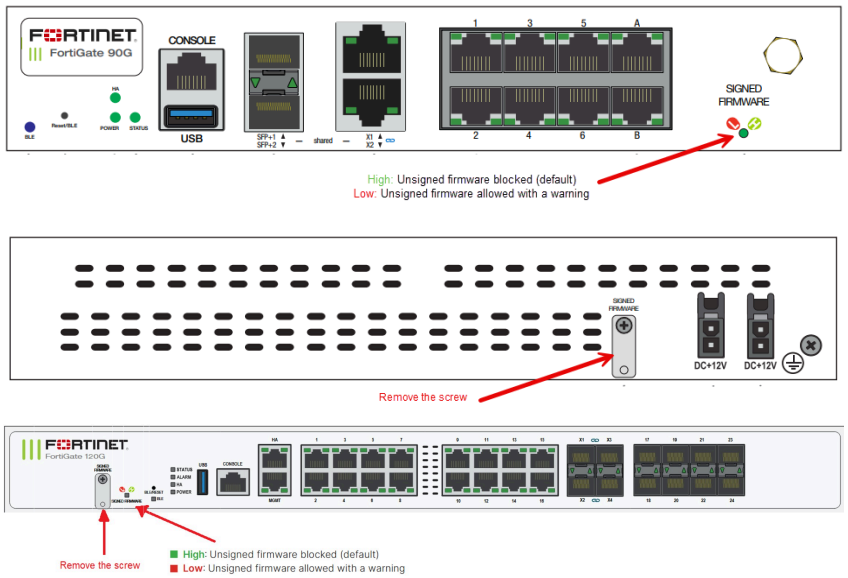


200G/700G 系列（前面板开关与 LED）





90G Gen2/90G Gen1/120G Gen2 代际差异示意



操作步骤

1. 断电并确认设备处于安全可操作状态，拆下 `Signed Firmware` 金属挡板固定螺丝。
2. 将物理开关拨到目标档位：
   - `H` / `High`：高安全级别（默认）。
   - `L` / `Low`：低安全级别。
3. 重新上电并重启设备。
4. 登录 CLI，使用以下命令确认当前安全级别：

```bash
# get system status | grep -i "security level"
Current Security Level: Low
```

注意事项

- 修改物理开关后，BIOS 安全级别会在下次重启后生效。
- `get system status` 可用于对比当前 BIOS 安全级别与开关设置是否一致。
- 当安全级别从 `High` 调整为 `Low` 时，会产生系统事件日志：`22906 - LOG_ID_SECURITY_LEVEL_CHANGE`。
- 90G/120G 存在多硬件代际差异，需先通过 Part Number 确认是否具备 `Signed Firmware` 硬件开关。

过期 License 升级限制

功能介绍

FortiGate 的固件升级能力由 `Firmware & General Updates (FMWR)` 合同控制。

从 FortiOS 7.4.2 开始，系统会阻止“支持合同已过期”的设备升级到不满足许可条件的目标版本（主要版本、次要版本以及相关补丁版本）。

版本号概念说明

FortiOS 版本通常写作 `x.y.z`，例如 `7.6.3`：

- `x`：主要版本（Major），例如 `7.x` 到 `8.x`。
- `y`：次要版本（Minor），例如 `7.4.x` 到 `7.6.x`。
- `z`：补丁版本（Patch），例如 `7.6.1` 到 `7.6.2`。

本文提到的“相关补丁版本”，指的是属于某个主/次版本分支的补丁版本。例如在 `7.6.x` 分支中，`7.6.1`、`7.6.2`、`7.6.3` 都是该分支的相关补丁版本。

重要

本文中的 GA 日期示例用于解释许可证校验机制，不代表 FortiOS 官方发布时间计划。

许可证校验规则

设备升级到目标版本前，会比较以下两个时间：

- 当前 FMWR 合同到期日期。
- 目标版本所属主/次版本的首个 `GA .0` 发布日期。

判定规则如下：

- 到期日期 **晚于或等于**目标 `GA .0` 发布日期：允许升级。
- 到期日期 **早于**目标 `GA .0` 发布日期：拒绝升级。

例如，目标版本是 `7.6.3`，实际对比的是 `7.6.0 GA` 的发布日期，而不是 `7.6.3` 的发布日期。

升级权限示例

假设各次版本首个 GA 版本发布日期如下（示例）：

版本	首个 GA 版本	发布日期
7.4.x	7.4.0 GA	2023-05-08
7.6.x	7.6.0 GA	2024-03-31
7.8.x	7.8.0 GA	2025-03-31

不同许可证到期日期下的升级权限示例：

FMWR 到期日期	可升级到 7.4.x	可升级到 7.6.x	可升级到 7.8.x
2025-03-31 或之后	是	是	是
2025-03-25	是	是	否
2024-03-25	是	否	否
2023-05-02	否	否	否

合同状态对应行为

合同有效

支持合同有效时，可进行以下操作：

- 主版本升级（例如 `7.0 → 8.0`）。
- 次版本升级（例如 `7.4 → 7.6`）。
- 补丁版本升级（例如 `7.6.1 → 7.6.2`）。
- 版本降级。

合同无效、过期或版本达到 EOES

在支持合同无效、过期，或版本达到 EOES（End of Engineering Support）时：

- 强制升级会升级到 **当前次版本**的最新补丁版本，不会自动跨次版本升级（例如不会从 `7.4.x` 自动升级到 `7.6.x`）。
- FortiGuard 在线升级按钮会受限（`Confirm and Backup Config` 置灰）。
- GUI 文件上传升级与 CLI 升级会按许可证规则被拒绝。

> **相关信息**
>
> 自动升级的详细行为请参考：
>
> - 系统管理 → 固件与配置管理 → 固件版本管理 → 有效 License 自动升级 ↗。
> - 系统管理 → 固件与配置管理 → 固件版本管理 → 过期 License 强制升级 ↗。

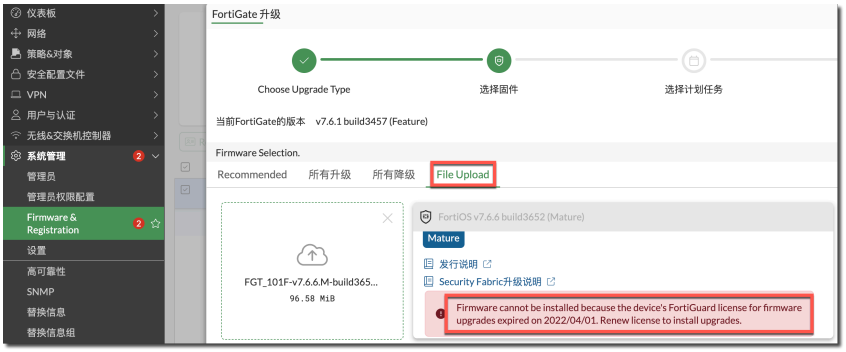
降级说明

跨主/次版本降级通常不会被阻止，因为目标旧版本的 `GA .0` 发布时间通常早于当前许可证到期日期。

禁止升级提示信息

GUI 文件上传升级

当合同到期日期早于目标版本所属 `GA .0` 发布日期时，升级会被拒绝并显示错误提示。



CLI 文件上传升级

```bash
# execute restore image tftp FGT_101F-v7.6.6.M-build3652-FORTINET.out 172.16.200.55
......
Firmware update licence is expired! Please update to a valid licence.
Command fail. Return code -180
```

GUI 通过 FortiGuard 升级

合同未续费前，FortiGuard 升级不可执行，但仍可查看 `Latest`、`All Upgrades`、`All Downgrades` 列表中的固件信息。



CLI 通过 FortiGuard 升级

```bash
# execute restore image management-station 07006000FIMG0023906006

Firmware updates from Management Station is not available.
 node_check_object fail! for management-station 07006000FIMG0023906006

value parse error before '07006000FIMG0023906006'
Command fail. Return code -651
```

排查方法

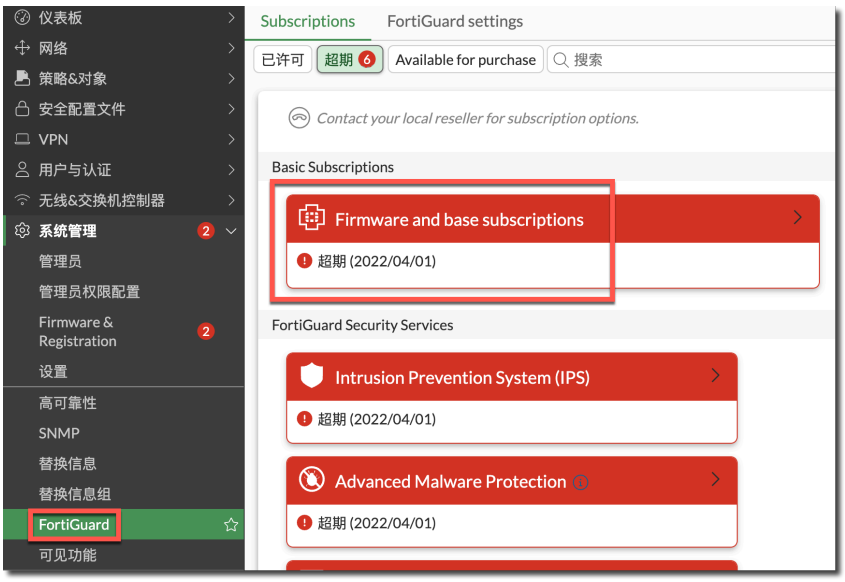
查看 FMWR 合同到期日期

CLI

```bash
FortiGate # diagnose test update info contract | grep FMWR
    FMWR,Fri Apr  1 2022
SerialNumber=FG101FTK20007637|Contract=AVDB-1-06-20220402:0:1:1:0*AVEN-1-06-20220402:0:1:1:0*NIDS-1-06-
20220402:0:1:1:0*SPRT-1-20-20220402:0:1:1:0*FMWR-1-06-20220402:0:1:1:0*FRVS-1-06-20220402:0:1:1:0*FURL-1-06-
20220402:0:1:1:0*HDWR-1-05-20220402:0:1:1:0*SBCL-1-06-20220402:0:1:1:0*SPAM-1-06-20220402:0:1:1:0*ZHVO-1-06-
20220402:0:1:1:0*ENHN-1-20-20220402:0:1:1:0*COMP-1-20-
20220402:0:1:1:0|AccountID=bbai@fortinet.com|Industry=Technology|Company=Fortinet|UserID=1134101|
SupportTypeDesc=AVDB:Advanced Malware Protection*COMP:*ENHN:*FMWR:Firmware & General Updates*FRVS:Vulnerability
Management*FURL:FortiGuard URL, DNS & Video Filtering Service*HDWR:Hardware*NIDS:FortiGuard IPS Service*SBCL:FortiSandbox
Cloud*SPAM:AntiSpam*SPRT:*ZHVO:FortiGuard Virus Outbreak Protection Service
```

GUI

- 系统管理 → FortiGuard。



- 仪表盘 → 状态（鼠标悬停在“许可”小部件的“更新”部分）。



检查 FMWR 合同是否存在

- 使用 `diagnose test update info contract` 命令，在输出的 `System contracts` 段确认存在 `FMWR`。

```bash
System contracts:
    ENHN,Thu May  1 2025
    COMP,Thu May  1 2025
    FMWR,Thu May  1 2025
```

- 若 FMWR 信息缺失，FortiGate 也不允许执行固件升级。该场景常见于 EVAL 许可证（不支持 FMWR）。

有效 License 自动升级

重要

自动升级功能的初衷是为了规避已经发现的系统漏洞，防止黑客利用漏洞攻击。但在一些重要的生产环境中，不在维护时间内的更新和重启会造成严重的生产事故，所以建议主动安排业务窗口期升级新版本。

功能介绍

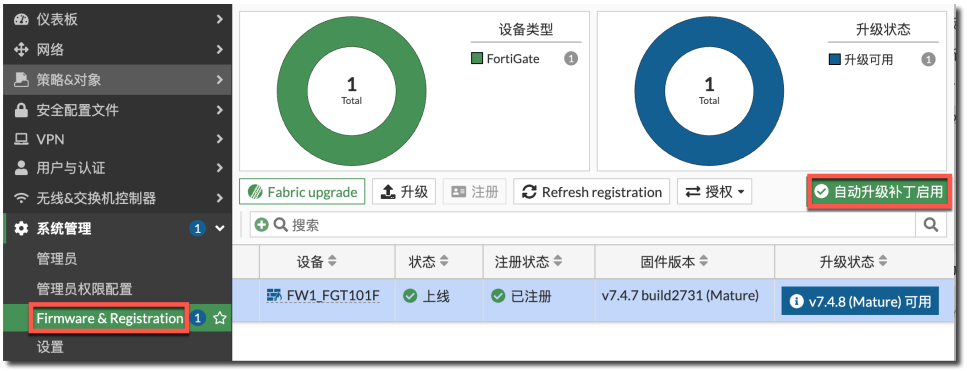
从 FortiOS 7.2.1 开始，FortiGate 支持在线自动升级功能。

- 只能升级相同的次要版本（x.y.z 中的 y）的小版本（x.y.z 中的 z），不能在大版本之间自动升级。例如，7.2.5 版本只能自动升级到 7.2.X，不能自动升级到 7.4.X。
- 如果存在 Mature 类型的新版本，自动升级会按照升级路径的版本顺序进行。
- 如果不存在 Mature 类型的新版本，则会参考 `diagnose fdsm image-upgrade-matrix` 中的升级路径。
- 每次自动升级只会升级一个版本，如果升级路径中存在多个版本，会产生多次重启。
- 命令 `diagnose test application forticldd 13` 可以查看自动升级的开启状态以及将要自动升级的版本和具体时间。

功能配置

GUI 配置

1. 进入设备的管理页面，访问“系统管理 → Firmware & Registration”，右上角显示“自动升级补丁启用”，点击该按钮。



2. 在弹出的“自动补丁升级”窗口中，可以禁用补丁自动升级或配置自动升级的相关参数，完成配置后点击“OK”按钮下发配置。

   - 升级时间表：
     - 延迟：当检测到新版本后，延迟升级的时间（默认 3 天）。
     - Specify days：当检测到新版本后，特定一周中的某一天或几天升级，如果选了多天，则会在最近一天的可用时间升级。
   - Install during specified time：升级当天执行升级动作具体的时间段，会在选定的时间内的随机时间安排升级。



CLI 配置

```bash
config system fortiguard
    set auto-firmware-upgrade { enable | disable* }
    set auto-firmware-upgrade-day {sunday monday tuesday wednesday thursday friday saturday}
    set auto-firmware-upgrade-start-hour <integer>
    set auto-firmware-upgrade-end-hour <integer>
    set auto-firmware-upgrade-delay <integer>
end
```

- `auto-firmware-upgrade-day`：指定某一天或某几天才能执行自动升级，比如只能周末升级版本，缺省是每周 7 天全选。
- `auto-firmware-upgrade-start-hour` / `auto-firmware-upgrade-end-hour`：指定升级时间点，缺省是 2:00~4:00 的随机时间点。
- `auto-firmware-upgrade-delay`（7.2.6 新增功能）：发现有新版本后延迟多少天再升级（0~14 天，缺省 3 天）。当 `set auto-firmware-upgrade-delay` 配置为 0 时，`auto-firmware-upgrade-day` 配置才会生效。

自动升级默认状态

在不同的版本/FortiGate 型号条件下，自动升级功能的默认开启状态如下所示：

设备版本	FortiGate 型号	自动升级默认状态
7.2.1~7.2.5/7.4.0~7.4.4/7.6.0	所有	关闭
7.2.6~7.2.X	<100	开启
	⩾100	关闭
7.4.5~7.4.X/7.6.1~7.6.X	所有	开启

查看自动升级状态

通过 `diagnose test application forticldd 13` 命令可以看到当前设备自动升级的相关状态，常见状态有如下几种。

- FortiGate 当前没有检测到新版本，没有升级计划，计划了下次检测新版本的时间为 `Wed Jun 4 03:07:58 2025`（计划检查时间也遵从已配置的自动更新时间范围）。

```bash
FortiGate # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Wed Jun  4 03:07:58 2025
```

- FortiGate 检测到新版本 7.4.8，并计划了升级时间为 `Thu Jun 5 05:13:10 2025`。

```bash
FortiGate # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Thu Jun  5 02:13:51 2025
        New image 7.4.8b2795(07004000FIMG0023904008) installation is scheduled to
                start at Thu Jun  5 05:13:10 2025
                end by Thu Jun  5 07:00:00 2025
        Last upgrade check executed at (local time) Wed Jun  4 03:07:58 2025
```

- 当 FortiGate 检测到新版本并计划升级时间后，会向设备注册的邮箱发送一封通知邮件，包含系统计划自动升级的事件日志。

```bash
date=2025-06-04 time=11:07:58 devid="FG101FTK20007637" devname="FW1_FGT101F" eventtime=1749006478360626440 tz="+0800"
logid="0100032263" type="event" subtype="system" level="notice" vd="root" logdesc="Automatic firmware upgrade schedule
changed" user="system" msg="System patch-level auto-upgrade new image installation (re)scheduled to between local time
```

```
Thu Jun  5 12:13:10 2025 and local time Thu Jun  5 14:00:00 2025."
```

- 设备关闭了自动升级功能。

> **重要**
>
> 如果 FortiGate 被 FortiCloud 管理，即使这里的状态为 `Disabled`，仍然可能通过 FortiCloud 自动升级，详见下方的“关闭自动升级 → 设备被 FortiCloud 管理↗”章节。

```bash
FortiGate # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Disabled.
```

关闭自动升级

> **提示**
>
> - 如果来不及安排主动升级窗口，可以临时关闭自动升级功能。
> - 如果在自动升级开启的状态下，已经计划了升级时间，用如下方式关闭后，已有的升级计划也会终止。

设备被 FortiGate Cloud 管理

> **重要**
>
> 如果设备加入了 FortiGate Cloud，即使设备自身关闭了自动升级，仍然会通过 FortiGate Cloud 进行自动升级。

1. 检查设备是否被 FortiGate Cloud 管理。



2. 如果设备已配置 FortiGate Cloud 管理，不仅要关闭设备自身的自动升级功能，还需要同时关闭 FortiGate Cloud 下的自动升级功能。

```bash
config system fortiguard
    set auto-firmware-upgrade disable
end

config system central-management
    set allow-push-firmware disable
    set allow-remote-firmware-upgrade disable
end
```

设备被 FMG/Security Fabric 管理

只需要关闭设备自身的自动升级功能。

```bash
config system fortiguard
    set auto-firmware-upgrade disable
end
```

过期 License 强制升级

触发条件

在 7.4.8、7.6.4、8.0.0 及以后版本，以下情况会触发强制自动升级（FortiGate 所有型号），**不受有效 License 自动升级 ↗设置的影响**：

1. **过期的 Firmware & General Updates 合同**：若设备未绑定有效的 FMGR 合同（Firmware & General Updates）或该合同已过期，系统将强制升级至最新补丁版本（主次版本号保持不变，例如 7.4.X 会升级到 7.4 最新的版本，而不会升级到 7.6.X）。
2. **固件版本支持周期终止**：当设备固件达到官方 **工程支持终止**日期（End of Engineering Support，简称 EOES，该时间可以登录 https://support.fortinet.com/support/#/lifecycle ↗ 查询）时，系统将强制升级至最新版本（主次版本号保持不变，例如 7.4.X 会升级到 7.4 最新的版本，而不会升级到 7.6.X）。

   提示

   注意 EOES 时间并非 EOS（End of Support）时间，通常比 EOS 时间要早：

   - EOES 指 Fortinet 不再对该版本做代码更新（除非是 critical 级别的问题）。
   - EOS 指 Fortinet 不再对该版本提供技术支持。
   - 关于产品生命周期的更多信息请参考系统管理 → 固件与配置管理 → 固件版本管理 → 产品生命周期 ↗章节。

   相关信息

   2026-03-31，Fortinet 发布公告 **CSB-260330-1**：FortiOS、FortiManager、FortiAnalyzer 的 7.4 与 7.6 版本支持生命周期延长 12 个月，EOES 与 EOS 日期整体顺延 1 年。

   因此，7.4.x/7.6.x 分支基于 EOES 触发的强制升级时间点也会相应后移；请以 https://support.fortinet.com/support/#/lifecycle ↗ 页面显示时间为准。
3. 如果设备已经加入了 FortiManager 管理，强制升级机制会被禁用。

避免强制升级的 Workaround

1. 将设备加入 FortiManager 管理（推荐）。

```bash
config system central-management
    set type fortimanager
end

The Serial Number for FortiManager is not entered.
In order to verify identity of FortiManager serial number is needed.
If serial number is not set, connection will be set as unverified.
FortiGate can establish a connection to obtain the serial number now.Do you want to try to connect now? (y/n)n
```

提示

- 配置中不需要填写实际的 FortiManager IP 地址。只要启用了 FortiManager 管理类型，强制升级机制就会被禁用。
- 在执行上述命令后，系统可能会弹出确认信息，键入 `n` 以提交配置。
- 启用 `fortimanager` 模式并不会自动开放防火墙的入站端口。接口的访问权限是由接口下的 `fgfm` 选项独立控制的。
- 如果实际环境中并不使用 FortiManager，建议检查并确保所有网络接口上的 `fgfm` 选项已被禁用（该选项仅用于 FortiManager 主动连接 FortiGate 的模式，如果是 FortiGate 主动连接 FortiManager，也不需要开启 `fgfm`）。

```bash
FortiGate # show sys int | grep fgfm -f
config system interface
    edit "wan1"
        set vdom "root"
        set ip 192.168.100.99 255.255.255.0
        set allowaccess ping https ssh http fgfm ←——
        set type physical
        set snmp-index 3
    next
end

config system interface
    edit wan1
        unselect allowaccess fgfm
    next
end
```

2. 将设备加入 Security Fabric 中。

延迟强制升级

- 强制自动升级可以手动延迟（一周），但无法取消。

提示

一旦新版本升级被确认，升级必须在 14 天内完成（此时将无法延期），无论执行几次手动延迟命令。

```bash
execute auto-upgrade delay-installation
```

- 查看当前强制自动升级的状态。

```bash
execute auto-upgrade status
```

- GUI 的升级页面会显示与常规自动升级相似但不同的提示信息，以便用户明确识别当前为强制自动升级。

配置管理

- 配置备份及恢复
- 备份配置文件到 Flash
- 备份不包含密码的配置文件
- 通过 SCP 备份配置文件
- 恢复出厂
- 配置回退

配置备份及恢复

需求

将防火墙的配置保存并导出备份，在必要时用来恢复之前的配置。

配置要点

1. 保存配置
2. 导出配置
3. 恢复配置

> **重要**
>
> - 导入的配置文件必须是 conf 格式，否则无法识别。
> - 导入配置后需要重启才会生效。
> - 备份配置设置的密码必须谨记，如遗忘则无法导入恢复。
> - 请使用具有“super_admin”权限的管理员导出配置文件，而不是“prof_admin”，否则导出的配置文件不完整（缺少 super_admin 账户配置，导入设备后无法再导入其他配置文件）。

操作步骤

保存配置

1. Web：Web 上配置完，及时生效，且自动保存配置，无需手工保存。 每次修改配点击确认按钮后，配置自动保存。

2. CLI：命令行下输入 next 或 end 后配置生效并自动保存。

导出配置

1. 进入 Web 页面点击右上方登录账号 admin，找到配置，点击 "备份" 按钮，对配置进行保存。

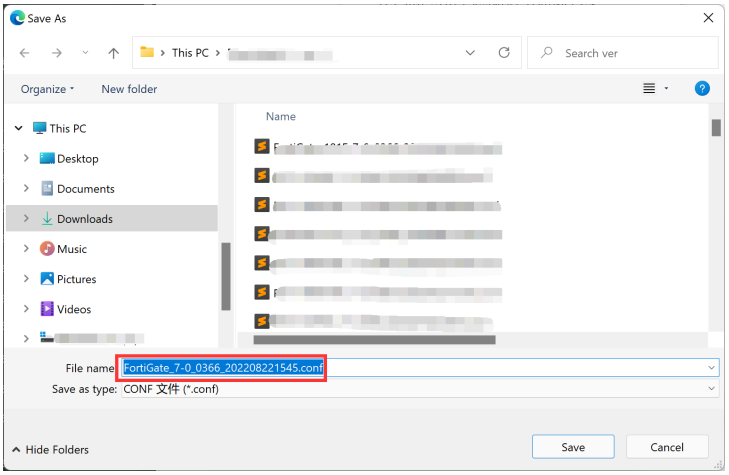
> **重要**
>
> 请使用具有“super_admin”权限的管理员导出配置文件，而不是“prof_admin”，否则导出的配置文件不完整（缺少 super_admin 账户配置，导入设备后无法再导入其他配置文件）。



2. 支持选择是否对配置文件进行加密，根据实际情况选择是否加密（加密需设置密码，请牢记该密码），点击确认。



3. 点击确定并把配置文件备份到本地磁盘上选择的位置，文件格式为 *.conf。



4. 同时，FortiGate 也支持在 CLI 下通过 FTP/TFTP/SFTP/USB（需将包含配置文件的 U 盘插入 FortiGate 的 USB 接口）等方式将配置文件导出，命令格式如下：

```
FTP:
exe backup config ftp <配置文件名称> <FTP服务器IP或域名>[:FTP端口号，标准端口无需填写] <FTP用户名> <FTP密码> [配置文件加密密码，如果不需要则不填]
举例:
exe backup config ftp FortiGate_7-2_1396_202305121543.conf 192.168.1.100 user1 Fortinet123#

SFTP:
exe backup config sftp <配置文件名称> <SFTP服务器IP或域名>[:SFTP端口号，标准端口无需填写] <SFTP用户名> <SFTP密码> [配置文件加密密码，如果不需要则不填]
举例:
exe backup config ftp FortiGate_7-2_1396_202305121543.conf 192.168.1.100 user1 Fortinet123#

TFTP:
exe backup config tftp <配置文件名称> <TFTP服务器IP或域名>[:TFTP端口号，标准端口无需填写] [配置文件加密密码，如果不需要则不填]
举例:
exe backup config tftp FortiGate_7-2_1396_202305121543.conf 192.168.1.100

USB:
exe backup config usb <配置文件名称> [配置文件加密密码，如果不需要则不填]
举例:
exe backup config usb FortiGate_7-2_1396_202305121543.conf
```

恢复配置

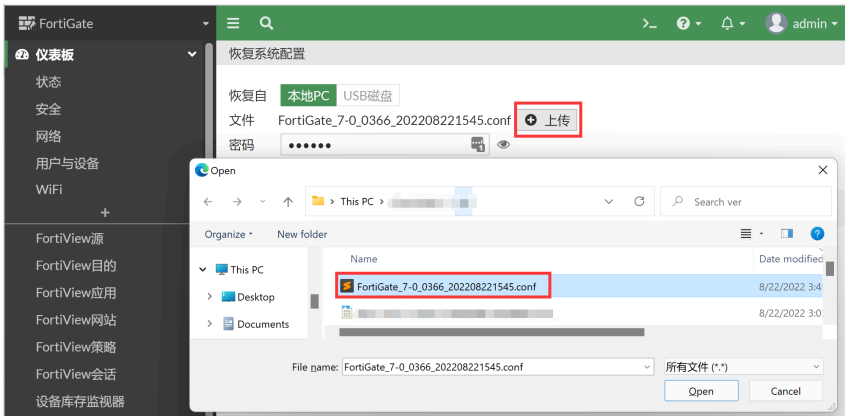
1. 页面右上角点击 admin，找到配置，点击 "恢复" 按钮，用原来存储在本地的配置文件恢复防火墙配置。

> **重要**
>
> 请使用具有“super_admin”权限的管理员导出配置文件，而不是“prof_admin”，否则导出的配置文件不完整（缺少 super_admin 账户配置，导入设备后无法再导入其他配置文件）。

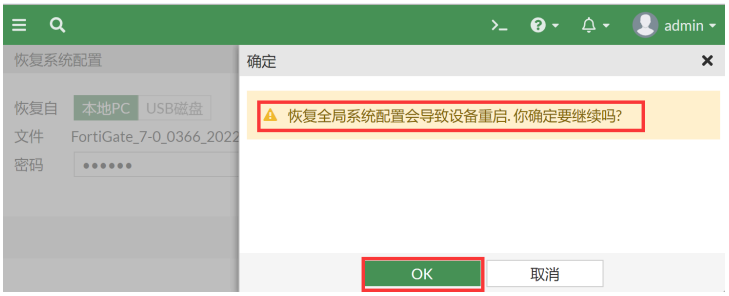


2. 弹出如上恢复界面（如配置文件未加密，则无需输入密码）。

3. 点击浏览，选择之前备份的配置文件，输入之前备份配置文件时设置的密码，点击确认按钮即可。



4. 导入成功后会提示系统重启后生效，故需要重启设备。点击“OK"。



5. 同时，FortiGate 也支持在 CLI 下通过 FTP/TFTP/USB（需将包含配置文件的 U 盘插入 FortiGate 的 USB 接口）等方式将配置文件导入，命令格式如下：

```
FTP:
exe restore config ftp <配置文件名称> <FTP服务器IP或域名>[:FTP端口号，标准端口无需填写] <FTP用户名> <FTP密码> [配置文件加密密码，如果导出时未配置则不填]
举例:
exe restore config ftp FortiGate_7-2_1396_202305121543.conf 192.168.1.100 user1 Fortinet123#

TFTP:
exe restore config tftp <配置文件名称> <TFTP服务器IP或域名>[:TFTP端口号，标准端口无需填写] [配置文件加密密码，如果不需要则不填]
举例:
exe restore config tftp FortiGate_7-2_1396_202305121543.conf 192.168.1.100

USB:
exe restore config usb <配置文件名称> [配置文件加密密码，如果不需要则不填]
举例:
exe restore config usb FortiGate_7-2_1396_202305121543.conf
```

重要

如果是 HA 环境，则导入配置文件后，主备设备均会重启。

验证效果

设备重启后还原为之前的配置。

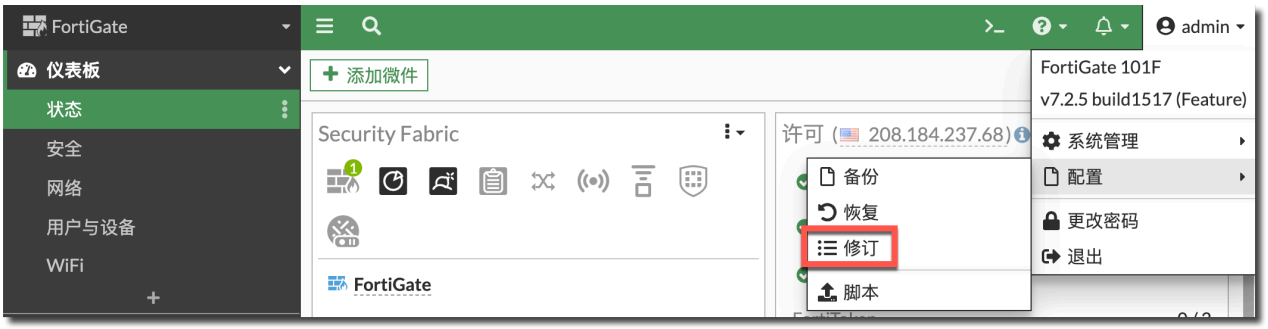
备份配置文件到 Flash

功能介绍

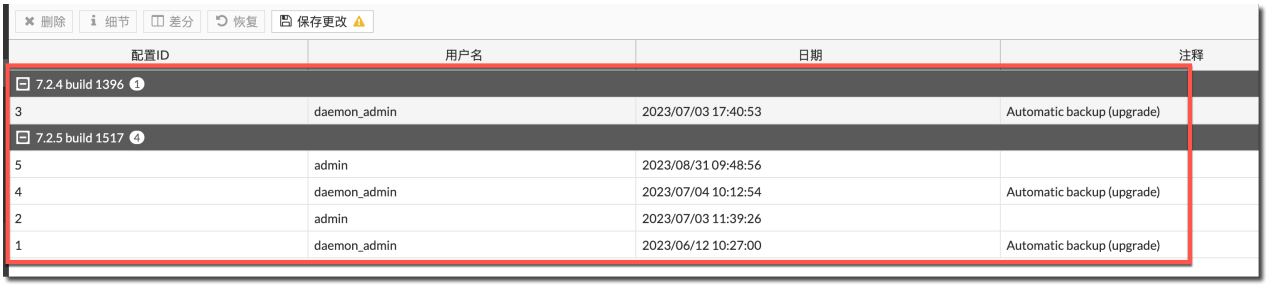
FortiGate 可以备份配置文件到 Flash 闪存中，保存后，可以在多个配置文件间进行切换，以及对比多个配置文件间的差异，重启不会丢失 Flash 中的所有配置文件。

使用方法

1. 登录 FortiGate，点击右上角的管理员账户 → 配置 → 修订按钮，进入修订页面。



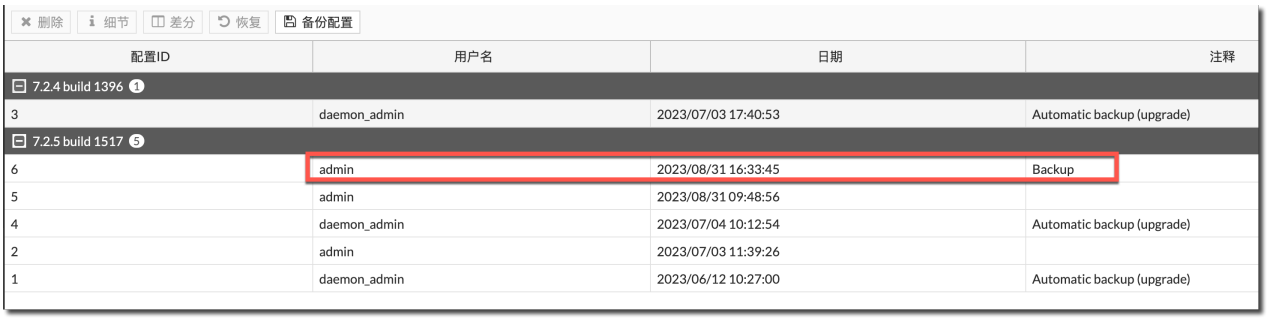
2. 在修订页面中，可以看到已经备份到 Flash 中的配置文件及对应版本分类（对应的 CLI 如下所示）。




```
FortiGate # execute revision list config
Last Firmware Version: V0.0.0-build000-REL0

ID  TIME                 ADMIN           FIRMWARE VERSION          COMMENT
1   2023-06-12 10:27:00  daemon_admin    V7.2.5-build1517-REL0     Automatic backup (upgrade)
2   2023-07-03 11:39:26  admin           V7.2.5-build1517-REL0
3   2023-07-03 17:40:53  daemon_admin    V7.2.4-build1396-REL0     Automatic backup (upgrade)
4   2023-07-04 10:12:54  daemon_admin    V7.2.5-build1517-REL0     Automatic backup (upgrade)
5   2023-08-31 09:48:56  admin           V7.2.5-build1517-REL0
```

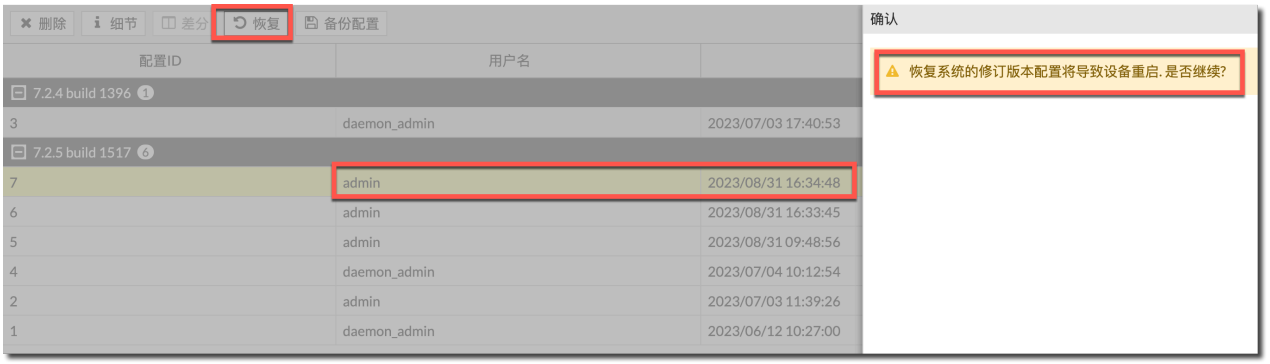
3. 点击页面上的“保存更改”按钮，可以保存当前的版本配置到 Flash 中，对应的 CLI 如下所示。





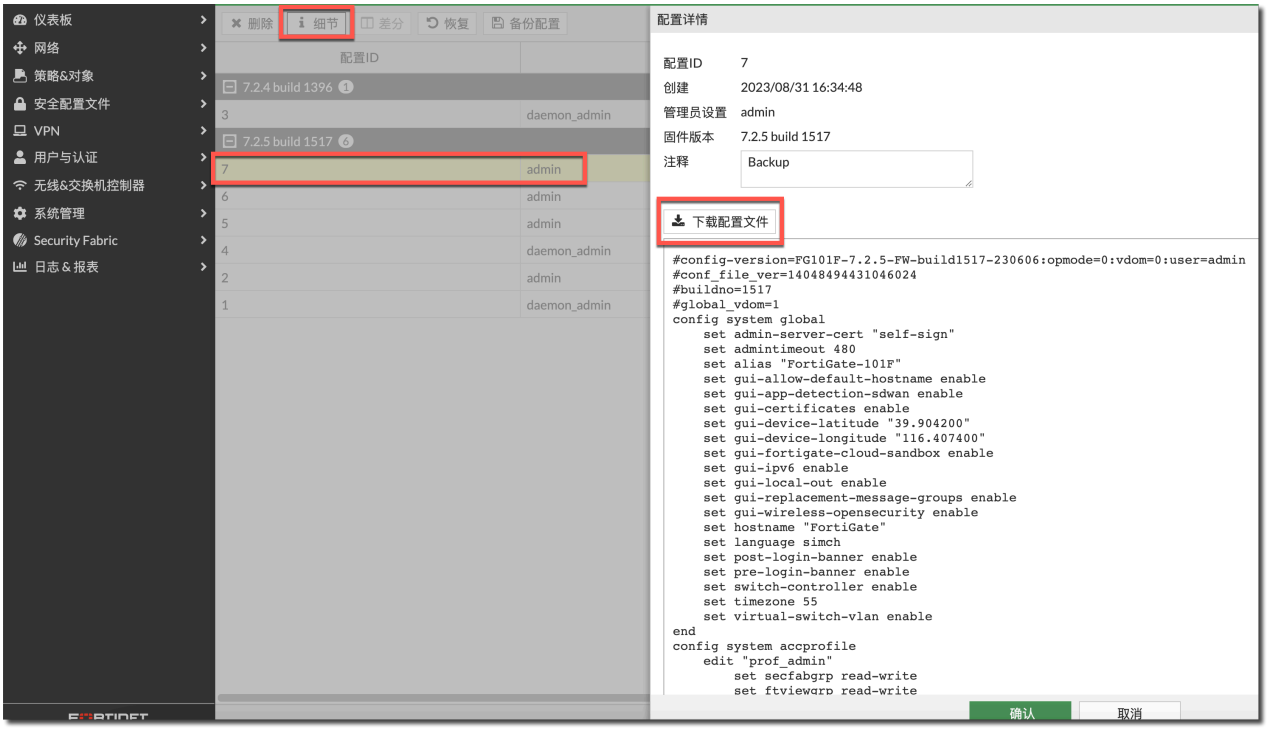

```
FortiGate # exe backup config flash Backup
Please wait...
Config backed up to flash disk done.
```

4. 选中某个配置文件，点击“恢复”按钮，可以将此配置恢复到当前 FortiGate 上，需要重启设备。

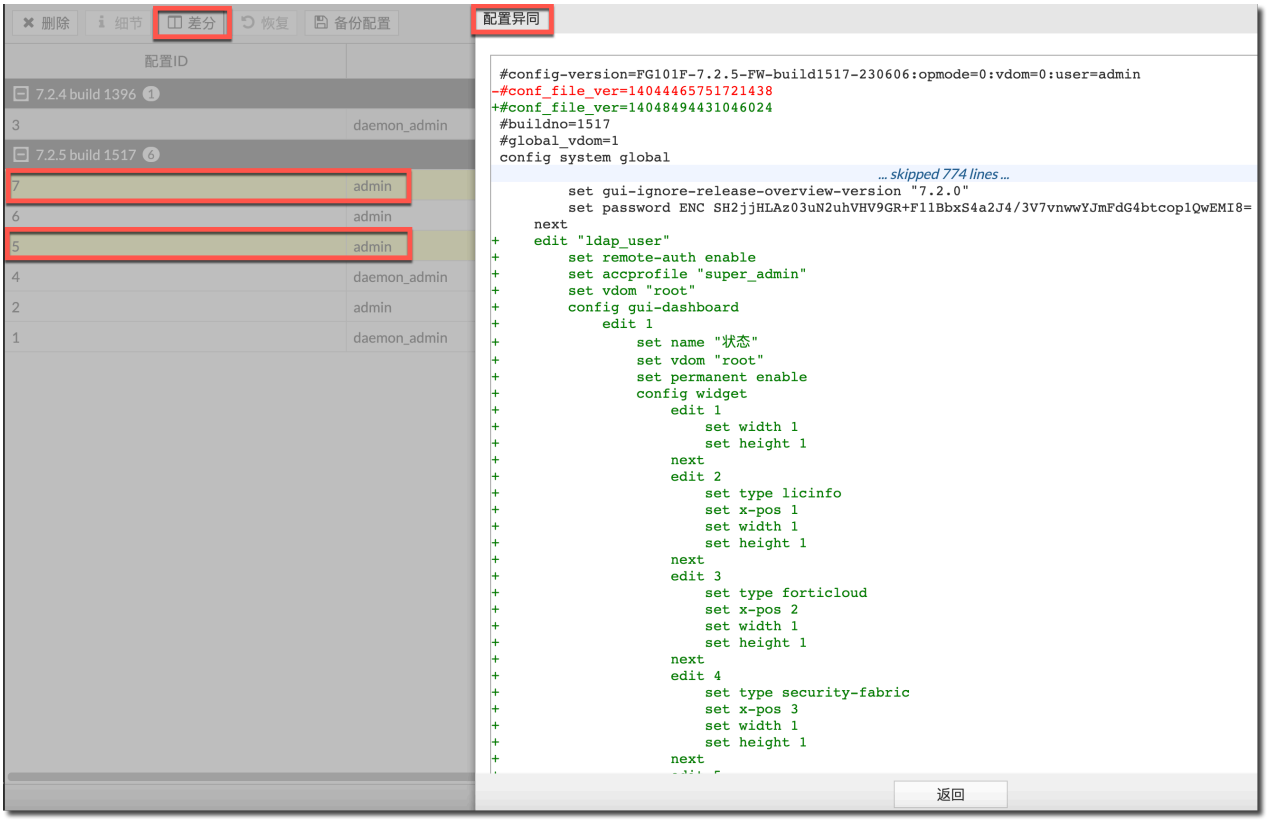



```
FortiGate # execute restore config flash 7
This operation will overwrite the current setting and could possibly reboot the system!
Do you want to continue? (y/n)
```

5. 选中某个配置文件，点击“细节”按钮，可以看到配置文件的信息和具体内容，也可以下载该配置文件。



6. 按住 Ctrl（macOS 为⌘command 键）键使用鼠标，选中两个配置文件，然后点击差分按钮，在弹出的页面中可以看到两个配置文件的差异。



7. 删除备份的配置文件。

```
execute revision delete config <revision>
```

自动备份配置

通过系统自动备份

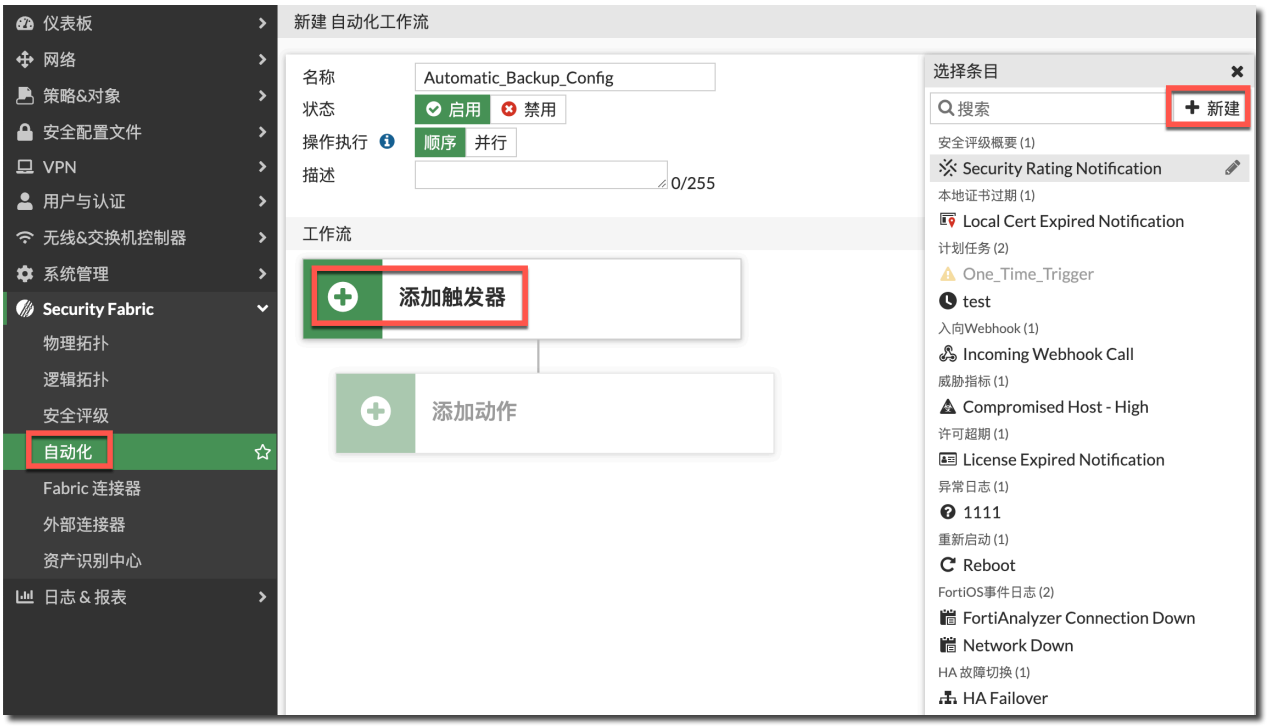
> **重要**
>
> 建议在设备部署前开启这两个全局配置，在误操作导致的设备配置丢失后可以快速恢复历史配置。

```
config system global
    set revision-image-auto-backup enable
    set revision-backup-on-logout enable
end
```

- 在设备固件升级后，自动将配置文件备份到 Flash。
- 在设备管理员登出后，自动将配置文件备份到 Flash，建议在设备部署前开启。

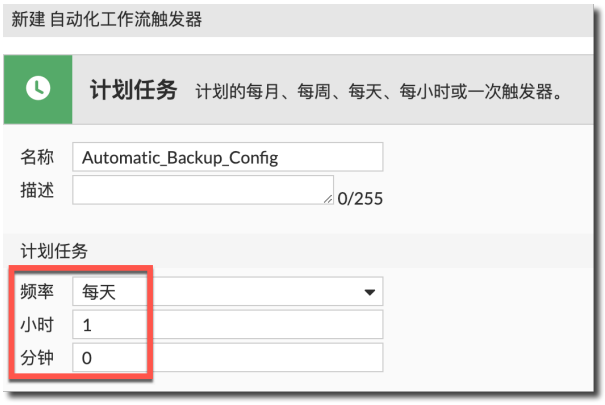
通过自动化功能备份

1. 使用 Security Fabric 中的自动化功能，可以自动定期将系统配置文件备份到 Flash 中，用于配置回溯或恢复。
2. 进入 FortiGate 的 Security Fabric → 自动化 → 工作流页面，新建工作流，配置工作流名称，点击添加触发器，在右侧弹出的条目窗口中点击新建按钮。

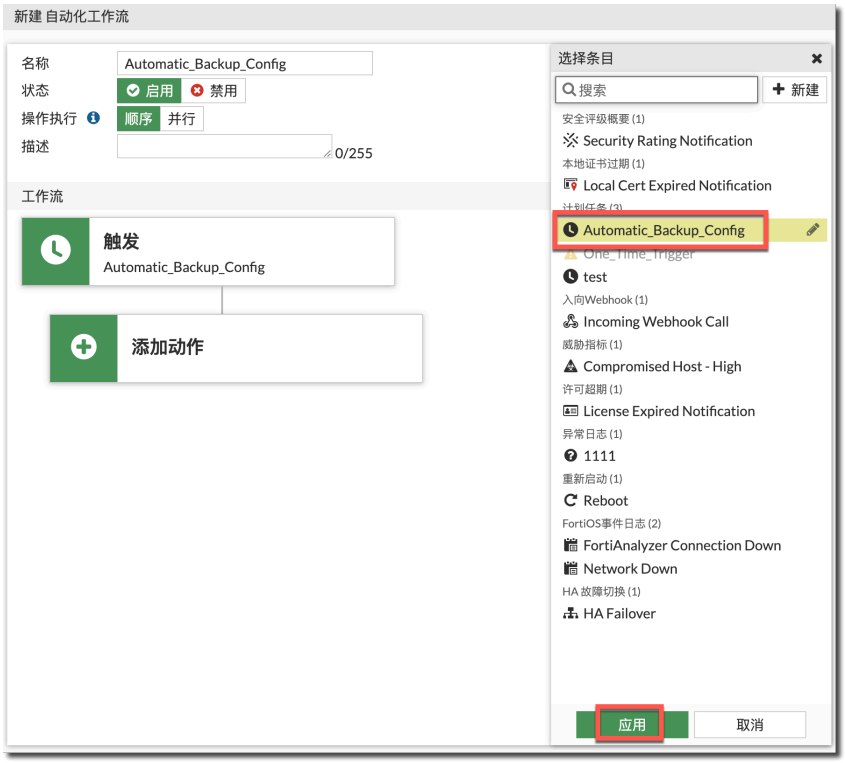


3. 在弹出的新建触发器窗口中选择其他条件中的计划任务触发器，配置触发器的时间为每天凌晨 1 点，点击确定下发配置。

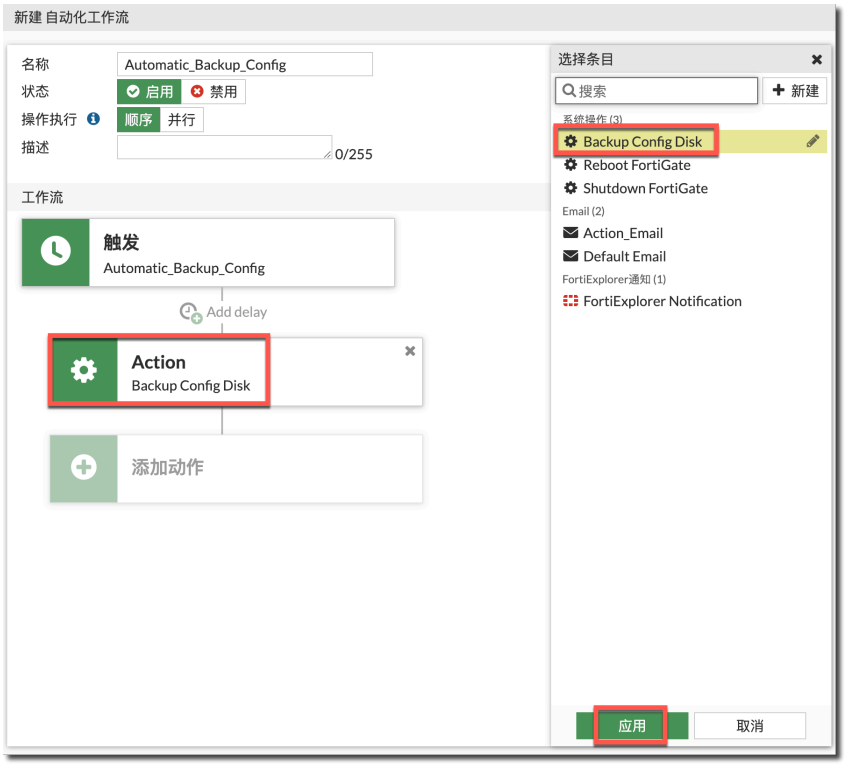




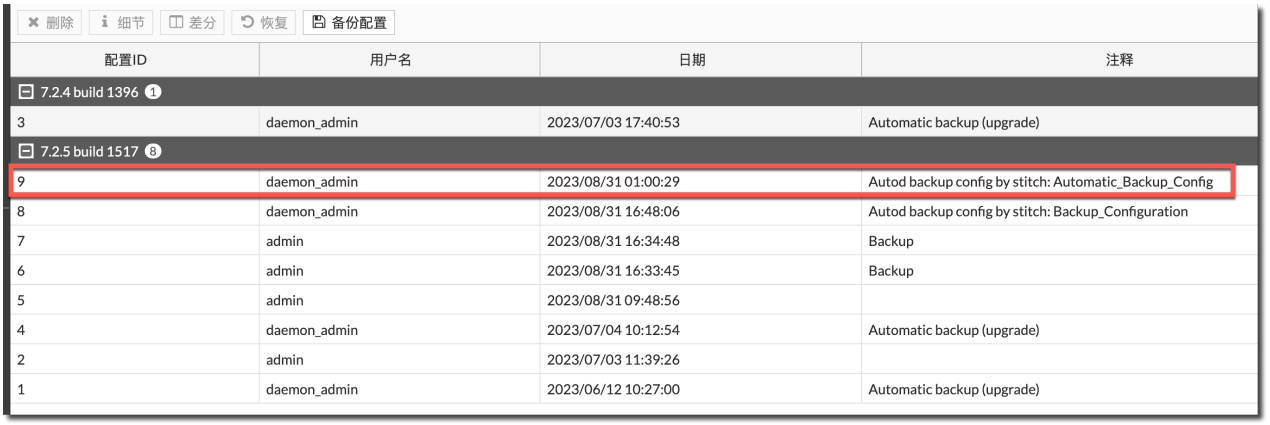
4. 在选择条目中选中上步创建的计划任务，点击下方的应用按钮。



5. 点击 Action 按钮，在右侧弹出的条目中选择系统预置的“Backup Config Disk”，该动作用于将配置文件保存至 Flash 中，点击应用按钮，最后点击确认下发该工作流的配置。



6. 查看自动化备份的配置文件。



删除 细节 差分 恢复 备份配置

配置ID	用户名	日期	注释
7.2.4 build 1396 1			
3	daemon_admin	2023/07/03 17:40:53	Automatic backup (upgrade)
7.2.5 build 1517 8			
9	daemon_admin	2023/08/31 01:00:29	Autod backup config by stitch: Automatic_Backup_Config
8	daemon_admin	2023/08/31 16:48:06	Autod backup config by stitch: Backup_Configuration
7	admin	2023/08/31 16:34:48	Backup
6	admin	2023/08/31 16:33:45	Backup
5	admin	2023/08/31 09:48:56	
4	daemon_admin	2023/07/04 10:12:54	Automatic backup (upgrade)
2	admin	2023/07/03 11:39:26	
1	daemon_admin	2023/06/12 10:27:00	Automatic backup (upgrade)

备份不包含密码的配置文件

重要

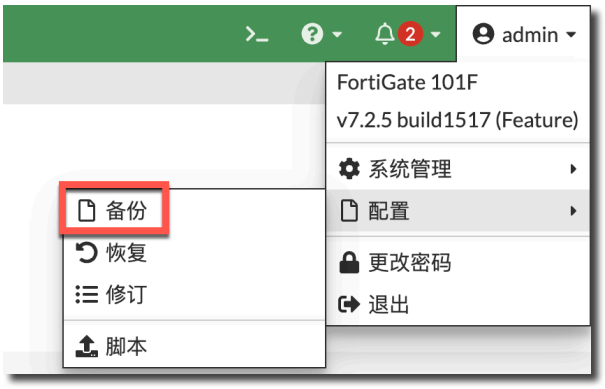
该功能在 7.2.0 及更新版本上支持。

重要

此方式导出的配置文件仅用于维护或支持人员定位问题，请不要用于日常的配置备份及配置恢复！否则会导致业务异常！

导出方法

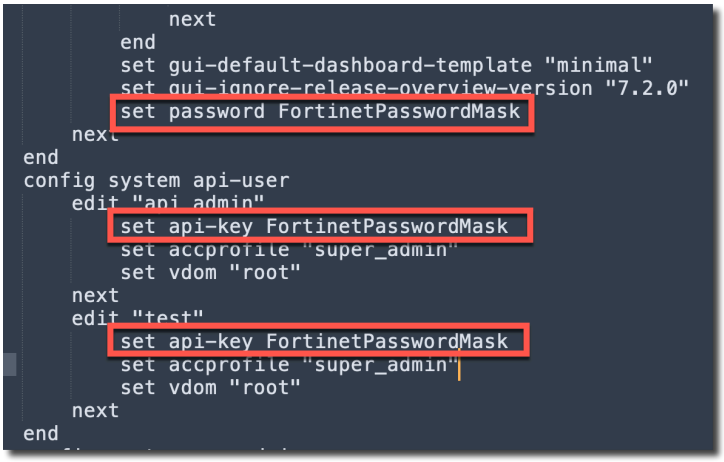
1. 点击页面右上角的管理员 → 配置 → 备份按钮，跳转到备份系统配置页面。



2. 备份系统配置时，选中“密码掩码”，然后点击确认备份配置。



3. 导出的配置中，所有密码字符均被替换为“FortinetPasswordMask”字符（包括管理员密码、用户密码、远程服务器密码、IPSec 预共享密钥、证书密码等）。


next
end
set gui-default-dashboard-template "minimal"
set gui-ignore-release-overview-version "7.2.0"
set password FortinetPasswordMask
next
end
config system api-user
edit "api_admin"
set api-key FortinetPasswordMask
set accprofile "super_admin"
set vdom "root"
next
edit "test"
set api-key FortinetPasswordMask
set accprofile "super_admin"
set vdom "root"
next
end

通过 SCP 备份配置文件

功能简介

在 FortiGate 设备上启用 SCP 管理后，用户可以使用 SCP 协议（基于 SSH）从 FortiGate 设备下载/上传配置或上传固件版本。本文介绍如何通过常见的 SCP 客户端来备份或上传 FortiGate 的配置文件。

SCP 传输分为两种认证模式：

- 用户名/密码认证（每次登录都需要输入密码）
- SSH 密钥认证（生成密钥对并保存后，每次登录不需要输入密码）

以下是通过 SCP 备份配置文件的步骤，用户可自行编写 Python 脚本执行定期配置文件的备份。

FortiGate 开启 SCP 管理

1. 在 FortiGate 需要用于 SCP 传输的接口上开启 SSH 管理协议。

```bash
config system interface
    edit "internal5"
        append allowaccess ssh
    next
end
```

2. 在全局模式下开启 SCP 管理功能，并确认 SSH 管理端口号（默认为 22）。

```bash
config system global
    set admin-scp enable
    set admin-ssh-port 22
end
```

SCP 使用用户名/密码认证

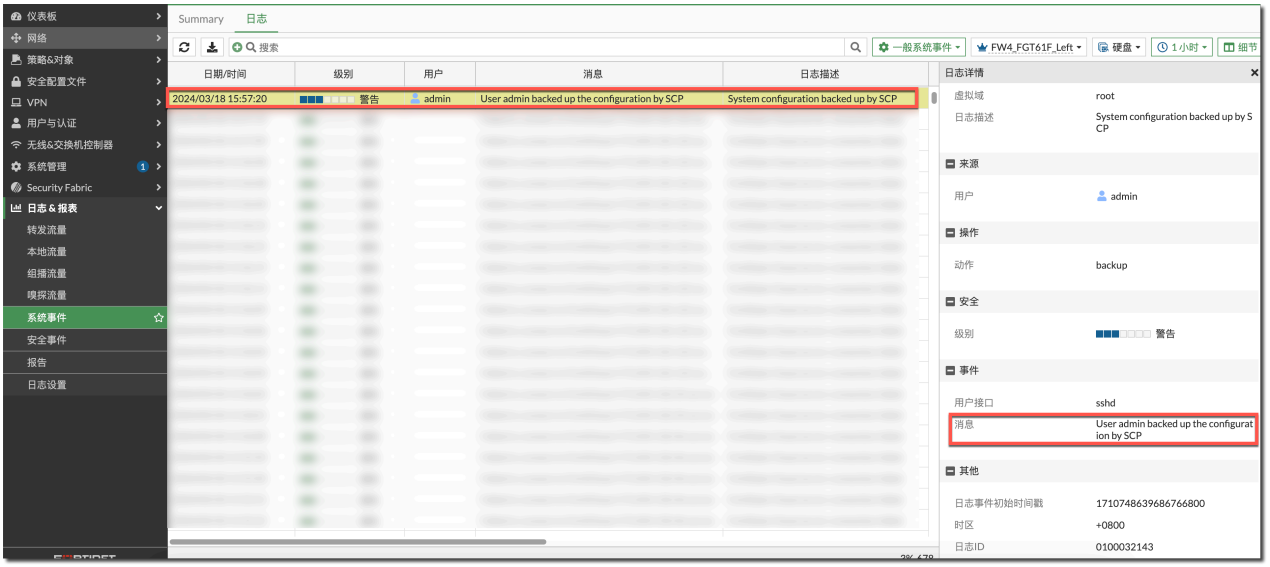
Windows

1. 打开 CMD 终端窗口，执行 `scp admin@<FortiGate_IP>:sys_config <location>` 命令下载配置文件到本地，如下所示，第一次 SCP 访问 FortiGate 时会提示是否信任该主机，输入 yes 后回车。输入 password 为 FortiGate 管理员 admin 的密码（也可使用其他有权限的管理员账户），备份配置成功，配置文件被下载到指定目录。

```bash
scp admin@10.10.12.1:sys_config d:\Downloads\ver
```



2. 在 FortiGate 上查看系统事件日志，可以看到客户端通过 SCP 备份配置文件的日志。




```
date=2024-03-18 time=15:57:20 eventtime=1710748639686766920 tz="+0800" logid="0100032143" type="event" subtype="system" level="warning" vd="root" logdesc="System configuration backed up by SCP" user="admin" ui="sshd" action="backup" msg="User admin backed up the configuration by SCP"
```
bash

重要

上传（恢复）配置文件请使用 `scp <path_to_config_file> admin@<FortiGate_IP>:fgt-restore-config` 命令，例如 `scp d:\Downloads\ver\sys_config admin@10.10.12.1:fgt-restore-config`。

注意上传配置文件后 FortiGate 会立即重启，如果 FortiGate 属于 HA 集群，那么 HA 集群内的所有 FortiGate 都会同时重启并恢复此配置（不包含 HA 不同步的配置）。

macOS/Linux

1. 打开终端窗口，执行 `scp admin@<FortiGate_IP>:sys_config <location>` 命令下载配置文件到本地，如下所示，第一次 SCP 访问 FortiGate 时会提示是否信任该主机，输入 yes 后回车。输入的 password 为管理员 admin 的密码，备份配置成功，配置文件被下载到指定目录。

重要

如果客户端 openSSH 为 9.0 及以上版本，需要在 `scp` 命令后加上 `-O` 选项，`scp -O admin@<FortiGate_IP>:sys_config <location>`。

```
scp -O admin@10.10.12.1:sys_config ~/Downloads/ver
```
bash



重要

上传（恢复）配置文件请使用 `scp <path_to_config_file> admin@<FortiGate_IP>:fgt-restore-config` 命令，例如 `scp ~/Downloads/ver/sys_config admin@10.10.12.1:fgt-restore-config`。

注意上传配置文件后 FortiGate 会立即重启，如果 FortiGate 属于 HA 集群，那么 HA 集群内的所有 FortiGate 都会同时重启并恢复此配置。

2. 在 FortiGate 上查看系统事件日志，可以看到客户端通过 SCP 备份配置文件的日志（略，参考 Windows SCP 备份配置文件 FortiGate 产生的日志）。

SCP 使 SSH 密钥认证

Windows

1. 打开 CMD 终端窗口，在默认路径下（一般为 `C:\Users\<username>`，如此例中的路径为 `C:\Users\cyndi`），使用 `ssh-keygen -t ed25519` 命令，生成 SSH 密钥对，推荐使用 ed25519 算法。

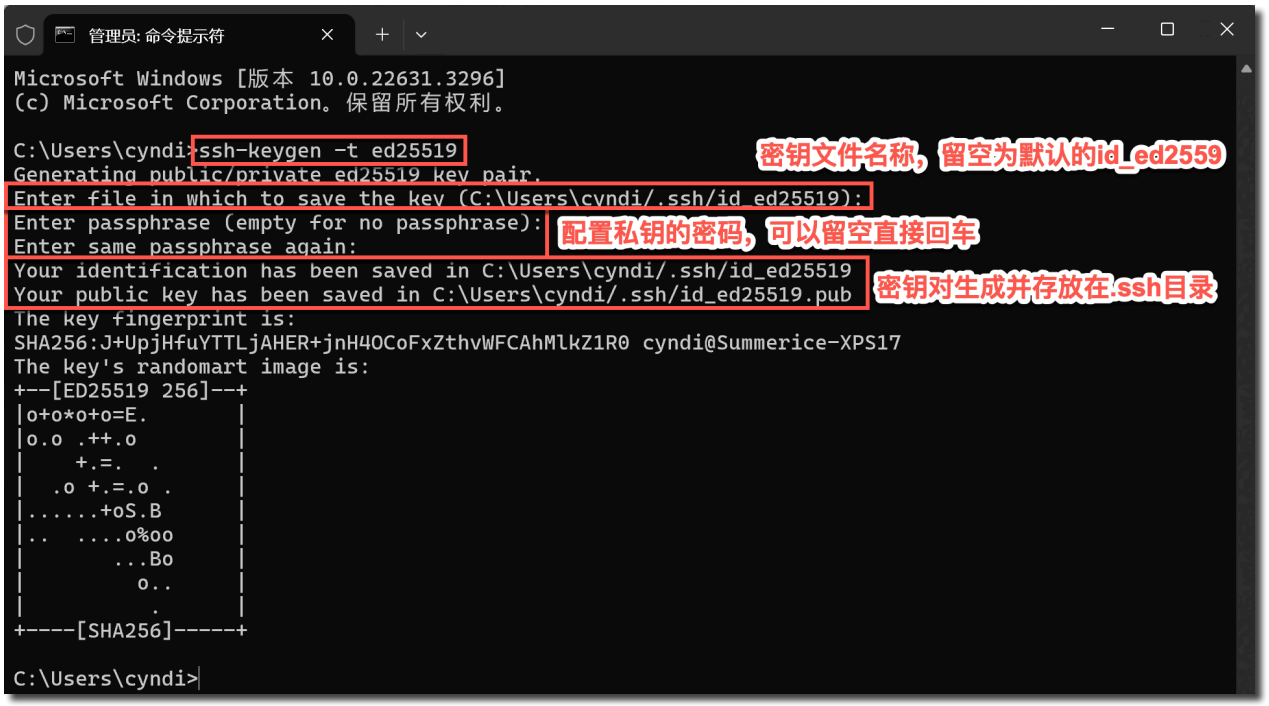
重要

`-t` 后跟随的 `ed25519` 表示 SSH 密钥算法，在 SCP 命令中可以使用 dsa、ecdsa、ecdsa-sk、ed25519、ed25519-sk、rsa。

FortiGate 使用 ed25519 和 rsa 作为默认的 SSH 密钥算法，在 FortiOS 7.4.2 之前，SSH 密钥算法是无法更改的。

在 FortiOS 7.4.2 及更新版本中，可以自定义 SSH 密钥算法：

```bash
FortiGate # config system global
FortiGate (global) # set ssh-hostkey-algo ?
ecdsa-sha2-nistp521    ecdsa-sha2-nistp521
ecdsa-sha2-nistp384    ecdsa-sha2-nistp384
ecdsa-sha2-nistp256    ecdsa-sha2-nistp256
rsa-sha2-256           rsa-sha2-256
rsa-sha2-512           rsa-sha2-512
ssh-ed25519            ssh-ed25519
```



2. 如下所示，客户端在 CMD 下的默认目录（`C:\Users\cyndi`）生成了 SSH 密钥对的文件夹，该密钥对所属文件夹的名称为 `.ssh`，由于在生成时没有自定义密钥文件名称，密钥对文件的名称为默认名称。



3. 使用记事本打开生成的公钥文件（后缀为 pub，此例中为 `C:\Users\cyndi\.ssh\id_ed25519.pub`），如下所示。

```bash
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIABtVpehNaFSWbOP13OaZKyxFsvAlSW/jtFidqa7l0×D cyndi@Summerice-XPS17
```

> **重要**
>
> 可以看到公钥文件 id_ed25519.pub 由三部分组成，需要收集的为公钥文件中的前两部分：
>
> - `ssh-ed25519`：SSH 使用的密钥算法。
> - `AAAAC3NzaC1lZDI1NTE5AAAAIABtVpehNaFSWbOP13OaZKyxFsvAlSW/jtFidqa7l0×D`：公钥文本。
> - `cyndi@Summerice-XPS17`：用户邮箱，系统根据系统用户名与主机名自动生成，也可在生成密钥对时使用 `-C` 参数手工指定。

4. 在 FortiGate 的 CLI 中，编辑管理员 admin 的配置，设置 `ssh-public-key1` 的内容为公钥文件中的前两部分，即 SSH 使用的密钥算法和公钥文本，中间以空格隔开，前后要加上英文的双引号 `""`，如下所示，`"ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIABtVpehNaFSWbOP13OaZKyxFsvAlSW/jtFidqa7l0×D"`。

> **重要**
>
> 最多可设置 3 个 `ssh-public-key`），`ssh-public-key` 的格式输入正确时，会提示 `SSH key is good.`。

```bash
FortiGate # config sys admin
FortiGate (admin) # edit admin
FortiGate (admin) # set ssh-public-key1 "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIABtVpehNaFSWbOP13OaZKyxFsvAlSW/jtFidqa7l0×D"
SSH key is good.
FortiGate (admin) # end
FortiGate #
```

5. 在 Windows 下打开 CMD 终端窗口，执行 `scp admin@<FortiGate_IP>:sys_config <location>` 命令下载配置文件到本地，如下所示，第一次 SCP 访问 FortiGate 时会提示是否信任该主机，输入 yes 后回车。不需要输入管理员密码，备份配置成功，配置文件被下载到指定目录。

> **重要**
>
> 如果 `.ssh` 文件夹下有多个 SSH 公钥文件，也可使用 `-i <pubkey_location>` 参数指定某个公钥文件进行 SCP 协商。

```bash
scp admin@10.10.12.1:sys_config d:\Downloads\ver
```



重要

上传（恢复）配置文件请使用 `scp <path_to_config_file> admin@<FortiGate_IP>:fgt-restore-config` 命令，例如 `scp d:\Downloads\ver\sys_config admin@10.10.12.1:fgt-restore-config`。

注意上传配置文件后 FortiGate 会立即重启，如果 FortiGate 属于 HA 集群，那么 HA 集群内的所有 FortiGate 都会同时重启并恢复此配置（不包含 HA 不同步的配置）。

6. 在 FortiGate 上查看系统事件日志，可以看到客户端通过 SCP 备份配置文件的日志（略，参考 Windows SCP 使用管理员账户密码方式备份配置文件 FortiGate 产生的日志）。

macOS/Linux

1. 以 macOS 为例，打开终端窗口，在默认路径下（一般为 `/Users/<username>`，如此例中的路径为 `/Users/summerice`），使用 `ssh-keygen -t ed25519` 命令，生成 SSH 密钥对，推荐使用 ed25519 算法。

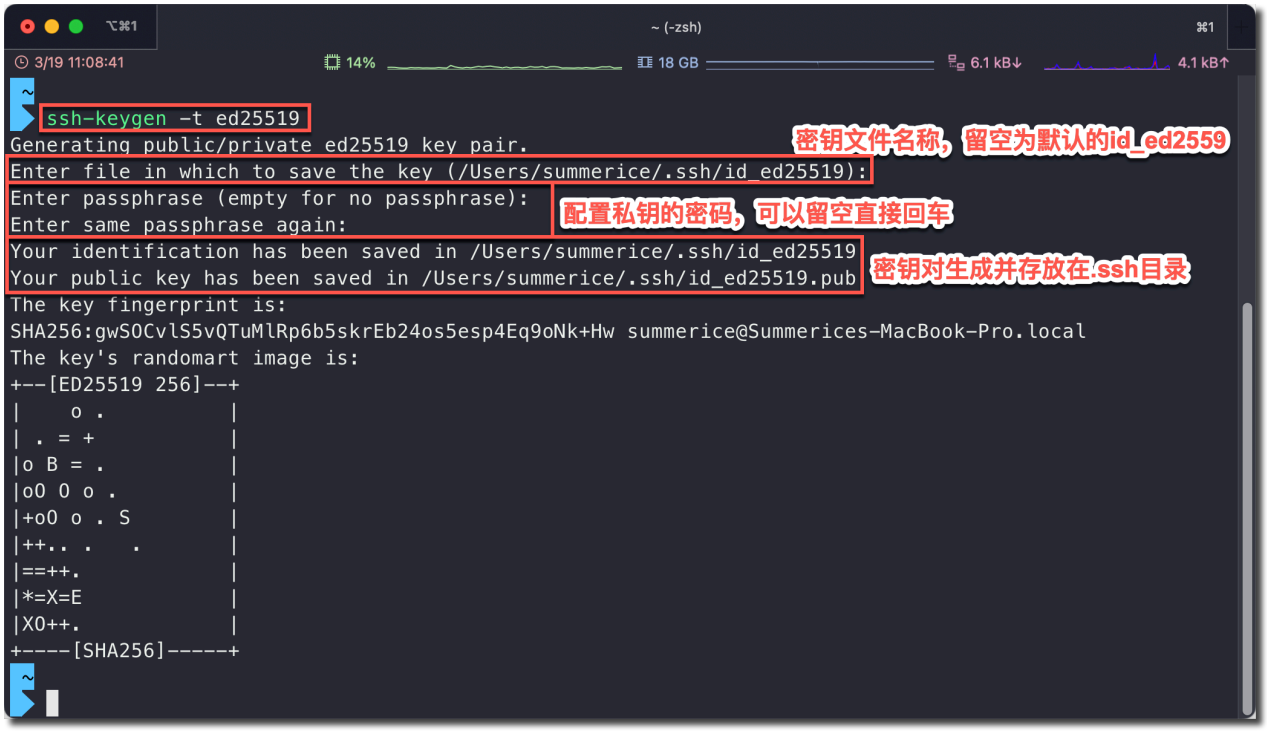
重要

`-t` 后跟随的 `ed25519` 表示 SSH 密钥算法，在 SCP 命令中可以使用 dsa、ecdsa、ecdsa-sk、ed25519、ed25519-sk、rsa。

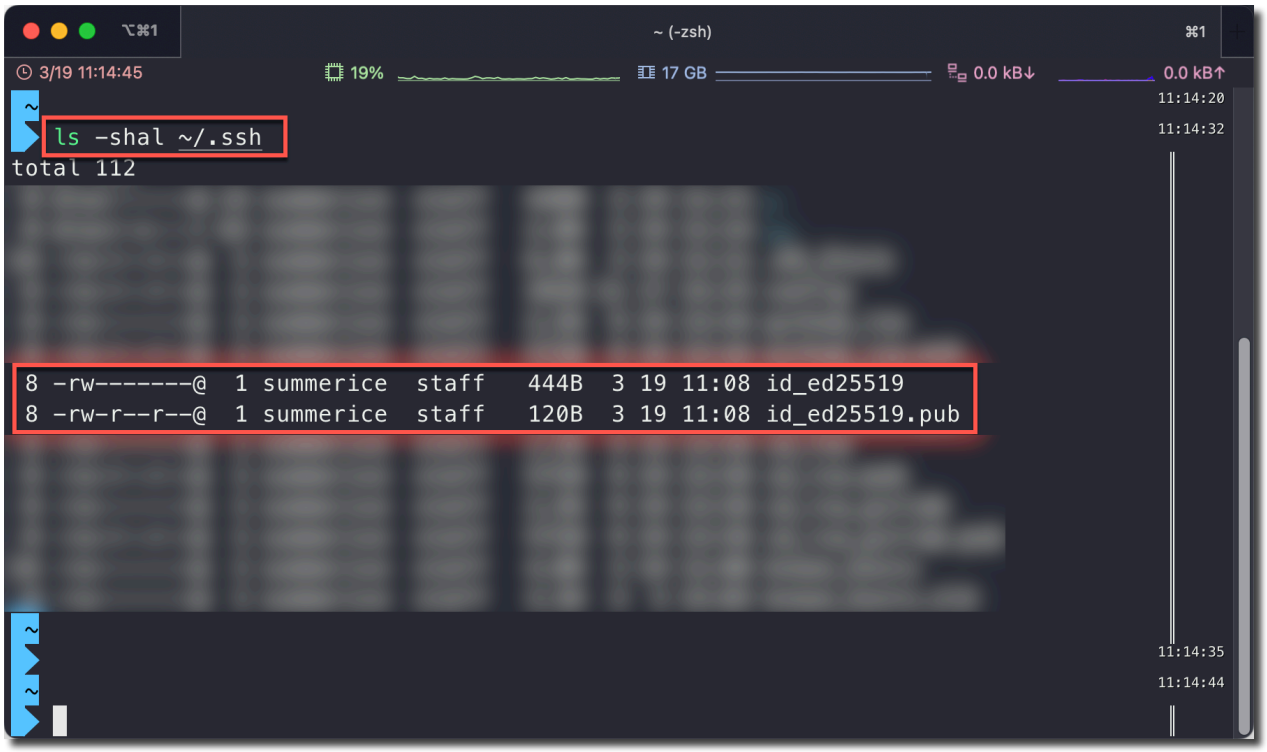
FortiGate 使用 ed25519 和 rsa 作为默认的 SSH 密钥算法，在 FortiOS 7.4.2 之前，SSH 密钥算法是无法更改的。

在 FortiOS 7.4.2 及更新版本中，可以自定义 SSH 密钥算法：

```bash
FortiGate # config system global
FortiGate (global) # set ssh-hostkey-algo ?
ecdsa-sha2-nistp521    ecdsa-sha2-nistp521
ecdsa-sha2-nistp384    ecdsa-sha2-nistp384
ecdsa-sha2-nistp256    ecdsa-sha2-nistp256
rsa-sha2-256           rsa-sha2-256
rsa-sha2-512           rsa-sha2-512
ssh-ed25519            ssh-ed25519
```



2. 如下所示，客户端在终端下的默认目录（`/Users/summerice`，在 macOS 或 Linux 下也可以用 `~` 表示该目录）生成了 SSH 密钥对的文件夹，该密钥对所属文件夹的名称为 `.ssh`，由于在生成时没有自定义密钥文件名称，密钥对文件的名称为默认名称。



3. 使用 `cat` 命令查看生成的公钥文件（后缀为 pub，此例中为 `~/.ssh/id_ed25519.pub`）的内容，如下所示。

```bash
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGGp5Pv2JGgGjmV+jOeSPwEbJsMAiz7tLJDsZQpCAvwg summerice@Summerices-MacBook-Pro.local
```

重要

可以看到公钥文件 id_ed25519.pub 由三部分组成，需要收集的为公钥文件中的前两部分：

- `ssh-ed25519`：SSH 使用的密钥算法。
- `AAAAC3NzaC1lZDI1NTE5AAAAIGGp5Pv2JGgGjmV+jOeSPwEbJsMAiz7tLJDsZQpCAvwg`：公钥文本。
- `summerice@Summerices-MacBook-Pro.local`：用户邮箱，系统根据系统用户名与主机名自动生成，也可在生成密钥对时使用 `-C` 参数手工指定。

4. 在 FortiGate 的 CLI 中，编辑管理员 admin 的配置，设置 `ssh-public-key2` 的内容为公钥文件中的前两部分，即 SSH 使用的密钥算法和公钥文本，中间以空格隔开，前后要加上英文的双引号 `""`，如下所示，`"ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGGp5Pv2JGgGjmV+jOeSPwEbJsMAiz7tLJDsZQpCAvwg"`。

> **重要**
>
> 最多可设置 3 个 `ssh-public-key`，`ssh-public-key` 的格式输入正确时，会提示 `SSH key is good.`。

```
FortiGate # config sys admin
FortiGate (admin) # edit admin
FortiGate (admin) # set ssh-public-key2 "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGGp5Pv2JGgGjmV+jOeSPwEbJsMAiz7tLJDsZQpCAvwg"
SSH key is good.
FortiGate (admin) # end
FortiGate #
```

5. 打开 macOS 的终端窗口，执行 `scp admin@<FortiGate_IP>:sys_config <location>` 命令下载配置文件到本地，如下所示，第一次 SCP 访问 FortiGate 时会提示是否信任该主机，输入 yes 后回车。不需要输入管理员密码，备份配置成功，配置文件被下载到指定目录。

> **重要**
>
> 如果客户端 openSSH 为 9.0 及以上版本，需要在 `scp` 命令后加上 `-O` 选项，`scp -O admin@<FortiGate_IP>:sys_config <location>`。

```
scp -O admin@10.10.12.1:sys_config ~/Downloads/ver
```
bash



> **重要**
>
> 上传（恢复）配置文件请使用 `sudo scp -O <path_to_config_file> admin@<FortiGate_IP>:fgt-restore-config` 命令，例如 `sudo scp -O ~/Downloads/ver/sys_config admin@10.10.12.1:fgt-restore-config`。
>
> 注意上传配置文件后 FortiGate 会立即重启，如果 FortiGate 属于 HA 集群，那么 HA 集群内的所有 FortiGate 都会同时重启并恢复此配置。

6. 在 FortiGate 上查看系统事件日志，可以看到客户端通过 SCP 备份配置文件的日志（略，参考 Windows SCP 使用管理员账户密码方式备份配置文件 FortiGate 产生的日志）。

恢复出厂

需求

当需要把设备的所有配置删除，可进行恢复出厂操作。恢复出厂后所有配置都将丢失，若一定要恢复出厂设置建议前先备份当前的配置；备份配置操作步骤请参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗。

> **重要**
>
> 设备的授权信息都在云端，恢复出厂配置后保证设备联网就可重新获取授权信息。

配置要点

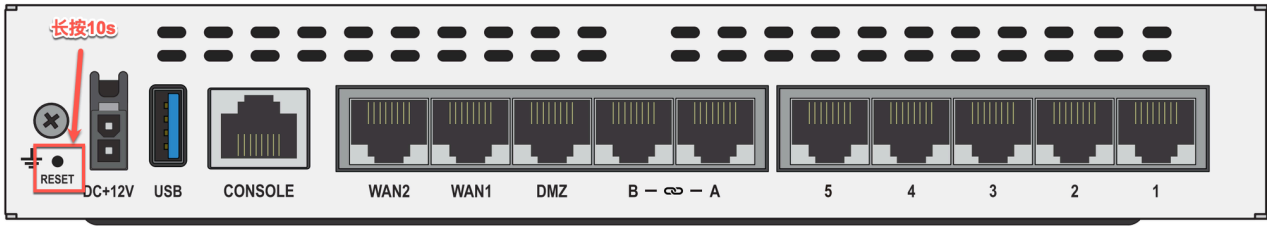
1. 恢复出厂设置会清空所有配置，系统会自动重启。
2. 恢复出厂设置后，internal 口或则 MGMT 口地址恢复为 192.168.1.99。

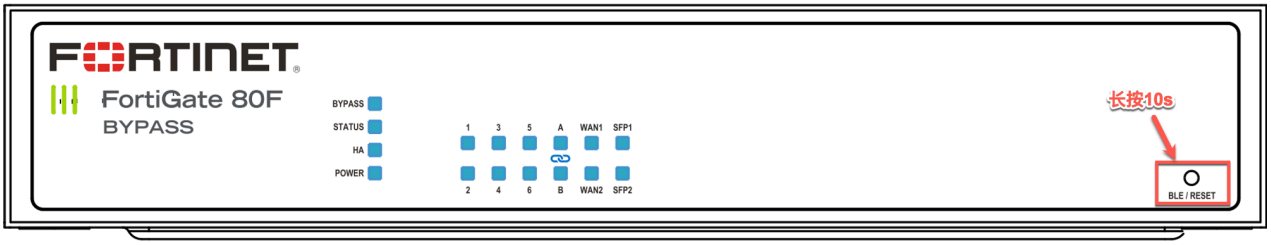
操作步骤

1. 第一种方式：命令行方式：进入命令行，执行 execute factoryrest 命令，回车后会提示此操作将会恢复出厂配置，是否继续，输入字母 "y" 即可。

```
FortiGate # execute  factoryreset
This operation will reset the system to factory default!
Do you want to continue? (y/n) y
```

2. 第二种方式：通过设备上 reset 复位键（仅部分型号等有 reset 键，其他型号无法执行此操作）：reset 键操作防火墙系统正常启动后（可以 ping 通或登录管理页面）30S 内，长按 10S 左右，系统将自动重启，届时将恢复为出厂设置。如下所示 60F 和 80F 的重置按钮位置。





验证效果

恢复出厂设置后，管理口地址恢复为 192.168.1.99，可通过此地址登陆 https://192.168.1.99 ↗ 进行管理，用户名和密码分别恢复默认的 admin 和空。

注意事项

恢复出厂不清空硬盘日志，恢复出厂仅针对配置文件而言。

配置回退

网络需求

当网络管理员通过远程方式变更 FortiGate 配置时，错误的配置可能会造成访问丢失。这种情况下，需要在配置错误并失去访问连接的情况下，自动回退 FortiGate 配置到错误配置下发前的状态。

配置步骤

> **重要**
>
> 强烈建议在修改设备关键配置（特别是与设备连通性相关的配置时），避免通过远程的方式进行操作。如果必须通过远程的方式进行操作，才考虑使用此方法。

1. 在进行远程配置变更前，将 FortiGate 的配置保存方式设置为 `revert`，`cfg-revert-timeout` 默认值为 600s。

   > **重要**
   >
   > 在配置变更结束后，请将配置保存模式恢复为“自动”。


   ```bash
   config system global
       set cfg-save revert
       set cfg-revert-timeout 600
   end
   ```

   配置保存模式有以下 3 种：

   - `automatic`（默认）：配置更改会实时下发到 Memory 和 Flash。
   - `manual`：配置更改只会下发到 Memory，不下发到 Flash，必须手动下发到 Flash 中（通过 GUI 或 `exe cfg save`），未手动保存到 Flash 的配置将会在重启后丢失。修改配置后，会显示有配置未保存的提示，点击提示可以查看具体修改的配置内容。
   - `revert`：配置更改只会下发到 Memory，不下发到 Flash。当配置更改后，如果在 cfg-revert-timeout 时间内未手动保存配置（通过 GUI 或 `exe cfg save`），设备会 Reload 并恢复 Flash 中的原有配置文件（重启前 GUI 会提示相关告警信息提示手动保存配置）。配置文件 Reload 期间，设备业务会产生中断，所以在配置变更结束后，请将配置保存模式恢复为“自动”。

2. 修改设备配置，例如修改接口 IP，查看修改接口前的 IP 地址如下。

   ```bash
   config system interface
       edit "wan1"
           set ip 3.3.3.3 255.255.255.0
       next
   end
   ```

3. 修改该接口 IP 到其他 IP 地址，此时配置会下发到内存中并生效，而不会下发到 Flash 中。

   ```bash
   config system interface
       edit "wan1"
           set ip 4.3.3.3 255.255.255.0
       next
   ```

```
end
```

4. 如果 GUI 仍然可以访问，GUI 页面右上角会提示配置已变更，若未在 600s 内手动点击“保存阶段性更改”（或 CLI 执行 `exe cfg save`）将配置保存到 Flash 中，设备会 Reload 并恢复 Flash 中的原有配置。



5. 假如步骤 3 为误操作，导致 FortiGate 无法远程管理，等待 600s 后，FortiGate 会重载（Reload，非重启设备）Flash 中的配置文件。可以看到 Reload 之后，接口 IP 恢复到修改之前的 IP 地址。

> **重要**
>
> - Reload 期间，设备业务会产生中断，所以在配置变更结束后，请将配置保存模式恢复为“自动”。
> - 在 HA 环境下，所有 HA 成员都会同时执行 Reload。

```bash
config system interface
    edit "wan1"
        set ip 3.3.3.3 255.255.255.0
    next
end
```

6. 设备 Reload 期间的 Console 记录。

```bash
FortiGate #
System will reload in the next 10 seconds ...
System will reload in the next 9 seconds ...
System will reload in the next 8 seconds ...
System will reload in the next 6 seconds ...
System will reload in the next 5 seconds ...
System will reload in the next 4 seconds ...
System will reload in the next 3 seconds ...
System will reload in the next 1 seconds ...
......
FortiGate login:
```

管理员配置

管理员设置

远程认证管理员

远程认证与本地认证的优先级

管理员登录锁定

管理员密码策略

通过 FortiCloud 账户登录 FortiGate

限制管理员 SSH/Telnet 跳板

密码恢复

API 用户

- API 文档获取
- API 用户配置
- API 常见问题

  API 认证失败

管理员设置

功能需求

设备出厂时设备内置的默认账号为 admin（拥有所有权限），默认密码：空，现在需求如下：

1. 修改管理员 admin 的密码为 fgt@123, 设置管理员 admin 账号的管理主机 IP 为 172.18.10.108/32，即只有 172.18.10.108 这台主机可以用 admin 账号管理设备。
2. 添加管理员账号 monitor，配置密码为 123456a!，权限设置为 test，管理主机 IP 不限，即所有主机都能用此账号登陆设备管理，权限为只读。
3. 定义密码规则，对密码的复杂度进行限制。
4. 设置 Web 页面超时时间：若设置为 90 分钟，则 90 分钟内管理员未做任何操作，则自动退出管理页面。

配置要点

1. 修改管理员密码，管理 IP。
2. 添加访问内容表 readonly。
3. 添加管理员 monitor。
4. 定义密码规则，修改设备管理参数。
5. 设置管理登录超时时间。

配置步骤

修改 admin 管理员密码/管理 IP

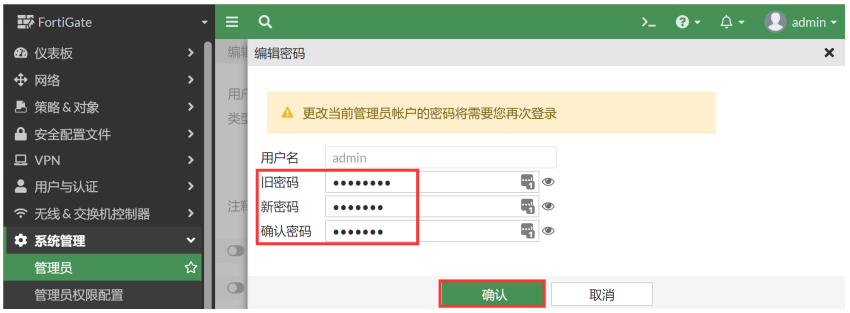
1. 进入系统管理 → 管理员页面。



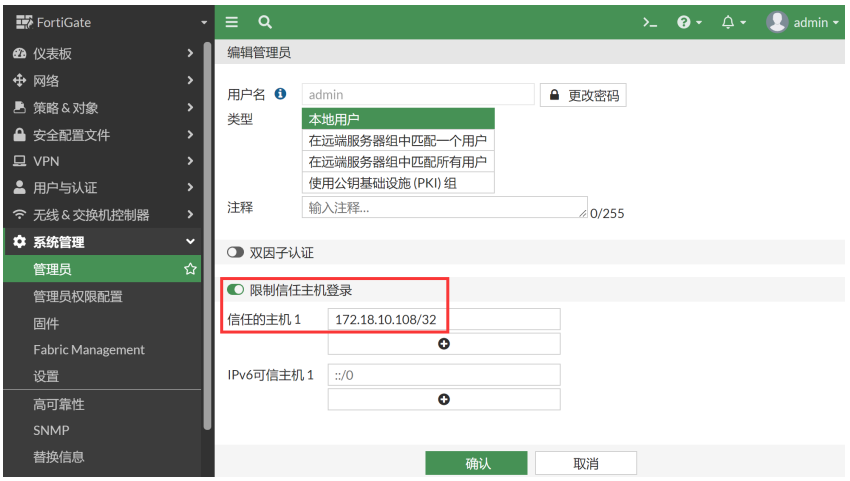
2. 编辑用户名 "admin"（双击或者点击编辑按钮）点击 "更改密码"。



3. 在随后出现的修改密码弹出窗口内，输入新密码 fgt@123，点击确定。



4. 启用 "限制信任主机登录" 在 "信任的主机 1" 处填写管理 IP：172.18.10.108/32，点击 "确定"。

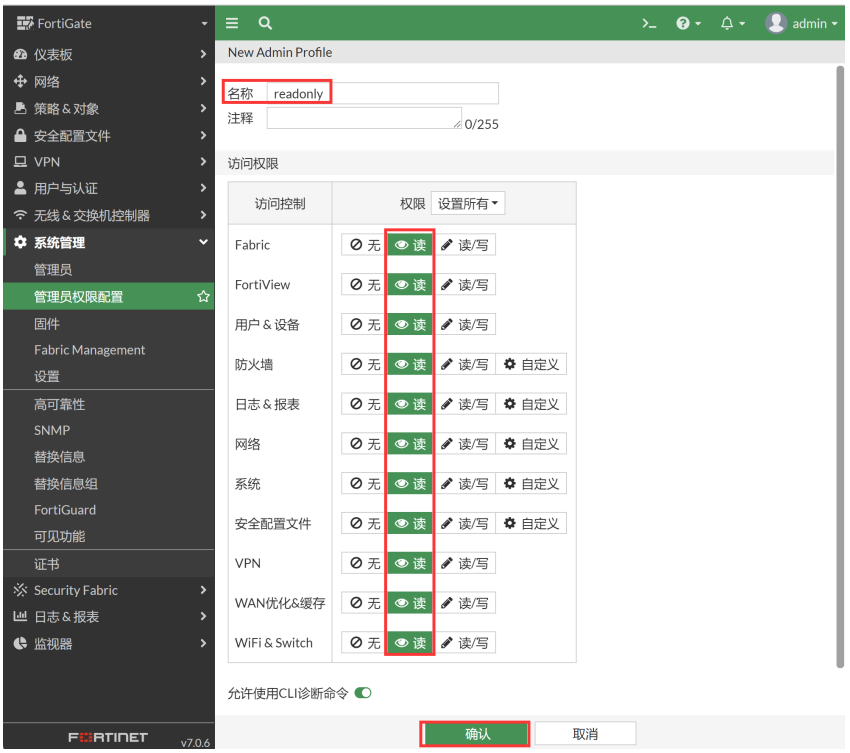


添加访问内容表 readonly

1. 进入系统管理 → 管理员权限配置 → 新建。



2. 输入名称：readonly，所有勾选只读权限，点击 "确认"。

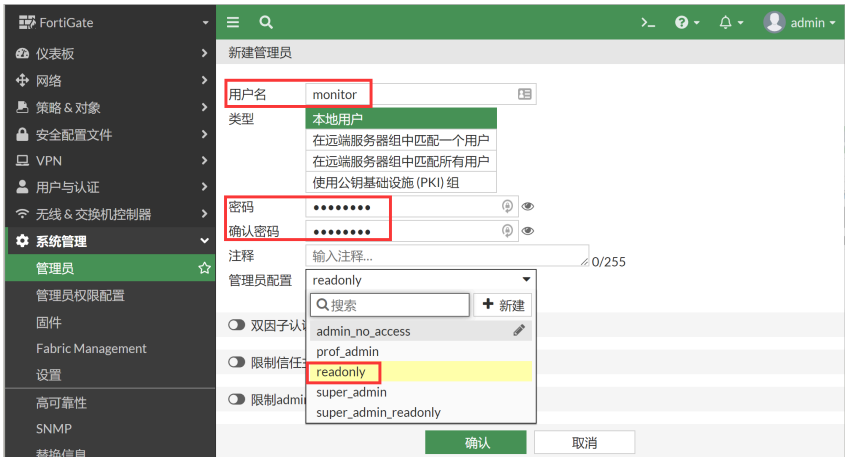


添加管理员 monitor

1. 进入系统管理 → 管理员，点击新建 → 管理员。



2. 添加管理员 monitor，密码：123456a!，权限设置为 readonly，管理主机 IP 不限制，如下图所示。



登录超时时间

1. 默认配置下，管理员在登录 FortiGate（GUI/SSH/Console 等）5 分钟内如果没有任何操作，系统会自动登出该管理员的登录状态。

2. 可以修改管理员的登录超时时间范围为 1~480min，如下所示修改管理员的空闲时间为 60 分钟。




```
config system global
    set admintimeout 60
end
```

结果验证

使用 monitor 账号登陆设备，修改配置，提示没有权限错误。



远程认证管理员

除了本地配置的管理员，FortiGate 还支持远程认证的管理员登录，如 LDAP、Radius、Tacacs + 等。

远程认证服务器配置

LDAP 服务器配置请参考：用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗。

Radius 服务器配置请参考：用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗。

远程认证管理员配置

1. 这里以 LDAP 用户为例，参考上述配置步骤，已经配置了 LDAP 远程服务器和远程用户组。

```bash
config user ldap
    edit "LDAP"
        set server "192.168.100.50"
        set cnid "cn"
        set dn "dc=summerice2019,dc=com"
        set type regular
        set username "administrator"
        set password xxxxxxxx
    next
end
config user group
    edit "LDAP_Users"
        set member "LDAP"
        config match
            edit 1
                set server-name "LDAP"
                set group-name "CN=ad_group1,DC=summerice2019,DC=com"
            next
            edit 2
                set server-name "LDAP"
                set group-name "CN=Users,CN=Builtin,DC=summerice2019,DC=com"
            next
        end
    next
end
```

2. 进入 FortiGate 的“系统管理 → 管理员”页面，新建管理员。



- 类型：
  - 用户名：管理员用户名，解释如下。
  - 类型：

- 在远端服务器组中匹配一个用户：使用配置的用户名匹配远程服务器中的用户，管理员配置的用户名必须属于远程认证服务器中的一个用户，用户登陆时也必须使用这个用户名。
- 在远端服务器组中匹配所有用户：使用管理员登录时输入的用户名匹配远程服务器中的用户，管理员配置的用户名不必属于远程认证服务器中的任何用户（可随意配置），用户登陆时使用 LDAP 服务器上的用户名输入。

- 备份密码：当远程认证服务器不可用时，可以使用备份密码作为逃生密码（使用“在远端服务器组中匹配所有用户”的类型时，无法配置备份密码）。
- 管理员配置：该管理员所拥有的管理员权限。
- 远端用户组：选择上步中创建的 LDAP 用户组。

3. 以上配置对应的 CLI 如下。

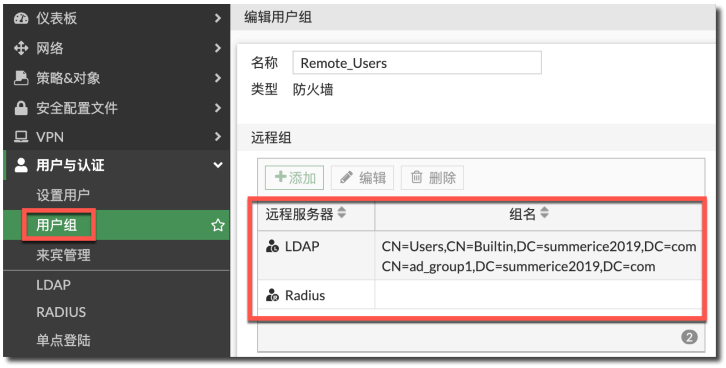
```
config system admin
    edit "ldap_user"
        set remote-auth enable
        set accprofile "super_admin"
        set vdom "root"
        set remote-group "LDAP_Users"
        set password ENC SH2Cn+7WDWNOQFNb12zdDAuL/XVDN6ga4qxCtdB9lEV8FwRkSjPQixI/Rm07s8=
    next
end
```

4. 如果使用“在远端服务器组中匹配所有用户”类型，对应的 CLI 如下。

```
config system admin
    edit "ldap_user"
        set remote-auth enable
        set accprofile "super_admin"
        set vdom "root"
        set wildcard enable
        set remote-group "LDAP_Users"
    next
end
```

注意事项

1. 如果 FortiGate 的认证用户组中配置了多个远程认证服务器，如下包含了 LDAP 和 Radius 两个服务器。




```
config user group
    edit "Remote_Users"
        set member "LDAP" "Radius"
        config match
            edit 1
                set server-name "LDAP"
                set group-name "CN=Users,CN=Builtin,DC=summerice2019,DC=com"
            next
            edit 2
                set server-name "LDAP"
                set group-name "CN=ad_group1,DC=summerice2019,DC=com"
            next
        end
    next
end
```

2. 那么 FortiGate 会先到哪个认证服务器去认证呢?

3. 对于 FortiGate 来说，两个认证服务器没有优先顺序，认证报文会同时发送至 LDAP 与 Radius，哪个服务器先返回了认证结果，FortiGate 就采用哪个服务器返回的认证结果。

4. 排除网络延迟的差异，由于 Radius 认证采用的是 UDP 协议，通常会比 LDAP 更快的返回认证结果，从而导致 FortiGate 优先采用 Radius 的认证结果，但这并不代表 Radius 认证优先于 LDAP 认证。

远程认证与本地认证的优先级

配置方法

远程认证与本地认证的优先级由以下命令决定，默认为 disable 状态。

```bash
config system global
    set admin-restrict-local ?    //Enable/disable local admin authentication restriction when remote authenticator is up
and running (default = disable).
    enable     Enable local admin authentication restriction.
    disable    Disable local admin authentication restriction.
end
```

默认情况（disable）

```bash
config system global
    set admin-restrict-local diable
end
```

- 如果远程和本地有相同的用户名，不同的密码，则本地认证有效。
- 如果用户名/密码都一样，也是本地认证有效。
- 使用 `diagnose debug application fnbamd -1` 可以观察到，认证直接在本地完成，不会送到远程服务器上。

修改为 enable

```bash
config system global
    set admin-restrict-local enable
end
```

- 当任何一个远程认证服务器可达时，远程认证服务器生效，本地认证无效。
- 只有当 FortiGate 到远程认证服务器不可达时，本地认证才有效。
- 如果远程服务器和本地有相同的用户名，认证会先到远程服务器上，但是登录会直接显示失败（无论使用远程用户还是本地用户的密码，无论远程用户和本地用户的密码是否相同）。

> **重要**
>
> 在 7.6.0 版本之前，Console 可以使用本地账户登录，不受此功能开启的限制。

多个远程认证服务器的情况（set admin-restrict-local enable）

7.2.0 之前

当 FortiGate 配置了多个远程认证服务器，只配置了一个远程认证服务器与管理员关联，举例：FortiGate 同时配置了 LDAP 与 Radius 服务器，但只有 LDAP 服务器与管理员配置了关联，Radius 服务器没有与任何管理员配置关联：

- 当任何一个远程认证服务器可达时，远程认证服务器生效，本地认证无效。
- 只有当 FortiGate 到所有的远程认证服务器（上述举例为 LDAP、Radius）不可达时，本地认证才有效，即使 Radius 认证服务器没有关联任何管理员。

7.2.0 及之后

当 FortiGate 配置了多个远程认证服务器，只配置了一个远程认证服务器与管理员关联，举例：FortiGate 同时配置了 LDAP 与 Radius 服务器，但只有 LDAP 服务器的用户组与管理员配置了关联，Radius 服务器没有与任何管理员配置关联：

- 只要防火墙管理认证调用的远程认证服务器不可达，本地用户认证就可以生效。

7.6.0 新增选项

提示

- 在 7.6.0 版本之前，Console 使用本地账户登录，不受 `admin-restrict-local` 开启状态的影响。
- 在 7.6.0 及之后版本，`admin-restrict-local` 针对 Console 登录添加了新的选项。将原来的 enable 选项拆成 `all` 和 `non-console-only`。

```
config system global
    set admin-restrict-local {all | non-console-only | disable}
end
```

- `non-console-only`：行为与 7.6.0 之前的 `enable` 一致，Console 登录不受 `admin-restrict-local` 影响。
- `all`：Console 登录同样受 admin-restrict-local 影响。
- `disable`：行为与 7.6.0 之前的 `disable` 一致。

管理员登录锁定

功能简介

默认情况下，管理员密码连续输错 3 次后，账户会被锁定 60 秒。

如果需要提高设备管理面的安全性，可以通过 CLI 调整管理员密码的重试次数与锁定时长。

配置方法

```bash
config system global
    set admin-lockout-threshold <failed_attempts>
    set admin-lockout-duration <seconds>
end
```

- `admin-lockout-threshold`：管理员登录失败后触发锁定的阈值，默认值为 `3`，范围 1~10。
- `admin-lockout-duration`：触发锁定后的等待时间，单位为秒，默认值为 `60`，范围 1~2147483647。

提示

- `admin-lockout-threshold` 设置得越大，管理员密码被暴力猜测成功的风险也越高。
- 如果从第一次登录失败开始，到达到锁定阈值的那次失败为止，时间间隔小于 `admin-lockout-duration`，则会触发锁定。
- 管理员锁定配置会同时影响本地管理员（包括 API 管理员）与远程认证管理员。

例如，设置管理员密码输错 1 次即锁定，锁定时间为 5 分钟。

```bash
config system global
    set admin-lockout-threshold 1
    set admin-lockout-duration 300
end
```

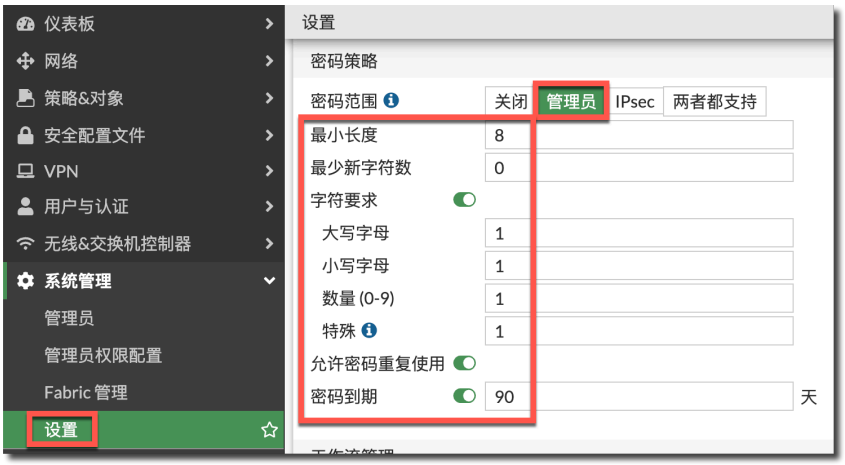
管理员密码策略

功能简介

默认情况下，FortiGate 的管理员的密码复杂度没有要求，如果需要对管理员设置密码时的字符复杂度做一定的限制，可以开启密码策略功能。

配置步骤

1. 进入系统管理 → 设置页面，在密码策略选项中，密码范围选择管理员（也可以配置 IPSec 预共享密钥的复杂度）：



- 最小长度：密码的最小字符长度
- 最少新字符数：修改密码时，允许新密码有多少个字符是旧密码中包含的
- 字母要求：
  - 大写字母：密码中至少包含的大写字母数量
  - 小写字母：密码中至少包含的小写字母数量
  - 数量：密码中至少包含的数字的数量
  - 特殊：码中至少包含的特殊字符（如_/!@）数量
- 允许密码重复使用：如果关闭，则在修改密码时不允许使用已经使用过的密码
- 密码到期：密码到期后会提示管理员修改密码

2. 相关的 CLI 配置：

```
config system password-policy
    set status enable
    set min-lower-case-letter 1
    set min-upper-case-letter 1
    set min-non-alphanumeric 1
    set min-number 1
    set expire-status enable
end

config system admin
    edit "admin"
        set accprofile "super_admin"
        set vdom "root"
        set password-expire 2023-11-22 16:21:57
        set password ENC SH2jjHLAz03uN2uhVHV9GR+F11BbxS4a2J4/3V7vnwwYJmFdG4btcop1QwEMI8=
    next
end
```

注意事项

密码策略仅支持管理员账户密码和 IPSec 预共享密钥，不支持防火墙认证用户。

通过 FortiCloud 账户登录 FortiGate

漏洞风险提示

https://www.fortiguard.com/psirt/FG-IR-25-647 ↗

功能介绍

可以使用 FortiGate 所注册的 FortiCloud 账户登陆 FortiGate，当用户有多台属于同一 FortiCloud 账户下的设备时，使用同一个 FortiCloud 账户登录多台 FortiGate，方便用户统一管理设备账户与密码，减少维护成本。

支持 FortiCloud 主账户登录，也支持 FortiCloud 下注册的 IAM 账户登录。

注意事项

1. 账户必须是 FortiGate 注册到 FortiCloud 的账户。
2. 开启后登录界面的本地账户密码输入框默认隐藏，点击用户名输入框后可以显示。
3. 权限选择继承模式时，FortiCloud 主账户总是拥有 Super admin 权限，IAM 用户可以在 FortiCloud 中设置 Super admin 或 Readonly 权限。
4. 如果 FortiCloud 主账号开启了双因子认证，则其下面的 IAM 账户也必须开启双因子认证才能登陆。

重要

在 7.2.4 GA 后，可以在 FortiGate 本地设置覆盖 FortiCloud 账号的权限。

配置步骤

FortiGate 配置

1. 进入设备管理页面的“系统管理 → 设置”页面中，在单点登录部分开启“FortiCloud 单点登录”。



- **继承自 FortiCloud**：默认情况下，FortiCloud 单点登录账户的权限（Default admin profile）会从 FortiCloud 的配置中继承，FortiCloud 主账户总是拥有 Super admin 权限，IAM 用户可以在 FortiCloud 中设置 Super admin 或 Readonly 权限。
- **指定**：在 7.2.4 GA 后，可以选择 Default admin profile 的模式为指定，本地覆盖 FortiCloud 账户的权限。

2. 对应的 CLI 如下。

```
config system global
    set admin-forticloud-sso-login enable
    set admin-forticloud-sso-default-profile 'super_admin'     //这里的配置为指定模式，如果为继承模式，则这里为空，即''
end
```

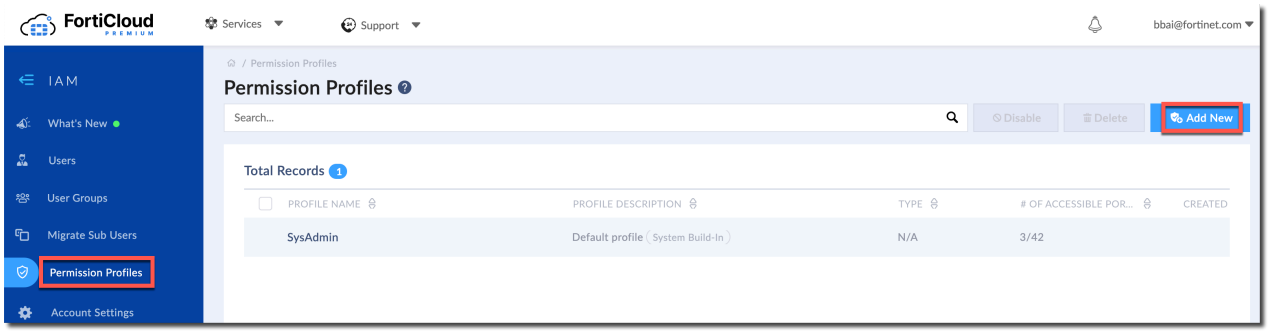
3. 这里我们先配置为默认的继承模式测试。

FortiCloud IAM 账户配置

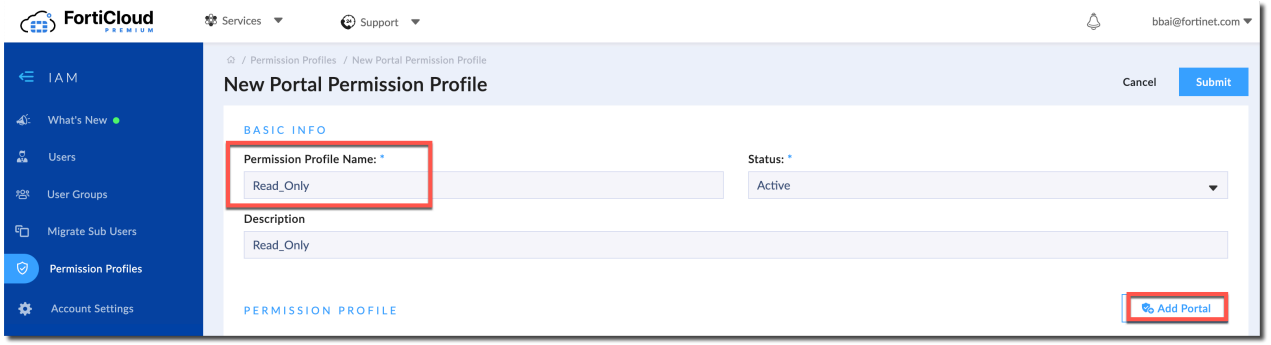
> **重要**
>
> 如果需要通过 FortiCloud 下的 IAM 子账户登录 FortiGate，则需要配置此部分。如果只需要使用 FortiCloud 主账户登录 FortiGate，则不需要配置此部分。

1. 通过 https://support.fortinet.com/iam ↗ 使用 FortiCloud 主账户登录 FortiCloud IAM 账户管理界面。

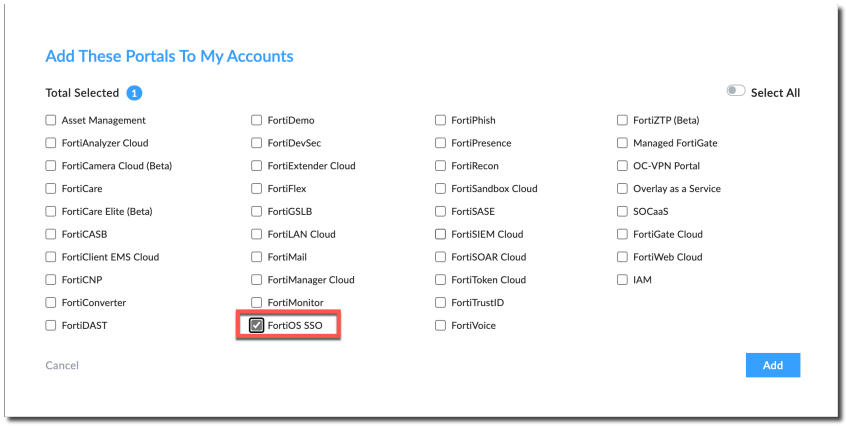
2. 进入 Permission Profiles 界面，右上角点击新建 Permission Profiles。



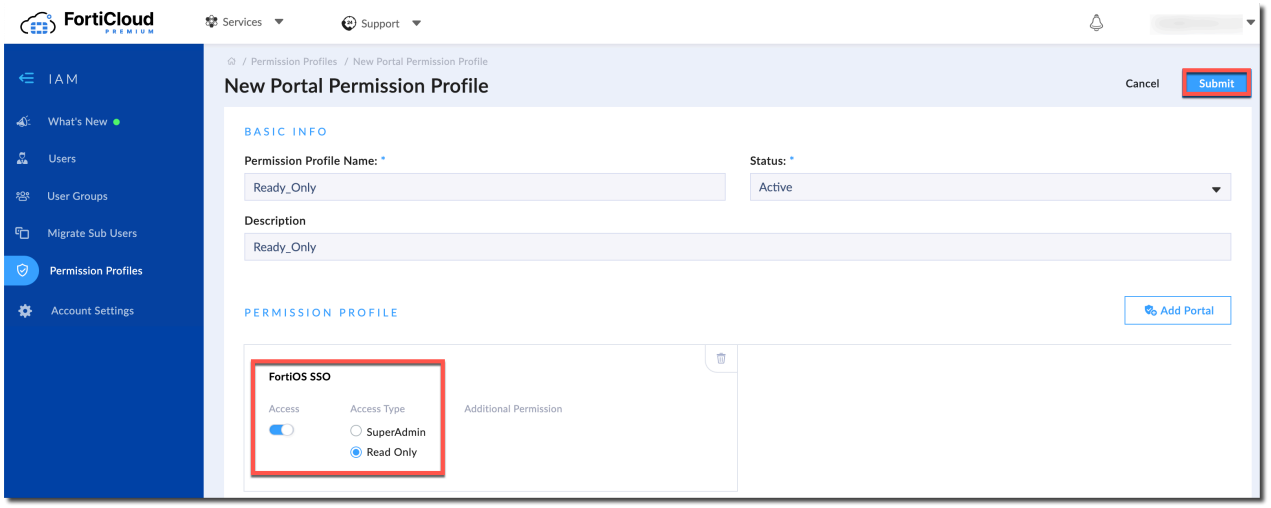
3. 填写 Permission Profiles 名称，点击右下角的 Add Portal 按钮。

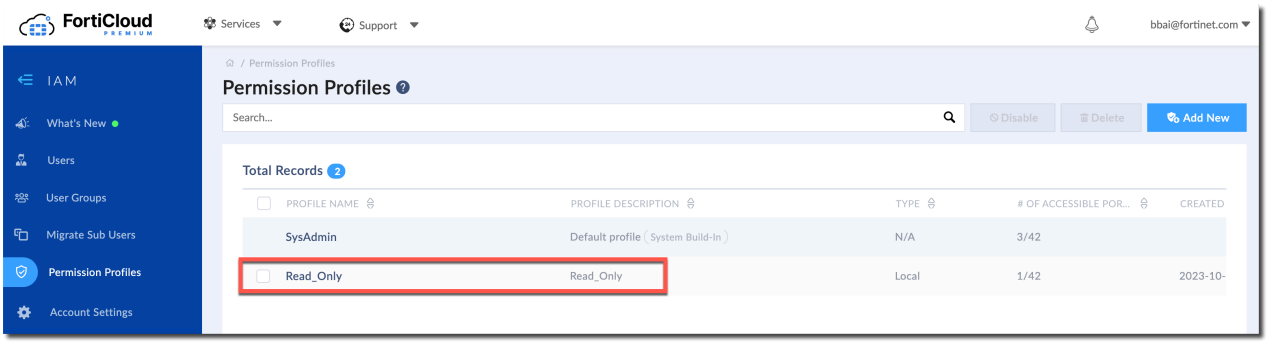


4. 在弹出的页面中选择“FortiOS SSO”，点击 Add。

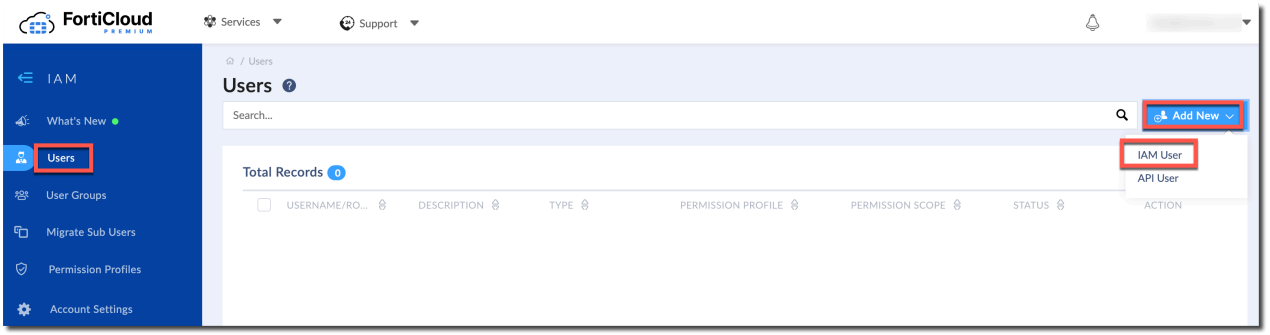


5. 开启 Access 按钮，右侧选择 Access Type 为 Ready Only，点击右上角的 Submit 按钮提交配置。

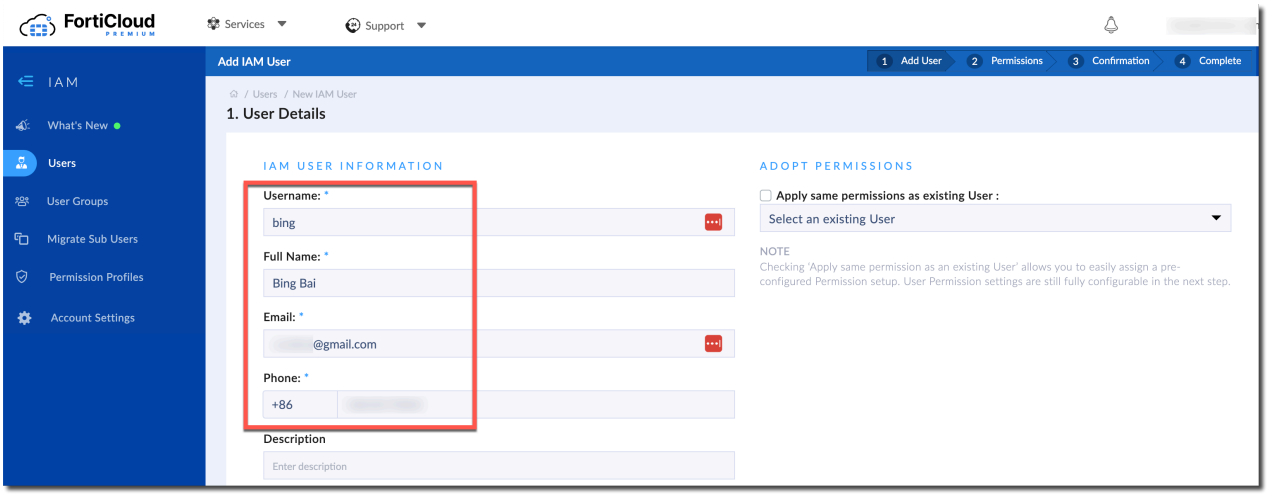




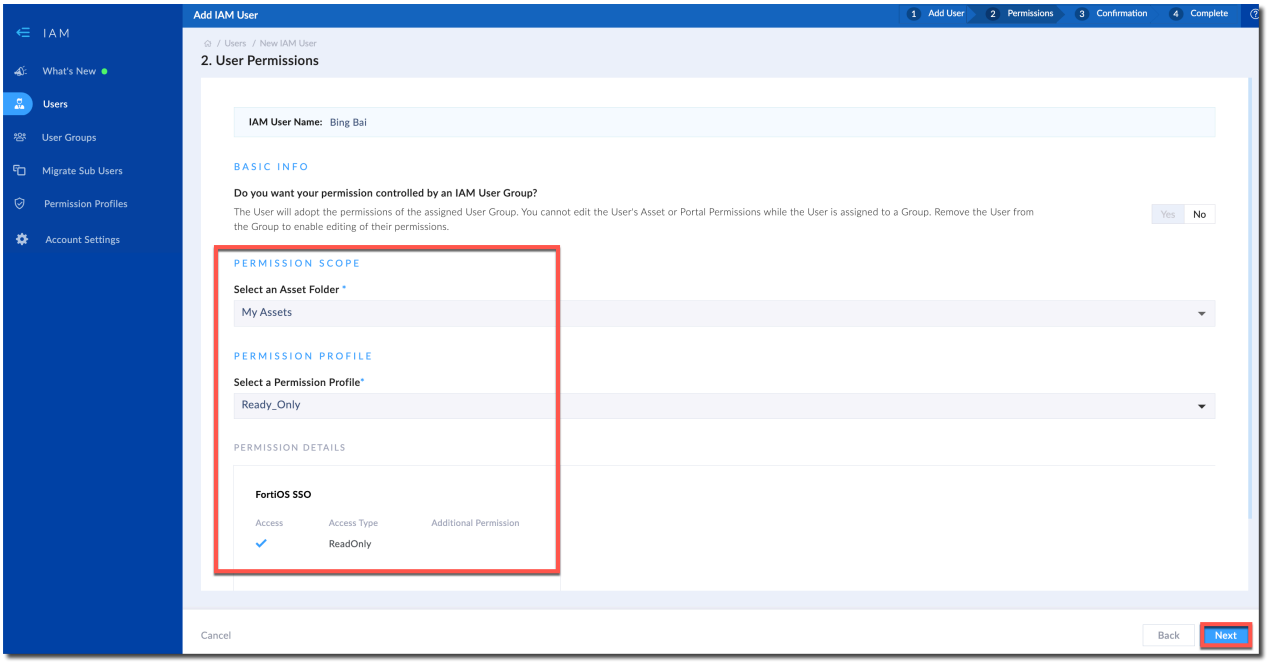
6. 进入 Users 界面，右上角点击新建 IAM User。



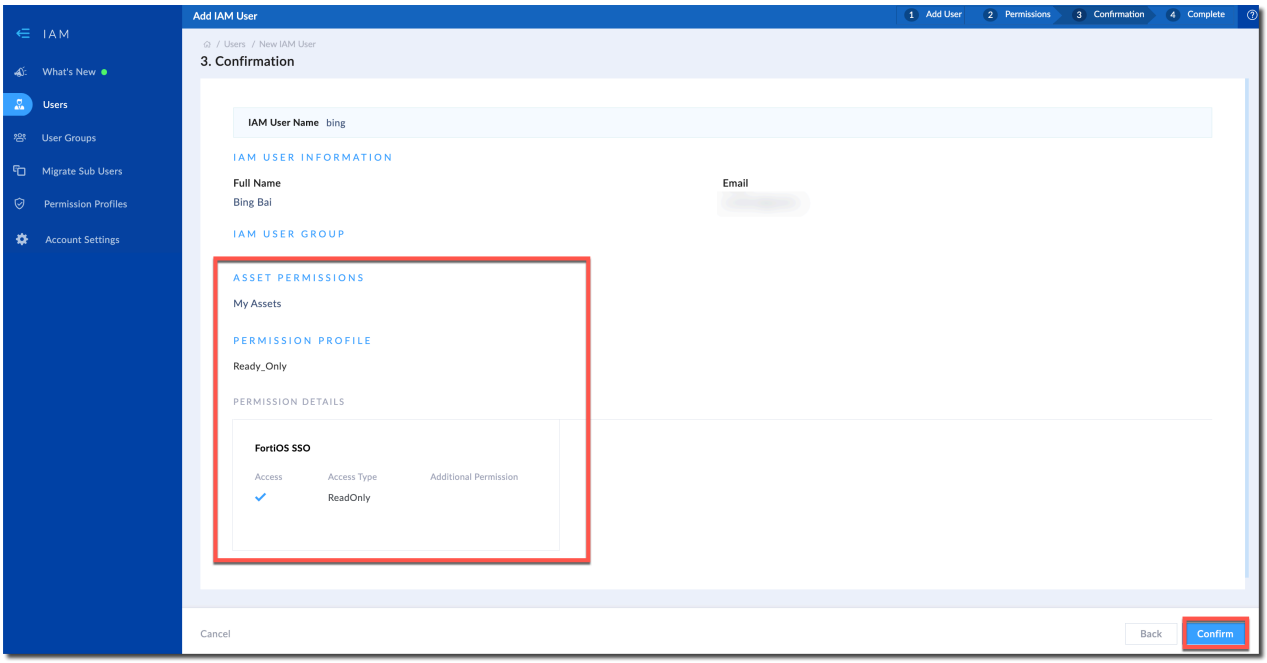
7. 填写用户相关信息，Username 用于登录 FortiGate，点击 Next。



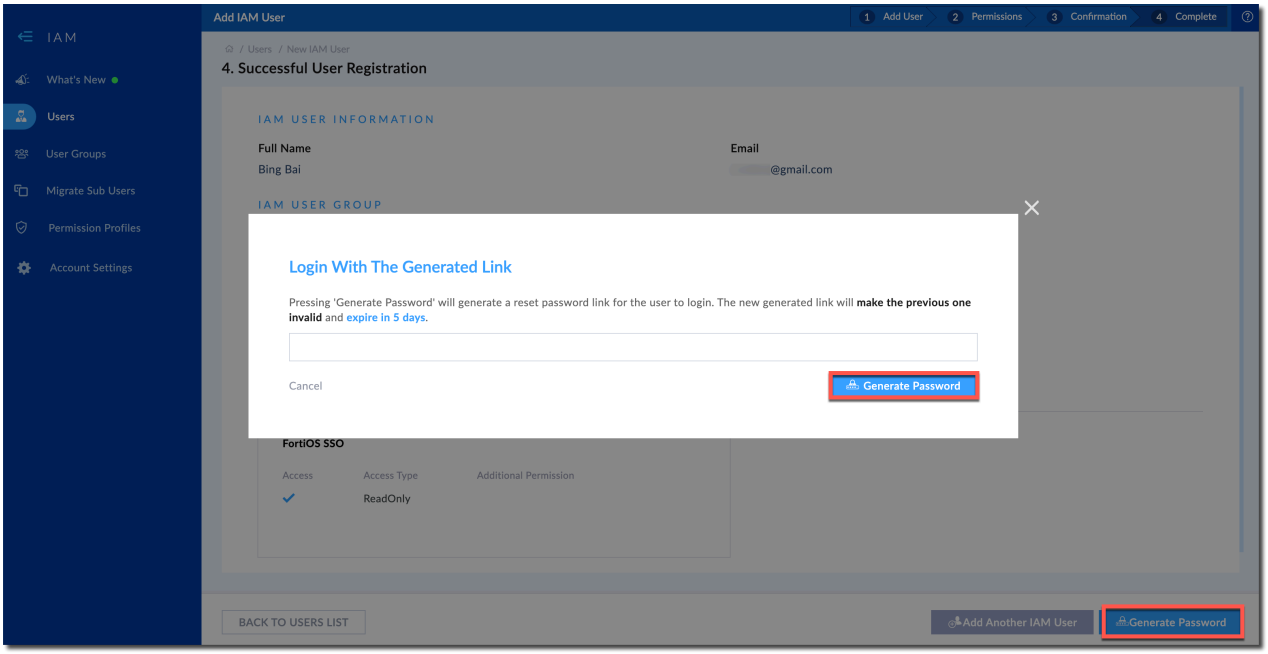
8. 选择该用户的 Asset Folder，选择之前创建的 Permission Profile Ready_Only，点击 Next。



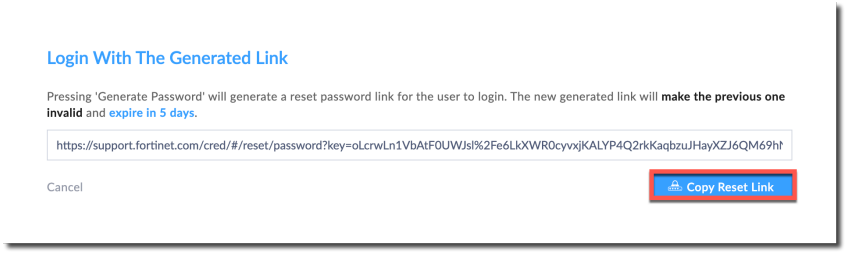
9. 确认信息无误后，点击 Confirm 创建账户。



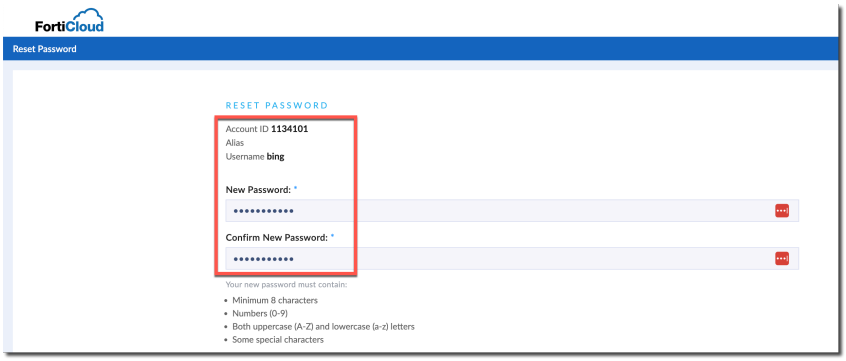
.0. 在后续页面的右下角点击 Generate Password 按钮，在弹出的窗口中点击 Generate Password 按钮。



1. 复制生成的 URL，并使用浏览器打开。



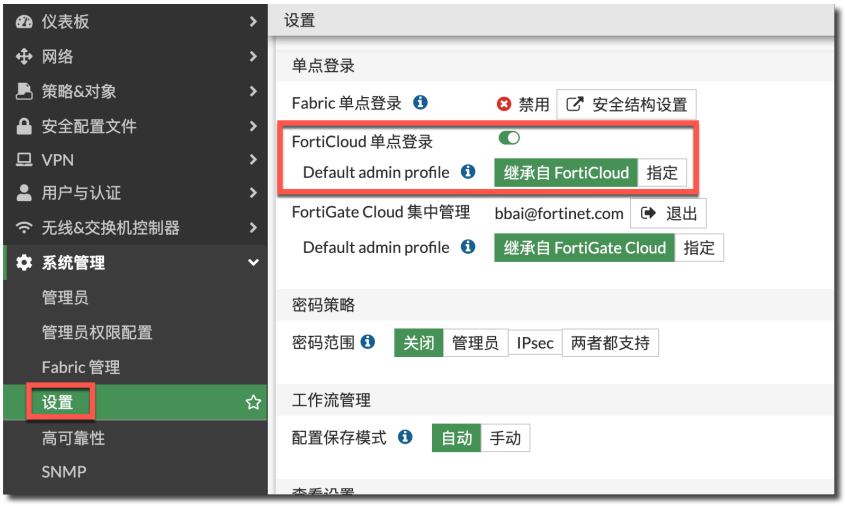
2. 在该 URL 下配置 IAM 用户的密码，记住这里的 Account ID，将用于 IAM 账户登录使用，点击 Submit 提交密码配置。



结果验证

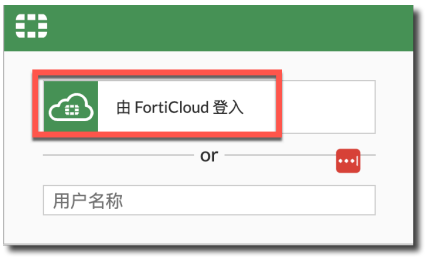
FortiCloud 主账户登录

1. Default admin profile 为继承自 FortiCloud。

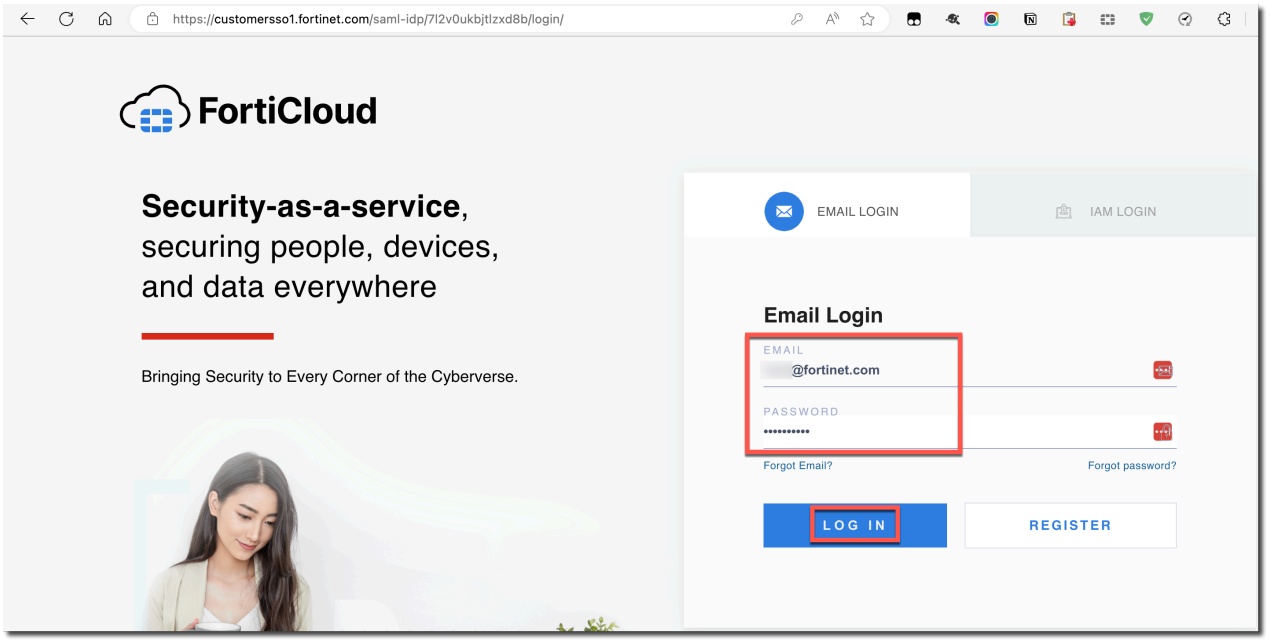



```
config system global
    set admin-forticloud-sso-login enable
    set admin-forticloud-sso-default-profile ''
end
```

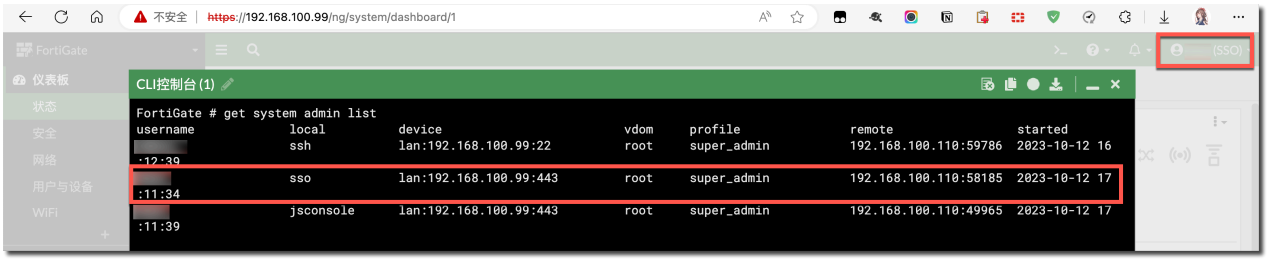
2. 访问 FortiGate 登陆页面，可以看到增加了“由 FortiCloud 登入”的选项。



3. 选择 Email Login 方式，填入设备注册到的 FortiCloud 主账户和密码，点击 Log in 按钮。



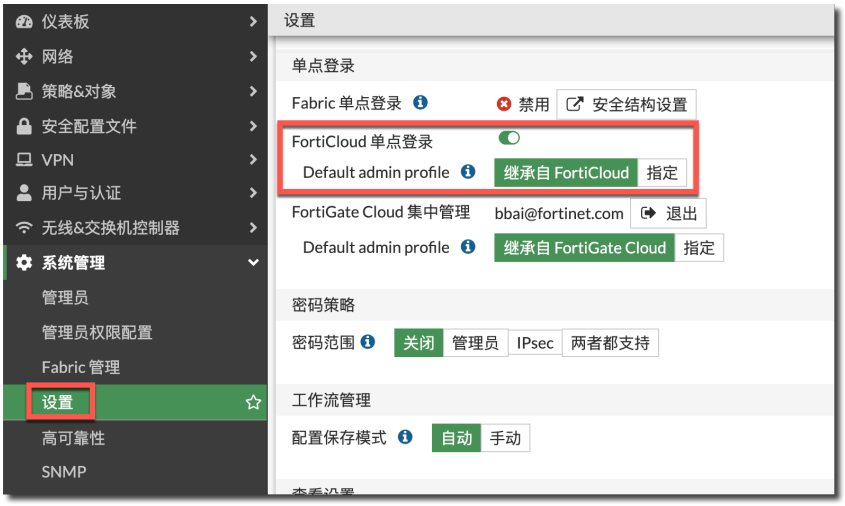
4. 登录 FortiGate 成功，登录后的权限为 super_admin。



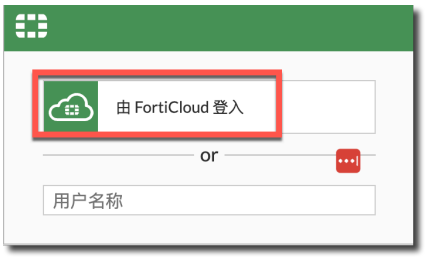

```
date=2023-10-12 time=17:11:35 eventtime=1697101894379458919 tz="+0800" logid="0100032001" type="event" subtype="system"
level="information" vd="root" logdesc="Admin login successful" sn="1697101894" user="xxxx" ui="sso(192.168.100.110)"
method="sso" srcip=192.168.100.110 dstip=192.168.100.99 action="login" status="success" reason="none"
profile="super_admin" msg="Administrator xxxx logged in successfully from sso(192.168.100.110)"
```

FortiCloud IAM 账户登录

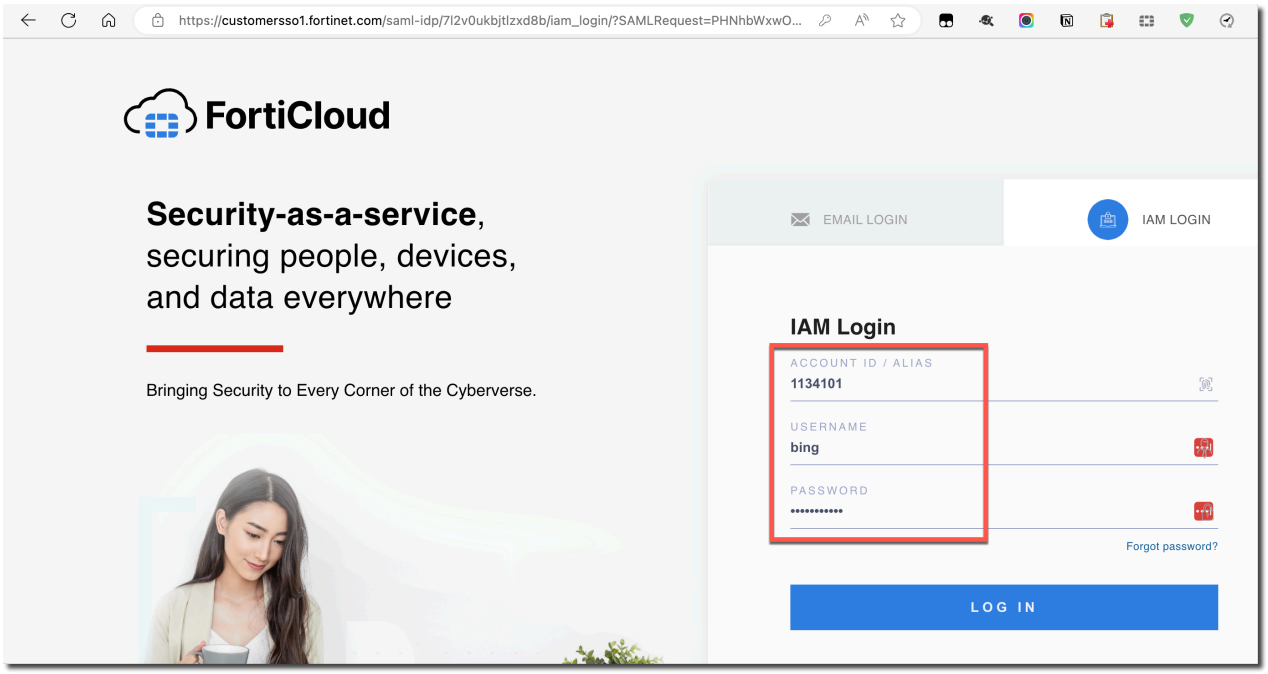
1. Default admin profile 为继承自 FortiCloud。



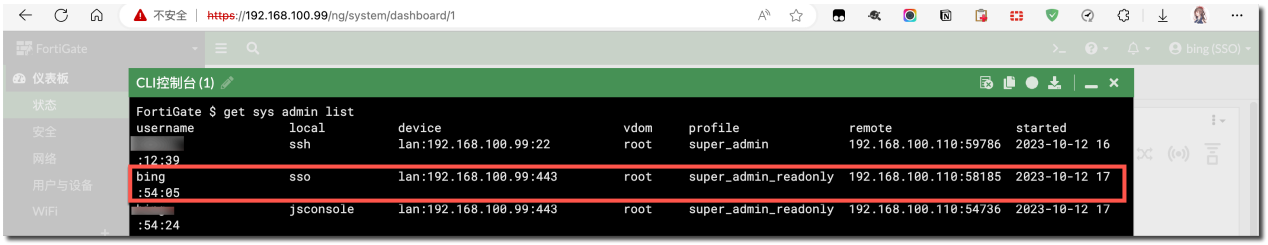
2. 访问 FortiGate 登陆页面，可以看到增加了“由 FortiCloud 登入”的选项。



3. 选择 IAM Login 方式，填入 IAM 账户的 Account ID、用户名和密码，点击 Log in 按钮。



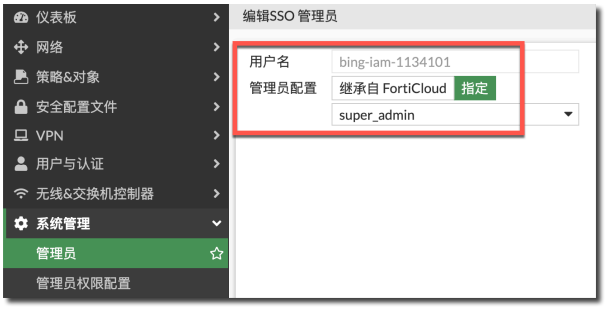
4. 登录 FortiGate 成功，登录后的权限为 read_only，且无法修改任何配置。



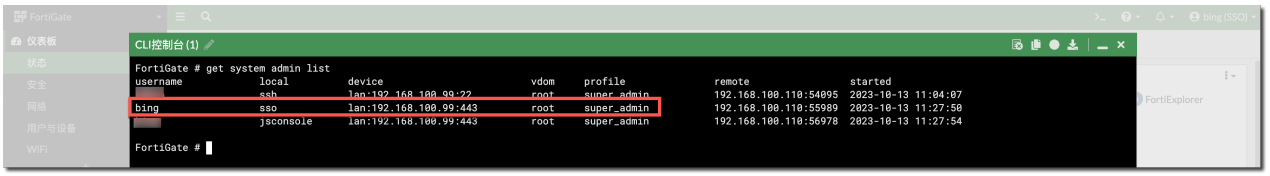

```
date=2023-10-12 time=17:54:05 eventtime=1697104445968808759 tz="+0800" logid="0100032001" type="event" subtype="system"
level="information" vd="root" logdesc="Admin login successful" sn="1697104445" user="bing" ui="sso(192.168.100.110)"
method="sso" srcip=192.168.100.110 dstip=192.168.100.99 action="login" status="success" reason="none"
profile="super_admin_readonly" msg="Administrator bing logged in successfully from sso(192.168.100.110)"
```

5. 使用本地管理员修改已创建的 FortiCloud IAM 用户的权限为 super_admin（如果是未登陆过的 FortiCloud 用户，则只需在系统管理 → 设置中修改 Default admin profile 为指定 super_admin 权限，如果是已登录过的 FortiCloud 账户，必须使用本地管理员在管理员设置中配置权限）。





6. 再次使用 FortiCloud IAM 用户登录，登录 FortiGate 成功，登录后的权限为 super_admin。



限制管理员 SSH/Telnet 跳板

重要

该功能在 FortiOS 7.2.1 及以上版本支持。

网络需求

默认配置下，管理员登录 FortiGate 后，可以使用 SSH、Telnet 功能远程连接到其他主机。

出于安全性的需求，FortiGate 可以配置管理员所使用的管理员配置文件，禁止其使用 FortiGate 的 SSH、Telnet 功能作为跳板来远程连接到其他主机。

配置方法

此功能仅在 CLI 下可配置。

重要

内置的 `super_admin` 管理员配置文件无法开启此功能。默认配置为 `enable`。

```bash
config system accprofile
    edit < name >
        set system-execute-ssh < enable | disable >    //开启或关闭SSH权限//
        set system-execute-telnet < enable | disable >    //开启或关闭Telnet权限//
    next
end
```

配置步骤

1. 在 FortiGate 上新建管理员配置文件，并禁止使用 FortiGate 的 SSH、Telnet 功能作为跳板来远程连接到其他主机的功能。

```bash
config system accprofile
    edit "Restrict_SSH_and_Telnet"
        set secfabgrp read-write
        set ftviewgrp read-write
        set authgrp read-write
        set sysgrp read-write
        set netgrp read-write
        set loggrp read-write
        set fwgrp read-write
        set vpngrp read-write
        set utmgrp read-write
        set wanoptgrp read-write
        set wifi read-write
        set system-execute-ssh disable
        set system-execute-telnet disable
    next
end
```

2. 创建管理员引用上步创建的管理员配置文件。

```bash
config system admin
    edit "admin_beijing"
        set accprofile "Restrict_SSH_and_Telnet"
        set vdom "root"
        set password fortinet
    next
end
```

配置验证

1. 使用管理员 `admin_beijing` 登录 FortiGate（通过 Web CLI 或 SSH 登录），并在 FortiGate 下使用 Telnet、SSH 命令访问其他服务器，提示没有权限运行该命令。

```bash
C:\Users\cyndi>ssh admin_beijing@10.10.12.1
admin_beijing@10.10.12.1's password:

FortiGate $ exe telnet 192.168.100.99
You are not entitled to run the command.
Command fail. Return code -37

FortiGate $ exe ssh 192.168.100.99
You are not entitled to run the command.
Command fail. Return code -37
```

2. 使用管理员 `admin` 登录 FortiGate（通过 Web CLI 或 SSH 登录），使用 Telnet、SSH 命令可以正常访问其他服务器。

```bash
C:\Users\cyndi>ssh admin@10.10.12.1
admin@10.10.12.1's password:

FortiGate # exe telnet 192.168.100.99
Trying 192.168.100.99 ...
Connected to 192.168.100.99.
FW1_FGT101F login: admin
Password:
Welcome!
FW1_FGT101F # exit
Connection closed by foreign host.

FortiGate # exe ssh admin@192.168.100.99
admin@192.168.100.99's password:
FW1_FGT101F # exit
Connection to 192.168.100.99 closed.
FortiGate #
```

密码恢复

重要

此功能仅支持 7.2.4 之前的 FortiOS 版本。为了安全性考虑，从 7.2.4 GA 版本开始，FortiOS 不再支持密码恢复功能，请务必妥善保存管理员密码以及定期做好配置备份！

功能需求

1. 若设备的密码忘记，需要用配置线进行密码恢复。
2. 密码恢复需要重启设备，并在设备的 底层菜单界面上操作，会造成网络中断，请在方便断网时操作。
3. 密码恢复后 配置不会改变。

配置要点

1. 使用终端软件（如 SecureCRT、XSHELL）连接防火墙串口。
2. 断电重启设备，输入使用内置账户 maintainer 登录。
3. 为管理员用户重新设置新的密码。

配置步骤

1. 连好 Console 线，设置并连接终端软件（详见系统管理 → 管理方式 → console 管理方式 ↗）。
2. 断电重启设备，重启后在 15 秒内输入用户名 maintainer， 密码为 bcpb + 该设备的 SN 号，通常是以 FGT 开头的 16 字符串，产品序列号见设备的底部或者侧面，如下图所示。



3. 该账户仅在断电重启后 15 秒内有效，且必须要通过 Console 口使用。

```
FGT60C3G10000413 login: maintainer
Password:  bcpbFGT60C3G10000413
FGT60C3G10000413 #
```

4. 修改管理员账号密码。

```
FGT60C3G10000413# config system admin
FGT60C3G10000413 (admin) # edit admin
FGT60C3G10000413(admin) # set pass 123455@!@#
FGT60C3G10000413 (admin) # end
```

结果验证

使用刚重置密码的 admin 账户，通过 https 或者 ssh 登陆防火墙。

API 用户

API 文档获取

API 用户配置

API 常见问题

- API 认证失败

API 文档获取

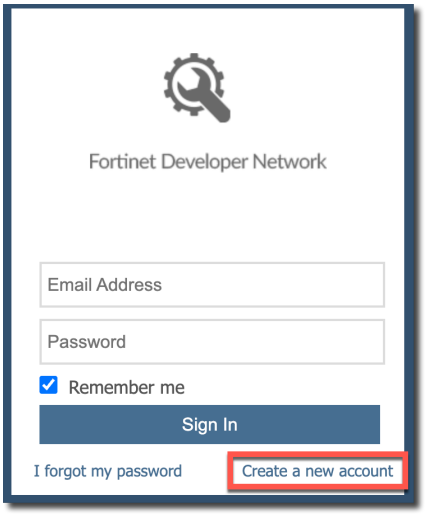
Fortinet Developer Network（FNDN）是一个基于订阅的社区，可帮助管理员提升和增强 Fortinet 产品的效能。您可以访问 FNDN 中的 FortiAPI 模块，以获取 FortiOS 各个版本的 API 列表与 API Demo。

要访问 FNDN，您需要先使用邮箱账户注册 FNDN，所有 FNDN 用户必须由两名 Fortinet 员工进行担保。担保人必须能够确认用户的身份以及访问需求，创建新的 FNDN 账户需要提供两位担保人的电子邮件地址。

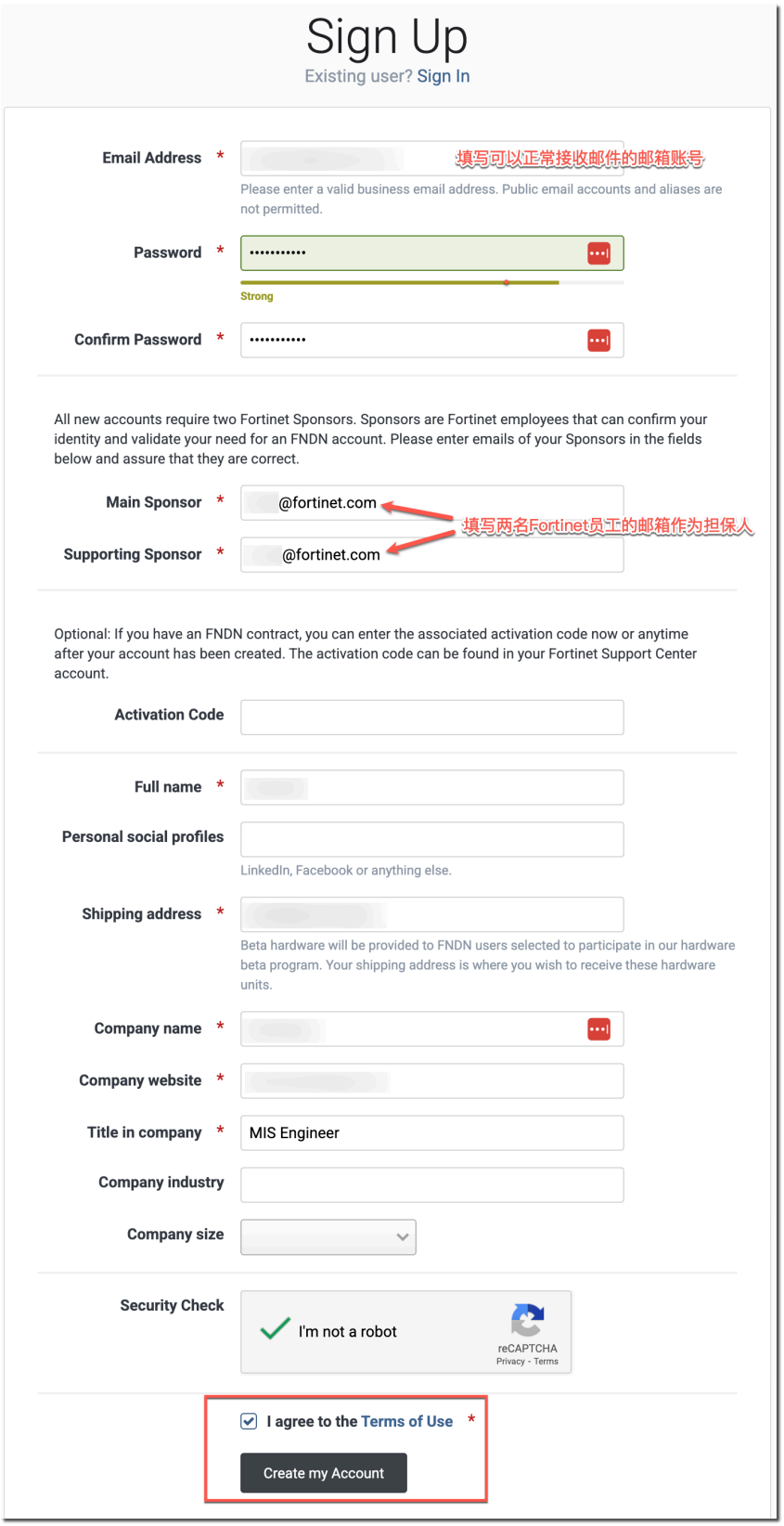
账号创建

1. 首先需要获得两名 Fortinet 员工的担保，您可以向您的销售或售前人员咨询，并获得两名担保人的邮箱。

2. 访问 FNDN 网站，https://fndn.fortinet.net/ ↗ ，点击“Create a new account”。



3. 在注册页面填写必填项，填写两名 Fortinet 员工的邮箱作为担保人，同意使用条款并创建账户。



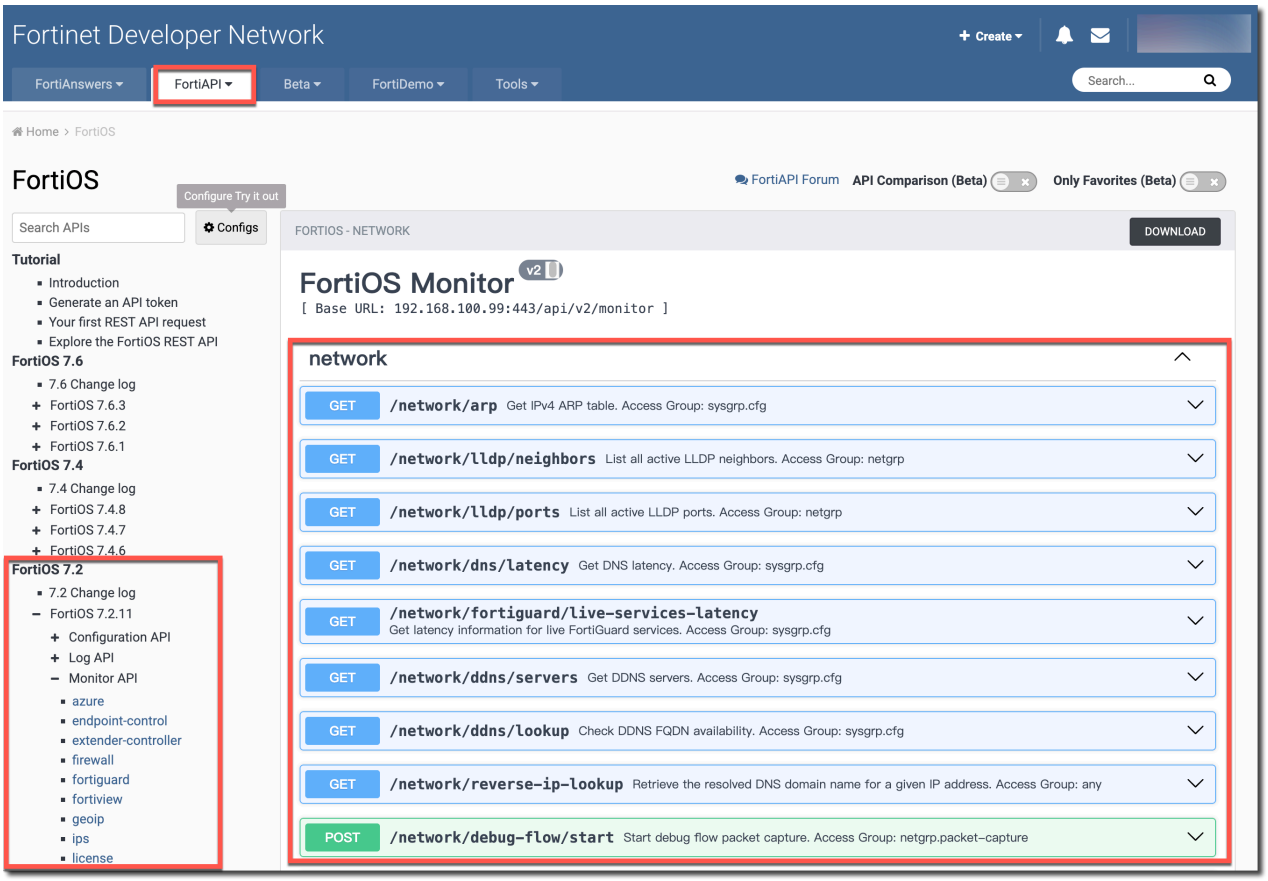
4. 随后在两位 Fortinet 担保人都批准申请后，若满足要求，FNDN 管理员将审核该申请，并在大约一个工作日内批准账户访问权限，您将会在注册邮箱中收到审核结果的通知邮件。

通过 FNDN 使用 API

重要

FortiGate API 用户的配置方法请参考：系统管理 → 管理员配置 → API 用户 → API 用户配置 ↗章节。

1. FNDN 账号审核通过后，登录到 https://fndn.fortinet.net/index.php?/fortiapi/1-fortios/ ↗ ，可以浏览对应 FortiOS 版本的 API 列表。

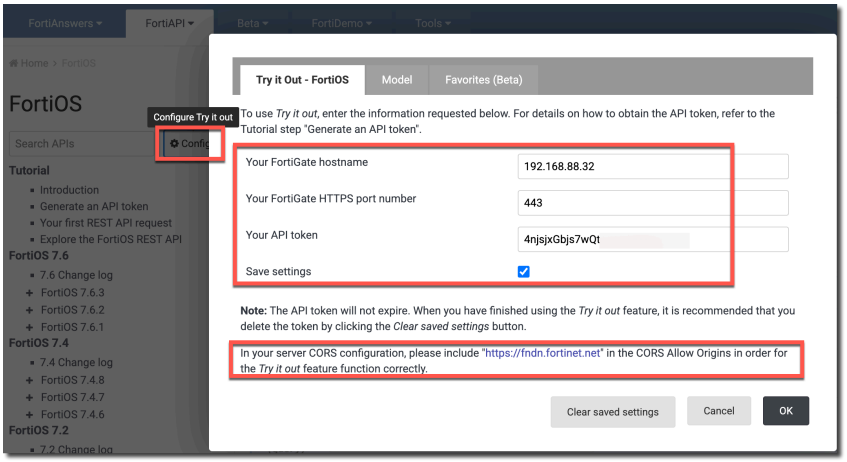


2. 点击左侧的“Config”按钮，填写本地 FortiGate 的 API 访问信息，可以在 FNDN 上通过本地网络访问目标设备的 API。

重要

需要 FortiGate 的 API 用户允许 `https://fndn.fortinet.net` 的 CORS 访问：

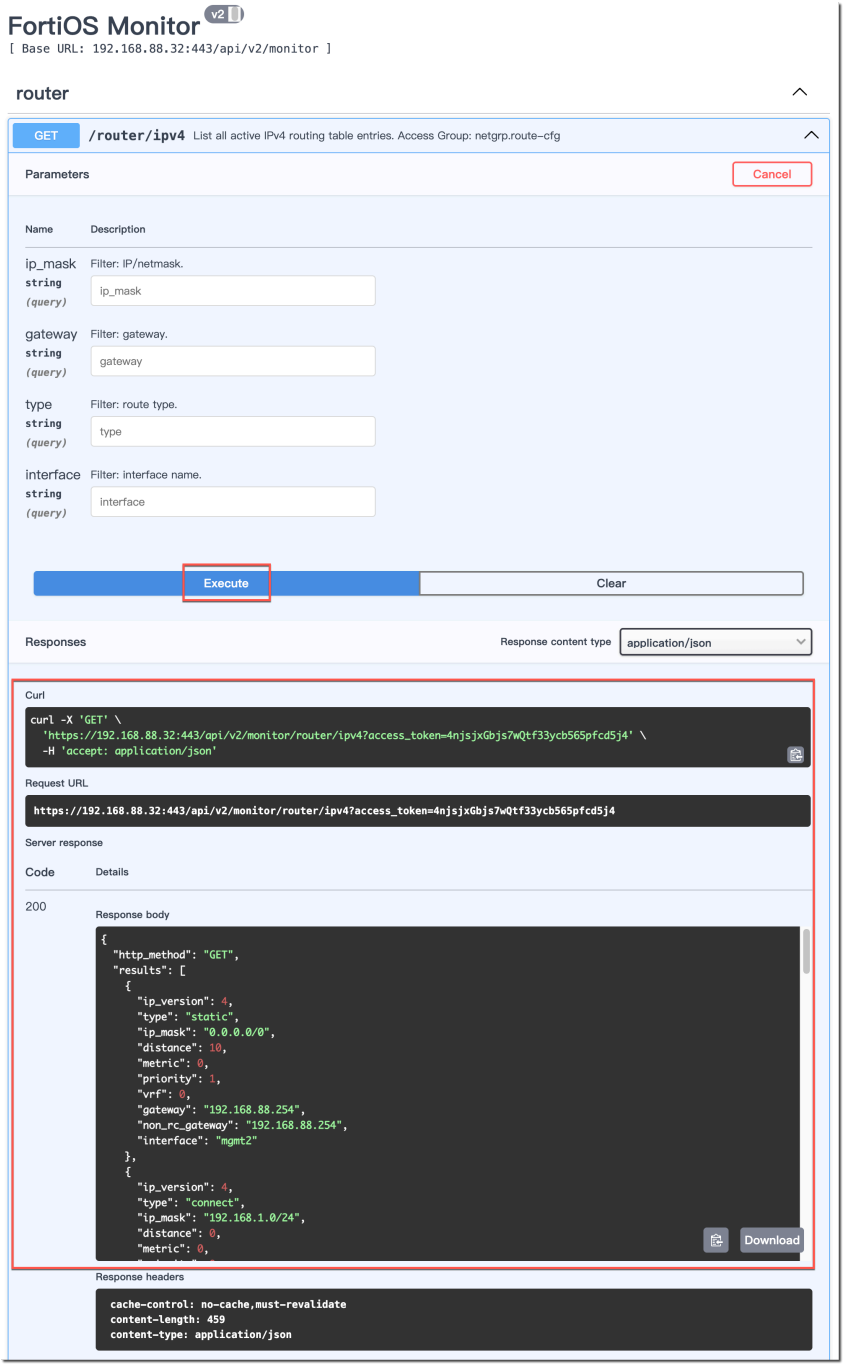
```bash
config system api-user
    edit "api_user"
        set cors-allow-origin "https://fndn.fortinet.net"
    next
end
```



3. 例如需要通过 API 获取 FortiGate 当前的 IPv4 路由表，在该 API 条目中点击“Try it out”按钮。



4. 可以根据需要配置过滤器，点击“Execute”按钮执行 API 查询动作，可以看到 FortiGate 返回了当前的 IPv4 路由表信息。另外也可以看到 API 具体执行的 Curl 格式与请求的 URL。



API 用户配置

相关信息

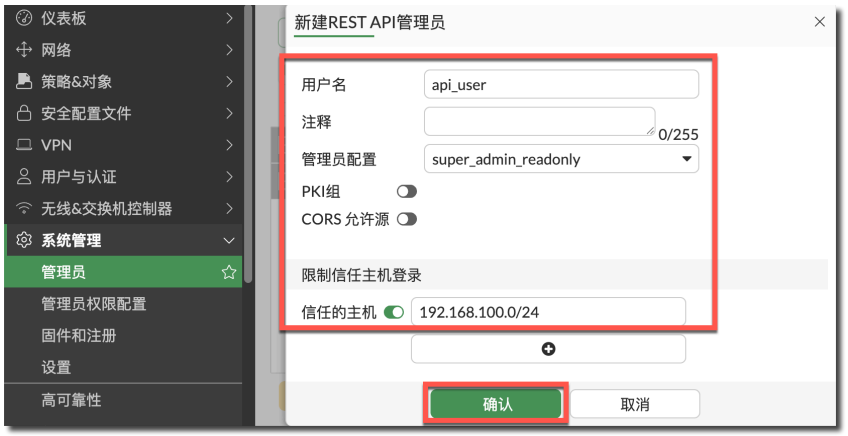
API 文档获取方式请参考：系统管理 → 管理员配置 → API 用户 → API 文档获取 ↗章节。

用户创建

1. 进入 GUI 的“系统管理 → 管理员”页面，点击“新建”按钮，选择“REST API 管理员”。

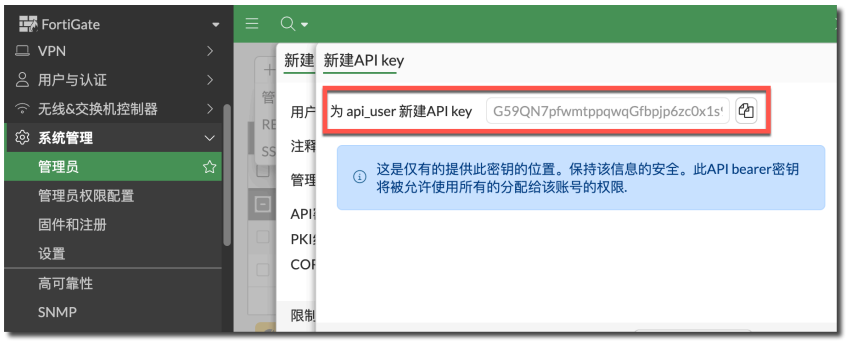


2. 配置 API 用户名与“管理员配置”，这里以预置的“super_admin_readonly”为例。按需要配置“PKI 组”、“CORS 允许源”、“信任的主机”配置。



- 管理员配置：选择 API 用户的管理员权限配置（配置请参考“系统管理 → 管理员配置 → 管理员设置 ↗”章节，在 GUI 下，只能选择“super_admin_readonly”、“admin_no_access”以及自定义的管理员权限文件。如果需要选择预置的“super_admin”和“prof_admin”，需要在 CLI 下配置。
- PKI 组：在 API key 认证的基础上开启 REST API 客户端证书认证，只有此 PKI 组中的证书用户可以访问 API。
- CORS 允许源：当 REST API 客户端是通过跨域请求（CORS）的方式访问 FortiGate 的 API 时，需要允许 CORS 的网站源地址（具体示例请见“系统管理 → 管理员配置 → API 用户 → API 文档获取 ↗”中的“通过 FNDN 使用 API”部分）。
- 信任的主机：限制允许访问的 REST API 客户端 IP。

3. 点击“确认”按钮后，会生成 API key。注意该 API key 只会在这里显示一次，请妥善保存 API key。



4. API 用户创建完成。




```bash
config system api-user
    edit "api_user"
        set api-key ENC SH281AhpqGoK53Cq8e/pcf89amfyatlAMsh3uP0427kvqCNdFFZViV9L6q11ao=
        set accprofile "super_admin_readonly"
        config trusthost
            edit 1
                set ipv4-trusthost 192.168.100.0 255.255.255.0
            next
        end
    next
end
```

5. 使用 API 方式获取设备信息，例如获取设备的 VDOM 信息。

> **提示**
>
> - 推荐使用 HTTP Header 传递 token 的方式，更安全。
> - 从 FortiOS v7.4.5 和 v7.6.1 开始，系统默认禁止通过 URL 参数传递 `access_token`（参考系统管理 → 管理员配置 → API 用户 → API 常见问题 → API 认证失败 ↗章节。

```bash
curl -k 'https://192.168.100.99/api/v2/cmdb/firewall/address/' \
    --header 'Authorization: Bearer 785yfsnmc3NzjnHj0jNkh87fdNcb63'

{"http_method":"GET","size":1,"limit_reached":false,"matched_count":1,"next_idx":0,"revision":"9b107552a7c5cc639bb9c77dc
3cf43b2","results":[{"name":"root","q_origin_key":"root","short-name":"root","vcluster-
id":0,"flag":0}],"vdom":"root","path":"system","name":"vdom","action":"","status":"success","http_status":200,"serial":"
FG101FTK20007637","version":"v7.6.6","build":3652}%
```

其他选项

CLI 配置

```bash
config system api-user
    edit <user_name>
        set comments <string>
        set api-key ************     ←——只能系统生成，无法在CLI中指定
        set accprofile <admin_accprofile>
        set schedule <schedule_object>
        set cors-allow-origin "https://fndn.fortinet.net"
        set peer-auth enable
        set peer-group <group>
        config trusthost
            edit 1
                set type ipv4-trusthost
                set ipv4-trusthost <class_ip&net_netmask>
            next
            edit 2
                set type ipv6-trusthost
                set ipv6-trusthost <class_ip&net_netmask>
            next
            ...
        end
    next
end
```

生成 API key

- 由系统生成或重新生成 API key，该 key 无法在 CLI 中指定。

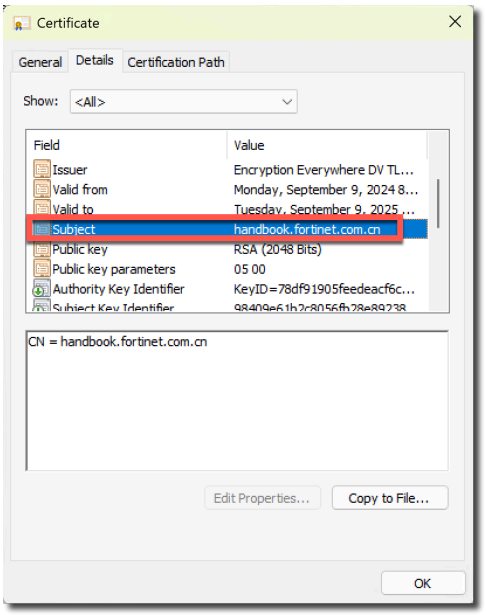
```bash
execute api-user generate-key <API username>

execute api-user generate-key api_user
```

- 从 FortiOS 7.6.0 版本开始，可以配置 API key 的过期时间，单位为 min。

```bash
execute api-user generate-key <API username> <expiry time>

execute api-user generate-key api_user 480
```

API PKI 证书认证

1. 配置 PKI 用户，指定用于 REST API 客户端认证使用的证书用户，这些信息可以从用于 REST API 客户端认证的用户证书属性中找到。




```bash
config user peer
    edit "pki_user"
        set mandatory-ca-verify enable    ←——默认开启
        set ca "DigiCertGlobalRootG2"
        set subject "CN = handbook.fortinet.com.cn"
        set cn "handbook.fortinet.com.cn"
    next
end
```

重要

`mandatory-ca-verify` 选项：

- `enable`：需要将 API 客户端证书对应的 CA 证书安装在 FortiGate 上（导入方法参考：系统管理 → 系统设置 → 证书管理 ↗章节），认证时 FortiGate 会验证客户端证书是否与 `set ca` 指定的 CA 证书匹配（默认启用）。同时也要验证客户端提供的证书中的字段是否与 PKI 用户中配置的一致。
- `disable`：只要 API 客户端提供的证书中的属性与 PKI 用户中配置的一致，FortiGate 即认为客户端证书有效。
- 验证证书字段有 3 种方式：Subject、CN 或 LDAP，所有字符串比较均区分大小写。

2. 创建用户组引用 PKI 用户。

```bash
config user group
    edit "api_usergrp"
        set member "pki_user"
    next
end
```

3. 在 API 用户中开启“PKI 组”认证，并引用 PKI 用户组。




```bash
config system api-user
    edit api_user
        set peer-auth enable
        set peer-group api_usergrp
    next
end
```

4. API 客户端可以使用如下命令指定 API 访问时使用的证书文件以及 API key。

```bash
# curl -k \
  --key ./<user private key> \
  --cert <path of user certificate> \
  -H "Authorization: Bearer <api_key>" \
  "https://<IP/FQDN>/api/v2/cmdb/firewall/address/"

# curl -k \
  --key ./useKeyr.pem \
  --cert ./handbook.fortinet.com.cn.crt \
  -H "Authorization: Bearer bbQfw68qrnyN3yyybznGtgGx897qH3" \
  "https://192.168.100.99/api/v2/cmdb/firewall/address/"
```

5. FortiGate 可以正确返回 API 查询结果。

```bash
# curl -kv \
  --key ./useKeyr.pem \
  --cert ./handbook.fortinet.com.cn.crt \
  -H "Authorization: Bearer bbQfw68qrnyN3yyybznGtgGx897qH3" \
  "https://192.168.100.99/api/v2/cmdb/firewall/address/"

* Server certificate:
* subject: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=FortiGate; CN=FGVM020000158352;
emailAddress=support@fortinet.com
* start date: Oct 23 11:04:36 2019 GMT
* expire date: Jan 19 03:14:07 2056 GMT
* common name: FG101FTK2000XXXX
* issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=fortinet-subca2001;
emailAddress=support@fortinet.com
::: important
GET /api/v2/cmdb/firewall/address/?access_token=bbQfw68qrnyN3yyybznGtgGx897qH3 HTTP/1.1
Host: FG101FTK2000XXXX
User-Agent: curl/8.7.1
Accept: */*

:::
< HTTP/1.1 200 OK
< Date: Wed, 08 Jul 2025 17:19:13 GMT
< Server: Apache
< X-Frame-Options: SAMEORIGIN
< Content-Security-Policy: frame-ancestors 'self'
< X-XSS-Protection: 1; mode=block
< Strict-Transport-Security: max-age=0
< X-UA-Compatible: IE=Edge
< Cache-Control: no-cache, must-revalidate
```

```
< ETag: 2a3997e13943e98865004c368a65a8de
< Content-Length: 7940
< Content-Type: application/json
<
{
"http_method":"GET",
"revision":"2a3997e13943e98865004c368a65a8de",
"results":[
{
"name":"FABRIC_DEVICE",
"q_origin_key":"FABRIC_DEVICE",
"uuid":"0f1a2390-525c-51f0-2ad2-b213f4cba84b",
"subnet":"0.0.0.0 0.0.0.0",
"type":"ipmask",
"sub-type":"sdn",
"clearpass-spt":"unknown",
"start-mac":"00:00:00:00:00:00",
"end-mac":"00:00:00:00:00:00",
"country":"",
"cache-ttl":0,
"sdn":"",
"fsso-group":[
],
......
```

API 常见问题

- API 认证失败

API 认证失败

问题现象

从 FortiOS v7.4.5 和 v7.6.1 开始，当 API 请求通过 URL 参数传递 `access_token` 时，会返回 `401 Unauthorized` 错误。

```bash
curl -vk 'https://192.168.100.99/api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63'
*   Trying 192.168.100.99:443 ...
* Connected to 192.168.100.99 (192.168.100.99) port 443
* ALPN: curl offers h2,http/1.1
* (304) (OUT), TLS handshake, Client hello (1):
* (304) (IN), TLS handshake, Server hello (2):
* (304) (IN), TLS handshake, Unknown (8):
* (304) (IN), TLS handshake, Certificate (11):
* (304) (IN), TLS handshake, CERT verify (15):
* (304) (IN), TLS handshake, Finished (20):
* (304) (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / AEAD-AES256-GCM-SHA384 / [blank] / UNDEF
* ALPN: server accepted h2
* Server certificate:
*  subject: C=US; ST=California; L=Sunnyvale; O=Fortinet Ltd.; OU=FortiGate; CN=FortiGate
*  start date: Mar 16 07:59:45 2026 GMT
*  expire date: Jun 18 07:59:45 2028 GMT
*  issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=FG101FTK20007637;
emailAddress=support@fortinet.com
*  SSL certificate verify ok.
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://192.168.100.99/api/v2/cmdb/system/vdom?
access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: 192.168.100.99]
* [HTTP/2] [1] [:path: /api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63]
* [HTTP/2] [1] [user-agent: curl/8.7.1]
* [HTTP/2] [1] [accept: */*]
> GET /api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63 HTTP/2
> Host: 192.168.100.99
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/2 401
< x-frame-options: SAMEORIGIN
< content-security-policy: frame-ancestors 'self'
< x-xss-protection: 1; mode=block
< strict-transport-security: max-age=63072000
< www-authenticate: error="invalid_token"
< date: Thu, 26 Mar 2026 02:48:46 GMT
<
* Connection #0 to host 192.168.100.99 left intact
```

相关信息

从 FortiSOAR 等第三方平台连接 FortiGate 时，也可能出现类似错误：`Unauthorized. Wrong API key provided OR check user/VDOM/API key permission`。

问题原因

从 FortiOS v7.4.5 和 v7.6.1 开始，系统默认禁止通过 URL 参数传递 `access_token`。此变更是出于安全最佳实践的考虑：token 直接拼在 URL 中，会被记录在浏览器历史、代理日志、服务器访问日志中，存在泄露风险。

新版本要求 `access_token` 必须通过 HTTP Header 传递，即使用 `Authorization: Bearer <token>` 的方式。

解决方法

使用 HTTP Header 传递 token（推荐）

1. 将 API token 放在请求 Header 中，不会出现在 URL 里，更安全。正常返回 API 结果：

```bash
curl -vk 'https://192.168.100.99/api/v2/cmdb/system/vdom' \
    --header 'Authorization: Bearer 785yfsnmc3NzjnHj0jNkh87fdNcb63'
*   Trying 192.168.100.99:443 ...
* Connected to 192.168.100.99 (192.168.100.99) port 443
* ALPN: curl offers h2,http/1.1
* (304) (OUT), TLS handshake, Client hello (1):
* (304) (IN), TLS handshake, Server hello (2):
* (304) (IN), TLS handshake, Unknown (8):
* (304) (IN), TLS handshake, Certificate (11):
* (304) (IN), TLS handshake, CERT verify (15):
* (304) (IN), TLS handshake, Finished (20):
* (304) (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / AEAD-AES256-GCM-SHA384 / [blank] / UNDEF
* ALPN: server accepted h2
* Server certificate:
*  subject: C=US; ST=California; L=Sunnyvale; O=Fortinet Ltd.; OU=FortiGate; CN=FortiGate
*  start date: Mar 16 07:59:45 2026 GMT
*  expire date: Jun 18 07:59:45 2028 GMT
*  issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=FG101FTK20007637;
emailAddress=support@fortinet.com
*  SSL certificate verify ok.
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://192.168.100.99/api/v2/cmdb/system/vdom
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: 192.168.100.99]
* [HTTP/2] [1] [:path: /api/v2/cmdb/system/vdom]
* [HTTP/2] [1] [user-agent: curl/8.7.1]
* [HTTP/2] [1] [accept: */*]
* [HTTP/2] [1] [authorization: Bearer 785yfsnmc3NzjnHj0jNkh87fdNcb63]
> GET /api/v2/cmdb/system/vdom HTTP/2
> Host: 192.168.100.99
> User-Agent: curl/8.7.1
> Accept: */*
> Authorization: Bearer 785yfsnmc3NzjnHj0jNkh87fdNcb63
>
* Request completely sent off
< HTTP/2 200
< content-security-policy: frame-ancestors 'self'; script-src 'self'; object-src 'none';
< strict-transport-security: max-age=63072000
< date: Thu, 26 Mar 2026 03:02:35 GMT
< etag: EE8EF382372E2A3C07F45B9C09FB034AD9564AD6B64F1EAF5510A7C7374DCF8A
< cache-control: no-cache, must-revalidate
< content-length: 377
< content-type: application/json
<
* Connection #0 to host 192.168.100.99 left intact
{"http_method":"GET","size":1,"limit_reached":false,"matched_count":1,"next_idx":0,"revision":"9b107552a7c5cc639bb9c77dc
3cf43b2","results":[{"name":"root","q_origin_key":"root","short-name":"root","vcluster-
id":0,"flag":0}],"vdom":"root","path":"system","name":"vdom","action":"","status":"success","http_status":200,"serial":"
FG101FTK20007637","version":"v7.6.6","build":3652}%
```

允许 URL 传递 token（不推荐）

如果确实需要通过 URL 参数传递 `access_token`（例如第三方平台暂不支持 Header 方式），可以通过以下命令启用：

> **警告**
>
> 启用此选项会将 API 密钥暴露在 URL 中，存在被日志记录、浏览器历史、代理服务器等截获的风险。建议优先使用 HTTP Header 方式传递 token。

```bash
config system global
    set rest-api-key-url-query enable
end
```

随后使用 URL 传递 token 的方式访问，可以正常返回 API 结果：

```bash
curl -vk 'https://192.168.100.99/api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63'
*   Trying 192.168.100.99:443 ...
* Connected to 192.168.100.99 (192.168.100.99) port 443
* ALPN: curl offers h2,http/1.1
* (304) (OUT), TLS handshake, Client hello (1):
* (304) (IN), TLS handshake, Server hello (2):
* (304) (IN), TLS handshake, Unknown (8):
* (304) (IN), TLS handshake, Certificate (11):
* (304) (IN), TLS handshake, CERT verify (15):
* (304) (IN), TLS handshake, Finished (20):
* (304) (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / AEAD-AES256-GCM-SHA384 / [blank] / UNDEF
* ALPN: server accepted h2
* Server certificate:
*  subject: C=US; ST=California; L=Sunnyvale; O=Fortinet Ltd.; OU=FortiGate; CN=FortiGate
*  start date: Mar 16 07:59:45 2026 GMT
*  expire date: Jun 18 07:59:45 2028 GMT
*  issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=FG101FTK20007637;
emailAddress=support@fortinet.com
*  SSL certificate verify ok.
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://192.168.100.99/api/v2/cmdb/system/vdom?
access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: 192.168.100.99]
* [HTTP/2] [1] [:path: /api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63]
* [HTTP/2] [1] [user-agent: curl/8.7.1]
* [HTTP/2] [1] [accept: */*]
> GET /api/v2/cmdb/system/vdom?access_token=785yfsnmc3NzjnHj0jNkh87fdNcb63 HTTP/2
> Host: 192.168.100.99
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/2 200
< content-security-policy: frame-ancestors 'self'; script-src 'self'; object-src 'none';
< strict-transport-security: max-age=63072000
< date: Thu, 26 Mar 2026 03:04:03 GMT
< etag: 27F6559DB07096435FA668AA55B19681FB124B76DA8D32FA77E125F613074EC1
< cache-control: no-cache, must-revalidate
< content-length: 377
< content-type: application/json
<
* Connection #0 to host 192.168.100.99 left intact
{"http_method":"GET","size":1,"limit_reached":false,"matched_count":1,"next_idx":0,"revision":"9b107552a7c5cc639bb9c77dc3c
f43b2","results":[{"name":"root","q_origin_key":"root","short-name":"root","vcluster-
id":0,"flag":0}],"vdom":"root","path":"system","name":"vdom","action":"","status":"success","http_status":200,"serial":"FG
101FTK20007637","version":"v7.6.6","build":3652}%
```

SNMP

- SNMP 介绍
- SNMP 公共 MIB 和私有 MIB
- SNMP 建议监控指标
- SNMP 的常用工具
- SNMP 的设置
- SNMP 与管理员可信主机
- SNMP 的 HA 独立管理
- 基于 VDOM 的 SNMP

SNMP 介绍

简单网络管理协议

简单网络管理协议（SNMP，Simple Network Management Protocol）构成了互联网工程工作小组（IETF，Internet Engineering Task Force）定义的 Internet 协议族的一部分。该协议能够支持网络管理系统，用以监测连接到网络上的设备是否有任何引起管理上关注的情况。它由一组网络管理的标准组成，包含一个应用层协议（application layer protocol）、数据库模式（database schema），和一组数据对象。

概论和基础观念

在典型的 SNMP 用法中，有许多系统被管理，而且是有一或多个系统在管理它们。每一个被管理的系统上有运行一个叫做代理者（agent）的软件组件，且透过 SNMP 对管理系统报告信息。

基本上，SNMP 代理者以变量呈现管理数据。管理系统透过 GET，GETNEXT 和 GETBULK 协议指令取回信息，或是代理者在没有被询问的情况下，使用 TRAP 或 INFORM 发送数据。管理系统也可以发送配置更新或控制的请求，透过 SET 协议指令达到主动管理系统的目的。配置和控制指令只有当网络基本结构需要改变的时候使用，而监控指令则通常是常态性的工作。

可透过 SNMP 访问的变量以层次结构的方式结合。这些分层和其他元数据（例如变量的类型和描述）以管理信息库（MIBs）的方式描述。

SNMP 协议

SNMP 第一版和 SMI 规格的数据类型 SNMP V1

SNMP 第一版 SMI 指定许多 SMI 规格的数据类型，它们被分为两大类：

1. 简单数据类型
2. 泛应用数据类型

SNMP 第二版和管理信息结构 SNMP V2

SNMP 第二版 SMI 在 RFC 2578 之中描述，它在 SNMP 第一版的 SMI 规格数据类型上进行增加和强化，例如位串（bit strings）、网络地址（network addresses）和计数器（counters）。

SNMP 协议在 OSI 模型的应用层（第七层）运作，在第一版中指定五种核心 PDU：

```
GET REQUEST
GET NEXT REQUEST
GET RESPONSE
SET REQUEST
TRAP
```

其他 PDU 在 SNMP 第二版加入，包含：

```
GETBULK REQUEST
INFORM
```

SNMP 第二版 SMI 也指定了信息模块来详细说明一群相关连的定义。有三种 SMI 信息模块：MIB 模块、回应状态、能力状态。

SNMP 第三版 SNMP V3

SNMP 第三版由 RFC 3411 - RFC 3418 定义，主要增加 SNMP 在安全性和远程配置方面的强化。SNMP 第三版提供重要的安全性功能：

1. 信息完整性：保证数据包在发送中没有被窜改。
2. 认证：检验信息来自正确的来源。
3. 数据包加密：避免被未授权的来源窥探。

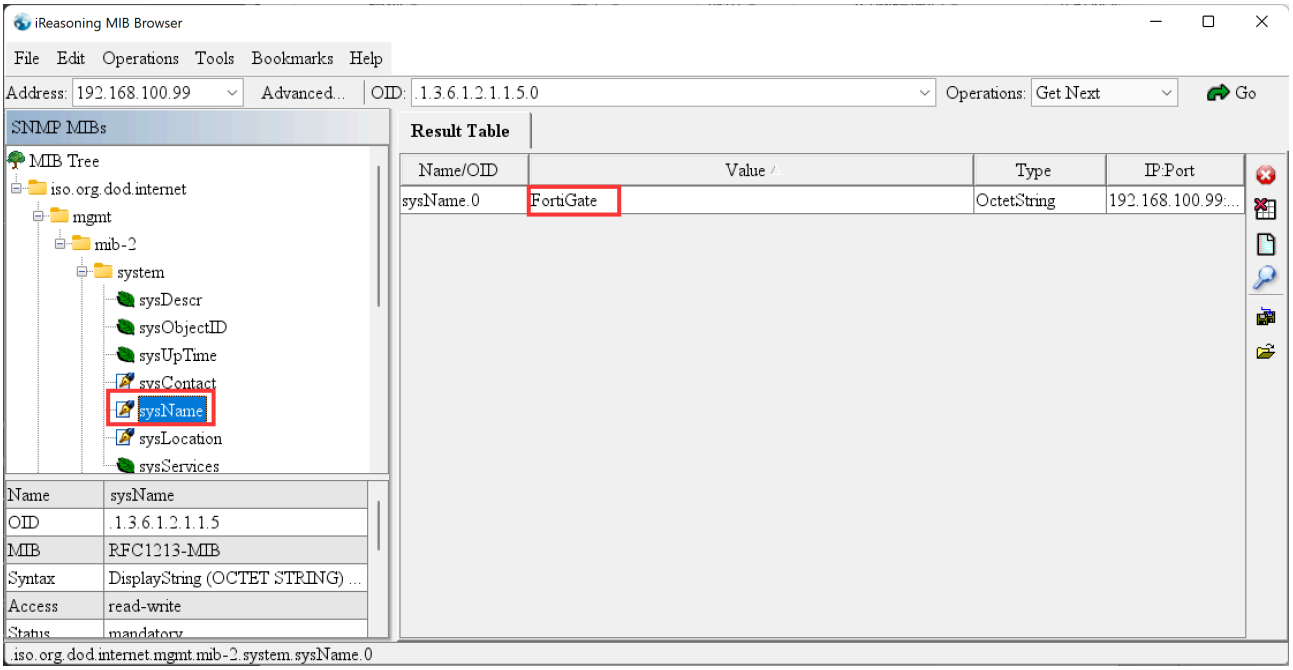
SNMP 公共 MIB 和私有 MIB

公共 MIB 库

公共 MIB 可以认为是大家共用使用的一些 MIB 查询节点，比如设备名称、设备运行时间、接口 MAC、接口 MTU、接口描述、接口收发报文数（接口流量）、路由表等等信息。

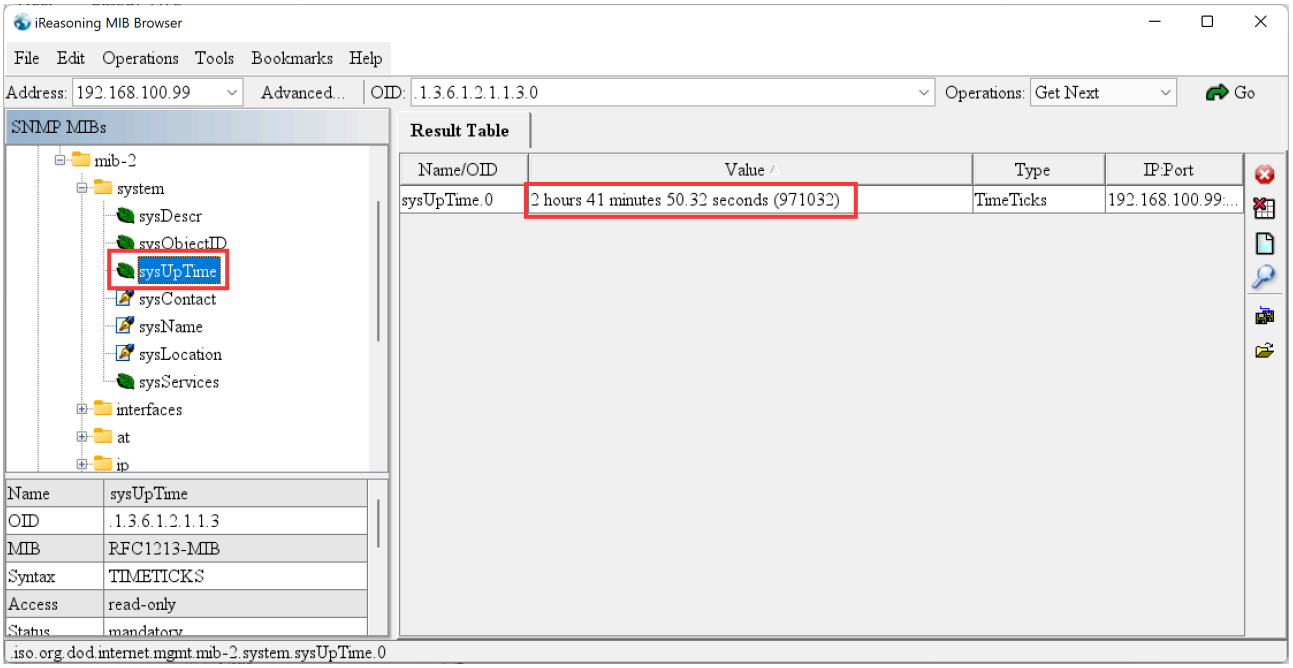
举例 1：设备名称

```
Name:       sysName
Type:       OBJECT-TYPE
OID:        1.3.6.1.2.1.1.5
```



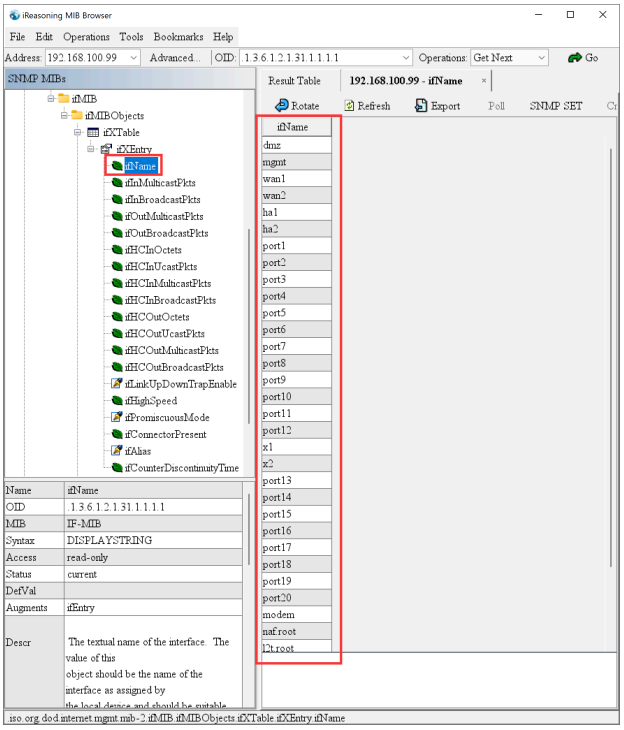
举例 2：设备启动时间

```
Name:       sysUpTime
Type:       OBJECT-TYPE
OID:        1.3.6.1.2.1.1.3
```



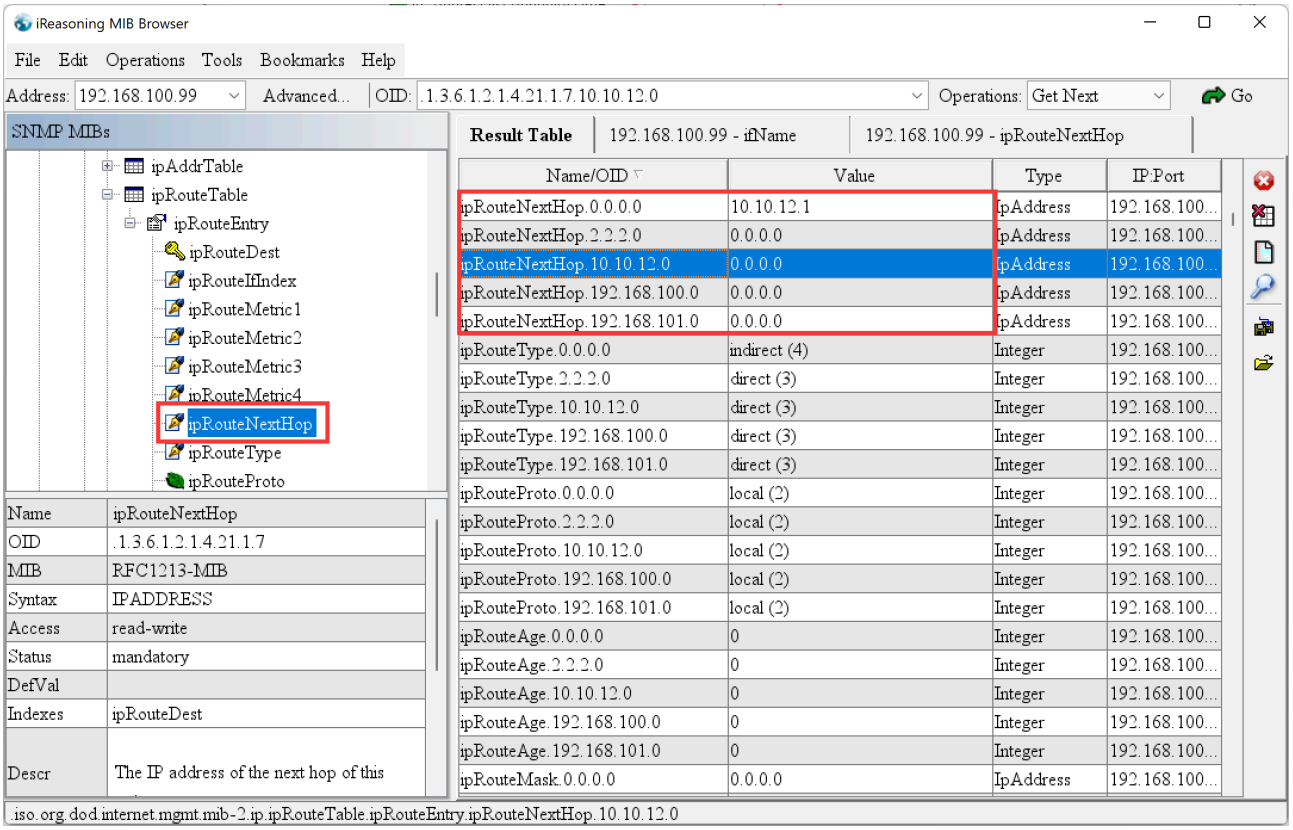
举例 3：设备接口名称、流量统计、接口描述、接口 Alias、等 OID

```
Name:        ifXTable
Type:        OBJECT-TYPE
OID:         1.3.6.1.2.1.31.1.1
Full path:       iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).ifMIB(31).ifMIBObjects(1).ifXTable(1)
Module:        IF-MIB

Parent:        ifMIBObjects
First child:       ifXEntry
Next sibling:       ifStackTable

Numerical syntax:       Sequence
Base syntax:       SEQUENCE OF IfXEntry
Composed syntax:       SEQUENCE OF IfXEntry
Status:        current
Max access:       not-accessible
Sequences:
       1: ifName - DisplayString(4 - octets)
       2: ifInMulticastPkts - Counter32(65 - counter (32 bit))
       3: ifInBroadcastPkts - Counter32(65 - counter (32 bit))
       4: ifOutMulticastPkts - Counter32(65 - counter (32 bit))
       5: ifOutBroadcastPkts - Counter32(65 - counter (32 bit))
       6: ifHCInOctets - Counter64(70 - counter (64 bit))
       7: ifHCInUcastPkts - Counter64(70 - counter (64 bit))
       8: ifHCInMulticastPkts - Counter64(70 - counter (64 bit))
       9: ifHCInBroadcastPkts - Counter64(70 - counter (64 bit))
       10: ifHCOutOctets - Counter64(70 - counter (64 bit))
       11: ifHCOutUcastPkts - Counter64(70 - counter (64 bit))
       12: ifHCOutMulticastPkts - Counter64(70 - counter (64 bit))
       13: ifHCOutBroadcastPkts - Counter64(70 - counter (64 bit))
       14: ifLinkUpDownTrapEnable - INTEGER(2 - integer (32 bit))
       15: ifHighSpeed - Gauge32(66 - gauge (32 bit))
       16: ifPromiscuousMode - TruthValue(2 - integer (32 bit))
       17: ifConnectorPresent - TruthValue(2 - integer (32 bit))
       18: ifAlias - DisplayString(4 - octets)
       19: ifCounterDiscontinuityTime - TimeStamp(67 - timeticks)
```

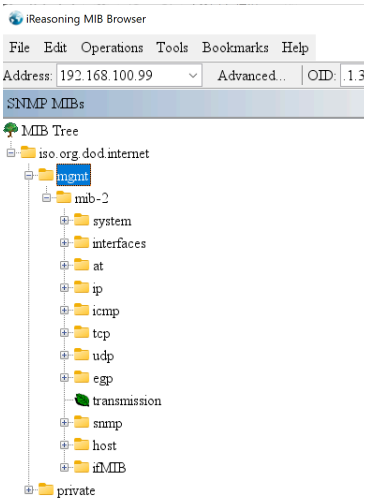


举例 4：设备路由表信息

```
Name:       ipRouteNextHop
Type:       OBJECT-TYPE
OID:        1.3.6.1.2.1.4.21.1.7
```



公共的 mib 库非常之多，其中有很多有用的 OID：



重要

具体可以通过以下链接查看到公共 MIB：

Net-SNMP Distributed MIBs：http://net-snmp.sourceforge.net/docs/mibs/

UCD-SNMP-MIB：http://net-snmp.sourceforge.net/docs/mibs/ucdavis.html

HOST-RESOURCES-MIB：http://net-snmp.sourceforge.net/docs/mibs/host.html

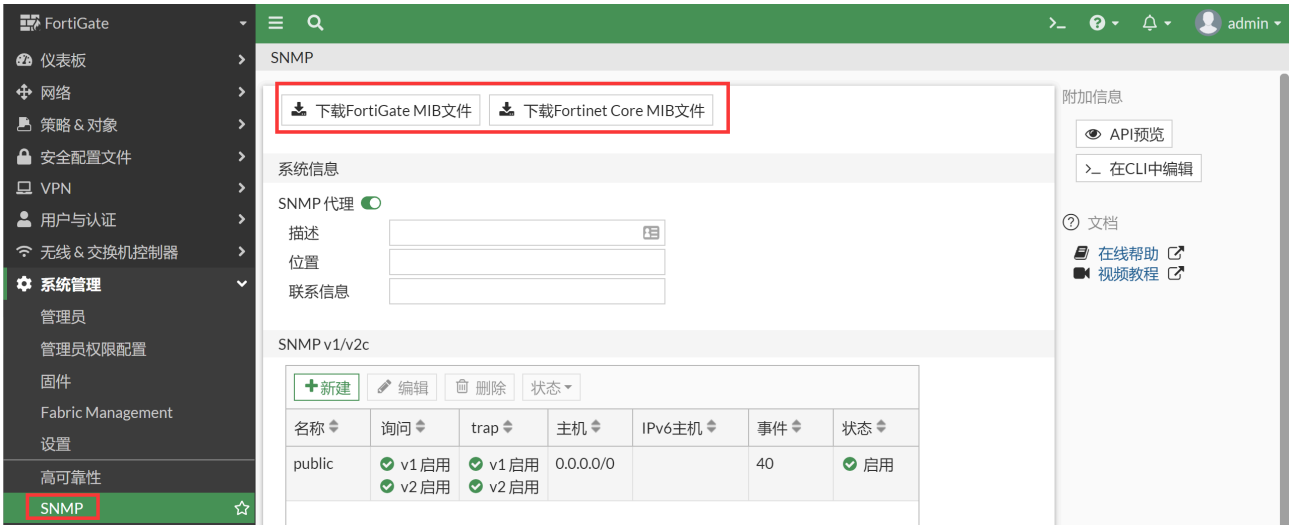
IF-MIB：http://net-snmp.sourceforge.net/docs/mibs/ifMIBObjects.html

mibDepot：http://www.mibdepot.com/index.shtml

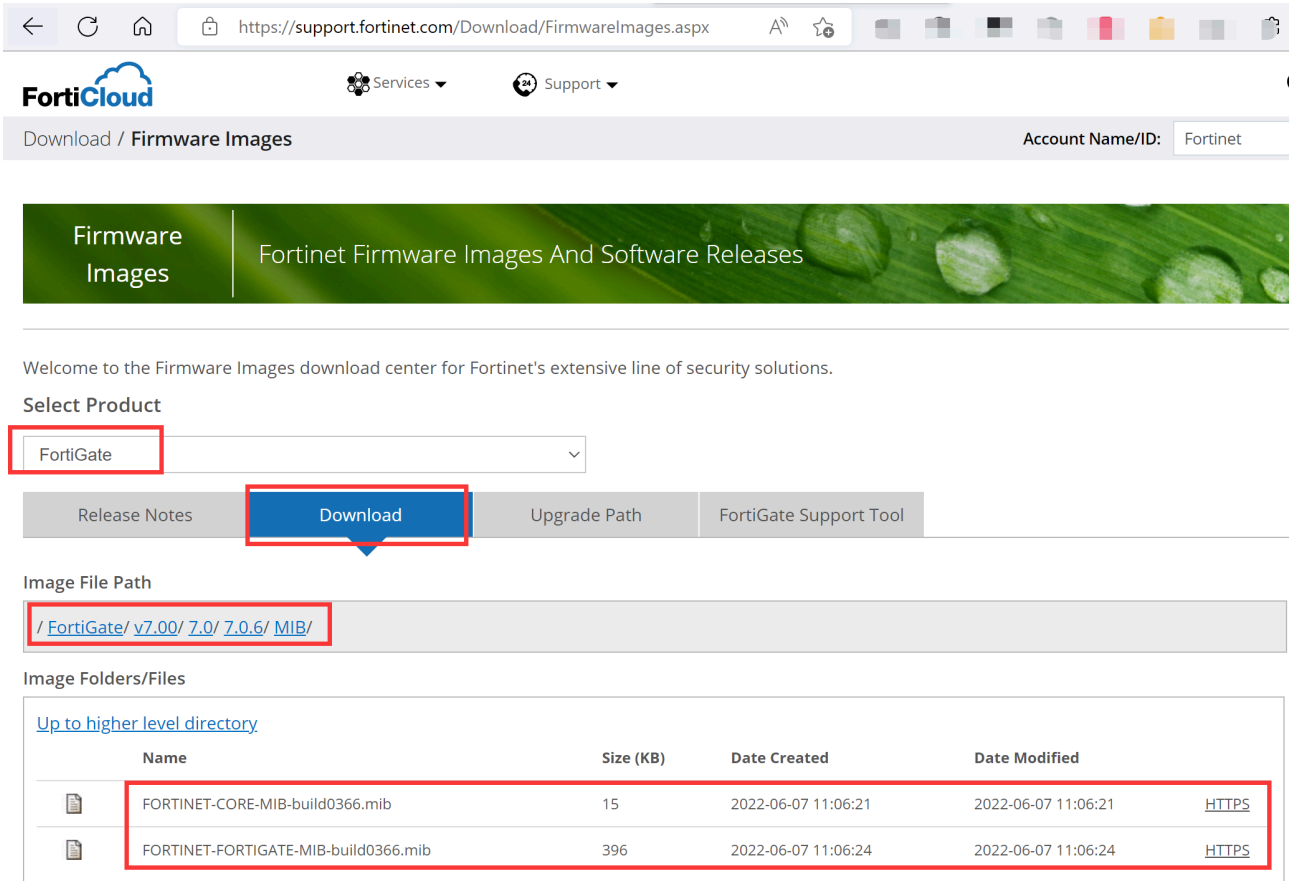
私有 MIB 库

FortiGate 的私有 mib 文件可以在 SNMP 配置界面处直接下载，或在 support 网站登录后下载。

FortiGate SNMP 配置界面处直接下载



SUPPORT 网站下载



注意事项

关于私有 mib 的 oid，Fortinet 的私有 number 是 12356，可通过以下链接获取私有厂家的 enterprise-numbers：
https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers ↗

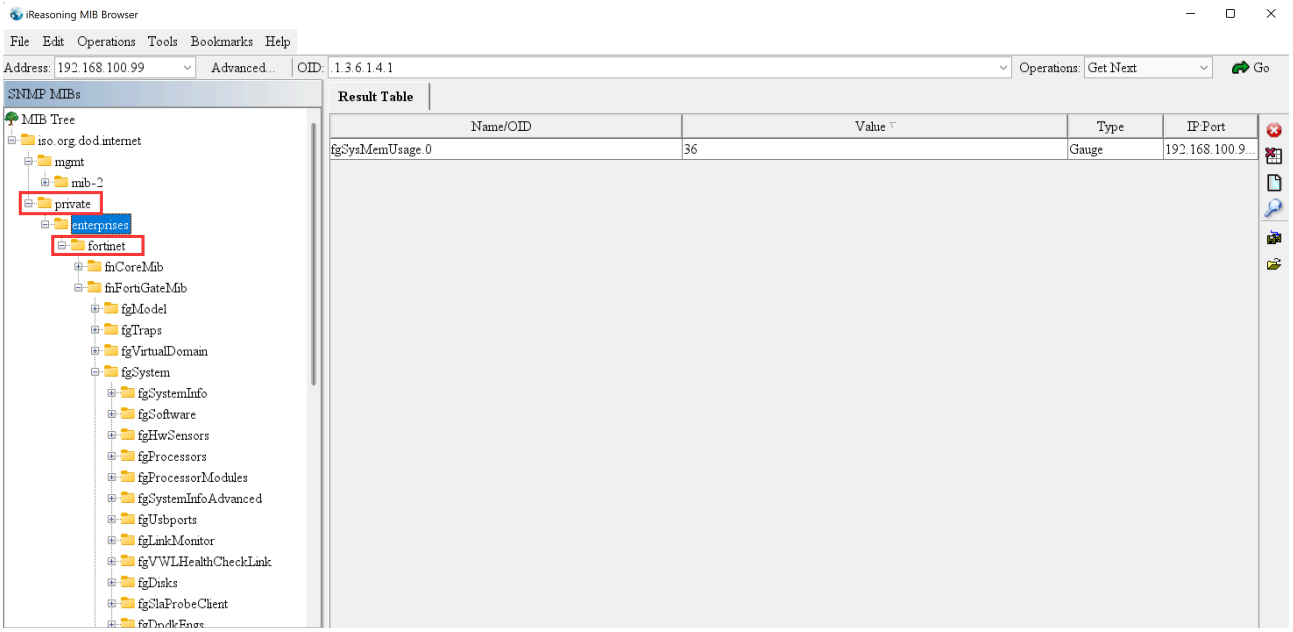
https://www.iana.org/assignments/enterprise-numbers/enterpris...

ad.thiers&atconsultancy.nl
12350
ATMEL Hellas, S.A.n
Effie Kirkou
ekirkou&patras.atmel.com
12351
Barking Cow
Jeff Hackert
jchack&barkingcow.com
12352
Central Data Services
Tyson Boellstorff
t.boellstorff&gwccnet.com
12353
Clariden Bank
Manuel Hilty
manuel.hilty&clariden.com
12354
Colbourn Associates Ltd
Charles Colbourn
charles&colbourn.net
12355
CTI PET Systems, Inc
Ellen Pearson
ellen.pearson&ctpspet.com
12356
Fortinet, Inc.
Michael Xie
mxie&fortinet.com
12357
Generic Media, Inc.
Michael Kellner
m&genericmedia.com
12358
Imagemedical
Gabor Horvath
ghorvath&imagemedical.com
12359
Indigo Tango Ltd.
Ewan Ferguson
ewan&indigotango.com
12360
Instituto Español de Informática y Derecho
Luis Rodriguez
lmr&ieid.org
12361
KGEx.com Co., Ltd
Josan Lin
josanl&kgex.com.tw
12362
Lombard Odier & Cie
Marc BERNEY
m.berney&lombardodier.com
12363
net-labs Systemhaus GmbH
Ulrich Eck
ueck&net-labs.de
12364
Newbreak LLC

SNMP 建议监控指标

FortiGate 私有 MIB

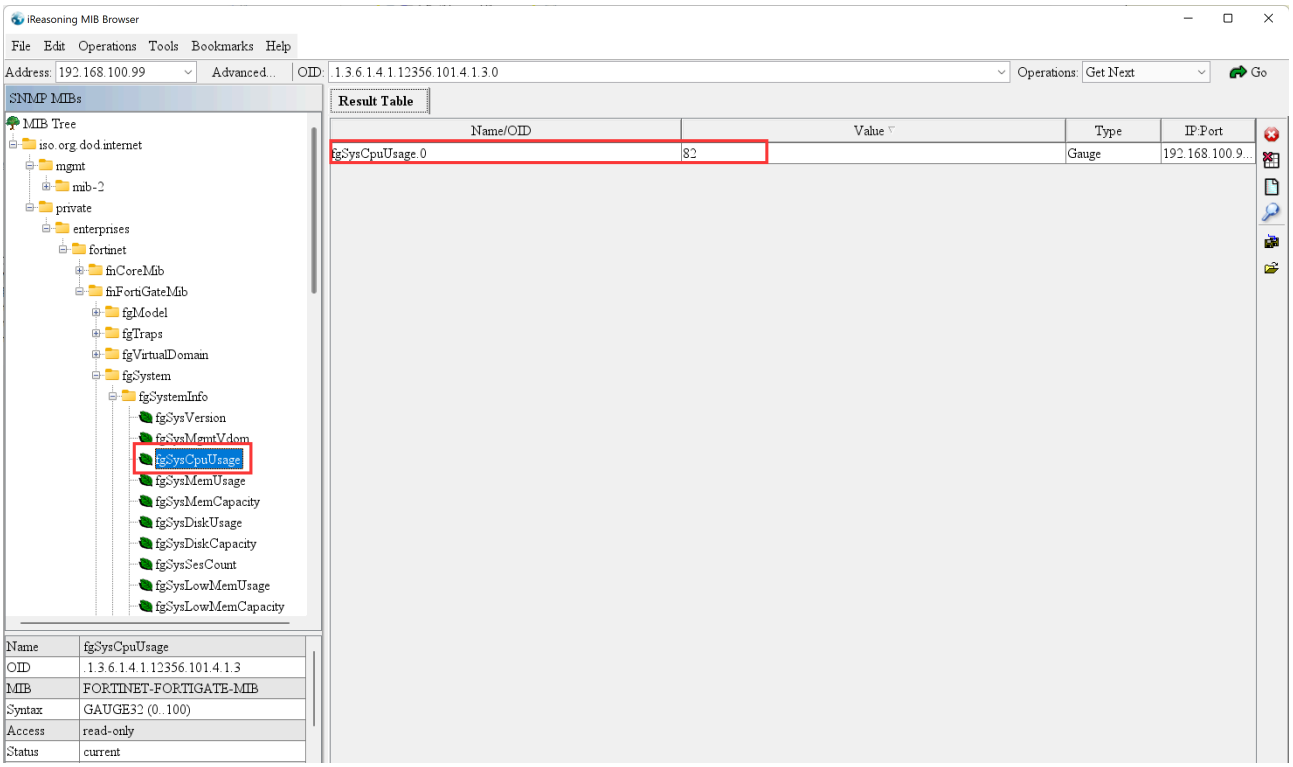
私有 MIB 库需要手工导入。



最常见的 SNMP 健康指标：FortiGate 的 CPU、MEM、新建、并发、接口流量等等。

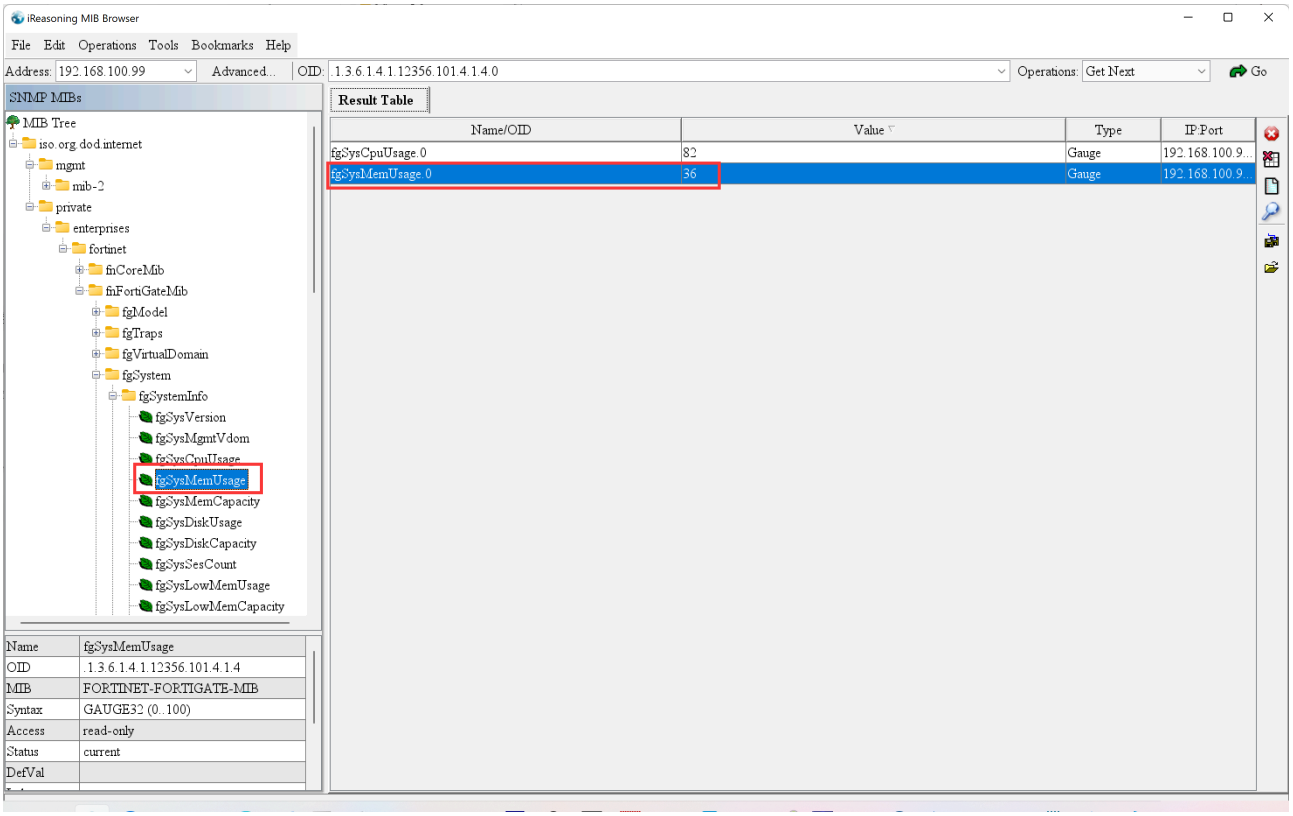
CPU OID

```
Name:        fgSysCpuUsage  OID:        1.3.6.1.4.1.12356.101.4.1.3.0
```



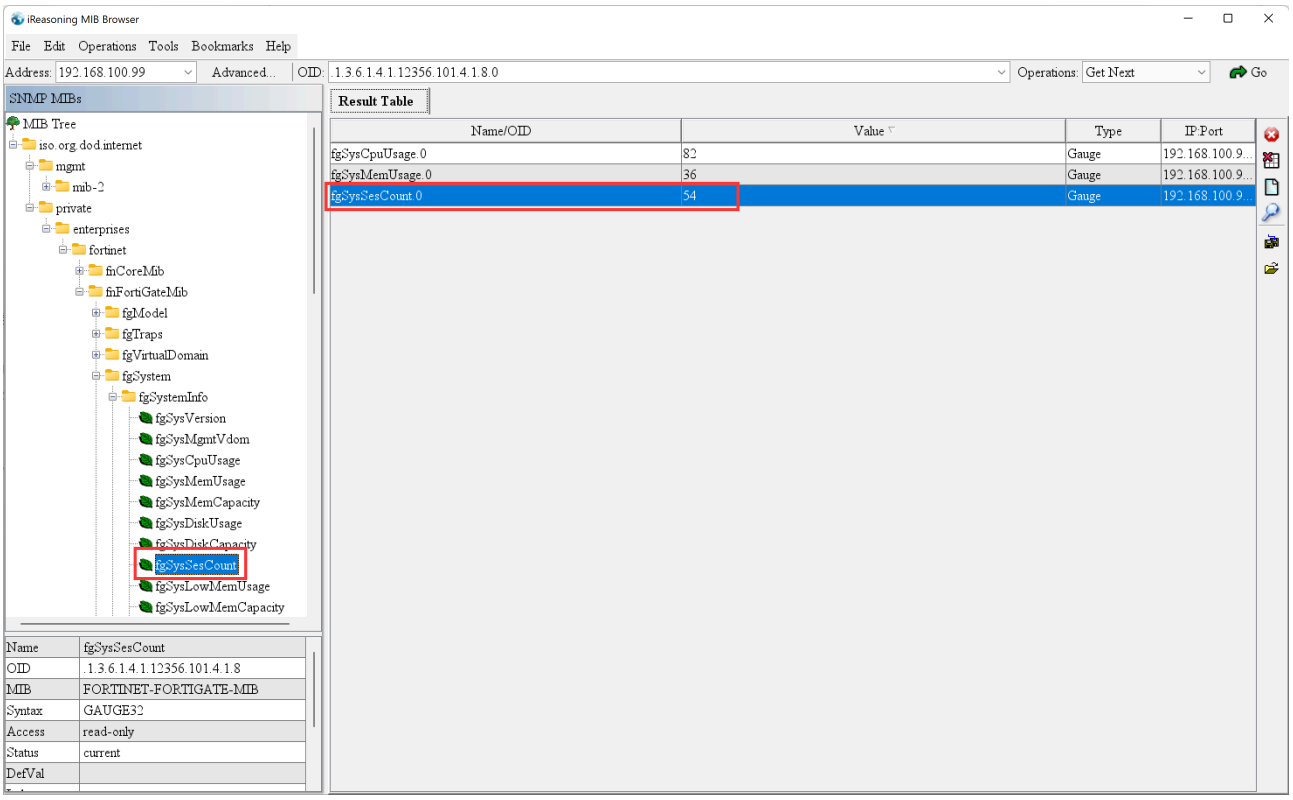
MEM OID

```
Name:        fgSysMemUsage  OID:        1.3.6.1.4.1.12356.101.4.1.4.0
```



并发会话数 OID

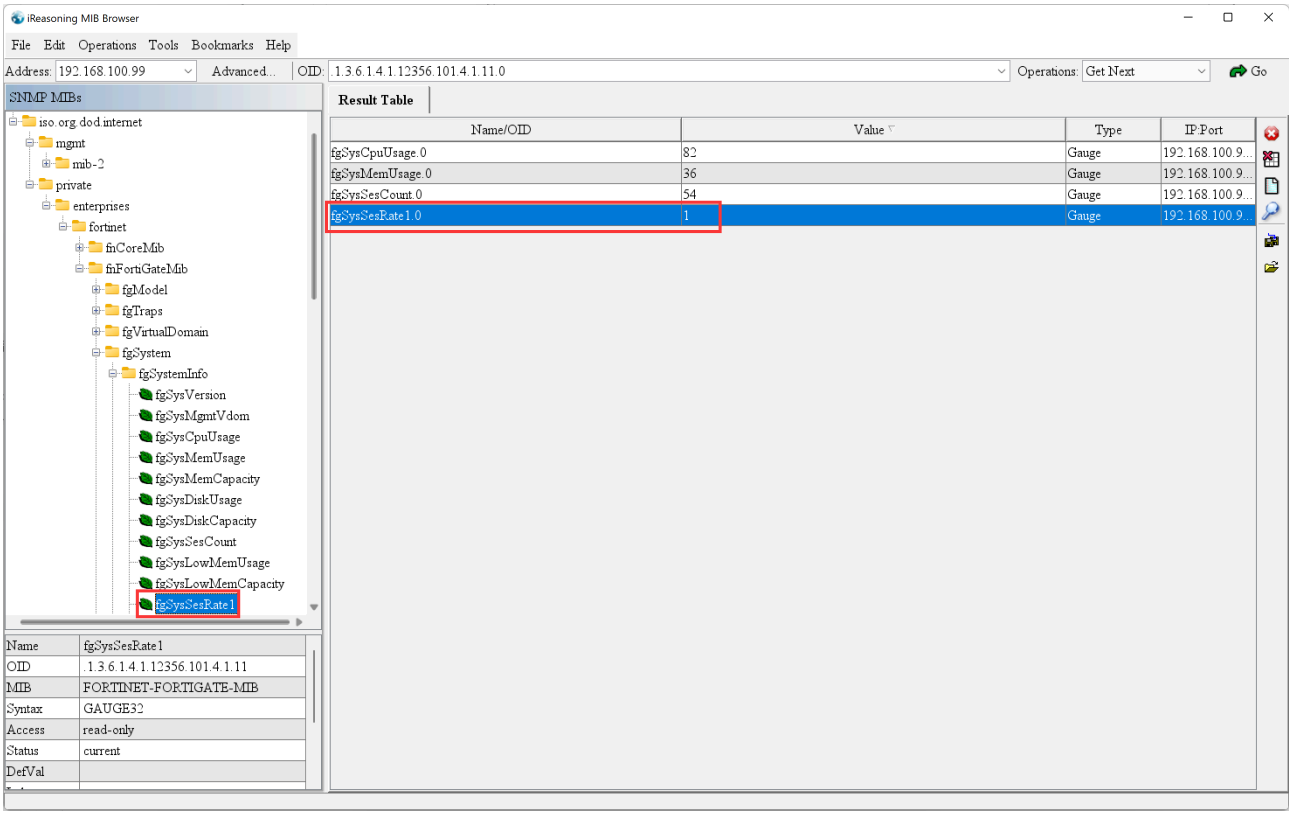
```
Name:        fgSysSesCount    OID:        1.3.6.1.4.1.12356.101.4.1.8.0
```



新建会话速率 OID

（1 分钟平均值）

```
Name:        fgSysSesRate1  OID:        1.3.6.1.4.1.12356.101.4.1.11.0
```



接口流量 OID

重要

这个部分使用的公共的 MIB 节点通过复合公式计算出来的接口带宽值，不是直接使用 MIB 值，通过这个计算方法来读取的接口流量。

复合参数	复合表达式	单位	备注
接口 bps（in）Inbound Throughput	吞吐量：间隔 t 秒执行 2 次获取 ifHCInOctets.x (.1.3.6.1.2.1.31.1.1.1.6.x) 的值，结果分别为 ifHCInOctets.x.time1 和 ifHCInOctets.x.time2。interface x 的 in_bps = (ifHCInOctets.x.time2 - ifHCInOctets.x.time1) * 8/t	bps	吞吐率（Inbound Utilization）＝接口吞吐量/接口最大吞吐量 = 60%
接口 bps (out) Outbound Throughput	吞吐量：间隔 t 秒执行 2 次获取 ifHCOutOctets.x(.1.3.6.1.2.1.31.1.1.1.10.x) 的值，结果分别为 ifHCOutOctets.x.time1 和 ifHCOutOctets.x.time2。interface x 的 out_bps = (ifHCOutOctets.x.time2 - ifHCOutOctets.x.time1) * 8/t	bps	吞吐率（Outbound Utilization）＝接口吞吐量/接口最大吞吐量 = 60%

```bash
Name:        ifHCInOctets
Type:        OBJECT-TYPE
OID:        1.3.6.1.2.1.31.1.1.1.6

Name:        ifHCOutOctets
Type:        OBJECT-TYPE
OID:        1.3.6.1.2.1.31.1.1.1.10

Type:        OBJECT-TYPE
```

```
OID:        1.3.6.1.2.1.31.1.1.1
Full path:        iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).ifMIB(31).ifMIBObjects(1).ifXTable(1).ifXEntry(1)
Sequences:
       1: ifName - DisplayString(4 - octets)
       2: ifInMulticastPkts - Counter32(65 - counter (32 bit))
       3: ifInBroadcastPkts - Counter32(65 - counter (32 bit))
       4: ifOutMulticastPkts - Counter32(65 - counter (32 bit))
       5: ifOutBroadcastPkts - Counter32(65 - counter (32 bit))
       6: ifHCInOctets - Counter64(70 - counter (64 bit))
       7: ifHCInUcastPkts - Counter64(70 - counter (64 bit))
       8: ifHCInMulticastPkts - Counter64(70 - counter (64 bit))
       9: ifHCInBroadcastPkts - Counter64(70 - counter (64 bit))
       10: ifHCOutOctets - Counter64(70 - counter (64 bit))
       11: ifHCOutUcastPkts - Counter64(70 - counter (64 bit))
       12: ifHCOutMulticastPkts - Counter64(70 - counter (64 bit))
       13: ifHCOutBroadcastPkts - Counter64(70 - counter (64 bit))
       14: ifLinkUpDownTrapEnable - INTEGER(2 - integer (32 bit))
       15: ifHighSpeed - Gauge32(66 - gauge (32 bit))
       16: ifPromiscuousMode - TruthValue(2 - integer (32 bit))
       17: ifConnectorPresent - TruthValue(2 - integer (32 bit))
       18: ifAlias - DisplayString(4 - octets)
       19: ifCounterDiscontinuityTime - TimeStamp(67 - timeticks)
```

FortiGate 防火墙监控指标参数

建议监控内容项

监控参数明细	Object	对应 OID 值	参数类别（单位）	SYSLOG 告警	SNMP 告警
系统运行时间	sysUpTime.0	.1.3.6.1.2.1.1.3.0	timeTicks (0.01s)		
当前接口状态（对于 VPN Tunnel 接口只能显示管理状态）	ifOperStatus.x	.1.3.6.1.2.1.2.2.1.8.x	INTEGER {up (1), down (2), testing (3), unknown (4), dormant (5), notPresent (6), lowerLayerDown (7)}	是	是
VPN Tunnel 二阶段状态	fgVpnTunEntStatus.x.y	.1.3.6.1.4.1.12356.101.12.2.2.1.20.x.y	INTEGER {down (1), up (2)}	是	是
每个 CPU 利用率	fgProcessorUsage.x	.1.3.6.1.4.1.12356.101.4.4.2.1.2.*	1-100（%）	是	是
多个 CPU 的总利用率	fgSysCpuUsage（percentage %）	.1.3.6.1.4.1.12356.101.4.1.3	1-100（%）	是	是
内存利用率	fgSysMemUsage	.1.3.6.1.4.1.12356.101.4.1.4	1-100（%）	是	是
并发连接数	fgSysSesCount.0	.1.3.6.1.4.1.12356.101.4.1.8.0	数字	是	是
每秒新建连接数（1 分钟平均）	fgSysSesRate1.0	.1.3.6.1.4.1.12356.101.4.1.11.0	数字	否	是
每秒新建连接数（10 分钟平均）	fgSysSesRate10.0	.1.3.6.1.4.1.12356.101.4.1.12.0	数字	否	是
每秒新建连接数（30 分钟平均）	fgSysSesRate30.0	.1.3.6.1.4.1.12356.101.4.1.13.0	数字	否	是
每秒新建连接数（60 分钟平均	fgSysSesRate60.0	.1.3.6.1.4.1.12356.101.4.1.14.0	数字	否	是
协议错误丢包	ipInHdrErrors.0	.1.3.6.1.2.1.4.4.0	数字，IP 包头错误个数，累加	否	是
接口流量（in）	ifHCInOctets.x	.1.3.6.1.2.1.31.1.1.1.6.x	数字，单位 bit	否	是
接口流量（out）	ifHCOutOctets.x	.1.3.6.1.2.1.31.1.1.1.10.x	数字，单位 bit	否	是
接口包转发数（in）	ifInUcastPkts.x	.1.3.6.1.2.1.2.2.1.11.x	数字：单播数据包数量	否	是
接口包转发数（out）	ifOutUcastPkts.x	.1.3.6.1.2.1.2.2.1.17.x	数字：单播数据包数量	否	是
接口丢包（in）	ifInDiscards.x	.1.3.6.1.2.1.2.2.1.13.x	数字：单播数据包数量	否	是
接口丢包（out）	ifOutDiscards.x	.1.3.6.1.2.1.2.2.1.19.x	数字：单播数据包数量	否	是
接口错包（in）	ifInErrors.x	.1.3.6.1.2.1.2.2.1.14.x	数字	否	是
接口错包（out）	ifOutErrors.x	.1.3.6.1.2.1.2.2.1.20.x	数字	否	是

系统状态监控项

监控参数明细	Object	对应 OID 值	参数类别（单位）	SYSLOG 告警	SNMP 告警
主机名称	sysName.0	.1.3.6.1.2.1.1.5.0	字符串		
设备序列号	fnSysSerial.0	.1.3.6.1.4.1.12356.100.1.1.1.0	字符串		
系统运行时间	sysUpTime.0	.1.3.6.1.2.1.1.3.0	timeTicks (0.01s)		
系统版本	fgsSysVersion	.1.3.6.1.4.1.12356.101.4.1.1.0	字符串		
系统 CPU 数量	fgProcModProcessorCount.1	.1.3.6.1.4.1.12356.101.4.5.3.1.5.1	正整数		
每个 CPU 利用率 (1 分钟)	fgProcessorUsage.x	.1.3.6.1.4.1.12356.101.4.4.2.1.2.X	1-100（%）	是	是
每个 CPU 利用率 (5 秒钟)	fgProcessorUsage5sec.x	.1.3.6.1.4.1.12356.101.4.4.2.1.3.x	1-100（%）		
多个 CPU 的总利用率	fgSysCpuUsage.0	.1.3.6.1.4.1.12356.101.4.1.3.0	1-100（%）	是	是
内存利用率	fgSysMemUsage.0	.1.3.6.1.4.1.12356.101.4.1.4.0	1-100（%）	是	是
总内存大小	fgSysMemCapacity.0	.1.3.6.1.4.1.12356.101.4.1.5.0	单位（kb）		
并发连接数	fgSysSesCount.0	.1.3.6.1.4.1.12356.101.4.1.8.0	数字	是	是
每秒新建连接数（1 分钟平均）	fgSysSesRate1.0	.1.3.6.1.4.1.12356.101.4.1.11.0	数字	否	是
每秒新建连接数（10 分钟平均）	fgSysSesRate10.0	.1.3.6.1.4.1.12356.101.4.1.12.0	数字	否	是
每秒新建连接数（30 分钟平均）	fgSysSesRate30.0	.1.3.6.1.4.1.12356.101.4.1.13.0	数字	否	是
每秒新建连接数（60 分钟平均）	fgSysSesRate60.0	.1.3.6.1.4.1.12356.101.4.1.14.0	数字	否	是
设备 License 状态 (table view)	FgLicContractEntry	.1.3.6.1.4.1.12356.101.4.6.3.1.2.1	字符串		
设备 Licence 名称	fgLicContractDesc	.1.3.6.1.4.1.12356.101.4.6.3.1.2.1.1	字符串		
设备 Licence 到期时间	fgLicContractExpiry	.1.3.6.1.4.1.12356.101.4.6.3.1.2.1.2	字符串		

接口状态监控项

监控参数明细	Object	对应 OID 值.x （x 为接口 ID）	参数类别（单位）	SYSLOG 告警	SNMP 告警
IfEntry (基础表)					
系统接口数量	ifNumber.0	.1.3.6.1.2.1.2.1.0	正整数		
接口 ID	ifIndex	.1.3.6.1.2.1.2.2.1.1	1.2.3….自然数		
接口 MTU	ifMtu.x	.1.3.6.1.2.1.2.2.1.4.x	正整数		
接口速率	ifSpeed.x	.1.3.6.1.2.1.2.2.1.5.x	正整数, 单位: bit		
接口管理状态	ifAdminStatus.x	.1.3.6.1.2.1.2.2.1.7.x	INTEGER {up (1), down (2)}		
接口工作状态（对于 VPN Tunnel 接口只能显示管理状态）	ifOperStatus.x	.1.3.6.1.2.1.2.2.1.8.x	INTEGER {up (1), down (2), testing (3), unknown (4), dormant (5), notPresent (6), lowerLayerDown (7)}	是	是
VPN Tunnel 二阶段状态	fgVpnTunEntStatus.x.y	.1.3.6.1.4.1.12356.101.12.2.2.1.20.x.y	INTEGER {down (1), up (2)}	是	是
接口流量（in）	ifInOctets.x	.1.3.6.1.2.1.2.2.1.10.x	数字 (32 位)，字节	否	是
接口单播包转发数（in）	ifInUcastPkts.x	.1.3.6.1.2.1.2.2.1.11.x	数字 (32 位)，个	否	是
接口非单播包转发数（in）	ifInNUcastPkts.x	.1.3.6.1.2.1.2.2.1.12.x	数字 (32 位)，个		
接口丢包（in）	ifInDiscards.x	.1.3.6.1.2.1.2.2.1.13.x	数字 (32 位)，个	否	是
接口错包（in）	ifInErrors.x	.1.3.6.1.2.1.2.2.1.14.x	数字 (32 位)，个	否	是
丢弃的未知协议包	ifInUnknownProtos.x	.1.3.6.1.2.1.2.2.1.15.x	数字 (32 位)，个	否	是
接口流量（out）	ifOutOctets.x	.1.3.6.1.2.1.2.2.1.16.x	数字 (32 位)，字节	否	是
接口单播包转发数（out）	ifOutUcastPkts.x	.1.3.6.1.2.1.2.2.1.17.x	数字 (32 位)，个	否	是
接口非单播包转发数（out）	ifOutNUcastPkts.x	.1.3.6.1.2.1.2.2.1.18.x	数字 (32 位)，个	否	是
接口丢包（out）	ifOutDiscards.x	.1.3.6.1.2.1.2.2.1.19.x	数字 (32 位)，个	否	是
接口错包（out）	ifOutErrors.x	.1.3.6.1.2.1.2.2.1.20.x	数字 (32 位)，个	否	是
IfXEntry (增强表)					
接口名字	ifName.x	.1.3.6.1.2.1.31.1.1.1.1.x	字符串		

接口多播包转发数（in）	ifInMulticastPkts.x	.1.3.6.1.2.1.31.1.1.1.2.x	数字 (32 位)，个	否	是
接口广播包转发数（in）	ifInBroadcastPkts.x	.1.3.6.1.2.1.31.1.1.1.3.x	数字 (32 位)，个	否	是
接口多播包转发数（out）	ifOutMulticastPkts.x	.1.3.6.1.2.1.31.1.1.1.4.x	数字 (32 位)，个	否	是
接口广播包转发数（out）	ifOutBroadcastPkts.x	.1.3.6.1.2.1.31.1.1.1.5.x	数字 (32 位)，个	否	是
接口流量（in）64 位	ifHCInOctets.x	.1.3.6.1.2.1.31.1.1.1.6.x	数字 (64 位)，字节	否	是
接口单播包转发数（in）64 位	ifHCInUcastPkts.x	.1.3.6.1.2.1.31.1.1.1.7.x	数字 (64 位)，个	否	是
接口多播包转发数（in）64 位	ifHCInMulticastPkts.x	.1.3.6.1.2.1.31.1.1.1.8.x	数字 (64 位)，个	否	是
接口广播包转发数（in）64 位	ifHCInBroadcastPkts.x	.1.3.6.1.2.1.31.1.1.1.9.x	数字 (64 位)，个	否	是
接口流量（out）64 位	ifHCOutOctets.x	.1.3.6.1.2.1.31.1.1.1.10.x	数字 (64 位)，字节	否	是
接口单播包转发数（out）64 位	ifHCOutUcastPkts.x	.1.3.6.1.2.1.31.1.1.1.11.x	数字 (64 位)，个	否	是
接口多播包转发数（out）64 位	ifHCOutMulticastPkts.x	.1.3.6.1.2.1.31.1.1.1.12.x	数字 (64 位)，个	否	是
接口广播包转发数（out）64 位	ifHCOutBroadcastPkts.x	.1.3.6.1.2.1.31.1.1.1.13.x	数字 (64 位)，个	否	是
接口 updown 状态变化是否发送 trap	ifLinkUpDownTrapEnable.x	.1.3.6.1.2.1.31.1.1.1.14.x	1：enable, 2: 其他	否	是
接口速率	ifHighSpeed.x	.1.3.6.1.2.1.31.1.1.1.15.x	正整数, 单位: Mbit		
复合参数					
接口 bps（in）Inbound Throughput	使用复合表达式	吞吐量：间隔 t 秒执行 2 次获取 ifHCInOctets.x (.1.3.6.1.2.1.31.1.1.1.6.x) 的值，结果分别为 ifHCInOctets.x.time1 和 ifHCInOctets.x.time2。interface x 的 in_bps = (ifHCInOctets.x.time2 - ifHCInOctets.x.time1) * 8/t	吞吐量：数字, bps	否	是
接口 bps（out）Outbound Throughput	使用复合表达式	吞吐量：间隔 t 秒执行 2 次获取 ifHCOutOctets.x 的值，结果分别为 ifHCOutOctets.x.time1 和 ifHCOutOctets.x.time2。interface x 的 out_bps = (ifHCOutOctets.x.time2 - ifHCOutOctets.x.time1) * 8/t	吞吐量：数字, bps	否	是

TCP/IP/策略监控项

监控参数明细	Object	对应 OID 值	参数类别（单位）	SYSLOG 告警	SNMP 告警
IP					
端口接收到的 ip 报文总数	ipInReceives.0	.1.3.6.1.2.1.4.3.0	数字（32 位），个	否	是
IP 头部错误丢包	ipInHdrErrors.0	.1.3.6.1.2.1.4.4.0	数字（32 位），个	否	是
IP 目标地址错误丢弃包	ipInAddrErrors.0	.1.3.6.1.2.1.4.5.0	数字（32 位），个	否	是
未知协议数据包	ipInUnknownProtos	.1.3.6.1.2.1.4.7.0	数字（32 位），个	否	是
被丢弃的 IP 包（如缓冲区不够）	ipInDiscards.0	.1.3.6.1.2.1.4.8.0	数字（32 位），个	否	是
被转发包的 IP 包，包括 ICMP	ipInDelivers.0	.1.3.6.1.2.1.4.9.0	数字（32 位），个	否	是
需重组数据包	ipReasmReqds.0	.1.3.6.1.2.1.4.14.0	数字（32 位），个	否	是
重组成功数据包	ipReasmOKs.0	.1.3.6.1.2.1.4.15.0	数字（32 位），个	否	是
重组失败数据包	ipReasmFails.0	.1.3.6.1.2.1.4.16.0	数字（32 位），个	否	是
分片成功数据包	ipFragOKs.0	.1.3.6.1.2.1.4.17.0	数字（32 位），个	否	是
分片失败数据包	ipFragFails.0	.1.3.6.1.2.1.4.18.0	数字（32 位），个	否	是
TCP					
TCP 数据分组（in）	tcpInSegs.0	.1.3.6.1.2.1.6.10.0	数字（32 位），个	否	是
TCP 数据分组（out）	tcpOutSegs.0	.1.3.6.1.2.1.6.11.0	数字（32 位），个	否	是
重传的数据分组	tcpRetransSegs.0	.1.3.6.1.2.1.6.12.0	数字（32 位），个	否	是
接收到的 TCP 错包	tcpInErrs.0	.1.3.6.1.2.1.6.14.0	数字（32 位），个	否	是
发送的 RST 数据包	tcpOutRsts.0	.1.3.6.1.2.1.6.15.0	数字（32 位），个	否	是
Policy					
策略命中包数	fgFwPolPktCount.1.x（x 为策略 ID）	.1.3.6.1.4.1.12356.101.5.1.2.1.1.2.1.x	数字（32 位），次		
策略命中字节数	fgFwPolByteCount.1.x（x 为策略 ID）	.1.3.6.1.4.1.12356.101.5.1.2.1.1.3.x	数字（32 位），byte		

SNMP 的常用工具

Net-SNMP

网址链接：http://www.net-snmp.org/ ↗

Download：http://www.net-snmp.org/download.html ↗

1. CentOS 7 安装 Net-SNMP 和 SNMP V2：

```
[Wed Aug 05 10:07:33 root@centos7~#yum install -y net-snmp*
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
 * base: mirror2.totbb.net
 * extras: mirror.hostlink.com.hk
 * updates: mirror01.idc.hinet.net
base                                                                                                | 3.6
kB  00:00:00
extras                                                                                              | 2.9
kB  00:00:00
updates                                                                                             | 2.9
kB  00:00:00
(1/2): extras/7/x86_64/primary_db                                                                   | 205
kB  00:00:00
(2/2): updates/7/x86_64/primary_db                                                                  |3.7 MB
00:00:11
......

Installed:
  net-snmp.x86_64 1:5.7.2-48.el7_8.1           net-snmp-agent-libs.x86_64 1:5.7.2-48.el7_8.1  net-snmp-devel.x86_64
1:5.7.2-48.el7_8.1
  net-snmp-gui.x86_64 1:5.7.2-48.el7_8.1       net-snmp-perl.x86_64 1:5.7.2-48.el7_8.1        net-snmp-python.x86_64
1:5.7.2-48.el7_8.1
  net-snmp-sysvinit.x86_64 1:5.7.2-48.el7_8.1  net-snmp-utils.x86_64 1:5.7.2-48.el7_8.1

Dependency Installed:
  elfutils-devel.x86_64 0:0.176-4.el7                         elfutils-libelf-devel.x86_64 0:0.176-4.el7
  libdb-devel.x86_64 0:5.3.21-25.el7                          lm_sensors-devel.x86_64 0:3.4.0-8.20160601gitf9185e5.el7
  perl-ExtUtils-Install.noarch 0:1.58-295.el7                 perl-ExtUtils-MakeMaker.noarch 0:6.68-3.el7
  perl-ExtUtils-Manifest.noarch 0:1.61-244.el7                perl-ExtUtils-ParseXS.noarch 1:3.18-3.el7
  perl-Tk.x86_64 0:804.030-6.el7                              perl-devel.x86_64 4:5.16.3-295.el7
  popt-devel.x86_64 0:1.13-16.el7                             rpm-devel.x86_64 0:4.11.3-43.el7
  systemtap-sdt-devel.x86_64 0:4.0-11.el7                     tcp_wrappers-devel.x86_64 0:7.6-77.el7

Updated:
  net-snmp-libs.x86_64 1:5.7.2-48.el7_8.1

Dependency Updated:
  elfutils.x86_64 0:0.176-4.el7            elfutils-libelf.x86_64 0:0.176-4.el7            elfutils-libs.x86_64 0:0.176-
4.el7

Complete!
[Thu Aug 06 09:56:37 root@centos7~#
[Thu Aug 06 10:09:13 root@centos7~# snmpd -v

NET-SNMP version:  5.7.2
Web:               http://www.net-snmp.org/
Email:             net-snmp-coders@lists.sourceforge.net

[Thu Aug 06 10:09:18 root@centos7~#
[Thu Aug 06 10:09:46 root@centos7~#systemctl start snmpd.service
[Thu Aug 06 10:09:49 root@centos7~#
[Thu Aug 06 10:10:17 root@centos7~#snmpwalk -v 1 -c public 127.0.0.1 .1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Linux centos7.lab.fortinet.com 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC
2019 x86_64
[Thu Aug 06 10:10:19 root@centos7~#
[Thu Aug 06 10:24:55 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.90.3 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: HUB2-ShangHai
[Thu Aug 06 10:24:59 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.90.3 .1.3.6.1.4.1.12356.100.1.1.1.0
```

```
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM04TM20003482"
[Thu Aug 06 10:25:01 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.90.3 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v6.4.1,build1637,200604 (GA)"
[Thu Aug 06 10:25:21 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.90.3 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301149616) 34 days, 20:31:36.16
[Thu Aug 06 10:25:55 root@centos7~#
```

2. Net-SNMP SNMP V3

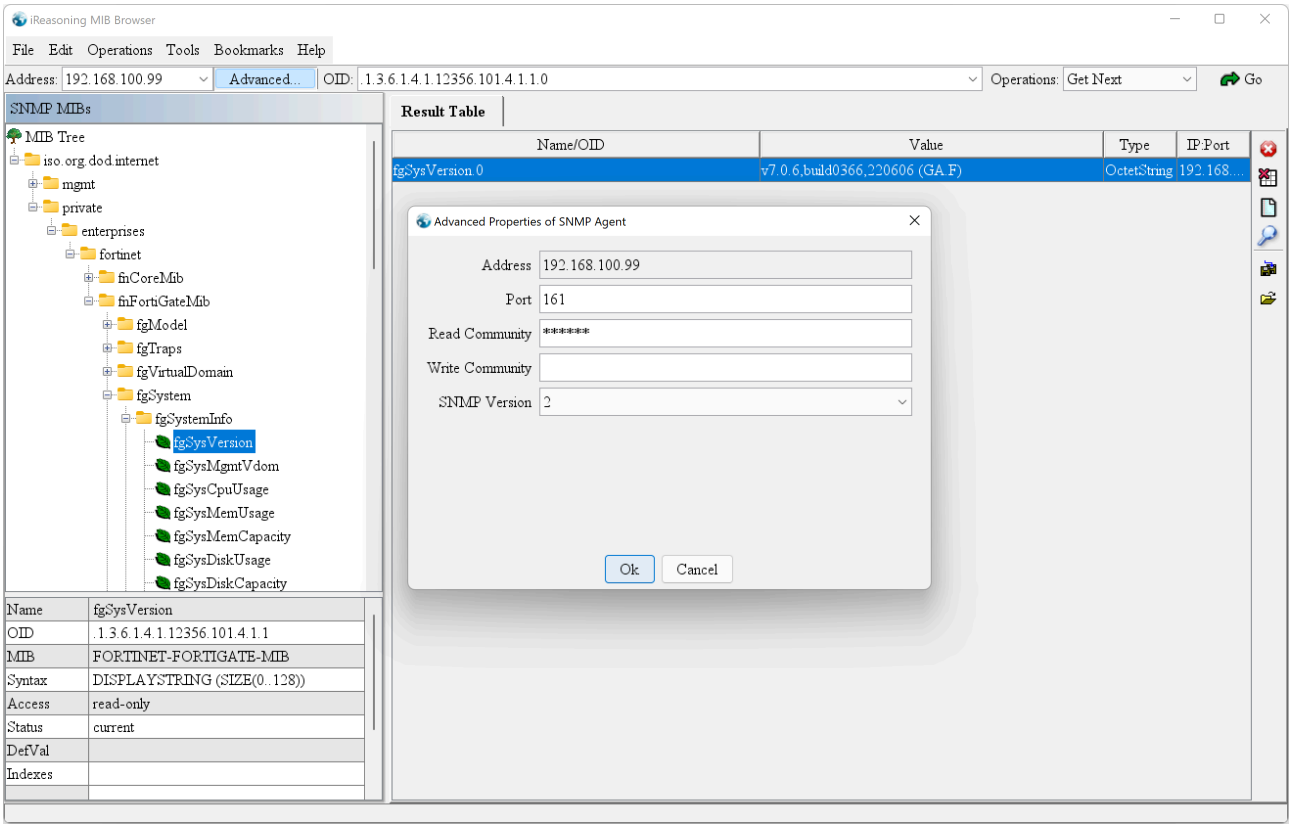
```
[Thu Aug 06 11:56:06 root@centos7~#service snmpd  stop
Stopping snmpd (via systemctl):                            [  OK  ]
[Thu Aug 06 11:56:20 root@centos7~#
[Thu Aug 06 12:00:52 root@centos7~#
[Thu Aug 06 12:01:01 root@centos7~#net-snmp-config --create-snmpv3-user -ro -a SHA -A forti3389 -x AES -X forti5566
fortinet
adding the following line to /var/lib/net-snmp/snmpd.conf:
   createUser fortinet SHA "forti3389" AES forti5566
adding the following line to /etc/snmp/snmpd.conf:
   rouser fortinet
[Thu Aug 06 12:01:09 root@centos7~#
[Thu Aug 06 12:01:10 root@centos7~#chkconfig snmpd on
Note: Forwarding request to 'systemctl enable snmpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/snmpd.service to /usr/lib/systemd/system/snmpd.service.
[Thu Aug 06 12:01:30 root@centos7~#service snmpd start
Starting snmpd (via systemctl):                            [  OK  ]
[Thu Aug 06 12:01:36 root@centos7~#
[Thu Aug 06 12:06:29 root@centos7~#
[Thu Aug 06 12:06:38 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.90.3
SNMPv2-MIB::sysDescr.0 = STRING: Fortinet_BeiJing_LAB_FGT-KVM
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.12356.101.1.60
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301754102) 34 days, 22:12:21.02
SNMPv2-MIB::sysContact.0 = STRING: kmliu@fortinet.com
SNMPv2-MIB::sysName.0 = STRING: HUB2-ShangHai
SNMPv2-MIB::sysLocation.0 = STRING: China_BeiJing_Lab
SNMPv2-MIB::sysServices.0 = INTEGER: 78
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
SNMPv2-MIB::sysORIndex.1 = INTEGER: 1
SNMPv2-MIB::sysORID.1 = OID: SNMPv2-SMI::zeroDotZero.0
SNMPv2-MIB::sysORDescr.1 = STRING:
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (0) 0:00:00.00
IF-MIB::ifNumber.0 = INTEGER: 19
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
^C
[Thu Aug 06 12:06:41 root@centos7~#
[Thu Aug 06 12:06:45 root@centos7~#
[Thu Aug 06 12:10:23 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.90.3 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM04TM20003482"
[Thu Aug 06 12:10:45 root@centos7~#
[Thu Aug 06 12:10:46 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.90.3 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301780209) 34 days, 22:16:42.09
[Thu Aug 06 12:11:00 root@centos7~#
[Thu Aug 06 12:11:12 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.90.3 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v6.4.1,build1637,200604 (GA)"
[Thu Aug 06 12:11:19 root@centos7~#
```

Net-SNMP v3 参考链接：https://www.thegeekdiary.com/centos-rhel-6-install-and-configure-snmpv3/ ↗

iReasoning MIB browser（提供免费版本）

网址链接：https://www.ireasoning.com/ ↗

Download：https://www.ireasoning.com/downloadmibbrowserfree.php ↗



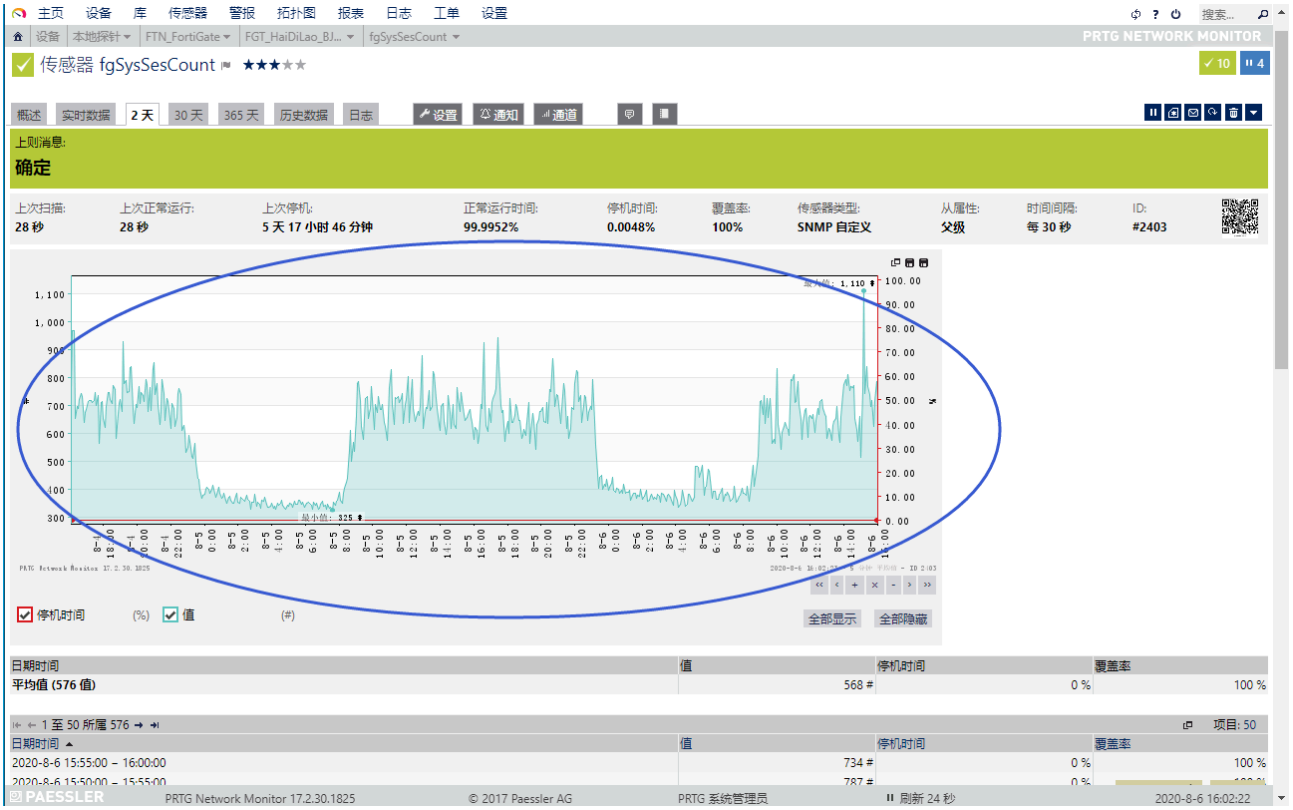
PRTG（提供 100 个传感器的免费版本）

网址链接：https://www.cn.paessler.com/

Download：https://www.cn.paessler.com/download/prtg-download/


不安全 | 192.168.91.9/device.htm?id=2393&tabid=1
主页 设备 库 传感器 警报 拓扑图 报表 日志 工单 设置
PRTG NETWORK MONITOR
设备 FGT_HaiDiLao_BJ_VPN070
概述 2 天 30 天 365 天 警报 系统信息 日志 设置 通知
状态: 确定
传感器: 9 1 (共 10)
DNS/IP: 36.112.117.34
从属性: 父级
默认时间间隔: 每 30 秒
上次自动发现: (从不)
上次推荐: 59 天 前
ID: #2393
销售点 传感器 状态 消息 图表 优先级
1. (001) Unicom Traffic 正常运行 确定 流量合计 410 kbit/秒
2. (006) Lan_Traffic 正常运行 确定 流量合计 449 kbit/秒
3. (007) MD_TO_AWS_NX Traffic 正常运行 确定 流量合计 0 kbit/秒
4. (008) IDC_Unicom Traffic 异常 2.10 kbit/秒 (入站流量) 的 1 小时 时间间隔平均值为异常 低 用于本周此小时 流量合计 18 kbit/秒
(014) MD_TO_Ali Traffic 正常运行 确定 流量合计 165 kbit/秒
Ping 正常运行 确定 Ping 时间 97 毫秒
7. MEM-fgSysMemUsage 正常运行 确定 值 24 %
8. CPU-fgSysCpuUsage 正常运行 确定 值 0 %
9. fgSysSesCount 正常运行 确定 值 780 #
10. fgSysSesRate 正常运行 确定 值 4 #
添加传感器 推荐立即
建议的传感器
优先级 传感器 合计 传感器 链接
当前没有任何传感器推荐。请点击"立即推荐"，以分析此设备。
这是什么?
PRTG 可以检查您的设备，以推荐有用的传感器类型。添加这些传感器，可更好且更详细地了解此设备的未来状态。
2 天
30 天
365 天
警报
响应时间指数
CPU 负载指数
流量指数


不安全 | 192.168.91.9/sensor.htm?id=2395&tabid=4
主页 设备 库 传感器 警报 拓扑图 报表 日志 工单 设置
PRTG NETWORK MONITOR
传感器 (014) MD_TO_Ali Traffic
概述 实时数据 2 天 30 天 365 天 历史数据 日志 设置 通知 通道
上则消息:
确定
上次扫描: 7 秒
上次正常运行: 7 秒
上次停机: 18 天
正常运行时间: 99.9905%
停机时间: 0.0095%
覆盖率: 100%
传感器类型: SNMP 流量 64bit
从属性: 父级
时间间隔: 每 30 秒
ID: #2395
最大值: 46.8 kbit/秒
最小值: 1.2 kbit/秒
kbit/秒
%
停机时间 (%) 流量合计 (kbit/秒) 入站流量 (kbit/秒) 出站流量 (kbit/秒)
全部显示 全部隐藏
日期时间 流量合计 (卷) 流量合计 (速度) 入站流量 (卷) 入站流量 (速度) 出站流量 (卷) 出站流量 (速度) 停机时间 覆盖率
总和 (699 值) 3,774,625 KB 2,585,996 KB 1,188,629 KB
平均值 (699 值) 5,400 KB 12 kbit/秒 3,700 KB 8.43 kbit/秒 1,700 KB 3.88 kbit/秒 <1 % 97 %
1 至 50 所属 720
项目: 50
日期时间 流量合计 (卷) 流量合计 (速度) 入站流量 (卷) 入站流量 (速度) 出站流量 (卷) 出站流量 (速度) 停机时间 覆盖率
2020-8-6 15:00:00 - 16:00:00 3.712 KB 8.45 kbit/秒 2.476 KB 5.64 kbit/秒 1.236 KB 2.81 kbit/秒 0 % 100 %





添加传感器到设备 FGT_HaiDiLao_BJ_VPN070 [36.112.117.34] (第 1 步，共 2 步)

搜索

71 正在匹配传感器类型

监控什么?

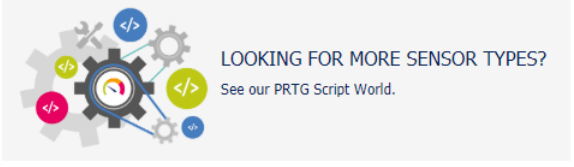
- 可用性/正常运行时间
- 带宽/流量
- 速度/性能
- CPU 使用情况
- 磁盘使用情况
- 内存使用情况
- 硬件参数
- 网络基础设施
- 自定义传感器

目标系统类型?

- Windows
- Linux/MacOS
- 虚拟化操作系统
- 文件服务器
- 电子邮件服务器
- 数据库
- 云服务

使用的技术?

- Ping
- **SNMP**
- WMI
- 性能计数器
- HTTP
- SSH
- 数据包嗅探
- NetFlow、sFlow、jFlow
- PowerShell
- 推送消息接收程序
- PRTG Cloud

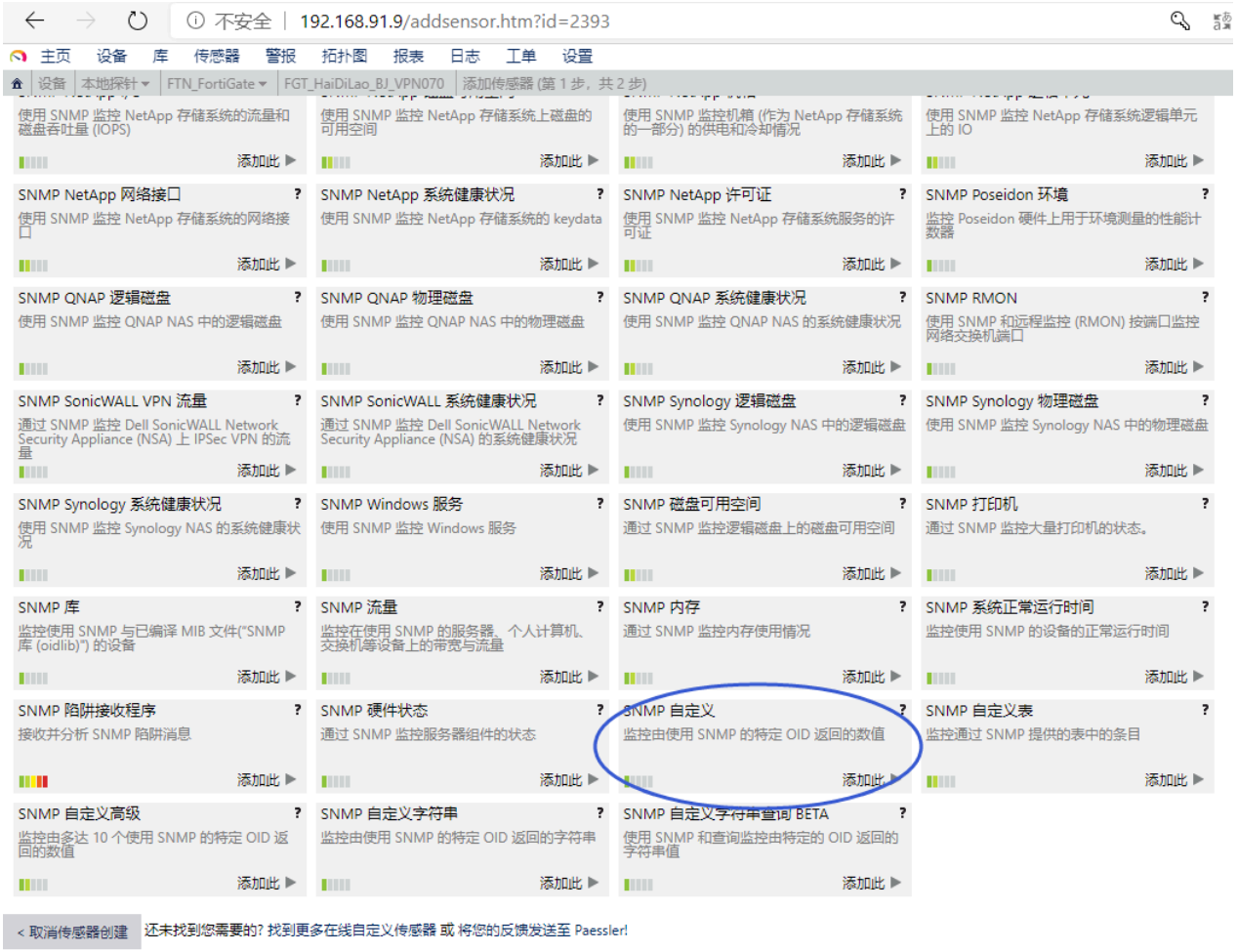


最常用的传感器类型

SNMP 流量 ?

监控在使用 SNMP 的服务器、个人计算机、交换机等设备上的带宽与流量

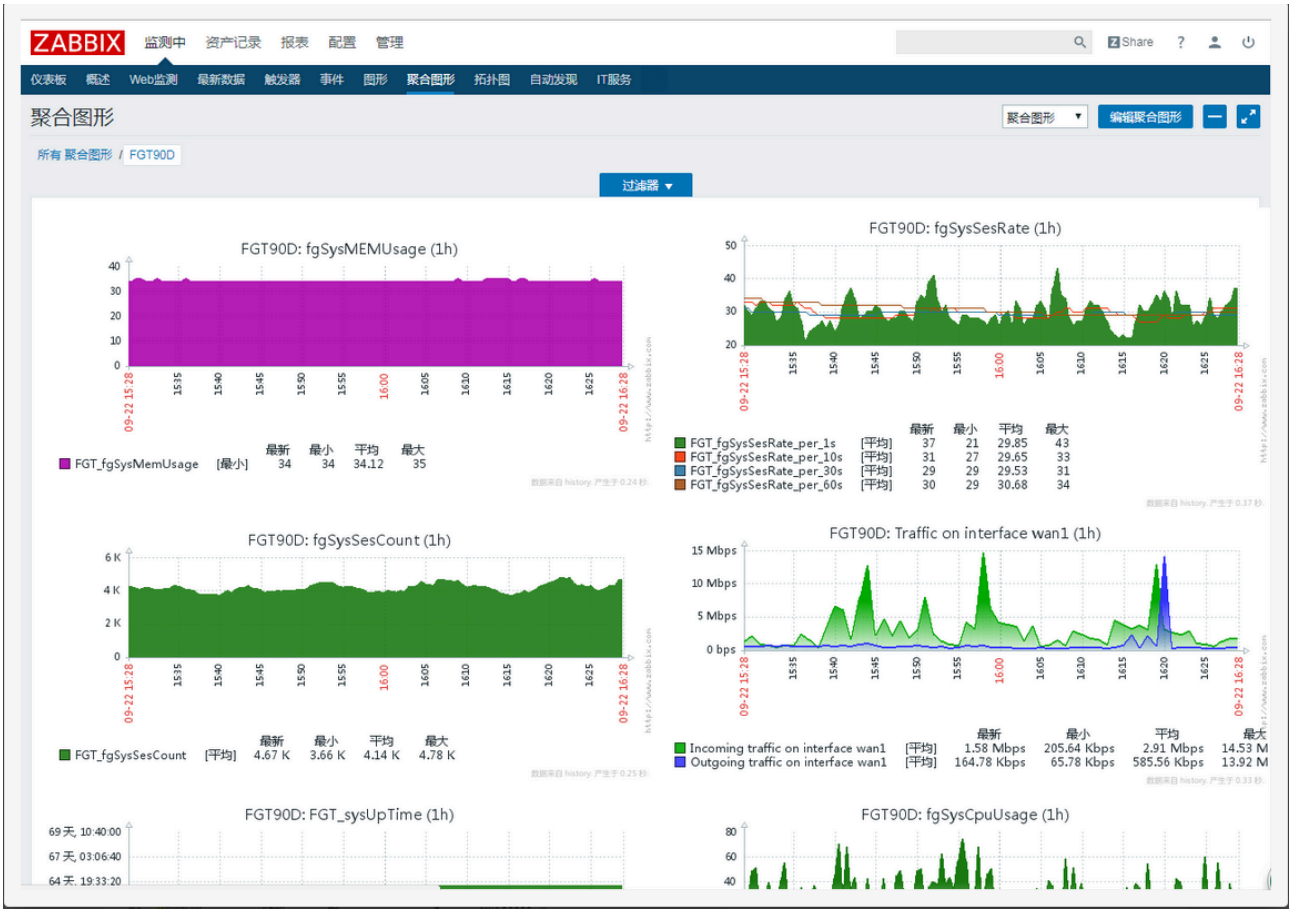
添加此 ▶



Zabbix（开源免费）

网址链接：https://www.zabbix.com/cn ↗

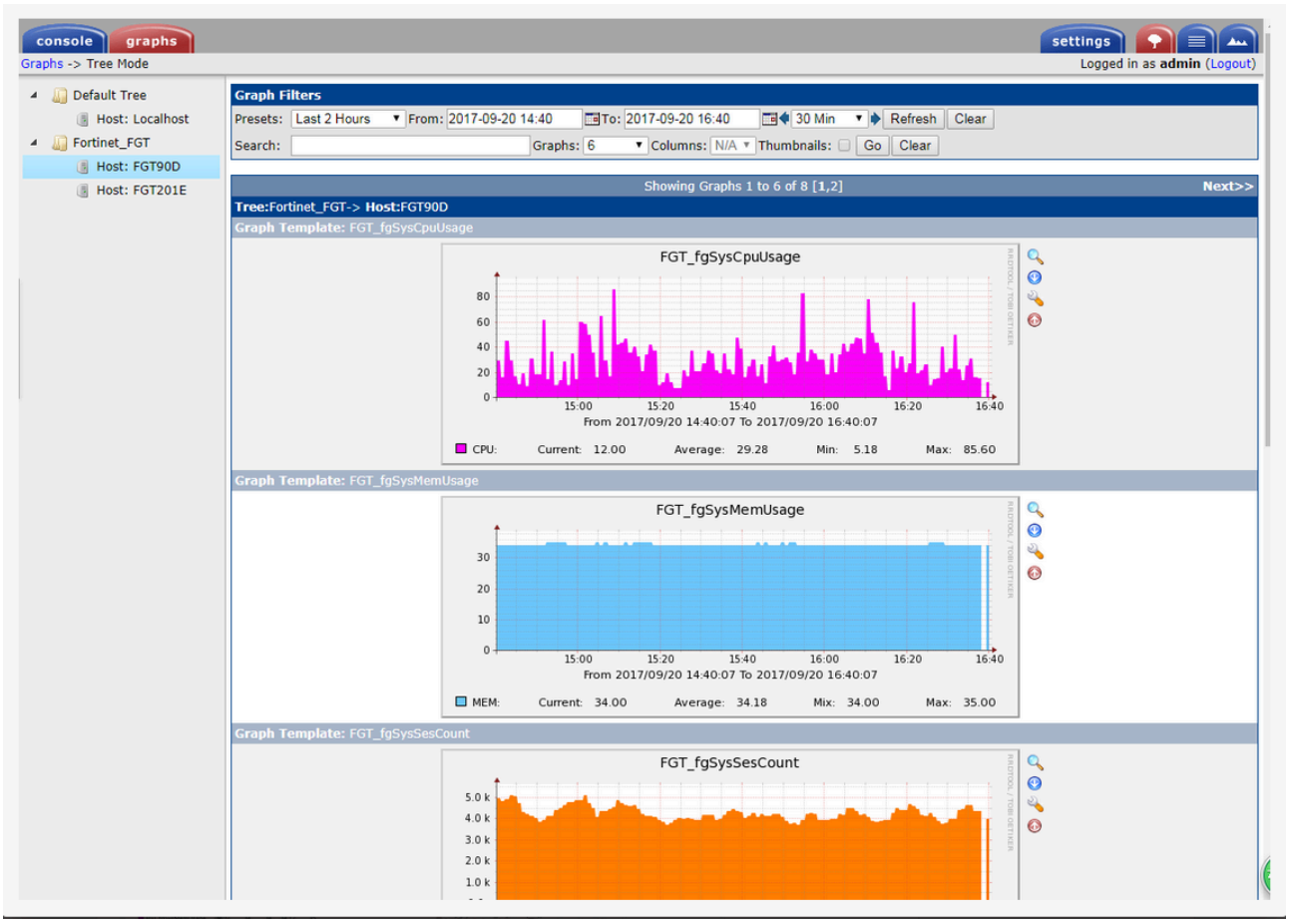
针对 FortiGate 的监控模板下载：https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/net/fortinet/fortigate_snmp?at ↗ = release/6.4



Cacti（开源免费）

网址链接：https://www.cacti.net/ ↗

download 网址：https://www.cacti.net/download_cacti.php ↗

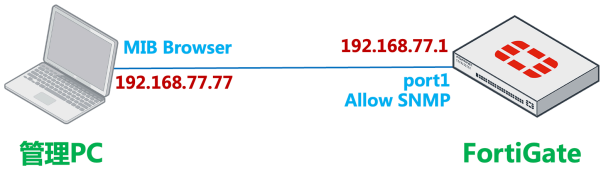

console
graphs
settings
Graphs -> Tree Mode
Logged in as admin (Logout)
Default Tree
Host: Localhost
Fortinet_FGT
Host: FGT90D
Host: FGT201E
Graph Filters
Presets: Last 2 Hours
From: 2017-09-20 14:40
To: 2017-09-20 16:40
30 Min
Refresh
Clear
Search:
Graphs: 6
Columns: N/A
Thumbnails:
Go
Clear
Showing Graphs 1 to 6 of 8 [1,2]
Next>>
Tree:Fortinet_FGT-> Host:FGT90D
Graph Template: FGT_fgSysCpuUsage
FGT_fgSysCpuUsage
80
60
40
20
0
15:00
15:20
15:40
16:00
16:20
16:40
From 2017/09/20 14:40:07 To 2017/09/20 16:40:07
CPU: Current: 12.00 Average: 29.28 Min: 5.18 Max: 85.60
Graph Template: FGT_fgSysMemUsage
FGT_fgSysMemUsage
30
20
10
0
15:00
15:20
15:40
16:00
16:20
16:40
From 2017/09/20 14:40:07 To 2017/09/20 16:40:07
MEM: Current: 34.00 Average: 34.18 Mix: 34.00 Max: 35.00
Graph Template: FGT_fgSysSesCount
FGT_fgSysSesCount
5.0 k
4.0 k
3.0 k
2.0 k
1.0 k

SNMP 的设置

网络需求

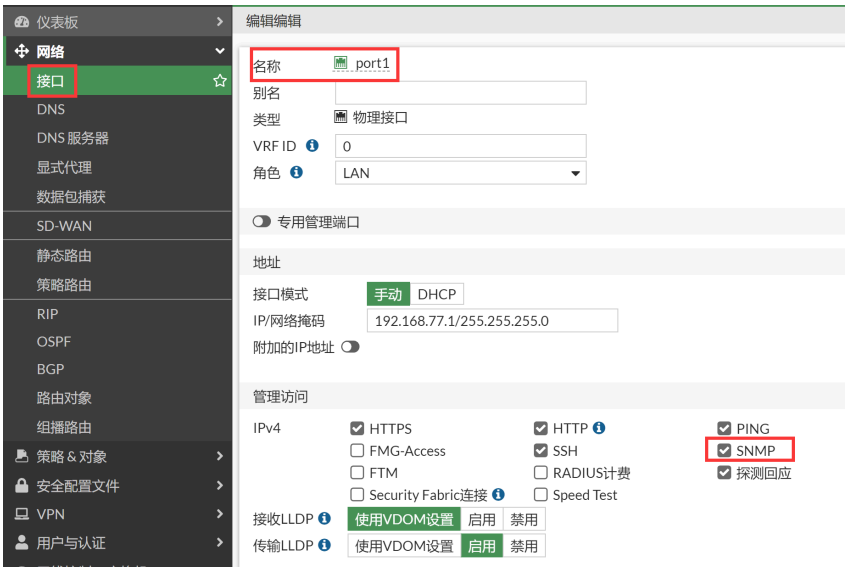
配置 FortiGate 的 SNMP。

网络拓扑

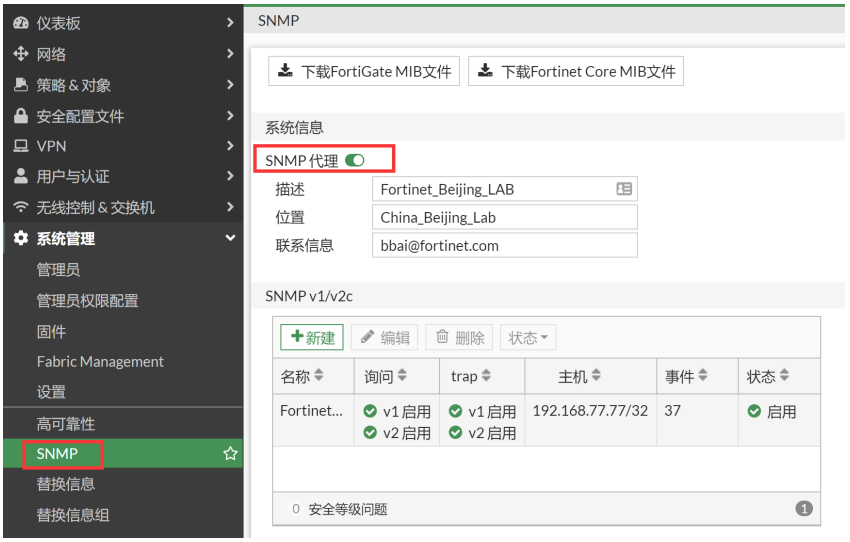


配置步骤

1. 接口上开启 SNMP 的允许访问。



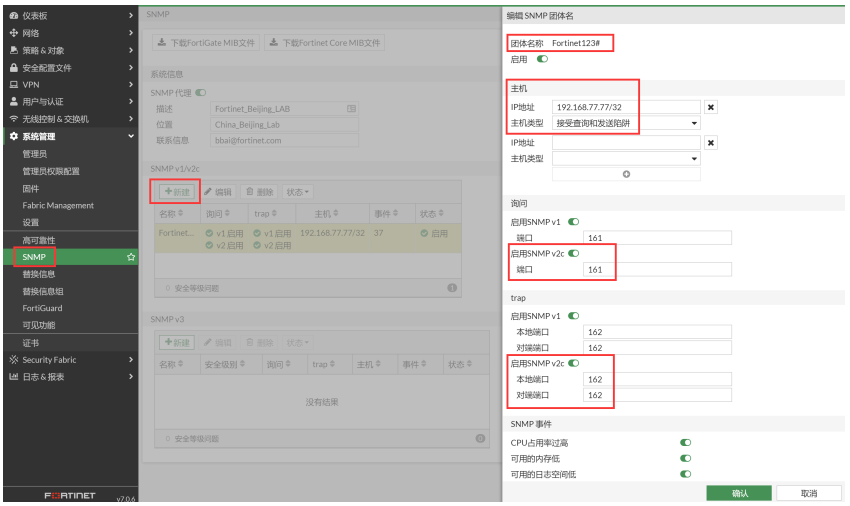
2. 开启 SNMP 的总开关。



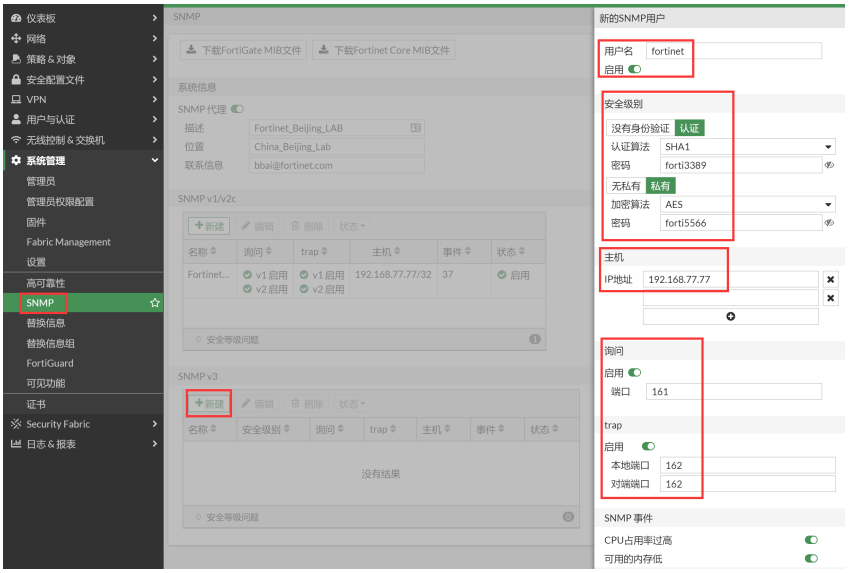
3. 配置 SNMP v2 的属性，比如配置 SNMPv2 的 SNMP 属性为 Fortinet123#，建议不要使用通用的 public，容易被探测和攻击。需要添加 SNMP 客户端的主机 IP（192.168.77.77）的 SNMP 权限，把主机的 IP 地址（192.168.77.77）添加到 SNMPv2 的配置里面。

重要

团体名称（Community）中不能包含字符 `$`，否则配置后读取 SNMP 会失败。



4. 配置 SNMPv3，配置 SNMPv3 的用户名为 fortinet，认证算法选择 SHA1，密码为 forti3389，加密算法选择为 AES，密码为 forti5566。需要添加 SNMP 客户端的主机 IP（192.168.77.77）的 SNMP 权限，把主机的 IP 地址（192.168.77.77）添加到 SNMPv2 的配置里面。



5. SNMP 命令行配置汇总。

```
config system interface
    edit "port1"
        set allowaccess ping https ssh snmp http probe-response
    next
end

config system snmp sysinfo
    set status enable
    set description "Fortinet_Beijing_LAB"
    set contact-info "bbai@fortinet.com"
    set location "China_Beijing_Lab"
end

config system snmp community
    edit 1
        set name "Fortinet123#"
        config hosts
            edit 1
                set ip 192.168.77.77 255.255.255.255
            next
        end
```

```
    next
end

config system snmp user
    edit "fortinet"
        set notify-hosts 192.168.77.77
        set security-level auth-priv
        set auth-pwd ENC
MTAwNG2U70/eePNhi1N3/u4q6FSRuv2ebcPGgqV+yxRuFtKxFilE6SmYZfpMvYOQBU4InxdTXnlIeYdMguMT8×7Hsqz/Q+3G2DWXlmtJohv0RukHWQK4nkY/
aYKCnujZkKGGyPxDKysAj4LDkR1CazeJkMKtVgGyoPF2WEjbPEt6PijsUZ67cDptDDzqnABzQFUemw==
        set priv-pwd ENC
MTAwNFIePySDQu7hqz+SFic3AUQ+1G4HX5er9qOeMxFTkZ8I7DnW+rw3XYNwgSfQtHuWEwTZGmztt6kUYUaT+oxLSdLG/RqU/3wbbE1/m4MH64bSYZgd+c+P
ks8S44UPmGBdQeewwpwQi7xHmL/y9Gyv+t7Wb6ge0WS3dQFe971Vp7n4evJgU51EK90Cnt7TjNX0lQ==
    next
end
```

重要

有一个特别的场景需要特别注意，那就是 HA 独立管理口的场景，需要一些特殊的配置才可以搞定。后边的章节：

系统管理 → SNMP → SNMP 有 HA 独立管理的设置 ↗

会描述这种情况。其他情况下都是以上类似的配置即可开启 FortiGate 的 SNMP。

结果验证

1. Net-SNMP SNMP V2 结果验证。

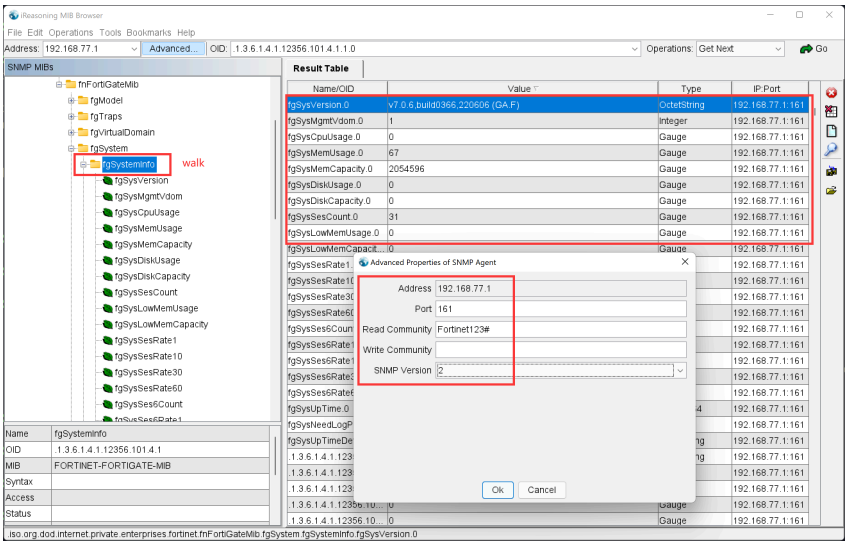
```
[Thu Aug 06 11:46:54 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.6,build0366,220606 (GA.F)"
[Thu Aug 06 11:46:55 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301635896) 34 days, 21:52:38.96
[Thu Aug 06 11:46:58 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000173"
[Thu Aug 06 11:47:46 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 1.3.6.1.2.1.31.1.1.1.1.1
IF-MIB::ifName.1 = STRING: port1
[Thu Aug 06 11:47:47 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 1.3.6.1.2.1.31.1.1.1.1.2
IF-MIB::ifName.2 = STRING: port2
[Thu Aug 06 11:47:51 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 1.3.6.1.4.1.12356.101.4.1.8.0
SNMPv2-SMI::enterprises.12356.101.4.1.8.0 = Gauge32: 77
```

2. Net-SNMP SNMPv3 结果验证。

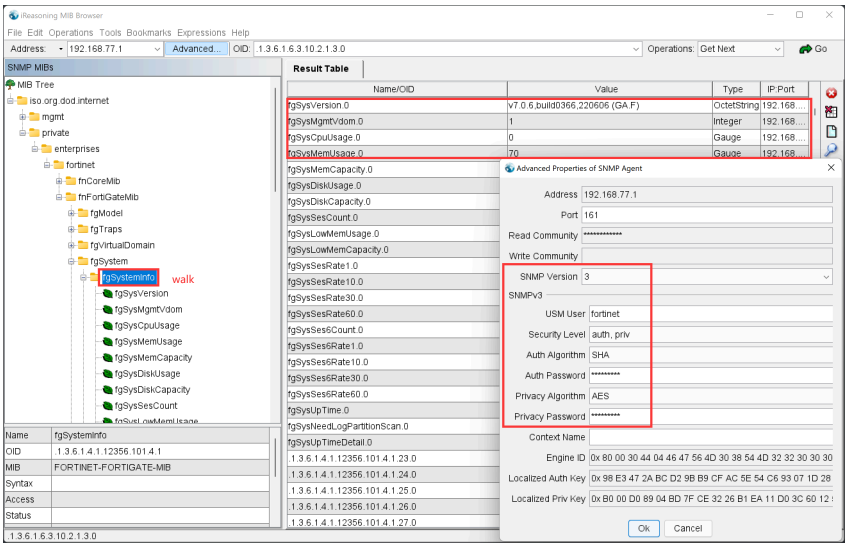
```
[Thu Aug 06 12:01:01 root@centos7~#net-snmp-config --create-snmpv3-user -ro -a SHA -A forti3389 -x AES -X forti5566
fortinet
adding the following line to /var/lib/net-snmp/snmpd.conf:
   createUser fortinet SHA "forti3389" AES forti5566
adding the following line to /etc/snmp/snmpd.conf:
   rouser fortinet
[Thu Aug 06 12:01:10 root@centos7~#chkconfig snmpd on
Note: Forwarding request to 'systemctl enable snmpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/snmpd.service to /usr/lib/systemd/system/snmpd.service.
[Thu Aug 06 12:01:30 root@centos7~#service snmpd start
Starting snmpd (via systemctl):                            [  OK  ]
[Thu Aug 06 12:06:38 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.1
SNMPv2-MIB::sysDescr.0 = STRING: Fortinet_Beijing_LAB
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.12356.101.1.60
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301754102) 34 days, 22:12:21.02
SNMPv2-MIB::sysContact.0 = STRING: bbai@fortinet.com
SNMPv2-MIB::sysName.0 = STRING: FortiGate
SNMPv2-MIB::sysLocation.0 = STRING: China_Beijing_Lab
SNMPv2-MIB::sysServices.0 = INTEGER: 78
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
SNMPv2-MIB::sysORIndex.1 = INTEGER: 1
SNMPv2-MIB::sysORID.1 = OID: SNMPv2-SMI::zeroDotZero.0
SNMPv2-MIB::sysORDescr.1 = STRING:
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (0) 0:00:00.00
IF-MIB::ifNumber.0 = INTEGER: 19
IF-MIB::ifIndex.1 = INTEGER: 1
```

```
IF-MIB::ifIndex.2 = INTEGER: 2
[Thu Aug 06 12:10:23 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.1 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000173"
[Thu Aug 06 12:10:45 root@centos7~#
[Thu Aug 06 12:10:46 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.1 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (301780209) 34 days, 22:16:42.09
[Thu Aug 06 12:11:00 root@centos7~#
[Thu Aug 06 12:11:12 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.1 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.6,build0366,220606 (GA.F)"
```

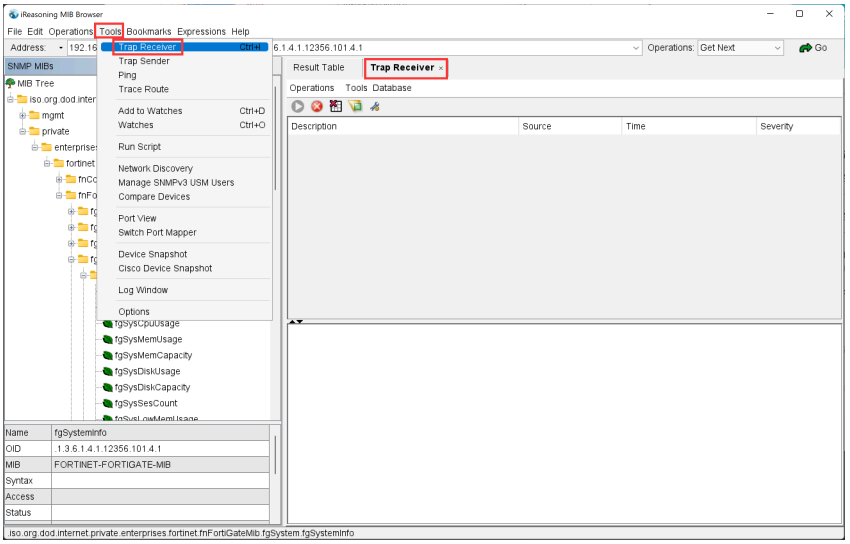
3. iReasoning MIB Browser SNMPv2 验证。

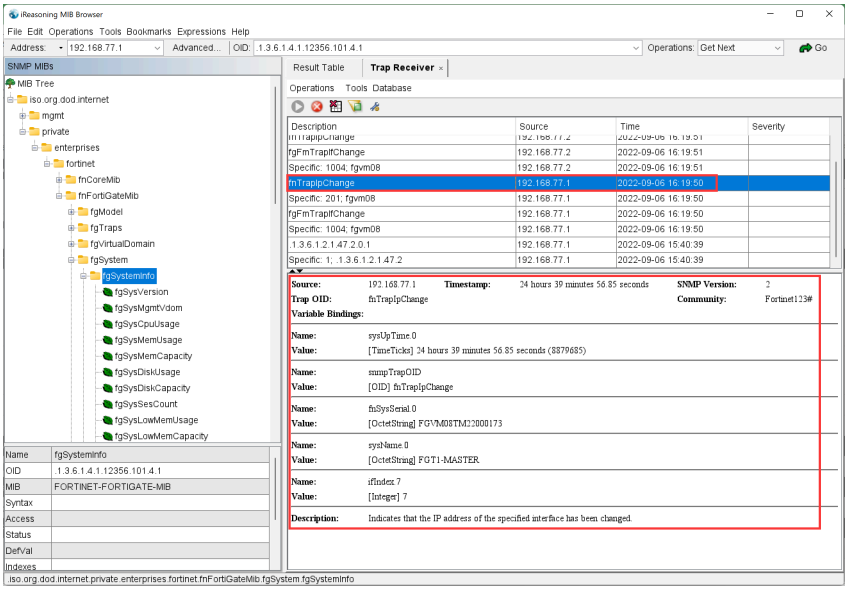


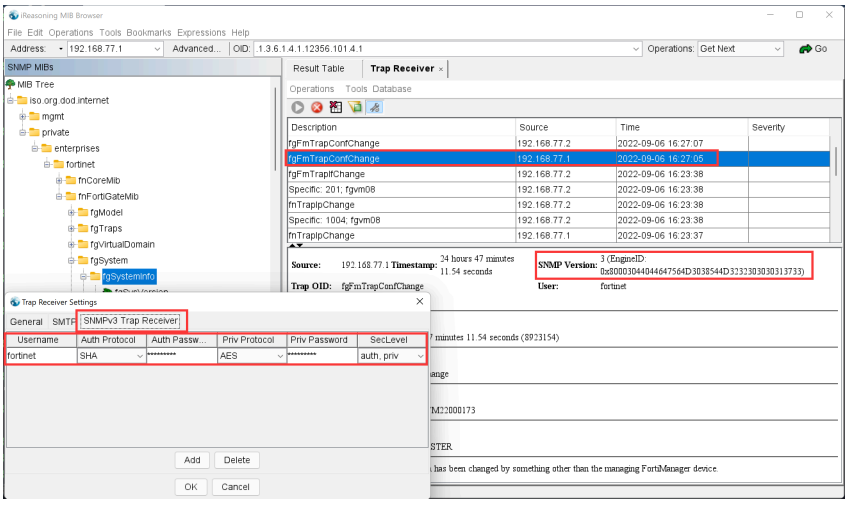
4. MIB Browser SNMP V3 验证。



5. iReasoning MIB Browser SNMP Trap 验证，SNMPv2 和 SNMPv3 均可以正常接收 Trap 消息。







SNMP 排错

1. 抓包，确认 SNMP UDP 161、162 的通信是正常的，正常 SNMP 是有去有回的。

```
# diagnose sniffer packet any "port 161 and host 192.168.77.1" 4
```

2. 通过第一步抓包看到 SNMP 流量有去有回了，但是 SNMP 还是失败，则开启 SNMP 进程的 debug，查看具体原因。

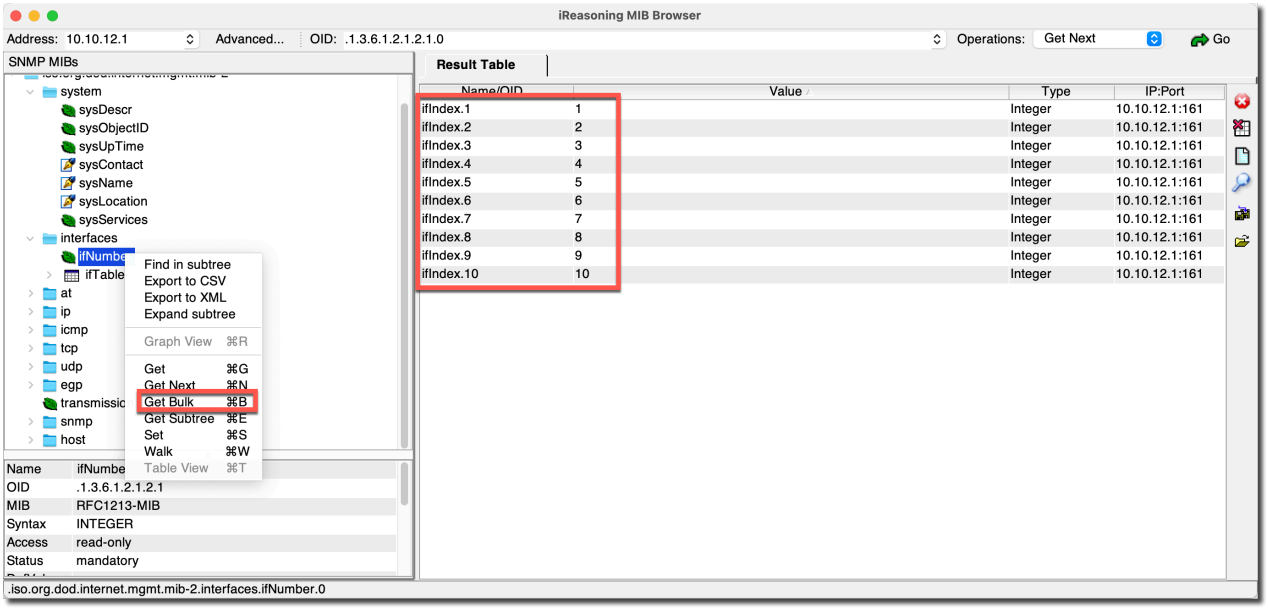
```
# diagnose debug application snmpd -1
# diagnose debug enable
```

比如，最常见的错误“community”不匹配、管理员配置了可信任主机，但是没有包括 SNMP 的 IP 地址等等...

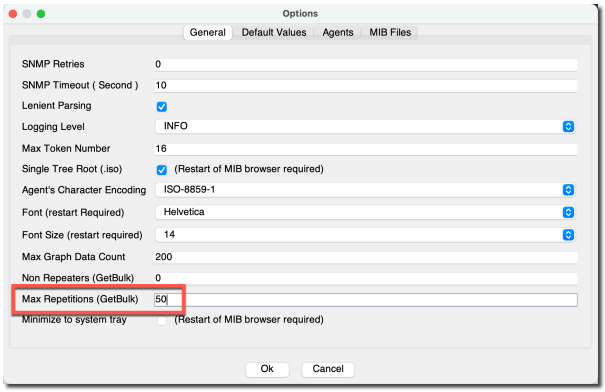
```
HUB2-ShangHai # snmpd: updating cache: idx_cache
snmpd: <msg> 47 bytes 192.168.90.254:44539 → 200.1.1.1/200.1.1.1:161 (itf 3.3)
snmpd: checking if community "Fortinet123#2" is valid
snmpd: updating cache: vdom_idx_map_cache
snmpd: updating vdom idx mapping
snmpd: Creating vdom_idx_cache for root
snmpd: Vdom created kernel-index=0, snmp-index=1, name=root
snmpd: checking against community "Fortinet123#"
snmpd: vdom name mismatch
snmpd: checking against community "FortiManager"
snmpd: name mismatch.
snmpd: failed to match community "Fortinet123#2"
snmpd: </msg> 0
```

注意事项

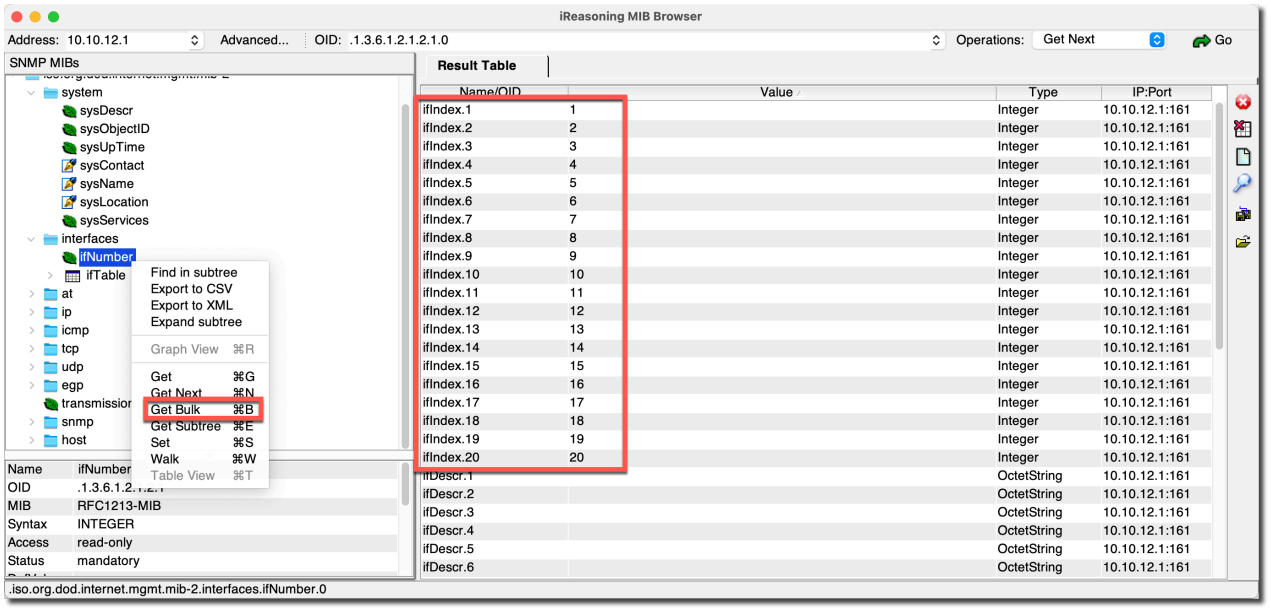
1. 使用 iReasoning MIB Browser 的“get bulk”方式去 get FortiGate 某节点的多个子节点时，默认情况下最多只能读取 10 个，如下所示读取 FortiGate 的接口列表时，使用 get bulk 方式只能读取 10 个接口信息：



2. 此时读取的接口信息是不全的，需要在 iReasoning MIB Browser 的软件配置中，修改“get bulk”的最大值，



3. 再次执行“get bulk”，所有的接口信息都可以 get 到：



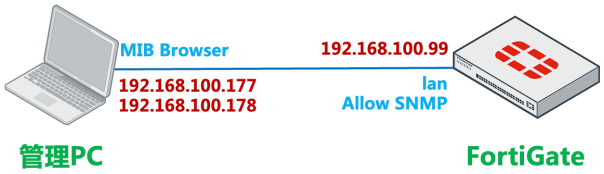
4. 也可以使用 walk 方式代替 get bulk 遍历节点。

SNMP 与管理员可信主机

网络需求

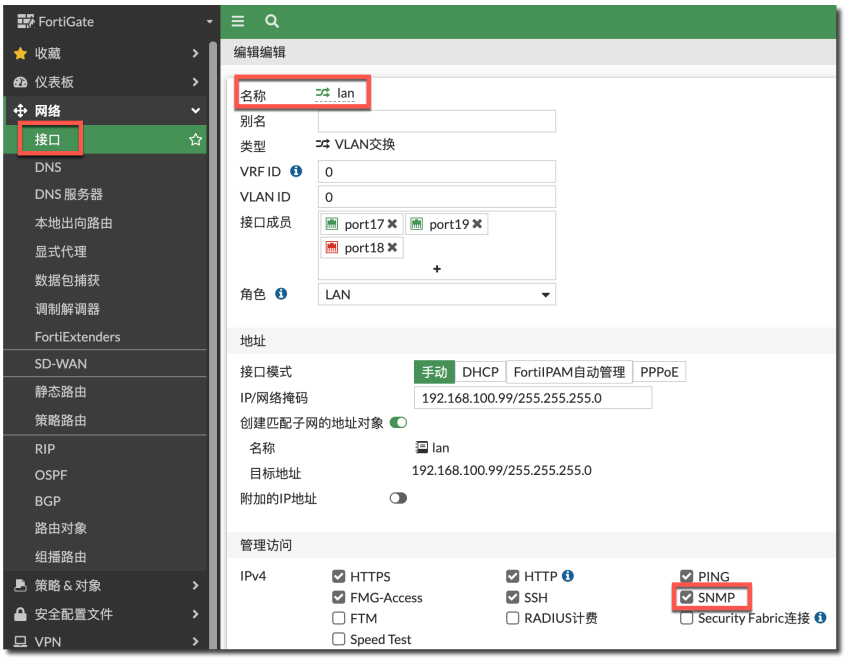
通过管理 PC 通过 FortiGate 的 lan 口访问 SNMP，同时 FortiGate 配置了管理员可信主机。

网络拓扑



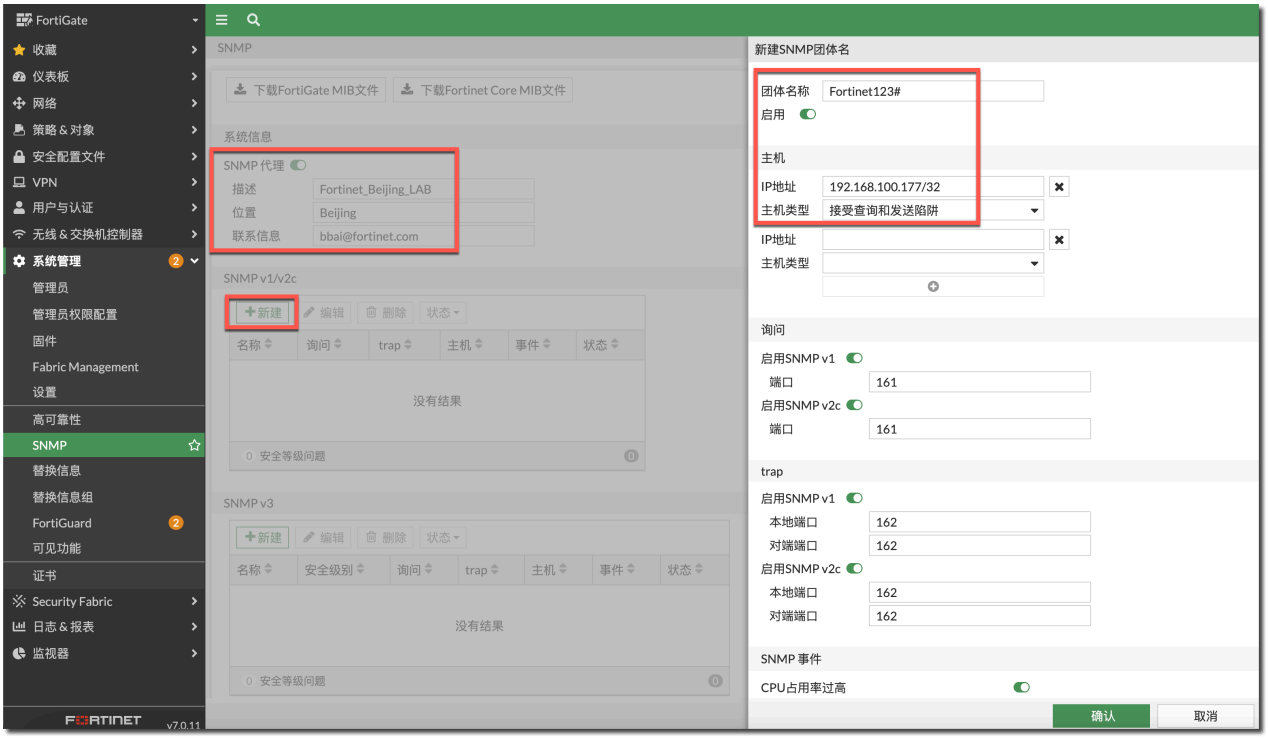
配置步骤

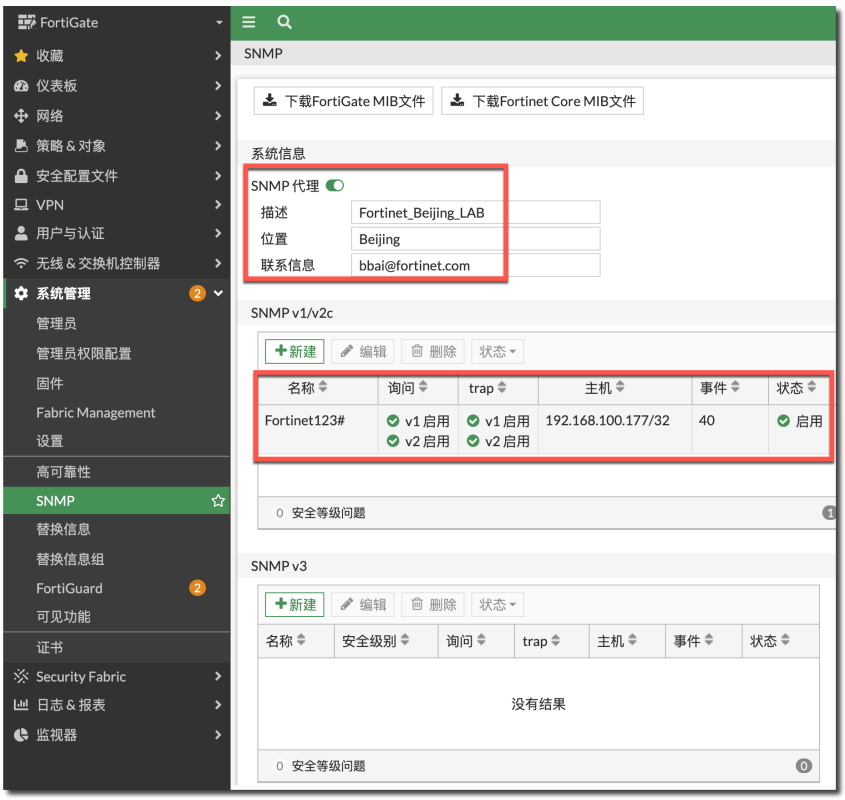
1. 在 FortiGate 的 lan 接口上开启 SNMP 的允许访问。




```
config system interface
    edit "lan"
        append allowaccess snmp
    next
end
```

2. 开启 SNMP 代理的总开关，并填写代理信息。新建并配置 SNMP v1 和 v2 的属性，例如配置 SNMP 团体名称为 Fortinet123#，建议不要使用通用的 public，容易被探测和攻击。配置 SNMP 主机为 192.168.100.177/32。






```
config system snmp sysinfo
    set status enable
    set description "Fortinet_Beijing_LAB"
    set contact-info "bbai@fortinet.com"
    set location "Beijing"
end

config system snmp community
    edit 2
        set name "Fortinet123#"
        config hosts
            edit 1
```

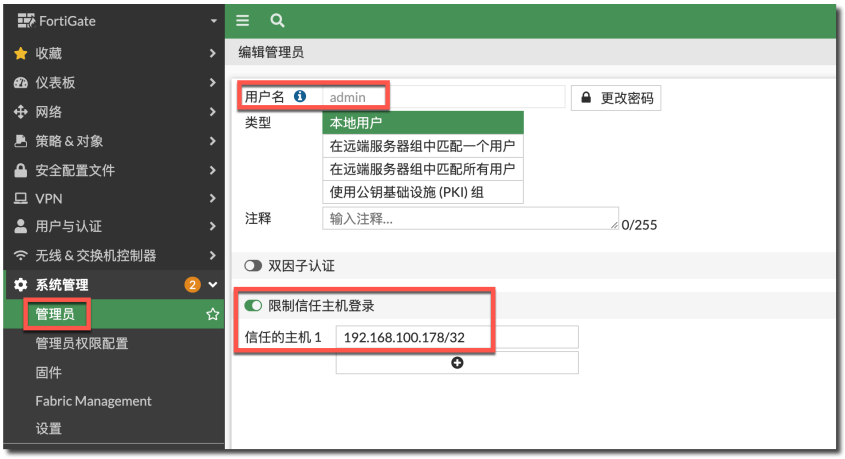
```
                set ip 192.168.100.177 255.255.255.255
            next
        end
    next
end
```

结果验证

1. 使用客户端 PC 192.168.100.177 对 FortiGate 进行 SNMP 读取，在没有配置管理员信任主机之前，SNMP 可以正常读取。

```
~ > ifconfig en7
...
    inet 192.168.100.177 netmask 0xffffff00 broadcast 192.168.100.255
...

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.11,build0489,230314 (GA.M)"

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (11060652) 1 day, 6:43:26.52

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FG101FTK20007637"
```

2. 配置管理员信任主机为 192.168.100.178/32，不包含 192.168.100.177/32。





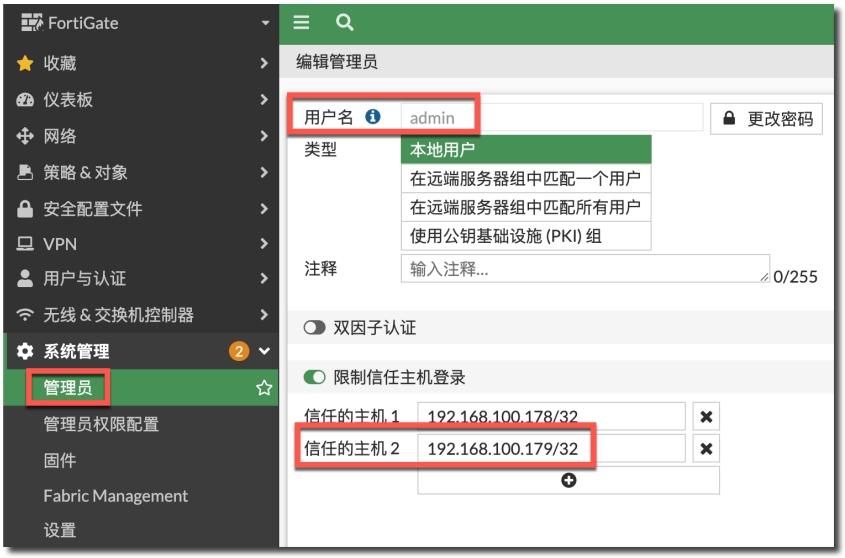

```
config system admin
    edit "admin"
        set trusthost1 192.168.100.178 255.255.255.255
    next
end
```

3. 再次使用客户端 PC 192.168.100.177 对 FortiGate 进行 SNMP 读取，全部超时失败。

```
~ > ifconfig en7
...
    inet 192.168.100.177 netmask 0xffffff00 broadcast 192.168.100.255
...

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.101.4.1.1.0
Timeout: No Response from 192.168.100.99

✘ ~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 1.3.6.1.2.1.1.3.0
Timeout: No Response from 192.168.100.99

✘ ~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.100.1.1.1.0
Timeout: No Response from 192.168.100.99
```

4. 很明显管理员信任主机是对接口下面的所有访问协议起作用的，包括所有的管理协议（除了 PING），自然也包括了 SNMP 协议。所以即使接口下开启了 SNMP 协议，但配置了管理主机后，也只有管理主机可以访问该接口的 SNMP 服务。

5. 开启管理员可信主机后要解决这个问题，需要将 SNMP 管理主机的 IP 地址也加入到管理员的可信任主机中，这样 SNMP 才可以正常获取信息。




```
config system admin
    edit "admin"
        set trusthost1 192.168.100.178 255.255.255.255
        set trusthost2 192.168.100.179 255.255.255.255
    next
end
```

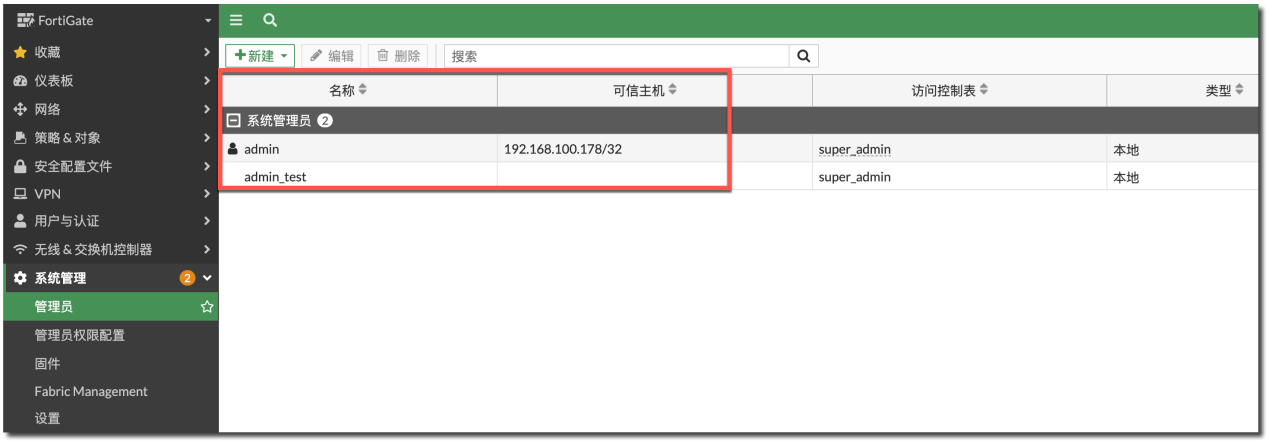
6. 再次使用客户端 PC 192.168.100.177 对 FortiGate 进行 SNMP 读取，读取成功。

```
~ > ifconfig en7
...
    inet 192.168.100.177 netmask 0xffffff00 broadcast 192.168.100.255
...

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.11,build0489,230314 (GA.M)"

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (11060652) 1 day, 6:43:26.52

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FG101FTK20007637"
```

注意事项

1. 以上情况中，FortiGate 只存在一个管理员账户 admin，所以当配置了管理员可信主机后，即使接口下开启了管理协议，可信主机外的 IP 也无法正常访问接口下的管理协议（除了 PING）。

2. 但如果存在多个管理员账户，而其中有管理员配置了可信主机，但有管理员没有配置可信主机（如下图所示），且接口下开启了各种管理协议。那么对于像 SNMP 这种不需要 FortiGate 管理员认证的协议，则可以通过可信主机外的 IP 正常访问。




```
~ > ifconfig en7
...
    inet 192.168.100.179 netmask 0xffffff00 broadcast 192.168.100.255
...

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.101.4.1.1.0
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.11,build0489,230314 (GA.M)"

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (11060652) 1 day, 6:43:26.52

~ > snmpwalk -v2c -c Fortinet123# 192.168.100.99 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FG101FTK20007637"
```

3. 对于 HTTPS、SSH、Telnet 这类需要认证 FortiGate 管理员账号的协议来说：

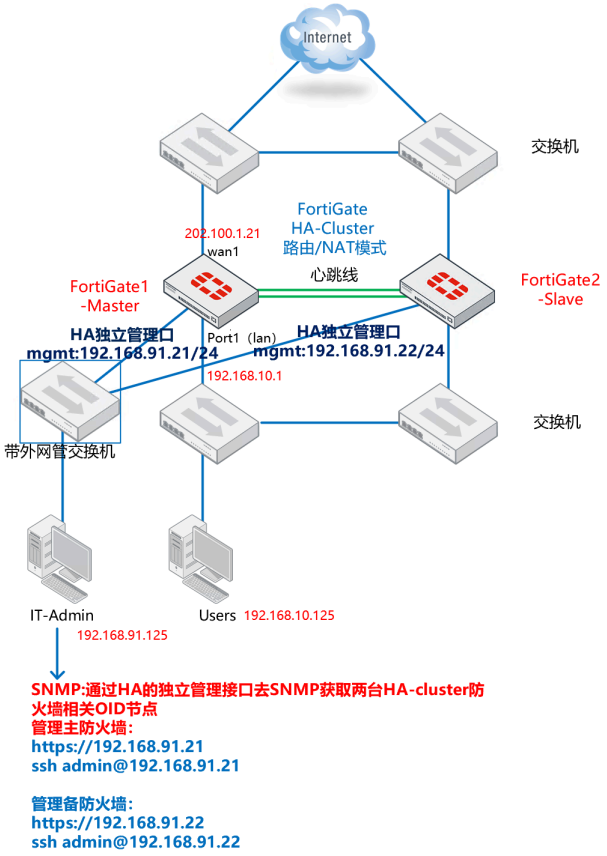
   - 当只有一个管理员并且配置了可信主机后，非可信主机访问 HTTPS、SSH、Telnet 服务会被直接拒绝（无法打开 HTTPS 登录页面和 SSH/Telnet 连接）。
   - 当像 2 中那样配置了多个管理员，有的管理员配置了可信主机，有的管理员没有配置可信主机，那么使用非可信主机访问 HTTPS 的登录页面是可以正常打开登录页面的，SSH/Telnet 登录信息也是可以显示的，会在登陆时检查是否为对应管理员的可信主机。

SNMP 的 HA 独立管理

需求

在 HA 环境下配置 FortiGate 的 SNMP，并且通过 HA 成员的独立管理接口进行 SNMP 操作。

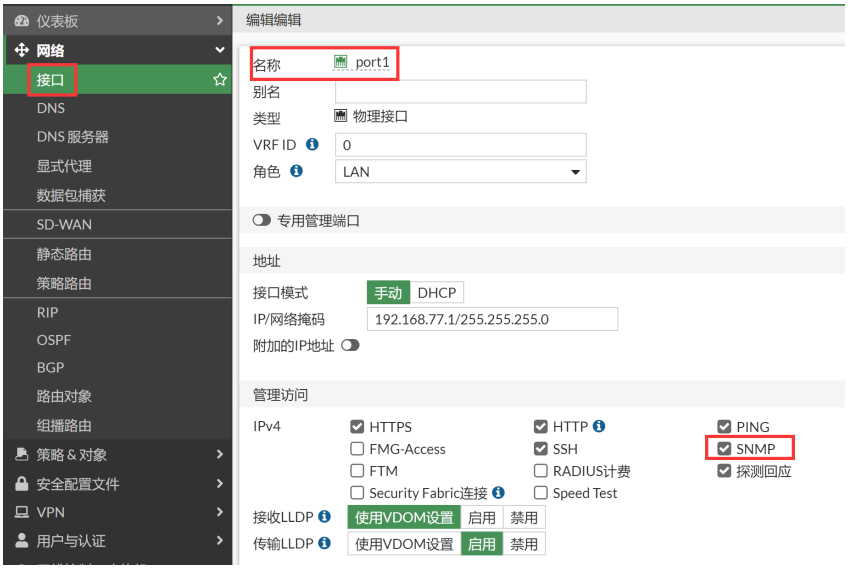
组网拓扑



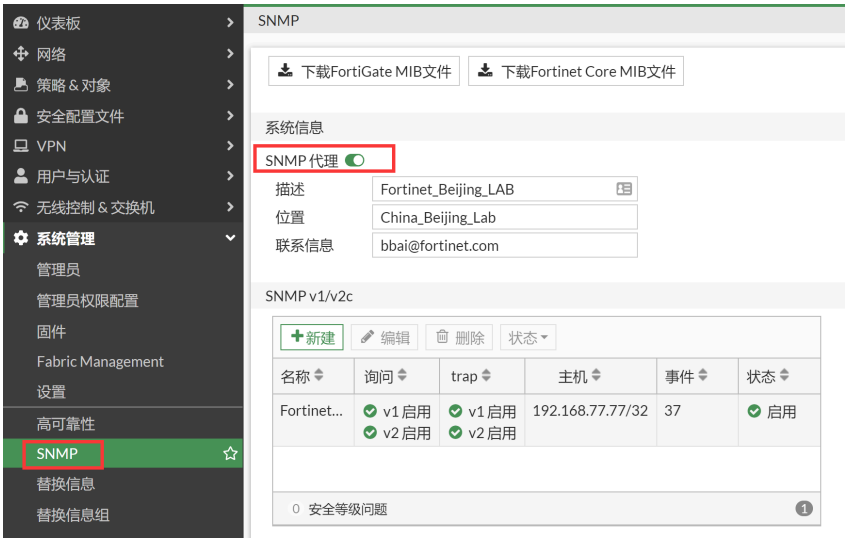
操作步骤

> **重要**
>
> HA 独立管理口的配置，请参考：HA 双机热备 → HA-Cluster 的网管 → HA 集群带外独立管理 - 独立管理接口 ↗章节，使用的同样的一个拓扑图。本章节将关注点放置于如何配置和使用 SNMP 上。

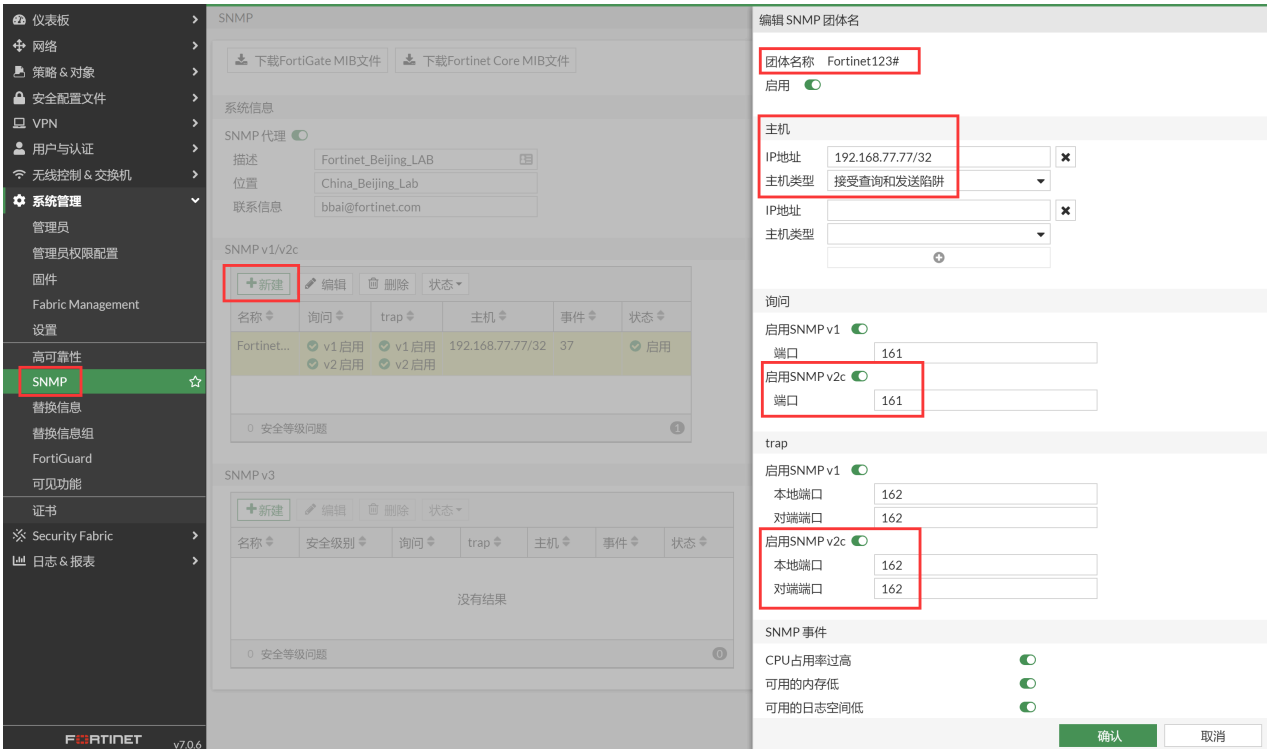
1. 在 HA 独立管理接口上开启 SNMP 的允许访问（包括主防火墙的 HA 独立管理口 port1 和备防火墙的 HA 独立管理口 port1）。



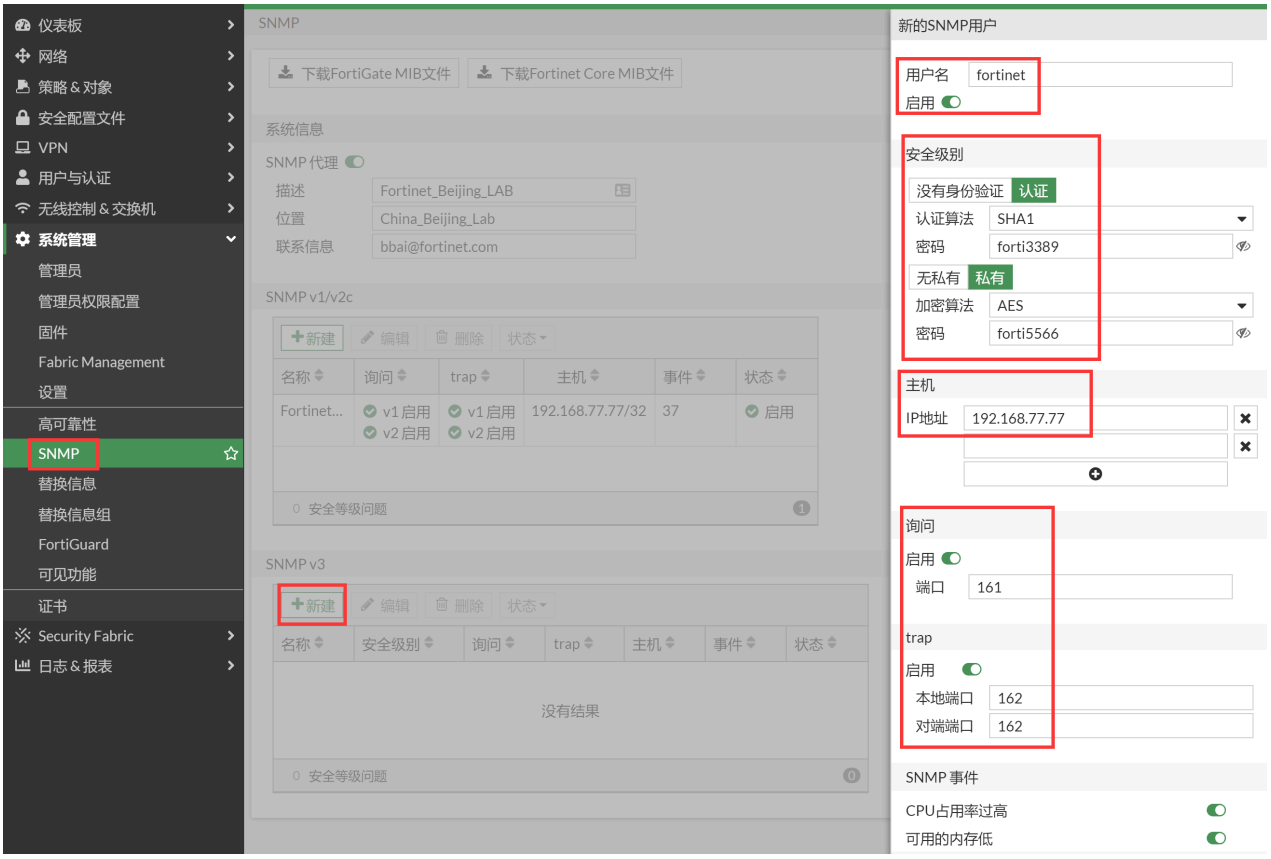
2. 开启 SNMP 的总开关。



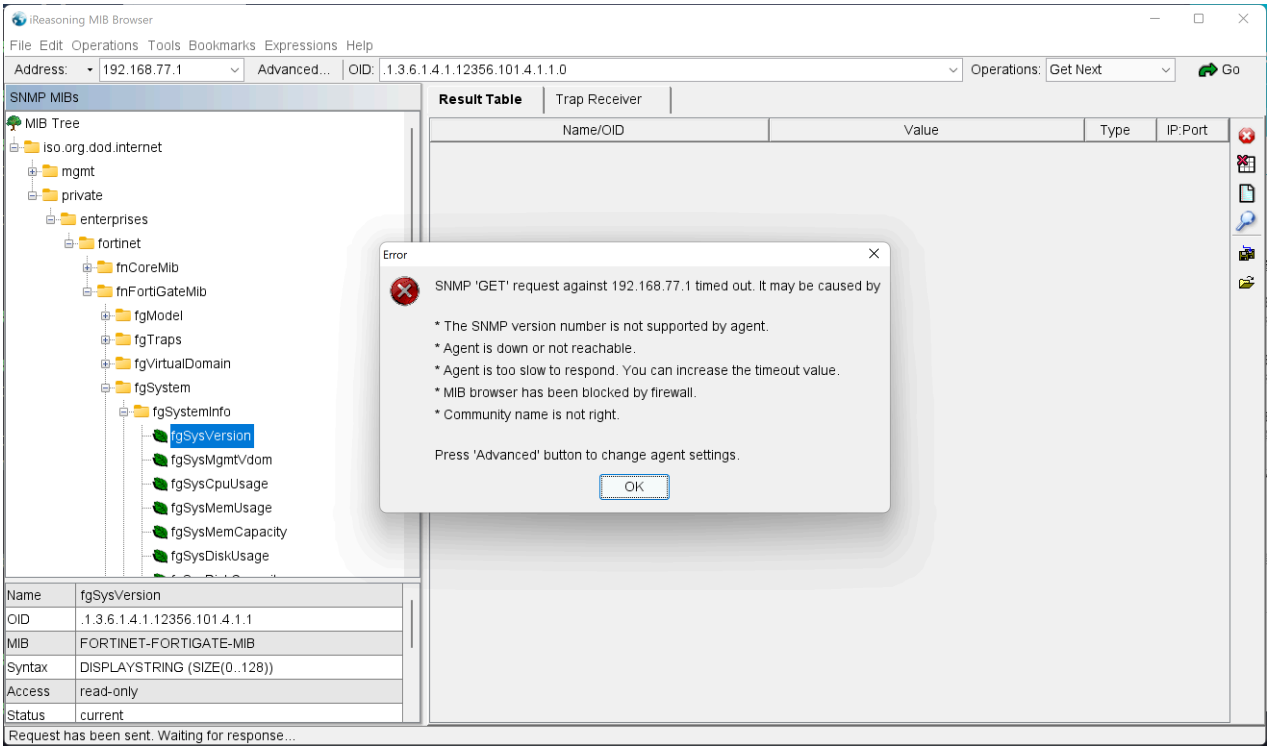
3. 配置 SNMP v2 的属性，比如配置 SNMPv2 的 SNMP 属性为 Fortinet123#，建议不要使用通用的 public，容易被探测和攻击。需要添加 SNMP 客户端的主机 IP（192.168.77.77）的 SNMP 权限，把主机的 IP 地址（192.168.77.77）添加到 SNMPv2 的配置里面。



4. 配置 SNMPv3，配置 SNMPv3 的用户名为 fortinet，认证算法选择 SHA1，密码为 forti3389，加密算法选择为 AES，密码为 forti5566。需要添加 SNMP 客户端的主机 IP（192.168.77.77）的 SNMP 权限，把主机的 IP 地址（192.168.77.77）添加到 SNMPv2 的配置里面。



5. 这个时候去测试 SNMP 会发现 SNMP 是无法成功获取到的。



6. 在 FortiGate 上进行抓包，可以看到 SNMP 查询没有任何回应，debug flow 显示 SNMP 请求在 vd-vsys_hamgmt 里面被丢弃的。

```
FGT1-MASTER # diagnose sniffer packet any "host 192.168.77.77 and port 161" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.77.77 and port 161]
7.576093 port1 in 192.168.77.77.53750 → 192.168.77.1.161: udp 53
7.576474 port1 in 192.168.77.77.53750 → 192.168.77.1.161: udp 53
15.795977 port1 in 192.168.77.77.53751 → 192.168.77.1.161: udp 53
15.796132 port1 in 192.168.77.77.53751 → 192.168.77.1.161: udp 53

FGT1-MASTER # diagnose debug flow filter port 161
FGT1-MASTER # diagnose debug flow filter addr 192.168.77.77
FGT1-MASTER # diagnose debug flow show function-name enable
show function name
FGT1-MASTER # diagnose debug flow trace start 10
FGT1-MASTER # diagnose debug enable
FGT1-MASTER # id=20085 trace_id=1 func=print_pkt_detail line=5863 msg="vd-vsys_hamgmt:0 received a packet(proto=17,
192.168.77.77:53831→192.168.77.1:161) tun_id=0.0.0.0 from port1. "
id=20085 trace_id=1 func=init_ip_session_common line=6042 msg="allocate a new session-00014576, tun_id=0.0.0.0"
id=20085 trace_id=1 func=vf_ip_route_input_common line=2605 msg="find a route: flag=84000000 gw-192.168.77.1 via
vsys_hamgmt"
id=20085 trace_id=2 func=print_pkt_detail line=5863 msg="vd-vsys_hamgmt:0 received a packet(proto=17,
192.168.77.77:53831→192.168.77.1:161) tun_id=0.0.0.0 from port1. "
id=20085 trace_id=2 func=resolve_ip_tuple_fast line=5949 msg="Find an existing session, id-00014576, original direction"
```

7. 查看 SNMP 的 debug 信息，显示无法在 vsys_hamgmt VDOM 下查找到对应的 community。

```
FGT1-MASTER # diagnose debug application snmpd -1
Debug messages will be on for 28 minutes.
FGT1-MASTER # diagnose debug enable
FGT1-MASTER # snmpd: <msg> 53 bytes 192.168.77.77:34002 → 192.168.77.1/192.168.77.1:161 (itf 3.3)
snmpd: checking if community "Fortinet123#" is valid
snmpd: updating cache: vdom_idx_map_cache
snmpd: updating vdom idx mapping
snmpd: Creating vdom_idx_cache for root
snmpd: Vdom created kernel-index=0, snmp-index=1, name=root
snmpd: cannot locate vdom with name vsys_hamgmt
snmpd: checking against community "Fortinet123#"
snmpd: request 2(vsys_hamgmt)/3/192.168.77.77 ≠ comm 1/0/192.168.77.77/255.255.255.255
```

```
snmpd: host or intf mismatch
snmpd: checking against community "FortiManager"
snmpd: full name mismatch
snmpd: failed to match community "Fortinet123#"
snmpd: </msg> 0
```

重要

SNMP 获取失败的原因是什么呢?

原因就是因为存在 HA 独立管理口，HA 独立管理口其实认为是一个独立的 VDOM，vsys_hamgmt，默认的 SNMP 处理是 root 这个 VDOM 进行处理的，而不是 vsys_hamgmt，因此 SNMP 请求 HA 的独立管理口是不行的。

这个时候如果是请求的是 root vdom 里面的业务接口，非 HA 独立管理口的话，就没啥问题的，此问题仅仅出现在请求 HA 独立管理口的 SNMP 的情况下。

因此为了满足这种场景下的 SNMP 使用需求，在 SNMP 的配置下有一个命令来适配 HA 独立管理口的使用，具体可以参考 Fortinet Community 文档的说明:

https://community.fortinet.com/t5/FortiGate/Technical-Tip-FortiGate-SNMP-polling-via-the-dedicated-HA/ta-p/196034 ↗

https://community.fortinet.com/t5/FortiGate/Technical-Tip-How-to-configure-SNMP-Community-over-Cluster/ta-p/192507 ↗

8.**关键的一步，配置 snmp 主机的 ha-direct enable**，这样 SNMP 才可以通过 HA 独立管理口获取到数据（只支持命令行配置）。以下分别是 SNMPv2 和 SNMPv3 下的配置方法。（如果只是需要 SNMP 相关流量从每台 HA 成员的独立管理口收发，则不需要在 config system ha 下开启 ha-direct）

```
config system snmp community
    edit 1
        set name "Fortinet123#"
        config hosts
            edit 1
                set ip 192.168.77.77 255.255.255.255
                set ha-direct enable
            next
    next
end
```

```
config system snmp user
    edit "fortinet"
        set notify-hosts 192.168.77.77
        set ha-direct enable
    next
end
```

重要

开启 SNMP 配置下的“HA-direct”后，主机和备机的 SNMP Trap 报文也会通过各自的独立管理口分别发出，而不是通过主机的业务口发出。

结果验证

1. Net-SNMP SNMPv2 结果验证。

```
[Tue Aug 11 15:28:55 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 .1.3.6.1.4.1.12356.101.4.1.1.0  // 主防火墙结果验证
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.6,build0366,220606 (GA.F)"
[Tue Aug 11 15:28:57 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (94369138) 10 days, 22:08:11.38
[Tue Aug 11 15:29:02 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.1 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000173"
[Tue Aug 11 15:29:15 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.2 .1.3.6.1.4.1.12356.101.4.1.1.0  // 备防火墙结果验证
SNMPv2-SMI::enterprises.12356.101.4.1.1.0 = STRING: "v7.0.6,build0366,220606 (GA.F)"
[Tue Aug 11 15:29:17 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.2 1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (93667021) 10 days, 20:11:10.21
[Tue Aug 11 15:29:21 root@centos7~#snmpwalk -v2c -c Fortinet123# 192.168.77.2 .1.3.6.1.4.1.12356.100.1.1.1.0
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000174"
```

2. Net-SNMP SNMP V3 结果验证。

```
[Thu Aug 06 12:01:01 root@centos7~# net-snmp-config --create-snmpv3-user -ro -a SHA -A forti3389 -x AES -X forti5566 fortinet
adding the following line to /var/lib/net-snmp/snmpd.conf:
   createUser fortinet SHA "forti3389" AES forti5566
adding the following line to /etc/snmp/snmpd.conf:
   rouser fortinet
[Thu Aug 06 12:01:10 root@centos7~#chkconfig snmpd on
Note: Forwarding request to 'systemctl enable snmpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/snmpd.service to /usr/lib/systemd/system/snmpd.service.
[Thu Aug 06 12:01:30 root@centos7~#service snmpd start
Starting snmpd (via systemctl):                            [  OK  ]
[Tue Aug 11 15:37:22 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.1 .1.3.6.1.4.1.12356.100.1.1.1.0 // 主防火墙
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000173"
[Tue Aug 11 15:37:23 root@centos7~#snmpwalk -v3 -u fortinet -l authPriv -a SHA -A forti3389 -x AES -X forti5566
192.168.77.2 .1.3.6.1.4.1.12356.100.1.1.1.0 // 备防火墙
SNMPv2-SMI::enterprises.12356.100.1.1.1.0 = STRING: "FGVM08TM22000174"
```

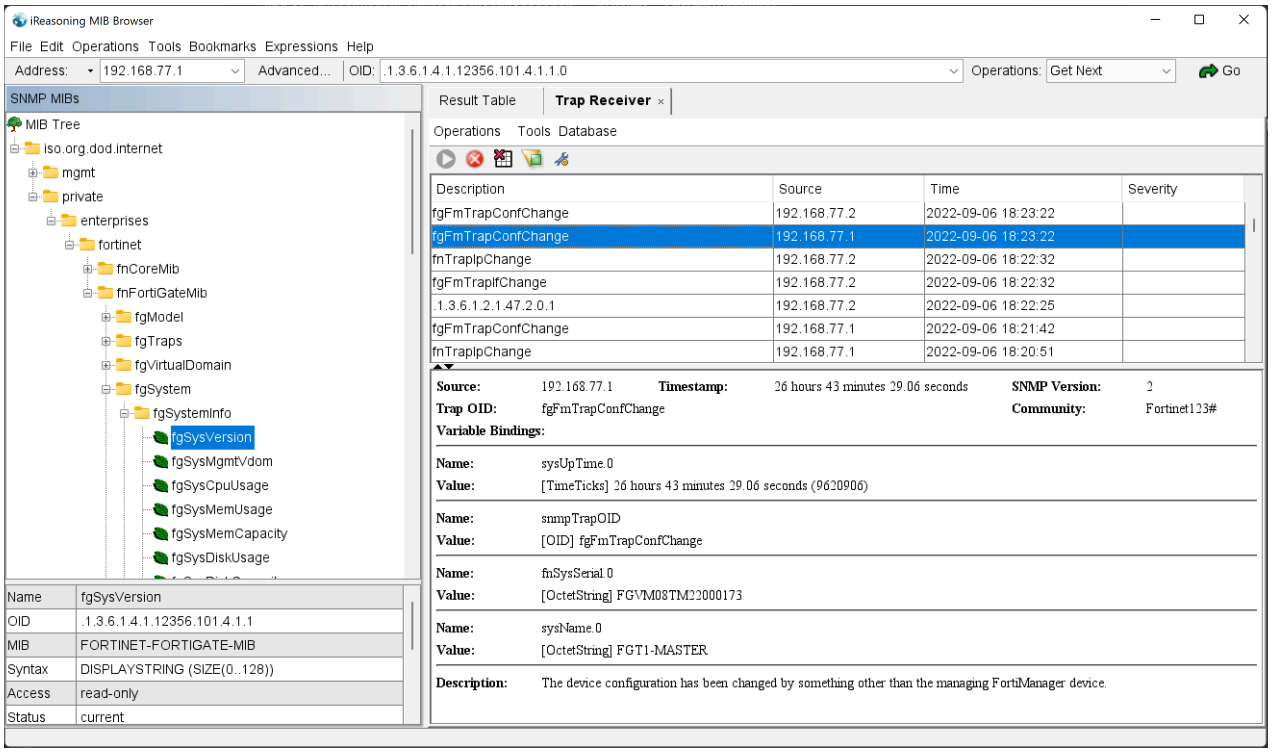
重要

Net-SNMP v3 参考链接：https://www.thegeekdiary.com/centos-rhel-6-install-and-configure-snmpv3/ ↗

3. SNMPv2 和 SNMPv3 的 SNMP 查询都正常，这个时候 SNMP 的查询就 OK 了，那么现在考虑另外一个问题，SNMP 的 Trap 是否也 OK 呢？我们抓包看一下 SNMP Trap 的情况，可以看到 SNMP Trap 仍然可以通过 HA 独立管理口发送，并不需要在 HA 下开启 ha-direct。

```
FGT2-SLAVE # diagnose sniffer packet any 'port 162' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 162]
15.428042 port1 out 192.168.77.2.162 → 192.168.77.77.162: udp 138
18.958515 port1 out 192.168.77.2.162 → 192.168.77.77.162: udp 253
133.104369 port1 out 192.168.77.2.162 → 192.168.77.77.162: udp 138
```

4. 查看 IReasoning MIB Browser 上的 Trap Receiver，可以看到主机和备机分别发送的 SNMP Trap。



监控 BGP/OSPF 的注意事项

1. 在 SNMP 启用 ha-direct 的情况下，查询 BGP/OSPF 相关 OID 时请使用“community-vdom”格式。例如，在 root VDOM 中 SNMP 配置的 community 为 public，查询 root VDOM 的 BGP/OSPF 相关 OID 需要使用 public-root 来作为 community（如果未开启 VDOM 功能，默认的 VDOM 就是 root）。

重要

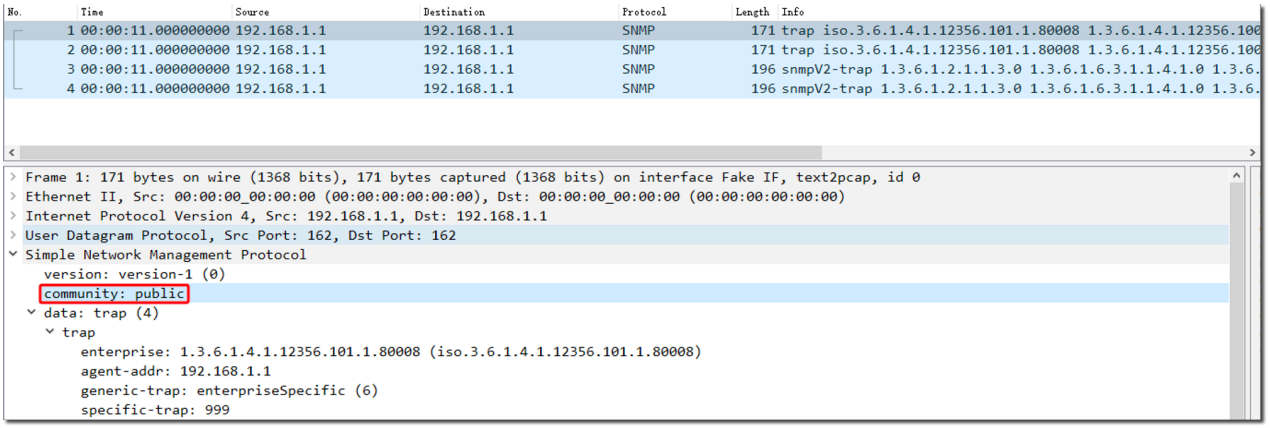
- 未启用 SNMP 的 ha-direct，通过业务口查询 BGP/OSPF 相关 OID，直接使用配置的 community 即可。
- 此特性只适用于 FortiOS 7.2.11/7.4.8 之前版本，在 FortiOS 7.2.11/7.4.8 及以后版本中，只需要使用 community/user 格式，不能使用 user/community-vdom 格式。

```bash
config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ha-direct enable
            next
        end
    next
end

root@centos7~# snmpwalk -v 2c -c public-root 192.168.88.87 .1.3.6.1.2.1.15.3.1.2
SNMPv2-SMI::mib-2.15.3.1.2.192.168.90.247 = INTEGER: 6
SNMPv2-SMI::mib-2.15.3.1.2.192.168.168.130 = INTEGER: 1
```

2. SNMPv3 也要使用一样的格式，如下 FortiGate 配置的 SNMPv3 community 为 user1，查询 root VDOM 的 BGP/OSPF OID 需要使用 public-user1 来作为 community（如果未开启 VDOM 功能，默认的 VDOM 就是 root）。

```bash
config system snmp user
    edit "public"
        set ha-direct enable
        set security-level auth-priv
        set auth-pwd fortinet
        set priv-pwd fortinet
    next
end

root@centos7~# snmpwalk -v3 -u public-root -l authPriv -a sha -A fortinet -x aes -X fortinet 192.168.88.87
.1.3.6.1.2.1.15.3.1.2
SNMPv2-SMI::mib-2.15.3.1.2.192.168.90.247 = INTEGER: 6
SNMPv2-SMI::mib-2.15.3.1.2.192.168.168.130 = INTEGER: 1
```

3. 针对 SNMP Trap，在 SNMP 启用 ha-direct 的情况下，FortiGate 主动发送的 BGP/OSPF 相关 Trap 报文中的 community 不会使用“community-vdom”格式，而是直接使用 community，如下报文所示。



4. 这样 SNMP 分析软件（如 Zabbix、SolarWinds 等）使用“community-vdom”格式作为 community 时，会由于团体子不匹配而报错。

5. 解决方法是在 FortiGate 上配置两个 Community，格式如下：

- 一个 Community 使用“community”格式（如 public），host-type 为 query，用于 OID 的读取。
- 另一个 Community 使用“community-vdom”格式（如 public-root），host-type 为 trap，用于发送 BGP/OSPF 相关 Trap。
- SNMP 分析软件配置的监控 Community 为“community-vdom”格式（如 public-root）。

```bash
config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ip 192.168.1.1 255.255.255.255
```

```
                set host-type query
            next
        end
    next
    edit 2
        set name "public-root"
        config hosts
            edit 1
                set ip 192.168.1.1 255.255.255.255
                set host-type trap
            next
        end
    next
end
```

基于 VDOM 的 SNMP

简介

本文章介绍 FortiGate 在启用 VDOM 时如何读取基于 VDOM 的 SNMP 参数。

FortiGate 的 SNMP 查询分为两类：

- Global 全局级别查询（无论是否启用 VDOM 均可使用）
- Per-VDOM 虚拟域级别查询（仅在启用 VDOM 时有效）

在启用 VDOM 后，每个 VDOM 都会分配一个独立的 snmp-index。所有基于 VDOM 的 SNMP OID 都需要携带该 index 才能正确读取数据。

使用方法

以会话相关监控为例，使用 Fortinet MIB 中的相关 OID 获取全局及 VDOM 级别的会话数及会话建立速率。

未启用 VDOM

无需考虑 snmp-index，所有 Session 相关 OID 都直接从系统读取：

- 总 Session 数（`fgSysSesCount`）：`.1.3.6.1.4.1.12356.101.4.1.8`。
- Session 建立速率（1/10/30/60 分钟平均，`fgSysSesRate1/10/30/60`）：`.1.3.6.1.4.1.12356.101.4.1.11~14`。

启用 VDOM

1. 启用 VDOM 后，每个 VDOM 拥有独立资源，因此 SNMP 的会话读取需使用 VDOM 专用 OID。

2. 根据 VDOM 的 snmp-index：

   - 当前会话数（`fgVdEntSesCount`）：`.1.3.6.1.4.1.12356.101.3.2.1.1.7.x`。
   - VDOM 的 Session 建立速率（`fgVdEntSesRate`）：`.1.3.6.1.4.1.12356.101.3.2.1.1.8.x`。

3. 如何找到 VDOM 的 snmp-index?

   - 方法 1：CLI 查看。

     ```bash
     config global
     show system vdom-property
     edit "root"
         set snmp-index 1     ←——root VDOM snmp-index = 1
     next
     edit "VPN"
         set snmp-index 2     ←——VPN VDOM snmp-index = 2
     next
     ......
     ```

   - 方法 2：SNMP 查询 VDOM 名称表（`fgVdEntName`：`.1.3.6.1.4.1.12356.101.3.2.1.1.2.x`）:，其中 x 为 VDOM 的 snmp-index，每个 x 对应一个 VDOM 名称。

4. 例如要读取名为 VPN 的 VDOM，snmp-index 为 2。就查询`.1.3.6.1.4.1.12356.101.3.2.1.1.7.2`。

总结

- 在未启用 VDOM 时，所有 SNMP 会话信息均通过 Global OID 获取。
- 启用 VDOM 后，每个 VDOM 拥有独立 SNMP index，必须使用 Per-VDOM 专用 OID，并带入 snmp-index 才能正确读取。

FortiGuard 管理

FortiGuard 服务介绍

License 到期的影响

服务指南

- FortiCare 服务介绍
- 设备注册步骤
- 设备强制注册
- 服务注册步骤
- PRMA 高级服务注册步骤
- 产品服务续费
- RMA 服务迁移
- RMA/DOA 硬件保修政策
- 产品注册信息变更
- 远程技术支持
- FortiEDR Cloud 注册指南

在线更新特征库

离线更新特征库

离线导入设备 License

FortiGuard 更新服务器位置

查看合同/更新状态

FortiGuard 连接问题

禁用云通信

特征库误报/缺失上报

FortiGuard 服务介绍

服务介绍

1. 设备服务是需要注册才会生效的（当然有特例，就是设备购买的前三个月，可以免费的试用所有的服务，而通常这些服务都是需要购买并且注册才生效的）。
2. 注册分为设备注册、服务注册（将服务绑定到固定的设备上，详细可见后续的设备注册、服务注册章节）。
3. 注册地址：https://support.fortinet.com ↗
4. 注册指南（点击查看） ↗。

服务分类

1. FortiCare 支持服务
   - RMA 硬件返修服务
   - 400 电话/邮件等等技术支持服务
2. 固件和通用更新
   - 设备固件下载与更新服务
   - APP Control 特征库更新
   - 设备识别库更新
   - Internet service DateBase 库更新

   > **重要**
   >
   > 只需要有基础的 FortiCare 服务即可得到以上服务更新。

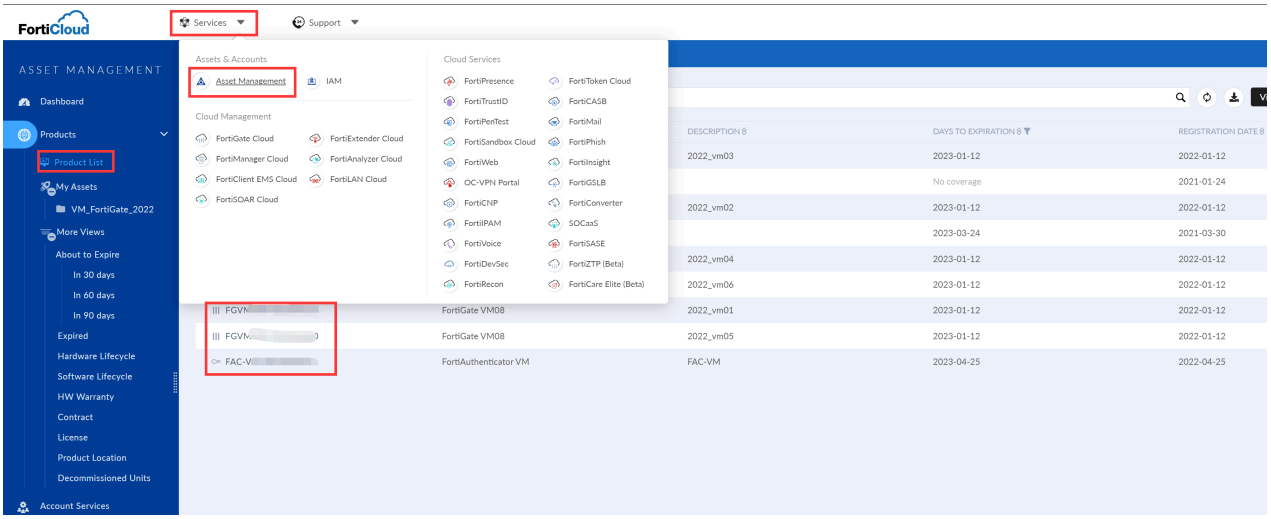
3. 入侵防御（IPS 特征库更新）
   - IPS 特征库
   - IPS 引擎
   - 恶意 URL
   - 僵尸网络 IP 库
   - 僵尸网络域名库
4. 反病毒（AV 特征库更新）
   - AV 特征库
   - AV 引擎
   - 移动端恶意软件
5. Web 过滤（URL 分类过滤）
   - FortiGuard URL 分类
   - 列入黑名单的网址数字证书
6. 工业安全数据库（Industrial DB）
7. FortiGate Cloud/FortiSandbox Cloud（有一定空间的免费许可）
8. Anti-Spam 过滤

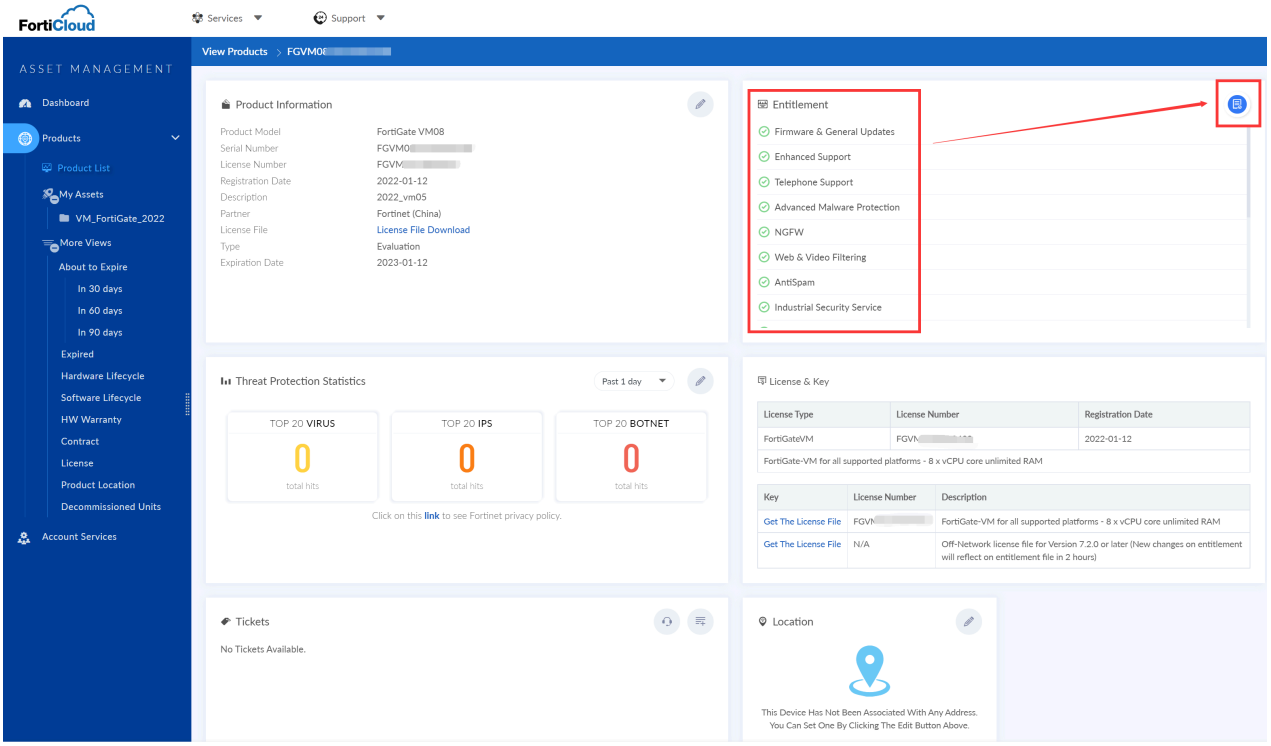
> **重要**
>
> 以上的 NGFW 类的特征服务，需要购买相应的 license 服务许可才可以正常的使用（设备开箱启动可以免费全功能使用 3 个月试用），具体的 Bondle 价格等等需要通过供应商处获取。关于特征库的更新和研究的网站：http://www.fortiguard.com/ ↗

9. 虚拟域许可（默认有 10 个 VDOM 的 license 许可）
0. 其他许可，不同的软件版本，以上内容不尽相同，但是都大致有上述的模块和内容，具体不一样的版本可能存在一些差异，以上基于 FortiOS 7.0.6 进行的说明。

服务查看

1. 通过 Support 网站登陆后查看：https://support.fortinet.com/asset/#/views/products ↗ 查看设备服务状态，注册设备之后才可以查看。


FortiCloud
Services
Support
ASSET MANAGEMENT
Dashboard
Products
Product List
My Assets
VM_FortiGate_2022
More Views
About to Expire
In 30 days
In 60 days
In 90 days
Expired
Hardware Lifecycle
Software Lifecycle
HW Warranty
Contract
License
Product Location
Decommissioned Units
Account Services
Assets & Accounts
Asset Management
IAM
Cloud Management
FortiGate Cloud
FortiExtender Cloud
FortiManager Cloud
FortiAnalyzer Cloud
FortiClient EMS Cloud
FortiLAN Cloud
FortiSOAR Cloud
Cloud Services
FortiPresence
FortiToken Cloud
FortiTrustID
FortiCASB
FortiPenTest
FortiMail
FortiSandbox Cloud
FortiPhish
FortiWeb
FortiInsight
OC-VPN Portal
FortiGSLB
FortiCNP
FortiConverter
FortiIPAM
SOCaaS
FortiVoice
FortiSASE
FortiDevSec
FortiZTP (Beta)
FortiRecon
FortiCare Elite (Beta)
DESCRIPTION
DAYS TO EXPIRATION
REGISTRATION DATE
2022_vm03
2023-01-12
2022-01-12
No coverage
2021-01-24
2022_vm02
2023-01-12
2022-01-12
2023-03-24
2021-03-30
2022_vm04
2023-01-12
2022-01-12
2022_vm06
2023-01-12
2022-01-12
FGVM
FortiGate VM08
2022_vm01
2023-01-12
2022-01-12
FGVM
FortiGate VM08
2022_vm05
2023-01-12
2022-01-12
FAC-V
FortiAuthenticator VM
FAC-VM
2023-04-25
2022-04-25


FortiCloud
Services
Support
ASSET MANAGEMENT
Dashboard
Products
Product List
My Assets
VM_FortiGate_2022
More Views
About to Expire
In 30 days
In 60 days
In 90 days
Expired
Hardware Lifecycle
Software Lifecycle
HW Warranty
Contract
License
Product Location
Decommissioned Units
Account Services
View Products > FGVM0
Product Information
Product Model FortiGate VM08
Serial Number FGVM0
License Number FGVM
Registration Date 2022-01-12
Description 2022_vm05
Partner Fortinet (China)
License File License File Download
Type Evaluation
Expiration Date 2023-01-12
Entitlement
Firmware & General Updates
Enhanced Support
Telephone Support
Advanced Malware Protection
NGFW
Web & Video Filtering
AntiSpam
Industrial Security Service
Threat Protection Statistics
Past 1 day
TOP 20 VIRUS
0
total hits
TOP 20 IPS
0
total hits
TOP 20 BOTNET
0
total hits
Click on this link to see Fortinet privacy policy.
License & Key
License Type
License Number
Registration Date
FortiGateVM
FGVM
2022-01-12
FortiGate-VM for all supported platforms - 8 x vCPU core unlimited RAM
Key
License Number
Description
Get The License File
FGVM
FortiGate-VM for all supported platforms - 8 x vCPU core unlimited RAM
Get The License File
N/A
Off-Network license file for Version 7.2.0 or later (New changes on entitlement will reflect on entitlement file in 2 hours)
Tickets
No Tickets Available.
Location
This Device Has Not Been Associated With Any Address.
You Can Set One By Clicking The Edit Button Above.

Product Entitlements

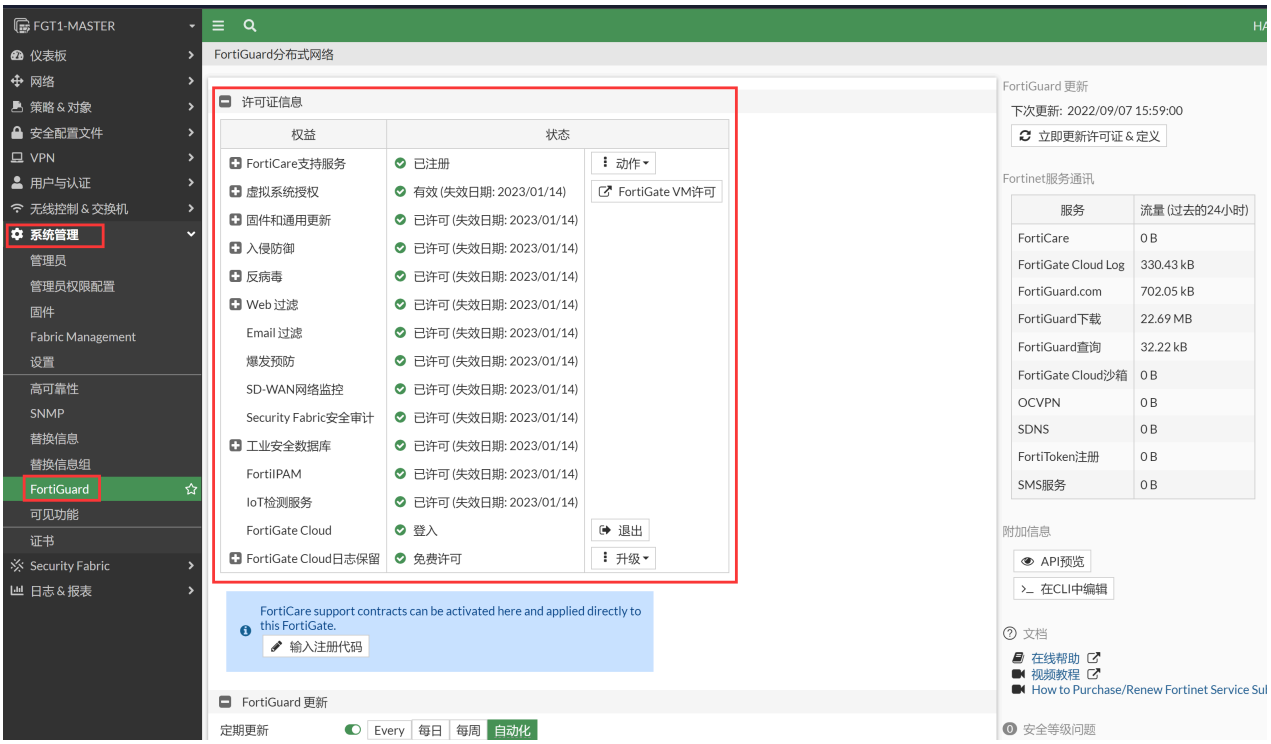
SUPPORT COVERAGE

Support Type	Support Level	Activation Date	Expiration Date
Firmware & General Updates	Web/Online	2022-01-12	2023-01-12
Enhanced Support	Premium	2022-01-12	2023-01-12
Telephone Support	Premium	2022-01-12	2023-01-12
Advanced Malware Protection	Web/Online	2022-01-12	2023-01-12
NGFW	Web/Online	2022-01-12	2023-01-12
Web & Video Filtering	Web/Online	2022-01-12	2023-01-12
AntiSpam	Web/Online	2022-01-12	2023-01-12
Industrial Security Service	Web/Online	2022-01-12	2023-01-12
FortiManager Cloud	Web/Online	2022-01-12	2023-01-12
FortiAnalyzer Cloud	Web/Online	2022-01-12	2023-01-12
FortiConverter Service	8x5	2022-01-12	2023-01-12
Security Rating Update	Web/Online	2022-01-12	2023-01-12
SD-WAN Monitoring	Web/Online	2022-01-12	2023-01-12
SD-WAN Overlay Controller	Web/Online	2022-01-12	2023-01-12
Elite Service	Web/Online	2022-01-12	2023-01-12
SD-WAN Orchestrator	Web/Online	2022-01-12	2023-01-12
IoT Detection	Web/Online	2022-01-12	2023-01-12
IPAM Cloud	Web/Online	2022-01-12	2023-01-12

REGISTERED SUPPORT CONTRACT

Contract Number	SKU	Creation Date	Registration Date
06[redacted]	FCX-15-00000-819-02-12	2022-01-12	2022-01-12

2. 通过 FortiGate 管理页面查看（FortiGate 的服务是通过互联网进行更新，因此 FortiGate 需要可以连接到互联网，才可以正常的自动更新），正常情况下，网站的服务和设备上的服务将会是同步的状态。



常见问题

1. FortiGate 设备服务无法正常更新。
   - FortiGate 无法连接到互联网，包括 FortiGate 无法上网、DNS 异常解析、运营商阻止了 FortiGuard 更新的目的端口，FortiGate 的某些源端口工作异常。

- FortiGate 当前配置的 DNS 服务器无法解析 FortiGuard 更新服务器域名，建议更换为公网上可信且未受污染的 DNS 服务器，以确保解析与更新正常。
- 服务没有在 https://support.fortinet.com ↗ 网站上注册，先到网站上确认设备服务状态。
- 有 HA 的环境下，只会显示服务到期时间最近的那台设备的服务时间，如果其中有一台设备过期，那么 HA 中的其他设备均会显示过期。
- FortiGate 默认为 2 小时更新一次，有时候不是立即就生效的，需要等一等，或者手工更新一下，手工命令如下：`execute update-now`。
- 拥有多个 VDOM，而可以上互联网的不是管理 VDOM（默认是管理 VDOM 是 root），这样也会导致更新失败，修改管理 VDOM 为可以上网的 VDOM，或者让 root VDOM 可以上互联网。

2. 排错思路和方法：
   - Troubleshooting Tip: Diagnosing FortiGuard problems of Antivirus, Intrusion Prevention, Web Filtering, Spam Filtering: https://community.fortinet.com/t5/FortiGate/Troubleshooting-Tip-Diagnosing-FortiGuard-problems-of-Antivirus/ta-p/194121 ↗
   - Technical Tip: Verifying and troubleshooting FortiGuard updates status and versions: https://community.fortinet.com/t5/FortiGate/Technical-Tip-Verifying-and-troubleshooting-FortiGuard-updates/ta-p/194931 ↗
   - Technical Note: FortiGuard Support licenses show as expired: https://community.fortinet.com/t5/FortiGate/Technical-Note-FortiGuard-Support-licenses-show-as-expired/ta-p/194093 ↗
3. FortiGate 无法连接互联网的情况下，如何更新特征库?
   - 手工导入特征库，GUI 导入特征库或者命令行导入特征库，特征库可以在 https://support.fortinet.com/support/#/downloads/service ↗ 下载离线版本特征库（需要登陆 FortiCloud 帐号）。
   - 统一通过 FortiManger 作为 FortiGuard 的中转站，需要 FortiManger 可以连接互联网（可以使用代理方式），FortiGate 无需上互联网，FortiGate 将会通过 FMG 更新服务和特征库。
4. FortiGuard 服务过期后会发生什么?
   - 基于特征库的服务到期，如 AV、IPS、应用控制、IPS，相关特征库将无法继续更新，但可以继续使用旧的特征库。
   - 基于 FortiGuard 实时查询的服务到期，如 WebFilter、DNS Filter，实时分类查询功能将无法使用，请注意可能导致开启 WebFilter 或 DNS Filter 的安全策略匹配的流量被阻断，但基于静态 URL Filter 的 Web/DNS Filter 可以继续使用。

License 到期的影响

当 FortiGate 的一个或多个 FortiGuard 许可证过期时，以下功能可能会受到影响。

License 类型	到期影响
固件和基本订阅（Firmware & General Updates）	• 应用控制特征、Inline-CASB 应用程序定义、设备和操作系统标识定义、虚拟补丁签名、ISDB 库、PSIRT 检查定义将继续工作，但特征库不再更新。 • 示例：如果在防火墙策略中使用了应用控制配置文件，并在目的地址中应用了 ISDB 条目，该策略仍会使用现有的应用控制特征库和 ISDB 库版本进行流量检测，正常流量不会中断。 • 设备无法手动升级/降级（包括 GUI 和 CLI）。 • 设备将自动计划升级到当前小版本的最新补丁。升级可手动延后 1 周，但无法取消（参考：系统管理 → 固件与配置管理 → 固件版本管理 → 过期 License 强制升级 ↗）。
入侵防御系统（Intrusion Prevention）	• IPS 签名、恶意 URL 定义将继续工作，但 IPS 签名不再更新。 • 示例：如果在防火墙策略中使用了启用 IPS 配置文件，该策略仍会使用现有 IPS 特征库和恶意 URL 定义进行检测，正常流量不会中断。 • 最新有效的 IPS 签名对于阻止复杂攻击和 0-day 攻击至关重要。
僵尸网络 IP/域名（Botnet IPs/Domains）	• 配置了 Botnet C & C 的 IPS/DNS 过滤配置文件将继续工作，但 Botnet IP 与域名数据库不再更新，正常流量不会中断。
反病毒（AntiVirus）	• 反病毒扫描将继续工作，但反病毒数据库不再更新，正常流量不会中断。
Web/DNS 过滤（Web/DNS Filter）	• 基于 FortiGuard 实时分类查询的 Web/DNS Filter 将停止工作，因为 URL/域名需要实时发送到 FortiGuard 进行分类查询。 • 如果防火墙策略中引用了 Web/DNS Filter 配置文件（开启了“基于 FortiGuard 分类的过滤器”），默认配置下，所有 Web 与 DNS 流量将被丢弃；如果 Web/DNS Filter 配置文件的“无法访问 FortiGuard 时的行为”选项配置为“允许所有网站”，则所有 Web/DNS 流量将直接通过，不进行过滤。 • 在 Web/DNS Filter 配置文件中配置的静态 URL 过滤规则仍可以正常工作。
邮件过滤（Email Filtering）	• 邮件过滤功能将停止工作，因为 FortiGate 需要实时查询 FortiGuard 垃圾邮件过滤服务器以检查发件 IP、邮件地址、钓鱼 URL、垃圾 URL、邮件校验值以及垃圾邮件提交信息（本地配置的除外）。 • 反垃圾邮件签名将不再更新。 • 基于“本地垃圾邮件过滤”的配置选项（如“黑白名单”）仍可使用。
爆发防御（Outbreak Prevention）	• 爆发防御将停止工作，因为该功能依赖于对 FortiGuard 全球威胁情报数据库的实时查询。 • 示例：如果在防火墙策略中使用了反病毒配置文件，AV 配置文件中开启了“使用 FortiGuard 爆发防御数据库”，该功能将不会生效，正常流量不会中断。

提示

FortiGuard 服务为连续服务模式，任何服务中断，在续费时都需要补齐至合同到期日的服务周期。详情参考：系统管理 → FortiGuard 管理 → 服务指南 → FortiCare 服务介绍 ↗章节。

在线更新特征库

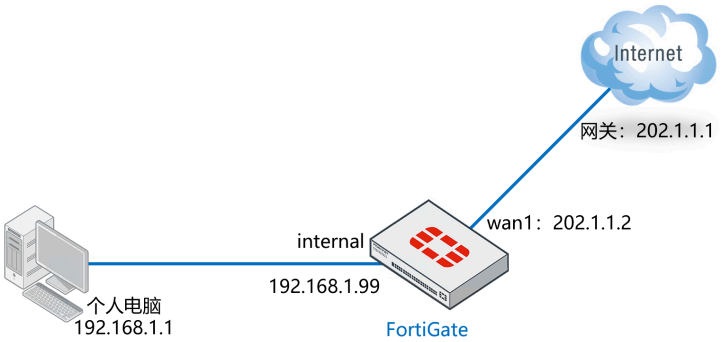
重要

- Web 分类过滤、垃圾邮件库说明：Web 分类过滤、垃圾邮件是存放在云端，其它库是下载在防火墙本地。
- 只有被防火墙策略引用的 UTM 功能的特征库才会自动更新，没有被引用的 UTM 功能特征库（或策略引用了 UTM 功能，但处于 disable 状态），不会自动更新（版本号会一直保持在旧版本或 0.000）。
- IoT 与 OT 特征库存在于在应用控制特征库中，需要防火墙策略引用应用控制配置文件后才会更新。
- ISDB 以及地理位置等非 UTM 特征库默认就会自动更新。

需求

防火墙需要每隔两个小时就升级更新病毒库、应用特征库、入侵防御特征库。同时能够离线升级更新特征库。

组网拓扑



配置要点

1. 配置防火墙联通互联网。
2. 在线自动升级后有效的前提是已经购买过相关的授权许可，且已经注册服务，更新授权的操作方式请参考系统管理 → FortiGuard 管理 → 注册指南 → 服务注册步骤 ↗章节。
3. 由于升级过程中，防火墙需要从外网下载升级包，会占用网络带宽，且会影响设备的数据转发，所以建议将自动升级时间设置在在上网低峰的时候。

重要

自动升级必须保证防火墙联网，且配置正确的 DNS 地址；可是使用 FortiGuard 服务器，也可以手动设定 DNS 服务器（尽量为知名公有 DNS 地址）。

配置步骤

1. 配置防火墙联通互联网，需要配置好接口地址，路由及 DNS，详细步骤请参考网络管理 → 路由模式 → 静态地址线路上网配置 ↗。
2. 定期及自动升级，进入系统管理 → FortiGuard，勾选“定期更新”，并设置每天凌晨 3 点更新。

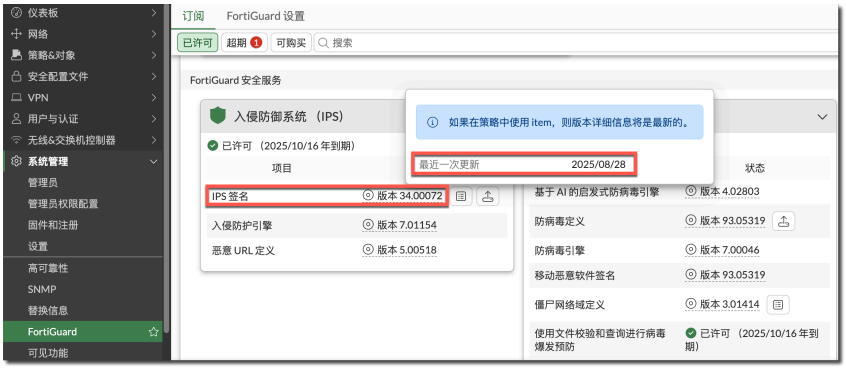



```bash
config system autoupdate schedule
    set frequency daily
    set time 03:60
end
```

重要

推荐将自动更新的周期设置为 1 天或 1 天以上，以节省性能消耗。

3. 确认自动升级成功，进入系统管理 → FortiGuard 确认所有的特征库均已升级成功，确认更新日期。



4. 也可以点击系统管理 → FortiGuard 页面右上方的更新按钮进行手动更新（或在 CLI 下执行 `execute update-now`）。



离线更新特征库

需求

防火墙无法联网或授权过期的情况下，技术支持 中心可以临时提供入侵（IPS）、应用（APP）和病毒（AV）的离线特征库进行手工升级。建议尽快购买授权，实现在线自动升级，确保库文件及时更新，提升安全性。

> **重要**
>
> Web Filter 分类过滤/DNS Filter 分类过滤/基于 FortiGuard 的 DLP/视频分类过滤等基于 FortiGuard 分类的服务，需购买相关授权并连接 FortiGuard 实时查询（设备需要能够连接 Internet）。

配置要点

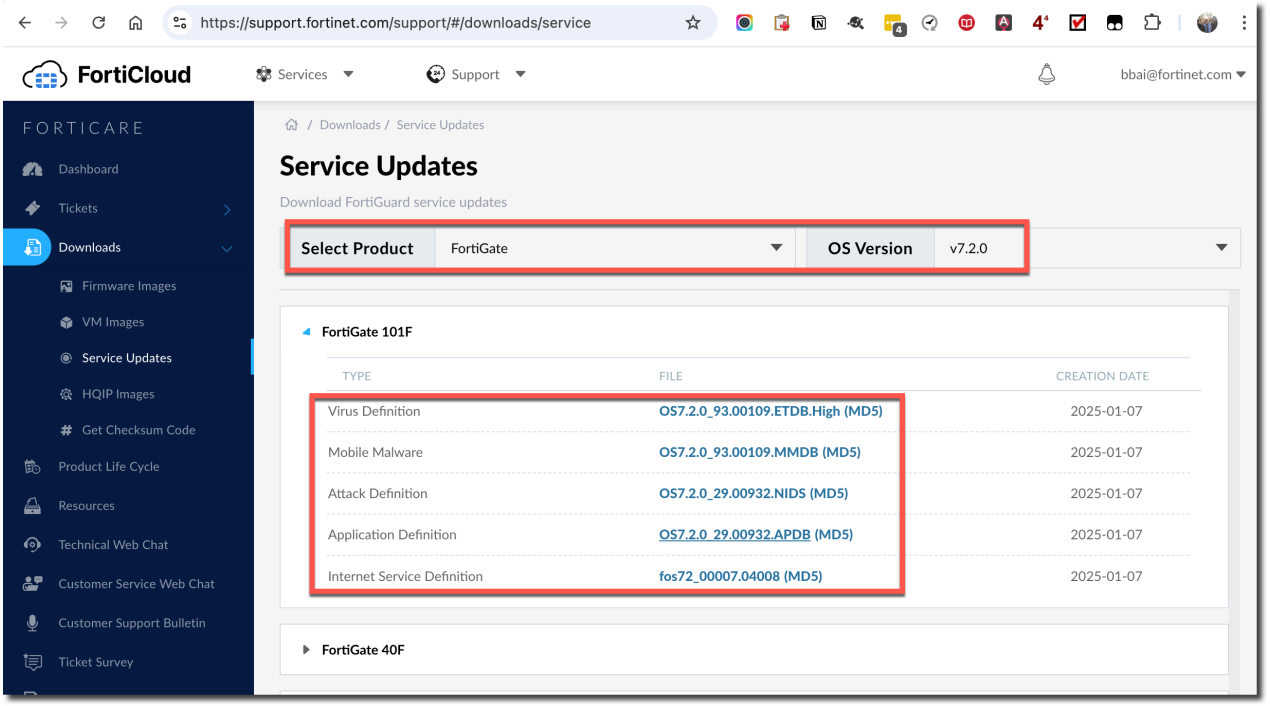
1. 获取离线特征库。
2. 手工更新。
3. 确认升级成功。

配置步骤

Web 方式离线升级

1. 获取离线特征库，访问并登录 https://support.fortinet.com/support/#/downloads/service ↗ 网站离线获取特征库文件。有服务的设备才有权限下载特征库。IPS 和 AV 的引擎是不能从这里下载的，如果需要最新的引擎，需要从技术支持处获取。

   > **重要**
   >
   > 离线特征库版本与 FortiOS 版本存在匹配关系。请在“OS Version”处选择正确的 FortiOS 版本，否则离线特征库会导入失败。

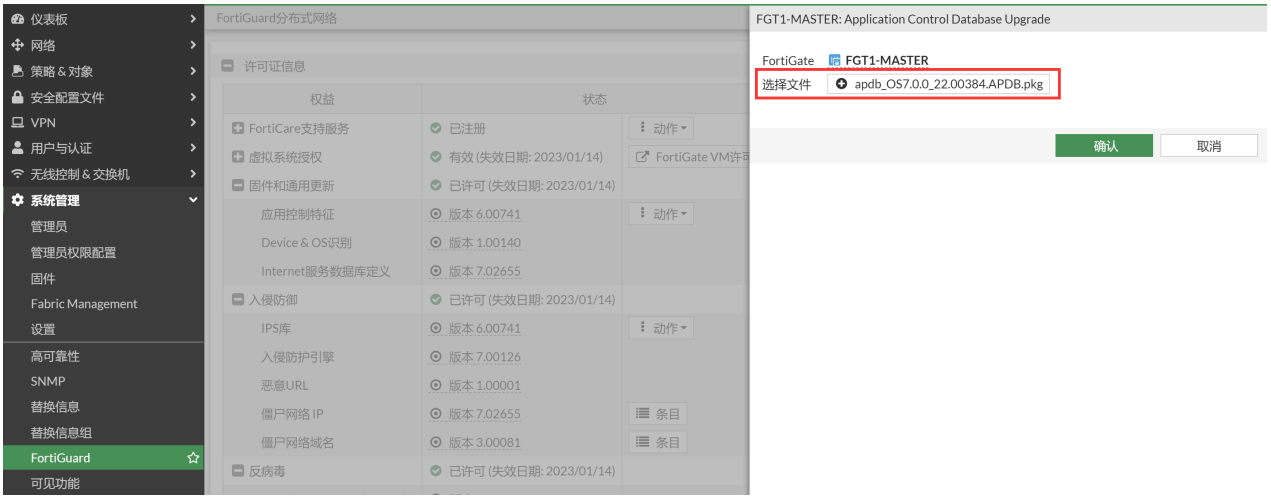
   

   - vsigupdate-OS7.0.0_90.05756.ETDB.High.pkg：AV 特征库
   - nids_OS7.0.0_22.00386.NIDS.pkg：IPS 特征库
   - apdb_OS7.0.0_22.00384.APDB.pkg：应用控制特征库
   - ffdb_fos70_00007.02656.pkg：ISDB 特征库

2. 下载对应的特征库定义后，进入系统管理 → FortiGuard，点击右侧的动作 → 升级数据库按钮，分别升级 ISDB、应用控制、病毒特征库、IPS 特征库。



3. 点击相应特征库后面的升级数据库，点击“上传”选择本地的离线特征库文件，确定即可。



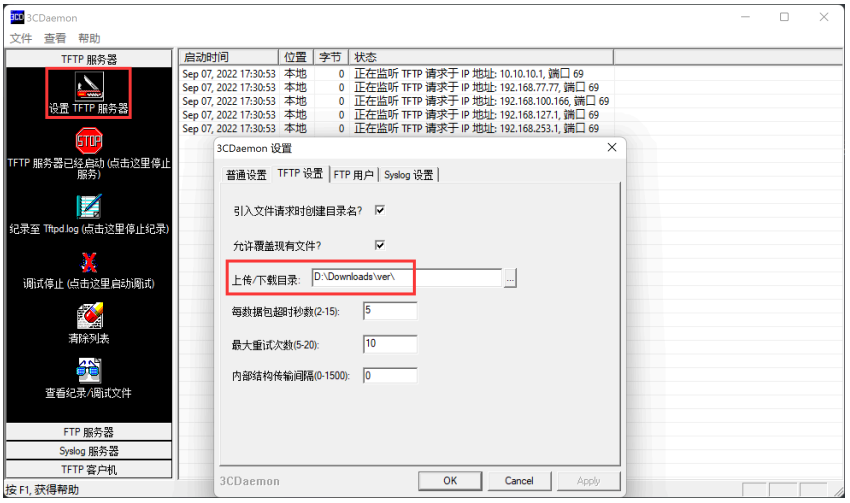
4. 分别导入到防火墙里升级完后，不需要重启防火墙，升级完毕之后页面会提示升级成功。更新特征库和引擎都不会影响到业务的正常运行。



CLI 方式 TFTP/FTP 升级

1. 也可以通过 TFTP/FTP 的方式进行升级，我们这里以 TFTP 升级 IPS 数据库为例。

2. 配置 TFTP 服务器，这里以 3CDaemon 为例，配置 TFTP 服务器的路径为特征库所在路径。



3. 在 FortiGate 的 CLI 下执行更新 IPS 特征库命令。

```bash
# exe restore ?
av                  av
config              Restore config.
image               image
ips                 ips
ipsuserdefsig       ipsuserdefsig
other-objects       other-objects
script              script
secondary-image     secondary-image

FortiGate #  execute restore ips tftp nids_OS7.0.0_22.00386.NIDS.pkg 192.168.100.166
This operation will overwrite the current IPS package!
Do you want to continue? (y/n)y
Please wait ...
Connect to tftp server 192.168.100.166  ...
##
Get IPS database from tftp server OK.
```

4. 提示更新成功后，使用 `diagnose autoupdate versions` 查看“Attack Definitions”的信息，显示上次更新为 Manual 模式，版本号与特征库文件一致。

```bash
# diagnose autoupdate versions
...
Attack Definitions
------------
Version: 22.00386
Contract Expiry Date: Sun Mar 26 2023
Last Updated using manual update on Wed Sep  7 17:56:43 2022
Last Update Attempt: Tue Aug 30 19:01:35 2022
...
```

离线导入设备 License

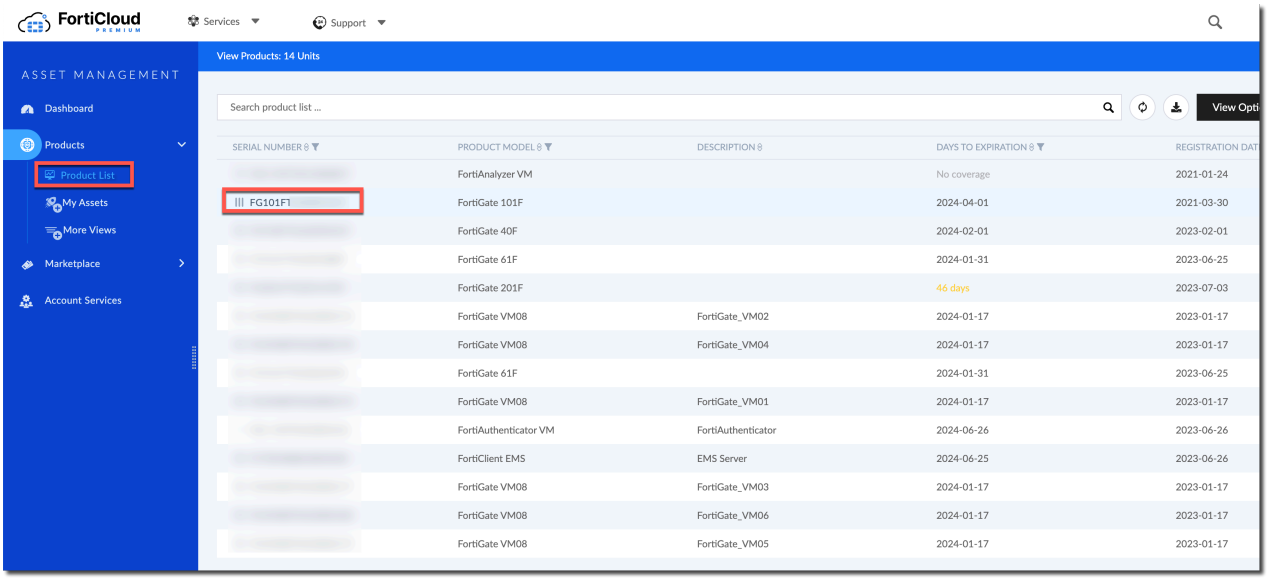
重要

该功能在 FortiOS 7.2.0 及以上版本上支持。

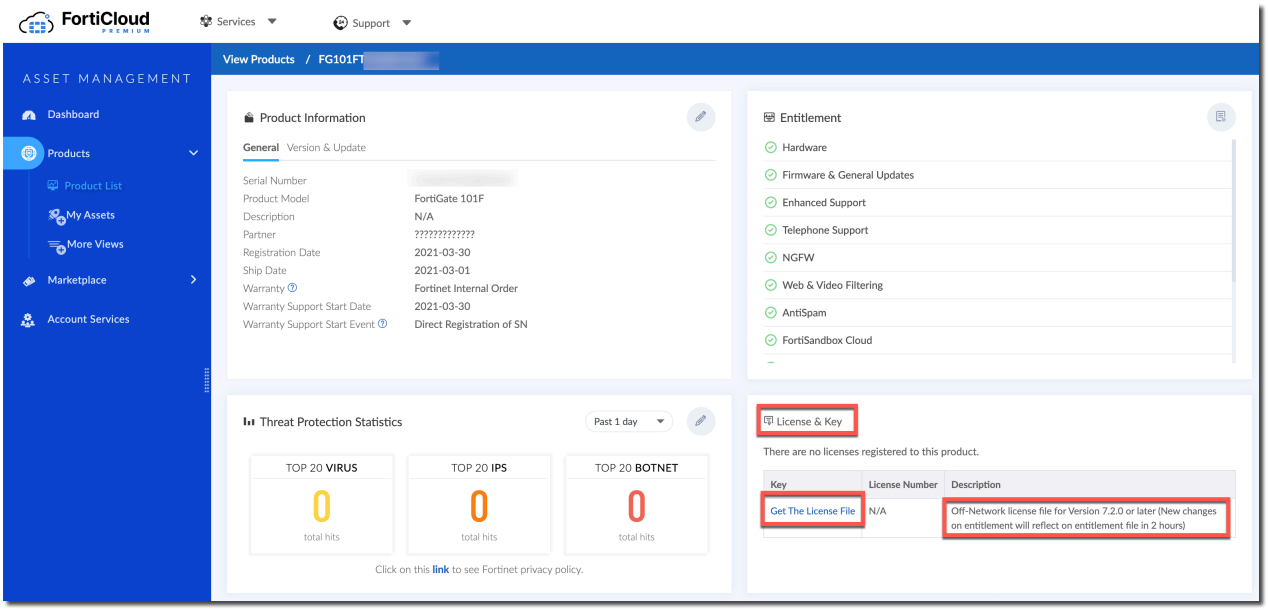
当由于安全性需要，用户的设备（硬件 FortiGate 或 VM FortiGate）不能连接 Internet，也没有 FortiManager 可以代理更新，无法通过 FortiGuard/FortiManager 同步服务 License 信息时，可以通过离线导入服务 License 的方式将 License 信息同步到 FortiGate 上。

导入过程

1. 设备与相关服务注册完成后，登录 https://support.fortinet.com/asset/#/views/products ↗ 网站（使用设备注册时的账户），进入 Product List，点击要离线导入 License 的设备：

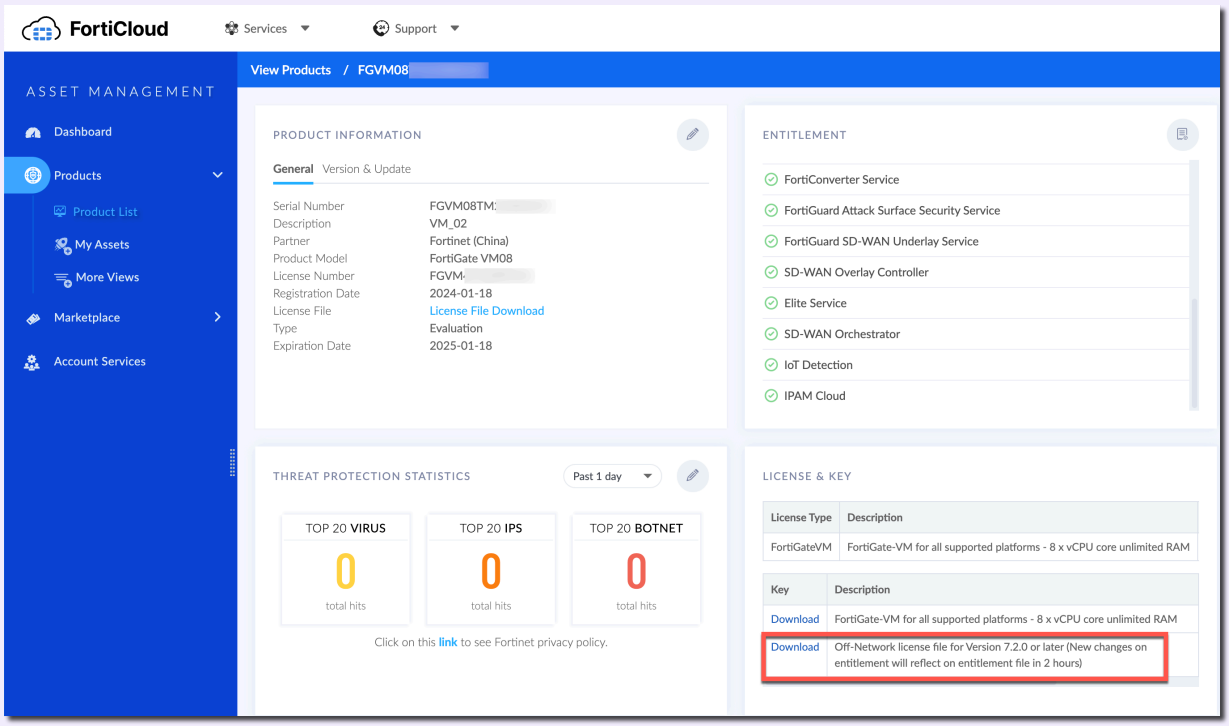


2. 在 License & Key 模块中，点击“Get the License File”按钮，下载设备的 License 文件（注意右侧的说明，已经注册的服务需要最多 2 小时才能更新到离线 License 文件中，该功能仅支持 FortiOS 7.2 及以上版本），下载的文件格式为 FG101FTKxxxxxxxxOff-NetworkLicenseFile.lic：

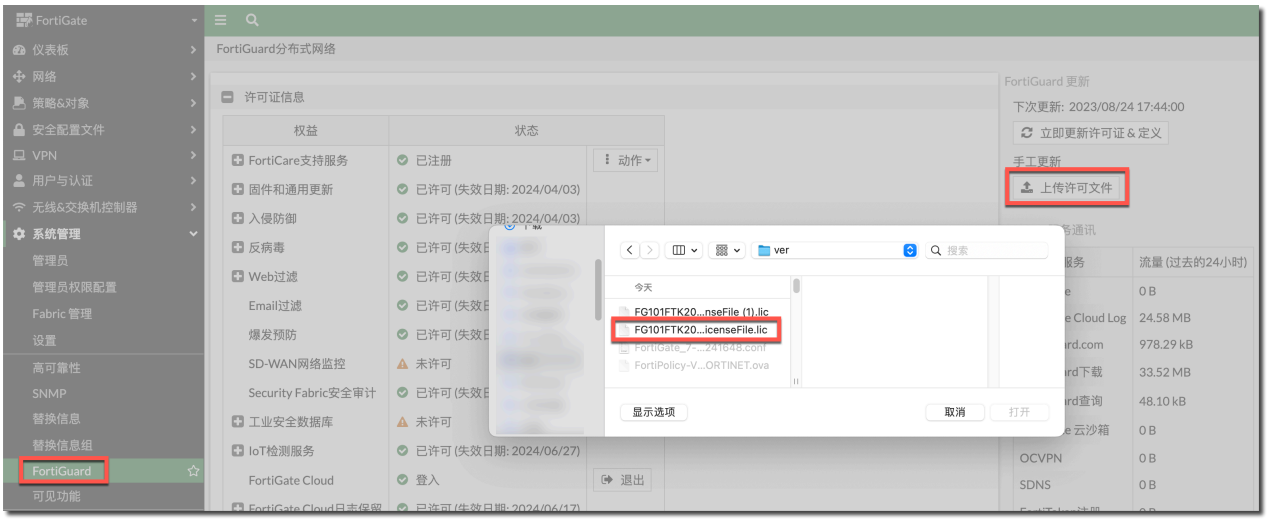


重要

如果是 FortiGate VM，请下载“Off-Network”类型的 License（注意右侧的说明，已经注册的服务需要最多 2 小时才能更新到离线 License 文件中，该功能仅支持 FortiOS 7.2 及以上版本）。下载的文件格式为 FGVMxxxxxxxxxxxxOff-NetworkLicenseFile.lic。



3. 登录设备的管理页面，进入系统管理 → FortiGuard 页面，在页面右侧点击“上传许可文件”按钮，选择要上传的离线 License 文件：



重要

VM FortiGate 需要先通过 Internet 或 FortiManager 激活一次后，才能离线更新服务 License，否则无法正常访问 GUI 页面。

4. 上传完成后，FortiGate 即可离线更新到在 Support 网站已注册的服务内容。

5. 也可以通过 CLI 通过 FTP/TFTP 离线更新 License 文件。

```
FortiGate # execute restore manual-license {ftp | tftp} <license file name> <server> [args]
```
bash

FortiGuard 更新服务器位置

功能简介

FortiGate 可以指定 FortiGuard 更新服务器的地理位置，用于控制 FortiGate 在进行以下云服务通信时所连接的服务器区域，该配置常用于优化访问延迟与稳定性或满足合规要求（如数据或流量需限定在特定区域）：

- 安全特征库更新（AV/IPS/应用控制等）。
- Web Filter/DNS Filter/Email Filter 等分类与评级服务查询。
- IoT 识别与相关云服务。

可选服务器位置

FortiGate 当前支持 3 种更新服务器位置策略。

最低延迟配置（automatic）

推荐配置，FortiGate 根据网络条件自动选择时延最低的 FortiGuard 服务器（可能位于不同区域）。

- 适用场景：
  - 硬件 FortiGate 的默认配置。
  - 追求最优性能与更新/查询速度。
- Anycast 域名：
  - `globalupdate.fortinet.net`
  - `globalguardservice.fortinet.net`
- 非 Anycast 域名：
  - `update.fortiguard.net`
  - UDP：`service.fortiguard.net`
  - HTTPS：`securewf.fortiguard.net`

仅欧洲（eu）

FortiGate 仅使用位于欧盟区域的 FortiGuard 服务器。

- 适用场景：
  - 欧盟数据合规（如 GDPR）。
  - 明确要求 FortiGuard 相关流量不得出 EU。
- Anycast 域名：
  - `euupdate.fortinet.net`
  - `euguardservice.fortinet.net`

仅美国（us）

FortiGate 仅使用位于美国的 FortiGuard 服务器。

- 适用场景：
  - 私有云/VM FortiGate 的默认配置。
  - 网络出口明确位于美国。
- Anycast 域名：
  - `usupdate.fortinet.net`
  - `usguardservice.fortinet.net`
- 非 Anycast 域名：
  - `usupdate.fortiguard.net`
  - UDP：`usservice.fortiguard.net`
  - HTTPS：`ussecurewf.fortiguard.net`

默认行为

设备类型	默认更新服务器位置
硬件 FortiGate	最低延迟配置（automatic）
FortiGate VM	美国（us）

配置方法

在“系统管理 → FortiGuard”页面的“更新服务器位置”选项下进行配置。




```bash
config system fortiguard
    set update-server-location {automatic | usa | eu}
end
```

注意事项

- 推荐使用“最低延迟配置”，追求最优性能与更新/查询速度。
- 若 FortiGuard 更新/查询异常，可参考：系统管理 → FortiGuard 管理 → FortiGuard 连接问题 ↗章节。

查看合同/更新状态

简介

在 FortiGate 上可以使用如下 CLI 命令 `diagnose test update info contract` 综合查看以下信息：

- 设备是否已成功注册到 FortiGuard/FortiCloud。
- 当前 FortiCare/FortiGuard 合同是否有效，以及到期时间。
- 最近一次 FortiGuard 自动更新的结果和日志。
- 各类安全数据库（AV、IPS、URL、应用控制等）的版本情况。
- 哪些 FortiGuard 组件已授权，哪些是未购买/未授权（unauthorized）。

在排查授权问题、FortiGuard 更新异常、固件升级受限（FMWR） 等场景时，这个命令非常关键。

查看方法

在 FortiGate CLI 中执行 `diagnose test update info contract`，如下所示，输出结果主要分为几个部分：Logs、System contracts、Account contracts、Object versions、Counters、Support contract 等：

- Logs：看最近自动更新是否成功。
- System contracts：看这台设备实际拥有哪些授权，何时到期（特别是 FMWR）。
- Account contracts：看 FortiCloud 账户层面的额外订阅（如 Sandbox、Endpoint）。
- Object versions & counters：确认数据库版本和更新任务运行是否正常。
- Support contract：确认设备是否已成功绑定到 FortiCloud 账户并获取合同信息。

```bash
FortiGate # diagnose test update info contract

Logs: idx=17
Thu Nov 27 20:43:37 2025 do_virus_report[886]-Starting VIRUS REPORT
Thu Nov 27 20:43:37 2025 do_virus_report[903]-No virus stats to report
Thu Nov 27 21:43:37 2025 do_virus_report[886]-Starting VIRUS REPORT
Thu Nov 27 21:43:37 2025 do_virus_report[903]-No virus stats to report
Thu Nov 27 22:43:39 2025 do_virus_report[886]-Starting VIRUS REPORT
Thu Nov 27 22:43:39 2025 do_virus_report[903]-No virus stats to report
Thu Nov 27 23:43:42 2025 do_virus_report[886]-Starting VIRUS REPORT
Thu Nov 27 23:43:42 2025 do_virus_report[903]-No virus stats to report
Fri Nov 28 00:43:46 2025 do_virus_report[886]-Starting VIRUS REPORT
Fri Nov 28 00:43:46 2025 do_virus_report[903]-No virus stats to report
Fri Nov 28 01:43:48 2025 do_virus_report[886]-Starting VIRUS REPORT
Fri Nov 28 01:43:48 2025 do_virus_report[903]-No virus stats to report
Fri Nov 28 02:43:51 2025 do_virus_report[886]-Starting VIRUS REPORT
```

自动更新与病毒报告日志

```bash
Logs: idx=17
Fri Nov 28 03:30:00 2025 do_update-Starting scheduled UPDATE (not final retry)
Fri Nov 28 03:30:02 2025 do_update-UPDATE successful

Fri Nov 28 11:44:19 2025 do_virus_report-Starting VIRUS REPORT
Fri Nov 28 11:44:19 2025 do_virus_report-No virus stats to report
```

如果怀疑 FortiGate 没有正常从 FortiGuard 拉取更新，先看这里是否有连续失败的记录。

- `do_update`：
  - 表示 FortiGuard 自动更新任务。
  - 有 `Starting scheduled UPDATE` 且紧跟 `UPDATE successful` 说明本次自动更新成功。
  - 这里可以快速判断最近是否存在更新失败、反复重试等情况。
- `do_virus_report`：
  - 表示“病毒统计上报”任务。
  - `No virus stats to report` 意味着目前没有病毒事件需要上报。

设备授权状态

这里列出的是绑定在当前 FortiGate 序列号上的系统级合同，后面是到期时间。重点关注：

- 是否存在 FMWR，以及 FMWR 的到期时间。在 FortiOS 7.4 及以后，如果 FMWR 已过期，设备可能会被限制升级到更高版本。
- 其他你实际使用的安全功能（如 AVDB、NIDB、FURL、FRVS）是否都在有效期内。

```bash
System contracts:
    HDWR,Tue Sep 29 2026
    ENHN,Tue Sep 29 2026
    COMP,Tue Sep 29 2026
    FMWR,Tue Sep 29 2026
    FURL,Tue Sep 29 2026
    SPAM,Tue Sep 29 2026
    SBCL,Tue Sep 29 2026
    ZHVO,Tue Sep 29 2026
    SPRT,Tue Sep 29 2026
    FRVS,Tue Sep 29 2026
     ...

    AVDB,Tue Sep 29 2026
    NIDB,Tue Sep 29 2026
    CASB,Tue Sep 29 2026
    WIPS,Tue Sep 29 2026
    WAPP,Tue Sep 29 2026
```

- `FMWR`：Firmware & General Updates（固件与通用更新授权）。
- `AVDB/AVEN/MMDB/FLDB`：防病毒引擎与特征库。
- `NIDB/NIET/NIML/NIEN`：IPS/入侵检测相关数据库。
- `FURL`：URL/DNS/Video Filtering 服务。
- `FRVS`：漏洞管理/IPS 相关服务。
- `SPAM`：AntiSpam。
- `ZHVO`：Virus Outbreak Protection。
- `CASB/WIPS/WAPP`：云应用安全、无线 IPS、Web 应用保护。
- `HDWR/SPRT/COMP/ENHN`：硬件保修与 FortiCare 支持等级。

FortiCloud 账户授权

这里是绑定在 FortiCloud 账户上的合同，如下所示。这些授权可能被多台设备共用，如果使用了 FortiSandbox Cloud、FortiClient EMS Cloud 等云服务，这里的有效期就非常关键。

```bash
Account contracts:
    FSAP,Wed Jun 19 2024
    FCEP,Wed Jun 19 2024
```

- `FSAP`：FortiSandbox Cloud/高级沙箱服务等账户级订阅。
- `FCEP`：FortiClient EMS/Endpoint Protection 相关订阅。

安全数据库版本

每一条对应一个 FortiGuard 对象（AV 库、IPS 库、URL 分类库等）的版本号。

```bash
Object versions:
    07006000AVDB00201-00093.05319-2508121826
    07006000MMDB00101-00093.05319-2508121828
    07006000NIDS02805-02507.00207-2507300100
    07006000UWDB00100-00004.00786-2511270202
     ...
```

- 确认当前是否已经更新到较新的数据库版本。
- 与 Fortinet 发布说明或 TAC 建议的版本号进行对比。
- 排查个别对象版本异常、停留在很早的版本不动等情况。
- 日常运维通常不用逐项分析，只需确认这些对象不是全部 `00000.00000-0000000000` 这类明显“未更新”状态即可。需要精确问题排查时，可配合 `diagnose autoupdate versions` 进行对照：

```bash
FortiGate # diagnose autoupdate versions

AV Engine
————————
Version: 7.00046 signed
Contract Expiry Date: Tue Sep 29 2026
Last Updated using scheduled update on Mon Jun 21 18:57:22 2021
Last Update Attempt: Wed Aug 13 11:10:15 2025
Result: No Updates

Virus Definitions
————————
Version: 93.05319 signed
Contract Expiry Date: Tue Sep 29 2026
Last Updated using scheduled update on Wed Aug 13 10:40:16 2025
Last Update Attempt: Wed Aug 13 11:10:15 2025
```

计数器与计划任务

```bash
Setup done once: yes
Next sched update: Sat Nov 29 03:30:00 2025
Next virus report: Fri Nov 28 14:44:24 2025
Next signature check: Fri Nov 28 14:45:56 2025

Ring         counters: pass=000000 fail=000000
Setup        counters: pass=000001 fail=000000
Update       counters: pass=000001 retry_fail=000000 final_fail=000000
Virus report counters: pass=000000 fail=000000 empty_stats=000021
```

- `Setup done once: yes`：表示初始 FortiGuard/合同信息同步成功完成过一次。
- `Next sched update/Next signature check`：
  - 下次计划执行自动更新/签名检查的时间。
  - 如果设备长期在线，但这里一直没有“下次任务”，可能需要排查 autoupdate 配置。
- `Update counters`：
  - `pass`：更新成功次数。
  - `retry_fail` / `final_fail`：自动更新重试、多次失败情况。
  - 如果失败计数不断增加，需要排查 DNS、与 FDS 的连通性等。
- `Virus report counters`：`empty_stats` 较大且 `fail=0`，多数是正常情况。

合同与账号绑定信息

```bash
Support contract: pending_registration=255 got_contract_info=1
    account_id=[user@example.com] company=[Example] industry=[Technology]

SerialNumber=FGxxxxxxxxxxxxx|Contract=AVDB-1-06-20260930:0:1:1:0*...
SupportLevelDesc=05:Advanced HW*06:Web/Online*20:Premium
SupportTypeDesc=AVDB:Advanced Malware Protection*FMWR:Firmware & General Updates*...

User ID: 1134101
```

- 设备已成功注册到某个 FortiCloud/FortiCare 账户。
- `SupportLevelDesc`：
- 05 对应 Advanced Hardware 服务。
- 06 对应 Web/Online 支持。
- 20 对应 Premium 支持等级。
- `SerialNumber=...|Contract=...` 是机器可读的合同详情，包含 SKU、支持等级、到期日等。
- `User ID` 是对应 FortiCloud 账号的内部 ID。
- 如果这里 `account_id` 为空且 `got_contract_info=0`，通常意味着设备尚未成功注册或没有有效合同。

unauthorized 对象

在更新日志中，可能看到类似 `unauthorized`，表示 FortiGate 知道这些对象存在，但当前设备没有相应的订阅/合同：

```bash
upd_install_pkg-DLDB003 is unauthorized
upd_install_pkg-SLAD000 is unauthorized
upd_install_pkg-FTLM000 is unauthorized
```

- 未授权对象不会更新，但不会影响你已购买服务的正常更新。
- 常见于某些高级功能、额外模块（例如 DLP、评分等相关组件）。
- 只要 AVDB/NIDB/FURL/FRVS 等你实际依赖的服务处于 `installed successfully` 或 `up-to-date` 状态，看到少数 `unauthorized` 一般是正常情况。

其他命令

- `get system fortiguard-service status`：快速查看 FortiGuard 安全模块版/特征库版本、最近更新时间/方法、合同过期时间。

```bash
FortiGate # get system fortiguard-service status
NAME                                    VERSION       LAST UPDATE           METHOD     EXPIRE
AV Engine                               7.048 signed  2021-06-21 18:57:22   scheduled  2026-09-29 23:59:59
Virus Definitions                       93.6414 signed  2026-01-09 03:29:31   scheduled  2026-09-29 23:59:59
Extended set                            93.6414 signed  2026-01-09 03:29:31   scheduled  2026-09-29 23:59:59
AI/Machine Learning Malware Detection Model  4.3550 signed  2026-01-09 03:29:31   scheduled  2026-09-29 23:59:59
Flow-based Virus Definitions            93.6414 signed  2026-01-09 03:29:31   scheduled  2026-09-29 23:59:59
Attack Definitions                      6.741 signed  2025-06-27 09:49:50   scheduled  2026-09-29 23:59:59
Attack Extended Definitions             35.157 signed  2026-01-27 03:45:40   scheduled  2026-09-29 23:59:59
AI/Machine Learning IPS Definitions     2507.207 signed  2025-08-28 10:54:15   manual     2026-09-29 23:59:59
IPS Malicious URL Database              5.620 signed  2026-01-27 03:45:40   scheduled  2026-09-29 23:59:59
IPS/FlowAV Engine                       7.1168 signed  2025-03-26 11:21:19   scheduled  2026-09-29 23:59:59
Application Definitions                 6.741 signed  2025-06-27 00:47:02   scheduled  2026-09-29 23:59:59
GenAI Application Definitions           0.000         2001-01-01 00:00:00   manual     2026-09-29 23:59:59
FMWP Definitions                        0.000         2025-08-13 01:36:06   scheduled  2026-09-29 23:59:59
OT Threat Definitions                   6.741 signed  2015-12-01 02:30:00   manual     n/a
IoT Detect Definitions                  0.000 signed  2022-08-17 17:31:00   manual     n/a
OT Detect Definitions                   0.000         2001-01-01 00:00:00   manual     n/a
OT Patch Definitions                    0.000         2001-01-01 00:00:00   manual     n/a
Proxy Attack Definitions                6.741 signed  2025-06-27 09:49:50   scheduled  2026-09-29 23:59:59
Proxy Attack Extended Definitions       35.157 signed  2026-01-27 03:45:40   scheduled  2026-09-29 23:59:59
Proxy Application Definitions           6.741 signed  2025-06-27 00:47:02   scheduled  2026-09-29 23:59:59
```

- `diagnose sys service-communication`：显示与 FortiGuard 服务通信的历史统计数据。

```bash
FortiGate # diagnose sys service-communication
FortiCare:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard Download:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard Query:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiCloud Log:
The last 1 hour(in bytes):  12453 36865 34114 41402 35643 59516 31634 59258 49278 44744 36840 26831
The last 24 hours(in bytes):  251627 442265 390826 289650 268367 271030 279281 285316 279659 275600 264113 275084 274777
284653 269965 286750 288521 368999 493300 508134 385346 376882 290777 268681
The last 7 days(in bytes):  3572818 7400892 5849461 5894928 6431629 6128193 8570799
FortiSandbox Cloud:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard.com:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 634 0 0 0 0
The last 24 hours(in bytes):  0 2034 634 634325 634 501507 634 634 10862911 634 634 2034 634 632937 2300 634 634 497730
634 2034 498964 5095 634 632941
The last 7 days(in bytes):  12006615 4113270 13058044 14905016 3863067 9454250 3906753
SDNS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiToken Registration:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
The last 7 days(in bytes):  0 0 0 0 0 0 0
SMS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
```

排查思路

- 检查设备是否正确注册且有合同：
  - 查看 Support contract 部分：
    - 是否有 `account_id`。
    - `got_contract_info` 是否为 1。
  - 看 System contracts 是否列出了 FMWR/AVDB/NIDB 等关键服务且未到期。
- 固件无法升级/提示无授权：
  - 在 System contracts 中确认是否存在 FMWR。
  - 检查 FMWR 到期日是否已过。
  - 如已过期，联系销售确认续约或更换合同。
- 怀疑 FortiGuard 更新失败：
  - 在 Logs 部分查找 `do_update`，确认是否有 `UPDATE successful`。
  - 查看 Update counters 中是否有大量 `retry_fail` / `final_fail`。
  - 必要时配合如下调试命令进一步抓取更新过程日志（完成后记得关闭 debug）：

    ```bash
    diagnose debug application update -1
    diagnose debug enable
    ```

- 某些功能提示“license missing”：
  - 对照 System contracts/Account contracts 部分，确认是否没有相关订阅。
  - 对照更新日志中的 `is unauthorized`，核实对应对象是否就是提示缺少的功能组件。

FortiGuard 连接问题

FortiGuard 通信机制

FortiGuard 并不是“只要买了/注册了许可证就自动生效”的服务。FortiGate 要正常使用 FortiGuard，必须同时满足以下四个条件：

1. 许可证在 Fortinet 官方支持平台（support.fortinet.com ↗）中真实有效。
2. FortiGate 能够与 FortiGuard 服务器建立通信。
3. 许可证信息已成功同步到 FortiGate 设备本地。
4. 相关安全功能（如 IPS/AV）在策略中被实际启用并使用。

只要其中任意一个条件不满足，就可能出现“许可证异常”或“特征库不更新”的现象。

更新常见问题

- Fortinet 官方支持平台（support.fortinet.com ↗）中显示许可证仍然有效，但 FortiGate GUI 显示 FortiGuard 服务或许可证已过期。
- 防病毒（AV）、入侵防御（IPS）等安全功能特征库长时间未更新。
- FortiGate 无法成功连接 FortiGuard 更新服务器。

上述情况并不一定代表许可证真实失效，而通常与许可证同步机制、网络连通方式或更新路径受限有关。

问题原因

- 许可证与设备同步问题。Fortinet 官方支持平台（support.fortinet.com ↗）中显示许可证仍然有效，但 FortiGate GUI 显示 FortiGuard 服务或许可证已过期。
  - 许可证刚完成激活或续费，FortiGuard 后端同步尚未完成（通常需要 24~48 小时）。
  - HA 集群中存在成员设备未注册或存在已到期服务，导致集群整体被判定为许可证异常。
  - FortiGate 设备长时间未与 FortiGuard 成功通信，本地的许可证状态未刷新。
- 无法访问 FortiGuard。
  - 当 FortiGate 无法与 FortiGuard 服务器建立连接时，特征库和许可证状态均无法更新。
  - FortiGate 无法直连互联网，或仅允许有限出口访问。
  - FortiGate 自发流量（更新请求、DNS 查询）被错误地引导进入错误接口（如无法访问 Internet 的 VPN 接口）。
  - 上游防火墙、ISP 或云环境限制 FortiGuard 所需的端口或目标地址。
  - Anycast 启用状态与网络实际出口策略不匹配。
- DNS 解析异常或被劫持。
  - FortiGuard 更新依赖域名访问，即使 IP 层可达，DNS 解析失败仍会导致更新失败。
  - FortiGate 使用的 DNS 服务器无法解析 FortiGuard 相关域名。
  - DNS 查询被重定向、污染或拦截。
  - DNS 服务器仅允许内网解析，未放行外部域名。
- FortiGuard 查询/更新端口受限。端口受限时，更新请求可能发出但无法建立有效会话。
  - Web Filter/DNS Filter 所需端口（53/443/8888）被防火墙或 ISP 阻断。
  - FortiGate 使用的源端口范围被上游设备限制。
  - Anycast 启用时，部分区域 FortiGuard 节点不可达。
- FortiGuard 功能未被实际启用。如果对应的安全功能未在防火墙策略中应用，即使许可证在本地有效，FortiGate 不会触发对应特征库的下载与更新。
  - AV/IPS 配置文件未绑定到任何防火墙策略。
  - 对应安全功能在“可见功能”中被隐藏或未启用。
- 受限网络或隔离网络架构。在受限网络中，必须额外配置 FortiGuard 通信路径，否则更新将始终失败。
  - FortiGate 部署在无互联网或严格隔离网络中。
  - 未配置 FortiManager 本地 FDS。
  - 未通过代理服务器访问 FortiGuard。

排查步骤

排查 FortiGuard 问题必须同时检查许可证、网络连通性和更新配置，而非仅依据 GUI 显示结果判断。

手动触发更新

> **重要**
>
> 若 AV/IPS Profile 未绑定到任何策略，即使许可证有效，特征库也不会更新。

- GUI：进入“系统管理 → FortiGuard”页面，点击“立即更新许可证和定义”按钮。



- CLI：执行 `execute update-now` 命令。
- 查看同步过程的 Debug 信息，如果更新失败，Debug 信息中会提示对应的失败原因：

```bash
diagnose debug reset
diagnose debug application update -1
diagnose debug enable
execute update-now

完成后执行：
diagnose debug disable
```

- 其他更细化的手动更新命令：

```bash
execute update-av
execute update-ips
execute update-geo-ip
execute update-list
execute update-external-resource
execute update-ffdb-on-demand
```

检查 HA 成员许可证

- HA 集群中所有成员必须有有效许可证。
- 所有设备必须绑定在同一个 FortiCare 账号下。
- GUI 只会显示 HA 成员中最早到期的许可证时间。
- 若 HA 成员设备未注册，HA 集群许可证将被判定为无效。

检查自动更新与 FDS 状态

```bash
FortiGate # diagnose autoupdate status
FDN availability:  available at Wed Dec 24 03:04:03 2025
                   last successful time: Wed Dec 24 03:04:03 2025
Scheduled update: enable
        Update daily:   at 3 after 60 minutes
Virus definitions update: enable
IPS definitions update: enable
DLP Signature status:
        DLP signature json file not found.
```

- `FDN availability`：是否可访问 FortiGuard 网络。
- `Scheduled update`：是否启用定时更新。
- `Virus definitions update` / `IPS definitions update`：更新是否开启。
- `Web proxy tunneling`：是否通过代理访问 FortiGuard。

查看特征库/引擎状态

```bash
FortiGate # diagnose autoupdate versions

AV Engine
————————
Version: 7.00048 signed
Contract Expiry Date: Tue Sep 29 2026
Last Updated using scheduled update on Mon Jun 21 18:57:22 2021
Last Update Attempt: Wed Aug 13 11:10:15 2025
Result: No Updates
```

```
Virus Definitions
------------------
Version: 93.05319 signed
Contract Expiry Date: Tue Sep 29 2026
Last Updated using scheduled update on Wed Aug 13 10:40:16 2025
Last Update Attempt: Wed Aug 13 11:10:15 2025
```

- 各 AV/IPS/ISDB/URL DB 的版本号。
- 合约到期时间。
- 最近更新时间。
- 实际使用的 FortiGuard 地址（IP: Port）。

查看合同/更新状态

> **相关信息**
>
> 详情参考：系统管理 → FortiGuard 管理 → 查看合同/更新状态↗章节。

```bash
diagnose test update info
```

检查 FortiGuard 联通性

1. 指定 ping 命令的出接口和源 IP 为可以正常访问 Internet 的出接口和源 IP，并执行 ping 以下 FortiGuard 相关域名，查看 DNS 解析与 FortiGuard 服务器基础联通性。

   ```bash
   execute ping-options interface wan1
   execute ping-options source 202.103.1.1

   execute ping service.fortiguard.net
   execute ping update.fortiguard.net
   execute ping guard.fortinet.net
   execute ping securewf.fortiguard.net
   ```

2. 若 DNS 解析失败，可能是 FortiGate 无法正常访问 DNS 服务器，或 FortiGate 设置的 DNS 服务器无法正常解析 FortiGuard 相关域名，需要检查 DNS 相关配置，建议使用公网知名可信的 DNS 服务器，避免 DNS 域名污染。

   > **相关信息**
   >
   > 详情参考：网络管理 → DNS → DNS 配置 ↗章节。

   ```bash
   config system dns
       set primary 223.5.5.5
       set secondary 114.114.114.114
   end
   ```

3. 若 DNS 解析成功，但 Ping 失败，可能是由于上游设备、ISP 或云环境限制 FortiGuard 所需的端口或目标地址。需要在上游设备、ISP 或云环境排查。

4. 若 DNS 解析成功，Ping 也可以成功，但 FortiGuard 更新仍然失败，则可能是 FortiGuard 更新流量走错了接口。

   - 默认配置下，FortiGuard 更新流量是 FortiGate 自身发起的流量，依赖路由表自动选择出接口与源 IP，如果默认路由在多个接口上负载，其中有接口无法正常访问 Internet，则可能会导致 FortiGate 连接 FortiGuard 服务器失败（`diagnose debug application update -1` 会显示实际更新时使用的出接口）。

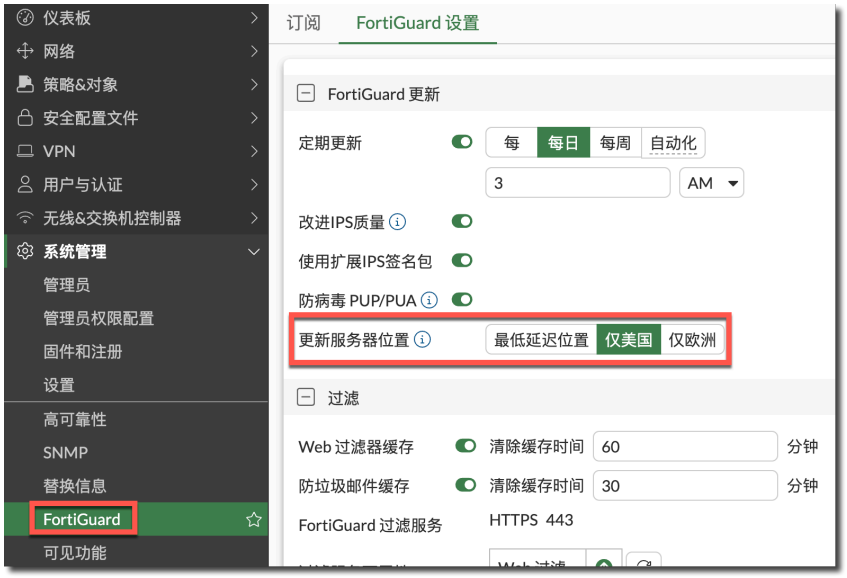
   - 如下所示，FortiGate 的默认路由在 wan1 和 vpn_line1 之间负载，wan1 可以正常访问 Internet，vpn_line1 是 IPSec 隧道，无法访问 Internet。

     ```bash
     FortiGate-101F # get router info routing-table all
     ......

     Routing table for VRF=0
     S*      0.0.0.0/0 [1/0] via 202.103.1.1, wan1, [1/0]
     S*                [1/0] via 169.254.10.1, vpn_line1, [1/0]
     ......
     ```

- 解决方法是在 FortiGuard 配置中手工指定出接口，与源 IP（可选，默认会使用出接口 IP），`interface-select-method` 也可以使用 SD-WAN 规则来选路。

```bash
config system fortiguard
    set interface-select-method specify
    set interface wan1
    set source-ip 202.103.1.12
end
```

- 如果仍然更新失败，`diagnose debug application update -1` 提示与服务器连接失败，可以尝试调整更新服务器的位置：
- 进入“系统管理 → FortiGuard → FortiGuard 设置”页面，更新服务器位置默认为“仅美国”。



- 可尝试修改为“最低延迟位置”或“仅欧洲”后，再进行更新尝试。

```bash
config system fortiguard
    set update-server-location { usa | eu | automatic }
end
```

Web/DNS Filter 查询

Web/DNS Filter 分类服务是基于 FortiGuard 服务器实时查询获取分类结果的，查询分类时默认会使用这 3 个端口之一：TCP 53/443/8888。当 Web/DNS Filter 分类服务失败时，可以尝试以下步骤排查。

1. 使用如下命令可以查看当前 FortiGuard 分类服务器的状态。

```bash
FortiGate # diagnose debug rating
Locale       : simch

Service      : Web-filter
Status       : Enable
License      : Contract

Service      : Antispam
Status       : Disable

Service      : Virus Outbreak Prevention
Status       : Disable

Num. of servers : 3
Protocol        : https
Port            : 443
Anycast         : Disable
Default servers : Included

-=- Server List (Wed Dec 24 16:48:37 2025) -=-

IP                                   Weight    RTT Flags   TZ   FortiGuard-requests  Curr Lost Total Lost
```

```
Updated Time
12.34.97.18                                120    202 DI      -5                  7          0          0 Wed
Dec 24 16:48:05 2025
209.40.106.70                              120    727         -6                  1          0          0 Wed
Dec 24 16:48:05 2025
140.174.22.68                              120    873         -5                  1          0          0 Wed
Dec 24 16:48:05 2025
```

2. 默认配置下 FortIGate 使用 Anycast 方式查询 FortiGuard，当 Anycast 地址不可用时，会出现查询失败的情况。可以尝试关闭 FortiGuard Anycast 功能。

```bash
config system fortiguard
    set fortiguard-anycast disable
end
```

3. 调整 FortiGate 源端口范围（ISP 限制场景）。

4. 调整 Web/DNS Filter 分类服务查询时使用的端口，默认为 443（开启 Anycast 后只能选择 443）。

```bash
config system fortiguard
    set fortiguard-anycast disable
    set port { 8888 | 53 | 443 }
end
```

受限网络

在受限网络或隔离网络架构中，必须额外配置 FortiGuard 通信路径，否则更新将始终失败。

1. Anycast 配置相关：
   - 确认 FortiGuard anycast 状态。

     ```
     config system fortiguard
         get | grep fortiguard-anycast
     ```

   - Anycast 关闭/开启时上游设备需放行的 FQDN。

FortiGuard 服务	Anycast 关闭	Anycast 开启
FortiGuard Object download	`update.fortiguard.net`	`globalupdate.fortinet.net`
Querying service (web-filtering, anti-spam ratings) over HTTPS	`securewf.fortiguard.net`	`globalguardservice.fortinet.net`
Querying service (web-filtering, anti-spam ratings) over UDP	`service.fortiguard.net`	Service only in Unicast
Device info Collection	Service only in Anycast	`globaldevcollect.fortinet.net`
Device info Query	Service only in Anycast	`globaldevquery.fortinet.net`
FortiGate Cloud logging	`logctrl1.fortinet.com`	`globallogctrl.fortinet.net`
FortiGate Cloud management	`mgrctrl1.fortinet.com`	`globalmgrctrl.fortinet.net`
FortiGate Cloud messaging	`msgctrl1.fortinet.com`	`globalmsgctrl.fortinet.net`
FortiGate Cloud sandbox	`aptctrl1.fortinet.com`	`globalaptctrl.fortinet.net`
Product API used by OCVPN registration and GUI icon download	`productapi.fortinet.net`	`globalproductapi.fortinet.net`
FortiCare registration	`directregistration.fortinet.com`	`globalregistration.fortinet.net`
Secure DNS	`sdns.fortinet.net`	`globalsdns.fortinet.net`
FortiCloud FortiClient	`forticlient.fortinet.net`	`globalfctupdate.fortinet.net`
FortiMobile Tokens	`directregistration.fortinet.com`	`globalftm.fortinet.net`
EMS cloud	`forticlient-emsproxy.forticloud.com`	Service only in Unicast
DDNS	`ddns.fortinet.net`	`globalddns.fortinet.net`
GeoIP	`gip.fortinet.net`	`globalgip.fortinet.net`
IP blocklist	`ipbl.fortinet.net`	N/A

2. 尝试调整 FortiGate 自身发起流量使用的源端口。

```bash
config system global
    set ip-src-port-range 1024-25000
end
```

3. 使用代理服务器连接 FortiGuard 服务器。

```bash
< FortiOS 7.6.4:
config system autoupdate tunneling
    set address <proxy_address>
    set password <password>
    set port <proxy_port>
    set status {enable | disable}
    set username <username>
end

≥ FortiOS 7.6.4:
config system fortiguard
```

```
    set proxy-server-ip <proxy_address>
    set proxy-server-port <proxy_port>
    set proxy-username <username>
    set proxy-password <password>
end
```

4. 使用 FortiManager 作为 FortiGuard 更新/查询服务器。

联系 Fortinet

在以下情况下建议提交 Fortinet Ticket：

- 已确认许可证有效。
- DNS/路由/端口配置无误。
- Debug/Sniffer 显示请求异常或无响应。

提交 Ticket 时建议附带设备配置文件及以下 CLI 的输出结果：

```bash
diagnose autoupdate status
diagnose autoupdate versions
get router info routing-table all
diagnose debug application update -1
diagnose debug enable
execute update-now
```

禁用云通信

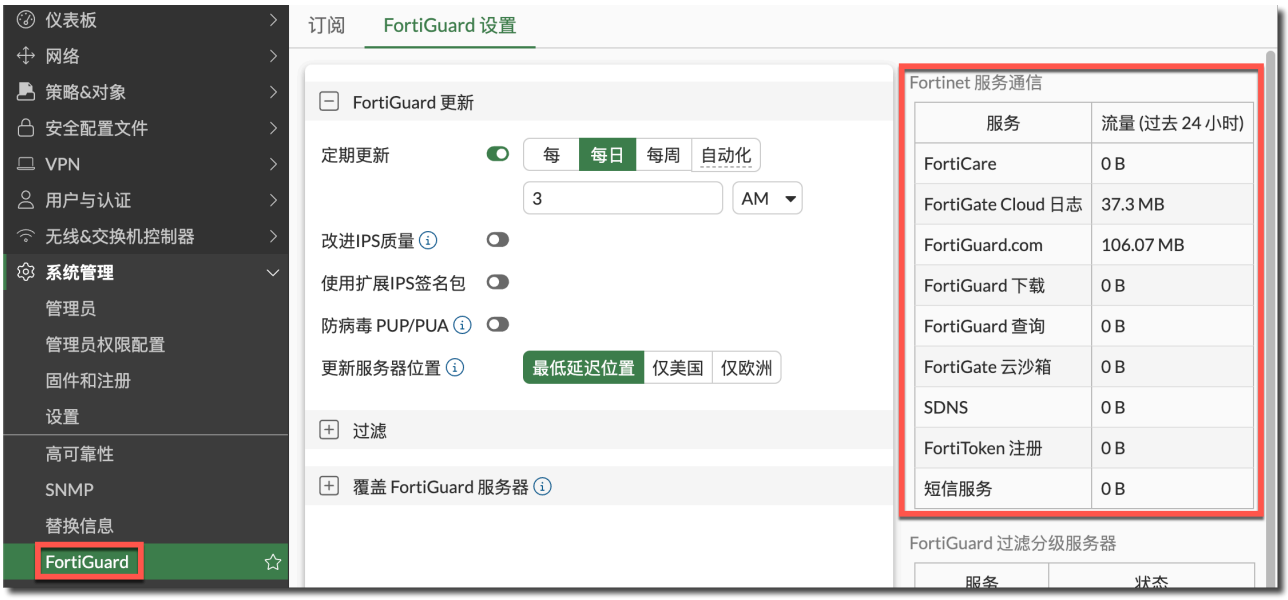
FortiGate 设备会与多个云端服务进行通信，例如 FortiGuard 下载与查询服务、FortiCloud 及其他云相关服务，用于下载服务包、执行实时安全过滤、日志上报及同步任务。

在某些安全或隔离网络（Air-Gap）环境下，管理员可能希望完全关闭这类外部通信。

云通信统计信息

GUI 查看

进入“系统管理 → FortiGuard → FortiGuard 设置”查看云通信统计计数。



CLI 查看

```bash
FortiGate # diagnose sys service-communication
FortiCare:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard Download:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard Query:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiCloud Log:
The last 1 hour(in bytes):  35541766 1103963 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  36645729 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  36645729 0 0 0 0 0 0
FortiSandbox Cloud:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard.com:
The last 1 hour(in bytes):  4851531 101213748 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  106065279 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  106065279 0 0 0 0 0 0
SDNS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
```

```
FortiToken Registration:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
SMS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
```

禁用云通信

执行以下 CLI 命令即可关闭所有云通信，当 `cloud-communication` 被禁用后，系统会自动关闭 `forticldd` 和 `updated` 守护进程，并同步禁用多项与云服务相关的功能。

提示

- 禁用云通信后，FortiGuard 实时威胁防护（例如 Web 过滤、反垃圾邮件、病毒库更新等）将不可用。
- FortiAnalyzer、FortiCloud 日志上传及 NTP 同步也会被同时关闭。
- 此操作建议仅用于离线或高安全隔离环境。

```bash
config system global
    set cloud-communication disable
end
```

自动修改项

执行以上 CLI 命令禁用云通信后，以下配置会被系统自动更改：

```bash
config system global
    set fds-statistics disable
end

config system central-management
    set type none
    set include-default-servers disable
end

config system fortiguard
    set antispam-force-off enable
    set outbreak-prevention-force-off enable
    set webfilter-force-off enable
end

config system email-server
    set server ''
end

config system ntp
    set ntpsync disable
end

config system autoupdate schedule
    set status disable
end

config system autoupdate tunneling
    set status disable
end

config log fortiguard setting
    set status disable
end
```

重新启用云通信

1. 若需恢复云通信，请首先重新启用全局选项：

```bash
config system global
    set cloud-communication enable
end
```

2. 根据需要逐项恢复相应功能。

   - 重新启用 FortiCloud 自动登录：

     ```bash
     config system fortiguard
         set auto-join-forticloud enable
     end
     ```

   - 重新启用邮件服务器：

     ```bash
     config system email-server
         set server "fortinet-notifications.com"
         set port 465
         set security smtps
     end
     ```

   - 重新启用 NTP 同步：

     ```bash
     config system ntp
         set ntpsync enable
     end
     ```

特征库误报/缺失上报

如果在使用 IPS、AV、Application 等特征库时，出现误报、缺少应用特征等问题时，您可自行联系 FortiGuard 团队进行相关信息的报告。

- 应用控制特征库缺失：https://www.fortiguard.com/faq/appctrlsubmit ↗
- 申诉应用控制特征库误报：https://www.fortiguard.com/faq/classificationdispute ↗
- 申诉被封的垃圾邮件 IP、URL 或电子邮件地址：https://www.fortiguard.com/faq/antispam/blocklist ↗
- 申诉网页过滤错误分类：https://www.fortiguard.com/faq/wfratingsubmit ↗
- 提交或申诉病毒文件：https://www.fortiguard.com/faq/antivirus-contact ↗
- 提交病毒文件并在线扫描：https://www.fortiguard.com/faq/onlinescanner ↗
- 报告 IPS 漏洞或 IPS 服务问题：https://www.fortiguard.com/faq/ips-contact ↗
- 报告终端系统漏洞问题：https://www.fortiguard.com/faq/ev-contact ↗
- 报告 ISDB 问题：https://www.fortiguard.com/faq/isdb-contact ↗
- 报告 Fortinet 产品漏洞：https://www.fortiguard.com/faq/psirt-contact ↗
- 报告 IOC（Indicators of Compromise）分类问题：https://www.fortiguard.com/faq/ioc ↗
- 报告 IP 地理分类问题：https://www.fortiguard.com/faq/ipge ↗
- 报告恶意 URL 分类问题：https://www.fortiguard.com/faq/malurl ↗
- 报告安全 URL 分类问题：https://www.fortiguard.com/faq/wlurl ↗
- 报告 IoT（Internet of Things）分类问题：https://www.fortiguard.com/faq/iotdispute ↗
- 报告 FortiWeb 应用程序分类问题：https://www.fortiguard.com/faq/contact-web-security ↗
- 报告 SSL 证书黑名单问题：https://www.fortiguard.com/faq/bad-cert ↗
- 报告应用控制功能问题：https://www.fortiguard.com/faq/appcontrol-contact ↗
- 报告网页过滤服务问题：https://www.fortiguard.com/faq/webfilter-contact ↗
- 报告反垃圾邮件服务问题：https://www.fortiguard.com/faq/antispam-contact ↗

服务指南

- FortiCare 服务介绍
- 设备注册步骤
- 设备强制注册
- 服务注册步骤
- PRMA 高级服务注册步骤
- 产品服务续费
- RMA 服务迁移
- RMA/DOA 硬件保修政策
- 产品注册信息变更
- 远程技术支持
- FortiEDR Cloud 注册指南

FortiCare 服务介绍

请参考 FortiCare 服务介绍 ↗。

设备注册步骤

1. 设备注册的两种方法

- 登录设备在管理界面注册（推荐）：
  - 不需要 FortiCloud Key。
  - 可避免因序列号录入错误导致误注册。
  - 即使暂时没有 FortiCloud Key，也可以顺利注册。
- 登录 https://support.fortinet.com/asset/#/dashboard ↗ 进行注册：
  - 注册时通常需要 FortiCloud Key（可在设备面板查看）。
  - 如果设备上架后无法获取 FortiCloud Key，建议改为登录设备管理界面注册。



2019 年及以后的 FortiAP 见此图

需要 FortiCloud Key 的型号说明

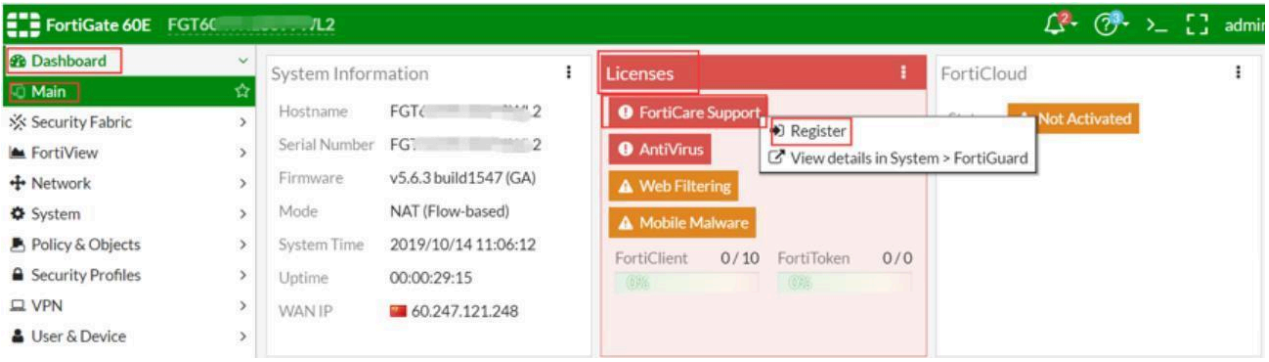
- FortiGate：
  - FGT-100 及以下型号需要 FortiCloud Key。
  - FGT-200 系列及以上型号注册时不需要 FortiCloud Key。
- FortiSwitch：全系列均需要 FortiCloud Key。
- FortiAP：全系列均需要 FortiCloud Key。

FortiCloud Key 位置说明

- 2018 年生产设备：FortiCloud Key 位于面板上方。
- 2019 年及以后生产设备：FortiCloud Key 位于设备背面，与产品序列号在一起。

2. 登录设备注册（推荐）

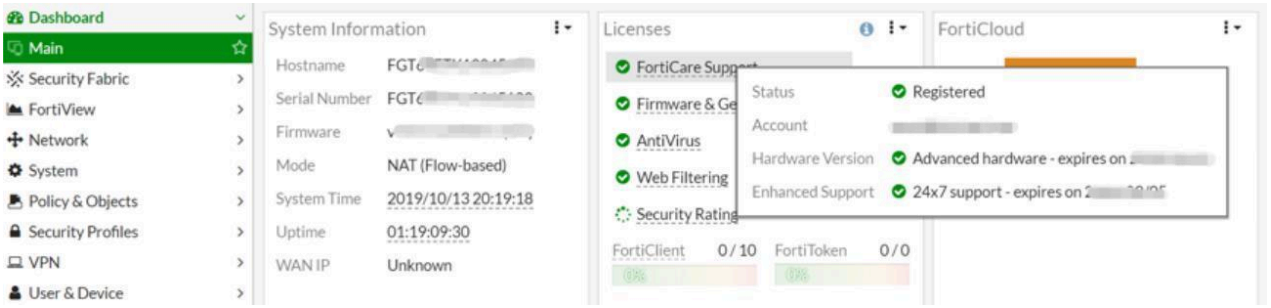
1. 登录设备，确保设备可以连接 Internet（该步骤必须）。
2. 进入 `Dashboard → Main → Licenses`，点击 `FortiCare Support` 下的 `Register`。



3. 在弹出的注册页面中输入账号、密码，选择国家和代理商，点击 `OK`。



4. 完成注册。

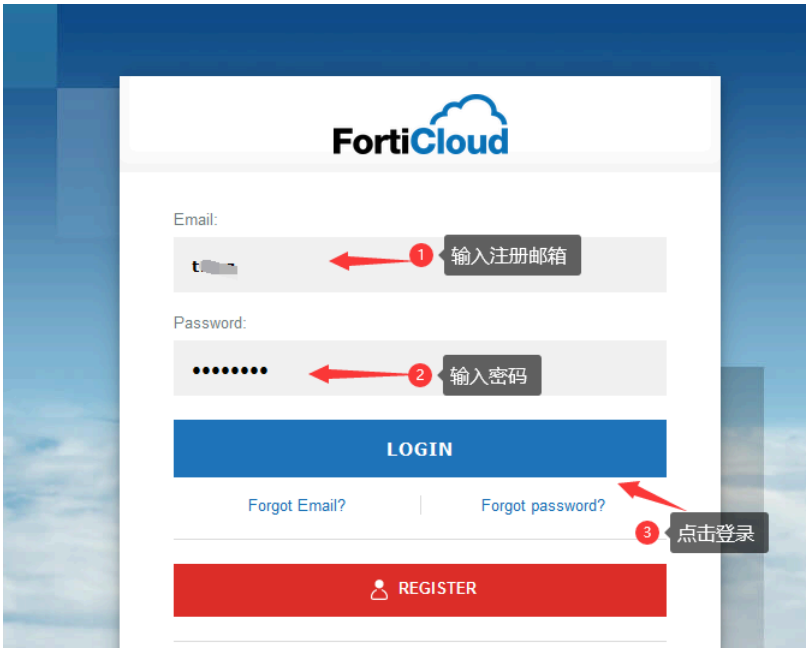


3. 登录 Support.fortinet.com ↗ 注册

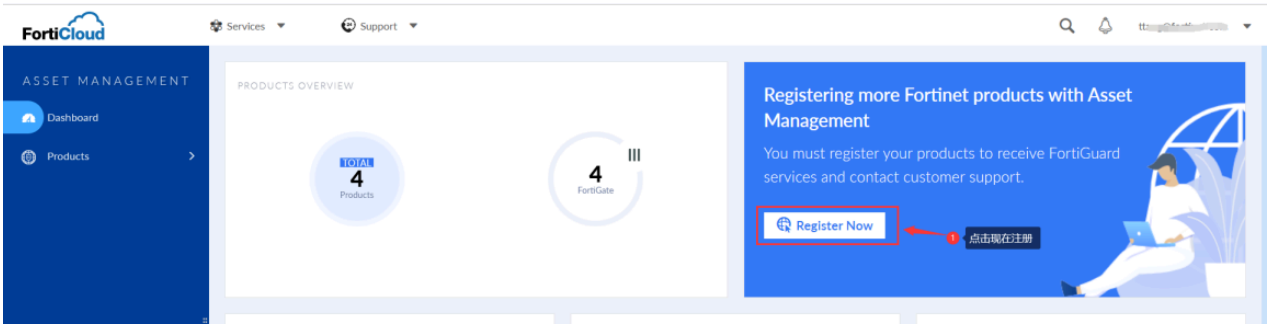
1. 访问 `https://support.fortinet.com`，点击 `Login Now`。



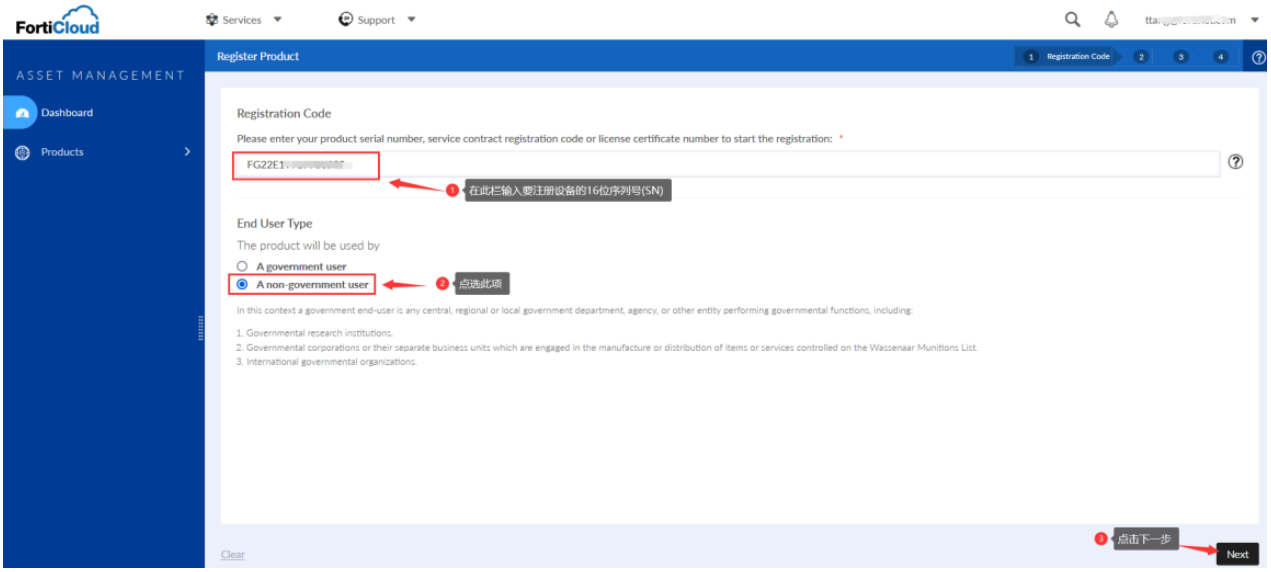
2. 输入注册邮箱和密码，点击 `Login`。



3. 进入账户后点击 `Register Now`，开始注册设备。

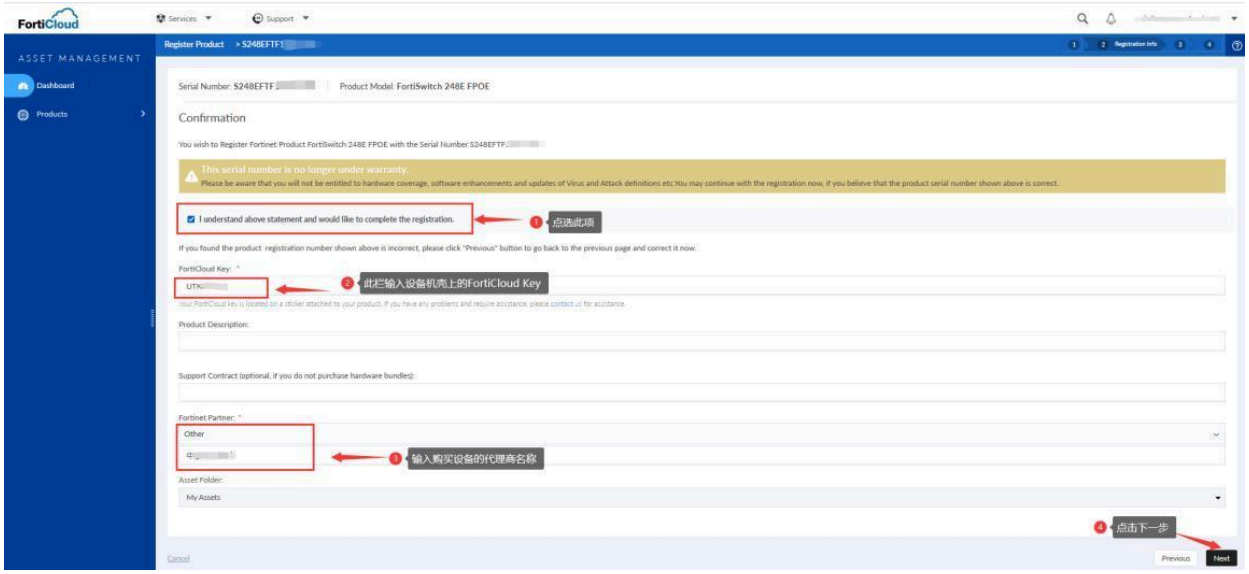


4. 输入要注册设备的 16 位序列号（SN），点击 `Next`。

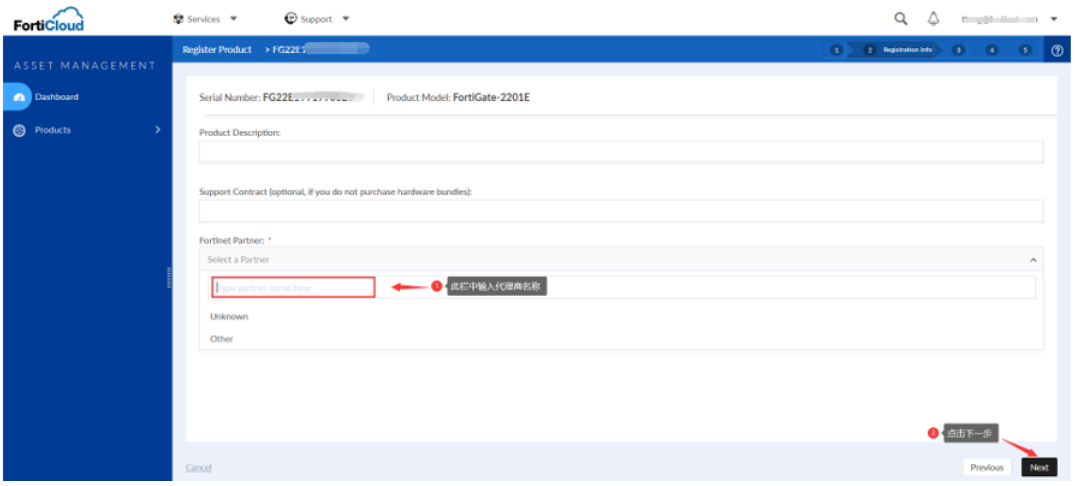


5. 选择购买产品的代理商。

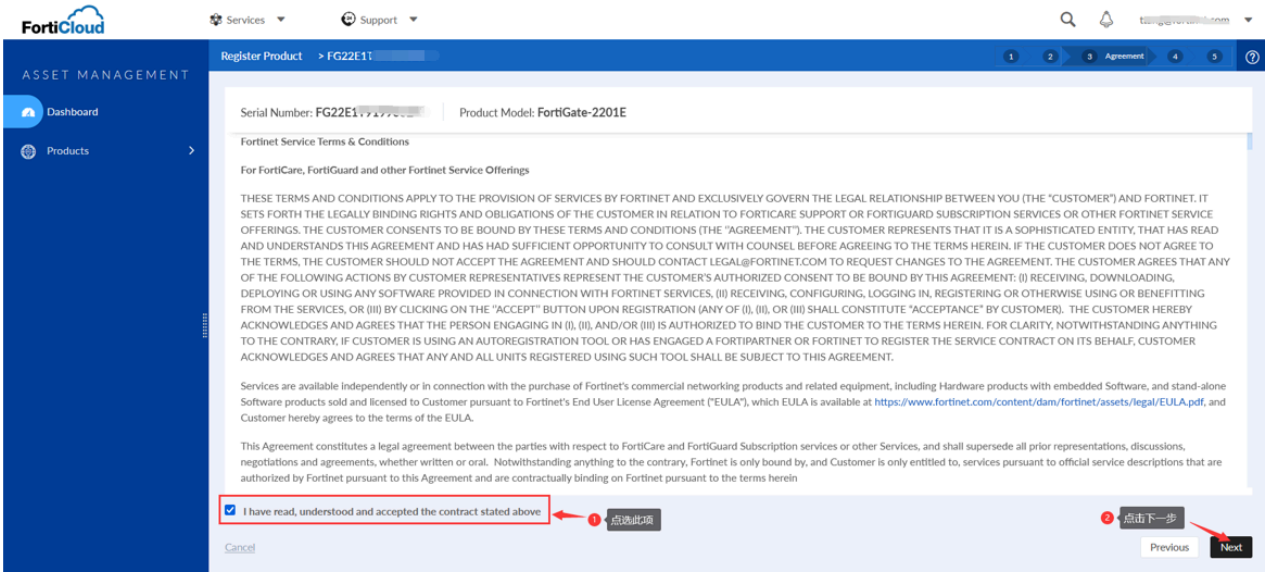
   - 需要 FortiCloud Key 的设备注册流程：



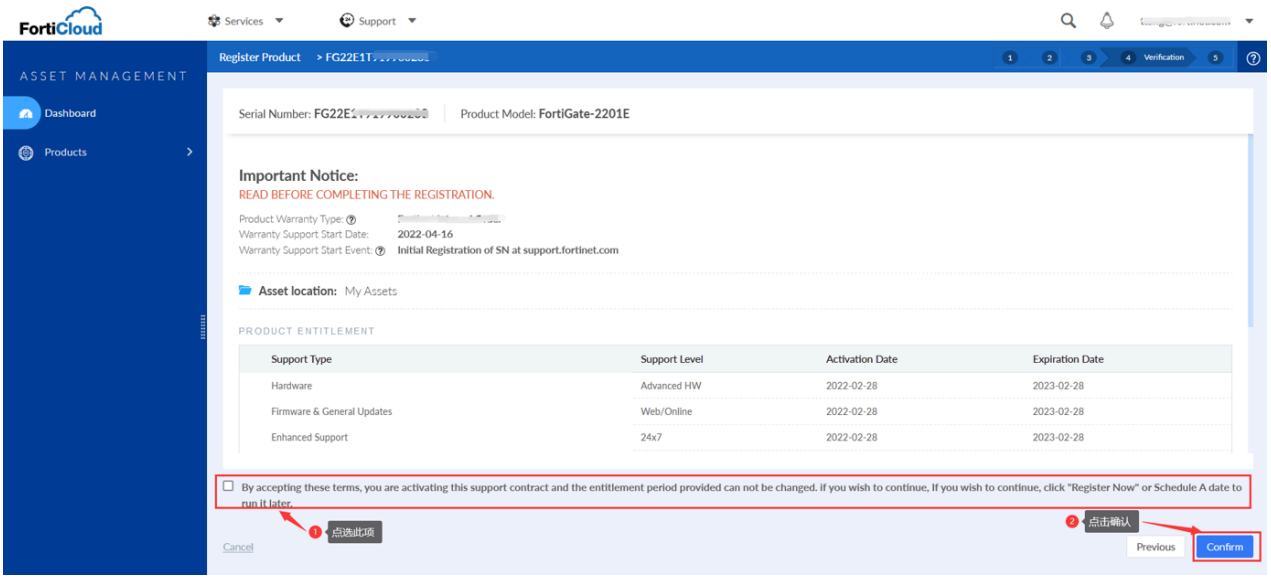
- 不需要 FortiCloud Key 的设备注册流程：



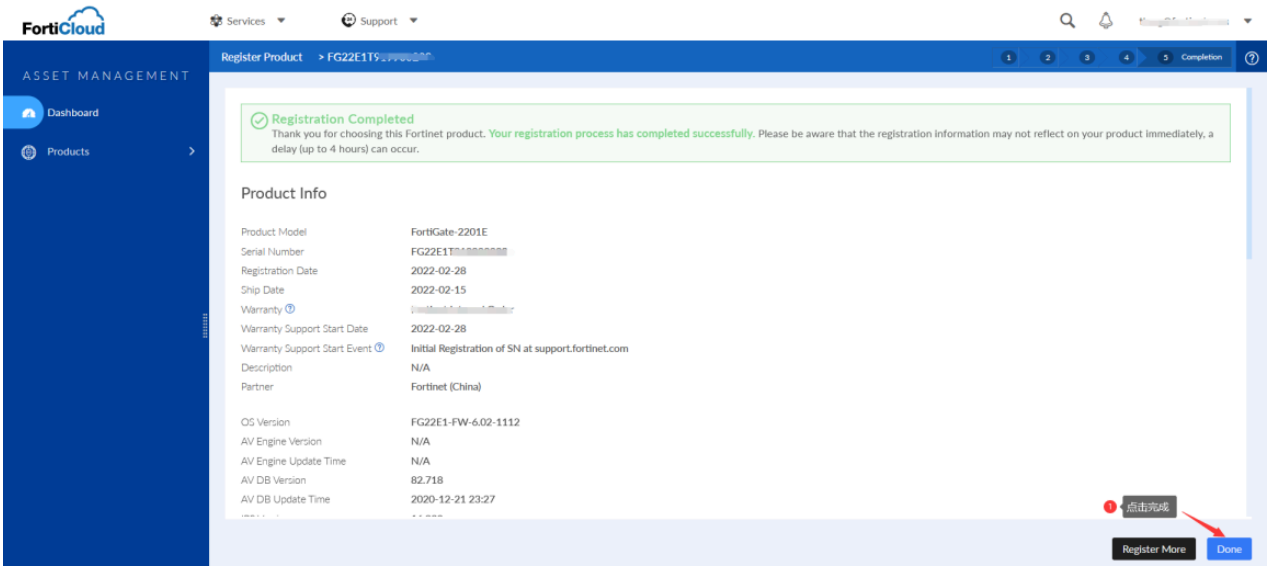
6. 确认 `Agreement`，点击 `Next`。



7. 确认设备信息，点击 `Confirm`。



8. 注册成功后，点击 `Done` 。



4. 查看产品列表

在产品列表中点击产品序列号，可查看该产品相关信息。



设备强制注册

功能简介

在 `FortiOS 7.2.11/7.4.8/7.6.5` 及后续版本的适用硬件机型上，FortiGate 启用“强制注册”机制。该机制要求设备先完成 FortiCare 注册，再开放完整配置与后续升级能力。

- `FortiOS 7.2.11/7.4.8`：引入强制注册机制（仅如下机型）。
  - `FG-200G`
  - `FG-201G`
  - `FG-900G`
  - `FG-901G`
- `FortiOS 7.4.9`：扩大机型范围。
  - FortiGate：`FG-50G/51G`、`FG-70G/71G`、`FG-90G/91G`、`FG-120G/121G`、`FG-200G/201G`、`FG-700G/701G`、`FG-900G/901G` 及对应 SFP/POE/DSL/5G 变体。
  - FortiWiFi：`FWF-50G/51G`、`FWF-70G/71G` 及对应 DSL/5G/SFP/POE 变体。
  - FortiGate Rugged：`FGR-50G-5G`、`FGR-70G`、`FGR-70G-5G-DUAL`。
- `FortiOS 7.4.10`：新增网络连通向导、离线许可证上传入口，并支持最多 7 天延后注册。
- `FortiOS 7.6.5`：在 7.6 分支启用该机制，并增加 CLI 只读限制与相关向导能力。

未注册行为

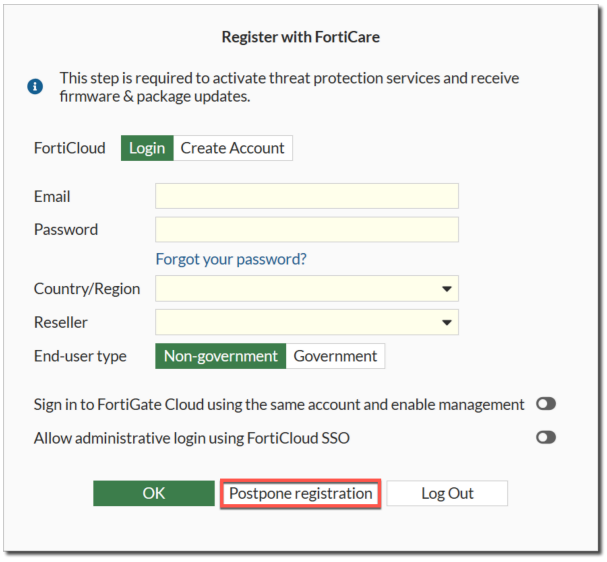
设备未完成注册时，会出现以下限制：

- 首次登录 GUI 会进入 `Register with FortiCare` 页面，无法继续后续 GUI 配置。
- CLI 处于受限状态，仅允许网络相关部分配置。
  - `config firewall`
  - `config ftp-proxy`
  - `config router`
  - `config system`
  - `config web-proxy`
- 固件升级能力也受注册状态限制。
- 设备需要可以连接到 FortiGuard 才能完成注册（或导入离线 License）并解除以上限制。

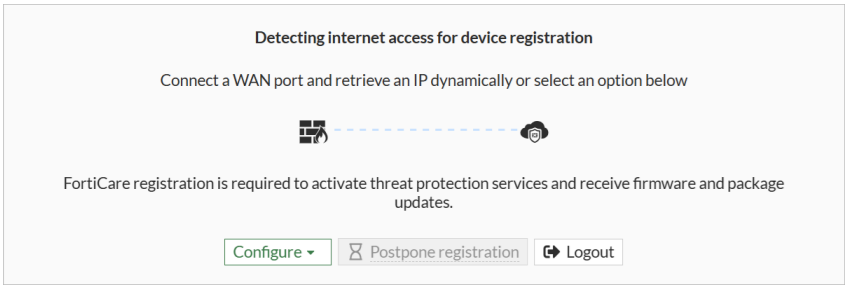
Workaround

延期注册（7.4.10 开始）

- 点击注册窗口内的“Postpone registration”按钮可以延期 7 天，期间可以正常通过 GUI 配置 FortiGate：



- 延期到期后会重新进入强制注册状态，无法继续延期：

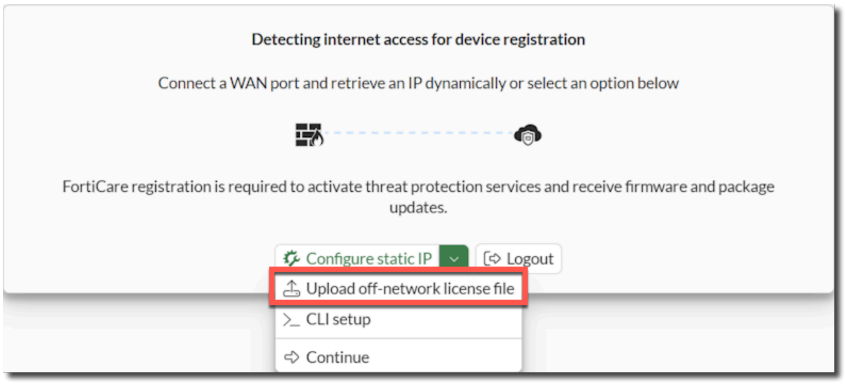


离线许可证方式

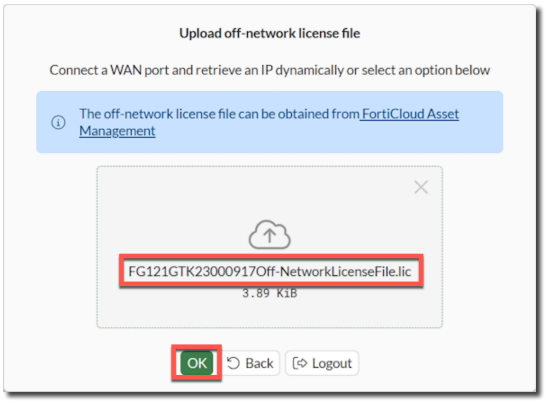
提示

- 适用于无法直连互联网、但需要满足强制注册机制的场景。
- 离线许可证文件中的序列号必须与设备序列号一致，系统才会判定注册有效。
- 离线 License 的下载方法可参考系统管理 → FortiGuard 管理 → 离线导入设备 License ↗ 章节。

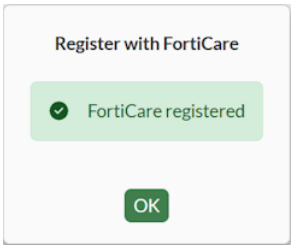
1. 设备登录后，在注册向导中选择离线 License 导入入口：



2. 进入上传页面后，导入离线 License 文件：



3. 导入成功且序列号匹配后，页面会提示离线 License 生效：



4. 也可使用 CLI 方式导入，CLI 导入前，需先保证 FortiGate 到 FTP/TFTP 服务器网络可达：

```bash
execute restore manual-license {ftp|tftp} <licensefilename> <server> [args]

execute restore manual-license tftp FG101FTKxxxxxxxxOff-NetworkLicenseFile.lic 192.168.1.10
```

BIOS 禁用强制注册

1. 使用 Console 线连接设备串口并重启设备。

2. 启动时看到以下提示，按任意键进入 BIOS 菜单：

```bash
Boot up, boot device capacity: 28626MB.
Press any key to display configuration menu...
```

3. 在 BIOS 主菜单输入 `I`，进入系统信息菜单：

```bash
[C]:  Configure TFTP parameters.
[R]:  Review TFTP parameters.
[T]:  Initiate TFTP firmware transfer.
[F]:  Format boot device.
[B]:  Boot with backup firmware and set as default.
[I]:  System configuration and information.
[Q]:  Quit menu and continue to boot.
[H]:  Display this list of options.
```

4. 在下一层菜单输入 `C`，进入 FortiCare 注册级别设置：

```bash
Enter C,R,T,F,B,I,Q,or H:
[S]:  Set serial port baudrate (will take effect on next boot).
[R]:  Set restricted mode.
[T]:  Set menu timeout.
[U]:  Set security level.
[C]:  Set FortiCare registration.
[I]:  Display system information.
[E]:  Reset system configuration.
[M]:  Display SPD information.
[Q]:  Quit menu and continue to boot.
[H]:  Display this list of options.
```

5. 输入 `1`，将注册级别从 `2`（Enforce）改为 `1`（Not Enforce）：

```bash
Enter S,R,T,U,C,I,E,M,Q,or H:
[1]:  Not Enforce
[2]:  Enforce
Enter FortiCare registration setting [2]: 1
```

6. 连续输入 `Q` 退出 BIOS 菜单并继续启动系统。启动后 GUI 可临时跳过强制注册继续配置。

服务注册步骤

1. 查看产品服务 License 文档。购买服务后，用户会收到 `Service Registration Document`，其中包含 `Contract Registration Code`（用于服务注册），同时文档内还有服务内容描述。

Service Entitlement Summary

Date: March 26, 2013

Purchase Order Number: TSING0322-SV

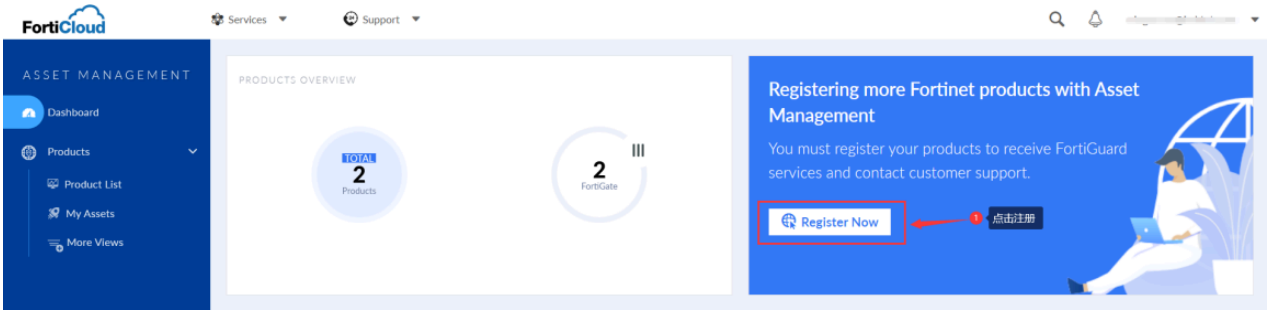
Contract Registration Code: 92237[redacted] 服务产品license不要泄露给第三方

Support / Maintenance / Subscription Services Included

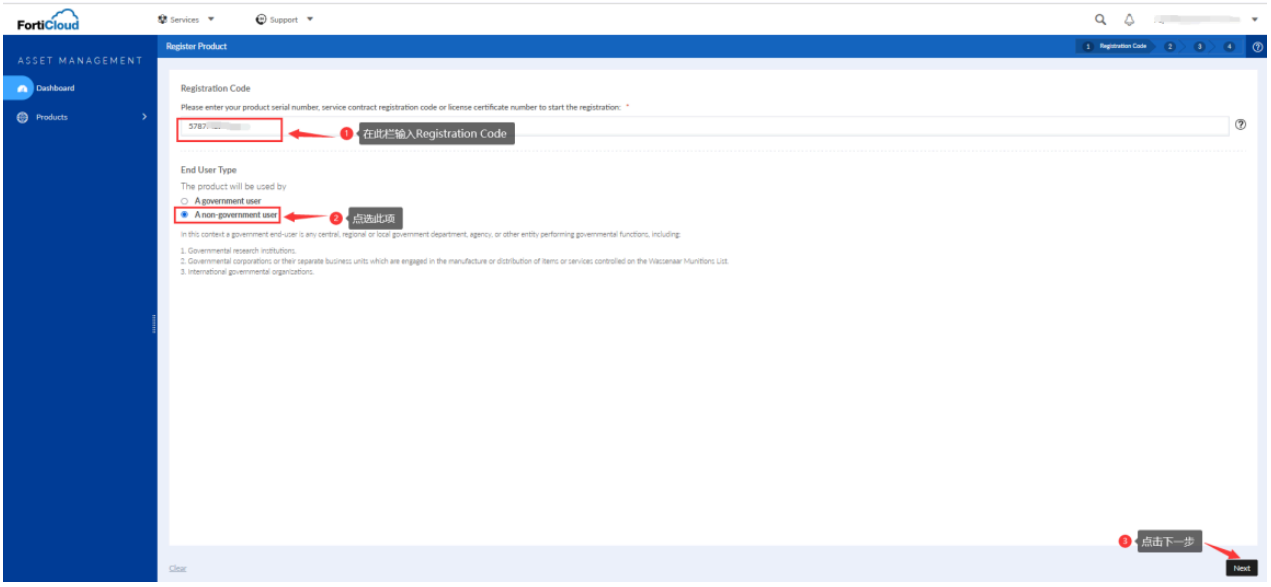
Qty	Part Number	Description
1	FC-10-00040-900-02-12	365 days coverage for FortiGate 40C include: Hardware Return To Factory Firmware & General Updates Enhanced Support 8x5 AntiVirus IPS Web Filtering AntiSpam

2. 登录 https://support.fortinet.com/asset/#/dashboard ↗。如该网站尚未注册账号，请先完成新产品注册。

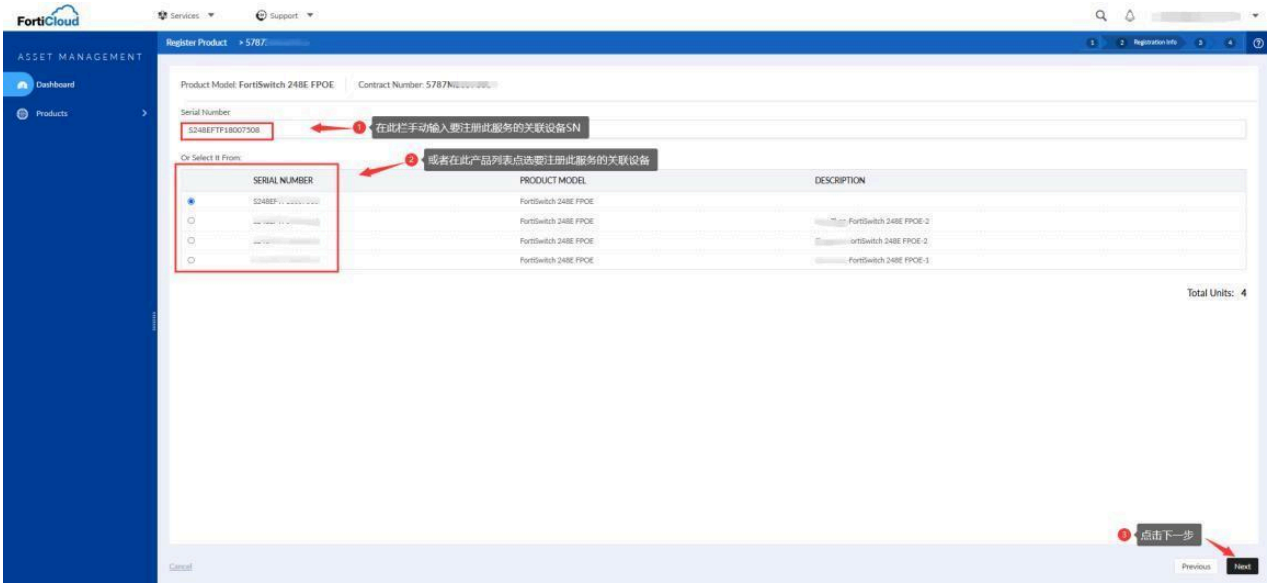
3. 在门户页面点击 `Register Now`。



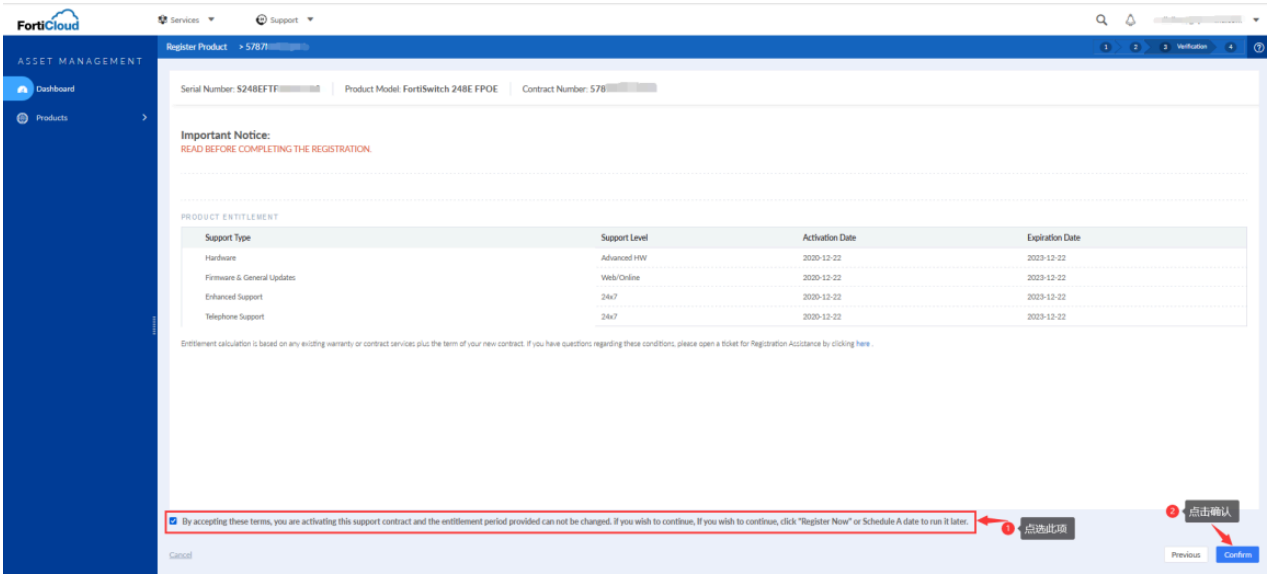
4. 填写 `Registration Code`，选择客户类型，然后点击 `Next`。



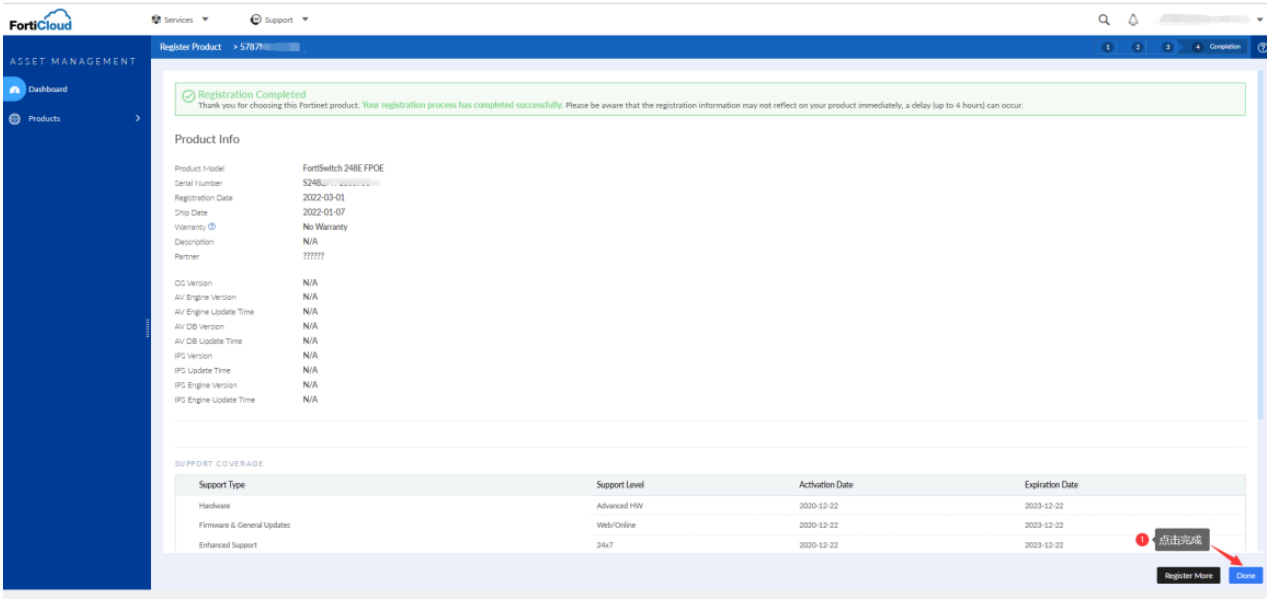
5. 手动输入产品序列号，或在列表中选择服务关联硬件产品，点击 `Next`。



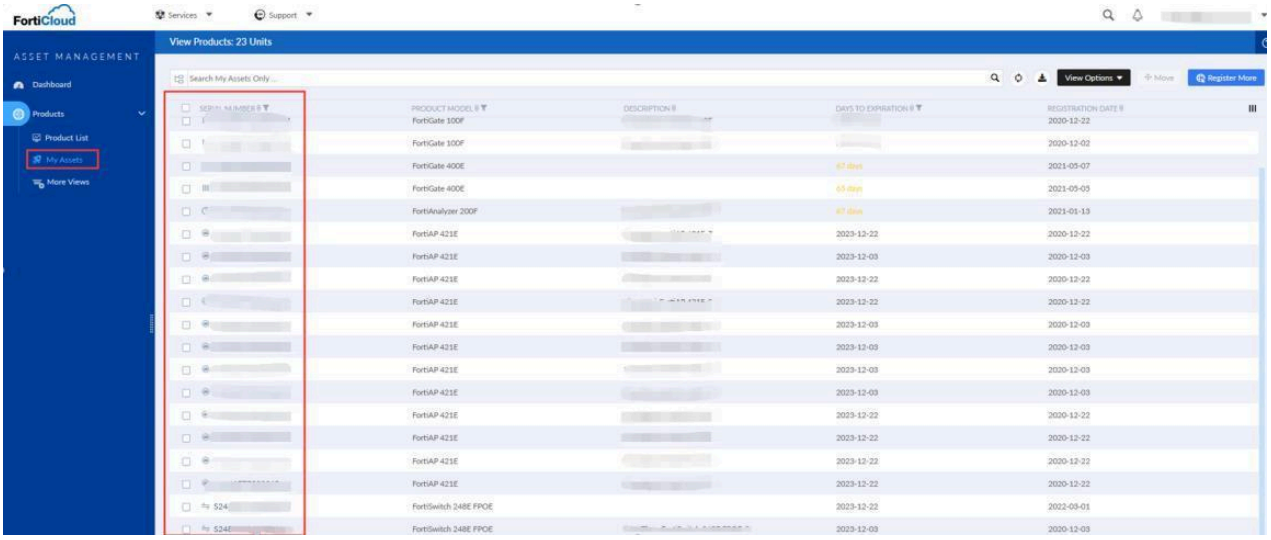
6. 阅读并确认产品注册协议，点击 `Confirm`。



7. 校验注册信息，点击 `Done`，完成服务注册。



8. 查看资产列表 `My Assets`，点击产品序列号，可查看相关服务信息。



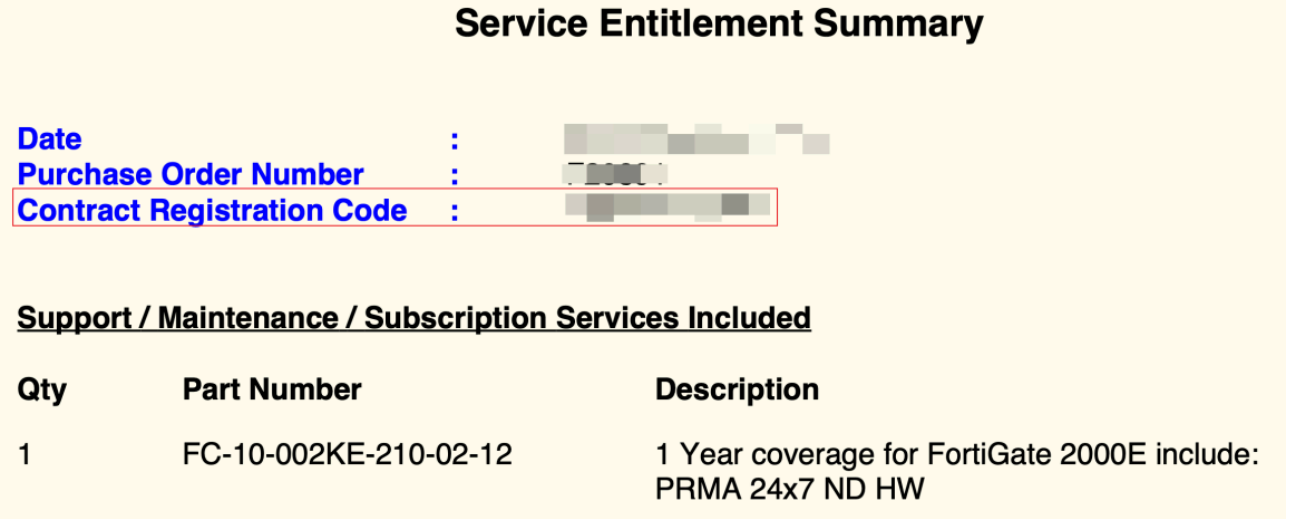
PRMA 高级服务注册步骤

目的

本文描述 PRMA 服务注册的逻辑和注意事项。

预先准备

- PRMA 合同注册码（Contract Registration Code）。
- 设备已完成基础注册（可参考“系统管理 → FortiGuard 管理 → 服务指南 → 设备注册步骤 ↗”）。



Service Entitlement Summary

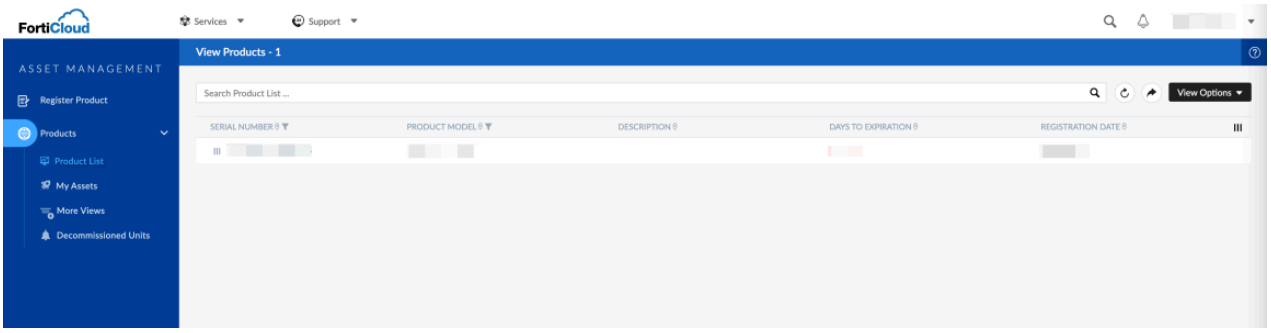
Date :
Purchase Order Number :
Contract Registration Code :

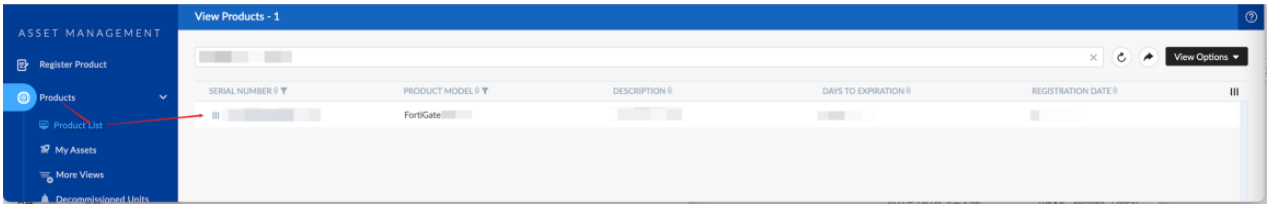
Support / Maintenance / Subscription Services Included

Qty	Part Number	Description
1	FC-10-002KE-210-02-12	1 Year coverage for FortiGate 2000E include: PRMA 24x7 ND HW

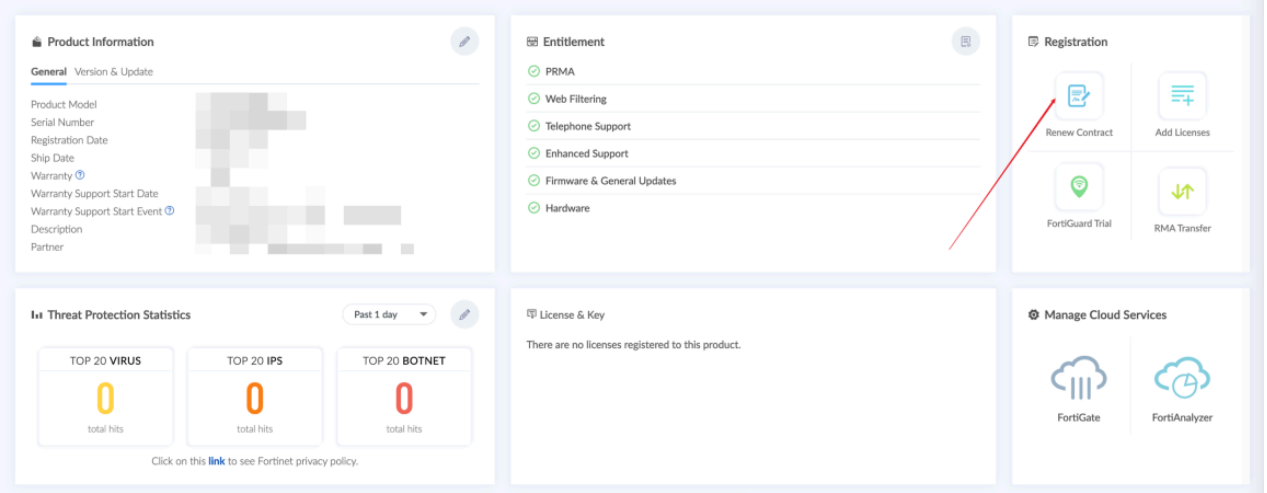
PRMA 注册流程

1. 打开 https://support.fortinet.com/asset/#/dashboard ↗ 并登录账号。
2. 进入 `Products → Product List`，搜索目标设备序列号，并点击对应设备。

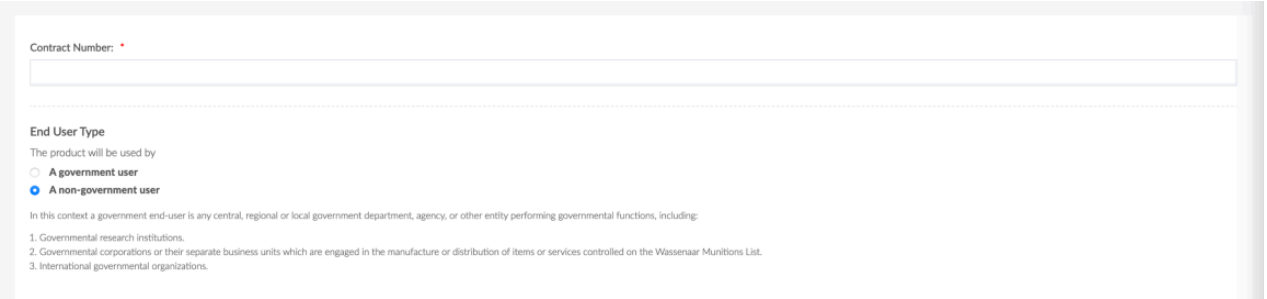




3. 在设备详情页点击 `Registration` 区域下的 `Renew Contract`。

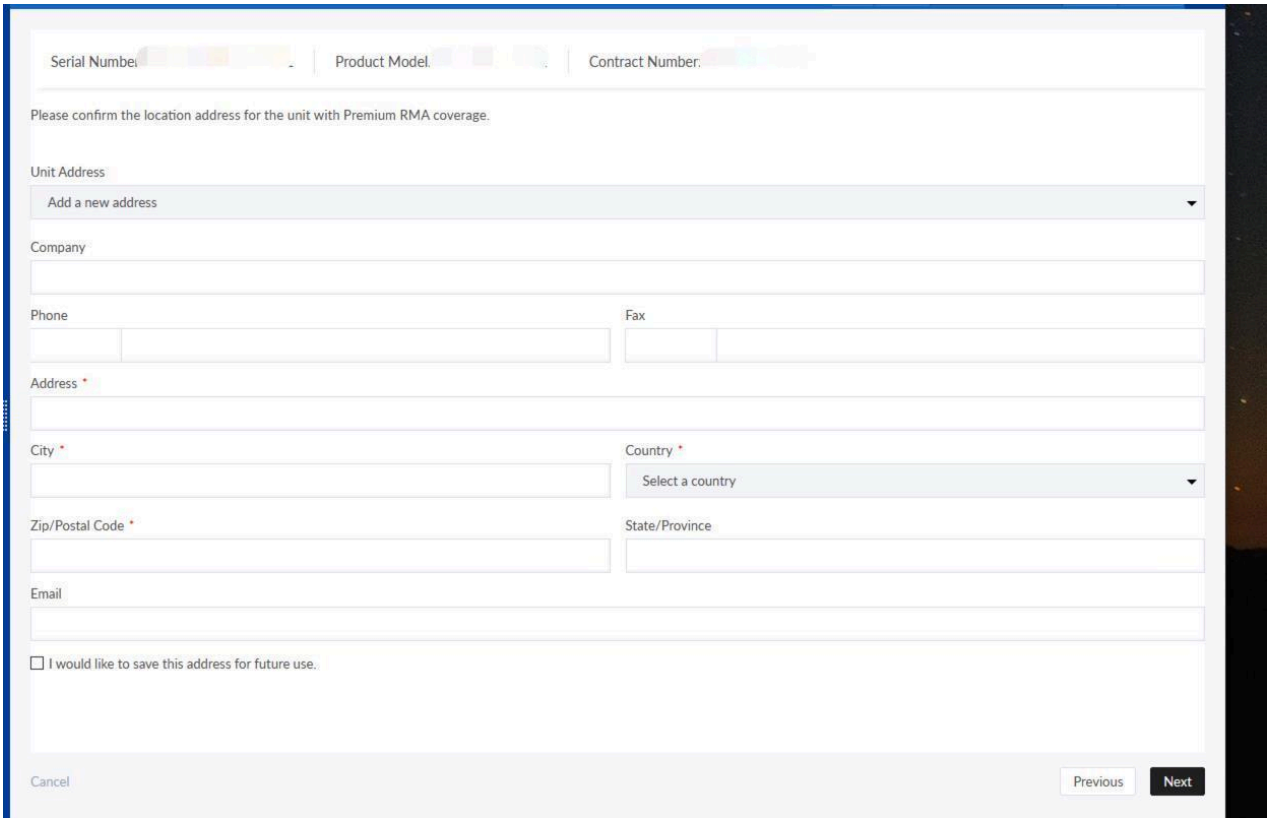


4. 在 `Contract Number` 中填写 PRMA 合同注册码，选择 `End User Type`，然后点击下一步。

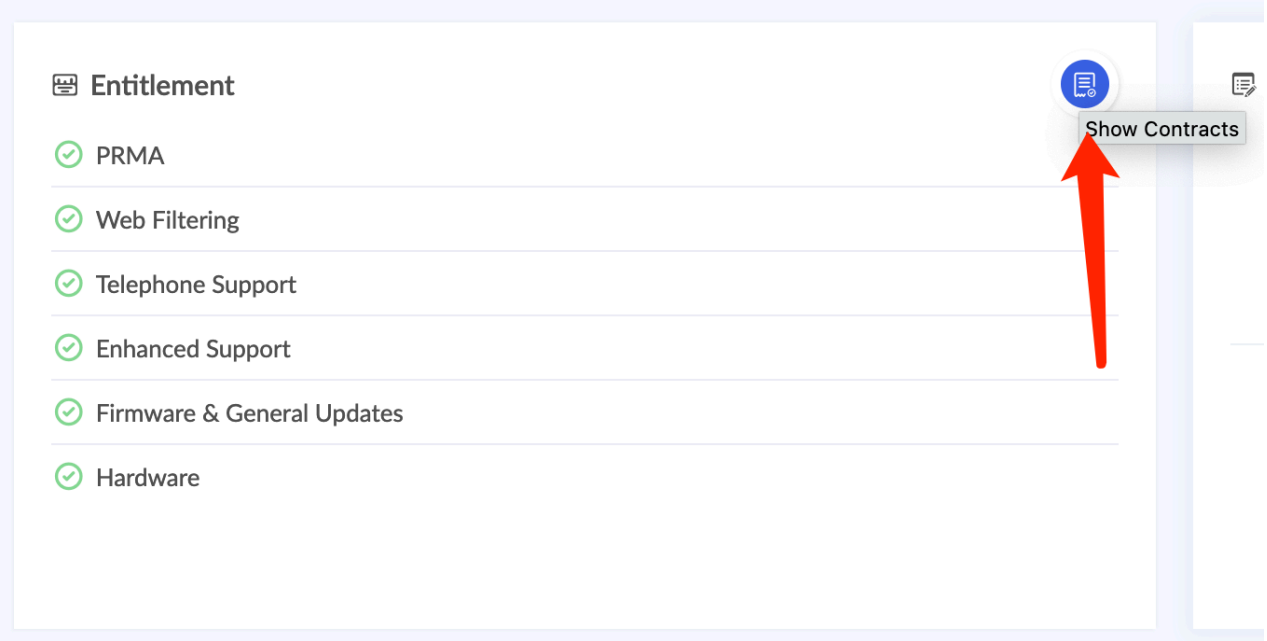


5. 填写设备实际安装地址信息并提交。

> **重要**
>
> PRMA 服务按设备实际安装地址生效，地址错误可能导致备件无法按预期送达。

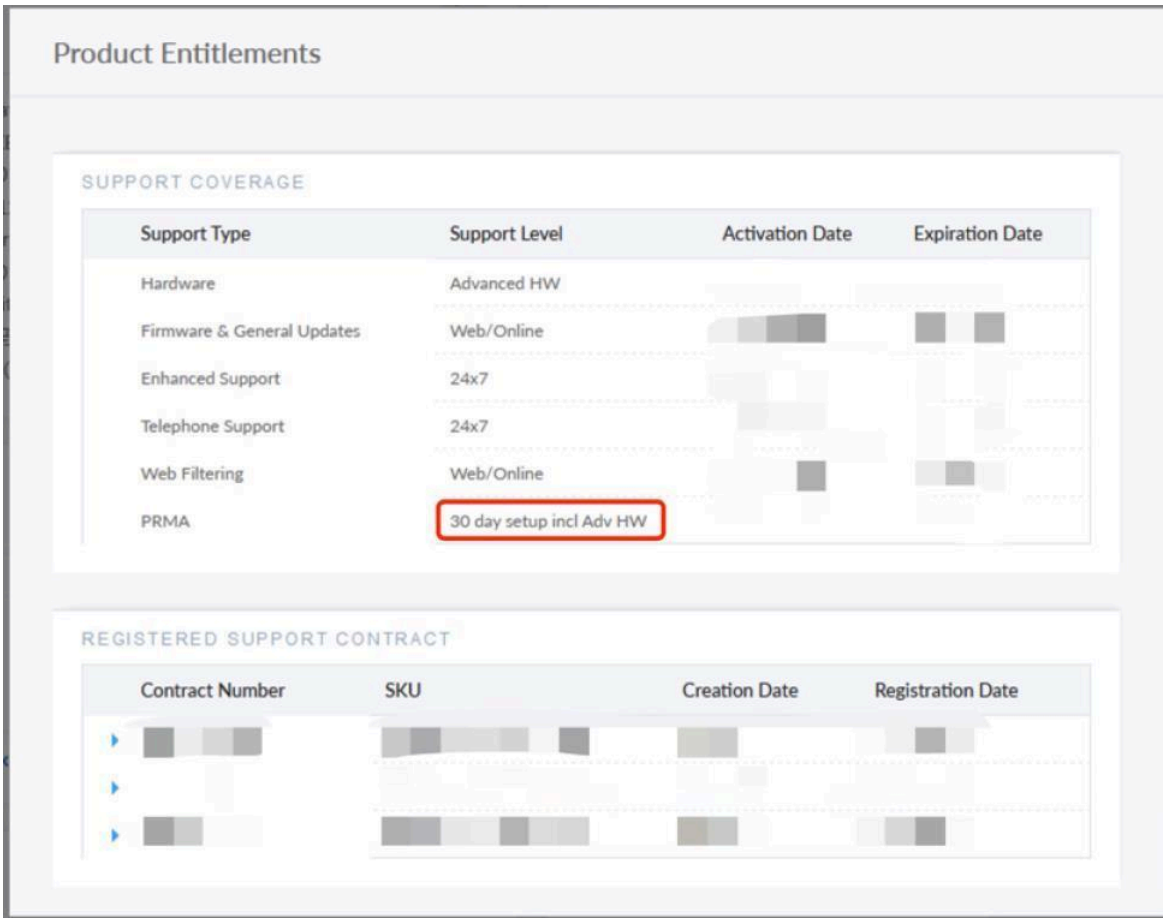


6. 返回设备详情页，在 `Entitlement` 栏右上角点击 `Show Contracts` 查看 PRMA 状态。

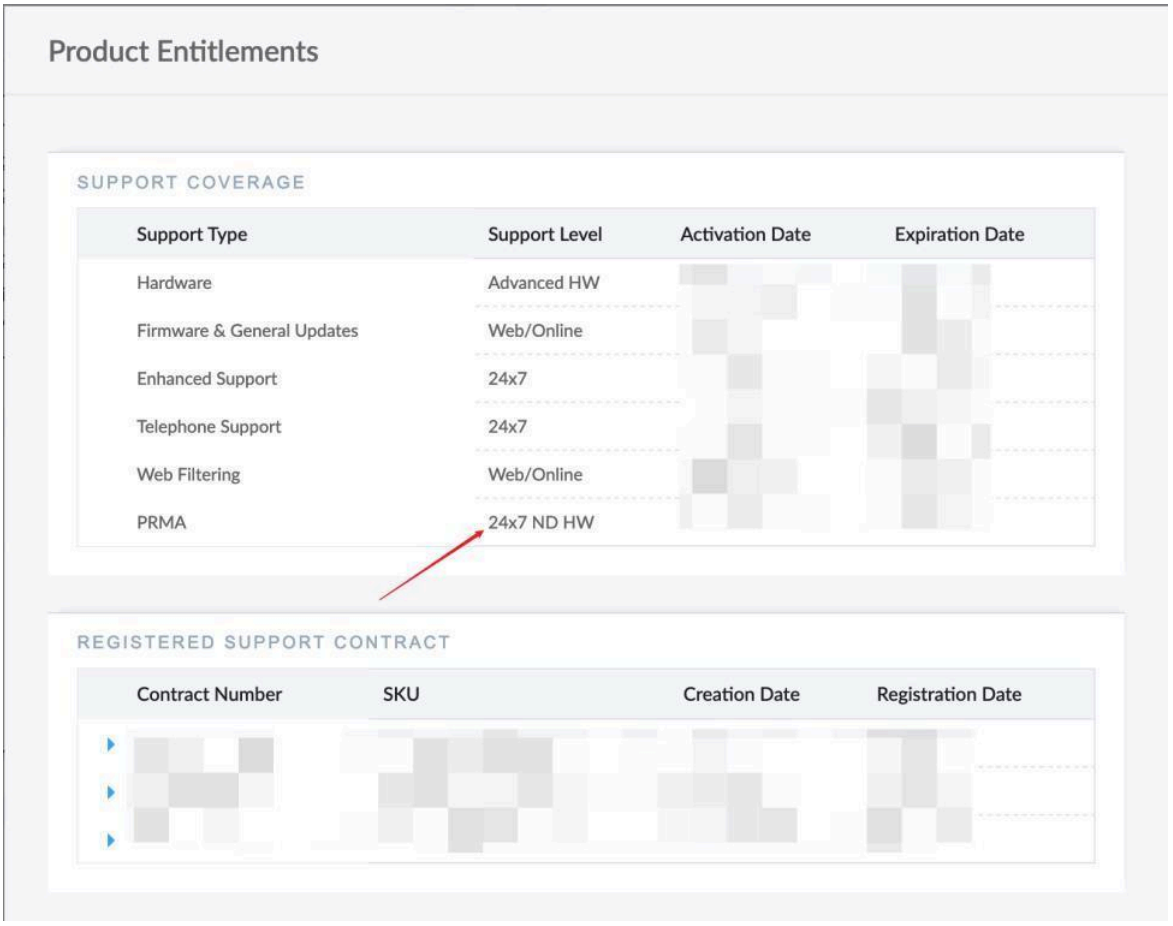


状态说明

- 刚注册完成后，PRMA 状态通常显示为 `30 day setup incl Adv HW`（备货周期内）。



- 一般在 5 个工作日内，状态会更新为 `24x7 ND HW`，表示流程完成。


Product Entitlements
SUPPORT COVERAGE
Support Type
Support Level
Activation Date
Expiration Date
Hardware
Advanced HW
Firmware & General Updates
Web/Online
Enhanced Support
24x7
Telephone Support
24x7
Web Filtering
Web/Online
PRMA
24x7 ND HW
REGISTERED SUPPORT CONTRACT
Contract Number
SKU
Creation Date
Registration Date

产品服务续费

Fortinet 是从何时实施服务连续政策的?

Fortinet 自 2010 年 5 月起实施服务连续政策。

什么是 Fortinet 服务连续政策?

Fortinet 的 FortiCare 和 FortiGuard 服务是连续服务。若服务到期前未完成续费，过期后续费将按原服务到期日计算新服务有效期，以保持服务连续性，这种方式通常称为“回溯”。

- 续费服务最大回溯期为 6 个月。
- 若服务已过期超过 6 个月，则一年续费中最多有 6 个月用于补齐失效期，剩余时间用于新的持续服务。

举例：过期 6 个月以内续费

若原服务到期时间为 2020 年 5 月 28 日，并于 2020 年 7 月 29 日续费一年，则续费后有效期为 2020 年 5 月 29 日至 2021 年 5 月 28 日。

举例：过期 6 个月以上续费

若原服务过期超过 6 个月，则续费开始时间从“续费生效日”向前回溯 6 个月起算。

例如：原服务到期为 2018 年 8 月 31 日，于 2020 年 5 月 30 日续费一年，则续费有效期为 2019 年 11 月 30 日至 2020 年 11 月 30 日。

哪些情况可豁免过期续费回溯?

1. 购买单项 FortiGuard 服务。
2. 购买两年及以上服务，且不超过 EOS 时间（Coterm 下单）。
3. 购买三年或五年服务包（多个一年服务包累加不属于此类）。

如果设备多个服务项过期日期不同，且都已过期怎么办?

将选用所有服务项中“最晚过期日”作为续费起算基准，并遵循回溯原则。

例如：多项服务均已过期，最晚过期项为 2020 年 4 月 1 日，并于 2020 年 7 月 3 日续费一年，则续费有效期为 2020 年 4 月 2 日至 2021 年 4 月 1 日。

注意：试用服务到期日不作为续费服务起算日期的计算标准。

如果设备多个服务项过期日期不同，且至少有一项未过期怎么办?

续费一年时：

- 未过期项：从原到期日累加一年。
- 已过期项：从续费生效日起算一年。

例如：设备多项服务到期时间不同，最晚过期项为 2020 年 6 月 1 日，于 2020 年 4 月 3 日续费一年，则：

- 未到期项有效期：2020 年 6 月 2 日至 2021 年 6 月 1 日。
- 已过期项有效期：2020 年 4 月 3 日至 2021 年 4 月 2 日。

此种情况下还有两项特例：

1. 试用服务到期时间不作为续费服务起算标准。
2. 若未到期项为设备出厂自带 Warranty 服务，购买并注册续费后，该未到期 Warranty 内容会被覆盖，而不是在原到期日上继续累加。

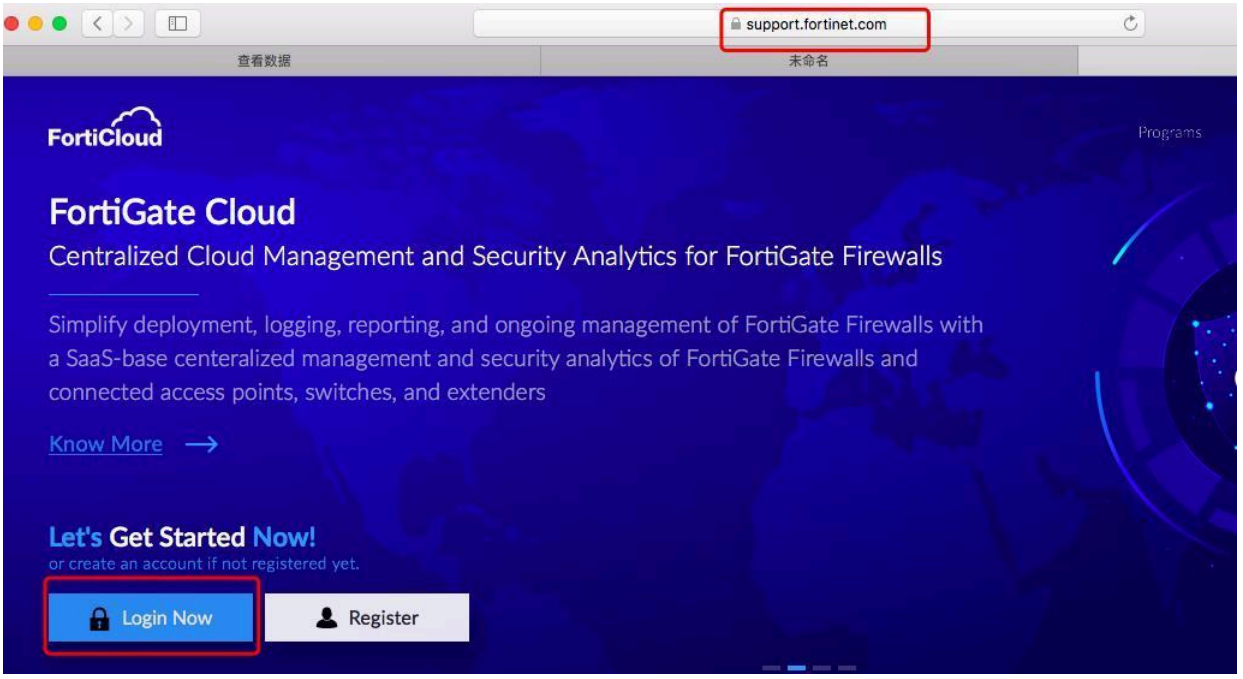
例如：设备注册后的出厂自带 Warranty 中，最晚一项到期时间为 2020 年 11 月 10 日，于 2020 年 6 月 10 日续费一年，则续费有效期为 2020 年 6 月 10 日至 2021 年 6 月 9 日。

RMA 服务迁移

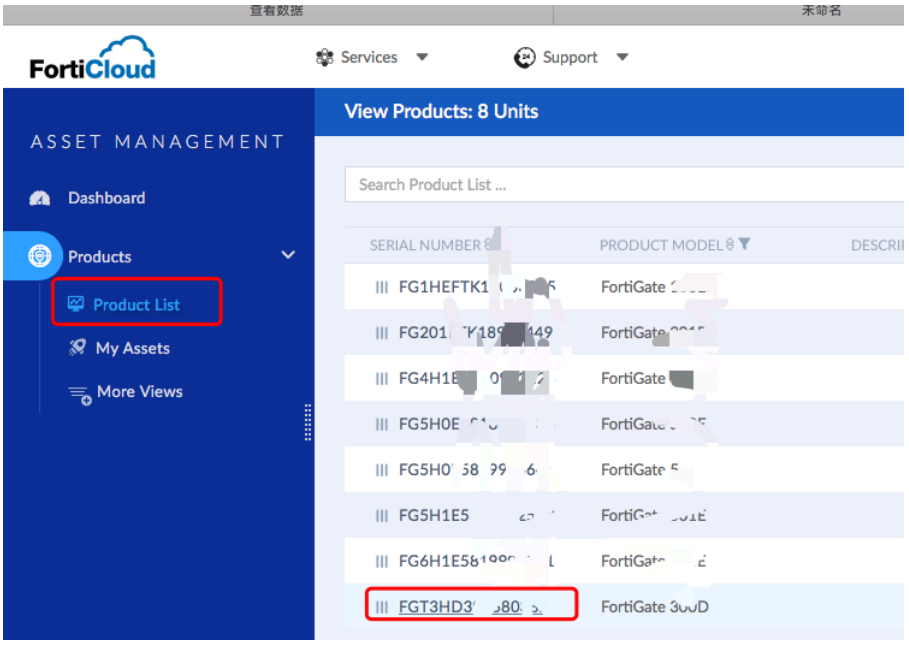
原设备服务未过期时，参与对等型号以旧换新后，应先将原设备服务迁移到新设备上，再注册新购服务。这样可以实现新设备服务与原设备剩余服务累加。

具体步骤如下：

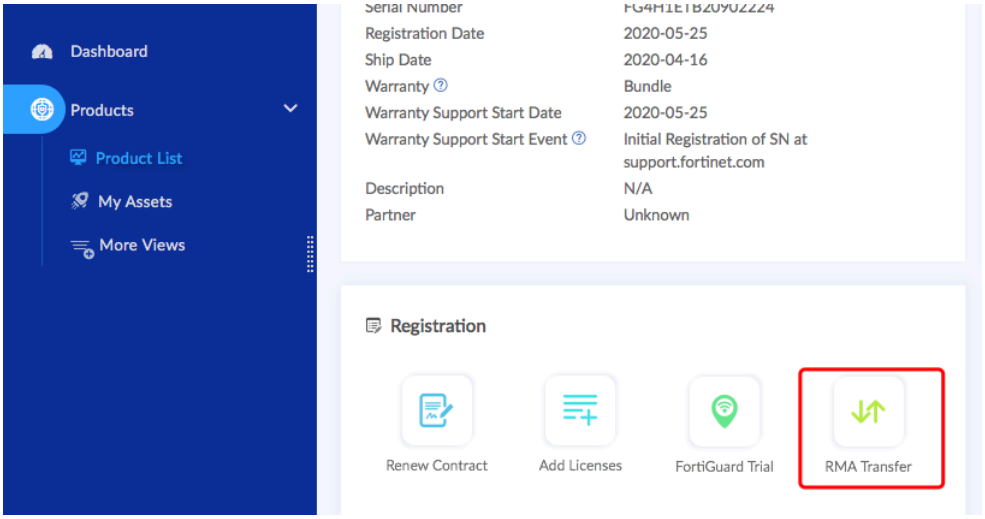
1. 最终用户收到新设备后，登录官方技术支持网站 https://support.fortinet.com/asset/#/dashboard ↗ 。



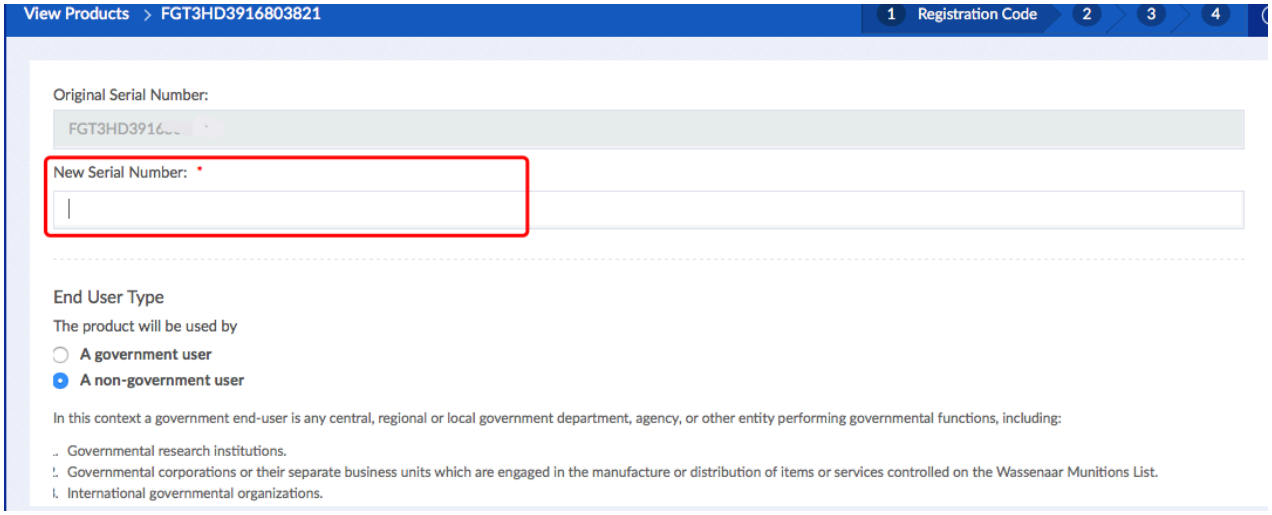
2. 在账号的产品列表中找到原设备 SN。



3. 点击该 SN 进入设备详细信息页面，在页面中部找到 `RMA Transfer` 按钮。



4. 点击 `RMA Transfer` 进入设备替换页面，填写新设备 SN，并完成相关信息后点击下一步，完成设备替换及服务迁移。



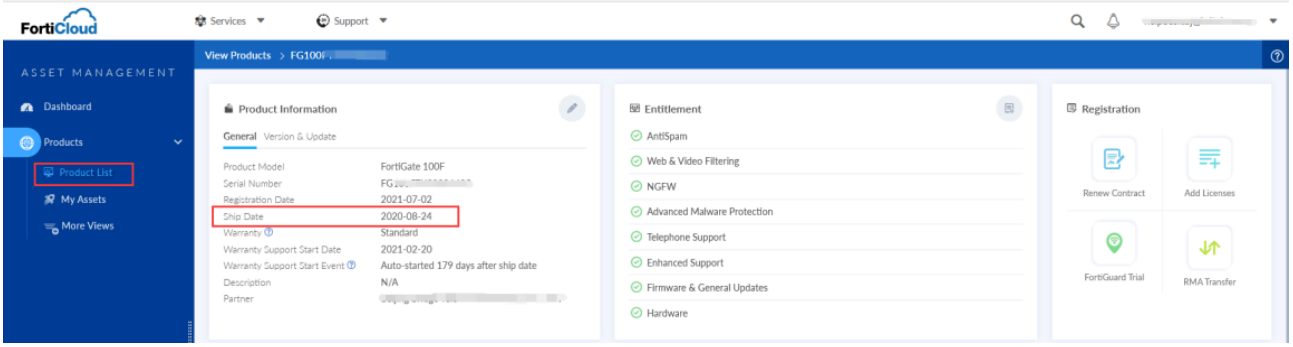
RMA/DOA 硬件保修政策

1. RMA/DOA 保修规定

RMA（Return of Materials Authorization）与 DOA（Dead On Arrival）是两种不同的硬件保修流程。

项目	RMA	DOA
时间界定	当前时间减去 `ship date` 大于 120 天，且硬件质保未过期	当前时间减去 `ship date` 小于 120 天，且硬件质保未过期
返修费用	客户承担单程运输费用	Fortinet 承担双程费用
返修周期	往返 RMA 中心双程时间	RMA 中心到代理商/客户单程时间

`ship date` 查询方法：登录 `support.fortinet.com`，在左侧菜单进入 `Product List`，查询设备 SN。



附加说明

1. RMA 返回设备不保证为原机，且新旧程度不保证；若返修设备已超过生命周期，Fortinet 可提供同等或更高配置替换。DOA 返回设备为全新设备。
2. 故障设备寄回要求：
   - RMA：通常仅寄故障设备本体，电源线、模块、挂耳等附件无需寄回。
   - DOA：需将故障设备及随机附件一并寄回。
   - 两种情况均需使用 Fortinet 原厂包装，避免运输损坏。
3. 禁止撕毁易碎贴（防篡改密封贴）。若撕毁，硬件服务将失效，Fortinet 不承担相关损坏责任。

备注：若客户无法联系代理商，可直接联系 Fortinet 中国售后支持中心进行返修，流程原则一致。

2. RMA Return & Replace 返修流程

1. 最终用户发现硬件故障并上报代理商。
2. 授权代理商在网上开 ticket 提交至 Fortinet。
3. Fortinet 工程师确认硬件故障后，转交 RMA 客服执行并跟踪流程。
4. RMA 客服发送 `shipping information` 表格，由代理商或客户填写；同时代理商或客户将故障设备寄至 Fortinet RMA 中心。
5. RMA 中心收到故障设备后 3 个工作日内，寄回修复原机或替代设备给代理商/客户。
6. 代理商确认后将设备交付最终用户。

流程简图（文字版）：

用户发现故障 -> 代理商确认 -> Fortinet TAC 确认 -> RMA 客服跟踪 -> 代理商/客户寄故障机到 RMA 中心 -> RMA 中心返回替代设备 -> 代理商交付最终用户

3. RMA Advanced Replace 返修流程

1. 最终用户发现硬件故障并上报代理商。
2. 授权代理商在网上开 ticket 提交至 Fortinet。
3. Fortinet 工程师确认硬件故障后，转交 RMA 客服执行并跟踪流程。
4. RMA 客服发送 `shipping information` 表格，由代理商或客户填写确认。
5. RMA 客服在收到反馈后的第二个工作日发出替换设备。

6. 代理商确认后将设备交付最终用户。
7. 最终用户需在收到替换设备后 30 个自然日内，将故障机寄回 Fortinet RMA 中心。

注：若逾期未返还故障机，Fortinet 有权按替换机全价收费。

流程简图（文字版）：
用户发现故障 -> 代理商确认 -> Fortinet TAC 确认 -> RMA 客服跟踪 -> RMA 中心先发替换设备 -> 代理商交付最终用户 -> 代理商/客户回寄故障机

4. DOA 返修流程

1. 最终用户在 DOA 时限内发现硬件故障并上报代理商。
2. 授权代理商在网上开 ticket 提交至 Fortinet。
3. Fortinet 工程师确认故障后，转交 RMA 客服执行并跟踪流程。
4. RMA 客服发送 `shipping information` 表格，由代理商填写确认。
5. RMA 中心在收到确认信息后的第二个工作日，将替换设备发给代理商。
6. 代理商确认后将设备交付最终用户。
7. 在收到替代设备后 30 个自然日内，代理商或客户将故障设备返还 Fortinet RMA 中心。

注：若逾期未返还故障机，Fortinet 有权按替换机全价收费。

流程简图（文字版）：
用户发现故障 -> 代理商确认 -> Fortinet TAC 确认 -> RMA 客服跟踪 -> RMA 中心返回替代设备给代理商 -> 代理商交付最终用户 -> 代理商回寄故障机到 RMA 中心

产品注册信息变更

1. 参考系统管理 → FortiGuard 管理 → 服务指南 → FortiCare 服务介绍 ↗中的第 31~43 页，提交 Ticket 处理（响应较快）。

2. 发送邮件至 support_cn@fortinet.com ↗ 提交变更请求。

3. 联系 400-600-5255 提交变更请求。

远程技术支持

由于当今网络的复杂性，很多时候需要在客户和 Fortinet TAC 工程师之间安排远程会话，以便有效地解决技术问题。

在向 Fortinet TAC 请求远程会话之前，请您确保遵守以下要求：

- 提供问题相关的日志文件、设备运行状态输出（如 `exe tac report`）、FortiGate 配置文件、网络拓扑、问题的详细描述以及问题的出现条件。
- 尝试在已有的 FortiGate 文档的帮助下独立解决该问题，并向 TAC 工程师详细描述解决该问题已经做过的尝试。
- 已考虑 TAC 工程师提出的建议，其中包括 Workaround 或版本升级，以缓解当前问题。
- **Fortinet TAC 支持的重点是故障排查，而不是配置协助。如果需要配置协助，请您先参考已有的 FortiGate 配置文档或联系您的代理商进行配置。**

如果确认满足以上要求，请遵循以下步骤，以实现高效的远程会话：

- 在加入远程之前，准备好排查环境。包括打开问题防火墙页面、终端工具等，以减少客户和工程师的时间浪费。
- 准备在 TAC 工程师远程期间明确解释并重现问题。
- 除非必要，请勿与其他同事共享会议链接，以避免给各方带来不必要的干扰。

创建 Ticket 的步骤请参考系统管理 → FortiGuard 管理 → 服务指南 → FortiCare 服务介绍 ↗中的第 31~43 页，提交 Ticket 处理（响应较快）。

FortiManager 管理

FortiManger 功能简介

FortiManger 产品介绍

FortiManger 版本推荐

FortiManger 网管登陆

FortiManger ADOM

- ADOM 简介
- 开启 ADOM
- ADOM 的高级特性

FortiManger 管理员

- 管理员账号管理
- 配置远程管理员
- 管理员和 ADOM 关联
- ADOM 的 Workspace 模式

FortiManger 功能简介

FortiManger 是 FortiGate 的统一网管系统

- 统一的集中管理防火墙的平台
- 最大限度的减少大型部署的初始成本和后续的运维成本
- 可以统一下发 CLI 脚本方便的保持所有 FGT 设备的合规性
- 可以作为本地的 FortiGuard 缓存服务器，FGT 无需去互联网更新 FortiGuard
- 自动化备份设备配置和策略模块维护
- SD-WAN 集中管理
- VPN 模块集中管理
- 日志收集和报告

什么时候应该在你的网络中使用 FortiManager?

在大型企业和托管安全服务提供商（MSSP）中，网络规模带来了小型网络所没有的挑战，通过 FortiManager 集中管理可以帮助客户更轻松地管理多种设备的不同部署类型，并降低操作成本：

- 批量部署
- 计划配置更改及下发
- 对许多变化进行维护
- 跟踪和审计

FortiManager 的主要功能是什么?

- 集中管理：您可以使用 FortiManager 从单个控制台管理所有 FortiGate，而不是单独登录数百个 FortiGate。
- 管理域（ADOM）：FortiManager 可以将设备分为地理或功能 ADOM，如果您拥有大型网络安全管理员团队，则更加适应 ADOM。
- 配置修订控制：您的 FortiManager 保存所有配置更改的历史记录。您可以安排 FortiManager 部署新的配置或将受管设备恢复到之前的配置。
- 本地 FortiGuard 服务供应：为了减少网络延迟并尽量减少互联网带宽使用，您的受管设备可以使用 FortiManager 作为专用的 FortiGuard 分布式网络（FDN）服务器。
- 固件管理：FortiManager 可以为受管设备安排固件升级。
- 脚本：FortiManager 支持基于 CLI 和基于 TCL 语言的脚本。
- 记录和报告：托管设备可以在 FortiManager 上存储日志。从该日志数据中，您可以生成基于 SQL 的报告，因为 FortiManager 具有与 FortiAnalyzer 相同的日志记录和报告功能。

FortiManager 能做什么?

- 在您的网络中配置防火墙策略
- 充当配置版本控制和安全审计的中央资料库（日志存档）
- 部署和管理复杂的网状和星型 IPsec VPN
- 充当受管设备和 FortiClient 安装的专用 FortiGuard 分发服务器（FDS）
- 使用 JSON API 脚本和自动化设备配置，策略更改等
- SD-WAN 集中管理中心

FortiManger 产品介绍

主要产品

硬件盒式设备
FMG 1000D
FMG 1000F
FMG 2000E
FMG 200F
FMG 200D
FMG 3000F
FMG 300D
FMG 300E
FMG 300F
FMG 3700F
FMG 3900E
FMG 4000D
FMG 4000E
FMG 400E

当然更多的我们会用的 VM 的方式部署 FortiManger，包括公有云和私有云平台：

VM 平台
Aliyun
AWS
Azure
Google GCP
Hyper-V
KVM
Oracle
VMware ESXi
Xen

重要

更多型号参考：https://www.fortinet.com/products/management/fortimanager#models-specs ↗

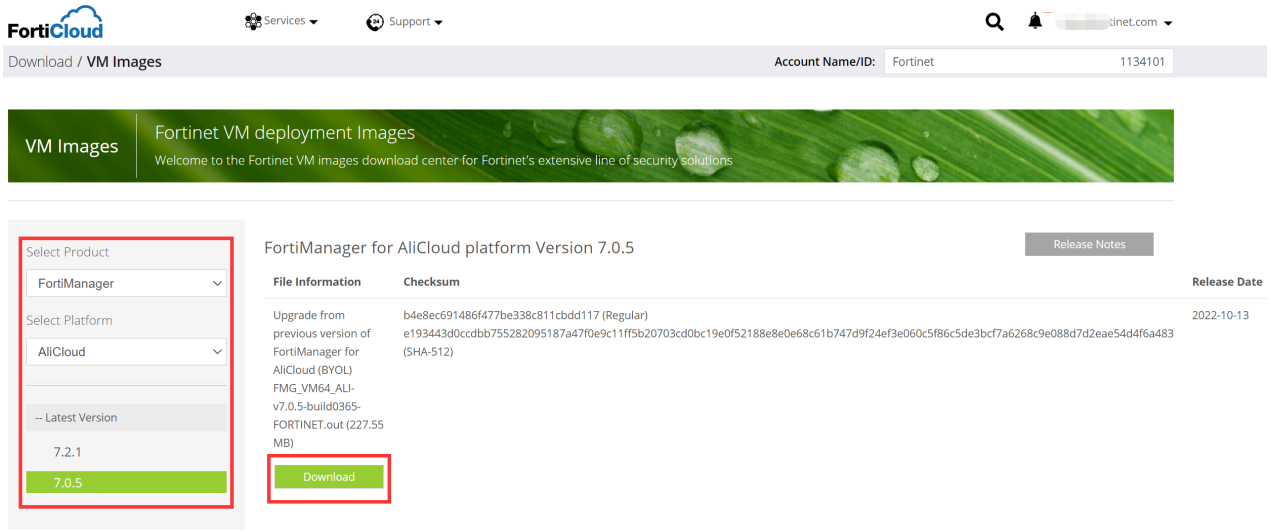
产品规格参考：https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/fortimanager.pdf ↗

VM 安装手册

https://docs.fortinet.com/cloud-solutions ↗

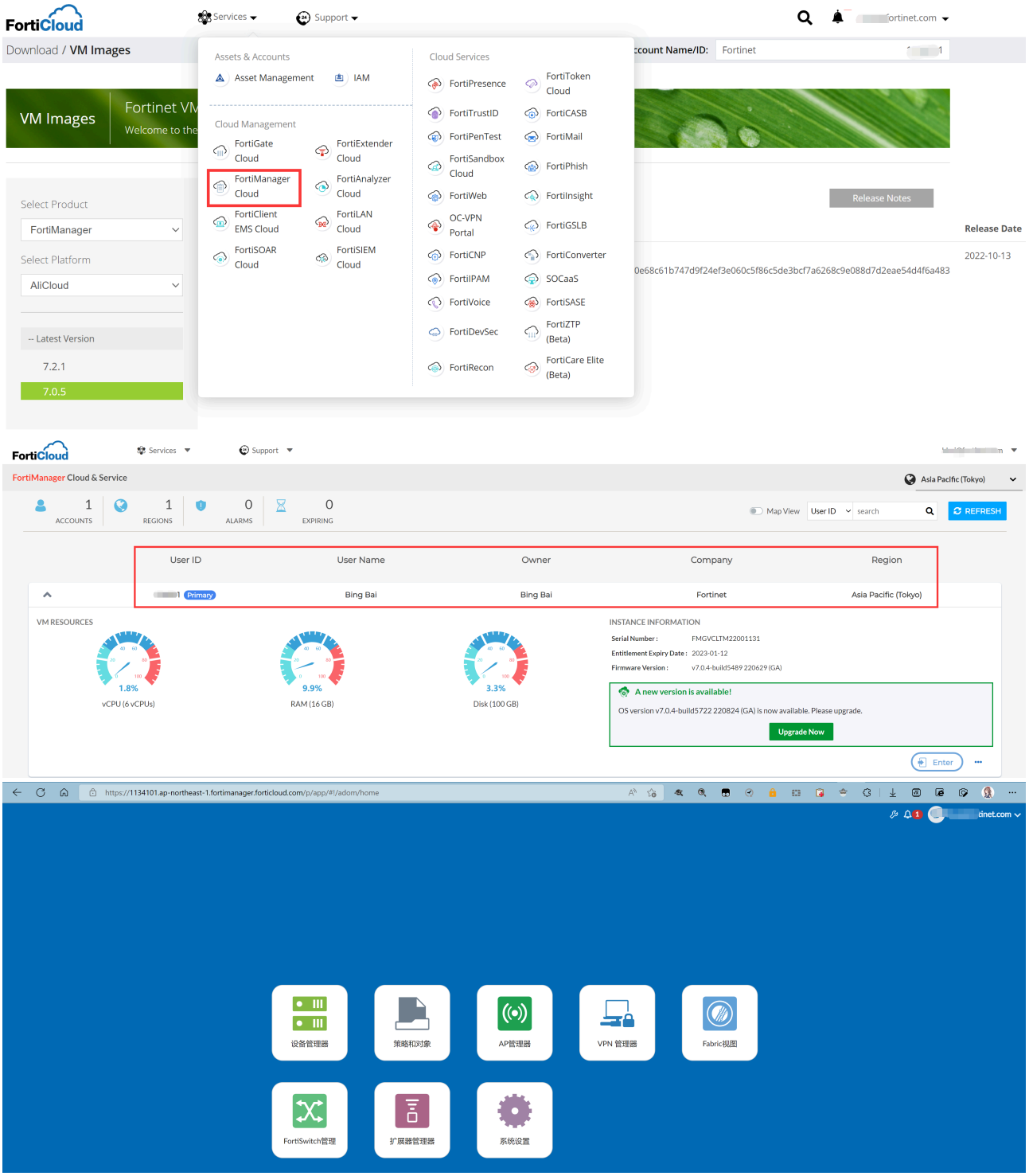
VM 版本下载

1. 进入 https://support.fortinet.com/Download/VMImages.aspx ↗ 并登陆。
2. 选择产品为 FortiManager，并选择想要下载的虚拟化平台和对应版本。



FortiManagerCloud

https://fortimanager.forticloud.com/ ↗ ，通过 FortiCloud 访问 FortiManager，服务器不在国内，相对比较慢。


FortiCloud
Services
Support
Download / VM Images
Assets & Accounts
Asset Management
IAM
Cloud Management
FortiGate Cloud
FortiExtender Cloud
FortiManager Cloud
FortiAnalyzer Cloud
FortiClient EMS Cloud
FortiLAN Cloud
FortiSOAR Cloud
FortiSIEM Cloud
Cloud Services
FortiPresence
FortiToken Cloud
FortiTrustID
FortiCASB
FortiPenTest
FortiMail
FortiSandbox Cloud
FortiPhish
FortiWeb
FortiInsight
OC-VPN Portal
FortiGSLB
FortiCNP
FortiConverter
FortiIPAM
SOCaaS
FortiVoice
FortiSASE
FortiDevSec
FortiZTP (Beta)
FortiRecon
FortiCare Elite (Beta)
VM Images
Fortinet VM
Welcome to the
Select Product
FortiManager
Select Platform
AliCloud
-- Latest Version
7.2.1
7.0.5
Release Notes
Release Date
2022-10-13
FortiManager Cloud & Service
Asia Pacific (Tokyo)
ACCOUNTS
REGIONS
ALARMS
EXPIRING
Map View
User ID
search
REFRESH
User ID
User Name
Owner
Company
Region
Primary
Bing Bai
Bing Bai
Fortinet
Asia Pacific (Tokyo)
VM RESOURCES
1.8%
vCPU (6 vCPUs)
9.9%
RAM (16 GB)
3.3%
Disk (100 GB)
INSTANCE INFORMATION
Serial Number : FMGVCLTM22001131
Entitlement Expiry Date : 2023-01-12
Firmware Version : v7.0.4-build5489 220629 (GA)
A new version is available!
OS version v7.0.4-build5722 220824 (GA) is now available. Please upgrade.
Upgrade Now
Enter
https://1134101.ap-northeast-1.fortimanager.forticloud.com/p/app/#!/adom/home
设备管理器
策略和对象
AP管理器
VPN 管理器
Fabric视图
FortiSwitch管理
扩展器管理器
系统设置

FortiManger 版本推荐

重要

FortiGate 版本与 FortiManager、FortiAnalyzer 产品版本的适应关系可通过如下网站查询：https://docs.fortinet.com/compatibility-tool/fortimanager ↗

固件命名规则

重要

FMG_A-B-C-FORTINET.out

如：**FMG_VM64-v7.0.12-build0623**-FORTINET.out

其中：

- **A**部分表示 FortiManager 型号，不同型号有不同的固件。
- **B**代表版本，有各个不同系列的版本，这里使用 x.y.z（如 7.0.12）来详细说明：
  - x.y 表示大版本号，我们已经发布的大版本有 v6.0、v6.2、v6.4、v7.0、v7.2 和 v7.4。
  - z 表示大版本下持续发布的小版本，z 越大，表示该版本越稳定。
- **C**表示内部代码编号。
- 正在运行的设备版本号，可以在 CLI 下使用 get system status 观察到，如下所示。

```bash
FortiManager # get sys status
Platform Type                    : FMG-VM64
Platform Full Name               : FortiManager-VM64
Version                          : v7.2.5-build1574 240313 (GA)
Serial Number                    : FMG-VMTM23011097
BIOS version                     : 04000002
Hostname                         : FMG-VM64
Max Number of Admin Domains      : 10000
Max Number of Device Groups      : 10000
Admin Domain Configuration       : Enabled
FIPS Mode                        : Disabled
HA Mode                          : Stand Alone
Branch Point                     : 1574
Release Version Information      : GA
Current Time                     : Wed Apr 24 17:19:20 HKT 2024
Daylight Time Saving             : Yes
Time Zone                        : (GMT+8:00) Beijing, ChongQing, HongKong, Urumqi.
x86-64 Applications              : Yes
Disk Usage                       : Free 476.43GB, Total 491.15GB
File System                      : Ext4
License Status                   : Valid
```

选择固件的原则

重要

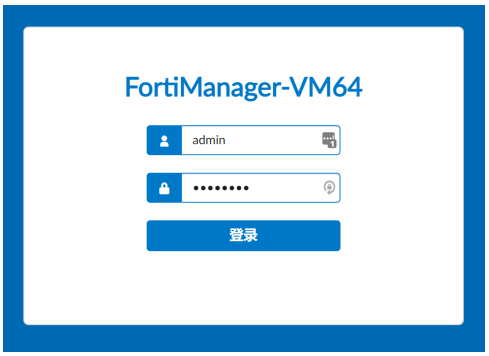
B (x.y.z) 中的 z 越大越稳定可靠。

1. 首先确认被 FortiManager 管理的 FortiGate 的版本（FortiOS 推荐版本请参考系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗章节），FortiManager 的版本可以向下兼容 2 个版本。例如 FortiGate 的版本为 FortiOS 7.0.15，那么 FortiManager 的版本可以使用的版本为 7.0、7.2 或 7.4。
2. 选定了大版本（x.y）后，小版本（z）尽量选择当前最新版本。例如选择了 FortiManager 7.0 版本，那么就使用该大版本（7.0）下的已发布的最新小版本（12，截止到 2024 年 4 月 24 日）。

FortiManger 网管登陆

默认的 FortiManger 和 FortiGate 的默认网管类似，默认情况下都可以通过 GUI 和 CLI 进行网管 FortiManger。

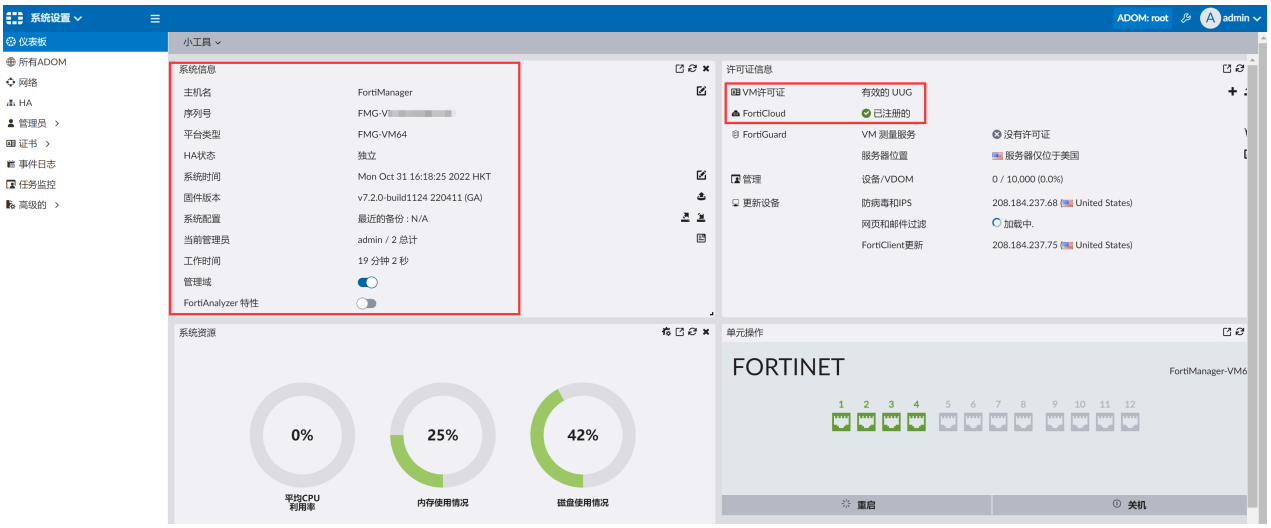
1. FortiManger 的管理口（mgmt1 或 port1）IP 地址默认为 192.168.1.99，可以进入 FortiManager 管理页面或 Console CLI 下修改。



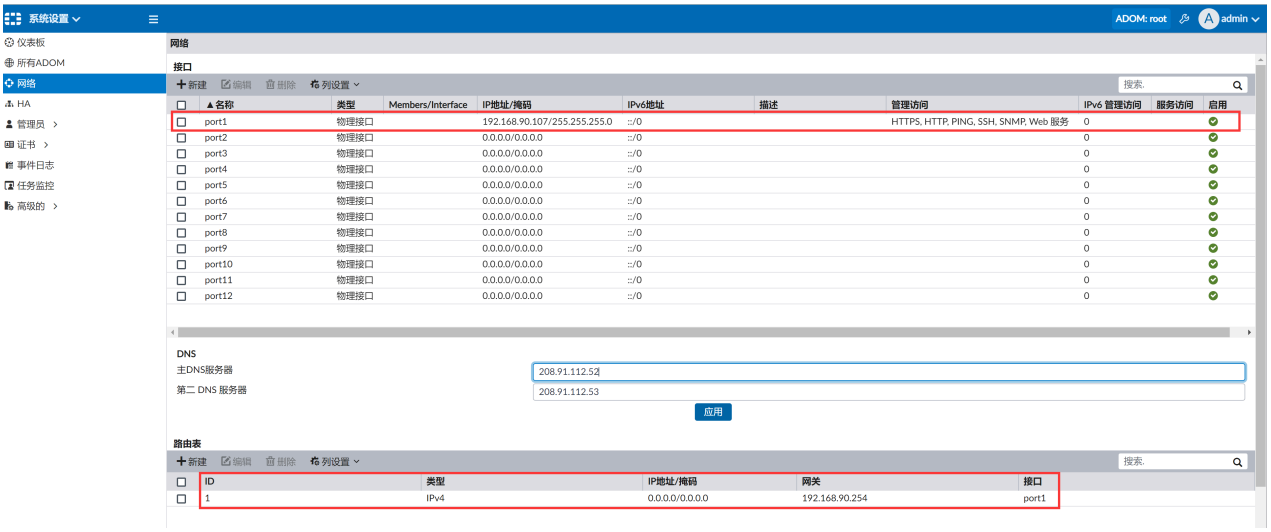
2. 登陆成功，可以看到管理界面。



3. 进入系统设置视图，可以看到系统信息与 License 信息。



4. 在网络页面，可以配置设备接口的 IP 地址及路由表。


系统设置
ADOM: root
admin
仪表板
所有ADOM
网络
HA
管理员
证书
事件日志
任务监控
高级的
网络
接口
新建
编辑
删除
列设置
搜索
名称
类型
Members/Interface
IP地址/掩码
IPv6地址
描述
管理访问
IPv6 管理访问
服务访问
启用
port1 物理接口 192.168.90.107/255.255.255.0 ::/0 HTTPS, HTTP, PING, SSH, SNMP, Web 服务 0
port2 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port3 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port4 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port5 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port6 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port7 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port8 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port9 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port10 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port11 物理接口 0.0.0.0/0.0.0.0 ::/0 0
port12 物理接口 0.0.0.0/0.0.0.0 ::/0 0
DNS
主DNS服务器
208.91.112.52
第二 DNS 服务器
208.91.112.53
应用
路由表
新建
编辑
删除
列设置
搜索
ID
类型
IP地址/掩码
网关
接口
1 IPv4 0.0.0.0/0.0.0.0 192.168.90.254 port1

FortiManger ADOM

- ADOM 简介
- 开启 ADOM
- ADOM 的高级特性

ADOM 简介

FortiManger 的 ADOM 和 FortiGate 的 VDOM 类似，是一个独立的虚拟管理空间，多个 ADOM 之间彼此独立，一台 FMG 可以划分多个 ADOM。

什么时候需要用到 ADOM?

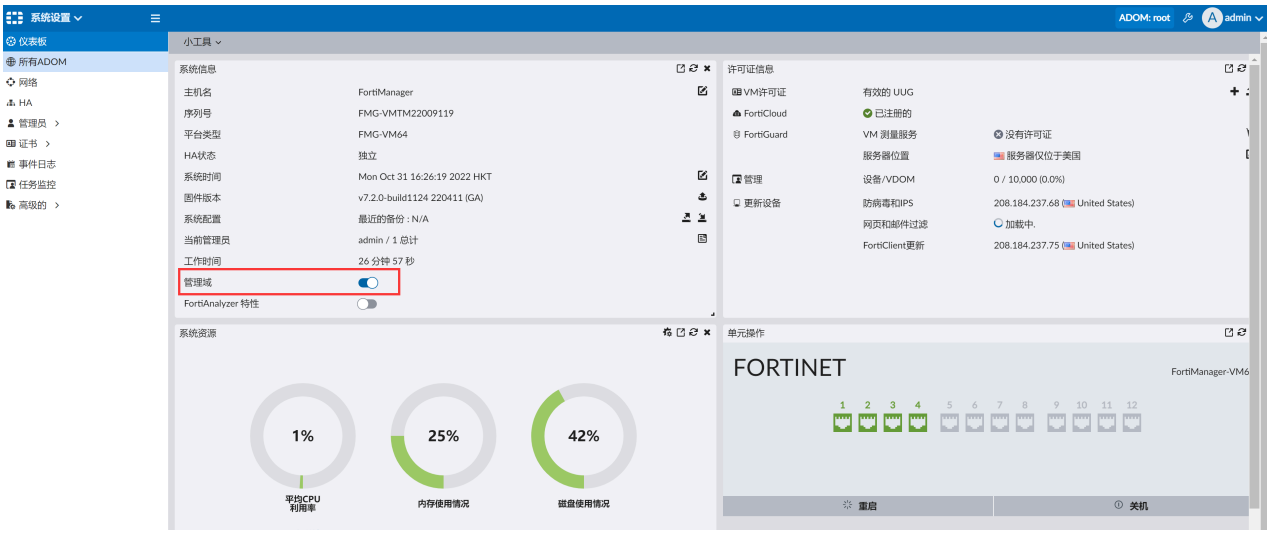
- 被管理的设备拥有不同的固件版本，如 FortiOS 6.4/7.0/7.2，因为不同版本的有效命令行语法有一定区别，因此必须以版本进行区分 ADOM，每一个版本平台拥有独立的空间。
- 管理员管理权限和区域划分，比如南区的管理员只能管理南区的设备，北区的管理员只能管理北区的设备，东区的管理员只能管理东区的设备，超级管理员可以管理所有的设备，那么这样的情况下就需要划分南区、北区、东区三个 VDOM，分别指派给三个区域的管理员。
- MSSP 业务情景，每一个租户拥有一个独立的 ADOM，管理自己独立的设备。
- 拥有不同的设备类型，同时拥有多种设备类型，比如 FortiGate, FortiMail, FortiWeb，那么则需要针对 FortiMail 和 FortiWeb 再分别创建两个 VDOM。
- 业务规划或测试隔离，根据业务或者部门来划分 ADOM，生产网络和测试网络等，拥有不同的 ADOM，彼此隔离，互不影响。
- ADOM 最大数目取决于设备型号，就和 FortiGate 类似的，如果开启 VDOM 功能的话，那么默认的那个防火墙就叫做 root，FortiManger 也是这样的，默认的那个 FortiManger 也叫做 root。其他的 ADOM 自己手工命名并创建。

开启 ADOM

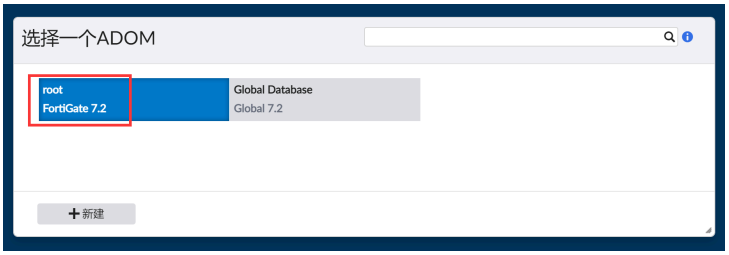
1. 进入系统设置。



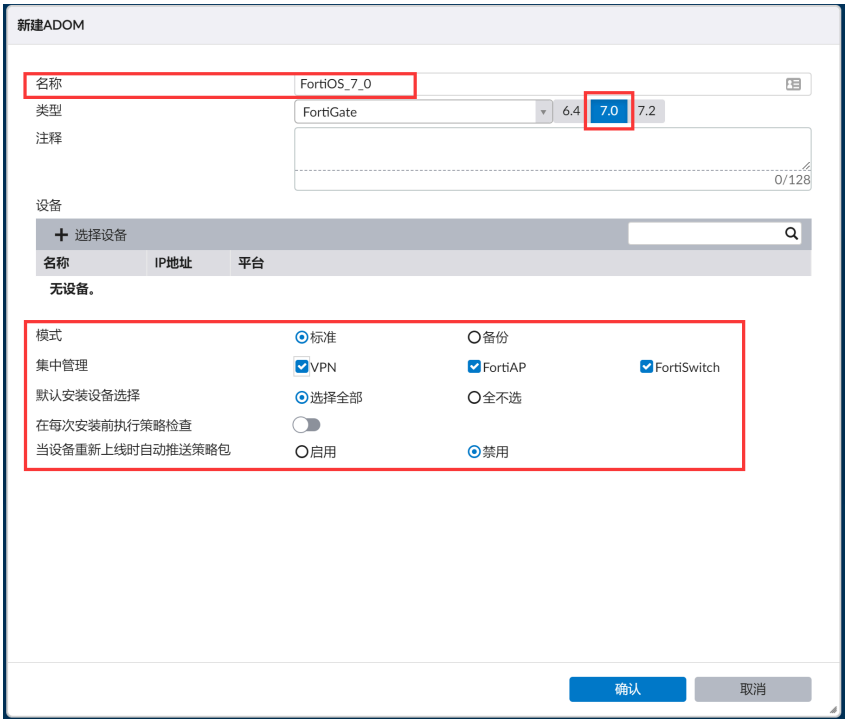
2. 开启管理域功能。



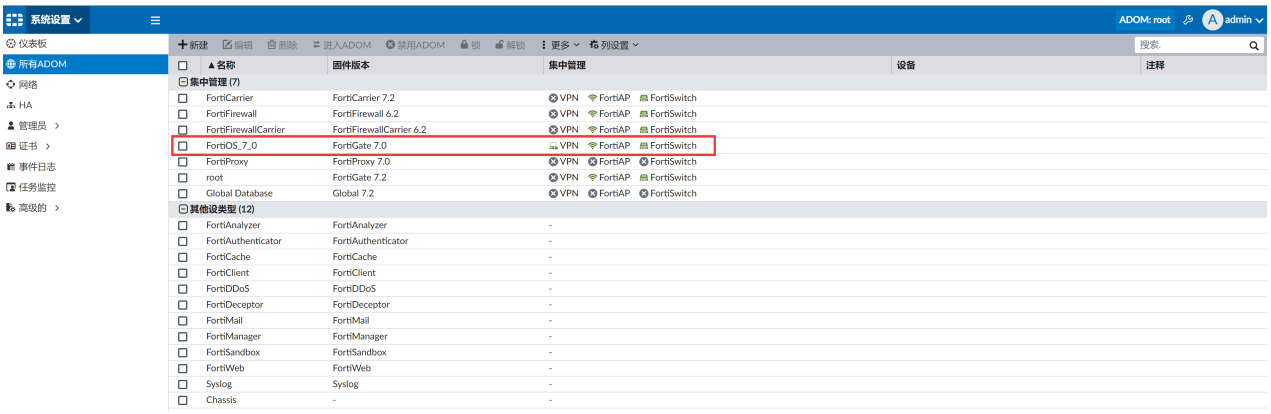
3. 再次登录 FortiManger 管理界面，原来的 FortiManger 变成了 root ADOM。



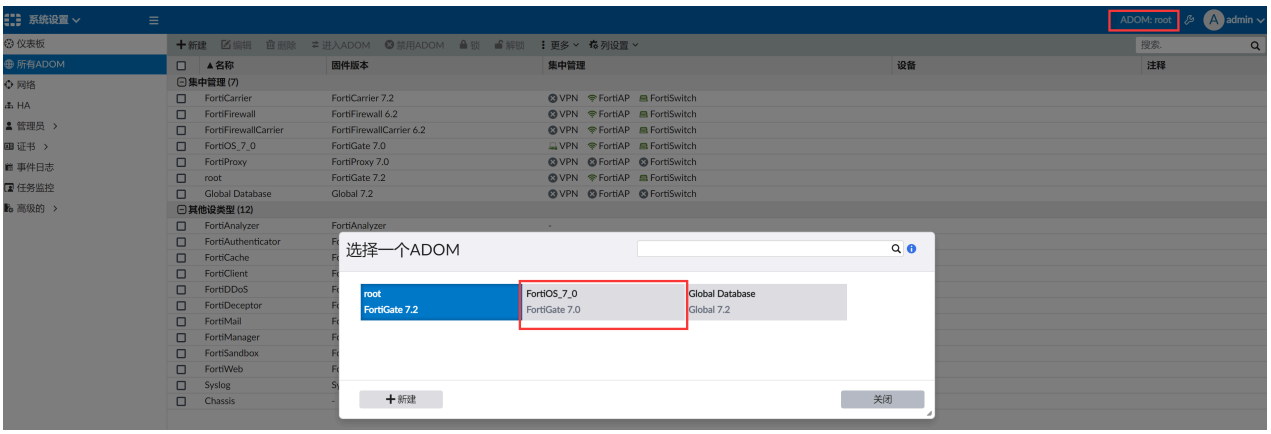
4. 点击新建按钮，新建一个 7.0 版本的 ADOM。



5. 创建完成后，在系统设置中可以看到新建的 ADOM。



6. 点击页面右上角的 ADOM 选择器，可以看到新建的 ADOM，点击可进入该 ADOM。



ADOM 的高级特性

ADOM 运行模式



标准模式

提供从 FortiManager 到 ADOM 和受管设备的配置更改的完全访问权限。

- 默认的模式
- ADOM 读写权限
- 所有管理面板可用
- 能通过 FortiManager 的界面对被管理的设备配置变更
- 直接在受管设备上进行的更改能自动更新到 FortiManager

> **重要**
>
> 通过 FGFM 协议，被管理的设备每 5 秒会对配置进行差异检查，并且会将差异修订发送到 FortiManager。

标准模式面板截图：



备份模式

直接在受管设备上进行备份配置更改，主要目的是备份直接在受管设备上进行的配置更改。

- ADOM 是只读的
- 更少的可用管理面板
- 只能通过脚本的方式来对被管理的设备进行配置变更
- 可以直接在被管理的设备上进行配置变更
- 必须满足配置修订要在 FortiManager 上备份的特定条件：
  - 配置变更然后管理员会话超时
  - 配置变更然后登出
  - 配置变更然后重启
  - 从被管理设备上手动备份配置

标准模式面板截图：



ADOM 高级模式

一台防火墙的不同 VDOM 可以加入到不同的 FortiManger 的 ADOM 里，默认不可以。



标准

FortiGate 设备仅能被添加到一个 ADOM，默认模式。

高级

一台 FortiGate 的不同 VDOM 能被指派到不同的 ADOM，全局应用到全部 ADOM，这个适合高级用户开启，相对比较复杂。

FortiManger 管理员

- 管理员账号管理
- 配置远程管理员
- 管理员和 ADOM 关联
- ADOM 的 Workspace 模式

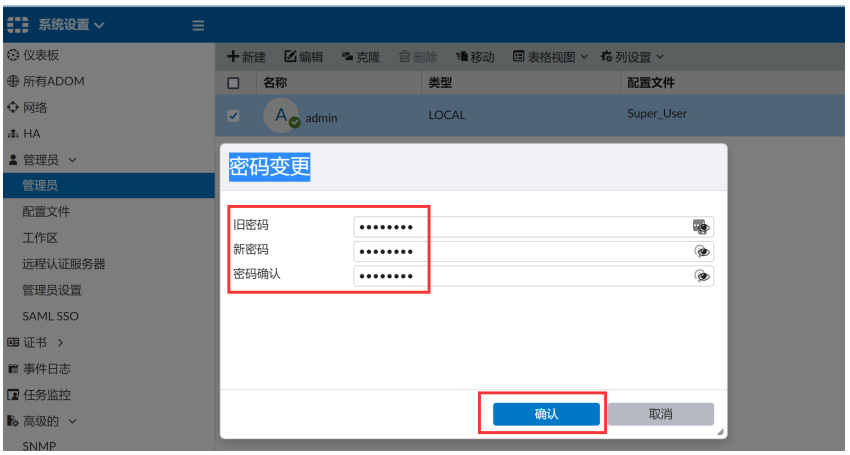
管理员账号管理

修改管理员密码

1. 进入系统设置 → 管理员界面，邮件点击管理员，选择密码变更。



2. 输入旧密码和新密码，并点击确认。



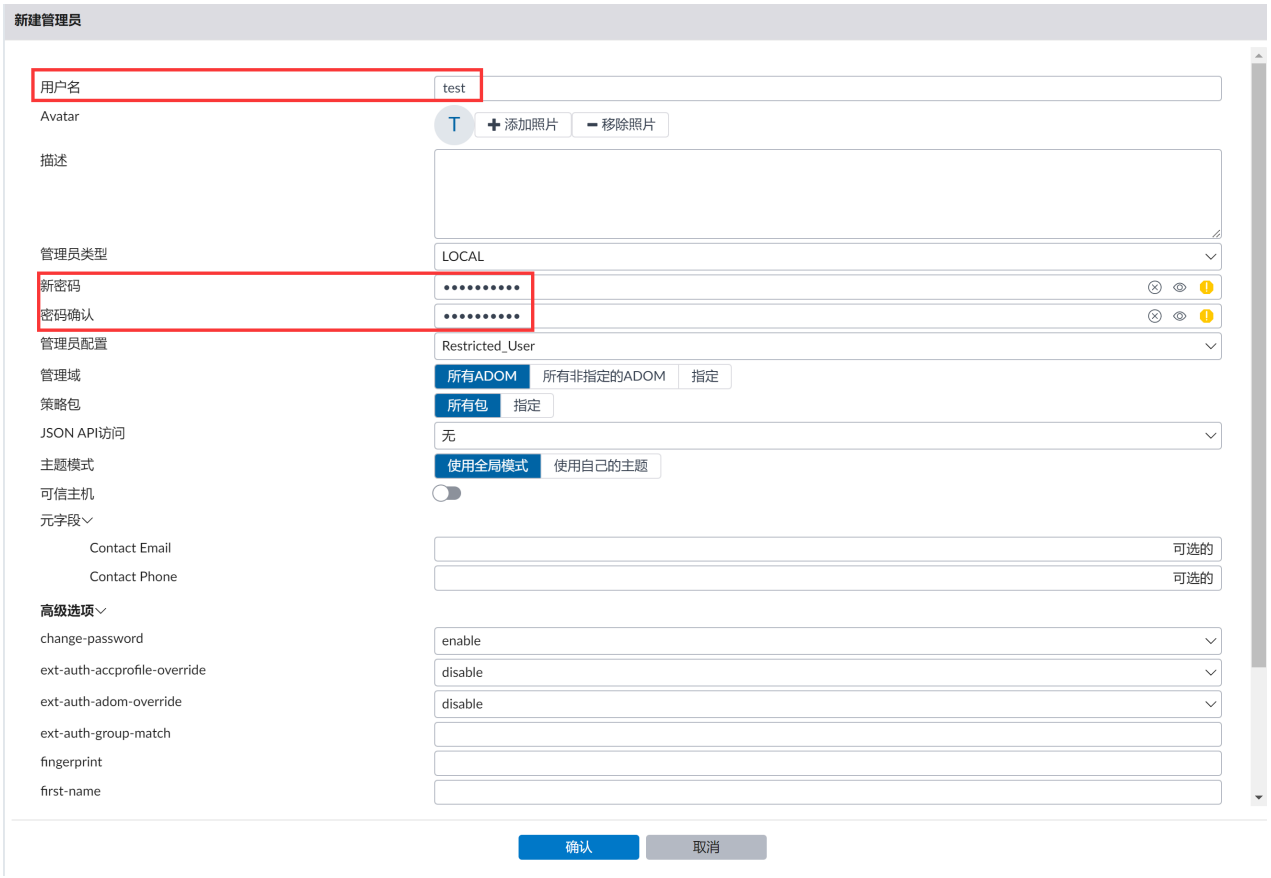
> **重要**
>
> 第一次登陆防火墙的时候也会提示让我们设置一个新的密码。这个位置是手工修改管理员的密码位置。

新建管理员

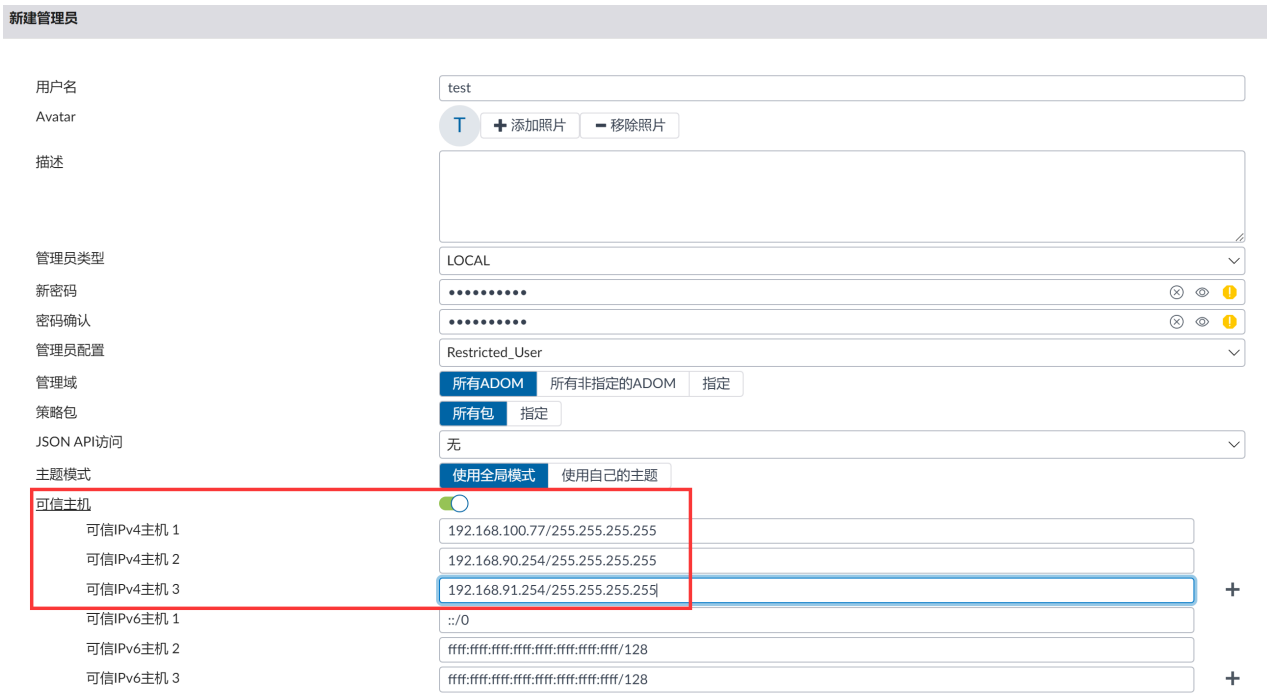
1. 进入管理员页面，点击新建按钮。



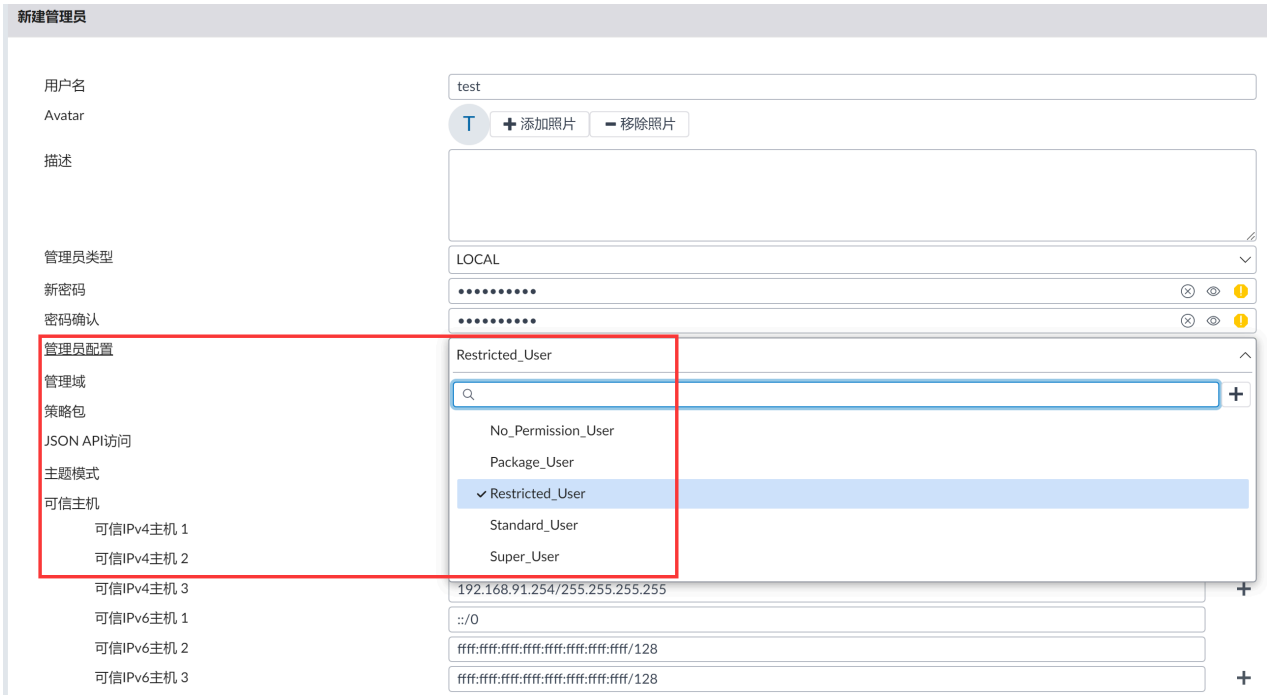
2. 在弹出的新建管理员页面中，填写管理员名称与密码。



3. 配置可信主机，只有在可信性主机范围内的 IP 地址可以管理 FortiManeger。



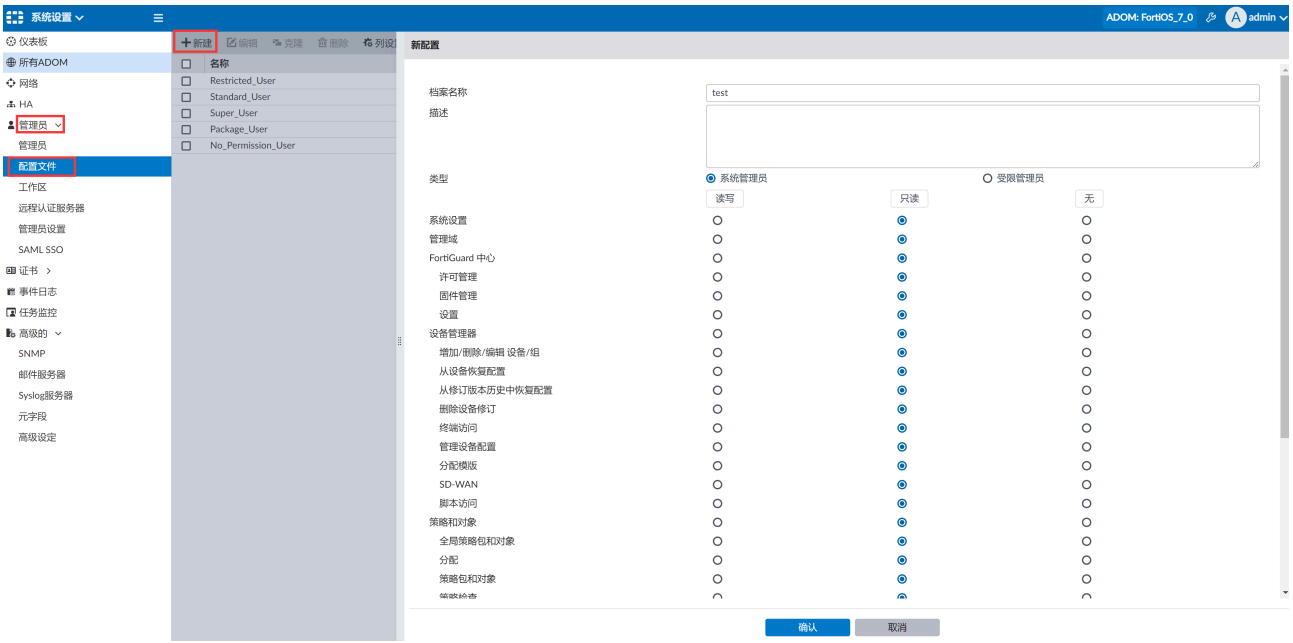
4. 配置管理员权限。



- Restricted_User：未启用系统权限，并且对所有设备权限具有只读访问权限。
- Standard_User：未启用系统权限，但对所有设备权限具有读/写访问权限。
- Super_User：已启用所有系统和设备权限。
- Package_User：已启用读/写策略包和对象权限，并具有系统和其他内容的只读访问权限。

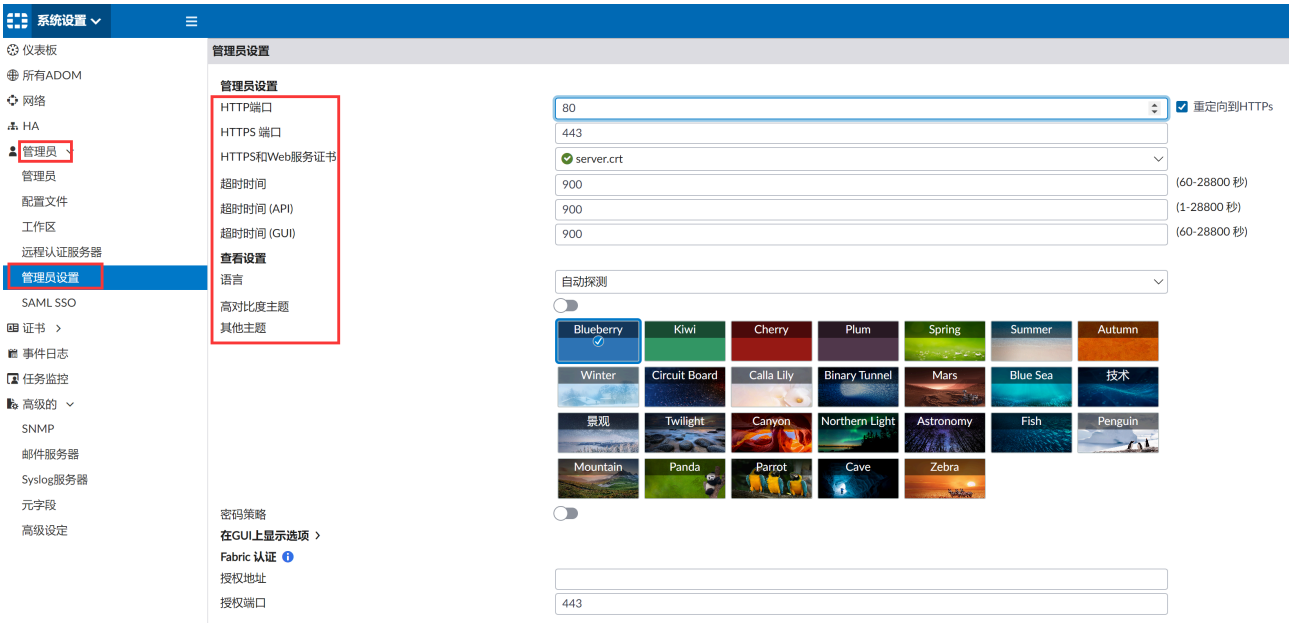
管理员配置文件

可以针对自己的需求进行新建管理员配置文件。



管理员高级配置

可修改管理端口、调整管理员超时时间、调整管理 FMG 的数字证书、语言、主题等等。

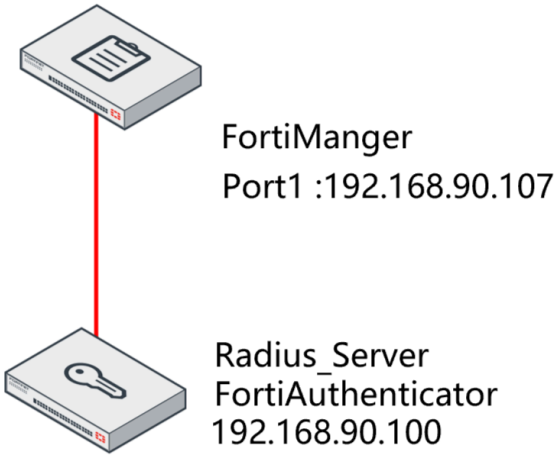

系统设置
仪表板
所有ADOM
网络
HA
管理员
管理员
配置文件
工作区
远程认证服务器
管理员设置
SAML SSO
证书
事件日志
任务监控
高级的
SNMP
邮件服务器
Syslog服务器
元字段
高级设定
管理员设置
管理员设置
HTTP端口
HTTPS 端口
HTTPS和Web服务证书
超时时间
超时时间 (API)
超时时间 (GUI)
查看设置
语言
高对比度主题
其他主题
密码策略
在GUI上显示选项
Fabric 认证
授权地址
授权端口
80
重定向到HTTPs
443
server.crt
900
(60-28800 秒)
900
(1-28800 秒)
900
(60-28800 秒)
自动探测
Blueberry
Kiwi
Cherry
Plum
Spring
Summer
Autumn
Winter
Circuit Board
Calla Lily
Binary Tunnel
Mars
Blue Sea
技术
景观
Twilight
Canyon
Northern Light
Astronomy
Fish
Penguin
Mountain
Panda
Parrot
Cave
Zebra
443

配置远程管理员

需求

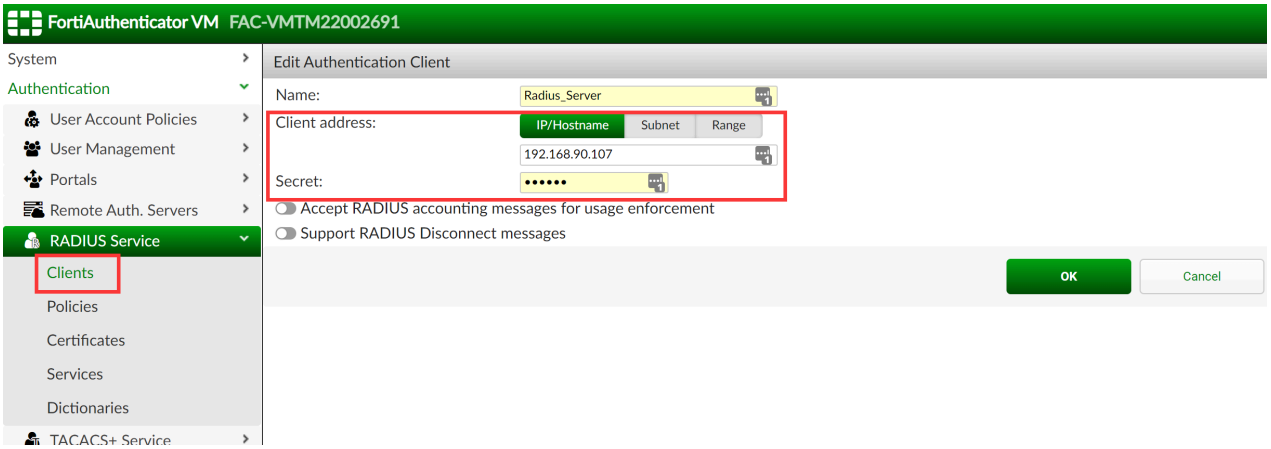
要求使用 test 这个远程账号登陆 FortiManger，如果远程认证服务器失效，则使用 test 这个账号的逃生密码 test 进行登陆，另外其他的远程用户也需要可以登陆到 FortiManger。

拓扑

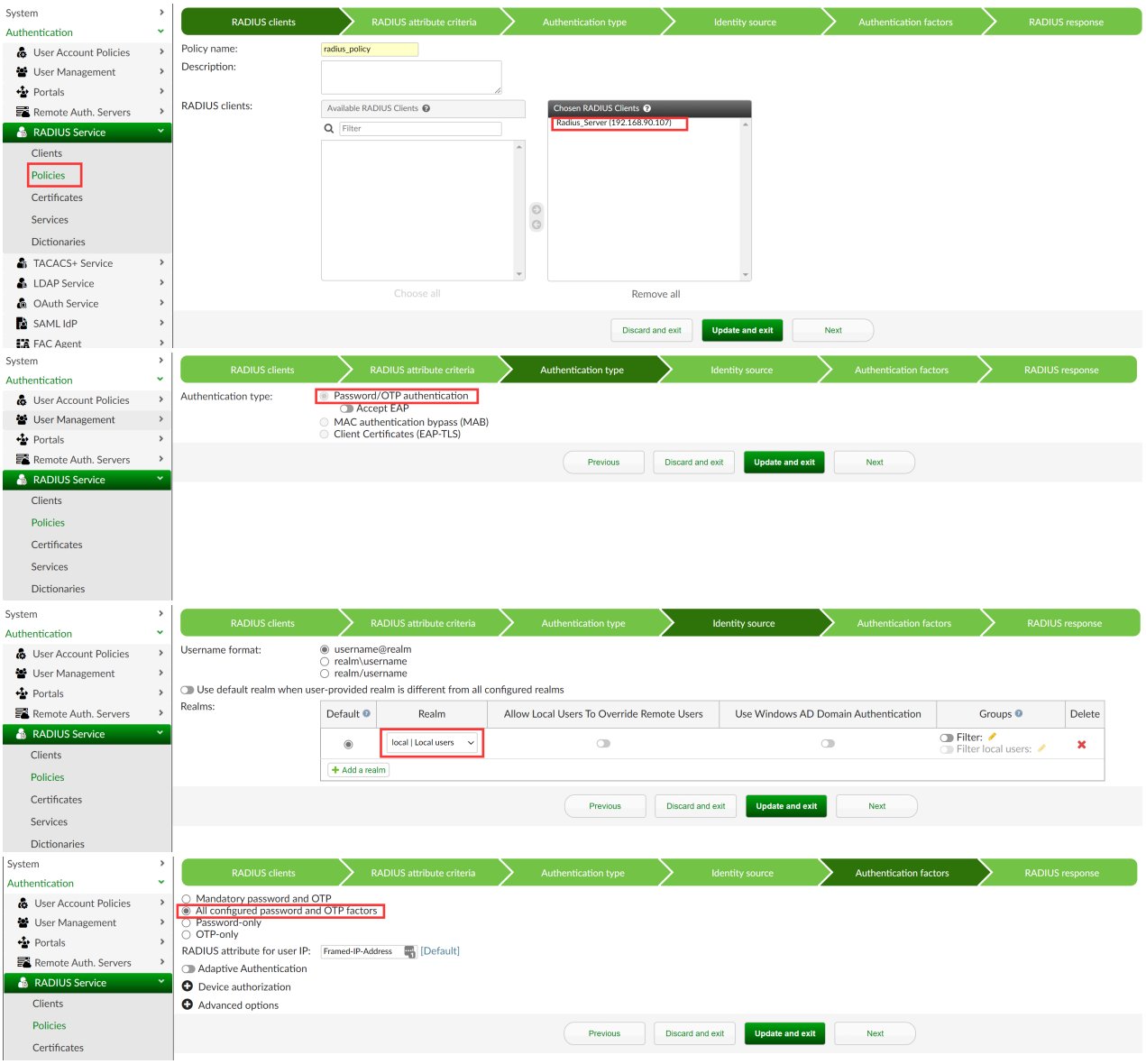


配置步骤

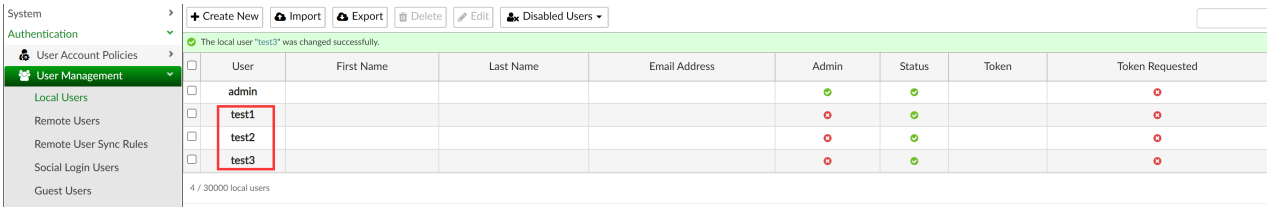
1. FortiAuthenticator 上配置 Radius 认证 client，IP 地址需要保函 FortiManger 的管理 IP 地址。



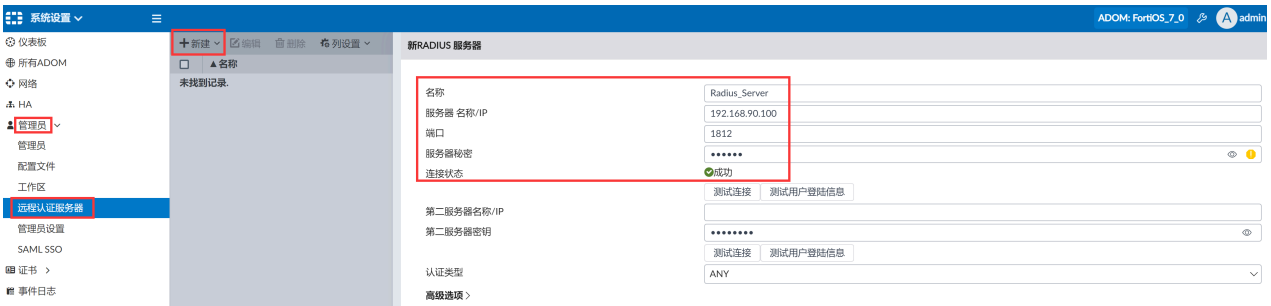
2. 配置 Radius Profile，认证方式为本地密码认证。



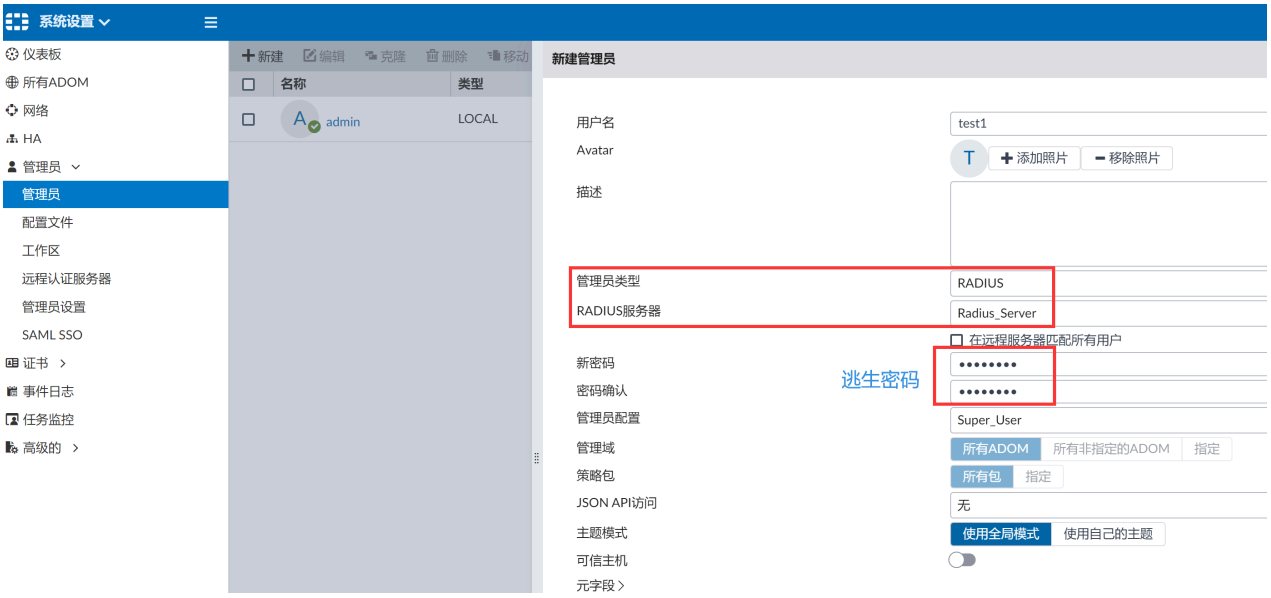
3. 配置本地 Radius 账户 test1、test2、test3（密码均为 11111111）。



4. 在 FortiManager 上创建远程 Radius 认证服务器，并测试成功。



5. 新建 Radius 管理员账户 test1（密码 Fortinet123#），不勾选“在远程服务器匹配所有用户”，这个账号必须先去 Radius 认证，认证通过可以登陆，如果 Radius 服务器出现了故障，则使用逃生账户密码（Fortinet123#）也可以登陆 FortiManger。



6. 新建 Radius 管理员账户 test2，勾选“在远程服务器匹配所有用户”，这个帐号必须在 Radius 服务器认证，无本地逃生密码。

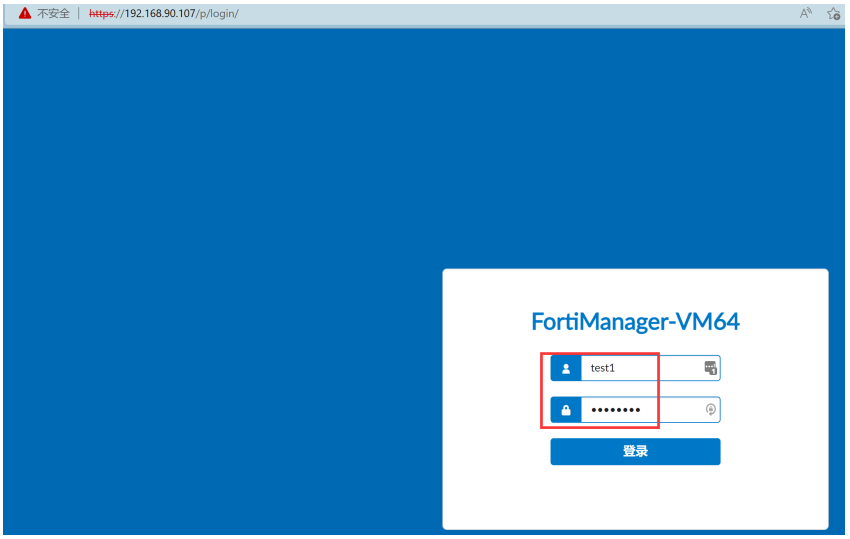


重要

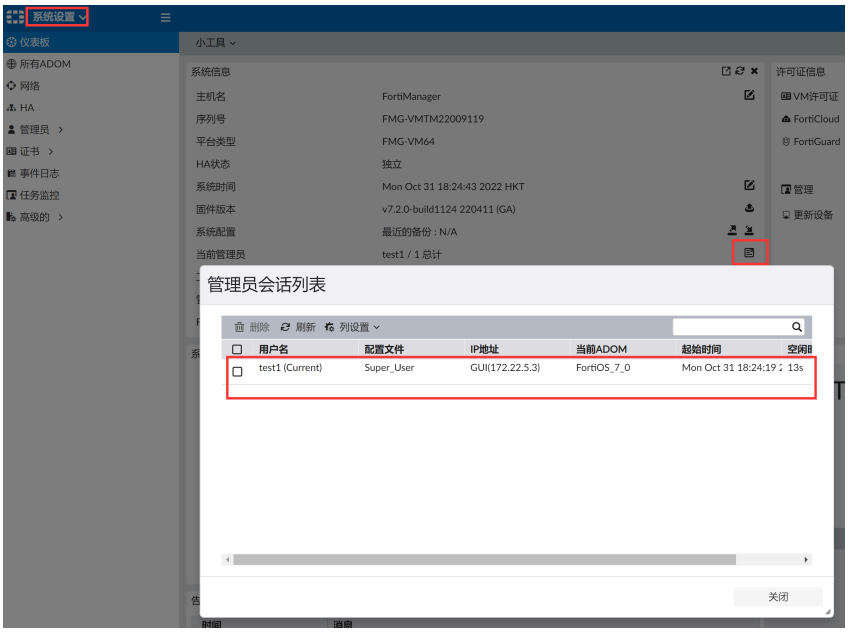
关于存在于 FMG 本地的 admin 账号，通常如果对于账号管理不是特别的严格的话，我们都是建议保留这个 admin 账号，这样比较好，防止没有配置 test1 这样的明细的具备逃生密码的账号且 Radius 服务器故障的时候，无法登陆到 FMG。本地的 admin 相当于是另外一个逃生账号，当然密码需要保存好，另外平时尽量不要使用到本地的 admin 账号，只有当 Radius 服务器故障，且没有 test1 这样的逃生账号的时候，这个时候本地的 admin 才用来作为逃生账号使用。

结果验证

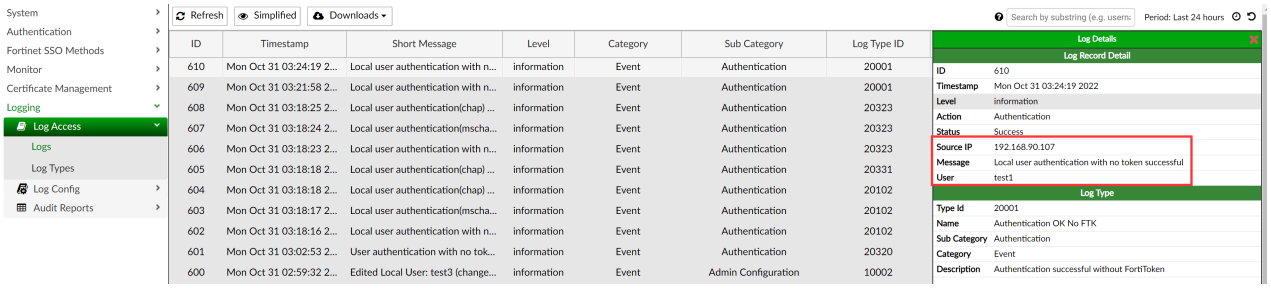
1. 使用 test1 用户的 Radius 密码 11111111 登陆 FortiManager。



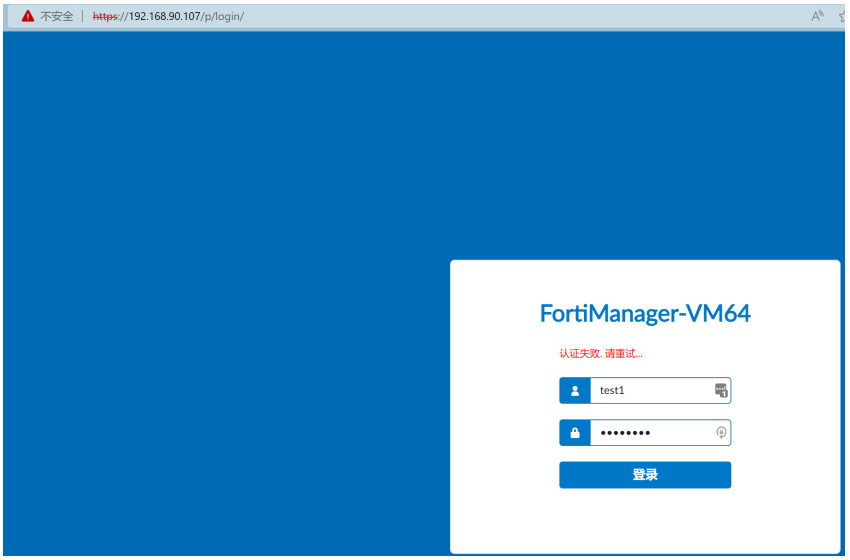
2. 登陆成功后，在系统设置中可以看到在线用户。



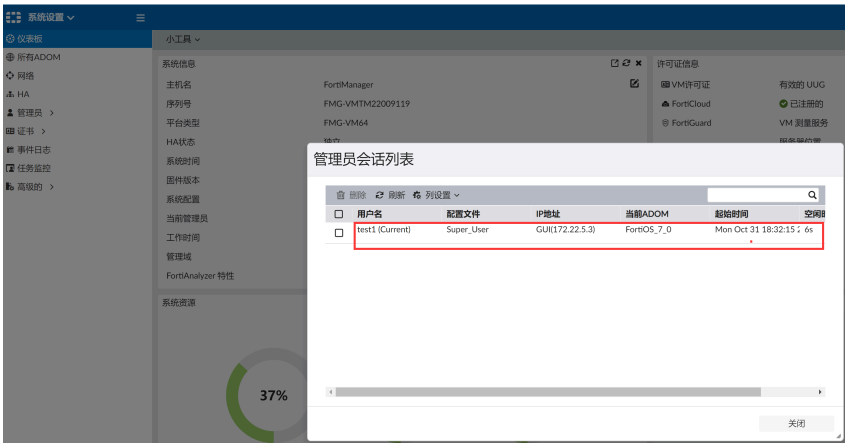
3. FAC 上 Radius 的登陆成功日志。



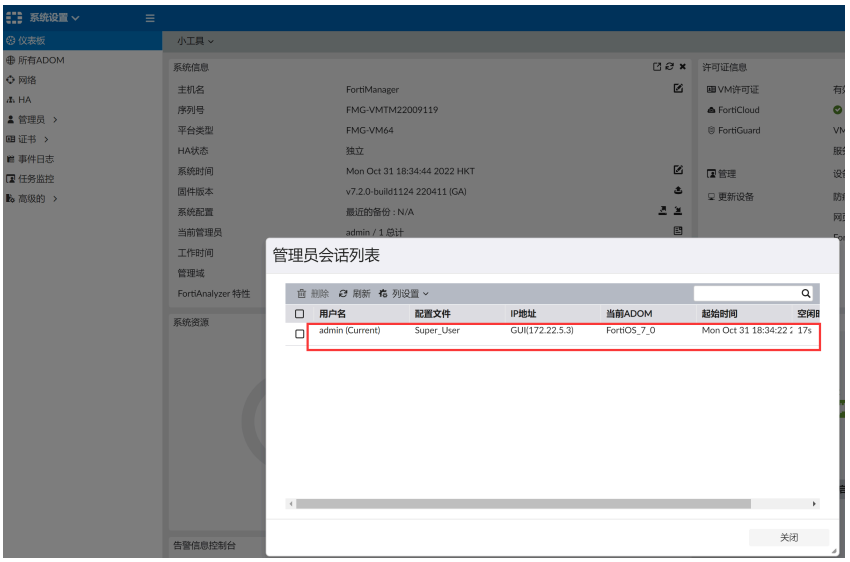
4. 将 Radius 服务器关闭，再次使用 test1 和他的 Radius 密码 11111111 登陆 FortiManager，登陆失败。



5. 这个时候就需要使用我们为 test1 设置的本地逃生密码 Fortinet123# 了，继续使用 zhangsan 账号和逃生密码登陆 FortiManger。



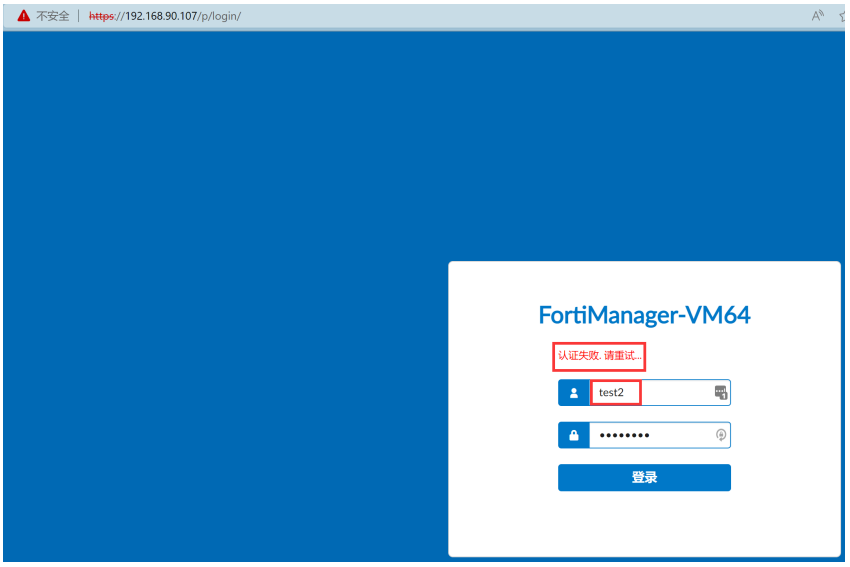
6. 当然此时使用本地的 admin 账号也是可以登陆 FortiManger 的另外一个逃生账号。



重要

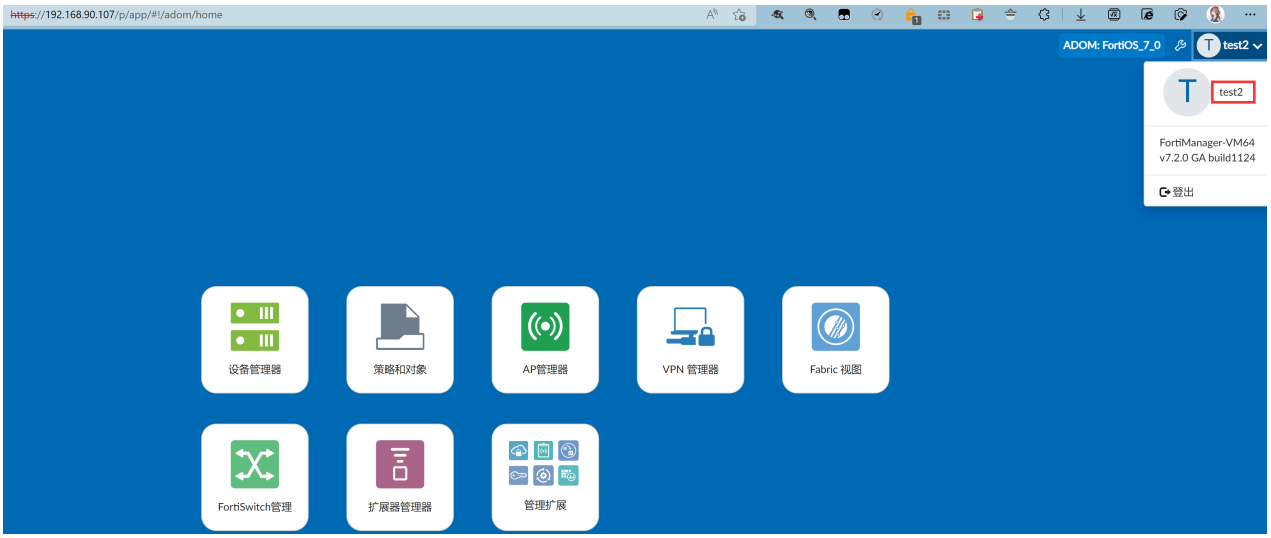
假如将 FMG 本地的 admin 账号删除了，这个情况下 admin 是没办法通过 Radius 认证成功的（Radius 故障了，不可达了），这个时候只能用 test1 的逃生密码登陆 FortiManger。而现在模拟的是我们没有删除本地的 admin 账号，因此这个情况下我们可以使用 admin 账号继续登陆 FortiManger，相当于另外的一个逃生账号。

7. 使用 test2 账户（Radius 用户密码 11111111）登陆，该账户只允许 Radius 认证，没有逃生密码，由于 FAC Radius 服务器已经关闭，该用户登录失败。



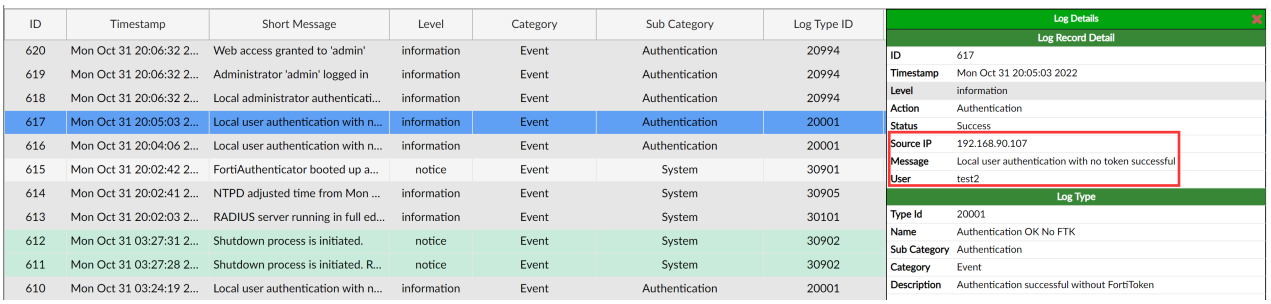
8. 恢复 FAC 的 Radius 服务状态。

9. 再次使用 test2（密码 11111111）用户登陆，使用 Radius 认证登陆成功。



> **重要**
>
> test2 配置的是标准管理员，不是超级管理员，因此看到的 FortiManger 内容有限。

.0. 查看 FAC Radius 认证成功的日志。



ID	Timestamp	Short Message	Level	Category	Sub Category	Log Type ID
620	Mon Oct 31 20:06:32 2...	Web access granted to 'admin'	information	Event	Authentication	20994
619	Mon Oct 31 20:06:32 2...	Administrator 'admin' logged in	information	Event	Authentication	20994
618	Mon Oct 31 20:06:32 2...	Local administrator authenticati...	information	Event	Authentication	20994
617	Mon Oct 31 20:05:03 2...	Local user authentication with n...	information	Event	Authentication	20001
616	Mon Oct 31 20:04:06 2...	Local user authentication with n...	information	Event	Authentication	20001
615	Mon Oct 31 20:02:42 2...	FortiAuthenticator booted up a...	notice	Event	System	30901
614	Mon Oct 31 20:02:41 2...	NTPD adjusted time from Mon ...	information	Event	System	30905
613	Mon Oct 31 20:02:03 2...	RADIUS server running in full ed...	information	Event	System	30101
612	Mon Oct 31 03:27:31 2...	Shutdown process is initiated.	notice	Event	System	30902
611	Mon Oct 31 03:27:28 2...	Shutdown process is initiated. R...	notice	Event	System	30902
610	Mon Oct 31 03:24:19 2...	Local user authentication with n...	information	Event	Authentication	20001

Log Details

Log Record Detail

ID	617
Timestamp	Mon Oct 31 20:05:03 2022
Level	information
Action	Authentication
Status	Success
Source IP	192.168.90.107
Message	Local user authentication with no token successful
User	test2

Log Type

Type Id	20001
Name	Authentication OK No FTK
Sub Category	Authentication
Category	Event
Description	Authentication successful without FortiToken

总结

推荐方式 1（以上配置举例）

- 配置一个明细的 test1 账号 + 逃生密码
- 配置另外一个匹配远程 Radius 所有用户 test2
- 保留本地 admin 账号（作为另外一个远程 Radius 的本地逃生账号）
- 远程用户全部可以登录，如果 Radius 服务器故障，可以使用 test1 的账号和逃生密码登陆，另外还可以使用本地的 admin 账号逃生登陆



推荐方式 2

- 保留本地 admin 账号（作为远程 Radius 的本地逃生账号）
- 配置一个匹配远程 Radius 所有用户的账户
- 远程用户全部可以登录，如果 Radius 服务器故障，可以使用本地的 admin 账号逃生登陆



管理员和 ADOM 关联

需求

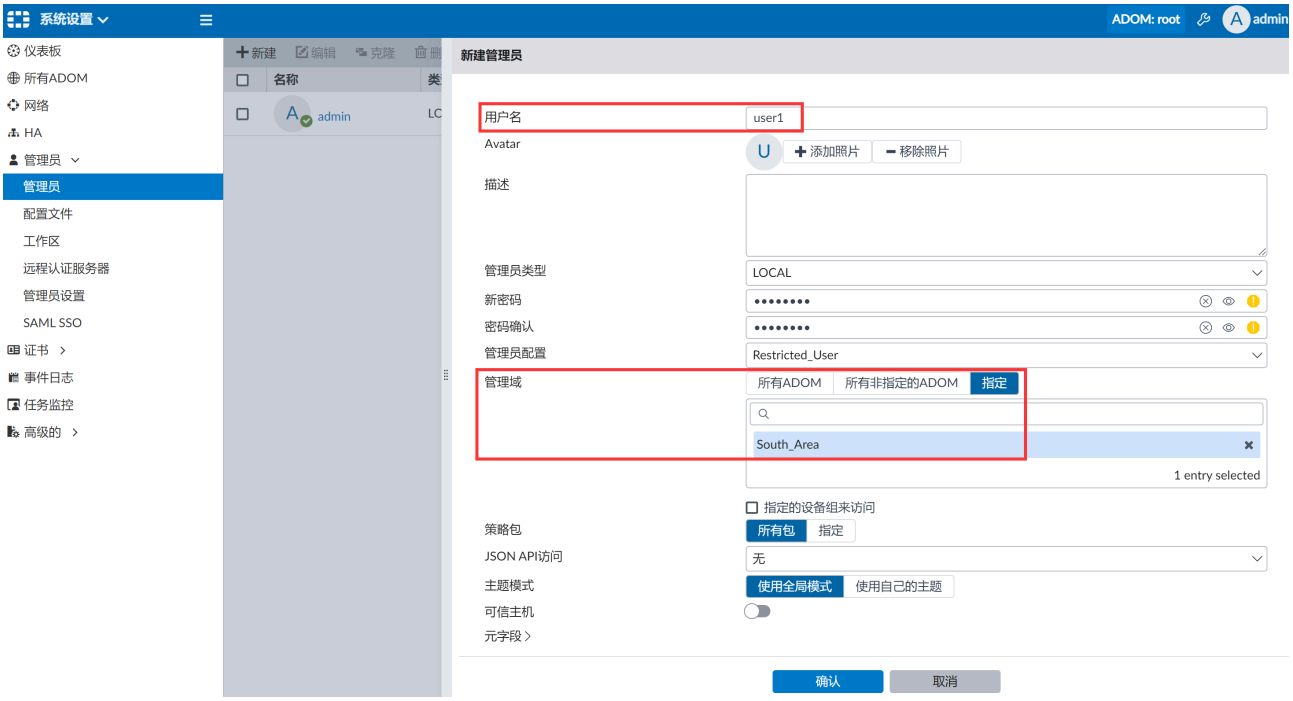
对于分区管理的需求，我们通常建议将管理员与 ADOM 进行绑定，每个区域的管理员只可以看到属于自己区域的管理设备，对于其他区域的设备不可见，这样可以实现管理员和设备的分区独立管理。

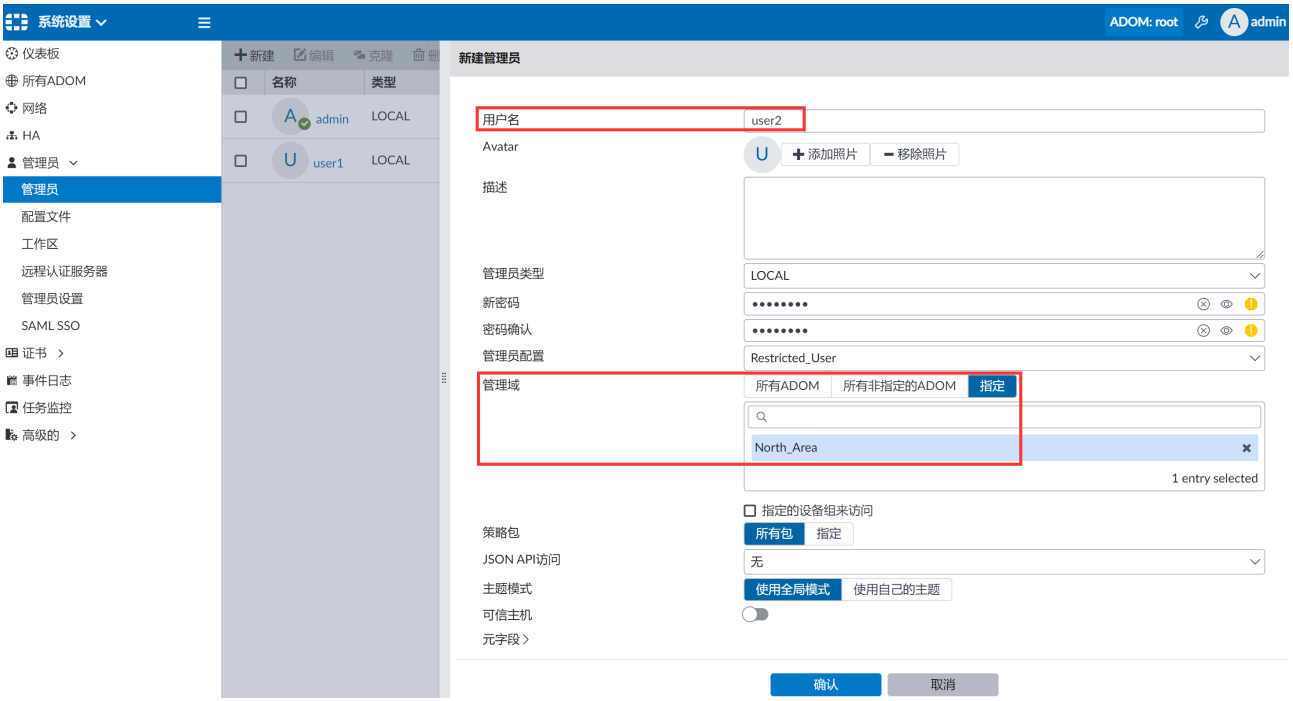
举例：有三个 ADOM，South_Area、North_Area、East_Area，三个区域的管理员分别管理各自的 FortiGate，相互之间独立彼此独立，管理员 user1 需要管理 ADOM South_Area，管理员 user2 需要管理 ADOM North_Area，管理员 user3 需要管理 ADOM East_Area，而超级管理员 admin 则可以管理所有区域的所有防火墙。

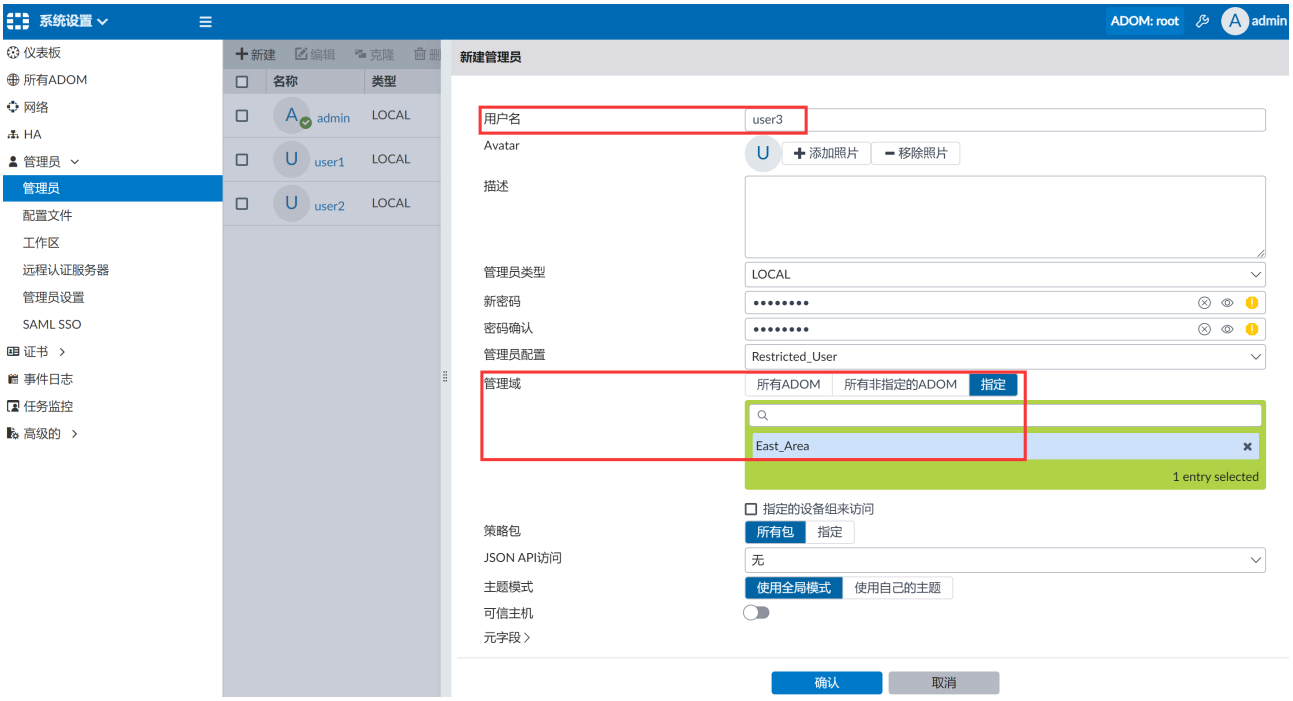


配置步骤

新建 user1、user2、user3 三个本地账号并绑定到相应的 ADOM 上。


系统设置
ADOM: root
admin
仪表板
所有ADOM
网络
HA
管理员
管理员
配置文件
工作区
远程认证服务器
管理员设置
SAML SSO
证书
事件日志
任务监控
高级的
新建
编辑
克隆
名称
admin
新建管理员
用户名
user1
Avatar
添加照片
移除照片
描述
管理员类型
LOCAL
新密码
密码确认
管理员配置
Restricted_User
管理域
所有ADOM
所有非指定的ADOM
指定
South_Area
1 entry selected
指定的设备组来访问
策略包
所有包
指定
JSON API访问
无
主题模式
使用全局模式
使用自己的主题
可信主机
元字段
确认
取消


系统设置
ADOM: root
admin
仪表板
所有ADOM
网络
HA
管理员
管理员
配置文件
工作区
远程认证服务器
管理员设置
SAML SSO
证书
事件日志
任务监控
高级的
新建
编辑
克隆
名称
类型
admin
LOCAL
user1
LOCAL
新建管理员
用户名
user2
Avatar
添加照片
移除照片
描述
管理员类型
LOCAL
新密码
密码确认
管理员配置
Restricted_User
管理域
所有ADOM
所有非指定的ADOM
指定
North_Area
1 entry selected
指定的设备组来访问
策略包
所有包
指定
JSON API访问
无
主题模式
使用全局模式
使用自己的主题
可信主机
元字段
确认
取消





结果验证

1. 使用 user1 登陆 FortiManager，可以看到 user1 只能访问 South_Area ADOM。



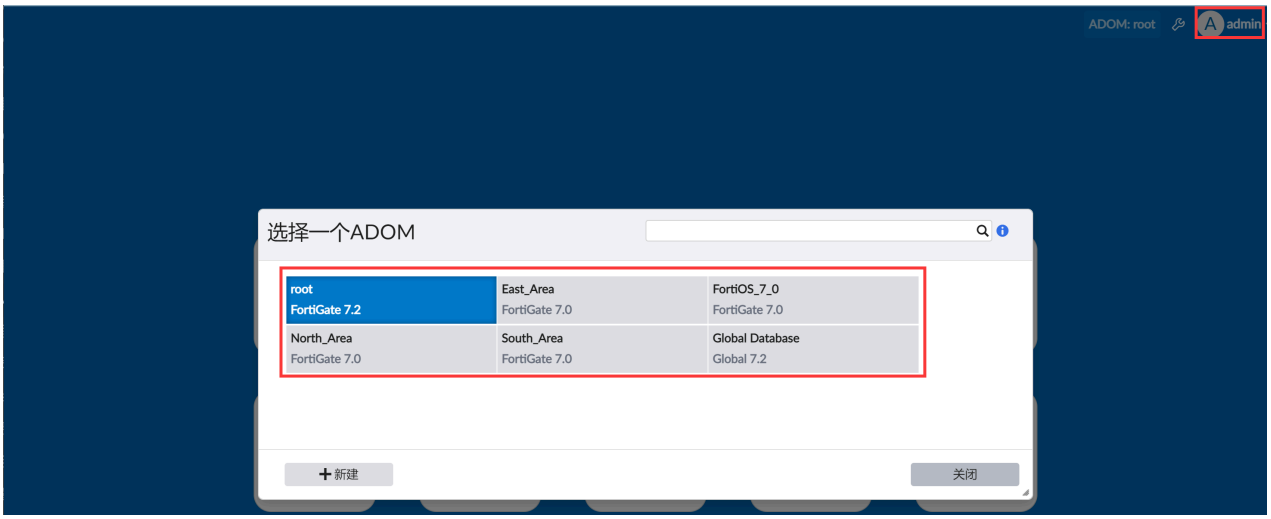
2. 使用 user2 登陆 FortiManager，可以看到 user2 只能访问 North_Area ADOM。



3. 使用 user3 登陆 FortiManager，可以看到 user3 只能访问 East_Area ADOM。



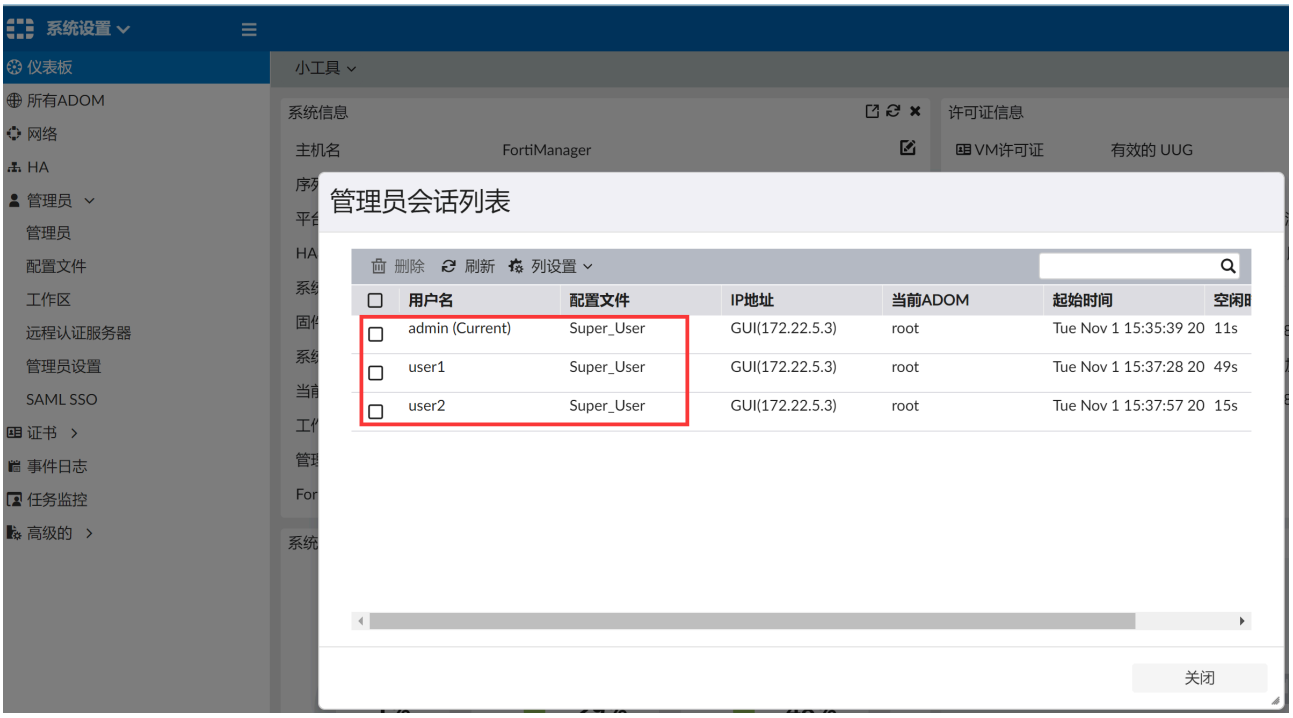
4. 使用 admin 登陆 FortiManager，可以看到 admin 能访问所有 ADOM。



ADOM 的 Workspace 模式

需求

当多个管理员同时登陆到同一个 ADOM，如果多个管理员试图同时更改相同的 ADOM，则会导致小概率配置冲突。



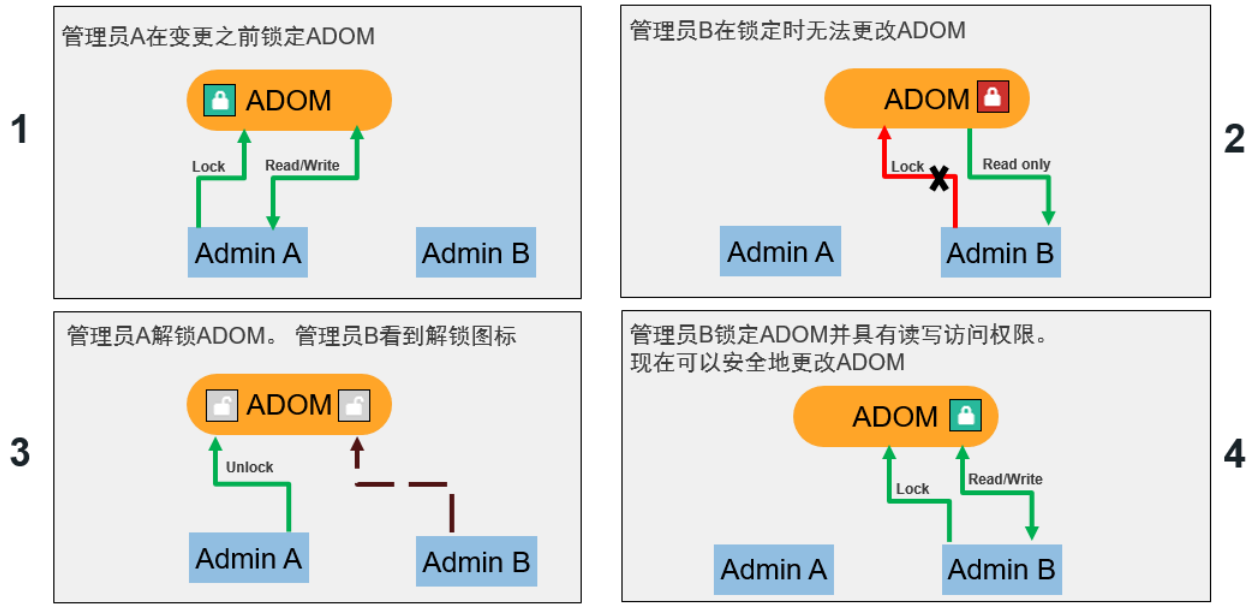
为了避免这种情况，我们可以在 CLI 下开启 workspace 模式。禁用并发管理员对 ADOM 的配置访问，添加 ADOM 锁定，同时只能有一个管理员具备读/写访问权限，所有其他管理员都只具有读访问权限。

```
config system global
  set workspace-mode normal
end
```

重要

在启用 Workspace 模式以保存其工作之前，必须通知其他管理员，否则可能会导致一些登录用户丢失未保存的数据。

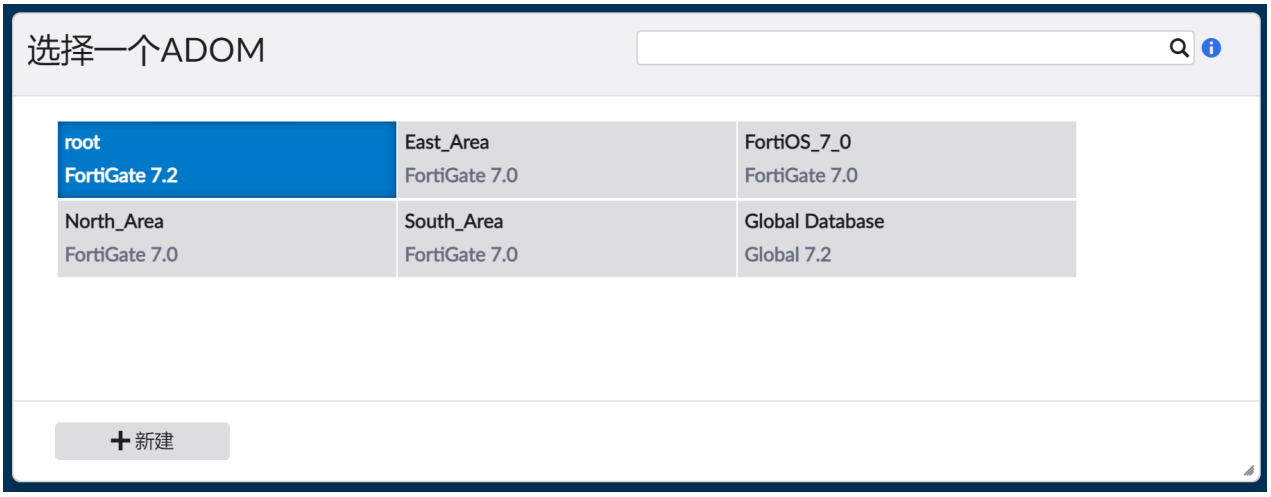
功能简介



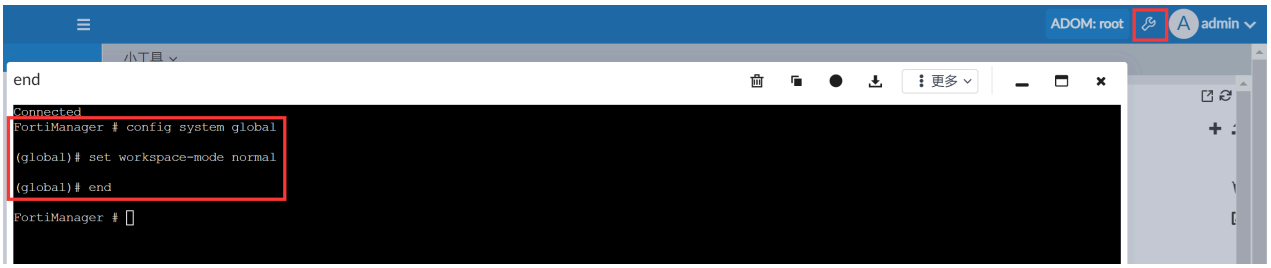
以此避免多个管理员同时修改同一配置的冲突风险。

配置步骤

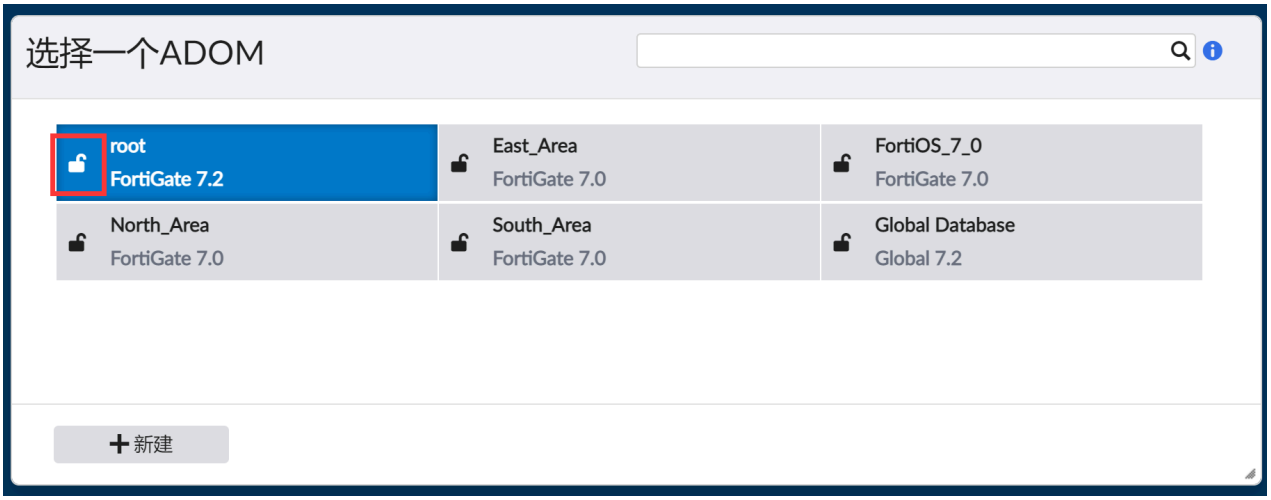
1. 使用 admin 账户登陆 FortiManager，默认情况下，workspace-mode 是“set workspace-mode disabled”，因此没有任何的锁定标志。



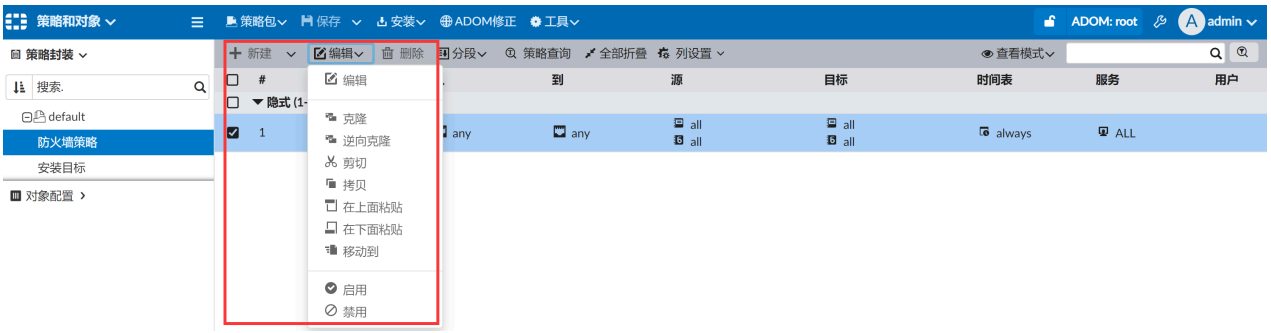
2. 进入 Web CLI，在“config system global”下开启“set workspace-mode normal”。



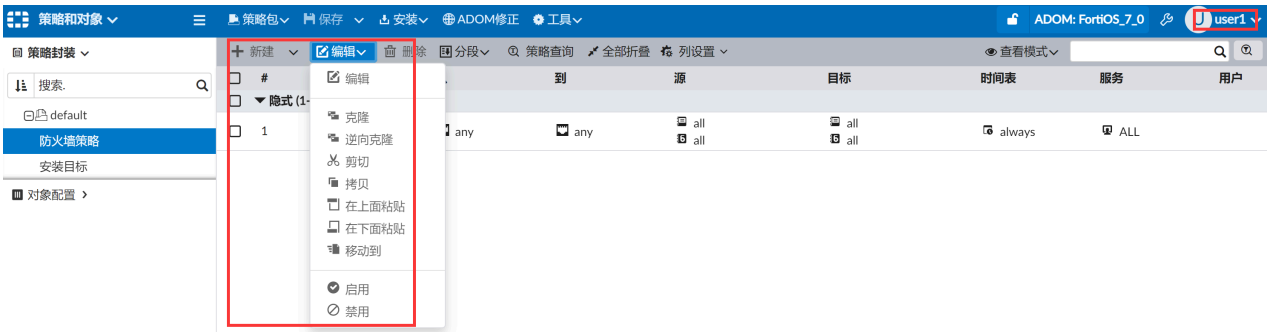
3. 再次登录 FortiManger，每一个 ADOM 都多了一把锁的标志，在没有锁定之前都仅仅拥有只读权限。



4. 在这个情况下，针对管理设备所有的配置按钮都是灰色的，都是只读权限。



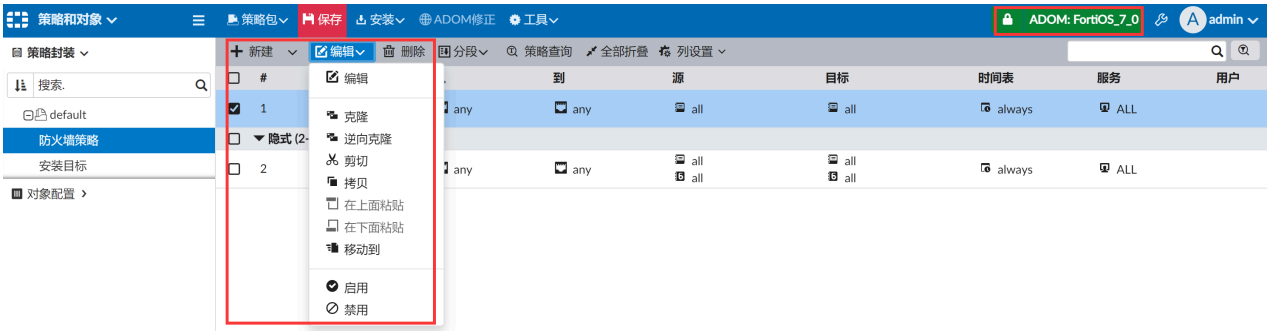
5. 使用另外一个管理员 user1 登陆查看，和 admin 一样的结果，仅仅具有只读权限。



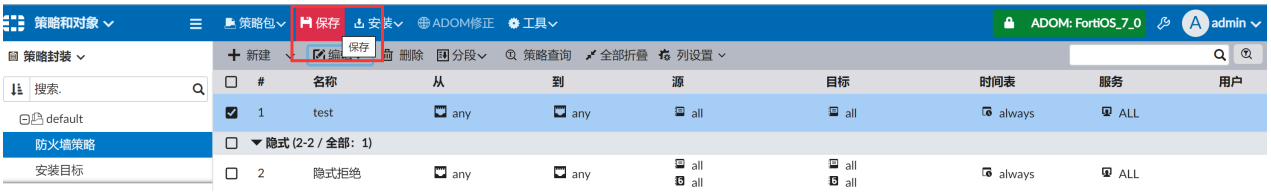
6. 此时 admin 账号想要在策略包里面添加一条策略（需要写的权限），这个时候我们就需要使用 admin 账户将这个 root ADOM 进行锁定了，只有锁定了 ADOM 的账户才会拥有读/写的操作权限。点击页面右上角的锁定按钮，锁定当前 ADOM。



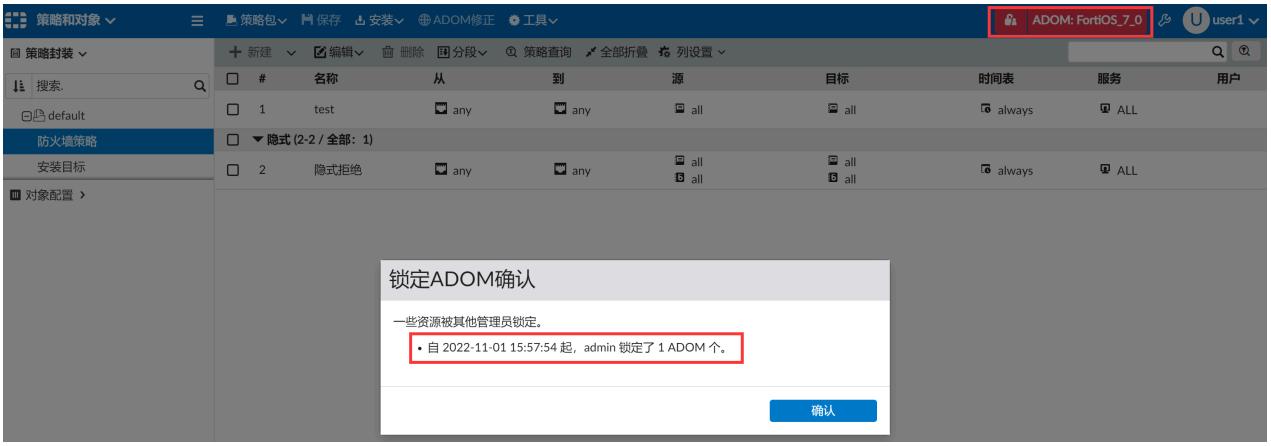
7. 新建或编辑策略，可以看到针对该 ADOM，admin 账户已经具备写权限。



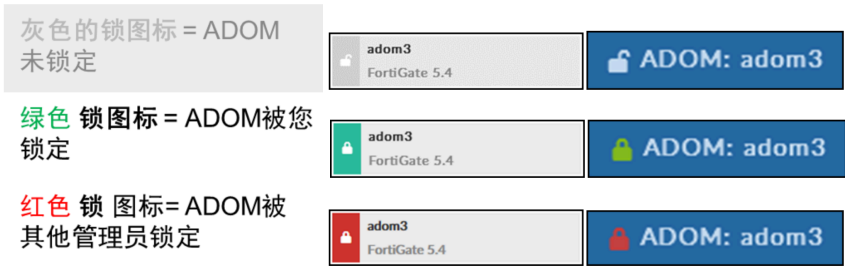
8. 新建完成后，注意点击页面左上角的“保存”按钮，配置才会被下发到启动配置文件中。



9. 此时 FortiOS_7_0 这个 ADOM 已经被 admin 账户锁定了，使用 user1 账户同时登陆后查看该 ADOM 的权限。可以看到由于 admin 已经锁定了该 ADOM，user1 无法针对该 ADOM 进行任何写操作。



.0. 针对 ADOM 锁定图标颜色的说明。



系统设置

- 证书管理
- Email 服务
- 页面替换
- 登录提示信息

证书管理

数字证书是一个经证书授权中心（CA）数字签名的、包含公钥拥有者信息以及公钥，并且在特定的时间段内有效的文件。

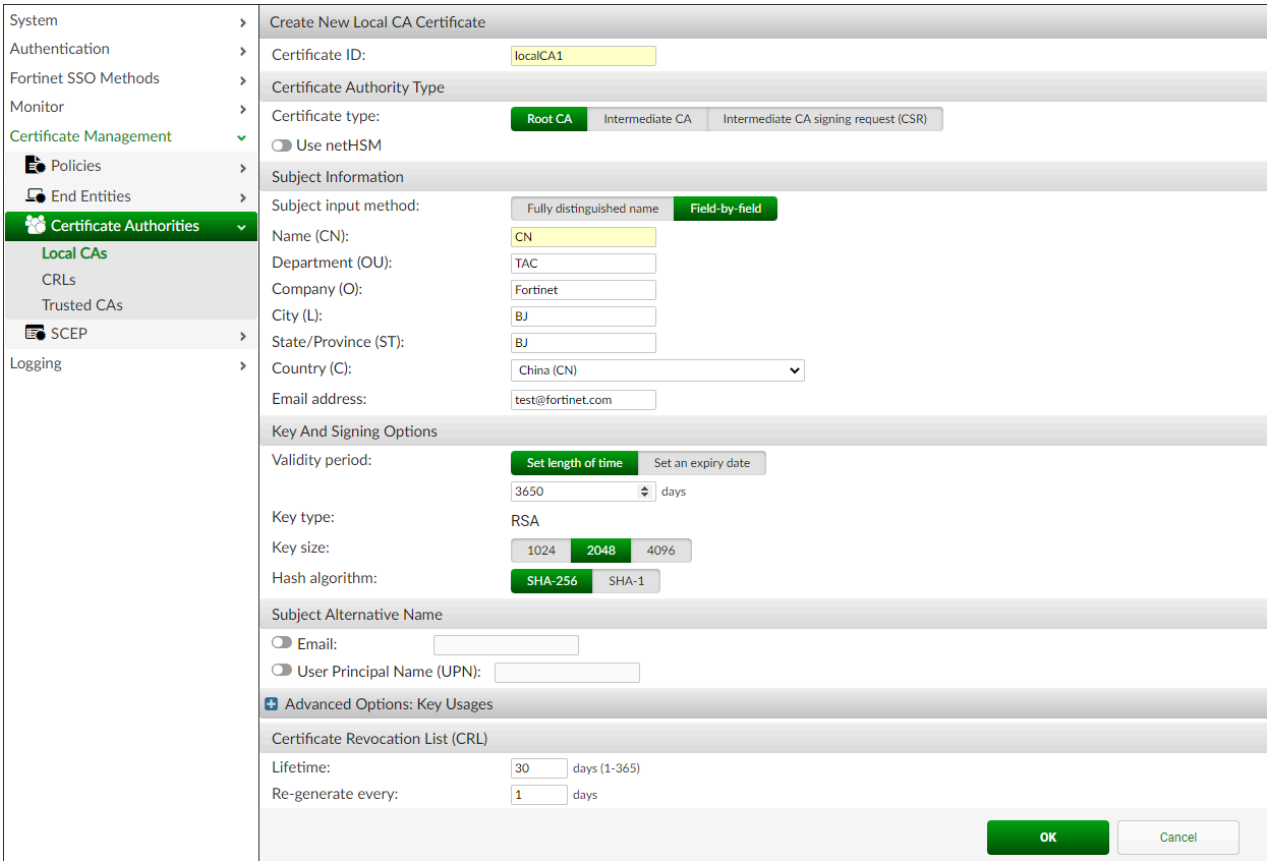
FortiGate 证书类型

这里使用 **FortiAuthenticator**作为证书服务器，生成证书并导入 FortiGate。

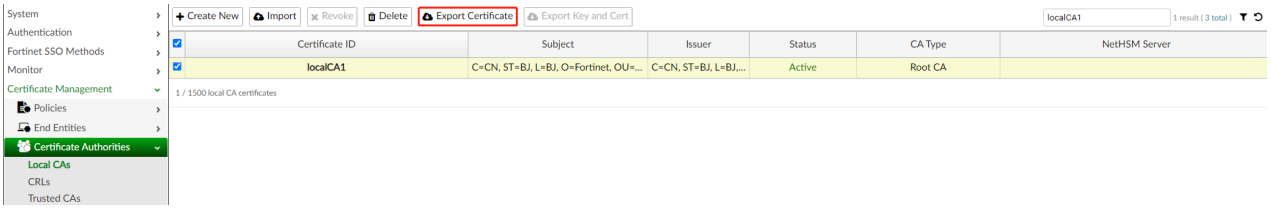
远端 CA 证书

远程 CA 证书是只有 CA 的公钥，用于验证证书的有效性，如 SSLVPN 中验证终端的证书。

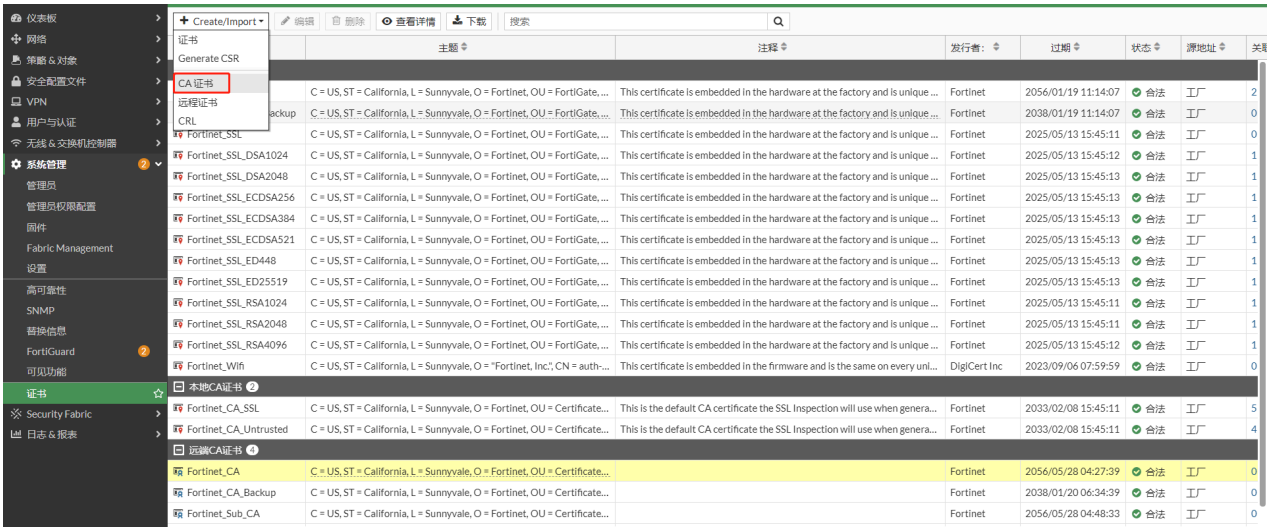
1. FortiAuthenticator 生成根 CA 证书。选择 Certicate Managerment → Certificate Authorities → Local CAs，点击 Create New，输入 CA 名称及证书信息，并点击“确认”。



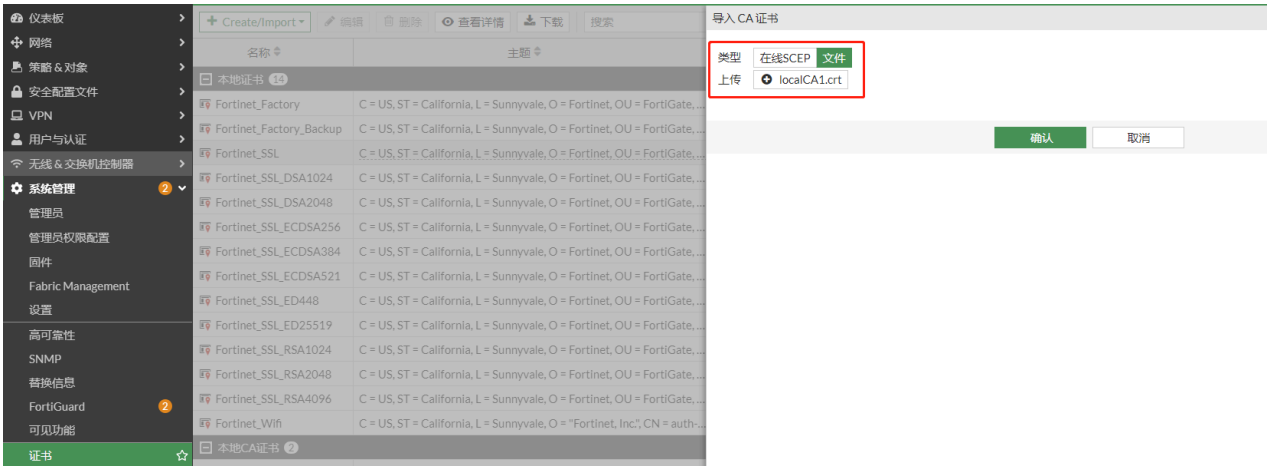
2. 导出 CA 证书，导出的文件为 localCA1.crt。



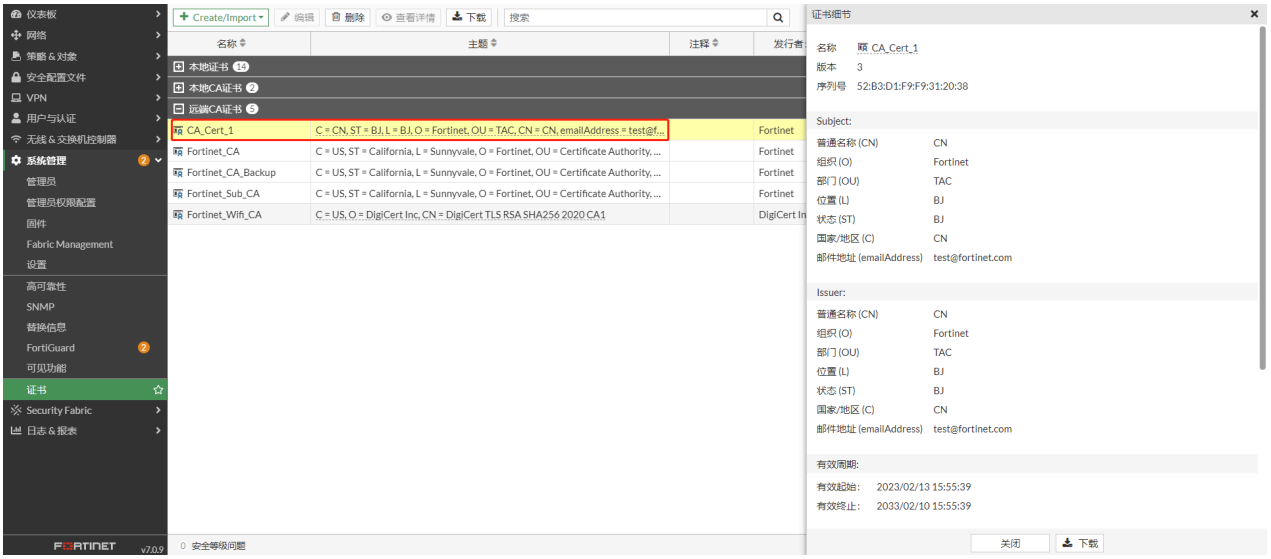
3. 导入 FortiGate。选择系统管理 → 证书，点击 Create/Import，选择 CA 证书。



4. 选择文件，上传 CA 证书文件 localCA1.crt。



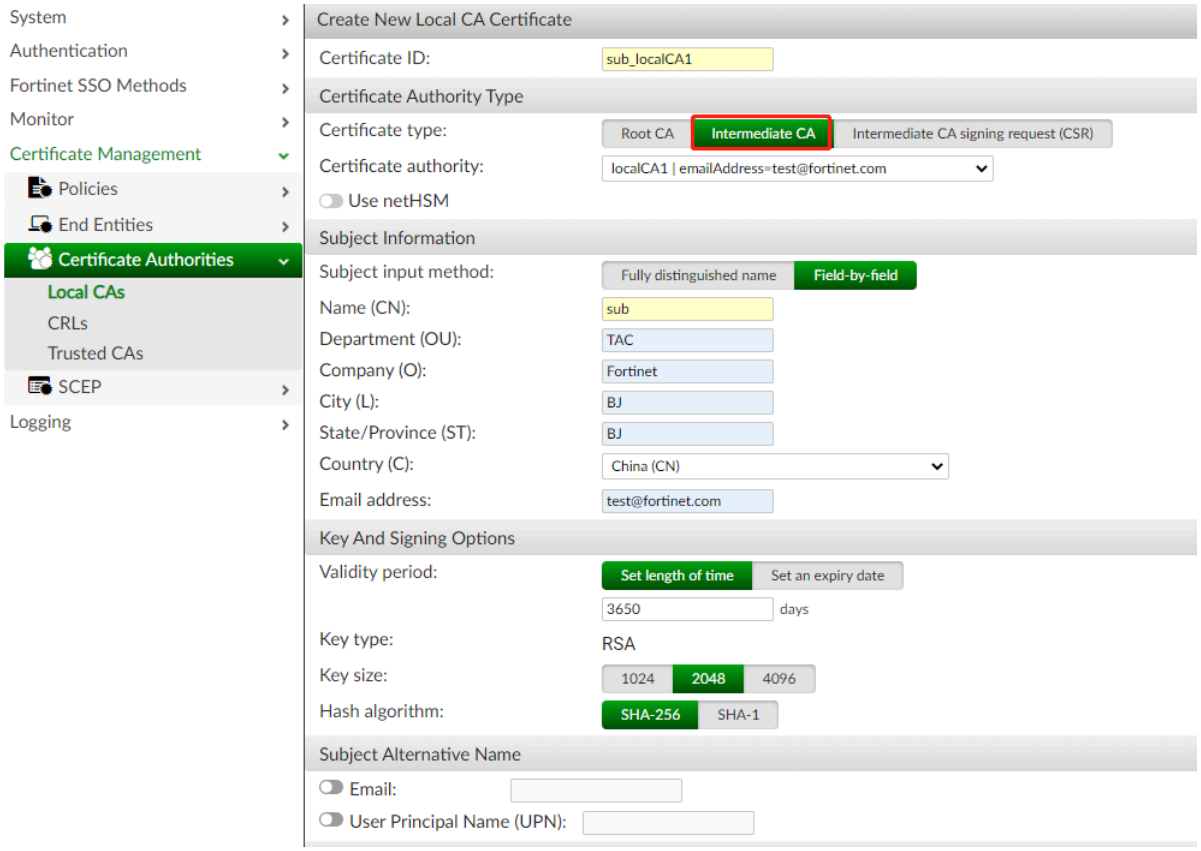
5. 查看导入的证书。



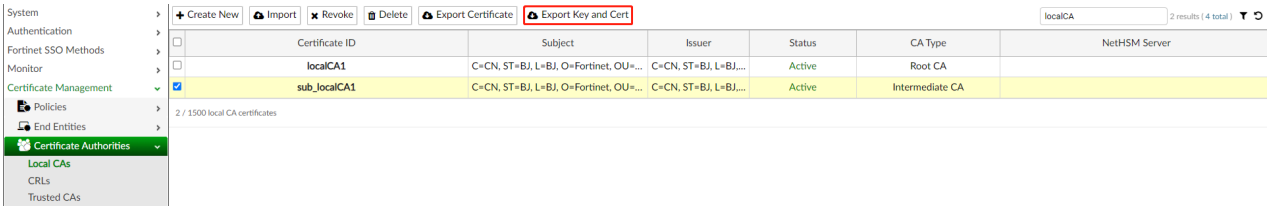
本地 CA 证书

本地 CA 证书是具有 CA 的公钥和私钥，具有签发能力的 CA 证书，如重签发 portal 认证的证书。

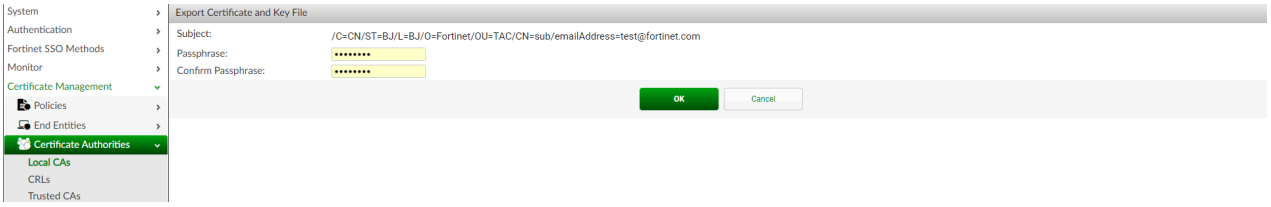
1. FortiAuthenticator 生成中间 CA 证书，选择“Certicate Managerment”-->” Certificate Authorities“-->“Local CAs”，点击“Create New”，输入 CA 名称及证书信息，证书类型选择“Intermediate CA“，指定签发的根 CA，并点击“确认”。



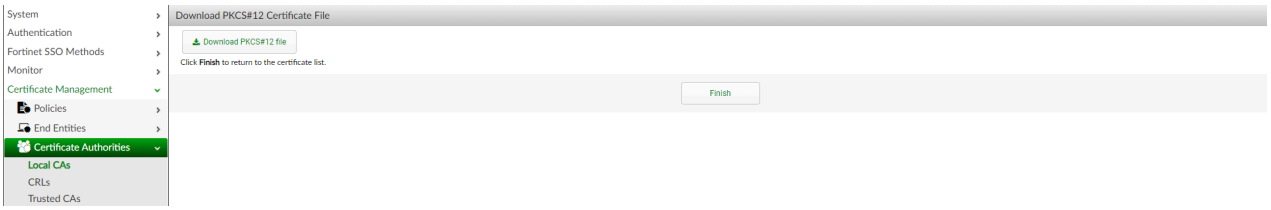
2. 导出 CA 公钥和私钥，点击” Export Key and Cert “。



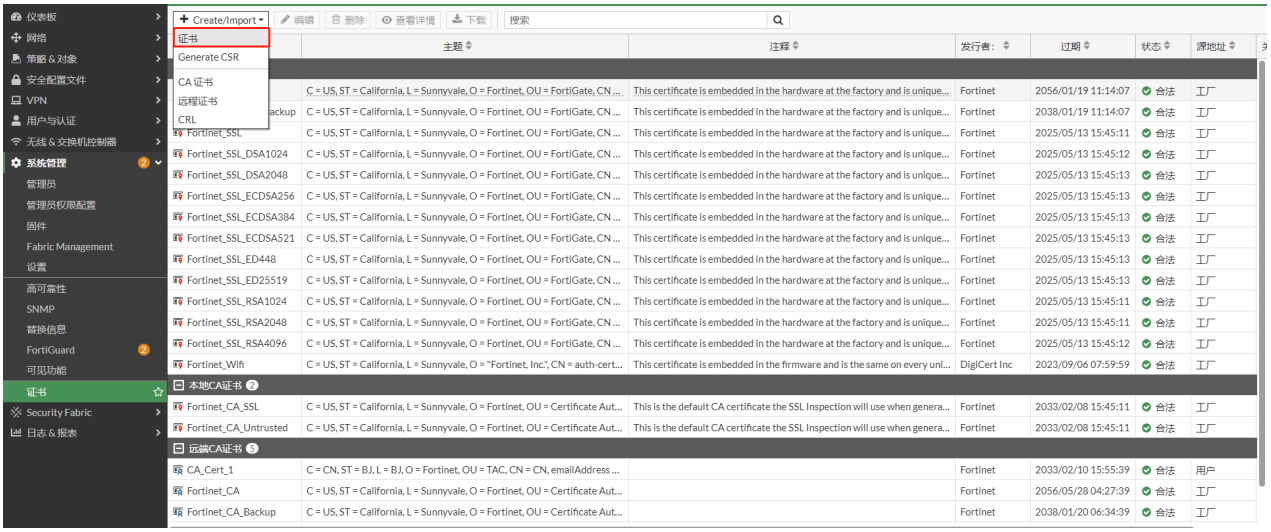
3. 输入用于保护证书的密码。



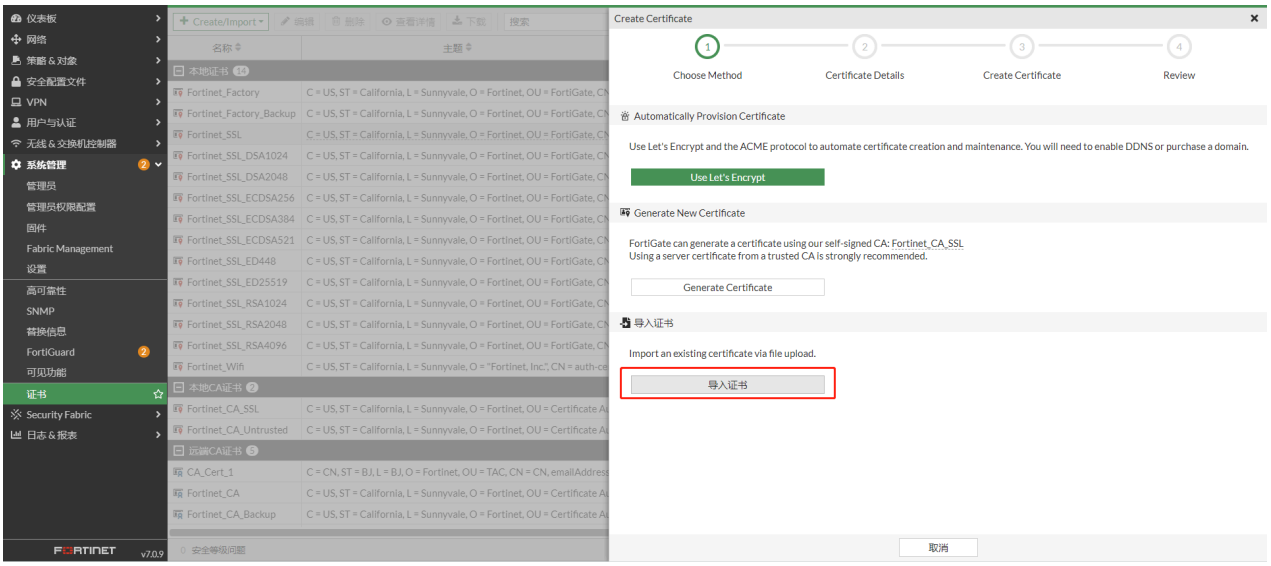
4. 下载证书，文件名是 sub_localCA1.p12。



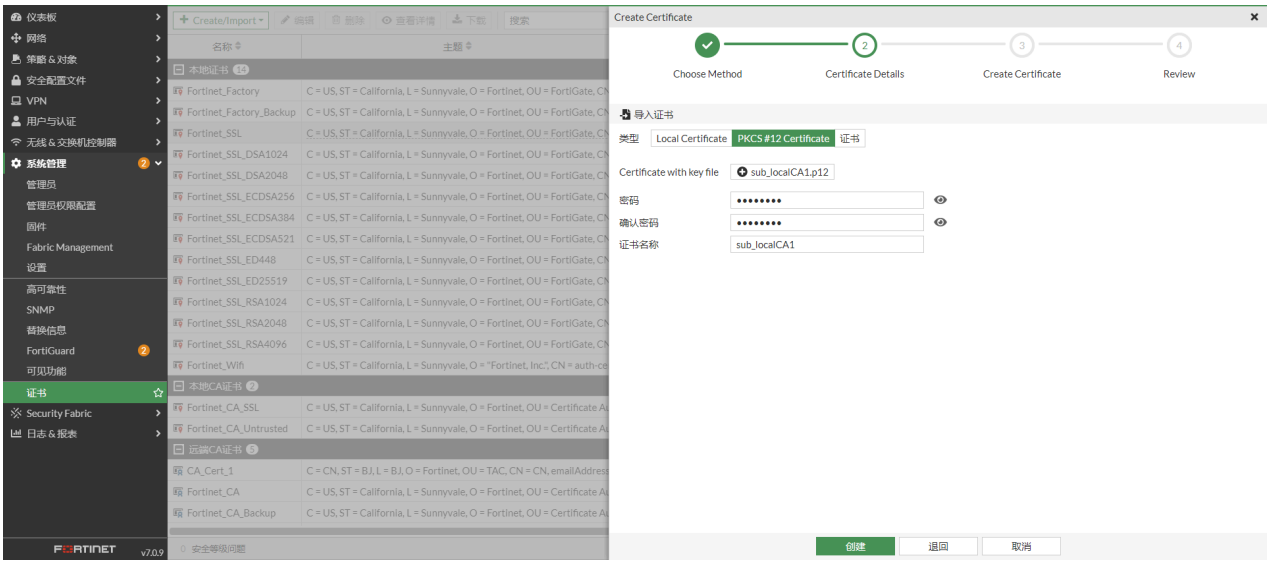
5. 导入 FortiGate，选择“系统管理”-->“证书”，点击“Create/Import”，选择“证书”。



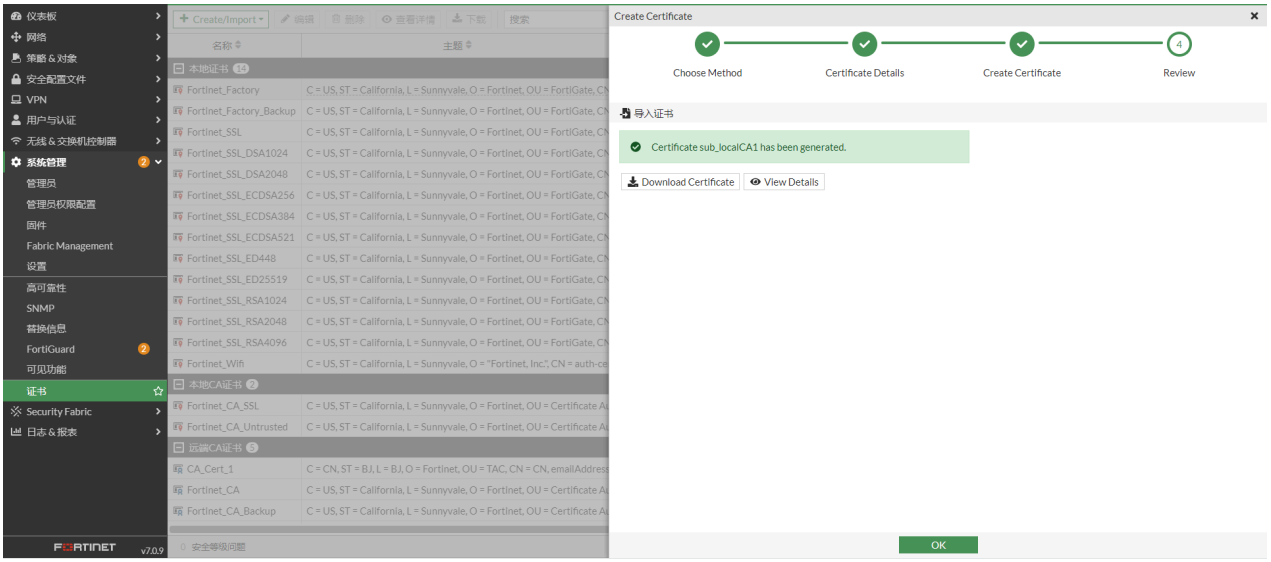
6. 点击”导入证书“。



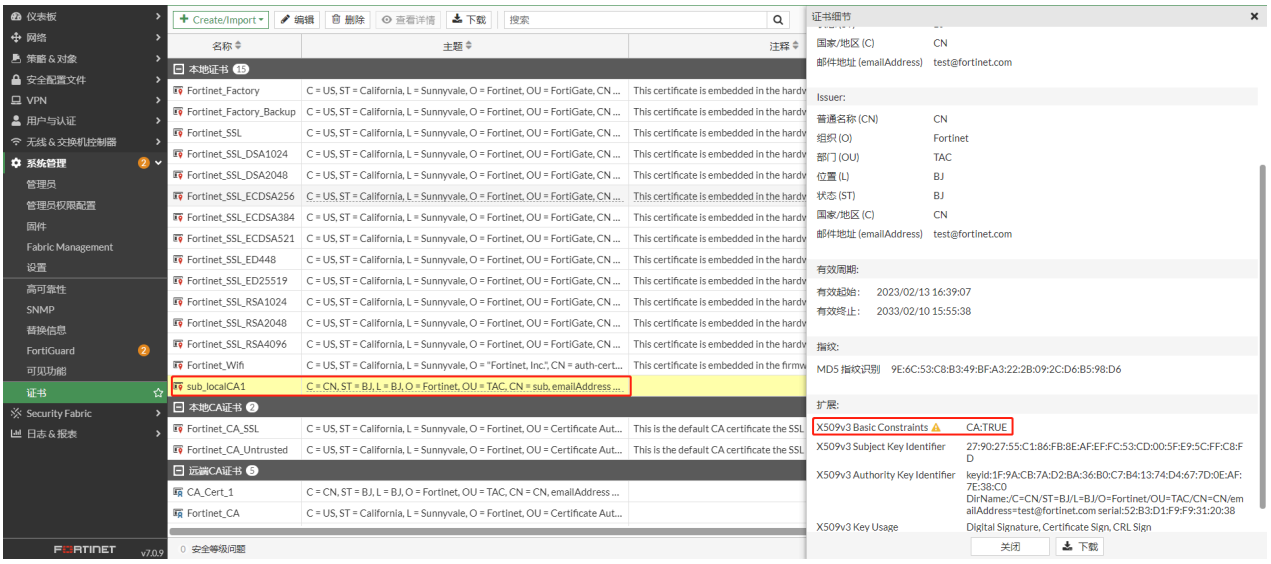
7. 选择” PKCS#12 Certificate“，输入证书的密码，点击”创建“。



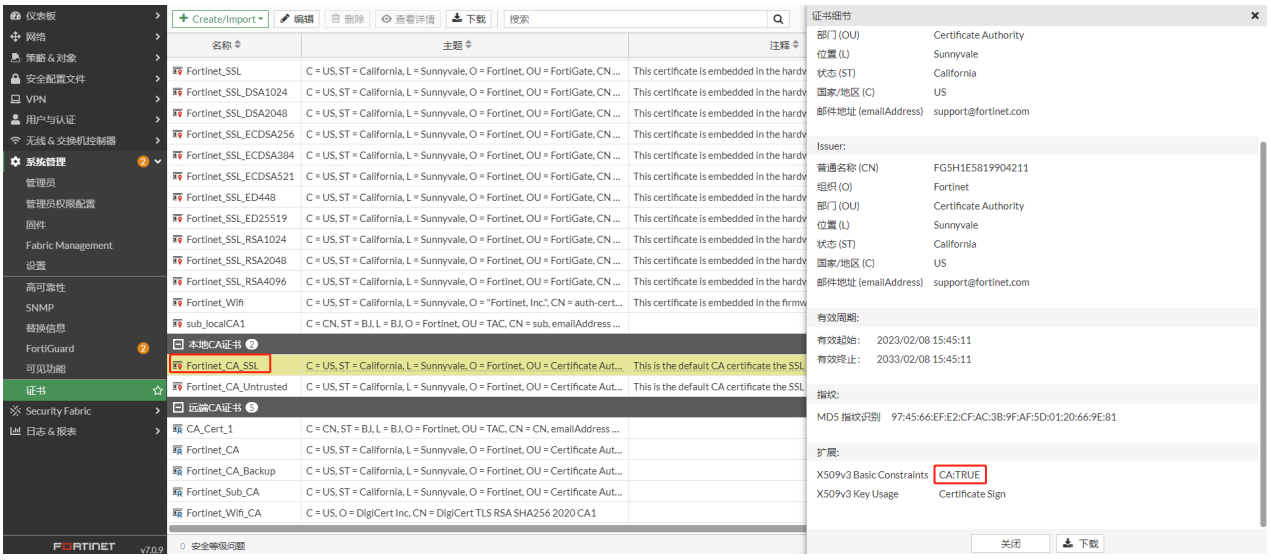
8. 导入成功。



9. 查看导入的证书。可以看到 sub_localCA1 的具有 CA:TRUE，是具有签发能力的 CA 证书。sub_localCA1 作为本地证书导入，位于本地证书菜单下，但和本地 CA 证书 Fortinet_CA_SSL 的作用是一样的。



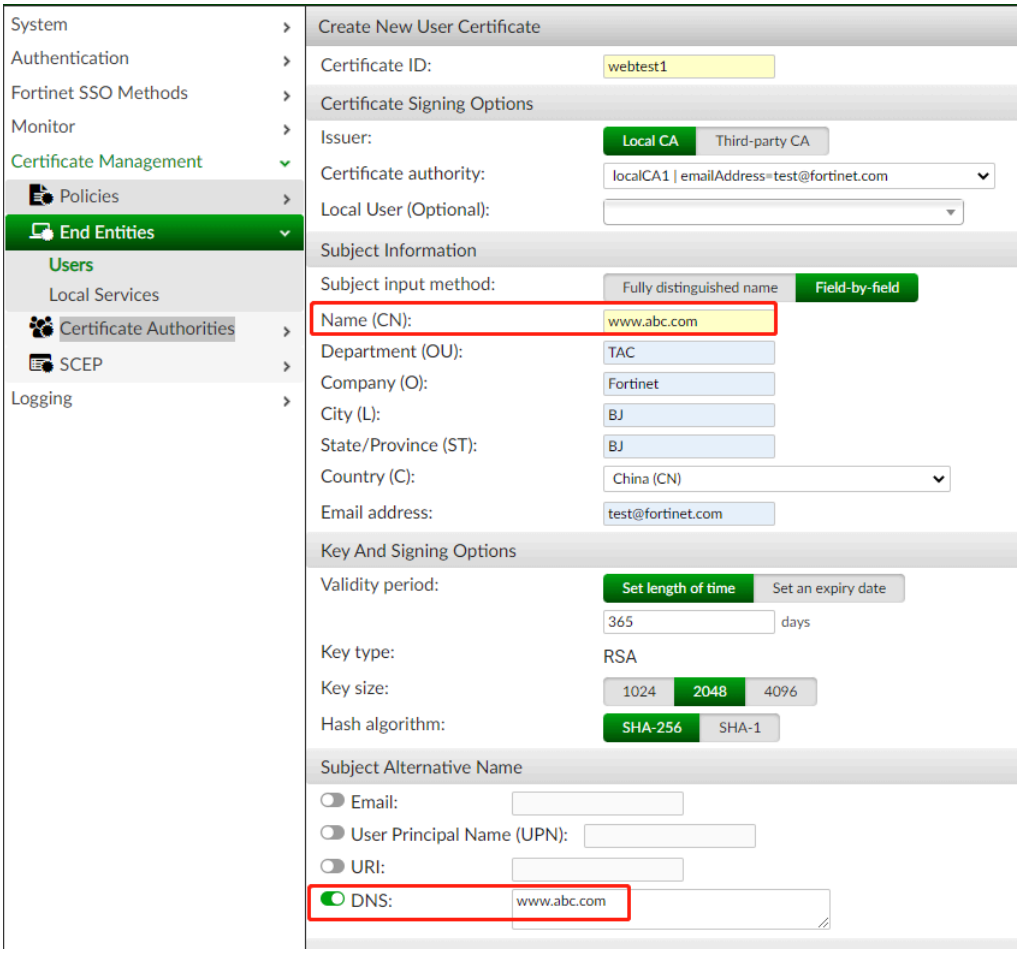
.0. Fortinet_CA_SSL 是具有签发能力的 CA 证书。



本地证书

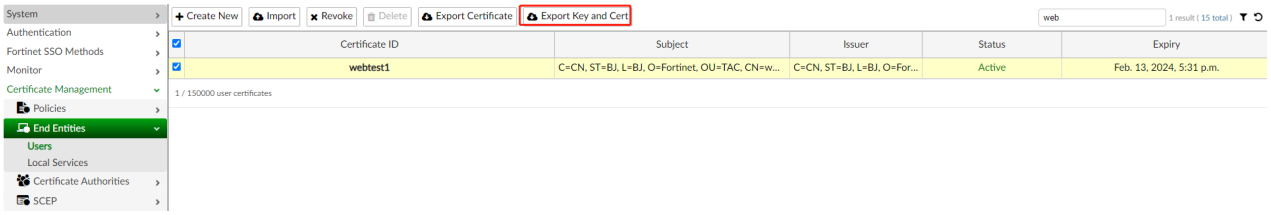
本地证书是具有公钥和私钥，为特定的服务颁发的证书，如 FortiGate GUI，SSLVPN，IPSEC VPN 证书。

1. FortiAuthenticator 生成用户证书，选择“Certicate Managerment”-->“End Entities”-->” Users“，点击“Create New”，输入证书名称及证书信息，指定签发的根 CA，点击“确认”。

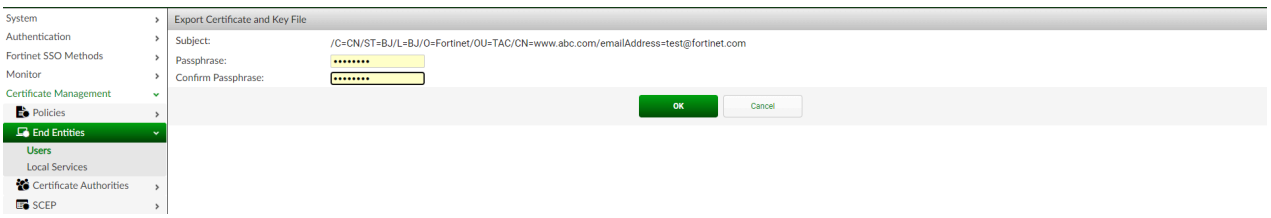


- CN：当浏览器尝试连接到远程服务器 (例如 HTTPS 服务器) 时，它将首先获取该服务器的 SSL 证书。 然后将要连接的域名与 SSL 证书中的 CN 进行比较。 如果它们相同，它将使用 SSL 证书来加密连接，否则将告警。
- SAN：单个 SSL 证书只能使用单个 Common Name，意味着 SSL 证书可用于单个主机名 + 域名。 为了解决此限制，创建了 Subject Alternative Name 。 SAN 用于在 SSL 证书中定义多名称或多通用名称。这里 SAN 和 CN 配置为一样即可。

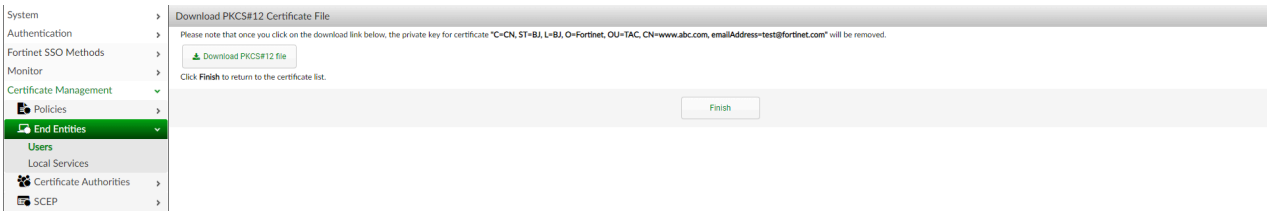
2. 导出用户证书公钥和私钥，点击” Export Key and Cert “。



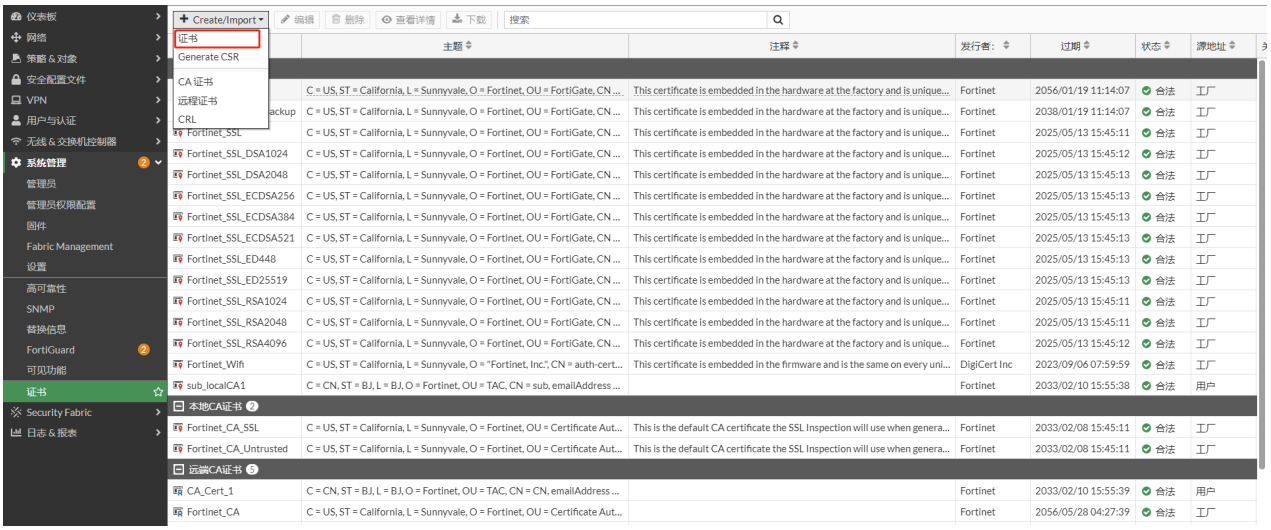
3. 输入证书保护的密码。



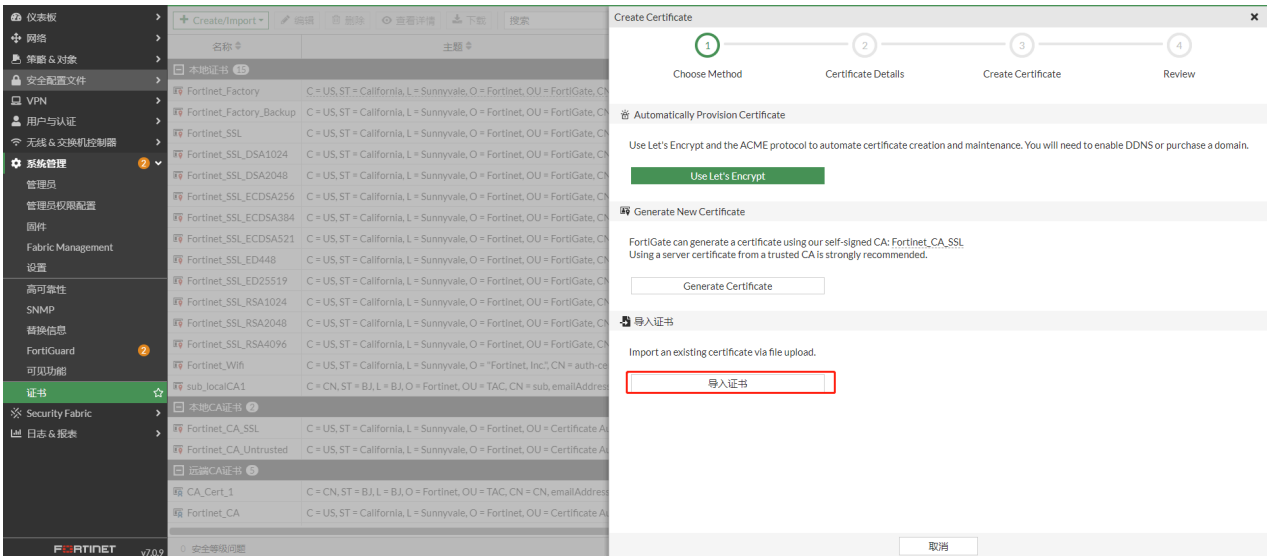
4. 导出证书，名称是 webtest1.p12。



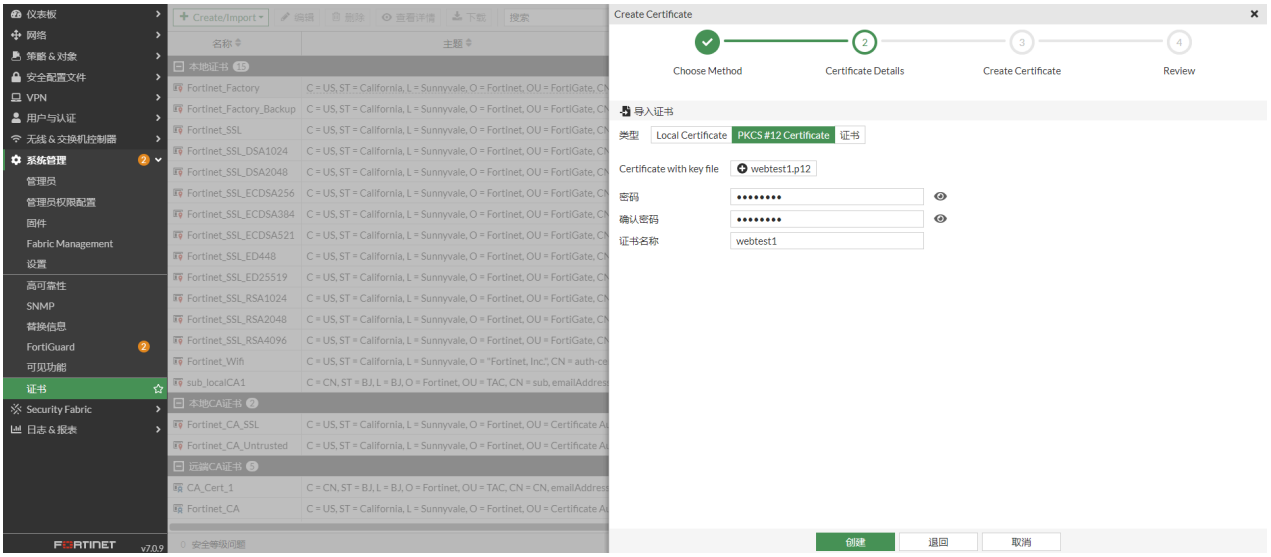
5. 导入 FortiGate，选择“系统管理”-->“证书”，点击“Create/Import”，选择“证书”。



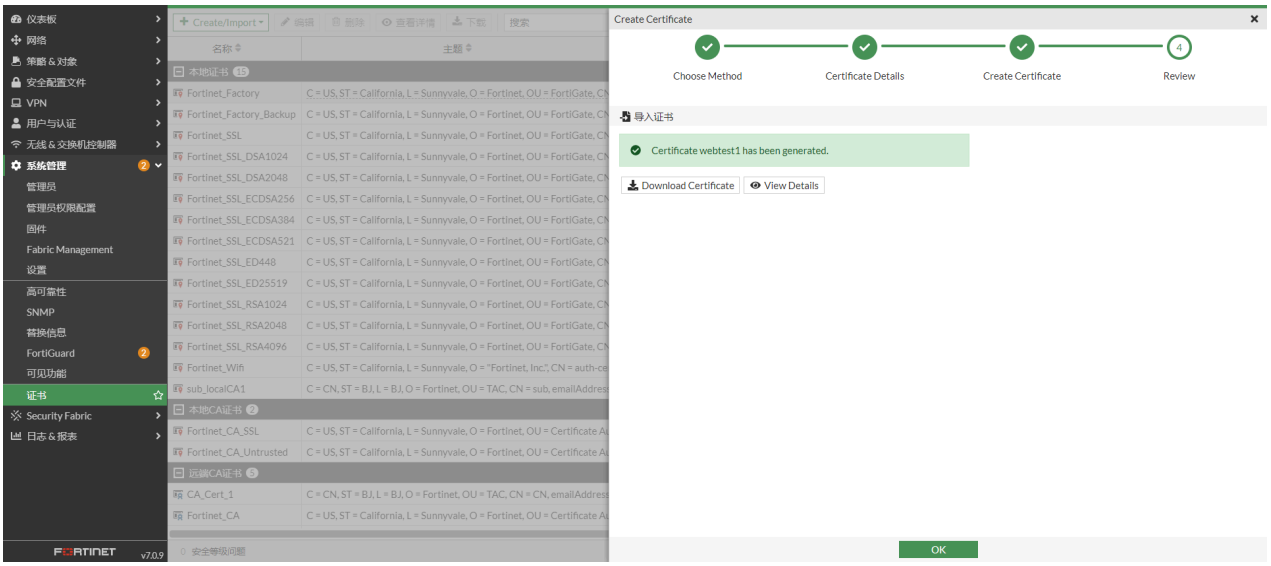
6. 选择”导入证书“。



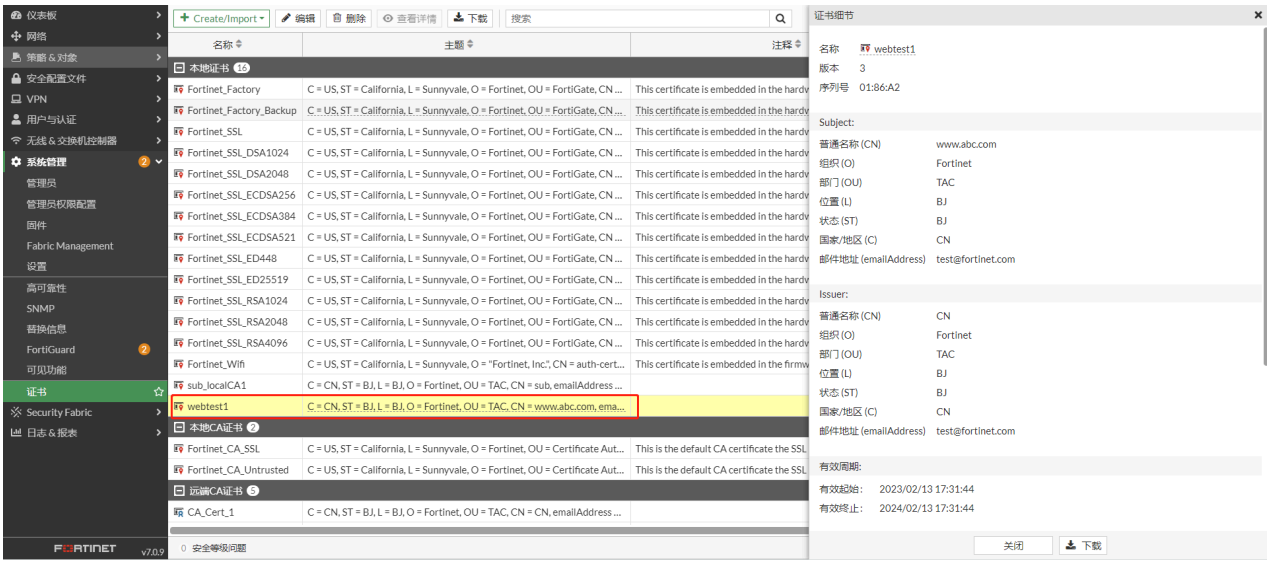
7. 选择“PKCS#12 Certificate”，输入证书的密码，点击“创建”。



8. 导入成功。



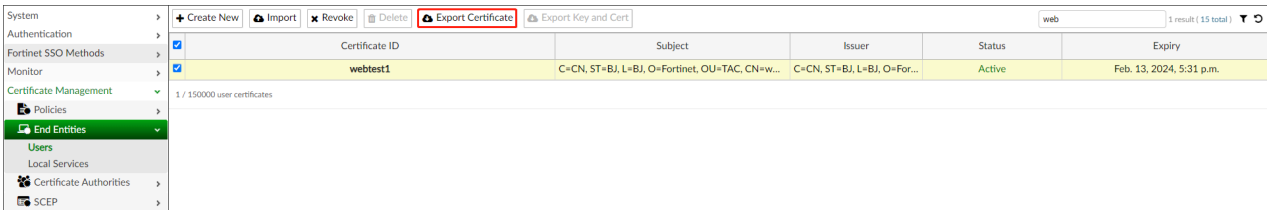
9. 查看证书。



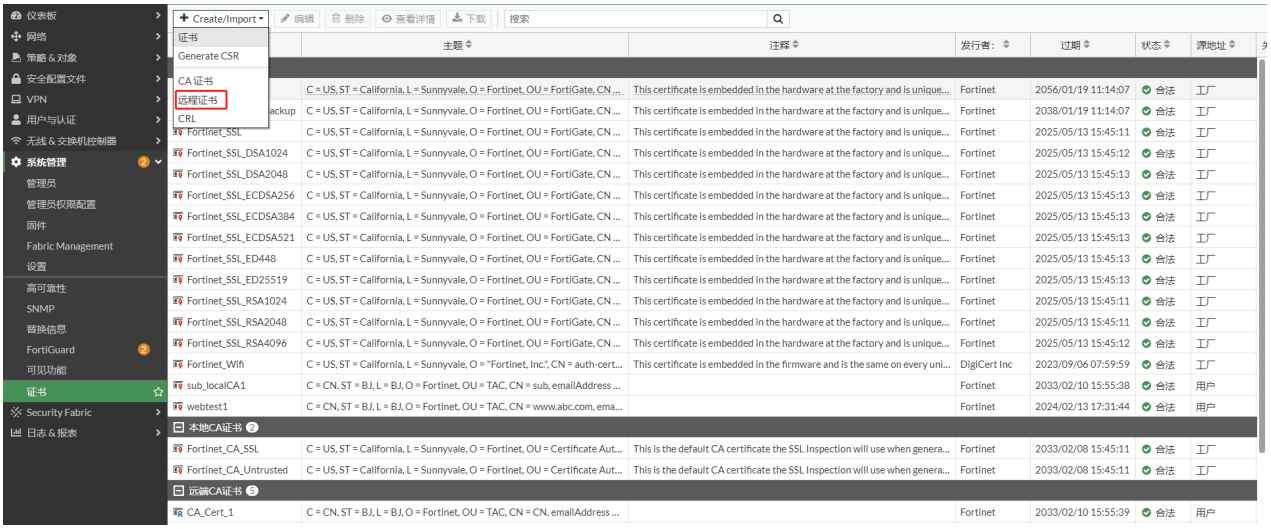
远程证书

远程是没有私钥的证书，用于验证私钥的签名，如 SAML 证书。

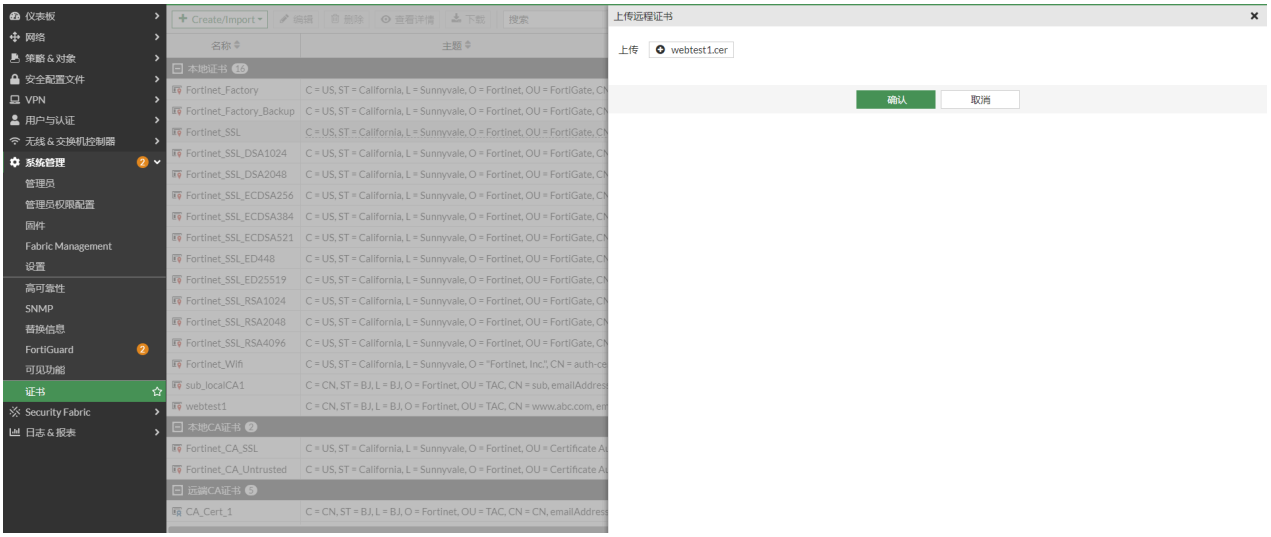
1. 导出证书。以 webtest1 证书为例。点击” Export Certificate“导出证书，文件名为 webtest1.cer。



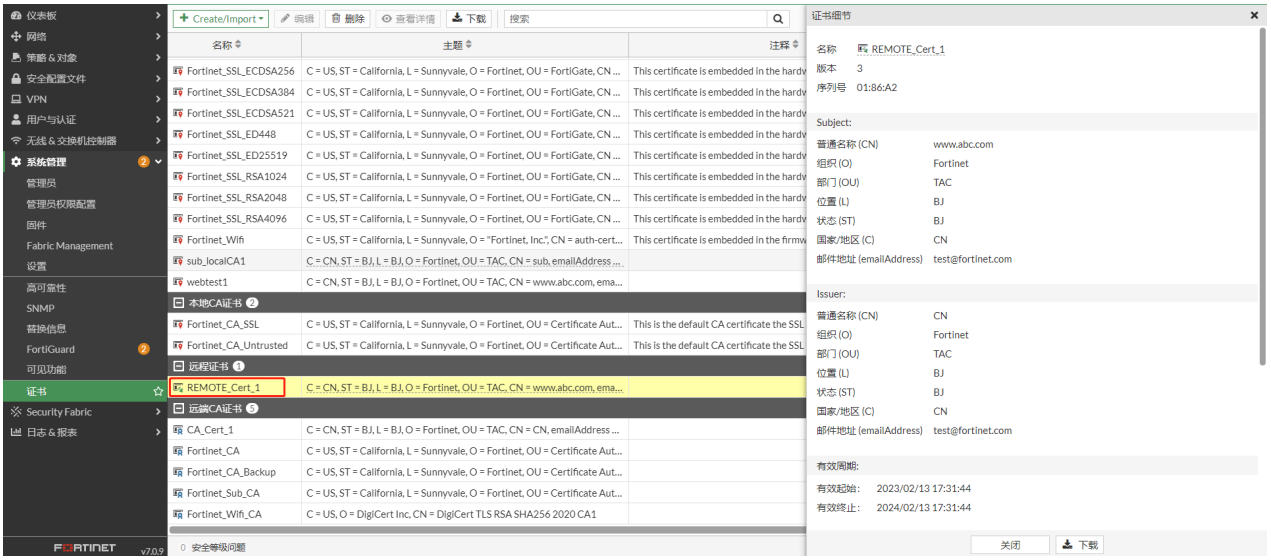
2. 导入 FortiGate。选择“系统管理”-->“证书”，点击“Create/Import”，选择“远程证书”。



3. 上传” webtest1.cer“。



4. 查看证书。


仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
管理员
管理员权限配置
固件
Fabric Management
设置
高可靠性
SNMP
替换信息
FortiGuard
可见功能
证书
Security Fabric
日志 & 报表
FORTINET
v7.0.9
Create/Import
编辑
删除
查看详情
下载
搜索
名称
主题
注释
Fortinet_SSL_ECDSA256
Fortinet_SSL_ECDSA384
Fortinet_SSL_ECDSA521
Fortinet_SSL_ED448
Fortinet_SSL_ED25519
Fortinet_SSL_RSA1024
Fortinet_SSL_RSA2048
Fortinet_SSL_RSA4096
Fortinet_Wifi
sub_localCA1
webtest1
本地CA证书 2
Fortinet_CA_SSL
Fortinet_CA_Untrusted
远程证书 1
REMOTE_Cert_1
远端CA证书 5
CA_Cert_1
Fortinet_CA
Fortinet_CA_Backup
Fortinet_Sub_CA
Fortinet_Wifi_CA
0 安全等级问题
证书细节
名称 REMOTE_Cert_1
版本 3
序列号 01:86:A2
Subject:
普通名称 (CN) www.abc.com
组织 (O) Fortinet
部门 (OU) TAC
位置 (L) BJ
状态 (ST) BJ
国家/地区 (C) CN
邮件地址 (emailAddress) test@fortinet.com
Issuer:
普通名称 (CN) CN
组织 (O) Fortinet
部门 (OU) TAC
位置 (L) BJ
状态 (ST) BJ
国家/地区 (C) CN
邮件地址 (emailAddress) test@fortinet.com
有效周期:
有效起始: 2023/02/13 17:31:44
有效终止: 2024/02/13 17:31:44
关闭
下载

Email 服务

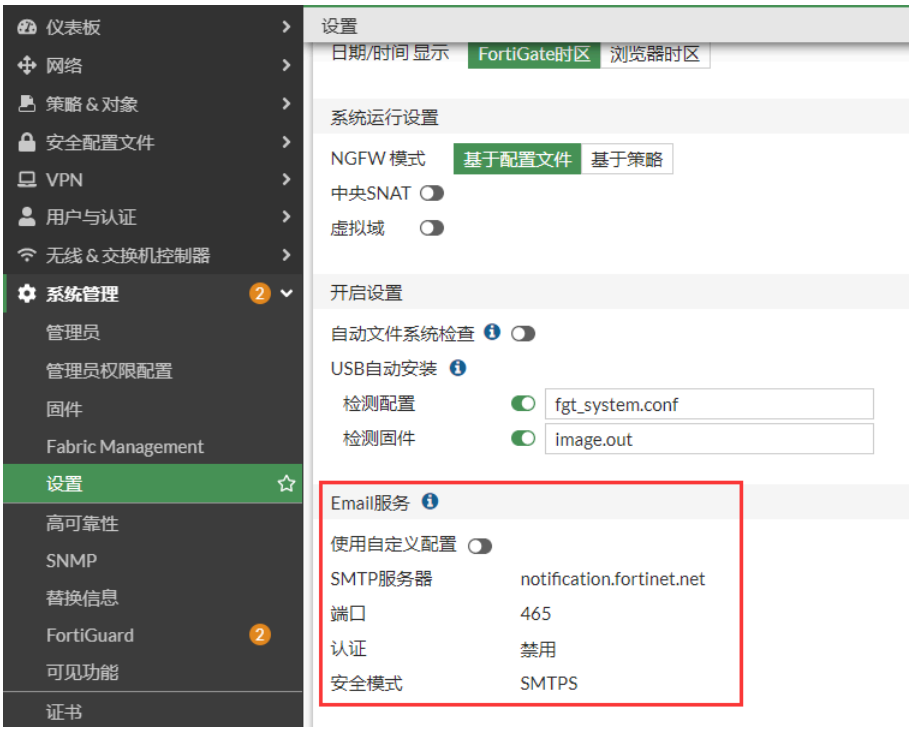
简介

Email 服务用于 FortiGate 发送的电子邮件，包括警报电子邮件，FortiToken 激活以及 automation stitch 邮件。FortiGate 有一个默认的 SMTP 服务器 notifation.fortinet.net，用于提供安全的 SMTP 邮件服务，也支持自定义的邮件服务。

默认的 SMTP 服务器

FortiGate 默认的 SMTP 服务器配置如下图，无需修改，确保 FortiGate 能够连接 STMP 服务器，即 FortiGate 能够上网。

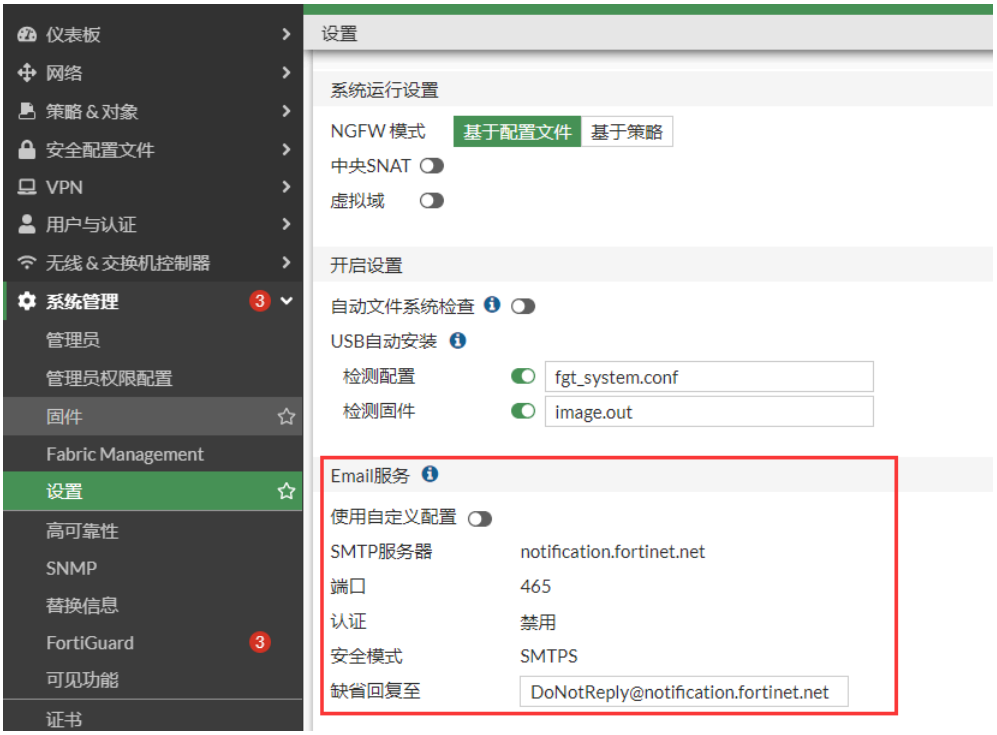
在 6.4.12，7.0.10，7.2.4 中，FortiGate 发件箱地址是 DoNotReply@notification.fortinet.net，“缺省回复至”选项已移除，该选项与发件箱一致，这样是为了避免被对端邮件服务器当成垃圾邮件给丢弃。



邮件发送的明细如下：

```
Date: Fri, 2 Dec 2022 11:35:33 +0800
From: DoNotReply@notification.fortinet.net
Message-Id: <1669952133.28@notification.fortinet.net>
Reply-To: "DoNotReply" <DoNotReply@notification.fortinet.net>
To: xxxxxx@test.com
Subject: Test
```

在之前的版本中，“缺省回复至”选项需要单独填写，请设置为“DoNotReply@notification.fortinet.net”。



自定义的 STMP 服务器

163 邮件服务器

1. 在 163 邮件中，开启 POP3/SMTP 服务，点击“开启”。



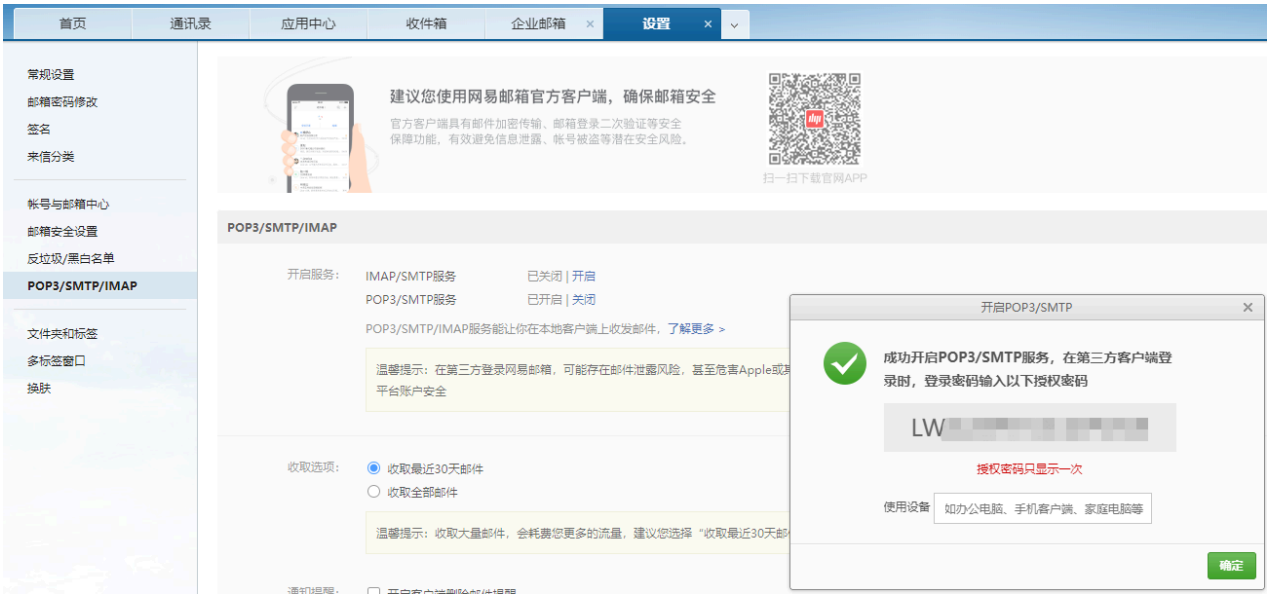
在弹出框中点击“继续开启”。



手机微信扫码发送短信，当发完后，点击“我已发送”。



然后就可以获取到“授权码”。

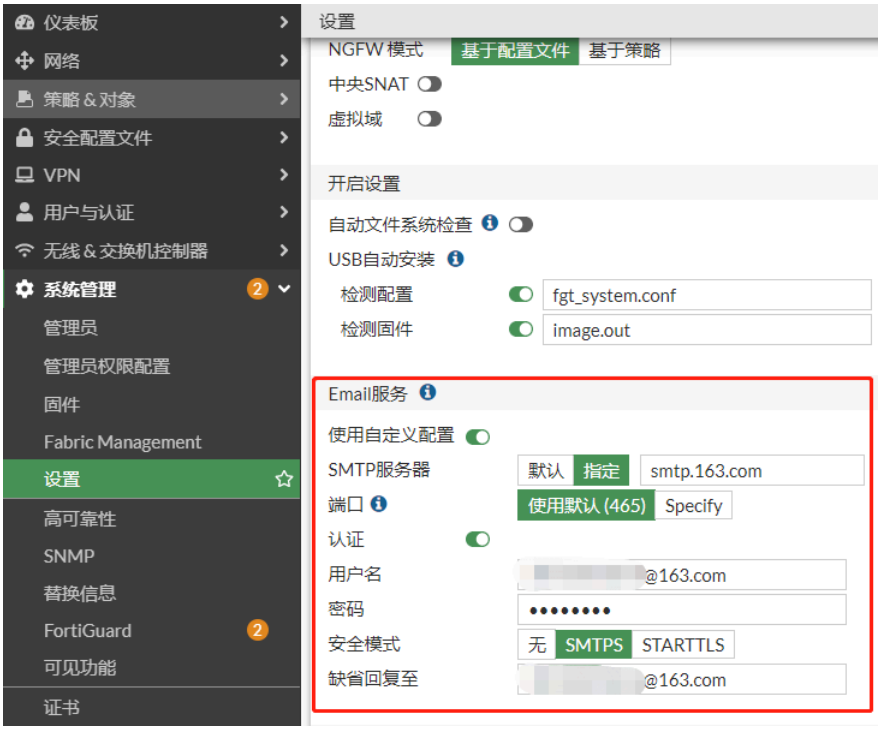


POP3/SMTP 服务开启成功。



2. Email 服务设置

163 的 SMTP 服务器是 smtp.163.com ↗，用户名是 163 的邮箱账号，密码是授权码（不是 163 邮箱密码），安全模式为 SMTPS，缺省回复至是 163 的邮箱账号。



QQ 邮件服务器

1. 在 QQ 邮箱中，点击“账户”，开启“POP3/SMTP 服务”

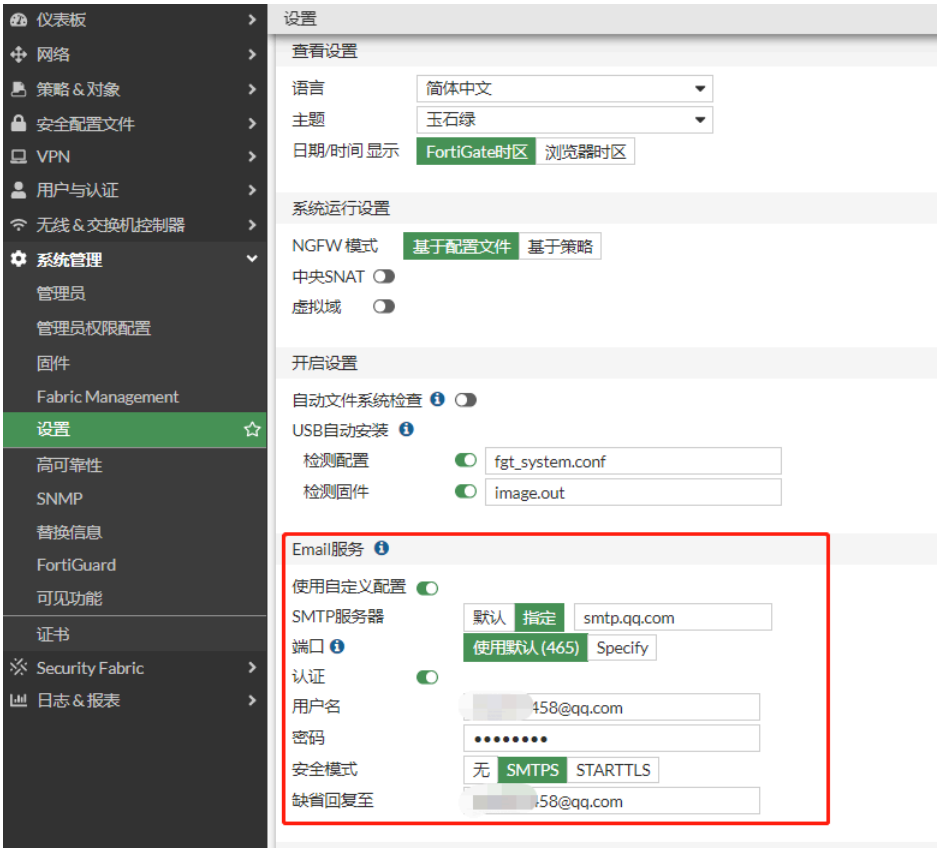


然后点击“生成授权码”，按照提示的步骤拿到授权码。



2. Email 服务设置

   QQ 的 SMTP 服务器是 smtp.qq.com ↗，用户名是 qq 的邮箱账号，密码是授权码（不是 qq 密码），安全模式为 SMTPS，缺省回复至是 qq 的邮箱账号。

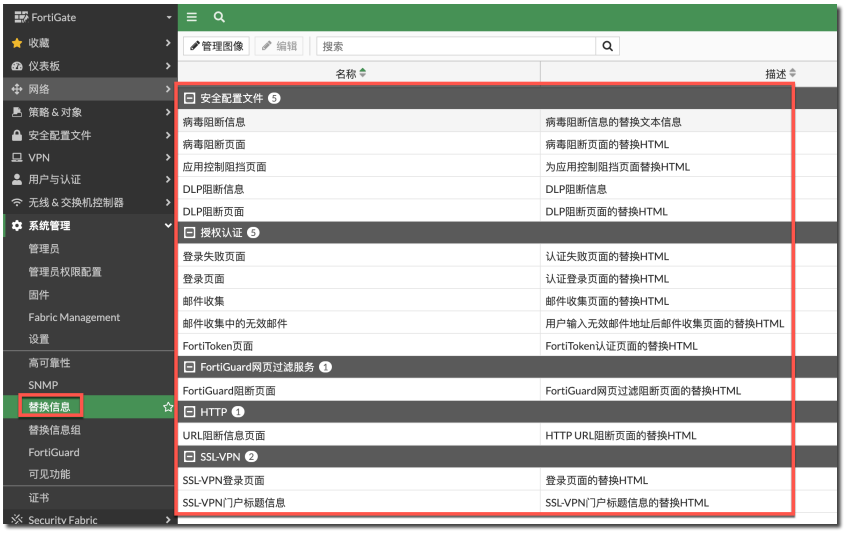

仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
管理员
管理员权限配置
固件
Fabric Management
设置
高可靠性
SNMP
替换信息
FortiGuard
可见功能
证书
Security Fabric
日志 & 报表
设置
查看设置
语言
简体中文
主题
玉石绿
日期/时间 显示
FortiGate时区
浏览器时区
系统运行设置
NGFW 模式
基于配置文件
基于策略
中央SNAT
虚拟域
开启设置
自动文件系统检查
USB自动安装
检测配置
fgt_system.conf
检测固件
image.out
Email服务
使用自定义配置
SMTP服务器
默认
指定
smtp.qq.com
端口
使用默认 (465)
Specify
认证
用户名
458@qq.com
密码
••••••••
安全模式
无
SMTPS
STARTTLS
缺省回复至
58@qq.com

页面替换

功能介绍

FortiOS 可以自定义部分用户页面样式，以满足用户的个性化需求。可以修改内容的内容包括 HTML、文本信息和图片 Logo。

可替换资源的页面如下：



修改替换信息

> **重要**
>
> “替换消息”页面有两个视图。简单查看视图（默认）显示最常用的替换消息。扩展查看视图显示所有替换消息类别。

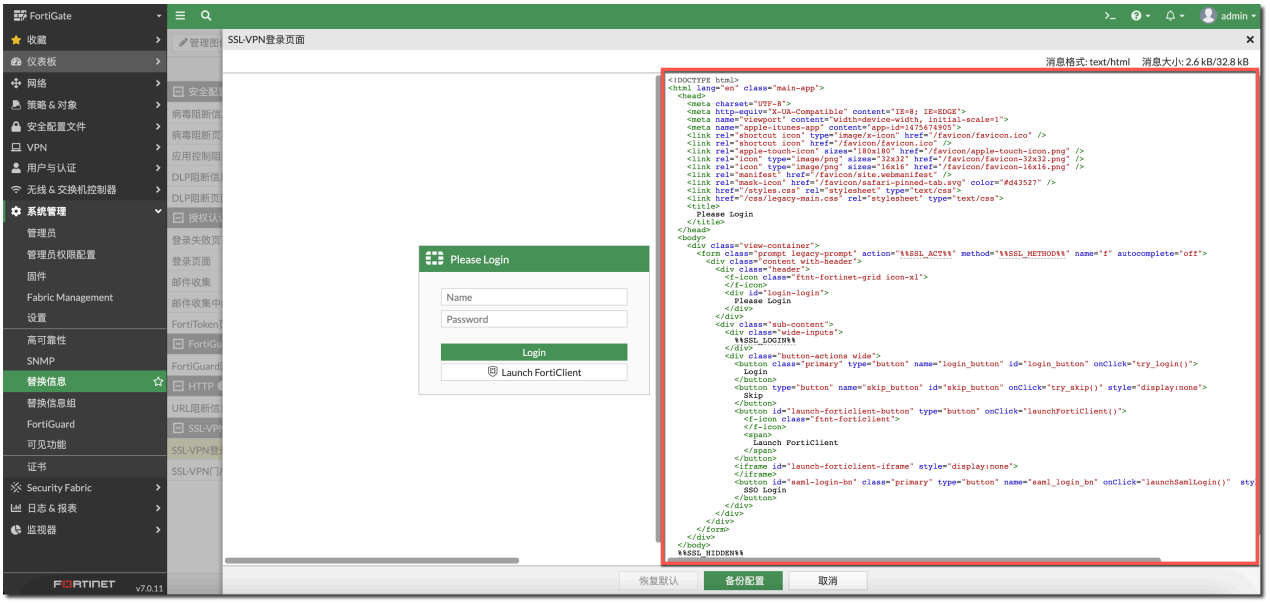
在 GUI 修改

1. 进入“系统管理 → 替换信息”页面， 编辑某个替换信息。

> **重要**
>
> 如果没有找到要编辑的替换消息条目，请单击右上角的扩展查看视图。

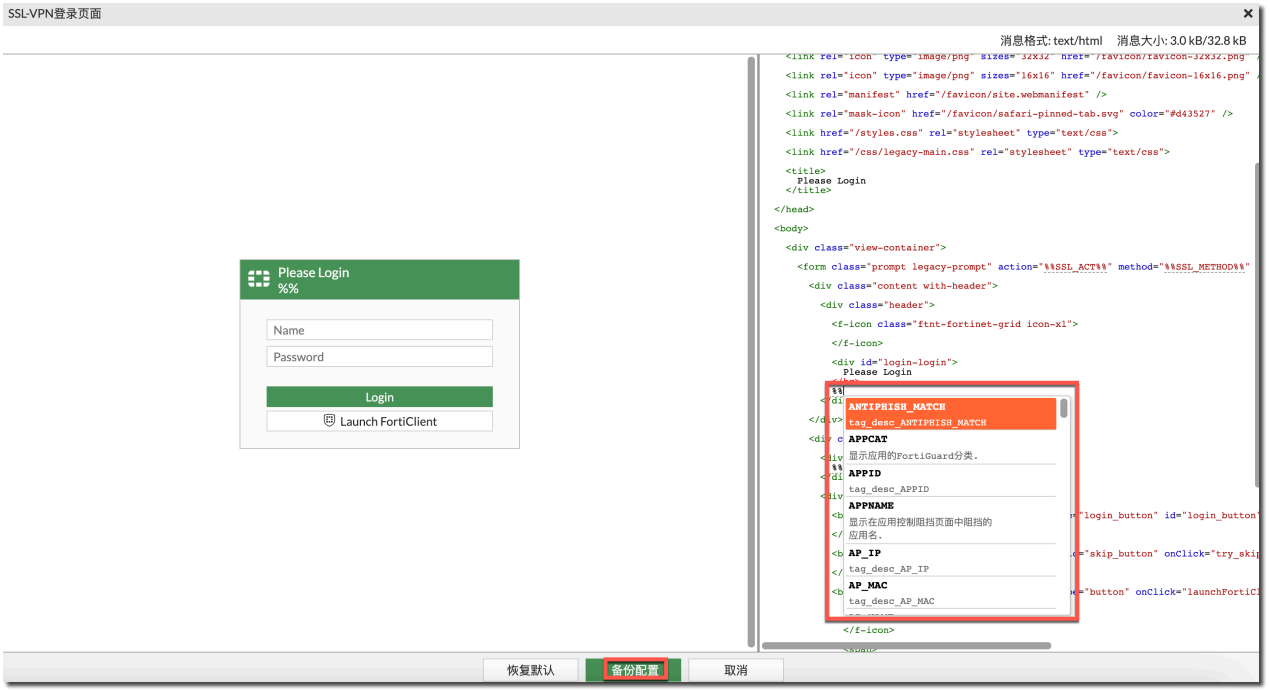


2. 在右侧修改 HTML 内容，左侧可以实时预览修改结果。



3. 需要将变量添加到代码时，右键单击并选择插入标记，或键入 %% 查看可用变量列表，也可以键入变量名称然后按 Enter 或 Tab 键自动完成变量名称，修改完成后，点击“备份配置”按钮下发配置。

> **重要**
>
> 点击“恢复默认”按钮可以将 HTML 恢复为出厂状态。



CLI 替换

例如想要修改“流量限制超限页面”（限速超速）的 HTML 内容，使用如下命令：

```
config system replacemsg traffic-quota "per-ip-shaper-block"
    set buffer "<html>
  <head>
    <title>
      Traffic Quota Control
    </title>
  </head>
  <body>
    <font size=2>
```

```
      <table width=\"100%\">
        <tr>
          <td bgcolor=#3300cc align=\"center\" colspan=2>
            <font color=#ffffff>
              <b>
                Traffic blocked because exceeded session quota
              </b>
            </font>
          </td>
        </tr>
      </table>
      <br>
      <br>
      Traffic blocked because it exceeded the per IP shaper session quota. Please contact the system administrator.
      <br>
      %%QUOTA_INFO%%
      <br>
      <br>
      <hr>
    </font>
  </body>
</html>"
    set header http
    set format html
end
```

修改替换图片

可以替换以下位置的图片：

- 免责声明页面（Disclaimer page）
- 拒绝免责声明页面（Declined disclaimer page）
- 登录页面（Login page）
- 登录失败页面（Login failed page）
- 登录要求页面（Login Challenge Page）
- 保持存活页面（Keepalive page）

GUI 替换

1. 进入“系统管理 → 替换信息”页面，点击“管理图像”按钮。

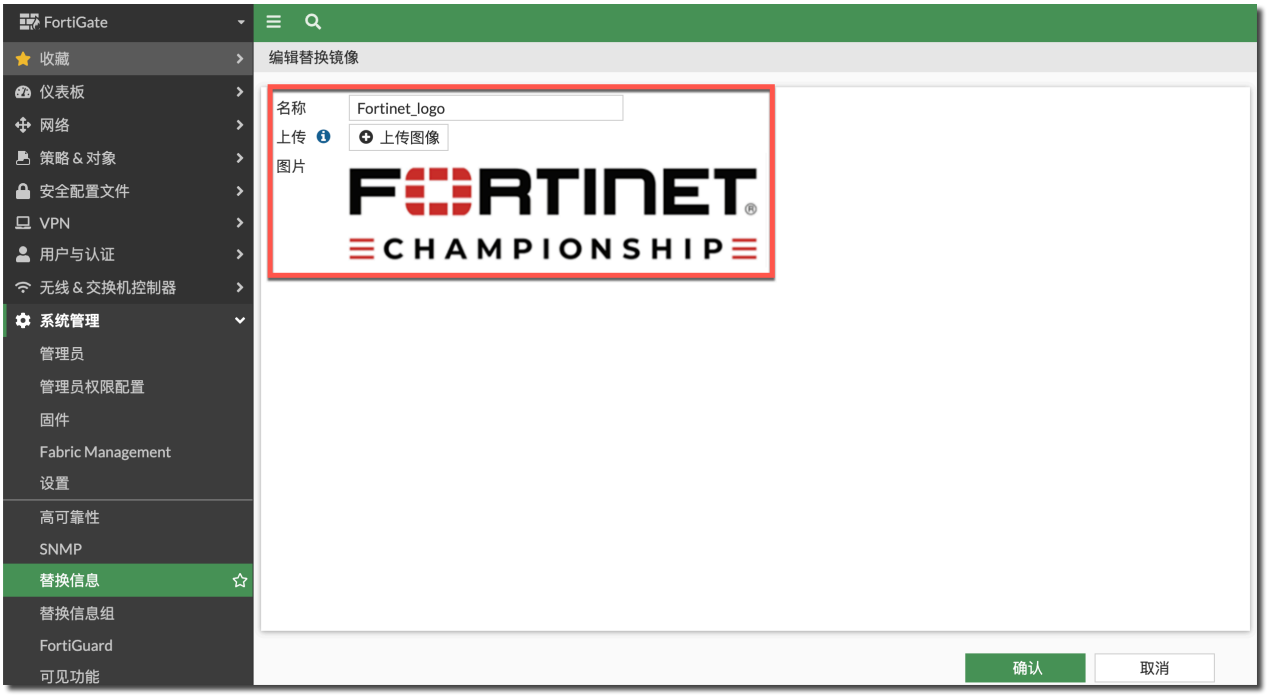


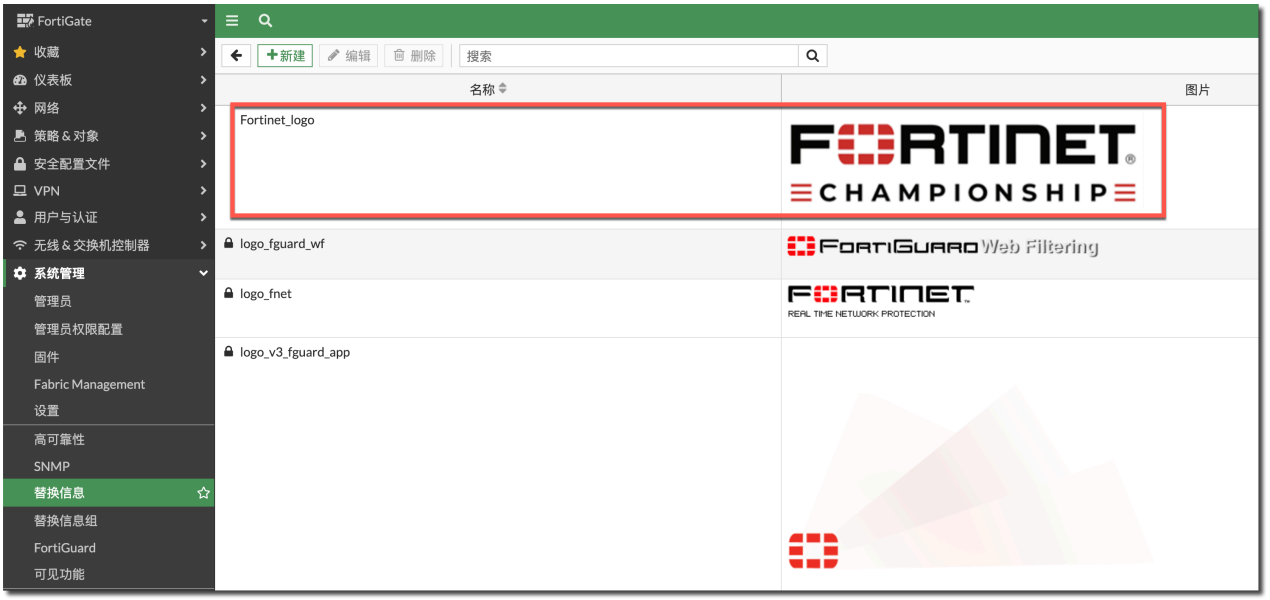
2. 在弹出的图片管理中，点击新建按钮。



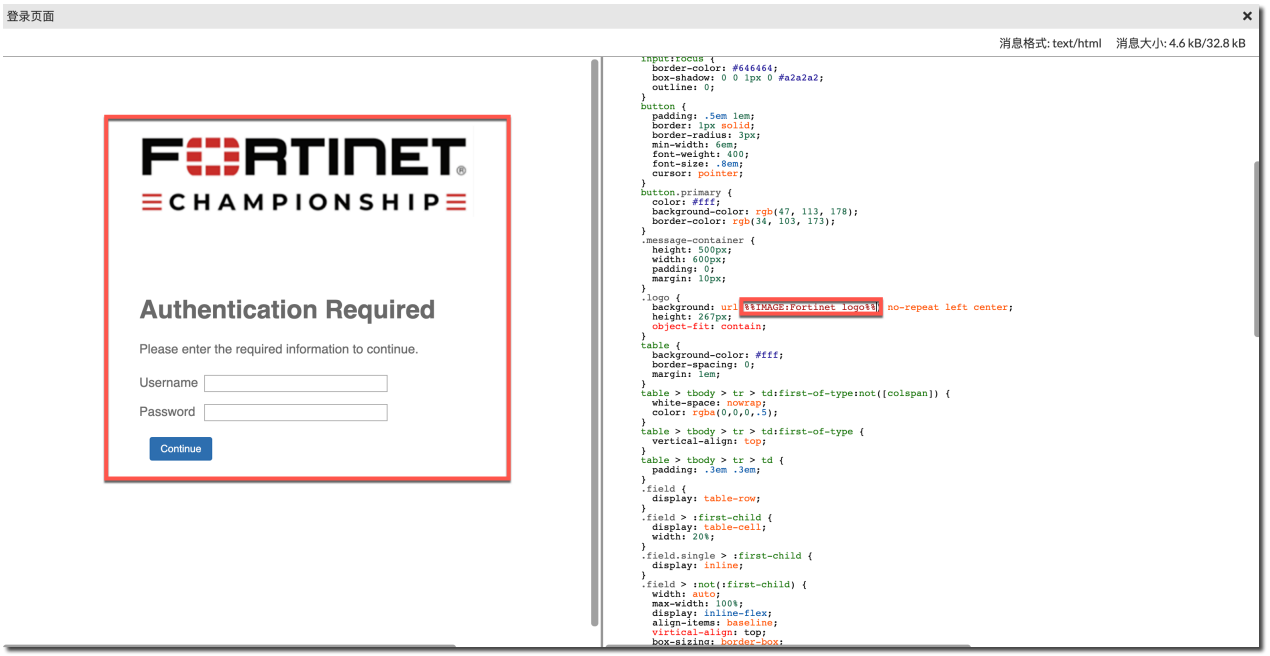
3. 配置图片名称，并上传图片。

> **重要**
>
> 支持的图像格式为 GIF、JPEG、TIFF 和 PNG，支持的最大文件大小为 24 KB。





4. 进入“系统管理 → 替换信息”页面，编辑要替换图片的替换信息，右键单击要添加图像的消息代码，然后选择“插入图像”，选择上步上传的图片，并双击选择，左侧的预览可以看到替换的 Logo。



CLI 替换

使用图片的 Base64 编码通过 CLI 替换图片，图片的 Base64 信息可以通过在线转换方式获取（如 https://tool.chinaz.com/tools/imgtobase ↗ ）。

```
config system replacemsg-image
    edit <image_name>
        set image-type {gif | jpg | tiff | png}
        set image-base64 <string>
    next
end
```

替换信息组

使用上述替换信息可以对所有的相关页面资源进行内容替换，但无法基于每个认证策略、UTM profile、自动化邮件等配置进行个性化定制，而替换信息组可以实现此需求。当认证策略、UTM profile 等配置引用策略信息组时，“替换信息组”中的替换内容优先于“替换信息”功能中配置的替换内容。

替换信息组类型	用途	替换资源
安全（utm）	在防火墙策略中引用的 UTM 配置文件、SSL VPN 等页面	admin alertmail custom-message fortiguard-wf ftp http icap mail nac-quar spam sslvpn traffic-quota utm webproxy
认证（auth）	防火墙策略中开启的认证页面	auth webproxy

配置举例

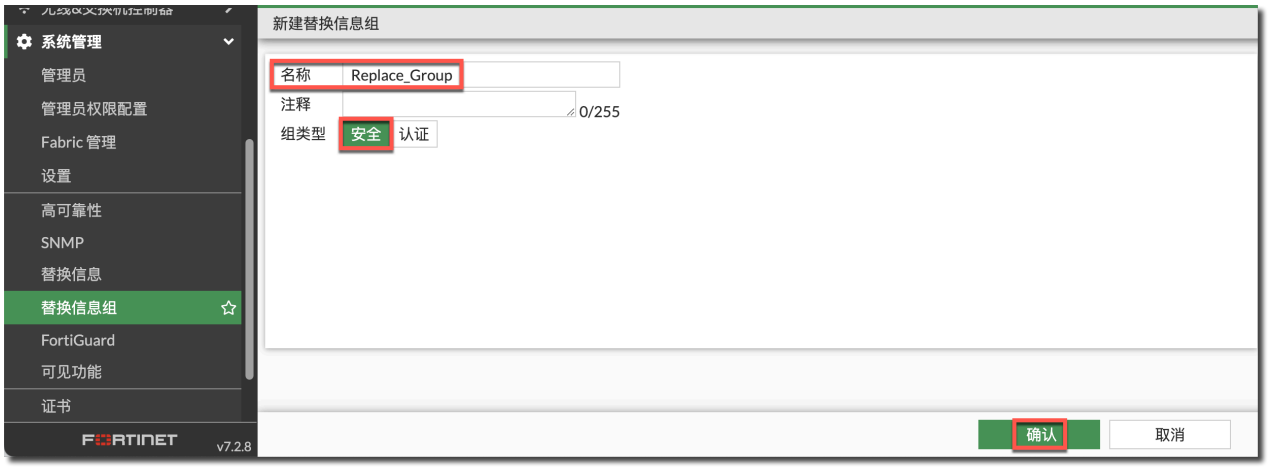
1. 在 GUI 界面开启“替换信息组”功能的显示。

```bash
config system settings
    set gui-replacement-message-groups enable
end
```

2. 进入 FortiGate 的“系统管理 → 替换信息组”页面，点击“新建”按钮。



3. 配置替换信息组名称，“组类型”选择“安全”，点击“确认”按钮下发配置。



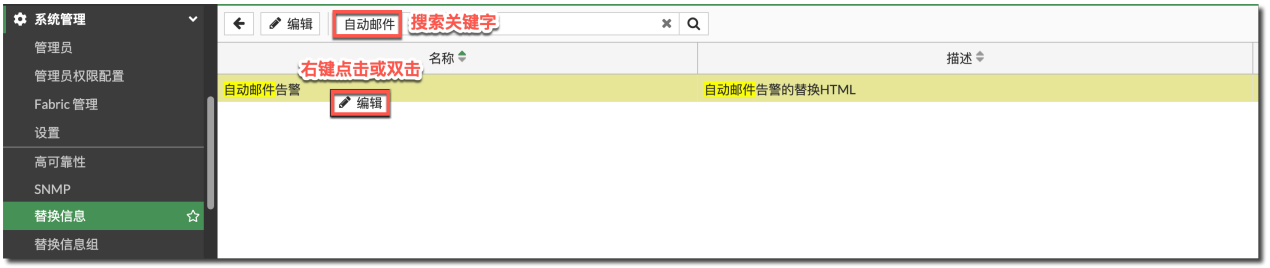
4. 右键点击新建的替换信息组，点击“编辑”按钮（或双击编辑）。



5. 搜索关键字“自动邮件”，找到“自动邮件告警”条目，右键或双击编辑该条目。

重要

替换信息组中也包含其他功能模块页面的替换信息，当用户需要替换多个功能模块的页面信息时，可以使用此功能。



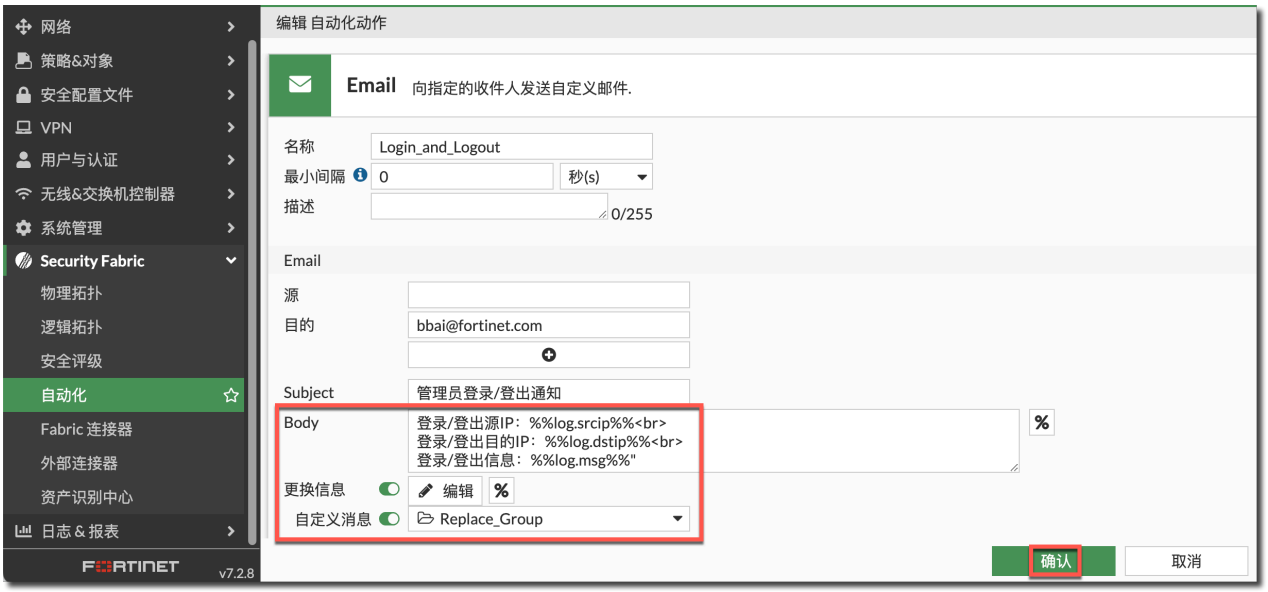
6. 在 HTML 编辑页面，将想要添加的告警图片 URL 添加到 `message-container` 部分，如下图所示（`<img src="https://www.fortinet.com/content/dam/fortinet/images/icons/benefits/icon-wireless-fortinet-ips.svg" alt="Fortinet Logo"/>`），左侧的预览可以正常显示该图片，点击“备份配置”按钮保存配置。




```bash
config system replacemsg-group
    edit "Replace_Group"
        set comment ''
        set group-type utm
        config automation
            edit "automation-email"
                set buffer "<!DOCTYPE html>
<html lang=\"en\">
 <head>
 <meta charset=\"UTF-8\">
 <meta http-equiv=\"X-UA-Compatible\" content=\"IE=8; IE=EDGE\">
 <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">
 <link href=\"https://fonts.googleapis.com/css?family=Roboto&display=swap\" rel=\"stylesheet\">
 <style>
 body {
 height: 100%;
 font-family: Roboto, Helvetica, Arial, sans-serif;
 margin: 0;
 display: flex;
 align-items: center;
 justify-content: center;
 }
 .message-container{
 margin: 0 auto;
 max-width: 580px;
 }
 .email-body {
 line-height: 1.5em;
 }
 </style>
 </head>
 <body>
 <div class=\"message-container\">
 <img src=\"https://www.fortinet.com/content/dam/fortinet/images/icons/benefits/icon-wireless-fortinet-ips.svg\"
alt=\"Fortinet Logo\"/>
 <h1>
 Security Fabric Automation
 </h1>
 <h3>
```

```
 %%AUTOMATION_FGT_SERIAL%%: %%AUTOMATION_STITCH_NAME%%
 </h3>
 <div class=\"email-body\">
 %%AUTOMATION_EMAIL_BODY%%
 </div>
 </div>
 </body>
</html>
"
                set header http
                set format html
            next
        end
    next
end
```

7. 进入 FortiGate 的“Security Fabric → 自动化 → 操作”页面，编辑自动邮件动作。开启“更换信息”选项，然后开启“自定义消息”选项，并选择上步创建的替换信息组。注意由于调用了替换信息组（HTML 格式），所以需要在“Body”中每行加入 HTML 换行符 `<br>`。点击“确认”按钮下发配置。




```bash
config system automation-action
    edit "Login_and_Logout"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "管理员登录/登出通知"
        set message "登录/登出源IP：%%log.srcip%%<br>
登录/登出目的IP：%%log.dstip%%<br>
登录/登出信息：%%log.msg%%\""
        set replacement-message enable
        set replacemsg-group "Replace_Group"
    next
end
```

其他功能引用

1. 在 WebFilter 配置文件中引用，所有被此 WebFilter 阻断返回的页面会被替换为替换信息组 newutm 中配置的内容。

```bash
config webfilter profile
    edit "default"
        set replacemsg-group "newutm"
    next
end
```

2. 在防火墙策略中引用，所有匹配该策略的认证用户的页面会被替换为替换信息组 newauth 中配置的内容。

```bash
config firewall policy
    edit 1
        ...
        set replacemsg-override-group "newauth"
        ...
    next
end
```

登录提示信息

在 Web/SSH 管理用户登录前和登录后都可以弹出提示信息，以展示警告信息，这些提示信息可以在替换信息页面进行自定义。

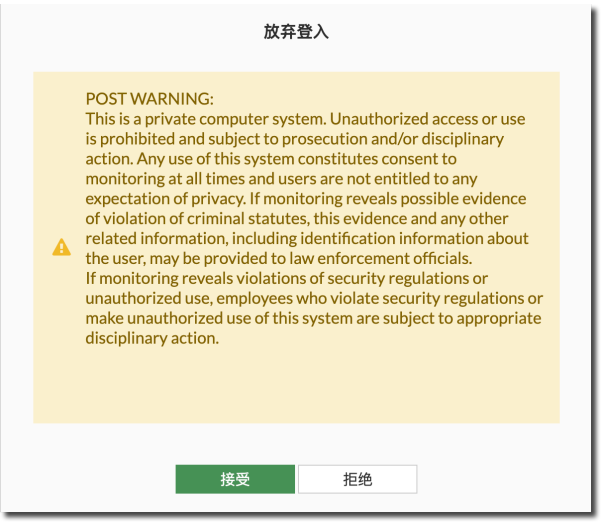
配置步骤

1. 在 CLI 下开启登录前或登录后的的提示信息。

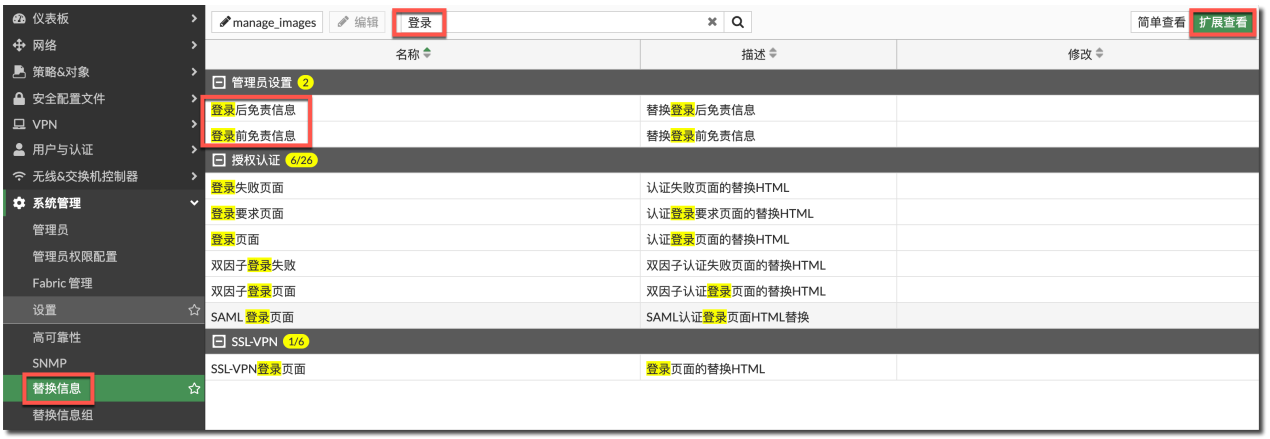
```
config system global
    set pre-login-banner enable
    set post-login-banner enable
end
```

2. 默认情况下，登录前和登录后提示信息均为英文。





3. 可以在替换信息页面对这两种提示信息进行自定义，进入“系统管理 → 替换信息”页面，右上角切换到拓展查看，搜索“登录”，找到管理员设置中的登录前免责信息和登录后免责信息。

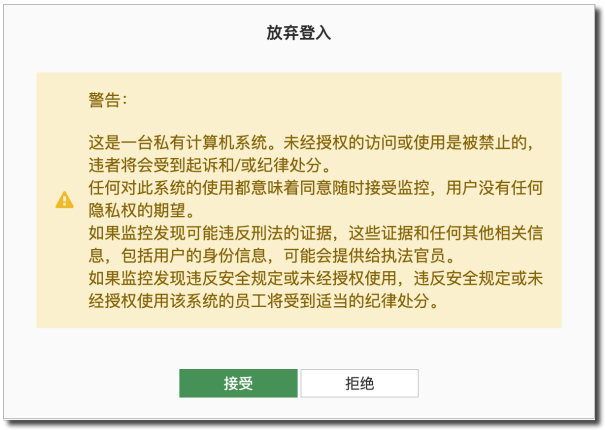


4. 编辑登录前或登录后免责信息配置，右侧填写自定义的提示内容，左侧显示预览信息，填写完成后点击备份配置按钮。

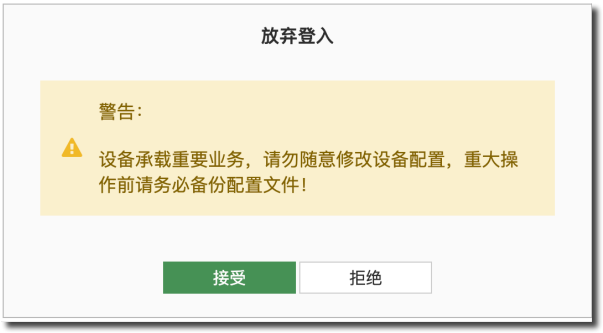


结果验证

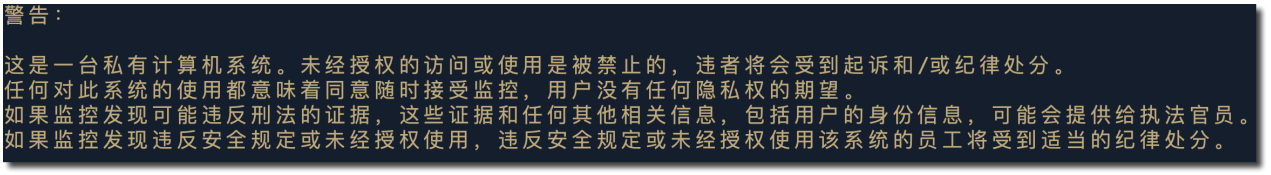
1. 使用 Web 方式登录 FortiGate，在登录前，就会弹出告警信息。



2. 点击接受继续登录，登录完成后，弹出登录后的告警信息。



3. 使用 SSH 登录 FortiGate，在登录前，就会弹出告警信息。



4. 登录完成后，提示登录后的告警信息，键盘键入 a 后登录成功。

```
警告：

设备承载重要业务，请勿随意修改设备配置，重大操作前请务必备份配置文件！

(Press 'a' to accept):

FortiGate #
```

网络管理

路由模式

- 静态地址上网配置
- ADSL 线路上网配置
- ARP 表
- npu-neighbor-update
- 代理 ARP

透明模式

- 虚拟接口对（VWP）[推荐]

  虚拟接口对介绍 开启虚拟接口对并保护上网流量 虚拟接口对的网络管理问题 VLAN 与虚拟接口对 组播 (OSPF) 与虚拟接口对 非 IP 流量 (PPPOE) 与虚拟接口对 生成树 (STP) 与虚拟接口对 虚拟接口对最佳配置实践
- 传统透明模式

  开启透明模式防火墙并保护上网流量 VLAN 与透明模式 组播 (OSPF) 与透明模式 非 IP 流量 (PPPOE) 与透明模式 生成树 (STP) 与透明模式 透明模式的带外管理 Bypass 部署 透明模式注意事项与排错

旁路模式

- 旁路模式典型案例

DNS

- DNS 配置
- 穿越接口的 DNS 服务
- 可选 DNS 服务器

DHCP 配置

- DHCP 服务器配置
- DHCP 静态绑定
- DHCP 中继配置

端口聚合配置

交换接口

- 软交换配置
- VLAN/Hardware Switch
- 交换接口的 MAC 地址表

EMAC VLAN

SD-WAN

- SD-WAN 介绍

  SD-WAN 数据转发逻辑 SD-WAN 接口成员组 SD-WAN 健康检查
- SD-WAN 规则

  Manual（手动选路） Best Quality（最佳质量） Lowest Cost (SLA) Maximize Bandwidth (SLA)
- SD-WAN 配置举例

  将已关联接口移至 SD-WAN 区域 互联网出口场景 互联网出口和 VPN 混合场景 ADVPN 场景
- 健康检查配置实践
- SD-WAN 常用 SNMP 监控项
- SD-WAN 排错

NetFlow

- NetFlow 的配置
- NetFlow 模板
- 在 VPN 接口上配置 NetFlow

端口聚合配置

LACP 应用场景

1. 在带宽比较紧张的情况下，可以通过逻辑聚合可以扩展带宽到原链路的 n 倍。
2. 在需要对链路进行动态备份的情况下，可以通过配置链路聚合实现同一聚合组各个成员端口之间彼此动态备份。

LACP 模式

LACP 的端口可以支持如下几种模式：static、passive、 active：

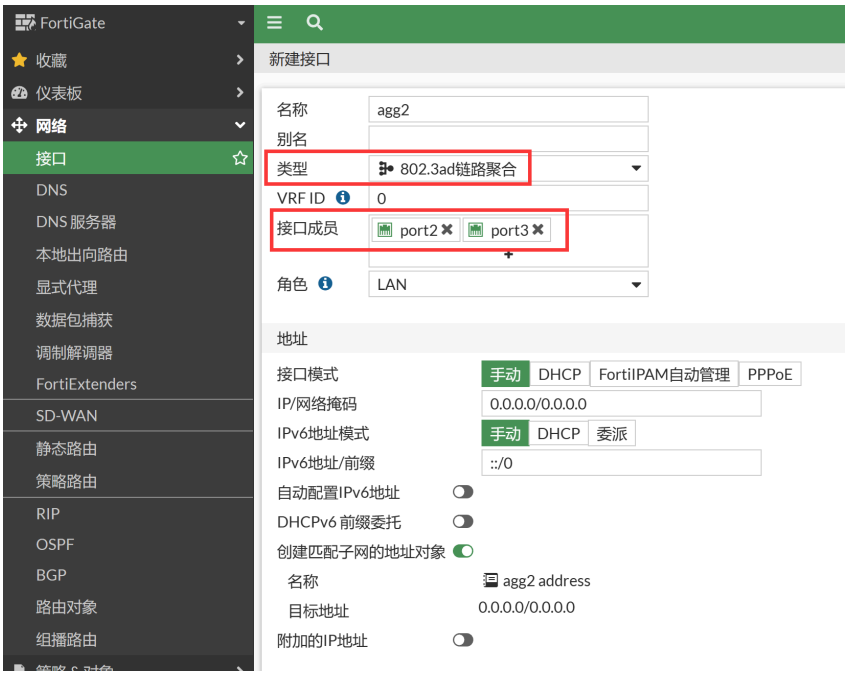
- Static：人为配置的聚合组，不允许系统自动添加或删除手工或静态聚合端口。
- passive：被动模式，该模式下端口不会主动发送 LACPDU 报文，在接收到对端发送的 LACP 报文后，该端口进入协议计算状态。
- active：主动模式，该模式下端口会主动向对端发送 LACPDU 报文，进行 LACP 协议的计算。

LACP 配置

1. 添加聚合接口，进入网络 → 接口，新建接口。



2. 类型选择 802.3ad 聚合，选择物理接口成员。



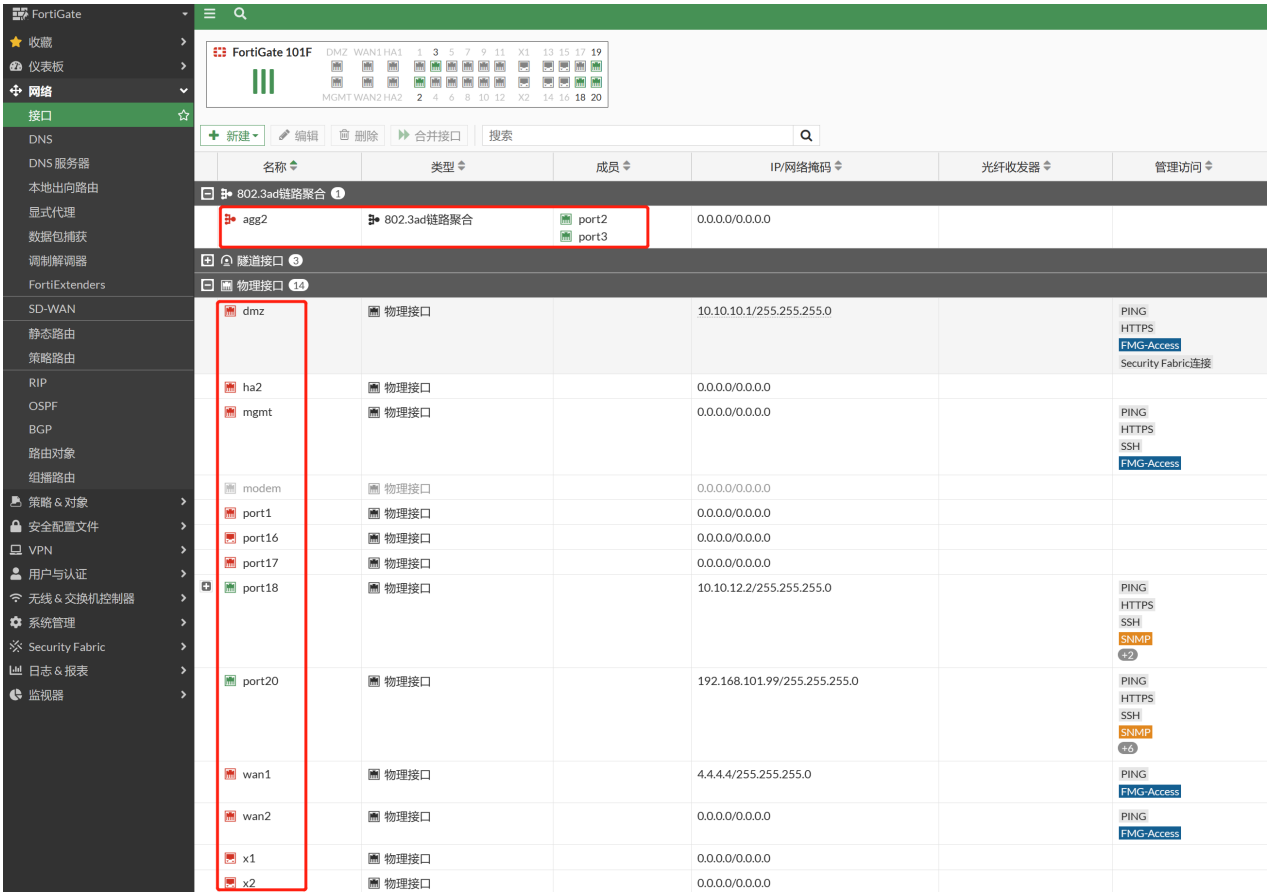
重要

建议在不需要的情况下关闭接口配置下的“设备探测”（Device detection）功能，该功能用于 MAC 地址厂商设备信息识别及 MAC 地址过滤，会消耗较多的设备资源。

3. 修改 LACP 的配置（可选）。

```
FortiGate # config system interface
FortiGate (interface) # edit agg2
FortiGate (lacp) # set lacp-mode active            //配置LACP协商模式：主动，被动或者静态，默认为动态
FortiGate (lacp) # set algorithm L3                //负载均衡算法：L3基于IP地址进行哈希，L4基于四层进行哈希
FortiGate (lacp) # end
```

4. 配置完成后查看配置聚合口配置，对应的物理口在 WEB/CLI 界面上将消失，不可配置。



5. 查看接口的命令行配置。

```
config system interface
    edit "agg2"
        set vdom "root"
        set type aggregate
        set member "port2" "port3"
        set lldp-transmission enable
        set role lan
        set snmp-index 40
        set algorithm L3
    next
end
```

6. 查看 LACP 接口状态。

```
FGT1-MASTER # diagnose netlink aggregate name agg2
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: up
npu: n
```

```
flush: n
asic helper: y
ports: 2
link-up-delay: 50ms
min-links: 1
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 33
actor MAC address: 50:00:00:03:00:01
partner key: 1
partner MAC address: aa:bb:cc:80:a0:00

slave: port9
  index: 0
  link status: up
  link failure count: 0
  permanent MAC addr: 50:00:00:03:00:01
  LACP state: established
  actor state: ASAIEE
  actor port number/key/priority: 1 33 255
  partner state: ASAIEE
  partner port number/key/priority: 1 1 32768
  partner system: 36864 aa:bb:cc:80:a0:00
  aggregator ID: 1
  speed/duplex: 10000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4

slave: port10
  index: 1
  link status: up
  link failure count: 0
  permanent MAC addr: 50:00:00:03:00:03
  LACP state: established
  actor state: ASAIEE
  actor port number/key/priority: 2 33 255
  partner state: ASAIEE
  partner port number/key/priority: 2 1 32768
  partner system: 36864 aa:bb:cc:80:a0:00
  aggregator ID: 1
  speed/duplex: 10000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4
```

EMAC VLAN

警告

- 在 NP6XLite（SoC4）平台设备（如 60F/80F/100F/200F 等），请勿在 EMAC VLAN 相关流量被 NPU 加速的状态下使用此功能，否则可能会出现业务中断，其他平台设备可正常使用。
- 如必须在 NP6XLite（SoC4）平台设备使用此功能，请关闭 EMAC VLAN 相关的防火墙策略（`config firewall policy`）的 NPU 加速功能（`set auto-asic-offload disable`）。

功能简介

增强型 MAC VLAN（Enhanced MAC VLAN，以下简称 EMAC VLAN）是 FortiGate 设备中实现 VLAN 的一种方式。EMAC VLAN 主要用于需要通过同一物理接口模拟多个 VLAN 接口的情况。与传统的 MAC VLAN 不同，它增加了以下功能：

- 支持多个 MAC 地址：共享同一底层接口的每个 EMAC VLAN 都有一个唯一的 MAC 地址，可以与不同的 IP 地址和网络配置相结合。
- VIP 和 IP Pool 支持：由于每个 EMAC VLAN 都具有唯一的 MAC 地址，每个 EMAC VLAN 接口都可以配置 VIP 和 IP Pool（普通的 VLAN 如果共享同一底层接口，由于多个 VLAN 的 MAC 地址相同，会使邻居路由器的 ARP 表无法区分）。
- SNAT 支持：在策略中，可以选择禁用 SNAT，从而保留源 IP 地址的原始信息（原因与上条一致）。

工作原理

EMAC VLAN 并不会直接将数据包分配到底层交换机。在配置了 VLAN ID 的情况下，数据包会被打上 VLAN Tag，并通过底层接口的驱动程序进行处理。如果 VLAN Tag 不匹配，数据包将不会通过该接口。

使用 VLAN ID 的限制

在 EMAC VLAN 中配置 VLAN ID 时，VLAN ID 和底层接口必须唯一配对。所以即使是不同的 VDOM，也不能使用相同的 VLAN ID。

透明模式的限制

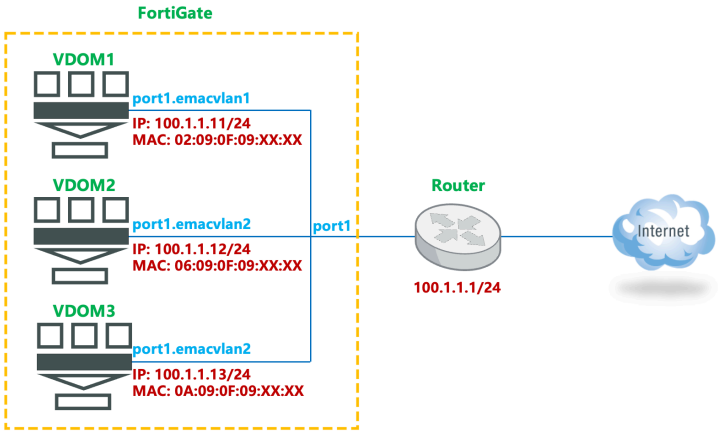
EMAC VLAN 不能在透明模式的 VDOM 中使用。在透明模式下，数据包会携带原始源的 MAC 地址，而不是接口的 MAC 地址，这与 EMAC VLAN 的工作方式冲突。

HA 中的处理

在 HA 配置中，EMAC VLAN 接口被视为物理接口，且有唯一的物理接口 ID。EMAC VLAN 接口的 MAC 表会与 HA 集群中的其他设备同步。在 HA 配置中，每个接口会被分配一个虚拟 MAC 地址。

配置示例

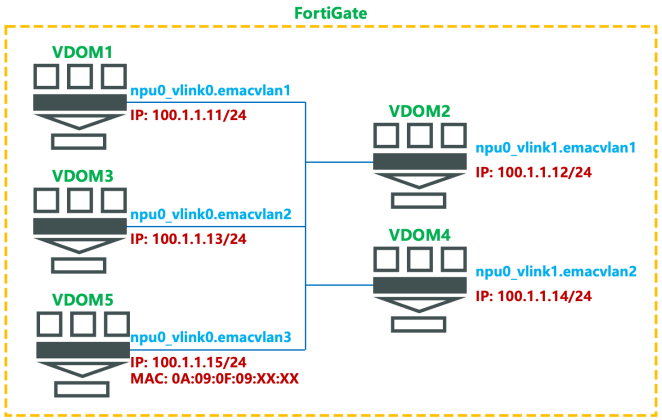
多个 VDOM 共享同一物理接口



- FortiGate 有 4 个 VDOM：root（默认）、VDOM1、VDOM2、VDOM3，port1 接口属于 root VDOM。
- FortiGate 通过 port1 连接到 Router，三个 VDOM 共享 port1。
- 每个 VDOM 配置一个 EMAC VLAN 接口，并且每个接口都有唯一的 MAC 地址，从而使得多个 VDOM 可以共享一个物理接口。

```bash
config global

config system interface
    edit "port1"
        set vdom "root"
        set type physical
        set snmp-index 9
    next
end

config system interface
    edit port1.emacvlan1
        set vdom VDOM1
        set ip 100.1.1.11 255.255.255.0
        set type emac-vlan
        set interface port1
    next
    edit port1.emacvlan2
        set vdom VDOM2
        set ip 100.1.1.12 255.255.255.0
        set type emac-vlan
        set interface port1
    next
    edit port1.emacvlan3
        set vdom VDOM3
        set ip 100.1.1.13 255.255.255.0
        set type emac-vlan
        set interface port1
    next
end
end
```

使用 NPU Vlink 接口进行连接



- FortiGate 有 6 个 VDOM：root（默认）、VDOM1、VDOM2、VDOM3、VDOM4、VDOM5，VDOM1/VDOM3/VDOM5 通过 NPU Vlink 与 VDOM2/VDOM4 互联。
- FortiGate NPU Vlink 为点对点连接，NPU Vlink 接口上的 VLAN 接口使用相同的 MAC 地址。由于 VLAN 接口共享相同 MAC 地址，不建议使用 NPU Vlink 和 VLAN 接口连接超过两个 VDOM。为避免同一 NPU Vlink 上 MAC 地址重叠，需要改用增强型 MAC VLAN。

```bash
config global
config system interface
    edit npu0_vlink0.emacvlan1
        set vdom VDOM1
        set ip 100.1.1.11 255.255.255.0
        set type emac-vlan
        set interface npu0_vlink0
    next
    edit npu0_vlink0.emacvlan2
        set vdom VDOM3
        set ip 100.1.1.13 255.255.255.0
        set type emac-vlan
        set interface npu0_vlink0
    next
    edit npu0_vlink0.emacvlan3
```

```
        set vdom VDOM5
        set ip 100.1.1.15 255.255.255.0
        set type emac-vlan
        set interface npu0_vlink0
    next
    edit npu0_vlink1.emacvlan1
        set vdom VDOM2
        set ip 100.1.1.12 255.255.255.0
        set type emac-vlan
        set interface npu0_vlink1
    next
    edit npu0_vlink1.emacvlan2
        set vdom VDOM4
        set ip 100.1.1.14 255.255.255.0
        set type emac-vlan
        set interface npu0_vlink1
    next
end
end
```

VLAN 接口使用独立的 MAC 地址

- 某些网络要求当 VLAN 接口共享同一物理端口时，每个 VLAN 接口必须拥有唯一的 MAC 地址。在此情况下，EMAC VLAN 接口的使用方式与普通 VLAN 接口相同。
- 还有一种情况是 FortiGate 做 HA 集群后，业务接口会使用 HA 的虚拟 MAC（如 `00:09:0f:09:07:00`，实际的物理 MAC 为 `94:F3:92:53:AE:7A`），某些运营商的 PPPoE Server 会对这种虚拟 MAC 地址做限制，导致 FortiGate 的 PPPoE 拨号失败。EMAC VLAN 在 HA 下会使用物理 MAC 地址（如 `9A:F3:92:53:AE:7A`，在 EMAC VLAN 上创建的 PPPoE 也会继承该物理 MAC 地址），从而规避该问题。
- 配置时使用 `set vlanid` 命令设置 VLAN Tag。VLAN ID 与接口必须构成唯一对应关系，即使它们属于不同的 VDOM。

```bash
config system interface
    edit emacvlan1
        set type emac-vlan
        set vlanid 10
        set interface wan1
    next
end

FortiGate # diagnose hardware deviceinfo nic emacvlan1
Description             EMacvlan Ethernet driver v1.0
System_Device_Name      emacvlan1
Lower_Device_Name       wan1
Current_HWaddr          9a:f3:92:53:ae:7a
Permanent_HWaddr        9a:f3:92:53:ae:7a
State                   up
Link                    up
npudev_oid              64
macvlan_id              0
vlan_id                 10
learn_mac               no
mode                    2
```

注意事项

- 每个物理接口最多可以创建 512 个增强型 MAC VLAN 接口。
- 所有接口的 MAC 地址总数不能超过 600 个，包括增强型 MAC VLAN 接口的 MAC 地址以及其他接口的 MAC 地址。

路由模式

- 静态地址上网配置
- ADSL 线路上网配置
- ARP 表
- npu-neighbor-update
- 代理 ARP

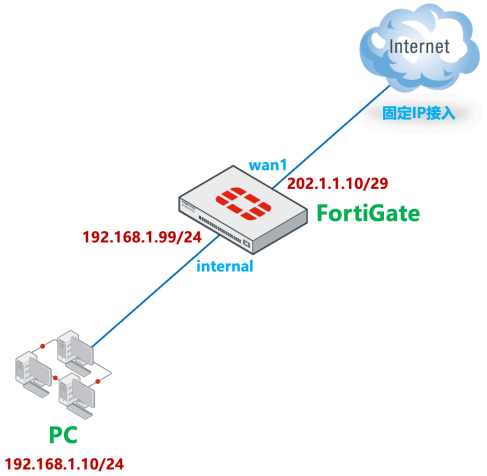
静态地址上网配置

网络需求

外网接口使用静态 IP，由运营商分配指定的静态地址，内网为 192.168.1.0/24 网段，实现基本上网功能。

网络拓扑

假设运营商分配的地址如下，网段：202.1.1.8/29，分配 IP 地址：202.1.1.10，网关地址：202.1.1.9，DNS：202.106.196.115。

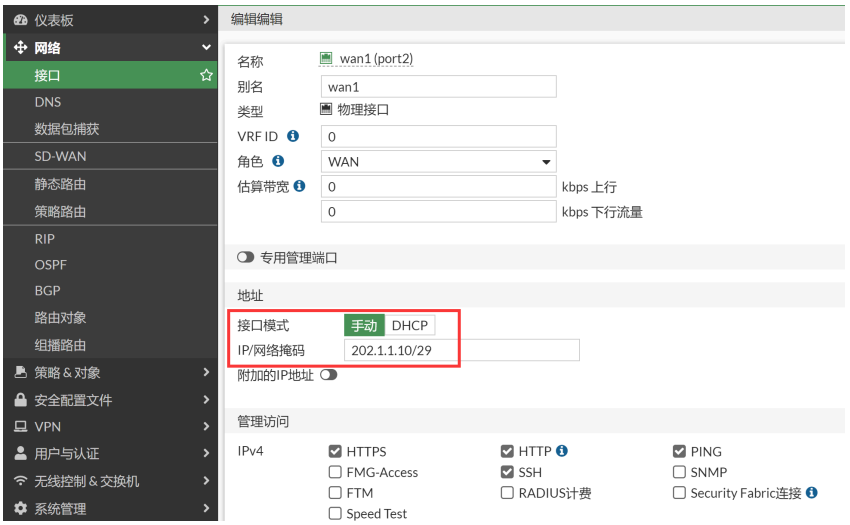


配置要点

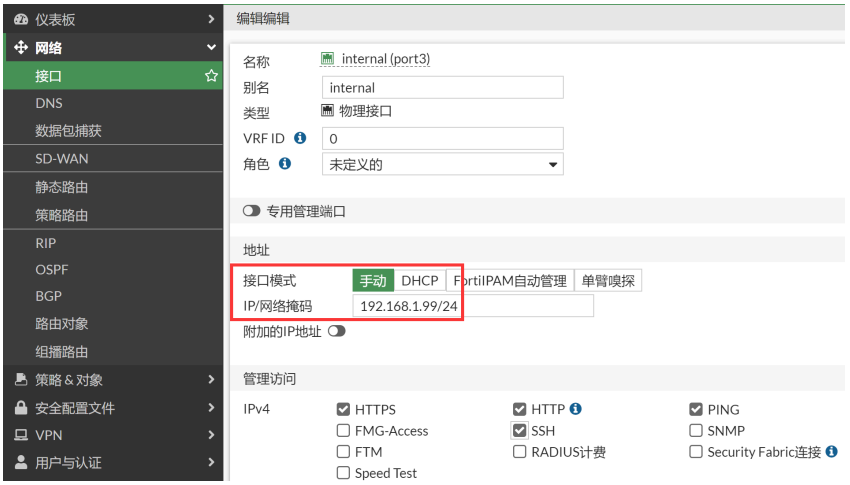
- 配置接口 IP
- wan1 口: IP 地址配置为互联网运营商所提供的地址
- internal 口: IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置静态路由表
- 配置地址对象 "lan"，内容为 192.168.1.0/24
- 配置从 internal 到 wan1 口的策略，并开启 NAT

操作步骤

1. 配置接口 IP，进入网络 → 接口 → 编辑 wan1，对于 202.1.1.8/29 网段，202.1.1.8 为网络地址和 202.1.1.15 广播地址均不可用，202.1.1.9 为运营商网关所用，202.1.1.9~202.1.1.14 为可用 IP 地址范围。所以配置给 wan1 接口的地址我们选择为 202.1.1.10。



2. 编辑 internal，该接口的默认地址为 192.168.1.99/24，根据实际情况更改即可。可选择性地开启接口的管理功能。建议内部开启 HTTPS、SSH、PING 服务。



3. 配置静态路由，进入网络 → 静态路由，点击新建。

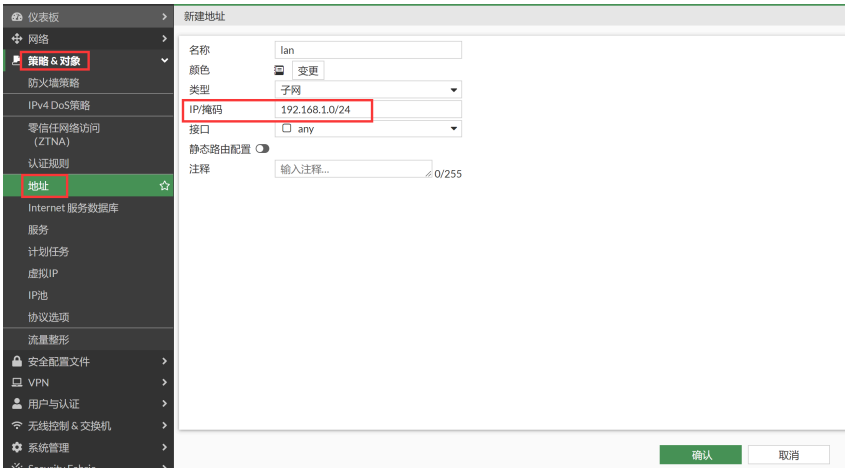


4. 新建静态路由。

- 目的地址：由于是默认网关，使用默认的 0.0.0.0/0.0.0.0 即可
- 接口：该路由所关联的接口，wan1 口，必须正确填写，否则该路由无法工作
- 网关：下一跳 IP 地址，即 wan1 口对端运营商设备接口的 IP 地址
- 管理距离：默认 10
- 高级选项 - 优先级：默认 1



5. 配置地址资源，进入策略 & 对象 → 地址，新建地址，名称配置为 "lan"，地址节点选择子网 "192.168.1.0/24"，点击确认。

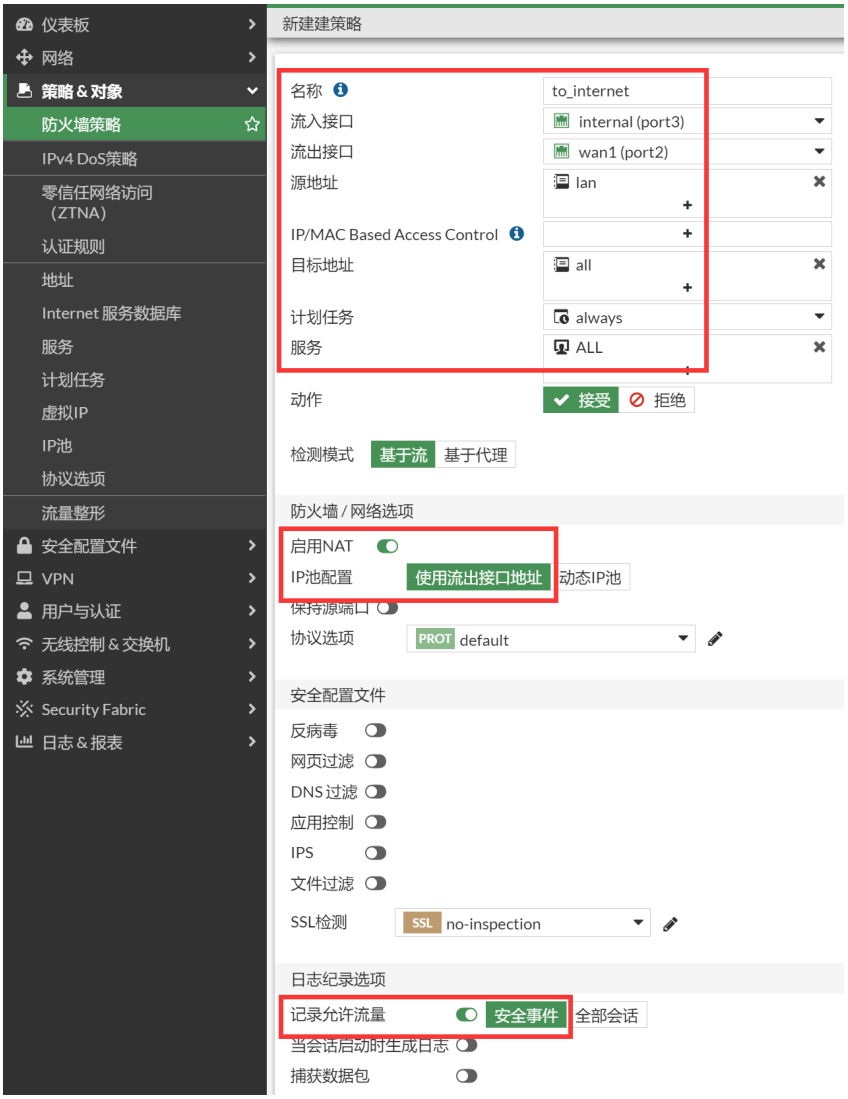


6. 配置防火墙策略（对于某些低端型号，系统会默认配置一条从 internal 到 wan1 的 NAT 策略），进入策略 & 对象 → 防火墙策略，点击新建按钮。



7. 在新建页面中，按照如下方式配置策略，点击确定按钮后，系统自动保存配置，策略生效。

   - 流入接口：internal
   - 源地址：选择刚才定义的地址对象“lan”
   - 流出接口：wan1
   - 目的地址选择: all，代表所有的地址
   - 时间表：always
   - 服务: ALL
   - 动作：ACCEPT
   - NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
   - 注意： 启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志



结果验证

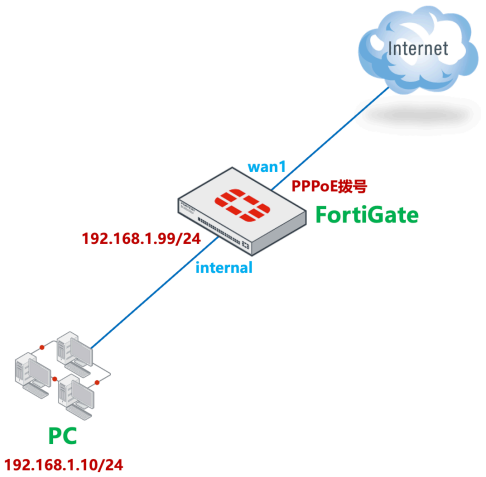
将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 配置为当地的 DNS，电脑可正常上网。

ADSL 线路上网配置

网络需求

外网接口使用 ADSL 拨号，内网为 192.168.1.0/24 网段，实现基本上网功能。

网络拓扑



配置要点

- wan1（port2）口: 接 ADSL 的接口，务必开启自动获取网关，这样 ADSL 拨号成功后设备会自动生成默认路由，无需手动配置默认路由
- internal（port3）口: IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置地址对象“lan”，内容为 192.168.1.0/24
- 配置从 internal（port3）到 pppoe 口的策略，并开启 NAT

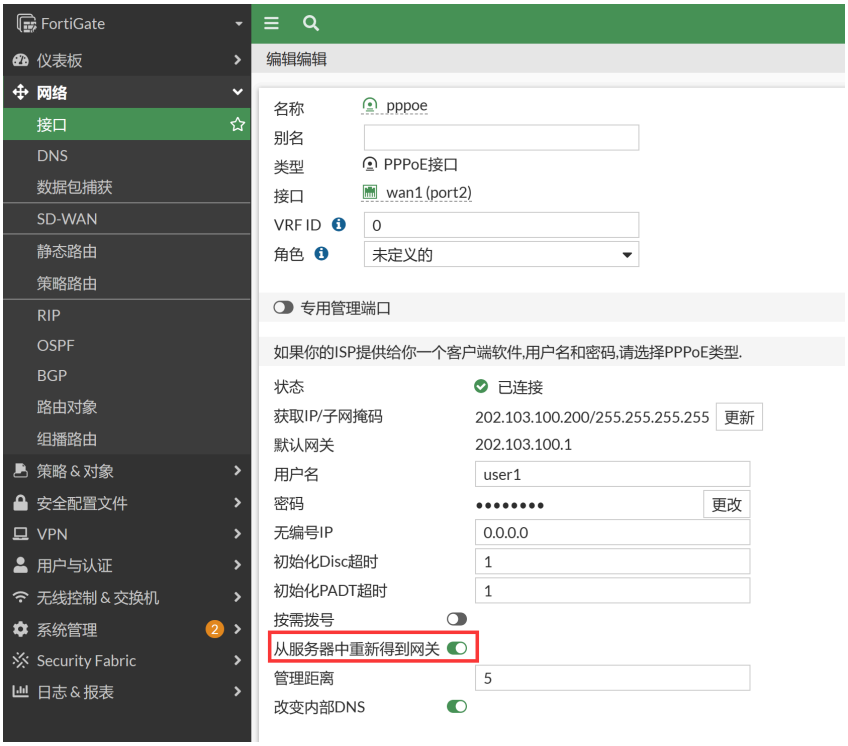
配置步骤

1. 配置 wan1 口的 PPPoE，建议在 CLI 下配置 PPPoE 接口，使用此方式配置获取到 IP 后会自动添加默认路由到运营商。

> **重要**
>
> 建议通过 CLI 的方式新建 PPPoE 虚拟接口，参考：部署前注意事项 → PPPoE 接口 ↗配置章节。

```
config system pppoe-interface
    edit "pppoe"
        set device "port2"
        set username "user1"
        set password xxxxxxxx
    next
end

以上命令执行后，会自动在system interface下创建同名接口，其中包含“从服务器中重新得到网关”的配置
FortiGate # show full sys int pppoe
config system interface
    edit "pppoe"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set defaultgw enable
        set interface "port2"
    next
end
```

2. 以上配置完成后，可以在 GUI 上看到 wan1（port2）下多了一个 PPPoE 接口，并获取到了运营商的 IP。可以看到“从服务器中重新得到网关”已开启。

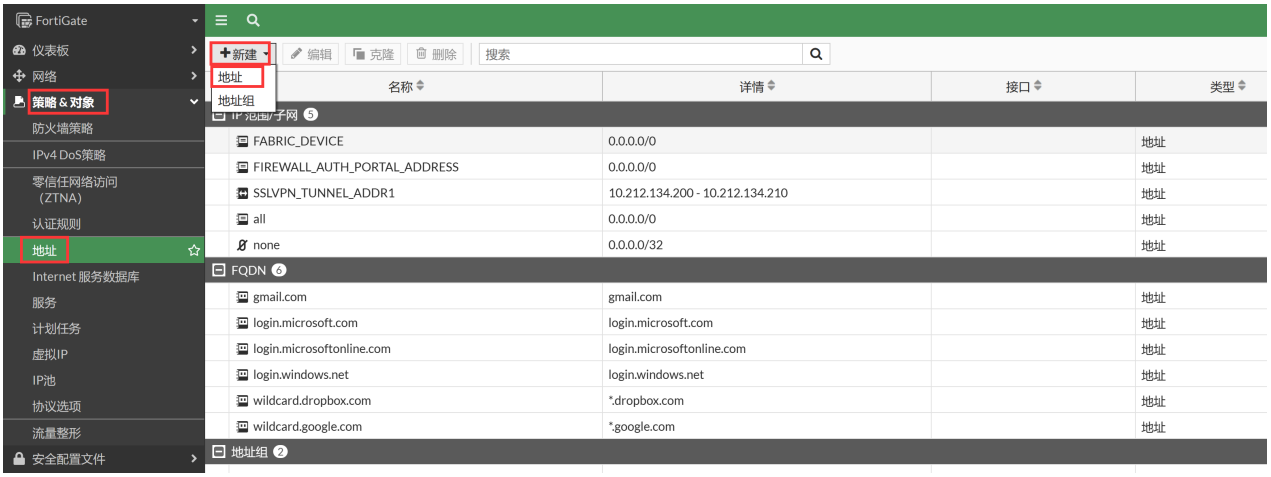



```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 202.103.100.1, pppoe, [1/0]
C       192.168.1.0/24 is directly connected, port3
C       202.103.100.1/32 is directly connected, pppoe
C       202.103.100.200/32 is directly connected, pppoe
```

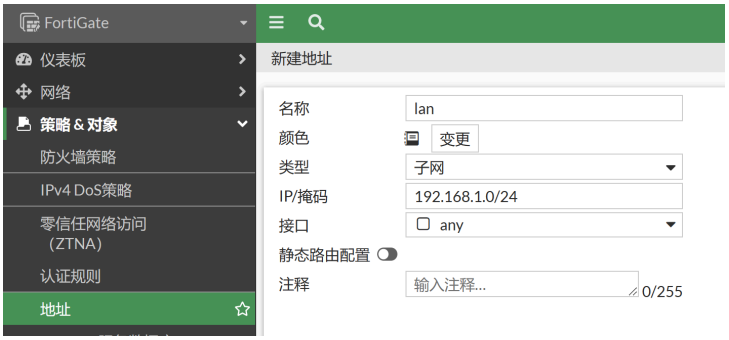
3. 配置 internal（port3）接口的 IP 为 192.168.1.99/24。选择性地开启接口的管理功能，建议内部开启 https ,ssh, ping 服务。



4. 配置地址资源，进入防火墙 & 对象 → 地址，点击新建地址按钮。

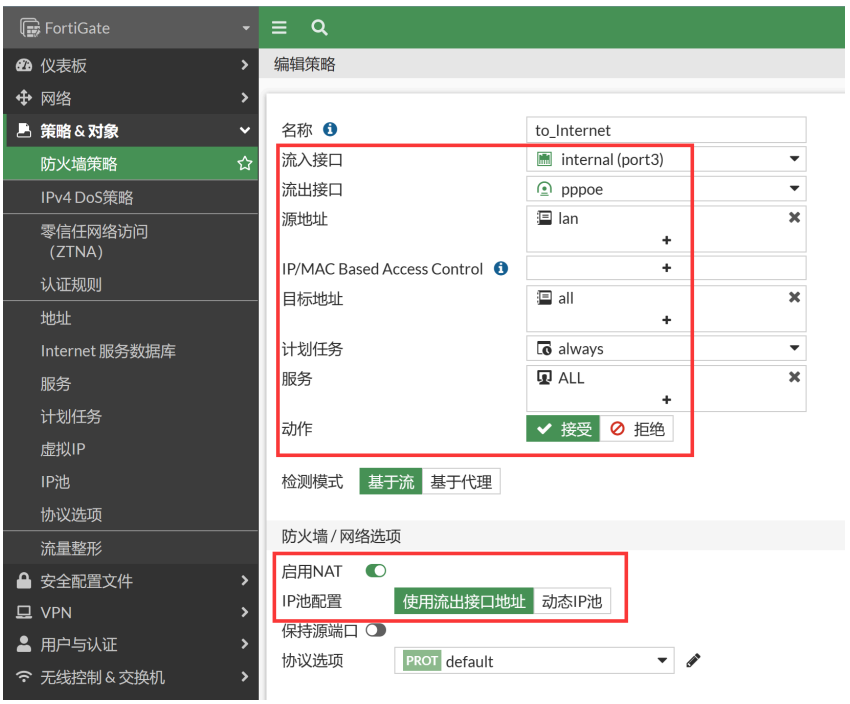


5. 名称配置为 lan，地址节点选择子网：192.168.1.0/24，点击确认。




```
config firewall address
    edit "lan"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，放通内网到 Internet 的流量，并开启 NAT，点击确定按钮后，系统自动保存配置，策略生效。



```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port3"
        set dstintf "port2"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：internal（port3）
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：pppoe 接口
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

结果验证

将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 一般设置为当地的 DNS 即可，电脑可正常上网。

ARP 表

功能介绍

ARP 表用于确定目标节点对应的 MAC 地址，以及该节点可达的 VLAN 和接口端口。

在 FortiGate 上，ARP 相关信息获取的方式主要通过如下命令：

- `get system arp`：查看当前 ARP 表（偏运维视图）。
- `diagnose ip arp list`：查看系统 ARP 缓存（包含状态、引用等细节）。
- `diagnose sys device list <vdom_name>`：查看 ARP/ARP6 的表规模与使用摘要。

查看 ARP 信息

查看 ARP 表

```bash
# get system arp

Address           Age(min)   Hardware Addr      Interface
10.10.1.3         1          50:b7:c3:75:ea:dd internal7
192.168.0.190     0          28:f1:0e:03:2a:97 wan1
192.168.0.97      0          f4:f2:6d:37:b0:99 wan1
```

查看 ARP 缓存明细

```bash
# diagnose ip arp list

index=7 ifname=dmz 169.254.0.1 94:f3:92:53:ae:7c state=00000080 use=198045467 confirm=148592500 update=148592500 ref=0
index=28 ifname=to_Home_WAN11 0.0.0.0  state=00000040 use=2078 confirm=2078 update=107680361 ref=1
index=12 ifname=internal5 10.10.12.2 00:01:02:03:04:05 state=00000002 use=0 confirm=0 update=1366 ref=84
index=12 ifname=internal5 10.10.12.254 08:5b:0e:b2:5e:69 state=00000004 use=4787 confirm=4787 update=150 ref=5
index=6 ifname=wan2 172.22.6.1 00:09:0f:09:01:18 state=00000002 use=34 confirm=1190 update=2268 ref=5
index=5 ifname=wan1 172.22.6.255 state=00000000 use=79961 confirm=85961 update=79961 ref=2
index=5 ifname=wan1 172.22.6.1 00:09:0f:09:01:18 state=00000002 use=24 confirm=234 update=2268 ref=108
```

- `state`：ARP 条目状态。
- `ref`：表示当前被其他对象引用次数。
- 常见 ARP 状态位：

> **提示**
>
> ARP 状态迁移受 ARP 请求/应答、邻居查找、路由与会话行为等多因素影响。单次抓到的状态需要结合 `ref` 与时间维度一起判断。

状态值	含义	说明
`0x02`	REACHABLE	已收到 ARP 响应
`0x04`	STALE	预期时间内未收到 ARP 响应
`0x08`	DELAY	从 STALE 过渡到 REACHABLE 前的中间状态
`0x20`	FAILED	在最大探测次数内仍未解析成功
`0x40`	NOARP	设备不支持 ARP（如 IPsec 接口）
`0x80`	PERMANENT	静态 ARP 项

查看某个 VDOM 下 ARP/ARP6 摘要

```bash
diagnose sys device list <vdom_name>

#diagnose sys device list root

list virtual firewall root info:
ip4 route_cache: table_size=65536 max_depth=3 used=164 total=171
arp: table_size=16 max_depth=2 used=9 total=10
proxy_arp: table_size=256 max_depth=0 used=0 total=0
arp6: table_size=64 max_depth=1 used=11 total=11
proxy_arp6: table_size=256 max_depth=0 used=0 total=0
local table version=00000000 main table version=0000d5ea rpdb version=0000002b
vf=root dev=root vrf=0
vf=root dev=naf.root vrf=0
vf=root dev=l2t.root vrf=0
vf=root dev=ssl.root vrf=0
vf=root dev=fortilink vrf=0
vf=root dev=emacvlan1 vrf=0
vf=root dev=internal vrf=0
vf=root dev=to_Home_WAN11 vrf=0
vf=root dev=dhcp vrf=0
vf=root dev=test111 vrf=0
vf=root dev=Loopback1 vrf=0
vf=root dev=a vrf=0
vf=root dev=b vrf=0
vf=root dev=dmz vrf=0
vf=root dev=internal2 vrf=0
vf=root dev=internal3 vrf=0
vf=root dev=internal4 vrf=0
vf=root dev=internal5 vrf=0
vf=root dev=modem vrf=0
vf=root dev=wan1 vrf=0
vf=root dev=wan2 vrf=0
vf=root dev=havdlink1 vrf=0
ses=0/0 ses6=0/0 rt=0/0 rt6=0/0
```

ARP 缓存时间

当 FortiGate 访问新目的地址时：

- 目的地址直连：对目标 IP 发起 ARP 请求。
- 目的地址非直连：对出口接口网关 IP 发起 ARP 请求。

ARP 应答成功后会生成缓存项。缓存有效期由“实际可达时间（actual reachable time）”决定：

- 实际可达时间 = 基准可达时间（base reachable time）的 50%~150% 随机确定。
- 默认基准值为 30 秒（可以通过如下 CLI 进行修改，取值范围：`30000` ~ `3600000` ms），因此默认实际范围是 15~45 秒。

```
config system interface
    edit port1
        set reachable-time <integer>
    next
end
```

- 实际可达时间每 5 分钟重新计算一次。

ARP 缓存回收与容量阈值

满足以下条件的 ARP 项会进入垃圾回收候选：

- 状态为 `STALE (0×04)` 或 `FAILED (0×20)`。
- `ref=0`（无引用）。

ARP 表项回收机制：

- 垃圾回收每 30 秒运行一次。
- 已 stale/failed/无引用超过 60 秒的项会被清理。
- 当 ARP 项数量超过阈值时，也会触发回收。
- 若超出阈值，新的 ARP 项将无法加入 ARP 表。

配置 ARP 表阈值

`arp-max-entry` 范围：`131072` ~ `2147483647`，默认 `131072`。

```bash
config system global
    set arp-max-entry <integer>
end
```

NP 会话的 ARP/ICMP6 探测

默认情况下，FortiOS 不会对 NP 已加速会话持续发送 ARP 探测。长时间无流量时，可能出现：

- FortiGate 侧 ARP 条目老化。
- 交换机 MAC 表老化。
- 后续流量触发广播 ARP，引入额外泛洪。

可以开启邻居更新探测，`npu-neighbor-update` 默认关闭。

> **相关信息**
>
> 详细机制参考：网络管理 → 路由模式 → npu-neighbor-update ↗。

```bash
config system global
    set npu-neighbor-update enable
end
```

手动维护 ARP 表

添加静态 ARP 表项

```bash
config system arp-table
    edit 1
        set interface "internal"
        set ip 192.168.50.8
        set mac bc:14:01:e9:77:02
    next
end
```

删除单条 ARP 表项

```bash
# diagnose ip arp delete <interface name> <IP address>
```

清空 ARP 表

```bash
# execute clear system arp table
```

排障建议

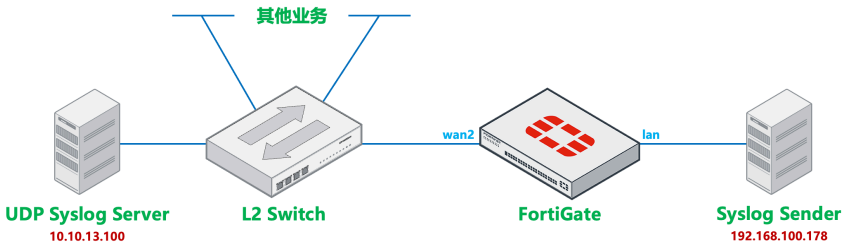
1. 先用 `get system arp` 看是否存在目标项。
2. 再用 `diagnose ip arp list` 看 `state` 与 `ref` 是否异常。
3. 如怀疑容量问题，检查 `arp-max-entry` 与 `diagnose sys device list root` 摘要。
4. 如为 NP 卸载的长连接场景，评估是否启用 `npu-neighbor-update`。

npu-neighbor-update

网络需求

在 FortiGate 上，流量在条件满足时会从 CPU 转发路径迁移到 NP（Network Processor）硬件加速，提升性能。

但在某些“单向/弱回包”业务（最常见是 UDP 单向发送，如 UDP Syslog）中，如果会话被 NP 加速，如下所示的场景中：



1. Syslog Sender 持续向 UDP Syslog Server 发送系统日志流量（UDP 端口 514）。UDP Syslog Server 不会主动发送任何数据包。
2. 默认配置下，在建立 UDP 会话且 UDP 流量经 NP 加速后，FortiGate 将不再对目标 IP（UDP Syslog Server）发送 ARP 探测。
3. 此期间因 Syslog Server 无任何数据发送，其在 L2 Switch MAC 地址表中的条目将超时删除（因为交换机是通过源 MAC 地址来学习 MAC 地址表）。
4. 随后发往该交换机的 UDP 流量将泛洪至交换机的所有端口，可能导致其他业务流量受阻。

解决方法

`npu-neighbor-update` 的目标就是：在这些 NP 加速会话存在时，通过主动探测来更新邻居信息，降低上述风险。该功能默认关闭，当 `npu-neighbor-update` 启用后：

```bash
config system global
    set npu-neighbor-update [enable/disable]
end
```

- 对 IPv4：FortiGate 会周期性发送 ARP 探测来更新邻居（ARP 表项/可达性）。
- 对 IPv6：FortiGate 会周期性发送 ICMPv6 NDP 探测（Neighbor Solicitation/Advertisement） 来更新邻居。

效果验证

1. 可使用 `diagnose sys session list` 相关命令查看会话 NP 加速状态，可以看到 Syslog 流量被 NP 单向加速。

```bash
session info: proto=17 proto_state=00 duration=33 expire=147 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=1056/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=44->8/8->44 gwy=10.10.13.1/0.0.0.0
hook=pre dir=org act=noop 192.168.100.110:161->10.10.13.100:514(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.13.100:514->192.168.100.110:161(0.0.0.0:0)
misc=0 policy_id=11 pol_uuid_idx=674 auth_info=0 chk_client_info=0 vd=0
serial=000014bf tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000400 ofld-O
npu info: flag=0x81/0x00, offload=8/0, ips_offload=0/0, epid=67/0, ipid=95/0, vlan=0x0000/0x0000
vlifid=95/0, vtag_in=0x0000/0x0000 in_npu=1/0, out_npu=1/0, fwd_en=0/0, qid=5/0
no_ofld_reason:
```

2. 抓包观察是否存在探测报文：

   - IPv4（ARP）：`diagnose sniffer packet <egress-intf> 'arp and host <dst-ip>' 6 0 l`。可以看到仅首次 ARP 探测为广播数据包，后续 ARP 探测数据包均为单播数据包。因此对网络影响甚微。

```bash
FortiGate # diagnose sniffer packet wan2 'arp and host 10.10.13.100' 6 0 l
interfaces=[wan2]
filters=[arp and host 10.10.13.100]
2023-09-22 09:29:55.517446 wan2 -- arp who-has 10.10.13.100 tell 10.10.13.2
0x0000   ffff ffff ffff e023 ff67 e39d 0806 0001        .......#.g......
0x0010   0800 0604 0001 e023 ff67 e39d 0a0a 0d02        .......#.g......
0x0020   0000 0000 0000 0a0a 0d64                       .........d

2023-09-22 09:29:55.517581 wan2 -- arp reply 10.10.13.100 is-at 00:09:0f:09:00:04
0x0000   e023 ff67 e39d 0009 0f09 0004 0806 0001        .#.g............
0x0010   0800 0604 0002 0009 0f09 0004 0a0a 0d64        ..............d
0x0020   e023 ff67 e39d 0a0a 0d02 0000 0000 0000        .#.g............
0x0030   0000 0000 0000 0000 0000 0000                  ............

2023-09-22 09:30:28.630294 wan2 -- arp who-has 10.10.13.100 tell 10.10.13.2
0x0000   0009 0f09 0004 e023 ff67 e39d 0806 0001        .......#.g......
0x0010   0800 0604 0001 e023 ff67 e39d 0a0a 0d02        .......#.g......
0x0020   0000 0000 0000 0a0a 0d64                       .........d

2023-09-22 09:30:28.630416 wan2 -- arp reply 10.10.13.100 is-at 00:09:0f:09:00:04
0x0000   e023 ff67 e39d 0009 0f09 0004 0806 0001        .#.g............
0x0010   0800 0604 0002 0009 0f09 0004 0a0a 0d64        ..............d
0x0020   e023 ff67 e39d 0a0a 0d02 0000 0000 0000        .#.g............
0x0030   0000 0000 0000 0000 0000 0000                  ............

2023-09-22 09:31:01.750293 wan2 -- arp who-has 10.10.13.100 tell 10.10.13.2
0x0000   0009 0f09 0004 e023 ff67 e39d 0806 0001        .......#.g......
0x0010   0800 0604 0001 e023 ff67 e39d 0a0a 0d02        .......#.g......
0x0020   0000 0000 0000 0a0a 0d64                       .........d

2023-09-22 09:31:01.750406 wan2 -- arp reply 10.10.13.100 is-at 00:09:0f:09:00:04
0x0000   e023 ff67 e39d 0009 0f09 0004 0806 0001        .#.g............
0x0010   0800 0604 0002 0009 0f09 0004 0a0a 0d64        ..............d
0x0020   e023 ff67 e39d 0a0a 0d02 0000 0000 0000        .#.g............
0x0030   0000 0000 0000 0000 0000 0000                  ............
......
```

- IPv6（NDP/ICMPv6）：`diagnose sniffer packet <egress-intf> 'icmp6 and host <dst-ipv6>' 4 0 l`。

```bash
FortiGate (Interim)# diagnose sniffer packet wan2 'host 2200::64' 4 0 l
interfaces=[wan2]
filters=[host 2200::64]
2023-09-21 10:59:29.680777 wan2 -- fe80::e223:ffff:fe67:e39d → 2200::64: icmp6: neighbor sol: who has 2200::64
2023-09-21 10:59:29.680929 wan2 -- 2200::64 → fe80::e223:ffff:fe67:e39d: icmp6: neighbor adv: tgt is 2200::64
2023-09-21 11:00:04.640780 wan2 -- fe80::e223:ffff:fe67:e39d → 2200::64: icmp6: neighbor sol: who has 2200::64
2023-09-21 11:00:04.640949 wan2 -- 2200::64 → fe80::e223:ffff:fe67:e39d: icmp6: neighbor adv: tgt is 2200::64
2023-09-21 11:00:39.600781 wan2 -- fe80::e223:ffff:fe67:e39d → 2200::64: icmp6: neighbor sol: who has 2200::64
2023-09-21 11:00:39.600920 wan2 -- 2200::64 → fe80::e223:ffff:fe67:e39d: icmp6: neighbor adv: tgt is 2200::64
......
```

3. 期望现象：

- 关闭时：一般只会看到业务自身触发的 ARP/NDP（或几乎看不到主动探测）。
- 开启后：可观察到由 FortiGate 发起的周期性 ARP probe 或 ICMPv6 邻居探测，从而帮助交换机刷新邻居可达性。

注意事项

该功能启用后会带来额外的 ARP/ICMPv6 探测流量与一定系统开销，建议只在问题场景中开启：

- 业务以 UDP 单向发送（如 UDP Syslog）为主、对端回包极少或没有。
- 会话被 NP 加速。
- 网络中存在对 MAC/邻居老化较敏感的二层域（例如交换机 MAC 表老化 导致的异常）。

如果您的业务本身是双向频繁通信（TCP 大量回包等），通常邻居/二层学习自然更稳定，一般不需要开启该功能。

VM 或关闭会话加速的场景下，由 CPU 维护邻居更新（无法关闭），不需要开启 `npu-neighbor-update`。

代理 ARP

功能简介

代理 ARP（Proxy ARP）允许 FortiGate 代表其他 IP 地址响应 ARP 查询。当对端主机误认为某目标 IP 与其位于同一网段，但实际该 IP 位于 FortiGate 后方或其他网络时，代理 ARP 可确保 ARP 能正常解析，使流量正确转发至 FortiGate。

- FortiGate 将以自身 MAC 地址回应指定 IP 的 ARP Request。
- 不会改变三层路由，仅处理二层解析问题。
- 支持 IP 段（start-ip/end-ip）。

典型应用场景包括：

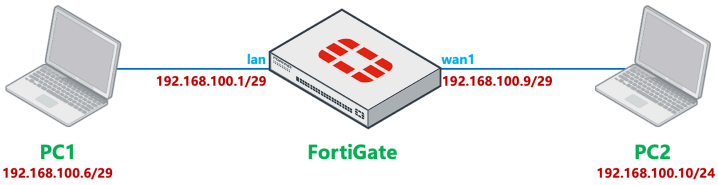
- 主机配置了错误的或更大的子网掩码导致跨网段访问失败。
- 公网地址无需配置为接口 Secondary IP，但仍需在接口上对外 ARP 响应（如 SNAT 使用“出接口地址”）。
- SD-WAN 或 ACME 不支持 Loopback 接口时，通过代理 ARP 实现公网 IP 的 ARP 响应。
- ISP 使用私网 Transit LAN 传递公网地址时，通过代理 ARP 让 CPE 学到下一跳 MAC。

配置方法

代理 ARP 在接口级别配置，可指定单个或多个 IP（连续地址段），用于在二层广播域内回应 ARP 请求。

```bash
config system proxy-arp
    edit <id>
        set interface {string}     ←——代理 ARP 生效的接口
        set ip {ipv4-address}      ←——需要代理响应 ARP 的起始 IP，如果只有一个IP，只需要配置此项
        set end-ip {ipv4-address} ←——需要代理响应 ARP 的结束 IP，需要配置 IP 范围时配置此项
    next
end
```

网络拓扑



- PC2 的地址为 `192.168.1.14/24`（而非/29）。PC2 需要访问 `192.168.1.6/29`（PC1） 时，由于使用 `/24` 掩码，它会认为 PC1 与其处于同一子网。
- 因此 PC2 会向本地网络广播 ARP 请求以解析 `192.168.1.6` 的 MAC 地址。该广播会在 FortiGate 的接口（地址 `192.168.1.9/29`）处终止。
- FortiGate 的该接口并不是 `192.168.1.6`。如果未在该接口上配置代理 ARP，FortiGate 将不会回应 PC2 的 ARP 请求，导致 PC2 无法访问 PC1。
- 启用代理 ARP 后，FortiGate 会使用自身的接口 MAC 地址回应 ARP，整个代理过程对 PC2 完全透明。
  - PC2 将 192.168.1.6 的 MAC 解析为 FortiGate 的接口 MAC。
  - PC2 把数据包交给 FortiGate。
  - FortiGate 再将数据包转发至真正的目标 PC1。

配置步骤

1. 基础网络配置（略）。

2. 在 FortiGate 的 wan1 接口配置代理 ARP，代理响应请求 PC1 的 IP `192.168.100.6` 的 ARP 广播。

```bash
config system proxy-arp
    edit 1
        set interface "wan1"
        set ip 192.168.100.6
    next
end
```

结果验证

1. 在 FortiGate 上查看代理 ARP 表项。

```bash
FortiGate # get sys proxy-arp
== [ 1 ]
id: 1   interface: wan1    ip: 192.168.100.6   end-ip: 0.0.0.0
```

2. 使用 PC2 Ping PC1，在 FortiGate 上抓包可以看到，PC2 首先直接向广播域发送 ARP 广播，请求 PC1 的 MAC 地址，FortiGate 的 wan1 接口代理响应了该 ARP 请求，并将自身 wan1 接口的 MAC 地址返回给 PC2。

```bash
FortiGate # diagnose sniffer packet any '(arp or icmp) and host 192.168.100.6' 4
interfaces=[any]
filters=[(arp or icmp) and host 192.168.100.6]
4.780241 wan1 in arp who-has 192.168.100.6 tell 192.168.100.10
5.532962 wan1 out arp reply 192.168.100.6 is-at e0:23:ff:67:e3:9c
5.533027 wan1 in 192.168.100.10 → 192.168.100.6: icmp: echo request
5.533068 lan out 192.168.100.10 → 192.168.100.6: icmp: echo request
5.533415 lan in 192.168.100.6 → 192.168.100.10: icmp: echo reply
5.533433 wan1 out 192.168.100.6 → 192.168.100.10: icmp: echo reply

FortiGate # diagnose netlink interface list wan1 | grep hw
Qdisc=mq hw_addr=e0:23:ff:67:e3:9c broadcast_addr=ff:ff:ff:ff:ff:ff
```

透明模式

虚拟接口对（VWP）[推荐]

- 虚拟接口对介绍
- 开启虚拟接口对并保护上网流量
- 虚拟接口对的网络管理问题
- VLAN 与虚拟接口对
- 组播 (OSPF) 与虚拟接口对
- 非 IP 流量 (PPPOE) 与虚拟接口对
- 生成树 (STP) 与虚拟接口对
- 虚拟接口对最佳配置实践

传统透明模式

- 开启透明模式防火墙并保护上网流量
- VLAN 与透明模式

  `Forward-Domain` `vlanforward` `Remapping VLAN ID (VLAN NAT，特殊场景使用)`
- 组播 (OSPF) 与透明模式
- 非 IP 流量 (PPPOE) 与透明模式
- 生成树 (STP) 与透明模式
- 透明模式的带外管理
- Bypass 部署
- 透明模式注意事项与排错

虚拟接口对（VWP）[推荐]

- 虚拟接口对介绍
- 开启虚拟接口对并保护上网流量
- 虚拟接口对的网络管理问题
- VLAN 与虚拟接口对
- 组播 (OSPF) 与虚拟接口对
- 非 IP 流量 (PPPOE) 与虚拟接口对
- 生成树 (STP) 与虚拟接口对
- 虚拟接口对最佳配置实践

虚拟接口对介绍

简介

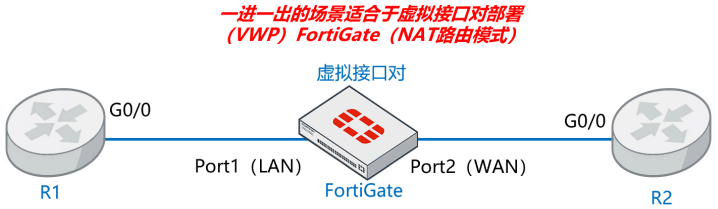
“Virtual Wire Pair”（虚拟接口对）功能提供了一种简便且灵活的类似于 FortiGate 透明模式的部署方式，“虚拟接口对”顾名思义包括了两个接口（一对接口，无 IP 地址），流量从一个接口进只能从另外一个接口出，无需查询二层表项，同时通过“虚拟接口对”转发的流量会走 NP 芯片进行 offload，具备很高的转发性能，当然转发流量是否通过最终还是由 FortiGate 的策略来控制的。由于“虚拟接口对”无需任何 IP 地址，因此“虚拟接口对”可以方便的插入现有网络中进行部署，和传统的 FortiGate 透明部署一样不需要改变现有组网的任何配置。

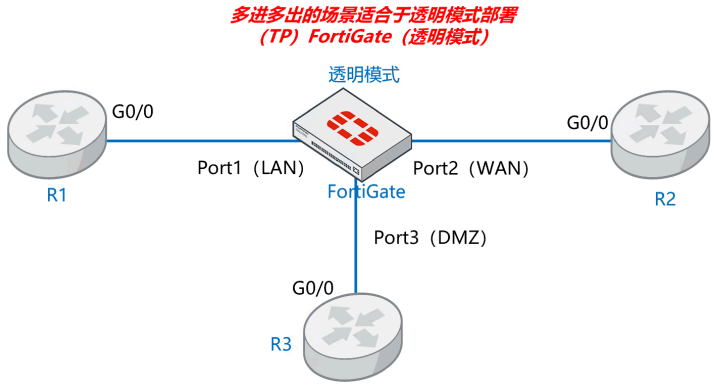
优势

- VWP 是直接运行在路由模式下的，FGT 无需切换到透明模式，无需配置 IP 地址，使用起来简单方便。
- 从一个接口进一定从另一个接口出，无需查询 MAC 表项，有更高的转发效率。
- 存在多 VLAN 的情况下，无需配置 VLAN 接口，无需配置 Forward-Domain，无需配置更多的策略，VWP 对于 VLAN 业务处理的设计和配置要比传统透明模式简洁很多。
- 一些简单的透明场景 VWP 更加合适，整个 VWP 的设计就是如此，部署起来比传统透明模式要简单很多。

劣势

只能一进一出两个接口部署，不能支持多接口的复杂透明模式场景，多进多出的透明部署场景需要使用到传统的透明模式部署。





总结

一进一出的透明模式场景，强烈推荐使用虚拟接口对部署（VWP），又简单又好用。

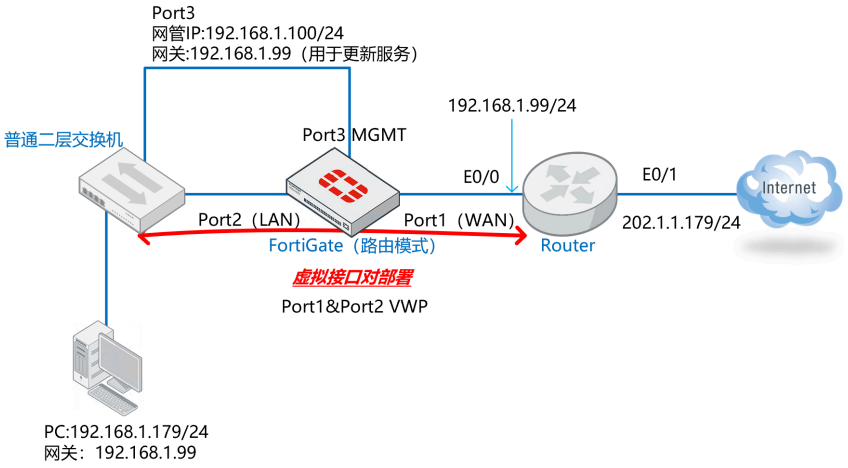
开启虚拟接口对并保护上网流量

网络需求

在不改变现有网络拓扑前提下，将防火墙路由模式部署到网络中，放在路由器和交换机之间，在此基础上开启虚拟接口对（VWP），对内网网段 192.168.1.0/24 的上网数据进行 4~7 层的安全防护。

虚拟接口对的策略和透明模式的策略/路由模式的策略使用起来几乎一样，没有太多的特别之处。

网络拓扑



配置要点

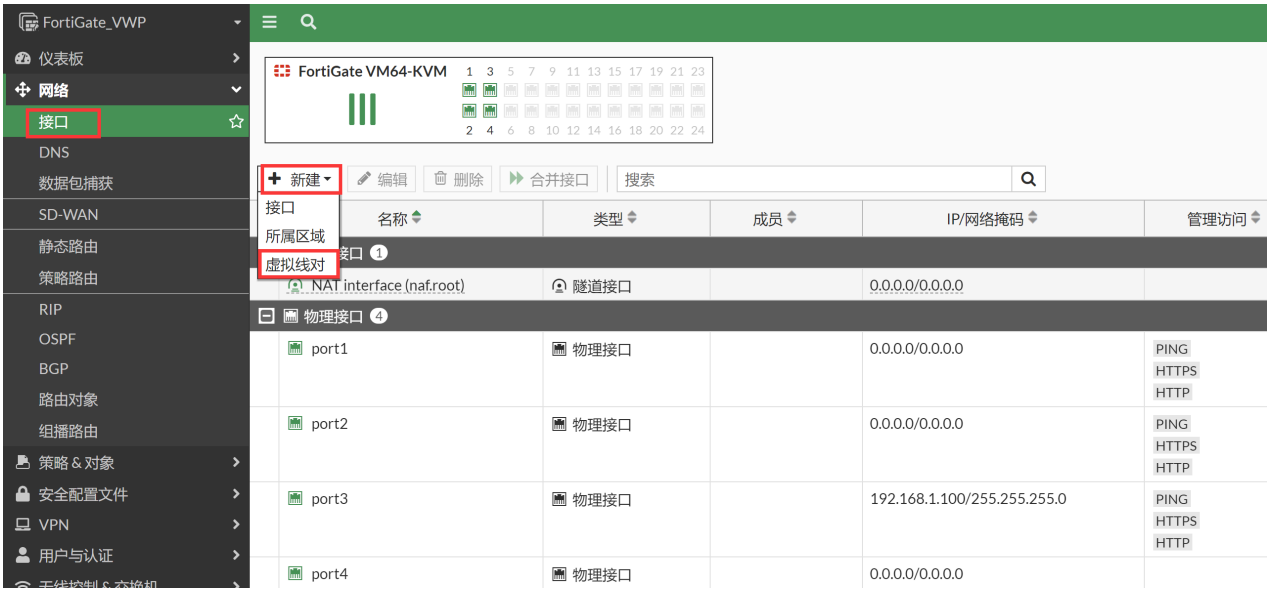
- Internet_R 路由器的基础配置
- 配置虚拟接口对（Port1 & Port2）
- 配置虚拟接口对安全策略允许内网 PC 访问互联网

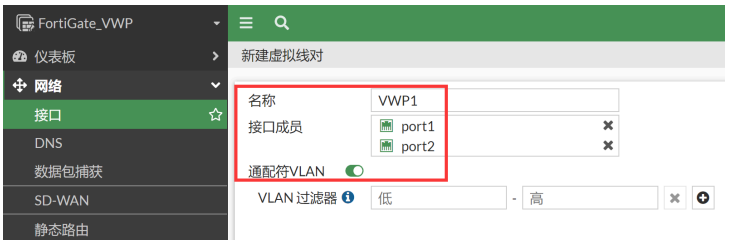
配置步骤

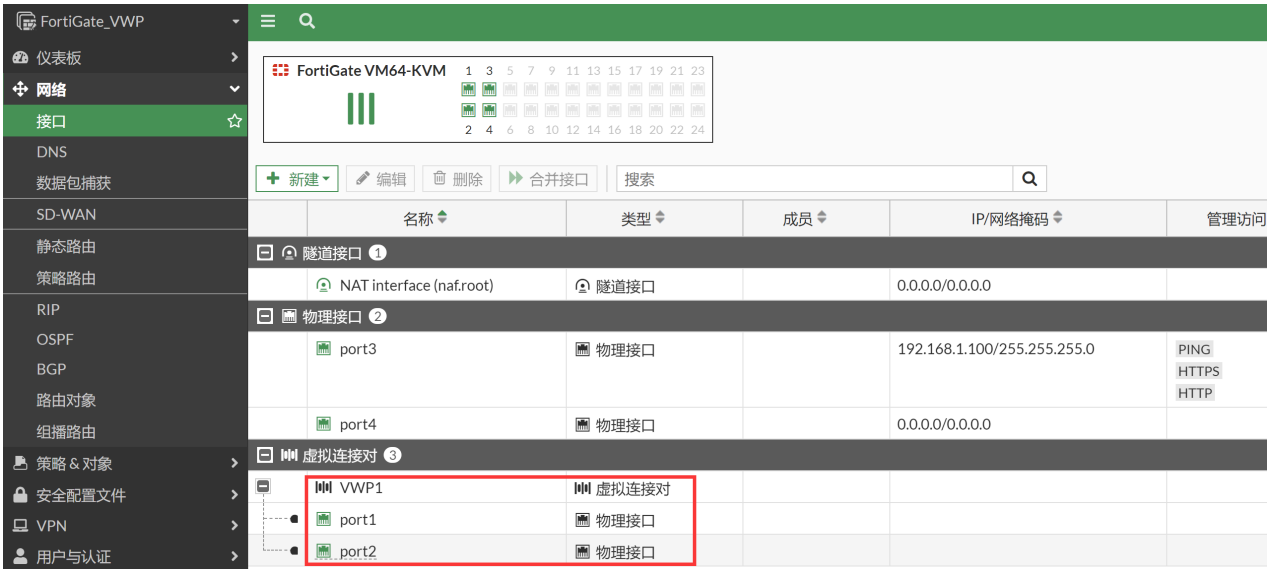
1. Internet_R 路由器的基础配置。

```
hostname Internet_R
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

2. 配置虚拟接口对（Port1 & Port2），防火墙处于路由模式的情况下，直接创建虚拟接口对：将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据。



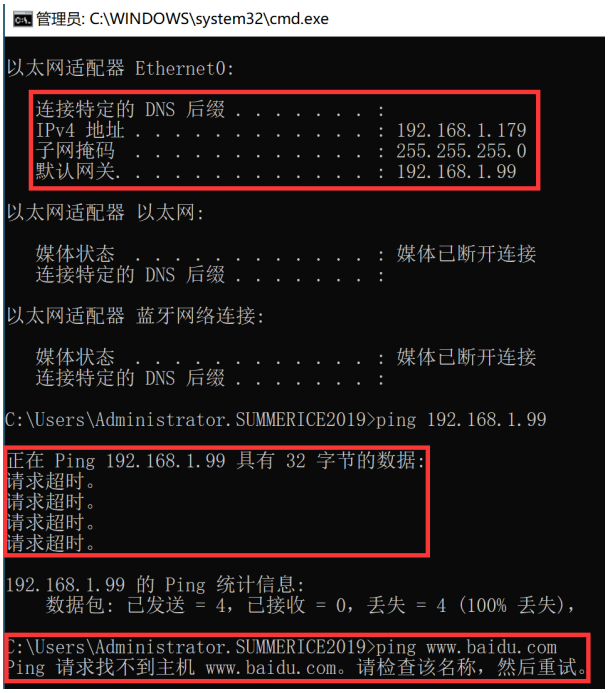




3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需要防火墙策略放通），对应的 CLI 如下：

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable
    next
end
```

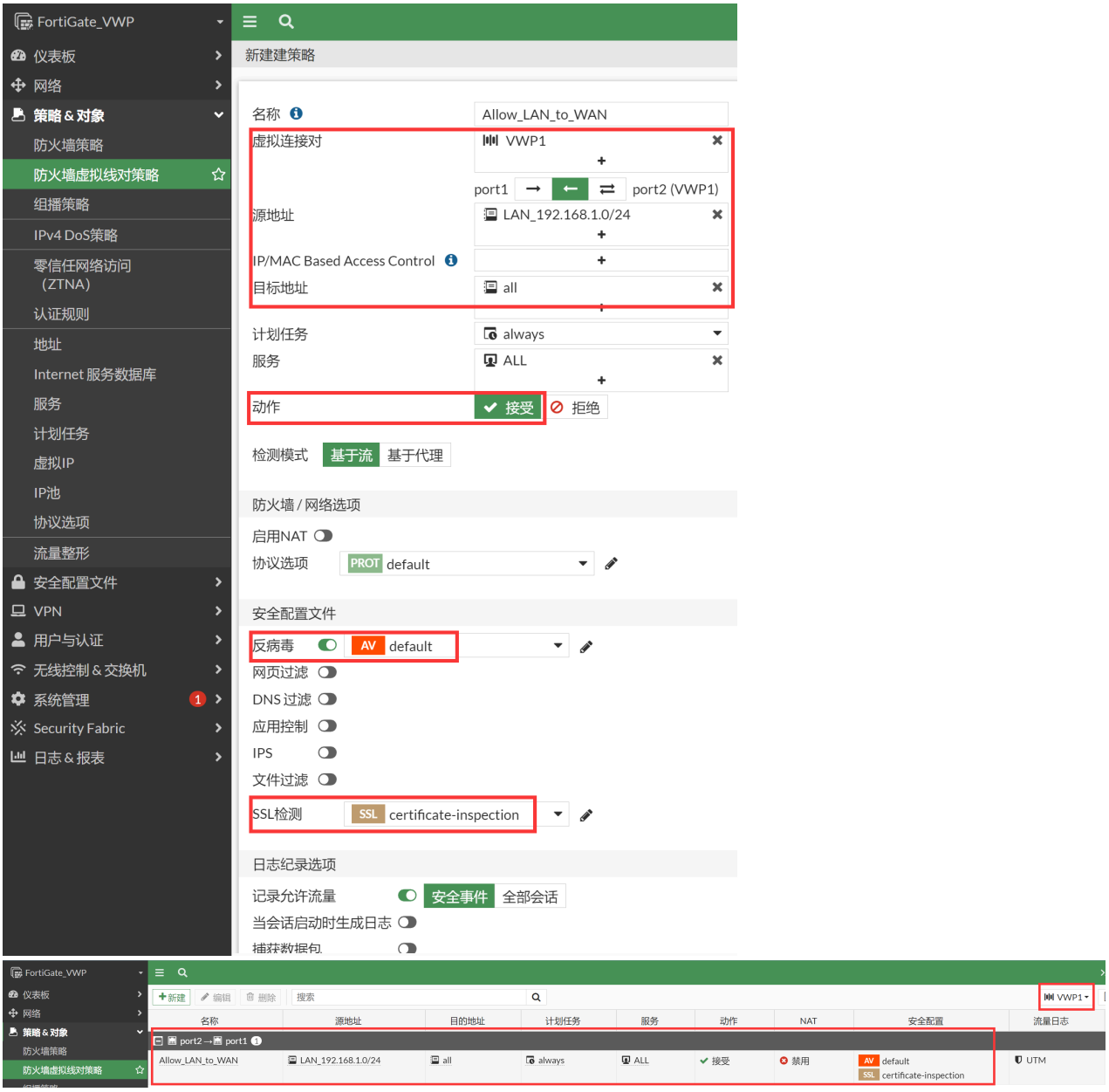
4. 配置虚拟接口对安全策略允许内网 PC 访问互联网，默认情况下 FortiGate 没有开启任何策略，此时 TCP/UDP/ICMP 的流量是全部不通的，例如默认 FGT 没有策略的情况下，内网 PC 访问网关和互联网资源都是被 Drop 的。



5. 如果需要想要 IPV4 业务可以通的话，则需要配置相应的安全策略。配置虚拟接口对的策略：允许 LAN（port2）到 WAN（port1）的上网流量穿过 FGT，策略 & 对象的位置会多出“防火墙虚拟线对策略”（如果没有看到，则刷新页面或注销管理登陆，重新登陆 FGT 即可看到）。



6. 右上角选择 VWP1 线对，方向选择 port2 → port1（也可选择双向的方向），新建 VWP 策略，开启 AV 功能并引用默认的 AV Profile（虚拟连接对可以选择多个，在同一条相同动作的策略中进行多接口控制）。

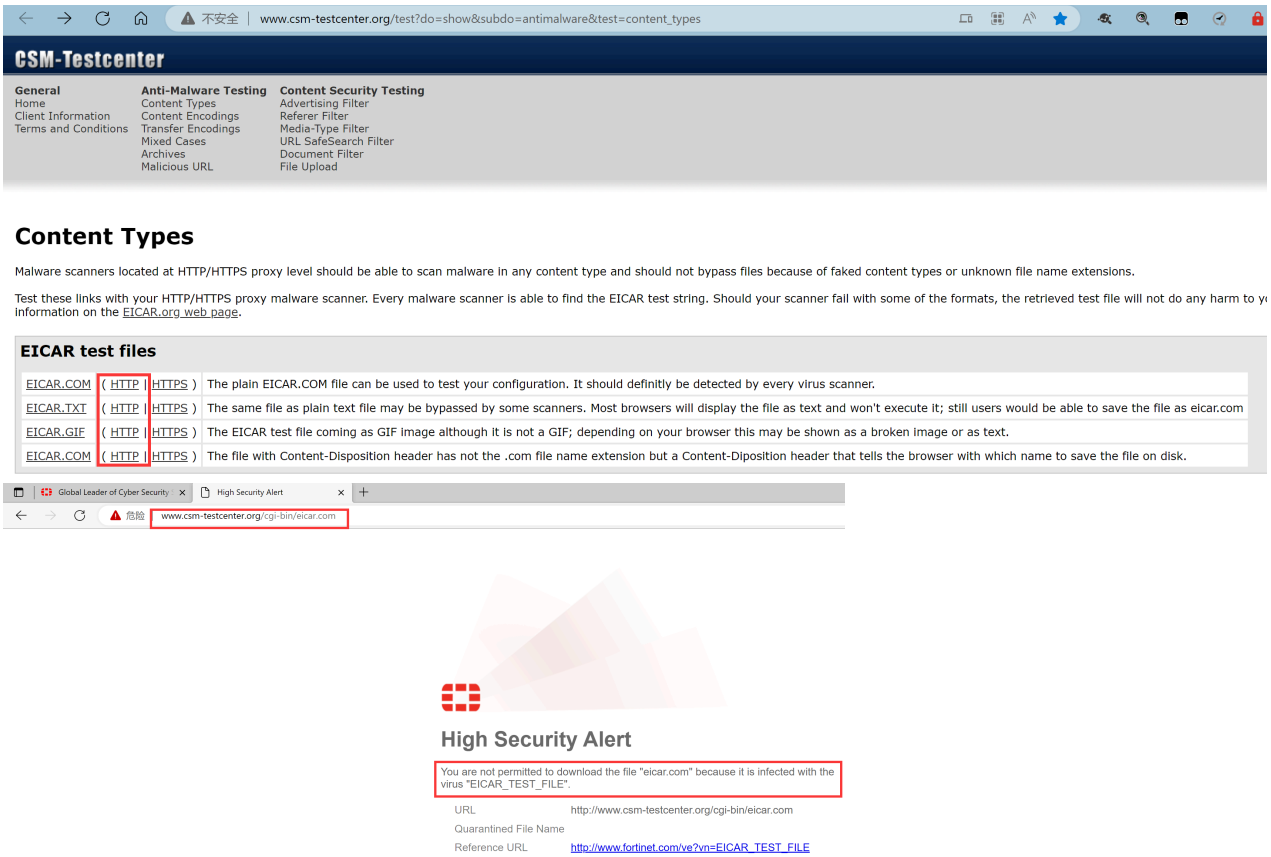


结果验证

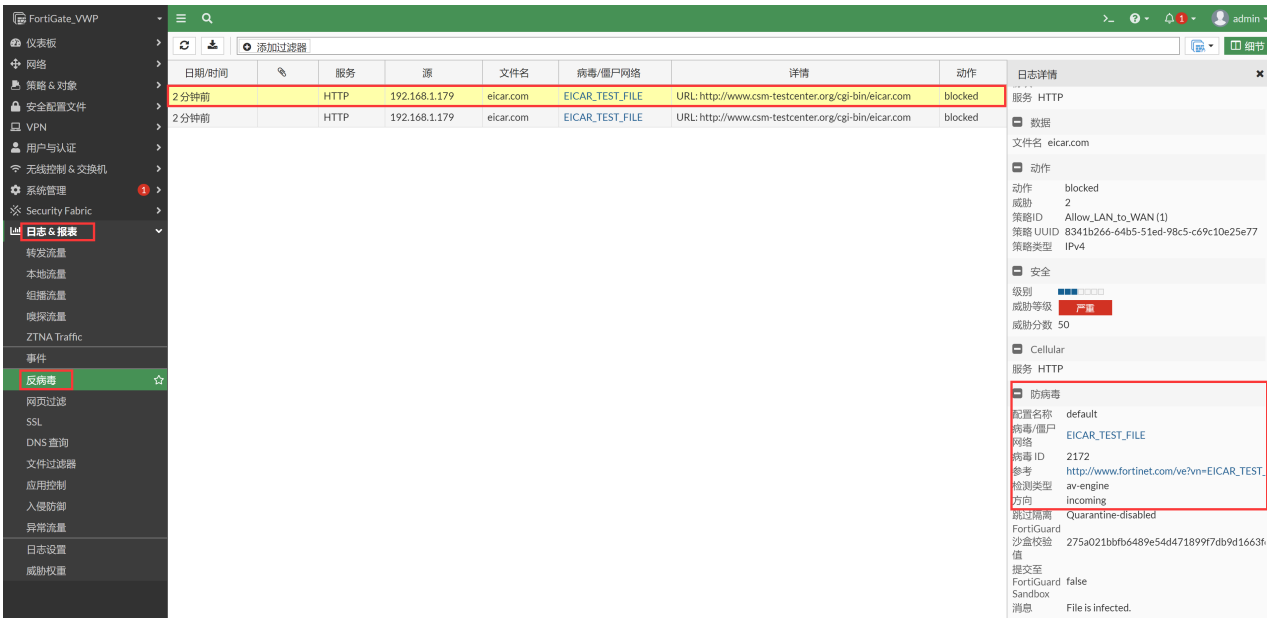
1. 使用内网 PC 访问外部网络，可以正常访问。



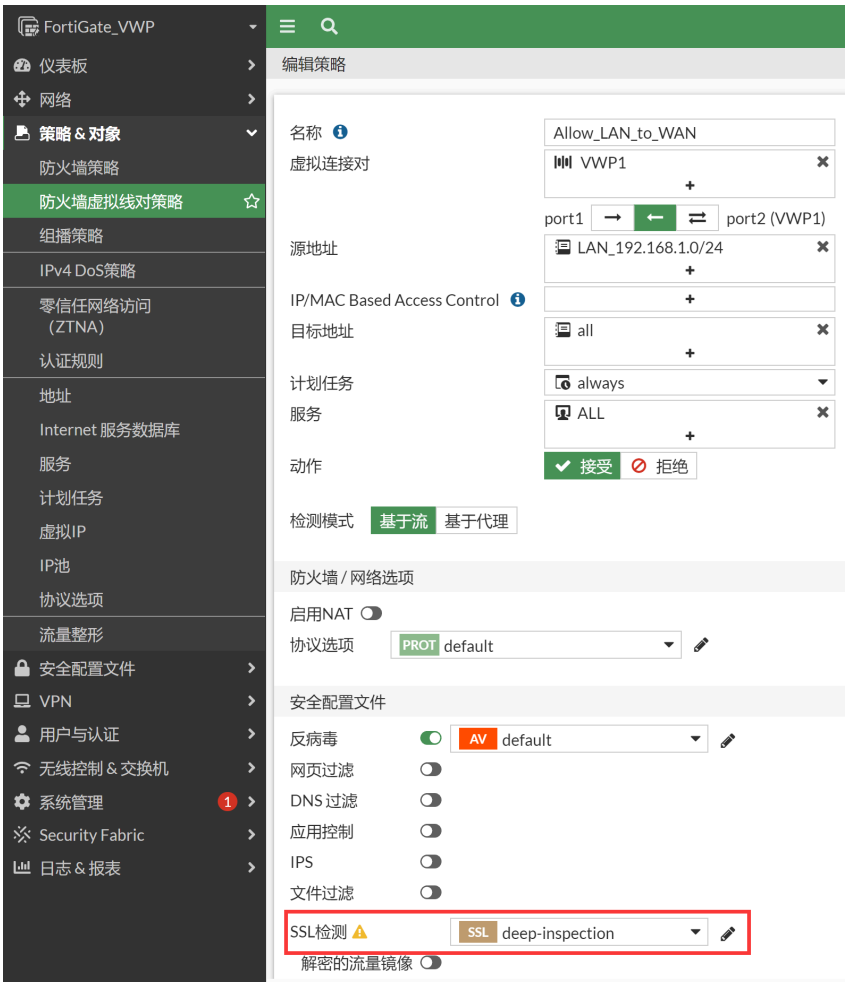
2. 测试通过 HTTP 下载病毒文件（http://www.csm-testcenter.org/test?do ↗ = show & subdo = antimalware & test = content_types），被 FortiGate 阻断。



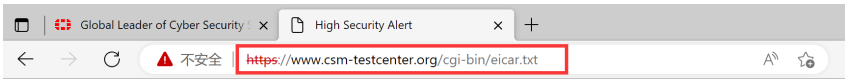
3. 在 FortiGate 上查看相应的 AV 日志。



4. 如果是通过 HTTPS 下载病毒，由于数据被 HTTPS 加密，如需要检测需要开启 SSL 深度检测，需要在虚拟接口对策略中调用 SSL 深度检测。

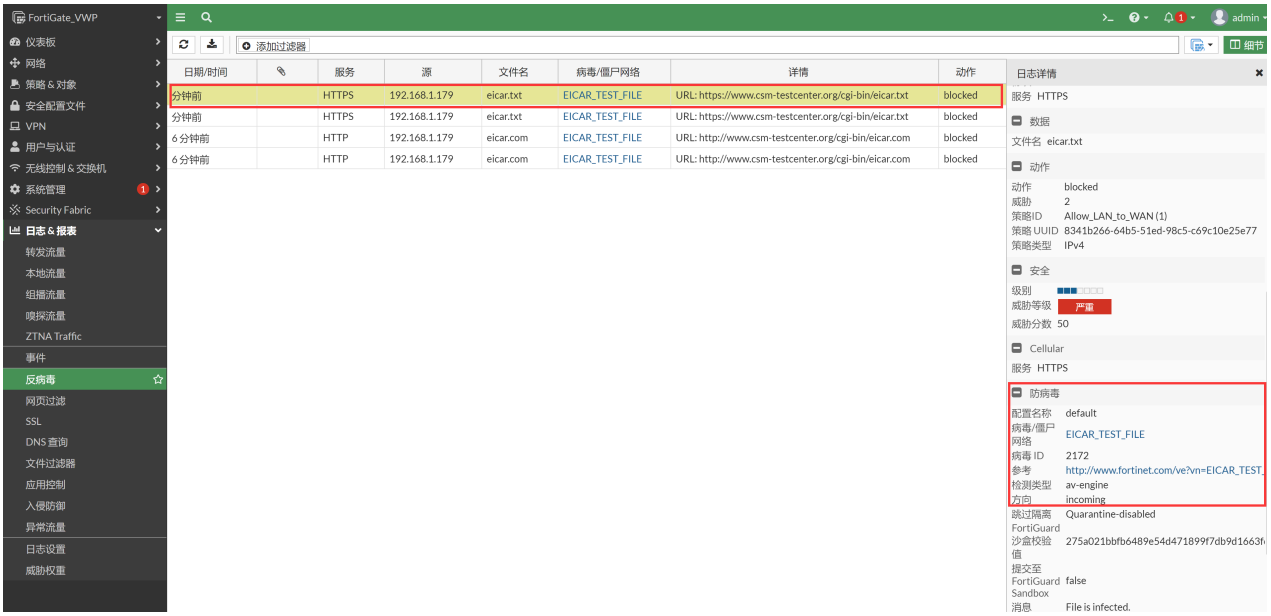


5. 测试通过 HTTPS 下载病毒文件（http://www.csm-testcenter.org/test?do ↗ = show & subdo = antimalware & test = content_types ），被 FortiGate 阻断。





6. 在 FortiGate 上查看相应的 AV 日志。

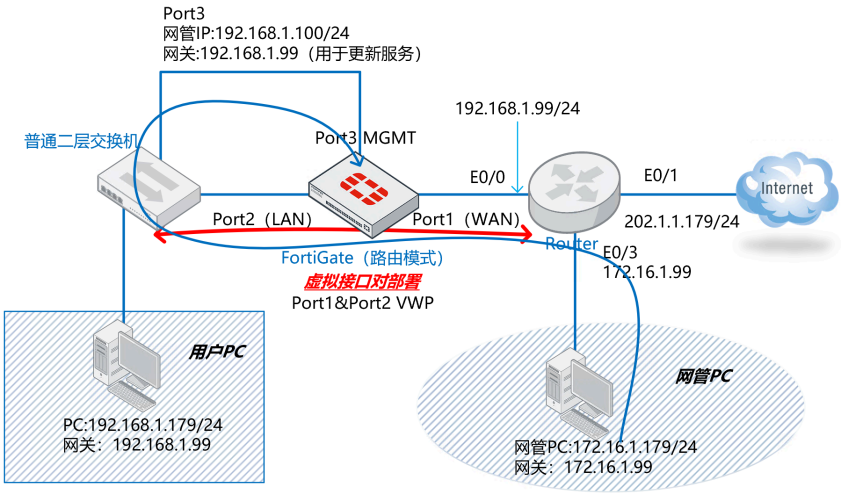


虚拟接口对的网络管理问题

网络需求

有一个特别的网管情况：防火墙是运行在路由模式下的，一旦开启了 VWP，VWP 相当于一个透明的接口对，可以透传流量，但是由于 FortiGate 本身需要被网管，而一旦网管的流量需要穿过 VWP 被网管的话，则就会遇到网管不了的情况，因为网管数据穿过 VWP 访问 FortiGate 的本机 IP，这样会话存在混乱，内核和转发将判断会话存在异常，会该网管的导致数据不通。

网络拓扑

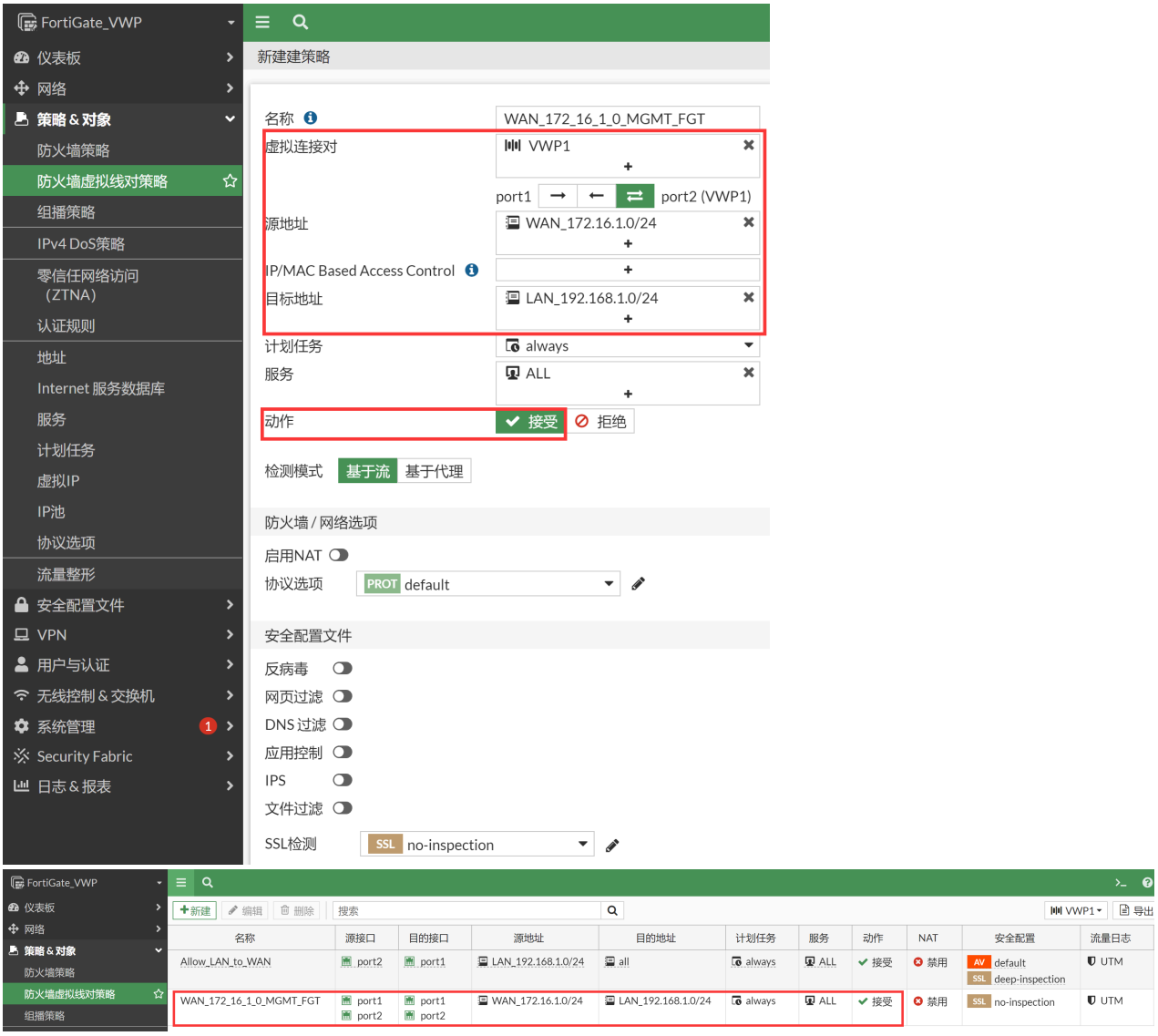


网管区域在 WAN 侧，而 FortiGate 的网管接口 Port3 在 LAN 侧，这种拓扑环境下网管有两种情况：

- 用户 PC 可以正常网管 FortiGate：用户 PC 192.168.1.179 由于和 FGT 的 Port3 网管接口同属于 LAN 侧流量，流量无需穿越 VWP，因此可以正常网管（推荐的网管规划和部署）。
- 网管 PC 无法网管 FortiGate：网管 PC 172.16.1.179 在 FortiGate 的 WAN 侧，发起去 FGT 的 Port3 网管接口的流量时候，正常流量会穿越 VWP 然后再达到 Port3 接口上，由于穿过 VWP 的时候发现目的 IP 是 Port3，这是一个访问 FGT 本机的报文，这样 FGT 在处理管理流量会存在问题。

问题复现

1. 我们来分析一下上述第二种异常的情况，为测试 WAN 的网管 PC 穿越 VWP 管理 FGT，需要新增一条允许 WAN 172.16.1.0/24 到 LAN 192.168.1.0/24 的策略。



2. 使用网管 PC 172.16.1.179 访问 FortiGate 的 Port3 192.168.1.100，在 FortiGate 上抓包查看，网管的数据在 Port3 接口 IN 之后就被丢弃了，PING/SSH/HTTPS 网管全部失败。

```
FortiGate_VWP # diagnose sniffer packet any "host 172.16.1.179" 4
interfaces=[any]
filters=[host 172.16.1.179]
1.216456 port1 in 172.16.1.179.51524 → 192.168.1.100.443: syn 3746720489
1.216496 port2 out 172.16.1.179.51524 → 192.168.1.100.443: syn 3746720489
1.259983 port3 in 172.16.1.179.51524 → 192.168.1.100.443: syn 3746720489

3.241235 port1 in 172.16.1.179 → 192.168.1.100: icmp: echo request
3.241320 port2 out 172.16.1.179 → 192.168.1.100: icmp: echo request
3.244140 port3 in 172.16.1.179 → 192.168.1.100: icmp: echo request

26.005468 port1 in 172.16.1.179.51533 → 192.168.1.100.22: syn 2838383157
26.005514 port2 out 172.16.1.179.51533 → 192.168.1.100.22: syn 2838383157
26.050901 port3 in 172.16.1.179.51533 → 192.168.1.100.22: syn 2838383157
```

3. debug flow 查看原因，由于数据从 port2 in → port1 out 穿越过 VWP 一次（创建了一个会话），然后又从 port3（root）发出一个回复报文，这个回复报文会匹配到之前创建会话，这样的会话接口存在混乱，会引起重放检测失败丢包。

```
diagnose debug flow filter addr 172.16.1.179
diagnose debug flow show function-name enable
diagnose debug flow trace start 100
diagnose debug enable

id=20085 trace_id=10 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=6, 172.16.1.179:51565-
>192.168.1.100:443) from port1. flag [S], seq 1925780389, ack 0, win 8192"
```

```
id=20085 trace_id=10 func=init_ip_session_common line=5654 msg="allocate a new session-000005b1"
id=20085 trace_id=10 func=br_fw_forward_handler line=539 msg="Allowed by Policy-2:"
id=20085 trace_id=10 func=__if_queue_push_xmit line=417 msg="send out via dev-port2, dst-mac-50:00:00:03:00:02"

id=20085 trace_id=11 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=6, 172.16.1.179:51565-
>192.168.1.100:443) from port3. flag [S], seq 1925780389, ack 0, win 8192"
id=20085 trace_id=11 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000005b1, original
direction"
id=20085 trace_id=11 func=vf_ip_route_input_common line=2591 msg="find a route: flag=80000000 gw-192.168.1.100 via root"

id=20085 trace_id=12 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=6, 192.168.1.100:443-
>172.16.1.179:51565) from local. flag [S.], seq 2874972251, ack 1925780390, win 14600"
id=20085 trace_id=12 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000005b1, reply direction"
id=20085 trace_id=12 func=ip_session_core_in line=6222 msg="outgoing dev changed:3→5 dir=reply, drop"

id=20085 trace_id=49 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 172.16.1.179:1-
>192.168.1.100:2048) from port1. type=8, code=0, id=1, seq=47."
id=20085 trace_id=49 func=init_ip_session_common line=5654 msg="allocate a new session-000005cc"
id=20085 trace_id=49 func=br_fw_forward_handler line=539 msg="Allowed by Policy-2:"
id=20085 trace_id=49 func=__if_queue_push_xmit line=417 msg="send out via dev-port2, dst-mac-50:00:00:03:00:02"

id=20085 trace_id=50 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 172.16.1.179:1-
>192.168.1.100:2048) from port3. type=8, code=0, id=1, seq=47."
id=20085 trace_id=50 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000005cc, original
direction"
id=20085 trace_id=50 func=vf_ip_route_input_common line=2591 msg="find a route: flag=80000000 gw-192.168.1.100 via root"

id=20085 trace_id=51 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 192.168.1.100:1-
>172.16.1.179:0) from local. type=0, code=0, id=1, seq=47."
id=20085 trace_id=51 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000005cc, reply direction"
id=20085 trace_id=51 func=ip_session_core_in line=6222 msg="outgoing dev changed:3→5 dir=reply, drop"
```

4. 值得注意的是，FGT（192.168.1.100）是可以主动去 ping 通 172.16.1.179 的，也就是说 FGT 是可以正常的更新 Fortiguard 服务和上互联网的。

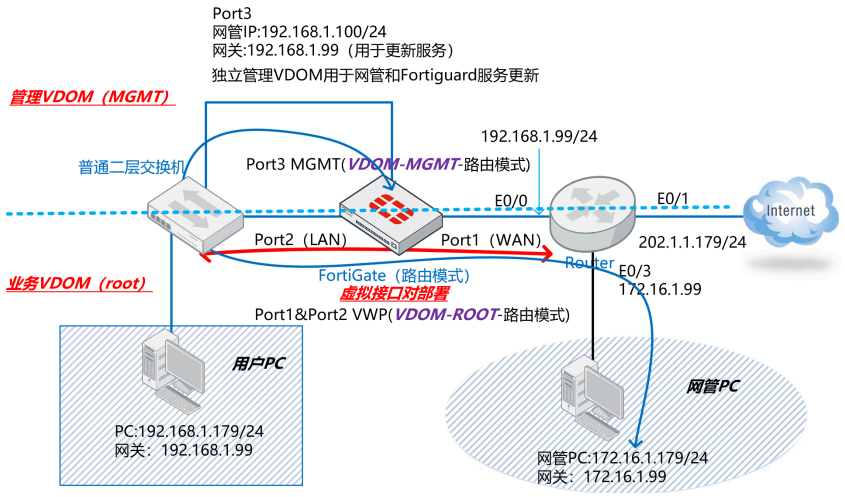
```
FortiGate_VWP # execute ping 172.16.1.179
PING 172.16.1.179 (172.16.1.179): 56 data bytes
64 bytes from 172.16.1.179: icmp_seq=0 ttl=127 time=7.8 ms
64 bytes from 172.16.1.179: icmp_seq=1 ttl=127 time=8.0 ms
64 bytes from 172.16.1.179: icmp_seq=2 ttl=127 time=7.4 ms
64 bytes from 172.16.1.179: icmp_seq=3 ttl=127 time=6.0 ms
64 bytes from 172.16.1.179: icmp_seq=4 ttl=127 time=8.8 ms

FortiGate_VWP # diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
55.845400 port3 out 192.168.1.100 → 172.16.1.179: icmp: echo request
55.848682 port2 in 192.168.1.100 → 172.16.1.179: icmp: echo request
55.848706 port1 out 192.168.1.100 → 172.16.1.179: icmp: echo request
55.851016 port1 in 172.16.1.179 → 192.168.1.100: icmp: echo reply
55.851030 port2 out 172.16.1.179 → 192.168.1.100: icmp: echo reply
55.855936 port3 in 172.16.1.179 → 192.168.1.100: icmp: echo reply
56.857317 port3 out 192.168.1.100 → 172.16.1.179: icmp: echo request
56.863493 port2 in 192.168.1.100 → 172.16.1.179: icmp: echo request
56.863515 port1 out 192.168.1.100 → 172.16.1.179: icmp: echo request
56.870026 port1 in 172.16.1.179 → 192.168.1.100: icmp: echo reply
56.870039 port2 out 172.16.1.179 → 192.168.1.100: icmp: echo reply
56.876321 port3 in 172.16.1.179 → 192.168.1.100: icmp: echo reply

id=20085 trace_id=93 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 192.168.1.100:2816-
>172.16.1.179:2048) from local. type=8, code=0, id=2816, seq=0."
id=20085 trace_id=93 func=init_ip_session_common line=5654 msg="allocate a new session-000013a5"

id=20085 trace_id=94 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 192.168.1.100:2816-
>172.16.1.179:2048) from port2. type=8, code=0, id=2816, seq=0."
id=20085 trace_id=94 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000013a5, original
direction"
id=20085 trace_id=94 func=br_fw_forward_dirty_handler line=318 msg="Found a reflect session: pro=1,
192.168.1.100/2816⇒172.16.1.179/8, dev=4→3"
id=20085 trace_id=94 func=__if_queue_push_xmit line=417 msg="send out via dev-port1, dst-mac-aa:bb:cc:00:20:00"

id=20085 trace_id=95 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 172.16.1.179:2816-
>192.168.1.100:0) from port1. type=0, code=0, id=2816, seq=0."
```

```
id=20085 trace_id=95 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000013a5, reply direction"
id=20085 trace_id=95 func=__if_queue_push_xmit line=417 msg="send out via dev-port2, dst-mac-50:00:00:03:00:02"

id=20085 trace_id=96 func=print_pkt_detail line=5494 msg="vd-root:0 received a packet(proto=1, 172.16.1.179:2816-
>192.168.1.100:0) from port3. type=0, code=0, id=2816, seq=0."
id=20085 trace_id=96 func=resolve_ip_tuple_fast line=5569 msg="Find an existing session, id-000013a5, reply direction"
id=20085 trace_id=96 func=vf_ip_route_input_common line=2591 msg="find a route: flag=80000000 gw-192.168.1.100 via root"
```

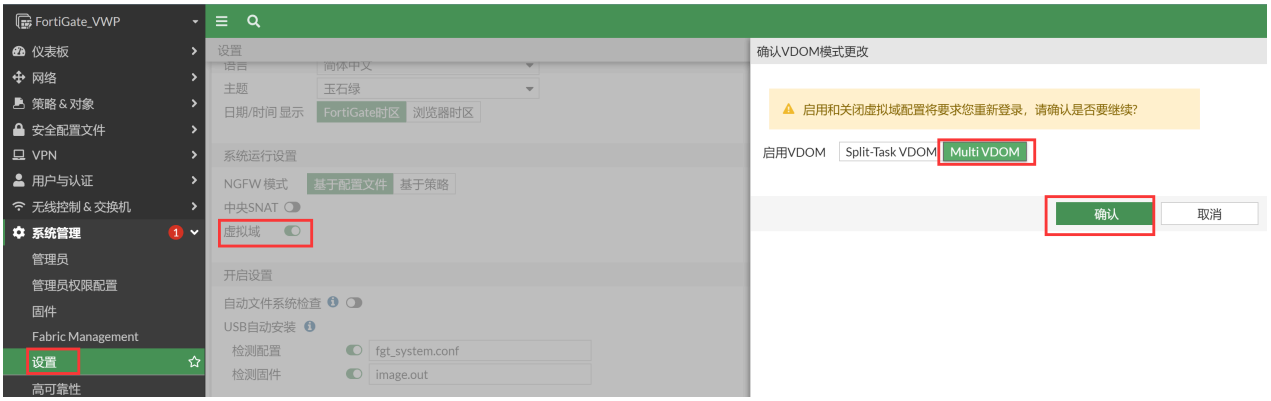
5. 以上就是 VWP 环境下的网管问题的故障现象，单向通信，跨 VWP 的网管流量无法正常通信。

解决办法

1. 从设计上避免这种跨 VWP 的网管 FGT 自身的流量，不跨 VWP 的网管是正常的。
2. 如果无法避免这样的流量，则使用 VDOM 技术将网管和业务二者独立开来，创建一个管理的 VDOM（MGMT）专门用于 MGMT，与 VWP 的所属的 VDOM（ROOT）完全独立运行即可，如下图所示。



3. 开启 FortiGate 的 VDOM 功能。



对应命令行：

```
FortiGate_VWP # config system global
FortiGate_VWP (global) # set vdom-mode multi-vdom
FortiGate_VWP (global) # end
You will be logged out for the operation to take effect
Do you want to continue? (y/n) y
exit
```

4. 创建 MGMT VDOM。



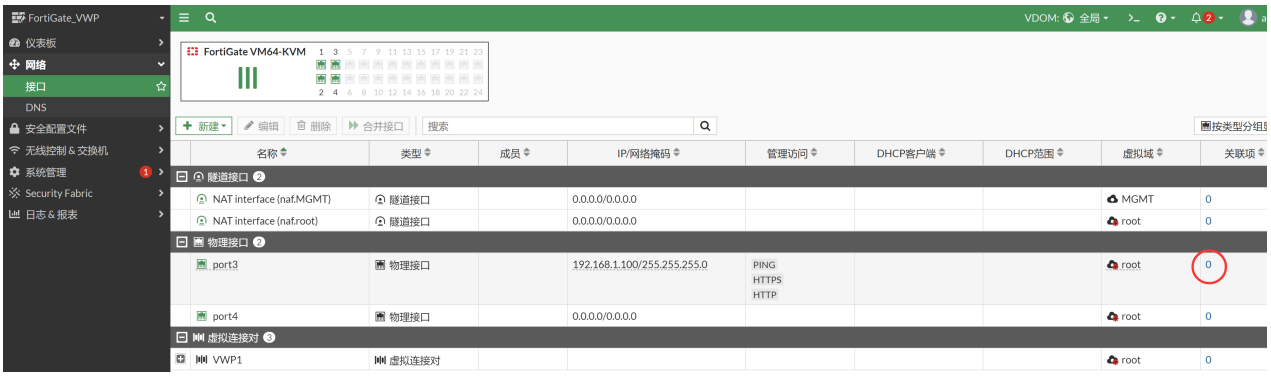




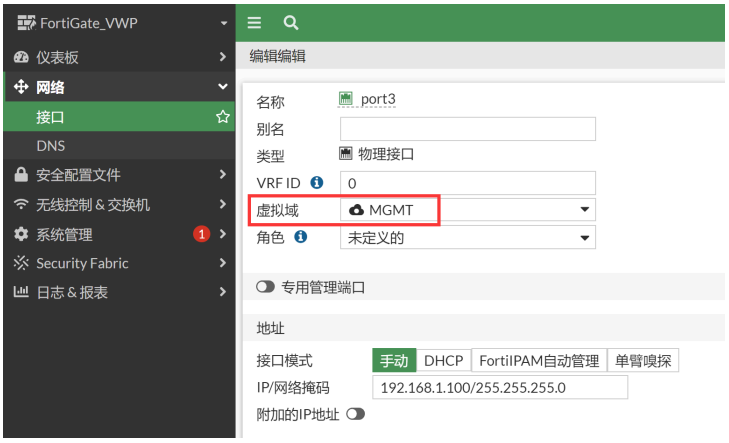
5. 准备将 port3 加入 MGMT VDOM 中，将 port3 加入 MGMT VDOM 之前需要去掉相关的配置关联项，也就是说 port3 不能在 root 下有相关的配置，比如此时有静态路由的配置，则需要删掉该路由。



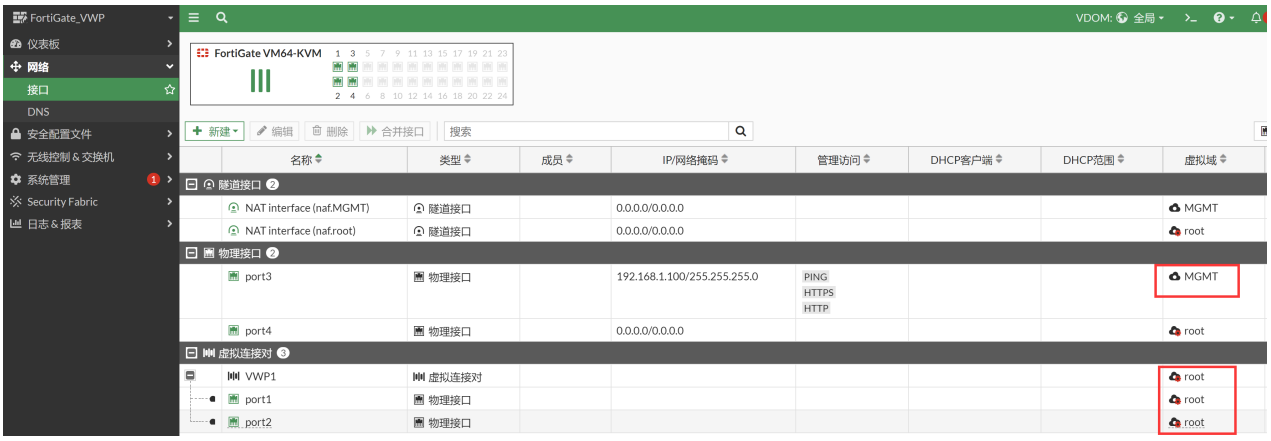




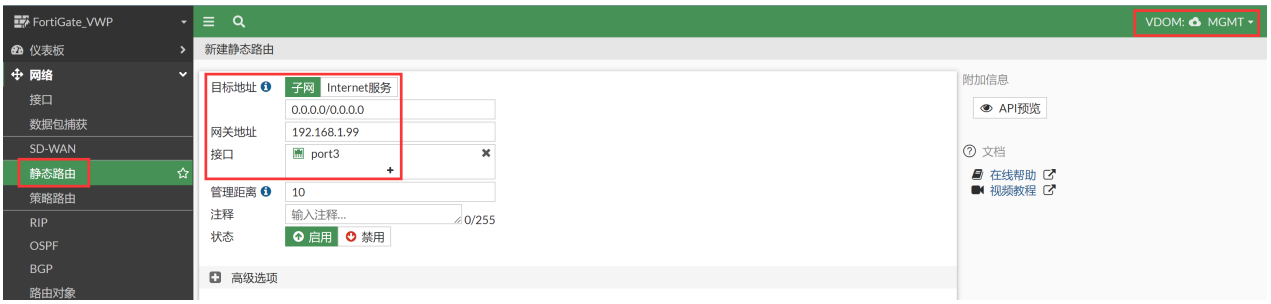
6. 此时可以将 port3 加入 MGMT VDOM。



7. 这样相当于 port3 属于 VDOM MGMT，而 port1 和 port2 属于 VDOM ROOT，二者完全独立，这样就可以解决会话混乱的问题了。



8. 再进入 MGMT VDOM，将管理接口的默认路由加上去。



9. 将 MGMT VDOM 设置为管理 VDOM，用于更新 Fortiguard（这一项很重要，如果管理 VDOM 是 root 则不需要此操作，默认的管理 VDOM 是 root）。

```
FortiGate_VWP # config global
FortiGate_VWP (global) # config system global
FortiGate_VWP (global) # set management-vdom MGMT
FortiGate_VWP (global) # end
FortiGate_VWP (global) # end
```

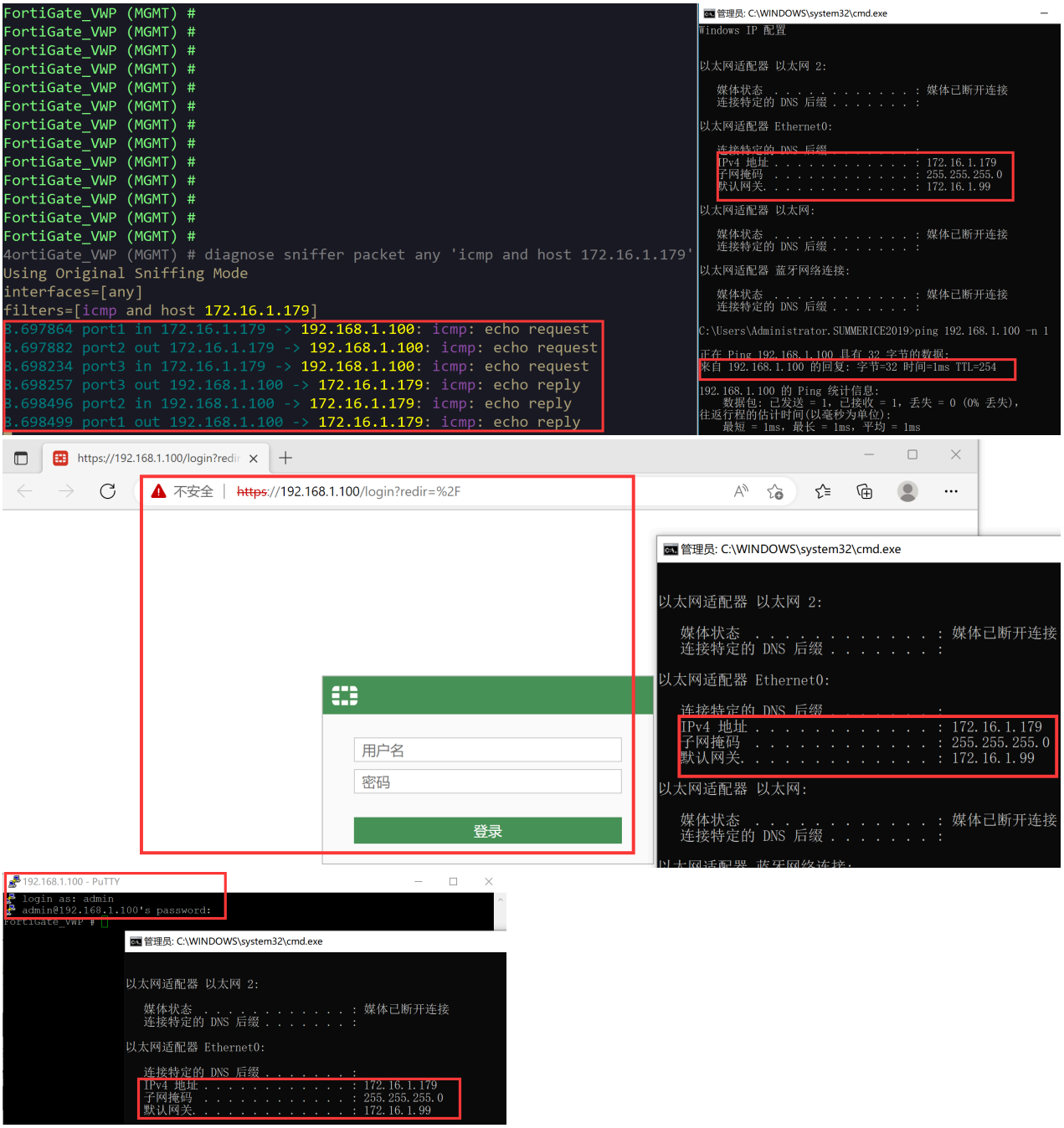
效果验证

1. 再次抓取网管 PC 跨 VWP 去管理 FGT 的网管流量。

```
FortiGate_VWP (root) # diagnose sniffer packet any "host 172.16.1.179" 4
interfaces=[any]
filters=[host 172.16.1.179]
6.911052 port1 in 172.16.1.179 → 192.168.1.100: icmp: echo request
6.911092 port2 out 172.16.1.179 → 192.168.1.100: icmp: echo request
6.914638 port3 in 172.16.1.179 → 192.168.1.100: icmp: echo request
6.914682 port3 out 192.168.1.100 → 172.16.1.179: icmp: echo reply
6.918603 port2 in 192.168.1.100 → 172.16.1.179: icmp: echo reply
```

```
6.918621 port1 out 192.168.1.100 → 172.16.1.179: icmp: echo reply

48.683279 port1 in 172.16.1.179.49159 → 192.168.1.100.80: syn 3128124765
48.683385 port2 out 172.16.1.179.49159 → 192.168.1.100.80: syn 3128124765
48.684797 port1 in 172.16.1.179.49160 → 192.168.1.100.80: syn 2745446281
48.684845 port2 out 172.16.1.179.49160 → 192.168.1.100.80: syn 2745446281
48.686611 port1 in 172.16.1.179.49161 → 192.168.1.100.443: syn 2797650417
48.686655 port2 out 172.16.1.179.49161 → 192.168.1.100.443: syn 2797650417
48.712056 port3 in 172.16.1.179.49159 → 192.168.1.100.80: syn 3128124765
48.712230 port3 out 192.168.1.100.80 → 172.16.1.179.49159: syn 1287655075 ack 3128124766
48.713907 port3 in 172.16.1.179.49160 → 192.168.1.100.80: syn 2745446281
48.713951 port3 out 192.168.1.100.80 → 172.16.1.179.49160: syn 1003036694 ack 2745446282
48.715772 port2 in 192.168.1.100.80 → 172.16.1.179.49159: syn 1287655075 ack 3128124766
48.715790 port1 out 192.168.1.100.80 → 172.16.1.179.49159: syn 1287655075 ack 3128124766
48.717463 port2 in 192.168.1.100.80 → 172.16.1.179.49160: syn 1003036694 ack 2745446282
48.717478 port1 out 192.168.1.100.80 → 172.16.1.179.49160: syn 1003036694 ack 2745446282
48.717537 port1 in 172.16.1.179.49159 → 192.168.1.100.80: ack 1287655076
48.717547 port2 out 172.16.1.179.49159 → 192.168.1.100.80: ack 1287655076
48.719277 port1 in 172.16.1.179.49160 → 192.168.1.100.80: ack 1003036695
48.719291 port2 out 172.16.1.179.49160 → 192.168.1.100.80: ack 1003036695

232.060671 port1 in 172.16.1.179.49158 → 192.168.1.100.22: syn 446444854
232.060794 port2 out 172.16.1.179.49158 → 192.168.1.100.22: syn 446444854
232.069282 port3 in 172.16.1.179.49158 → 192.168.1.100.22: syn 446444854
232.069473 port3 out 192.168.1.100.22 → 172.16.1.179.49158: syn 1275691481 ack 446444855
232.072262 port2 in 192.168.1.100.22 → 172.16.1.179.49158: syn 1275691481 ack 446444855
232.072278 port1 out 192.168.1.100.22 → 172.16.1.179.49158: syn 1275691481 ack 446444855
232.074496 port1 in 172.16.1.179.49158 → 192.168.1.100.22: ack 1275691482
232.074512 port2 out 172.16.1.179.49158 → 192.168.1.100.22: ack 1275691482
232.084068 port3 in 172.16.1.179.49158 → 192.168.1.100.22: ack 1275691482
232.106501 port3 out 192.168.1.100.22 → 172.16.1.179.49158: psh 1275691482 ack 446444855
232.111292 port2 in 192.168.1.100.22 → 172.16.1.179.49158: psh 1275691482 ack 446444855
232.111311 port1 out 192.168.1.100.22 → 172.16.1.179.49158: psh 1275691482 ack 446444855
232.129332 port1 in 172.16.1.179.49158 → 192.168.1.100.22: psh 446444855 ack 1275691499
232.129361 port2 out 172.16.1.179.49158 → 192.168.1.100.22: psh 446444855 ack 1275691499
232.132511 port3 in 172.16.1.179.49158 → 192.168.1.100.22: psh 446444855 ack 1275691499
```

2. 跨 VWP 的网管工作正常。



总结

开启了 VWP 的 FGT 网管尽量不要跨 VWP 进行网管，如果不能避免这样的部署就通过上述所示的 VDOM 去解决。

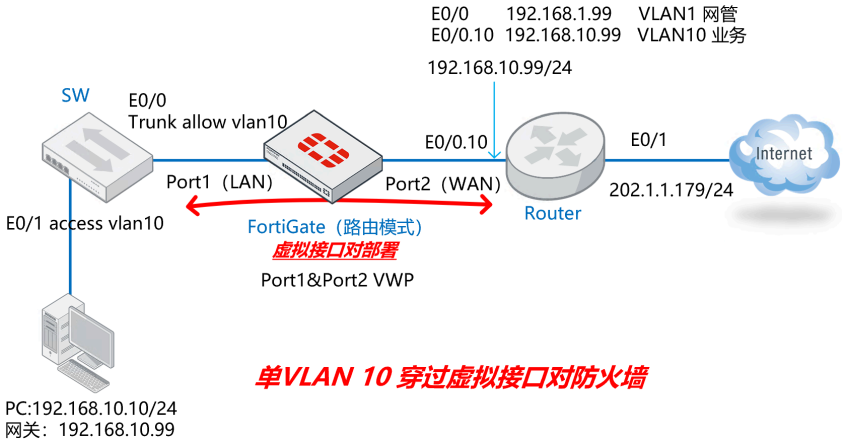
VLAN 与虚拟接口对

网络需求

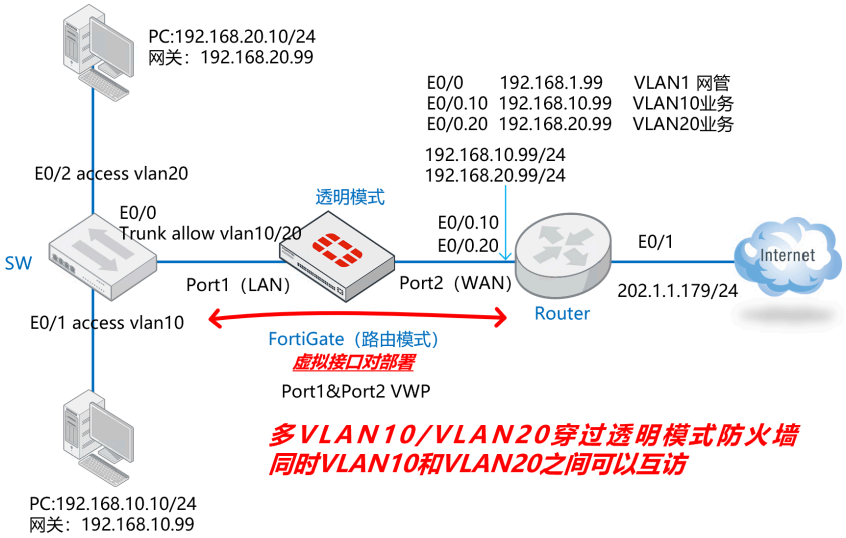
内部有两个 VLAN（trunk 环境），网关在路由器上，防火墙工作在虚拟接口对方式部署在核心交换与核心路由之间，允许两个 VLAN 访问外网，同时两个 VLAN 之间需要互访，并且对这两个 VLAN 进行保护，开启反病毒过滤。

网络拓扑

拓扑 1：单 VLAN 透传



拓扑 2：多 VLAN 透传，VLAN 之间互访



配置要点

拓扑 1：单 VLAN 透传

- SW 交换机/Internet_R 路由器的基础配置
- 配置虚拟接口对（Port1 & Port2）
- 配置安全策略允许内网 PC（VLAN10 内主机）访问互联网

拓扑 2：多 VLAN 透传，VLAN 之间互访

- SW 交换机/Internet_R 路由器的基础配置
- 配置虚拟接口对（Port1 & Port2）
- 配置安全策略允许内网 PC（VLAN10 和 VLAN20 内主机）访问互联网

配置步骤

拓扑 1：单 VLAN 透传

1. SW 交换机/Internet_R 路由器的基础配置。

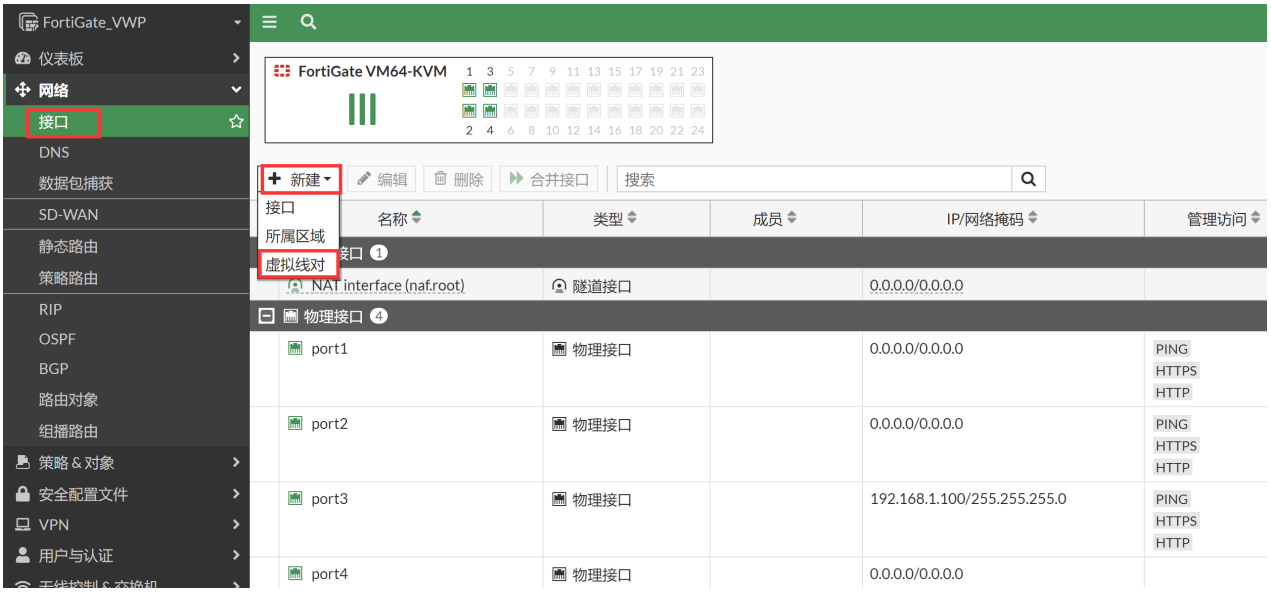
   SW：

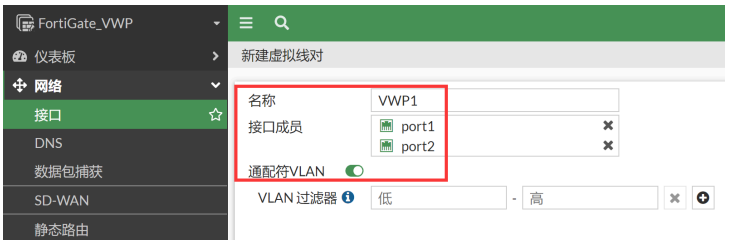
   ```
   interface Ethernet0/0
   switchport trunk allowed vlan 1,10
   switchport trunk encapsulation dot1q
   switchport mode trunk
   !
   interface Ethernet0/1
   switchport access vlan 10
   switchport mode access
   ```

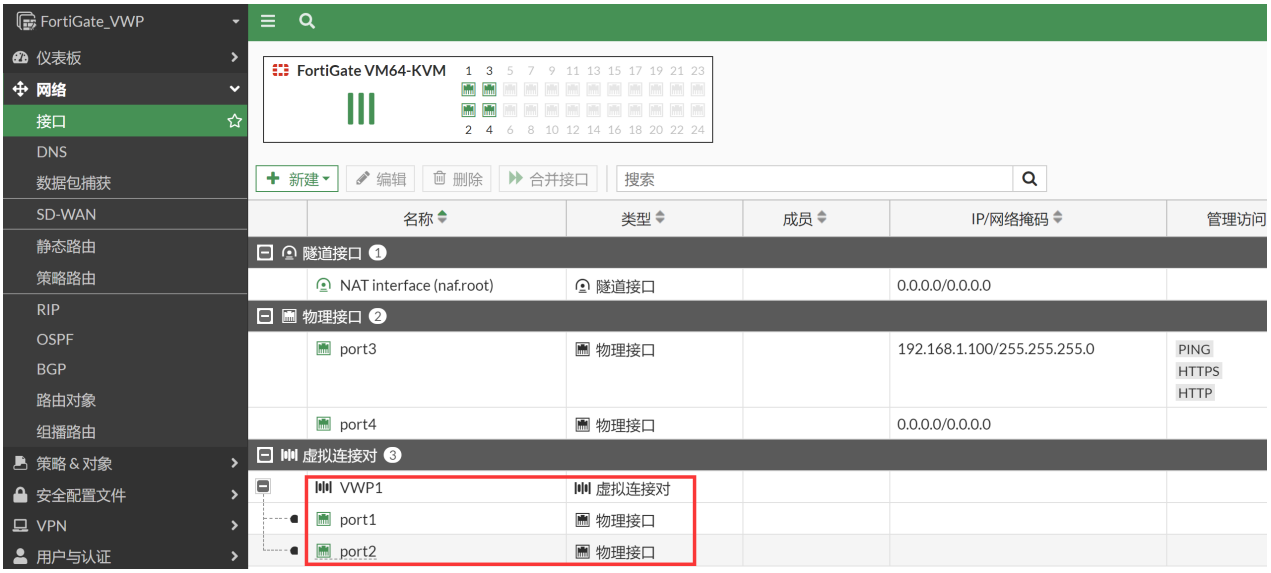
   Internet_R：

   ```
   hostname Internet_R
   !
   interface Ethernet0/0
   ip address 192.168.1.99 255.255.255.0
   no shutdown
   ip nat inside
   !
   interface Ethernet0/0.10
   encapsulation dot1Q 10
   ip address 192.168.10.99 255.255.255.0
   no shutdown
   ip nat inside
   !
   interface Ethernet0/1
   ip address 202.100.1.179 255.255.255.0
   no shutdown
   ip nat outside
   !
   ip route 0.0.0.0 0.0.0.0 202.100.1.192
   !
   access-list 101 permit ip any any
   ip nat inside source list 101 interface Ethernet0/1 overload
   ```

2. 配置虚拟接口对（Port1 & Port2），防火墙处于路由模式的情况下，直接创建虚拟接口对，将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据。







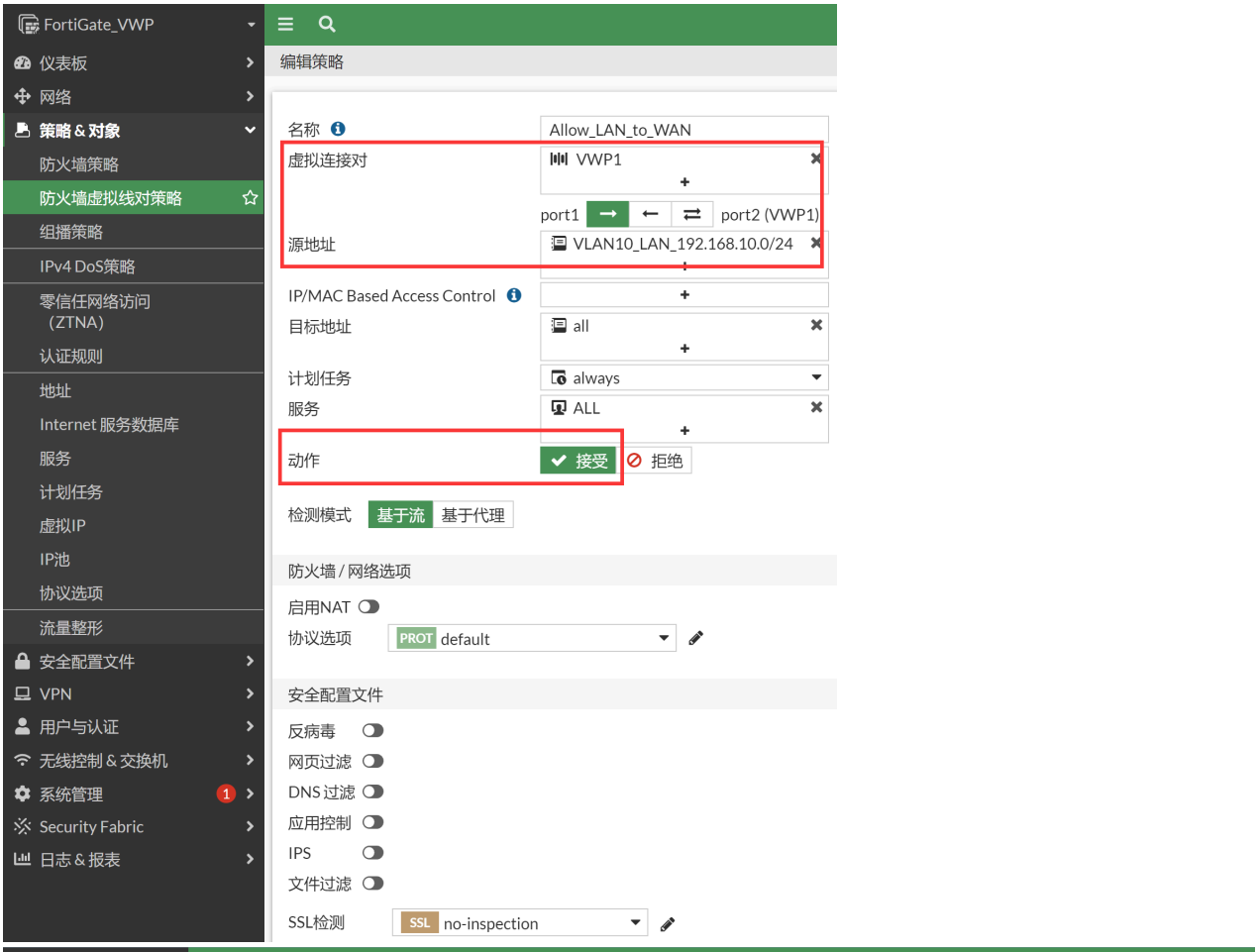
3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需要防火墙策略放通），对应的 CLI 如下。

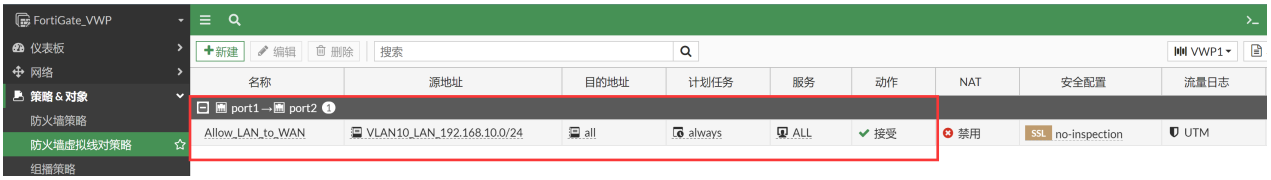
```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable     //这个命令很关键，识别VLAN-TAG，VWP模式下无需再到本地创建对应的VLAN接口和Forwar-Domain了，大大的简化了策略的配置量
    next
end
```

4. 配置虚拟接口对的策略：允许 LAN（Port2，VLAN10 内网主机）到 WAN（port1）的上网流量穿过 FGT，策略 & 对象的位置会多出“防火墙虚拟线对策略”（如果没有看到，则刷新页面或注销管理登陆，重新登陆 FGT 即可看到）。



5. 右上角选择 VWP1 线对，方向选择 port1 → port2（也可选择双向的方向），新建 VWP 策略（虚拟连接对可以选择多个，在同一条相同动作的策略中进行多接口控制）。





> **重要**
>
> 只需要配置 VALN10 网段到 WAN 的策略即可实现，配置很简单，可以忽略 VLAN - TAG 接口和 Forward-Domain 等透明模式下需要考虑的因素。

拓扑 2：多 VLAN 透传，VLAN 之间互访

1. SW 交换机/Internet_R 路由器的基础配置。

SW：

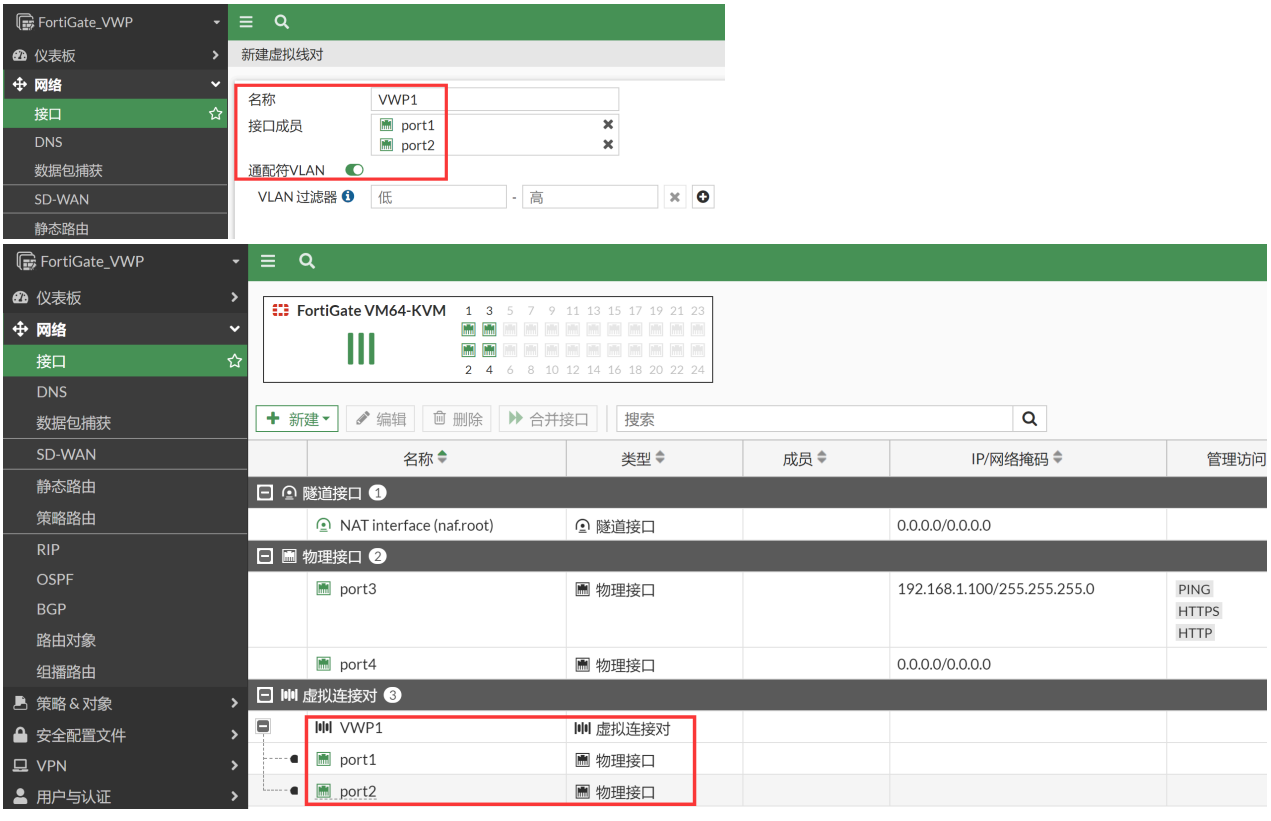
```
interface Ethernet0/0
switchport trunk allowed vlan 1,10,20
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface Ethernet0/1
```

```
switchport access vlan 10
switchport mode access
!
interface Ethernet0/2
switchport access vlan 20
switchport mode access
```

Internet_R:

```
hostname Internet_R
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

2. 配置虚拟接口对（Port1 & Port2），防火墙处于路由模式的情况下，直接创建虚拟接口对，将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据。



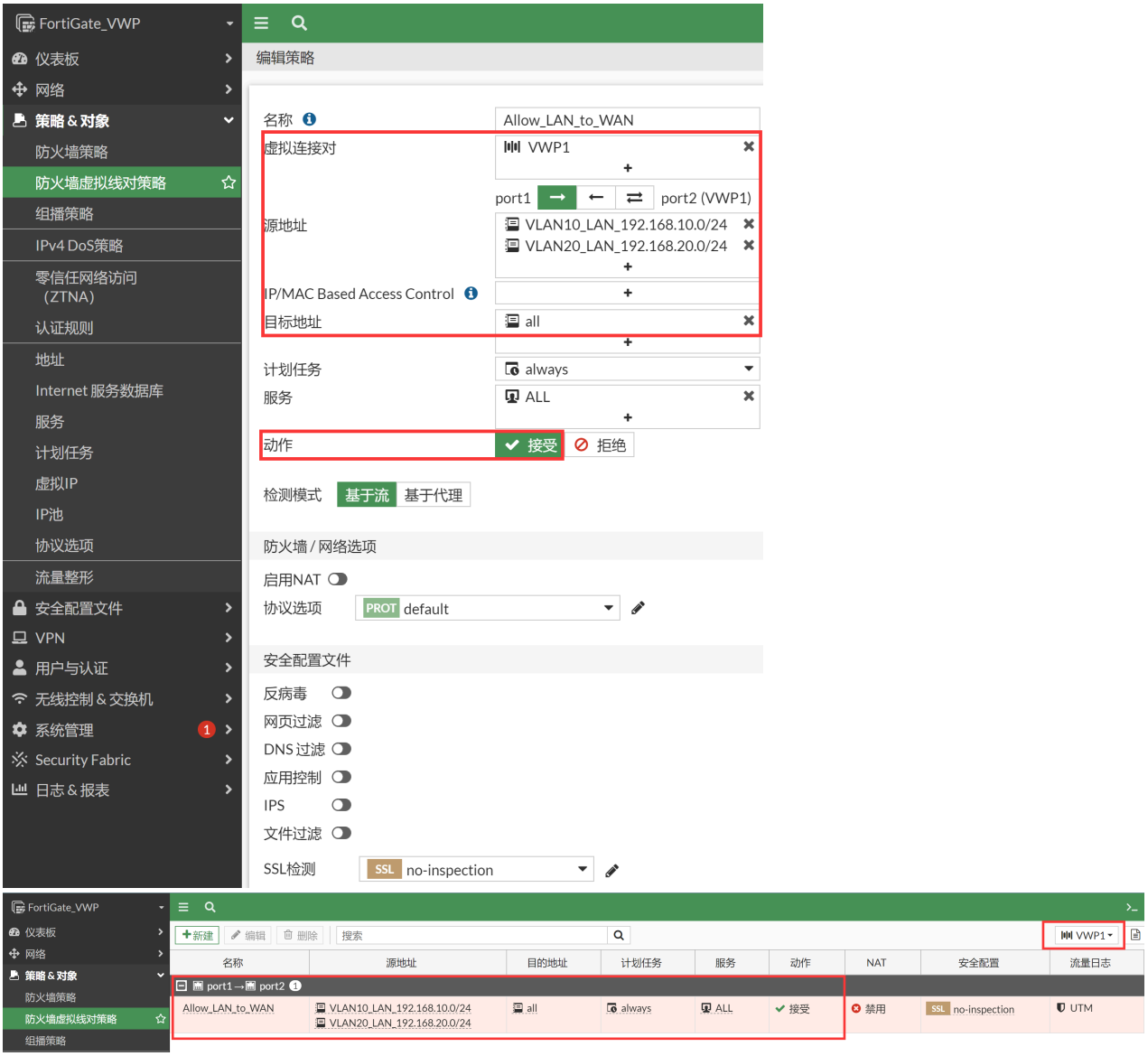
3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需要防火墙策略放通），对应的 CLI 如下。

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable     //这个命令很关键，识别VLAN-TAG，VWP模式下无需再到本地创建对应的VLAN接口和Forwar-Domain了，大大的简化了策略的配置量
    next
end
```

4. 配置虚拟接口对的策略：允许 LAN（port1）到 WAN（port2）的上网流量穿过 FGT，策略 & 对象的位置会多出“防火墙虚拟线对策略”（如果没有看到，则刷新页面或注销管理登陆，重新登陆 FGT 即可看到）。



5. 右上角选择 VWP1 线对，方向选择 port1 → port2（也可选择双向的方向），新建 VWP 策略（虚拟连接对可以选择多个，在同一条相同动作的策略中进行多接口控制）。



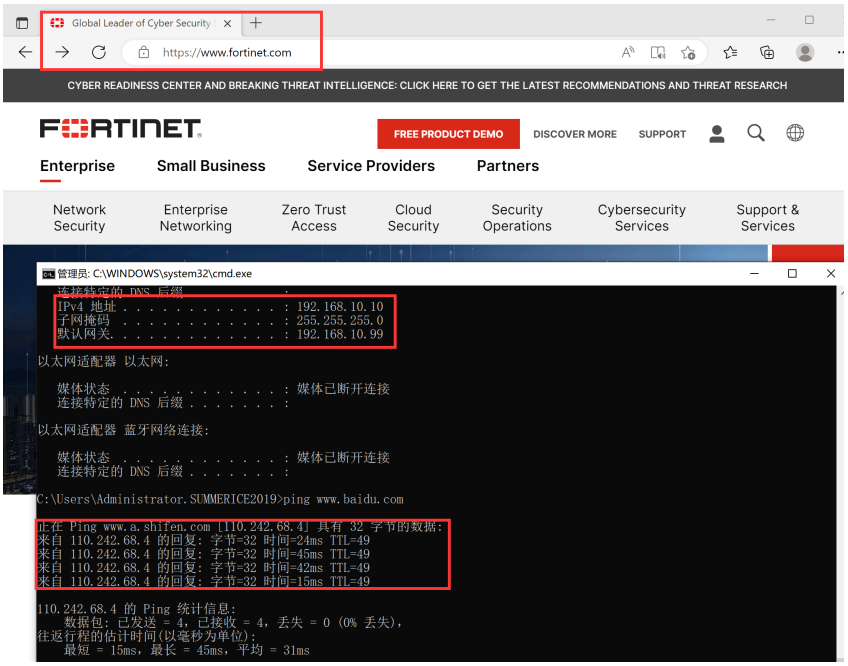
> **重要**
>
> 其实配置上上述两条策略数据就已经通了，因为两条策略的目的 IP 都是 any，包括了 192.168.20.0/24 和 192.168.10.0/24，如果要明细配置策略则需要将策略中的目标地址改为明细的地址对象，并放置到策略的最前面，以便优先匹配。

6. 只需要配置 VALN10/VLAN20 网段相互之间的策略即可，配置很简单，可以忽略 VLAN - TAG 接口和 Forward-Domain 等透明模式下需要考虑的因素。配置相对很简单。

效果验证

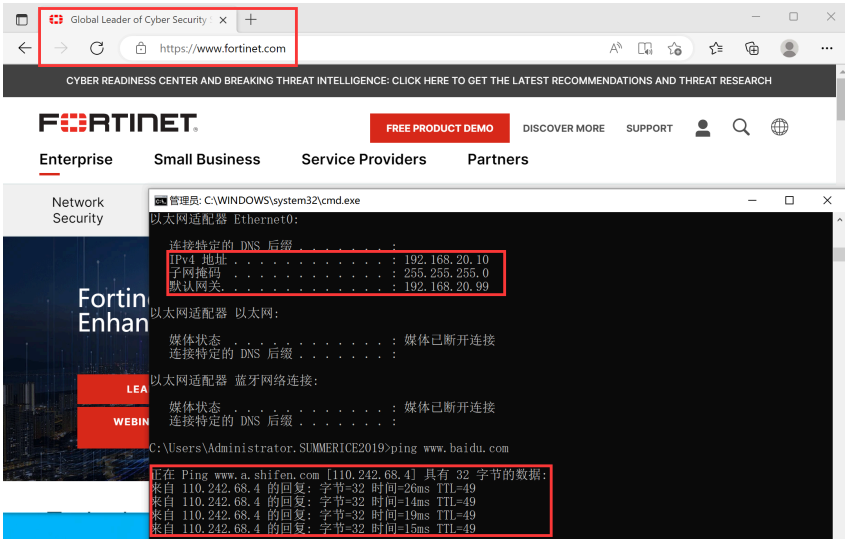
拓扑 1：单 VLAN 透传

VLAN10 内的 PC 通过透明模式的 FortiGate 访问互联网，可以正常访问。

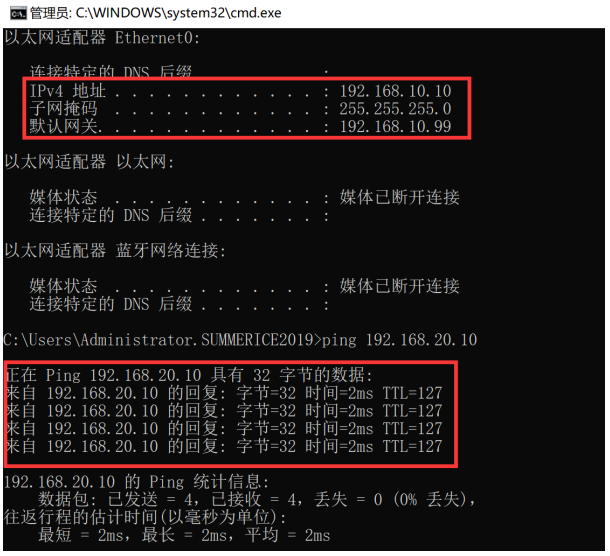


拓扑 2：多 VLAN10/20 透传，VLAN 之间互访

1. VLAN20 内的 PC 访问互联网，可以正常访问。



2. VLAN10 内的 PC 访问 VLAN20 的 PC，可以正常访问。



3. 抓包查看 VLAN10 内的 PC 访问 VLAN20 的 PC 的流量，注意入方向携带 VLAN - TAG 的数据 sniffer 无法正常的显示明细数据。

```
FortiGate_VWP # diagnose sniffer packet any 'none' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[none]
21.060512 port1 in 802.1Q vlan#10 P0
21.060542 port2 out 192.168.10.10 → 192.168.20.10: icmp: echo request
21.062168 port2 in 802.1Q vlan#20 P0
21.062178 port1 out 192.168.10.10 → 192.168.20.10: icmp: echo request
21.065095 port1 in 802.1Q vlan#20 P0
21.065106 port2 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
21.065540 port2 in 802.1Q vlan#10 P0
21.065546 port1 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
```

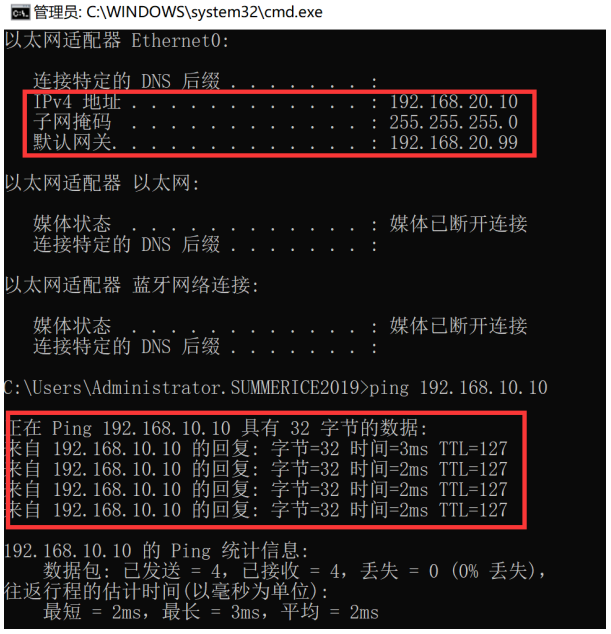
4. debug flow 以上流量。

```
id=20085 trace_id=1 func=print_pkt_detail line=5871 msg="vd-root:0 received a packet(proto=1, 192.168.10.10:1-
>192.168.20"
id=20085 trace_id=1 func=init_ip_session_common line=6050 msg="allocate a new session-00000821, tun_id=0.0.0.0"
id=20085 trace_id=1 func=br_fw_forward_handler line=573 msg="Allowed by Policy-1:"
id=20085 trace_id=1 func=__if_queue_push_xmit line=391 msg="send out via dev-port2, dst-mac-aa:bb:cc:00:20:00"
id=20085 trace_id=2 func=print_pkt_detail line=5871 msg="vd-root:0 received a packet(proto=1, 192.168.10.10:1-
>192.168.20.10:2048) tun_id"
id=20085 trace_id=2 func=resolve_ip_tuple_fast line=5957 msg="Find an existing session, id-00000821, original direction"
id=20085 trace_id=2 func=br_fw_forward_dirty_handler line=274 msg="auxiliary ses proto=1 dev=4→3
192.168.10.10/1⇒192.168.20.10/8"
id=20085 trace_id=2 func=npu_handle_session44 line=1183 msg="Trying to offloading session from port2 to port1,
skb.npu_flag=00000400 ses."
id=20085 trace_id=2 func=br_fw_forward_dirty_handler line=297 msg="state=00008200, state2=00000000, npu_state=00000100"
id=20085 trace_id=2 func=__if_queue_push_xmit line=391 msg="send out via dev-port1, dst-mac-00:0c:29:ae:55:70"
id=20085 trace_id=3 func=print_pkt_detail line=5871 msg="vd-root:0 received a packet(proto=1, 192.168.20.10:1-
>192.168.10.10:0) tun_id=0."
id=20085 trace_id=3 func=resolve_ip_tuple_fast line=5957 msg="Find an existing session, id-00000821, reply direction"
id=20085 trace_id=3 func=br_fw_forward_dirty_handler line=274 msg="auxiliary ses proto=1 dev=4→3
192.168.10.10/1⇒192.168.20.10/8"
id=20085 trace_id=3 func=npu_handle_session44 line=1183 msg="Trying to offloading session from port1 to port2,
skb.npu_flag=00000000 ses."
id=20085 trace_id=3 func=br_fw_forward_dirty_handler line=297 msg="state=00008200, state2=00000000, npu_state=00000100"
id=20085 trace_id=3 func=__if_queue_push_xmit line=391 msg="send out via dev-port2, dst-mac-aa:bb:cc:00:20:00"
id=20085 trace_id=4 func=print_pkt_detail line=5871 msg="vd-root:0 received a packet(proto=1, 192.168.20.10:1-
>192.168.10.10:0) tun_id=0."
id=20085 trace_id=4 func=resolve_ip_tuple_fast line=5957 msg="Find an existing session, id-00000821, reply direction"
id=20085 trace_id=4 func=npu_handle_session44 line=1183 msg="Trying to offloading session from port2 to port1,
skb.npu_flag=00000400 ses."
id=20085 trace_id=4 func=br_fw_forward_dirty_handler line=297 msg="state=00008200, state2=00000000, npu_state=00000100"
id=20085 trace_id=4 func=__if_queue_push_xmit line=391 msg="send out via dev-port1, dst-mac-00:0c:29:57:97:29"
```

5. 相关会话信息，可以看到 reflect session 的数量为 1，为反向流量会话。

```
session info: proto=1 proto_state=00 duration=2 expire=57 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=120/2/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 50/0 rx speed(Bps/kbps): 50/0
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 192.168.10.10:1->192.168.20.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.20.10:1->192.168.10.10:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=14730 auth_info=0 chk_client_info=0 vd=0
serial=0000096d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
vlanid=10 npu_state=0x000100
no_ofld_reason:  npu-flag-off
ext_header_type=0x22:0x22
reflect info 0:
dev=4->3/3->4
npu_state=0x000100
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
total reflect session num: 1
total session 1
```

6. 使用 VLAN20 内网 PC 访问 VLAN10 内网 PC，也可以正常访问（具体抓包和 debug 过程和上述一致）。



补充说明

1. 关于 VLAN-forward 参数，默认在 VWP 的环境下，成员接口的 vlanforward 是打开的。

```
onfig system interface
    edit "port1"
        set vdom "root"
        set vlanforward enable    //默认开启
        set type physical
        set alias "WAN"
        set snmp-index 1
    next
end
FortiGate_VWP (port1) # next
FortiGate_VWP (interface) # edit port2
FortiGate_VWP (port2) # show
config system interface
    edit "port2"
```

```
        set vdom "root"
        set vlanforward enable    //默认开启
        set type physical
        set alias "LAN"
        set snmp-index 2
    next
end
```

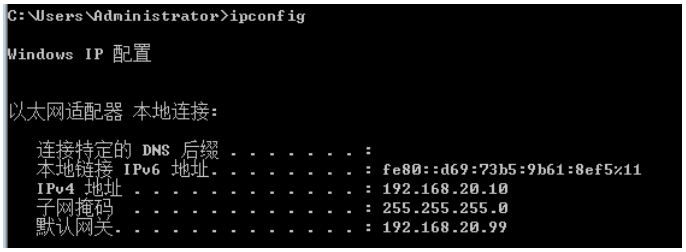
2. 但是如果已经开启了 set wildcard-vlan enable 的情况下，set vlanforward enable 是不生效的。set wildcard-vlan enable 的优先级比 vlanforward 要高，都开启的情况下，FortiGate 会优先使用 set wildcard-vlan enable。

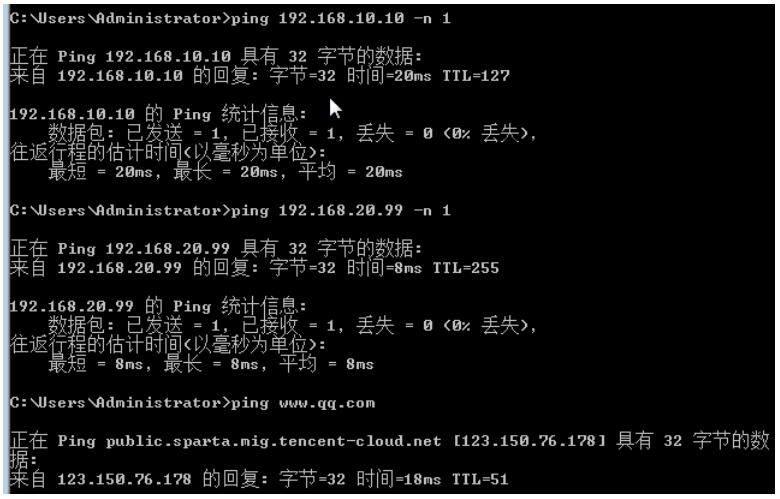
总结

1. 开启了 wildcard-vlan enable，vlanforward 不生效，wildcard-vlan 优先，识别 VLAN - TAG，并上送到 FGT 的 CPU 上去策略处理。这是推荐的配置。
2. 如果关闭了 wildcard-vlan disable，则 vlanforward 生效，则携带了 vlan-tag 的数据会直接穿透 VWP 接口，不再匹配策略，直接 bypass 可通（不推荐这样使用，流量不可见也不可控，这不是安全设备想要的结果，具体参考网络管理 → 透明模式 → 传统透明模式 → VLAN 与透明模式 → vlanforward ↗ 章节）。
3. 具体现象可以尝试关闭 wildcard-vlan，清空 VWP 策略。

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan disable  //关闭wildcard
    next
end
config system interface
    edit "port1"
        set vdom "root"
        set vlanforward enable
        set type physical
        set alias "WAN"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set vlanforward enable
        set type physical
        set alias "LAN"
        set snmp-index 2
    next
end

FortiGate_VWP # config firewall policy
FortiGate_VWP (policy) # purge    （清除策略，让设备没有任何策略）
This operation will clear all table!
Do you want to continue? (y/n)y
FortiGate_VWP (policy) # end
```

4. 具体的数据转发现象呢？我们会发现，所有的携带了 VLAN - TAG 的数据都是通的，VLAN10/20 的 PC 可以上网，VLAN10 和 VLAN20 的 PC 相互可通信。但是在 FGT 上看不到任何的信息，抓不到任何的包，直接 bypass 掉携带了 VLAN - TAG 的数据。VWP 不处理携带了 VLAN - TAG 的数据。而不携带 VLAN - TAG 的数据（VLAN1）将会上送到 FGT 的 CPU 进行策略匹配处理。携带了 VLAN - TAG 的数据被 bypass 掉了，直接可通。（vlan1 的不携带 vlan-tag 的数据会上送到 CPU 进行处理，需要策略放通才可以正常通信。）





5. 抓不到相应的报文。但是数据是可通的。

```
FortiGate_VWP # diagnose sniffer packet any "host 192.168.20.10" 4
interfaces=[any]
filters=[host 192.168.20.10]
```

6. 实际上可以抓到 VLAN - TAG 的数据，只是 FGT 由于开启了 vlanforward，不处理携带了 VLAN - TAG 的数据，直接 bypass 掉了（不推荐这样的部署）。

```
FortiGate_VWP # diagnose sniffer packet any "none" 4
interfaces=[any]
filters=[none]
1.189503 port2 in 802.1Q vlan#20 P0
1.475067 port2 in 802.1Q vlan#20 P0
1.475081 port1 out 802.1Q vlan#20 P0
1.475749 port1 in 802.1Q vlan#20 P0
1.475756 port2 out 802.1Q vlan#20 P0
1.568623 port2 in 802.1Q vlan#10 P0
1.568668 port1 out 802.1Q vlan#10 P0
2.211987 port2 in 802.1Q vlan#10 P0
2.226707 port2 in 802.1Q vlan#10 P0
2.226719 port1 out 802.1Q vlan#10 P0
2.246979 port2 in 802.1Q vlan#10 P0
2.246985 port1 out 802.1Q vlan#10 P0
2.248370 port2 in 802.1Q vlan#10 P0
2.248377 port1 out 802.1Q vlan#10 P0
2.356281 port2 in 802.1Q vlan#10 P0
2.356287 port1 out 802.1Q vlan#10 P0
2.357778 port2 in 802.1Q vlan#10 P0
2.357784 port1 out 802.1Q vlan#10 P0
2.468629 port2 in 802.1Q vlan#10 P0
2.468643 port1 out 802.1Q vlan#10 P0
2.489493 port2 in 802.1Q vlan#20 P0
2.489500 port1 out 802.1Q vlan#20 P0
2.490062 port1 in 802.1Q vlan#20 P0
2.490067 port2 out 802.1Q vlan#20 P0
3.290091 port2 in 802.1Q vlan#20 P0
3.502555 port2 in 802.1Q vlan#20 P0
3.502570 port1 out 802.1Q vlan#20 P0
3.503316 port1 in 802.1Q vlan#20 P0
3.503322 port2 out 802.1Q vlan#20 P0
```

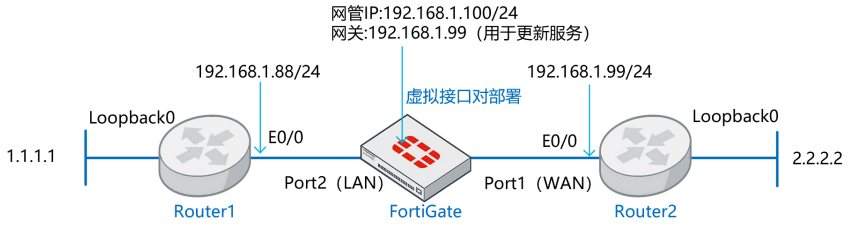
7. 推荐开启 wildcard-vlan （set wildcard-vlan enable），让 FGT 可以识别 VLAN - TAG 的数据，然后使用策略控制所有的携带和不携带 VLAN - TAG 的数据流。

组播 (OSPF) 与虚拟接口对

网络需求

路由器 1 和路由器 2 之间运行了动态路由协议，防火墙以虚拟接口对的方式运行在两台路由器之间，需要确保两台路由器之间的动态路由协议运行正常，并且数据可通。

网络拓扑



R1和R2之间运行了RIP/OSPF/EIGRP协议

配置要点

- Router1/Router2 路由器的基础配置
- 将防火墙的 port1 和 port2 配置为虚拟接口对
- 默认不配置策略情况下，观察动态路由协议的情况
- 配置安全策略解决 OSPF/EIGRP 邻居关系异常

配置步骤与结果验证

1. Router1/Router2 路由器的基础配置。

   Router1：

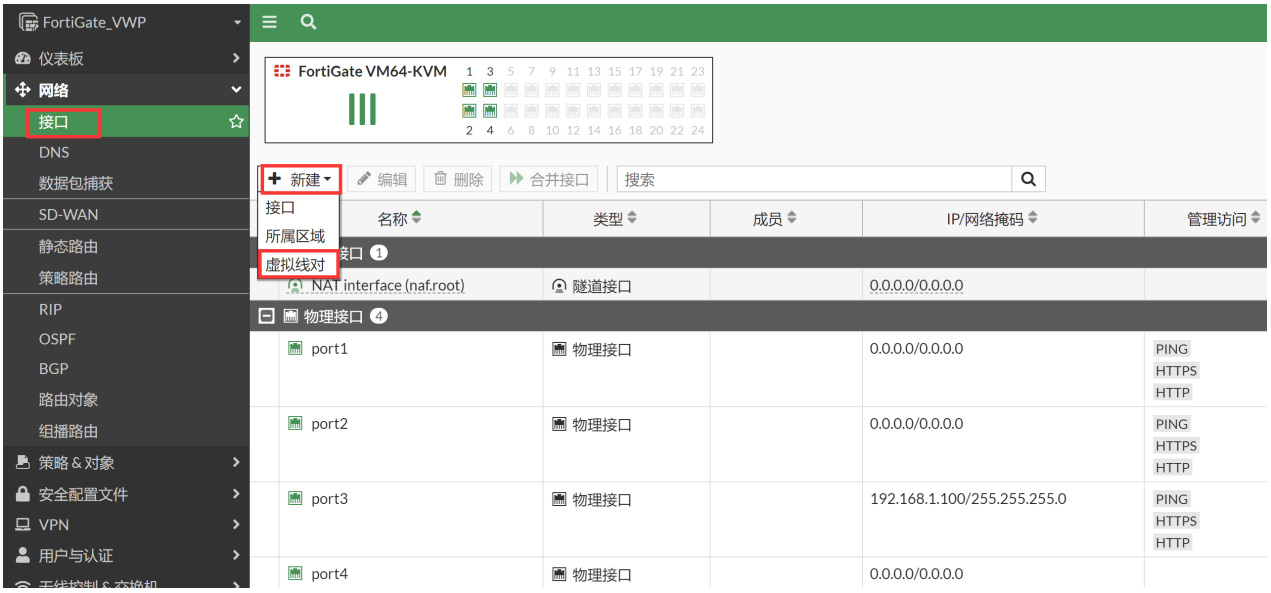
```
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
ip address 192.168.1.88 255.255.255.0
no shutdown
!
router ospf 10
network 1.1.1.1 0.0.0.0 area 0
network 192.168.1.0 0.0.0.255 area 0
!
router rip
version 2
network 1.0.0.0
network 192.168.1.0
no auto-summary
!
router eigrp 10
network 1.1.1.1 0.0.0.0
network 192.168.1.0
```

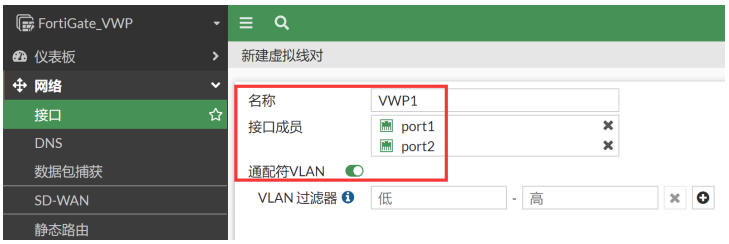
   Router2：

```
interface Loopback0
ip address 2.2.2.2 255.255.255.255
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
!
```

```
router ospf 10
network 2.2.2.2 0.0.0.0 area 0
network 192.168.1.0 0.0.0.255 area 0
!
router rip
version 2
network 2.0.0.0
network 192.168.1.0
no auto-summary
!
router eigrp 10
network 2.2.2.2 0.0.0.0
network 192.168.1.0
```

2. 将防火墙的 port1 和 port2 配置为虚拟接口对，防火墙处于路由模式的情况下，直接创建虚拟接口对，将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据（推荐开启通配符 VLAN）。







3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需要防火墙策略放通），对应的 CLI 如下。

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable    //这个命令很关键，识别VLAN-TAG，VWP模式下无需再到本地创建对应的VLAN接口和Forwar-Domain了，大大的简化了策略的配置量
    next
end
```

4. 默认不配置任何策略情况下，观察动态路由协议的情况，RIP 邻居是不正常的（无邻居，无路由），OSPF 邻居无法建立成功，EIGRP 邻居无法建立成功。

```
Router1#show ip rip database
1.0.0.0/8    auto-summary
1.1.1.1/32    directly connected, Loopback0
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, Ethernet0/0

Router1#show ip route
...
Gateway of last resort is not set
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.88/32 is directly connected, Ethernet0/0

Router1#show ip ospf  neighbor

Router1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(10)
```

原因分析

1. 我们知道 RIP/OSPF/EIGRP 都是用组播 IP 来建立邻居关系，同时 OSPF 和 RIGRP 也有单播报文的交互：

```
RIP:  UDP 520 || 组播 224.0.0.9
OSPF: IP 89   || 组播 224.0.0.5/224.0.0.6 || 单播 DBD报文使用单独发送
EIGRP: IP 88 || 组播 224.0.0.10   || 单播 UPDATE报文使用单播传递我们会发现在VWP环境中，RIP/OSPF/EIGRP路由协议统统失败。这和透明模式下的现象是不一样的。
```

2. 我们会发现在 VWP 环境中，RIP/OSPF/EIGRP 路由协议统统失败。这和透明模式下的现象是不一样的。
   - 透明模式下：RIP 是正常的，OSPF 的邻居虽然无法建立但组播交互是正常的，EIRGP 的邻居虽然失败但是组播交互也是正常的。
   - VWP 模式下：RIP/OSPF/EIGRP 都不行，组播单播都无法通信。
3. 我们再 VWP 模式下抓包查看，邻居交互的组播报文都只有 IN，没有 OUT，组播报文默认都被丢弃了。这是透明模式的 VWP 模式最大的一个区别，透明模式下组播默认存在一条 any 到 any 允许可以通的组播策略，而 VWP 实际上是运行在路由模式下的，而路由模式下的组播默认是被隐藏丢弃的。这是二者现象不一样的最关键原因，因此 RIP/OSPF/EIGRP 三者全挂。

```
FortiGate_VWP # diagnose sniffer packet any "host 192.168.1.88" 4
interfaces=[any]
filters=[host 192.168.1.88]
1501.278671 port2 in 192.168.1.88.520 → 224.0.0.9.520: udp 24
1501.278765 port3 in 192.168.1.88.520 → 224.0.0.9.520: udp 24

1502.138508 port2 in 192.168.1.88 → 224.0.0.5:  ip-proto-89 56
1502.138837 port3 in 192.168.1.88 → 224.0.0.5:  ip-proto-89 56

1511.257832 port2 in 192.168.1.88 → 224.0.0.10:  ip-proto-88 40
1511.258000 port3 in 192.168.1.88 → 224.0.0.10:  ip-proto-88 40

1512.054185 port2 in 192.168.1.88 → 224.0.0.5:  ip-proto-89 56
1512.054332 port3 in 192.168.1.88 → 224.0.0.5:  ip-proto-89 56

1520.487732 port2 in 192.168.1.88 → 224.0.0.10:  ip-proto-88 40
1520.487883 port3 in 192.168.1.88 → 224.0.0.10:  ip-proto-88 40

1527.402323 port2 in 192.168.1.88.520 → 224.0.0.9.520: udp 24
1527.402499 port3 in 192.168.1.88.520 → 224.0.0.9.520: udp 24
```

解决办法

1. 既然知道问题的原因，那么解决办法就很简单了，只需要放通组播的策略，然后再放通单播的策略即可。

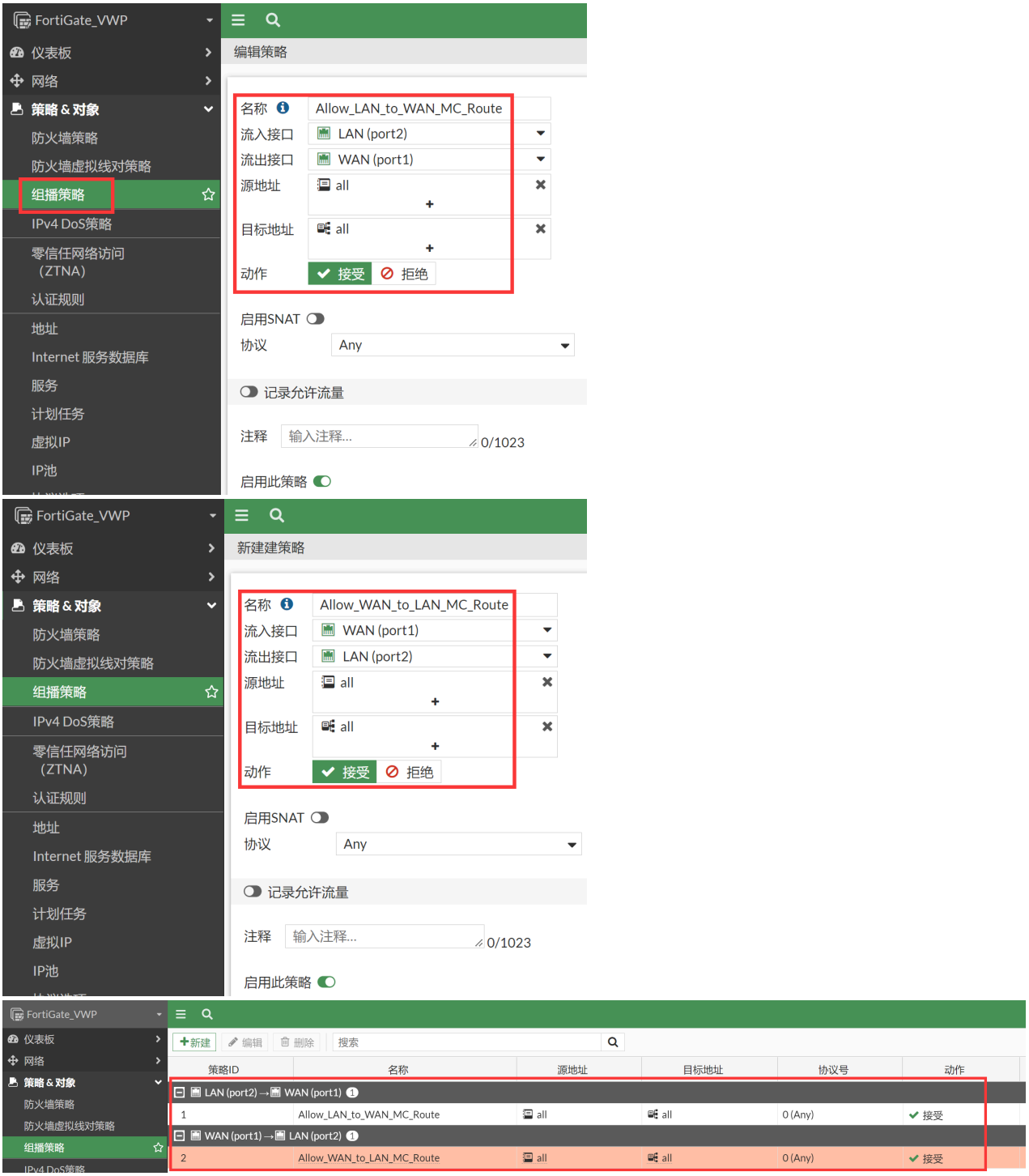
2. 首先在系统管理 → 可见功能，打开组播策略 GUI 的显示。



3. 进入策略 & 对象 → 组播策略中，可以看到默认的丢弃策略是隐藏的。



4. 新增组播策略允许路由组播通过。



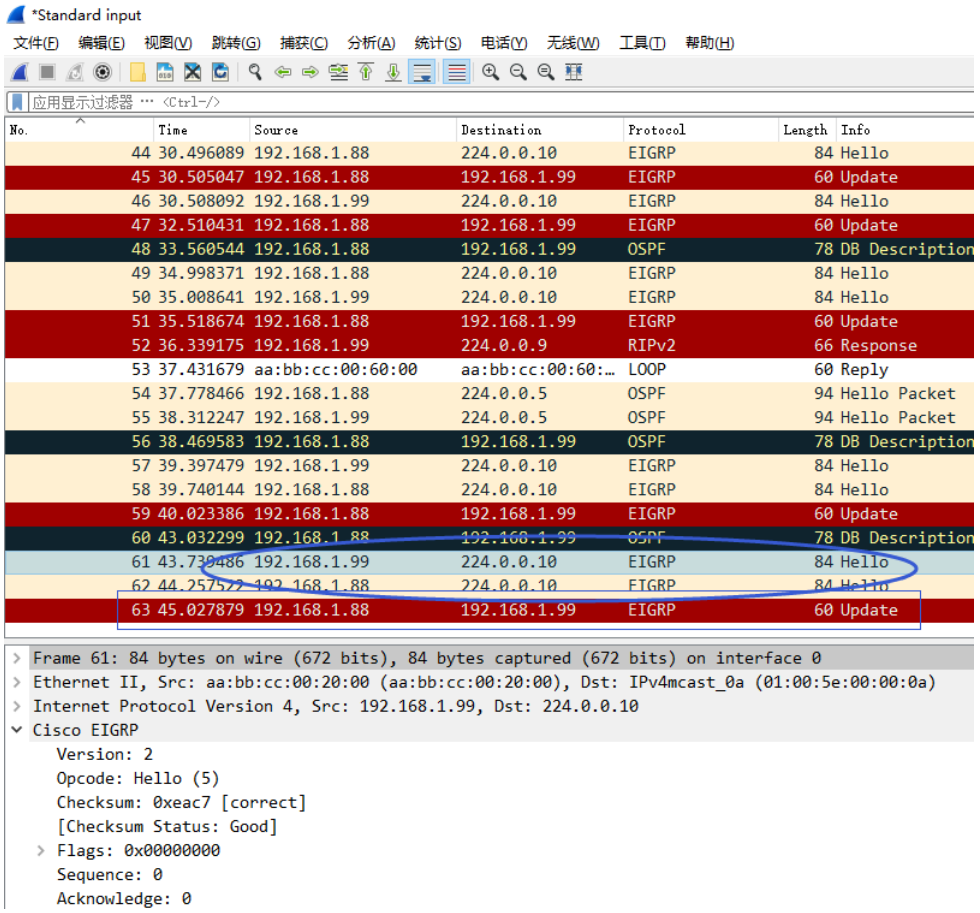
5. 组播策略配置完毕，RIP 协议就工作正常了。

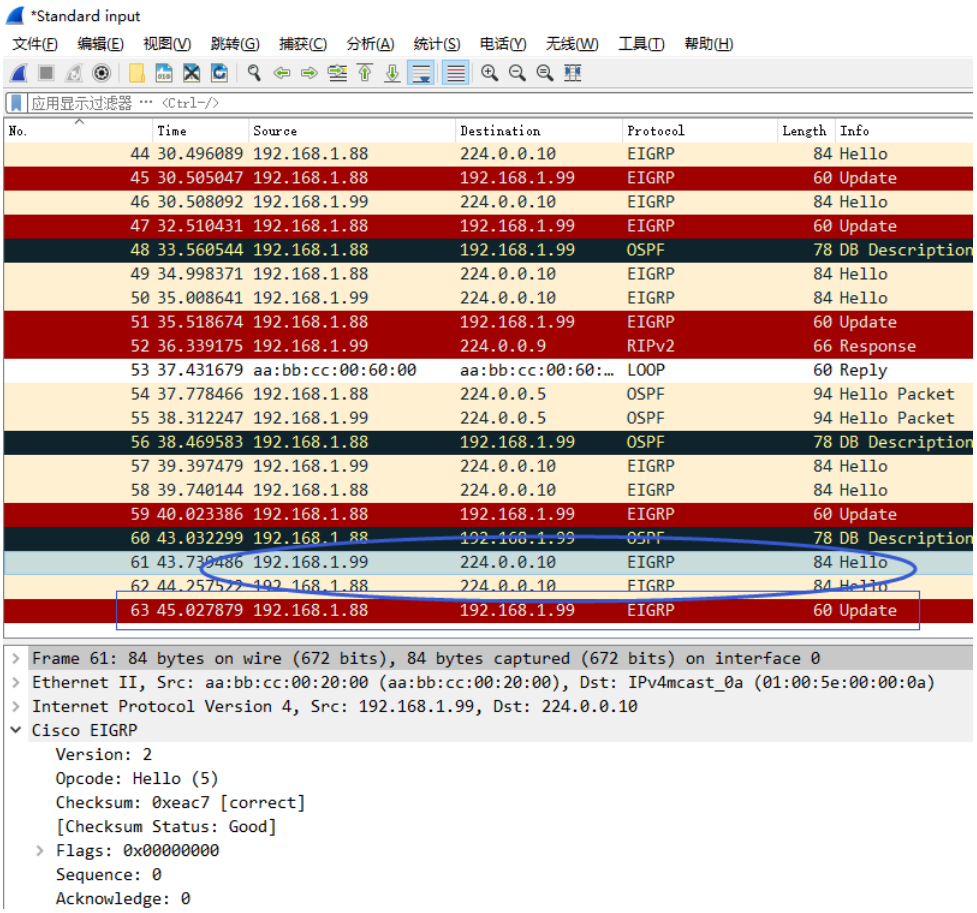
```
Router1#show ip rip database
1.0.0.0/8    auto-summary
1.1.1.1/32    directly connected, Loopback0
2.0.0.0/8    auto-summary
2.2.2.2/32
    [1] via 192.168.1.99, 00:00:01, Ethernet0/0
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, Ethernet0/0

Router1#show ip route
 ...
R        2.2.2.2 [120/1] via 192.168.1.99, 00:00:03, Ethernet0/0
 ...
```

6. RIP 协议只涉及到组播报文发送，但是 OSFP 和 EIGRP 邻居还是失败的，因为 OSPF 和 EIGRP 的邻居建立过程中需要单播的通信，我们来看现象。

```
Router1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 10)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         274         0×80000003 0×00F9A5 2

Router1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   INIT/DROTHER    00:00:37    192.168.1.99    Ethernet0/0

Router1#
*Jul  2 07:55:15.999: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on Ethernet0/0 from EXSTART to DOWN, Neighbor Down: Too
many retransmissions
*Jul  2 07:56:16.002: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on Ethernet0/0 from DOWN to DOWN, Neighbor Down: Ignore
timer expired
```

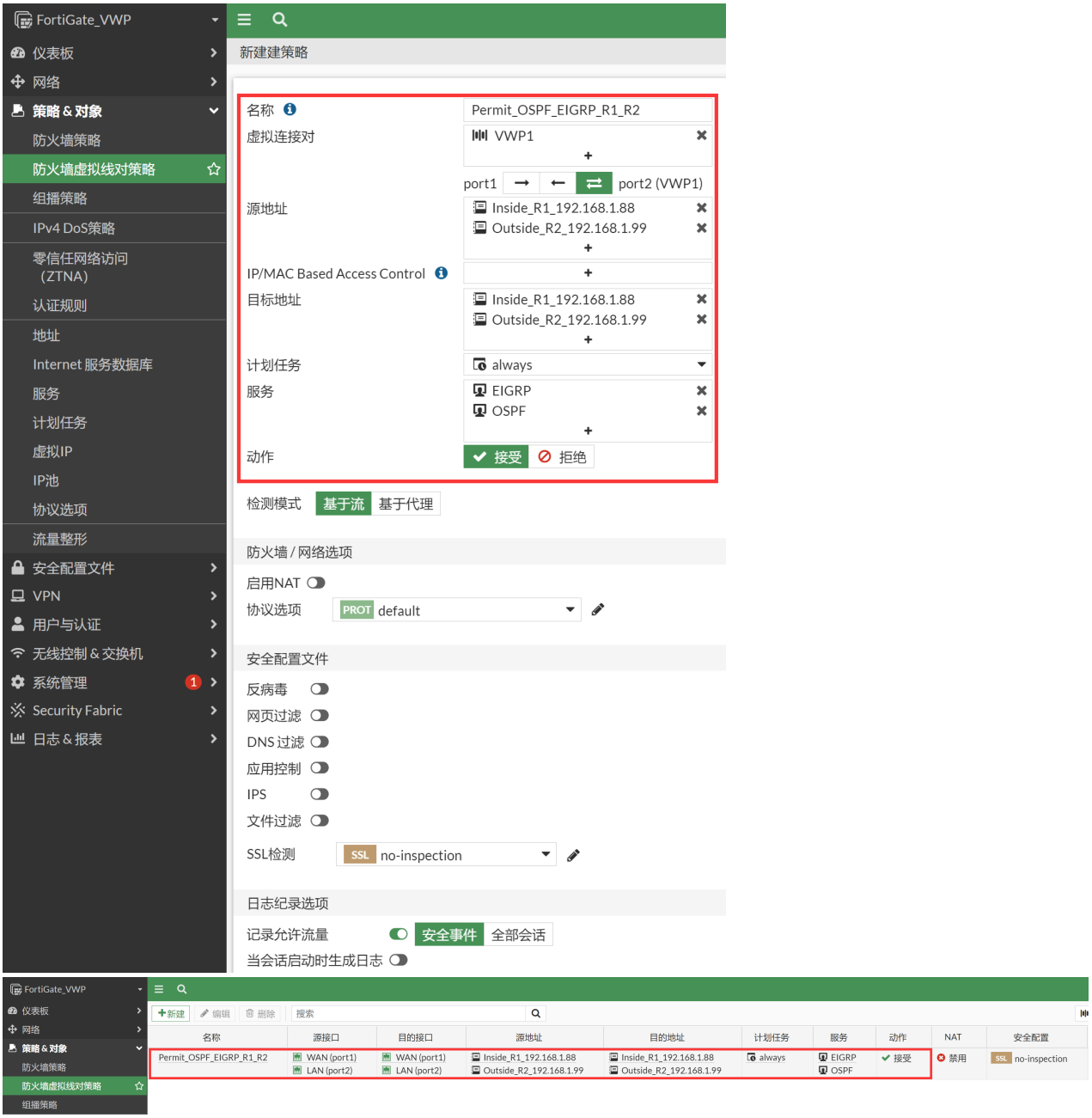
7. 这就和透明模式的现象完全一致了，进一步分析原因，OSPF 和 EIGRP 的组播其实是正常转发的，但是 OSPF 和 EIGRP 建立邻居关系不仅仅只使用了组播，而且还使用单播，默认情况下没有任何策略情况下，单播的数据会被默认丢弃，因此邻居关系异常。OSPF 的 DB Description 报文使用单播，EIGRP 的 Update 报文使用单播。这是 OSPF 和 EIGRP 邻居关系还是异常的原因。如下抓包所示，OSPF 和 EIGRP 邻居建立过程中的单播报文被 FGT 丢弃了。






```
FortiGate_VWP # diagnose sniffer packet any "host 192.168.1.88" 4
interfaces=[any]
filters=[host 192.168.1.88]
1.720001 port1 in 192.168.1.99 → 192.168.1.88:  ip-proto-88 20   //EIGRP单播数据只有IN没有OUT，被策略丢弃
1.747272 port2 in 192.168.1.88 → 192.168.1.99:  ip-proto-88 20

3.144458 port2 in 192.168.1.88 → 224.0.0.10:  ip-proto-88 40   //EIGRP组播数据有IN有OUT
3.144472 port1 out 192.168.1.88 → 224.0.0.10:  ip-proto-88 40

21.561302 port2 in 192.168.1.88 → 224.0.0.5:  ip-proto-89 56   //OSPF组播数据有IN有OUT
21.561320 port1 out 192.168.1.88 → 224.0.0.5:  ip-proto-89 56

22.021894 port2 in 192.168.1.88 → 192.168.1.99:  ip-proto-89 44  //OSPF单播数据只有IN没有OUT，被策略丢弃
22.023435 port2 in 192.168.1.88 → 192.168.1.99:  ip-proto-89 60
```

8. 既然 OSPF 和 EIGRP 邻居建立不起来的原因清楚了，那么要解决这个问题就很简单了。只需要增加普通的 IPv4 单播策略即可。



9. 上述策略配置完毕，会发现 RIP/OSPF/EIGRP 的三者的邻居关系都已经正常建立，路由学习也正常。

```
*Oct 15 09:20:13.561: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 192.168.1.99 (Ethernet0/0) is up: new adjacency
*Oct 15 09:20:39.923: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on Ethernet0/0 from LOADING to FULL, Loading Done
Router1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/DR         00:00:34    192.168.1.99    Ethernet0/0

Router1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(10)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   192.168.1.99            Et0/0                    13 00:03:06   23   138  0  35

Router1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(10)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 192.168.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 2.2.2.2/32, 1 successors, FD is 409600
        via 192.168.1.99 (409600/128256), Ethernet0/0
P 1.1.1.1/32, 1 successors, FD is 128256
```

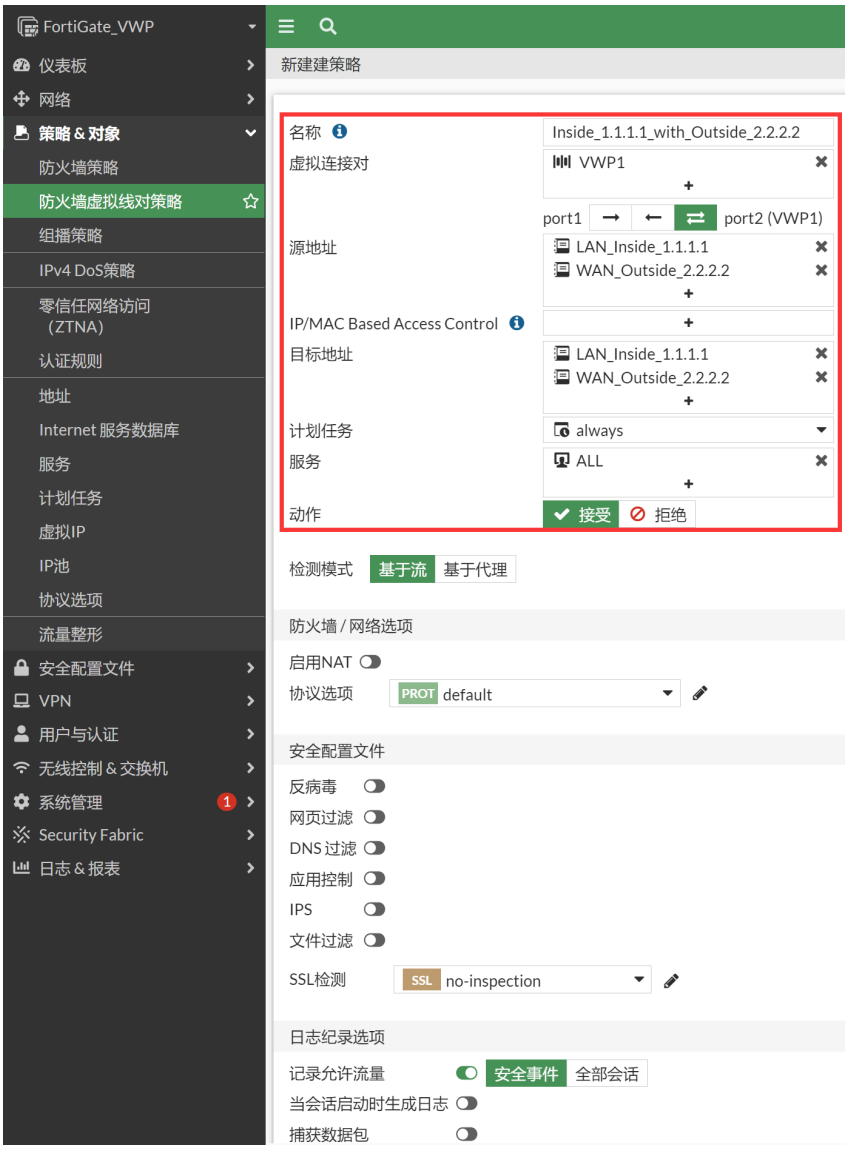
```
        via Connected, Loopback0

Router1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 10)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         252         0×80000005 0×0098DF 2
2.2.2.2         2.2.2.2         253         0×80000005 0×008DD3 2
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.99    2.2.2.2         253         0×80000001 0×003B1D

Router1#show ip rip database
1.0.0.0/8    auto-summary
1.1.1.1/32    directly connected, Loopback0
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, Ethernet0/0

Router1#show ip route
 …
D        2.2.2.2 [90/409600] via 192.168.1.99, 00:02:51, Ethernet0/0
 …
```

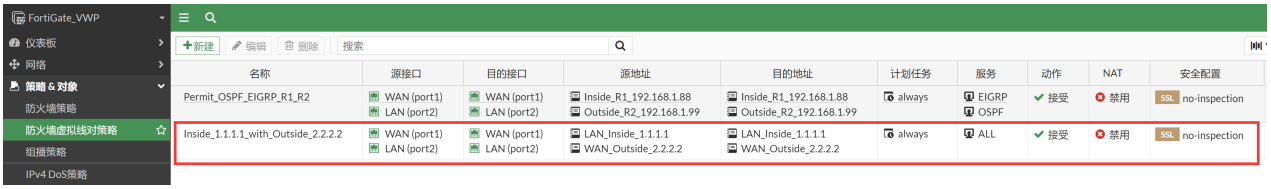
0. 需要配置业务策略允许 Router1 的 1.1.1.1 访问 Router2 的 2.2.2.2，没配置策略之前是不通的。

```
Router1#ping 2.2.2.2 source  1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)
```

1. 配置 VWP 策略允许 Router1 的 1.1.1.1 访问 Router2 的 2.2.2.2。







2. 测试 Router1 的 1.1.1.1 访问 Router2 的 2.2.2.2，可以正常访问。

```
Router1#ping 2.2.2.2 source  1.1.1.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/9 ms
```

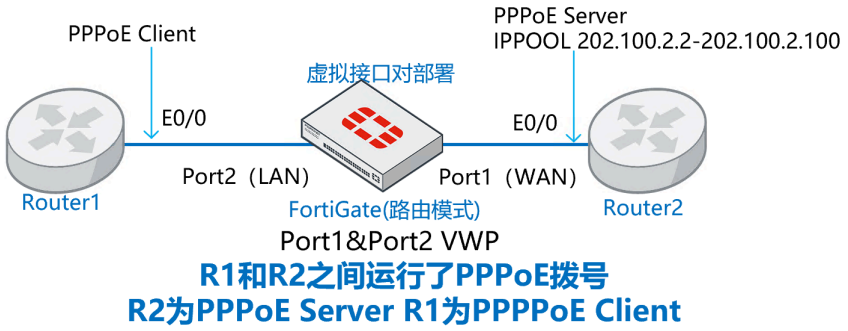
```
FortiGate_VWP # diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
5.551382 port2 in 1.1.1.1 → 2.2.2.2: icmp: echo request
5.551455 port1 out 1.1.1.1 → 2.2.2.2: icmp: echo request
5.552281 port1 in 2.2.2.2 → 1.1.1.1: icmp: echo reply
5.552292 port2 out 2.2.2.2 → 1.1.1.1: icmp: echo reply
```

非 IP 流量 (PPPOE) 与虚拟接口对

网络需求

路由器 1 和路由器 2 之间运行了 PPPOE 协议（非 IP 协议），防火墙以虚拟接口对运行在两台路由器之间，需要确保路由器 1 的 PPPOE 拨号正常。

网络拓扑



配置要点

- Router1/Router2 路由器的基础配置
- 将防火墙的 port1 和 port2 配置为虚拟接口对
- 默认情况下，观察 PPPOE 拨号的情况
- 配置 l2forward 解决非 IP 流量转发问题

配置步骤与结果验证

1. Router1/Router2 路由器的基础配置。

   Router1：

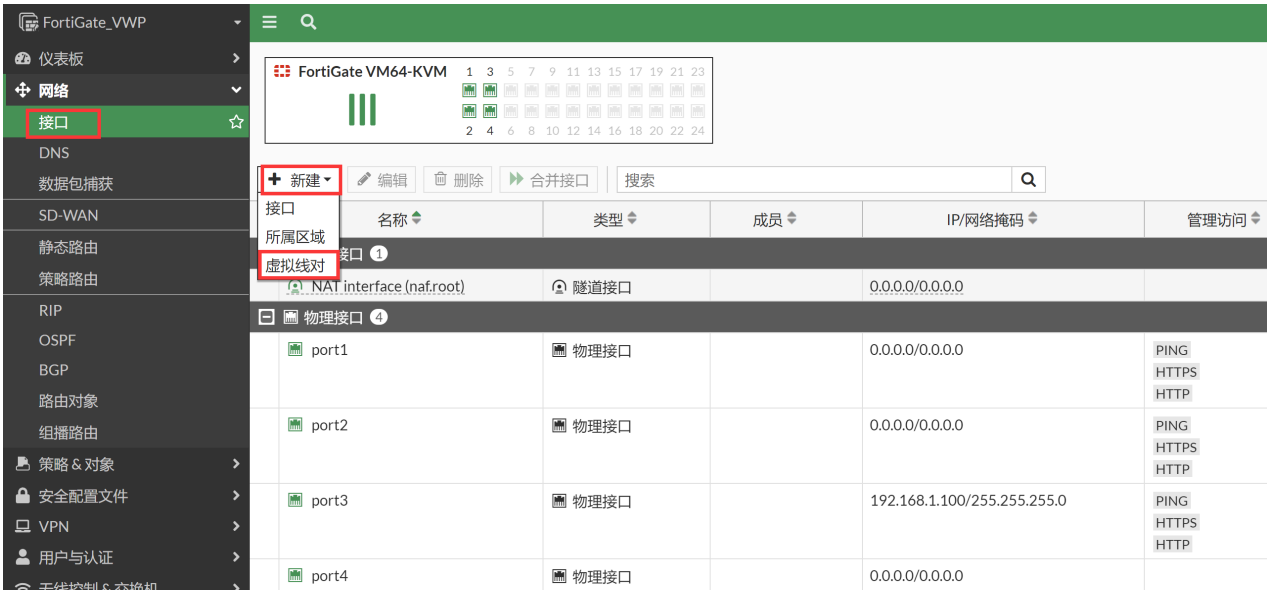
```
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
no ip address
no shutdown
pppoe enable group global
pppoe-client dial-pool-number 1
!
interface Dialer1
mtu 1492
ip address negotiated
encapsulation ppp
dialer pool 1
ppp pap sent-username pppoeuser1 password 0 fortinet
ppp ipcp route default
```

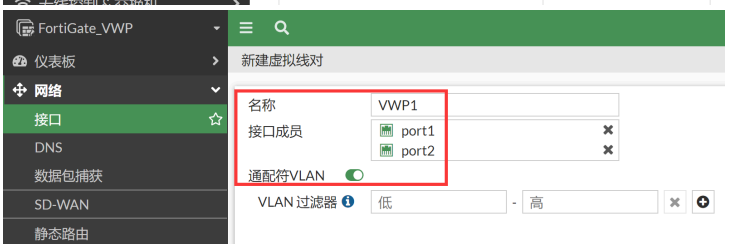
   Router2：

```
interface Loopback0
ip address 2.2.2.2 255.255.255.255
!
username pppoeuser1 password 0 fortinet
!
bba-group pppoe bba1
virtual-template 1
!
interface Loopback1
ip address 202.100.2.1 255.255.255.0
```

```
!
interface Ethernet0/0
no ip address
no shutdown
pppoe enable group bba1
!
interface Virtual-Template1
description PPPOE bba1
mtu 1492
ip unnumbered Loopback1
peer default ip address pool pool1
ppp authentication pap
!
ip local pool pool1 202.100.2.2 202.100.2.100
```

2. 将防火墙的 port1 和 port2 配置为虚拟接口对，防火墙处于路由模式的情况下，直接创建虚拟接口对：将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据（推荐开启通配符 VLAN）。







3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需策略放通），对应的命令行如下。

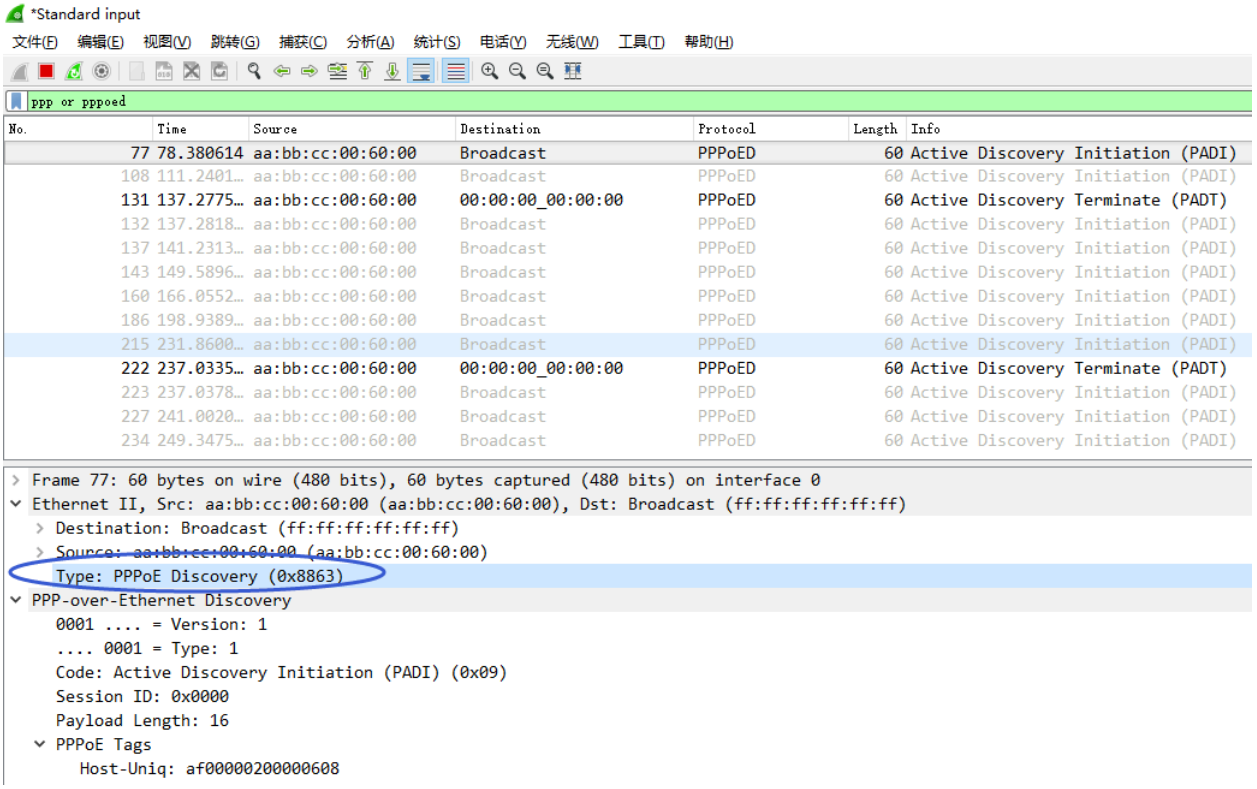
```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable
    next
end
```

4. 默认情况下，观察 PPPOE 拨号的情况，Router1（PPPOE Client）无法 PPPPOE 拨号成功。

```
Router1#show ip route
...
Gateway of last resort is not set
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0

Router1#show ip int brief
...
Dialer1                    unassigned      YES IPCP   up                    up
...
```

5. 通过在 FGT 上抓包可以看到 FGT 并不转发 PPPOE 的报文，非 IP 的 PPPOE 数据默认不会被 FGT 所转发。

```
FortiGate_VWP # diagnose sniffer packet any "pppoed" 4
interfaces=[any]
filters=[pppoed]
4.124126 port1 in pppoe printer hasn't been added to sniffer
37.014209 port1 in pppoe printer hasn't been added to sniffer
69.908430 port1 in pppoe printer hasn't been added to sniffer
102.798634 port1 in pppoe printer hasn't been added to sniffer
135.705063 port1 in pppoe printer hasn't been added to sniffer
```



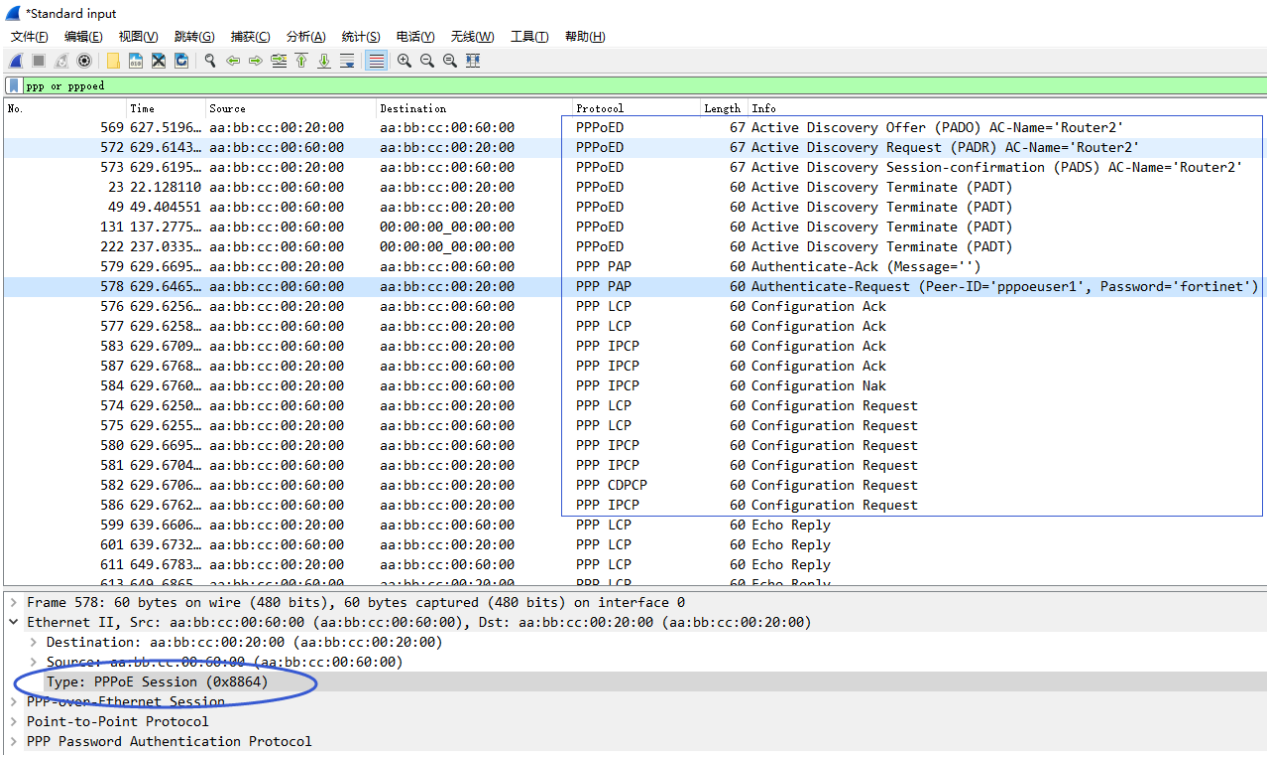
6. 那么要如何解决此情况呢？在 FGT 的虚拟接口对的具体接口下配置 set l2forward enable 即可！

```
FortiGate_VWP # config system interface
FortiGate_VWP (interface) # edit port1
FortiGate_VWP (port1) # set l2forward enable

FortiGate_VWP (port1) # next
```

```
FortiGate_VWP (interface) # edit port2
FortiGate_VWP (port2) # set l2forward enable
FortiGate_VWP (port2) # end
```

7. 再次抓包查看，此时 FGT 可以转发 PPPOE 请求了。

```
FortiGate_VWP # diagnose sniffer packet any "pppoed" 4
interfaces=[any]
filters=[pppoed]
5.324865 port2 in pppoe printer hasn't been added to sniffer
5.324880 port1 out pppoe printer hasn't been added to sniffer
5.325468 port1 in pppoe printer hasn't been added to sniffer
5.325474 port2 out pppoe printer hasn't been added to sniffer
7.420329 port2 in pppoe printer hasn't been added to sniffer
7.420342 port1 out pppoe printer hasn't been added to sniffer
7.425307 port1 in pppoe printer hasn't been added to sniffer
7.425313 port2 out pppoe printer hasn't been added to sniffer
```



8. 路由器 1 PPPOE 拨号成功。

```
Router1#show ppp all
Interface/ID OPEN+ Nego* Fail-     Stage    Peer Address    Peer Name
------------ ------------------- -------- --------------- --------------------

Vi2          LCP+ IPCP+ CDPCP-     LocalT   202.100.2.1

Router1#show pppoe session
     1 client session
Uniq ID  PPPoE  RemMAC          Port                   VT  VA         State
           SID  LocMAC                                     VA-st      Type
    N/A      2  aabb.cc00.2000  Et0/0                  Di1 Vi2        UP
                aabb.cc00.6000                             UP

Router1#show ip route
 ...
Gateway of last resort is 202.100.2.1 to network 0.0.0.0
S*    0.0.0.0/0 [1/0] via 202.100.2.1
 ...

Router1#
```

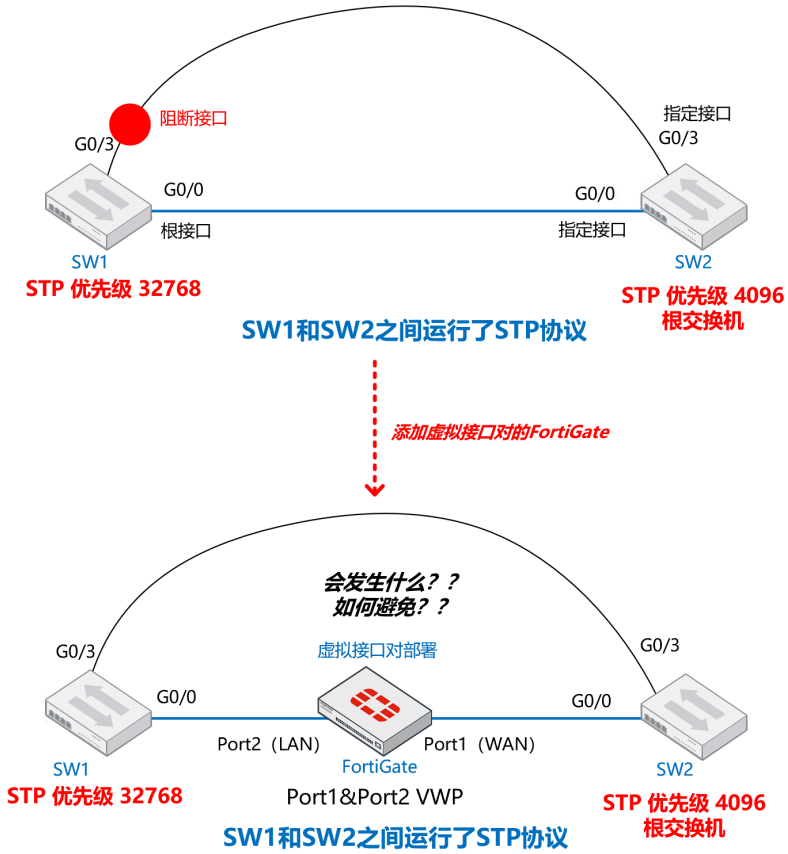
```
Router1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
 ...
Dialer1                    202.100.2.3     YES IPCP   up                    up
 ...
```

生成树 (STP) 与虚拟接口对

网络需求

SW1 和 SW2 之间运行了 STP 协议，防火墙以虚拟接口对方式运行在两台 SW 之间，SW 和 SW 之间原本存在环路，通过 STP 进行隔离阻断了环路接口，在加入虚拟接口对的 FGT 之后，需要继续确保 STP 工作正常。

网络拓扑



配置要点

- Router1/Router2 路由器的基础配置
- 将防火墙配置为虚拟接口对并开启网管
- 默认情况下，观察 STP BPDU 报文与收敛情况
- 配置 stpforward 解决 STP 流量转发问题

配置步骤与结果验证

1. SW1/SW2 交换机的基础配置和 STP 状态确认。

SW1:

```
hostname SW1
spanning-tree vlan 1 priority 32768
```

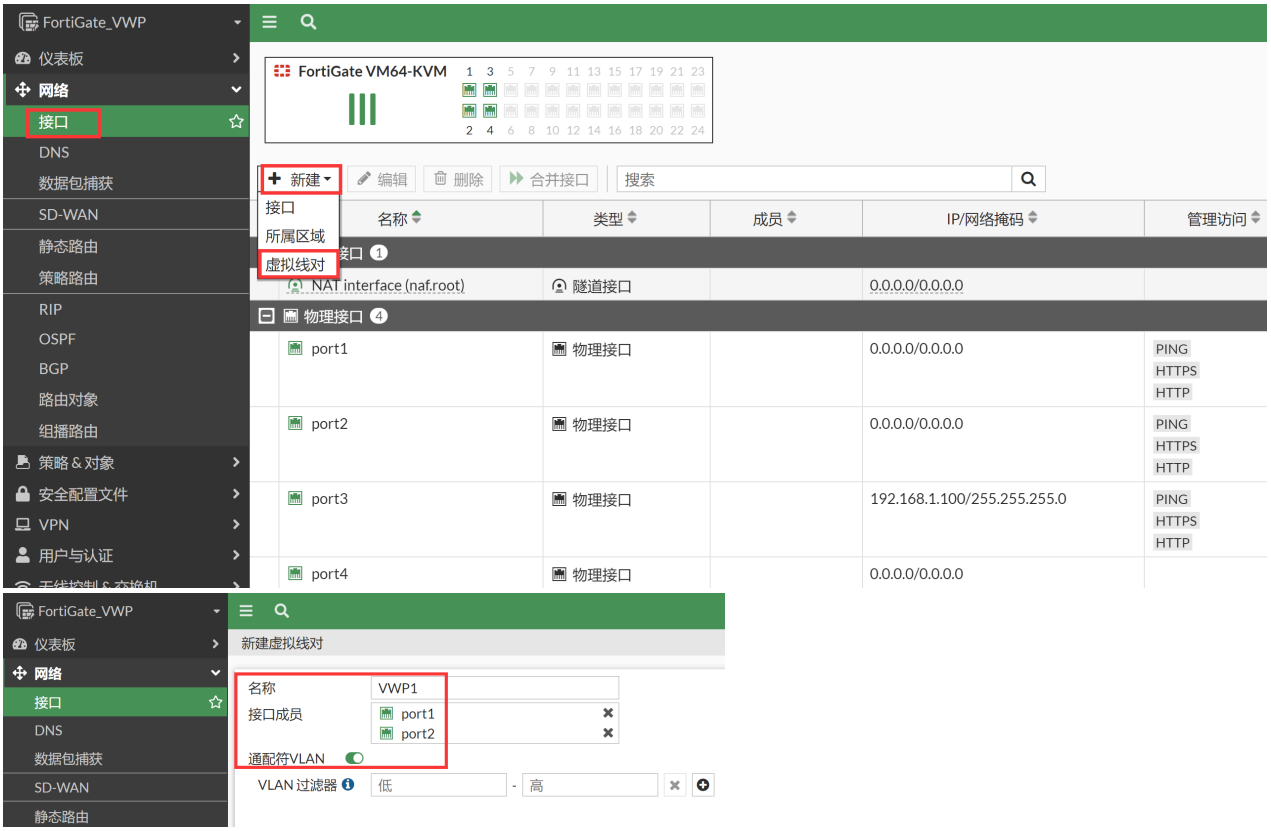
SW2:

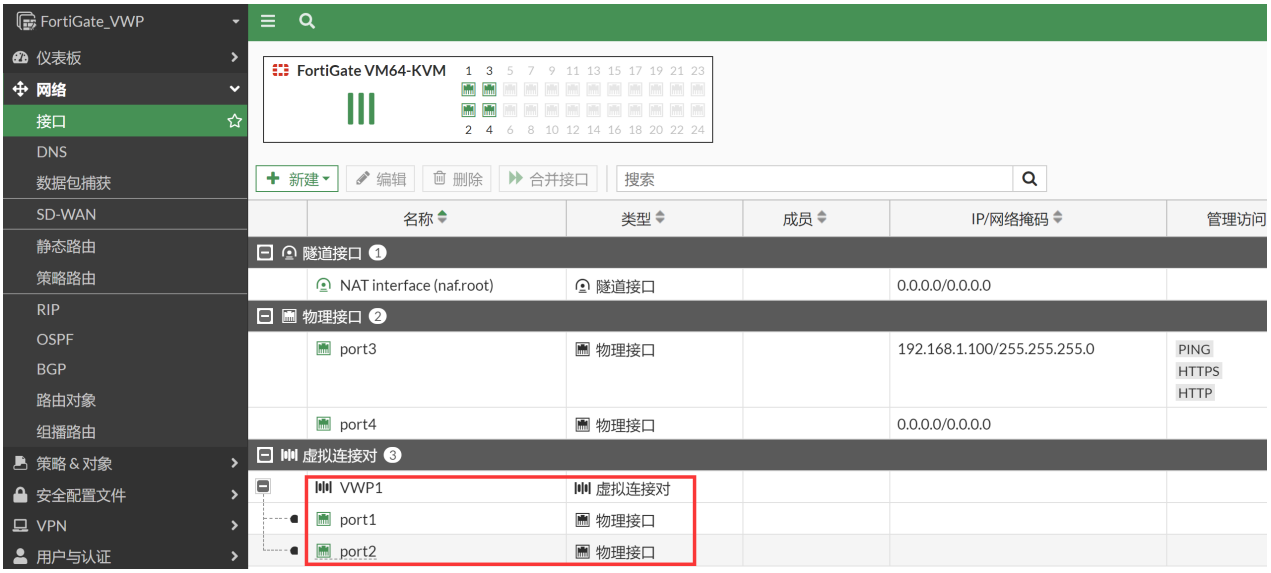
```
hostname SW2
!
spanning-tree vlan 1 priority 4096
```

没有加入虚拟接口对的 FortiGate 之前 STP 状态：

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    Shr
Gi0/3               Altn BLK 4         128.4    Shr      // G0/3 被阻断，STP工作正常

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr
```

2. 将防火墙的 port1 和 port2 配置为虚拟接口对并加入到 SW1 和 SW2 之间，防火墙处于路由模式的情况下，直接创建虚拟接口对：将 port1 和 port2 加入到虚拟接口对中并开启通配符 VLAN，识别 VLAN-Tag 的数据（推荐开启通配符 VLAN）。





3. 虚拟接口对通道正式打通，可以认为 port1 和 port2 之间可以透明的传输数据了（需策略放通），对应的命令行如下。

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enable
    next
end
```

4. 加入 FGT 在没有任何配置的情况下，观察 STP 的运行情况，一旦 FGT 加入到 SW1 和 SW2 之间，很明显网络中出现了环路，设备操作很卡顿，SW1 和 SW2 基本上不再响应请求，三台设备均出现了异常情况（默认情况下虚拟接口对的 FGT 只转发 ARP 报文，因此只需要一个 arp 报文就会产生广播风暴，引起环路导致设备、业务中断）。

```
FortiGate_VWP # diagnose sniffer packet any "none" 4
interfaces=[any]
filters=[none]
0.814720 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.814733 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.817141 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.817148 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.818761 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.818771 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.819709 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.819720 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.821533 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.821544 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.825484 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.825492 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.827856 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.827866 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.828752 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.828762 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.829235 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.829244 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.831930 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.831947 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.834842 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.834853 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.836686 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.836694 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.838965 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.838972 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.841965 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.841974 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.843858 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.843866 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
```

5. SW1/SW2 的 STP 状态均异常，所有接口都处于转发状态。

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        4 (GigabitEthernet0/3)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Root FWD 4         128.4    Shr     //SW1的两个接口都处于转发状态

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr  //SW2的两个接口都处于转发状态
```

6. 在 FGT 上抓包可以看到 STP 的 BPDU 报文没有被转发，而是被默认丢弃，这是引起环路的根本原因。STP BPDU 数据默认不会被虚拟接口对的 FGT 所转发，因此 STP 计算不正确，会计算不出网络中的环路，从而引起广播风暴。

```
FortiGate_VWP # diagnose sniffer packet any "none" 4
interfaces=[any]
filters=[none]
1.234590 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
2.182694 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
3.247650 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
4.193908 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
5.260154 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
6.204003 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
7.278226 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
8.210732 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
9.294582 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
10.223021 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
11.302829 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
12.232585 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
13.310023 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
14.242616 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
```

7. 那么要如何解决此情况呢？在 FGT 的二层接口下配置 set stpforward enable 即可！

```
FortiGate_VWP # config system interface
FortiGate_VWP (interface) # edit port1
FortiGate_VWP (port1) # set stpforward enable

FortiGate_VWP (port1) # next
FortiGate_VWP (interface) # edit port2
FortiGate_VWP (port2) # set stpforward enable
FortiGate_VWP (port2) # end
```

8. 再次抓包查看，此时 FGT 可以转发 STP BPDU 报文了。

```
FortiGate_VWP # diagnose sniffer packet any "stp" 4
interfaces=[any]
filters=[stp]
2.070782 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
2.070797 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

4.085338 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
4.085355 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

6.097429 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
6.097444 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

8.107806 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
8.107822 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

10.114977 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
10.114993 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

12.123637 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
12.123651 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001

14.131744 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
14.131757 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
```

9. 此时再次查看 STP 的状态，恢复正常选举状态。

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    Shr
Gi0/3               Altn BLK 4         128.4    Shr    // G0/3 被阻断，STP工作正常

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr
```

重要

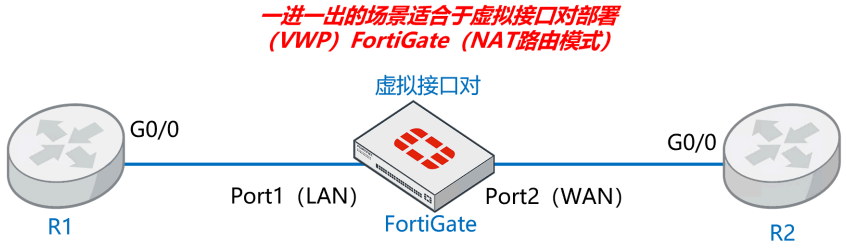
要特别注意此情况！避免虚拟接口对的 FGT 一上线引起 STP 计算异常，引起广播风暴，造成重大业务故障。

虚拟接口对最佳配置实践

网络需求

一进一出的这样的透明需求的组网推荐使用虚拟接口对（virtual-wire-pair）。

网络拓扑



推荐配置

1. 无论网络中是否存在 VLAN 都推荐开启 wildcard-vlan （set wildcard-vlan enable），让 FGT 可以识别 VLAN - TAG 的数据，然后使用策略控制所有的携带和不携带 VLAN - TAG 的数据流。

2. 开启了 wildcard-vlan enable，vlanforward 不生效，wildcard-vlan 优先，识别 VLAN - TAG，并上送到 FGT 的 CPU 上去策略处理。这是推荐的配置（推荐）。

3. 如果关闭了 wildcard-vlan disable，则 vlanforward 生效，则携带了 vlan-tag 的数据会直接穿透 VWP 接口，不再匹配策略，直接 bypass 可通。（不推荐这样使用，流量不可见也不可控，这不是安全设备想要的结果）。

```
config system virtual-wire-pair
    edit "VWP1"
        set member "port1" "port2"
        set wildcard-vlan enbale  //手工开启wildcard
    next
end

config system interface
    edit "port1"
        set vdom "root"
        set vlanforward enable    //保持默认enable状态
        set type physical
        set alias "WAN"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set vlanforward enable  //保持默认enable状态
        set type physical
        set alias "LAN"
        set snmp-index 2
    next
end
```

4. 虚拟接口对的 FortiGate 其实默认只转发 IPv4 ARP 广播和 IPv6 ND 组播报文，其他的流量通通会被丢弃。如果你想让虚拟接口对更像一个 SW 一样的存在，那么在使用的接口下应该这样去修改，这样 FortiGate 对接入的网络环境影响应该来说是更小的，对于那种不是很清楚网络中有一些什么流量的环境，建议按照上述建议进行配置优化，让 FortiGate 更加接近于一个 SW 的存在。

```
config system interface
    edit "port1"
        set broadcast-forward enable    //开启广播转发
        set l2forward enable           //开启非IP流量转发
        set stpforward enable         //开启STP转发
    next
    edit "port2"
```

```
        set broadcast-forward enable
        set l2forward enable
        set stpforward enable
    next
end
```

5. 使用一个独立的的 VDOM 用来保护二层（虚拟接口对）的业务流量，然后再使用一个独立管理 VDOM（NAT 路由模式）用于设备的管理和 fortiguard 升级等，利用 VDOM 功能将管理（NAT 路由模式）和业务（虚拟接口对）分离。

故障排错

1. 先检查 FortiGate 的二层 MAC 地址表，下面的命令能导出每一个虚拟接口对的二层转发表，在转发表里可以单独看到 HOST 与接口的对应 MAC 地址关系。

```
FortiGate_VWP # diag netlink brctl list
list bridge information
1. VWP1_v.b         fdb: size=2048 used=2    num=2    depth=1
Total 1 bridges
diag netlink brctl name host <VWP_name_v>.b

虚拟接口对名字VWP1的查看二层转发表的例子：
FortiGate_VWP # diag netlink brctl name host VWP1_v.b
show bridge control interface VWP1_v.b host.
fdb: size=2048, used=2, num=2, depth=1
Bridge VWP1_v.b host table
port no device  devname mac addr                ttl     attributes
  2     4       port2   50:00:00:03:00:01       0       Local Static
  1     3       port1   50:00:00:03:00:00       0       Local Static
```

2. 使用 sniffer 验证数据包是否到达防火墙，通过以下命令使用 sniffer 抓取相应信息；通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题。这个能看到在 FortiGate 上面流入以及流出的数据包。

```
进行抓取目的地址是114.114.119.119的数据：
diagnose sniffer packet any "host 114.114.119.119" 4 0 l           按CTRL+C进行停止

基于端口进行数所包抓取
diagnose sniffer packet portA  "host 114.114.119.119" 4 0  l
diagnose sniffer packet portB  "host 114.114.119.119" 4 0  l

如有需要，可以通过每个vlan进行获取更明细内容
diagnose sniffer packet <each_vlan_interface> "" 6 0 l

参数4和6的区别（用于显示输出抓取数据参数）：
1: print header of packets (只有IP头部)
2: print header and data from ip of packets (有ip头部和数据内容)
3: print header and data from ethernet of packets  (if available)
4: print header of packets with interface name (只有IP头部和端口)
5: print header and data from ip of packets with interface name (有ip头部和数据内容以及端口)
6: print header and data from ethernet of packets (if available) with intf name

参数0的用处：输出个数

参数l的用处(用于输出时候显示的时间格式)：
a：absolute UTC time, yyyy-mm-dd hh:mm:ss.ms
l：absolute LOCAL time, yyyy-mm-dd hh:mm:ss.ms
```

3. 使用 debug flow 验证数据包是否穿越防火墙（vlanforward 无效），在每一个 VDOM 使用 "debug flow" 命令进行跟踪流量，通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题。这种排障的方法能够查看到流量是否因为某条策略导致数据包不通或者通过设备。

```
以源地址为抓取条件
═══════════════════════════════════════════════════════════
diag debug flow filter add <IP_address_of_source_device>
diag debug flow show function-name enable
diag debug flow show console enable
diag debug flow show iprope enable
diag debug flow trace start 100
diag debug enable
═══════════════════════════════════════════════════════════
如需要停止，可以敲"diag debug flow trace stop"
```

以目标地址为抓取条件

```
diag debug flow filter add <IP_address_of_destination_device>
diag debug flow show function-name enable
diag debug flow show console enable
diag debug flow show iprope enable
diag debug flow trace start 100
diag debug enable
```

如需要停止，可以敲"diag debug flow trace stop"

当有触发排障的流,输出如下:

```
id=20085 trace_id=113 msg="vd-tp_mode received a packet(proto=6,10.160.0.160:4370→10.160.0.152:23) from internal."
id=20085 trace_id=113 msg="Find an existing session, id-00000a40, original direction"
id=20085 trace_id=113 msg="enter fast path"
id=20085 trace_id=113 msg="send out via dev-dmz1, dst-mac-00:01:02:03:04:05"
```

4. 查看数据包在防火墙上创建的 session。在每一个 VDOM 使用 "diag sys session" 命令进行防火墙会话检查，通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题，这种排障的方法能够查看防火墙是否会为流量创建会话，以及是否硬件加速以及命中哪一条策略。

以源地址为抓取条件（单播）

```
diag sys session filter src <IP_address_of_source_device>
diag sys session list
```

当有触发排障的会话,输出如下:

```
session info: proto=1 proto_state=00 duration=2 expire=58 timeout=0 flags=00000000 sockflag=00000000 sockport=0 av_idx=0
use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=500/5/1 reply=500/5/1 tuples=2
tx speed(Bps/kbps): 248/1 rx speed(Bps/kbps): 248/1
orgin→sink: org pre→post, reply pre→post dev=9→10/10→9 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 192.168.1.1:64→192.168.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.2.1:64→192.168.1.1:0(0.0.0.0:0)
misc=0 policy_id=4 auth_info=0 chk_client_info=0 vd=1
serial=00000512 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000
vlanid=100 dd_type=0 dd_mode=0
total session 1
```

以源地址为抓取条件（多播）

```
diagnose sys mcast-session list
```

当有触发排障的会话,输出如下:

```
session info: id=123 vf=1 proto=112 192.168.1.2.0→224.0.0.18.0
used=2 path=1 duration=1353 expire=179 indev=10 pkts=1360 bytes=54400
state=00000002:tp
path: policy=0, outdev=9

session info: id=124 vf=1 proto=112 192.168.2.99.0→224.0.0.18.0
used=2 path=1 duration=1353 expire=179 indev=10 pkts=1360 bytes=54400
state=00000002:tp
path: policy=0, outdev=9
Total 2 sessions
diag sys session list
```

参考文档

https://community.fortinet.com/t5/FortiGate/Technical-Tip-Troubleshoot-and-verify-if-traffic-is-hitting-a/ta-p/197261 ↗

https://community.fortinet.com/t5/FortiGate/Troubleshooting-Tip-First-steps-to-troubleshoot-connectivity/ta-p/192560 ↗

https://community.fortinet.com/t5/FortiGate/Troubleshooting-Tip-FortiGate-session-table-information/ta-p/196988 ↗

https://community.fortinet.com/t5/FortiGate/Technical-Tip-Configuring-a-FortiGate-in-Transparent-mode-to/ta-p/197611 ↗

https://community.fortinet.com/t5/FortiGate/Technical-Note-FortiGate-and-Gratuitous-ARP-GARP/ta-p/192962 ↗

传统透明模式

开启透明模式防火墙并保护上网流量

VLAN 与透明模式

- Forward-Domain
- vlanforward
- Remapping VLAN ID (VLAN NAT，特殊场景使用)

组播 (OSPF) 与透明模式

非 IP 流量 (PPPOE) 与透明模式

生成树 (STP) 与透明模式

透明模式的带外管理

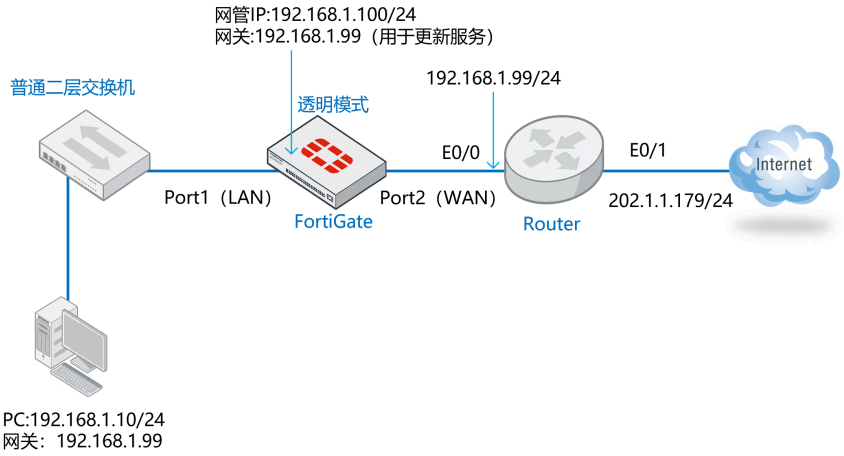
Bypass 部署

透明模式注意事项与排错

开启透明模式防火墙并保护上网流量

网络需求

在不改变现有网络拓扑前提下，将防火墙 NGFW 以透明模式部署到网络中，放在路由器和交换机之间，防火墙为透明模式，对内网网段 192.168.1.0/24 的上网进行 4~7 层的安全防护。



配置要点

- Internet_R 路由器的基础配置
- 将防火墙配置为透明模式并开启网管
- 配置安全策略允许内网 PC 访问互联网

配置步骤

1. Internet_R 路由器的基础配置（以思科为例）。

```
hostname Internet_R
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

2. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。配置可以管理防火墙的本地 IP 和网关。

```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 192.168.1.100 255.255.255.0
    set gateway 192.168.1.99
end
```

3. 查看当前设备的状态已变为透明模式。

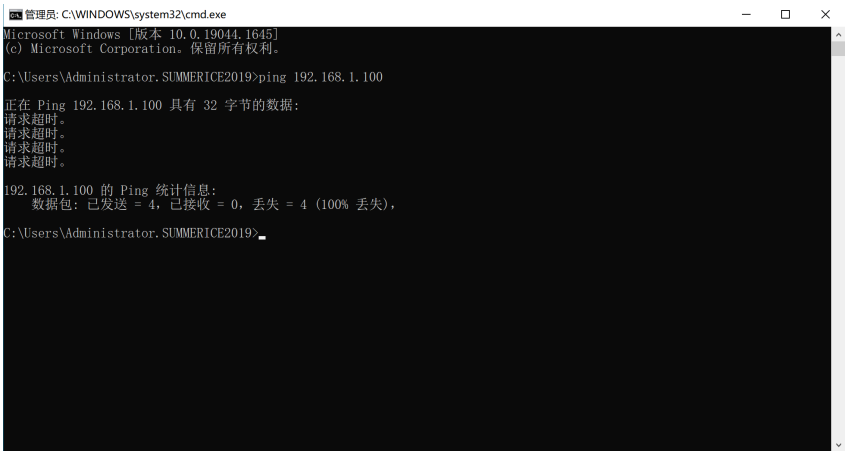
```
FortiGate_Transparent # get system status
Version: FortiGate-VM64-KVM v7.0.8,build0418,221012 (GA.F)
...
Operation Mode: Transparent
...
Virtual domains status: 0 in NAT mode, 1 in TP mode
...
```

重要

透明模式下，所有的接口都无法写入 IP 地址，整机只有一个设备 IP 地址。当需要从某个接口管理设备时，需在命令行下开启相应接口的管理权限。

开启透明模式前，需要删除系统自带的 fortilink 接口（需要先删除 fortilink 接口关联的 DHCP 与 NTP 配置）。

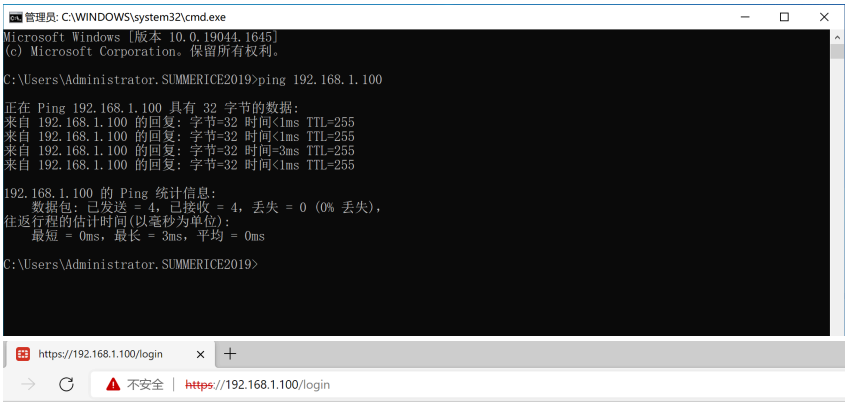
4. 通过内网 PC 尝试登陆 FortiGate 的 Manage IP：192.168.1.100，此时 FortiGate 的 port1 未开启 HTTP/HTTPS/PING，所以是无法访问的。



5. MGMT 口默认有管理权限，要通过 port1（LAN）接口管理设备为例，开启 port1（LAN）管理 FGT 的命令如下。允许网管协议从 Port1 接口通过 https/http/SSH/Ping 访问透明模式的 FortiGate。

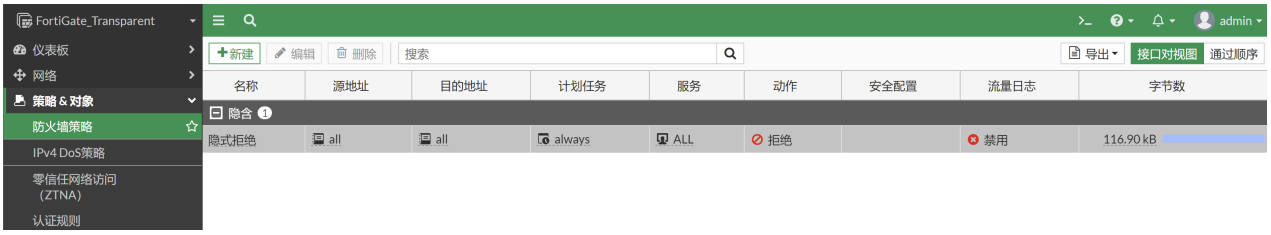
```bash
config system interface
    edit port1
        set allowaccess https http ping ssh
    next
end
```

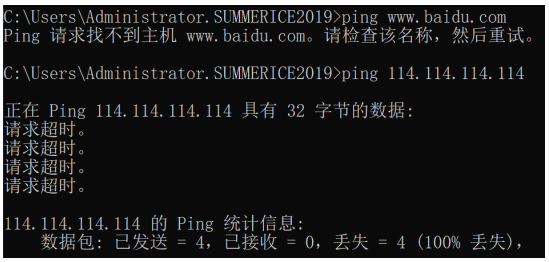
6. 使用 PC PING FortiGate 的管理 IP，并使用 HTTPS 登陆管理页面。





7. 配置安全策略允许内网 PC 访问互联网，默认情况下透明模式的 FortiGate 没有开启任何策略，此时 TCP/UDP/ICMP 的流量是全部不通的。

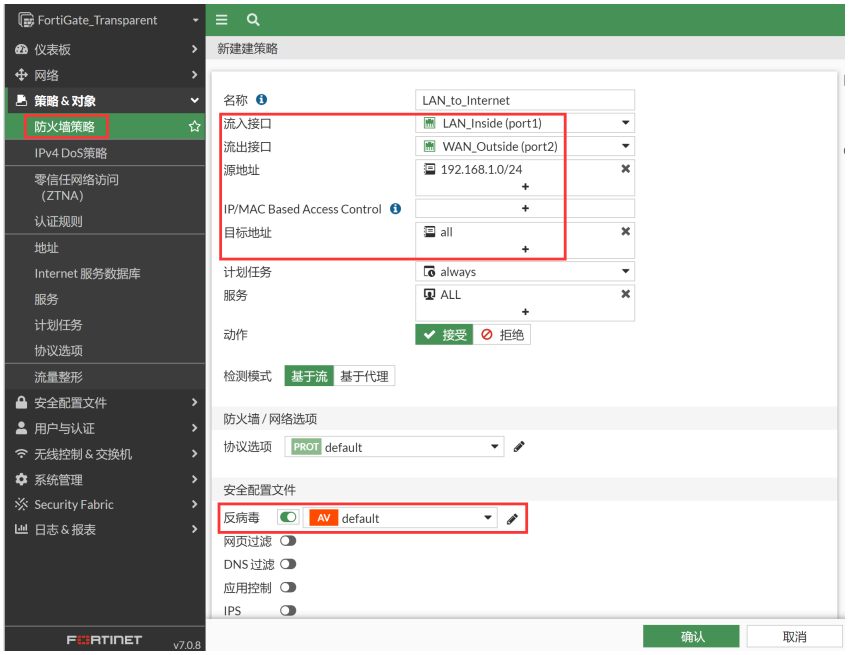




8. 建议先给接口一个别名，这样配置策略的时候可以很清楚的知道接口的位置和功能。



9. 进入策略 & 对象 → 防火墙策略，新建防火墙策略，放通 port1 到 port2 的流量，源地址创建 LAN 网段的地址对象并引用，开启防病毒功能，这里使用默认的防病毒配置文件。



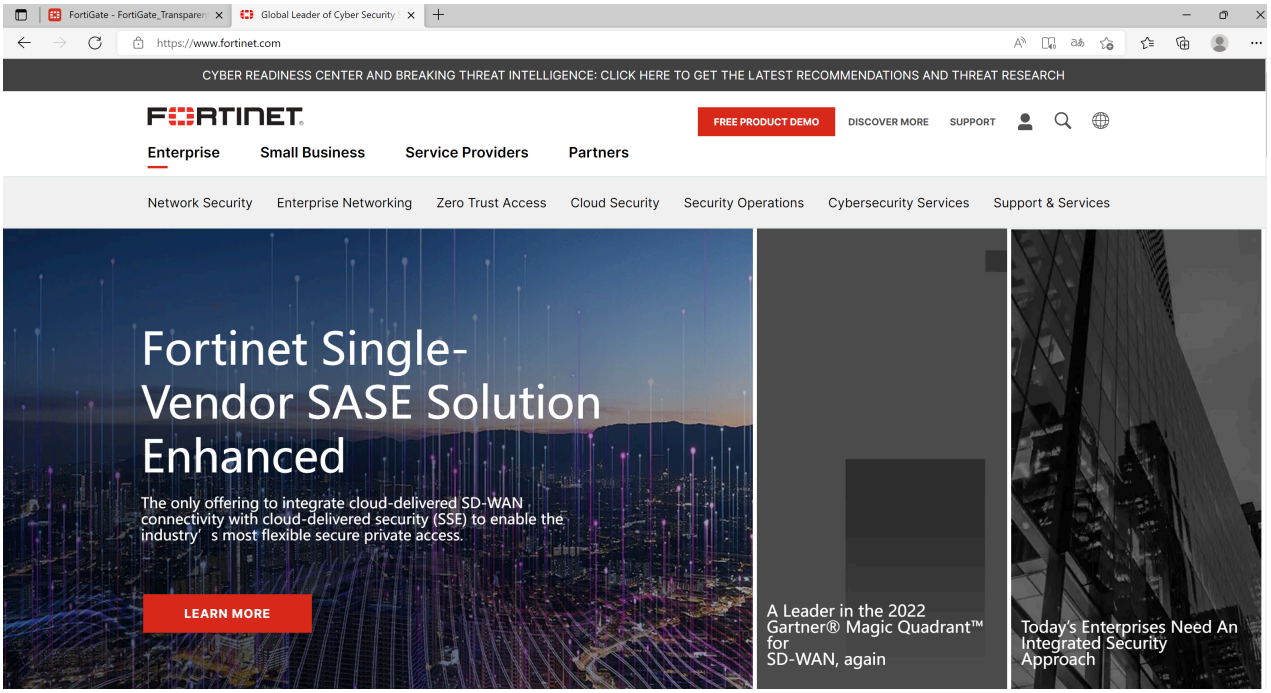
重要

建议在不需要的情况下关闭接口配置下的“设备探测”（Device detection）功能，该功能用于 MAC 地址厂商设备信息识别及 MAC 地址过滤，会消耗较多的设备资源。

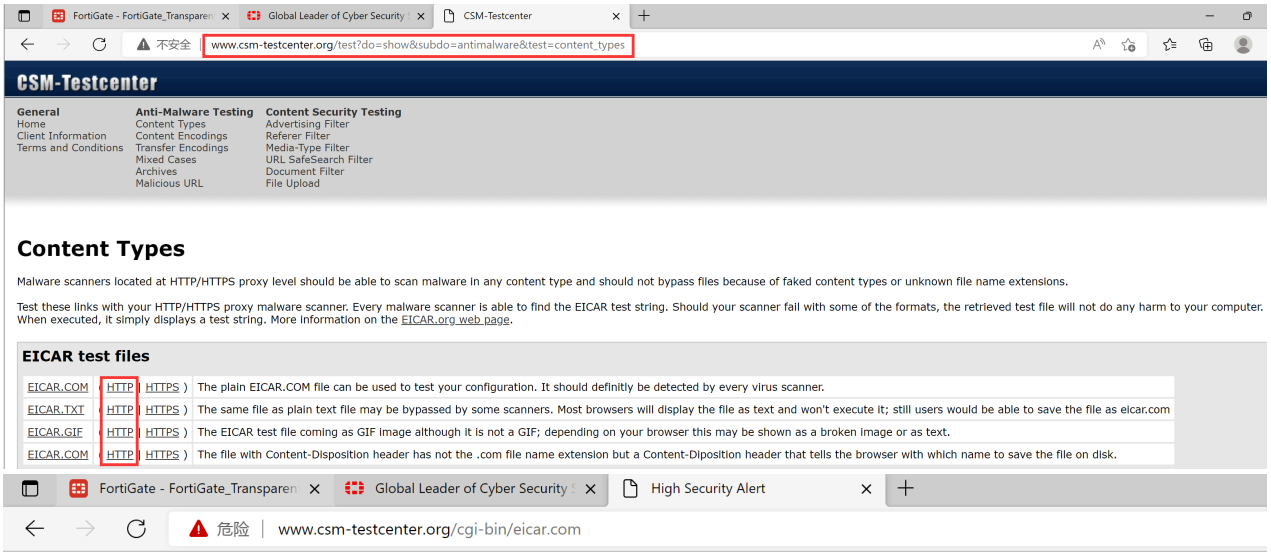
结果验证

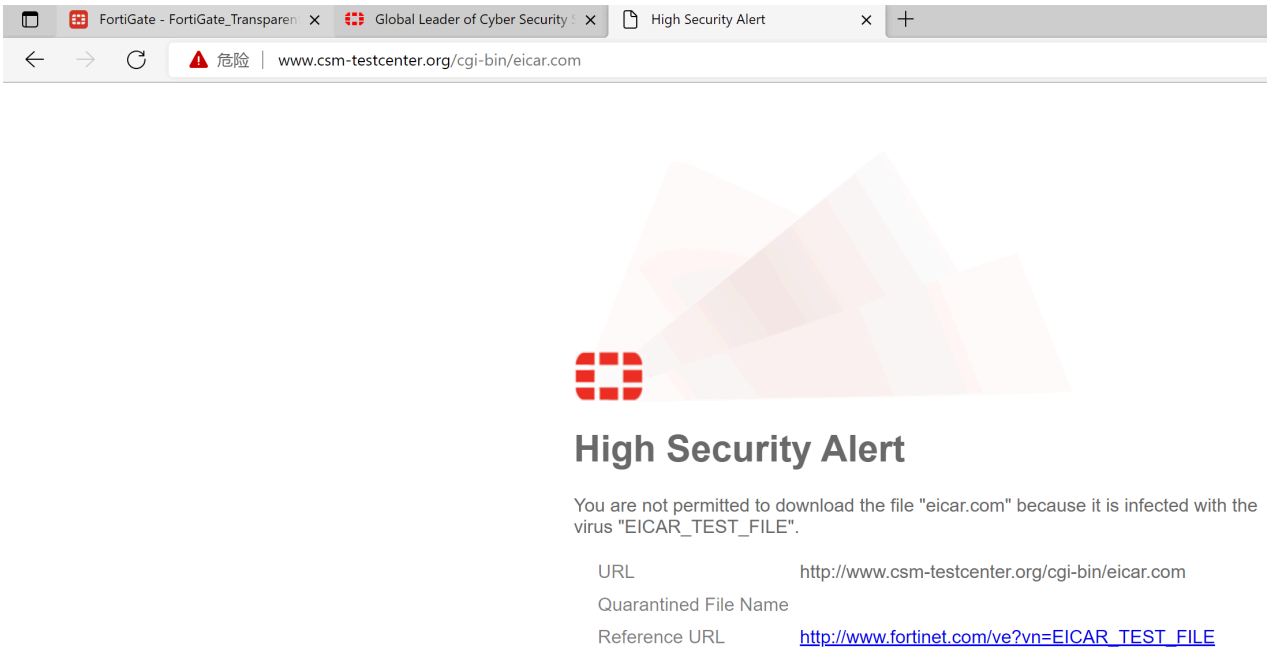
1. 使用内网 PC 测试 Internet 业务。

```
C:\Users\Administrator.SUMMERICE2019>ping 114.114.114.114

正在 Ping 114.114.114.114 具有 32 字节的数据:
来自 114.114.114.114 的回复: 字节=32 时间=21ms TTL=84
来自 114.114.114.114 的回复: 字节=32 时间=19ms TTL=76
来自 114.114.114.114 的回复: 字节=32 时间=23ms TTL=69
来自 114.114.114.114 的回复: 字节=32 时间=19ms TTL=85

114.114.114.114 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 19ms, 最长 = 23ms, 平均 = 20ms

C:\Users\Administrator.SUMMERICE2019>ping www.baidu.com

正在 Ping www.a.shifen.com [110.242.68.4] 具有 32 字节的数据:
来自 110.242.68.4 的回复: 字节=32 时间=14ms TTL=48
来自 110.242.68.4 的回复: 字节=32 时间=13ms TTL=48
来自 110.242.68.4 的回复: 字节=32 时间=12ms TTL=48
来自 110.242.68.4 的回复: 字节=32 时间=12ms TTL=48

110.242.68.4 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 12ms, 最长 = 14ms, 平均 = 12ms
```

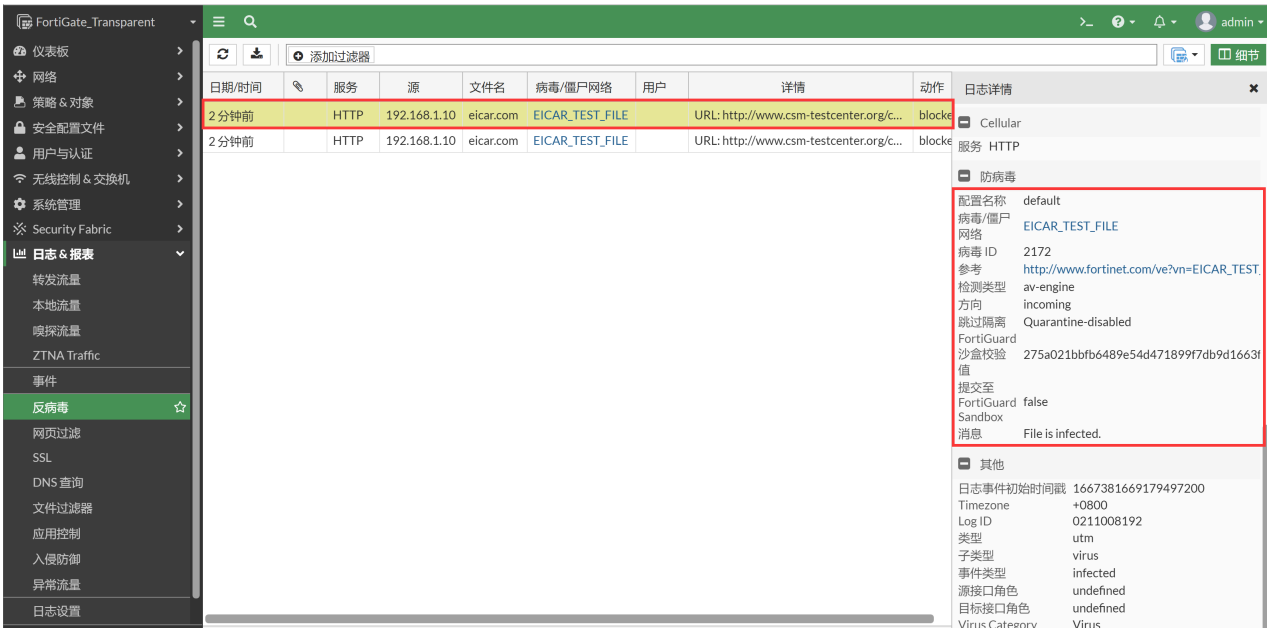


2. 使用 PC 测试通过 HTTP 协议下载病毒测试文件，可以看到下载动作被 FortiGate 拦截。

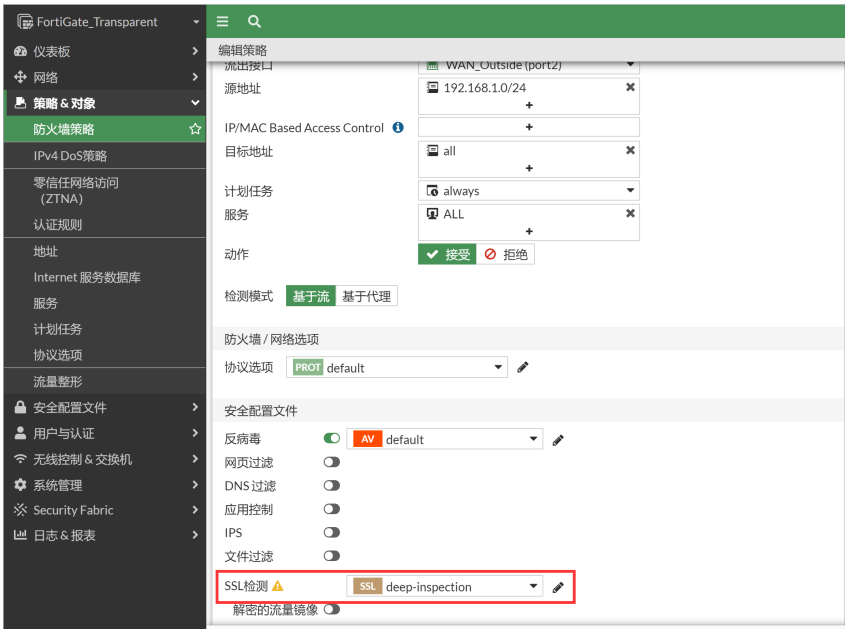




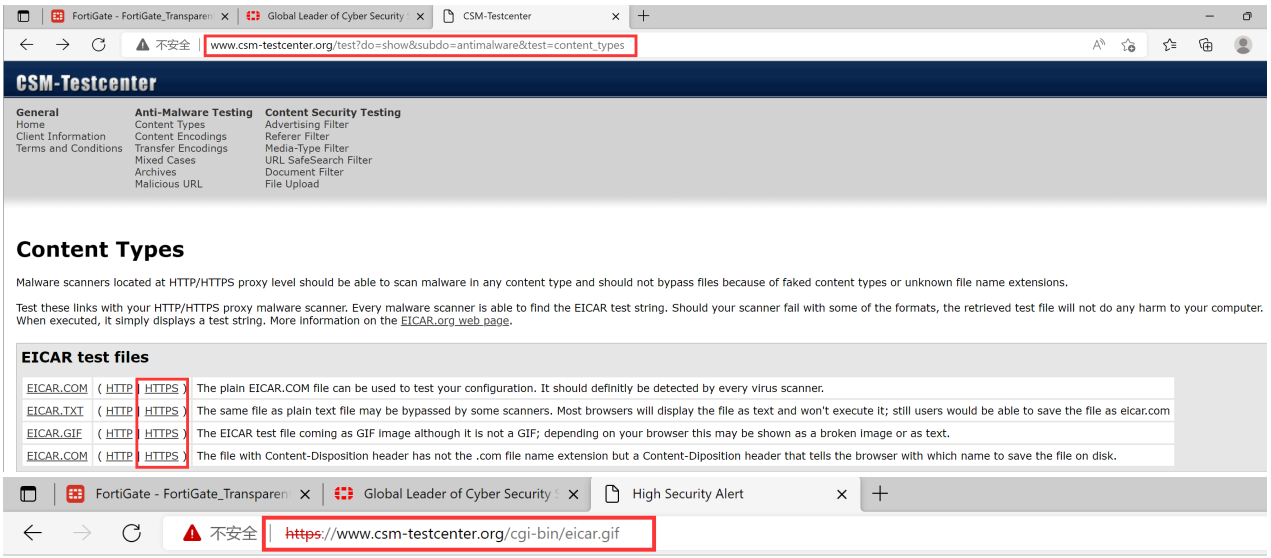
3. 在日志 & 报表 → 反病毒中可以看到相关拦截日志，协议为 HTTP。

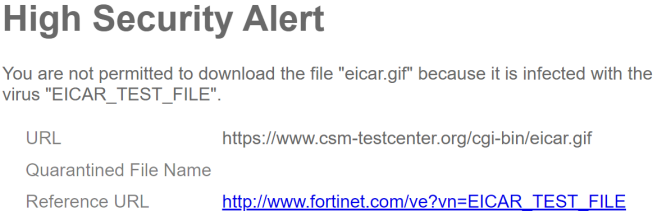


4. 如果是通过 HTTPS 协议下载病毒文件，由于数据被 HTTPS 加密，需要在策略中调用 SSL 深度检测。

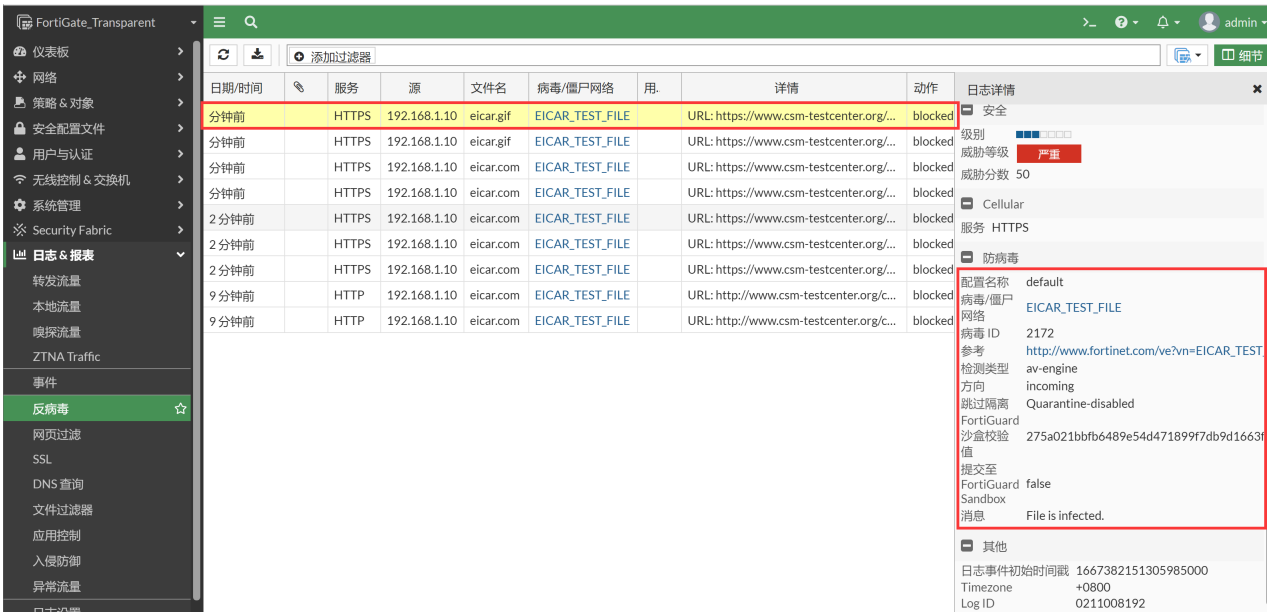


5. 使用 PC 测试通过 HTTP 协议下载病毒测试文件，可以看到下载动作被 FortiGate 拦截。





6. 在日志 & 报表 → 反病毒中可以看到相关拦截日志，协议为 HTTPS。

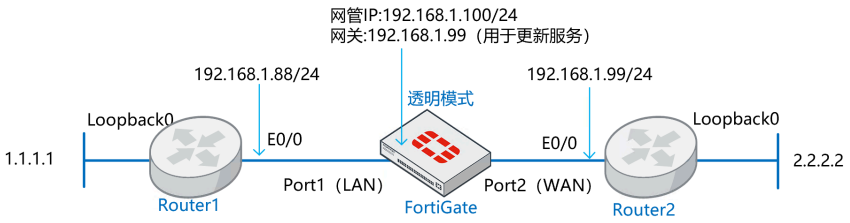

FortiGate_Transparent
admin
仪表板
网络
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
转发流量
本地流量
嗅探流量
ZTNA Traffic
事件
反病毒
网页过滤
SSL
DNS 查询
文件过滤器
应用控制
入侵防御
异常流量
添加过滤器
细节
日期/时间 服务 源 文件名 病毒/僵尸网络 用. 详情 动作
分钟前 HTTPS 192.168.1.10 eicar.gif EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
分钟前 HTTPS 192.168.1.10 eicar.gif EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
分钟前 HTTPS 192.168.1.10 eicar.com EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
分钟前 HTTPS 192.168.1.10 eicar.com EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
2 分钟前 HTTPS 192.168.1.10 eicar.com EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
2 分钟前 HTTPS 192.168.1.10 eicar.com EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
2 分钟前 HTTPS 192.168.1.10 eicar.com EICAR_TEST_FILE URL: https://www.csm-testcenter.org/... blocked
9 分钟前 HTTP 192.168.1.10 eicar.com EICAR_TEST_FILE URL: http://www.csm-testcenter.org/c... blocked
9 分钟前 HTTP 192.168.1.10 eicar.com EICAR_TEST_FILE URL: http://www.csm-testcenter.org/c... blocked
日志详情
安全
级别
威胁等级 严重
威胁分数 50
Cellular
服务 HTTPS
防病毒
配置名称 default
病毒/僵尸网络 EICAR_TEST_FILE
病毒 ID 2172
参考 http://www.fortinet.com/ve?vn=EICAR_TEST
检测类型 av-engine
方向 incoming
跳过隔离 Quarantine-disabled
FortiGuard 沙盒校验值 275a021bbfb6489e54d471899f7db9d1663f
提交至 FortiGuard Sandbox false
消息 File is infected.
其他
日志事件初始时间戳 1667382151305985000
Timezone +0800
Log ID 0211008192

组播 (OSPF) 与透明模式

网络需求

路由器 1 和路由器 2 之间运行了动态路由协议，防火墙以透明模式运行在两台路由器之间，需要确保两台路由器之间的动态路由协议运行正常，并且数据可通。

网络拓扑



R1和R2之间运行了RIP/OSPF/EIGRP协议

配置要点

- Router1/Router2 路由器的基础配置
- 将防火墙配置为透明模式并开启网管
- 默认不配置策略情况下，观察动态路由协议的情况
- 配置安全策略解决 OSPF/EIGRP 邻居关系异常

配置步骤和结果验证

1. Router1/Router2 路由器的基础配置。

   Router1：

```
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
ip address 192.168.1.88 255.255.255.0
no shutdown
!
router ospf 10
network 1.1.1.1 0.0.0.0 area 0
network 192.168.1.0 0.0.0.255 area 0
!
router rip
version 2
network 1.0.0.0
network 192.168.1.0
no auto-summary
!
router eigrp 10
network 1.1.1.1 0.0.0.0
network 192.168.1.0
```

   Router2：

```
interface Loopback0
ip address 2.2.2.2 255.255.255.255
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
!
```

```
router ospf 10
network 2.2.2.2 0.0.0.0 area 0
network 192.168.1.0 0.0.0.255 area 0
!
router rip
version 2
network 2.0.0.0
network 192.168.1.0
no auto-summary
!
router eigrp 10
network 2.2.2.2 0.0.0.0
network 192.168.1.0
```

2. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。配置可以管理防火墙的本地 IP 和网关。

```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 192.168.1.100 255.255.255.0
    set gateway 192.168.1.99
end
```

3. MGMT 口默认有管理权限，要通过 port1（LAN）接口管理设备为例，开启 port1（LAN）管理 FGT 的命令如下。允许网管协议从 Port1 接口通过 https/http/SSH/Ping 访问透明模式的 FortiGate。

```bash
config system interface
    edit port1
    edit allowaccess https http ping ssh
end
```

4. 默认不配置策略情况下，观察动态路由协议的情况，RIP 邻居是正常的，但是 OSPF 邻居无法建立成功，EIGRP 邻居路由学习异常，邻居反复的震荡。

```
Router1#show ip rip database
1.0.0.0/8    auto-summary
1.1.1.1/32    directly connected, Loopback0
2.0.0.0/8    auto-summary
2.2.2.2/32
    [1] via 192.168.1.99, 00:00:01, Ethernet0/0
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, Ethernet0/0

Router1#show ip route
...
R        2.2.2.2 [120/1] via 192.168.1.99, 00:00:03, Ethernet0/0
...
```

```
Router1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 10)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         274         0×80000003 0×00F9A5 2

Router1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   EXSTART/DR      00:00:31    192.168.1.99    Ethernet0/0

Router1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   INIT/DROTHER    00:00:37    192.168.1.99    Ethernet0/0

*Jul  2 07:55:15.999: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on Ethernet0/0 from EXSTART to DOWN, Neighbor Down: Too
many retransmissions
*Jul  2 07:56:16.002: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on Ethernet0/0 from DOWN to DOWN, Neighbor Down: Ignore
timer expired
```

```
Router1#
*Jul  2 07:55:24.200: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 192.168.1.99 (Ethernet0/0) is down: retry limit
exceeded
*Jul  2 07:55:28.663: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 192.168.1.99 (Ethernet0/0) is up: new adjacency
```

重要

原因分析：

我们知道 RIP/OSPF/EIGRP 都是用组播 IP 来建立邻居关系：RIP:224.0.0.9，OSPF:224.0.0.5/224.0.0.6，EIGRP:224.0.0.10。

对应 FGT 来说默认情况下：

如果组播是不通，为何 RIP 协议又可以正常使用？

如果组播是通的，为何 OSPF/EIGRP 邻居关系异常呢？

怎么看都是有点矛盾的！

5. 让我们带着这个疑惑来分析一下原因，首先，我们来看 FortiGate 透明模式下组播到底通还是不通，答案是：通的。原因是默认情况下有两个系统默认的设置允许组播通过 FortiGate 的。

   配置 1：系统设置：

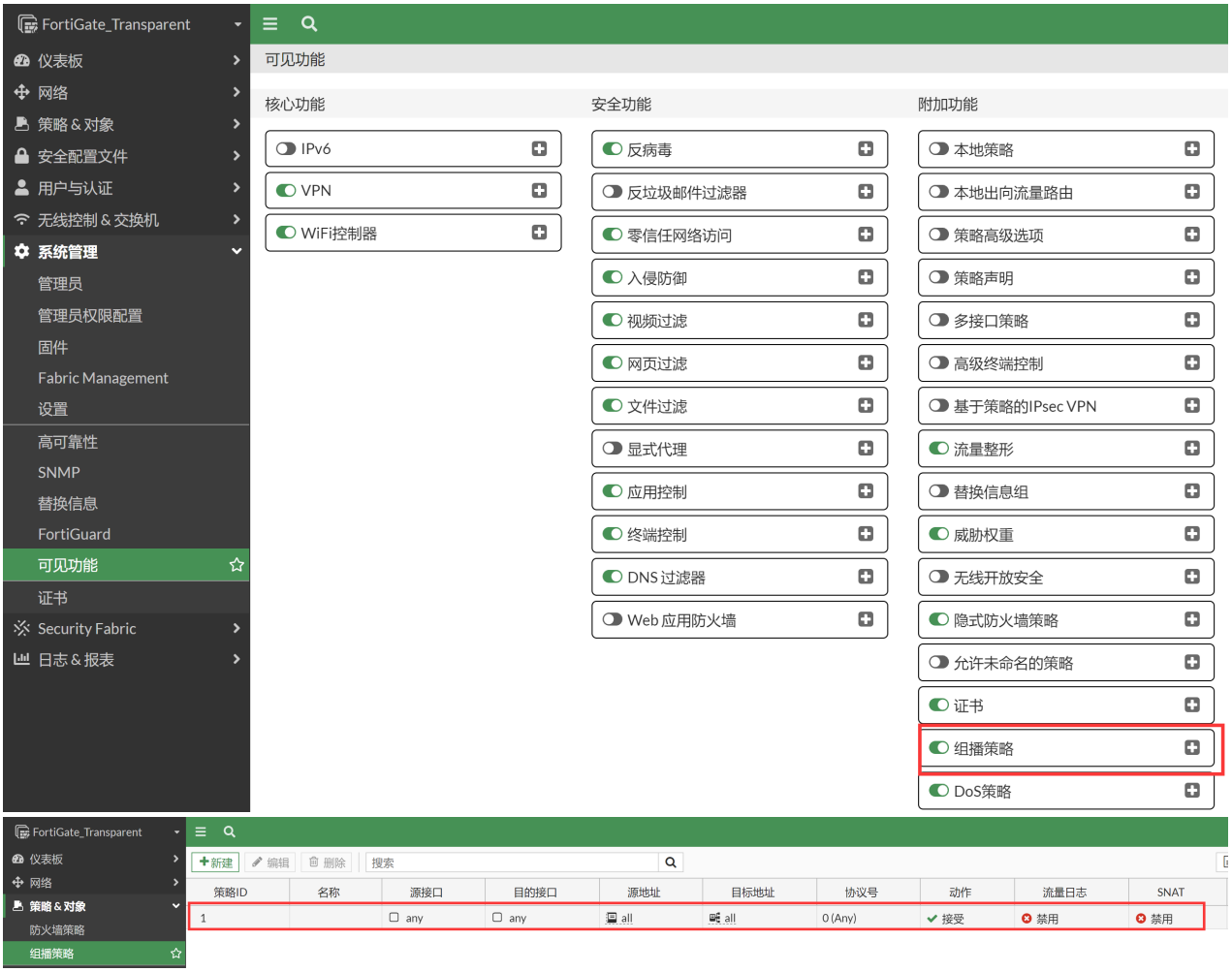
```
FortiGate_Transparent # config system settings
FortiGate_Transparent (settings) # show full-configuration | grep multicast
    set multicast-skip-policy disable    //默认为disable，组播流量默认需要查询策略。如果修改为enable，则会bypass掉组播流量，组播流量
也是可以通的，但是FGT上不会建立组播的会话
    set gui-multicast-policy enable
```

   配置 2: 组播策略：

   透明模式下的 FortiGate 默认存在有一条 permit any 到 any 全通的组播策略。

```
FortiGate_Transparent # config firewall multicast-policy
FortiGate_Transparent (multicast-policy) # show
config firewall multicast-policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

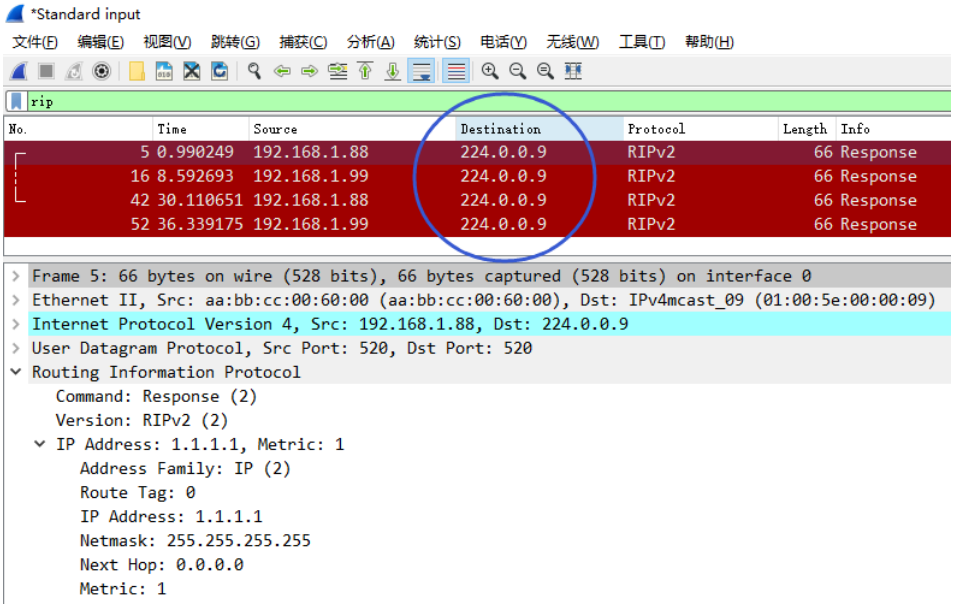
6. 在 GUI 中查看组播策略（需要在系统管理 → 可见功能中打开组播策略显示），可以看到这条默认放通的组播策略。



7. 通过上述两个默认的配置，可以看出 FGT 的组播默认就是通的。在 FGT 查看组播会话，这样就可以解释通为什么 RIP 路由协议是 OK 的了，抓包可以看到 RIP 协议只有 224.0.0.9 的组播报文。

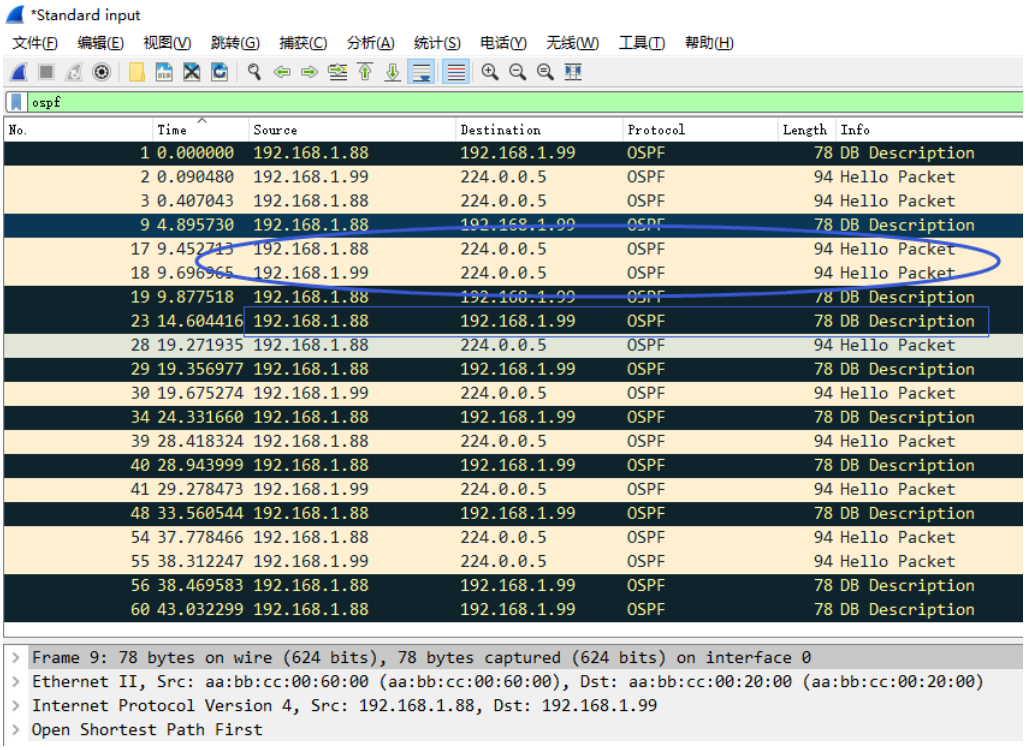
```
FortiGate_Transparent # diagnose sys mcast-session list
session info: id=6 vf=0 proto=17 192.168.1.1.520→224.0.0.9.520    //RIP会话
used=2 path=3 duration=137 expire=177 indev=3 pkts=6 bytes=312
state=00000002:tp
path: policy=1, outdev=4

session info: id=4 vf=0 proto=17 192.168.1.99.520→224.0.0.9.520    //RIP会话
used=2 path=3 duration=1035 expire=174 indev=4 pkts=38 bytes=1976
state=00000002:tp
path: policy=1, outdev=3
Total 6 sessions

session info: id=3 vf=0 proto=89 192.168.1.1.0→224.0.0.5.0    //OSPF会话
used=2 path=3 duration=1037 expire=179 indev=3 pkts=110 bytes=8720
state=00000002:tp
path: policy=1, outdev=4

session info: id=5 vf=0 proto=89 192.168.1.99.0→224.0.0.5.0    //OSPF会话
used=2 path=3 duration=1034 expire=176 indev=4 pkts=110 bytes=8688
state=00000002:tp
path: policy=1, outdev=3

session info: id=7 vf=0 proto=88 192.168.1.1.0→224.0.0.10.0    //EIGRP会话
used=2 path=3 duration=22 expire=175 indev=3 pkts=6 bytes=410
state=00000002:tp
path: policy=1, outdev=4
```

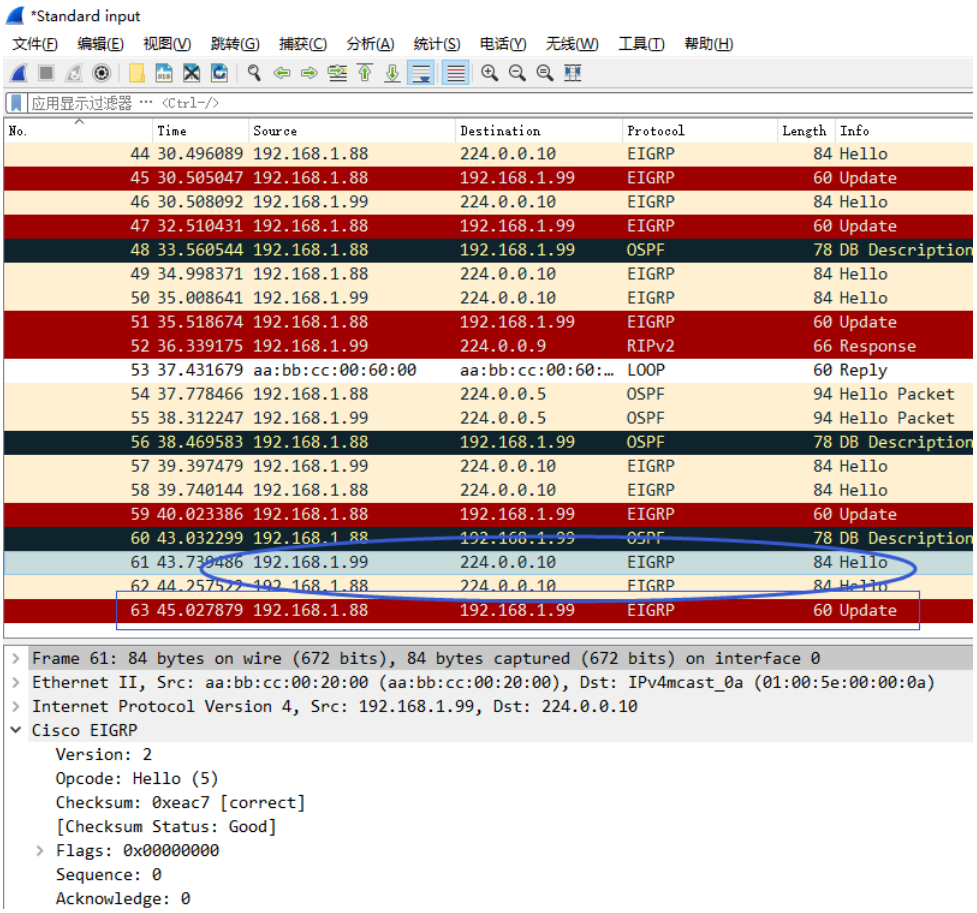
```
session info: id=2 vf=0 proto=88 192.168.1.99.0→224.0.0.10.0    //EIGRP会话
used=2 path=3 duration=1041 expire=176 indev=4 pkts=226 bytes=13610
state=00000002:tp
path: policy=1, outdev=3
```



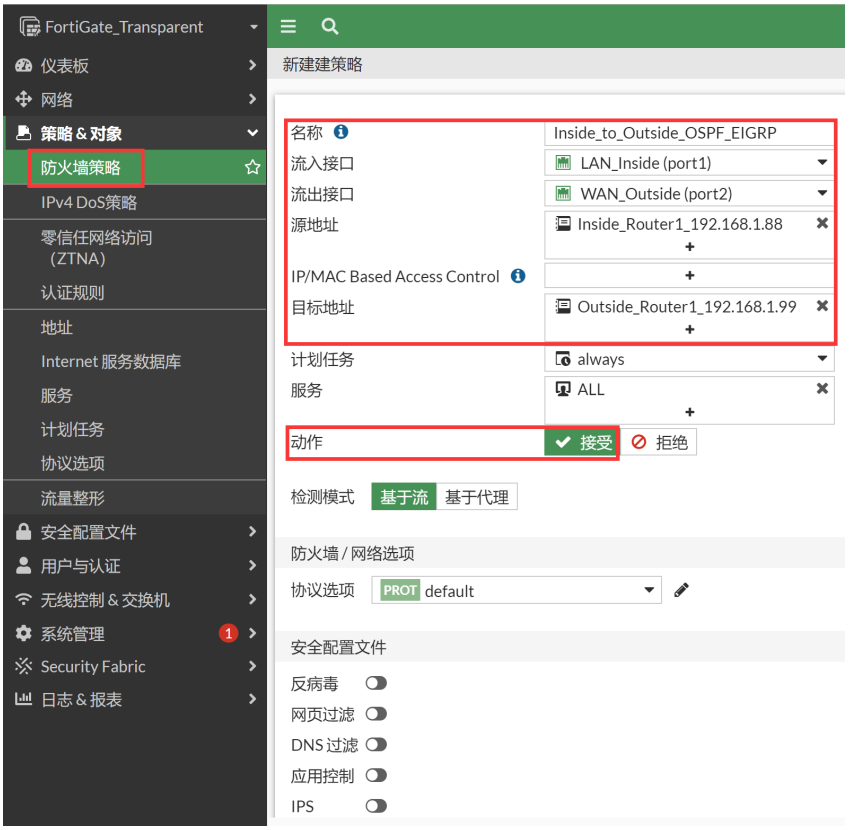
8. 既然 RIP 路由协议可以正常使用，那为什么 OSPF 和 EIGRP 却不行呢？进一步的分析你会发现，OSPF 和 EIGRP 的组播其实是正常转发的，但是 OSPF 和 EIGRP 建立邻居关系不仅仅只使用了组播，而且还使用单播，默认情况下没有任何策略情况下，单播的数据会被默认丢弃，因此邻居关系异常。

> **重要**
>
> OSPF 的 DB Description 报文使用单播，EIGRP 的 Update 报文使用单播。这是 OSPF 和 EIGRP 邻居关系异常的真正原因。

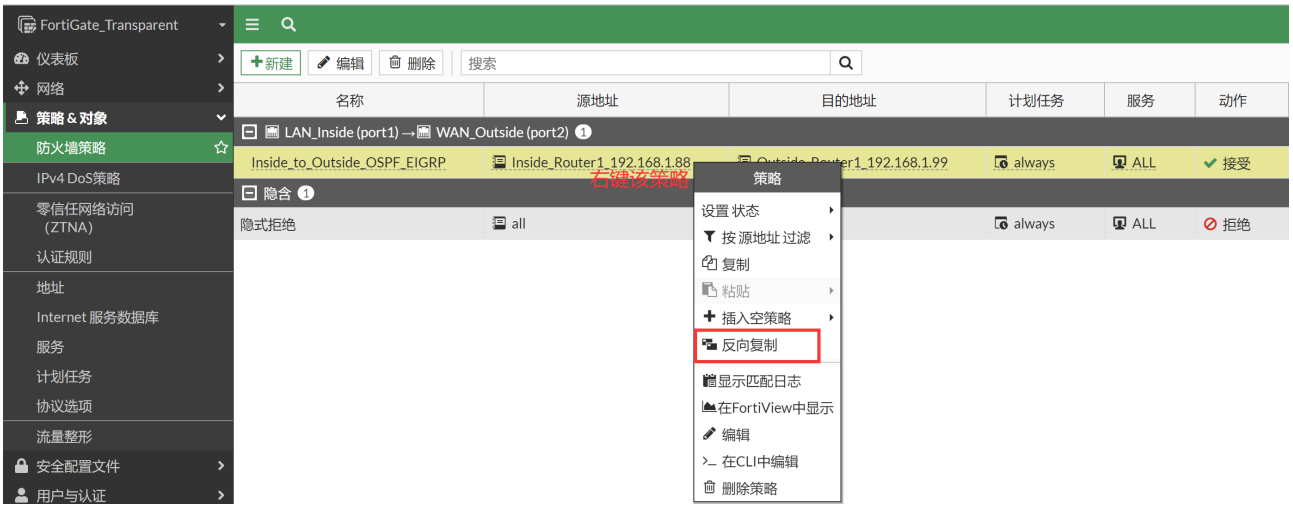


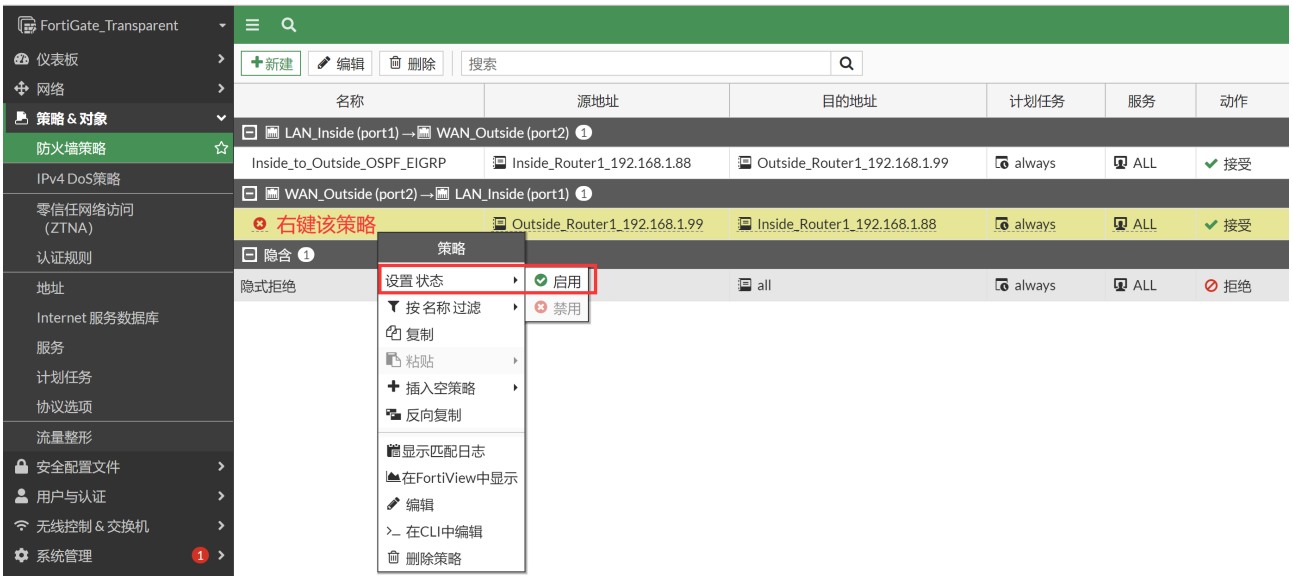


9. 既然原因清楚了，那么要解决这个问题就很简单了。只需要配置单播安全策略解决 OSPF/EIGRP 邻居关系异常即可。首先创建 port1 → port2 的单播放通策略，源目为 Router1 和 Router2 的 IP 地址。



0. 创建完成后，右键点击该策略，选择创建反向策略，放通 Router2 到 Router1 的单播流量，并启用该策略（自行修改名称）。







1. 上述两条策略配置完毕，会发现 RIP/OSPF/EIGRP 的邻居关系都已经正常建立，路由学习也正常。

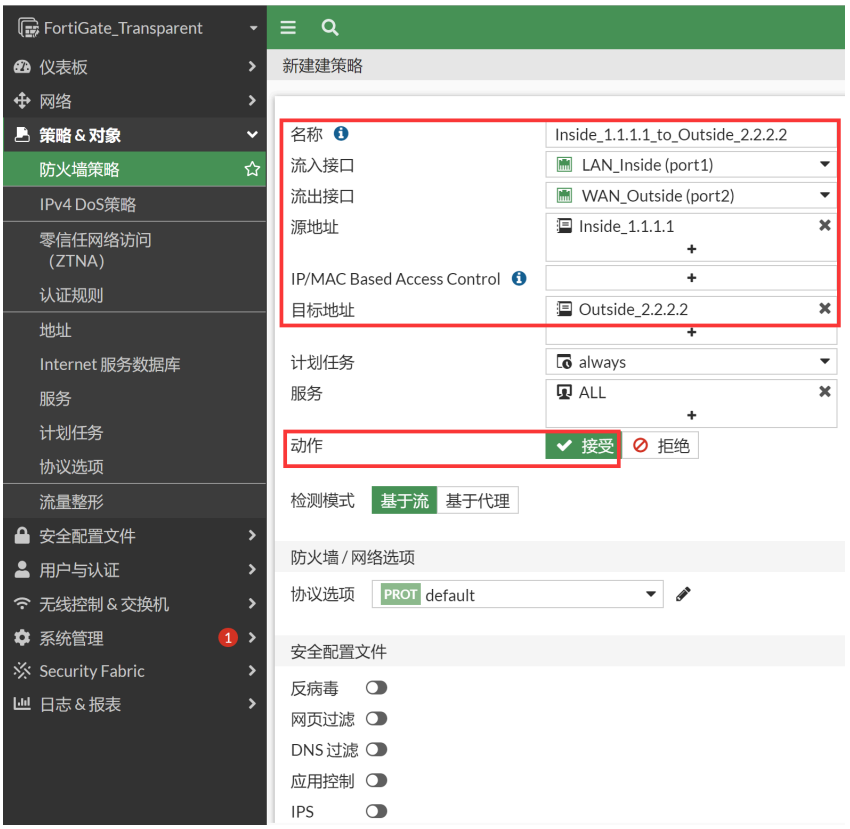
```
Router1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/DR         00:00:34    192.168.1.99    Ethernet0/0

Router1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(10)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   192.168.1.99            Et0/0                    13 00:03:06   23   138  0  35

Router1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(10)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 192.168.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 2.2.2.2/32, 1 successors, FD is 409600
        via 192.168.1.99 (409600/128256), Ethernet0/0
P 1.1.1.1/32, 1 successors, FD is 128256
```

```
        via Connected, Loopback0

Router1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 10)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         252         0x80000005 0x0098DF 2
2.2.2.2         2.2.2.2         253         0x80000005 0x008DD3 2
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.99    2.2.2.2         253         0x80000001 0x003B1D

Router1#show ip rip  database
1.0.0.0/8    auto-summary
1.1.1.1/32    directly connected, Loopback0
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, Ethernet0/0

Router1#show ip route
...
D        2.2.2.2 [90/409600] via 192.168.1.99, 00:02:51, Ethernet0/0
...
```

2. 此时使用 Router1 的 1.1.1.1 Ping Router2 的 2.2.2.2，是不通的，因为 FortiGate 还未放通 1.1.1.1 和 2.2.2.2 之间的单播流量。

```
Router1#ping 2.2.2.2 source  1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)
```

3. 配置单播安全策略允许 Router1 的 1.1.1.1 访问 Router2 的 2.2.2.2。



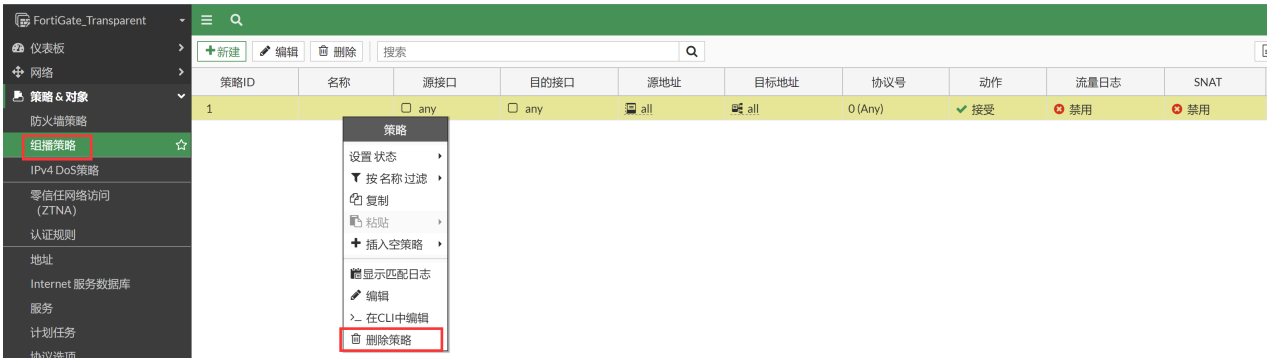
4. 再次测试 Router1 的 1.1.1.1 Ping Router2 的 2.2.2.2。

```
Router1#ping 2.2.2.2 source  1.1.1.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/9 ms

FortiGate抓包:
FortiGate_Transparent # diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
5.551382 port1 in 1.1.1.1 → 2.2.2.2: icmp: echo request
5.551455 port2 out 1.1.1.1 → 2.2.2.2: icmp: echo request
5.552281 port2 in 2.2.2.2 → 1.1.1.1: icmp: echo reply
5.552292 port1 out 2.2.2.2 → 1.1.1.1: icmp: echo reply
```

补充 1

如果组播不希望默认全通，而需要策略明确的策略控制，怎么操作?

1. 进入 FortiGate 的组播策略页面，删除组播策略的默认放通策略。



2. 配置明细的组播策略代替 any 到 any 全通的默认组播策略即可。



补充 2

如果不想配置组播策略，可以全局系统设置中 bypass 组播流量。

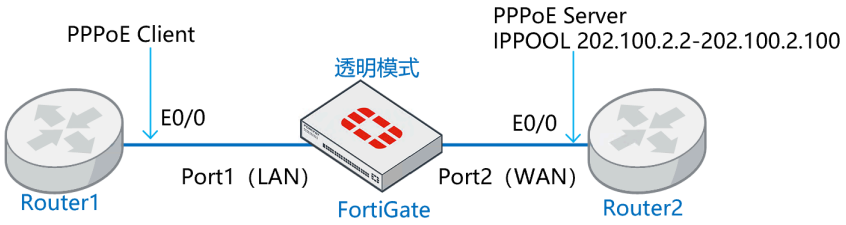
```
FortiGate_Transparent # config system settings
FortiGate_Transparent (settings) # set multicast-skip-policy
enable     Allowing multicast traffic through the FortiGate without creating a multicast firewall policy.
disable    Require a multicast policy to allow multicast traffic to pass through the FortiGate.    //默认是disable
FortiGate_Transparent (settings) # set multicast-skip-policy enable
FortiGate_Transparent (settings) # end
```

非 IP 流量 (PPPOE) 与透明模式

网络需求

路由器 1 和路由器 2 之间运行了 PPPOE 协议（非 IP 协议），防火墙以透明模式运行在两台路由器之间，需要确保路由器 1 的 PPPOE 拨号正常。

网络拓扑



R1和R2之间运行了PPPoE拨号
R2为PPPoE Server R1为PPPPoE Client

配置要点

- Router1/Router2 路由器的基础配置
- 将防火墙配置为透明模式并开启网管
- 默认情况下，观察 PPPOE 拨号的情况
- 配置 l2forward 解决非 IP 流量转发问题

配置步骤与结果验证

1. Router1/Router2 路由器的基础配置。

   Router1 (PPPOE Client)：

```
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
no ip address
no shutdown
pppoe enable group global
pppoe-client dial-pool-number 1
!
interface Dialer1
mtu 1492
ip address negotiated
encapsulation ppp
dialer pool 1
ppp pap sent-username pppoeuser1 password 0 fortinet
ppp ipcp route default
```

   Router2 (PPPOE Server):

```
interface Loopback0
ip address 2.2.2.2 255.255.255.255
!
username pppoeuser1 password 0 fortinet
!
bba-group pppoe bba1
virtual-template 1
!
interface Loopback1
ip address 202.100.2.1 255.255.255.0
```

```
!
interface Ethernet0/0
no ip address
no shutdown
pppoe enable group bba1
!
interface Virtual-Template1
description PPPOE bba1
mtu 1492
ip unnumbered Loopback1
peer default ip address pool pool1
ppp authentication pap
!
ip local pool pool1 202.100.2.2 202.100.2.100
```

2. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。配置可以管理防火墙的本地 IP 和网关。

```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 192.168.1.100 255.255.255.0
    set gateway 192.168.1.99
end
```

3. MGMT 口默认有管理权限，要通过 port1（LAN）接口管理设备为例，开启 port1（LAN）管理 FGT 的命令如下。允许网管协议从 Port1 接口通过 https/http/SSH/Ping 访问透明模式的 FortiGate。

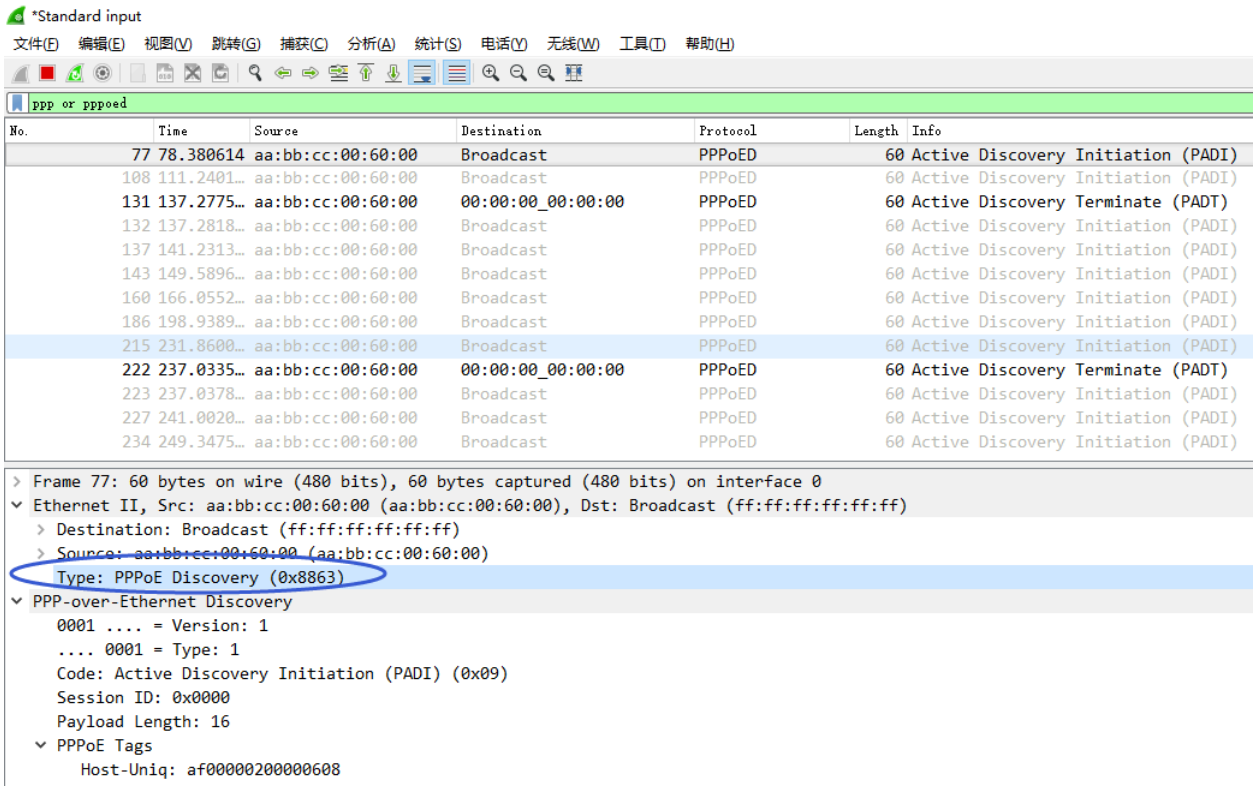
```bash
config system interface
    edit port1
    edit allowaccess https http ping ssh
end
```

4. 默认情况下，观察 PPPOE 拨号的情况，Router1（PPPOE Client）无法 PPPPOE 拨号成功。

```
Router1#show ip route
...
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
...
Router1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                unassigned      YES manual up                    up
Ethernet0/1                unassigned      YES unset  administratively down down
Ethernet0/2                unassigned      YES unset  administratively down down
Ethernet0/3                unassigned      YES unset  administratively down down
Dialer1                    unassigned      YES IPCP   up                    up
Loopback0                  1.1.1.1         YES manual up                    up
Virtual-Access1            unassigned      YES unset  up                    up
Virtual-Access2            unassigned      YES unset  down                  down
```

5. 通过在 FGT 上抓包可以看到 FGT 并不转发 PPPOE 的报文，非 IP 的 PPPOE 数据默认不会被 FGT 所转发。

```
FortiGate_Transparent # diagnose sniffer packet any "pppoed" 4
interfaces=[any]
filters=[pppoed]
4.124126 port1 in pppoe printer hasn't been added to sniffer
37.014209 port1 in pppoe printer hasn't been added to sniffer
69.908430 port1 in pppoe printer hasn't been added to sniffer
102.798634 port1 in pppoe printer hasn't been added to sniffer
135.705063 port1 in pppoe printer hasn't been added to sniffer
```

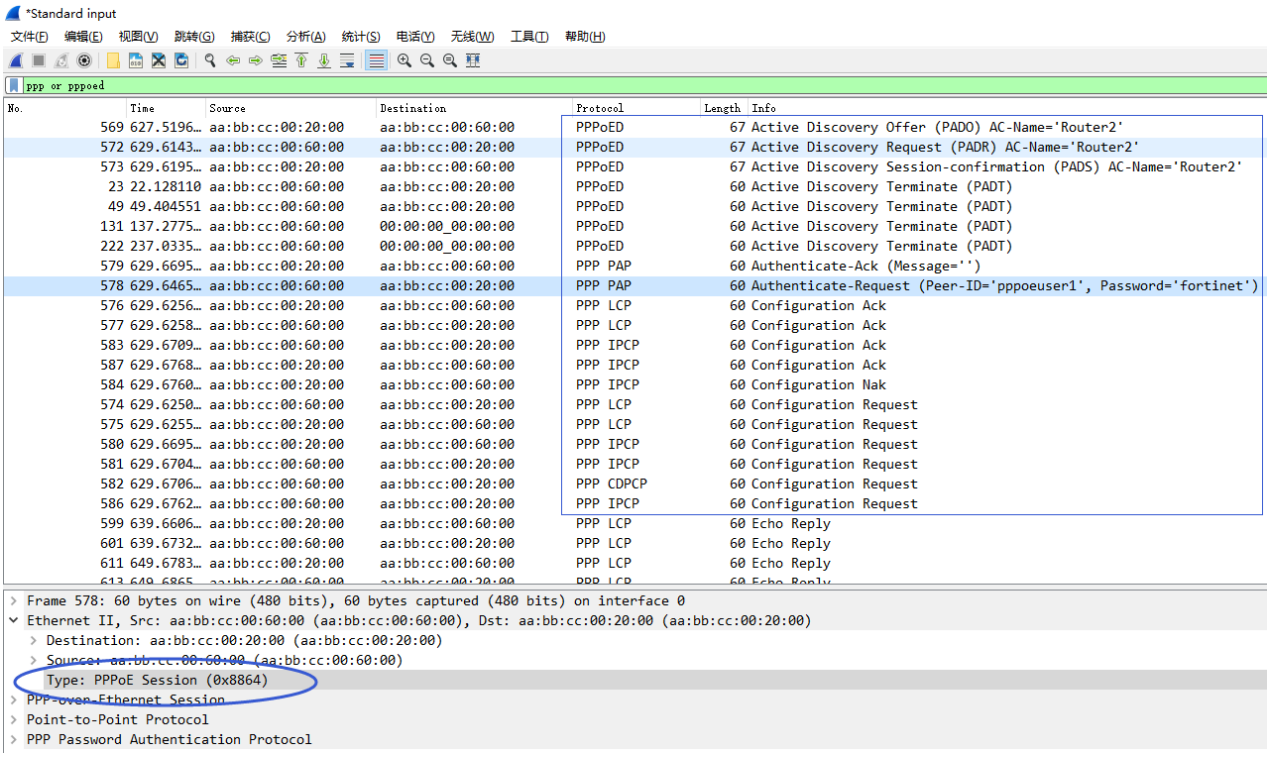


6. 解决非 IP 流量转发问题，在 FGT 的二层接口下配置 set l2forward enable 即可！

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port1
FortiGate_Transparent (port1) # set l2forward enable

FortiGate_Transparent (port1) # next
FortiGate_Transparent (interface) # edit port2
FortiGate_Transparent (port2) # set l2forward enable
FortiGate_Transparent (port2) # end
```

7. 再次抓包查看，此时 FGT 可以转发 PPPOE 请求了。

```
FortiGate_Transparent # diagnose sniffer packet any "pppoed" 4
interfaces=[any]
filters=[pppoed]
5.324865 port1 in pppoe printer hasn't been added to sniffer
5.324880 port2 out pppoe printer hasn't been added to sniffer
5.325468 port2 in pppoe printer hasn't been added to sniffer
5.325474 port1 out pppoe printer hasn't been added to sniffer
```



8. 查看 Router1 的 Router1 PPPOE 拨号情况，可以看到拨号成功了。

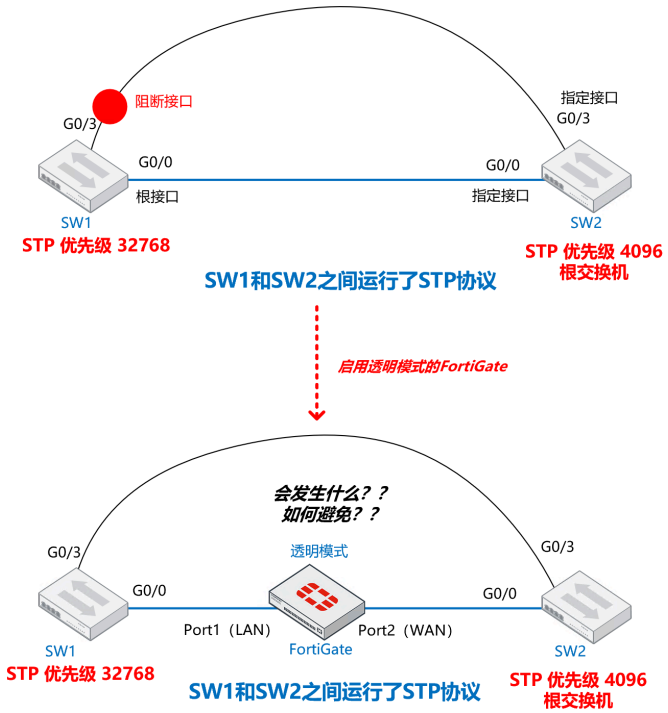
```
Router1#show ppp all
Interface/ID OPEN+ Nego* Fail-     Stage    Peer Address    Peer Name
------------ ------------------- -------- --------------- --------------------

Vi2          LCP+ IPCP+ CDPCP-     LocalT   202.100.2.1

Router1#show pppoe session
     1 client session
Uniq ID  PPPoE  RemMAC          Port                   VT  VA         State
           SID  LocMAC                                    VA-st      Type
    N/A      2  aabb.cc00.2000  Et0/0                  Di1 Vi2        UP
                aabb.cc00.6000                             UP

Router1#show  ip route
 ...
Gateway of last resort is 202.100.2.1 to network 0.0.0.0
S*    0.0.0.0/0 [1/0] via 202.100.2.1
 ...
C        202.100.2.1 is directly connected, Dialer1
C        202.100.2.3 is directly connected, Dialer1

Router1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
 ...
Dialer1                    202.100.2.3     YES IPCP   up                    up
 ...
```

生成树 (STP) 与透明模式

网络需求

SW1 和 SW2 之间运行了 STP 协议，防火墙以透明模式运行在两台 SW 之间，SW 和 SW 之间原本存在环路，通过 STP 进行隔离阻断了环路接口，在加入透明模式的 FGT 之后，需要继续确保 STP 工作正常。

网络拓扑



配置要点

- SW1/SW2 交换机的基础配置
- 将防火墙配置为透明模式并开启网管
- 默认情况下，观察 STP 报文的情况
- 配置 l2forward 解决 STP 流量转发问题

配置步骤与结果验证

1. SW1/SW2 交换机的基础配置。

   SW1：

   ```
   hostname SW1
   spanning-tree vlan 1 priority 32768
   ```

   SW2：

   ```
   hostname SW2
   !
   spanning-tree vlan 1 priority 4096
   ```

2. 查看没有加入透明模式的 FortiGate 之前 STP 状态，STP 选举和接口状态正常。

```
SW1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    Shr
Gi0/3               Altn BLK 4         128.4    Shr     //G0/3被阻断，STP工作正常

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr
```

3. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。配置可以管理防火墙的本地 IP 和网关。

```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 192.168.1.100 255.255.255.0
    set gateway 192.168.1.99
end
```

4. MGMT 口默认有管理权限，要通过 port1（LAN）接口管理设备为例，开启 port1（LAN）管理 FGT 的命令如下。允许网管协议从 Port1 接口通过 https/http/SSH/Ping 访问透明模式的 FortiGate。

```bash
config system interface
    edit port1
    edit allowaccess https http ping ssh
end
```

5. 加入 FGT 在没有任何配置的情况下，观察 STP 的运行情况，一旦 FGT 加入到 SW1 和 SW2 之间，很明显网络中出现了环路，设备操作很卡顿，SW1 和 SW2 基本上不再响应请求，三台设备均出现了异常情况（默认情况下透明模式的 FGT 只转发 ARP 报文，因此只需要一个 ARP 报文就会产生广播风暴，引起环路导致设备、业务中断）。

```
FortiGate_Transparent # diagnose sniffer packet any "" 4
interfaces=[any]
filters=[none]
0.814720 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.814733 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.817141 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.817148 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.818761 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.818771 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.819709 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
```

```
0.819720 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.821533 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.821544 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.825484 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.825492 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.827856 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.827866 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.828752 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.828762 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.829235 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.829244 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.831930 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.831947 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.834842 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.834853 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.836686 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.836694 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
0.838965 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.838972 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.841965 port1 in arp who-has 192.168.1.99 tell 192.168.1.100
0.841974 port2 out arp who-has 192.168.1.99 tell 192.168.1.100
0.843858 port2 in arp who-has 192.168.1.99 tell 192.168.1.100
0.843866 port1 out arp who-has 192.168.1.99 tell 192.168.1.100
```

6. 查看 SW1/SW2 的 STP 状态均异常，所有接口都处于转发状态。

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        4 (GigabitEthernet0/3)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Root FWD 4         128.4    Shr     //SW1的两个接口都处于转发状态

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr  //SW2的两个接口都处于转发状态
```

7. 在 FGT 上抓包可以看到 STP 的 BPDU 报文没有被转发，而是被默认丢弃，这是引起环路的根本原因。STP BPDU 数据默认不会被透明模式的 FGT 所转发，因此 STP 计算不正确，会计算不出网络中的环路，从而引起广播风暴。

```
FortiGate_Transparent # diagnose sniffer packet any "none" 4
interfaces=[any]
filters=[none]
1.234590 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
2.182694 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
3.247650 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
4.193908 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
5.260154 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
6.204003 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
```

```
7.278226 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
8.210732 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
9.294582 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
10.223021 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
11.302829 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
12.232585 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
13.310023 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
14.242616 port1 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 8001.50:00:00:07:00:00.8001
```

8. 那么要如何解决此情况呢？在 FGT 的二层接口下配置 set stpforward enable 即可！

> **重要**
>
> 要特别注意此情况！避免透明模式的 FortiGate 一上线引起 STP 计算异常，引起广播风暴，造成重大业务故障。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port1
FortiGate_Transparent (port1) # set stpforward enable

FortiGate_Transparent (port1) # next
FortiGate_Transparent (interface) # edit port2
FortiGate_Transparent (port2) # set stpforward enable
FortiGate_Transparent (port2) # end
```

9. 再次抓包查看，此时 FGT 可以转发 STP BPDU 报文了。

```
FortiGate_Transparent # diagnose sniffer packet any "stp" 4
interfaces=[any]
filters=[stp]
2.070782 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
2.070797 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
4.085338 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
4.085355 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
6.097429 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
6.097444 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
8.107806 port2 in stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
8.107822 port1 out stp 802.1w, rapid stp, flags [learn, forward], bridge-id 1001.50:00:00:08:00:00.8001
```

.0. 此时再次查看 STP 的状态。

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     5000.0007.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    Shr
Gi0/3               Altn BLK 4         128.4    Shr    //G0/3 被阻断，STP工作正常

SW2#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     5000.0008.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     5000.0008.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
              Aging Time  300 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/3               Desg FWD 4         128.4    Shr
```

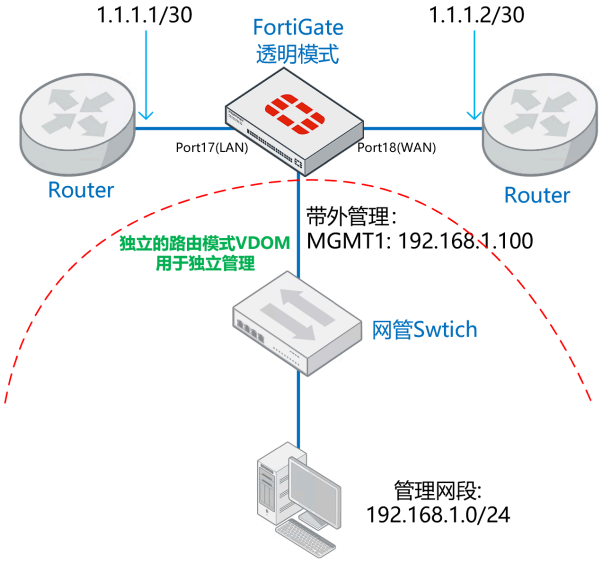
透明模式的带外管理

网络需求

透明模式部署下的防火墙，需实现带外管理：

- 防火墙上须配置一个与管理网段相同的 IP 地址进行管理
- 添加的管理 IP 产生的本地路由，不会对正常业务数据传输造成异步路由等冲突

网络拓扑



配置要点

- 将防火墙配置为透明模式
- 配置 VDOM
- 将 MGMT1 加入到独立管理的 VDOM 中

配置步骤

1. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。配置可以管理防火墙的本地 IP 和网关。

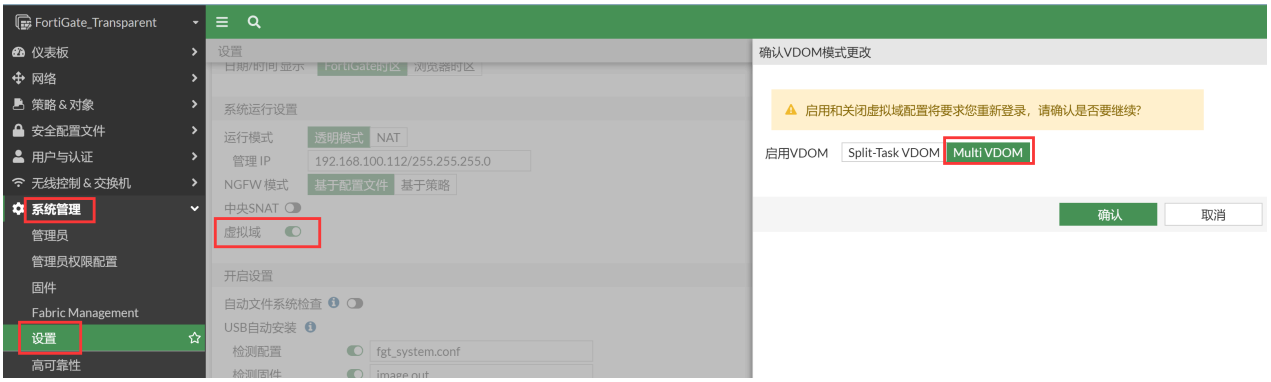
```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 1.1.1.3 255.255.255.0
    set gateway 1.1.1.2
end
```

2. MGMT 口默认有管理权限，要通过 port17（LAN）接口管理设备为例，开启 port17（LAN）管理 FGT 的命令如下。允许网管协议从 Port1 接口通过 https/http/SSH/Ping 访问透明模式的 FortiGate。

```bash
config system interface
    edit port17
    edit allowaccess https http ping ssh
end
```

3. 当前的配置只能通过 1.1.1.3 去管理 FGT，如果需要独立管理 FortiGate 需要开启 VDOM 功能，并且划分独立的网段和路由给到 FGT 进行网管。

4. 在 GUI 或命令行下开启 VDOM 功能，开启后当前管理员会退出登陆。

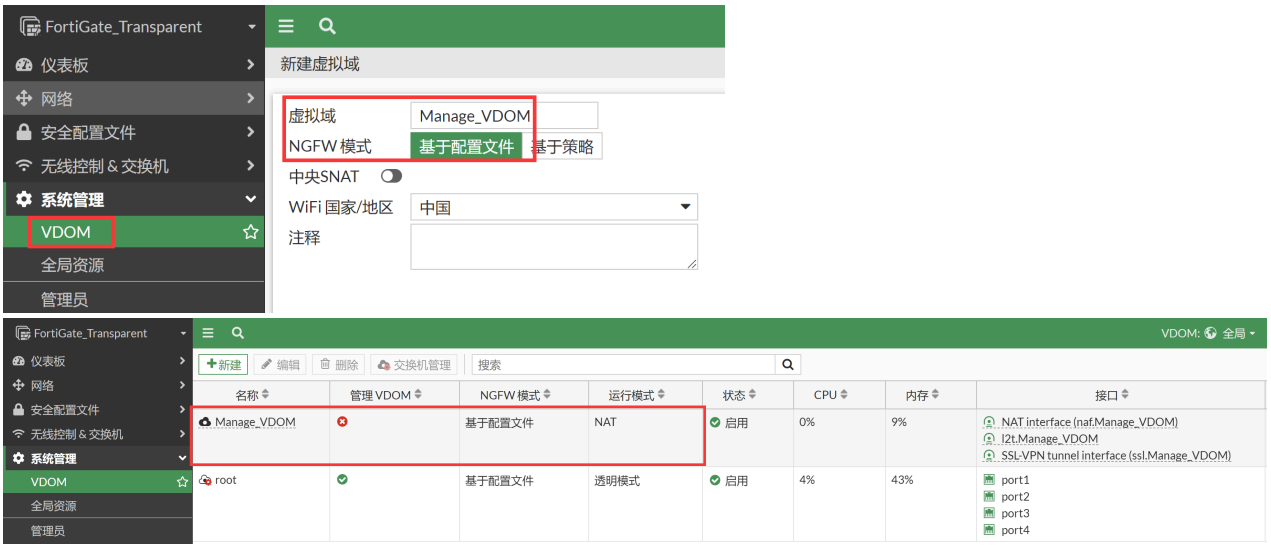



```bash
FortiGate_Transparent # config system global
FortiGate_Transparent (global) # set vdom-mode multi-vdom
FortiGate_Transparent (global) # end
You will be logged out for the operation to take effect.
Do you want to continue? (y/n)y
exit
FortiGate_Transparent login:
```

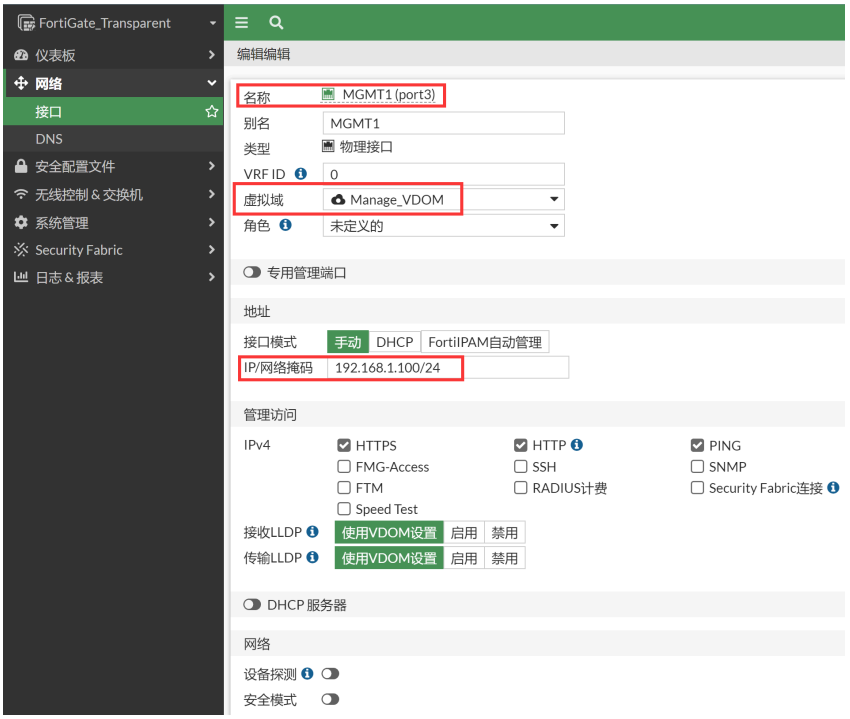
5. 开启 VDOM 之后 WEB 页面系统管理下出现 VDOM 的配置模块。



6. 新建一个 MGMT 的 VDOM，默认的 root 已经是透明模式的状态了。



7. 编辑 MGMT1 接口，将 FortiGate 用于独立管理的接口加入 Manage_VDOM，并配置独立管理 IP。



8. 进入 Mange_VDOM，为 MGMT1 配置默认路由。



9. 注意：如果需要服务更新、syslog、SNMP 网管等等都需要从 MGMT1 发出，需要将管理 VDOM 切换到 Mange_VDOM。

```bash
config global
    config system global
        set management-vdom Mange_VDOM
    end
end
```



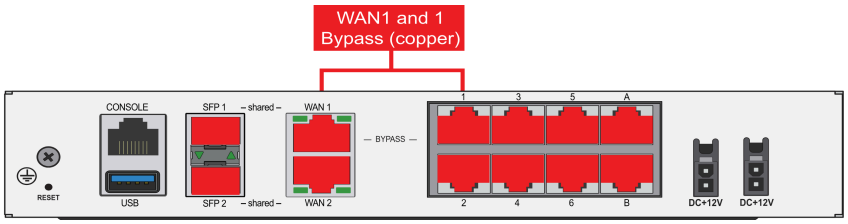
注意事项

1. 当不需要带外管理端口，对处于透明模式的防火墙直接进行管理时（如此例中的 port17 口或 port18 口），需注意：
   - manage IP 是整个防火墙本身的 IP 地址，不针对某个接口，根据配置此处 FGT 本机管理 IP 为 1.1.1.3。
   - 需通过某个接口对桥模式防火墙进行管理时，需开启相应接口下的如 ping，https，ssh 等管理功能，如此例中，当要通过 port17 或者 port18 来对防火墙进行管理时，需开启 port17 和 port18 接口下的 ping、https、ssh 等管理服务。
2. 而独立管理的 VDOM，则是完全由另外一个 VDOM（Mange_VDOM）来充当管理功能的防火墙，而原本的 root 只用来转发业务数据，将管理和业务分离出来。

Bypass 部署

模式说明

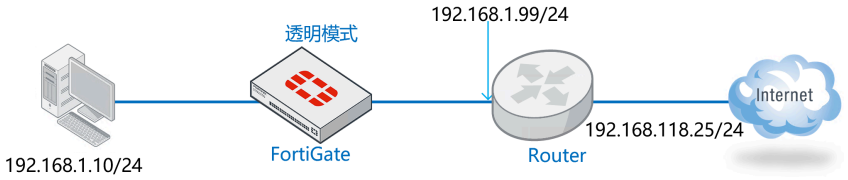
FortiGate 部分型号支持 Bypass 接口，即设备断电后或者重启，依然能够通讯（只能在透明模式下工作）。如下举例设备（80F Bypass），WAN1 和 internal1 接口形成一个 Bypass Pair（接口 internal1 是名为 internal 的硬件交换接口的一部分。要启用 Bypass 模式，必须从 internal 接口中删除 internal1）。



重要

目前支持 Bypass 的 FortiGate 型号包含：FortiGate 2500E、FortiGate 400E Bypass、FortiGate 800D、FortiGate 80F Bypass、FortiGateRugged 60F 3G4G、FortiGateRugged 60F、FortiGateRugged 70F 3G4G、FortiGateRugged 70F 等。具体可参考：
https://docs.fortinet.com/document/fortigate/7.0.99/cli-reference/530620/config-system-bypass ↗

网络拓扑



防火墙工作在透明模式下，开启防病毒功能。使用 bypass 口，当防火墙故障后，确保链路工作正常。

配置要点

- 将防火墙配置为透明模式
- 配置防火墙策略
- 开启 Bypass 模式

配置步骤

1. 将防火墙配置为透明模式，进入设备命令行（CLI）中进行配置，修改 FGT 的运行模式为透明模式（默认为 NAT 路由模式）。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置，配置可以管理防火墙的本地 IP 和网关。

```bash
config system global
    set hostname FortiGate_Transparent
end
config system settings
    set opmode transparent
    set manageip 1.1.1.3 255.255.255.0
    set gateway 1.1.1.2
end
```

2. MGMT 口默认有管理权限，要通过 port17（LAN）接口管理设备为例，开启 port17（LAN）管理 FGT 的命令如下。

```bash
config system interface
    edit port17
    edit allowaccess https http ping ssh
end
```

3. 配置防火墙策略，保证流量可以正常通过 FortiGate。

4. 开 Bypass 模式。

```
config system bypass
    set bypass-watchdog enable
    set poweroff-bypass enable
end
```

结果验证

将系统断电或重启设备，客户业务不中断。

透明模式注意事项与排错

注意事项

1. 默认情况下全新 FortiGate 不转发 BPDU 报文，这样可能会导致 STP 问题从而引起网络出现二层环路，可以登陆到全新 FortiGate 命令行下，进入到接口编辑下面，执行命令启用 BPDU 转发：set stpforward enable。

```
 FGT # config system interface
 FGT (interface) #edit port1
 FGT (port1) #set stpforwad enable        #默认为关闭
 FGT (port1) #next
 然后依次进入各接口启用stpforward功能
```

2. 可以使用 forward domain 来控制指定的接口间数据转发，具有相同的 forward domain 号不同的接口间数据包可转发。

```
FGT # config system interface
FGT (interface) # edit wan1
FGT (wan1) # set forward-domain 10
FGT (wan1) # next
FGT (interface) # edit wan2
FGT (wan2) # set forward-domain 10
FGT (wan2) # end
```

> **重要**
>
> forward-domain 无需提前定义，直接配置生效；forward-domain 如同转发域的概念，广播包只会在同个转发域进行广播。

3. 只有 Ethernet II 类型帧可以被转发，在默认的配置下，所有的其他类型的二层协议都不能被转发。需要转发这些帧请在接口下面启用 l2forward 功能。

```
FGT # config system interface
FGT (interface) #edit port1
FGT (port1) #set l2forward enable       #默认为关闭
FGT (port1) #next
然后依次登陆各接口启用l2forward功能
```

4. 多播包默认情况下是转发的，所以如果部署透明模式的防火墙接入到多播环境中如果不通，要注意下面的两个参数值，默认是转发的状态。

```
FGT # config system settings
FGT (settings)set multicast-skip-policy disable   #默认为关闭
FGT (settings)end

另外多播策略存在一条默认的any到any的全通策略：
config firewall multicast-policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

5. 缺省情况下广播是不受策略控制的，除了 ARP 广播是允许的，其他的广播流量全部是被禁止转发的，看默认配置如下，如需开启广播转发需要在接口开启 broadcast-forward。

```
config system interface
    edit "port1"
        set allowaccess ping https ssh http fgfm
        set broadcast-forward enable   //默认为关闭
    next
end
然后依次进入各接口启用broadcast-forward功能
```

6. 如果明确需要带外管理功能的话，请设置多 VDOM，其中 VDOM root 仅用来管理其他相关的透明 VDOM 使用。

```
FortiGate_Transparent # config system global
FortiGate_Transparent (global) # set vdom-mode
no-vdom       Disable split/multiple VDOMs mode.
split-vdom    Enable split VDOMs mode.
multi-vdom    Enable multiple VDOMs mode.
FortiGate_Transparent (global) # set vdom-mode multi-vdom    //开启多VDOM模式
FortiGate_Transparent (global) # end
You will be logged out for the operation to take effect.
Do you want to continue? (y/n)y
exit
FortiGate_Transparent login: admin
Password:
Welcome !
```

7. 如果部署透明模式的全新 FortiGate 到路由器和主机直连的时候，请注意在这段线路中是否会出现相同的源目的 IP 地址数据流在防火墙不同方向上时的 MAC 地址不一致的情况，简单的应用如 VRRP 之类的主机路由备份协议，这种流量的数据流源目 MAC，正向和反向是不一样的，这种情况下请尽量在防火墙上面设置静态 IP/MAC 以告诉全新 FortiGate 指定的虚拟 MAC 地址属于某个 VRRP 组。

> **重要**
>
> 一个 forward_domain 里面只能指定一个相同 MAC 地址对。

8. 透明模式下查查看 MAC 表，使用 diag netlink brctl name host VDOM_name.b 查看，以下以 root 的 vdom 为例：

```
FGT# diag netlink brctl name host root.b
show bridge control interface root.b host.
fdb: size=256, used=6, num=7, depth=2, simple=no
Bridge root.b host table
port no device devname mac addr ttl attributes
2 7 wan2 02:09:0f:78:69:00 0 Local Static
5 6 trunk_1 02:09:0f:78:69:01 0 Local Static
3 8 dmz 02:09:0f:78:69:01 0 Local Static
4 9 internal 02:09:0f:78:69:02 0 Local Static
3 8 dmz 00:80:c8:39:87:5a 194
4 9 internal 02:09:0f:78:67:68 8
1 3 wan1 00:09:0f:78:69:fe 0 Local Static
```

9. 透明模式的功能限制：
   - 只支持策略模式的 IPSec，支持用户认证。
   - 不支持接口模式 IPSec，不支持 SSL VPN，不支持动态路由和策略路由、不支持 DHCP。

总结

透明模式的 FortiGate 其实默认只转发 ARP 广播和组播报文（默认存在一条全通的组播策略，如果删除了这条默认的组播全通策略，仍然会放通 IPv6 ND 的组播流量，而丢弃其他的组播流量），其他的流量通通会被丢弃。如果你想让防火墙更像一个 SW 一样的存在，那么在使用的二层接口下应该这样去修改。

```
config system interface
    edit "port1"
        set broadcast-forward enable    //开启广播转发
        set l2forward enable            //开启非IP流量转发
        set stpforward enable           //开启STP转发
    next
    edit "port2"
        set broadcast-forward enable
        set l2forward enable
        set stpforward enable
    next
end
```

这样 FortiGate 对接入的网络环境影响应该来说是更小的，对于那种不是很清楚网络中有一些什么流量的环境，建议按照上述建议进行配置优化，让 FortiGate 更加接近于一个 SW 的存在。

> **重要**
>
> 接口下的 vlanforward 保持为默认的 disable，不要去动，不要去动，可能会引起环路！

最佳实践

1. 不要将透明模式的两个物理接口连接到同一个 VLAN 或同一个 SW/HUB (相同广播域)，SW 一旦检测到相同的 MAC 地址出现在不同的接口，大部分的二层交换机会出现 MAC 地址表翻转，会让网络非常的不稳定或环路。

2. 如果网络中存在 VLAN，需要在 FGT 上创建了多个 VLAN，要确保给每一个 VLAN - ID 分配一个独立的 Forwarding Domain ID 确保广播在 VLAN 之间被隔离（透明模式下 VLAN 之间不隔离广播，Fortiward-Domain 才隔离广播）。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit Inside_VLAN10
FortiGate_Transparent (Inside_VLAN10) # set forward-domain 10
FortiGate_Transparent (Inside_VLAN10) # next
```

3. 在所有接口上开启 stpforward enable（开启二层转发的一些特性，防止误丢一些二层数据帧)。

```
config system interface
    edit "portX"
        set broadcast-forward enable
        set l2forward enable
        set stpforward enable
    next
end
```

4. 使用一个独立的透明模式的 VDOM 用来保护二层的业务流量，然后再使用一个独立管理 VDOM（NAT 路由模式）用于设备的管理和 fortiguard 升级等，利用 VDOM 功能将管理（NAT 路由模式）和业务（透明模式）分离。

5. 仅将用于生产的业务接口置于透明模式 VDOM 中。将所有其他接口置于 NAT 路由模式 VDOM 中，这可以防止潜在的第 2 层环路。

故障排错

1. 先检查 FortiGate 的二层 MAC 地址表。注意，在透明模式下，FortiGate 是根据二层转发数据库进行转发二层数据，该数据库只能在 Global 通过命令” diag netlink brctl name host root.b”进行导出，ARP 表只能在有 IP 访问 FortiGate 的情况下出现。外部设备发送的 ARP/GARP 会映射到二层转发表，并且当 MAC 地址变更的时候会在会话同步更改目标 MAC 地址。下面的命令能导出每一个 VDOM 桥实例的二层转发表，在转发表里可以单独看到 VLAN/VDOM 里的 MAC 地址。

```
diag netlink brctl list
diag netlink brctl name host <VDOM_name>.b

trans VDOM查看二层转发表的例子:
diag netlink brctl name host trans.b

输出如下:
diagnose netlink brctl name host trans.b
show bridge control interface trans.b host.
fdb: size=2048, used=4, num=8, depth=3
Bridge trans.b host table
port no device  devname mac addr                ttl     attributes
  7     21      port3vlan200    00:50:79:66:68:01       0        Static
  1     3        port1   52:c9:fd:91:54:00       0       Local Static
  6     20      port2vlan200    52:c9:fd:91:54:01       0       Local Static
  3     17      port2vlan100    52:c9:fd:91:54:01       0       Local Static
  2     4        port2   52:c9:fd:91:54:01       0       Local Static
  8     5        port3   52:c9:fd:91:54:02       0       Local Static
  7     21      port3vlan200    52:c9:fd:91:54:02       0       Local Static
  5     19      port3vlan100    52:c9:fd:91:54:02       0       Local Static
```

2. 使用 sniffer 验证数据包是否到达防火墙，通过以下命令使用 sniffer 抓取相应信息。同时通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题。这个能看到在 FortiGate 上面流入以及流出的数据包。

```
进行抓取目的地址是114.114.119.119的数据:
diagnose sniffer packet any "host 114.114.119.119" 4 0 l    //按CTRL+C进行停止

基于端口进行数所包抓取
diagnose sniffer packet portA  "host 114.114.119.119" 4 0  l
diagnose sniffer packet portB  "host 114.114.119.119" 4 0  l
```

```
如有需要，可以通过每个vlan进行获取更明细内容
diagnose sniffer packet <each_vlan_interface> "" 6 0 l

参数4和6的区别（用于显示输出抓取数据参数）：
1: print header of packets (只有IP头部)
2: print header and data from ip of packets (有ip头部和数据内容)
3: print header and data from ethernet of packets  (if available)
4: print header of packets with interface name (只有IP头部和端口)
5: print header and data from ip of packets with interface name (有ip头部和数据内容以及端口)
6: print header and data from ethernet of packets (if available) with intf name

参数0的用处：输出个数

参数l的用处(用于输出时候显示的时间格式)：
a：absolute UTC time, yyyy-mm-dd hh:mm:ss.ms
l：absolute LOCAL time, yyyy-mm-dd hh:mm:ss.ms
```

3. 使用 debug flow 验证数据包是否穿越防火墙（vlanforward 无效）。在每一个 VDOM 使用 "debug flow" 命令进行跟踪流量，通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题，这种排障的方法能够查看到流量是否因为某条策略导致数据包不通或者通过设备。

```
以源地址为抓取条件
═══════════════════════════════════════════════════════════
diag debug flow filter add <IP_address_of_source_device>
diag debug flow show function-name enable
diag debug flow show console enable
diag debug flow show iprope enable
diag debug flow trace start 100
diag debug enable
═══════════════════════════════════════════════════════════
如需要停止，可以敲"diag debug flow trace stop"

以目标地址为抓取条件
═══════════════════════════════════════════════════════════
diag debug flow filter add <IP_address_of_destination_device>
diag debug flow show function-name enable
diag debug flow show console enable
diag debug flow show iprope enable
diag debug flow trace start 100
diag debug enable
═══════════════════════════════════════════════════════════
如需要停止，可以敲"diag debug flow trace stop"

当有触发排障的流，输出如下：
id=20085 trace_id=113 msg="vd-tp_mode received a packet(proto=6,10.160.0.160:4370→10.160.0.152:23) from internal."
id=20085 trace_id=113 msg="Find an existing session, id-00000a40, original direction"
id=20085 trace_id=113 msg="enter fast path"
id=20085 trace_id=113 msg="send out via dev-dmz1, dst-mac-00:01:02:03:04:05"
```

4. 查看数据包在防火墙上创建的 session。在每一个 VDOM 使用 "diag sys session" 命令进行防火墙会话检查，通过使用 ping 在 FortiGate 两端进行测试可以初步分析问题，这种排障的方法能够查看防火墙是否会为流量创建会话，以及是否硬件加速以及命中哪一条策略。

```
以源地址为抓取条件（单播）
═══════════════════════════════════════════════════════════
diag sys session filter src <IP_address_of_source_device>
diag sys session list
═══════════════════════════════════════════════════════════
当有触发排障的会话,输出如下：
session info: proto=1 proto_state=00 duration=2 expire=58 timeout=0 flags=00000000 sockflag=00000000 sockport=0 av_idx=0
use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=500/5/1 reply=500/5/1 tuples=2
tx speed(Bps/kbps): 248/1 rx speed(Bps/kbps): 248/1
orgin→sink: org pre→post, reply pre→post dev=9→10/10→9 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 192.168.1.1:64→192.168.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.2.1:64→192.168.1.1:0(0.0.0.0:0)
misc=0 policy_id=4 auth_info=0 chk_client_info=0 vd=1
```

```
serial=00000512 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000
vlanid=100 dd_type=0 dd_mode=0
total session 1

以源地址为抓取条件（多播）
═══════════════════════════════════════════════════════════
diagnose sys mcast-session list
═══════════════════════════════════════════════════════════
当有触发排障的会话,输出如下:
session info: id=123 vf=1 proto=112 192.168.1.2.0→224.0.0.18.0
used=2 path=1 duration=1353 expire=179 indev=10 pkts=1360 bytes=54400
state=00000002:tp
path: policy=0, outdev=9

session info: id=124 vf=1 proto=112 192.168.2.99.0→224.0.0.18.0
used=2 path=1 duration=1353 expire=179 indev=10 pkts=1360 bytes=54400
state=00000002:tp
path: policy=0, outdev=9
Total 2 sessions
```

参考文档

https://community.fortinet.com/t5/FortiGate/Technical-Tip-Troubleshoot-and-verify-if-traffic-is-hitting-a/ta-p/197261 ↗

https://community.fortinet.com/t5/FortiGate/Troubleshooting-Tip-First-steps-to-troubleshoot-connectivity/ta-p/192560 ↗

https://community.fortinet.com/t5/FortiGate/Troubleshooting-Tip-FortiGate-session-table-information/ta-p/196988 ↗

https://community.fortinet.com/t5/FortiGate/Technical-Tip-Configuring-a-FortiGate-in-Transparent-mode-to/ta-p/197611 ↗

https://community.fortinet.com/t5/FortiGate/Technical-Note-FortiGate-and-Gratuitous-ARP-GARP/ta-p/192962 ↗

VLAN 与透明模式

- Forward-Domain
- vlanforward
- Remapping VLAN ID (VLAN NAT，特殊场景使用)

Forward-Domain

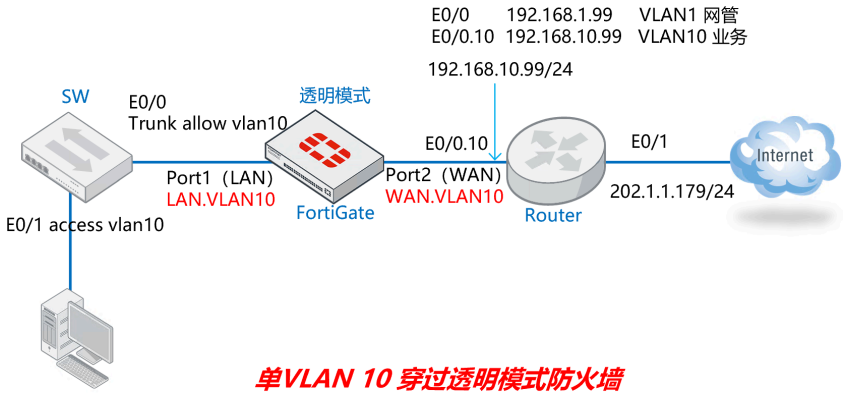
> **重要**
>
> 有 VLAN 的场景建议配置。

网络需求

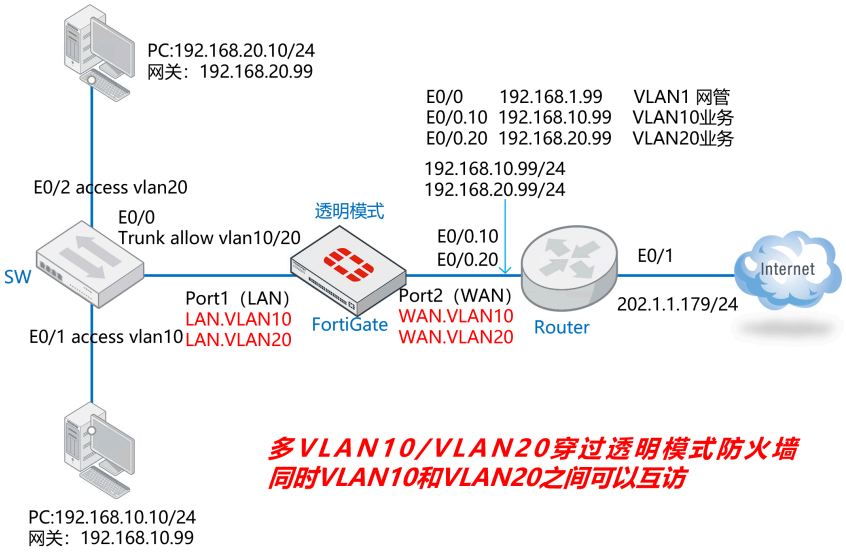
内部网络有两个 vlan（trunk 环境），网关在路由器上，防火墙工作在透明模式，部署在核心交换与核心路由之间，允许两个 vlan 访问外网，同时两个 vlan 之间需要互访，并且对这两个 vlan 进行保护，开启反病毒过滤。

网络拓扑

拓扑 1：单 VLAN 透传



拓扑 2：多 VLAN 透传，VLAN 之间互访



配置要点

拓扑 1：单 VLAN 透传

- SW 交换机/Internet_R 路由器的基础配置
- 将防火墙配置为透明模式并开启网管
- 配置 LAN 和 WAN 接口的 VLAN 子接口并加入到 forward-domain
- 配置安全策略允许内网 PC（VLAN10 内主机）访问互联网

拓扑 2：多 VLAN 透传，以及 VLAN 之间互访

- SW 交换机/Internet_R 路由器的基础配置
- 新增 Inside 和 Outside 关于 VLAN20 的子接口并加入到 Forward-domain
- 配置 Inside_VLAN20 到 Outside_VLAN20 的安全策略允许内网 PC（VLAN20 内主机）访问互联网
- 配置 VLAN10 与 VLAN20 之间互访的策略

配置步骤

拓扑 1：单 VLAN 透传

1. SW 交换机/Internet_R 路由器的基础配置（以思科为例）。

   SW:

```bash
SW:
interface Ethernet0/0
switchport trunk allowed vlan 1,10
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface Ethernet0/1
switchport access vlan 10
switchport mode access
!
```

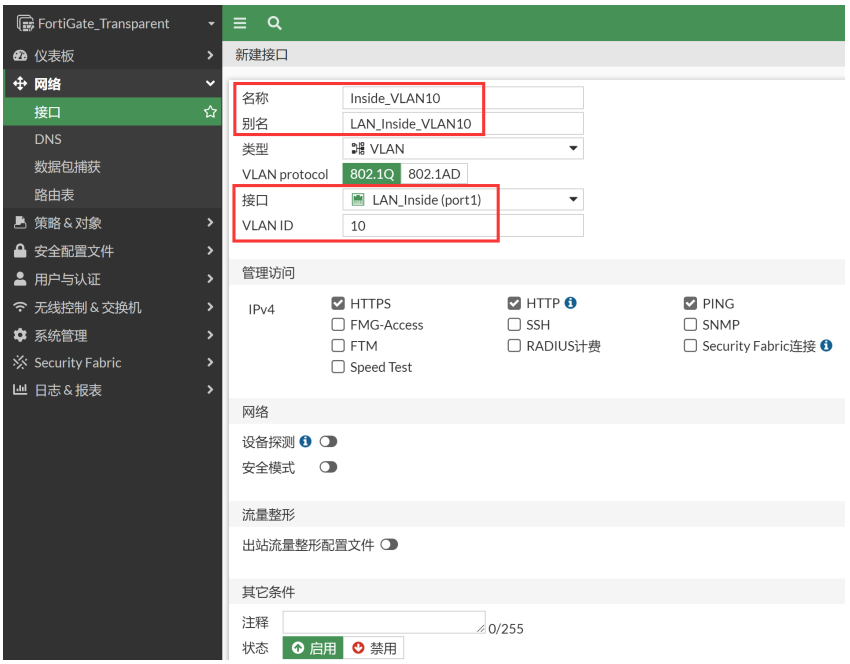
   Router:

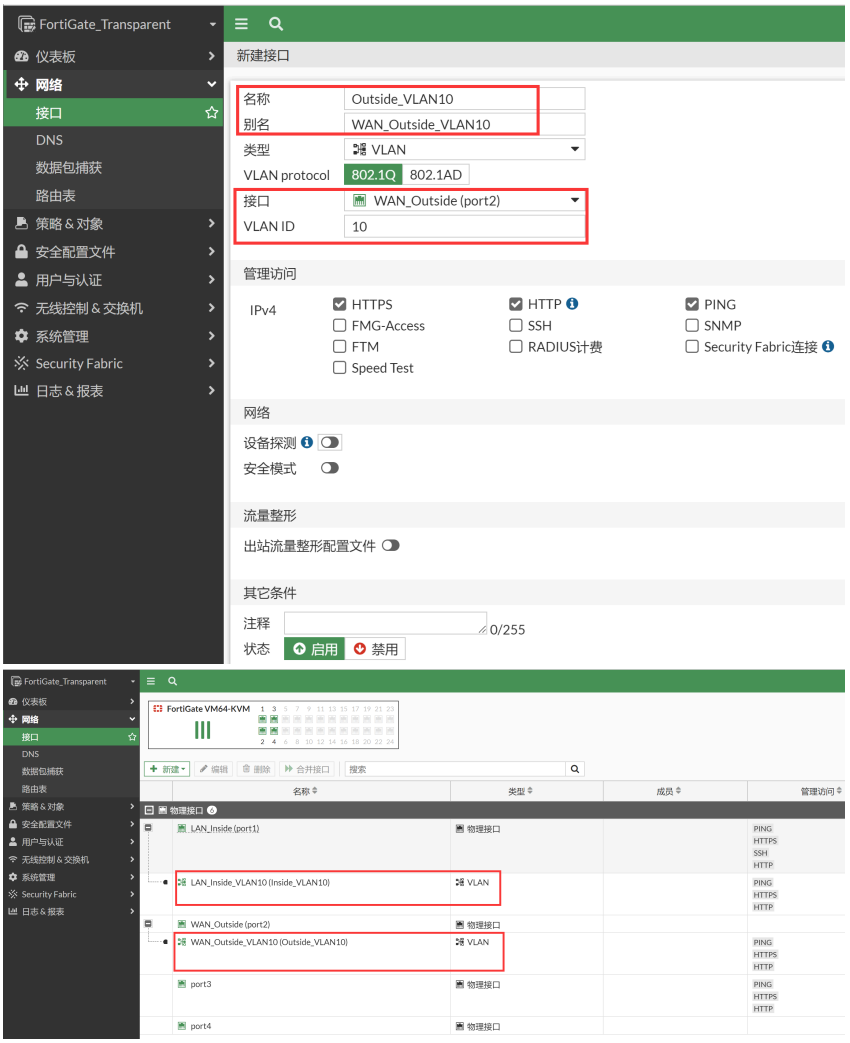
```bash
Router:
hostname Internet_R
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

2. 将防火墙配置为透明模式并开启网管（参考网络管理 → 透明模式 → 传统透明模式 → 开启透明模式防火墙并保护上网流量 ↗）。

3. 新建 LAN 和 WAN 接口的 VLAN10 子接口并加入到 forward-domain。


FortiGate_Transparent
仪表板
网络
接口
DNS
数据包捕获
路由表
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FortiGate VM64-KVM
新建
编辑
删除
合并接口
搜索
接口
所属区域
虚拟线对
名称
类型
成员
port4
物理接口
LAN_Inside (port1)
物理接口
PING
HTTPS
SSH
HTTP
WAN_Outside (port2)
物理接口
port3
物理接口
PING
HTTPS
HTTP


FortiGate_Transparent
仪表板
网络
接口
DNS
数据包捕获
路由表
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新建接口
名称
Inside_VLAN10
别名
LAN_Inside_VLAN10
类型
VLAN
VLAN protocol
802.1Q
802.1AD
接口
LAN_Inside (port1)
VLAN ID
10
管理访问
IPv4
HTTPS
HTTP
PING
FMG-Access
SSH
SNMP
FTM
RADIUS计费
Security Fabric连接
Speed Test
网络
设备探测
安全模式
流量整形
出站流量整形配置文件
其它条件
注释
0/255
状态
启用
禁用




```bash
config system interface
    edit "Inside_VLAN10"
        set vdom "root"
        set alias "LAN_Inside_VLAN10"
        set interface "port1"
        set vlanid 10
    next
    edit "Outside_VLAN10"
        set vdom "root"
        set alias "WAN_Outside_VLAN10"
        set interface "port2"
        set vlanid 10
    next
end
```

重要

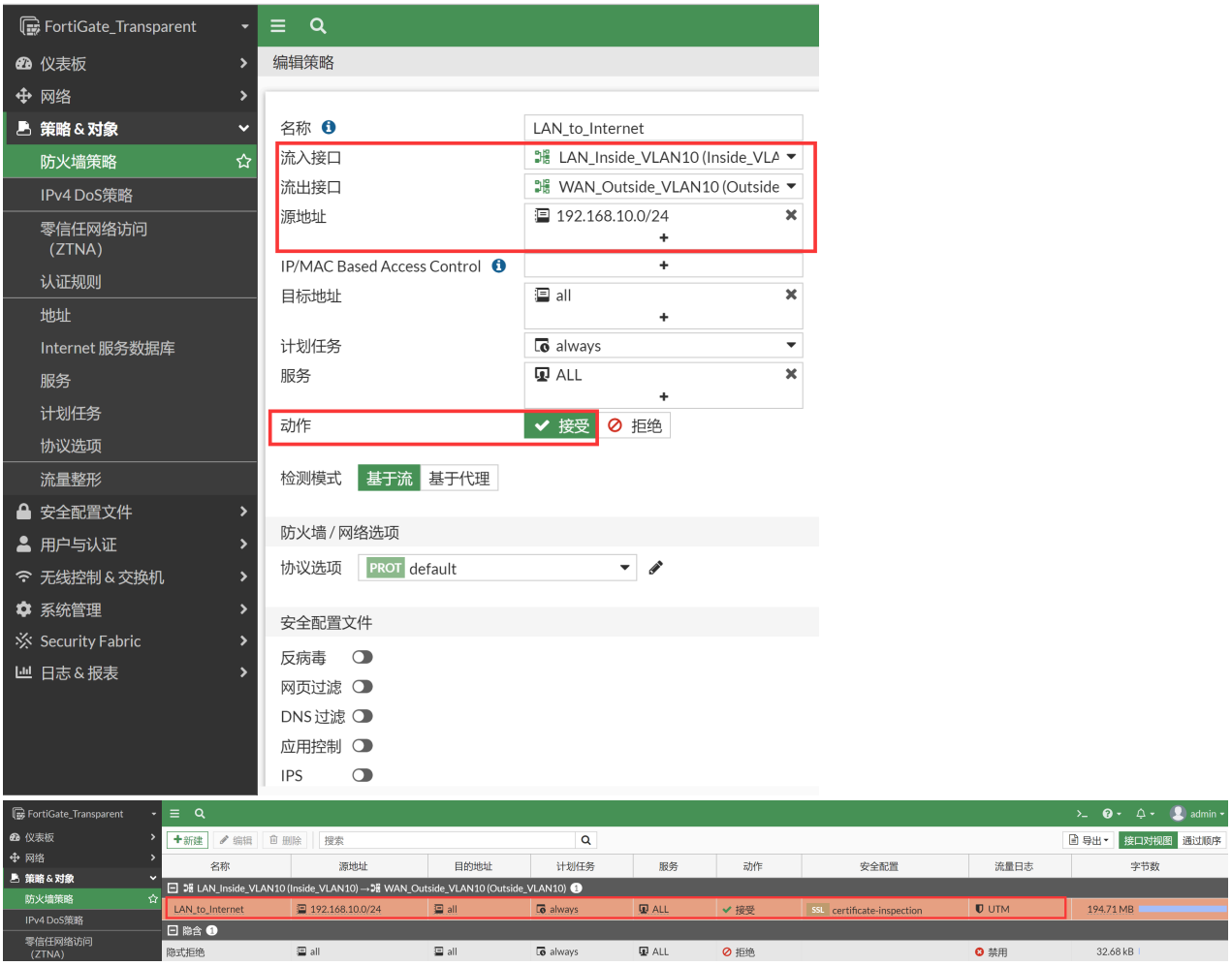
建议在不需要的情况下关闭接口配置下的“设备探测”（Device detection）功能，该功能用于 MAC 地址厂商设备信息识别及 MAC 地址过滤，会消耗较多的设备资源。

4. 为 Inside_VLAN10 和 Outside_VLAN10 接口配置 forward-domain。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit Inside_VLAN10
FortiGate_Transparent (Inside_VLAN10) # set forward-domain 10
FortiGate_Transparent (Inside_VLAN10) # next

FortiGate_Transparent (interface) # edit Outside_VLAN10
FortiGate_Transparent (Outside_VLAN10) # set forward-domain 10
FortiGate_Transparent (Outside_VLAN10) # end
```

> **重要**
>
> Forward-Domain 说明：
>
> 默认所有的接口都属于 Forward-Domain 0，可以理解为同一个广播域，在 FortiGate 的概念中，Forward-Domain 才代表真正的广播域，并非 VLAN，而一般的理解交换机的常识都是 VLAN 隔离广播域，因此如果需要使用 VLAN 则务必将 VLAN - ID 和 Fortiward-Domain ID 关联起来，这样就可以以正常的交换机 VLAN 隔离广播域的思维去考虑 FortiGate 的透明模式。Forward-Domain 是一个 FGT 透明模式下比较特别的东西，记住：只要有 VLAN - ID 的场景就需要配置对应的 Forward-Domain ID 这样就不会出现什么问题。

5. 配置防火墙策略允许内网 PC（VLAN10 内主机）访问互联网，注意这里要选择 VLAN 接口。



拓扑 2：多 VLAN 透传，VLAN 之间互访

1. SW 交换机/Internet_R 路由器的基础配置（以思科为例）。

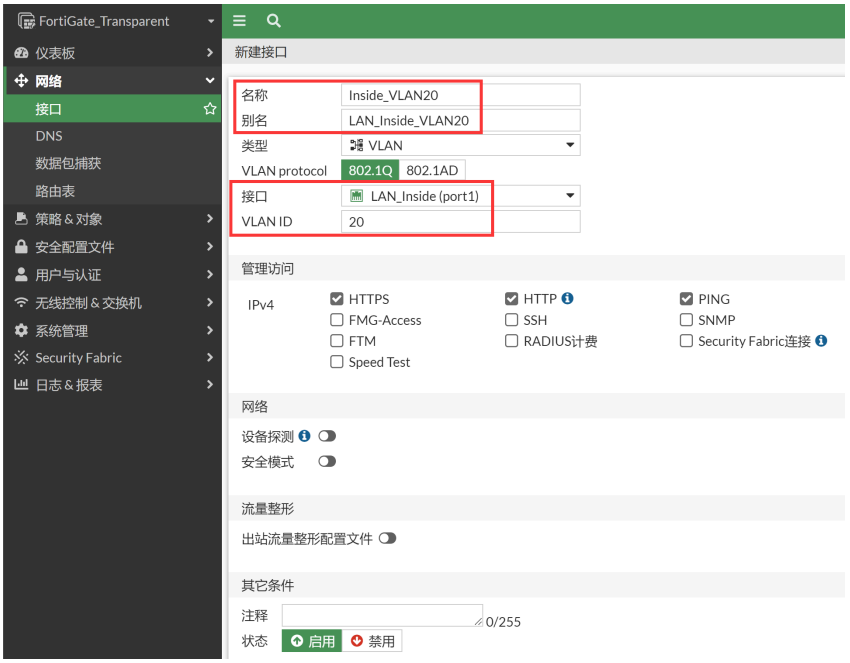
SW:

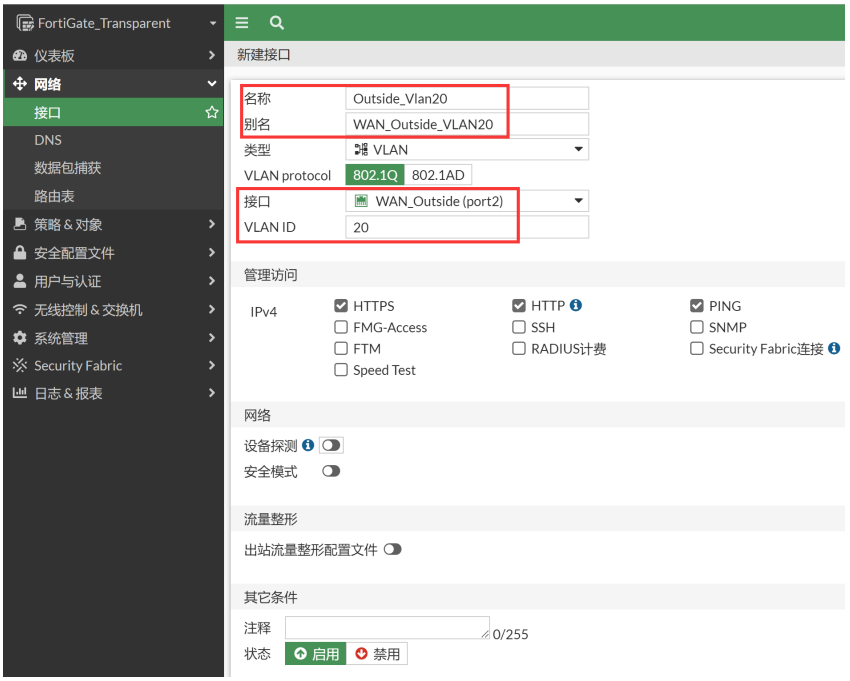
```bash
interface Ethernet0/0
switchport trunk allowed vlan 1,10,20
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface Ethernet0/1
switchport access vlan 10
switchport mode access
!
interface Ethernet0/2
switchport access vlan 20
switchport mode access
!
```

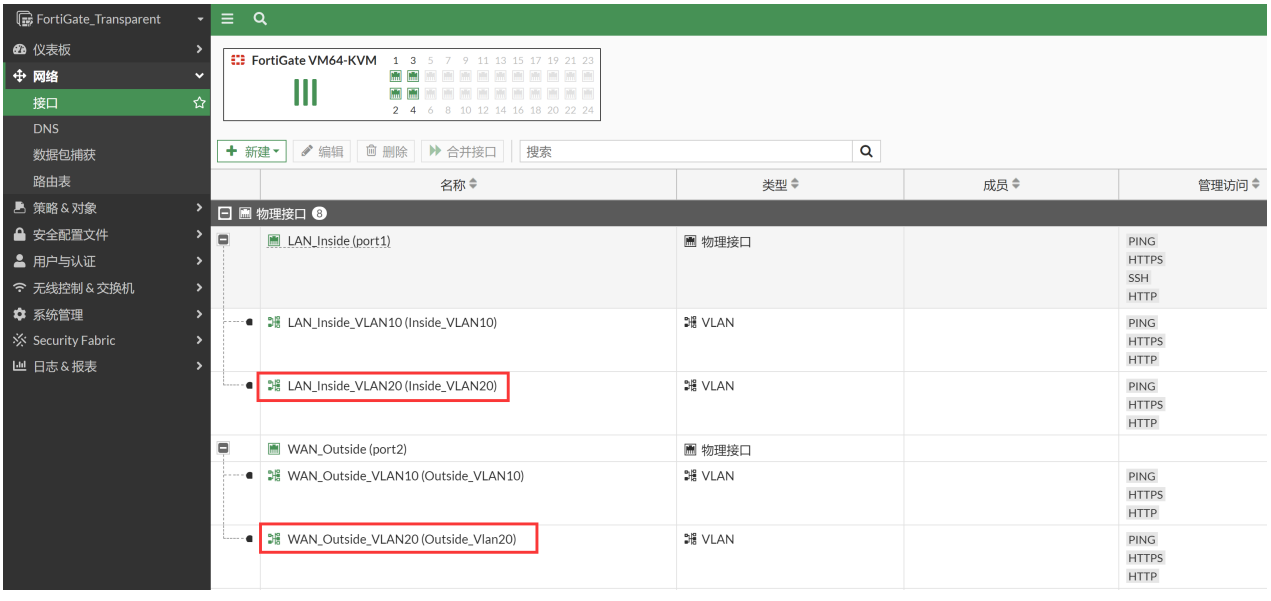
Router:

```bash
hostname Internet_R
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.99 255.255.255.0
no shutdown
ip nat inside
!
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

2. 新增 Inside 和 Outside 关于 VLAN20 的子接口并加入到 Forward-domain。


FortiGate_Transparent
仪表板
网络
接口
DNS
数据包捕获
路由表
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新建接口
名称 Inside_VLAN20
别名 LAN_Inside_VLAN20
类型 VLAN
VLAN protocol 802.1Q 802.1AD
接口 LAN_Inside (port1)
VLAN ID 20
管理访问
IPv4
HTTPS
HTTP
PING
FMG-Access
SSH
SNMP
FTM
RADIUS计费
Security Fabric连接
Speed Test
网络
设备探测
安全模式
流量整形
出站流量整形配置文件
其它条件
注释
0/255
状态 启用 禁用


FortiGate_Transparent
仪表板
网络
接口
DNS
数据包捕获
路由表
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新建接口
名称 Outside_Vlan20
别名 WAN_Outside_VLAN20
类型 VLAN
VLAN protocol 802.1Q 802.1AD
接口 WAN_Outside (port2)
VLAN ID 20
管理访问
IPv4
HTTPS
HTTP
PING
FMG-Access
SSH
SNMP
FTM
RADIUS计费
Security Fabric连接
Speed Test
网络
设备探测
安全模式
流量整形
出站流量整形配置文件
其它条件
注释
0/255
状态 启用 禁用

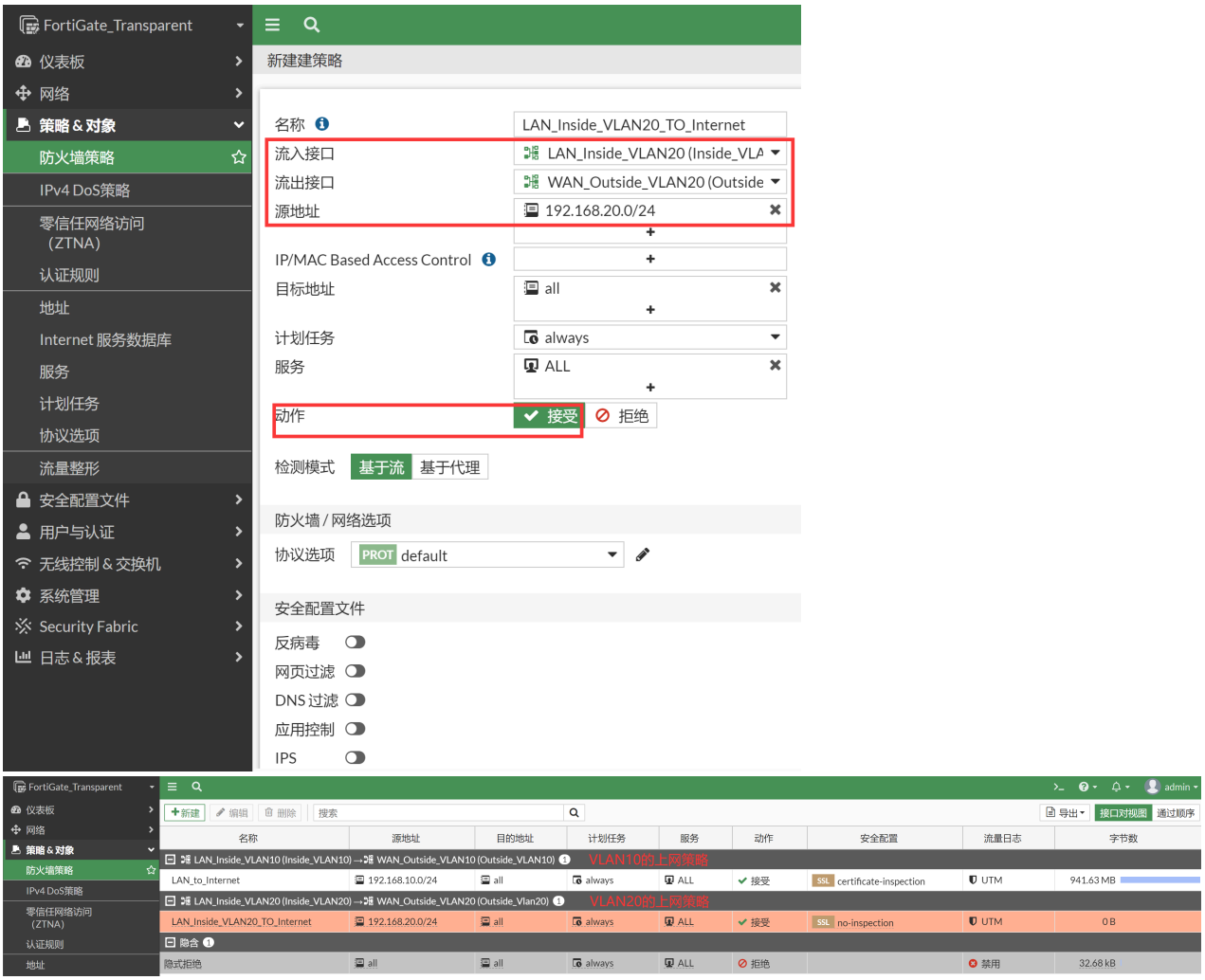

FortiGate_Transparent
仪表板
网络
接口
DNS
数据包捕获
路由表
策略 & 对象
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FortiGate VM64-KVM
新建
编辑
删除
合并接口
搜索
名称
类型
成员
管理访问
物理接口 8
LAN_Inside (port1)
物理接口
PING HTTPS SSH HTTP
LAN_Inside_VLAN10 (Inside_VLAN10)
VLAN
PING HTTPS HTTP
LAN_Inside_VLAN20 (Inside_VLAN20)
VLAN
PING HTTPS HTTP
WAN_Outside (port2)
物理接口
WAN_Outside_VLAN10 (Outside_VLAN10)
VLAN
PING HTTPS HTTP
WAN_Outside_VLAN20 (Outside_Vlan20)
VLAN
PING HTTPS HTTP

```
config system interface
    edit "Inside_VLAN20"
        set vdom "root"
        set alias "LAN_Inside_VLAN20"
        set interface "port1"
        set vlanid 20
    next
    edit "Outside_VLAN20"
        set vdom "root"
        set alias "WAN_Outside_VLAN20"
        set interface "port2"
        set vlanid 20
    next
end
```

3. 为 Inside_VLAN20 和 Outside_VLAN20 接口配置 forward-domain。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit Inside_VLAN20
FortiGate_Transparent (Inside_VLAN20) # set forward-domain 20
FortiGate_Transparent (Inside_VLAN20) # next

FortiGate_Transparent (interface) # edit Outside_Vlan20
FortiGate_Transparent (Outside_VLAN20) # set forward-domain 20
FortiGate_Transparent (Outside_VLAN20) # end
```

4. 配置 Inside_VLAN20 到 Outside_VLAN20 的上网安全策略，允许 VLAN20 内的主机访问 Internet。



5. 此时 VLAN 10 下的 PC 已经可以和 VLAN 20 下的 PC 互访，不需要配置其他策略。

重要

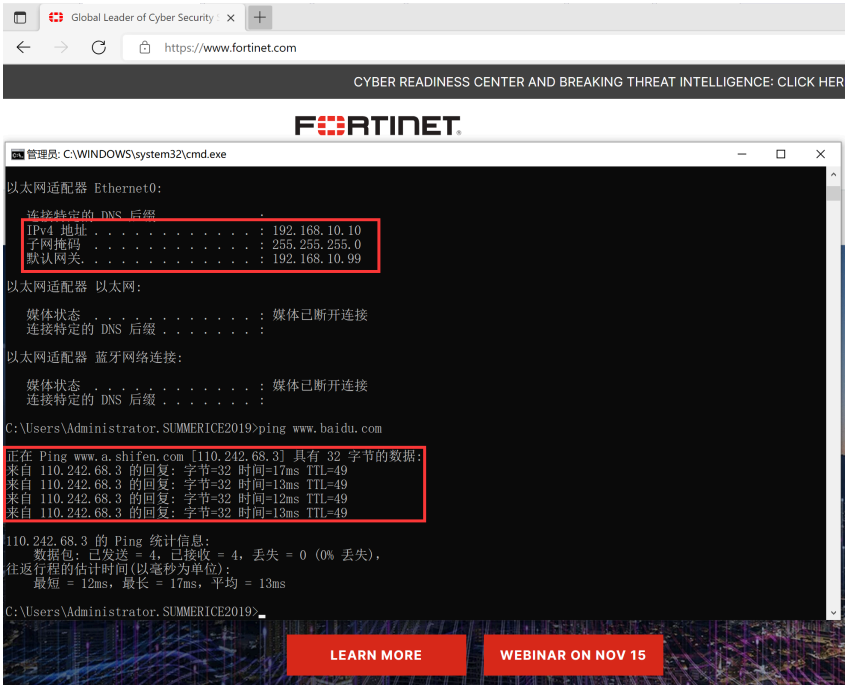
不同 Forward-Domain 下的 VLAN 接口是不能配置在同一条安全策略的源目接口中的。在 VLAN 10 下的 PC 访问 VLAN 20 下的 PC（192.168.10.10 → 192.168.20.10）时，流量会经过 2 次 FortiGate：

- 流量先从 Inside_VLAN10 送到 Outside_VLAN10（匹配 VLAN 10 的上网策略）。此时 FortiGate 创建 192.168.10.10 → 192.168.20.10（Inside_VLAN10 → Outside_VLAN10）的会话。
- 同时，FortiGate 会自动创建 192.168.10.10 → 192.168.20.10（Outside_VLAN20 → Inside_VLAN20）的辅助会话（上述创建会话的 Reflect 会话），流量第二次经过 FortiGate 时，从 Outside_VLAN20 进入并送到 Inside_VLAN20，匹配辅助会话送到 VLAN 20 下的 PC。

结果验证

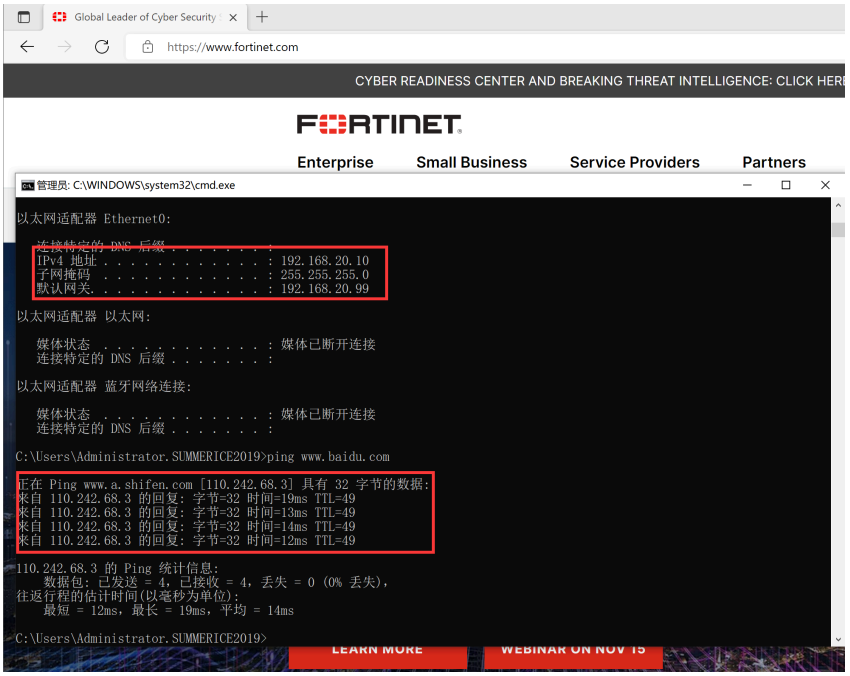
拓扑 1：单 VLAN10 透传

1. VLAN10 内的 PC 通过透明模式的 FortiGate 访问互联网。

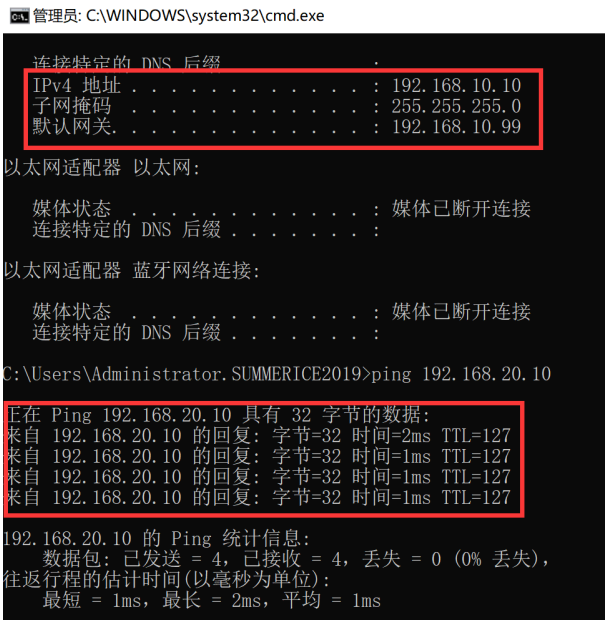


拓扑 2：多 VLAN 透传，VLAN 之间互访

1. VLAN20 内的 PC 访问互联网。



2. VLAN10 内的 PC 访问 VLAN20 的 PC。



3. 查看对应 sniffer 抓包，VLAN 10 主机访问 VLAN 20 主机的 Ping Request 从 Inside_VLAN10 进，从 Outside_VLAN10 出，随后从 Outside_Vlan20 进，从 Inside_VLAN20 出。

```bash
FortiGate_Transparent # diagnose sniffer packet any "host 192.168.20.10 and icmp" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.20.10 and icmp]
3.530807 Inside_VLAN10 in 192.168.10.10 → 192.168.20.10: icmp: echo request
3.530823 Outside_VLAN10 out 192.168.10.10 → 192.168.20.10: icmp: echo request
3.530826 port2 out 192.168.10.10 → 192.168.20.10: icmp: echo request
3.531120 Outside_Vlan20 in 192.168.10.10 → 192.168.20.10: icmp: echo request
3.531129 Inside_VLAN20 out 192.168.10.10 → 192.168.20.10: icmp: echo request
3.531130 port1 out 192.168.10.10 → 192.168.20.10: icmp: echo request

3.531696 Inside_VLAN20 in 192.168.20.10 → 192.168.10.10: icmp: echo reply
3.531703 Outside_Vlan20 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
3.531703 port2 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
3.531856 Outside_VLAN10 in 192.168.20.10 → 192.168.10.10: icmp: echo reply
3.531859 Inside_VLAN10 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
3.531859 port1 out 192.168.20.10 → 192.168.10.10: icmp: echo reply
```

4. 查看 Request 方向与 Reply 方向对应的 debug flow。

   - Ping Request 第 1 次从 Inside_VLAN10 送到 Outside_VLAN10（匹配 VLAN 10 的上网策略）。此时 FortiGate 创建 192.168.10.10 → 192.168.20.10（Inside_VLAN10 → Outside_VLAN10）的会话。

```bash
Request方向第1次过FortiGate:
id=65308 trace_id=10 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=1, 192.168.10.10:2->192.168.20.10:2048) tun_id=0.0.0.0 from Inside_VLAN10. type=8, code=0, id=2, seq=2879."
id=65308 trace_id=10 func=init_ip_session_common line=6043 msg="allocate a new session-00000386, tun_id=0.0.0.0"
id=65308 trace_id=10 func=br_fw_forward_handler line=578 msg="Allowed by Policy-1:"
id=65308 trace_id=10 func=__if_queue_push_xmit line=392 msg="send out via dev-Outside_VLAN10, dst-mac-aa:bb:cc:00:30:00"
```

   - 同时，FortiGate 会自动创建 192.168.10.10 → 192.168.20.10（Outside_VLAN20 → Inside_VLAN20）的辅助会话（上述创建会话的 Reflect 会话），流量第二次经过 FortiGate 时，从 Outside_VLAN20 进入并送到 Inside_VLAN20，匹配辅助会话送到 VLAN 20 下的 PC。

```bash
Request方向第2次过FortiGate:
id=65308 trace_id=11 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=1, 192.168.10.10:2->192.168.20.10:2048) tun_id=0.0.0.0 from Outside_VLAN20. type=8, code=0, id=2, seq=2879."
id=65308 trace_id=11 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00000386, original direction"
id=65308 trace_id=11 func=br_fw_forward_dirty_handler line=282 msg="auxiliary ses proto=1 dev=19→18 192.168.10.10/2⇒192.168.20.10/8"
id=65308 trace_id=11 func=npu_handle_session44 line=1322 msg="Trying to offloading session from Outside_VLAN20 to Inside_VLAN20, skb.npu_flag=00000400 ses.state=00008200 ses.npu_state=0×00000100"
id=65308 trace_id=11 func=__if_queue_push_xmit line=392 msg="send out via dev-Inside_VLAN20, dst-mac-00:0c:29:39:f3:ac"
```

   - Reply 方向的流量匹配辅助会话和主会话返回。

```bash
Reply方向:
id=65308 trace_id=12 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=1, 192.168.20.10:2->192.168.10.10:0) tun_id=0.0.0.0 from Inside_VLAN20. type=0, code=0, id=2, seq=2879."
id=65308 trace_id=12 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00000386, reply direction"
id=65308 trace_id=12 func=br_fw_forward_dirty_handler line=282 msg="auxiliary ses proto=1 dev=19→18 192.168.10.10/2⇒192.168.20.10/8"
id=65308 trace_id=12 func=npu_handle_session44 line=1322 msg="Trying to offloading session from Inside_VLAN20 to Outside_VLAN20, skb.npu_flag=00000000 ses.state=00008200 ses.npu_state=0×00000100"
id=65308 trace_id=12 func=__if_queue_push_xmit line=392 msg="send out via dev-Outside_VLAN20, dst-mac-aa:bb:cc:00:30:00"
id=65308 trace_id=13 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=1, 192.168.20.10:2->192.168.10.10:0) tun_id=0.0.0.0 from Outside_VLAN10. type=0, code=0, id=2, seq=2879."
id=65308 trace_id=13 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00000386, reply direction"
id=65308 trace_id=13 func=__if_queue_push_xmit line=392 msg="send out via dev-Inside_VLAN10, dst-mac-00:0c:29:ae:55:70"
```

5. 查看对应会话，可以看到主会话为 192.168.10.10 → 192.168.20.10（Inside_VLAN10 → Outside_VLAN10），辅助会话有 reflect 标记，Outside_VLAN20 → Inside_VLAN20。

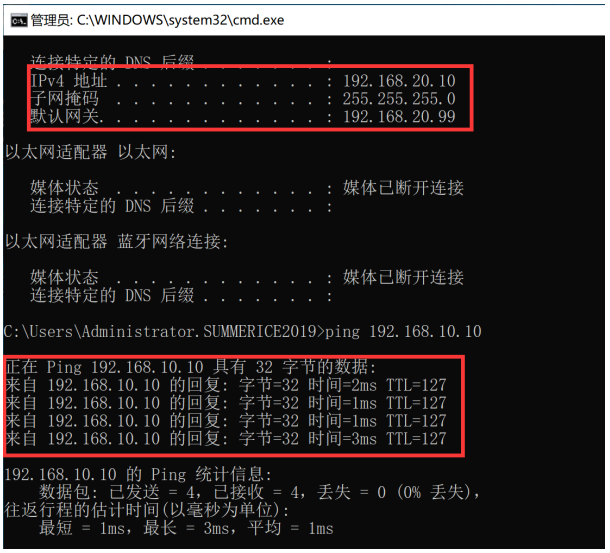
```bash
session info: proto=1 proto_state=00 duration=38 expire=21 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=120/2/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 3/0 rx speed(Bps/kbps): 3/0
orgin→sink: org pre→post, reply pre→post dev=16→17/17→16 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 192.168.10.10:2→192.168.20.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.20.10:2→192.168.10.10:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=15744 auth_info=0 chk_client_info=0 vd=0
serial=00000386 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×000100
no_ofld_reason:  npu-flag-off
reflect info 0:
dev=19→18/18→19
```

```
npu_state=0×000100
npu info: flag=0×00/0×00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0×0000/0×0000
vlifid=0/0, vtag_in=0×0000/0×0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
total reflect session num: 1
total session 1
```

6. 查看当前接口列表，可以看到 4 个 VLAN 接口的 index，与会话中的 dev 对应。

```bash
FortiGate # diagnose netlink interface list | grep VLAN
if=Inside_VLAN10 family=00 type=1 index=16 mtu=1500 link=0 master=0
if=Outside_VLAN10 family=00 type=1 index=17 mtu=1500 link=0 master=0
if=Inside_VLAN20 family=00 type=1 index=18 mtu=1500 link=0 master=0
if=Outside_VLAN20 family=00 type=1 index=19 mtu=1500 link=0 master=0
```

7. 使用 VLAN20 下的 PC 访问 VLAN 10 下的 PC，也可以正常访问。



重要

Forward-Domain 参考文档：https://community.fortinet.com/t5/FortiGate/Technical-Note-Configure-a-FortiGate-unit-in-Transparent-mode/ta-p/194458 ↗

vlanforward

重要

默认关闭，谨慎使用。

功能简介

FortiGate 在不配置 VLAN 接口的情况下，默认不转发那些携带了 VLAN - TAG 的数据，携带了 VLAN - TAG 的数据会被认为是非 IP 数据，直接全部丢弃（set vlanforward disable）—— 默认 disable。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port1
FortiGate_Transparent (port1) # show full-configuration | grep vlanforward
set vlanforward disable    //默认即为disable
```

假如 FortiGate 在不配置 VLAN 接口的情况下，默认转发那些携带了 VLAN - TAG 的数据（set vlanforward enable），也就是说 FortiGate（透明）上线一个新的环境，不清楚网络中是否跑了 VLAN，为了不影响那些不知道的 VLAN 的业务，于是开启 vlanforward，以便那些未知 VLAN 的业务可以正常通信。

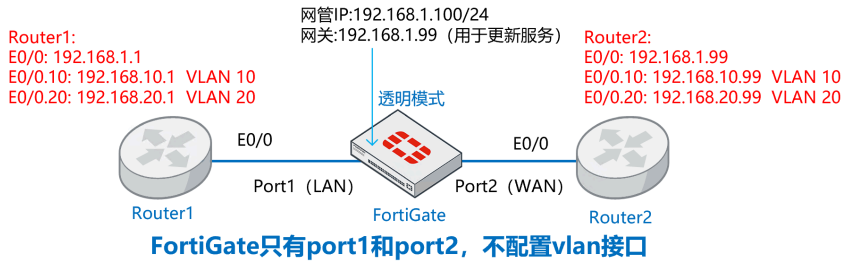
听起来美好，但是实际上问题很大，考虑到 FortiGate 默认所有接口都是一个 forward-domain 0，如此的情况很容易引起广播风暴。因此在目前版本中 vlanforward 默认就为 disable 的状态。

通常来说我们上线 FortiGate 透明的时候需要详细的了解客户的业务情况，网络中是否有 VLAN 和 VLAN - ID 多少这些信息都是必须要去了解的，vlanforward enable 这种省事的方法不可取，非特别特别的情况下尽量不要使用，同时也是为了避免忘记配置 forward-domain 而引起广播风暴的风险。通常来说 vlanforward 保持在 disable 的状态就好了，不要 enable。

重要

详情参考：https://community.fortinet.com/t5/FortiGate/Technical-Note-vlanforward-interface-parameter/ta-p/198171 ↗

网络拓扑



场景一：关闭 vlanforward

基本配置

Router1 和 Router2 运行了子接口 E0/0.10 和 E0/0.20，FortiGate 透明模式运行与 Router1 和 Router2 之间，三者的基础配置如下：

Router1：

```
hostname Router1
!
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
!
interface Ethernet0/0.10
encapsulation dot1Q 10
```

```
ip address 192.168.10.1 255.255.255.0
!
interface Ethernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
```

Router2：

```
hostname Router2
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
!
interface Ethernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.99 255.255.255.0
!
interface Ethernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.99 255.255.255.0
```

FortiGate：

```
FortiGate-VM64-KVM # config system global
FortiGate-VM64-KVM (global) # set hostname FortiGate_Transparent
FortiGate_Transparent (global) # set timezone 55
FortiGate_Transparent (global) # set language simch
FortiGate-VM64-KVM (global) # end
FortiGate_Transparent #
FortiGate_Transparent # config system settings

FortiGate_Transparent (settings) # set opmode transparent    // 修改FGT的运行模式为透明模式，默认为NAT路由模式。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。
FortiGate_Transparent (settings) # set manageip 192.168.1.100 255.255.255.0  // 配置可以管理防火墙的本地IP和网关，以便HTTP/SSH管理防火墙及防火墙的服务更新
FortiGate_Transparent (settings) # set gateway 192.168.1.99
FortiGate_Transparent (settings) # end
Changing to TP mode

（MGMT1或MGMT2口默认有管理权限），以要通过port1（LAN）接口管理设备为例，开启port1（LAN）管理FGT的命令如下：
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port1
FortiGate_Transparent (port1) # set allowaccess https http ping ssh    //允许网管协议从Port1接口通过https/http/SSH/Ping访问透明模式的FortiGate
FortiGate_Transparent (port1) # end

port1和port2都是默认配置，同时也没有配置任何安全策略，纯透明防火墙初始化空配置。
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port1
FortiGate_Transparent (port1) # show full-configuration | grep vlanforward
        set vlanforward disable    //默认vlanforward是disable的
```

实验步骤

1. 首先测试 Router1 的 e0/0 192.168.1.1 到 192.168.1.99 类似于不带 vlan-tag 的本地 vlan vlan1 的业务。

```
Router1#ping 192.168.1.99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.99, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

2. 结果不通，在 FortiGate 上抓包，

```
FortiGate_Transparent # diagnose sniffer packet any "arp or icmp" 4 0 l
interfaces=[any]
filters=[arp or icmp]
2019-07-03 18:05:12.481488 port1 in arp who-has 192.168.1.99 (aa:bb:cc:0:20:0) tell 192.168.1.1
2019-07-03 18:05:12.481508 port2 out arp who-has 192.168.1.99 (aa:bb:cc:0:20:0) tell 192.168.1.1
```

```
2019-07-03 18:05:12.486286 port2 in arp reply 192.168.1.99 is-at aa:bb:cc:0:20:0
2019-07-03 18:05:12.486295 port1 out arp reply 192.168.1.99 is-at aa:bb:cc:0:20:0
2019-07-03 18:05:15.442315 port1 in 192.168.1.1 → 192.168.1.99: icmp: echo request
2019-07-03 18:05:17.444526 port1 in 192.168.1.1 → 192.168.1.99: icmp: echo request
2019-07-03 18:05:19.444349 port1 in 192.168.1.1 → 192.168.1.99: icmp: echo request
2019-07-03 18:05:21.444162 port1 in 192.168.1.1 → 192.168.1.99: icmp: echo request
2019-07-03 18:05:23.444217 port1 in 192.168.1.1 → 192.168.1.99: icmp: echo request
```

3. ARP 解析正常，但是 icmp 请求被防火墙丢弃，deubug flow 分析不通的原因。

```
diagnose debug flow filter addr 192.168.1.1
diagnose debug flow filter  proto 1
diagnose debug flow show function-name enable
diagnose debug flow trace start 10
diagnose debug enable

id=20085 trace_id=1 func=print_pkt_detail line=5428 msg="vd-root:0 received a packet(proto=1, 192.168.1.1:10-
>192.168.1.99:2048) from port1. type=8, code=0, id=10, seq=0."
id=20085 trace_id=1 func=init_ip_session_common line=5593 msg="allocate a new session-0000007d"
id=20085 trace_id=1 func=br_fw_forward_handler line=565 msg="Denied by forward policy check"    //没有配置安全策略允许，自然
是会丢弃的。
```

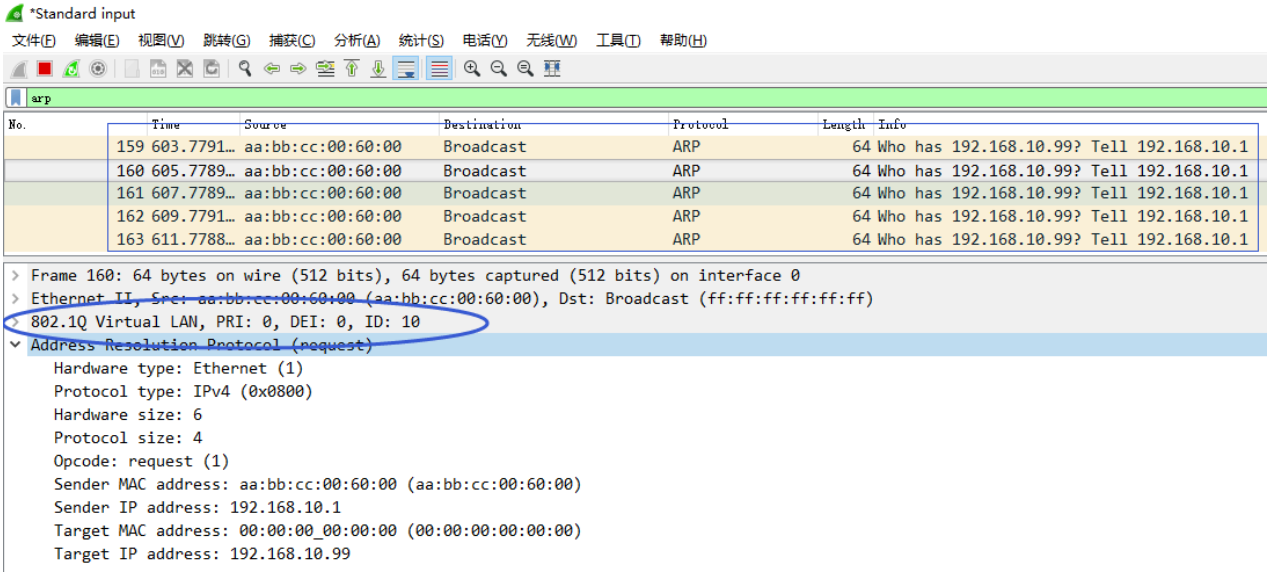
4. 测试 Router1 的 E0/0.10 192.168.10.1 → 192.168.10.99 的业务，此业务携带了 VLAN - TAG 10。

```
Router1#ping 192.168.10.99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.99, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

5. 结果不通，在 FGT 上抓包。

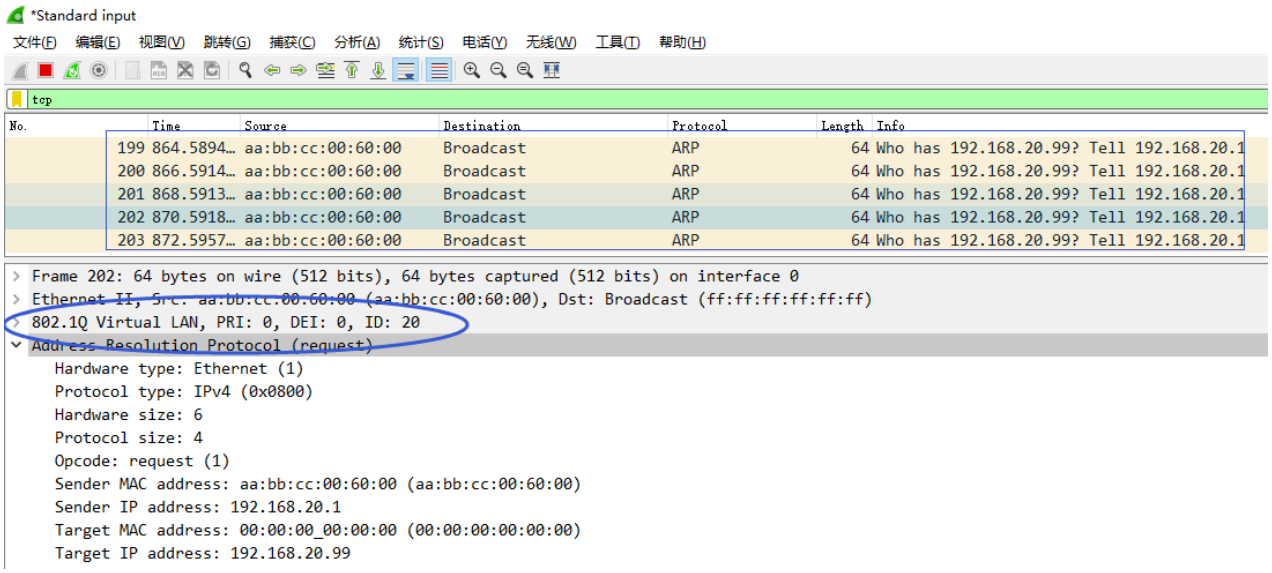
```
FortiGate_Transparent # diagnose sniffer packet any "arp or icmp" 4 0 l
interfaces=[any]
filters=[arp or icmp]
```

6. 什么都抓不到，通过镜像抓包查看，发现携带了 vlan-tag 10 的 ARP 报文都无法被转发，也就是说接口下的这条命令：set vlanforward disable，且 FGT 本地没有对应 vlan-tag 10 的接口，会丢弃携带了 vlan-tag 10 的任何报文，包括 ARP 报文，甚至数据报文 CPU 都不上送，携带了 VLAN - TAG 的数据会被当做非 IP 流量数据直接丢弃。



7. 测试 VLAN20 的数据和 vlan10 同样的效果。

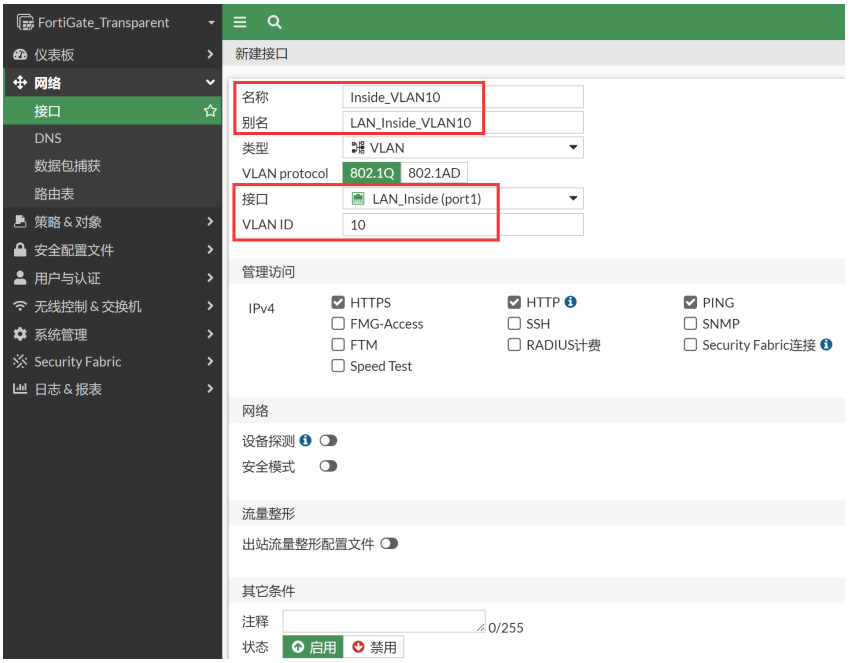
```
Router1#ping 192.168.20.99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.99, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

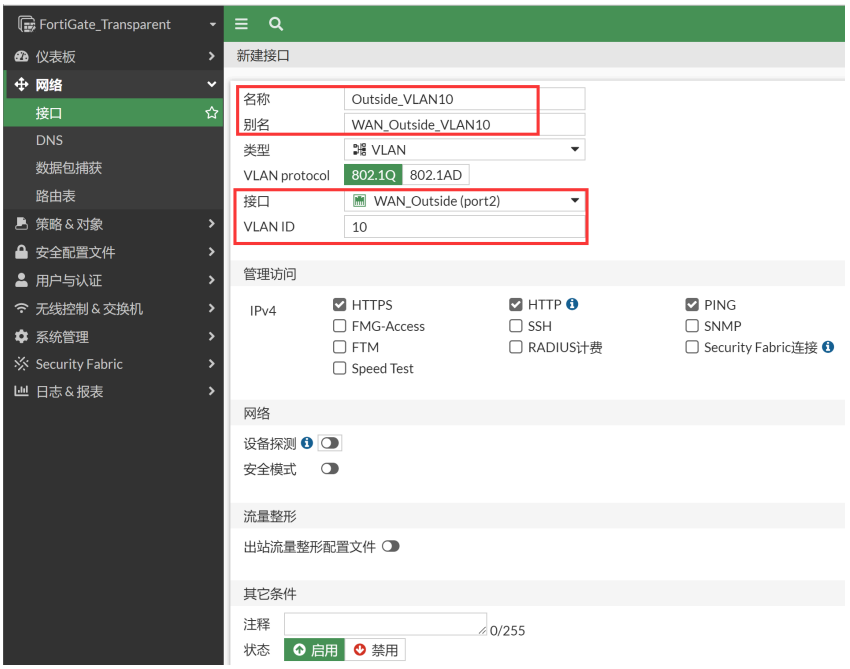


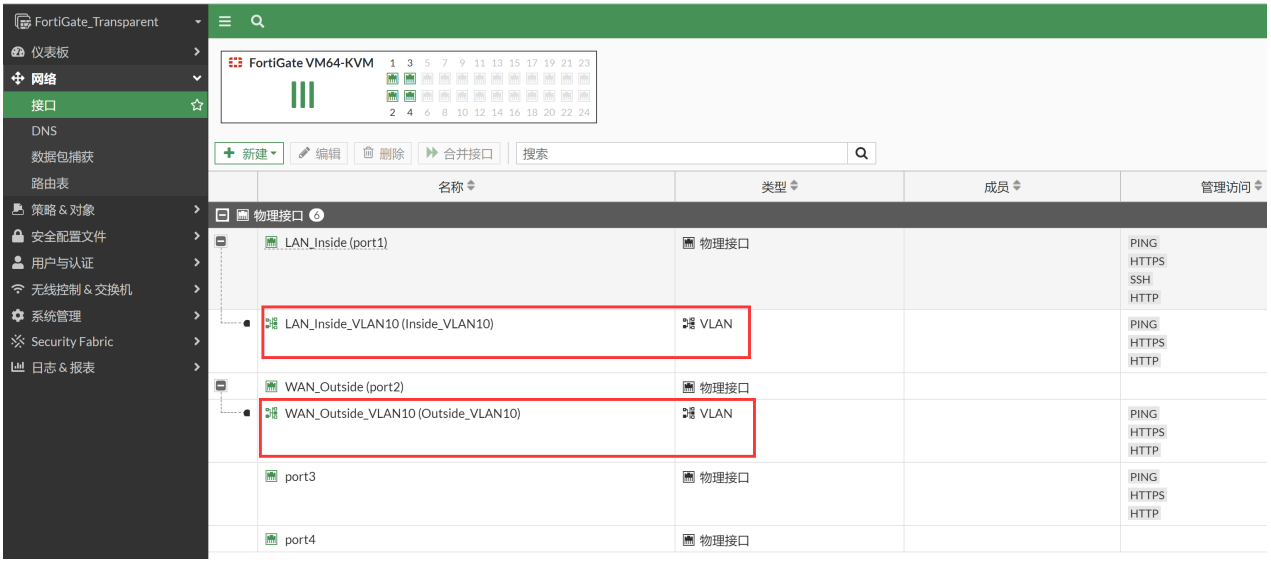
8. 查看 Router1 的 ARP 表，发现无法学习到相关的 ARP 条目。

```
Router1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
 …
Internet  192.168.10.99           0   Incomplete      ARPA
Internet  192.168.20.99           0   Incomplete      ARPA
 …
```

9. 在 FGT 的 Port1 和 Port2 上增加 Inside_VLAN10 和 Outside_VLAN10 的两个 VLAN 10 的接口。







.0. 将两个 VLAN 接口加入到 forward-domain 10 里。

```
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit Inside_VLAN10
FortiGate_Transparent (Inside_VLAN10) # set forward-domain 10
FortiGate_Transparent (Inside_VLAN10) # next
FortiGate_Transparent (interface) # edit Outside_VLAN10
FortiGate_Transparent (Outside_VLAN10) # set forward-domain 10
FortiGate_Transparent (Outside_VLAN10) # end
```

重要

配置 VLAN - ID 的场景下则必须配置“forward-domain”否则 FGT 的 MAC 地址表项会错乱翻转，业务不通，甚至存在广播风暴的风险。

重要

假设不配置 forward-domain，由于 port2、port2.10（Outside_VLAN10）都是属于同一个广播域，那么 192.168.10.1 请求 192.168.10.99 的 ARP 广播请求会被发送到 port2、port2.10（Outside_VLAN10）甚至 port1 都会被广播出去，同时由于 Router2 的 E0/0（192.168.1.99）和 E0/0.10（192.168.10.99）的 MAC 地址一样都是：aa:bb:cc:00:20:00，由于在同一个广播域中的不同接口不能存在相同的 MAC 地址，此时 port2 也可以学习到 aa:bb:cc:00:20:00, 同时 port2.10（Outside_VLAN10）也可以学习到 aa:bb:cc:00:20:00，这样会造成 MAC 地址表冲突，同时 MAC 表中只能存在一个 MAC 表项，会操作 MAC 表翻转，如果配置全通策略，甚至可能引起环路的风险。因此配置 VLAN - ID 务必将 forward-domain 配置上，这是一个很必要的配置操作。

1. 查看配置了 forward-domain 的 FortiGate 的 MAC 表和 Router1 的 arp 表。

```
FortiGate_Transparent # diag netlink brctl name host root.b
show bridge control interface root.b host.
fdb: size=2048, used=6, num=10, depth=2
Bridge root.b host table
port no device  devname mac addr                ttl     attributes
...
  6     14      Outside_VLAN10  aa:bb:cc:00:20:00       5        Hit(5)
  2     4       port2   aa:bb:cc:00:20:00       0        Hit(0)
...
```

```
Router1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
...
Internet  192.168.1.99           0   aabb.cc00.2000  ARPA   Ethernet0/0
Internet  192.168.10.99          0   aabb.cc00.2000  ARPA   Ethernet0/0.10
...
```

2. 此时再看 VLAN10 的业务，在 Router1 上使用 192.168.10.1 ping 192.168.10.99。

```
Router1#ping 192.168.10.99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.99, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

3. 虽然还是不通，但是情况已经不一样了，明显 VLAN10 的数据全部上送到了 CPU 进行处理，ARP 可正常学习，只是 ping 不通，原因是防火墙没有配置策略，所以匹配了默认的 drop 策略。

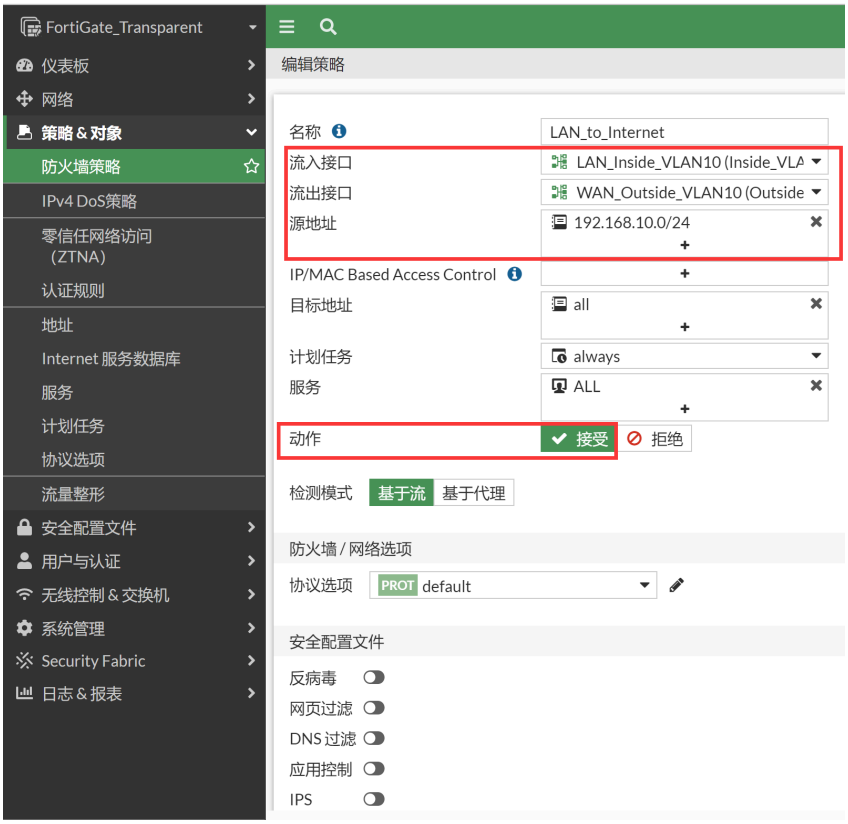
```
FortiGate_Transparent # diagnose sniffer packet any "arp or icmp" 4 0 l
interfaces=[any]
filters=[arp or icmp]
2019-07-03 18:24:49.697190 Inside_VLAN10 in arp who-has 192.168.10.99 tell 192.168.10.1
2019-07-03 18:24:49.697218 Outside_VLAN10 out arp who-has 192.168.10.99 tell 192.168.10.1
2019-07-03 18:24:49.698089 Outside_VLAN10 in arp reply 192.168.10.99 is-at aa:bb:cc:0:20:0
2019-07-03 18:24:49.698102 Inside_VLAN10 out arp reply 192.168.10.99 is-at aa:bb:cc:0:20:
2019-07-03 18:24:51.700405 Inside_VLAN10 in 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:24:53.700372 Inside_VLAN10 in 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:24:55.704350 Inside_VLAN10 in 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:24:57.704314 Inside_VLAN10 in 192.168.10.1 → 192.168.10.99: icmp: echo request
```

```
Router1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1            -   aabb.cc00.6000  ARPA   Ethernet0/0
Internet  192.168.1.99           5   aabb.cc00.2000  ARPA   Ethernet0/0
Internet  192.168.10.1           -   aabb.cc00.6000  ARPA   Ethernet0/0.10
```

```
Internet  192.168.10.99          0   aabb.cc00.2000  ARPA   Ethernet0/0.10  // arp可以学习到
Internet  192.168.20.1           -   aabb.cc00.6000  ARPA   Ethernet0/0.20
```

```
diagnose debug flow filter addr 192.168.10.1
diagnose debug flow filter  proto 1
diagnose debug flow show function-name enable
diagnose debug flow trace start 10
diagnose debug enable

id=20085 trace_id=2 func=print_pkt_detail line=5428 msg="vd-root:0 received a packet(proto=1, 192.168.10.1:17-
>192.168.10.99:2048) from Inside_VLAN10. type=8, code=0, id=17, seq=0."
id=20085 trace_id=2 func=init_ip_session_common line=5593 msg="allocate a new session-00000244"
id=20085 trace_id=2 func=br_fw_forward_handler line=565 msg="Denied by forward policy check"
```

4. 此时在 FortiGate 上配置 VLAN 10 的放通安全策略。



5. VLAN10 的数据可以正常通信。

```
Router1#ping 192.168.10.99 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.10.99, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/2 ms
```

```
FortiGate_Transparent # diagnose sniffer packet any "icmp or arp" 4 0 l
interfaces=[any]
filters=[icmp or arp]
2019-07-03 18:57:39.090299 Inside_VLAN10 in 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:57:39.090376 Outside_VLAN10 out 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:57:39.090379 port2 out 192.168.10.1 → 192.168.10.99: icmp: echo request
2019-07-03 18:57:39.091154 Outside_VLAN10 in 192.168.10.99 → 192.168.10.1: icmp: echo reply
2019-07-03 18:57:39.091166 Inside_VLAN10 out 192.168.10.99 → 192.168.10.1: icmp: echo reply
2019-07-03 18:57:39.091168 port1 out 192.168.10.99 → 192.168.10.1: icmp: echo reply
```

总结

在默认的“set vlanforward disable”的情况下：

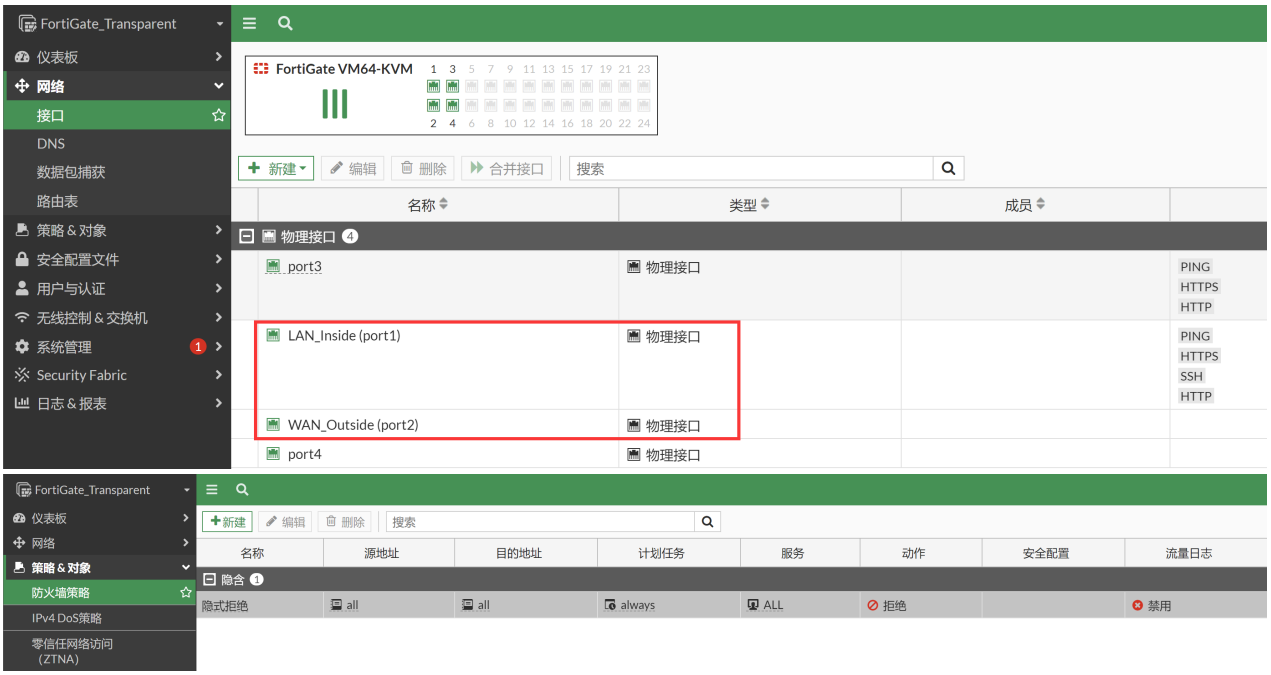
- 如果 FGT 本地存在 VLAN - ID 接口（此处为类似 VLAN1 的接口 port1）则数据会上送 CPU 处理，如果配置 VLAN10 的接口，则接收到了 VLAN - TAG10 的 VLAN 数据也会上送 CPU 处理，那么通不通就完全看安全策略是否放通了。
- 如果 FGT 本地不存在 VLAN - ID 的接口，收到了 VLAN - TAG 的数据，且本地并没有相应的 VLAN 接口，那么这样的数据将会被 FGT 底层当做非 IP 数据丢弃，上送 CPU 的机会都没有，直接丢弃，连 arp 都不处理，直接丢弃。

场景二：开启 vlanforward

实验步骤

1. 在 port1 和 port2 上开启 vlanforward（不配置任何 VLAN 接口和安全策略）。

```
config system interface
    edit "port1"
        set vdom "root"
        set allowaccess ping https ssh http
        set vlanforward enable
        set type physical
        set alias "LAN_Inside"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set allowaccess ping https ssh http
        set vlanforward enable
        set type physical
        set alias "WAN_Outside"
        set snmp-index 2
    next
end
```



2. 再次测试一下 Router1 到 Router2 的 VLAN 20 的业务，VLAN 20 的数据是通的。

```
Router1#ping 192.168.20.99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.99, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

3. 但是在 FGT 上抓取不到任何的 VLAN20 相关的数据，数据没有上送到 CPU 处理，直接被 bypass 掉了。

```
FortiGate_Transparent # diagnose sniffer packet any "icmp or arp" 4 0 l
interfaces=[any]
filters=[icmp or arp]
```

总结

如果 FGT 上没有创建 VLAN20 的这个接口，同时接口开启了 vlanforward，那么相当于这些未知的携带了 VLAN - TAG 的数据，将会统统 bypass 掉，不上 CPU 处理直接转发走。

如果不熟悉 vlanforward 和 forward-domain 的工作原理，这样的 bypass 流量风险比较大，因为那些 bypass 的流量 FGT 完全不可控，也不可见，一旦出问题完全无法掌控，因此 vlanforward 不建议开启，保持默认 disable 即可。

Remapping VLAN ID (VLAN NAT，特殊场景使用)

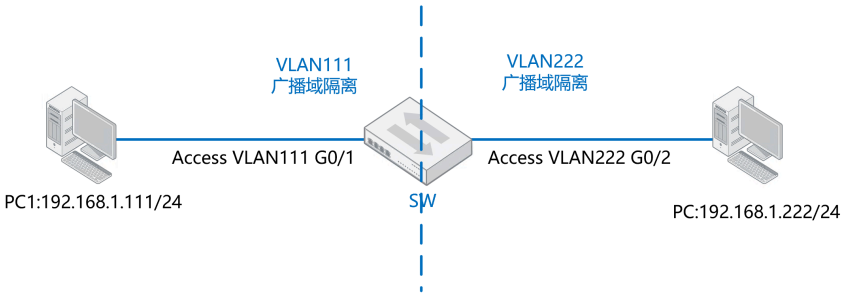
> **重要**
>
> VLAN NAT，特殊场景使用。

简介

Remapping VLAN 简单的可以理解为，透明模式下 VLAN ID 的 NAT 转换。比如数据本来在 VLAN104 中，一旦需要透明穿过 VLAN105 去访问目的主机，这个时候 FGT 可以将原本携带了 VLAN-Tag 104 的数据转换为携带 VLAN-Tag 105 的数据，进行了一个 VLAN-Tag 的替换，以便数据可以在两个 VLAN 之间通信。

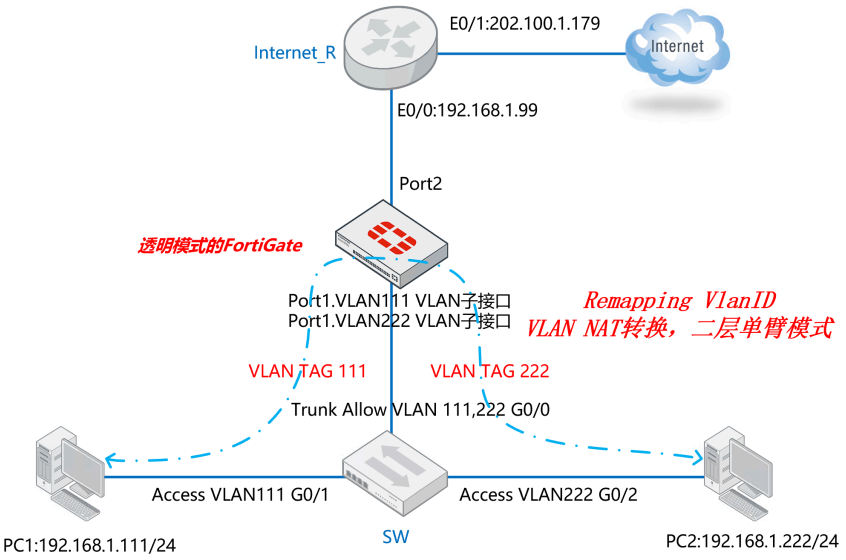
> **重要**
>
> 谁说的跨 VLAN 之间通信的数据必须经过路由器（三层设备），不，Fortigate 二层也可以！但是这个技术使用场景有限，一般用在私有云环境中的东西向流量的控制和过滤，用的比较少，了解即可。

原始网络拓扑和需求



需求：要求 PC1 和 PC2 之间虽然是相同的网段，但是要相互隔离，同时要求防火墙策略对二者进行访问控制，PC1 可以访问 PC2，但是 PC2 不能访问 PC1，同时要求 PC1 和 PC2 都可以正常的访问互联网。

解决方案拓扑



基础配置

SW:

```
interface GigabitEthernet0/0
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface GigabitEthernet0/1
switchport access vlan 111
switchport mode access
no shutdown
spanning-tree portfast edge
!
interface GigabitEthernet0/2
switchport access vlan 222
switchport mode access
no shutdown
spanning-tree portfast edge
```

Internet_R：

```
interface Ethernet0/1
ip address 202.100.1.179 255.255.255.0
no shutdown
ip nat outside
!
interface Ethernet0/0
ip address 192.168.1.99 255.255.255.0
no shutdown
ip nat inside
!
ip route 0.0.0.0 0.0.0.0 202.100.1.192
!
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/1 overload
```

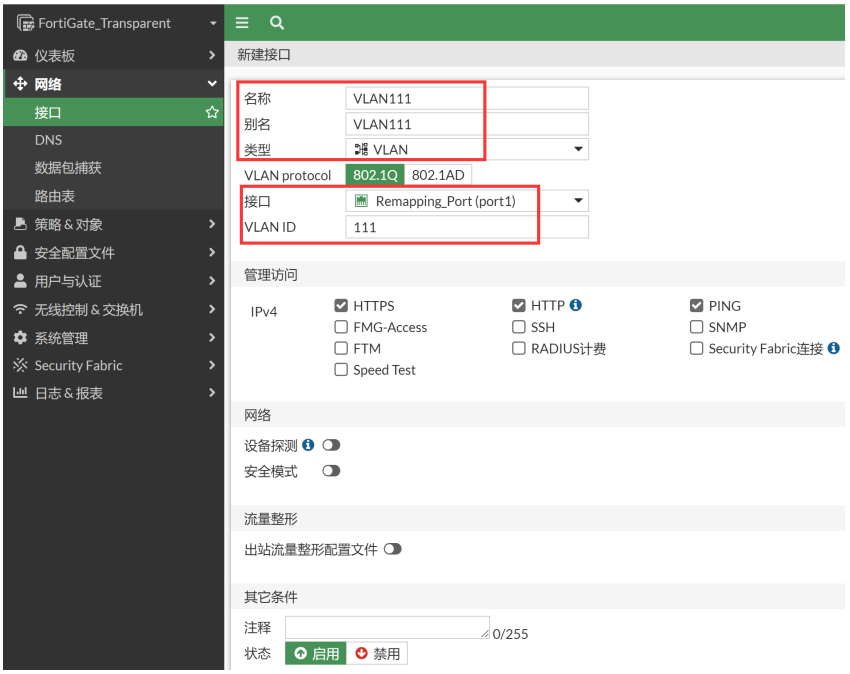
FortiGate：

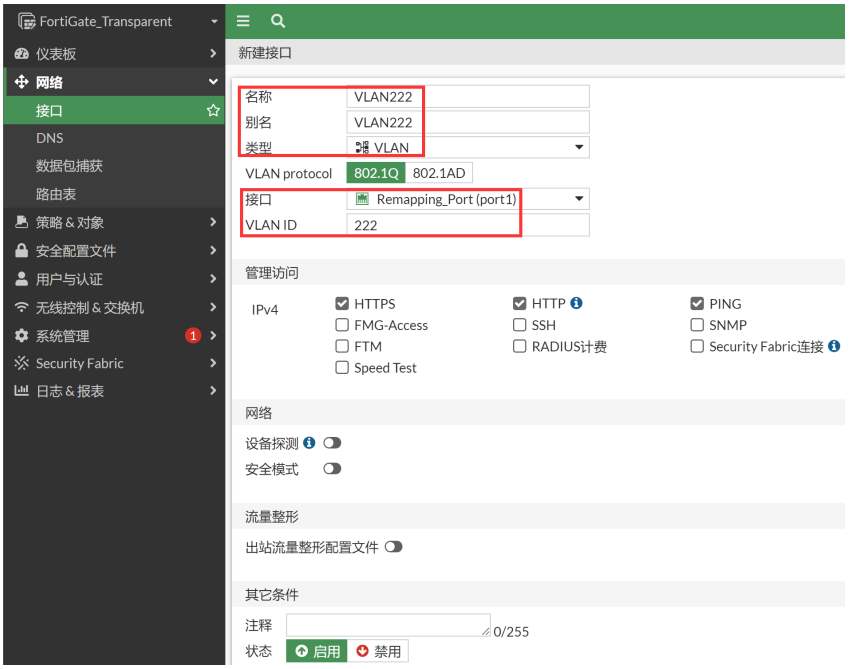
```
FortiGate-VM64-KVM # config system global
FortiGate-VM64-KVM (global) # set hostname FortiGate_Transparent
FortiGate_Transparent (global) # set timezone 55
FortiGate_Transparent (global) # set language simch
FortiGate-VM64-KVM (global) # end
FortiGate_Transparent # config system settings
FortiGate_Transparent (settings) # set opmode transparent    //修改FGT的运行模式为透明模式，默认为NAT路由模式。注意切换透明模式防火墙需要防火墙没有相关接口、策略、路由等配置。
FortiGate_Transparent (settings) # set manageip 192.168.1.100 255.255.255.0    //配置可以管理防火墙的本地IP和网关，以便HTTP/SSH管理防火墙及防火墙的服务更新。
FortiGate_Transparent (settings) # set gateway 192.168.1.99
FortiGate_Transparent (settings) # end
Changing to TP mode

MGMT1或MGMT2口默认有管理权限，以要通过port2 接口管理设备为例，开启port2管理FGT的命令如下：
FortiGate_Transparent # config system interface
FortiGate_Transparent (interface) # edit port2
FortiGate_Transparent (port1) # set allowaccess https http ping ssh    //允许网管协议从Port2接口通过https/http/SSH/Ping访问透明模式的FortiGate
FortiGate_Transparent (port1) # end
```

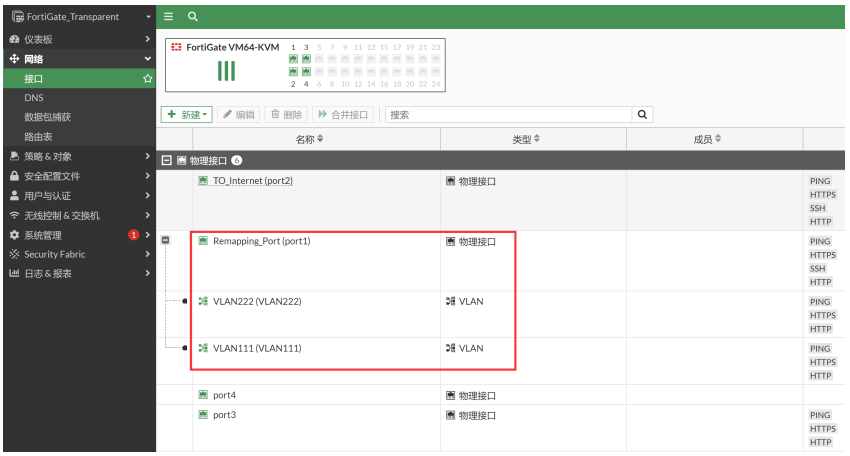
配置步骤

1. 在 FortiGate 上创建 VLAN 子接口，VLAN111 和 VLAN222，分别绑定在 port1 和 port2 下。

> **重要**
>
> 建议在不需要的情况下关闭接口配置下的“设备探测”（Device detection）功能，该功能用于 MAC 地址厂商设备信息识别及 MAC 地址过滤，会消耗较多的设备资源。





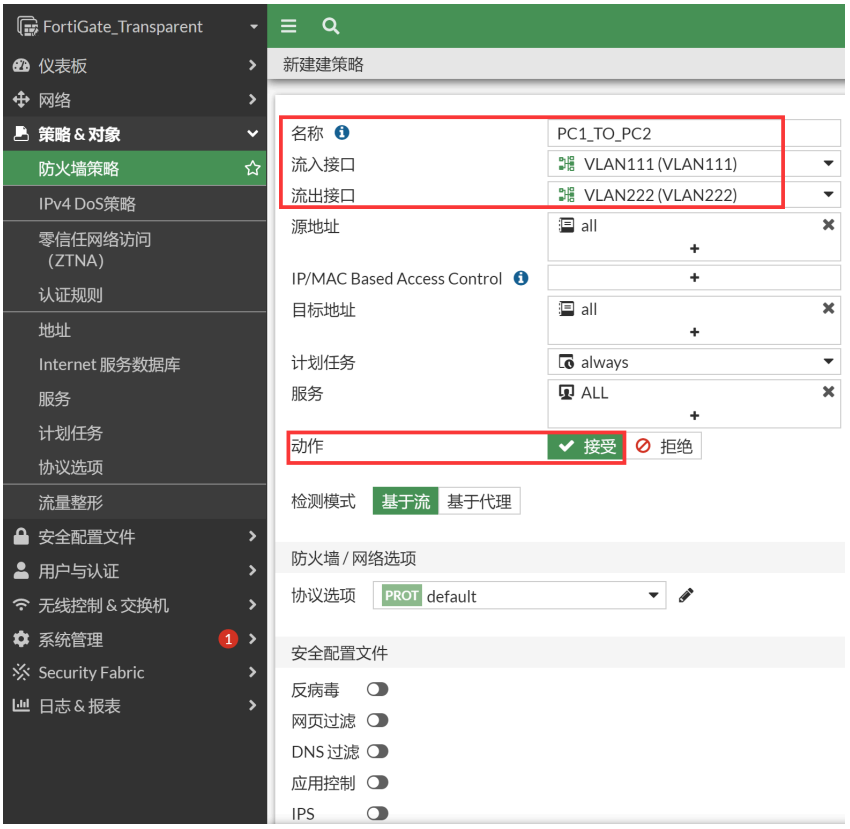


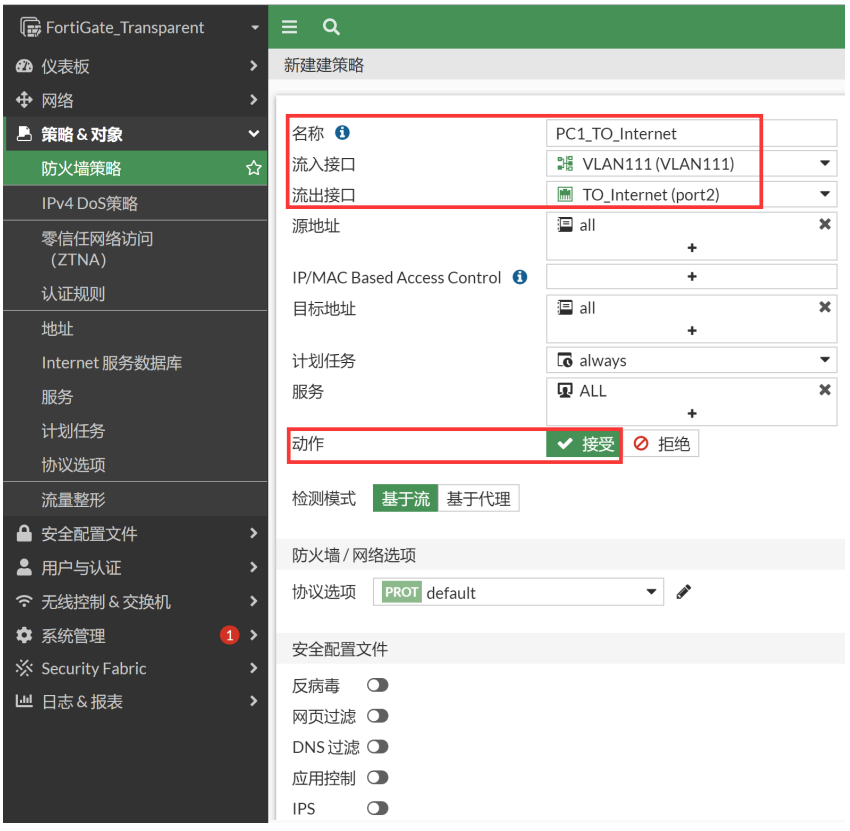
2. 页面右上角打开 Web CLI，将 VLAN111、VLAN222、port2 三个接口划入到同一个 Forward-Domain 里。这样就将 VLAN111 和 VLAN222 之间的广播域打通了，VLAN111 去访问 VLAN222，则 VLAN TAG 111 就会被替换成 VLAN TAG222，反过来也一样的效果。同时 VLAN111 和 VLAN222 和 Port2 也是一个广播域，这样 VLAN111 和 VLAN222 配置策略后，可以通过 Port2 上互联网。

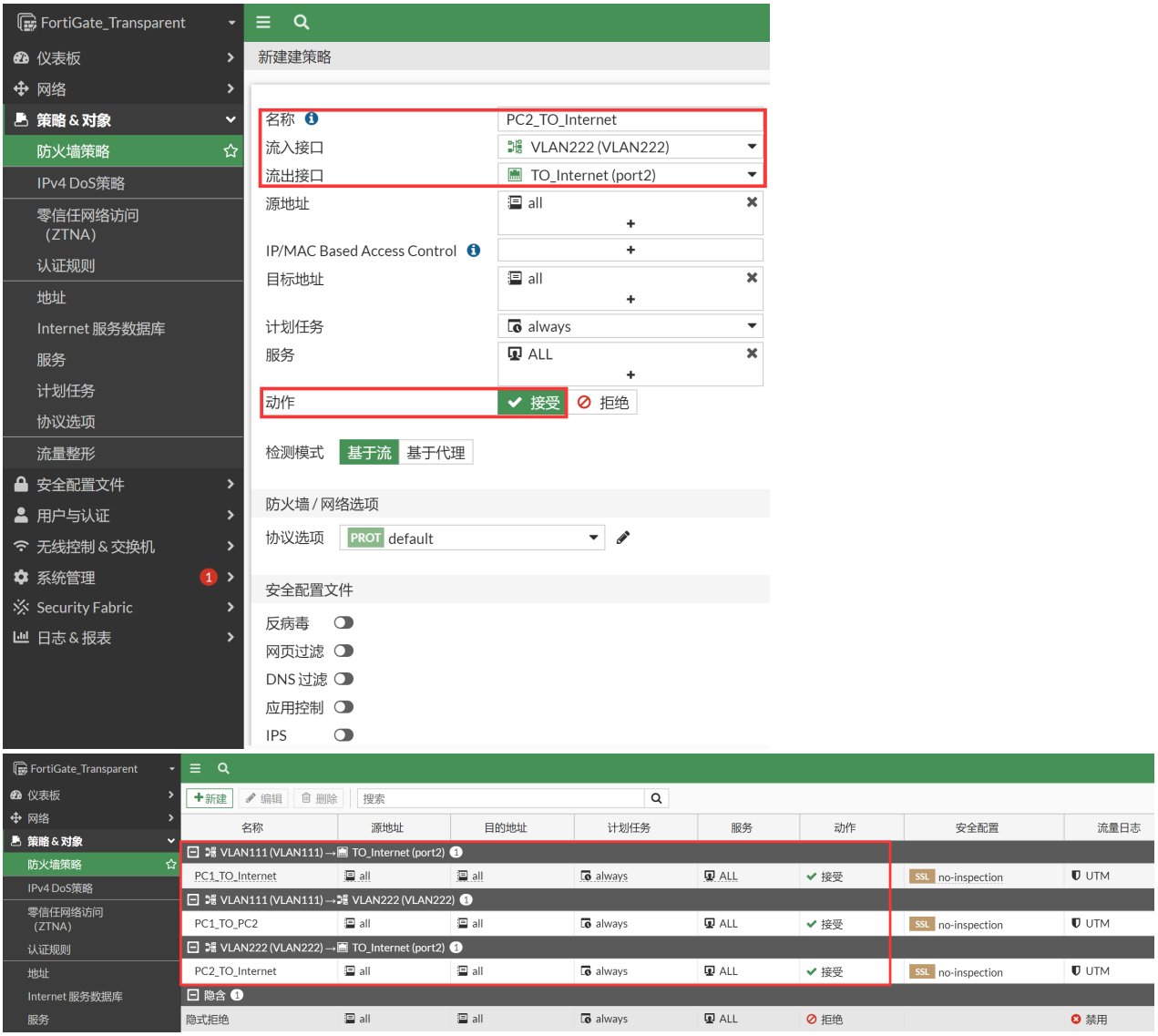
切换到命令行，将三个接口划入到同一个Forward-Domain里：

```
config system interface
    edit "VLAN111"
        set forward-domain 111222
        set alias "VLAN111"
        set interface "port1"
        set vlanid 111
    next
    edit "VLAN222"
        set forward-domain 111222
        set alias "VLAN222"
        set interface "port1"
        set vlanid 222
    next
    edit "port2"
        set allowaccess ping https ssh http
        set forward-domain 111222
        set alias "TO_Internet"
    next
end
```

3. 这样流量是通过防火墙策略控制的，因此接下来需要配置防火墙策略，实现：PC1 和访问 PC2，但是 PC2 不能访问 PC1，同时要求 PC1 和 PC2 都可以正常的访问互联网。


FortiGate_Transparent
仪表板
网络
策略 & 对象
防火墙策略
IPv4 DoS策略
零信任网络访问（ZTNA）
认证规则
地址
Internet 服务数据库
服务
计划任务
协议选项
流量整形
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新建建策略
名称
PC1_TO_PC2
流入接口
VLAN111 (VLAN111)
流出接口
VLAN222 (VLAN222)
源地址
all
IP/MAC Based Access Control
目标地址
all
计划任务
always
服务
ALL
动作
接受
拒绝
检测模式
基于流
基于代理
防火墙 / 网络选项
协议选项
PROT default
安全配置文件
反病毒
网页过滤
DNS 过滤
应用控制
IPS


FortiGate_Transparent
仪表板
网络
策略 & 对象
防火墙策略
IPv4 DoS策略
零信任网络访问（ZTNA）
认证规则
地址
Internet 服务数据库
服务
计划任务
协议选项
流量整形
安全配置文件
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新建建策略
名称
PC1_TO_Internet
流入接口
VLAN111 (VLAN111)
流出接口
TO_Internet (port2)
源地址
all
IP/MAC Based Access Control
目标地址
all
计划任务
always
服务
ALL
动作
接受
拒绝
检测模式
基于流
基于代理
防火墙 / 网络选项
协议选项
PROT default
安全配置文件
反病毒
网页过滤
DNS 过滤
应用控制
IPS



结果验证

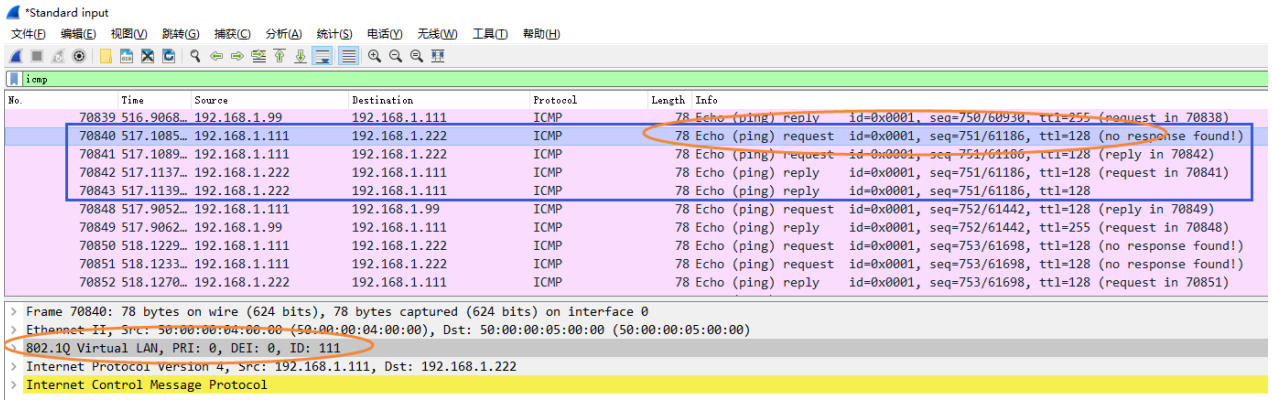
1. PC1 访问 PC2 和 Internet，访问成功。

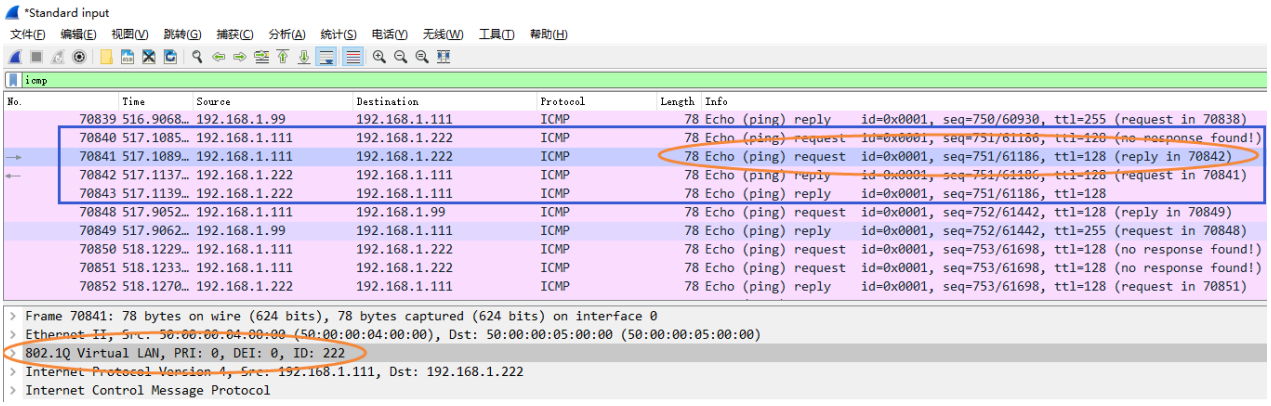
```
管理员: C:\WINDOWS\system32\cmd.exe

   连接特定的 DNS 后缀 . . . . . :
   IPv4 地址 . . . . . . . . . . . . : 192.168.1.111
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.1.99

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

C:\Users\Administrator.SUMMERICE2019>ping 192.168.1.222

正在 Ping 192.168.1.222 具有 32 字节的数据:
来自 192.168.1.222 的回复: 字节=32 时间=1ms TTL=255
来自 192.168.1.222 的回复: 字节=32 时间=1ms TTL=255
来自 192.168.1.222 的回复: 字节=32 时间=1ms TTL=255
来自 192.168.1.222 的回复: 字节=32 时间=1ms TTL=255

192.168.1.222 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 1ms, 平均 = 1ms

C:\Users\Administrator.SUMMERICE2019>ping 114.114.114.114

正在 Ping 114.114.114.114 具有 32 字节的数据:
来自 114.114.114.114 的回复: 字节=32 时间=1ms TTL=255
来自 114.114.114.114 的回复: 字节=32 时间=1ms TTL=255
来自 114.114.114.114 的回复: 字节=32 时间=1ms TTL=255
来自 114.114.114.114 的回复: 字节=32 时间=1ms TTL=255

114.114.114.114 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 1ms, 平均 = 1ms
```

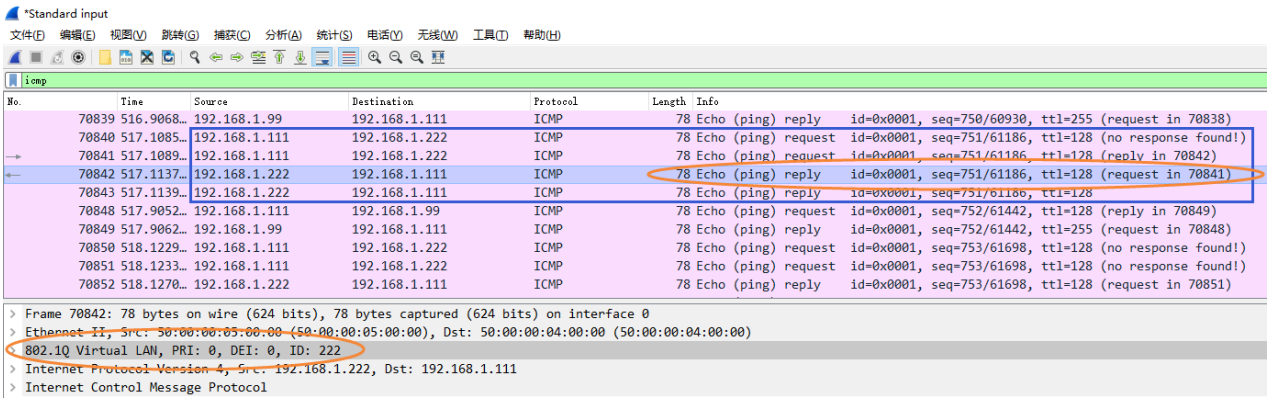
2. 抓包查看 PC1 访问 PC2。

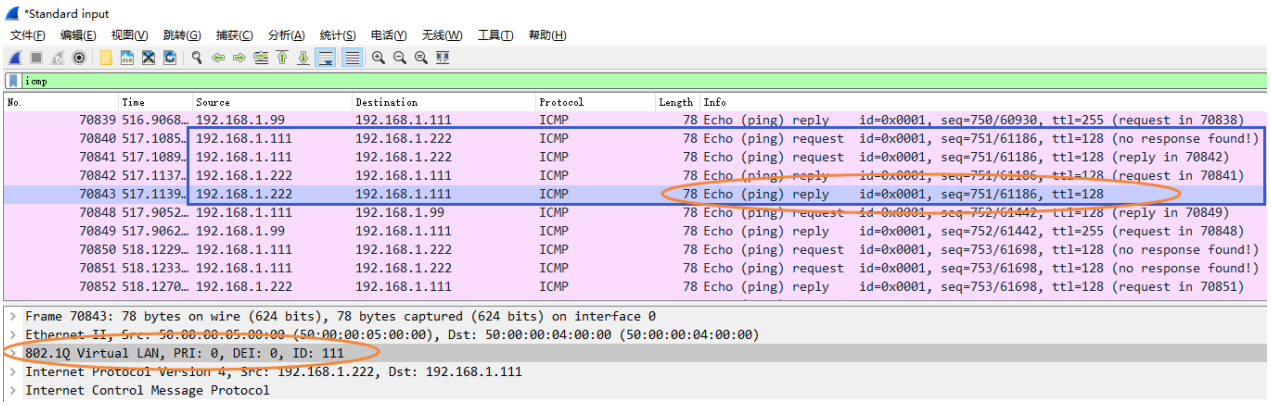
```
FortiGate_Transparent # dia sni pac any "host 192.168.1.222 and icmp" 4
interfaces=[any]
filters=[host 192.168.1.222 and icmp]
0.765120 VLAN111 in 192.168.1.111 → 192.168.1.222: icmp: echo request
0.765133 VLAN222 out 192.168.1.111 → 192.168.1.222: icmp: echo request
0.765136 port1 out 192.168.1.111 → 192.168.1.222: icmp: echo request
0.769002 VLAN222 in 192.168.1.222 → 192.168.1.111: icmp: echo reply
0.769012 VLAN111 out 192.168.1.222 → 192.168.1.111: icmp: echo reply
0.769014 port1 out 192.168.1.222 → 192.168.1.111: icmp: echo reply
```

3. 在 FortiGate 上镜像 port1 的报文。分析一个 PC1 ping PC2 的过程 VLAN TAG 的变化，非常直接的可以看到请求报文的 VLAN - TAG 从 111 变成了 222，回复报文的 VLAN - TAG 从 222 变成了 111，实现了 remapping VlanID 的功能。







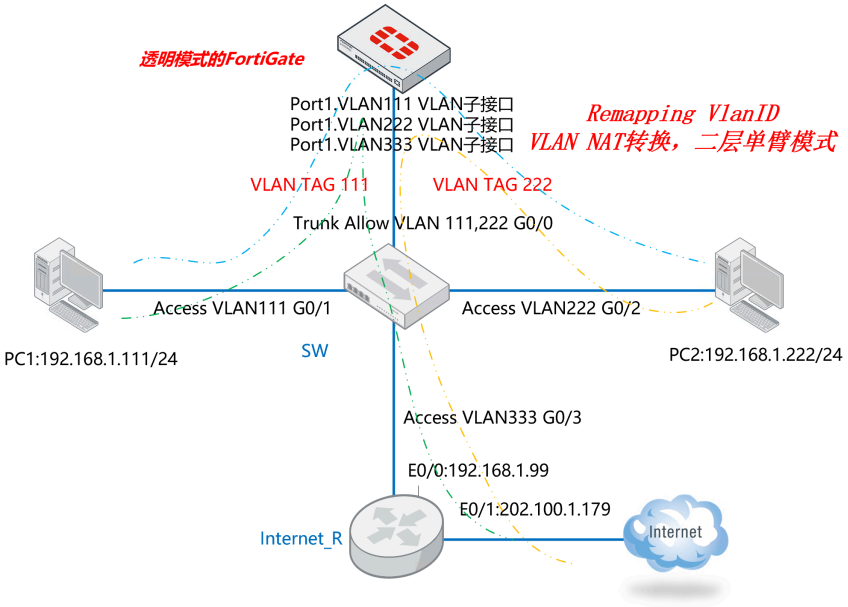


补充

当然组网可以调整为：防火墙完全单臂部署，PC1 属于 VLAN111、PC2 属于 VLAN222、Internet_R 属于 VLAN333，默认情况下相互隔离。

FGT 配置 remapping VlanID，可以让 Port1 子接口 VLAN111、VLAN222、VLAN333 加入到同一个 Forward-Domain 中，让三者广播域互通，同时 FGT 可以通过策略控制三者的流量。

这种方案可以用来作为内网相同网段主机的安全隔离，针对二层环境下东西向流量进行安全控制。


透明模式的FortiGate
Port1.VLAN111 VLAN子接口
Port1.VLAN222 VLAN子接口
Port1.VLAN333 VLAN子接口
Remapping VlanID
VLAN NAT转换，二层单臂模式
VLAN TAG 111
VLAN TAG 222
Trunk Allow VLAN 111,222 G0/0
Access VLAN111 G0/1
Access VLAN222 G0/2
PC1:192.168.1.111/24
SW
PC2:192.168.1.222/24
Access VLAN333 G0/3
E0/0:192.168.1.99
E0/1:202.100.1.179
Internet_R
Internet

旁路模式

- 旁路模式典型案例

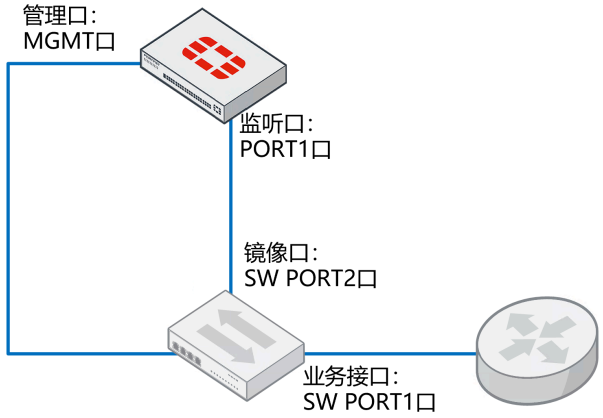
旁路模式典型案例

网络需求

有时客户为了保证重要业务连续不中断，且避免引入新的单点故障点：

- 需采用旁路部署的方式部署防火墙
- 启用防火墙的 UTM 功能，对防病毒、IPS 等安全事件进行检测及记录

网络拓扑



配置要点

- 交换机配置（根据实际情况进行配置，建议由客户方维护人员进行配置）：配置 sw G1/0 为镜像口，将 sw1 口上下行流量镜像到 sw2 口
- 防火墙配置：
  - 配置接口，开启 port1 的 sniffer 模式，让 port1 口可以对收到的数据包进行分析
  - 编辑 IPS、防病毒 规则参数（本文中均以 default 规则为例）
  - 配置防火墙策略，将 IPS、防病毒等规则应用于 port1 口
  - 开启 IPS、防病毒等日志

配置步骤

1. 接口启用 sniffer 模式。

```
FGT #config system interface
FGT (interface) #edit port1
FGT (port1) #set ips-sniffer-mode enable
FGT (port1) #end
```

2. 根据实际情况，编辑相应的 UTM 规则参数，例如 IPS（如 default 规则）、防病毒（default 规则）以及 UTM 功能日志记录等，具体参数设置，详见 UTM 安全应用章节。

3. 配置防火墙 sniffer 策略，将 UTM 安全策略加载与 port1 口。

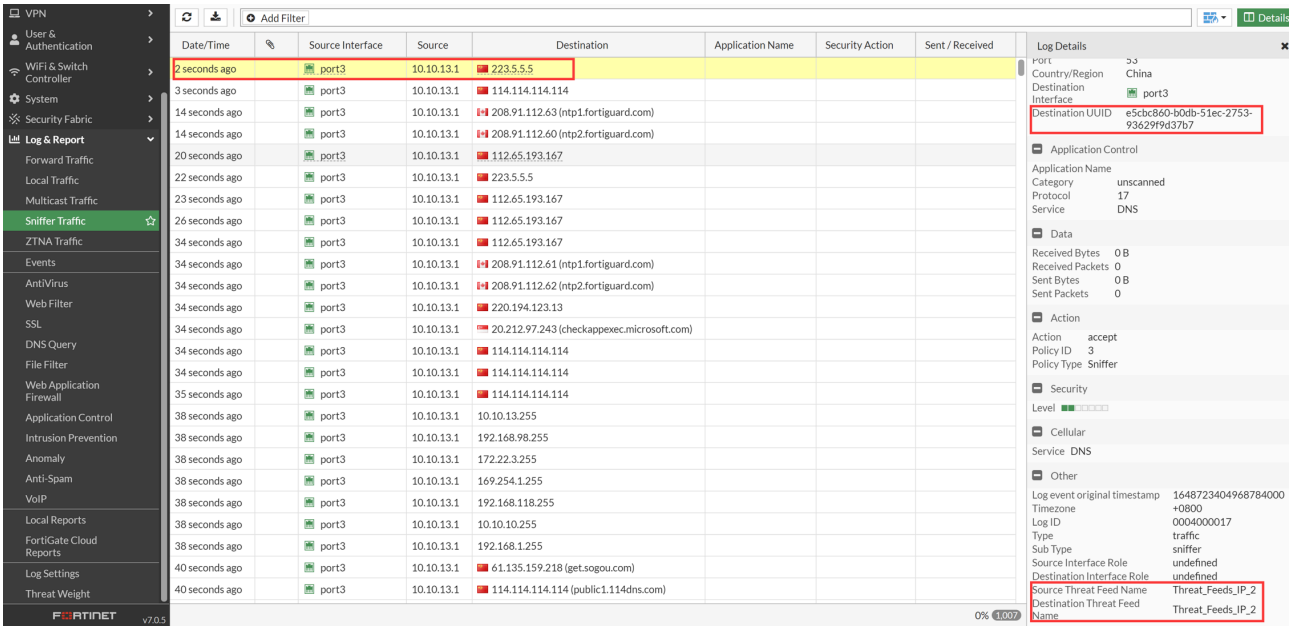
```
FGT # config firewall sniffer
FGT (sniffer) # edit 1
FGT (1) # set logtraffic utm                          //记录UTM日志
FGT (1) set interface "port1"                         //指定port1接口
FGT (1) set application-list-status enable            //开启应用控制监控
FGT (1) set application-list "default"                //根据应用控制规则"default"进行记录
FGT (1) set ips-sensor-status enable                  //开启IPS监控
FGT (1) set ips-sensor "default"                      //根据IPS规则"default"进行记录
FGT (1) set av-profile-status enable                  //开启防病毒监控
FGT (1) set av-profile "AV-flow"                      //根据防病毒规则"AV-flow"进行记录
FGT (1) set webfilter-profile-status enable           //开启web过滤监控
FGT (1) set webfilter-profile "flow-monitor-all "     //根据web过滤规则"default"进行记录
FGT (1) end
```

4. 查看配置。

```
FGT # config firewall sniffer
FGT (sniffer) # get
== [ 1 ]
FGT (sniffer) # edit 1
FGT (1) # get
id                   : 1
status               : enable
logtraffic           : utm
ipv6                 : disable
non-ip               : disable
interface            : port1
host                 :
port                 :
protocol             :
vlan                 :
application-list-status: enable
application-list     : default
ips-sensor-status    : enable
ips-sensor           : default
av-profile-status    : enable
av-profile           : default
webfilter-profile-status: enable
webfilter-profile    : flow-monitor-all
spamfilter-profile-status: disable
dlp-sensor-status    : disable
ips-dos-status       : disable
max-packet-count     : 4000
```

结果验证

进入日志 & 报表 → 嗅探流量，进行查看。



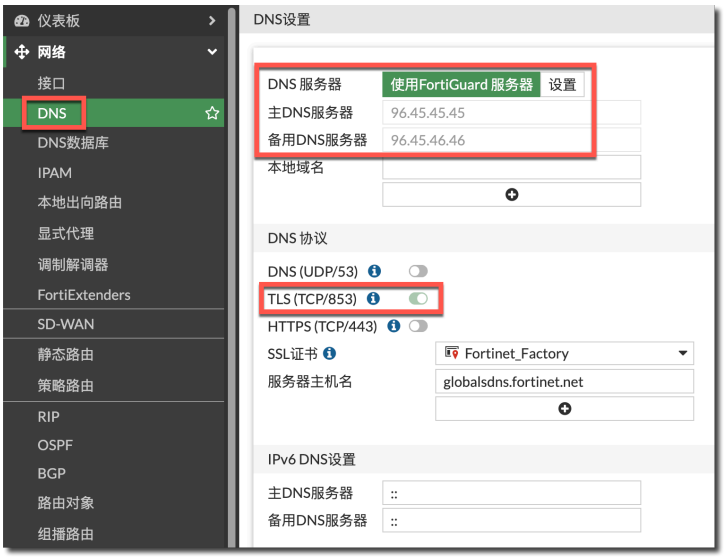
DNS

- DNS 配置
- 穿越接口的 DNS 服务
- 可选 DNS 服务器

DNS 配置

功能介绍

FortiGate 在需要将域名解析为 IP 地址（如被域名定义的 NTP 服务器、FortiGuard 更新服务器、FQDN 地址对象、FortiGate 作为 DNS 服务器、SD-WAN SLA 使用域名作为探测服务器等场景）时会查询已配置的 DNS 服务器。默认配置下，FortiGate 使用 FortiGuard 的 DoT（DNS over TLS）作为自身的 IPv4 DNS 服务器。

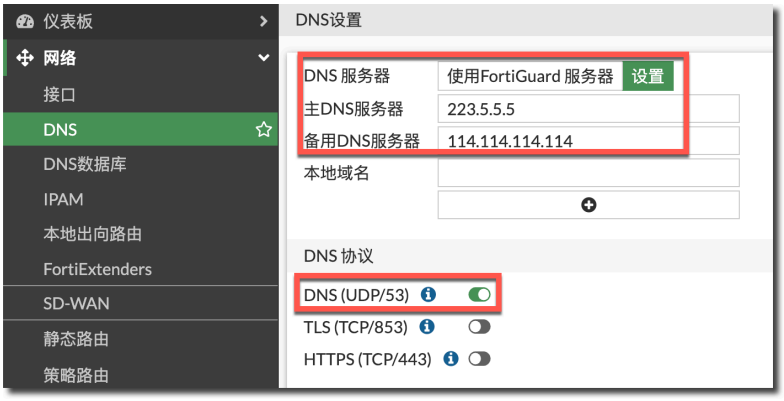



```bash
config system dns
    set primary 96.45.45.45
    set secondary 96.45.46.46
    set protocol dot
    set server-hostname "globalsdns.fortinet.net"
end
```

自定义 DNS 服务器

如果需要修改为自定义的 DNS 服务器，需要点击“设置”按钮，然后填入自定义的 DNS 服务器并选择 DNS 协议。

> **重要**
>
> 在取消使用 FortiGuard 服务器后，DNS 协议默认会选择“TLS (TCP/853)”，需要手动修改为自定义 DNS 服务器使用的 DNS 协议。




```bash
config system dns
    set primary 223.5.5.5
    set secondary 114.114.114.114
end
```

DNS 服务器主备模式

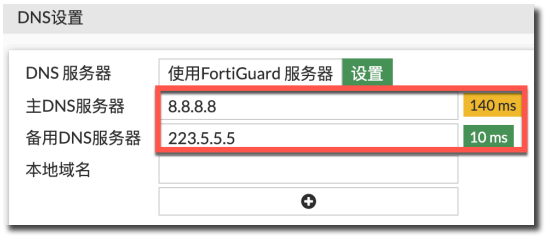
1. DNS 服务器在主备的选择上有两种模式：

   -**least-rtt（默认配置）**：根据最后一次往返时间（RTT）最短的作为当前 DNS 服务器。 -**failover**：根据服务器的配置顺序选择服务器，首先选择主 DNS 服务器进行解析，主 DNS 服务器无法访问或出现问题时裁切换到备用 DNS 服务器。

2. 该选项只能通过 CLI 进行更改。

```bash
config system dns
    set server-select-method {least-rtt | failover}
end
```

3. 通过 `diagnose test application dnsproxy 2` 命令可以看到两个服务器的延迟差距，如下所示，使用 `least-rtt` 模式时，FortiGate 会选择 223.5.5.5 作为 DNS 服务器，使用 `failover` 模式时，FortiGate 会选择 8.8.8.8 作为 DNS 服务器。



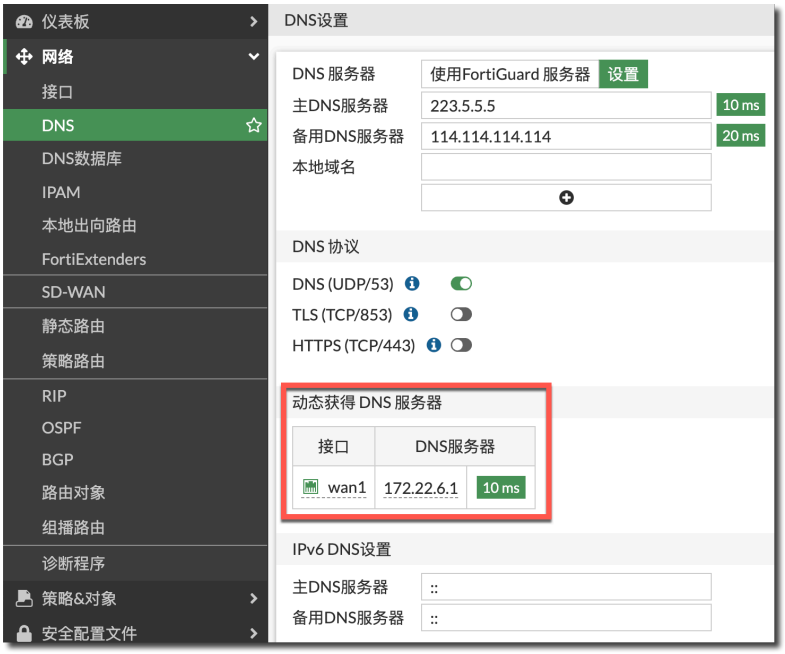

```bash
FortiGate # diagnose test application dnsproxy 2
......
DNS latency info:
vfid=0 server=8.8.8.8 latency=14 updated=126
vfid=0 server=223.5.5.5 latency=1 updated=41
......
```

4. 当其中一个 DNS 服务器无法访问时，会使用另一个 DNS 服务器进行解析，`diagnose test application dnsproxy 3` 命令可以看到服务器的探测失败的次数（`failure`）以及距离上次探测失败过去的时间（`last_failed`）。

```bash
FortiGate # diagnose test application dnsproxy 3
......
DNS servers:
8.8.8.8:53 vrf=0 tz=0 encrypt=none req=4 to=4 res=0 rt=0 ready=1 timer=0 probe=0 failure=5 last_failed=1215
223.5.5.5:53 vrf=0 tz=0 encrypt=none req=5 to=0 res=5 rt=1 ready=1 timer=0 probe=0 failure=0 last_failed=0
......
```

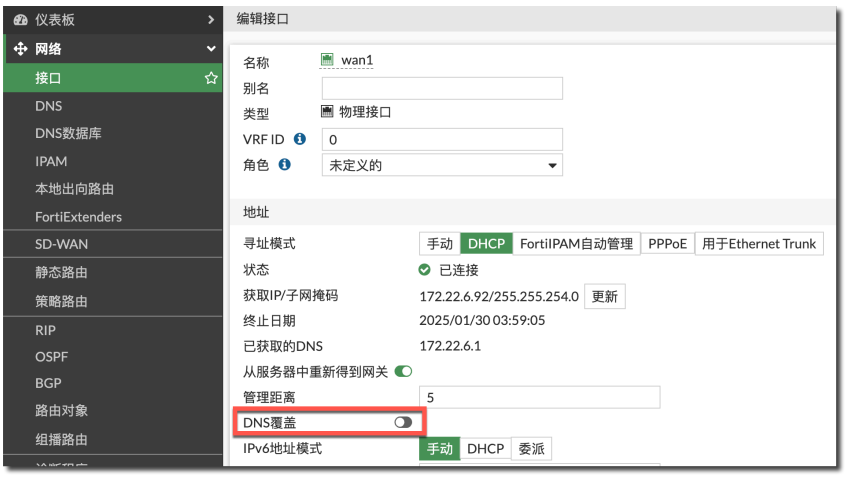
DNS 服务器覆盖

1. 要特别注意，如果配置了 PPPoE 或 DHCP，并从 ISP 侧获取了 IP 地址与 DNS 服务器。默认配置下，从 ISP 获取的 DNS 服务器会覆盖手动配置的 DNS 服务器。如下所示，wan1 接口 DHCP 获取的 DNS 服务器 172.22.6.1 覆盖了手工配置的 223.5.5.5/114.114.114.114。



2. 如果需要手工配置的 DNS 服务器生效，需要在相关 PPPoE/DHCP 接口下关闭“DNS 覆盖”功能。

> **重要**
>
> 如果使用的是将普通接口修改为 PPPoE 的模式，在接口上关闭“DNS 覆盖”功能，会导致接口重新拨号，请务必在窗口期进行操作。
>
> 所以强烈建议新建 PPPoE 接口来作为拨号接口，而不要使用普通接口上的 PPPoE 模式（参考：部署前注意事项 → PPPoE 接口配置 ↗。




```bash
config system interface
    edit "wan1"
        set dns-server-override disable
    next
end
```

VDOM DNS

当 FortiGate 处于多 VDOM 模式时，DNS 解析由管理 VDOM（默认为 root VDOM）处理。但在某些情况下，用户可能希望在非管理 VDOM 上配置自定义 DNS 设置。例如在多租户情况下，每个 VDOM 可能被不同的租户占用，每个租户可能需要自己的 DNS 服务器。

```bash
config vdom
    edit <vdom>
        config system vdom-dns
            set vdom-dns enable
            set primary <primary_DNS>
            set secondary <secondary_DNS>
            set protocol {cleartext dot doh}
```

```
            set ip6-primary <primary_IPv6_DNS>
            set ip6-secondary <secondary_IPv6_DNS>
            set source-ip <IP_address>
            set interface-select-method {auto | sdwan | specify}
        end
    next
end
```

DNS CLI 配置

interface-select-method

当有多个线路可以访问 DNS 服务器时，可以配置 DNS 请求的选路方式（默认为 `auto`），选择 `specify` 模式时，可以选择具体的接口。

```bash
config system dns
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
end
```

source-ip

设置 DNS 请求报文使用的源 IP 地址。

```bash
config system dns
    set source-ip <class_ip>
end
```

dns-cache-limit

FortiGate 会将 DNS 服务器返回的解析结果缓存起来，需要再次解析缓存中的相同域名时，FortiGate 会根据缓存条目快速返回结果，避免每次都需通过互联网重新获取相同的解析结果。此参数可以设置缓存中可以存储的最大 DNS 条目数（0~4294967295，默认值为 5000）。

```bash
config system dns
    set dns-cache-limit <integer>
end
```

1. 通过 `diagnose test application dnsproxy 7` 可以查看已有的 DNS 缓存条目。

```bash
FortiGate # diagnose test application dnsproxy 7
worker idx: 0
vfid=0, name=docs.fortinet.com, ttl=360:36:1476
         154.52.22.201 (ttl=360)
vfid=0, name=api.forticonverter.com, ttl=300:0:1455
         3.223.208.97 (ttl=300)
vfid=0, name=ntp2.fortiguard.com, ttl=5382:4934:1352
         208.91.112.62 (ttl=5382) 208.91.112.60 (ttl=5382)
vfid=0, name=ntp1.fortiguard.com, ttl=18545:17755:1010
         208.91.112.61 (ttl=18704) 208.91.112.63 (ttl=18704)
vfid=0, name=globalmsgctrl.fortinet.net, ttl=51:0:948
         38.21.192.4 (ttl=51)
vfid=0, name=globalfctupdate.fortinet.net, ttl=18608:17585:777
         173.243.143.6 (ttl=18608)
vfid=0, name=globalfctupdate.fortinet.net, ttl=14975:13947:772
         2620:101:9000:143:173:243:143:6 (ttl=14975)
vfid=0, name=globalupdate.fortinet.net, ttl=16397:15288:691
         173.243.129.6 (ttl=21498) 173.243.140.6 (ttl=21498)
vfid=0, name=globalupdate.fortinet.net, ttl=11075:9966:691
         2620:101:9000:140:173:243:140:6 (ttl=11075)
vfid=0, name=globaldevquery.fortinet.net, ttl=19376:18265:689
         173.243.140.16 (ttl=19376)
CACHE num=10
```

2. 通过 `diagnose test application dnsproxy 1` 可以清空 DNS 缓存条目。

dns-cache-ttl

DNS 缓存条目保留的时间（单位为 s，60~86400，默认值为 1800）。

```bash
config system dns
    set dns-cache-limit <integer>
end
```

fqdn-cache-ttl

> **重要**
>
> 如果设备开启了多 VDOM 模式，且 VDOM 下的 FQDN 没有自定义 `cache-ttl`，在全局下修改 `fqdn-cache-ttl`，也会覆盖 VDOM 下的 FQDN 的 `cache-ttl`（如果 VDOM 下的 FQDN 自定义了 `cache-ttl`，则还是以 VDOM 下的 FQDN 自定义 `cache-ttl` 为准）。

配置 FQDN 类型的地址对象后，FQDN 地址对象会用 2 种方式学习域名对应的 IP 地址：

- FortiGate 会主动定期去配置的 DNS 服务器查询 FQDN 对应的解析结果（适用于明确的 FQDN，如 `www.fortinet.com`，不适用于通配符 FQDN，如 `*.fortinet.com`）。
- 只要穿过防火墙的 DNS 请求与响应（A/AAAA 记录），匹配上了 FQDN 中的域名，FortiGate 就会记录该 IP 地址到 FQDN 地址对象中（同时适用于明确的 FQDN，如 `www.fortinet.com`，也适用于通配符 FQDN，如 `*.fortinet.com`）。

FQDN 地址对象会将解析结果缓存在 FortiGate 中。以下选项可以配置 FQDN 地址对象每个解析结果缓存的最大存活时间 （TTL，单位为 s，0~86400，默认值为 0，即使用 DNS 服务器返回的解析结果中的 TTL）。这是 FQDN 地址对象在没有主动刷新的情况下的最大 TTL。

```bash
config system dns
    set fqdn-cache-ttl 2000
end
```

1. 通过 `diagnose test application dnsproxy 6` 可以看到所有 FQDN 类型的地址对象已解析到的缓存条目，其中 `cache_ttl` 字段表示 FQDN 地址对象每个缓存 IP 条目的最大 TTL。

   ```bash
   FortiGate # diagnose test application dnsproxy 6
   worker idx: 0
   vfid=0 name=handbook.fortinet.com.cn ver=IPv4 wait_list=0 timer=590 min_ttl=600 cache_ttl=2000 slot=-1 num=1 wildcard=0
           39.105.115.226 (ttl=600:596:1996)
   vfid=0 name=www.baidu.com ver=IPv4 wait_list=0 timer=50 min_ttl=32 cache_ttl=2000 slot=-1 num=2 wildcard=0
           220.181.111.1 (ttl=32:28:1996) 220.181.111.232 (ttl=32:28:1996)
   ```

   > **重要**
   >
   > **TTL 的 3 个数字解释（如上所示的 ttl = 600:596:1996）：**
   >
   > 第 1 个数字（600）：DNS 服务器返回的解析结果中的最大 TTL。
   >
   > 第 2 个数字（596）：DNS 服务器返回的解析结果中的剩余 TTL。
   >
   > 第 3 个数字（1996）：DNS 缓存条目的剩余 TTL。

2. `fqdn-cache-ttl` 为全局设置。如果为某个 FQDN 地址单独配置了 `cache-ttl` 值，它将取代该地址的全局 `fqdn-cache-ttl` 设置。如下所示，单独配置 FQDN 地址对象 `www.baidu.com` 的 `cache-ttl` 为 800，`diagnose test application dnsproxy 6` 命令的结果中显示 `www.baidu.com` 的 `cache_ttl` 为 800，而 `handbook.fortinet.com.cn` 的 `cache_ttl` 仍为全局配置 `fqdn-cache-ttl` 中配置的 2000。

   ```bash
   config firewall address
       edit "www.baidu.com"
           set type fqdn
           set fqdn "www.baidu.com"
           set cache-ttl 800
       next
   end
   ```

```
FortiGate # diagnose test application dnsproxy 6                                                                    bash
worker idx: 0
vfid=0 name=handbook.fortinet.com.cn ver=IPv4 wait_list=0 timer=370 min_ttl=600 cache_ttl=2000 slot=-1 num=1 wildcard=0
        39.105.115.226 (ttl=600:376:1776)
vfid=0 name=www.baidu.com ver=IPv4 wait_list=0 timer=17 min_ttl=11 cache_ttl=800 slot=-1 num=4 wildcard=0
        103.235.47.188 (ttl=247:155:708) 103.235.46.96 (ttl=247:155:708) 220.181.111.232 (ttl=11:0:763) 220.181.111.1
(ttl=11:0:763)
```

3. 使用 `diagnose test application dnsproxy 4` 可以清空当前的 FQDN 缓存。

fqdn-min-refresh

FQDN 缓存最小刷新时间（单位 s，10~3600，默认为 60s）。FQDN 根据 DNS 服务器返回的所有该条目解析结果中最小的 TTL 间隔重新获取更新。默认配置下（`fqdn-min-refresh` =60，`fqdn-cache-ttl` =0），如果 DNS 服务器返回的解析结果中的最小 TTL 小于 60s，FortiGate 仍需要至少 60s 来重新查询 FQDN。如果 DNS 服务器返回的解析结果中的最小 TTL 大于 60s，FQDN 根据 DNS 服务器返回的所有该条目解析结果中最小的 TTL 间隔重新获取更新。刷新动作不会删除已有的 FQDN 缓存 IP 条目。

> **重要**
>
> 实际的 FQDN 刷新动作会在 `fqdn-min-refresh` 时间到期前 5s 执行。

1. 默认配置（`fqdn-min-refresh` =60，`fqdn-cache-ttl` =0）下，使用 `diagnose test application dnsproxy 3` 查看当前 `fqdn-min-refresh` 配置。

```
FortiGate # diagnose test application dnsproxy 3                                                                    bash
......
FQDN: min_refresh=60 max_refresh=3600
......
```

2. 执行 `diagnose test application dnsproxy 6` 命令，如下所示 `www.baidu.com` 的解析结果有 4 个，DNS 服务器返回的所有该条目解析结果中最小的 TTL（`min_ttl`）为 48，但重新查询域名的等待时间（`timer`）仍然按照 `fqdn-min-refresh` 的默认配置 60s 进行倒计时。

```
FortiGate # diagnose test application dnsproxy 6                                                                    bash
worker idx: 0
vfid=0 name=www.baidu.com ver=IPv4 wait_list=0 timer=56 min_ttl=48 cache_ttl=0 slot=-1 num=4 wildcard=0
        220.181.111.1 (ttl=69:54:54) 220.181.111.232 (ttl=69:54:54) 110.242.70.57 (ttl=48:47:47) 110.242.69.21
(ttl=48:47:47)

FQDN num=2
```

   - `min_ttl`：DNS 服务器返回的所有该条目解析结果中最小的 TTL。
   - `timer`：重新查询域名的等待时间计时器。

3. 将 `fqdn-min-refresh` 设置为 20，小于 `min_ttl`。

```
config system dns                                                                                                   bash
    set fqdn-min-refresh 20
end
```

4. 执行 `diagnose test application dnsproxy 6` 命令，如下所示 `www.baidu.com` 的解析结果有 2 个，DNS 服务器返回的所有该条目解析结果中最小的 TTL（`min_ttl`）为 54，重新查询域名的等待时间（`timer`）按照 `min_ttl` 的 54s 进行倒计时。

```
FortiGate # diagnose test application dnsproxy 6                                                                    bash
worker idx: 0
......
vfid=0 name=www.baidu.com ver=IPv4 wait_list=0 timer=52 min_ttl=54 cache_ttl=0 slot=-1 num=2 wildcard=0
        110.242.70.57 (ttl=54:52:52) 110.242.69.21 (ttl=54:52:52)

FQDN num=2
```

5. `fqdn-min-refresh` 可与 `fqdn-cache-ttl`、`cache-ttl` 结合使用，以发送更频繁的查询并在缓存中存储更多已解析的地址。这在 FQDN 有较多解析结果且变化非常频繁的情况下非常有用。

6. 总结如下：

   - `min_ttl` < `fqdn-min-refresh`：使用 `fqdn-min-refresh` 作为重新请求 FQDN 的计时器（`timer`）。
   - `min_ttl` > `fqdn-min-refresh`：使用 `min_ttl` 作为重新请求 FQDN 的计时器（`timer`）。
   - `fqdn-min-refresh/min_ttl` < `fqdn-cache-ttl/cache-ttl`：使用 `fqdn-min-refresh/min_ttl` 作为重新请求 FQDN 的计时器（`timer`）。

- `fqdn-min-refresh/min_ttl` > `fqdn-cache-ttl/cache-ttl`：使用 `fqdn-cache-ttl/cache-ttl` 作为重新请求 FQDN 的计时器（`timer`）。

fqdn-max-refresh

FQDN 缓存最大刷新时间（单位为 s，3600~86400，默认值为 3600）。`fqdn-max-refresh` 用于控制 FQDN 刷新计时器的全局上限。最小 TTL 大于 fqdn-max-refresh 值的 FQDN 条目，其刷新定时器将减少到该上限。最小 TTL 小于 fqdn-max-refresh 值的 FQDN 条目，其刷新定时器将取决于该条目所有解析结果中最小的 TTL 值。

`fqdn-max-refresh` 的测试结果与 `fqdn-min-refresh` 类似，这里就不再赘述。

dnsproxy-worker-count

对于有多个逻辑 CPU 核心的 FortiGate，可以设置用于运行 dnsproxy 进程的 CPU 核心数，默认配置为 1。

```bash
config system global
    set dnsproxy-worker-count <integer>
end
```

在有大量 FDQN 地址对象的情况下，编辑 DNS 相关配置时，dnsproxy 进程可能会消耗大量 CPU 资源，它会同时处理 DNS 相关配置更改和 DNS 解析任务。当 `dnsproxy-worker-count` 从 1 增加到 2 时：

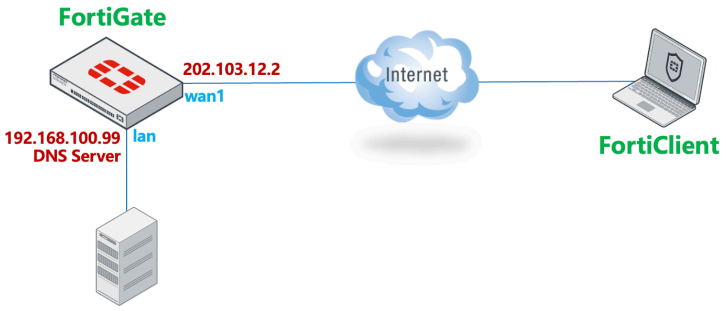
- dnsproxy-worker-1：处理 DNS 相关配置更改。
- dnsproxy-worker-2：处理 DNS 解析任务。

其他 CLI 配置

```bash
config system dns
    set primary <ip_address>
    set secondary <ip_address>
    set protocol {cleartext dot doh}
    set ssl-certificate <string>
    set server-hostname <hostname>
    set domain <domains>
    set ip6-primary <ip6_address>
    set ip6-secondary <ip6_address>
    set timeout <integer>
    set retry <integer>
    set dns-cache-limit <integer>
    set dns-cache-ttl <integer>
    set cache-notfound-responses {enable | disable}
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
    set source-ip <class_ip>
    set root-servers <DNS root name server IP address>
    set server-select-method {least-rtt | failover}
    set alt-primary <ip_address>
    set alt-secondary <ip_address>
    set log {disable |error | all}
    set fqdn-cache-ttl <integer>
    set fqdn-min-refresh <integer>
    set fqdn-max-refresh <integer>
end
```

穿越接口的 DNS 服务

网络需求



FortiGate 内网口（lan）开启 DNS 服务器，作为内部网络和 SSL VPN/IPSec VPN 客户端的 DNS 服务器。SSL VPN 客户端连接后，使用 FortiGate 内网口上提供的 DNS 服务进行域名解析。

1. 在 FortiGate 上的“主 DNS 服务器”和“备用 DNS 服务器”均为公网的 DNS 服务器，如下所示。

```bash
config system dns
    set primary 223.5.5.5
    set secondary 114.114.114.114
end
```

2. FortiGate 仅在内网口上启用 DNS 服务器（递归方式）。

```bash
config system interface
    edit "lan"
        set ip 192.168.100.99 255.255.255.0
    next
end

config system dns-server
    edit "lan"
        set mode recursive
    next
end
```

3. SSL VPN 配置分配的 DNS 服务器为 FortiGate 内网接口 IP 192.168.100.99。

```bash
config vpn ssl settings
    set servercert "Fortinet_Factory"
    set tunnel-ip-pools "sslvpn_pool"
    set dns-server1 192.168.100.99     ←——设置分配的DNS服务器为FortiGate内网口地址
    set source-interface "wan1"
    set default-portal "web-access"
    config authentication-rule
        edit 2
            set groups "VPN"
            set portal "full-access"
        next
    end
end
```

4. 防火墙策略、SSL VPN 门户、用户等配置略。

问题现象

1. FortiClient 客户端连接 FortiGate 的 SSL VPN 后，使用 FortiGate 内网口上提供的 DNS 服务器（192.168.100.99）无法进行域名解析。

```bash
::: important
nslookup www.baidu.com
:::
Server:     192.168.100.99
Address:    192.168.100.99#53
;; connection timed out; no servers could be reached
```

2. 在 FortiGate 上抓包查看 FortiClient（3.3.3.9）发送的 DNS 请求，只有从 `ssl.root` 接口进的流量，没有返回的流量。

```bash
FortiGate # diagnose sniffer packet any 'host 3.3.3.9 and port 53' 4
interfaces=[any]
filters=[host 3.3.3.9 and port 53]
0.614261 ssl.root in 3.3.3.9.65295 → 192.168.100.99.53: udp 47
0.614514 ssl.root in 3.3.3.9.53355 → 192.168.100.99.53: udp 47
1.660982 ssl.root in 3.3.3.9.65295 → 192.168.100.99.53: udp 47
1.661521 ssl.root in 3.3.3.9.53355 → 192.168.100.99.53: udp 47
1.793631 ssl.root in 3.3.3.9.56286 → 192.168.100.99.53: udp 31
```

3. 在 FortiGate 上查看对应的 dnsproxy debug 信息，提示解析结果为 `non-active`。

```bash
diagnose debug application dnsproxy -1
diagnose debug enable

[worker 0] udp_receive_request()-3194: vd=0, vrf=0, intf=39, len=31, alen=16, 3.3.3.9:59503⇒192.168.100.99
[worker 0] handle_dns_request()-2489: vfid=0 real_vfid=0 id=0×1aec pktlen=31 qr=0 req_type=3
[worker 0] dns_parse_message()-603
[worker 0] dns_nat64_ptr_lookup()-272
[worker 0] dns_nat64_ptr_lookup()-285: vdname:root request:256

[worker 0] dns_nat64_update_request()-305
[worker 0] dns_nat64_update_request()-319: vdname:root request:1

[worker 0] get_intf_policy()-1469: ifindex=39
[worker 0] dns_query_delete()-585: orig id:0×1aec local id:0×0000 domain=www.baidu.com use=5 non-active    ←——提示解析结果为non-active
```

解决方法

1. 对于这种穿越接口之间提供 DNS 服务的场景，DNS 服务器接口的配置除了要添加提供 DNS 服务的接口（此例中为内网口 `lan`），还要添加 DNS 请求流量的入接口（此例中为内网口 `ssl.root`）。

```bash
config system dns-server
    edit "lan"
        set mode recursive
    next
    edit "ssl.root"
        set mode recursive
    next
end
```

2. FortiClient 客户端再次使用 FortiGate 内网口上提供的 DNS 服务器（192.168.100.99）进行域名解析，可以正常获取解析结果。

```bash
::: important
nslookup www.baidu.com
:::
Server:     192.168.100.99
Address:    192.168.100.99#53

Non-authoritative answer:
Name:   www.baidu.com
Address: 110.242.70.57
Name:   www.baidu.com
Address: 110.242.69.21
```

3. 在 FortiGate 上查看对应的 dnsproxy debug 信息，显示 FortiGate 的内网口 `lan` 收到 FortiClient（3.3.3.9）的 DNS 请求后，递归到公网 DNS 服务器 114.114.114.114 进行查询，并将 DNS 解析结果返回给 FortiClient（3.3.3.9）。

```bash
diagnose debug application dnsproxy -1
diagnose debug enable

[worker 0] udp_receive_request()-3194: vd=0, vrf=0, intf=39, len=31, alen=16, 3.3.3.9:59143⇒192.168.100.99

[worker 0] get_intf_policy()-1469: ifindex=39
[worker 0] dns_local_lookup_common()-2553: vfid=0, real_vfid=0, view=2, qname=www.baidu.com, qtype=1, qclass=1,
offset=31, map#=3 max_sz=512
[worker 0] dns_lookup_aa_zone()-608: vfid=0, fqdn=www.baidu.com
[worker 0] dns_send_request()-1490
[worker 0] dns_send_resol_request()-1344: orig id: 0×95f3 local id: 0×95f3 domain=www.baidu.com
[worker 0] dns_find_best_server()-652: found server: 114.114.114.114 (vfid=0 vrf=0)
[worker 0] dns_udp_forward_request()-1089: vdom=root req_type=3 domain=www.baidu.com oif=0
[worker 0] dns_udp_forward_request()-1210: Send 31B to [114.114.114.114]:53 via fd=20 request:1
[worker 0] udp_receive_request()-3144
[worker 0] batch_on_read()-3555
[worker 0] _udp_receive_response()-3406: vd-0: len=63, addr=114.114.114.114:53, rating=0
[worker 0] dns_query_handle_response()-2743: vfid=0 real_vfid=0 vrf=0 id=0×95f3 domain=www.baidu.com pktlen=63
[worker 0] dns_query_save_response()-2724: domain=www.baidu.com pktlen=63
[worker 0] dns_set_min_ttl()-190: QR: www.baidu.com
[worker 0] dns_set_min_ttl()-197: Offset of 1st RR: 31 Number of RR's: 2
[worker 0] dns_set_min_ttl()-218: RR TTL: 78, RR type: 1
[worker 0] dns_set_min_ttl()-218: RR TTL: 78, RR type: 1
[worker 0] dns_cache_response()-329: Min ttl = 78
[worker 0] dns_forward_response()-1720
[worker 0] dns_secure_forward_response()-1676: category=255 profile=none
[worker 0] dns_visibility_log_hostname()-241: vd=0 pktlen=63
[worker 0] wildcard_fqdn_response_cb()-965: vd=0 pktlen=63
[worker 0] hostname_entry_insert()-143: af=2 domain=www.baidu.com
[worker 0] hostname_entry_insert()-143: af=2 domain=www.baidu.com
[worker 0] dns_send_response()-1645: domain=www.baidu.com reslen=63
[worker 0] __dns_udp_forward_response()-1522
[worker 0] __dns_udp_forward_response()-1533: vd-0 Send 63B via fd=7, family=2
[worker 0] __dns_udp_forward_response()-1537: set svf of fd to 0
[worker 0] __dns_udp_forward_response()-1587: vd=0 send 63B response 192.168.100.99:53⇒3.3.3.9:59143
[worker 0] dns_query_delete()-585: orig id:0×95f3 local id:0×95f3 domain=www.baidu.com use=7 active
[worker 0] dns_query_cleanup()-564: use=2
```

总结

1. 对于这种穿越接口之间提供 DNS 服务的场景，如 FortiGate 不同 VLAN 之间、不同物理口之间、VPN 接口到内网口的 DNS 访问，均需要在 DNS 服务器接口配置中增加 DNS 请求流量的入接口。
2. 从 FortiOS 7.6.1 开始，这种场景不再需要在 DNS 服务器上配置 DNS 请求的入接口。

可选 DNS 服务器

网络需求

FortiGate 配置 DNS 为公网的 DNS 服务器，同时 FortiGate 作为内网 PC 的 DNS 服务器。客户端 PC 无法通过 FortiGate 解析内网服务器的域名，只有内网的 DNS 服务器可以解析内网域名。需要 FortiGate 能够解析公共 DNS 服务器无法解析的内部域名。

1. 在 FortiGate 上的“主 DNS 服务器”和“备用 DNS 服务器”均为公网的 DNS 服务器，如下所示。

```bash
config system dns
    set primary 223.5.5.5
    set secondary 114.114.114.114
end
```

2. FortiGate 作为内部网络的 DNS 服务器（递归方式），未配置 DNS 数据库。内网 PC 的 DNS 服务器均设定为 FortiGate 的内网口 IP 192.168.100.99，通过 FortiGate 来解析域名。

```bash
config system interface
    edit "lan"
        set ip 192.168.100.99 255.255.255.0
    next
end

config system dns-server
    edit "lan"
    next
end
```

3. 在此配置下，客户端 PC 访问域名时，会向 FortiGate 的内网口 192.168.100.99 发起 DNS 解析请求，默认配置（`set server-select-method`）下，FortiGate 会选择延迟最小的服务器转发 DNS 请求，此例中 223.5.5.5 的延迟更低。

4. 客户端 PC 有访问内网服务器域名的需求，如 `www.fortidocs.com`，这种域名通过公网的 DNS 服务器是无法解析的，会直接返回 DNS NXDOMAIN，导致客户端 PC 无法通过域名访问内网服务器。

```bash
::: important
nslookup www.fortidocs.com 192.168.100.99
:::
Server:     192.168.100.99
Address:    192.168.100.99#53

** server can't find www.fortidocs.com: NXDOMAIN
```

5. 只有内网的 DNS 服务器可以解析这些内网服务器的域名，需要 FortiGate 在解析域名失败后，将解析请求转发至内网 DNS 服务器。

6. 内网 DNS 服务器中的 DNS 条目较多或没有权限访问，无法通过在 FortiGate 上配置 DNS Database 来解决此问题。

解决方法

1. 配置可选 DNS（Alt-DNS）服务器。当 FortiGate 的主 DNS 服务器/备用 DNS 服务器返回 DNS 的名称解析错误（NXDOMAIN）时，FortiGate 会查询配置的 Alt-DNS 服务器。可选 DNS 服务器的常见用途是解析公共 DNS 服务器无法解析的内部域名。

2. 配置方法（只能通过 CLI 配置），与 DNS 服务器一样，可以配置主 Alt-DNS 和备用 Alt-DNS，默认配置（`set server-select-method`）下，FortiGate 会选择延迟最小的 Alt-DNS 服务器转发 DNS 请求：

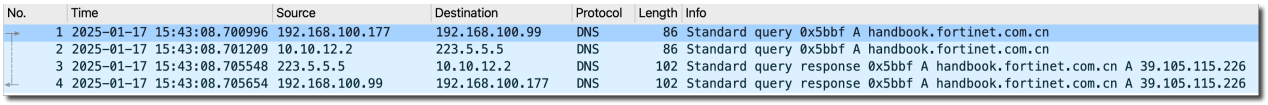
> **重要**
>
> 使用条件：
>
> - FortiGate 的主 DNS 服务器/备用 DNS 服务器返回 DNS 的错误类型为 NXDOMAIN 时，FortiGate 才会查询配置的 Alt-DNS 服务器。
> - 没有在 DNS Database 中配置对应内网域名的条目。

```bash
config system dns
    set alt-primary {ipv4-address}
    set alt-secondary {ipv4-address}
end
```

3. 在以上场景的配置基础上，增加 Alt-DNS 服务器的配置 10.10.12.1 和 10.10.12.3。

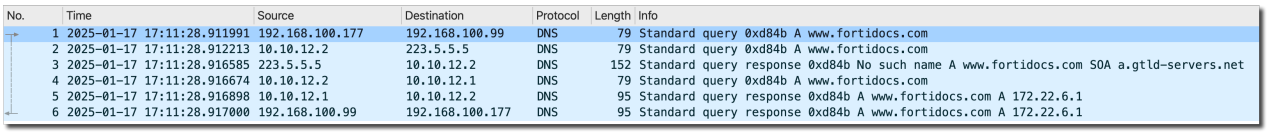
```bash
config system dns
    set primary 223.5.5.5
    set secondary 114.114.114.114
    set alt-primary 10.10.12.1
    set alt-secondary 10.10.12.3
end
```

4. 使用 PC 客户端指定 FortiGate 内网口来解析公网 DNS 服务器可以解析的域名（这里以 `handbook.fortinet.com.cn` 为例），FortiGate 将 DNS 请求递归到主 DNS 服务器查询，主 DNS 服务器返回解析结果（A 记录），FortiGate 将解析结果以自身内网口为源 IP 返回给 PC 客户端。



No.	Time	Source	Destination	Protocol	Length	Info
1	2025-01-17 15:43:08.700996	192.168.100.177	192.168.100.99	DNS	86	Standard query 0x5bbf A handbook.fortinet.com.cn
2	2025-01-17 15:43:08.701209	10.10.12.2	223.5.5.5	DNS	86	Standard query 0x5bbf A handbook.fortinet.com.cn
3	2025-01-17 15:43:08.705548	223.5.5.5	10.10.12.2	DNS	102	Standard query response 0x5bbf A handbook.fortinet.com.cn A 39.105.115.226
4	2025-01-17 15:43:08.705654	192.168.100.99	192.168.100.177	DNS	102	Standard query response 0x5bbf A handbook.fortinet.com.cn A 39.105.115.226

5. 使用 PC 客户端指定 FortiGate 内网口来解析公网 DNS 服务器无法解析的域名（这里以 `www.fortidocs.com` 为例），FortiGate 将 DNS 请求递归到主 DNS 服务器查询，主 DNS 服务器返回无法解析的结果（NXDOMAIN），FortiGate 不会将此 DNS NXDOMAIN 返回给客户端 PC，随后使用位于内网的主 Alt-DNS 服务器（10.10.12.1）进行查询，主 Alt-DNS 服务器（10.10.12.1）返回解析结果（A 记录）给 FortiGate，FortiGate 将解析结果返回给 PC 客户端。



No.	Time	Source	Destination	Protocol	Length	Info
1	2025-01-17 17:11:28.911991	192.168.100.177	192.168.100.99	DNS	79	Standard query 0xd84b A www.fortidocs.com
2	2025-01-17 17:11:28.912213	10.10.12.2	223.5.5.5	DNS	79	Standard query 0xd84b A www.fortidocs.com
3	2025-01-17 17:11:28.916585	223.5.5.5	10.10.12.2	DNS	152	Standard query response 0xd84b No such name A www.fortidocs.com SOA a.gtld-servers.net
4	2025-01-17 17:11:28.916674	10.10.12.2	10.10.12.1	DNS	79	Standard query 0xd84b A www.fortidocs.com
5	2025-01-17 17:11:28.916898	10.10.12.1	10.10.12.2	DNS	95	Standard query response 0xd84b A www.fortidocs.com A 172.22.6.1
6	2025-01-17 17:11:28.917000	192.168.100.99	192.168.100.177	DNS	95	Standard query response 0xd84b A www.fortidocs.com A 172.22.6.1

```bash
::: important
nslookup www.fortidocs.com 192.168.100.99
:::
Server:     192.168.100.99
Address:    192.168.100.99#53

Non-authoritative answer:
Name:   www.fortidocs.com
Address: 172.22.6.1
```

6. 查看 Alt-DNS 整个过程的 dnsproxy debug 信息。

```bash
diagnose debug application dnsproxy -1
diagnose debug enable

[worker 0] udp_receive_request()-3144
[worker 0] udp_receive_request()-3194: vd=0, vrf=0, intf=46, len=35, alen=16, 192.168.100.177:64454⇒192.168.100.99
[worker 0] handle_dns_request()-2489: vfid=0 real_vfid=0 id=0×e5ae pktlen=35 qr=0 req_type=3
[worker 0] get_intf_policy()-1469: ifindex=46
[worker 0] dns_local_lookup_common()-2553: vfid=0, real_vfid=0, view=2, qname=www.fortidocs.com, qtype=1, qclass=1,
offset=35, map#=3 max_sz=512
[worker 0] dns_lookup_aa_zone()-608: vfid=0, fqdn=www.fortidocs.com
[worker 0] dns_send_request()-1490
[worker 0] dns_send_resol_request()-1344: orig id: 0×e5ae local id: 0×e5ae domain=www.fortidocs.com
[worker 0] dns_find_best_server()-652: found server: 223.5.5.5 (vfid=0 vrf=0)
[worker 0] dns_udp_forward_request()-1089: vdom=root req_type=3 domain=www.fortidocs.com oif=0
[worker 0] dns_udp_forward_request()-1210: Send 35B to [223.5.5.5]:53 via fd=20 request:1
[worker 0] udp_receive_request()-3144
[worker 0] batch_on_read()-3555
[worker 0] _udp_receive_response()-3406: vd-0: len=108, addr=223.5.5.5:53, rating=0
[worker 0] dns_query_handle_response()-2743: vfid=0 real_vfid=0 vrf=0 id=0×e5ae domain=www.fortidocs.com pktlen=108
[worker 0] dns_retransmit_func()-1763: jiffies=183874965 created=183874965 wait_cat=0 wait_res=1 profile=
last_tx=183874964 ftg_last_tx=0 domain=www.fortidocs.com (orig id: 0×e5ae local id:0×e5ae active)
[worker 0] dns_send_request()-1490
[worker 0] dns_send_resol_request()-1344: orig id: 0×e5ae local id: 0×e5ae domain=www.fortidocs.com
[worker 0] dns_find_best_server()-652: found server: 10.10.12.1 (vfid=0 vrf=0)
[worker 0] dns_udp_forward_request()-1089: vdom=root req_type=3 domain=www.fortidocs.com oif=0
[worker 0] dns_udp_forward_request()-1210: Send 35B to [10.10.12.1]:53 via fd=20 request:1
[worker 0] batch_on_read()-3555
[worker 0] _udp_receive_response()-3406: vd-0: len=51, addr=10.10.12.1:53, rating=0
[worker 0] dns_query_handle_response()-2743: vfid=0 real_vfid=0 vrf=0 id=0×e5ae domain=www.fortidocs.com pktlen=51
[worker 0] dns_query_save_response()-2724: domain=www.fortidocs.com pktlen=51
[worker 0] dns_set_min_ttl()-190: QR: www.fortidocs.com
[worker 0] dns_set_min_ttl()-197: Offset of 1st RR: 35 Number of RR's: 1
```

```
[worker 0] dns_set_min_ttl()-218: RR TTL: 86400, RR type: 1
[worker 0] dns_cache_response()-329: Min ttl = 86400
[worker 0] dns_forward_response()-1720
[worker 0] dns_secure_forward_response()-1676: category=255 profile=none
[worker 0] dns_visibility_log_hostname()-241: vd=0 pktlen=51
[worker 0] wildcard_fqdn_response_cb()-965: vd=0 pktlen=51
[worker 0] hostname_entry_insert()-143: af=2 domain=www.fortidocs.com
[worker 0] dns_send_response()-1645: domain=www.fortidocs.com reslen=51
[worker 0] __dns_udp_forward_response()-1522
[worker 0] __dns_udp_forward_response()-1533: vd-0 Send 51B via fd=7, family=2
[worker 0] __dns_udp_forward_response()-1537: set svf of fd to 0
[worker 0] __dns_udp_forward_response()-1587: vd=0 send 51B response 192.168.100.99:53⇒192.168.100.177:64454
[worker 0] dns_query_delete()-585: orig id:0×e5ae local id:0×e5ae domain=www.fortidocs.com use=7 active
```

DHCP 配置

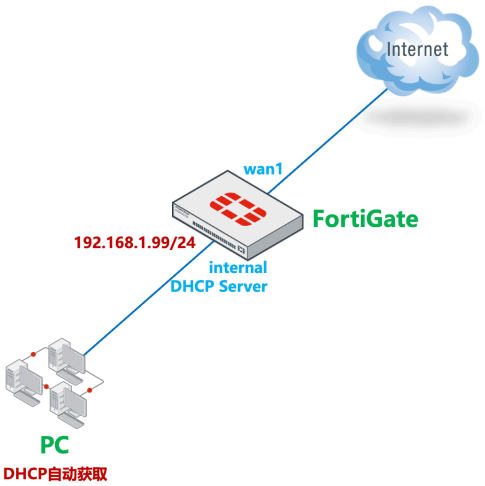
- DHCP 服务器配置
- DHCP 静态绑定
- DHCP 中继配置

DHCP 服务器配置

网络需求

开启 FortiGate 的 DHCP 服务器功能，内网电脑可通过自动获取地址的方式上网。内网用户网段为 192.168.1.0/24，网关为 192.168.1.99。

网络拓扑



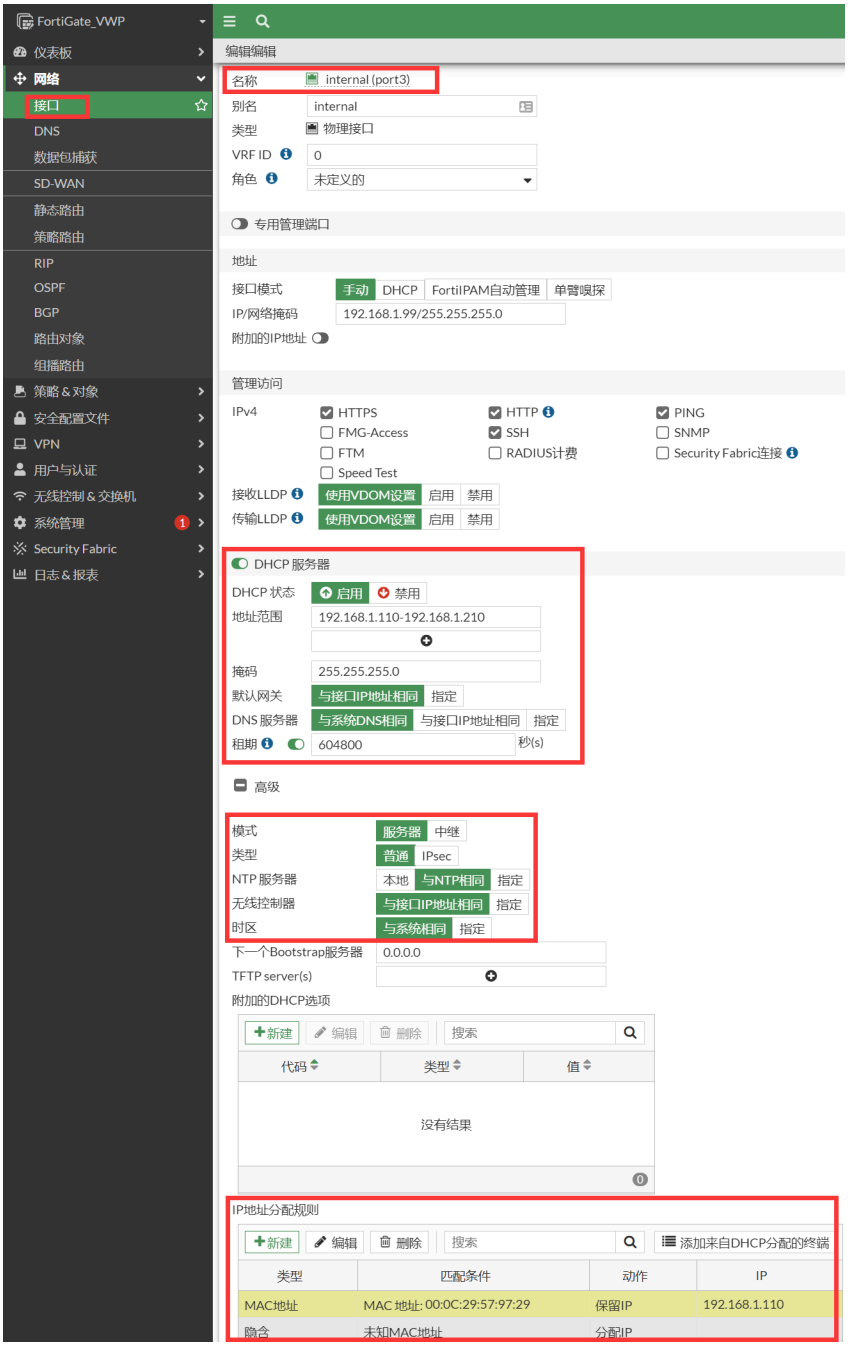
配置要点

- 基本上网配置
- 配置 DHCP 服务器

配置步骤

1. 基本上网配置，请参考策略与对象 → 单线路上网配置 → 静态地址线路上网配置 ↗章节。

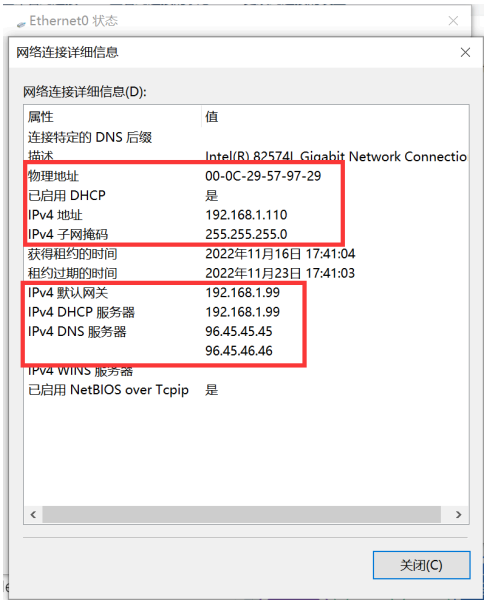
2. 开启 DHCP 服务，进入网络 → 接口，选择需要开启 DHCP 服务的接口，启用 DHCP 服务器。



- 地址范围：用于为用户分配的 IP 地址范围
- 掩码：分配的地址池地址掩码
- 默认网关：一般为 DHCP 所在接口的 ip 地址，也可以手工指定
- DNS 服务器：直接使用系统的 DNS 配置，也可以手工指定
- 模式：选择“DHCP 服务器模式”
- IP 地址分配规则：可指定固定的 MAC 地址分配固定的 IP 地址，也可以针对固定的 MAC 地址进行阻断过滤的策略控制（不需要控制 IP 分配到具体的主机时不需要配置）
- 类型：普通（或 IPsec，用于为 IPsec 用户分配 IP 地址）

结果验证

将电脑配置为自动获取 IP 地址模式，会获取配置的 MAC 地址对应的 IP 地址。



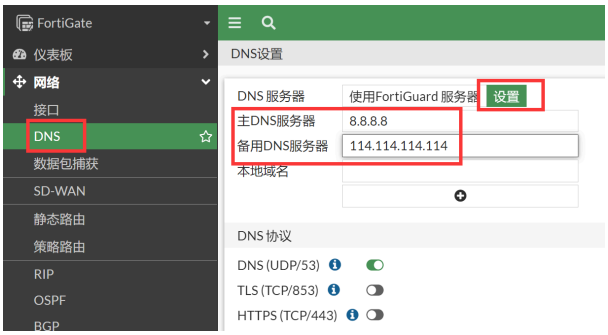
注意事项

1. DHCP 配置中，利用系统 DNS，为防火墙自身系统的 DNS 配置。当执行命令 set dns-service default 时，PC 终端获取到防火墙自身系统配置的 DNS。

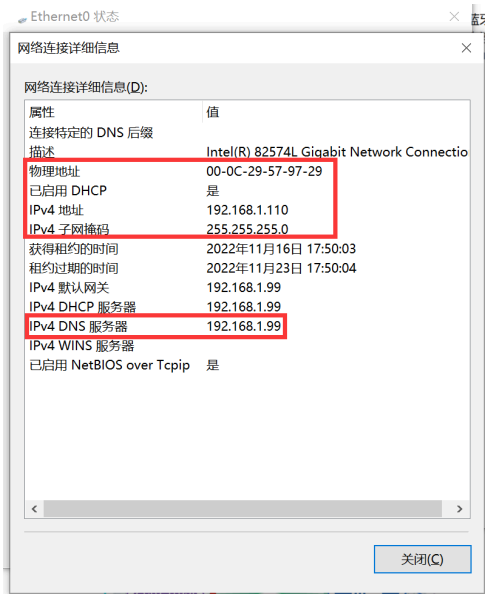
```
FortiGate# config system dhcp server
FortiGate(server) #edit 1
FortiGate(1)#set dns-service default     //默认参数
default    Use system DNS settings.      //防火墙上配置的系统DNS服务器,
local      Use this RGT as DNS server.   //防火墙接口的IP地址,
specify    Specify DNS servers.          //指定DNS
```

2. 可以自定义设置防火墙自身系统 DNS。

```
FortiGate#config system dns      //防火墙上配置的dns
FortiGate(dns) #set primary 8.8.8.8
FortiGate(dns) #end
```



3. 当 FortiGate 执行命令 set dns-service local 时，PC 终端获取到防火墙启用 DHCP 接口的 IP 日志。

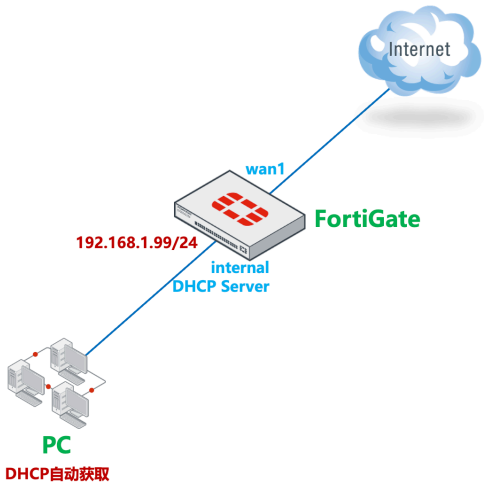

Ethernet0 状态
网络连接详细信息
网络连接详细信息(D):
属性
值
连接特定的 DNS 后缀
描述
Intel(R) 82574L Gigabit Network Connectio
物理地址
00-0C-29-57-97-29
已启用 DHCP
是
IPv4 地址
192.168.1.110
IPv4 子网掩码
255.255.255.0
获得租约的时间
2022年11月16日 17:50:03
租约过期的时间
2022年11月23日 17:50:04
IPv4 默认网关
192.168.1.99
IPv4 DHCP 服务器
192.168.1.99
IPv4 DNS 服务器
192.168.1.99
IPv4 WINS 服务器
已启用 NetBIOS over Tcpip
是
关闭(C)

DHCP 静态绑定

网络需求

开启 FortiGate 的 DHCP 服务器功能，内网电脑可通过自动获取地址的方式上网。内网用户网段为 192.168.1.0/24，网关为 192.168.1.99。为 MAC 地址 00:0c:29:57:97:29 的主机保留 ip 地址 192.168.1.110。

网络拓扑



配置要点

- 基本上网配置
- 配置 DHCP 服务器

配置步骤

1. 基本上网配置，请参考策略与对象 → 单线路上网配置 → 静态地址线路上网配置 ↗章节。

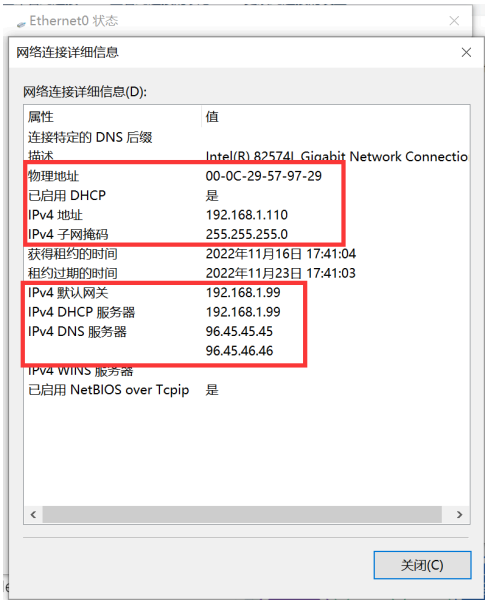
2. 配置 DHCP 服务，请参考网络管理 → DHCP 配置 → DHCP 服务器配置 ↗章节。

3. 配置保留 IP，也可参考网络管理 → DHCP 配置 → DHCP 服务器配置 ↗中的 GUI 方式配置。

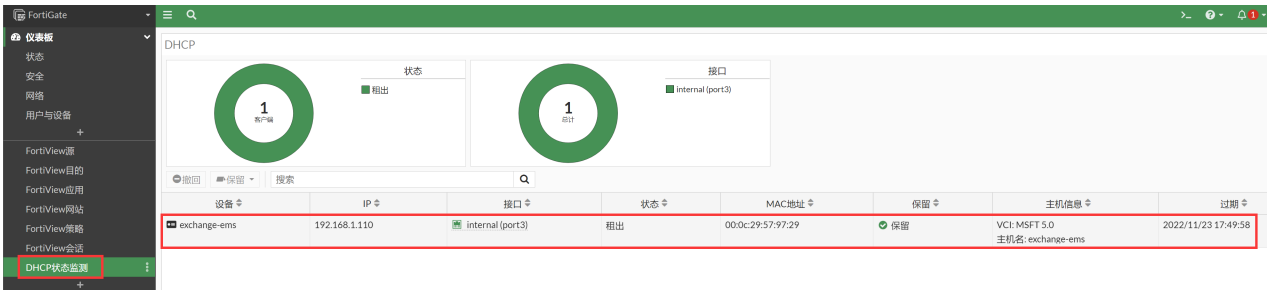
```
FortiGate# config system dhcp server
FortiGate(server) # edit 1                          //DHCP基础配置
FortiGate(1)#set dns-service default
FortiGate(1)#set default-gateway 192.168.1.99
FortiGate(1)#set netmask 255.255.255.0
FortiGate(1)#set interface internal
FortiGate(1) # config ip-range
FortiGate(ip-range) #edit 1
FortiGate(1)set start-ip 192.168.1.110
FortiGate(1)set end-ip 192.168.1.210
FortiGate(1) # next
FortiGate(ip-range) # end                            //DHCP基础配置
FortiGate(1)#config reserved-address                 //配置保留ＩＰ
FortiGate(reserved-address)#edit 1                  //条目：1，2，3，仅做为标识作用，可以写多个条目。
FortiGate(1) # set ip 192.168.1.110                  //为指定MAC地址分配该IP地址
FortiGate(1) # set mac 00:0c:29:57:97:29             //指定MAC地址
FortiGate(1) # next
FortiGate(reserved-address) # end
FortiGate(1) # next
FortiGate(server) #end
```

结果验证

1. 将电脑配置为自动获取 IP 地址模式，MAC 地址为 00:0c:29:57:97:29 的主机会获取 192.168.1.110。



2. 查看防火墙上 dhcp 地址池分配情况。

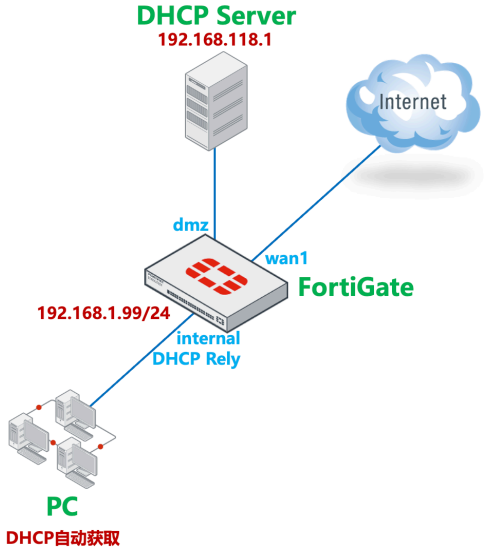


DHCP 中继配置

网络需求

在 FortiGate 上开启 DHCP 中继，让内网电脑能获取来自 DHCP 服务器分配的地址。

网络拓扑

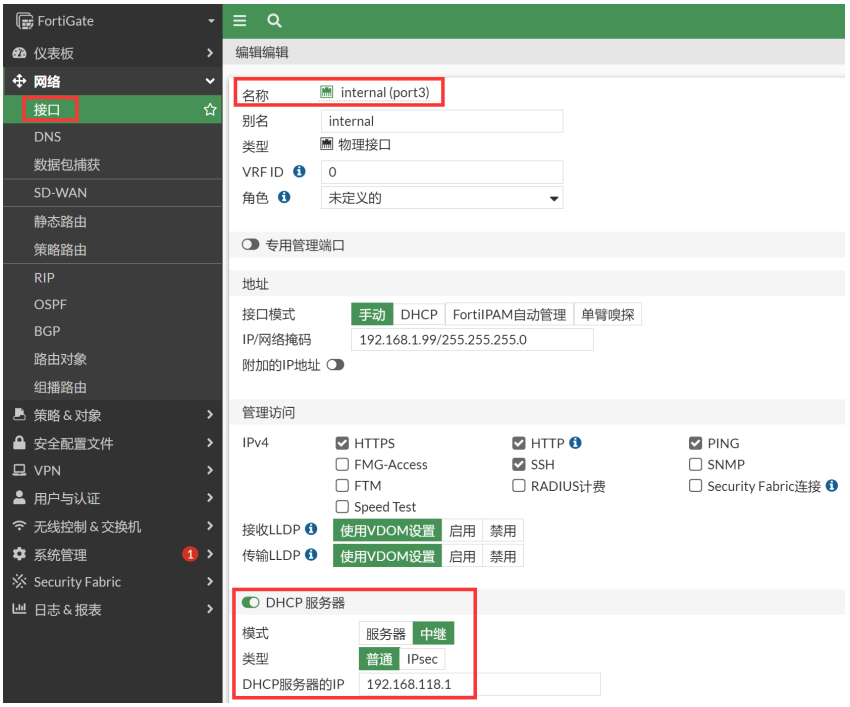


配置要点

- 基础上网配置
- 开启 DHCP 中继功能，并填写 DHCP 服务器地址

配置步骤

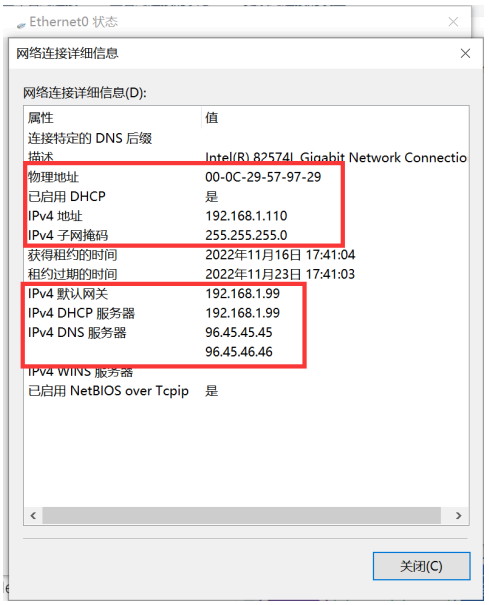
1. 基本上网配置，请参考策略与对象 → 单线路上网配置 → 静态地址线路上网配置 ↗章节。
2. 开启 DHCP 中继功能，并填写 DHCP 服务器地址。进入网络 → 接口，编辑需要开启 DHCP 服务的接口，开启 DHCP 服务器，打开高级选项，模式选择中继。



- DHCP 服务器：开启
- DHCP 服务器的 IP：填写真实 DHCP 服务器的 IP 地址，需要防火墙进行中继的 DHCP 服务器地址
- Mode：中继
- 类型： 普通（或 IPsec，用于为 IPsec 用户分配 IP 地址）

结果验证

将电脑配置为自动获取 IP 地址模式，会获取相应的 IP 地址。



交换接口

- 软交换配置
- VLAN/Hardware Switch
- 交换接口的 MAC 地址表

软交换配置

功能说明

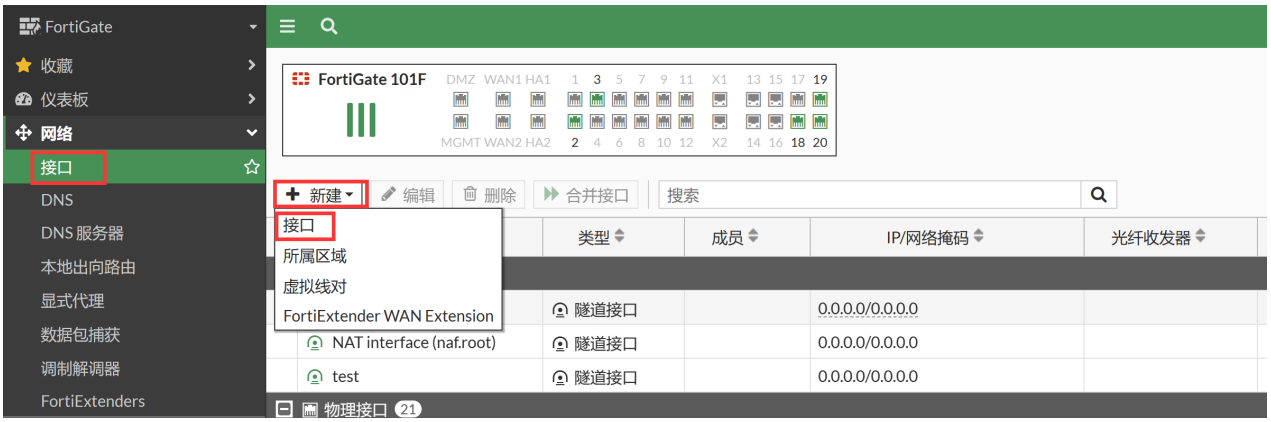
软交换口是将防火墙的多个 3 层接口，通过软件的方式，组成一个 2 层交换接口。 当 FortiGate 的每个口都为 3 层路由接口时，可以将其中的接口组成软件交换接口。

VLAN/Hardware 与 Software Switch 的区别

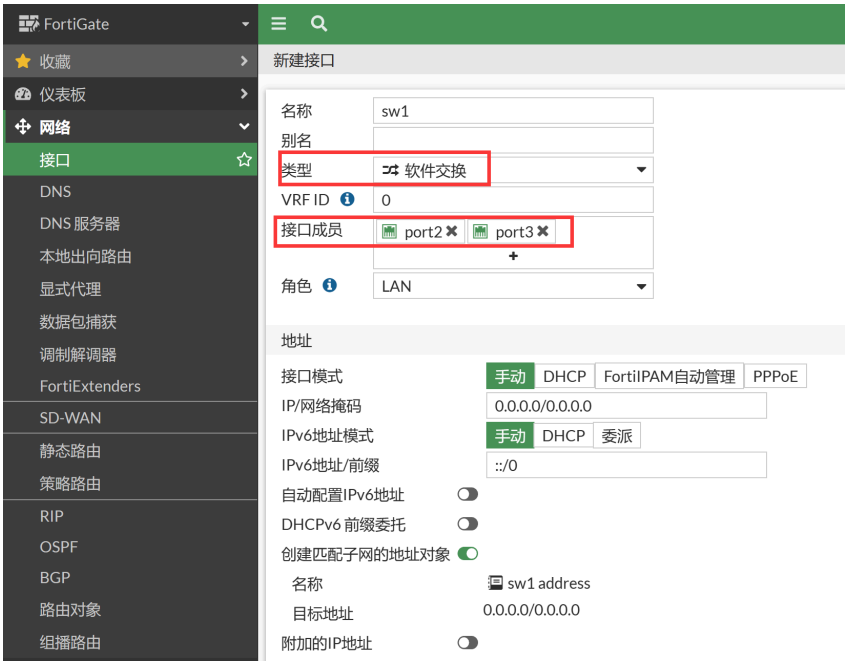
功能	VLAN/Hardware switch	Software switch
转发	数据包由设备硬件交换机或 SPU 在硬件层面处理	数据包由 CPU 处理
STP	支持	不支持
无线 SSIDs	不支持	支持
交换接口互访	默认允许	默认允许，可以通过策略控制

功能配置

1. 添加聚合接口，进入网络 → 接口，新建接口。



2. 类型选择软件交换，选择物理接口成员。



重要

建议在不需要的情况下关闭接口配置下的“设备探测”（Device detection）功能，该功能用于 MAC 地址厂商设备信息识别及 MAC 地址过滤，会消耗较多的设备资源，可能导致流量无法被芯片加速。

3. 接口配置页面查看建立的软交换口。



4. 在命令行查看软交换接口的配置。

```
FortiGate # show full-configuration system switch-interface sw1
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port2" "port3"
        set type switch
        set intra-switch-policy implicit
        set mac-ttl 300
        set span disable
    next
end
```

注意事项

1. 软交换口是通过软件的方式模拟出的，需要由 CPU 处理，无法被芯片加速，会影响系统的性能，谨慎使用。（部分型号的设备自带硬交换接口，支持硬件加速，如 100F）。

2. 默认配置下，交换机内部的成员接口之间通信是放通的，可以通过如下命令修改为必须匹配防火墙策略。

```
FortiGate # config sys switch-interface
FortiGate (switch-interface) # edit sw1
FortiGate (test) # set intra-switch-policy ?
implicit    Traffic between switch members is implicitly allowed.    //默认配置，成员之间的流量全部放通
explicit    Traffic between switch members must match firewall policies.    //根据配置的防火墙策略执行放通或拒绝动作
```

VLAN/Hardware Switch

功能介绍

VLAN/Hardware Switch 是一个虚拟交换机接口，它将不同的物理接口组合在一起，以便 FortiGate 可以将这些成员接口组合成单个接口。部分支持此接口的 FortiGate 型号有一个默认的硬件交换接口，称为 internal 或 lan，VLAN/Hardware Switch 可以被硬件级别的芯片支持。

连接到同一 VLAN/Hardware Switch 的接口，类似于位于同一广播域中的同一物理交换机上一样。接口成员可以从 VLAN/Hardware Switch 中移除并分配给另一个交换机或用作独立物理接口。

VLAN/Hardware 与 Software Switch 的区别

功能	VLAN/Hardware switch	Software switch
转发	数据包由设备硬件交换机或 SPU 在硬件层面处理	数据包由 CPU 处理
STP	支持	不支持
无线 SSIDs	不支持	支持
交换接口互访	默认允许	默认允许，可以通过策略控制

VLAN Switch 与 Hardware Switch 的区别

1. 60F、80F、100F、200F 等具有硬件交换模块的型号，通过内部硬件交换机创建的交换接口为 VLAN Switch。这些型号的设备一般在默认配置下会存在一个名称为 lan 或 Internal 的 VLAN Switch 接口，交换接口成员默认包含硬件交换模块中的成员接口，这些接口可以从预置的交换接口中取出，变为普通物理口。

   > **重要**
   >
   > 具体型号与硬件结构请参考：
   >
   > https://docs.fortinet.com/document/fortigate/7.99.99/hardware-acceleration/366011/fortigate-np6xlite-architectures ↗
   >
   > https://docs.fortinet.com/document/fortigate/7.99.99/hardware-acceleration/112876/fortigate-np6-architectures ↗

2. 40F、300E、400E、1100E、2200E、3600E、3980E、400F、600F、1800F、2600F、3000F、4400F 等具有硬件交换模块的型号，通过内部硬件交换机创建的交换接口为 Hardware Switch。

   > **重要**
   >
   > 具体型号与硬件结构请参考：
   >
   > https://docs.fortinet.com/document/fortigate/7.99.99/hardware-acceleration/46115/fortigate-np7-architectures ↗

3. VLAN Switch 接口可以直接在接口下配置 VLAN ID，同时配置 IP 后，可以直接将 VLAN Switch 配置为 VLAN Interface（携带 VLAN Tag），实现接口成员同属于该 VLAN Interface 的效果，而无需在 VLAN Switch 接口上再配置 VLAN 接口，而 Hardware Switch 无法配置 VLAN ID，想实现上述效果，必须在 Hardware Switch 上配置 VLAN Interface。

4. 除此之外，VLAN Switch 与 Hardware Switch 这两种交换接口在功能使用上没有差别。

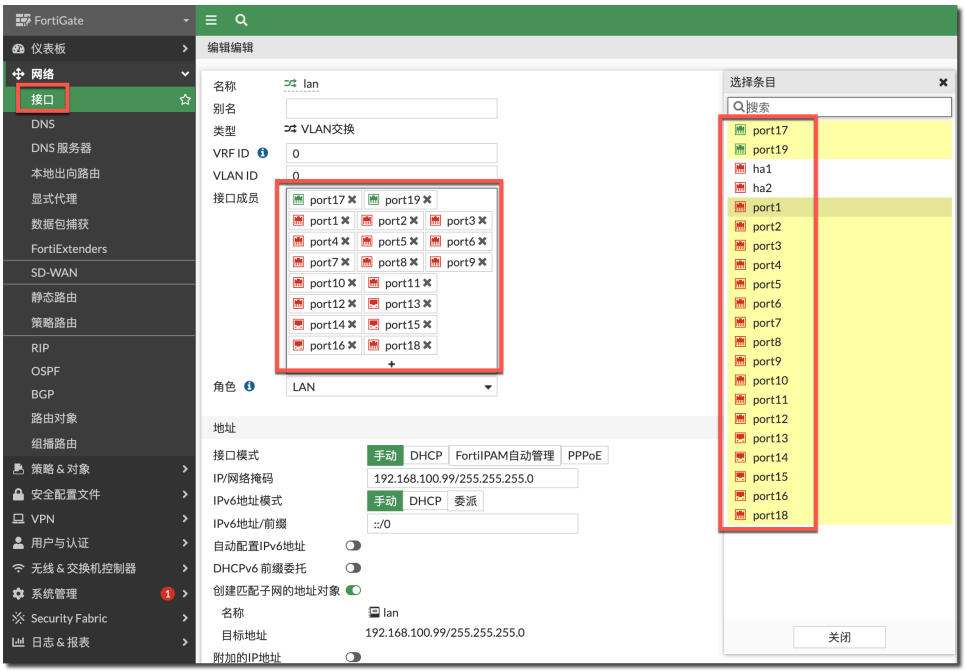
配置 VLAN/Hardware Switch

创建/编辑接口

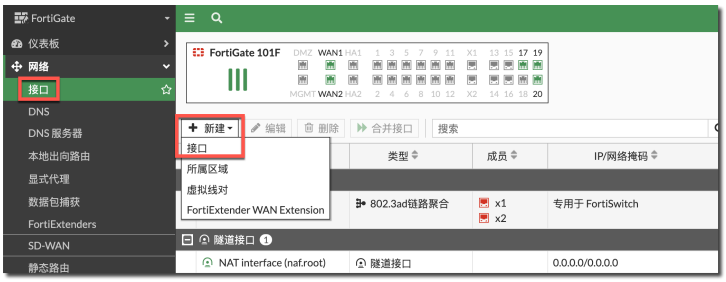
1. 在 SoC 平台（如 40F、60F、80F、100F、200F 等）默认配置下会存在一个名称为 lan 或 Internal 的 VLAN Switch 接口，编辑该接口，交换接口成员默认包含硬件交换模块中的成员接口，这些接口可以从预置的交换接口中取出，变为普通物理口，也可将硬件交换模块的接口加入该交换接口，如下 FortiGate101F 所示。

重要

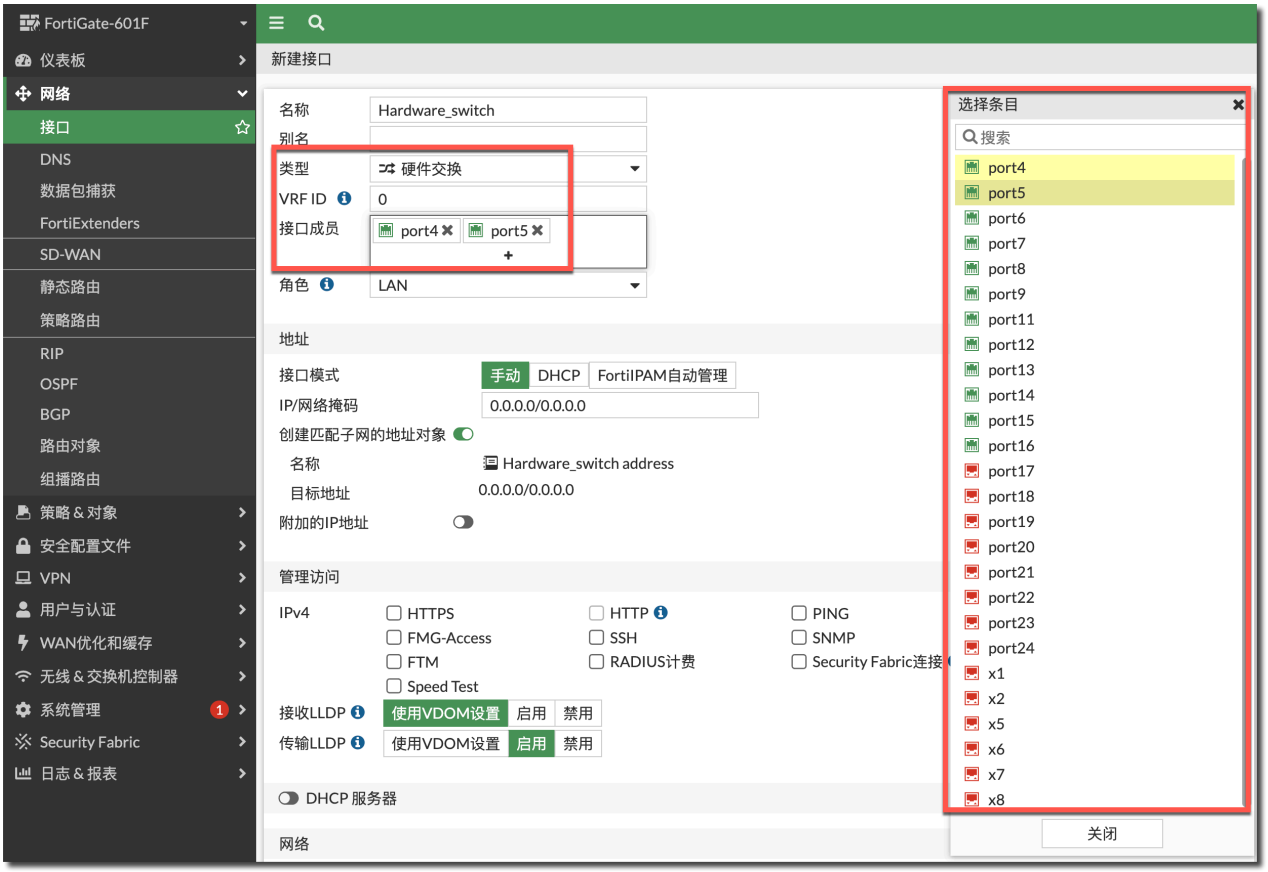
将物理口加入 Switch 接口时，接口需要清除 IP 地址（配置为 0.0.0.0/0），且不能被其他功能引用。



2. 创建新的 VLAN/Hardware Switch，进入“网络 → 接口”页面，新建接口。



3. 选择类型为“VLAN 交换/硬件交换”，如下图所示为 FortiGate601F，交换接口为“硬件交换”，并添加或删除接口成员。



4. 通过 CLI 删除 VLAN/Hardware Switch 接口。

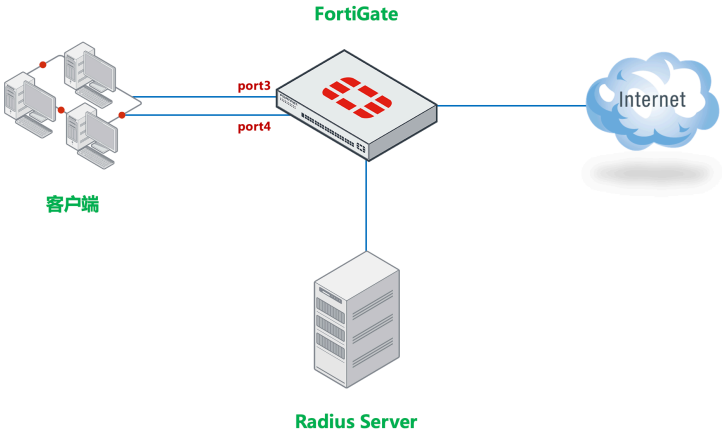
```
config system virtual-switch
    edit "internal"
        config port
            delete internal2
            delete internal7
            ...
        end
    next
end
```

5. 通过 CLI 添加 VLAN/Hardware Switch 接口。

```
config system virtual-switch
    edit "internal"
        set physical-switch "sw0"
        config port
            edit "internal3"
            next
            edit "internal5"
            next
            edit "internal4"
            next
            edit "internal6"
            next
        end
    next
end
```

在 VLAN/Hardware Switch 上使用 802.1X

网络拓扑



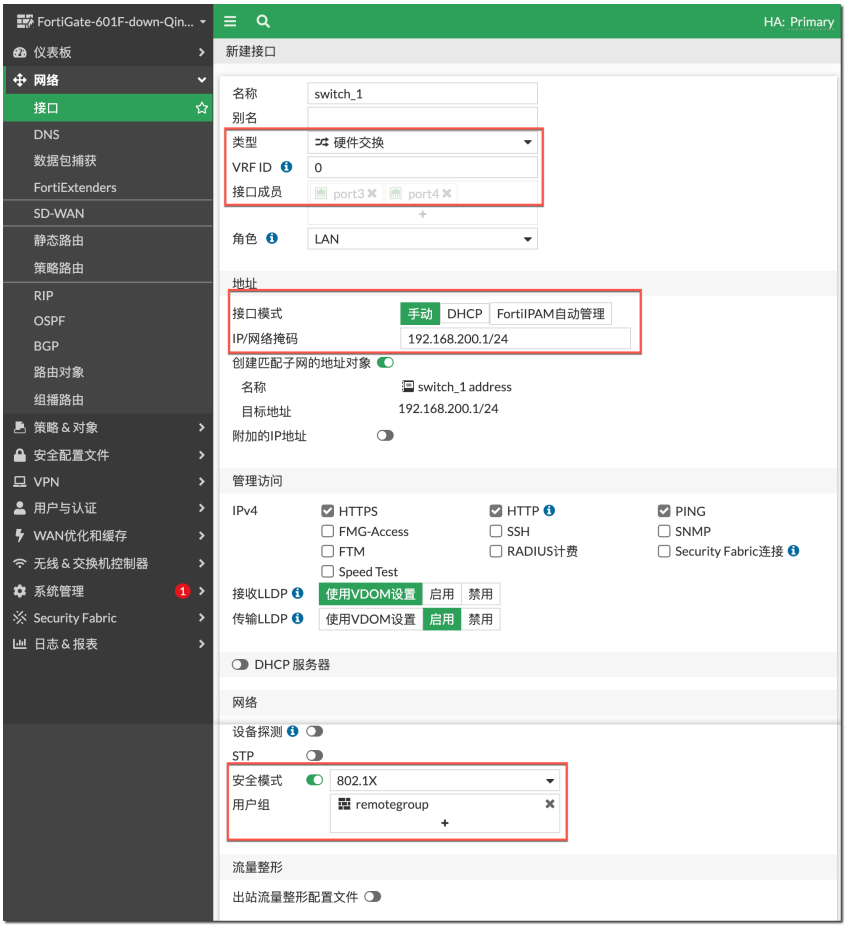
port3 和 port4 是硬件交换模块的一部分，配置 Hardware Switch，并将 port3 和 port4 加入，在该接口上执行 802.1X 身份验证。

配置步骤

1. 在 FortiGate 上配置 Radius 服务器（配置方法参考用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。
2. 在 FortiGate 上配置用于 802.1X 远程 Radius 认证的用户组（配置方法参考用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。
3. 在 FortiGate 的“网络 → 接口”页面新建 Hardware/VLAN Switch 接口，将 port3 和 port4 加入该接口，对应 CLI 如下。

```
config system virtual-switch
    edit "switch_1"
        set physical-switch "sw0"
        config port
            edit "port3"
            next
            edit "port4"
            next
        end
    next
end
```

4. 在 FortiGate 上配置防火墙策略允许新建的 Hardware/VLAN Switch 下的流量访问 Internet。
5. 在 Hardware/VLAN Switch 上开启 802.1X 认证，选择创建的 Radius 认证用户组。




```
config system interface
    edit "switch_1"
        set vdom "vdom1"
        set ip 192.168.200.1 255.255.255.0
        set allowaccess ping http https
        set type hard-switch
        set security-mode 802.1X
        set security-groups "remotegroup"
        set device-identification disable
        set role lan
    next
end
```

结果验证

1. 客户端通过 port3 或 port4 连接，并通过 802.1X 认证后，可以正常访问 Internet。

2. 在 FortiGate 上查看 802.1X 认证端口状态。

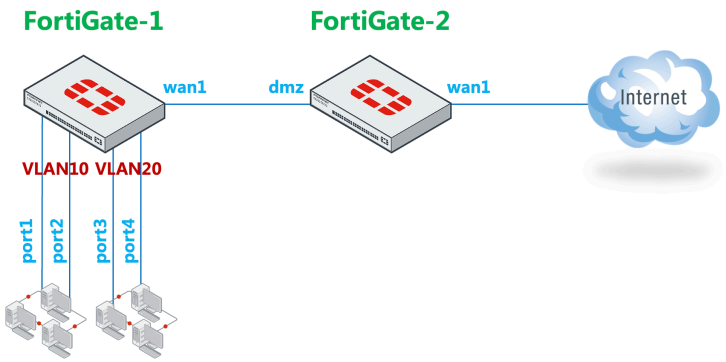
```
FortiGate # diagnose sys 802-1x status

Virtual switch 'switch_1' (default mode) 802.1x member status:
  port3: Link up, 802.1X state: authorized
  port4: Link up, 802.1X state: unauthorized
```

在 VLAN/Hardware Switch 传输带 VLAN Tag 的流量

网络需求

VLAN/Hardware Switch 也可以让 FortiGate 像二层 VLAN 交换机一样工作，进行 VLAN 交换，并可以设置一个物理口作为 Trunk 接口。支持的型号如：60F、80F、100E、100F、140E、200F、300E、400E、1100E、1800F、2600F、3500F、4200F、4400F。

网络拓扑



- FortiGate-1 上有两个硬件交换机接口（port1、port2）被分配了 VLAN 10，两个接口（port3、port4）被分配了 VLAN 20。wan1 接口被指定为 Trunk 端口，与上游的 FortiGate-2 相连。
- 上游 FortiGate-2 的子接口 VLAN10 和 VLAN20 允许 FortiGate-1 过来的带 VLAN Tag 10 和 20 的流量访问 Internet。

配置步骤

基础配置

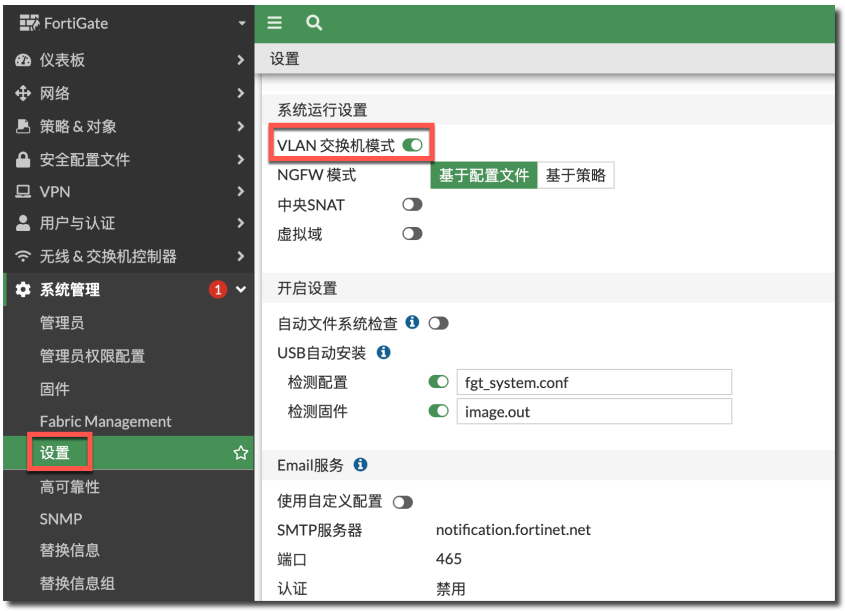
需要确保在“config sys global”下，“virtual-switch-vlan”选项处于“enable”状态。

```
config sys global
    set virtual-switch-vlan enable
end
```

FortiGate-1

1. 需要确保在“config sys global”下，“virtual-switch-vlan”选项处于“enable”状态；或在设备管理页面的“系统管理 → 设置 → 系统运行设置”中，“VLAN 交换机模式”选项处于开启状态。

```
config sys global
    set virtual-switch-vlan enable
end
```



2. 新建两个 VLAN/Hardware Switch，分别配置 VLAN ID 为 10 和 20，分别添加成员 port1、port2 和 port3、port4。

```
config system virtual-switch
    edit "VLAN10"
        set physical-switch "sw0"
        set vlan 10
        config port
            edit "port1"
            next
```

```
            edit "port2"
            next
        end
    next
    edit "VLAN20"
        set physical-switch "sw0"
        set vlan 20
        config port
            edit "port3"
            next
            edit "port4"
            next
        end
    next
end
```

3. 配置两个 VLAN/Hardware Switch 接口的 IP 地址。

```
config system interface
    edit "VLAN10"
        set vdom "root"
        set ip 192.168.10.99 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type hard-switch
    next
    edit "VLAN20"
        set vdom "root"
        set ip 192.168.20.99 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type hard-switch
    next
end
```

4. 配置 wan1 接口为 Trunk 口，承载 VLAN10 和 VLAN20 的流量。

```
config system interface
    edit wan1
        set trunk enable
    next
end
```

FortiGate-2

1. 新建 VLAN 子接口 VLAN10 和 VLAN20，并配置 IP 地址。

```
config system interface
    edit "VLAN10"
        set ip 192.168.10.98 255.255.255.0
        set allowaccess ping https ssh
        set role lan
        set interface "dmz"
        set vlanid 10
    next
    edit "VLAN20"
        set ip 192.168.20.98 255.255.255.0
        set allowaccess ping https ssh
        set role lan
        set interface "dmz"
        set vlanid 20
    next
end
```

2. 在 VLAN10 和 VLAN20 接口上分别配置 DHCP 服务器，分别为 VLAN10 和 VLAN20 的客户端分配 IP 地址。

```
config system dhcp server
    edit 0
        set dns-service default
        set default-gateway 192.168.10.98
        set netmask 255.255.255.0
        set interface "VLAN10"
```

```
        config ip-range
            edit 1
                set start-ip 192.168.10.100
                set end-ip 192.168.10.254
            next
        end
        set timezone-option default
    next
        edit 0
        set dns-service default
        set default-gateway 192.168.20.98
        set netmask 255.255.255.0
        set interface "VLAN20"
        config ip-range
            edit 1
                set start-ip 192.168.20.100
                set end-ip 192.168.20.254
            next
        end
        set timezone-option default
    next
end
```

3. 配置防火墙策略允许 VLAN10 和 VLAN20 的流量访问 Internet。

```
config firewall policy
    edit 0
        set name "VLAN10-out"
        set srcintf "VLAN10"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 0
        set name "VLAN20-out"
        set srcintf "VLAN20"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

1. FortiGate-1 下 VLAN 10 和 VLAN 20 的 PC 均可以通过 FortiGate-2 分别获取到 VLAN10 和 VLAN20 分配的 IP 地址。
2. FortiGate-1 下 VLAN 10 和 VLAN 20 的 PC 均可以 Ping 通 FortiGate-1 的 VLAN/Hardware Switch 接口 IP。
3. FortiGate-1 下 VLAN 10 和 VLAN 20 的 PC 均可以 Ping 通 FortiGate-2 的 VLAN 子接口 IP。
4. FortiGate-1 下 VLAN 10 和 VLAN 20 的 PC 均可以正常访问 Internet，流量在 FortiGate-1 和 FortiGate-2 之间携带对应的 VLAN Tag。

交换接口的 MAC 地址表

网络需求

通过执行 ARP 表项显示命令可以查看与 FortiGate 相邻的所有设备的 MAC 地址表。

```bash
FortiGate # get system arp
Address           Age(min)   Hardware Addr      Interface
10.10.12.1        0          00:09:0f:09:07:07 wan1
192.168.100.77    0          00:e0:4c:b9:97:7c LAN
192.168.100.177   0          0c:37:96:06:d7:5d LAN
```

但该命令无法显示设备连接到交换接口的具体哪个成员接口，此时需要查看 Hardware/Software 交换接口的成员接口所连接设备的 MAC 表。

查看方法

1. 例如一个软交换接口的配置如下，名称为 LAN。

   ```bash
   config system switch-interface
       edit "LAN"
           set vdom "root"
           set member "port2" "port3" "port4" "port5"
       next
   end
   ```

2. 如果想知道连接到了这个软交换接口的成员接口的设备 MAC，可执行如下命令，可以看到 00:41:72:74:1e:01（192.168.120.2）连接到软件交换接口的 port2，00:41:74:6c:11:01（192.168.120.3）连接到 port3。

   重要

   只支持 Hardware/Software 交换接口，VLAN 交换接口无法查看。

   ```bash
   FortiGate # diagnose netlink brctl name host LAN
   show bridge control interface LAN host.
   fdb: hash size=32768, used=6, num=6, depth=1, gc_time=4, ageing_time=3, simple=switch
   Bridge LAN host table
   port  no     device  devname mac addr      ttl     attributes
     1    4      port2   00:41:72:74:1e:01     30      Hit(30)
     2    5      port3   00:41:74:6c:11:01      7      Hit(7)
   ```

SD-WAN

SD-WAN 介绍

- SD-WAN 数据转发逻辑
- SD-WAN 接口成员组
- SD-WAN 健康检查

SD-WAN 规则

- Manual（手动选路）
- Best Quality（最佳质量）
- Lowest Cost (SLA)
- Maximize Bandwidth (SLA)

  基于轮询的 Maximize Bandwidth (SLA)　基于带宽的 Maximize Bandwidth (SLA)

SD-WAN 配置举例

- 将已关联接口移至 SD-WAN 区域
- 互联网出口场景
- 互联网出口和 VPN 混合场景
- ADVPN 场景

健康检查配置实践

SD-WAN 常用 SNMP 监控项

SD-WAN 排错

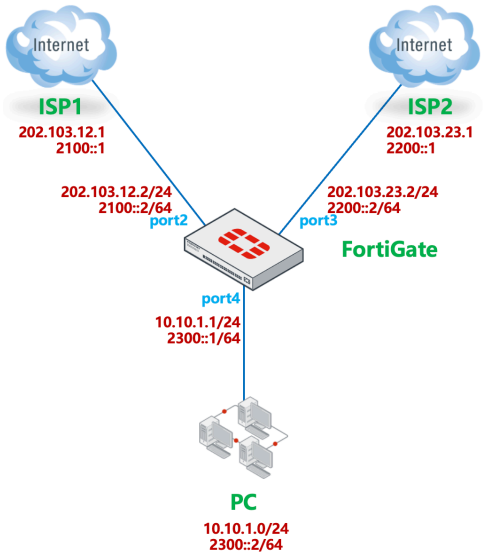
健康检查配置实践

概述

在 SD-WAN 的健康检查中，探测目标可以配置 1 个 Server 或 2 个 Server，本文主要探讨：

- 健康检查探测目标个数/检测状态（up/down）对健康检查状态（alive/dead）的影响。
- 目标服务器数量对 Manual 模式的 SD-WAN 规则选路产生的影响。
- 健康检查选项中的“更新静态路由”（`update-static-route`）对 SD-WAN 选路的作用，以及与探测目标服务器数量相关的差异化表现。
- 针对不同应用场景，提供相应的配置建议。

网络拓扑



- FortiGate 有两条 ISP 线路（IPv4/IPv6 双栈）连接 Internet，两条 ISP 线路做双栈 SD-WAN。
- FortiGate 配置健康检查，检查 Internet 上目标服务器（IPv4/IPv6）的状态，以健康检查的结果决定 SD-WAN 选路。
- PC 通过 FortiGate 的 SD-WAN 选路来访问 Internet。
- 基础网络配置略，SD-WAN 基础配置与初始路由表，路由表中默认路由在两个 ISP 接口上负载：

```bash
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port2"
            set gateway 202.103.12.1
            set gateway6 2100::1
        next
        edit 2
            set interface "port3"
            set gateway 202.103.23.1
            set gateway6 2200::1
        next
    end
end
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

```
config router static6
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end

FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
FortiGate # get router info6 routing-table
Routing table for VRF=0
S*      ::/0 [1/0] via 2100::1, port2, 17:24:49, [1024/0]
             [1/0] via 2200::1, port3, 17:24:49, [1024/0]
C       ::1/128 via ::, root, 3d18h28m
C       2100::/64 via ::, port2, 17:25:23
C       2200::/64 via ::, port3, 17:25:13
C       2300::/64 via ::, port4, 00:00:04
```

1 个健康检查配置 2 个 Server

基础配置

1. 在 FortiGate 上创建健康检查，配置 2 个 IPv4 探测 Server，检测 223.5.5.5（在前）和 223.6.6.6（在后），参与接口选择所有 SD-WAN 成员（port2 和 port3），默认开启 `update-static-route`。

```bash
config health-check
    edit "Server"
        set server "223.5.5.5" "223.6.6.6"
        set members 0
    next
end
```

2. 查看健康检查的状态，两条线路均为 alive 状态。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(15.662), jitter(11.245), mos(4.385), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(15.519), jitter(11.240), mos(4.385), bandwidth-up(9999998),
bandwidth-dw(9999998), bandwidth-bi(19999996) sla_map=0x0
```

3. 配置 Manual 模式的 SD-WAN 规则，将 PC 网段的流量优先通过 ISP1（port2）线路进行转发，其次通过 ISP2（port3）线路进行转发。

```bash
config system sdwan
    config service
        edit 1
            set name "to_Internet"
            set dst "all"
```

```
            set src "all"
            set priority-members 1 2
        next
    end
end
```

4. 查看 SD-WAN 规则选路状态，优先选择 ISP1（port2）线路转发。




```
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 port2), alive, selected
    2: Seq_num(2 port3), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

5. 抓取探测报文可以看到，ISP1（port2）和 ISP2（port3）接口会同时探测 223.5.5.5 和 223.6.6.6 两个 Server。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.386576 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.386658 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.386675 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.386698 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.392926 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.393021 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
0.393048 port2 in 223.5.5.5 → 202.103.12.2: icmp: echo reply
0.393075 port2 in 223.6.6.6 → 202.103.12.2: icmp: echo reply
```

> **重要**
>
> 后续本文使用如下简写来表示每个线路上对两个 Server 的探测结果：
>
> - `ISP1#1`：ISP1 上对 223.5.5.5 的探测结果。
> - `ISP1#2`：ISP1 上对 223.6.6.6 的探测结果。
> - `ISP2#1`：ISP2 上对 223.5.5.5 的探测结果。
> - `ISP2#2`：ISP2 上对 223.6.6.6 的探测结果。

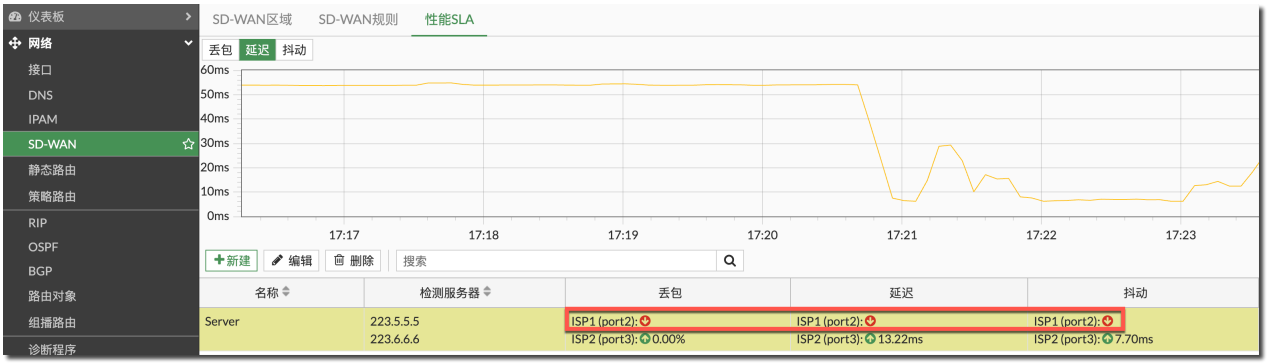
`ISP1#1` down，其他 up

1. 在 ISP1 上模拟探测的目标服务器 223.5.5.5（位于健康检查的 Server 序列 1）失效，其他正常访问。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.316221 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.316311 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
```

```
0.316327 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.316354 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.322336 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.322364 port2 in 223.6.6.6 → 202.103.12.2: icmp: echo reply
0.322412 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
```

2. 此时查看健康检查状态，显示 ISP1 的状态变为 dead。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(7.209), jitter(2.282), mos(4.399), bandwidth-up(9999998),
bandwidth-dw(9999998), bandwidth-bi(19999996) sla_map=0x0
```

3. 查看 FortiGate 的 SD-WAN 规则选路状态，选择 ISP2 线路（port3）进行转发。




```bash
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(2 port3), alive, selected
    2: Seq_num(1 port2), dead
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 查看路由表更新状态，port2 的默认路由仍然存在于路由表中，没有被删除。

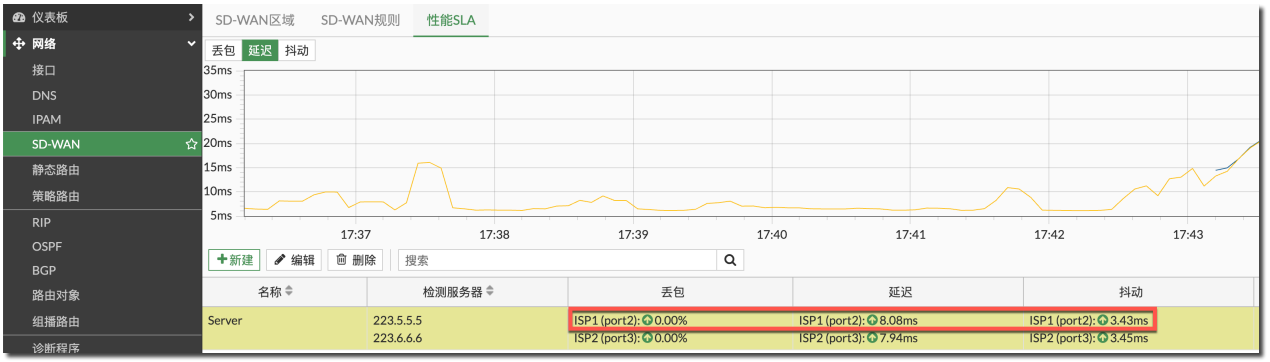
```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

`ISP1#2` down，其他 up

1. 将所有 ISP 故障恢复。在 ISP1 上模拟 223.6.6.6（位于健康检查的 Server 序列 2）故障，其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.228094 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.228174 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.228194 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.228218 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.234403 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.234449 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
0.234472 port2 in 223.5.5.5 → 202.103.12.2: icmp: echo reply
```

2. 此时查看健康检查状态，显示 ISP1 的状态变为仍为 alive。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(6.296), jitter(0.547), mos(4.401), bandwidth-up(9999998),
bandwidth-dw(9999999), bandwidth-bi(19999997) sla_map=0×0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(6.159), jitter(0.528), mos(4.401), bandwidth-up(9999998),
bandwidth-dw(9999998), bandwidth-bi(19999996) sla_map=0×0
```

3. 查看 FortiGate 的 SD-WAN 规则选路状态，仍然选择 ISP1 线路（port2）进行转发。




```bash
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(5), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
    1: Seq_num(1 port2), alive, selected
    2: Seq_num(2 port3), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 查看路由表更新状态，port2 的默认路由仍然存在于路由表中，没有被删除。

```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

重要

健康检查状态对 Manual 模式 SD-WAN 规则的影响（1 个健康检查引用接口）：

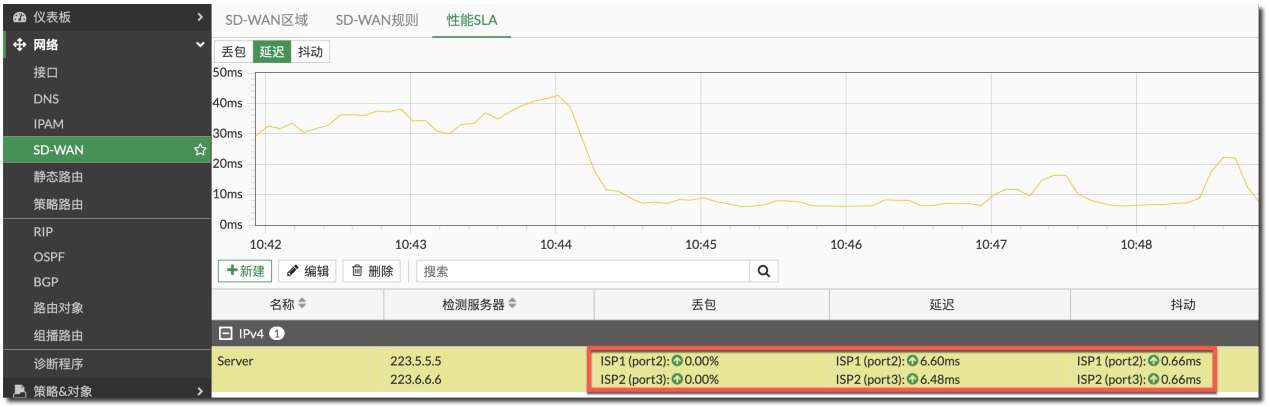
- 只要健康检查在这个接口上为 alive 状态，该线路就会被认为 alive，SD-WAN 规则不会切换下一个成员接口转发。
- 只要健康检查在这个接口上变为 dead 状态，该线路就会被认为 dead，SD-WAN 规则切换下一个 alive 成员接口转发。

`ISP1#1` & `ISP2#1` down，其他 up

1. 将所有 ISP 故障恢复。在 ISP1 和 ISP2 上同时模拟 223.5.5.5（位于健康检查的 Server 序列 1）故障，其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.386576 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.386658 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.386675 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.386698 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.393021 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
0.393075 port2 in 223.6.6.6 → 202.103.12.2: icmp: echo reply
```

2. 此时查看健康检查状态，显示 ISP1 和 ISP2 的状态均为 alive。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(7.390), jitter(1.991), mos(4.399), bandwidth-up(9999998),
bandwidth-dw(9999999), bandwidth-bi(19999997) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(7.218), jitter(1.961), mos(4.399), bandwidth-up(9999998),
bandwidth-dw(9999999), bandwidth-bi(19999997) sla_map=0x0
```

3. 查看 FortiGate 的 SD-WAN 规则选路状态，仍然选择 ISP1 线路（port2）进行转发。



4. 查看路由表更新状态，port2 的默认路由仍然存在于路由表中，没有被删除。

```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

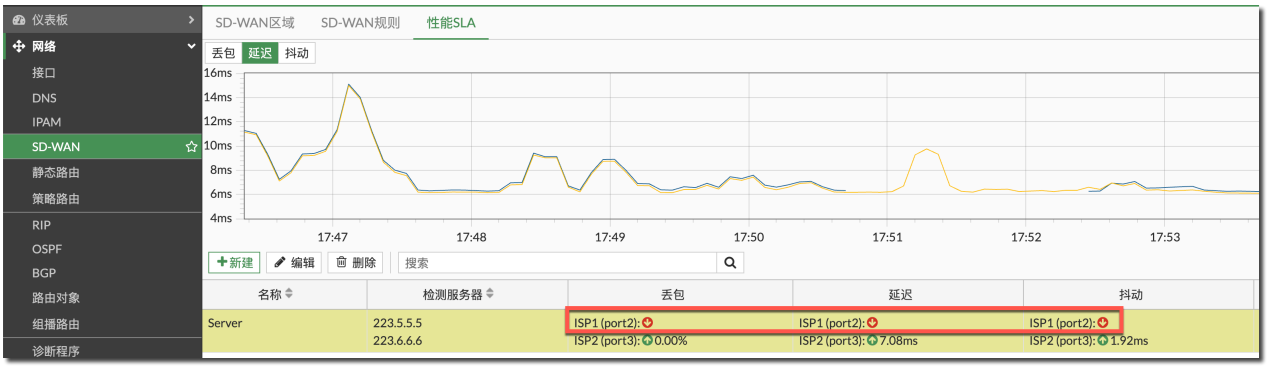
> **重要**
>
> **2 个 Server 与健康检查状态的关系：健康检查在每条线路上的状态只由 1 个 Server 的探测结果决定。**
>
> -**优先使用 #1 的探测结果**：只要 #1 在任何一条线路上为 up，就忽略 #2 的探测结果（例如，#1 在 ISP1 为 down，但在 ISP2 为 up，此时 FortiGate 会判断这是线路问题，而不是 #1 服务器宕机，所以仍然采纳 #1 的探测结果，不会采纳 #2 的探测结果）。 -**当 #1 在所有线路上为 down 时**：健康检查才会采纳 #2 的探测结果（此时 FortiGate 认定 #1 服务器已经宕机，并开始采纳 #2 的探测结果）。

`ISP1#1` & `ISP1#2` down，其他 up

1. 将所有 ISP 故障恢复。在 ISP1 上模拟 223.5.5.5 和 223.6.6.6 同时故障（ISP2 上 223.5.5.5 和 223.6.6.6 都可以访问），其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.382564 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.382643 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.382669 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.382692 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.388660 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.388690 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
```

2. 此时查看健康检查状态，显示 ISP1（port2）的状态变为 dead。



3. 查看 FortiGate 的 SD-WAN 规则选路状态，选择 ISP2 线路（port3）进行转发。




```bash
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
    1: Seq_num(2 port3), alive, selected
    2: Seq_num(1 port2), dead
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 查看路由表更新状态，port2 的默认路由被删除。

```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

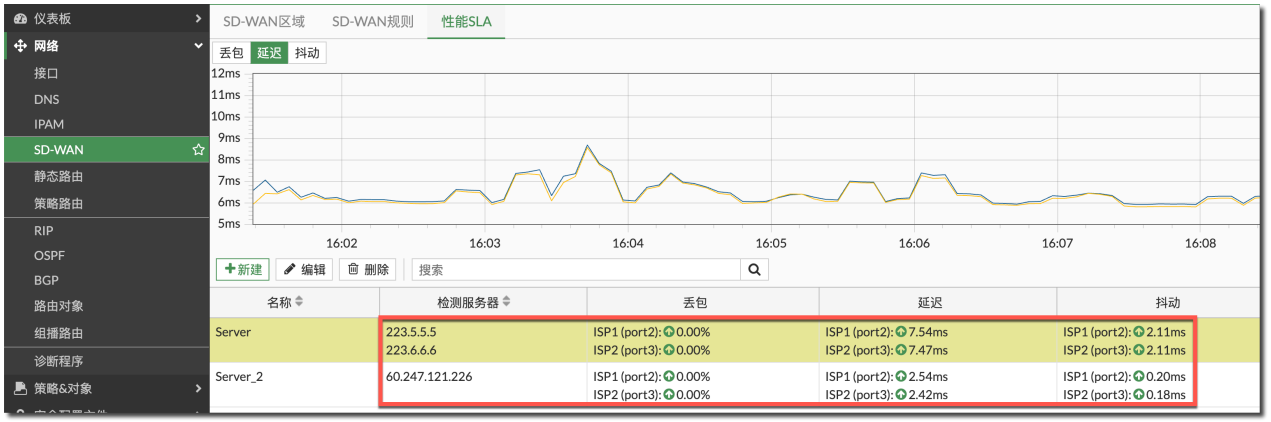
2 个健康检查引用相同接口（IPv4）

基础配置

1. 恢复所有的 ISP 故障模拟。新建 1 个健康检查“Server_2”，探测 Server 为 60.247.121.226，同时引用 ISP1 线路接口（port2）和 ISP2 线路接口（port3）。

```bash
config system sdwan
    config health-check
        edit "Server"
            set server "223.5.5.5" "223.6.6.6"
            set members 0
        next
        edit "Server_2"
            set server "60.247.121.226"
            set members 0
        next
    end
end
```

2. 查看健康检查的状态，两个健康检查在所有 ISP 上均为 alive 状态。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(7.180), jitter(3.035), mos(4.398), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(7.134), jitter(3.012), mos(4.398), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
Health Check(Server_2):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(2.530), jitter(0.208), mos(4.403), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(2.419), jitter(0.215), mos(4.403), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
```

重要

后续本文使用如下简写来表示每个线路上对两个 Server 的探测结果：

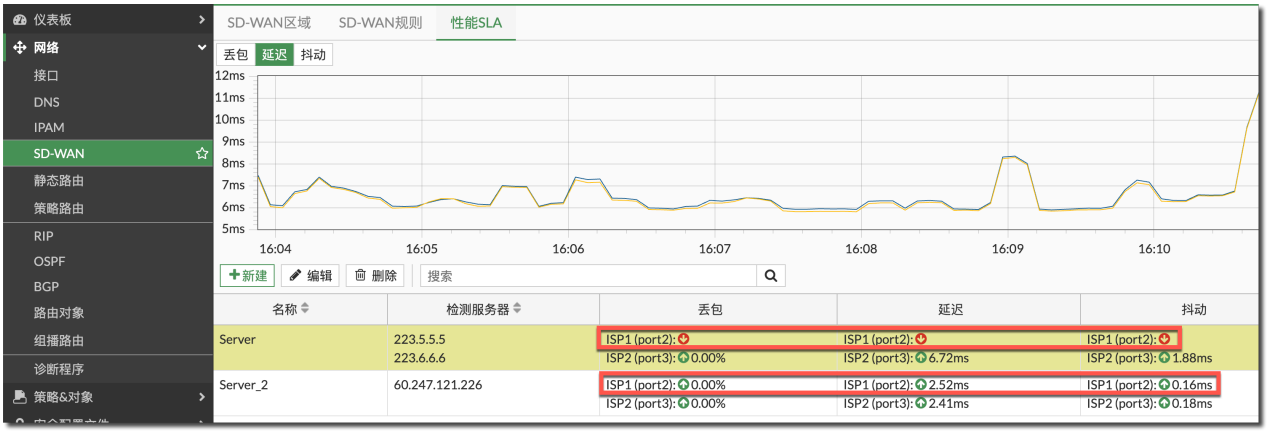
- `ISP1#1`：健康检查“” Server“ISP1 上对 223.5.5.5 的探测结果。
- `ISP1#2`：健康检查“” Server“ISP1 上对 223.6.6.6 的探测结果。
- `ISP2#1`：健康检查“” Server“ISP2 上对 223.5.5.5 的探测结果。
- `ISP2#2`：健康检查“” Server“ISP2 上对 223.6.6.6 的探测结果。
- `ISP1#Server_2`：健康检查“” Server_2“ISP1 上对 60.247.121.226 的探测结果。
- `ISP2#Server_2`：健康检查“” Server_2“ISP2 上对 60.247.121.226 的探测结果。

`ISP1#1` down，其他 up

1. 在 ISP1 上模拟探测的目标服务器 223.5.5.5（位于健康检查“Server”的序列 1）失效，其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.156992 port2 out 202.103.12.2 → 60.247.121.226: icmp: echo request
0.157072 port3 out 202.103.23.2 → 60.247.121.226: icmp: echo request
0.157106 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.157127 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.157196 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.157220 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.159406 port2 in 60.247.121.226 → 202.103.12.2: icmp: echo reply
0.159485 port3 in 60.247.121.226 → 202.103.23.2: icmp: echo reply
0.162822 port2 in 223.6.6.6 → 202.103.12.2: icmp: echo reply
0.162899 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.162916 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
```

2. 此时查看健康检查状态，健康检查“Server”中的 ISP1 线路变为 dead 状态，健康检查“Server_2”中的 ISP1 线路仍为 alive 状态。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0×0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(9.357), jitter(4.683), mos(4.395), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0×0
Health Check(Server_2):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(2.509), jitter(0.169), mos(4.403), bandwidth-up(9999996),
bandwidth-dw(9999996), bandwidth-bi(19999992) sla_map=0×0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(2.403), jitter(0.152), mos(4.403), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0×0
```

3. 查看 SD-WAN 规则选路状态，仍然优先选择 ISP1（port2）线路转发。




```bash
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
```

```
    1: Seq_num(1 port2), alive, selected
    2: Seq_num(2 port3), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 查看路由表更新状态，port2 的默认路由仍然存在于路由表中，没有被删除。

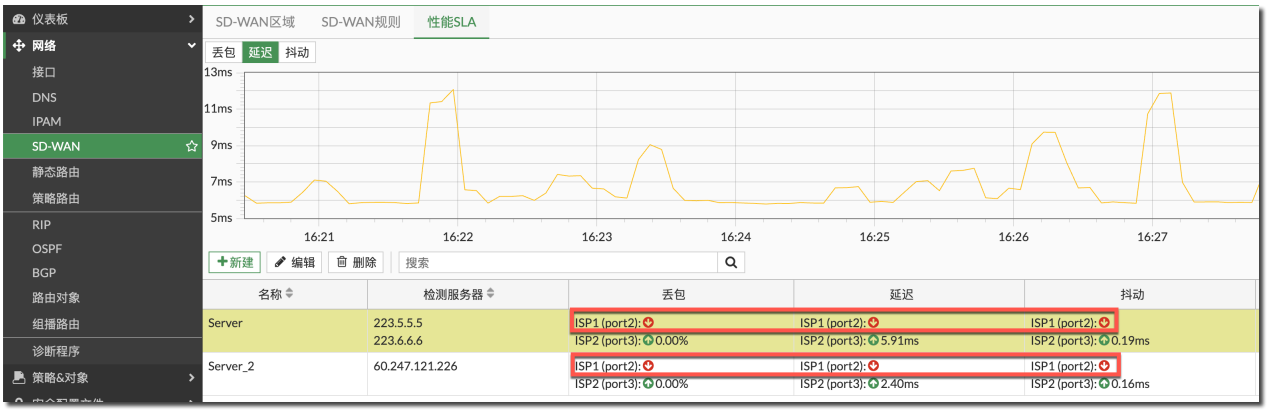
```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

`ISP1#1` & `ISP1#Server_2` down，其他 up

1. 将所有 ISP 故障恢复。在 ISP1 线路（port2）上模拟探测的目标服务器 223.5.5.5（位于健康检查“Server”的序列 1）和 60.247.121.226（位于健康检查“Server_2”）失效，其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.424523 port2 out 202.103.12.2 → 60.247.121.226: icmp: echo request
0.424724 port3 out 202.103.23.2 → 60.247.121.226: icmp: echo request
0.425029 port2 out 202.103.12.2 → 223.5.5.5: icmp: echo request
0.425091 port2 out 202.103.12.2 → 223.6.6.6: icmp: echo request
0.425162 port3 out 202.103.23.2 → 223.5.5.5: icmp: echo request
0.425210 port3 out 202.103.23.2 → 223.6.6.6: icmp: echo request
0.427347 port3 in 60.247.121.226 → 202.103.23.2: icmp: echo reply
0.430943 port3 in 223.5.5.5 → 202.103.23.2: icmp: echo reply
0.430983 port3 in 223.6.6.6 → 202.103.23.2: icmp: echo reply
0.431002 port2 in 223.6.6.6 → 202.103.12.2: icmp: echo reply
```

2. 此时查看健康检查状态，健康检查“Server_2”中的 ISP1 线路变为 dead 状态。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(7.006), jitter(2.388), mos(4.398), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
Health Check(Server_2):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(2.331), jitter(0.189), mos(4.403), bandwidth-up(9999997),
bandwidth-dw(9999997), bandwidth-bi(19999994) sla_map=0x0
```

3. 查看 SD-WAN 规则选路状态，变为优先选择 ISP2（port3）线路转发。




```bash
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(2 port3), alive, selected
    2: Seq_num(1 port2), dead
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

重要

健康检查状态对 Manual 模式 SD-WAN 规则的影响（多个健康检查引用接口）：

当配置多个健康检查引用了同一个接口时，需要所有引用此接口的健康检查同时变为 dead 状态，该线路才会被认为 dead，SD-WAN 规则选择从下一个 alive 成员接口转发。

4. 查看路由表更新状态，port2 的默认路由仍然存在于路由表中，没有被删除。

```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.12.1, port2, [1/0]
                  [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

`ISP1#1` & `ISP1#2` & `ISP1#Server_2` down，其他 up

1. 将所有 ISP 故障恢复。在 ISP1 线路（port2）上模拟探测的目标服务器 223.5.5.5（位于健康检查“Server”的序列 1）、223.6.6.6（位于健康检查“Server”的序列 2）和 60.247.121.226（位于健康检查“Server_2”）失效，其他正常访问。

```bash
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.534318 port2 out 202.103.12.2 -> 223.5.5.5: icmp: echo request
0.534508 port2 out 202.103.12.2 -> 223.6.6.6: icmp: echo request
0.534587 port3 out 202.103.23.2 -> 223.5.5.5: icmp: echo request
0.534648 port3 out 202.103.23.2 -> 223.6.6.6: icmp: echo request
0.534686 port2 out 202.103.12.2 -> 60.247.121.226: icmp: echo request
0.534748 port3 out 202.103.23.2 -> 60.247.121.226: icmp: echo request
0.536783 port3 in 60.247.121.226 -> 202.103.23.2: icmp: echo reply
0.540581 port3 in 223.5.5.5 -> 202.103.23.2: icmp: echo reply
0.540609 port3 in 223.6.6.6 -> 202.103.23.2: icmp: echo reply
```

2. 查看路由表更新状态，port2 的默认路由被删除。

```bash
FortiGate # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.23.1, port3, [1/0]
C       10.10.1.0/24 is directly connected, port4
C       202.103.12.0/24 is directly connected, port2
C       202.103.23.0/24 is directly connected, port3
```

重要

多个健康检查与更新静态路由的关系：

- 只有当接口关联的所有健康检查中的探测服务器都处于 down 状态时，相关接口上的路由才会被删除。
- 即使接口关联的所有健康检查显示为 dead 状态，但只要有一个健康检查中的服务器为 up（例如：#1 为 down，#2 为 up），就不会触发更新静态路由（`update-static-route`）的动作。

2 个健康检查引用相同接口（IPv4 + IPv6）

1. 在以上配置和故障模拟的基础上，新建健康检查“Server_3”的探测目标为 IPv6 地址。

```bash
config system sdwan
    config health-check
        edit "Server"
            set server "223.5.5.5" "223.6.6.6"
            set members 0
        next
        edit "Server_2"
            set server "60.247.121.226"
            set members 0
        next
        edit "Server_3"
            set addr-mode ipv6
            set server "::223:5:5:5"
            set members 0
        next
    end
end
```

2. 查看健康检查的状态，IPv4 健康检查在 ISP1 线路（port2）上均为 dead 状态，IPv6 健康检查在 ISP2 线路（port3）上为 alive 状态。




```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(Server):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(2.000%) latency(5.965), jitter(0.724), mos(4.400), bandwidth-up(9999996),
bandwidth-dw(9999996), bandwidth-bi(19999992) sla_map=0x0
Health Check(Server_2):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(2.000%) latency(2.414), jitter(0.160), mos(4.402), bandwidth-up(9999996),
bandwidth-dw(9999996), bandwidth-bi(19999992) sla_map=0x0
Health Check(Server_3):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(2.351), jitter(0.188), mos(4.403), bandwidth-up(9999996),
bandwidth-dw(9999998), bandwidth-bi(19999994) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(2.281), jitter(0.224), mos(4.403), bandwidth-up(9999996),
bandwidth-dw(9999996), bandwidth-bi(19999992) sla_map=0x0
```

3. 查看 SD-WAN 规则（IPv4）选路状态，变为优先选择 ISP1（port2）线路转发。




```
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
    1: Seq_num(1 port2), alive, selected
    2: Seq_num(2 port3), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

bash

重要

健康检查状态对 Manual 模式 SD-WAN 规则的影响（多个 IPv4/IPv6 健康检查同时引用接口）：

引用同一接口的健康检查探测目标 Server 有 IPv4、IPv6 两种，必须满足所有接口相关的健康检查为 dead 状态时，该线路才会被认为 dead，Manual 模式的 SD-WAN 规则（无论是 IPv4 规则还是 IPv6 规则）才会选择从下一个 alive 成员接口转发。

总结

健康检查目标个数对健康检查状态的影响

- **1 个目标服务器**：如果健康检查配置只有 1 个探测 Server，则健康检查状态直接由该目标的状态决定。探测目标 down 时，健康检查被标记为 dead。
- **2 个目标服务器**：健康检查在每条线路上的状态只由 1 个 Server 的探测结果决定。 -**优先使用 #1 的探测结果**：只要 #1 在任何一条线路上为 up，就忽略 #2 的探测结果（例如，#1 在 ISP1 为 down，但在 ISP2 为 up，此时 FortiGate 会判断这是线路问题，而不是 #1 服务器宕机，所以仍然采纳 #1 的探测结果，不会采纳 #2 的探测结果）。 -**当 #1 在所有线路上为 down 时**：健康检查才会采纳 #2 的探测结果（此时 FortiGate 认定 #1 服务器已经宕机，并开始采纳 #2 的探测结果）。

健康检查状态对 Manual 模式 SD-WAN 规则选路的影响

- **1 个健康检查引用接口：**
  - 只要健康检查在这个接口上为 alive 状态，该线路就会被认为 alive，SD-WAN 规则不会切换下一个成员接口转发。
  - 只要健康检查在这个接口上变为 dead 状态，该线路就会被认为 dead，SD-WAN 规则切换下一个 alive 成员接口转发（即使健康检查关闭了 `update-static-route`），如果 SD-WAN 规则的所有成员都变为 dead 状态，则该条 SD-WAN 规则会被跳过匹配（即使健康检查关闭了 `update-static-route`）。
- **多个健康检查引用接口：**

  重要

  Best quality、Lowest cost (SLA)、Maximize bandwidth (SLA) 模式的 SD-WAN 必须选择一个具体的健康检查，不需要考虑下列情况。

  - 当配置多个健康检查引用了同一个接口时，需要所有引用此接口的健康检查同时变为 dead 状态，该线路才会被认为 dead，SD-WAN 规则选择从下一个 alive 成员接口转发。
  - 引用同一接口的健康检查探测目标 Server 有 IPv4、IPv6 两种，必须满足所有接口相关的健康检查为 dead 状态时，该线路才会被认为 dead，SD-WAN 规则（无论是 IPv4 规则还是 IPv6 规则）才会选择从下一个 alive 成员接口转发。

多个健康检查与更新静态路由的关系

- 只有当接口关联的所有健康检查中的探测服务器都处于 down 状态时，相关接口上的路由才会被删除。

- 即使接口关联的所有健康检查显示为 dead 状态，但只要有一个健康检查中的服务器为 up（例如：#1 为 down，#2 为 up），就不会触发更新静态路由（`update-static-route`）的动作。

配置建议

健康检查的探测目标

- 优选稳定、低延迟、高可靠性的探测目标（如公共 DNS 服务器）。
- 在高可用需求场景中，可以在健康检查中配置 2 个目标服务器并合理设置先后顺序。
- 确保接口关联的健康检查数量和目标足够精准，避免冗余，尽量不要使用多个健康检查引用同一个接口。

更新静态路由

在配置健康检查时，是否开启 `update-static-route` 功能取决于具体的网络环境和需求。以下是关于何时应开启/关闭此功能的建议：

- 建议关闭 `update-static-route` 的情况： -**仅作为测试的健康检查：**即使没有 SD-WAN 规则引用，`update-static-route` 仍会生效，开启该功能可能导致网络中断。因此，对于任何非生产性的测试配置，请关闭 `update-static-route` 选项。 -**单一探测目标：**在健康检查仅配置了 1 个探测服务器，并且 FortiGate 设备自身需要发起流量（例如 IPSec、SSL VPN 或 BGP 通信）的情况下，启用 `update-static-route` 可能导致相关流量路径变化，进而引起服务中断。
- 可以考虑开启 `update-static-route` 的情况： -**多 Server 健康检查：**健康检查配置多个 Server 时，可以根据实际情况考虑开启更新静态路由（`update-static-route`）。 -**IPv4/IPv6 共存：**对于同时支持 IPv4 和 IPv6 的网络环境，为 IPv4/IPv6 分别配置了健康检查目标。

SD-WAN 常用 SNMP 监控项

SD-WAN SNMP OID

SNMP 节点名称	OID	说明
fgVWLHealthCheckLinkNumber	.1.3.6.1.4.1.12356.101.4.9.1	fgVWLHealthCheckLinkTable 中的健康检查的链路数
fgVWLHealthCheckLinkTable	.1.3.6.1.4.1.12356.101.4.9.2	SD-WAN 健康检查统计表，包含以下的所有 OID 结果。该表与 fgVdTable 存在依赖扩展关系，仅包含已配置成员的健康检查
fgVWLHealthCheckLinkEntry	.1.3.6.1.4.1.12356.101.4.9.2.1	VDOM ID 1 上的 SD-WAN 健康检查统计表（如果是 VDOM ID 2，则是.1.3.6.1.4.1.12356.101.4.9.2.2，以此类推）
fgVWLHealthCheckLinkID	.1.3.6.1.4.1.12356.101.4.9.2.1.1	SD-WAN 健康检查的 Seq ID（对应 diagnose sys sdwan health-check 中的 Seq）。此表中仅包含已配置成员的健康检查（SD-WAN 的健康检查链路 ID 在每个 VDOM 内是唯一的）
fgVWLHealthCheckLinkName	.1.3.6.1.4.1.12356.101.4.9.2.1.2	健康检查名称
fgVWLHealthCheckLinkSeq	.1.3.6.1.4.1.12356.101.4.9.2.1.3	SD-WAN 成员线路的 ID
fgVWLHealthCheckLinkState	.1.3.6.1.4.1.12356.101.4.9.2.1.4	某个成员的健康检查状态
fgVWLHealthCheckLinkLatency	.1.3.6.1.4.1.12356.101.4.9.2.1.5	在过去的 30 次探测周期内，成员健康检查的平均延迟，以 float 格式表示
fgVWLHealthCheckLinkJitter	.1.3.6.1.4.1.12356.101.4.9.2.1.6	在过去的 30 次探测周期内，成员健康检查的平均抖动，以 float 格式表示
fgVWLHealthCheckLinkPacketSend	.1.3.6.1.4.1.12356.101.4.9.2.1.7	在某个成员线路上已发送的健康检查报文总数
fgVWLHealthCheckLinkPacketRecv	.1.3.6.1.4.1.12356.101.4.9.2.1.8	在某个成员线路上已接收的健康检查报文总数
fgVWLHealthCheckLinkPacketLoss	.1.3.6.1.4.1.12356.101.4.9.2.1.9	在过去的 30 次探测周期内，成员健康检查的平均丢包率，以 float 格式表示
fgVWLHealthCheckLinkVdom	.1.3.6.1.4.1.12356.101.4.9.2.1.10	健康检查所在的 VDOM 名称，该名称对应 fgVdTable 中使用的 fgVdEntName
fgVWLHealthCheckLinkBandwidthIn	.1.3.6.1.4.1.12356.101.4.9.2.1.11	健康检查在某个成员链接上检测到的入向可用带宽，以 Mbps 为单位
fgVWLHealthCheckLinkBandwidthOut	.1.3.6.1.4.1.12356.101.4.9.2.1.12	健康检查在某个成员链接上检测到的出向可用带宽，以 Mbps 为单位
fgVWLHealthCheckLinkBandwidthBi	.1.3.6.1.4.1.12356.101.4.9.2.1.13	健康检查在某个成员链接上检测到的双向可用带宽，以 Mbps 为单位
fgVWLHealthCheckLinkIfName	.1.3.6.1.4.1.12356.101.4.9.2.1.14	SD-WAN 成员接口名称

读取结果参考

读取 SNMP 项	OID			
fgVWLHealthCheckLinkNumber	.1.3.6.1.4.1.12356.101.4.9.1	3		
fgVWLHealthCheckLinkID	.1.3.6.1.4.1.12356.101.4.9.2.1.1	1	2	3
fgVWLHealthCheckLinkName	.1.3.6.1.4.1.12356.101.4.9.2.1.2	pingserver	pingserver	pingserver
fgVWLHealthCheckLinkSeq	.1.3.6.1.4.1.12356.101.4.9.2.1.3	2	1	3
fgVWLHealthCheckLinkState	.1.3.6.1.4.1.12356.101.4.9.2.1.4	0	0	0
fgVWLHealthCheckLinkLatency	.1.3.6.1.4.1.12356.101.4.9.2.1.5	39.302	43.124	44.348
fgVWLHealthCheckLinkJitter	.1.3.6.1.4.1.12356.101.4.9.2.1.6	4.346	3.951	5.05
fgVWLHealthCheckLinkPacketSend	.1.3.6.1.4.1.12356.101.4.9.2.1.7	3657689	3657689	3657689
fgVWLHealthCheckLinkPacketRecv	.1.3.6.1.4.1.12356.101.4.9.2.1.8	3196258	3220258	3219466
fgVWLHealthCheckLinkPacketLoss	.1.3.6.1.4.1.12356.101.4.9.2.1.9	0	0	0
fgVWLHealthCheckLinkVdom	.1.3.6.1.4.1.12356.101.4.9.2.1.10	root	root	root
fgVWLHealthCheckLinkBandwidthIn	.1.3.6.1.4.1.12356.101.4.9.2.1.11	9999963	9999937	9999999
fgVWLHealthCheckLinkBandwidthOut	.1.3.6.1.4.1.12356.101.4.9.2.1.12	9999981	9999953	9999998
fgVWLHealthCheckLinkBandwidthBi	.1.3.6.1.4.1.12356.101.4.9.2.1.13	19999944	19999890	19999997
fgVWLHealthCheckLinkIfName	.1.3.6.1.4.1.12356.101.4.9.2.1.14	MPLS	port1	port2

SD-WAN 排错

SD-WAN 日志

健康检查

1. 健康检查对路由的影响：

   - 检测到中断，删除对应接口的静态路由。

   ```bash
   date=2024-01-20 time=17:06:31 eventtime=1618963591590008160 tz="-0700" logid="0100022921" type="event"
   subtype="system" level="critical" vd="root" logdesc="Routing information changed" name="test" interface="R150"
   status="down" msg="Static route on interface R150 may be removed by health-check test. Route:  (10.100.1.2-
   >10.100.2.22 ping-down)"
   ```

   - 健康检查检测到恢复，恢复对应接口的静态路由。

   ```bash
   date=2024-01-20 time=17:11:46 eventtime=1618963906950174240 tz="-0700" logid="0100022921" type="event"
   subtype="system" level="critical" vd="root" logdesc="Routing information changed" name="test" interface="R150"
   status="up" msg="Static route on interface R150 may be added by health-check test. Route:  (10.100.1.2→10.100.2.22
   ping-up)"
   ```

2. SD-WAN 健康检查某个成员的状态变化：

   - 健康检查失败（Dead），停止转发流量。

   ```bash
   date=2024-01-20 time=23:04:32 eventtime=1618985072898756700 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" interface="R150" member="1" serviceid=1
   service="test" gateway=10.100.1.1 msg="Member link is unreachable or miss threshold. Stop forwarding traffic. "
   ```

   - 健康检查从失败（Dead）恢复为 alive，继续转发流量。

   ```bash
   date=2024-01-20 time=23:06:08 eventtime=1618985168018789600 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" interface="R150" member="1" serviceid=1
   service="test" gateway=10.100.1.1 msg="Member link is available. Start forwarding traffic. "
   ```

3. 健康检查中的 SLA 目标：

   - 不满足 SLA 目标。

   ```bash
   date=2024-01-20 time=21:32:33 eventtime=1618979553388763760 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN status" eventtype="Health Check" healthcheck="test" slatargetid=1 oldvalue="2"
   newvalue="1" msg="Number of pass member changed."
   date=2024-01-20 time=21:32:33 eventtime=1618979553388751880 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN status" eventtype="Health Check" healthcheck="test" slatargetid=1 member="1"
   msg="Member status changed. Member out-of-sla."
   ```

   - SLA 目标从不满足目标恢复为满足目标。

   ```bash
   date=2024-01-20 time=21:38:49 eventtime=1618979929908765200 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN status" eventtype="Health Check" healthcheck="test" slatargetid=1 oldvalue="1"
   newvalue="2" msg="Number of pass member changed."
   date=2024-01-20 time=21:38:49 eventtime=1618979929908754060 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
   level="information" vd="root" logdesc="SDWAN status" eventtype="Health Check" healthcheck="test" slatargetid=1
   member="1" msg="Member status changed. Member in sla."
   ```

成员转发状态

1. 某链路带宽占用：

   - 已经达到了配置的该成员的带宽值，停止转发流量。

   ```bash
   date=2024-01-20 time=21:55:14 eventtime=1618980914728863220 tz="-0700" logid="0113022924" type="event" subtype="sdwan"
   level="notice" vd="root" logdesc="SDWAN volume status" eventtype="Volume" interface="R160" member="2" msg="Member
   enters into conservative status with limited ablity to receive new sessions for too much traffic."
   ```

- 占用已经恢复小于配置的带宽值，并继续开始转发流量。

```
date=2024-01-20 time=22:12:52 eventtime=1618981972698753360 tz="-0700" logid="0113022924" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN volume status" eventtype="Volume" interface="R160" member="2" msg="Member
resume normal status to receive new sessions for internal adjustment"
```

2. 配置 SLA 类型（Lowest Cost/Maximize bandwidth）的 SD-WAN 规则，由于 SLA 检查失败，转发成员顺序发生变化。

```
date=2024-01-20 time=22:40:46 eventtime=1618983646428803040 tz="-0700" logid="0113022923" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test" seq="2,1" msg="Service
prioritized by SLA will be redirected in sequence order."
```

3. 配置 Lowest Cost 类型的 SD-WAN 规则，由于 SLA 检查从失败恢复为通过，转发成员顺序发生变化。

```
date=2024-01-20 time=22:41:51 eventtime=1618983711678827920 tz="-0700" logid="0113022923" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test" seq="1,2" msg="Service
prioritized by SLA will be redirected in sequence order."
```

4. 配置 Best Quality 类型的 SD-WAN 规则，转发成员的顺序发生变化。

```
date=2024-01-20 time=22:56:55 eventtime=1618984615708804760 tz="-0700" logid="0113022923" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test" metric="packet-loss"
seq="2,1" msg="Service prioritized by performance metric will be redirected in sequence order."

date=2024-01-20 time=22:56:58 eventtime=1618984618278852140 tz="-0700" logid="0113022923" type="event" subtype="sdwan"
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test" metric="packet-loss"
seq="1,2" msg="Service prioritized by performance metric will be redirected in sequence order."
```

5. 配置 Maximize bandwidth 类型的 SD-WAN 规则：

- 转发成员不满足 SLA 标准，此成员停止转发流量。

```
date=2024-01-20 time=23:10:24 eventtime=1618985425048820800 tz="-0700" logid="0113022923" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test" member="2(R160)"
msg="Service will be load balanced among members with available routing."
```

- 配置 Maximize bandwidth 类型的 SD-WAN 规则，转发成员从不满足 SLA 标准恢复到满足 SLA 标准，此成员可以继续转发流量。

```
date=2024-01-20 time=23:11:34 eventtime=1618985494478807100 tz="-0700" logid="0113022923" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="test"
member="2(R160),1(R150)" msg="Service will be load balanced among members with available routing."
```

6. 在健康检查的配置中开启了 SLA 周期日志：

- 在健康检查的配置中开启了 `sla-fail-log-period`，健康检查周期性产生的 SLA 失败日志。

```
date=2024-01-20 time=23:18:10 eventtime=1618985890469018260 tz="-0700" logid="0113022925" type="event" subtype="sdwan" bash
level="notice" vd="root" logdesc="SDWAN SLA information" eventtype="SLA" healthcheck="test" slatargetid=1
interface="R150" status="up" latency="0.061" jitter="0.004" packetloss="2.000%" inbandwidthavailable="0kbps"
outbandwidthavailable="200.00Mbps" bibandwidthavailable="200.00Mbps" inbandwidthused="1kbps" outbandwidthused="1kbps"
bibandwidthused="2kbps" slamap="0x0" metric="packetloss" msg="Health Check SLA status. SLA failed due to being over
the performance metric threshold."
```

- 在健康检查的配置中开启了 `sla-pass-log-period`，健康检查周期性产生的 SLA 成功日志。

```
date=2024-01-20 time=23:18:12 eventtime=1618985892509027220 tz="-0700" logid="0113022925" type="event" subtype="sdwan" bash
level="information" vd="root" logdesc="SDWAN SLA information" eventtype="SLA" healthcheck="test" slatargetid=1
interface="R150" status="up" latency="0.060" jitter="0.003" packetloss="0.000%" inbandwidthavailable="0kbps"
outbandwidthavailable="200.00Mbps" bibandwidthavailable="200.00Mbps" inbandwidthused="1kbps" outbandwidthused="1kbps"
bibandwidthused="2kbps" slamap="0x1" msg="Health Check SLA status."
```

SD-WAN 调试命令

SD-WAN 健康检查

查看 SD-WAN 健康检查的状态。

```bash
FGT # diagnose sys sdwan health-check
Health Check(server):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.110), jitter(0.024) sla_map=0x0
Seq(2 R160): state(alive), packet-loss(0.000%) latency(0.068), jitter(0.009) sla_map=0x0

Health Check(ping):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.017) sla_map=0x0
Seq(2 R160): state(dead), packet-loss(100.000%) sla_map=0x0

FGT # diagnose sys sdwan health-check ping
Health Check(ping):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.017) sla_map=0x0
Seq(2 R160): state(dead), packet-loss(100.000%) sla_map=0x0
```

SD-WAN 成员状态

1. 使用 `source-ip-based` 或 `source-dest-ip-based` 作为 `load-balance` 的模式时，查看 SD-WAN 成员的状态。

   ```bash
   FGT # diagnose sys sdwan member
   Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 0
   Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 0
   ```

2. 使用 `weight-based` 作为 `load-balance` 的模式时，查看 SD-WAN 成员的状态。

   ```bash
   FGT # diagnose sys sdwan member
   Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 33
     Session count: 15
   Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 66
     Session count: 1
   ```

3. 使用 `measured-volume-based` 作为 `load-balance` 的模式时：

   - 所有的成员均还有使用余量，查看 SD-WAN 成员的状态。

     ```bash
     FGT # diagnose sys sdwan member
     Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 33
       Config volume ratio: 33, last reading: 218067B, volume room 33MB
     Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 66
       Config volume ratio: 66, last reading: 202317B, volume room 66MB
     ```

   - 某个成员的用量已经用尽。

     ```bash
     FGT # diagnose sys sdwan member
     Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 0
       Config volume ratio: 33, last reading: 1287767633B, overload volume 517MB
     Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 63
       Config volume ratio: 66, last reading: 1686997898B, volume room 63MB
     ```

4. 使用 `usage-based` 或``spillover`作为load-balance`的模式时：

   - 当溢出未发生时，查看 SD-WAN 成员的状态。

     ```bash
     FGT # diagnose sys sdwan member
     Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 255
       Egress-spillover-threshold: 400kbit/s, ingress-spillover-threshold: 300kbit/s
       Egress-overbps=0, ingress-overbps=0
     Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 254
       Egress-spillover-threshold: 0kbit/s, ingress-spillover-threshold: 0kbit/s
       Egress-overbps=0, ingress-overbps=0
     ```

   - 当某个成员发生溢出时。

     ```bash
     FGT # diagnose sys sdwan member
     Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 255
       Egress-spillover-threshold: 400kbit/s, ingress-spillover-threshold: 300kbit/s
       Egress-overbps=1, ingress-overbps=0
     Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 254
       Egress-spillover-threshold: 0kbit/s, ingress-spillover-threshold: 0kbit/s
       Egress-overbps=0, ingress-overbps=0
     ```

- `diagnose netlink dstmac list` 命令也可以查看是否发生了溢出情况。

```
FGT # diagnose netlink dstmac list R150
dev=R150 mac=00:00:00:00:00:00 vwl rx_tcp_mss=0 tx_tcp_mss=0 egress_overspill_threshold=50000 egress_bytes=100982
egress_over_bps=1 ingress_overspill_threshold=37500 ingress_bytes=40 ingress_over_bps=0 sampler_rate=0 vwl_zone_id=1
intf_qua=0
```
bash

SD-WAN 规则状态

1. 使用 `manual` 模式的 SD-WAN 规则。

```
FGT # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 R150), alive, selected
    2: Seq_num(2 R160), alive, selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```
bash

2. 使用 `auto` 模式的 SD-WAN 规则。

```
FGT # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(auto), link-cost-factor(latency), link-cost-threshold(10), heath-
check(ping)
  Members(2):
    1: Seq_num(2 R160), alive, latency: 0.066, selected
    2: Seq_num(1 R150), alive, latency: 0.093
  Dst address(1):
        10.100.21.0-10.100.21.255
```
bash

3. 使用 `Priority` 模式（Best Quality）的 SD-WAN 规则。

```
FGT # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(ping)
  Members(2):
    1: Seq_num(2 R160), alive, latency: 0.059, selected
    2: Seq_num(1 R150), alive, latency: 0.077, selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```
bash

4. 使用 `sla` 模式（Lowest Cost）的 SD-WAN 规则。

```
FGT # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 R150), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 R160), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```
bash

5. 使用 `load-balance` 模式（Best Quality）的 SD-WAN 规则。

```
FGT # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=round-robin)
  Members(2):
    1: Seq_num(1 R150), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(2 R160), alive, sla(0x1), gid(2), num of pass(1), selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```
bash

SD-WAN 统计信息

1. 过去 15 分钟的 SD-WAN 接口状态统计日志。

```bash
FGT (root) # diagnose sys sdwan intf-sla-log R150
Timestamp: Wed Apr 21 16:58:27 2021, used inbandwidth: 655bps, used outbandwidth: 81655306bps, used bibandwidth:
81655961bps, tx bys: 3413479982bytes, rx bytes: 207769bytes.
Timestamp: Wed Apr 21 16:58:37 2021, used inbandwidth: 649bps, used outbandwidth: 81655540bps, used bibandwidth:
81656189bps, tx bys: 3515590414bytes, rx bytes: 208529bytes.
Timestamp: Wed Apr 21 16:58:47 2021, used inbandwidth: 655bps, used outbandwidth: 81655546bps, used bibandwidth:
81656201bps, tx bys: 3617700886bytes, rx bytes: 209329bytes.
Timestamp: Wed Apr 21 16:58:57 2021, used inbandwidth: 620bps, used outbandwidth: 81671580bps, used bibandwidth:
81672200bps, tx bys: 3719811318bytes, rx bytes: 210089bytes.
Timestamp: Wed Apr 21 16:59:07 2021, used inbandwidth: 620bps, used outbandwidth: 81671580bps, used bibandwidth:
81672200bps, tx bys: 3821921790bytes, rx bytes: 210889bytes.
Timestamp: Wed Apr 21 16:59:17 2021, used inbandwidth: 665bps, used outbandwidth: 81688152bps, used bibandwidth:
81688817bps, tx bys: 3924030936bytes, rx bytes: 211926bytes.
Timestamp: Wed Apr 21 16:59:27 2021, used inbandwidth: 671bps, used outbandwidth: 81688159bps, used bibandwidth:
81688830bps, tx bys: 4026141408bytes, rx bytes: 212726bytes.
......
```

2. 过去 10 分钟的 SLA 统计日志（`diagnose sys sdwan sla-log <health-check-name> <seq-num> or <health-check-name> <seq-num> <childname>`，如下所示，`1` 为 SD-WAN 成员接口在 SD-WAN 区域中的成员 ID（seq-num）。

```bash
FGT (root) # diagnose sys sdwan sla-log ping 1
Timestamp: Wed Apr 21 17:10:11 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.079, jitter:
0.023, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:12 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.079, jitter:
0.023, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:12 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.081, jitter:
0.024, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:13 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.081, jitter:
0.025, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:13 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.082, jitter:
0.026, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:14 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.083, jitter:
0.026, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:14 2021, vdom root, health-check ping, interface: R150, status: up, latency: 0.084, jitter:
0.026, packet loss: 0.000%.
......
```

SD-WAN 状态信息

1. SD-WAN 规则引用应用控制特征后，应用控制学习的条目状态。

```bash
FGT # diagnose sys sdwan internet-service-app-ctrl-list
Gmail(15817 4294836957): 64.233.191.19 6 443 Thu Apr 22 10:10:34 2021
Gmail(15817 4294836957): 142.250.128.83 6 443 Thu Apr 22 10:06:47 2021
Facebook(15832 4294836806): 69.171.250.35 6 443 Thu Apr 22 10:12:00 2021
Amazon(16492 4294836342): 3.226.60.231 6 443 Thu Apr 22 10:10:57 2021
Amazon(16492 4294836342): 52.46.135.211 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 52.46.141.85 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 52.46.155.13 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 54.82.242.32 6 443 Thu Apr 22 10:10:59 2021
YouTube(31077 4294838537): 74.125.202.138 6 443 Thu Apr 22 10:06:51 2021
YouTube(31077 4294838537): 108.177.121.119 6 443 Thu Apr 22 10:08:24 2021
YouTube(31077 4294838537): 142.250.136.119 6 443 Thu Apr 22 10:02:02 2021
YouTube(31077 4294838537): 142.250.136.132 6 443 Thu Apr 22 10:08:16 2021
YouTube(31077 4294838537): 142.250.148.100 6 443 Thu Apr 22 10:07:28 2021
YouTube(31077 4294838537): 142.250.148.132 6 443 Thu Apr 22 10:10:32 2021
YouTube(31077 4294838537): 172.253.119.91 6 443 Thu Apr 22 10:02:01 2021
YouTube(31077 4294838537): 184.150.64.211 6 443 Thu Apr 22 10:04:36 2021
```

```
YouTube(31077 4294838537): 184.150.168.175 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 184.150.168.211 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 184.150.186.141 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 209.85.145.190 6 443 Thu Apr 22 10:10:36 2021
YouTube(31077 4294838537): 209.85.200.132 6 443 Thu Apr 22 10:02:03 2021
```

2. 查看 IPSec 创建的 Shortcut 隧道的健康检查状态（`diagnose sys link-monitor interface <name> <name>_0`）。

```bash
Spoke1 # diagnose sys link-monitor interface Spoke1_WAN2
Interface(Spoke1_WAN2): state(down, since Fri Jan 19 16:58:13 2024), bandwidth(up:288bps, down:0bps), session
count(IPv4:7, IPv6:0), tx(26595 bytes), rx(21568 bytes).

Spoke1 # diagnose sys link-monitor interface Spoke1_WAN2 Spoke1_WAN2_0
Interface(Spoke1_WAN2_0): state(up, since Fri Jan 19 16:58:08 2024), bandwidth(up:320bps, down:320bps), session
count(IPv4:0, IPv6:0), tx(12360 bytes), rx(12240 bytes), latency(2.15), jitter(0.53), packet-loss(0.00).
```

3. 查看 SD-WAN 中使用的 BGP route-tag。

```bash
FGT # get router info bgp network 10.100.11.0/24
VRF 0 BGP routing table entry for 10.100.11.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   10.100.1.1
  Original VRF 0
  20 10
    10.100.1.1 from 10.100.1.1 (5.5.5.5)
      Origin incomplete metric 0, route tag 15, localpref 100, valid, external, best
      Community: 30:5
      Advertised Path ID: 2
       Last update: Thu Apr 22 10:27:27 2021

  Original VRF 0
  20 10
    10.100.1.5 from 10.100.1.5 (6.6.6.6)
      Origin incomplete metric 0, route tag 15, localpref 100, valid, external, best
      Community: 30:5
      Advertised Path ID: 1
       Last update: Thu Apr 22 10:25:50 2021

FGT # diagnose sys sdwan route-tag-list
Route-tag: 15, address: v4(1), v6(0)Last write/now: 6543391 6566007
        service(1), last read route-tag 15 at 6543420
Prefix(24): Address list(1):
        10.100.11.0-10.100.11.255 oif: 50 48

FGT # diagnose firewall proute list
list route policy info(vf=root):
id=2133196801(0x7f260001) vwl_service=1(DataCenter) vwl_mbr_seq=1 2 dscp_tag=0xff 0xff flags=0x40 order-addr tos=0x00
tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 oif=48(R150) oif=50(R160)
destination(1): 10.100.11.0-10.100.11.255
source wildcard(1): 0.0.0.0/0.0.0.0
hit_count=0 last_used=2021-04-22 10:25:10
```

SD-WAN 介绍

- SD-WAN 数据转发逻辑
- SD-WAN 接口成员组
- SD-WAN 健康检查

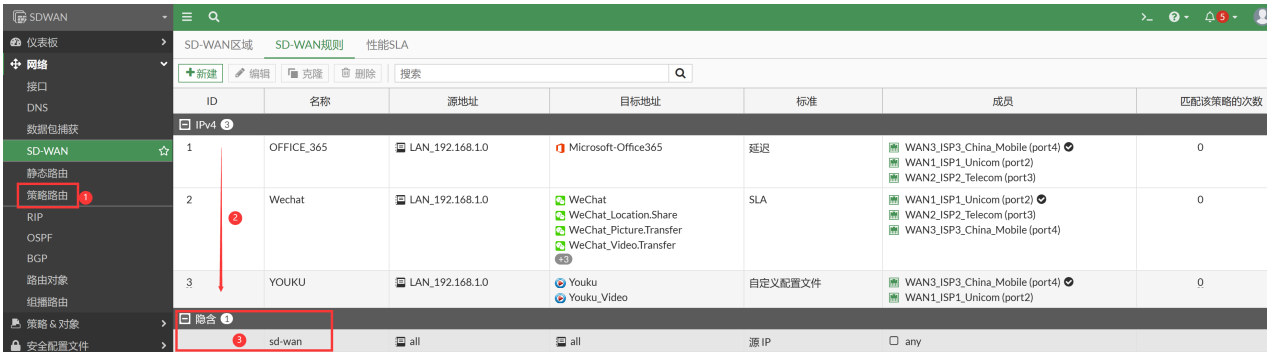
SD-WAN 数据转发逻辑

SD-WAN 规则介绍

1. SD-WAN 规则 = 策略路由 (PBR)，但是同时 SD-WAN rule 规则又比策略路由的优先级低一些。



2. SD-WAN 规则和策略路由一样，也是从上往下按照顺序匹配，可以按照特定的算法将指定的数据流转发到相应的出接口上。



3. 在命令行下查看 SDWAN 规则在策略路由中的条目，SD-WAN 规则本质为策略路由（ID > 7f000001），优先级高于普通 FIB 路由。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2131230721(0x7f080001) vwl_service=1(OFFICE_365) vwl_mbr_seq=3 1 2 dscp_tag=0xff 0xff flags=0x0 tos=0x00
tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=6(port4) oif=4(port2) oif=5(port3)
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=0 last_used=2023-01-02 22:15:18

id=2131230722(0x7f080002) vwl_service=2(Wechat) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) oif=5(port3) oif=6(port4)
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(7): WeChat(4294838375,0,0,0, 33760) WeChat_Location.Share(4294838376,0,0,0, 36743)
WeChat_Picture.Transfer(4294838377,0,0,0, 36730) WeChat_Video.Transfer(4294838378,0,0,0, 36741)
WeChat_Voice.Chat(4294838379,0,0,0, 36742) WeChat_Voip.Call(4294838380,0,0,0, 36648) WeChat_Web(4294838381,0,0,0, 39523)
hit_count=0 last_used=2023-01-02 22:17:55

id=2131230723(0x7f080003) vwl_service=3(YOUKU) vwl_mbr_seq=3 1 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(2) oif=6(port4) oif=4(port2)
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(2): Youku(4294838550,0,0,0, 16531) Youku_Video(4294838551,0,0,0, 35285)
hit_count=0 last_used=2023-01-02 22:20:41
```

4. SD-WAN 规则的最后面存在一条垫底的默认隐含规则，实际的作用就是当没有匹配到上面 SD-WAN 规则的数据时，最终将会匹配到这条隐含的规则，隐含规则的本质就是普通的 FIB 路由表。也就说没有匹配 SD-WAN 规则的数据将会匹配普通路由表进行转发。

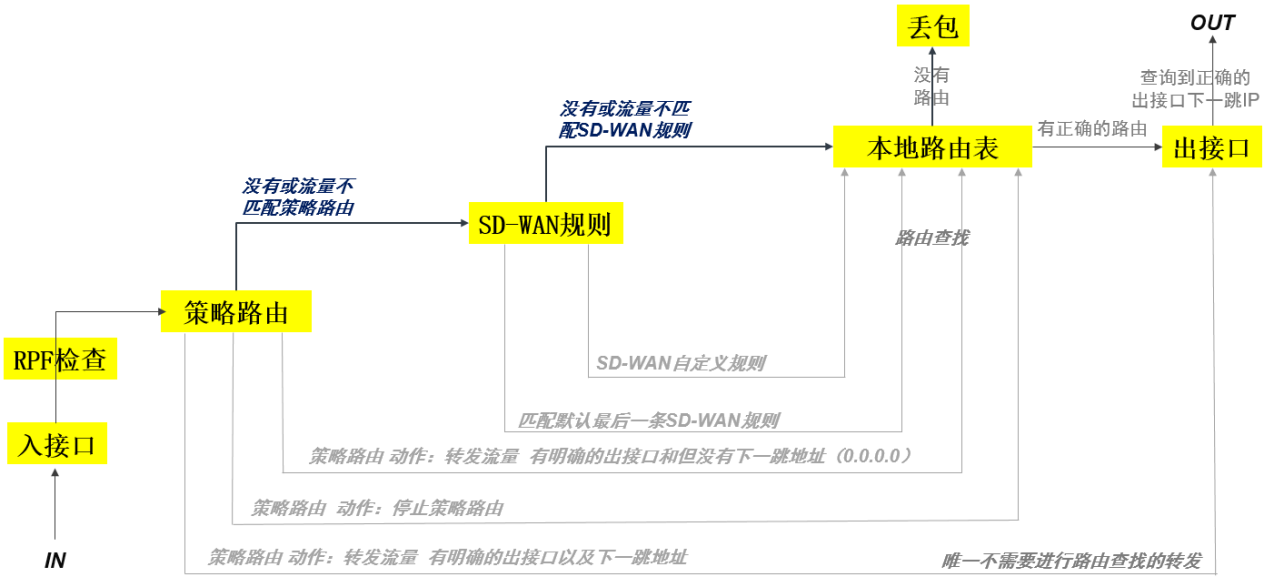


5. 没有匹配 SD-WAN 规则的数据流将查询路由表，在 port2/port3/Port13 三条默认路由出口上进行负载均衡（默认按源 IP 方式负载流量）。

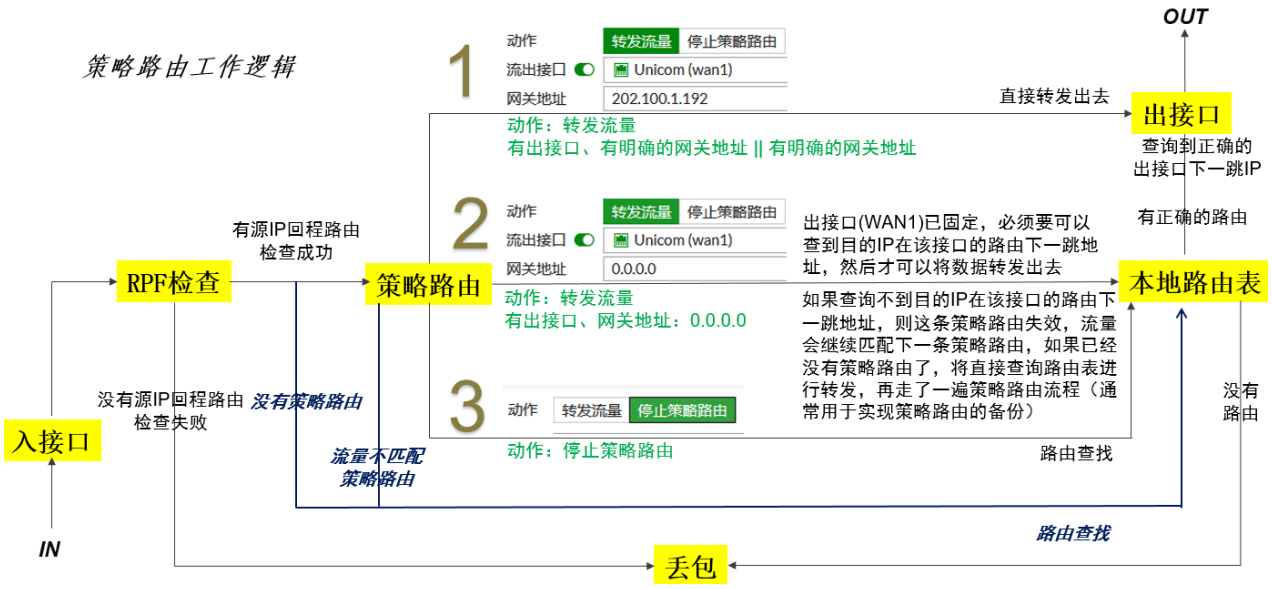
```
SDWAN # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→0.0.0.0/0 pref=0.0.0.0
        gwy=101.100.1.192 flag=04 hops=0 oif=5(port3)
        gwy=111.100.1.192 flag=04 hops=0 oif=6(port4)
        gwy=202.100.1.192 flag=04 hops=0 oif=4(port2)
...

SDWAN # get router info routing-table  all
...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
...
```

SDWAN 工作流程

1. 策略路由、SD-WAN 规则、本地路由表的查找与转发逻辑关系：策略路由 → SD-WAN 规则 → 本地路由。



2. 对比策略路由和 SD-WAN 规则的工作过程（二者有细微的差别，大部分情况下很相似）。



重要

- 默认配置下（在 config system sdwan → config service → edit X 下，set gateway disable + set default disable），在流量匹配 SD-WAN 规则后，被 SD-WAN 规则执行动作前，会查找 FIB 表项（get router info kernel）：
- 如果最佳路由的出接口不是 SD-WAN 成员，则忽略 SD-WAN 规则，直接路由转发。
- 如果目标地址属于直连路由，则忽略 SD-WAN 规则，直接路由转发。
- 如果最佳路由出接口是 SD-WAN 成员，则通过该 SD-WAN 规则找到最佳的出接口转发。
- 开启 gateway 和 default 后（在 config system sdwan → config service → edit X 下，set gateway enable + set default enable），则跳过 FIB 路由查找，直接被 SD-WAN 规则转发。

SDWAN 规则

1. SD-WAN 规则分成两种：
   - 手动自定义的 SD-WAN 规则：低优先级的策略路由（优先级比普通策略路由低），手工添加的 SD-WAN 规则则类似支持多出口的动态选择、支持应用控制、支持路由属性等的增强版策略路由，其转发优先级比普通策略路由低，比普通路由优先级高。
   - 最后一条默认的 SD-WAN 规则：依据 LLB 的负载方式查询本地静态路由（默认源 IP 负载），LLB 负载支持源 IP 负载、溢出、会话权重负载、源目 IP 负载等多种方式，就是旧版本中的 WAN LLB 功能。
2. 手工自定义配置的 SD-WAN 规则相当于是一条只配置了出接口但没有配置下一跳地址（下一跳地址为 0.0.0.0）的策略路由，因此必须要再次查询一遍路由表才可以找到正确的下一跳地址，如果无法查找到该接口的下一跳地址，自定义的 SD-WAN 规则也将失效，由此可见 SD-WAN 规则十分依赖 FIB 路由表。

Debug Flow自定义SD-WAN规则的流量转发过程：

```
diagnose sys session filter proto 1
diagnose sys session filter dst 114.114.114.114
diagnose sys session clear
diagnose debug flow filter addr 114.114.114.114
diagnose debug flow filter proto 1
diagnose debug flow show function-name enable
diagnose debug flow trace start 10
diagnose debug enable
```

```
config system virtual-wan-link
    config service
        edit 2
            set name "114.114.114.114_OUT"
            set mode priority
            set dst "HOST_114.114.114.114"
            set health-check "Unicom"
            set priority-members 1 (WAN1)
        next
```

先match policy routing 找到出接口 via ifindex-7（WAN1）

```
# id=20085 trace_id=11 func=print_pkt_detail line=5320 msg="vd-root:0 received a packet(proto=1, 10.193.1.79:512-
>114.114.114.114:2048) from lan. type=8, code=0, id=512, seq=49585."
id=20085 trace_id=11 func=init_ip_session_common line=5480 msg="allocate a new session-00382b05"
id=20085 trace_id=11 func=vf_ip_route_input_common line=2578 msg="Match policy routing: to 114.114.114.114 via ifindex-7"
id=20085 trace_id=11 func=vf_ip_route_input_common line=2590 msg="find a route: flag=04000000 gw-202.100.1.192 via wan1"
id=20085 trace_id=11 func=fw_forward_handler line=749 msg="Allowed by Policy-1: SNAT"
id=20085 trace_id=11 func=ids_receive line=282 msg="send to ips"
id=20085 trace_id=11 func=__ip_session_run_tuple line=3226 msg="SNAT 10.193.1.79->202.100.1.25:60928"
id=20085 trace_id=12 func=print_pkt_detail line=5320 msg="vd-root:0 received a packet(proto=1, 114.114.114.114:60928-
>202.100.1.25:0) from wan1. type=0, code=0, id=60928, seq=49585."
id=20085 trace_id=12 func=resolve_ip_tuple_fast line=5395 msg="Find an existing session, id-00382b05, reply direction"
id=20085 trace_id=12 func=__ip_session_run_tuple line=3240 msg="DNAT 202.100.1.25:0->10.193.1.79:512"
id=20085 trace_id=12 func=vf_ip_route_input_common line=2590 msg="find a route: flag=00000000 gw-10.193.1.79 via lan"
id=20085 trace_id=12 func=ids_receive line=282 msg="send to ips"
```

然后再 find a route 找到下一跳地址 gw-202.100.1.192

3. SD-WAN 规则这样的处理方式，带来的最大的好处是可以实现 SD-WAN 出口故障的快速切换，SD-WAN 的健康检查完全依赖于这种处理方式。一旦健康检查失败，出接口的静态路由将会消失，SD-WAN 将无法在路由表中查找到下一跳地址，因此相应的 SD-WAN 规则也跟随着会失效，从而实现了 SD-WAN 的多出口故障的自动切换。

> **重要**
>
> SD-WAN 的路由必不可少，否则 SD-WAN 规则将无法使用。

4. 必须要配置的内容：手工配置默认路由或明细路由指向 SD-WAN 出口（默认源 IP 负载），或者使用动态路由协议（OSPF/BGP/RIP）学习到相应的路由，总之路由表中一定要有去往 SD-WAN 规则中的目的 IP 的路由才可以。

5. 可以根据需要配置到具体 port2/port3/port4 分别的路由，也可以直接配置指向 SD-WAN 接口的路由。



（网关在静态路由中配置）

或：



（网关在 SDWAN 成员中配置）

6. 如果 SD-WAN 是针对去往 Internet 的负载：则必须配置默认路由指向 SD-WAN；如果 SD-WAN 是针对去往 VPN 业务网段的负载：则必须配置 VPN 明细路由指向 SD-WAN。默认情况下路由表中需要存在 SD-WAN 接口的路由，这样 SD-WAN 规则才可以生效。



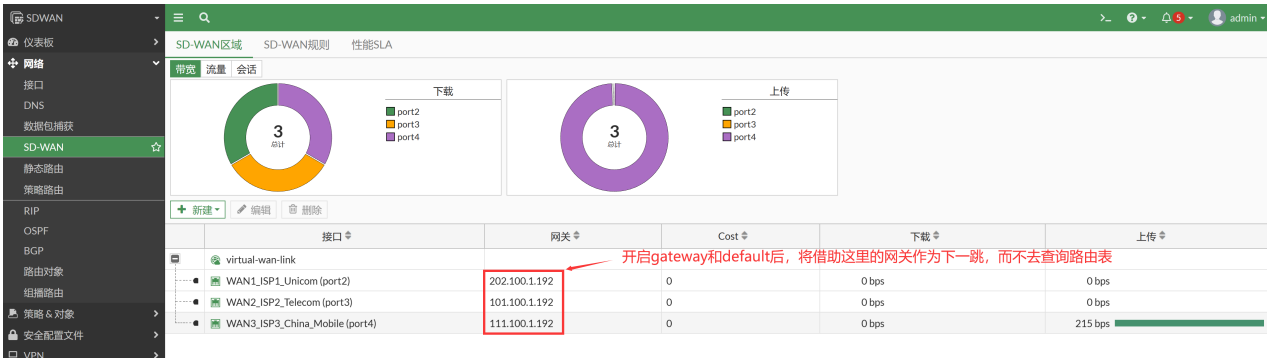
7. 总结来看：默认情况下， SD-WAN 规则由于只能选择出接口而无法直接配置下一跳的 IP 地址（策略路由可以配置出接口和下一跳 IP），因此需要继续通过 FIB 路由表的查询以便找到正确的下一跳 IP 地址，才可以将数据转发出去，可以看出 SD-WAN 规则十分的依赖路由转发表，几乎 SD-WAN 规则就是依赖 FIB 路由表而工作的，二者是相互依存的关系。

8. 那么是否可以打破这样的依赖关系呢？是否可以和策略路由一样，SD-WAN 规则也可以有一个明确的出接口和下一跳 IP，而不需要依赖 FIB 路由表而找到下一跳 IP 呢？答案是：有的，虽然我们不推荐这样做，这样不利于多出接口的快速切换。具体的做法如下：

```
SDWAN # config system sdwan
SDWAN (sdwan) # config members
SDWAN (members) # show
config members
    edit 1
        set interface "port2"
        set gateway 202.100.1.192
    next
    edit 2
        set interface "port3"
        set gateway 101.100.1.192
    next
    edit 3
        set interface "port4"
        set gateway 111.100.1.192
    next
end
SDWAN (members) # end

SDWAN (sdwan) # config service
SDWAN (service) # edit 1
SDWAN (1) # show full-configuration
config service
    edit 1
        set name "OFFICE_365"
        set addr-mode ipv4
        set input-device-negate disable
        set mode priority
        set role standalone
        set standalone-action disable
        set tos 0x00
        set tos-mask 0x00
        set src "LAN_192.168.1.0"
        set src-negate disable
        set internet-service enable
        set internet-service-name "Microsoft-Office365"
        set health-check "Default_DNS"
        set link-cost-factor latency
        set link-cost-threshold 10
        set hold-down-time 0
        set dscp-forward disable
        set dscp-reverse disable
        set priority-members 3 1 2
        set status enable
        ***set gateway disable***
        ***set default disable***
        set tie-break zone
        set use-shortcut-sla enable
        set passive-measurement disable
    next
end
```

如果将 SD-WAN 规则下面的 gateway 和 default 默认状态有 disable 改成 enable，则 SD-WAN 规则将跳过 FIB 的查询，直接选择 members 接口里面配置的下一跳 IP 作为 SDWAN 规则（策略路由）的下一跳 IP，直接转发流量，而不再查询 FIB 路由表了。

```
SDWAN # config system sdwan
SDWAN (sdwan) # config service
SDWAN (service) # edit 1
SDWAN (1) # set gateway enable
SDWAN (1) # set default enable
SDWAN (1) # end
SDWAN (sdwan) # end
```



重要

这种使用场景比较特别，需要将 SD-WAN 流量固定地从某个出口和下一跳 IP 转发，而不受到底层 FIB 路由表路由切换的影响（健康检查切换/link-monitor 等等），这种使用场景比较少用到。

9. 默认配置下，本机发出的报文不受 SDWAN 规则的影响。

总结

1. SD-WAN 规则可以认为是调用了多个有序出接口（oif）列表的策略路由，但是优先级比普通的策略路由要低一些，oif 在策略路由中从左（最优先）到右（最不优先）进行排序。

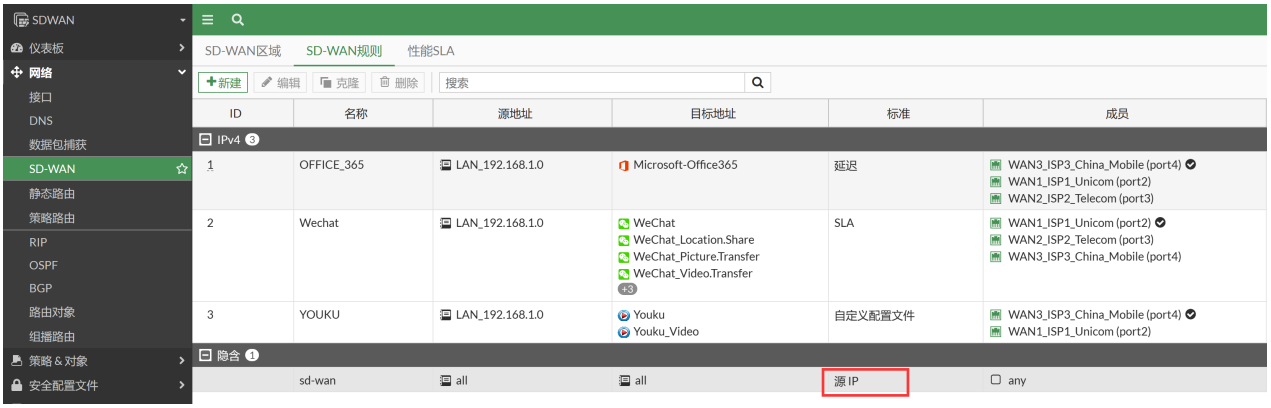
```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2098397185(0x7d130001) vwl_service=1(OFFICE_365) vwl_mbr_seq=3 1 2 dscp_tag=0xff 0xff flags=0x0 tos=0x00
tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=6(port4) gwy=111.100.1.192 oif=4(port2)
gwy=202.100.1.192 oif=5(port3) gwy=101.100.1.192 default
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=0 last_used=2023-01-02 23:10:21
```

2. SD-WAN 规则的类别（Best Quality, Lowest Cost, Maximize Bandwidth）决定了当 SLA 检查值（延迟、抖动、丢失）发生变化时，接口列表中的出接口（oif）顺序是如何变化的。

3. 默认情况下，FIB 路由表中必须有去往目的 IP 的路由，这样 SD-WAN 规则才可以生效，SD-WAN 规则的出接口的下一跳是通过 FIB 路由表查询获取的，因此 SD-WAN 目的网段的路由非常重要。

```
SDWAN # get router info routing-table  all
 ...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
 ...
```

4. 如果没有匹配的 SD-WAN 的策略路由，则路由的决策会基于 FIB 进行查找（默认是源 IP 方式的 ECMP）。




```
# config system sdwan
# set load-balance-mode ?
source-ip-based          Source IP load balancing. All traffic from a source IP is sent to the same interface.

weight-based             Weight-based load balancing. Interfaces with higher weights have higher priority and get more
traffic.

usage-based              Usage-based load balancing. All traffic is sent to the first interface on the list. When the
bandwidth on that interface exceeds the spill-over limit new traffic is sent to the next interface.

source-dest-ip-based     Source and destination IP load balancing. All traffic from a source IP to a destination IP is
sent to the same interface.

measured-volume-based    Volume-based load balancing. Traffic is load balanced based on traffic volume (in bytes). More
traffic is sent to interfaces with higher volume ratios.
```

5. 路由（或 SD-WAN 转发顺序）发生改变后（比如出接口列表重新排序、健康检查失败触发、动态路由协议更新），会话将重新上送 CPU 进行匹配处理，重新查询策略和路由，这样流量可能会迁移到另一个 SD-WAN 成员接口上（具体详见网络管理 → SD-WAN → SD-WAN 介绍 → SD-WAN 健康状态检查↗章节）。

6. 有 NAT 的会话的流量不会随着路由的变化而改变流量的出接口（默认情况下），只有纯三层的流量转发才会正常的进行出接口切换，如下命令行决定了以上特性。

> **重要**
>
> 默认配置下的 SNAT 场景，即使转发流量的 SD-WAN 成员的健康检查变为 dead，只要该线路仍然存在路由（可能关闭了 `update-static-route`），已有的旧会话仍然不会切换到其他线路。

```
SDWAN # config system global
SDWAN (global) # show full-configuration | grep snat
    set snat-route-change disable
```

7. 没有 NAT 的流量，在接口下设置会话保持（preserve-session-route）可以允许在路由更改后强制将会话保持在同一个 sd-wan 成员上，不跟随这出接口的变化而变化已有流量会话的出接口信息。

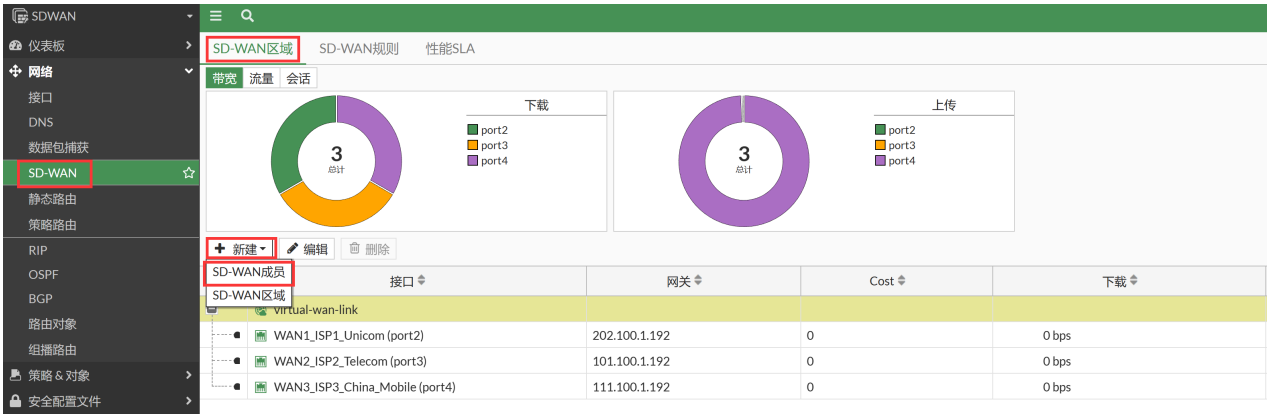
```
SDWAN # config system interface
SDWAN (interface) # edit port2
SDWAN (port2) # set preserve-session-route enable
SDWAN (port2) # end
```

SD-WAN 接口成员组

静态 IP 接口成员配置

1. 进入网络 → SD-WAN → SD-WAN 区域，选择默认的 SD-WAN 区域（virtual-wan-link）或新建 SD-WAN 区域。

2. 点击新建 → SD-WAN 成员，将想要加入 SD-WAN 成员的接口依次加入 SD-WAN 区域。



3. 对应 CLI：

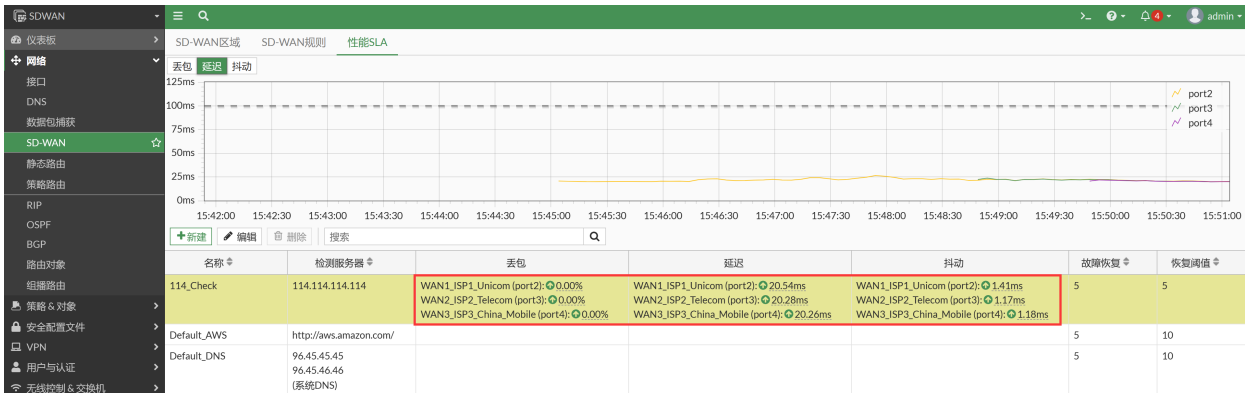
```
SDWAN # config system sdwan
SDWAN (sdwan) # config members
SDWAN (members) # show
config members
    edit 1
        set interface "port2"
        set zone "virtual-wan-link"
        set gateway 202.100.1.192
        set source 0.0.0.0
        set gateway6 ::
        set source6 ::
        set cost 0
        set priority 1
        set priority6 1024
        set status enable
        set comment ''
    next
    edit 2
        set interface "port3"
        set zone "virtual-wan-link"
        set gateway 101.100.1.192
        set source 0.0.0.0
        set gateway6 ::
        set source6 ::
        set cost 0
        set priority 1
        set priority6 1024
        set status enable
        set comment ''
    next
    edit 3
        set interface "port4"
        set zone "virtual-wan-link"
        set gateway 111.100.1.192
        set source 0.0.0.0
        set gateway6 ::
        set source6 ::
        set cost 0
        set priority 1
```

```
        set priority6 1024
        set status enable
        set comment ''
    next
end
```

参数说明

1. set gateway 202.100.1.192：

- 作用 1：这个参数会被用在 SD-WAN 健康检查目的 IP 的下一跳地址，比如我们检测 114.114.114.114 的 SD-WAN 健康检查，实际后台会在内核下发去往 114.114.114.114 的明细路由，以便 port2/port3/Port13 可以各自发起去往 114.114.114.114 的探测报文。




```
SDWAN # get router info kernel | grep 114.114.114.114
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 111.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=111.100.1.192 dev=6(port4)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 101.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=101.100.1.192 dev=5(port3)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 202.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=202.100.1.192 dev=4(port2)

# get router info routing-table  all | grep 114.114.114.114
```

路由表是看不到 114.114.114.114 的路由的，而这个 kernel 路由的 gwy 地址就是在 sd-wan 接口成员组中所配置的 gateway 所得来的。

- 作用 2：配置 SD-WAN 的路由的时候，也会借用 SD-WAN 接口组的网关作为 SD-WAN 路由的下一跳 IP。



对应 CLI：

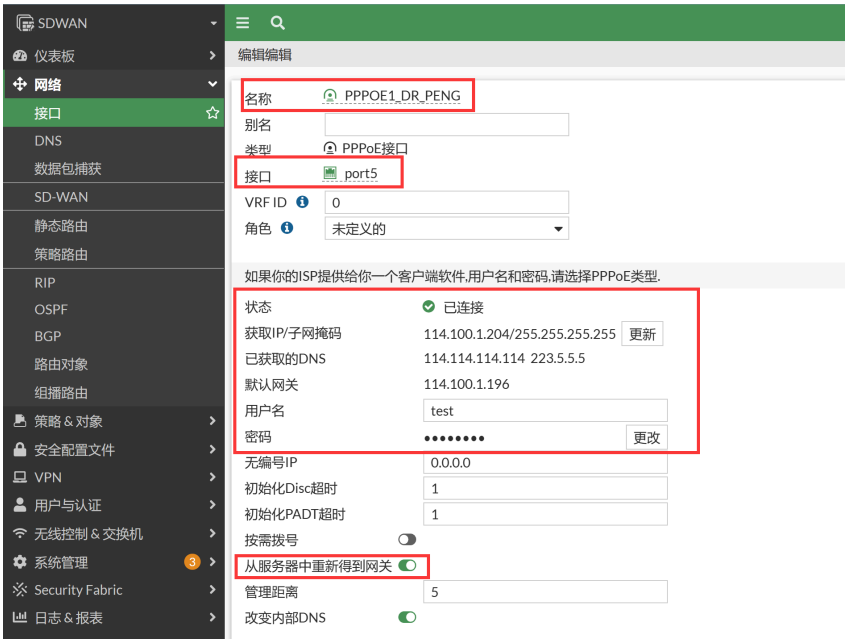
```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

静态路由借用下一跳 SD-WAN 接口成员组里下一跳 IP 的效果：

```
SDWAN # get router info routing-table all
...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
...
```

动态 IP 接口成员配置

1. 如果是 PPPOE/DHCP 这种动态接口，在 SD-WAN 接口组的时候，通常需要保持 set gateway 0.0.0.0，SD-WAN 的路由会自动去查找到 PPPOE/DHCP 接口的动态的默认网关 IP 地址。





2. SD-WAN 接口成员中的 PPPOE 接口网关配置为动态，由于 PPPOE 接口下面的默认参数：defaultgw 为 enable 状态，因此 SD-WAN 接口成员里面会自动去迭代查询到 PPPOE 默认路由的真实下一跳 IP 地址。

```
SDWAN # config system interface
SDWAN (interface) # edit PPPOE1_DR_PENG
SDWAN (PPPOE1_DR_PENG) # show
config system interface
    edit "PPPOE1_DR_PENG"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set snmp-index 13
        set interface "port5"
    next
end

SDWAN (PPPOE1_DR_PENG) # show full-configuration | grep default
        set defaultgw enable    //默认为enable状态
```

3. 效果：SD-WAN 健康检查的内核路由下发会找到 PPP 接口默认路由的真实下一跳 IP。

```
SDWAN # get router info kernel  | grep 114.114.114.114
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 114.100.1.204/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=114.100.1.196 dev=19(PPP)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 111.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=111.100.1.192 dev=6(port4)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 101.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=101.100.1.192 dev=5(port3)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 202.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=202.100.1.192 dev=4(port2)
```

4. SD-WAN PPPOE 接口的静态路由也会自动找到 PPP 接口默认路由的真实下一跳 IP。

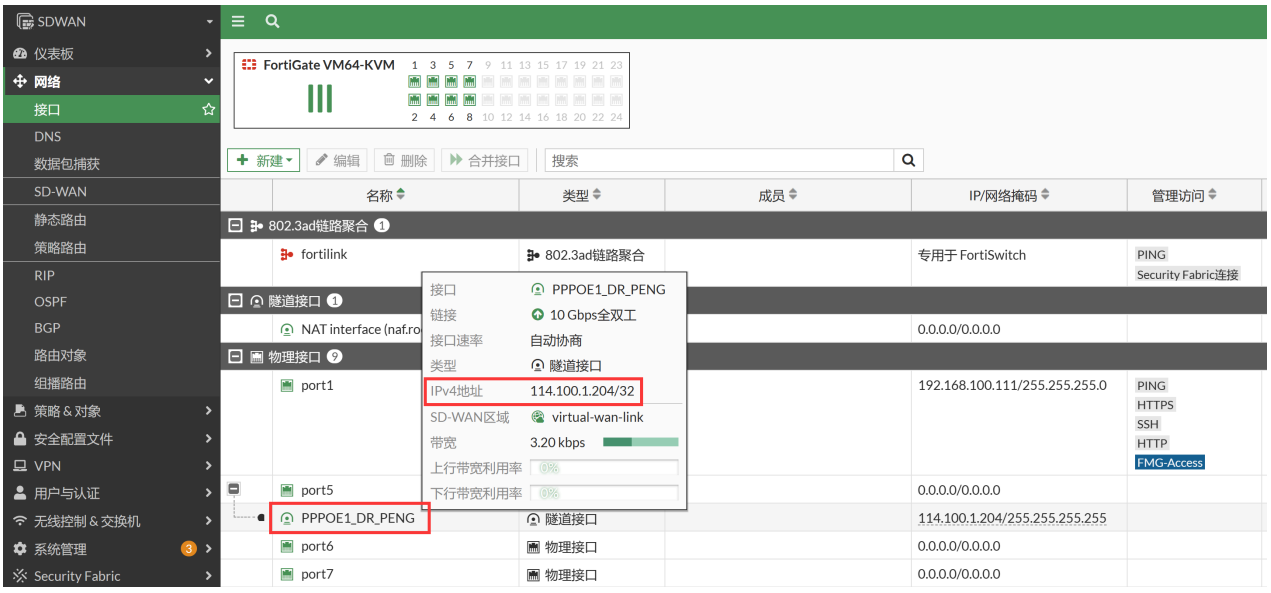
```
SDWAN # get router info routing-table all
...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
...
```

5. 补充：PPPOE 接口的推荐配置方式（只能命令行配置）。

```
config system pppoe-interface
    edit "PPPOE1_DR_PENG"
        set device "port5"
        set username "test"
        set password ENC
yPSGPPFbTeo7o9/FjjzgRe9vBUUkYXXXHxKlvsLj3Gk9cgS88ik8ifl5yhclIvjJ0XRAJGW5A4PqL3nczbmg/FgLTJCad4OmtjBuly1fjdwcdODr=
    next
end
```

然后会自动产生“PPPOE1_DR_PENG”接口：

```
config system interface
    edit "PPPOE1_DR_PENG"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set snmp-index 13
        set interface "port5"
    next
end

SDWAN (PPPOE1_DR_PENG) # show full-configuration | grep def
        set defaultgw enable
```



参数说明

1. `set source 0.0.0.0`：SD-WAN 健康监控的源 IP 会借用 set source 里面设置的 IP，默认为 0.0.0.0，则是选择 SD-WAN 接口的主 IP 作为源 IP 进行发送健康检查探测报文。

   > **重要**
   >
   > IPsec VPN 隧道、GRE 隧道等 VPN 接口默认情况下接口是没有 IP 地址的，进行 SD-WAN 健康监控探测的时候默认是没有办法发送探测报文的，这种情况下监控检测会失败的。
   >
   > 解决办法：
   >
   > - 给 VPN 隧道接口配置上 IP 地址（推荐）。
   > - 在 SD-WAN 接口成员配置的时候，命令行添加 set source ip，手工给没有 IP 的隧道指定一个 IP 地址，用于 sd-wan 监控检测发送探测包的源 IP 地址。

2. `set cost 0`：用在“Lowest Cost & Maximize Bandwidth (SLA) rules”里面。用于 SD-WAN 算法中的出接口选择使用。具体可见网络管理 → SD-WAN → SD-WAN 介绍 → SD-WAN 健康状态检查章节。

3. `set priority 0`：和静态路由中的 priority 参数一致，用于浮动静态路由时候，让备份的路由也存在于路由表，但是不用于出接口流量的路由转发，效果如下：

```
SDWAN # config system sdwan
SDWAN (sdwan) # config members
SDWAN (members) # show
config members
    edit 1
        set interface "port2"
        set gateway 202.100.1.192
    next
    edit 2
        set interface "port3"
        set gateway 101.100.1.192
    next
    edit 3
        set interface "port4"
        set gateway 111.100.1.192
    next
    edit 4
        set interface "PPPOE1_DR_PENG"
    next
end

SDWAN (members) # edit 2
SDWAN (2) # set priority 100
SDWAN (2) # next

SDWAN (members) # show
config members
    edit 1
        set interface "port2"
        set gateway 202.100.1.192
    next
    edit 2
        set interface "port3"
        set gateway 101.100.1.192
        set priority 100
    next
    edit 3
        set interface "port4"
        set gateway 111.100.1.192
    next
    edit 4
        set interface "PPPOE1_DR_PENG"
```

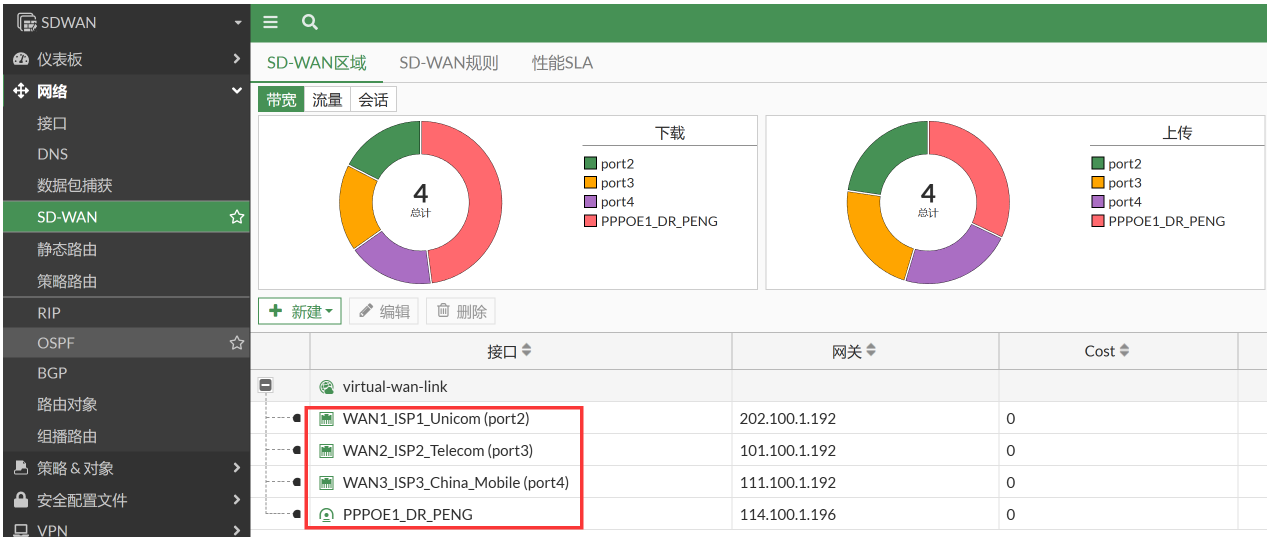
```
    next
end

SDWAN (members) # end
SDWAN (sdwan) # end
```

当没匹配到 SD-WAN 自定义规则（策略路由），同时有存在 SD-WAN 的默认路由的时候，就会去匹配 SD-WAN 的负载路由（最后一条隐藏默认 SD-WAN 规则），这时候 priority 参数会生效（priority 效果和普通路由一致）。port3 的路由 priority 是 100，此路由存在于路由表中，但是不用于出方向的路由转发（策略路由可以使用此路由），仅仅存在路由表中，以便入方向的 RPF（反向路径检测）可以正常工作。

```
SDWAN # get router info routing-table all
 ...
S*      0.0.0.0/0 [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
                  [1/0] via 101.100.1.192, port3, [100/0]     //priority为100，对于出方向的策略路由是生效的，不用于出方向的FIB路由的转发。可以解决入方向的RPF问题
 ...
```

成员序号与接口 index 的关系

1. 查看 SD-WAN 成员的序号。




```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port2"
            set gateway 202.100.1.192
        next
        edit 2
            set interface "port3"
            set gateway 101.100.1.192
            set priority 100
        next
        edit 3
            set interface "port4"
            set gateway 111.100.1.192
        next
        edit 4
            set interface "PPPOE1_DR_PENG"
        next
    end
```

```
SDWAN # diagnose sys sdwan member    //SD-WAN接口组成员
Member(1): interface: port2, flags=0×0 , gateway: 202.100.1.192, priority: 1 1024, weight: 0
Member(2): interface: port3, flags=0×0 , gateway: 101.100.1.192, priority: 100 1024, weight: 0
Member(3): interface: port4, flags=0×0 , gateway: 111.100.1.192, priority: 1 1024, weight: 0
Member(4): interface: PPPOE1_DR_PENG, flags=0×8 , gateway: 114.100.1.196, priority: 1 1024, weight: 0
```

2. 在 SD-WAN SLA 监控中查看 SD-WAN 成员序号。

```
SDWAN # diagnose sys sdwan health-check status    //SD-WAN健康检查
Health Check(114_Check):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(20.472), jitter(0.790), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(20.289), jitter(0.750), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(20.199), jitter(0.746), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(4 PPPOE1_DR_PENG): state(alive), packet-loss(0.000%) latency(20.371), jitter(0.861), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0×1
```

3. 在 SD-WAN 规则中查看 SD-WAN 成员序号。

```
SDWAN # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0×260 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(114_Check)
  Members(4):
    1: Seq_num(1 port2), alive, latency: 21.736, selected
    2: Seq_num(4 PPPOE1_DR_PENG), alive, latency: 21.599, selected
    3: Seq_num(2 port3), alive, latency: 21.524, selected
    4: Seq_num(3 port4), alive, latency: 21.507, selected
  Internet Service(1): Microsoft-Office365(327782,0,0,0)
  Src address(1):
        192.168.1.0-192.168.1.255
```

4. 查看策略路由中对应的 service 1 的条目，记录其中的 oif 参数。

```
SDWAN # diag firewall proute list
list route policy info(vf=root):

id=2097676289(0×7d080001) vwl_service=1(OFFICE_365) vwl_mbr_seq=1 4 2 3 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(4) oif=4(port2) gwy=202.100.1.192
oif=19(PPPOE1_DR_PENG) gwy=114.100.1.196 ot
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=0 last_used=2023-01-03 00:00:13
```

5. 查看接口索引与具体接口的对应关系。

```
SDWAN # diag ip address list
...
IP=202.100.1.10→202.100.1.10/255.255.255.0 index=4 devname=port2
IP=101.100.1.10→101.100.1.10/255.255.255.0 index=5 devname=port3
IP=111.100.1.10→111.100.1.10/255.255.255.0 index=6 devname=port4
...
IP=114.100.1.204→114.100.1.196/255.255.255.255 index=19 devname=PPPOE1_DR_PENG
```

6. 综上所述可以得出 SD-WAN 接口组成员序号与各个功能模块里面的接口调用关系。

```
接口索引 4      --- 对应 port2 ---对应SD-WAN接口成员组序号1
接口索引 5     --- 对应 port3 ---对应SD-WAN接口成员组序号2
接口索引 6   --- 对应 port4 ---对应SD-WAN接口成员组序号3
接口索引 19   --- 对应 PPPOE1_DR_PENG ---对应SD-WAN接口成员组序号4
```

其他

1. 统计 SD-WAN 成员接口最近 10 分钟流量的命令。

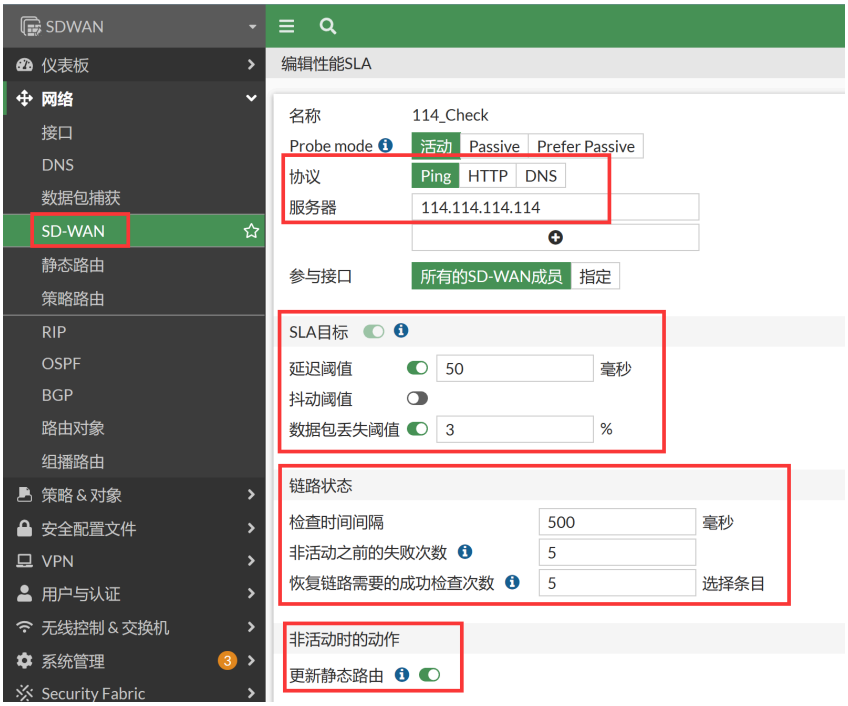
```
SDWAN # diagnose sys sdwan intf-sla-log port2
Timestamp: Tue Jan  3 01:04:04 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 422346bytes, rx bytes: 422106bytes.
Timestamp: Tue Jan  3 01:04:14 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 423468bytes, rx bytes: 423228bytes.
Timestamp: Tue Jan  3 01:04:24 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 424590bytes, rx bytes: 424350bytes.
Timestamp: Tue Jan  3 01:04:34 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 425712bytes, rx bytes: 425472bytes.
Timestamp: Tue Jan  3 01:04:44 2023, used inbandwidth: 875bps, used outbandwidth: 884bps, used bibandwidth: 1759bps, tx
bytes: 426792bytes, rx bytes: 426498bytes.
Timestamp: Tue Jan  3 01:04:54 2023, used inbandwidth: 882bps, used outbandwidth: 882bps, used bibandwidth: 1764bps, tx
bytes: 427860bytes, rx bytes: 427620bytes.
Timestamp: Tue Jan  3 01:05:04 2023, used inbandwidth: 875bps, used outbandwidth: 875bps, used bibandwidth: 1750bps, tx
bytes: 428940bytes, rx bytes: 428700bytes.
Timestamp: Tue Jan  3 01:05:14 2023, used inbandwidth: 882bps, used outbandwidth: 882bps, used bibandwidth: 1764bps, tx
bytes: 430104bytes, rx bytes: 429864bytes.
Timestamp: Tue Jan  3 01:05:24 2023, used inbandwidth: 875bps, used outbandwidth: 875bps, used bibandwidth: 1750bps, tx
bytes: 431184bytes, rx bytes: 430944bytes.
Timestamp: Tue Jan  3 01:05:34 2023, used inbandwidth: 890bps, used outbandwidth: 882bps, used bibandwidth: 1772bps, tx
bytes: 432306bytes, rx bytes: 432066bytes.
Timestamp: Tue Jan  3 01:05:44 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 433386bytes, rx bytes: 433146bytes.
Timestamp: Tue Jan  3 01:05:54 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 434508bytes, rx bytes: 434268bytes.
Timestamp: Tue Jan  3 01:06:04 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 435630bytes, rx bytes: 435390bytes.
Timestamp: Tue Jan  3 01:06:14 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 436752bytes, rx bytes: 436512bytes.
Timestamp: Tue Jan  3 01:06:24 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 437832bytes, rx bytes: 437592bytes.
Timestamp: Tue Jan  3 01:06:34 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 438954bytes, rx bytes: 438714bytes.
Timestamp: Tue Jan  3 01:06:44 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 440034bytes, rx bytes: 439794bytes.
Timestamp: Tue Jan  3 01:06:54 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 441198bytes, rx bytes: 440958bytes.
Timestamp: Tue Jan  3 01:07:04 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 442278bytes, rx bytes: 442038bytes.
Timestamp: Tue Jan  3 01:07:14 2023, used inbandwidth: 890bps, used outbandwidth: 892bps, used bibandwidth: 1782bps, tx
bytes: 443412bytes, rx bytes: 443160bytes.
Timestamp: Tue Jan  3 01:07:24 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 444480bytes, rx bytes: 444240bytes.
Timestamp: Tue Jan  3 01:07:34 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 445560bytes, rx bytes: 445320bytes.
Timestamp: Tue Jan  3 01:07:44 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 446724bytes, rx bytes: 446484bytes.
Timestamp: Tue Jan  3 01:07:54 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 447804bytes, rx bytes: 447564bytes.
Timestamp: Tue Jan  3 01:08:04 2023, used inbandwidth: 890bps, used outbandwidth: 888bps, used bibandwidth: 1778bps, tx
bytes: 448968bytes, rx bytes: 448728bytes.
Timestamp: Tue Jan  3 01:08:14 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 450048bytes, rx bytes: 449808bytes.
Timestamp: Tue Jan  3 01:08:24 2023, used inbandwidth: 904bps, used outbandwidth: 904bps, used bibandwidth: 1808bps, tx
bytes: 451212bytes, rx bytes: 450972bytes.
Timestamp: Tue Jan  3 01:08:34 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 452292bytes, rx bytes: 452052bytes.
Timestamp: Tue Jan  3 01:08:44 2023, used inbandwidth: 897bps, used outbandwidth: 897bps, used bibandwidth: 1794bps, tx
bytes: 453414bytes, rx bytes: 453174bytes.
Timestamp: Tue Jan  3 01:08:54 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 454536bytes, rx bytes: 454296bytes.
Timestamp: Tue Jan  3 01:09:04 2023, used inbandwidth: 897bps, used outbandwidth: 897bps, used bibandwidth: 1794bps, tx
bytes: 455658bytes, rx bytes: 455418bytes.
Timestamp: Tue Jan  3 01:09:14 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 456738bytes, rx bytes: 456498bytes.
Timestamp: Tue Jan  3 01:09:24 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 457860bytes, rx bytes: 457620bytes.
Timestamp: Tue Jan  3 01:09:34 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 458982bytes, rx bytes: 458742bytes.
Timestamp: Tue Jan  3 01:09:44 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 460062bytes, rx bytes: 459822bytes.
```

```
Timestamp: Tue Jan  3 01:09:54 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 461226bytes, rx bytes: 460986bytes.
Timestamp: Tue Jan  3 01:10:04 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 462306bytes, rx bytes: 462066bytes.
Timestamp: Tue Jan  3 01:10:14 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 463428bytes, rx bytes: 463188bytes.
Timestamp: Tue Jan  3 01:10:24 2023, used inbandwidth: 882bps, used outbandwidth: 882bps, used bibandwidth: 1764bps, tx
bytes: 464496bytes, rx bytes: 464256bytes.
Timestamp: Tue Jan  3 01:10:34 2023, used inbandwidth: 880bps, used outbandwidth: 888bps, used bibandwidth: 1768bps, tx
bytes: 465618bytes, rx bytes: 465324bytes.
Timestamp: Tue Jan  3 01:10:44 2023, used inbandwidth: 866bps, used outbandwidth: 875bps, used bibandwidth: 1741bps, tx
bytes: 466698bytes, rx bytes: 466404bytes.
Timestamp: Tue Jan  3 01:10:54 2023, used inbandwidth: 873bps, used outbandwidth: 882bps, used bibandwidth: 1755bps, tx
bytes: 467820bytes, rx bytes: 467526bytes.
Timestamp: Tue Jan  3 01:11:04 2023, used inbandwidth: 873bps, used outbandwidth: 882bps, used bibandwidth: 1755bps, tx
bytes: 468942bytes, rx bytes: 468648bytes.
Timestamp: Tue Jan  3 01:11:14 2023, used inbandwidth: 873bps, used outbandwidth: 890bps, used bibandwidth: 1763bps, tx
bytes: 470064bytes, rx bytes: 469716bytes.
Timestamp: Tue Jan  3 01:11:24 2023, used inbandwidth: 875bps, used outbandwidth: 884bps, used bibandwidth: 1759bps, tx
bytes: 471144bytes, rx bytes: 470796bytes.
Timestamp: Tue Jan  3 01:11:34 2023, used inbandwidth: 888bps, used outbandwidth: 897bps, used bibandwidth: 1785bps, tx
bytes: 472308bytes, rx bytes: 471960bytes.
Timestamp: Tue Jan  3 01:11:44 2023, used inbandwidth: 882bps, used outbandwidth: 890bps, used bibandwidth: 1772bps, tx
bytes: 473388bytes, rx bytes: 473040bytes.
Timestamp: Tue Jan  3 01:11:54 2023, used inbandwidth: 882bps, used outbandwidth: 890bps, used bibandwidth: 1772bps, tx
bytes: 474510bytes, rx bytes: 474162bytes.
Timestamp: Tue Jan  3 01:12:04 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 475632bytes, rx bytes: 475284bytes.
Timestamp: Tue Jan  3 01:12:14 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 476712bytes, rx bytes: 476364bytes.
Timestamp: Tue Jan  3 01:12:24 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 477834bytes, rx bytes: 477486bytes.
Timestamp: Tue Jan  3 01:12:34 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 478956bytes, rx bytes: 478608bytes.
Timestamp: Tue Jan  3 01:12:44 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 480078bytes, rx bytes: 479730bytes.
Timestamp: Tue Jan  3 01:12:54 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 481158bytes, rx bytes: 480810bytes.
Timestamp: Tue Jan  3 01:13:04 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 482280bytes, rx bytes: 481932bytes.
Timestamp: Tue Jan  3 01:13:14 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 483402bytes, rx bytes: 483054bytes.
Timestamp: Tue Jan  3 01:13:24 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 484482bytes, rx bytes: 484134bytes.
Timestamp: Tue Jan  3 01:13:34 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 485604bytes, rx bytes: 485256bytes.
Timestamp: Tue Jan  3 01:13:44 2023, used inbandwidth: 890bps, used outbandwidth: 892bps, used bibandwidth: 1782bps, tx
bytes: 486738bytes, rx bytes: 486378bytes.
Timestamp: Tue Jan  3 01:13:54 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 487806bytes, rx bytes: 487458bytes.
Timestamp: Tue Jan  3 01:14:04 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 488928bytes, rx bytes: 488580bytes.
```

SD-WAN 健康检查

基础配置

1. 进入网络 → SD-WAN → 性能 SLA，新建 SLA 检测。



2. 对应 CLI。

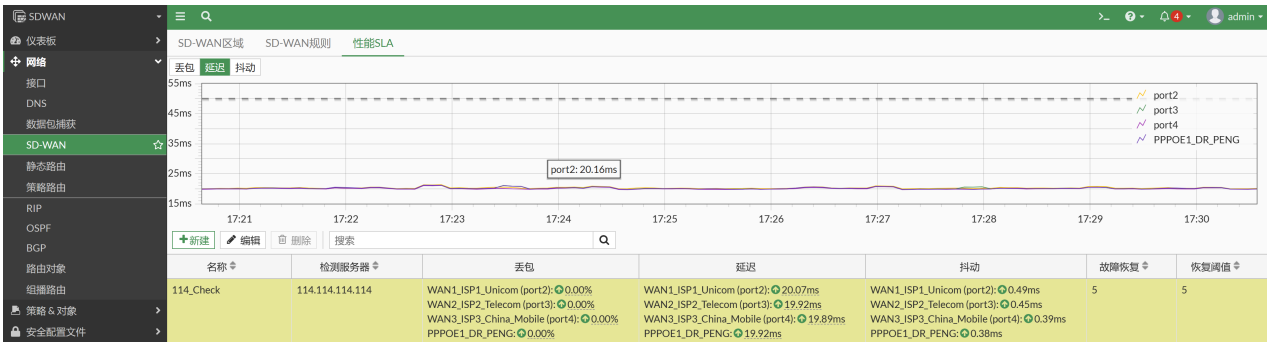
```bash
SDWAN # config system sdwan
SDWAN (sdwan) # config health-check
SDWAN (health-check) # edit "114_Check"
SDWAN (114_Check) # show
config health-check
    edit "114_Check"
        set server "114.114.114.114"    //可以指定两个被检测的server，server可以是IP或者FQDN//
        set protocol ping    //SLA探测使用的协议，可以选择ping、tcp-echo、udp-echo、http、twamp、dns、tcp-connect、ftp//
        set interval 500    //每隔500 milliseconds发送一下健康检测的探测//
        set probe-timeout 500    //探测数据包被视为丢失前的等待时间（毫秒），默认 = 500）//
        set failtime 5    //两个被探测的sla server在连续发起的5次探测都没有回应之后，没有回应的sd-wan接口成员将会被置位为dead状态，会将其在SD-WAN的选路接口组中剔除//
        set recoverytime 5    //sla状态在两个sla server中其中的一个连续响应5次后切换回alive状态，会恢复到SD-WAN的选路接口中//
        set members 0    //指定健康检查发送的出接口，0表示所有//
        config sla    //SLA Targets是可选项,SD-WAN规则中的 “Lowest Cost”规则 (mode sla) 和 “Maximize Bandwidth“规则 (mode load-balance)需要用到SLA Targets中的检测值，基于SLA检测值进行流量路径的判断和选择//
            edit 1
                set link-cost-factor latency packet-loss
                set latency-threshold 50
                set packetloss-threshold 3
            next
        end
    next
end
```

效果验证

1. Server IP 后台的处理：填写的 SLA server IP 会在 FGT 的内核中添加一个 Fib 路由条目项，以便每个 SD-WAN 接口拥有独立指向 sla server ip 的路由。这些特定的内核路由（被标记为 “proto = 17”）将使得 SD-WAN 接口成员到 sla 服务器的可访问性完全独立于通常的路由表转发表（静态路由、动态路由），从而不管静态/动态路由状态如何都可以对 sla 服务器的发起探测请求。

```
SDWAN # get router info kernel | grep 114.114.114.114     //内核下发的独立于路由转发表的路由//
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 114.100.1.204/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=114.100.1.196 dev=19(PPPOE1_DR_PENG)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 111.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=111.100.1.192 dev=6(port4)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 101.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=101.100.1.192 dev=5(port3)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 202.100.1.10/255.255.255.255/0→114.114.114.114/32 pref=0.0.0.0
gwy=202.100.1.192 dev=4(port2)
```

2. 健康检测 SLA 历史记录。



3. 实时 SLA 健康检查参数记录。

```
SDWAN # diagnose sys sdwan health-check status 114_Check
Health Check(114_Check):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(19.972), jitter(0.253), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(19.809), jitter(0.266), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(19.721), jitter(0.256), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(4 PPPOE1_DR_PENG): state(alive), packet-loss(0.000%) latency(19.856), jitter(0.299), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```

SLA 日志

1. 某一个接口的十分钟的使用统计。

```
SDWAN # diagnose sys sdwan intf-sla-log port3
Timestamp: Tue Jan  3 01:22:04 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 542424bytes, rx bytes: 542130bytes.
Timestamp: Tue Jan  3 01:22:14 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 543504bytes, rx bytes: 543210bytes.
Timestamp: Tue Jan  3 01:22:24 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 544626bytes, rx bytes: 544332bytes.
Timestamp: Tue Jan  3 01:22:34 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 545706bytes, rx bytes: 545412bytes.
Timestamp: Tue Jan  3 01:22:44 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 546828bytes, rx bytes: 546534bytes.
Timestamp: Tue Jan  3 01:22:54 2023, used inbandwidth: 877bps, used outbandwidth: 877bps, used bibandwidth: 1754bps, tx
bytes: 547908bytes, rx bytes: 547614bytes.
Timestamp: Tue Jan  3 01:23:04 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 549030bytes, rx bytes: 548736bytes.
Timestamp: Tue Jan  3 01:23:14 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 550152bytes, rx bytes: 549858bytes.
Timestamp: Tue Jan  3 01:23:24 2023, used inbandwidth: 890bps, used outbandwidth: 890bps, used bibandwidth: 1780bps, tx
bytes: 551274bytes, rx bytes: 550980bytes.
Timestamp: Tue Jan  3 01:23:34 2023, used inbandwidth: 884bps, used outbandwidth: 884bps, used bibandwidth: 1768bps, tx
bytes: 552354bytes, rx bytes: 552060bytes.
......
```

2. SLA 的日志记录开启（间隔单位为 s）。

```bash
SDWAN # config system sdwan
SDWAN (sdwan) # config health-check
SDWAN (health-check) # edit 114_Check
SDWAN (114_Check) # set sla-fail-log-period 10    //当成员处于失败状态时，将按指定的间隔生成sla日志//
SDWAN (114_Check) # set sla-pass-log-period 10    //当成员处于活动状态时，按指定的间隔生成sla日志//
SDWAN (114_Check) # end
SDWAN (sdwan) # end
```

3. SLA PASS 日志。

```bash
date=2023-01-03 time=01:36:49 eventtime=1672738609392716352 tz="-0800" logid="0113022925" type="event" subtype="sdwan"
level="information" vd="root" logdesc="SDWAN SLA information" eventtype="SLA" healthcheck="114_Check" slatargetid=1
interface="port4" status="up" latency="20.333" jitter="1.324" packetloss="0.000" inbandwidthavailable="10.00Gbps"
outbandwidthavailable="10.00Gbps" bibandwidthavailable="20.00Gbps" inbandwidthused="0kbps" outbandwidthused="0kbps"
bibandwidthused="0kbps" slamap="0x1" msg="Health Check SLA status."
```

4. SLA FAIL 日志。

```bash
date=2023-01-03 time=01:38:29 eventtime=1672738709588764152 tz="-0800" logid="0113022933" type="event" subtype="sdwan"
level="notice" vd="root" logdesc="SDWAN SLA notification" eventtype="SLA" healthcheck="114_Check" slatargetid=1
interface="port4" status="down" latency="0.000" jitter="0.000" packetloss="4.000" inbandwidthavailable="10.00Gbps"
outbandwidthavailable="10.00Gbps" bibandwidthavailable="20.00Gbps" inbandwidthused="0kbps" outbandwidthused="0kbps"
bibandwidthused="0kbps" slamap="0x0" metric="packetloss" msg="Health Check SLA status. SLA failed due to being over the
performance metric threshold."
```

5. 健康检查全部成功的情况。

```bash
SDWAN # diagnose sys sdwan health-check status 114_Check
Health Check(114_Check):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(20.186), jitter(0.659), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(19.979), jitter(0.519), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(70.000%) latency(20.073), jitter(0.695), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0
Seq(4 PPPOE1_DR_PENG): state(alive), packet-loss(0.000%) latency(20.122), jitter(0.552), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```

6. 制造接口的 Member3（Port4）丢包，查看失败的健康检查结果为 dead。

```bash
SDWAN # diagnose sys sdwan member
Member(1): interface: port2, flags=0x0 , gateway: 202.100.1.192, priority: 1 1024, weight: 0
Member(2): interface: port3, flags=0x0 , gateway: 101.100.1.192, priority: 100 1024, weight: 0
Member(3): interface: port4, flags=0x0 , gateway: 111.100.1.192, priority: 1 1024, weight: 0
Member(4): interface: PPPOE1_DR_PENG, flags=0x8 , gateway: 114.100.1.196, priority: 1 1024, weight: 0

SDWAN # diagnose sys sdwan health-check status 114_Check
Health Check(114_Check):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(20.068), jitter(0.454), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(19.887), jitter(0.483), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(dead), packet-loss(18.000%) sla_map=0x0
Seq(4 PPPOE1_DR_PENG): state(alive), packet-loss(0.000%) latency(19.876), jitter(0.497), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```

7. 对应的 SD-WAN 规则下 port4 接口也会显示 dead。

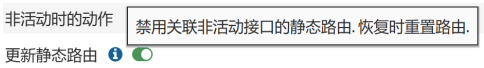
```bash
SDWAN # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x260 use-shortcut-sla
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(114_Check)
  Members(4):
    1: Seq_num(1 port2), alive, latency: 20.087, selected
    2: Seq_num(4 PPPOE1_DR_PENG), alive, latency: 19.904, selected
```

```
    3: Seq_num(2 port3), alive, latency: 19.950, selected
    4: Seq_num(3 port4), dead    //不再被SD-WAN规则选路使用//
  Internet Service(1): Microsoft-Office365(327782,0,0,0)
  Src address(1):
        192.168.1.0-192.168.1.255
```

8. 查看策略路由列表，SD-WAN 转发表中移除了 oif index 6：port4 接口。

```bash
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2097807361(0×7d0a0001) vwl_service=1(OFFICE_365) vwl_mbr_seq=1 4 2 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) gwy=202.100.1.192
oif=19(PPPOE1_DR_PENG) gwy=114.100.1.196 oif=5(port3) gwy=101.100.1.192 default    // SD-WAN转发表中移除了oif index 6:
port4接口
source(1): 192.168.1.0-192.168.1.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=0 last_used=2023-01-03 00:00:13
```

9. 由于健康检查里面的更新静态路由打开了（默认开启），因此相应的 SD-WAN 静态路由也会将 port4 对应的 SD-WAN 路由状态置位为 inactive（可以在路由 database 中看到，但会出现在路由表中）。




```bash
config system sdwan
    config health-check
        edit "114_Check"
            set update-static-route enable
        next
    end
end
```

```bash
SDWAN # get router info routing-table database
...
S      > 0.0.0.0/0 [1/0] via 111.100.1.192, port4 inactive, [1/0]
     *>            [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
     *>            [1/0] via 202.100.1.192, port2, [1/0]
     *>            [1/0] via 101.100.1.192, port3, [100/0]
...
```

10. 只要健康检查失败，则 port4 就不会被策略路由/SD-WAN 规则/路由转发数据了，和 link-moniter 的效果是类似的，其实进程也是使用的同样一个进程“link-monitor process (lnkmtd)”，debug lnkmtd 进程。

link-monitor process (*lnkmtd*)

```
FGT-1 # diag test application lnkmtd
1: Show memory info.
2: Show vdom monitor info.
3: Show fail detection info.
```

lnkmtd is the process responsible for processing *Performance SLA* probes

```
FGT-1 # diag debug enable
FGT-1 # diag debug application link-monitor <level>
Debug messages will be on for 30 minutes.
```

debugging of *lnkmtd*

Level	Description criteria
1	Server up or down events
2	Config changes
4	Engine events
8	ICMP probes
16	UDP probes
32	TCP probes

Level	Description criteria
64	HTTP probes
128	TWAMP probes
256	GRE probes
512	Link detection
1024	DNS messages
2048	Application messages
4096	Policy route messages

FORTINET

补充

讨论另外一个事情：SD-WAN 健康检查检测到出口异常，就会引起路由出接口变化，那么策略路由/SD-WAN 规则/路由在出接口发生了变化的时候，FGT 在转发数据的时候是什么样的一个处理逻辑。

1. 首先回答这个问题：SD-WAN 规则的下一跳/出接口切换了，流量就会立即切换吗？

2. 有两条命令需要知道：

```bash
config system global
      set snat-route-change disable（默认状态disable）
end
```

路由变化，Snat 的会话表是否跟随着一起变化更新？

- disable：会话表项不变化，继续保持该会话，保持原有出口 （默认值为 disable）
- enable：会话表项变化，会话状态会置位为 dirty，并且新来的数据将会重新上送 CPU 查询策略/路由等参数，重新进行会话的匹配，一旦路由变化将会匹配新的出口进行数据的转发

> **重要**
>
> 默认配置下的 SNAT 场景，即使转发流量的 SD-WAN 成员的健康检查变为 dead，只要该线路仍然存在路由（可能关闭了 `update-static-route`），已有的旧会话仍然不会切换到其他线路。

路由变化，**非 SNAT 会话** 表是否跟随着一起变化更新？

```bash
config system interface
    edit "port2"
        set preserve-session-route disable（默认状态disable）
    next
```

end

```
- disable：不保持会话。
- enable：强制保持会话，不会主动的更新会话中路由的下一跳地址。
- 由于是非 SNAT 会话，所以与 snat-route-change 是否是 enable 无关。

3. 总结来看：默认情况已存在的 SNAT 会话将会进行会话保持，不会跟随着 SD-WAN 规则/路由的出接口变化，而改变数据的转发出口，还是会保持原有出接口进行转发数据。这样的会话数据是不会切换的。新建的 SNAT 的会话，则会按照新建会话处理流程上送 CPU，查询当前最新的 SD-WAN 规则/路由等信息进行数据转发，会切换到新的出口上去。

4. 而如果没有 SNAT 的环境下，仅仅是三层路由转发的会话，会话是会随着路由的变化而 dirty，强制上送 CPU 进行 SD-WAN 规则/路由/策略的重新查询和转发处理，一旦路由出接口有变化，转发数据会立即跟换改会话数据流的出接口。

5. 一旦在接口下配置了会话保持 preserve-session-route enable，则会有最高的优先级保障会话保持的机制。

6. 还有另外一个参数（异步路由）会影响到路由变化时候的数据转发逻辑。

```bash
config system settings
    set asymroute disable（默认状态disable）
end
```
```

- disable：默认配置，不允许非对称路由。此时路由变化，Session（NAT/非 NAT）是否保持基于“snat-route-change”和“preserve-session-route”的配置结果而定。
- enable：此时无论“snat-route-change”和“preserve-session-route”如何配置，只要路由发生了变化，流量会重新查询路由强制将流量转发到新的下一跳出口。此时任何会话保持都会失效。这个拥有很高的优先级，打破一切的会话保持机制，当然考虑到安全性，不建议开启异步路由。

7. SD-WAN 规则的下一跳/出接口切换后，业务流量切换注意：因为 SD-WAN 的切换会相对频繁一些，以下参数如果调整不合适就会出现 SD-WAN 一切换就出现业务中断的情况。因此非特殊情况下，尽量保持以下三个参数为默认的状态，这样有利于 SD-WAN 路由切换时候业务的平滑切换。务必先理解清楚三个参数的含义，然后才可按照自己的需求选择性的开启。

```
config system global
      set snat-route-change disable（默认状态disable）
end
```

```
config system interface
    edit "wan1"
        set preserve-session-route disable（默认状态disable）
    next
end
```

```
config system settings
     set asymroute disable（默认状态disable）
end
```

SD-WAN 规则

Manual（手动选路）

Best Quality（最佳质量）

Lowest Cost (SLA)

Maximize Bandwidth (SLA)

- 基于轮询的 Maximize Bandwidth (SLA)
- 基于带宽的 Maximize Bandwidth (SLA)

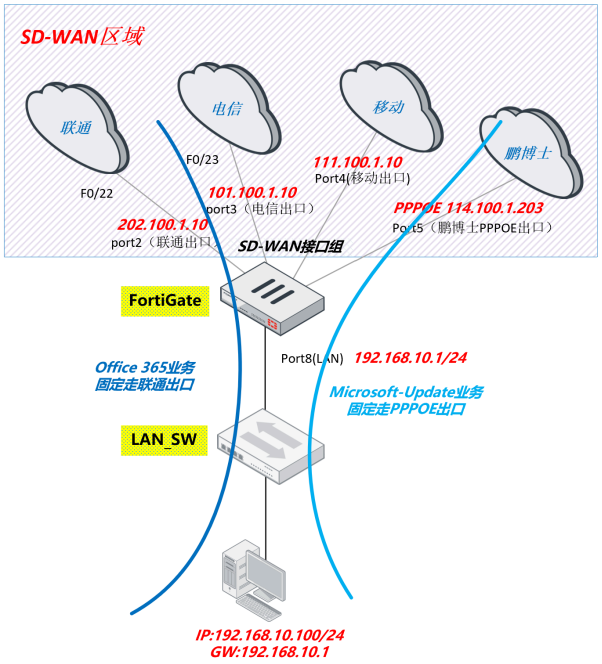
Manual（手动选路）

基本概念

1. 顾名思义让指定的 SD-WAN 流量从人工选择的固定线路出去。这时候 SD-WAN 规则的出口是固定的，不存在切换的问题，除非线路 DOWN/SD-WAN 健康检查失败/Link-Moniter 等因素影响让其出口的相关路由消失，SD-WAN 规则才会失效，流量才会切换到其他的线路上去，否则将强制让 SD-WAN 的流量出口固定到这条线路上。
2. 一般有两种常见的使用场景：将优先保障的重要业务固定指向到一条质量好的线路上去；或者反过来使用，将一些优先级不高（比如一些自动更新等占用较大带宽的应用）的业务固定指向到一条质量差的线路上去。具体的使用可按照自己的需求去定义。
3. 手工选路更像传统的策略路由，只能选择一个出接口，但相比策略路由会多跟多的匹配项，比如应用、Internet Service Database 等等，更加灵活一些。
4. 总结：
   - 相当于在 SD-WAN 接口上配置常规的策略路由
   - 不需要调用 Performance SLA（SD-WAN 状态检查）
   - 源 IP 地址建议配置为明细，不要配置为 any，避免匹配到本机发起流量
   - 目的地址可以调用 IP 地址 + 协议 + 端口或 Internet Service + Application Control
   - 只能选择一个出接口（单接口成员选择）

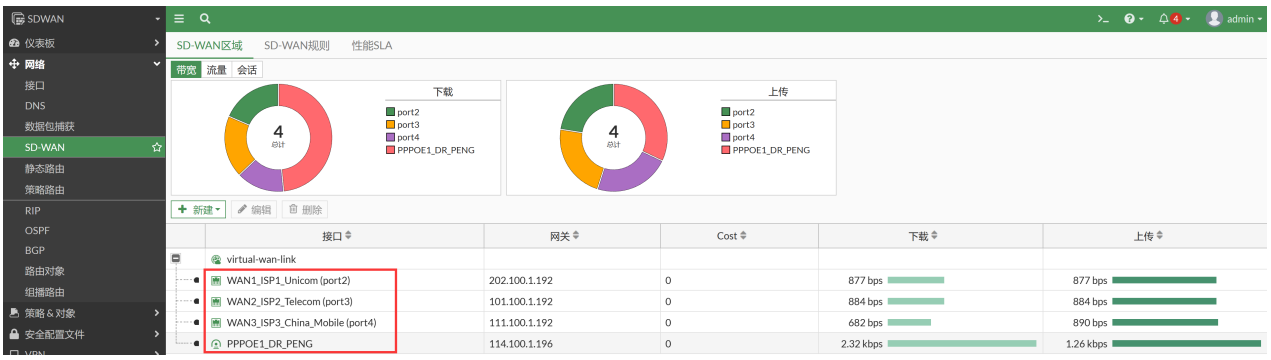
配置举例

网络拓扑



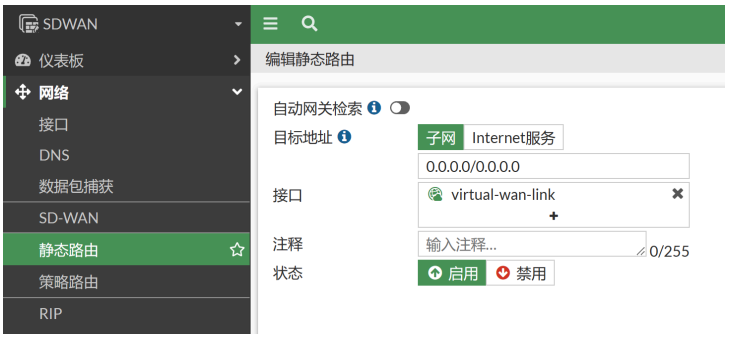
配置步骤

1. SD-WAN 接口成员定义。



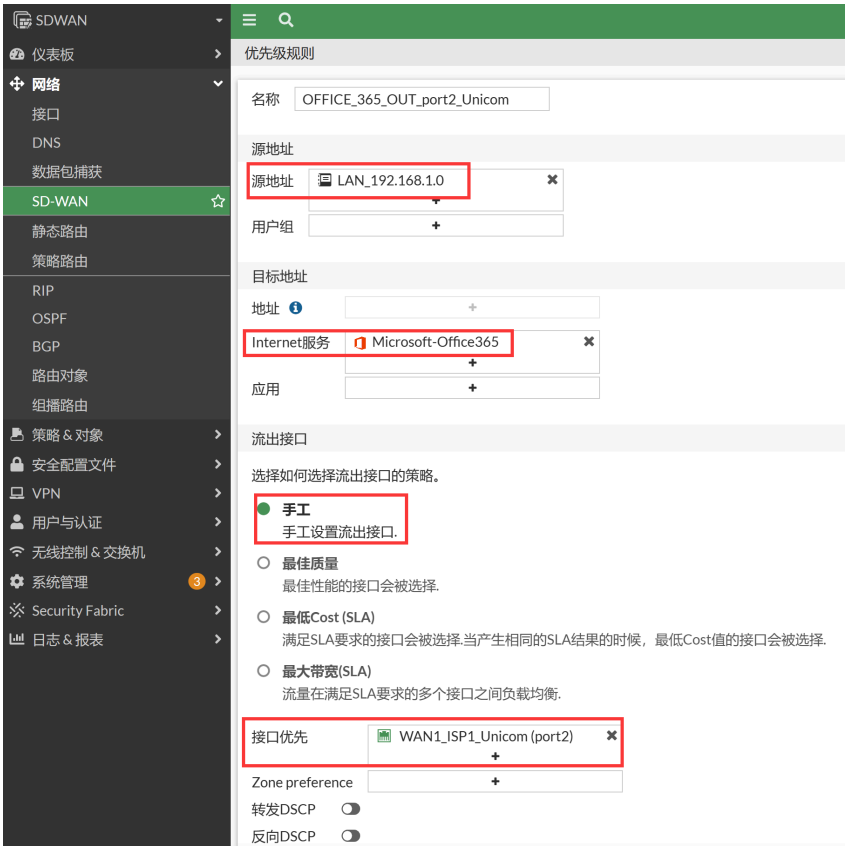
```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port2"
            set gateway 202.100.1.192
        next
        edit 2
            set interface "port3"
            set gateway 101.100.1.192
        next
        edit 3
            set interface "port4"
            set gateway 111.100.1.192
        next
        edit 4
            set interface "PPPOE1_DR_PENG"
        next
    end
 end
```

2. 配置 SD-WAN 关联的默认路由。




```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

3. SD-WAN 规则 1：Office 365 业务固定走联通出口。



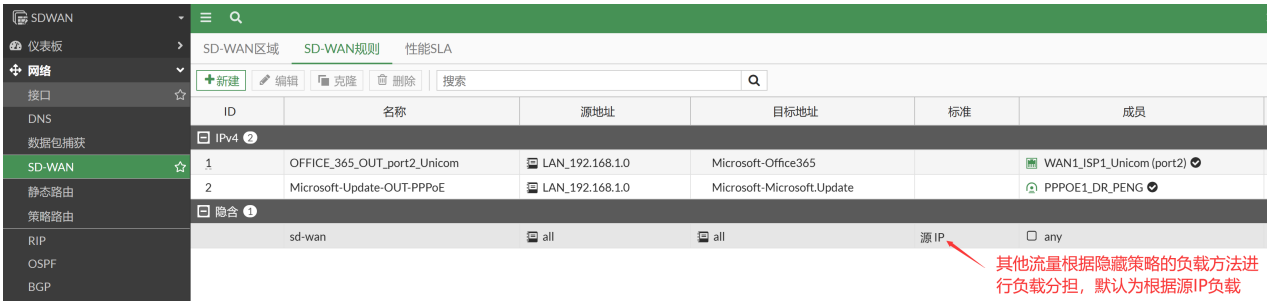

```
config system sdwan
        config service
        edit 1
            set name "OFFICE_365_OUT_port2_Unicom"
            set src "LAN_192.168.1.0"
            set internet-service enable
            set internet-service-name "Microsoft-Office365"
            set priority-members 1
            set gateway enable
            set default enable
        next
    end
end
```

4. SD-WAN 规则 2：Microsoft-Update 业务固定走 PPPOE 出口。




```
config system sdwan
        config service
        edit 2
            set name "Microsoft-Update-OUT-PPPoE"
            set src "LAN_192.168.1.0"
            set internet-service enable
            set internet-service-name "Microsoft-Microsoft.Update"
            set priority-members 4
        next
    end
end
```

5. 剩余流量匹配隐藏规则，根据源 IP 进行负载。



6. 配置上网策略，放通 SDWAN 流量。

重要

如果使用 ippool 的方式做 SNAT，在 SD-WAN 环境下，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。

```
config firewall ippool
    edit xxxx
        set associated-interface port2
    next
end
```




```
config firewall policy
    edit 1
        set name "To_Internet"
        set srcintf "port8"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

1. 查看 SD-WAN 成员状态。

```
SDWAN # diagnose sys sdwan member
Member(1): interface: port2, flags=0×0 , gateway: 202.100.1.192, priority: 1 1024, weight: 0
Member(2): interface: port3, flags=0×0 , gateway: 101.100.1.192, priority: 1 1024, weight: 0
Member(3): interface: port4, flags=0×0 , gateway: 111.100.1.192, priority: 1 1024, weight: 0
Member(4): interface: PPPOE1_DR_PENG, flags=0×8 , gateway: 114.100.1.196, priority: 1 1024, weight: 0
```

2. 查看 SDWAN 规则状态，两条规则中的线路均为 alive 状态。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×260 use-shortcut-sla
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(1):
    1: Seq_num(1 port2), alive, selected
  Internet Service(1): Microsoft-Office365(327782,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255

Service(2): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(1):
    1: Seq_num(4 PPPOE1_DR_PENG), alive, selected
  Internet Service(1): Microsoft-Microsoft.Update(327793,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

3. 查看策略路由条目，可以看到每条策略对应的出接口 SD-WAN 成员编号与接口 index。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2097545217(0×7d060001) vwl_service=1(OFFICE_365_OUT_port2_Unicom) vwl_mbr_seq=1 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(1) oif=4(port2) gwy=202.100.1.192 default
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=11 last_used=2023-01-03 03:13:30

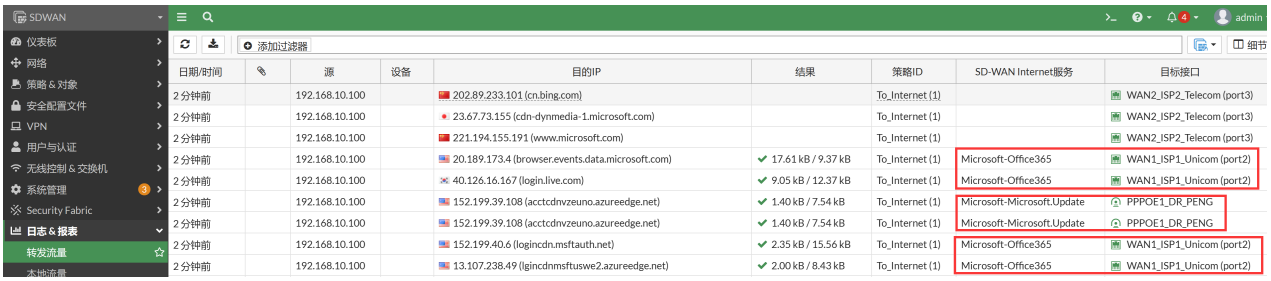
id=2131099650(0×7f060002) vwl_service=2(Microsoft-Update-OUT-PPPoE) vwl_mbr_seq=4 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(1) oif=16(PPPOE1_DR_PENG)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Microsoft.Update(327793,0,0,0)
hit_count=0 last_used=2023-01-03 02:59:29
```

```
SDWAN # diagnose ip address list
...
IP=202.100.1.10→202.100.1.10/255.255.255.0 index=4 devname=port2
IP=101.100.1.10→101.100.1.10/255.255.255.0 index=5 devname=port3
IP=111.100.1.10→111.100.1.10/255.255.255.0 index=6 devname=port4
...
IP=114.100.1.204→114.100.1.196/255.255.255.255 index=16 devname=PPPOE1_DR_PENG
...
```

4. 查看 SD-WAN 设备路由表。

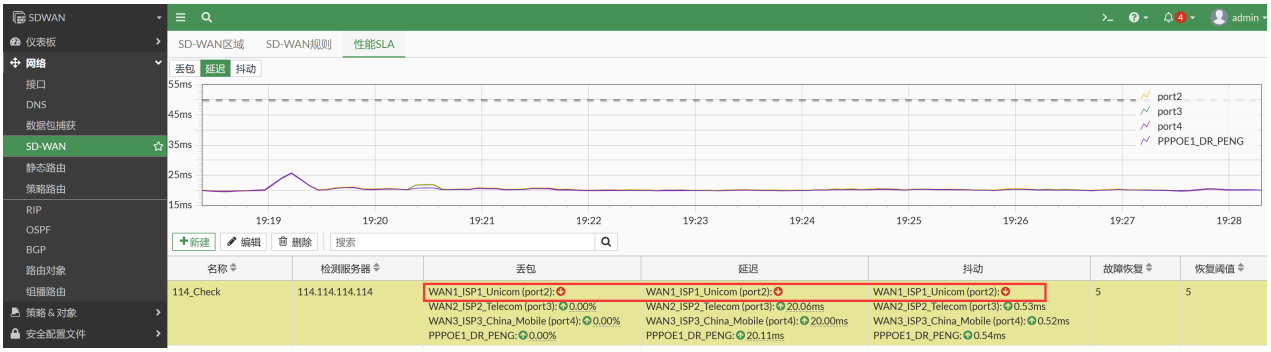
```
SDWAN # get router info routing-table all
...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
...
```

5. 在客户端 PC 上访问 Office365 的 Internet 资源，在 SD-WAN 设备上查看转发流量日志，Office 365 只走 port2，MS.Windows.Update 只走 PPPOE 接口出。



故障切换测试

1. 模拟 port2 故障（SD-WAN 健康检查，虽然不需要调用，但是依旧对链路状态和切换起关键作用，也包括 DOWN 接口/Link-Moniter 等其他切换方式）。




```
SDWAN # diagnose sys sdwan health-check
Health Check(114_Check):
Seq(1 port2): state(dead), packet-loss(100.000%) sla_map=0×0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(20.345), jitter(1.146), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(20.197), jitter(1.158), bandwidth-up(9999998), bandwidth-
dw(9999997), bandwidth-bi(19999995) sla_map=0×1
Seq(4 PPPOE1_DR_PENG): state(alive), packet-loss(0.000%) latency(20.336), jitter(1.103), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0×1
```

2. 查看 SD-WAN 成员状态，port2 已经变为 dead。

```
SDWAN # diagnose sys sdwan member
Member(1): interface: port2, flags=0×0 , gateway: 202.100.1.192, priority: 1 1024, weight: 0
Member(2): interface: port3, flags=0×0 , gateway: 101.100.1.192, priority: 1 1024, weight: 0
Member(3): interface: port4, flags=0×0 , gateway: 111.100.1.192, priority: 1 1024, weight: 0
Member(4): interface: PPPOE1_DR_PENG, flags=0×8 , gateway: 114.100.1.196, priority: 1 1024, weight: 0
```

```
SDWAN # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×260 use-shortcut-sla
  Gen(3), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Service disabled caused by no outgoing path.
  Members(1):
    1: Seq_num(1 port2), dead    //检测到port2 dead
  Internet Service(1): Microsoft-Office365(327782,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
Service(2): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(1):
    1: Seq_num(4 PPPOE1_DR_PENG), alive, selected    //PPPOE接口不受影响
  Internet Service(1): Microsoft-Microsoft.Update(327793,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

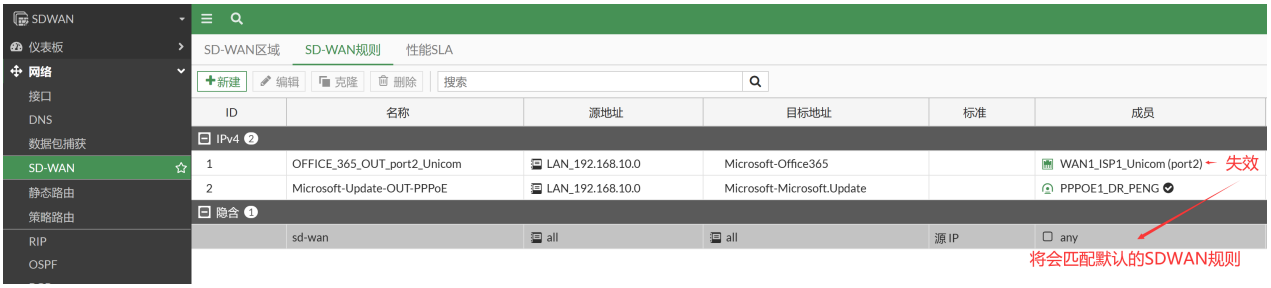
3. 查看 SD-WAN 设备的路由表，健康检查失败，会触发 port2 的 SD-WAN 路由失效。

```
SDWAN # get router info routing-table all
 ...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
 ...
```

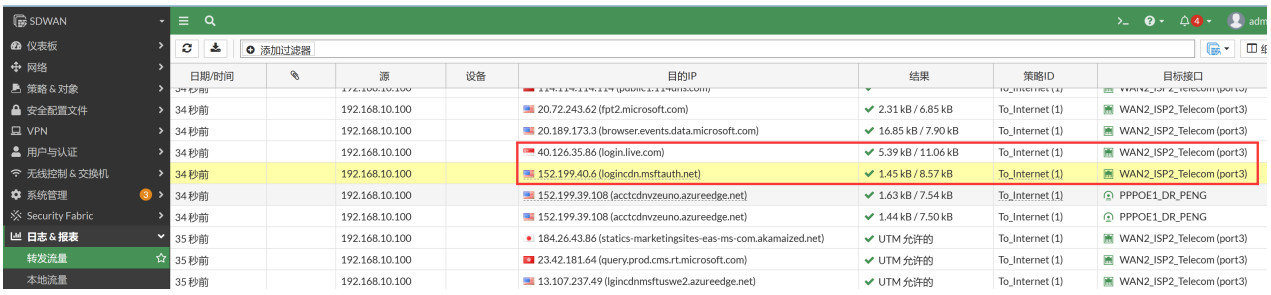
4. 查看策略路由列表。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):
id=2097545217(0×7d060001) vwl_service=1(OFFICE_365_OUT_port2_Unicom) dscp_tag=0×ff 0×ff flags=0×8 disable tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(0) default
source(1): 192.168.10.0-192.168.10.255    //由于FIB路由表中的port2路由消失，相应的对应SD-WAN规则1也会失效。流量将会匹配到最后一条默认的SD-WAN负载均衡规则，走普通路由表进行转发处理
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Office365(327782,0,0,0)
hit_count=39 last_used=2023-01-03 03:20:36

id=2131099650(0×7f060002) vwl_service=2(Microsoft-Update-OUT-PPPoE) vwl_mbr_seq=4 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00
tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(1) oif=16(PPPOE1_DR_PENG)
source(1): 192.168.10.0-192.168.10.255    //SD-WAN规则2不受影响
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): Microsoft-Microsoft.Update(327793,0,0,0)
hit_count=20 last_used=2023-01-03 03:21:06
```



5. 切换后，将不再匹配 SD-WAN 规则 1 了，而会匹配 SD-WAN 的默认规则，实际为查询默认路由（按照源 IP 负载的方式）选择出接口，此处选择的时候 port3 接口为出接口。

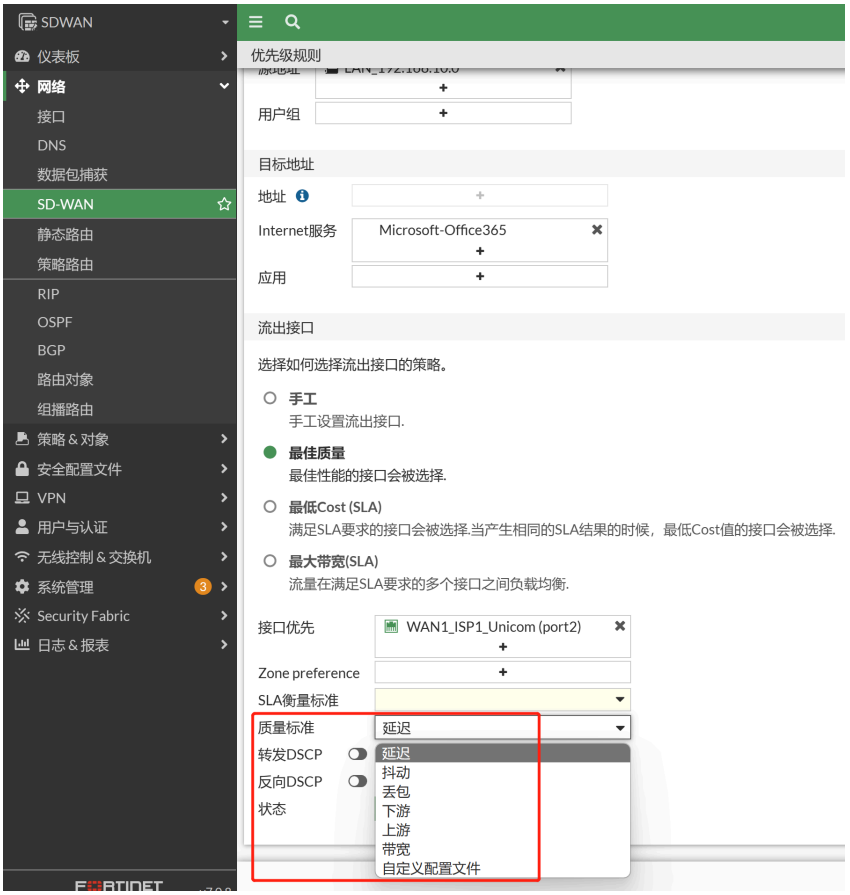


Best Quality（最佳质量）

基本概念

1. SD-WAN 规则选路原则是：选择具有“最佳质量”的链路。

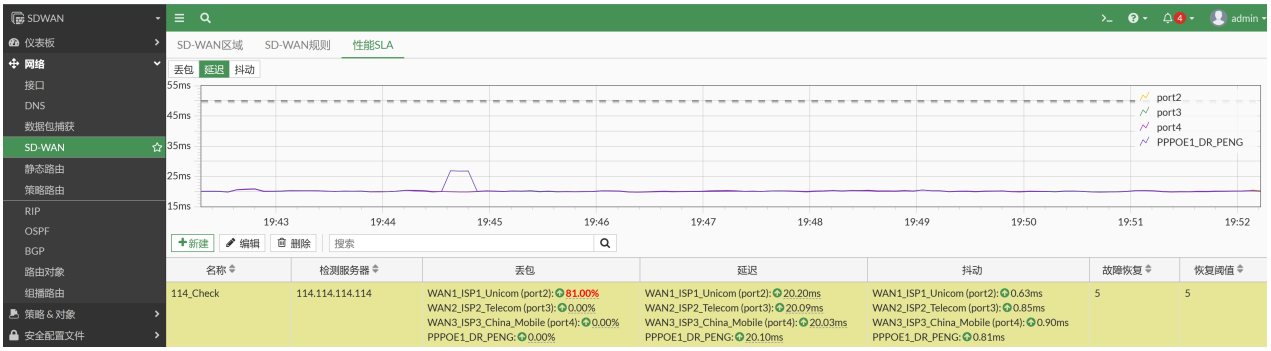
2. 最佳质量的标准有：



- 延迟
  - 一个数据包从一个节点到另一个节点所需的时间
  - 更少的延迟 = 更好的吞吐量
  - 延迟大引起的问题：访问速度慢，连接失败
  - 推荐应用：建议用于需要最佳响应时间的应用程序。例如：视频/VoIP
- 抖动
  - 在网络中传输数据包，相邻的数据包被转发之间的时间延迟差异（毫秒级别）。这是设备在按序转发数据包，包和包之间的一个短暂的中断时间。抖动通常是由 IP 网络中的拥塞引起的
  - 抖动大引起的问题：实时应用程序延迟
  - 推荐应用：建议用于需要实时转发传递数据包的应用程序。例如：VoIP
- 丢包
  - 发生在当通过网络传输的一个或多个数据包无法到达其目标地时候
  - 丢包而引起的问题：信息超时、加载时间慢、加载中断、连接关闭和信息丢失
  - 推荐应用：Oracle DB 和 SSH 这样的 CS 应用程序
- 下游带宽
  - 内部用户通过网络传输下载数据
  - 带宽不足而引起的问题：传输速度慢
  - 推荐应用：需要网络资源下载数据的应用程序。例如：文件服务器、云存储（Dropbox、OneDrive、百度网盘）
- 上游带宽
  - 内部用户通过网络传输上传数据
  - 带宽不足而引起的问题：传输速度慢，无法完成上传
  - 推荐应用：需要网络资源上传数据的应用程序。例如：备份系统
- 总体带宽
  - 内部用户通过网络传输上传下载数据的总和
- 自定义配置文件

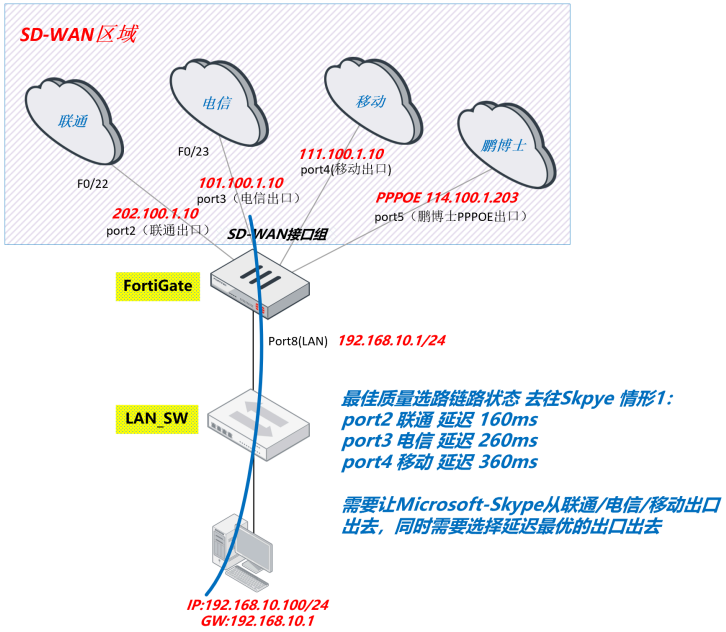
- 基于延迟、抖动、丢包、带宽权重的自定义。计算公式：【Link Quality Index = (packet-loss-weight * packet loss) + (latency-weight * latency) + (jitter-weight * jitter) + (bandwidth-weight/bandwidth)】
- 带宽不足而引起的问题：传输速度慢
- 推荐应用：需要网络资源上传和下载数据的应用程序。示例：文件服务器、云存储（Dropbox、OneDrive）

3. 最佳质量顾名思义就是在 SD-WAN 规则里配置的多个出接口里面：选择最好的！



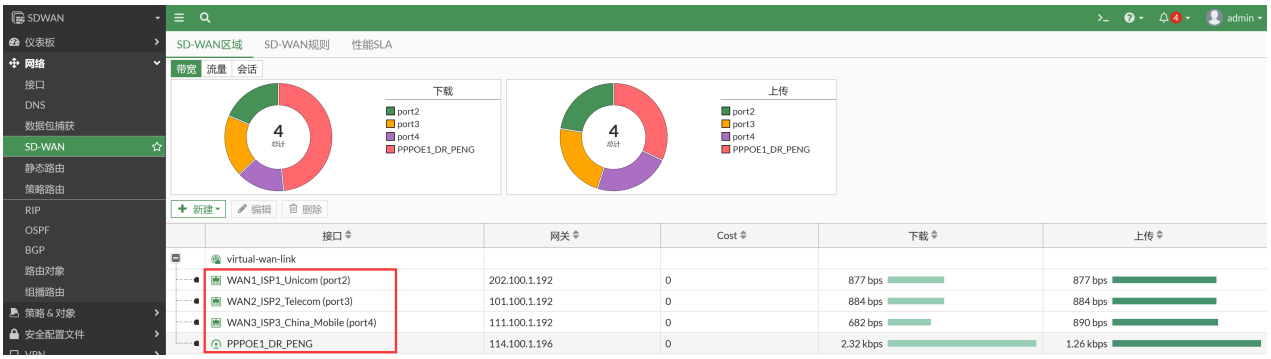
配置举例

网络拓扑



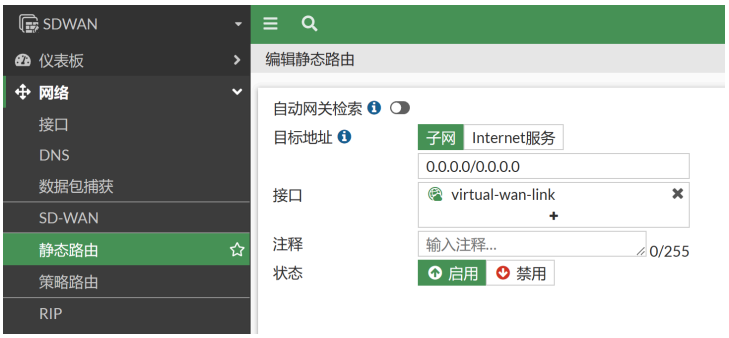
配置步骤

1. SD-WAN 接口成员定义。



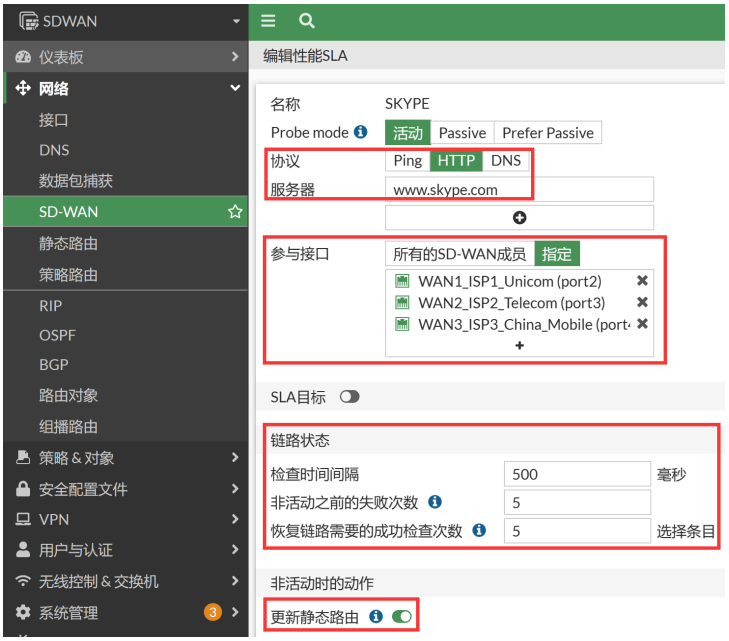
```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port2"
            set gateway 202.100.1.192
        next
        edit 2
            set interface "port3"
            set gateway 101.100.1.192
        next
        edit 3
            set interface "port4"
            set gateway 111.100.1.192
        next
        edit 4
            set interface "PPPOE1_DR_PENG"
        next
    end
 end
```

2. 配置 SD-WAN 关联的默认路由。




```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

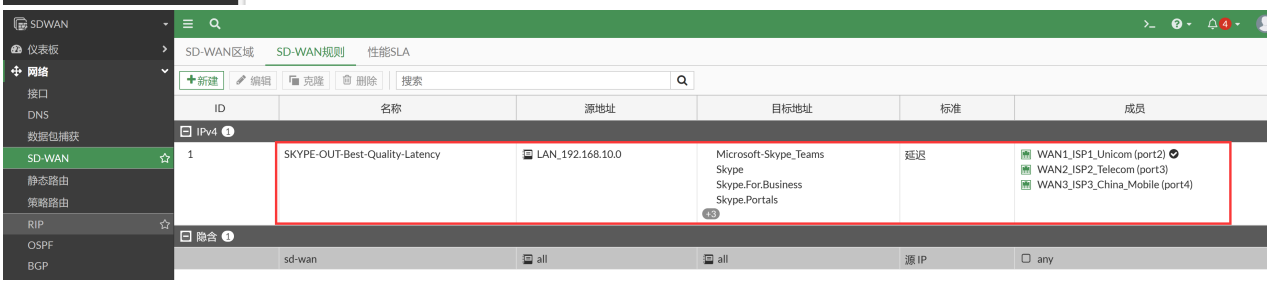
3. 配置性能 SLA，使用 HTTP 方法监控 www.skype.com ↗ 网站。




```
config system sdwan
    config health-check
        edit "SKYPE"
            set server "www.skype.com"
            set protocol http
            set members 1 2 3
        next
    end
end
```

4. 配置 SD-WAN 规则，目标地址为 Skype 相关应用和 Internet 服务，流出接口策略选择最佳质量，SLA 衡量标准选择上步创建的性能 SLA，质量标准选择延迟。






```
config system sdwan
    config service
        edit 1
            set name "SKYPE-OUT-Best-Quality-Latency"
            set mode priority
            set link-cost-factor latency     //标准为延迟
            set link-cost-threshold 10     //误差范围10%，避免频繁切换 （默认值）
            set src "LAN_192.168.10.0"
            set internet-service enable
            set internet-service-name "Microsoft-Skype_Teams"
            set internet-service-app-ctrl 10 28554 43540 28587 29350 28597
            set health-check "SKYPE"
            set priority-members 1 2 3
        next
    end
end
```

5. 配置 SD-WAN 的上网策略。

重要

如果使用 ippool 的方式做 SNAT，在 SD-WAN 环境下，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。

```
config firewall ippool
       edit xxxx
           set associated-interface port2
       next
end
```




```
config firewall policy
    edit 1
        set name "To_Internet"
        set srcintf "port8"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "default"
        set application-list "default"
        set logtraffic all
        set nat enable
    next
end
```

结果验证

1. 查看 SD-WAN 成员状态，以及健康检查监控状态，可以看到 port2、port3、port4 的监控延迟分别为 160ms 左右、260ms 左右、360ms 左右。

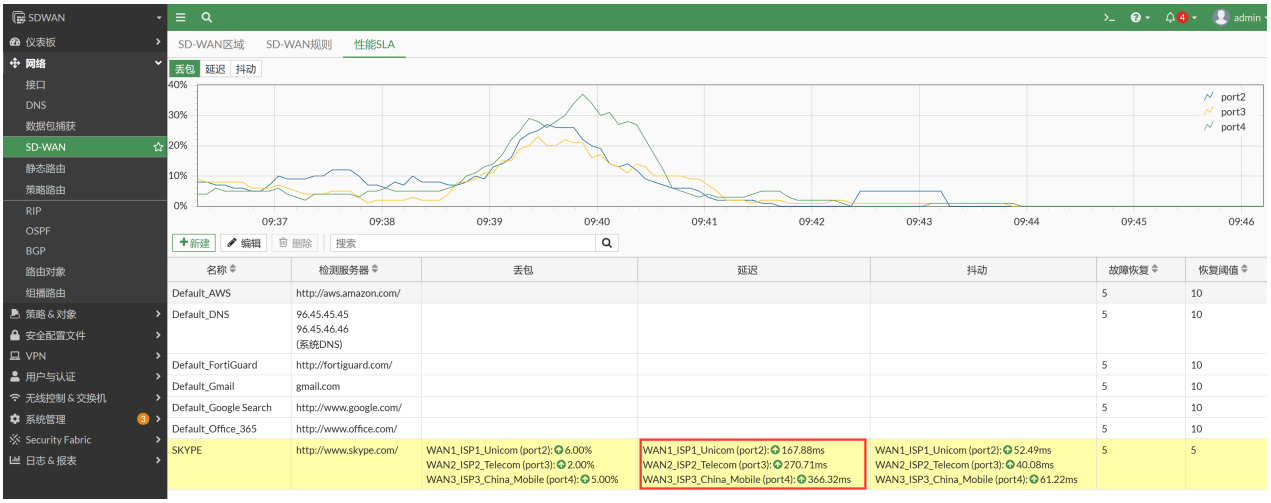
```
SDWAN # diagnose sys sdwan member
Member(1): interface: port2, flags=0×0 , gateway: 202.100.1.192, priority: 1 1024, weight: 0
Member(2): interface: port3, flags=0×0 , gateway: 101.100.1.192, priority: 1 1024, weight: 0
Member(3): interface: port4, flags=0×0 , gateway: 111.100.1.192, priority: 1 1024, weight: 0
Member(4): interface: PPPOE1_DR_PENG, flags=0×8 , gateway: 114.100.1.196, priority: 1 1024, weight: 0

SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(SKYPE)
  Members(3):
    1: Seq_num(1 port2), alive, latency: 171.847, selected
    2: Seq_num(2 port3), alive, latency: 267.559, selected
    3: Seq_num(3 port4), alive, latency: 366.488, selected
  Internet Service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0 10) Microsoft.Lync(4294837471,0,0,0
28554) Skype.Portals(4294838961,0,0,0 43540) Lync_Audio(4294837364,0,0,0 28587) Lync_Apps.Sharing(4294837363,0,0,0
29350) Lync_Video(4294837366,0,0,0 28597)
  Src address(1):
        192.168.10.0-192.168.10.255

SDWAN # diagnose sys sdwan health-check
Health Check(SKYPE):
Seq(1 port2): state(alive), packet-loss(15.000%) latency(178.808), jitter(53.983), bandwidth-up(9999995), bandwidth-
```

```
dw(9999954), bandwidth-bi(19999949) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(14.000%) latency(261.427), jitter(62.005), bandwidth-up(9999995), bandwidth-
dw(9999952), bandwidth-bi(19999947) sla_map=0x0
Seq(3 port4): state(alive), packet-loss(9.000%) latency(367.605), jitter(58.608), bandwidth-up(9999995), bandwidth-
dw(9999951), bandwidth-bi(19999946) sla_map=0x0
```



2. 查看策略路由列表，将会按照延迟的顺序 port2、port3、port4 这样排序，优先使用延迟小的出口 port2。

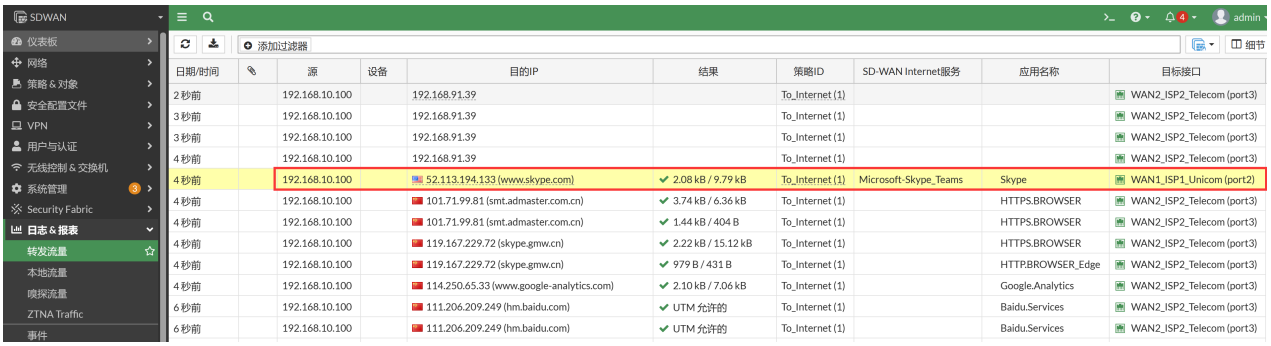
```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2131492865(0x7f0c0001) vwl_service=1(SKYPE-OUT-Best-Quality-Latency) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) oif=5(port3) oif=6(port4)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0, 10) Microsoft.Lync(4294837471,0,0,0,
28554) Skype.Portals(4294838961,0,0,0, 43540) Lync_Audio(4294837364,0,0,0, 28587) Lync_Apps.Sharing(4294837363,0,0,0,
29350) Lync_Video(4294837366,0,0,0, 28597)
hit_count=13 last_used=2023-01-03 17:40:51
```

3. 查看 SD-WAN 设备路由表。

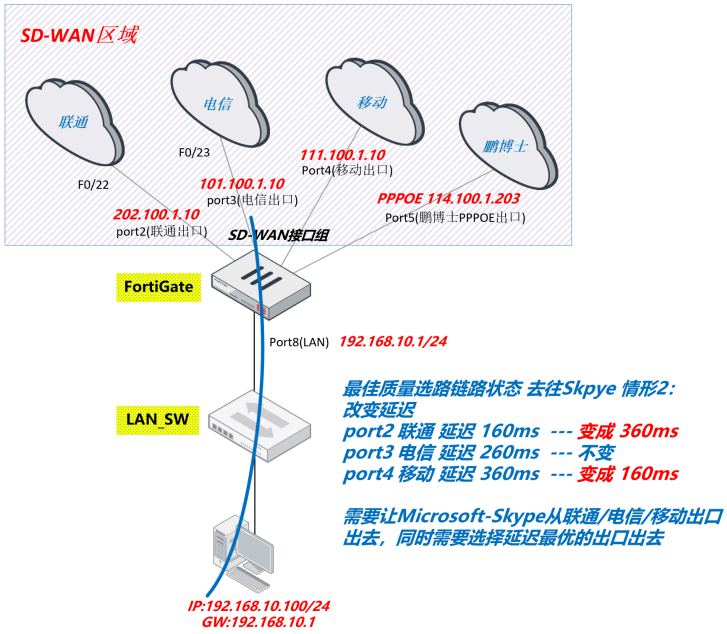
```
SDWAN # get router info routing-table all
...
S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
...
```

4. 使用客户端 PC 访问 Skype 相关资源，在 SD-WAN 设备上查看流量日志，可以看到相关流量被分配到 port2，符合上述 SD-WAN 检测结果。



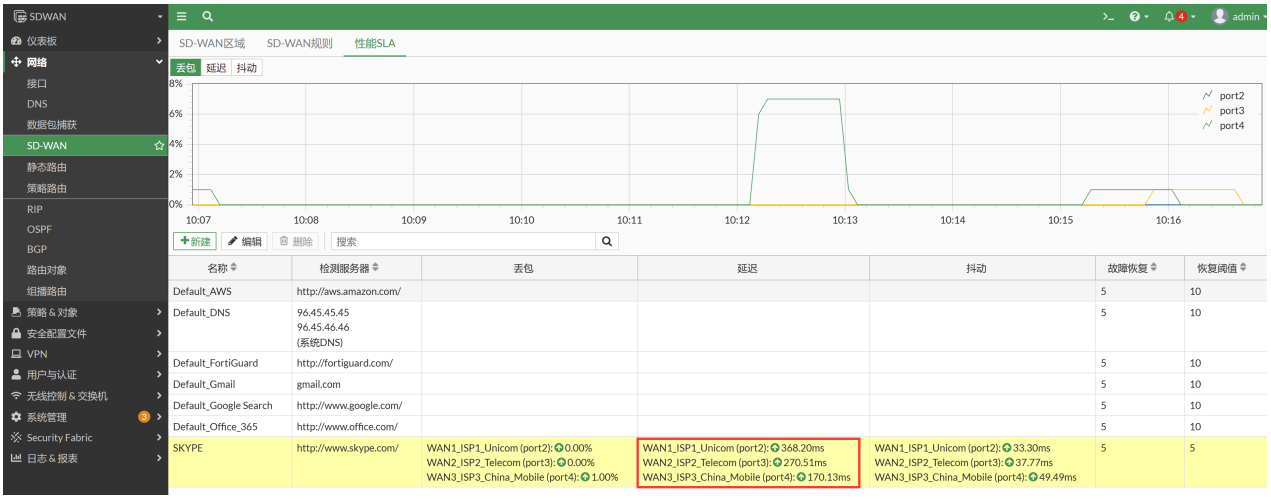
故障切换测试

1. 调整 port2 和 port4 的延迟，模拟延迟变化，将 port2 的延迟模拟为 360ms，port4 的延迟模拟为 160ms。



2. 切换延迟后，查看 SLA 的检查状态，port2 和 port4 的延迟进行了对调。

```
Seq(1 port2): state(alive), packet-loss(0.000%) latency(367.850), jitter(36.092), bandwidth-up(9999995), bandwidth-
dw(9999953), bandwidth-bi(19999948) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(270.216), jitter(37.558), bandwidth-up(9999995), bandwidth-
dw(9999953), bandwidth-bi(19999948) sla_map=0x0
SDWAN # diagnose sys sdwan health-check
Health Check(SKYPE):
Seq(3 port4): state(alive), packet-loss(0.000%) latency(169.290), jitter(28.239), bandwidth-up(9999995), bandwidth-
dw(9999952), bandwidth-bi(19999947) sla_map=0x0
```



3. 查看 SD-WAN 规则出接口的变化，出接口顺序也相应的对调，变为 port4，port3，port2。

```
SDWAN # diagnose sys sdwan service

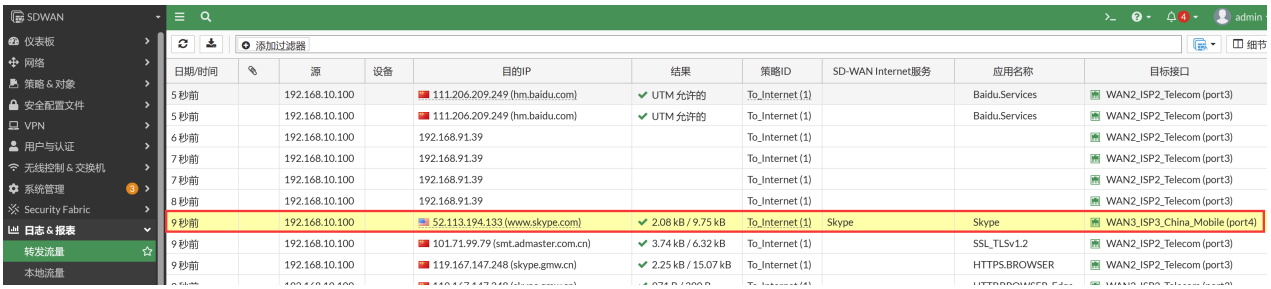
Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(25), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(SKYPE)
  Members(3):
    1: Seq_num(3 port4), alive, latency: 168.681, selected    //port4，优先从port13转发
    2: Seq_num(2 port3), alive, latency: 269.862, selected    //port3
    3: Seq_num(1 port2), alive, latency: 367.583, selected    //port2
  Internet Service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0 10) Microsoft.Lync(4294837471,0,0,0
28554) Skype.Portals(4294838961,0,0,0 43540) Lync_Audio(4294837364,0,0,0 28587) Lync_Apps.Sharing(4294837363,0,0,0
29350) Lync_Video(4294837366,0,0,0 28597)
  Src address(1):
        192.168.10.0-192.168.10.255
```

4. 查看策略路由中的 SD-WAN 规则列表，接口顺序已经变化为 port4、port3、port2 了，依据延迟的大小排序，优先选择延迟小的 port4 出口。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

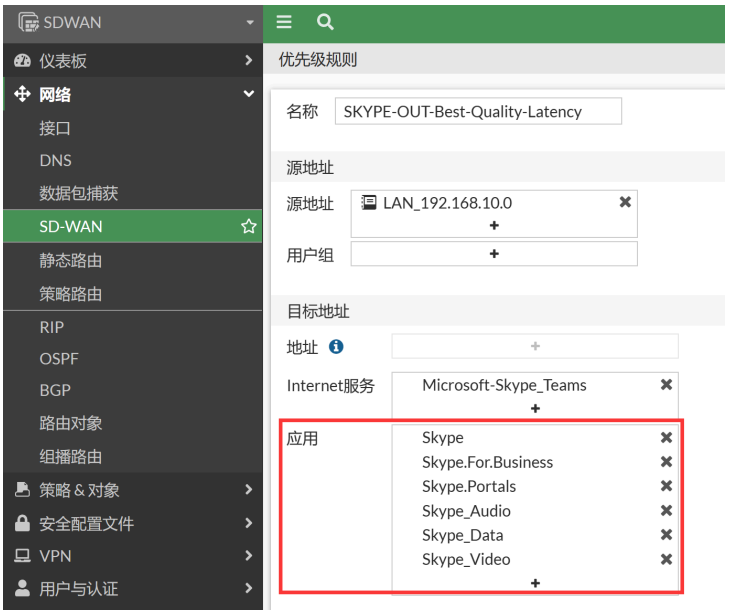
id=2131492865(0x7f0c0001) vwl_service=1(SKYPE-OUT-Best-Quality-Latency) vwl_mbr_seq=3 2 1 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=6(port4) oif=5(port3) oif=4(port2)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0, 10) Microsoft.Lync(4294837471,0,0,0,
28554) Skype.Portals(4294838961,0,0,0, 43540) Lync_Audio(4294837364,0,0,0, 28587) Lync_Apps.Sharing(4294837363,0,0,0,
29350) Lync_Video(4294837366,0,0,0, 28597)
hit_count=19 last_used=2023-01-03 18:05:39
```

5. 查看客户端访问 SKYPE 相关资源的流量日志，可以看到流量被分配到了 port4。



注意事项

1. 由于我的 SD-WAN 规则里面选择了应用程序 Skype，这样基于应用的 SD-WAN 规则（策略路由）要生效的话需要满足一些条件：



- 第一步：应用程序是需要先通过内网用户通过使用 SKPYE，然后 FGT 进行应用识别的，因此策略里面一定要开启 APP Control 才可以（需要识别加密流量需要开启 SSL 深度检测）。



- 第二步：识别之后将会学习到相应的应用程序（SKYPE）的 IP 地址，将会形成一个动态的应用程序（SKYPE）IP 地址数据库，最终才会将这个学习到的 IP 数据库加入到 SD-WAN 规则（策略路由）里面，因此这需要有一个学习的过程，SD-WAN 中的应用程序的调用不会立即生效，因此在没有成功学习之前，SKPYE 的流量会走到其他线路上去，FGT 会去学习这些识别出来的应用程序的 IP 地址，这并非问题，而是 SD-WAN 调用应用的

工作逻辑就是这样，学习完毕之后，新发起的流量才会匹配到 SD-WAN 规则（策略路由），我们可以通过命令行查看到学习的应用程序 IP 数据库信息。

```
SDWAN # diagnose sys sdwan internet-service-app-ctrl-list

Skype(10 4294838051): 52.113.194.133 6 443 Tue Jan  3 18:05:37 2023
Skype.Portals(43540 4294838961): 13.107.42.16 6 443 Tue Jan  3 15:40:51 2023
```

- 如果需要学习将到的 Skype 数据库立马完全生效，旧的会话是会保持原有出口的，想要这些旧的会话立即生效需要清除旧的会话和路由缓存信息，而新建的会话则不需要此操作，具体命令行如下。

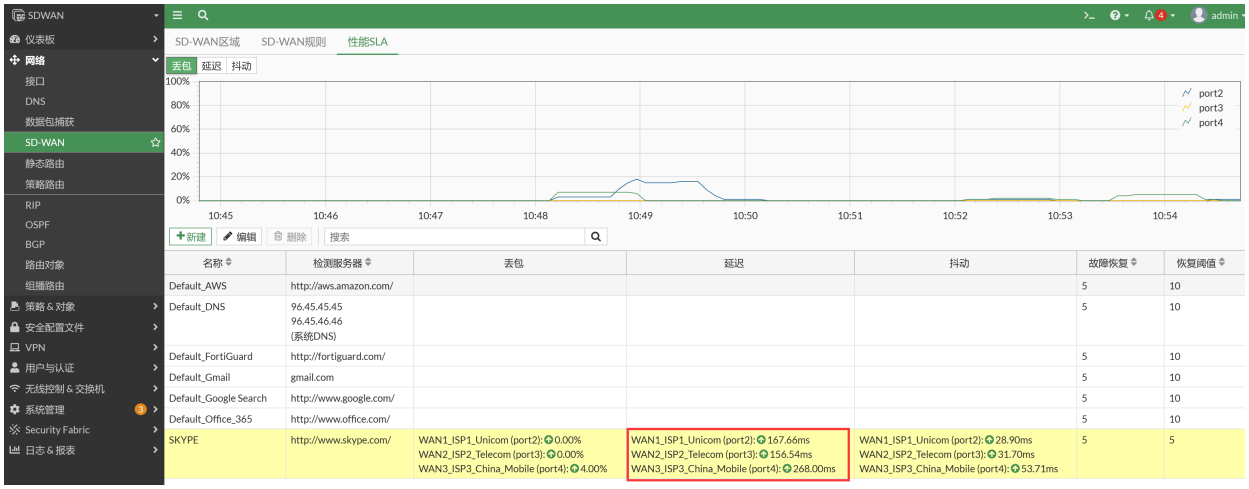
```
SD-WAN # diagnose sys session filter src 192.168.10.100
SD-WAN # diagnose sys session clear    //清除测试机器的会话，让其重新匹配新的SD-WAN规则（有skype的IP数据库更新）
SD-WAN # diagnose ip rtcache flush    //清除路由缓存
```

2. SD-WAN 规则中参数：link-cost-threshold 的作用。

```
SDWAN #  config system sdwan
SDWAN (sdwan) # config service
SDWAN (service) # show full-configuration
config service
    edit 1
        set name "SKYPE-OUT-Best-Quality-Latency"
...
        set link-cost-threshold 10
...
    next
end
```

- 根据上述的实验测试可以看出 SD-WAN 规则的工作逻辑：
  - 切换前延迟：
    - port2 联通延迟 160ms //选择延迟小的
    - port3 电信延迟 260ms
    - port4 移动延迟 360ms
    - 优先选择 port2
  - 切换后的延迟：
    - port2 联通延迟 360ms
    - port3 电信延迟 260ms
    - port4 移动延迟 160ms //选择延迟小的
    - 优先选择 port4
- 我们再来做一个测试，让延迟变成下面这个样子：
  - port2 联通延迟 166ms
  - port3 电信延迟 156ms //设置 port3 的延迟比 port2 低，但是低的不多
  - Port4 移动延迟 266ms
- 此时如何优先出接口呢？我们来看具体情况。

```
SDWAN # diagnose sys sdwan health-check
Health Check(SKYPE):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(166.441), jitter(42.834), bandwidth-up(9999995), bandwidth-
dw(9999953), bandwidth-bi(19999948) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(156.610), jitter(29.137), bandwidth-up(9999995), bandwidth-
dw(9999952), bandwidth-bi(19999947) sla_map=0x0
Seq(3 port4): state(alive), packet-loss(0.000%) latency(268.445), jitter(21.758), bandwidth-up(9999995), bandwidth-
dw(9999953), bandwidth-bi(19999948) sla_map=0x0
```



- 此时看 SD-WAN 规则的选择，此时 port2 会被优选，port3 延迟虽然比 port2 低，但是没有被优选，这是为什么呢?

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(SKYPE)
  Members(3):
    1: Seq_num(1 port2), alive, latency: 166.993, selected    //此时port2会被优选
    2: Seq_num(2 port3), alive, latency: 156.219, selected    //port3延迟虽然比port2低，但是没有被优选，这是为什么呢?
    3: Seq_num(3 port4), alive, latency: 268.677, selected
  Internet Service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0 10) Microsoft.Lync(4294837471,0,0,0
28554) Skype.Portals(4294838961,0,0,0 43540) Lync_Audio(4294837364,0,0,0 28587) Lync_Apps.Sharing(4294837363,0,0,0
29350) Lync_Video(4294837366,0,0,0 28597)
  Src address(1):
        192.168.10.0-192.168.10.255
```

- 查看策略路由列表，策略路由的接口顺序 port2、port3、port4。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2131623937(0×7f0e0001) vwl_service=1(SKYPE-OUT-Best-Quality-Latency) vwl_mbr_seq=1 2 3 dscp_tag=0×ff 0×ff flags=0×0
tos=0×00 tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) oif=5(port3)
oif=6(port4)    //策略路由的接口顺序port2、port3、port4
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0, 10) Microsoft.Lync(4294837471,0,0,0,
28554) Skype.Portals(4294838961,0,0,0, 43540) Lync_Audio(4294837364,0,0,0, 28587) Lync_Apps.Sharing(4294837363,0,0,0,
29350) Lync_Video(4294837366,0,0,0, 28597)
hit_count=24 last_used=2023-01-03 18:34:50
```

- 此时 SD-WAN 规则选择了 port2 作为出接口，port3 延迟虽然比 port2 低，但是没有被优选，这是为什么呢?
- 答案正是因为 set link-cost-threshold 10 此参数在起作用：当两个 SD-WAN 接口成员链路之间的 SLA 测量值存在显著差异时，结果是可以预期的，越小则越优先被选择。然而，当标准的 SLA 测量值足够接近时，SD-WAN 会优先考虑配置中的靠前位置位置的出口链路。比如 port2/port3/port4 的延迟差不多的时候，会优先选择 port2，这种逻辑是为了避免当 SLA 相对接近时链接之间的频繁切换而设计的。它由一个名为“set link-cost-threshold”的设置进行控制，该设置默认为 10%，如果差距在 10% 以内的值，会优先选择 SD-WAN 规则中配置靠前的接口进行数据转发。
  - port2 联通延迟 166ms，10% 的最低值：166 /（1 + 0.1）= 150.9，150.9 ~ 166 之间的值对于 port2 来说都是会被忽略的，即便延迟比 port2 要低，但是没有超过 10%，因此就忽略了比较计算值，依旧保持 port2 出口的选择。
  - port3 电信延迟 156ms，延迟比 port2 虽然要低，但是没有超过 < 150.9 ~166 （port2 百分之 10 的忽略范围），因此 port3 不会被优选。
  - port4 移动延迟 266ms，超过了 150.9 ~166 的范围，肯定不会被优选。
- 此时将 port3 的延迟设置为 149ms，查看 SD-WAN 策略中的顺序，发现 port3 的延迟 149.5 < 150.9 马上就会被优选。

```
Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(SKYPE)
  Members(3):
    1: Seq_num(2 port3), alive, latency: 149.569, selected
```

```
    2: Seq_num(1 port2), alive, latency: 164.794, selected
    3: Seq_num(3 port4), alive, latency: 259.659, selected
  Internet Service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0 10) Microsoft.Lync(4294837471,0,0,0
28554) Skype.Portals(4294838961,0,0,0 43540) Lync_Audio(4294837364,0,0,0 28587) Lync_Apps.Sharing(4294837363,0,0,0
29350) Lync_Video(4294837366,0,0,0 28597)
  Src address(1):
        192.168.10.0-192.168.10.255
```

- 查看策略路由列表，出接口顺序变为 port3，port2，port4。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2131623937(0×7f0e0001) vwl_service=1(SKYPE-OUT-Best-Quality-Latency) vwl_mbr_seq=2 1 3 dscp_tag=0×ff 0×ff flags=0×0
tos=0×00 tos_mask=0×00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=5(port3) oif=4(port2)
oif=6(port4)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(7): Microsoft-Skype_Teams(327781,0,0,0) Skype(4294838051,0,0,0, 10) Microsoft.Lync(4294837471,0,0,0,
28554) Skype.Portals(4294838961,0,0,0, 43540) Lync_Audio(4294837364,0,0,0, 28587) Lync_Apps.Sharing(4294837363,0,0,0,
29350) Lync_Video(4294837366,0,0,0, 28597)
hit_count=24 last_used=2023-01-03 18:34:50
```

- 总结来看：Best Quality（最佳质量）在 SD-WAN 规则中会选择最好质量的线路（延迟、丢包、抖动）最低值的出接口优先用于 SD-WAN 流量的转发，为了避免频繁的切换，当所有的出接口质量都差不多的时候（差距在 10% 以内），则会选择 SD-WAN 规则配置接口顺序靠前的接口用于 SD-WAN 流量的转发。

Lowest Cost (SLA)

概念介绍

SLA

Service-Level Agreement，服务品质协议，是服务提供者和客户之间的一个协议，用来保证可计量的网络性能达到所定义的服务品质。SD-WAN 的选路基于这个 SLA 品质标准来判断，SD-WAN 规则保障让流量一直走符合 SLA 品质的链路出去。从而达到业务/客户的 SLA 品质要求。

SLA-Target

1. SD-WAN 可定义 SLA-Target 设置保障的目标延迟、抖动和丢包率，一旦超出 SLA-Target 配置的目标值，则切换另外一条线路，以确保持续提供 SLA-Target 品质级别的服务。
2. SLA Target 有三种类型的判断阈值：
   - Latency (ms)
   - Jitter (ms)
   - Packet Loss (%)




```
config system sdwan
    config health-check
    edit "114_Check"
        set server "114.114.114.114"
        set members 0
        config sla
            edit 1
                set latency-threshold 200
                set packetloss-threshold 2
            next
            edit 2
                set latency-threshold 250
                set jitter-threshold 10
                set packetloss-threshold 5
            next
            edit 3
                set latency-threshold 300
```

```
                set jitter-threshold 15
                set packetloss-threshold 8
            next
        end
    next
end
```

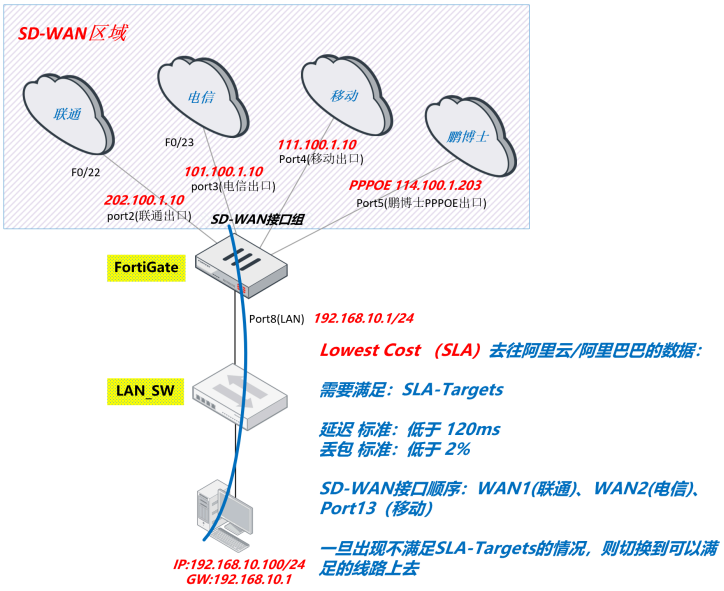
3.Lowest Cost (SLA) 和 Maximize Bandwidth (SLA) 都需要调用 SLA Target，基于 SLA Target 进行相关的判断和选路。

选路原则

1. 只有满足 SLA-Targets 的出接口才有机会被选中，如果低于 SLA-Targets 的接口将会被移除选中列表中。
2. 如果有多个接口满足 SLA-Targets，那么就选择 SD-WAN 规则配置接口顺序靠前的接口用于转发 SD-WAN 流量。
3. 同时只能一个出接口被优选，用于 SD-WAN 流量的转发。
4. Lowest Cost（SLA）完全基于 SLA-Targets 进行工作，因此首先需要在 SD-WAN 状态检查里面配置具体的 SLA-Targets 标准，然后在 SD-WAN 规则中选择相应的 SLA-Targets，只有符合选择的 SLA-Targets 标准的出口，才会被 SD-WAN 规则所计算并用于出口数据的转发，将选择符合 SLA-Targets 且接口顺序靠前的出口用于数据转发，同时只有一个接口用于数据的转发。

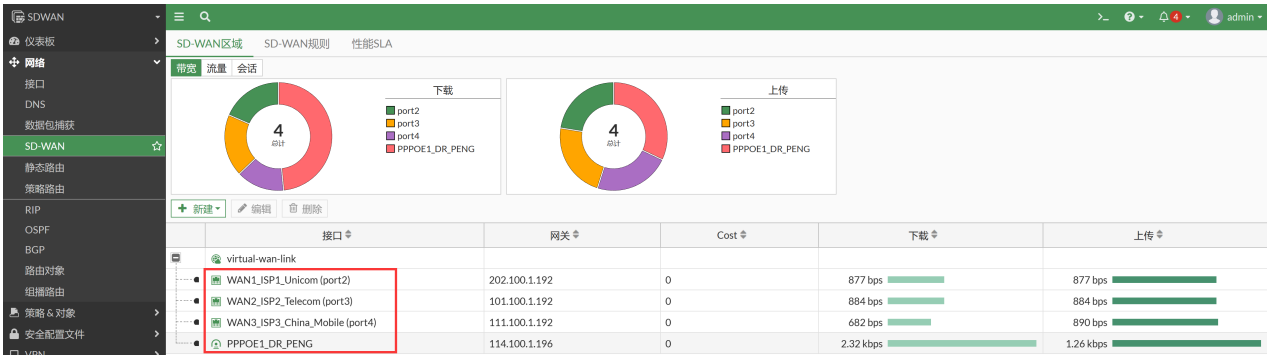
配置举例

网络拓扑



配置步骤

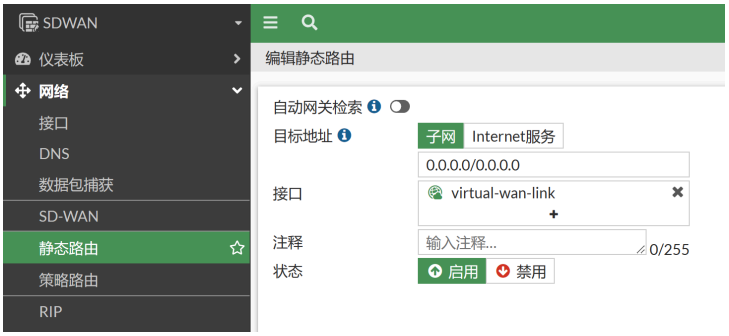
1. SD-WAN 接口成员定义。




```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
```

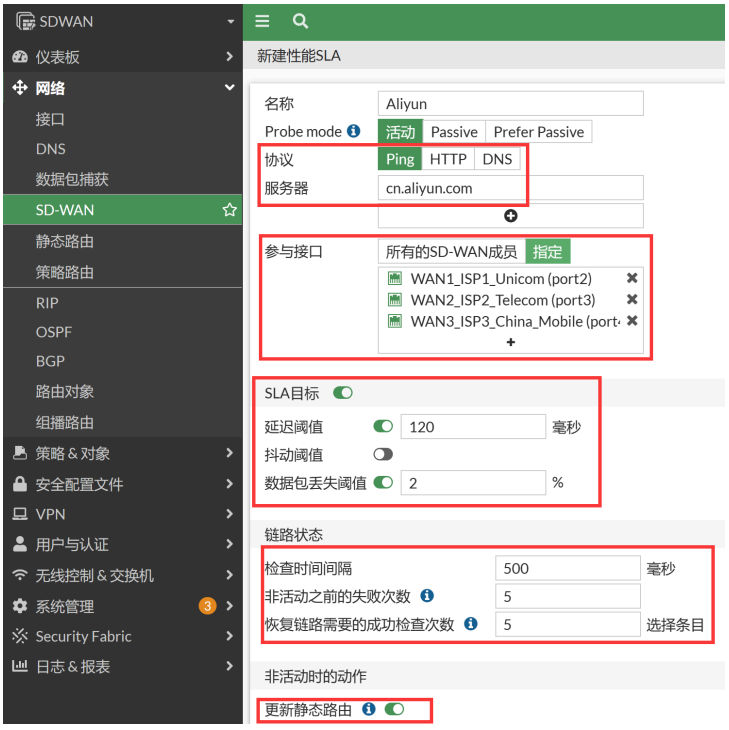
```
    config members
        edit 1
            set interface "port2"
            set gateway 202.100.1.192
        next
        edit 2
            set interface "port3"
            set gateway 101.100.1.192
        next
        edit 3
            set interface "port4"
            set gateway 111.100.1.192
        next
        edit 4
            set interface "PPPOE1_DR_PENG"
        next
    end
end
```

2. 配置 SD-WAN 关联的默认路由。

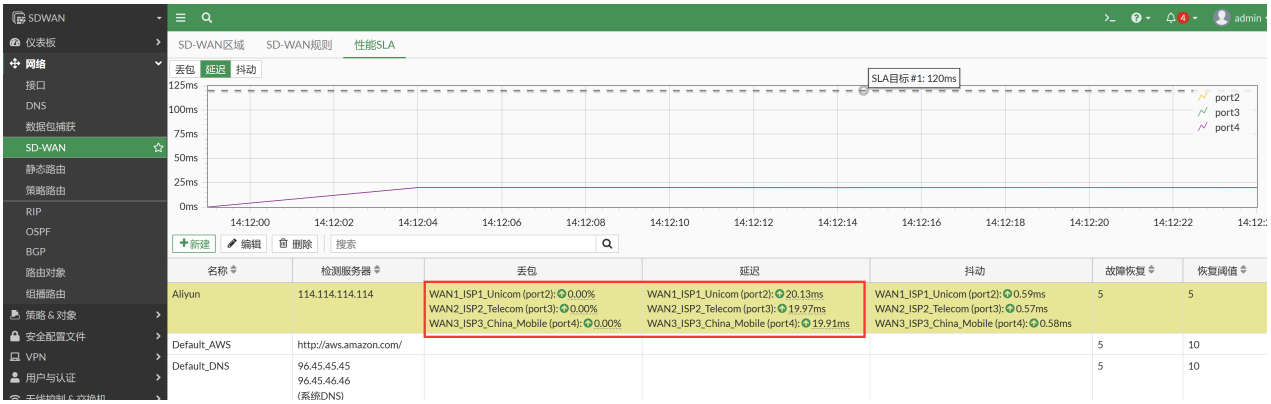



```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

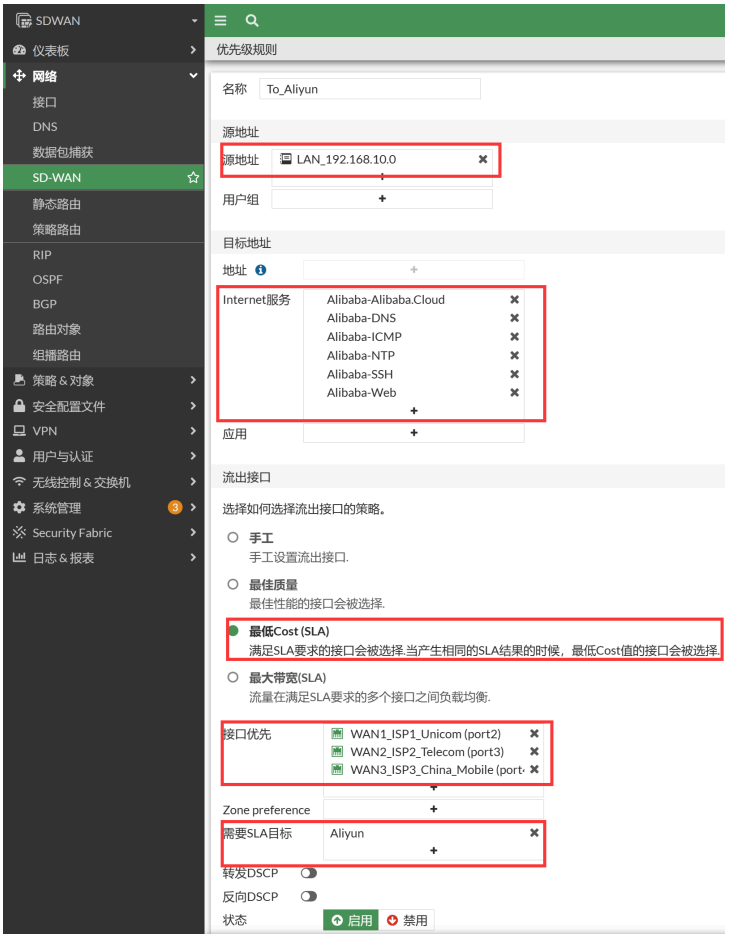
3. 配置健康检查，SLA 目标监控阿里云的延迟及丢包率。



```
config system sdwan
config health-check
    edit "Aliyun"
        set server "cn.aliyun.com"
        set members 1 2 3
        config sla
            edit 1
                set link-cost-factor latency packet-loss
                set latency-threshold 120
                set packetloss-threshold 2
            next
        end
    next
end
```



4. 配置 SD-WAN 规则，目标为阿里云相关 Internet 服务，引用上步配置的 SLA Target。



```
config system sdwan
config service
    edit 1
        set name "To_Aliyun"
        set mode sla
        set src "LAN_192.168.10.0"
        set internet-service enable
        set internet-service-name "Alibaba-Alibaba.Cloud" "Alibaba-DNS" "Alibaba-ICMP" "Alibaba-NTP" "Alibaba-SSH"
"Alibaba-Web"
        config sla
            edit "Aliyun"
                set id 1
            next
        end
        set priority-members 1 2 3
    next
end
```

5. 配置安全策略允许 SD-WAN 流量。

> **重要**
>
> 如果使用 ippool 的方式做 SNAT，在 SD-WAN 环境下，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。
>
> ```
> config firewall ippool
>     edit xxxx
>         set associated-interface port2
>     next
> end
> ```




```
config firewall policy
    edit 1
        set name "To_Internet"
        set srcintf "port8"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "default"
        set application-list "default"
        set logtraffic all
        set nat enable
    next
end
```

结果验证

1. 查看健康检查状态，三条链路均为 alive 状态。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(19.893), jitter(0.531), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(19.704), jitter(0.408), bandwidth-up(9999999), bandwidth-
```

```
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(19.719), jitter(0.556), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```

2. 查看 SD-WAN 规则状态，出接口顺序为 port2，port3，port4。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(11), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(1 port2), alive, sla(0x0), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 port3), alive, sla(0x0), gid(0), cfg_order(1), cost(0), selected
    3: Seq_num(3 port4), alive, sla(0x0), gid(0), cfg_order(2), cost(0), selected
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

3. 查看策略路由中 SD-WAN 规则的列表，出接口顺序为 port2，port3，port4。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2132213761(0x7f170001) vwl_service=1(To_Aliyun) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) oif=5(port3) oif=6(port4)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=0 last_used=2023-01-03 22:16:17
```

4. 查看路由表。

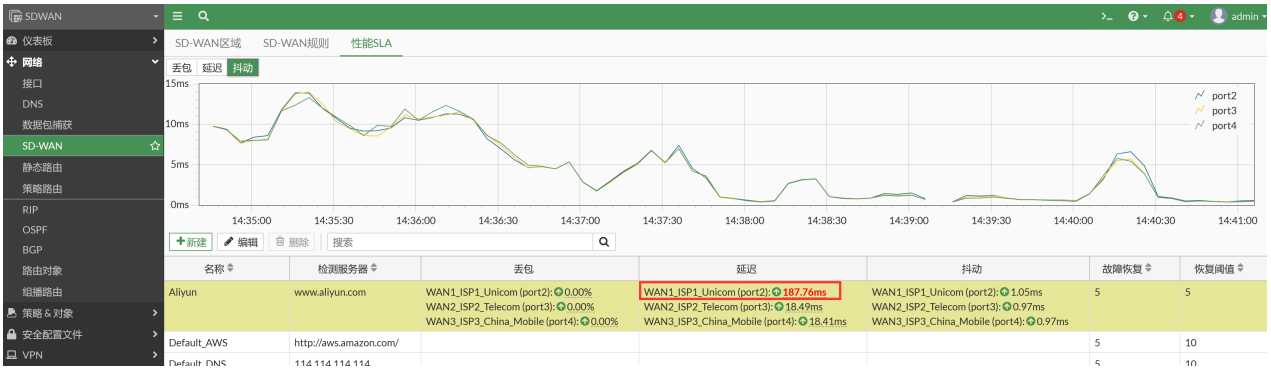
```
SDWAN # get router info routing-table all
 ...

S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
 ...
```

线路切换测试

1. 让 port2 的延迟超过 SLA-Target。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(1.000%) latency(187.502), jitter(8.489), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(1.000%) latency(48.102), jitter(9.499), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(1.000%) latency(48.508), jitter(8.618), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```



2. 查看 SD-WAN 策略状态，由于 port2 的延迟不满足 SLA Target，出接口顺序变为 port3，port4，port2。

```
SDWAN # diagnose sys sdwan service

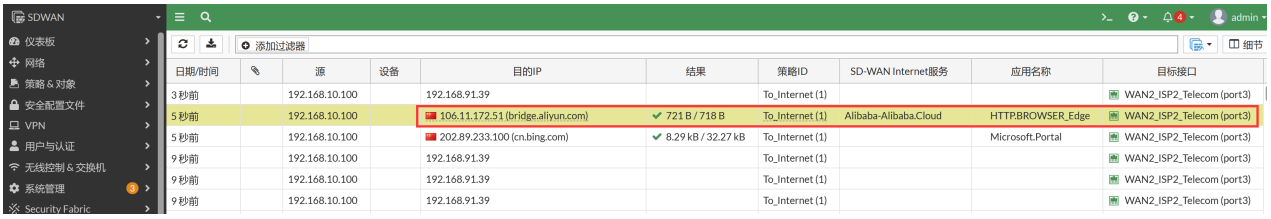
Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(2 port3), alive, sla(0×1), gid(0), cfg_order(1), cost(0), selected    //port3将会优先选择wan2进行转发
    2: Seq_num(3 port4), alive, sla(0×1), gid(0), cfg_order(2), cost(0), selected
    3: Seq_num(1 port2), alive, sla(0×0), gid(0), cfg_order(0), cost(0), selected    //port2不满足的SLA，顺序置于最后面了
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

3. 查看策略路由列表中的 SD-WAN 规则，由于 port2 的延迟不满足 SLA Target，出接口顺序变为 port3，port4，port2。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2132344833(0×7f190001) vwl_service=1(To_Aliyun) vwl_mbr_seq=2 3 1 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00 tos_mask=0×00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=5(port3) oif=6(port4) oif=4(port2)
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=0 last_used=2023-01-03 22:16:17
```

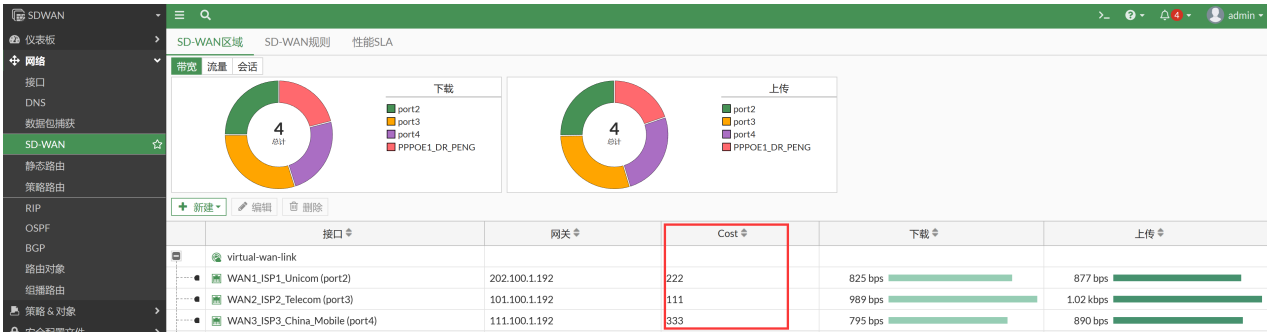
4. 使用内网 PC 客户端访问阿里云相关资源，查看流量日志，可以看到流量切换到了 port3。



接口成员 Cost 值对选路的影响

1. SD-WAN 接口成员的接口 Cost 值将影响 Lowest Cost (SLA) 的接口选择顺序，如果 SD-WAN 成员接口均达到了 SLA Target，那么 Cost 值越小越优先，这将打破配置顺序而进行用户自定义的 SD-WAN 接口优先级。

2. 比如自定义：port2 Cost 222，port3 Cost 111，port4 Cost 333。此时 Lowest Cost (SLA) 将会优选 Cost 较小的 WAN2，而忽略配置接口的顺序，Cost 值优先进行比较。



3. 查看健康检查状态，三个接口成员均达到了 SLA Target。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(20.301), jitter(0.578), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(20.225), jitter(0.610), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(20.143), jitter(0.571), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×1
```

4. 查看 SD-WAN 策略状态，Cost 小的 port3 优先。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(2 port3), alive, sla(0×1), gid(0), cfg_order(0), cost(111), selected    //Cost小的port3优先
    2: Seq_num(1 port2), alive, sla(0×1), gid(0), cfg_order(2), cost(222), selected
    3: Seq_num(3 port4), alive, sla(0×1), gid(0), cfg_order(1), cost(333), selected
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

5. 查看策略路由列表中的 SD-WAN 规则，Cost 小的 port3 优先。

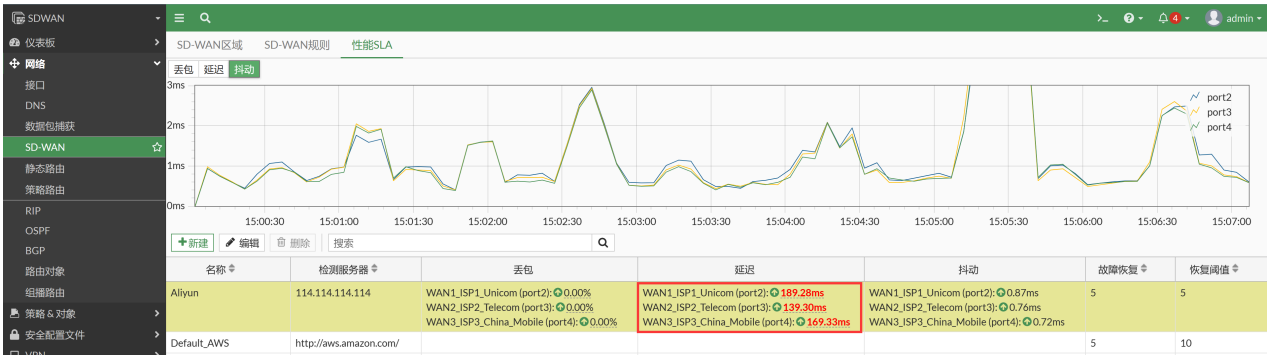
```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2132869121(0×7f210001) vwl_service=1(To_Aliyun) vwl_mbr_seq=2 1 3 dscp_tag=0×ff 0×ff flags=0×0 tos=0×00 tos_mask=0×00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=5(port3) oif=4(port2) oif=6(port4)    //Cost小的port3优先
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=25 last_used=2023-01-03 22:49:28
```

接口成员顺序对选路的影响

1. 将 SD-WAN 成员接口的 cost 全部改回 0，如果 port2、port3、port4 三者都无法满足 SLA 目标了，那么 SD-WAN 规则如何选择出接口呢?
2. 答案是：如果三者都无法满足 SLA 目标值了，那么将会按照 SD-WAN 规则配置的接口顺序选择出接口，也就是以 port2、port3、port4 这样的顺序选择，port2 将会优先转发数据。此时顺序优先。
3. 将 port2、port3、port4 的延迟调整超过 SLA Target。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(156.008), jitter(2.069), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(186.951), jitter(2.078), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×0
Seq(3 port4): state(alive), packet-loss(0.000%) latency(176.968), jitter(2.055), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0×0
```



4. 查看 SD-WAN 规则状态，全部不符合 SLA Target 的时候，按照配置的接口顺序进行选择出接口。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(1 port2), alive, sla(0×0), gid(0), cfg_order(0), cost(0), selected    //全部不符合的时候，按照配置的接口顺序进行
选择出接口
    2: Seq_num(2 port3), alive, sla(0×0), gid(0), cfg_order(1), cost(0), selected
```

```
    3: Seq_num(3 port4), alive, sla(0x0), gid(0), cfg_order(2), cost(0), selected
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

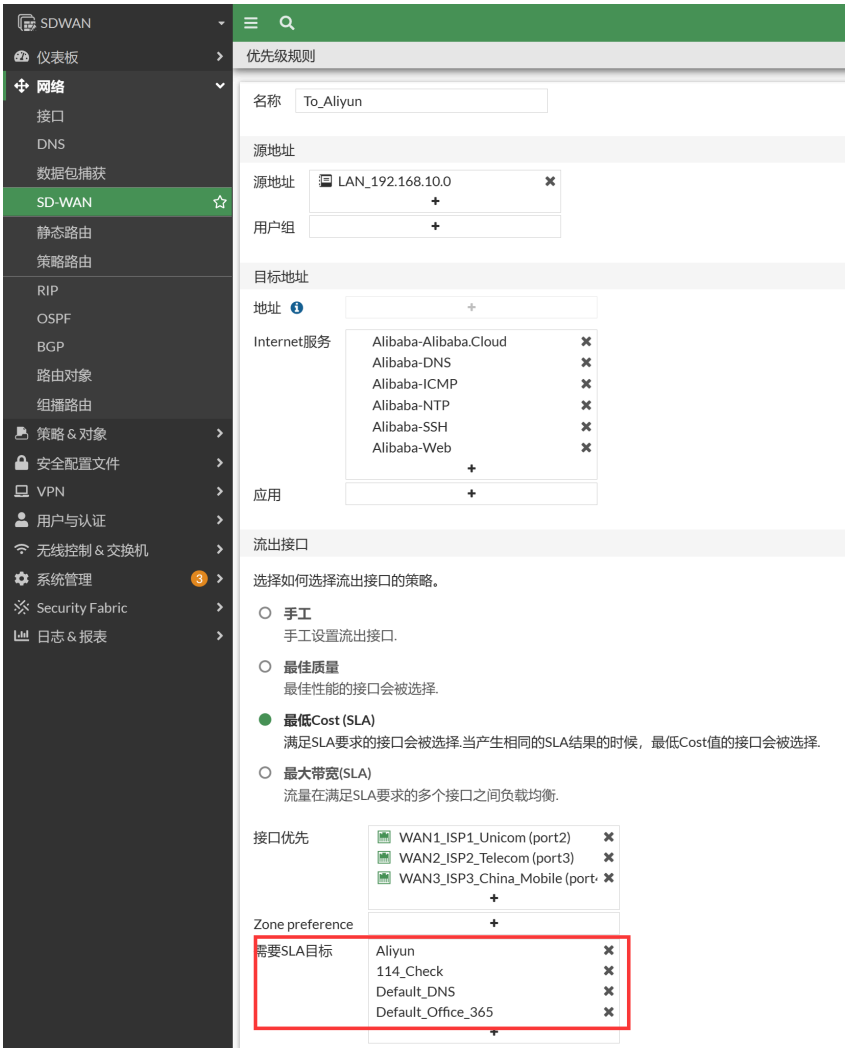
5. 查看策略路由列表，全部不符合 SLA Target 的时候，按照配置的接口顺序进行选择出接口。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2133131265(0x7f250001) vwl_service=1(To_Aliyun) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2) oif=5(port3) oif=6(port4)    //全部不符合的时候，按照
配置的接口顺序进行选择出接口
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=25 last_used=2023-01-03 22:49:28
```

sla-compare-method number 对选路的影响

1. 上述情况的一个补充命令：如果全部不符合则相对于没有 SLA Target 了，这显然不够合理，还有优化的空间，这时候就引出了另外一个参数：set sla-compare-method number。
2. SD-WAN 规则可以调用多个 SLA Target，然后优化计算的方法。



```
config system sdwan
config service
    edit 1
…
        config sla
            edit "Aliyun"
                set id 1
            next
            edit "114_Check"
                set id 1
            next
            edit "Default_DNS"
                set id 1
            next
            edit "Default_Office_365"
                set id 1
            next
        end
…
        set sla-compare-method order     //默认值，如果按此设置，多个SLA目标之间的逻辑关系是and，只要有其中一个SLA目标不符合，则将该接口剔除SD-WAN规则的选择，如果SLA全部失效，则按照配置的接口顺序进行选择出接口
…
    next
end
```

3. 默认情况下，CLI 下 SLA 的设置为 set sla-compare-method order，如果按此设置，多个 SLA 目标之间的逻辑关系是 and，只要有其中一个 SLA 目标不符合，则将该接口剔除 SD-WAN 规则的选择，如果 SLA 全部失效，则按照配置的接口顺序进行选择出接口。

```
SDWAN (1) # set sla-compare-method
order     Compare SLA value based on the order of health-check.
number    Compare SLA value based on the number of satisfied health-check.  Limits health-checks to only configured
member interfaces.
```

4. set sla-compare-method numbe，此参数是 SLA 目标全部失效的一个补充，如果全部失效，则选择失败数较小的接口进行 SD-WAN 流量的转发，比如 port2 失败 2 个，port3 失败 3 个，port4 失败 1 个，则会选择 port4 进行数据的转发。

5. 我们配置多个服务器的 SLA Target，在 SD-WAN 规则里面也调用多个 SLA Target，然后虽然全部都失败了，但是可以选择符合 SLA Target 条件更多的接口作为 SD-WAN 规则的出接口，在全部失败的矮子中选择一个最优的出接口。当然我们需要设置不同的目的 IP 进行健康检查，同时配置不同的 SLA 条目，在 SD-WAN 规则调用的时候也只能调用不同服务器的 SLA 目标，才可以进行这样的进一步比较。

Maximize Bandwidth (SLA)

- 基于轮询的 Maximize Bandwidth (SLA)
- 基于带宽的 Maximize Bandwidth (SLA)

基于轮询的 Maximize Bandwidth (SLA)

重要

ADVPN 目前不支持 Maximize Bandwidth (SLA) 模式。

概念介绍

SLA

Service-Level Agreement，服务品质协议，是服务提供者和客户之间的一个协议，用来保证可计量的网络性能达到所定义的服务品质。SD-WAN 的选路基于这个 SLA 品质标准来判断，SD-WAN 规则保障让流量一直走符合 SLA 品质的链路出去。从而达到业务/客户的 SLA 品质要求。

SLA-Target

1. SD-WAN 可定义 SLA-Targets 设置最低保障的延迟、抖动和丢包率，一旦超过 SLA-Targets 提供的保障最低值，则立即切换另外一条线路，以确保持续提供 SLA-Targets 品质级别的服务。
2. SLA Target 有三种类型的判断阈值：
   - Latency (ms)
   - Jitter (ms)
   - Packet Loss (%)




```bash
config system sdwan
    config health-check
    edit "114_Check"
        set server "114.114.114.114"
        set members 0
        config sla
            edit 1
                set latency-threshold 200
                set packetloss-threshold 2
            next
            edit 2
                set latency-threshold 250
```

```
                set jitter-threshold 10
                set packetloss-threshold 5
            next
            edit 3
                set latency-threshold 300
                set jitter-threshold 15
                set packetloss-threshold 8
            next
        end
    next
end
```

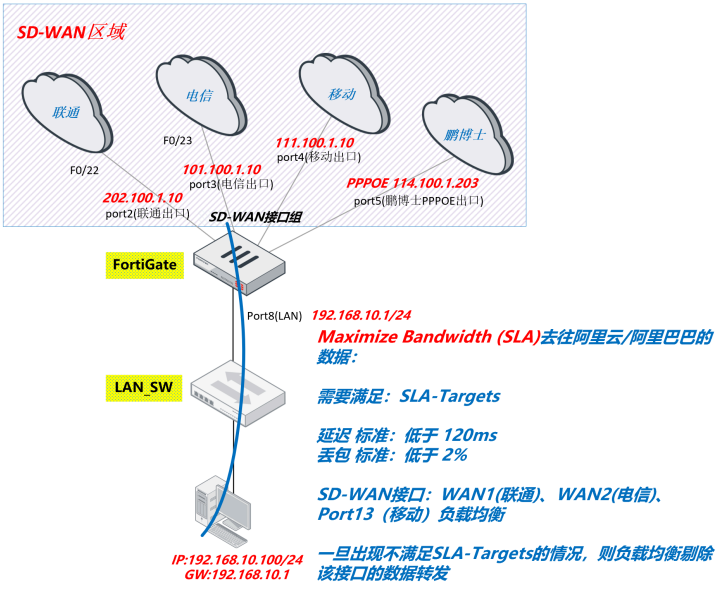
3.Lowest Cost (SLA) 和 Maximize Bandwidth (SLA) 都需要调用 SLA Target，基于 SLA Target 进行相关的判断和选路。

选路原则

1. 只有满足 SLA-Targets 的出接口才有机会被选中，如果低于 SLA-Targets 的接口将会被移除选中列表中。
2. 如果有多个接口满足 SLA-Targets，这些接口将按照配置的负载方式进行负载均衡来转发 SD-WAN 的流量，以便达到带宽最大利用率的效果。
3. 负载方法支持 round-robin（默认）、source-ip-based、source-dest-ip-based、inbandwidth、outbandwidth、bibandwidth。
4. 本文使用 round-robin 模式为例，所有流量将按循环顺序等量分配到 SD-WAN 规则中的成员接口（前提是该接口满足 SLA target）。
5. Maximize Bandwidth (SLA) 基于 SLA-Targets 进行工作，因此首先需要在 SD-WAN 状态检查里面配置具体的 SLA-Targets 标准，然后再 SD-WAN 规则中选择相应的 SLA-Targets，只有符合选择的 SLA-Targets 标准的出口，才会被 SD-WAN 规则所计算并用于出口数据的转发，符合 SLA-Targets 的接口都会被用于数据的转发，多个接口按照配置的负载方式进行负载均衡处理。

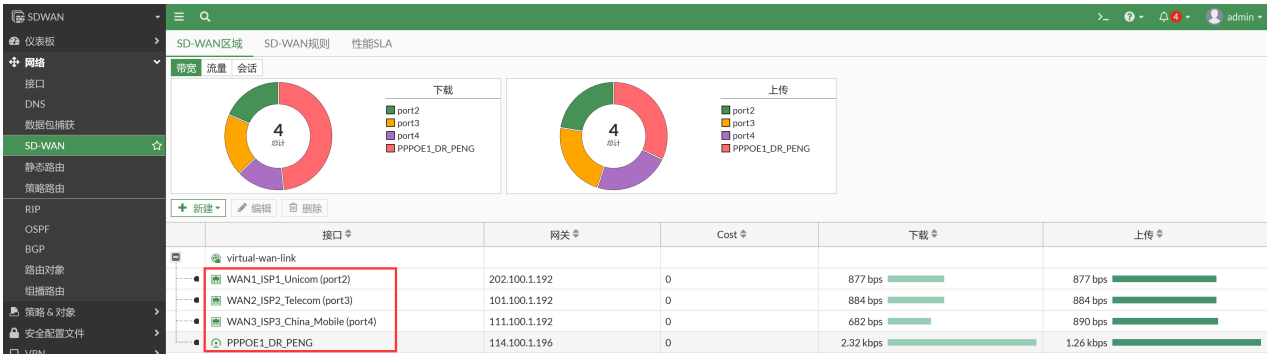
配置举例

网络拓扑



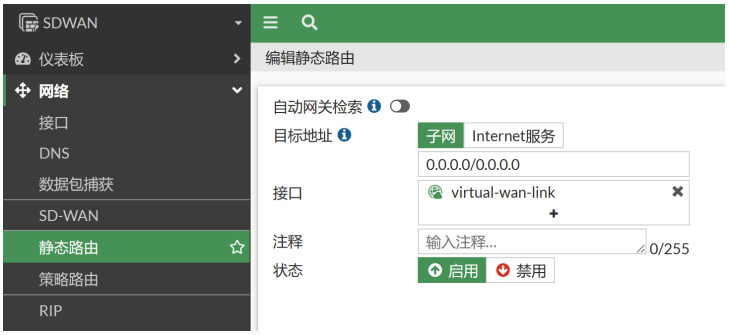
配置步骤

1. 不需要配置 SD-WAN 成员接口的预估带宽，当 Maximize Bandwidth (SLA) 类型的 SD-WAN 规则使用 `round-robin` 作为 hash-mode 时（默认配置），SD-WAN 不会考虑接口的带宽使用率，SD-WAN 会以轮询的方式将流量送至满足 SLA target 的成员接口。
2. SD-WAN 接口成员定义。



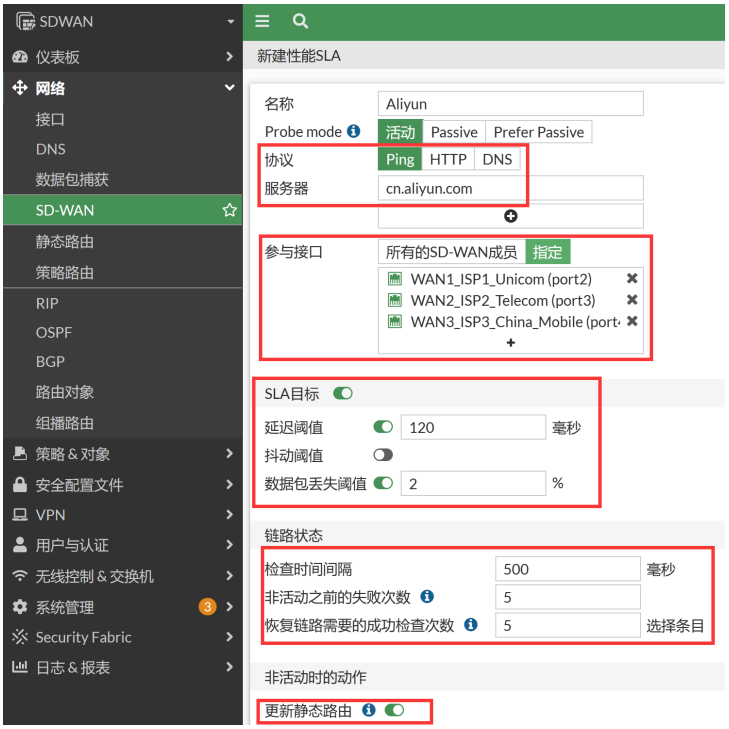

```bash
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port2"
            set gateway 202.100.1.192
        next
        edit 2
            set interface "port3"
            set gateway 101.100.1.192
        next
        edit 3
            set interface "port4"
            set gateway 111.100.1.192
        next
        edit 4
            set interface "PPPOE1_DR_PENG"
        next
    end
 end
```

3. 配置 SD-WAN 关联的默认路由。

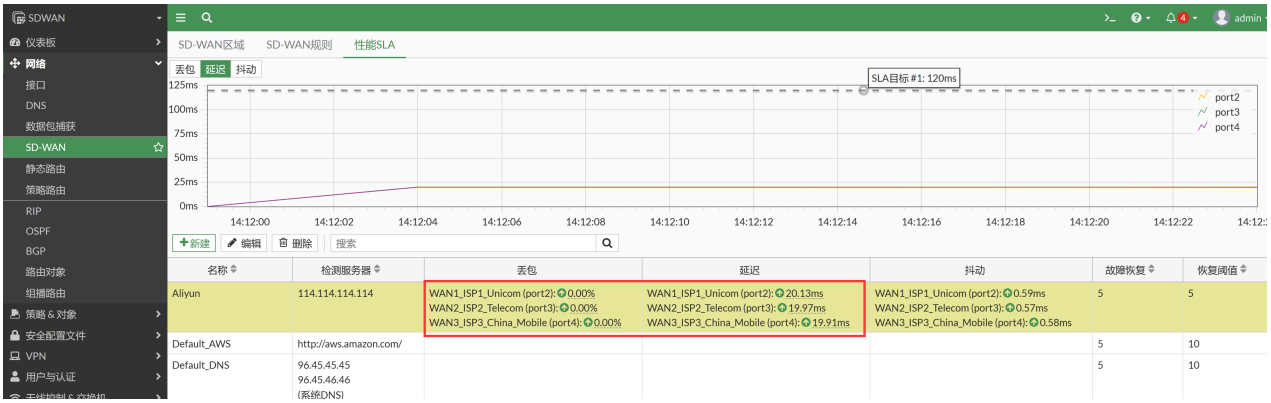



```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

4. 配置健康检查，SLA 目标监控阿里云的延迟及丢包率。




```
config system sdwan
config health-check
    edit "Aliyun"
        set server "cn.aliyun.com"
        set members 1 2 3
        config sla
            edit 1
                set link-cost-factor latency packet-loss
                set latency-threshold 120
                set packetloss-threshold 2
            next
        end
    next
end
```



5. 配置 SD-WAN 规则，目标为阿里云相关 Internet 服务，引用上步配置的 SLA Target，默认的负载方式为轮询，负载方式只能在 CLI 下修改。




```
config system sdwan
config service
    edit 1
        set name "To_Aliyun"
        set mode load-balance
        set hash-mode round-robin     //默认负载方式为轮询，即基于会话的负载
        set src "LAN_192.168.10.0"
        set internet-service enable
        set internet-service-name "Alibaba-Alibaba.Cloud" "Alibaba-DNS" "Alibaba-ICMP" "Alibaba-NTP" "Alibaba-SSH"
"Alibaba-Web"
        config sla
            edit "Aliyun"
                set id 1
            next
        end
        set priority-members 1 2 3
    next
end
```

6. 配置安全策略允许 SD-WAN 流量。

重要

如果使用 ippool 的方式做 SNAT，在 SD-WAN 环境下，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。

```
config firewall ippool
       edit xxxx
           set associated-interface port2
       next
end
```




```
config firewall policy
    edit 1
        set name "To_Internet"
        set srcintf "port8"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "default"
        set application-list "default"
        set logtraffic all
        set nat enable
    next
end
```

结果验证

1. 查看健康检查状态，三条链路均为 alive 状态。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(29.467), jitter(1.361), bandwidth-up(9999998), bandwidth-
dw(9999998), bandwidth-bi(19999996) sla_map=0x1
Seq(2 port3): state(alive), packet-loss(0.000%) latency(29.235), jitter(1.343), bandwidth-up(9999997), bandwidth-
dw(9999995), bandwidth-bi(19999992) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(29.169), jitter(1.304), bandwidth-up(9999998), bandwidth-
dw(9999998), bandwidth-bi(19999996) sla_map=0x1
```

2. 查看 SD-WAN 规则状态。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance  hash-mode=round-robin)
  Members(3):
    1: Seq_num(1 port2), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(2 port3), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(3 port4), alive, sla(0x1), gid(2), num of pass(1), selected
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

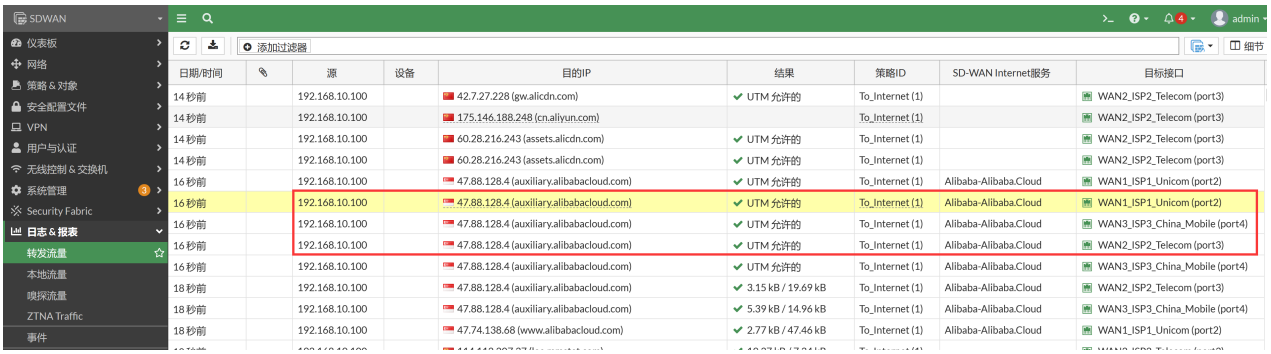
3. 查看策略路由中 SD-WAN 规则的列表，符合 SLA 目标的接口就会用于基于会话的负载均衡。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2133458945(0x7f2a0001) vwl_service=1(To_Aliyun) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x10 load-balance hash-
mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2)
num_pass=1 oif=5(port3) num_pass=1 oif=6(port4) num_pass=1
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=25 last_used=2023-01-03 22:49:28
```

4. 查看路由表。

```
SDWAN # get router info routing-table all
 ...

S*      0.0.0.0/0 [1/0] via 101.100.1.192, port3, [1/0]
                  [1/0] via 111.100.1.192, port4, [1/0]
                  [1/0] via 114.100.1.196, PPPOE1_DR_PENG, [1/0]
                  [1/0] via 202.100.1.192, port2, [1/0]
 ...
```

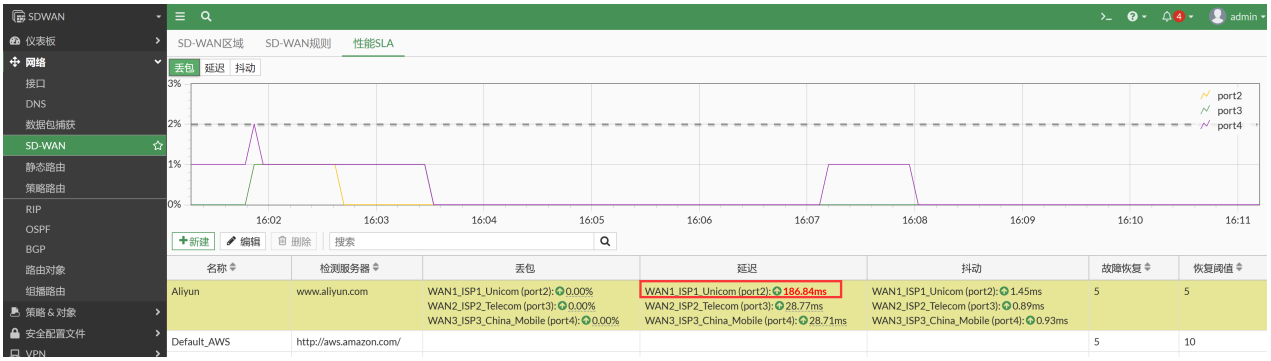
5. 查看 SD-WAN 设备的流量日志，可以看到客户端访问阿里云相关的业务时，会根据轮询算法负载到三条线路上。



线路切换测试

1. 让 port2 的延迟超过 SLA-Target。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(187.104), jitter(0.777), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0    //超过SLA的目标值
Seq(2 port3): state(alive), packet-loss(0.000%) latency(28.857), jitter(0.705), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
Seq(3 port4): state(alive), packet-loss(0.000%) latency(28.824), jitter(0.750), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x1
```



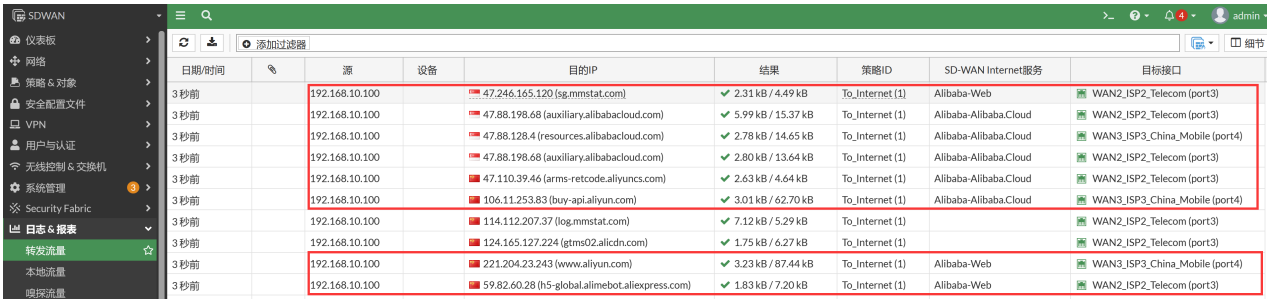
2. 查看 SD-WAN 规则状态，port2 不符合 SLA 目标，从接口列表中剔除，SLA 置位为 0x0。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance  hash-mode=round-robin)
  Members(3):
    1: Seq_num(2 port3), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(3 port4), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(1 port2), alive, sla(0x0), gid(2), num of pass(0), selected  //不符合SLA目标，从接口列表中剔除，SLA置位为0x0
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

3. 查看策略路由列表，port2 的 num_pass 为 0，不用于数据转发。

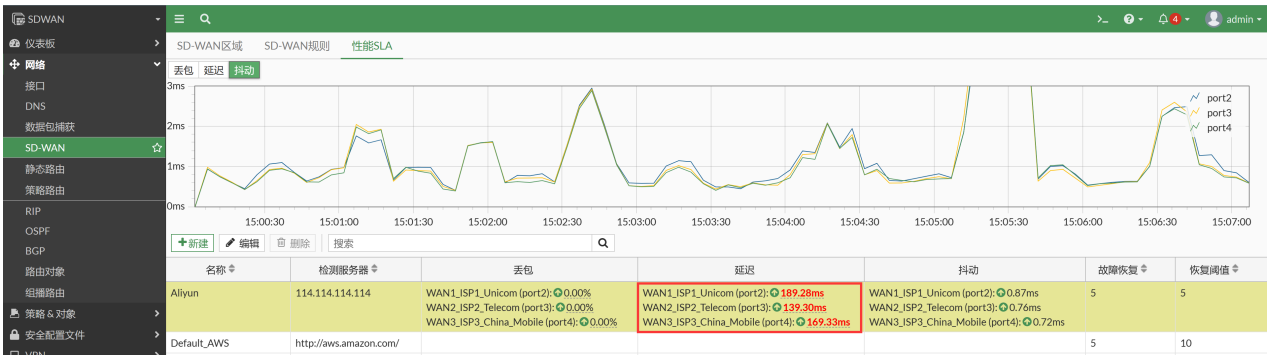
```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2133458945(0x7f2a0001) vwl_service=1(To_Aliyun) vwl_mbr_seq=2 3 1 dscp_tag=0xff 0xff flags=0x10 load-balance hash-
mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=5(port3)
num_pass=1 oif=6(port4) num_pass=1 oif=4(port2) num_pass=0    //port2不用于数据转发
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=86 last_used=2023-01-04 00:02:56
```

4. 查看 SD-WAN 设备的流量日志，客户端访问阿里云的流量被负载均衡到 port3 和 port4 上，不会负载到 port2 上。



5. 如果 port2、port3、port4 三者都无法满足 SLA 目标了，那么 SD-WAN 规则如何选择出接口呢？将三条链路的延迟均调整为无法满足 SLA Target。

```
SDWAN # diagnose sys sdwan health-check
Health Check(Aliyun):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(182.008), jitter(2.069), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(139.951), jitter(2.078), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0
Seq(3 port4): state(alive), packet-loss(0.000%) latency(168.968), jitter(2.055), bandwidth-up(9999999), bandwidth-
dw(9999999), bandwidth-bi(19999998) sla_map=0x0
```



6. 答案是：如果三者都无法满足 SLA 目标值了，那么还是将会以 port2、port3、port4 进行基于会话的负载均衡转发。

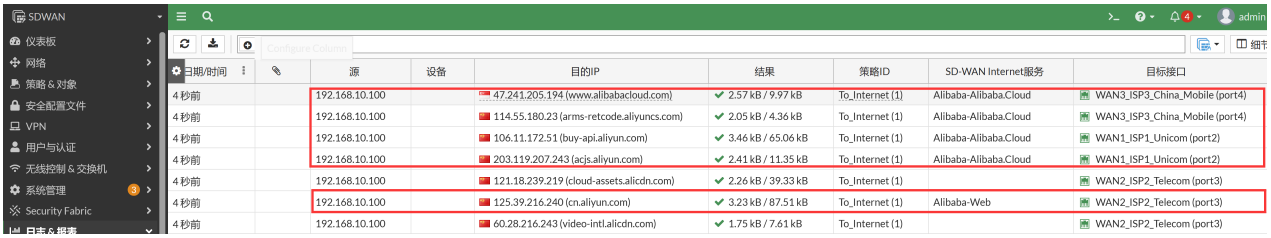
7. 查看 SD-WAN 规则状态，三条链路均被选中。

```
SDWAN # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance  hash-mode=round-robin)
  Members(3):
    1: Seq_num(1 port2), dead, sla(0x0), gid(1), num of pass(0), selected
    2: Seq_num(2 port3), alive, sla(0x0), gid(2), num of pass(0), selected
    3: Seq_num(3 port4), alive, sla(0x0), gid(2), num of pass(0), selected
  Internet Service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

8. 查看策略路由列表。

```
SDWAN # diagnose firewall proute list
list route policy info(vf=root):

id=2133458945(0x7f2a0001) vwl_service=1(To_Aliyun) vwl_mbr_seq=1 2 3 dscp_tag=0xff 0xff flags=0x10 load-balance hash-
mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=4(port2)
num_pass=0 oif=5(port3) num_pass=0 oif=6(port4) num_pass=0
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(6): Alibaba-Alibaba.Cloud(6881402,0,0,0) Alibaba-DNS(6881283,0,0,0) Alibaba-ICMP(6881282,0,0,0)
Alibaba-NTP(6881288,0,0,0) Alibaba-SSH(6881286,0,0,0) Alibaba-Web(6881281,0,0,0)
hit_count=25 last_used=2023-01-03 22:49:28
```

9. 查看 SD-WAN 设备的流量日志，依旧会在三条链路进行负载均衡。



基于带宽的 Maximize Bandwidth (SLA)

> **重要**
>
> ADVPN 目前不支持 Maximize Bandwidth (SLA) 模式。

原理介绍

1. FGT 监控每个接口的带宽，每秒更新一次，SDWAN 成员的优先级根据可用带宽从高到低排序。当流量匹配到该规则时，FGT 将选择第一个成员进行转发。

2. Maximize Bandwidth 有三种基于带宽的算法：

```bash
FortiGate # config system sdwan
FortiGate # config service
FortiGate # edit 1
FortiGate # set hash-mode ?

round-robin             All traffic are distributed to selected interfaces in equal portions and circular order.
source-ip-based         All traffic from a source IP is sent to the same interface.
source-dest-ip-based    All traffic from a source IP to a destination IP is sent to the same interface.
inbandwidth             All traffic are distributed to a selected interface with most available bandwidth for incoming
traffic.
outbandwidth            All traffic are distributed to a selected interface with most available bandwidth for outgoing
traffic.
bibandwidth             All traffic are distributed to a selected interface with most available bandwidth for both
incoming and outgoing traffic.
```

网络拓扑



配置步骤

1. 配置预估带宽。

```bash
config system interface
    edit "port1"
        set ip 202.101.1.1 255.255.255.0
        set estimated-upstream-bandwidth 200000
        set estimated-downstream-bandwidth 200000
    next
    edit "port2"
        set ip 202.101.2.1 255.255.255.0
        set estimated-upstream-bandwidth 100000
        set estimated-downstream-bandwidth 100000
    next
end
```

2. 配置 SD-WAN。

```bash
config system sdwan
    set status enable
    config zone
        edit "Internet"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "Internet"
            set gateway 202.101.1.2
```

```
        next
        edit 2
            set interface "port2"
            set zone "Internet"
            set gateway 202.101.2.2
        next
    end
    config health-check
        edit "100.100.100.1"
            set server "100.100.100.1"
            set members 0
            config sla
                edit 1
                    set latency-threshold 100
                    set jitter-threshold 100
                next
            end
        next
    end
config service
    edit 1
        set name "test1"
        set mode load-balance
        set hash-mode inbandwidth
        set dst "all"
        set src "10.0.0.0/8"
        config sla
            edit "100.100.100.1"
                set id 1
            next
        end
        set priority-members 1 2
        next
    end
end
```

3. 配置上网策略。

```
config firewall policy
    edit 3
        set srcintf "port3"
        set dstintf "Internet"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```
bash

负载方法说明

1. 当 hash-mode 是 inbandwidth 时，sdwan 只监控 inbandwith 的带宽，可以在接口下只配置：estimated-downstream-bandwidth。如下所示，初始 port1 剩余带宽 200M，port2 剩余带宽 100M，port1 优先。

```
FortiGate # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
Members(2):
1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 199991Kbps, selected  //port1优先//
2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 99993Kbps, selected
Src address(1):
10.0.0.0-10.255.255.255

Dst address(1):
0.0.0.0-255.255.255.255
```
bash

2. 当 hash-mode 是 outbandwidth 时，sdwan 只监控 outbandwith 的带宽，可以在接口下只配置：estimated-upstream-bandwidth。

```bash
diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200
Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
Members(2):
1: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 199997Kbps, selected //port1优先//
2: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
Src address(1):
10.0.0.0-10.255.255.255

Dst address(1):
0.0.0.0-255.255.255.255
```

3. 当 hash-mode 是 bibandwidth 时，sdwan 监控的是 inbandwidth 和 outbandwith 的带宽之和，需要配置：estimated-upstream-bandwidth 和 estimated-downstream-bandwidth。

```bash
diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0×200
Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=bibandwidth)
Members(2):
1: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), bibandwidth: 399988Kbps, selected //port1优先//
2: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), bibandwidth: 199992Kbps, selected
Src address(1):
10.0.0.0-10.255.255.255

Dst address(1):
0.0.0.0-255.255.255.255
```

效果验证

iPerf 测试

1. 使用 iperf 测试，hash-mode 设置为 outbandwidth，因为 iperf 打流是 upload。

```bash
server命令：iperf3 -s   //默认监听端口5201//
client命令：iperf3 -c 100.100.100.2 -t 6000 -b 150M
```

2. iperf 第 1 条流 150M。

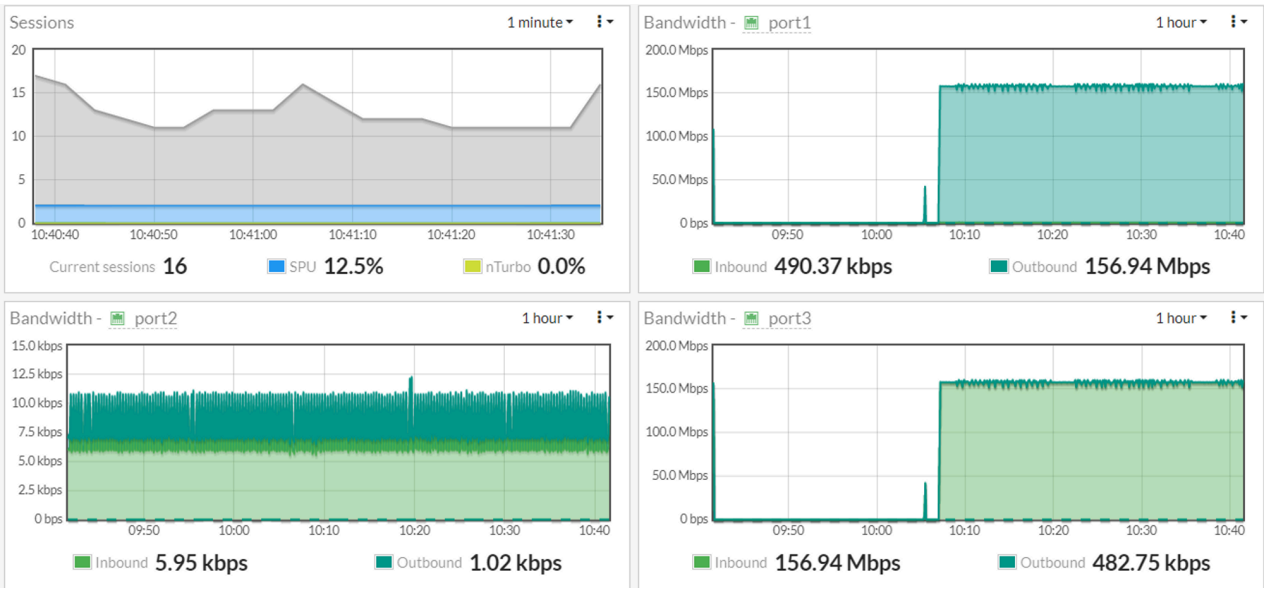
```
# iperf3 -c 100.100.100.2 -b 150M -t 6000  //client记录
Connecting to host 100.100.100.2, port 5201
[  4] local 10.1.1.2 port 50470 connected to 100.100.100.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  16.2 MBytes   136 Mbits/sec    0    389 KBytes
[  4]   1.00-2.00   sec  17.9 MBytes   150 Mbits/sec    0    389 KBytes
[  4]   2.00-3.00   sec  17.9 MBytes   150 Mbits/sec    0    389 KBytes
[  4]   3.00-4.00   sec  17.9 MBytes   150 Mbits/sec    0    389 KBytes
[  4]   4.00-5.00   sec  17.9 MBytes   150 Mbits/sec    0    400 KBytes
[  4]   5.00-6.00   sec  17.9 MBytes   150 Mbits/sec    0    400 KBytes
[  4]   6.00-7.00   sec  17.9 MBytes   150 Mbits/sec    0    400 KBytes
[  4]   7.00-8.00   sec  17.9 MBytes   150 Mbits/sec    0    400 KBytes
```

3. SDWAN 监控，手动每隔 10s，敲一次 `diagnose sys sdwan service` 查看接口的带宽及 SDWAN 的排序，SDWAN 逐渐监测到接口使用了 150M 的带宽，约为 60s。

```bash
# diagnose sys sdwan service  //初始
Service(1): Address Mode(IPV4) flags=0×200
  Gen(5), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 199997Kbps, selected //port1剩余的可用带宽多，port1优先
    2: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

```
# diagnose sys sdwan service //10s
Service(1): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 171669Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service //20s
Service(1): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 140229Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service //30s

Service(1): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 108768Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service //40s
Service(1): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected  //port2剩余的可用带宽
超过port1，port优先
    2: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 77328Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service  //50s
Service(1): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
    2: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 45866Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service //60s
Service(1): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected
    2: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 42754Kbps, selected //1分钟的时间，port1的
带宽监测到使用了150M，剩余40多M
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 查看接口流量图和会话，第 1 条 iperf 的流从 port1 转发。



5. 查看会话列表。

```bash
# diagnose sys session list
session info: proto=6 proto_state=01 duration=1534 expire=3590 timeout=3600 flags=00000000 socktype=0 sockport=0
av_idx=0 use=3
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->9/9->11 gwy=202.101.1.2/10.1.1.2
hook=pre dir=org act=noop 10.1.1.2:50482->100.100.100.2:5201(0.0.0.0:0)
hook=post dir=reply act=noop 100.100.100.2:5201->10.1.1.2:50482(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=4 auth_info=0 chk_client_info=0 vd=0
serial=00731dbb tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=1 sdwan_service_id=1
rpdb_link_id=ff000001 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000c00
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=132/136, ipid=136/132, vlan=0x0000/0x0000
vlifid=136/132, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=1/1
session info: proto=6 proto_state=01 duration=1534 expire=2065 timeout=3600 flags=00000000 socktype=0 sockport=0
av_idx=0 use=3
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->9/9->11 gwy=202.101.1.2/10.1.1.2
hook=pre dir=org act=noop 10.1.1.2:50480->100.100.100.2:5201(0.0.0.0:0)
hook=post dir=reply act=noop 100.100.100.2:5201->10.1.1.2:50480(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=4 auth_info=0 chk_client_info=0 vd=0
serial=00731dba tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=1 sdwan_service_id=1
rpdb_link_id=ff000001 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000c00
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=132/136, ipid=136/132, vlan=0x0000/0x0000
vlifid=136/132, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=5/5
total session 2

# diagnose netlink interface list  | grep port1
if=port1 family=00 type=1 index=9 mtu=1500 link=0 master=0
```

6. 使用 diagnose sys sdwan intf-sla-log port1 查看 port1 监控的带宽也是一样的效果，逐渐查看到 port1 的带宽消耗了 150M。

```bash
#diagnose sys sdwan intf-sla-log port1
Timestamp: Wed Dec 15 17:38:11 2021, used inbandwidth: 78029bps, used outbandwidth: 5949335bps, used bibandwidth:
6027364bps, tx bytes: 4837359868bytes, rx bytes: 20421280bytes.
Timestamp: Wed Dec 15 17:38:21 2021, used inbandwidth: 206010bps, used outbandwidth: 37408566bps, used bibandwidth:
37614576bps, tx bytes: 5034405643bytes, rx bytes: 21255211bytes.
```

```
Timestamp: Wed Dec 15 17:38:31 2021, used inbandwidth: 315768bps, used outbandwidth: 68591735bps, used bibandwidth:
68907503bps, tx bytes: 5229856283bytes, rx bytes: 21990459bytes.
Timestamp: Wed Dec 15 17:38:41 2021, used inbandwidth: 455947bps, used outbandwidth: 100163534bps, used bibandwidth:
100619481bps, tx bytes: 5427312580bytes, rx bytes: 22883469bytes.
Timestamp: Wed Dec 15 17:38:51 2021, used inbandwidth: 564881bps, used outbandwidth: 131613259bps, used bibandwidth:
132178140bps, tx bytes: 5624002692bytes, rx bytes: 23577318bytes.
Timestamp: Wed Dec 15 17:39:01 2021, used inbandwidth: 628638bps, used outbandwidth: 157006755bps, used bibandwidth:
157635393bps, tx bytes: 5818652090bytes, rx bytes: 24350270bytes.
```

7. iperf 打第 2 条流 80M。

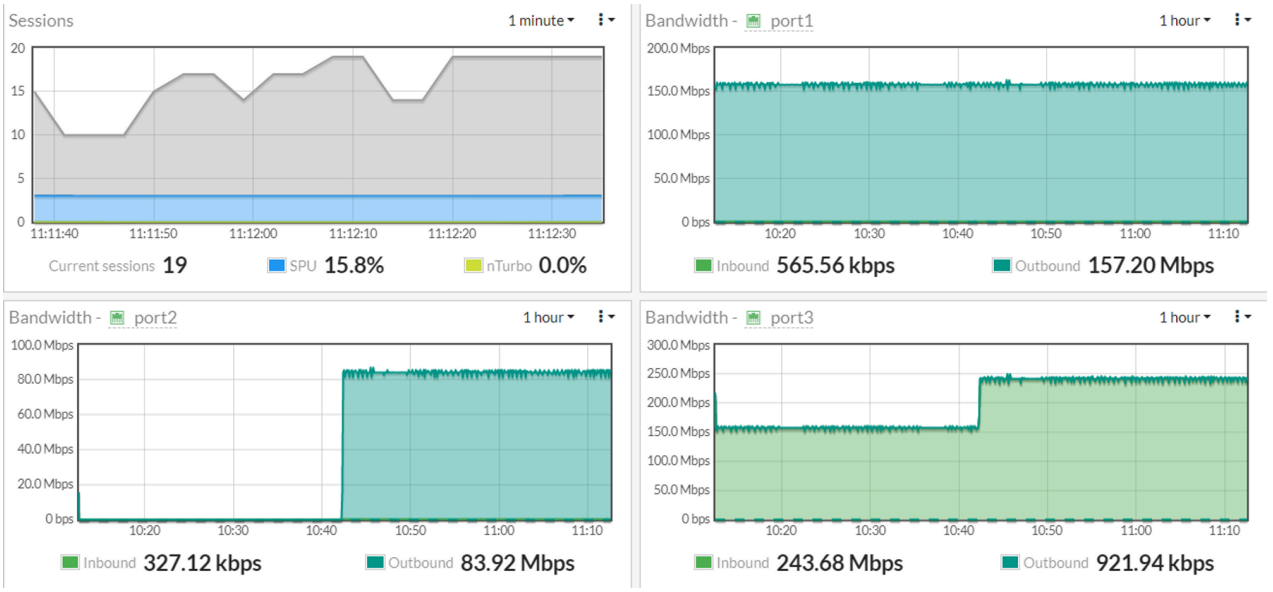
```bash
server命令：iperf3 -s -p 5202
client命令：iperf3 -c 100.100.100.2 -t 6000 -b 80M -p 5202
```

8. SDWAN 监控，查看接口的带宽及 SDWAN 的排序。

```bash
# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×200
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 99999Kbps, selected //开始打第2条流前,
port2优先
    2: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 42732Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

 # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×200
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 76158Kbps, selected
    2: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 42754Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

 # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×200
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 59404Kbps, selected
    2: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 43062Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

 # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×200
  Gen(6), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 42453Kbps, selected
    2: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 42424Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

 # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0×200
  Gen(7), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 42832Kbps, selected  //port1剩余的可用带宽
多，port1变优先
    2: Seq_num(2 port2), alive, sla(0×1), gid(2), num of pass(1), outbandwidth: 25853Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
```

```
        0.0.0.0-255.255.255.255

 # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(7), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=outbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 42832Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), outbandwidth: 25853Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

9. 查看接口流量图和会话，第 2 条 iperf 流从 port2 转发。第 1 条 iperf 流无变化。



.0. 查看会话列表。

```
# diagnose sys session list                                                
```

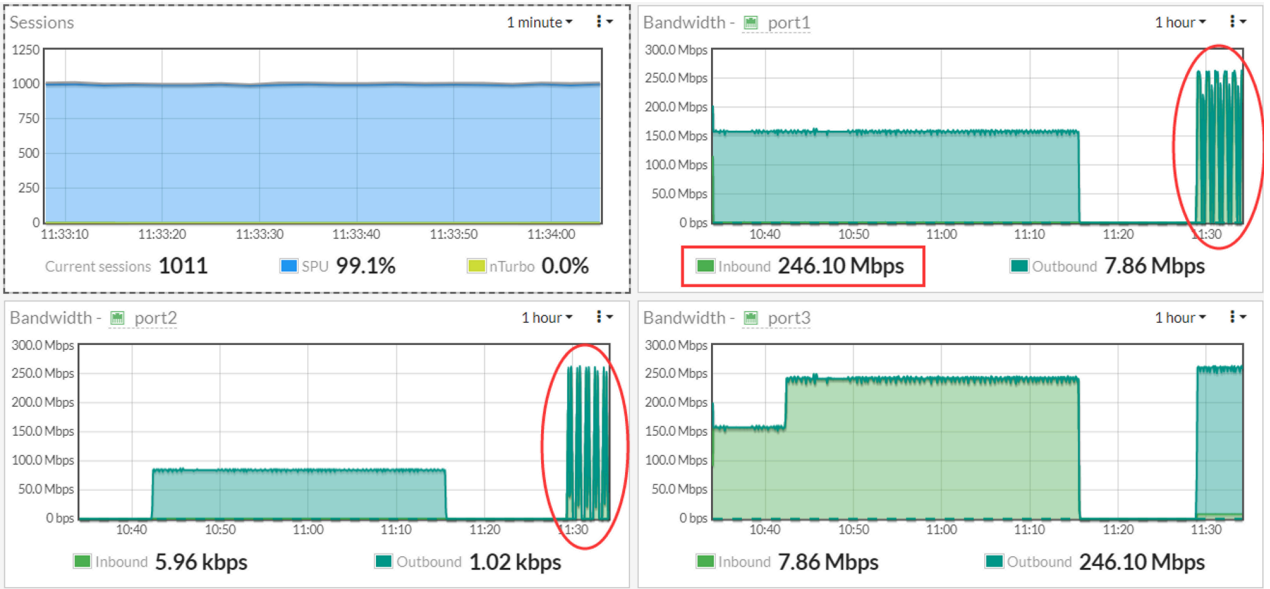
```
hook=pre dir=org act=noop 10.1.1.3:59560→100.100.100.2:5202(0.0.0.0:0)
hook=post dir=reply act=noop 100.100.100.2:5202→10.1.1.3:59560(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=4 auth_info=0 chk_client_info=0 vd=0
serial=0073229c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=2 sdwan_service_id=1
rpdb_link_id=ff000001 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000c00
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=134/136, ipid=136/134, vlan=0x0000/0x0000
vlifid=136/134, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=6/6
total session 2

# diagnose netlink interface list | grep port2
if=port2 family=00 type=1 index=10 mtu=1500 link=0 master=0
```

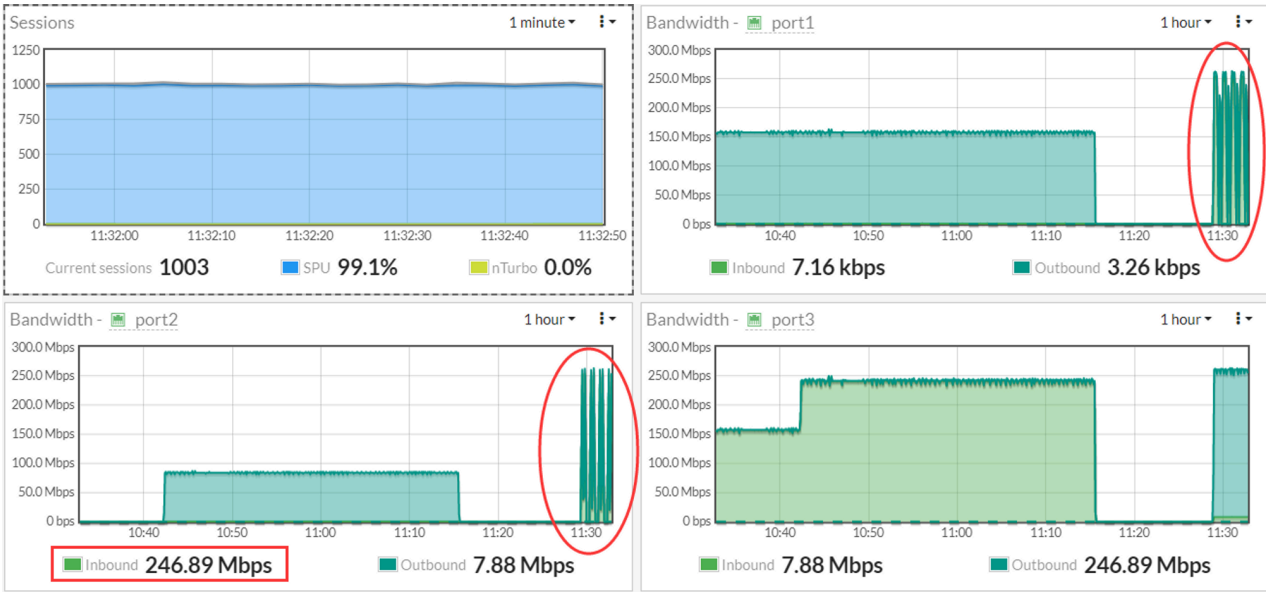
FortiTester 测试

1. hash-mode 设置为 inbandwidth，因为 iperf 打流是 download，cps = 1000，get 30kb。

```
# diagnose sys sdwan service                                                    bash

Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1→65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 199991Kbps, selected //port1优先//
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 99993Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

2. 从接口流量图看，流量一会从 port1 走，一会从 port2，没有负载到两条链路，是由于 FortiTester 打流新建会话和拆除会话太快，SDWAN 带宽监控反应不过来。





3. 对应的 SD-WAN 服务状态。

```bash
# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(13), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 48600Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 0Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(13), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 48600Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 0Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(14), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 48225Kbps, selected
    2: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 554Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(14), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 48225Kbps, selected
    2: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 554Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

真实流量测试

1. 使用迅雷下载一部电影，速度平均 5.2MB/s 左右，查看接口带宽分配的实际效果。

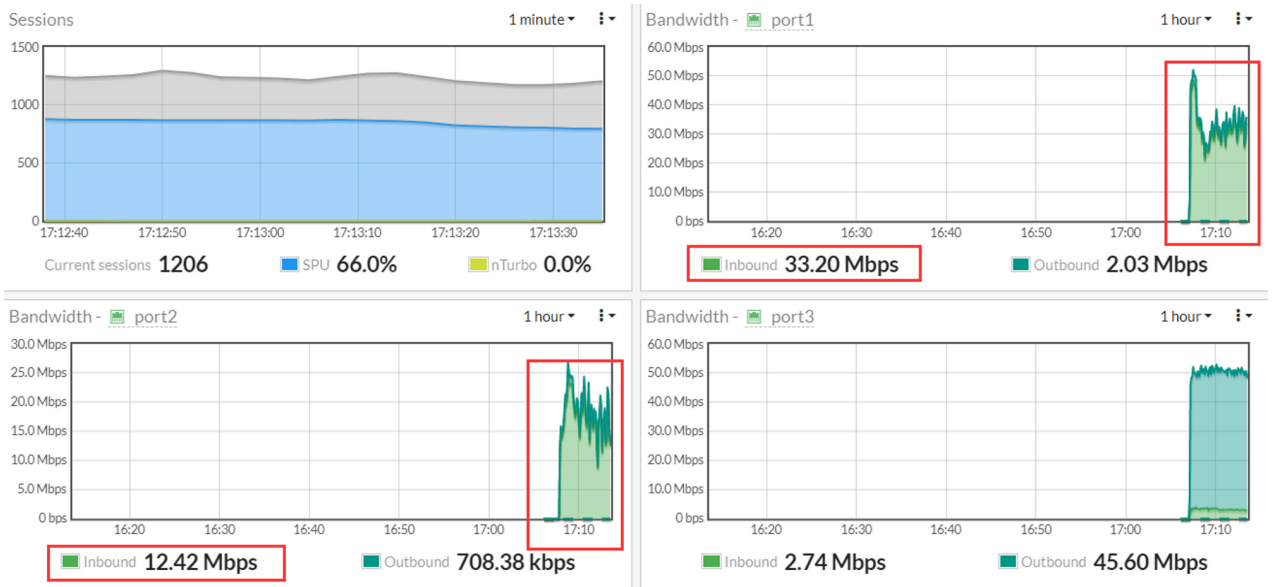
2. 公网实际带宽不大，因此将预估带宽调小，hash-mode 使用 inbandwidth。

```bash
config system interface
    edit "port1"
        set ip 202.101.1.1 255.255.255.0
        set estimated-upstream-bandwidth 40000
        set estimated-downstream-bandwidth 40000
    next
    edit "port2"
        set vdom "root"
        set estimated-upstream-bandwidth 20000
        set estimated-downstream-bandwidth 20000
    next
end
```

3. 对应的 SD-WAN 服务状态。

```bash
# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200
  Gen(7), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=inbandwidth)
  Members(2):
    1: Seq_num(1 port1), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 39936Kbps, selected
    2: Seq_num(2 port2), alive, sla(0x1), gid(2), num of pass(1), inbandwidth: 19993Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. 对应带宽图。



总结

1. 基于带宽的 Maximize Bandwidth 适用于流量缓慢增长，多会话的场景。
2. 下面两种情况不适用：
   - 只有一条会话，且带宽很大，是不会切换的。
   - 突发流量，短时间流量很大，SD-WAN 无法及时感知。

SD-WAN 配置举例

- 将已关联接口移至 SD-WAN 区域
- 互联网出口场景
- 互联网出口和 VPN 混合场景
- ADVPN 场景

将已关联接口移至 SD-WAN 区域

网络需求

某台 FortiGate 已经部署上线，上线时没有使用 SD-WAN 功能，公网接口已被其他配置引用。现在客户想在这台 FortiGate 上开启 SD-WAN 功能，但删除接口关联的配置太过麻烦，容易出现意外，同时影响业务的时间也较长。

解决方法

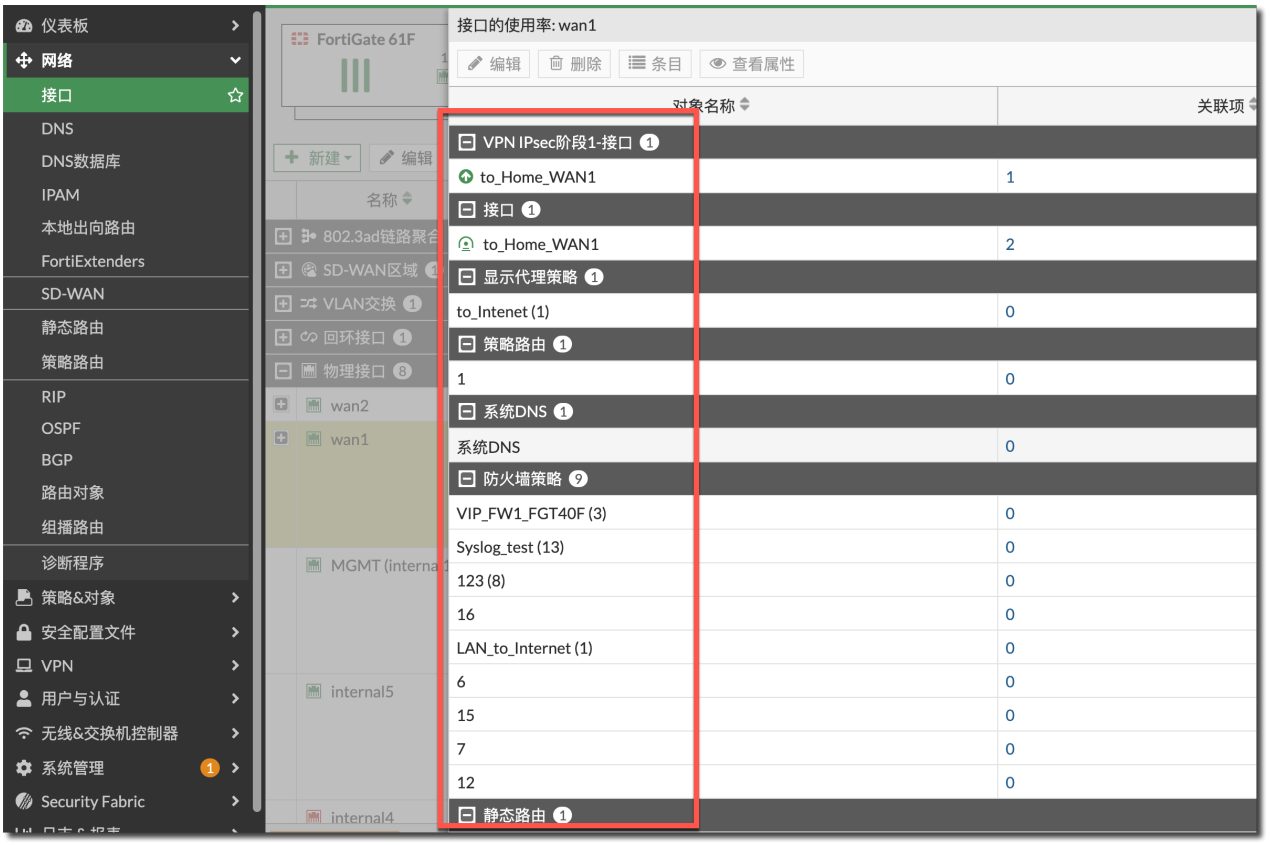
从 FortiOS 7.0 版本开始，FortiGate 支持将已被其他配置关联的接口直接加入 SD-WAN 区域（不需要删除接口关联的配置）。并自动修改关联配置中引用的接口为 SD-WAN 区域。

配置步骤

重要

如果 FortiGate 为 HA 模式，且需要加入 SD-WAN 区域的接口被 HA 监控接口引用，请先手动将该接口从 HA 监控接口移除。将接口加入 SD-WAN 区域后，再手动将接口加回 HA 监控接口。

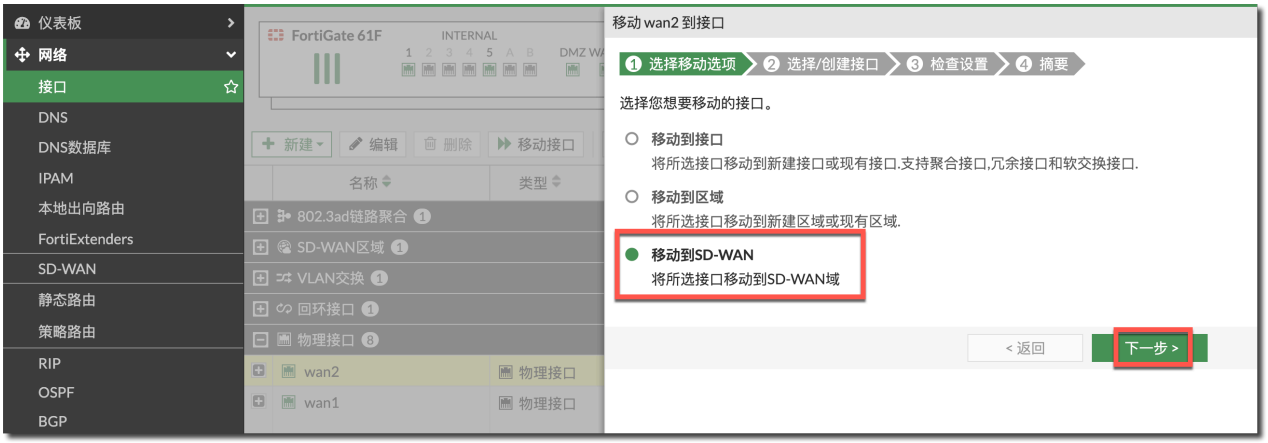
1. 在一台 wan1、wan2 接口已经被其他配置引用的 FortiGate 上，想要将 wan1 和 wan2 加入 SD-WAN 区域。查看 wan1 和 wan2 的接口引用，此时直接在 SD-WAN 区域中加入 wan1 和 wan2 是无法执行的。



2. 右键点击要加入 SD-WAN 区域的接口（这里以 wan2 为例），然后在“设置状态”中选择“移动接口”。



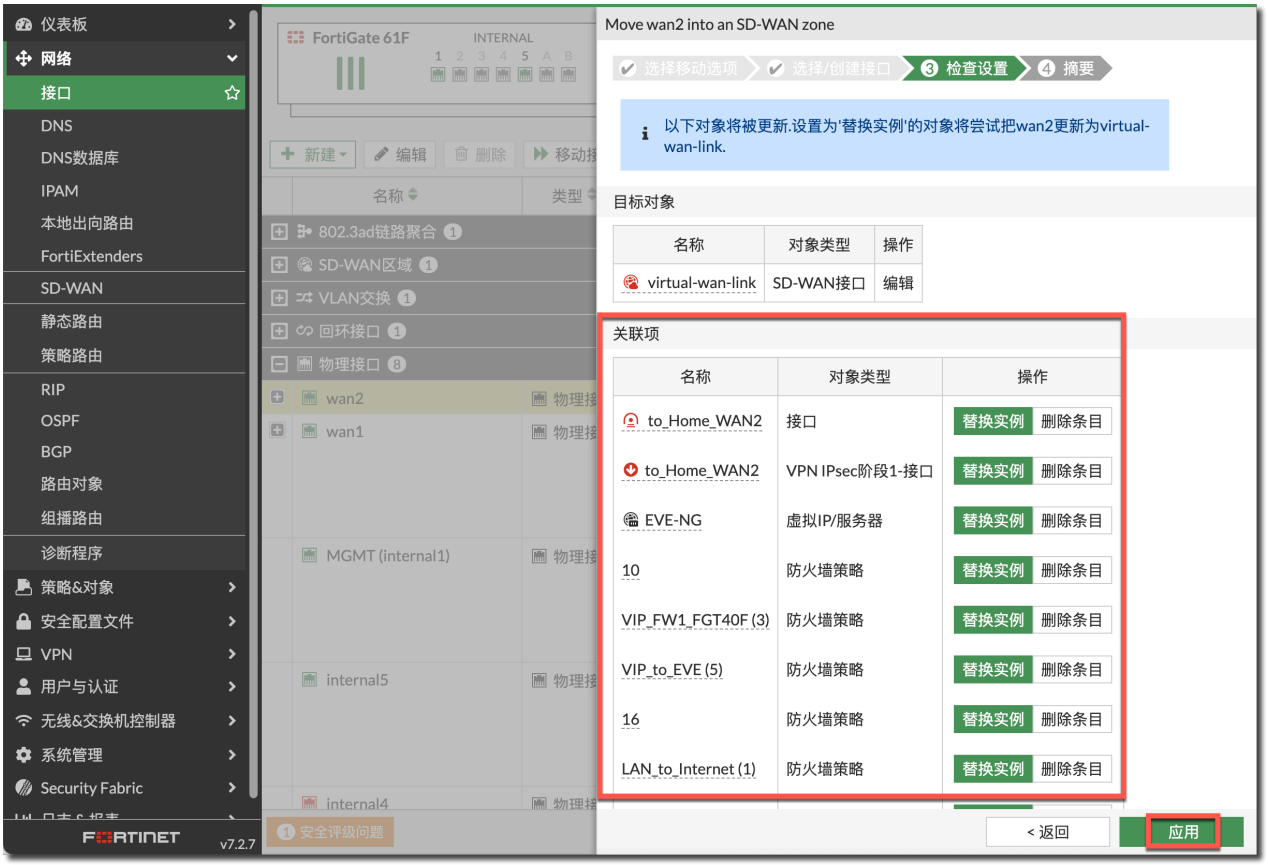
3. 在右侧弹出的窗口中，选择“移动到 SD-WAN”选项，点击下一步。



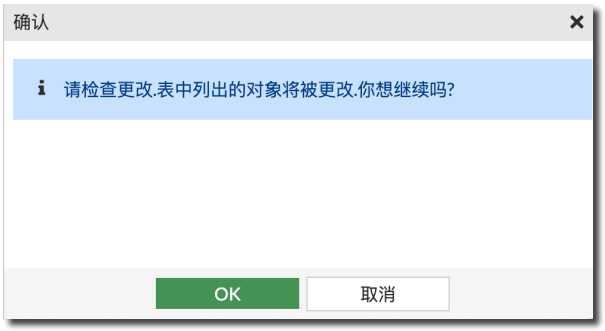
4. 选择要加入的 SD-WAN 区域，这里以“virtual-wan-link 为例”，点击下一步。



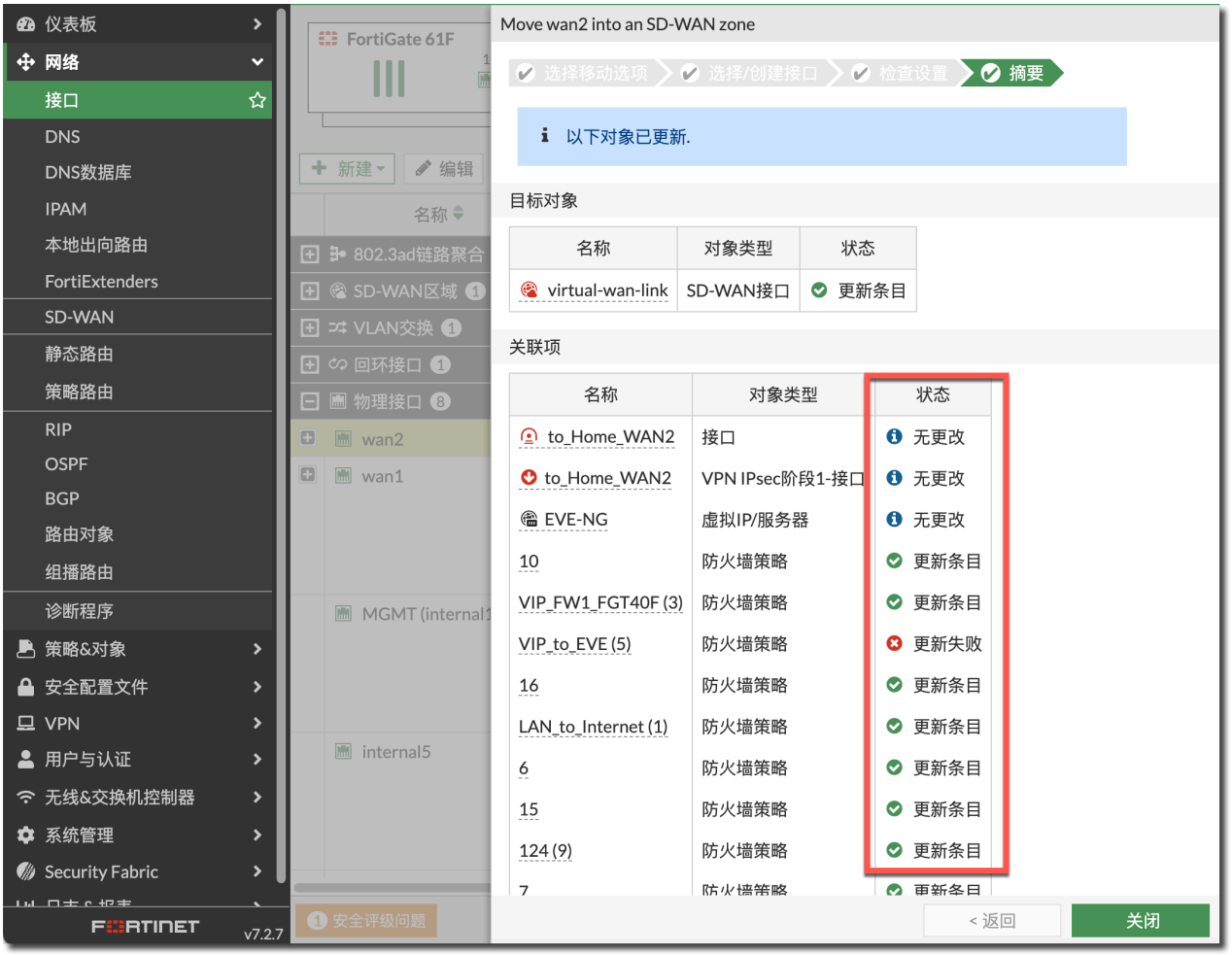
5. “检查设置”步骤中显示关联的配置中的接口将从 wan2 替换为 SD-WAN 区域 virtual-wan-link（也可以修改为“删除”操作）。检查无误后，点击应用按钮。



6. 随后会弹出确认窗口，点击 OK。



7. 执行完成后会显示执行结果，不需要取消关联的配置会显示“无更改”，已经替换为 SD-WAN 区域的配置会显示“更新条目”。



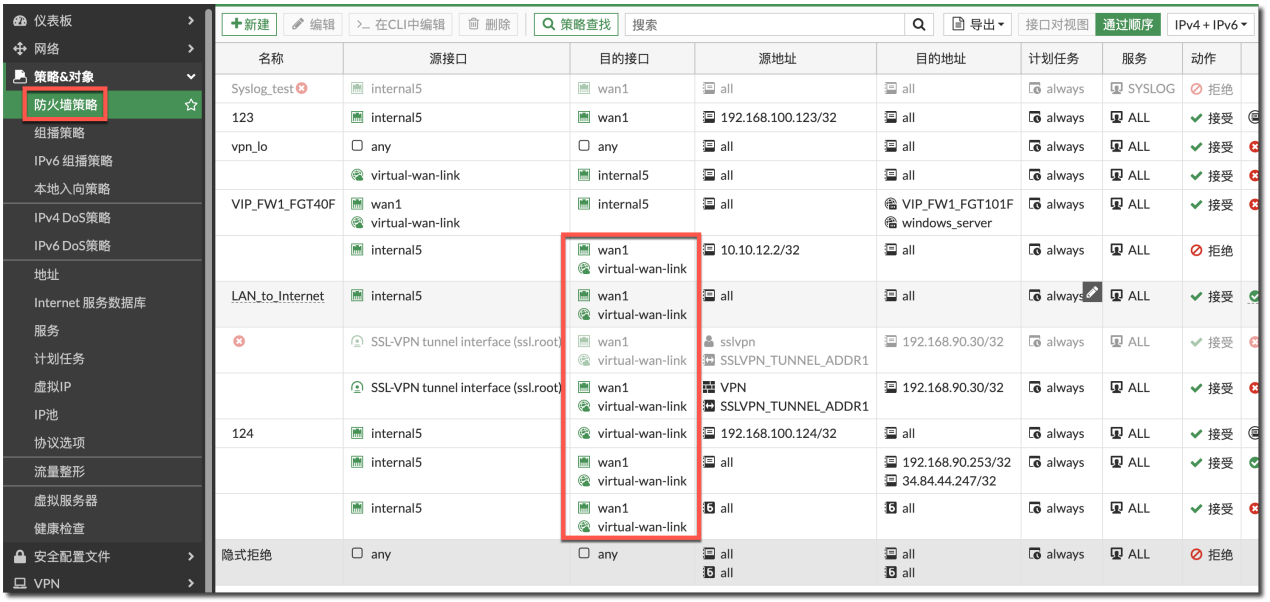
结果验证

1. 查看 SD-WAN 区域的成员，wan2 已经成功加入（需要手动配置成员的网关）。




```bash
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 2
            set interface "wan2"
        next
    end
end
```

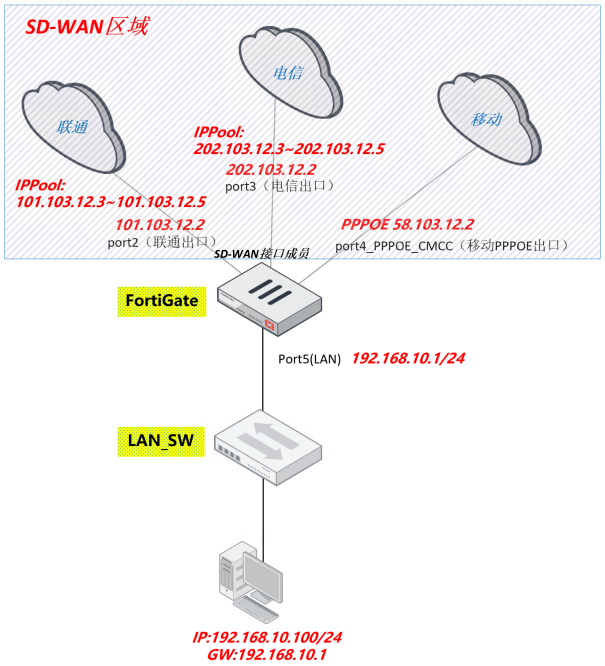
2. 查看防火墙策略，所有原来 wan2 关联的策略中，wan2 都被替换为 SD-WAN 区域“virtual-wan-link”。



3. 其他不需要替换的关联配置或没有引用 wan2 的配置不会有变化。

互联网出口场景

网络拓扑与需求

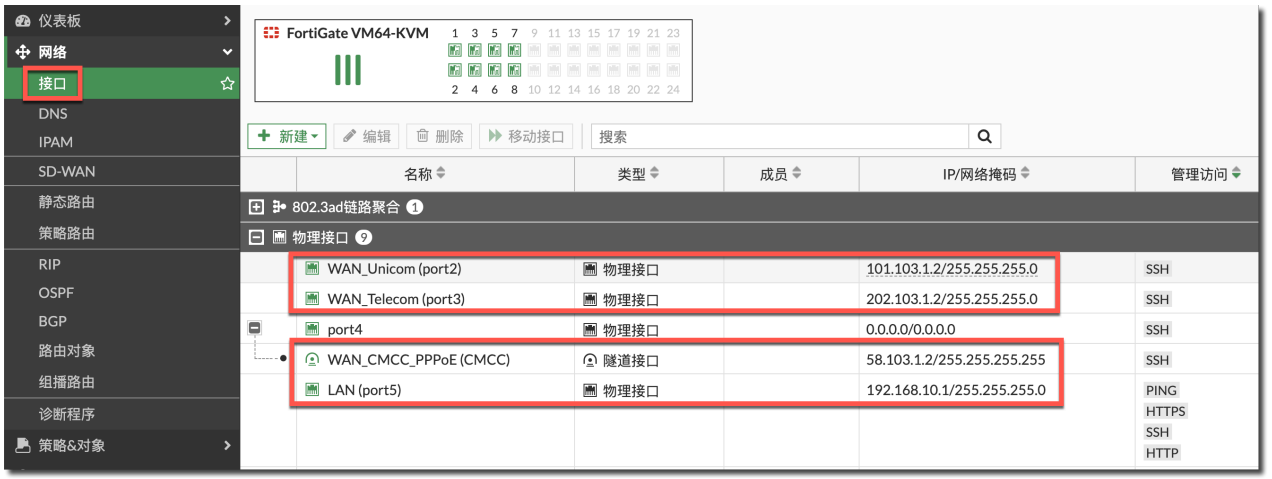


1. FortiGate 通过三个运营商（联通、电信、移动）接入 Internet，其中联通和电信线路为固定 IP，移动线路为 PPPoE 线路。
2. 联通和电信线路使用运营商分配的 IPPool 转换 SNAT 访问 Internet，移动线路借用 PPPoE 接口 IP 作为 SNAT 的源地址访问 Internet。
3. 配置 SD-WAN，基于应用、用户、ISDB 等做选路的调用，可以实现更加智能的互联网出口选择和备份，这也是 SD-WAN 的价值所在。
4. 部署 SD-WAN 通常在一个全新的设备进行配置，如果需要加入 SD-WAN 区域的接口已经被其他配置（如防火墙策略）引用，是无法直接加入 SD-WAN 区域的。要加入 SD-WAN 区域，又不想删除关联的配置，请参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域 ↗章节。

配置步骤

SD-WAN 基础配置

1. 配置三条运营商线路的 IP 或 PPPoE 用户（PPPoE 请通过 CLI 配置，参考部署前注意事项 → PPPoE 接口配置 ↗章节）与内网口 IP。




```bash
PPPoE接口配置：
config system pppoe-interface
    edit "CMCC"
        set device "port4"
        set username "user1"
```

```
        set password xxxxxxxx
    next
end

固定IP接口配置：
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.1.2 255.255.255.0
        set allowaccess ssh
        set alias "WAN_Unicom"
    next
    edit "port3"
        set vdom "root"
        set ip 202.103.1.2 255.255.255.0
        set allowaccess ssh
        set alias "WAN_Telecom"
    next
    edit "port5"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh http
        set alias "LAN"
    next
end
```

2. 进入 FortiGate 的网络 → SD-WAN → SD-WAN 区域页面，新建用于访问 Internet 的区域“to_Internet”（也可以直接使用预置的 SD-WAN 区域）。





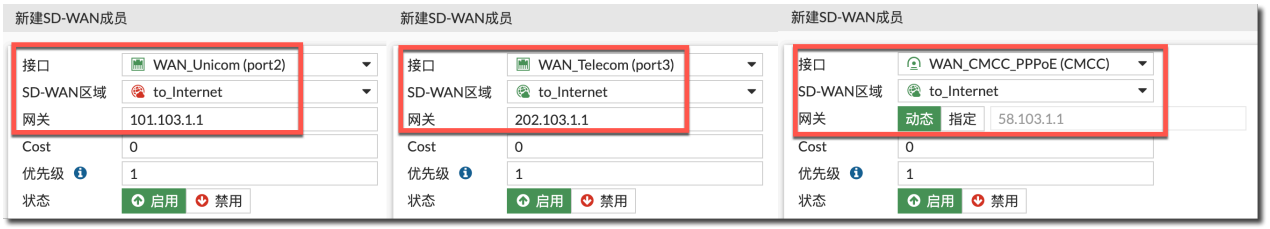


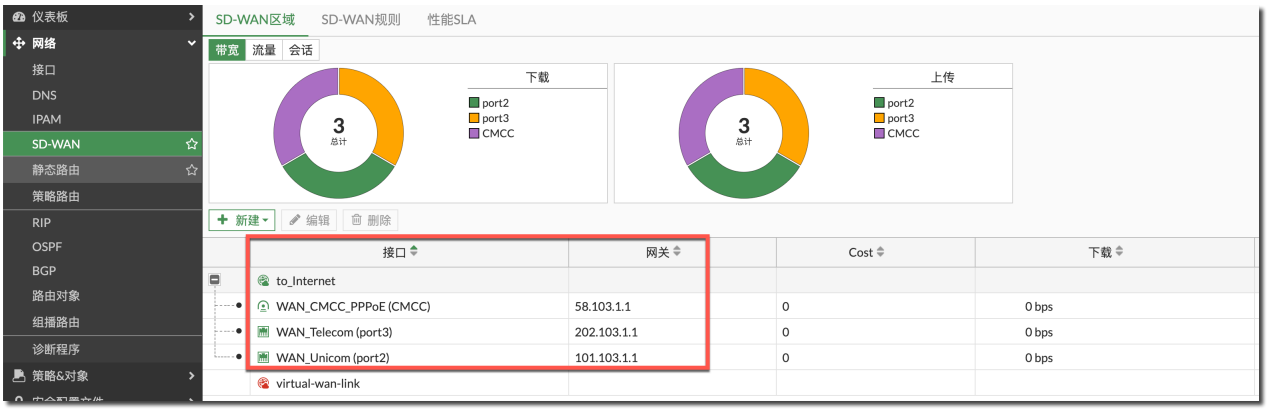
```bash
config system sdwan
    set status enable
    config zone
        edit "to_Internet"
        next
    end
end
```

3. 在此页面新建用于访问 Internet 的 SD-WAN 成员（联通、电信、移动线路 3 个接口）加入上步创建的 SD-WAN 区域，注意 PPPoE 加入 SD-WAN 区域时不需要填写网关。

> **重要**
>
> 如果 WAN 接口存在相关配置引用的情况下，是无法直接加入到 SD-WAN 接口成员的，可参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域 ↗章节快速将接口加入 SD-WAN 区域，并自动修改关联配置（也可选择手动将 WAN 接口的相关配置删除引用，包括策略、Link-monitor、所属区域等）。








```bash
config system sdwan
    config members
        edit 1
            set interface "port2"
            set zone "to_Internet"
            set gateway 101.103.1.1
        next
        edit 2
            set interface "port3"
            set zone "to_Internet"
            set gateway 202.103.1.1
        next
```

```
        edit 3
            set interface "CMCC"
            set zone "to_Internet"
        next
    end
end
```

4. 配置默认路由指向 SD-WAN 区域“to_Internet”。




```
config router static
    edit 1
        set distance 1
        set sdwan-zone "to_Internet"
    next
end
```

bash

5. 查看 FortiGate 路由表，默认路由在三个运营商线路上负载。

```
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 202.103.1.1, port3, [1/0]
                  [1/0] via 58.103.1.1, CMCC, [1/0]
                  [1/0] via 101.103.1.1, port2, [1/0]
C       58.103.1.1/32 is directly connected, CMCC
C       58.103.1.2/32 is directly connected, CMCC
C       101.103.1.0/24 is directly connected, port2
C       192.168.10.0/24 is directly connected, port5
C       202.103.1.0/24 is directly connected, port3
```

bash

6. 配置联通和电信线路的 IPPool，并绑定对应的 WAN 接口。

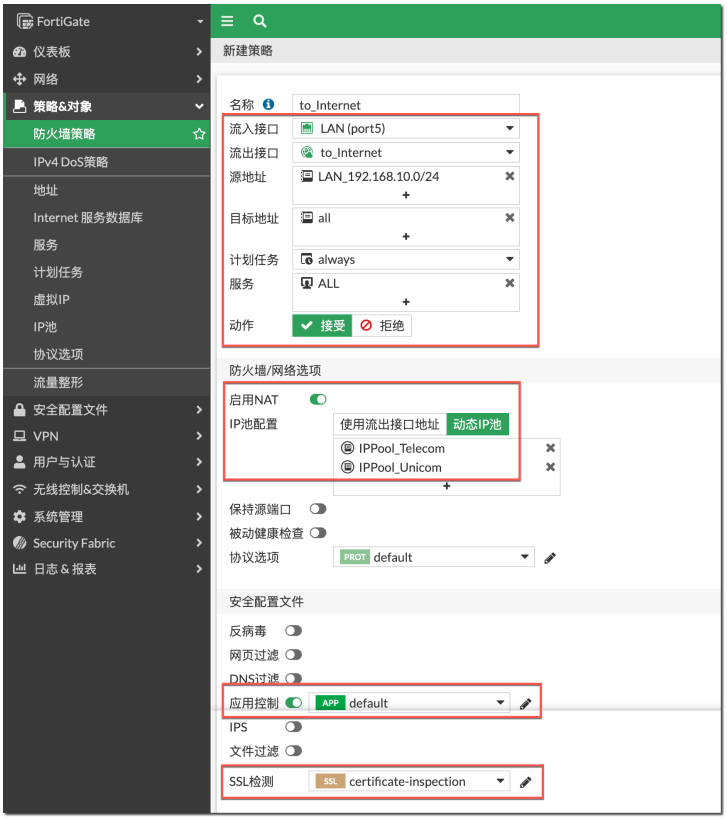
> **重要**
>
> 重要：在 SD-WAN 场景下，防火墙上网策略选用出接口时只能选择 SD-WAN 区域接口“to_Internet”（Central-NAT 模式例外），不能选择具体的 WAN 接口，所以配置 IPPool 时必须将 IPPool 与具体的 WAN 接口绑定，否则 FortiGate 只会使用防火墙策略中配置的第一个 IPPool 做 SNAT。

```
config firewall ippool
    edit "IPPool_Unicom"
        set startip 101.203.1.3
        set endip 101.203.1.5
        set associated-interface "port2"
    next
    edit "IPPool_Telecom"
        set startip 202.103.1.3
        set endip 202.103.1.5
        set associated-interface "port3"
    next
end
```

bash

7. 配置防火墙策略，放通内网 port5（LAN）到三个运营商线路的 Internet 访问，并开启 SNAT，引用联通和电信线路上的 IPPool，开启应用控制功能，用于 SD-WAN 策略基于应用的识别。

> **重要**
>
> 有了第 6 步中 IPPool 关联接口的配置，防火墙策略的 SNAT 引用 IPPool 时，会使用具体 WAN 接口上关联的 IPPool 中的地址来执行 SNAT，而移动线路（PPPoE）则仍会使用出接口地址来作为 SNAT 的源地址。

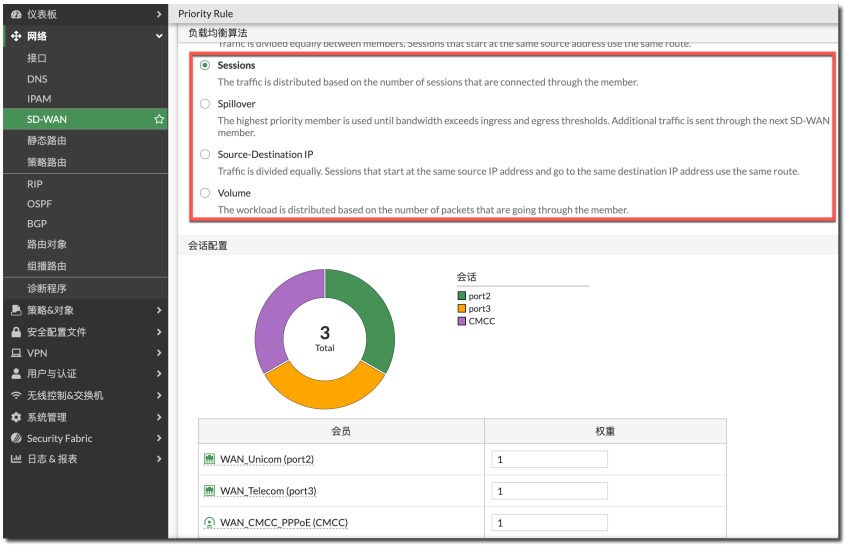



```bash
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port5"
        set dstintf "to_Internet"
        set action accept
        set srcaddr "LAN_192.168.10.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "default"
        set nat enable
        set ippool enable
        set poolname "IPPool_Telecom" "IPPool_Unicom"
    next
end
```

8. 此时 SD-WAN 的基本配置已经完成，如果不配置 SD-WAN 规则，所有的访问 Internet 的流量将会按照源 IP 负载均衡的方式，也就是我们手工配置的 SD-WAN 默认路由，按照三条默认路由进行负载均衡处理，如下默认隐式 SD-WAN 策略所示，编辑此策略，可以修改为其他负载模式。




```bash
config system sdwan
    set load-balance-mode source-ip-based
end
```



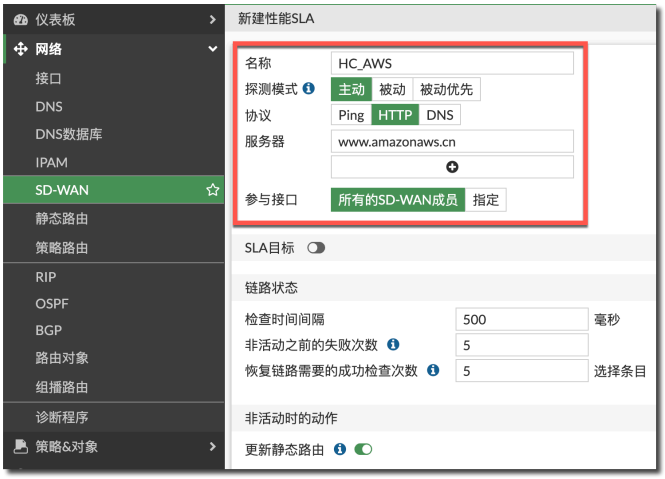
SD-WAN 策略配置

需求举例 1

去往 AWS 的重要业务从联通或电信线路（固定 IP），选择延迟较低的出口出去。

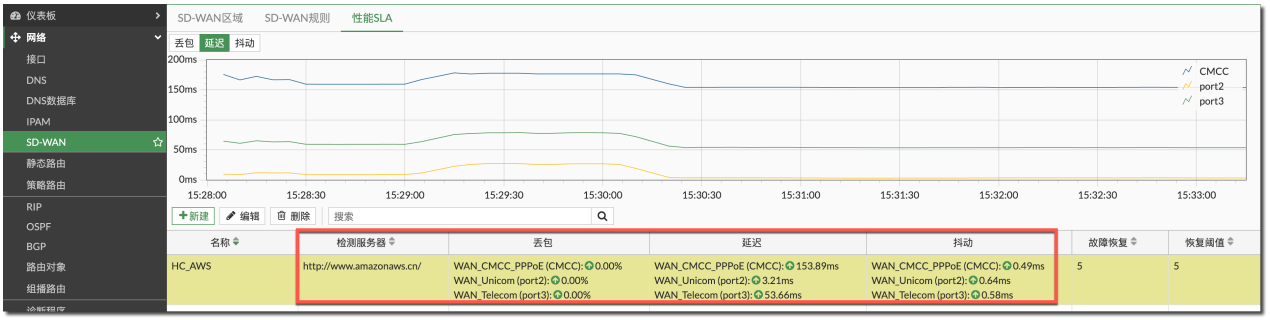
1. 配置去往 AWS 的 HTTP 健康检查。






```bash
config system sdwan
    config health-check
        edit "HC_AWS"
            set server "www.amazonaws.cn"
            set protocol http
            set members 0
        next
    end
end
```

2. 查看健康检查的状态，可以看到目前联通线路延迟最低，其次是电信线路，移动线路延迟最高。

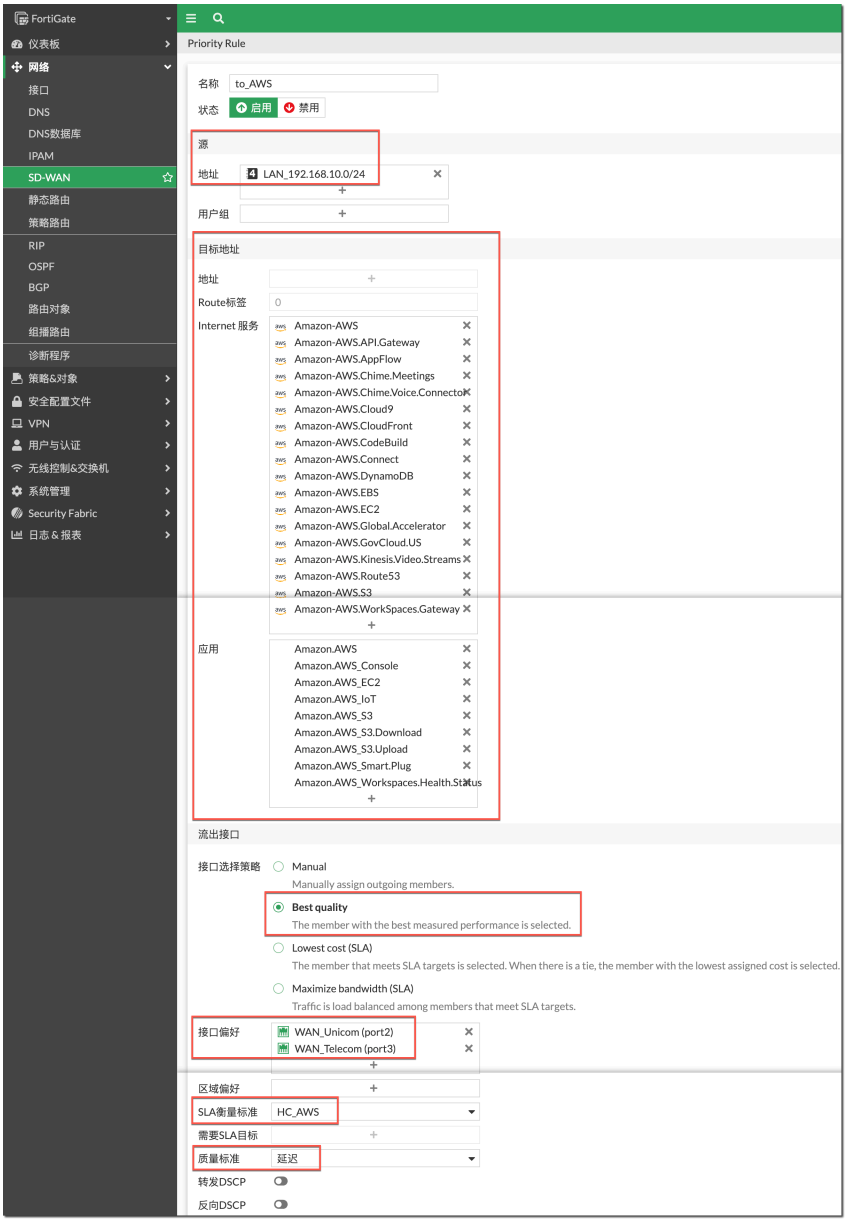



```bash
FortiGate # diagnose sys sdwan health-check status
Health Check(HC_AWS):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(3.216), jitter(0.347), mos(4.402), bandwidth-up(65534995),
bandwidth-dw(65534993), bandwidth-bi(131069988) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(53.634), jitter(0.451), mos(4.376), bandwidth-up(65534995),
bandwidth-dw(65534993), bandwidth-bi(131069988) sla_map=0x0
Seq(3 CMCC): state(alive), packet-loss(0.000%) latency(153.830), jitter(0.392), mos(4.307), bandwidth-up(65534995),
bandwidth-dw(65534993), bandwidth-bi(131069988) sla_map=0x0
```

3. 新建 SD-WAN 规则 1：

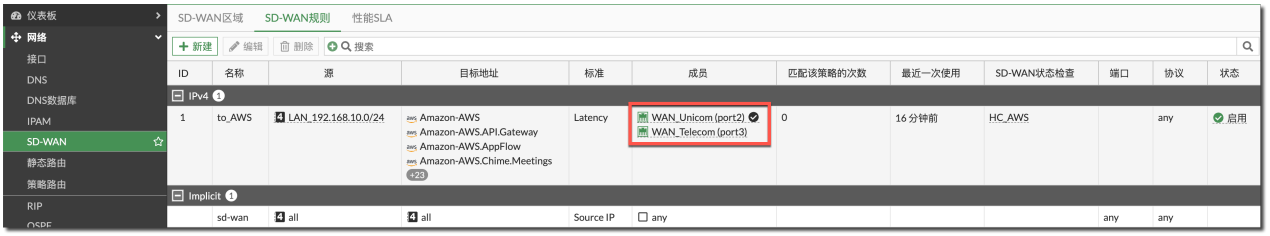
- 源地址选择内网网段；目标地址使用 ISDB，选择 AWS 相关的 ISDB 条目，以及 AWS 相关的应用控制特征库条目。
- 接口选择策略与质量标准：选择 Best Quality 和延迟，选用 SD-WAN 规则接口成员中延迟最低的接口，如果成员接口的延迟差小于 10%（默认值），则使用接口偏好中配置较前的接口。
- SLA 衡量标准：选择上步创建的健康检查。
- 接口偏好：选择联通线路接口与电信线路接口，联通线路接口排序在前。






```bash
config system sdwan
    config service
        edit 1
            set name "to_AWS"
            set mode priority
            set src "LAN_192.168.10.0/24"
            set internet-service enable
            set internet-service-name "Amazon-AWS" "Amazon-AWS.API.Gateway" "Amazon-AWS.AppFlow" "Amazon-AWS.Chime.Meetings" "Amazon-AWS.Chime.Voice.Connector" "Amazon-AWS.Cloud9" "Amazon-AWS.CloudFront" "Amazon-AWS.CodeBuild" "Amazon-AWS.Connect" "Amazon-AWS.DynamoDB" "Amazon-AWS.EBS" "Amazon-AWS.EC2" "Amazon-AWS.Global.Accelerator" "Amazon-AWS.GovCloud.US" "Amazon-AWS.Kinesis.Video.Streams" "Amazon-AWS.Route53" "Amazon-AWS.S3" "Amazon-AWS.WorkSpaces.Gateway"
            set internet-service-app-ctrl 27210 37172 36740 10000051 35944 47432 47433 10000199 51269
            set health-check "HC_AWS"
            set priority-members 1 2
        next
    end
end
```

4. 查看上步创建的 SD-WAN 规则目前的选路状态，由于联通线路（port2）的延迟较低，电信线路（port3）的延迟较高，访问 AWS 的业务优先选择联通线路（port2）。




```bash
FortiGate # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(HC_AWS)
  Members(2):
    1: Seq_num(1 port2), alive, latency: 3.135, selected    //联通线路优先
    2: Seq_num(2 port3), alive, latency: 53.637, selected   //电信线路其次
  Internet Service(27): Amazon-AWS(393320,0,0,0,0) Amazon-AWS.API.Gateway(393478,0,0,0,0) Amazon-
AWS.AppFlow(393484,0,0,0,0) Amazon-AWS.Chime.Meetings(393483,0,0,0,0) Amazon-AWS.Chime.Voice.Connector(393479,0,0,0,0)
Amazon-AWS.Cloud9(393471,0,0,0,0) Amazon-AWS.CloudFront(393481,0,0,0,0) Amazon-AWS.CodeBuild(393482,0,0,0,0) Amazon-
AWS.Connect(393480,0,0,0,0) Amazon-AWS.DynamoDB(393472,0,0,0,0) Amazon-AWS.EBS(393470,0,0,0,0) Amazon-
AWS.EC2(393477,0,0,0,0) Amazon-AWS.Global.Accelerator(393476,0,0,0,0) Amazon-AWS.GovCloud.US(393452,0,0,0,0) Amazon-
AWS.Kinesis.Video.Streams(393475,0,0,0,0) Amazon-AWS.Route53(393473,0,0,0,0) Amazon-AWS.S3(393474,0,0,0,0) Amazon-
AWS.WorkSpaces.Gateway(393403,0,0,0,0) Amazon.AWS(4294836343,0,0,0,0 27210) Amazon.AWS.Console(4294836344,0,0,0,0 37172)
Amazon.AWS_EC2(4294836345,0,0,0,0 36740) Amazon.AWS_IoT(4294838823,0,0,0,0 10000051) Amazon.AWS_S3(4294836346,0,0,0,0
35944) Amazon.AWS_S3.Download(4294838824,0,0,0,0 47432) Amazon.AWS_S3.Upload(4294838825,0,0,0,0 47433)
Amazon.AWS_Smart.Plug(4294838826,0,0,0,0 10000199) Amazon.AWS_Workspaces.Health.Status(4294838827,0,0,0,0 51269)
  Src address(1):
        192.168.10.0-192.168.10.255
```

5. 查看 SD-WAN 规则对应的策略路由状态，出接口优先选择延迟较低的联通线路（port2）。

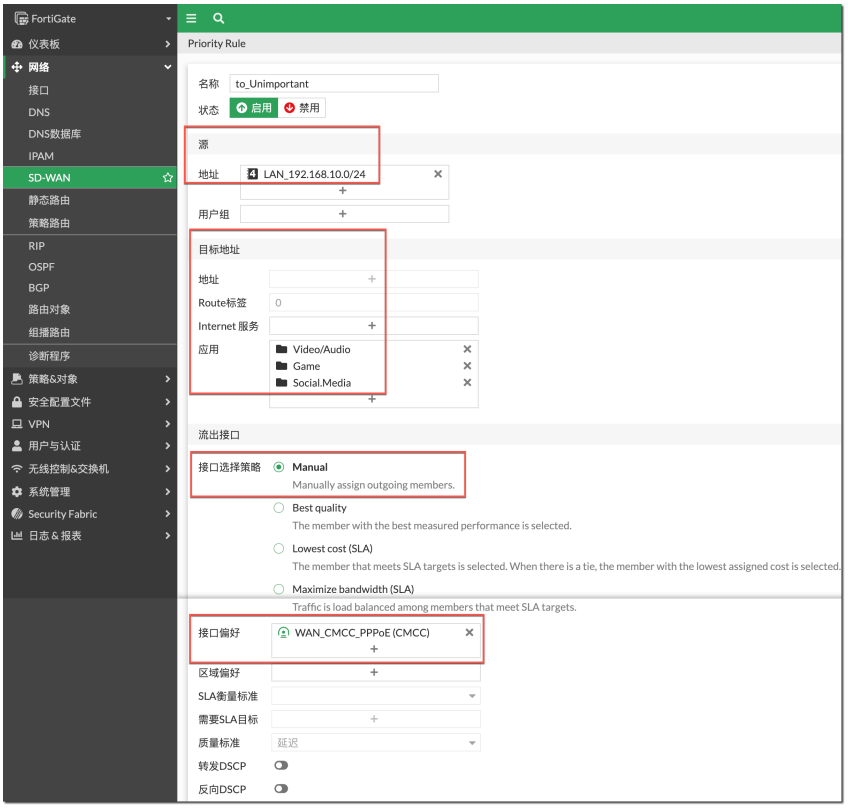
```bash
FortiGate # diagnose firewall proute list
list route policy info(vf=root):

id=2132869121(0×7f210001) vwl_service=1(to_AWS) vwl_mbr_seq=1 2 dscp_tag=0×fc 0×fc flags=0×0 tos=0×00 tos_mask=0×00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(2) oif=4(port2) oif=5(port3)     //选择延迟较低的联通线路
source(1): 192.168.10.0-192.168.10.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(27): Amazon-AWS(393320,0,0,0,0) Amazon-AWS.API.Gateway(393478,0,0,0,0) Amazon-
AWS.AppFlow(393484,0,0,0,0) Amazon-AWS.Chime.Meetings(393483,0,0,0,0) Amazon-AWS.Chime.Voice.Connector(393479,0,0,0,0)
Amazon-AWS.Cloud9(393471,0,0,0,0) Amazon-AWS.CloudFront(393481,0,0,0,0) Amazon-AWS.CodeBuild(393482,0,0,0,0) Amazon-
AWS.Connect(393480,0,0,0,0) Amazon-AWS.DynamoDB(393472,0,0,0,0) Amazon-AWS.EBS(393470,0,0,0,0) Amazon-
AWS.EC2(393477,0,0,0,0) Amazon-AWS.Global.Accelerator(393476,0,0,0,0) Amazon-AWS.GovCloud.US(393452,0,0,0,0) Amazon-
AWS.Kinesis.Video.Streams(393475,0,0,0,0) Amazon-AWS.Route53(393473,0,0,0,0) Amazon-AWS.S3(393474,0,0,0,0) Amazon-
AWS.WorkSpaces.Gateway(393403,0,0,0,0) Amazon.AWS(4294836343,0,0,0,0, 27210) Amazon.AWS.Console(4294836344,0,0,0,0,
37172) Amazon.AWS_EC2(4294836345,0,0,0,0, 36740) Amazon.AWS_IoT(4294838823,0,0,0,0, 10000051)
Amazon.AWS_S3(4294836346,0,0,0,0, 35944) Amazon.AWS_S3.Download(4294838824,0,0,0,0, 47432)
Amazon.AWS_S3.Upload(4294838825,0,0,0,0, 47433) Amazon.AWS_Smart.Plug(4294838826,0,0,0,0, 10000199)
Amazon.AWS_Workspaces.Health.Status(4294838827,0,0,0,0, 51269)
hit_count=0 last_used=2023-12-15 15:59:10
```

需求举例 2

其他不重要的业务从移动线路（PPPoE）接口转发。

1. 新建 SD-WAN 规则 2：

- 源地址选择内网网段；目标地址使用应用分类，选择视频/语音、游戏、社交媒体等对于客户不重要的业务应用分类（SD-WAN 规则中的目标地址可以选择应用分类在 7.2.0 开始支持，之前的版本只支持选择具体的应用特征条目）。
- 接口选择策略：选择 Manual 模式。
- 接口偏好：选用 SD-WAN 规则接口成员中的移动 PPPoE 线路。




```bash
config system sdwan
    config service
        edit 2
            set name "to_Unimportant"
            set src "LAN_192.168.10.0/24"
            set internet-service enable
            set internet-service-app-ctrl-category 5 8 23
            set priority-members 3
        next
    end
end
```

2. 查看此 SD-WAN 规则状态。

```bash
FortiGate # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(1):
    1: Seq_num(3 CMCC), alive, selected
  Internet Service(3): Video/Audio(0,5,0,0,0) Game(0,8,0,0,0) Social.Media(0,23,0,0,0)
  Src address(1):
        192.168.10.0-192.168.10.255
```

需求举例 3

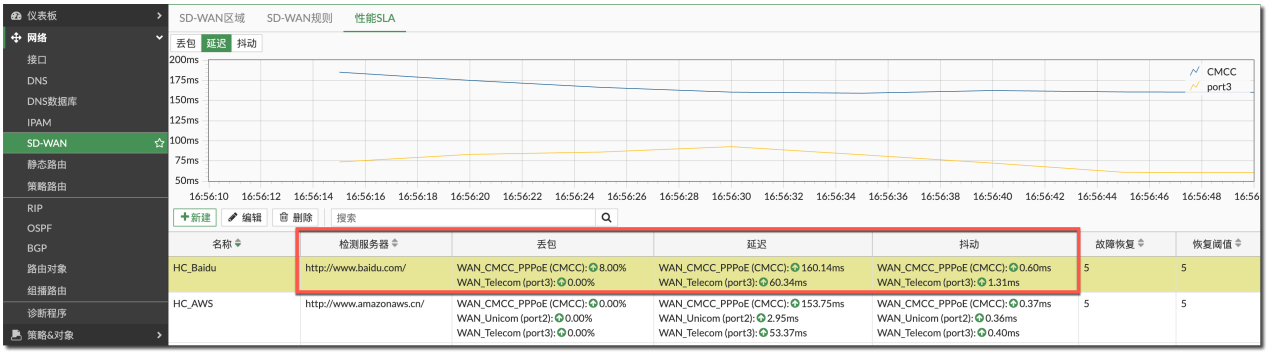
其他类型的流量从电信线路（port3）或移动线路（PPPoE）中延迟较低的线路转发。

1. 新建 SD-WAN 健康检查，协议为 HTTP，检查目标为网站 www.baidu.com ↗ ，参与接口选择电信线路（port3）和移动线路（PPPoE）。




```
config system sdwan
    config health-check
        edit "HC_Baidu"
            set server "www.baidu.com"
            set protocol http
            set members 3 2
        next
    end
end
```

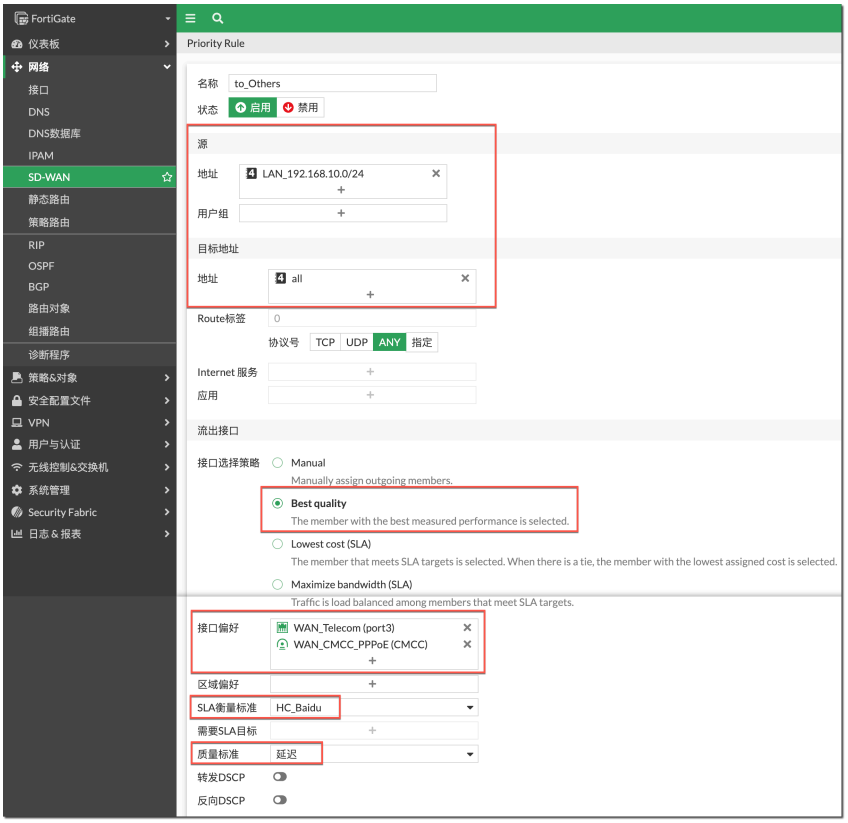
2. 创建完成后，可以看到电信线路（port3）比移动线路（PPPoE）的延迟要低（超过了 10% 的阈值）。




```bash
FortiGate # diagnose sys sdwan health-check status HC_Baidu
Health Check(HC_Baidu):
Seq(3 CMCC): state(alive), packet-loss(1.000%) latency(165.551), jitter(1.748), mos(4.267), bandwidth-up(65534991),
bandwidth-dw(65534974), bandwidth-bi(131069965) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(64.622), jitter(1.523), mos(4.369), bandwidth-up(65534989),
bandwidth-dw(65534969), bandwidth-bi(131069958) sla_map=0x0
```

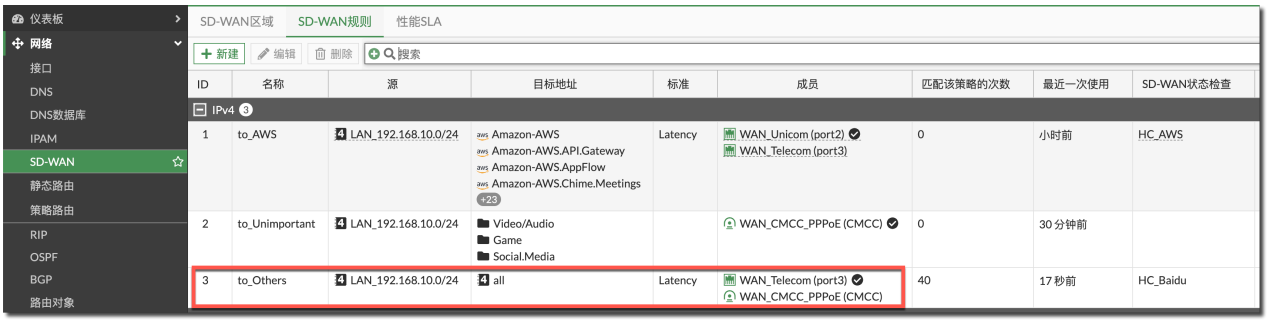
3. 新建 SD-WAN 规则 3：

- 源地址选择内网网段；目标地址选择 all。
- 接口选择策略和质量标准：选择 Best Quality 和延迟，选用 SD-WAN 规则接口成员中延迟最低的接口，如果成员接口的延迟差小于 10%（默认值），则使用接口偏好中配置较前的接口。
- SLA 衡量标准：选择上步创建的健康检查。
- 接口偏好：选择电信线路接口（port3）与移动线路接口（PPPoE），电信线路接口（port3）排序在前。




```bash
config system sdwan
    config service
        edit 3
            set name "to_Others"
            set mode priority
            set dst "all"
            set src "LAN_192.168.10.0/24"
            set health-check "HC_Baidu"
            set priority-members 2 3
        next
    end
end
```

4. 查看上步创建的 SD-WAN 规则目前的选路状态，由于电信线路（port2）的延迟较低，移动线路（PPPoE）的延迟较高，访问其他的业务优先选择电信线路（port3）。

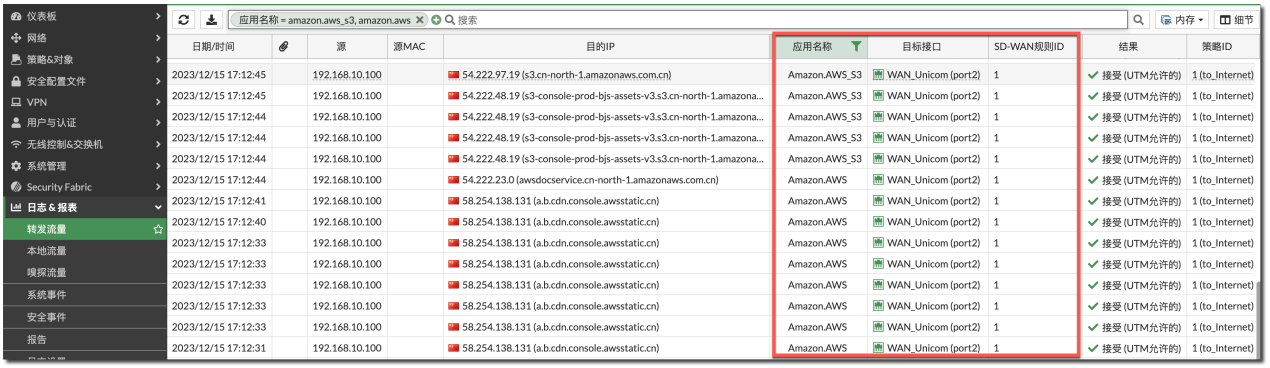



```bash
FortiGate # diagnose sys sdwan service 3

Service(3): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(HC_Baidu)
  Members(2):
    1: Seq_num(2 port3), alive, latency: 66.413, selected    //优先选择电信线路
    2: Seq_num(3 CMCC), alive, latency: 167.769, selected
```

```
Src address(1):
      192.168.10.0-192.168.10.255

Dst address(1):
      0.0.0.0-255.255.255.255
```

结果验证

1. 使用内网 PC 客户端访问 AWS 业务，在转发流量日志中查看相关流量，可以看到出接口从联通线路（port2）转发。



2. 查看 SD-WAN 规则学习应用控制特征的列表。

```
FortiGate # diagnose sys sdwan internet-service-app-ctrl-list                    bash

Amazon.AWS(27210 4294836343): 54.222.23.0 6 443 Fri Dec 15 17:12:37 2023
Amazon.AWS(27210 4294836343): 58.254.138.131 6 443 Fri Dec 15 17:12:28 2023
Amazon.AWS_S3(35944 4294836346): 52.82.189.181 6 443 Fri Dec 15 17:14:35 2023
Amazon.AWS_S3(35944 4294836346): 54.222.48.19 6 443 Fri Dec 15 17:12:33 2023
Amazon.AWS_S3(35944 4294836346): 54.222.52.141 6 443 Fri Dec 15 17:10:13 2023
Amazon.AWS_S3(35944 4294836346): 54.222.97.19 6 443 Fri Dec 15 17:11:45 2023

通过如下命令可以清除此列表：
diagnose sys sdwan internet-service-app-ctrl-flush
```

重要

关于 SD-WAN 调用应用特征，需要以下学习过程：

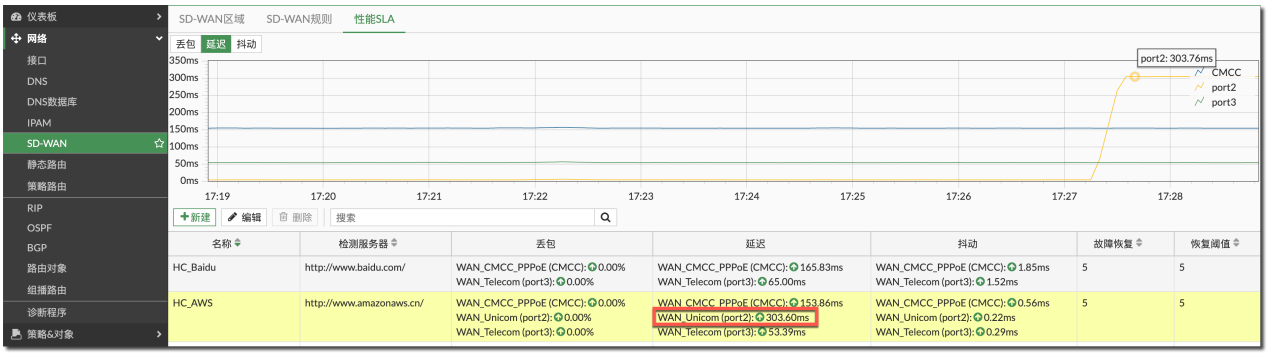
第 1 步：应用特征是需要先通过内网用户使用 AWS，然后 FortiGate 进行应用识别的，因此策略里面一定要开启 APP Control 才可以。

第 2 步：识别之后将会学习到相应的应用特征的 IP 地址，然后形成一个动态的应用程序 IP 地址数据库（如上所示），最终才会将这个学习到的 IP 数据库加入到 SD-WAN 规则（策略路由）里面。

因此这需要有一个学习的过程，因此在没有成功学习之前，AWS 的流量会走到其他线路上去（匹配 SD-WAN 规则 3），这并非问题，而是 SD-WAN 调用应用的工作逻辑就是这样，学习完毕之后，新发起的 AWS 流量才会匹配到 SD-WAN 规则 1。

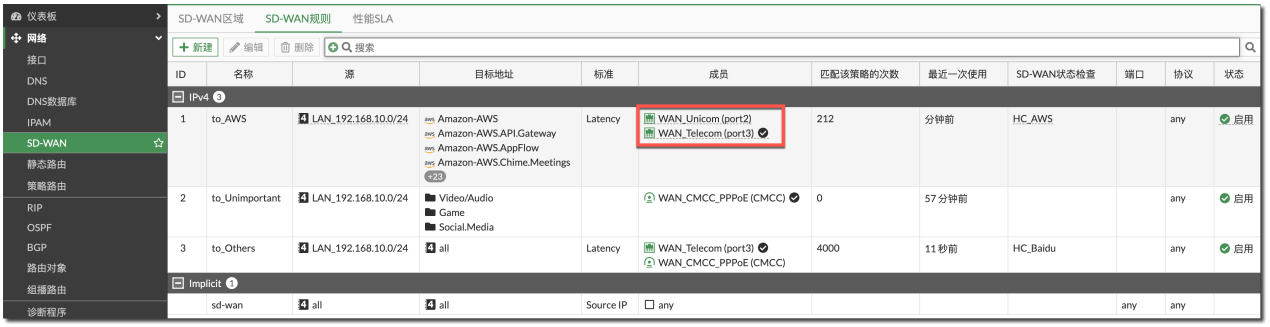
如果直接匹配上了 ISDB 条目，则会直接匹配 SD-WAN 规则，不会有以上需要学习的过程。

3. 模拟联通线路（port2）到 AWS 的延迟增加到 300ms。



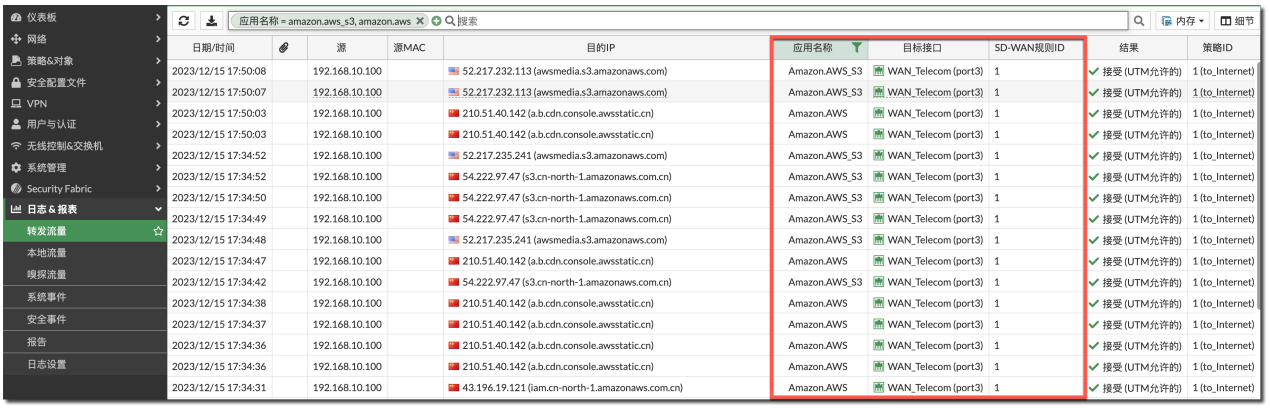
```bash
FortiGate # diagnose sys sdwan health-check status HC_AWS
Health Check(HC_AWS):
Seq(1 port2): state(alive), packet-loss(0.000%) latency(303.826), jitter(0.410), mos(3.767), bandwidth-up(65534995),
bandwidth-dw(65534993), bandwidth-bi(131069988) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(0.000%) latency(53.633), jitter(0.326), mos(4.376), bandwidth-up(65534991),
bandwidth-dw(65534973), bandwidth-bi(131069964) sla_map=0x0
Seq(3 CMCC): state(alive), packet-loss(0.000%) latency(153.905), jitter(0.378), mos(4.307), bandwidth-up(65534992),
bandwidth-dw(65534975), bandwidth-bi(131069967) sla_map=0x0
```

4. 查看 SD-WAN 规则 1 的选路优先选择电信线路（port3）去往 AWS 业务。

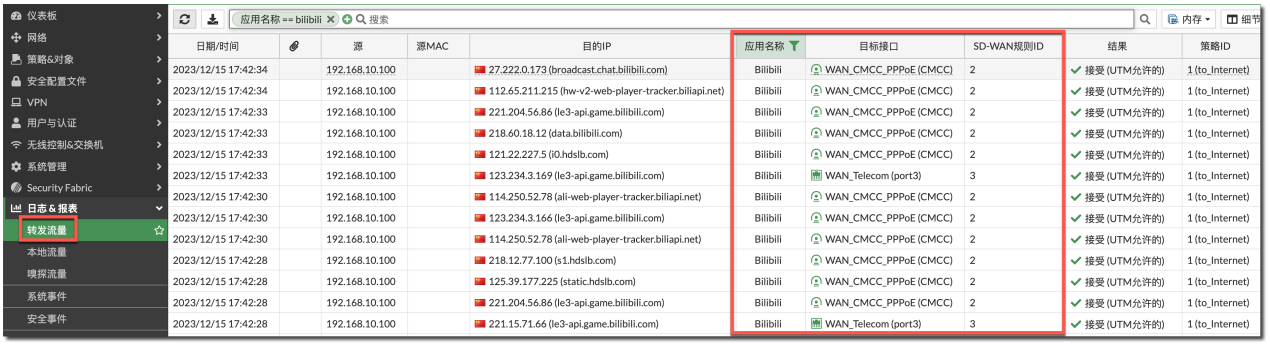



```bash
FortiGate # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10),
heath-check(HC_AWS)
  Members(2):
    1: Seq_num(2 port3), alive, latency: 53.560, selected
    //优先选择延迟较低的电信线路//
    2: Seq_num(1 port2), alive, latency: 304.084, selected
  Internet Service(27): Amazon-AWS(393320,0,0,0,0) Amazon-AWS.API.Gateway(393478,0,0,0,0) Amazon-
AWS.AppFlow(393484,0,0,0,0) Amazon-AWS.Chime.Meetings(393483,0,0,0,0) Amazon-AWS.Chime.Voice.Connector(393479,0,0,0,0)
Amazon-AWS.Cloud9(393471,0,0,0,0) Amazon-AWS.CloudFront(393481,0,0,0,0) Amazon-AWS.CodeBuild(393482,0,0,0,0) Amazon-
AWS.Connect(393480,0,0,0,0) Amazon-AWS.DynamoDB(393472,0,0,0,0) Amazon-AWS.EBS(393470,0,0,0,0) Amazon-
AWS.EC2(393477,0,0,0,0) Amazon-AWS.Global.Accelerator(393476,0,0,0,0) Amazon-AWS.GovCloud.US(393452,0,0,0,0) Amazon-
AWS.Kinesis.Video.Streams(393475,0,0,0,0) Amazon-AWS.Route53(393473,0,0,0,0) Amazon-AWS.S3(393474,0,0,0,0) Amazon-
AWS.WorkSpaces.Gateway(393403,0,0,0,0) Amazon.AWS(4294836343,0,0,0,0 27210) Amazon.AWS.Console(4294836344,0,0,0,0 37172)
Amazon.AWS_EC2(4294836345,0,0,0,0 36740) Amazon.AWS_IoT(4294838823,0,0,0,0 10000051) Amazon.AWS_S3(4294836346,0,0,0,0
35944) Amazon.AWS_S3.Download(4294838824,0,0,0,0 47432) Amazon.AWS_S3.Upload(4294838825,0,0,0,0 47433)
Amazon.AWS_Smart.Plug(4294838826,0,0,0,0 10000199) Amazon.AWS_Workspaces.Health.Status(4294838827,0,0,0,0 51269)
  Src address(1):
        192.168.10.0-192.168.10.255
```

5. 再次使用内网 PC 客户端访问 AWS 业务，在转发流量日志中查看相关流量，可以看到出接口从电信线路（port3）转发。



6. 使用内网 PC 客户端访问视频网站业务，在转发流量日志中查看相关流量，可以看到出接口从移动线路（PPPoE）转发。



7. 查看 SD-WAN 规则学习应用控制特征分类的列表（注意和规则 1 是不一样的命令，这里看的是应用分类，而不是应用）。

```bash
FortiGate # diagnose sys sdwan internet-service-app-ctrl-category-list

Bilibili(53243 4294839067): 27.222.0.173 6 7826 Fri Dec 15 17:41:47 2023
Bilibili(53243 4294839067): 112.65.211.215 6 443 Fri Dec 15 17:41:48 2023
Bilibili(53243 4294839067): 112.83.140.17 6 443 Fri Dec 15 17:39:16 2023
Bilibili(53243 4294839067): 114.250.52.78 6 443 Fri Dec 15 17:41:48 2023
Bilibili(53243 4294839067): 116.130.196.134 6 443 Fri Dec 15 17:41:43 2023
Bilibili(53243 4294839067): 121.22.227.2 6 443 Fri Dec 15 17:39:14 2023
Bilibili(53243 4294839067): 121.22.227.5 6 443 Fri Dec 15 17:41:47 2023
Bilibili(53243 4294839067): 123.234.3.166 6 443 Fri Dec 15 17:41:43 2023
Bilibili(53243 4294839067): 123.234.3.169 6 443 Fri Dec 15 17:42:28 2023
Bilibili(53243 4294839067): 125.39.177.219 6 443 Fri Dec 15 17:39:13 2023
......

通过如下命令可以清除此列表：
diagnose sys sdwan internet-service-app-ctrl-flush
```

重要

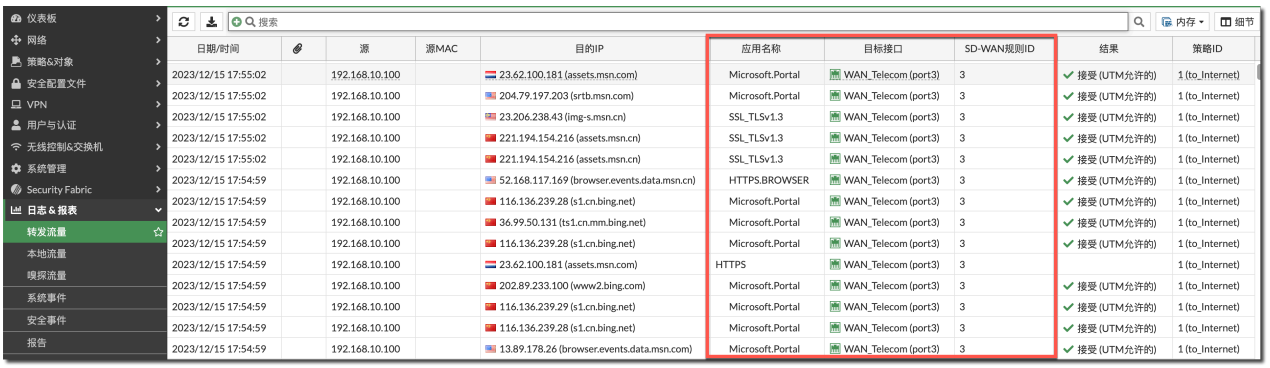
关于 SD-WAN 调用应用特征，大致需要满足以下要求：

第 1 步：应用特征是需要内网用户先访问视频网站，然后 FortiGate 进行应用识别的，因此策略里面一定要开启 APP Control 才可以。

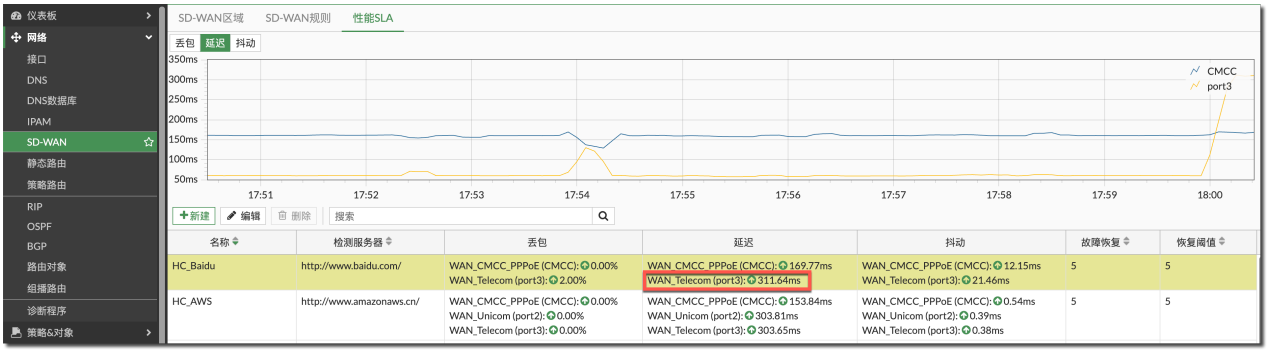
第 2 步：识别之后将会学习到相应的应用特征的 IP 地址，然后形成一个动态的应用程序 IP 地址数据库（如上所示），最终才会将这个学习到的 IP 数据库加入到 SD-WAN 规则（策略路由）里面。

因此这需要有一个学习的过程，因此在没有成功学习之前，AWS 的流量会走到其他线路上去（匹配 SD-WAN 规则 3），这并非问题，而是 SD-WAN 调用应用的工作逻辑就是这样，学习完毕之后，新发起的 AWS 流量才会匹配到 SD-WAN 规则 2。

8. 使用内网 PC 客户端访问其他业务网站，如搜索引擎，在转发流量日志中查看相关流量，可以看到出接口从电信线路（port3）转发。

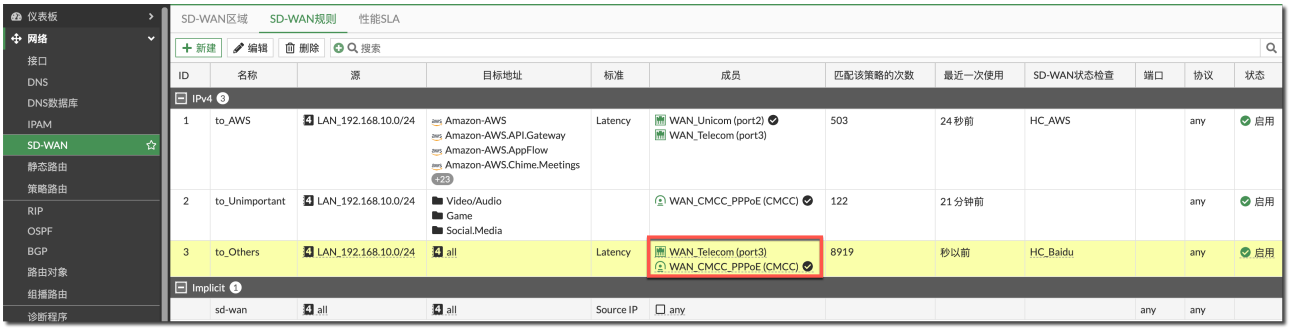


9. 模拟电信线路（port3）到 Baidu 的延迟增加到 300ms。



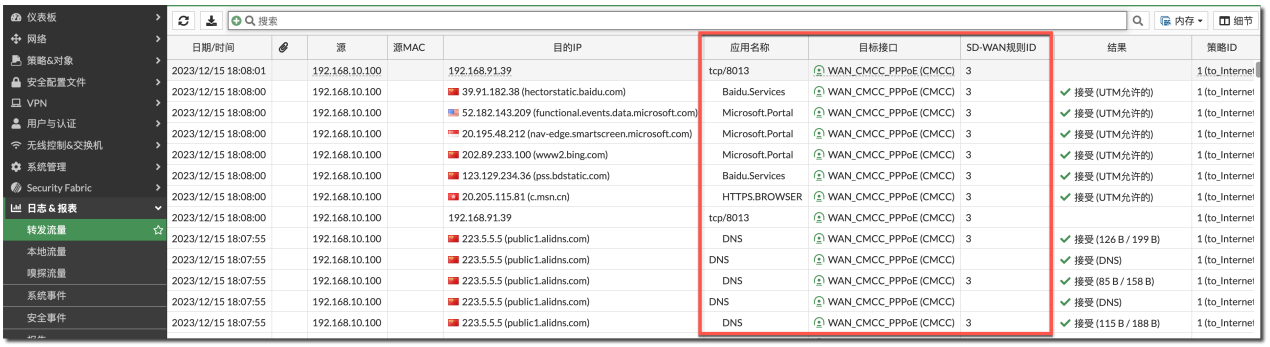

```bash
FortiGate # diagnose sys sdwan health-check status HC_Baidu
Health Check(HC_Baidu):
Seq(3 CMCC): state(alive), packet-loss(0.000%) latency(165.041), jitter(0.734), mos(4.274), bandwidth-up(65534990),
bandwidth-dw(65534969), bandwidth-bi(131069959) sla_map=0x0
Seq(2 port3): state(alive), packet-loss(2.000%) latency(304.182), jitter(21.180), mos(3.573), bandwidth-up(65534991),
bandwidth-dw(65534973), bandwidth-bi(131069964) sla_map=0x0
```

.0. 查看 SD-WAN 规则 3 的选路优先选择移动线路（PPPoE）去往其他业务。




```bash
FortiGate # diagnose sys sdwan service 3

Service(3): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-cost-threshold(10), heath-
check(HC_Baidu)
  Members(2):
    1: Seq_num(3 CMCC), alive, latency: 160.563, selected
    //优先从移动线路转发//
    2: Seq_num(2 port3), alive, latency: 310.831, selected
  Src address(1):
        192.168.10.0-192.168.10.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

.1. 使用内网 PC 客户端访问其他业务网站，在转发流量日志中查看相关流量，可以看到出接口从移动线路（PPPoE）转发。



注意事项

1. 如果需要学习将到的应用数据库立马完全生效，旧的会话是会保持原有出口的，想要这些旧的会话立即生效需要清除旧的会话和路由缓存信息，而新建的会话则不需要此操作，具体命令如下。

```bash
diagnose sys session filter src 192.168.10.100
diagnose sys session clear    //清除测试机器的会话，让其重新匹配新的SD-WAN规则（有skype的IP数据库更新）
```

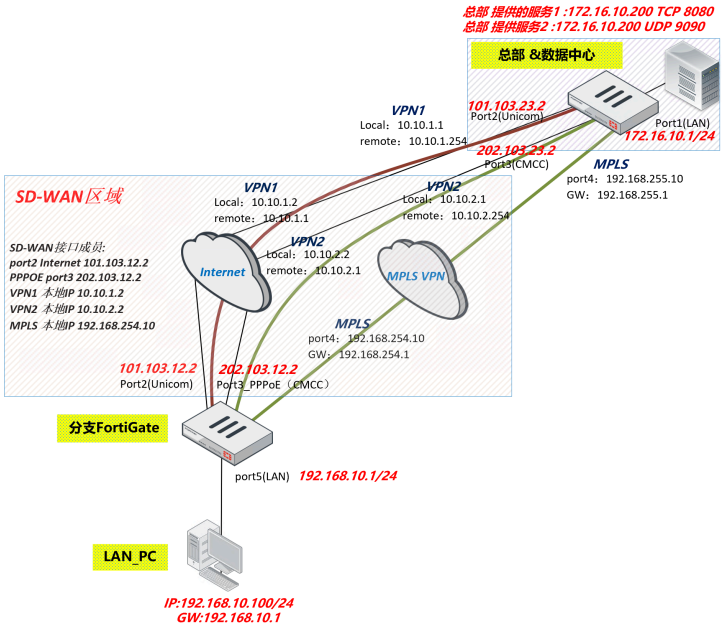
2. 在 SNAT 环境的默认配置下，当 SD-WAN 选路出现变化后，已有的会话不会立即切换到新链路上，只有线路切换后新创建的会话会建立在新的线路上，如果想在 SD-WAN 线路切换后让已有会话立刻失效，需要使用如下命令（可能会增加 CPU 消耗，谨慎开启）。

> **重要**
>
> 默认配置下的 SNAT 场景，即使转发流量的 SD-WAN 成员的健康检查变为 dead，只要该线路仍然存在路由（可能关闭了 `update-static-route`），已有的旧会话仍然不会切换到其他线路。

```bash
config system global
    set snat-route-change enable    //默认为disable//
end
```

互联网出口和 VPN 混合场景

网络拓扑与需求



1. Internet 访问：
   - 分支 FortiGate 通过 2 个运营商（联通 port2、移动 PPPoE）接入 Internet，其中联通为固定 IP，移动线路为 PPPoE 线路。联通和移动线路借用出接口 IP 作为 SNAT 的源地址访问 Internet。
   - 总部 FortiGate 通过 2 个运营商（联通 port2、移动 port3，均为固定 IP）接入 Internet。
2. 分支 FortiGate 与总部 FortiGate 间的流量：
   - 通过 MPLS 专线互联（port4）。
   - 通过两条 Internet 线路上的 IPSec 隧道互联（作为 MPLS 专线的备份链路）。
3. 分支 FortiGate 配置 SD-WAN，划分 2 个 SD-WAN 区域：
   - 一个区域用于访问 Internet，成员为联通 port2、移动 PPPoE 接口。
   - 另一个区域用于分支 FortiGate 与总部 FortiGate 间的流量，成员为两条 Internet 线路上的 IPSec 接口和 MPLS 专线接口 port4。
4. 分支 FortiGate 的流量控制：
   - 访问 Internet 的流量优先从联通线路（port2）转发，联通线路的质量不符合标准时，通过移动线路（PPPoE）转发。
   - 访问总部 FortiGate 内网的流量优先通过 MPLS 线路（port4）转发，MPLS 线路质量不佳或中断时，优先通过联通线路上的 IPSec 隧道转发，其次通过移动线路上的 IPSec 隧道转发。
5. 部署 SD-WAN 通常在一个全新的设备进行配置，如果需要加入 SD-WAN 区域的接口已经被其他配置（如防火墙策略）引用，是无法直接加入 SD-WAN 区域的。要加入 SD-WAN 区域，又不想删除关联的配置，请参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域 ↗章节。

配置步骤

基础配置

Spoke

配置分支 FortiGate 2 条运营商线路的 IP、PPPoE 用户（PPPoE 请通过 CLI 配置，参考部署前注意事项 → PPPoE 接口配置 ↗章节）、MPLS 线路 IP 与内网口 IP。




```bash
PPPoE接口配置:
config system pppoe-interface
    edit "WAN_CMCC_PPPoE"
        set device "port3"
        set username "user1"
        set password xxxxxx
    next
end

固定IP接口配置:
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.12.2 255.255.255.0
        set allowaccess ssh
        set alias "WAN_Unicom"
    next
    edit "port4"
        set vdom "root"
        set ip 192.168.254.10 255.255.255.0
        set allowaccess ssh
        set alias "MPLS"
    next
    edit "port5"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh http
        set alias "LAN"
    next
end
```

Hub

配置总部 FortiGate 2 条运营商线路的 IP、MPLS 线路 IP 与内网口 IP。




```bash
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.23.2 255.255.255.0
        set allowaccess ssh
        set alias "WAN_Unicom"
    next
    edit "port3"
        set vdom "root"
        set ip 202.103.23.2 255.255.255.0
        set allowaccess ssh
        set alias "WAN_CMCC"
    next
    edit "port4"
        set vdom "root"
        set ip 192.168.255.10 255.255.255.0
        set allowaccess ssh
        set alias "MPLS"
    next
    edit "port5"
        set vdom "root"
        set ip 172.16.10.1 255.255.255.0
        set allowaccess ping https ssh http
        set alias "LAN"
    next
end
```

IPSec 配置

Spoke

1. 在分支（Spoke）FortiGate 上进入 VPN → IPSec 隧道页面，新建联通线路（port2）上的 IPSec 隧道，选择自定义模式，按照下图所示配置，算法默认即可。




Spoke
仪表板
网络
策略&对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec隧道
IPsec向导
IPsec 隧道模板
VPN创建向导
1 VPN建立
名称
to_Hub_Unicom
模板类型
站到站
Hub-and-Spoke
远程拨号
自定义
< 返回
下一步 >
取消


新建VPN隧道
名称
to_Hub_Unicom
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
101.103.23.2
接口
WAN_Unicom (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
阶段1提案
添加
加密
AES128
认证
SHA256
AES256
AES128GCM
PRF
PRFSHA256
AES256GCM
PRFSHA384
Diffie-Hellman组
密钥生存时间(秒)
86400
本地ID
阶段2选择器
名称
本地地址
对端地址
to_Hub_Unicom
192.168.10.0/24
0.0.0.0/0.0.0.0
新建阶段 2
本地地址
子网
192.168.10.0/24
对端地址
阶段2提案
SHA1
CHACHA20POLY1305
启用重放检测
启用完全前向保密 (PFS)
本地端口
对端端口
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
43200
填写Hub端联通线路地址
接口选择本端联通线路接口
建议DPD模式配置为空闲
更快地探测到线路中断
填写预共享密钥
IKE版本推荐IKEv2
如果没有在IPSec上部署动态路由
感兴趣流的源地址建议填写明细网段
Hub端自动添加去往Spoke内网路由

```bash
config vpn ipsec phase1-interface
    edit "to_Hub_Unicom"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set remote-gw 101.103.23.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_Hub_Unicom"
        set phase1name "to_Hub_Unicom"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.10.0 255.255.255.0
    next
end
```

2. 新建联通线路（port2）上的 IPSec 隧道，选择自定义模式，按照下图所示配置，算法默认即可。




```bash
config vpn ipsec phase1-interface
    edit "to_Hub_CMCC"
        set interface "WAN_CMCC_PPPoE"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set remote-gw 202.103.23.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
```

```
    edit "to_Hub_CMCC"
        set phase1name "to_Hub_CMCC"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.10.0 255.255.255.0
    next
end
```

Hub

1. 在总部（Hub）FortiGate 上进入 VPN → IPSec 隧道页面，新建联通线路（port2）上的 IPSec 隧道（拨号模式），选择自定义模式，按照下图所示配置，算法默认即可。确保 add-route 功能开启（默认开启）。

> **重要**
>
> 如果 Hub 端不使用 SD-WAN 功能，请配置联通线路 IPSec add-route 添加路由的优先级为 100。优先于移动线路 IPSec 的优先级 200，当 Hub 端的 MPLS 专线中断时，优先使用联通线路主动访问 Spoke。

> **重要**
>
> 由于 Spoke 两条线路的 IPSec 配置的源保护网段一致，Hub 端的两个拨号连接在 add-route 时会产生冲突，导致 Spoke 两条线路的两条隧道发生震荡，所以在 GUI 配置下发完成后，需要编辑二阶段接口配置，添加“set route-overlap allow”配置，如下 CLI 所示。


Hub
仪表板
网络
策略&对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec隧道
IPsec向导
IPsec 隧道模板
SSL-VPN门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制&交换机
系统管理
Security Fabric
日志 & 报表
新建VPN隧道
名称 Hub_Unicom
注释 0/255
网络
IP版本 IPv4 IPv6
对端网关 拨号用户
接口 WAN_Unicom (port2)
对端网关选择拨号用户
接受来自所有Spoke的协商
接口选择本端联通线路接口
本地网关
模式配置
NAT穿越 启用 禁用 强制的
对等体状态探测 禁用 空闲 按需
DPD重试计数 3
DPD重试间隔 60 s
前向纠错 Egress Ingress
高级
添加路由 启用 禁用
开启“添加路由”（默认开启）
根据Spoke的明细源保护网段
自动添加去往Spoke内网的路由
自动发现发送者 启用 禁用
自动发现接收者 启用 禁用
Exchange接口IP 启用 禁用
设备创建 启用 禁用
隧道聚合成员 启用 禁用
认证
方法 预共享密钥
预共享密钥 ••••••
填写预共享密钥
IKE
版本 1 2
IKE版本推荐IKEv2
对等体选项
访问类型 任意对等体ID
访问时选择任意对等体ID
阶段1提案 添加
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM PRF PRFSHA256
加密 AES256GCM PRF PRFSHA384
加密 CHACHA20P PRF PRFSHA256
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
密钥生存时间(秒) 86400
本地ID
阶段2选择器
名称 本地地址 对端地址
Hub_Unicom 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
新建阶段 2
名称 Hub_Unicom
注释
本地地址 子网 0.0.0.0/0.0.0.0
对端地址 子网 0.0.0.0/0.0.0.0
二阶段感兴趣流保持全0
高级
阶段2提案 添加
加密 AES128 认证 SHA1
加密 AES256 认证 SHA1
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM
加密 AES256GCM
加密 CHACHA20POLY1305
启用重放检测
启用完全前向保密 (PFS)
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
本地端口 全部
对端端口 全部
协议 全部
自动密钥保持存活
密钥周期(秒/kb) 秒
秒 43200

```bash
config vpn ipsec phase1-interface
    edit "Hub_Unicom"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set add-route enable    //确认add-route为enable，如果是disable，手动enable它//
        set dpd on-idle
        set psksecret xxxxxx
        set priority 100    //如果Hub端不配置SD-WAN，配置此项赋值add-route添加路由的优先级//
        set dpd-retryinterval 60
    next
end

config vpn ipsec phase2-interface
    edit "Hub_Unicom"
        set phase1name "Hub_Unicom"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set keepalive enable
        set route-overlap allow    //重要：注意在CLI下添加此配置//
    next
end
```

2. 在总部（Hub）FortiGate 上新建移动线路（port3）上的 IPSec 隧道（拨号模式），选择自定义模式，按照下图所示配置，算法默认即可。确保 add-route 功能开启（默认开启）。

> **重要**
>
> 如果 Hub 端不使用 SD-WAN 功能，请配置移动线路 IPSec add-route 添加路由的优先级为 200。次于移动线路 IPSec 的优先级 100，当 Hub 端的 MPLS 专线中断时，优先使用联通线路的 IPSec 主动访问 Spoke 内网，其次使用移动线路的 IPSec 主动访问 Spoke 内网。

> **重要**
>
> 由于 Spoke 两条线路的 IPSec 配置的源保护网段一致，Hub 端的两个拨号连接在 add-route 时会产生冲突，导致 Spoke 两条线路的两条隧道发生震荡，所以在 GUI 配置下发完成后，需要编辑二阶段接口配置，添加“set route-overlap allow”配置，如下 CLI 所示。


Hub
仪表板
网络
策略&对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec隧道
IPsec向导
IPsec 隧道模板
SSL-VPN门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制&交换机
系统管理
Security Fabric
日志 & 报表
新建VPN隧道
名称 Hub_CMCC
注释 注释 0/255
网络
IP版本 IPv4 IPv6
对端网关 拨号用户
接口 WAN_CMCC (port3)
对端网关选择拨号用户
接受来自所有Spoke的协商
接口选择本端移动线路接口
本地网关
模式配置
NAT穿越 启用 禁用 强制的
对等体状态探测 禁用 空闲 按需
DPD重试计数 3
DPD重试间隔 60 s
前向纠错 Egress Ingress
高级
添加路由 启用 禁用
自动发现发送者 启用 禁用
自动发现接收者 启用 禁用
Exchange接口IP 启用 禁用
设备创建 启用 禁用
隧道聚合成员 启用 禁用
开启“添加路由”（默认开启）
根据Spoke的明细源保护网段
自动添加去往Spoke内网的路由
认证
方法 预共享密钥
预共享密钥 ••••••
IKE
版本 1 2
对等体选项
访问类型 任意对等体ID
填写预共享密钥
IKE版本推荐IKEv2
访问类型选择任意对等体ID
阶段1提案 添加
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM PRF PRFSHA256
加密 AES256GCM PRF PRFSHA384
加密 CHACHA20P PRF PRFSHA256
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
密钥生存时间(秒) 86400
本地ID
阶段2选择器
名称 本地地址 对端地址
Hub_CMCC 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
新建阶段 2
名称 Hub_CMCC
注释 注释
本地地址 子网 0.0.0.0/0.0.0.0
对端地址 子网 0.0.0.0/0.0.0.0
二阶段感兴趣流保持全0
高级
阶段2提案 添加
加密 AES128 认证 SHA1
加密 AES256 认证 SHA1
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM
加密 AES256GCM
加密 CHACHA20POLY1305
启用重放检测
启用完全前向保密 (PFS)
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
本地端口 全部
对端端口 全部
协议 全部
自动密钥保持存活
密钥周期(秒/kb) 秒
秒 43200

```bash
config vpn ipsec phase1-interface
    edit "Hub_CMCC"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set add-route enable    //确认add-route为enable，如果是disable，手动enable它//
        set dpd on-idle
        set psksecret xxxxxx
        set priority 200    //如果Hub端不配置SD-WAN，配置此项赋值add-route添加路由的优先级//
        set dpd-retryinterval 60
    next
end

config vpn ipsec phase2-interface
    edit "Hub_CMCC"
        set phase1name "Hub_CMCC"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set keepalive enable
        set route-overlap allow    //重要：注意在CLI下添加此配置//
    next
end
```

SD-WAN 基础配置

Spoke

1. 进入分支 FortiGate 的网络 → SD-WAN → SD-WAN 区域页面，新建用于访问 Internet 的区域“to_Internet”（也可以直接使用预置的 SD-WAN 区域）和用于访问总部 FortiGate 内网的区域“to_Hub”。








```bash
config system sdwan
    set status enable
    config zone
        edit "to_Internet"
        next
```

```
        edit "to_Hub"
        next
    end
end
```

2. 在此页面新建用于访问 Internet 的 SD-WAN 成员（联通、移动线路 2 个接口）加入上步创建的 SD-WAN 区域 "to_Internet"，注意 PPPoE 加入 SD-WAN 区域时不需要填写网关。

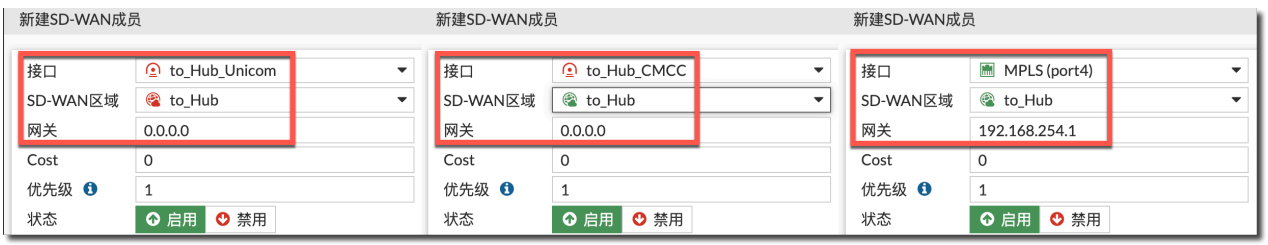
> **重要**
>
> 如果 WAN 接口存在相关配置引用的情况下，是无法直接加入到 SD-WAN 接口成员的，可参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域 ↗章节快速将接口加入 SD-WAN 区域，并自动修改关联配置（也可选择手动将 WAN 接口的相关配置删除引用，包括策略、Link-monitor、所属区域等）。



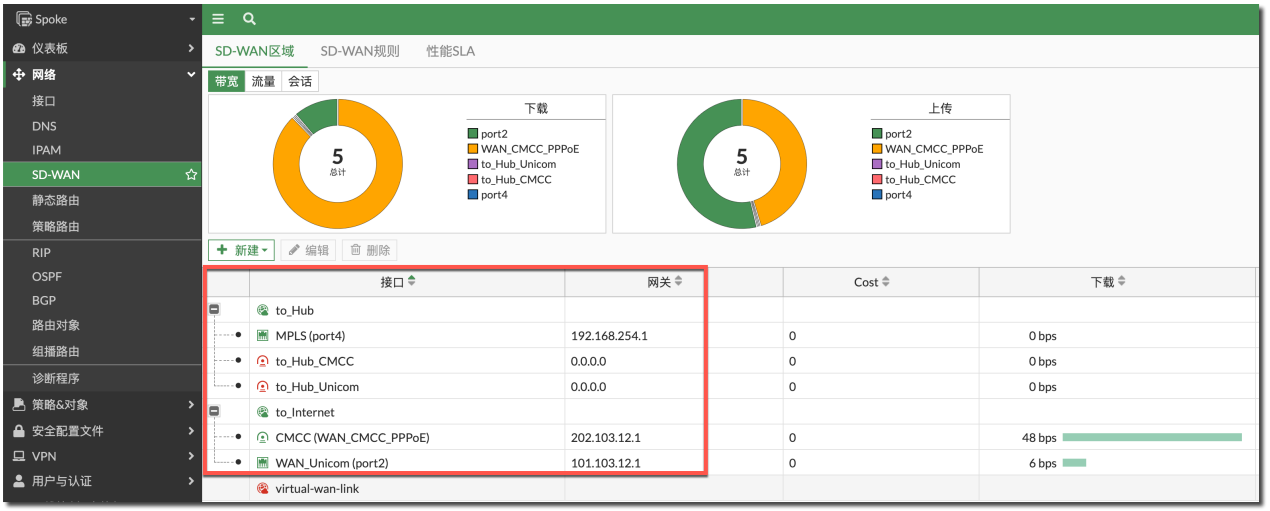



```bash
config system sdwan
    config members
        edit 1
            set interface "port2"
            set zone "to_Internet"
            set gateway 101.103.12.1
        next
        edit 2
            set interface "WAN_CMCC_PPPoE"
            set zone "to_Internet"
        next
    end
end
```

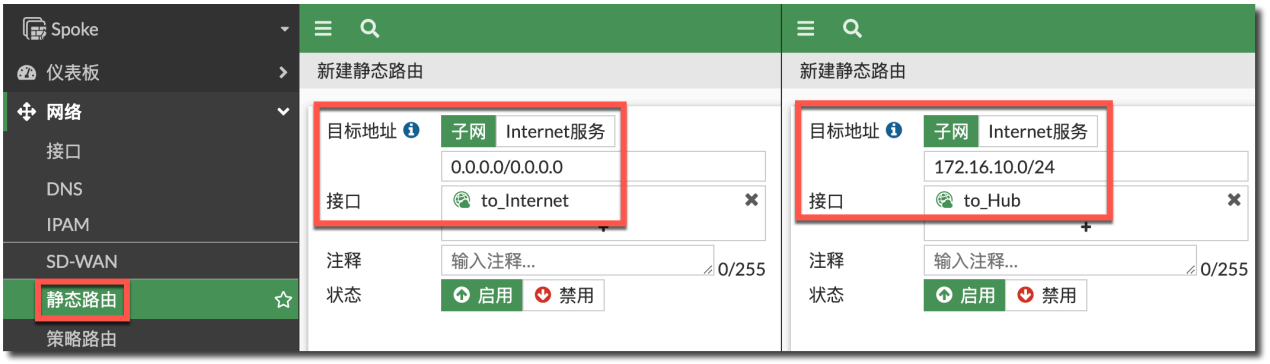
3. 新建用于访问 Hub 内网的 SD-WAN 成员（两个 IPSec Tunnel 接口和 MPLS 线路接口）加入步骤 1 创建的 SD-WAN 区域 "to_Hub"。注意 IPSec Tunnel 接口加入 SD-WAN 区域时不需要填写网关。



```bash
config system sdwan
    config members
        edit 3
            set interface "to_Hub_Unicom"
            set zone "to_Hub"
        next
        edit 4
            set interface "to_Hub_CMCC"
            set zone "to_Hub"
        next
        edit 5
            set interface "port4"
            set zone "to_Hub"
            set gateway 192.168.254.1
        next
    end
end
```

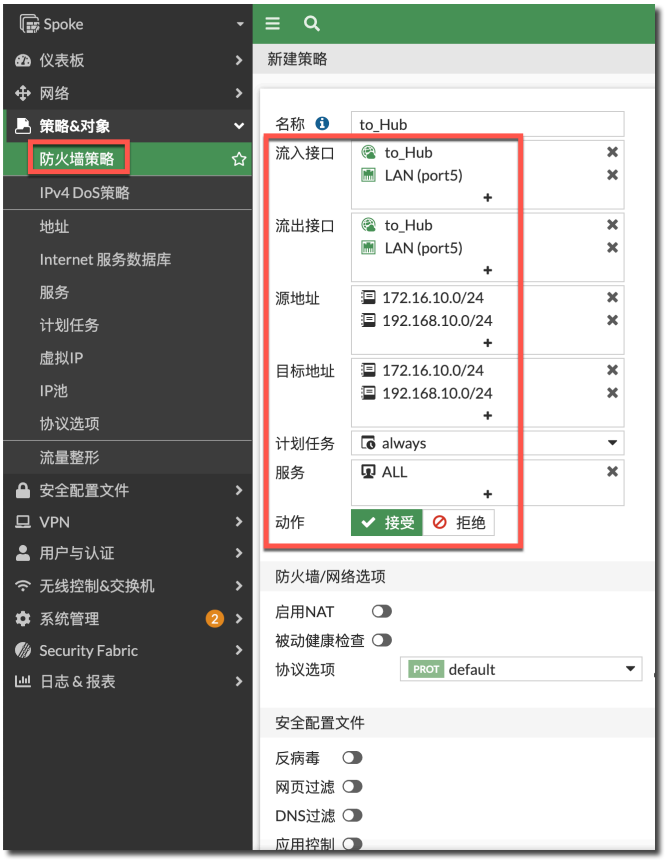


4. 配置默认路由指向 SD-WAN 区域“to_Internet”，配置去往 Hub 端内网网段的路由指向 SD-WAN 区域“to_Hub”。




```bash
config router static
    edit 4
        set distance 1
        set sdwan-zone "to_Internet"
    next
    edit 5
        set dst 172.16.10.0 255.255.255.0
        set distance 1
        set sdwan-zone "to_Hub"
    next
end
```

5. 配置防火墙策略 1，放通内网 port5（LAN）与 Hub 内网之间互相访问的流量（配置前在“系统管理 → 可见功能”中开启“多接口策略”），通常情况下内网之间的访问不需要开启 SNAT。根据需求开启 UTM 功能。




```bash
config firewall policy
    edit
        set name "to_Hub"
        set srcintf "to_Hub" "port5"
        set dstintf "to_Hub" "port5"
        set action accept
        set srcaddr "172.16.10.0/24" "192.168.10.0/24"
        set dstaddr "172.16.10.0/24" "192.168.10.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置防火墙策略 2，放通内网 port5（LAN）到 2 个运营商线路（SD-WAN 区域“to_Internet”）的 Internet 访问，并开启 SNAT（借用出接口地址）。根据需求开启 UTM 功能。




```bash
config firewall policy
    edit 2
        set name "to_Internet"
        set srcintf "port5"
        set dstintf "to_Internet"
        set action accept
        set srcaddr "192.168.10.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```



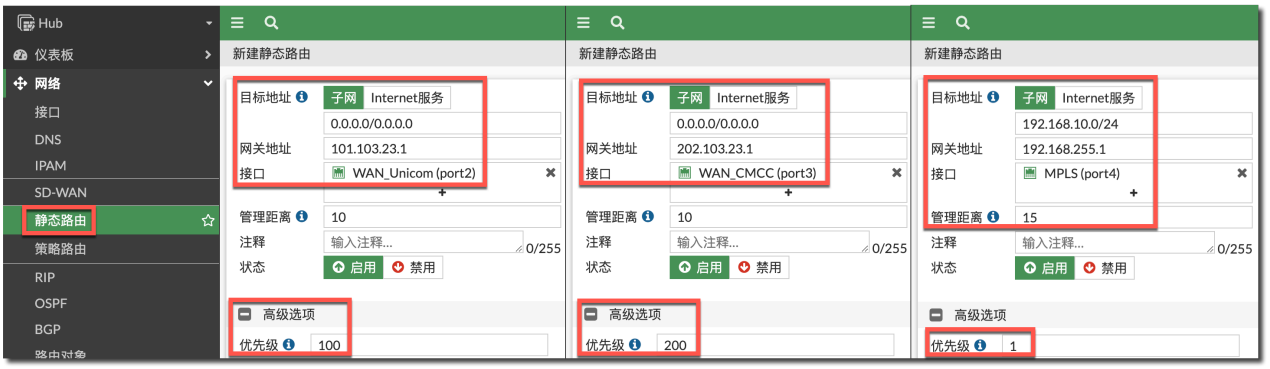
Hub

1. Hub 端如果需要配置 SD-WAN，可以参考 Spoke 对应的配置。

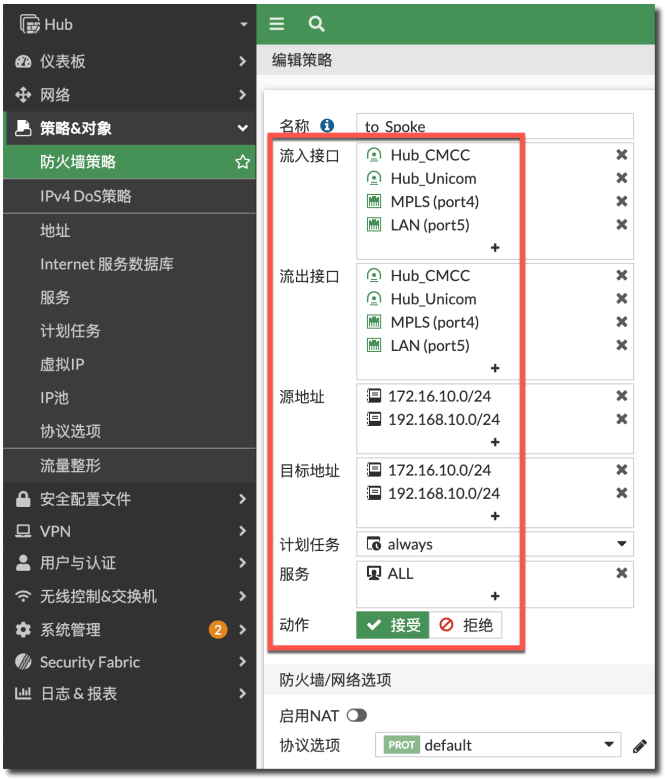
2. Hub 也可以不配置 SD-WAN。如果不配置 SD-WAN 选路，可以通过静态路由不同的优先级进行选路。这里我们以 Hub 端不配置 SD-WAN 为例。

3. 配置路由：

   - 指向联通线路（port2，优先级为 100）和移动线路（port3，优先级为 200）。优先从联通线路访问 Internet，联通线路 DOWN 时，从移动线路访问 Internet。
   - 通过 MPLS 专线（port4）去往 Spoke 内网的路由，管理距离配置为 15（由于 IPSec add-route 添加的路由 distance 默认为 15，这里需要配置 distance 为 15，与 IPSec add-route 添加的两条路由负载，否则无法进入路由表），优先级配置为默认的 1（优先于 IPSec add-route 添加的两条路由）。



4. 去往 Spoke 内网的路由通过 IPSec 一阶段配置中的 add-route 实现，联通线路 IPSec 自动添加的路由优先级为 100，移动线路 IPSec 自动添加的路由优先级为 200。
   - Hub 内网主动访问 Spoke 内网的流量，优先从联通线路的 IPSec 访问 Spoke 内网，联通线路 IPSec DOWN 时，从移动线路的 IPSec 访问 Spoke 内网。
   - Spoke 内网主动访问 Hub 内网的流量，在 Hub 的两条 IPSec 隧道中源进源出。
5. 配置防火墙策略 1，放通内网 port5（LAN）与 Spoke 内网之间互相访问的流量，通常情况下内网之间的访问不需要开启 SNAT。根据需求开启 UTM 功能。




```bash
config firewall policy
    edit 1
        set name "to_Spoke"
        set srcintf "Hub_CMCC" "Hub_Unicom" "port5" "port4"
        set dstintf "Hub_CMCC" "Hub_Unicom" "port5" "port4"
        set action accept
        set srcaddr "172.16.10.0/24" "192.168.10.0/24"
        set dstaddr "172.16.10.0/24" "192.168.10.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置防火墙策略 2，放通内网 port5（LAN）到 2 个运营商线路的 Internet 访问（配置前在“系统管理 → 可见功能”中开启“多接口策略”），并开启 SNAT（借用出接口地址）。根据需求开启 UTM 功能。




```bash
config firewall policy
    edit 2
        set name "to_Internet"
        set srcintf "port5"
        set dstintf "port2" "port3"
        set action accept
        set srcaddr "172.16.10.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

状态检查

1. 查看 Hub 端 IPSec 隧道的建立状态，两条线路上的 IPSec 均有一个拨号连接建立成功。




```bash
Hub # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
......
------------------------------------------------------
```

```
name=Hub_CMCC_0 ver=2 serial=4 202.103.23.2:0->202.103.12.2:0 tun_id=202.103.12.2 tun_id6=::10.0.0.1 dst_mtu=1500 dpd-
link=on weight=1
bound_if=5 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accep0

parent=Hub_CMCC index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=6
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Hub_CMCC proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:192.168.10.0-192.168.10.255:0
  SA:  ref=3 options=20683 type=00 soft=0 mtu=1438 expire=42881/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43191/43200
  dec: spi=f6555239 esp=aes key=16 9a00355173ec08c39a49eae81d06f85c
       ah=sha1 key=20 217275e45ec62629aee93a6e786f411e9b3469c9
  enc: spi=1b2b0940 esp=aes key=16 2b84a446403798f83ff9ccc559590fac
       ah=sha1 key=20 a811652f1656b85115327fe3504fbfa29566d587
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=202.103.12.2 npu_lgwy=202.103.23.2 npu_selid=0 dec_npuid=0 enc_npuid=0
--------------------------------------------------------

......
--------------------------------------------------------
name=Hub_Unicom_0 ver=2 serial=5 101.103.23.2:0->101.103.12.2:0 tun_id=101.103.12.2 tun_id6=::10.0.0.4 dst_mtu=1500 dpd-
link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accep0

parent=Hub_Unicom index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=6
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Hub_Unicom proto=0 sa=1 ref=2 serial=4 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:192.168.10.0-192.168.10.255:0
  SA:  ref=3 options=20683 type=00 soft=0 mtu=1438 expire=42972/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=f655523d esp=aes key=16 2a81dd8faebba5ead18f799af04aebdf
       ah=sha1 key=20 2769c11ebb1bb5fd25f6cafcfa830b738001857a
  enc: spi=1b2b0944 esp=aes key=16 ee9877c4c329c9d95a194859439bcbbc
       ah=sha1 key=20 11338a4d28ff61cbc71052f326d2a3719c07f323
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=101.103.12.2 npu_lgwy=101.103.23.2 npu_selid=4 dec_npuid=0 enc_npuid=0
```

2. 查看分支（Spoke）FortiGate 路由表，默认路由在 2 个运营商线路上负载，去往 Hub 端内网的路由在两条线路的 IPSec 和 MPLS 线路上负载。

```
Spoke # get router info routing-table all                                                                    bash

Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 101.103.12.1, port2, [1/0]
                  [1/0] via 202.103.12.1, WAN_CMCC_PPPoE, [1/0]
C       101.103.12.0/24 is directly connected, port2
S       172.16.10.0/24 [1/0] via to_Hub_Unicom tunnel 101.103.23.2, [1/0]
                       [1/0] via 192.168.254.1, port4, [1/0]
                       [1/0] via to_Hub_CMCC tunnel 202.103.23.2, [1/0]
C       192.168.10.0/24 is directly connected, port5
C       192.168.254.0/24 is directly connected, port4
C       202.103.12.1/32 is directly connected, WAN_CMCC_PPPoE
C       202.103.12.2/32 is directly connected, WAN_CMCC_PPPoE
```

3. 查看总部（Hub）FortiGate 路由表，默认路由在 2 个运营商线路上负载，联通线路的优先级为 100，移动线路的优先级为 200；去往 Spoke 端内网的路由在两条线路的 IPSec 和 MPLS 线路上负载，MPLS 线路 IPSec 的优先级为 1，联通线路 IPSec 的优先级为 100，移动线路 IPSec 的优先级为 200。

```
Hub # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.103.23.1, port2, [100/0]
                  [10/0] via 202.103.23.1, port3, [200/0]
C       101.103.23.0/24 is directly connected, port2
C       172.16.10.0/24 is directly connected, port5
S       192.168.10.0/24 [15/0] via 192.168.255.1, port4, [1/0]
                        [15/0] via Hub_Unicom tunnel 101.103.12.2, [100/0]
                        [15/0] via Hub_CMCC tunnel 202.103.12.2, [200/0]
C       192.168.255.0/24 is directly connected, port4
C       202.103.23.0/24 is directly connected, port3
```

bash

SD-WAN 策略配置

需求举例 1

Spoke 去往 Hub 内网的流量优先从 MPLS 专线（port4）转发，MPLS 专线中断或不满足 SLA 标准后，优先从联通线路的 IPSec 隧道转发，其次是移动线路的 IPSec 隧道。

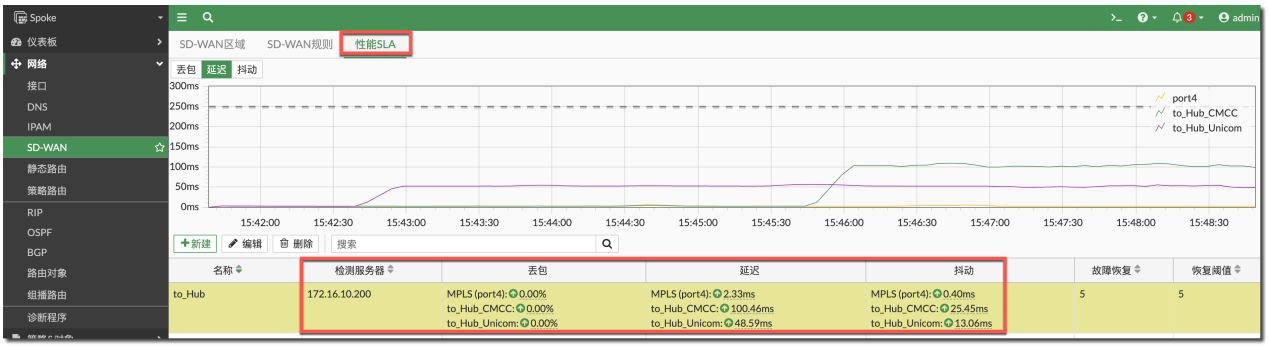
1. 在 Spoke 的 SD-WAN 中配置探测 Hub 端内网服务器的健康检查：

- 成员选择 MPLS 专线接口（port4）、联通线路的 IPSec 接口和移动线路的 IPSec 接口。
- 探测模式为 tcp-connect（TCP 三次握手），探测端口为 8080，指定探测的源 IP 为 Spoke 内网口 IP（需要通过 CLI 配置）。
- 开启 SLA 目标：延迟 < 250ms/抖动 < 50ms/丢包率 < 5%。

```
config system sdwan
    config health-check
        edit "to_Hub"
            set server "172.16.10.200"
            set protocol tcp-connect    //探测模式为TCP三次握手//
            set port 8080    //探测端口为8080//
            set source 192.168.10.1    //指定源IP为Spoke内网口地址（在IPSec感兴趣流中）//
            set members 5 4 3
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
end
```

bash

2. 查看健康检查的状态，可以看到目前 3 条去往 Hub 内网的线路都满足 SLA 标准。



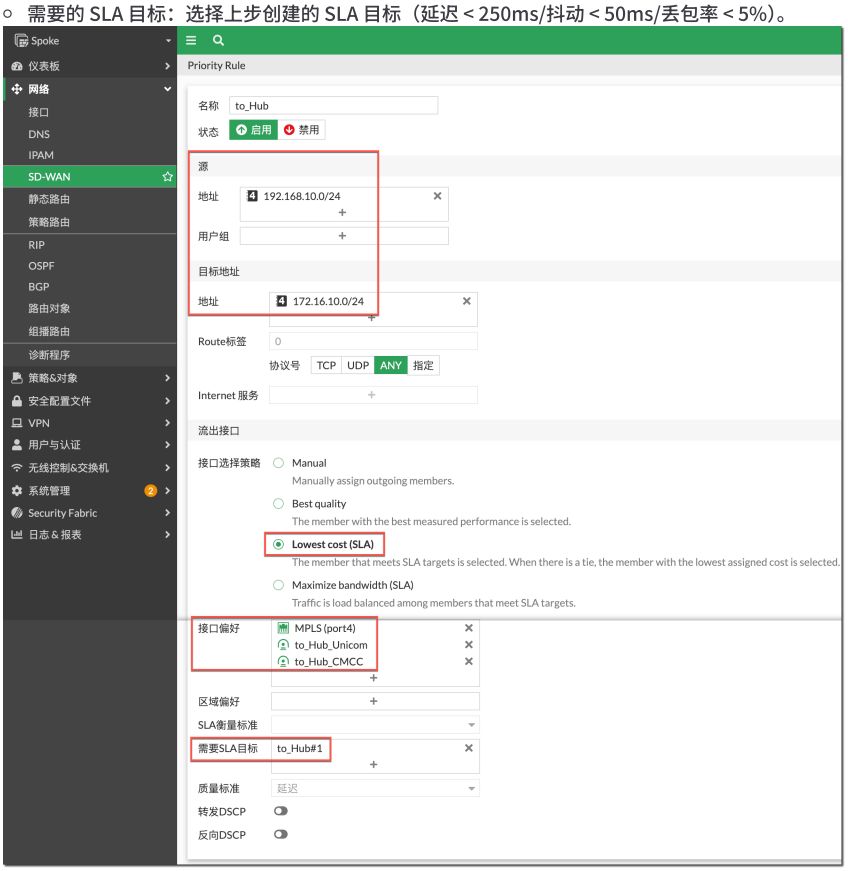

```
Spoke # diagnose sys sdwan health-check status to_Hub
Health Check(to_Hub):
Seq(5 port4): state(alive), packet-loss(0.000%) latency(2.661), jitter(0.672), mos(4.402), bandwidth-up(65534998),
bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0×1
Seq(4 to_Hub_CMCC): state(alive), packet-loss(0.000%) latency(108.022), jitter(17.742), mos(4.323), bandwidth-
up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0×1
Seq(3 to_Hub_Unicom): state(alive), packet-loss(0.000%) latency(51.611), jitter(13.453), mos(4.362), bandwidth-
up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0×1
```

bash

3. 新建 SD-WAN 规则 1：

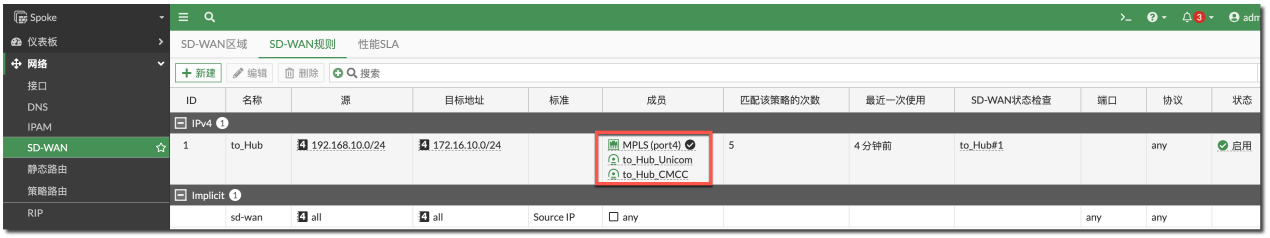


- 源地址选择 Spoke 内网网段；目标地址为 Hub 内网网段。
- 接口选择策略：选择 Lowest Cost (SLA)，Spoke 内部访问 Hub 内网的流量优先从满足 SLA 标准（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）的线路转发。如果 3 条线路都满足 SLA 标准，则优先从接口偏好中靠前的接口转发。
- 接口偏好：选择 MPLS 专线接口（port4）、联通线路 IPSec 接口与移动线路 IPSec 接口。MPLS 专线接口（port4）排序在前，其次是联通线路 IPSec 接口，最后是移动线路 IPSec 接口。
- 需要的 SLA 目标：选择上步创建的 SLA 目标（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）。




```bash
config system sdwan
    config service
        edit 1
            set name "to_Hub"
            set mode sla
            set dst "172.16.10.0/24"
            set src "192.168.10.0/24"
            config sla
                edit "to_Hub"
                    set id 1
                next
            end
            set priority-members 5 3 4
        next
    end
end
```

4. 查看上步创建的 SD-WAN 规则目前的选路状态，由于目前 3 条去往 Hub 内网的线路均满足 SLA 目标，所以选择接口偏好配置中靠前的 MPLS 线路接口（port4）。




```bash
Spoke # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(5 port4), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected    //优先走MPLS专线接口//
    2: Seq_num(3 to_Hub_Unicom), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    3: Seq_num(4 to_Hub_CMCC), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
  Src address(1):
        192.168.10.0-192.168.10.255

  Dst address(1):
        172.16.10.0-172.16.10.255
```

5. 查看 SD-WAN 规则对应的策略路由状态，出接口优先选择 MPLS 线路接口（port4）。

```bash
Spoke # diagnose firewall proute list
list route policy info(vf=root):

id=2134835201(0x7f3f0001) vwl_service=1(to_Hub) vwl_mbr_seq=5 3 4 dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(3) oif=6(port4) oif=22(to_Hub_Unicom) oif=23(to_Hub_CMCC)    //优先从MPLS线路接口转发//
source(1): 192.168.10.0-192.168.10.255
destination(1): 172.16.10.0-172.16.10.255
hit_count=5 last_used=2023-12-21 16:02:01
```

需求举例 2

Spoke 内部访问 Internet 的流量优先从满足 SLA 标准（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）的线路转发。如果两条 Internet 线路都满足 SLA 标准，则优先从联通线路转发。

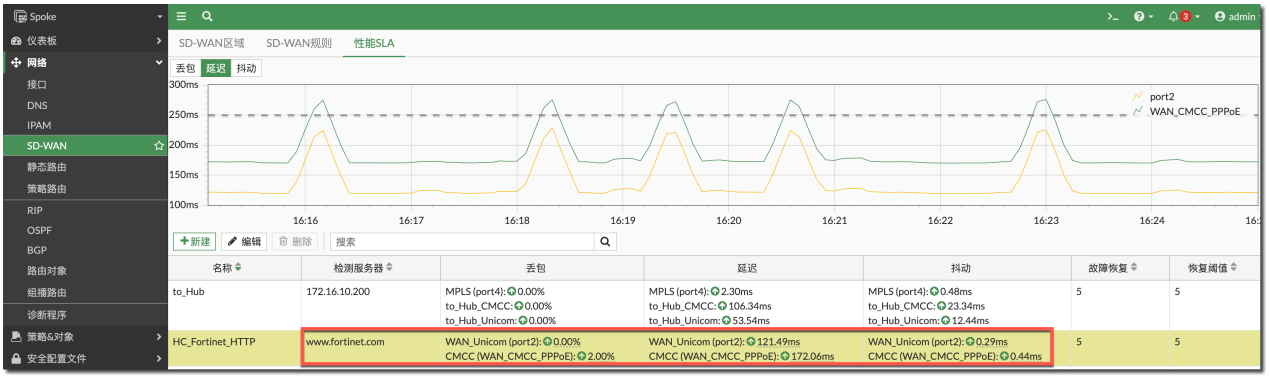
1. 在 Spoke 的 SD-WAN 中配置探测 Internet 某个网站的 HTTP 健康检查，成员选择联通（port2）和移动（PPPoE）线路，开启 SLA 目标：延迟 < 250ms/抖动 < 50ms/丢包率 < 5%。






```bash
config system sdwan
    config health-check
        edit "HC_Fortinet_HTTP"
            set server "www.fortinet.com"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
end
```

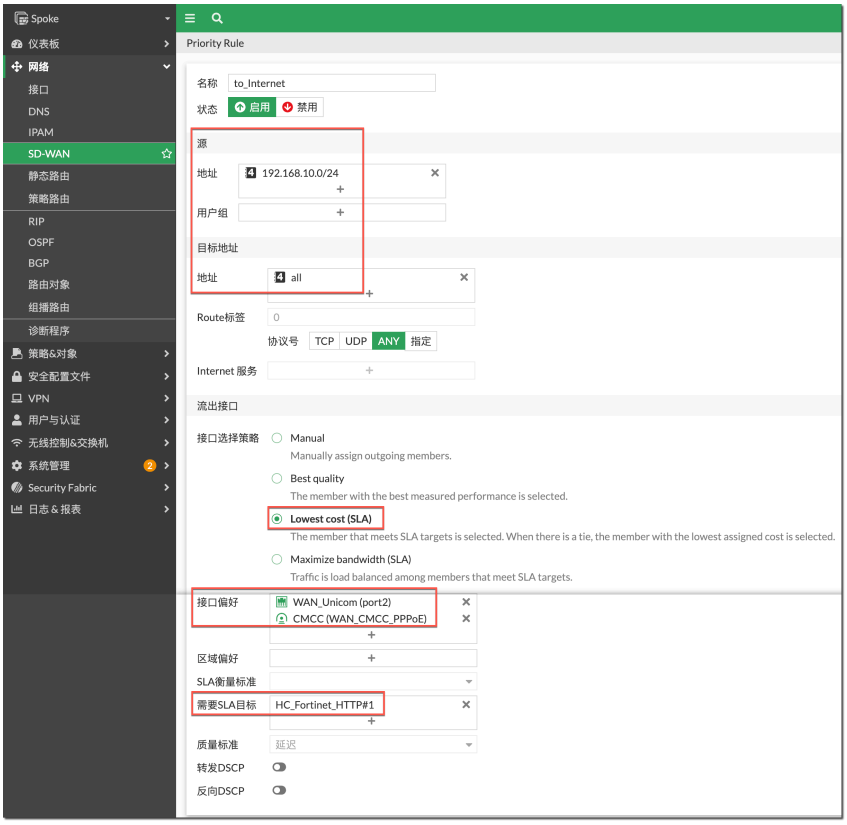
2. 查看健康检查的状态，可以看到目前两条 Internet 线路都满足 SLA 标准。




```bash
Spoke # diagnose sys sdwan health-check status HC_Fortinet_HTTP
Health Check(HC_Fortinet_HTTP):
Seq(1 port2): state(alive), packet-loss(1.000%) latency(115.769), jitter(0.618), mos(4.339), bandwidth-up(65534995), bandwidth-dw(65534996), bandwidth-bi(131069991) sla_map=0x1
Seq(2 WAN_CMCC_PPPoE): state(alive), packet-loss(2.000%) latency(166.407), jitter(0.782), mos(4.269), bandwidth-up(65534996), bandwidth-dw(65534997), bandwidth-bi(131069993) sla_map=0x1
```

3. 新建 SD-WAN 规则 2：

- 源地址选择内网网段；目标地址为 all。
- 接口选择策略：选择 Lowest Cost (SLA)，Spoke 内部访问 Internet 的流量优先从满足 SLA 标准（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）的线路转发。如果两条 Internet 线路都满足 SLA 标准，则优先从接口偏好中靠前的接口转发。
- 接口偏好：选择联通线路接口（port2）与移动线路接口（PPPoE），联通线路接口排序在前。
- 需要 SLA 目标：选择上步创建的 SLA 目标（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）。




```bash
config system sdwan
    config service
        edit 2
            set name "to_Internet"
            set mode sla
            set dst "all"
            set src "192.168.10.0/24"
            config sla
                edit "HC_Fortinet_HTTP"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

4. 查看上步创建的 SD-WAN 规则目前的选路状态，由于目前两条 Internet 线路均满足 SLA 目标，所以选择接口偏好配置中靠前的联通线路（port2）。



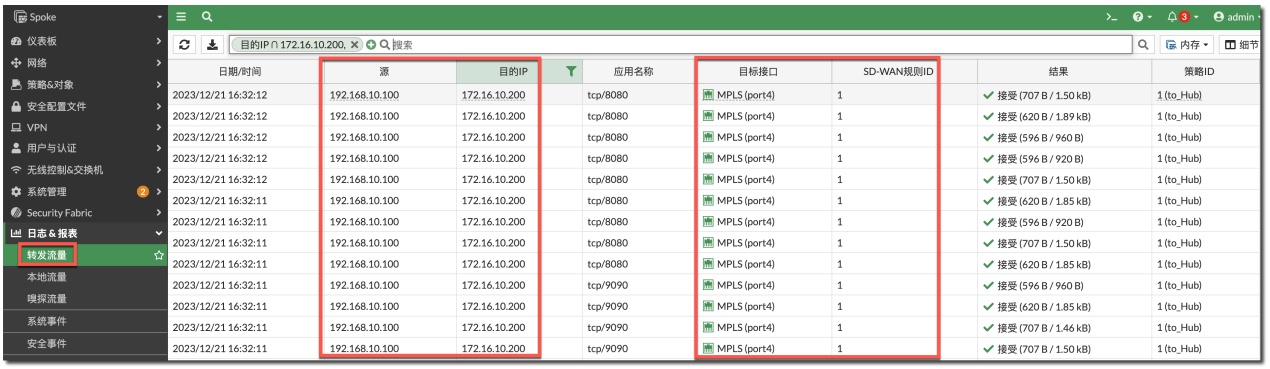

```bash
Spoke # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 port2), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected    //联通线路（port2）优先//
    2: Seq_num(2 WAN_CMCC_PPPoE), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
```

```
Src address(1):
      192.168.10.0-192.168.10.255

Dst address(1):
      0.0.0.0-255.255.255.255
```

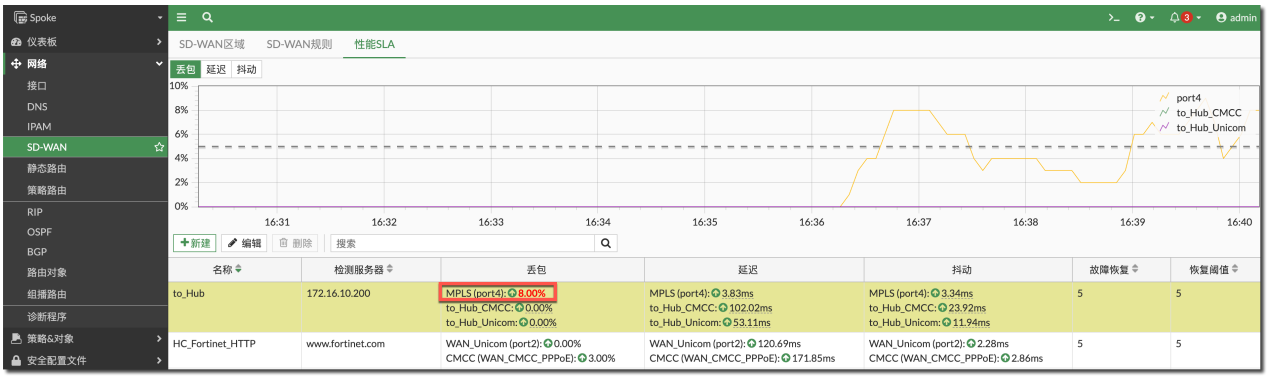
结果验证

Spoke 访问 Hub

1. 使用 Spoke 内网 PC 客户端访问 Hub 端的业务，流量可以正确的从 MPLS 线路接口（port4）转发。

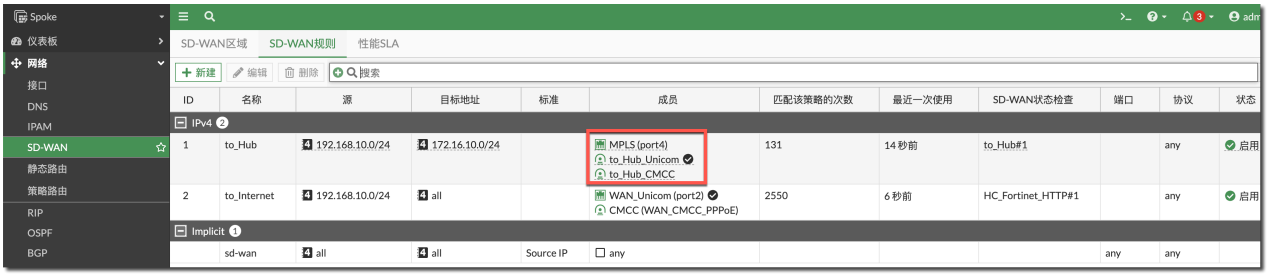


2. 模拟 MPLS 线路（port4）的健康检查的丢包率增加到 8%。



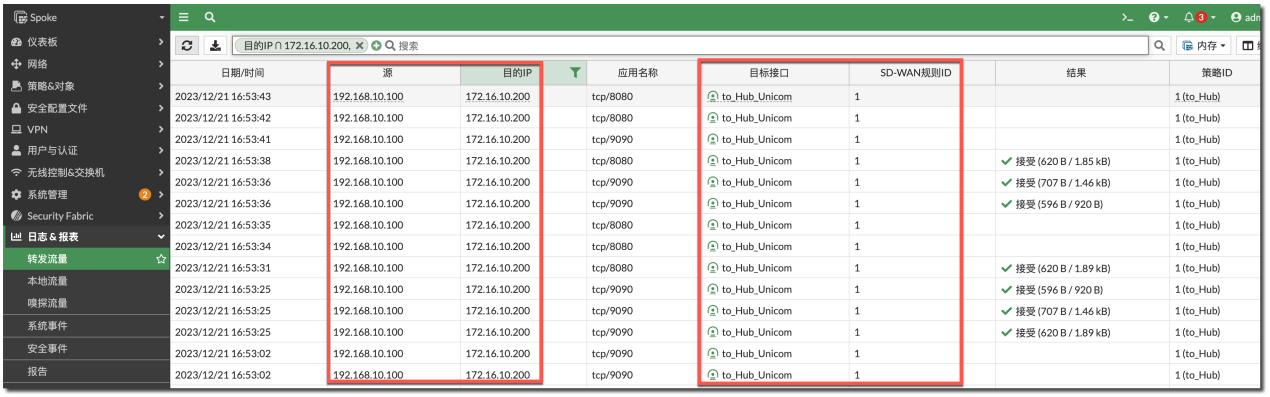

```bash
Spoke # diagnose sys sdwan health-check status to_Hub
Health Check(to_Hub):
Seq(5 port4): state(alive), packet-loss(8.000%) latency(2.444), jitter(0.530), mos(4.399), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x0
Seq(4 to_Hub_CMCC): state(alive), packet-loss(0.000%) latency(106.955), jitter(26.069), mos(4.295), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x1
Seq(3 to_Hub_Unicom): state(alive), packet-loss(0.000%) latency(57.014), jitter(16.204), mos(4.356), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x1
```

3. 查看 SD-WAN 规则 1 的选路优先选择联通线路 IPSec 接口去往 Hub 内网。

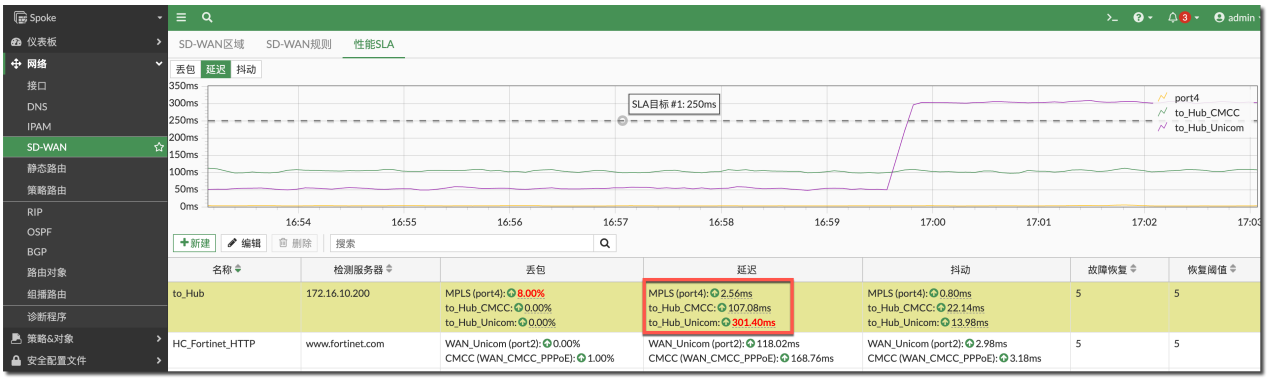


```bash
Spoke # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(22), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(3 to_Hub_Unicom), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected  //优先选择满足SLA线路中配置靠前的线路//
    2: Seq_num(4 to_Hub_CMCC), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
    3: Seq_num(5 port4), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected    //MPLS线路不满足SLA目标，被移至转发顺序最后//
  Src address(1):
        192.168.10.0-192.168.10.255

  Dst address(1):
        172.16.10.0-172.16.10.255
```

4. 再次使用 Spoke 内网 PC 客户端访问 Hub 端的业务，流量开始从联通线路 IPSec 接口转发。

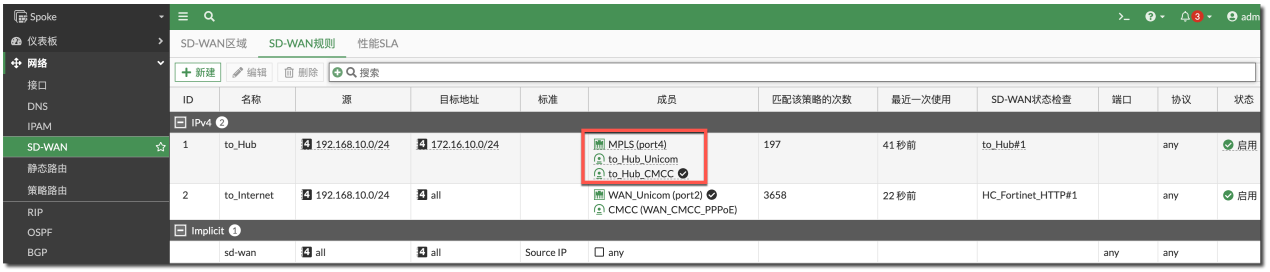


5. 模拟联通线路 IPSec 隧道健康检查的延迟增加到 300ms。



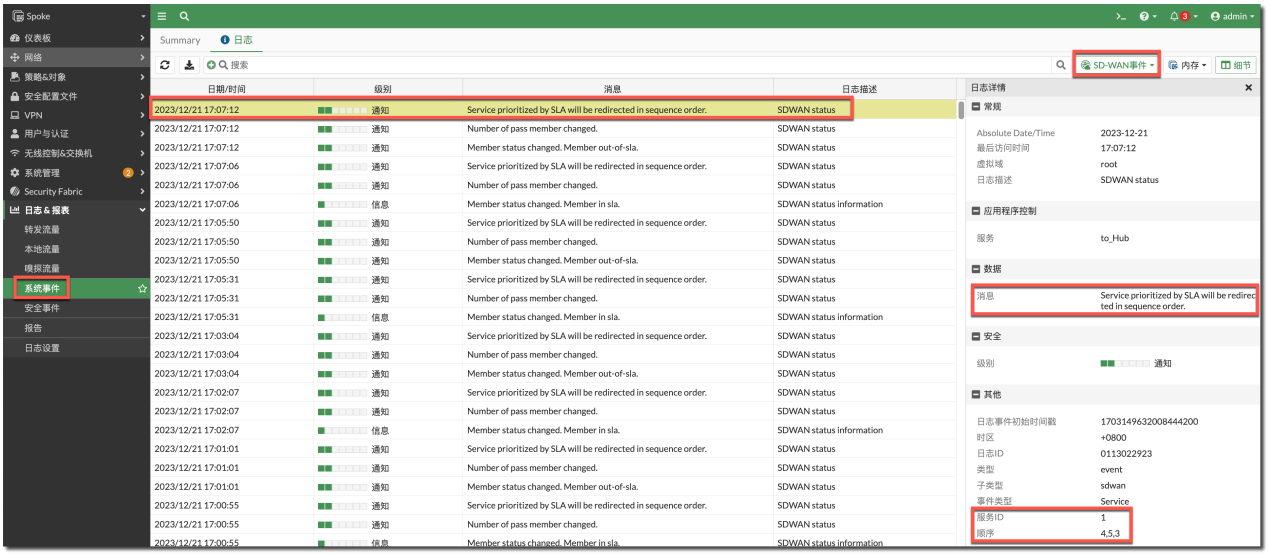

```bash
Spoke # diagnose sys sdwan health-check status to_Hub
Health Check(to_Hub):
Seq(5 port4): state(alive), packet-loss(8.000%) latency(2.540), jitter(0.578), mos(4.399), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x0
Seq(4 to_Hub_CMCC): state(alive), packet-loss(0.000%) latency(106.433), jitter(17.167), mos(4.325), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x1
Seq(3 to_Hub_Unicom): state(alive), packet-loss(0.000%) latency(303.372), jitter(15.259), mos(3.635), bandwidth-up(65534998), bandwidth-dw(65534999), bandwidth-bi(131069997) sla_map=0x0
```

6. 查看 SD-WAN 规则 1 的选路优先选择移动线路 IPSec 接口去往 Hub 内网。



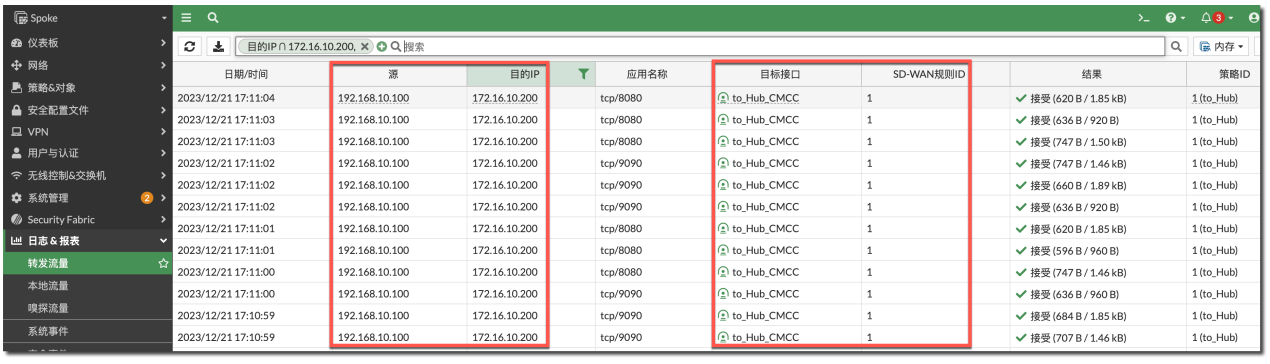

```bash
Spoke # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(33), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(sla), sla-compare-order
  Members(3):
    1: Seq_num(4 to_Hub_CMCC), alive, sla(0×1), gid(0), cfg_order(2), local cost(0), selected    //优先从移动线路IPSec隧道转发//
    2: Seq_num(5 port4), alive, sla(0×0), gid(0), cfg_order(0), local cost(0), selected    //MPLS线路不满足SLA目标//
    3: Seq_num(3 to_Hub_Unicom), alive, sla(0×0), gid(0), cfg_order(1), local cost(0), selected    //联通线路IPSec隧道不满足SLA目标//
  Src address(1):
        192.168.10.0-192.168.10.255

  Dst address(1):
        172.16.10.0-172.16.10.255
```

7. 对应的 SD-WAN 线路切换系统日志。




```bash
date=2023-12-21 time=17:07:12 eventtime=1703149632008444112 tz="+0800" logid="0113022923" type="event" subtype="sdwan"
level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="to_Hub" seq="4,5,3"
msg="Service prioritized by SLA will be redirected in sequence order."
```

8. 再次使用 Spoke 内网 PC 客户端访问 Hub 端的业务，流量开始从移动线路 IPSec 接口转发。



Spoke 访问 Internet

测试结果请参考 Spoke 访问 Hub 内网，线路可以根据健康检查结果是否满 SLA 目标正常切换，这里不再赘述。

注意事项

1. 默认配置下，当 Internet 线路（有 SNAT）SD-WAN 选路出现变化后，已有的会话不会立即切换到新链路上，只有线路切换后新创建的会话会建立在新的线路上，如果想在 SD-WAN 线路切换后让已有会话立刻失效，需要使用如下命令（可能会增加 CPU 消耗，谨慎开启）。

   > **重要**
   >
   > 默认配置下的 SNAT 场景，即使转发流量的 SD-WAN 成员的健康检查变为 dead，只要该线路仍然存在路由（可能关闭了 `update-static-route`），已有的旧会话仍然不会切换到其他线路。

   ```bash
   config system global
       set snat-route-change enable    //默认为disable//
   end
   ```

2. 默认配置下，当 VPN 或专线线路（无 SNAT）SD-WAN 选路出现变化后，已有的会话会立即切换到新链路上，如果想在 SD-WAN 线路切换后让已有会话保持在原来的线路上，只有新会话创建在新线路上，需要在对应 VPN 接口或专线接口上使用如下命令。

   ```bash
   config system interface
       edit "to_Hub_Unicom"
           set preserve-session-route enable    //默认为disable//
       next
   end
   ```

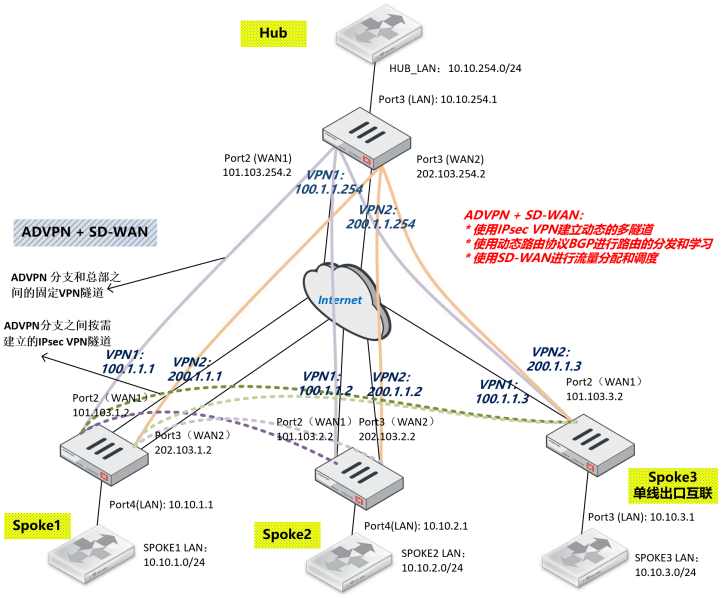
3. 如何设置 `add-route` 功能添加的路由优先级（priority）和管理距离（distance）？开启 `add-route` 后，默认添加的路由优先级为 1，管理距离为 15，如果要自定义路由优先级或管理距离，请参考如下配置（注意只有拨号模式的 IPSec 可以使用 `add-route` 功能。

   ```bash
   config vpn ipsec phase1-interface
       edit "Hub_Unicom"
           set type dynamic
           set add-route enable
           set priority 100
           set distance 20
       next
   end
   ```

ADVPN 场景

> **重要**
>
> ADVPN 目前不支持 Maximize Bandwidth (SLA) 模式。

网络拓扑与需求



1. 使用 IPsec VPN 建立动态的多隧道，Hub 和 Spoke 之间是固定的 IPSec 隧道，而 Spoke 和 Spoke 之间则是流量触发的按需动态创建的 VPN 隧道。
2. 使用动态路由协议 BGP 进行路由的分发和学习，HUB 充当 BGP 路由反射器进行路由的传递，同时分支需要开启“additional-path”，形成 VPN 业务路由的负载分担 BGP 路由，这是 SD-WAN 可以调度的前提（路由表中有路由）。
3. 使用 SD-WAN 规则对流量进行分配和调度，可以配置特定的 SD-WAN 算法进行流量的调度。
4. 部署 SD-WAN 通常在一个全新的设备进行配置，如果需要加入 SD-WAN 区域的接口已经被其他配置（如防火墙策略）引用，是无法直接加入 SD-WAN 区域的。要加入 SD-WAN 区域，又不想删除关联的配置，请参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域↗章节。

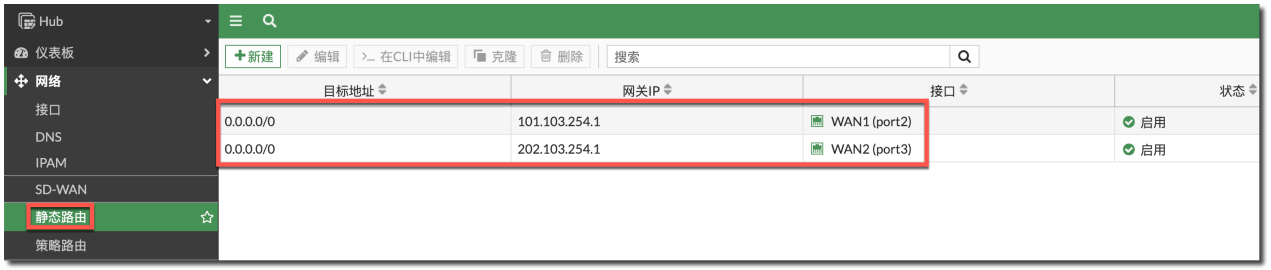
配置步骤

基础配置

Hub

1. 在 Hub 设备上配置 WAN1（port2）、WAN2（port3）以及 LAN（port4）的接口 IP（略）。

2. 配置两个 WAN 接口上访问 Internet 的默认路由。




```bash
config router static
    edit 1
        set gateway 101.103.254.1
        set device "port2"
    next
```

```
    edit 2
        set gateway 202.103.254.1
        set device "port3"
    next
end

Hub # get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.103.254.1, port2, [1/0]
                  [10/0] via 202.103.254.1, port3, [1/0]
C       10.10.254.0/24 is directly connected, port4
C       101.103.254.0/24 is directly connected, port2
C       202.103.254.0/24 is directly connected, port3
```

Spoke

1. 配置每个 Spoke FortiGate 运营商线路的 IP（如果是 PPPoE 线路，PPPoE 请通过 CLI 配置，参考部署前注意事项 → PPPoE 接口配置 ↗章节）与内网口 IP（略）。
2. 在每个 Spoke 配置 WAN 接口上访问 Internet 的默认路由（略，参考 Hub 上的配置）。

IPSec 相关配置

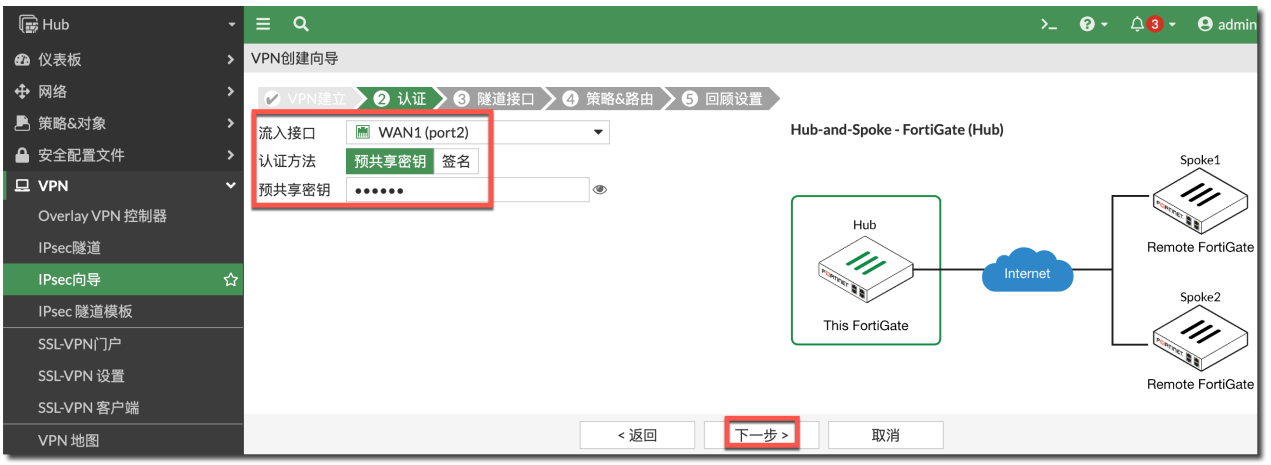
Hub

1. 通过 ADVPN 向导来配置 ADVPN，进入“VPN → IPSec 向导”页面，模板类型选择“Hub-and-Spoke”类型，角色选择 Hub，点击下一步。

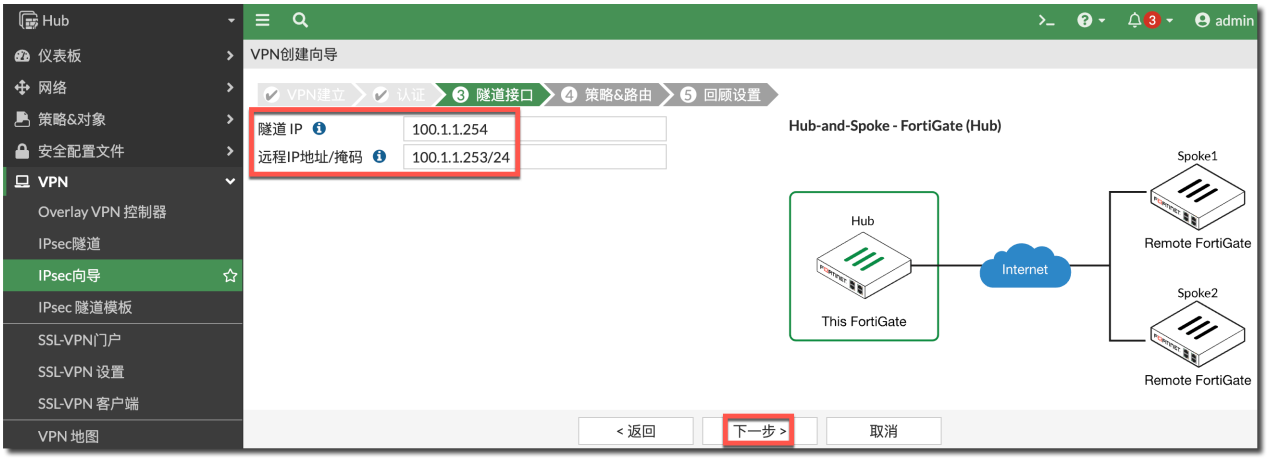
> **重要**
>
> 通过 ADVPN 向导来配置 ADVPN 可以快速地部署 ADVPN + BGP 环境，而不用在每个 Spoke 上进行相关的具体功能配置，大大增加了部署效率与难度，部署完成后，可以再根据实际需求对配置进行调整。



2. 配置 WAN1（port2）接口上的 IPSec 拨号连接，流入接口选择 WAN1（port2），配置认证方法，这里以预共享密钥为例，填入预共享密钥，点击下一步。

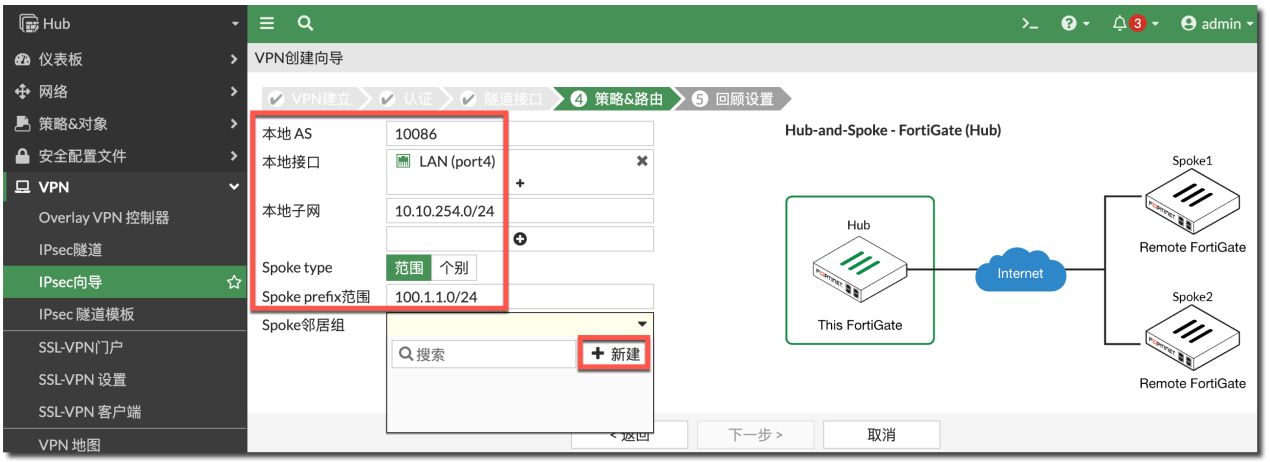


3. 配置 Hub 端 WAN1（port2）上的 IPSec Tunnel IP，用于 BGP 邻居建立，配置“远程 IP 地址/掩码”为一个 24 位网络地址（根据 Spoke 的数量判断掩码位数），点击下一步。

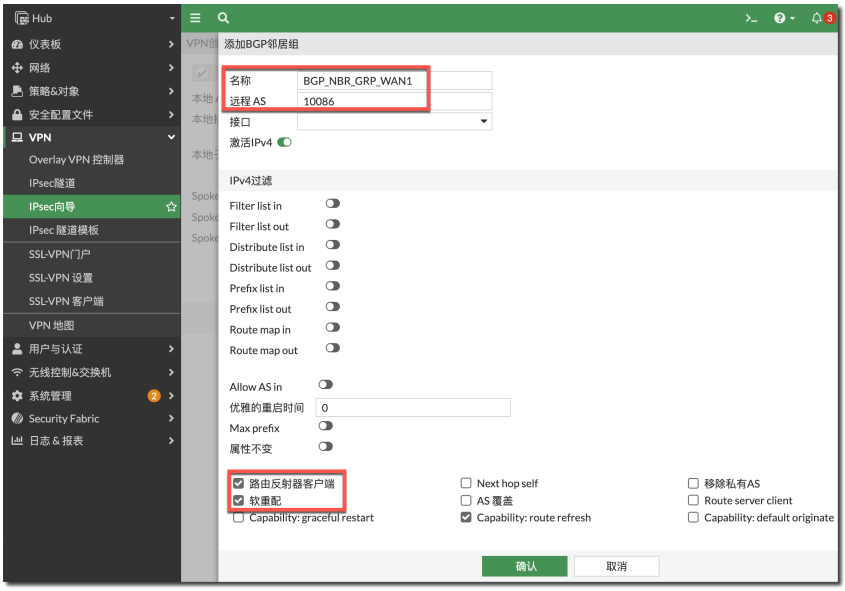


4. 配置 BGP 路由的本地 AS 号（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），Hub 端内网接口与网段（用于自动创建安全策略和 BGP 路由发布），Spoke type 选择“范围”，填写 Spoke prefix 范围为 WAN1 线路 IPSec 对应的 Spoke Tunnel 接口 IP 的范围子网，点击 Spoke 邻居组中的新建按钮。

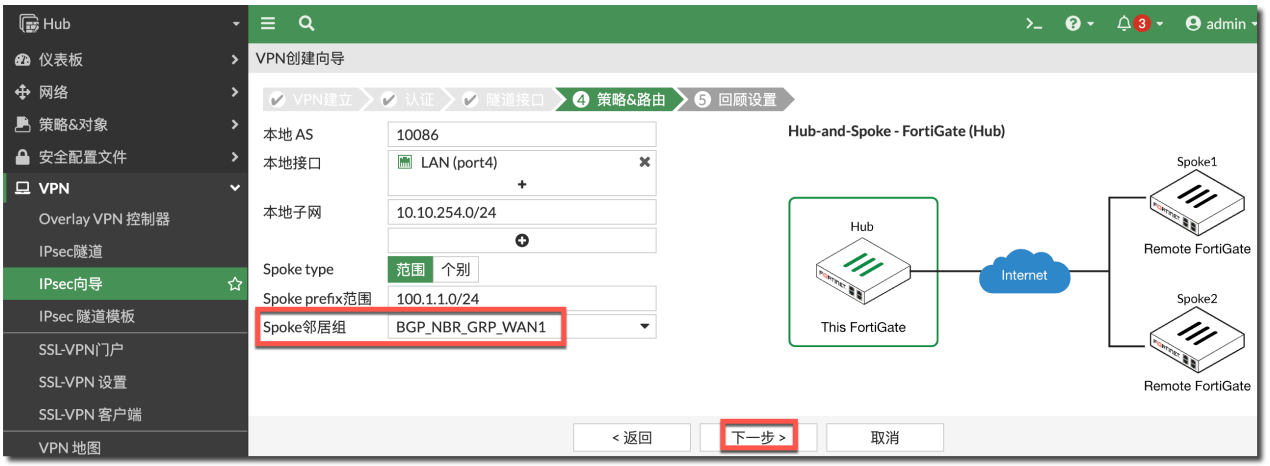
> **重要**
>
> Spoke type 推荐使用“范围”模式，并结合 BGP 邻居范围和 BGP 邻居组使用，否则如果 Spoke 较多，每个 Spoke 配置一个 BGP 邻居将是一个巨大的工作量。



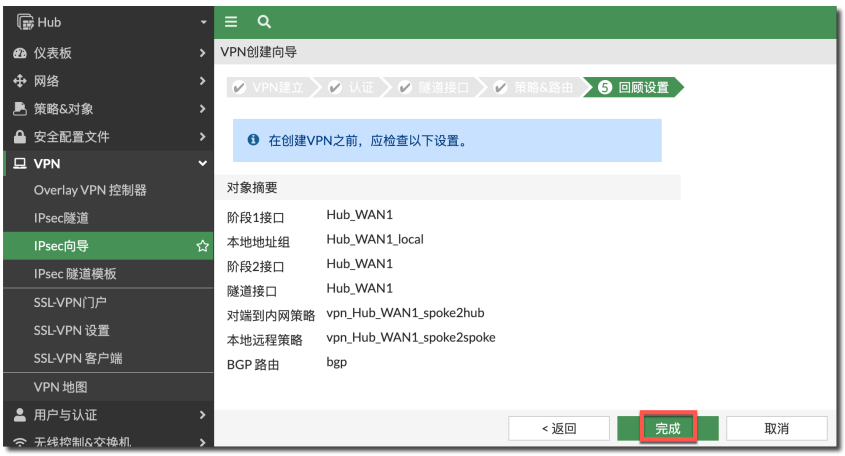
5. 在弹出的 BGP 邻居组窗口中，配置远程 AS（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），开启路由反射器和软重配。点击确认按钮下发 BGP 邻居组配置。



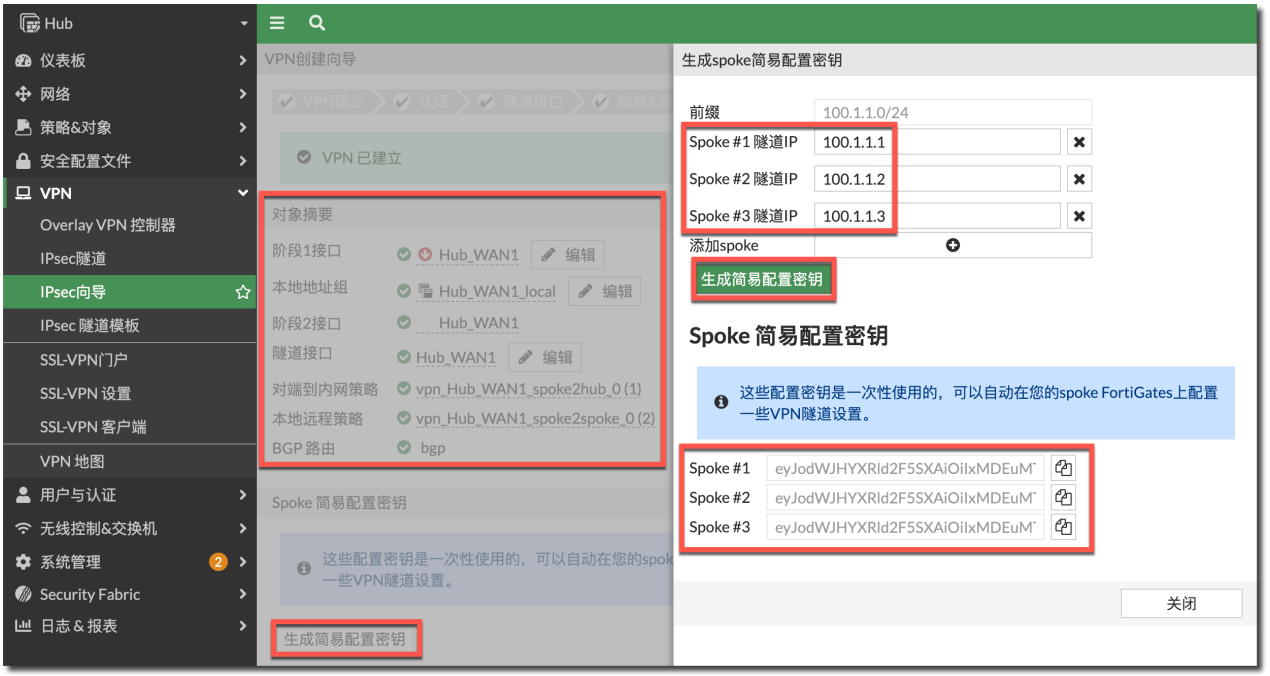
6. Spoke 邻居组选择上步创建的 BGP 邻居组，点击下一步。



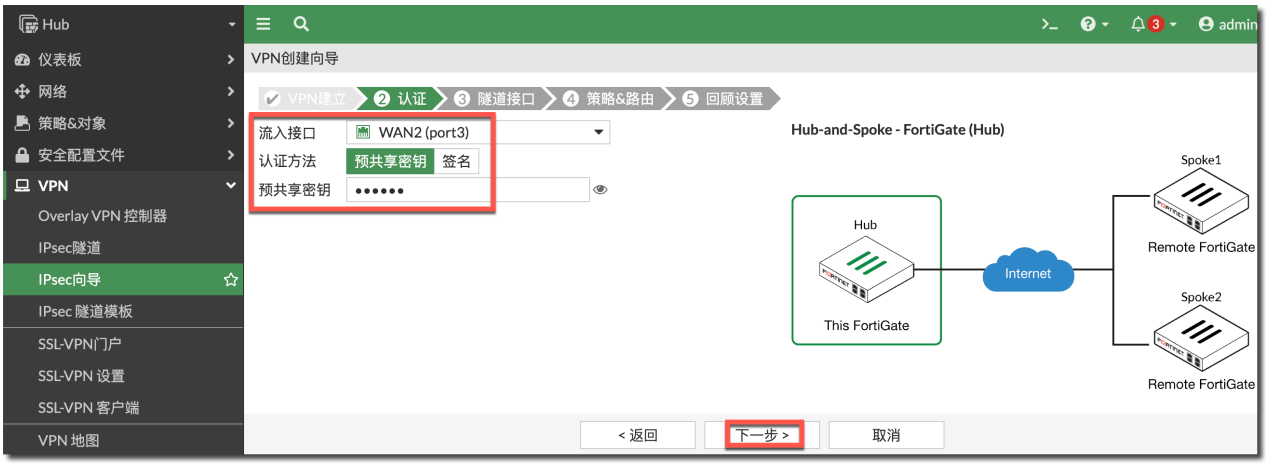
7. 查看即将创建的配置内容，点击完成按钮。



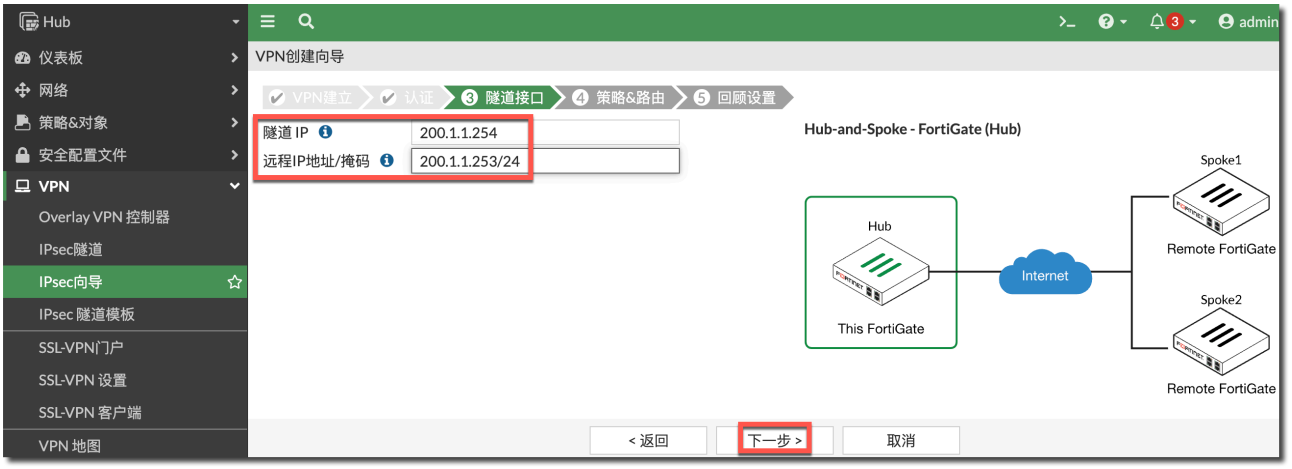
8. 检查对象摘要中的配置是否全部下发成功。然后点击下方的“生成简易配置密钥”按钮，在右侧弹出的窗口中，添加 WAN1（port2）线路 IPSec 隧道对应的三个 Spoke 的 Tunnel 接口 IP。然后点击“生成简易配置密钥”按钮，将生成的 3 个 Spoke 简易配置密钥复制出来并保存。



9. 配置 WAN2（port3）接口上的 IPSec 拨号连接，流入接口选择 WAN2（port3），配置认证方法，这里以预共享密钥为例，填入预共享密钥，点击下一步。



10. 配置 Hub 端 WAN2（port3）上的 IPSec Tunnel IP，用于 BGP 邻居建立，配置“远程 IP 地址/掩码”为一个 24 位网络地址（根据 Spoke 的数量判断掩码位数），点击下一步。



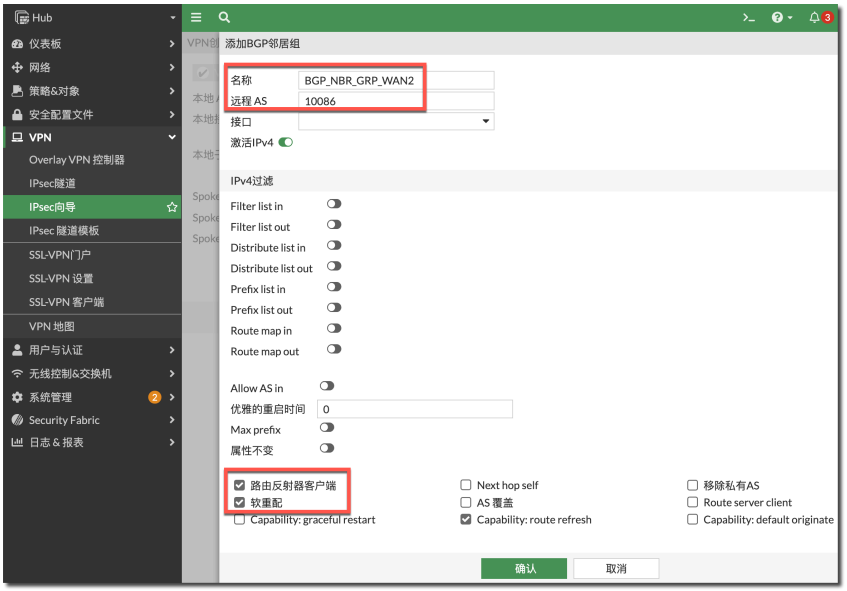
11. 配置 BGP 路由的本地 AS 号（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），Hub 端内网接口与网段（用于自动创建安全策略和 BGP 路由发布），Spoke type 选择“范围”，填写 Spoke prefix 范围为 WAN1 线路 IPSec 对应的 Spoke Tunnel 接口 IP 的范围子网，点击 Spoke 邻居组中的新建按钮。

重要

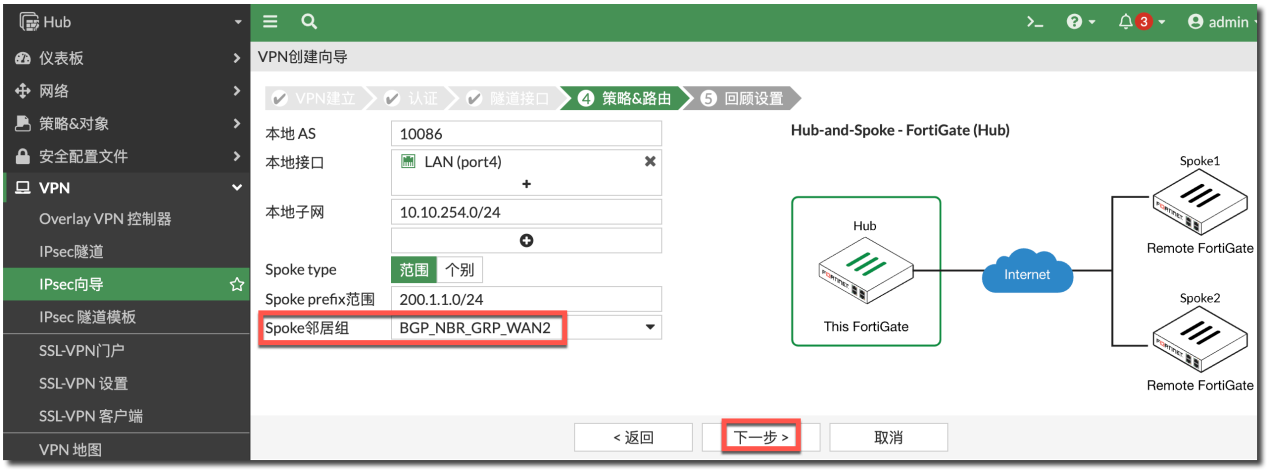
Spoke type 推荐使用“范围”模式，并结合 BGP 邻居范围和 BGP 邻居组使用，否则如果 Spoke 较多，每个 Spoke 配置一个 BGP 邻居将是一个巨大的工作量。



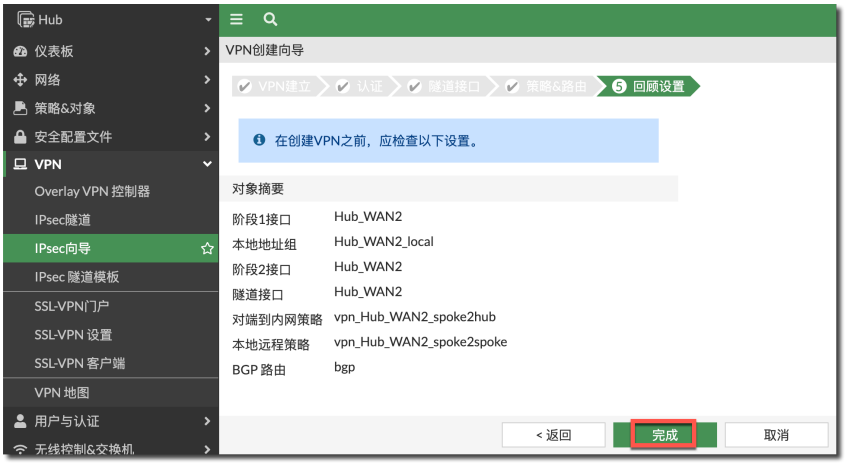
2. 在弹出的 BGP 邻居组窗口中，配置远程 AS（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），开启路由反射器和软重配。点击确认按钮下发 BGP 邻居组配置。



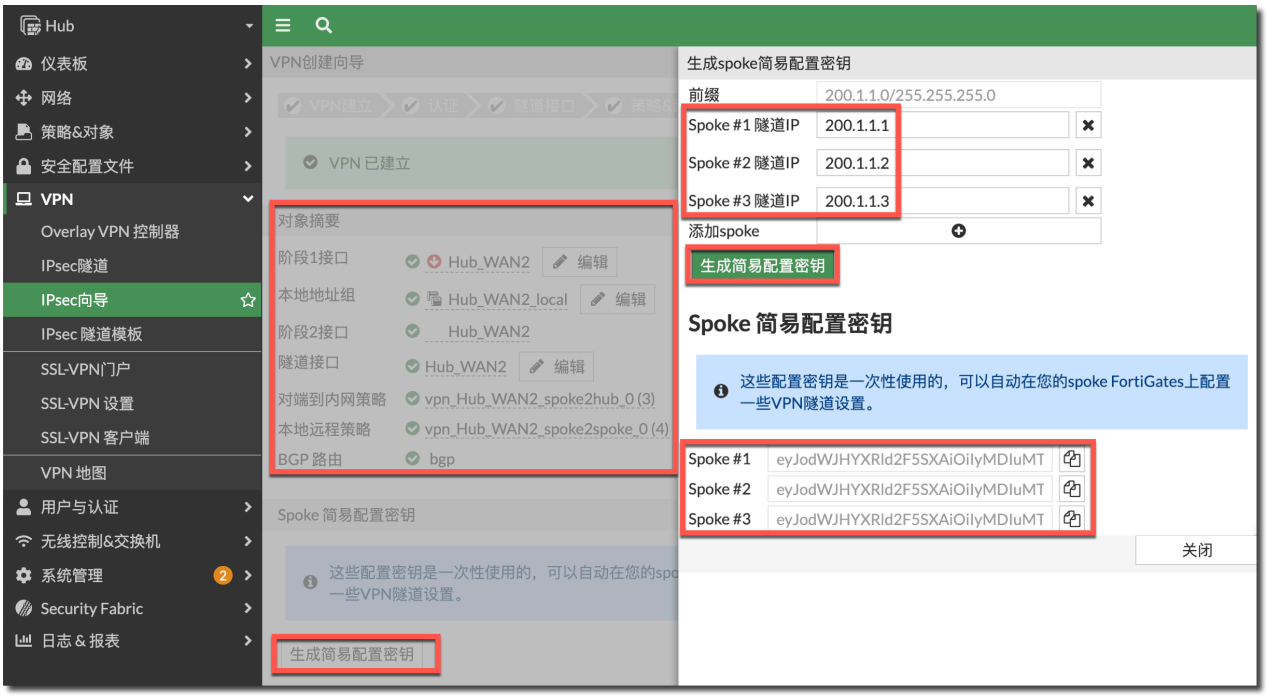
3. Spoke 邻居组选择上步创建的 BGP 邻居组，点击下一步。



4. 查看即将创建的配置内容，点击完成按钮。



5. 检查对象摘要中的配置是否全部下发成功。然后点击下方的“生成简易配置密钥”按钮，在右侧弹出的窗口中，添加 WAN1（port2）线路 IPSec 隧道对应的三个 Spoke 的 Tunnel 接口 IP。然后点击“生成简易配置密钥”按钮，将生成的 3 个 Spoke 简易配置密钥复制出来并保存。



6. 查看 IPSec 向导下发的 IPSec 配置，以及 IPSec Tunnel 接口配置。注意要在 IPSec 一阶段中增加下步中的配置。

```bash
config vpn ipsec phase1-interface
    edit "Hub_WAN1"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: Hub_WAN1 (Created by VPN wizard)"
        set wizard-type hub-fortigate-auto-discovery
        set auto-discovery-sender enable    //确认开启了ADVPN sender//
        set psksecret ENC
LRPVkrjrtBp5qTlA1Bo5E3M0Cr/79r/kqV7vZmqrNqckeTk2kJf2/W5Xi2lni2CPCfSukYx0CY9cyWwu6C2IuwxiK4chuJXZbuhYejUguiHmZE=
    next
    edit "Hub_WAN2"
        set type dynamic
        set interface "port3"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
```

```
        set dpd on-idle
        set comments "VPN: Hub_WAN2 (Created by VPN wizard)"
        set wizard-type hub-fortigate-auto-discovery
        set auto-discovery-sender enable    //确认开启了ADVPN sender//
        set psksecret ENC
g3UIj62MLpQkr791Zf+pujPQ2QqbzDnvcwLDmF5xr5apZLk8FDRs+YtDJGtJ5vnMEG6K2lEhluoO/M5gjGu5qjJatbTRvblp0FPBUQGW2X2eFY=
    next
end

config vpn ipsec phase2-interface
    edit "Hub_WAN1"
        set phase1name "Hub_WAN1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: Hub_WAN1 (Created by VPN wizard)"
    next
    edit "Hub_WAN2"
        set phase1name "Hub_WAN2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: Hub_WAN2 (Created by VPN wizard)"
    next
end

config system interface
    edit "Hub_WAN1"
        set vdom "root"
        set ip 100.1.1.254 255.255.255.255
        set type tunnel
        set remote-ip 100.1.1.253 255.255.255.0
        set interface "port2"
    next
    edit "Hub_WAN2"
        set vdom "root"
        set ip 200.1.1.254 255.255.255.255
        set type tunnel
        set remote-ip 200.1.1.253 255.255.255.0
        set interface "port3"
    next
end
```

7. 需要在向导配置的基础上，将 IKEv1 主模式修改为 IKEv2，并在每个线路的 IPSec 上开启 network-overlay，配置不同的 network-id，WAN1 上的 IPSec 的 network-id 为 1，WAN2 上的 IPSec 的 network-id 为 2。

> **重要**
>
> 使用 ADVPN 向导创建的 IPSec 为 IKEv1 主模式，Hub 双线路的情况下，Spoke 与 Spoke 在建立 Shortcut 时会出现问题（参考 VPN 技术 → IPSec VPN → 拨号 VPN → 野蛮模式/network-id 适用场景 → ADVPN 下的 network-id 或野蛮模式 ID ↗ 的说明）。

```bash
config vpn ipsec phase1-interface
    edit "Hub_WAN1"
        set ike-version 2             //增加配置：将IKE版本修改为2//
        set network-overlay enable    //增加配置：开启network-overlay//
        set network-id 1              //增加配置：第一条线路上的network-id为1//
    next
    edit "Hub_WAN2"
        set ike-version 2             //增加配置：将IKE版本修改为2//
        set network-overlay enable    //增加配置：开启network-overlay//
        set network-id 2              //增加配置：配置第二条线路上的network-id为2//
    next
end
```

8. 查看 IPSec 向导下发的地址、地址组和安全策略配置，包含放通 Hub 端 IPSec 隧道的入向流量、放通 IPSec 隧道的中转流量，可根据实际需求进行调整或新增（如 Hub 向 Spoke 主动发起的方向策略）。

> **重要**
>
> 后续如果需要在 Hub 上配置 SD-WAN，安全策略配置需要删除，否则 IPSec Tunnel 接口无法加入 SD-WAN 区域。

```bash
config firewall address
    edit "Hub_WAN1_local_subnet_1"
        set allow-routing enable
        set subnet 10.10.254.0 255.255.255.0
    next
    edit "Hub_WAN2_local_subnet_1"
        set allow-routing enable
        set subnet 10.10.254.0 255.255.255.0
    next
end
config firewall addrgrp
    edit "Hub_WAN1_local"
        set member "Hub_WAN1_local_subnet_1"
        set comment "VPN: Hub_WAN1 (Created by VPN wizard)"
        set allow-routing enable
    next
    edit "Hub_WAN2_local"
        set member "Hub_WAN2_local_subnet_1"
        set comment "VPN: Hub_WAN2 (Created by VPN wizard)"
        set allow-routing enable
    next
end

config firewall policy
    edit 1    //放通了WAN1上IPSec隧道的入向流量//
        set name "vpn_Hub_WAN1_spoke2hub_0"
        set srcintf "Hub_WAN1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Hub_WAN1_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Hub_WAN1 (Created by VPN wizard)"
    next
    edit 2    //放通了WAN1上IPSec隧道的中转流量//
        set name "vpn_Hub_WAN1_spoke2spoke_0"
        set srcintf "Hub_WAN1"
        set dstintf "Hub_WAN1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Hub_WAN1 (Created by VPN wizard)"
    next
    edit 3    //放通了WAN2上IPSec隧道的入向流量//
        set name "vpn_Hub_WAN2_spoke2hub_0"
        set srcintf "Hub_WAN2"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Hub_WAN2_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Hub_WAN2 (Created by VPN wizard)"
    next
    edit 4    //放通了WAN2上IPSec隧道的中转流量//
        set name "vpn_Hub_WAN2_spoke2spoke_0"
        set srcintf "Hub_WAN2"
        set dstintf "Hub_WAN2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Hub_WAN2 (Created by VPN wizard)"
    next
end
```

9. 查看 IPSec 向导下发的 BGP 配置。

```bash
config router bgp
    set as 10086
    config neighbor-group
        edit "BGP_NBR_GRP_WAN1"
            set soft-reconfiguration enable
            set remote-as 10086
            set route-reflector-client enable
        next
        edit "BGP_NBR_GRP_WAN2"
            set soft-reconfiguration enable
            set remote-as 10086
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 100.1.1.0 255.255.255.0
            set neighbor-group "BGP_NBR_GRP_WAN1"
        next
        edit 2
            set prefix 200.1.1.0 255.255.255.0
            set neighbor-group "BGP_NBR_GRP_WAN2"
        next
    end
    config network
        edit 1
            set prefix 10.10.254.0 255.255.255.0
        next
    end
end
```

10. 在 ADVPN 环境下，以上 BGP 配置还需要手动增加如下参数。

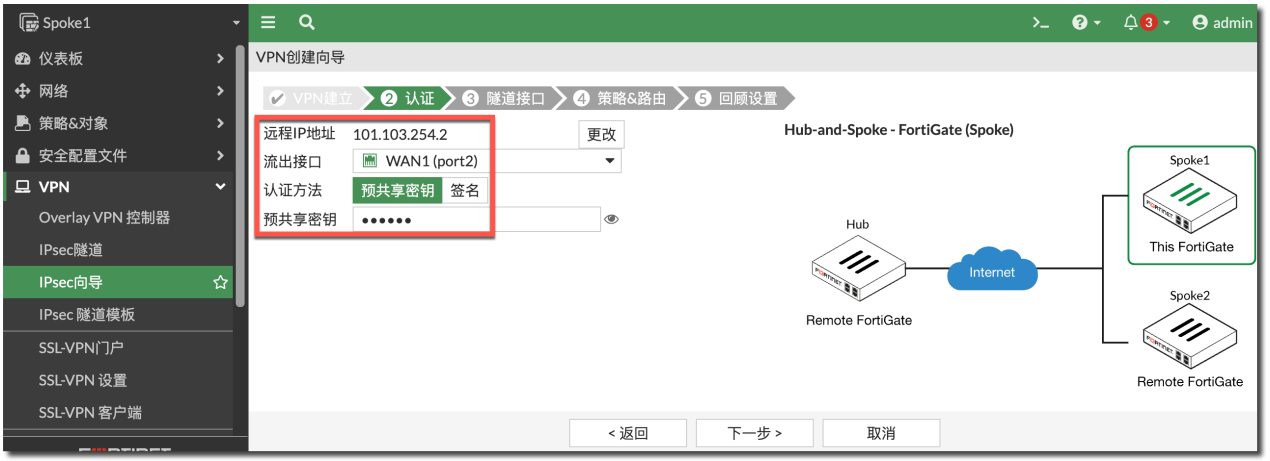
```bash
config router bgp
    set ibgp-multipath enable    //必须配置，否则去往Spoke的路由无法负载//
    set additional-path enable    //多线路ADVPN，Hub端必须配置，否则Spoke学到的路由可能不全//
    set additional-path-select 4
    config neighbor-group
        edit "BGP_NBR_GRP_WAN1"
            set link-down-failover enable
            set additional-path send    //多线路ADVPN，Hub端必须配置，否则Spoke学到的路由可能不全//
            set adv-additional-path 4    //配置从几条额外路径上去发送路由//
            set soft-reconfiguration enable
        next
        edit "BGP_NBR_GRP_WAN2"
            set link-down-failover enable
            set additional-path send    //多线路ADVPN，Hub端必须配置，否则Spoke学到的路由可能不全//
            set adv-additional-path 4    //配置从几条额外路径上去发送路由//
            set soft-reconfiguration enable
        next
    end
end
```

Spoke

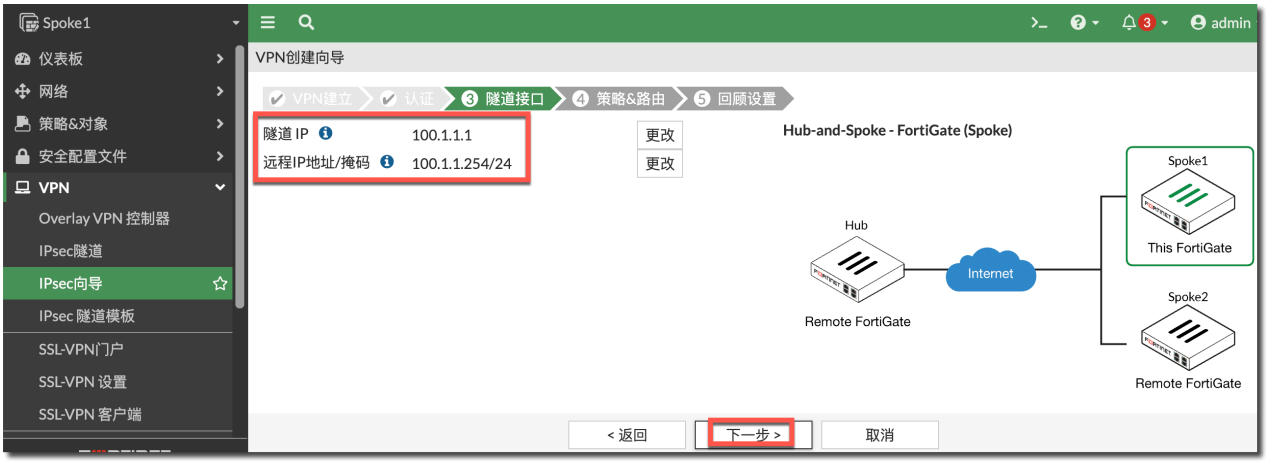
1. Hub 端通过向导配置 ADVPN 后，在 Spoke 端的配置就轻松很多。准备好在 Hub 端复制出的 Spoke 简易配置密钥。
2. 在 Spoke1 FortiGate 上进入 VPN → IPSec 隧道页面，新建 WAN1（port2）上的 IPSec 隧道，选择“Hub-and-Spoke”模式，角色选择 Spoke，将 Spoke1 WAN1 的 IPSec 隧道对应的简易配置密钥粘贴进去，然后点击右侧的应用按钮，会弹出提示“配置已应用”，然后点击下一步。



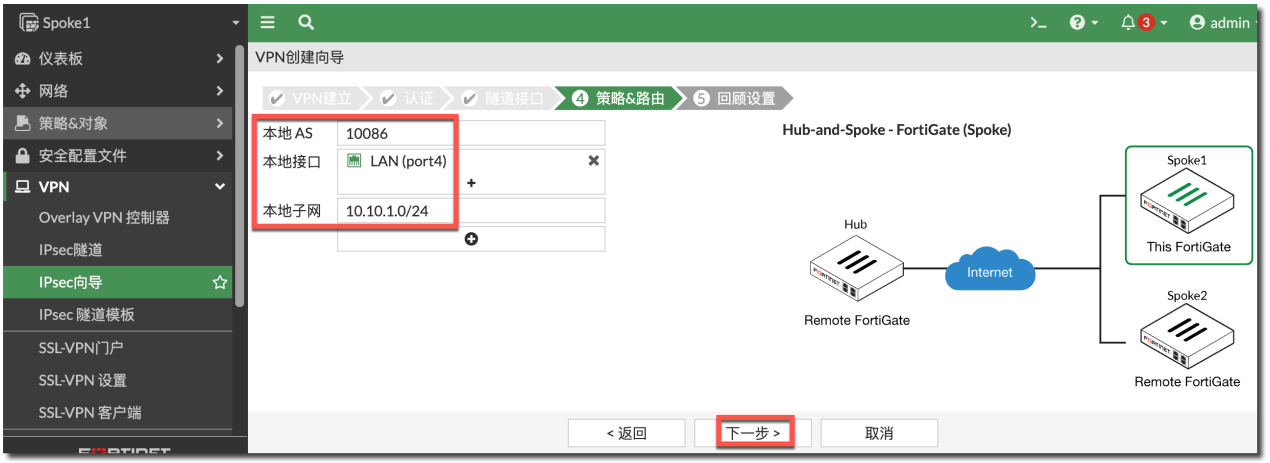
3. Spoke1 自动将 Hub 端 WAN1（port2）的 IP 地址填入，配置流出接口为 WAN1（port2），配置预共享密钥，点击下一步。



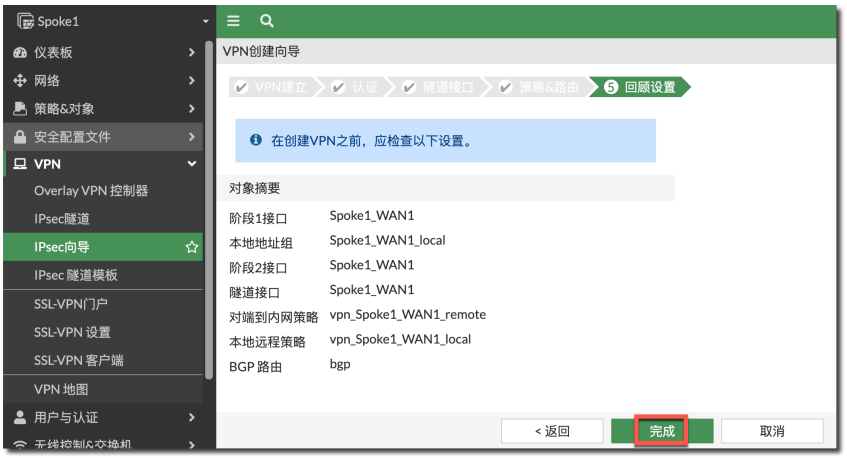
4. Spoke1 自动配置了 WAN1（port2）线路 IPSec Tunnel 口的地址（在 Hub 端向导中的配置），远程 IP 地址/掩码为 Hub 端 WAN1 线路 IPSec Tunnel 接口地址。点击下一步。



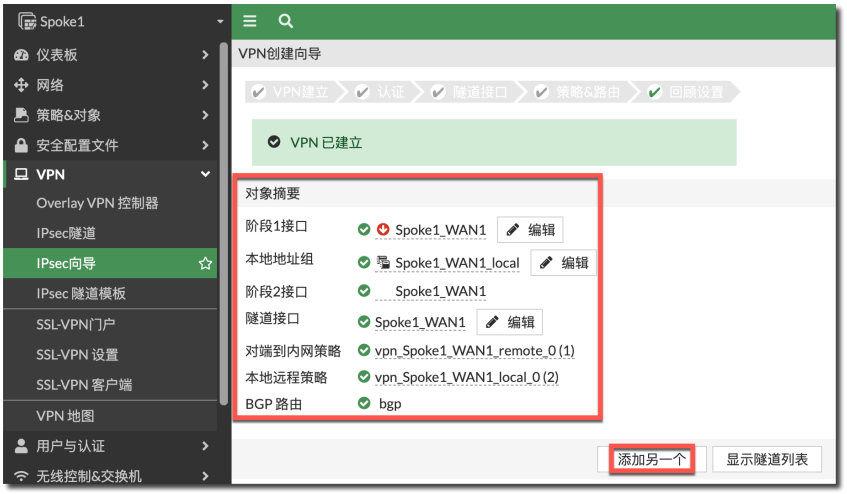
5. 配置 BGP 本地 AS（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），配置本地接口和本地子网（用于自动创建安全策略和 BGP 路由发布）。点击下一步。



6. 查看即将下发的相关配置，点击完成按钮。



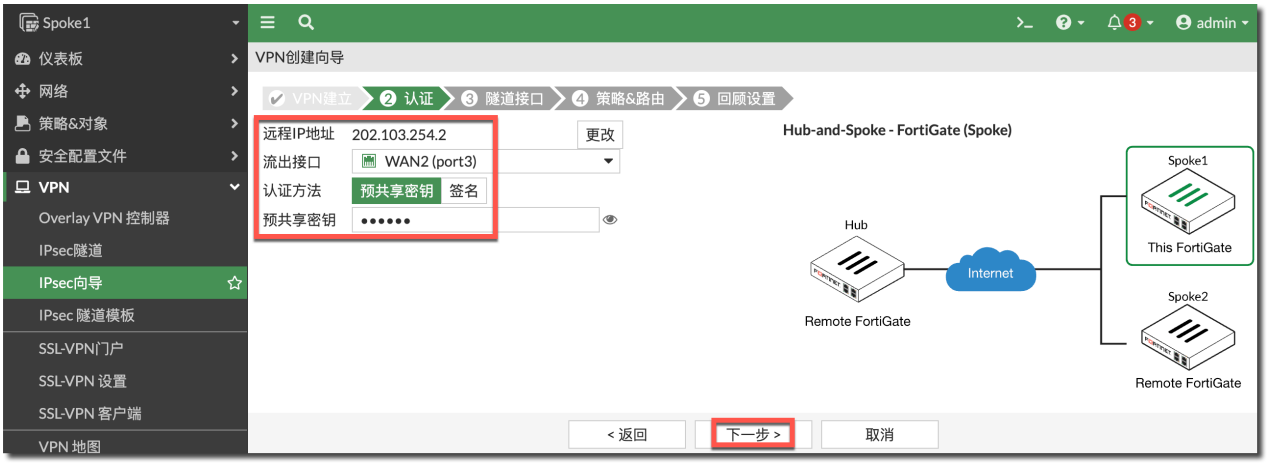
7. 检查相关配置均下发成功后，点击“添加另一个”按钮。



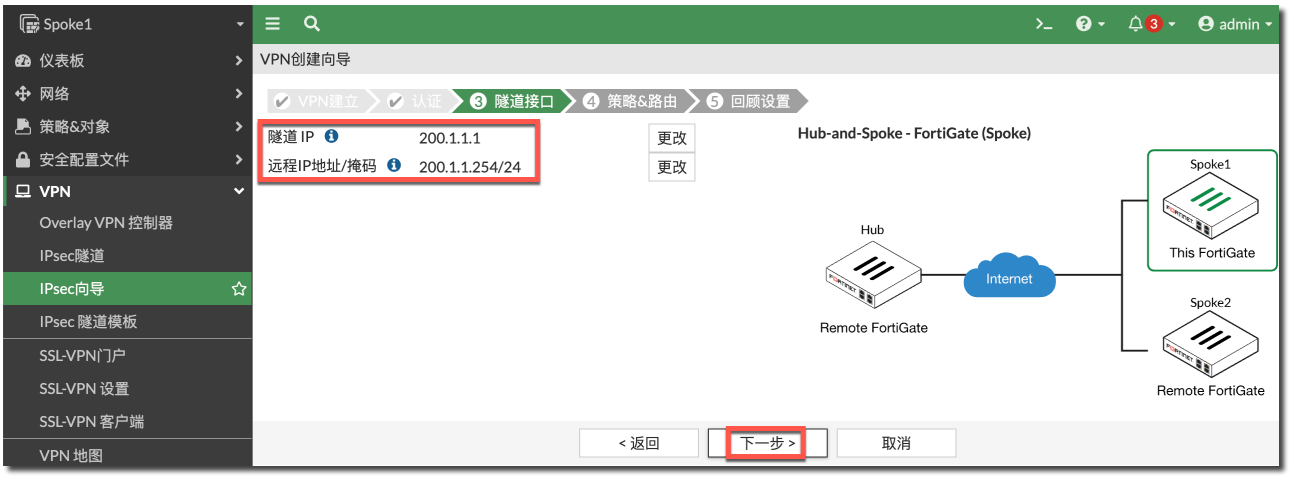
8. 新建 WAN2（port3）上的 IPSec 隧道，选择“Hub-and-Spoke”模式，角色选择 Spoke，将 Spoke1 WAN2 的 IPSec 隧道对应的简易配置密钥粘贴进去，然后点击右侧的应用按钮，会弹出提示“配置已应用”，然后点击下一步。



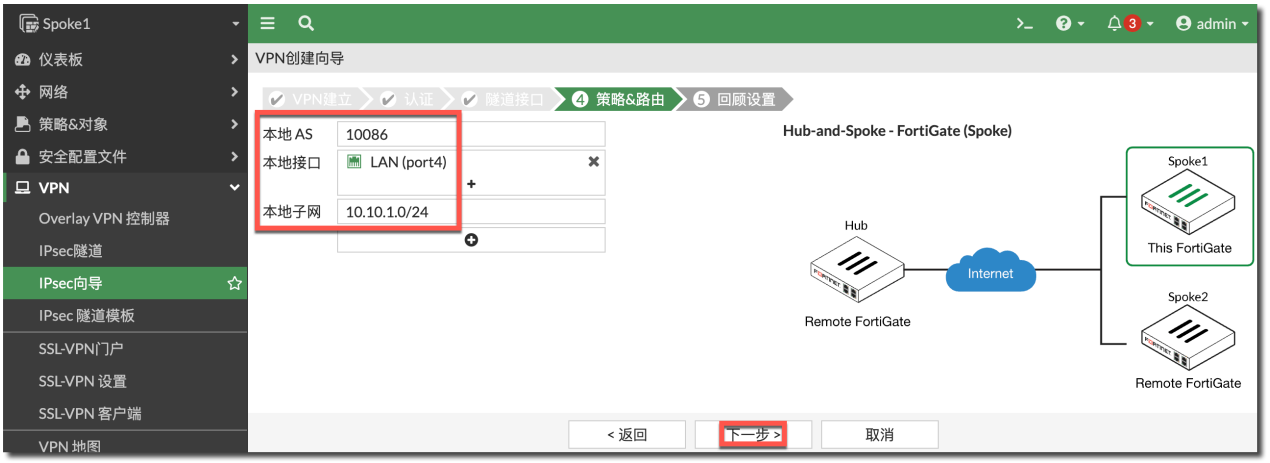
9. Spoke1 自动将 Hub 端 WAN2（port3）的 IP 地址填入，配置流出接口为 WAN2（port3），配置预共享密钥，点击下一步。



0. Spoke1 自动配置了 WAN2（port3）线路 IPSec Tunnel 口的地址（在 Hub 端向导中的配置），远程 IP 地址/掩码为 Hub 端 WAN2 线路 IPSec Tunnel 接口地址。点击下一步。



1. 配置 BGP 本地 AS（此例中 Hub 和所有的 Spoke 之间为 IBGP 邻居），配置本地接口和本地子网（用于自动创建安全策略和 BGP 路由发布）。点击下一步。



2. 查看即将下发的相关配置，点击完成按钮。



3. 检查相关配置均下发成功。



4. 在 Spoke1 上查看向导下发的 IPSec 配置以及 IPSec Tunnel 接口配置。注意要在 IPSec 一阶段中增加下步中的配置。

```bash
config vpn ipsec phase1-interface
    edit "Spoke1_WAN1"
        set interface "port2"
        set peertype any
        set net-device enable    //确保Spoke的net-device开启，否则使用SD-WAN时，Shortcut的健康检查无法正常工作//
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: Spoke1_WAN1 (Created by VPN wizard)"
        set wizard-type spoke-fortigate-auto-discovery
        set auto-discovery-receiver enable    //确认开启了ADVPN receiver//
```

```
        set remote-gw 101.103.254.2
        set psksecret ENC
jZJjdqGe4N3alz21hgoMkyE1t5HDNMSanAQpDEtCwitgjX2hvApP24ZC3axQP4f9qdqT/oeVmj/5bWBXt1drV/rrwPOHteGqFhMmDuRGpHK7qA=
    next
    edit "Spoke1_WAN2"
        set interface "port3"
        set peertype any
        set net-device enable    //确保Spoke的net-device开启，否则使用SD-WAN时，Shortcut的健康检查无法正常工作//
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: Spoke1_WAN2 (Created by VPN wizard)"
        set wizard-type spoke-fortigate-auto-discovery
        set auto-discovery-receiver enable    //确认开启了ADVPN receiver//
        set remote-gw 202.103.254.2
        set psksecret ENC
uH/eLzEpwUxUfKKcFdZYZ9q/bNsDSKCal2iF7LuF978SppZrpdHQi2BpWdTcSlQaDWZ6QaeyeXn+8crQLCavpKv7fa9XhyI+U/OLf7J0LSHc/n=
    next
end

config vpn ipsec phase2-interface
    edit "Spoke1_WAN1"
        set phase1name "Spoke1_WAN1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: Spoke1_WAN1 (Created by VPN wizard)"
    next
    edit "Spoke1_WAN2"
        set phase1name "Spoke1_WAN2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: Spoke1_WAN2 (Created by VPN wizard)"
    next
end

config system interface
    edit "Spoke1_WAN1"
        set vdom "root"
        set ip 100.1.1.1 255.255.255.255
        set type tunnel
        set remote-ip 100.1.1.254 255.255.255.0
        set interface "port2"
    next
    edit "Spoke1_WAN2"
        set vdom "root"
        set ip 200.1.1.1 255.255.255.255
        set type tunnel
        set remote-ip 200.1.1.254 255.255.255.0
        set interface "port3"
    next
end
```

5. 需要在向导配置的基础上，将 IKEv1 主模式修改为 IKEv2，并在每个线路的 IPSec 上开启 network-overlay，配置不同的 network-id，WAN1 上的 IPSec 的 network-id 为 1，WAN2 上的 IPSec 的 network-id 为 2。

> **重要**
>
> 使用 ADVPN 向导创建的 IPSec 为 IKEv1 主模式，Hub 双线路的情况下，Spoke 与 Spoke 在建立 Shortcut 时会出现问题（参考 VPN 技术 → IPSec VPN → 拨号 VPN → 野蛮模式/network-id 适用场景 → ADVPN 下的 network-id 或野蛮模式 ID ↗ 的说明）。

bash

```
config vpn ipsec phase1-interface
    edit "Spoke1_WAN1"
        set ike-version 2              //增加配置：将IKE版本修改为2//
        set network-overlay enable     //增加配置：开启network-overlay//
        set network-id 1               //增加配置：第一条线路上的network-id为1//
    next
    edit "Spoke1_WAN2"
        set ike-version 2              //增加配置：将IKE版本修改为2//
        set network-overlay enable     //增加配置：开启network-overlay//
        set network-id 2               //增加配置：第二条线路上的network-id为2//
    next
end
```

6. 在 SD-WAN（配置了健康检查）+ ADVPN 环境下，以上 IPSec Tunnel 接口还需要手动增加允许 Ping 的配置，用于 SD-WAN 在 Spoke 之间的 Shortcut 隧道自动创建的健康检查。

> **重要**
>
> SD-WAN 在 Shortcut 间自动创建的健康检查为 Ping 模式，即使配置的健去往 Hub 的健康检查为其他模式。

```bash
config system interface
    edit "Spoke1_WAN1"
        append allowaccess ping
    next
    edit "Spoke1_WAN2"
        append allowaccess ping
    next
end
```

7. 查看 Spoke1 上 IPSec 向导下发的地址、地址组和安全策略配置，包含放通 Spoke IPSec 隧道的双向流量的策略。

> **重要**
>
> 后续要在 Spoke 配置 SD-WAN，这些安全策略配置需要删除，否则 IPSec 接口无法加入 SD-WAN 区域。

```bash
config firewall address
    edit "Spoke1_WAN1_local_subnet_1"
        set allow-routing enable
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Spoke1_WAN2_local_subnet_1"
        set allow-routing enable
        set subnet 10.10.1.0 255.255.255.0
    next
end

config firewall addrgrp
    edit "Spoke1_WAN1_local"
        set member "Spoke1_WAN1_local_subnet_1"
        set comment "VPN: Spoke1_WAN1 (Created by VPN wizard)"
        set allow-routing enable
    next
    edit "Spoke1_WAN2_local"
        set member "Spoke1_WAN2_local_subnet_1"
        set comment "VPN: Spoke1_WAN2 (Created by VPN wizard)"
        set allow-routing enable
    next
end

config firewall policy
    edit 1
        set name "vpn_Spoke1_WAN1_remote_0"
        set srcintf "Spoke1_WAN1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Spoke1_WAN1_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Spoke1_WAN1 (Created by VPN wizard)"
    next
    edit 2
        set name "vpn_Spoke1_WAN1_local_0"
        set srcintf "port4"
        set dstintf "Spoke1_WAN1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Spoke1_WAN1 (Created by VPN wizard)"
    next
    edit 3
```

```
        set name "vpn_Spoke1_WAN2_remote_0"
        set srcintf "Spoke1_WAN2"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Spoke1_WAN2_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Spoke1_WAN2 (Created by VPN wizard)"
    next
    edit 4
        set name "vpn_Spoke1_WAN2_local_0"
        set srcintf "port4"
        set dstintf "Spoke1_WAN2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Spoke1_WAN2 (Created by VPN wizard)"
    next
end
```

8. 查看 Spoke1 上 IPSec 向导下发的 BGP 配置。

```bash
config router bgp
    set as 10086
    config neighbor
        edit "100.1.1.254"
            set remote-as 10086
        next
        edit "200.1.1.254"
            set remote-as 10086
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

9. 在 ADVPN 环境下，以上 BGP 配置还需要手动增加如下参数。

```bash
config router bgp
    set ibgp-multipath enable    //必须配置，否则去往Hub或其他Spoke的路由无法负载//
    config neighbor
        edit "100.1.1.254"
            set additional-path receive    //多线路ADVPN，Spoke端必须配置，否则Spoke学到的路由可能不全//
            set soft-reconfiguration enable
        next
        edit "200.1.1.254"
            set additional-path receive    //多线路ADVPN，Spoke端必须配置，否则Spoke学到的路由可能不全//
            set soft-reconfiguration enable
        next
    end
end
```

10. Spoke2 与 Spoke3 的配置参考 Spoke1，这里就不再赘述（注意 Spoke3 只有一条 Internet 线路，与 Hub 的两条线路建立 IPSec）。

SD-WAN 基础配置

Spoke

1. 在 Spoke1 上删除 IPSec 向导自动创建的安全策略（否则 IPSec 接口无法加入 SD-WAN 区域）。

2. 进入分支 FortiGate 的网络 → SD-WAN → SD-WAN 区域页面，新建用于 ADVPN 访问的区域“Spoke1_ADVPN”。



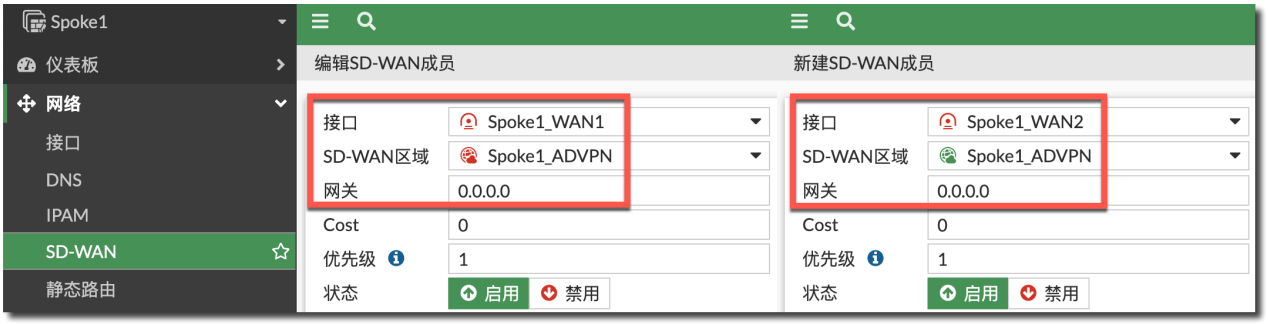


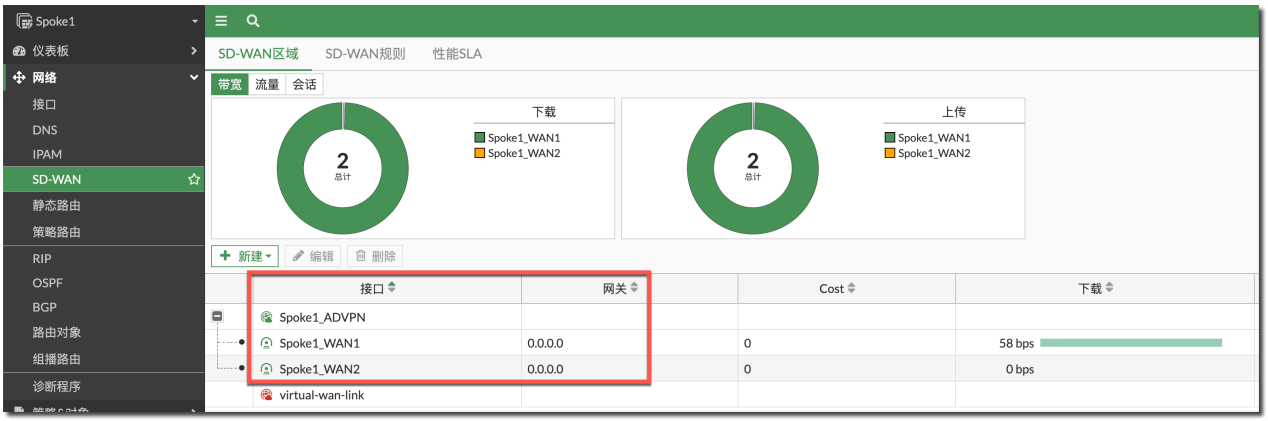

```bash
config system sdwan
    set status enable
    config zone
        edit "Spoke1_ADVPN"
        next
    end
end
```

3. 新建用于访问 Hub 内网和其他 Spoke 的 SD-WAN 成员（两个 IPSec Tunnel 接口）加入步骤 2 创建的 SD-WAN 区域“Spoke1_ADVPN”。注意 IPSec Tunnel 接口加入 SD-WAN 区域时不需要填写网关。

> **重要**
>
> 如果 IPSec 口存在相关配置引用的情况下，是无法直接加入到 SD-WAN 接口成员的，可参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域↗章节快速将接口加入 SD-WAN 区域，并自动修改关联配置（也可选择手动将 WAN 接口的相关配置删除引用，包括策略、Link-monitor、所属区域等）。

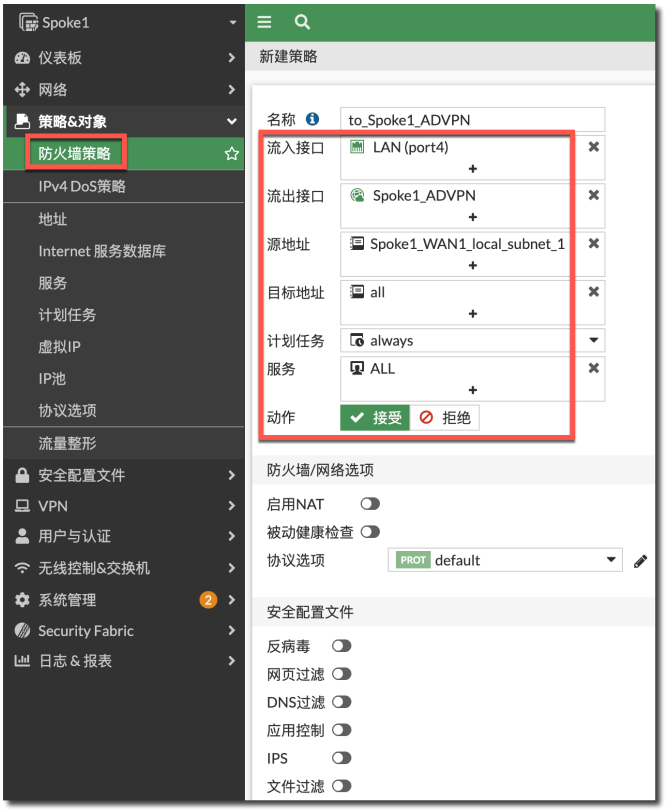







```bash
config system sdwan
    config members
        edit 1
            set interface "Spoke1_WAN1"
            set zone "Spoke1_ADVPN"
        next
        edit 2
            set interface "Spoke1_WAN2"
            set zone "Spoke1_ADVPN"
        next
    end
end
```

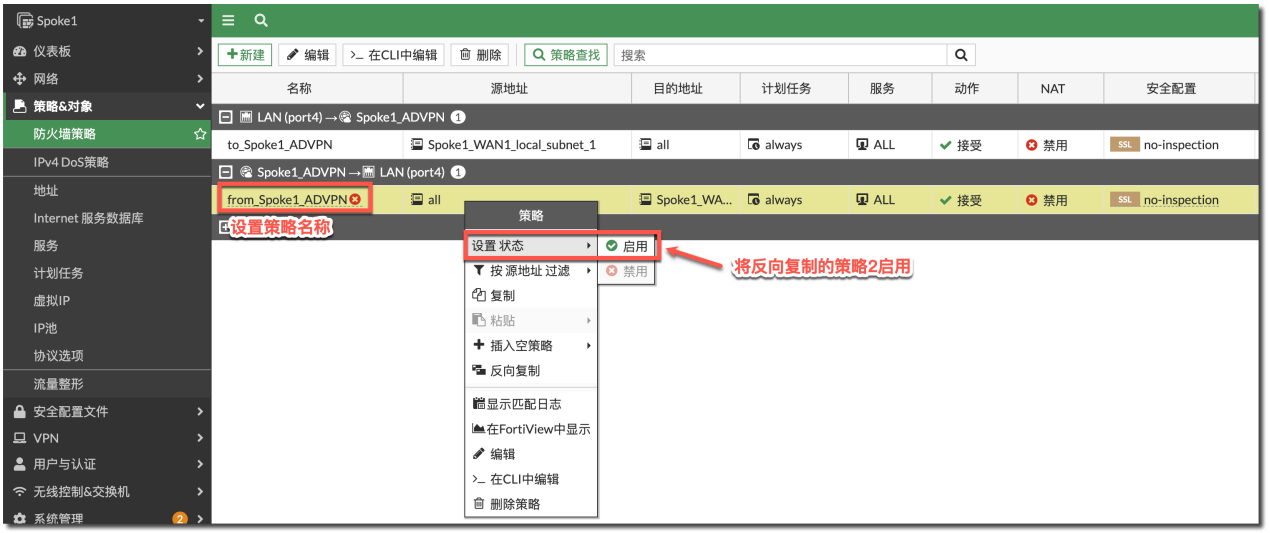
4. 配置防火墙策略 1，放通内网 port4（LAN）到 SD-WAN ADVPN 区域的流量，通常情况下内网之间的 VPN 访问不需要开启 SNAT。根据需求开启 UTM 功能。




```bash
config firewall policy
    edit 1
        set name "to_Spoke1_ADVPN"
        set srcintf "port4"
        set dstintf "Spoke1_ADVPN"
        set action accept
        set srcaddr "Spoke1_WAN1_local_subnet_1"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

5. 右键点击防火墙策略 1，选择反向复制，生成策略 2。放通 SD-WAN ADVPN 区域到内网 port4（LAN）的流量，通常情况下内网之间的 VPN 访问不需要开启 SNAT。根据需求开启 UTM 功能。反向复制的策略名称为空，且为禁用状态，需要手动设置策略名称，然后手动启用该策略。






```bash
config firewall policy
    edit 2
        set name "from_Spoke1_ADVPN"
        set srcintf "Spoke1_ADVPN"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Spoke1_WAN1_local_subnet_1"
        set schedule "always"
        set service "ALL"
        set comments " (Copy of to_Spoke1_ADVPN) (Reverse of to_Spoke1_ADVPN)"
    next
end
```



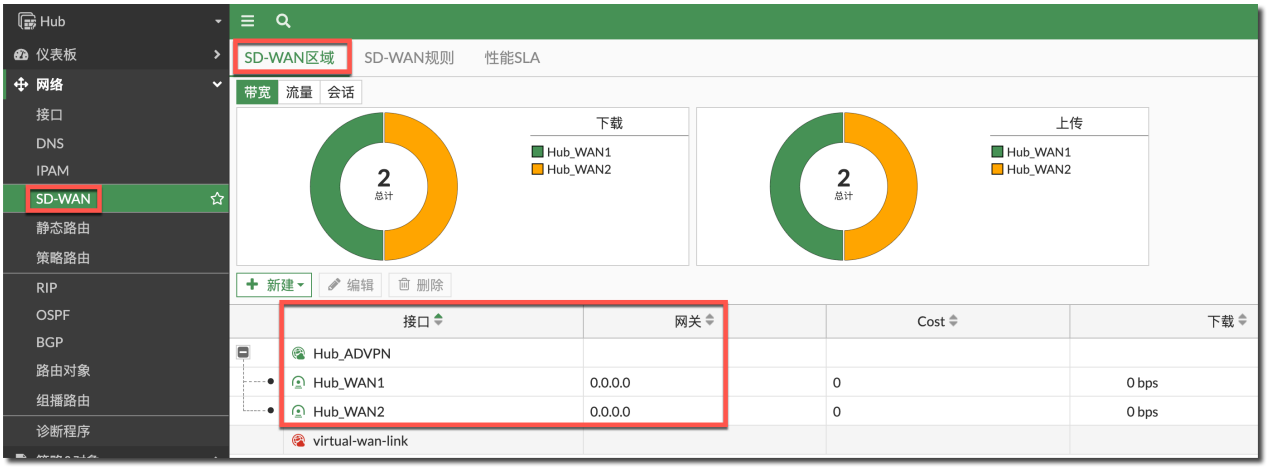
6. Spoke2 和 Spoke3 的配置参考 Spoke1，这里就不再赘述。

Hub

> **重要**
>
> Hub 可以不配置 SD-WAN。如果不配置 SD-WAN 选路，可以通过静态路由不同的优先级进行选路。这里我们以 Hub 端配置 SD-WAN 为例。

1. 删除 IPSec 向导生成的所有安全策略（否则 IPSec 接口无法加入 SD-WAN 区域）。
2. 在 Hub 上新建 SD-WAN 区域“Hub_ADVPN”与 SD-WAN 成员（包含 WAN1 和 WAN2 上的 IPSec 拨号连接），将两个 IPSec 接口加入 SD-WAN 区域。

> **重要**
>
> 如果 IPSec 口存在相关配置引用的情况下，是无法直接加入到 SD-WAN 接口成员的，可参考网络管理 → SD-WAN → SD-WAN 配置举例 → 将已关联接口移至 SD-WAN 区域 ↗章节快速将接口加入 SD-WAN 区域，并自动修改关联配置（也可选择手动将 WAN 接口的相关配置删除引用，包括策略、Link-monitor、所属区域等）。




```bash
config system sdwan
    set status enable
    config zone
        edit "Hub_ADVPN"
        next
    end
    config members
        edit 1
            set interface "Hub_WAN1"
            set zone "Hub_ADVPN"
        next
        edit 2
            set interface "Hub_WAN2"
            set zone "Hub_ADVPN"
        next
    end
end
```

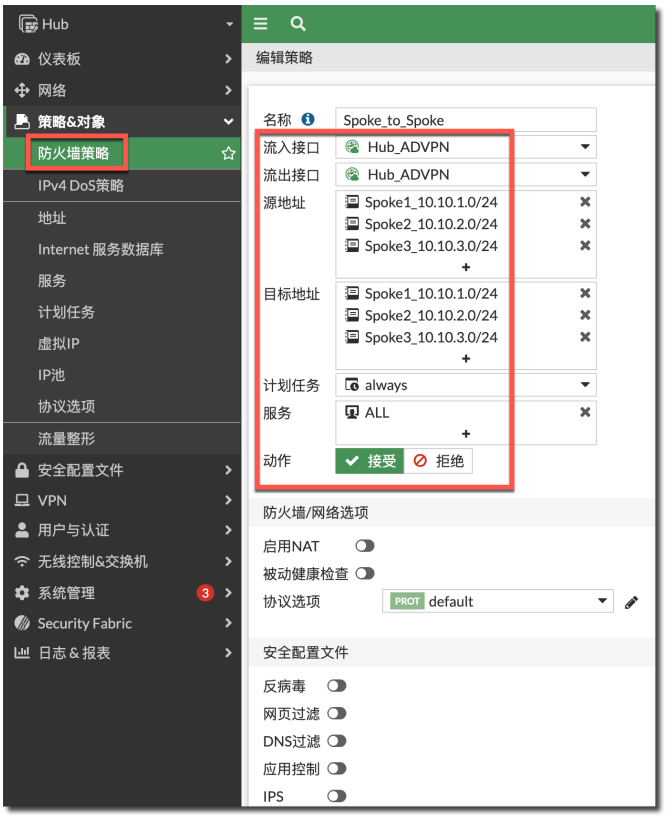
3. 配置防火墙策略 1 和防火墙策略 2，放通内网 port4（LAN）与所有 Spoke 内网之间互相访问的流量，通常情况下内网之间的 VPN 访问不需要开启 SNAT。根据需求开启 UTM 功能。



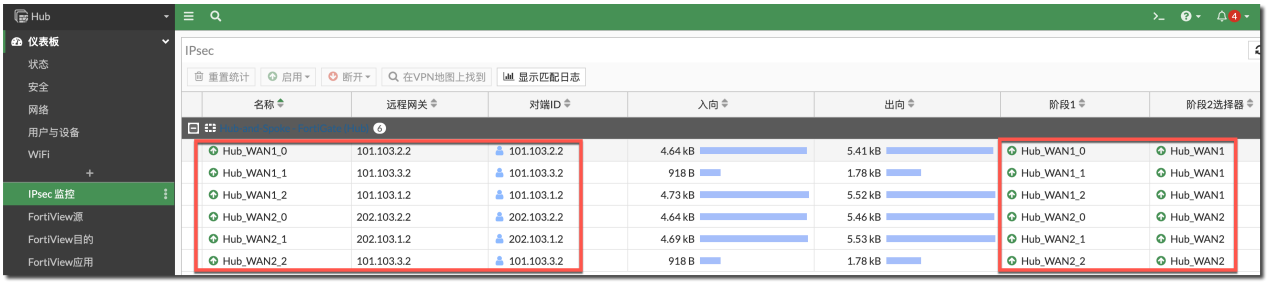

```bash
config firewall policy
    edit 1
        set name "to_Hub_ADVPN"
        set srcintf "port4"
        set dstintf "Hub_ADVPN"
        set action accept
        set srcaddr "Hub_WAN1_local_subnet_1"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "from_Hub_ADVPN"
        set srcintf "Hub_ADVPN"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "Hub_WAN1_local_subnet_1"
```

```
        set schedule "always"
        set service "ALL"
        set comments " (Copy of to_Hub_ADVPN) (Reverse of to_Hub_ADVPN)"
    next
end
```

4. 配置防火墙策略 3，放通通过 Hub 中转的 Spoke 之间的流量。通常情况下 VPN 之间的 VPN 访问不需要开启 SNAT。根据需求开启 UTM 功能。




```
config firewall policy                                                
```




```bash
Hub # get vpn ipsec tunnel summary
'Hub_WAN1_0' 101.103.2.2:0  selectors(total,up): 1/1  rx(pkt,err): 84/0  tx(pkt,err): 83/0
'Hub_WAN1_1' 101.103.3.2:0  selectors(total,up): 1/1  rx(pkt,err): 20/0  tx(pkt,err): 27/0
'Hub_WAN1_2' 101.103.1.2:0  selectors(total,up): 1/1  rx(pkt,err): 84/0  tx(pkt,err): 83/0
'Hub_WAN2_0' 202.103.2.2:0  selectors(total,up): 1/1  rx(pkt,err): 84/0  tx(pkt,err): 83/1
'Hub_WAN2_1' 202.103.1.2:0  selectors(total,up): 1/1  rx(pkt,err): 84/0  tx(pkt,err): 83/1
'Hub_WAN2_2' 101.103.3.2:0  selectors(total,up): 1/1  rx(pkt,err): 20/0  tx(pkt,err): 27/0
```

2. 查看 Hub 端 FortiGate 路由表中的 BGP 路由，Hub 去往每个 Spoke 的路由在两条线路的 IPSec 隧道上负载。

```bash
Hub # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 100.1.1.1 (recursive is directly connected, Hub_WAN1), 00:28:29, [1/0]
                     [200/0] via 200.1.1.1 (recursive is directly connected, Hub_WAN2), 00:28:29, [1/0]
B       10.10.2.0/24 [200/0] via 100.1.1.2 (recursive is directly connected, Hub_WAN1), 00:28:32, [1/0]
                     [200/0] via 200.1.1.2 (recursive is directly connected, Hub_WAN2), 00:28:32, [1/0]
B       10.10.3.0/24 [200/0] via 100.1.1.3 (recursive is directly connected, Hub_WAN1), 00:05:24, [1/0]
                     [200/0] via 200.1.1.3 (recursive is directly connected, Hub_WAN2), 00:05:24, [1/0]
```

3. 查看 Spoke1 FortiGate 路由表中的 BGP 路由：

- 去往 Hub 内网的路由在两个线路的 IPSec 隧道上负载；
- 去往其他 Spoke 内网的路由通过 Hub 端的 BGP 路由反射器反射后学习，通过两个 IPSec 隧道负载，出接口为去往 Hub 的 IPSec 隧道接口（由于 Hub 和 Spoke 开启了 BGP additional-path send/receive，且 Hub 端 BGP 在两个 IPSec 隧道上都开启了路由反射，Spoke1 每条隧道学习另一个 Spoke 内网的路由会有 2 条相同的路由）。

```bash
Spoke1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 100.1.1.2 [2] (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 00:29:51, [1/0]
                     [200/0] via 200.1.1.2 [2] (recursive via Spoke1_WAN2 tunnel 202.103.254.2), 00:29:51, [1/0]
B       10.10.3.0/24 [200/0] via 100.1.1.3 [2] (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 00:06:55, [1/0]
                     [200/0] via 200.1.1.3 [2] (recursive via Spoke1_WAN2 tunnel 202.103.254.2), 00:06:55, [1/0]
B       10.10.254.0/24 [200/0] via 100.1.1.254 (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 00:30:21, [1/0]
                       [200/0] via 200.1.1.254 (recursive via Spoke1_WAN2 tunnel 202.103.254.2), 00:30:21, [1/0]
```

4. Spoke2、Spoke3 的路由表与 Spoke1 类似，这里不再赘述。

SD-WAN 策略配置

Spoke

需求 1：Spoke 去往 Hub 内网的流量优先从 WAN1（port2）的 IPSec 隧道转发，WAN1 的 IPSec 隧道中断或不满足 SLA 标准后，从 WAN2（port3）的 IPSec 隧道转发。

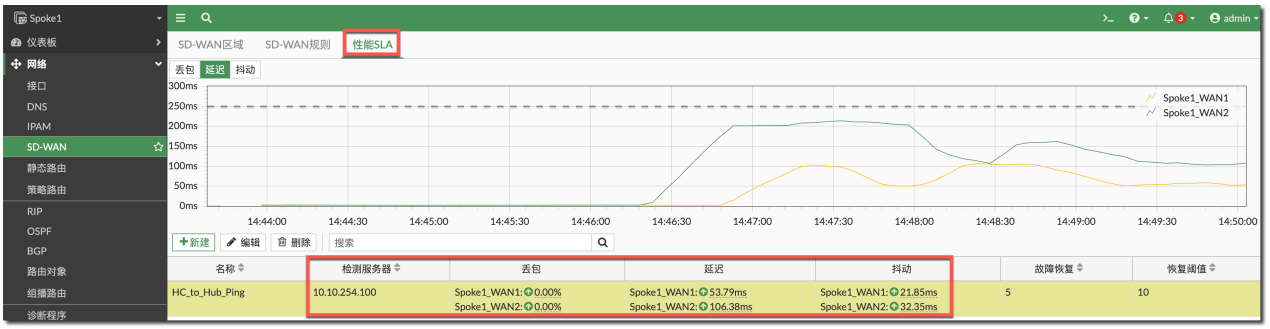
1. 在 Spoke1 的 SD-WAN 中配置探测 Hub 端内网服务器的健康检查：

- 成员选择 WAN1 的 IPSec Tunnel 接口和 WAN2 的 IPSec Tunnel 接口。
- 探测模式为 Ping。
- 开启 SLA 目标：延迟 < 250ms/抖动 < 50ms/丢包率 < 5%。




```bash
config system sdwan
    config health-check
        edit "HC_to_Hub_Ping"
            set server "10.10.254.100"
            set interval 1000
            set recoverytime 10
            set members 0
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
end
```

2. 查看健康检查的状态，可以看到目前 2 条去往 Hub 内网的线路都满足 SLA 标准。

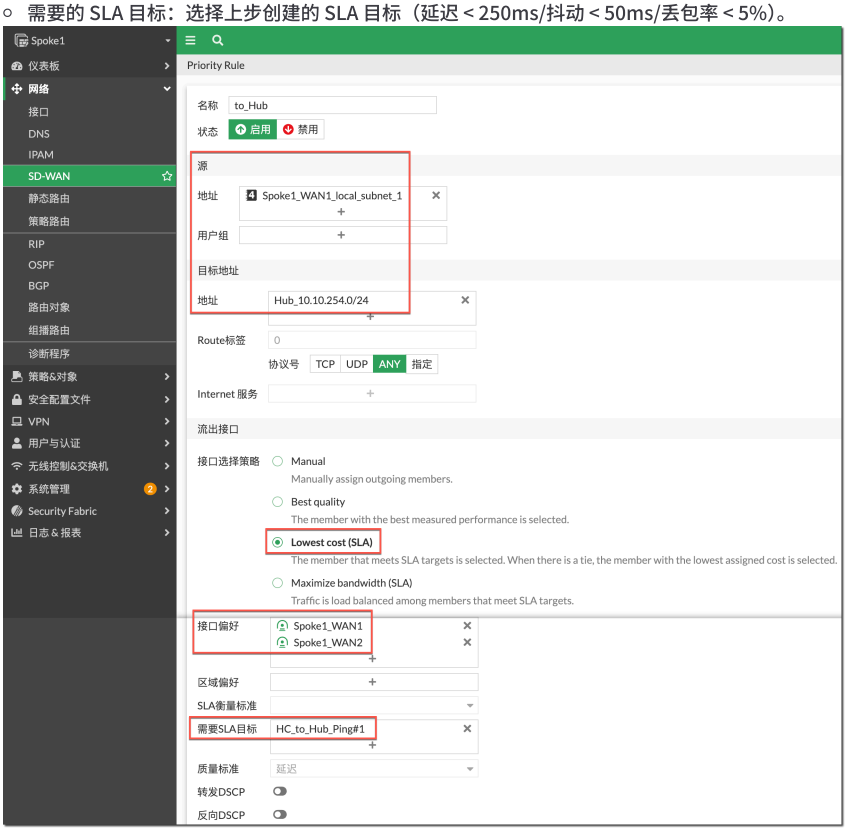



```bash
Spoke1 # diagnose sys sdwan health-check status HC_to_Hub_Ping
Health Check(HC_to_Hub_Ping):
Seq(1 Spoke1_WAN1): state(alive), packet-loss(0.000%) latency(48.060), jitter(20.974), mos(4.356), bandwidth-
up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x1
Seq(2 Spoke1_WAN2): state(alive), packet-loss(0.000%) latency(106.973), jitter(30.848), mos(4.268), bandwidth-
up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x1
```

3. 新建 SD-WAN 规则 1，用于 Spoke 内网去往 Hub 内网的流量：



- 源地址选择 Spoke1 内网网段；目标地址为 Hub 内网网段。
- 接口选择策略：选择 Lowest Cost (SLA)，Spoke 内部访问 Hub 内网的流量优先从满足 SLA 标准（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）的线路转发。如果 2 条线路都满足 SLA 标准，则优先从接口偏好中靠前的接口转发。
- 接口偏好：选择 WAN1 线路 IPSec 接口与 WAN2 线路 IPSec 接口。WAN1 线路 IPSec 接口排序在前，其次是 WAN2 线路 IPSec 接口。
- 需要的 SLA 目标：选择上步创建的 SLA 目标（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）。




```bash
config system sdwan
    config service
        edit 1
            set name "to_Hub"
            set mode sla
            set dst "Hub_10.10.254.0/24"
            set src "Spoke1_WAN1_local_subnet_1"
            config sla
                edit "HC_to_Hub_Ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

4. 查看上步创建的 SD-WAN 规则目前的选路状态，由于目前 2 条去往 Hub 内网的线路均满足 SLA 目标，所以选择接口偏好配置中靠前的 WAN1 上的 IPSec 线路。




```bash
Spoke1 #  diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 Spoke1_WAN1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected    //优先从WAN1上的IPSec隧道转发//
    2: Seq_num(2 Spoke1_WAN2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
  Src address(1):
        10.10.1.0-10.10.1.255

  Dst address(1):
        10.10.254.0-10.10.254.255
```

5. 查看 SD-WAN 规则对应的策略路由状态，出接口优先选择 MPLS 线路接口（port4）。

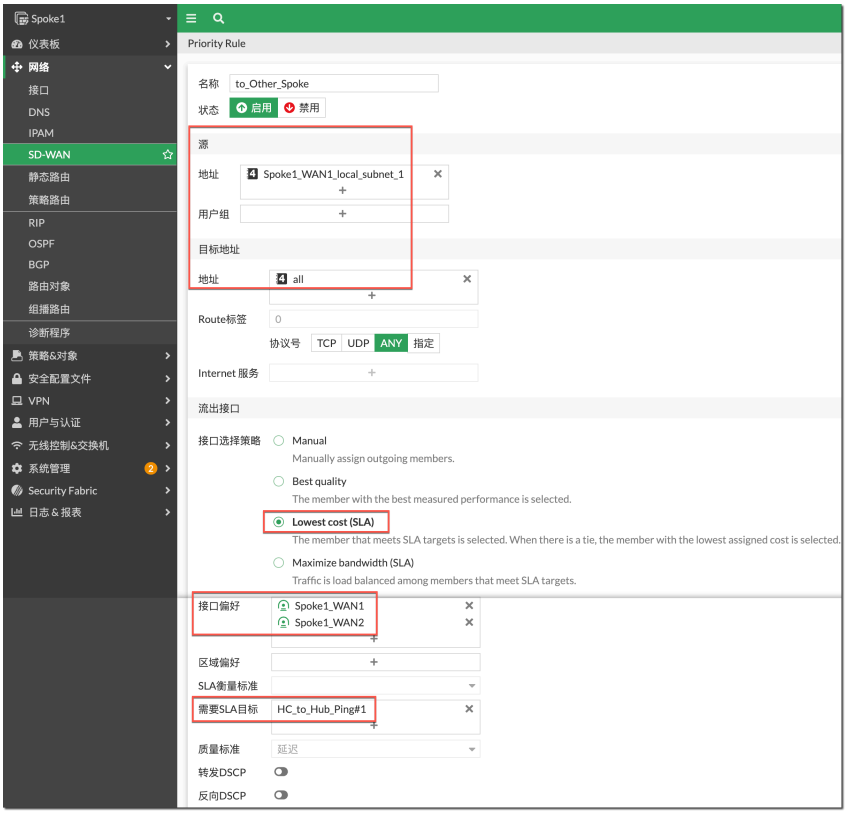
```bash
Spoke1 # diagnose firewall proute list
list route policy info(vf=root):

id=2130837505(0x7f020001) vwl_service=1(to_Hub) vwl_mbr_seq=1 2 dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(2) oif=17(Spoke1_WAN1) oif=18(Spoke1_WAN2)    //优先从WAN1上的IPSec隧道转发//
source(1): 10.10.1.0-10.10.1.255
destination(1): 10.10.254.0-10.10.254.255
hit_count=0 last_used=2023-12-26 15:03:45
```

6. Spoke2 与 Spoke3 的配置请参考 Spoke1，这里不再赘述。

需求 2：Spoke 内网访问其他 Spoke 内网的流量优先从 Shortcut 隧道转发。Shortcut 隧道无法建立时，通过 Hub 中转。

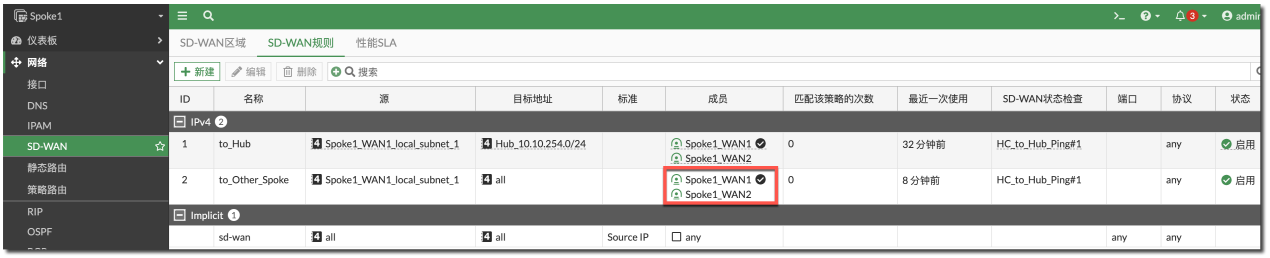
1. 新建 SD-WAN 规则 2：

- 源地址选择 Spoke1 内网网段；目标地址为 all（或其他 Spoke 内网的网段集合）。
- 接口选择策略：选择 Lowest Cost (SLA)，Spoke 内部访问 Internet 的流量优先从满足 SLA 标准（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）的 IPSec 线路创建的 Shortcut 隧道转发（如果 Shortcut 无法建立，则通过去往 Hub 的 IPSec 隧道转发）。如果两条 IPSec 线路都满足 SLA 标准，则优先从接口偏好中靠前的 IPSec 隧道创建的 Shortcut 转发（如果 Shortcut 无法建立，则通过去往 Hub 的 IPSec 隧道转发）。
- 接口偏好：选择 WAN1 线路 IPSec 接口与 WAN2 线路 IPSec 接口。WAN1 线路 IPSec 接口排序在前，其次是 WAN2 线路 IPSec 接口。
- 需要 SLA 目标：选择上步创建的 SLA 目标（延迟 < 250ms/抖动 < 50ms/丢包率 < 5%）。




```bash
config system sdwan
    config service
        edit 2
            set name "to_Other_Spoke"
            set mode sla
            set dst "all"
            set src "Spoke1_WAN1_local_subnet_1"
            config sla
                edit "HC_to_Hub_Ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. 查看上步创建的 SD-WAN 规则目前的选路状态，由于目前两条 IPSec 线路均满足 SLA 目标，所以选择接口偏好配置中靠前的 Spoke1_WAN1 隧道。




```bash
Spoke1 # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 Spoke1_WAN1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected    //优先从WAN1上的IPSec隧道转发//
    2: Seq_num(2 Spoke1_WAN2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
```

```
Src address(1):
      10.10.1.0-10.10.1.255

Dst address(1):
      0.0.0.0-255.255.255.255
```

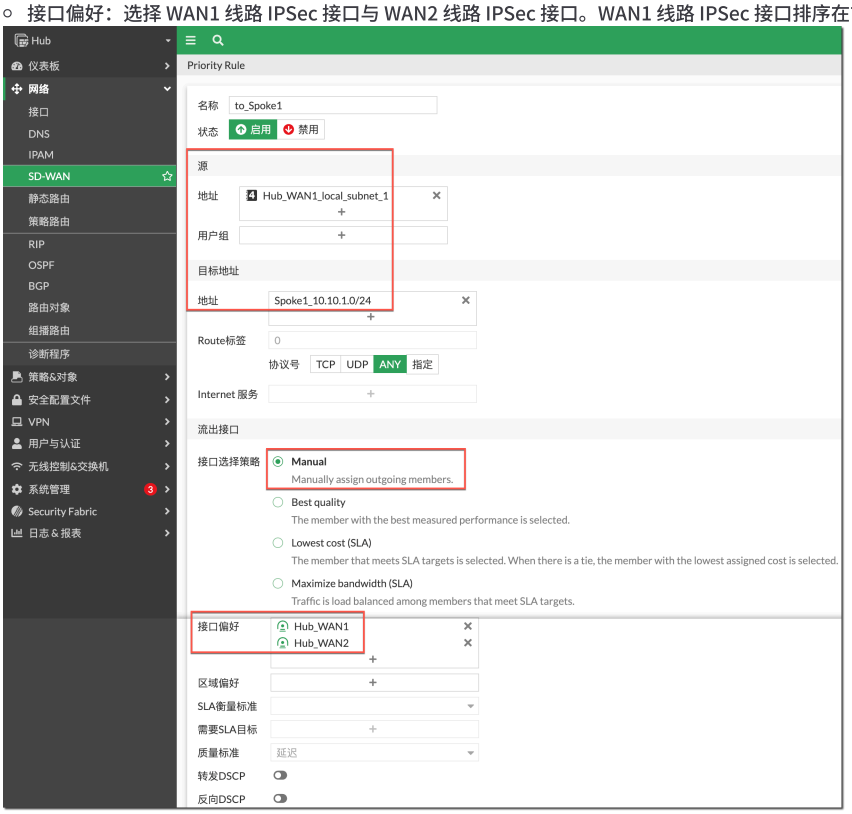
3. Spoke2 与 Spoke3 的配置请参考 Spoke1，这里不再赘述。

Hub

需求 3：Hub 内网主动访问 Spoke 内网的流量强制从 WAN1 的 IPSec 线路转发，当 WAN1 的 IPSec 线路中断时，从 WAN2 的 IPSec 线路转发。

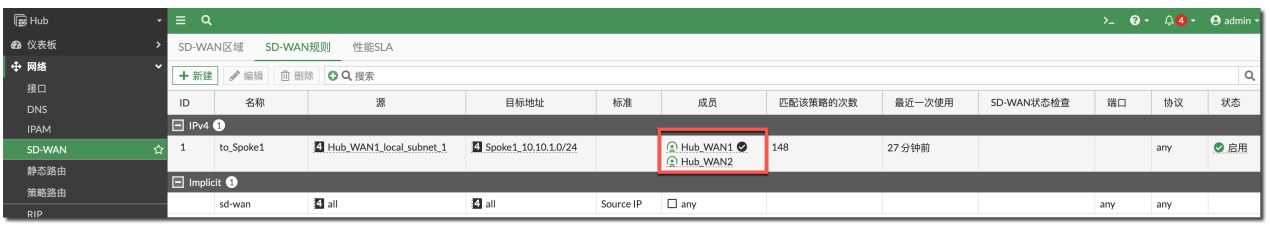
1. 新建 SD-WAN 规则 1（可根据需求使用其他模式）：

   - 源地址选择 Hub 内网网段；目标地址为 Spoke1 内网网段。
   - 接口选择策略：选择 Manual，Hub 内网访问 Spoke1 的流量优先从 WAN1 的 IPSec 线路转发，当 WAN1 的 IPSec 线路中断时，从 WAN2 的 IPSec 线路转发。
   - 接口偏好：选择 WAN1 线路 IPSec 接口与 WAN2 线路 IPSec 接口。WAN1 线路 IPSec 接口排序在前，其次是 WAN2 线路 IPSec 接口。




```bash
config system sdwan
    config service
        edit 1
            set name "to_Spoke1"
            set dst "Spoke1_10.10.1.0/24"
            set src "Hub_WAN1_local_subnet_1"
            set priority-members 1 2
        next
    end
end
```

2. 查看上步创建的 SD-WAN 规则目前的选路状态，选择接口偏好配置中靠前的 Spoke1_WAN1 隧道。



```
Hub # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
    1: Seq_num(1 Hub_WAN1), alive, selected    //Hub访问Spoke1的流量优先从WAN1的IPSec隧道转发//
    2: Seq_num(2 Hub_WAN2), alive, selected
  Src address(1):
        10.10.254.0-10.10.254.255

  Dst address(1):
        10.10.1.0-10.10.1.255
```

3. 去往 Spoke2 和 Spoke3 的 SD-WAN 规则可参考步骤 1，这里不再赘述。

需求 4：当 Spoke 内网之间的 Shortcut 隧道无法建立，或 Shortcut 还未建立 Spoke 内网之间第一次进行访问时，流量要通过 Hub 进行中转。Hub 中转这类流量时，需要保证流入 IPSec 隧道和流出 IPSec 隧道为同一个隧道。

1. 通过 CLI 新建 SD-WAN 规则 2：

- 源地址（src）选择和目标地址（dst）都选择所有 Spoke 的内网网段地址。
- 接口选择策略（mode）：使用 Manual。
- 接口偏好（priority-members）：选择 WAN1 的 IPSec 隧道和 WAN2 的 IPSec 隧道。
- tie-break：选择 input-device 方式，根据入接口来决定 SD-WAN 策略转发时使用相同的接口作为出接口，接口偏好中配置的接口顺序不会生效（该选项需要 7.2.1 及以上版本支持，如低于 7.2.1 版本，请使用策略路由实现此需求）。

```
config system sdwan
    config service
        edit 2
            set name "Spoke_to_Spoke"
            set mode manual
            set dst "Spoke1_10.10.1.0/24" "Spoke2_10.10.2.0/24" "Spoke3_10.10.3.0/24"
            set src "Spoke1_10.10.1.0/24" "Spoke2_10.10.2.0/24" "Spoke3_10.10.3.0/24"
            set priority-members 1 2
            set tie-break input-device    //重要配置//
        next
    end
end
```

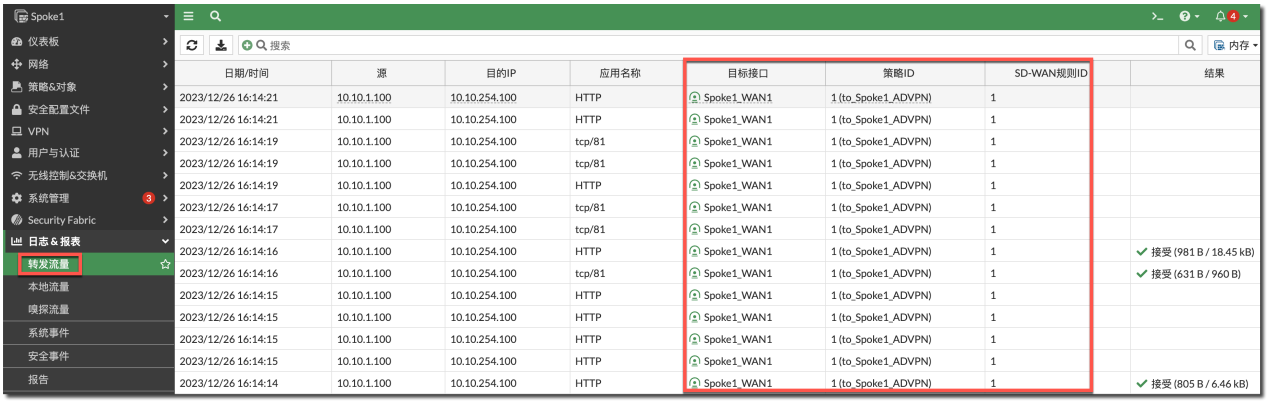
2. 查看上步创建的 SD-WAN 规则目前的状态，Tie break 模式为 input-device，下方的 Member 顺序不会生效。

```
Hub # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: input-device    //tie-break模式为input-device时，下方的Member顺序不会生效//
  Gen(1), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(manual)
  Members(2):
    1: Seq_num(1 Hub_WAN1), alive, selected
    2: Seq_num(2 Hub_WAN2), alive, selected
  Src address(3):
        10.10.3.0-10.10.3.255
        10.10.2.0-10.10.2.255
        10.10.1.0-10.10.1.255

  Dst address(3):
        10.10.3.0-10.10.3.255
        10.10.2.0-10.10.2.255
        10.10.1.0-10.10.1.255
```

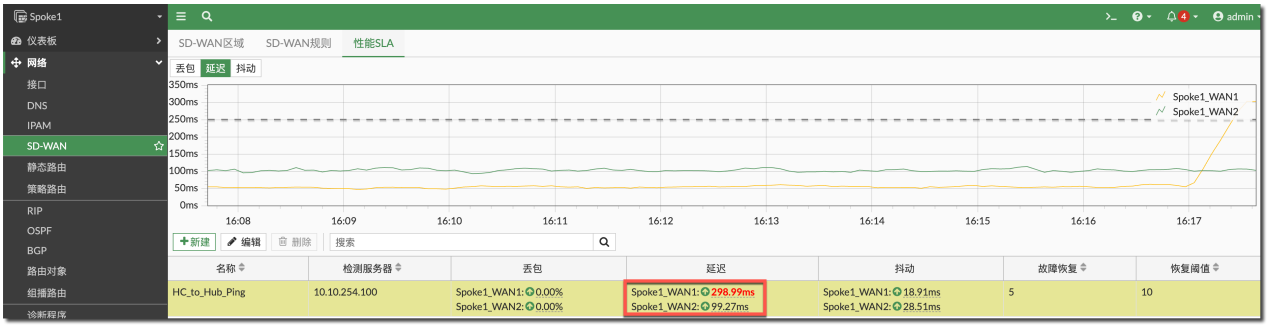
结果验证

Spoke 访问 Hub

1. 使用 Spoke1 内网 PC 客户端访问 Hub 端的内网业务，流量可以正确从 WAN1 上的 IPSec 隧道转发。

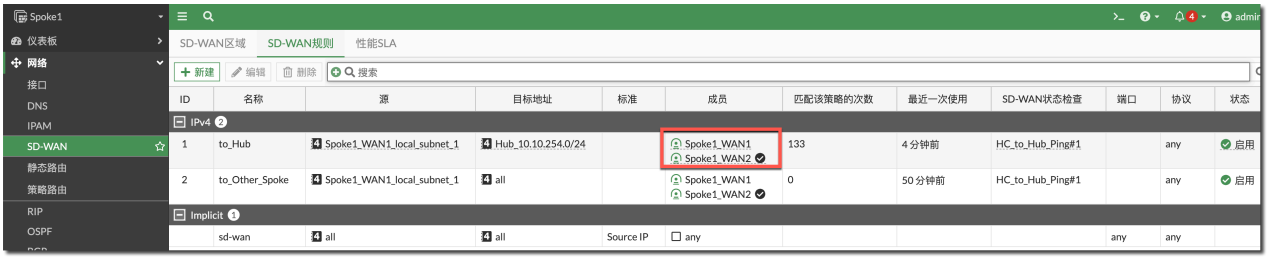


2. 模拟 Spoke1 WAN1 上 IPSec 隧道的健康检查的延迟增加到 300ms。



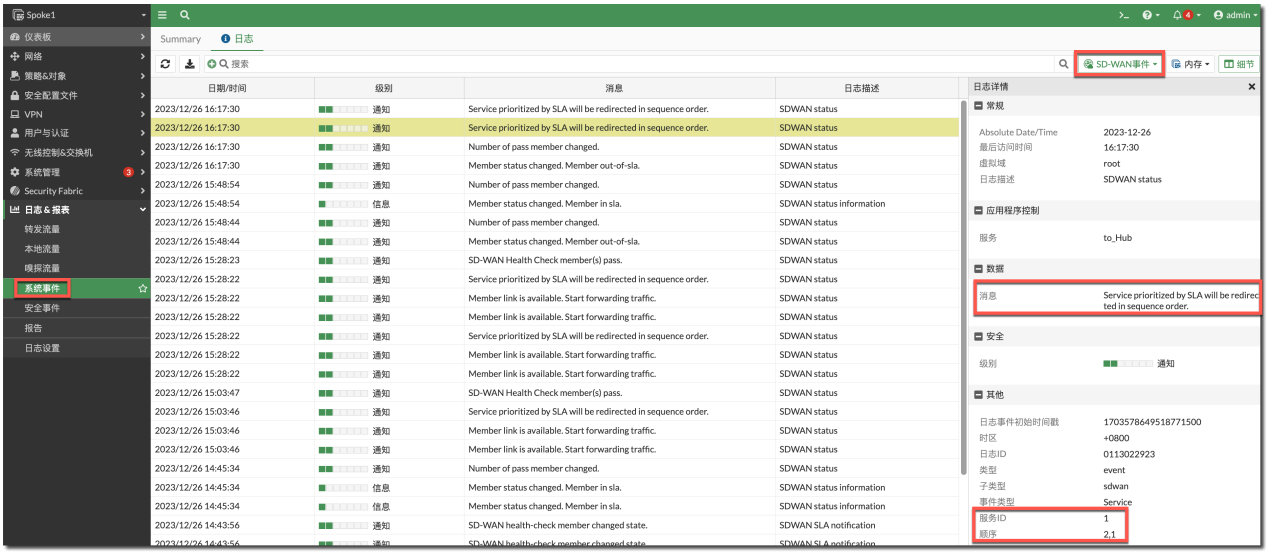

```bash
Spoke1 # diagnose sys sdwan health-check status HC_to_Hub_Ping
Health Check(HC_to_Hub_Ping):
Seq(1 Spoke1_WAN1): state(alive), packet-loss(0.000%) latency(303.576), jitter(19.449), mos(3.595), bandwidth-up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x0
Seq(2 Spoke1_WAN2): state(alive), packet-loss(0.000%) latency(103.569), jitter(34.299), mos(4.258), bandwidth-up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x1
```

3. 在 Spoke1 上查看 SD-WAN 规则 1 的选路优先选择联通线路 IPSec 接口去往 Hub 内网。



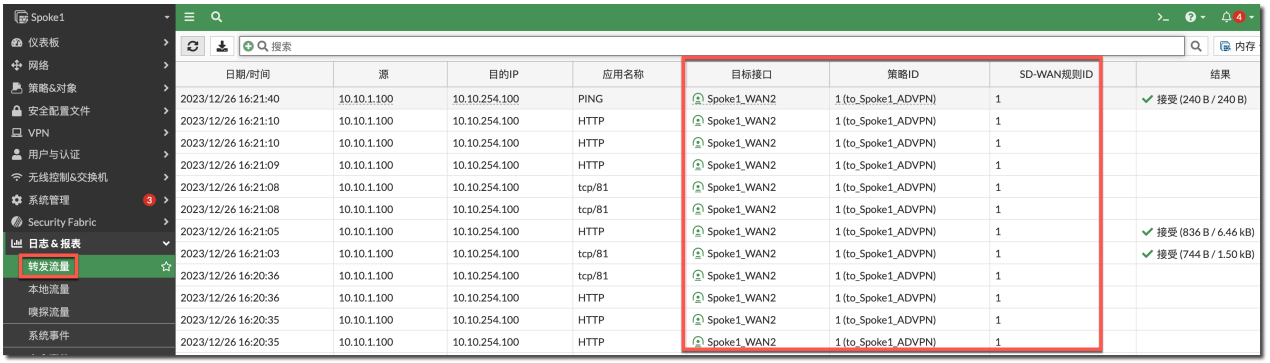

```bash
Spoke1 # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(2 Spoke1_WAN2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected    //去往Hub内网的流量优先从WAN2的IPSec隧道转发//
    2: Seq_num(1 Spoke1_WAN1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.10.1.0-10.10.1.255

  Dst address(1):
        10.10.254.0-10.10.254.255
```

4. 对应的 SD-WAN 线路切换系统日志。




```
date=2023-12-26 time=16:17:30 eventtime=1703578649518771555 tz="+0800" logid="0113022923" type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Service" serviceid=1 service="to_Hub" seq="2,1" msg="Service prioritized by SLA will be redirected in sequence order."
```
bash

5. 再次使用 Spoke1 内网 PC 客户端访问 Hub 端内网的业务，流量开始从 WAN2 的 IPSec 接口转发。



6. Spoke2、Spoke3 内网访问 Hub 内网的测试参考 Spoke1，这里不再赘述。

Spoke 访问 Spoke

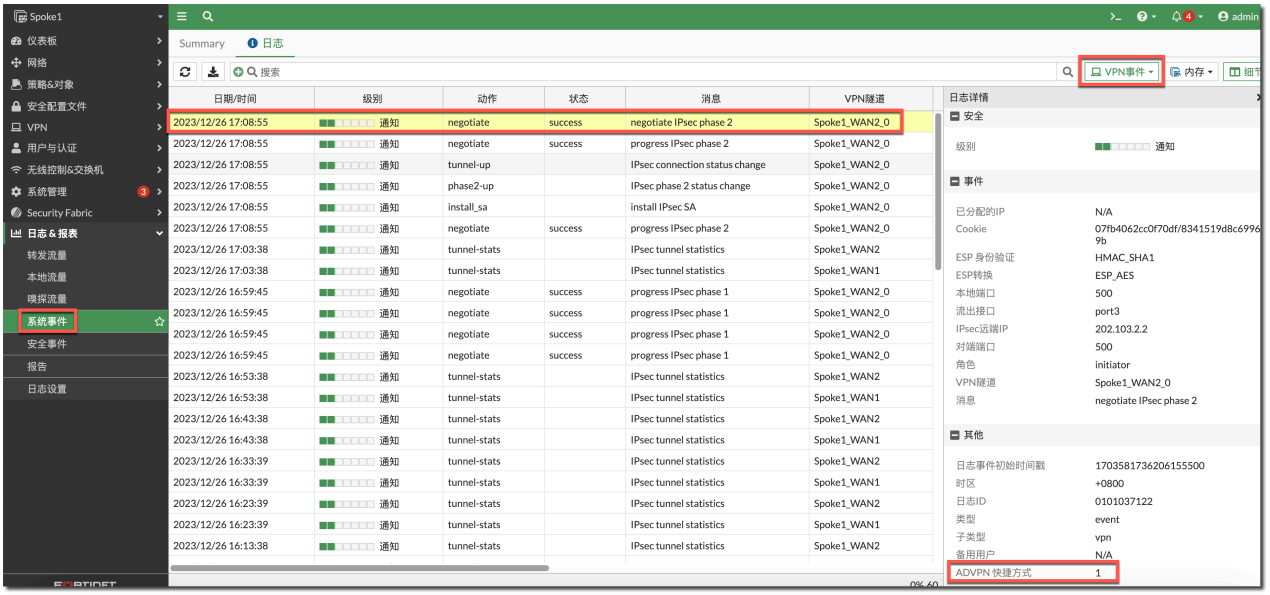
1. Spoke 与 Spoke 之间互访流量会触发 Shortcut 隧道的建立（前提是需要 Hub 开启 BGP RR，在 Spoke 之间反射路由，且 Hub 开启 ADVPN sender，Spoke 开启 ADVPN receiver），在创建 Shortcut 隧道之前，第一次 Spoke 访问其他 Spoke 的流量会经过 Hub 进行中转，同时通过 Hub 进行相关 Shortcut 信息专递。Shortcut 隧道建立后，Spoke 与 Spoke 之间直接通过 Shortcut 隧道通信。

2. 保留之前线路的延迟状态（Spoke 访问 Hub 时应优先从 WAN2 的 IPSec 转发），查看 Spoke1 的隧道建立状态，只有去往 Hub 的 IPSec 隧道，没有去往其他 Spoke 的 Shortcut 隧道。

```
Spoke1 # get vpn ipsec tunnel summary
'Spoke1_WAN2' 202.103.254.2:0  selectors(total,up): 1/1  rx(pkt,err): 8423/0  tx(pkt,err): 8408/11
'Spoke1_WAN1' 101.103.254.2:0  selectors(total,up): 1/1  rx(pkt,err): 9284/0  tx(pkt,err): 9193/11
```
bash

3. 使用 Spoke1 内网 PC 访问 Spoke2 内网 PC，触发 Shortcut 隧道建立，可以看到 Shortcut 隧道已经建立，名称为“Spoke1_WAN2_0”，通过 WAN2 上的 IPSec 隧道建立。

```
Spoke1 # get vpn ipsec tunnel summary
'Spoke1_WAN2' 202.103.254.2:0  selectors(total,up): 1/1  rx(pkt,err): 8825/0  tx(pkt,err): 8810/13
'Spoke1_WAN2_0' 202.103.2.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/2
'Spoke1_WAN1' 101.103.254.2:0  selectors(total,up): 1/1  rx(pkt,err): 9686/0  tx(pkt,err): 9595/11
```
bash

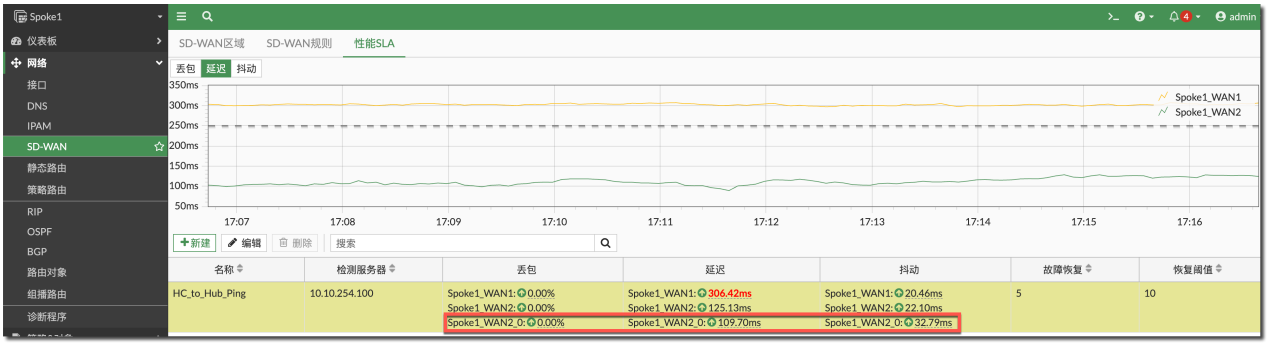



```
date=2023-12-26 time=17:08:55 eventtime=1703581736206155498 tz="+0800" logid="0101037122" type="event" subtype="vpn"
level="notice" vd="root" logdesc="Negotiate IPsec phase 2" msg="negotiate IPsec phase 2" action="negotiate"
remip=202.103.2.2 locip=202.103.1.2 remport=500 locport=500 outintf="port3" cookies="07fb4062cc0f70df/8341519d8c69969b"
user="202.103.2.2" group="N/A" useralt="N/A" xauthuser="N/A" xauthgroup="N/A" assignip=N/A vpntunnel="Spoke1_WAN2_0"
status="success" role="initiator" esptransform="ESP_AES" espauth="HMAC_SHA1" advpnsc=1
```

4. 在 Hub 上抓包查看 Spoke1 内网访问 Spoke2 内网第一次访问时的流量转发，可以看到在 Spoke1 和 Spoke2 之间的 Shortcut 建立前，流量是通过 Hub 中转的，Hub 中转时匹配 SD-WAN 规则 2，tie-break 的方式 input-device 生效，流量从 Hub_WAN2 进入，也从 Hub_WAN2 发出（不会按照 Manual 模式的 SD-WAN Member 顺序从 Hub_WAN1 发出）。

```
Hub # diagnose sniffer packet any 'host 10.10.1.100 and host 10.10.2.100' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.1.100 and host 10.10.2.100]
23.920132 Hub_WAN2 in 10.10.1.100 → 10.10.2.100: icmp: echo request
23.920239 Hub_WAN2 out 10.10.1.100 → 10.10.2.100: icmp: echo request
23.922899 Hub_WAN2 in 10.10.2.100 → 10.10.1.100: icmp: echo reply
23.922913 Hub_WAN2 out 10.10.2.100 → 10.10.1.100: icmp: echo reply
```

5. Shortcut 隧道建立后，SD-WAN 会基于父隧道（Spoke1_WAN2）关联的 SLA 配置，自动创建健康检查检测 Spoke2 的 Tunnel 接口 IP，检测方式固定为 Ping（这也就是为什么要在所有 Spoke 的 IPSec Tunnel 接口上开启允许 Ping 访问）。




```
Spoke1 # diagnose sys sdwan health-check
Health Check(HC_to_Hub_Ping):
Seq(1 Spoke1_WAN1): state(alive), packet-loss(0.000%) latency(301.470), jitter(20.919), mos(3.591), bandwidth-
up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x0
Seq(2 Spoke1_WAN2): state(alive), packet-loss(0.000%) latency(127.110), jitter(22.480), mos(4.259), bandwidth-
up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x1
Seq(2 Spoke1_WAN2_0): state(alive), packet-loss(0.000%) latency(112.727), jitter(29.479), mos(4.260), bandwidth-
up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0x1
```

6. 查看 Spoke1 上的 BGP 路由表，去往 Spoke2 内网路由的出接口已变为在 Shortcut 隧道接口（Spoke1_WAN2_0）和 WAN1 上的 IPSec 隧道接口（Spoke1_WAN1）之间负载。

```bash
Spoke1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 100.1.1.2 [2] (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 00:19:40, [1/0]
                     [200/0] via 200.1.1.2 [2] (recursive is directly connected, Spoke1_WAN2_0), 00:19:40, [1/0]
B       10.10.3.0/24 [200/0] via 100.1.1.3 [2] (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 02:33:25, [1/0]
                     [200/0] via 200.1.1.3 [2] (recursive via Spoke1_WAN2 tunnel 202.103.254.2), 02:33:25, [1/0]
B       10.10.254.0/24 [200/0] via 100.1.1.254 (recursive via Spoke1_WAN1 tunnel 101.103.254.2), 02:35:09, [1/0]
                       [200/0] via 200.1.1.254 (recursive via Spoke1_WAN2 tunnel 202.103.254.2), 02:35:09, [1/0]
```

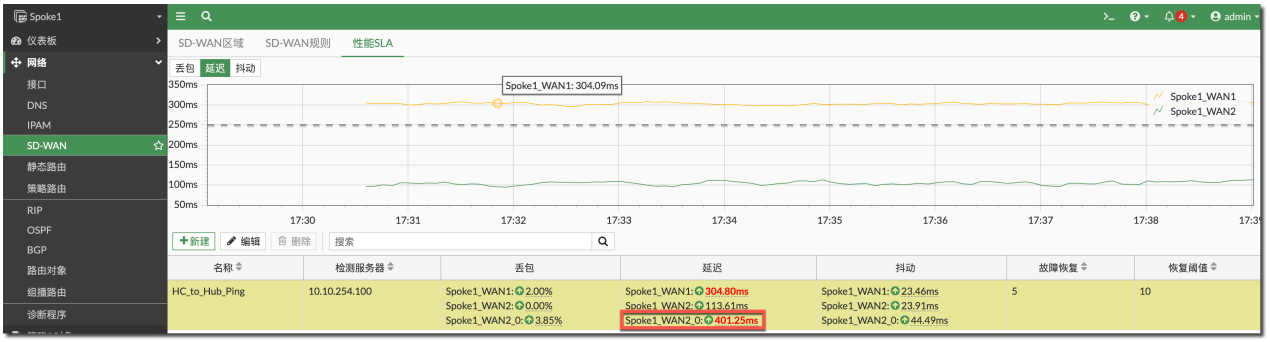
7. 查看 Spoke1 上 SD-WAN 规则 2 的选路状态，现在优先从 WAN2 上 IPSec 隧道的 Shortcut（Spoke1_WAN2_0）进行转发，而不是从父隧道（Spoke1_WAN2）转发，这说明在 SLA 类型的 SD-WAN 规则中，Shortcut 隧道优先于其父隧道，除非 Shortcut 隧道的 SLA 不满足目标。

```bash
Spoke1 # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0×200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0×0/0×0), Protocol(0: 1→65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    2: seq_num(2), interface(Spoke1_WAN2):
       1: Spoke1_WAN2_0(22)
  Members(3):
    1: Seq_num(2 Spoke1_WAN2_0), alive, sla(0×1), gid(0), cfg_order(1), local cost(0), selected
    2: Seq_num(2 Spoke1_WAN2), alive, sla(0×1), gid(0), cfg_order(1), local cost(0), selected
    3: Seq_num(1 Spoke1_WAN1), alive, sla(0×0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.10.1.0-10.10.1.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

8. 再次使用 Spoke1 内网 PC 访问 Spoke2 内网 PC，在 Spoke1 上抓包可以看到流量通过 WAN2 上的 Shortcut 隧道（Spoke1_WAN2_0）隧道转发。

```bash
Spoke1 # diagnose sniffer packet any 'host 10.10.2.100' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.100]
7.778827 port4 in 10.10.1.100 → 10.10.2.100: icmp: echo request
7.778945 Spoke1_WAN2_0 out 10.10.1.100 → 10.10.2.100: icmp: echo request
7.870455 Spoke1_WAN2_0 in 10.10.2.100 → 10.10.1.100: icmp: echo reply
7.870486 port4 out 10.10.2.100 → 10.10.1.100: icmp: echo reply
8.798494 port4 in 10.10.1.100 → 10.10.2.100: icmp: echo request
8.798518 Spoke1_WAN2_0 out 10.10.1.100 → 10.10.2.100: icmp: echo request
8.870069 Spoke1_WAN2_0 in 10.10.2.100 → 10.10.1.100: icmp: echo reply
8.870081 port4 out 10.10.2.100 → 10.10.1.100: icmp: echo reply
```

9. 调整 Spoke1 与 Spoke2 之间 Shortcut 隧道（Spoke1_WAN2_0）的延迟到 400ms，但父隧道（Spoke1_WAN2）延迟不变。




```bash
Spoke1 # diagnose sys sdwan health-check
Health Check(HC_to_Hub_Ping):
Seq(1 Spoke1_WAN1): state(alive), packet-loss(1.000%) latency(314.047), jitter(16.431), mos(3.573), bandwidth-up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0×0
Seq(2 Spoke1_WAN2): state(alive), packet-loss(0.000%) latency(112.318), jitter(24.059), mos(4.291), bandwidth-up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0×1
Seq(2 Spoke1_WAN2_0): state(alive), packet-loss(2.410%) latency(393.256), jitter(26.858), mos(3.071), bandwidth-up(65535000), bandwidth-dw(65535000), bandwidth-bi(131070000) sla_map=0×0
```

10. 查看 Spoke1 上 SD-WAN 规则 2 的选路状态，现在优先从 WAN2 上 IPSec 父隧道（Spoke1_WAN2）进行转发，而不是从 Shortcut 隧道（Spoke1_WAN2_0）转发，事实的确如此吗?

```bash
Spoke1 # diagnose sys sdwan service 2

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(15), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    1: seq_num(2), interface(Spoke1_WAN2):
       1: Spoke1_WAN2_0(25)
  Members(3):
    1: Seq_num(2 Spoke1_WAN2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    2: Seq_num(1 Spoke1_WAN1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(2 Spoke1_WAN2_0), alive, sla(0x0), gid(0), cfg_order(1), local cost(0), selected
  Src address(1):
        10.10.1.0-10.10.1.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

11. 再次使用 Spoke1 内网 PC 访问 Spoke2 内网 PC，在 Spoke1 上抓包可以看到流量通过 WAN1 上 IPSec 隧道（Spoke1_WAN1）隧道转发，而不是 Spoke1_WAN2 隧道，并没有按照 SD-WAN 规则状态中的选路从 Spoke1_WAN2 隧道转发。

```bash
Spoke1 # diagnose sniffer packet any 'host 10.10.2.100' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.100]
4.313351 port4 in 10.10.1.100 -> 10.10.2.100: icmp: echo request
4.313453 Spoke1_WAN1 out 10.10.1.100 -> 10.10.2.100: icmp: echo request
4.639634 Spoke1_WAN1 in 10.10.2.100 -> 10.10.1.100: icmp: echo reply
4.639664 port4 out 10.10.2.100 -> 10.10.1.100: icmp: echo reply
```

12. 这是由于 SD-WAN 规则选路时也会根据流量的目的地址查询路由表，如果最佳路由下一跳接口是 SD-WAN 成员，则通过该 SD-WAN 规则的 SLA 找到最佳的接口转发，该接口必须要有路由。在 Spoke1 的路由表中，通过 Spoke1_WAN2 隧道去往 Spoke2 内网的路由出接口已变为 Shortcut 隧道 Spoke1_WAN2_0，而不是 Spoke1_WAN2，所以跳过成员 Spoke1_WAN2，从下一个成员 Spoke1_WAN1 转发（参考网络管理 → SD-WAN → SD-WAN 介绍 → SD-WAN 数据转发逻辑 ↗章节中的 SD-WAN 工作流程注意事项）。

13. 随后 Spoke1 触发了 Spoke1_WAN1 隧道上的去往 Spoke2 的 Shortcut 隧道连接 Spoke1_WAN1_0，后续流量通过 Shortcut 隧道连接 Spoke1_WAN1_0 转发。

```bash
Spoke1 # diagnose sniffer packet any 'host 10.10.2.100' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.100]
36.862757 port4 in 10.10.1.100 -> 10.10.2.100: icmp: echo request
36.862888 Spoke1_WAN1_0 out 10.10.1.100 -> 10.10.2.100: icmp: echo request
37.167399 Spoke1_WAN1_0 in 10.10.2.100 -> 10.10.1.100: icmp: echo reply
37.167437 port4 out 10.10.2.100 -> 10.10.1.100: icmp: echo reply
```

14. Spoke2、Spoke3 之间的 Shortcut 隧道测试与 Spoke1 一致，这里不再赘述。

Hub 访问 Spoke

1. 保留之前线路的延迟状态（Spoke1 主动访问 Hub 时应优先从 WAN2 的 IPSec 转发）。
2. 使用 Hub 内网 PC 访问 Spoke1 的内网 PC，由于 Hub 的 SD-WAN 规则配置为 Manual 模式，流量强制从 Hub_WAN1 转发（除非 Hub_WAN1 隧道中断）。



3. 在 Spoke1 上查看返回的流量走向，只要路由表中存在返回的路由，就会遵循源进源出原则，而不是匹配 SD-WAN 规则。

```bash
Spoke1 # diagnose sys sdwan service 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(12), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(4):
    1: Seq_num(2 Spoke1_WAN2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    2: Seq_num(1 Spoke1_WAN1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.10.1.0-10.10.1.255

  Dst address(1):
        10.10.254.0-10.10.254.255

Spoke1 # diagnose sniffer packet any 'host 10.10.254.100 and host 10.10.1.100' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.254.100 and host 10.10.1.100]
1.105333 Spoke1_WAN1 in 10.10.254.100 → 10.10.1.100: icmp: echo request
1.105377 port4 out 10.10.254.100 → 10.10.1.100: icmp: echo request
1.105956 port4 in 10.10.1.100 → 10.10.254.100: icmp: echo reply
1.105979 Spoke1_WAN1 out 10.10.1.100 → 10.10.254.100: icmp: echo reply
```

注意事项

默认配置下，当 VPN 或专线线路（无 SNAT）SD-WAN 选路出现变化后，已有的会话会立即切换到新链路上，如果想在 SD-WAN 线路切换后让已有会话保持在原来的线路上，只有新会话创建在新线路上，需要在对应 VPN 接口或专线接口上使用如下命令。

```bash
config system interface
    edit "Spoke1_WAN1"
        set preserve-session-route enable    //默认为disable//
    next
end
```

NetFlow

- NetFlow 的配置
- NetFlow 模板
- 在 VPN 接口上配置 NetFlow

NetFlow 的配置

NetFlow 介绍

NetFlow 允许基于设备接口收集网络流量的统计信息，它提供了在数据进入或离开接口时收集 IP 网络流量的能力。通过分析 NetFlow 提供的数据，网络管理员可以确定流量的源和目的地、服务类别以及拥塞的原因。

NetFlow 通常使用 UDP 协议进行导出，并使用 NetFlow Collector 进行收集。NetFlow Collector 的 IP 地址和目标 UDP 端口必须在发送设备（FortiGate）上进行配置。标准的 NetFlow UDP 端口为 2055。

在 FortiGate 中，根据 netflow-sampler 监控的接口，发送所有此接口相关的会话信息，无法设置 NetFlow 的接口采样率。

> **重要**
>
> 在任何接口上配置 NetFlow 都不会影响 NP 芯片的加速功能。
>
> 但如果使用的是 sFlow，是无法被任何 NP 芯片加速的。

NetFlow 配置

NetFlow 发送配置

未开启 VDOM 的配置

```bash
7.0.X、7.2.0~7.2.7、7.4.0~7.4.3版本（只能配置1个Collector）：
config system netflow
    set collector-ip <ip>
    set collector-port <port>
    set source-ip <ip>
    set active-flow-timeout <integer>
    set inactive-flow-timeout <integer>
    set template-tx-timeout <integer>
    set template-tx-counter <integer>
end

其他7.X.X版本（可以配置最多6个Collector）：
config system netflow
    set active-flow-timeout {integer}
    config collectors
        edit <id>
            set collector-ip {string}
            set collector-port {integer}
            set source-ip {string}
            set interface-select-method [auto|sdwan| ... ]
            set interface {string}
        next
    end
    set inactive-flow-timeout {integer}
    set template-tx-counter {integer}
    set template-tx-timeout {integer}
end
```

参数名称	参数说明
collector-ip [ip]	NetFlow collector 的 IPv4 或 IPv6 地址（IPv6 在 7.2.0 GA 后支持）
collector-port [port]	NetFlow collector 用于接收 NetFlow 流量的端口，默认为 2055
source-ip [ip]	指定 FortiGate 发送 NetFlow 流量的源 IP，默认根据路由出接口指定源 IP
active-flow-timeout [integer]	报告活动中的会话 NetFlow 信息的间隔，默认为 30 分钟，范围 1~60
inactive-flow-timeout [integer]	报告已结束的会话 NetFlow 信息的间隔，默认为 15 秒，范围 10~600
template-tx-timeout [integer]	模板数据流的发送间隔，默认 30 分钟，范围 1~1440
template-tx-counter [integer]	数据流记录的产生计数，达到这个计数后也会发送模板数据流到 collector，默认为 20，范围 10~6000

开启多 VDOM 的配置

重要

在多 VDOM 环境中，无法在 root VDOM 或管理 VDOM 上配置 Netflow，root VDOM 或管理 VDOM 的 NetFlow 配置将从 `config system global` 下的 NetFlow 配置继承。

在多 VDOM + HA 环境中，当 HA 禁用“ha-direct”功能时，Netflow 还是会在独立管理口上发送流量。如果需要通过业务 VDOM 中的路由接口发送 NetFlow 流量，必须在 `config system global` 下禁用 NetFlow IP 或配置 NetFlow IP 为 0.0.0.0。

bash
```
7.0.X、7.2.0~7.2.7、7.4.0~7.4.3版本:
config vdom
    edit <vdom>
        config system vdom-netflow
            set vdom-netflow enable
            set collector-ip <ip>
            set collector-port <port>
            set source-ip <ip>
        end
    next
end

其他7.X.X版本:
config vdom
    edit <vdom>
        config system vdom-netflow
            set active-flow-timeout {integer}
            config collectors
                edit <id>
                    set collector-ip {string}
                    set collector-port {integer}
                    set source-ip {string}
                    set interface-select-method [auto|sdwan| ... ]
                    set interface {string}
                next
            end
            set inactive-flow-timeout {integer}
            set template-tx-counter {integer}
            set template-tx-timeout {integer}
        end
    next
end
```

NetFlow 采样器配置

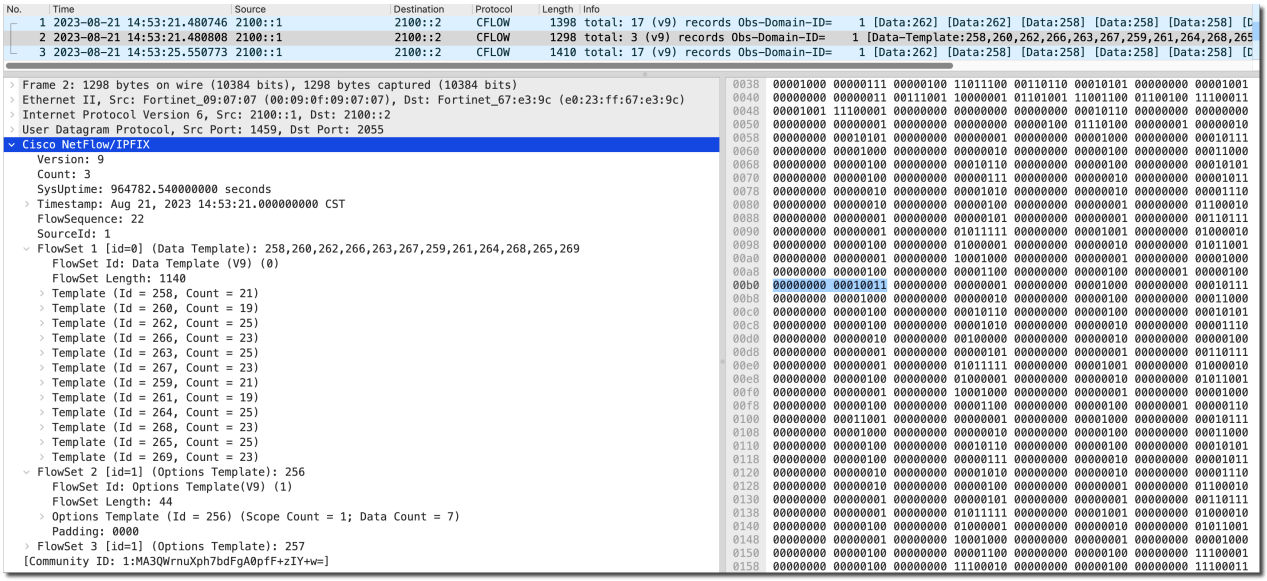
bash
```
config system interface
    edit <interface>
        set netflow-sampler {disable | tx | rx | both}
    next
end
```

参数名称	参数说明
disable	接口下关闭 NetFlow 采样，默认配置
tx	采样此接口发送的流量
rx	采样此接口接收的流量
both	同时采样此接口发送和接收的流量

配置验证

1. 如果在 NetFlow 上无法查询到 FortiGate 的 NetFlow 数据，可以在 FortiGate 上抓包查看 NetFlow 数据是否已发送，如下图所示。

```
diagnose sniffer packet 'host <collector-ip> and port <collector-port>' 6 0 l
```



2. NetFlow 功能通过 FortiGate 的 sflow 进程工作，通过以下命令，可以看到 Netflow 的工作状态。

```
FortiGate # diagnose test application sflowd 3

==== Netflow Vdom Configuration ====
Global collector:172.22.5.41:[2055] source ip:  active-timeout(seconds):1800 inactive-timeout(seconds):15
___ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.22.5.41:2055,src_ip:172.22.5.19
   |_ seq_num:301 pkts/time to next template: 7/1259
   |_ exported: Bytes:114028124, Packets:322978, Sessions:13457 Flows:13494
   |_ active_intf: 1
   |___ interface:wan2 sample_direction:both device_index:6 snmp_index:2
```

3. 查看流量会话是否匹配 NetFlow，如下所示，会话被打上了 NetFlow-reply 方向的标记。

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1732 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0
use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=145572/1733/1 reply=145572/1733/1 tuples=2
tx speed(Bps/kbps): 83/0 rx speed(Bps/kbps): 83/0
orgin->sink: org pre->post, reply pre->post dev=5->26/26->5 gwy=10.39.252.244/172.16.200.55
hook=post dir=org act=snat 172.16.200.55:61290->8.8.8.8:8(10.39.252.243:61290)
hook=pre dir=reply act=dnat 8.8.8.8:61290->10.39.252.243:0(172.16.200.55:61290)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
```

```
serial=00001298 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
total session 1
```

NetFlow 模板

模板介绍

NetFlow 使用模板来捕获和分类收集到的数据流，FortiOS 支持以下 NetFlow 模板 ID（7.2.5 GA）：

模板名称	模板 ID	模板描述
STAT_OPTIONS	256	发送方的统计信息
APP_ID_OPTIONS	257	应用信息
IPV4	258	非 NAT IPv4 流量
IPV6	259	非 NAT IPv6 流量
ICMP4	260	非 NAT ICMPv4 流量
ICMP6	261	非 NAT ICMPv6 流量
IPV4_NAT	262	源/目 NAT IPv4 流量
IPV4_AF_NAT	263	NAT46 的 IPv4 流量
IPV6_NAT	264	源/目 NAT IPv6 流量
IPV6_AF_NAT	265	NAT64 的 IPv6 流量
ICMP4_NAT	266	源/目 NAT ICMPv4 流量
ICMP4_AF_NAT	267	NAT46 的 ICMPv4 流量
ICMP6_NAT	268	源/目 NAT ICMPv6 流量
ICMPv6_AF_NAT	269	NAT64 的 ICMPv6 流量

模板支持参数

STAT_OPTIONS 模板

Description	Statistics information about exporter
Scope Field Count	1
Data Field Count	7
Option Scope Length	4
Option Length	28
Padding	0000

Scope 字段

Field #	Field	Type	Length
1	System	System (1)	2

Data 字段

Field #	Field	Type	Length
1	TOTAL_BYTES_EXP	TOTAL_BYTES_EXP (40)	8
2	TOTAL_PKTS_EXP	TOTAL_PKTS_EXP (41)	8
3	TOTAL_FLOWS_EXP	TOTAL_FLOWS_EXP (42)	8
4	FLOW_ACTIVE_TIMEOUT	FLOW_ACTIVE_TIMEOUT (36)	2
5	FLOW_INACTIVE_TIMEOUT	FLOW_INACTIVE_TIMEOUT (37)	2
6	SAMPLING_INTERVAL	SAMPLING_INTERVAL (34)	4
7	SAMPLING_ALGORITHM	SAMPLING_ALGORITHM (35)	1

APP_ID_OPTIONS 模板

Description	Application information
Scope Field Count	1
Data Field Count	4
Option Scope Length	4
Option Length	16
Padding	0000

Scope 字段

Field #	Field	Type	Length
1	System	System (1)	2

Data 字段

Field #	Field	Type	Length
1	APPLICATION_ID	APPLICATION_ID (95)	9
2	APPLICATION_NAME	APPLICATION_NAME (96)	64
3	APPLICATION_DESC	APPLICATION_DESC (94)	64
4	applicationCategoryName	applicationCategoryName (372)	32

IPV4 模板

Description	Application information
Data Field Count	17

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	APPLICATION_ID	APPLICATION_ID (95)	9
13	FLOW_FLAGS	FLOW_FLAGS (65)	2
14	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
15	flowEndReason	flowEndReason (136)	1
16	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
17	IP_DST_ADDR	IP_DST_ADDR (12)	4

IPV6 模板

Description	No NAT IPv6 traffic
Data Field Count	17

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	APPLICATION_ID	APPLICATION_ID (95)	9
13	FLOW_FLAGS	FLOW_FLAGS (65)	2
14	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
15	flowEndReason	flowEndReason (136)	1
16	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
17	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16

ICMP4 模板

Description	No NAT IPv4 traffic
Data Field Count	16

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
16	IP_DST_ADDR	IP_DST_ADDR(12)	4

ICMP6 模板

Description	No NAT ICMPv6 traffic
Data Field Count	16

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
16	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16

IPv4_NAT 模板

Description	Source/Destination NAT IPv4 traffic
Data Field Count	25

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	IP_TOS	ipClassofService (5)	1
14	DST_DOS	postIpClassOfService (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID(66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
21	IP_DST_ADDR	IP_DST_ADDR (12)	4
22	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
23	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
24	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
25	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

IPV4_AF_NAT 模板

Description	AF NAT IPv4 traffic (4->6)
Data Field Count	21

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	APPLICATION_ID	APPLICATION_ID (95)	9
13	FLOW_FLAGS	FLOW_FLAGS (65)	2
14	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
15	flowEndReason	flowEndReason (136)	1
16	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
17	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
18	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
19	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
20	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
21	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

IPV6_NAT 模板

Description	Source/Destination NAT IPv6 traffic
Data Field Count	21

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	APPLICATION_ID	APPLICATION_ID (95)	9
13	FLOW_FLAGS	FLOW_FLAGS (65)	2
14	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
15	flowEndReason	flowEndReason (136)	1
16	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
17	IP_DST_ADDR	IP_DST_ADDR (12)	4
18	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
19	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
20	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
21	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

IPV6_AF_NAT 模板

Description	AF NAT IPv6 traffic (6->4)
Data Field Count	21

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	APPLICATION_ID	APPLICATION_ID (95)	9
13	FLOW_FLAGS	FLOW_FLAGS (65)	2
14	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
15	flowEndReason	flowEndReason (136)	1
16	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
17	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
18	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
19	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
20	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
21	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

ICMPV4_NAT 模板

Description	Source/Destination NAT ICMPv4 traffic
Data Field Count	20

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
16	IP_DST_ADDR	IP_DST_ADDR (12)	4
17	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
18	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
19	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
20	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

ICMPV4_AF_NAT 模板

Description	AF NAT ICMPv4 traffic (4->6)
Data Field Count	20

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
16	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
17	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
18	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
19	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
20	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

ICMPV6_NAT 模板

Description	Source/Destination NAT ICMPv6 traffic
Data Field Count	20

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
16	IP_DST_ADDR	IP_DST_ADDR (12)	4
17	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
18	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
19	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
20	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

ICMPV6_AF_NAT 模板

Description	AF NAT ICMPv6 traffic (6->4)
Data Field Count	20

Data 字段

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	APPLICATION_ID	APPLICATION_ID (95)	9
12	FLOW_FLAGS	FLOW_FLAGS (65)	2
13	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
14	flowEndReason	flowEndReason (136)	1
15	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
16	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
17	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
18	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
19	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
20	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

在 VPN 接口上配置 NetFlow

NetFlow 支持在 VPN 接口（如 IPSec Tunnel、SSL VPN Tunnel、IP in IP Tunnel、GRE Tunnel 等）以及 FortiExtender 接口上配置。

NetFlow 配置在 VPN 接口上时，不会影响到 NP 的加速功能。

配置举例

1. 配置 NetFlow collector 信息后，在 IPSec tunnel 接口下开启 NetFlow 采样。

```
config system interface
    edit "A-to-B_vpn"
        set vdom "vdom1"
        set type tunnel
        set netflow-sampler both
        set snmp-index 42
        set interface "port3"
    next
end
```

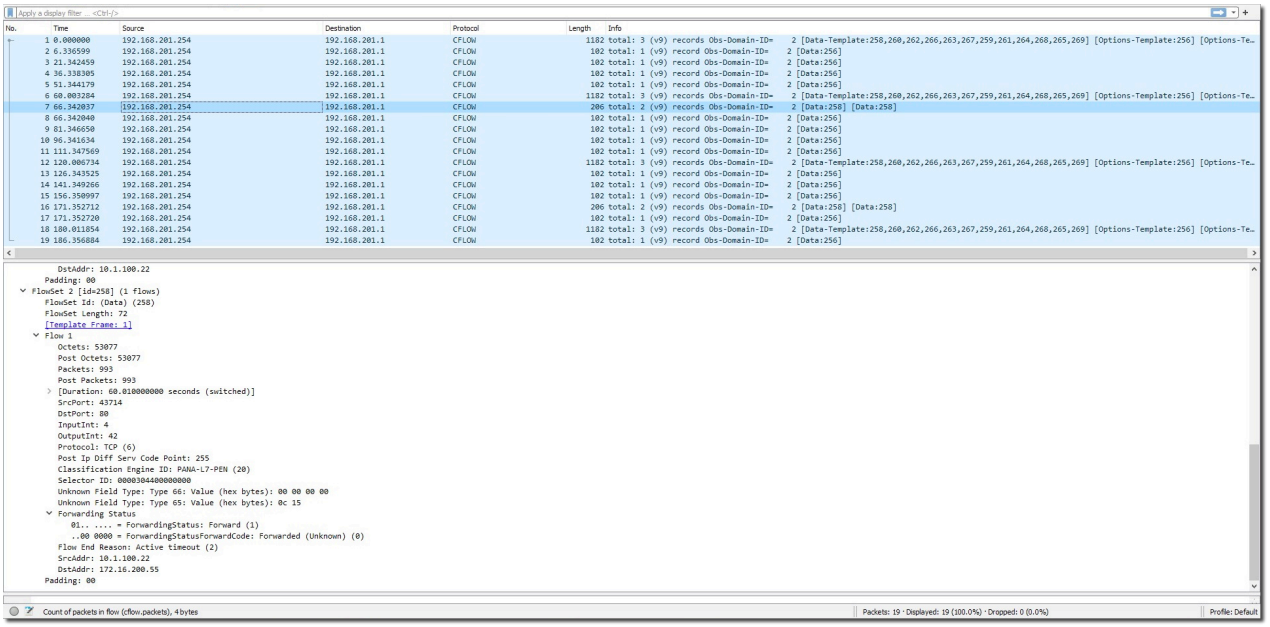
2. 查看 NetFlow 采样的状态信息。

```
# diagnose test application sflowd 3
═══ Netflow Vdom Configuration ═══
Global collector:172.18.60.80:[2055] source ip: 0.0.0.0 active-timeout(seconds):60 inactive-timeout(seconds):15
____ vdom: vdom1, index=1, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.18.60.80[2055],src_ip:10.1.100.1,seq_num:60,pkts/time to next template: 15/6
   |_ exported: Bytes:11795591, Packets:48160, Sessions:10 Flows:34
   |____ interface:A-to-B_vpn sample_direction:both device_index:52 snmp_index:42
```

3. 查看 IPSec 隧道内流量相关会话的 NetFlow 标记。

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=6 expire=3599 timeout=3600 flags=00000000 socktype=0 sockport=0 av_idx=0
use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=6433/120/1 reply=884384/713/1 tuples=2
tx speed(Bps/kbps): 992/7 rx speed(Bps/kbps): 136479/1091
orgin→sink: org pre→post, reply pre→post dev=10→52/52→10 gwy=10.2.2.2/10.1.100.22
hook=pre dir=org act=noop 10.1.100.22:43714→172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80→10.1.100.22:43714(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:0c:29:ac:ae:4f
misc=0 policy_id=5 auth_info=0 chk_client_info=0 vd=1
serial=00003b6c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0×000001 no_offload
npu info: flag=0×82/0×00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0×0000/0×0000
vlifid=0/0, vtag_in=0×0000/0×0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
total session 1
```

4. 抓包查看 FortiGate 向 Collector 发送的 NetFlow 流量。


Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000 192.168.201.254 192.168.201.1 CFLOW 1182 total: 3 (v9) records Obs-Domain-ID= 2 [Data-Template:258,260,262,266,263,267,259,261,264,268,265,269] [Options-Template:256] [Options-Te…
2 6.336599 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
3 21.342459 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
4 36.338305 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
5 51.344179 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
6 60.003284 192.168.201.254 192.168.201.1 CFLOW 1182 total: 3 (v9) records Obs-Domain-ID= 2 [Data-Template:258,260,262,266,263,267,259,261,264,268,265,269] [Options-Template:256] [Options-Te…
7 66.342037 192.168.201.254 192.168.201.1 CFLOW 206 total: 2 (v9) records Obs-Domain-ID= 2 [Data:258] [Data:258]
8 66.342040 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
9 81.346650 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
10 96.341634 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
11 111.347569 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
12 120.006734 192.168.201.254 192.168.201.1 CFLOW 1182 total: 3 (v9) records Obs-Domain-ID= 2 [Data-Template:258,260,262,266,263,267,259,261,264,268,265,269] [Options-Template:256] [Options-Te…
13 126.343525 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
14 141.349266 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
15 156.350997 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
16 171.352712 192.168.201.254 192.168.201.1 CFLOW 206 total: 2 (v9) records Obs-Domain-ID= 2 [Data:258] [Data:258]
17 171.352720 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
18 180.011854 192.168.201.254 192.168.201.1 CFLOW 1182 total: 3 (v9) records Obs-Domain-ID= 2 [Data-Template:258,260,262,266,263,267,259,261,264,268,265,269] [Options-Template:256] [Options-Te…
19 186.356884 192.168.201.254 192.168.201.1 CFLOW 102 total: 1 (v9) record Obs-Domain-ID= 2 [Data:256]
DstAddr: 10.1.100.22
Padding: 00
FlowSet 2 [id=258] (1 flows)
FlowSet Id: (Data) (258)
FlowSet Length: 72
[Template Frame: 1]
Flow 1
Octets: 53077
Post Octets: 53077
Packets: 993
Post Packets: 993
[Duration: 60.010000000 seconds (switched)]
SrcPort: 43714
DstPort: 80
InputInt: 4
OutputInt: 42
Protocol: TCP (6)
Post Ip Diff Serv Code Point: 255
Classification Engine ID: PANA-L7-PEN (20)
Selector ID: 0000304400000000
Unknown Field Type: Type 66: Value (hex bytes): 00 00 00 00
Unknown Field Type: Type 65: Value (hex bytes): 0c 15
Forwarding Status
01.. .... = ForwardingStatus: Forward (1)
..00 0000 = ForwardingStatusForwardCode: Forwarded (Unknown) (0)
Flow End Reason: Active timeout (2)
SrcAddr: 10.1.100.22
DstAddr: 172.16.200.55
Padding: 00
Count of packets in flow (cflow.packets), 4 bytes
Packets: 19 · Displayed: 19 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

SAA-FortiLink

FortiLink 简介和上线指南

- FortiLink 简介
- FortiLink 简要上线指南

FortiLink 部署方案和举例

- FortiLink 星型组网
- FortiLink 星型组网配置举例
- FortiLink 环形组网
- FortiLink 环形组网配置举例
- FortiLink 星型环形混合组网
- FortiLink 交换机零配置跨三层部署
- 交换机零配置跨二层部署

FortiLink 堆叠组网和举例

- FortiLink 部署之 MCLAG 堆叠组网
- FortiLink MCLAG - 堆叠组网 1 Tier 配置举例
- FortiLink MCLAG - 堆叠组网 2 Tier 配置举例
- FortiLink MCLAG - 与第三方设备进行跨堆叠交换机聚合

FortiLink 排障

- 快速查找异常流量

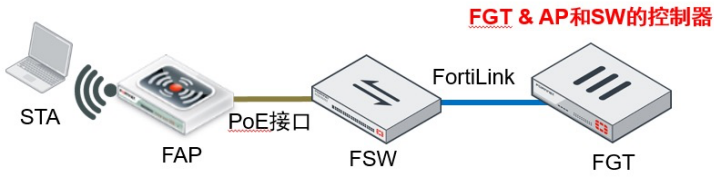
FortiLink 简介和上线指南

- FortiLink 简介
- FortiLink 简要上线指南

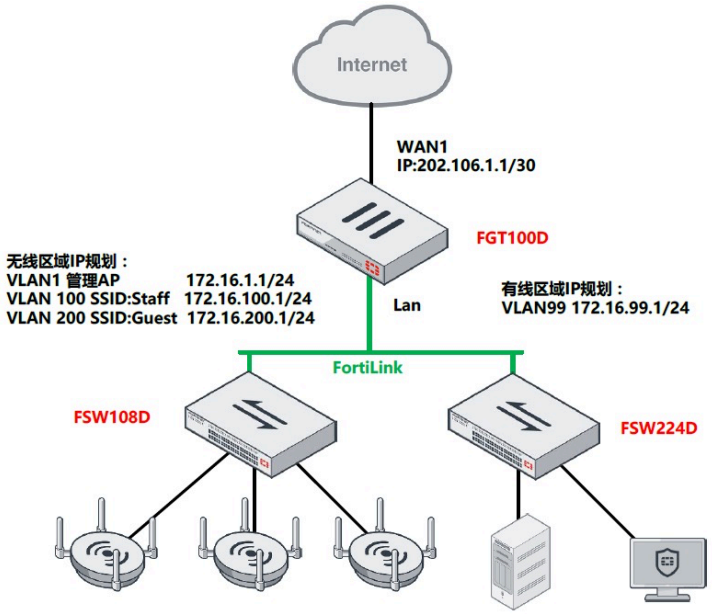
FortiLink 简介

简介

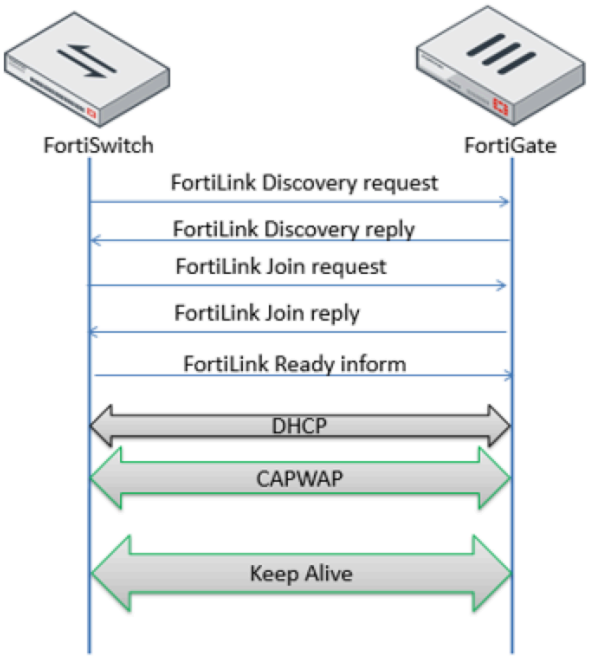
1. FortiLink 是防火墙和被管理的交换机之间的通讯协议。
2. Fortinet 的私有协议，FortiGate 基于 FortiLink 来管理 FortiSwitch。
3. FortiLink 链路承载了 FortiSwitch/FortiAP 和 FortiGate 之间的通讯报文以及用户业务数据报文（FortiLink 报文 + CAPWAP 控制报文 + 真实业务流量）



典型组网环境



FortiLink 报文交互



CAPWAP 交互过程记录（三层报文, UDP Port 5246）

Discovery:

```
FG100D3G14800154 # diagnose debug application cu_acd 255
79749.723 ==========================cuAcProcRawMsg            14  1==========================
79749.723 cuAcProcDiscoverReqMsg: received DISCOVERY_REQ from S224DF3X15000446 169.254.1.2/5246
79749.724 cuAcProcDiscoverReqMsg: sent CW_CTL_MSG_TYPE_DISCOVERY_RESP to ws (0-169.254.1.2:5246)
79749.724 ==========================cuAcProcRawMsg            14  2==========================
79749.729 ==========================cuAcProcRawMsg            14  1==========================
```

Join:

```
79749.822 cuAcProcPlainCtlMsg: received JOIN_REQ from ws (0-169.254.1.2:5246)
79749.822 AC_EV_GEN - CWAE_JOIN_REQ_RECV me 0x1501d2b0 ws ws (0-169.254.1.2:5246)
79749.825 CUAS_JOIN_enter: sending JOIN RESP msg.
```

Run:

```
79752.513 FSM: old CWAS_DATA_CHECK(10) ev CWAE_DATA_CHECK_TIMER_EXPIRE(33) new CWAS_RUN(11)
79752.513 cu_set_managed_switch_state
79752.513 cu_set_managed_switch_state managed-switch S224DF3X15000446
```

Keep Alive（FGT 侧）

FGT 对于 FortiLink 状态的健康，FGT 和 FSW 各自使用不同的 Keep alive 机制, FGT 使用基于 CAPWAP 机制的 Keep Alive，FGT 每 30 秒发送一次 REQ_MSG 给 FSW, 如果没有收到 RSP, 则重传 6 次, 如果重传 6 次都没有成功, 则认为 FSW 掉线。

```
FG100D3G14800154 # diagnose debug application cu_acd 255
82790.513 cuAcSendPoeGeneralStatsReq wtp general poe stats request
82790.513 cuAcSend_REQ_MSG: sending  (31) msg.
82790.513 cuAcSendRawMsgQueue: SENDING OUT type 31 seqNum 3

82793.513 cuAcSendPendingMsg: for ws (0-169.254.1.2:5246)... 31 3 0 82790 82793
82793.513 cuAcSendPendingMsg: RESENDING Msg type 31 seqNum 3 retransmitCnt 0.
.................
82859.514 cuAcSendPendingMsg: for ws (0-169.254.1.2:5246)... 31 3 6 82859 82859
82859.514 cuAcSendPendingMsg: Msg type 31 seqNum 3 send failure. MaxRetry is reached.

82859.515 FSM: old CWAS_RUN(11) ev CWAE_MAX_RRTRANSMIT_REACHED(21) new CWAS_DTLS_TD(6)
82859.515 cu_set_managed_switch_state
82859.515 cu_set_managed_switch_state managed-switch S224DF3X15000446
```

Keep Alive（FSW 侧）

FSW 对于 FortiLink 状态的健康检查：

1. FSW 会实时监控 Fortilink 的物理端口状态, 如果端口 down, 则 FSW 认为 FortiLink down。

```
S224DF3X15000446 # diagnose debug application fortilinkd 255

448s:942ms:1us fl_node_rec_scan[2356]:connected to controller
449s:152ms:756us flp_event_handler[1120]:node: port24 received event FL_EV_ECHO_TIMEOUT state FL_STATE_READY
449s:213ms:580us flp_event_handler[1120]:node: port24 received event FL_EV_RECV_ECHO_REPLY state FL_STATE_READY
449s:214ms:176us flp_event_handler[1120]:node: port24 received event FL_EV_ECHO_REPLY_VALID state FL_STATE_READY
452s:161ms:286us flp_event_handler[1120]:node: port24 received event FL_EV_ECHO_TIMEOUT state FL_STATE_READY
452s:201ms:355us flp_event_handler[1120]:node: port24 received event FL_EV_RECV_ECHO_REPLY state FL_STATE_READY
452s:201ms:940us flp_event_handler[1120]:node: port24 received event FL_EV_ECHO_REPLY_VALID state FL_STATE_READY

505s:766ms:155us fl_node_rec_scan[2363]:not connected to controller
506s:76ms:152us flp_event_handler[1120]:node: port24 received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
507s:426ms:644us flp_event_handler[1120]:node: internal received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
509s:77ms:514us flp_event_handler[1120]:node: port24 received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
510s:453ms:208us flp_event_handler[1120]:node: internal received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
512s:86ms:133us flp_event_handler[1120]:node: port24 received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
513s:436ms:182us flp_event_handler[1120]:node: internal received event FL_EV_DISC_TIMEOUT state FL_STATE_DISC
```

2. FortiLink down 后，FSW 会继续尝试与 FGT 建立 FortiLink，每 3 秒发送一次 FortiLink Discovery 报文，如果由于链路原因没有收到 FGT 返回的应答，FSW 默认有一个 30 秒的超时时间，超过此时间，则 FSW 认为 FortiLink down。FSW 会继续尝试与 FGT 建立 FortiLink，重复 FSW 上线过程，包括重新 CAPWAP 协商。

3. FSW 对于 CAPWAP 的监控：

   - switch 每 30 秒发送一次 echo request 并等待 echo response
   - 如果连续 10 个 echo 没有 response, 则重置 CAPWAP 状态, 交换机认为自己掉线, 即 capwap 超时时间 300 秒
   - 每次 FortiLink 重新建立后, 如果当前 CAPWAP 状态为 RUN, 会重新开始 CAPWAP 进程

```
fl_node_set_controller_info[2303]:fl_node_set_controller_info:restaring capwap
```

FortiLink 中断检测过程

1. FSW 继续使用当前的配置工作，本地转发的数据继续转发。
2. 如果此时重启 FSW，FSW 重启后会继续使用重启之前的配置工作，本地转发的数据继续转发。
3. 如果 FSW 在与 FGT 断了后，被别的 FGT 发现并授权，此时 FSW 不会再重启, 但会丢弃之前在旧 FGT 上获得的配置，重新从新的 FGT 上获取配置。这种情况下，FSW 从一个 FGT 手动切换到另外一个 FGT 上，最好先清一下配置。
4. 如果 FSW 掉线后，在 FSW 上如果又修改了配置后，在 FSW 重新上线后，FGT 会重新下发配置，之前在 FSW 上修改的配置会被丢弃。

其他

官方参考配置文档：

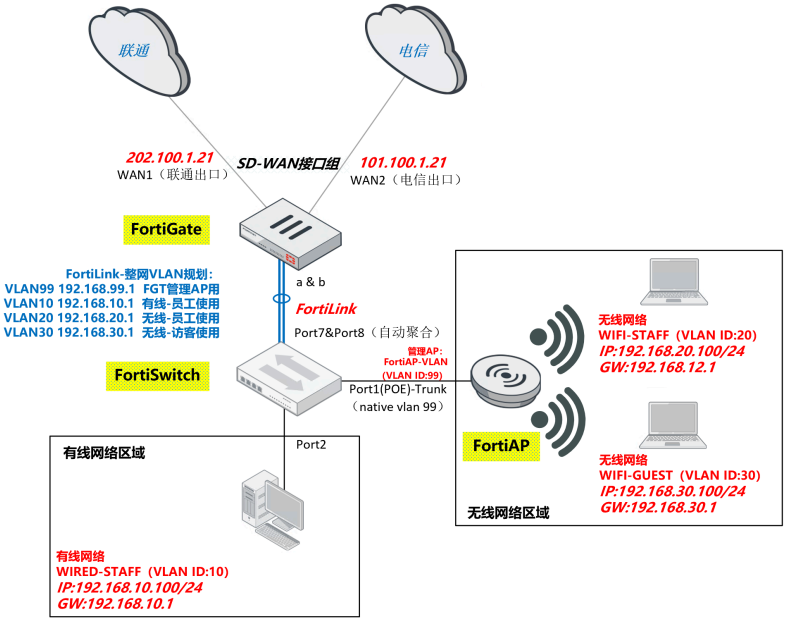
https://docs.fortinet.com/document/fortiswitch/7.0.8/devices-managed-by-fortios/ ↗

FortiLink 简要上线指南

网络需求

1. SAA（Secure Access Architecture）是 Fortinet 针对中小企业推出的有线无线一体的安全解决方案，包括有线无线设备的接入和认证，通过防火墙来统一管理网络中的有线和无线设备，并且有线交换机和无线 AP 都是即插即用，非常方便易用。
2. 如果再加入 SD-WAN 的部分可以组成叫 SD-WAN Branch 的方案。

网络拓扑



1. 一个最简单的 FortiLink 部署环境：一台防火墙 + 一台 FortiSwtich + 一台 FortiAP，通过 FortiLink 非常便捷的实现一个通过 FGT 统一管理的可视化的有线 + 无线的安全网络。
2. 整体规划：
   - 外网 IP 网段规划：
     - WAN1 联通出口：IP 202.100.1.21，GW 202.100.1.192
     - WAN2 电信出口：IP 101.100.1.21，GW 101.100.1.192
   - WAN1 和 WAN2 合并成 SD-WAN 接口。
   - 内网 IP 网段规划，FGT 通过 FortiLink + Capwap 协议统一的接管 FortiSW 和 FortiAP，为有线和无线统一进行管理和业务处理：
     - 有线网络规划：VLAN 10，192.168.10.1
     - 无线网络规划：员工 WIFI VLAN 20，192.168.20.1
     - 无线网络规划：访客 WIFI VLAN 30，192.168.30.1
     - 管理 FortiAP：VLAN 99，192.168.99.1
   - FortiLink 互联部分规划：
     - FortiGate 的 a 和 b 做聚合接口与 Fortiswitch 的后两个接口互联即可实现聚合接口的 FortiLink 管理。
     - 加入 FortiLink 的接口自动变为聚合口成员。

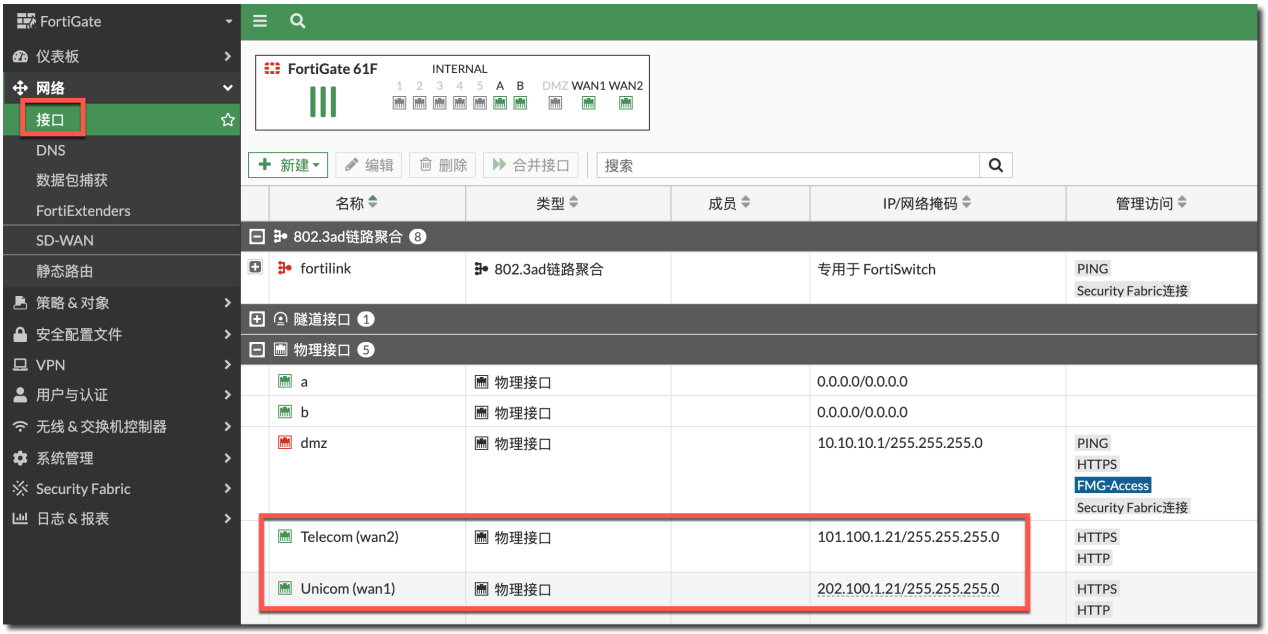
配置步骤

准备事项

1. 首先确认 FortiGate 如下全局配置已开启（一般默认为开启状态）。

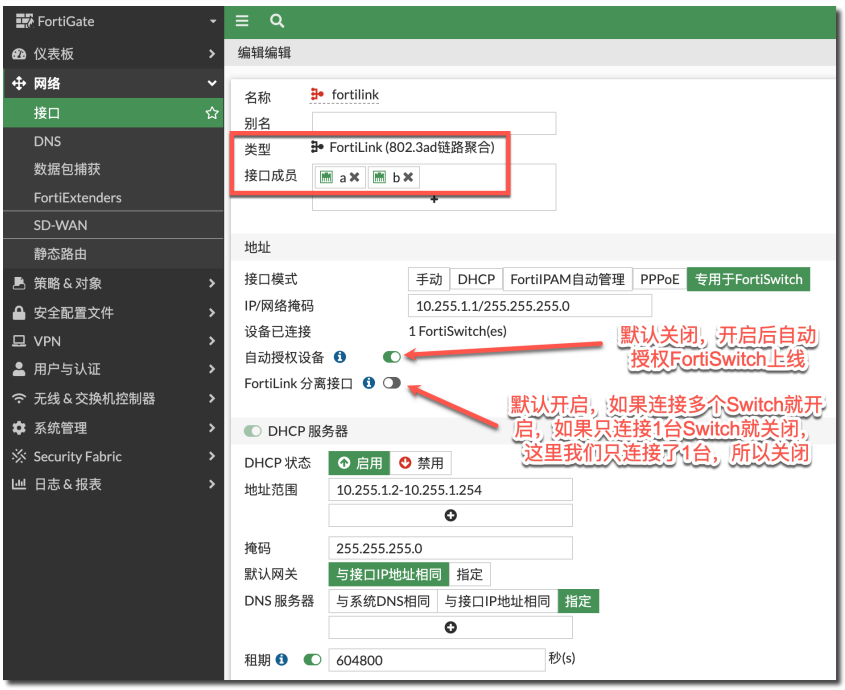
```
config system settings
    set gui-switch-controller enable
end
```

2. 配置 wan1 和 wan2 的 IP 地址。



3. 配置 FortiLink 实现 FortiSwith 上线，我们只需要将用于 FortiLink 的成员接口加入 FortiLink 接口即可。此拓扑中，FortiGate 用于连接 FortiSwitch 的接口为 a 和 b，编辑 FortiLink 接口，将 a 和 b 加入 FortiLink（某些产品型号会默认将两个接口加入 FortiLink），开启自动授权设备，关闭 FortiLink 分离接口，其他配置保持默认，可以看到 FortiLink 为一个聚合接口。


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
FortiGate 61F
INTERNAL
DMZ WAN1 WAN2
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 8
fortilink
802.3ad链路聚合
专用于 FortiSwitch
PING
Security Fabric连接
隧道接口 1
物理接口 5
a
物理接口
0.0.0.0/0.0.0.0
b
物理接口
0.0.0.0/0.0.0.0
dmz
物理接口
10.10.10.1/255.255.255.0
PING
HTTPS
FMG-Access
Security Fabric连接
Telecom (wan2)
物理接口
101.100.1.21/255.255.255.0
HTTPS
HTTP
Unicom (wan1)
物理接口
202.100.1.21/255.255.255.0
HTTPS
HTTP

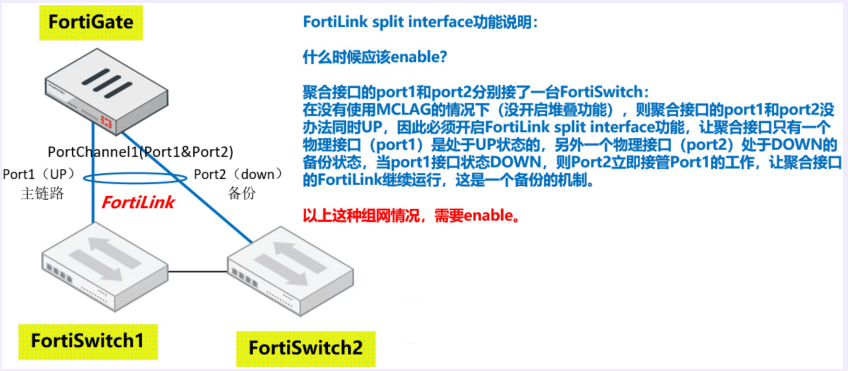

FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
编辑编辑
名称
fortilink
别名
类型
FortiLink (802.3ad链路聚合)
接口成员
a
b
地址
接口模式
手动
DHCP
FortiIPAM自动管理
PPPoE
专用于FortiSwitch
IP/网络掩码
10.255.1.1/255.255.255.0
设备已连接
1 FortiSwitch(es)
自动授权设备
默认关闭，开启后自动授权FortiSwitch上线
FortiLink 分离接口
默认开启，如果连接多个Switch就开启，如果只连接1台Switch就关闭，这里我们只连接了1台，所以关闭
DHCP 服务器
DHCP 状态
启用
禁用
地址范围
10.255.1.2-10.255.1.254
掩码
255.255.255.0
默认网关
与接口IP地址相同
指定
DNS 服务器
与系统DNS相同
与接口IP地址相同
指定
租期
604800
秒(s)


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
FortiGate 61F
INTERNAL
DMZ WAN1 WAN2
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 8
fortilink
802.3ad链路聚合
a
b
专用于 FortiSwitch
PING
Security Fabric连接
隧道接口 1
物理接口 3
dmz
物理接口
10.10.10.1/255.255.255.0
PING
HTTPS
FMG-Access
Security Fabric连接
Telecom (wan2)
物理接口
101.100.1.21/255.255.255.0
HTTPS
HTTP
Unicom (wan1)
物理接口
202.100.1.21/255.255.255.0
HTTPS
HTTP

重要

FortiLink 分离接口什么时候应该开启?

分离接口的意是，如果聚合接口的 port1 和 port2 分别连接了一台 SW，在没有使用 MCLAG 的情况下（没开启堆叠功能），则聚合接口的 port1 和 port2 没办法同时 UP（同时开启聚合接口无法正常工作），因此必须开启 FortiLink 分离接口功能，让聚合接口只有一个物理接口（port1）是处于 UP 状态的，另外一个物理接口（port2）处于 DOWN 的备份状态，当 port1 接口状态 DOWN，则 Port2 立即接管 Port1 的工作，让聚合接口的 FortiLink 继续运行，这是一个聚合接口在 FortiLink 管理时候的备份机制。



重要

FortiLink split interface 什么时候应该 disable?

如果 FGT 的聚合接口只接到了一台交换机上，并没有跨交换机聚合（没有 MCLAG，也不需要 FortiLink split interface），因此这种情况应该让聚合接口的两个物理接口 port1 和 port2 同时 UP，两个接口同时工作起来，以便提供更高带宽和备份的作用。



相关 CLI:

```
NTP:
config system ntp
    set ntpsync enable
    set server-mode enable
    set interface "fortilink"
end
-------------------------------------------------------------------

DHCP:
config system dhcp server
    edit 2
        set dns-service local
        set ntp-service local
        set default-gateway 10.255.1.1
        set netmask 255.255.255.0
        set interface "fortilink"
        config ip-range
            edit 1
                set start-ip 10.255.1.2
                set end-ip 10.255.1.254
            next
        end
        set vci-match enable
        set vci-string "FortiSwitch" "FortiExtender"
    next
```

```
end
————————————————————————————————————————————
FortiLink:
config system interface
    edit "fortilink"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set type aggregate
        set member "a" "b"
        set lldp-reception enable
        set lldp-transmission enable
        set auto-auth-extension-device enable
        set fortilink-split-interface disable
        set switch-controller-nac "fortilink"
        set switch-controller-dynamic "fortilink"
        set swc-first-create 255
    next
end
```

连接 FortiLink

1. 配置完 FortiGate 之后，这个时候再去接 FortiGate 的 FortiLink 到 FortiSwitch 的线，以及 FortiSwitch 之间的线，严格按照拓扑图，从核心交换机往接入交换机这样的顺序接线。FortiSwitch 如果以前有使用过，建议都恢复一下出厂设置，然后再加入到 FortiLink。
2. FortiSwitch 的靠后接口一般都默认开启了 FortiLink 接口属性，如果接前面的接口可能会无法加入到 FortiLink，因此 FortiSwitch 的 FortiLink 最好都使用靠后的接口进行接线。FortiSwitch 各型号默认开启 FortiLink 功能的接口如下：

FortiSwitch 型号	默认开启 FortiLink 的接口
FS-108D-POE	port9–port10
FS-108E, FS-108E-POE, FS-108E-FPOE	port7–port10
FSR-112D-POE	port5–port12
FS-124D, FS-124D-POE	port23–port26
FSR-124D	port1-port4, port21–port28
FS-124E, FS-124E-POE, FS-124E-FPOE	port21–port28
FS-148E, FS-148E-POE	port21–port52
FS-224D-POE	port21–port24
FS-224D-FPOE	port21–port28
FS-224E, FS-224E-POE	port21–port28
FS-248D, FS-248D-FPOE	port45–port52
FS-248D-POE	port47–port50
FS-248E-POE, FS-248E-FPOE	port45–port52
FS-424D, FS-424D-POE, FS-424D-FPOE	port23–port26
FS-424E-Fiber	port1-port30
FS-426E-FPOE-MG	port23-port30
FS-448D, FS-448D-POE, FS-448D-FPOE	port45–port52
FS-524D, FS-524D-FPOE	port21–port30
FS-548D	port39–port54
FS-548D-FPOE, FS-548DN	port45–port54
FS-1024D	port1–port24
FS-1048D, FS-1048E	port1–port52
FS-3032D, FS-3032E	port1–port32

3. 如果 FortiSwitch 无法被 FortiGate 管理，登陆到 FortiSwitch 上进行确认接口是否开启了 FortiLink。

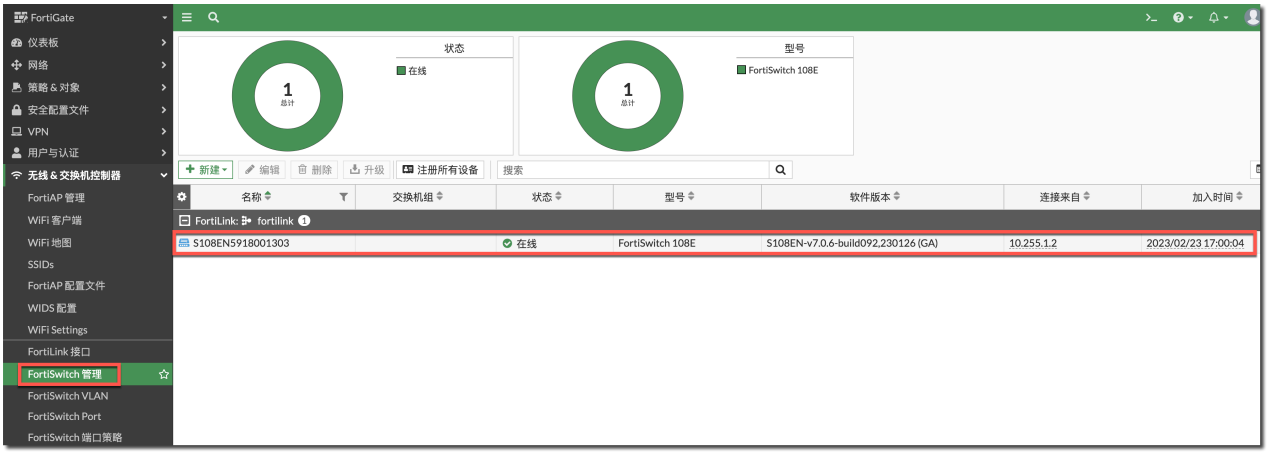
```
FortiSwitch # show switch interface port9
config switch interface
    edit "port9
        set auto-discovery-fortilink enable    //接口下又此命令的接口才可以自动加入到FortiLink中
    next
end
```

4. 交换机之间如果接多根线，会自动进行聚合，但是 FortiLink 自动聚合会花比较长的时间，建议可以先接单线，等 FortiLink 的拓扑较快的形成结束之后，再增加彼此之间的另外一根线，再进行自动聚合，这样效率会比较高一些。

5. 接线是很重要的步骤，仔细看拓扑图，做好接口标签，不要接错。

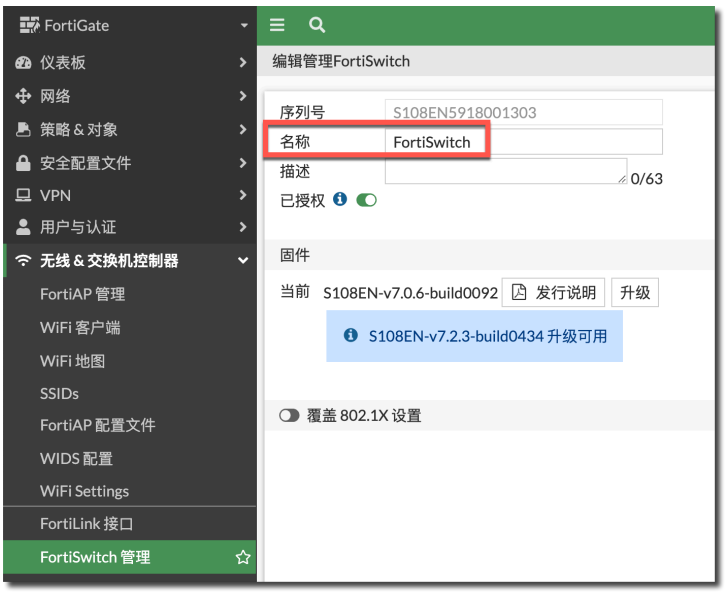
6. 接好线之后，FortiSwitch 和 FortiGate 之间会自动在后台进行 FortiLink、CAPWAP、MSTP、聚合接口的协商加入等操作，等 FortiLink 的管理结束之后，在 FortiGate 上会看到一个网站的 FortiSwitch 拓扑图，和我们规划的拓扑图应该是一致的，可以仔细核对接线是否正确。

7. 在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，查看交换机被管理状态。



8. 交换机加入到 FortiGate 的 FortiLink，会先清除配置再重启，然后从 standalone 的状态进入到“Fortilink remote control”的状态，后续所有的配置都通过 FortiGate 下发即可。

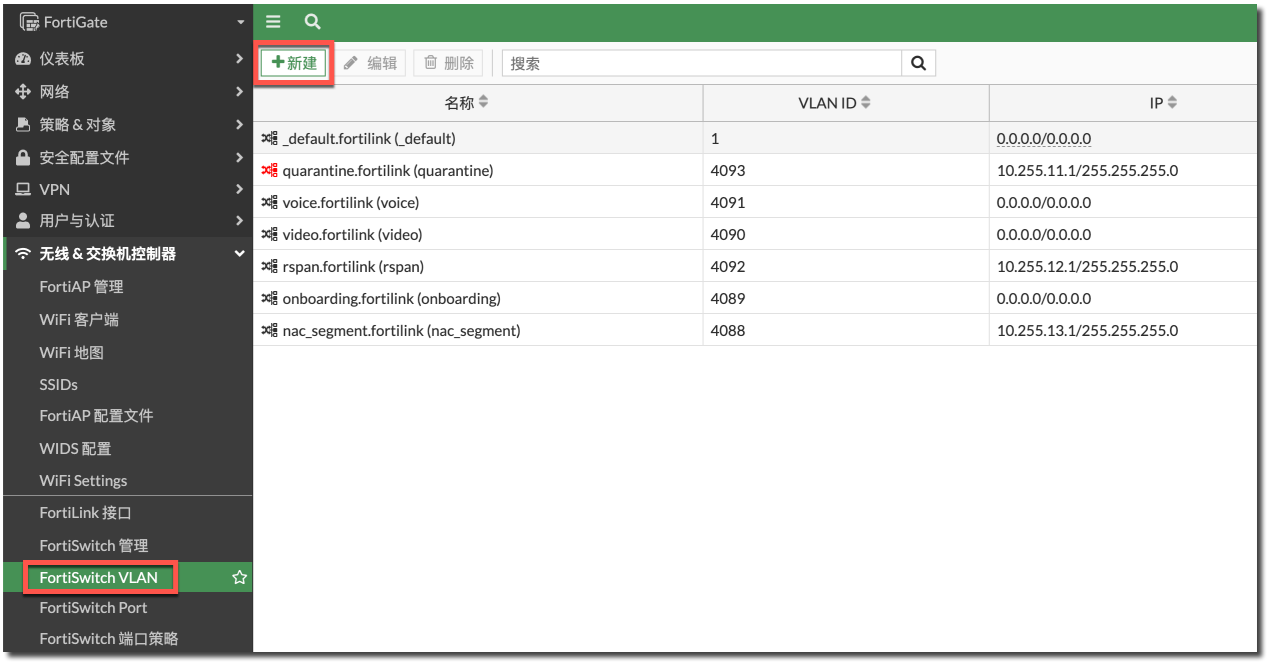
9. 编辑该交换机，可以修改交换机名字以及升级版本，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。

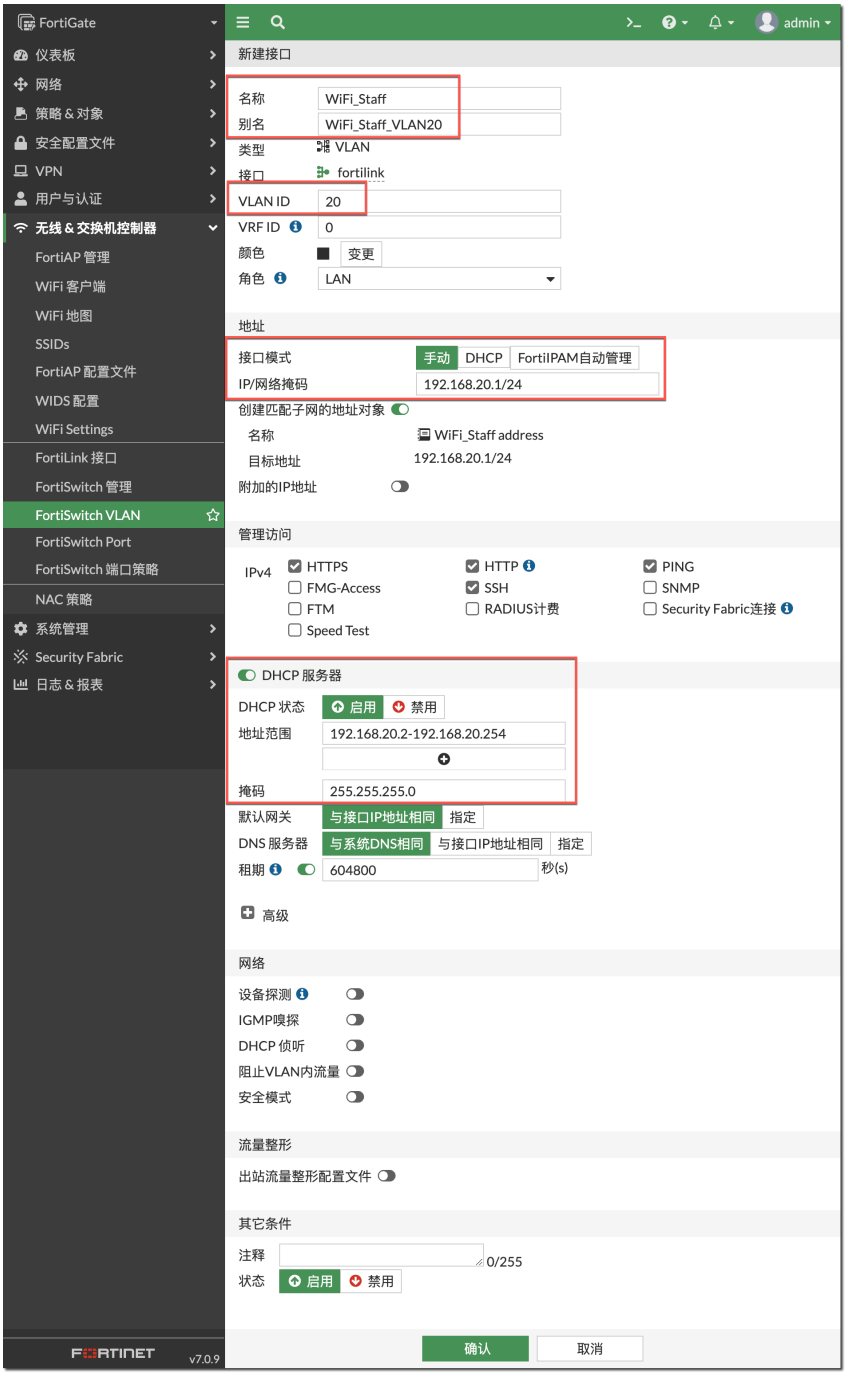


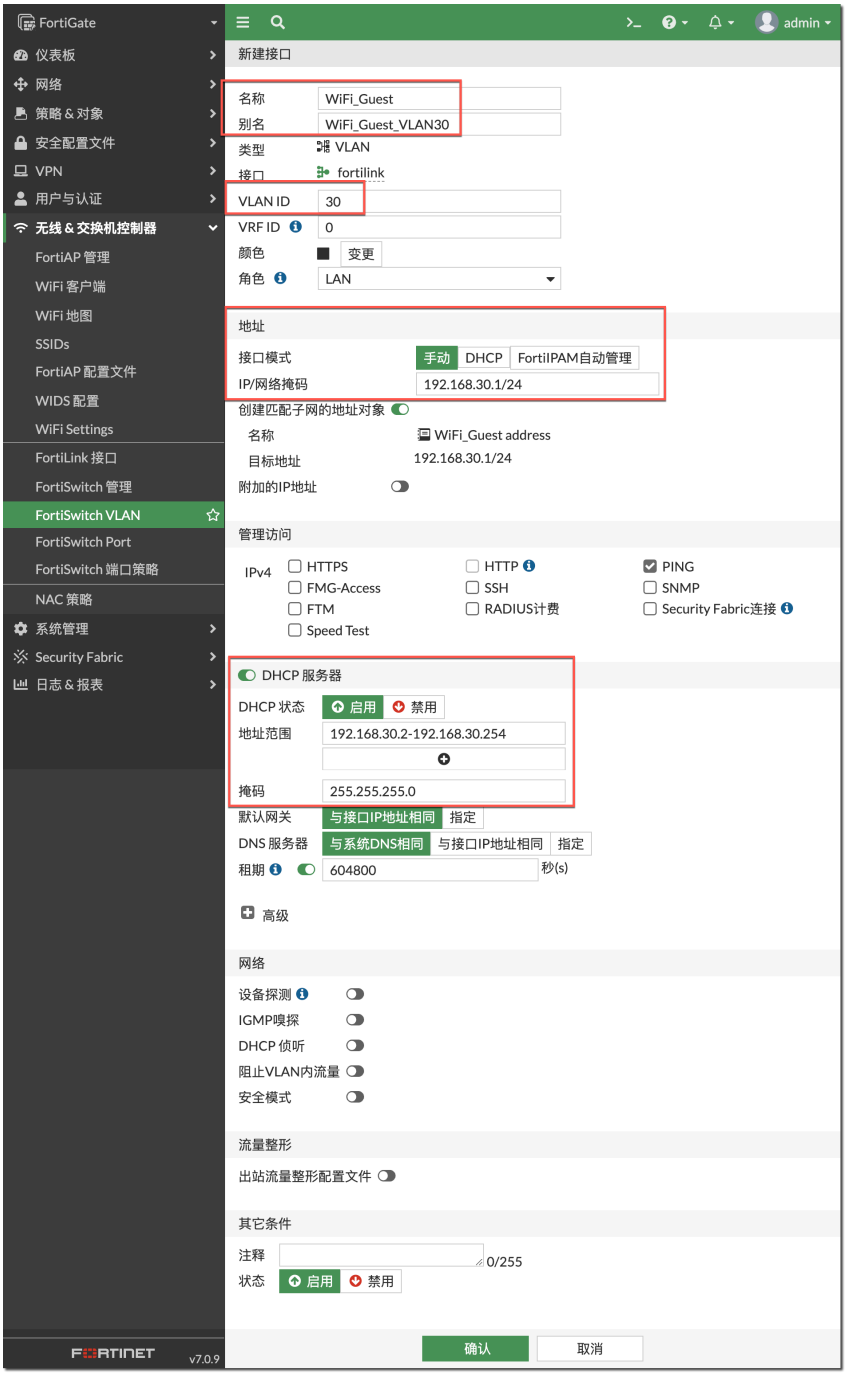
创建 VLAN

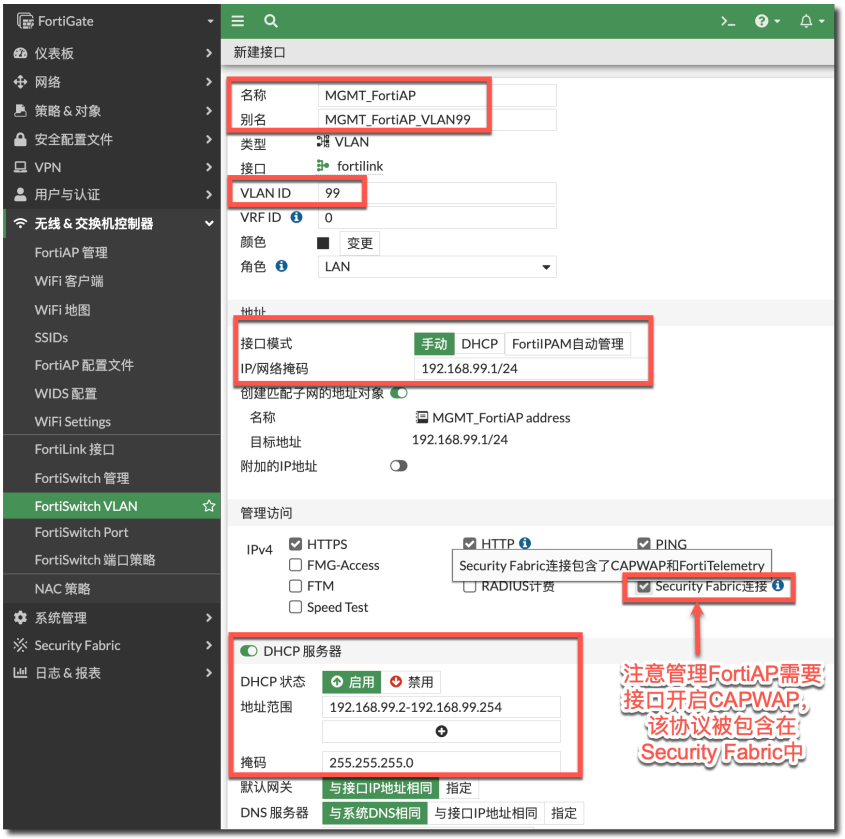
1. FortiGate 通过 FortiLink + Capwap 协议统一的接管 FortiSwitch 和 FortiAP，为有线和无线统一进行管理和业务处理。

2. 进入 FortiGate 的无线 & 交换机控制器 → FortiSwitch VLAN 页面，新建有线网络 VLAN 10、员工无线网络 VLAN 20、访客无线网络 VLAN 30、管理 FortiAP VLAN 99。注意管理 FortiAP 的 VLAN 接口要开启 Security Fabric，包含 CAPWAP。

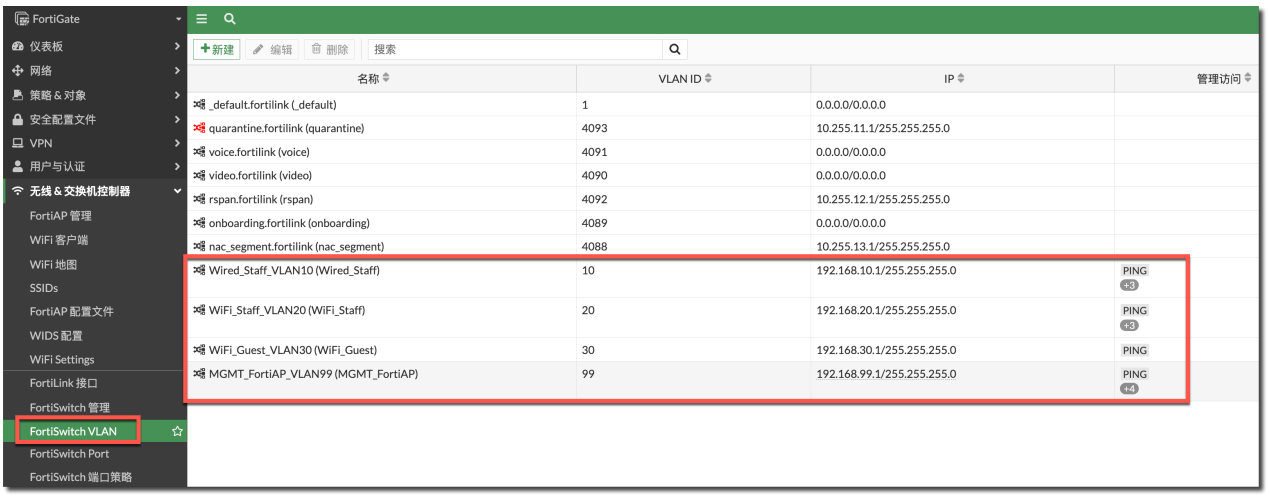

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
+新建
编辑
删除
搜索
名称
VLAN ID
IP
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 WiFi_Staff
别名 WiFi_Staff_VLAN20
类型 VLAN
接口 fortilink
VLAN ID 20
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.20.1/24
创建匹配子网的地址对象
名称 WiFi_Staff address
目标地址 192.168.20.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.20.2-192.168.20.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
新建接口
名称 WiFi_Guest
别名 WiFi_Guest_VLAN30
类型 VLAN
接口 fortilink
VLAN ID 30
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.30.1/24
创建匹配子网的地址对象
名称 WiFi_Guest address
目标地址 192.168.30.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.30.2-192.168.30.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消
FORTINET
v7.0.9



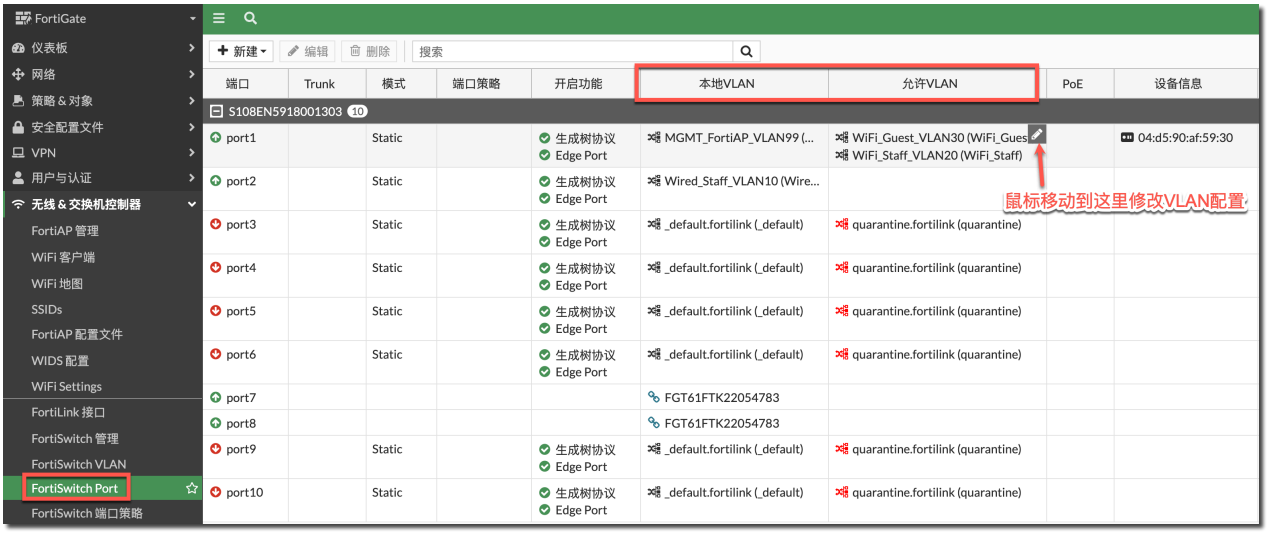
3. VLAN 创建完成。



分配 VLAN

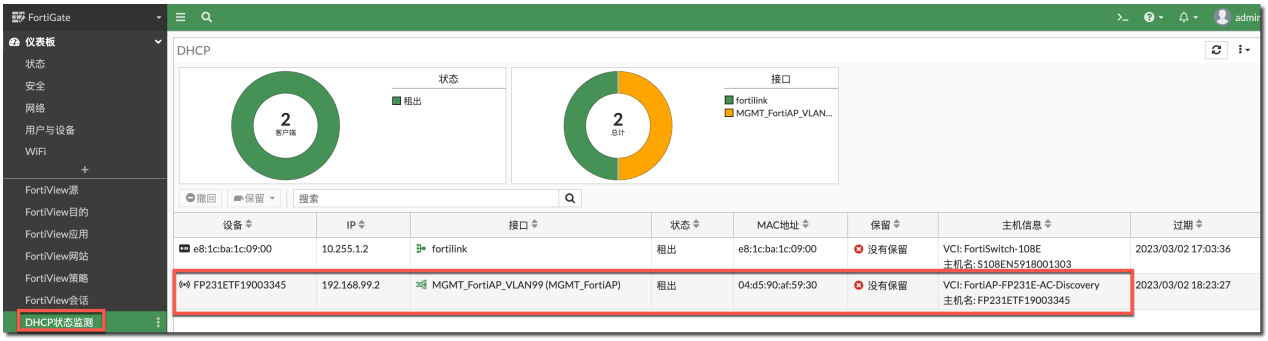
1. 交换机的 port1 接 FortiAP，配置为 Trunk 模式，允许 VLAN20 和 VLAN30，VLAN20 用于 WIFI - STAFF 的 SSID 用户，VLAN30 用于 WIFI - GUEST 的 SSID 用户。另外本地 VLAN 设置为 VLAN 99，用于 FortiAP 的管理（FortiAP 的管理相当于使用不打 vlan-tag 的 VLAN 99，本地 VLAN 默认是 VLAN 1，因此只需要将本地 VLAN 设置为 VLAN 99 即可）。

2. Port2 接具体的物理 PC，因此只需要将 Port2 设置为 access 口，分配 VLAN 10 即可，用 FortiSwitch 的概念来说就是将本地 VLAN 设置为 VLAN 10，本地 VLAN 不打 vlan-tag，相当于就是设置接口为 access vlan 10。

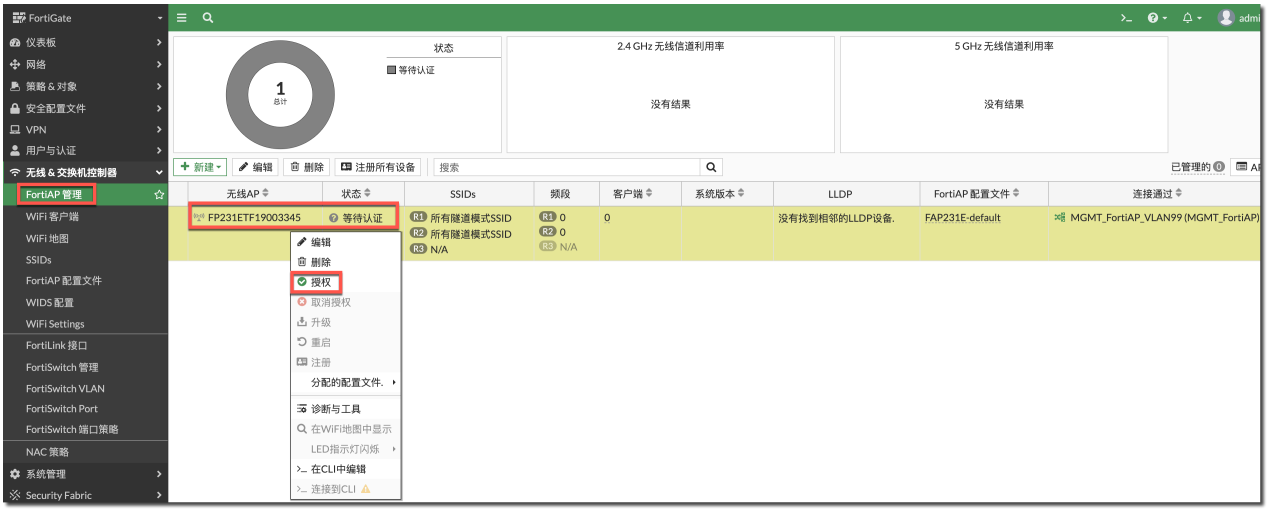


无线配置

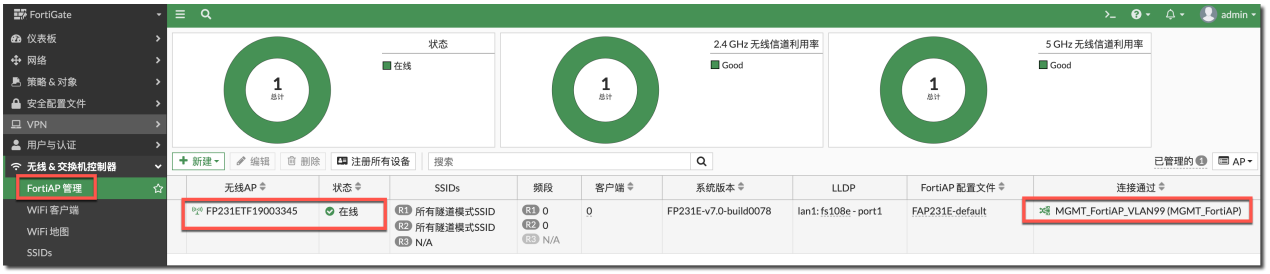
1. FortiAP 连接后，会通过 VLAN 99 的 DHCP 获取到 192.168.99.0/24 网段的 IP 地址。



2. 在之前的配置中，VLAN 99 的接口开启了 CAPWAP 协议（包含在 Security Fabric 中），激活了 FortiGate 的无线管理功能，这时候 FortiAP 会自动找到 FGT 进行注册。因此在 FortiAP 管理的地方可以看到这个 FortiAP 的信息（如果没有看到检查配置，并将 FortiAP 恢复出厂，让 FortiAP 重新获取 IP 地址并发起注册）。



3. 右键点击上图中等待认证的 FortiAP，点击授权按钮，等待几分钟，FortiAP 成功上线。

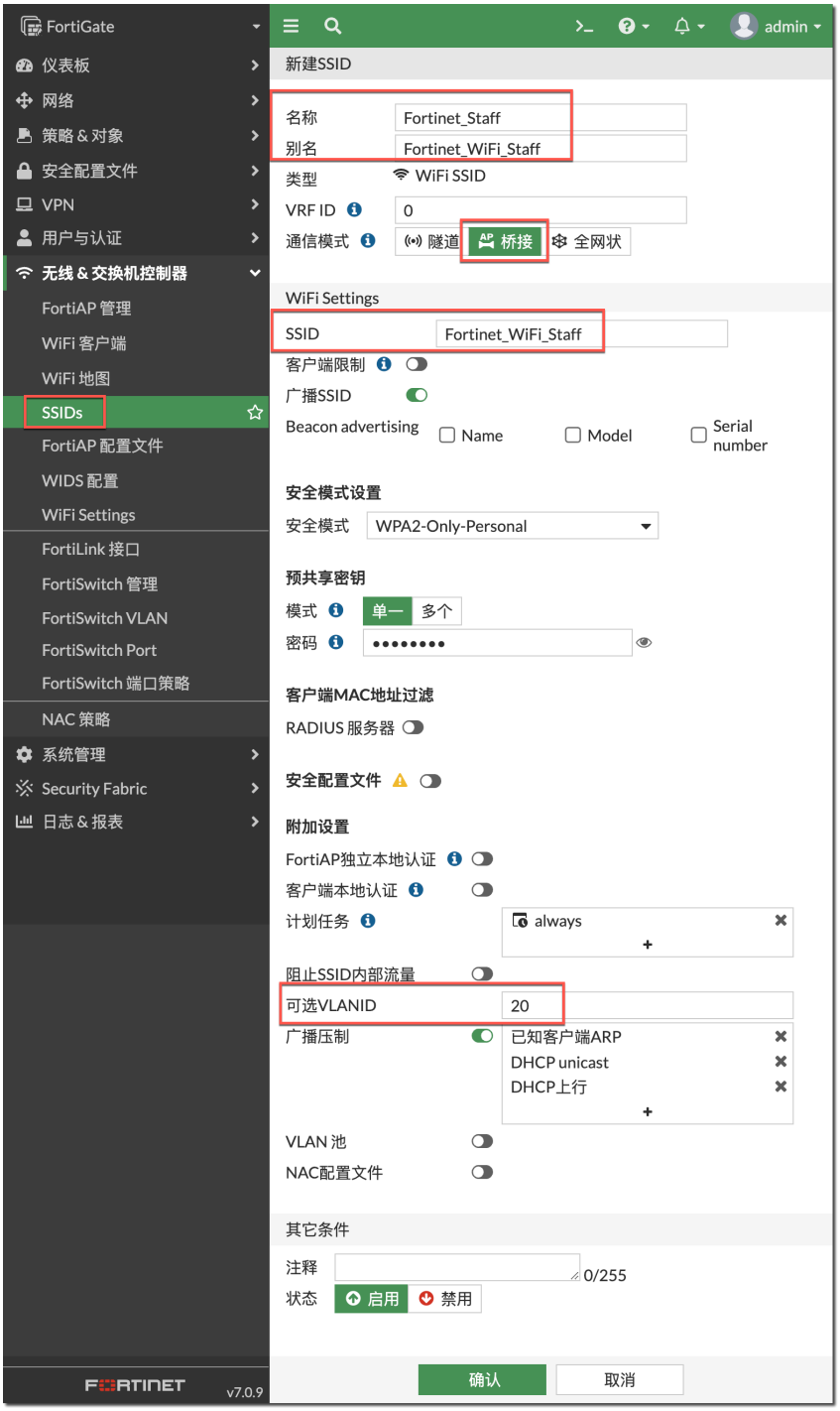


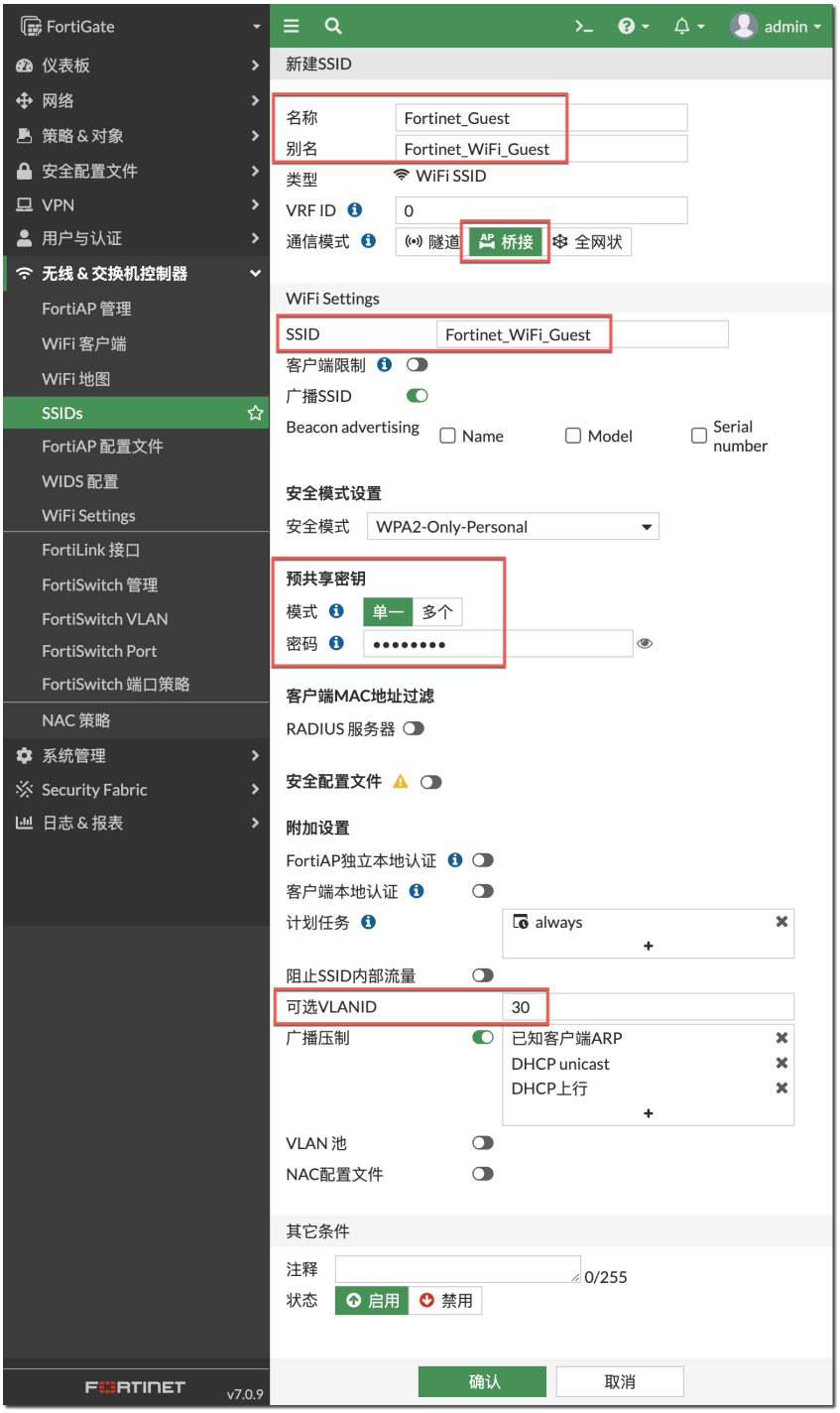
4. 编辑 FortiAP 设置，配置 FortiAP 的名称，或进行升级操作。

5. 进入无线 & 交换机控制器 → SSIDs 页面，点击新建按钮。



6. 新建 SSID：Fortinet_WiFi_Staff VLAN 20 用于员工 WIFI 使用，Fortinet_WiFi_Guest VLAN 30 用于访客 WiFi 使用。


FortiGate
admin
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
新建SSID
名称
Fortinet_Staff
别名
Fortinet_WiFi_Staff
类型
WiFi SSID
VRF ID
0
通信模式
隧道
桥接
全网状
WiFi Settings
SSID
Fortinet_WiFi_Staff
客户端限制
广播SSID
Beacon advertising
Name
Model
Serial number
安全模式设置
安全模式
WPA2-Only-Personal
预共享密钥
模式
单一
多个
密码
客户端MAC地址过滤
RADIUS 服务器
安全配置文件
附加设置
FortiAP独立本地认证
客户端本地认证
计划任务
always
阻止SSID内部流量
可选VLANID
20
广播压制
已知客户端ARP
DHCP unicast
DHCP上行
VLAN 池
NAC配置文件
其它条件
注释
0/255
状态
启用
禁用
确认
取消

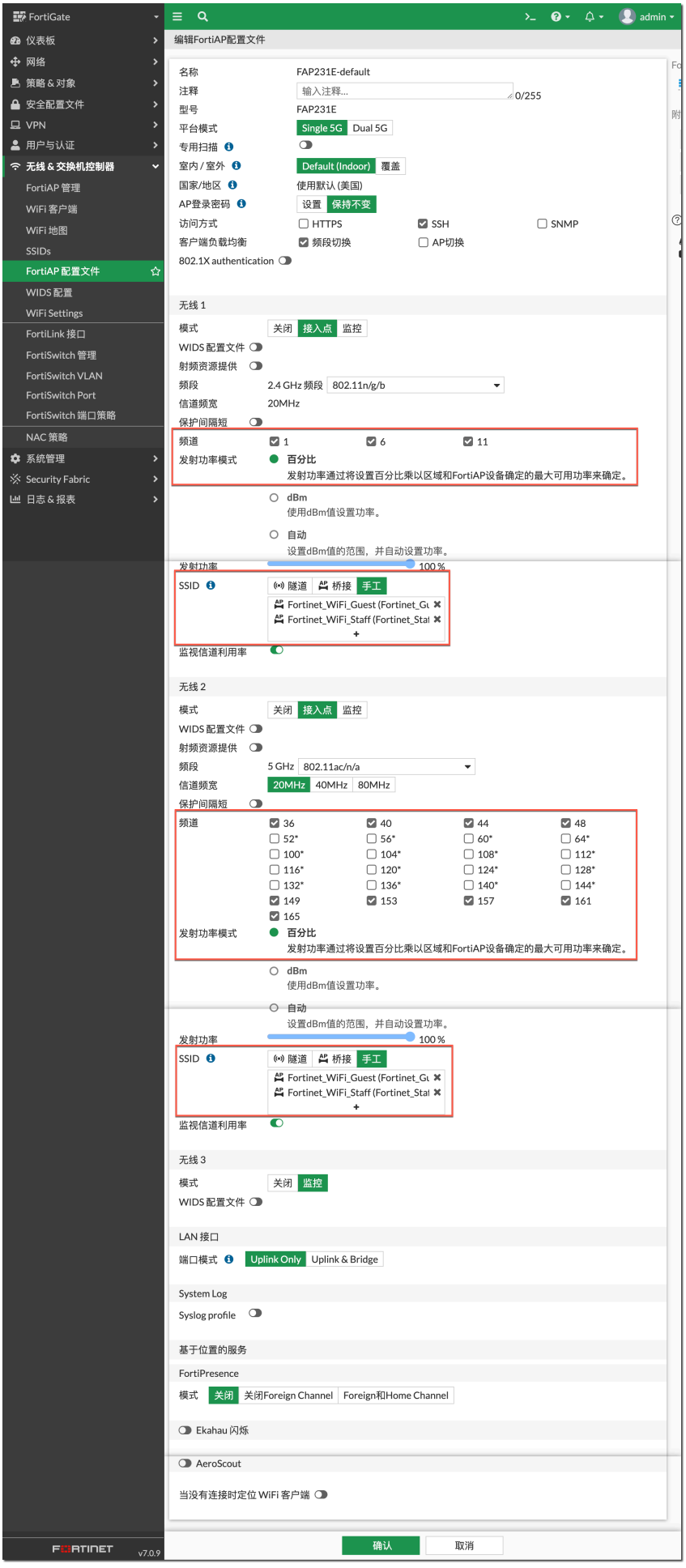




7. 将 SSID 关联到 FortiAP 上，然后让 FortiAP 发布无线 SSID 信号，这样用户就可以通过 SSID 连接到无线了。进入 FortiAP 配置文件页面，编辑 FortiAP 关联的配置文件。



8. 桥模式的 SSID 需要在 FortiAP 配置文件里面手工指定一下，另外下图标示出的配置为推荐配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
编辑FortiAP配置文件
名称 FAP231E-default
注释 输入注释... 0/255
型号 FAP231E
平台模式 Single 5G Dual 5G
专用扫描
室内 / 室外 Default (Indoor) 覆盖
国家/地区 使用默认 (美国)
AP登录密码 设置 保持不变
访问方式 HTTPS SSH SNMP
客户端负载均衡 频段切换 AP切换
802.1X authentication
无线 1
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 2.4 GHz 频段 802.11n/g/b
信道频宽 20MHz
保护间隔短
频道 1 6 11
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 2
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 5 GHz 802.11ac/n/a
信道频宽 20MHz 40MHz 80MHz
保护间隔短
频道 36 40 44 48
52* 56* 60* 64*
100* 104* 108* 112*
116* 120* 124* 128*
132* 136* 140* 144*
149 153 157 161
165
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 3
模式 关闭 监控
WIDS 配置文件
LAN 接口
端口模式 Uplink Only Uplink & Bridge
System Log
Syslog profile
基于位置的服务
FortiPresence
模式 关闭 关闭Foreign Channel Foreign和Home Channel
Ekahau 闪烁
AeroScout
当没有连接时定位 WiFi 客户端
确认
取消
FORTINET
v7.0.9

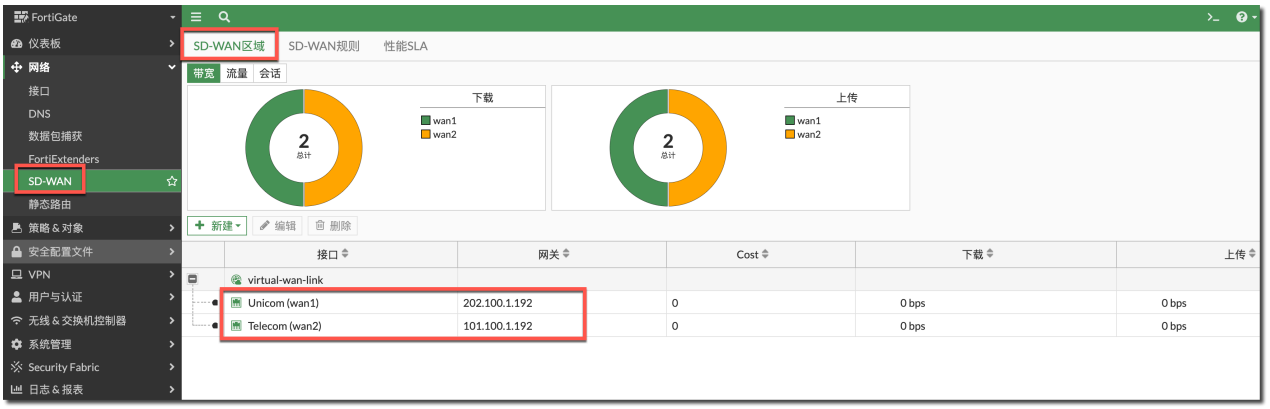


9. FortiAP 会自动关联到这个 FortiAP 配置文件里面的内容，发起 SSID 的信号。



SD-WAN 配置

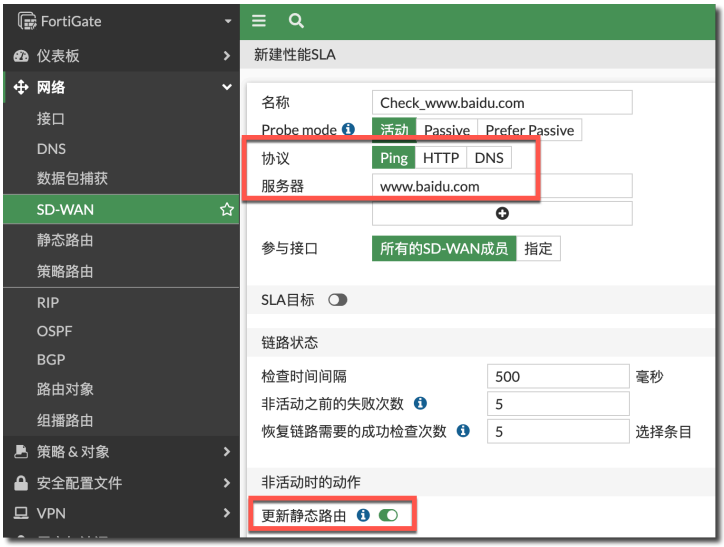
1. 将 wan1 和 wan2 加入 SD-WAN 接口，并配置各自的网关。

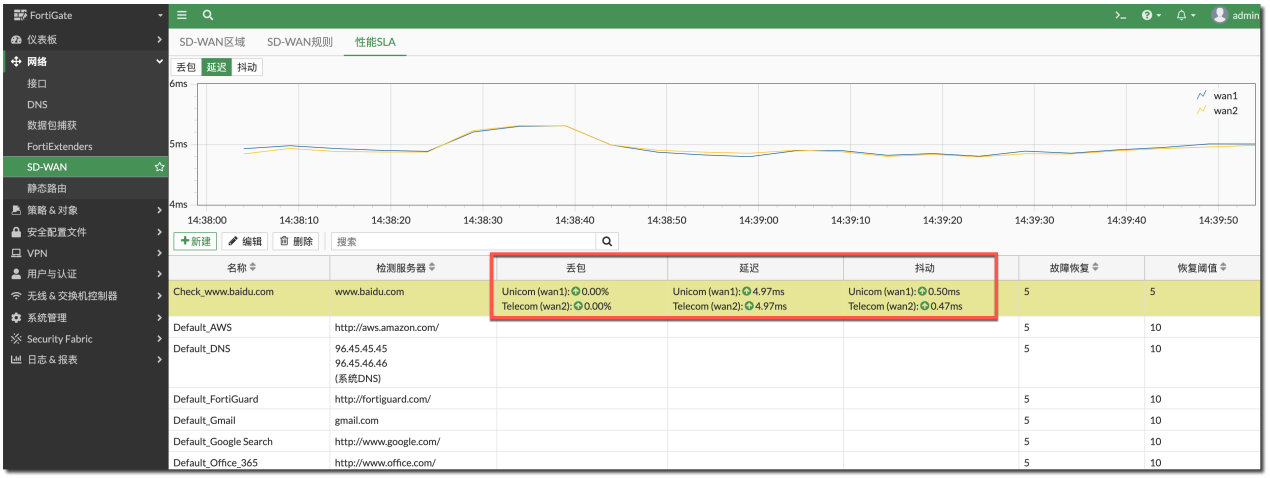


2. 添加 SD-WAN 的默认路由。



3. 配置 SD-WAN 的 SLA，以便 wan1 和 wan2 相互备份和切换。



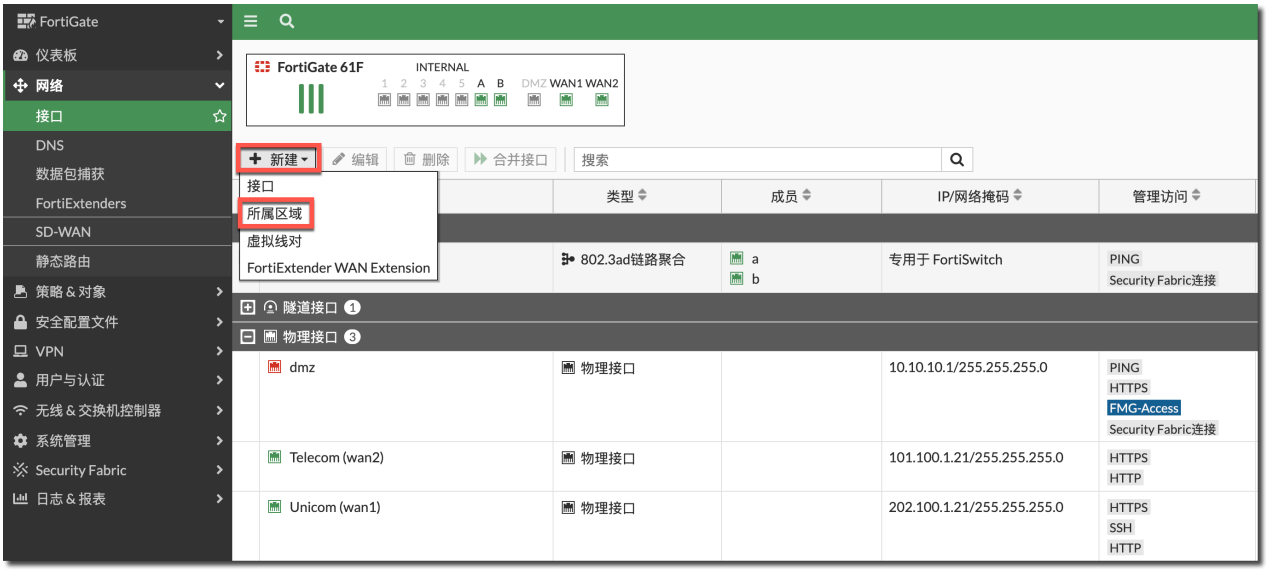


4. SD-WAN 规则可根据需求配置，这里保持默认的按照源 IP 负载均衡。

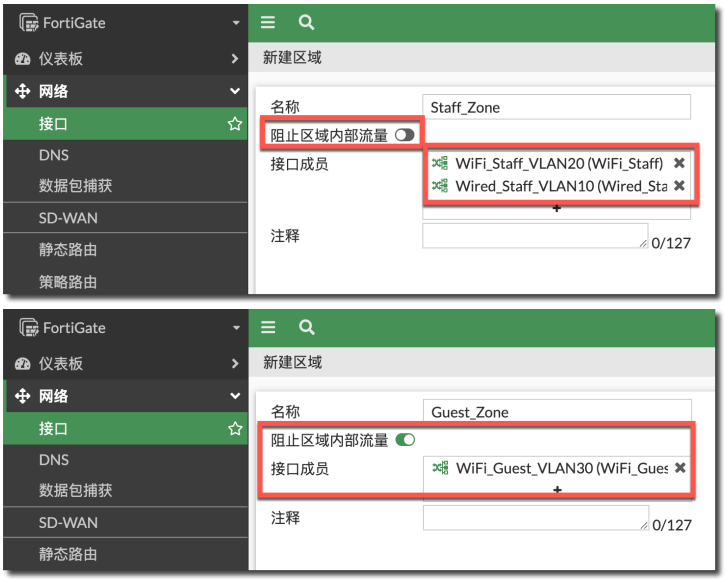


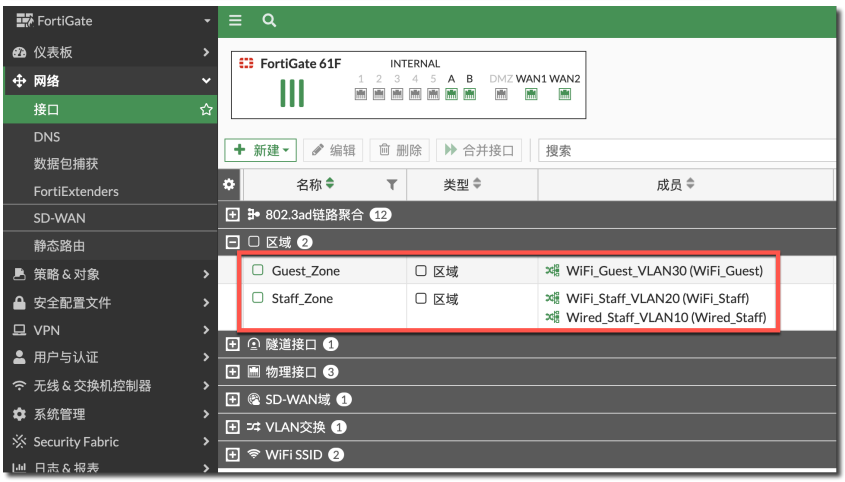
安全策略配置

1. 需求与规划：
   - 员工的有线与无线都需要访问互联网
   - 访客的无线仅仅可以访问互联网
   - 员工的有线和无线需要互访
2. 建议将防火墙的接口划分区域，以区域的方式配置策略，有时候可以做到简化配置的目的，甚至未来替换设备的时候，策略的备份和还原也会更加简单一些。规划区域两个区域：
   - 员工网络区域：包括员工的有线 VLAN 10 和员工的无线 VLAN 20 （区域之间的隔离关闭，这样 VLAN 10 和 VLAN 20 之间的数据可通）
   - 访问网络区域：包括访问的无线 VLAN 30
3. 进入网络 → 接口页面，新建所属区域。



4. 分别创建 Staff_Zone（包含 VLAN 10 和 VLAN20）和 Guest_Zone（包含 VLAN30）。





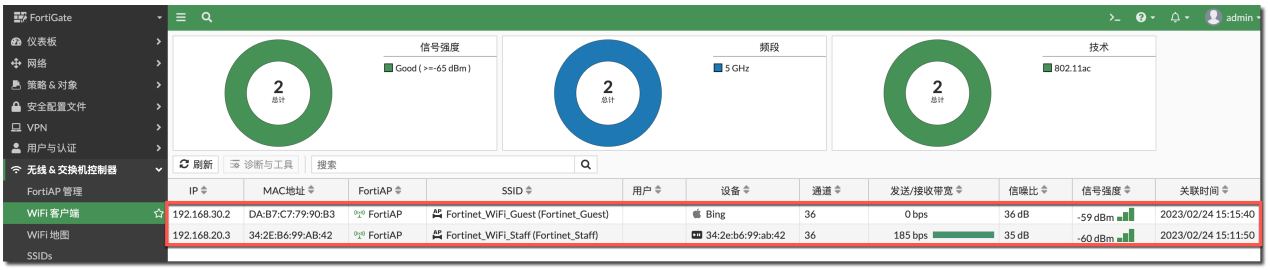
5. 如此只需要配置两条策略，就可以实现我们规划的需求，新建 Staff-Zone 到 SD-WAN 与 Guest-Zone 到 SD-WAN 的安全策略。



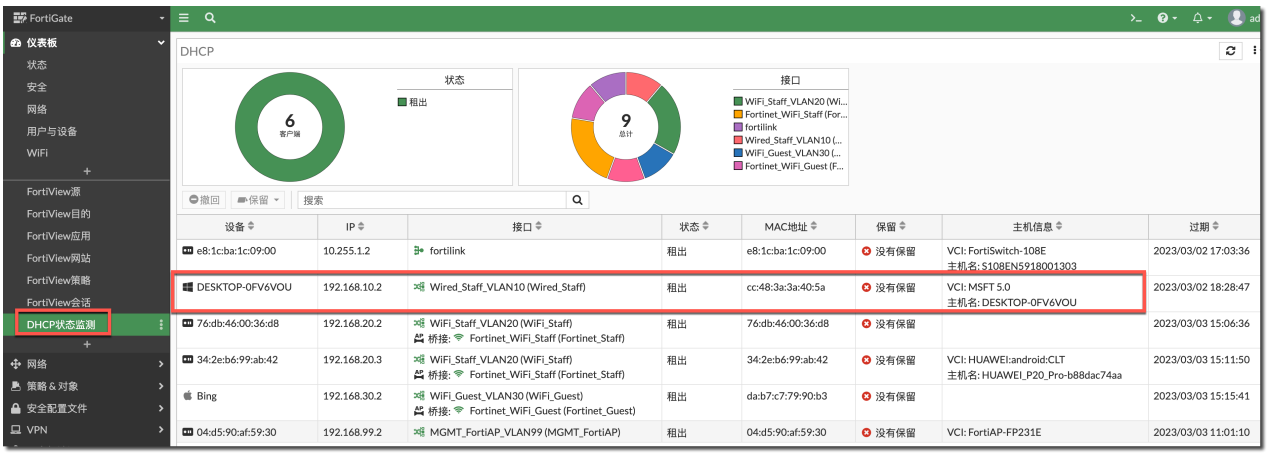
结果验证

客户端访问

1. 进入无线 & 交换机控制器 → WiFi 客户端，查看在线的无线客户端。可以看到 WiFi_Staff 和 WiFi_Guest 各有一个终端在线。



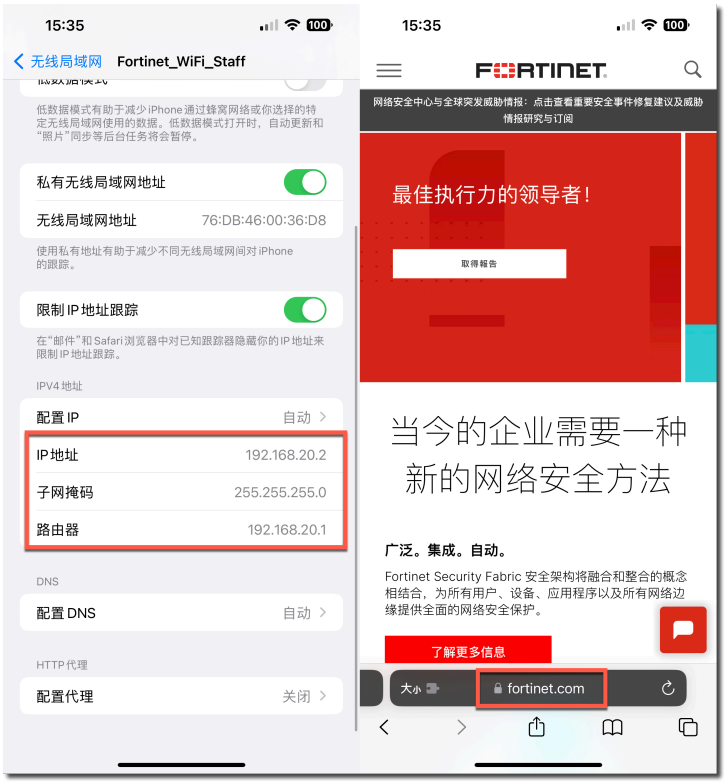
2. 在 DHCP 状态检测中查看所有的客户端信息，可以看到有线网络的 PC 也正常上线。



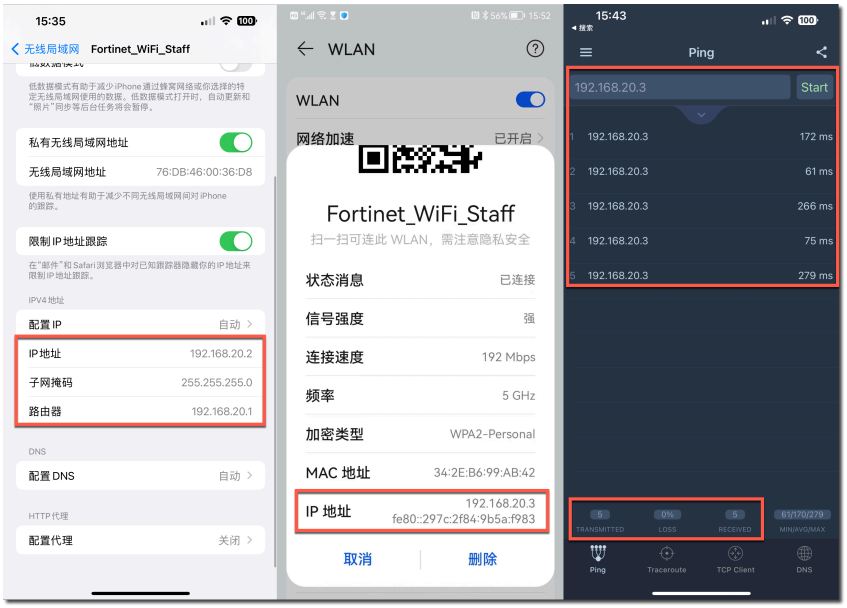
3. 有线网络（VLAN 10）用户访问互联网。



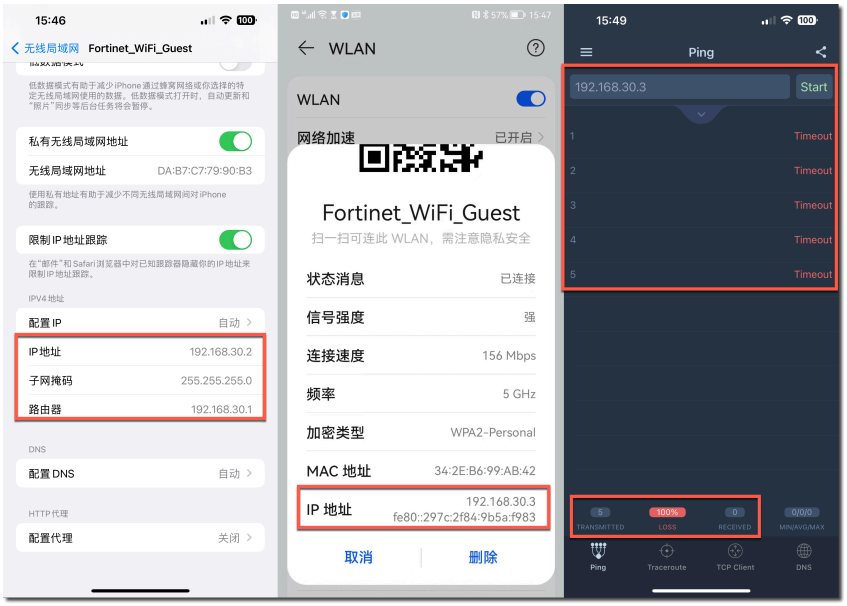
4. 无线网络用户访问互联网。



5. 使用 VLAN 20 下的客户端访问 VLAN 20 下的另一个客户端（Staff 区域）。



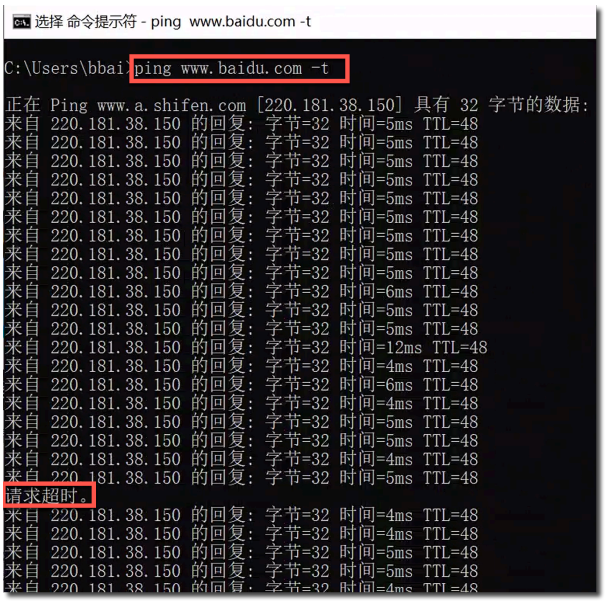
6. 使用 VLAN 30 下的客户端访问 VLAN 30 下的另一个客户端（Guest 区域），可以看到由于该区域配置了禁止相互访问，所以 VLAN 30 下的客户端是无法互相访问的。



切换测试

1. 在我们的测试场景下 FortiLink 聚合接口的 a 和 b 是同时 UP 起来的，如果 a 或 b DOWN（不能同时 DOWN），是否有影响业务呢?

2. 经过测试，DOWN a 或 b 接口，客户端丢包 1 个或者不丢包。



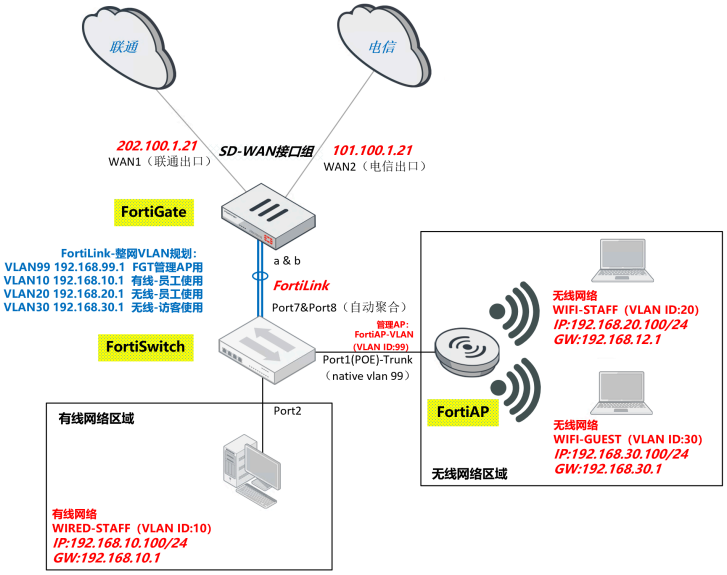
FortiLink 部署方案和举例

- FortiLink 星型组网
- FortiLink 星型组网配置举例
- FortiLink 环形组网
- FortiLink 环形组网配置举例
- FortiLink 星型环形混合组网
- FortiLink 交换机零配置跨三层部署
- 交换机零配置跨二层部署

FortiLink 星型组网

通过简要上线指南中的拓扑，来说明星型组网部署方案。

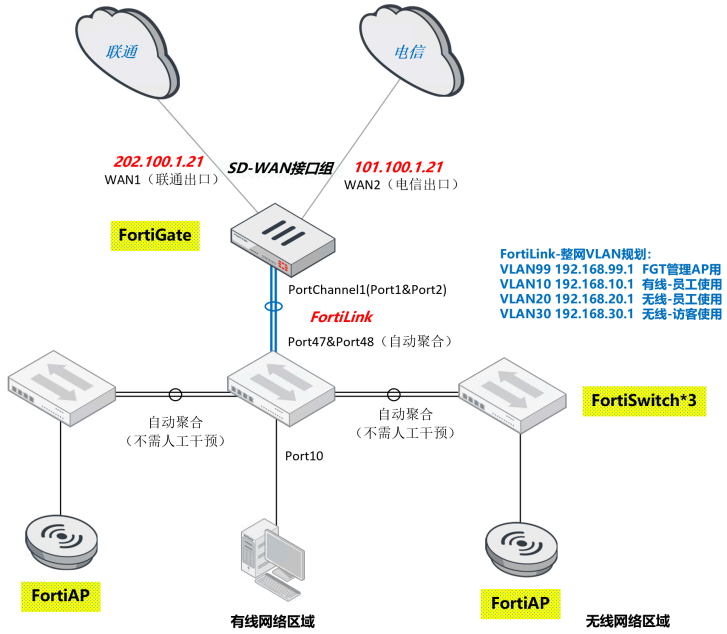
简要上线指南拓扑



星型拓扑

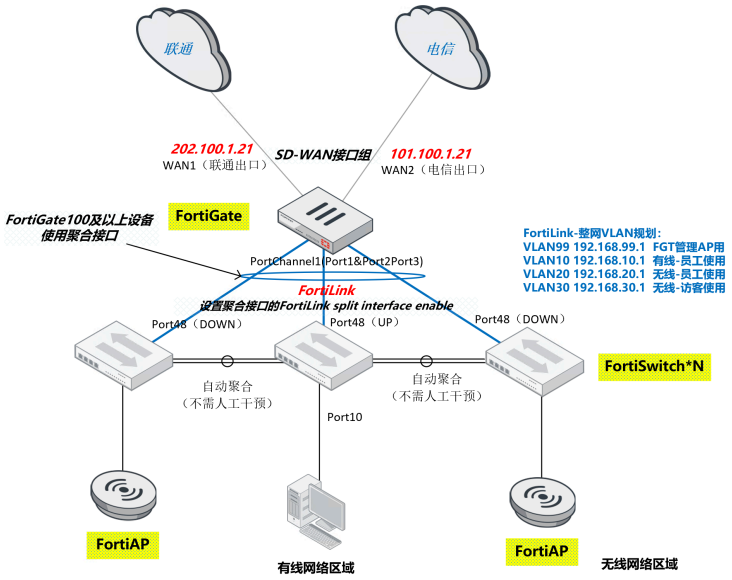
改造方案一

1. 接入交换机直接串联到核心交换机上的 FortiLink 部署。
2. 这种方案比较简洁，容易理解，存在的问题：核心交换机存在单点故障风险，一旦核心交换机出现故障（比如断电），整网将全部中断。

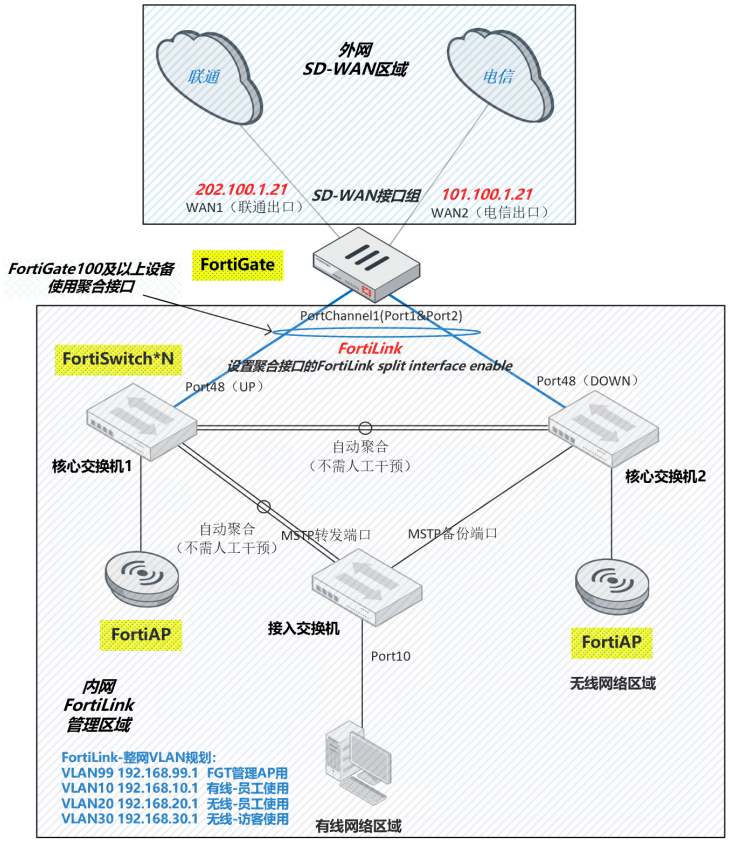


改造方案二

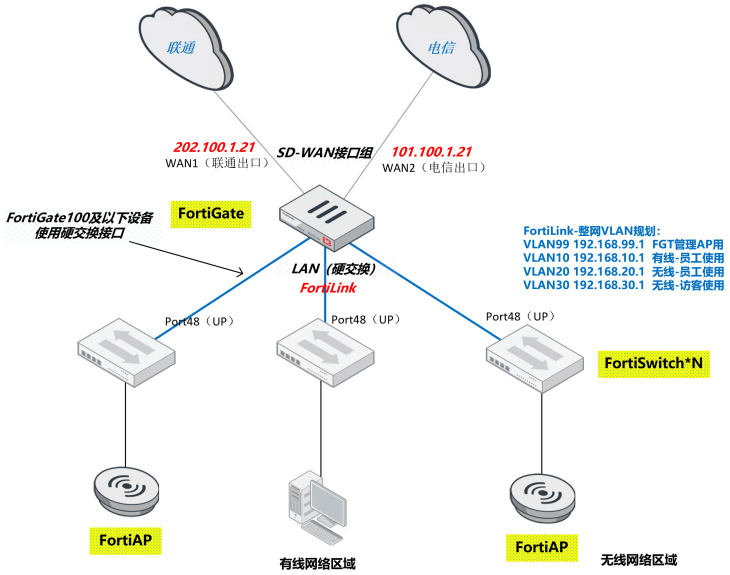
1. 交换机串联，FortiGate 使用“FortiLink split interface”特性互联到每一台 FortiSwitch 核心交换机上。
2. 使用聚合接口并利用“FortiLink split interface”，将防火墙的聚合接口分别接到三台交换机上，但是正常情况下只有接核心交换机的接口是 UP 的，其他的接口处于备份状态。



3. 如果交换机很多，可以使用以下方案拓扑。使用聚合接口并利用“FortiLink split interface”，将防火墙的聚合接口分别接到两台核心交换机上，但是正常情况下只有接其中一台核心交换机的接口是 UP 的，另外一台核心交换机的接口是 DOWN 的，处于备份状态。

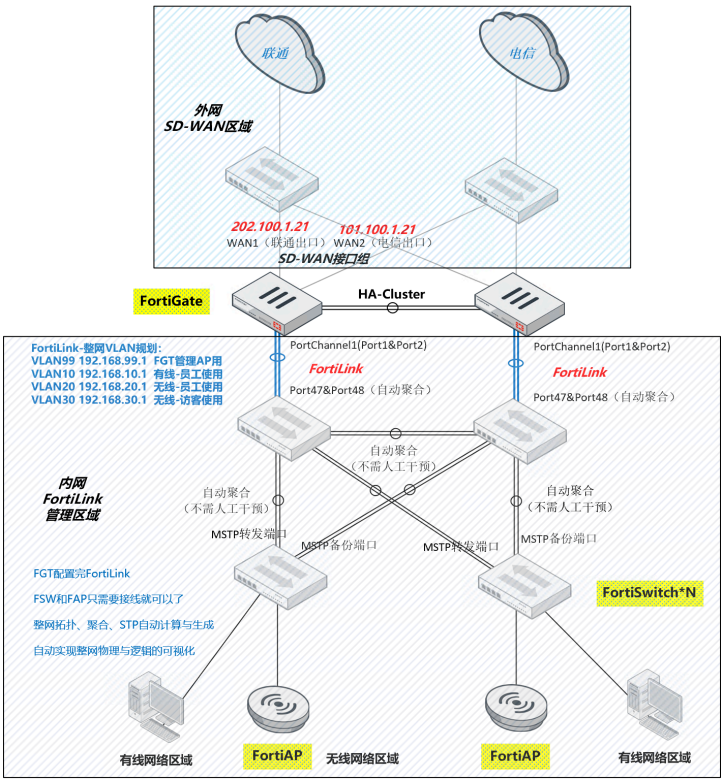


4. FortiGate100 及以下型号的设备改进方案。使用硬交换机接口（LAN/Internal）对接三台 FortiSwitch，使用硬交换部署 FortiLink 的时候尽量不要有 STP（环路）。



最终方案三

1. 以上两种改善方案也存在自身的问题，实际的情况是，大部分的交换机都不在同一个机架上，可能每个楼层一台交换机，这样的话，如果需要防火墙和每一个交换机进行直接互联，这个部分有时候由于环境问题会存在一定的困难。
2. 另外还有一个问题是防火墙本身也存在单点故障风险。因此一个更加好一点的改进方案是，引进 FortiGate 的 HA 机制，然后启用双核心交换机组网完成 FortiLink 的架构。
3. 继续改善的 FortiGate HA 加双核心 FortiSwitch 的星型组网方案。



4. 由于 FortiGate 使用聚合接口 + HA 部署 FortiLink，这个时候可以不需要使用“FortiLink split interface”进行跨交换机的对接，因为将聚合都接到一台交换机可以增加带宽的使用效率，同时 HA 可以 monitor 聚合接口，一旦核心交换机或聚合接口故障，可以 HA 切换到备防火墙上，相应的流量也会切换到另外一台核心交换机上，从而实现整体的业务备份效果。
5. 对于上述所有的拓扑图，从配置的角度来说只需要选择好适合的方案，然后只需配置 FortiGate 的 FortiLink 接口即可，FortiSwitch 和 FortiAP 只需要接物理的线即可，不需要有任何的手工配置 FortiSwitch 和 FortiAP，按照拓扑图接好物理的线之后，等待 FortiLink 整网收敛完毕之后，就可以通过 FortiGate 统一的管理和配置 FortiSwitch 和 FortiAP，整网的所有的有线和无线的配置统统都在 FortiGate 上完成，通过 FortiGate 的 FortiLink 做到整网可控、可配、可视化。

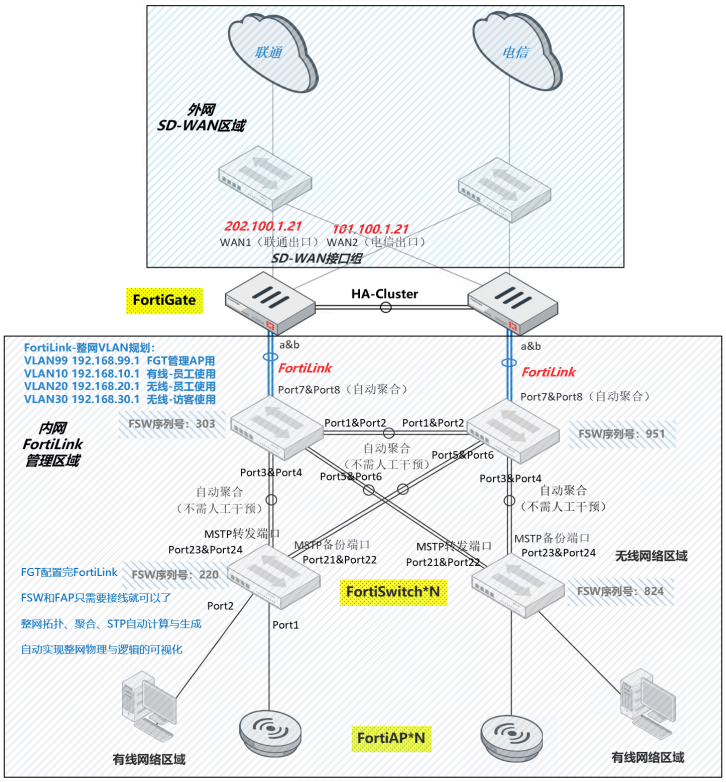
6. 总结来看，越完美的东西就意味着越复杂，有时候简单也是一种优势，因此选择用什么样的 FortiLink 组网，需要考虑自身的需求和对业务的重视情况而定，并非越简单越好，也并非越复杂越好，适合自己的才是最好的。

FortiLink 星型组网配置举例

网络需求

FortiGate 运行 HA-Cluster，核心交换和接入交换机采用全聚合接口（FortiSwitch 全自动聚合，接线即可，无需手工配置聚合接口）对接的星型组网结构。

网络拓扑



1. 两台 FortiGate 配置为 HA 主备模式。
2. 整体规划：
   - 外网 IP 网段规划：
     - WAN1 联通出口：IP 202.100.1.21，GW 202.100.1.192
     - WAN2 电信出口：IP 101.100.1.21，GW 101.100.1.192
   - WAN1 和 WAN2 合并成 SD-WAN 接口。
   - 内网 IP 网段规划，FGT 通过 FortiLink + Capwap 协议统一的接管 FortiSW 和 FortiAP，为有线和无线统一进行管理和业务处理：
     - 有线网络规划：VLAN 10，192.168.10.1
     - 无线网络规划：员工 WIFI VLAN 20，192.168.20.1
     - 无线网络规划：访客 WIFI VLAN 30，192.168.30.1
     - 管理 FortiAP：VLAN 99，192.168.99.1
   - FortiLink 互联部分规划：
     - FortiGate 的 a 和 b 做聚合接口与 Fortiswitch 的后两个接口互联即可实现聚合接口的 FortiLink 管理。
     - 加入 FortiLink 的接口自动变为聚合口成员。

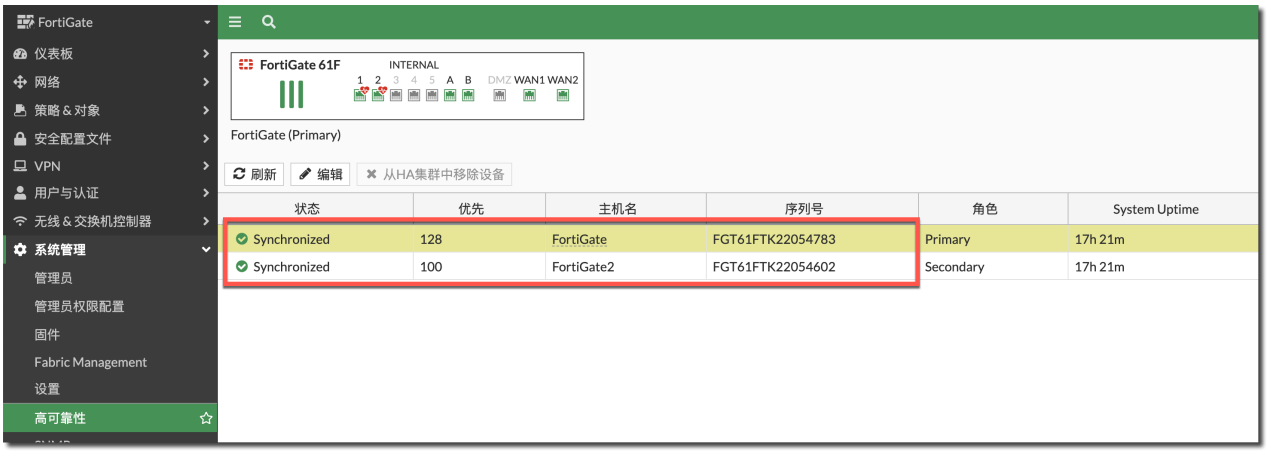
配置步骤

准备事项

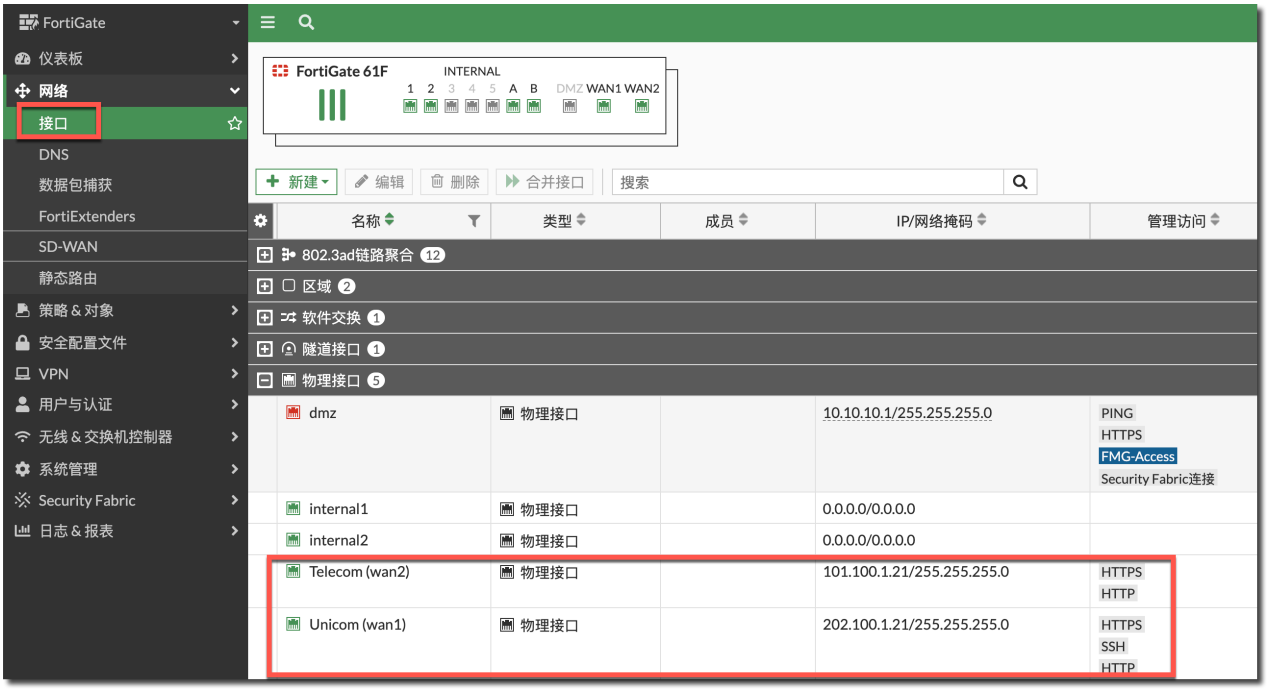
1. 首先确认 FortiGate 如下全局配置已开启（一般默认为开启状态）。

```
config system settings
    set gui-switch-controller enable
end
```

2. 配置两台 FortiGate 的 HA 主备模式（配置方法请参考 HA 双机热备 → HA 典型基础配置 ↗）。

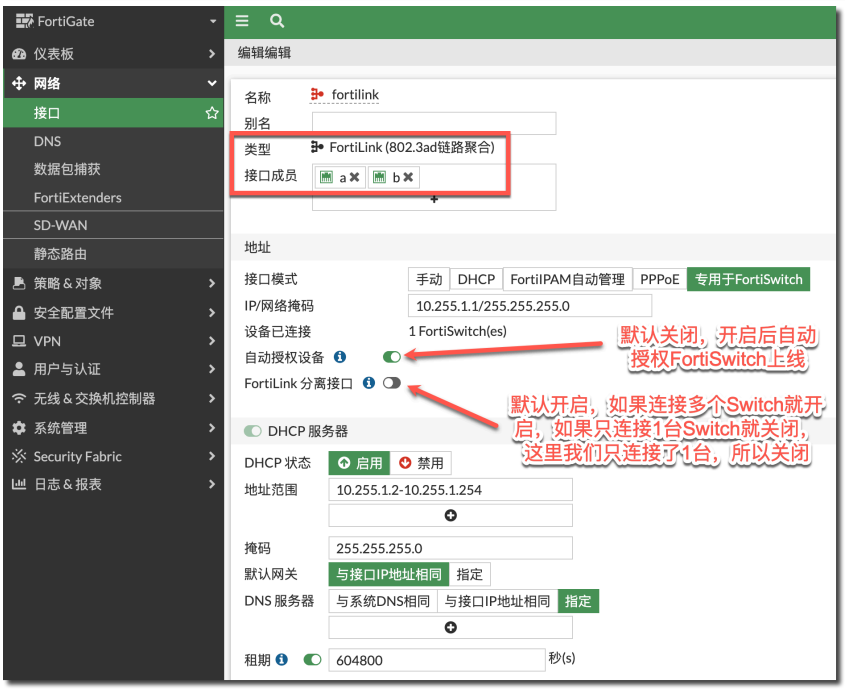


3. 配置 wan1 和 wan2 的 IP 地址。



4. 配置 FortiLink 实现 FortiSwith 上线，我们只需要将用于 FortiLink 的成员接口加入 FortiLink 接口即可。此拓扑中，FortiGate 用于连接 FortiSwitch 的接口为 a 和 b，编辑 FortiLink 接口，将 a 和 b 加入 FortiLink（某些产品型号会默认将两个接口加入 FortiLink），开启自动授权设备，关闭 FortiLink 分离接口，其他配置保持默认，可以看到 FortiLink 为一个聚合接口。


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 8
fortilink
802.3ad链路聚合
专用于 FortiSwitch
PING
Security Fabric连接
隧道接口 1
物理接口 5
a
物理接口
0.0.0.0/0.0.0.0
b
物理接口
0.0.0.0/0.0.0.0
dmz
物理接口
10.10.10.1/255.255.255.0
PING
HTTPS
FMG-Access
Security Fabric连接
Telecom (wan2)
物理接口
101.100.1.21/255.255.255.0
HTTPS
HTTP
Unicom (wan1)
物理接口
202.100.1.21/255.255.255.0
HTTPS
HTTP

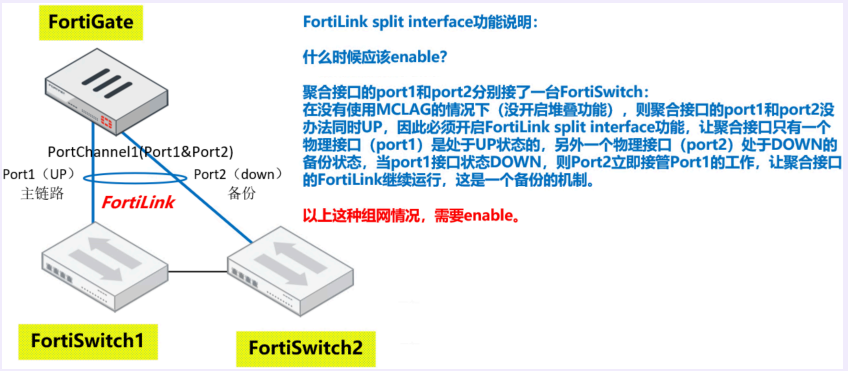

FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
编辑编辑
名称
fortilink
别名
类型
FortiLink (802.3ad链路聚合)
接口成员
a
b
地址
接口模式
手动
DHCP
FortiIPAM自动管理
PPPoE
专用于FortiSwitch
IP/网络掩码
10.255.1.1/255.255.255.0
设备已连接
1 FortiSwitch(es)
自动授权设备
默认关闭，开启后自动授权FortiSwitch上线
FortiLink 分离接口
默认开启，如果连接多个Switch就开启，如果只连接1台Switch就关闭，这里我们只连接了1台，所以关闭
DHCP 服务器
DHCP 状态
启用
禁用
地址范围
10.255.1.2-10.255.1.254
掩码
255.255.255.0
默认网关
与接口IP地址相同
指定
DNS 服务器
与系统DNS相同
与接口IP地址相同
指定
租期
604800
秒(s)


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
FortiGate 61F
INTERNAL
1 2 3 4 5 A B
DMZ WAN1 WAN2
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 12
fortilink
802.3ad链路聚合
a
b
专用于 FortiSwitch
PING
Security Fabric连接

重要

FortiLink 分离接口什么时候应该开启?

分离接口的意是，如果聚合接口的 port1 和 port2 分别连接了一台 SW，在没有使用 MCLAG 的情况下（没开启堆叠功能），则聚合接口的 port1 和 port2 没办法同时 UP（同时开启聚合接口无法正常工作），因此必须开启 FortiLink 分离接口功能，让聚合接口只有一个物理接口（port1）是处于 UP 状态的，另外一个物理接口（port2）处于 DOWN 的备份状态，当 port1 接口状态 DOWN，则 Port2 立即接管 Port1 的工作，让聚合接口的 FortiLink 继续运行，这是一个聚合接口在 FortiLink 管理时候的备份机制。



重要

FortiLink split interface 什么时候应该 disable?

如果 FGT 的聚合接口只接到了一台交换机上，并没有跨交换机聚合（没有 MCLAG，也不需要 FortiLink split interface），因此这种情况应该让聚合接口的两个物理接口 port1 和 port2 同时 UP，两个接口同时工作起来，以便提供更高带宽和备份的作用。



相关 CLI:

```
NTP:
config system ntp
    set ntpsync enable
    set server-mode enable
    set interface "fortilink"
end
-------------------------------------------------------------------

DHCP:
config system dhcp server
    edit 2
        set dns-service local
        set ntp-service local
        set default-gateway 10.255.1.1
        set netmask 255.255.255.0
        set interface "fortilink"
        config ip-range
            edit 1
                set start-ip 10.255.1.2
                set end-ip 10.255.1.254
            next
        end
        set vci-match enable
        set vci-string "FortiSwitch" "FortiExtender"
    next
```

```
end
————————————————————————————————————————————
FortiLink:
config system interface
    edit "fortilink"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set type aggregate
        set member "a" "b"
        set lldp-reception enable
        set lldp-transmission enable
        set auto-auth-extension-device enable
        set fortilink-split-interface disable
        set switch-controller-nac "fortilink"
        set switch-controller-dynamic "fortilink"
        set swc-first-create 255
    next
end
```

连接 FortiLink

1. 配置完 FortiGate 之后，这个时候再去接 FortiGate 的 FortiLink 到 FortiSwitch 的线，以及 FortiSwitch 之间的线，严格按照拓扑图，从核心交换机往接入交换机这样的顺序接线。FortiSwitch 如果以前有使用过，建议都恢复一下出厂设置，然后再加入到 FortiLink。
2. FortiSwitch 的靠后接口一般都默认开启了 FortiLink 接口属性，如果接前面的接口可能会无法加入到 FortiLink，因此 FortiSwitch 的 FortiLink 最好都使用靠后的接口进行接线。FortiSwitch 各型号默认开启 FortiLink 功能的接口如下：

FortiSwitch 型号	默认开启 FortiLink 的接口
FS-108D-POE	port9–port10
FS-108E, FS-108E-POE, FS-108E-FPOE	port7–port10
FSR-112D-POE	port5–port12
FS-124D, FS-124D-POE	port23–port26
FSR-124D	port1-port4, port21–port28
FS-124E, FS-124E-POE, FS-124E-FPOE	port21–port28
FS-148E, FS-148E-POE	port21–port52
FS-224D-POE	port21–port24
FS-224D-FPOE	port21–port28
FS-224E, FS-224E-POE	port21–port28
FS-248D, FS-248D-FPOE	port45–port52
FS-248D-POE	port47–port50
FS-248E-POE, FS-248E-FPOE	port45–port52
FS-424D, FS-424D-POE, FS-424D-FPOE	port23–port26
FS-424E-Fiber	port1-port30
FS-426E-FPOE-MG	port23-port30
FS-448D, FS-448D-POE, FS-448D-FPOE	port45–port52
FS-524D, FS-524D-FPOE	port21–port30
FS-548D	port39–port54
FS-548D-FPOE, FS-548DN	port45–port54
FS-1024D	port1–port24
FS-1048D, FS-1048E	port1–port52
FS-3032D, FS-3032E	port1–port32

3. 如果 FortiSwitch 无法被 FortiGate 管理，登陆到 FortiSwitch 上进行确认接口是否开启了 FortiLink。

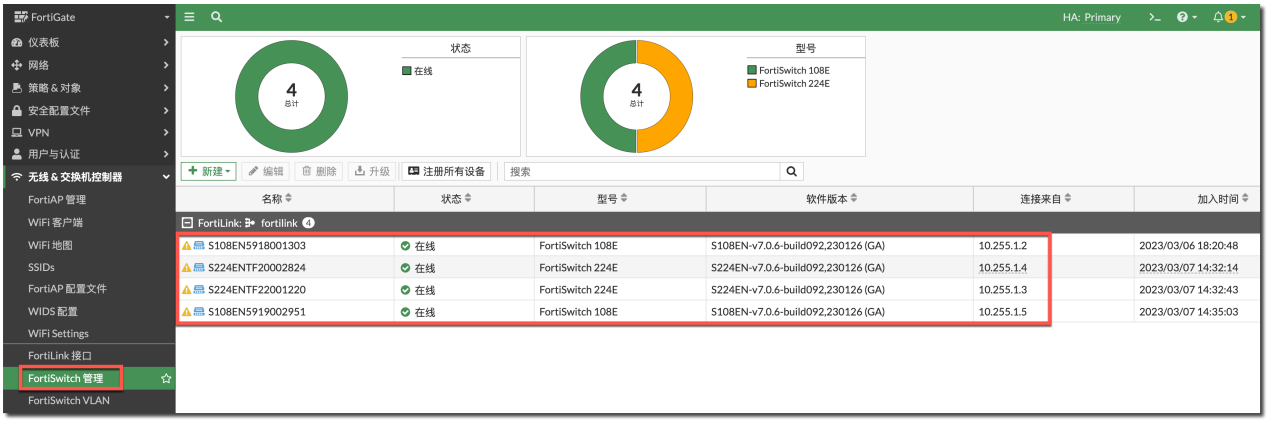
```
FortiSwitch # show switch interface port9
config switch interface
    edit "port9
        set auto-discovery-fortilink enable    //接口下又此命令的接口才可以自动加入到FortiLink中
    next
end
```

4. 交换机之间如果接多根线，会自动进行聚合，但是 FortiLink 自动聚合会花比较长的时间，建议可以先接单线，等 FortiLink 的拓扑较快的形成结束之后，再增加彼此之间的另外一根线，再进行自动聚合，这样效率会比较高一些。

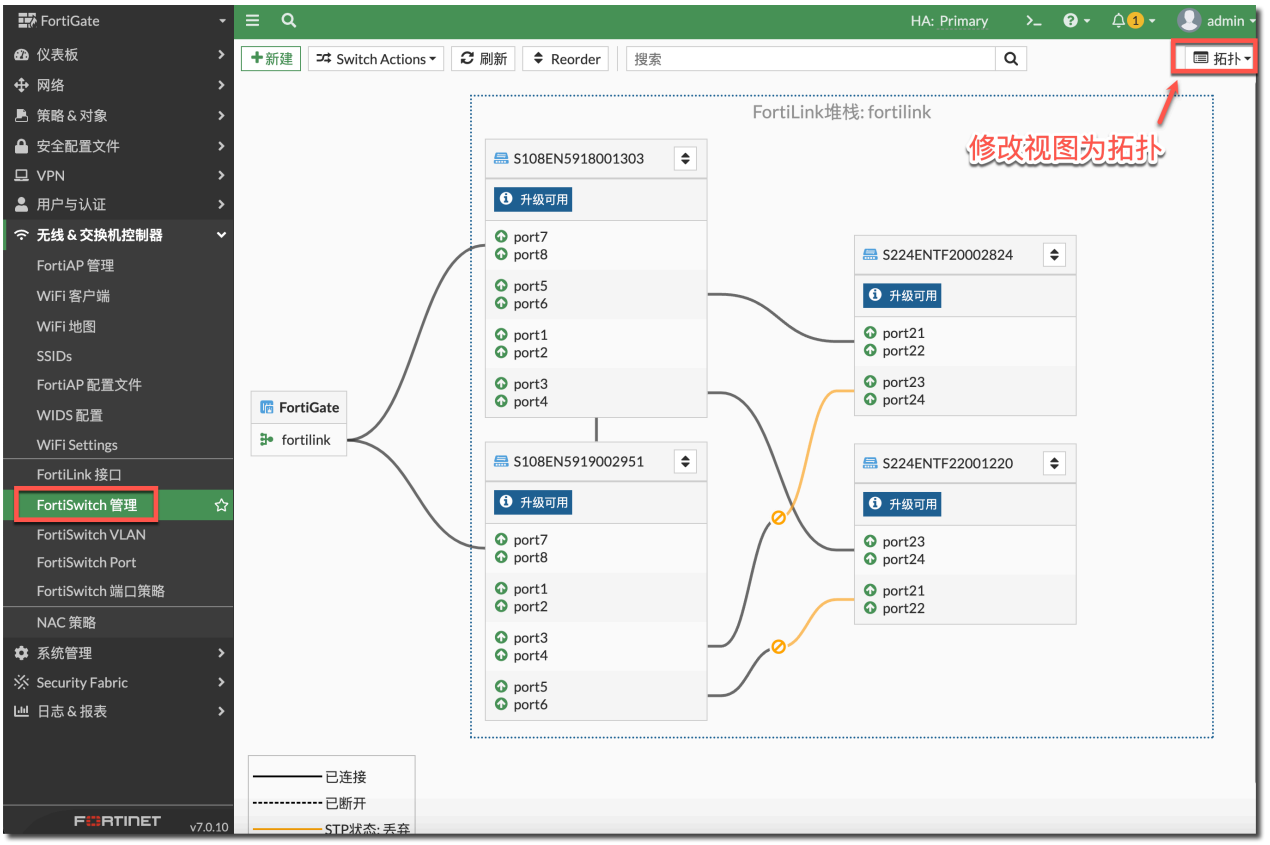
5. 接线是很重要的步骤，仔细看拓扑图，做好接口标签，不要接错。

6. 接好线之后，FortiSwitch 和 FortiGate 之间会自动在后台进行 FortiLink、CAPWAP、MSTP、聚合接口的协商加入等操作，等 FortiLink 的管理结束之后，在 FortiGate 上会看到一个网站的 FortiSwitch 拓扑图，和我们规划的拓扑图应该是一致的，可以仔细核对接线是否正确。

7. 在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，查看交换机被管理状态，可以看到 4 台交换机均已上线。

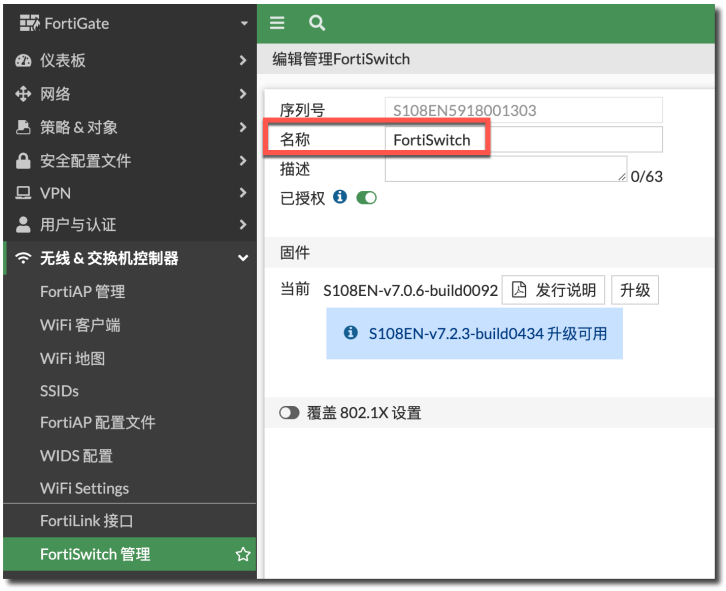


8. 修改视图为拓扑，可以更直观的看到当前防火墙与交换机的接口连接。



9. 交换机加入到 FortiGate 的 FortiLink，会先清除配置再重启，然后从 standalone 的状态进入到“Fortilink remote control”的状态，后续所有的配置都通过 FortiGate 下发即可。

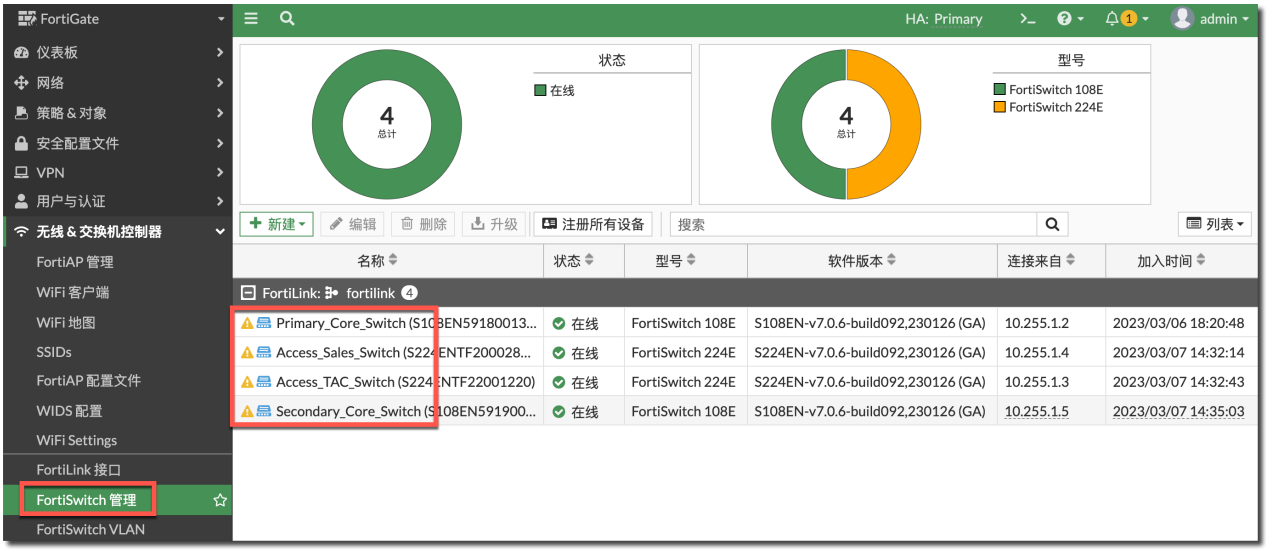
10. 编辑交换机，可以修改交换机名字以及升级版本，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。



1. FortiSwitch 之间接多根线进行 FortiLink 互联，会默认自动聚合。其实接一根网线也是聚合的，只是聚合组里面只有一个接口而已，因此多根网线直连，这些直连的线都会自动加入到一个聚合接口里面（注意不能跨交换机，跨交换机的聚合是 MCLAG）。

```
FortiGate # exec ssh admin@10.255.1.2
Warning: Permanently added '10.255.1.2' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

S108EN5918001303 # show switch trunk
path=switch, objname=trunk, tablename=(null), size=752
config switch trunk
    edit "GT61FTK22054783"
        set mode lacp-active
        set auto-isl 1
        set fortilink 1
        set members "port7" "port8"
    next
    edit "4ENTF20002824-0"
        set mode lacp-active
        set auto-isl 1
        set members "port5" "port6"
    next
    edit "4ENTF22001220-0"
        set mode lacp-active
        set auto-isl 1
        set members "port3" "port4"
    next
    edit "8EN5919002951-0"
        set mode lacp-active
        set auto-isl 1
        set members "port1" "port2"
    next
end
```

2. 另外建议给每台交换机命名一个名字，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。



STP 状态查看

1. FortiLink 之间的交换机互联，如果不用 MCLAG 的话，本质上是 MSTP（多实例生成树）在整网统一的进行接口环路计算，从而实现网络的备份，与快速收敛恢复。 可以通过 FGT 的命令行界面用 Telent/SSH 到 FortiSwitch 上并分别查看四台 FortiSwitch 的 STP 状态。

   - 主核心交换机（303）上查看：

```
FortiGate # exec ssh admin@10.255.1.2
Warning: Permanently added '10.255.1.2' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

Primary_Core_Switch #  diagnose stp instance list
MST Instance Information, primary-Channel:
...
Instance ID 15
  Config         Priority 24576, VLANs 4094
                 Bridge MAC e81cba1c0900
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 0
                 (This bridge is the regional root)
  //直连主FGT的FSW会自动将网桥优先级降低到：Priority 24576，因此整个FortiLink二层网络中，自己就是根网桥
  TCN Events     Triggered 10 (0d 0h 24m 59s ago), Received 67 (0d 0h 24m 26s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   ______  _________  _________  ___________  ___________  ________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  GT61FTK22054783    2G      10000      128        DESIGNATED   FORWARDING   ED
  //与主FortiGate_FortiLink互联的自动生成的聚合接口，STP为指定接口，转发状态
  4ENTF20002824-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号824结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态
  4ENTF22001220-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号220结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态
  8EN5919002951-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与备核心交换机303互联的自动生成的聚合接口，STP为指定接口，转发状态
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

   - 备核心交换机（951）上查看：

```
FortiGate # exec ssh admin@10.255.1.5
Warning: Permanently added '10.255.1.5' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

Secondary_Core_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
...
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC 04d59010ff48
                 //自己的MAC
```

```
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口 8EN5918001303-0
  TCN Events     Triggered 4 (0d 0h 41m 4s ago), Received 2 (0d 0h 41m 32s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  _________________  ______  _________  _________  ___________  ___________  _________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  4ENTF22001220-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号220结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态
  8EN5918001303-0    2G      1          128        ROOT         FORWARDING   EN
  //与主核心交换机303互联的自动生成的聚合接口，STP为root接口，转发状态
  4ENTF20002824-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号824结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态
  GT61FTK22054602    2G      10000      128        DESIGNATED   FORWARDING   ED
  //与备FortiGate_FortiLink互联的自动生成的聚合接口，STP为指定接口，转发状态（但由于连的是备防火墙，实际物理口是discarding状态）
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

- 接入交换机（220）上查看：

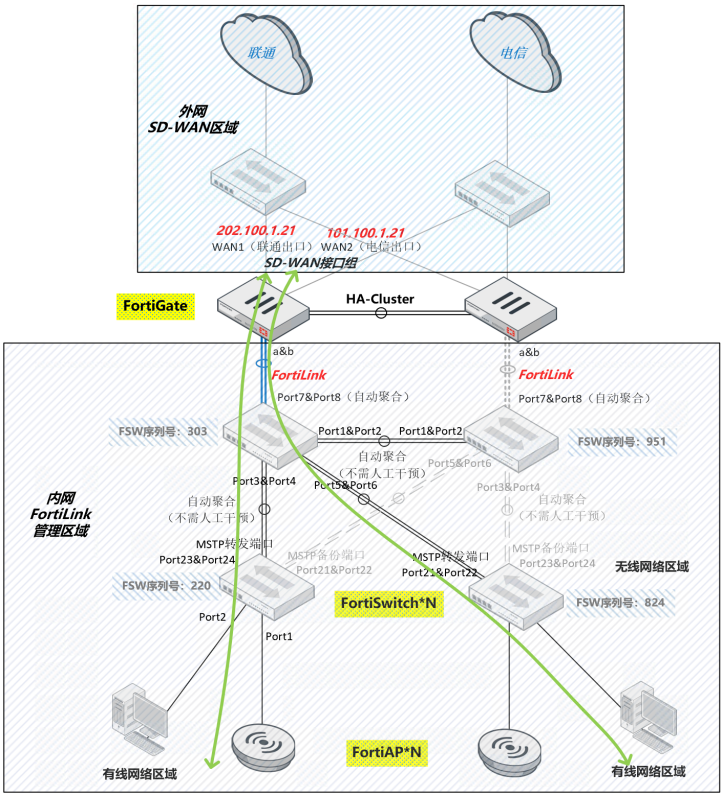
```
FortiGate # exec ssh admin@10.255.1.3
admin@10.255.1.3's password:

Access_TAC_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC ac712e465448
                 //自己的MAC地址
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口8EN5918001303-0
  TCN Events     Triggered 1 (0d 1h 12m 46s ago), Received 6 (0d 1h 9m 25s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  _________________  ______  _________  _________  ___________  ___________  _________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  8EN5918001303-0    2G      1          128        ROOT         FORWARDING   EN
  //接主核心交换机303，STP为根接口，转发状态
  8EN5919002951-0    2G      1          128        ALTERNATIVE  DISCARDING   EN
  //接备核心交换机951，STP为备份接口ALTERNATIVE  DISCARDING，处于阻塞状态
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

- 接入交换机（824）上查看：

```
FortiGate # exec ssh admin@10.255.1.4
admin@10.255.1.4's password:

Access_Sales_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC e023ff54e7e0
                 //自己的MAC地址
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口8EN5918001303-0
  TCN Events     Triggered 1 (0d 1h 22m 45s ago), Received 7 (0d 1h 18m 58s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  _________________  ______  _________  _________  ___________  ___________  _________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  8EN5918001303-0    2G      1          128        ROOT         FORWARDING   EN
  //接主核心交换机303，STP为根接口，转发状态
  8EN5919002951-0    2G      1          128        ALTERNATIVE  DISCARDING   EN
  //接备核心交换机951，STP为备份接口ALTERNATIVE  DISCARDING，处于阻塞状态
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

2. FortiLink 组网中 STP 是很重要部分，通常 STP 是一个稳定的状态，一旦出现 FortiLink 不稳定，则需要观察 STP 的状态和选举结果，从中找到相关线索，以便查出为何 STP 不稳定的原因，对于排错来说是很重要的。

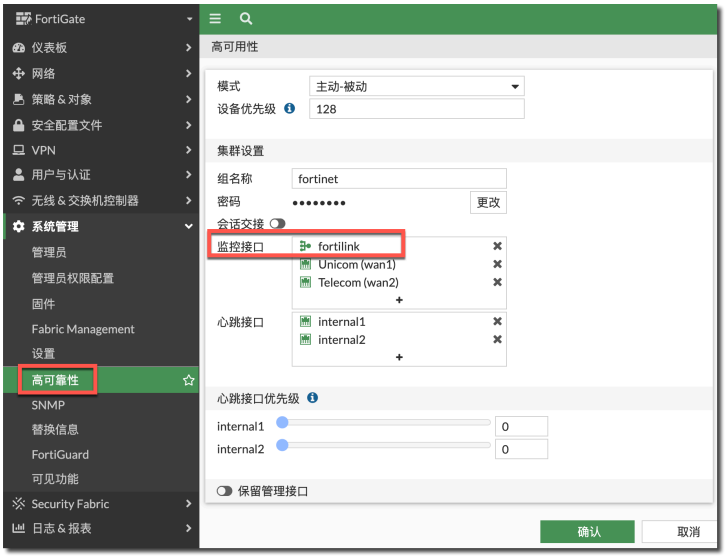
3. FortiLink + STP 稳定状态的实际数据转发拓扑图如下。



4. 在 FortiGate 查看交换机管理状态。

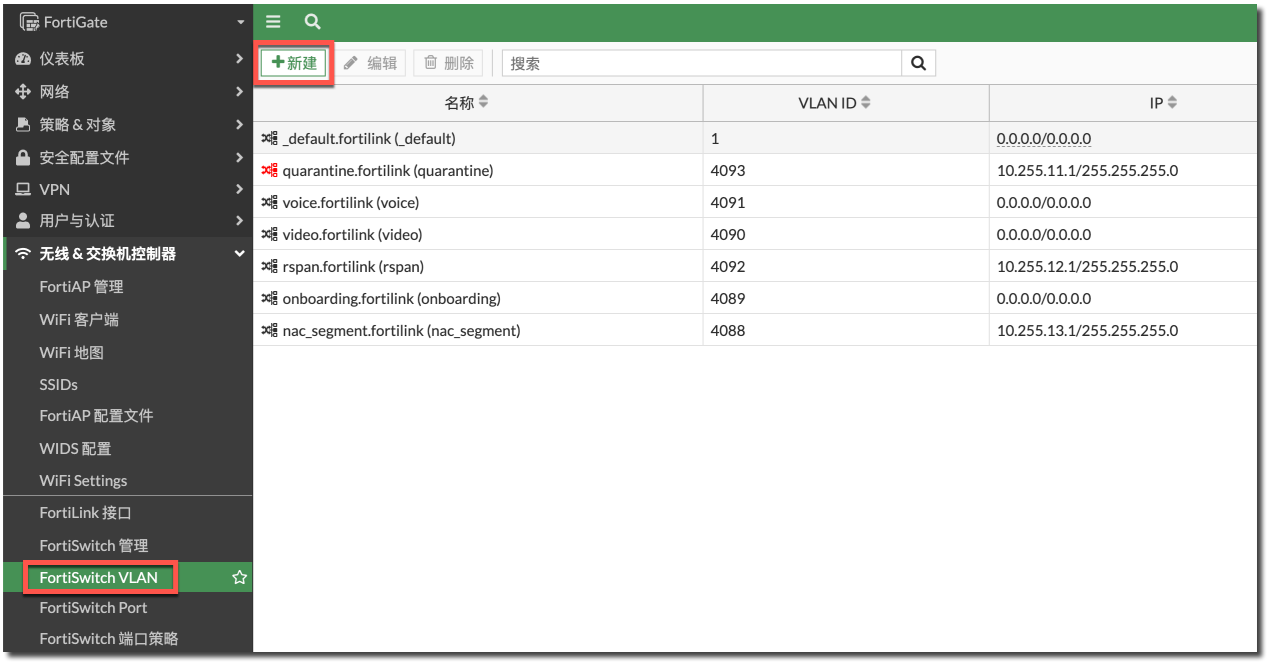
```
FortiGate # execute switch-controller get-conn-status
Managed-devices in current vdom root:
FortiLink interface : fortilink
SWITCH-ID         VERSION           STATUS          FLAG   ADDRESS              JOIN-TIME              NAME
S108EN5918001303  v7.0.6 (092)      Authorized/Up   -    10.255.1.2      Mon Mar  6 18:20:48 2023    Primary_Core_Switch
S224ENTF20002824  v7.0.6 (092)      Authorized/Up   -    10.255.1.4      Tue Mar  7 14:32:14 2023    Access_Sales_Switch
S224ENTF22001220  v7.0.6 (092)      Authorized/Up   -    10.255.1.3      Tue Mar  7 14:32:43 2023    Access_TAC_Switch
S108EN5919002951  v7.0.6 (092)      Authorized/Up   -    10.255.1.5      Tue Mar  7 14:35:03 2023
Secondary_Core_Switch
        Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config sync error, 3=L3
        Managed-Switches: 4 (UP: 4 DOWN: 0)
```

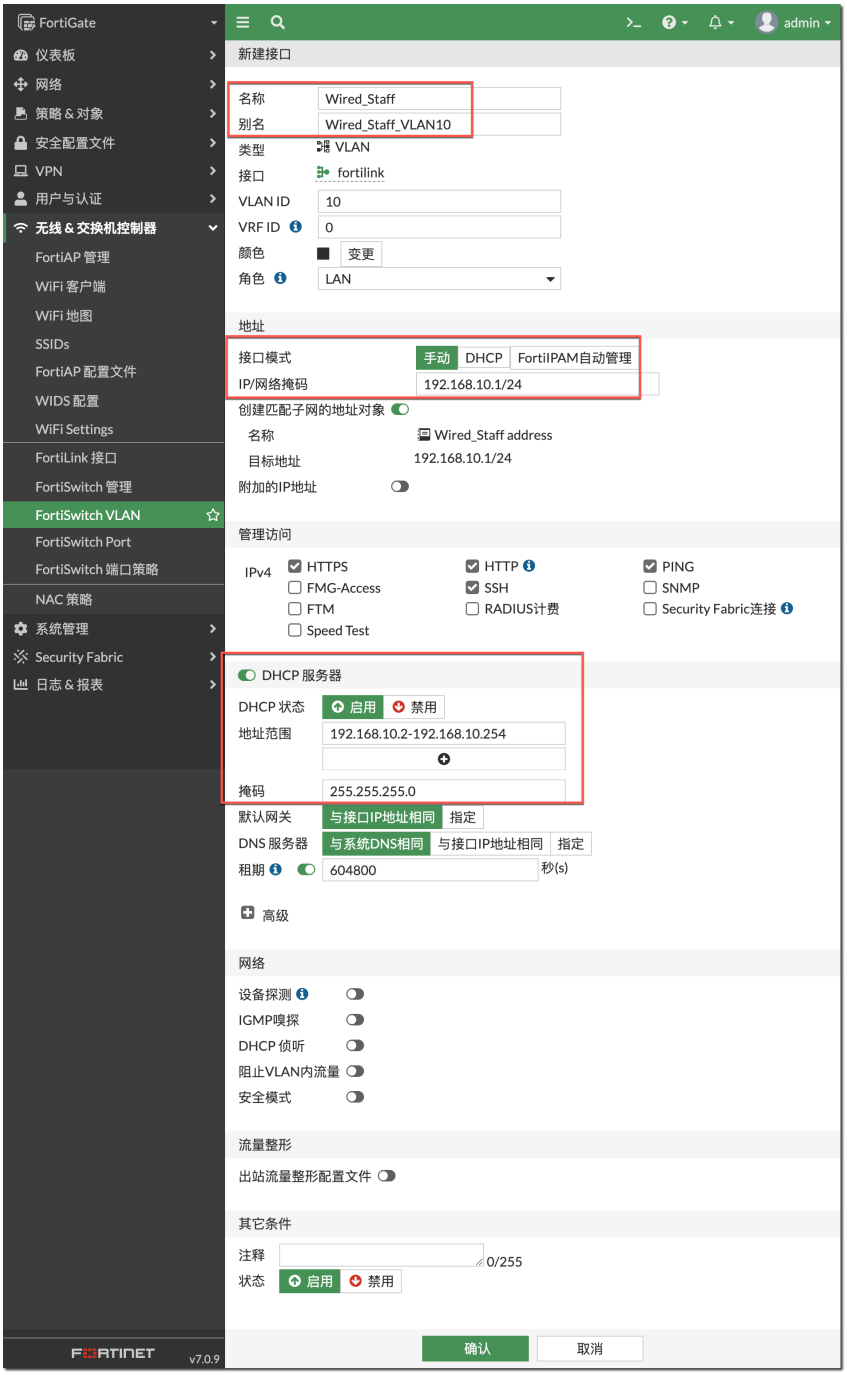
5. 最后不要忘记，在 FortiLink 的聚合接口和 FortiLink 管理配置完毕，记得把这个 FortiLink 聚合接口加入到 HA 的监控接口列表中，这样当主核心交换机挂了（比如断电）之后，业务才可以平稳切换到备防火墙上。

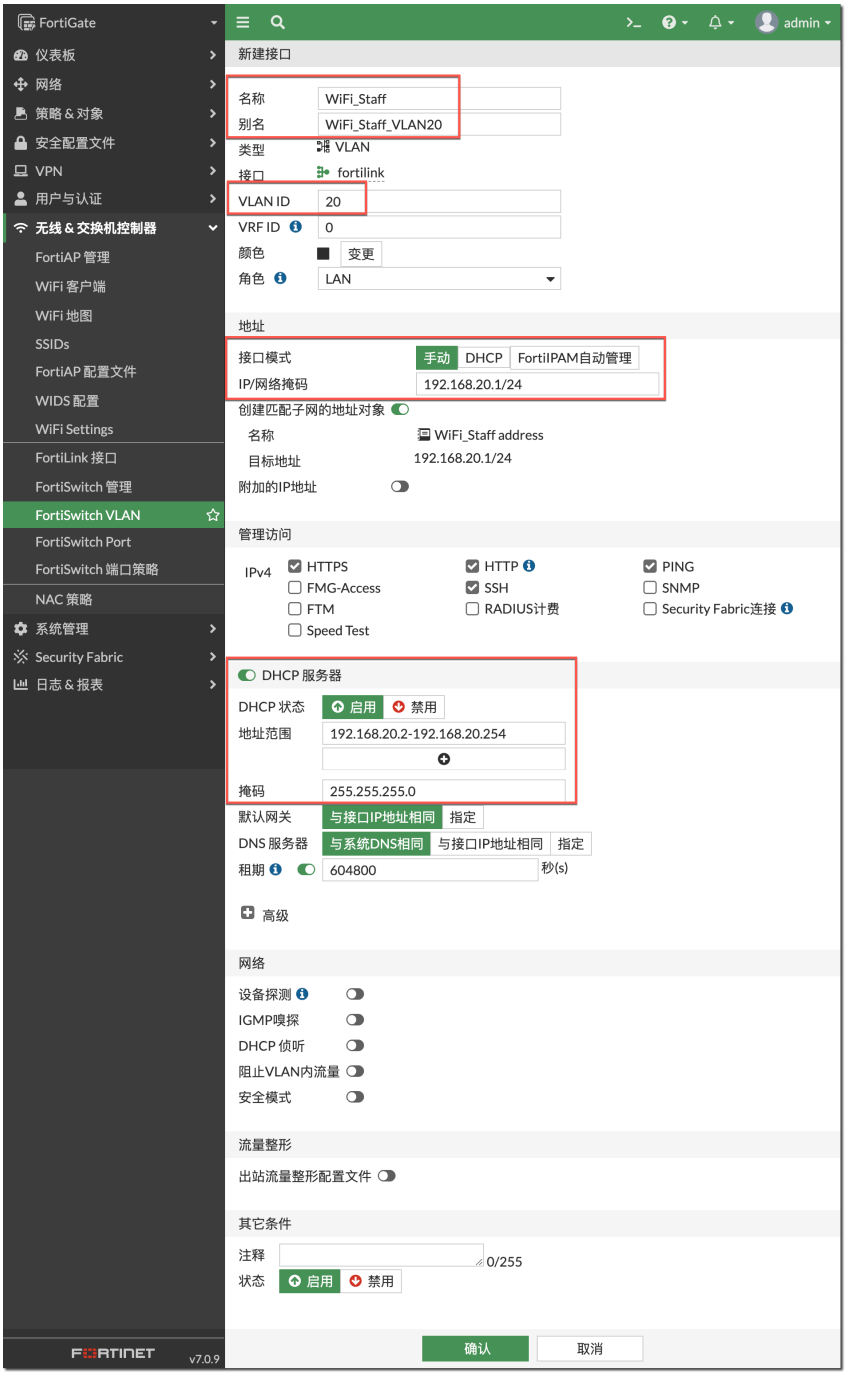


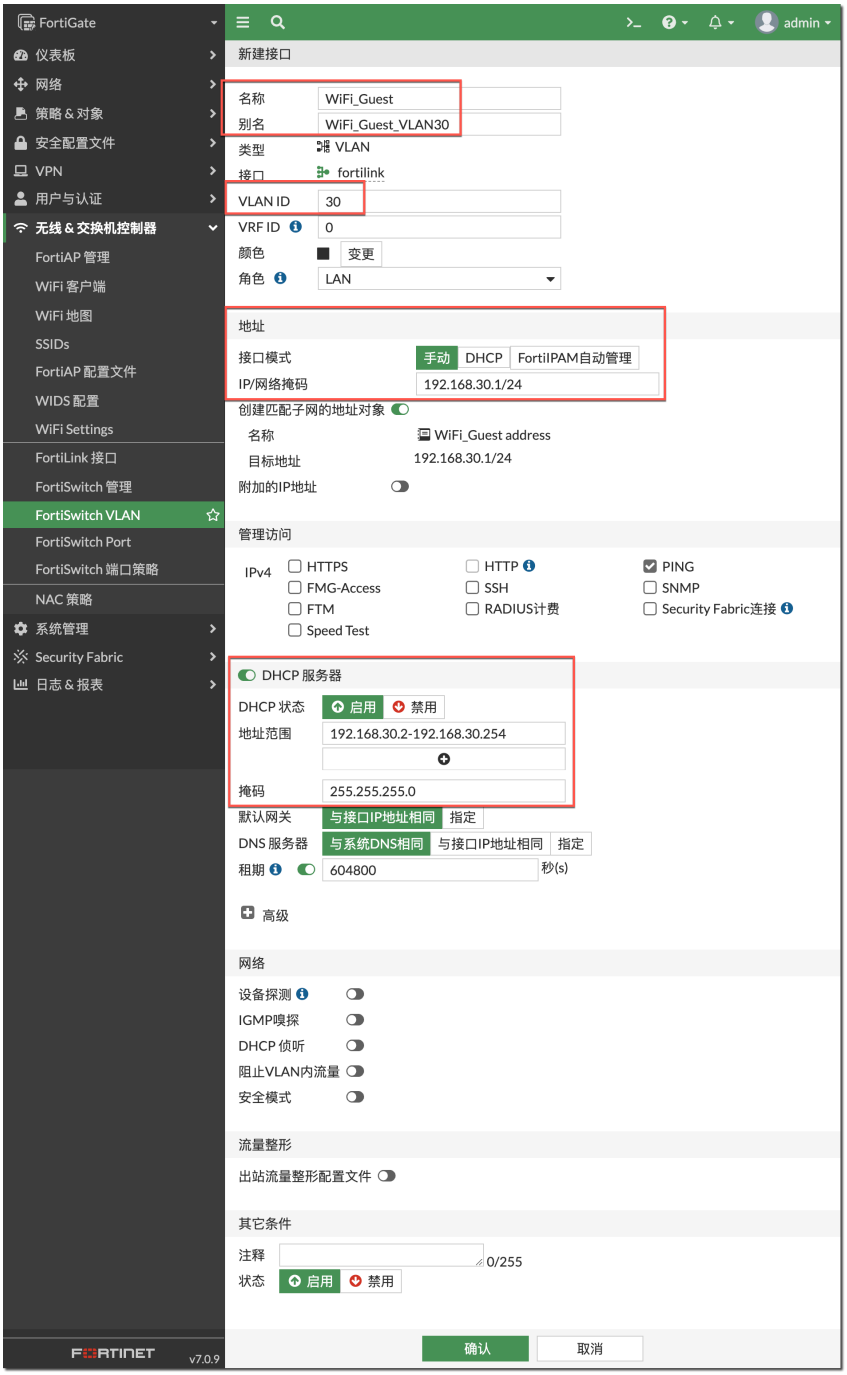
创建 VLAN

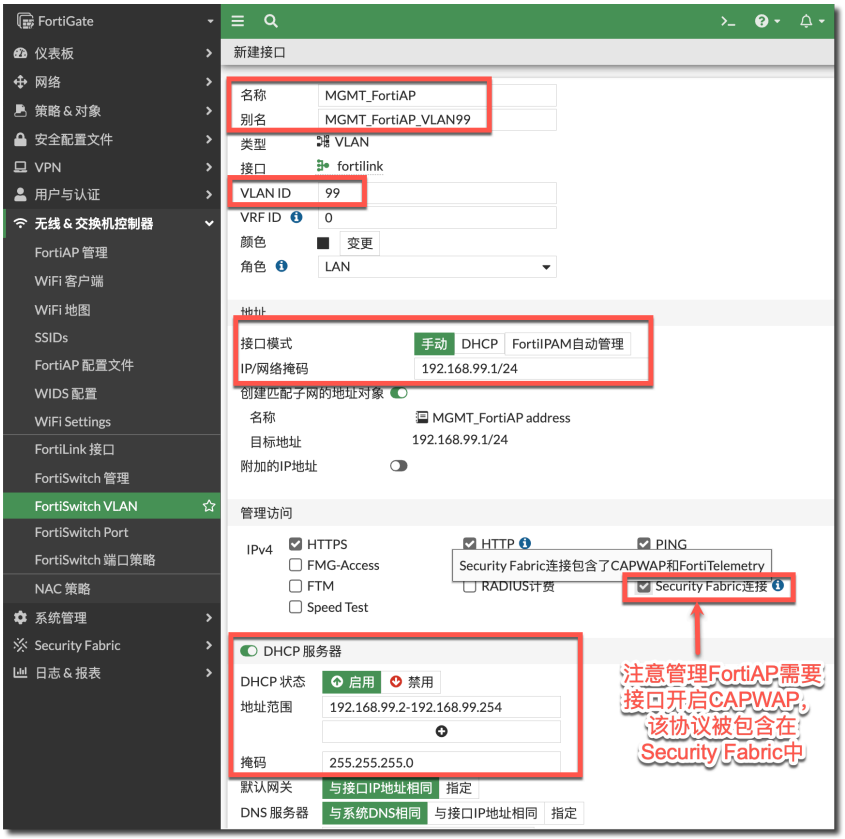
1. FortiGate 通过 FortiLink + Capwap 协议统一的接管 FortiSwitch 和 FortiAP，为有线和无线统一进行管理和业务处理。
2. 进入 FortiGate 的无线 & 交换机控制器 → FortiSwitch VLAN 页面，新建有线网络 VLAN 10、员工无线网络 VLAN 20、访客无线网络 VLAN 30、管理 FortiAP VLAN 99。注意管理 FortiAP 的 VLAN 接口要开启 Security Fabric，包含 CAPWAP。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
+新建
编辑
删除
搜索
名称
VLAN ID
IP
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0

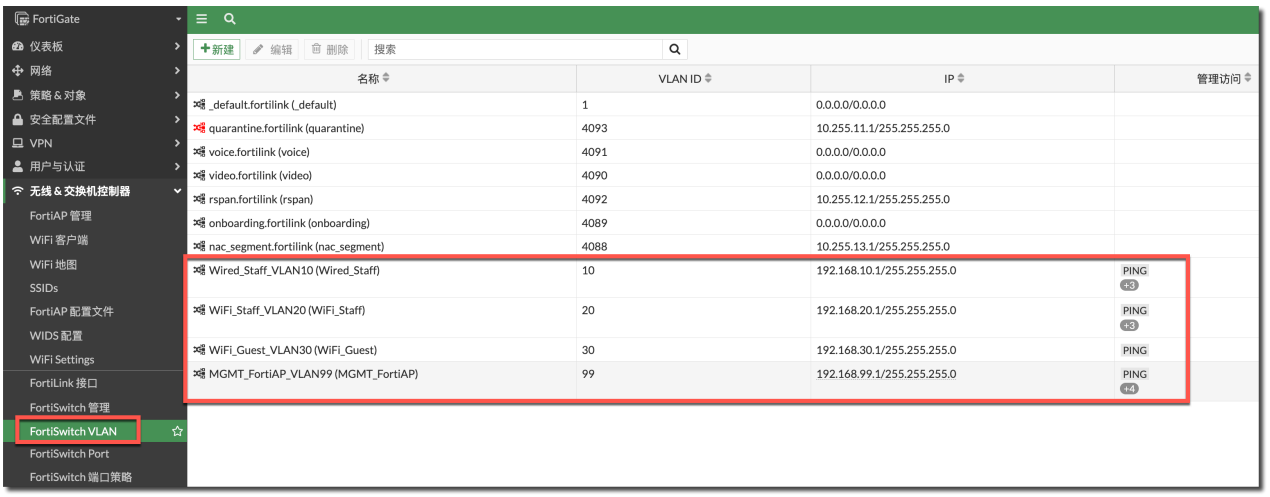

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 Wired_Staff
别名 Wired_Staff_VLAN10
类型 VLAN
接口 fortilink
VLAN ID 10
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.10.1/24
创建匹配子网的地址对象
名称 Wired_Staff address
目标地址 192.168.10.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.10.2-192.168.10.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 WiFi_Staff
别名 WiFi_Staff_VLAN20
类型 VLAN
接口 fortilink
VLAN ID 20
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.20.1/24
创建匹配子网的地址对象
名称 WiFi_Staff address
目标地址 192.168.20.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.20.2-192.168.20.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消

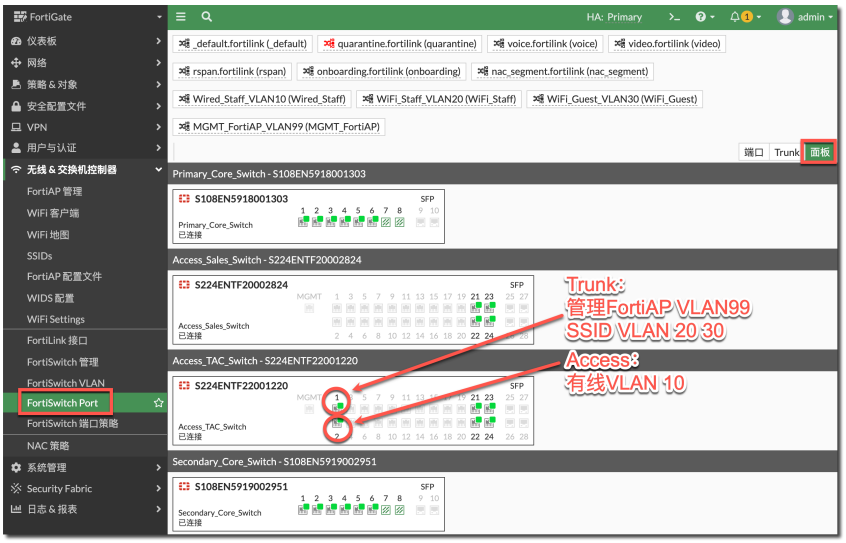

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
新建接口
名称 WiFi_Guest
别名 WiFi_Guest_VLAN30
类型 VLAN
接口 fortilink
VLAN ID 30
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.30.1/24
创建匹配子网的地址对象
名称 WiFi_Guest address
目标地址 192.168.30.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.30.2-192.168.30.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消
FORTINET
v7.0.9


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
新建接口
名称 MGMT_FortiAP
别名 MGMT_FortiAP_VLAN99
类型 VLAN
接口 fortilink
VLAN ID 99
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.99.1/24
创建匹配子网的地址对象
名称 MGMT_FortiAP address
目标地址 192.168.99.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access
FTM
Speed Test
Security Fabric连接包含了CAPWAP和FortiTelemetry
RADIUS计费
Security Fabric连接
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.99.2-192.168.99.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
注意管理FortiAP需要接口开启CAPWAP，该协议被包含在Security Fabric中

3. VLAN 创建完成。

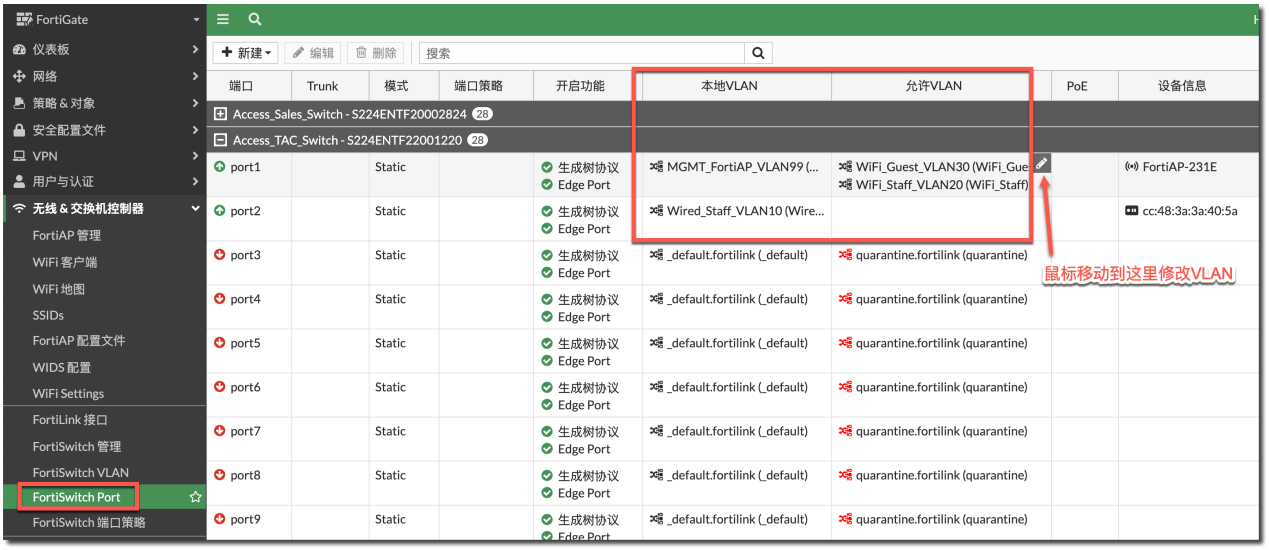

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
新建 编辑 删除 搜索
名称 VLAN ID IP 管理访问
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0
Wired_Staff_VLAN10 (Wired_Staff) 10 192.168.10.1/255.255.255.0 PING
WiFi_Staff_VLAN20 (WiFi_Staff) 20 192.168.20.1/255.255.255.0 PING
WiFi_Guest_VLAN30 (WiFi_Guest) 30 192.168.30.1/255.255.255.0 PING
MGMT_FortiAP_VLAN99 (MGMT_FortiAP) 99 192.168.99.1/255.255.255.0 PING

分配 VLAN



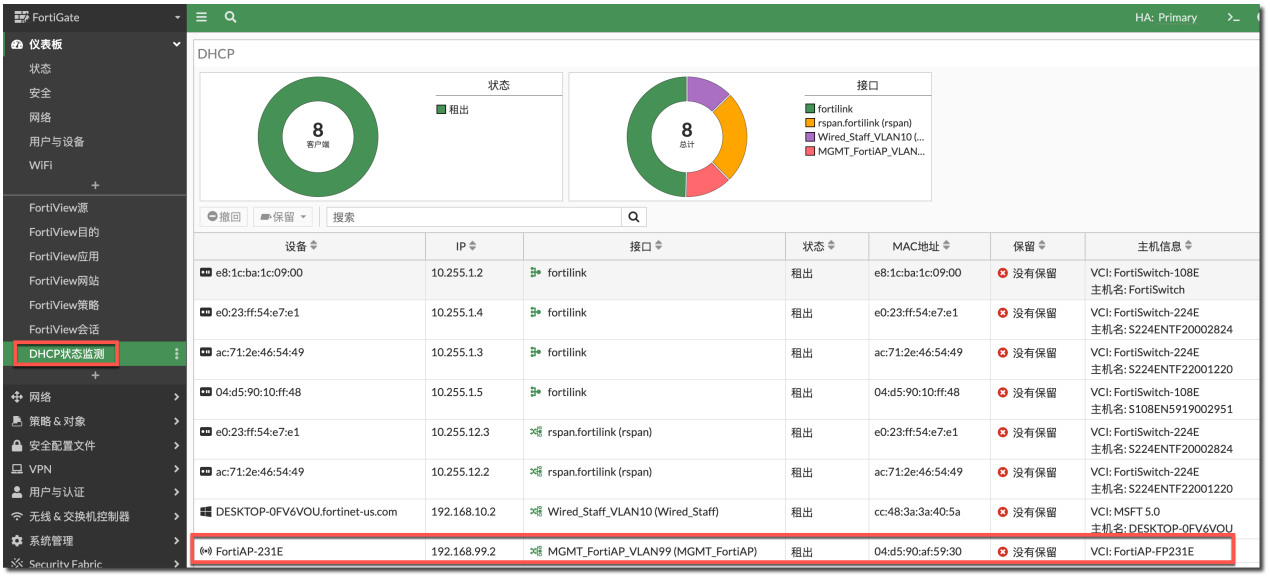
1. 接入交换机的 port1 接 FortiAP，配置为 Trunk 模式，允许 VLAN20 和 VLAN30，VLAN20 用于 WIFI - STAFF 的 SSID 用户，VLAN30 用于 WIFI - GUEST 的 SSID 用户。另外本地 VLAN 设置为 VLAN 99，用于 FortiAP 的管理（FortiAP 的管理相当于使用不打 vlan-tag 的 VLAN 99，本地 VLAN 默认是 VLAN 1，因此只需要将本地 VLAN 设置为 VLAN 99 即可）。

2. Port2 接具体的物理 PC，因此只需要将 Port2 设置为 access 口，分配 VLAN 10 即可，用 FortiSwitch 的概念来说就是将本地 VLAN 设置为 VLAN 10，本地 VLAN 不打 vlan-tag，相当于就是设置接口为 access vlan 10。

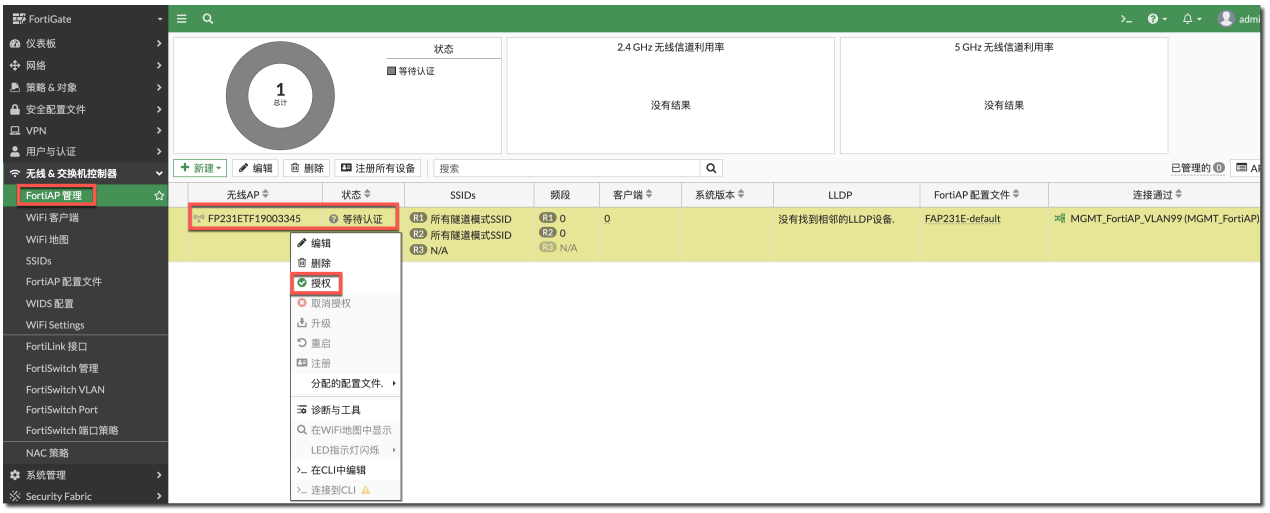


无线配置

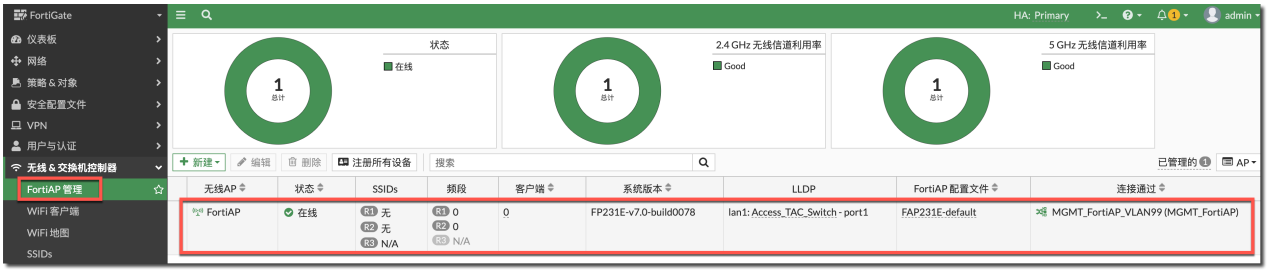
1. FortiAP 连接后，会通过 VLAN 99 的 DHCP 获取到 192.168.99.0/24 网段的 IP 地址。



2. 在之前的配置中，VLAN 99 的接口开启了 CAPWAP 协议（包含在 Security Fabric 中），激活了 FortiGate 的无线管理功能，这时候 FortiAP 会自动找到 FGT 进行注册。因此在 FortiAP 管理的地方可以看到这个 FortiAP 的信息（如果没有看到检查配置，并将 FortiAP 恢复出厂，让 FortiAP 重新获取 IP 地址并发起注册）。



3. 右键点击上图中等待认证的 FortiAP，点击授权按钮，等待几分钟，FortiAP 成功上线。



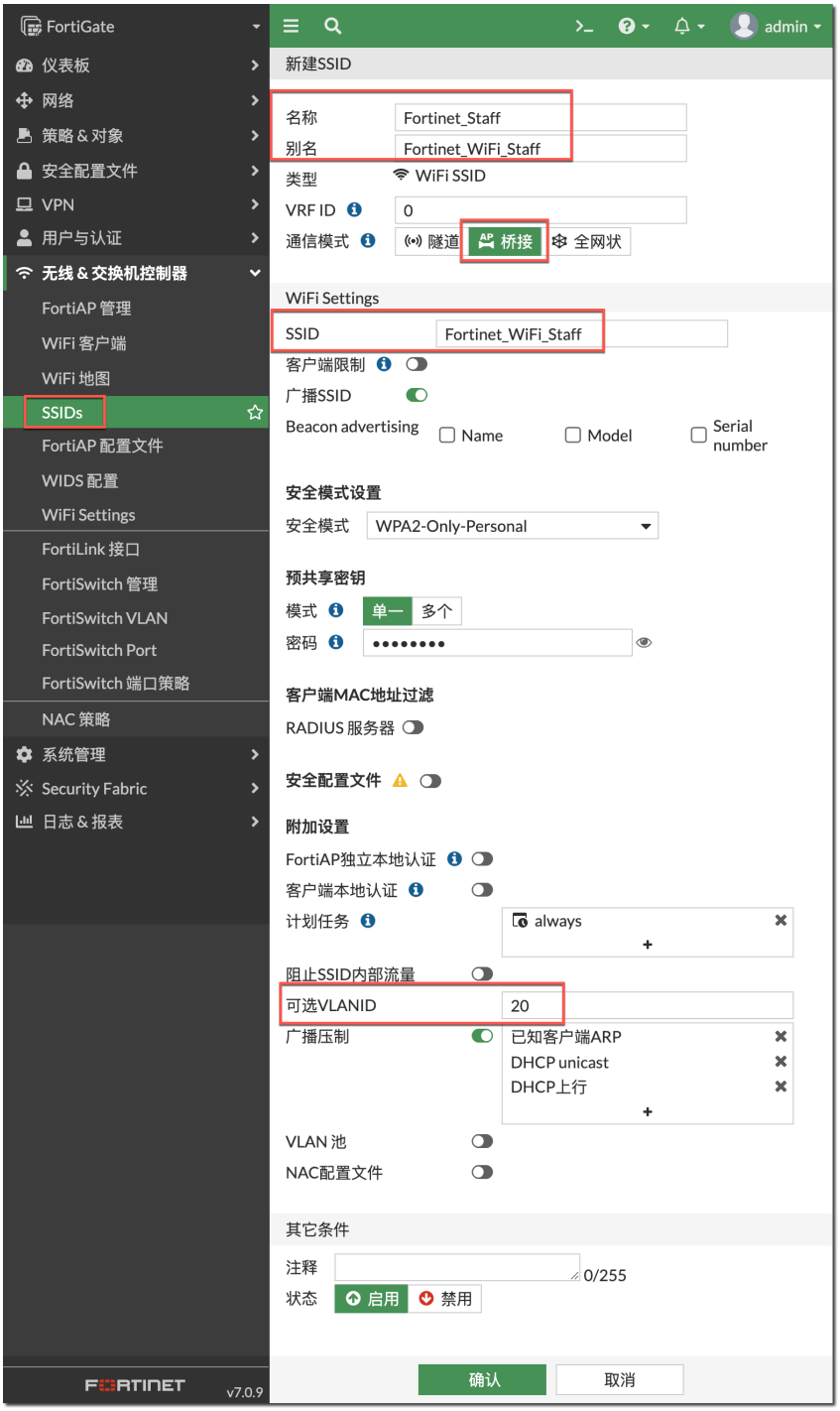
4. 编辑 FortiAP 设置，配置 FortiAP 的名称，或进行升级操作。

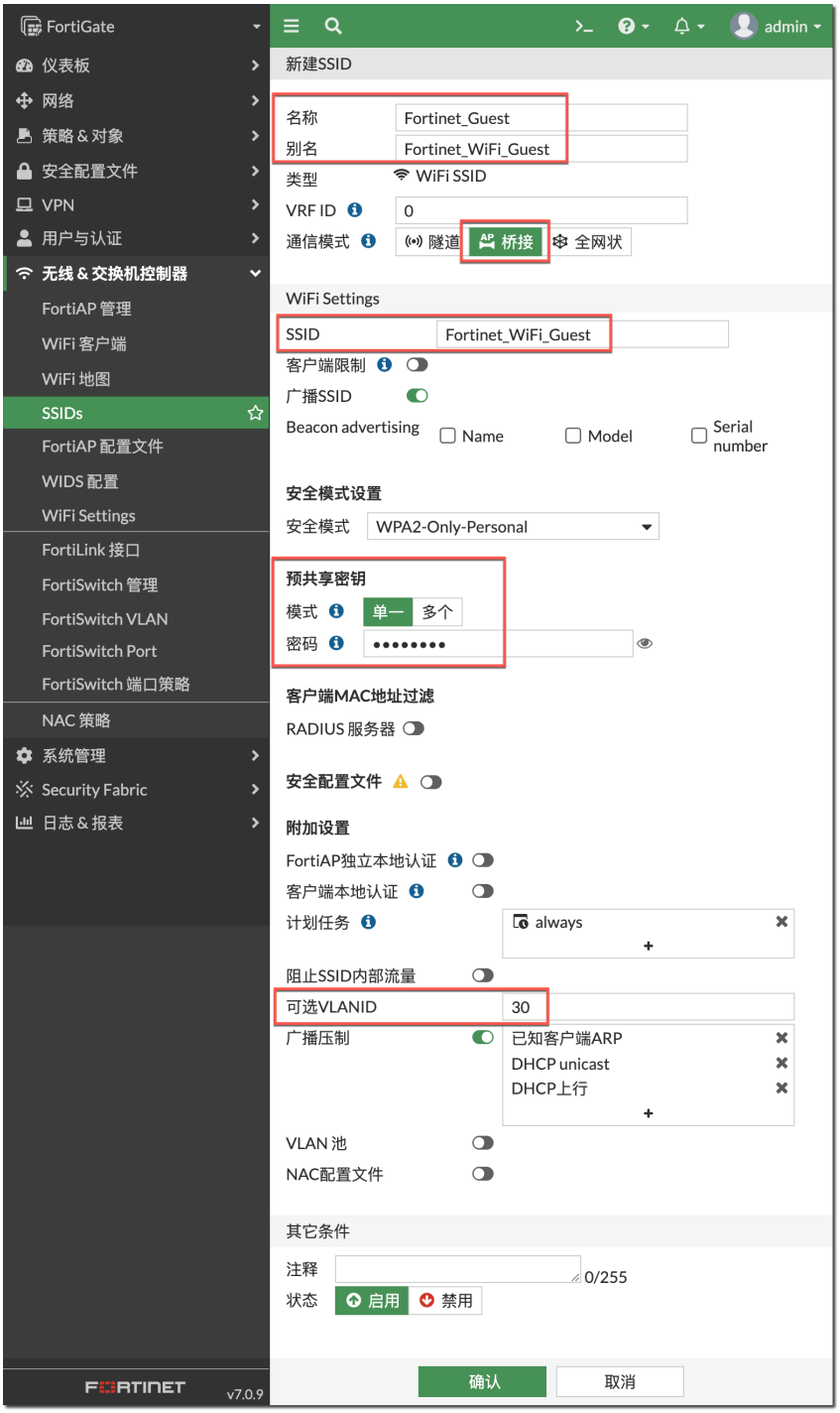


5. 进入无线 & 交换机控制器 → SSIDs 页面，点击新建按钮。



6. 新建 SSID：Fortinet_WiFi_Staff VLAN 20 用于员工 WIFI 使用，Fortinet_WiFi_Guest VLAN 30 用于访客 WiFi 使用。


FortiGate
admin
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
新建SSID
名称
Fortinet_Staff
别名
Fortinet_WiFi_Staff
类型
WiFi SSID
VRF ID
0
通信模式
隧道
桥接
全网状
WiFi Settings
SSID
Fortinet_WiFi_Staff
客户端限制
广播SSID
Beacon advertising
Name
Model
Serial number
安全模式设置
安全模式
WPA2-Only-Personal
预共享密钥
模式
单一
多个
密码
客户端MAC地址过滤
RADIUS 服务器
安全配置文件
附加设置
FortiAP独立本地认证
客户端本地认证
计划任务
always
阻止SSID内部流量
可选VLANID
20
广播压制
已知客户端ARP
DHCP unicast
DHCP上行
VLAN 池
NAC配置文件
其它条件
注释
0/255
状态
启用
禁用
确认
取消

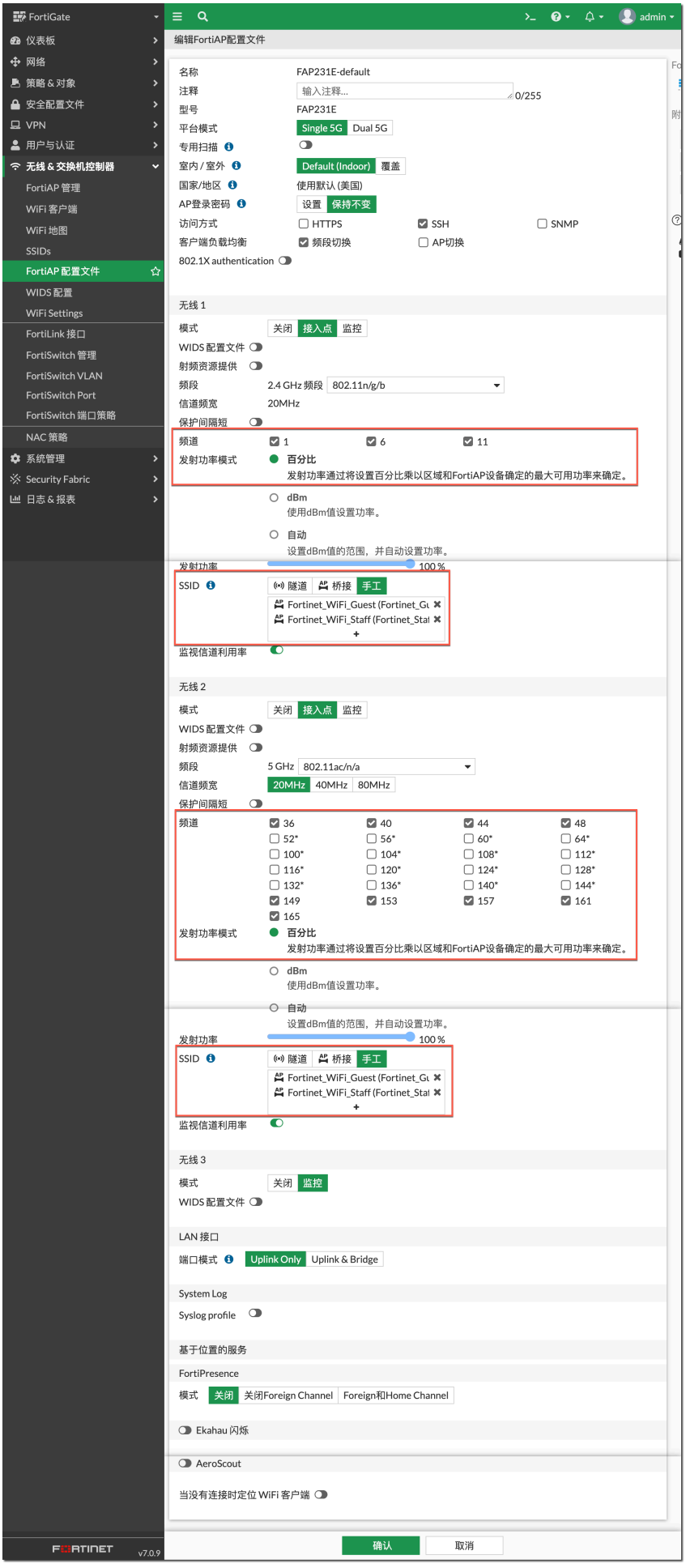




7. 将 SSID 关联到 FortiAP 上，然后让 FortiAP 发布无线 SSID 信号，这样用户就可以通过 SSID 连接到无线了。进入 FortiAP 配置文件页面，编辑 FortiAP 关联的配置文件。



8. 桥模式的 SSID 需要在 FortiAP 配置文件里面手工指定一下，另外下图标示出的配置为推荐配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
编辑FortiAP配置文件
名称 FAP231E-default
注释 输入注释... 0/255
型号 FAP231E
平台模式 Single 5G Dual 5G
专用扫描
室内 / 室外 Default (Indoor) 覆盖
国家/地区 使用默认 (美国)
AP登录密码 设置 保持不变
访问方式 HTTPS SSH SNMP
客户端负载均衡 频段切换 AP切换
802.1X authentication
无线 1
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 2.4 GHz 频段 802.11n/g/b
信道频宽 20MHz
保护间隔短
频道 1 6 11
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 2
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 5 GHz 802.11ac/n/a
信道频宽 20MHz 40MHz 80MHz
保护间隔短
频道 36 40 44 48
52* 56* 60* 64*
100* 104* 108* 112*
116* 120* 124* 128*
132* 136* 140* 144*
149 153 157 161
165
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 3
模式 关闭 监控
WIDS 配置文件
LAN 接口
端口模式 Uplink Only Uplink & Bridge
System Log
Syslog profile
基于位置的服务
FortiPresence
模式 关闭 关闭Foreign Channel Foreign和Home Channel
Ekahau 闪烁
AeroScout
当没有连接时定位 WiFi 客户端
确认
取消
FORTINET
v7.0.9

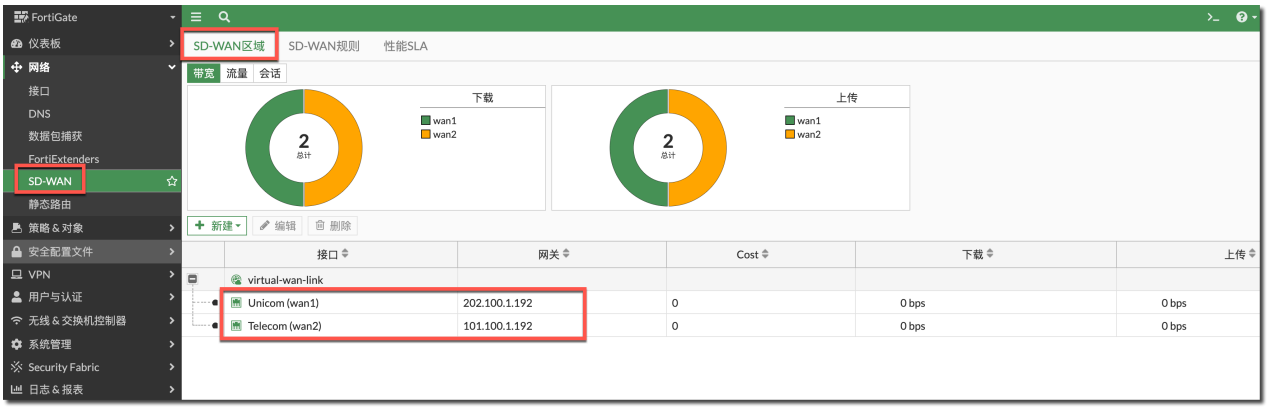


9. FortiAP 会自动关联到这个 FortiAP 配置文件里面的内容，发起 SSID 的信号。



SD-WAN 配置

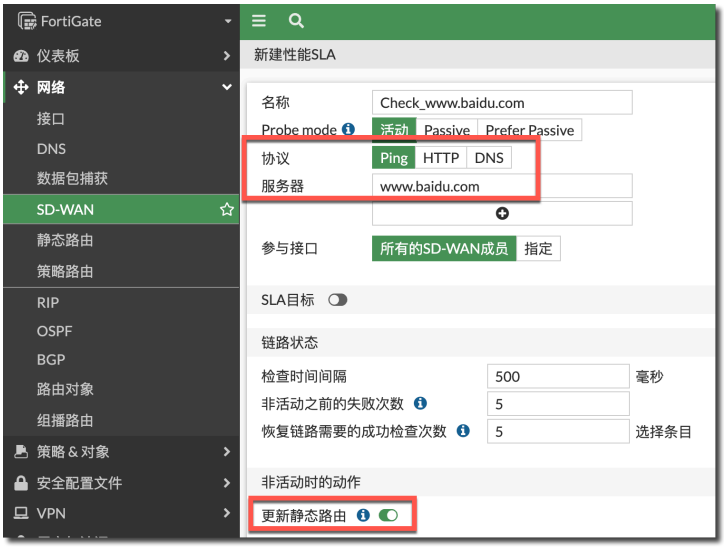
1. 将 wan1 和 wan2 加入 SD-WAN 接口，并配置各自的网关。

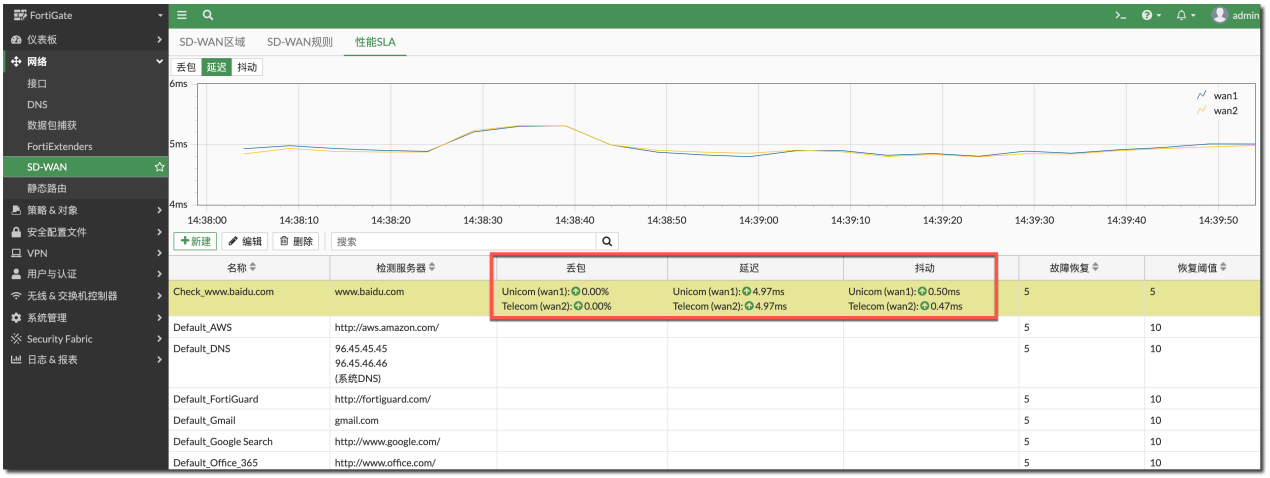


2. 添加 SD-WAN 的默认路由。



3. 配置 SD-WAN 的 SLA，以便 wan1 和 wan2 相互备份和切换。



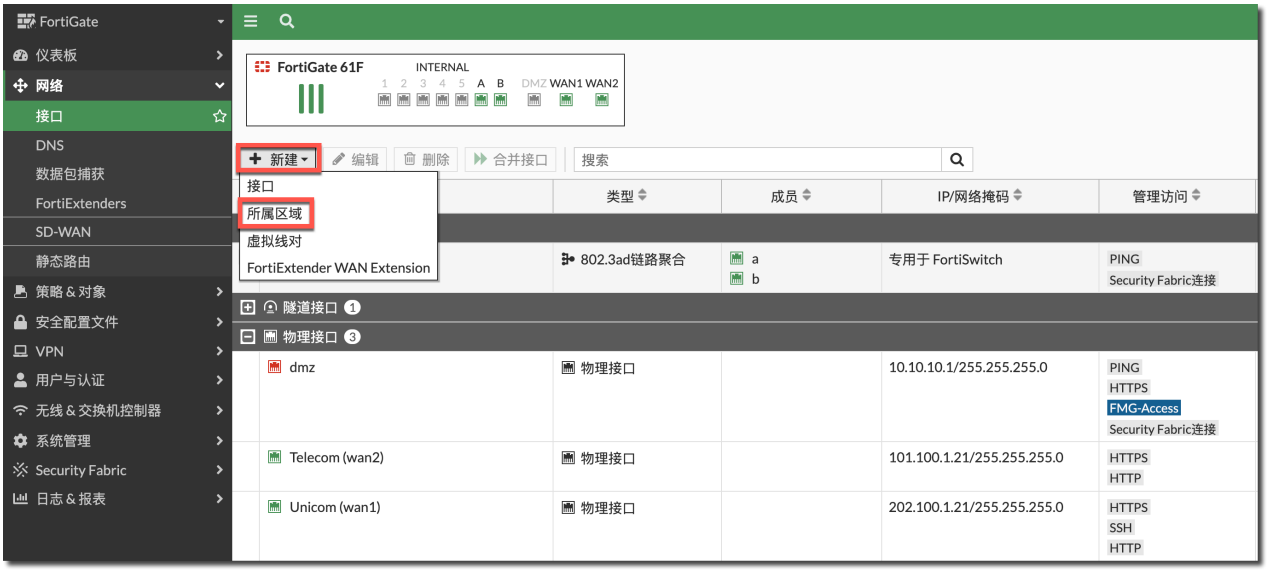


4. SD-WAN 规则可根据需求配置，这里保持默认的按照源 IP 负载均衡。

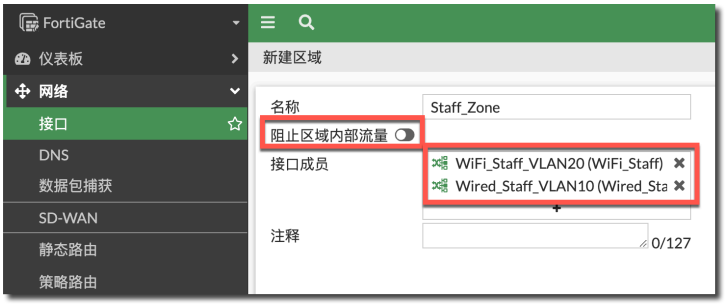


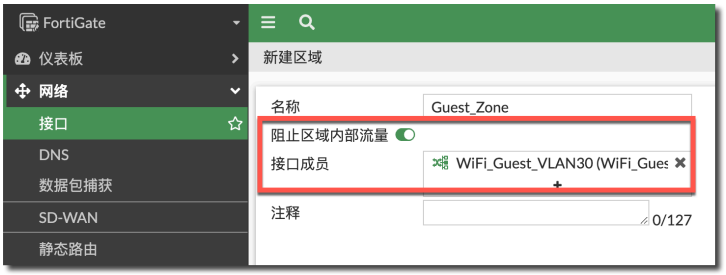
安全策略配置

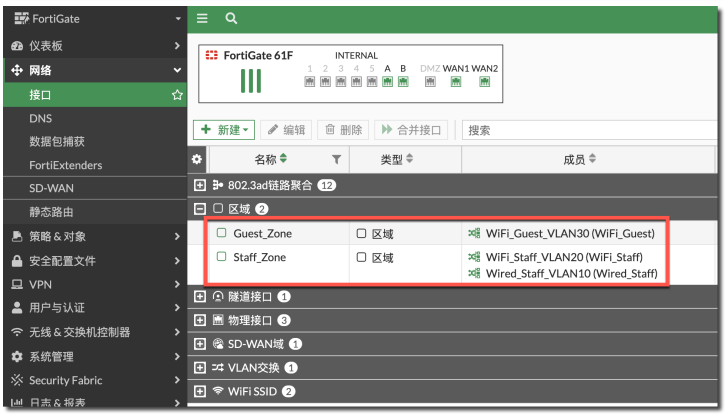
1. 需求与规划：
   - 员工的有线与无线都需要访问互联网
   - 访客的无线仅仅可以访问互联网
   - 员工的有线和无线需要互访
2. 建议将防火墙的接口划分区域，以区域的方式配置策略，有时候可以做到简化配置的目的，甚至未来替换设备的时候，策略的备份和还原也会更加简单一些。规划区域两个区域：
   - 员工网络区域：包括员工的有线 VLAN 10 和员工的无线 VLAN 20 （区域之间的隔离关闭，这样 VLAN 10 和 VLAN 20 之间的数据可通）
   - 访问网络区域：包括访问的无线 VLAN 30
3. 进入网络 → 接口页面，新建所属区域。



4. 分别创建 Staff_Zone（包含 VLAN 10 和 VLAN20）和 Guest_Zone（包含 VLAN30）。







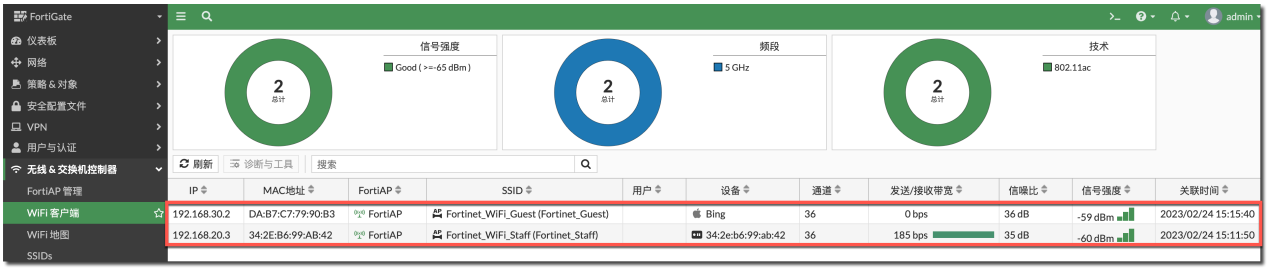
5. 如此只需要配置两条策略，就可以实现我们规划的需求，新建 Staff-Zone 到 SD-WAN 与 Guest-Zone 到 SD-WAN 的安全策略。



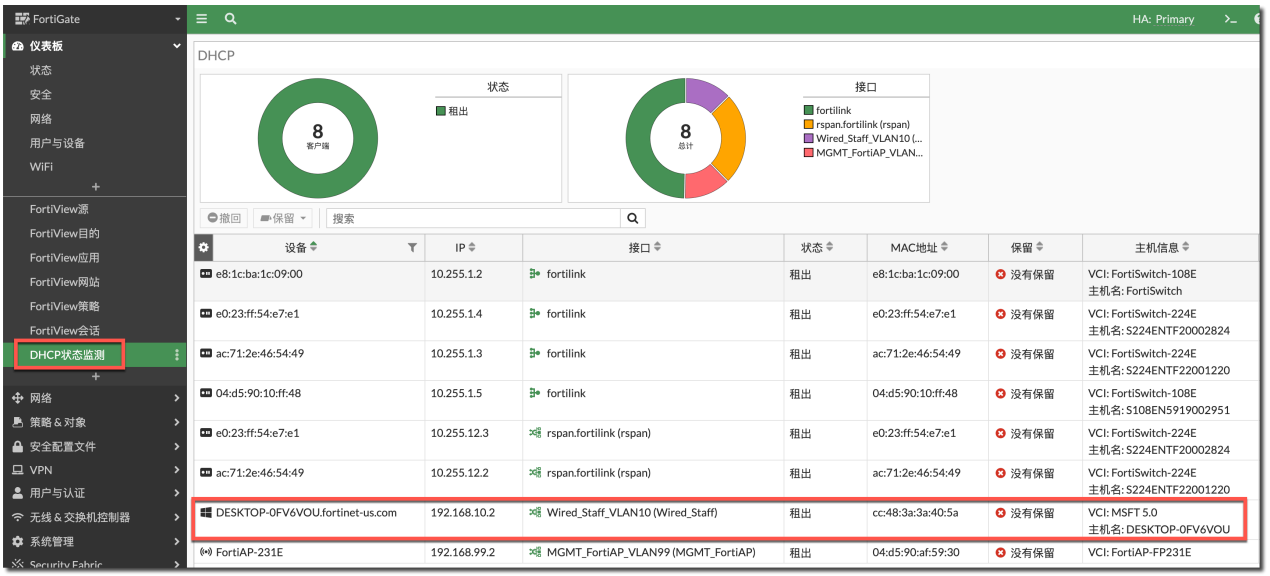
结果验证

客户端访问

1. 进入无线 & 交换机控制器 → WiFi 客户端，查看在线的无线客户端。可以看到 WiFi_Staff 和 WiFi_Guest 各有一个终端在线。



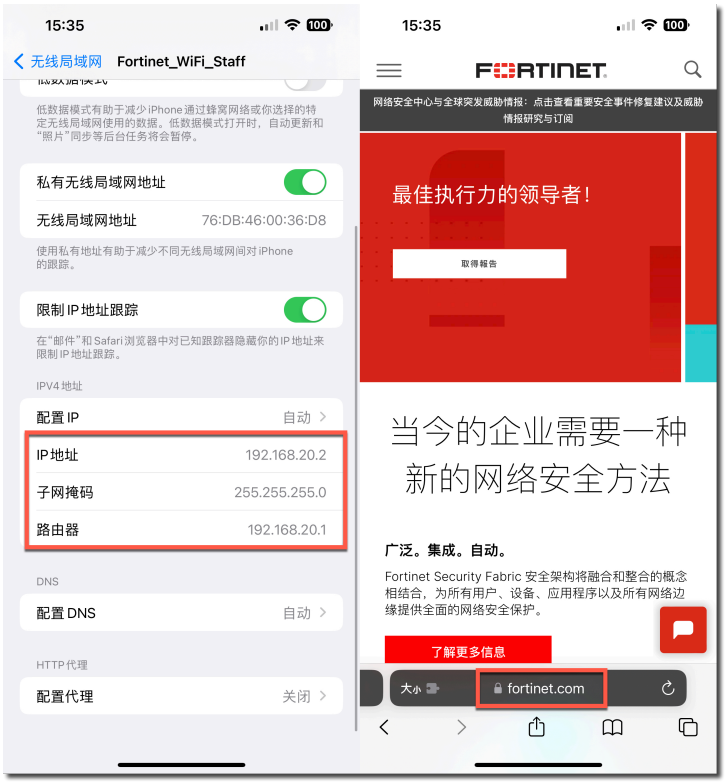
2. 在 DHCP 状态检测中查看所有的客户端信息，可以看到有线网络的 PC 也正常上线。



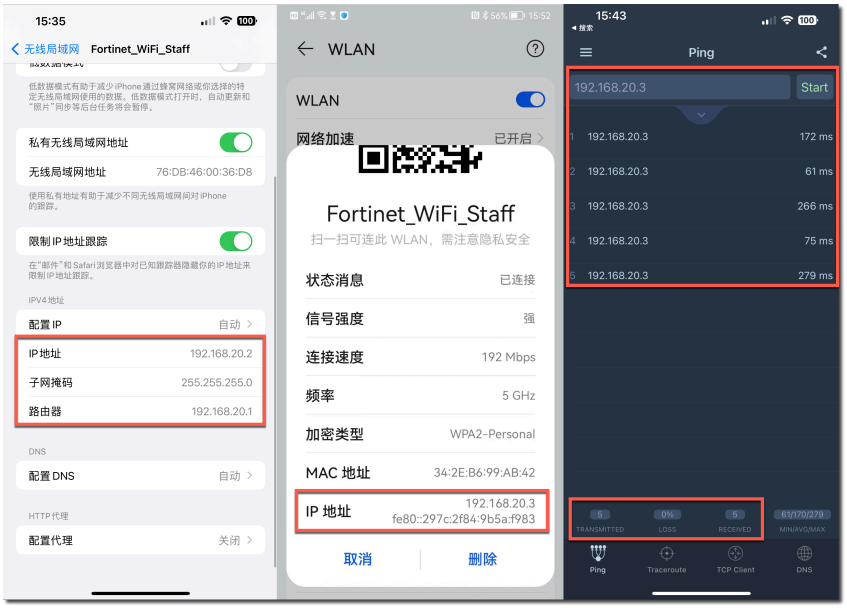
3. 有线网络（VLAN 10）用户访问互联网。



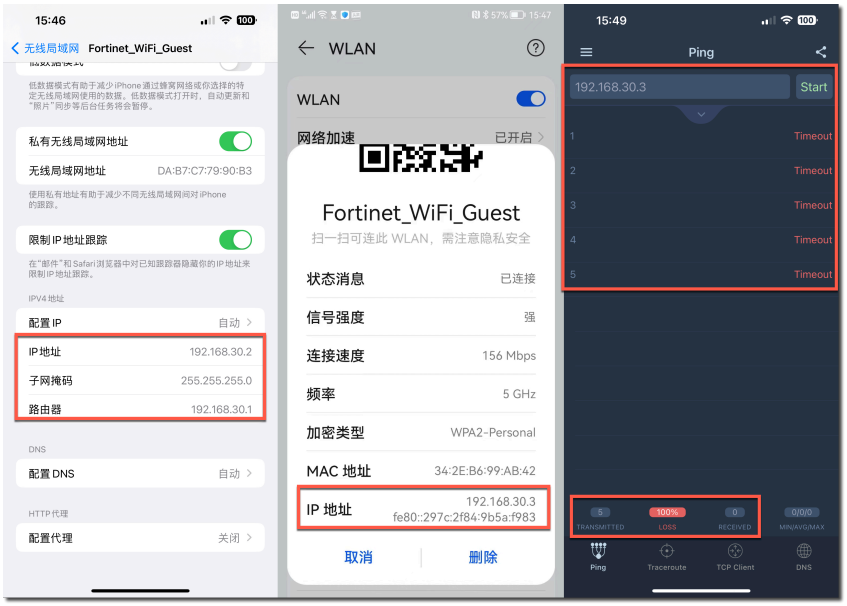
4. 无线网络用户访问互联网。



5. 使用 VLAN 20 下的客户端访问 VLAN 20 下的另一个客户端（Staff 区域）。



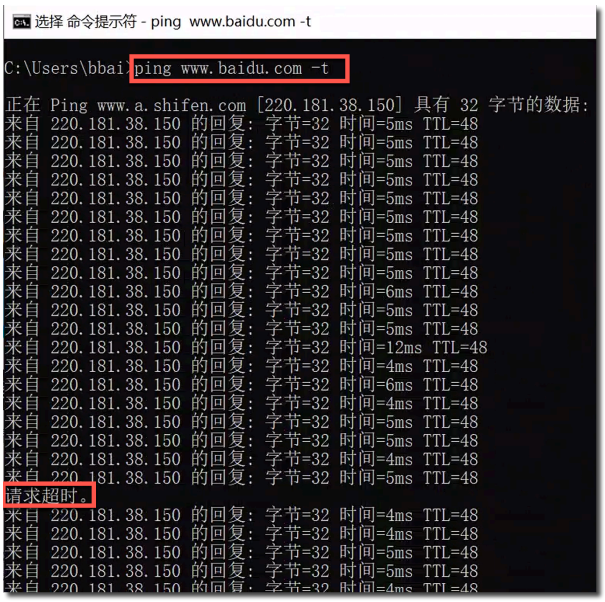
6. 使用 VLAN 30 下的客户端访问 VLAN 30 下的另一个客户端（Guest 区域），可以看到由于该区域配置了禁止相互访问，所以 VLAN 30 下的客户端是无法互相访问的。



FortiLink 切换测试

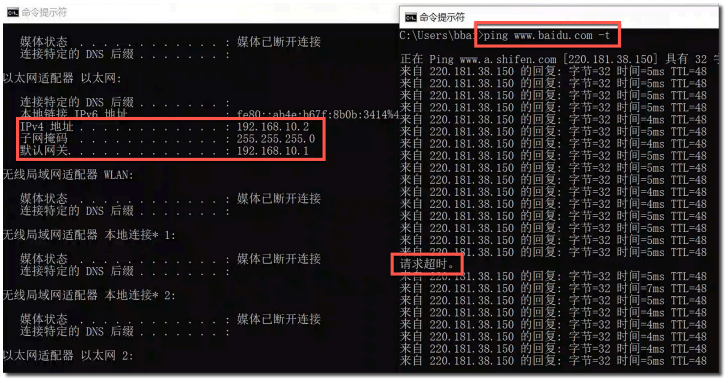
1. 在我们的测试场景下 FortiLink 聚合接口的 a 和 b 是同时 UP 起来的，如果 a 或 b DOWN（不能同时 DOWN），是否有影响业务呢?

2. 经过测试，DOWN a 或 b 接口，客户端丢包 1 个或者不丢包。

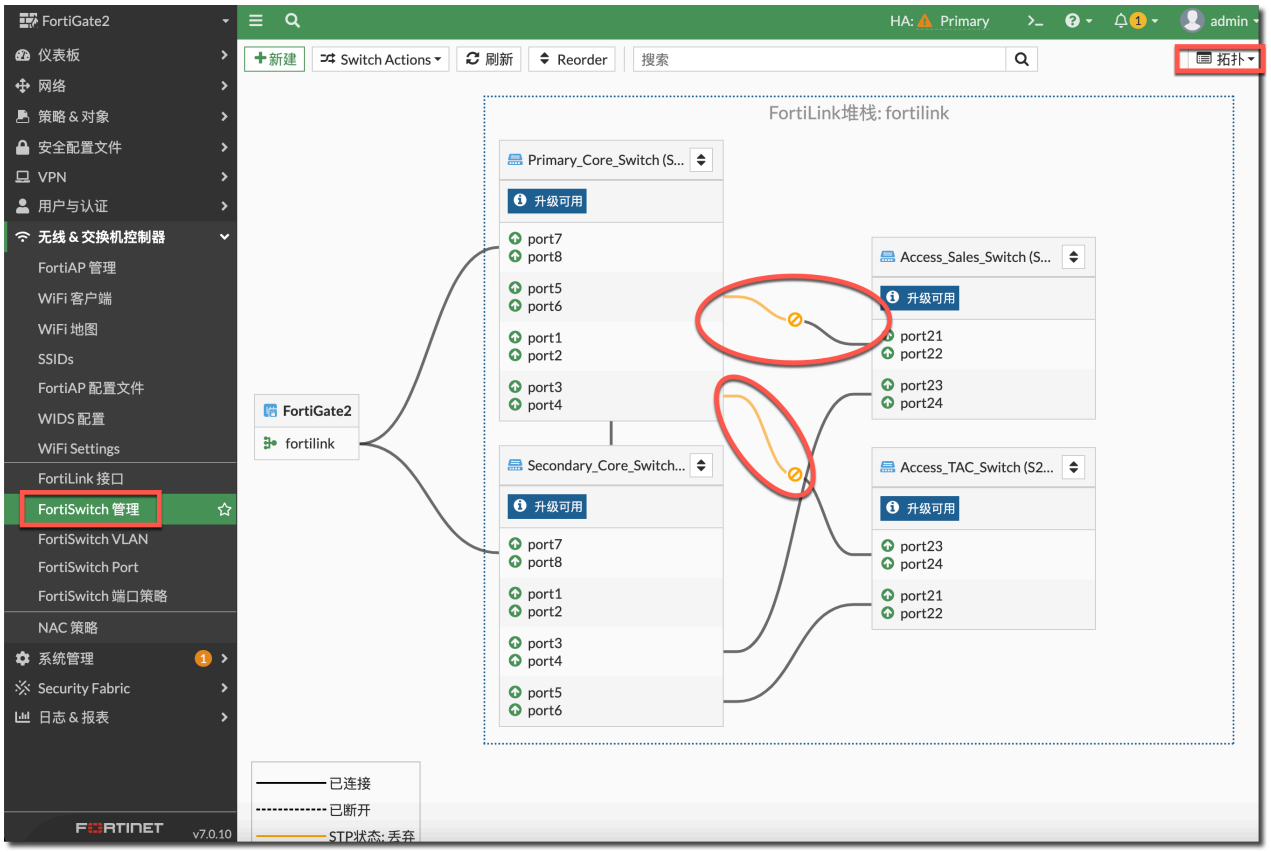


HA 切换测试

1. 重启主防火墙查看内网 PC 丢包情况，HA 切换丢包 1 个。



2. 此时查看原备（新主）防火墙的 FortiLink 交换机接管状态，流量全部会切换到备防火墙和备核心交换机上。



FortiLink 环形组网

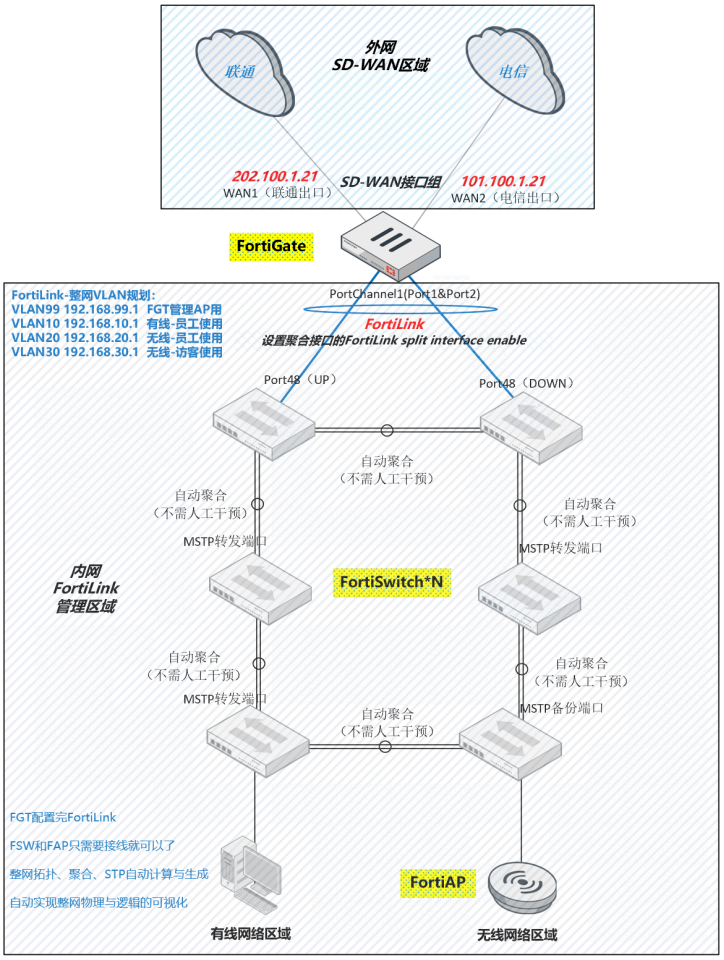
星型组网的另外一个组网方式是环形组网部署。

FortiLink 交换机环形组网的特性是设计上无单点故障，任何交换机或交换机的任意接口故障，都不会影响到整体网络的连通性，MSTP 自动收敛后业务即可恢复，可靠性比较高。

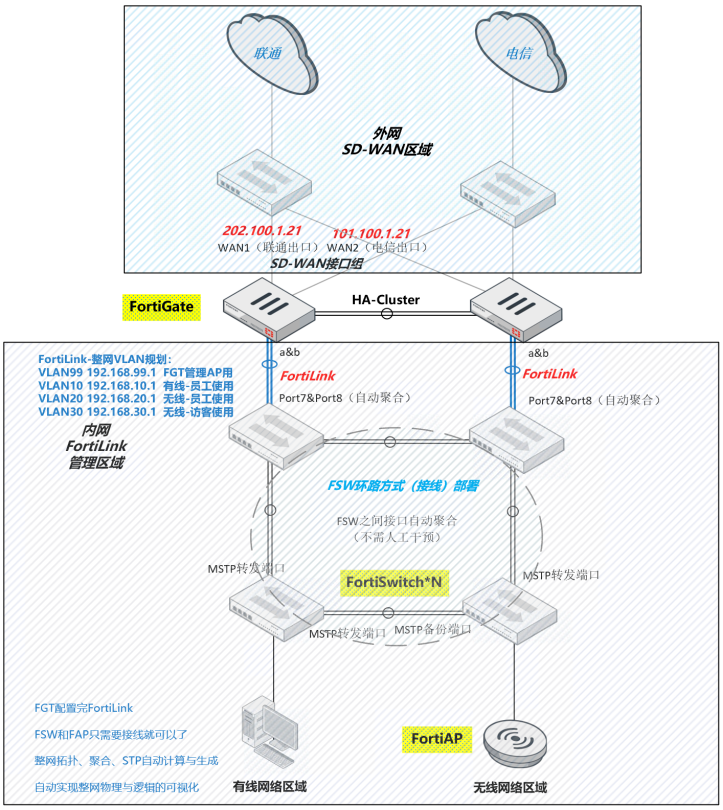
环形拓扑

单 FortiGate 组网

此方案防火墙存在单点故障。



FortiGate HA 组网



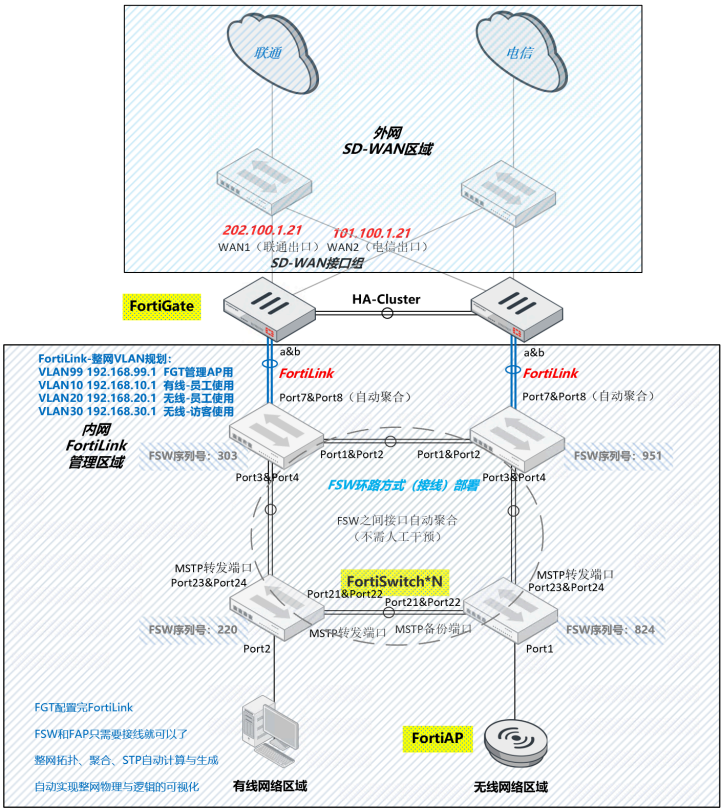
1. 环形组网是比较好的方式，唯一的问题是环路太大了，一旦网络中出现接口震荡等情况，MSTP 会出现反复的计算和收敛，一个环内的交换机越多 STP 速度就越慢，通常不推荐超过 16 台的 FortiLink 环（最多 20 台 FortiGate 的环路）。
2. 有没有更好的解决办法呢？实际上需要将星型组网和环形组网结合起来就可以比较好的解决环过大的问题。我们可以取名叫做星型环形混合组网，请参考 SAA-FortiLink → FortiLink 部署方案说明和举例 → FortiLink 部署之星型环形混合组网 ↗章节。

FortiLink 环形组网配置举例

网络需求

FortiGate 运行 HA-Cluster，核心交换和接入交换机采用全聚合接口（FortiSwitch 全自动聚合，接线即可，无需手工配置聚合接口）对接的环形组网结构。

网络拓扑



1. 两台 FortiGate 配置为 HA 主备模式。
2. 整体规划：
   - 外网 IP 网段规划：
     - WAN1 联通出口：IP 202.100.1.21，GW 202.100.1.192
     - WAN2 电信出口：IP 101.100.1.21，GW 101.100.1.192
   - WAN1 和 WAN2 合并成 SD-WAN 接口。
   - 内网 IP 网段规划，FGT 通过 FortiLink + Capwap 协议统一的接管 FortiSW 和 FortiAP，为有线和无线统一进行管理和业务处理：
     - 有线网络规划：VLAN 10，192.168.10.1
     - 无线网络规划：员工 WIFI VLAN 20，192.168.20.1
     - 无线网络规划：访客 WIFI VLAN 30，192.168.30.1
     - 管理 FortiAP：VLAN 99，192.168.99.1
   - FortiLink 互联部分规划：
     - FortiGate 的 a 和 b 做聚合接口与 Fortiswitch 的后两个接口互联即可实现聚合接口的 FortiLink 管理。
     - 加入 FortiLink 的接口自动变为聚合口成员。

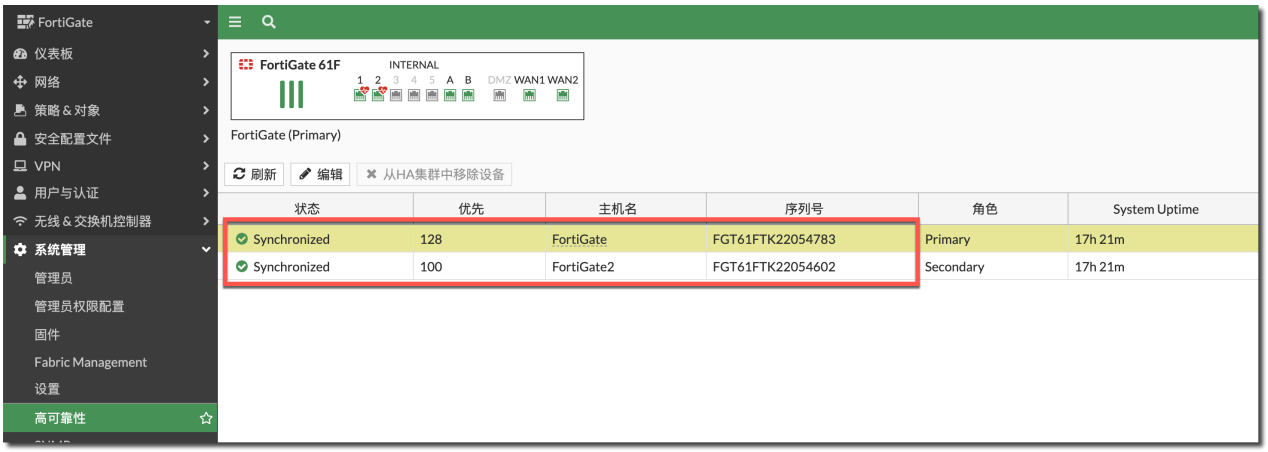
配置步骤

准备事项

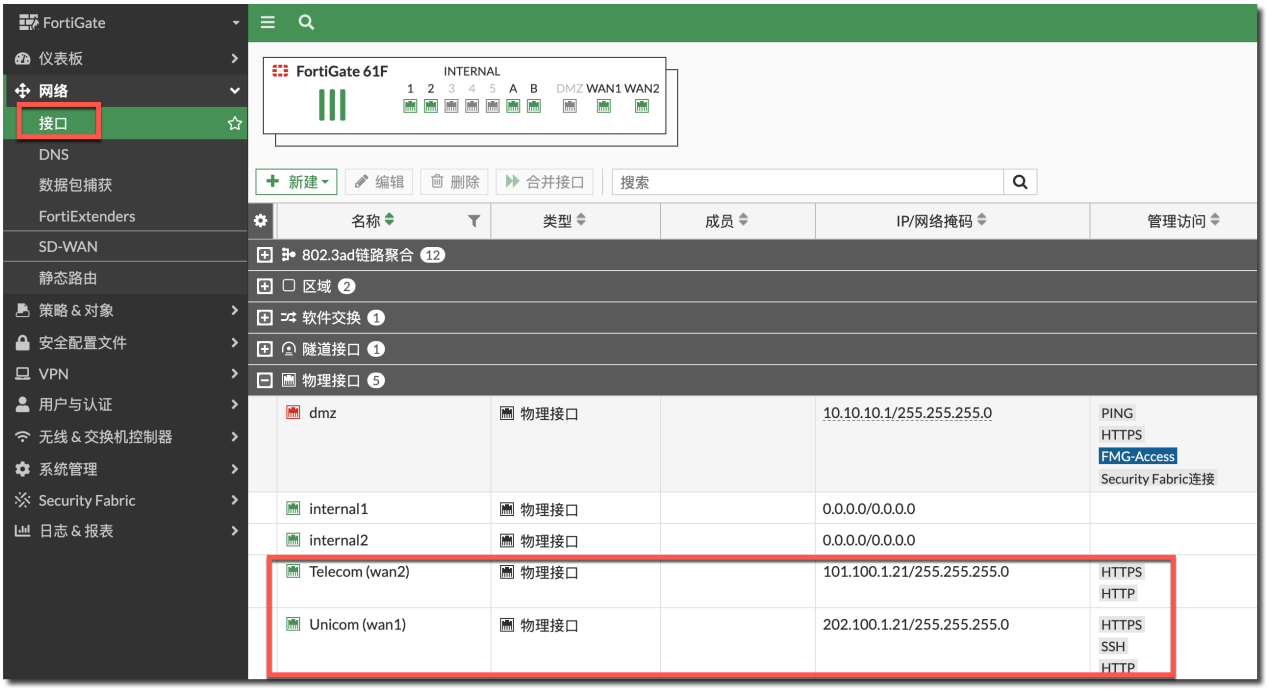
1. 首先确认 FortiGate 如下全局配置已开启（一般默认为开启状态）。

```
config system settings
    set gui-switch-controller enable
end
```

2. 配置两台 FortiGate 的 HA 主备模式（配置方法请参考 HA 双机热备 → HA 典型基础配置 ↗）。

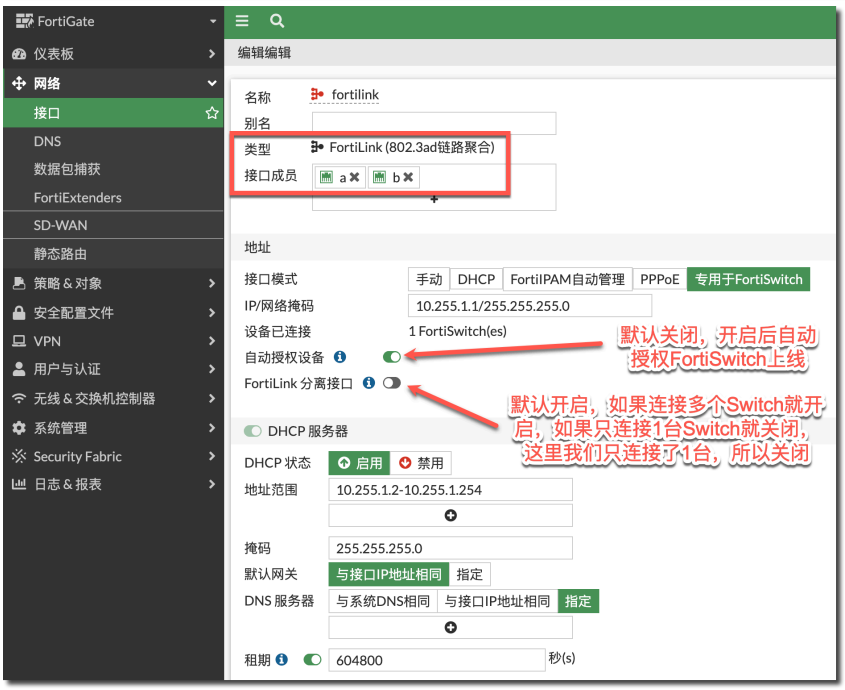


3. 配置 wan1 和 wan2 的 IP 地址。



4. 配置 FortiLink 实现 FortiSwith 上线，我们只需要将用于 FortiLink 的成员接口加入 FortiLink 接口即可。此拓扑中，FortiGate 用于连接 FortiSwitch 的接口为 a 和 b，编辑 FortiLink 接口，将 a 和 b 加入 FortiLink（某些产品型号会默认将两个接口加入 FortiLink），开启自动授权设备，关闭 FortiLink 分离接口，其他配置保持默认，可以看到 FortiLink 为一个聚合接口。


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 8
fortilink
802.3ad链路聚合
专用于 FortiSwitch
PING
Security Fabric连接
隧道接口 1
物理接口 5
a
物理接口
0.0.0.0/0.0.0.0
b
物理接口
0.0.0.0/0.0.0.0
dmz
物理接口
10.10.10.1/255.255.255.0
PING
HTTPS
FMG-Access
Security Fabric连接
Telecom (wan2)
物理接口
101.100.1.21/255.255.255.0
HTTPS
HTTP
Unicom (wan1)
物理接口
202.100.1.21/255.255.255.0
HTTPS
HTTP

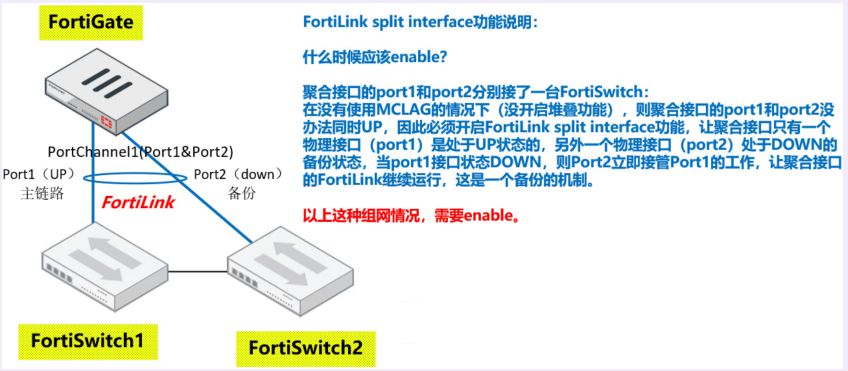

FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
编辑编辑
名称
fortilink
别名
类型
FortiLink (802.3ad链路聚合)
接口成员
a
b
地址
接口模式
手动
DHCP
FortiIPAM自动管理
PPPoE
专用于FortiSwitch
IP/网络掩码
10.255.1.1/255.255.255.0
设备已连接
1 FortiSwitch(es)
自动授权设备
默认关闭，开启后自动授权FortiSwitch上线
FortiLink 分离接口
默认开启，如果连接多个Switch就开启，如果只连接1台Switch就关闭，这里我们只连接了1台，所以关闭
DHCP 服务器
DHCP 状态
启用
禁用
地址范围
10.255.1.2-10.255.1.254
掩码
255.255.255.0
默认网关
与接口IP地址相同
指定
DNS 服务器
与系统DNS相同
与接口IP地址相同
指定
租期
604800
秒(s)


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
FortiGate 61F
INTERNAL
1 2 3 4 5 A B
DMZ WAN1 WAN2
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 12
fortilink
802.3ad链路聚合
a
b
专用于 FortiSwitch
PING
Security Fabric连接

重要

FortiLink 分离接口什么时候应该开启?

分离接口的意是，如果聚合接口的 port1 和 port2 分别连接了一台 SW，在没有使用 MCLAG 的情况下（没开启堆叠功能），则聚合接口的 port1 和 port2 没办法同时 UP（同时开启聚合接口无法正常工作），因此必须开启 FortiLink 分离接口功能，让聚合接口只有一个物理接口（port1）是处于 UP 状态的，另外一个物理接口（port2）处于 DOWN 的备份状态，当 port1 接口状态 DOWN，则 Port2 立即接管 Port1 的工作，让聚合接口的 FortiLink 继续运行，这是一个聚合接口在 FortiLink 管理时候的备份机制。



重要

FortiLink split interface 什么时候应该 disable?

如果 FGT 的聚合接口只接到了一台交换机上，并没有跨交换机聚合（没有 MCLAG，也不需要 FortiLink split interface），因此这种情况应该让聚合接口的两个物理接口 port1 和 port2 同时 UP，两个接口同时工作起来，以便提供更高带宽和备份的作用。



相关 CLI:

```
NTP:
config system ntp
    set ntpsync enable
    set server-mode enable
    set interface "fortilink"
end
-------------------------------------------------------------------

DHCP:
config system dhcp server
    edit 2
        set dns-service local
        set ntp-service local
        set default-gateway 10.255.1.1
        set netmask 255.255.255.0
        set interface "fortilink"
        config ip-range
            edit 1
                set start-ip 10.255.1.2
                set end-ip 10.255.1.254
            next
        end
        set vci-match enable
        set vci-string "FortiSwitch" "FortiExtender"
    next
```

```
end
————————————————————————————————————————————
FortiLink:
config system interface
    edit "fortilink"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set type aggregate
        set member "a" "b"
        set lldp-reception enable
        set lldp-transmission enable
        set auto-auth-extension-device enable
        set fortilink-split-interface disable
        set switch-controller-nac "fortilink"
        set switch-controller-dynamic "fortilink"
        set swc-first-create 255
    next
end
```

连接 FortiLink

1. 配置完 FortiGate 之后，这个时候再去接 FortiGate 的 FortiLink 到 FortiSwitch 的线，以及 FortiSwitch 之间的线，严格按照拓扑图，从核心交换机往接入交换机这样的顺序接线。FortiSwitch 如果以前有使用过，建议都恢复一下出厂设置，然后再加入到 FortiLink。
2. FortiSwitch 的靠后接口一般都默认开启了 FortiLink 接口属性，如果接前面的接口可能会无法加入到 FortiLink，因此 FortiSwitch 的 FortiLink 最好都使用靠后的接口进行接线。FortiSwitch 各型号默认开启 FortiLink 功能的接口如下：

FortiSwitch 型号	默认开启 FortiLink 的接口
FS-108D-POE	port9–port10
FS-108E, FS-108E-POE, FS-108E-FPOE	port7–port10
FSR-112D-POE	port5–port12
FS-124D, FS-124D-POE	port23–port26
FSR-124D	port1-port4, port21–port28
FS-124E, FS-124E-POE, FS-124E-FPOE	port21–port28
FS-148E, FS-148E-POE	port21–port52
FS-224D-POE	port21–port24
FS-224D-FPOE	port21–port28
FS-224E, FS-224E-POE	port21–port28
FS-248D, FS-248D-FPOE	port45–port52
FS-248D-POE	port47–port50
FS-248E-POE, FS-248E-FPOE	port45–port52
FS-424D, FS-424D-POE, FS-424D-FPOE	port23–port26
FS-424E-Fiber	port1-port30
FS-426E-FPOE-MG	port23-port30
FS-448D, FS-448D-POE, FS-448D-FPOE	port45–port52
FS-524D, FS-524D-FPOE	port21–port30
FS-548D	port39–port54
FS-548D-FPOE, FS-548DN	port45–port54
FS-1024D	port1–port24
FS-1048D, FS-1048E	port1–port52
FS-3032D, FS-3032E	port1–port32

3. 如果 FortiSwitch 无法被 FortiGate 管理，登陆到 FortiSwitch 上进行确认接口是否开启了 FortiLink。

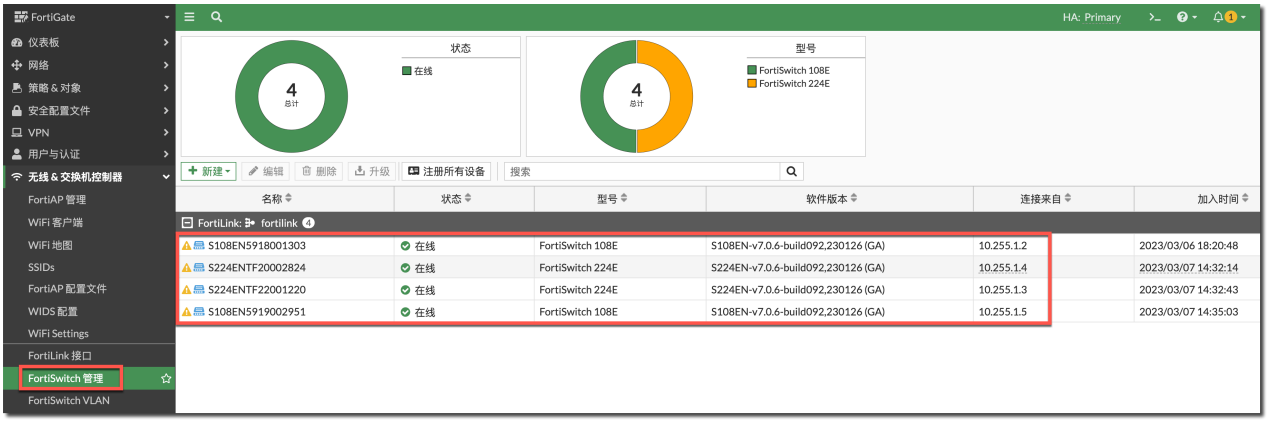
```
FortiSwitch # show switch interface port9
config switch interface
    edit "port9
        set auto-discovery-fortilink enable    //接口下又此命令的接口才可以自动加入到FortiLink中//
    next
end
```

4. 交换机之间如果接多根线，会自动进行聚合，但是 FortiLink 自动聚合会花比较长的时间，建议可以先接单线，等 FortiLink 的拓扑较快的形成结束之后，再增加彼此之间的另外一根线，再进行自动聚合，这样效率会比较高一些。

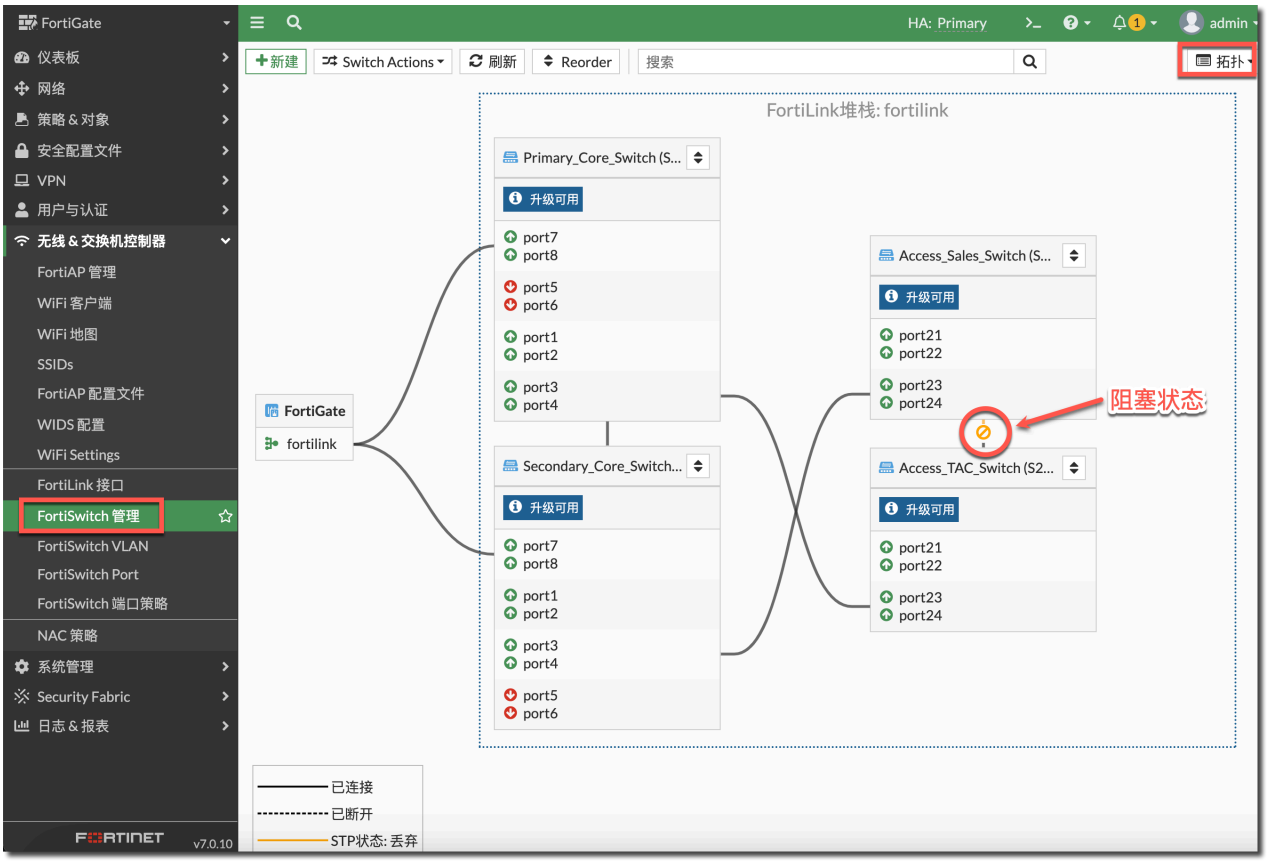
5. 接线是很重要的步骤，仔细看拓扑图，做好接口标签，不要接错。

6. 接好线之后，FortiSwitch 和 FortiGate 之间会自动在后台进行 FortiLink、CAPWAP、MSTP、聚合接口的协商加入等操作，等 FortiLink 的管理结束之后，在 FortiGate 上会看到一个网站的 FortiSwitch 拓扑图，和我们规划的拓扑图应该是一致的，可以仔细核对接线是否正确。

7. 在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，查看交换机被管理状态，可以看到 4 台交换机均已上线。

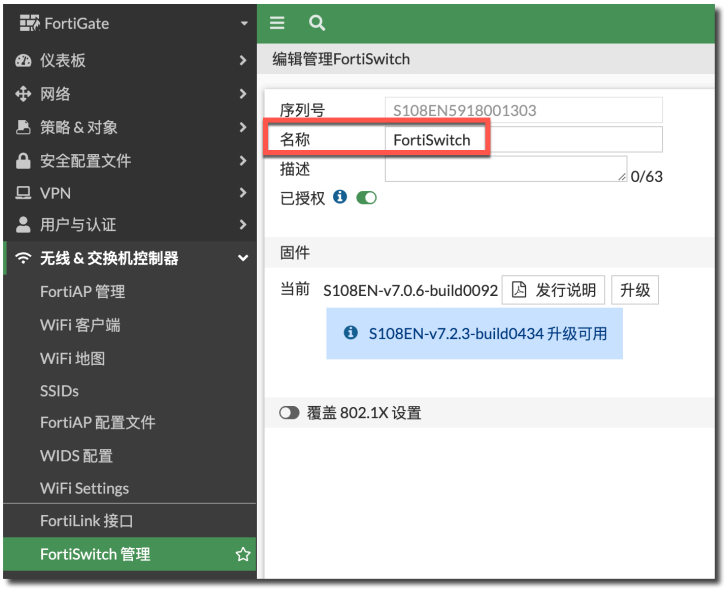


8. 修改视图为拓扑，可以更直观的看到当前防火墙与交换机的接口连接。



9. 交换机加入到 FortiGate 的 FortiLink，会先清除配置再重启，然后从 standalone 的状态进入到“Fortilink remote control”的状态，后续所有的配置都通过 FortiGate 下发即可。

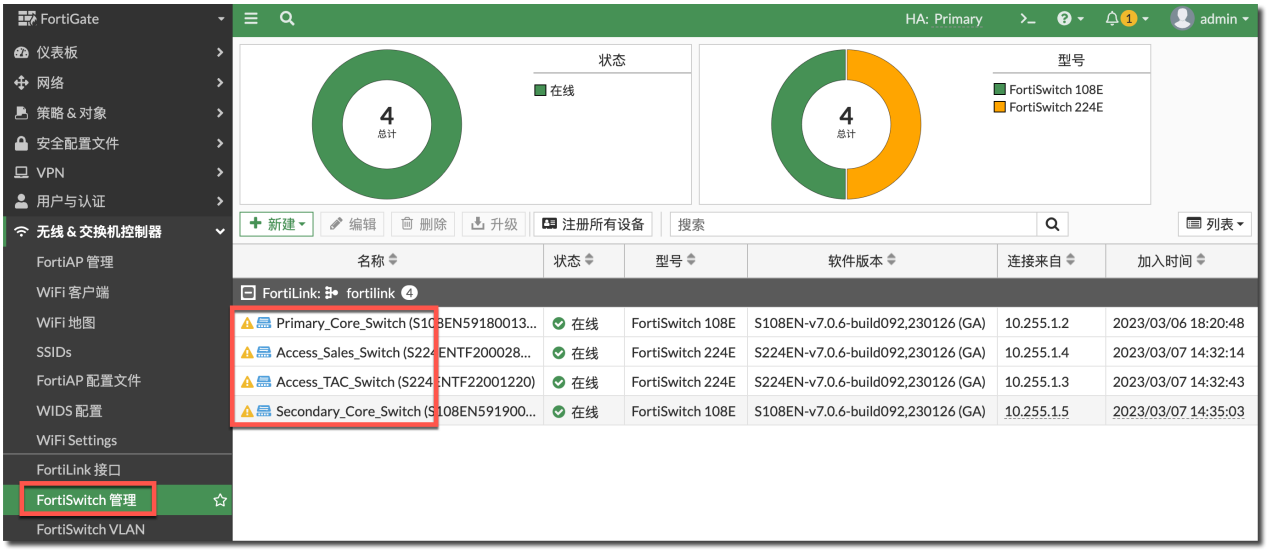
0. 编辑交换机，可以修改交换机名字以及升级版本，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。



1. FortiSwitch 之间接多根线进行 FortiLink 互联，会默认自动聚合。其实接一根网线也是聚合的，只是聚合组里面只有一个接口而已，因此多根网线直连，这些直连的线都会自动加入到一个聚合接口里面（注意不能跨交换机，跨交换机的聚合是 MCLAG）。

```bash
FortiGate # exec ssh admin@10.255.1.2
Warning: Permanently added '10.255.1.2' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

S108EN5918001303 # show switch trunk
path=switch, objname=trunk, tablename=(null), size=752
config switch trunk
    edit "GT61FTK22054783"
        set mode lacp-active
        set auto-isl 1
        set fortilink 1
        set members "port7" "port8"
    next
    edit "4ENTF20002824-0"
        set mode lacp-active
        set auto-isl 1
        set members "port5" "port6"
    next
    edit "4ENTF22001220-0"
        set mode lacp-active
        set auto-isl 1
        set members "port3" "port4"
    next
    edit "8EN5919002951-0"
        set mode lacp-active
        set auto-isl 1
        set members "port1" "port2"
    next
end
```

2. 另外建议给每台交换机命名一个名字，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。



STP 状态查看

1. FortiLink 之间的交换机互联，如果不用 MCLAG 的话，本质上是 MSTP（多实例生成树）在整网统一的进行接口环路计算，从而实现网络的备份，与快速收敛恢复。 可以通过 FGT 的命令行界面用 Telent/SSH 到 FortiSwitch 上并分别查看四台 FortiSwitch 的 STP 状态。

   - 主核心交换机（303）上查看：

```
FortiGate # exec ssh admin@10.255.1.2
Warning: Permanently added '10.255.1.2' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

Primary_Core_Switch #  diagnose stp instance list
MST Instance Information, primary-Channel:
...
Instance ID 15
  Config         Priority 24576, VLANs 4094
                 Bridge MAC e81cba1c0900
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 0
                 (This bridge is the regional root)
  //直连主FGT的FSW会自动将网桥优先级降低到：Priority 24576，因此整个FortiLink二层网络中，自己就是根网桥//
  TCN Events     Triggered 10 (0d 0h 24m 59s ago), Received 67 (0d 0h 24m 26s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   ______  _________  _________  ___________  ___________  ________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  GT61FTK22054783    2G      10000      128        DESIGNATED   FORWARDING   ED
  //与主FortiGate_FortiLink互联的自动生成的聚合接口，STP为指定接口，转发状态//
  4ENTF22001220-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号220结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态//
  8EN5919002951-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与备核心交换机303互联的自动生成的聚合接口，STP为指定接口，转发状态//
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

   - 备核心交换机（951）上查看：

```
FortiGate # exec ssh admin@10.255.1.5
Warning: Permanently added '10.255.1.5' (ED25519) to the list of known hosts.
admin@10.255.1.2's password:

Secondary_Core_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
...
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC 04d59010ff48
                 //自己的MAC
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口 8EN5918001303-0
```

```
  TCN Events     Triggered 4 (0d 0h 41m 4s ago), Received 2 (0d 0h 41m 32s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   ______  _________  _________  ___________  ___________  ________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  8EN5918001303-0    2G      1          128        ROOT         FORWARDING   EN
  //与主核心交换机303互联的自动生成的聚合接口，STP为root接口，转发状态
  4ENTF20002824-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //与序列号824结尾的FSW互联的自动生成的聚合接口，STP为指定接口，转发状态
  GT61FTK22054602    2G      10000      128        DESIGNATED   FORWARDING   ED
  //与备FortiGate_FortiLink互联的自动生成的聚合接口，STP为指定接口，转发状态（但由于连的是备防火墙，实际物理口是discarding状态）
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

- 接入交换机（220）上查看：

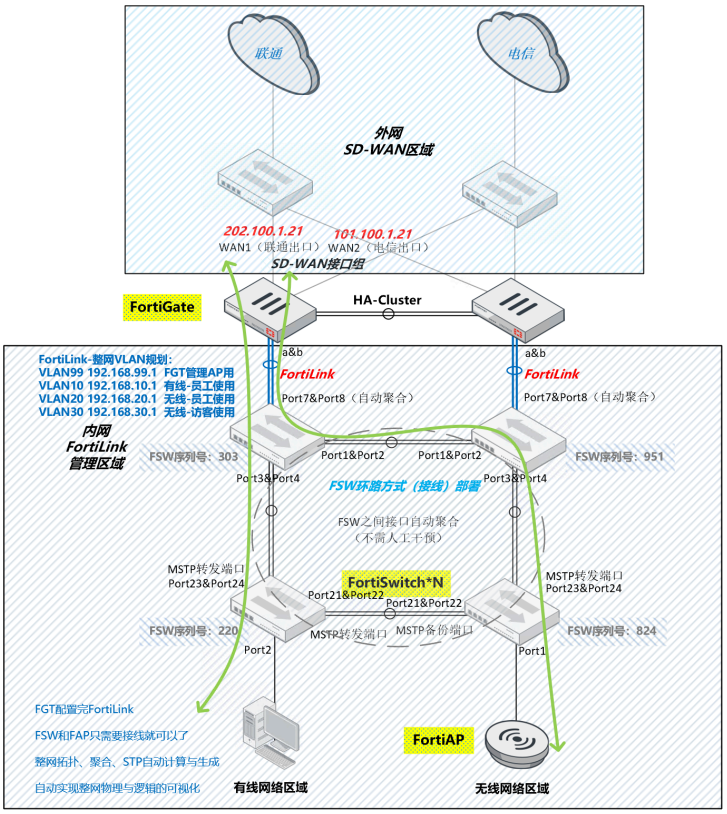
```
FortiGate # exec ssh admin@10.255.1.3
admin@10.255.1.3's password:

Access_TAC_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC ac712e465448
                 //自己的MAC地址
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口8EN5918001303-0
  TCN Events     Triggered 1 (0d 1h 12m 46s ago), Received 6 (0d 1h 9m 25s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   ______  _________  _________  ___________  ___________  ________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  8EN5918001303-0    2G      1          128        ROOT         FORWARDING   EN
  //接主核心交换机303，STP为根接口，转发状态
  4ENTF20002824-0    2G      1          128        DESIGNATED   FORWARDING   EN
  //接Sales接入交换机824，STP为指定接口，转发状态
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

- 接入交换机（824）上查看：

```
FortiGate # exec ssh admin@10.255.1.4
admin@10.255.1.4's password:

Access_Sales_Switch # diagnose stp instance list
MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC e023ff54e7e0
                 //自己的MAC地址
  Regional Root  MAC e81cba1c0900, Priority 24576, Path Cost 1, Root Port 8EN5918001303-0
  //互联根网桥的信息：根网桥MAC e81cba1c0900，根网桥优先级 24576，互联根网桥的接口8EN5918001303-0
  TCN Events     Triggered 1 (0d 1h 22m 45s ago), Received 7 (0d 1h 18m 58s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   ______  _________  _________  ___________  ___________  ________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  4ENTF22001220-0    2G      1          128        ALTERNATIVE  DISCARDING   EN
  //接TAC接入交换机303，STP为备份接口ALTERNATIVE DISCARDING，处于阻塞状态
  8EN5919002951-0    2G      1          128        ROOT         FORWARDING   EN
  //接备核心交换机951，STP为根接口，转发状态
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

2. FortiLink 组网中 STP 是很重要部分，通常 STP 是一个稳定的状态，一旦出现 FortiLink 不稳定，则需要观察 STP 的状态和选举结果，从中找到相关线索，以便查出为何 STP 不稳定的原因，对于排错来说是很重要的。

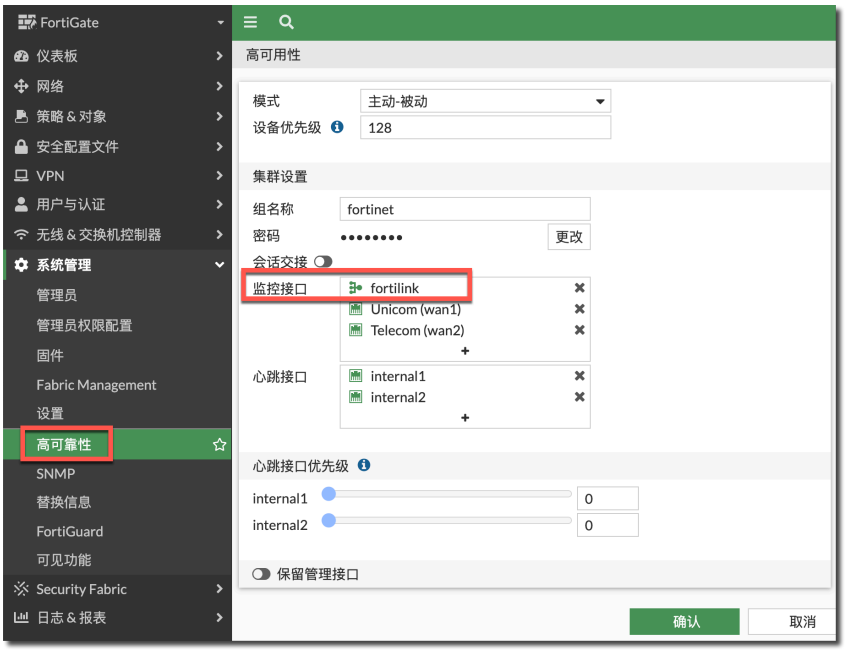
3. FortiLink + STP 稳定状态的实际数据转发拓扑图如下。



4. 在 FortiGate 查看交换机管理状态。

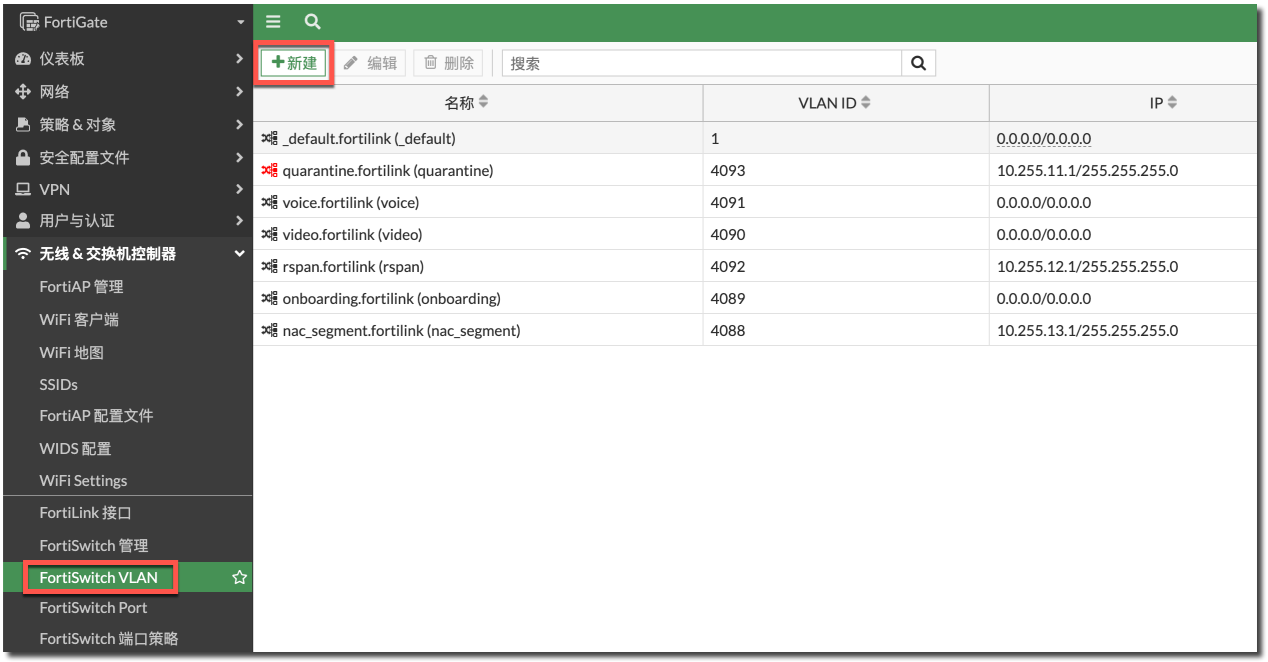
```
FortiGate # execute switch-controller get-conn-status
Managed-devices in current vdom root:
FortiLink interface : fortilink
SWITCH-ID         VERSION           STATUS         FLAG   ADDRESS              JOIN-TIME              NAME
S108EN5918001303  v7.0.6 (092)      Authorized/Up   -   10.255.1.2      Mon Mar  6 18:20:48 2023    Primary_Core_Switch
S224ENTF20002824  v7.0.6 (092)      Authorized/Up   -   10.255.1.4      Tue Mar  7 14:32:14 2023    Access_Sales_Switch
S224ENTF22001220  v7.0.6 (092)      Authorized/Up   -   10.255.1.3      Tue Mar  7 14:32:43 2023    Access_TAC_Switch
S108EN5919002951  v7.0.6 (092)      Authorized/Up   -   10.255.1.5      Tue Mar  7 14:35:03 2023
Secondary_Core_Switch
        Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config sync error, 3=L3
        Managed-Switches: 4 (UP: 4 DOWN: 0)
```

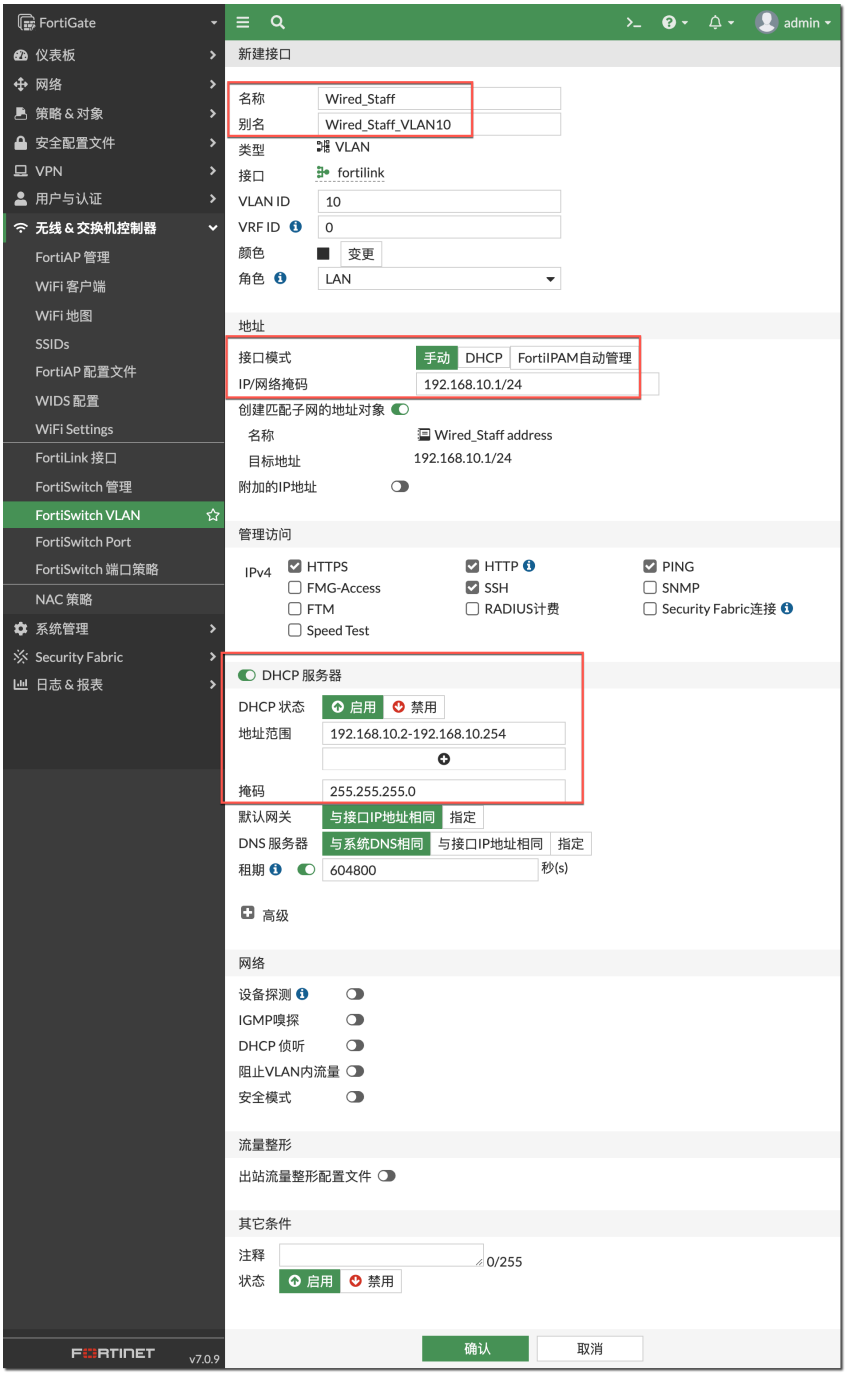
5. 最后不要忘记，在 FortiLink 的聚合接口和 FortiLink 管理配置完毕，记得把这个 FortiLink 聚合接口加入到 HA 的监控接口列表中，这样当主核心交换机挂了（比如断电）之后，业务才可以平稳切换到备防火墙上。

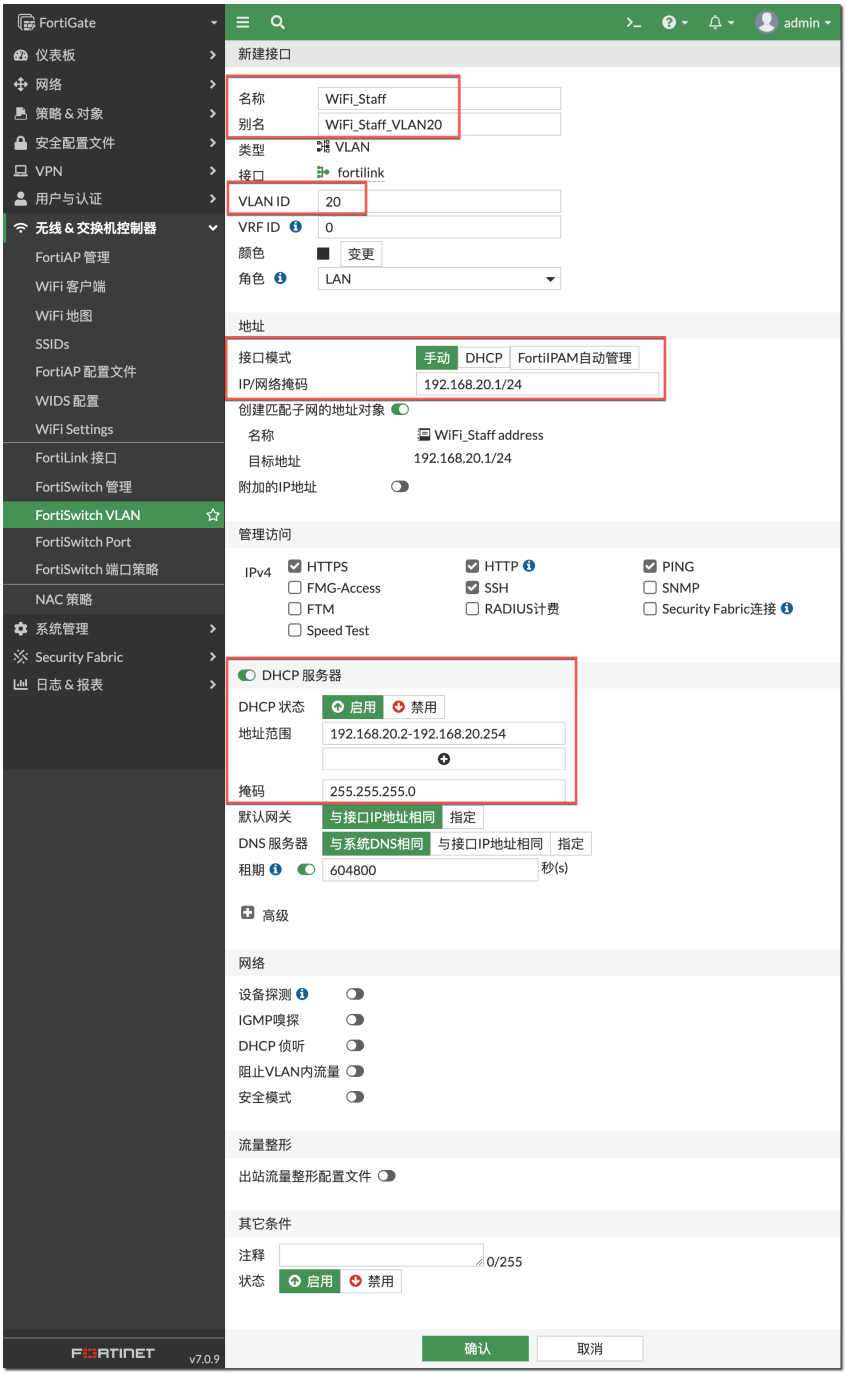


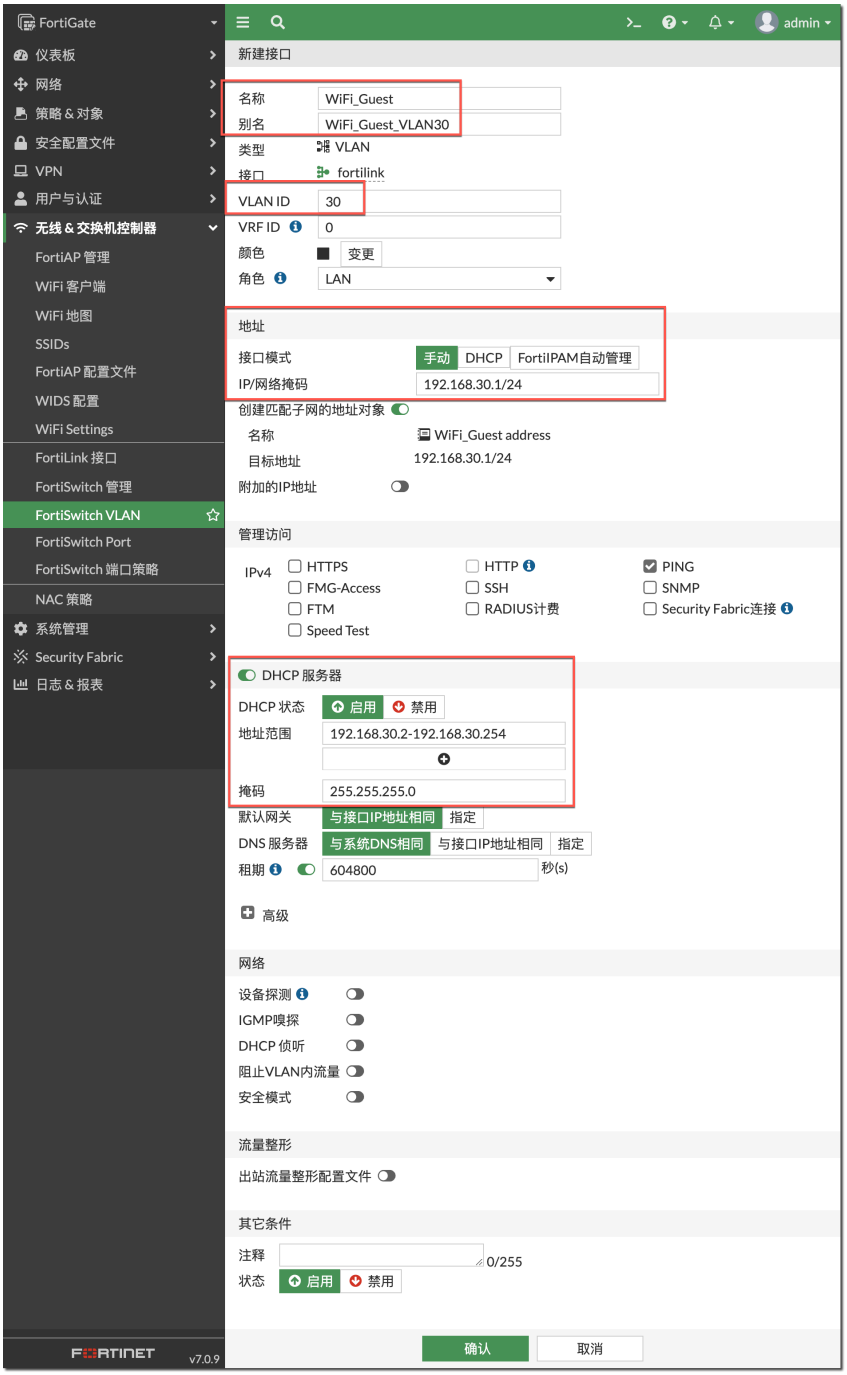
创建 VLAN

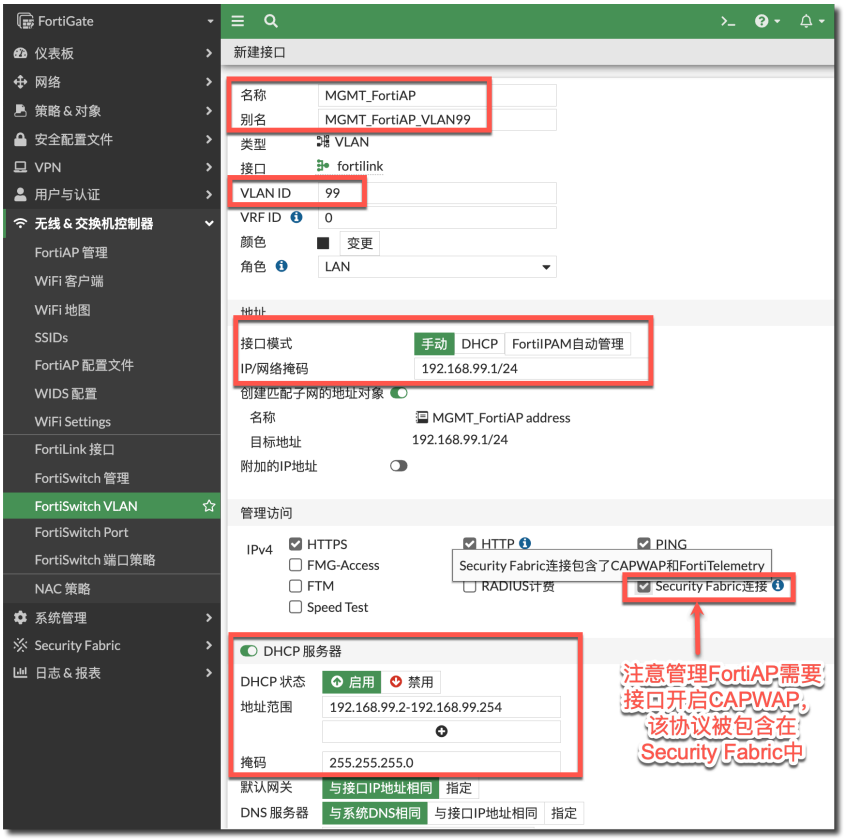
1. FortiGate 通过 FortiLink + Capwap 协议统一的接管 FortiSwitch 和 FortiAP，为有线和无线统一进行管理和业务处理。
2. 进入 FortiGate 的无线 & 交换机控制器 → FortiSwitch VLAN 页面，新建有线网络 VLAN 10、员工无线网络 VLAN 20、访客无线网络 VLAN 30、管理 FortiAP VLAN 99。注意管理 FortiAP 的 VLAN 接口要开启 Security Fabric，包含 CAPWAP。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
+新建
编辑
删除
搜索
名称
VLAN ID
IP
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0

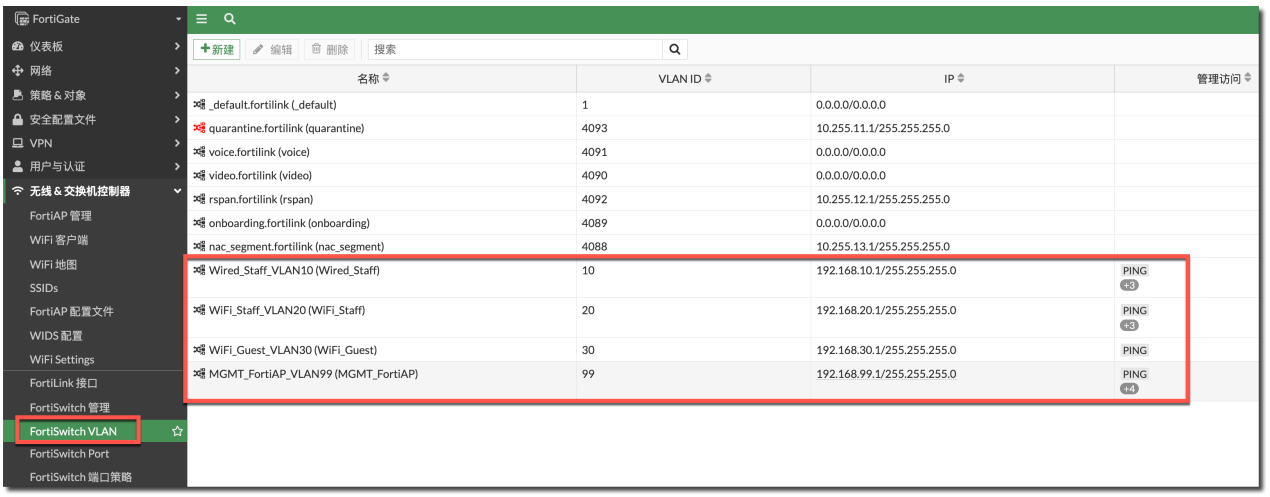

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 Wired_Staff
别名 Wired_Staff_VLAN10
类型 VLAN
接口 fortilink
VLAN ID 10
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.10.1/24
创建匹配子网的地址对象
名称 Wired_Staff address
目标地址 192.168.10.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.10.2-192.168.10.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 WiFi_Staff
别名 WiFi_Staff_VLAN20
类型 VLAN
接口 fortilink
VLAN ID 20
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.20.1/24
创建匹配子网的地址对象
名称 WiFi_Staff address
目标地址 192.168.20.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.20.2-192.168.20.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消

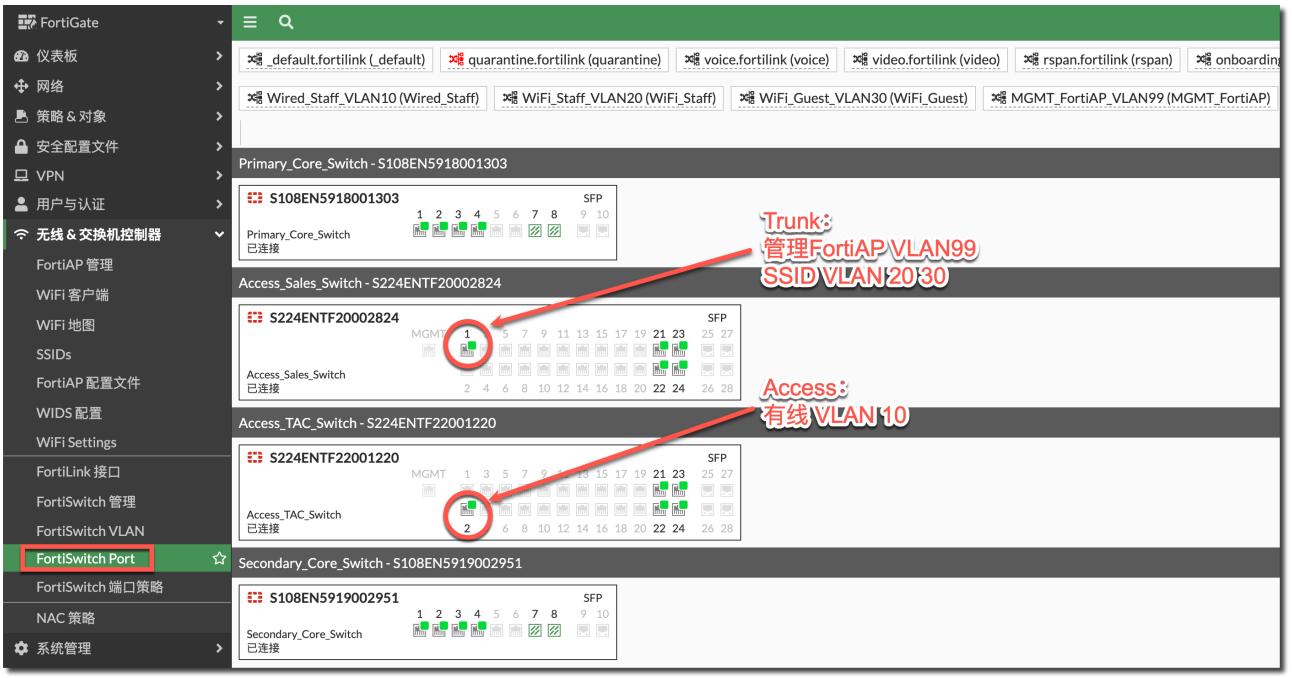

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
新建接口
名称 WiFi_Guest
别名 WiFi_Guest_VLAN30
类型 VLAN
接口 fortilink
VLAN ID 30
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.30.1/24
创建匹配子网的地址对象
名称 WiFi_Guest address
目标地址 192.168.30.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.30.2-192.168.30.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消
FORTINET
v7.0.9


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
新建接口
名称 MGMT_FortiAP
别名 MGMT_FortiAP_VLAN99
类型 VLAN
接口 fortilink
VLAN ID 99
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.99.1/24
创建匹配子网的地址对象
名称 MGMT_FortiAP address
目标地址 192.168.99.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access
FTM
Speed Test
Security Fabric连接包含了CAPWAP和FortiTelemetry
RADIUS计费
Security Fabric连接
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.99.2-192.168.99.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
注意管理FortiAP需要接口开启CAPWAP，该协议被包含在Security Fabric中

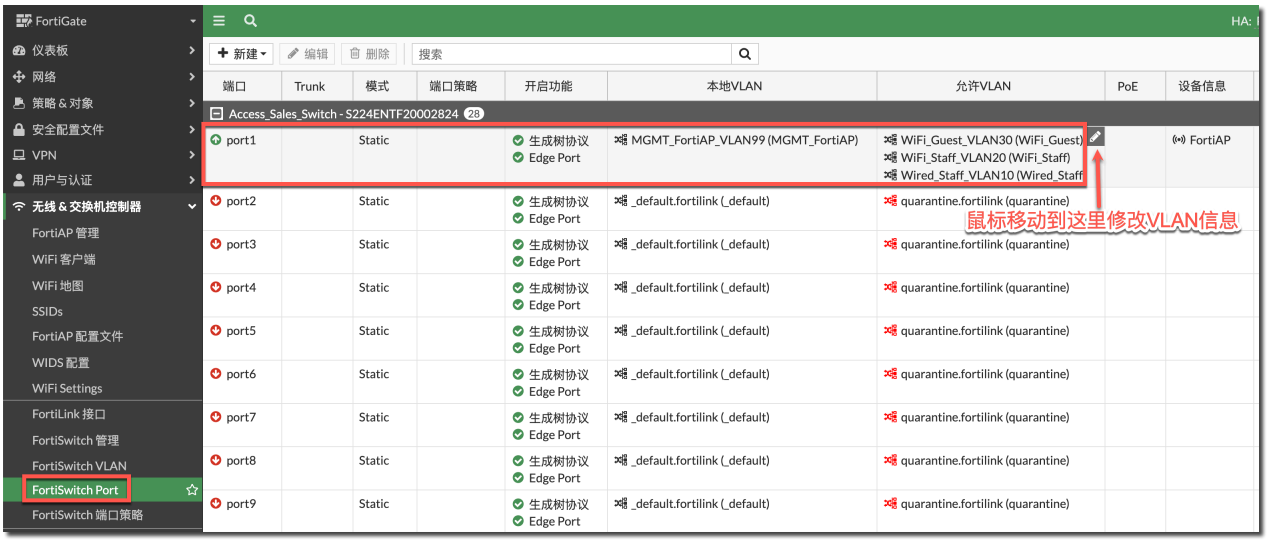
3. VLAN 创建完成。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
新建 编辑 删除 搜索
名称 VLAN ID IP 管理访问
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0
Wired_Staff_VLAN10 (Wired_Staff) 10 192.168.10.1/255.255.255.0 PING
WiFi_Staff_VLAN20 (WiFi_Staff) 20 192.168.20.1/255.255.255.0 PING
WiFi_Guest_VLAN30 (WiFi_Guest) 30 192.168.30.1/255.255.255.0 PING
MGMT_FortiAP_VLAN99 (MGMT_FortiAP) 99 192.168.99.1/255.255.255.0 PING

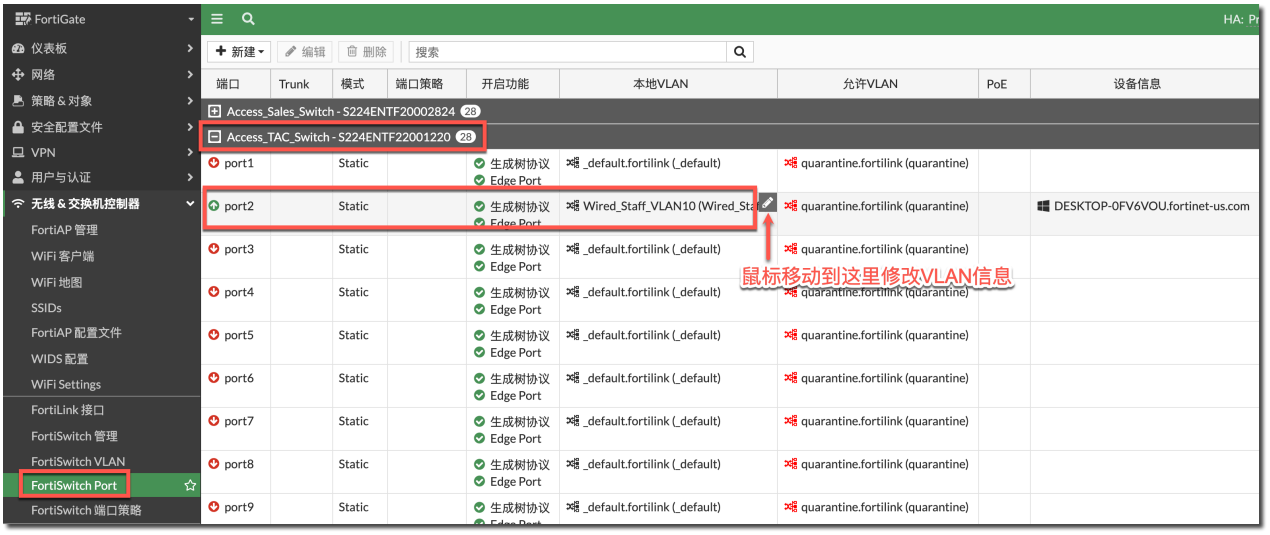
分配 VLAN



1. Sales 接入交换机的 port1 接 FortiAP，配置为 Trunk 模式，允许 VLAN20 和 VLAN30，VLAN20 用于 WIFI - STAFF 的 SSID 用户，VLAN30 用于 WIFI - GUEST 的 SSID 用户。另外本地 VLAN 设置为 VLAN 99，用于 FortiAP 的管理（FortiAP 的管理相当于使用不打 vlan-tag 的 VLAN 99，本地 VLAN 默认是 VLAN 1，因此只需要将本地 VLAN 设置为 VLAN 99 即可）。

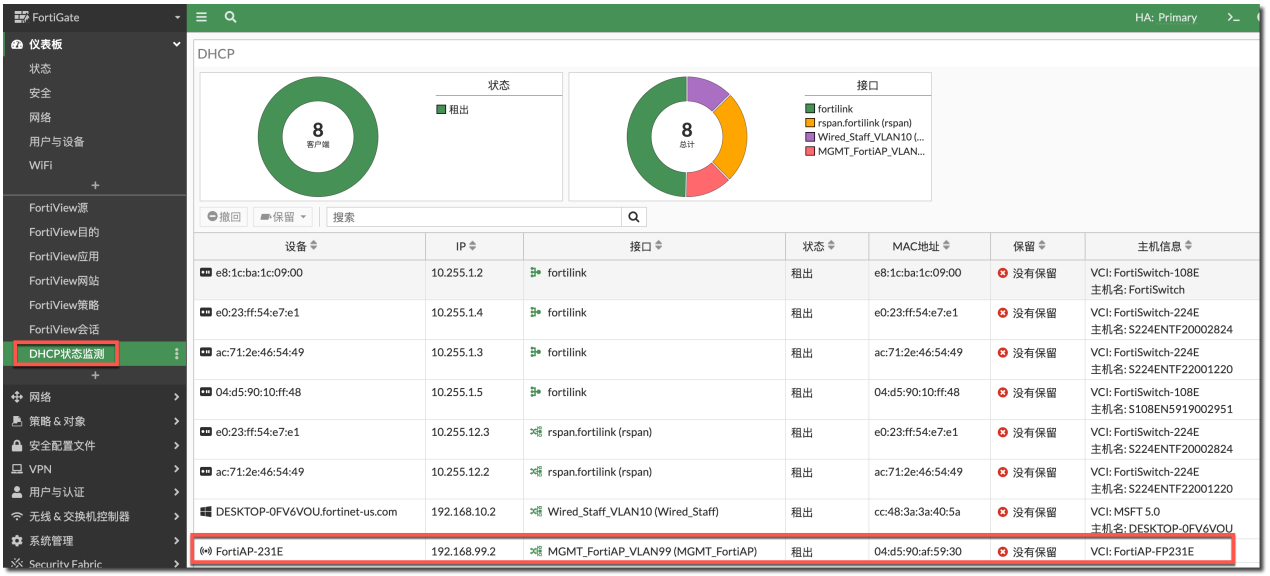


2. TAC 接入交换机的 Port2 接具体的物理 PC，因此只需要将 Port2 设置为 access 口，分配 VLAN 10 即可，用 FortiSwitch 的概念来说就是将本地 VLAN 设置为 VLAN 10，本地 VLAN 不打 vlan-tag，相当于就是设置接口为 access vlan 10。

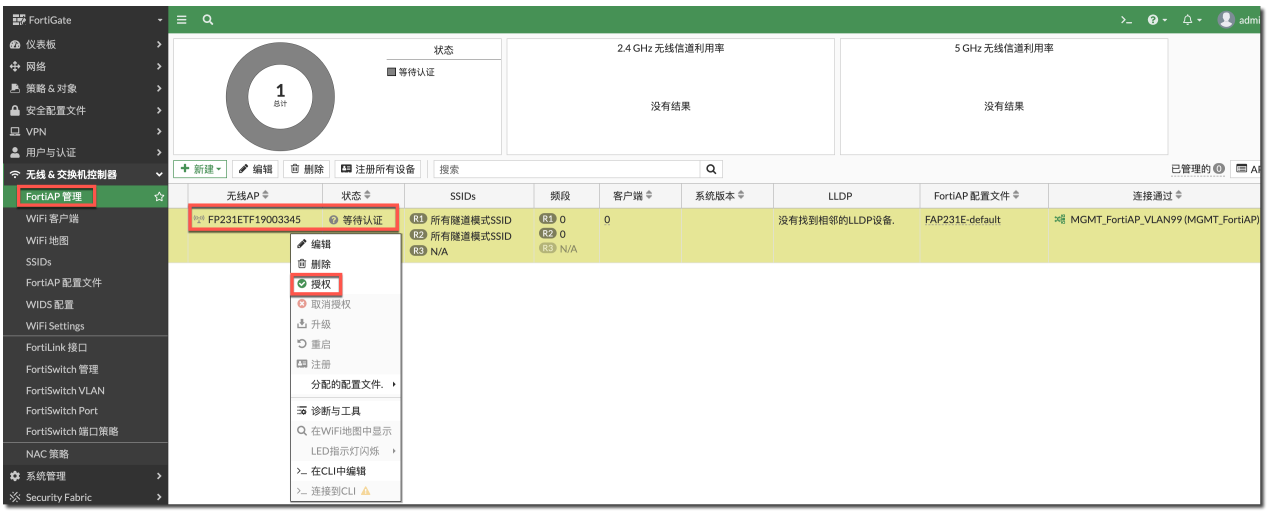


无线配置

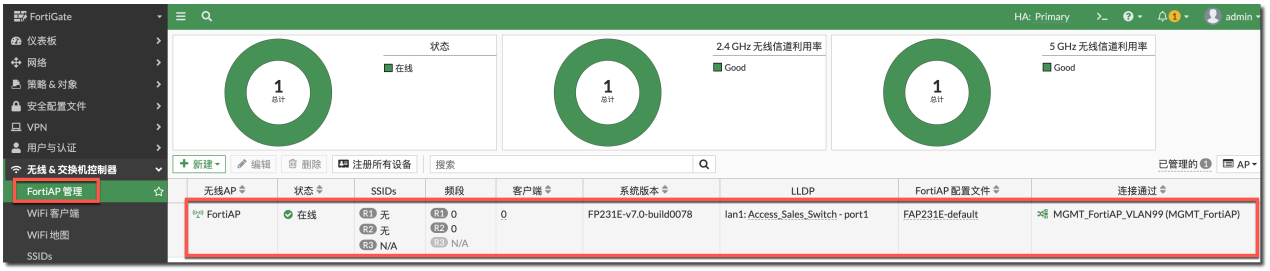
1. FortiAP 连接后，会通过 VLAN 99 的 DHCP 获取到 192.168.99.0/24 网段的 IP 地址。



2. 在之前的配置中，VLAN 99 的接口开启了 CAPWAP 协议（包含在 Security Fabric 中），激活了 FortiGate 的无线管理功能，这时候 FortiAP 会自动找到 FGT 进行注册。因此在 FortiAP 管理的地方可以看到这个 FortiAP 的信息（如果没有看到检查配置，并将 FortiAP 恢复出厂，让 FortiAP 重新获取 IP 地址并发起注册）。



3. 右键点击上图中等待认证的 FortiAP，点击授权按钮，等待几分钟，FortiAP 成功上线。



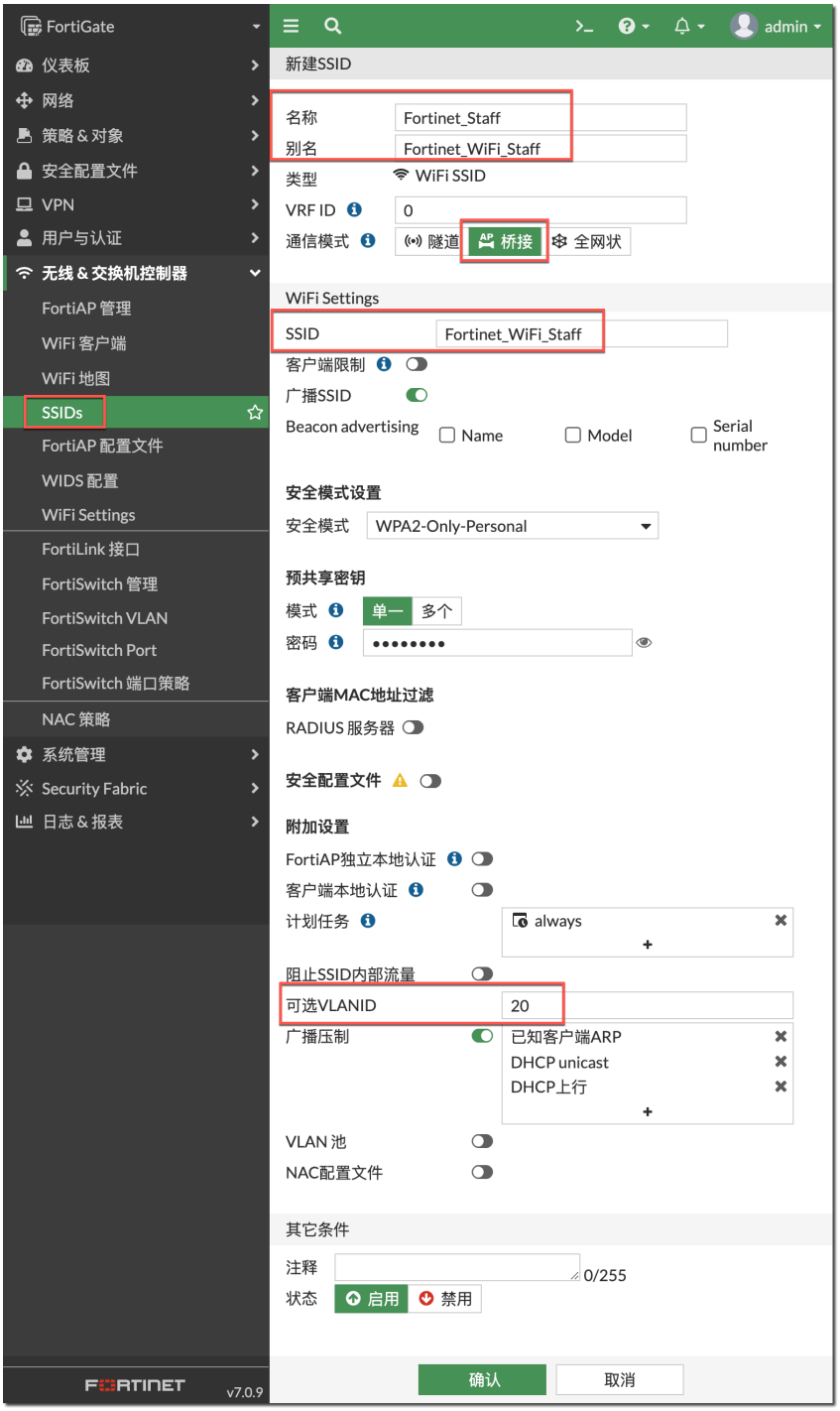
4. 编辑 FortiAP 设置，配置 FortiAP 的名称，或进行升级操作。

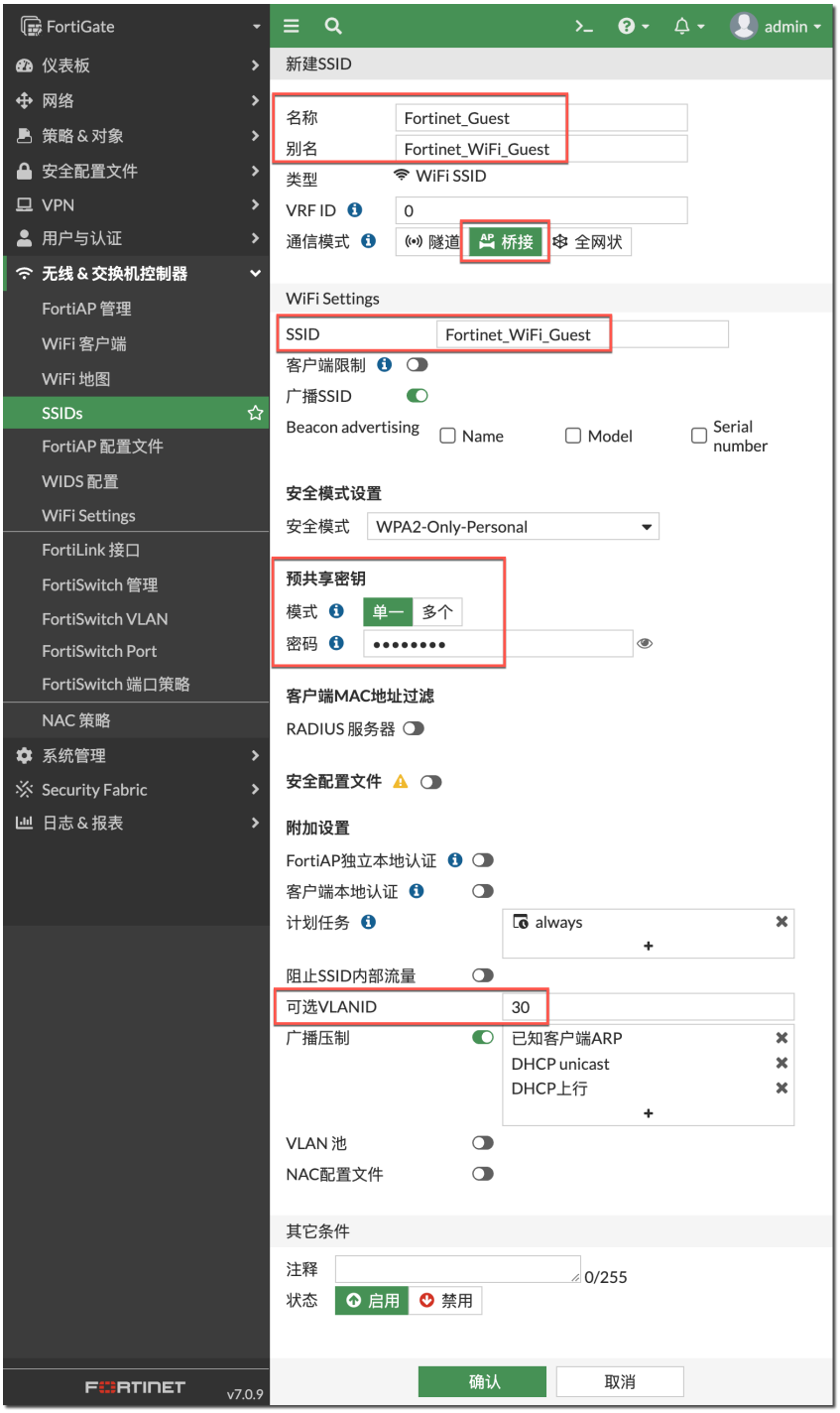


5. 进入无线 & 交换机控制器 → SSIDs 页面，点击新建按钮。



6. 新建 SSID：Fortinet_WiFi_Staff VLAN 20 用于员工 WIFI 使用，Fortinet_WiFi_Guest VLAN 30 用于访客 WiFi 使用。


FortiGate
admin
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
新建SSID
名称
Fortinet_Staff
别名
Fortinet_WiFi_Staff
类型
WiFi SSID
VRF ID
0
通信模式
隧道
桥接
全网状
WiFi Settings
SSID
Fortinet_WiFi_Staff
客户端限制
广播SSID
Beacon advertising
Name
Model
Serial number
安全模式设置
安全模式
WPA2-Only-Personal
预共享密钥
模式
单一
多个
密码
客户端MAC地址过滤
RADIUS 服务器
安全配置文件
附加设置
FortiAP独立本地认证
客户端本地认证
计划任务
always
阻止SSID内部流量
可选VLANID
20
广播压制
已知客户端ARP
DHCP unicast
DHCP上行
VLAN 池
NAC配置文件
其它条件
注释
0/255
状态
启用
禁用
确认
取消

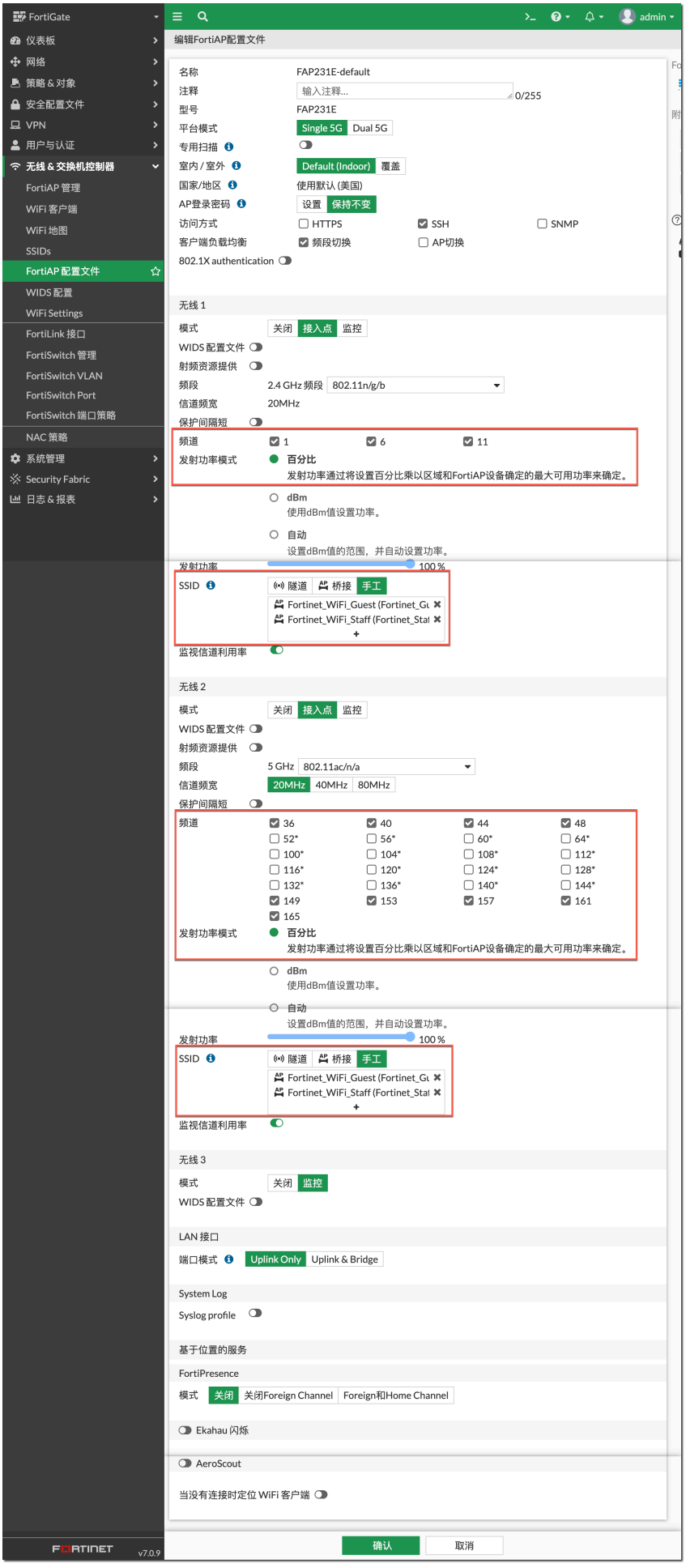




7. 将 SSID 关联到 FortiAP 上，然后让 FortiAP 发布无线 SSID 信号，这样用户就可以通过 SSID 连接到无线了。进入 FortiAP 配置文件页面，编辑 FortiAP 关联的配置文件。



8. 桥模式的 SSID 需要在 FortiAP 配置文件里面手工指定一下，另外下图标示出的配置为推荐配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
编辑FortiAP配置文件
名称 FAP231E-default
注释 输入注释... 0/255
型号 FAP231E
平台模式 Single 5G Dual 5G
专用扫描
室内 / 室外 Default (Indoor) 覆盖
国家/地区 使用默认 (美国)
AP登录密码 设置 保持不变
访问方式 HTTPS SSH SNMP
客户端负载均衡 频段切换 AP切换
802.1X authentication
无线 1
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 2.4 GHz 频段 802.11n/g/b
信道频宽 20MHz
保护间隔短
频道 1 6 11
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 2
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 5 GHz 802.11ac/n/a
信道频宽 20MHz 40MHz 80MHz
保护间隔短
频道 36 40 44 48
52* 56* 60* 64*
100* 104* 108* 112*
116* 120* 124* 128*
132* 136* 140* 144*
149 153 157 161
165
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 3
模式 关闭 监控
WIDS 配置文件
LAN 接口
端口模式 Uplink Only Uplink & Bridge
System Log
Syslog profile
基于位置的服务
FortiPresence
模式 关闭 关闭Foreign Channel Foreign和Home Channel
Ekahau 闪烁
AeroScout
当没有连接时定位 WiFi 客户端
确认
取消
FORTINET
v7.0.9

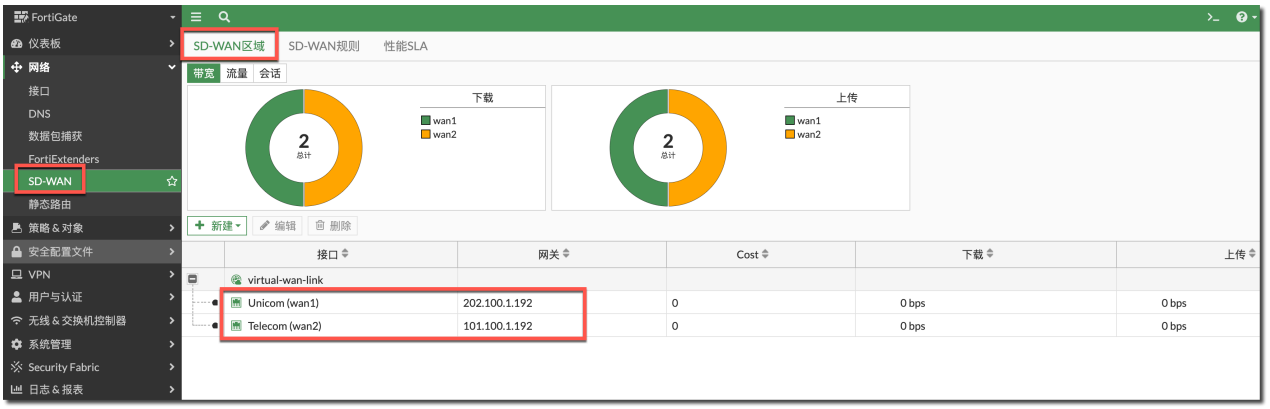


9. FortiAP 会自动关联到这个 FortiAP 配置文件里面的内容，发起 SSID 的信号。



SD-WAN 配置

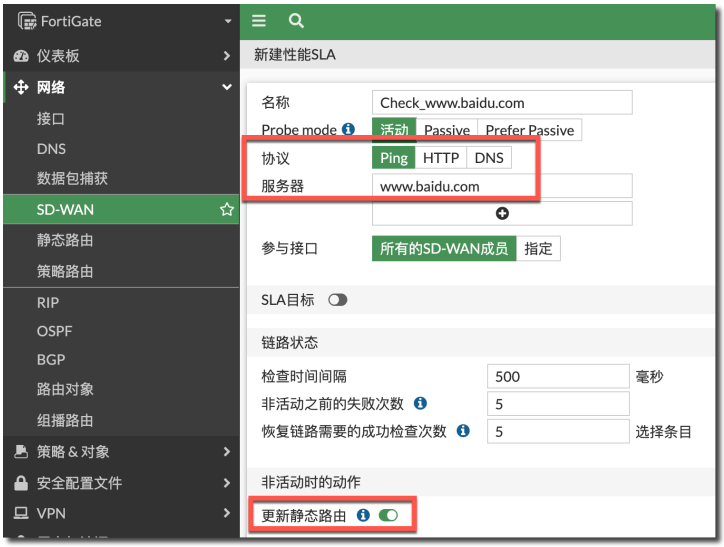
1. 将 wan1 和 wan2 加入 SD-WAN 接口，并配置各自的网关。

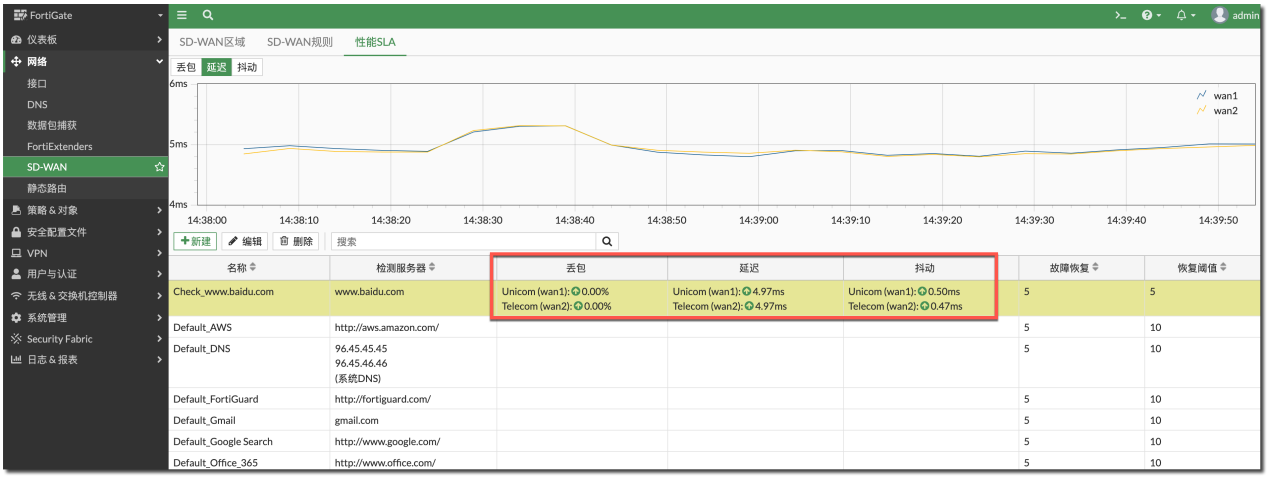


2. 添加 SD-WAN 的默认路由。



3. 配置 SD-WAN 的 SLA，以便 wan1 和 wan2 相互备份和切换。



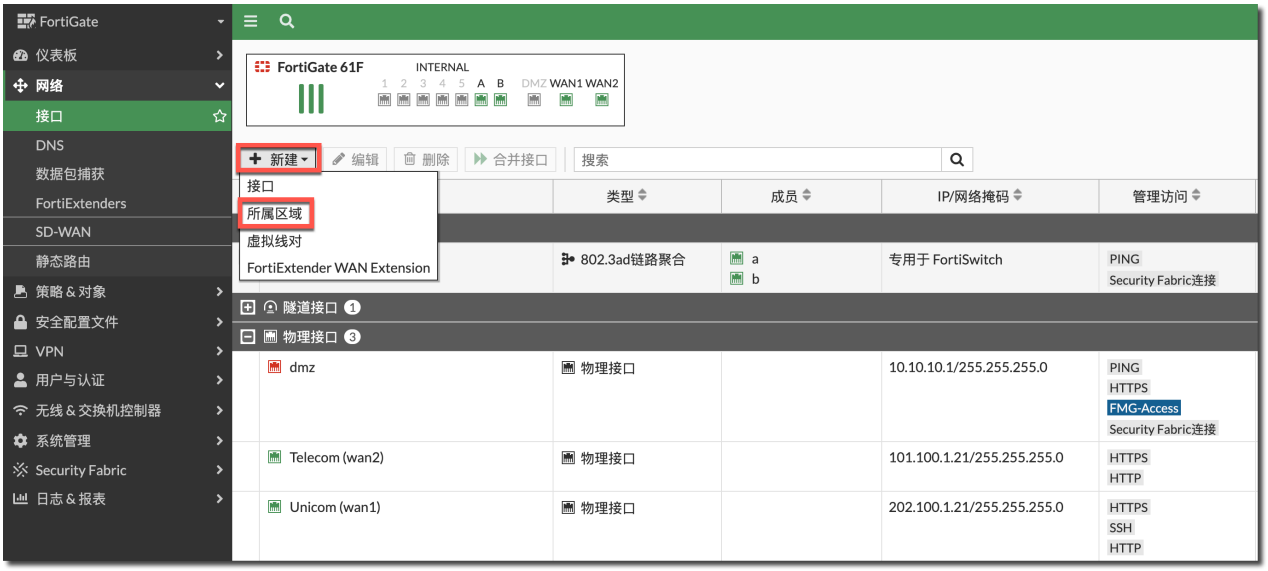


4. SD-WAN 规则可根据需求配置，这里保持默认的按照源 IP 负载均衡。

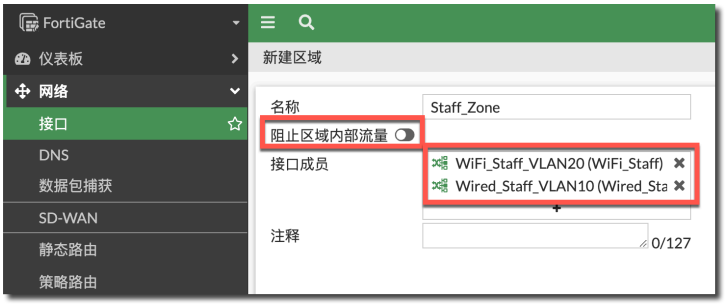


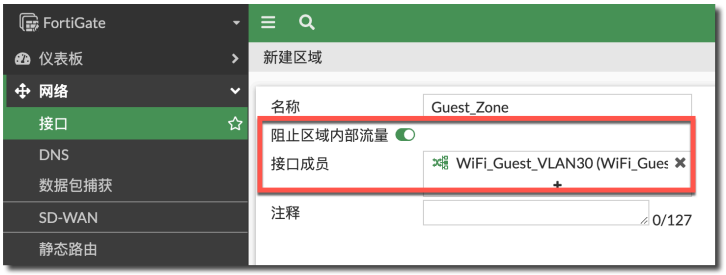
安全策略配置

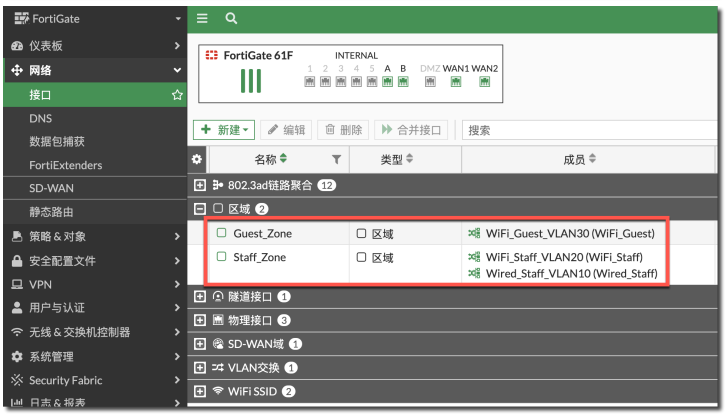
1. 需求与规划：
   - 员工的有线与无线都需要访问互联网
   - 访客的无线仅仅可以访问互联网
   - 员工的有线和无线需要互访
2. 建议将防火墙的接口划分区域，以区域的方式配置策略，有时候可以做到简化配置的目的，甚至未来替换设备的时候，策略的备份和还原也会更加简单一些。规划区域两个区域：
   - 员工网络区域：包括员工的有线 VLAN 10 和员工的无线 VLAN 20 （区域之间的隔离关闭，这样 VLAN 10 和 VLAN 20 之间的数据可通）
   - 访问网络区域：包括访问的无线 VLAN 30
3. 进入网络 → 接口页面，新建所属区域。



4. 分别创建 Staff_Zone（包含 VLAN 10 和 VLAN20）和 Guest_Zone（包含 VLAN30）。







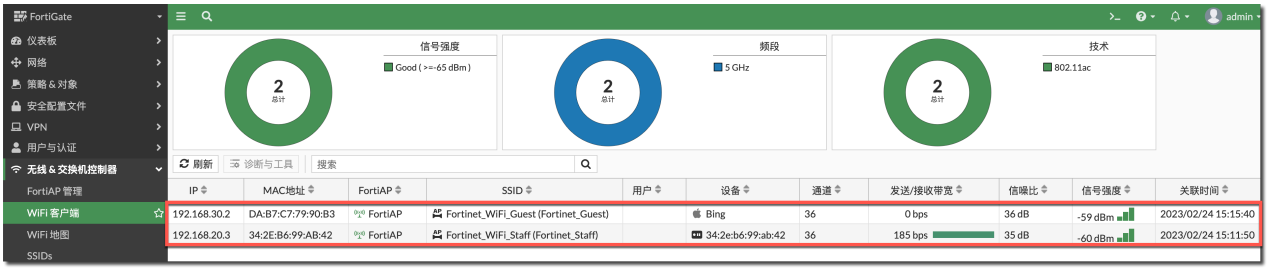
5. 如此只需要配置两条策略，就可以实现我们规划的需求，新建 Staff-Zone 到 SD-WAN 与 Guest-Zone 到 SD-WAN 的安全策略。



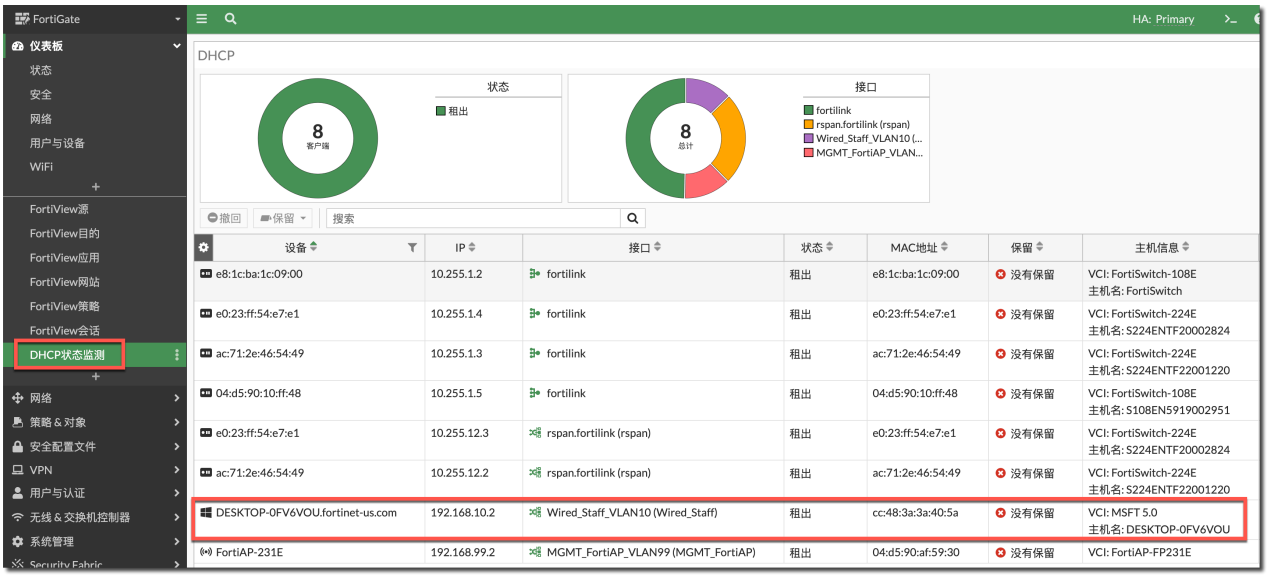
结果验证

客户端访问

1. 进入无线 & 交换机控制器 → WiFi 客户端，查看在线的无线客户端。可以看到 WiFi_Staff 和 WiFi_Guest 各有一个终端在线。



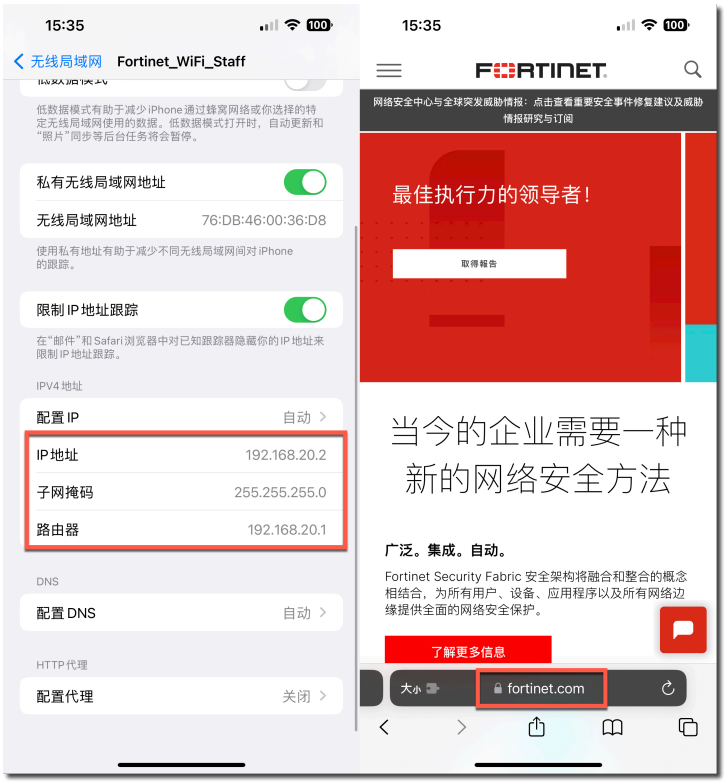
2. 在 DHCP 状态检测中查看所有的客户端信息，可以看到有线网络的 PC 也正常上线。



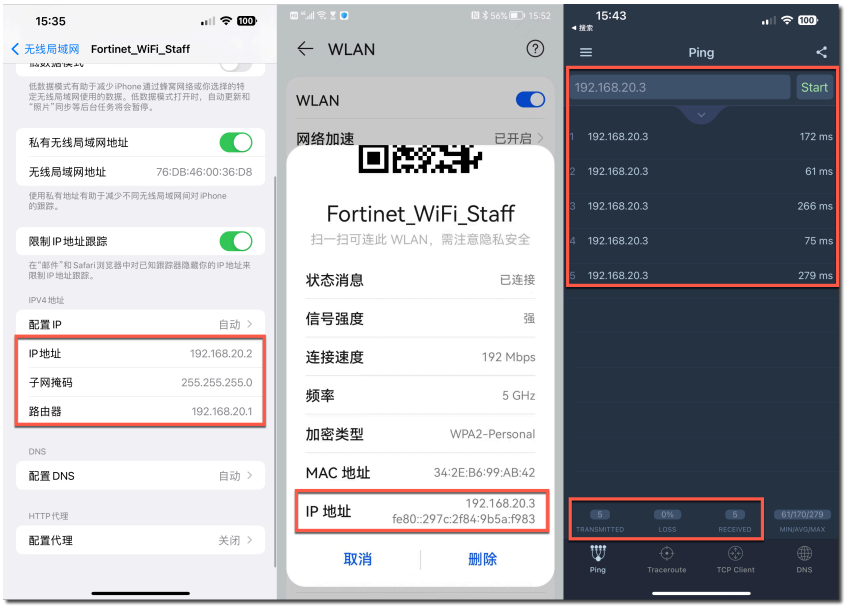
3. 有线网络（VLAN 10）用户访问互联网。



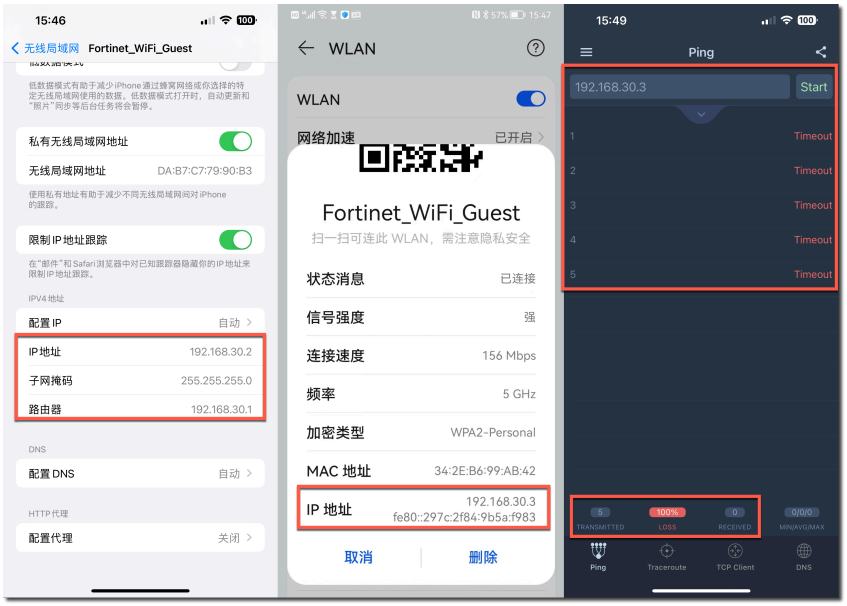
4. 无线网络用户访问互联网。



5. 使用 VLAN 20 下的客户端访问 VLAN 20 下的另一个客户端（Staff 区域）。



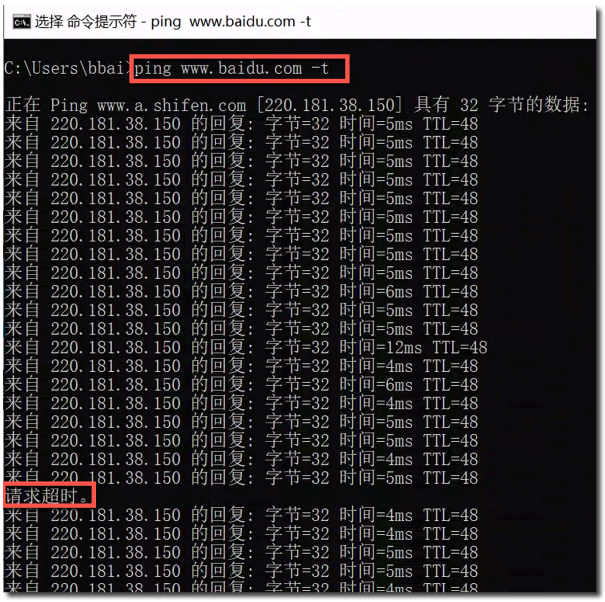
6. 使用 VLAN 30 下的客户端访问 VLAN 30 下的另一个客户端（Guest 区域），可以看到由于该区域配置了禁止相互访问，所以 VLAN 30 下的客户端是无法互相访问的。



FortiLink 切换测试

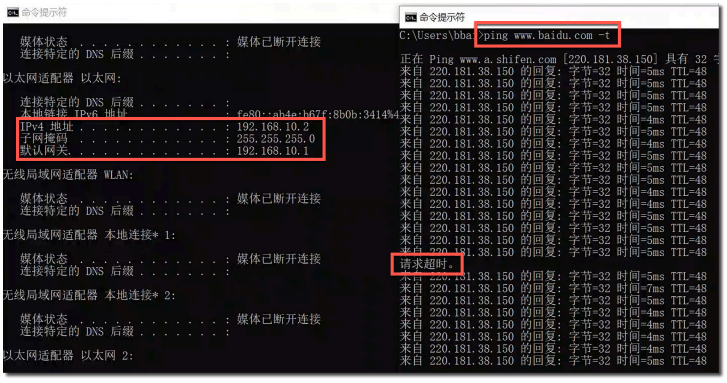
1. 在我们的测试场景下 FortiLink 聚合接口的 a 和 b 是同时 UP 起来的，如果 a 或 b DOWN（不能同时 DOWN），是否有影响业务呢?

2. 经过测试，DOWN a 或 b 接口，客户端丢包 1 个或者不丢包。

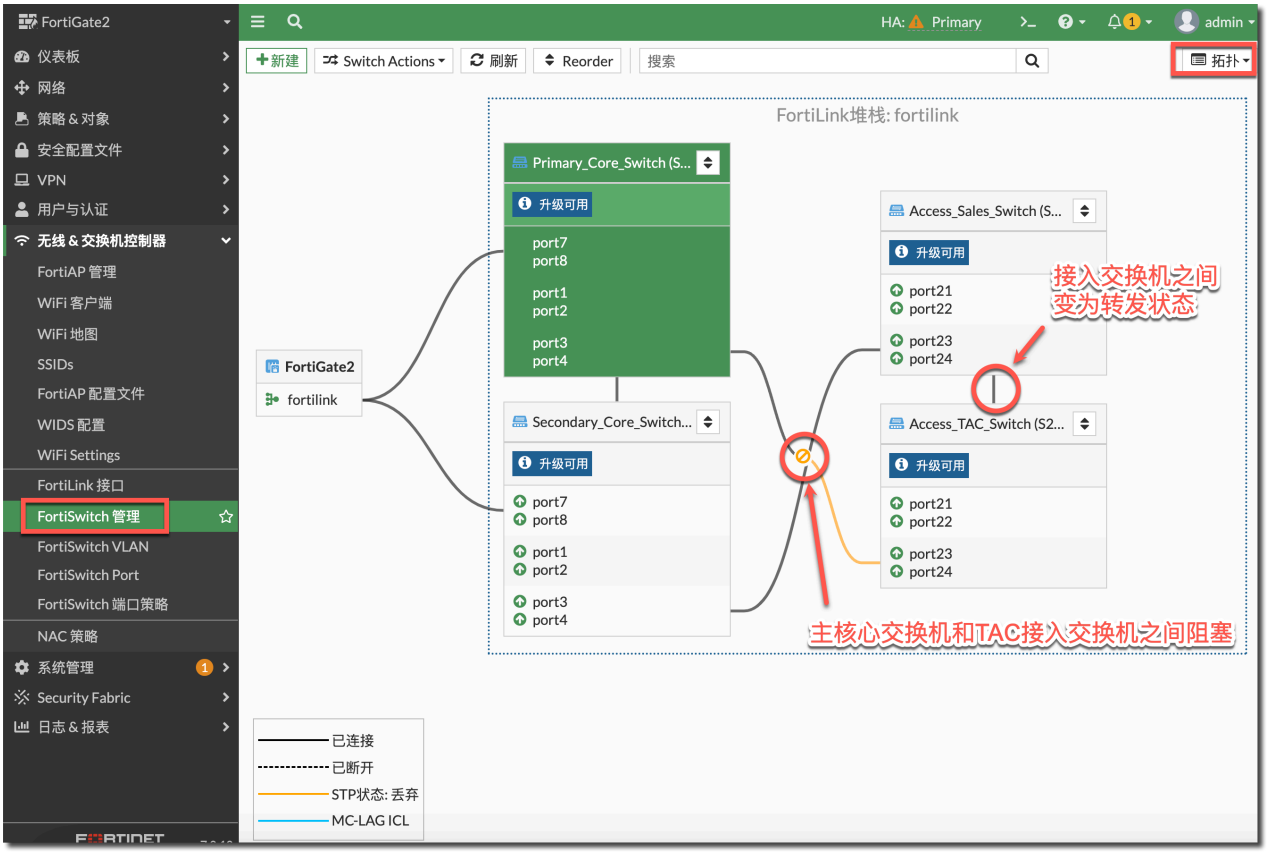


HA 切换测试

1. 重启当前主防火墙查看内网 PC 丢包情况，HA 切换丢包 1 个。



2. 此时查看原备（新主）防火墙的 FortiLink 交换机接管状态，流量全部会切换到备防火墙和备核心交换机上（FortiGate HA 主备切换后，原主 FortiGate 下的核心交换机会在新主 FortiGate 重新上线）。



FortiLink 星型环形混合组网

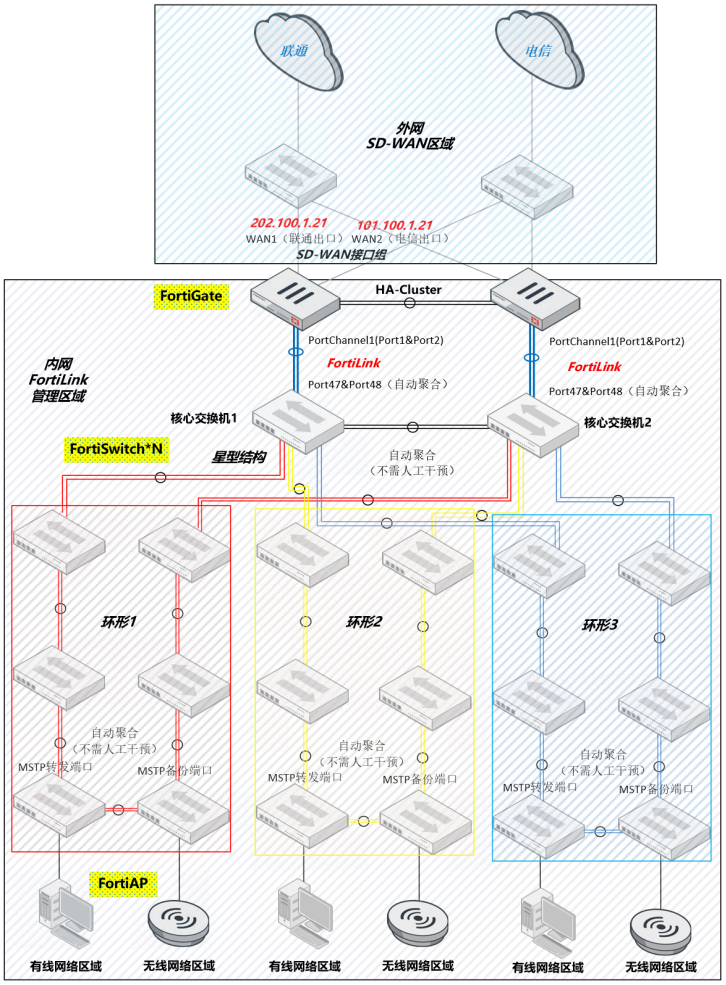
网络需求

虽然环形组网是比较好的方式，唯一的问题是环路太大了，一旦网络中出现接口震荡等情况，MSTP 会出现反复的计算和收敛，一个环内的交换机越多 STP 速度就越慢，通常不推荐超过 16 台的 FortiLink 环。这样有没有好的解决办法呢?

解决方案

1. 实际上需要将星型组网和环形组网结合起来就可以比较好的解决环过大的问题。我们可以取名叫做星型环形混合组网。

2. FortiGate 采用 HA-Cluster 方式部署，核心到汇聚的部分采用星型架构组网，接入的部分采用环形组网，如下图所示的混合组网方案。



3. FGT 配置完 FortiLink，FortiSwitch 和 FortiAP 只需要接物理的线即可，不需要有任何的手工配置 FSW 和 FAP，按照拓扑图接好物理的线之后，等待 FortiLink 整网收敛完毕之后，就可以通过 FortiGate 统一的管理和配置 FSW 和 FAP，整网的所有的有线和无线的配置统统都在 FGT 上完成，通过 FGT 的 FortiLink 做到整网可控、可配、可视化。

4. 混合型组网的举例略过，其实 FortiGate 的配置是一样的，混合组网无非就接线的方式不一样而已。

FortiLink 交换机零配置跨三层部署

从 FortiOS 7.2 + FortiSWOS 7.2 开始，交换机添加了一个新的功能叫 ZTP（Zero-touch Provisioning，也叫 Zero-touch Management），主要有以下变化：

- 交换机默认所有的接口都开启了 FortiLink 管理，之前的版本有的交换机只有后面的端口开启了 FortiLink 管理。
- 交换机被防火墙授权管理后，不会再自动重启一次，所以建议在交换机上线之前 reset 一下交换机。
- 交换机取消了命令“switch-mgmt-mode”，增加了命令“config switch auto-network”。

由于上面的变化，在部署交换机的 FortiLink 跨三层部署时，配置上也会有相应的变化：

- 如果交换机通过 DHCP 获取 IP 地址并且通过 option 138 获得防火墙 FortiLink 接口 IP 地址，那么对于交换机的跨三层 FortiLink 部署也是即插即用，不需要在交换机上添加任何配置，即可上线交换机。
- 如果需要在交换机上手动指定防火墙 FortiLink 接口 IP 地址，那么需要在交换机上增加一些配置。

本文档描述了 FortiLink 跨三层部署时，上面两种情况的配置介绍。请参考：FortiLink 交换机零配置跨三层部署 ↗。

交换机零配置跨二层部署

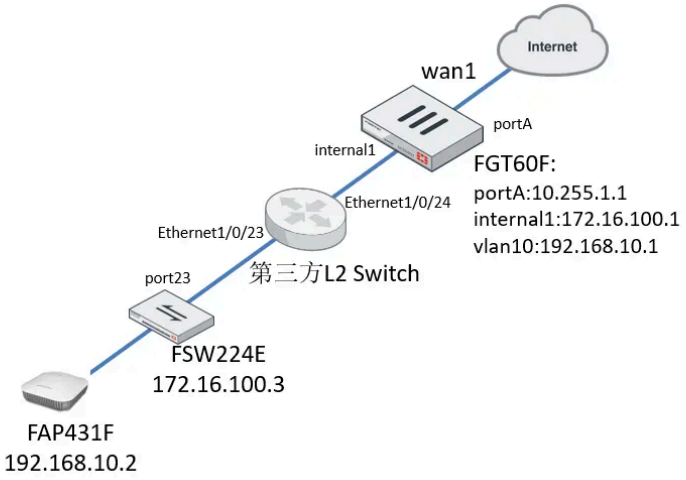
功能介绍

FortiSwitch 的 FortiLink 部署通常有两种方式：

- FortiSwitch 与防火墙直连时的 FortiLink 二层部署方式。
- FortiSwitch 与 FortiGate 之间串接第三方二层交换机时的 FortiLink 三层部署。

本文重点介绍第二种方式。

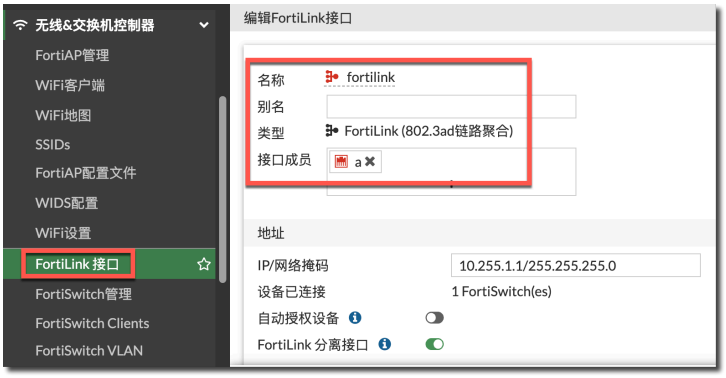
网络拓扑

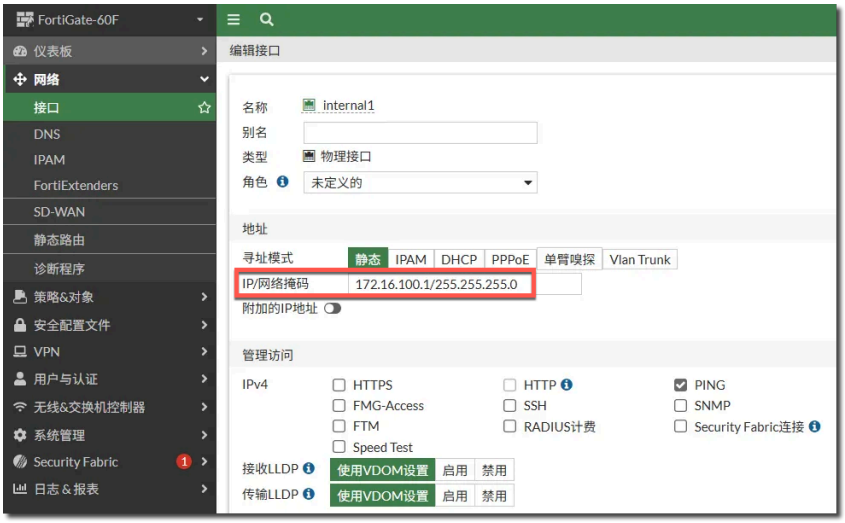


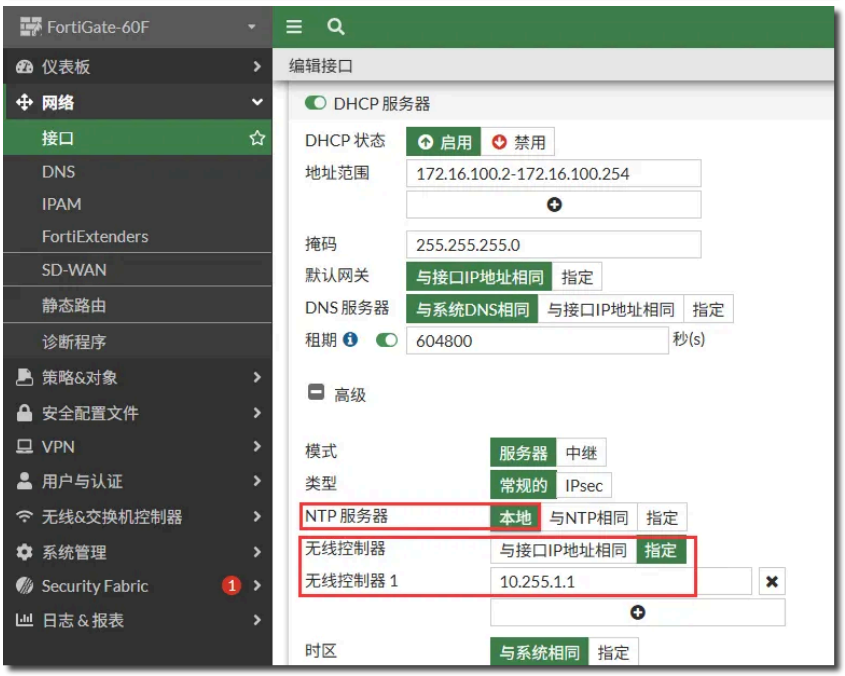
- FortiSwitch 与 FortiGate 之间串接第三方二层交换机，网络中所有设备的网关都在 FortiGate 防火墙上。
- FSW224E 交换机零配置通过 FortiLink 三层部署自动上线。
- FortiGate A 口开启 FortiLink，管理 FortiSwitch，FortiGate A 口不需要接入网线。
- FortiGate internal1 口作为交换机的网关。
- FortiGate internal1 口的 vlan10 子接口作为 FAP 的管理接口。

配置步骤/结果验证

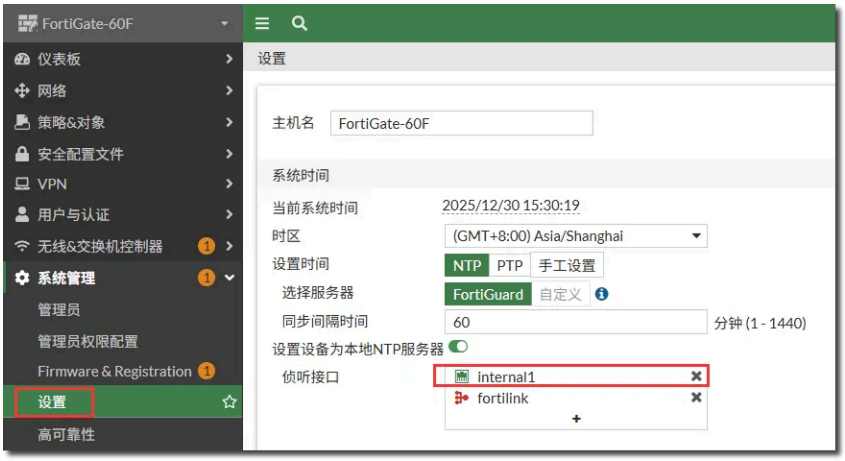
1. 配置防火墙 internal1 接口，internal1 接口作为交换机的网关，通过 DHCP 给交换机分配 IP 地址，以及通过 DHCP Option 138 下发的 FortiLink 接口管理地址。



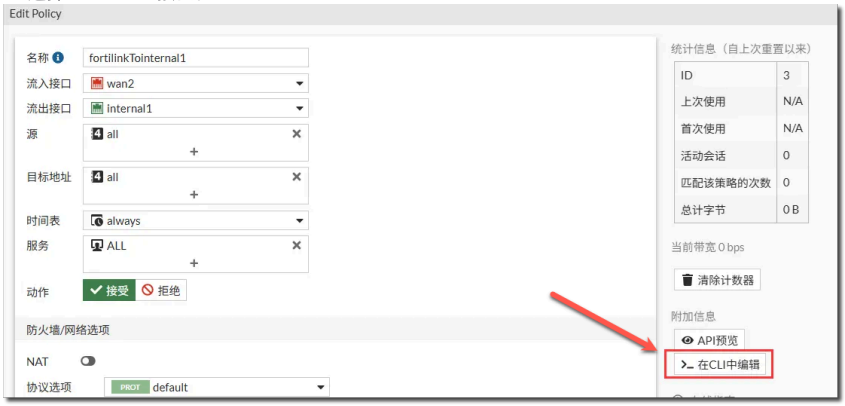




2. 在“系统管理 → 设置”中的 NTP 侦听端口中增加 internal1 作为交换机 NTP Server。



3. 配置 internal1 → FortiLink 接口之间放通的 FortiGate 策略。
   - 默认在 FortiGate 的 GUI 页面是无法选择到 FortiLink 接口的，可以先在 GUI 页面选择一个别的不用的接口，然后在这个 FortiGate 策略的 CLI 下选择 FortiLink 接口。



   - 然后 GUI 页面可以显示出 FortiLink 接口。



   - internal1 和 FortiLink 接口之间不能使用 NAT 方式通讯。

4. 第三方二层交换机配置（以 Cisco 交换机为例）。

```bash
#
vlan 10
#
vlan 4094
#
interface Ethernet1/0/23
 port link-type trunk
 port trunk permit vlan all
 port trunk pvid vlan 4094
#
interface Ethernet1/0/24
 port link-type trunk
 port trunk permit vlan all
 port trunk pvid vlan 4094
#
```

5. 交换机上线。



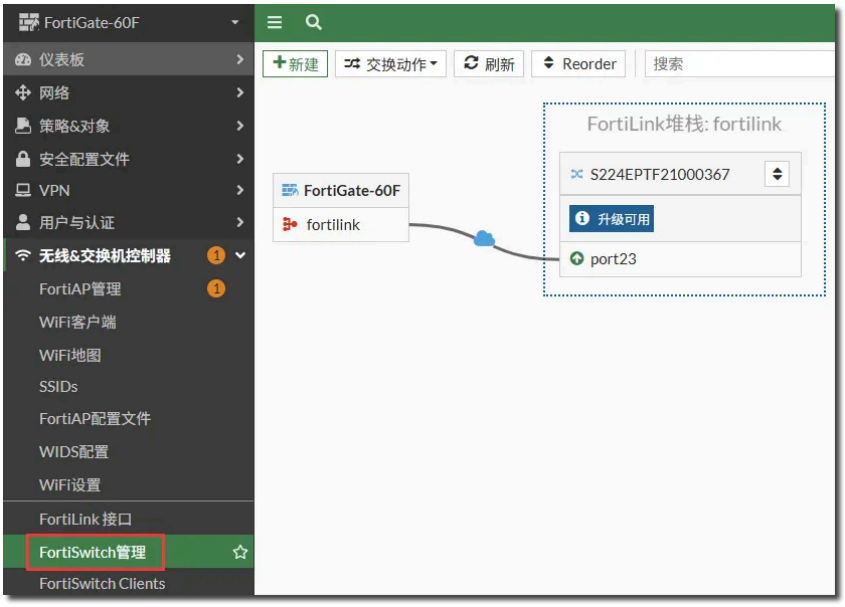
6. 授权交换机。



7. 授权交换机后一段时间，交换机上线。




```bash
FortiLink interface : fortilink
SWITCH-ID          VERSION           STATUS          FLAG   ADDRESS              JOIN-TIME              SERIAL
S224EPTF21000367   v7.4.8 (929)      Authorized/Up   3      172.16.100.3     Tue Dec 30 15:32:00 2025     S224EPTF21000367

        Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config sync error, 2=L2, 3=L3,
V=VXLAN, T=tunnel, X=External
        Managed-Switches: 1 (UP: 1 DOWN: 0 MAX: 24)
```



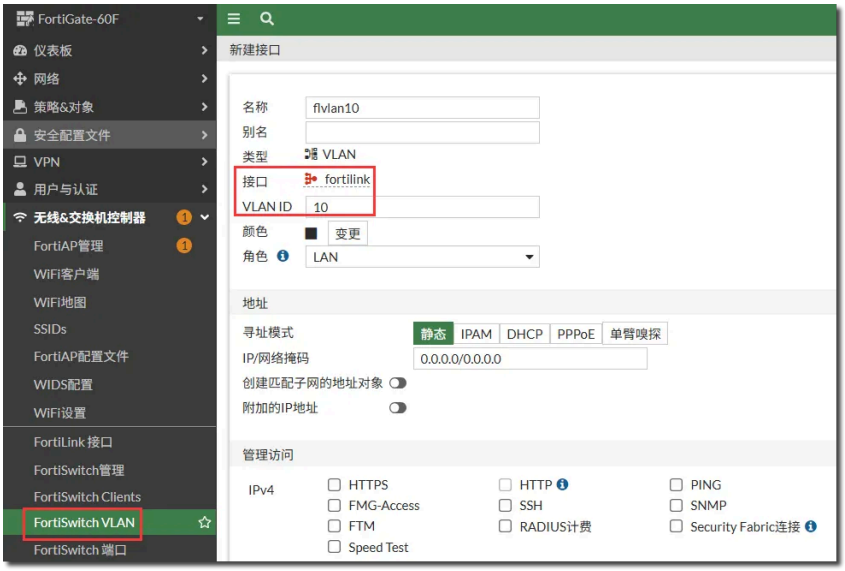
8. 交换机状态查询。

```bash
S224EPTF21000367 # execute switch-controller get-conn-status

Get managed-switch S224EPTF21000367 connection status:
Connection: Connected
Image Version: FGT60F-v7.4-build2829
Remote Address: 10.255.1.1
Join Time: Tue Dec 30 15:32:00 2025
DTLS Version: DTLSv1.2
```

9. 交换机上会自动建立下面的 Foritlink trunk：

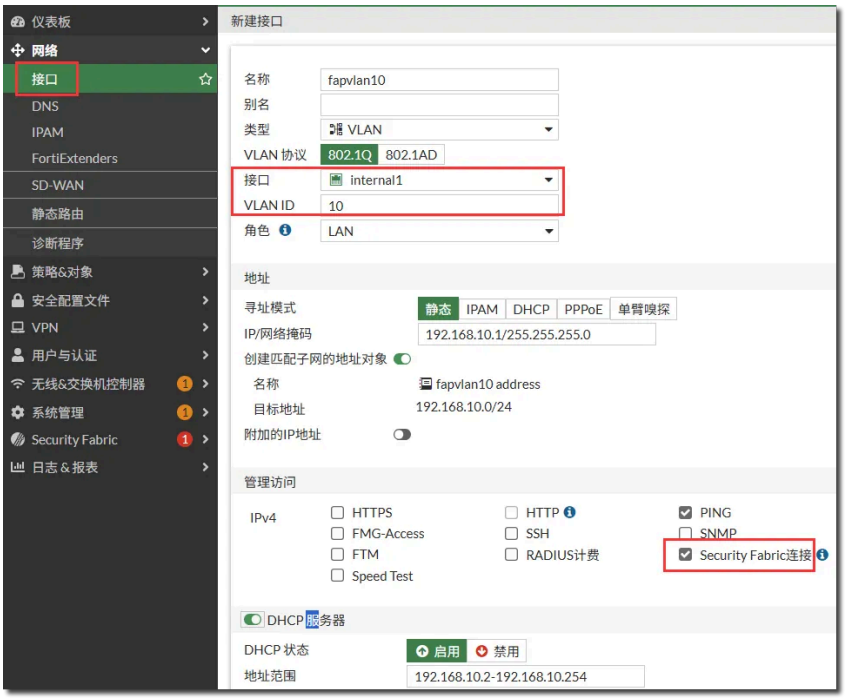
```bash
S224EPTF21000367 # show switch trunk
config switch trunk
    edit "_FLinkDhcpDisc_"
        set auto-isl 1
        set static-isl enable
        set members "port23"
    next
end
```

.0. 配置 FAP 的管理 VLAN。

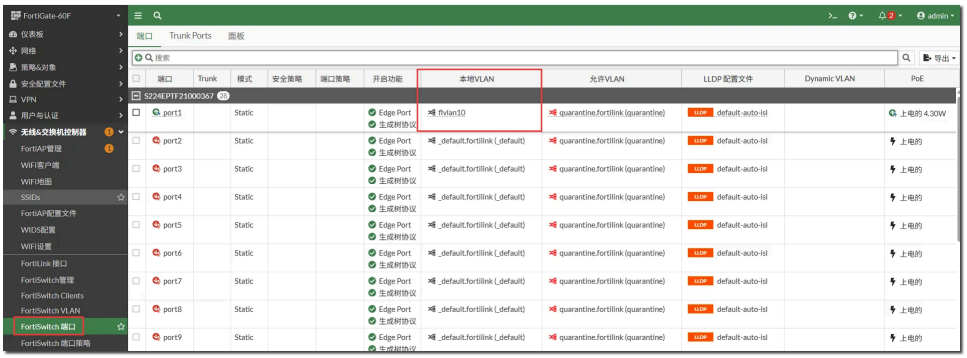


- 需要同时在 FortiLink 接口和 internal1 接口下配置 VLAN 子接口。

- FortiLink 接口下的 vlan10 子接口只是用来做 FortiLink VLAN 下发给交换机用的，所以不需要配置 IP 地址。internal1 下 vlan10 接口的配置：



1. 配置交换机 port1 口的 AP 管理 VLAN。



2. AP 上线，AP 接入网络中后会从 internal1 的 vlan10 接口获取到 IP 地址，然后 FortiGate 上需要授权 AP。



3. 授权 AP 后，AP 上线。



FortiLink 堆叠组网和举例

- FortiLink 部署之 MCLAG 堆叠组网
- FortiLink MCLAG - 堆叠组网 1 Tier 配置举例
- FortiLink MCLAG - 堆叠组网 2 Tier 配置举例
- FortiLink MCLAG - 与第三方设备进行跨堆叠交换机聚合

FortiLink 部署之 MCLAG 堆叠组网

方案说明

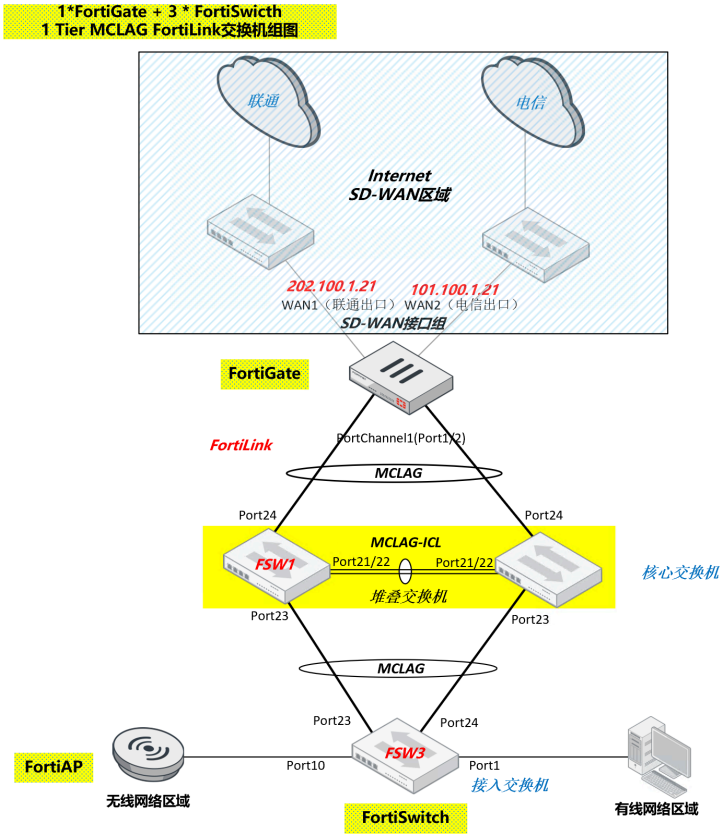
堆叠组网（MCLAG）是最理想的 FortiLink 部署方法了，目前支持 1 层堆叠（只堆叠核心交换机）和 2 层堆叠（堆叠核心交换和接入/汇聚交换机）。

合理的 MCLAG（堆叠）组网规划可以规避掉 STP 阻塞接口（避免环路），整网不需要 STP 了，并且所有的接口都是更好的利用起来，增加使用带宽的同时且无环路，因此 MCLAG 的部署在 FortiLink 部署中是更加优质的选择。

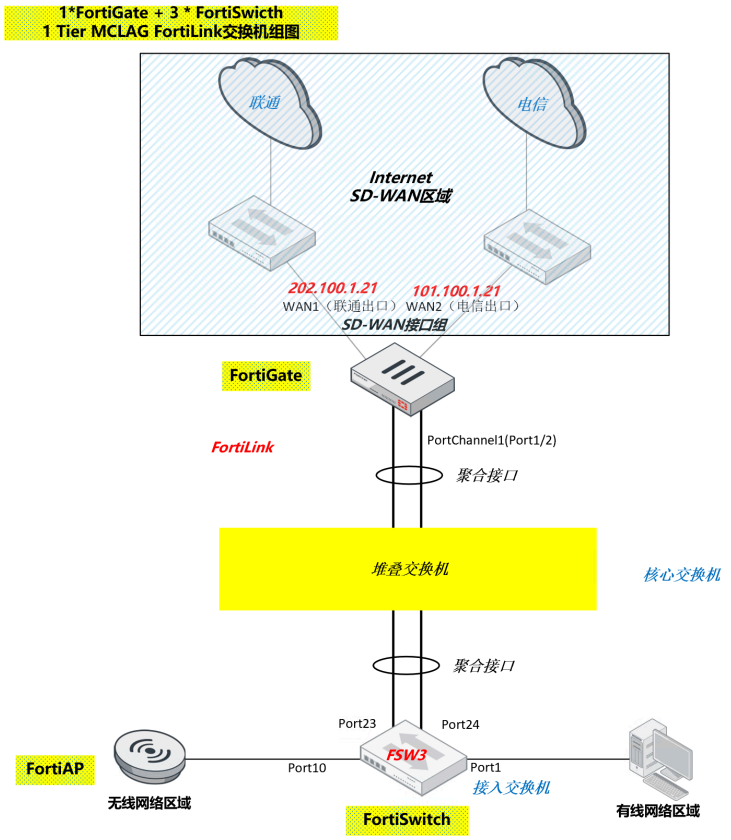
1 Tier MCLAG FortiLink 组网方案

单机 FortiGate 部署

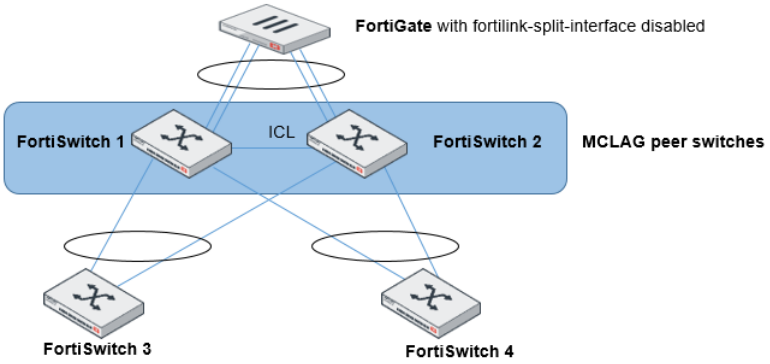
1. 最简单的一个 MCLAG 部署拓扑，FortiGate * 1 + FortiSwitch * 3。



堆叠后的逻辑拓扑：

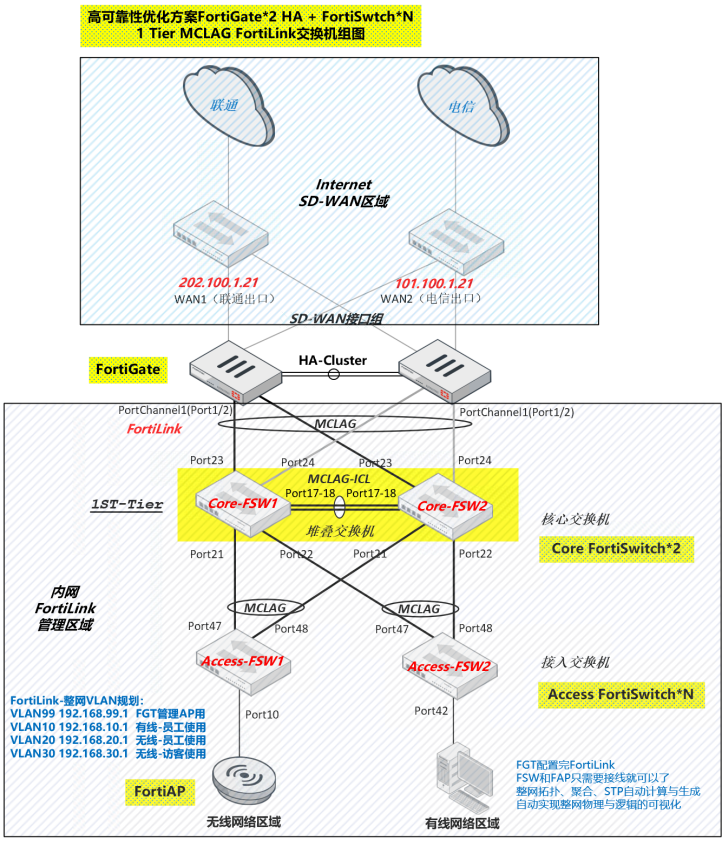


2. 类似的组网。

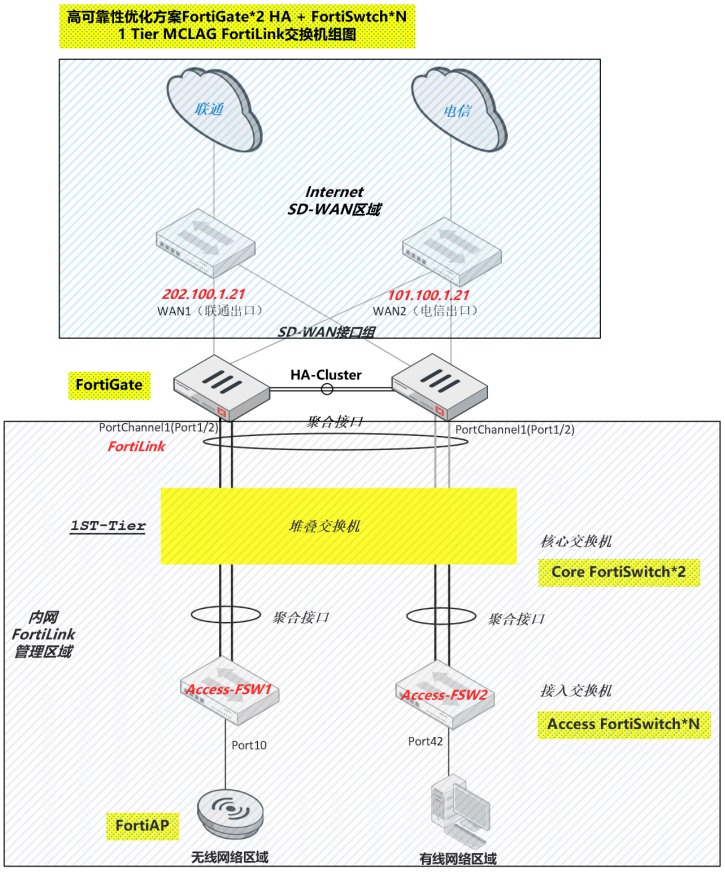


双机 FortiGate HA 部署

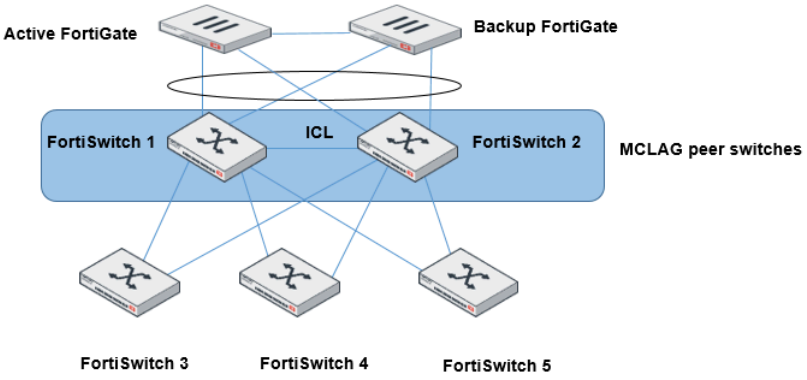
1. FortiGate 采用 HA-Cluster 方式部署，核心到汇聚的部分采用星型架构组网，交换机接入的部分采用 MCLAG 堆叠组网，如下图所示的 1 Tier 交换机堆叠组网方案。



堆叠后的逻辑拓扑：

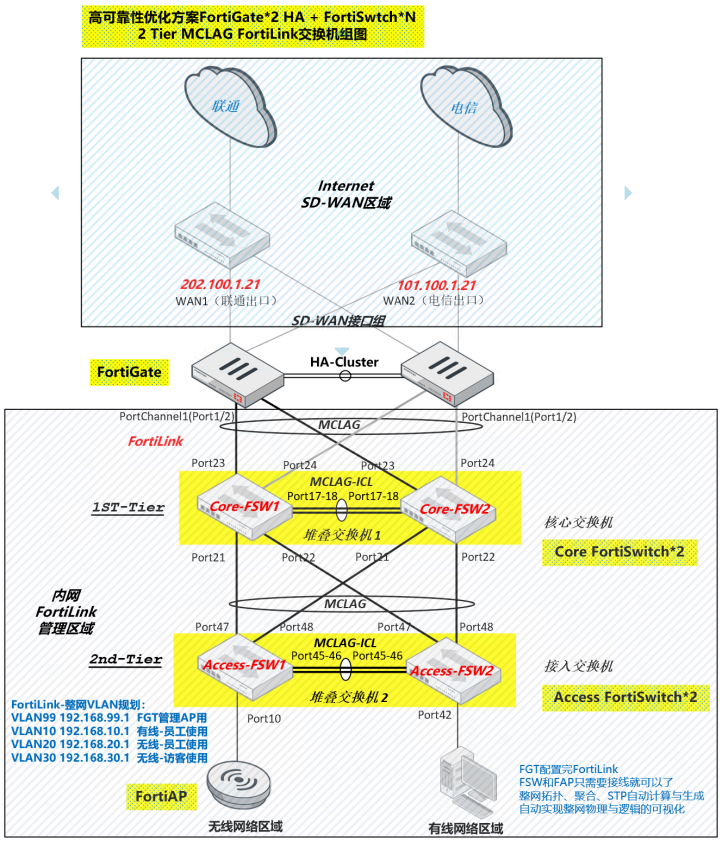


2. 类似的组网。



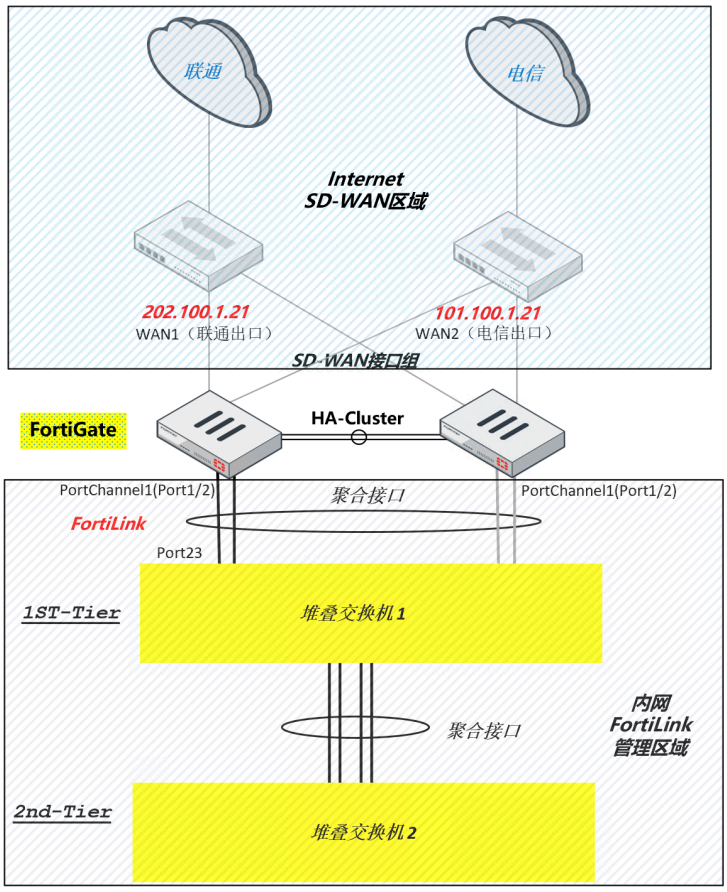
2 Tier MCLAG FortiLink 组网方案

1. 双机 FortiGate HA 部署。

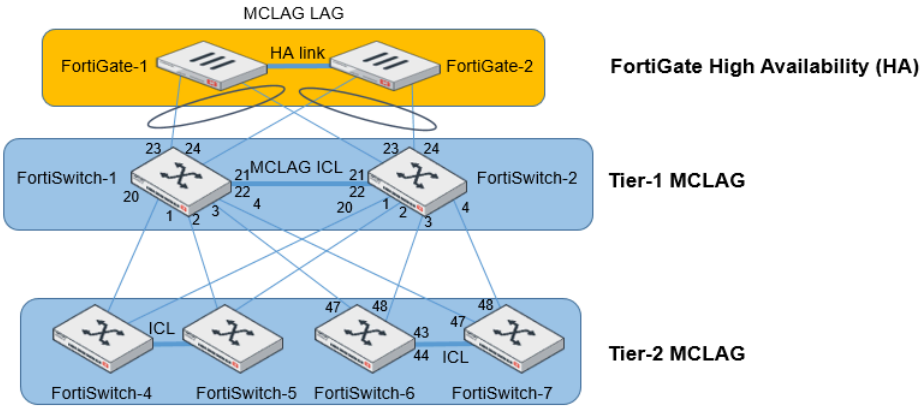


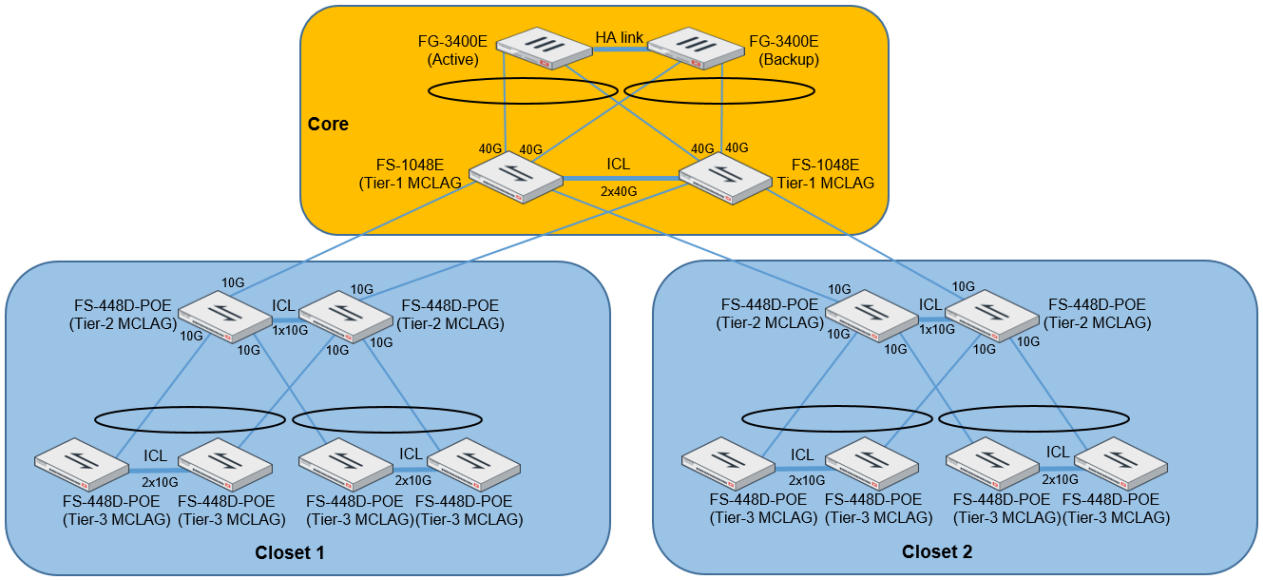
堆叠后的逻辑拓扑：

高可靠性优化方案FortiGate*2 HA + FortiSwtch*N
2 Tier MCLAG FortiLink交换机组图



2. 类似的拓扑。





注意事项

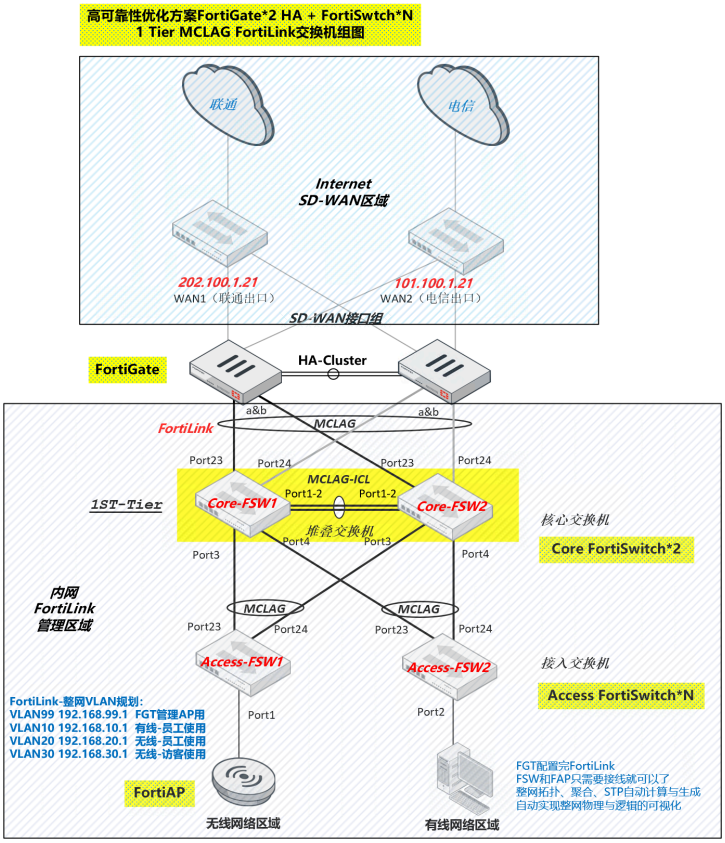
- 最多只能是两个交换机组成一对 MCLAG。
- 组成 MCLAG 的两个交换机必须是相同的硬件型号以及相同的软件版本。
- FortiGate 的聚合端口在 MCLAG Switch 上线前为 split enable 状态，MCLAG Switch 上线并配置为 MCLAG 后，手动修改 FortiGate 的聚合端口为 split disable 状态。
- FortiSwitch 必须是 3.6.4 之后的版本才支持 MCLAG。
- FortiOS 6.2 及以后的版本, 可以配置 FortiGate 聚合端口为 Active 模式。
- 某些设备交换机不支持 MCLAG, 比如 FS - 1XX 系列, 具体请参考 Release Notes。
- 1 Tier MCLAG 只支持核心的两台防火墙进行堆叠，往下的接入交换机不再支持堆叠，整个 FortiLlink 网络中只有一组堆叠点，通常用于 16~32 台以内的 FSW 组网，更多的交换机可以考虑 2 Tier MCLAG FortiLink 组网。

FortiLink MCLAG - 堆叠组网 1 Tier 配置举例

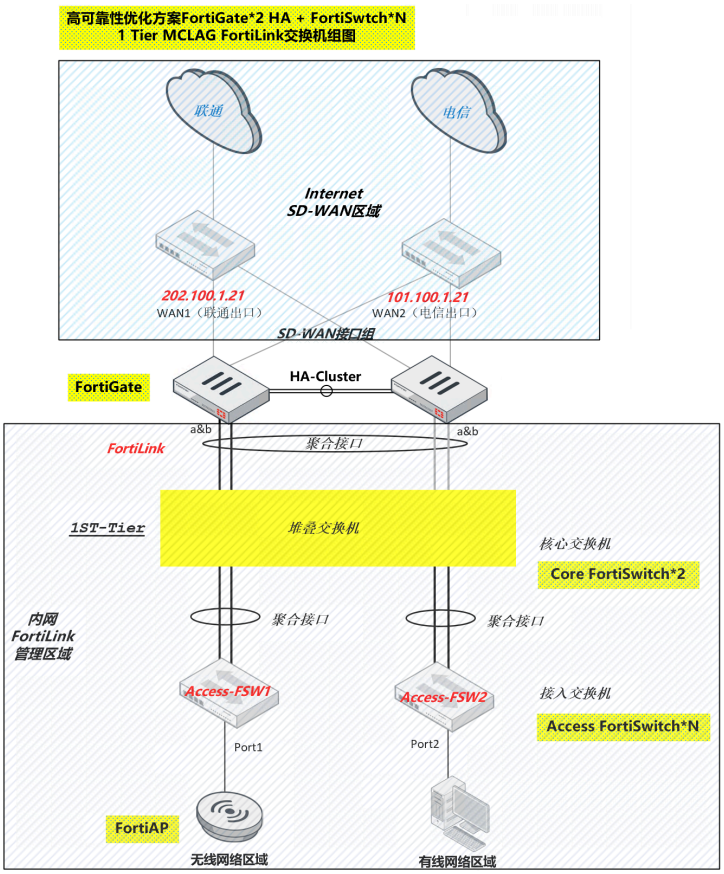
网络需求

FortiGate 运行 HA-Cluster，核心交换机采用 MCLAG 堆叠组网结构。

网络拓扑



形成堆叠之后的逻辑拓扑：



1. 两台 FortiGate 配置为 HA 主备模式。
2. 整体规划：
   - 外网 IP 网段规划：
     - WAN1 联通出口：IP 202.100.1.21，GW 202.100.1.192
     - WAN2 电信出口：IP 101.100.1.21，GW 101.100.1.192
   - WAN1 和 WAN2 合并成 SD-WAN 接口。
   - 内网 IP 网段规划，FGT 通过 FortiLink + Capwap 协议统一的接管 FortiSW 和 FortiAP，为有线和无线统一进行管理和业务处理：
     - 有线网络规划：VLAN 10，192.168.10.1
     - 无线网络规划：员工 WIFI VLAN 20，192.168.20.1
     - 无线网络规划：访客 WIFI VLAN 30，192.168.30.1
     - 管理 FortiAP：VLAN 99，192.168.99.1
   - FortiLink 互联部分规划：
     - FortiGate 的 a 和 b 做聚合接口与 Fortiswitch 的后两个接口互联即可实现聚合接口的 FortiLink 管理。
     - 加入 FortiLink 的接口自动变为聚合口成员。

配置要点

1. FortiGate 配置聚合端口，并设置 split 为 enable 模式。
2. 上线 Core-FSW1 和 Core-FSW2，将 FortiGate 和 Core-FSW1 以及 Core-FSW2 连接起来。
3. 分别登录到 Core-FSW2 和 Core-FSW1 上，配置二者直连的 ISL 接口为“mclag-icl enable”。配置为 ICL 堆叠线路（Inter-Chassis-Link），形成两台核心交换机的堆叠。
4. 配置 FortiGate 聚合端口为 split disable 模式。
5. 上线 Access-FSW1 和 Access-FSW2。
6. 上线 FortiAP 和有线 PC，并测试无线网络和有线网络的连通性。
7. 注意：如果 Switch 上线时，有的 Switch 是旧的版本，最好先升级 Switch 的版本后，再组建 mclag 或是让交换机成环。

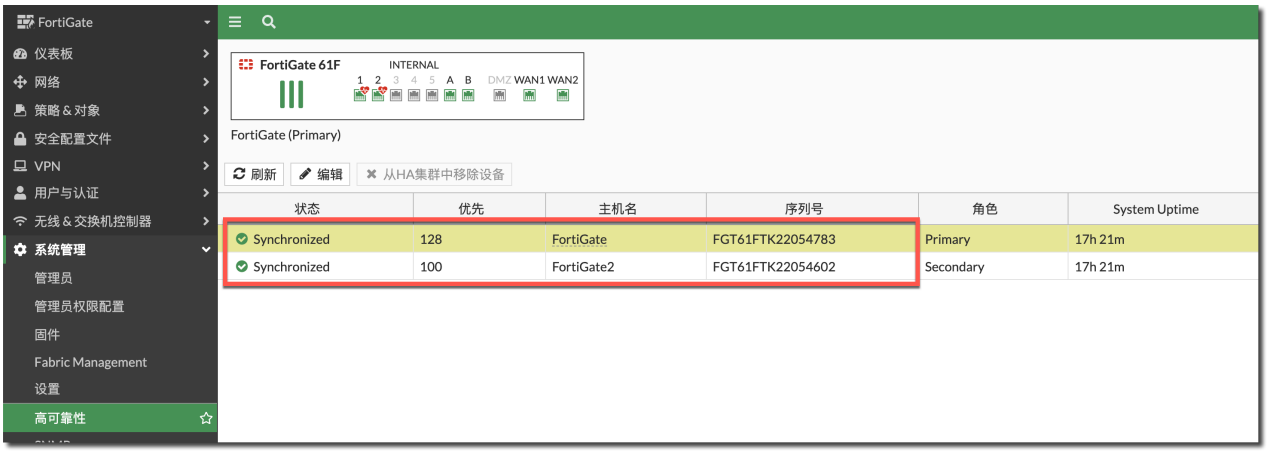
配置步骤

准备事项

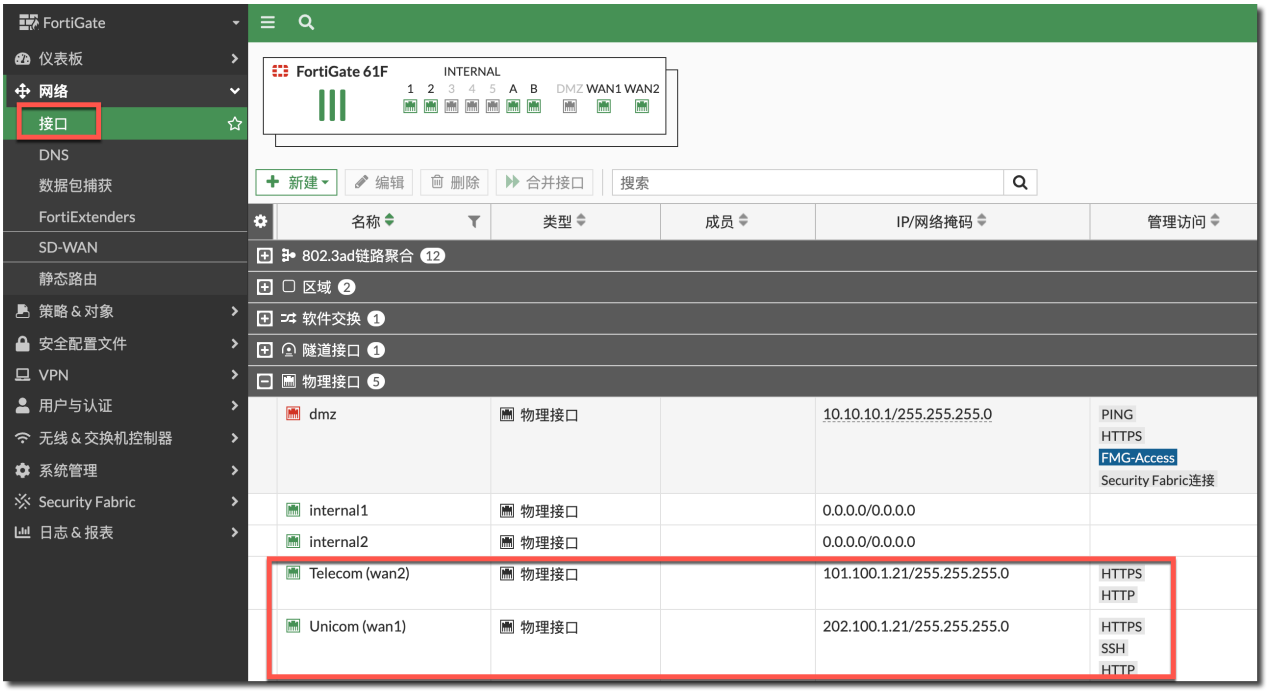
1. 首先确认 FortiGate 如下全局配置已开启（一般默认为开启状态）。

```
config system settings
    set gui-switch-controller enable
end
```

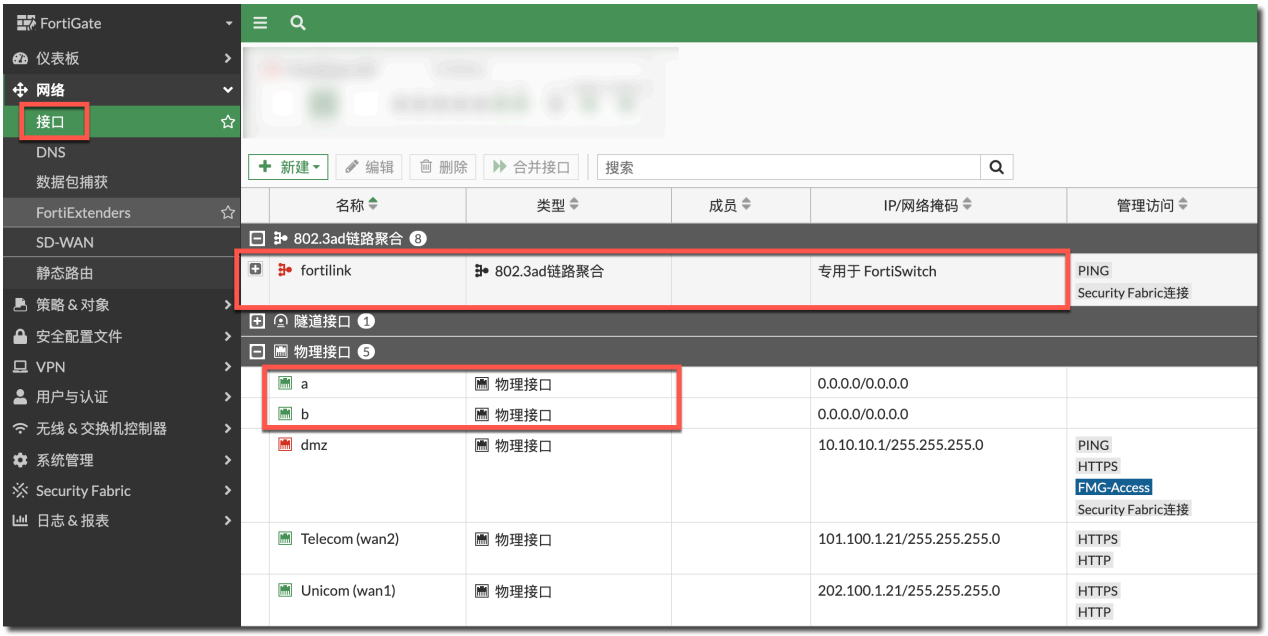
2. 配置两台 FortiGate 的 HA 主备模式（配置方法请参考 HA 双机热备 → HA 典型基础配置 ↗）。



3. 配置 wan1 和 wan2 的 IP 地址。



4. 配置 FortiLink 实现 FortiSwith 上线，我们只需要将用于 FortiLink 的成员接口加入 FortiLink 接口即可。此拓扑中，FortiGate 用于连接 FortiSwitch 的接口为 a 和 b，编辑 FortiLink 接口，将 a 和 b 加入 FortiLink（某些产品型号会默认将两个接口加入 FortiLink），开启自动授权设备，关闭 FortiLink 分离接口，其他配置保持默认，可以看到 FortiLink 为一个聚合接口。

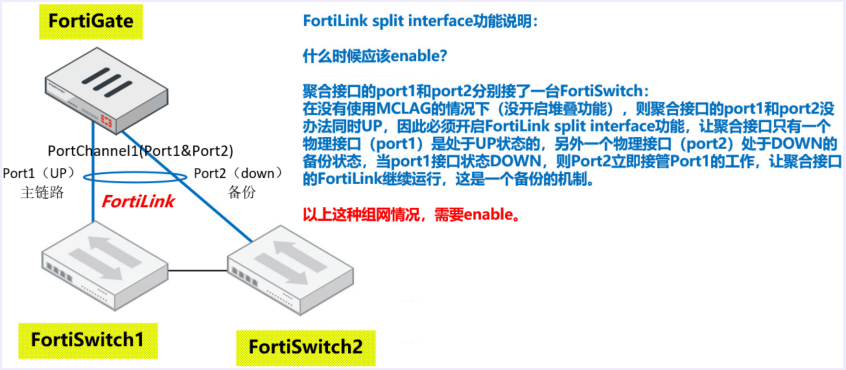




重要

FortiLink 分离接口什么时候应该开启?

分离接口的意是，如果聚合接口的 port1 和 port2 分别连接了一台 SW，在没有使用 MCLAG 的情况下（没开启堆叠功能），则聚合接口的 port1 和 port2 没办法同时 UP（同时开启聚合接口无法正常工作），因此必须开启 FortiLink 分离接口功能，让聚合接口只有一个物理接口（port1）是处于 UP 状态的，另外一个物理接口（port2）处于 DOWN 的备份状态，当 port1 接口状态 DOWN，则 Port2 立即接管 Port1 的工作，让聚合接口的 FortiLink 继续运行，这是一个聚合接口在 FortiLink 管理时候的备份机制。

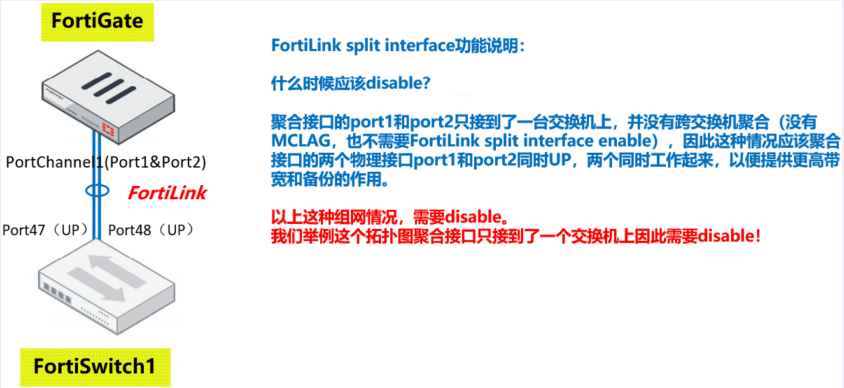


MCLAG 配置前是上述组网，因此刚刚开始的时候需要，保持 enable 状态。聚合接口的 Port1 和 Port2 只会 UP 一个接口，另外一个接口处于 DOWN 的状态。

重要

FortiLink split interface 什么时候应该 disable?

如果 FGT 的聚合接口只接到了一台交换机上，并没有跨交换机聚合（没有 MCLAG，也不需要 FortiLink split interface），因此这种情况应该让聚合接口的两个物理接口 port1 和 port2 同时 UP，两个接口同时工作起来，以便提供更高带宽和备份的作用。



配置两台交换机 MCLAG 堆叠之后，就是上述组网，因此交换机堆叠完毕之后，需要修改为 disable，让两个聚合接口都可以同时工作起来，最大的利用聚合接口的带宽和资源。这个配置后续会说明。

相关 CLI：

```
NTP:
config system ntp
    set ntpsync enable
    set server-mode enable
    set interface "fortilink"
end
--------------------------------------------------------------------------------
DHCP:
config system dhcp server
    edit 2
        set dns-service local
        set ntp-service local
        set default-gateway 10.255.1.1
        set netmask 255.255.255.0
        set interface "fortilink"
        config ip-range
            edit 1
                set start-ip 10.255.1.2
                set end-ip 10.255.1.254
            next
        end
        set vci-match enable
        set vci-string "FortiSwitch" "FortiExtender"
    next
end
--------------------------------------------------------------------------------
FortiLink:
config system interface
    edit "fortilink"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set type aggregate
        set member "a" "b"
        set lldp-reception enable
        set lldp-transmission enable
        set auto-auth-extension-device enable
        set fortilink-split-interface enable
        set switch-controller-nac "fortilink"
        set switch-controller-dynamic "fortilink"
        set swc-first-create 255
    next
end
```

连接 FortiLink

连接核心交换机

1. 配置完 FortiGate 之后，这个时候再去接 FortiGate 的 FortiLink 到 FortiSwitch 的线，配置 Core-FSW1 和 Core-FSW2 设备的互联接口 MCLAG 注意，先不要把所有的线都接上去，一台一台的来操作，MCLAG 配置尽量使用 Console 去配置，而不是 GUI，有时候由于在 Fortilink 计算和收敛会导致 GUI 连接不上去，或者先将 CORE - FSW1 下线，先单独配置 CORE - FSW2，然后再将 CORE - FSW2 下线，再单独配置 CORE - FSW1，然后再将二者互联的线接上去，这样是最稳妥的办法，避免 GUI 的 CLI 连不上的情况出现。而接入交换机这个时候是不建议接入的，等 CORE - FSW1 和 CORE - FSW2 的堆叠形成了之后，再将接入层交换机的线。

   > **重要**
   >
   > FortiSwitch 如果以前有使用过，建议都恢复一下出厂设置，然后再加入到 FortiLink。

2. FortiSwitch 的靠后接口一般都默认开启了 FortiLink 接口属性，如果接前面的接口可能会无法加入到 FortiLink，因此 FortiSwitch 的 FortiLink 最好都使用靠后的接口进行接线。FortiSwitch 各型号默认开启 FortiLink 功能的接口如下：

FortiSwitch 型号	默认开启 FortiLink 的接口
FS-108D-POE	port9–port10
FS-108E, FS-108E-POE, FS-108E-FPOE	port7–port10
FSR-112D-POE	port5–port12
FS-124D, FS-124D-POE	port23–port26
FSR-124D	port1-port4, port21–port28
FS-124E, FS-124E-POE, FS-124E-FPOE	port21–port28
FS-148E, FS-148E-POE	port21–port52
FS-224D-POE	port21–port24
FS-224D-FPOE	port21–port28
FS-224E, FS-224E-POE	port21–port28
FS-248D, FS-248D-FPOE	port45–port52
FS-248D-POE	port47–port50
FS-248E-POE, FS-248E-FPOE	port45–port52
FS-424D, FS-424D-POE, FS-424D-FPOE	port23–port26
FS-424E-Fiber	port1-port30
FS-426E-FPOE-MG	port23-port30
FS-448D, FS-448D-POE, FS-448D-FPOE	port45–port52
FS-524D, FS-524D-FPOE	port21–port30
FS-548D	port39–port54
FS-548D-FPOE, FS-548DN	port45–port54
FS-1024D	port1–port24
FS-1048D, FS-1048E	port1–port52
FS-3032D, FS-3032E	port1–port32

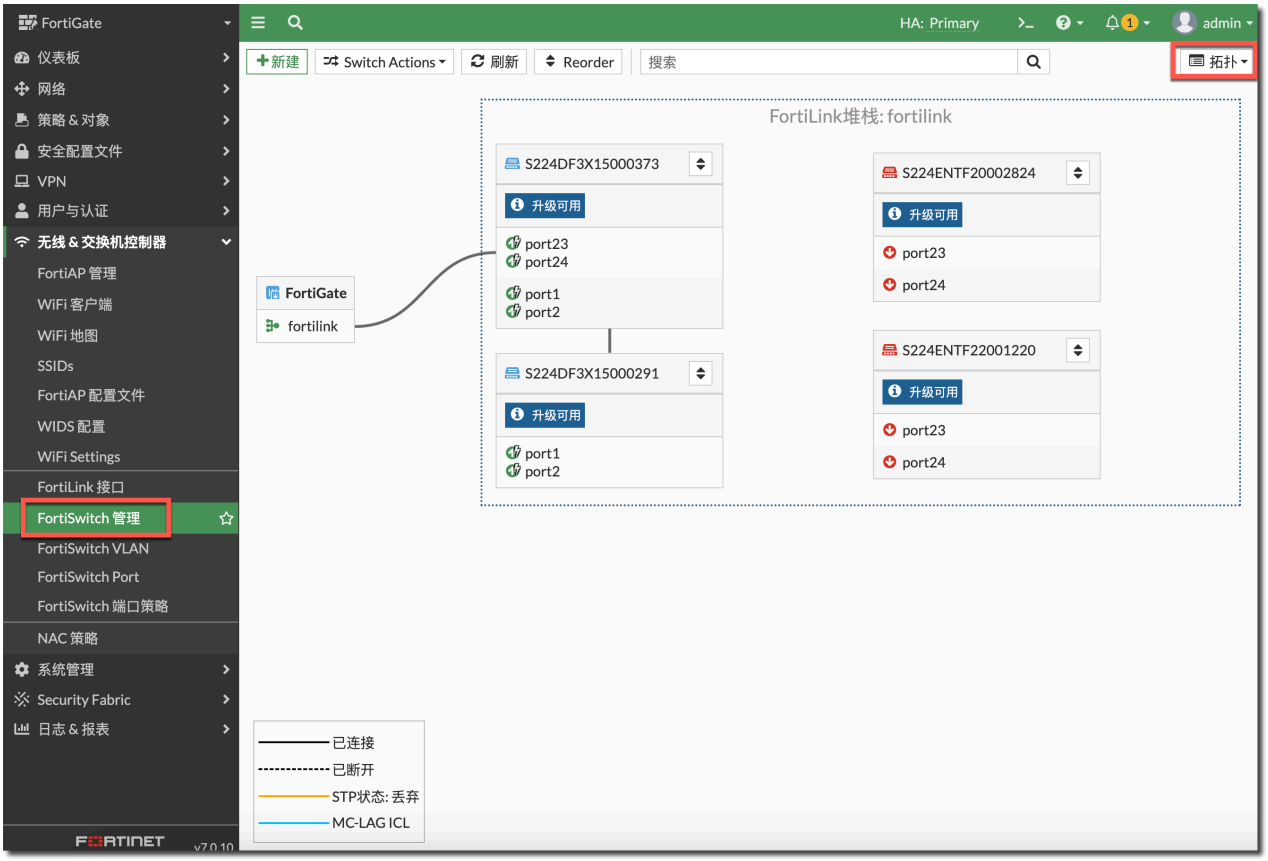
3. 如果 FortiSwitch 无法被 FortiGate 管理，登陆到 FortiSwitch 上进行确认接口是否开启了 FortiLink。

```
FortiSwitch # show switch interface port9
config switch interface
    edit "port9
        set auto-discovery-fortilink enable    //接口下又此命令的接口才可以自动加入到FortiLink中
    next
end
```

4. 交换机之间如果接多根线，会自动进行聚合，但是 FortiLink 自动聚合会花比较长的时间，建议可以先接单线，等 FortiLink 的拓扑较快的形成结束之后，再增加彼此之间的另外一根线，再进行自动聚合，这样效率会比较高一些。

5. 接线是很重要的步骤，仔细看拓扑图，做好接口标签，不要接错。

6. 先接 Core-FSW1 的线，再接 Core-FSW1 和 Core-FSW2 互联的线，先不要接接入层交换机的线，在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，右上角选择拓扑视图，查看交换机被管理状态。

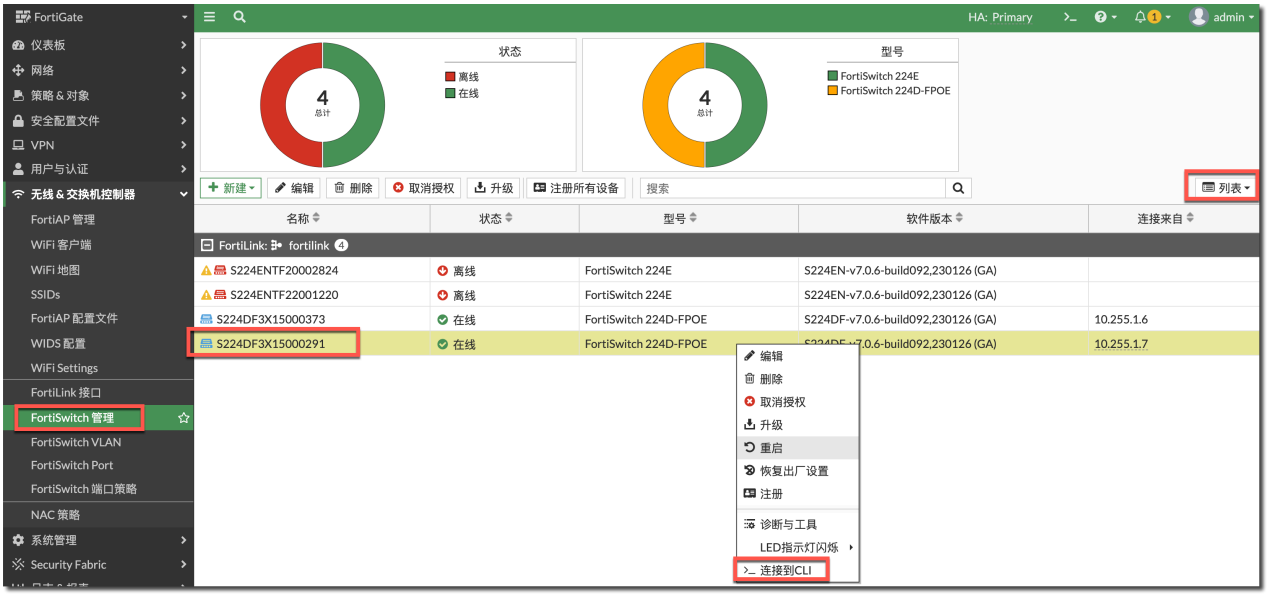


7. 此时我们看防火墙的 FortiLink 聚合接口，此时应该只有 a 或 b 其中一个物理接口是 UP 的，这是 FortiLink split interface enable 的结果。

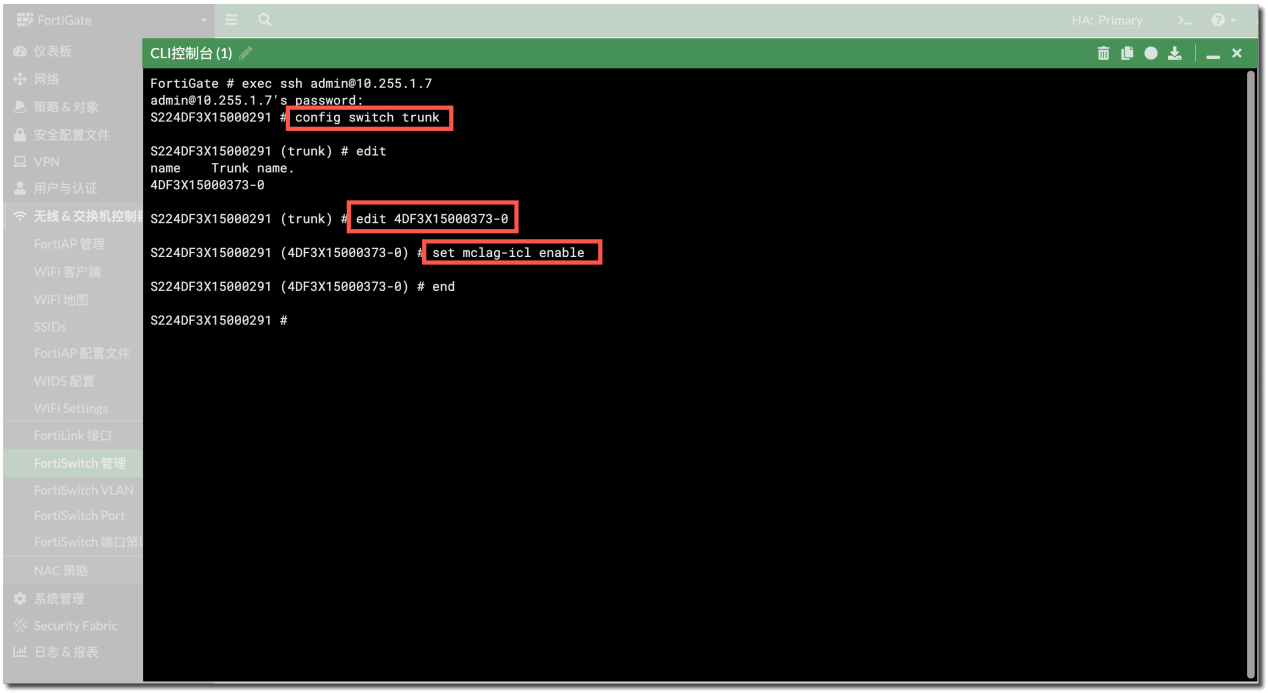


配置 MCLAG

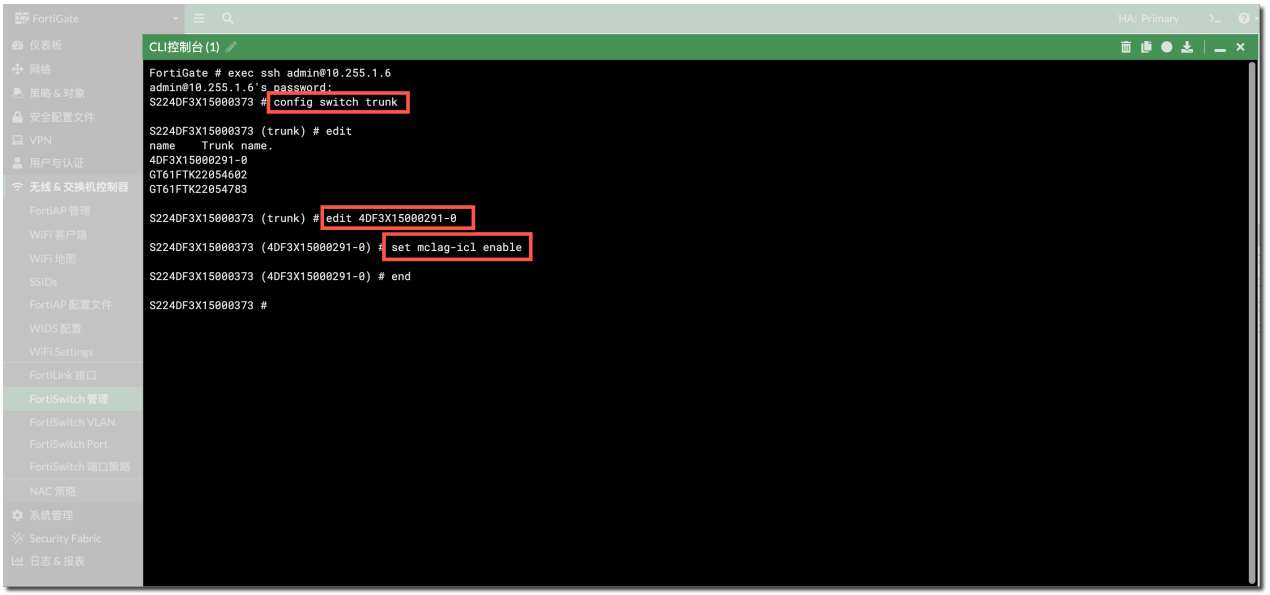
1. 通过 FortiGate 管理交换机页面右键交换机即可 SSH 到交换机，分别登录到 Core-FSW1 和 Core-FSW2 上。



2. 配置二者的互联聚合接口为“set mclag-icl enable”（请注意此例中在 GUI 配置时一定要先配置 CORE - FSW2，因为 CORE - FSW2 与 FortiGate 的 FortiLink 接口处于 down 的状态，如果先配置了 CORE - FSW1，那么 CORE - FSW2 将无法通过 GUI 管理）。

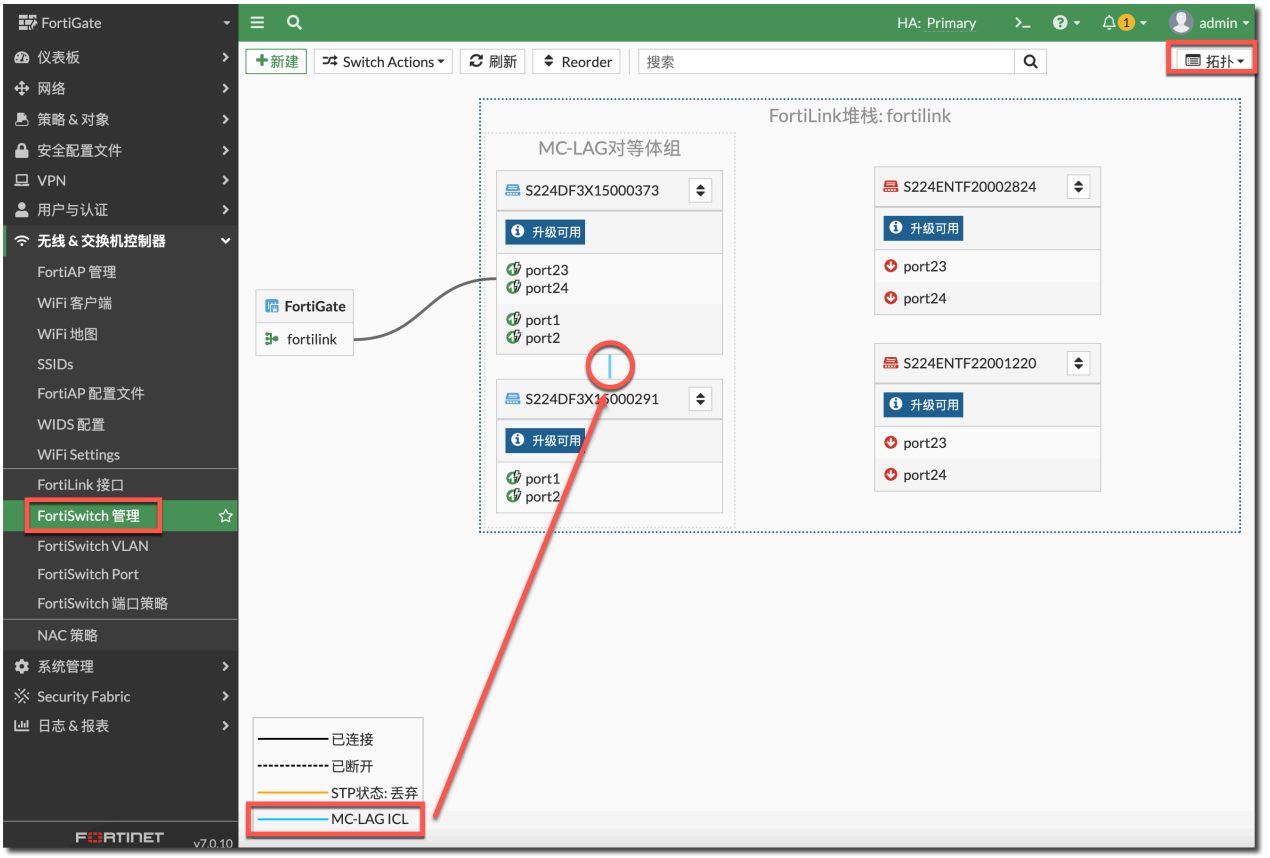



```
config switch trunk
    edit "4DF3X15000373-0"
        set mclag-icl enable
    next
end
```

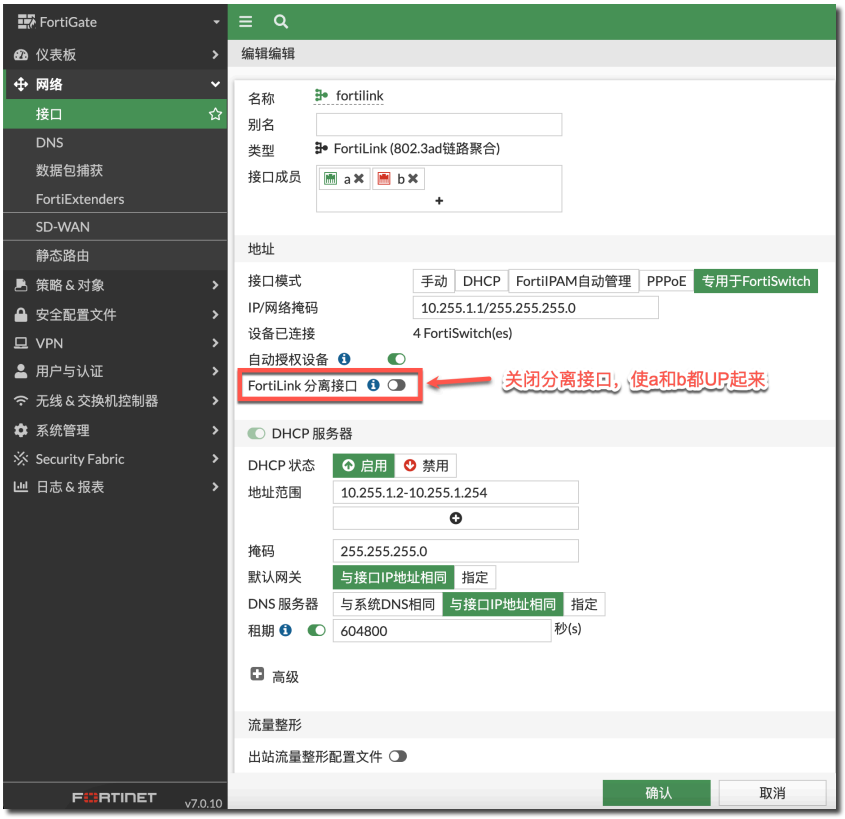



```
config switch trunk
    edit "4DF3X15000291-0"
        set mclag-icl enable
    next
end
```

3. 查看配置 MCLAG 后的交换机拓扑，可以看到两台交换机已经变为一个 MC - LAG 组。

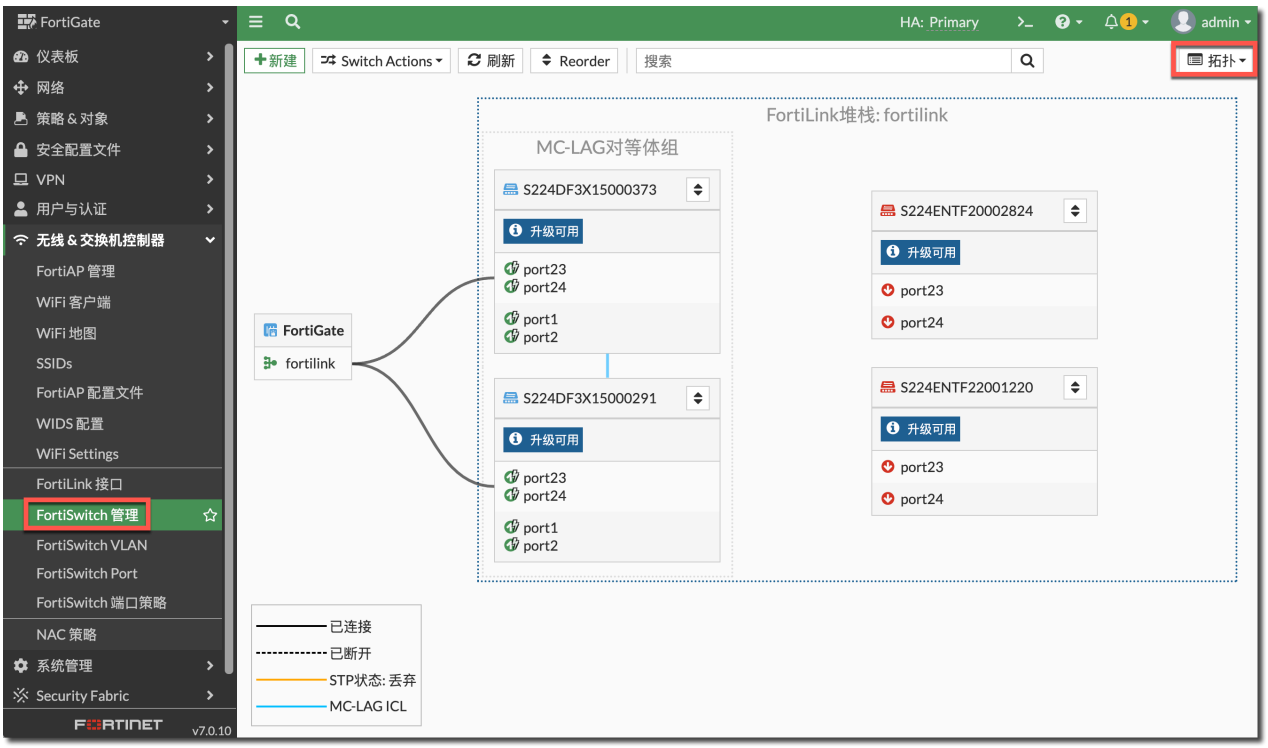


4. MC - LAG 配置完成后，将 FortiLink 接口下的 split interface 设置为 disable，让聚合接口（a 和 b）全部 UP 起来，充分利用聚合接口的资源，实现 FortiGate 与两台交换机（堆叠）的跨交换机聚合。




```
config system interface
    edit "fortilink"
        set fortilink-split-interface disable
    next
end
```

5. 等待 CORE - FSW2 重新在 FortiGate 上线后，查看 Switch 管理拓扑，FortiLink 的两个成员接口均已 UP。

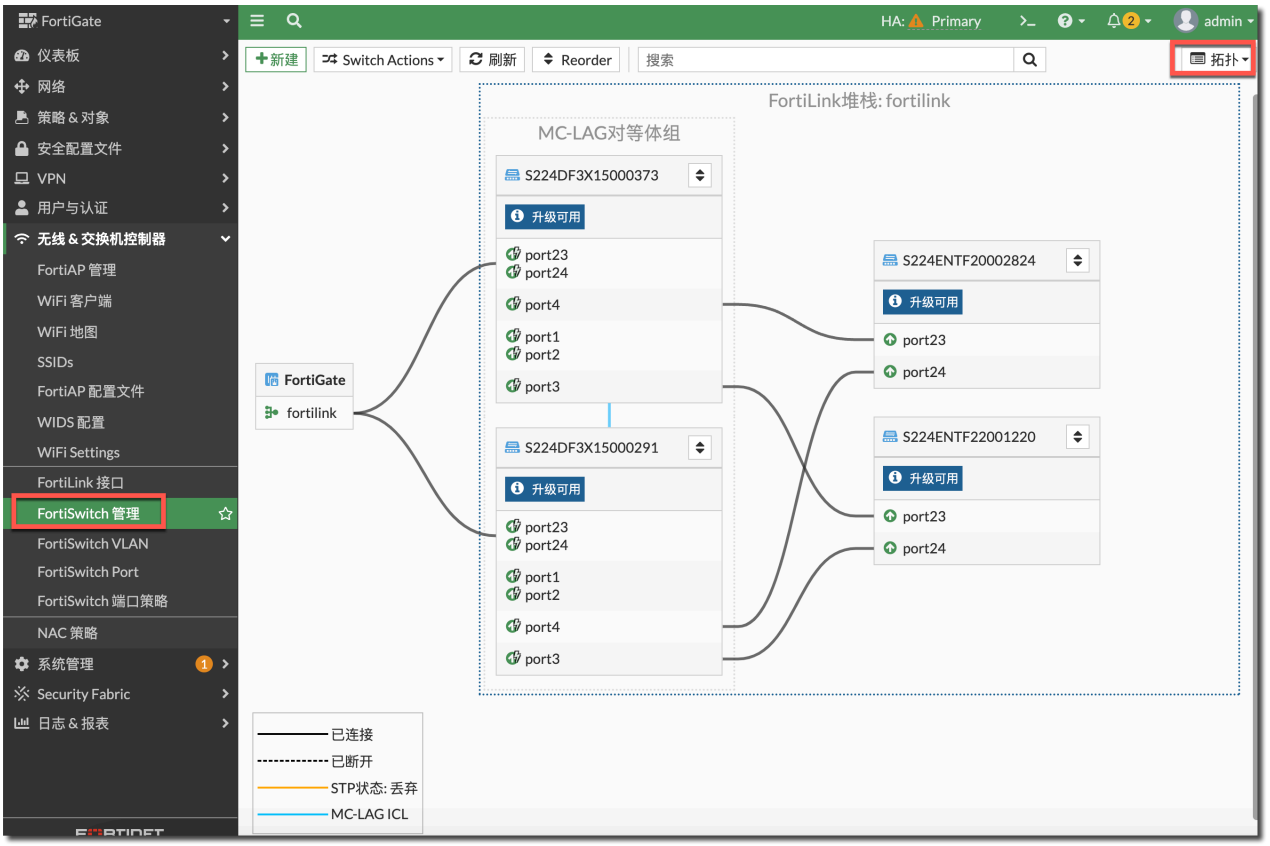


6. 在 FortiGate 的 CLI 下查看 FortiLink 成员状态，可以看到与实际物理连线一致。

```
FortiGate # execute switch-controller get-physical-conn standard fortilink

This will display connectivity graph information for FortiLink from FortiGate's perspective
NOTE : If FortiSwitch is not authorized, no connectivity information will be shown
NOTE : If FortiSwitch is in idle state, no connectivity information will be shown
NOTE : If FortiSwitch ISL peer has inconsistent info, no connectivity information will be shown
       FortiLink interface : fortilink
FortiGate(s)
FGT61FTK22054783(a)  <<----------->>  S224DF3X15000373(port23)
FGT61FTK22054602(a)  <<----------->>  S224DF3X15000373(port24)
FGT61FTK22054783(b)  <<----------->>  S224DF3X15000291(port23)
FGT61FTK22054602(b)  <<----------->>  S224DF3X15000291(port24)

Tier 1
S224DF3X15000373(port23)  <<----------->>  FGT61FTK22054783(a)
S224DF3X15000373(port24)  <<----------->>  FGT61FTK22054602(a)
S224DF3X15000291(port23)  <<----------->>  FGT61FTK22054783(b)
S224DF3X15000291(port24)  <<----------->>  FGT61FTK22054602(b)

Tier 2+
S224DF3X15000373(port1/4DF3X15000291-0)  <<----------->>  S224DF3X15000291(port1/4DF3X15000373-0)
S224DF3X15000373(port2/4DF3X15000291-0)  <<----------->>  S224DF3X15000291(port2/4DF3X15000373-0)
```

连接接入交换机

1. 将接入层两台交换机的连线接入。在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，查看交换机被管理状态，修改视图为拓扑，可以看到 4 台交换机均已上线。



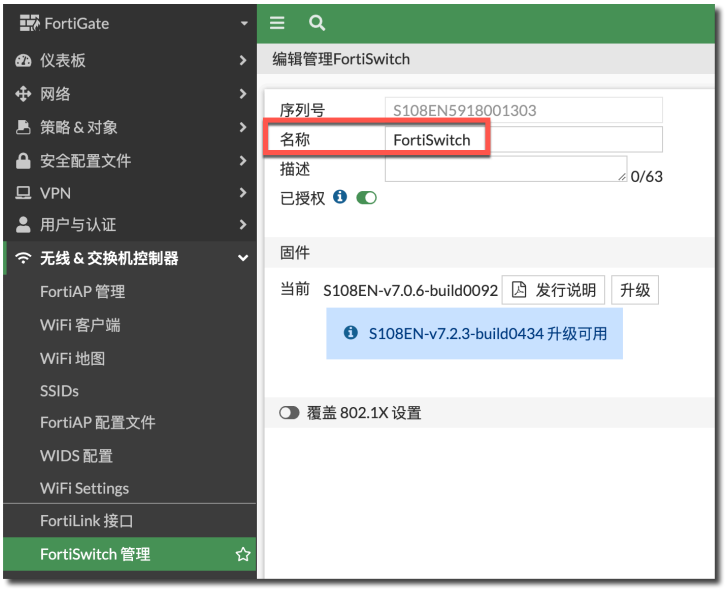
2. 接好线之后，FortiSwitch 和 FortiGate 之间会自动在后台进行 FortiLink、CAPWAP、MSTP、聚合接口的协商加入等操作，等 FortiLink 的管理结束之后，在 FortiGate 上会看到一个网站的 FortiSwitch 拓扑图，和我们规划的拓扑图应该是一致的，可以仔细核对接线是否正确。

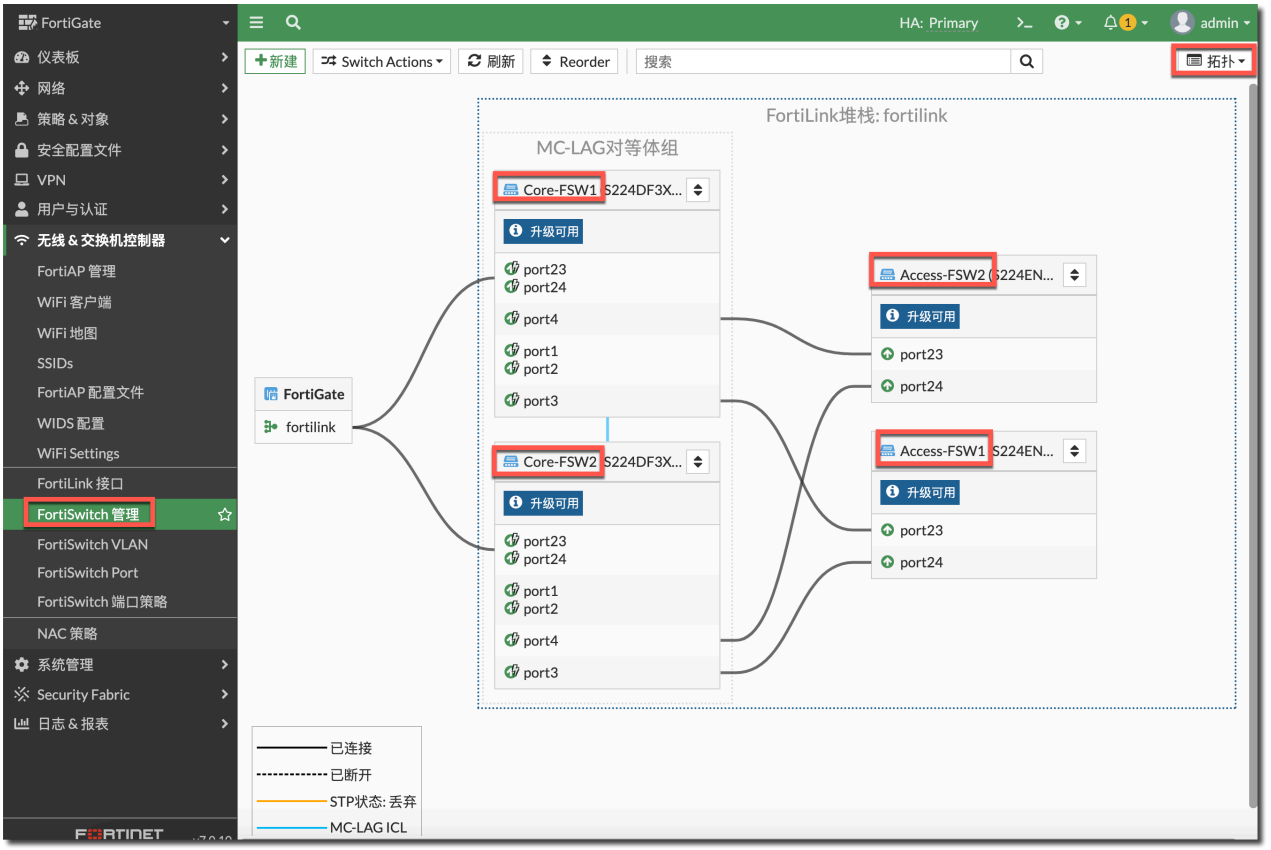
3. 交换机加入到 FortiGate 的 FortiLink，会先清除配置再重启，然后从 standalone 的状态进入到“Fortilink remote control”的状态，后续所有的配置都通过 FortiGate 下发即可。

4. 再次查看 FortiGate 的 FortiLink 成员，可以看到两台接入交换机已上线，且连线与拓扑一致。

```
FortiGate # execute switch-controller get-physical-conn standard fortilink

This will display connectivity graph information for FortiLink from FortiGate's
        FortiLink interface : fortilink
FortiGate(s)
FGT61FTK22054783(a)  «————————————»  S224DF3X15000373(port23)
FGT61FTK22054602(a)  «————————————»  S224DF3X15000373(port24)
FGT61FTK22054783(b)  «————————————»  S224DF3X15000291(port23)
FGT61FTK22054602(b)  «————————————»  S224DF3X15000291(port24)
Tier 1
S224DF3X15000373(port23)  «————————————»  FGT61FTK22054783(a)
S224DF3X15000373(port24)  «————————————»  FGT61FTK22054602(a)
S224DF3X15000291(port23)  «————————————»  FGT61FTK22054783(b)
S224DF3X15000291(port24)  «————————————»  FGT61FTK22054602(b)
Tier 2+
S224ENTF20002824(port23/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000373(port4/4ENTF20002824-0)
S224ENTF22001220(port23/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000373(port3/4ENTF22001220-0)
S224DF3X15000373(port1/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port1/4DF3X15000373-0)
S224DF3X15000373(port2/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port2/4DF3X15000373-0)
```

5. 编辑交换机，可以修改交换机名字以及升级版本，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。



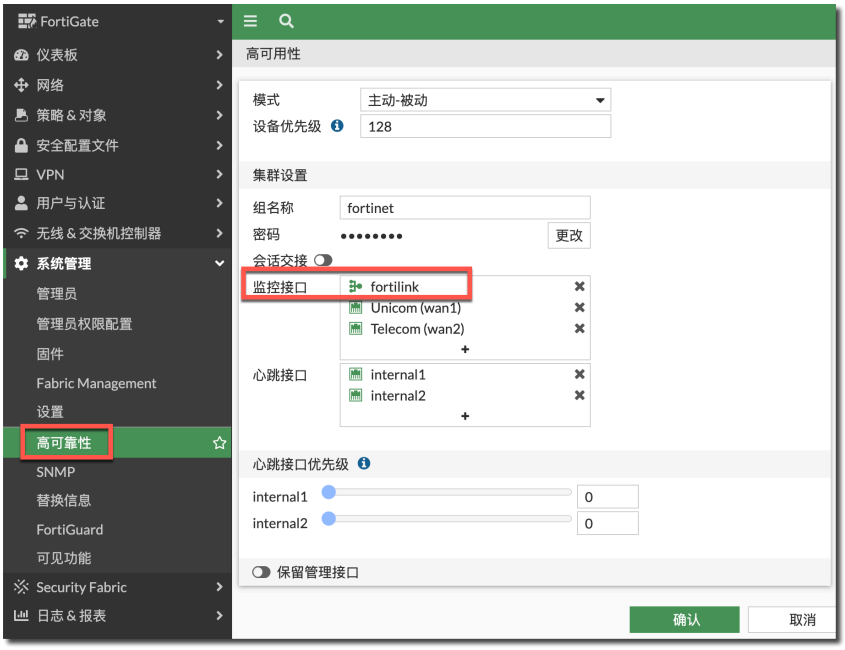


6. FortiSwitch 之间接多根线进行 FortiLink 互联，会默认自动聚合。其实接一根网线也是聚合的，只是聚合组里面只有一个接口而已，因此多根网线直连，这些直连的线都会自动加入到一个聚合接口里面（建议 MC - LAG 环境尽量使用两条或以上的连线）。

```
FortiGate # exec ssh admin@10.255.1.6
admin@10.255.1.6's password:

Core-FSW1 # show switch trunk

config switch trunk
    edit "GT61FTK22054602"
        set mode lacp-active
        set auto-isl 1
        set fortilink 1
        set mclag enable
        set members "port24"
    next
    edit "GT61FTK22054783"
```

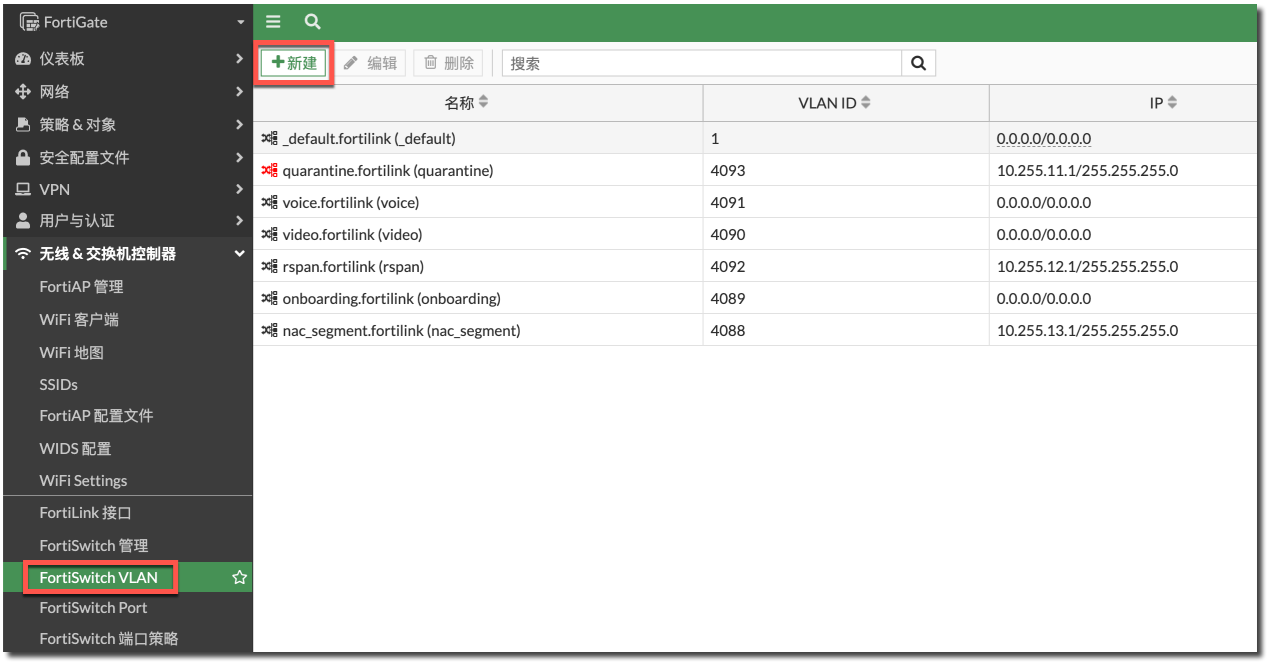
```
        set mode lacp-active
        set auto-isl 1
        set fortilink 1
        set mclag enable
        set members "port23"
    next
    edit "4DF3X15000291-0"
        set mode lacp-active
        set auto-isl 1
        set mclag-icl enable
        set members "port1" "port2"
    next
    edit "4ENTF20002824-0"
        set mode lacp-active
        set auto-isl 1
        set mclag enable
        set members "port4"
    next
    edit "4ENTF22001220-0"
        set mode lacp-active
        set auto-isl 1
        set mclag enable
        set members "port3"
    next
end
```

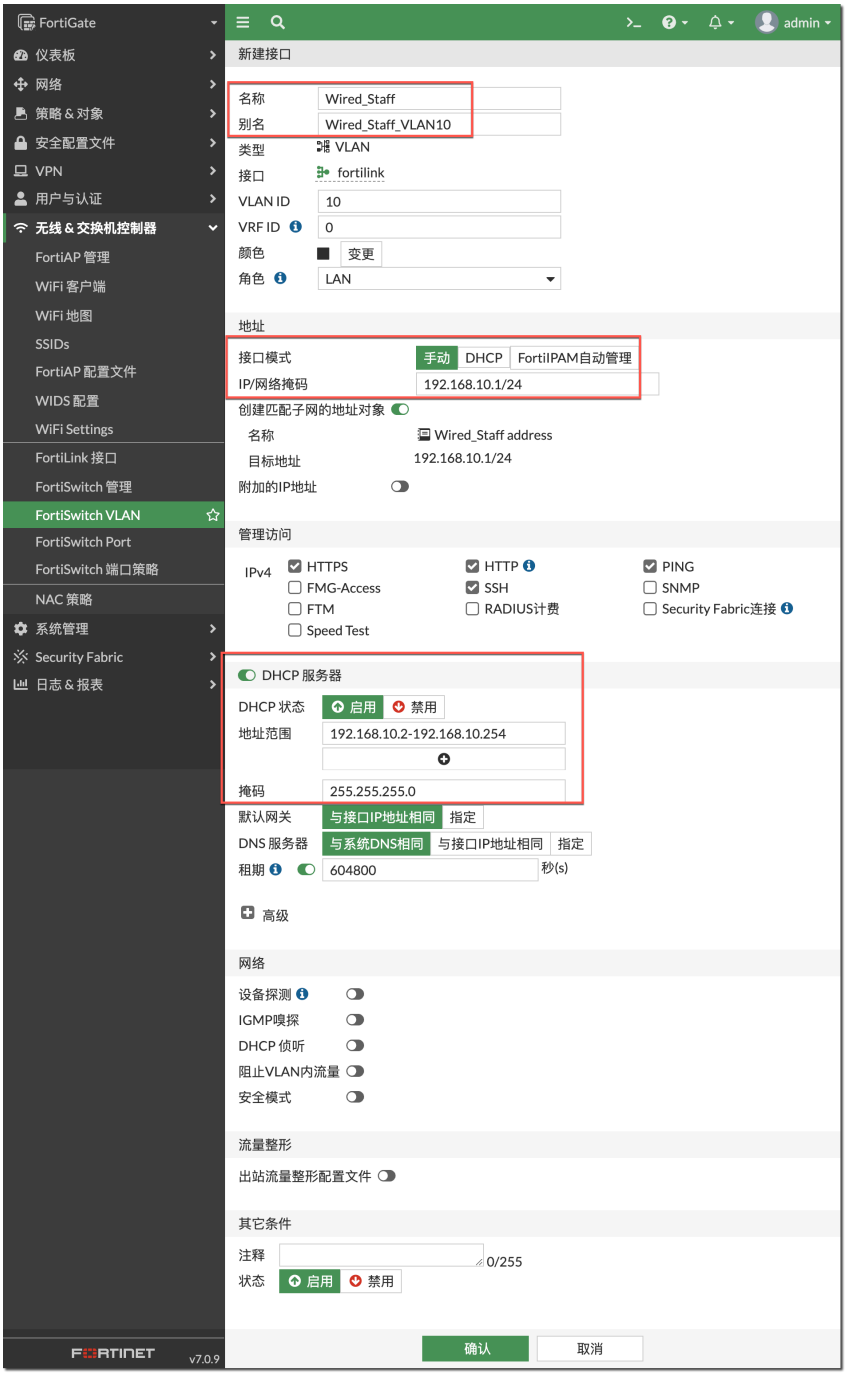
7. 最后不要忘记，在 FortiLink 的聚合接口和 FortiLink 管理配置完毕，记得把这个 FortiLink 聚合接口加入到 HA 的监控接口列表中，这样当主核心交换机挂了（比如断电）之后，业务才可以平稳切换到备防火墙上。

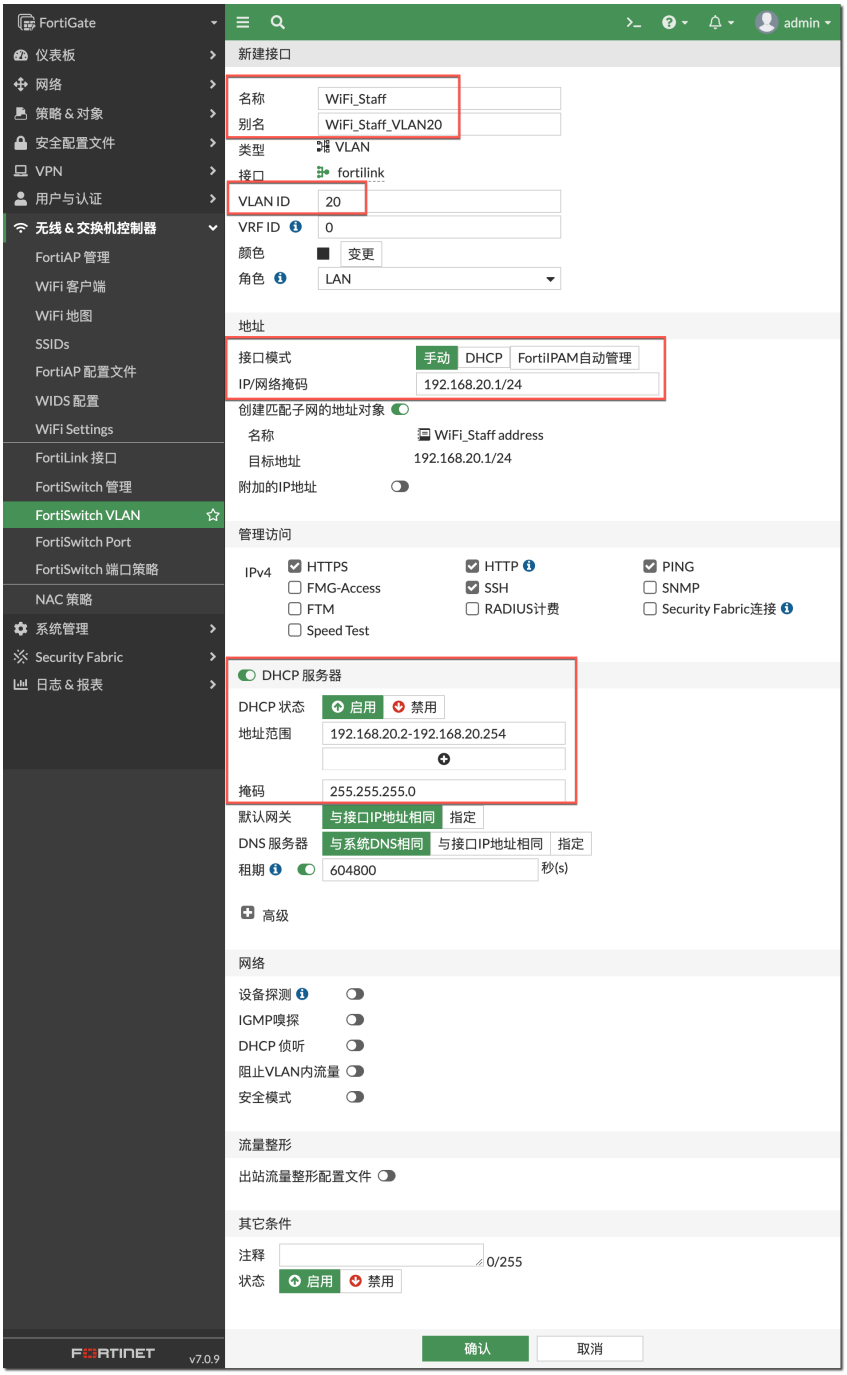


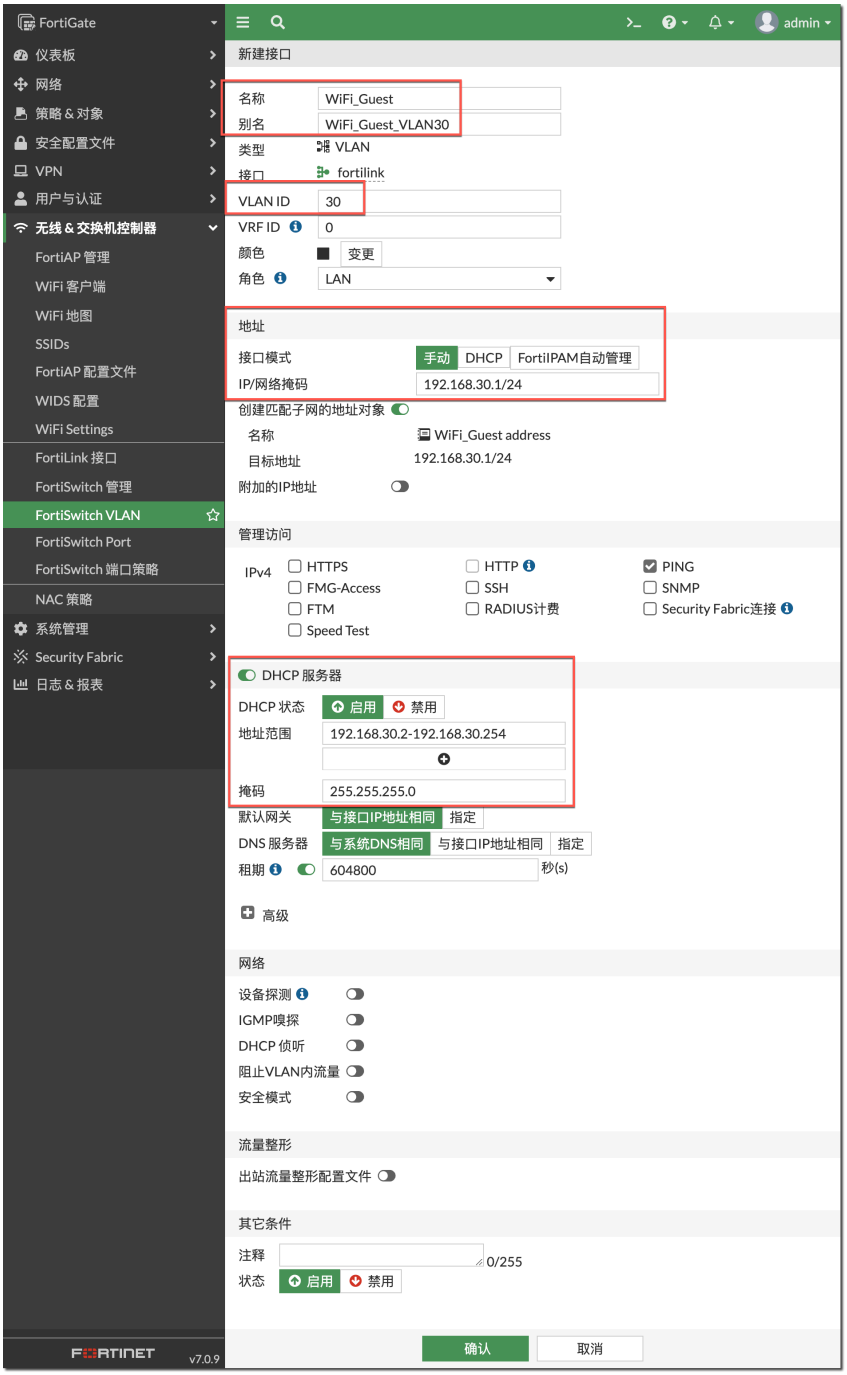
创建 VLAN

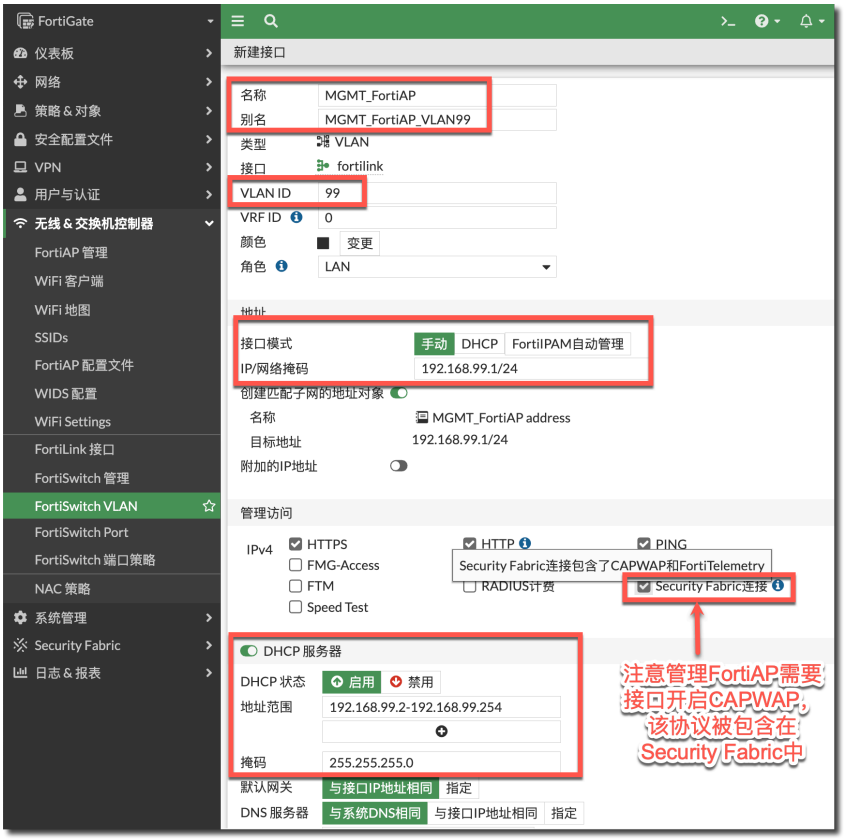
1. FortiGate 通过 FortiLink + Capwap 协议统一的接管 FortiSwitch 和 FortiAP，为有线和无线统一进行管理和业务处理。
2. 进入 FortiGate 的无线 & 交换机控制器 → FortiSwitch VLAN 页面，新建有线网络 VLAN 10、员工无线网络 VLAN 20、访客无线网络 VLAN 30、管理 FortiAP VLAN 99。注意管理 FortiAP 的 VLAN 接口要开启 Security Fabric，包含 CAPWAP。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
+新建
编辑
删除
搜索
名称
VLAN ID
IP
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0

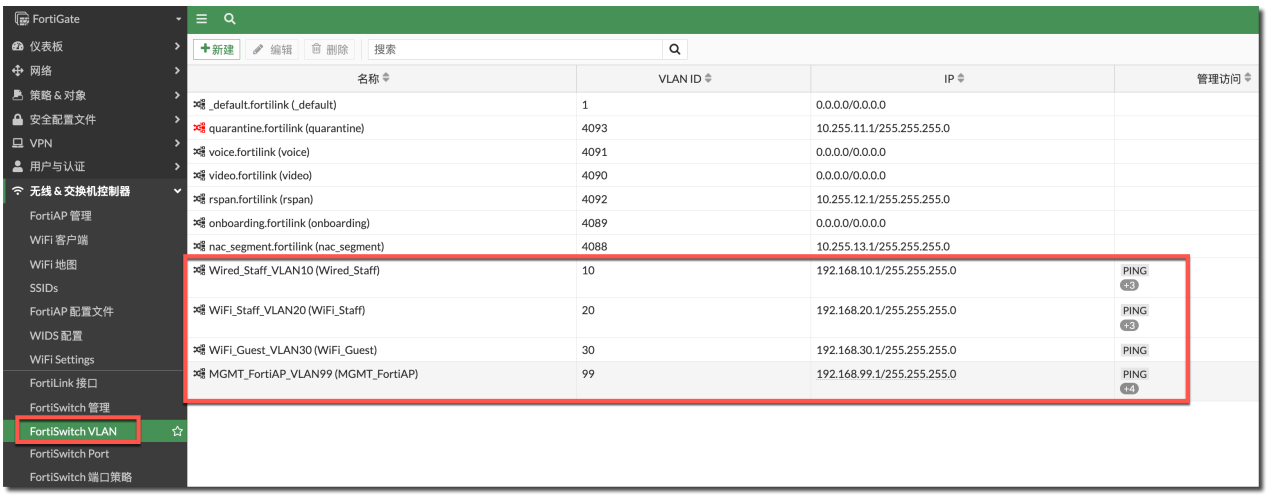

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 Wired_Staff
别名 Wired_Staff_VLAN10
类型 VLAN
接口 fortilink
VLAN ID 10
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.10.1/24
创建匹配子网的地址对象
名称 Wired_Staff address
目标地址 192.168.10.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.10.2-192.168.10.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 WiFi_Staff
别名 WiFi_Staff_VLAN20
类型 VLAN
接口 fortilink
VLAN ID 20
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.20.1/24
创建匹配子网的地址对象
名称 WiFi_Staff address
目标地址 192.168.20.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.20.2-192.168.20.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消

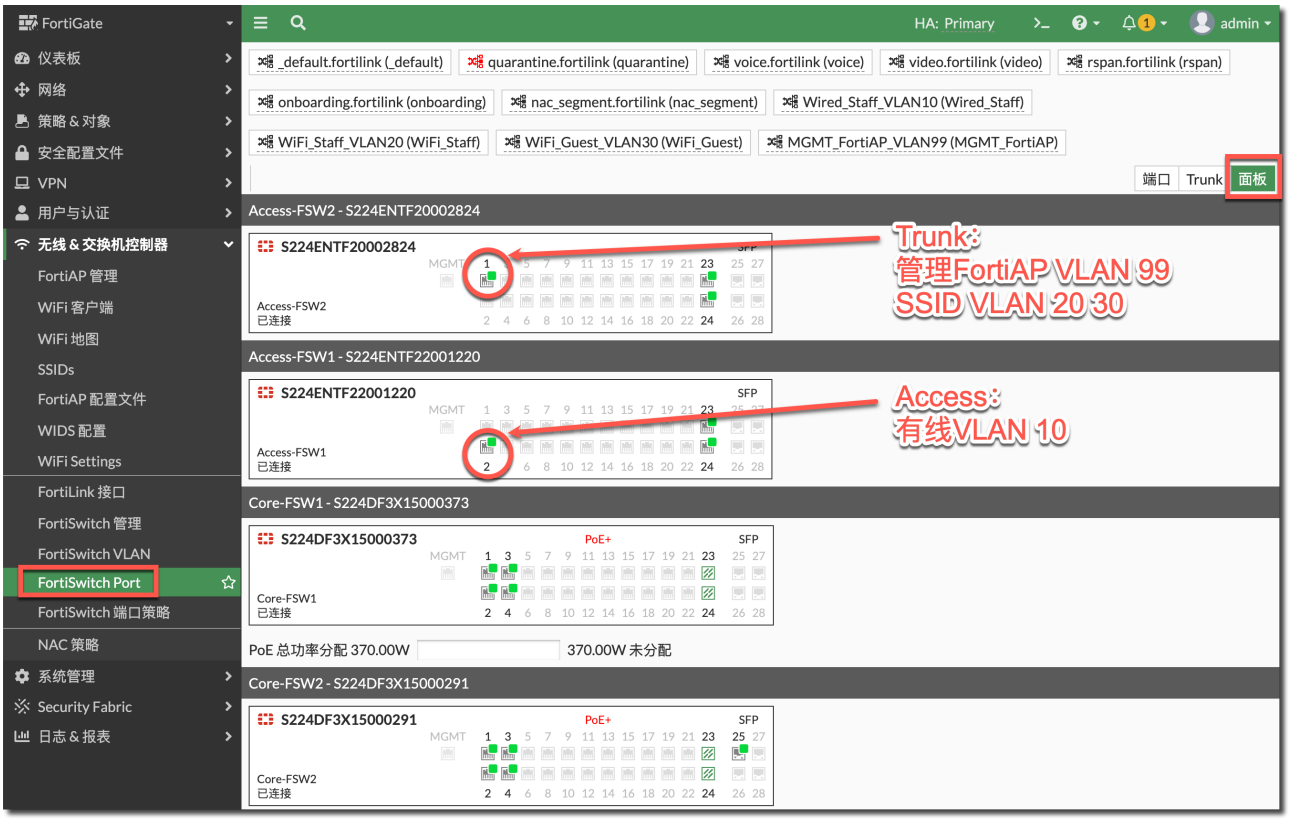

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
新建接口
名称 WiFi_Guest
别名 WiFi_Guest_VLAN30
类型 VLAN
接口 fortilink
VLAN ID 30
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.30.1/24
创建匹配子网的地址对象
名称 WiFi_Guest address
目标地址 192.168.30.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.30.2-192.168.30.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消
FORTINET
v7.0.9


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
新建接口
名称 MGMT_FortiAP
别名 MGMT_FortiAP_VLAN99
类型 VLAN
接口 fortilink
VLAN ID 99
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.99.1/24
创建匹配子网的地址对象
名称 MGMT_FortiAP address
目标地址 192.168.99.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access
FTM
Speed Test
Security Fabric连接包含了CAPWAP和FortiTelemetry
RADIUS计费
Security Fabric连接
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.99.2-192.168.99.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
注意管理FortiAP需要接口开启CAPWAP，该协议被包含在Security Fabric中

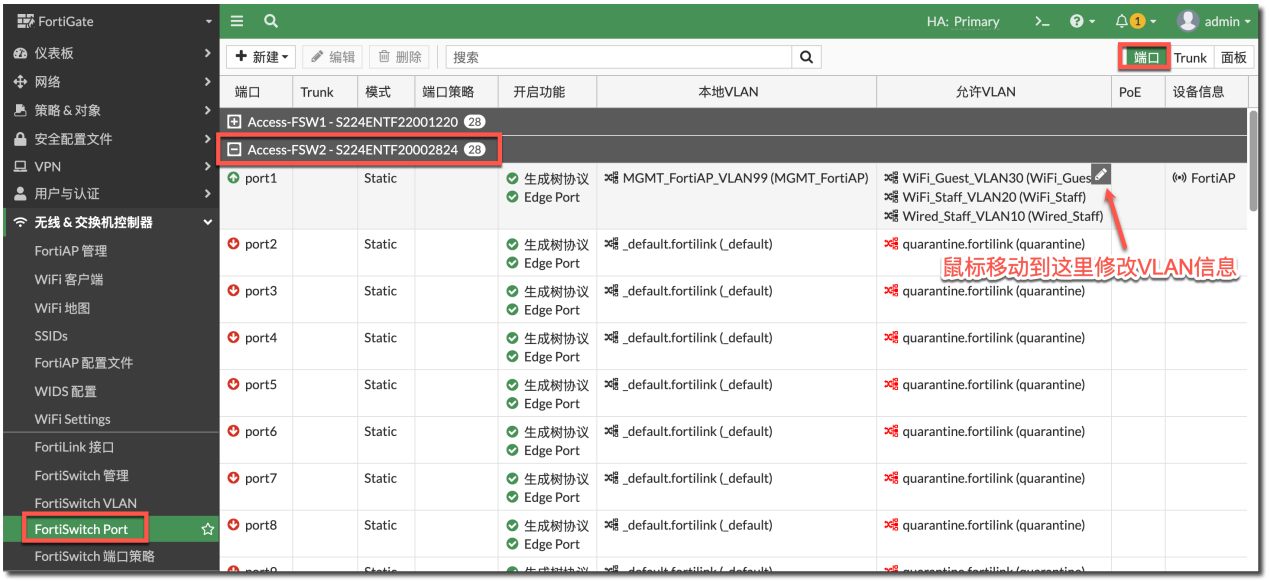
3. VLAN 创建完成。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
新建 编辑 删除 搜索
名称 VLAN ID IP 管理访问
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0
Wired_Staff_VLAN10 (Wired_Staff) 10 192.168.10.1/255.255.255.0 PING +3
WiFi_Staff_VLAN20 (WiFi_Staff) 20 192.168.20.1/255.255.255.0 PING +3
WiFi_Guest_VLAN30 (WiFi_Guest) 30 192.168.30.1/255.255.255.0 PING
MGMT_FortiAP_VLAN99 (MGMT_FortiAP) 99 192.168.99.1/255.255.255.0 PING +4

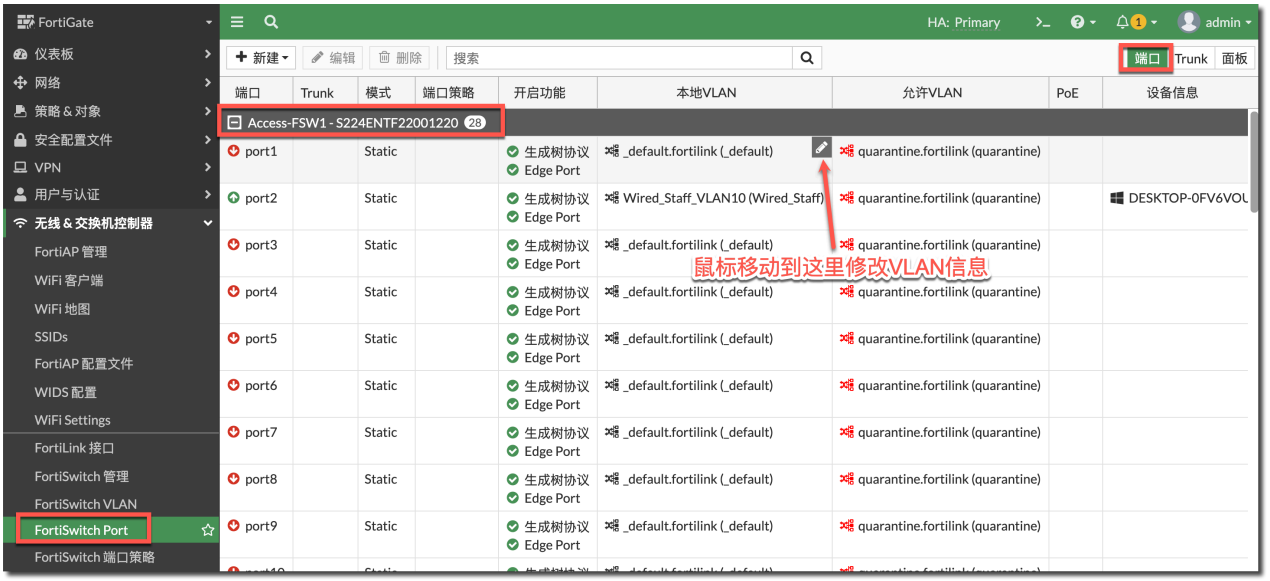
分配 VLAN



1. Access-FSW2 交换机的 port1 接 FortiAP，配置为 Trunk 模式，允许 VLAN20 和 VLAN30，VLAN20 用于 WIFI - STAFF 的 SSID 用户，VLAN30 用于 WIFI - GUEST 的 SSID 用户。另外本地 VLAN 设置为 VLAN 99，用于 FortiAP 的管理（FortiAP 的管理相当于使用不打 vlan-tag 的 VLAN 99，本地 VLAN 默认是 VLAN 1，因此只需要将本地 VLAN 设置为 VLAN 99 即可）。

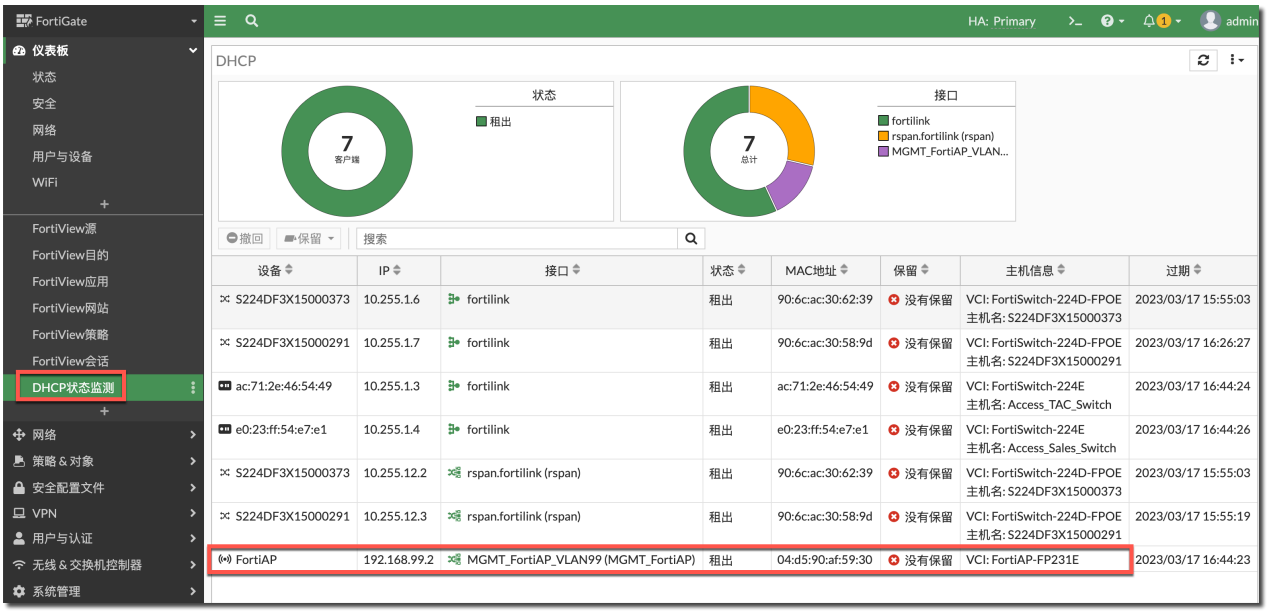


2. Access-FSW1 交换机的 Port2 接具体的物理 PC，因此只需要将 Port2 设置为 access 口，分配 VLAN 10 即可，用 FortiSwitch 的概念来说就是将本地 VLAN 设置为 VLAN 10，本地 VLAN 不打 vlan-tag，相当于就是设置接口为 access vlan 10。

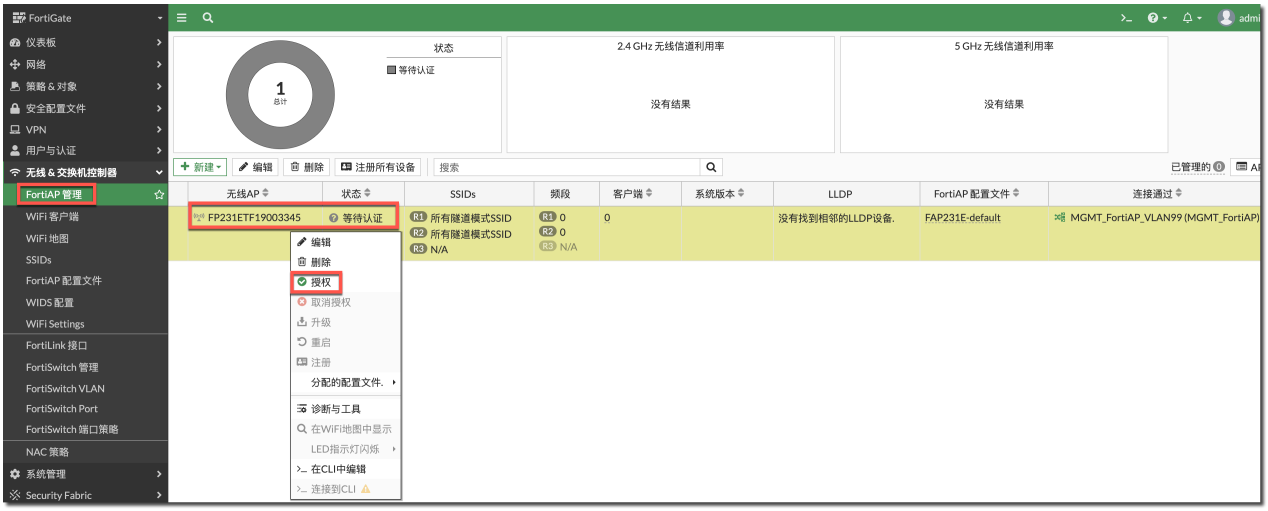


无线配置

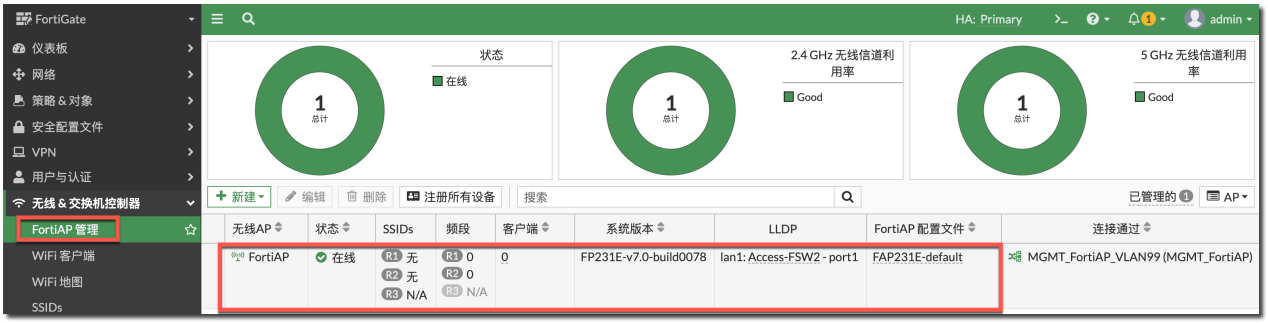
1. FortiAP 连接后，会通过 VLAN 99 的 DHCP 获取到 192.168.99.0/24 网段的 IP 地址。



2. 在之前的配置中，VLAN 99 的接口开启了 CAPWAP 协议（包含在 Security Fabric 中），激活了 FortiGate 的无线管理功能，这时候 FortiAP 会自动找到 FGT 进行注册。因此在 FortiAP 管理的地方可以看到这个 FortiAP 的信息（如果没有看到检查配置，并将 FortiAP 恢复出厂，让 FortiAP 重新获取 IP 地址并发起注册）。



3. 右键点击上图中等待认证的 FortiAP，点击授权按钮，等待几分钟，FortiAP 成功上线。



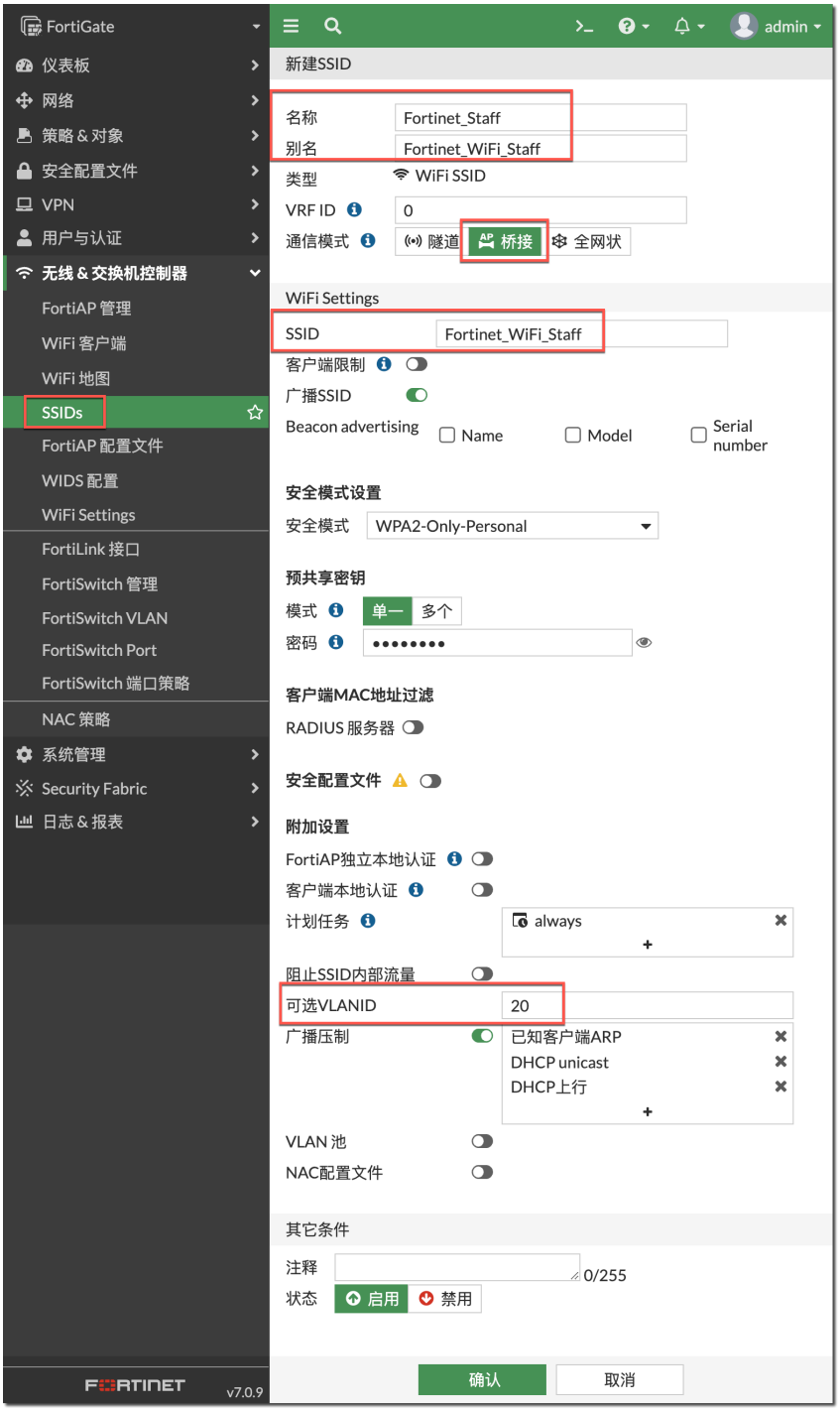
4. 编辑 FortiAP 设置，配置 FortiAP 的名称，或进行升级操作。

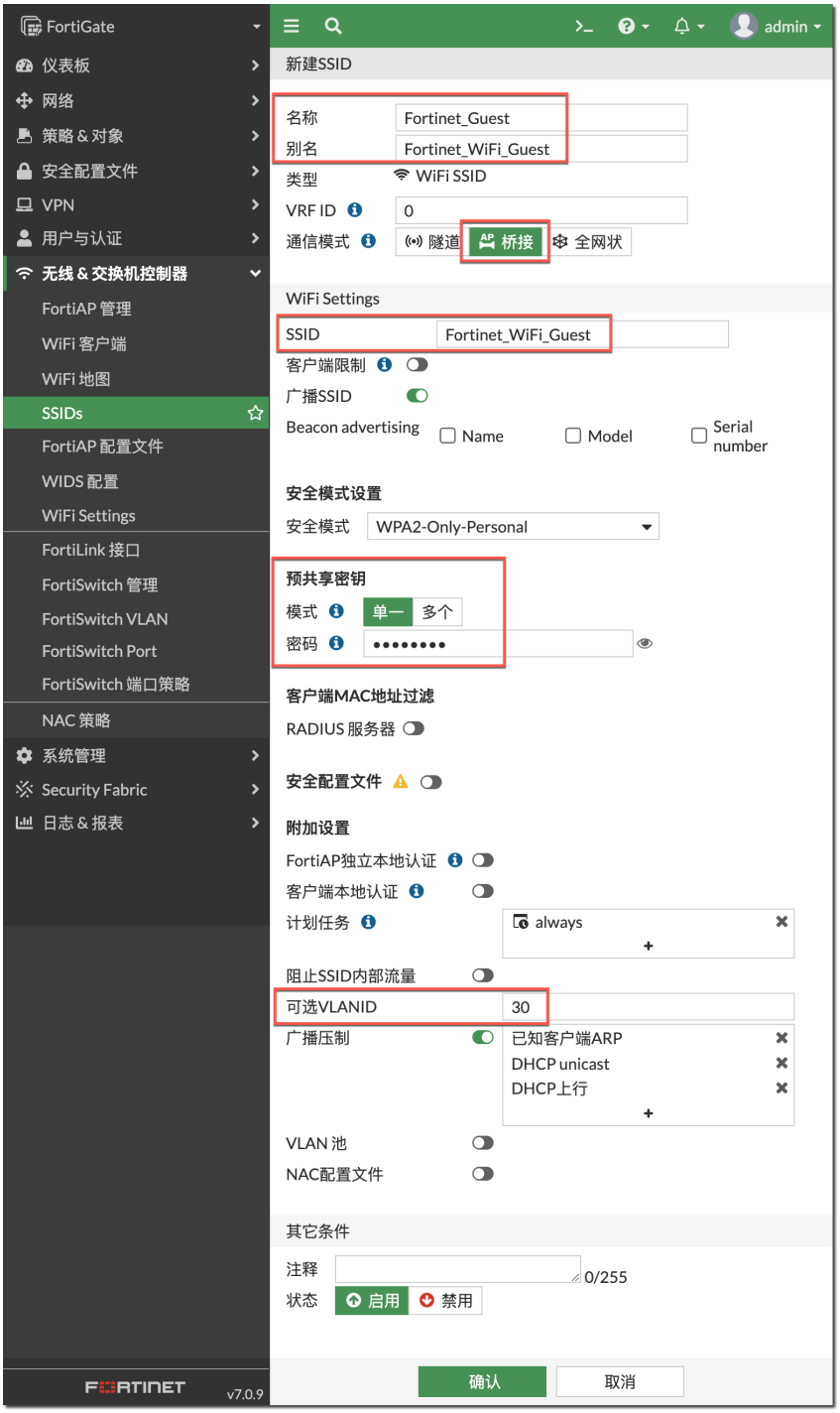


5. 进入无线 & 交换机控制器 → SSIDs 页面，点击新建按钮。



6. 新建 SSID：Fortinet_WiFi_Staff VLAN 20 用于员工 WIFI 使用，Fortinet_WiFi_Guest VLAN 30 用于访客 WiFi 使用。


FortiGate
admin
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
新建SSID
名称
Fortinet_Staff
别名
Fortinet_WiFi_Staff
类型
WiFi SSID
VRF ID
0
通信模式
隧道
桥接
全网状
WiFi Settings
SSID
Fortinet_WiFi_Staff
客户端限制
广播SSID
Beacon advertising
Name
Model
Serial number
安全模式设置
安全模式
WPA2-Only-Personal
预共享密钥
模式
单一
多个
密码
客户端MAC地址过滤
RADIUS 服务器
安全配置文件
附加设置
FortiAP独立本地认证
客户端本地认证
计划任务
always
阻止SSID内部流量
可选VLANID
20
广播压制
已知客户端ARP
DHCP unicast
DHCP上行
VLAN 池
NAC配置文件
其它条件
注释
0/255
状态
启用
禁用
确认
取消

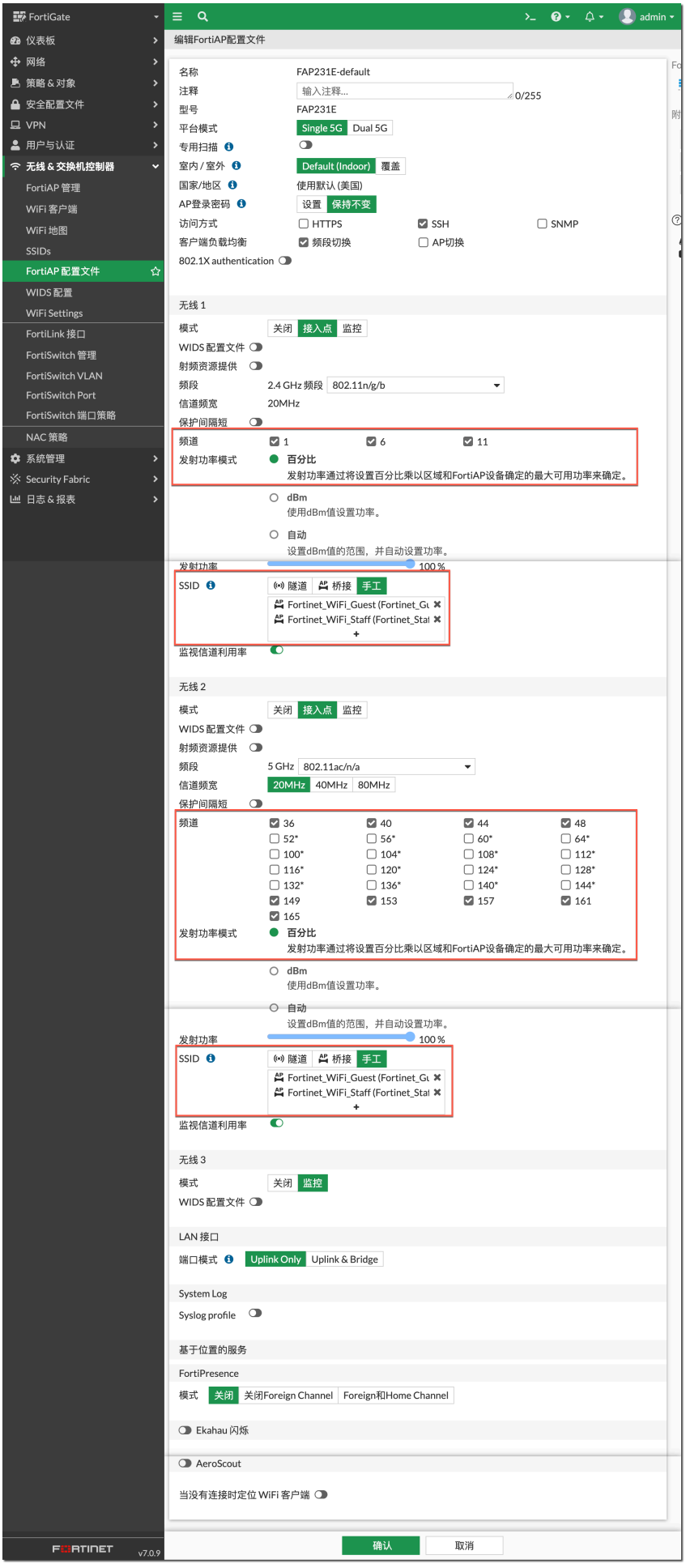




7. 将 SSID 关联到 FortiAP 上，然后让 FortiAP 发布无线 SSID 信号，这样用户就可以通过 SSID 连接到无线了。进入 FortiAP 配置文件页面，编辑 FortiAP 关联的配置文件。



8. 桥模式的 SSID 需要在 FortiAP 配置文件里面手工指定一下，另外下图标示出的配置为推荐配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
编辑FortiAP配置文件
名称 FAP231E-default
注释 输入注释... 0/255
型号 FAP231E
平台模式 Single 5G Dual 5G
专用扫描
室内 / 室外 Default (Indoor) 覆盖
国家/地区 使用默认 (美国)
AP登录密码 设置 保持不变
访问方式 HTTPS SSH SNMP
客户端负载均衡 频段切换 AP切换
802.1X authentication
无线 1
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 2.4 GHz 频段 802.11n/g/b
信道频宽 20MHz
保护间隔短
频道 1 6 11
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 2
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 5 GHz 802.11ac/n/a
信道频宽 20MHz 40MHz 80MHz
保护间隔短
频道 36 40 44 48
52* 56* 60* 64*
100* 104* 108* 112*
116* 120* 124* 128*
132* 136* 140* 144*
149 153 157 161
165
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 3
模式 关闭 监控
WIDS 配置文件
LAN 接口
端口模式 Uplink Only Uplink & Bridge
System Log
Syslog profile
基于位置的服务
FortiPresence
模式 关闭 关闭Foreign Channel Foreign和Home Channel
Ekahau 闪烁
AeroScout
当没有连接时定位 WiFi 客户端
确认
取消
FORTINET
v7.0.9

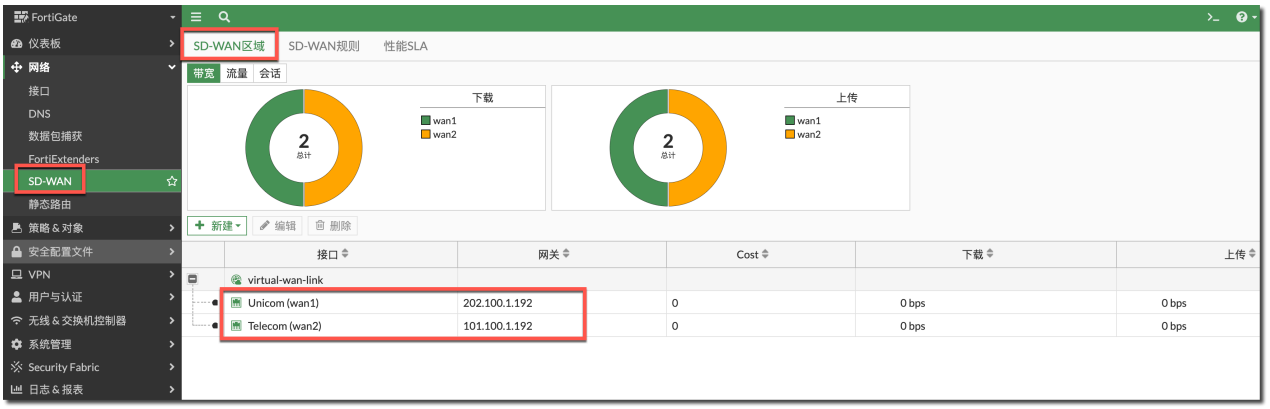


9. FortiAP 会自动关联到这个 FortiAP 配置文件里面的内容，发起 SSID 的信号。



SD-WAN 配置

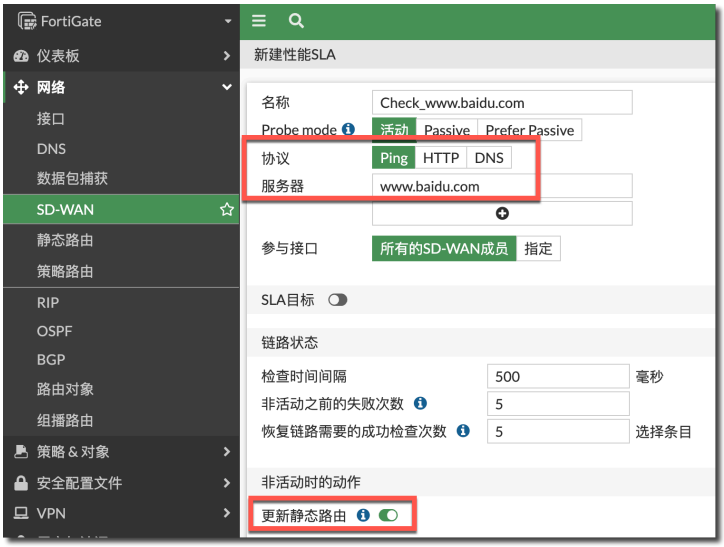
1. 将 wan1 和 wan2 加入 SD-WAN 接口，并配置各自的网关。

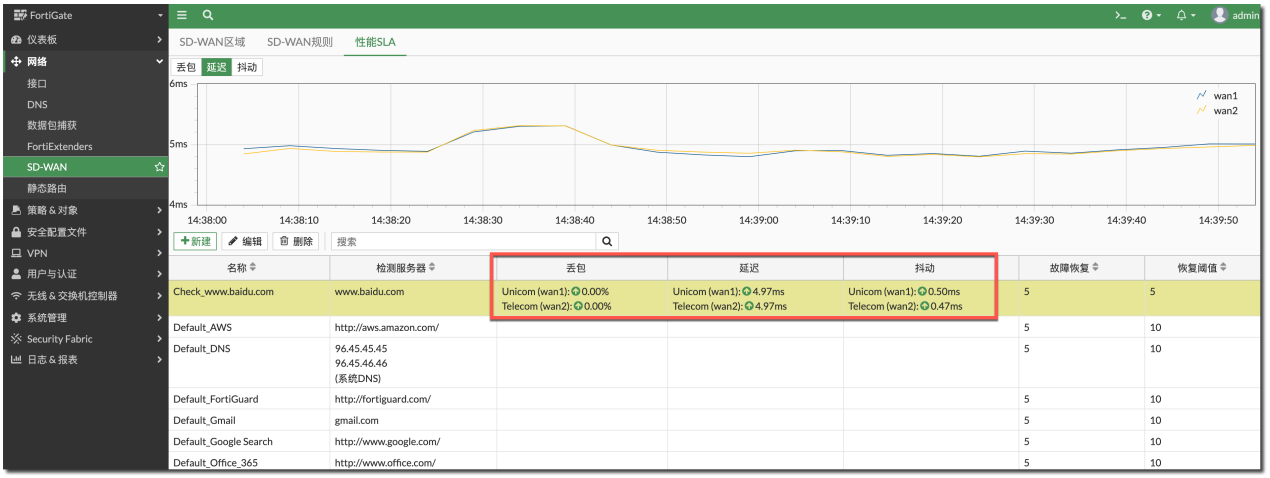


2. 添加 SD-WAN 的默认路由。



3. 配置 SD-WAN 的 SLA，以便 wan1 和 wan2 相互备份和切换。





4. SD-WAN 规则可根据需求配置，这里保持默认的按照源 IP 负载均衡。



安全策略配置

1. 需求与规划：

   - 员工的有线与无线都需要访问互联网
   - 访客的无线仅仅可以访问互联网
   - 员工的有线和无线需要互访

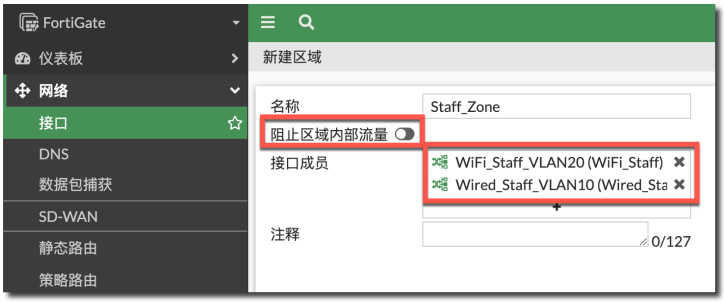
2. 建议将防火墙的接口划分区域，以区域的方式配置策略，有时候可以做到简化配置的目的，甚至未来替换设备的时候，策略的备份和还原也会更加简单一些。规划区域两个区域：

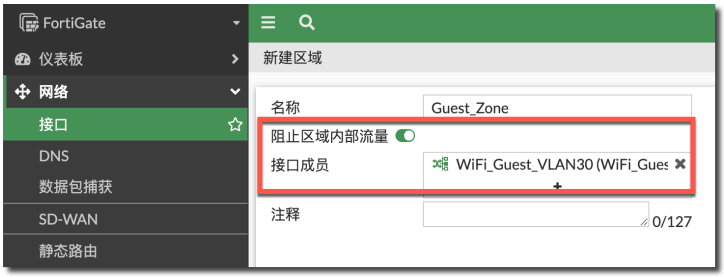
   - 员工网络区域：包括员工的有线 VLAN 10 和员工的无线 VLAN 20 （区域之间的隔离关闭，这样 VLAN 10 和 VLAN 20 之间的数据可通）
   - 访问网络区域：包括访问的无线 VLAN 30

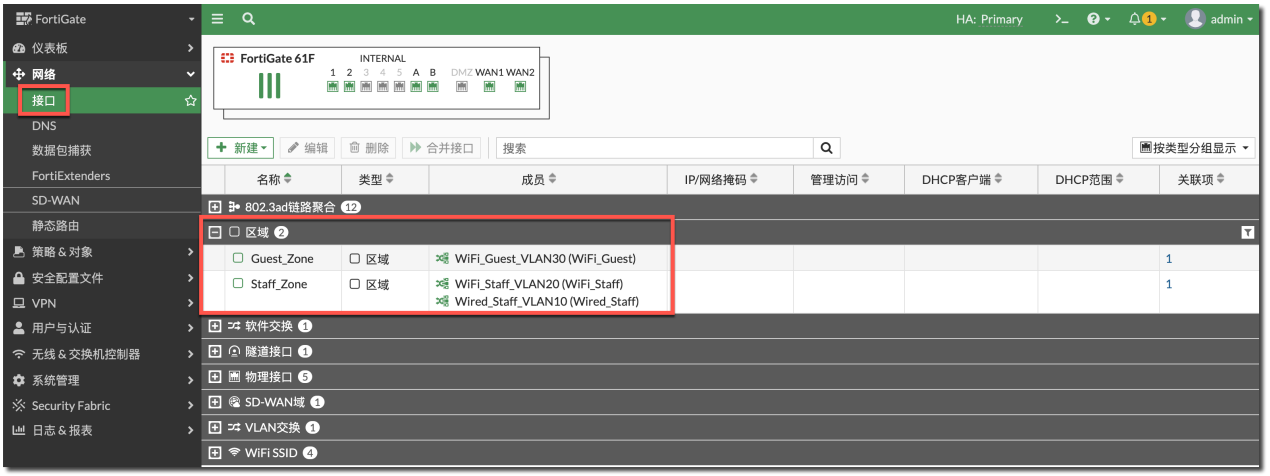
3. 进入网络 → 接口页面，新建所属区域。



4. 分别创建 Staff_Zone（包含 VLAN 10 和 VLAN20）和 Guest_Zone（包含 VLAN30）。







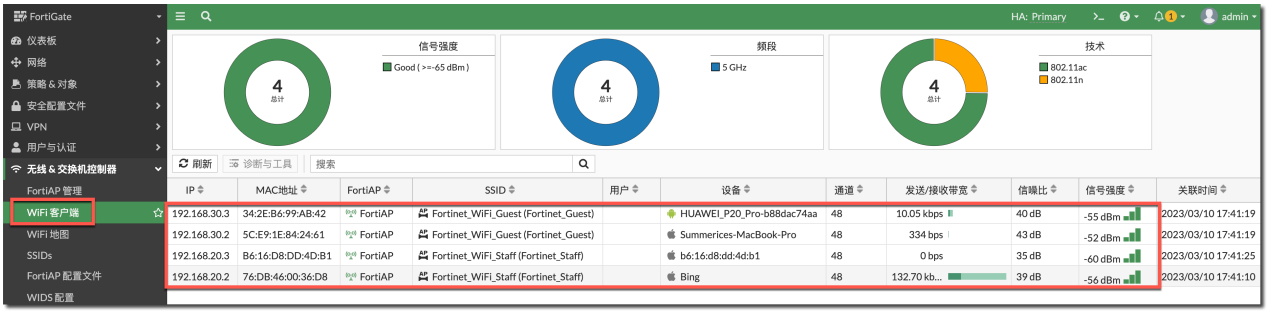
5. 如此只需要配置两条策略，就可以实现我们规划的需求，新建 Staff-Zone 到 SD-WAN 与 Guest-Zone 到 SD-WAN 的安全策略。



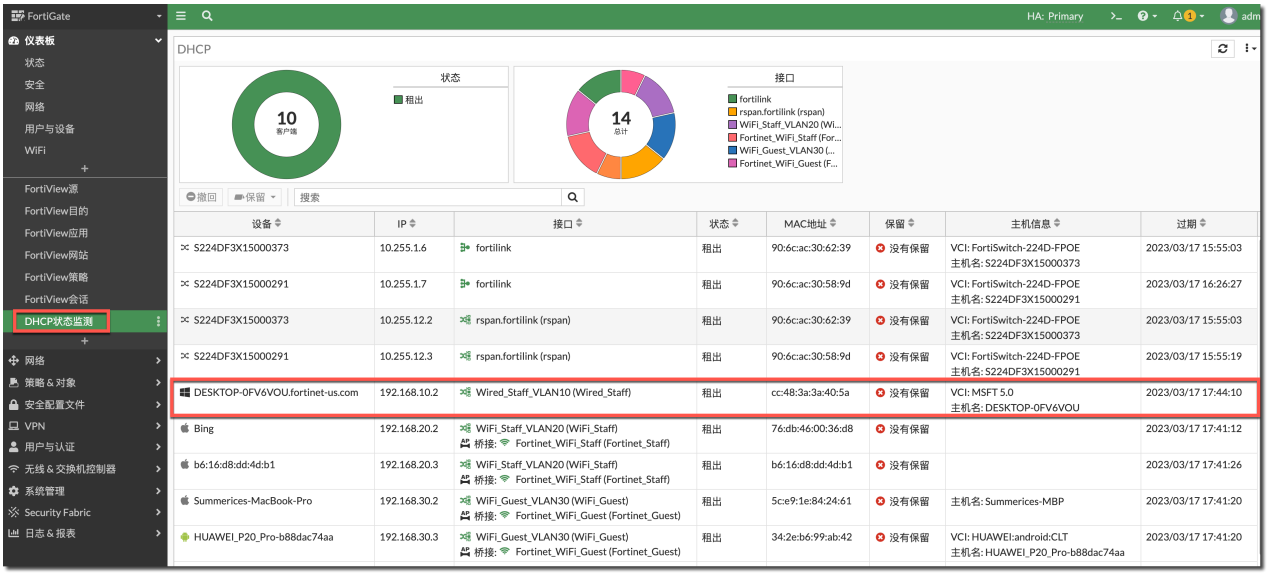
结果验证

客户端访问

1. 进入无线 & 交换机控制器 → WiFi 客户端，查看在线的无线客户端。可以看到 WiFi_Staff 和 WiFi_Guest 各有一个终端在线。



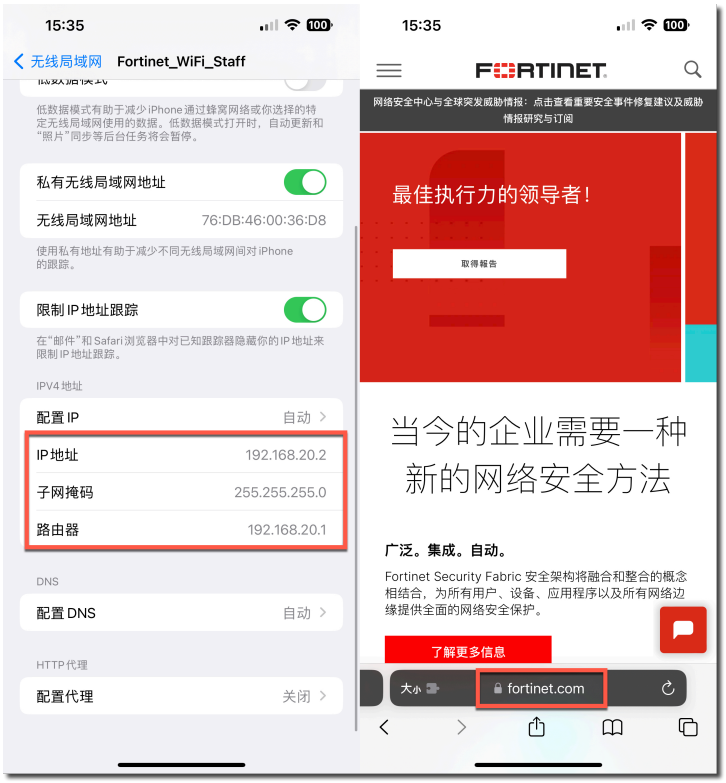
2. 在 DHCP 状态检测中查看所有的客户端信息，可以看到有线网络的 PC 也正常上线。



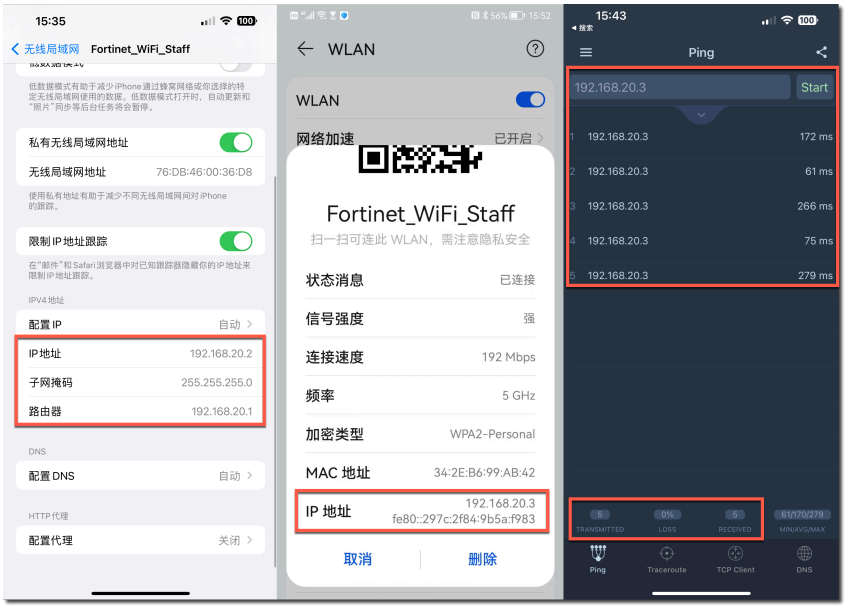
3. 有线网络（VLAN 10）用户访问互联网。



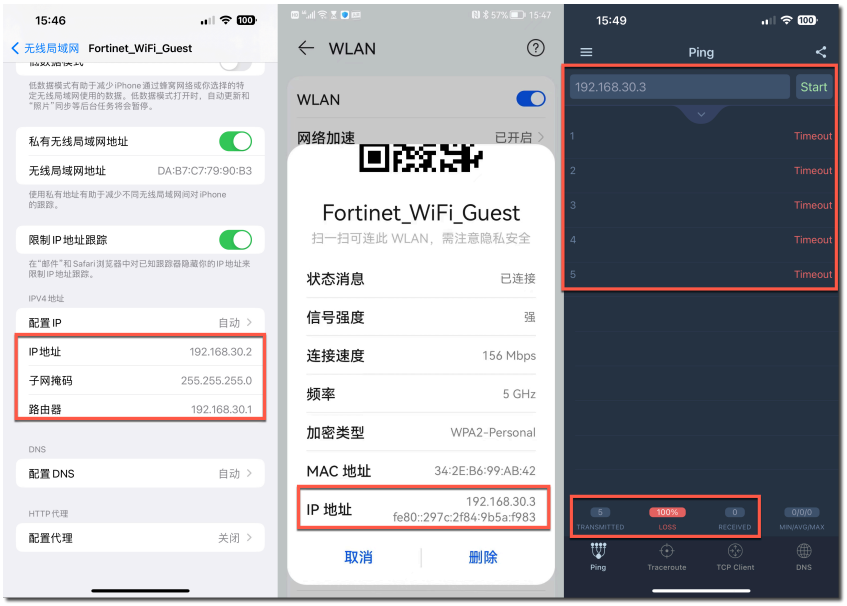
4. 无线网络用户访问互联网。



5. 使用 VLAN 20 下的客户端访问 VLAN 20 下的另一个客户端（Staff 区域）。



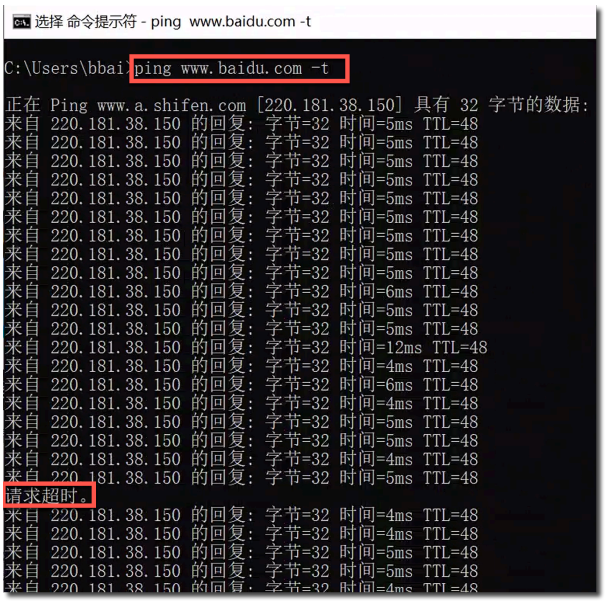
6. 使用 VLAN 30 下的客户端访问 VLAN 30 下的另一个客户端（Guest 区域），可以看到由于该区域配置了禁止相互访问，所以 VLAN 30 下的客户端是无法互相访问的。



FortiLink 切换测试

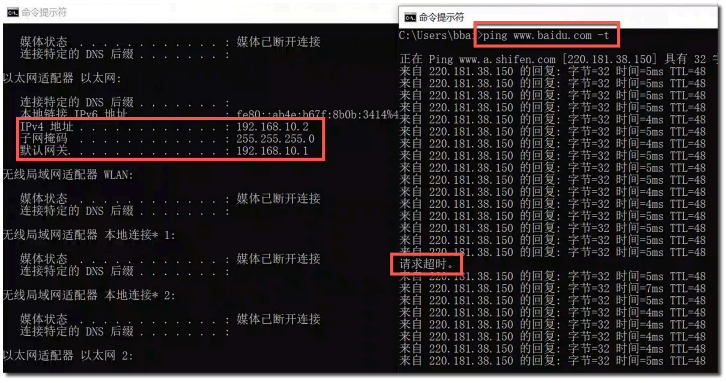
1. 在我们的测试场景下 FortiLink 聚合接口的 a 和 b 是同时 UP 起来的，如果 a 或 b DOWN（不能同时 DOWN），是否有影响业务呢?

2. 经过测试，DOWN a 或 b 接口，客户端丢包 1 个或者不丢包。



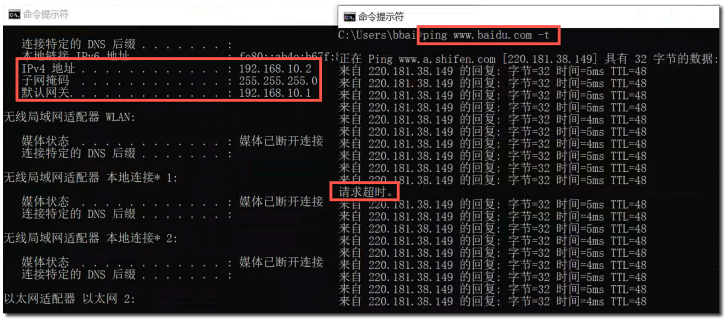
HA 切换测试

重启当前主防火墙查看内网 PC 丢包情况，HA 切换丢包 1 个。

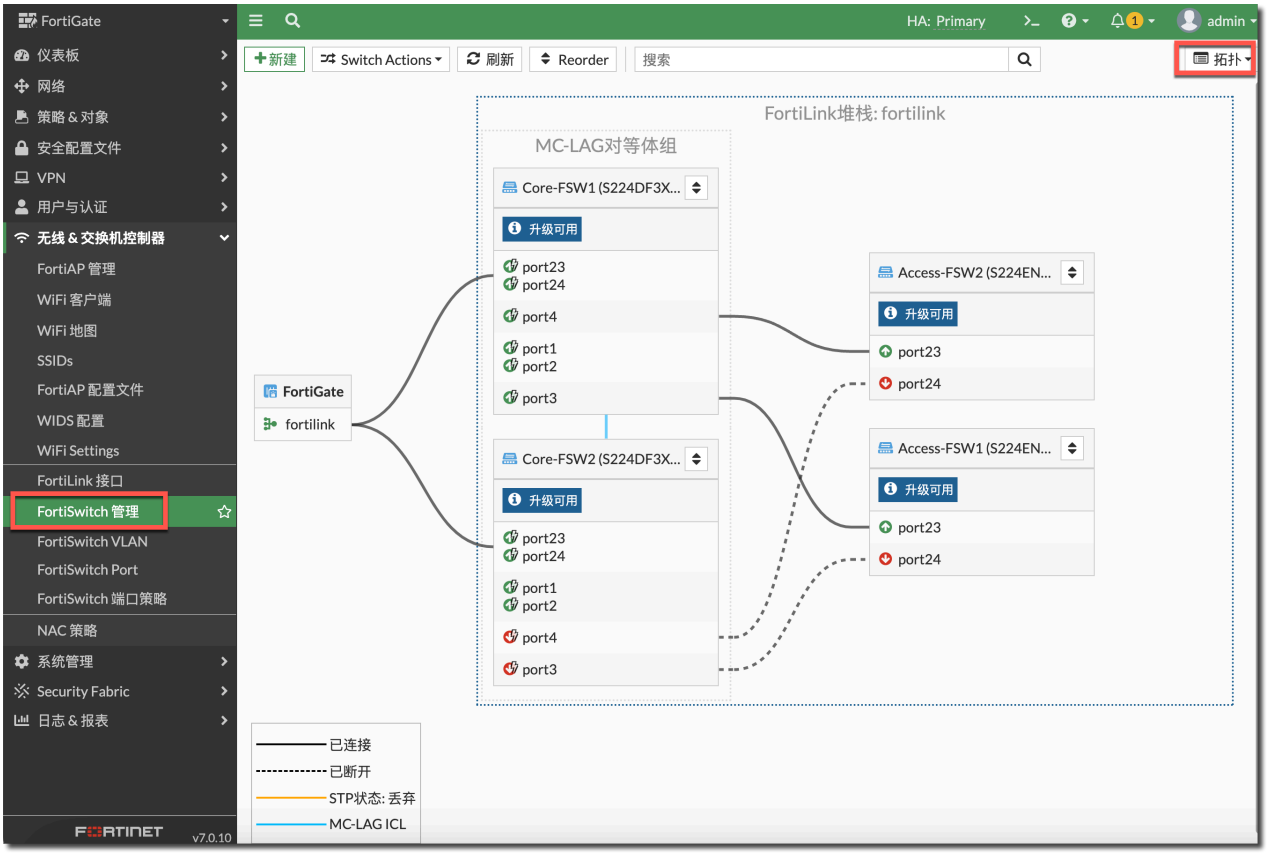


核心交换机接口故障模拟

1. Down 掉 Core-FSW2 的 Port3 和 Port4 接口，查看内网 PC 丢包情况，可以看到 Core-FSW2 连接接入层交换机的接口 Down 掉时，丢包 1 个。



2. 查看和 Switch 管理拓扑，Core-FSW2 与接入层交换机的连线已断开。



注意事项

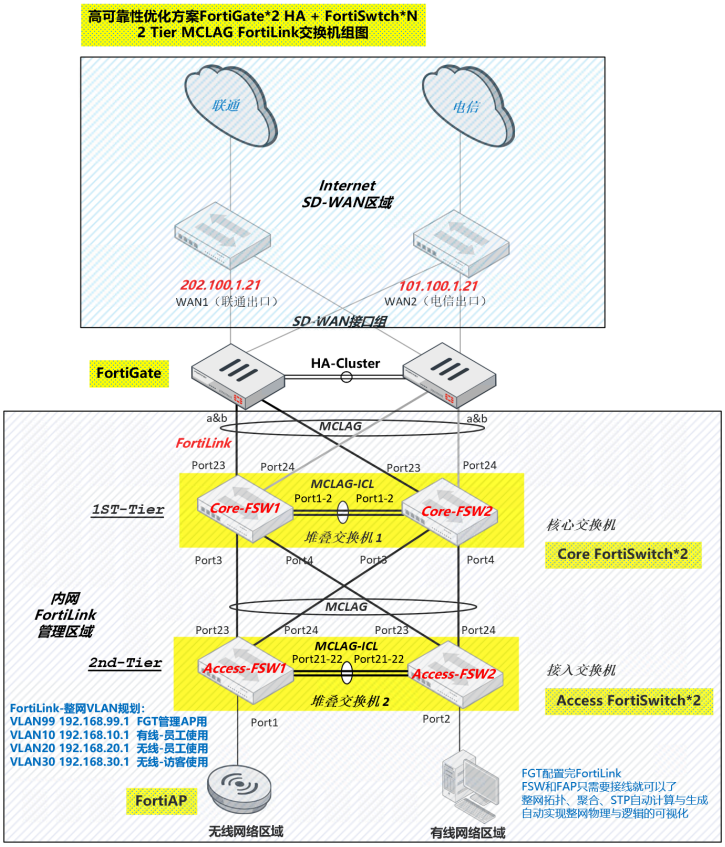
1. 只能是堆叠交换机才可以跨交换机聚合，不同的堆叠交换机组，不支持配置为聚合接口。
2. 如果是与 FortiSwith 对接，则是自动 MCLAG 聚合。
3. 如果是和第三方设备对接，比如交换机、路由器、友商防火墙、服务器等等对接的话，推荐使用 LACP Active 方式动态聚合。

FortiLink MCLAG - 堆叠组网 2 Tier 配置举例

网络需求

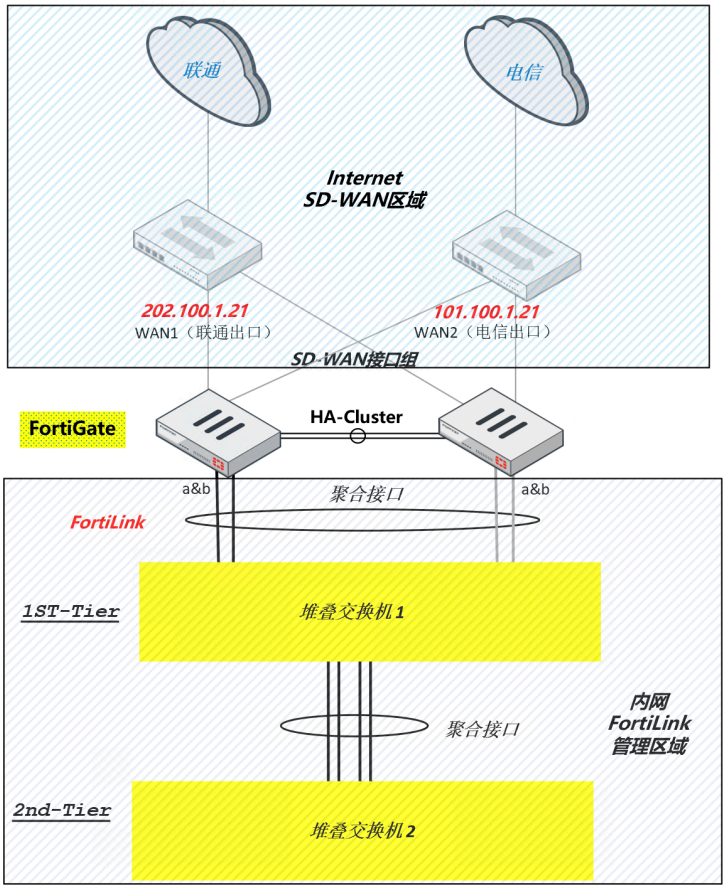
FortiGate 运行 HA-Cluster，核心交换和接入交换机采用 MCLAG 堆叠组网结构。

网络拓扑



形成堆叠之后的逻辑拓扑：

高可靠性优化方案FortiGate*2 HA + FortiSwtch*N
2 Tier MCLAG FortiLink交换机组图



1. 两台 FortiGate 配置为 HA 主备模式。
2. 整体规划：
   - 外网 IP 网段规划：
     - WAN1 联通出口：IP 202.100.1.21，GW 202.100.1.192
     - WAN2 电信出口：IP 101.100.1.21，GW 101.100.1.192
   - WAN1 和 WAN2 合并成 SD-WAN 接口。
   - 内网 IP 网段规划，FGT 通过 FortiLink + Capwap 协议统一的接管 FortiSW 和 FortiAP，为有线和无线统一进行管理和业务处理：
     - 有线网络规划：VLAN 10，192.168.10.1
     - 无线网络规划：员工 WIFI VLAN 20，192.168.20.1
     - 无线网络规划：访客 WIFI VLAN 30，192.168.30.1
     - 管理 FortiAP：VLAN 99，192.168.99.1
   - FortiLink 互联部分规划：
     - FortiGate 的 a 和 b 做聚合接口与 Fortiswitch 的后两个接口互联即可实现聚合接口的 FortiLink 管理。
     - 加入 FortiLink 的接口自动变为聚合口成员。

配置要点

1. FortiGate 配置聚合端口，并设置 split 为 enable 模式。
2. 上线 Core-FSW1 和 Core-FSW2，将 FortiGate 和 Core-FSW1 以及 Core-FSW2 连接起来。
3. 分别登录到 Core-FSW2 和 Core-FSW1 上，配置二者直连的 ISL 接口为“mclag-icl enable”。配置为 ICL 堆叠线路（Inter-Chassis-Link），形成两台核心交换机的堆叠。
4. 配置 FortiGate 聚合端口为 split disable 模式。
5. 上线 Access-FSW1 和 Access-FSW2。
6. 分别登录到 Access-FSW1 和 Access-FSW2 上，配置二者直连的 ISL 接口为“mclag-icl enable”。配置为 ICL 堆叠线路（Inter-Chassis-Link），形成两台接入交换机的堆叠。
7. 在 Core-FSW1 和 Core-FSW2 上分别配置 auto-isl-port-group，注意 group 的名字必须一致，端口可以不一样，这样是为了让核心的堆叠交换机与接入的堆叠交换机相互跨交换机手工形成聚合。
8. 上线 FortiAP 和有线 PC，并测试无线网络和有线网络的连通性。
9. 注意：如果 Switch 上线时，有的 Switch 是旧的版本，最好先升级 Switch 的版本后，再组建 mclag 或是让交换机成环。

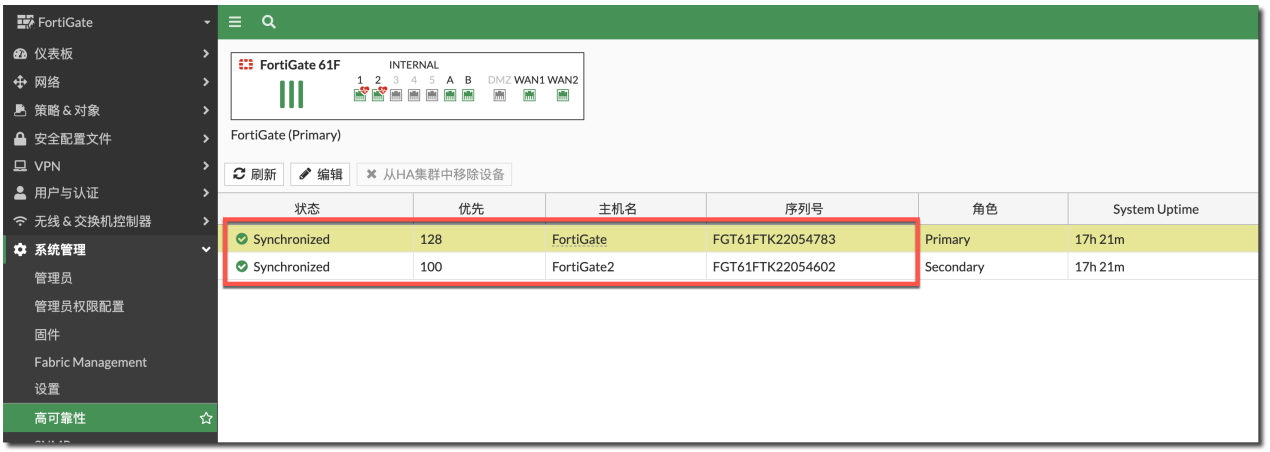
配置步骤

准备事项

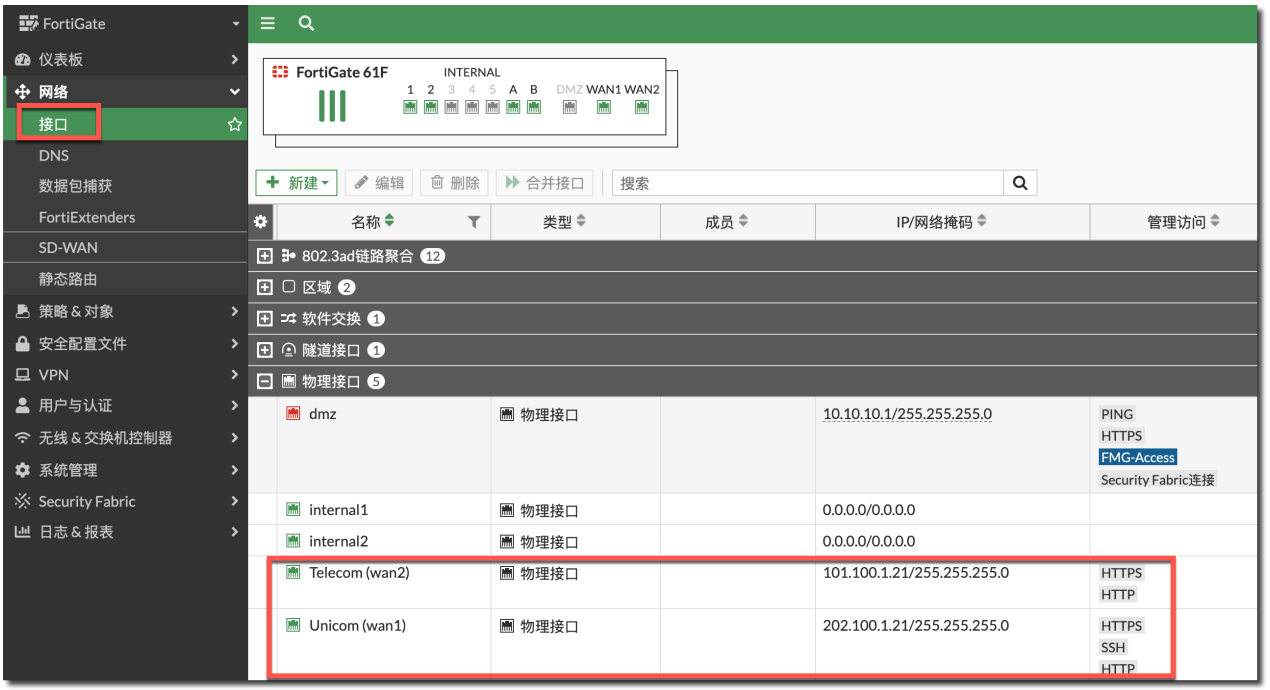
1. 首先确认 FortiGate 如下全局配置已开启（一般默认为开启状态）。

```
config system settings
    set gui-switch-controller enable
end
```

2. 配置两台 FortiGate 的 HA 主备模式（配置方法请参考 HA 双机热备 → HA 典型基础配置 ↗）。

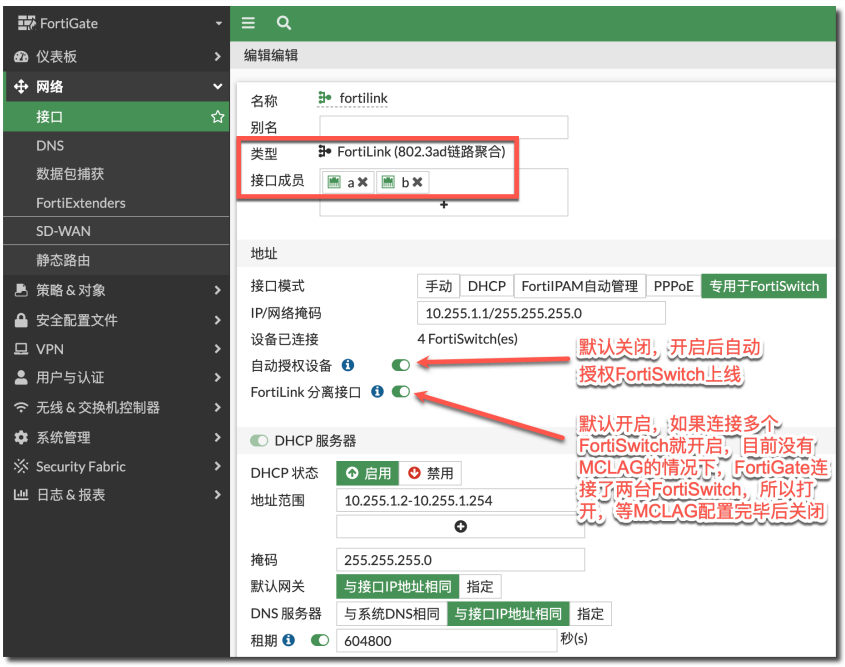


3. 配置 wan1 和 wan2 的 IP 地址。



4. 配置 FortiLink 实现 FortiSwith 上线，我们只需要将用于 FortiLink 的成员接口加入 FortiLink 接口即可。此拓扑中，FortiGate 用于连接 FortiSwitch 的接口为 a 和 b，编辑 FortiLink 接口，将 a 和 b 加入 FortiLink（某些产品型号会默认将两个接口加入 FortiLink），开启自动授权设备，关闭 FortiLink 分离接口，其他配置保持默认，可以看到 FortiLink 为一个聚合接口。


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 8
fortilink
802.3ad链路聚合
专用于 FortiSwitch
PING
Security Fabric连接
隧道接口 1
物理接口 5
a
物理接口
0.0.0.0/0.0.0.0
b
物理接口
0.0.0.0/0.0.0.0
dmz
物理接口
10.10.10.1/255.255.255.0
PING
HTTPS
FMG-Access
Security Fabric连接
Telecom (wan2)
物理接口
101.100.1.21/255.255.255.0
HTTPS
HTTP
Unicom (wan1)
物理接口
202.100.1.21/255.255.255.0
HTTPS
HTTP

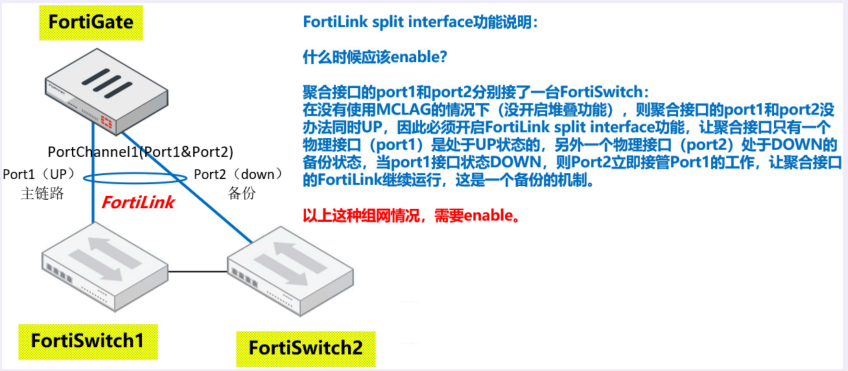

FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
编辑编辑
名称
fortilink
别名
类型
FortiLink (802.3ad链路聚合)
接口成员
a
b
地址
接口模式
手动
DHCP
FortiIPAM自动管理
PPPoE
专用于FortiSwitch
IP/网络掩码
10.255.1.1/255.255.255.0
设备已连接
4 FortiSwitch(es)
自动授权设备
FortiLink 分离接口
默认关闭，开启后自动授权FortiSwitch上线
DHCP 服务器
DHCP 状态
启用
禁用
地址范围
10.255.1.2-10.255.1.254
掩码
255.255.255.0
默认网关
与接口IP地址相同
指定
DNS 服务器
与系统DNS相同
与接口IP地址相同
指定
租期
604800
秒(s)
默认开启，如果连接多个FortiSwitch就开启，目前没有MCLAG的情况下，FortiGate连接了两台FortiSwitch，所以打开，等MCLAG配置完毕后关闭


FortiGate
仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略 & 对象
FortiGate 61F
INTERNAL
1 2 3 4 5 A B
DMZ WAN1 WAN2
新建
编辑
删除
合并接口
搜索
名称
类型
成员
IP/网络掩码
管理访问
802.3ad链路聚合 12
fortilink
802.3ad链路聚合
a
b
专用于 FortiSwitch
PING
Security Fabric连接

重要

FortiLink 分离接口什么时候应该开启?

分离接口的意是，如果聚合接口的 port1 和 port2 分别连接了一台 SW，在没有使用 MCLAG 的情况下（没开启堆叠功能），则聚合接口的 port1 和 port2 没办法同时 UP（同时开启聚合接口无法正常工作），因此必须开启 FortiLink 分离接口功能，让聚合接口只有一个物理接口（port1）是处于 UP 状态的，另外一个物理接口（port2）处于 DOWN 的备份状态，当 port1 接口状态 DOWN，则 Port2 立即接管 Port1 的工作，让聚合接口的 FortiLink 继续运行，这是一个聚合接口在 FortiLink 管理时候的备份机制。



MCLAG 配置前是上述组网，因此刚刚开始的时候需要，保持 enable 状态。聚合接口的 Port1 和 Port2 只会 UP 一个接口，另外一个接口处于 DOWN 的状态。

重要

FortiLink split interface 什么时候应该 disable?

如果 FGT 的聚合接口只接到了一台交换机上，并没有跨交换机聚合（没有 MCLAG，也不需要 FortiLink split interface），因此这种情况应该让聚合接口的两个物理接口 port1 和 port2 同时 UP，两个接口同时工作起来，以便提供更高带宽和备份的作用。



配置两台交换机 MCLAG 堆叠之后，就是上述组网，因此交换机堆叠完毕之后，需要修改为 disable，让两个聚合接口都可以同时工作起来，最大的利用聚合接口的带宽和资源。这个配置后续会说明。

相关 CLI:

```
NTP:
config system ntp
    set ntpsync enable
    set server-mode enable
    set interface "fortilink"
end
------------------------------------------------------------------
DHCP:
config system dhcp server
    edit 2
        set dns-service local
        set ntp-service local
        set default-gateway 10.255.1.1
        set netmask 255.255.255.0
        set interface "fortilink"
        config ip-range
            edit 1
                set start-ip 10.255.1.2
```

```
                set end-ip 10.255.1.254
            next
        end
        set vci-match enable
        set vci-string "FortiSwitch" "FortiExtender"
    next
end
```

```
FortiLink:
config system interface
    edit "fortilink"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set type aggregate
        set member "a" "b"
        set lldp-reception enable
        set lldp-transmission enable
        set auto-auth-extension-device enable
        set fortilink-split-interface enable
        set switch-controller-nac "fortilink"
        set switch-controller-dynamic "fortilink"
        set swc-first-create 255
    next
end
```

连接 FortiLink

连接核心交换机

1. 配置完 FortiGate 之后，这个时候再去接 FortiGate 的 FortiLink 到 FortiSwitch 的线，配置 Core-FSW1 和 Core-FSW2 设备的互联接口 MCLAG 注意，先不要把所有的线都接上去，一台一台的来操作，MCLAG 配置尽量使用 Console 去配置，而不是 GUI，有时候由于在 Fortilink 计算和收敛会导致 GUI 连接不上去，或者先将 CORE - FSW1 下线，先单独配置 CORE - FSW2，然后再将 CORE - FSW2 下线，再单独配置 CORE - FSW1，然后再将二者互联的线接上去，这样是最稳妥的办法，避免 GUI 的 CLI 连不上的情况出现。而接入交换机这个时候是不建议接入的，等 CORE - FSW1 和 CORE - FSW2 的堆叠形成了之后，再将接入层交换机的线。

   > **重要**
   >
   > FortiSwitch 如果以前有使用过，建议都恢复一下出厂设置，然后再加入到 FortiLink。

2. FortiSwitch 的靠后接口一般都默认开启了 FortiLink 接口属性，如果接前面的接口可能会无法加入到 FortiLink，因此 FortiSwitch 的 FortiLink 最好都使用靠后的接口进行接线。FortiSwitch 各型号默认开启 FortiLink 功能的接口如下：

FortiSwitch 型号	默认开启 FortiLink 的接口
FS-108D-POE	port9–port10
FS-108E, FS-108E-POE, FS-108E-FPOE	port7–port10
FSR-112D-POE	port5–port12
FS-124D, FS-124D-POE	port23–port26
FSR-124D	port1-port4, port21–port28
FS-124E, FS-124E-POE, FS-124E-FPOE	port21–port28
FS-148E, FS-148E-POE	port21–port52
FS-224D-POE	port21–port24
FS-224D-FPOE	port21–port28
FS-224E, FS-224E-POE	port21–port28
FS-248D, FS-248D-FPOE	port45–port52
FS-248D-POE	port47–port50
FS-248E-POE, FS-248E-FPOE	port45–port52
FS-424D, FS-424D-POE, FS-424D-FPOE	port23–port26
FS-424E-Fiber	port1-port30
FS-426E-FPOE-MG	port23-port30
FS-448D, FS-448D-POE, FS-448D-FPOE	port45–port52
FS-524D, FS-524D-FPOE	port21–port30
FS-548D	port39–port54
FS-548D-FPOE, FS-548DN	port45–port54
FS-1024D	port1–port24
FS-1048D, FS-1048E	port1–port52
FS-3032D, FS-3032E	port1–port32

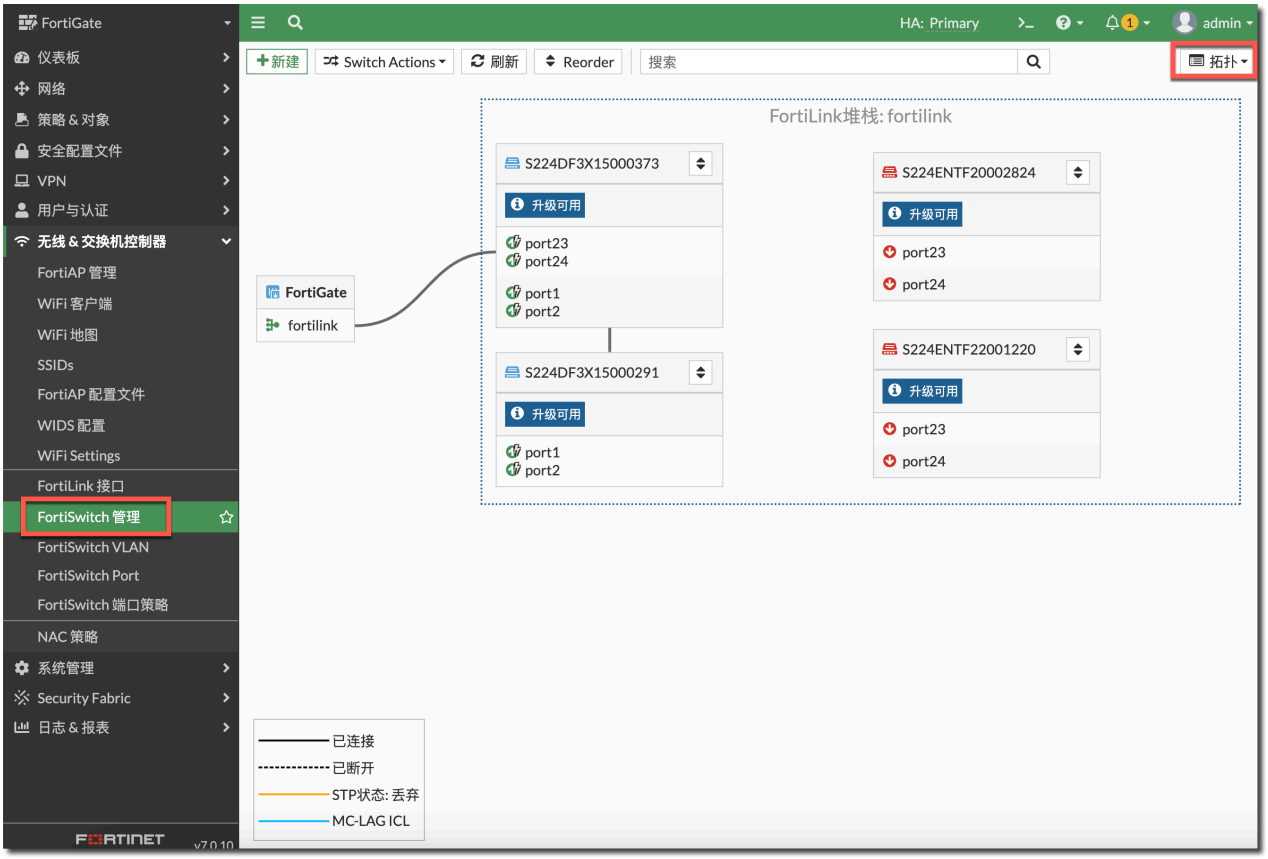
3. 如果 FortiSwitch 无法被 FortiGate 管理，登陆到 FortiSwitch 上进行确认接口是否开启了 FortiLink。

```
FortiSwitch # show switch interface port9
config switch interface
    edit "port9
        set auto-discovery-fortilink enable    //接口下又此命令的接口才可以自动加入到FortiLink中
    next
end
```

4. 交换机之间如果接多根线，会自动进行聚合，但是 FortiLink 自动聚合会花比较长的时间，建议可以先接单线，等 FortiLink 的拓扑较快的形成结束之后，再增加彼此之间的另外一根线，再进行自动聚合，这样效率会比较高一些。

5. 接线是很重要的步骤，仔细看拓扑图，做好接口标签，不要接错。

6. 先接 Core-FSW1 的线，再接 Core-FSW1 和 Core-FSW2 互联的线，先不要接接入层交换机的线，在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，右上角选择拓扑视图，查看交换机被管理状态。

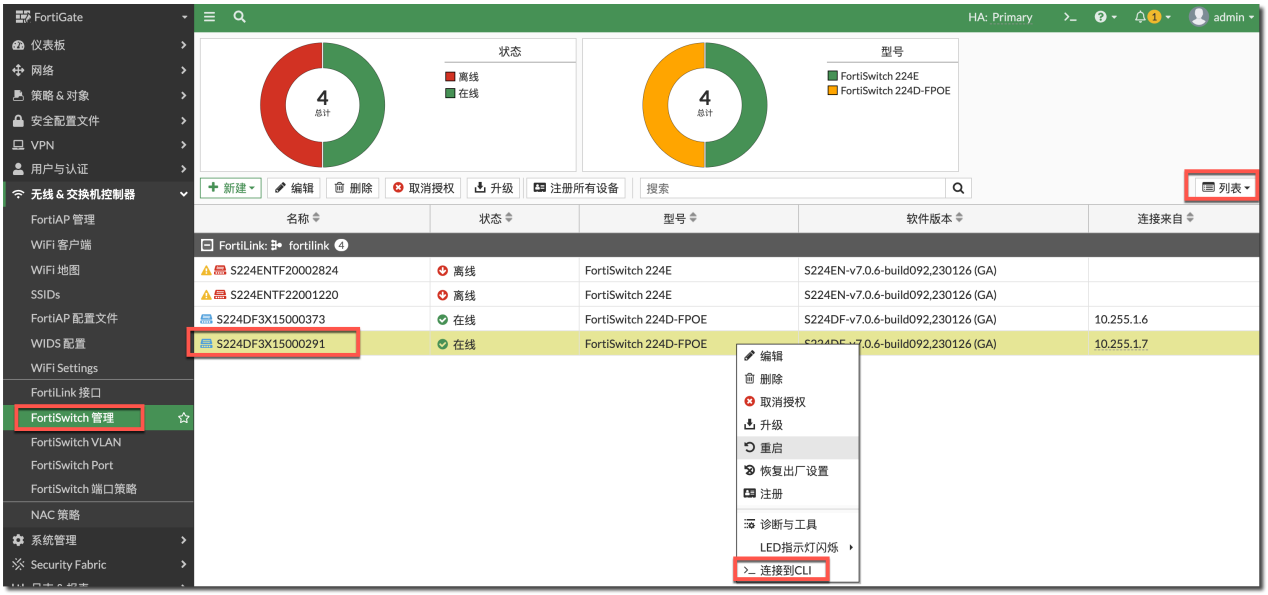


7. 此时我们看防火墙的 FortiLink 聚合接口，此时应该只有 a 或 b 其中一个物理接口是 UP 的，这是 FortiLink split interface enable 的结果。

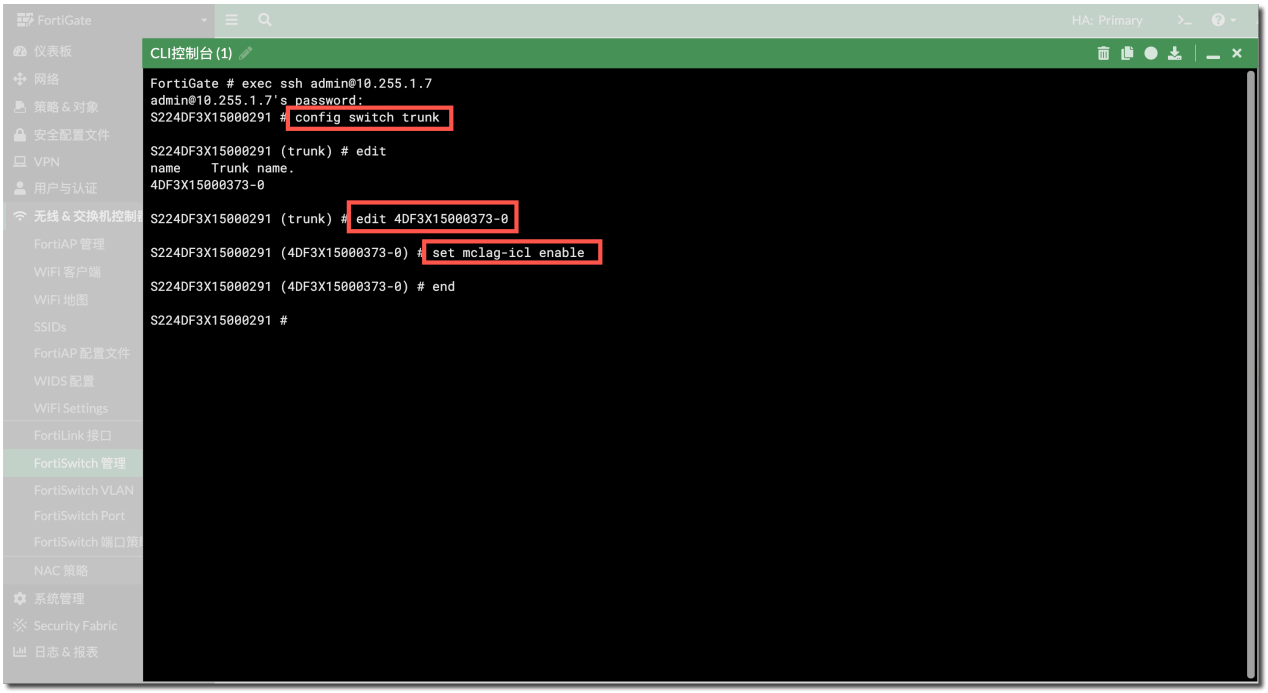


配置核心交换机 MC - LAG

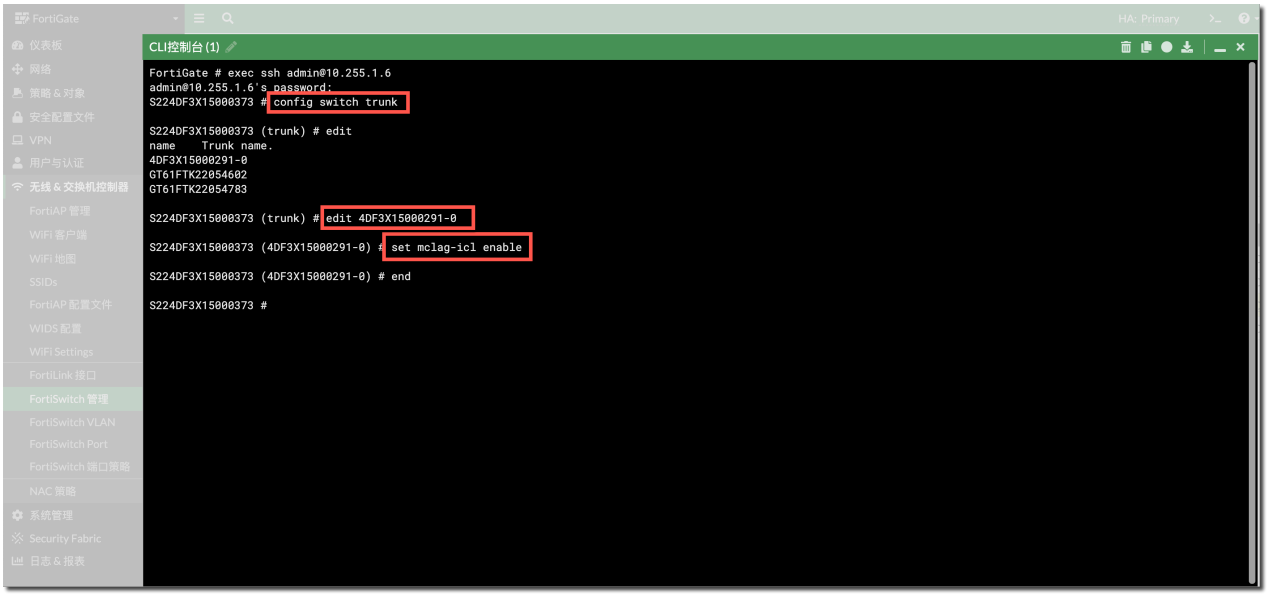
1. 通过 FortiGate 管理交换机页面右键交换机即可 SSH 到交换机，分别登录到 Core-FSW1 和 Core-FSW2 上。



2. 配置二者的互联聚合接口为“set mclag-icl enable”（请注意此例中在 GUI 配置时一定要先配置 CORE - FSW2，因为 CORE - FSW2 与 FortiGate 的 FortiLink 接口处于 down 的状态，如果先配置了 CORE - FSW1，那么 CORE - FSW2 将无法通过 GUI 管理）。

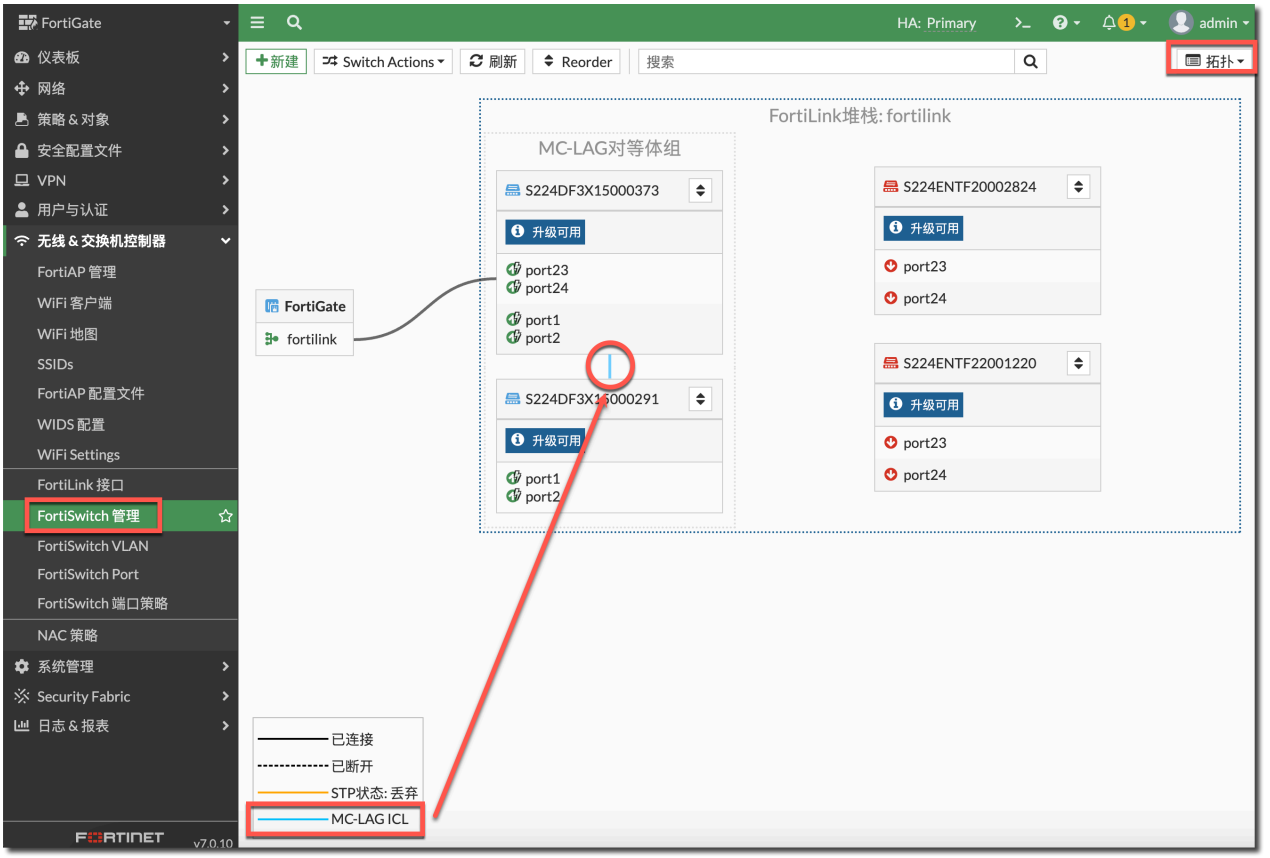



```
config switch trunk
    edit "4DF3X15000373-0"
        set mclag-icl enable
    next
end
```

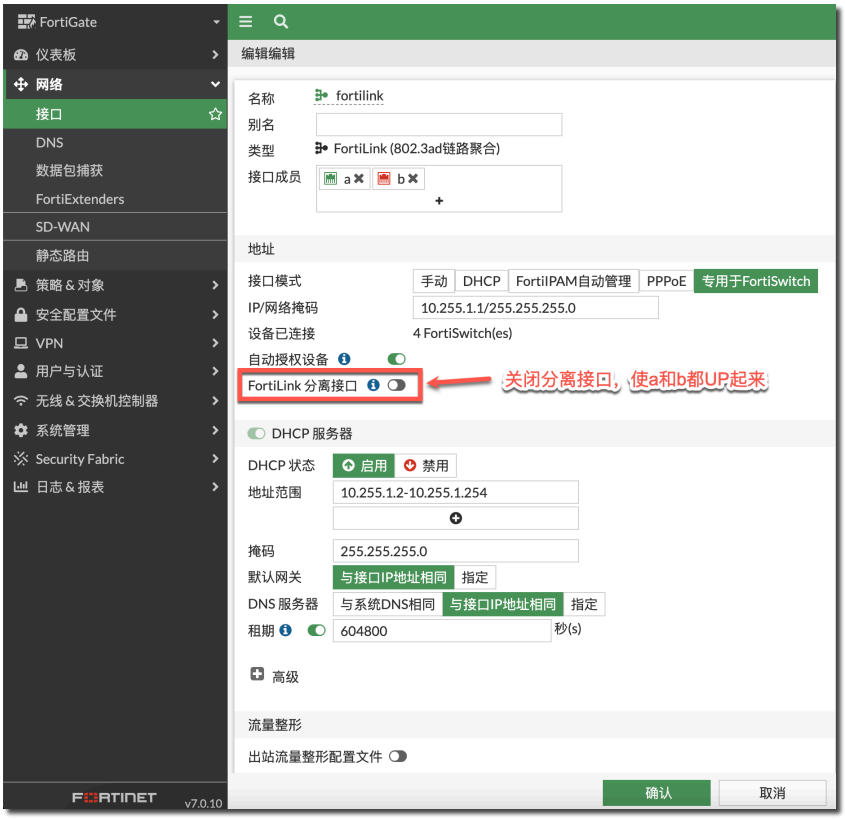



```
config switch trunk
    edit "4DF3X15000291-0"
        set mclag-icl enable
    next
end
```

3. 查看配置 MCLAG 后的交换机拓扑，可以看到两台交换机已经变为一个 MC - LAG 组。

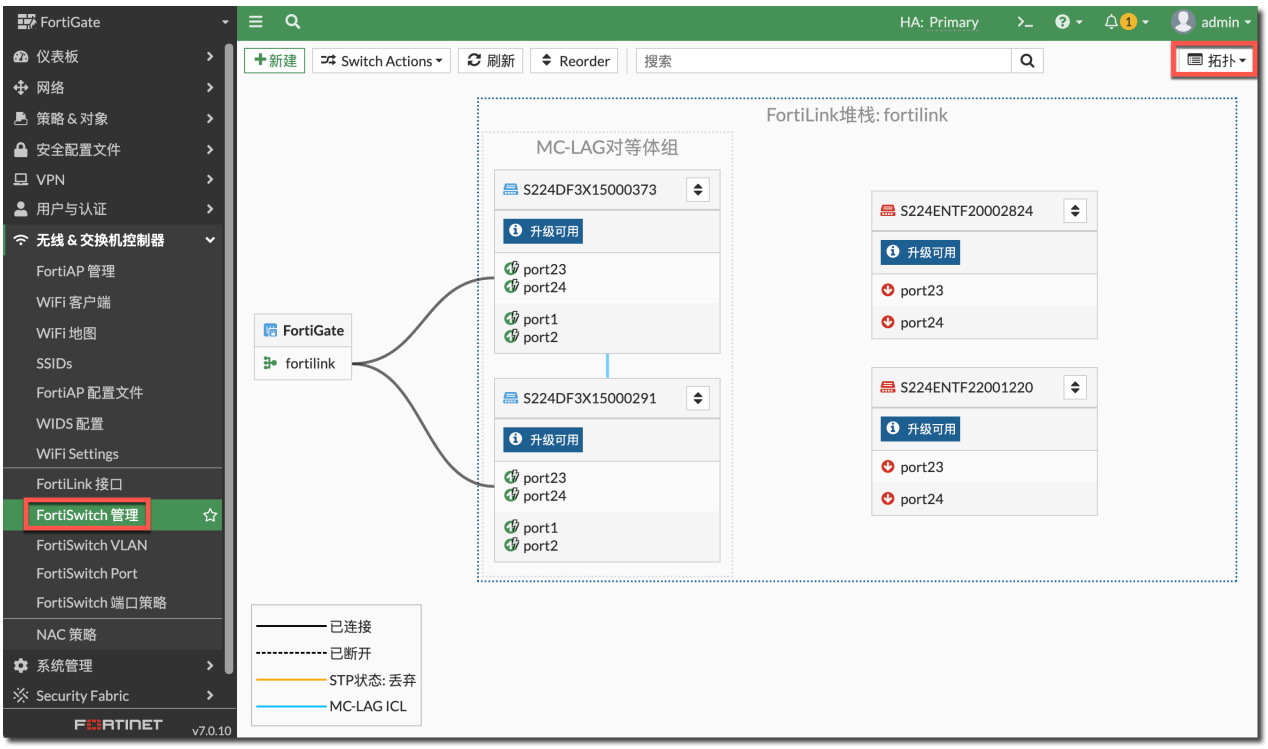


4. MC - LAG 配置完成后，将 FortiLink 接口下的 split interface 设置为 disable，让聚合接口（a 和 b）全部 UP 起来，充分利用聚合接口的资源，实现 FortiGate 与两台交换机（堆叠）的跨交换机聚合。




```
config system interface
    edit "fortilink"
        set fortilink-split-interface disable
    next
end
```

5. 等待 CORE - FSW2 重新在 FortiGate 上线后，查看 Switch 管理拓扑，FortiLink 的两个成员接口均已 UP。

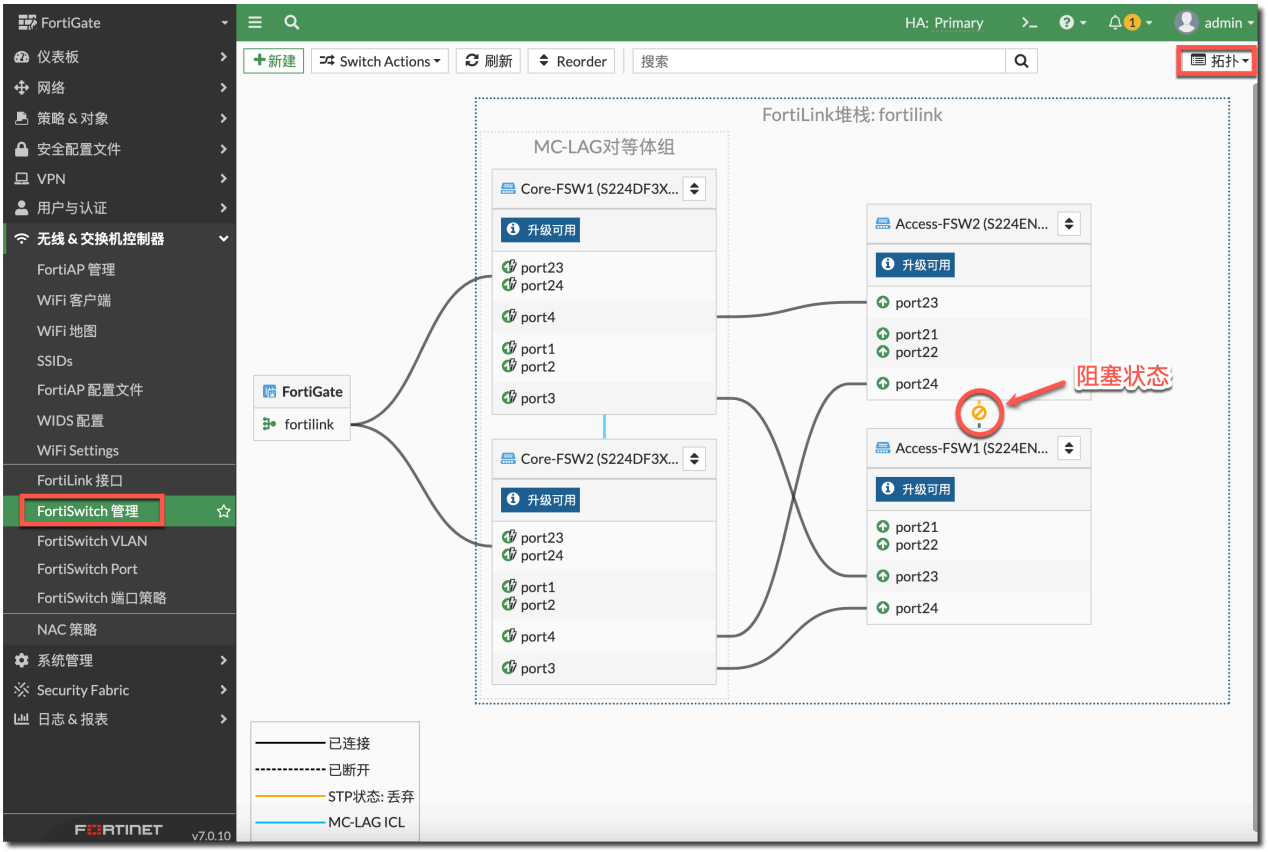


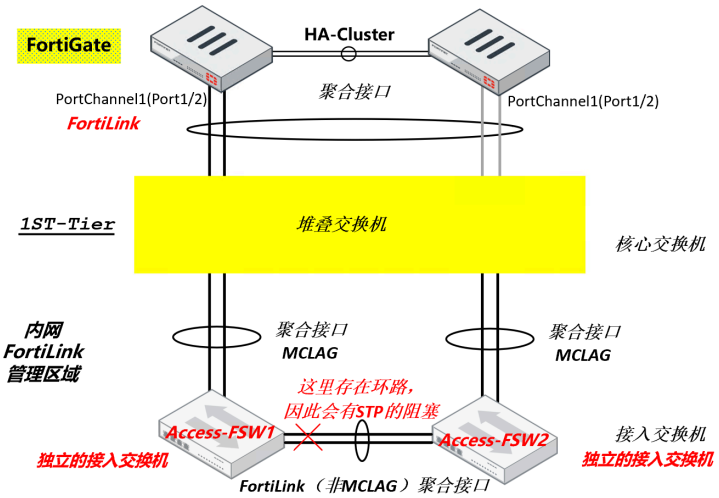
6. 在 FortiGate 的 CLI 下查看 FortiLink 成员状态，可以看到与实际物理连线一致。

```
FortiGate # execute switch-controller get-physical-conn standard fortilink

This will display connectivity graph information for FortiLink from FortiGate's perspective
NOTE : If FortiSwitch is not authorized, no connectivity information will be shown
NOTE : If FortiSwitch is in idle state, no connectivity information will be shown
NOTE : If FortiSwitch ISL peer has inconsistent info, no connectivity information will be shown
        FortiLink interface : fortilink
FortiGate(s)
FGT61FTK22054783(a)  «————————————»  S224DF3X15000373(port23)
FGT61FTK22054602(a)  «————————————»  S224DF3X15000373(port24)
FGT61FTK22054783(b)  «————————————»  S224DF3X15000291(port23)
FGT61FTK22054602(b)  «————————————»  S224DF3X15000291(port24)

Tier 1
S224DF3X15000373(port23)  «————————————»  FGT61FTK22054783(a)
S224DF3X15000373(port24)  «————————————»  FGT61FTK22054602(a)
S224DF3X15000291(port23)  «————————————»  FGT61FTK22054783(b)
S224DF3X15000291(port24)  «————————————»  FGT61FTK22054602(b)

Tier 2+
S224DF3X15000373(port1/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port1/4DF3X15000373-0)
S224DF3X15000373(port2/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port2/4DF3X15000373-0)
```

连接接入交换机

1. 将接入层两台交换机的连线接入。在 FortiGate 上进入无线 & 交换机控制器 → FortiSwitch 管理页面，查看交换机被管理状态，修改视图为拓扑，可以看到 4 台交换机均已上线，两台交换机之间的接口为丢弃状态。





2. 查看两台接入交换机的 STP 状态，Access-FSW1 的接口均为转发状态，Access-FSW2 与 Access-FSW1 的互联的接口为阻塞状态。

```
FortiGate # exec ssh admin@10.255.1.3
admin@10.255.1.3's password:

Access-FSW1 # diagnose stp instance list 15

MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC ac712e465448
  Regional Root  MAC 906cac30589c, Priority 20480, Path Cost 1, Root Port _FlInK1_MLAG0_
  TCN Events     Triggered 30 (0d 0h 21m 37s ago), Received 76 (0d 0h 21m 31s ago)
  Port               Speed   Cost       Priority   Role         State        Flags
  ________________   _____   ________   ________   ___________  ___________  _____________

  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
```

```
  _FlInK1_MLAG0_     2G      1          128        ROOT         FORWARDING   EN
  4ENTF20002824-0    2G      1          128        DESIGNATED   FORWARDING   EN
  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

```
FortiGate # exec ssh admin@10.255.1.4
admin@10.255.1.4's password:

Access-FSW2 # diagnose stp instance list 15

MST Instance Information, primary-Channel:
Instance ID 15
  Config         Priority 28672, VLANs 4094
                 Bridge MAC e023ff54e7e0
  Regional Root  MAC 906cac30589c, Priority 20480, Path Cost 1, Root Port _FlInK1_MLAG0_
  TCN Events     Triggered 32 (0d 2h 48m 38s ago), Received 99 (0d 2h 48m 33s ago)

  Port               Speed   Cost       Priority   Role         State        Flags
  _________________  ______  _________  _________  ___________  ___________  _________________
  internal           1G      20000      128        DESIGNATED   FORWARDING   ED
  _FlInK1_MLAG0_     2G      1          128        ROOT         FORWARDING   EN
  4ENTF22001220-0    2G      1          128        ALTERNATIVE  DISCARDING   EN

  Flags: EN(STP enable), ED(Edge), LP(Loop Protection Triggered)
  RG(Root Guard Triggered), BG(BPDU Guard Triggered), IC(PVST Port Inconsistent)
  MV(PVST Port Vlan Mismatch)
```

3. 为了防止出现环路导致 FortiGate 无法管理 FortiSwitch，通过 Console 登录并配置两台接入交换机的互联聚合接口为“set mclag-icl enable”（或先将接入交换机之间互联口 Down 掉，通过 FortiGate SSH 到交换机配置）。

```
Access-FSW1:
config switch trunk
    edit "4ENTF20002824-0"
        set mclag-icl enable
    next
end

Access-FSW2:
config switch trunk
    edit "4ENTF22001220-0"
        set mclag-icl enable
    next
end
```

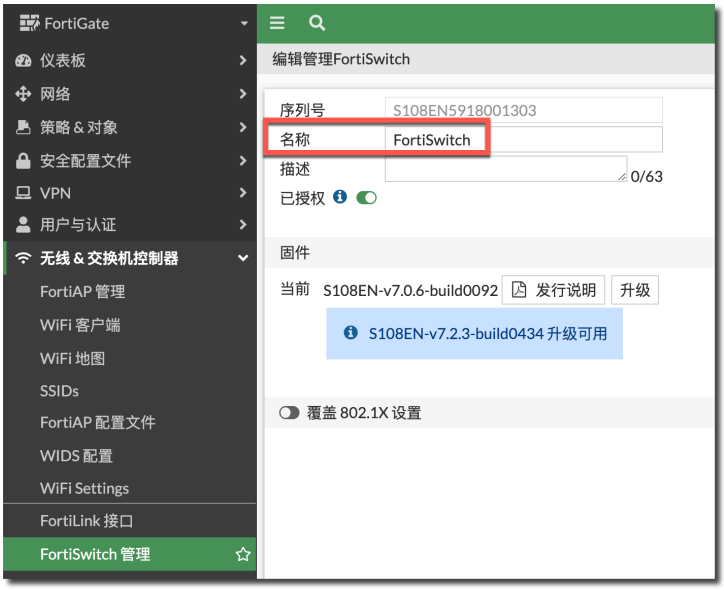
4. 在 Core-FSW1 和 Core-FSW2 上分别配置 auto-isl-port-group，注意 group 的名字必须一致，端口成员为与接入交换机互联的接口，这样是为了让核心的堆叠交换机与接入的堆叠交换机相互跨交换机手工形成聚合。

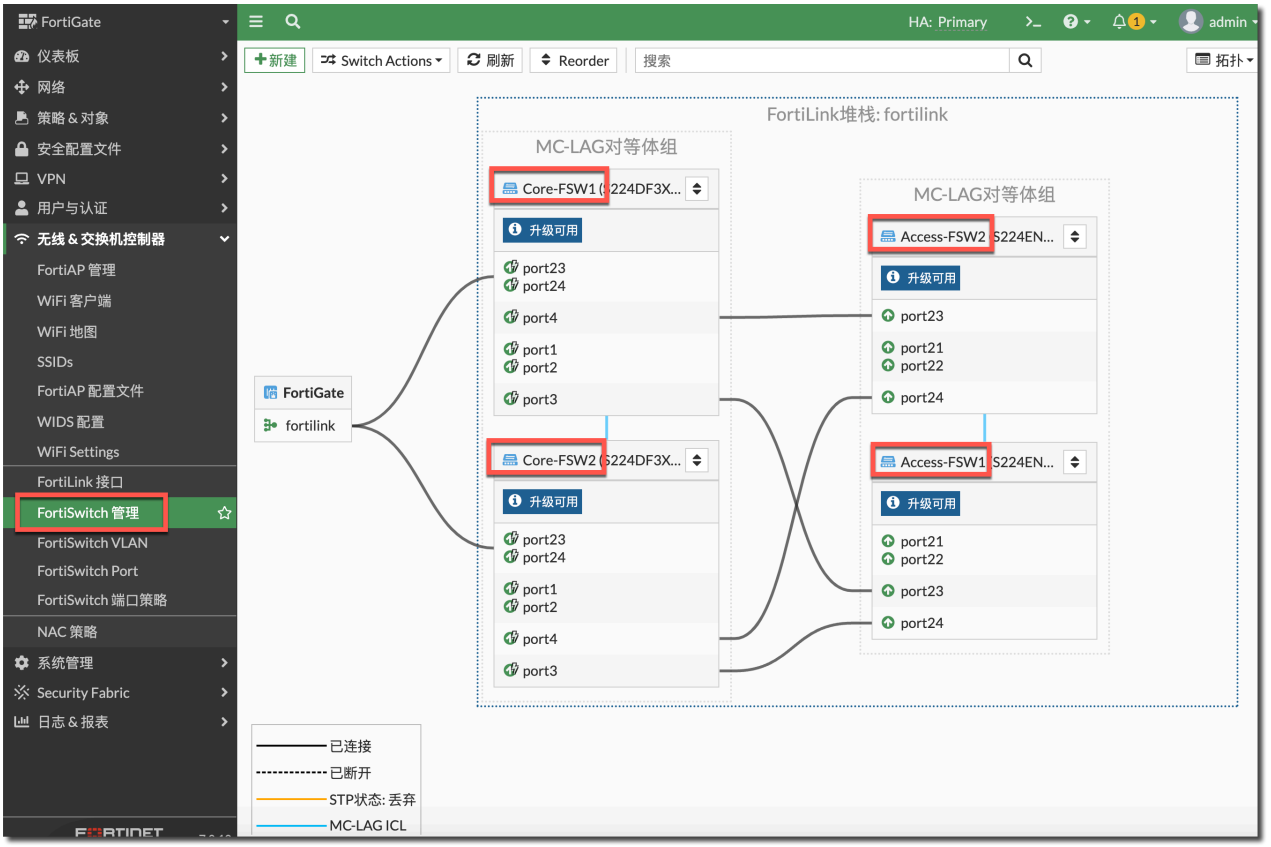
```
Core-FSW1:
config switch auto-isl-port-group
    edit "mclag-tier2"    //手工配置auto-isl-port-group，Core-FSW1和Core-FSW2的名字需要一样，都为mclag-tier2
        set members "port3" "port4"
    next
end

Core-FSW2:
config switch auto-isl-port-group
    edit "mclag-tier2"    //手工配置auto-isl-port-group，Core-FSW1和Core-FSW2的名字需要一样，都为mclag-tier2
        set members "port3" "port4"
    next
end
```

5. 配置完毕之后 Core-FSW1 和 Core-FSW2 后会自动生成聚合接口，无需手工配置，此配置可以理解为手工创建了跨堆叠交换机的聚合接口：“mclag-tier2”包括四个物理接口，CORE - FSW1 上的两个物理接口 Port3 和 Port4 以及 CORE - FSW2 上的两个物理接口 Port3 和 Port4，一共四个物理接口，共同属于聚合接口：“mclag-tier2”。

```
Core-FSW1 # show switch trunk
config switch trunk
...
    edit "mclag-tier2"
        set mode lacp-active
        set auto-isl 1
        set mclag enable
        set members "port4" "port3"    //对于堆叠交换机而言，名字一样的聚合接口即为跨交换机聚合的聚合接口，这是"mclag-tier2"聚合接口
在CORE-FSW1上的两个物理接口Port3和Port4
    next
end

Core-FSW2 # show switch trunk
config switch trunk
...
    edit "mclag-tier2"
        set mode lacp-active
        set auto-isl 1
        set mclag enable
        set members "port4" "port3"    //对于堆叠交换机而言，名字一样的聚合接口即为跨交换机聚合的聚合接口，这是"mclag-tier2"聚合接口
在CORE-FSW1上的两个物理接口Port3和Port4
    next
end
```

6. 接好线之后，FortiSwitch 和 FortiGate 之间会自动在后台进行 FortiLink、CAPWAP、MSTP、聚合接口的协商加入等操作，等 FortiLink 的管理结束之后，在 FortiGate 上会看到一个网站的 FortiSwitch 拓扑图，和我们规划的拓扑图应该是一致的，可以仔细核对接线是否正确。
7. 交换机加入到 FortiGate 的 FortiLink，会先清除配置再重启，然后从 standalone 的状态进入到“Fortilink remote control”的状态，后续所有的配置都通过 FortiGate 下发即可。
8. 编辑交换机，可以修改交换机名字以及升级版本，如果交换机很多的话，规划好的命名可以有效的迅速找到交换机的位置。





9. FortiSwitch 之间接多根线进行 FortiLink 互联，会默认自动聚合。其实接一根网线也是聚合的，只是聚合组里面只有一个接口而已，因此多根网线直连，这些直连的线都会自动加入到一个聚合接口里面（建议 MC - LAG 环境尽量使用两条或以上的连线）。

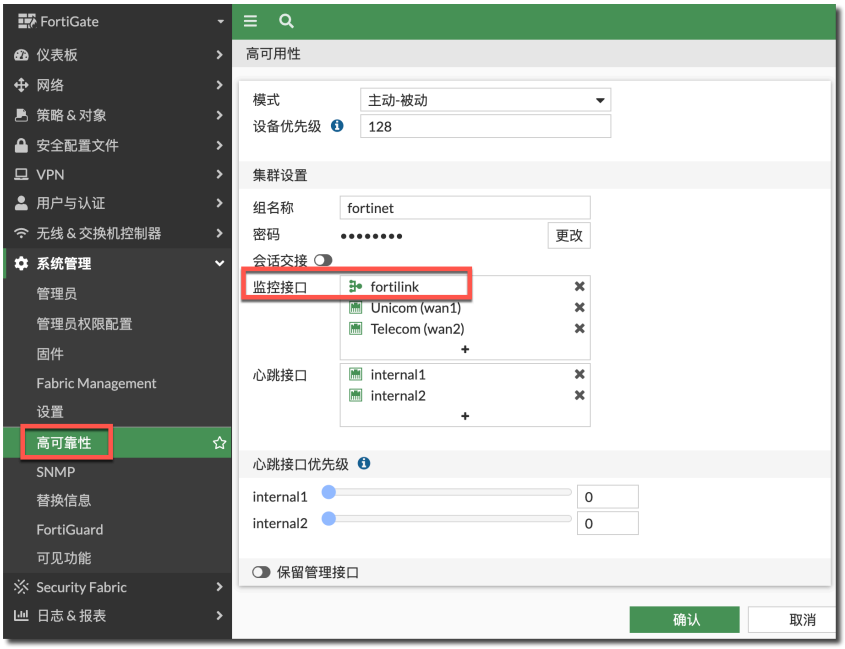
```
FortiGate # exec ssh admin@10.255.1.6
admin@10.255.1.6's password:

Core-FSW1 # show switch trunk

config switch trunk
    edit "4DF3X15000291-0"
        set mode lacp-active
        set auto-isl 1
        set mclag-icl enable
        set members "port1" "port2"
    next
    edit "GT61FTK22054602"
        set mode lacp-active
```

```
        set auto-isl 1
        set fortilink 1
        set mclag enable
        set members "port24"
    next
    edit "GT61FTK22054783"
        set mode lacp-active
        set auto-isl 1
        set fortilink 1
        set mclag enable
        set members "port23"
    next
    edit "mclag-tier2"
        set mode lacp-active
        set auto-isl 1
        set mclag enable
        set members "port4" "port3"
    next
end
```

10. 再次查看 FortiGate 的 FortiLink 成员，可以看到两台接入交换机已上线，且连线与拓扑一致。

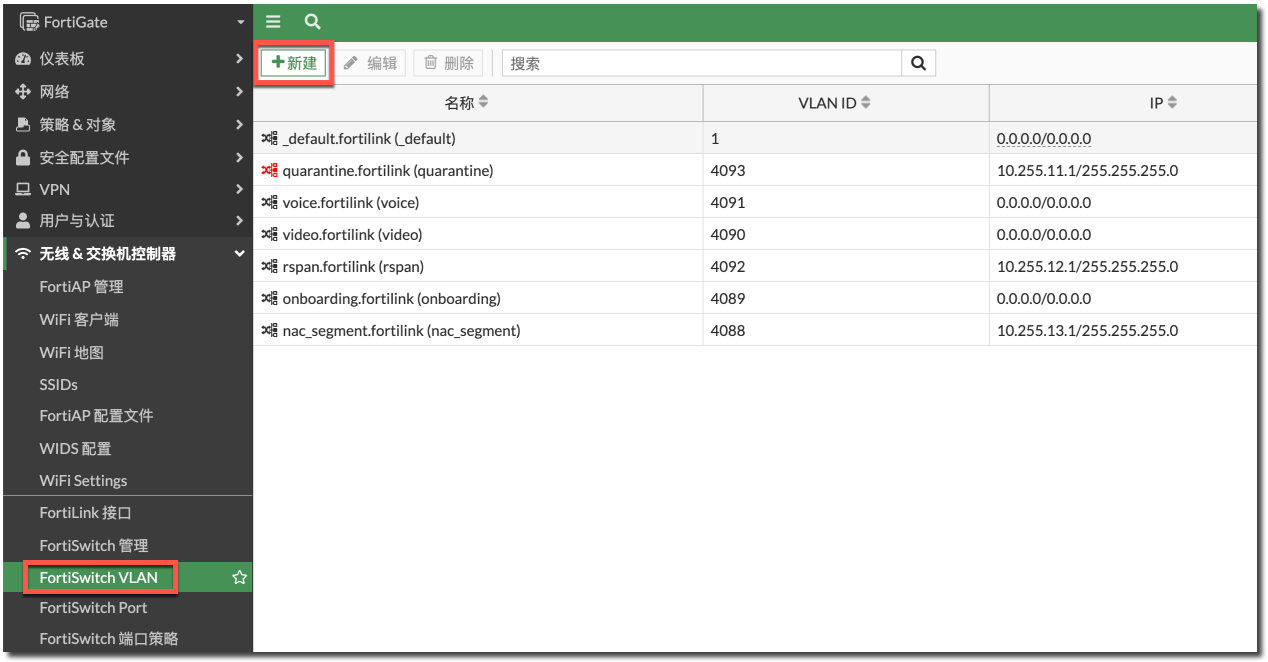
```
FortiGate # execute switch-controller get-physical-conn standard fortilink

This will display connectivity graph information for FortiLink from FortiGate's perspective
       FortiLink interface : fortilink
FortiGate(s)
FGT61FTK22054783(a)  «————————————»  S224DF3X15000373(port23)
FGT61FTK22054602(a)  «————————————»  S224DF3X15000373(port24)
FGT61FTK22054783(b)  «————————————»  S224DF3X15000291(port23)
FGT61FTK22054602(b)  «————————————»  S224DF3X15000291(port24)
Tier 1
S224DF3X15000373(port23)  «————————————»  FGT61FTK22054783(a)
S224DF3X15000373(port24)  «————————————»  FGT61FTK22054602(a)
S224DF3X15000291(port23)  «————————————»  FGT61FTK22054783(b)
S224DF3X15000291(port24)  «————————————»  FGT61FTK22054602(b)
Tier 2+
S224ENTF20002824(port23/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000373(port4/mclag-tier2)
S224ENTF20002824(port24/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000291(port4/mclag-tier2)
S224ENTF22001220(port21/4ENTF20002824-0)  «————————————»  S224ENTF20002824(port21/4ENTF22001220-0)
S224ENTF22001220(port22/4ENTF20002824-0)  «————————————»  S224ENTF20002824(port22/4ENTF22001220-0)
S224ENTF22001220(port23/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000373(port3/mclag-tier2)
S224ENTF22001220(port24/_FlInK1_MLAG0_)  «————————————»  S224DF3X15000291(port3/mclag-tier2)
S224DF3X15000373(port1/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port1/4DF3X15000373-0)
S224DF3X15000373(port2/4DF3X15000291-0)  «————————————»  S224DF3X15000291(port2/4DF3X15000373-0)
```

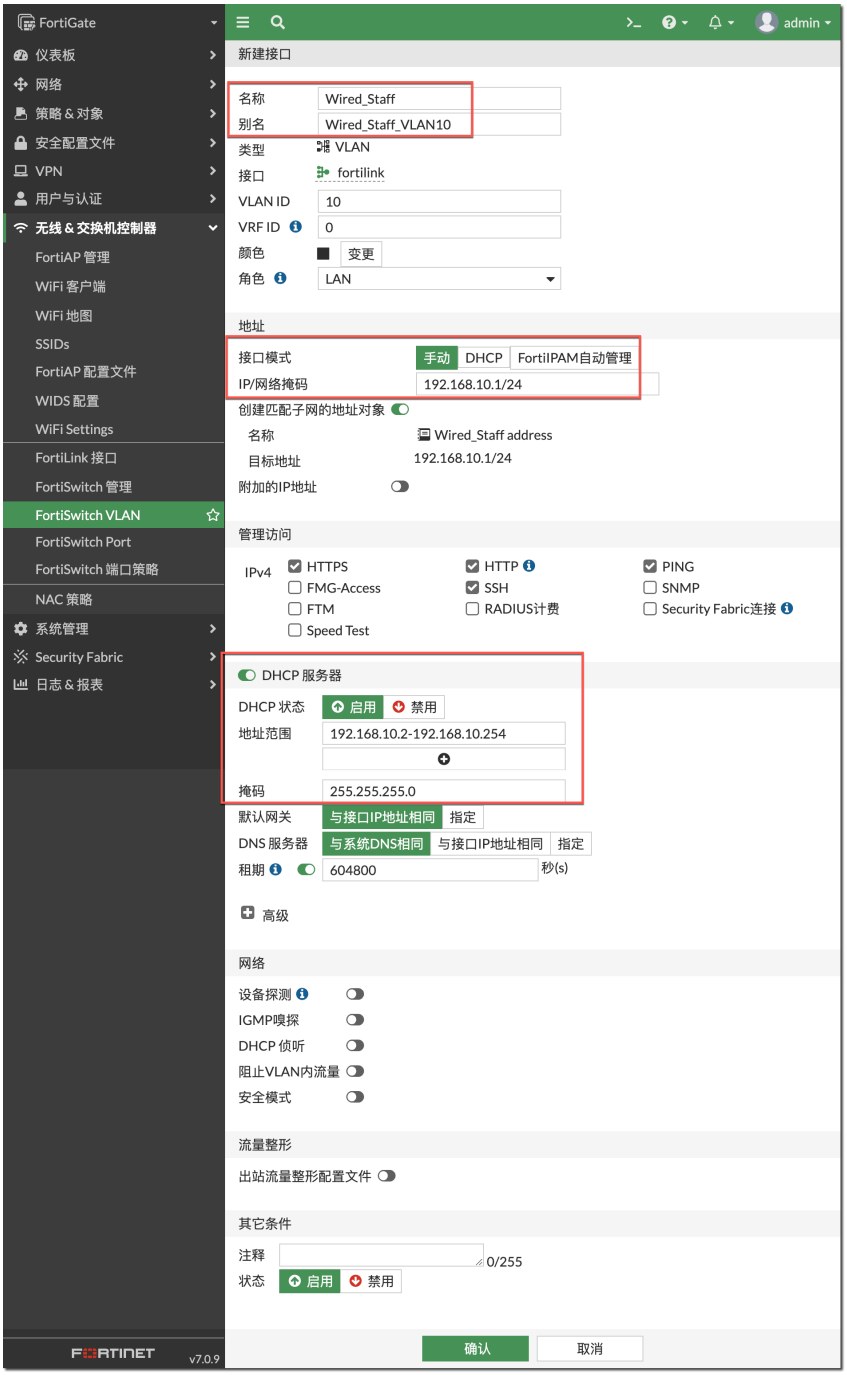
11. 最后不要忘记，在 FortiLink 的聚合接口和 FortiLink 管理配置完毕，记得把这个 FortiLink 聚合接口加入到 HA 的监控接口列表中，这样当主核心交换机挂了（比如断电）之后，业务才可以平稳切换到备防火墙上。

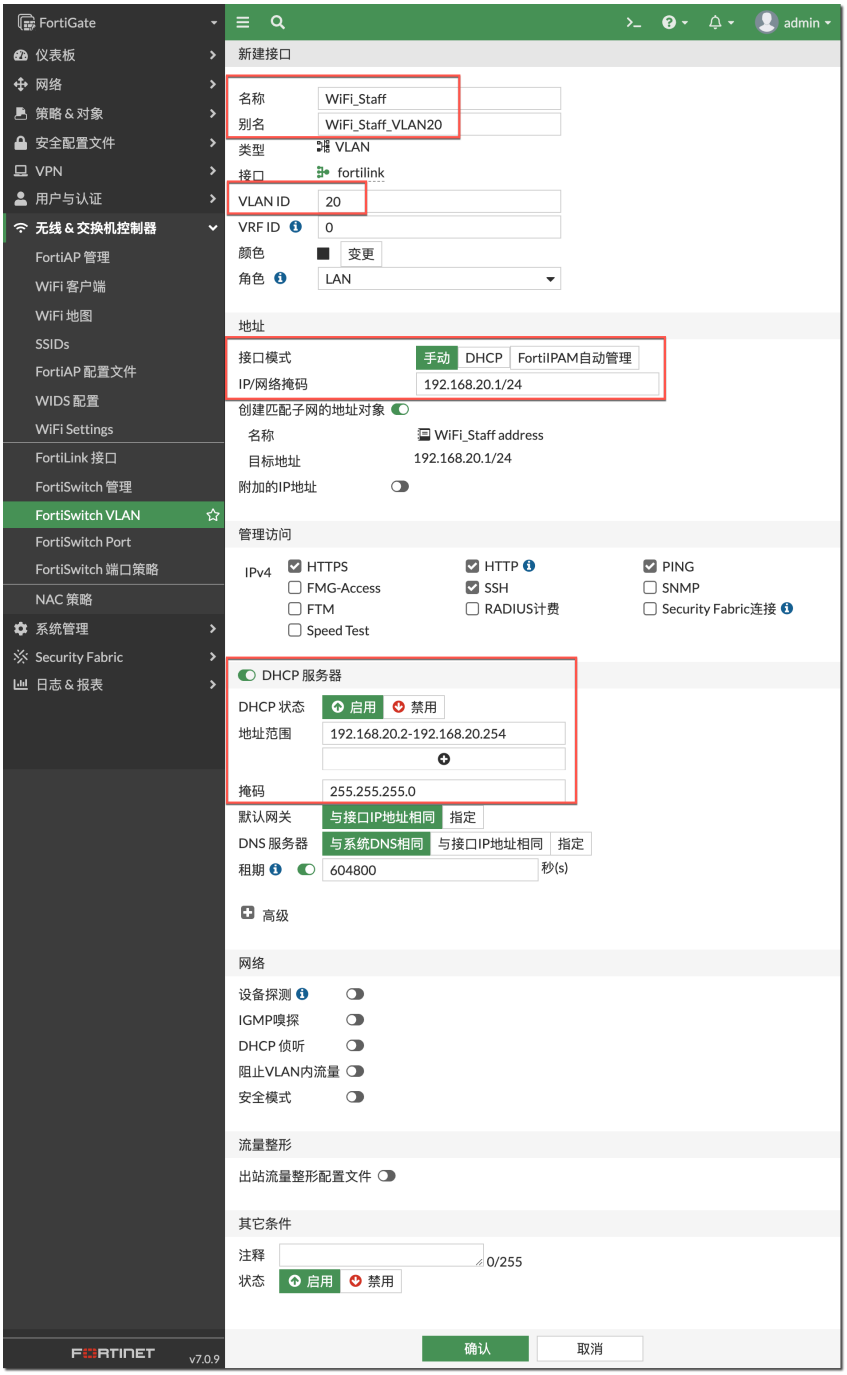


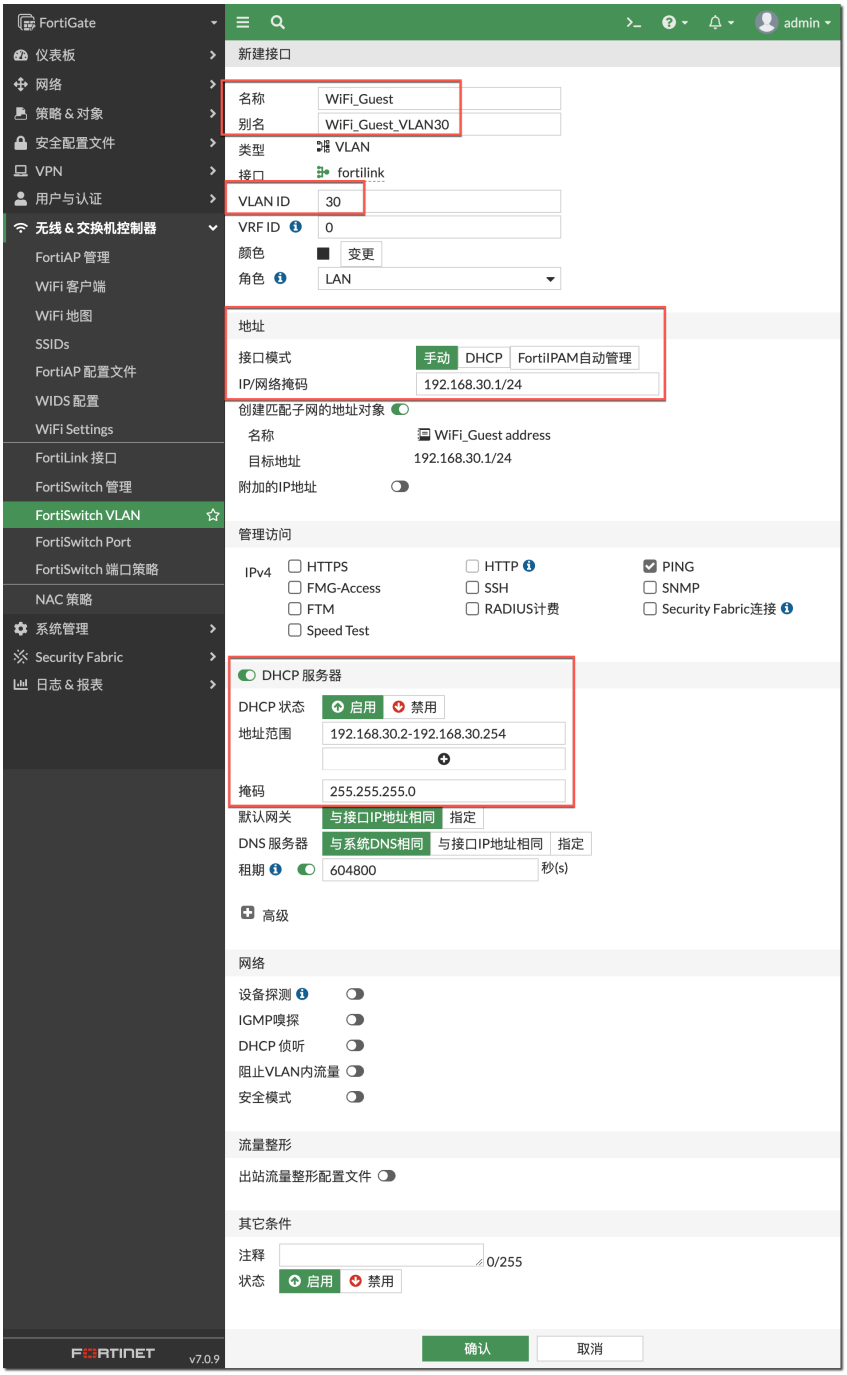
创建 VLAN

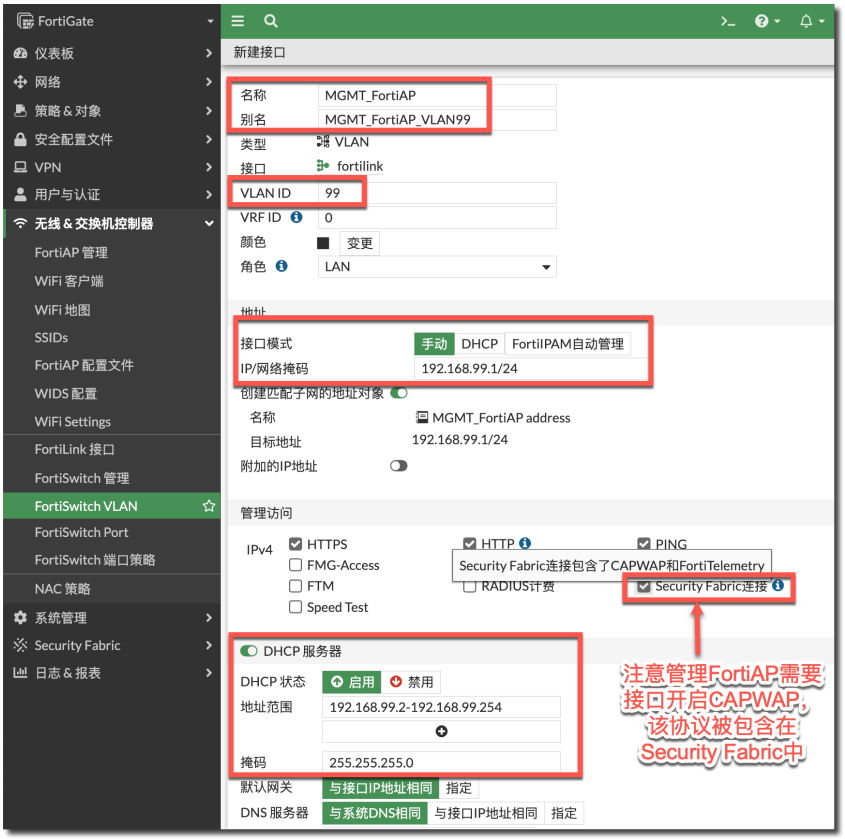
1. FortiGate 通过 FortiLink + Capwap 协议统一的接管 FortiSwitch 和 FortiAP，为有线和无线统一进行管理和业务处理。
2. 进入 FortiGate 的无线 & 交换机控制器 → FortiSwitch VLAN 页面，新建有线网络 VLAN 10、员工无线网络 VLAN 20、访客无线网络 VLAN 30、管理 FortiAP VLAN 99。注意管理 FortiAP 的 VLAN 接口要开启 Security Fabric，包含 CAPWAP。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
+新建
编辑
删除
搜索
名称
VLAN ID
IP
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0

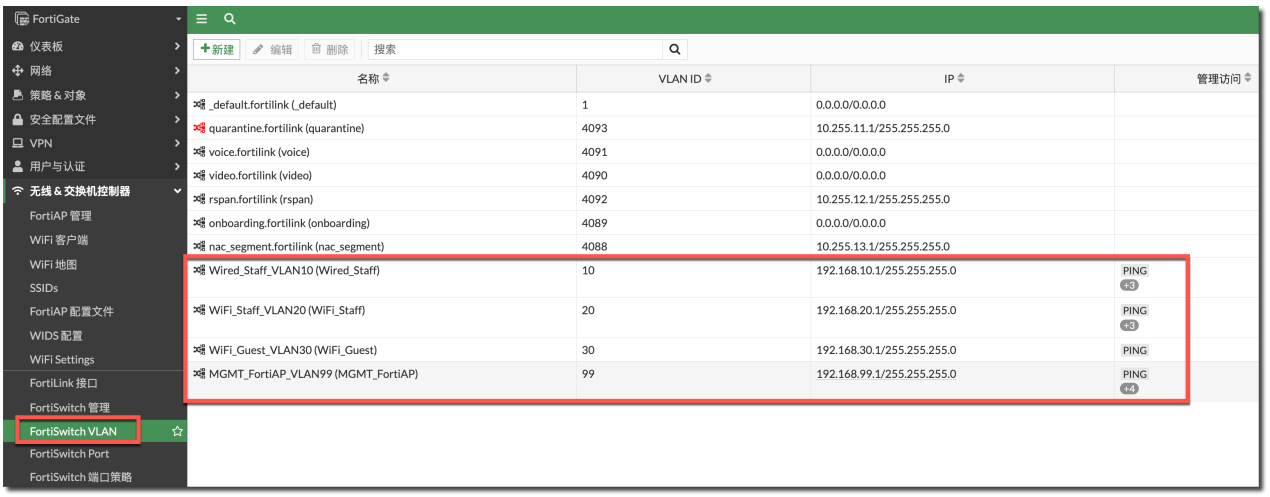

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 Wired_Staff
别名 Wired_Staff_VLAN10
类型 VLAN
接口 fortilink
VLAN ID 10
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.10.1/24
创建匹配子网的地址对象
名称 Wired_Staff address
目标地址 192.168.10.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.10.2-192.168.10.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
admin
新建接口
名称 WiFi_Staff
别名 WiFi_Staff_VLAN20
类型 VLAN
接口 fortilink
VLAN ID 20
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.20.1/24
创建匹配子网的地址对象
名称 WiFi_Staff address
目标地址 192.168.20.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.20.2-192.168.20.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消

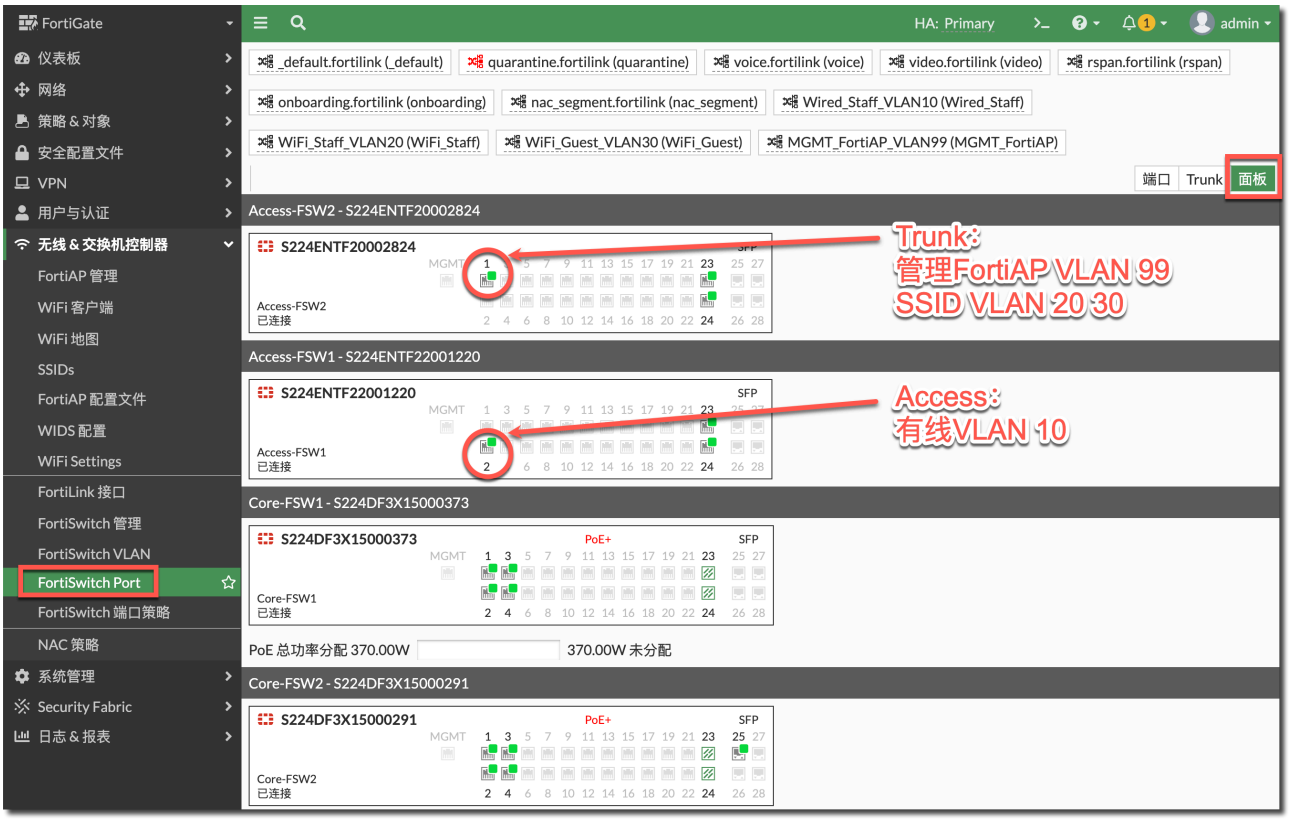

FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
新建接口
名称 WiFi_Guest
别名 WiFi_Guest_VLAN30
类型 VLAN
接口 fortilink
VLAN ID 30
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.30.1/24
创建匹配子网的地址对象
名称 WiFi_Guest address
目标地址 192.168.30.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access SSH SNMP
FTM RADIUS计费 Security Fabric连接
Speed Test
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.30.2-192.168.30.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
租期 604800 秒(s)
高级
网络
设备探测
IGMP嗅探
DHCP 侦听
阻止VLAN内流量
安全模式
流量整形
出站流量整形配置文件
其它条件
注释 0/255
状态 启用 禁用
确认
取消
FORTINET
v7.0.9


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
新建接口
名称 MGMT_FortiAP
别名 MGMT_FortiAP_VLAN99
类型 VLAN
接口 fortilink
VLAN ID 99
VRF ID 0
颜色 变更
角色 LAN
地址
接口模式 手动 DHCP FortiIPAM自动管理
IP/网络掩码 192.168.99.1/24
创建匹配子网的地址对象
名称 MGMT_FortiAP address
目标地址 192.168.99.1/24
附加的IP地址
管理访问
IPv4 HTTPS HTTP PING
FMG-Access
FTM
Speed Test
Security Fabric连接包含了CAPWAP和FortiTelemetry
RADIUS计费
Security Fabric连接
DHCP 服务器
DHCP 状态 启用 禁用
地址范围 192.168.99.2-192.168.99.254
掩码 255.255.255.0
默认网关 与接口IP地址相同 指定
DNS 服务器 与系统DNS相同 与接口IP地址相同 指定
注意管理FortiAP需要接口开启CAPWAP，该协议被包含在Security Fabric中

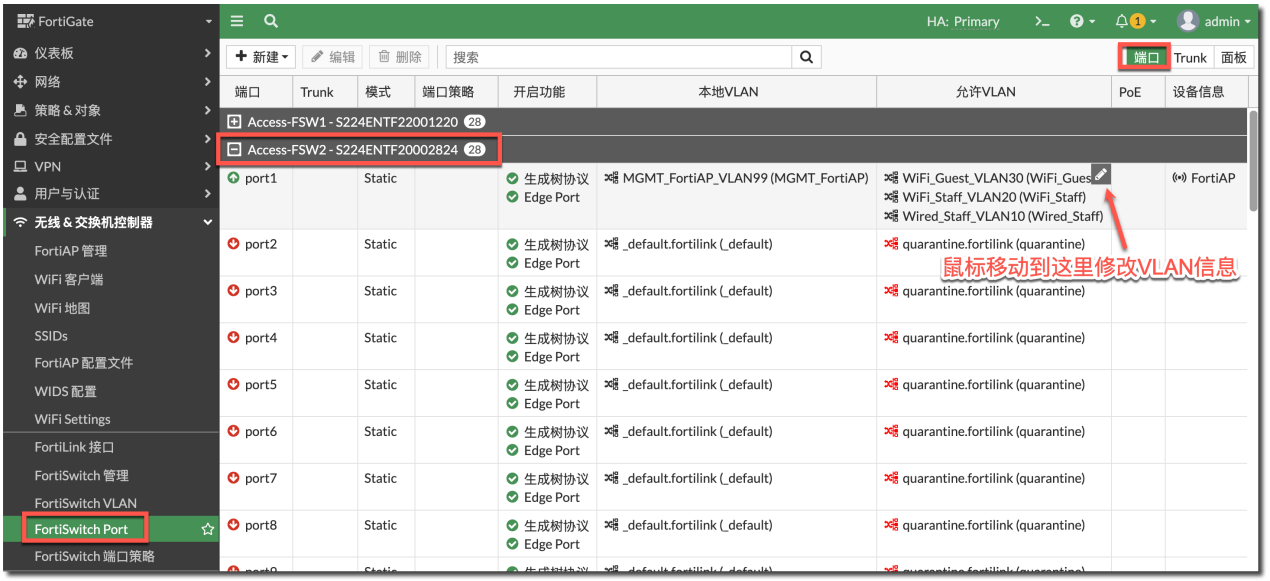
3. VLAN 创建完成。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
新建 编辑 删除 搜索
名称 VLAN ID IP 管理访问
_default.fortilink (_default) 1 0.0.0.0/0.0.0.0
quarantine.fortilink (quarantine) 4093 10.255.11.1/255.255.255.0
voice.fortilink (voice) 4091 0.0.0.0/0.0.0.0
video.fortilink (video) 4090 0.0.0.0/0.0.0.0
rspan.fortilink (rspan) 4092 10.255.12.1/255.255.255.0
onboarding.fortilink (onboarding) 4089 0.0.0.0/0.0.0.0
nac_segment.fortilink (nac_segment) 4088 10.255.13.1/255.255.255.0
Wired_Staff_VLAN10 (Wired_Staff) 10 192.168.10.1/255.255.255.0 PING
WiFi_Staff_VLAN20 (WiFi_Staff) 20 192.168.20.1/255.255.255.0 PING
WiFi_Guest_VLAN30 (WiFi_Guest) 30 192.168.30.1/255.255.255.0 PING
MGMT_FortiAP_VLAN99 (MGMT_FortiAP) 99 192.168.99.1/255.255.255.0 PING

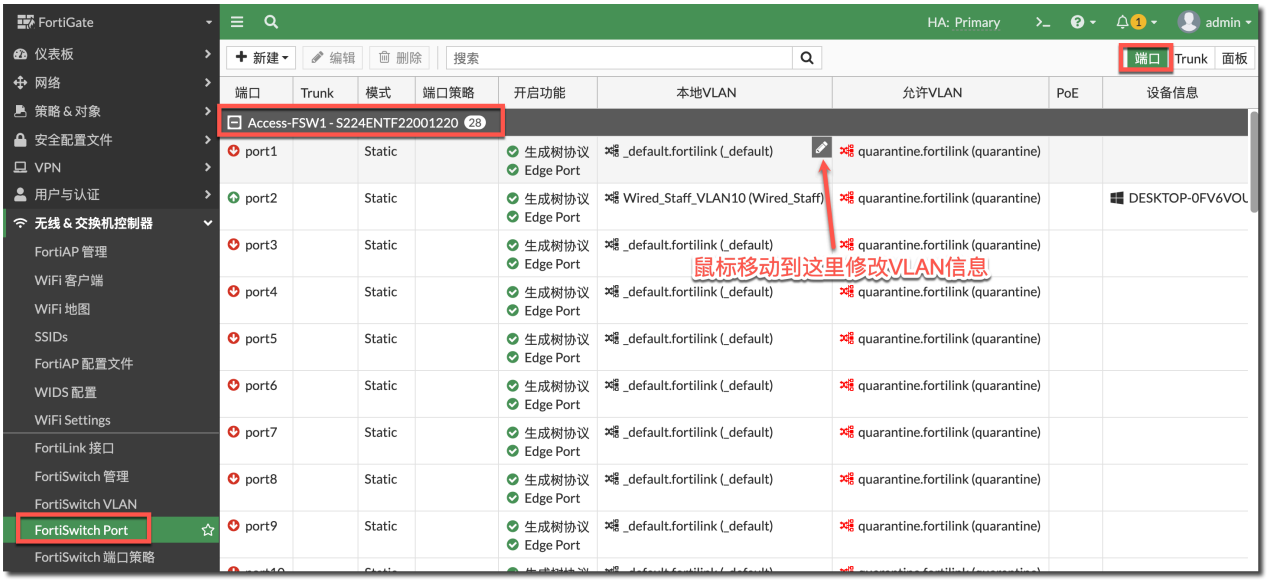
分配 VLAN



1. Access-FSW2 交换机的 port1 接 FortiAP，配置为 Trunk 模式，允许 VLAN20 和 VLAN30，VLAN20 用于 WIFI - STAFF 的 SSID 用户，VLAN30 用于 WIFI - GUEST 的 SSID 用户。另外本地 VLAN 设置为 VLAN 99，用于 FortiAP 的管理（FortiAP 的管理相当于使用不打 vlan-tag 的 VLAN 99，本地 VLAN 默认是 VLAN 1，因此只需要将本地 VLAN 设置为 VLAN 99 即可）。

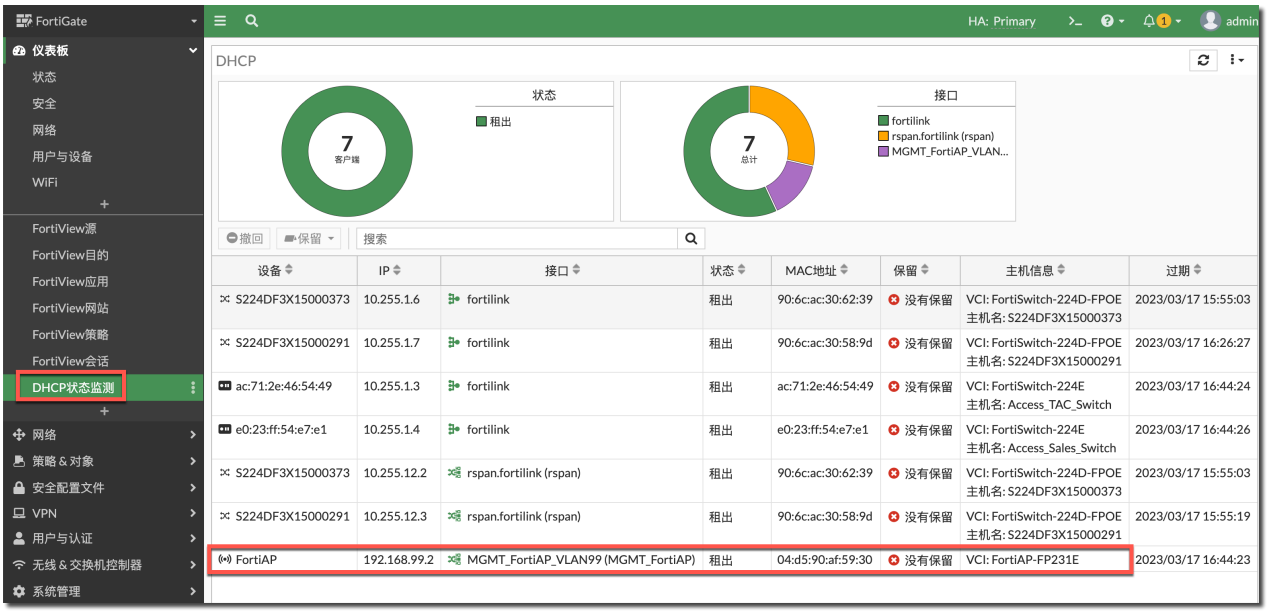


2. Access-FSW1 交换机的 Port2 接具体的物理 PC，因此只需要将 Port2 设置为 access 口，分配 VLAN 10 即可，用 FortiSwitch 的概念来说就是将本地 VLAN 设置为 VLAN 10，本地 VLAN 不打 vlan-tag，相当于就是设置接口为 access vlan 10。

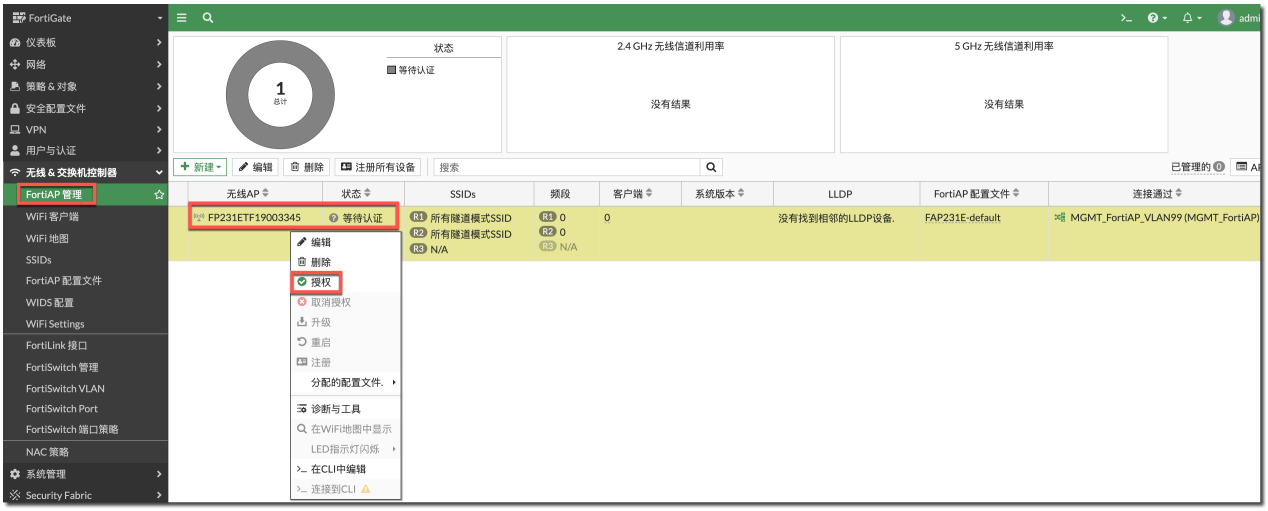


无线配置

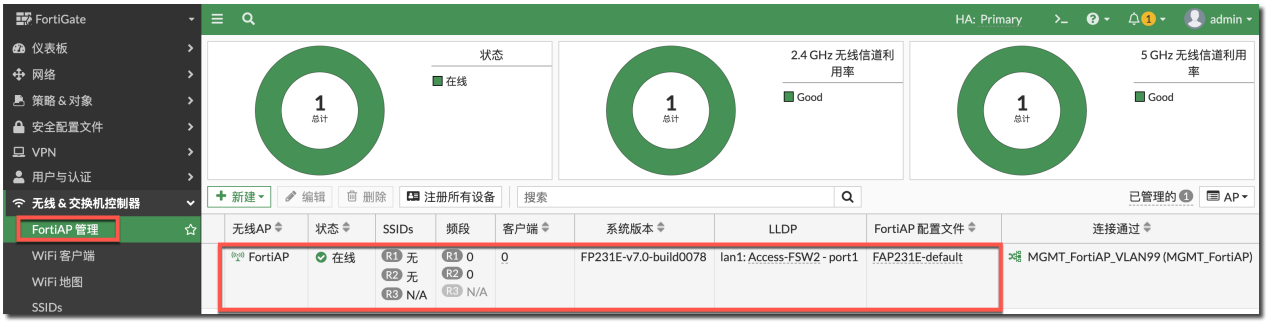
1. FortiAP 连接后，会通过 VLAN 99 的 DHCP 获取到 192.168.99.0/24 网段的 IP 地址。



2. 在之前的配置中，VLAN 99 的接口开启了 CAPWAP 协议（包含在 Security Fabric 中），激活了 FortiGate 的无线管理功能，这时候 FortiAP 会自动找到 FGT 进行注册。因此在 FortiAP 管理的地方可以看到这个 FortiAP 的信息（如果没有看到检查配置，并将 FortiAP 恢复出厂，让 FortiAP 重新获取 IP 地址并发起注册）。



3. 右键点击上图中等待认证的 FortiAP，点击授权按钮，等待几分钟，FortiAP 成功上线。



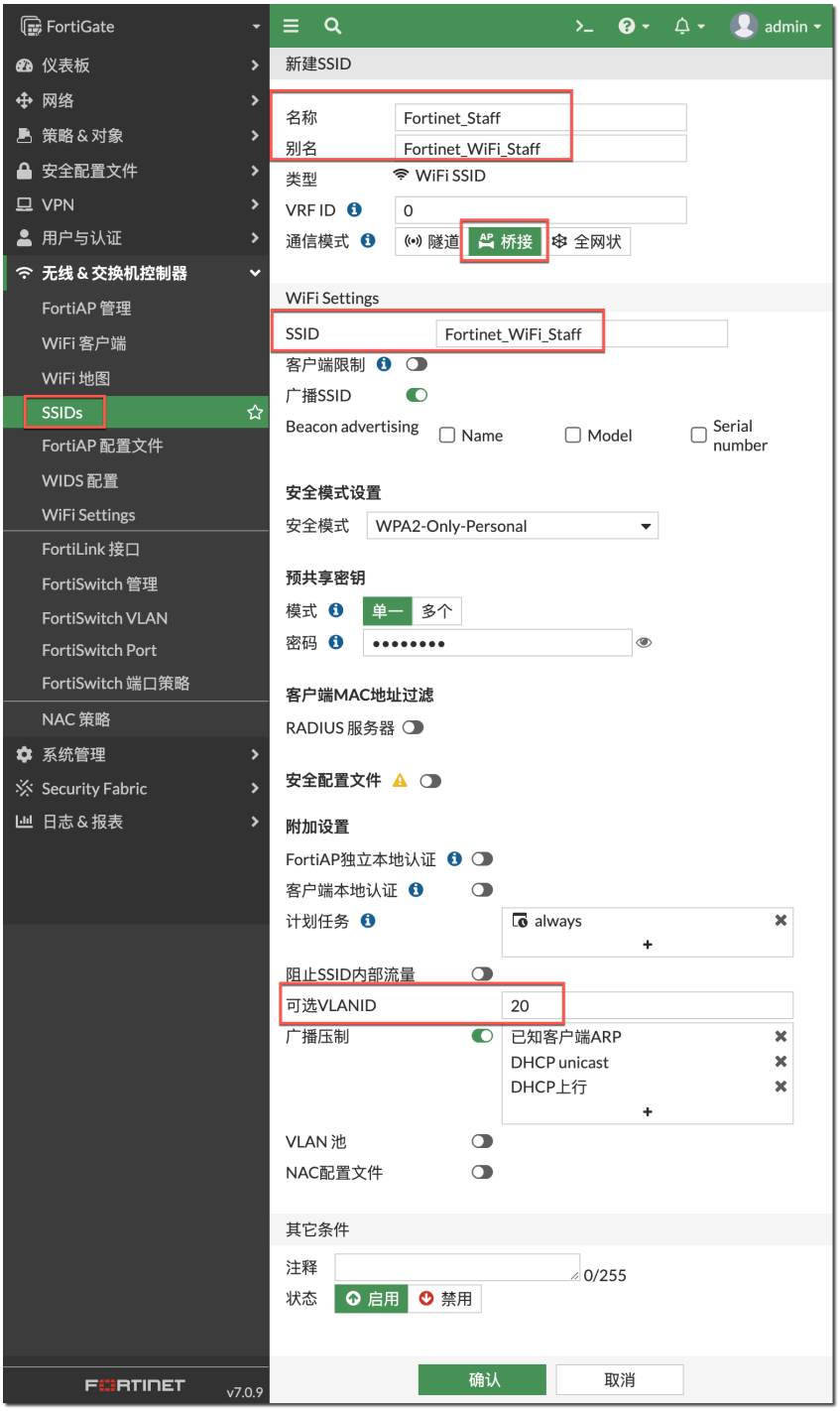
4. 编辑 FortiAP 设置，配置 FortiAP 的名称，或进行升级操作。

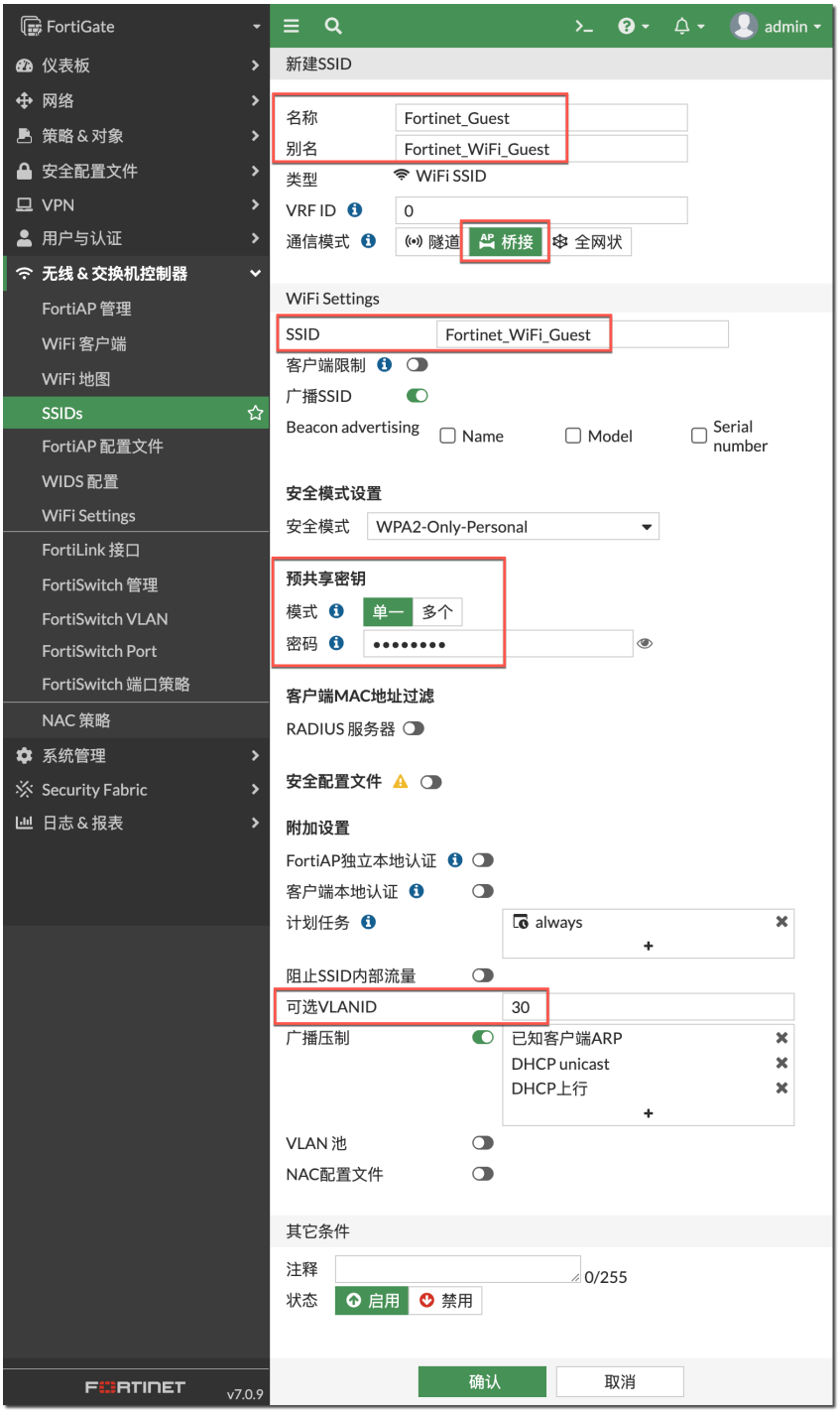


5. 进入无线 & 交换机控制器 → SSIDs 页面，点击新建按钮。



6. 新建 SSID：Fortinet_WiFi_Staff VLAN 20 用于员工 WIFI 使用，Fortinet_WiFi_Guest VLAN 30 用于访客 WiFi 使用。


FortiGate
admin
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.9
新建SSID
名称
Fortinet_Staff
别名
Fortinet_WiFi_Staff
类型
WiFi SSID
VRF ID
0
通信模式
隧道
桥接
全网状
WiFi Settings
SSID
Fortinet_WiFi_Staff
客户端限制
广播SSID
Beacon advertising
Name
Model
Serial number
安全模式设置
安全模式
WPA2-Only-Personal
预共享密钥
模式
单一
多个
密码
客户端MAC地址过滤
RADIUS 服务器
安全配置文件
附加设置
FortiAP独立本地认证
客户端本地认证
计划任务
always
阻止SSID内部流量
可选VLANID
20
广播压制
已知客户端ARP
DHCP unicast
DHCP上行
VLAN 池
NAC配置文件
其它条件
注释
0/255
状态
启用
禁用
确认
取消

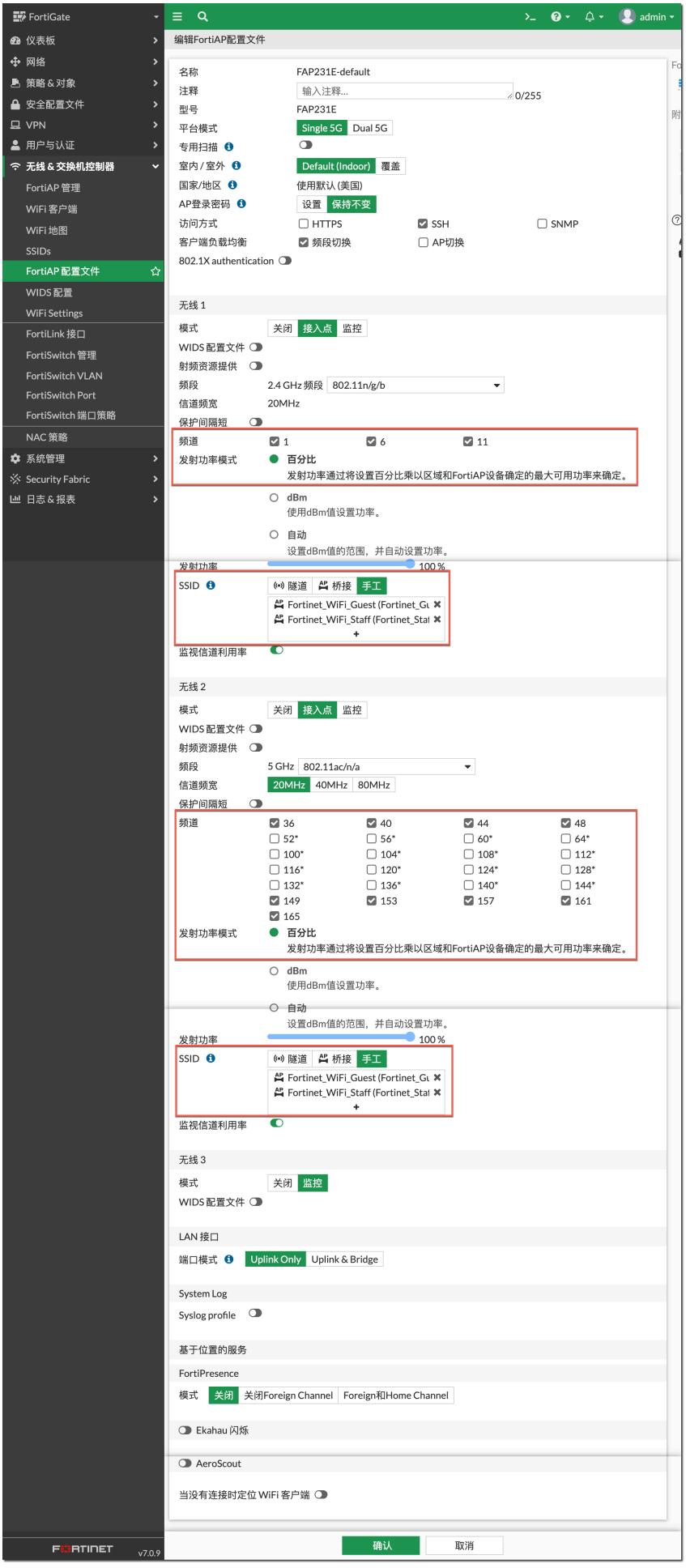




7. 将 SSID 关联到 FortiAP 上，然后让 FortiAP 发布无线 SSID 信号，这样用户就可以通过 SSID 连接到无线了。进入 FortiAP 配置文件页面，编辑 FortiAP 关联的配置文件。



8. 桥模式的 SSID 需要在 FortiAP 配置文件里面手工指定一下，另外下图标示出的配置为推荐配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
FortiAP 管理
WiFi 客户端
WiFi 地图
SSIDs
FortiAP 配置文件
WIDS 配置
WiFi Settings
FortiLink 接口
FortiSwitch 管理
FortiSwitch VLAN
FortiSwitch Port
FortiSwitch 端口策略
NAC 策略
系统管理
Security Fabric
日志 & 报表
admin
编辑FortiAP配置文件
名称 FAP231E-default
注释 输入注释... 0/255
型号 FAP231E
平台模式 Single 5G Dual 5G
专用扫描
室内 / 室外 Default (Indoor) 覆盖
国家/地区 使用默认 (美国)
AP登录密码 设置 保持不变
访问方式 HTTPS SSH SNMP
客户端负载均衡 频段切换 AP切换
802.1X authentication
无线 1
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 2.4 GHz 频段 802.11n/g/b
信道频宽 20MHz
保护间隔短
频道 1 6 11
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 2
模式 关闭 接入点 监控
WIDS 配置文件
射频资源提供
频段 5 GHz 802.11ac/n/a
信道频宽 20MHz 40MHz 80MHz
保护间隔短
频道 36 40 44 48
52* 56* 60* 64*
100* 104* 108* 112*
116* 120* 124* 128*
132* 136* 140* 144*
149 153 157 161
165
发射功率模式 百分比
发射功率通过将设置百分比乘以区域和FortiAP设备确定的最大可用功率来确定。
dBm
使用dBm值设置功率。
自动
设置dBm值的范围，并自动设置功率。
发射功率 100 %
SSID 隧道 桥接 手工
Fortinet_WiFi_Guest (Fortinet_Gu
Fortinet_WiFi_Staff (Fortinet_Stal
监视信道利用率
无线 3
模式 关闭 监控
WIDS 配置文件
LAN 接口
端口模式 Uplink Only Uplink & Bridge
System Log
Syslog profile
基于位置的服务
FortiPresence
模式 关闭 关闭Foreign Channel Foreign和Home Channel
Ekahau 闪烁
AeroScout
当没有连接时定位 WiFi 客户端
确认
取消
FORTINET
v7.0.9

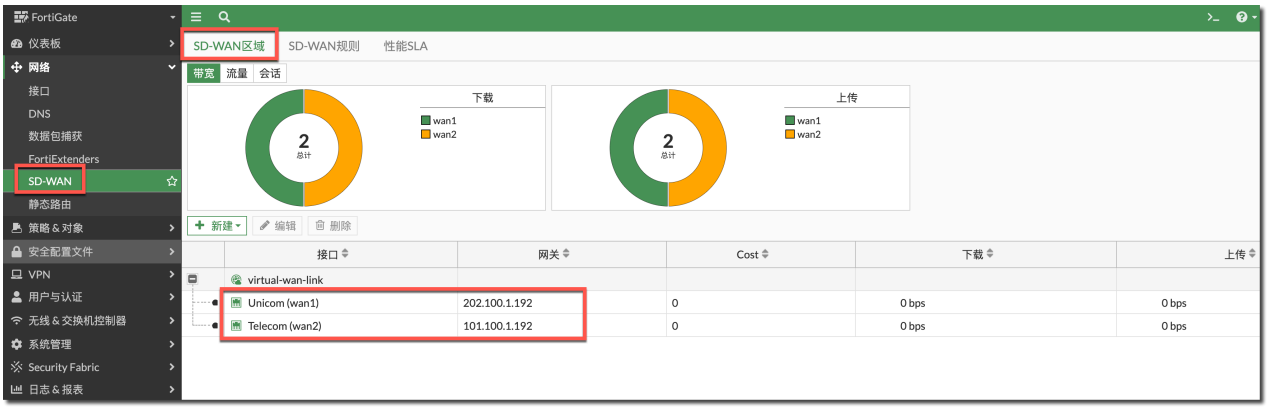


9. FortiAP 会自动关联到这个 FortiAP 配置文件里面的内容，发起 SSID 的信号。



SD-WAN 配置

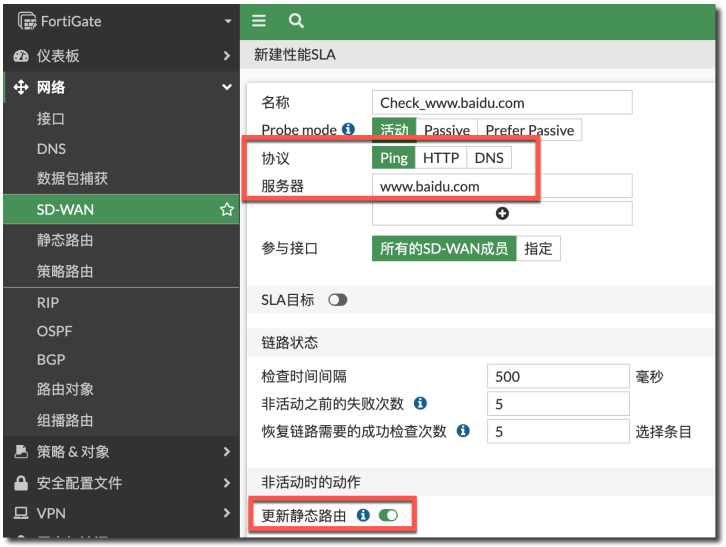
1. 将 wan1 和 wan2 加入 SD-WAN 接口，并配置各自的网关。

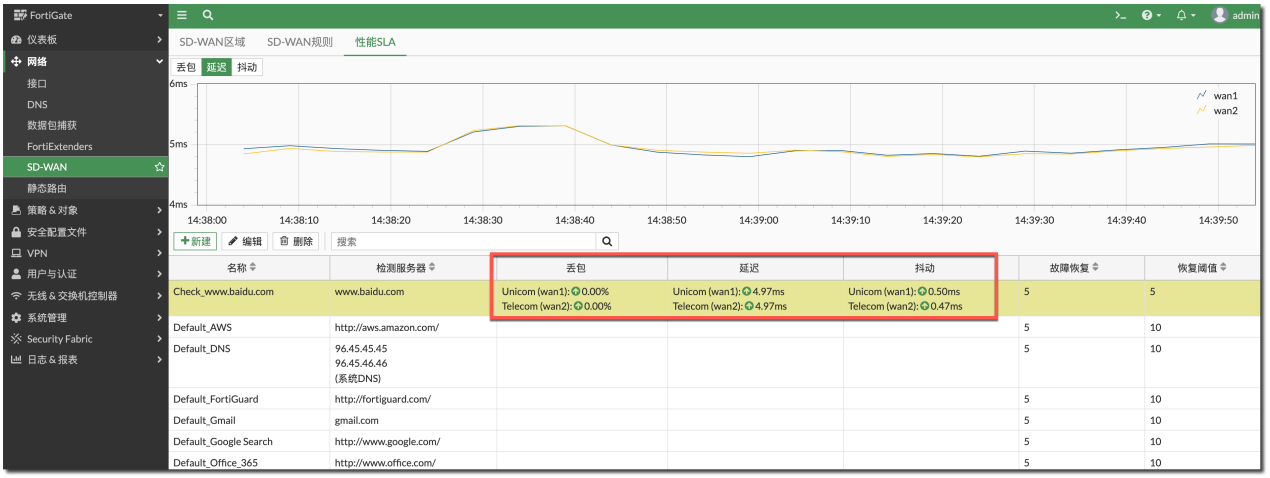


2. 添加 SD-WAN 的默认路由。



3. 配置 SD-WAN 的 SLA，以便 wan1 和 wan2 相互备份和切换。





4. SD-WAN 规则可根据需求配置，这里保持默认的按照源 IP 负载均衡。

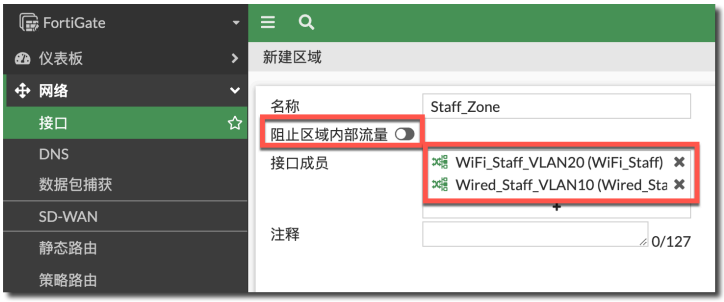


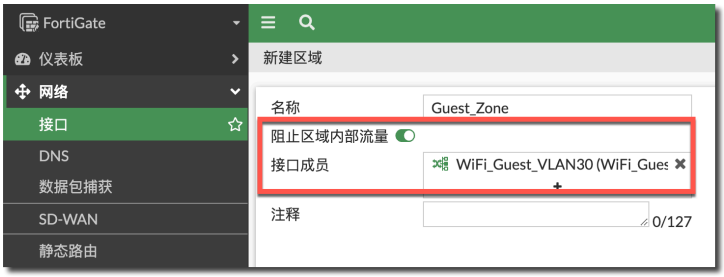
安全策略配置

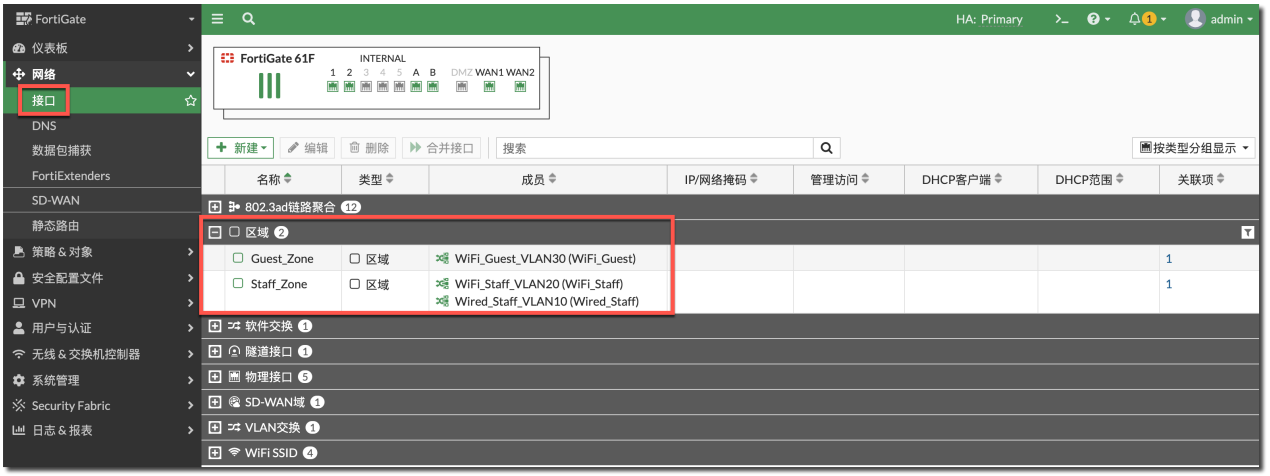
1. 需求与规划：
   - 员工的有线与无线都需要访问互联网
   - 访客的无线仅仅可以访问互联网
   - 员工的有线和无线需要互访
2. 建议将防火墙的接口划分区域，以区域的方式配置策略，有时候可以做到简化配置的目的，甚至未来替换设备的时候，策略的备份和还原也会更加简单一些。规划区域两个区域：
   - 员工网络区域：包括员工的有线 VLAN 10 和员工的无线 VLAN 20 （区域之间的隔离关闭，这样 VLAN 10 和 VLAN 20 之间的数据可通）
   - 访问网络区域：包括访问的无线 VLAN 30
3. 进入网络 → 接口页面，新建所属区域。



4. 分别创建 Staff_Zone（包含 VLAN 10 和 VLAN20）和 Guest_Zone（包含 VLAN30）。







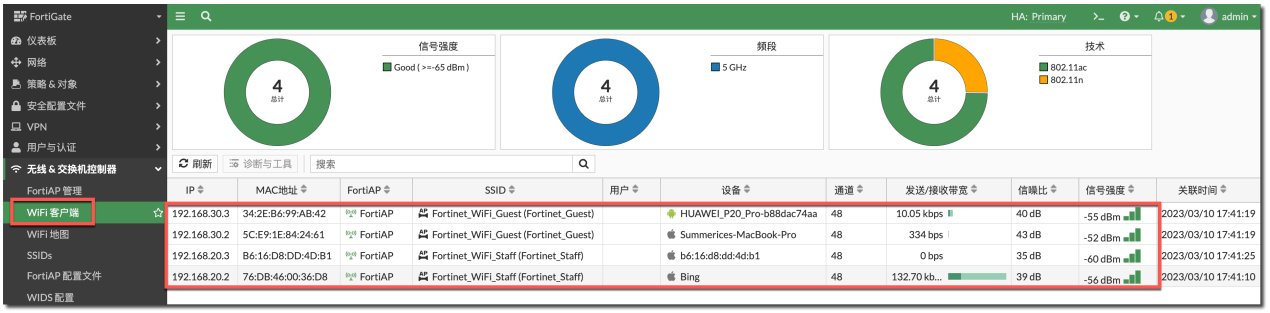
5. 如此只需要配置两条策略，就可以实现我们规划的需求，新建 Staff-Zone 到 SD-WAN 与 Guest-Zone 到 SD-WAN 的安全策略。



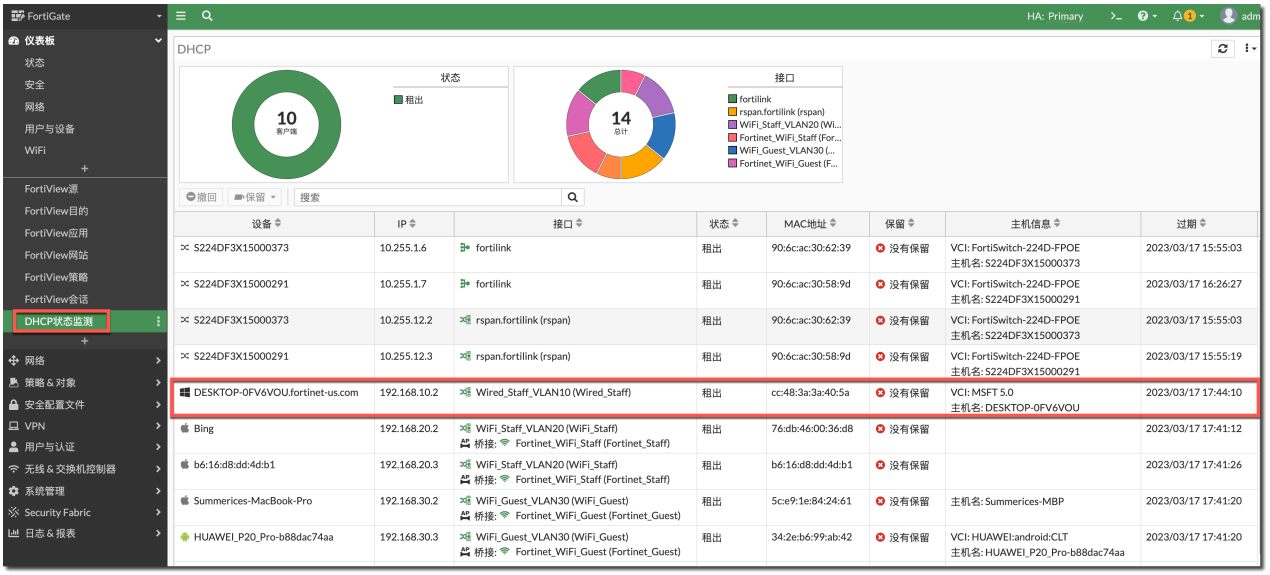
结果验证

客户端访问

1. 进入无线 & 交换机控制器 → WiFi 客户端，查看在线的无线客户端。可以看到 WiFi_Staff 和 WiFi_Guest 各有一个终端在线。



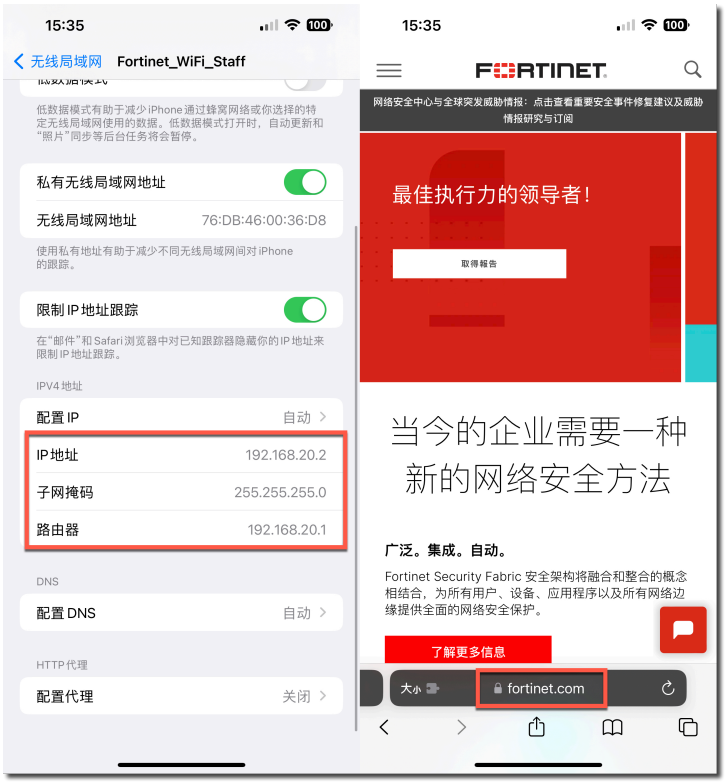
2. 在 DHCP 状态检测中查看所有的客户端信息，可以看到有线网络的 PC 也正常上线。



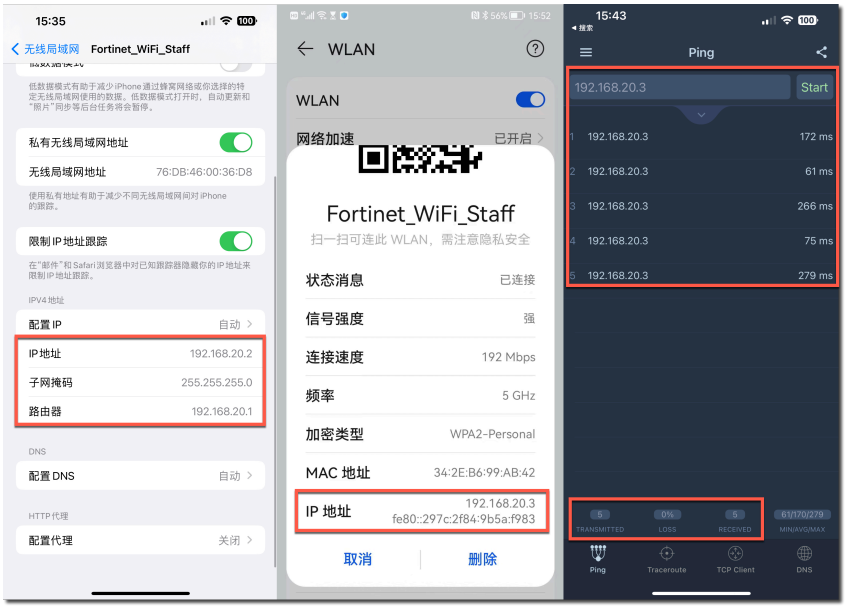
3. 有线网络（VLAN 10）用户访问互联网。



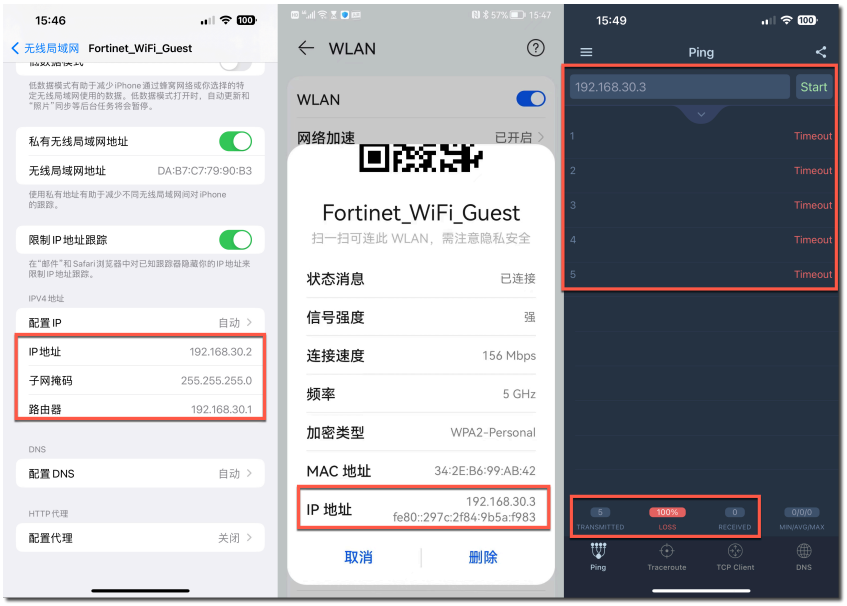
4. 无线网络用户访问互联网。



5. 使用 VLAN 20 下的客户端访问 VLAN 20 下的另一个客户端（Staff 区域）。



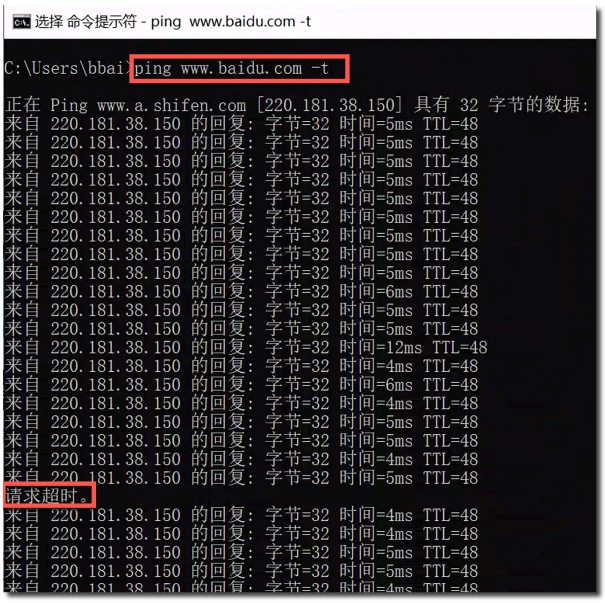
6. 使用 VLAN 30 下的客户端访问 VLAN 30 下的另一个客户端（Guest 区域），可以看到由于该区域配置了禁止相互访问，所以 VLAN 30 下的客户端是无法互相访问的。



FortiLink 切换测试

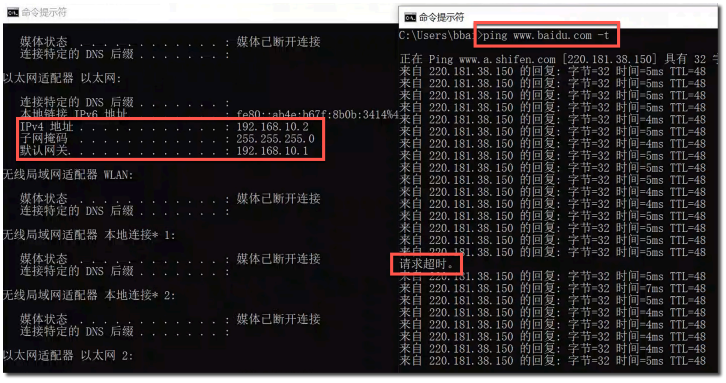
1. 在我们的测试场景下 FortiLink 聚合接口的 a 和 b 是同时 UP 起来的，如果 a 或 b DOWN（不能同时 DOWN），是否有影响业务呢?

2. 经过测试，DOWN a 或 b 接口，客户端丢包 1 个或者不丢包。



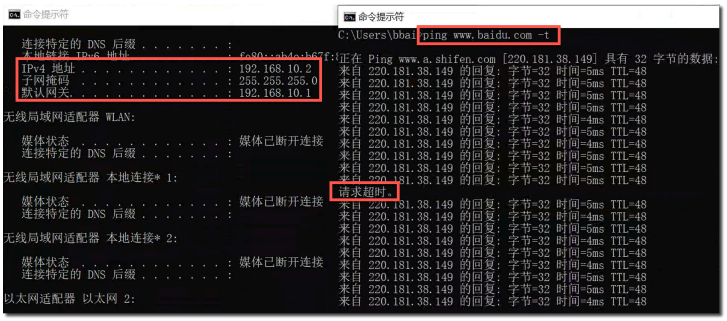
HA 切换测试

重启当前主防火墙查看内网 PC 丢包情况，HA 切换丢包 1 个。

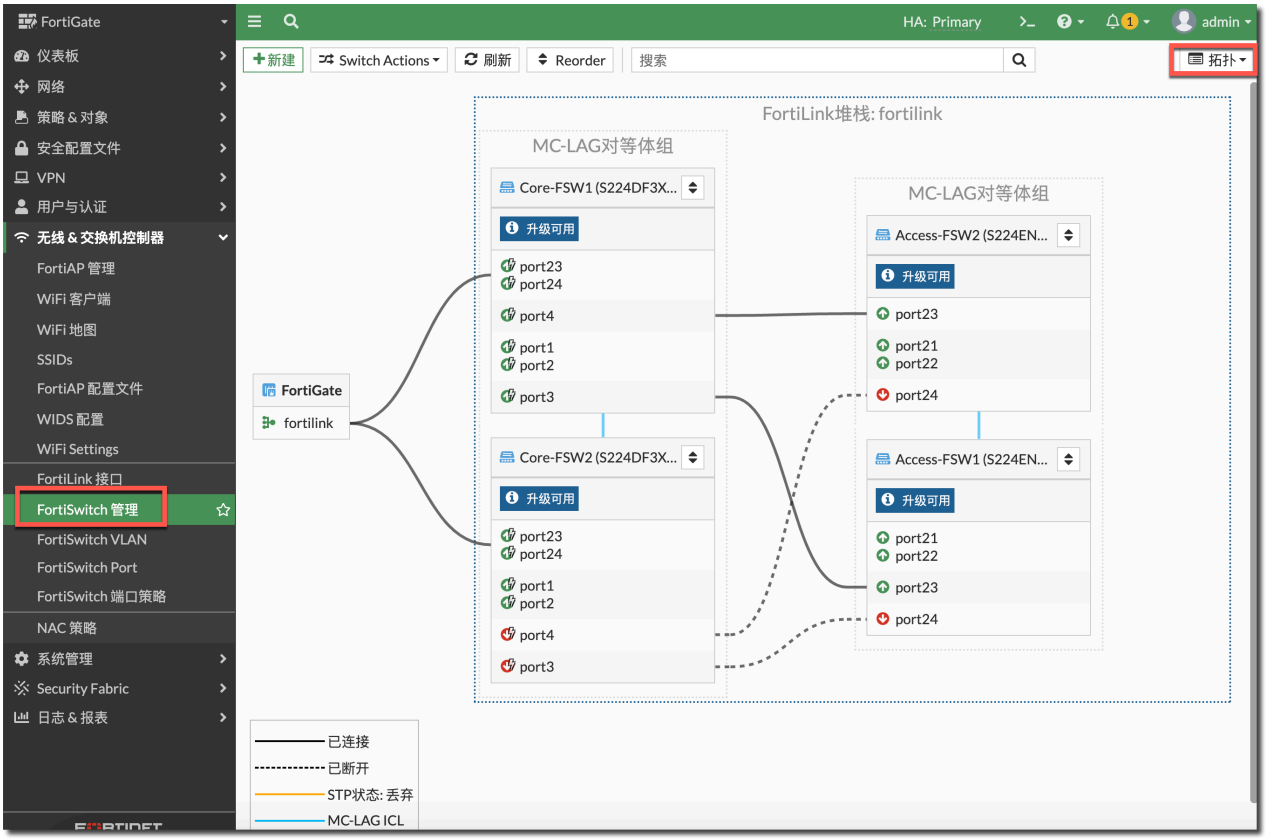


核心交换机接口故障模拟

1. Down 掉 Core-FSW2 的 Port3 和 Port4 接口，查看内网 PC 丢包情况，可以看到 Core-FSW2 连接接入层交换机的接口 Down 掉时，丢包 1 个。



2. 查看和 Switch 管理拓扑，Core-FSW2 与接入层交换机的连线已断开。



注意事项

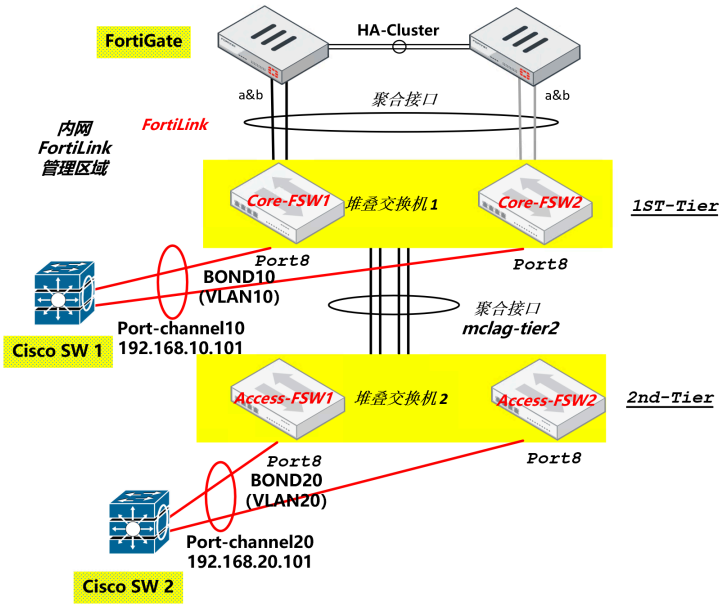
1. 只能是堆叠交换机才可以跨交换机聚合，不同的堆叠交换机组，不支持配置为聚合接口。
2. 如果是与 FortiSwith 对接，则是自动 MCLAG 聚合。
3. 如果是和第三方设备对接，比如交换机、路由器、友商防火墙、服务器等等对接的话，推荐使用 LACP Active 方式动态聚合。

FortiLink MCLAG - 与第三方设备进行跨堆叠交换机聚合

网络需求

本测试基于上一个章节“FortiLink MCLAG - 堆叠组网 2 Tier 配置举例 ↗”进行配置。

网络拓扑



1. Cisco SW1 的 Port-Channel 10 为三层接口，IP 地址为 192.168.10.101，我们在 Core-FSW1 和 Core-FSW2 跨交换机配置一个 BOND 10 的聚合接口与其对接，将其划分到 VLAN10 里面。
2. Cisco SW2 的 Port-Channel 20 为三层接口，IP 地址为 192.168.20.101，我们在 Access-FSW1 和 Access-FSW2 跨交换机配置一个 BOND 20 的聚合接口与其对接，将其划分到 VLAN20 里面。

配置步骤

Cisco 交换机配置

1. Cisco SW1 的聚合配置。

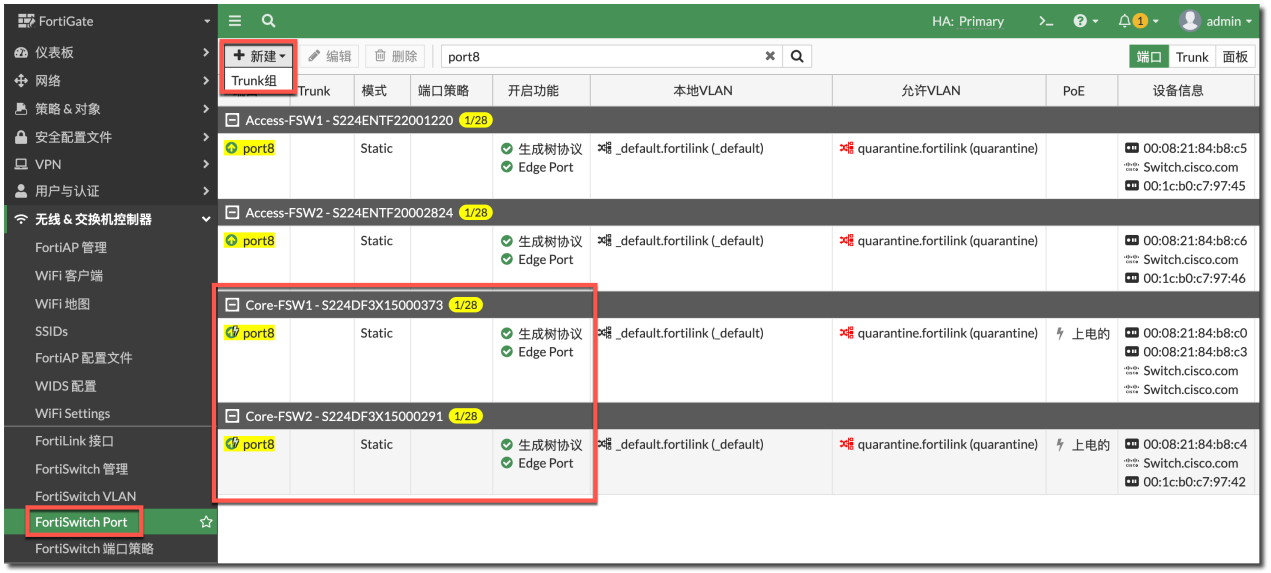
```
!
interface port-channel 10
no switchport
ip address 192.168.10.101 255.255.255.0
!
!
interface FastEthernet2/0/3
no switchport
no ip address
channel-group 10 mode active
!
interface FastEthernet2/0/4
no switchport
no ip address
channel-group 10 mode active
!
```

2. Cisco SW2 的聚合配置。

```
!
interface Port-channel20
no switchport
ip address 192.168.20.101 255.255.255.0
!
interface FastEthernet2/0/5
no switchport
no ip address
channel-group 10 mode active
!
interface FastEthernet2/0/6
no switchport
no ip address
channel-group 20 mode active
!
```

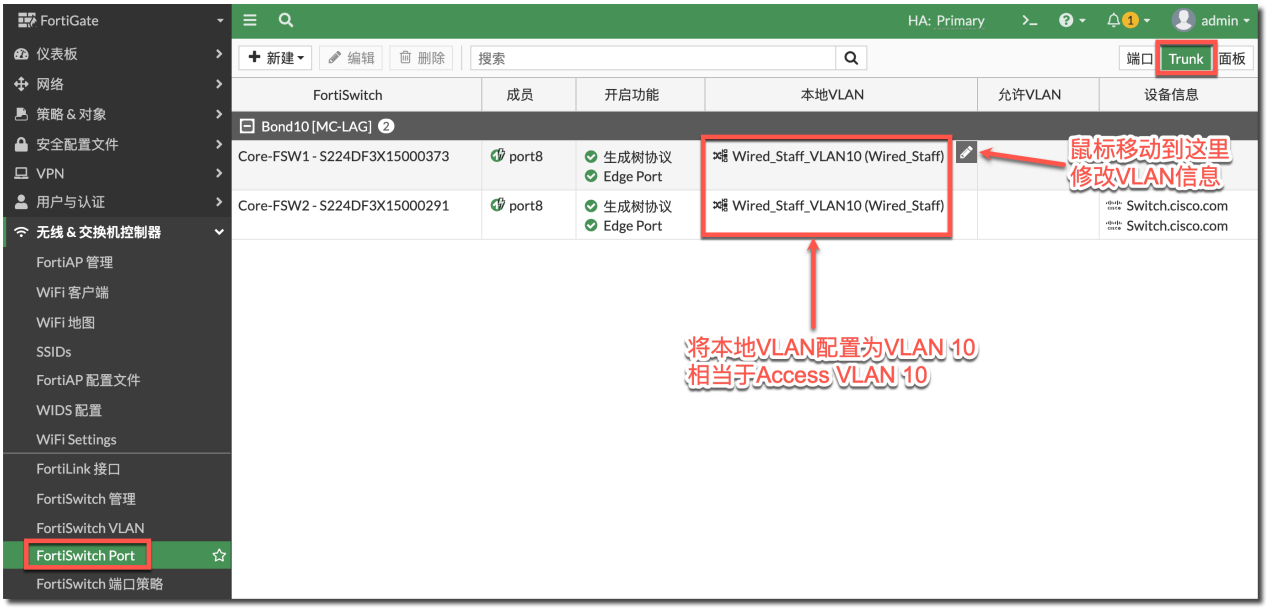
FortiGate 配置

1. 配置基于 Core-FSW1 和 Core-FSW2 跨交换机对接的 BOND 10。



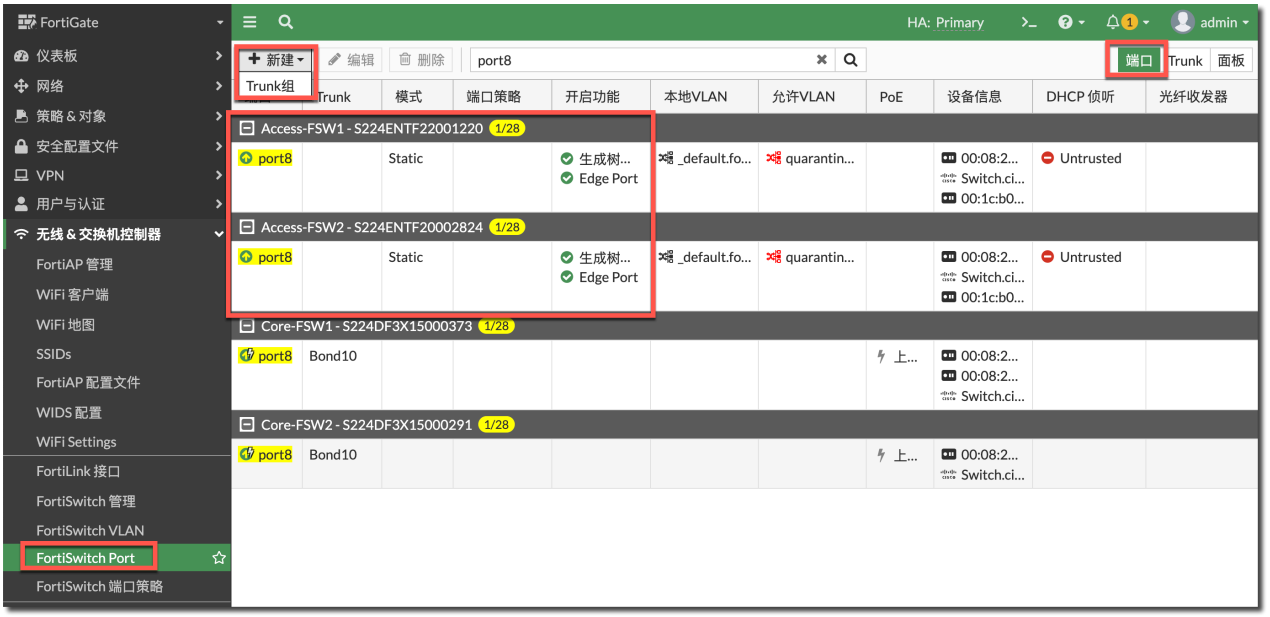


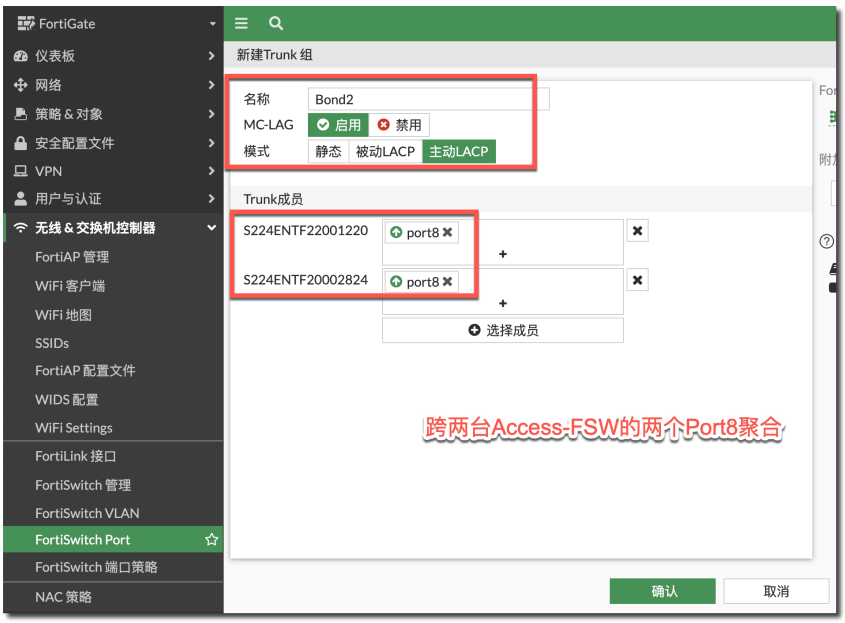
2. 进入 FortiSwitch Port 页面的 Trunk 视图模式，配置 Bond10 成员接口的本地 VLAN 为 10（相当于 Access VLAN 10）。



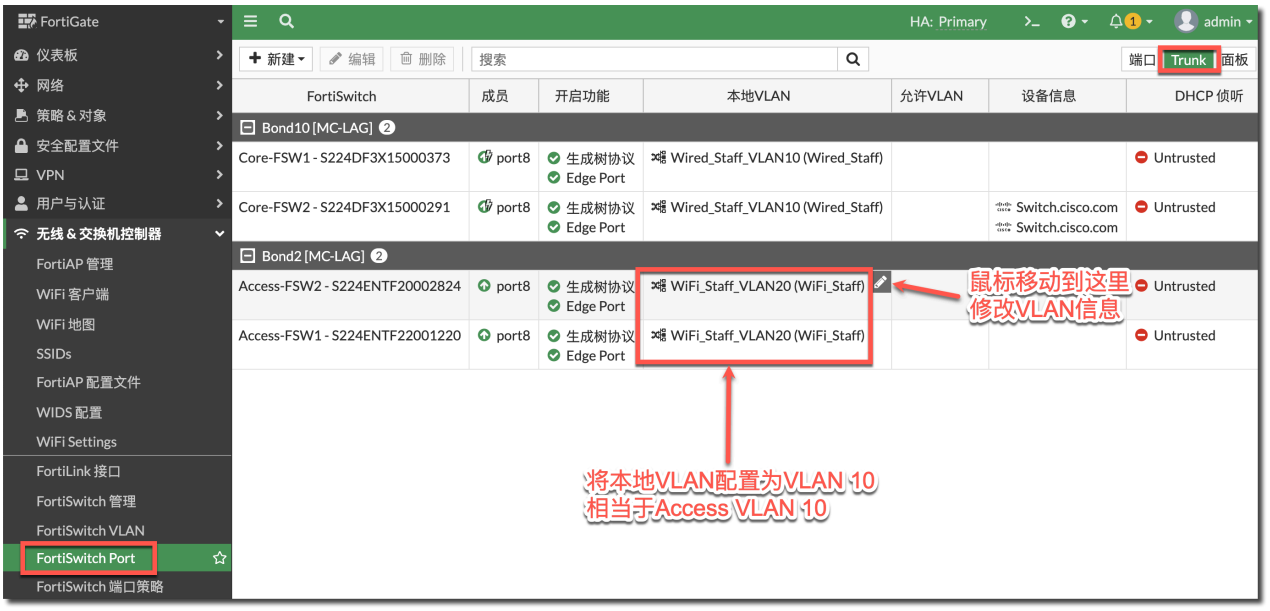

```
Core-FSW1:
config switch trunk
    edit "Bond10"
        set mode lacp-active
        set mclag enable
        set members "port8"
    next
end
config switch interface
    edit "Bond10"
        set native-vlan 10
    next
end

Core-FSW2:
config switch trunk
    edit "Bond10"
        set mode lacp-active
        set mclag enable
        set members "port8"
    next
end
config switch interface
    edit "Bond10"
        set native-vlan 10
    next
end
```

3. 配置基于 Access-FSW1 和 Access-FSW2 跨交换机对接的 BOND 20。





4. 进入 FortiSwitch Port 页面的 Trunk 视图模式，配置 Bond20 成员接口的本地 VLAN 为 20（相当于 Access VLAN 20）。



```
Access-FSW1:
config switch trunk
    edit "Bond20"
        set mode lacp-active
        set mclag enable
        set members "port8"
    next
end
config switch interface
    edit "Bond20"
        set native-vlan 20
    next
end

Access-FSW2:
config switch trunk
    edit "Bond20"
        set mode lacp-active
        set mclag enable
        set members "port8"
    next
end
config switch interface
    edit "Bond20"
        set native-vlan 20
    next
end
```

注意事项

重要

FortiSwitch 的 mclag-icl 堆叠组网中，与 Cisco 交换机聚合口对接时，还必须在 FortiSwitch 的 STP 配置中增加如下配置（set mclag-stp-bpdu single）。

bash

```
Core-FSW1:
config switch stp settings
    set mclag-stp-bpdu single
end

Core-FSW2:
config switch stp settings
    set mclag-stp-bpdu single
end

Access-FSW1:
config switch stp settings
    set mclag-stp-bpdu single
end

Access-FSW2:
config switch stp settings
    set mclag-stp-bpdu single
end
```

结果验证

Cisco-SW1

1. 在思科交换机 Cisco-SW1 上查看聚合状态。

```
Cisco-SW1#show interfaces port-channel 10 etherchannel

Port-channel10   (Primary aggregator)
Age of the Port-channel   = 0d:00h:35m:49s
Logical slot/port   = 10/10          Number of ports = 2
HotStandBy port = null
Passive port list   = Fa2/0/3 Fa2/0/4 Fa2/0/5
```

```
Port state          = Port-channel L3-Ag Ag-Inuse
Protocol            =   LACP
Port security       = Disabled
Ports in the Port-channel:
Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Fa2/0/3  Active             0
  0     00     Fa2/0/4  Active             0
Time since last port bundled:    0d:00h:28m:41s    Fa2/0/4

Cisco-SW1#show interfaces port-channel 10

Port-channel10 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 001c.b0c7.9741 (bia 001c.b0c7.9741)
  Internet address is 192.168.10.101/24
  MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Full-duplex, 100Mb/s, link type is auto, media type is unknown
  input flow-control is off, output flow-control is unsupported
  Members in this channel: Fa2/0/3 Fa2/0/4
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:30:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 6000 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4264 packets input, 1406340 bytes, 0 no buffer
     Received 4264 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 3034 multicast, 0 pause input
     0 input packets with dribble condition detected
     680 packets output, 100803 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

2. 在思科交换机 Cisco-SW1 上访问 FortiGate 的 VLAN10。

```
Cisco-SW1#ping 192.168.10.1 repeat 100

Sending 100, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/11/1007 ms
```

Cisco-SW2

1. 在思科交换机 Cisco-SW2 上查看聚合状态。

```
Cisco-SW2#show interfaces port-channel 20 etherchannel

Port-channel20   (Primary aggregator)
Age of the Port-channel   = 0d:00h:38m:50s
Logical slot/port   = 10/20          Number of ports = 1
HotStandBy port = null
Passive port list   = Fa2/0/6
Port state          = Port-channel L3-Ag Ag-Inuse
Protocol            =   LACP
Port security       = Disabled
Ports in the Port-channel:
Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Fa2/0/6  Active             0
Time since last port bundled:    0d:00h:10m:47s    Fa2/0/6
```

```
Cisco-SW2#show interfaces port-channel 20

Port-channel20 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 001c.b0c7.9744 (bia 001c.b0c7.9744)
  Internet address is 192.168.20.101/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Full-duplex, 100Mb/s, link type is auto, media type is unknown
  input flow-control is off, output flow-control is unsupported
  Members in this channel: Fa2/0/6
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:10:57, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     729 packets input, 249974 bytes, 0 no buffer
     Received 729 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 542 multicast, 0 pause input
     0 input packets with dribble condition detected
     126 packets output, 18528 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

2. 在思科交换机 Cisco-SW2 上访问 FortiGate 的 VLAN20。

```
Cisco-SW2#ping 192.168.20.1 repeat 100

Sending 100, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/11/1007 ms
```

注意事项

1. 只能是堆叠交换机才可以跨交换机聚合，不同的堆叠交换机组，不支持配置为聚合接口。
2. 如果是与 FortiSwith 对接，则是自动 MCLAG 聚合。
3. 如果是和第三方设备对接，比如交换机、路由器、友商防火墙、服务器等等对接的话，推荐使用 LACP Active 方式动态聚合。

FortiLink 排障

- 快速查找异常流量

快速查找异常流量

Fortilink 部署场景下， 如果网络中有大流量的异常流量（如大量广播、未知组播影响到网络正常运行），可以通过下面的步骤来查找异常流量的来源。

排查举例

1. 在 FortiGate 抓包（sniffer 4），发现网络中有大量的 ARP 广播流量（抓包时指定抓包数量，可以看到在某一段时间内哪些报文出现较多）。

```
FortiGate # diagnose sniffer packet any "" 4 100 l
interfaces=[any]
filters=[]
2023-05-30 09:16:42.833594 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833594 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833595 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833595 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833596 default in arp who-has 183.230.156.254 tell 183.230.156.148
2023-05-30 09:16:42.833596 default in arp who-has 183.230.20.126 tell 183.230.20.89
2023-05-30 09:16:42.833597 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833597 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833598 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833598 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833598 default in arp who-has 183.230.156.254 tell 183.230.156.148
2023-05-30 09:16:42.833598 default in arp who-has 58.254.217.97 tell 58.254.217.99
2023-05-30 09:16:42.833600 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833601 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833601 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833602 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833602 default in arp who-has 192.168.100.118 tell 192.168.100.99
2023-05-30 09:16:42.833602 default in arp who-has 183.230.156.254 tell 183.230.156.148
2023-05-30 09:16:42.833604 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833604 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833605 default in arp who-has 113.204.198.1 tell 113.204.198.2
2023-05-30 09:16:42.833607 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833608 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833608 default in arp who-has 183.230.156.254 tell 183.230.156.148
2023-05-30 09:16:42.833611 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833612 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833612 default in arp who-has 192.168.110.99 tell 192.168.110.7
2023-05-30 09:16:42.833618 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833618 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833619 port6 in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833619 default in arp who-has 183.230.156.254 tell 183.230.156.148
2023-05-30 09:16:42.833619 fortilink in 802.1Q vlan#1 P0
2023-05-30 09:16:42.833620 default in arp who-has 183.230.20.126 tell 183.230.20.89
```

2. 在 FortiGate 使用 sniffer 6 抓 ARP 包，确认异常 ARP 广播流量的源 MAC，源 MAC 可以在 sniffer 6 的输出中直接看到，在“0x0000”行的第 4~6 段，如下所示，“ac71 2ed2 35f2”和“0009 0f09 3c0e”。

```
FortiGate # diagnose sniffer packet any "arp" 6 100 l
interfaces=[any]
filters=[arp]
2023-05-30 09:16:50.423541 default in arp who-has 58.254.217.97 tell 58.254.217.99
0x0000   ffff ffff ffff **→ac71 2ed2 35f2←** 0806 0001        .......q..5.....
0x0010   0800 0604 0001 ac71 2ed2 35f2 3afe d963        .......q..5.:..c
0x0020   0000 0000 0000 3afe d961 0000 0000 0000        ......:..a......
0x0030   0000 0000 0000 0000 0000 0000                  ............

2023-05-30 09:16:50.423554 default in arp who-has 222.178.231.33 tell 222.178.231.34
0x0000   ffff ffff ffff **→0009 0f09 3c0e←** 0806 0001        ..........<.....
0x0010   0800 0604 0001 0009 0f09 3c0e deb2 e722        ..........<...."
0x0020   0000 0000 0000 deb2 e721 0000 0000 0000        .........!......
0x0030   0000 0000 0000 0000 0000 0000                  ............

2023-05-30 09:16:50.423556 default in arp who-has 222.178.231.33 tell 222.178.231.34
0x0000   ffff ffff ffff 0009 0f09 3c0e 0806 0001        ..........<.....
0x0010   0800 0604 0001 0009 0f09 3c0e deb2 e722        ..........<...."
0x0020   0000 0000 0000 deb2 e721 0000 0000 0000        .........!......
0x0030   0000 0000 0000 0000 0000 0000                  ............
```

```
2023-05-30 09:16:50.423560 default in arp who-has 58.254.217.97 tell 58.254.217.99
0×0000   ffff ffff ffff ac71 2ed2 35f2 0806 0001        .......q..5.....
0×0010   0800 0604 0001 ac71 2ed2 35f2 3afe d963        .......q..5.:..c
0×0020   0000 0000 0000 3afe d961 0000 0000 0000        ......:..a......
0×0030   0000 0000 0000 0000 0000 0000                  ............

2023-05-30 09:16:50.423563 default in arp who-has 58.254.217.97 tell 58.254.217.99
0×0000   ffff ffff ffff ac71 2ed2 35f2 0806 0001        .......q..5.....
0×0010   0800 0604 0001 ac71 2ed2 35f2 3afe d963        .......q..5.:..c
0×0020   0000 0000 0000 3afe d961 0000 0000 0000        ......:..a......
0×0030   0000 0000 0000 0000 0000 0000                  ............

2023-05-30 09:16:50.423574 default in arp who-has 58.254.217.97 tell 58.254.217.99
0×0000   ffff ffff ffff ac71 2ed2 35f2 0806 0001        .......q..5.....
0×0010   0800 0604 0001 ac71 2ed2 35f2 3afe d963        .......q..5.:..c
0×0020   0000 0000 0000 3afe d961 0000 0000 0000        ......:..a......
0×0030   0000 0000 0000 0000 0000 0000                  ............

2023-05-30 09:16:50.423584 default in arp who-has 58.254.217.97 tell 58.254.217.99
0×0000   ffff ffff ffff ac71 2ed2 35f2 0806 0001        .......q..5.....
0×0010   0800 0604 0001 ac71 2ed2 35f2 3afe d963        .......q..5.:..c
0×0020   0000 0000 0000 3afe d961 0000 0000 0000        ......:..a......
0×0030   0000 0000 0000 0000 0000 0000                  ............
```

3. 查看“0009 0f09 3c0e”这个 MAC 地址是接入在哪个交换机的哪个端口，可以看到这个源 MAC 地址是从交换机 S548DN4K16000312 的 port34 口学习到的。

```
FortiGate # diagnose switch-controller mac-cache show | grep 00:09:0f:09:3c:0e -B 50
.........................................
managed-switch: S548DN4K16000312 vfid:   0
running-clients:
VLANID   PORTID   ACCESS   MAC               LAST-SEEN(secs ago) INTF-NAME
1        34       accept   00:09:0f:09:3c:0a  52                 port34
```

4. 在 FortiLink 成员交换机 S548DN4K16000312 上检查流量，发现这个交换机的 port34 口和 port35 口的流量异常。

```
S548DN4K16000312 # diagnose switch physical-ports linerate up
```

```
  port12 |          94275130490 |   171.2255 Mbps ||                56642 |     0.0000 Mbps |
  port13 |          94275010153 |   171.1271 Mbps ||                56642 |     0.0000 Mbps |
  port14 |          94274885486 |   171.0361 Mbps ||                56642 |     0.0000 Mbps |
  port15 |          94274902209 |   170.9338 Mbps ||                56642 |     0.0000 Mbps |
  port16 |          94274813829 |   170.8185 Mbps ||                56642 |     0.0000 Mbps |
  port17 |          94274911097 |   170.4430 Mbps ||                56642 |     0.0000 Mbps |
  port18 |          94274867710 |   170.3595 Mbps ||                56642 |     0.0000 Mbps |
  port19 |          94274709770 |   170.3216 Mbps ||                56642 |     0.0000 Mbps |
  port20 |          94274769282 |   170.2678 Mbps ||                56642 |     0.0000 Mbps |
  port22 |           2565696451 |     0.0410 Mbps ||              2757465 |     0.0000 Mbps |
  port23 |           2555226658 |     0.0127 Mbps ||               144944 |     0.0000 Mbps |
  port24 |         112619791276 |   170.3063 Mbps ||              1113452 |     0.0012 Mbps |
  port25 |         112600069903 |   170.1315 Mbps ||                88603 |     0.0000 Mbps |
  port26 |         112600044067 |   170.0476 Mbps ||                88603 |     0.0000 Mbps |
  port27 |         112599917226 |   169.9863 Mbps ||                88603 |     0.0000 Mbps |
  port28 |         112599916735 |   169.9153 Mbps ||                88603 |     0.0000 Mbps |
  port29 |         112599757702 |   169.8759 Mbps ||                88603 |     0.0000 Mbps |
  port30 |         112599608054 |   169.8013 Mbps ||                88603 |     0.0000 Mbps |
  port31 |         112599588686 |   169.8915 Mbps ||                88618 |     0.0000 Mbps |
  port32 |         112599546651 |   169.9003 Mbps ||                88603 |     0.0000 Mbps |
  port33 |         112600620342 |   169.9212 Mbps ||                88603 |     0.0000 Mbps |
  port34 |         194843122934 |   140.2551 Mbps ||         169016453213 |   722.9183 Mbps |
  port35 |         192140484212 |   722.9053 Mbps ||         171691896046 |   140.2281 Mbps |
  port36 |             13002345 |     0.0202 Mbps ||                88602 |     0.0000 Mbps |
  port39 |             10409162 |     0.0021 Mbps ||                69124 |     0.0000 Mbps |
  port40 |         112613197123 |   169.8033 Mbps ||                   13 |     0.0000 Mbps |
  port41 |         112610491519 |   169.8336 Mbps ||              1078842 |     0.0000 Mbps |
  port42 |         112613831915 |   169.8294 Mbps ||             57731789 |     0.0000 Mbps |
  port43 |         112612713061 |   169.8099 Mbps ||            821887254 |     0.0233 Mbps |
  port44 |             11529838 |     0.0091 Mbps ||                93459 |     0.0000 Mbps |
  port45 |         112609755277 |   169.8417 Mbps ||                   18 |     0.0000 Mbps |
  port46 |         112602860412 |   169.7973 Mbps ||              1526419 |     0.0000 Mbps |
  port47 |         112604799750 |   169.7663 Mbps ||             44337052 |     0.0039 Mbps |
  port48 |         112605154331 |   169.7150 Mbps ||           1180979181 |     0.0323 Mbps |
  port51 |          93309196191 |   177.7362 Mbps ||          42263111161 |     0.4944 Mbps |
  port52 |         123512702971 |   177.6112 Mbps ||           8140918167 |     0.3305 Mbps |
internal |            149248028 |     0.2892 Mbps ||             13255285 |     0.0425 Mbps |
-------------------------------------------------------------------------------------------
                                |  6843.2485 Mbps ||                      |   864.0743 Mbps |
```

5. 查找异常的流量的来源设备，通过 LLDP 可以查询到接入的设备信息，从而确定异常流量的来源设备。

```
S548DN4K16000312 # get switch lldp neighbors-summary
......
  port34      Up       **→FortiGate-601F-down-Test←**  120   BR          -         port13
  port35      Up       **→FortiGate-601F-down-Test←**  120   BR          -         port14
......
```

VDOM

- VDOM 基本配置
- VDOM 使用
- VDOM 间通信
- VDOM 无法删除的处理方法

VDOM 基本配置

VDOM 简介

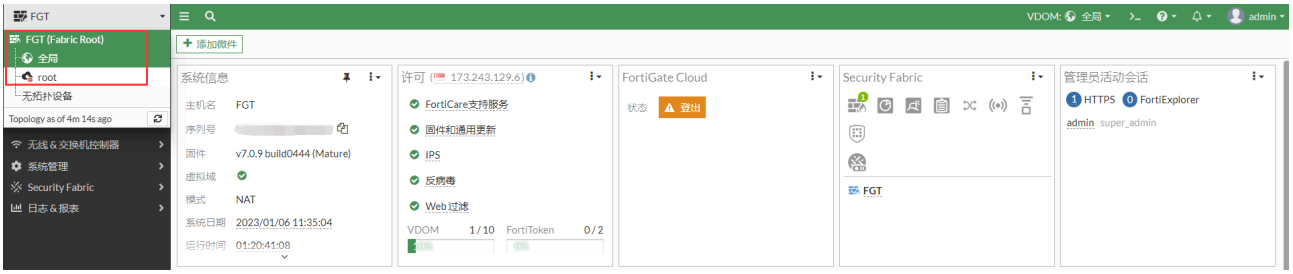
虚拟域 (VDOM)，即虚拟防火墙，是将一台物理的 FortiGate 划分为两个或两个以上的可以独 立运行的虚拟防火墙的技术，虚拟域技术可以将 FortiGate 划分为多个不同的 NAT/透明模式多个虚拟 FortiGate，为每一个虚拟域提供独立的安全策略，路由及 VPN 配置等。

开启 VDOM

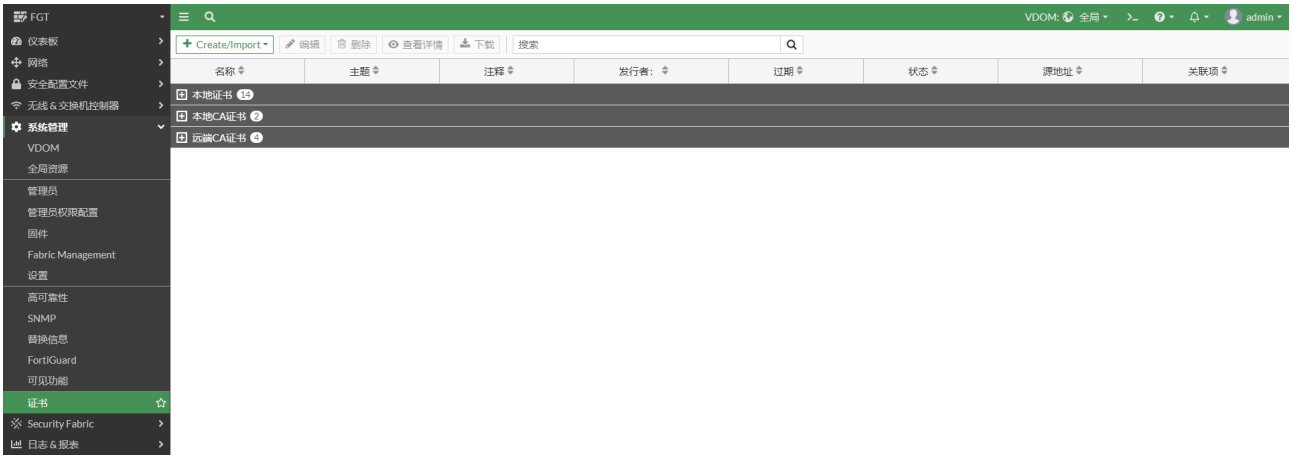
在 config system global 下开启 vdom，vdom 模式选择 multi-vdom，开启 vdom 后需要重新登录 FortiGate。

```
config system global
    set vdom-mode multi-vdom
end
You will be logged out for the operation to take effect.
Do you want to continue? (y/n)y
```

开启 vdom 后，FortiGate 默认将设备分为全局域和 root vdom。可以直接点击“全局”或者“root”来切换全局域和 root vdom。



全局用于所有 vdom 都共享的配置，如 HA，SNMP，证书等。



使用 CLI 配置时，需要使用 config global 进入全局模式，再配置具体的功能。如配置 snmp：

```
config global
    config system snmp sysinfo
        set status enable
    end
end
```

root vdom 是一个默认的虚拟防火墙。root vdom 默认是管理 vdom，用于和 FortiGuard 通信来更新 license，特征库等。



使用 CLI 配置时，需要使用 config vdom 进入对应的 vdom，在配置具体的功能，如配置防火墙策略：

```
config vdom
    edit root
    config firewall policy
        edit 3
            set name "test"
            set srcintf "port5"
            set dstintf "port2"
            set action accept
            set srcaddr "all"
            set dstaddr "all"
            set schedule "always"
            set service "ALL"
            set nat enable
        next
    end
end
```

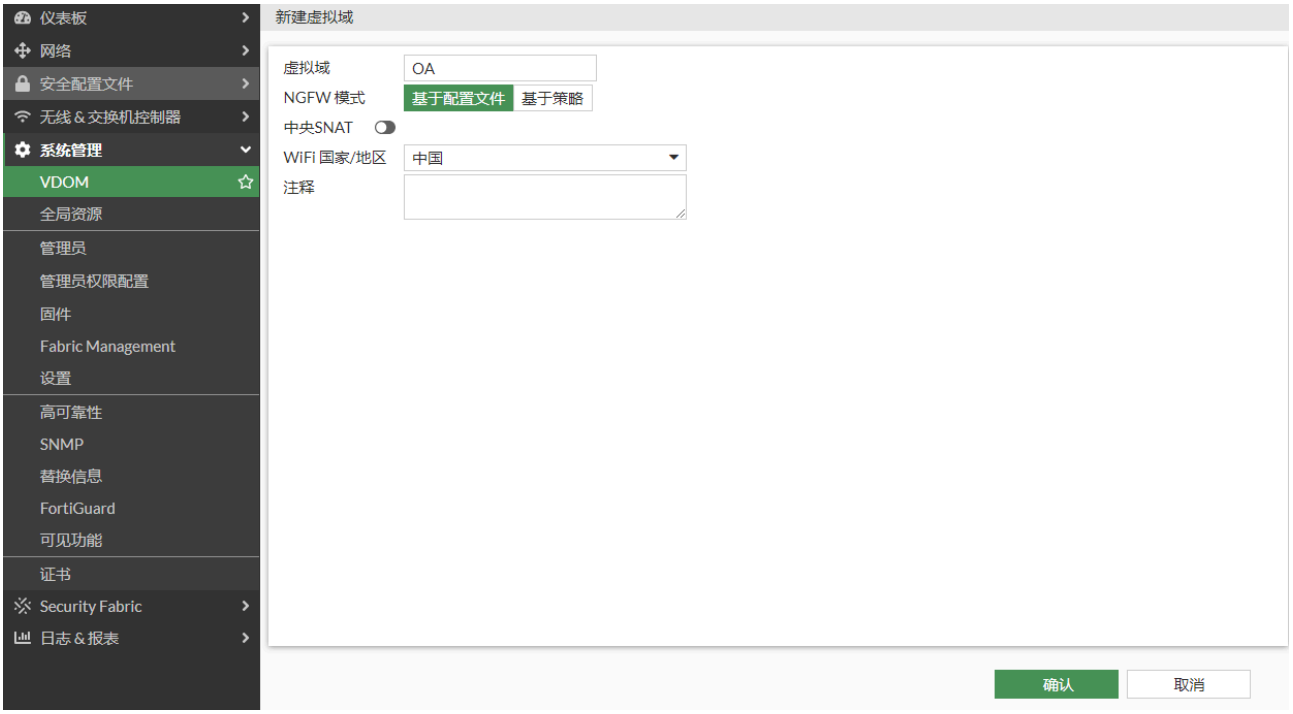
创建 vdom

1.GUI 创建 vdom

在全局中，选择“系统管理”



创建 OA vdom，用于独立的 OA 业务。创建的 vdom 默认是 NAT 模式的。



点击确认后，创建完成。



2.CLI 创建 vdom

进入 vdom，edit vdom 名称就会直接创建一个 vdom，默认是 NAT 模式的 vdom。

```
config vdom
    edit Database
end
```

创建完成后，可以在 GUI 界面查看。



可以避免 CLI 误创建 vdom，可以将 vdom 创建提示打开。

```
config global
    config system global
        set edit-vdom-prompt enable
    end
end
```

再次创建 vdom 时，CLI 会提示。

```
config vdom
edit Web

The input VDOM name doesn't exist.
Do you want to create a new VDOM?
```

```
Please press 'y' to continue, or press 'n' to cancel. (y/n)y

current vf=Web:5

end
```

3.将 vdom 改为透明模式

创建 vdom 时默认是 NAT 模式，需要通过 CLI 将 vdom 修改为透明模式。

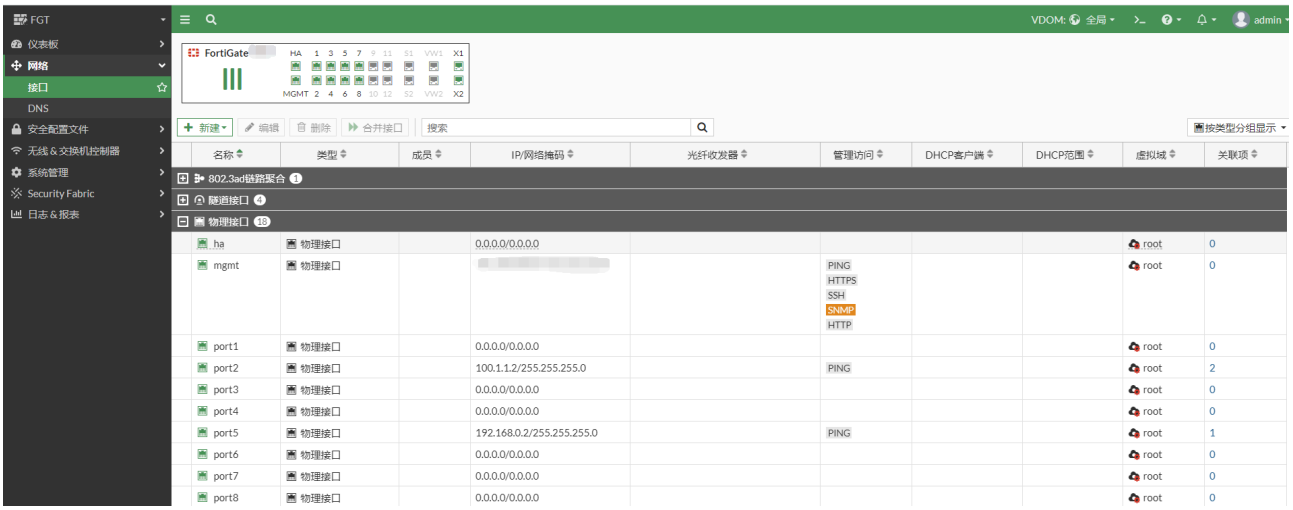
> **重要**
>
> 如果透明模式是一进一出的环境，建议使用虚拟接口对。

```
config vdom
    edit Web
    config system settings
        set opmode transparent
        set manageip 1.1.1.1/255.255.255.0
        set gateway 1.1.1.254
    end
end
```

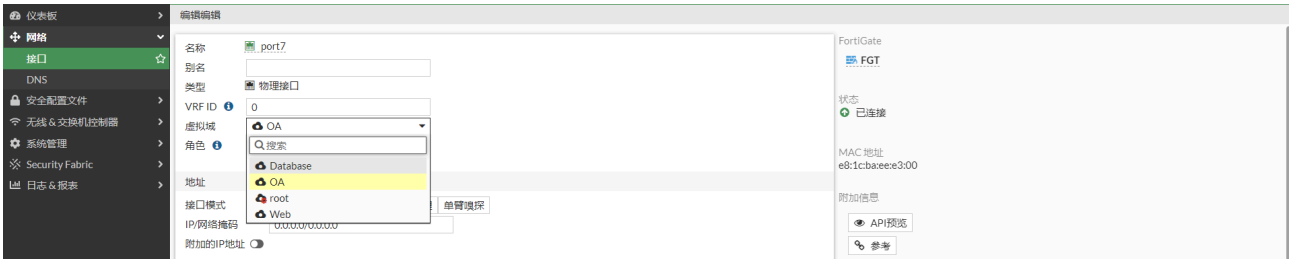
将接口划入 vdom

1.GUI 方式将接口划入 vdom

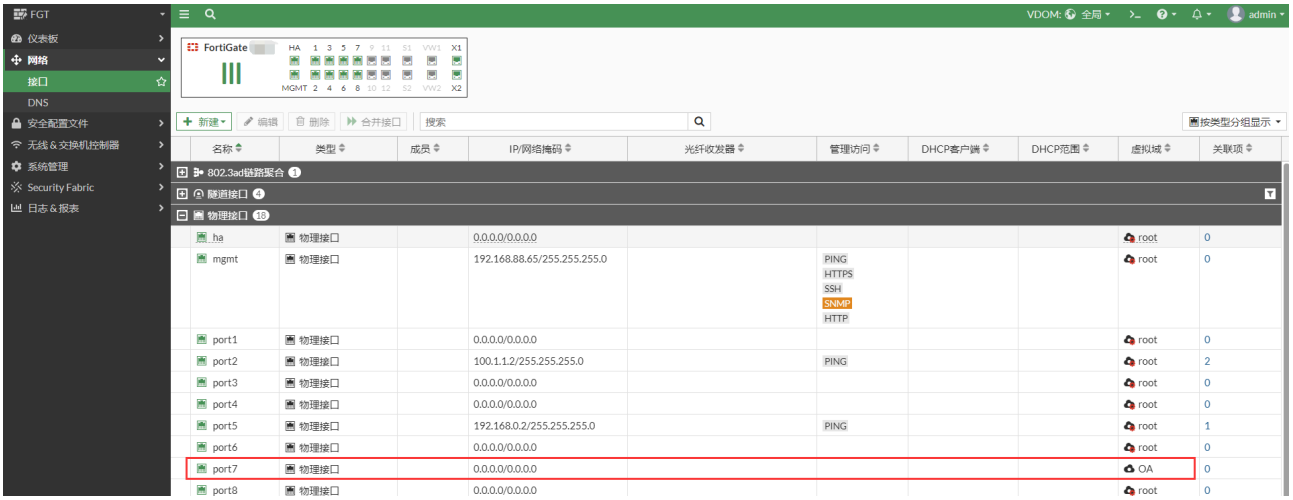
切换到全局，选择“网络”-->”接口“。



将接口划入 OA vdom。



可以看到 OA 已经划入到 OA vdom。



切换到 OA vdom 查看接口。



2.CLI 方式将接口划入 vdom

命令行如下：

```
config global
    config system interface
        edit port8
            set vdom OA
        next
    end
end
```

GUI 查看 OA vdom 的接口。（port7 是上一个操作划入 OA vdom 的）

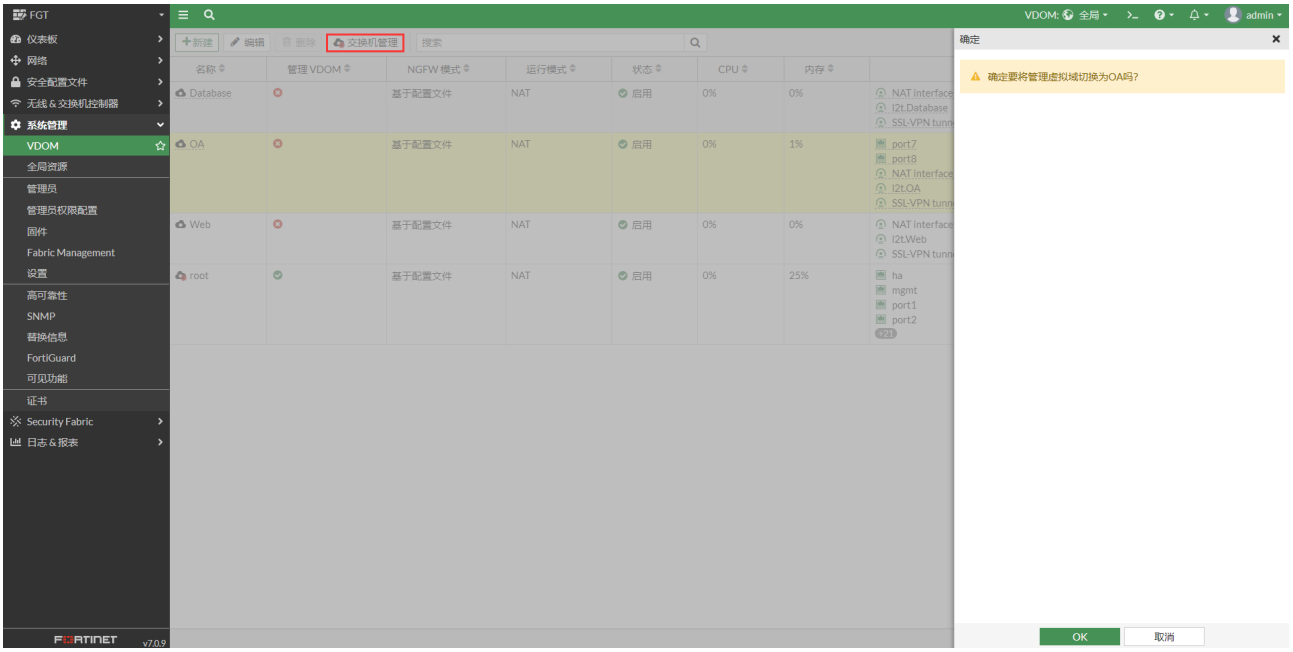


修改管理 vdom

管理 vdom 用于和 FortiGuard 通信来更新 license，特征库等，因此需要连接 Internet；也可以连接 FortiManager 更新，则 FortiManager 与 FortiGuard 通信。

1.GUI 修改管理 vdom

切换到全局，选择“系统管理”-->“VDOM”，选择 vdom，点击“交换机管理”切换为管理 VDOM，点击“OK”。



可以看到 OA 已设置为管理 vdom。



2.CLI 修改管理 vdom

```
config global
    config system global
        set management-vdom OA
    end
end
```

删除 vdom

1.GUI 删除 vdom

在删除 vdom 前，需要把 vdom 中的接口以及相关的配置都移除，在没有完全移除之前，是不能删除该 vdom 的。如 OA vdom。



选择“网络”-->“接口”，将接口划入到其他的 vdom。



当 OA 中的接口及其他相关的配置都移除后，OA vdom 可以删除。



点击“删除”。



确认后，OA vdom 移除。



2.CLI 删除 vdom

在删除 vdom 前，需要把 vdom 中的接口以及相关的配置都移除，在没有完全移除之前，是不能删除该 vdom 的。

```
config vdom
    delete Database
end
```

GUI 查看该 vdom 已经删除。



关闭 vdom

将 FortiGate 的其他 vdom 都删除后，才可以关闭 vdom。



CLI 关闭 vdom，关闭 vdom 后，需要重新登录。

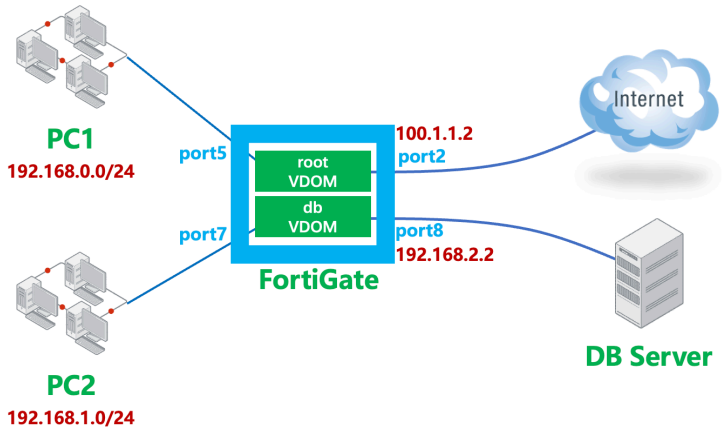
```
config global
    config system global
        set vdom-mode no-vdom
    end

You will be logged out for the operation to take effect.
Do you want to continue? (y/n)y
```

VDOM 使用

网络需求

客户上网的业务和内部数据的业务分离。将 FortiGate 划分两个不同的 vdom，一个 root vdom 用于上网，一个 db vdom 用于内网数据库访问。

##网络拓扑



配置步骤

1.创建 vdom

开启 VDOM

```
config system global
    set vdom-mode multi-vdom
end
```

创建 VDOM



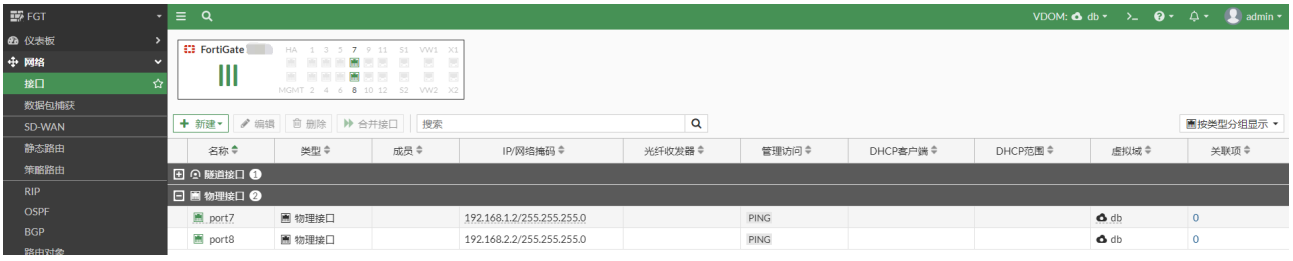
2.将接口划入 VDOM

所有的接口默认都在 root VDOM 中，因此只需要将 port7 和 port8 划入 db vdom 即可。

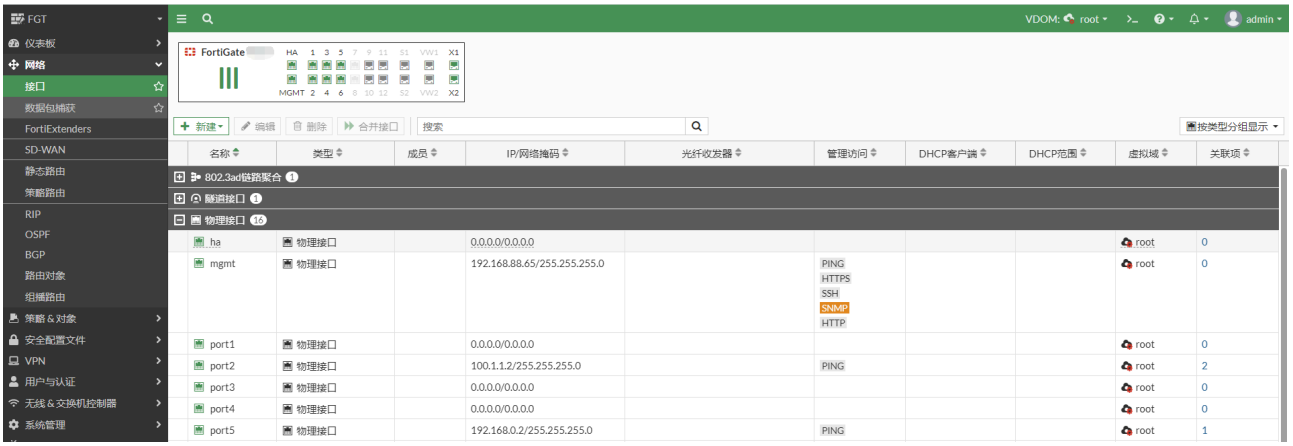




查看 db vdom 中的接口。

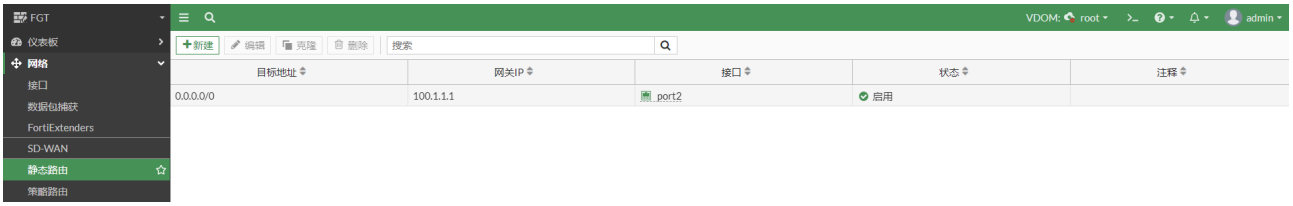


查看 root vdom 中的接口。

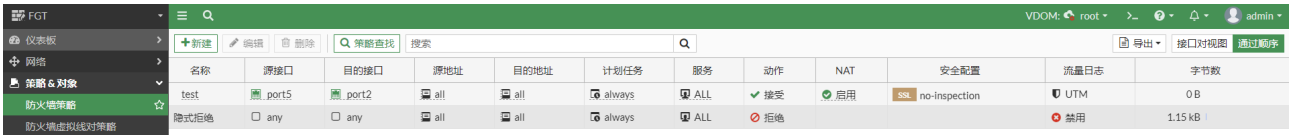


2.root vdom 配置路由和策略

配置路由。



配置策略。



3.db vdom 配置路由和策略

配置路由。



配置策略。



业务测试

root vdom 测试

1. 查看 root vdom 路由表

```
进入root vdom
#config vdom
#edit root

# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
```

2.业务测试

PC1 ping 114.114.114.114

```
PC1# ping 114.114.114.114
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.
64 bytes from 114.114.114.114: icmp_seq=1 ttl=64 time=23.2 ms
64 bytes from 114.114.114.114: icmp_seq=2 ttl=82 time=23.3 ms
64 bytes from 114.114.114.114: icmp_seq=3 ttl=79 time=23.3 ms
64 bytes from 114.114.114.114: icmp_seq=4 ttl=78 time=23.4 ms
64 bytes from 114.114.114.114: icmp_seq=5 ttl=77 time=23.1 ms
```

FortiGate 抓包

```
进入root vdom
#config vdom
#edit root

抓包
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
1.674117 port5 in 192.168.0.10 → 114.114.114.114: icmp: echo request
1.674142 port2 out 100.1.1.2 → 114.114.114.114: icmp: echo request
1.697234 port2 in 114.114.114.114 → 100.1.1.2: icmp: echo reply
1.697242 port5 out 114.114.114.114 → 192.168.0.10: icmp: echo reply
```

FortiGate 查看会话

```
进入vdom
#config vdom
#edit root

查看会话
#diagnose sys session filter proto 1
#diagnose sys session filter vd-name root
#diagnose sys session list

该会话的vdom索引是0，即vd=0，通过diagnose sys vd list查看vdom索引，root vdom的索引是0
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin→sink: org pre→post, reply pre→post dev=13→10/10→13 gwy=100.1.1.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:3646→114.114.114.114:8(100.1.1.2:64062)
hook=pre dir=reply act=dnat 114.114.114.114:64062→100.1.1.2:0(192.168.0.10:3646)
misc=0 policy_id=3 pol_uuid_idx=515 auth_info=0 chk_client_info=0 vd=0
serial=0003413a tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4000c00 ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=0/0, epid=150/156, ipid=156/150, vlan=0×0000/0×0000
vlifid=156/150, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/3

进入全局，查看vdom索引
#config global
# diagnose sys vd list | grep name=root
name=root/root index=0 enabled fib_ver=36 rpdb_ver=2 use=168 rt_num=48 asym_rt=0 sip_helper=0, sip_nat_trace=1, mc_fwd=0,
mc_ttl_nc=0, tpmc_sk_pl=0
```

db vdom 测试

1.查看路由表

```
进入db vdom
#config vdom
#edit db

查看路由表
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 192.168.2.1, port8, [1/0]
C       192.168.1.0/24 is directly connected, port7
C       192.168.2.0/24 is directly connected, port8
```

2.业务测试

PC2 ping DB Server

```
PC2# ping 192.168.91.201
PING 192.168.91.201 (192.168.91.201) 56(84) bytes of data.
64 bytes from 192.168.91.201: icmp_seq=1 ttl=61 time=0.898 ms
64 bytes from 192.168.91.201: icmp_seq=2 ttl=61 time=1.05 ms
64 bytes from 192.168.91.201: icmp_seq=3 ttl=61 time=0.935 ms
64 bytes from 192.168.91.201: icmp_seq=4 ttl=61 time=0.687 ms

--- 192.168.91.201 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.687/0.892/1.050/0.134 ms
```

FortiGate 抓包

```
进入db vdom
#config vdom
#edit db

查看路由表
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
2.484117 port7 in 192.168.1.10 → 192.168.91.201: icmp: echo request
2.484133 port8 out 192.168.1.10 → 192.168.91.201: icmp: echo request
2.484914 port8 in 192.168.91.201 → 192.168.1.10: icmp: echo reply
2.485049 port7 out 192.168.91.201 → 192.168.1.10: icmp: echo reply
```

FortiGate 查看会话

```
进入db vdom
#config vdom
#edit db

查看会话
# diagnose sys session filter proto 1
# diagnose sys session filter vd-name db
# diagnose sys session list

该会话的vdom索引是3，即vd=3，通过diagnose sys vd list查看vdom索引，db vdom的索引是3
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
```

```
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=15->16/16->15 gwy=192.168.2.1/192.168.1.10
hook=pre dir=org act=noop 192.168.1.10:2605->192.168.91.201:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.91.201:2605->192.168.1.10:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=617 auth_info=0 chk_client_info=0 vd=3
serial=0003466c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4000c00 ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=0/0, epid=162/160, ipid=160/162, vlan=0×0000/0×0000
vlifid=160/162, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=3/2

进入全局，查看vdom索引
#config global
# diagnose sys vd list | grep name=db
name=db/db index=3 enabled fib_ver=23 rpdb_ver=0 use=62 rt_num=7 asym_rt=0 sip_helper=0, sip_nat_trace=1, mc_fwd=0,
mc_ttl_nc=0, tpmc_sk_pl=0
```

VDOM 间通信

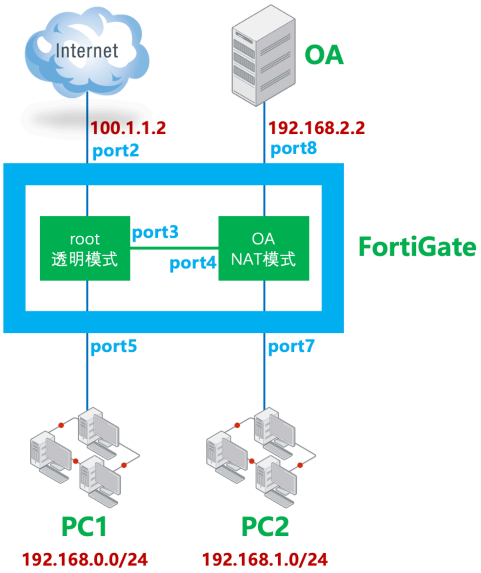
网络需求

OA 业务 vdom 的用户通过 Internet vdom 上网。

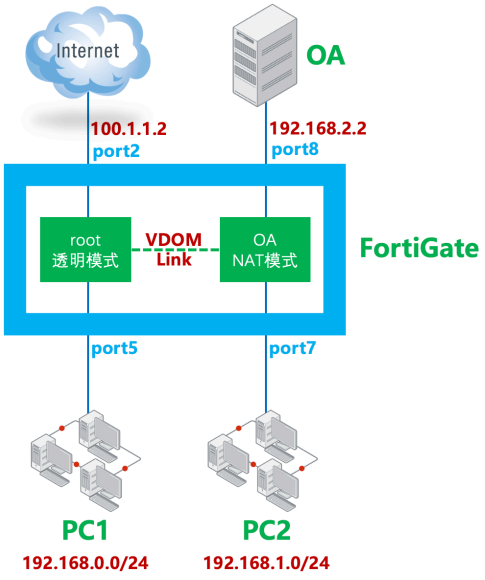
##网络拓扑

有两个方式可以让 VDOM 之间通信：

1. 通过物理线路连接；



2. 通过 NPU link 内部线路连接，部分硬件设备支持（也可以手动创建虚拟 link 进行连接，但无法被 NP 加速）。



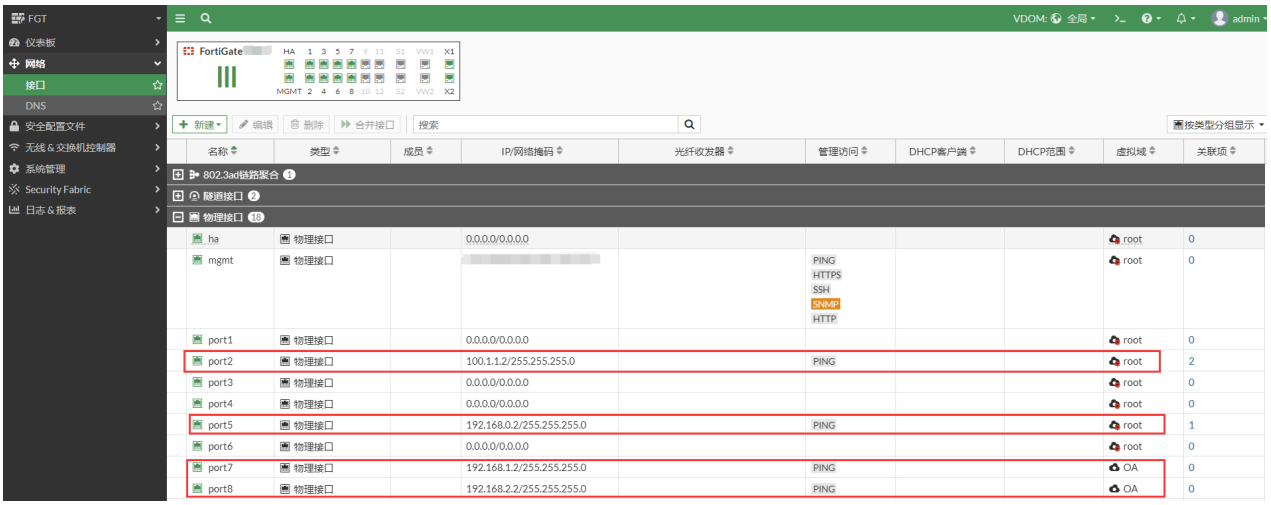
配置步骤

1. 开启 vdom 并创建 vdom。

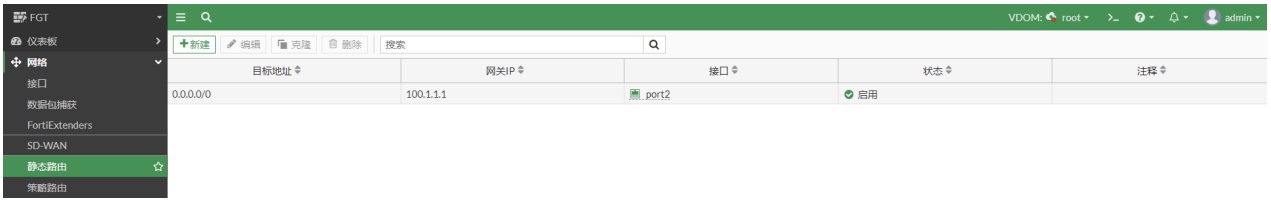


```
config system global
    set vdom-mode multi-vdom
end
```

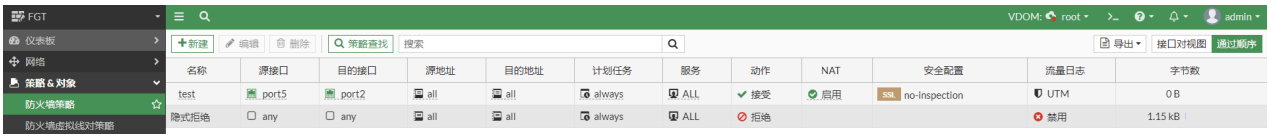
2. 将接口划入 vdom，所有的接口默认都在 root vdom 中，因此只需要将 port7 和 port8 划入 OA vdom 即可。



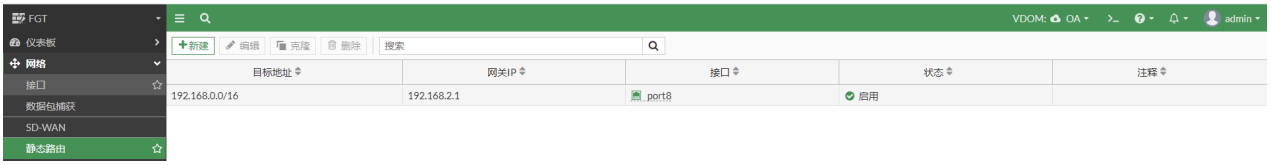
3. root vdom 配置路由和策略，这是 root vdom 自身业务转发的配置。配置路由。



4. 配置策略。



5. OA vdom 配置路由和策略，这是 OA vdom 自身业务转发的配置，配置路由，OA 的目的地址范围假设是 192.168.0.0/16。



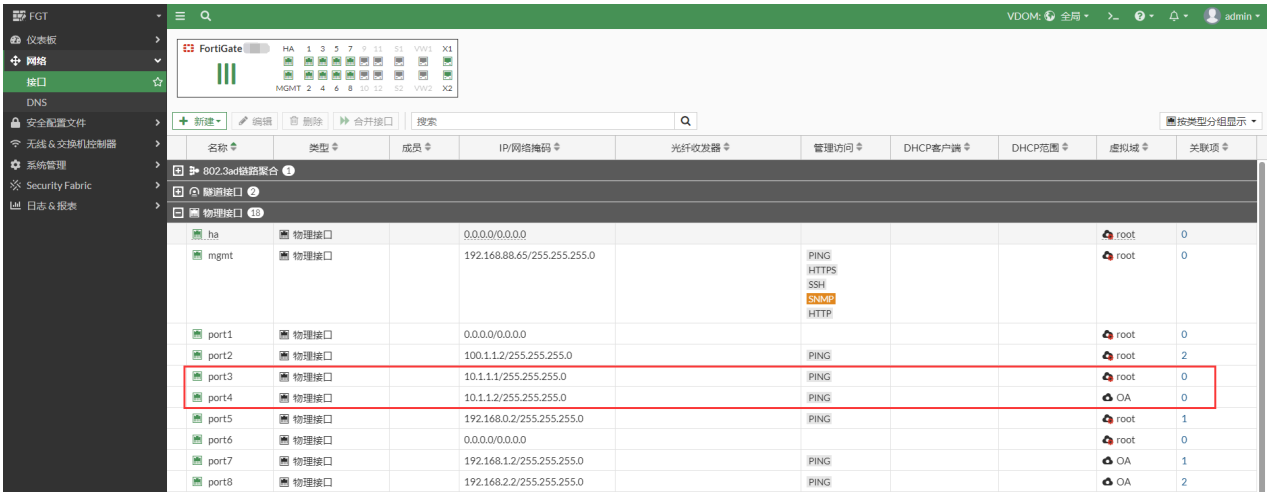
6. 配置策略。



VDOM 间通信配置

物理线路方式

1. 将接口划入 vdom，将 port3 划入 root vdom，将 port4 划入 OA vdom，配置 ip 地址，然后使用网线连接 port3 和 port4。



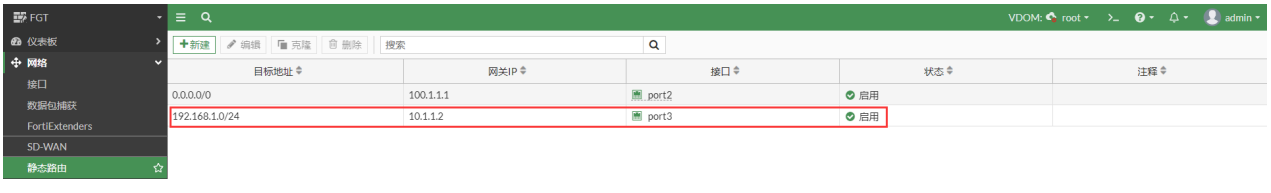
2. 测试 port3 和 port4 之间的连通性。进入 root vdom，从 root vdom ping OA vdom port4 接口地址。

```
进入root vdom，从root vdom ping OA vdom port4接口地址
#config vdom
#edit root
# execute ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2): 56 data bytes
64 bytes from 10.1.1.2: icmp_seq=0 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=1 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=3 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=4 ttl=255 time=0.0 ms

--- 10.1.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

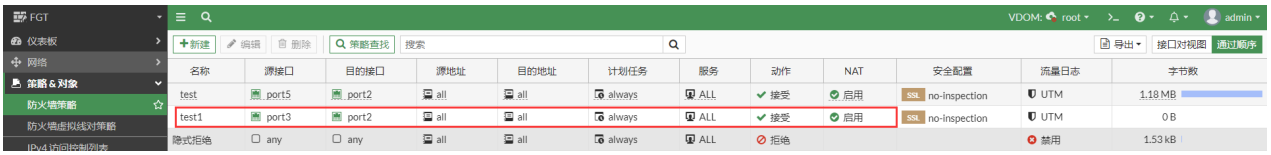
3. 进入 OA vdom，从 OA vdom Ping root vdom port3 接口地址。

```
#config vdom
#edit OA
#execute ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1): 56 data bytes
64 bytes from 10.1.1.1: icmp_seq=0 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=1 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=255 time=0.0 ms

--- 10.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

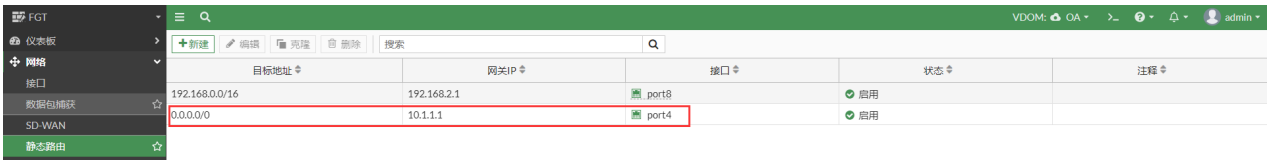
4. root vdom 配置路由和策略，配置路由，配置 root vdom 到 OA vdom 子网的回程路由。



5. 配置策略，允许 OA vdom 到 Internet 的流量。



6. OA vdom 配置路由和策略，配置路由，使 OA vdom 访问 Internet 的路由指向 root vdom。



7. 配置策略，允许 OA vdom 到 root vdom 的流量。



VDOM 间通信测试

1. PC2 访问 Internet。

```
PC2# ping 114.114.114.114
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.
64 bytes from 114.114.114.114: icmp_seq=1 ttl=69 time=23.5 ms
64 bytes from 114.114.114.114: icmp_seq=2 ttl=85 time=23.2 ms
64 bytes from 114.114.114.114: icmp_seq=3 ttl=86 time=23.2 ms
64 bytes from 114.114.114.114: icmp_seq=4 ttl=73 time=23.2 ms
```

2. FortiGate 抓包可以看出数据流穿过了 port3 和 port4 接口（在 root vdom 或者在 OA vdom 下抓包都是可以的）。

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
2.923717 port7 in 192.168.1.10 → 114.114.114.114: icmp: echo request
2.923771 port4 out 192.168.1.10 → 114.114.114.114: icmp: echo request
2.923782 port3 in 192.168.1.10 → 114.114.114.114: icmp: echo request
2.923794 port2 out 100.1.1.2 → 114.114.114.114: icmp: echo request
2.946967 port2 in 114.114.114.114 → 100.1.1.2: icmp: echo reply
2.946978 port3 out 114.114.114.114 → 192.168.1.10: icmp: echo reply
2.946989 port4 in 114.114.114.114 → 192.168.1.10: icmp: echo reply
2.947163 port7 out 114.114.114.114 → 192.168.1.10: icmp: echo reply
```

3. FortiGate 查看会话，ping 流量在 root vdom 和 OA vdom 都产生了会话。在 root vdom 或者在 OA vdom 下查看都可以看到会话，vd = 0 是 root vdom，vd = 3 是 OA vdom。

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=29 expire=31 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->10/10->11 gwy=100.1.1.1/10.1.1.2
hook=post dir=org act=snat 192.168.1.10:2691->114.114.114.114:8(100.1.1.2:63107)
hook=pre dir=reply act=dnat 114.114.114.114:63107->100.1.1.2:0(192.168.1.10:2691)
misc=0 policy_id=4 pol_uuid_idx=636 auth_info=0 chk_client_info=0 vd=0
serial=00036a12 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=150/152, ipid=152/150, vlan=0x0000/0x0000
```

```
vlifid=152/150, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=2/7

session info: proto=1 proto_state=00 duration=29 expire=31 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin→sink: org pre→post, reply pre→post dev=15→12/12→15 gwy=10.1.1.1/192.168.1.10
hook=pre dir=org act=noop 192.168.1.10:2691→114.114.114.114:8(0.0.0.0:0)
hook=post dir=reply act=noop 114.114.114.114:2691→192.168.1.10:0(0.0.0.0:0)
misc=0 policy_id=2 pol_uuid_idx=637 auth_info=0 chk_client_info=0 vd=3
serial=00036a11 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4000c00 ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=0/0, epid=154/160, ipid=160/154, vlan=0×0000/0×0000
vlifid=160/154, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=2/3
```

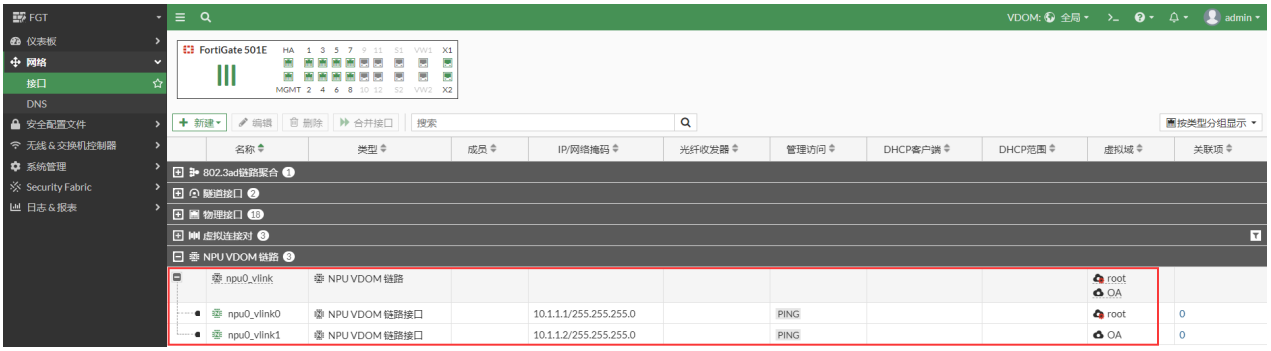
4. 进入全局，查看 vdom 索引。

```
#config global
# diagnose sys vd list  | grep name=root
name=root/root index=0 enabled fib_ver=49 rpdb_ver=2 use=162 rt_num=46 asym_rt=0 sip_helper=0, sip_nat_trace=1,
mc_fwd=0, mc_ttl_nc=0, tpmc_sk_pl=0

# diagnose sys vd list  | grep name=OA
name=OA/OA index=3 enabled fib_ver=18 rpdb_ver=0 use=65 rt_num=7 asym_rt=0 sip_helper=0, sip_nat_trace=1, mc_fwd=0,
mc_ttl_nc=0, tpmc_sk_pl=0
```

NPU link 方式

NPU link 方式和物理线路方式思路是一样的，只是接口不一样。

1. 将 NPU link 接口划入 vdom，中高端设备支持 NPU link 接口，NPU link 接口支持 NP 加速。可以使用 NPU link 接口进行 vdom 之间的通信。将 npu0_vlink0 划入 root vdom，将 npu0_vlink1 划入 OA vdom，并配置 IP 地址。



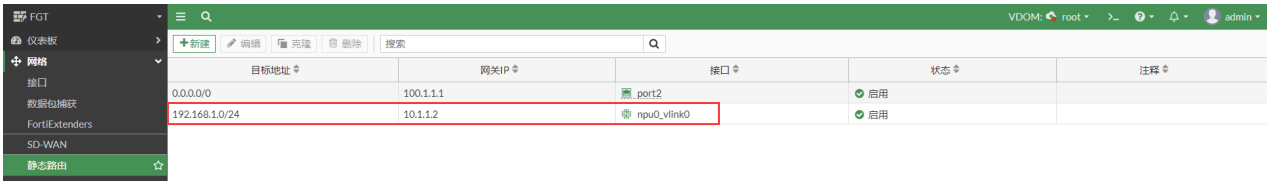
2. 测试 NPU link 之间的连通性，进入 root vdom，从 root vdom ping OA vdom npu_vlink1 接口地址。

```
#config vdom
#edit root
# execute ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2): 56 data bytes
64 bytes from 10.1.1.2: icmp_seq=0 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=1 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=3 ttl=255 time=0.0 ms
64 bytes from 10.1.1.2: icmp_seq=4 ttl=255 time=0.0 ms

--- 10.1.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

3. 进入 OA vdom，从 OA vdom ping root vdom npu_vlink0 接口地址。

```
#config vdom
#edit OA
#execute ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1): 56 data bytes
64 bytes from 10.1.1.1: icmp_seq=0 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=1 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=255 time=0.0 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=255 time=0.0 ms

--- 10.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

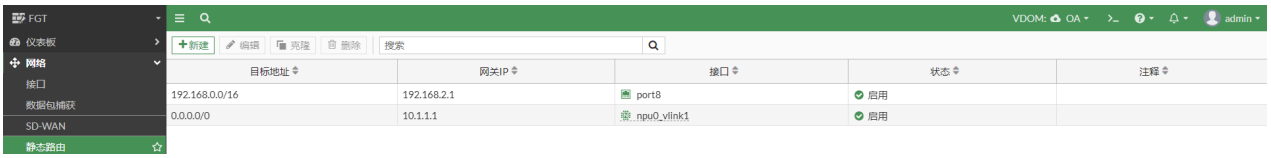
4. root vdom 配置路由和策略：配置路由，配置 root vdom 到 OA vdom 子网的回程路由。



5. 配置策略，允许 OA vdom 到 Internet 的流量。



6. OA vdom 配置路由和策略：配置路由，使 OA vdom 访问 Internet 的路由指向 root vdom。



7. 配置策略，允许 OA vdom 到 root vdom 的流量。



VDOM 间通信测试

1. PC2 ping 114.114.114.114。

```
PC2# ping 114.114.114.114
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.
64 bytes from 114.114.114.114: icmp_seq=1 ttl=69 time=23.5 ms
64 bytes from 114.114.114.114: icmp_seq=2 ttl=85 time=23.2 ms
64 bytes from 114.114.114.114: icmp_seq=3 ttl=86 time=23.2 ms
64 bytes from 114.114.114.114: icmp_seq=4 ttl=73 time=23.2 ms
```

2. FortiGate 抓包可以看出数据流穿过了 npu_vlink0 和 npu_vlink1 接口（在 root vdom 或者在 OA vdom 下抓包都是可以的）。

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
3.064987 port7 in 192.168.1.10 → 114.114.114.114: icmp: echo request
3.065007 npu0_vlink1 out 192.168.1.10 → 114.114.114.114: icmp: echo request
3.065012 npu0_vlink0 in 192.168.1.10 → 114.114.114.114: icmp: echo request
```

```
3.065024 port2 out 100.1.1.2 → 114.114.114.114: icmp: echo request
3.088172 port2 in 114.114.114.114 → 100.1.1.2: icmp: echo reply
3.088199 npu0_vlink0 out 114.114.114.114 → 192.168.1.10: icmp: echo reply
3.088205 npu0_vlink1 in 114.114.114.114 → 192.168.1.10: icmp: echo reply
3.088431 port7 out 114.114.114.114 → 192.168.1.10: icmp: echo reply
```

3. FortiGate 查看会话，ping 流量在 root vdom 和 OA vdom 都产生了会话，在 root vdom 或者在 OA vdom 下查看会话都是可以的, vd = 0 是 root vdom，vd = 3 是 OA vdom。

```
# diagnose sys session list

session info: proto=1 proto_state=00 duration=3 expire=57 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->10/10->7 gwy=100.1.1.1/10.1.1.2
hook=post dir=org act=snat 192.168.1.10:2733->114.114.114.114:8(100.1.1.2:63149)
hook=pre dir=reply act=dnat 114.114.114.114:63149->100.1.1.2:0(192.168.1.10:2733)
misc=0 policy_id=4 pol_uuid_idx=638 auth_info=0 chk_client_info=0 vd=0
serial=00037b32 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x91, offload=8/8, ips_offload=0/0, epid=150/142, ipid=132/150, vlan=0x0000/0x0000
vlifid=132/150, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/5

session info: proto=1 proto_state=00 duration=3 expire=57 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=15->8/8->15 gwy=10.1.1.1/192.168.1.10
hook=pre dir=org act=noop 192.168.1.10:2733->114.114.114.114:8(0.0.0.0:0)
hook=post dir=reply act=noop 114.114.114.114:2733->192.168.1.10:0(0.0.0.0:0)
misc=0 policy_id=2 pol_uuid_idx=637 auth_info=0 chk_client_info=0 vd=3
serial=00037b31 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x91/0x81, offload=8/8, ips_offload=0/0, epid=132/160, ipid=160/142, vlan=0x0000/0x0000
vlifid=160/142, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/1
```

4. 进入全局，查看 vdom 索引。

```
# config global
# diagnose sys vd list  | grep name=root
name=root/root index=0 enabled fib_ver=49 rpdb_ver=2 use=162 rt_num=46 asym_rt=0 sip_helper=0, sip_nat_trace=1,
mc_fwd=0, mc_ttl_nc=0, tpmc_sk_pl=0

# diagnose sys vd list  | grep name=OA
name=OA/OA index=3 enabled fib_ver=18 rpdb_ver=0 use=65 rt_num=7 asym_rt=0 sip_helper=0, sip_nat_trace=1, mc_fwd=0,
mc_ttl_nc=0, tpmc_sk_pl=0
```

VDOM 无法删除的处理方法

问题现象

删除 VDOM 时，有时会遇到删除了 VDOM 的所有关联对象后，仍无法删除空 VDOM，如下 VDOM1 无法删除。

```
FortiGate # config vdom
FortiGate (vdom) # delete VDOM1
Domain VDOM1: used by interface, can not delete
Command fail. Return code -23
```

解决方法

1. 通过查 VDOM 相关对象，只有系统自建的对象的引用，但还是无法删除 VDOM。

```
FortiGate (global) # diagnose sys cmdb refcnt show system.vdom.name VDOM1
entry used by table system.interface:name 'l2t.VDOM1'
entry used by table system.interface:name 'naf.VDOM1'
entry used by table system.interface:name 'ssl.VDOM1'
entry used by child table gui-dashboard:id '46' of entry used by child table gui-dashboard:id '47' of
entry used by child table gui-dashboard:id '48' of entry used by child table gui-dashboard:id '49' of
entry used by child table gui-dashboard:id '50' of entry used by child table gui-dashboard:id '51' of
entry used by child table gui-dashboard:id '52' of entry used by child table gui-dashboard:id '53' of
entry used by child table gui-dashboard:id '54' of entry used by child table gui-dashboard:id '55' of
entry used by child table gui-dashboard:id '56' of table system.admin:name 'admin'
entry used by child table gui-dashboard:id '22' of entry used by child table gui-dashboard:id '23' of
entry used by child table gui-dashboard:id '24' of entry used by child table gui-dashboard:id '25' of
entry used by child table gui-dashboard:id '26' of entry used by child table gui-dashboard:id '27' of
entry used by child table gui-dashboard:id '28' of entry used by child table gui-dashboard:id '29' of
entry used by child table gui-dashboard:id '30' of entry used by child table gui-dashboard:id '31' of
table system.admin:name 'nap_fw_admin'
entry used by table system.vdom-property:name 'VDOM1'
```

2. 通过如下命令可以重置 VDOM 的引用关系：

```
diagnose sys cmdb refcnt reset system.vdom.name VDOM1（VDOM名称）
```

3. 重置引用关系后，可以删除空 VDOM VDOM1。

```
FortiGate # config vdom
FortiGate (vdom) # delete VDOM1
FortiGate (vdom) #
```

HA 双机热备

HA 工作模式和配置要求

HA 选取主设备的方法

HA 典型基础配置（FGCP）

HA 虚拟 MAC

HA-Cluster 不间断升级

HA 共享单一 License

HA 的 ping server 配置

HA 配置命令集

异步流量负载分担（FGSP）

HA-Cluster 的网管

- HA 的网管介绍
- HA 集群普通管理
- HA 集群带外管理

  独立管理接口　独立管理 VDOM　单独指定用于更新的 VDOM

HA 常见问题

- HA 同步机制和故障诊断
- HA 无法更新特征库/License

HA 工作模式和配置要求

工作模式

Active-Passive (A - P) 模式

1. 集群中的所有防火墙必须工作在同一个模式下。可以对运行中的 HA 集群进行模式的修改，但会造成一定的延时，因为集群需要重新协商并选取新的主设备。A - P 模式提供了备机保护。HA 集群中由一台主设备，和一台以上的从设备组成。

2. 从设备与主设备一样连接到网络，但不处理任何的数据包，从设备处于备用状态。从设备会自动同步主设备的配置，并时刻监视主设备的运行状态。整个失效保护的过程是透明的，一旦主设备失效，从设备会自动接替其工作。如果设备的接口或链路出现故障，集群内会更新链路状态数据库，重新选举新的主设备。

3. 通常建议部署 A - P 模式。

Active-Active (A - A) 模式

1. A - A 模式下会对占用资源较多的进程进行在各个设备中进行分担。需要处理协议识别、病毒扫描、ips、网页过滤、邮件过滤、数据防泄露、应用程序控制、VoIP 内容扫描、协议保护（HTTP/HTTPS/FTP/IMAP/IMAPS/POP3/SMTP/SMTPS/IM/NNTP/SIP/SIMPLE/SCCP）、协议控制等。通过对如上内容的负载均担，A - A 模式可以提供更高的 UTM 性能。安全策略中的终端控制，流控，用户认证功能，在 A - A 模式下没有什么提高效果。其他非 UTM 功能不会进行负载分担，还是由主设备进行处理。
2. A - A 模式下，集群中的主设备负责对所有通信会话的处理，然后将部分负载分发到所有备机上。备机可以说是活动的，因为要处理 UTM 的相关会话。但备机只处理由主设备分配的数据，不会响应 ARP 等。其他方面 A - A 模式和 A - P 模式是相同的。
3. 除非单台设备的 UTM 性能不满足要求，通常建议部署 A - P 模式。

单机配置同步、单机会话同步模式（FGSP）

从 5.0 版本开始，全新 NGFW 下一代防火墙支持单机配置同步、会话同步功能，在一些应用场景中它可以代替双机 HA 功能，实现控制异步流量的目的。

注意在 FGSP + 虚拟接口对（VWP）/透明模式 + UTM 检测开启的部署环境下，只支持 TCP 流量，不支持 UDP/ICMP 协议流量。

HA 配置要求

1. 进行 HA 配置， 硬件和软件版本需满足如下要求：

   - 防火墙硬件型号相同。
   - 同型号硬件要求硬件版本、内存容量、CPU 型号、硬盘容量等相同。
   - 相同的固件版本。 -**HA 的所有成员必须注册在同一 FortiCloud 账号下，否则 HA 虽然可以建立，但无法通过 FortiGuard 更新。**

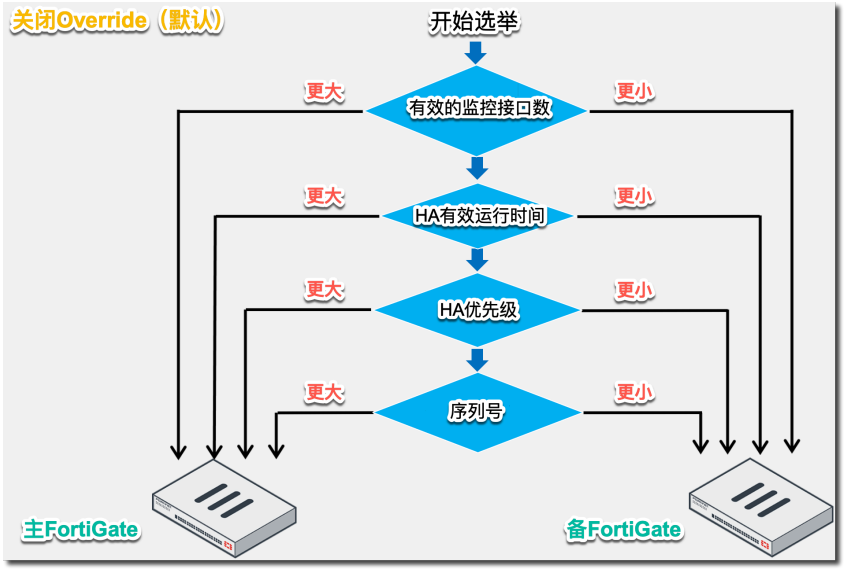
2. 假如 HA 的两台 FGT 存在上述不一致的情况，那么会出现两端无法同步或无法更新等问题。

HA 选取主设备的方法

主设备选举顺序

> **重要**
>
> FortiGate 6000/7000 系列的选举规则与此不同，请参考如下文档：https://community.fortinet.com/t5/FortiGate/Technical-Tip-HA-Master-Election-Process-on-FortiGate-6000-7000/ta-p/193109 ↗

防火墙在组成集群到时候，需要选举一台主设备。主设备的选举根据下图所示进行，主设备之外的设备则成为从设备。主设备硬件故障或者被监控的端口链路故障都会重新进行主设备的选举。防火墙按照如下顺序进行比较，进行主设备的选举。这些因素分别是：监控端口中的有效接口数量，设备 HA 有效运行时间，HA 优先级，设备的序列号。当 Override 关闭（默认配置）时，选举流程如下。



有效地被监控端口数量

1. 配置了需要被监控的业务端口之后，具有最多数量的有效监控端口的设备会成为主设备。通常情况下当一个 HA 集群建立起来的时候，所有到被监控的端口都是连接好并工作正常的，所以这个时候监控端口的数量不会成为影响到主设备选举的决定性因素。
2. 当一个被监控的端口出现故障或者链路失效，就会出发协商进行主设备选举，同样这些故障端口恢复到时候也会触发协商。当主设备的某个端口或链路出现故障，而其他设备相关端口和链路都是正常的就会从新选举出一个新的主设备。如 port3、port4 为被监控端口，当主设备的 port3 状态 down，则其监控端口有效数量减少，从设备此时有效接口数量不变会成为主设备，继续工作。
3. 如果从设备上的监控端口出现故障，也会进行协商选举，选举结果是主设备不变，因为从设备上监控端口数量较少，自然不会成为主设备。如果不同设备的不同接口现了不同的端口故障，则每次都会进行一次选举。

4.链路故障切换功能的目的就是要保证具有最多有效端口，最少故障点的那台设备成为主设备。

```
config system ha
    set monitor port3 port4
end
```

HA 有效运行时间

1. HA 有效运行时间较长的设备会成为主设备。**请注意该有效运行时间并不是设备已上电的运行时间**，该时间参数是以从上一次设备出现故障以来所正常运行的时间，当然设备重新启动后该时间值也会复位为 0。当集群内设备同时启动时，每台设备的 HA 有效运行时间是相同的。当设备上的监控端口发生设备故障时，HA 有效运行时间复位，会低于集群内其他设备，与此同时因为拥有有效监控端口数少于其他设备。当出现故障的监控端口恢复时，虽然与未发生故障的设备具有相同的监控端口数，由于 HA 有效运行时间被复位后小于其他设备，也无法成为主设备。
2. 在大多数情况下，集群通过 age 参数减少了选举新的主设备所需要的时间，使集群更加稳定，因为选举过程有可能会中断数据传输。

> **重要**
>
> 设备重新启动后 age 复位为 0，接口失效后设备 age 复位为 0。

集群启动时间差

有时集群内的某些主机需要更多的启动时间，HA 有效启动时间的不同会导致一系列的问题出现，但不会影响集群的操作。为了确保后启动的主机可以成为主设备，FGCP 协议默认为忽略 5 分钟的 HA 有效运行时间的差异，在大多数情况下该时间差的忽略可以帮助用户更容易实现自己预期的配置。在下列情况下 HA 有效运行时间差会造成不可以预料的结果：

1. 在升级固件版本的时候，默认情况下 uninterruptable-upgrade enabled 参数是开启的，集群会在所有的主机都升级完成后选举新的主设备。但由于主机之间升级后造成的 HA 有效运行时间的差异会小于 5 分钟，该差异会被忽略，其他的设备会成为新的主机。
2. 在测试失效切换的时候，由于反复切换，造成集群内主机 HA 有效运行时间的差小于 5 分钟。在通常情况下，失效切换后，失效的主机重新加入集群，他的 HA 有效运行时间要短于其他机，因而不会被选举为主设备。但如果失效的机器加入集群后他与其他主机之间的 HA 有效运行时间差小于 5 分钟，就有可能被选举为新的主机。

修改时间差

1. 用户可以通过如下命令来修改该参数。

```
config system ha
    set ha-uptime-diff-margin 60
end
```

2. 这个命令设置时间差为 60 秒，运行范围为 1-65535 秒，默认为 300 秒。如果你不想等待 5 分钟的时间以便进行测试，你可以减少差值；在非中断升级防火墙 OS 的过程中减少差值，或者当集群内机器启动时间差加大的时候增加该差值。

查看时间差

通过如下命令可以看到 HA 成员设备之间的当前 uptime 时间差，如下所示，FGT61FTKxxxx887 和 FGT61FTKxxxx594 的 uptime 间隔为 572721s（FGT61FTKxxxx594 的 uptime 比 FGT61FTKxxxx887 多 572721s）。FGT61FTKxxxx887 的历史 uptime reset 次数为 0，FGT61FTKxxxx594 的历史 uptime reset 次数为 1。

```bash
FortiGate # diagnose sys ha dump-by vcluster
            HA information.

vcluster_nr=1
vcluster-1: start_time=1765471976(2025-12-12 00:52:56), state/o/chg_time=2(work)/2(work)/1765471976(2025-12-12 00:52:56)
        pingsvr_flip_timeout/expire=3600s/0s
        mondev: internal5(prio=50,is_aggr=0,status=1)
        'FGT61FTKxxxx887': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0, flag=0×00000000, mem_failover=0,
uptime/reset_cnt=0/1
        'FGT61FTKxxxx594': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0×00000001, mem_failover=0,
uptime/reset_cnt=572721/1
```

HA 优先级

1. 当具有相同监控端口数，且同时启动运行的防火墙，具有较高优先级的设备优先成为主设备。防火墙默认的优先级为 128，可以通过配置优先级参数使某台设备优先成为主设备。该参数同主机名一样，不会在 HA 成员之间同步。

2. 当安装一组新的设备的时候，有效接口相同和 HA 有效运行时间相同，则优先级高会成为选主设备选举的因素。当一个台具有高优先级新设备加入一个现有的 HA 集群的时候，并不会触发从新协商，直到机群再次协商的时候该设备会由从设备被选举成为主设备。可以通过图形界面，或者如下的命令对优先级参数进行修改。

```bash
config system ha
    set priority 200
end
```

3. 可以通过 execute ha manage 命令对机群中的从设备进行优先级的修改。修改该参数后进行重新选举。

SN 序列号

当集群中的设备具相同的有效接口数，HA 有效运行时间和优先级的时候，设备的序号会成为最后比较的参数，因为不同的设备具有不同的序列号，具有较大数值的 SN number 的会成为主设备。

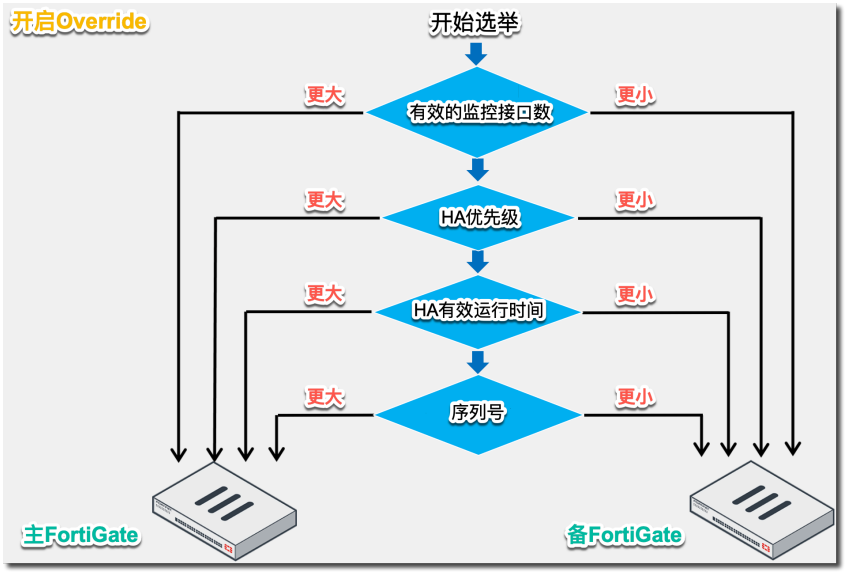
Override 参数

1. 在 HA 的配置中，override 参数会影响到主设备的选举过程。

```
config system ha
    set override [disable|enable]
end
```

2. 该参数默认需要通过 CLI 命令行进行配置，默认值为 disable。

3. 当 Override 参数开启后，主设备的选举方法发生改变，优先级参数成为比 HA 有效运行时间更优先比较的参数。

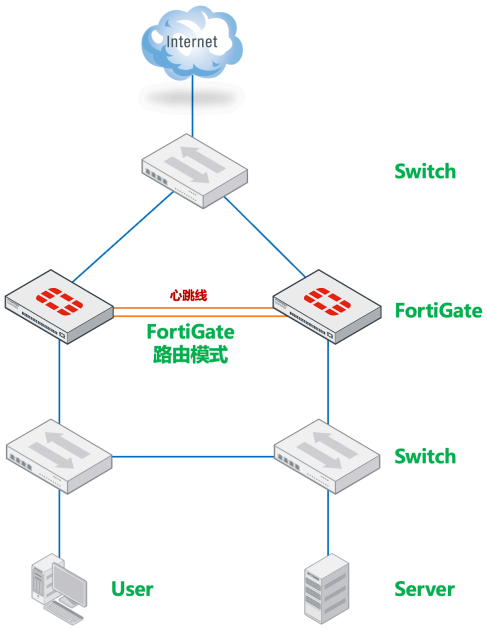


4. 如果一个设备的优先级较高，同时开启了 overide 选项，则在有效接口相同的情况下，该设备将始终作为主设备工作。由于该参数的这个特性，在实际操作中可能因为误操作造成设备配置的丢失，如下面的例子：

   - A 设备优先级 200，override enable，B 设备优先级 100，override disable。
   - A 设备故障，B 设备成为主设备。
   - 更换新的 A 设备，新设配仅设置 HA 参数，优先级 200，override enable，但未做任何的业务配置。
   - 新设备所有线路连接好后开启设备，由于新的设备和 B 设备具有相同的有效接口数量，同时具有较高优先级，选举成主设备。
   - 原来主设备 B 设备的配置文件被同步成为新设备的空配置，数据丢失。
   - 规避方法：检查 override 是否开启，检查优先级参数，或者新的设备接入的时候，不连接任何监控端口的线缆，使其具有少的有效端口数。

HA 典型基础配置（FGCP）

网络拓扑



配置建议

1. 进行 HA 环境下更换设备前，进行配置备份，防止操作失误而造成的配置丢失。
2. 建议配置两条以上的心跳线缆，防止单心跳故障造成 HA 机群崩溃，使用独立的心跳接口，尽量避免与业务口混用。
3. 优先使用光纤接口。
4. 开启会话同步：set session-pickup enable（默认关闭）或者在 WEB 上启用“会话交接”。
5. 谨慎使用 override 功能。开启 override 后设备选举过程中 HA 优先级参数高于设备运行时间参数，可能造成期望成为备机的设备被选举为主设备，造成反向同步配置信息。
6. 在同一个二层环境里面有多个 HA Cluster 时，注意 Group ID 不能相同，否则会产生 MAC 地址冲突。
7. 选择正确的监控端口和心跳端口，在开启 vdom 虚拟 cluster 时候，每个 cluster 需要单独配置。
8. 如果开启 ping server 功能，则需要再 HA 配置中添加相应的配置命令。
9. 建议将与防火墙相连的交换机接口配置为 stp portfast 模式，发生接口 Up/Down 切换时，交换机的接口可立刻进入转发状态，而不需要 stp 状态进行侦听、学习、转发等延迟步骤后接口才转发数据。

配置要点

- 进行 HA 的配置，硬件和软件版本需满足如下要求：
  - 防火墙硬件型号相同、同型号硬件需要为相同的硬件版本。
  - 内存容量、CPU 型号、硬盘容量等相同。
  - 相同的软件版本版本。
- 配置设备 1 的 HA。
- 配置设备 2 的 HA。
- 组建 HA。
- 查看 HA 集群。

配置步骤

1. 配置主设备，进入菜单系统管理 → 高可用性，模式选择“主动 - 被动”模式，优先级配置 200（主机高于从机）；组名/密码自定义；勾选“启用会话交接”。



- 模式：单机模式、主动 - 被动、主动 - 主动。在大多数网络中会选择主动 - 被动，即主设备进行业务处理，备机处于备份状态，当主设备发生设备故障或者接口链路故障后，则由备机继续处理业务。
- 设备优先级：优先级高的设备，优先被选为主设备。
- 组 ID：在同一个二层环境里面有多个 HA Cluster 时，注意 Group ID 不能相同，否则会产生 MAC 地址冲突。
- 组名和密码：自行定义即可，做 HA 的两台机器需要配置相同的参数。
- 启用会话交接：建议启用。主墙和备墙之间实时进行会话信息的同步，当发生 HA 切换到时候，备墙上有同样的会话信息会对原来的会话进行处理，不会产生会话中断。
- HA 的接口配置：
  - 端口监控：HA 监控的接口，作为 HA 切换依据之一；本案例中监控 port2（外网口）和 port4（内网口）。业务端口需要被防火墙监控，当端口出现故障时会进行切换，具有数量多的有效监控端口的设备会作为主墙处理数据。
  - 心跳接口：启用两个心跳接口：port7、port8。用于配置同步，会话同步，对方存活心跳检测等，为了集群的稳定建议配置 2 条或以上的线条线。在有多条心跳线路的情况下，心跳端口的 心跳优先级 决定了优先使用那条线路同步心跳（优先级高的端口连接的线路优先使用）

2. 配置从设备，除了优先级不同外（优先级低于主机），其他参数均与设备 1 相同。

3. 组建 HA，连接心跳线，FGT 主的 port7、port8，连接到 FGT 从的 port7、port8。

4. 防火墙开始协商建立 HA 集群，此时会暂时失去和防火墙到连接，这是因为在 HA 协商过程中会改变防火墙接口到 MAC 地址。可以通过更新电脑的 arp 表来恢复连接，Windows 下的命令为 arp -d。

5. 连接业务口链路。

6. 组建好 HA 后，两台防火墙配置同步，具有相同的配置，通过访问主防火墙来进行业务配置，如 IP 地址，策略等，更新的配置会自动同步。

> **重要**
>
> HA 建立完成后，只能通过主机进行访问、管理主机。如果需要登陆备机管理，请参考 HA 的网管介绍 ↗章节。

结果验证

1. 进入菜单系统管理 → 高可用性，就可以看到 HA 的建立情况。



2. 在主设备的 CLI 下查看 HA 状态。

```bash
FGT1-LEFT # get system ha status
HA Health Status: OK
Model: FortiGate-VM64-KVM
Mode: HA A-P
Group: 77
Debug: 0
Cluster Uptime: 0 days 0:23:56
Cluster state change time: 2023-01-04 17:01:52
Primary selected using:
    <2023/01/04 17:01:52> FGVM08TM22000173 is selected as the primary because its override priority is larger than peer
member FGVM08TM22000174.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
Configuration Status:
    FGVM08TM22000173(updated 3 seconds ago): in-sync
    FGVM08TM22000174(updated 4 seconds ago): in-sync
System Usage stats:
    FGVM08TM22000173(updated 3 seconds ago):
        sessions=9, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=33%
    FGVM08TM22000174(updated 4 seconds ago):
        sessions=9, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=33%
HBDEV stats:
    FGVM08TM22000173(updated 3 seconds ago):
        port7: physical/10000full, up, rx-bytes/packets/dropped/errors=4230515/13569/0/0, tx=5053711/11897/0/0
        port8: physical/10000full, up, rx-bytes/packets/dropped/errors=3215271/7193/0/0, tx=3246636/7167/0/0
    FGVM08TM22000174(updated 4 seconds ago):
        port7: physical/10000full, up, rx-bytes/packets/dropped/errors=5035221/11873/0/0, tx=4224168/13553/0/0
        port8: physical/10000full, up, rx-bytes/packets/dropped/errors=3244371/7162/0/0, tx=3212142/7186/0/0
MONDEV stats:
    FGVM08TM22000173(updated 3 seconds ago):
        port2: physical/10000full, up, rx-bytes/packets/dropped/errors=263241/3765/0/0, tx=46197/492/0/0
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=70859/610/0/0, tx=15210/114/0/0
    FGVM08TM22000174(updated 4 seconds ago):
        port2: physical/10000full, up, rx-bytes/packets/dropped/errors=291104/4091/0/0, tx=14698/104/0/0
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=69842/597/0/0, tx=14886/106/0/0
Primary     : FGT1-LEFT       , FGVM08TM22000173, HA cluster index = 1
Secondary   : FGT2-RIGHT      , FGVM08TM22000174, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.2
Primary: FGVM08TM22000173, HA operating index = 0
Secondary: FGVM08TM22000174, HA operating index = 1
```

```bash
FGT1-LEFT # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:51672 b:21448340
        traffic.total = s:0 p:51674 b:21448444
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=77 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM22000173:      Primary, serialno_prio=1, usr_priority=200, hostname=FGT1-LEFT
FGVM08TM22000174:    Secondary, serialno_prio=0, usr_priority=100, hostname=FGT2-RIGHT

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0:
FGVM08TM22000173:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM22000174:    Secondary, ha_prio/o_ha_prio=1/1
```

HA 虚拟 MAC

功能简介

在 FortiGate HA（FGCP）集群中，主设备（primary）的接口会使用虚拟 MAC（VMAC）。这样在故障切换（failover）后，新的主设备可以继承与旧的主设备相同的 VMAC 与 IP，使上下联交换机等网络设备仍把它识别为“同一台设备”，从而更快恢复转发与通信。

接口的两类 MAC

- Current_HWaddr：接口当前在网络中可见的 MAC 地址，在 standalone（单机）模式下可通过 CLI 修改。
- Permanent_HWaddr：网卡出厂写入的物理 MAC，不可更改。
- 默认配置（standalone 模式）下二者相同，开启 HA FGCP 后，主设备接口对外可见的 MAC（Current_HWaddr）会改为 HA VMAC（与 VDOM/接口/HA group ID 等因素相关）。

VMAC 工作模式

- NAT 模式：
  - 主设备每个接口通常获得不同的 VMAC。
  - VLAN 子接口继承其所在物理口的 VMAC（EMAC VLAN 是个例外）。
  - 冗余/聚合（802.3ad）接口继承列表中第一个成员接口的 VMAC。
- Transparent 模式：
  - FGCP 将 VMAC 分配给主设备的管理 IP。
  - 由于管理 IP 可从任意接口访问，表现为“所有接口看起来像同一个 VMAC”。
- HA 独立管理口：会保留原始 MAC，不会变成 VMAC。
- 备设备（Secondary）：接口 MAC 不会变为 VMAC，仍然使用原始 MAC。

查看接口 MAC/VMAC

- Standalone 模式：同时显示 Current_HWaddr 与 Permanent_HWaddr，未手动修改接口 MAC 的情况下，二者通常相同（物理 MAC）。

```bash
FortiGate # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       e0:23:ff:67:e3:9c
Permanent_HWaddr     e0:23:ff:67:e3:9c

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr e0:23:ff:67:e3:9c
        inet addr:10.10.12.2  Bcast:10.10.12.255  Mask:255.255.255.0
        inet addr6: 2100::2 prefixlen 64
        link-local6: fe80::e223:ffff:fe67:e39c prefixlen 64
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:179819739 errors:0 dropped:0 overruns:0 frame:0
        TX packets:124048087 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:203328470553 (189.4 GB)  TX bytes:60298053563 (56.2 GB)
```

- HA 模式：
  - 主设备：Current_HWaddr 变为 HA VMAC，而 Permanent_HWaddr 仍是物理 MAC，`ifconfig` 命令看到的 `HWaddr` 为 HA VMAC。

```bash
FortiGate # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       00:09:0f:09:07:00
Permanent_HWaddr     94:f3:92:53:ae:7a

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr 00:09:0F:09:00:00
        inet addr:172.22.6.65  Bcast:172.22.7.255  Mask:255.255.254.0
```

```
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:336270575 errors:0 dropped:0 overruns:0 frame:0
        TX packets:213599289 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:316749603468 (294.10 GB)  TX bytes:110547160311 (102.10 GB)
```

- 备设备：Current_HWaddr 与 Permanent_HWaddr 相同，为真实物理 MAC，`ifconfig` 命令看到的 `HWaddr` 为真实物理 MAC。

```bash
FortiGate #  diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       ac:71:2e:fd:21:be
Permanent_HWaddr     ac:71:2e:fd:21:be

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr AC:71:2E:FD:21:BE
        inet addr:172.22.6.65  Bcast:172.22.7.255  Mask:255.255.254.0
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:37615511 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:6160997309 (5.7 GB)  TX bytes:0 (0  Bytes)
```

接口 MAC/VMAC 映射表

```bash
FortiGate # diagnose sys ha mac

HA mac msg
serial#=FGT61FTK2206xxxx Primary
prio=0, phy_index= 0, itf_name=  wan1, mac=94.f3.92.53.ae.7a, vmac=00.09.0f.09.07.00, linkfail=0
prio=0, phy_index= 1, itf_name=  wan2, mac=94.f3.92.53.ae.7b, vmac=00.09.0f.09.07.01, linkfail=0
prio=0, phy_index= 2, itf_name=   dmz, mac=94.f3.92.53.ae.7c, vmac=00.09.0f.09.07.02, linkfail=0
prio=0, phy_index= 3, itf_name=internal1, mac=94.f3.92.53.ae.7d, vmac=--.--.--.--.--.--, linkfail=0
prio=0, phy_index= 4, itf_name=internal2, mac=94.f3.92.53.ae.7e, vmac=--.--.--.--.--.--, linkfail=1
prio=0, phy_index= 5, itf_name=internal3, mac=94.f3.92.53.ae.7f, vmac=00.09.0f.09.07.05, linkfail=1
prio=0, phy_index= 6, itf_name=internal4, mac=94.f3.92.53.ae.80, vmac=00.09.0f.09.07.06, linkfail=1
prio=0, phy_index= 7, itf_name=internal5, mac=94.f3.92.53.ae.81, vmac=00.09.0f.09.07.07, linkfail=0
prio=0, phy_index= 8, itf_name=     a, mac=94.f3.92.53.ae.82, vmac=00.09.0f.09.07.08, linkfail=1
prio=0, phy_index= 9, itf_name=     b, mac=94.f3.92.53.ae.83, vmac=00.09.0f.09.07.08, linkfail=1
prio=0, phy_index=10, itf_name=internal, mac=94.f3.92.53.ae.7e, vmac=00.09.0f.09.07.04, linkfail=1
prio=0, phy_index=11, itf_name=fortilink, mac=94.f3.92.53.ae.82, vmac=00.09.0f.09.07.08, linkfail=1
prio=0, phy_index=12, itf_name=emacvlan1, mac=9a.f3.92.53.ae.7a, vmac=00.09.0f.09.07.0c, linkfail=0
serial#=FGT61FTK22053887 Secondary
prio=1, phy_index= 0, itf_name=  wan1, mac=ac.71.2e.fd.21.be, vmac=--.--.--.--.--.--, linkfail=0
prio=1, phy_index= 1, itf_name=  wan2, mac=ac.71.2e.fd.21.bf, vmac=--.--.--.--.--.--, linkfail=0
prio=1, phy_index= 2, itf_name=   dmz, mac=ac.71.2e.fd.21.c0, vmac=--.--.--.--.--.--, linkfail=0
prio=1, phy_index= 3, itf_name=internal1, mac=ac.71.2e.fd.21.c1, vmac=--.--.--.--.--.--, linkfail=0
prio=1, phy_index= 4, itf_name=internal2, mac=ac.71.2e.fd.21.c2, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index= 5, itf_name=internal3, mac=ac.71.2e.fd.21.c3, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index= 6, itf_name=internal4, mac=ac.71.2e.fd.21.c4, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index= 7, itf_name=internal5, mac=ac.71.2e.fd.21.c5, vmac=--.--.--.--.--.--, linkfail=0
prio=1, phy_index= 8, itf_name=     a, mac=ac.71.2e.fd.21.c6, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index= 9, itf_name=     b, mac=ac.71.2e.fd.21.c7, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index=10, itf_name=internal, mac=ac.71.2e.fd.21.c2, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index=11, itf_name=fortilink, mac=ac.71.2e.fd.21.c6, vmac=--.--.--.--.--.--, linkfail=1
prio=1, phy_index=12, itf_name=emacvlan1, mac=b2.71.2e.fd.21.be, vmac=--.--.--.--.--.--, linkfail=0
```

- 输出包含每个接口的 phy_index、itf_name、mac（物理）、vmac（虚拟）、linkfail 等字段。
- 这个输出对排查“某个接口当前应该用哪个 VMAC”很直观。

VMAC 分配方式

有 3 种 VMAC 分配方式：默认方式、自动分配、手动分配。

默认方式

> **相关信息**
>
> - 以下 VMAC 算法仅适用于比较常见的情况：`vcluster_integer` = 0 或 1，`group-id` < 256，`phy_index` < 128的情况。
> - 其他情况下的 VMAC 算法会有所区别，具体请参考官方 Docs 文档 ↗。

默认配置下，HA VMAC 地址根据以下公式确定：

$$\langle group\text{-}prefix \rangle : \langle group\text{-}id\_hex \rangle : (\langle vcluster\_integer \rangle + \langle phy\_index \rangle)$$

- HA 配置示例：

```bash
config system ha
    set group-id 0
    set group-name "summerice"
    set mode a-p
    set password ENC
8d7WHMRiOFwjt7pOVnRmVrs9+rVayb7F2h1BKknc/3NsIjK1iEetIIVMzaGcdy3ONs7YzupDTFE09VWflnewYi91ZZ9V63QFh+EsKYhxjphPRf8cBmkQAq9D
cu6ymAo/Sm60+yVAUl5T+sUFbBivGJIG55rGnVV6BuN8VB3UYupQAUFOA9hZovr0TE6GcKkM/+lBf1lmMjY3dkVA
    set hbdev "port4" 0
    set override disable
end
```

- `group-prefix`：此情况下固定为 00:09:0f:09，占据 VMAC 的前 4 段（32 bit）。
- `group-id_hex`：vluster 的 group-id 转换为十六进制的形式，占据 VMAC 的第 5 段。下表列出了为每个 group-id 设置的 VMAC 地址。

> **相关信息**
>
> - 可用 `get system ha | grep group-id` 查看 group-id。

Integer Group ID	Hexadecimal Group ID
0	00
1	01
2	02
3	03
…	…
7	07
…	…
63	3f
…	…
255	ff

- `vcluster_integer`，占据 VMAC 的第 6 段二进制的第 1bit，其值 = `cluster_id` - 1：
  - 在未启用 VDOM + vcluster 时，默认所有接口均位于 root VDOM 中，`vcluster_id` 为 1，`vcluster_integer` = 0。
  - 在启用 VDOM + vcluster 时，`vcluster_id` 的十进制值取自如下 CLI 命令，如下所示，`vcluster_id` = 2，对应的：`vcluster_integer` = 1。

```bash
config global
config system ha
    set vcluster-status enable
    config vcluster
        edit 2
            set override disable
            set priority 200
            set vdom "vdom1"
        next
    end
end
end

FortiGate (global) # get sys ha | grep vcluster
vcluster-status     : enable
vcluster:
    vcluster-id:     1
    vcluster-id:     2
```

- `phy_index`：接口索引号。占据 VMAC 的第 6 段二进制的第 2~8 bit。接口编号从 0 到 x（其中 x 为接口总数）。可以使用 `diagnose sys ha mac` / `diagnose sys ha dump-by debug-zone` 命令查看接口的 `phy_index`。
- 举例：HA `group-id` 为 7，未开启 VDOM，wan1 接口的 `phy_index` = 0。
  - VMAC 前 4 段：固定值 00:09:0F:09。
  - VMAC 第 5 段：为 `group-id`（7）对应的 `group-id_hex`（07）。
  - 示例设备未开启 VDOM，`vcluster_id` 为 0，VMAC 的第 6 段的二进制第 1bit 为 0。
  - wan1 接口的 `phy_index` 为 0，VMAC 的第 6 段的二进制第 2~8 bit 为 0。
  - wan1 接口的 VMAC = 00:09:0F:09:07:00。

```bash
FortiGate # get sys ha | grep group-id
group-id            : 7

FortiGate # diagnose sys ha mac | grep wan1
prio=0, phy_index= 0, itf_name=  wan1, mac=94.f3.92.53.ae.7a, vmac=00.09.0f.09.07.00, linkfail=0    ←——主设备wan1的MAC和VMAC
prio=1, phy_index= 0, itf_name=  wan1, mac=ac.71.2e.fd.21.be, vmac=--.--.--.--.--.--, linkfail=0    ←——备设备wan1的MAC和VMAC（由于备机不使用VMAC，所以VMAC为空）

FortiGate # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr      00:09:0f:09:07:00
Permanent_HWaddr    94:f3:92:53:ae:7a

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr 00:09:0F:09:07:00
        inet addr:172.22.6.80  Bcast:172.22.7.255  Mask:255.255.254.0
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:336288265 errors:0 dropped:0 overruns:0 frame:0
        TX packets:213612129 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:316754405177 (295.0 GB)  TX bytes:110549910225 (102.10 GB)
```

  - 在多 VDOM 环境下，开启 HA 的 `vcluster-status`，将 wan1 切换至 vdom1（`vcluster_id` = 2，对应的：`vcluster_integer` = 1），则 VMAC 的第 6 段二进制第 1bit 变为 1（1000 0000），最终 VMAC = 00:09:0F:09:07:80。

```bash
FortiGate (global) # show sys interface wan1 | grep vdom
        set vdom "vdom1"

FortiGate (global) # show system ha
config system ha
......
    set vcluster-status enable
    config vcluster
        edit 1
            set override disable
            set priority 200
            set vdom "root" "vdom2" "vdom3"
        next
```

```
        edit 2
            set override disable
            set vdom "vdom1"
        next
    end
end

FortiGate (global) # diagnose sys ha mac | grep wan1
prio=0, phy_index= 0, itf_name=  wan1, mac=94.f3.92.53.ae.7a, vmac=00.09.0f.09.07.80, linkfail=0
prio=1, phy_index= 0, itf_name=  wan1, mac=ac.71.2e.fd.21.be, vmac=--.--.--.--.--.--, linkfail=0

FortiGate (global) # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr 00:09:0F:09:07:80
        inet addr:172.22.6.80  Bcast:172.22.7.255  Mask:255.255.254.0
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:340321975 errors:0 dropped:0 overruns:0 frame:0
        TX packets:216361446 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:319365771796 (297.4 GB)  TX bytes:110923401801 (103.3 GB)
```

自动分配

提示

- 该功能在 FortiOS 7.6.0 及以后版本支持。
- 只能配置物理接口。

在 HA 配置下针对特定接口启用 `auto-virtual-mac-interface` 命令启用此功能。FGCP 可根据主设备（Primary）的接口物理 MAC 地址，将本地管理位（U/L 位）修改为 1，作为该接口的 VMAC 地址。

相关信息

在 48 bit 的物理 MAC 地址中，U/L 位指 16 进制 MAC 地址首字节（8 bit）的倒数第 2 bit：

- 当该 bit 为 0 时：表示该 MAC 地址为通用地址，即由权威机构分配。
- 当该 bit 为 1 时：表示该 MAC 地址为本地地址，即由本地分配。

1. 例如，wan1 接口十六进制地址 E0:23:FF:67:E3:9C 的首字节（8 bit）二进制表示为 1110 0000，其中 U/L 位为 0。默认 HA 配置下，该接口的 VMAC 使用基于 `group-id` 的分配方式。

```bash
FortiGate # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       00:09:0f:09:00:02
Permanent_HWaddr     e0:23:ff:67:e3:9c
```

2. 针对 wan1 接口启用 `auto-virtual-mac-interface`（可配置多个接口，也可配置为 `all`）。

```bash
config system ha
    set auto-virtual-mac-interface wan1 wan2  ←——可配置多个接口，也可配置为all
end
```

3. 再次查看 wan1 接口的 VMAC，其中 U/L 位被设置为 1，变为 1110 0010（E2），VMAC = E2:23:FF:67:E3:9C。

```bash
FortiGate-101F # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       e2:23:ff:67:e3:9c
Permanent_HWaddr     e0:23:ff:67:e3:9c

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr E2:23:FF:67:E3:9C
        inet addr:10.10.12.2  Bcast:10.10.12.255  Mask:255.255.255.0
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:182841979 errors:0 dropped:0 overruns:0 frame:0
        TX packets:126672095 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:205975479283 (191.8 GB)  TX bytes:60698259991 (56.5 GB)
```

手动分配

> **提示**
>
> - 该功能在 FortiOS 7.6.0 及以后版本支持。
> - 手动分配 VMAC 可在物理接口、EMAC VLAN 接口或 FortiExtender 接口上进行配置。该设置将覆盖该接口上的其他 VMAC 地址分配。

1. 例如，wan1 接口的 VMAC 为 00:09:0f:09:00:02。

```bash
FortiGate # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       00:09:0f:09:00:02
Permanent_HWaddr     e0:23:ff:67:e3:9c
```

2. 手动修改 wan1 接口的 VMAC 为 12:23:34:45:56:67。

```bash
config system interface
    edit "wan1"
        set virtual-mac 12:23:34:45:56:67
    next
end
```

3. 再次查看 wan1 接口的 VMAC，其中 U/L 位被设置为 1，变为 1110 0010（E2），VMAC = E2:23:FF:67:E3:9C。

```bash
FortiGate # diagnose sys ha mac | grep wan1
prio=0, phy_index= 2, itf_name=  wan1, mac=94.f3.92.53.ae.7a, vmac=12.23.34.45.56.67, linkfail=0
prio=1, phy_index= 0, itf_name=  wan1, mac=ac.71.2e.fd.21.be, vmac=--.--.--.--.--.--, linkfail=0

FortiGate # fnsysctl ifconfig wan1
wan1    Link encap:Ethernet  HWaddr 12:23:34:45:56:67
        inet addr:172.22.6.80  Bcast:172.22.7.255  Mask:255.255.254.0
        UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
        RX packets:340321975 errors:0 dropped:0 overruns:0 frame:0
        TX packets:216361446 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:319365771796 (297.4 GB)  TX bytes:110923401801 (103.3 GB)
```

单机模式修改 MAC 地址

当 FortiGate 处于 Standalone（单机）模式时：在接口层面可以通过 `macaddr` 选项修改 MAC 地址。

```bash
config system interface
  edit "wan1"
    set macaddr 00:01:02:03:04:05
  next
end

FortiGate-101F # diagnose hardware deviceinfo nic wan1 | grep HWaddr
Current_HWaddr       00:01:02:03:04:05
Permanent_HWaddr     e0:23:ff:67:e3:9c
......
```

VMAC 工作机制

GARP

当 HA 发生主备切换后，新主设备会发送包含接口 VMAC 的 gratuitous ARP（GARP）来尽快更新上下联交换机/路由器的转发表（MAC 表/ARP 表），让上下游设备更快把流量转发到新的主设备接口。

GARP 不是绝对必须（网络设备最终也会自学习），但通常能显著加快收敛。

配置参数

```bash
config system ha
  set arps <integer>
  set arps-interval <integer>
  set gratuitous-arps {enable | disable}
  set link-failed-signal {enable | disable}
end
```

- `gratuitous-arps`：是否发送 GARP（默认开启）。
- `arps`：HA 切换后发送的 GARP 次数（1-60，默认 5）。
- `arps-interval`：GARP 间隔秒数（1-20，默认 8）。
- `link-failed-signal`：
  - HA 切换后让所有接口 shutdown 1 秒再 up，用于强制上下游设备链路状态变化，接口 down/up 后，上下游设备会重新学习 MAC/ARP 表（默认关闭）。
  - 当关闭了 `gratuitous-arps` 或 GARP 不能有效更新网络时（例如上下游设备配置了安全功能忽略了 FortiGate 发出的 GARP），建议启用 `link-failed-signal`。

HA 部署注意事项

- 同一广播域内每套 HA 集群建议使用唯一的 HA group ID，避免 VMAC 冲突。
- 在 FortiGate 透明模式下观察到“多个接口同 VMAC”可能是预期行为（VMAC 绑定到管理 IP 的表现），不要先入为主当成异常。
- 如果 HA 切换后交换机收敛慢，优先从 GARP（次数/间隔）和 `link-failed-signal` 这两个参数调整。
- 排查 VMAC 生成逻辑时，先明确你处于哪一种分配方式（默认逻辑/自动/手动），再去对照算法，避免计算结果偏差。

常见问题

HA 切换后业务中断或间歇丢包

1. 查看 FortiGate 是否发送 GARP 通知上下游设备。
2. 上下联设备是否及时学习到新的 VMAC 入口（MAC 表/ARP 更新）。
3. `arps/arps-interval` 是否过小、是否需要启用 `link-failed-signal`。
4. 是否存在 VMAC 冲突（同广播域多个集群 group-id 相同）。
   - 多个 HA 集群处在同一广播域且使用相同 HA group ID 时，可能引发 MAC 冲突。
   - 可通过修改其中一套 HA 集群的 `group-id` 解决（或使用自动、手动方式分配 VMAC）。

聚合接口修改成员后 VMAC 变化

- 冗余/聚合（802.3ad）接口继承列表中第一个成员接口的 VMAC。
- 在 FortiGate 设备上向聚合接口添加或移除成员时如果操作的是聚合口的第一个成员接口，会导致聚合接口的 MAC/VMAC 地址变化，可能会导致上下游设备业务抖动。
- 如果不对聚合口的第一个成员接口做修改，建议在聚合接口中增加成员时，使用 `append` 命令追加成员接口，新增成员接口会添加到聚合口成员列表末尾。

```bash
config system interface
    edit "agg1"
        append member port18
    next
end
```

命令速查

命令	用途	关键字段/结果
`diagnose hardware deviceinfo nic`	列接口名	接口列表
`diagnose hardware deviceinfo nic`	查单接口 MAC	Current_HWaddr/Permanent_HWaddr
`diagnose sys ha mac`	查接口/VMAC 映射	itf_name/phy_index/mac/vmac
`diagnose sys ha dump-by debug-zone`	查接口 phyindex	phyindex
`get system ha`	查看 HA 配置	用于确认 `group-id` 等信息

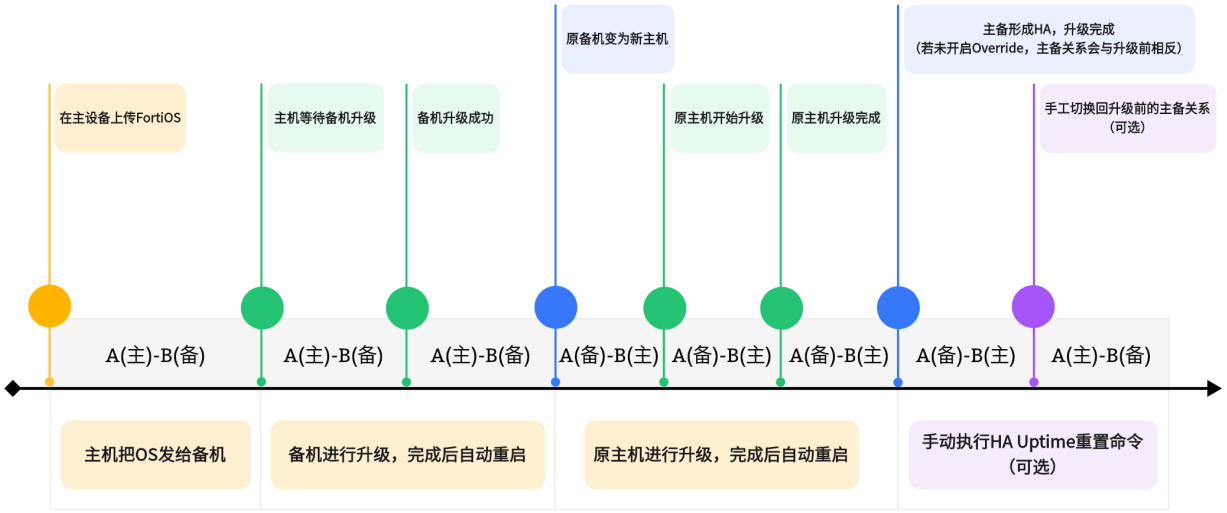
HA-Cluster 不间断升级

重要

在 HA 环境下降级，没有不间断的机制，降级时 HA 所有成员会同时重启，请务必注意。 降级建议使用系统管理 → 固件与配置管理 → 固件切换 ↗ 中介绍的方法。

不间断升级介绍

1. “HA 不间断升级”故名思义就是说在 HA 升级过程中，HA-cluster 中的成员会依次进行升级，从而保证了业务在升级过程中不中断。
2. 上传 FortiOS 的方法分为 Web 界面上传和 CLI 上传，本文介绍以 Web 界面上传的方式，两台设备升级完成需要 15 分钟左右，过程如下（未开启 Override）：



升级过程

1. 在主机 Web 界面上点击升级按钮。
2. 主机把 FortiOS 发给备机，备机准备并开始升级，然后自动重启，此过程需要 5 分钟（或更长时间），请耐心等待。
3. 备机升级完成，并切换成主机。
4. 主机观察到备机升级成功并切换为新的主机后，开始准备升级，此过程需要 5 分钟（或更长时间），请耐心等待。
5. 一旦准备完成，主机（此时为备）自动重启加载升级后 OS，重启约 2 分钟。
6. 主机（此时为备）重启后，形成 HA 。主机从原来的备同步配置（外部配置和内部配置），约 1~3 分钟。
7. 此时，HA 重新建立，如果 HA 没有开启 Override，主备关系与升级前相反，如果想要主备关系与升级前一致，在新的主机（原备机）手工执行 diagnose sys ha reset-uptime 命令切换回升级前的主备关系（此命令需要原主机升级完成启动 5 分钟后，在新主机上执行才会进行主备切换）。

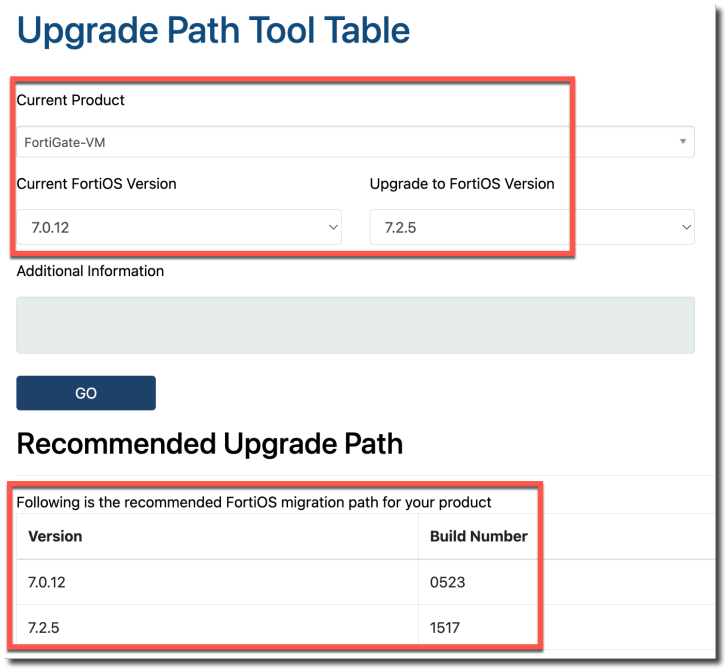
操作建议

1. 操作前下载并阅读 Release 文档和升级说明文档，做到注意事项心中有数。可以参考官方文档：https://docs.fortinet.com/document/fortigate/7.2.99/fortios-release-notes/ ↗ （进入后找到对应版本的 Release Note，此链接为目前 7.2.X 最新版本的 Release Note）
2. 升级前后备份设备配置。
3. 升级固件必须遵循升级路径，参考系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗ 章节。
4. 接入 Console 在主设备上，记录 log 观察升级过程，做到心中有数。 有问题也可以反馈厂商。
5. 耐心等待，主备机升级的时间都会比较长。

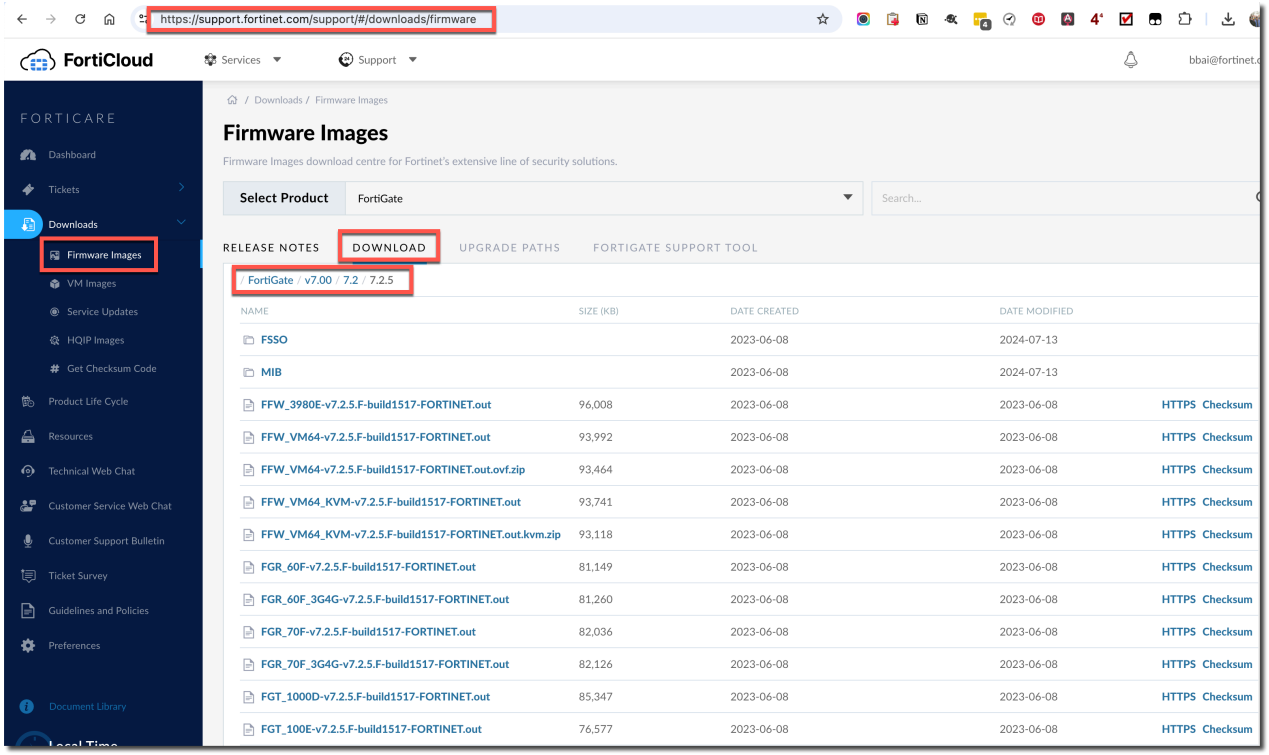
操作步骤

1. 分别备份主备设备的配置文件，参考系统管理 → 固件与配置管理 → 配置管理 → 配置备份及恢复 ↗ 章节。

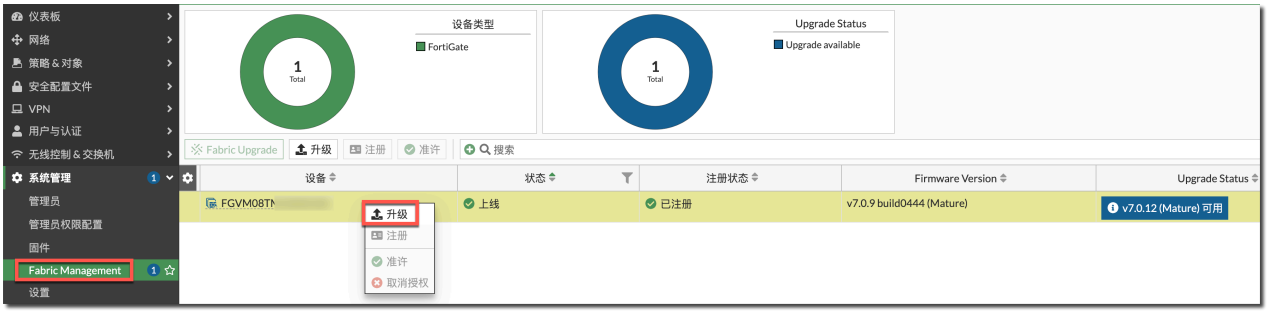
2. 举例：FortiGate-VM 从 FortiOS 7.0.12 升级到 FortiOS 7.2.5，从 https://docs.fortinet.com/upgrade-tool ↗ 得知直接升级即可，无需升级多次版本。



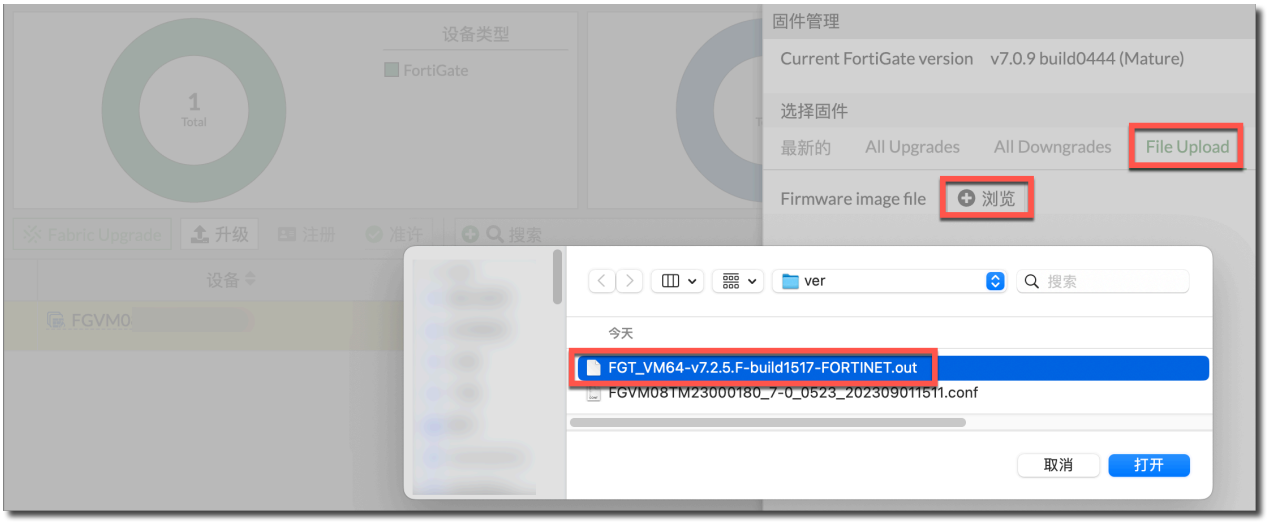
3. 从 https://support.fortinet.com/support/#/downloads/firmware ↗ 下载 7.2.5 版本对应产品的文件到本地，以便通过 GUI 导入软件版本进行升级。



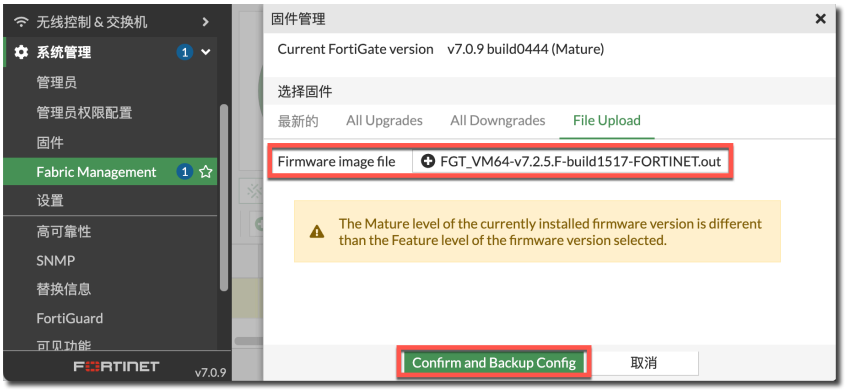
4. 准备工作做完，正式开始升级，在主设备进入系统管理 → Fabric 管理页面（7.4 为 Firmware & Registration 页面），右键点击要升级的设备，选择升级按钮。



5. 在弹出的页面中，选择文件上传，并浏览选择需要升级的版本。



6. 点击确认并备份配置，会在升级前下载设备当前的配置文件到本地（只有当前主机的）。





7. 此时 FortiOS 将先传给备机，业务还是运行在主机上，等备机重启后将接管主机的业务，然后主机再重启升级，此过程业务是不中断的。



Console 记录

1. 耐心等待与观察 Console 输出信息，做到心中有数。以下为升级过程 Left 设备 Consloe 打印信息，其实管理员并不需要做什么，只需要请注意阅读并理解，做到心中有数，万一出现故障可以知道大概是哪一步的问题。

```
FGT1-LEFT # Wait for HA to be primary of all clusters ...     //主设备发送FortiOS到备设备
Send image to HA secondary.
..Wait for secondary to restart ...    //备机先升级，升级重启约5分钟，请耐心等待
Wait for first secondary to become new primary..     //备机升级并重启完毕后，主机会先等待备机变成新的主机，以便业务被备机先接管
Firmware upgrade in progress ...     //当备机成为新主机之后，主机将准备升级
Done.    //主机准备升级结束，开始正式升级主设备，升级过程约5分钟，请耐心等待
The system is going down NOW !!
```

```
Please stand by while rebooting the system.
Restarting system.
SeaBIOS (version rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org)
Machine UUID 934483a5-b727-447c-b98c-9cb5dd2209fb
Loading flatkc ... ok
Loading /rootfs.gz ... ok
System is starting ...
The config file may contain errors.
Please see details by the command 'diagnose debug config-error-log read'.
Serial number is FGVM08TM22000173
FGT1-LEFT login: admin
Password:
Welcome!
FGT1-LEFT # get system status
Version: FortiGate-VM64-KVM v7.2.5,build1517,230606 (GA.F)    //原主机后升级完毕
Security Level: 1
Firmware Signature: certified
Virus-DB: 1.00000(2018-04-09 18:07)
Extended DB: 1.00000(2018-04-09 18:07)
Extreme DB: 1.00000(2018-04-09 18:07)
AV AI/ML Model: 0.00000(2001-01-01 00:00)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 6.00741(2015-12-01 02:30)
APP-DB: 6.00741(2015-12-01 02:30)
INDUSTRIAL-DB: 6.00741(2015-12-01 02:30)
IPS Malicious URL Database: 1.00001(2015-01-01 01:01)
IoT-Detect: 0.00000(2022-08-17 17:31)
Serial-Number: FGVM08TM23000180
License Status: Valid
License Expiration Date: 2024-01-19
VM Resources: 2 CPU/8 allowed, 3946 MB RAM
Log hard disk: Available
Hostname: FGVM08TM23000180
Private Encryption: Disable
Operation Mode: NAT
Current virtual domain: root
Max number of virtual domains: 10
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: disable
Current HA mode: a-p, secondary    //升级完毕之后状态是主还是备将视具体情况而定：主防火墙和备防火墙重启完成时间相差五分钟以内，则后面重启的主机（优先级高的FGT）将会主动抢占为主设备，如果二者升级重启时间相差超过5分钟，则后重启的主设备将会处于备的状态，不会抢占回主的状态
Cluster uptime: 2 hours, 15 minutes, 17 seconds
Cluster state change time: 2023-09-01 05:16:51
Branch point: 1517
Release Version Information: GA
FortiOS x86-64: Yes
System time: Fri Sep  1 15:41:51 2023
Last reboot reason: warm reboot
```

2. RIGHT 防火墙的 Console 记录。

```
FGT2-RIGHT # geGet image from ha primary OK.    //主机发送FortiOS到备机上
Verifying the integrity of the firmware image ...
Check image OK.
Please wait for system to restart.    //备机先开始升级并重启，因此业务在主机上运行
Firmware upgrade in progress ...
Done.
The system is going down NOW !!
Please stand by while rebooting the system.
Restarting system.
SeaBIOS (version rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org)
Machine UUID fe581bd9-2eb0-4115-896a-dd50f521d0ee
Booting from Hard Disk ...
SYSLINUX 6.04 EDD 6.04-pre1 Copyright (C) 1994-2015 H. Peter Anvin et al
Loading flatkc ... ok
Loading /rootfs.gz ... ok
System is starting ...
The config file may contain errors.
Please see details by the command 'diagnose debug config-error-log read'.
Serial number is FGVM08TM22000174
```

```
FGT2-RIGHT login: admin
Password:
Welcome!
FGT2-RIGHT # get system status
Version: FortiGate-VM64-KVM v7.2.5,build1513,230606 (GA.F)    //原备机升级成功
Security Level: 1
Firmware Signature: certified
Virus-DB: 1.00000(2018-04-09 18:07)
Extended DB: 1.00000(2018-04-09 18:07)
Extreme DB: 1.00000(2018-04-09 18:07)
AV AI/ML Model: 0.00000(2001-01-01 00:00)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 6.00741(2015-12-01 02:30)
APP-DB: 6.00741(2015-12-01 02:30)
INDUSTRIAL-DB: 6.00741(2015-12-01 02:30)
IPS Malicious URL Database: 1.00001(2015-01-01 01:01)
IoT-Detect: 0.00000(2022-08-17 17:31)
Serial-Number: FGVM08TM23000180
License Status: Valid
License Expiration Date: 2024-01-19
VM Resources: 2 CPU/8 allowed, 3946 MB RAM
Log hard disk: Available
Hostname: FGVM08TM23000180
Private Encryption: Disable
Operation Mode: NAT
Current virtual domain: root
Max number of virtual domains: 10
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: disable
Current HA mode: a-p, primary    //升级完毕之后状态是主还是备将视具体情况而定：主防火墙和备防火墙重启完成时间相差五分钟以内，则后面重启的主机（优先级高的FGT）将会主动抢占为主设备，如果二者升级重启时间相差超过5分钟，则后重启的主设备将会处于备的状态，不会抢占回主的状态
Cluster uptime: 2 hours, 16 minutes, 28 seconds
Cluster state change time: 2023-09-01 05:16:51
Branch point: 1517
Release Version Information: GA
FortiOS x86-64: Yes
System time: Fri Sep  1 15:41:51 2023
Last reboot reason: warm reboot
```

3. 手工切换 HA 状态，此步骤不是必要，升级完毕之后状态是主还是备将视具体情况而定：
   - 情况 1：主防火墙和备防火墙重启完成时间相差五分钟以内，则后面重启的主机（优先级高的 FGT）将会主动抢占为主设备。如果是这种情况，则不需要任何的额外切换动作。
   - 情况 2：如果二者升级重启时间相差超过 5 分钟，则后重启的主设备将会处于备的状态，不会抢占回主的状态。这个时候才需要命令行进行强制的 HA 切换一下，让 HA 状态恢复到升级之前的主备状态。
4. 如果有以上第 2 种情况，HA-Cluster 升级完毕之后，备机成了新的主，而旧的主机成为了备，并且想要将 HA 的状态切换到升级之前的状态，则只需要重启完成 5 分钟后在备机（新的主机）上敲一条命令即可完成主备切换。

```
FGT2-RIGHT # diagnose sys ha reset-uptime    //此命令会Reset当前设备的HA时间，以便重新进行HA的选举。reset-uptime相当于把备机的HA运行时间清0，这样运行时间长且优先级高的主设备，一定会重新选举为主的
```

5. 再观察 HA 的状态是否是我们所需要的状态，以下这三个命令都可以看到 HA 的选举结果和 HA 状态。

```
# diagnose sys ha status
# get system status
# get system ha status
```

6. 升级后检查是否有重要配置丢失。

```
diagnose debug  config-error-log read
//不丢路由、接口地址、服务对象、防火墙策略、VPN这些，一般问题不大。丢一些默认的配置可能是因为新版本做了功能变
```

7. 检查配置是否同步，查看主备防火墙的配置 HA checksum，正常来说二者的配置 checksum 是一致的，如果状态不同步，可以参考 HA 双机热备 → HA 常见问题 → HA 同步机制和故障诊断↗章节排查（如果升级路径有多个版本，建议在每个版本升级完成，等待 HA 同步后，再升级下个版本）。

```
FGT1-LEFT # diagnose sys ha checksum show
is_manage_primary()=0, is_root_primary()=0
debugzone
global: a5 20 2d 71 a0 22 13 92 59 82 45 6d 43 b8 fb d9
root: d0 f3 bd a2 b5 f8 df 39 7c 6c 25 c9 91 e1 b9 99
all: b9 3b b3 36 86 d7 ff 08 1d e8 ce 48 e4 85 85 48
checksum
global: a5 20 2d 71 a0 22 13 92 59 82 45 6d 43 b8 fb d9
root: d0 f3 bd a2 b5 f8 df 39 7c 6c 25 c9 91 e1 b9 99
all: b9 3b b3 36 86 d7 ff 08 1d e8 ce 48 e4 85 85 48

FGT2-RIGHT # diagnose sys ha checksum show
is_manage_primary()=0, is_root_primary()=0
debugzone
global: a5 20 2d 71 a0 22 13 92 59 82 45 6d 43 b8 fb d9
root: d0 f3 bd a2 b5 f8 df 39 7c 6c 25 c9 91 e1 b9 99
all: b9 3b b3 36 86 d7 ff 08 1d e8 ce 48 e4 85 85 48
checksum
global: a5 20 2d 71 a0 22 13 92 59 82 45 6d 43 b8 fb d9
root: d0 f3 bd a2 b5 f8 df 39 7c 6c 25 c9 91 e1 b9 99
all: b9 3b b3 36 86 d7 ff 08 1d e8 ce 48 e4 85 85 48
```

8. HA 升级完毕，升级完毕之后查看 HA 的状态、测试业务、最好再次备份一份软件版本升级之后的配置文件。

关闭不间断升级（谨慎使用）

某些特殊情况下，用户需要关闭不间断升级功能，通过如下命令可以关闭不间断升级功能。

> **重要**
>
> 关闭 HA 的不间断升级功能，执行升级操作（上传版本文件）后，HA 的所有成员设备会同时重启升级，业务会发生中断！请务必谨慎使用此命令！！！

```
config system ha
    set uninterruptible-upgrade disable
end

7.4.1版本后，命令变为：
config system ha
    set upgrade-mode simultaneous
end
```

具体解释可参考官方文档：https://docs.fortinet.com/document/fortigate/7.0.99/administration-guide/247944/upgrading-fortigates-in-an-ha-cluster ↗ 中的“Interrupted upgrade”部分。

HA 切换测试注意事项

1. 部分用户在做完 HA 升级或上线后，会做 HA 的切换测试，上边的步骤中已经介绍，通常建议通过命令“diagnose sys ha reset-uptime”进行 HA 主备的切换（默认情况下需要主备的 HA uptime 间隔超过 5 分钟）。
2. 当需要通过 HA 主设备监控接口 Down 的方式测试触发 HA 切换时，尽量使用手工拔线的方式，切记不要直接在页面将主设备监控接口 Shutdown，因为 HA 备机会同步该配置，随后也将监控接口 Shutdown，这会导致 HA 发生主备切换后再发生一次主备切换。
3. 如果不方便到机房手工拔线，可以使用如下 CLI 将主机的监控接口物理性 Down 掉，该操作仅本机生效，不会同步到备机。这样就可以模拟手工拔线相同的效果。

   ```
   fnsysctl ifconfig port1 down
   ```

4. HA 切换测试完成后，记得将原主机 HA 监控接口 UP。

   ```
   fnsysctl ifconfig port1 up
   ```

HA 共享单一 License

重要

- 此功能在 FortiOS 7.2.9、7.4.6、7.6.1 及更高版本支持。
- 在 FortiOS 7.2.12 和 7.4.9 及之后版本上，支持把 FortiToken 绑定到 vSN。之前的版本无法将 FortiToken 绑定在 vSN 上。

功能介绍

FortiGate A - P 模式的 HA 现在支持为以下型号的 HA 成员共享单个 FortiGuard 服务 License：

- 40F-HA 及衍生型号
- 60F-HA 及衍生型号
- 70F-HA 及衍生型号
- 80F-HA 及衍生型号
- 100F-HA 及衍生型号（**100F-HA 在 7.2.12、7.4.8、7.6.3 及更高版本支持**）

在这之前，客户需要为 HA 成员的每台设备单独购买 License，如果两台设备的服务到期时间不一致，会以较短的那台为准。

现在，当客户购买两台 FortiGate 设备组成 A - P HA 集群时（如 2 x FortiGate-40F-HA），以上型号可以共享一个 FortiGuard 服务 License（如 1x FC-10-0040F-809-02-DD），无需为每台设备单独购买，可共享的服务只支持如下三种：

- Enterprise Protection
- Unified Threat Protection (UTP)
- Advanced Threat Protection (ATP)

两个 FortiGate 序列号将在 FortiCare 中关联，FortiCare 系统会将两台设备的序列号绑定，生成一个虚拟序列号（vSN），上述服务将注册至这个 vSN，而非单个设备的序列号。

实施步骤

重要

HA vSN 集群的建立方式有所限制：

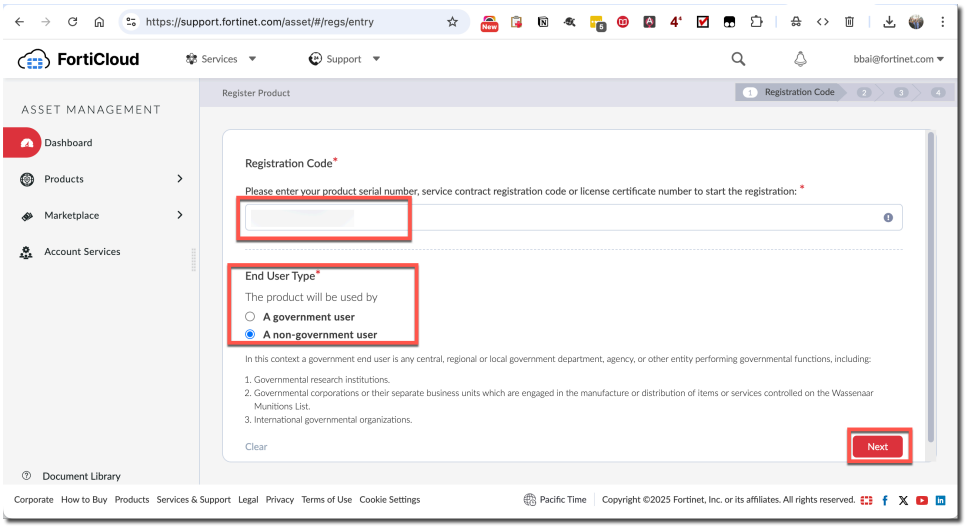
- 仅支持 HA A - P 模式，其他部署模式（如 Active-Active）不适用。
- FortiGate-HA（如 FortiGate-40F-HA）是单独的 SKU，您必须购买带有“HA”后缀的特定型号设备（如 2 * FortiGate-40F-HA，这种设备是成对出售的），才能组成 HA vSN 集群。
- 不带“HA”后缀的普通型号设备（如 FortiGate-40F）无法组成 HA vSN 集群。即使您之前已经购买了两台不带“HA”后缀的设备，它们也无法组成 HA vSN，这种设备必须单独购买服务。

注册 HA 设备

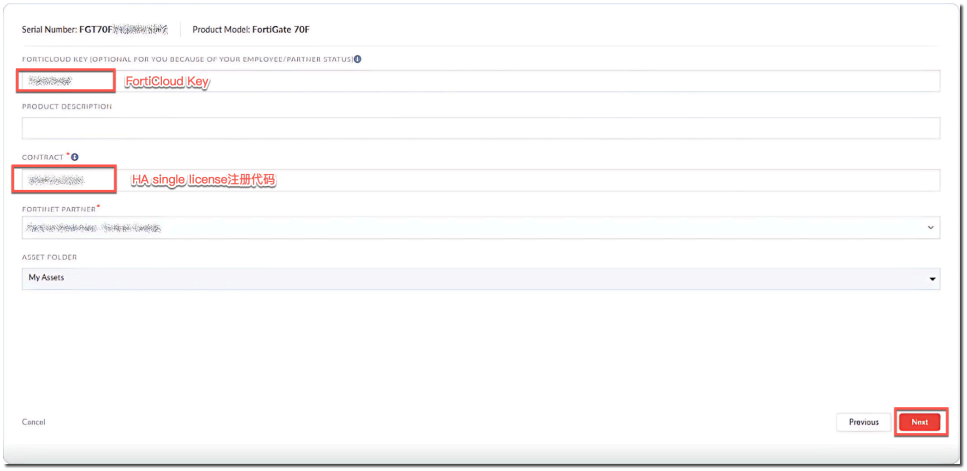
1. 访问 https://support.fortinet.com/asset/#/dashboard ↗ 并使用要注册设备的 FortiCloud 账号登录。

2. 点击页面上的“Register Now”按钮。

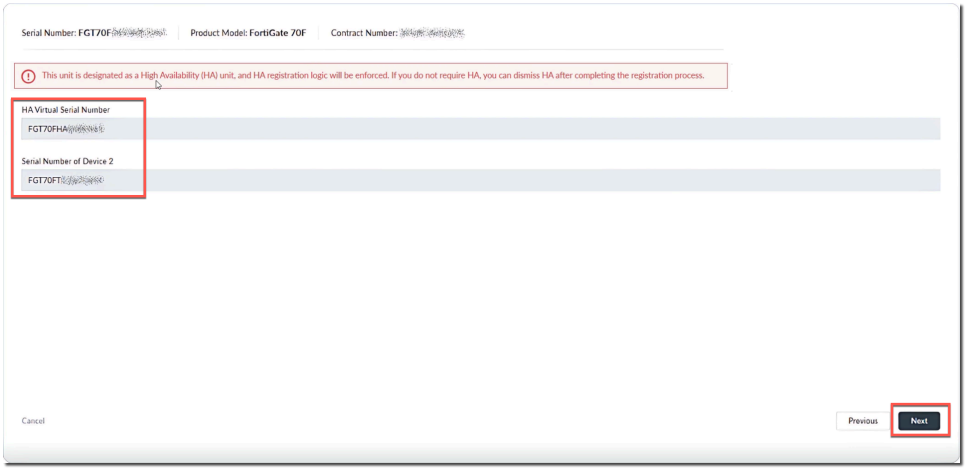
3. 在“Registration Code”中，输入一对 FortiGate-HA 其中任意一台（Device 1）的序列号，选择“End User Type”，然后点击“Next”。



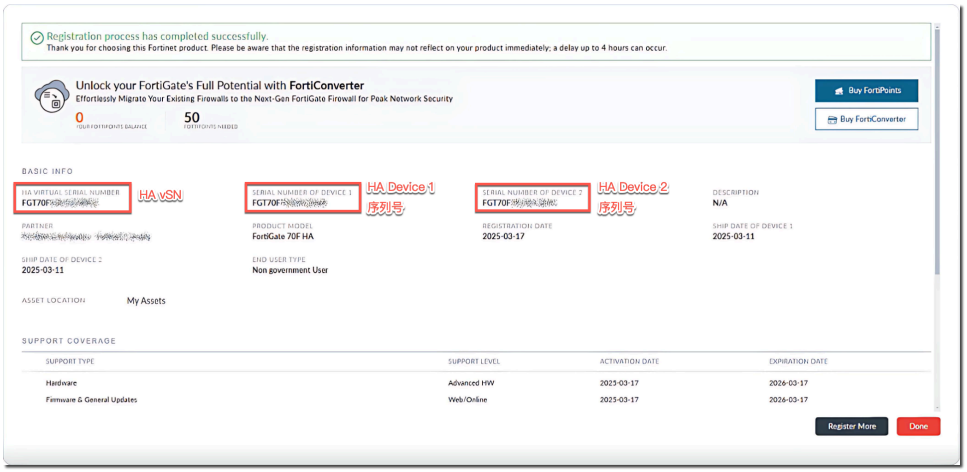
4. 填入“FortiCloud Key”（从机身粘贴的标签上获取），HA single license 的注册代码（从购买的 HA license PDF 文件中获取），然后点击“Next”。



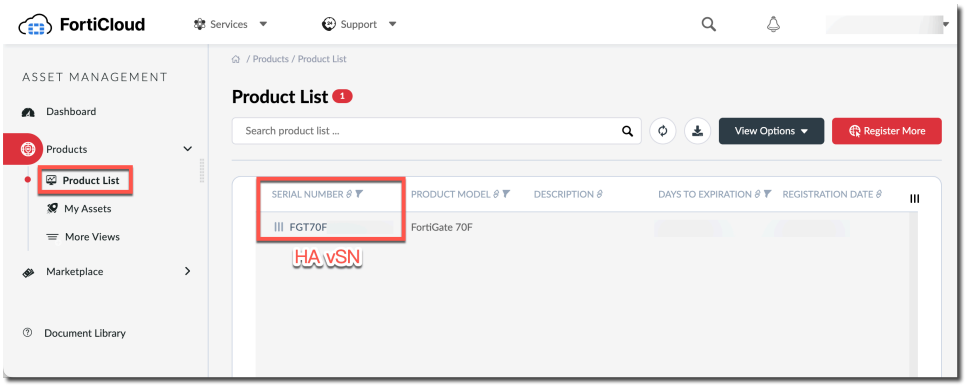
5. 随后会生成 HA 的 vSN，并显示 HA 另一台设备（Device 2）的序列号。



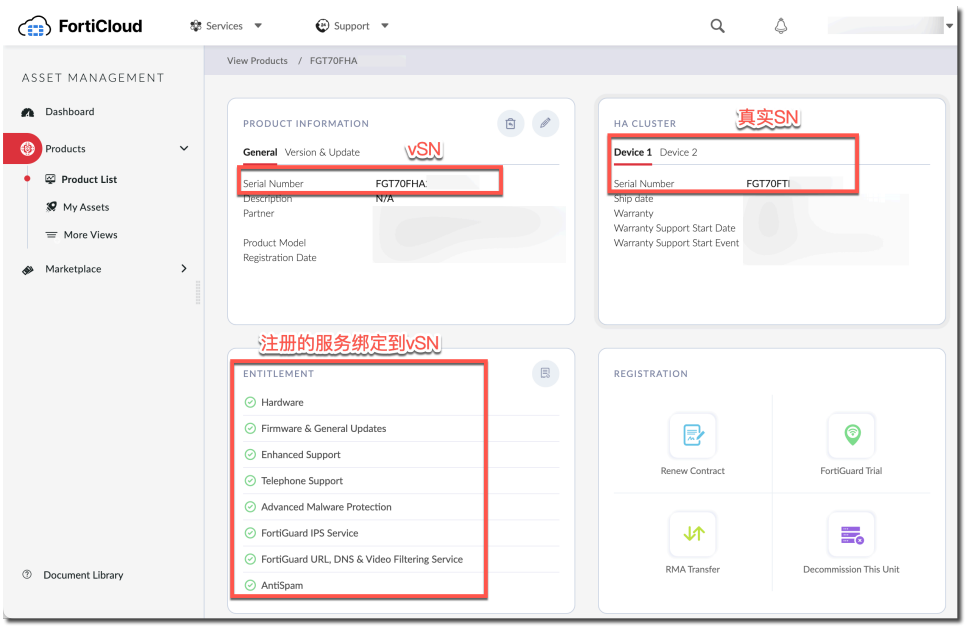
6. 注册完成，显示 HA 的 vSN 和两台成员设备的真实序列号。



7. 注册完成后，vSN 会出现在设备列表中。



8. 订阅的服务授权（如 UTP/ATP）绑定到了 vSN，而非成员设备。



HA 设备配置

1. 将两台设备组成 A - P 模式的 HA，参考“HA 双机热备 → HA 典型基础配置（FGCP）↗”。

2. 在 HA 任意一台设备的 CLI 下敲入如下命令（主备机会同步该配置）：

```bash
config system ha
    set mode a-p    //仅适用于A-P HA集群，其他部署模式（Active-Active集群）不适用//
    set logical-sn enable    //必须启用以支持共享许可证//
end
```

3. HA 与 FortiGuard 同步完成后，通过如下 CLI 可以看到 HA 的 `logical serial number`，这就是 HA 的 vSN。

```bash
FortiGate-70F # get system ha status
HA Health Status: OK
Model: FortiGate-80F
Mode: HA A-P
Group Name: Branch1-HA
Group ID: 100
Debug: 0
Cluster Uptime: 0 days 2h:33m:2s
......
vcluster 1: work 169.254.0.1
Primary: FGT70FTK22023xxx, HA operating index = 0
Secondary: FGT70FTK20000xxx, HA operating index = 1
Logical Serial Number: FGT70FHA24090xxx

FortiGate-70F # diagnose system ha dump-by debug-zone
            HA information.
is_manage_primary=1,manage_vd=root,ip=169.254.0.1,num=2,nvcluster=1,jiffies=938038.
logical serial number is FGT70FHA24090xxx,
local serial number is FGT70FTK22023xxx,
member's serial number is FGT70FTK20000xxx
```

4. 如果 HA 未启用 `logical-sn`，License 将会无法更新成功，`diagnose debug application update -1` 会有以下报错。

```bash
upd_status_extract_contract_info[1309]-pending registration(1) support acct() company() industry()
upd_status_extract_contract_info[1317]-valid contract percent=11%

installUpdObjRest[846]-Step 9:Delete backup /tmp/update.backup
upd_install_pkg[1404]-FSCI000(contract) installed successfully
upd_pkg_get_obj_resp_code[1669]-No response code for obj FSSI000
upd_install_pkg[1372]-Failed to find resp code for FSSI000
upd_status_save_status[135]-try to save on status file
upd_status_save_status[201]-Wrote status file
__upd_act_update[312]-Failed installing pkg
upd_comm_disconnect_fds[498]-Disconnecting FDS 149.5.232.66:443
[206] __ssl_data_ctx_free: Done
[1094] ssl_free: Done
[198] __ssl_cert_ctx_free: Done
[1104] ssl_ctx_free: Done
[1085] ssl_disconnect: Shutdown
upd_act_HA_contract_info[747]-Error updating FSCI -1
__update_upd_comp_by_settings[495]-Disabling NIDSDB/ISDB/MUDB components.
do_update[720]-UPDATE failed
calculate_next_update_time_auto[192]-interval=60, cur_time=Fri Jun 13 18:21:00 2025
, next_update=Fri Jun 13 18:21:00 2025
```

5. 此外，“系统管理 → FortiGuard”和“系统管理 → 高可靠性”页面也可以查看 HA 的 vSN。

注意事项

1. 如果 HA 型号的设备加入了 FortiGate Cloud，如下所示。




```bash
config system central-management
    set type fortiguard
end
```

2. FortiGate Cloud 可能会自动向 HA 设备推送 HA 配置（不需要手动配置，使用的 HA 心跳接口为设备接口数字最大的接口），如下所示，其中也包含 `logical-sn` 配置。

```bash
config system ha
    set group-id 994
    set group-name "ZTP-VSN-HA-194"
    set mode a-p
    setpassword ENC xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    set hbdev "lan3" 50
    set override disable
    set logical-sn enable
end
```

3. 默认配置下，推送的 HA 配置无法修改（即使手动修改也会被 FortiGate Cloud 推送的模板配置覆盖）。如果需要 HA 两台设备单独使用，或自定义 HA 配置，则需要关闭 FortiGate Cloud 的配置推送功能（默认开启），之后就可以手动配置 HA 功能。

```bash
config system central-management
    set allow-push-configuration disable
end
```

4. 也可以关闭 FortiGate Cloud 管理功能。

```bash
config system central-management
    set type none
end
```

HA 的 ping server 配置

功能介绍

1. Ping Server 功能是为了防止所谓端口“假死”的问题，链路状态正常，但链路无法工作，防火墙可以通过利用发送 ping 包，根据对方的响应来判断该端口链路是否可用。

2. 配置命令如下：

```
config system link-monitor
    edit "wan1"
        set srcintf "wan1"
        set server "100.1.1.254"
        set failtime 3
        set ha-priority 5
    next
end
```

3. 如果只配置上述的配置，ping 检测失败时 HA 不会切换，只是去往该接口的路由不再有效。需要在HA 配置中告诉防火墙 wan1 口的 pingserver 会作为 HA 切换的触发条件。

```bash
config system ha
    set pingserver-monitor-interface wan1     //监视wan1口上的pingserver//
    set pingserver-failover-threshold 0         //ha切换的阈值,默认为0//
    set pingserver-flip-timeout 60               //连续2次ping server触发的HA切换的间隔，单位为分钟，在这个时间内，ping server不会连续触发HA切换//
end
```

HA 相关配置

1. set ha-priority 1 与 pingserver-failover-threshold 0 相关联。两者相比较，当 wan1 接口 pingserver 检查失效后赋值增加 1，已经达到了阈值 0，所以触发 HA 会切换。
2. 如果 pingserver-failover-threshold 2，即使该接口 wan1 的 link-moniter 检测失败，set ha-priority 1 小于 pingserver-failover-threshold 2，达不到触发的阈值，HA 不会切换。

HA 配置命令集

HA 配置命令 config system ha，如下是常用的配置命令。

配置参数	参数说明
set group-id 0	配置 HA 机群的组 ID，一个机群内的成员必须有相同的组 ID。该 ID 会成为生成防火墙接口的的虚拟 MAC 的一个组成因素，因此当同一个广播域有 2 组以上的 HA 机群的时候需要配置不同的组 ID, 防止 MAC 地址冲突
set group-name "FGT - HA"	一个集群内的成员必须有相同的组名字
set mode standalone/a-a/a-p	HA 工作模式，常用为 a-p 模式。AA 模式下在 HA 状态中查看到 HA 的角色，有主设备及从设备, 通常会被认为工作在主被模式下, 实际上主主下设备虽然都在工作, 仍会有一台作为集群的主设备用来控制和分配流量和会话给集群中的其他设备。AA 模式默认情况下仅负载均衡 UTM 的流量, 所以在下不使用 UTM 功能时建议使用 AP 模式
set password	一个集群内的成员必须有相同的密码
set hbdev "port1" 50 "port2" 50	配置心跳接口。其中 50 为优先级，优先级高的被优先使用
unset session-sync-dev	可以配置专门的心跳接口用于会话信息同步，默认和控制信息为同一心跳接口
set route-ttl 10	当一个备机被选举成主机后，其从原主机同步来的转发表的存活时间，单位秒。 HA 设备之间只同步转发表，不同步路由表。随后新主机通过静态或动态路由协议生成新的转发表，继续工作。 建议在 HA + 动态路由环境下，配置为较长时间（可以覆盖新主机动态路由邻居建立与学习的时间，如 120s）。
set route-wait 0	主设备收到新的路由条目后，会等待 x 秒后，再同步给从设备
set route-hold 10	主设备进行 2 次路由同步之间的间隔，防止路由震荡而造成反复更新路由
set sync-config enable	配置文件自动同步，需要开启
set encryption disable	是否允许使用 AES-128 和 SHA1 对心跳信息进行加密和完整性验证
set authentication disable	是否使用 SHA1 算法验证心跳信息的完整性
set hb-interval 2	发送心跳数据包的间隔，单位为每 100ms. 如配置 2, 则每 200ms 发送一个心跳信息
set hb-lost-threshold 6	心跳信息连续丢失 6 个后则认为对方不再存活
set helo-holddown 20	Hello 状态时间。设备加入 HA 机群前等待的时间，防止由于未能发现所有的机群成员而造成 Ha 的反复协商
set arps 5	设备成为主设备后，要发送免费 arp 来宣布自己的 MAC 地址，以便相连的交换机能够及时更新 MAC 地址表，该参数用于配置其发送的数量
set arps-interval 8	发送免费 arp 的间隔，单位秒
set session-pickup enable/disable	关闭或者开启会话同步，默认为 disable。一般需要开启
set session-pickup-delay {enable/disable}	仅对存活 30 秒以上的会话进行同步。开启后会对性能有所优化，但小于 30 秒的会话在 HA 切换的时候会丢失。默认为关闭，谨慎使用
set link-failed-signal disable	防火墙上发生被监控端口失效触发 HA 切换的时候，是否将除心跳口外的所有端口 shutdown 一秒钟的时间，便于与之相连的交换机及时更新 MAC 表。
set uninterruptable-upgrade enable	是否允许无中断升级 OS。系统自动分别对机群内的设备升级，并自动切换，不会造成业务的中断
set ha-uptime-diff-margin 300	当进行 HA 选举时，启动时间为一个选举的一个参数，当 2 台设备启动时间差小于 300 时则将该部分差异忽略，视为相同
set override disable	默认为 disable,H A 选举按如下顺序进行比较：有效接口数量 > 运行时间 > HA 优先级 > 设备序列号。Enable 情况下，讯据顺序改变。有效接口数量 > HA 优先级 > 运行时间 > 设备序列号。每次设备加入或者离开机群，都会强制整个机群重新进行主设备的选举
set priority 128	HA 优先级，为便于管理，建议主设备 200, 从设备 100

set monitor port3 port4	配置需要被监控的端口，其有效数量多的设备成为主设备
unset pingserver-monitor-interface	是否设置 pingserver 监控端口
set pingserver-failover-threshold 0	pingserver 触发的阀值，0 则意味着任何的 pingserver 失效都会触发 HA 的切换
set pingserver-flip-timeout 60	两次 pingserver 失效切换之间的间隔。如 A 发生失效，切换到 B. 切过去之后发现 B 也是失效的，则需要等待 60 分钟的时间允许切换回 A
set ha-mgmt-status enable HA 的带外管理配置命令	set ha-mgmt-status enable set ha-mgmt-interface port1 set ha-mgmt-interface-gateway x.x.x.x

异步流量负载分担（FGSP）

请参考如下文档配置：

https://handbook.fortinet.com.cn/files/FGSP配置指导.pdf ↗

HA-Cluster 的网管

HA 的网管介绍

HA 集群普通管理

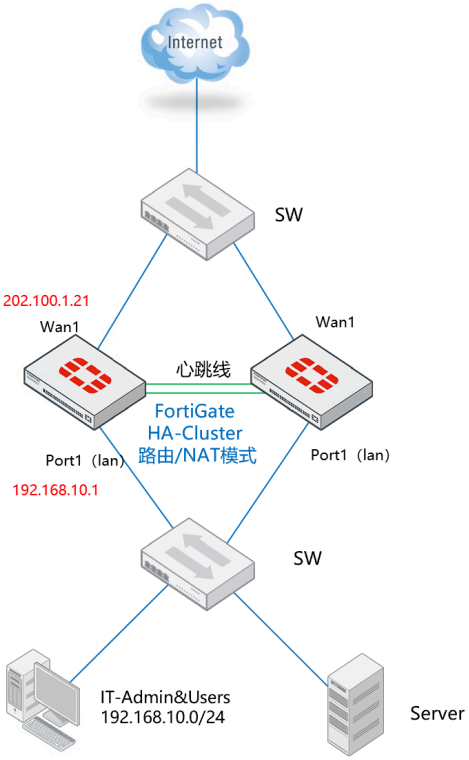
HA 集群带外管理

- 独立管理接口
- 独立管理 VDOM
- 单独指定用于更新的 VDOM

HA 的网管介绍

使用业务接口进行网管

1. 对网管要求不是太高的情况下推荐使用。



网管方式1：直接使用lan业务接口进行网管：
https://192.168.10.1
ssh admin@192.168.10.1

2. 优点：

- 够直接、够简单、无需考虑太多网管的因素
- 可以直接使用 mgmt、lan 或 wan 接口进行网管，直接网管即可，只需要开放接口的网管协议即可。（另外需注意管理员的可信任主机配置）

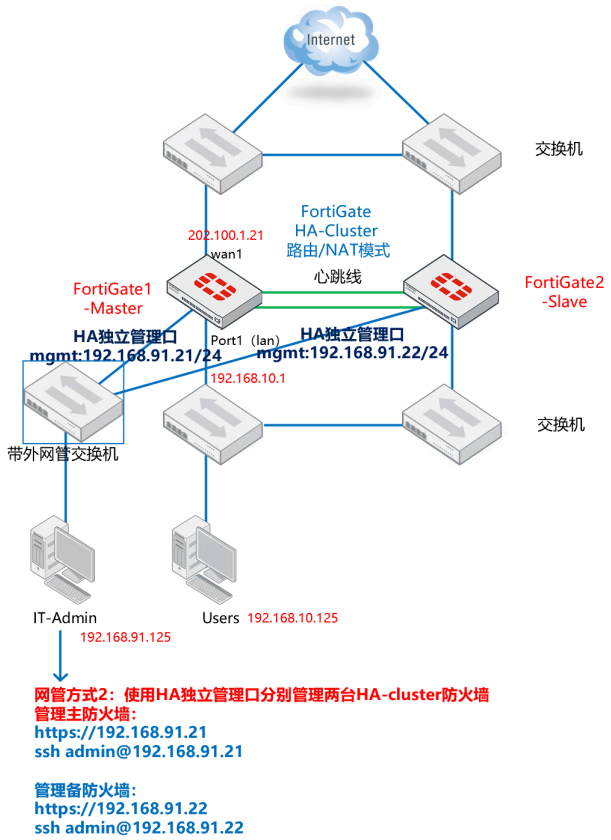
```
config system interface
    edit "port1"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh http fgfm
        set alias "LAN_Inside"
    next
end
```

3. 缺点：

- 管理员无法登陆到备机上进行管理和查看，备机处于一种未知的状态（只能通过主机查看到备机的状态，管理员无法直接网管 HA 备机）
- 对于 HA-Cluster 来说 HA 是一个大的整体，此时的两台防火墙配置完全一样，换一句话说备机是完全不可控/不可登陆，完全处于热备的状态，只有主防火墙出现故障的时候备防火墙才会被激活，而在没有激活之前，管理员是无法登陆到备机上进行管理和查看的。这样就有一种小概率，备机坏了的时候（比如硬件故障），相对来说比较难被发现（可以通过主设备查看 HA 状态可以查看到备机掉线的异常情况）。如果对于设备管理要求较高的话，显然备机的这种未知状态会让人无法完全放心。
- 这就是简单带来的烦恼，既然要解决这样的烦恼，就有了下面的另外两种解决办法，当然这两种办法就不能用简单来形容了，相对要复杂一些。

使用 HA 独立管理接口进行网管

1. 备机管理员可以直接登陆，对 HA 的主机和备机进行单独的管理。



2. 优点：

   - HA Cluster 的主备两台机器都可以独立的进行网管和配置。
   - HA 独立管理口拥有一个完全独立的 IP 转发和路由空间，将网络管理区域和业务数据区域的 IP/路由完全独立，二者相互不影响。相当于在后台创建了一个 vsys_hamgmt 的 vdom，是一个轻量级的 vdom。GUI 是隐藏不可见的状态，让用户尽量较少的感受到此 vdom 的存在，意在简化用户对于独立管理口的使用和理解。命令行下实际可以进入到 vsys_hamgmt 进去 ping 等测试，而 GUI 是几乎是感受不到 vsys_hamgmt 的存在的。

     ```
     FGT-LEFT # execute enter ?
     <name>    vdom name
     vsys_hamgmt    //HA独立管理口轻量级vdom
     root    //业务数据的vdom
     ```

     我们知道业务接口的配置 HA-Cluster 之间是完全同步的，但 HA 独立管理口的配置 HA-Cluster 之间是不同步的，这样才可以确保 HA-Cluster 各自的独立管理口 IP 地址可以不一致，以便管理员可以通过不同的管理 IP 分别管理 HA 的主和备。
   - 基于以上特性，我们可以知道，root 和 vsys_hamgmt 拥有独立的 IP/路由空间，因此独立管理口的 IP 和业务接口的 IP 地址是可以重叠使用的（允许 IP/网段冲突），这个部分可以按照个人需求进行操作，不一定非要如此 IP 网段冲突的设置。比如业务接口 lan 的 IP 为 192.168.1.99，这个时候其实可以将独立管理口 mgmt 的 IP 设置为 192.168.1.198（主）、192.168.1.199（备），二者可以在同一个网段，管理员可以通过 192.168.1.198 管理到主防火墙，192.168.1.199 管理到备防火墙。而 192.168.1.99 这个业务接口地址则永远只属于当前 HA-Cluster 的主防火墙。通常来说我们都是不建议管理和业务网段 IP 地址重叠/冲突的，这样不便于设计、理解、排错，非不得已不要这样使用。

3. 缺点：

   - 独立管理口网管是独立的，但是 fortiguard 的更新无法使用独立管理 mgmt 进行更新，还是需要通过业务接口进行 fortiguard 服务的更新。
   - 另外关于 syslog、Radius 等防火墙主动往外发起的业务，默认也是使用业务接口进行处理的，如果有使用独立管理口进行上述业务的需求，需要在 HA 下面开启 `set ha-direct enable`，这样由 FGT - LEFT 本地发起的数据才会通过独立管理口 mgmt 去发送。

     ```
     FGT-LEFT # config system ha
     …
         set ha-direct ?

         enable    Enable using ha-mgmt interface for syslog, remote authentication (RADIUS), FortiAnalyzer, FortiSandbox,
     sFlow, and Netflow.
         disable    Disable using ha-mgmt interface for syslog, remote authentication (RADIUS), FortiAnalyzer,
     FortiSandbox, sFlow, and Netflow.    //默认为disable，使用业务接口处理以上业务。有需要独立管理的mgmt发送数据的时候才将其enable
     end
     ```

     以上无疑增加了业务和管理的复杂性，需要考虑的东西更多，容易犯错的机会也就更大。

4. HA 独立管理口的使用建议：

   - 如果仅仅是单纯的管理 HA-Cluster 的主和备防火墙，这个方法还是挺合适的
   - 如果有更多更复杂的需求，则需要搞清楚 HA 独立管理口的工作逻辑，比如 Radius 请求/syslog 日志是由谁发送的问题。注意命令适当的时候需要开启“set ha-direct enable”命令。可以通过 sniffer 抓包协助判断 HA 场景下的 FGT - LEFT 主动往外发包情况，以便确认是否是你需要的发包效果：# diagnose sniffer packet any "port 1812" 4 0 l （抓取防火墙发起的 Radius 认证数据）

     > **重要**
     >
     > 开启 HA 配置下的“HA-direct”后，主机和备机的 syslog、FortiAnalyzer、LDAP、Radius、TACACS+、sFlow、NetFlow、NTP 等报文均会通过各自的独立管理口发出，而不是通过主机的业务口发出。

   - 另外如果需要使用独立管理口 mgmt 进行 fortiguard 更新的话，则必须使用更加彻底的独立管理 VDOM（方法 3），将管理与业务的完全隔离，那个也是最彻底的方法，当然相对也更加复杂一些。（DNS 解析和 Fortiguard 更新还是走的业务接口）

5. HA 独立管理口命令行配置。

```
主防火墙：
config system ha
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "mgmt"
            set gateway 192.168.91.254
        next
    end
end
config system interface
    edit "mgmt"
        set ip 192.168.91.21 255.255.255.0
        set allowaccess ping https ssh http
        set alias "HA_Dedicated_MGMT"
    next
end

备防火墙：
config system ha
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "mgmt"
            set gateway 192.168.91.254
        next
    end
end
config system interface
    edit "port1"
        set ip 192.168.91.22 255.255.255.0
        set allowaccess ping https ssh http
        set alias "mgmt"
    next
end
```

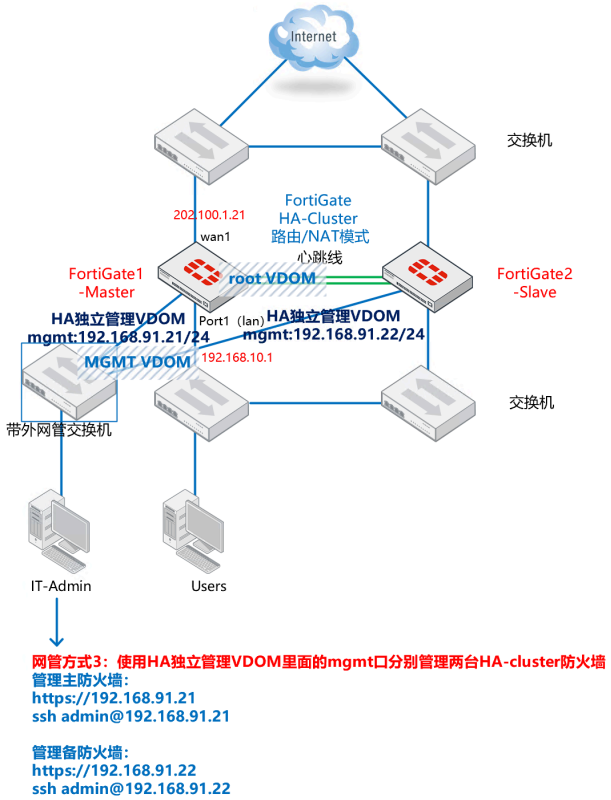
6. HA 独立管理口 GUI 配置。



使用独立 VDOM 进行网管

1. HA 独立管理口可以简单的理解是一个轻量级的 VDOM 隔离，而 HA 独立管理 VDOM 则是一个完完全全的真实 VDOM 隔离。如果需要将 HA-Cluster 的管理和业务彻底隔离开，则建议使用 HA 独立管理 VDOM。

2. HA 独立管理口 + 配置了 set ha-direct enable 其实实现了一部分的独立管理 VDOM 的功能，但是还是不够彻底，比如 fortigaurd 更新等需求，因此这就是独立管理 VDOM 存在的原因。



3. 优点：
   - 完全的管理和业务独立，管理是管理 VDOM，业务是业务 VDOM，完全隔离，最符合隔离管理的需求。
4. 缺点：
   - 需要开启 VDOM 特性，配置相对会复杂一些。
5. 独立管理 VDOM 的配置，首先需要开启 VDOM 特性。

```
FGT-LEFT # config system global
FGT-LEFT (global) # set vdom multi-vdom
FGT-LEFT (global) # end
You will be logged out for the operation to take effect
Do you want to continue? (y/n)y
Auto backup config ...
exit
FGT-LEFT login:
FGT-LEFT login: admin
Password: ********
Welcome !
```

6. 创建新的 MGMT 独立管理 VDOM。




```
FGT-LEFT # config vdom
FGT-LEFT (vdom) # edit MGMT
```

```
FGT-LEFT (MGMt) # end
```

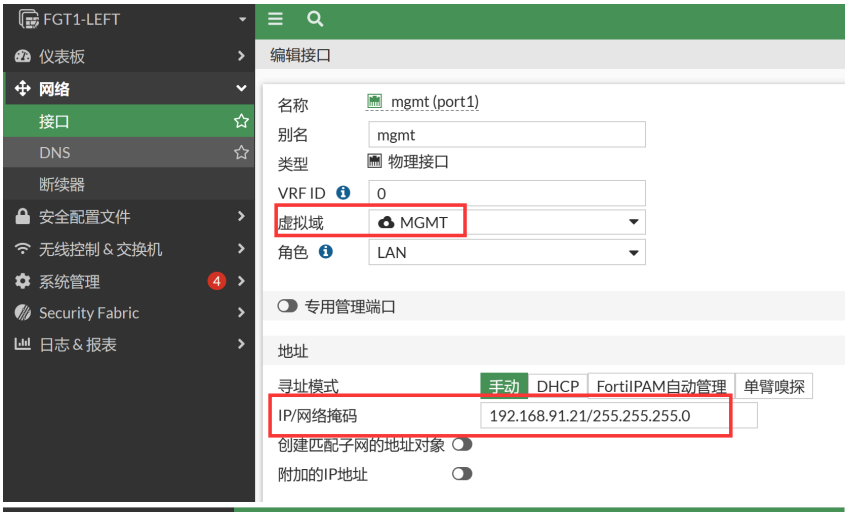
7. 将 MGMT 独立管理 VDOM 修改为管理 VDOM（只能命令行修改）。

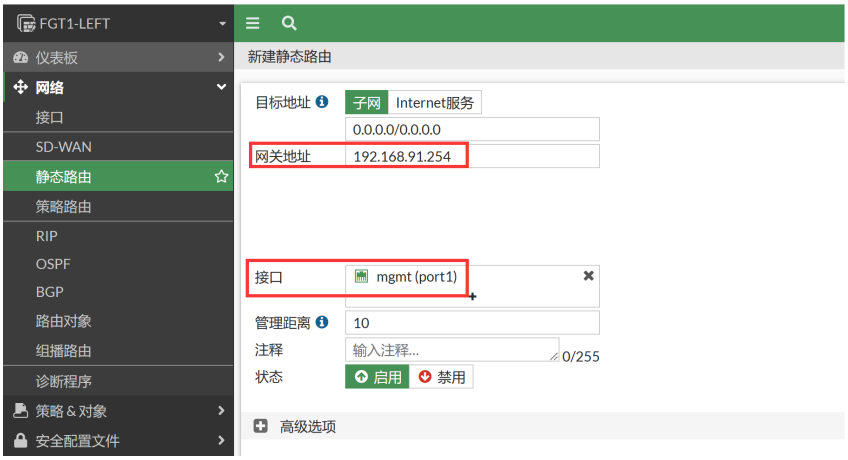
```
FGT-LEFT # config global
FGT-LEFT (global) # config system global
FGT-LEFT (global) # set management-vdom
<string>    please input string value
MGMT    vdom
root    vdom
FGT-LEFT (global) # set management-vdom MGMT
FGT-LEFT (global) # end
FGT-LEFT (global) # end
```

8. HA 里面指定 MGMT VDOM 为独立管理 VDOM。

```
FGT-LEFT # config  global
FGT-LEFT (global) # config system ha
FGT-LEFT (ha) # set standalone-mgmt-vdom enable     //此配置HA的主备同步，在主设备配置会自动同步到备设备上
FGT-LEFT (ha) # end
```

9. 将 mgmt 接口加入到 MGMT VDOM 中，并配置上管理 IP 和默认路由。






```
配置mgmt加入到VDOM-MGMT以及配置管理IP地址：
FGT-LEFT # config global
FGT-LEFT (global) # config system interface
FGT-LEFT (interface) # edit port1
FGT-LEFT (mgmt) # set vdom MGMT
FGT-LEFT (mgmt) # set ip 192.168.91.21/24
FGT-LEFT (mgmt) # set alias "MGMT"
FGT-LEFT (mgmt) # end
FGT-LEFT (global) # end

配置默认路由：
FGT-LEFT # config vdom
```

```
FGT-LEFT (vdom) # edit MGMT
current vf=MGMT:1
FGT-LEFT (MGMT) # config router static
FGT-LEFT (static) #
FGT-LEFT (static) # edit 0
new entry '0' added
FGT-LEFT (0) # set gateway 192.168.91.254
FGT-LEFT (0) # set device port1
FGT-LEFT (0) # end

FGT-LEFT (MGMT) # execute ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=255 time=0.2 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=3 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=4 ttl=255 time=0.1 ms
--- 192.168.91.254 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.2 ms
```

10. 此时备机是无法 WEB/SSH 登陆的需要通过主机的命令（或者 Console 线）连到备机的命令行界面。

```
FGT-LEFT # config global
FGT-LEFT (global) # execute ha  manage
<id>    please input peer box index.
<1>     Subsidary unit FG101E4Q17000045

FGT-LEFT (global) # execute ha  manage  1    //从主机登陆到备机的命令行

FFGT-LEFT-RIGHT login: admin
Password: ********
Welcome !

FFGT-LEFT-RIGHT #
```

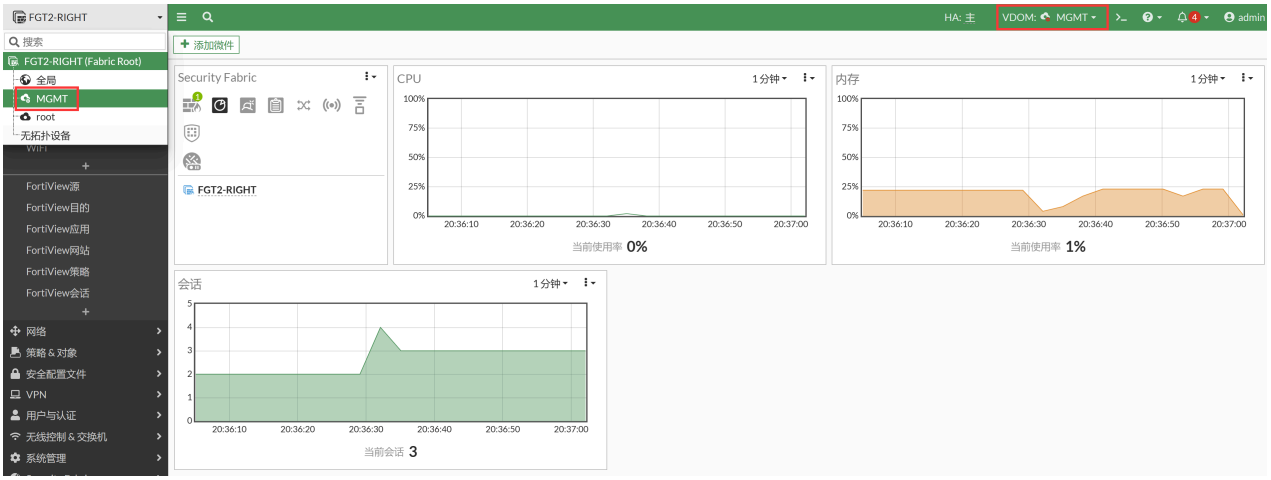
11. 备防火墙那边会自动同步 VDOM 和 mgmt 加入到 MGMT 的配置（如果配置同步不正常，可以将备机重启，并等待配置同步完成，再操作），而独立管理 MGMT - VDOM 内部的配置是不会同步的，因此只需要修改备防火墙管理 VODM、mgmt 的接口 IP 和路由即可。

```
FGT-LEFT # config global
FGT-LEFT (global) # config system global
FGT-LEFT (global) # set management-vdom MGMT
FGT-LEFT (global) # end
FGT-LEFT (global) # end
FGT-LEFT-RIGHT # config vdom
FGT-LEFT-RIGHT (vdom) # edit MGMT
current vf=MGMT:1
FGT-LEFT-RIGHT (MGMT) # config system interface
FGT-LEFT-RIGHT (interface) # edit mgmt
FGT-LEFT-RIGHT (mgmt) # set ip 192.168.91.22/24
FGT-LEFT-RIGHT (mgmt) # set allowaccess https http  ping ssh
FGT-LEFT-RIGHT (mgmt) # set alias "HA_Dedicated_MGMT"
FGT-LEFT-RIGHT (mgmt) # show
FGT-LEFT-RIGHT (mgmt) # end

FGT-LEFT-RIGHT (MGMT) # config router static
FGT-LEFT-RIGHT (static) # edit 0
new entry '0' added
FGT-LEFT-RIGHT (0) # set device mgmt
FGT-LEFT-RIGHT (0) # set  gateway  192.168.91.254
FGT-LEFT-RIGHT (0) # end

FGT-LEFT-RIGHT (MGMT) # execute ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=255 time=0.2 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=3 ttl=255 time=0.1 ms
```

```
64 bytes from 192.168.91.254: icmp_seq=4 ttl=255 time=0.1 ms

--- 192.168.91.254 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.2 ms
```

2. 这个时候可以通过 GUI 登录到备机了（使用备机的独立管理 IP）。

3. 这时候可以通过 192.168.91.21 和 192.168.91.22 分别管理到主防火墙和备防火墙了。



4. 此时所有的 Radius/SYSLOG/DNS/FORTGUARD 都是通过 MGMT 这个 VDOM 进行更新的，完全独立于 root 业务 VDOM，彻底的管理是管理，业务是业务。这是 HA 独立管理的终极解决方案。

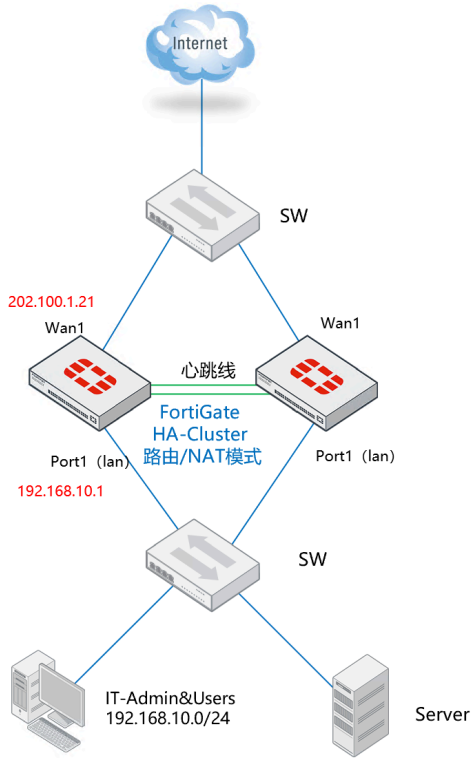
5. 越完美的方案，有时候也就是意味着越复杂，实际部署中需按照个人需求进行选择如何管理 HA-Cluster。

HA 集群普通管理

网络需求

通过业务接口管理和配置 HA 集群。

网络拓扑



网管方式1：直接使用lan业务接口进行网管：
https://192.168.10.1
ssh admin@192.168.10.1

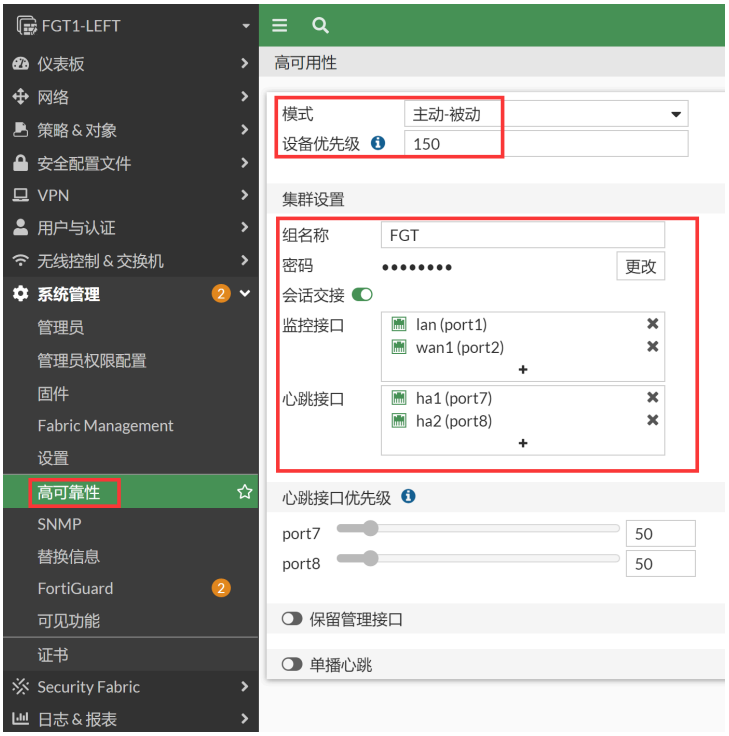
配置要点

- 配置 HA 基础配置
- 配置基础 LAN/WAN 上网服务
- 配置网络管理的相关参数
- 配置防火墙的 SYSLOG、SNMP、FMG

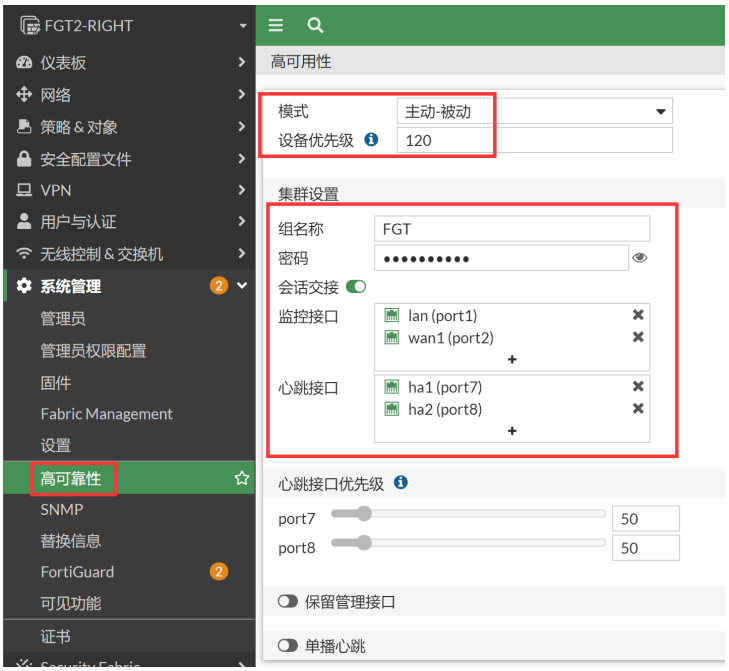
配置步骤和结果验证

HA 基础配置

1. 配置 HA 基础配置，初始化配置防火墙的时候可以通过 mgmt 口登陆到防火墙上，一般 mgmt 口的管理 IP 是 192.168.1.99，第一步可以通过此 IP 登陆防火墙（参考系统管理 → 管理方式 → Web 管理 ↗章节），或者通过 console 口管理防火墙（参考系统管理 → 管理方式 → Console 管理↗章节）。具体型号初始化如果网管和接口的网管 IP 信息等可参考 https://docs.fortinet.com/product/fortigate/hardware ↗ 或 https://docs.fortinet.com/document/fortigate/hardware/fortigate-quickstart-guide-high-end?model ↗ = all 。
2. 按照以上方式登陆第一次开箱的防火墙。先登陆主防火墙，并配置进行 HA 的基础配置，然后登陆到备防火墙，配置备防火墙的 HA 基础配置。
3. 分别配置主防火墙和备防火墙的 HA，主防火墙优先级调整为 150，备防火墙的优先级保持默认的 120，配置 HA-cluster 组名字和密码，监控业务接口 wan1 和 port1，使用 HA1 和 HA2 接口将二者互联。




```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
XU4Yv532eMH20TXkbxDMQWRNs+ChZrIVXcotCFVP2C86QBl2zhznTNU+0PDKw5Vlvtm6B+NCZ+f+1EiT3MsUteDHW82Aten6q+Yu6zAT5NuXYYIKnVpG=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set override disable
    set priority 150
    set monitor "port1" "port2"
end
```




```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
HzpTLoO69D1nrZ9QZbKFDbuUmCefmDRbZqTvgSseAgh2zU8JOHq1bthC750a+p5S/rkfRSROmhVMqOKRbBZkir3C8VWW9l+h7jl7Xp4N5sdA1CmDSUzX=
    set hbdev "port7" 50 "port8" 50
```

```
    set session-pickup enable
    set override disable
    set priority 120
    set monitor "port1" "port2"
end
```

4. 双方的 HA 配置完毕之后将 ha1 和 ha2 线相互直连，将会进行 HA 的选举，此时优先级 150 高的 FGT 将成为主防火墙，优先级低的 120 的 FGT 将成为备防火墙。然后备防火墙的配置将会和主防火墙进行同步，备防火墙所有配置和主防火墙一致，此时备防火墙也将不再可以 WEB/SSH 登录，仅仅可以 Console 登录。后续所有的操作和配置都在主防火墙上的 GUI、SSH 或 Console 完成。

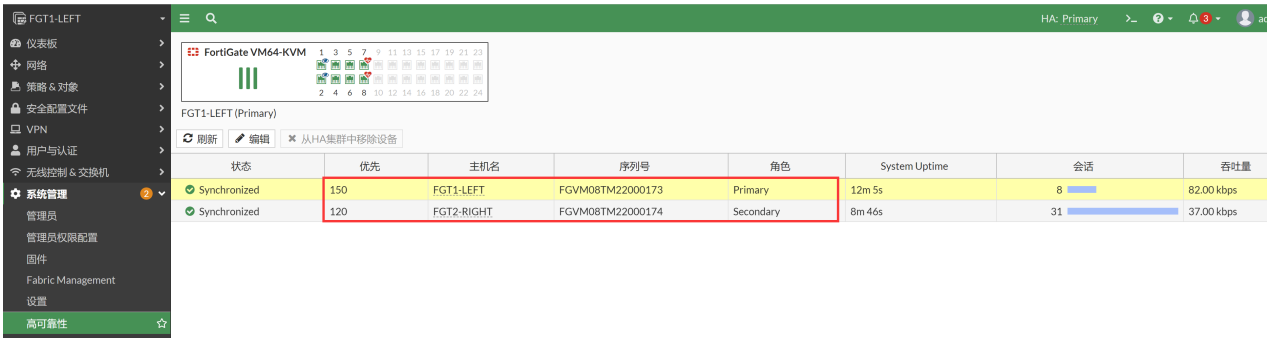
5. 选举结果如下，FGT1-LEFT 成为主设备。

```
FGT1-LEFT # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:22783 b:12890083
        traffic.total = s:0 p:22793 b:12890691
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=0 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM22000173:      Primary, serialno_prio=1, usr_priority=150, hostname=FGT1-LEFT
FGVM08TM22000174:    Secondary, serialno_prio=0, usr_priority=120, hostname=FGT2-RIGHT

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0:
FGVM08TM22000173:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM22000174:    Secondary, ha_prio/o_ha_prio=1/1
```

6. 这三个命令都可以看到 HA 的选举结果和 HA 状态。

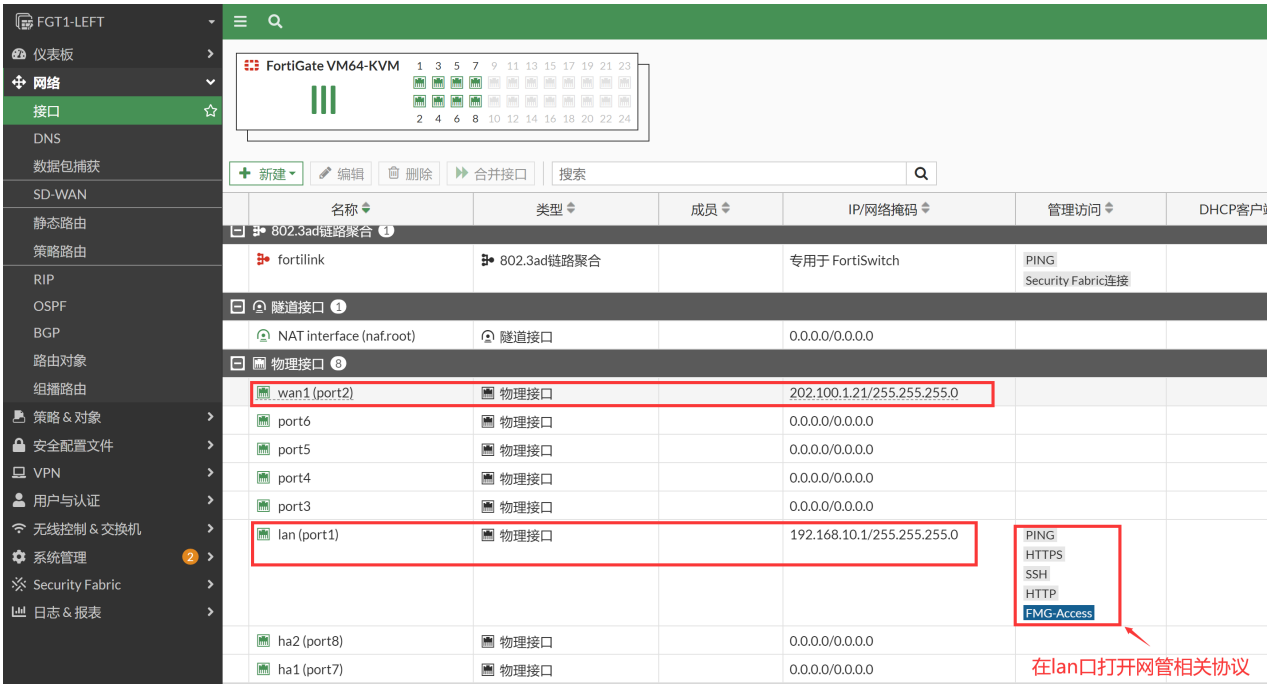
```
# diagnose sys ha status
# get system status
# get system ha status
```

7. GUI 查看 HA 的选举结果和状态。



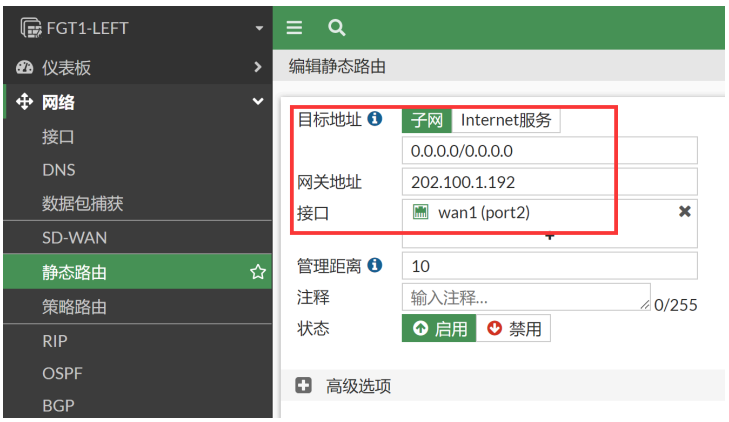
8. 配置基础 LAN/WAN 上网服务。此步骤为正常上网的基础配置，大概的步骤是配置接口 IP、配置默认路由、配置策略即可实现防火墙的简单上网配置。（所有配置都在主防火墙上完成，备防火墙此时不可以登陆）

9. 配置接口 IP。



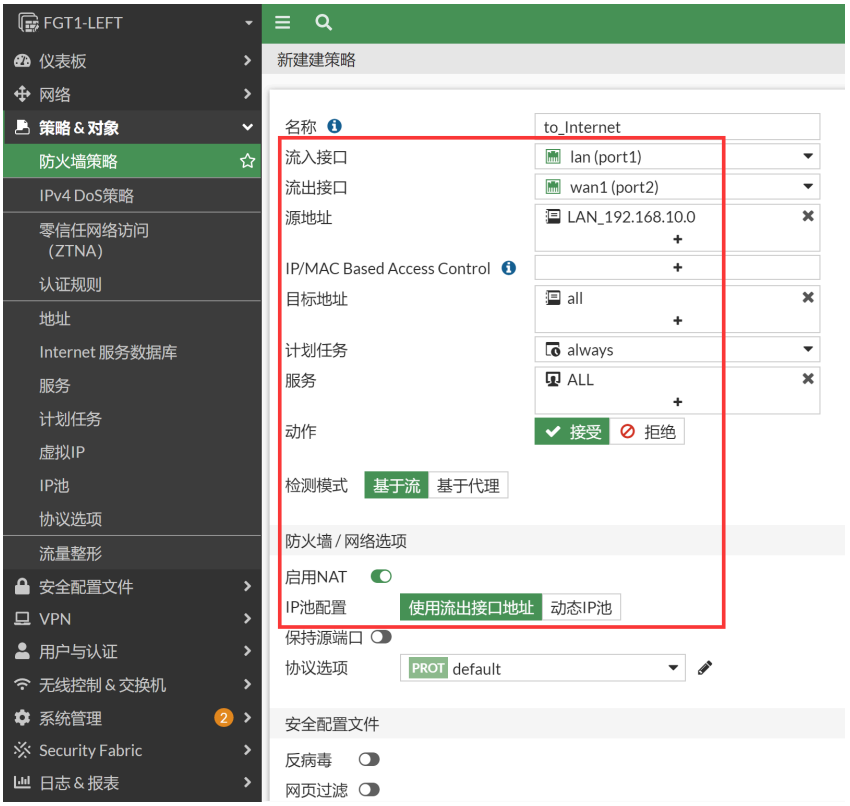

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set ip 202.100.1.21 255.255.255.0
        set type physical
        set alias "wan1"
        set snmp-index 2
    next
end
```

.0. 配置默认路由。




```
config router static
    edit 1
        set gateway 202.100.1.192
        set device "port2"
    next
end
```

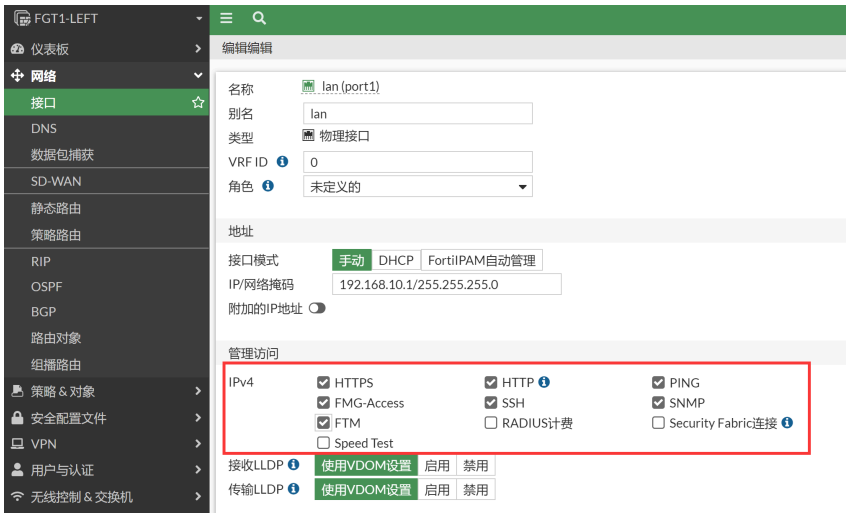
.1. 配置上网策略。



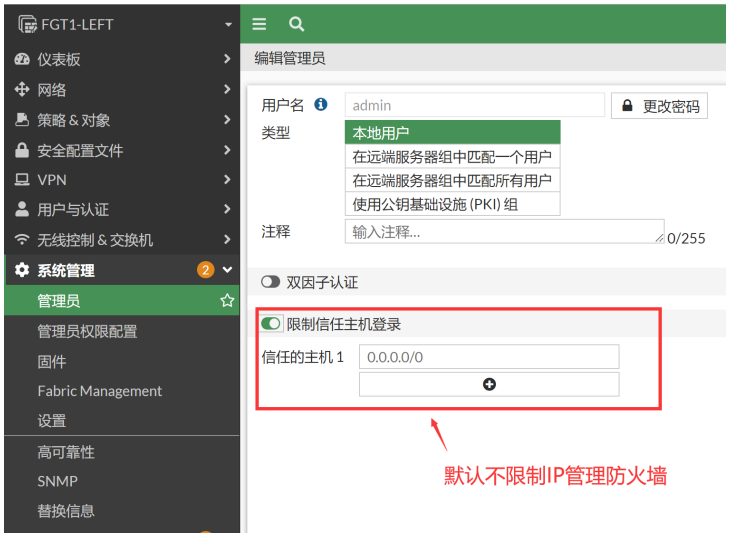

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

2. 配置 LAN 口网管相关参数，关于网管目前可以通过 port1（lan）的 192.168.10.1 进行管理，比较关键的网管因素是：

   - port1 接口下开启了网管的协议，比如 https、http、ssh、snmp 等。



   - 管理员可信任主机如果配置了的话，需要包含发起管理请求的主机 IP 或网段。



- 客户端 PC 到 FGT 的来回路由可达，可以通过 ping 和 tracert 测试。
- 上面 1、2 点确认没有问题之后，还是遇到不通的情况，可以通过 sniffer 抓包和 debug flow 确认问题所在。

```
FGT # diagnose sniffer packet any "host 192.168.10.1" 4 0 l
```

- 如果跨接口管理防火墙，比如处于 port1 的 PC 要通过防火墙 port2 接口的 IP 管理防火墙，则需要主要有 port1 到 port2 的放通策略，否则数据也是不通的。
- 出现网管不了的情况，99% 就是以上的原因了。

网管配置

SYSLOG

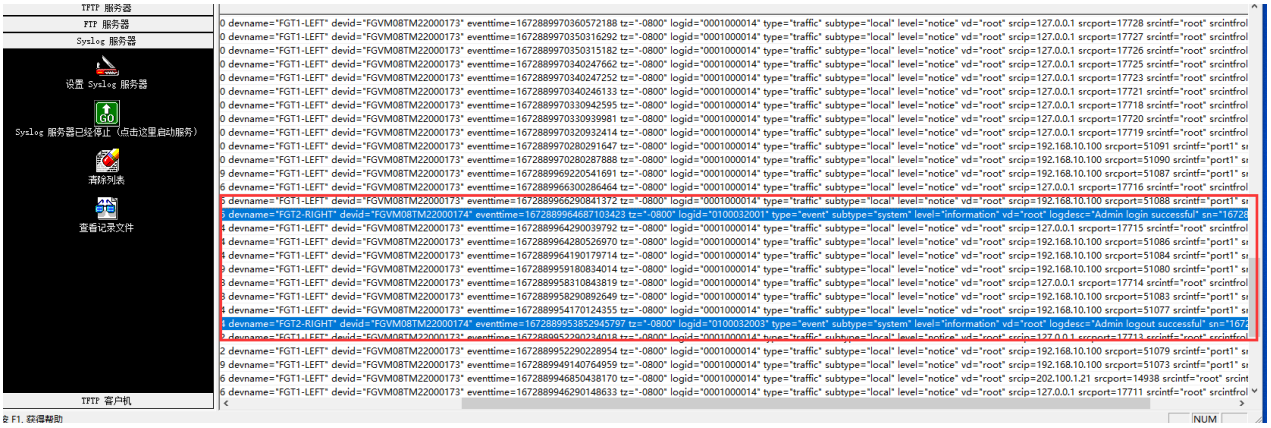
1. 确认通过 192.168.10.1 网管没问题的话，配置 SYSLOG 等本机发出的网管配置。



```
config log syslogd setting
    set status enable
    set server "192.168.10.100"
end
```

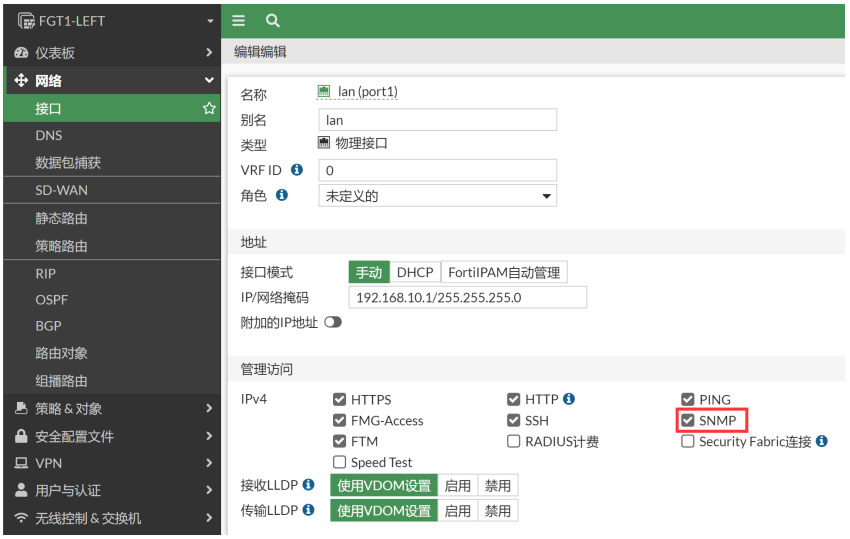
2. 主/备防火墙的日志都会通过业务接口 port1 192.168.10.1 将 syslog 日志发送出去，备防火墙的 syslog 日志会通过心跳线到主机，然后由主机代为转发。

```
FGT1-LEFT # diagnose sniffer packet any "port 514" 4
interfaces=[any]
filters=[port 514]
0.918154 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 635    //主防火墙的syslog日志
0.918225 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 649
5.492057 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 649
5.961902 port_ha in 169.254.0.2.514 → 169.254.0.1.514: udp 454    //备防火墙的console登陆日志
5.962018 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 418
5.968684 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 554
8.314617 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 730
8.507346 port_ha in 169.254.0.2.514 → 169.254.0.1.514: udp 473    //备防火墙的console退出日志
8.507479 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 437
8.964545 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 724
9.074989 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 597
```



SNMP

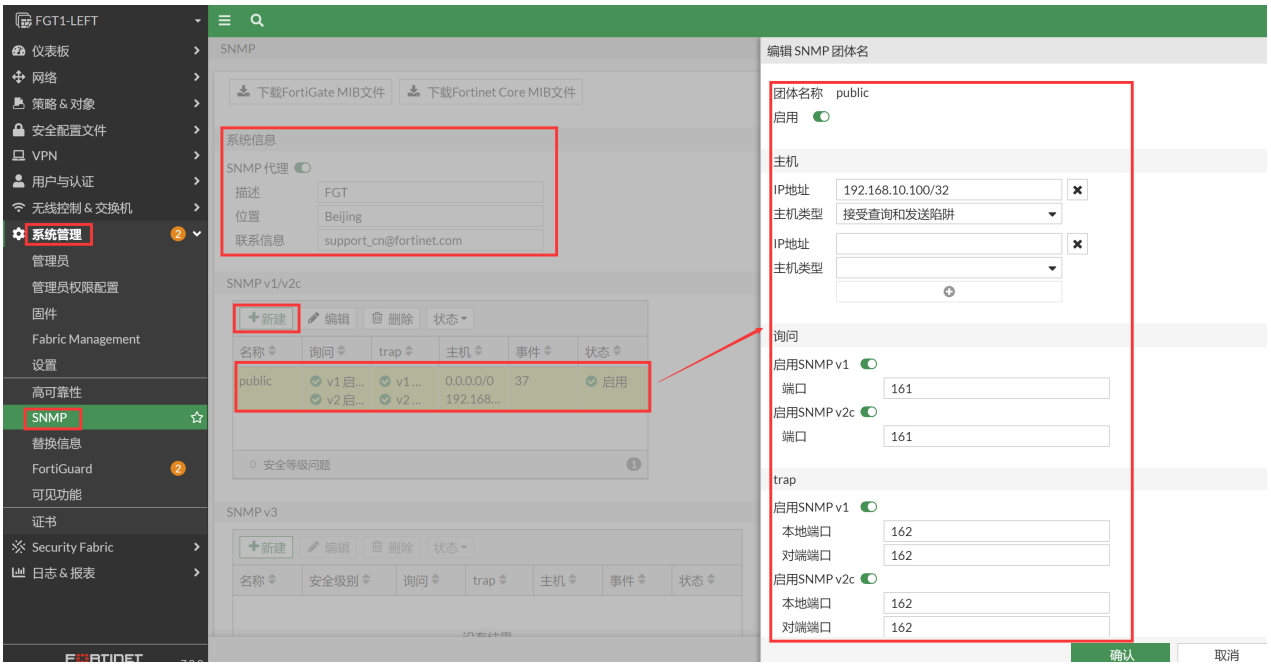
1. lan 接口下开启 SNMP 协议。




```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
```

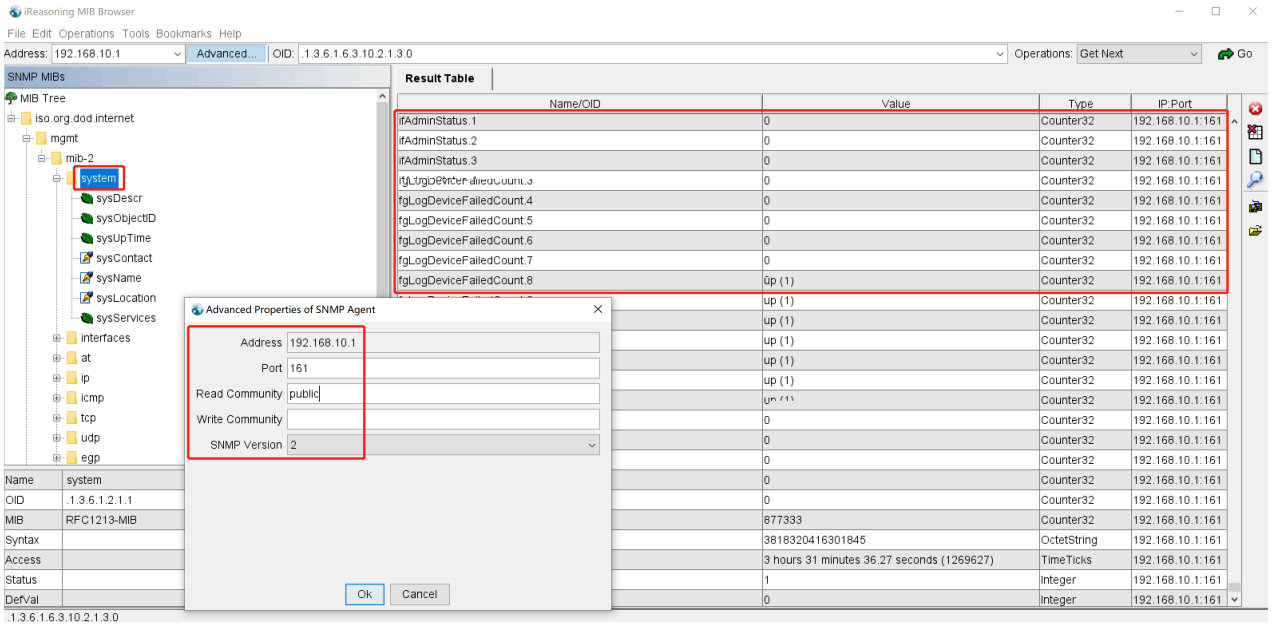
```
        set type physical
        set alias "lan"
        set snmp-index 1
    next
end
```

2. 开启 SNMP 功能。

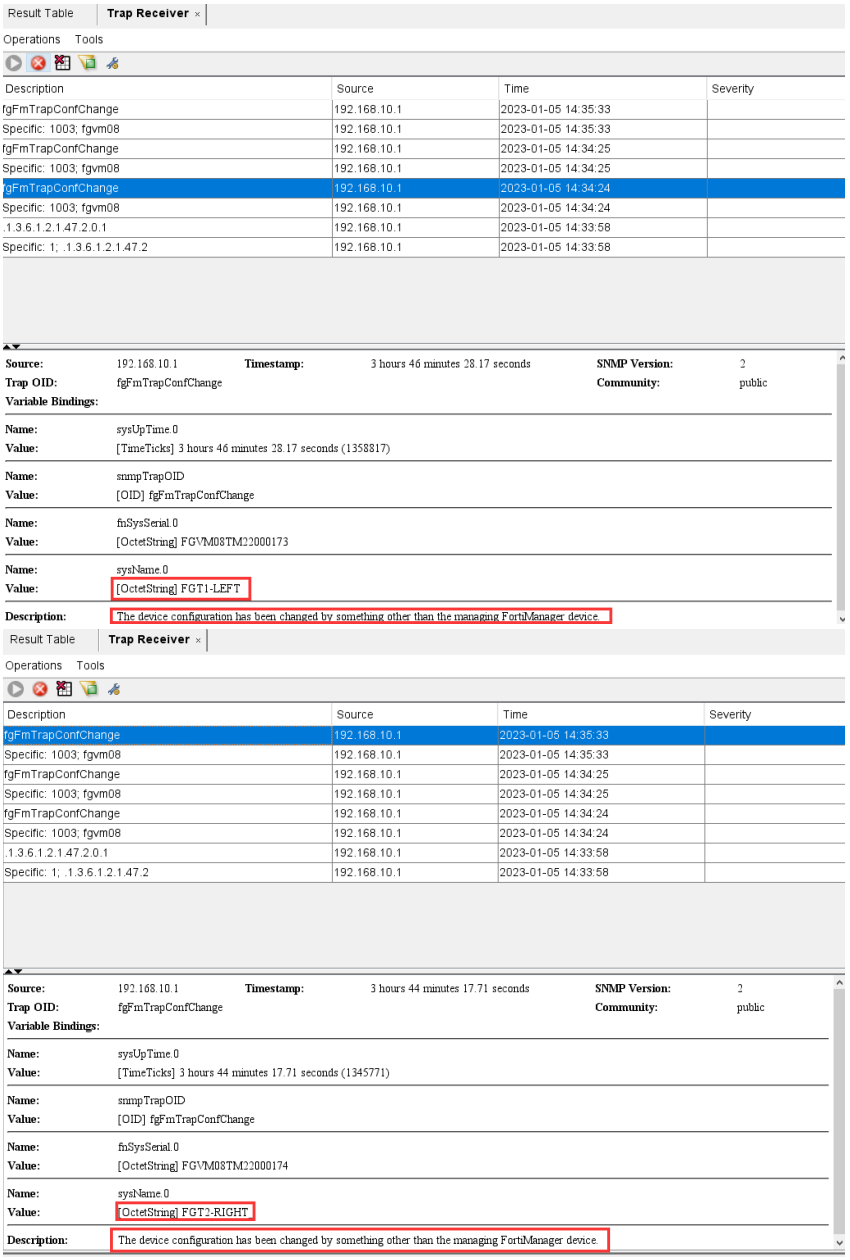



```
config system snmp sysinfo
    set status enable
    set description "FGT"
    set contact-info "support_cn@fortinet.com"
    set location "Beijing"
end

config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ip 192.168.10.100 255.255.255.255
            next
        end
    next
end
```

3. 在 SNMP 客户端对 FortiGate 的 lan 口进行 get/walk 操作，这里以 IReasoning MIB Browser 为例。

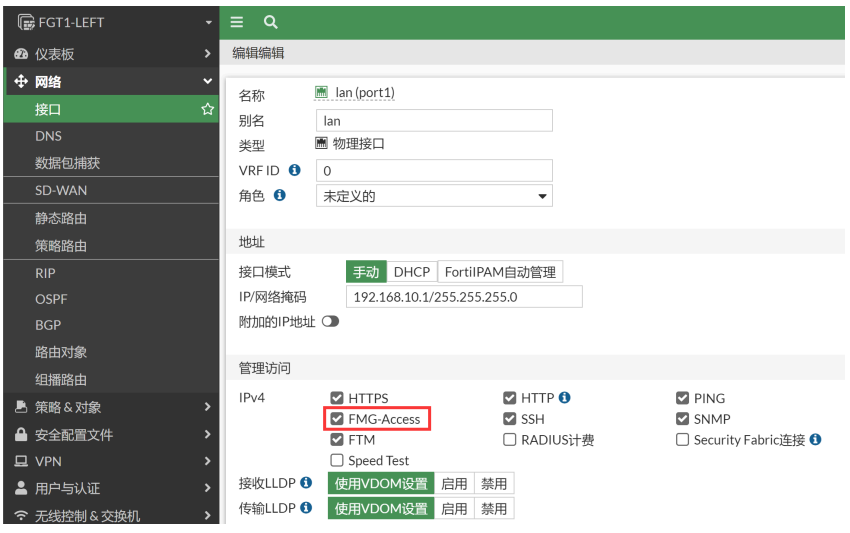


4. SNMP 客户端接收到的 FGT 发送的 trap 信息，可以看到主备设备均通过主设备进行发送。



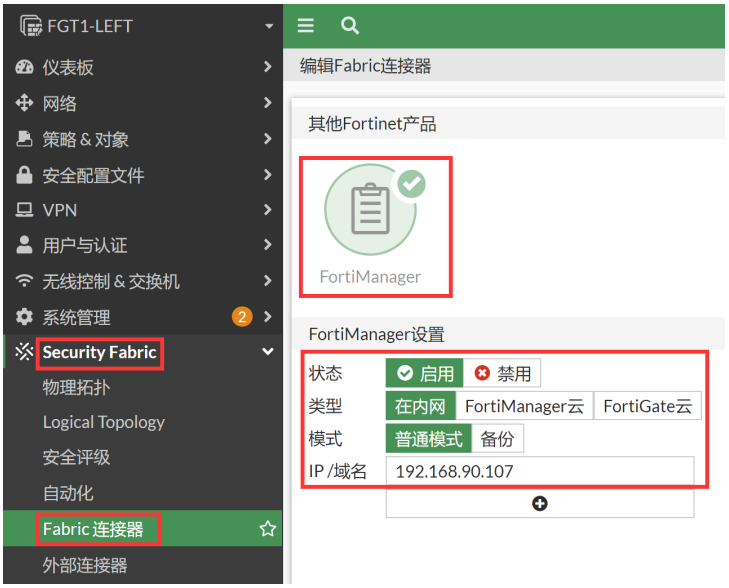
FortiManger 管理

1. FortiManager 通过 FGT 的 lan 口添加设备，在 FGT 的 lan 口下开启 FMG 管理协议。



```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
end
```

2. 配置防火墙主动连接 FMG（如果 FMG 可以主动访问到 FGT，也可以通过 FMG 主动添加 FGT）。

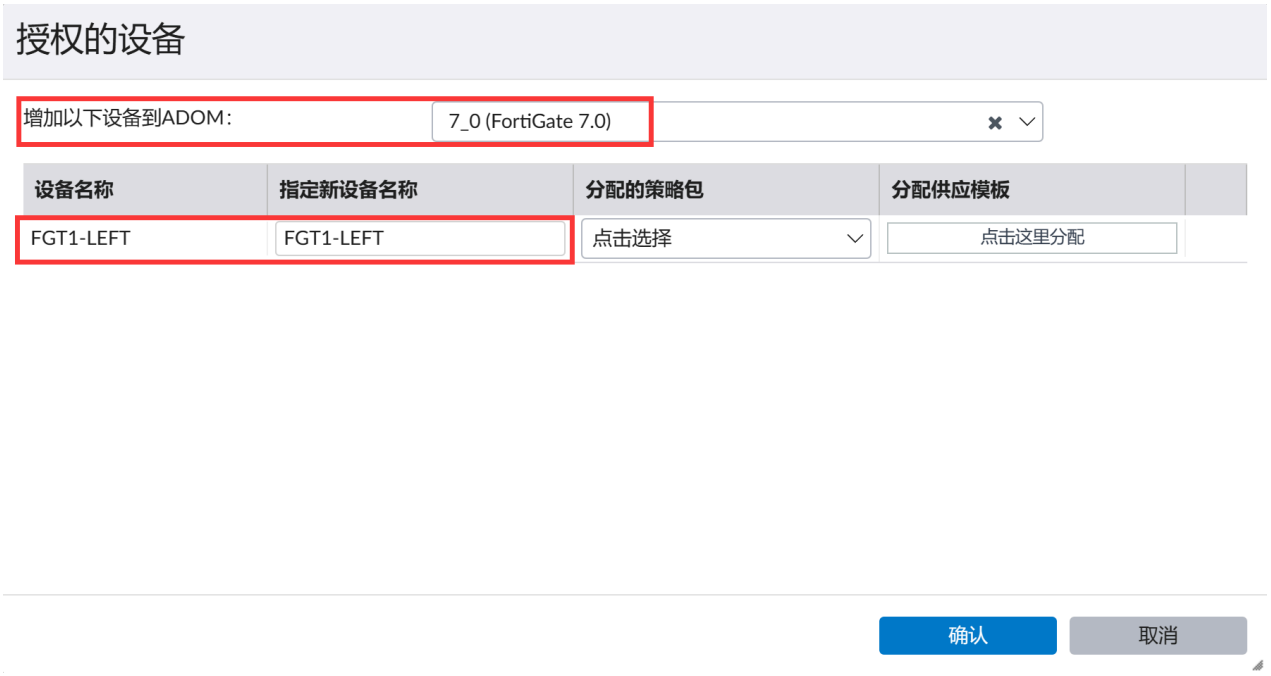



```
config system central-management
    set type fortimanager
    set fmg "192.168.90.107"
end
```

3. 如果是 FGT 主动去注册 FMG，则 FMG 会有未注册的设备提示。



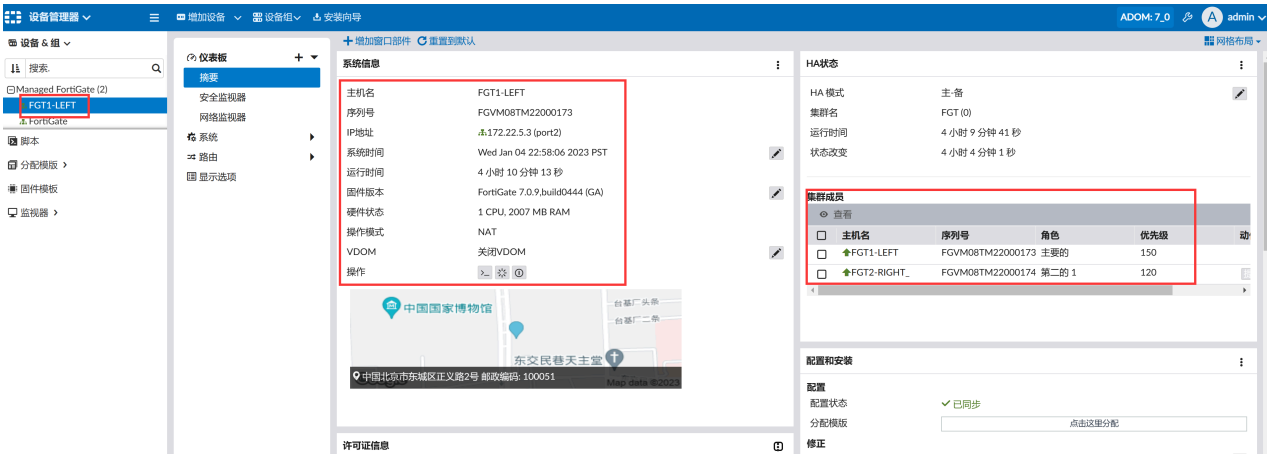
4. 只需要将设备添加到 FMG 的对应 ADOM 的设备管理中即可。



5. 授权完成。



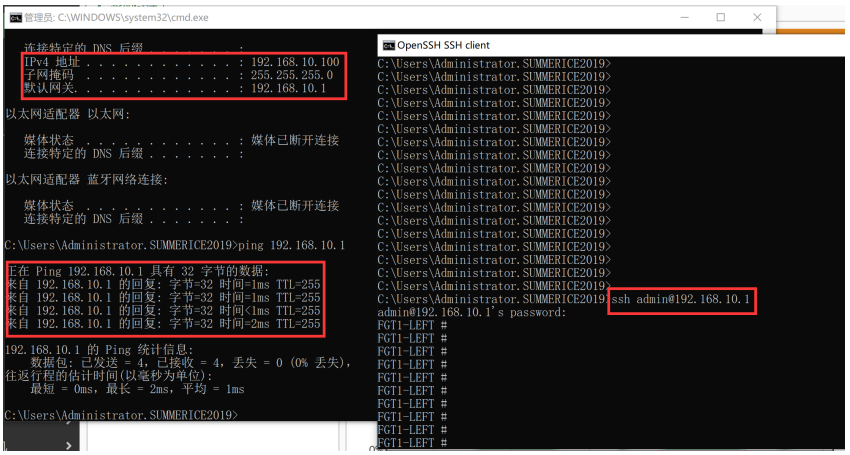
6. 在 FMG 中可以对 FGT 进行管理。



SSH、Ping

1. 在 lan 接口下开启 SSH 和 Ping。

2. 使用 PC 客户端访问 FGT 主设备的 Ping 和 SSH。



HA 集群带外管理

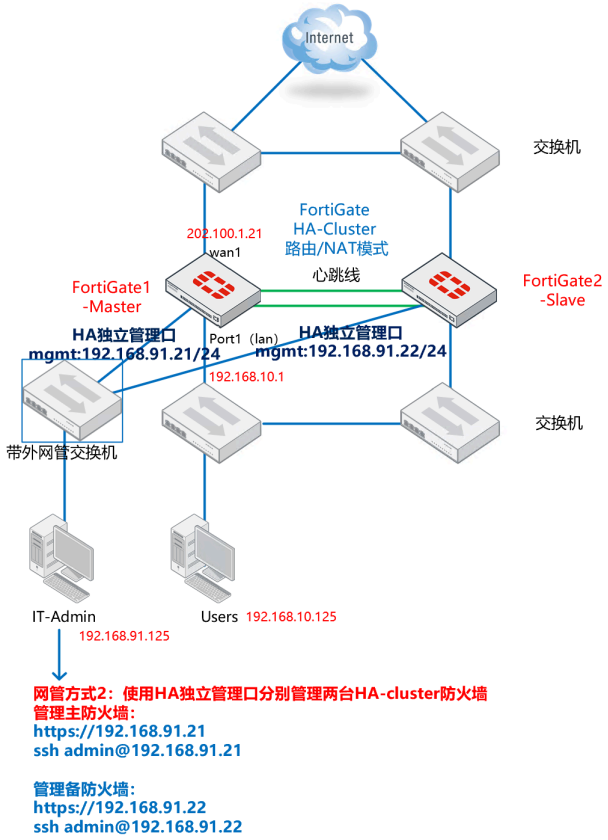
- 独立管理接口
- 独立管理 VDOM
- 单独指定用于更新的 VDOM

独立管理接口

网络需求

1. 在 HA 集群中，所有集群成员的配置全部相同，通过 IP 访问只能管理到主设备，无法通过 IP 访问对每个 slave 设备进行单独的管理。同时为了业务的安全，需要将管理网络与业务网络独立开。
2. 为实现如上需求，可以使用 HA 使用独立的管理接口来实现带外管理功能。独立管理口的相关配置不会被 HA 所同步。

网络拓扑

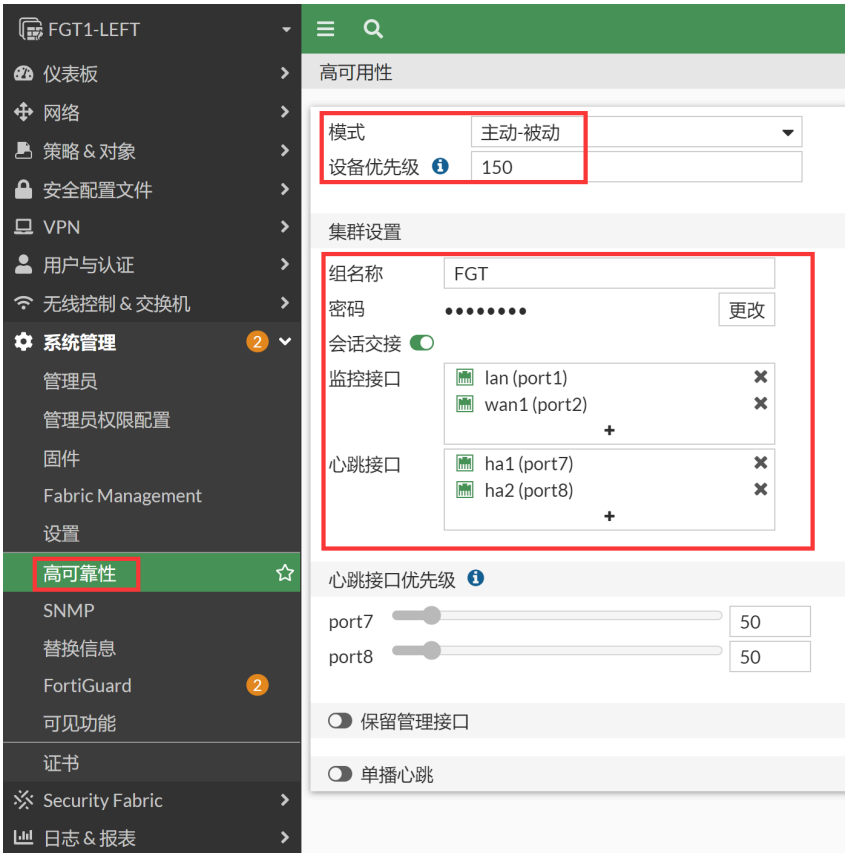


配置要点

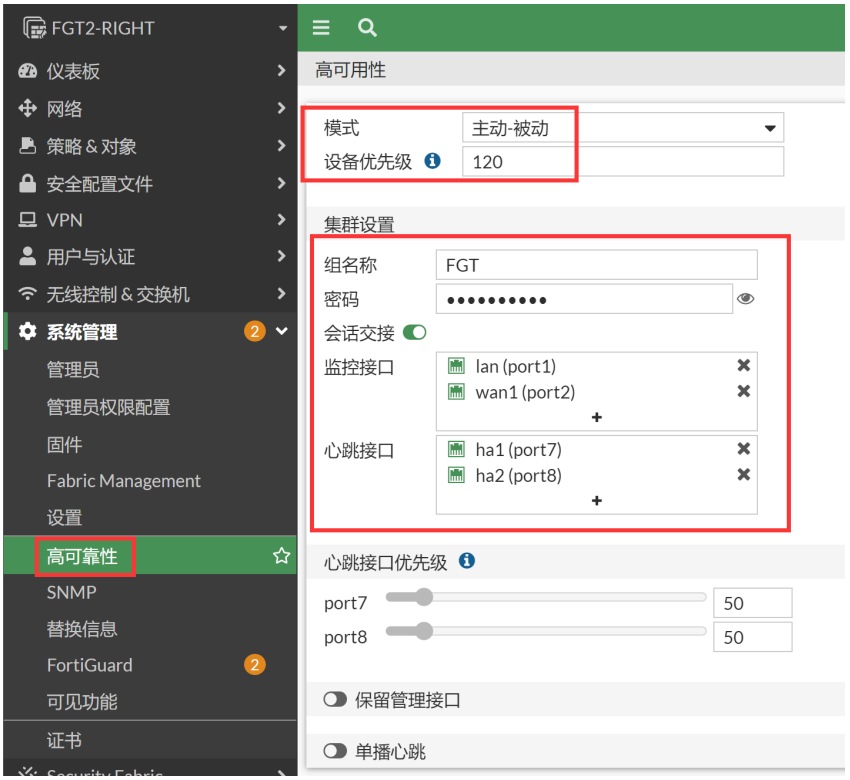
- 配置 HA 基础配置
- 配置基础 LAN/WAN 上网服务
- 配置“HA 独立管理口”
- 配置防火墙的 SYSLOG、SNMP 和 FMG

配置步骤和结果验证

1. 配置 HA 基础配置，初始化配置防火墙的时候可以通过 mgmt 口登陆到防火墙上，一般 mgmt 口的管理 IP 是 192.168.1.99，第一步可以通过此 IP 登陆防火墙（参考系统管理 → 管理方式 → Web 管理 ↗章节），或者通过 console 口管理防火墙（参考系统管理 → 管理方式 → Console 管理↗章节）。具体型号初始化如果网管和接口的网管 IP 信息等可参考 https://docs.fortinet.com/product/fortigate/hardware ↗ 或 https://docs.fortinet.com/document/fortigate/hardware/fortigate-quickstart-guide-high-end?model ↗ = all 。
2. 按照以上方式登陆第一次开箱的防火墙。先登陆主防火墙，并配置进行 HA 的基础配置，然后登陆到备防火墙，配置备防火墙的 HA 基础配置。
3. 分别配置主防火墙和备防火墙的 HA，主防火墙优先级调整为 150，备防火墙的优先级保持默认的 120，配置 HA-cluster 组名字和密码，监控业务接口 wan1 和 port1，使用 HA1 和 HA2 接口将二者互联。




```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
XU4Yv532eMH20TXkbxDMQWRNs+ChZrIVXcotCFVP2C86QBl2zhznTNU+0PDKw5Vlvtm6B+NCZ+f+1EiT3MsUteDHW82Aten6q+Yu6zAT5NuXYYIKnVpG=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set override disable
    set priority 150
    set monitor "port1" "port2"
end
```




```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
HzpTLoO69D1nrZ9QZbKFDbuUmCefmDRbZqTvgSseAgh2zU8JOHq1bthC750a+p5S/rkfRSROmhVMqOKRbBZkir3C8VWW9l+h7jl7Xp4N5sdA1CmDSUzX=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set override disable
    set priority 120
    set monitor "port1" "port2"
end
```

4. 双方的 HA 配置完毕之后将 ha1 和 ha2 线相互直连，将会进行 HA 的选举，此时优先级 150 高的 FGT 将成为主防火墙，优先级低的 120 的 FGT 将成为备防火墙。然后备防火墙的配置将会和主防火墙进行同步，备防火墙所有配置和主防火墙一致，此时备防火墙也将不再可以 WEB/SSH 登录，仅仅可以 Console 登录。后续所有的操作和配置都在主防火墙上的 GUI、SSH 或 Console 完成。

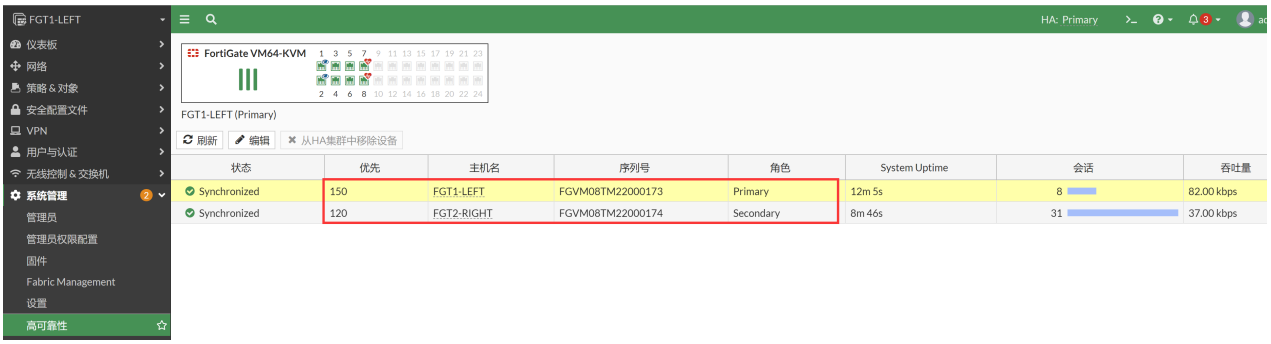
5. 选举结果如下，FGT1-LEFT 成为主设备。

```
FGT1-LEFT # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:22783 b:12890083
        traffic.total = s:0 p:22793 b:12890691
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=0 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM22000173:      Primary, serialno_prio=1, usr_priority=150, hostname=FGT1-LEFT
FGVM08TM22000174:    Secondary, serialno_prio=0, usr_priority=120, hostname=FGT2-RIGHT

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0:
FGVM08TM22000173:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM22000174:    Secondary, ha_prio/o_ha_prio=1/1
```

6. 这三个命令都可以看到 HA 的选举结果和 HA 状态。

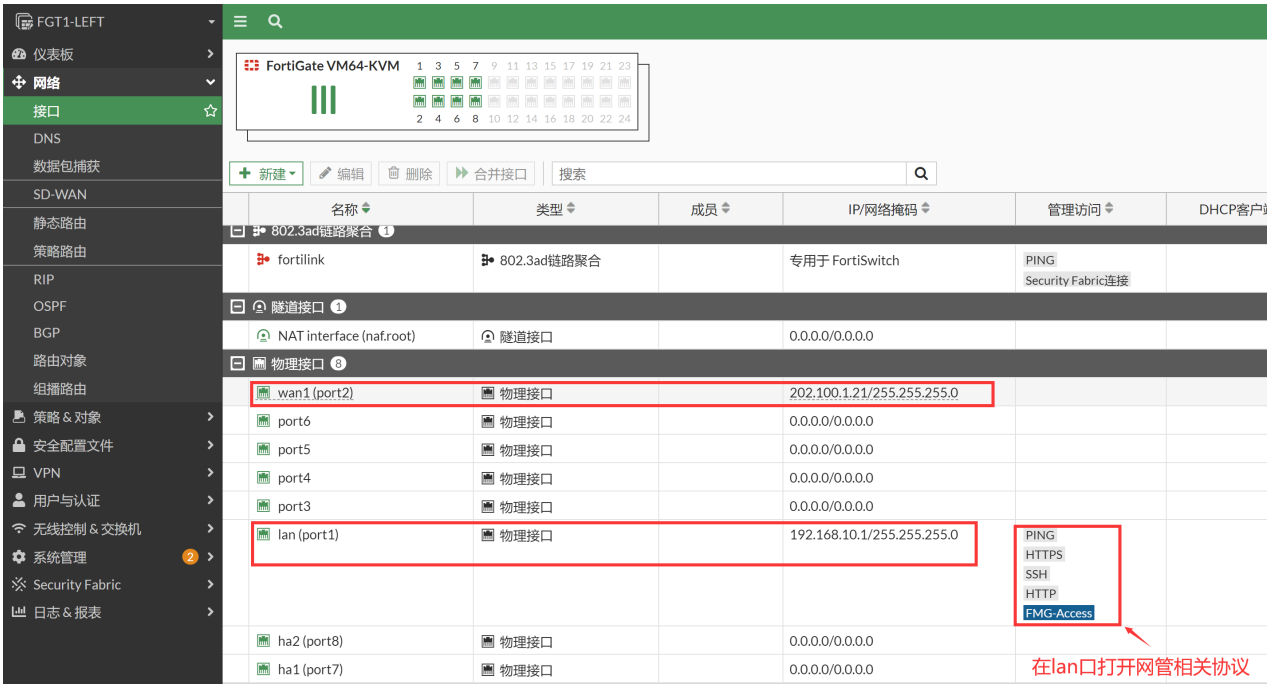
```
# diagnose sys ha status
# get system status
# get system ha status
```

7. GUI 查看 HA 的选举结果和状态。



8. 配置基础 LAN/WAN 上网服务。此步骤为正常上网的基础配置，大概的步骤是配置接口 IP、配置默认路由、配置策略即可实现防火墙的简单上网配置。（所有配置都在主防火墙上完成，备防火墙此时不可以登陆）

9. 配置接口 IP。



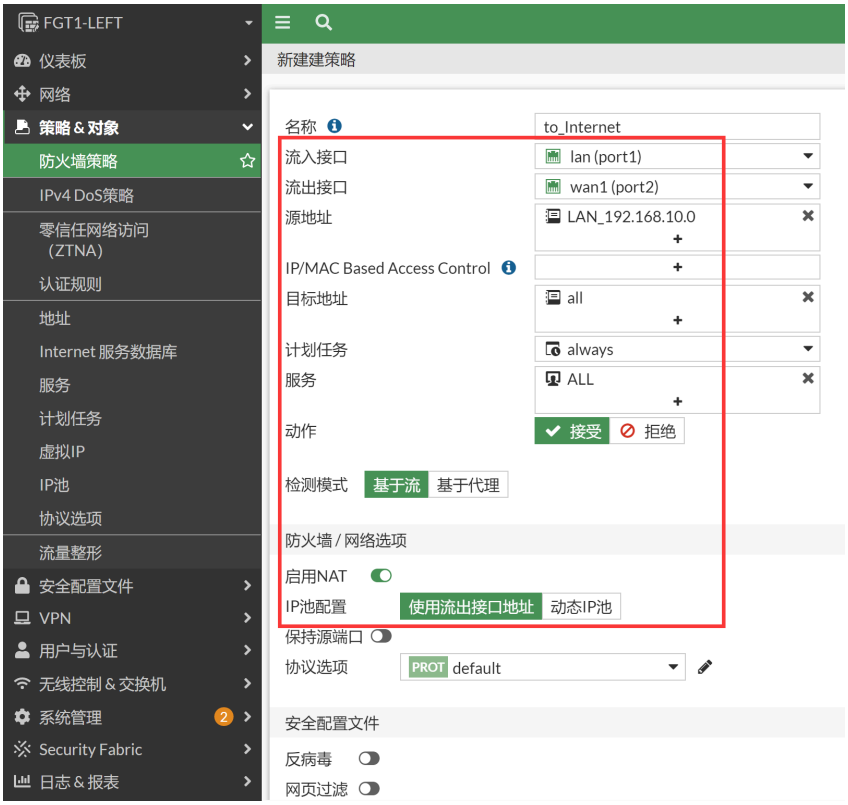

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set ip 202.100.1.21 255.255.255.0
        set type physical
        set alias "wan1"
        set snmp-index 2
    next
end
```

.0. 配置默认路由。




```
config router static
    edit 1
        set gateway 202.100.1.192
        set device "port2"
    next
end
```

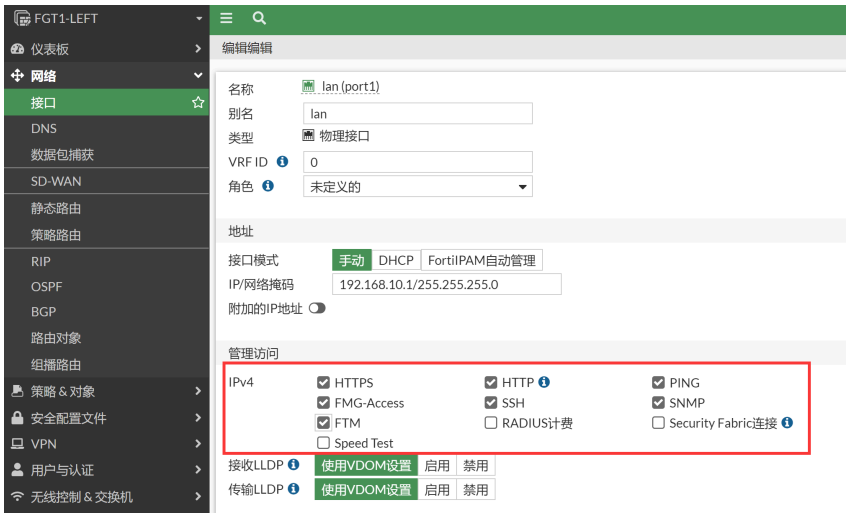
1. 配置上网策略。



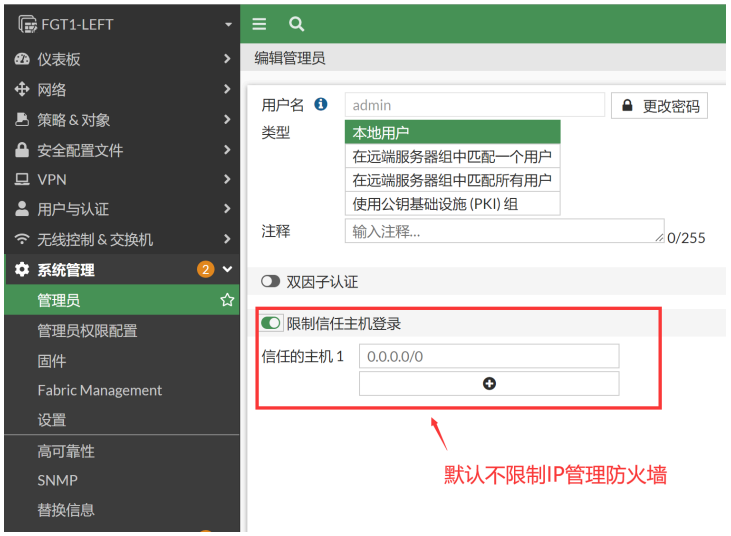

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

2. 配置 LAN 口网管相关参数，关于网管目前可以通过 port1（lan）的 192.168.10.1 进行管理，比较关键的网管因素是：

- port1 接口下开启了网管的协议，比如 https、http、ssh、snmp 等。



- 管理员可信任主机如果配置了的话，需要包含发起管理请求的主机 IP 或网段。



- 客户端 PC 到 FGT 的来回路由可达，可以通过 ping 和 tracert 测试。
- 上面 1、2 点确认没有问题之后，还是遇到不通的情况，可以通过 sniffer 抓包和 debug flow 确认问题所在。

```
FGT # diagnose sniffer packet any "host 192.168.10.1" 4 0 l
```

- 如果跨接口管理防火墙，比如处于 port1 的 PC 要通过防火墙 port2 接口的 IP 管理防火墙，则需要主要有 port1 到 port2 的放通策略，否则数据也是不通的。
- 出现网管不了的情况，99% 就是以上的原因了。

3. 配置 HA 独立管理口，**要注意如果作为独立管理的接口有关联的配置，则独立管理口不能选择该接口，比如某接口配置了 DHCP 服务器，则需要把 DHCP 相关的功能先去掉，然后才可以将该接口指定为独立管理口**。



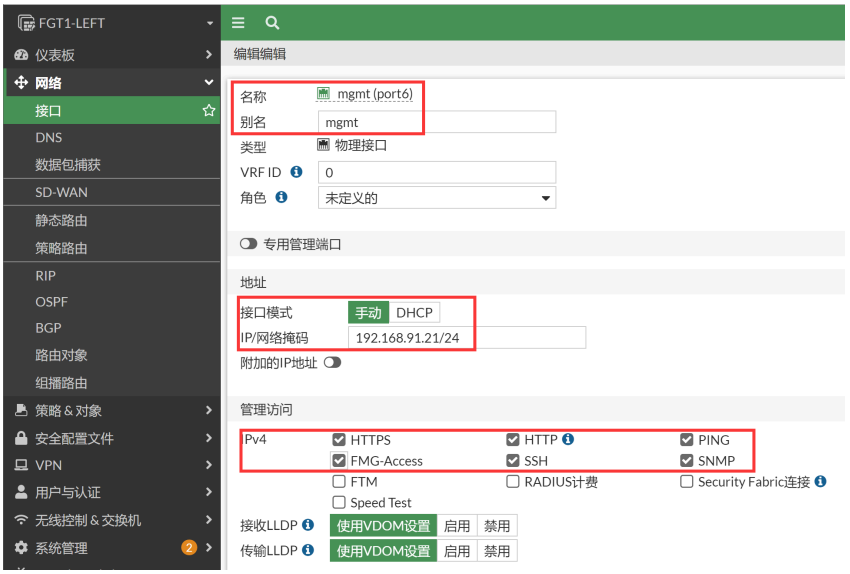
4. 在主设备和备设备上将 port6（mgmt）接口加入独立管理口，并配置独立管理口的网关。




```
config system ha
...
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port6"     //此命令会同步到备机
            set gateway 192.168.91.254     //独立管理口的网关IP地址配置不会同步到备防火墙，因为二者可以拥有不同的IP网段和网关地址，不同步可以满足更多的需求，也更加灵活
        next
    end
...
end
```

5. 此时如果本身就是通过 mgmt 接口管理的防火墙，则可能管理会中断，必须使用 console 口进行 mgmt 接口 IP 的配置。如果使用的业务接口 lan（port1）管理的防火墙，则直接修改 mgmt 接口 IP 地址即可。

6. 配置 mgmt 接口（port6）的 IP，并开启相关的网管服务。




```
config system interface
    edit "port6"
        set ip 192.168.91.21 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type physical
        set alias "mgmt"
        set snmp-index 6
    next
end
```

7. 主机的 HA 独立管理口配置完毕，开始配置备机的独立管理口，此时备机是无法 WEB/SSH 登陆的需要通过主机的 execute ha manage 命令（或者 Console 线）连到备机的命令行界面。

```
FGT1-LEFT # exe ha manage ?
<id>    please input peer box index.
<0>     Subsidiary unit FGVM08TM22000174

FGT1-LEFT # exe ha manage 0 admin
admin@169.254.0.1's password:
FGT2-RIGHT #
```

8. 配置备防火墙的 mgmt 管理口 IP 地址（开启 HA 独立管理口功能之后，mgmt 的配置不会再同步，需要手工配置 mgmt 的 IP 地址）。

```
FGT2-RIGHT # config system interface
FGT2-RIGHT (interface) # edit port6
FGT2-RIGHT (port6) # set ip 192.168.91.22/24
FGT2-RIGHT (port6) # set allowaccess ping https ssh snmp http fgfm
FGT2-RIGHT (port6) # end
```

9. 配置 HA 独立管理口的网关（此配置不会从主设备同步，需要单独配置）。

```
FGT2-RIGHT # config system ha
FGT2-RIGHT (ha) # config ha-mgmt-interfaces
FGT2-RIGHT (ha-mgmt-interfaces) # edit 1
FGT2-RIGHT (1) # set gateway 192.168.91.254
FGT2-RIGHT (1) # end
FGT2-RIGHT (ha) # show
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
L2tPmaCKq4nyH9Y6/+wKJQTXF2ROKYww0zCKqylOTEiACqph0eB+Ji5FDv5OimBxc4XvxZD8cU6bsWbRP8l/TcHRgpOvoPB/9m/Qu23voGOSxhLxlgiz=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port6"
            set gateway 192.168.91.254
```

```
        next
    end
    set override disable
    set priority 120
    set monitor "port1" "port2"
end
FGT2-RIGHT (ha) # end
```

10. HA 独立管理口的配置完毕，此时管理员可以通过 192.168.91.21 管理主防火墙，192.168.91.22 管理备防火墙。两台防火墙拥有独立的管理 IP。

11. 如何测试独立管理口和独立管理口的网关是否可以通信？独立管理口实际可以理解为一个轻量级的 vdom，如果要测试此 vdom 里面的数据转发，需要先进入到此轻量级 vdom 里面然后再发起相应的 ping 测试，测试才是我们想要的结果。否则直接发起 ping 的话，就相当于在 root vdom（业务处理的 VDOM）发包，实际和独立管理口是隔离的是无关的，会引起误会/误解。

12. 测试独立管理口和独立管理口网关的测试，可以看到 vsys_hamgmt 有独立的路由表和管理路由。

> **提示**
>
> 在已开启多 VDOM 的环境下，需要先进入 root VDOM，再进入 vsys_hamgmt VDOM，才能执行后续 CLI 命令。
>
> ```bash
> config vdom
> edit root
> exe enter vsys_hamgmt
> ```

```bash
FGT1-LEFT # exe enter ?
<name>    vdom name
root
vsys_hamgmt

FGT1-LEFT # exe enter vsys_hamgmt    //进入轻量级独立管理vdom
current vdom=vsys_hamgmt:1

FGT1-LEFT # get router info routing-table all    //拥有独立的路由表
...
Routing table for VRF=0
C       192.168.91.0/24 is directly connected, port6

FGT1-LEFT # diagnose ip route list
...
tab=254 vf=1 scope=0 type=1 proto=17 prio=0 0.0.0.0/0.0.0.0/0->0.0.0.0/0 pref=0.0.0.0 gwy=192.168.91.254 dev=8(port6)
//独立管理的默认路由
...

FGT1-LEFT # exe ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=128 time=0.9 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=128 time=1.3 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=128 time=0.8 ms
64 bytes from 192.168.91.254: icmp_seq=3 ttl=128 time=0.5 ms
64 bytes from 192.168.91.254: icmp_seq=4 ttl=128 time=0.7 ms

--- 192.168.91.254 ping statistics ---
5 packets transmitted, 5 packets received, 4 duplicates, 0% packet loss
round-trip min/avg/max = 0.5/1.9/4.3 ms

FGT-LEFT # diagnose sniffer packet any "host 192.168.91.254" 4
interfaces=[any]
filters=[host 192.168.91.254]
3.235814 port6 out 192.168.91.21 -> 192.168.91.254: icmp: echo request    //匹配明细直连路由，ping数据从port6 out，可以成功ping通网关
3.235900 port6 in 192.168.91.254 -> 192.168.91.21: icmp: echo reply
4.245672 port6 out 192.168.91.21 -> 192.168.91.254: icmp: echo request
4.245747 port6 in 192.168.91.254 -> 192.168.91.21: icmp: echo reply
```

13. 退出独立管理口轻量级 vdom，进入 root vdom，查看路由表，并尝试访问独立管理口的网关。

```bash
FGT1-LEFT # exe enter root
current vdom=root:0

FGT1-LEFT # get router info routing-table all    //业务的独立路由表，和独立管理的路由表完全独立，相当于路由的轻量级隔离
...
```

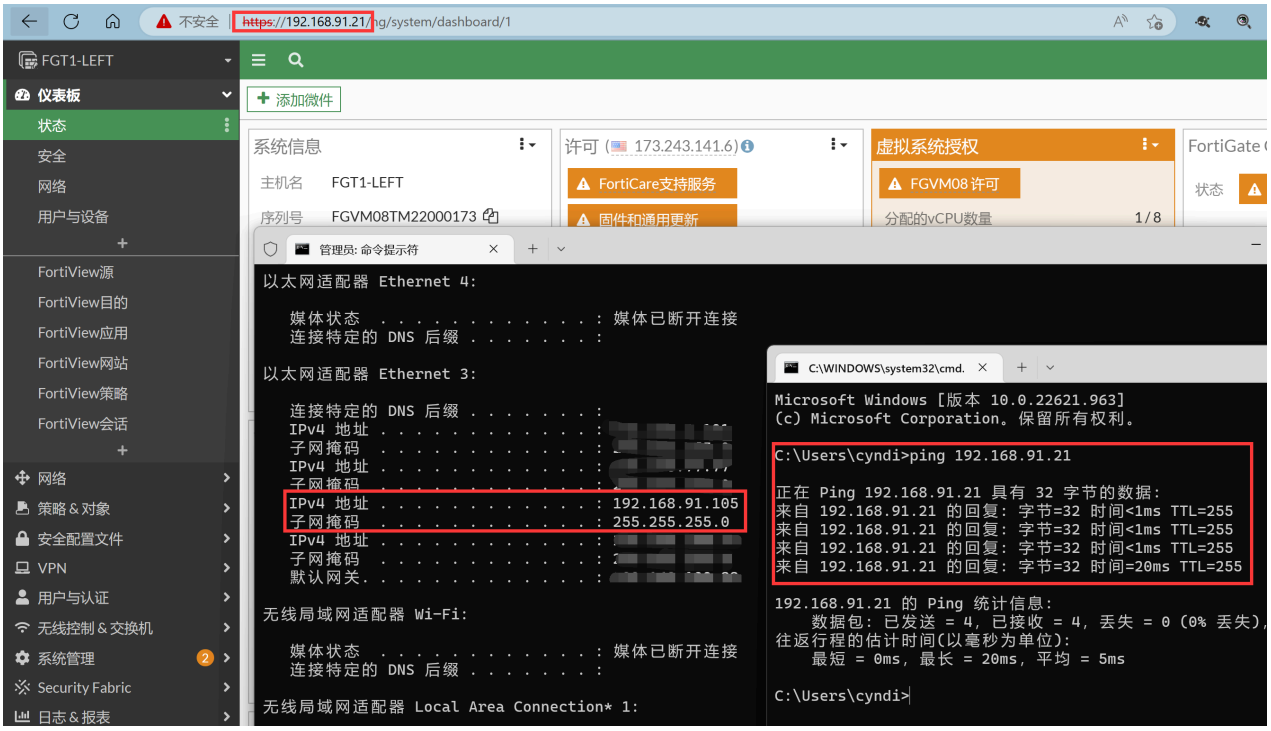
```
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.100.1.192, port2, [1/0]
C       192.168.10.0/24 is directly connected, port1
C       202.100.1.0/24 is directly connected, port2

FGT101E_Master_379 # execute ping 192.168.91.254  // 通过root ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes

--- 192.168.91.254 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss

FGT-LEFT # diagnose sniffer packet any "host 192.168.91.254 and icmp" 4
interfaces=[any]
filters=[host 192.168.91.254 and icmp]
12.785835 port2 out 202.100.1.21 → 192.168.91.254: icmp: echo request  // 匹配到默认路由，ping数据从port2 out，无法ping通独立管理的网关，结果是正确的，但这不是我们想要的结果。处于一个错误的状态，实际并不是不通，只是没有进入vsys_hamgmt而已
13.795618 port2 out 202.100.1.21 → 192.168.91.254: icmp: echo request
14.805621 port2 out 202.100.1.21 → 192.168.91.254: icmp: echo request
15.815615 port2 out 202.100.1.21 → 192.168.91.254: icmp: echo request
16.825621 port2 out 202.100.1.21 → 192.168.91.254: icmp: echo request
```

4. 测试备机的独立管理口网关 ping 测试。

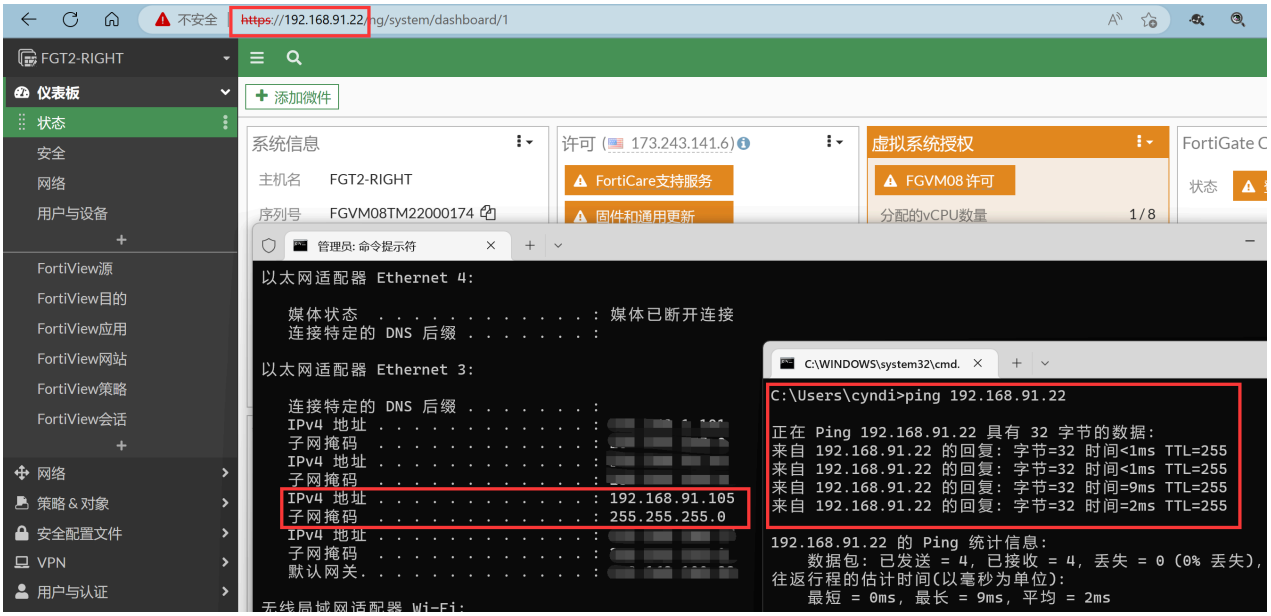
```
FGT2-RIGHT # exe enter vsys_hamgmt
current vdom=vsys_hamgmt:1

FGT2-RIGHT # execute ping 192.168.91.254    //发起ping测试
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=3 ttl=255 time=0.1 ms
64 bytes from 192.168.91.254: icmp_seq=4 ttl=255 time=0.1 ms
--- 192.168.91.254 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms

FGT2-RIGHT # execute enter root    //测试完毕，退出轻量级独立管理vdom
current vdom=root:0
```

HTTPS、Ping 管理

1. 通过独立管理 IP：192.168.91.21 管理主防火墙。



2. 通过独立管理 IP：192.168.91.22 管理备防火墙。



SYSLOG

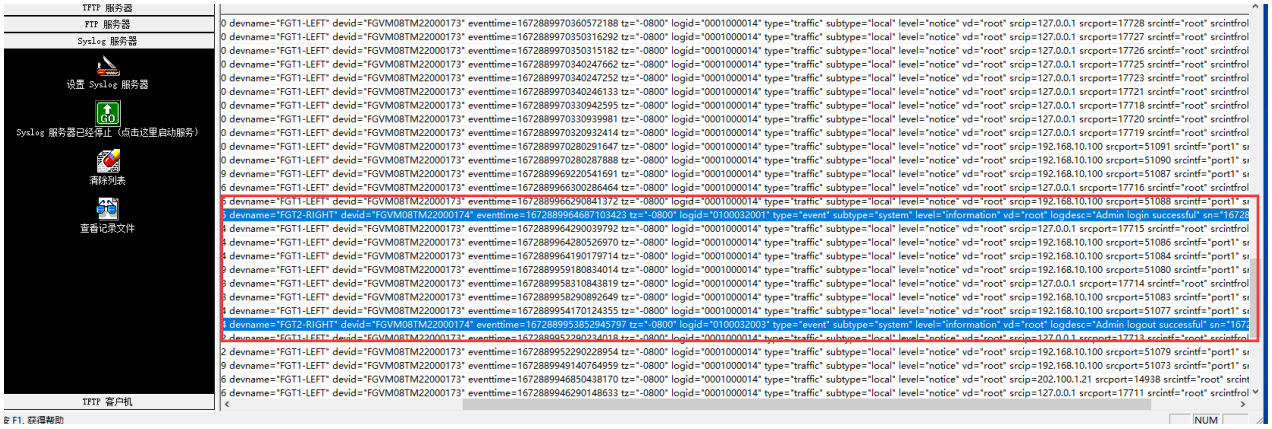
1. 确认通过 192.168.10.1 网管没问题的话，配置 SYSLOG 等本机发出的网管配置。




```
config log syslogd setting
    set status enable
    set server "192.168.10.100"
end
```

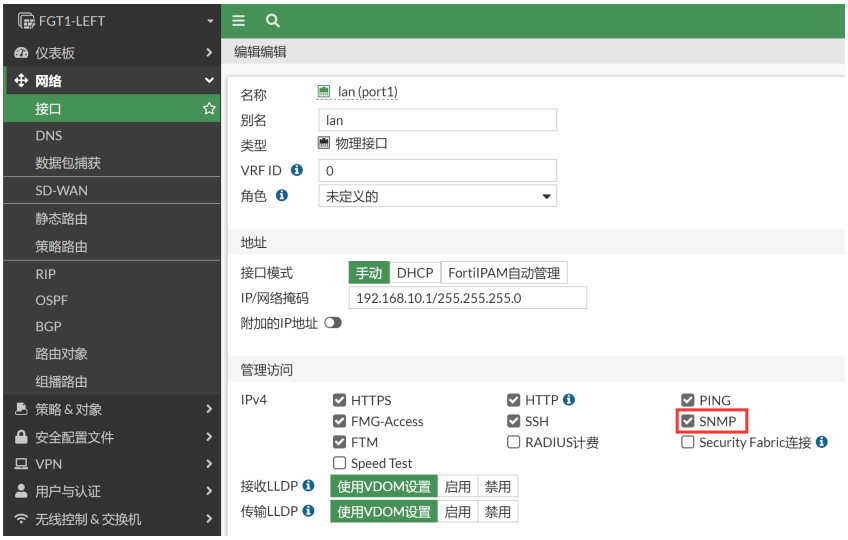
2. 主/备防火墙的日志都会通过业务接口 port1 192.168.10.1 将 syslog 日志发送出去，备防火墙的 syslog 日志会通过心跳线到主机，然后由主机代为转发。

```
FGT1-LEFT # diagnose sniffer packet any "port 514" 4
interfaces=[any]
filters=[port 514]
0.918154 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 635    //主防火墙的syslog日志
0.918225 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 649
5.492057 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 649
5.961902 port_ha in 169.254.0.2.514 → 169.254.0.1.514: udp 454    //备防火墙的console登陆日志
5.962018 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 418
5.968684 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 554
8.314617 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 730
8.507346 port_ha in 169.254.0.2.514 → 169.254.0.1.514: udp 473    //备防火墙的console退出日志
8.507479 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 437
8.964545 port1 out 192.168.10.1.22418 → 192.168.10.125.514: udp 724
9.074989 port1 out 192.168.10.1.15840 → 192.168.10.125.514: udp 597
```



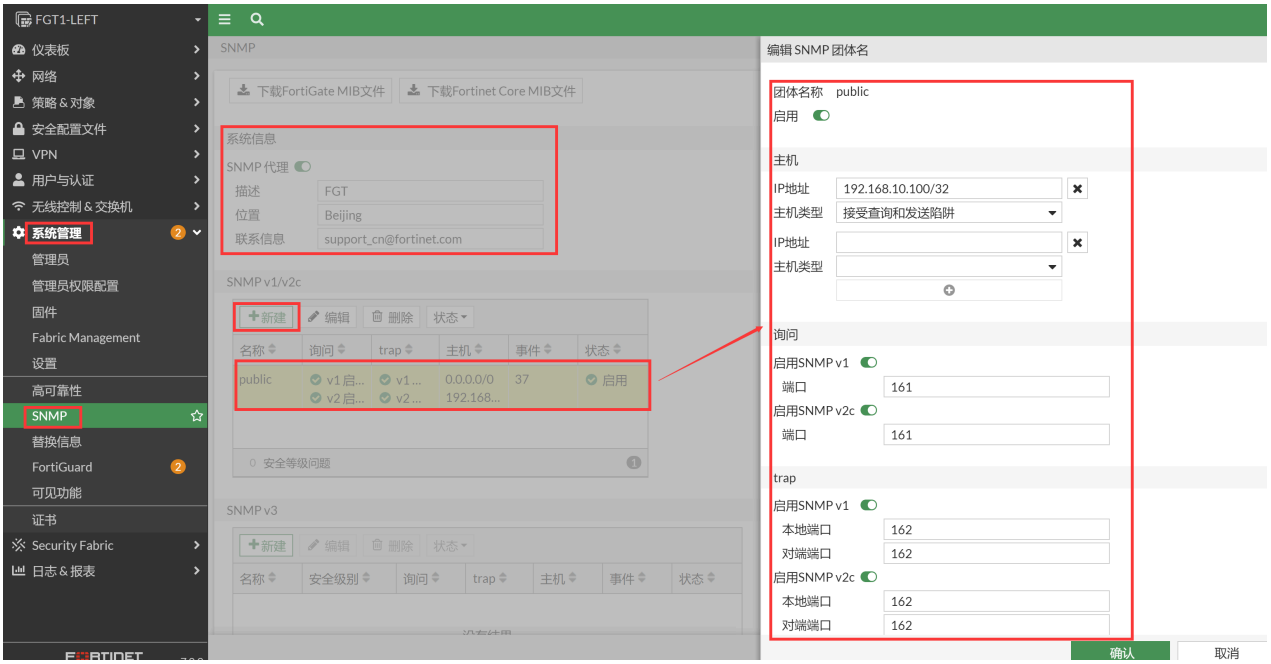
SNMP

1. lan 接口下开启 SNMP 协议。



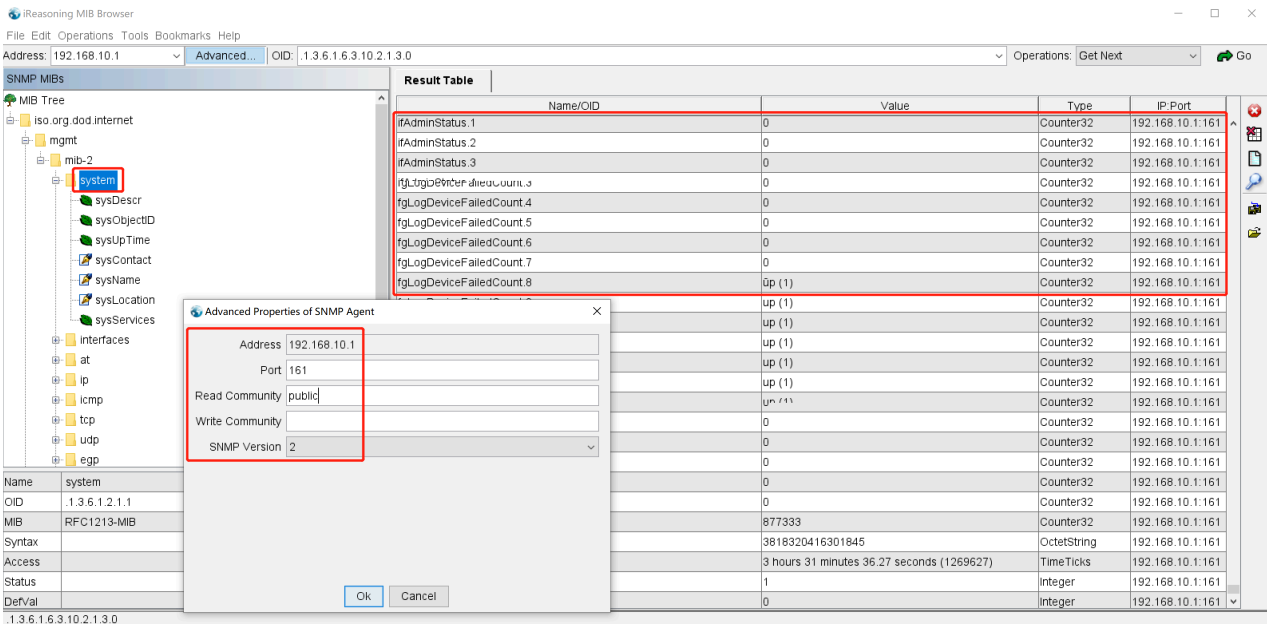

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
end
```

2. 开启 SNMP 功能。

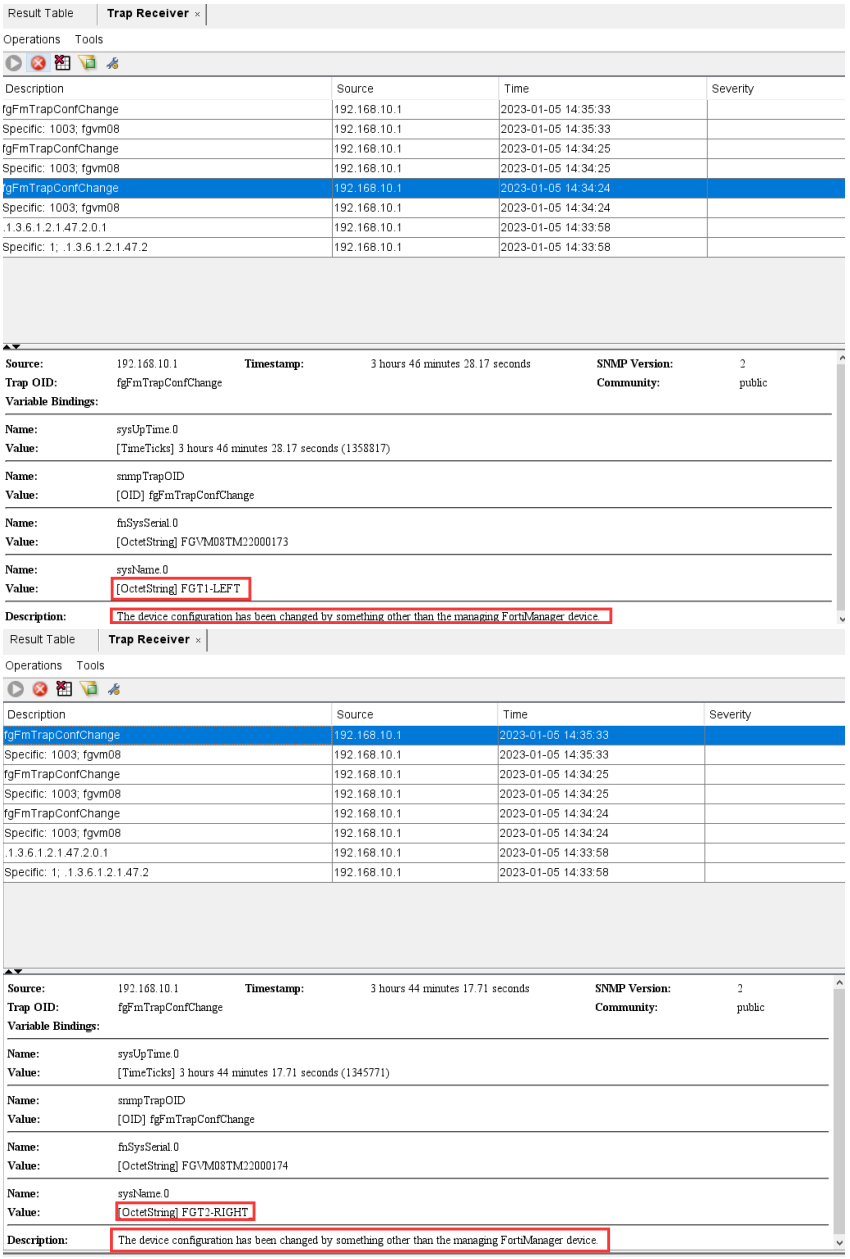



```
config system snmp sysinfo
    set status enable
    set description "FGT"
    set contact-info "support_cn@fortinet.com"
    set location "Beijing"
end

config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ip 192.168.10.100 255.255.255.255
            next
        end
    next
end
```

3. 在 SNMP 客户端对 FortiGate 的 lan 口进行 get/walk 操作，这里以 IReasoning MIB Browser 为例。

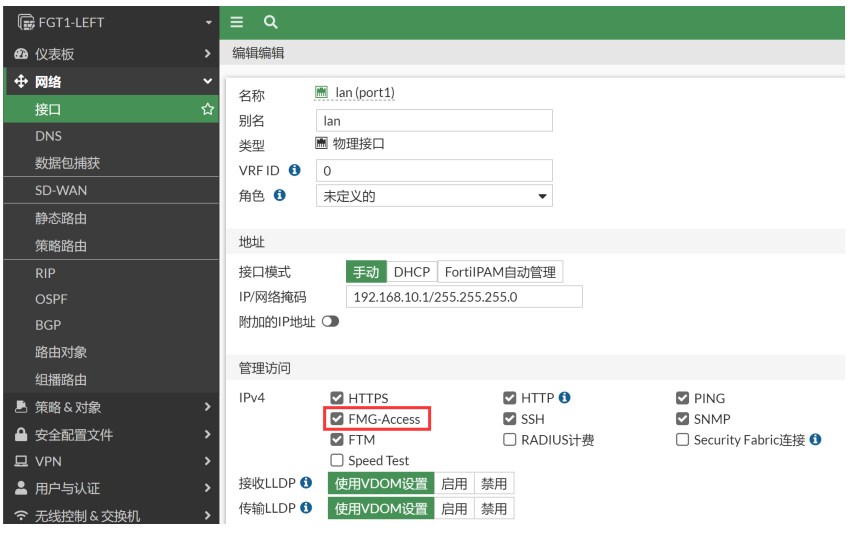


4. SNMP 客户端接收到的 FGT 发送的 trap 信息，可以看到主备设备均通过主设备进行发送。



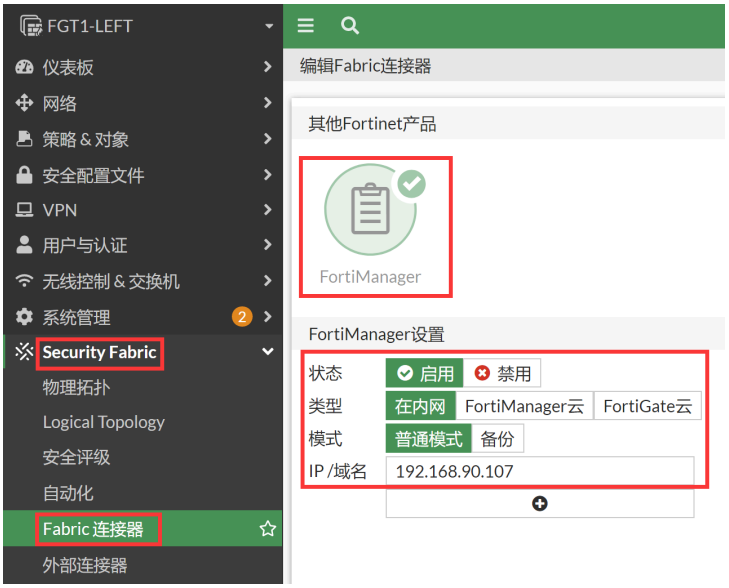
FortiManger 管理

1. FortiManager 通过 FGT 的 lan 口添加设备，在 FGT 的 lan 口下开启 FMG 管理协议。



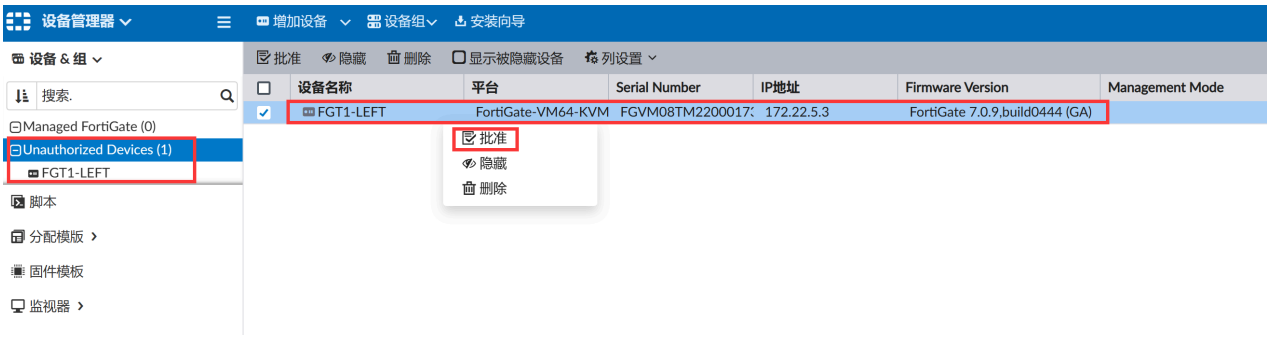
```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
end
```

2. 配置防火墙主动连接 FMG（如果 FMG 可以主动访问到 FGT，也可以通过 FMG 主动添加 FGT）。

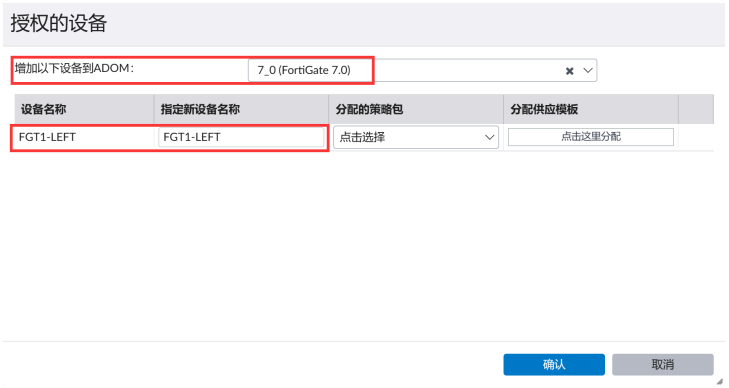



```
config system central-management
    set type fortimanager
    set fmg "192.168.90.107"
end
```

3. 如果是 FGT 主动去注册 FMG，则 FMG 会有未注册的设备提示。



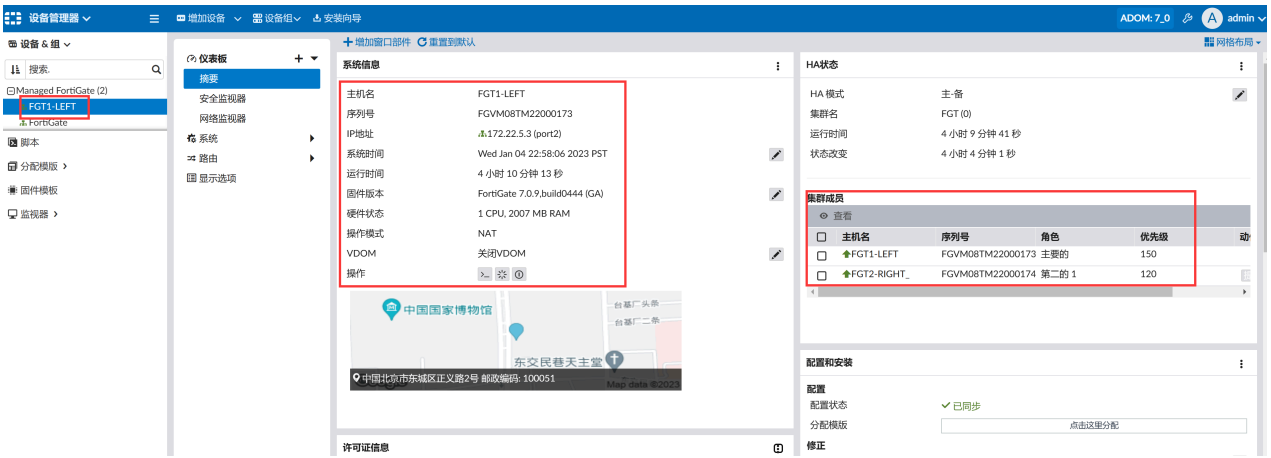
4. 只需要将设备添加到 FMG 的对应 ADOM 的设备管理中即可。



5. 授权完成。



6. 在 FMG 中可以对 FGT 进行管理。



HA-direct

1. 以上的发送 syslog 都是通过业务接口 port1 192.168.10.1 发送的，如果想要如 syslog、FortiAnalyzer、LDAP、Radius、TACACS+、sFlow、NetFlow、NTP 等报文通过 mgmt（port6）发送出去，则需要在 HA 里面开启一个 ha-direct 功能才可以实现。该命令在主机开启后，也会同步到备机。

> **重要**
>
> 开启 HA 配置下的“HA-direct”后，主机和备机的 syslog、FortiAnalyzer、LDAP、Radius、TACACS+、sFlow、NetFlow、NTP 等报文均会通过各自的独立管理口发出，而不是通过主机的业务口发出。

```bash
FGT1-LEFT # config sys ha

FGT1-LEFT (ha) # set ha-direct ?
enable     Enable using ha-mgmt interface for syslog, SNMP, remote authentication (RADIUS), FortiAnalyzer, FortiSandbox, sFlow, and Netfl.
disable    Disable using ha-mgmt interface for syslog, SNMP, remote authentication (RADIUS), FortiAnalyzer, FortiSandbox, sFlow, and Netf.

FGT1-LEFT (ha) # set ha-direct enable
FGT1-LEFT (ha) # end
When ha-direct is enabled, source ip may not work.
We recommend to unset all log-related, netflow and sflow source ip.
By selecting to continue, all source ip will be unset.
Do you want to continue? (y/n)y
```

2. 再次查看 syslog 的发送情况，syslog 这次就从 mgmt（port6）发送出去了。

```
FGT1-LEFT # diagnose sniffer packet any "port 514" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 514]
1.189223 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 553
1.189276 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 553
1.189278 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 553
2.189291 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 556
3.188617 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 613
4.194761 port6 out 192.168.91.21.3320 → 192.168.10.100.514: udp 488

FGT2-RIGHT # diagnose sniffer packet any "port 514" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 514]
7.137889 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 489
8.136846 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 554
8.136907 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 554
8.136909 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 554
8.136911 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 554
10.135873 port6 out 192.168.91.22.5532 → 192.168.10.100.514: udp 554
```

3. 如果是 snmp 的 trap 想从独立管理口发送（或通过独立管理口读取 OID），则需要在 SNMP 的 community 配置中，对 host 开启 ha-direct 功能，主备同步。

```
config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ip 192.168.10.100 255.255.255.255
                set ha-direct enable
            next
        end
    next
end
```

4. 再次查看 snmp trap 的发送情况，trap 这次就从 mgmt（port6）发送出去了。

```
FGT1-LEFT # diagnose sniffer packet any "port 162" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 162]
10.358519 port6 out 192.168.91.21.162 → 192.168.10.100.162: udp 106
10.358662 port6 out 192.168.91.21.162 → 192.168.10.100.162: udp 128

FGT2-RIGHT # diagnose sniffer packet any "port 162" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 162]
6.257505 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 43
6.257678 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 68
77.293372 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 43
77.293562 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 68
122.341966 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 107
122.342170 port6 out 192.168.91.22.162 → 192.168.10.100.162: udp 129
```

5. 关于 FMG 还是建议使用业务接口进行管理，不能使用独立管理口进行管理。对于 FMG 来说 HA-Cluster 就是一台设备，因此需要一个固定的 IP 地址来网管 FGT 会比较合适，可以有效的避免 HA 发生切换的时候，FGT 和 FMG 的 tunnel 中断的情况。

6. DNS 和 FortiGuard 更新流量默认还是会通过系统管理 VDOM（默认为 root，该默认配置启动即存在，与是否启用 HA、ha-direct、独立管理口无关）进行解析和更新的。

```bash
config system global
    set management-vdom root
end
```

7. 要彻底的完完全全的实现 HA-Cluster 的独立管理（既包含如 Syslog 的管理流量，又包含 FortiGuard 更新流量）同时与业务完全隔离，则需要使用独立管理 VDOM，请参考 HA 双机热备 → HA-Cluster 的网管 → HA 集群带外管理 → 独立管理 VDOM ↗ 章节。

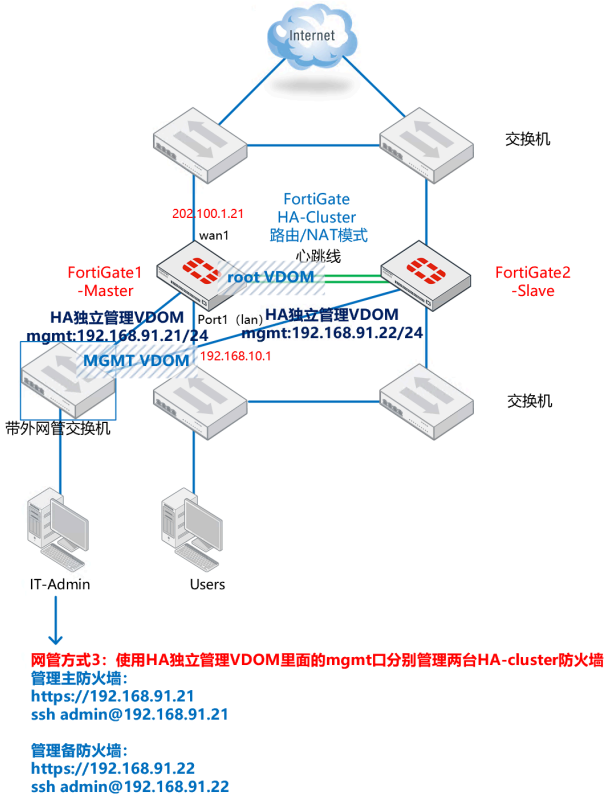
独立管理 VDOM

网络需求

1. 在 HA 集群中，所有集群成员的配置全部相同，通过 IP 只能管理到主设备，无法通过 IP 对每个 slave 设备进行单独的管理。同时为了业务的安全，需要将管理网络与业务网络独立开。包括独立管理，包括 syslog 日志、SNMP、Radius、TACACS + 等。
2. DNS 和 FortiGuard 更新等流量默认还是会通过系统管理 VDOM（默认为 root，该默认配置与是否启用 HA、ha-direct、独立管理口无关）进行解析和更新的。
3. 如果需要让 DNS、FortiGuard 更新流量也需要通过管理口进行处理，仅使用 HA 独立管理口是无法满足的，这就需要使用到独立管理 VDOM。

> **重要**
>
> 该功能主要用于单独的 VDOM 进行 FortiGuard 更新、DNS 查询以及日志/认证/管理业务，HA 成员间独立管理 VDOM 内的配置是不同步的，如果 FortiGate 被 FortiManager 管理，请勿开启此功能，否则 HA 切换后可能导致 FortiManager 配置与 FortiGate 不一致。

网络拓扑



配置要点

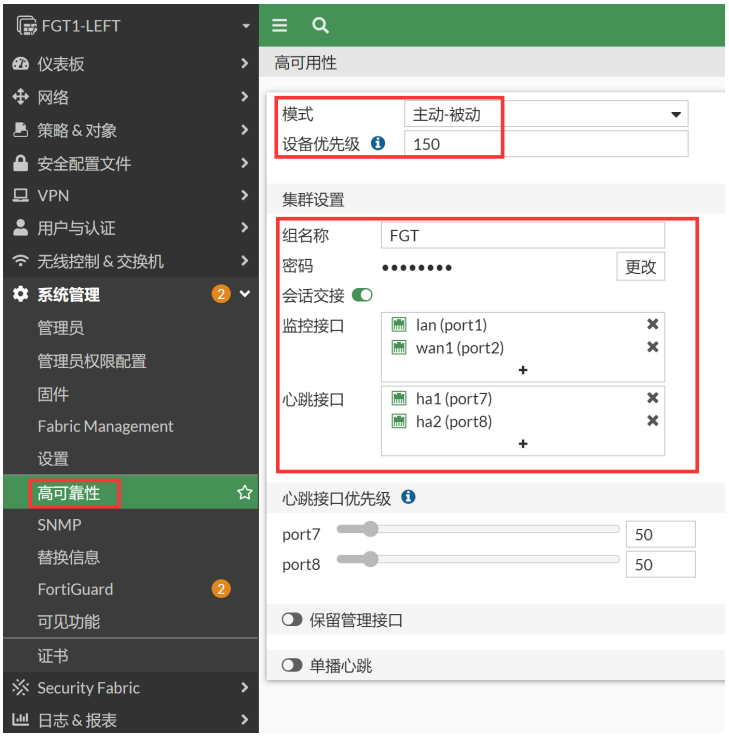
- 配置 HA 基础配置
- 配置基础 LAN/WAN 上网服务
- 配置 HA 独立 VDOM
- 配置防火墙的 SYSLOG、SNMP 和 FMG

配置步骤和结果验证

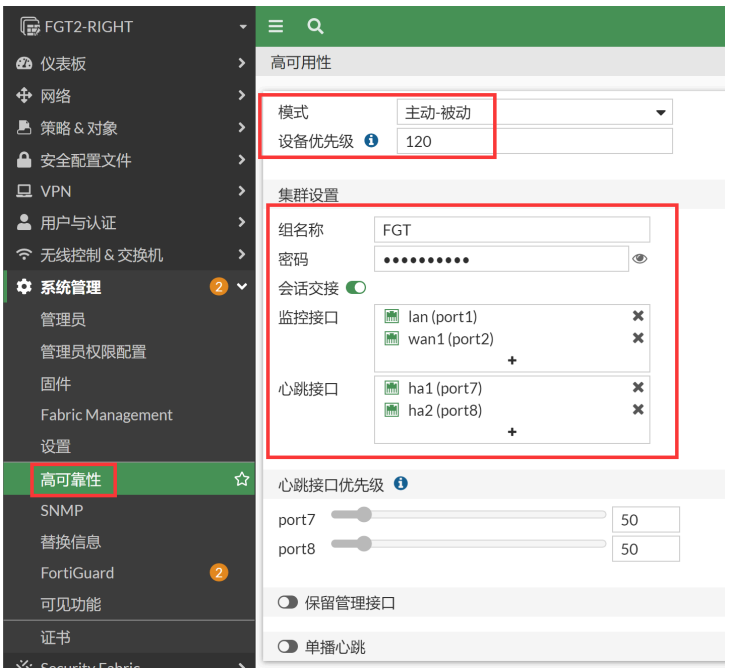
1. 配置 HA 基础配置，初始化配置防火墙的时候可以通过 mgmt 口登陆到防火墙上，一般 mgmt 口的管理 IP 是 192.168.1.99，第一步可以通过此 IP 登陆防火墙（参考系统管理 → 管理方式 → Web 管理 ↗章节），或者通过 console 口管理防火墙（参考系统管理 → 管理方式 → Console 管理↗章节）。具体型号初始化如果网管和接口的网管 IP 信息等可参考 https://docs.fortinet.com/product/fortigate/hardware ↗ 或 https://docs.fortinet.com/document/fortigate/hardware/fortigate-quickstart-guide-high-end?model ↗ = all 。

2. 按照以上方式登陆第一次开箱的防火墙。先登陆主防火墙，并配置进行 HA 的基础配置，然后登陆到备防火墙，配置备防火墙的 HA 基础配置。

3. 分别配置主防火墙和备防火墙的 HA，主防火墙优先级调整为 150，备防火墙的优先级保持默认的 120，配置 HA-cluster 组名字和密码，监控业务接口 wan1 和 port1，使用 HA1 和 HA2 接口将二者互联。




```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
XU4Yv532eMH20TXkbxDMQWRNs+ChZrIVXcotCFVP2C86QBl2zhznTNU+0PDKw5Vlvtm6B+NCZ+f+1EiT3MsUteDHW82Aten6q+Yu6zAT5NuXYYIKnVpG=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set override disable
    set priority 150
    set monitor "port1" "port2"
end
```



```
config system ha
    set group-name "FGT"
    set mode a-p
    set password ENC
HzpTLoO69D1nrZ9QZbKFDbuUmCefmDRbZqTvgSseAgh2zU8JOHq1bthC750a+p5S/rkfRSROmhVMqOKRbBZkir3C8VWW9l+h7jl7Xp4N5sdA1CmDSUzX=
    set hbdev "port7" 50 "port8" 50
    set session-pickup enable
    set override disable
    set priority 120
    set monitor "port1" "port2"
end
```

4. 双方的 HA 配置完毕之后将 ha1 和 ha2 线相互直连，将会进行 HA 的选举，此时优先级 150 高的 FGT 将成为主防火墙，优先级低的 120 的 FGT 将成为备防火墙。然后备防火墙的配置将会和主防火墙进行同步，备防火墙所有配置和主防火墙一致，此时备防火墙也将不再可以 WEB/SSH 登录，仅仅可以 Console 登录。后续所有的操作和配置都在主防火墙上的 GUI、SSH 或 Console 完成。

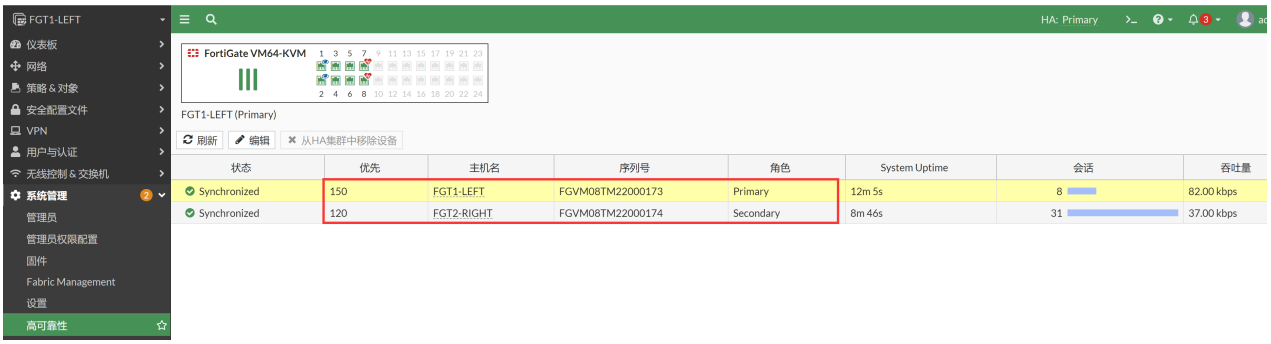
5. 选举结果如下，FGT1-LEFT 成为主设备。

```
FGT1-LEFT # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:22783 b:12890083
        traffic.total = s:0 p:22793 b:12890691
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=0 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM22000173:      Primary, serialno_prio=1, usr_priority=150, hostname=FGT1-LEFT
FGVM08TM22000174:    Secondary, serialno_prio=0, usr_priority=120, hostname=FGT2-RIGHT

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0:
FGVM08TM22000173:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM22000174:    Secondary, ha_prio/o_ha_prio=1/1
```

6. 这三个命令都可以看到 HA 的选举结果和 HA 状态。

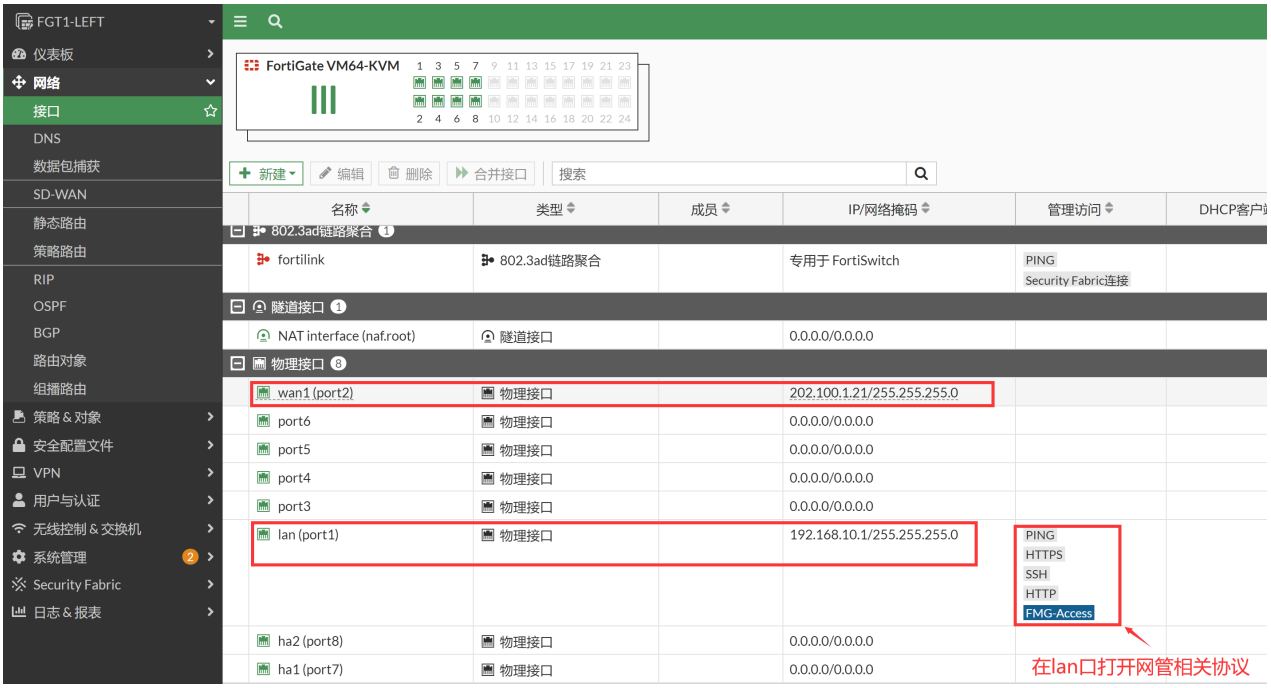
```
# diagnose sys ha status
# get system status
# get system ha status
```

7. GUI 查看 HA 的选举结果和状态。



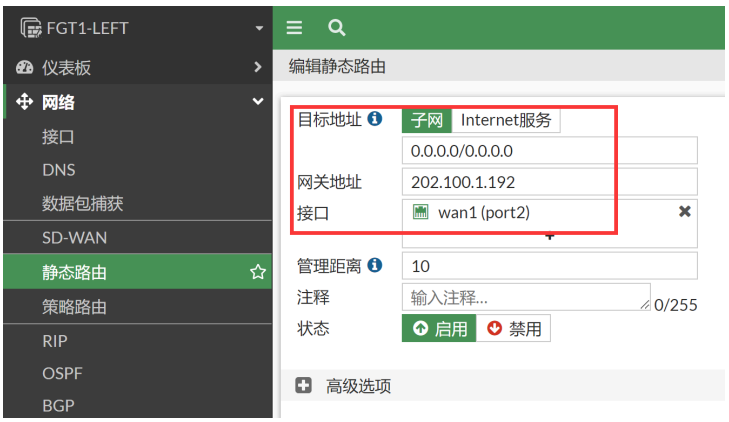
8. 配置基础 LAN/WAN 上网服务。此步骤为正常上网的基础配置，大概的步骤是配置接口 IP、配置默认路由、配置策略即可实现防火墙的简单上网配置。（所有配置都在主防火墙上完成，备防火墙此时不可以登陆）

9. 配置接口 IP。



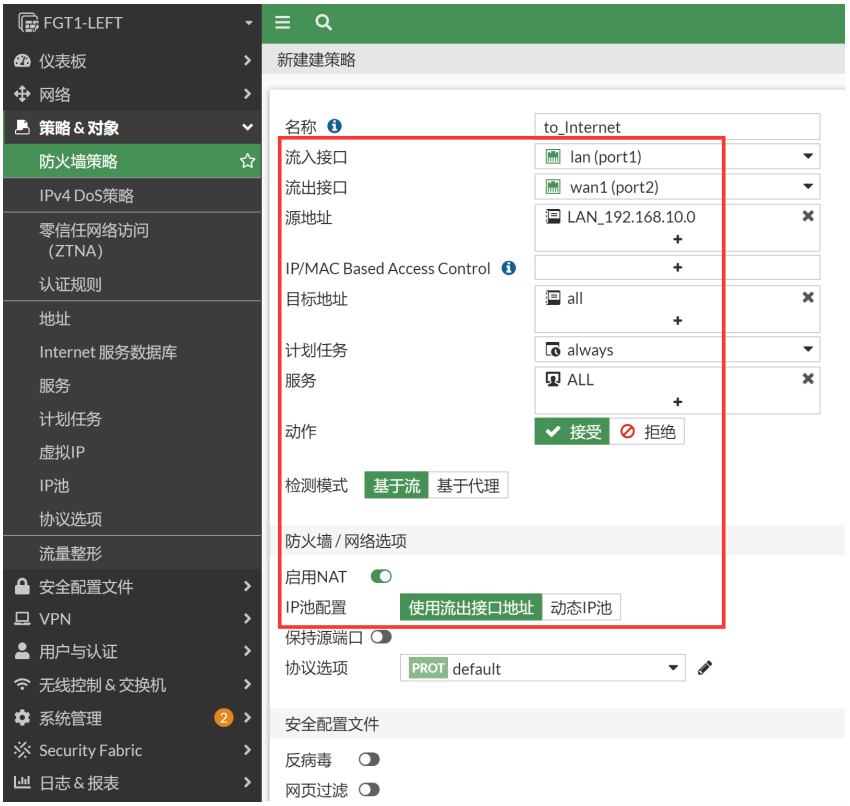

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
    edit "port2"
        set vdom "root"
        set ip 202.100.1.21 255.255.255.0
        set type physical
        set alias "wan1"
        set snmp-index 2
    next
end
```

.0. 配置默认路由。




```
config router static
    edit 1
        set gateway 202.100.1.192
        set device "port2"
    next
end
```

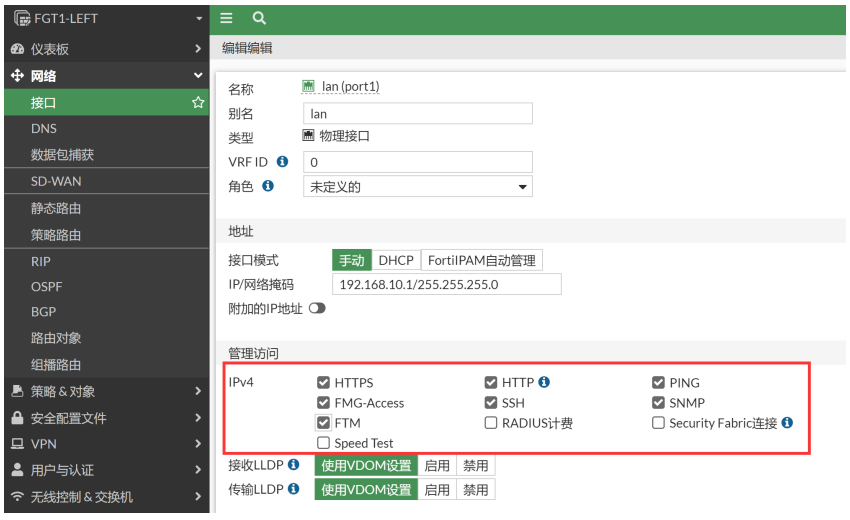
.1. 配置上网策略。



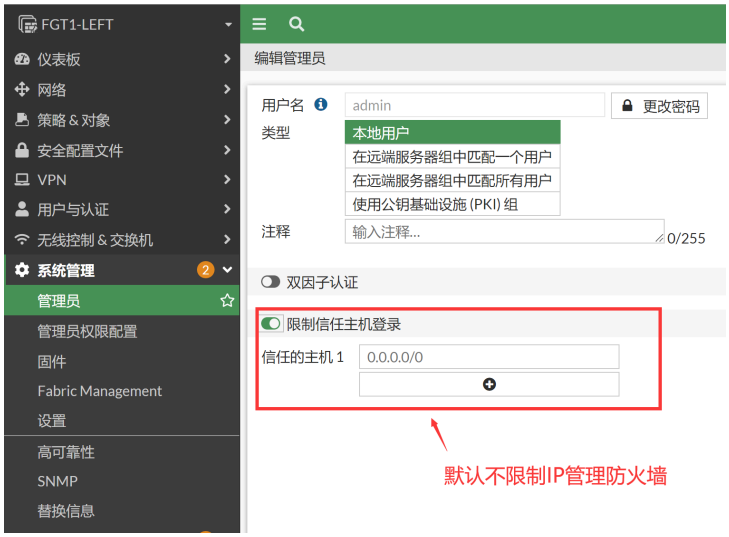

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "LAN_192.168.10.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

2. 配置 LAN 口网管相关参数，关于网管目前可以通过 port1（lan）的 192.168.10.1 进行管理，比较关键的网管因素是：

   - port1 接口下开启了网管的协议，比如 https、http、ssh、snmp 等。



   - 管理员可信任主机如果配置了的话，需要包含发起管理请求的主机 IP 或网段。



- 客户端 PC 到 FGT 的来回路由可达，可以通过 ping 和 tracert 测试。
- 上面 1、2 点确认没有问题之后，还是遇到不通的情况，可以通过 sniffer 抓包和 debug flow 确认问题所在。

```
FGT # diagnose sniffer packet any "host 192.168.10.1" 4 0 l
```

- 如果跨接口管理防火墙，比如处于 port1 的 PC 要通过防火墙 port2 接口的 IP 管理防火墙，则需要主要有 port1 到 port2 的放通策略，否则数据也是不通的。
- 出现网管不了的情况，99% 就是以上的原因了。

3. 配置 HA 独立 VDOM，首先需要开启 VDOM 特性（主设备上配置，会自动同步到备机）。

```
FGT1-LEFT # config sys global
FGT1-LEFT (global) # set vdom-mode
FGT1-LEFT (global) # set vdom-mode multi-vdom
FGT1-LEFT (global) # end

You will be logged out for the operation to take effect.
Do you want to continue? (y/n)y
exit

FGT1-LEFT login: admin
Password:
Welcome!
```

4. 创建新的 MGMT 独立管理 VDOM（主设备上配置，会自动同步到备机）。






```
FGT1-LEFT # config vdom
FGT1-LEFT (vdom) # edit MGMT
FGT1-LEFT (MGMt) # end
```

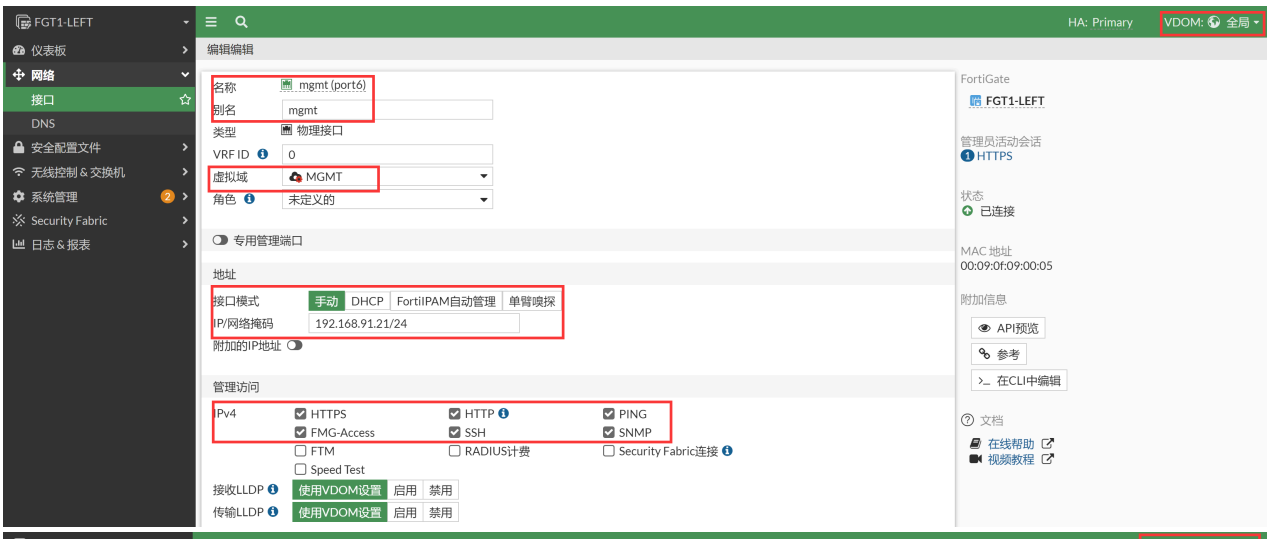
5. 将 MGMT 独立管理 VDOM 修改为管理 VDOM（只能命令行修改），该配置主备会同步。

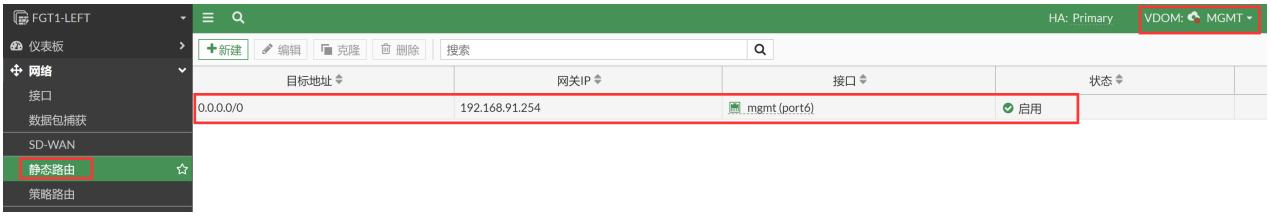
```
FGT1-LEFT # config global
FGT1-LEFT (global) # config system global
FGT1-LEFT (global) # set management-vdom MGMT     //设置MGMT为管理VDOM，也就是说fortiguard、dns、ntp ... 等等主动更新，都是从MGMT这个VDOM发起，是一个管理的VDOM
FGT1-LEFT (global) # end
FGT1-LEFT (global) # end
```

6. HA 里面开启独立管理 VDOM，该配置主备会同步。

```
FGT1-LEFT # config global
FGT1-LEFT (global) # config system ha
FGT1-LEFT (ha) # set standalone-mgmt-vdom enable
FGT1-LEFT (ha) # end
FGT1-LEFT (global) # end
```

7. 将 mgmt（port6）接口加入到 MGMT VDOM 中，并配置上管理 IP 和默认路由。






```
config global
config system interface
    edit "port6"
        set vdom "MGMT"
        set ip 192.168.91.21 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type physical
        set alias "mgmt"
        set snmp-index 6
    next
end
```

```
FGT1-LEFT # config vdom
FGT1-LEFT (vdom) # edit MGMT
current vf=MGMT:4
config router static
    edit 1
        set gateway 192.168.91.254
        set device "port6"
    next
end
```

8. 在主机的 MGMT VDOM 中 ping 网关 192.168.91.254。

```
FGT1-LEFT # config vdom
FGT1-LEFT (vdom) # edit MGMT
current vf=MGMT:4

FGT1-LEFT (MGMT) # exe ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=128 time=1.3 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=128 time=0.6 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=128 time=0.6 ms
64 bytes from 192.168.91.254: icmp_seq=3 ttl=128 time=0.5 ms
64 bytes from 192.168.91.254: icmp_seq=4 ttl=128 time=0.8 ms
--- 192.168.91.254 ping statistics ---
5 packets transmitted, 5 packets received, 4 duplicates, 0% packet loss
round-trip min/avg/max = 0.5/0.7/1.4 ms
```

9. 此时备机是无法 WEB/SSH 登陆的需要通过主机的命令（或者 Console 线）连到备机的命令行界面。

```
FGT1-LEFT # config global
FGT1-LEFT (global) # exe ha manage ?
<id>    please input peer box index.
<0>     Subsidiary unit FGVM08TM22000174

FGT1-LEFT (global) # exe ha manage 0 admin
Warning: Permanently added '169.254.0.1' (ED25519) to the list of known hosts.
admin@169.254.0.1's password:
FGT2-RIGHT #
```

10. 备防火墙那边会自动同步 VDOM 和 mgmt 加入到 MGMT 的配置（如果配置同步不正常，可以将备机重启，并等待配置同步完成，再操作），而独立管理 MGMT - VDOM 内部的配置是不会同步的，因此只需要修改备防火墙的 mgmt 的接口 IP 和路由即可。

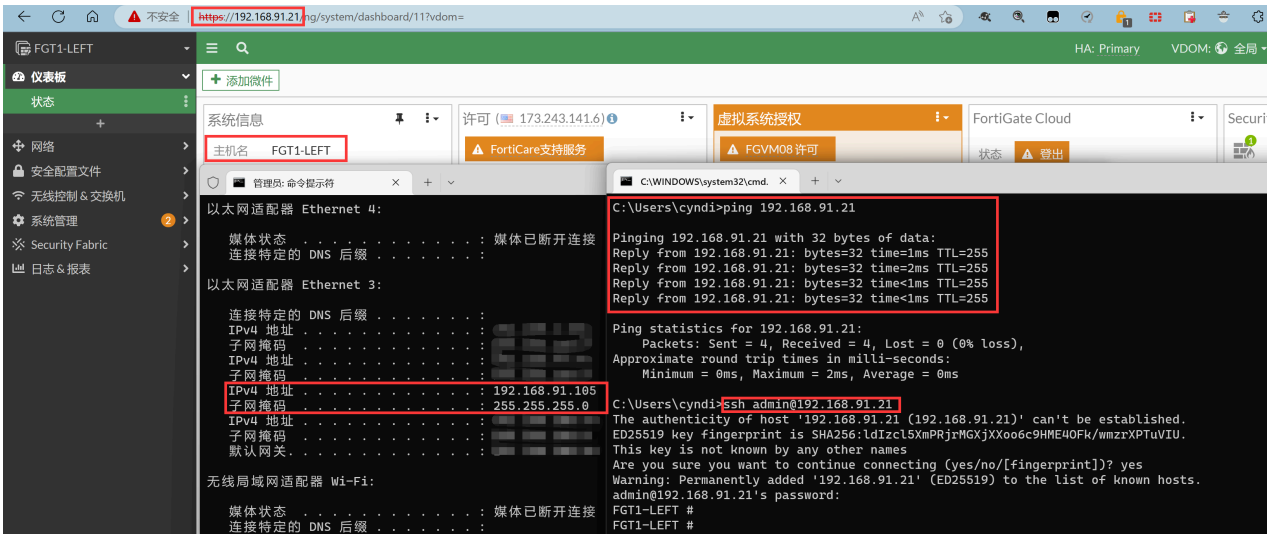
```
FGT2-RIGHT # config vdom
FGT2-RIGHT (vdom) # edit MGMT
current vf=MGMT:4

FGT2-RIGHT (MGMT) # config system interface
FGT2-RIGHT (interface) # edit port6
FGT2-RIGHT (port6) # set ip 192.168.91.22/24
FGT2-RIGHT (port6) # set allowaccess ping https ssh snmp http fgfm
FGT2-RIGHT (port6) # set alias "mgmt"
FGT2-RIGHT (port6) # end
FGT2-RIGHT (MGMT) # end
```

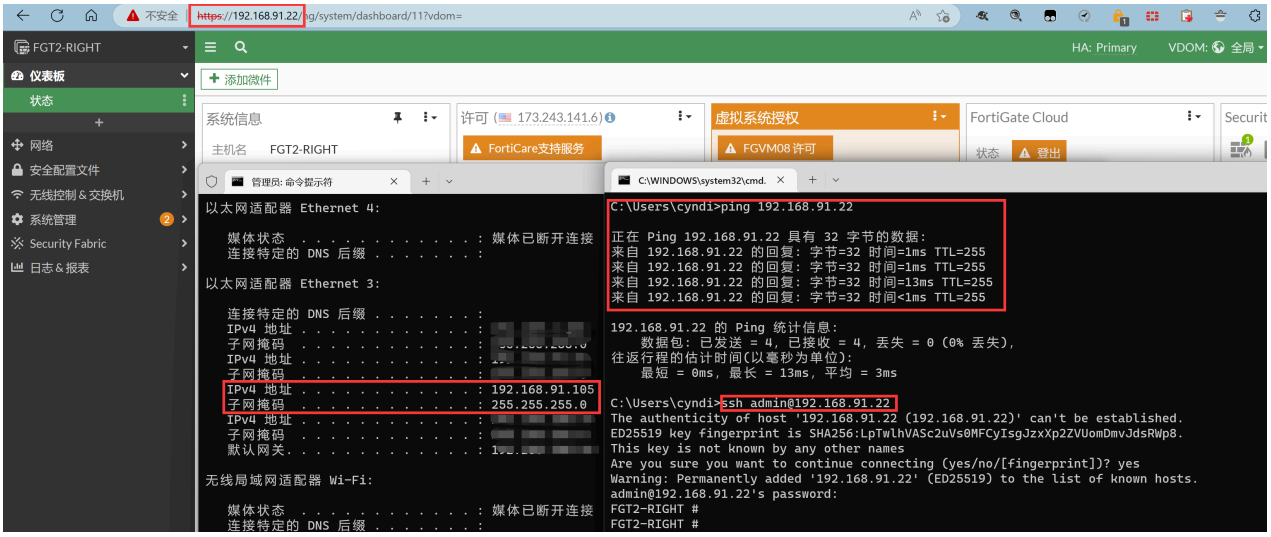
11. 备机使用 MGMT VDOM ping 网关 192.168.91.254。

```
FGT2-RIGHT # config vdom
FGT2-RIGHT (vdom) # edit MGMT
current vf=MGMT:4

FGT2-RIGHT (MGMT) # exe ping 192.168.91.254
PING 192.168.91.254 (192.168.91.254): 56 data bytes
64 bytes from 192.168.91.254: icmp_seq=0 ttl=128 time=2.2 ms
64 bytes from 192.168.91.254: icmp_seq=1 ttl=128 time=0.6 ms
64 bytes from 192.168.91.254: icmp_seq=2 ttl=128 time=0.7 ms
```

```
64 bytes from 192.168.91.254: icmp_seq=3 ttl=128 time=0.7 ms
64 bytes from 192.168.91.254: icmp_seq=4 ttl=128 time=0.5 ms
--- 192.168.91.254 ping statistics ---
5 packets transmitted, 5 packets received, 4 duplicates, 0% packet loss
round-trip min/avg/max = 0.5/1.0/2.3 ms
```

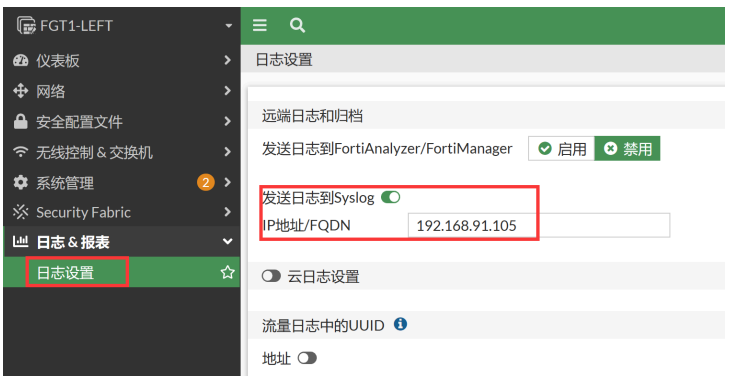
2. 备机也配置完毕，这个时候可以通过 GUI 登录到备机，查看备机的 GUI 配置。可以通过 192.168.91.21 和 192.168.91.22 分别管理到主防火墙和备防火墙了。





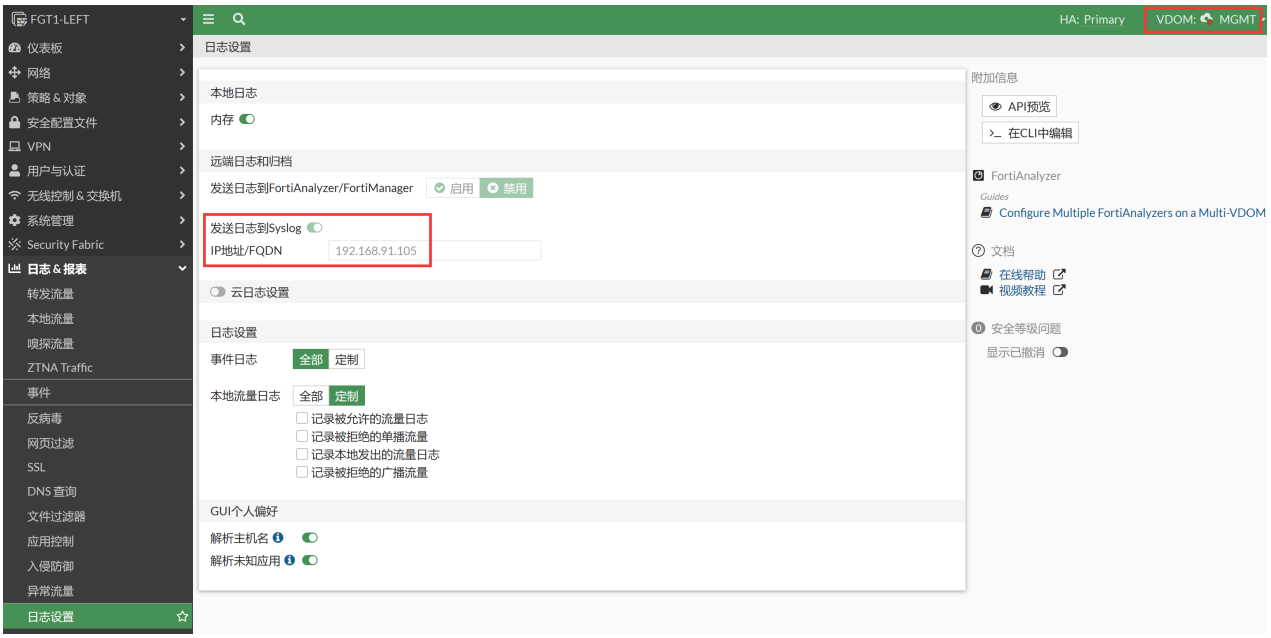
SYSLOG

1. 配置 SYSLOG 等本机发出的网管配置。



```
config global
config log syslogd setting
    set status enable
    set server "192.168.91.105"
end
```

2. 默认情况下 MGMT 和 root 会继承 global 的 syslog 日志配置。



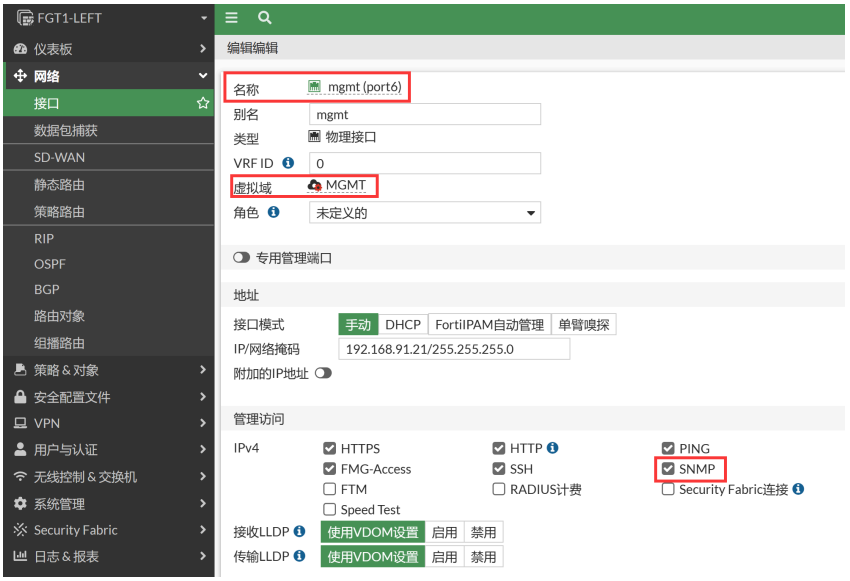
3. 主/备防火墙的日志都会通过独立管理 VDOM 的 mgmt（port6）192.168.91.21 和 192.168.91.22 将 syslog 日志发送出去。

```
FGT1-LEFT (MGMT) # diagnose sniffer packet any "port 514" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 514]
1.159125 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 600
1.159404 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 553
1.159522 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 553
3.158564 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 553
5.159318 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 608
6.158835 port6 out 192.168.91.21.19604 → 192.168.91.105.514: udp 600

FGT2-RIGHT (MGMT) # diagnose sniffer packet any "port 514" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 514]
4.513098 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
10.513817 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
11.512382 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
11.512444 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
11.512445 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
11.512447 port6 out 192.168.91.22.12931 → 192.168.91.105.514: udp 554
```



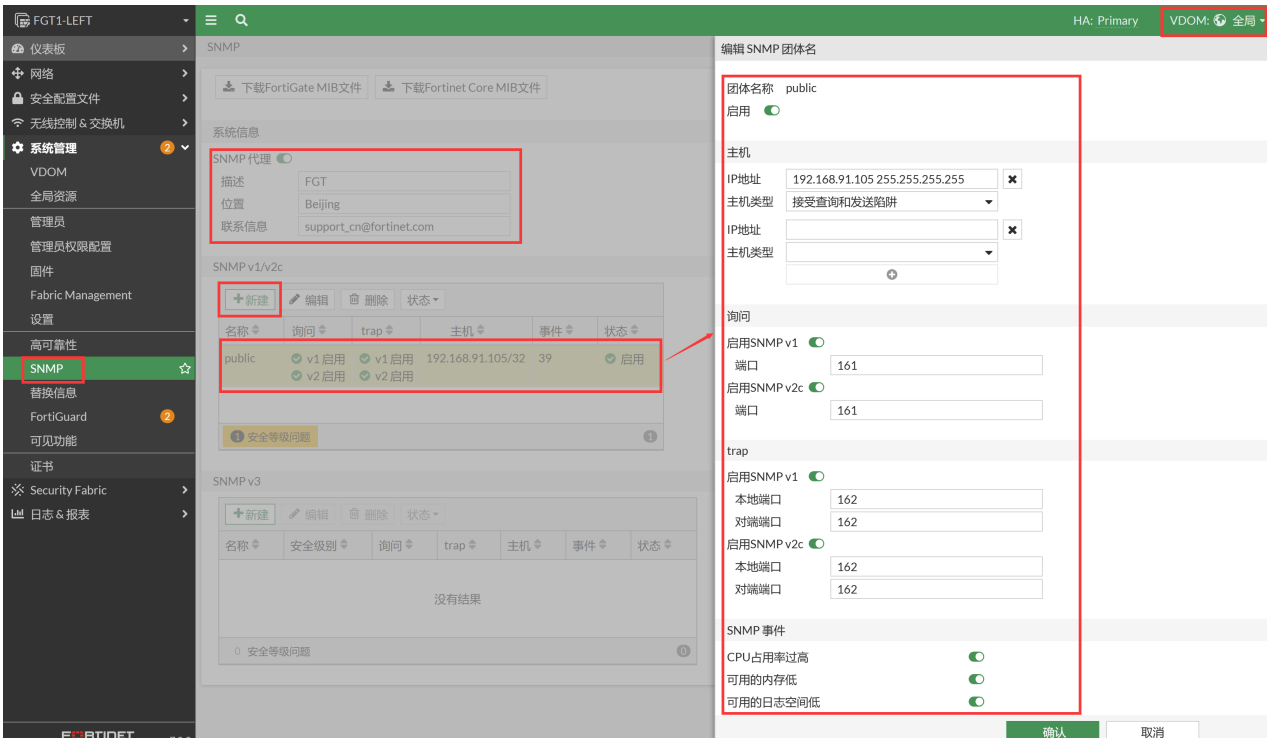
SNMP

1. VDOM - MGMT 的 mgmt（port6）接口下开启 SNMP 协议，主设备和备设备都要开，VDOM - MGMT 里面的配置不同步。




```
FGT1-LEFT:
config vdom
edit MGMT
config system interface
    edit "port6"
        set vdom "root"
        set ip 192.168.91.21 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "mgmt"
    next
end

FGT2-RIGHT:
config vdom
edit MGMT
config system interface
    edit "port6"
        set vdom "root"
        set ip 192.168.91.22 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "mgmt"
    next
end
```

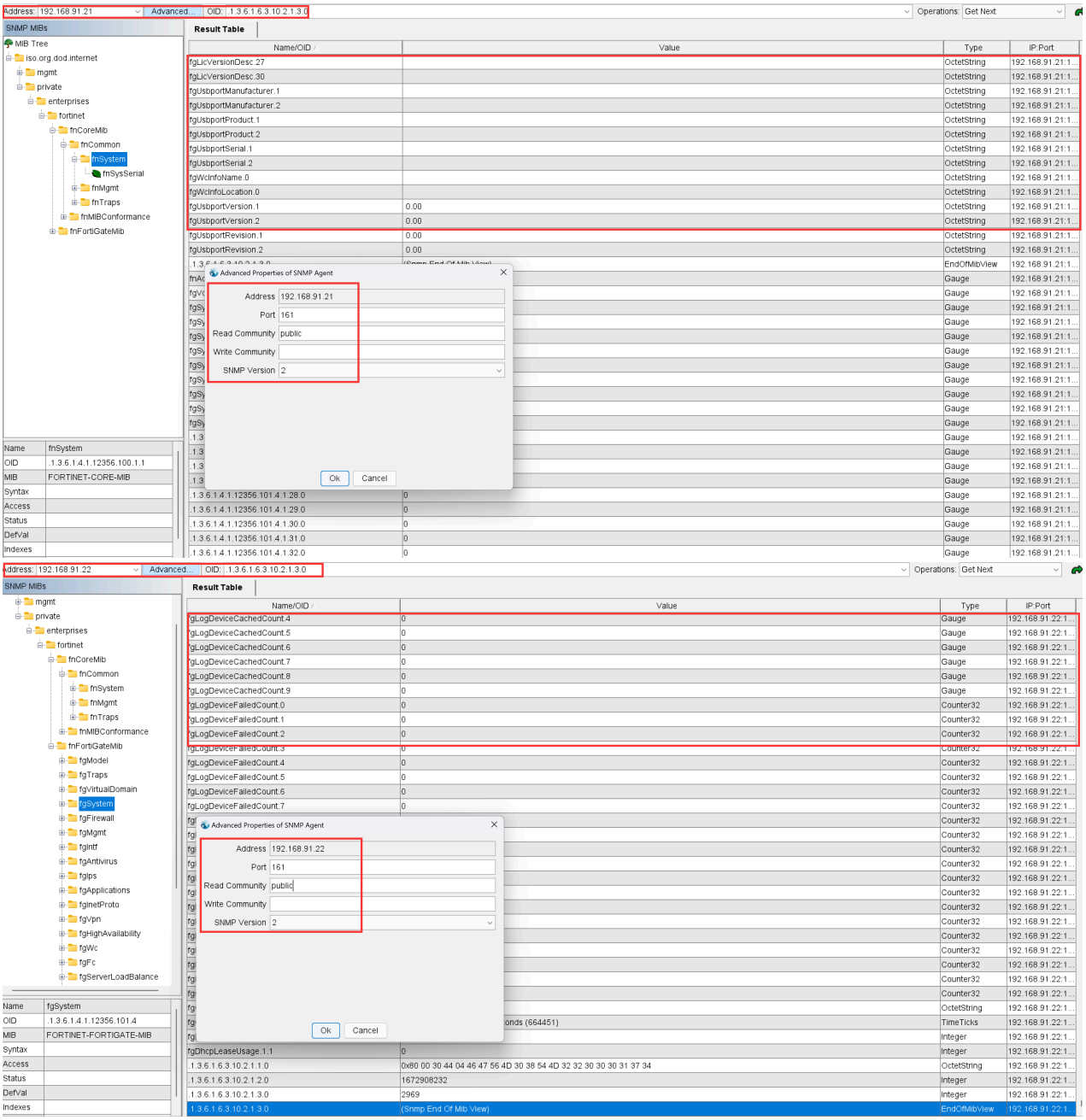
2. 进入全局 VDOM，开启 SNMP 功能，此配置主备同步。




```
config global
config system snmp sysinfo
    set status enable
    set description "FGT"
    set contact-info "support_cn@fortinet.com"
    set location "Beijing"
end

config system snmp community
    edit 1
        set name "public"
        config hosts
            edit 1
                set ip 192.168.91.105 255.255.255.255
            next
        end
    next
end
```

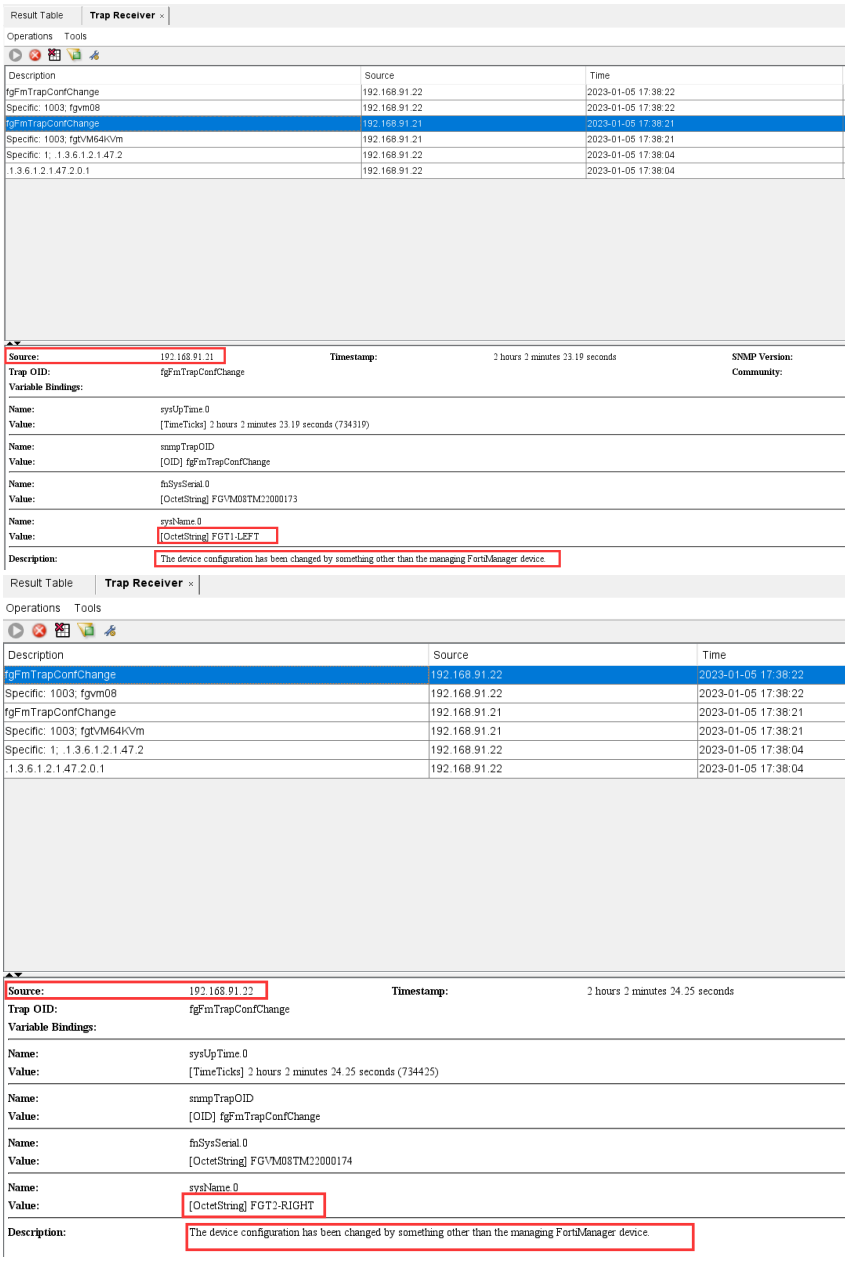
3. 在 SNMP 客户端对主设备和备设备的 port6 口进行 get/walk 操作，这里以 iReasoning MIB Browser 为例，可以正常读取 SNMP 节点信息。

```
FGT1-LEFT (MGMT) # dia sniff pa any "port 161" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 161]
44.345632 port6 in 192.168.91.105.50268 → 192.168.91.21.161: udp 45
44.346025 port6 in 192.168.91.105.50268 → 192.168.91.21.161: udp 45
44.346912 port6 out 192.168.91.21.161 → 192.168.91.105.50268: udp 45
44.347040 port6 out 192.168.91.21.161 → 192.168.91.105.50268: udp 45

FGT2-RIGHT (MGMT) # dia sniff pa any "port 161" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 161]
8.956697 port6 in 192.168.91.105.50265 → 192.168.91.22.161: udp 45
8.957084 port6 in 192.168.91.105.50265 → 192.168.91.22.161: udp 45
8.958027 port6 out 192.168.91.22.161 → 192.168.91.105.50265: udp 45
8.958123 port6 out 192.168.91.22.161 → 192.168.91.105.50265: udp 45
```



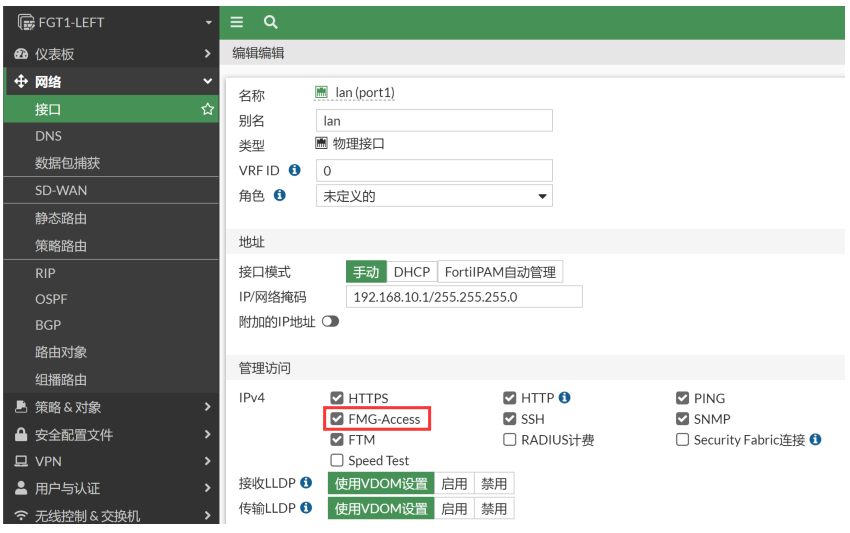
4. SNMP 客户端接收到的 FGT 发送的 trap 信息，可以看到主备设备均通过自身进行发送。

```
FGT1-LEFT (MGMT) # dia sniff pa any "port 162" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 162]
38.445858 port6 out 192.168.91.21.162 → 192.168.91.105.162: udp 104
38.446035 port6 out 192.168.91.21.162 → 192.168.91.105.162: udp 128

FGT2-RIGHT (MGMT) # dia sniff pa any "port 162" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 162]
150.833321 port6 out 192.168.91.22.162 → 192.168.91.105.162: udp 43
150.833537 port6 out 192.168.91.22.162 → 192.168.91.105.162: udp 68
168.867406 port6 out 192.168.91.22.162 → 192.168.91.105.162: udp 107
168.867504 port6 out 192.168.91.22.162 → 192.168.91.105.162: udp 129
```



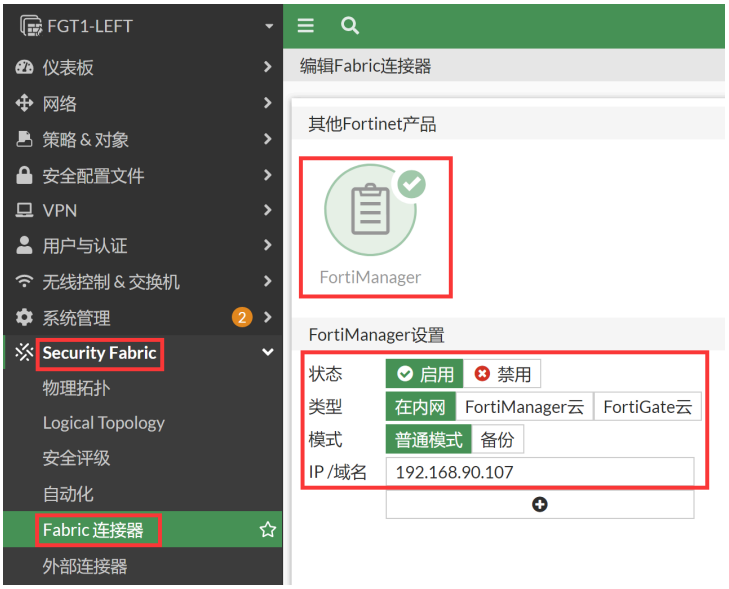
FortiManger 管理

1. FortiManager 只能通过业务接口添加，通过 FGT 的 port2（wan）口添加设备，在 FGT 的 wan 口下开启 FMG 管理协议。



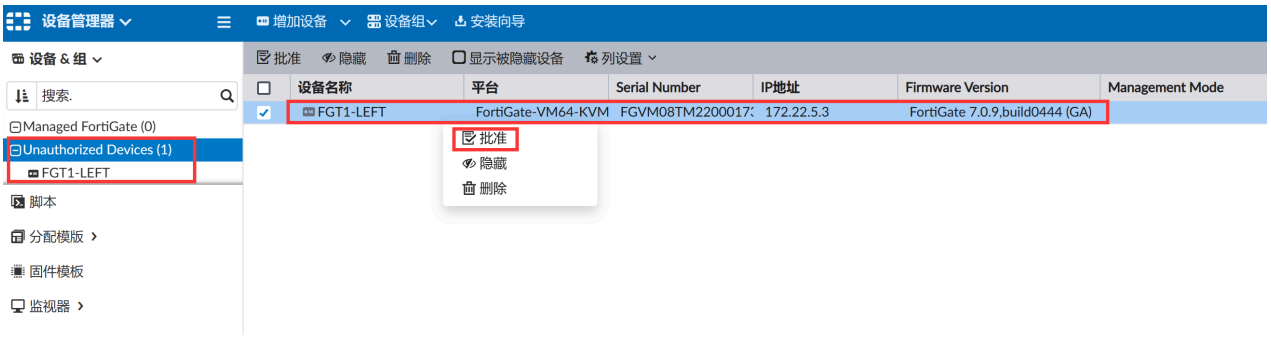
```
config global
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm ftm
        set type physical
        set alias "lan"
        set snmp-index 1
    next
end
```

2. 配置防火墙主动连接 FMG（如果 FMG 可以主动访问到 FGT，也可以通过 FMG 主动添加 FGT）。

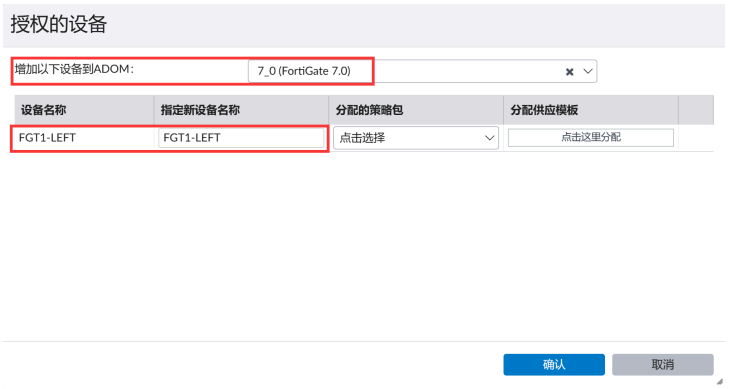



```
config global
config system central-management
    set type fortimanager
    set fmg "192.168.90.107"
end
```

3. 如果是 FGT 主动去注册 FMG，则 FMG 会有未注册的设备提示。



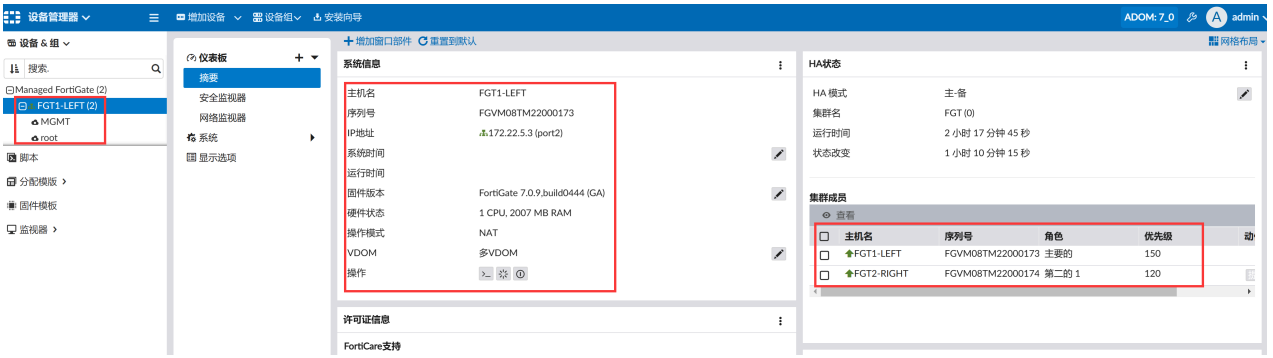
4. 只需要将设备添加到 FMG 的对应 ADOM 的设备管理中即可。



5. 授权完成。



6. 在 FMG 中可以对 FGT 进行管理。



7. 关于 FMG 还是建议使用业务接口进行管理，不能使用独立管理口/独立管理 VDOM 进行管理。对于 FMG 来说 HA-Cluster 就是一台设备，因此需要一个固定的 IP 地址来网管 FGT 会比较合适，可以有效的避免 HA 发生切换的时候，FGT 和 FMG 的 tunnel 中断的情况。

总结

此时所有的 SNMP/Radius/SYSLOG/DNS/FORTGUARD 都是通过 MGMT 这个 VDOM 进行更新的，完全独立于 root 业务 VDOM，彻底的管理是管理，业务是业务。这是 HA 独立管理的终极解决方案。

> **重要**
>
> FortiManger 管理 HA-Cluster 的时候只能通过业务接口进行管理，不能通过独立管理口或者独立管理 VDOM 进行管理。

单独指定用于更新的 VDOM

重要

此功能在 FortiOS 7.2.3 及以后版本上支持。

网络需求

FortiGate 工作于多 VDOM 场景，管理 VDOM 无法访问 Internet（如下管理 VDOM 配置），业务 VDOM 是可以访问 Internet 的，客户想使用业务 VDOM 来实现 Fortiguard 更新。

```
config system global
    set management-vdom "mgmt_vdom"
end
```

解决方案

1. 通过非管理 VDOM 实现 Fortiguard 更新。

```
config system fortiguard
    set vdom "UPDATE"
end
```

2. 连接 FortiGuard 需要该 VDOM 通过 DNS 服务器解析 FortiGuard 服务器域名，需要配置对应的 VDOM DNS，如下所示。

```
config system vdom-dns
    set vdom-dns enable
    set primary 223.5.5.5
end
```

具体请参考：https://community.fortinet.com/t5/FortiGate/Technical-Tip-How-to-use-non-management-VDOM-for-Fortiguard/ta-p/231010 ↗

HA 常见问题

- HA 同步机制和故障诊断
- HA 无法更新特征库/License

HA 同步机制和故障诊断

同步配置说明

1. FGCP 协议用联合的增量同步和周期性同步去确保所有 HA Cluster 中的成员与主设备的配置一致，下面的配置不会在 cluster 成员中同步：
   - 主机名
   - GUI 仪表盘中的微件
   - HA 中的 override 抢占
   - HA 设备 priority 优先级
   - 虚拟 cluster priority 优先级
   - HA 下的 Ping server 优先级
   - HA 专用管理接口的配置
   - HA 专用管理接口的下的默认路由配置
2. 多数订阅服务和授权是不同步的，每一个 FortiGate 授权是独立的（在系统 → FortiGuard 中，License 的到期时间只会显示较近到期的那台的时间，如果 HA 成员的某个 License 过期，那么所有 HA 成员的该 License 都会过期）。

   > **重要**
   >
   > FortiToken Mobile 是一个例外，它们注册到主设备然后同步给备设备。

3. 主设备同步所有其它的配置给集群成员，包括 HA 下的一些配置。
4. 所有同步活动在 HA hearbert 心跳接口用 TCP/703 和 UDP/703。

禁用自动同步配置

1. 在一些情况下你可能想用下面的命令关掉自动同步主设备的配置到所有集群中的设备。

```
config system ha
    set sync-config disable
end
```

2. 当这个选项 disable 禁用后，集群不再同步配置变更。如果一个设备发生故障，新的主设备可能没有相同的配置和失败的主设备。结果，新的主设备可能处理会话不同或者不能以相同的方式工作。
3. 在绝大多数情况下不应该 disable 禁用自动同步配置。然而，如果你 disable 自动同步，你可以用 execute ha synchronize 命令手工的同步配置。
4. 你必须在备设备上执行 execute ha synchronize。用 execute ha manage 能切换到备设备上。
5. 例子：访问备设备然后执行 HA 同步，即便自动同步是禁用的。

```
execute ha manage 0    //切到备设备上
execute ha synchronize start    //执行HA同步
```

6. 你能用下面的命令停止在同步中的过程。

```
execute ha synchronize stop
```

增量同步

1. 当你登录进集群用 GUI 或者 CLI 做配置变更时，你实际上登录在主设备上。
2. 所有的配置首先被做应用到主设备，增量同步然后立即同步这些变更给其它成员设备。
3. 当你登录备设备的 CLI（例如使用 execute ha manage 切换登录进备设备），所有的在备设备做的配置变更也立即同步到所有集群成员，包括主设备。
4. 增量同步也同步其它动态的配置信息例如 DHCP Server address lease 数据库，路由更新，IPSec SAs，MAC 地址表 等等。
5. 只要在一个集群成员做配置变更，增量同步将通过心跳链接发送相同的信息到所有其它成员。
6. 运行在每一个成员 HA 同步进程接收配置变更并应用它。
7. HA 同步进程通过输入 CLI 命令进行配置更改，该命令看起来像是由第一次进行配置更改的管理员输入的。

8. 同步静默的进行，没有日志记录 HA 同步活动。然而，当同步进程执行 CLI 命令，日志消息能被记录在成员设备，你能看到这些消息在成员设备如果你启用了事件日志并且日志级别设置为 information 信息。
9. 你也能看到这些日志在主设备当你做了配置变更在成员设备上。

周期性同步

1. 增量同步确保管理员做的配置变更，这边配置变更依然是相同的在集群成员。
2. 然而，一些因素可能导致一个或多个集群成员与主设备不同步。例如，当增加一个新的成员到集群中，这个新的成员的配置没有匹配其它成员，使用增量同步配置变更对于新的成员来说是不现实的。
3. 周期性同步是一种寻找同步问题并修复它们的机制，集群每分钟都会将主设备的配置文件校验和与每个成员设备的配置文件校验和进行比较。
4. 如果所有备设备 checksums 校验和是与主设备一致，所有的集群成员认为是同步的。
5. 如果一个或多个备设备的 checksums 校验和与主设备不同，备设备的配置被认为 out of sync 不同步与主设备，checksums 校验和不同步的备设备每 15 秒检查一次，如果由于增量配置序列未完成而导致配置不同步，则需要重新检查。
6. 如果校验和在 5 次检查后不匹配，那么不同步的备设备将从主设备索要配置。
7. 然后，备设备重新加载其配置，并作为备设备以与主设备相同的配置继续运行。
8. 备设备的配置通过这种方式重置因为当备设备配置与主设备配置不同步时，没有有效的方法来确定配置的差异是什么并纠正它们。重置备设备配置成为重新同步备设备的最有效方法。
9. 同步要求所有集群成员运行在相同的 FortiOS 版本，如果一些成员运行在不同的版本，不稳定和同步错误有可能发生。

同步成功的 Console 上输出的消息

当集群最初形成时，或者当一个新成员作为从属成员添加到集群中时，CLI 控制台上会出现以下消息，表明该成员加入了集群，并使其配置与主设备同步。

```
slave's configuration is not in sync with master's, sequence:0
slave's configuration is not in sync with master's, sequence:1
slave's configuration is not in sync with master's, sequence:2
slave's configuration is not in sync with master's, sequence:3
slave's configuration is not in sync with master's, sequence:4
slave starts to sync with master
logout all admin users
slave succeeded to sync with master
```

同步失败的 Console 上输出的消息

1. 如果您连接到与主设备不同步的备设备的 console 控制台，将显示类似如下的消息。

```
slave is not in sync with master, sequence:0. (type 0×3)
slave is not in sync with master, sequence:1. (type 0×3)
slave is not in sync with master, sequence:2. (type 0×3)
slave is not in sync with master, sequence:3. (type 0×3)
slave is not in sync with master, sequence:4. (type 0×3)
global compared not matched
```

2. 如果出现同步问题，Console 控制台消息序列可能会一再重复。
3. 所有消息都包含一个类型值 (type 0x3)。type 值可以帮助诊断同步问题。

HA 对象消息和它们引用的配置对象不同步的参考

消息类型	配置对象
HA_SYNC_SETTING_CONFIGURATION = 0x03	/data/config
HA_SYNC_SETTING_AV = 0x10	
HA_SYNC_SETTING_VIR_DB = 0x11	/etc/vir
HA_SYNC_SETTING_SHARED_LIB = 0x12	/data/lib/libav.so
HA_SYNC_SETTING_SCAN_UNIT = 0x13	/bin/scanunitd
HA_SYNC_SETTING_IMAP_PRXY = 0x14	/bin/imapd
HA_SYNC_SETTING_SMTP_PRXY = 0x15	/bin/smtp
HA_SYNC_SETTING_POP3_PRXY = 0x16	/bin/pop3
HA_SYNC_SETTING_HTTP_PRXY = 0x17	/bin/thttp
HA_SYNC_SETTING_FTP_PRXY = 0x18	/bin/ftpd
HA_SYNC_SETTING_FCNI = 0x19	/etc/fcni.dat
HA_SYNC_SETTING_FDNI = 0x1a	/etc/fdnservers.dat
HA_SYNC_SETTING_FSCI = 0x1b	/etc/sci.dat
HA_SYNC_SETTING_FSAE = 0x1c	/etc/fsae_adgrp.cache
HA_SYNC_SETTING_IDS = 0x20	/etc/ids.rules
HA_SYNC_SETTING_IDSUSER_RULES = 0x21	/etc/idsuser.rules
HA_SYNC_SETTING_IDSCUSTOM = 0x22	
HA_SYNC_SETTING_IDS_MONITOR = 0x23	/bin/ipsmonitor
HA_SYNC_SETTING_IDS_SENSOR = 0x24	/bin/ipsengine
HA_SYNC_SETTING_NIDS_LIB = 0x25	/data/lib/libips.so
HA_SYNC_SETTING_WEBLISTS = 0x30	
HA_SYNC_SETTING_CONTENTFILTER = 0x31	/data/cmdb/webfilter.bword
HA_SYNC_SETTING_URLFILTER = 0x32	/data/cmdb/webfilter.urlfilter
HA_SYNC_SETTING_FTGD_OVRD = 0x33	/data/cmdb/webfilter.fgtd-ovrd
HA_SYNC_SETTING_FTGD_LRATING = 0x34	/data/cmdb/webfilter.fgtd-ovrd
HA_SYNC_SETTING_EMAILLISTS = 0x40	
HA_SYNC_SETTING_EMAILCONTENT = 0x41	/data/cmdb/spamfilter.bword
HA_SYNC_SETTING_EMAILBWLIST = 0x42	/data/cmdb/spamfilter.emailbwl
HA_SYNC_SETTING_IPBWL = 0x43	/data/cmdb/spamfilter.ipbwl
HA_SYNC_SETTING_MHEADER = 0x44	/data/cmdb/spamfilter.mheader
HA_SYNC_SETTING_RBL = 0x45	/data/cmdb/spamfilter.rbl
HA_SYNC_SETTING_CERT_CONF = 0x50	/etc/cert/cert.conf
HA_SYNC_SETTING_CERT_CA = 0x51	/etc/cert/ca
HA_SYNC_SETTING_CERT_LOCAL = 0x52	/etc/cert/local

HA_SYNC_SETTING_CERT_CRL = 0x53	/etc/cert/crl
HA_SYNC_SETTING_DB_VER = 0x55	
HA_GET_DETAIL_CSUM = 0x71	
HA_SYNC_CC_SIG = 0x75	/etc/cc_sig.dat
HA_SYNC_CC_OP = 0x76	/etc/cc_op
HA_SYNC_CC_MAIN = 0x77	/etc/cc_main
HA_SYNC_FTGD_CAT_LIST = 0x7a	/migadmin/webfilter/ublock/ftgd/data/

checksums 校验和

1. 可以使用 diagnose sys ha checksum show 命令比较所有集群单元的配置校验和。该命令的输出显示了标记为 global 和 all 的校验和，以及每个 VDOM (包括 root VDOM) 的校验和。可以使用 get system ha-nonsync-csum 命令显示类似信息（这个命令仅供 FortiManager 使用）。
2. 主设备和备设备用这个命令 show 出来的 checksums 校验和应该是相同的。如果他们不同你可以用 execute ha synchronize start 命令做一下强制同步。
3. 下面命令的输出是主设备的（没有多个 VDOM 启用）。

```
diagnose sys ha checksum show
is_manage_master()=1, is_root_master()=1
debugzone
global: a0 7f a7 ff ac 00 d5 b6 82 37 cc 13 3e 0b 9b 77
root: 43 72 47 68 7b da 81 17 c8 f5 10 dd fd 6b e9 57
all: c5 90 ed 22 24 3e 96 06 44 35 b6 63 7c 84 88 d5

checksum
global: a0 7f a7 ff ac 00 d5 b6 82 37 cc 13 3e 0b 9b 77
root: 43 72 47 68 7b da 81 17 c8 f5 10 dd fd 6b e9 57
all: c5 90 ed 22 24 3e 96 06 44 35 b6 63 7c 84 88 d5
```

4. 下面命令的输出是在相同集群中的备设备的。

```
diagnose sys ha checksum show
is_manage_master()=0, is_root_master()=0
debugzone
global: a0 7f a7 ff ac 00 d5 b6 82 37 cc 13 3e 0b 9b 77
root: 43 72 47 68 7b da 81 17 c8 f5 10 dd fd 6b e9 57
all: c5 90 ed 22 24 3e 96 06 44 35 b6 63 7c 84 88 d5

checksum
global: a0 7f a7 ff ac 00 d5 b6 82 37 cc 13 3e 0b 9b 77
root: 43 72 47 68 7b da 81 17 c8 f5 10 dd fd 6b e9 57
all: c5 90 ed 22 24 3e 96 06 44 35 b6 63 7c 84 88 d5
```

5. 下面的输出是启用了 VDOM（test 和 Eng_vdm.）的主设备输出。

```
config global
diagnose sys ha checksum show
is_manage_master()=1, is_root_master()=1
debugzone
global: 65 75 88 97 2d 58 1b bf 38 d3 3d 52 5b 0e 30 a9
test: a5 16 34 8c 7a 46 d6 a4 1e 1f c8 64 ec 1b 53 fe
root: 3c 12 45 98 69 f2 d8 08 24 cf 02 ea 71 57 a7 01
Eng_vdm: 64 51 7c 58 97 79 b1 b3 b3 ed 5c ec cd 07 74 09
all: 30 68 77 82 a1 5d 13 99 d1 42 a3 2f 9f b9 15 53

checksum
```

```
global: 65 75 88 97 2d 58 1b bf 38 d3 3d 52 5b 0e 30 a9
test: a5 16 34 8c 7a 46 d6 a4 1e 1f c8 64 ec 1b 53 fe
root: 3c 12 45 98 69 f2 d8 08 24 cf 02 ea 71 57 a7 01
Eng_vdm: 64 51 7c 58 97 79 b1 b3 b3 ed 5c ec cd 07 74 09
all: 30 68 77 82 a1 5d 13 99 d1 42 a3 2f 9f b9 15 53
```

6. 备设备输出。

```
config global
diagnose sys ha checksum show
is_manage_master()=0, is_root_master()=0
debugzone
global: 65 75 88 97 2d 58 1b bf 38 d3 3d 52 5b 0e 30 a9
test: a5 16 34 8c 7a 46 d6 a4 1e 1f c8 64 ec 1b 53 fe
root: 3c 12 45 98 69 f2 d8 08 24 cf 02 ea 71 57 a7 01
Eng_vdm: 64 51 7c 58 97 79 b1 b3 b3 ed 5c ec cd 07 74 09
all: 30 68 77 82 a1 5d 13 99 d1 42 a3 2f 9f b9 15 53

checksum
global: 65 75 88 97 2d 58 1b bf 38 d3 3d 52 5b 0e 30 a9
test: a5 16 34 8c 7a 46 d6 a4 1e 1f c8 64 ec 1b 53 fe
root: 3c 12 45 98 69 f2 d8 08 24 cf 02 ea 71 57 a7 01
Eng_vdm: 64 51 7c 58 97 79 b1 b3 b3 ed 5c ec cd 07 74 09
all: 30 68 77 82 a1 5d 13 99 d1 42 a3 2f 9f b9 15 53
```

诊断 HA out of sync 不同步问题

这部分描述怎样使用 diagnose sys ha checksum show 和 diagnose debug 命令诊断 HA 不同步消息。如果同步不成功，用下面的步骤在每一个成员执行发现问题。

确定为什么没有进行 HA 同步

1. 连接每一个设备的 console 口（或者 SSH 登录）。
2. 使用下面的 debug 命令启用诊断。

```
diagnose debug enable
diagnose debug console timestamp enable
diagnose debug application hatalk -1
diagnose debug application hasync -1
```

3. 收集 debug 输出信息，然后对比上个章节中的消息信息表格。
4. 关掉 debug。

```
diagnose debug disable
diagnose debug reset
```

确定哪一部分配置导致的问题

1. 如果上面的步骤显示的消息中包含了同步对象 0x30（例如，HA_SYNC_SETTING_CONFIGURATION = 0x03），则说明配置存在同步问题。使用以下步骤来确定导致问题的配置部分。
2. 如果您的集群由两个集群成员组成，则使用此过程捕获每个成员的配置校验和。如果您的集群包含两个以上的集群成员，则对所有返回包含 0x30 同步对象消息的集群成员重复此过程。
3. 连接每个成员的 console 口（或者 SSH 命令行）。
4. 启用 debug 命令。

```
diagnose debug enable
```

5. 停止 HA 同步。

```
execute ha sync stop
```

6. 显示 global 下的 checksum 校验和。

```
diagnose sys ha checksum show global
```

7. 把输出存为一个 text 文件。
8. 在所有集群设备重复这个过程。
9. 比较主设备的 text 文件和成员设备的，发现不匹配的部分。可以用一个文本对比器来对比。
10. 对比 root vdom 的校验和。

```
diagnose sys ha checksum show root
```

11. 对比所有 VDOM 的校验和。

```
diagnose sys ha checksum show <vdom-name>
```

12. 你也可以用 grep 选项仅仅显示部分的 checksums。例如，仅仅显示和 system 相关的 checksums 在 root Vdom 或者 log 相关的 checksums 在 global 下的，通常情况下第一个不匹配的校验和意味这个是导致不同问题的原因。

```
diagnose sys ha checksum root | grep system
diagnose sys ha chechsum global | grep log
```

13. 尝试删除或者改变这导致问题配置部分，你可以在主设备或者备设备上修改配置。
14. 然后用下面的命令重启开启 HA 同步，停掉 debug。

```
execute ha sync start
diagnose debug disable
diagnose debug reset
```

重新计算 checksums 校验和解决 HA 不同步问题

1. 有时，在集群计算校验和时可能会发生错误。由于这种计算错误，即使集群正常运行，CLI 控制台也可能显示 out of sync 错误消息。有时，在 diagnostic sys ha checksum 命令的输出中也会出现校验和计算错误，原因是 debug zone 中的校验和与输出的校验和部分不匹配。
2. 解决这个问题的一个方法是重新计算校验和。重新计算的校验和应该匹配，不同步的错误消息应该停止出现。
3. 您可以使用以下命令重新计算 HA 校验和。

```
diagnose sys ha checksum recalculate [<vdom-name> | global]
```

4. 仅仅输入不带选项的命令就会重新计算所有校验和。您可以指定一个 VDOM 名称来重新计算该 VDOM 的校验和。您也可以输入 global 来重新计算全局校验和。

确定导致配置同步问题的原因

有 25 个配置同步的 FortiOS 模块。要分析这么多数据，它是困难的找到导致同步问题的原因。您可以使用以下诊断命令更容易地找到可能导致同步问题的模块。

```
diagnose sys ha hasync-stats {all | most-recent <seconds> | by object <number>}
```

- all 显示自 hasync 进程开始运行 (通常是集群启动后) 以来发生的所有模块的同步活动。
- `most-recent [<seconds>]` 显示最近发生的同步事件。可以包括以秒为单位的时间，以显示在时间间隔内发生的最近事件。如果不包含秒数，该命令将显示最近 5 秒内的最新事件。此选项可用于确定当前正在同步或试图同步的一个或多个模块。如果当前没有正在同步的模块，该命令只显示最近的同步事件。
- `by-object [<number>]` 显示指定模块的同步活动，`<number>` 是模块号，取值范围是 1 ~ 25。要显示所有 25 个模块及其编号，请输入：

```
diagnose sys ha hasync-stats by-object ?
```

- 要显示最近的活动，输入：

```
diagnose sys ha hasync-stats most-recent 10
current-time/jiffies=2018-03-28 13:01:42/1148242:
hasync-obj=2(arp):
   epoll_handler=1(ev_arp_handler): start=1522256500.354400(2018-03-28 13:01:40), end=1522256500.354406(2018-03-28
13:01:40), total=0.000006/1699
```

```
hasync-obj=5(config):
    timer=0(check_sync_status), add=1141764(2018-03-28 13:01:26), expire=1142764(2018-03-28 13:01:36),
end=1142764(25018-03-28 13:01:36), del=0(), total_call=1143
hasync-obj=8(time):
   obj_handler=0(packet): start=1522256497.851550(2018-03-28 13:01:37), end=1522256497.851570(2018-03-28 13:01:37),
total=0.000020/381
   timer=0(sync_timer), add=1140106(2018-03-28 13:01:10), expire=1143106(2018-03-28 13:01:40), end=1143106(2018-03-28
13:01:40), del=0(), total_call=381
hasync-obj=21(hastats):
   obj_handler=0(packet): start=1522256499.760934(2018-03-28 13:01:39), end=1522256499.760936(2018-03-28 13:01:39),
total=0.000002/2285
   timer=0(hastats_timer), add=1142556(2018-03-28 13:01:34), expire=1143056(2018-03-28 13:01:39), end=1143056(2018-03-28
13:01:39), del=0(), total_call=2286
```

- 该输出的最后几行显示了 hastats 模块的活动，该模块是模块 21。您可以使用以下命令查看更多关于此模块的同步活动的信息：

```
diagnose sys ha hasync-stats by-object 21
```

HA 无法更新特征库/License

问题现象

自 2025 年 2 月下旬起，当 FortiGate 的 HA 成员注册到不同 FortiCloud 账户（或其中一个未注册账户）时，HA 集群会出现如下情况：

- 无法执行在线固件升级。
- 不能在线更新 IPS、AV、应用控制等特征库。
- 不能更新 License 合同信息。

诊断方法

1. 在 CLI 下执行以下 Debug 命令。

```bash
diagnose debug application update -1
diagnose debug enable
execute update-now
```

2. 完成后，使用以下命令清除调试信息。

```bash
diagnose debug disable
diagnose debug reset
```

3. Debug 日志中会出现类似“Missing contracts, got X, expect X”的错误信息，如下所示。

```bash
upd_act_report_fmg_list[846]-ContractItem (1) does not contain all HA (2): FGVM04TMXXXXXX03
<>
upd_status_set_ha_expiry[1532]-Serial Number: FGVM04TMXXXXXX03 - contract processed
upd_status_set_ha_expiry[1498]-Extracting contract ... (SupportLevelDesc=06:Web/Online*10:8*5*20:Premium)
<>
upd_status_set_ha_expiry[1547]-Missing contracts, got 1, expect 2  ←——错误信息
upd_status_set_ha_expiry[1565]-Reset expiry
do_update[690]-UPDATE failed
```

解决方法

1. 确保 FortiGate HA 集群成员注册到相同的 FortiCloud 账户。
2. 如果 HA 成员不在同一 FortiCloud 账户下，请联系 support_cn@fortinet.com ↗ 进行账户转移。

Workaround

1. 可以关闭 HA 成员的备机，或将其进行网络隔离，然后通过 HA 主设备执行 FortiGuard 更新。
2. 使用相同的办法更新 HA 的其他成员。
3. 更新完成后，将其他成员恢复到 HA 集群中。

路由

路由介绍

静态路由

策略路由

组播

- 组播转发
- TTL=1 的组播无法转发
- 将被拒绝的组播会话加入会话表

路由常见问题

- 黑洞路由不生效
- 策略路由不生效

动态路由

- RIP
- OSPF

  OSPF 基础配置 OSPF GR
- BGP

  BGP 基础配置 BGP 计时器 BGP GR BGP 路由过滤 BGP 路由刷新 BGP 下一跳递归 路由反射器 BGP 多路径 additional-path 附加 AS-Path BGP Debug BGP 常见问题
- 路由对象

  Route Map Access List Prefix List AS Path List Community List
- 动态路由协议排错

BFD

- BFD 原理
- BFD 基本配置
- BFD 与静态路由
- BFD 与 OSPF
- BFD 与 BGP
- BFD 多跳模式

路由介绍

路由转发表

1. FIB (Forwarding Information Base)，包含所有已知的本地或非本地路由，从路由表载入，由内核访问。

2. 内核转发数据时看 diagnose ip route list，而非 get router info routing-table all。

```
diagnose ip route list
```

路由表

1. 路由表包括最具体的路由条目、最长掩码匹配原则、路由的管理距离、度量值等决定哪个协议产生的哪条路由被内核采用。

```
get router info routing-table all
```

2. 路由表中的相关元素。

Routing-table（路由表）元素

```
# get router info routing-table database
O E2 *> 10.0.0.1/32     [110/10] via 192.168.90.41, wan1, 18:40:59
O IA *> 192.168.8.0/24  [110/2] via 192.168.90.41, wan1, 18:40:59
O       192.168.90.0/24 [110/1] is directly connected, wan1, 19:15:57
C    *> 192.168.90.0/24 is directly connected, wan1
S    *> 192.168.0.0/16 [5/0] via 192.168.90.253, wan1, [2/4]
S       192.168.0.0/16 [10/0] via 192.168.90.254, wan1, [1/2]
```

目的网络 /子网掩码　　下一跳地址/出接口

Distance（管理距离）　Metric（度量值）　Priority（优先级）　Weight（权重）

```
#get router info routing-table all
O E2    10.0.0.1/32     [110/10] via 192.168.90.41, wan1, 18:44:24
O IA    192.168.8.0/24  [110/2] via 192.168.90.41, wan1, 18:44:24
C       192.168.90.0/24 is directly connected, wan1
O       192.168.90.0/24[110/1] is directly connected, wan1, 19:07:01
S       192.168.0.0/16 [5/0] via 192.168.90.253, wan1, [2/4]
```

- Distance（管理距离）

Distance（管理距离）

一条相同的路由条目通过不同的路由协议（RIP/IS-IS/OSPF）学习到，那么哪一条应该安装进转发路由表呢？

管理距离：路由协议及该路由协议产生的路由的优先级，0-256，越小越优先。一般只在本设备内有效。

FortiGate默认管理距离值：

直连路由	0
DHCP网关	5
静态路由	10
EBGP	20
OSPF	110
IS-IS	115
RIP	120
IBGP	200

- Metric（度量值）

Metric（度量值）

一条相同的路由条目通过相同的路由协议（RIP）学习到，那么哪一条应该安装进转发路由表呢?

度量值：路由协议产生的路由到目标网络的开销，越小越优先，只在同一路由协议内起作用，不同的路由协议产生的路由度量值没有可比性。

RIP	路由的跳数作为度量值
OSPF/IS-IS	路径Cost之和作为度量值
BGP	Metric是起一个可选不可传递的属性

- Priority（优先级）

Priority（优先级）

想要实现浮动路由的效果，同时备份的出口又需要能够作为SSL VPN的登录IP/VIP映射/管理IP使用?

- 静态路由使用它来计算到达目的地址的最佳路由
- 如果多条静态路由拥有相同的distance值，而拥有不同的Priority
  - 他们全都会被加载到路由表中
  - 但是，只有拥有最低优先级priority值的路由条目会被用来指导转发

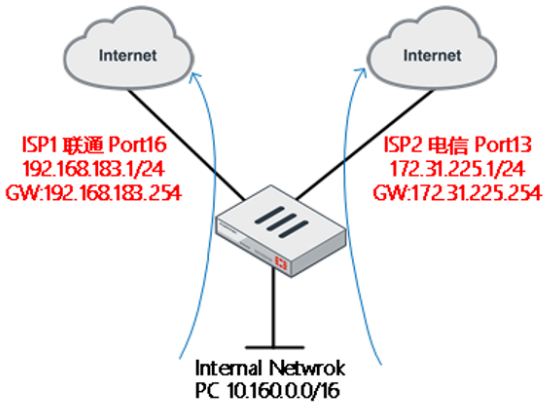
Priority通常用于以下情形中：为了接受入方向的流量（VIP/VPN/SNMP请求），需要在路由转发表中存在一条静态路由，以便通过Reverse Path Forwarding检查（Anti-Spoofing），否则数据将被丢弃。

Priority 越小越优先，低Priority的路由用于出方向的目的路由转发，高Priority的路由只是存在于路由表中，不用于出方向目的路由转发，当有数据从高Priority的接口入时，会启用RPF源路由检查，只有通过RPF检查的数据才会被防火墙所处理，这条高Priority的路由存在的意义正在于此。

路由选路

1. 情景 1：相同的 distance，相同的 priority。

Distance（管理距离） /Priority（优先级）路由选择



Scenario 1 : same distance , same priority.

配置:

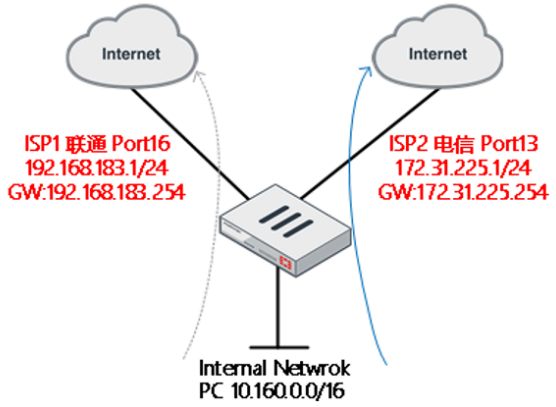
```
config router static
    edit 1
        set device "port16"
        set distance 10 （默认）
        set gateway 192.168.183.254
    next
    edit 2
        set device "port14"
        set distance 10 （默认）
        set gateway 172.31.225.254
    next
end
```

结果：EMCP（等价链路负载）

```
FGT# get router  info routing-table static
S*      0.0.0.0/0 [10/0] via 172.31.225.254, port13
                  [10/0] via 192.168.183.254, port16
```

2. 情景 2：不同的 distance，相同的 priority。

Distance（管理距离） /Priority（优先级）路由选择



Scenario 2 : different distance , same priority.

配置:

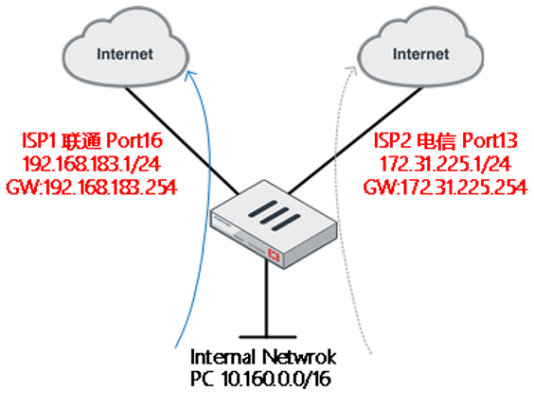
```
config router static
    edit 1
        set device "port16"
        set distance 10 （默认）
        set gateway 192.168.183.254
    next
    edit 2
        set device "port13"
        set distance 5
        set gateway 172.31.225.254
    next
end
```

结果：选择管理距离小的

```
FGT# get router  info routing-table static
S*      0.0.0.0/0 [5/0] via 172.31.225.254, port13
```

3. 情景 3：相同的 distance，不同的 priority。

Distance（管理距离） /Priority（优先级）路由选择



Scenario 3a : same distance , different priority.

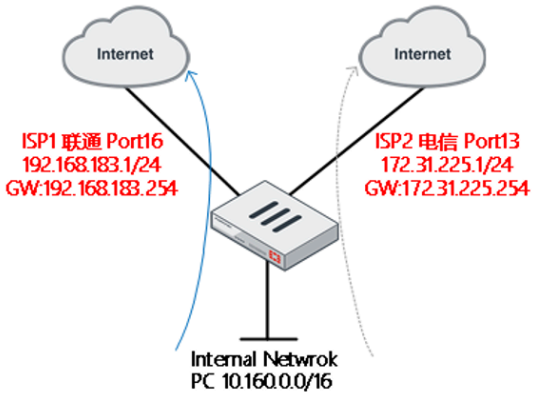
配置:

```
config router static
    edit 1
        set device "port16"
        set gateway 192.168.183.254
        set priority 2
    next
    edit 2
        set device "port13"
        set gateway 172.31.225.254
        set priority 5
    next
end
```

结果：都被加入路由表，但只有优先级小的用于路由转发数据

```
FGT# get router  info routing-table static
S*      0.0.0.0/0 [10/0] via 192.168.183.254, port16, [2/0]
                  [10/0] via 172.31.225.254,  port13, [5/0]
```

Distance（管理距离） /Priority（优先级）路由选择



Scenario 3b : same distance , different priority.

配置:

```
config router static
    edit 1
        set device "port16"
        set gateway 192.168.183.254
        set priority 2
    next
    edit 2
        set device "port13"
        set gateway 172.31.225.254
        set priority 1
    next
end
```

结果：都被加入路由表，但只有优先级小的用于路由转发数据

```
FGT# get router  info routing-table static
S*      0.0.0.0/0 [10/0] via 172.31.225.254,  port13, [1/0]
                  [10/0] via 192.168.183.254, port16, [2/0]
```

4. Weight（权重）对选路的影响。

Weight（权重）

多出口组网中如果配置了多条等价缺省路由，设备使用HASH算法使流量均衡分担到各个链路上。缺省情况下基于源IP负载。当链路带宽不同时，为了充分利用链路带宽还可以按带宽比例进行流量分担。例如WAN1对应链路带宽为10M，WAN2对应链路带宽为20M，可以在出接口上指定权重比为10:20

```
system virtual-wan-link
set v4-ecmp-mode weight-based
end
config system interface（配置接口的权重）
    edit "wan1"
       set weight 10
       next
    edit "wan2"
       set weight 20
end
```

OR

```
system virtual-wan-link
    set v4-ecmp-mode weight-based
end
config router static （配置路由的权重）
    edit 1
        set gateway 101.1.1.2
        set weight 10
        set device "wan1"
    next
    edit 2
        set gateway 202.1.1.2
        set weight 20
        set device "wan2"
    next
end
```

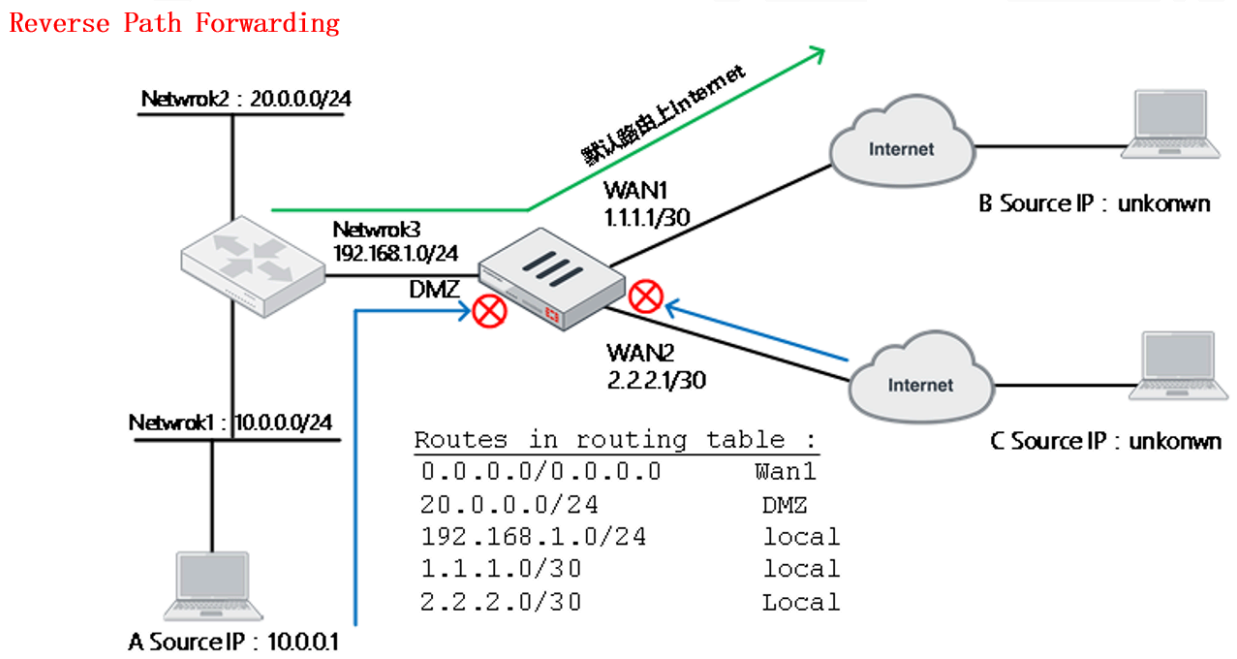
5. 总结：

- 策略路由比路由表优先
- 如果接口 down 则没有相关的路由（包括没有直连路由）
- 目标网段不同，都优选
- 目标网段相同，子网掩码不同时，都优选: 转发时最长掩码匹配目标网段、子网掩码都相同时，最低管理距离优先
- 目标网段、子网掩码、管理距离都相同时，最低度量值优先目标网段、子网掩码、管理距离、度量值都相同时，视具体协议而定，一般可形成等开销负载分担路由

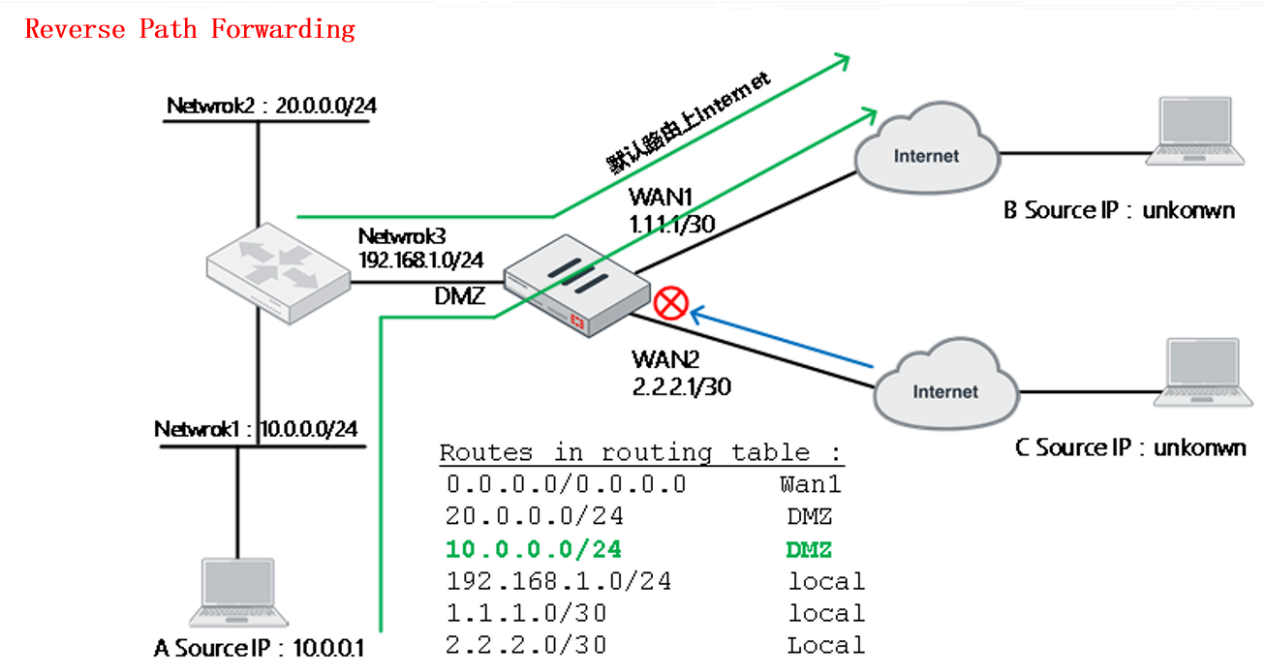
RPF（Reverse Path Forwarding，反向路径转发）

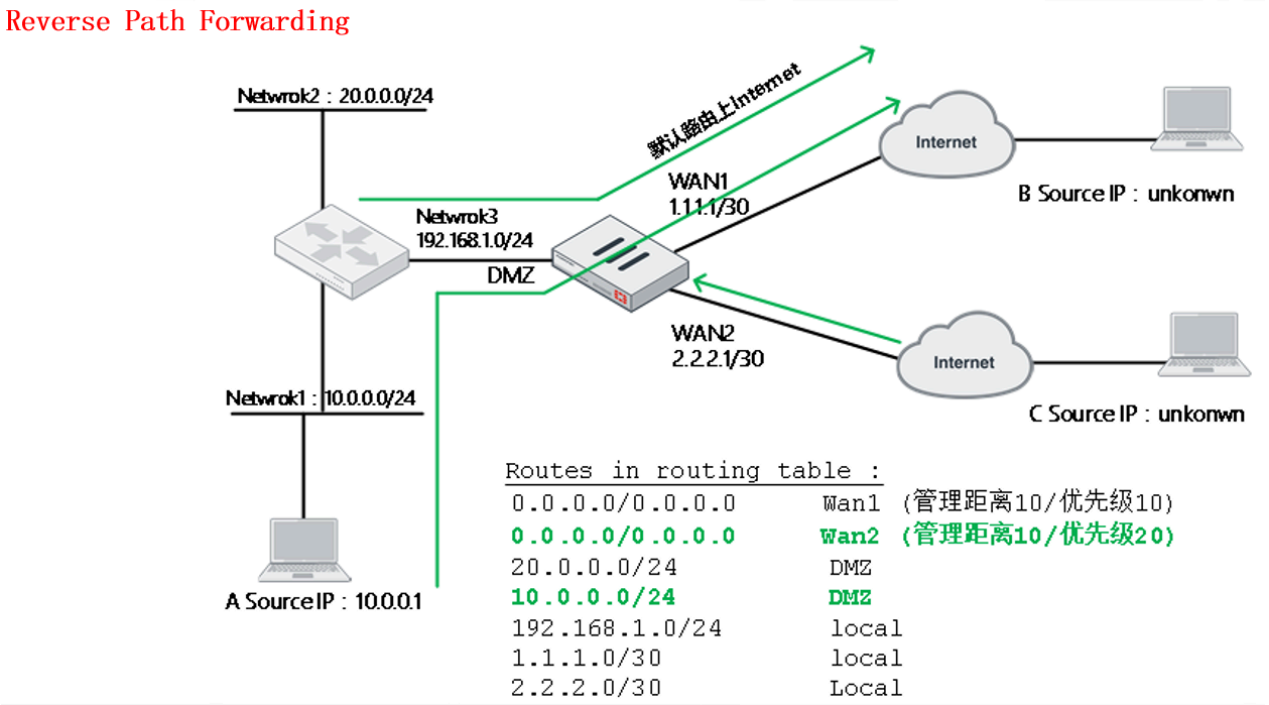
RPF 的作用：

- 阻止 IP 欺骗攻击
  - 测试 DDoS 不成功的原因（攻击源 IP 为随机 IP）
- 校验所有包的源地址
  - 和接口相关的路由比较 IP 地址
  - 丢弃那些源 IP 没有返回路由的包



- 如果源 IP 地址的返回路径与包来的路径不匹配，则这个包被丢弃



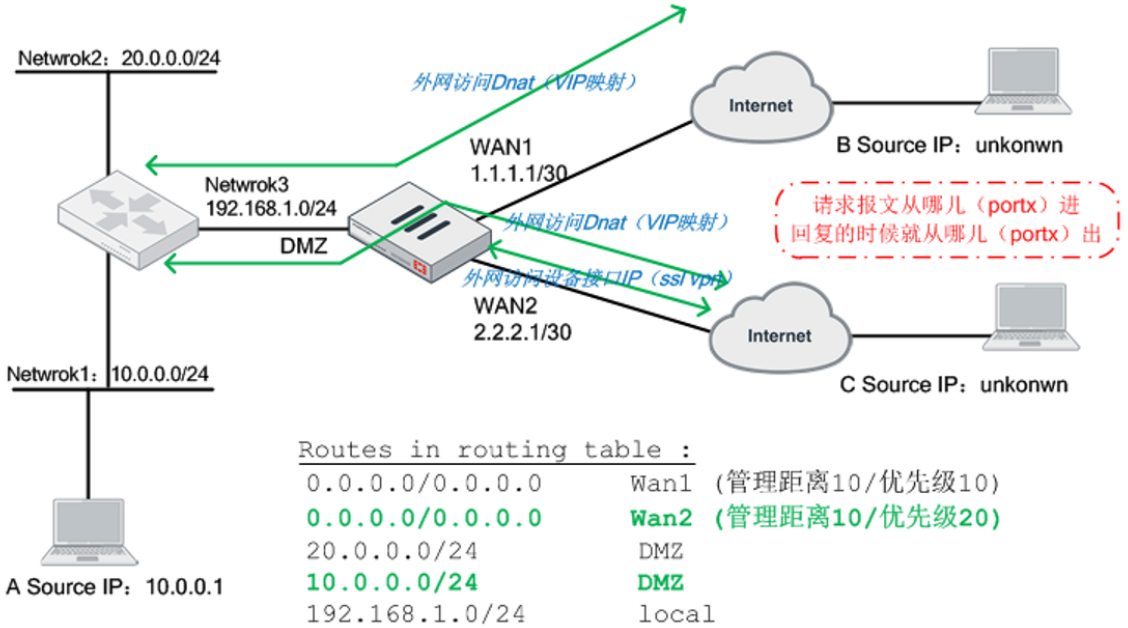


- RPF 只检查会话的第一个包
  - 来去路径相同时，不检查回应包（也没有必要）
  - 异步情况下，对回包也执行检查

源进源出

1. 源进源出介绍：

原接口进原接口出属性（请求报文从哪个接口进，回复报文必然从哪个接口出）




```
Routes in routing table :
0.0.0.0/0.0.0.0          Wan1 (管理距离10/优先级10)
0.0.0.0/0.0.0.0          Wan2 (管理距离10/优先级20)
20.0.0.0/24              DMZ
10.0.0.0/24              DMZ
192.168.1.0/24           local
1.1.1.0/30               local
2.2.2.0/30               Local
```

2. 如何关闭 RPF（同时源进源出特性跟随关闭）：
   - 开启异步路由（一般情况下不建议，会话安全性检查将降低）：

```
config system settings
    set asymroute enable    //default = disable
end
```

   - 基于接口关闭 RPF：

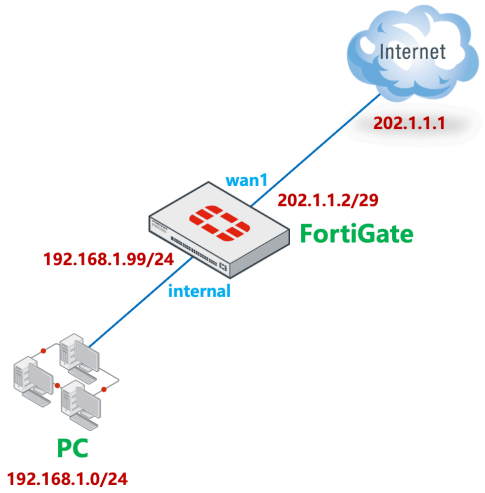
```
config system interface
    edit <interface>
        set src-check disable
end
```

静态路由

网络需求

静态路由是由系统管理员，根据网络的结构，在防火墙上手工添加路由条目。对于防火墙设备来说，静态路由是最基本的方式，也是最常用的添加路由的方式。

网络拓扑

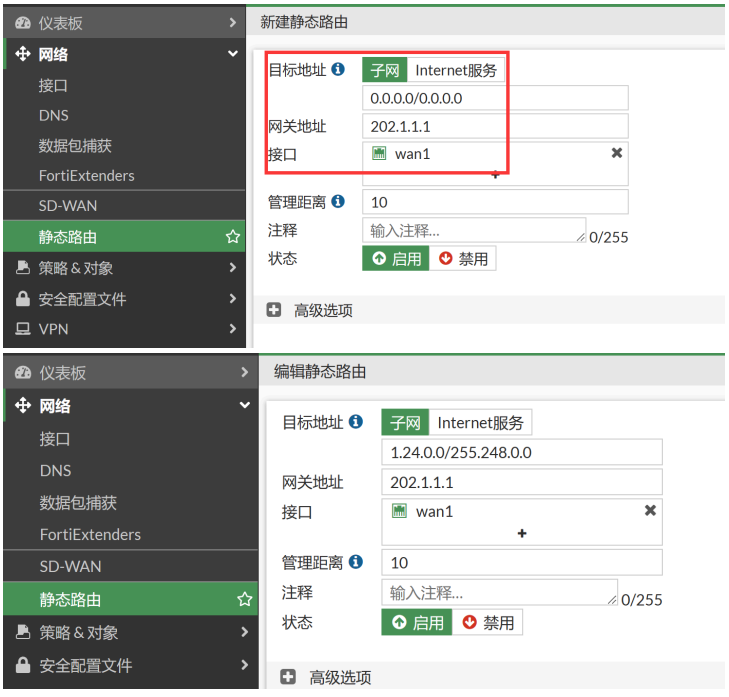


防火墙外网口 wan1 ip 地址为 202.1.1.2， 对端 ISP 路由器地址为 202.1.1.1。

配置步骤

1. 配置相关接口 IP（略）。

2. 进入网络 → 静态路由，点击 新建，按如下方式创建路由表。



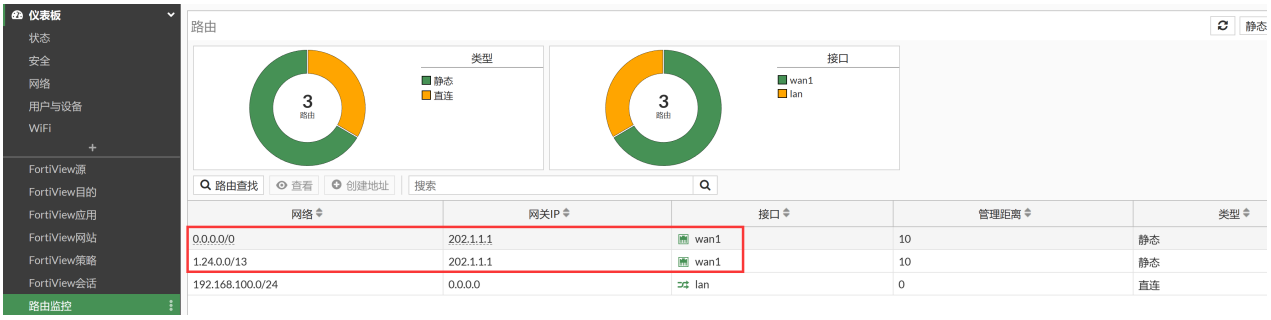
- 目的的 IP/子网掩码： 由于是默认网关，使用默认的 0.0.0.0/0.0.0.0 即可
- 设备： 该路由所关联的接口，wan1 口，必须正确填写，否则该路由无法工作
- 网关： 下一跳 ip 地址， 即 wan1 口对端运营商设备接口的 ip 地址
- 路径长度：默认 10。 对于相同的路由条目，管理距离短的会被放入路由表。如果管理距离相同，则都会被放入路由表
- 优先级：默认 0. 管理距离相同的两条路由，防火墙优先使用优先级参数较低的路由工作。

3. 对应 CLI。

```
FortiGate # config router static
FortiGate (static) # edit 1
FortiGate (1) # set gateway 202.1.1.1    //本条目没有定义dst目的网络，则默认为为0.0.0.0/0.0.0.0
FortiGate (1) # set device wan1
FortiGate (1) # next

FortiGate (static) # edit 2
FortiGate (2) # set dst 1.24.0.0 255.248.0.0
FortiGate (2) # set gateway 202.1.1.1
FortiGate (2) # set device wan1
FortiGate (2) # end
```

结果验证

1. 通过图形页面查看路由表，进入仪表盘，添加路由监控组件。



2. 或通过命令行查看该路由是否生效：get router info routing-table static。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.1.1.1, wan1, [1/0]
S       1.24.0.0/13 [10/0] via 202.1.1.1, wan1, [1/0]
C       192.168.100.0/24 is directly connected, lan
```

3. 通过 ping 202.1.1.1，确认链路正常。

策略路由

基本概念

1. 静态动态路由等，都属于目的路由，即根据目的地址进行选路。
2. 策略路由则可以根据原地址，协议类型，流控标签，目的地址等多个参数进行选路。
3. 策略路由优于静态路由，被先执行。

应用示例

1. 场景参考策略与对象 → 多线路上网配置 → 双线路不同运营商上网配置 ↗。

2. 强制 192.168.1.0/29 的地址，从 wan2 口访问互联网。

3. 进入系统管理 → 可见功能，开启高级路由功能。



4. 进入网络 → 策略路由，新建策略路由，强制 192.168.1.0/29 的地址，从 wan2 口访问互联网，该策略路由定义所有从 internal1 进入的，源地址是 192.168.1.0 255.255.255.248，目的地址是 0.0.0.0 0.0.0.0 的数据包，都会被强制由 wan2 转发，转发时下一条的网关地址为 100.1.1.1。



- 协议：协议类型，ANY 为所有任何协议，可以指定 6 tcp,17 udp,132 等
- 流入接口：流量进入接口
- 源地址掩码： 数据包的源地址
- 目的地址掩码：数据包的目的地址
- 目的端口：目的端口，默认为所有。从 1-65536
- 强制流量到：
  - 流出接口： 数据包的流出接口
  - 网关地址： 流出接口的下一条网关地址

组播

- 组播转发
- TTL=1 的组播无法转发
- 将被拒绝的组播会话加入会话表

组播转发

功能简介

在术语使用上，组播 "转发（forwarding）" 和 "路由（routing）" 这两个功能有时会让人混淆。这两个功能不应同时进行：

- 当 FortiGate 处于 NAT 模式并且需要在组播路由器和接收者之间转发组播数据包时，应启用组播转发。
- 当 FortiGate 本身作为组播路由器运行，或者具有使用组播的适用路由协议时，应启用 PIM 协议。

> **重要**
>
> 组播转发不支持 EMAC VLAN 接口。要在 EMAC VLAN 接口上使用组播，请使用 PIM 协议。

组播转发行为

组播转发的行为在 NAT 模式和透明模式下有所区别。

NAT 模式

```bash
config system settings
    set multicast-forward {enable | disable}
end
```

- `enable`（默认）：FortiGate 会将 TTL ≥ 2 的所有组播 IP 数据包转发到所有接口和 VLAN 接口（接收接口除外）。IP 报文头中的 TTL 值会减 1，必须添加组播策略以允许组播数据包通过 FortiGate。

  > **相关信息**
  >
  > 如需转发 TTL=1 的组播，请参考：路由 → 组播 → TTL=1 的组播无法转发 ↗章节。

- `disable`：不会转发任何组播报文，即使配置了放通的组播策略。

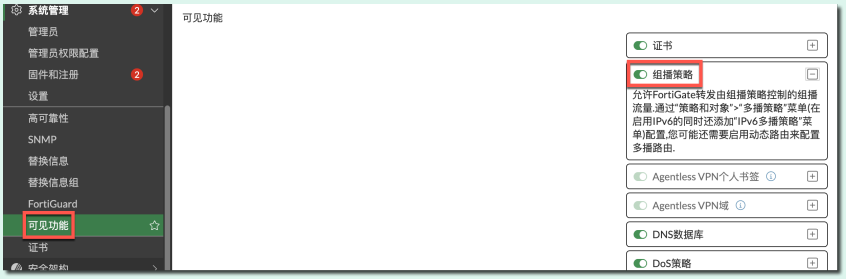
透明模式

```bash
config system settings
    set multicast-skip-policy disable
end
```

- `enable`：允许通过 FortiGate 传输组播流量，无需创建组播策略。
- `disable`（默认）：需要配置组播策略才能放通对应的组播流量（但透明模式下默认存在一条 any ↔any 的全放通组播策略，所以组播在默认配置下是放通的）。

组播策略

> **提示**
>
> IPv4 和 IPv6 组播策略可以在 GUI 中配置。进入系统管理 > 可见功能，启用组播策略和 IPv6。



```bash
config firewall multicast-policy
    edit <policy_id>
        set name "<policy_name>"
        set srcintf "<interface>"
        set dstintf "<interface>"
        set srcaddr "<address_object>"
        set dstaddr "<multicast_address>"
        set action {accept | deny}
        set status {enable | disable}
        set snat {enable | disable}          # 是否启用源 NAT
        set snat-ip <IP_address>             # SNAT 转换的 IP 地址
        set dnat <IP_address>                # 目标 NAT 地址
        set comments "<string>"
    next
end
```

- 默认配置下，无论是 NAT 模式还是透明模式，组播数据包都需要组播策略才能允许数据包从一个接口传递到另一个接口。
- 与防火墙策略类似，在组播策略中，需要指定源接口和目标接口，以及数据包源地址和目标地址允许的地址范围。还可以使用组播策略为组播数据包配置 SNAT 和 DNAT。

配置示例

FortiGate（NAT 模式）接受从 IP 地址为 192.168.100.77 的 PC 发送到目标地址范围 239.168.4.0-255 的组播数据包。该策略允许组播数据包进入 `lan` 接口，然后从 `wan1` 接口发出。当数据包离开 `wan1` 接口时，其源地址将转换为 10.10.12.2。

1. 创建源地址对象。

```bash
config firewall address
    edit "192.168.100.77/32"
        set subnet 192.168.100.77 255.255.255.255
    next
end
```

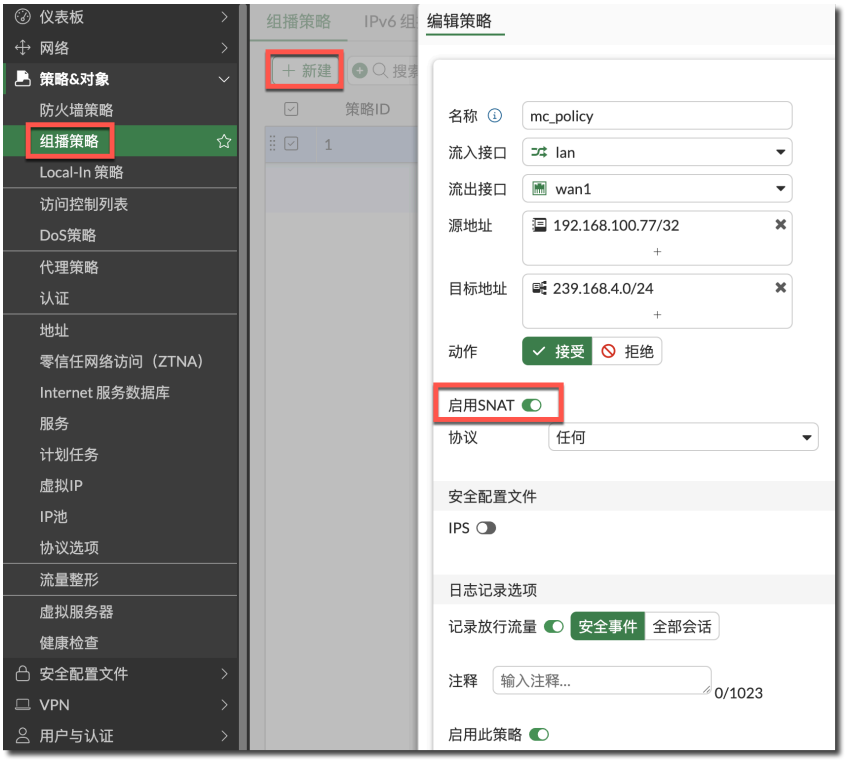
2. 创建组播目标地址对象。

```bash
config firewall multicast-address
    edit "239.168.4.0/24"
        set start-ip 239.168.4.0
        set end-ip 239.168.4.255
    next
end
```

3. 配置包含 SNAT 的组播策略。

提示

- SNAT 转换的源地址默认为出接口 wan1 的 IP。
- 如需修改 SNAT 转换的源地址，需要在 CLI 中配置。




```bash
config firewall multicast-policy
    edit 1
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "192.168.100.77/32"
        set dstaddr "239.168.4.0/24"
        set snat enable         # 启用 SNAT
        set snat-ip 10.10.12.2  # SNAT 转换后的 IP 地址
    next
end
```

多 VDOM 模式

在 FortiGate 开启多 VDOM 模式时，避免创建全通（any-to-any）的全通组播策略引起组播网络环路。

- 默认情况下，在支持 NPU 虚拟链路的型号上，将 `vdom-mode` 更改为 `multi-vdom` 时会在同一个 root VDOM 中创建一对 npu0_vlink0 和 npu0_vlink1 接口。
- 由于全通组播策略和 npu0_vlink 接口实际上是虚拟连接的，因此会形成组播网络环路。

因此，在开启多 VDOM 模式时：

1. 在将 FortiGate 切换到多 VDOM 模式之前，确保不存在全通的组播策略。

2. 如果必须定义全通组播策略，确保没有两个相连的接口（如 npu0_vlink0 和 npu0_vlink1）属于同一个 VDOM。

组播环路配置示例

> **警告**
>
> 如下配置可能会引起组播环路。

1. 启用 multi-VDOM 模式。

```bash
config system global
    set vdom-mode multi-vdom
end
```

2. 配置全放通的组播策略。

```
config firewall multicast-policy
    edit 1
        set logtraffic enable
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```
bash

3. 查看接口配置，两个 vlink 接口都在同一个 root VDOM 中。

```
show system interface
config system interface
    edit "npu0_vlink0"
        set vdom "root"    ←---两个虚拟链路接口在同一个 VDOM
        set type physical
    next
    edit "npu0_vlink1"
        set vdom "root"    ←---两个虚拟链路接口在同一个 VDOM
        set type physical
    next
end
```
bash

调试命令

- 查看组播策略组（0010000a）的 iprope 列表。

> 注
> - 格式为 `diagnose firewall iprope list <policy-group>`。
> - `<policy-group>` 的类别请参考：策略与对象 → iprope 列表 ↗章节。

```
FortiGate # diagnose firewall iprope list 0010000a

policy index=1 uuid_idx=689 action=accept
flag (8050104): f_p nat master use_src pol_stats
schedule()
cos_fwd=0  cos_rev=0
group=0010000a av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 41 → zone(1): 7
source(1): 192.168.100.77-192.168.100.77, uuid_idx=688,
dest(1): 239.168.4.0-239.168.4.255, uuid_idx=0,
service(1):
        [0:0x0:0/(0,65535)→(1,65535)] flags:0 helper:auto
nat(1): flag=0 base=192.168.100.77:0 10.10.12.222-10.10.12.222(0:0)

policy index=4294967295 uuid_idx=30 action=drop
flag (800): d_rm
schedule()
cos_fwd=0  cos_rev=0
group=0010000a av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 → zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
service(1):
        [0:0x0:0/(0,0)→(0,0)] flags:0 helper:auto
```
bash

- 查看组播策略 ID1 的命中统计。

> 注
> - 格式为 `diagnose firewall iprope show <policy-group> <policy-idx> [<policy-idx2> ...]`。
> - `<policy-group>` 的类别请参考：策略与对象 → iprope 列表 ↗章节。

```
FortiGate-101F # diagnose firewall iprope show 0010000a 1
idx:1
pkts:976636032 (3441885 8086085 13636702 8140295 8402592 12647607 6949171 7675145)
bytes:596025890544 (1029042774 3908006492 8871865919 3137288791 3349698158 7678746102 3996182994 5052661371)
asic_pkts:724249323 (0 0 0 0 0 0 0 0)
asic_bytes:537390769164 (0 0 0 0 0 0 0 0)
nturbo_pkts:4804934 (0 0 0 0 0 0 0 0)
nturbo_bytes:2351221290 (0 0 0 0 0 0 0 0)
flag:0x0
hit count:20314739 (202119 397770 465433 483860 489723 287429 311404 161334)
    first hit:2025-09-18 14:46:27 last hit:2026-02-04 11:31:45
established session count:8
    first est:2025-09-18 14:46:27 last est:2026-02-04 11:31:05
```
bash

- Debug Flow 调试组播流量（与普通单播流量一样）。

```
diagnose debug console timestamp enable
diagnose debug flow show function-name enable
diagnose debug flow filter addr <multicast_address>
diagnose debug enable
diagnose debug flow trace start 100
```
bash

TTL=1 的组播无法转发

问题现象

在默认配置（启用 `multicast-forward`）下：

```bash
config system settings
    set multicast-forward enable
end
```

- FortiGate 会丢弃 TTL（Time To Live）值为 1 的组播（Multicast）报文。
- 仅转发 TTL≥2 的组播 IP 报文（转发后变为 TTL=1）。

而 TTL=1 的组播流量在实际网络中非常常见，尤其用于链路本地（link-local）组播通信。在某些业务场景下（如特定协议或设备发现机制），需要 FortiGate 能够转发 TTL=1 的组播报文：

- 链路本地组播协议需要跨接口转发。
- 依赖 TTL=1 进行设备发现（如打印机发现）或服务通告的业务。
- 测试/实验环境中对组播 TTL 行为有明确控制需求。

解决方法

要允许 FortiGate 转发 TTL=1 的组播报文，需要启用 `multicast-ttl-notchange` 配置。启用后，FortiGate 不会修改组播报文的 TTL 值，即使 TTL=1，也允许进行组播转发。

```shell
config system settings
    set multicast-ttl-notchange enable
end
```

注意事项

1. `multicast-forward` 模式与 `multicast-routing` 不能同时启用，如果使用 `multicast-forward`，请确保 `multicast-routing` 已关闭。
2. 组播相关配置可参考官方文档：https://docs.fortinet.com/document/fortigate/7.4.99/administration-guide/999756/multicast ↗

将被拒绝的组播会话加入会话表

功能简介

可以为被拒绝的组播流量创建会话并纳入会话表中，使后续数据包能够直接匹配并丢弃，从而降低 CPU 使用率并提高性能。

```bash
config system setting
    set ses-denied-multicast-traffic {disable | enable}
end
```

- `disable`（默认）：不将被拒绝的组播会话添加到会话表中（默认，每次拒绝流量都会创建新的拒绝会话并立即删除）。
- `enable`：将被拒绝的组播会话包含在会话表中（只创建一次拒绝流量的会话并保持在会话表中）。

配置示例

在此示例中，被拒绝的组播会话被包含在 VDOM 的会话表中。创建一条拒绝（deny）组播策略，然后发送命中该策略的数据包。检查组播会话列表可以看到被拒绝的组播会话已被创建。

1. 启用 `ses-denied-multicast-traffic`。

   ```bash
   config system setting
       set ses-denied-multicast-traffic enable
   end
   ```

2. 在组播策略表中创建拒绝策略。

   ```bash
   config firewall multicast-policy
       edit 1
           set name "Deny_Multicast_Policy"
           set srcintf "port1"
           set dstintf "port3"
           set srcaddr "172.16.200.0"
           set dstaddr "230.0.0.1"
           set action deny
           set logtraffic all
           set auto-asic-offload disable
       next
   end
   ```

3. 发送数据包命中拒绝策略后，检查组播会话列表。输出中显示的第二个会话为被拒绝的组播会话：

   ```bash
   FortiGate # diagnose sys mcast-session list

   session info: id=259 vf=1 proto=17 172.16.200.55.34896→230.0.0.10.7878
   used=2 path=1 duration=8 expire=174 indev=9 pkts=4 bytes=2160
   state=00000000:
   session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_map=0×00000000
   path: log npu-deny policy=2, outdev=11, tos=0×ff

   session info: id=260 vf=1 proto=17 172.16.200.55.33488→230.0.0.1.7878
   used=2 path=0 duration=6 expire=177 indev=9 pkts=5 bytes=2700
   state=00000200: deny
   session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_map=0×00000000
   Total 2 sessions
   ```

4. 可以看到第二个会话的 `state` 字段标记为 `deny`，表示该组播会话已被策略拒绝并记录在会话表中，后续被拒绝的流量直接匹配该会话被丢弃，不需要重新建立拒绝会话。

路由常见问题

- 黑洞路由不生效
- 策略路由不生效

黑洞路由不生效

问题现象

通常会在 IPSec VPN 环境下配置高管理距离（distance）的黑洞路由，避免 IPSec 连接中断时，VPN 流量匹配默认路由被送到 Internet。

1. 某用户在配置 VPN 路由对应的黑洞路由，如下所示，distance 配置为 255。

```bash
config router static
    edit 15
        set status disable
        set dst 192.168.123.0 255.255.255.0
        set distance 20
        set device "VPN_Tunnel1"
    next
    edit 16
        set dst 192.168.123.0 255.255.255.0
        set distance 255
        set blackhole enable
        set vrf 0
    next
end
```

2. 查看路由表中没有该黑洞路由，VPN 隧道中断后，流量仍然匹配默认路由从 wan1 接口转发。

```bash
FortiGate # get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.22.6.1, wan1, [1/0]
```

3. 但在路由 Database 中可以看到该黑洞路由。

```bash
FortiGate # get router info routing-table database | grep 192.168.123.0
S       192.168.123.0/24 [20/0] via VPN_Tunnel1 tunnel 10.0.0.2, [1/0]
S       192.168.123.0/24 [255/0] is a summary, Null, [1/0]
```

问题原因

路由管理距离（distance）255 表示该路由无效。路由只能在 routing-table database 中可见。不会插入到路由转发表中，所以 VPN 流量不会按预期丢弃（路由），而是匹配默认路由从 wan1 接口转发。

解决方法

任何静态路由的管理距离都需要 ≤ 254。否则，路由条目将被视为无效路由。

```bash
config router static
    edit 16
        set dst 192.168.123.0 255.255.255.0
        set distance 254
        set blackhole enable
        set vrf 0
    next
end
```

策略路由不生效

问题现象

1. 配置的策略路由如下。

```
config router policy
    edit 1
        set input-device "lan"
        set src "192.168.100.77/255.255.255.255"
        set dst "223.5.5.5/255.255.255.255"
        set gateway 10.10.12.1
        set output-device "wan1"
    next
end
```

bash

2. 实际使用中该策略路由不生效，查看策略路由列表（`diagnose firewall proute list`），该条策略路由有 `disable` 标记，但该策略路由配置中未 disable。



问题原因

1. 可能是由于配置了 link-monitor 或 SD-WAN 的健康检查功能，默认配置下 `update-policy-route` 为开启状态，当 link-monitor 的检查状态由 UP 变为 DOWN 后，会将相应出接口的策略路由禁用。

```
config system link-monitor
    edit "test"
        set srcintf "wan1"
        set server "10.10.12.1"
        set update-cascade-interface enable    //默认开启//
        set update-static-route enable    //默认开启//
        set update-policy-route enable    //默认开启//
    next
end
```

bash

2. 查看 link-monitor 的状态为 DOWN，导致对应出接口的策略路由为 disable 状态。

```
FW1_FGT101F # diagnose sys link-monitor status

Link Monitor: test, Status: dead, Server num(1), HA state: local(dead), shared(dead)
Flags=0x9 init log_downgateway, Create time: Wed Apr 24 16:19:13 2024
Source interface: wan1 (7)
VRF: 0
Interval: 500 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
  Peer: 10.10.12.1(10.10.12.1)
        Source IP(10.10.12.2)
        Route: 10.10.12.2->10.10.12.1/32, gwy(10.10.12.2)
        protocol: ping, state: dead
                Packet lost: 100.000%
                MOS: 4.350
                Number of out-of-sequence packets: 0
                Recovery times(0/5) Fail Times(1/5)
                Packet sent: 1231, received: 287, Sequence(sent/rcvd/exp): 1232/475/476
```

bash

解决方法

1. 关闭 link-monitor 中的 `update-policy-route`。

```bash
config system link-monitor
    edit "test"
        set update-policy-route disable
    next
end
```

2. 再次查看对应出接口的策略路由状态，disable 标记消失，流量可以正确匹配该策略路由。

```bash
FW1_FGT101F # diagnose firewall proute list
list route policy info(vf=root):

id=1(0x01) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-0 iif=48(lan) dport=0-65535 path(1)
oif=7(wan1) gwy=10.10.12.1
source wildcard(1): 192.168.100.77/255.255.255.255
destination wildcard(1): 223.5.5.5/255.255.255.255
hit_count=1 last_used=2024-04-24 16:32:24
```

动态路由

RIP

OSPF

- OSPF 基础配置
- OSPF GR

BGP

- BGP 基础配置
- BGP 计时器
- BGP GR
- BGP 路由过滤
- BGP 路由刷新
- BGP 下一跳递归
- 路由反射器
- BGP 多路径
- additional-path
- 附加 AS-Path
- BGP Debug
- BGP 常见问题

  清除 BGP 路由条目　路由震荡　使用 Loopback 接口建立 BGP　BGP 与静态路由的优先顺序　关闭 Leftmost AS 检查

  BGP 选路忽略 AS-PATH　多个本地 AS 的配置和通告　RR 反射路由无法修改属性

路由对象

- Route Map
- Access List
- Prefix List
- AS Path List
- Community List

动态路由协议排错

RIP

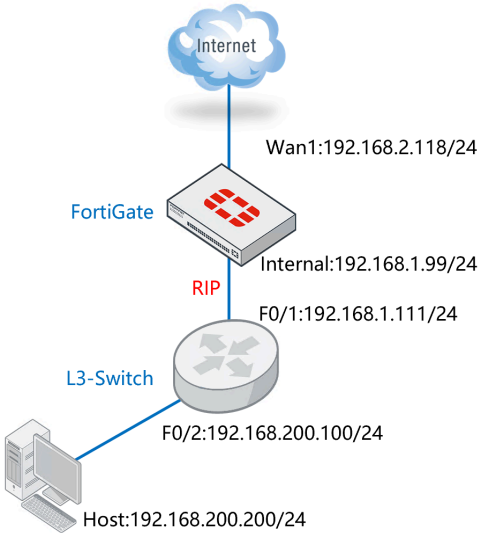
应用场景

1. 当网络路由设备较多，且不超过 16 个，为避免写大量静态路由，且路由可自动学习，可在 FortiGate 上配置 RIP 路由协议，让 FortiGate 也能动态学习到其他网络的路由，且能自动老化更新。
2. 若网络中路由设备超过 16 个，建议使用 OSPF，它的路由学习更新速度更快，更适合 16 个路由设备以上的网络。
3. 当网络中路由设备较少，建议使用静态路由即可，维护简单，且对路由器要求不高，因为路由器都支持静态路由，而普通小路由一般不支持 RIP 协议。

网络需求

1. 内网三层交换机和出口 FortiGate 间通过动态路由 RIP 互相通告路由，实现内网用户正常上网。
2. FortiGate 手动配置默认路由，并将其重发布引入 RIP，三层交换机与 FortiGate 之间通过 RIP 互相学习路由，实现内网用户正常上网。

网络拓扑

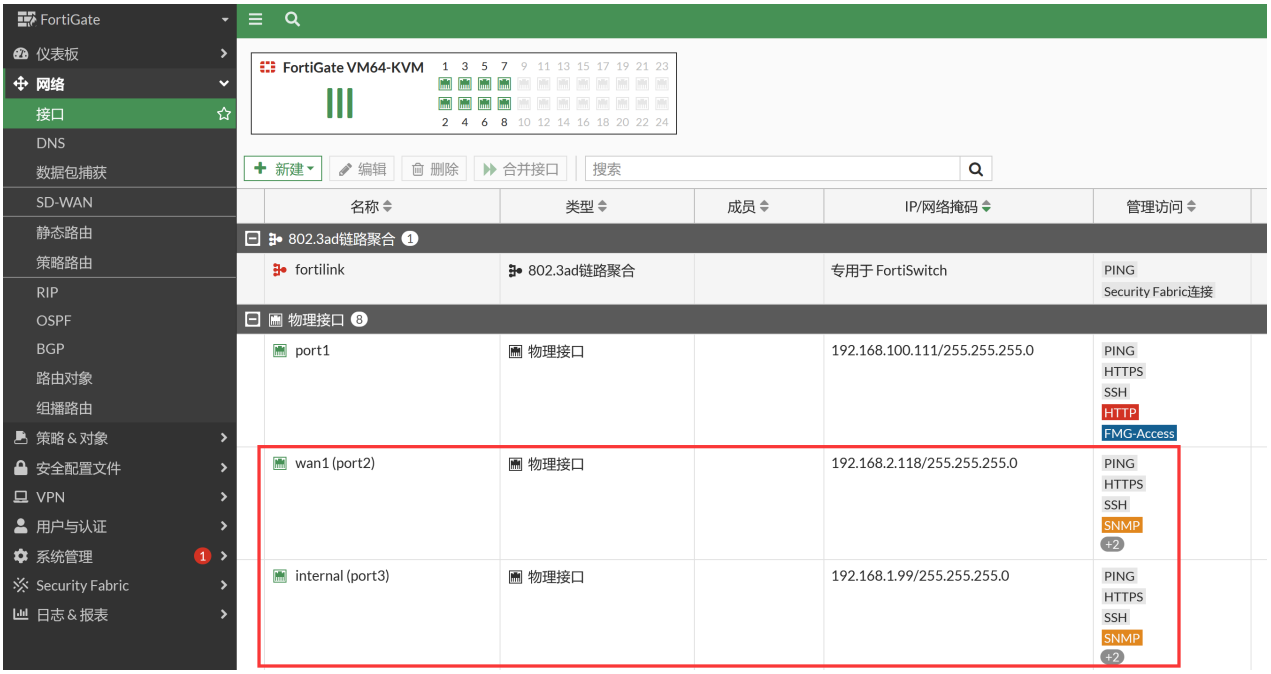


配置要点

- 配置接口地址
- 配置防火墙
- 配置路由器

配置步骤

1. 配置 FortiGate 接口 IP。



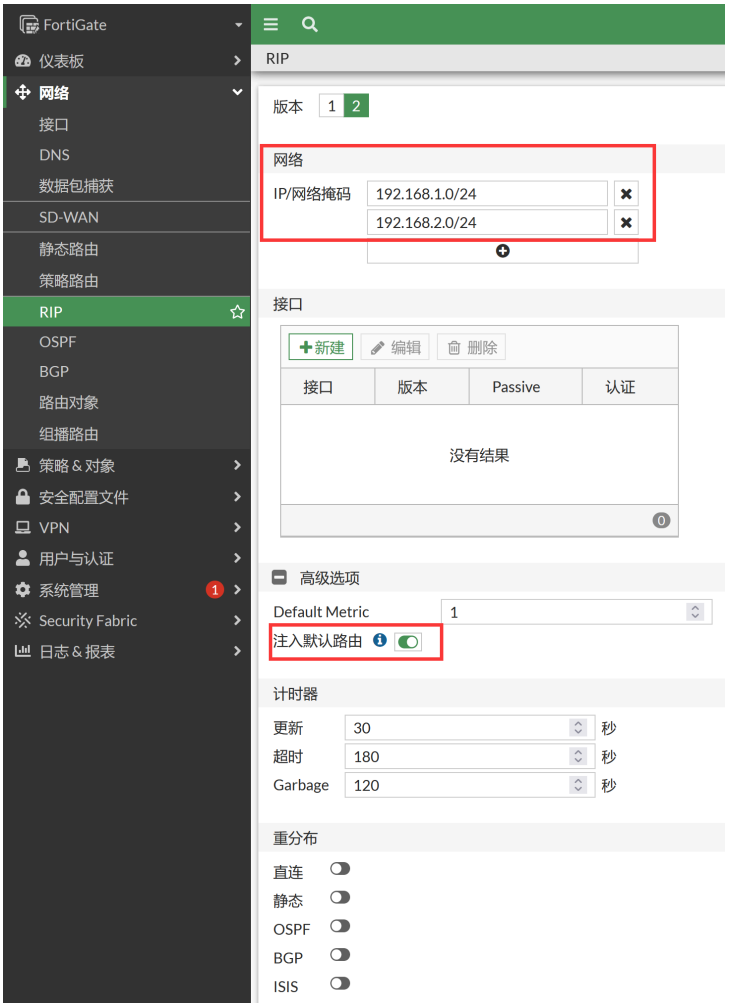
2. 配置默认路由。



3. 配置 RIP，需要进入系统管理 → 可见功能中打开高级路由。



4. 进入网络 → RIP，配置基本信息：选择版本 2，添加 RIP 网络，启用产生缺省路由，向邻居（路由器）发送缺省路由（路由分发：是否对其他协议的路由进行分发）。




```
config router rip
    set default-information-originate enable
    config network
        edit 1
            set prefix 192.168.1.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.2.0 255.255.255.0
        next
    end
end
```

5. 配置路由器。

```
interface Ethernet0/1
 ip address 192.168.1.111 255.255.255.0
 duplex auto
!
interface Ethernet0/2
 ip address 192.168.200.100 255.255.255.0
!
router rip
version 2
network 192.168.1.0
network 192.168.10.0
no auto-summary
```

结果验证

1. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 192.168.100.99, port1, [1/0]
C       192.168.1.0/24 is directly connected, port3
C       192.168.2.0/24 is directly connected, port2
C       192.168.100.0/24 is directly connected, port1
R       192.168.200.0/24 [120/2] via 192.168.1.111, port3, 00:01:18

FortiGate # get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
Rs 0.0.0.0/0                              1                    --
Rc 192.168.1.0/24                         1                   port3
Rc 192.168.2.0/24                         1                   port2
R  192.168.200.0/24   192.168.1.111       2 192.168.1.111     port3  02:18
```

2. 查看思科路由器的路由表。

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 192.168.1.99 to network 0.0.0.0

R*    0.0.0.0/0 [120/1] via 192.168.1.99, 00:00:07, Ethernet0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/1
L        192.168.1.111/32 is directly connected, Ethernet0/1
R     192.168.2.0/24 [120/1] via 192.168.1.99, 00:00:07, Ethernet0/1
      192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.200.0/24 is directly connected, Ethernet0/2
L        192.168.200.100/32 is directly connected, Ethernet0/2
```

动态路由协议排错

OSPF 邻居无法建立

OSPF邻居建立不起来原因？

最常见的是如下几种情况：
1. OSPF网络类型是NBMA的，但忘记在OSPF协议模式下配置Peer邻居了；
2. OSPF邻居的Hello及Dead Interval值不一致；
3. 在Stub或NSSA区域，有些FortiGate没有配置成Stub或NSSA；
4. OSPF验证配置错误；
5. OSPF Router ID有问题，可能和某个其他设备一样了；
6. OSPF链路两端的网络类型不一致；
7. OSPF链路两端的MTU相差比较大，尤其注意和不同厂商实现互通时（需要在其接口下配置OSPF忽略MTU检查或修改MTU）；
8. 该网络根本就没有启动OSPF；
9. 区域号不一致；链路的网络地址不一致，注意检查两边的mask；

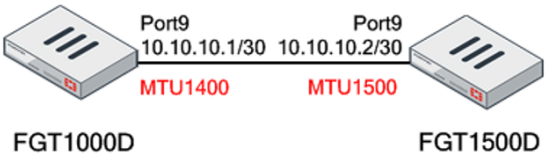
有什么好的办法知道OSPF出了什么问题？

调试开关是需要打开的，它能让你对OSPF的大部分问题看的一目了然

```
#diagnose ip router ospf  level warn
#diagnose ip router ospf  all  enable
#diagnose debug enable
```

OSPF不能建立邻接关系

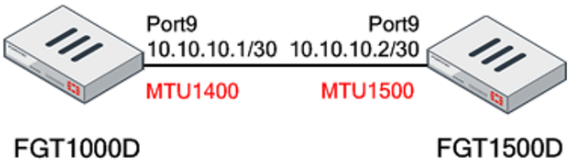
症状：OSPF邻居关系卡在Exstart/Exchange状态




```
FG1000D # get router info ospf neighbor
OSPF process 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.10.2        1   ExStart/Backup  00:00:36    10.10.10.2      port9

FG1000D # get router info ospf interface
port9 is up, line protocol is up
Internet Address 10.10.10.1/30, Area 0.0.0.0, MTU 1400
Process ID 0, Router ID 10.10.10.1, Network Type BROADCAST, Cost: 1

FG1500D # get router info ospf interface
port9 is up, line protocol is up
Internet Address 10.10.10.2/30, Area 0.0.0.0, MTU 1500
Process ID 0, Router ID 10.10.10.2, Network Type BROADCAST, Cost: 1
```

```
FG1000D # diagnose ip router ospf  level warn
FG1000D # diagnose ip router ospf all enable
FG1000D # diagnose  debug enable
FG1000D # id=20301 logdesc="Routing log" msg="OSPF: RECV[DD]: From
10.10.10.2 via port9:10.10.10.1: MTU size is too large (1500)"     //开启调试
```

OSPF不能建立邻接关系
解决办法



解决办法一：修改接口MTU为一致

```
FG1000D # config system interface
FG1000D (interface) # edit port9
FG1000D (port9) # set mtu 1500
FG1000D (port9) # end
//或者//
config router ospf
          config ospf-interface
            edit "port9"
                set interface "port9"
                set mtu 1500
                end
```

解决办法二：在OSPF-interface配置mtu-ignore

```
config router ospf
          config ospf-interface
            edit "port9"
                set interface "port9"
                set mtu-ignore enable
            next
        end
```

```
FG1000D # get router info ospf neighbor
OSPF process 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.10.2        1   Full/DR         00:00:37    10.10.10.2      port9

FG1500D # get router info ospf  neighbor          //邻居关系变为Full
OSPF process 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.10.1        1   Full/Backup     00:00:38    10.10.10.1      port9
```

BGP 邻居无法建立

有哪些原因会导致BGP连接建立不起来？

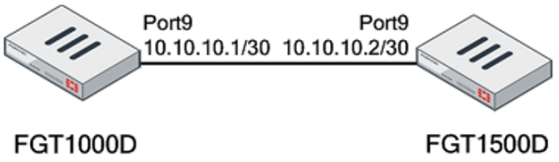
最常见原因如下：

1) 两边BGP peer地址不可达，一般是底层原因或者缺少可达的路由，可以使用扩展的ping命令检查TCP/IP(TCP 179)连接是否正常,由于一台路由器可能有多个接口能够到达对端，应使用execute ping-options source x.x.x.x 命令指定发送ping包的源IP地址
2) 对等体IP地址和AS配置错误，常为配置错误；
3) OPEN报文协商失败，OPEN报文需要协商BGP版本、Holdtime、RouterID以及可选项参数（包括各种能力参数）等；
4) BGP的MD5验证配置错误；
5) BGP的RouterID冲突；

有哪些排错手段针对BGP连接建立不起来的情况？

1) 首先打开调试开关，确认状态机在哪一步出现错误；
   diagnose ip router bgp level warn 或 info
   diagnose ip router bgp all enable
   diagnose debug enable
2) 如果BGP状态始终在active状态徘徊，表示TCP建立不起来，首先排除底层不通和路由不可达的情况；
3) 如果BGP状态始终在active状态徘徊，其次排除BGP的MD5验证问题；
4) 如果BGP状态始终在active状态徘徊，最后特殊配置问题，比如EBGP peer的ebgp-enforce-multihop；
5) 如果是Open报文协商错误，通过调试开关可以很方便的查看到具体的错误类别和信息，然后根据错误提示采取具体措施；

BGP不能建立邻接关系
症状：邻居无法起来




```
FG1000D # get router info bgp neighbors
BGP neighbor is 10.10.10.2, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle

FG1500D # get router info bgp neighbors
BGP neighbor is 10.10.10.1, remote AS 100, local AS 200, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
```

```
FG1000D # diagnose ip router bgp level warn
FG1000D # diagnose ip router bgp all  enable
FG1000D # diagnose  debug enable
FG1000D # id=20301 logdesc="Routing log" msg="BGP: 10.10.10.2-Outgoing [DECODE] Open: Invalid Router ID
1.1.1.1"  //通过diagnose发现，Router-id存在异常
id=20301 logdesc="Routing log" msg="BGP: %BGP-3-NOTIFICATION: sending to 10.10.10.2 2/3 (OPEN Message
Error/Bad BGP Identifier.) 4 data-bytes [01 01 01 01]"
```

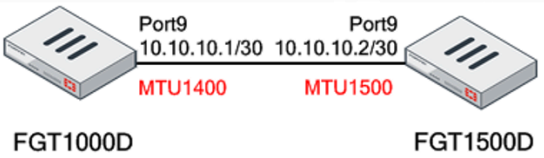
重要

如果 BGP 邻居较多，可以使用如下 CLI 命令来过滤特定 BGP 邻居的 Debug 信息：

```
diagnose ip router bgp set-filter neighbor x.x.x.x
```
bash

BGP不能建立邻接关系

解决办法：通过diagnose_debug得知是Router-ID错误，检查BGP配置，修改Router-ID



检查配置：

```
FG1000D # show router bgp
config router bgp
    set as 100
    set router-id 1.1.1.1
        config neighbor
            edit "10.10.10.2"
                set remote-as 200
```

检查配置：

```
FG1500D # show router bgp
config router bgp
    set as 200
    set router-id 1.1.1.1 //冲突的R-ID
        config neighbor
            edit "10.10.10.1"
                set remote-as 100
```

解决办法：修改Router-ID不冲突即可

```
FG1500D # config router bgp
FG1500D (bgp) # set router-id 1.1.1.2
FG1500D (bgp) # end
```

修改结果：

```
FG1000D # get router info bgp neighbors
BGP neighbor is 10.10.10.2, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 1.1.1.2
  BGP state = Established, up for 00:00:12

FG1500D # get router  info bgp neighbors
BGP neighbor is 10.10.10.1, remote AS 100, local AS 200, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:00:29
```

重启路由进程

```
execute router clear ospf process
execute router clear bgp all (in/out/soft/prefix-filter)
```
bash

OSPF

- OSPF 基础配置
- OSPF GR

OSPF 基础配置

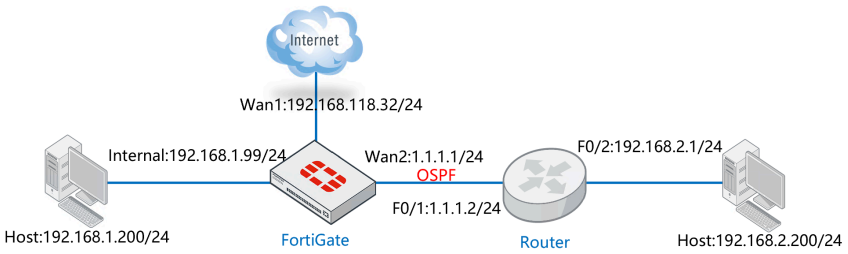
应用场景

1. 若网络中路由设备超过 16 个，建议使用 OSPF，它的路由学习更新速度更快，更适合 16 个路由设备以上的网络。
2. 当网络路由设备较多，且不超过 16 个，为避免写大量静态路由，且路由可自动学习，可在 FortiGate 上配置 RIP 路由协议功能，让 FortiGate 也能动态学习到其他网络的路由， 且能自动老化更新。
3. 当网络中路由设备较少，建议使用静态路由即可，维护简单，且对路由器要求不高，因为路由器都支持静态路由，而普通小路由一般不支持 RIP 协议。

网络需求

1. 内网三层交换机和出口 FortiGate 间通过动态路由 OSPF 互相通告路由，实现内网用户正常上网。
2. FortiGate 手动配置默认路由，并将其重发布引入 OSPF，三层交换机与 FortiGate 之间通过 OSPF 互相学习路由，实现内网用户正常上网。

网络拓扑

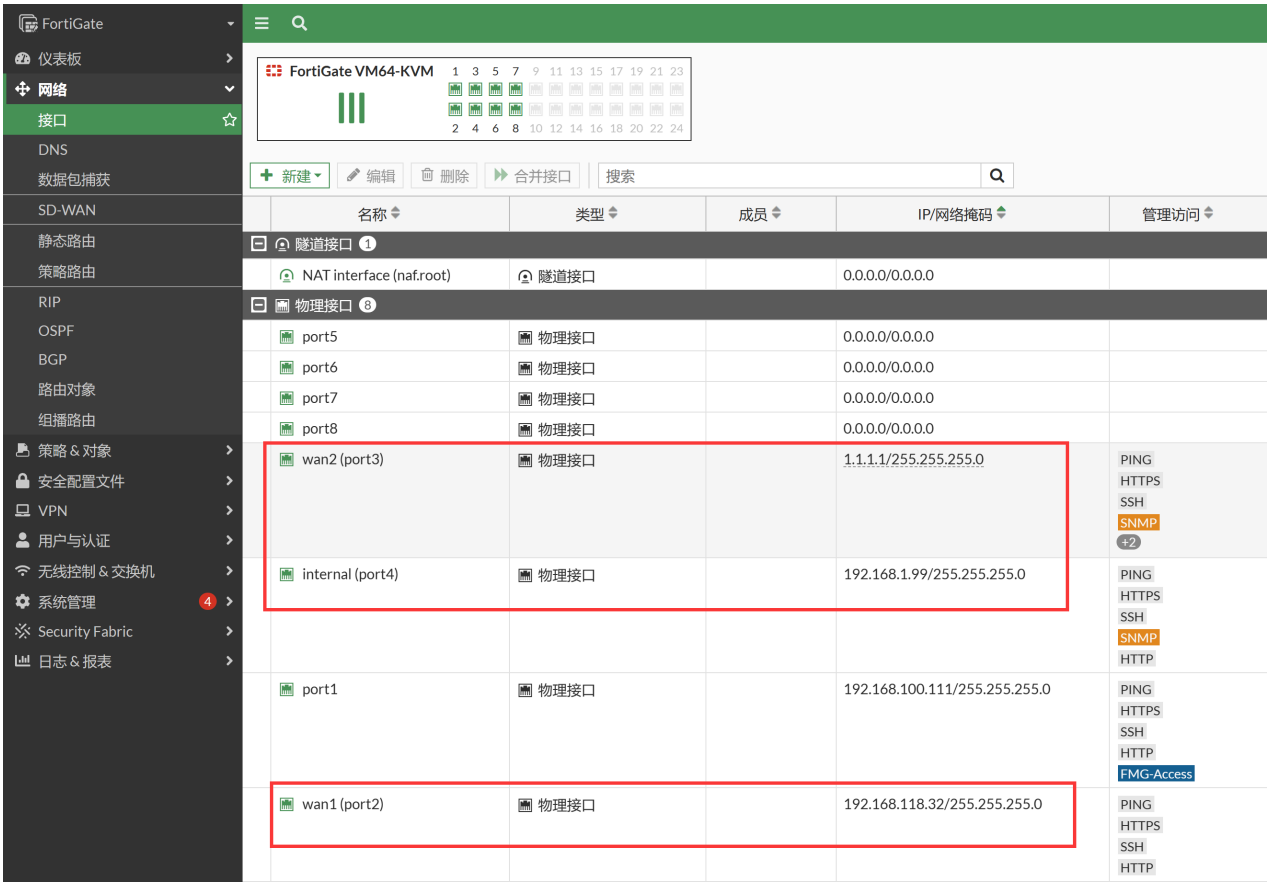


配置要点

- 配置接口地址
- 配置默认路由
- 配置 OSPF
- 配置路由器 ID
- 发布缺省路由
- 重发布
- 建立 OSPF 区
- 添加 OSPF 网络
- 添加接口
- 配置对端路由器

配置步骤

1. 配置接口地址。



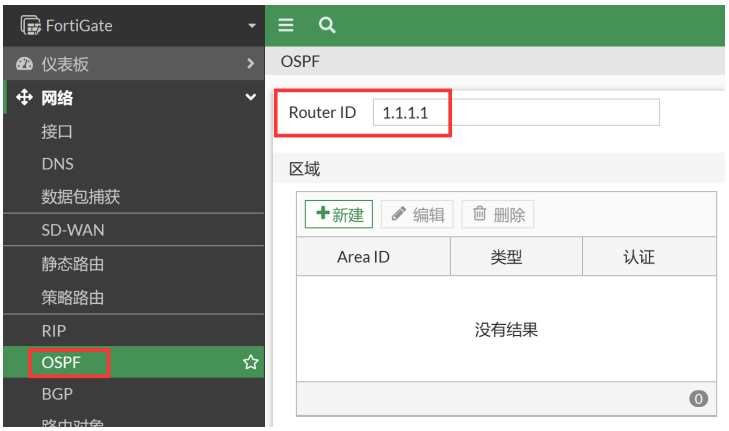
2. 配置默认路由。



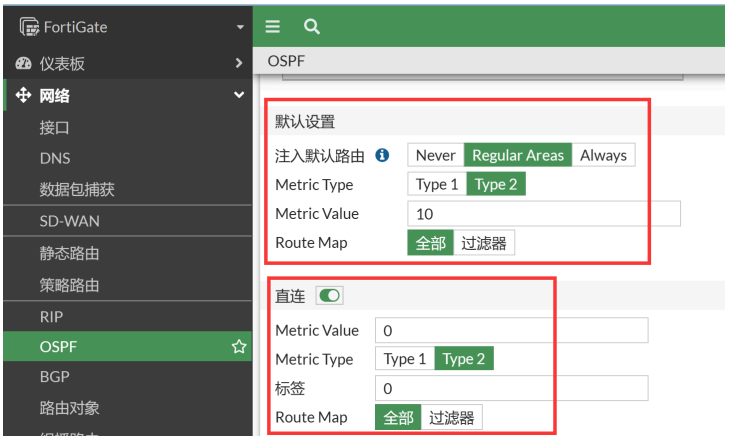
3. 配置 OSPF，首先需要在系统配置 → 可见功能中，开启高级路由功能。



4. 进入网络 → OSPF 页面，配置 Router ID。

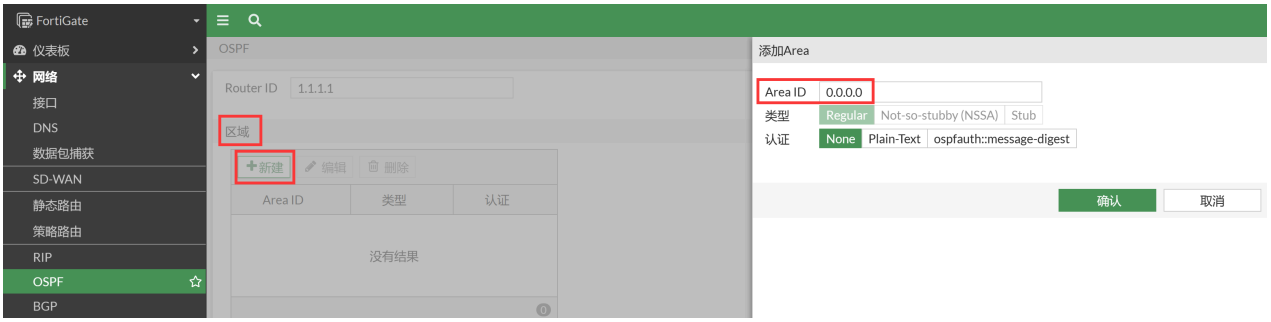


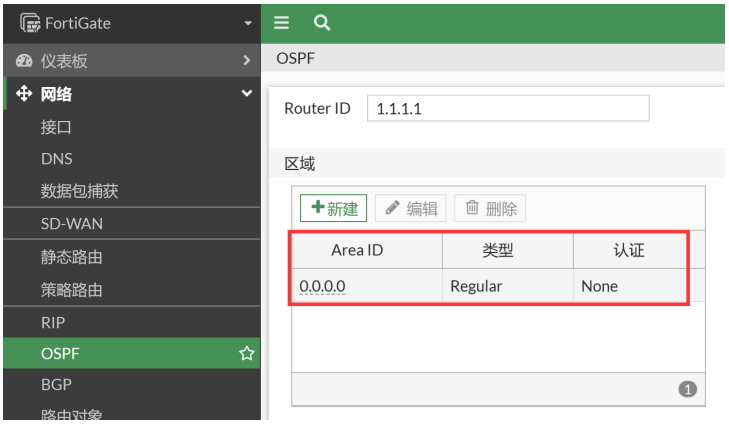
5. 开启默认路由注入，并开启重发布直连路由。



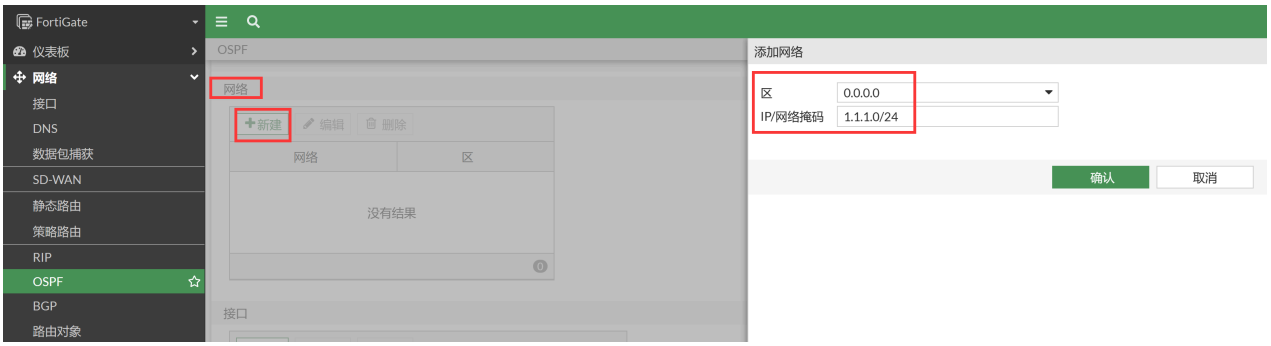
- 配置 Router ID，填写 ID 为 1.1.1.1
- 注入缺省路由：选择 "Regular Areas"，其三个选项意义如下：
  - Never：从不发送缺省路由表
  - Regular Areas：如果有则发布，没有则不发布
  - Always：无论是否有缺省路由表，都发布一条缺省路由出去
- 重发布：勾选 "直连"，将直连的 192.168.1.0/24 路由信息发送给 OSPF 邻居

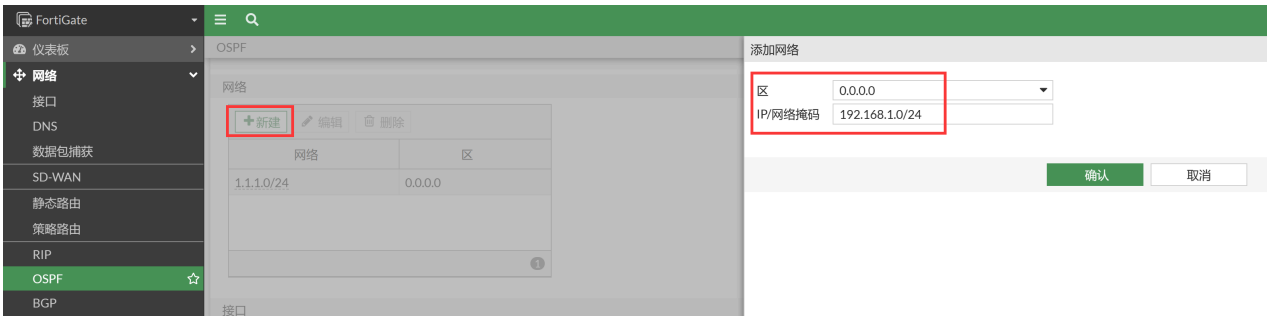
6. 建立 OSPF 区域，在区域部分点击新建按钮，创建根区域 0.0.0.0（area 0）。

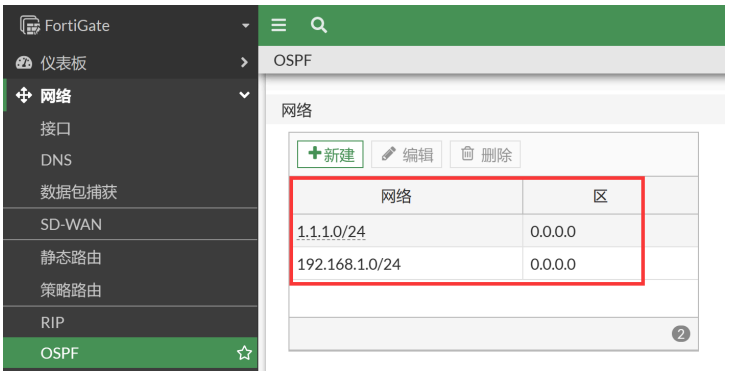




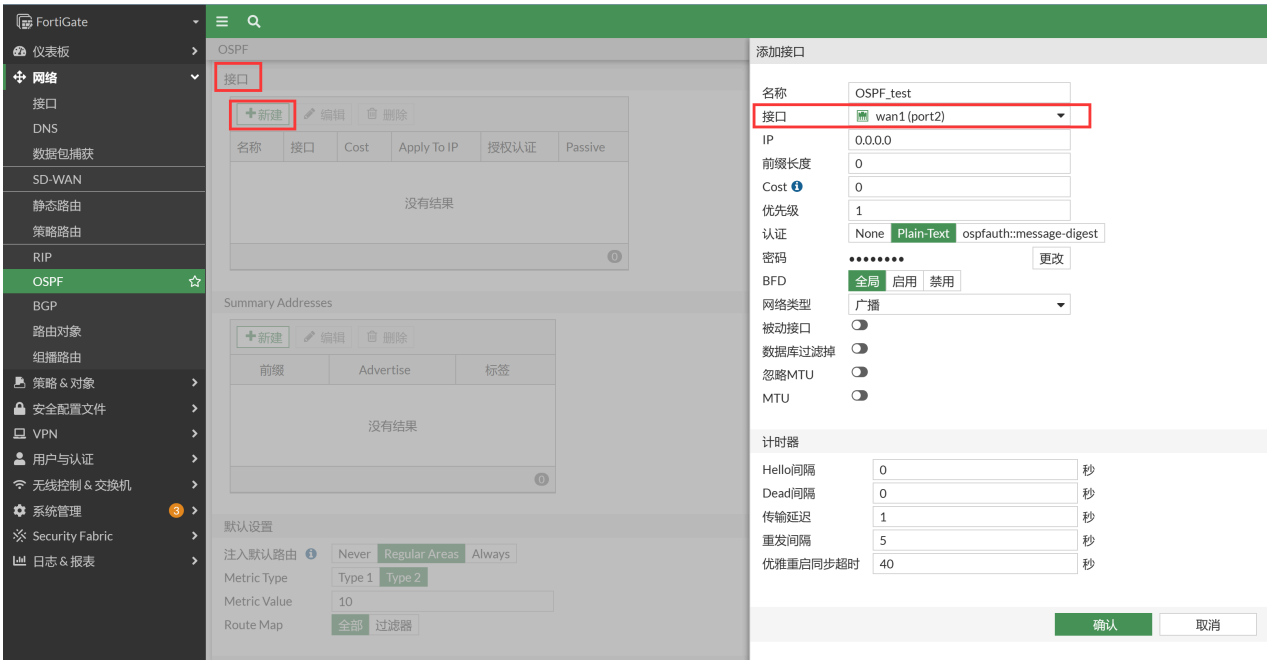
7. 添加 OSPF 网络，在网络部分点击新建按钮，将 1.1.1.0/24 和 192.168.1.0/24 网络加入 OSPF 区域 0.0.0.0，之后点击页面下方的应用按钮应用配置。







8. 添加 OSPF 接口（可选，可以自定义 OSPF 的协商参数），在接口部分点击新建按钮，配置相关的协商参数。



- 名称： 标识作用
- 接口：需要编辑的接口
- 认证： 是否需要对该接口进行 OSPF 的认证，支持 Plain-Text（明文），none（无），几种方式
- Cost：接口 OSPF cost 值，用于 SPF 的计算，值越小代表越可用于优先转发
- 密码： 输入 key 的秘钥
- 计时：
  - Hello 间隔： hello 包的发送间隔默认为 10 秒，根据需要进行修改，对 OSPf 邻居协商时，该参数需要相同
  - Dead 间隔： 默认为 40 秒，根据需要进行修改，对 OSPf 邻居协商时，该参数需要相同

9. 对应的 CLI。

```
config router ospf
    set default-information-originate enable
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 1.1.1.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.1.0 255.255.255.0
        next
    end
    config redistribute "connected"
        set status enable
    end
    config redistribute "static"
    end
    config redistribute "rip"
    end
    config redistribute "bgp"
    end
    config redistribute "isis"
    end
end
```

.0. 配置路由器。

```
配置接口IP:
interface e 0/1
ip address 1.1.1.2 255.255.255.0
interface e 0/2
ip address 192.168.2.1 255.255.255.0
```

```
配置OSPF:
router ospf 10
network 1.1.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0    //该条目也可以采用直连分发的方式发布
```

结果验证

1. 查看 FortiGate 的路由表，可以看到 FortiGate 学到了路由器发布的路由 192.168.2.0/24。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 192.168.118.1, port2, [1/0]
C       1.1.1.0/24 is directly connected, port3
C       192.168.1.0/24 is directly connected, port4
O       192.168.2.0/24 [110/11] via 1.1.1.2, port3, 00:00:33
C       192.168.100.0/24 is directly connected, port1
C       192.168.118.0/24 is directly connected, port2
```

2. 查看路由器的路由表，可以看到路由器学到了 FortiGate 重发布的两个直连网段 192.168.1.0/24、192.168.118.0/24，以及 FortiGate 注入的默认路由。

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 1.1.1.1 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/10] via 1.1.1.1, 00:00:49, Ethernet0/1
      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        1.1.1.0/24 is directly connected, Ethernet0/1
L        1.1.1.2/32 is directly connected, Ethernet0/1
O     192.168.1.0/24 [110/11] via 1.1.1.1, 00:00:49, Ethernet0/1
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Ethernet0/2
L        192.168.2.1/32 is directly connected, Ethernet0/2
O E2  192.168.118.0/24 [110/10] via 1.1.1.1, 00:00:49, Ethernet0/1
```

注意事项

route-ttl

1. 在 HA 主备模式部署环境下，主机会将自己的路由转发表项（FIB）同步给备机。
2. 默认配置下，当一个备机被选举成主机后，其会保留从原主机同步来的 FIB 表项 10s，同时动态路由进程会发起邻居建立过程，重新学习相关的动态路由。HA 切换 10s 后，从原主机同步来的 FIB 表项会被删除，如果在这 10s 内动态路由邻居没有建立，无法学习到动态路由表项，会造成 HA 切换后业务中断的情况。
3. 建议 HA 主备模式 + 动态路由的部署模式下，在 HA 配置中配置 route-ttl 的时间可以覆盖新主机建立动态路由邻居与学习路由的时间，如下所示。

```
config system ha
    set route-ttl 120
end
```

graceful-restart

1. 在 HA 主备模式 + 动态路由的部署环境下，默认配置下，当 HA 切换后（或动态路由进程重启），上下游的动态路由邻居设备会删除动态路由邻居，并删除从 FortiGate 学习到的动态路由，这也会造成 HA 切换后业务的中断。
2. 解决方法是开启动态路由的 graceful-restart 功能（并配置 restart-period），FortiGate 与上下游动态路由邻居设备都需要开启 GR 功能。

```
config router ospf
    set restart-mode graceful-restart
    set restart-period 120
end
```

3. 开启 GR 后，FortiGate 发生 HA 主备切换后，上下游的动态路由邻居设备会自动开启 GR Helper 模式，在此模式下，上下游的动态路由邻居不会删除 FortiGate HA 切换前与原主机的 OSPF 邻居关系与其学习到的动态路由，直到 FortiGate 的新主机与上下游邻居重新建立动态路由邻居或 FortiGate 配置的 restart-period（默认为 120s）到期。

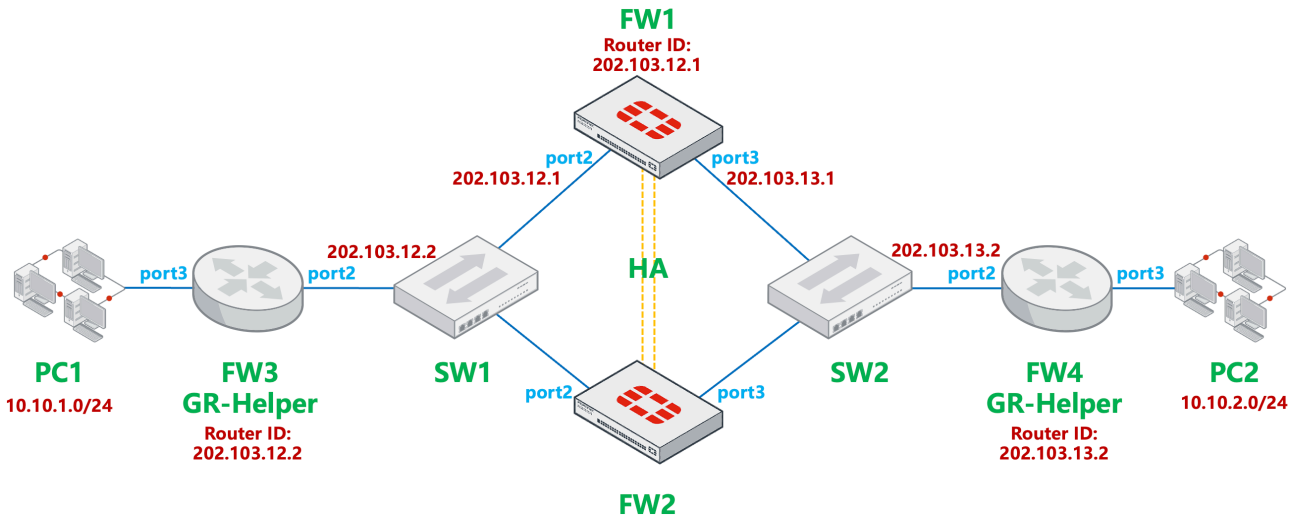
OSPF GR

网络需求

FortiGate HA 集群与上下游路由器运行 OSPF 协议，并开启 GR。在 FortiGate 的 HA 发生主备切换时，GR 需要确保每个路由器（包括 FortiGate 及其它路由器）在 HA 切换期间保持其路由表中的 OSPF 路由（GR Helper），以避免流量中断。

在 FortiGate HA 集群中，OSPF 进程相关工作仅在主设备上运行。当发生 HA 切换时，将在新选举的主机上启动一个新的 OSPF 协商过程，在此过程中，新的主机需要在路由 Kernel 表中维持从原主机学习到的 OSPF 路由，直至新的 OSPF 邻居形成。

网络拓扑



1. FW1 与 FW2 建立 A - P 模式的 HA。
2. FW1/FW2 与 FW3、FW4 分别建立 OSPF 邻居，并宣告各自的网段。
3. FW1/FW2 与 FW3、FW4 开启 OSPF Graceful Restart，FW1/FW2 发生主备切换后，FW3、FW4 可以进入 GR Helper 模式，协助 FW1/FW2 的 GR 过程。

配置步骤

1. 基础网络配置（略）。

2. FW1 与 FW2 的 HA 配置（略）。

3. 安全策略配置（略）。

4. 配置 FW1/FW2 的 OSPF，宣告网络，开启 GR 功能，restart-period 设置为 600s。

```bash
config router ospf
    set router-id 202.103.12.1
    set restart-mode graceful-restart
    set restart-period 600
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 202.103.12.0 255.255.255.0
        next
        edit 2
            set prefix 202.103.13.0 255.255.255.0
        next
    end
end
```

5. 配置 FW3 的 OSPF，宣告网络，路由器作为 GR-Helper 也需要开启 GR 功能。

```
config router ospf
    set router-id 202.103.12.2
    set restart-mode graceful-restart
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 2
            set prefix 202.103.12.0 255.255.255.0
        next
        edit 3
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```
bash

6. 配置 FW4 的 OSPF，路由器作为 GR-Helper 也需要开启 GR 功能。

```
config router ospf
    set router-id 202.103.13.2
    set restart-mode graceful-restart
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 202.103.13.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```
bash

7. 配置 FW1/FW2 HA 配置中的 route-ttl 为 600s（默认 10s），在此时间内，新的主机需要在路由 Kernel 表中维持从原主机学习到的 OSPF 路由，直至新的 OSPF 邻居形成。

> **重要**
>
> 重要步骤，防止 HA 切换后，新主机建立新的 OSPF 邻居前，路由消失导致业务中断。

```
config system ha
    set route-ttl 600
end
```
bash

结果验证

1. 初始状态下，FW1 为主设备，FW2 为备设备。

```
FW1 # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:42869 b:39035389
        traffic.total = s:0 p:42867 b:39033889
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=7 Debug=0
nvcluster=1, ses_pickup=0, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM23000175:      Primary, serialno_prio=1, usr_priority=128, hostname=FW1
FGVM08TM23000176:    Secondary, serialno_prio=0, usr_priority=100, hostname=FW2
```
bash

```
[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0
FGVM08TM23000175:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM23000176:    Secondary, ha_prio/o_ha_prio=1/1
```

2. 初始状态下查看 FW3 的 OSPF 邻居状态与路由信息，邻居状态为 Full，OSPF 路由学习正常，未进入 GR Helper 状态。

```bash
FW3 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
202.103.12.1      1   Full/DR         00:00:34    202.103.12.1    port2

FW3 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.2.0/24 [110/3] via 202.103.12.1, port2, 02:47:00, [1/0]
O       202.103.13.0/24 [110/2] via 202.103.12.1, port2, 02:47:00, [1/0]
```

3. 初始状态下查看 FW4 的 OSPF 邻居状态与路由信息，邻居状态为 Full，OSPF 路由学习正常，未进入 GR Helper 状态。

```bash
FW4 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
202.103.12.1      1   Full/Backup     00:00:33    202.103.13.1    port2

FW4 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.1.0/24 [110/3] via 202.103.13.1, port2, 02:50:08, [1/0]
O       202.103.12.0/24 [110/2] via 202.103.13.1, port2, 02:50:08, [1/0]
```

4. 初始状态下查看 FW1 的 OSPF 邻居状态与路由信息，邻居状态为 Full，OSPF 路由学习正常。

```bash
FW1 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
202.103.12.2      1   Full/Backup     00:00:38    202.103.12.2    port2
202.103.13.2      1   Full/DR         00:00:40    202.103.13.2    port3

FW1 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.1.0/24 [110/2] via 202.103.12.2, port2, 02:49:28, [1/0]
O       10.10.2.0/24 [110/2] via 202.103.13.2, port3, 02:49:28, [1/0]
```

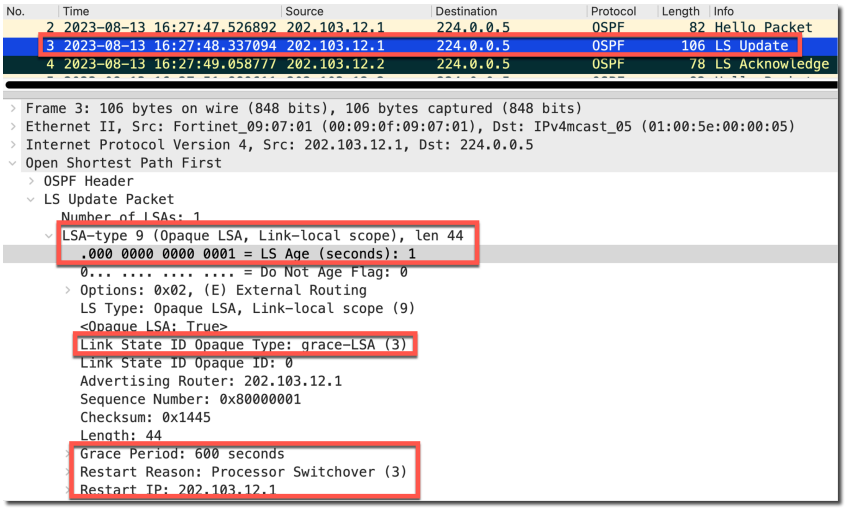
5. 使 FW1 和 FW2 发生 HA 主备倒换，此时 FW2 成为 HA 主设备。

```bash
FW2 # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:6233 b:4732229
        traffic.total = s:0 p:6286 b:4735596
        activity.ha_id_changes = 4
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=7 Debug=0
nvcluster=1, ses_pickup=0, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM23000176:      Primary, serialno_prio=0, usr_priority=100, hostname=FW2
FGVM08TM23000175:    Secondary, serialno_prio=1, usr_priority=128, hostname=FW1

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.1, primary_id=0
FGVM08TM23000176:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM23000175:    Secondary, ha_prio/o_ha_prio=1/1
```

6. 查看 FW2 的 OSPF Debug 信息，FW2 成为 HA 主后，向 OSPF 邻居发送 grace LSA（Type 9），通知邻居自己进入了 GR 过程，包含 GR 的原因和 GR 最大时间，LS Age 为 1s。

```bash
FW2 # diagnose ip router ospf level info
FW2 # diagnose ip router ospf all enable
FW2 # diagnose debug console time enable

2023-08-13 16:27:48 OSPF: SEND[LS-Upd]: To 224.0.0.5 via port2:202.103.12.1, length 120
2023-08-13 16:27:48 OSPF:     LS type 9 (Link-Local Opaque-LSA)
2023-08-13 16:27:48 OSPF:     Grace-LSA
2023-08-13 16:27:48 OSPF:     Grace Period: 600
2023-08-13 16:27:48 OSPF:     Graceful Restart Reason: Switch to redundant control processor
2023-08-13 16:27:48 OSPF:     IP interface Address: 202.103.12.1
```

7. 该报文格式如下图所示，LS age 为 1。



8. 在 FW2 进入 GR 状态后，查看 FW3 的 OSPF 邻居状态与 OSPF 路由状态，FW3 进入 GR Helper 状态（* 标记），Dead Time 变为 FW1/FW2 配置的 GR period 600s。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
202.103.12.1      1   Full/Backup     00:09:51*   202.103.12.1    port2

FW3 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.2.0/24 [110/3] via 202.103.12.1, port2, 00:33:17, [1/0]
O       202.103.13.0/24 [110/2] via 202.103.12.1, port2, 00:34:16, [1/0]
```

9. 在 FW3 上查看 OSPF 的 Debug 信息，FW3 收到 GR LSA 后，进入 GR Helper 模式。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # diagnose ip router ospf level info
FW3 # diagnose ip router ospf all enable
FW3 # diagnose debug console time enable

2023-08-13 16:27:48 OSPF: RECV[LS-Upd]: From 202.103.12.1 via port2:202.103.12.2 (202.103.12.1 -> 224.0.0.5)
2023-08-13 16:27:48 OSPF:     LS type 9 (Link-Local Opaque-LSA)
2023-08-13 16:27:48 OSPF:   Grace-LSA
2023-08-13 16:27:48 OSPF:     Grace Period: 600
2023-08-13 16:27:48 OSPF:     Graceful Restart Reason: Switch to redundant control processor
2023-08-13 16:27:48 OSPF:     IP interface Address: 202.103.12.1
2023-08-13 16:27:48 OSPF: Router: Enter Helper mode by grace LSA
```

0. 查看 FW3 上的 OSPF 9 类 LSA 的状态，此时 LS Age 未到达 3600s。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # get router info ospf database opaque-area

            OSPF Router with ID (202.103.12.2) (Process ID 0, VRF 0)
                Link-Local Opaque-LSA (Link port2:202.103.12.2)
  LS age: 5
  Options: 0x2 (*|-|-|-|-|-|E|-)
  LS Type: Link-Local Opaque-LSA
  Link State ID: 3.0.0.0 (Link-Local Opaque-Type/ID)
  Opaque Type: 3
```

```
Opaque ID: 0
Advertising Router: 202.103.12.1
LS Seq Number: 80000001
Checksum: 0x1445
Length: 44
    Grace Period: 600
    Graceful Restart Reason: Switch to redundant control processor
    IP Interface Address: 202.103.12.1
```

1. 等待 FW2 与 FW3 的 OSPF 邻居变为 Full 状态后，FW2 的 GR 完成，FW3 退出 GR Helper 状态，期间 OSPF 路由没有丢失，Dead Time 从 40s 开始倒计时。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
202.103.12.1      1   Full/Backup     00:00:36    202.103.12.1    port2

FW3 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.2.0/24 [110/3] via 202.103.12.1, port2, 00:00:03, [1/0]
O       202.103.13.0/24 [110/2] via 202.103.12.1, port2, 00:34:31, [1/0]
```

2. 在 FW2 上查看 OSPF 的 Debug 信息，FW2 正常退出了 GR，并发送 Age 为 3600 的 grace-LSA。

```bash
FW2 # diagnose ip router ospf level info
FW2 # diagnose ip router ospf all enable
FW2 # diagnose debug console time enable

2023-08-13 16:27:57 OSPF: ROUTER[Process:0, RouterID:202.103.12.1]: Exit Restarting normally
2023-08-13 16:27:57 OSPF: SEND[LS-Upd]: To 224.0.0.5 via port3:202.103.13.1, length 72
2023-08-13 19:05:59 OSPF:     LS age 3600
2023-08-13 19:05:59 OSPF:     LS type 9 (Link-Local Opaque-LSA)
2023-08-13 19:05:59 OSPF:   Grace-LSA
```

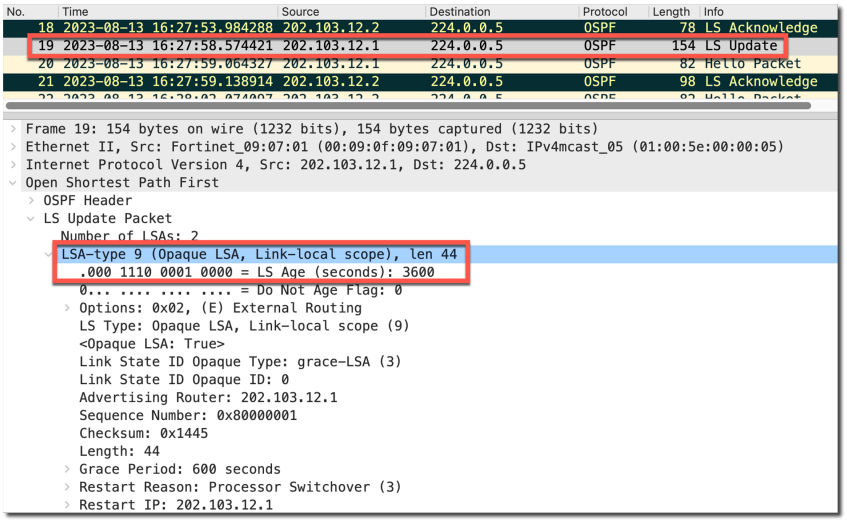
3. 在 FW3 上查看 OSPF 的 Debug 信息，FW3 收到 FW2 发送的 maxage 的 LSA 后，等待几秒后退出 GR Helper 状态。

```bash
2023-08-13 16:28:08 OSPF: ROUTER: Exit Restart Helper mode for neighbor(port2:202.103.12.2-202.103.12.1) by receiving maxage grace-LSA
```

4. 再次查看 FW3 上的 OSPF 9 类 LSA 的状态，此时 LS Age 直接到达 3600s。

```bash
FW3 # get router info ospf database opaque-link

            OSPF Router with ID (202.103.12.2) (Process ID 0, VRF 0)
                Link-Local Opaque-LSA (Link port2:202.103.12.2)
  LS age: 3600
  Options: 0x2 (*|-|-|-|-|-|E|-)
  LS Type: Link-Local Opaque-LSA
  Link State ID: 3.0.0.0 (Link-Local Opaque-Type/ID)
  Opaque Type: 3
  Opaque ID: 0
  Advertising Router: 202.103.12.1
  LS Seq Number: 80000001
  Checksum: 0x1445
  Length: 44
      Grace Period: 600
      Graceful Restart Reason: Switch to redundant control processor
      IP Interface Address: 202.103.12.1
```

5. 抓包查看 FW3 收到的这个 maxage grace-LSA，可以看到 LSA 的 age 为 3600s，达到了 OSPF 定义的 maxage（https://www.rfc-editor.org/rfc/rfc2328#appendix-B ↗），可以理解为 FW2 的 OSPF 邻居状态为 Full 后，会通过再次发送此类型的 LSA 告知 GR Helper，可以退出 GR Helper 模式了。



6. GR 期间，FW2/FW4 从 FW1 重启前同步的路由一直存在（route-ttl 600，prio = 2164260865）。

```
FW2 # get router info kernel | grep 10.10.
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2
dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2
dev=5(port3)
```

7. GR 结束后，学到了新的 OSPF 路由（prio = 1）。

```
FW2 # get router info kernel | grep 10.10.
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2 dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2
dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2 dev=5(port3)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2
dev=5(port3)
```

8. 整个 HA 切换引起的 GR 期间，所有设备路由转发表未发生实际变化，流量不会中断。

注意事项

1. 在以上测试步骤过程中，如果 FW1/FW2 在 GR 的过程中发现了 OSPF 拓扑变化，则会直接终端并退出 GR 的过程，同时通知 FW3/FW4 退出 GR Helper 状态，导致 OSPF 路由提前从 GR Helper 设备上消失，导致业务临时中断。

```
FW2 # diagnose ip router ospf level info
FW2 # diagnose ip router ospf all enable
FW2 # diagnose debug console time enable

2023-08-13 17:45:46 OSPF: ROUTER[Process:0, RouterID:202.103.12.1]: Exit Restarting by NbrChanged (port3:202.103.13.1-
202.103.13.2)
2023-08-13 17:45:46 OSPF: LSA[-:Type9:3.0.0.0:(self)]: Flooding via interface[port2:202.103.12.1]
2023-08-13 17:45:46 OSPF: LSA[-:Type9:3.0.0.0:(self)]: Flooding to neighbor[202.103.12.2]
......
2023-08-13 17:45:46 OSPF: SEND[LS-Upd]: To 224.0.0.5 via port2:202.103.12.1, length 72
2023-08-13 17:45:46 OSPF:     LS age 3600
2023-08-13 17:45:46 OSPF:     LS type 9 (Link-Local Opaque-LSA)
```

2. 为了避免 OSPF 拓扑变化导致的 GR 提前退出，可以在 FW1/FW2 上开启 OSPF 的 restart-on-topology-change，开启后，OSPF 进程在 GR 的过程中将忽略拓扑变化，继续完成 GR 的整个过程。

> **重要**
>
> 仅支持 7.2.0 及以后版本。

```
config router ospf
    set restart-on-topology-change enable
end
```

3. 如下 Debug 信息所示，FW1/FW2 在 GR 期间探测到拓扑变化，但没有立即退出 GR，而是继续正常完成了 GR。

```
2023-08-13 19:05:56 OSPF: ROUTER[Process:0, RouterID:202.103.12.1]: Exit Restarting by NbrChanged (port3:202.103.13.1- bash
202.103.13.2)
2023-08-13 19:05:56 OSPF: ROUTER[Process:0, RouterID:202.103.12.1]: skip exiting restart due to topology change
......
2023-08-13 19:05:59 OSPF: ROUTER[Process:0, RouterID:202.103.12.1]: Exit Restarting normally
......
2023-08-13 19:05:59 OSPF: SEND[LS-Upd]: To 224.0.0.6 via port2:202.103.12.1, length 72
2023-08-13 19:05:59 OSPF:     LS age 3600
2023-08-13 19:05:59 OSPF:     Options 0×2
2023-08-13 19:05:59 OSPF:     LS type 9 (Link-Local Opaque-LSA)
2023-08-13 19:05:59 OSPF:     Advertising Router 202.103.12.1
2023-08-13 19:05:59 OSPF:   Grace-LSA
......
```

GR 与 BFD 的使用关系

> **重要**
>
> BGP GR 或 OSPF GR 不建议与 BFD 同时使用。

GR 用于 FortiGate 通知邻居自己的路由进程将重启，并且邻居从 FortiGate 学到的路由应在重启期间保留。但 BFD 检测到故障时会将动态路由邻居重置。这种操作上的差异使得在 GR 的同时使用 BFD 会导致功能上的冲突，可能导致 GR 期间流量中断。

业界通用的做法是在使用 GR 时关闭 BFD。例如在 FortiGate 和 GCP（Google 云平台）之间配置动态路由 + GR + BFD 时：

- GCP 将关闭 BFD 并发送 AdminDown（BFD 事件）到 FortiGate（参考 https://cloud.google.com/network-connectivity/docs/router/concepts/bfd#graceful-restart-and-bfd ↗ ）。

BGP

BGP 基础配置

BGP 计时器

BGP GR

BGP 路由过滤

BGP 路由刷新

BGP 下一跳递归

路由反射器

BGP 多路径

additional-path

附加 AS-Path

BGP Debug

BGP 常见问题

- 清除 BGP 路由条目
- 路由震荡
- 使用 Loopback 接口建立 BGP
- BGP 与静态路由的优先顺序
- 关闭 Leftmost AS 检查
- BGP 选路忽略 AS-PATH
- 多个本地 AS 的配置和通告
- RR 反射路由无法修改属性

BGP 基础配置

BGP 简介

边界网关协议（BGP）是一种标准化的路由协议，用于在互联网上路由流量。它在互联网上的自治系统（AS）之间交换路由信息，并根据路径、网络策略和规则集做出路由决策。BGP 包含两个子集：内部 BGP（iBGP）和外部 BGP（eBGP）。iBGP 用于在自己的网络内部使用。eBGP 用于连接不同的网络，并且是互联网骨干的主要路由协议。

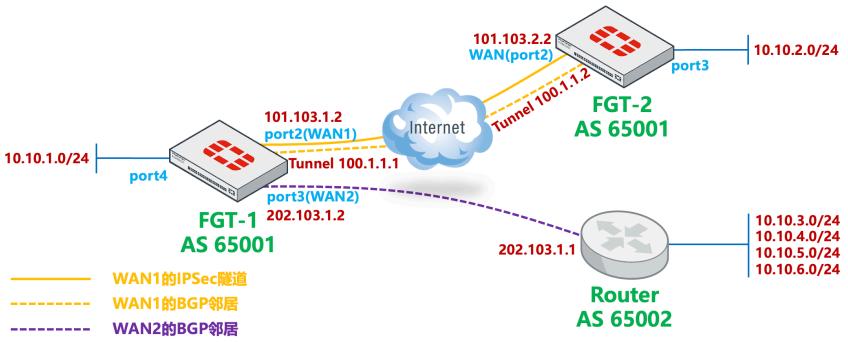
FortiGate 的 BGP 基础配置参数如下：

选项	描述
本地 AS	本地 FortiGate 的 AS 号。
Router ID	一个用于在网络中标识你的路由器的唯一 ID，通常以 x.x.x.x 的格式表示。
邻居	FortiGate 将与之进行 BGP 连接的邻居。配置远程路由器的自治系统（Remote AS）号码，与邻居连接所使用的任何其他属性，以及 IPv4 和 IPv6 路由过滤。
邻居组	共享相同出站策略配置的邻居组，一般用于被多个 BGP 邻居主动连接的情况。
邻居范围	BGP 邻居的源地址范围，用于分配给一个邻居组。
IPv4 & IPv6 网络	要宣告给其他 BGP 路由器的网络。
IPv4 & IPv6 重分发	启用协议的重分发。指定某个路由协议的路由，可以通过 route-map 进行过滤。
Dampening	启用路由振荡抑制以减少振荡路由的传播。
优雅重启（graceful-restart）	启用 BGP 优雅重启（graceful-restart），相邻的路由器在 FortiGate 重启 BGP 进程时保持 BGP 路由条目。在 HA 中发生主备切换时非常有用（暂不支持 GR Helper 配置为 GR Helper-only 模式）。
高级选项	多种 BGP 高级选项，如 Local Preference、Local Preference、Distance internal、Keepalive、Holdtime 等。
最佳路径选择	路径选择的相关属性配置。

BGP 选路原则

1. 优先选择具有最高的 Weight 的路由（Weight 仅本地生效）。
2. 优先选择具有最高的 Local Preference 的路由（仅在本 AS 内生效）。
3. 优先选择本地路由器产生的路由（下一跳为 0.0.0.0）。
4. 优先选择具有最短的 AS-Path 的路由。
5. IGP 路由 > EGP 路由 > incomplete 路由。
6. 优先选择具有最低 MED（可在 AS 之间交换）的路由。
7. eBGP 路径优先于 iBGP 路由。
8. iBGP 路由优先选择更近的 iBGP 邻居学习到的路由。
9. eBGP 路由优先选择更早学习到的 eBGP 路由。
0. 优先选择具有最小邻居 Router ID 的路由。
1. 优先选择具有最小邻居 IP 的路由。

网络拓扑



1. 网络连接：

   - FGT-1 与 FGT-2 通过 ISP1 的 Internet 建立 IPSec 隧道，二阶段保护网段两端均为 0.0.0.0/0（配置略）。
   - 为两台 FortiGate 的 IPSec Tunnel 接口配置 IP，并配置 IPSec 业务互访的放通策略（配置略）：
     - FGT-1 Tunnel IP：100.1.1.1。
     - FGT-2 Tunnel IP：100.1.1.2。
   - FGT-1 与 Router 为直连网段设备。

2. BGP 邻居：

   - FGT-1 与 FGT-2 通过 IPSec 隧道建立 iBGP 邻居，BGP 均创建在 IPSec Tunnel 接口上。
   - FGT-1 与 Router 之间使用物理口建立 eBGP 邻居。

3. 路由学习：

   - FGT-1 和 FGT-2 互相学习 IBGP 路由并通过 IPSec 隧道通信。
   - FGT-1 从 Router 学习 eBGP 路由，并发布到 FGT-2，但不允许发布任何 iBGP 路由到 Router。
   - FGT-1 通过 eBGP 向 Router 发布默认路由。

配置步骤

基础配置

1. FGT-1 的 VPN 配置。

```bash
config vpn ipsec phase1-interface
    edit "to_FGT-2"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set remote-gw 101.103.2.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FGT-2"
        set phase1name "to_FGT-2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

2. FGT-2 的 VPN 配置。

```bash
config vpn ipsec phase1-interface
    edit "to_FGT-1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set remote-gw 101.103.1.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FGT-1"
        set phase1name "to_FGT-1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

3. FGT-1 的接口与静态路由配置。

```bash
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.1.2 255.255.255.0
        set alias "WAN1"
    next
    edit "port3"
        set vdom "root"
        set ip 202.103.1.2 255.255.255.0
        set alias "WAN2"
    next
    edit "port4"
        set vdom "root"
        set ip 10.10.1.1 255.255.255.0
        set allowaccess ping https http
        set alias "LAN"
    next
    edit "to_FGT-2"
        set vdom "root"
        set ip 100.1.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.1.254 255.255.255.0
        set interface "port2"
    next
end

config router static
    edit 1
        set gateway 101.103.1.1
        set device "port2"
    next
end
```

4. FGT-2 的接口与静态路由配置。

```bash
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.2.2 255.255.255.0
        set alias "WAN"
    next
    edit "port3"
        set vdom "root"
        set ip 10.10.2.1 255.255.255.0
        set alias "LAN"
    next
    edit "to_FGT-1"
```

```
        set vdom "root"
        set ip 100.1.1.2 255.255.255.255
        set allowaccess ping https http
        set type tunnel
        set remote-ip 100.1.1.254 255.255.255.0
        set interface "port2"
    next
end

config router static
    edit 1
        set gateway 101.103.2.1
        set device "port2"
    next
end
```

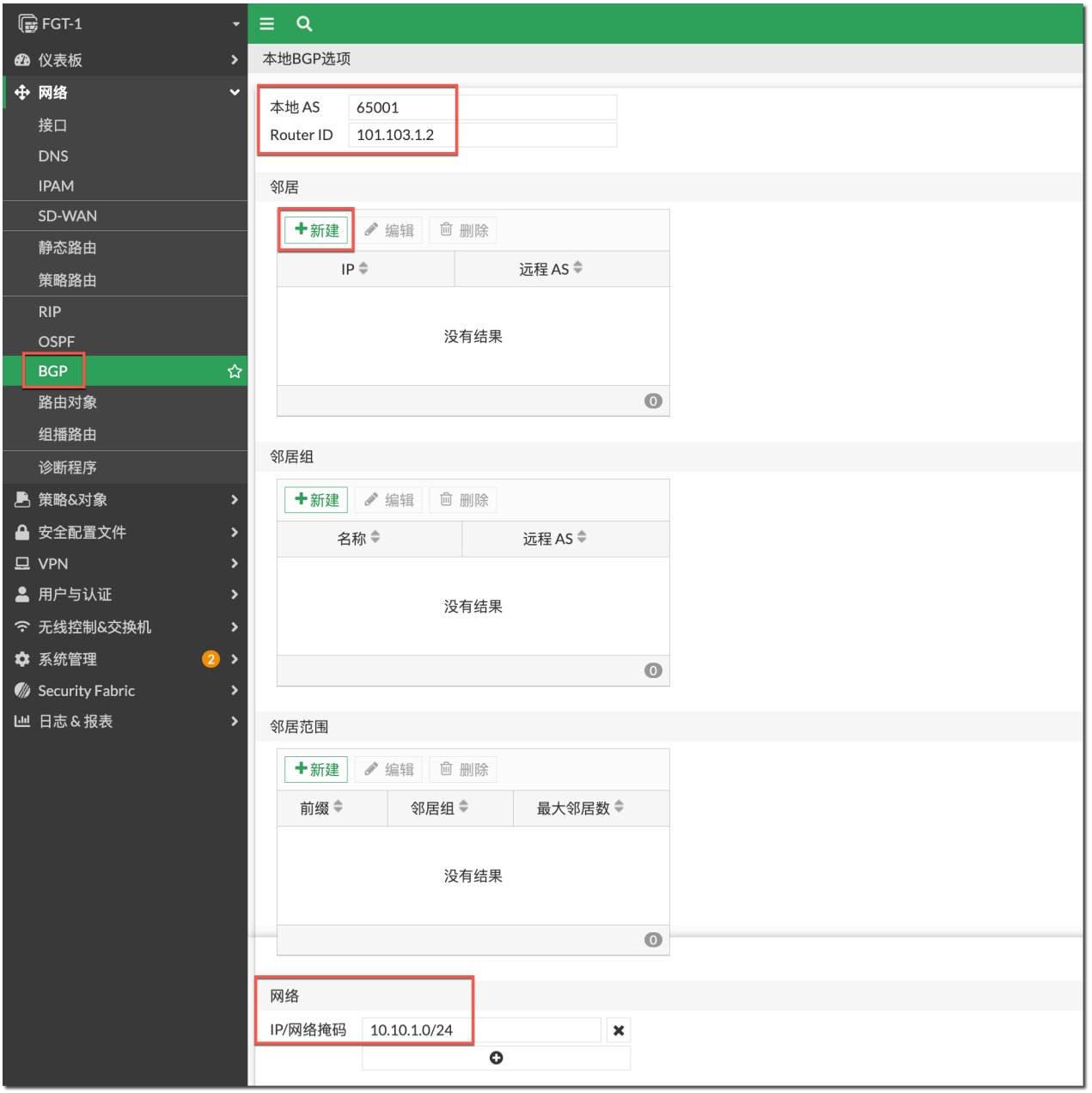
5. 配置安全策略，放通 FGT-1 和 FGT-2 间 IPSec 隧道的流量，按需配置其他安全策略（略）。

6. 等待 FGT-1 和 FGT-2 之间的 IPSec 隧道建立成功。

```bash
FGT-1 # get vpn ipsec tunnel summary
'to_FGT-2' 101.103.2.2:0  selectors(total,up): 1/1  rx(pkt,err): 395/0  tx(pkt,err): 396/0

FGT-2 # get vpn ipsec tunnel summary
'to_FGT-1' 101.103.1.2:0  selectors(total,up): 1/1  rx(pkt,err): 397/0  tx(pkt,err): 396/0
```

iBGP（FGT-1 to FGT-2）

1. 在 FGT-1 上进入网络 → BGP 页面，配置本地 AS 和 Router ID，在网络选项中配置要发布的本地路由。

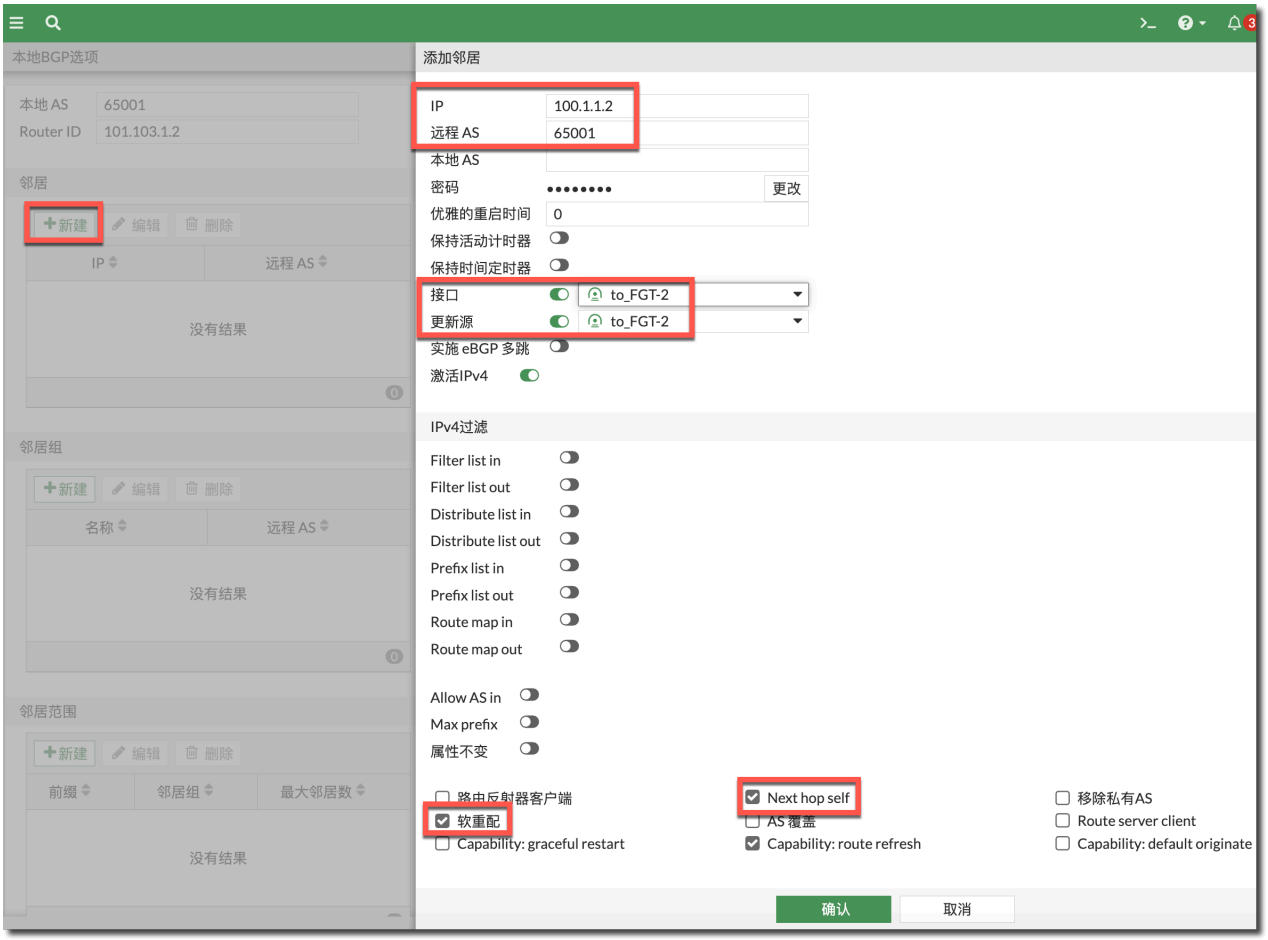


重要

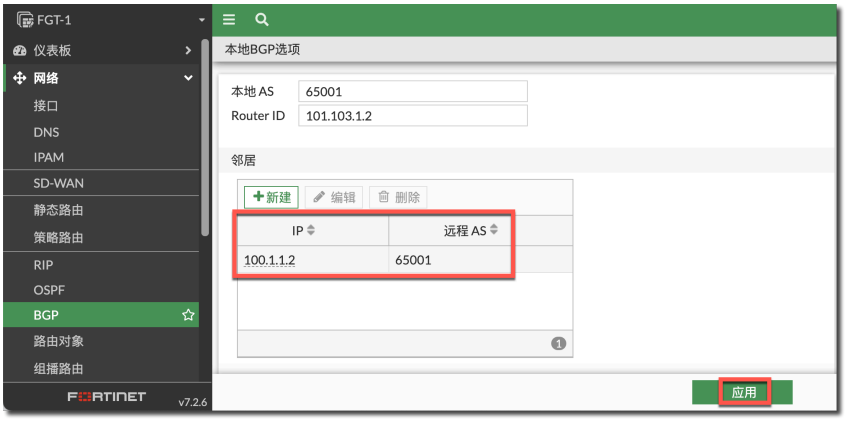
默认配置下，BGP 只能发布设备自身路由表中已经存在的路由条目（路由与掩码都要一致），如果需要发布路由表中不存在的路由条目，需要使用如下 CLI 关闭发布路由的检查。

```
config router bgp
    set network-import-check disable
end
```

2. 点击邻居选项中的新建按钮，在弹出的新建邻居页面中配置 iBGP 邻居的 IP、远程 AS，接口和更新源都选择 IPSec Tunnel 接口，开启“Next hop self”（FGT-1 向 FGT-2 宣告 Router 发送的 eBGP 路由时，将下一跳改为自己的 IPSec tunnel 口 IP），建议开启软重配功能，点击确认下发 BGP 邻居配置。



3. 在 BGP 页面点击应用下发 BGP 配置。



4. 相关 CLI 如下。

```bash
config router bgp
    set as 65001
    set router-id 101.103.1.2
    config neighbor
        edit "100.1.1.2"
            set next-hop-self enable
            set soft-reconfiguration enable
            set interface "to_FGT-2"
            set remote-as 65001
            set update-source "to_FGT-2"
        next
    end
    config network
```

```
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

5. 在 FGT-2 使用与 FGT-1 同样的步骤配置 iBGP，过程略，相关配置 CLI 如下。

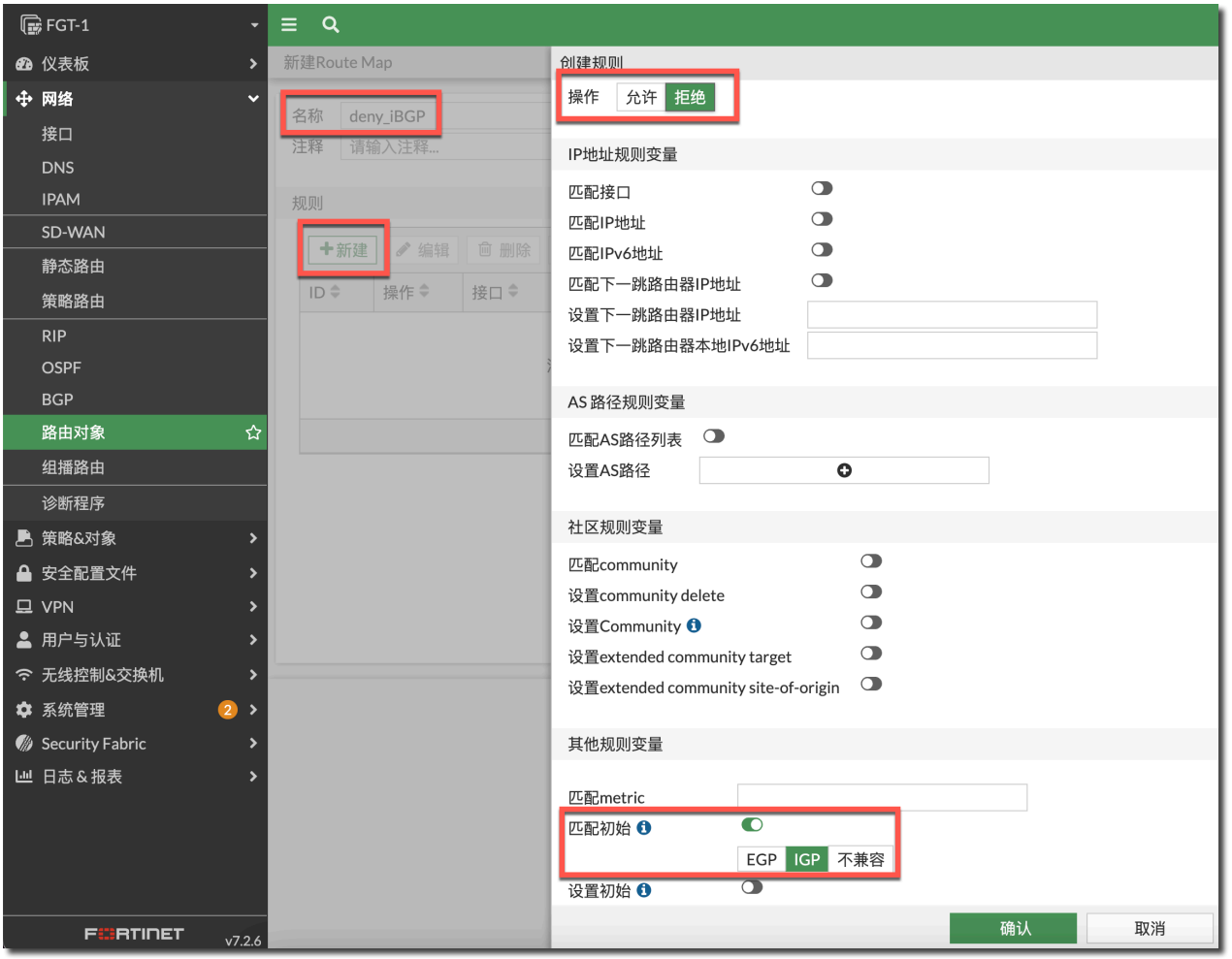
```bash
config router bgp
    set as 65001
    set router-id 101.103.2.2
    config neighbor
        edit "100.1.1.1"
            set soft-reconfiguration enable
            set interface "to_FGT-1"
            set remote-as 65001
            set update-source "to_FGT-1"
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

eBGP

1. 在 FGT-1 上新建 Route Map，用于阻止从 FGT-2 学到的 iBGP 路由发布到 eBGP 邻居 Router。进入网络 → 路由对象页面，点击新建 → Route Map。



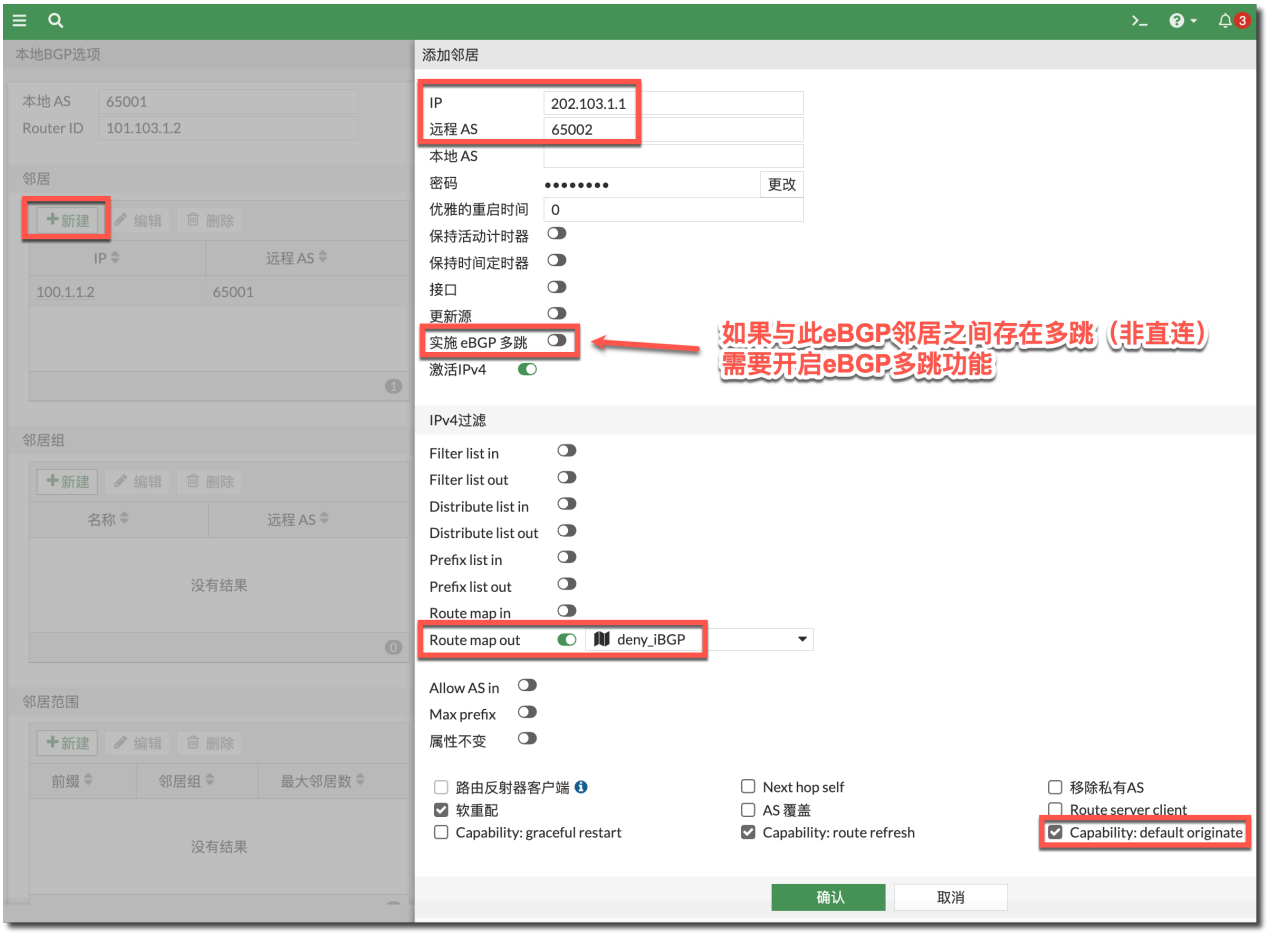
2. 配置 Route Map 的名称，并点击规则下的新建按钮。在弹出的新建规则页面中配置动作为拒绝，在“其他规则变量”中开启“匹配初始”，并选择 IGP，下发规则和 Route Map 的配置。




```bash
config router route-map
    edit "deny_iBGP"
        config rule
            edit 1
                set action deny
                set match-origin igp
            next
        end
    next
end
```

3. 在 FGT-1 上进入网络 → BGP 页面，点击邻居选项中的新建按钮，在弹出的新建邻居页面中配置 eBGP 邻居的 IP、远程 AS，在“IPv4 过滤”中的“Route map out”引用上步创建的 Route Map，开启“Capability: default originate”功能，向 Router 宣告默认路由，建议开启软重配功能，随后下发 BGP 邻居配置和 BGP 配置。

> **重要**
>
> 如果与此 eBGP 邻居之间存在多跳（非直连），需要开启 eBGP 多跳功能（set ebgp-enforce-multihop enable）。




```bash
config router bgp
    config neighbor
        edit "202.103.1.1"
            set capability-default-originate enable
            set soft-reconfiguration enable
            set remote-as 65002
            set route-map-out "deny_iBGP"
        next
    end
end
```

4. 在 Router 上参考 FGT-1 配置 eBGP 邻居，配置略，相关 CLI 如下。

```bash
config router bgp
    set as 65002
    set router-id 202.103.1.1
    config neighbor
        edit "202.103.1.2"
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.3.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.4.0 255.255.255.0
        next
        edit 3
            set prefix 10.10.5.0 255.255.255.0
        next
```

```
        edit 4
            set prefix 10.10.6.0 255.255.255.0
        next
    end
end
```

结果验证

FGT-1

1. 在 FGT-1 上查看 BGP 邻居状态，可以看到和 FGT-2、Router 分别建立了 BGP 邻居，并学习到路由（从 FGT-2 学到 1 条，从 Router 学习到 4 条）。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors`。

```bash
FGT-1 # get router info bgp summary

VRF 0 BGP router identifier 101.103.1.2, local AS number 65001
BGP table version is 8
2 BGP AS-PATH entries
0 BGP community entries

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.1.1.2       4      65001     142     162        8    0    0 00:29:39        1
202.103.1.1     4      65002      15      16        8    0    0 00:11:01        4

Total number of neighbors 2
```

2. 在 FGT-1 上查看从不同邻居收到的路由条目。

```bash
FGT-1 # get router info bgp neighbors 100.1.1.2 received-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.2.0/24     100.1.1.2                   100      0        0 i ←/→

Total number of prefixes 1

FGT-1 # get router info bgp neighbors 202.103.1.1 received-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.3.0/24     202.103.1.1                           0        0 65002 i ←/→
*> 10.10.4.0/24     202.103.1.1                           0        0 65002 i ←/→
*> 10.10.5.0/24     202.103.1.1                           0        0 65002 i ←/→
*> 10.10.6.0/24     202.103.1.1                           0        0 65002 i ←/→

Total number of prefixes 4
```

3. 在 FGT-1 上查看向不同邻居发布的路由条目。

```bash
FGT-1 # get router info bgp neighbors 100.1.1.2 advertised-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     100.1.1.1                   100  32768        0 i ←/→
*>i10.10.3.0/24     100.1.1.1                   100      0        0 65002 i ←/→
*>i10.10.4.0/24     100.1.1.1                   100      0        0 65002 i ←/→
*>i10.10.5.0/24     100.1.1.1                   100      0        0 65002 i ←/→
*>i10.10.6.0/24     100.1.1.1                   100      0        0 65002 i ←/→
```

```
Total number of prefixes 5

FGT-1 # get router info bgp neighbors 202.103.1.1 advertised-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        202.103.1.2                    100  32768        0 i ←/→

Total number of prefixes 1
```

4. 在 FGT-1 上查看 BGP 路由数据库，可以看到 iBGP 路由前边会有 i 标记，包含本地发布的路由（weight 为 32768）。

```
FGT-1 # get router info bgp network                                                    bash
VRF 0 BGP table version is 8, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                        100  32768        0 i ←/1>
*>i10.10.2.0/24     100.1.1.2       0              100      0        0 i ←/1>
*> 10.10.3.0/24     202.103.1.1     0                       0        0 65002 i ←/1>
*> 10.10.4.0/24     202.103.1.1     0                       0        0 65002 i ←/1>
*> 10.10.5.0/24     202.103.1.1     0                       0        0 65002 i ←/1>
*> 10.10.6.0/24     202.103.1.1     0                       0        0 65002 i ←/1>

Total number of prefixes 6
```

5. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```
FGT-1 # get router info bgp network 10.10.3.0                                          bash
VRF 0 BGP routing table entry for 10.10.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   100.1.1.2
  Original VRF 0
  65002
    202.103.1.1 from 202.103.1.1 (202.103.1.1)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Fri Jan  5 17:12:47 2024
```

6. 查看 FGT-1 的路由表中的 BGP 路由，可以看到 FGT-1 从 FGT-2、Router 学习到的路由均已被放入路由表，iBGP 路由的 distance 为 200，eBGP 路由的 distance 为 20。

```
FGT-1 # get router info routing-table bgp                                              bash
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 100.1.1.2 (recursive via to_FGT-2 tunnel 101.103.2.2), 00:34:19, [1/0]
B       10.10.3.0/24 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 00:17:59, [1/0]
B       10.10.4.0/24 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 00:17:59, [1/0]
B       10.10.5.0/24 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 00:17:59, [1/0]
B       10.10.6.0/24 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 00:17:59, [1/0]
```

7. FGT-1 内网 PC 可以正常访问 FGT-2 和 Router 内网 PC。

FGT-2

1. 在 FGT-2 上查看 BGP 邻居状态，可以看到和 FGT-1 建立了 BGP 邻居，并学习到路由（从 FGT-1 学到 5 条，其中 1 条为 FGT-1 发布，4 条为 Router 发布）。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors` 。

```bash
FGT-2 # get router info bgp summary

VRF 0 BGP router identifier 101.103.2.2, local AS number 65001
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.1.1.1 4      65001     172     154        1    0    0 00:39:35        5

Total number of neighbors 1
```

2. 在 FGT-2 上查看从 BGP 邻居收到的路由条目。

```bash
FGT-2 # get router info bgp neighbors 100.1.1.1 received-routes
VRF 0 BGP table version is 2, local router ID is 101.103.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     100.1.1.1                 100      0        0 i ←/→
*>i10.10.3.0/24     100.1.1.1                 100      0        0 65002 i ←/→
*>i10.10.4.0/24     100.1.1.1                 100      0        0 65002 i ←/→
*>i10.10.5.0/24     100.1.1.1                 100      0        0 65002 i ←/→
*>i10.10.6.0/24     100.1.1.1                 100      0        0 65002 i ←/→

Total number of prefixes 5
```

3. 在 FGT-2 上查看向邻居发布的路由条目。

```bash
FGT-2 # get router info bgp neighbors 100.1.1.1 advertised-routes
VRF 0 BGP table version is 2, local router ID is 101.103.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.2.0/24     100.1.1.2                 100  32768        0 i ←/→

Total number of prefixes 1
```

4. 在 FGT-2 上查看 BGP 路由数据库，可以看到 iBGP 路由前边会有 i 标记，包含本地发布的路由（weight 为 32768）。Router 发布的路由经过 FGT-1 发送给 FGT-2，变为 iBGP 路由。

```bash
FGT-2 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 101.103.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     100.1.1.1       0         100      0        0 i ←/1>
*> 10.10.2.0/24     0.0.0.0                   100  32768        0 i ←/1>
*>i10.10.3.0/24     100.1.1.1       0         100      0        0 65002 i ←/1>
*>i10.10.4.0/24     100.1.1.1       0         100      0        0 65002 i ←/1>
*>i10.10.5.0/24     100.1.1.1       0         100      0        0 65002 i ←/1>
*>i10.10.6.0/24     100.1.1.1       0         100      0        0 65002 i ←/1>

Total number of prefixes 6
```

5. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```bash
FGT-2 # get router info bgp network 10.10.3.0
VRF 0 BGP routing table entry for 10.10.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65002
    100.1.1.1 from 100.1.1.1 (101.103.1.2)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Last update: Fri Jan  5 17:12:50 2024
```

6. 查看 FGT-2 的路由表中的 BGP 路由，可以看到 FGT-2 从 FGT-1 学习到的路由均已被放入路由表，iBGP 路由的 distance 为 200。

```bash
FGT-2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 100.1.1.1 (recursive via to_FGT-1 tunnel 101.103.1.2), 00:43:14, [1/0]
B       10.10.3.0/24 [200/0] via 100.1.1.1 (recursive via to_FGT-1 tunnel 101.103.1.2), 00:24:36, [1/0]
B       10.10.4.0/24 [200/0] via 100.1.1.1 (recursive via to_FGT-1 tunnel 101.103.1.2), 00:24:36, [1/0]
B       10.10.5.0/24 [200/0] via 100.1.1.1 (recursive via to_FGT-1 tunnel 101.103.1.2), 00:24:36, [1/0]
B       10.10.6.0/24 [200/0] via 100.1.1.1 (recursive via to_FGT-1 tunnel 101.103.1.2), 00:24:36, [1/0]
```

7. FGT-2 内网 PC 可以正常访问 FGT-1 和 Router 内网 PC。

Router

1. 在 Router 上查看 BGP 邻居状态，可以看到和 FGT-1 建立了 BGP 邻居，并学习到路由（从 FGT-1 学到 1 条，为 FGT-1 发布的默认路由）。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors` 。

```bash
Router # get router info bgp summary

VRF 0 BGP router identifier 202.103.1.1, local AS number 65002
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.1.2 4      65001      86      95        1    0    0 00:27:29        1

Total number of neighbors 1
```

2. 在 Router 上查看从 BGP 邻居收到的路由条目。

```bash
Router # get router info bgp neighbors 202.103.1.2 received-routes
VRF 0 BGP table version is 1, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        202.103.1.2                             0        0 65001 i ←/→

Total number of prefixes 1
```

3. 在 FGT-2 上查看向邻居发布的路由条目。

```bash
Router # get router info bgp neighbors 202.103.1.2 advertised-routes
VRF 0 BGP table version is 1, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.3.0/24     202.103.1.1                       100  32768        0 i ←/→
*> 10.10.4.0/24     202.103.1.1                       100  32768        0 i ←/→
*> 10.10.5.0/24     202.103.1.1                       100  32768        0 i ←/→
*> 10.10.6.0/24     202.103.1.1                       100  32768        0 i ←/→

Total number of prefixes 4
```

4. 在 Router 上查看 BGP 路由数据库，可以看到除了本地发布的路由（weight 为 32768）和 FGT-1 发布的默认路由，无法学习到 FGT-1 和 FGT-2 发布的 iBGP 路由，这说明在 FGT-1 上配置的 route-map 已生效（过滤 IGP 路由）。

```bash
Router # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
```

```
*> 0.0.0.0/0        202.103.1.2     0                      0        0 65001 i <-/1>
*> 10.10.3.0/24     0.0.0.0                      100  32768        0 i <-/1>
*> 10.10.4.0/24     0.0.0.0                      100  32768        0 i <-/1>
*> 10.10.5.0/24     0.0.0.0                      100  32768        0 i <-/1>
*> 10.10.6.0/24     0.0.0.0                      100  32768        0 i <-/1>

Total number of prefixes 5
```

5. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```bash
Router # get router info bgp network 0.0.0.0
VRF 0 BGP routing table entry for 0.0.0.0/0
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65001
    202.103.1.2 from 202.103.1.2 (101.103.1.2)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Fri Jan  5 17:12:51 2024
```

6. 查看 Router 的路由表中的 BGP 路由，只有一条默认路由指向 FGT-1，distance 为 20（eBGP 路由）。

```bash
Router # get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [20/0] via 202.103.1.2 (recursive is directly connected, port2), 00:31:31, [1/0]
```

7. Router 内网 PC 可以正常访问 FGT-1 和 FGT-2 内网 PC。

BGP 邻居建立排错

1. 检查 FGT-1 的 WAN1 接口与 FGT-2 的 WAN 接口之间的连通性。
2. 检查 FGT-1 和 FGT-2 之间的 IPSec 隧道是否建立成功，并验证 IPSec 隧道的连通性。
3. 检查 BGP 建立的状态：
   - Idle：本地 FortiGate 尚未与邻居启动 BGP 进程。这可能是因为 eBGP 对等方距离较远，但未启用 multihop。
   - Connect：本地 FortiGate 已启动 BGP 进程，但尚未发起 TCP 连接，可能是由于不正确的路由导致无法发起协商。
   - Active：本地 FortiGate 已启动 TCP 连接，但没有收到对端的响应。
4. 如果 BGP 邻居间建立 TCP 连接存在问题，可以使用命令 `diagnose sniffer packet any 'tcp and port 179' 4` 来判断协商问题。

BGP 计时器

基础计时器

keepalive-timer

- 设置 FortiGate 向已建立连接的对等体发送 BGP 保活消息的频率（0~65535s，默认值为 60s）。
- `keepalive-timer` 是 BGP 的全局设置（`config router bgp`）。可以在每个对等体的配置上（`config neighbor` / `config neighbor-group`）使用 `keep-alive-timer` 选项覆盖全局配置。

holdtime-timer

- 设置 BGP 在将对等体标记为不可用之前等待的最长时间（3~65535s，默认值为 180s）。当 FortiGate 收到来自对等体的 keepalive 消息时，`holdtime-timer` 会重置为配置的值。如果在此期间没有收到保活消息，FortiGate 会认为对等体不再活跃，并将重置 BGP 邻居关系。
- `holdtime-timer` 是 BGP 的全局设置（`config router bgp`）。可以在每个对等体的配置上（`config neighbor` / `config neighbor-group`）使用 `holdtime-timer` 选项（名称相同）覆盖全局配置。

重要

- 可以在配置中将 `keepalive-timer` 设置为大于 `holdtime-timer` 的三分之一，但 FortiOS 实际使用的 `keepalive-timer` 最多会限制为 `holdtime-timer` 的三分之一（执行 RFC 4271 中的建议）。允许使用短于 `holdtime-timer` 三分之一的 `keepalive-timer`。例如：
- 若 `keepalive-timer` = 20s 且 `holdtime-timer` = 30s，FortiOS 将使用 10s 的实际 `keepalive-timer` 和 30s 的 `holdtime-timer`。
- 若 `keepalive-timer` = 5s 且 `holdtime-timer` = 30s，FortiOS 将使用 5s 的实际 `keepalive-timer` 和 30s 的 `holdtime-timer`。
- BGP `keepalive-timer` / `holdtime-timer` 在对等体之间协商，双方会选择通告的最小保活/保持时间值。
- 更改 BGP `keepalive-timer` 和 `holdtime-timer` 不会影响已建立的 BGP 邻居关系，除非它们被重置并重新建立，软重置（soft-reconfiguration）在此情况下无效。

advertisement-interval

- 设置 BGP 发送路由更新的最小间隔（0~600s，默认值为 30s）。
- 受 `advertisement-interval` 影响的路由更新包括：BGP 邻居状态变为 ESTABLISHED 后的初始路由通告、添加或撤销路由，以及软重置（soft-reconfiguration）后通告路由。
- 对等体之间的 `advertisement-interval` 值不需要协商。每个 BGP 对等体根据本地配置的 `advertisement-interval` 等待发送自己的路由通告。
- 建议为 iBGP 连接设置较短的间隔（如 RFC 4271 的第 10 节建议）。

connect-timer

- 设置 BGP 邻居状态在转换到 idle 状态之前，尝试通过 TCP 与 BGP 对等体尝试建立连接的最长时间（1~65535s，默认值为 120s）。
- 此设置仅在每个对等体（`config neighbor` / `config neighbor-group`）上配置，没有全局设置。

scan-time

- 设置 BGP 扫描所有 BGP 路由下一跳可达性的间隔（5~60s，默认值为 60s，设置为 0 表示禁用）。如果 BGP 路由条目的下一跳失效，则不使用该 BGP 路由条目。

路由抑制计时器

重要

- 必须先在 `config router bgp` 下设置 `set dampening enable`，这些路由抑制计时器才会出现在 CLI 中，且这些设置仅对 BGP 全局有效（不能在对等体的基础上配置）。
- 具体配置方法请参考：路由 → 动态路由 → BGP → BGP 常见问题 → 路由震荡 ↗章节。

dampening-max-suppress-time

- 设置路由可以被抑制的最长时间（1~255min，默认值为 60min），无论路由的稳定性如何。
- 默认值为 `dampening-reachability-half-life` 的四倍。

dampening-reachability-half-life

- 设置惩罚的半衰期时间（1~45min，默认值为 15min）。如果路由被认为是可达的（无论是否被抑制），路由的惩罚将减少一半。

dampening-unreachability-half-life

- 设置惩罚的半衰期时间（1~45min，默认值为 15min）。如果路由被认为是不可达的，路由的惩罚将减少一半。

平滑重启计时器

> **重要**
>
> 具体配置请参考：路由 → 动态路由 → BGP → BGP GR ↗ 章节。

graceful-restart-time

- 设置 BGP 在对等体重启时等待的时间长度（1~3600s，默认值为 120s）。一旦 FortiGate 在对等体重启后收到 BGP OPEN 消息，此计时器就会结束。请确保 `graceful-restart-time` 小于或等于 `holdtime-timer`。
- `graceful-restart-time` 是 BGP 的全局设置（`config router bgp`）。可以在每个对等体的基础上（`config neighbor` / `config neighbor-group`）使用 `restart-time` 选项覆盖此设置。

graceful-stalepath-time

- 设置 BGP 保留之前从正在重启的 BGP 对等体接收的陈旧路由（stalepath）的时间（1~3600s，默认值为 360s）。
- 一旦 BGP 检测到对等体重启（例如，收到带有 Graceful Restart Capability 字段，且“Restart State”置位的 BGP Open 报文），此计时器就会启动，允许 BGP 保留现有的陈旧路由（stalepath）足够长的时间，以保证 BGP 与正在重启的对等体重建过程中路由不会中断。
- `graceful-stalepath-time` 是 BGP 的全局设置（`config router bgp`）。可以在每个对等体的基础上（`config neighbor` / `config neighbor-group`）使用 `retain-stale-time` 选项覆盖此设置。

graceful-update-delay

- 设置 BGP 进程在重启后等待更新 BGP 路由选择的时间（1~3600s，默认值为 120s）。
- 与前两个计时器不同，此计时器适用于 FortiGate 自身重启 BGP 的情况（而非对等体重启）。
- FortiGate 将保留重启 BGP 前的陈旧路由（stalepath），并且在以下情况之前不会更新 BGP 路由选择：
  - 它从所有对等体（除了不支持 GR 的对等体以及“Restart State”置位的对等体）收到 RIB 结束的 BGP 标记。
  - 此计时器到期。

BGP GR

网络需求

FortiGate HA 集群与上下游路由器运行 BGP 协议，并开启 GR。在 FortiGate 的 HA 发生主备切换时，GR 需要确保每个路由器（包括 FortiGate 及其它路由器）在 HA 切换期间保持其路由表中的 BGP 路由（GR Helper），以避免流量中断。

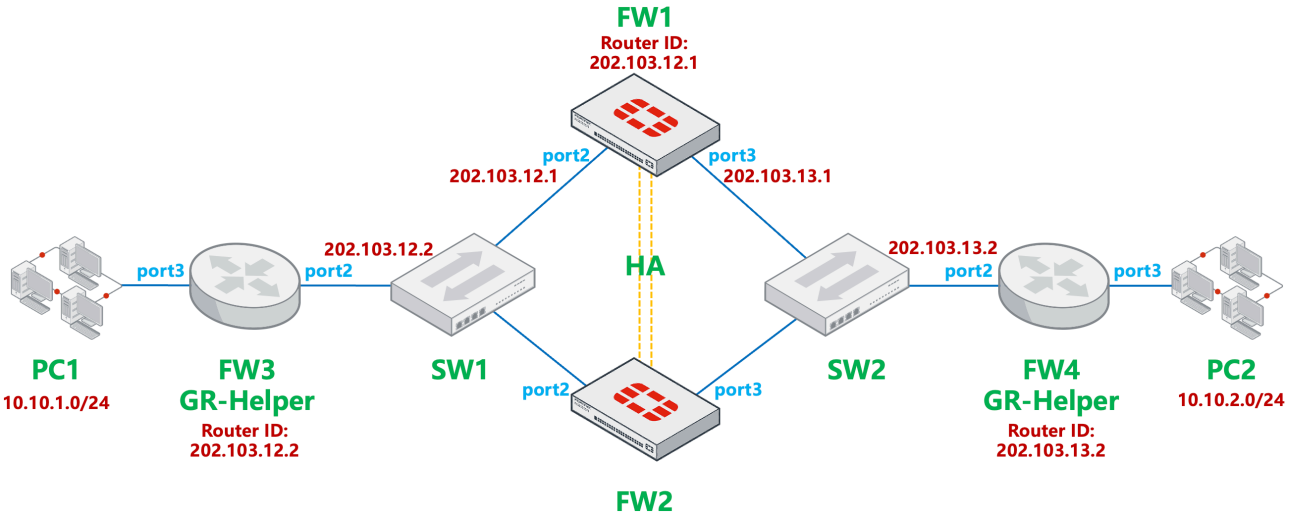
> **重要**
>
> 如果 FortiGate 的上下游为只支持 GR Helper Only（只能作为 GR Helper，不具有 GR Restarter 功能）模式的路由器，需要 FortiGate 运行 FortiOS 7.4.2 及以上版本才能正常完成 GR 过程。

在 FortiGate HA 集群中，BGP 进程相关工作仅在主设备上运行。当发生 HA 切换时，将在新选举的主机上启动一个新的 BGP 协商过程，在此过程中，新的主机需要在路由 Kernel 表中维持从原主机学习到的 BGP 路由，直至新的 BGP 邻居形成。

对于 BGP 的 GR，有以下几个计时器需要了解：

- `holdtime-timer`（默认 180s）：将 BGP 邻居标记为死亡的秒数。即在考虑与对等方的连接关闭之前，在 Keepalive、Update 或 Notification 消息之间等待的秒数。
- `graceful-restart-time`（默认 120s）：邻居重新启动所需的时间，即 GR-Helper 在删除过时路由之前等待 OPEN 消息的秒数。请确保 `graceful-restart-time` 小于或等于 `holdtime-timer`。
- `graceful-update-delay`（默认 120s）：GR 完成后，路由宣告/学习的延迟时间。
- `graceful-stalepath-time`（默认 360s）：在 GR-Helper 上保持 GR 邻居的过时路由的时间，即在删除之前保持过时路由的最大总时间。

网络拓扑



1. FW1 与 FW2 建立 A - P 模式的 HA。
2. FW1/FW2 与 FW3、FW4 分别建立 IBGP 邻居，FW3、FW4 宣告各自的网段。
3. FW1/FW2 与 FW3、FW4 开启 BGP Graceful Restart，FW1/FW2 发生主备切换后，FW3、FW4 可以进入 GR Helper 模式，协助 FW1/FW2 的 GR 过程。

配置步骤

1. 基础网络配置（略）。

2. FW1 与 FW2 的 HA 配置（略）。

3. 安全策略配置（略）。

4. 配置 FW1/FW2 的 BGP，开启 GR 功能，restart-period 设置为 600s，计时器配置均保持默认，邻居配置中开启路由反射器（RR），将两个 IBGP 邻居的路由互相反射，反射时将下一跳改为自己。

重要

- 虽然 GR Restarter 配置了 `graceful-restart-time`，但该参数只是告知邻居（GR Helper）GR 过程的最大时间。只要 GR Restarter 认为自己完成了 GR（路由收敛完成），即使时间没有达到配置的 `graceful-restart-time`，也会向 GR Helper 发送 EoR（End-of-Rib）报文来告知邻居自身 GR 已完成，此时 GR Helper 会直接删除陈旧（Stale）路由。
- 这在某些 ADVPN 环境下下可能存在问题：例如 Hub 的 BGP 使用 neighbor-group 模式，且 Hub 与内网路由器还有 BGP GR 关系，当 Hub 发生 HA 切换后，由于 Hub 无法主动向 Spoke 发起 BGP 协商，在 Spoke 再次与 Hub 建立 BGP 并收敛前，Hub 会认为 GR 已经完成并提前结束 GR，从而导致内网路由器无法保持 Spoke 的相关路由。
- 如果想让 GR Restarter 只在 `graceful-restart-time` 到期时才向 GR Helper 发送 EoR（End-of-Rib），则需要开启如下命令：

```bash
config router bgp
    set graceful-end-on-timer enable
end
```

```bash
config router bgp
    set as 65001
    set router-id 202.103.12.1
    set graceful-restart enable
    set graceful-restart-time 120
    set graceful-stalepath-time 360
    set graceful-update-delay 120
    config neighbor
        edit "202.103.12.2"
            set capability-graceful-restart enable
            set next-hop-self-rr enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-reflector-client enable
        next
        edit "202.103.13.2"
            set capability-graceful-restart enable
            set next-hop-self-rr enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-reflector-client enable
        next
    end
end
```

5. 配置 FW3 的 BGP，宣告内网网络，路由器作为 GR-Helper 也需要开启 GR 功能，计时器配置均保持默认。

```bash
config router bgp
    set as 65001
    set router-id 202.103.12.2
    set graceful-restart enable
    config neighbor
        edit "202.103.12.1"
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

6. 配置 FW4 的 BGP，宣告内网网络，路由器作为 GR-Helper 也需要开启 GR 功能，计时器配置均保持默认。

```bash
config router bgp
    set as 65001
    set router-id 202.103.13.2
    set graceful-restart enable
    config neighbor
        edit "202.103.13.1"
            set capability-graceful-restart enable
            set soft-reconfiguration enable
```

```
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

7. 配置 FW1/FW2 HA 配置中的 route-ttl 为 360s（默认 10s），在此时间内，新的主机需要在路由 Kernel 表中维持从原主机学习到的 BGP 路由，直至新的 BGP 邻居形成，并完成 BGP 路由学习。

> **重要**
>
> 重要步骤，防止 HA 切换后，新主机建立新的 BGP 邻居前，路由消失导致业务中断。

```bash
config system ha
    set route-ttl 360
end
```

结果验证

1. 初始状态下，FW1 为主设备，FW2 为备设备。

```bash
FW1 # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:42869 b:39035389
        traffic.total = s:0 p:42867 b:39033889
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=7 Debug=0
nvcluster=1, ses_pickup=0, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM23000175:      Primary, serialno_prio=1, usr_priority=128, hostname=FW1
FGVM08TM23000176:    Secondary, serialno_prio=0, usr_priority=100, hostname=FW2

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.2, primary_id=0
FGVM08TM23000175:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM23000176:    Secondary, ha_prio/o_ha_prio=1/1
```

2. 初始状态下查看 FW3 的 BGP 邻居状态与路由信息，邻居状态为已建立状态，BGP 路由学习正常。

```bash
FW3 # get router info bgp summary

VRF 0 BGP router identifier 202.103.12.2, local AS number 65001
BGP table version is 3
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.1 4      65001    1229    1229        2    0    0 16:55:54        1

Total number of neighbors 1

FW3 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 202.103.12.1 (recursive is directly connected, port2), 00:02:15, [1/0]
```

3. 初始状态下查看 FW4 的 BGP 邻居状态与路由信息，邻居状态为已建立状态，BGP 路由学习正常。

```bash
FW4 # get router info bgp summary

VRF 0 BGP router identifier 202.103.13.2, local AS number 65001
BGP table version is 3
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.13.1 4      65001    1232    1233        2    0    0 17:01:22        1

Total number of neighbors 1

FW4 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 202.103.13.1 (recursive is directly connected, port2), 00:03:00, [1/0]
```

4. 初始状态下查看 FW1 的 BGP 邻居状态与路由信息，邻居状态为已建立，BGP 路由学习正常。

```bash
FW1 # get router info bgp summary

VRF 0 BGP router identifier 202.103.12.1, local AS number 65001
BGP table version is 1
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.2 4      65001    1232    1241        1    0    0 17:03:12        1
202.103.13.2 4      65001    1233    1238        1    0    0 17:03:11        1

Total number of neighbors 2

FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 202.103.12.2 (recursive is directly connected, port2), 17:02:56, [1/0]
B       10.10.2.0/24 [200/0] via 202.103.13.2 (recursive is directly connected, port3), 17:02:55, [1/0]
```

5. 使 FW1 和 FW2 发生 HA 主备倒换，此时 FW2 成为 HA 主设备。

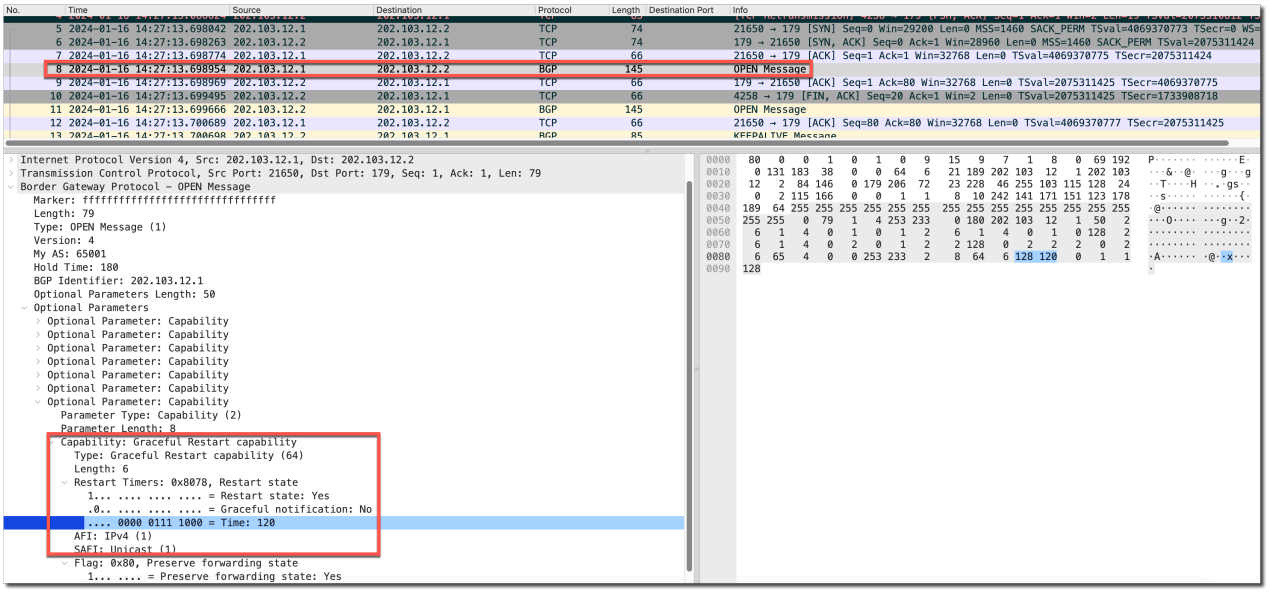
```bash
FW2 # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:6233 b:4732229
        traffic.total = s:0 p:6286 b:4735596
        activity.ha_id_changes = 4
        activity.fdb  = c:0 q:0

Model=80008, Mode=2 Group=7 Debug=0
nvcluster=1, ses_pickup=0, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM08TM23000176:      Primary, serialno_prio=0, usr_priority=100, hostname=FW2
FGVM08TM23000175:    Secondary, serialno_prio=1, usr_priority=128, hostname=FW1

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.1, primary_id=0
FGVM08TM23000176:      Primary, ha_prio/o_ha_prio=0/0
FGVM08TM23000175:    Secondary, ha_prio/o_ha_prio=1/1
```

6. 查看 FW2 的 BGP Debug 信息，FW2 成为 HA 主后，向 BGP 邻居发送 BGP OPEN 消息，其中包含 Cap GR，Restarting 时间为 120s。

```bash
FW2 # diagnose ip router bgp level info
FW2 # diagnose ip router bgp all enable
FW2 # diagnose debug console time enable
......
2024-01-16 14:53:33 BGP: 202.103.12.2-Outgoing [ENCODE] Open: Ver 4 MyAS 65001 Holdtime 180
......
2024-01-16 14:53:33 BGP: 202.103.12.2-Outgoing [DECODE] Open Opt: Option Type 2, Option Len 8
2024-01-16 14:53:33 BGP: 202.103.12.2-Outgoing [DECODE] Open Cap: Cap Code 64, Cap Len 6
2024-01-16 14:53:33 BGP: 202.103.12.2-Outgoing [DECODE] Cap GR: Restart Flag Off, Restart Time 120
```

```
diagnose ip2024-01-16 14:53:33 BGP: 202.103.12.2-Outgoing [DECODE] Cap GR: AFI/SAFI 1/1 Fwd-state Flag 1, action: Set
......
2024-01-16 14:53:34 2024-01-16 14:53:34 BGP: 202.103.12.2-Outgoing [GRST] GRST Init Announce:GRST Defer Announce
Timer(120) Started
id=20300 msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 202.103.12.2 Up "
......
2024-01-16 14:53:59 BGP: 202.103.12.2-Outgoing [FSM] Update: IPv4 Unicast End-Of-Rib Marker Received
2024-01-16 14:53:59 BGP: 202.103.12.2-Outgoing [FSM] Process End-of-RIB: Received for afi/safi: 1/1
```

7. 该报文格式如下图所示。



8. 在 FW2 进入 GR 状态后，查看 FW3 的 BGP 邻居状态与 BGP 路由状态，FW3 进入 GR Helper 状态，开始与 FW2 建立新的 BGP 邻居，之前已学习到的 BGP 路由未消失，被打上 S（Stale）标记。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # get router info bgp summary

VRF 0 BGP router identifier 202.103.12.2, local AS number 65001
BGP table version is 10
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.1 4      65001     101     104        0    0    0    never OpenConfirm

Total number of neighbors 1

FW3 # get router info bgp network
VRF 0 BGP table version is 7, local router ID is 202.103.12.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                           100  32768        0 i <-/1>
S>i10.10.2.0/24     202.103.12.1    0                 100      0        0 i <-/1>

Total number of prefixes 2

FW3 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 202.103.12.1 (recursive is directly connected, port2), 00:01:06, [1/0]
```

9. 在 FW3 上查看 BGP 的 Debug 信息，FW3 与 FW1 的 BGP 连接断开后，进入 GR Helper 模式，GR 倒计时 120s。（FW4 也是一样的状态，这里就不再赘述）

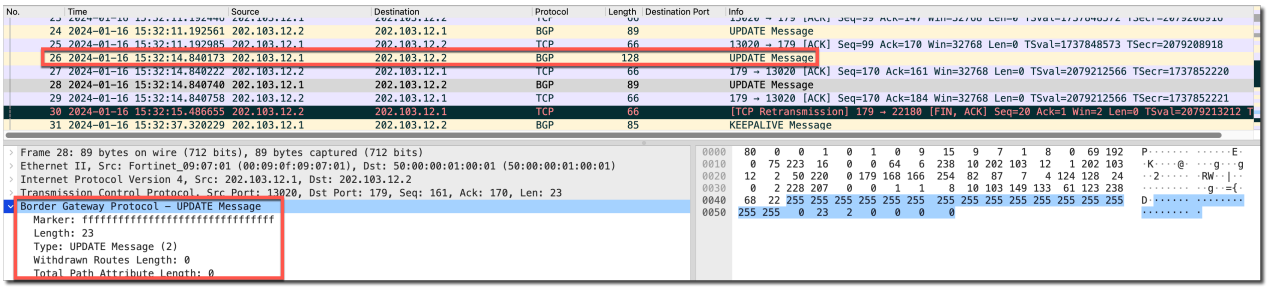
```bash
FW3 # diagnose ip router bgp level info
FW3 # diagnose ip router bgp all enable
FW3 # diagnose debug console time enable

2024-01-16 15:13:53 BGP: [NETWORK] Accept Thread: Incoming conn from host 202.103.12.1 (FD=26 VRF=0)
2024-01-16 15:13:53 BGP: 202.103.12.1-Outgoing [FSM] State: Established Event: 14
......
2024-01-16 15:14:22 BGP: 202.103.12.1-Outgoing [RIB] Update: Received Prefix 10.10.2.0/24 path_id 0
......
```

10. 等待 FW2 与 FW3 的 BGP 邻居建立完成后，FW2 的 GR 完成，FW3 退出 GR Helper 状态，期间被打上 S 标记的旧 BGP 路由没有消失，重新和 FW2 建立 BGP 邻居并学习 BGP 路由后，S 标记变为 * 标记。（FW4 也是一样的状态，这里就不再赘述）

```bash
FW3 # get router info bgp summary

VRF 0 BGP router identifier 202.103.12.2, local AS number 65001
BGP table version is 7
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.1 4      65001      79      81        0    0    0 00:00:05        1

FW3 # get router info bgp network
VRF 0 BGP table version is 12, local router ID is 202.103.12.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                           100  32768        0 i ←/1>
*>i10.10.2.0/24     202.103.12.1    0                 100      0        0 i ←/1>

Total number of prefixes 2

FW3 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 202.103.12.1 (recursive is directly connected, port2), 00:01:06, [1/0]
```

11. 在 FW3 上查看 BGP 的 Debug 信息，FW3 从 FW2 完成了新的 BGP 路由学习后，向 FW2 发送了 End-of-RIB 报文，报文格式如下（空的 BGP Update 报文），通知其路由已经学习完成。

```bash
2024-01-16 15:14:22 BGP: VRF 0 NSM announce: 10.10.2.0/24
2024-01-16 15:14:22 BGP: 202.103.12.1-Outgoing [FSM] Update: IPv4 Unicast End-Of-Rib Marker Received
2024-01-16 15:14:22 BGP: 202.103.12.1-Outgoing [FSM] Process End-of-RIB: Received for afi/safi: 1/1
```



12. 在 FW3 上查看 BGP 邻居状态，其中包含 GR 相关信息，如邻居的 GR 时间为 120s。

```bash
FW3 # get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 202.103.12.1, remote AS 65001, local AS 65001, internal link
  BGP version 4, remote router ID 202.103.12.1
  BGP state = Established, up for 00:04:58
  Last read 00:00:02, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
 ......
 For address family: IPv4 Unicast
  BGP table version 13, neighbor version 12
  Index 1, Offset 0, Mask 0x2
```

```
  AF-dependant capabilities:
    Graceful restart: advertised, received, negotiated
......
 Connections established 13; dropped 12
 Graceful-restart Status:
  Remote restart-time is 120 sec
......
```

3. GR 期间，FW2 从 FW1 重启前同步的路由一直存在（route-ttl 360，prio = 2164260865）。

```bash
FW2 # get router info kernel | grep 10.10.
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2
dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2
dev=5(port3)
```

4. GR 结束后，学到了新的 BGP 路由（prio = 1）。

```bash
FW2 # get router info kernel | grep 10.10.
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2 dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.1.0/24 pref=0.0.0.0 gwy=202.103.12.2
dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2 dev=5(port3)
tab=254 vf=0 scope=0 type=1 proto=30 prio=2164260865 0.0.0.0/0.0.0.0/0→10.10.2.0/24 pref=0.0.0.0 gwy=202.103.13.2
dev=5(port3)
```

5. 整个 HA 切换引起的 GR 期间，所有设备路由转发表未发生实际变化，流量不会中断。

GR 与 BFD 的使用关系

> **重要**
>
> BGP GR 或 OSPF GR 不建议与 BFD 同时使用。

GR 用于 FortiGate 通知邻居自己的路由进程将重启，并且邻居从 FortiGate 学到的路由应在重启期间保留。但 BFD 检测到故障时会将动态路由邻居重置。这种操作上的差异使得在 GR 的同时使用 BFD 会导致功能上的冲突，可能导致 GR 期间流量中断。

业界通用的做法是在使用 GR 时关闭 BFD。例如在 FortiGate 和 GCP（Google 云平台）之间配置动态路由 + GR + BFD 时：

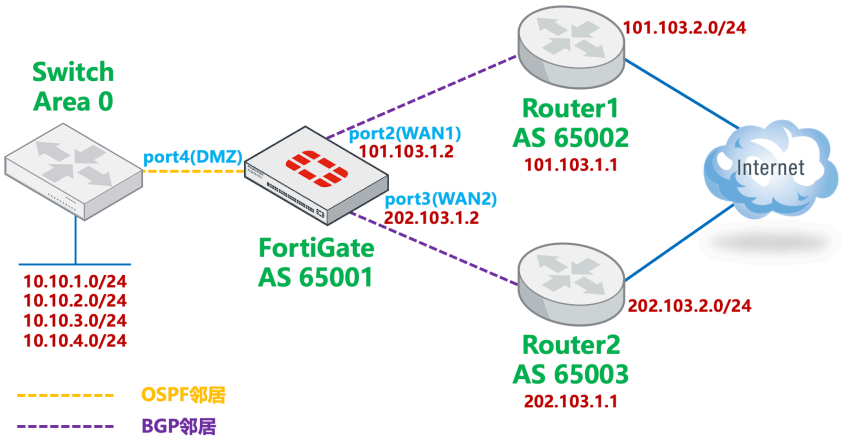
- GCP 将关闭 BFD 并发送 AdminDown（BFD 事件）到 FortiGate（参考 https://cloud.google.com/network-connectivity/docs/router/concepts/bfd#graceful-restart-and-bfd ↗ ）。
- FortiGate 不会因为 AdminDown 类型的 BFD 通知重置 BGP 连接（参考 https://handbook.fortinet.com.cn/routing/bfd/bfd_and_bgp.html#注意事项 ↗ ）。

BGP 路由过滤

网络需求

在 BGP 路由中通过 prefix-list + route-map 实现路由过滤功能。

网络拓扑



1. OSPF 邻居：

   - FortiGate 与内网 Switch 建立 OSPF 邻居。
   - Switch 发布内网网段路由，宣告给 FortiGate。

2. BGP 邻居：

   - FortiGate 与 Router1 建立 eBGP 邻居。
   - FortiGate 与 Router2 建立 eBGP 邻居。

3. 路由学习：

   - FortiGate 将学习到的 OSPF 路由重发布到两个 eBGP 邻居。
   - FortiGate 将从 Router1 和 Router2 学习到的 BGP 路由重发布到 OSPF 邻居。
   - Router1 和 Router2 都向 FortiGate 发布默认路由。
   - FortiGate 从 Router1 学习到的 BGP 路由不能发布给 Router2。
   - FortiGate 从 Router1 和 Router2 学习到的 BGP 默认路由不能发布给 Router2 和 Router1。

配置步骤

基础配置

1. FortiGate 的接口配置。

```bash
config system interface
    edit "port2"
        set vdom "root"
        set ip 101.103.1.2 255.255.255.0
        set allowaccess ping
        set alias "WAN1"
    next
    edit "port3"
        set vdom "root"
        set ip 202.103.1.2 255.255.255.0
        set allowaccess ping
        set alias "WAN2"
    next
    edit "port4"
        set vdom "root"
```

```
        set ip 10.10.254.1 255.255.255.0
        set allowaccess ping
        set alias "DMZ"
    next
end
```

2. 配置安全策略，放通 FortiGate、Switch、Router 间的流量，按需配置其他安全策略（略）。

OSPF（FortiGate to Switch）

配置 FortiGate 的 OSPF，将 BGP 路由重发布至 OSPF。

```bash
config router ospf
    set router-id 10.10.254.1
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.10.254.0 255.255.255.0
        next
    end
    config redistribute "bgp"
        set status enable
    end
end
```

BGP（FortiGate to Router）

1. 在 FortiGate 上配置 2 个 prefix-list：

   - 一个 prefix-list 匹配 Router1 宣告的 BGP 路由 101.103.2.0/24，不设置 ge 和 le（表示严格匹配），动作为 permit（表示匹配）。
   - 另一个 prefix-list 匹配 Router1 和 Router2 宣告的默认路由，不设置 ge 和 le（表示严格匹配），动作为 permit（表示匹配）。

```bash
config router prefix-list
    edit "Block_101.103.2.0/24"
        config rule
            edit 1
                set prefix 101.103.2.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
    edit "Block_0.0.0.0/0"
        config rule
            edit 1
                set prefix 0.0.0.0 0.0.0.0
                unset ge
                unset le
            next
        end
    next
end
```

2. 在 FortiGate 上配置 Route-map 引用 prefix-list，FortiGate 从 Router1 学习到的 BGP 细化路由不能发布给 Router2，FortiGate 从 Router1 和 Router2 学习到的 BGP 默认路由不能发布给 Router2 和 Router1，其他路由条目可以正常宣告。

   - 序号 10：匹配条件为 Router1 宣告网段的 prefix-list：Block_101.103.2.0/24，动作为 deny（表示拒绝）。
   - 序号 20：匹配条件为 prefix-list：Block_0.0.0.0/0，动作为 deny（表示拒绝）。
   - 序号 100：匹配条件为 any，即所有路由，动作为 permit（表示允许）。

```bash
config router route-map
    edit "RMP_out"
        config rule
            edit 10
                set action deny
                set match-ip-address "Block_101.103.2.0/24"
            next
```

```
            edit 20
                set action deny
                set match-ip-address "Block_0.0.0.0/0"
            next
            edit 100
            next
        end
    next
end
```

3. 在 FortiGate 上配置 BGP，在邻居 Router1 和 Router2 上分别使用 route-map-out 调用上步创建的 route-map，重发布 OSPF 路由，开启 ebgp-multipath。

```bash
config router bgp
    set as 65001
    set router-id 101.103.1.2
    set ebgp-multipath enable
    config neighbor
        edit "101.103.1.1"
            set soft-reconfiguration enable
            set route-map-out "RMP_out"
            set remote-as 65002
        next
        edit "202.103.1.1"
            set soft-reconfiguration enable
            set route-map-out "RMP_out"
            set remote-as 65003
        next
    end
    config redistribute "ospf"
        set status enable
    end
end
```

4. 在 Router1（使用 FortiGate 模拟）上配置 BGP，向 FortiGate 发布自身网段和默认路由。

```bash
config router bgp
    set as 65002
    set router-id 101.103.1.1
    config neighbor
        edit "101.103.1.2"
            set capability-default-originate enable
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 101.103.2.0 255.255.255.0
        next
    end
end
```

5. 在 Router2（使用 FortiGate 模拟）上配置 BGP，向 FortiGate 发布自身网段和默认路由。

```bash
config router bgp
    set as 65003
    set router-id 202.103.1.1
    config neighbor
        edit "202.103.1.2"
            set capability-default-originate enable
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
```

```
        edit 1
            set prefix 202.103.2.0 255.255.255.0
        next
    end
end
```

结果验证

FortiGate

1. 在 FortiGate 上查看 OSPF 邻居状态与 OSPF 路由，可以看到和 Switch 建立了 OSPF 邻居，并学习到了 Switch 发布的 4 条 OSPF 路由。

```bash
FortiGate # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.254.2       1   Full/DR         00:00:36    10.10.254.2     port4

FortiGate # get router info routing-table ospf
Routing table for VRF=0
O       10.10.1.0/24 [110/2] via 10.10.254.2, port4, 01:23:49, [1/0]
O       10.10.2.0/24 [110/2] via 10.10.254.2, port4, 01:23:49, [1/0]
O       10.10.3.0/24 [110/2] via 10.10.254.2, port4, 01:23:49, [1/0]
O       10.10.4.0/24 [110/2] via 10.10.254.2, port4, 01:23:49, [1/0]
```

2. 在 FortiGate 上查看 BGP 邻居状态，可以看到和 Router1、Router2 分别建立了 BGP 邻居，并学习到路由。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors` 。

```bash
FortiGate # get router info bgp summary

VRF 0 BGP router identifier 101.103.1.2, local AS number 65001
BGP table version is 5
3 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
101.103.1.1 4      65002     101     117        5    0    0 01:25:48        2
202.103.1.1 4      65003     106     115        5    0    0 01:17:25        2

Total number of neighbors 2
```

3. 在 FortiGate 上查看从邻居 Router1 和 Router2 收到的路由条目，可以看到收到了 Router1/Router2 发布的自身网段路由和默认路由。

```bash
FortiGate # get router info bgp neighbors 101.103.1.1 received-routes
VRF 0 BGP table version is 7, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        101.103.1.1                             0        0 65002 i ←/→
*> 101.103.2.0/24   101.103.1.1                             0        0 65002 i ←/→

Total number of prefixes 2

FortiGate # get router info bgp neighbors 202.103.1.1 received-routes
VRF 0 BGP table version is 7, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        202.103.1.1                             0        0 65003 i ←/→
*> 202.103.2.0      202.103.1.1                             0        0 65003 i ←/→

Total number of prefixes 2
```

4. 在 FortiGate 上查看向邻居 Router1、Router2 发布的路由条目，可以看到 route-map-out 生效，没有将学习到的 Router1 的 BGP 路由发布给 Router2（但将学习到的 Router2 的 BGP 路由发布给了 Router1），没有发布从 Router1、Router2 学习到的默认路由，重发布了 OSPF 路由。

```bash
FortiGate # get router info bgp neighbors 101.103.1.1 advertised-routes
VRF 0 BGP table version is 7, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     101.103.1.2     2                  32768        0 ? ←/→
*> 10.10.2.0/24     101.103.1.2     2                  32768        0 ? ←/→
*> 10.10.3.0/24     101.103.1.2     2                  32768        0 ? ←/→
*> 10.10.4.0/24     101.103.1.2     2                  32768        0 ? ←/→
*> 202.103.2.0      101.103.1.2                            0        0 65003 i ←/→

Total number of prefixes 5

FortiGate # get router info bgp neighbors 202.103.1.1 advertised-routes
VRF 0 BGP table version is 7, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     202.103.1.2     2                  32768        0 ? ←/→
*> 10.10.2.0/24     202.103.1.2     2                  32768        0 ? ←/→
*> 10.10.3.0/24     202.103.1.2     2                  32768        0 ? ←/→
*> 10.10.4.0/24     202.103.1.2     2                  32768        0 ? ←/→

Total number of prefixes 4
```

5. 在 FortiGate 上查看 BGP 路由数据库，可以看到 eBGP 路由的路径（AS-Path）信息。

```bash
FortiGate # get router info bgp network
VRF 0 BGP table version is 7, local router ID is 101.103.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*  0.0.0.0/0        202.103.1.1     0                      0        0 65003 i ←/→
*>                  101.103.1.1     0                      0        0 65002 i ←/1>
*> 10.10.1.0/24     10.10.254.2     2                  32768        0 ? ←/1>
*> 10.10.2.0/24     10.10.254.2     2                  32768        0 ? ←/1>
*> 10.10.3.0/24     10.10.254.2     2                  32768        0 ? ←/1>
*> 10.10.4.0/24     10.10.254.2     2                  32768        0 ? ←/1>
*> 101.103.2.0/24   101.103.1.1     0                      0        0 65002 i ←/1>
*> 202.103.2.0      202.103.1.1     0                      0        0 65003 i ←/1>

Total number of prefixes 7
```

6. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```bash
FortiGate # get router info bgp network 0.0.0.0
VRF 0 BGP routing table entry for 0.0.0.0/0
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65003
    202.103.1.1 from 202.103.1.1 (202.103.1.1)
      Origin IGP metric 0, localpref 100, valid, external
      Last update: Wed Jan 17 11:32:12 2024

  Original VRF 0
  65002
    101.103.1.1 from 101.103.1.1 (101.103.1.1)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Wed Jan 17 11:32:12 2024
```

7. 查看 FortiGate 的路由表中的 BGP 路由，由于开启了 ebgp-multipath，从 Router1 和 Router2 学习到的默认路由在路由表中负载。

```bash
FortiGate # get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [20/0] via 101.103.1.1 (recursive is directly connected, port2), 01:50:31, [1/0]
                 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 01:50:31, [1/0]
B       101.103.2.0/24 [20/0] via 101.103.1.1 (recursive is directly connected, port2), 00:09:04, [1/0]
B       202.103.2.0/24 [20/0] via 202.103.1.1 (recursive is directly connected, port3), 00:09:08, [1/0]
```

Router1

1. 在 Router1 上查看 BGP 邻居状态，可以看到和 FortiGate 建立了 BGP 邻居，并学习到路由。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors` 。

```bash
Router1 # get router info bgp summary

VRF 0 BGP router identifier 101.103.1.1, local AS number 65002
BGP table version is 3
3 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
101.103.1.2 4      65001     126     112        2    0    0 01:34:48        5

Total number of neighbors 1
```

2. 在 Router1 上查看从 BGP 邻居收到的路由条目，可以看到学习到了 FortiGate 重发布的 OSPF 路由，以及 FortiGate 从 Router2 学习到的 BGP 路由。

```bash
Router1 # get router info bgp neighbors 101.103.1.2 received-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     101.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.2.0/24     101.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.3.0/24     101.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.4.0/24     101.103.1.2     2                      0        0 65001 ? ←/→
*> 202.103.2.0      101.103.1.2                            0        0 65001 65003 i ←/→

Total number of prefixes 5
```

3. 在 Router1 上查看向邻居发布的路由条目，包含自身网段路由和默认路由。

```bash
Router1 # get router info bgp neighbors 101.103.1.2 advertised-routes
VRF 0 BGP table version is 3, local router ID is 101.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        101.103.1.1                   100  32768        0 i ←/→
*> 101.103.2.0/24   101.103.1.1                   100  32768        0 i ←/→

Total number of prefixes 2
```

4. 在 Router1 上查看 BGP 路由数据库，可以看到 eBGP 路由的路径（AS-Path）信息。

```bash
Router1 # get router info bgp network
VRF 0 BGP table version is 3, local router ID is 101.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     101.103.1.2     2                      0        0 65001 ? ←/1>
*> 10.10.2.0/24     101.103.1.2     2                      0        0 65001 ? ←/1>
```

```
*> 10.10.3.0/24     101.103.1.2     2                      0        0 65001 ? ←/1>
*> 10.10.4.0/24     101.103.1.2     2                      0        0 65001 ? ←/1>
*> 101.103.2.0/24   0.0.0.0                       100  32768        0 i ←/1>
*> 202.103.2.0      101.103.1.2     0                      0        0 65001 65003 i ←/1>

Total number of prefixes 6
```

5. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```
Router1 # get router info bgp network 202.103.2.0                                                    bash
VRF 0 BGP routing table entry for 202.103.2.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65001 65003
    101.103.1.2 from 101.103.1.2 (101.103.1.2)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Wed Jan 17 11:32:27 2024
```

6. 查看 Router1 的路由表中的 BGP 路由，可以看到 Router1 从 FortiGate 学习到的路由均已被放入路由表（包含 FortiGate 重发布的 OSPF 路由，以及 FortiGate 从 Router2 学习到的 BGP 路由），eBGP 路由的 distance 为 20。

```
Router1 # get router info routing-table bgp                                                          bash
Routing table for VRF=0
B       10.10.1.0/24 [20/2] via 101.103.1.2 (recursive is directly connected, port2), 01:59:33, [1/0]
B       10.10.2.0/24 [20/2] via 101.103.1.2 (recursive is directly connected, port2), 01:59:33, [1/0]
B       10.10.3.0/24 [20/2] via 101.103.1.2 (recursive is directly connected, port2), 01:59:33, [1/0]
B       10.10.4.0/24 [20/2] via 101.103.1.2 (recursive is directly connected, port2), 01:59:33, [1/0]
B       202.103.2.0/24 [20/0] via 101.103.1.2 (recursive is directly connected, port2), 00:10:12, [1/0]
```

Router2

1. 在 Router2 上查看 BGP 邻居状态，可以看到和 FortiGate 建立了 BGP 邻居，并学习到路由。

> **重要**
>
> 查看 BGP 邻居的详细信息，请使用 `get router info bgp neighbors`。

```
Router2 # get router info bgp summary                                                                bash

VRF 0 BGP router identifier 202.103.1.1, local AS number 65003
BGP table version is 4
2 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.1.2 4      65001     156     147        3    0    0 01:52:16        4

Total number of neighbors 1
```

2. 在 Router2 上查看从 BGP 邻居收到的路由条目，可以看到学习到了 FortiGate 重发布的 OSPF 路由，没有 FortiGate 从 Router1 学习到的 BGP 路由。

```
Router2 # get router info bgp neighbors 202.103.1.2 received-routes                                  bash
VRF 0 BGP table version is 4, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     202.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.2.0/24     202.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.3.0/24     202.103.1.2     2                      0        0 65001 ? ←/→
*> 10.10.4.0/24     202.103.1.2     2                      0        0 65001 ? ←/→

Total number of prefixes 4
```

3. 在 Router2 上查看向邻居发布的路由条目，包含自身网段路由和默认路由。

```bash
Router2 # get router info bgp neighbors 202.103.1.2 advertised-routes
VRF 0 BGP table version is 4, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 0.0.0.0/0        202.103.1.1                    100  32768        0 i ←/→
*> 202.103.2.0      202.103.1.1                    100  32768        0 i ←/→

Total number of prefixes 2
```

4. 在 Router2 上查看 BGP 路由数据库，可以看到 eBGP 路由的路径（AS-Path）信息。

```bash
Router2 # get router info bgp network
VRF 0 BGP table version is 4, local router ID is 202.103.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     202.103.1.2     2                      0        0 65001 ? ←/1>
*> 10.10.2.0/24     202.103.1.2     2                      0        0 65001 ? ←/1>
*> 10.10.3.0/24     202.103.1.2     2                      0        0 65001 ? ←/1>
*> 10.10.4.0/24     202.103.1.2     2                      0        0 65001 ? ←/1>
*> 202.103.2.0      0.0.0.0                        100  32768        0 i ←/1>

Total number of prefixes 5
```

5. 查看指定的某个 BGP 路由详细信息可以在该命令后加相应网段。

```bash
Router2 # get router info bgp network 10.10.1.0
VRF 0 BGP routing table entry for 10.10.1.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65001
    202.103.1.2 from 202.103.1.2 (101.103.1.2)
      Origin incomplete metric 2, localpref 100, valid, external, best
      Last update: Wed Jan 17 11:32:27 2024
```

6. 查看 Router2 的路由表中的 BGP 路由，包含 FortiGate 重发布的 OSPF 路由，没有 FortiGate 从 Router1 学习到的路由，distance 为 20（eBGP 路由）。

```bash
Router2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [20/2] via 202.103.1.2 (recursive is directly connected, port2), 01:43:33, [1/0]
B       10.10.2.0/24 [20/2] via 202.103.1.2 (recursive is directly connected, port2), 01:43:33, [1/0]
B       10.10.3.0/24 [20/2] via 202.103.1.2 (recursive is directly connected, port2), 01:43:33, [1/0]
B       10.10.4.0/24 [20/2] via 202.103.1.2 (recursive is directly connected, port2), 01:43:33, [1/0]
```

Switch

在 Switch 上查看路由表，可以看到 Switch 学习到了 FortiGate 在 OSPF 中重发布的 BGP 路由。

```bash
Switch # get router info routing-table ospf
Routing table for VRF=0
O E2    101.103.2.0/24 [110/10] via 10.10.254.1, port2, 00:11:32, [1/0]
O E2    202.103.2.0/24 [110/10] via 10.10.254.1, port2, 00:11:36, [1/0]
```

BGP 路由刷新

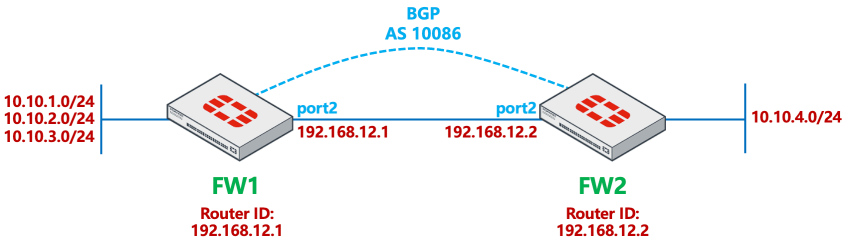
功能介绍

默认的 BGP 配置下，当 BGP 邻居配置中引用的路由策略（如 route-map、distribute-list 等）修改了属性或过滤的路由条目时，BGP 路由表并不会立即更新，需要执行路由刷新来重新获取 BGP 路由，并进行路由过滤/修改。

FortiGate 路由刷新的方法有以下 3 种：

- 方法 1 - 使用 BGP route-refresh
- 方法 2 - 使用 soft-reconfiguration
- 方法 3 - 重置 BGP 邻居

网络拓扑



1. FW1 与 FW2 建立 iBGP 邻居。
2. 初始状态下，FW1 发布自己的路由 10.10.1.0/24、10.10.2.0/24、10.10.3.0/24 给 FW2，FW1 与 FW2 都不开启 `soft-reconfiguration` 功能。
3. 初始状态下，FW2 的 BGP 入方向引用 route-map，只允许学习 FW1 发送的路由 10.10.1.0/24、10.10.2.0/24。

配置步骤

1. 基础网络配置（略）。

2. 配置 FW1 的 BGP，与 FW2 建立 iBGP 邻居，并发布自己的路由 10.10.1.0/24、10.10.2.0/24、10.10.3.0/24 给 FW2。

```bash
config router bgp
    set as 10086
    set router-id 192.168.12.1
    config neighbor
        edit "192.168.12.2"
            set remote-as 10086
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.2.0 255.255.255.0
        next
        edit 3
            set prefix 10.10.3.0 255.255.255.0
    end
end
```

3. 在 FW2 上配置 prefix-list，内容为 FW1 的发布的 BGP 路由 10.10.1.0/24、10.10.2.0/24。

```bash
config router prefix-list
    edit "Prefix_FW1"
        config rule
            edit 1
                set prefix 10.10.1.0 255.255.255.0
                unset ge
                unset le
            next
            edit 2
```

```
                set prefix 10.10.2.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end
```

4. 在 FW2 上配置 route-map，允许 Prefix `Prefix_FW1` 中的 BGP 路由，拒绝其他路由。

```bash
config router route-map
    edit "RMP_from_FW1"
        config rule
            edit 1
                set match-ip-address "Prefix_FW1"
            next
            edit 2
                set action deny
            next
        end
    next
end
```

5. 在 FW2 上配置 BGP，与 FW2 建立 iBGP 邻居，并发布自己的路由 10.10.4.0/24 给 FW2。在入方向引用步骤 4 配置的 route-map，只允许接收 FW1 发布的 BGP 路由 10.10.1.0/24、10.10.2.0/24。

```bash
config router bgp
    set as 10086
    set router-id 192.168.12.2
    config neighbor
        edit "192.168.12.1"
            set remote-as 10086
            set route-map-in "RMP_from_FW1"
        next
    end
    config network
        edit 1
            set prefix 10.10.4.0 255.255.255.0
        next
    end
end
```

6. 在 FW1 上查看 BGP 路由学习情况，可以正确学习到 FW2 宣告的路由 10.10.4.0/24。

```bash
FW1 # get router info bgp summary

VRF 0 BGP router identifier 192.168.12.1, local AS number 10086
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.2 4      10086      22      23        1    0    0 00:17:28        1
Total number of neighbors 1

FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.4.0/24 [200/0] via 192.168.12.2 (recursive is directly connected, port2), 00:17:13, [1/0]
```

7. 在 FW1 上查看通告给 FW2 的 BGP 路由，与配置一致。

```bash
FW1 # get router info bgp neighbors 192.168.12.2 advertised-routes
VRF 0 BGP table version is 6, local router ID is 192.168.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
```

```
*>i10.10.1.0/24     192.168.12.1                100  32768        0 i ←/→
*>i10.10.2.0/24     192.168.12.1                100  32768        0 i ←/→
*>i10.10.3.0/24     192.168.12.1                100  32768        0 i ←/→

Total number of prefixes 3
```

8. 在 FW2 上查看 BGP 路由学习情况，可以正确学习到 FW1 发布的 BGP 路由 10.10.1.0/24、10.10.2.0/24，而 10.10.3.0/24 被 route-map 过滤掉。

```bash
FW2 # get router info bgp summary

VRF 0 BGP router identifier 192.168.12.2, local AS number 10086
BGP table version is 1
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.1 4      10086       3       4        1    0    0 00:00:30        2
Total number of neighbors 1

FW2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:00:07, [1/0]
B       10.10.2.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:00:07, [1/0]
```

9. 在 FW2 的 route-map 引用的 prefix 中增加 10.10.3.0/24 的路由，这样 route-map 就开始允许接收 10.10.3.0/24 的 BGP 路由。

```bash
config router prefix-list
    edit "Prefix_FW1"
        config rule
            edit 3
                set prefix 10.10.3.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end
```

0. 在 FW2 上查看学习到的 BGP 路由，仍然只有之前学习到的 BGP 路由 10.10.1.0/24、10.10.2.0/24，没有 FW1 发布的 10.10.3.0/24 路由。

```bash
FW2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 02:37:10, [1/0]
B       10.10.2.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 02:37:10, [1/0]
```

1. 如果没有任何操作，路由将一直不会更新。可以使用如下 3 种方法进行路由刷新。

路由刷新方法

方法 1 - 使用 BGP route-refresh

1. 在 FW2 上查看在内存中保存的从 FW1 收到的 BGP 路由，由于没有开启软重配（soft-reconfiguration）功能，内存中不会保存从 FW1 收到的所有路由。

```bash
FW2 # get router info bgp neighbors 192.168.12.1 received-routes
% Inbound soft reconfiguration not enabled
% No prefix for neighbor 192.168.12.1
```

2. 确保 FW1 和 FW2 的 BGP 邻居配置中都开启了 `capability-route-refresh` 功能（默认开启）。

> **重要**
>
> `capability-route-refresh` 用于手动路由刷新时向邻居发送 route-refresh 请求。
>
> 该功能符合 RFC 2918，如果 BGP 邻居设备不是 FortiGate，需要也支持 RFC 2918 才行。

```
config router bgp                                                                    bash
        edit "192.168.12.1"
            set capability-route-refresh enable
        next
    end
end
```

3. 在 FW2 上使用 `execute router clear bgp ip 192.168.12.1 in` 命令进行 BGP 邻居路由入方向的刷新，向 FW1 请求 BGP Update，并重新匹配 route-map 接收路由。等待若干秒后，再次查看 FW2 的 BGP 路由表，可以看到 10.10.3.0/24 已经入表。

> **重要**
>
> 如果 BGP 邻居配置中没有开启 `soft-reconfiguration`，使用 `execute router clear bgp ip 192.168.12.1 soft in`，将会与此步命令（`execute router clear bgp ip 192.168.12.1 in`）执行相同的操作。

```
FW2 # execute router clear bgp ip 192.168.12.1 in                                   bash

FW2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 03:08:47, [1/0]
B       10.10.2.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 03:08:47, [1/0]
B       10.10.3.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:10:43, [1/0]
```

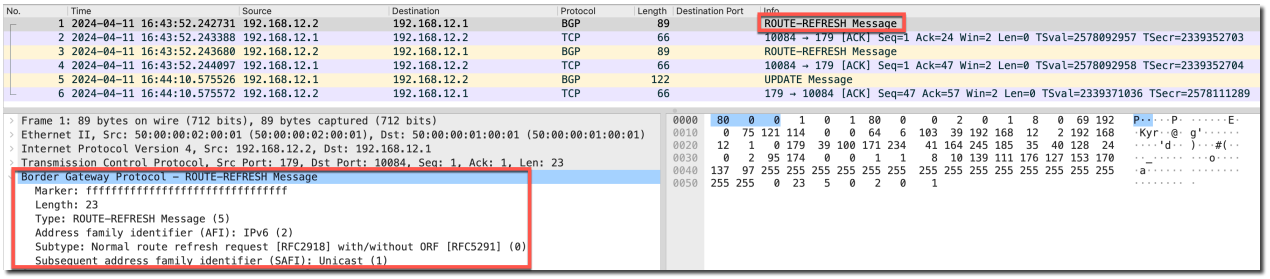
4. FW2 上的 BGP Debug 信息，显示 FW2 向 FW1 发送了 NLRI（Route-Refresh）的更新请求（RFC 2918），收到 FW1 的 BGP Update 后，从中取出 10.10.1.0/24、10.10.2.0/24、10.10.3.0/24 路由条目，10.10.3.0/24 被放入路由表，10.10.1.0/24 和 10.10.2.0/24 由于路由表中已存在而忽略。

```
FW2 # diagnose ip router bgp level info                                             bash
FW2 # diagnose ip router bgp all enable
FW2 # diagnose debug console time enable
FW2 # diagnose debug enable

2024-04-11 14:46:27 BGP: 192.168.12.1-Outgoing [ENCODE] Route-Refresh: RR-Type New MsgSize 23
2024-04-11 14:46:27 BGP: 192.168.12.1-Outgoing [ENCODE] Msg-Hdr: Type 5
2024-04-11 14:46:27 BGP: 192.168.12.1-Outgoing [ENCODE] Route-Refresh: RR-Type New MsgSize 23
2024-04-11 14:46:27 BGP: 192.168.12.1-Outgoing [ENCODE] Msg-Hdr: Type 5
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [DECODE] Update: Starting UPDATE decoding... Bytes To Read (37), msg_size
(37)
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [DECODE] Update: NLRI Len(12)
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [FSM] State: Established Event: 27
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.1.0/24 path_id 0
2024-04-11 14:46:33 BGP: [DAMP] bgp_rfd_rt_update(): Route State: NONE, ret=0
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [RIB] Update: ...duplicate route ignored
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.3.0/24 path_id 0
2024-04-11 14:46:33 BGP: VRF 0 NSM announce: 10.10.3.0/24
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.2.0/24 path_id 0
2024-04-11 14:46:33 BGP: [DAMP] bgp_rfd_rt_update(): Route State: NONE, ret=0
2024-04-11 14:46:33 BGP: 192.168.12.1-Outgoing [RIB] Update: ...duplicate route ignored
```

5. FW1 上的 BGP Debug 信息，显示 FW1 收到了 FW2 的 BGP NLRI 更新请求（Route-Refresh），并向 FW2 发送了 BGP Update。

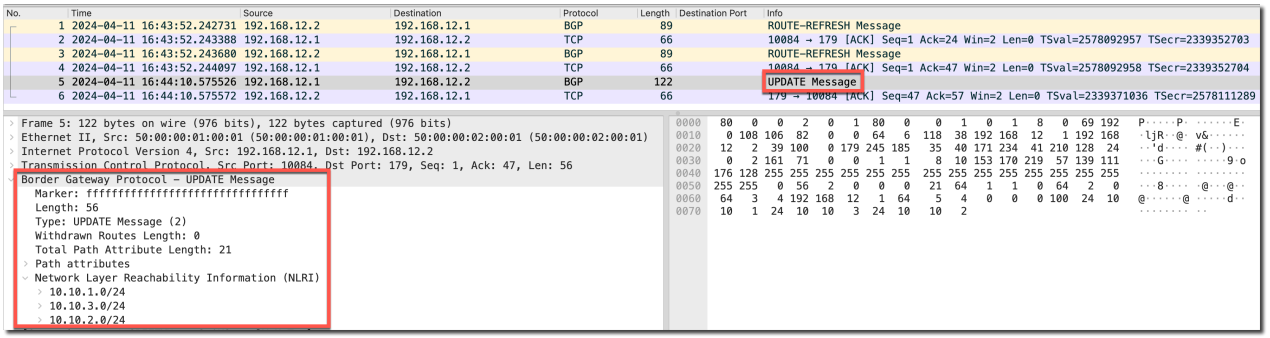
```
FW1 # diagnose ip router bgp level info                                             bash
FW1 # diagnose ip router bgp all enable
FW1 # diagnose debug console time enable
FW1 # diagnose debug enable

2024-04-11 14:46:27 BGP: 192.168.12.2-Outgoing [FSM] State: Established Event: 29
2024-04-11 14:46:27 BGP: 192.168.12.2-Outgoing [DECODE] Msg-Hdr: type 5, length 23
2024-04-11 14:46:27 BGP: 192.168.12.2-Outgoing [DECODE] R-Refresh: AFI/SAFI (1/1)
2024-04-11 14:46:27 BGP: 192.168.12.2-Outgoing [DECODE] PEER_STATUS_ORF_NOT_WAIT_REFRESH set!

2024-04-11 14:46:33 BGP: 192.168.12.2-Outgoing [ENCODE] Msg-Hdr: Type 2
2024-04-11 14:46:33 BGP: 192.168.12.2-Outgoing [ENCODE] Attr IP-Unicast: Tot-attr-len 21
2024-04-11 14:46:33 BGP: 192.168.12.2-Outgoing [ENCODE] Update: Msg #9 Size 56
```

6. FW2 发出的 BGP Route-Rfresh 请求报文的格式如下。



7. FW1 返回的 BGP Update 报文如下，包含自己宣告的 3 条路由，抓包文件请参考 bgp_route_refresh.pcap ↗。

> **重要**
>
> 由于未开启 `soft-reconfiguration`，邻居发布的全量路由不会保存在内存表中。所以使用 `capability-route-refresh` 功能请求对端更新路由，并不会马上得到对端的更新响应，会由对端配置的 `advertisement-interval`（FortiGate 默认 30s）决定。



8. FW2 上查看 BGP capability-route-refresh 功能的开启状态，如下所示，在 BGP 邻居配置中开启 capability-route-refresh 的状态下，FW2 即有 Route refresh 的通告功能，也有 Route refresh 的接受功能。

```bash
FW2 # get router info bgp neighbors 192.168.12.1 | grep refresh
    Route refresh: advertised and received (old and new)
  Route refresh request: received 0, sent 0
```

9. 期间相关 BGP 邻居不会中断，路由表中原有的 BGP 路由没有消失。

方法 2 - 使用 soft-reconfiguration

1. 在 FW2 上的 BGP 中开启 `soft-reconfiguration` 功能（对于 BGP IPv6 路由，该功能为 `soft-reconfiguration6`）。

```bash
config router bgp
    config neighbor
        edit "192.168.12.1"
            set soft-reconfiguration enable
        next
    end
end
```

2. 在 FW2 上查看在内存中保存的从 FW1 收到的 BGP 路由，由于开启了软重配（soft-reconfiguration）功能，内存中会保存从 FW1 收到的所有 BGP 路由（被 route-map 过滤前的路由）。

```bash
FW2 # get router info bgp neighbors 192.168.12.1 received-routes
VRF 0 BGP table version is 2, local router ID is 192.168.12.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     192.168.12.1                100      0         0 i ←/→
*>i10.10.2.0/24     192.168.12.1                100      0         0 i ←/→
*>i10.10.3.0/24     192.168.12.1                100      0         0 i ←/→

Total number of prefixes 3
```

3. 在 FW2 上使用 `execute router clear bgp ip 192.168.12.1 soft in` 命令进行 BGP 邻居路由入方向的刷新，不需要向 FW1 请求 BGP Update，直接从内存中保存的 BGP 路由遍历，并重新匹配 route-map 接收路由。再次查看 FW2 的 BGP 路由表，可以看到 10.10.3.0/24 直接入表（不需要像“方法 1 - 使用 BGP route-refresh”等待若干秒）。

```bash
FW2 # execute router clear bgp ip 192.168.12.1 soft in

FW2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:08:26, [1/0]
B       10.10.2.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:08:26, [1/0]
B       10.10.3.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:03:02, [1/0]
```

4. FW2 上的 BGP Debug 信息，显示 BGP 从内存中读取从 FW1 收到的所有 BGP 路由，从中取出 10.10.1.0/24、10.10.2.0/24、10.10.3.0/24 路由条目，10.10.3.0/24 被放入路由表，10.10.1.0/24 和 10.10.2.0/24 由于路由表中已存在而忽略。

> **重要**
>
> 不需要向 FW1 发送 BGP Route-Refresh 请求报文。

```bash
FW2 # diagnose ip router bgp level info
FW2 # diagnose ip router bgp all enable
FW2 # diagnose debug console time enable
FW2 # diagnose debug enable

2024-04-11 17:22:54 BGP: [RIB] Scanning BGP RIB for VRF 0...
2024-04-11 17:22:54 BGP: [NSM] Verified NH 192.168.12.1 with NSM
2024-04-11 17:22:58 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.1.0/24 path_id 0
2024-04-11 17:22:58 BGP: [DAMP] bgp_rfd_rt_update(): Route State: NONE, ret=0
2024-04-11 17:22:58 BGP: 192.168.12.1-Outgoing [RIB] Update: ...duplicate route ignored
2024-04-11 17:22:58 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.2.0/24 path_id 0
2024-04-11 17:22:58 BGP: [DAMP] bgp_rfd_rt_update(): Route State: NONE, ret=0
2024-04-11 17:22:58 BGP: 192.168.12.1-Outgoing [RIB] Update: ...duplicate route ignored
2024-04-11 17:22:58 BGP: 192.168.12.1-Outgoing [RIB] Update: Received Prefix 10.10.3.0/24 path_id 0
2024-04-11 17:22:58 BGP: VRF 0 NSM announce: 10.10.3.0/24
```

5. 期间相关 BGP 邻居不会中断，路由表中原有的 BGP 路由没有消失。

方法 3 - 重置 BGP 邻居

1. 在 FW2 上使用 `execute router clear bgp ip 192.168.12.1` 命令进行 BGP 邻居重置，重新建立与 FW1 的 BGP 邻居并学习 BGP 路由。

> **重要**
>
> 不推荐使用此方法，BGP 邻居需要重新建立，流量中断时间长。
>
> 使用 execute router clear bgp all 可以重置所有的 BGP 邻居，请谨慎使用。

```bash
FW2 # execute router clear bgp ip 192.168.12.1
```

2. 随后 FW2 与 FW1 的 BGP 邻居被重置，并重新建立 BGP 邻居。

```bash
FW2 # get router info bgp summary

VRF 0 BGP router identifier 192.168.12.2, local AS number 10086
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.1 4      10086      17      20        0    0    0    never Idle
Total number of neighbors 1
```

3. 等待 BGP 邻居建立完成并重新学习 BGP 路由后，10.10.3.0/24 被放入路由表。

```bash
FW2 # get router info bgp summary

VRF 0 BGP router identifier 192.168.12.2, local AS number 10086
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.1 4      10086      30      33        2    0    0 00:09:03        3
Total number of neighbors 1

FW2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:08:45, [1/0]
B       10.10.2.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:08:45, [1/0]
B       10.10.3.0/24 [200/0] via 192.168.12.1 (recursive is directly connected, port2), 00:08:45, [1/0]
```

4. FW2 上的 BGP Debug 信息，显示 FW2 主动重置了 BGP 邻居，并通知了 FW1。

> **重要**
>
> 不需要向 FW1 发送 BGP Route-Refresh 请求报文。

```bash
FW2 # diagnose ip router bgp level info
FW2 # diagnose ip router bgp all enable
FW2 # diagnose debug console time enable
FW2 # diagnose debug enable

2024-04-11 17:28:23 id=20300 msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 192.168.12.1 Down BGP Notification CEASE"
2024-04-11 17:28:23 id=20300 msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 192.168.12.1 Down User reset"
```

5. 期间相关 BGP 中断，路由表中原有的 BGP 路由消失。

总结

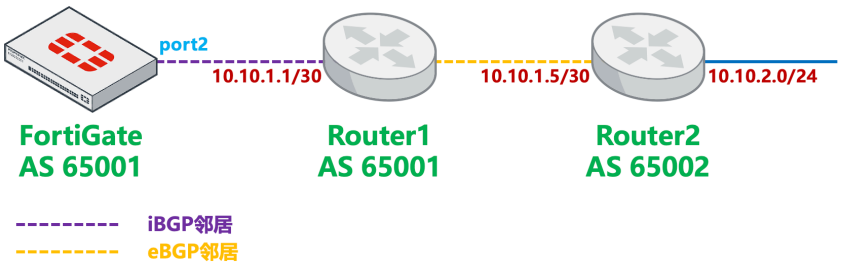
1. 三种刷新路由的方法优缺点总结：
   - 方法 1 - 使用 BGP route-refresh：
     - 优点：BGP 邻居和已有路由不会中断，相比方法 2，不需要在本地内存中保存未经路由策略过滤/修改的路由，节省内存。
     - 缺点：需要 BGP 邻居设备厂商支持 RFC 2918（FortiGate 支持）；根据对端配置的 `advertisement-interval`，收到更新报文存在一定延迟。
     - 配合 `exe router clear bgp ip 192.168.12.2 in` 使用。
   - 方法 2 - 使用 soft-reconfiguration（推荐）：
     - 优点：BGP 邻居和已有路由不会中断，不需要向 BGP 邻居发送 Refresh-Route 请求，仅本地执行。
     - 缺点：针对每个已建立的 BGP 邻居需要在内存中保存一份未经路由策略过滤/修改的路由列表，内存消耗较方法 1 稍大。
     - 配合 `exe router clear bgp ip 192.168.12.2 soft in` 使用。
   - 方法 3 - 重置 BGP 邻居：
     - 优点：不需要 BGP 邻居设备的任何 RFC 支持，所有厂商原生支持。
     - 缺点：BGP 邻居会重新建立并通告路由，流量中断时间长。
     - 配合 `exe router clear bgp ip 192.168.12.2 all` 或 `exe router clear bgp all` 使用。
2. FortiGate 不支持路由过滤/修改条件更改后的 BGP 路由条目自动更新，必须使用上述三种方法手动刷新路由。
3. 对于 out 方向的路由策略（路由过滤/修改），使用 `exe router clear bgp ip 192.168.12.2 out` 和 `exe router clear bgp ip 192.168.12.2 soft out` 都可以触发 out 方向的路由更新，从而向邻居发送 BGP Update 更新报文，本文就不再赘述。

BGP 下一跳递归

网络需求

默认情况下，当 BGP 路由的下一跳需要递归解析时，不会被考虑通过其他 BGP 路由进行递归。启 recursive-next-hop 后，才通过其他 BGP 路由进行递归。

网络拓扑



1. FortiGate 想要学习 Router2 的网段 10.10.2.0/24。
2. FortiGate 从 Router1 学习到了 Router2 的网段 10.10.2.0/24，下一跳为 10.10.1.5。
3. FortiGate 从 Router1 学习到了 10.10.1.4/30 的路由，下一跳为 10.10.1.1。

配置步骤

1. 配置基础网络（略）。

2. 配置 FortiGate 的 BGP，与 Router1 建立 iBGP 邻居。

```bash
config router bgp
    set as 65001
    set router-id 10.10.1.2
    config neighbor
        edit "10.10.1.1"
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
end
```

3. 配置 Router1 的 BGP，与 FortiGate 建立 iBGP 邻居，与 Router2 建立 eBGP 邻居，发布网段 10.10.1.4/30。

```bash
config router bgp
    set as 65001
    set router-id 10.10.1.1
    config neighbor
        edit "10.10.1.2"
            set soft-reconfiguration enable
            set remote-as 65001
        next
        edit "10.10.1.5"
            set soft-reconfiguration enable
            set remote-as 65002
        next
    end
    config network
        edit 1
            set prefix 10.10.1.4 255.255.255.252
        next
    end
end
```

4. 配置 Router2 的 BGP，与 Router1 建立 eBGP 邻居，发布网段 10.10.2.0/24。

```bash
config router bgp
    set as 65002
    set router-id 10.10.1.5
    config neighbor
        edit "10.10.1.6"
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

结果验证

1. FortiGate 从 Router1 学习到了两条 BGP 路由：

   - Router2 发布的路由，下一跳为 Router2 的接口 IP。
   - Router1 发布的路由，下一跳为 Router1 的接口 IP。

```bash
FortiGate # get router info bgp network
VRF 0 BGP table version is 1, local router ID is 10.10.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.4/30     10.10.1.1       0             100      0        0 i ←/1>
*>i10.10.2.0/24     10.10.1.5       0             100      0        0 65002 i ←/1>

Total number of prefixes 2
```

2. 但是 Router2 发布的路由未被 FortiGate 放入路由表，因为默认配置下，其下一跳 10.10.1.5 是无法通过 BGP 路由递归的。

```bash
FortiGate # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.4/30 [200/0] via 10.10.1.1 (recursive is directly connected, port2), 00:06:20, [1/0]
```

3. 在 FortiGate 上开启 BGP 的 recursive-next-hop。

```bash
config router bgp
    set recursive-next-hop enable
end
```

4. 再次查看 FortiGate 的路由表中的 BGP 路由，Router2 发布的路由被放入路由表，下一跳通过 Router1 发布的 BGP 路由递归。

```bash
FortiGate # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.4/30 [200/0] via 10.10.1.1 (recursive is directly connected, port2), 00:08:35, [1/0]
B       10.10.2.0/24 [200/0] via 10.10.1.5 (recursive via 10.10.1.1, port2), 00:00:02, [1/0]
```

路由反射器

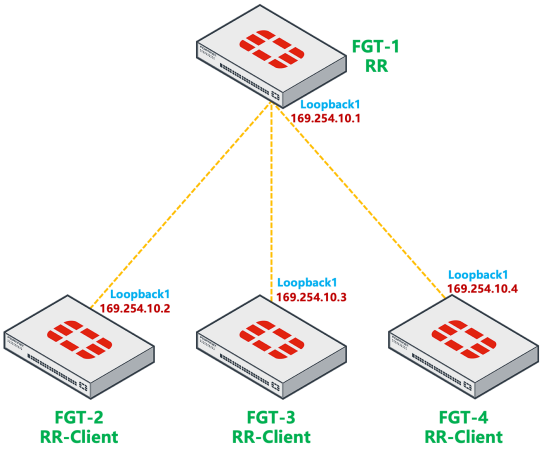
背景介绍

在 iBGP 环境中，BGP 水平分割规则会阻止从一个 iBGP 邻居学习到的路由再通告给其他 iBGP 邻居。

为保证 AS 内路由的完整传播，传统设计需要在所有 iBGP 路由器之间建立全互联邻居关系。当 AS 内 BGP 节点数量增多时，会导致 iBGP 邻居数量急剧增加，配置复杂、控制平面负担加重。

通过在 AS 内部署 BGP 路由反射器（RR，下文简称 RR），集中建立 iBGP 邻居关系，其余 iBGP 路由器作为路由反射客户端（RR Client，下文简称 RR 客户端），仅与 RR 建立会话，从而减少 iBGP 全互联需求。

网络拓扑



1. FGT-1 作为 RR（RR）。
2. FGT-2、FGT-3、FGT-4、FGT-5 作为 RR 客户端。
3. 所有设备位于同一 AS 65000。
4. RR 客户端之间不直接建立 iBGP 邻居，仅与 FGT-1 建立邻居。
5. RR 在 RR 客户端之间反射 iBGP 路由。

配置方法

RR 客户端

在 RR 客户端（FGT-2/FGT-3/FGT-4）上，仅需与 RR 建立 iBGP 邻居关系，配置示例如下（基础 BGP 配置略）：

```bash
config router bgp
    set as 65000
    set router-id 169.254.10.2
    config neighbor
        edit "169.254.10.1"
            set next-hop-self enable
            set interface "Loopback1"
            set remote-as 65000
            set update-source "Loopback1"
        next
    end
end
```

RR

1. RR（FGT-1）需要与 AS 内所有 RR 客户端（FGT-2/FGT-3/FGT-4）分别建立 iBGP 邻居关系，并在邻居配置中启用 `route-reflector-client` / `route-reflector-client6` 参数：

```bash
config router bgp
    set as 65000
    set router-id 169.254.10.1
    config neighbor
        edit "169.254.10.2"
            set soft-reconfiguration enable
            set interface "Loopback1"
            set remote-as 65000
            set update-source "Loopback1"
            set route-reflector-client enable
        next
        edit "169.254.10.3"
            set soft-reconfiguration enable
            set interface "Loopback1"
            set remote-as 65000
            set update-source "Loopback1"
            set route-reflector-client enable
        next
        edit "169.254.10.4"
            set soft-reconfiguration enable
            set interface "Loopback1"
            set remote-as 65000
            set update-source "Loopback1"
            set route-reflector-client enable
        next
    end
end
```

2. 由于 RR 的客户端可能数量众多，也可以选择使用 BGP 邻居组的方式来简化配置：

```bash
config router bgp
    config neighbor-group
        edit "RR"
            set soft-reconfiguration enable
            set interface "Loopback1"
            set remote-as 65000
            set update-source "Loopback1"
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 169.254.10.0 255.255.255.0
            set neighbor-group "RR"
        next
    end
end
```

NEXT_HOP 属性

在 FortiGate 上，RR 仅改变 iBGP 路由的传播方式，而不会默认修改任何 BGP 路径属性，详情参考：路由 → 动态路由 → BGP → BGP 常见问题 → RR 反射路由无法修改属性↗章节。

在默认行为下，即使 RR 反射了路由，NEXT_HOP 仍保持为原始学习到的下一跳。例如：

- RR（FGT-1）从 FGT-2 学习到的 iBGP 路由，其 NEXT_HOP 仍指向 FGT-1 的 169.254.10.2。
- RR 客户端 FGT-3/FGT-4 接收到该反射路由后，NEXT_HOP 仍为 FGT-2 的 169.254.10.2，而不是 RR 本身（169.254.10.1）。
- 若 RR 客户端 FGT-3/FGT-4 无法通过 IGP 到达该 NEXT_HOP，则可能导致转发失败。

修改非反射路由 NEXT_HOP

对于非反射路由（如 eBGP 学习的路由），可以在 BGP 邻居配置上开启 `next-hop-self` / `next-hop-self6`，该方式不会影响被反射的 iBGP 路由：

```bash
config router bgp
    config neighbor
        edit "169.254.10.2"
            set remote-as 65001
            set next-hop-self enable
```

```
        next
    end
end
```

修改反射路由 NEXT_HOP

在 iBGP 路由反射场景中，`next-hop-self` / `next-hop-self6` 不会修改被反射的 iBGP 路由的 NEXT_HOP，但 `next-hop-self-rr` / `next-hop-self-rr6` 参数可以修改反射路由的 NEXT_HOP。

启用后，所有由 RR 反射给 RR 客户端的 iBGP 路由，其 NEXT_HOP 将被统一修改为 RR 自身地址。如下配置：

- 在 RR（FGT-1）上，对每一个 RR 客户端启用：

```bash
config router bgp
    config neighbor
        edit "169.254.10.2"
            set next-hop-self-rr enable
        next
        edit "169.254.10.3"
            set next-hop-self-rr enable
        next
        edit "169.254.10.4"
            set next-hop-self-rr enable
        next
    end
end
```

- 或在 RR 的 BGP 邻居组中启用：

```bash
config router bgp
    config neighbor-group
        edit "RR"
            set route-reflector-client enable
        next
    end
end
```

- 配置完成后，需要在 RR 上执行路由刷新，以触发路由重新学习（详情参考：路由 → 动态路由 → BGP → BGP 路由刷新 ↗章节）：

```bash
execute router clear bgp all soft
```

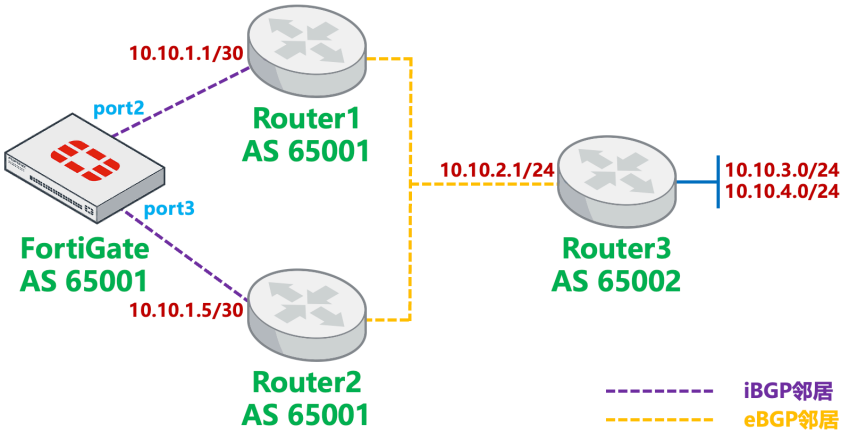
BGP 多路径

网络需求

当存在多条 ECMP（Equal-Cost Multipath）的 BGP 路由到达目标网络时，默认情况下只有第 1 个路由（先学习到的）会被放入路由表。

当开启 BGP multipath 功能后，所有这些 ECMP 的 BGP 路由都会被放入路由表，实现出站流量的负载均衡。允许同时使用多条等效路径，以提高网络性能和容错性。

网络拓扑



1. FortiGate 与 Router1 和 Router2 分别建立 iBGP 邻居，FortiGate 宣告自己的内网网段。
2. Router1 和 Router2 分别与 Router3 建立 eBGP 邻居，Router3 宣告自己的内网网段。

配置步骤

1. 基础网络配置（略）。

2. 配置 FortiGate 的 BGP，与 Router1 和 Router2 分别建立 iBGP 邻居。

```bash
config router bgp
    set as 65001
    set router-id 10.10.1.2
    config neighbor
        edit "10.10.1.1"
            set soft-reconfiguration enable
            set remote-as 65001
        next
        edit "10.10.1.5"
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.3.0 255.255.255.0
        next
    end
end
```

3. 配置 Router1 的 BGP，与 FortiGate 建立 iBGP 邻居，将路由下一跳修改为自己。与 Router3 建立 eBGP 邻居。

```bash
config router bgp
    set as 65001
    set router-id 10.10.1.1
    config neighbor
        edit "10.10.1.2"
            set next-hop-self enable
            set soft-reconfiguration enable
```

```
            set remote-as 65001
        next
        edit "10.10.2.1"
            set soft-reconfiguration enable
            set remote-as 65002
        next
    end
end
```

4. 配置 Router2 的 BGP，与 FortiGate 建立 iBGP 邻居，将路由下一跳修改为自己。与 Router3 建立 eBGP 邻居。

```bash
config router bgp
    set as 65001
    set router-id 10.10.1.5
    config neighbor
        edit "10.10.1.6"
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
        next
        edit "10.10.2.1"
            set soft-reconfiguration enable
            set remote-as 65002
        next
    end
end
```

5. 配置 Router3 的 BGP，分别与 Router1 和 Router2 建立 eBGP 邻居。

```bash
config router bgp
    set as 65002
    set router-id 10.10.2.1
    config neighbor
        edit "10.10.2.2"
            set soft-reconfiguration enable
            set remote-as 65001
        next
        edit "10.10.2.3"
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.10.4.0 255.255.255.0
        next
    end
end
```

结果验证

1. 在 FortiGate 上查看 BGP 学习到的路由，可以看到分别从 iBGP 邻居 Router1 和 Router2 学习到了 Router3 发布的 BGP 路由，下一跳分别为 Router1 和 Router2。

```bash
FortiGate # get router info bgp network
VRF 0 BGP table version is 4, local router ID is 10.10.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.3.0/24     0.0.0.0                          100  32768        0 i ←/1>
*>i10.10.4.0/24     10.10.1.1       0                100      0        0 65002 i ←/1>
* i                 10.10.1.5       0                100      0        0 65002 i ←/→

Total number of prefixes 2

FortiGate # get router info bgp network 10.10.4.0
VRF 0 BGP routing table entry for 10.10.4.0/24
```

```
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  65002
    10.10.1.1 from 10.10.1.1 (10.10.1.1)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Last update: Thu Jan 18 11:46:37 2024

  Original VRF 0
  65002
    10.10.1.5 from 10.10.1.5 (10.10.1.5)
      Origin IGP metric 0, localpref 100, valid, internal
      Last update: Thu Jan 18 11:47:05 2024
```

提示

`get router info bgp network` 中的 `<x/y>`：

- `x` 表示接收 Path ID（由对端设置）。
- `y` 表示发送 Path ID（由本机设置）。
- 由于未开启 BGP 的多路径功能，只有标记 `←/1>` 的路由会被放入路由表。

2. 但在 FortiGate 上只有下一跳为 Router1 的 BGP 路由被放入路由表。

```
FortiGate # get router info routing-table bgp
Routing table for VRF=0
B       10.10.4.0/24 [200/0] via 10.10.1.1 (recursive is directly connected, port2), 02:41:00, [1/0]
```

3. 配置 FortiGate 的 BGP 开启 ibgp-multipath。

```
config router bgp
    set ibgp-multipath enable
end
```

4. 再次查看 FortiGate 的路由表，下一跳为 Router2 的 BGP 路由也被放入路由表，路由下一跳在 Router1 和 Router2 间负载。

```
FortiGate # get router info routing-table bgp
Routing table for VRF=0
B       10.10.4.0/24 [200/0] via 10.10.1.1 (recursive is directly connected, port2), 00:00:55, [1/0]
                     [200/0] via 10.10.1.5 (recursive is directly connected, port3), 00:00:55, [1/0]
```

5. 在 Router3 上查看 BGP 学习到的路由，可以看到分别从 eBGP 邻居 Router1 和 Router2 学习到了 FortiGate 发布的 BGP 路由，下一跳分别为 Router1 和 Router2（eBGP 路由会自动更改下一跳为自身）。

```
Router3 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.10.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.3.0/24     10.10.2.2       0                      0        0 65001 i ←/1>
*                   10.10.2.3       0                      0        0 65001 i ←/→
*> 10.10.4.0/24     0.0.0.0                       100  32768        0 i ←/1>

Total number of prefixes 2
```

6. 但在 Router3 上只有下一跳为 Router1 的 BGP 路由被放入路由表。

```
Router3 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.10.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.3.0/24     10.10.2.2       0                      0        0 65001 i ←/1>
*                   10.10.2.3       0                      0        0 65001 i ←/→
*> 10.10.4.0/24     0.0.0.0                       100  32768        0 i ←/1>
```

```
Total number of prefixes 2

Router3 # get router info bgp network 10.10.3.0
VRF 0 BGP routing table entry for 10.10.3.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   10.10.2.3
  Original VRF 0
  65001
    10.10.2.2 from 10.10.2.2 (10.10.1.1)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Thu Jan 18 14:17:19 2024

  Original VRF 0
  65001
    10.10.2.3 from 10.10.2.3 (10.10.1.5)
      Origin IGP metric 0, localpref 100, valid, external
      Last update: Thu Jan 18 14:17:13 2024
```

7. 配置 FortiGate 的 BGP 开启 ebgp-multipath。

```bash
config router bgp
    set ebgp-multipath enable
end
```

8. 再次查看 Router3 的路由表，下一跳为 Router2 的 BGP 路由也被放入路由表，路由下一跳在 Router1 和 Router2 间负载。

```bash
Router3 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.3.0/24 [20/0] via 10.10.2.2 (recursive is directly connected, port2), 00:00:21, [1/0]
                     [20/0] via 10.10.2.3 (recursive is directly connected, port2), 00:00:21, [1/0]
```

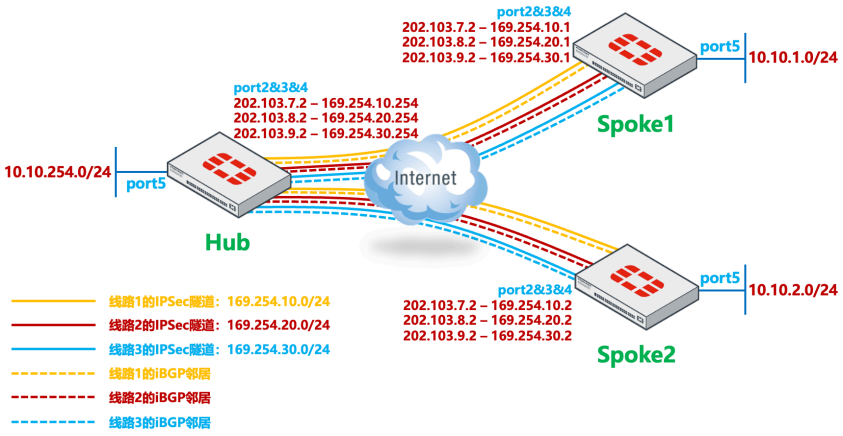
additional-path

功能简介

在 Hub-Spoke 场景中，每个 Spoke 往往通过多个链路（如 ISP1、ISP2、MPLS）与 Hub 建立多条 VPN 隧道。当 BGP 建立后，Hub 在向 Spoke 反射其他 Spoke 的路由时，可能只向 Spoke 通告单一路径，使得 Spoke 无法基于多路径进行更优的路径选择或负载分担。

通过启用 BGP Additional Path 机制：`additional-path-select` 与 `adv-additional-path`，Hub 可以向 Spoke 通告多个等价路径，使 Spoke 能够正确学习多条路由，提升 Hub-Spoke 的性能与稳定性。

网络拓扑



- IPSec VPN：
  - Hub 有 3 条 ISP，每条上线 1 条 dynamic 模式 IPSec，供 Spoke 拨入。
  - Spoke1、Spoke2 各有 3 条对应 ISP，与 Hub 的 3 条 IPSec 隧道一一对应连接。
- BGP：
  - Spoke 通过 3 条 IPSec 隧道分别与 Hub 的 3 条隧道建立 iBGP 邻居，并发布各自内网路由。
  - Hub 作为 BGP RR，在 Spoke 之间反射 iBGP 路由。
  - Hub 需要在 3 个 IPSec 隧道上负载去往 Spoke 的路由，Spoke 需要在 3 个 IPSec 隧道上负载去往 Hub 与其他 Spoke 的路由。

配置步骤

基础网络与安全策略配置（略）。

Hub

1. 配置 Hub 的 3 条 IPSec VPN dynamic 模式隧道，用于 Spoke 拨入。开启 `exchange-interface-ip` 用于交换 Tunnel 接口 IP，关闭 `add-route` 功能。

```bash
config vpn ipsec phase1-interface
    edit "vpn_line1"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-interface-ip enable
        set proposal aes128-sha256
        set add-route disable
        set dpd on-idle
        set psksecret xxxxxx
        set dpd-retryinterval 60
    next
    edit "vpn_line2"
        set type dynamic
```

2. 根据拓扑中的信息配置 Hub 的 IPSec 隧道 IP 地址，用于与 Spoke 建立 IBGP 邻居。

```bash
config system interface
    edit "vpn_line1"
        set ip 169.254.10.254 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.10.254 255.255.255.0
    next
    edit "vpn_line2"
        set ip 169.254.20.254 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.20.254 255.255.255.0
    next
    edit "vpn_line3"
        set ip 169.254.30.254 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.30.254 255.255.255.0
    next
end
```

3. 在 Hub 的 3 个 IPSec 隧道上分别配置 BGP 邻居组和邻居范围，用于接受 Spoke 的 iBGP 邻居建立，在每个邻居组中开启 RR 功能，并发布自身内网路由。

> **提示**
>
> 由于需要多线路负载 iBGP 路由，需要开启 `ibgp-multipath`，原理请参考路由 → 动态路由 → BGP → BGP 多路径↗章节。

```bash
config router bgp
    set as 10086
    set router-id 169.254.10.254
    set ibgp-multipath enable
    config neighbor-group
        edit "vpn_line1"
            set soft-reconfiguration enable
            set interface "vpn_line1"
            set remote-as 10086
            set route-reflector-client enable
        next
        edit "vpn_line2"
            set soft-reconfiguration enable
            set interface "vpn_line2"
            set remote-as 10086
            set route-reflector-client enable
```

Spoke1

1. 配置 Spoke1 的 3 条 IPSec VPN 隧道，分别对接 Hub 的 3 条 IPSec 隧道。开启 `exchange-interface-ip` 用于交换 Tunnel 接口 IP。

```bash
config vpn ipsec phase1-interface
    edit "vpn_line1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set exchange-interface-ip enable
        set proposal aes128-sha256
        set dpd on-idle
        set remote-gw 202.103.7.2
        set psksecret xxxxxx
    next
    edit "vpn_line2"
        set interface "port3"
        set ike-version 2
        set peertype any
        set exchange-interface-ip enable
```

2. 根据拓扑中的信息配置 Spoke1 的 IPSec 隧道 IP 地址，用于与 Hub 建立 iBGP 邻居。

```bash
config system interface
    edit "vpn_line1"
        set ip 169.254.10.1 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.10.254 255.255.255.0
    next
    edit "vpn_line2"
        set ip 169.254.20.1 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.20.254 255.255.255.0
    next
    edit "vpn_line3"
        set ip 169.254.30.1 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.30.254 255.255.255.0
    next
end
```

3. 在 Spoke1 的 3 个 IPSec 隧道上分别配置 BGP 邻居，用于向 Hub 发起 iBGP 邻居建立，发布自身内网路由。

> **提示**
>
> 由于需要多线路负载 iBGP 路由，需要开启 `ibgp-multipath`，原理请参考路由 → 动态路由 → BGP → BGP 多路径↗章节。

```bash
config router bgp
    set as 10086
    set router-id 169.254.10.1
    set ibgp-multipath enable
    config neighbor
        edit "169.254.10.254"
            set soft-reconfiguration enable
            set interface "vpn_line1"
            set remote-as 10086
            set update-source "vpn_line1"
        next
        edit "169.254.20.254"
            set soft-reconfiguration enable
            set interface "vpn_line2"
            set remote-as 10086
            set update-source "vpn_line2"
```

Spoke2

与 Spoke1 基本一致，点击展开查看。

1. 配置 Spoke2 的 3 条 IPSec VPN 隧道，分别对接 Hub 的 3 条 IPSec 隧道。开启 `exchange-interface-ip` 用于交换 Tunnel 接口 IP。

```bash
config vpn ipsec phase1-interface
    edit "vpn_line1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set exchange-interface-ip enable
        set proposal aes128-sha256
        set dpd on-idle
        set remote-gw 202.103.7.2
        set psksecret xxxxxx
    next
    edit "vpn_line2"
        set interface "port3"
        set ike-version 2
        set peertype any
        set exchange-interface-ip enable
```

2. 根据拓扑中的信息配置 Spoke2 的 IPSec 隧道 IP 地址，用于与 Hub 建立 iBGP 邻居。

```bash
config system interface
    edit "vpn_line1"
        set ip 169.254.10.2 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.10.254 255.255.255.0
    next
    edit "vpn_line2"
        set ip 169.254.20.2 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.20.254 255.255.255.0
    next
    edit "vpn_line3"
        set ip 169.254.30.2 255.255.255.255
        set allowaccess ping
        set remote-ip 169.254.30.254 255.255.255.0
    next
end
```

3. 在 Spoke2 的 3 个 IPSec 隧道上分别配置 BGP 邻居，用于向 Hub 发起 iBGP 邻居建立，发布自身内网路由。

> **提示**
>
> 由于需要多线路负载 iBGP 路由，需要开启 `ibgp-multipath`，原理请参考路由 → 动态路由 → BGP → BGP 多路径 ↗章节。

```bash
config router bgp
    set as 10086
    set router-id 169.254.10.2
    set ibgp-multipath enable
    config neighbor
        edit "169.254.10.254"
            set soft-reconfiguration enable
            set interface "vpn_line1"
            set remote-as 10086
            set update-source "vpn_line1"
        next
        edit "169.254.20.254"
            set soft-reconfiguration enable
            set interface "vpn_line2"
            set remote-as 10086
            set update-source "vpn_line2"
```

初始状态

1. 在 Hub 查看路由表中的 BGP 路由，可以看到从 Spoke1 和 Spoke2 的 3 条路径分别学到了 Spoke1 和 Spoke2 发布的 BGP 路由，符合预期。

```bash
Hub # get router info routing-table all
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 169.254.10.1 (recursive is directly connected, vpn_line1), 10:31:39, [1/0]
                     [200/0] via 169.254.20.1 (recursive is directly connected, vpn_line2), 10:31:39, [1/0]
                     [200/0] via 169.254.30.1 (recursive is directly connected, vpn_line3), 10:31:39, [1/0]
B       10.10.2.0/24 [200/0] via 169.254.10.2 (recursive is directly connected, vpn_line1), 10:31:19, [1/0]
                     [200/0] via 169.254.20.2 (recursive is directly connected, vpn_line2), 10:31:19, [1/0]
                     [200/0] via 169.254.30.2 (recursive is directly connected, vpn_line3), 10:31:19, [1/0]
......
S       169.254.10.0/24 [5/0] via vpn_line1 tunnel 202.103.7.2, [1/0]
C       169.254.10.1/32 is directly connected, vpn_line1
......
```

2. 在 Spoke1 的路由表中查看 BGP 路由，从 Hub 学到的 3 条 Hub 内网路由可以在 3 条 IPSec 隧道负载分担，符合预期。但经过 Hub 反射的 Spoke2 的 3 条 BGP 路由（10.10.2.0/24，`[3]`），只有一个下一跳（169.254.10.2，Spoke2 的 vpn_line1）。（Spoke2 也是一样的情况）

```bash
Spoke1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.254.0/24 [200/0] via 169.254.10.254 (recursive via vpn_line1 tunnel 202.103.7.2), 00:00:17, [1/0]
                       [200/0] via 169.254.20.254 (recursive via vpn_line2 tunnel 202.103.8.2), 00:00:17, [1/0]
                       [200/0] via 169.254.30.254 (recursive via vpn_line3 tunnel 202.103.9.2), 00:00:17, [1/0]
B       10.10.2.0/24 [200/0] via 169.254.10.2 [3] (recursive via vpn_line1 tunnel 202.103.7.2), 00:00:43, [1/0]
```

3. 从 Spoke1 的 BGP 路由状态中可以看到，虽然 Spoke1 确实从 3 个隧道分别收到了 Hub 反射的 Spoke2 路由，但下一跳均是 Spoke2 vpn_line1 的 IP，导致路由在递归时全部递归到 vpn_line1。

```bash
Spoke1 # get router info bgp network 10.10.2.0
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (3 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    169.254.10.2 from 169.254.10.254 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal, best
      Originator: 169.254.10.2, Cluster list: 169.254.10.254
      Last update: Wed Dec  3 16:23:58 2025

  Original VRF 0
  Local
    169.254.10.2 from 169.254.30.254 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal
      Originator: 169.254.10.2, Cluster list: 169.254.10.254
      Last update: Wed Dec  3 16:23:57 2025

  Original VRF 0
```

```
  Local
    169.254.10.2 from 169.254.20.254 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal
      Originator: 169.254.10.2, Cluster list: 169.254.10.254
      Last update: Wed Dec  3 16:23:57 2025
```

4. 在 Hub 查看 3 条隧道上向 Spoke1 的 3 个邻居反射的 Spoke2 的 BGP 路由（10.10.2.0/24），下一跳均为 169.254.10.2（Spoke2 的 vpn_line1）。

```bash
Hub # get router info bgp neighbors 169.254.10.1 advertised-routes | grep 10.10.2.0/24
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i ←/→

Hub # get router info bgp neighbors 169.254.20.1 advertised-routes | grep 10.10.2.0/24
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i ←/→

Hub # get router info bgp neighbors 169.254.30.1 advertised-routes | grep 10.10.2.0/24
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i ←/→
```

5. 这是由于当前配置下，虽然 Hub 的路由表里可看到 3 条来自 Spoke2 的 BGP 路由，但 Hub 仅向 Spoke1 通告最佳路径（Best Path，169.254.10.2），导致 Spoke1 学习到 3 条 Spoke2 的同一个下一跳路由。

```bash
Hub # get router info bgp network 10.10.2.0
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  Hub_line1 Hub_line2 Hub_line3
  Original VRF 0
  Local, (Received from a RR-client)
    169.254.30.2 from 169.254.30.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal
      Last update: Tue Dec  2 14:56:56 2025

  Original VRF 0
  Local, (Received from a RR-client)
    169.254.20.2 from 169.254.20.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal
      Last update: Tue Dec  2 14:56:40 2025

  Original VRF 0
  Local, (Received from a RR-client)
    169.254.10.2 from 169.254.10.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal, best
      Last update: Tue Dec  2 14:56:35 2025
```

开启 additional-path

在以上配置的基础上，启用 BGP Additional Path 机制，使 Hub 和 Spoke 均能通告与学习多个 BGP 等价路径。

1. 在 Hub 上配置 BGP additional-path。

> **提示**
>
> 需要在 `config router bgp` 中开启 `ibgp-multipath` 才能配置 `additional-path`。

```bash
config router bgp
    set additional-path enable
    set additional-path-select 3
    config neighbor-group
        edit "Hub_line1"
            set additional-path send
            set adv-additional-path 3
        next
        edit "Hub_line2"
            set additional-path send
            set adv-additional-path 3
        next
        edit "Hub_line3"
```

```
            set additional-path send
            set adv-additional-path 3
        next
    end
end
```

- `set additional-path enable`：BGP additional-path 路由学习的总开关，在 Hub（RR）上需要开启。需要先开启 `ibgp-multipath` 才能配置。
- `set additional-path-select 3`：默认值为 2。此配置下，相同 BGP 属性但下一跳不一样的路由，最多选取（select）3 条作为 `best` 路由。
- `set additional-path send`：在 BGP 协商时，向 BGP 邻居通告在发送方向启用 `IPv4 additional-path capability`，也就是自身具有发送 additional-path 路由的能力，也可以使用 `both`。
- `set adv-additional-path 3`：默认值为 2。向每个邻居通告的 IPv4 路由最大路径数量。仅在 `set additional-path send` 或 `set additional-path both` 时可以配置。

2. 在 Spoke1 和 Spoke2 上配置 BGP additional-path，仅需要在邻居配置中配置 `set additional-path receive`（此环境不需要在 Spoke 的 `config router bgp` 下配置 `set additional-path enable`）。

```bash
config router bgp
    config neighbor
        edit "169.254.10.254"
            set additional-path receive
        next
        edit "169.254.20.254"
            set additional-path receive
        next
        edit "169.254.30.254"
            set additional-path receive
        next
    end
end
```

- `set additional-path receive`：在 BGP 协商时，向 BGP 邻居通告在接收方向启用 `IPv4 additional-path capability`，也就是自身具有接收 additional-path 路由的能力，也可以使用 `both`。

> **重要**
>
> `additional-path send/receive/both` 为 BGP 邻居协商时交换的 BGP capability，需要 BGP 邻居两端同时开启才能生效，可以根据实际需求进行模式选择：
>
> - 一端配置为 `send`，另一端配置为 `receive/both`。
> - 一端配置为 `both`，另一端配置为 `send/receive`。
> - 两端均配置为 `both`。

3. 等待 Hub 的 BGP 路由重新收敛，查看 Hub 从 3 条 IPSec 隧道上学到的 Spoke2 的 3 条 BGP 路由，均变为具有 `best` 标记的路由（Hub 上配置的 `set additional-path-select 3` 作用效果）。

```bash
Hub # get router info bgp network 10.10.2.0
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  Hub_line1 Hub_line2 Hub_line3
  Original VRF 0
  Local, (Received from a RR-client)
    169.254.30.2 from 169.254.30.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal, best
      Advertised Path ID: 3
       Last update: Tue Dec  2 16:21:36 2025

  Original VRF 0
  Local, (Received from a RR-client)
    169.254.20.2 from 169.254.20.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal, best
      Advertised Path ID: 2
       Last update: Tue Dec  2 16:21:26 2025

  Original VRF 0
```

```
  Local, (Received from a RR-client)
    169.254.10.2 from 169.254.10.2 (169.254.10.2)
      Origin IGP distance 200 metric 0, localpref 100, valid, internal, best
      Advertised Path ID: 1
       Last update: Tue Dec  2 16:21:09 2025
```

4. 在 Hub 查看 3 条隧道上向 Spoke1 的 3 个邻居反射的 Spoke2 的 BGP 路由（10.10.2.0/24），可以看到 Hub 在每个邻居上都反射了 3 个路径的路由（Hub 上配置的 `set adv-additional-path 3` 作用效果）。

```bash
Hub # get router info bgp neighbors 169.254.10.1 advertised-routes | grep 10.10.2.0
*>i10.10.2.0/24     169.254.30.2                 100      0        0 i <-/3>
*>i10.10.2.0/24     169.254.20.2                 100      0        0 i <-/2>
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i <-/1>

Hub # get router info bgp neighbors 169.254.20.1 advertised-routes | grep 10.10.2.0
*>i10.10.2.0/24     169.254.30.2                 100      0        0 i <-/3>
*>i10.10.2.0/24     169.254.20.2                 100      0        0 i <-/2>
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i <-/1>

Hub # get router info bgp neighbors 169.254.30.1 advertised-routes | grep 10.10.2.0
*>i10.10.2.0/24     169.254.30.2                 100      0        0 i <-/3>
*>i10.10.2.0/24     169.254.20.2                 100      0        0 i <-/2>
*>i10.10.2.0/24     169.254.10.2                 100      0        0 i <-/1>
```

5. 在 Spoke1 查看路由表中的 BGP 路由，可以看到 Spoke1 在每个隧道上都学到了 3 条路径的 Spoke2 发布的 BGP 路由（Spoke 上配置的 `set additional-path receive` 作用效果）。

```bash
Spoke1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 169.254.10.2 [3] (recursive via vpn_line1 tunnel 202.103.7.2), 00:17:34, [1/0]
                     [200/0] via 169.254.20.2 [3] (recursive via vpn_line2 tunnel 202.103.8.2), 00:17:34, [1/0]
                     [200/0] via 169.254.30.2 [3] (recursive via vpn_line3 tunnel 202.103.9.2), 00:17:34, [1/0]
......
```

6. 在 Hub 上查看邻居 Spoke1 的状态，可以看到邻居 Spoke1 具有“Additional Path”的接收能力，而 Hub 自身具有“Additional Path”的发送能力。

```bash
Hub # get router info bgp neighbors 169.254.10.1 | grep Additional -A 2
    Additional Path:
      Send-mode: advertised
      Receive-mode: received
```

> **提示**
>
> 如果 Hub 配置为 `set additional-path both`，则会显示：
>
> ```
> Hub # get router info bgp neighbors 169.254.10.1 | grep Additional -A 2
>     Additional Path:
>       Send-mode: advertised
>       Receive-mode: advertised, received
> ```

7. 在 Spoke1 上查看邻居 Spoke1 的状态，可以看到邻居 Hub 具有“Additional Path”的发送能力，而 Spoke1 自身具有“Additional Path”的接收能力。

```bash
Spoke1 # get router info bgp neighbors 169.254.10.254 | grep Additional -A 2
    Additional Path:
      Send-mode: received
      Receive-mode: advertised
```

注意事项

1. BGP 只有在这些关键属性完全一致时，才会把路径视为 ECMP 等价路径：

   - Weight
   - Local Preference
   - AS Path
   - Origin
   - MED

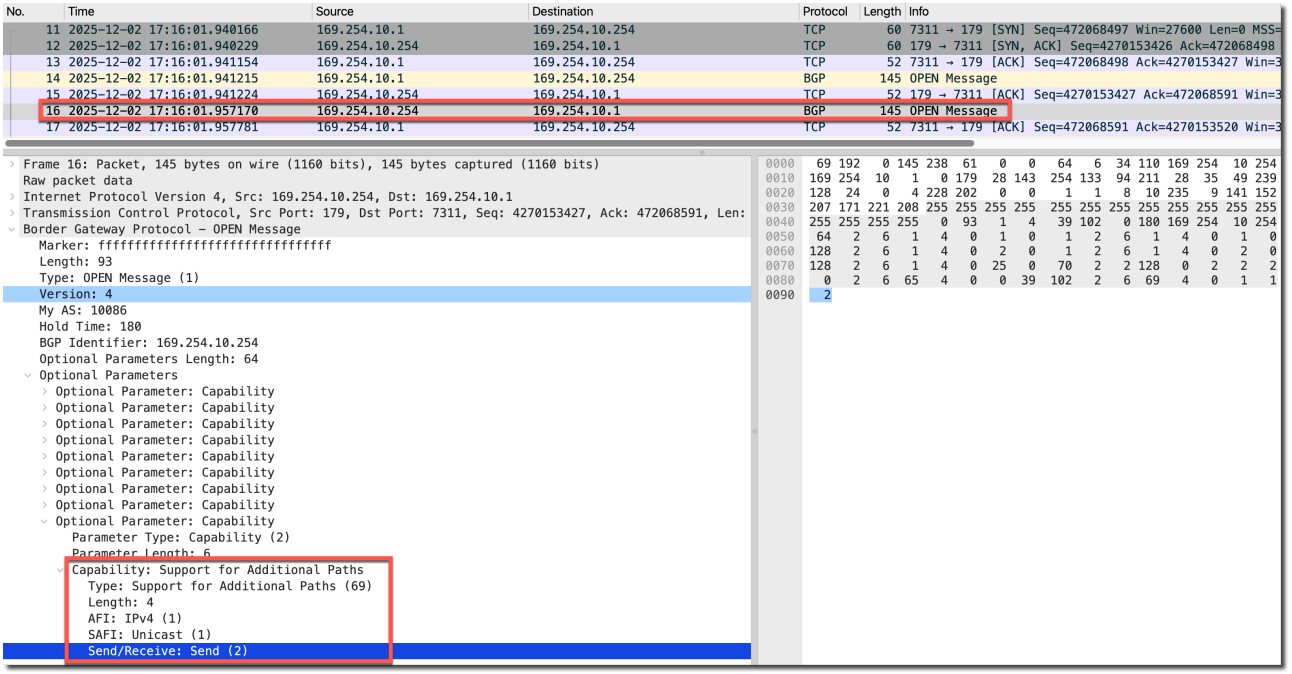
- IGP Metric

2. 配置 additional-path 后，如 Spoke 仍只学到一个路径，可启用 BGP 调试。若看到 `Duplicate route ignored`，则说明多路径属性不一致，需要检查以上字段是否完全相同。

```bash
diagnose ip router bgp all enable
diagnose ip router bgp level info
diagnose debug enable
```

附件

Hub 与 Spoke1 BGP 邻居协商 additional-path capability 的报文，如下所示，Hub 向 Spoke1 宣告的 `IPv4 additional-path capability` 为 `send`（bgp_additional_path.pcap ↗）。



总结

- `config router bgp`：
  - `additional-path enable`：开启全局 additional-path 能力，使 RR 本地 BGP 路由表可以同时保留多条等价路径，通常在 RR（Hub）上开启。需要先开启 `ibgp-multipath` 才能配置。
  - `additional-path-select <n>`：默认值为 2。依赖 `additional-path enable`，指定最多从多少条等价路径中“选取（select）”并标记为 best，用于控制本地路由表中可保留的等价路径数量，通常在 RR 上配置为与实际链路数一致。
- `config neighbor-group/neighbor`：
  - `set additional-path send/receive/both`：定义在 BGP capability 协商中，本端是否向对端通告具备发送或接收 `additional-path` 能力。只有双方能力匹配（send ↔ receive/both）后，多路径通告才会生效。典型用法是：RR 使用 `send`，RR Client 使用 `receive`。
  - `adv-additional-path <n>`：默认值为 2（范围 2~255）。决定 RR 本端针对单个邻居最多反射多少条路径。通常与 `additional-path-select` 数量相同，使 RR 可以将多条等价路径同时反射给下游 Spoke。

附加 AS-Path

功能简介

本文主要介绍在 BGP 多链路（如连接多个 ISP）场景中，如何通过 `AS-PATH prepend` 影响入向（回程）流量路径。

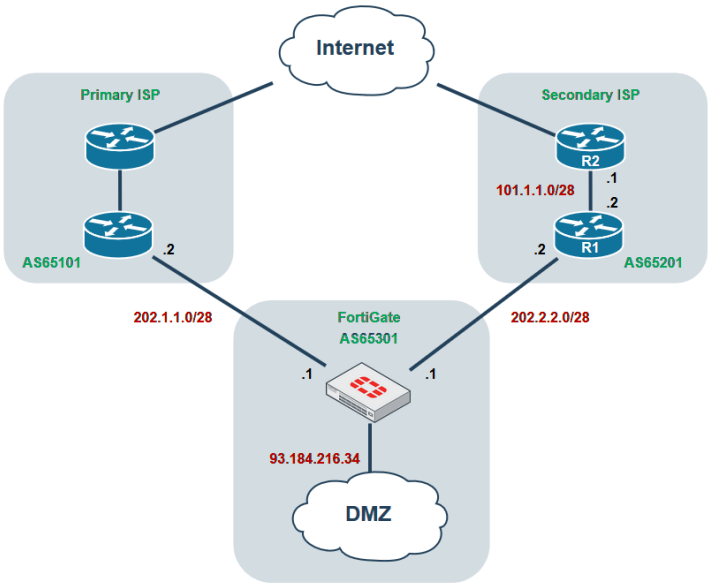
典型需求是将互联网出口设计为主备链路：

- 主线路（Primary ISP）作为优先路径。
- 备线路（Secondary ISP）仅在主线路故障时承载流量。

在此类场景中，仅在 ISP 侧设置 `Local Preference` 可能无法完全达到预期。因为 `Local Preference` 是非传递属性，只在各自 AS 内部生效，不会跨 AS 传播到其他 ISP。

因此可使用 `AS-PATH`（传递属性）进行路径影响：在向备线路通告前缀时，追加多个本地 AS 号，使该路径更“长”，从而降低其被优选概率。

网络拓扑



使用要点

- 互联网侧路由器的回程选路可通过 `AS-PATH` 属性进行影响。
- BGP 默认优选 AS-PATH 更短的路径。
- 在向 Secondary ISP 通告 `93.184.216.34/32` 时进行 prepend，可以让回程更倾向 Primary ISP，而不是依赖 ISP 侧 Local Preference。
- Secondary ISP（AS65201）到达 `93.184.216.34/32` 时，会因 Primary ISP 宣告路径更短而优选经 AS65101。
- 若需同时控制 FortiGate 出向优先主线路，可在本地对 Primary ISP 邻居设置更高 `Local Preference` 或 `Weight`。

配置步骤

示例参数如下：

- FortiGate 本地 AS：`65301`
- 主线路邻居（WAN1）：`202.1.1.2`（AS `65101`）
- 备线路邻居（WAN2）：`202.2.2.2`（AS `65201`）
- 对外发布前缀（DMZ）：`93.184.216.34/32`（通过 Loopback 模拟）

1. 接口配置（WAN1/WAN2/DMZ）。

```bash
config system interface
    edit "port2"
        set vdom "root"
        set ip 202.1.1.1 255.255.255.240
        set allowaccess ping
        set type physical
```

```
        set alias "WAN1"
    next
    edit "port3"
        set vdom "root"
        set ip 202.2.2.1 255.255.255.240
        set allowaccess ping
        set type physical
        set alias "WAN2"
    next
    edit "DMZ"
        set vdom "root"
        set ip 93.184.216.34 255.255.255.255
        set allowaccess ping
        set type loopback
    next
end
```

2. 备线路出站 route-map（AS-PATH prepend）。

```
config router route-map                                                    bash
    edit "Secondary_ISP"
        config rule
            edit 1
                set set-aspath "65301 65301 65301"
            next
        end
    next
end
```

3. 主线路入站 route-map（提高本地优先级）。

```
config router route-map                                                    bash
    edit "Primary_ISP"
        config rule
            edit 1
                set set-local-preference 200
            next
        end
    next
end
```

4. 在 BGP 邻居上应用 route-map。

```
config router bgp                                                          bash
    set as 65301
    set router-id 3.3.3.3
    config neighbor
        edit "202.1.1.2"
            set remote-as 65101
            set route-map-in "Primary_ISP"
        next
        edit "202.2.2.2"
            set remote-as 65201
            set route-map-out "Secondary_ISP"
        next
    end
    config network
        edit 1
            set prefix 93.184.216.34 255.255.255.255
        next
    end
end
```

验证步骤

变更前（未应用 prepend）

在 Secondary ISP 路由器查看，此时 `93.184.216.34/32` 的最佳路径为 `202.2.2.1`（经 Secondary ISP）。

```bash
ISP2-R1 # get router info bgp network
BGP table version is 8, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop      Metric LocPrf Weight RouteTag Path
*> 93.184.216.34/32  202.2.2.1          0      0      0        0 65301 i
* i                  101.1.1.1          0    100      0        0 65101 65301 i
*> 202.2.2.0/28      0.0.0.0                 100  32768        0 i
```

变更后（应用 prepend）

应用 prepend 后，最佳路径切换为 `101.1.1.1`（经 Primary ISP），并且备线路路径 AS-PATH 变长（出现多个 `65301`）。

```bash
ISP2-R1 # get router info bgp network
BGP table version is 10, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop      Metric LocPrf Weight RouteTag Path
*>i93.184.216.34/32  101.1.1.1          0    100      0        0 65101 65301 i
*  202.2.2.1                              0      0      0        0 65301 65301 65301 65301 i
*> 202.2.2.0/28      0.0.0.0                 100  32768        0 i
```

总结

1. `AS-PATH prepend` 主要用于影响入向（回程）流量。
2. 本地出向优选应配合 `Local Preference` / `Weight` 控制。
3. 变更后建议执行软重配或路由刷新并复核路由表。
4. 建议在维护窗口先小范围验证，避免生产回程路径抖动。

BGP Debug

1. BGP Debug 过滤器：

```bash
diagnose ip router bgp set-filter [vrf <vrf-id> | neighbor <neighbor-ip> | reset]
diagnose ip router bgp filters enable
```

2. 例如过滤 BGP 邻居 202.103.1.1 的 Debug 信息：

```bash
diagnose ip router bgp set-filter neighbor 202.103.1.1
diagnose ip router bgp filters enable
```

3. 开启 BGP Debug：

```
diagnose ip router bgp level info
diagnose ip router bgp all enable
diagnose debug console time enable
diagnose debug enable
```

4. 关闭 BGP Debug，清除 BGP Debug 过滤器：

```bash
diagnose debug disable
diagnose debug reset
diagnose ip router bgp set-filter reset
```

BGP 常见问题

- 清除 BGP 路由条目
- 路由震荡
- 使用 Loopback 接口建立 BGP
- BGP 与静态路由的优先顺序
- 关闭 Leftmost AS 检查
- BGP 选路忽略 AS-PATH
- 多个本地 AS 的配置和通告
- RR 反射路由无法修改属性

清除 BGP 路由条目

> **重要**
>
> 在 route-map 等路由对象更新后，路由刷新需要手动执行，请参考路由 → 动态路由 → BGP → 路由刷新 ↗章节。

1. 为了查看新路由是否被正确添加到路由表中，可以使用 `execute router clear bgp` 命令来清除所有或一些 BGP 邻居的 BGP 连接（会话）。
2. 例如，如果你在 BGP 路由表中有 10 个路由，并且想要清除邻居为 10.10.10.1 的特定路由，使用如下 CLI 实现。

```bash
execute router clear bgp ip 10.10.10.1
```

3. 要移除 AS 号为 65001 的所有路由，使用如下 CLI 实现。

```bash
execute router clear bgp as 65001
```

4. 注意，如果 BGP 的邻居配置中开启了 soft-reconfiguration，则在执行 `execute router clear bgp` 时可以在后边加入 soft 命令进行软重配，不需要重新与对端重新建立 BGP 邻居，仅重新学习路由。

```bash
exe router clear bgp all soft
```

路由震荡

产生原因

1. 当一条线路上的路由器频繁下线并重新上线，导致其他路由成员频繁的学习和删除相关路由，这被称为路由震荡（Flapping）。在 BGP 中，路由振荡是一个很大的问题，每当 BGP 邻居下线时，连接到该邻居的所有路由器都会在其路由表中发生对应变化。这会在网络上产生大量的 BGP 管理流量。当该路由器重新上线时，相同的流量会再次出现。如果是由于网络线路故障，比如每隔 10 秒就会交替在线和离线，就可能发送大量非必要的路由更新。
2. 另一个可能导致路由振荡的原因是在 HA 模式下有多个 FortiGate 设备。当 HA 的主设备切换到另一个成员时，网络上的其他路由器可能会将 HA 集群视为离线，导致路由振荡。虽然这不经常发生，但仍可能导致网络用户的服务中断。解决此问题的简单方法是增加 HA 集群上的 route-ttl，以确保从原主设备上同步的 BGP 路由在故障切换后不会立即过期。另外，配置 HA 集群上的 Graceful Restart 有助于实现平滑故障切换。

处理方法

1. 处理路由振荡的首先应该检查硬件。如果线缆松动或损坏，可以更换并消除问题。如果是路由器上的接口有问题，要么避免使用该接口，要么更换一个正常的路由器。如果路由器上的电源有问题，要么更换电源，要么使用备用电源。这些快速简单的修复方法可以避免配置更复杂的 BGP 选项。
2. 如果路由振荡是由于其他原因，配置 BGP 以处理中断将确保网络用户获得不间断的服务。一些缓解 BGP 路由振荡的方法包括：
   - 调整 Holdtime 计时器
   - Dampening
   - Graceful Restart
   - BFD

调整 Holdtime 计时器

在排查路由振荡问题时应先考虑调整 Holdtime 计时器，此计时器可以降低路由器下线导致路由更新的频率。

一旦 Holdtime 计时器激活，其将在持续时间内禁止 FortiGate 接受对该邻居的任何更改。例如在 Holdtime 计时器期间对端路由器振荡 5 次，FortiGate 只会识别到第 1 次中断。在后续 Holdtime 计时器持续时间内，不会发生任何变化，FortiGate 在 Holdtime 期间内都将这个路由器视为离线。如果路由器在 Holdtime 计时器到期后仍在振荡，将重新启动计时器。

如果对端路由器没有振荡（例如下线后重新上线，并保持在线），计时器仍然会倒计时，并在计时器持续时间内忽略该路由器的上线。在这种情况下，路由被视为下线的时间比实际时间长。在实际使用中，这不会引起严重问题，因为更新不是频繁的。而且当一个在线的路由器被视为离线时，可以看作是一种 BGP 的健壮性功能。通常情况下，我们不希望大部分流量通过一个不稳定的路由，这由 Holdtime 计时器来实现。

以下 3 种不同的路由振荡情况，都可以使用 holdtime 计时器来处理：

- 路由频繁上下波动。
- 路由在很长一段时间内下线再上线一次。
- 或者路由在很长一段时间内下线并保持下线。

例如，你的网络有两个路由邻居需要设置计时器：

- 一个是主路由邻居（101.103.1.1），所有的互联网流量都经过这个路由，如果它下线时间过长则会引起较大影响，主要路由的 Holdtime 计时器应该相对较短，如 60s。
- 另一个是很少使用的邻居（202.103.1.1）。主要路由的计时器应该相对较短（例如，60 秒）。次要路由邻居的 Holdtime 计时器可以较长，如默认值 180s。

```bash
config router bgp
    config neighbor
        edit 101.103.1.1
            set holdtime-timer 60
            set keepalive-timer 60
        next
        edit 202.103.1.1
            set holdtime-timer 180
            set keepalive-timer 60
        next
    end
end
```

Dampening

Dampening 是一种通过限制对端学习的路由而限制路由振荡的方法。使用 Dampening 时，虽然路由仍然会振荡，但对该对端路由的关注会逐渐减少。第一次振荡不会启动 Dampening，但第二次振荡将启动一个计时器，FortiGate 在计时器持续时间内将无视该路由器发送的路由。如果在 Dampening 计时器到期之前对端路由器再次振荡，计时器将继续增加。有一个称为 dampening-reachability-half-life 的时间，之后路由振荡将被抑制 Dampening

计时器一半的时间。当一个路由已经稳定一段时间，但尚未完全清除所有 Dampening 时，这个 half-life 就会生效。想要再次将振荡的路由加入路由表中，必须等待 Dampening 计时器到期。

如果路由振荡是暂时的，可以通过使用 `execute router clear bgp` CLI 从 FortiGate 设备的缓存中清除 Dampening：

```
# execute router clear bgp dampening {<ip_address> | <ip_address/netmask>}
或
# execute router clear bgp flap-statistics {<ip_address> | <ip_address/netmask>}
```

例如，要删除 10.10.1.0/24 子网的路由 Dampening 信息，请输入以下 CLI 命令：

```
# execute router clear bgp dampening 10.10.1.0/24
```

配置 Dampening：

```bash
config router bgp
    set dampening {enable | disable}
    set dampening-max-suppress-time <minutes_integer>
    set dampening-reachability-half-life <minutes_integer>
    set dampening-reuse <reuse_integer>
    set dampening-route-map <routemap-name_str>
    set dampening-suppress <limit_integer>
    set dampening-unreachability-half-life <minutes_integer>
end
```

Graceful Restart

在某些情况下，路由振荡是由 BGP 邻居中断但硬件部分（控制平面）仍然可以正常运行的情况引起的。例如当 BGP 进程重新启动，但硬件仍然可以正常运行和转发流量。Graceful Restart 适用于这种情况，GR 期间路由不会中断，但路由器对路由更新是不响应的。需要网络中的其他 BGP 路由器也支持优雅重启。

FortiGate HA 也可以从 Graceful Restart 中受益。当发生 HA 切换时，HA 集群会广播它即将下线，其他路由器不会视其为路由振荡。例如，FortiGate 是四个相互发送更新的 BGP 路由器之一。其中任何一个路由器都支持 GR。当 FortiGate（GR Restarter）发生 HA 切换后，GR Restarter 向其邻居发送一条消息，通知它正在重新启动。BGP 邻居（GR Helper）将所有正在重新启动的路由器的路由标记为 Stale，但它们继续使用这些路由。在重新启动完成后，将检查其路由并在需要时对其进行处理。对等方还知道在路由器重新启动期间可以维护哪些服务。在路由器完成重新启动后，路由器向其对等方发送一条消息，通知它已完成重新启动。

FortiGate 设备支持其自己的 BGP GR 和作为邻居 BGP 路由器的 GR Helper，配置请参考路由 → 动态路由 → BGP → BGP GR ↗ 章节。

BFD

Bidirectional Forwarding Detection（BFD）是一种用于快速定位网络中硬件故障的协议。运行 BFD 的路由器相互通信，如果检测计时器超时，那么该路由器将被宣告为下线。然后，BFD 将此信息通知给路由协议，路由信息将得到更新。

BFD 的原理和相关配置请参考：https://handbook.fortinet.com.cn/routing/bfd/ ↗

使用 Loopback 接口建立 BGP

当使用 Loopback 接口建立 BGP 邻居时，注意两端 FortiGate 的 Loopback 接口尽量使用 32 位掩码的 IP 地址，否则 BGP 邻居可能无法建立。

BGP 与静态路由的优先顺序

优先顺序

在路由表中，如果存在 2 条相同目的地址、AD（distance）的 BGP 路由和静态路由，那么 2 条路由无法同时存在于路由表中，较新一次更新的路由会被放入路由表，而旧路由会被取代。也就是说，谁最后有更新谁进路由表。

1. 如下所示，本来路由表中已存在 IBGP 路由 192.168.123.0/24（distance 为 200）。

```bash
61F_A # get router info routing-table bgp
Routing table for VRF=0
B       192.168.123.0/24 [200/0] via 10.10.10.3 (recursive via to_SH_WAN1 tunnel 10.0.0.1), 03w3d12h, [1/0]
```

2. 配置 distance 为 200 且与 BGP 路由内容一致的静态路由。

```bash
config router static
    edit 5
        set dst 192.168.123.0 255.255.255.0
        set distance 200
        set device "to_SH_WAN1"
    next
end
```

3. 查看路由表，静态路由取代 BGP 路由入表，路由优选静态路由。

```bash
FortiGate # get router info routing-table all
......
Routing table for VRF=0
......
S       192.168.123.0/24 [200/0] via to_SH_WAN1 tunnel 10.0.0.1, [1/0]


FortiGate # get router info routing-table details 192.168.123.0
Routing table for VRF=0
Routing entry for 192.168.123.0/24
  Known via "static", distance 200, metric 0, best ←——静态路由被优选
  * via to_SH_WAN1 tunnel 10.0.0.1 vrf 0, tun_id

Routing entry for 192.168.123.0/24
  Known via "bgp", distance 200, metric 0
  Last update 03w3d13h ago
    vrf 0 10.10.10.3 priority 1 (recursive via to_SH_WAN1 tunnel 10.0.0.1)
```

4. 此时重置 BGP 邻居，让 FortiGate 重新学习 BGP 路由，再次查看路由表，BGP 路由取代静态路由入表，路由优选 BGP 路由。

```bash
FortiGate # get router info routing-table all
......
Routing table for VRF=0
......
B       192.168.123.0/24 [200/0] via 10.10.10.3 (recursive via to_SH_WAN1 tunnel 10.0.0.1), 00:00:26, [1/0]

FortiGate # get router info routing-table details 192.168.123.0
Routing table for VRF=0
Routing entry for 192.168.123.0/24
  Known via "bgp", distance 200, metric 0, best ←——BGP路由被优选
  Last update 00:01:50 ago
  * vrf 0 10.10.10.3 priority 1 (recursive via to_SH_WAN1 tunnel 10.0.0.1)

Routing entry for 192.168.123.0/24
  Known via "static", distance 200, metric 0
    via to_SH_WAN1 tunnel 10.0.0.1 vrf 0, tun_id
```

其他

1. 如果想确保 BGP 路由进路由表，而静态路由不进路由表（作为 BGP 路由失效时的备份路由），可以通过 Router Map 改大 BGP 学习到的路由的 Weight（默认是 0，与静态路由一致）。

```bash
config router prefix-list
    edit "acl-weight"
        config rule
            edit 1
                set prefix 192.168.123.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end

config router route-map
    edit "weight-in"
        config rule
            edit 1
                set match-ip-address "acl-weight"
                set set-weight 40000
            next
            edit 2
            next
        end
    next
end

config router bgp
    config neighbor
        edit "10.10.10.3"
            set route-map-in "weight-in"
        next
    end
end
```

2. 此时 BGP 路由入表，而静态路由只存在于路由 database 中，当 BGP 路由异常消失时，静态路由可以作为备用路由生效。

```bash
FortiGate # get router info routing-table database    //查看路由数据库//
...
B    *> 192.168.123.0/24 [200/0] via 10.10.10.3 (recursive via to_SH_WAN1 tunnel 10.0.0.1), 00:03:41, [1/0]
S       192.168.123.0/24 [200/0] via to_Home_WAN1 tunnel 10.0.0.1, [1/0]
...

FortiGate # get router info routing-table all    //查看路由表//
...
B       192.168.123.0/24 [200/0] via 10.10.10.3 (recursive via to_SH_WAN1 tunnel 10.0.0.1), 00:03:46, [1/0]
...
```

关闭 Leftmost AS 检查

1. 默认情况下，FortiGate 会检查学习到的 BGP 路由的 AS Path 属性，要求 AS Path 最左侧的 AS（Leftmost AS，最近的 AS）要和本地 AS 一致，当收到的 BGP 路由的 AS Path 属性的 Leftmost AS 与本地 AS 不一致时，会拒绝接收 BGP Update，并在 BGP Debug 信息中显示类似 `Attr ASPATH: Incorrect leftmost AS number, should be 65001` 信息。

```bash
diagnose ip router bgp all enable
diagnose ip router bgp level info
diagnose debug enable

BGP: 10.10.10.2-Outgoing [DECODE] Update: Starting UPDATE decoding... Bytes To Read (4005), msg_size (86)
BGP: 10.10.10.2-Outgoing [DECODE] Attr ASPATH: Incorrect leftmost AS number, should be 65001
BGP: 10.10.10.2-Outgoing [FSM] State: Established Event: 28
BGP: 10.10.10.2-Outgoing [ENCODE] Msg-Hdr: Type 3
BGP: %BGP-3-NOTIFICATION: sending to 10.10.10.2 3/11 (UPDATE Message Error/Malformed AS_PATH.) 13 data-bytes [40 02 0a
02 02 00 00 4f f9 00 00 f4 73]
id=20300 logdesc="BGP neighbor status changed" msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.10.10.2 Down BGP
Notification FSM-ERR"
```

2. 通过如下 CLI 可以关闭对 BGP 路由 AS Path 的 Leftmost AS 的检查。

```bash
config router bgp
    set enforce-first-as disable
end
```

BGP 选路忽略 AS-PATH

功能简介

在 BGP 协议中，路由选择过程中，默认会通过多个路由属性算法来选择最佳路径。其中，AS-PATH 长度的比较是一个关键步骤。通常，AS-PATH 越短，表示路径越短，经过的 AS 数目越少，延迟和跳数更低，通常会被认为是性能更好的路径。

然而在某些场景下，AS-PATH 的长度可能并不反映网络的实际性能或可用性，管理员可能希望优先选择其他路由属性，如 localpref（local preference）、MED 或 IGP（内部网关协议）度量。

启用 `bestpath-as-path-ignore` 功能，可以使 BGP 在选择最佳路径时跳过对 AS-PATH 长度的比较，而将其他路由属性（如 localpref）作为决定因素，从而根据更合适的标准进行路径选择。

配置方法

```bash
config router bgp
    set bestpath-as-path-ignore [enable | disable]
end
```

网络拓扑

- **FGT-1（AS 65001）**：与外部网络和多个本地路由器（如 FGT-2）建立 eBGP 邻居。
- **FGT-2（AS 65006）**：与 FGT-1（AS 65001）和外部 AS 65007 建立 eBGP 邻居。
- FGT-1 与外部 AS 65007 之间存在多个 BGP 路径，有的路径的 AS-PATH 较短，而有的路径的 AS-PATH 较长（如通过其他 AS 进行路径预处理）。
- FGT-2 上，有两条路径到达 77.77.77.0/24 网络：
  - **路径 1**：通过 101.101.101.1，AS-PATH 为 65001 65007（长度为 2），localpref 为 100。
  - **路径 2**：通过 201.201.201.1，AS-PATH 为 65001 65002 65007（长度为 3），localpref 为 200。
- 在默认情况下，BGP 会选择路径 1，因为 AS-PATH 较短。然而，通过启用 `bestpath-as-path-ignore` 命令后，BGP 将忽略 AS-PATH 长度的比较，而选择路径 2，因为路径 2 的 localpref（200）高于路径 1 的 localpref（100）。

```bash
FGT-2 # get router info bgp network 77.77.77.0/24
VRF 0 BGP routing table entry for 77.77.77.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Original VRF 0
65001 65007 # Shorter path (length 2)
101.101.101.1 from 101.101.101.1 (19.19.19.19)
Origin IGP metric 0, localpref 100, valid, external, best
Last update: Tue Aug 26 12:15:57 2025
Original VRF 0
65001 65002 65007 # Longer path (length 3)
201.201.201.1 from 201.201.201.1 (19.19.19.19)
Origin IGP metric 0, localpref 200, valid, external
Last update: Tue Aug 26 12:15:57 2025
```

- 此时需要 BGP 选择一个路径时，不仅仅依赖 AS-PATH 长度，还能够根据其他属性（如 localpref 或 MED）来进行优化选择。

验证步骤

1. 在 FGT-2 上启用 `set bestpath-as-path-ignore` 命令，以忽略 AS-PATH 长度，并优先选择 localpref 更高的路径。

```bash
config router bgp
    set as 65006
    set bestpath-as-path-ignore enable
    config neighbor
        edit "101.101.101.1"
```

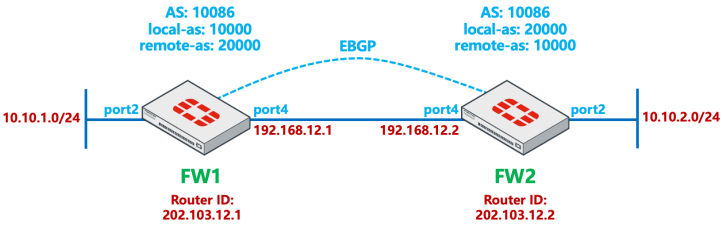
```
            set remote-as 65001
            set local-preference 200    <---设置localpref较高的路径
        next
    end
end
```

2. 在启用命令后，我们查看 BGP 路由表的变化。可以看到启用 `bestpath-as-path-ignore` 后，BGP 选择了 localpref 更高的路径，即使其 AS-PATH 更长。

```bash
fgt-2 # get router info bgp network 77.77.77.0/24
VRF 0 BGP routing table entry for 77.77.77.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Original VRF 0
65001 65007 # Shorter path
101.101.101.1 from 101.101.101.1 (19.19.19.19)
Origin IGP metric 0, localpref 100, valid, external
Last update: Tue Aug 26 12:15:57 2025
Original VRF 0
65001 65002 65007 # Longer path now best due to higher local pref
201.201.201.1 from 201.201.201.1 (19.19.19.19)
Origin IGP metric 0, localpref 200, valid, external, best
Last update: Tue Aug 26 12:15:57 2025
```

多个本地 AS 的配置和通告

网络拓扑



1. 例如同一数据中心的双 Hub 场景：
   - 两个 Hub（FW1 与 FW2）与 Spoke 建立 IBGP 邻居（使用 AS 10086），并做路由反射。
   - 两个 Hub 之间通过 EBGP（AS 使用 10000 和 20000）备份 IBGP 路由。
   - 此例为了简要说明问题，使用简化拓扑。

2. FW1 与 FW2 在 `config router bgp` 下（以下简称“全局配置”）配置的 AS 均为 10086，用于维护同一个 AS 下的 IBGP 邻居路由。

```bash
FW1:
config router bgp
    set as 10086
    set router-id 202.103.12.1
end

FW2:
config router bgp
    set as 10086
    set router-id 202.103.12.2
end
```

3. FW1 与 FW2 建立 EBGP 邻居，FW1 在 `config neighbor` 下配置 FW2 的 EBGP 邻居，`local-as` 为 10000，`remote-as` 为 20000，并宣告自身的路由 `10.10.1.0/24`。

```bash
config router bgp
    config neighbor
        edit "202.103.12.2"
            set soft-reconfiguration enable
            set interface "port4"
            set remote-as 20000
            set local-as 10000
            set update-source "port4"
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

4. FW2 在 `config neighbor` 下配置 FW1 的 EBGP 邻居，`local-as` 为 20000，`remote-as` 为 10000，并宣告自身的路由 `10.10.2.0/24`。

```bash
config router bgp
    config neighbor
        edit "202.103.12.1"
            set soft-reconfiguration enable
            set interface "port4"
            set remote-as 10000
            set local-as 20000
            set update-source "port4"
        next
    end
    config network
```

```
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

问题现象

1. FW1 与 FW2 的 EBGP 邻居建立完成。

2. 查看 FW1 的 BGP 邻居状态和 BGP 路由表，邻居已建立，但学不到任何路由。

```bash
FW1 # get router info bgp summary
VRF 0 BGP router identifier 202.103.12.1, local AS number 10086
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.2 4      20000      84      83        2    0    0 00:13:31        0

Total number of neighbors 1

FW1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                  100  32768        0 i ←/1>

Total number of prefixes 1
```

3. 查看 FW2 的 BGP 邻居状态和 BGP 路由表，邻居已建立，但学不到任何路由。

```bash
FW2 # get router info bgp summary
VRF 0 BGP router identifier 202.103.12.2, local AS number 10086
BGP table version is 3
2 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.1 4      10000     112     128        3    0    0 00:13:48        0

Total number of neighbors 1

FW2 # get router info bgp network
VRF 0 BGP table version is 3, local router ID is 202.103.12.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.2.0/24     0.0.0.0                  100  32768        0 i ←/1>

Total number of prefixes 1
```

4. 在 FW2 上查看向 FW1 发送的 BGP 路由，可以看到 FW2 向 FW1 宣告了自身路由 `10.10.2.0/24` 。

```bash
FW2 # get router info bgp neighbors 202.103.12.1 advertised-routes
VRF 0 BGP table version is 3, local router ID is 202.103.12.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.2.0/24     202.103.12.2             100  32768        0 i ←/→

Total number of prefixes 1
```

5. 在 FW1 上查看从 FW2 接收的 BGP 路由，可以看到 FW1 从 FW2 收到了路由 `10.10.2.0/24`，但并未放入 BGP 路由表。该路由的 AS Path 为 `20000 10086`，`10086` 为 FW2 在全局（`config router bgp`）下配置的 `local-as`。

```bash
FW1 # get router info bgp neighbors 202.103.12.2 received-routes
VRF 0 BGP table version is 2, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.2.0/24     202.103.12.2                         0        0 20000 10086 i ←/→

Total number of prefixes 1
```

问题原因

1. 如果 `config neighbor` 下没有配置 `local-as`，EBGP 会以 AS 路径长度为 1 的方式公布源自其 AS 内的路由，只列出在 BGP 全局配置下指定的 AS 号。
2. 当 `config neighbor` 下的 `local-as` 设置了 AS 号（此例中为 20000），默认配置下，BGP 会将其作为全局配置的 AS 号（此例中为 10086）的前缀，从而导致 FW2 发布给 FW1 的路由的 AS Path 变为 `20000 10086`。
3. 同理，如果路由是从一个 EBGP 邻居学到，发布给另一个 EBGP 邻居，则会 AS Path 长度会变成 3。例如 FW2 从另一个 EBGP 邻居（AS 30000）学到的路由发送给 FW1，在 FW1 上看到的 AS Path 为 `20000 10086 30000`。
4. 在 FW1 上查看 BGP Debug 信息，可以看到 FW1 收到 FW2 发布的路由 `10.10.2.0/24`，但 AS Path 为 `20000 10086`，其中包含在全局配置中设置的 AS 10086，根据 BGP 防环机制，FW1 将该路由丢弃（`Prefix 10.10.2.0/24 path_id 0 denied due to as-path contains our own AS`）。

```bash
diagnose ip router bgp set-filter neighbor 202.103.12.2
diagnose ip router bgp filters enable
diagnose ip router bgp level info
diagnose ip router bgp all enable

FW1# BGP: 202.103.12.2-Outgoing [DECODE] Msg-Hdr: type 2, length 51
BGP: 202.103.12.2-Outgoing [DECODE] Update: Starting UPDATE decoding... Bytes To Read (32), msg_size (32)
BGP: 202.103.12.2-Outgoing [DECODE] Update: NLRI Len(4)
BGP: 202.103.12.2-Outgoing [FSM] State: Established Event: 27
BGP: 202.103.12.2-Outgoing [RIB] Update: Prefix 10.10.2.0/24 path_id 0 denied due to as-path contains our own AS
```

5. FW2 学不到 FW1 发布的路由 `10.10.1.0/24` 也是同样的原因。

解决方法

local-as-replace-as（推荐）

> **重要**
>
> `local-as-replace-as` 与 `local-as-no-prepend` 都需要在 `config router bgp` 下配置了 `local-as`，且与全局（`config router bgp`）下配置的 AS 号不一样时才能配置。

1. 在 FW2 上的 EBGP 邻居配置中，开启 `local-as-replace-as`。开启后，FW2 会使用 `config neighbor` 下配置的 `local-as` 替换在全局（`config router bgp`）下配置的 AS 号。

```bash
config router bgp
    config neighbor
        edit "202.103.12.1"
            set local-as-replace-as enable
        next
    end
end
```

2. 在 FW1 上查看收到的 BGP 路由，FW1 将收到 AS Patch 长度为 1 的路由，如下所示，AS Patch 为 `20000`，此为 FW2 在 `config neighbor` 下配置的 `local-as`，FW2 没有添加在全局（`config router bgp`）下配置的 AS 号 `10086`。

重要

`local-as-replace-as` 只会在出方向使用 `config neighbor` 下配置的 `local-as` 替换全局（`config router bgp`）下配置的 AS 号，已有的 AS Path 不会更改。

```bash
FW1 # get router info bgp neighbors 202.103.12.2 received-routes
VRF 0 BGP table version is 3, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.2.0/24     202.103.12.2                           0        0 20000 i ←/→

Total number of prefixes 1
```

3. 在 FW1 上查看 BGP 路由表，从 FW2 学习到的路由 `10.10.2.0/24` 已被正确加载，并添加了自己在 `config neighbor` 下配置的 `local-as` 号 `10000`。

```bash
FW1 # get router info bgp network
VRF 0 BGP table version is 3, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                          100  32768        0 i ←/1>
*> 10.10.2.0/24     202.103.12.2    0                     0        0 10000 20000 i ←/1>

Total number of prefixes 2
```

4. 在 FW1 上查看路由表中的 BGP 路由，从 FW2 学习到的 EBGP 路由被正确加载。

```bash
FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [20/0] via 202.103.12.2 (recursive is directly connected, port4), 00:07:42, [1/0]
```

5. 同理，在 FW1 上的 EBGP 邻居中开启 `local-as-replace-as`，FW2 也可以正常学习到 FW1 发布的路由。

6.**附加测试** `local-as-no-prepend`：在 FW1 的 EBGP 邻居配置中，开启 `local-as-no-prepend`，FW1 将不会在 BGP 路由表中添加自己在 `config neighbor` 中配置的 `local-as`。

重要

`local-as-no-prepend` 作用于入方向，例如 FGT 在与 EBGP 邻居 A（配置了 `local-as`），开启 `local-as-no-prepend` 后，向 EBGP 邻居 B（未配置 `local-as`）转发通告 A 的 EBGP 路由时，不会在 AS Path 中加上与 EBGP 邻居 A 配置的 `local-as`。

```bash
config router bgp
    config neighbor
        edit "202.103.12.2"
            set local-as-no-prepend enable
        next
    end
end

FW1 # get router info bgp network
VRF 0 BGP table version is 5, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                          100  32768        0 i ←/1>
*> 10.10.2.0/24     202.103.12.2    0                     0        0 20000 i ←/1>

Total number of prefixes 2
```

allowas-in-enable

1. 恢复默认配置（在 FW1 和 FW2 上禁用 `local-as-replace-as`）。

2. 在 FW1 上的 EBGP 邻居配置中，开启 `allowas-in-enable`。开启后，FW1 会允许 AS Path 包含自身 AS 号的 BGP 路由加载（EBGP 防环失效）。

```bash
config router bgp
    config neighbor
        edit "202.103.12.2"
            set allowas-in-enable enable
            set allowas-in 3    //默认配置，允许自身AS号在AS Path中出现的最大次数//
        next
    end
```

3. 在 FW1 上查看从 FW2 接收的 BGP 路由，可以看到 FW1 从 FW2 收到了路由 `10.10.2.0/24`。该路由的 AS Path 为 `20000 10086`，`10086` 为 FW2 在全局（`config router bgp`）下配置的 `local-as`。

```bash
FW1 # get router info bgp neighbors 202.103.12.2 received-routes
VRF 0 BGP table version is 6, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.2.0/24     202.103.12.2                           0        0 20000 10086 i ←/→

Total number of prefixes 1
```

4. 在 FW1 上查看 BGP 路由表，从 FW2 学习到的路由 `10.10.2.0/24` 已被正确加载，并添加了自己在 `config neighbor` 下配置的 `local-as` 号 `10000`。

```bash
FW1 # get router info bgp network
VRF 0 BGP table version is 6, local router ID is 202.103.12.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                          100  32768        0 i ←/1>
*> 10.10.2.0/24     202.103.12.2    0                     0        0 10000 20000 10086 i ←/1>

Total number of prefixes 2
```

5. 在 FW1 上查看路由表中的 BGP 路由，从 FW2 学习到的 EBGP 路由被正确加载。

```bash
FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [20/0] via 202.103.12.2 (recursive is directly connected, port4), 00:09:25, [1/0]
```

6. 同理，在 FW2 上的 EBGP 邻居中开启 `allowas-in-enable`，FW2 也可以正常学习到 FW1 发布的路由。

7. 不推荐使用此方法解决，会破坏 EBGP 的防环机制。

RR 反射路由无法修改属性

网络环境

Hub 与 Spoke 在同一个 AS 65000 内：

- Hub：作为 BGP Route Reflector，IP 为 `169.254.10.254`。
- Spoke1：Hub 的 iBGP 邻居，IP 为 `169.254.10.1`。
- Spoke2：Hub 的 iBGP 邻居，IP 为 `169.254.10.2`。
- Hub 通过 BGP 通告 `10.10.254.0/24` 网段。
- Spoke1 向 Hub 通告前缀 `10.10.1.0/24`，Hub 通过 RR 将 `10.10.1.0/24` 反射给 Spoke2。
- Hub 上对发往 Spoke2 的方向配置了 `route-map-out`，意图修改自身发布和反射给 Spoke2 的路由的 Local Preference。

问题现象

1. Spoke1 向 Hub 通告网段：`10.10.1.0/24`。

2. Hub 作为 Route Reflector 的 BGP 配置如下。Hub 对 Spoke2 的出站 route-map 用于试图修改通告/反射到 Spoke2 的路由的 Local Preference 为 500。

```bash
config router route-map
    edit "Hub-RM_Spoke2-OUTBOUND"
        config rule
            edit 1
                set set-local-preference 500
            next
        end
    next
end

config router bgp
    set as 65000
    config neighbor
        edit "169.254.10.1" ←———Spoke1 IP
            set remote-as 65000
            set route-reflector-client enable
        next
        edit "169.254.10.2" ←———Spoke2 IP
            set remote-as 65000
            set route-map-out "Hub-RM_Spoke2-OUTBOUND"
            set route-reflector-client enable
        next
    end

    config network
        edit 1
            set prefix 10.10.254.0 255.255.255.0 ←———Hub 发布到 BGP 的网段
        next
    end
end
```

3. Hub 从 Spoke1 学到的 BGP 路由 `10.10.1.0/24`，`LocPrf` 为 100。

```bash
Hub # get router info bgp neighbors 169.254.10.1 routes

VRF 0 BGP table version is 44, local router ID is 10.255.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop        Metric `LocPrf` Weight RouteTag Path
*  >i10.10.1.0/24 169.254.10.1   0      100    0      0        i    ←———由Spoke1发布并安装在Hub的路由
```

4. 由于 Hub 被配置为 Route Reflector，它会将从 Spoke1 学到的路由 `10.10.1.0/24` 反射给 Spoke2，同时也会把自己发布的路由 `10.10.254.0/24` 一并通告。这里可以看到发送给 Spoke2 的 2 条路由中，只有 Hub 自身发布的路由 `LocPrf` 被改为 500，反射的路由仍然为 100。

```bash
Hub # get router info bgp neighbors 169.254.10.2 advertised-routes

VRF 0 BGP table version is 44, local router ID is 10.255.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop        Metric `LocPrf` Weight RouteTag Path
*  >i10.10.1.0/24   169.254.10.1    100     0       0      i  ←——Spoke1发布被Hub反射路由
*  >i10.10.254.0/24 169.254.10.254    500    32768  0      i  ←——Hub自身发布的路由

Total number of prefixes 2
```

5. 在 Spoke2 上查看来自 Hub（`169.254.10.254`）的路由，可以看到 2 条前缀，Hub 发布的路由的 `LocPrf` 被修改为 500，但反射的 Spoke1 的路由 `LocPrf` 没有被改变，仍然为 100。

```bash
Spoke2 # get router info bgp neighbors 169.254.10.254 received-routes

VRF 0 BGP table version is 16, local router ID is 10.255.255.30
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop        Metric `LocPrf` Weight RouteTag Path
*  >i10.10.1.0/24   169.254.10.1   0       100    0      0        i ←——Spoke1发布的反射路由
*  >i10.10.254.0/24 169.254.10.254    0      500    0      0        i ←——Hub自身发布的路由
Total number of prefixes 2
```

问题原因

- BGP Route Reflection 由 RFC 4456 定义，Route Reflector 会将从 IBGP peer 学到的前缀通告给所有 route-reflector clients。
- 按 RFC 4456 第 10 节的建议，Route Reflector 不应修改反射路由的，也就是说，对反射路由不会应用 `route-map-out` 中的属性修改：
  - NEXT_HOP
  - AS_PATH
  - LOCAL_PREF
  - MED（metric）
  - community

总结

- 当 FortiGate 作为 Route Reflector 时，`route-map-out` 不会修改反射路由的属性。
- 在以上示例中，只有 Hub 自己通过 `network` 注入的 `10.10.254.0/24` 被成功改写 Local Preference。
- 来自 Spoke1 的 `10.10.1.0/24` 作为反射路由，其 Local Preference 保持不变。

解决方法

若希望在 Spoke1 → Spoke2 经过 Hub 反射的路径上对 BGP 属性进行调整，可以在以下任一位置进行修改：

- Spoke1 的 route-map-out to Hub。
- Hub 的 route-map-in from Spoke1。
- Spoke2 的 route-map-in from Hub。
- 在 Hub 上开启 `next-hop-self-rr`，可以将反射路由的 NEXT_HOP 修改为 Hub 自身地址（详情参考：路由 → 动态路由 → BGP → 路由反射器 ↗章节）。

路由对象

- Route Map
- Access List
- Prefix List
- AS Path List
- Community List

Route Map

Route Map 一种强大的工具，可以根据特定条件对动态路由协议进行操作。主要用于 BGP，用于操作由 FortiGate 宣告的路由（route-map-out）或从其他路由器接收的路由（route-map-in）。

在 OSPF 中，可以使用 Route Map 进行条件性的 default-information-originate（宣告默认路由）、过滤邻居宣告的路由，或者匹配特定路由进行重分发。Route Map 也可以被 RIP 使用来匹配重分发的路由。

一个 Route Map 可以包含多个规则，按照自上而下的顺序进行处理。每个规则都有允许或拒绝的动作。这些规则根据各种属性匹配路由，并根据匹配的路由设置路由相关属性，最后的隐含动作为 deny any。

例如可以使用 Route Map 来匹配特定 Community 的 BGP 路由，然后设置匹配路由的 AS Path。然后在 BGP 邻居的设置中应用此 Route Map。

配置方法

```bash
config router route-map
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set match-as-path <string>
                set match-community <string>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

- `match-as-path <string>`：匹配一个 BGP AS Path List。
- `match-community <string>`：匹配一个 BGP Community。
- `match-ip-address <string>`：匹配一个 access-list 或 prefix-list。
- `match-ip6-address <string>`：匹配一个 access-list6 或 prefix-list6。
- `match-ip-nexthop <string>`：匹配 IPv4 路由的下一跳，通过 access-list 或 prefix-list 表示。
- `match-ip6-nexthop <string>`：匹配 IPv6 路由的下一跳，通过 access-list6 或 prefix-list6 表示。

引用 Route Map

被 RIP 引用

```bash
config router rip
    config redistribute
        edit <name>
            set routemap <string>
        next
    end
end
```

被 OSPF 引用

```bash
config router ospf
    set default-information-route-map <string>
    set distribute-route-map-in <string>
    config redistribute <string>
        set routemap <string>
    end
end
```

- `default-information-route-map <string>`：根据 Route Map 中匹配的路由是否存在于路由表，决定是否宣告默认路由。
- `distribute-route-map-in <string>`：入方向根据引用的 Route Map 操作路由（仅针对 external 路由）。
- `redistribute <string>`：在重发布中引用 Route Map 过滤路由。

被 BGP 引用

```bash
config router bgp
    config neighbor
        edit <ip>
            set route-map-in <string>
            set route-map-in6 <string>
            set route-map-in-vpnv4 <string>
            set route-map-out <string>
            set route-map-out-preferable <string>
            set route-map-out6 <string>
            set route-map-out6-preferable <string>
            set route-map-out-vpnv4 <string>
            set route-map-out-vpnv4-preferable <string>
        next
    end
    config network
        edit <id>
            set prefix <IP/netmask>
            set route-map <string>
        next
    end
    config redistribute <string>
        set route-map <string>
    end
end
```

- `route-map-in <string>`：在入方向引用 Route Map 操作 IPv4 路由。
- `route-map-in6 <string>`：在入方向引用 Route Map 操作 IPv6 路由。
- `route-map-in-vpnv4 <string>`：在入方向引用 Route Map 操作 VPNv4 路由。
- `route-map-out <string>`：在出方向引用 Route Map 操作 IPv4 路由。
- `route-map-out-preferable <string>`：在 SD-WAN Neighbor 中满足 SLA 标准的 BGP 邻居使用的出方向 Route Map，针对 IPv4 路由（不满足 SLA 标准的 BGP 路由使用普通的 route-map-out）。
- `route-map-out6 <string>`：在出方向引用 Route Map 操作 IPv6 路由。
- `route-map-out6-preferable <string>`：在 SD-WAN Neighbor 中满足 SLA 标准的邻居使用的出方向 Route Map，针对 IPv6 路由（不满足 SLA 标准的 BGP 路由使用普通的 route-map-out6）。
- `route-map-out-vpnv4 <string>`：在出方向引用 Route Map 操作 VPNv4 路由。
- `route-map-out-vpnv4-preferable <string>`：在 SD-WAN Neighbor 中满足 SLA 标准的 BGP 邻居使用的出方向 Route Map，针对 VPNv4 路由（不满足 SLA 标准的 BGP 路由使用普通的 route-map-out-vpnv4）。
- `route-map <string>`：引用 Route Map 操作本地宣告的 BGP 路由。
- `redistribute <string>`：在重发布中调用 Route Map。

```bash
config router bgp
    set as <AS_number>
    config neighbor
        edit <ip>
            set remote-as <AS_number>
            config conditional-advertise
                edit <advertise-routemap>
                    set condition-routemap <name1>, <name2>, ...
                    set condition-type {exist | non-exist}
                next
            end
        next
    end
end
```

- `<advertise-routemap>`：引用 Route Map 匹配要宣告的网络。
- `condition-routemap <name1>, <name2>, ...`：Route Map 匹配条件的路由是否存在于路由表。

Access List

访问列表（Access List）是用于基于 IPv4/IPv6 地址 + 掩码的 prefix 进行路由过滤的简单列表。

配置方法

```bash
IPv4:
config router access-list
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set prefix <IPv4_address>
                set wildcard <wildcard_filter>
                set exact-match {enable | disable}
            next
        end
    next
end

IPv6:
config router access-list6
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set prefix <IPv6_address>
                set exact-match {enable | disable}
            next
        end
    next
end
```

- `action {permit | deny}`：动作是允许还是拒绝路由条目。
- `prefix <IPv4_address>`：IPv4 路由的前缀 + 掩码，如 `10.10.1.0/24`。该配置与 `wildcard` 互斥。
- `wildcard <wildcard_filter>`：以通配符掩码（反掩码）方式匹配路由，如 `10.0.104.0 0.255.0.255`（匹配掩码为 1 的位，为 0 的位忽略），该配置与 `prefix` 互斥。
- `exact-match {enable | disable}`：是否严格匹配，如果开启严格匹配，只有 prefix 和 wildcard（掩码）完全相同的路由会匹配此 Access List；如果关闭严格匹配，大于该掩码的 CIDR 路由也会匹配，默认关闭（仅在使用 `prefix` 模式时生效）。

引用 Access List

被 RIP 引用

```bash
config router rip
    config distribute-list
        edit <id>
            set direction {in | out}
            set listname <string>
        next
    end
    config offset-list
        edit <id>
            set direction {in | out}
            set access-list <string>
            set offset <integer>
        next
    end
end
```

- `listname <string>`：引用 Access List 或 Prefix List 用于出/入方向路由过滤。
- `access-list <string>`：针对 Access List 匹配的路由修改路由的跳数（这里 Access List 中的动作 permit/deny 表示匹配/不匹配）。

被 OSPF 引用

```bash
config router ospf
    set distribute-list-in <string>
    config distribute-list
        edit <id>
            set access-list <string>
            set protocol {connected | static | rip}
        next
    end
end
```

- `distribute-list-in <string>`：入方向引用 Access List 过滤路由。
- `access-list <string>`：发布路由时引用 Access List 过滤路由。

被 BGP 引用

```bash
config router bgp
    config neighbor
        edit <ip>
            set distribute-list-in <string>
            set distribute-list-in6 <string>
            set distribute-list-in-vpnv4 <string>
            set distribute-list-out <string>
            set distribute-list-out6 <string>
            set distribute-list-out-vpnv4 <string>
        next
    end
end
```

- `distribute-list-in <string>`：入方向引用 Access List 过滤 IPv4 路由。
- `distribute-list-in6 <string>`：入方向引用 Access List 过滤 IPv6 路由。
- `distribute-list-in-vpnv4 <string>`：入方向引用 Access List 过滤 VPNv4 路由。
- `distribute-list-out <string>`：出方向引用 Access List 过滤 IPv4 路由。
- `distribute-list-out6 <string>`：出方向引用 Access List 过滤 IPv6 路由。
- `distribute-list-out-vpnv4 <string>`：出方向引用 Access List 过滤 VPNv4 路由。

被 Route Map 引用

> **重要**
>
> 被 Route Map 引用时，Access List 中的动作 permit/deny 表示匹配/不匹配，而不是允许/禁用。

```bash
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

- `match-ip-address <string>`：引用 Access List 匹配 IPv4 路由条目。
- `match-ip6-address <string>`：引用 Access List 匹配 IPv6 路由条目。
- `match-ip-nexthop <string>`：引用 Access List 匹配 IPv4 路由下一跳。
- `match-ip6-nexthop <string>`：引用 Access List 匹配 IPv6 路由下一跳。

Prefix List

与 Access List 类似，前缀列表（Prefix List）是用于匹配 IPv4/IPv6 地址 + 掩码的前缀进行路由过滤的简单列表。但前缀列表可以指定要匹配的最小（ge，大于或等于）和最大（le，小于或等于）前缀长度。例如，具有 ge 为 16 的 10.0.0.0/8 前缀将匹配 10.0.0.0/8 网络中的任何具有/16 或更长前缀的内容（10.10.0.0/16 匹配，而 10.10.0.0/12 不匹配）。

配置方法

```bash
IPv4:
config router prefix-list
    edit "prefix-list1"
        config rule
            edit 1
                set action {permit | deny}
                set prefix <IPv4_address>
                set ge <integer>
                set le <integer>
            next
        end
    next
end

IPv6:
config router prefix-list6
    edit "prefix-list-IPv6"
        config rule
            edit 1
                set action {permit | deny}
                set prefix6 <IPv6_address>
                set ge <integer>
                set le <integer>
            next
        end
    next
end
```

引用 Prefix List

被 RIP 引用

```bash
config router rip
    config distribute-list
        edit <id>
            set direction {in | out}
            set listname <string>
        next
    end
end
```

- `listname <string>`：引用 Access List 或 Prefix List 用于出/入方向路由过滤。

被 OSPF 引用

```bash
config router ospf
    set distribute-list-in <string>
end
```

- `distribute-list-in <string>`：引用 Prefix List 用于入方向路由过滤。

被 BGP 引用

```bash
config router bgp
    config neighbor
        edit <ip>
            set prefix-list-in <string>
            set prefix-list-in6 <string>
            set prefix-list-in-vpnv4 <string>
            set prefix-list-out <string>
            set prefix-list-out6 <string>
            set prefix-list-out-vpnv4 <string>
        next
    end
end
```

- `prefix-list-in <string>`：引用 Prefix List 用于该邻居入方向 IPv4 路由过滤。
- `prefix-list-in6 <string>`：引用 Prefix List 用于该邻居入方向 IPv6 路由过滤。
- `prefix-list-in-vpnv4 <string>`：引用 Prefix List 用于该邻居入方向 VPNv4 路由过滤。
- `prefix-list-out <string>`：引用 Prefix List 用于该邻居出方向 IPv4 路由过滤。
- `prefix-list-out6 <string>`：引用 Prefix List 用于该邻居出方向 IPv6 路由过滤。
- `prefix-list-out-vpnv4 <string>`：引用 Prefix List 用于该邻居出方向 VPNv4 路由过滤。

被 Route Map 引用

> **重要**
>
> 被 Route Map 引用时，Prefix List 中的动作 permit/deny 表示匹配/不匹配，而不是允许/禁用。

```bash
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

- `match-ip-address <string>`：引用 Prefix List 匹配 IPv4 路由条目。
- `match-ip6-address <string>`：引用 Prefix List 匹配 IPv6 路由条目。
- `match-ip-nexthop <string>`：引用 Prefix List 匹配 IPv4 路由下一跳。
- `match-ip6-nexthop <string>`：引用 Prefix List 匹配 IPv6 路由下一跳。

AS Path List

AS Path List 使用正则表达式来匹配 BGP 路由的 AS_PATH 属性。可用于从 BGP 邻居筛选入站或出站路由，或者作为 Route Map 中的匹配条件，以匹配 BGP 路由中的 AS_PATH 属性。

配置方法

```bash
config router aspath-list
    edit <name>
        config rule
            edit <id>
                set action {deny | permit}
                set regexp <string>
            next
        end
    next
end
```

引用 AS Path List

被 BGP 引用

```bash
config router bgp
    config neighbor
        edit <ip>
            set filter-list-in <string>
            set filter-list-in6 <string>
            set filter-list-out <string>
            set filter-list-out6 <string>
        next
    end
end
```

- `filter-list-in <string>`：针对该邻居引用 AS Path List 过滤入方向的 IPv4 路由。
- `filter-list-in6 <string>`：针对该邻居引用 AS Path List 过滤入方向的 IPv6 路由。
- `filter-list-out <string>`：针对该邻居引用 AS Path List 过滤出方向的 IPv4 路由。
- `filter-list-out6 <string>`：针对该邻居引用 AS Path List 过滤出方向的 IPv6 路由。

被 Rout Map 引用

```bash
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-as-path <string>
            next
        end
    next
end
```

- `match-as-path <string>`：引用 AS Path List 匹配路由条目。

配置举例

1. 查看设备 BGP 学习到的路由属性，学习到两条 BGP 路由，需要过滤掉 AS Path 起源是 100 的路由。

```bash
get router info bgp neighbors 169.254.1.2 routes
VRF 0 BGP table version is 1, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
Network Next Hop Metric LocPrf Weight RouteTag Path
*> 100.100.100.0/24 169.254.1.2 0 0 300 100 i ←/→     //过滤AS Path起源是100的路由//
*> 200.200.200.0/24 169.254.1.2 0 0 300 200 i ←/→
```

2. 配置 AS Path List，使用正则表达式匹配 AS Path 末尾是 `100` 结尾的 AS Path。

```bash
config router aspath-list
    edit "AS100"
        config rule
            edit 1
                set action permit
                set regexp "_100$"
            next
        end
    next
end
```

3. 在 Route Map 中引用该 AS Path List，禁止接收匹配该 AS Path List 的路由，其他路由可以正常接收。

```bash
config router route-ma
    edit "RM_INBOUND"
        config rule
            edit 1
                set action deny
                set match-as-path "AS100"
            next
            edit 2
                set action permit
            next
        end
    next
end
```

4. 在 BGP 邻居中引用该 Route Map。

```
config router bgp
    config neighbor
        edit "169.254.1.2"
            set route-map-in "RM_INBOUND"
        next
    end
end
```

5. 刷新入方向的 BGP 路由，再次查看学习到的 BGP 路由，可以看到起源为 AS 100 的 BGP 路由已经被过滤掉。

```bash
FortiGate # get router info bgp neighbors 169.254.1.2 routes
VRF 0 BGP table version is 1, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight RouteTag Path
*> 200.200.200.0/24 169.254.1.2 0 0 0 300 200 i ←/1>

Total number of prefixes 1
```

Community List

Community List 使用 Community 字符串过滤/操作 BGP 路由，可以在 Route Map 中使用，匹配在 Community List 中定义的具有特定 Community 属性的路由。

配置方法

```bash
config router community-list
    edit <name>
        set type {standard | expanded}
        config rule
            edit <id>
                set action {deny | permit}
                set regexp <string>
                set match <string>
            next
        end
    next
end
```

- `type {standard | expanded}`：Community List 的类型：
  - standard：匹配精确的 Community 字符，通常用于直接过滤 Community 精确字符。
  - expanded：基于正则表达式进行匹配。在处理较多的 Community 时，非常有用。
- `action {deny | permit}`：action 的 permit/deny 表示匹配/不匹配。
- `regexp <string>`：使用正则表达式匹配 Community List，`type` 配置为 `expanded` 时可用。
- `match <string>`：使用完整 Community 字符方式匹配 Community List，`type` 配置为 `standard` 时可用。

引用 Community List

被 Route Map 引用

```bash
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-community <string>
            next
        end
    next
end
```

- `match-community <string>`：匹配 BGP 路由中的 community 属性。

BFD

- BFD 原理
- BFD 基本配置
- BFD 与静态路由
- BFD 与 OSPF
- BFD 与 BGP
- BFD 多跳模式

BFD 原理

BFD 简介

Bidirectional Forwarding Detection（BFD）是一种可用于快速检测对端网络硬件/软件故障的协议。运行 BFD 的两台路由设备以协商的速率相互发送 BFD 数据包。如果来自启用 BFD 的对端路由器的 BFD 包未能到达，则对端路由器将被本端认为故障。 BFD 将此信息传递给关联的路由协议（如静态路由/OSPF/BGP 等）并更新对应的路由信息。它有助于检测单向设备故障并用于路由协议的快速收敛。

BFD 优点

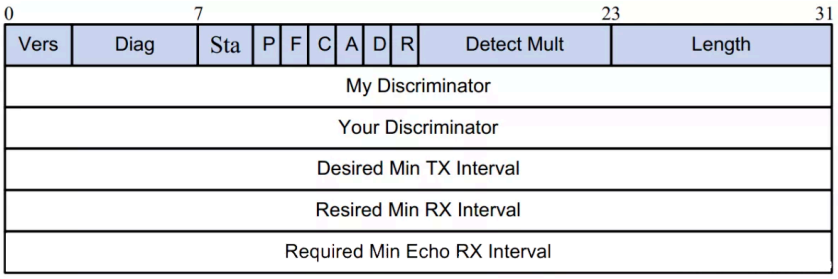
上层协议（如 BGP、OSPF）的检测时间都在 1s 以上，这样的故障检测时间对某些应用来说是不能容忍的。一般动态路由协议检测时间最小只能达到 1 秒的精度，且无法为其它路由协议提供快速故障检测。

- 可以为不同的上层应用服务（如 OSPF、BGP、静态路由等），提供一致的快速故障检测时间。
- 提供小于 1 秒的检测时间，从而加快网络收敛速度，减少应用中断时间。
- 可以对设备间多种类型的双向转发路径进行故障检测，包括直连物理链路、VPN 隧道、多跳路径（FortiOS 7.2.0 开始支持）等。

BFD 缺点

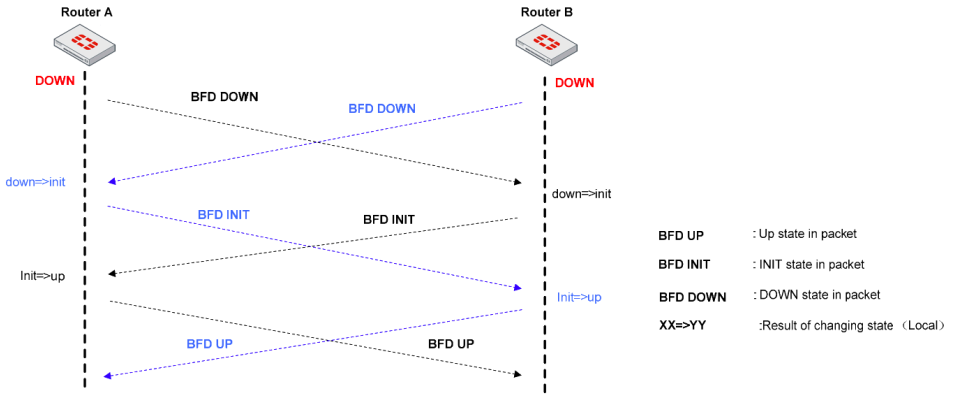
- BFD 进行链路故障的检测依赖于 BFD 会话能够正常协商完成。
- 当创建 BFD 会话过程中链路存在问题时，即 BFD 会话无法通过三次握手状态切换为 UP 时，BFD 无法检测链路状态，此时当上层应用不存在自身保活时，就无法感知链路故障，导致上层应用无法进行相应的处理。
- 由于对网络质量非常敏感，只适用于链路状态较好的环境下，如专线、直连环境。链路质量较差的环境下 BFD 可能会产生误报行为导致线路波动，所以不建议在诸如 Internet 上的 VPN 线路上使用该功能。
- 与动态路由的 Graceful Restart（GR）功能冲突，不建议同时使用。GR 用于 FortiGate 通知邻居自己的路由进程将重启，并且邻居从 FortiGate 学到的路由应在重启期间保留。但 BFD 检测到故障时会将动态路由邻居重置。这种操作上的差异使得在 GR 的同时使用 BFD 会导致功能上的冲突，可能导致 GR 期间流量中断。业界通用的做法是在使用 GR 时关闭 BFD。

BFD 报文格式


0
7
23
31
Vers
Diag
Sta
P
F
C
A
D
R
Detect Mult
Length
My Discriminator
Your Discriminator
Desired Min TX Interval
Resired Min RX Interval
Required Min Echo RX Interval

字段	含义
Vers	BFD 协议版本号，目前版本号为 1
Diag	诊断码，表明发送方最近一次会话 Down 的原因： 0：No Diagnostic 1：Control Detection Time Expired 2：Echo Function Failed 3：Neighbor Signaled Session Down 4：Forwarding Plane Reset 5：Path Down 6：Concatenated Path Down 7：Administratively Down 8：Reverse Concatenated Path Down 9~31：Reserved for future use **For Fortinet:** 0x01: Local inactive timer expired 0x03: Remote peer indicates it's down 0x23: Remote peer indicates restarting
Sta	发送方 BFD 会话当前状态，取值为： 0：AdminDown 1：Down 2：Init 3：Up
P	会话参数变化时置位，参数发生改变时，发送方在 BFD 报文中置该标志，接收方必须立即响应该报文： 1：表示发送系统请求进行连接确认，或者发送请求参数改变的确认 0：表示发送系统不请求确认
F	如果收到的 BFD 控制报文 P 字段置位，则将下一个发送的 BFD 控制报文的 F 字段置位作为应答： 1：表示发送系统响应一个接收到 P 比特为 1 的 BFD 包 0：表示发送系统不响应一个 P 比特为 1 的包
C	转发/控制分离标志，一旦置位，控制平面的变化不影响 BFD 检测，如：控制平面为 IS - IS，当 IS - IS 重启/GR 时，BFD 可以继续监测链路状态： 1：表示发送系统的 BFD 实现不依赖于它的控制平面。即，BFD 报文在转发平面传输，即使控制平面失效，BFD 仍然能够起作用 0：表示 BFD 报文在控制平面传输
A	该字段置位表明报文包含认证部分，会话需要进行认证
D	该字段置位表明发送方希望以查询模式运行，不置位表明不希望以查询模式运行或不支持查询模式
R	保留位，发送时设为 0，接收时忽略该字段
Detect Mult	检测时间倍数，用于检测方计算检测超时时间
Length	BFD 控制报文长度，单位为字节
My Discriminator	BFD 会话连接本地标识符。发送系统产生的一个唯一的、非 0 鉴别值，用来区分一个系统的多个 BFD 会话
Your Discriminator	BFD 会话连接远端标识符。从远端系统接收到的鉴别值，这个域直接返回接收到的“My Discriminator”，如果不知道这个值就返回 0
Desired Min TX Interval	本地支持的最小 BFD 报文接收间隔，单位为毫秒
Required Min RX Interval	发送方支持的最小 BFD 控制报文接收时间间隔，单位为毫秒
Required Min Echo RX Interval	本地支持的最小 Echo 报文接收间隔，单位为微秒 （如果本地不支持 Echo 功能，则设置 0）
Auth Type	认证类型
Auth Len	可选认证部分长度，包括 Auth Type 和 Auth Len 字段，单位为字节

BFD 会话建立



BFD 使用三路握手的机制来建立会话，发送方在发送 BFD 控制报文时会在 Sta 字段填入本地当前的会话状态，接收方根据收到的 BFD 控制报文的 Sta 字段以及本地当前会话状态来进行状态机的迁移，建立会话。Router A 和 Router B 的 BFD 收到上层应用的通知后，发送状态为 DOWN 的 BFD 控制报文。Router B 的 BFD 状态变化同 Router A。

- Router B 收到对端状态为 DOWN 的 BFD 控制报文后，本地会话状态由 DOWN 迁移到 INIT，随后发送的 BFD 控制报文中将 Sta 字段填为 2 表明会话状态为 INIT。Router A 的 BFD 状态变化同 Router B。
- Router A 收到对端状态为 INIT 的 BFD 控制报文后，本地会话状态由 INIT 迁移到 UP，随后发送的 BFD 控制报文中将 Sta 字段填为 3 表明会话状态为 UP。Router B 的 BFD 状态变化同 Router A。
- BFD 双方状态都为 UP，会话成功建立并开始检测链路状态。

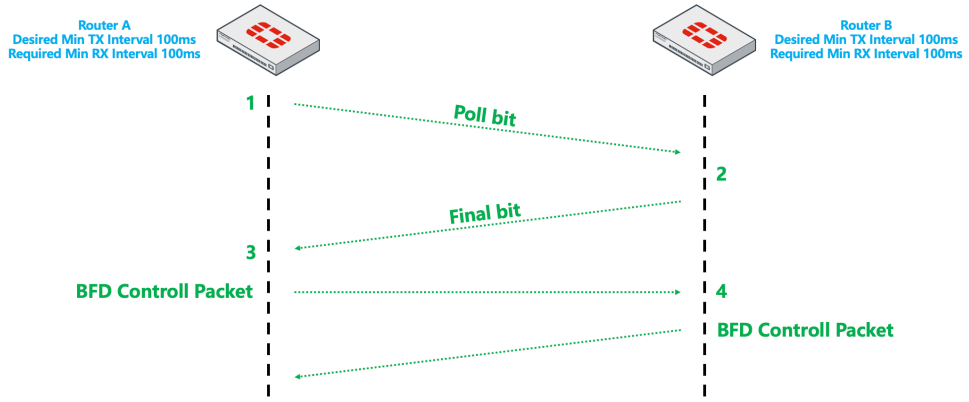
BFD 定时器协商

FortiGate BFD 会话建立前 BFD 控制报文以 5s 的时间间隔周期发送以减小报文流量。在 BFD 会话建立后则以协商到的时间间隔发送 BFD 控制报文以实现快速检测的目的。在 BFD 会话建立的同时，BFD 控制报文发送时间间隔以及检测超时时间也会通过报文交互协商确定。在 BFD 会话有效期间，这些定时器可以随时协商修改而不影响会话状态。

BFD 会话不同方向的定时器协商是分别独立进行的，两端定时器时间可以不同。BFD 控制报文发送时间间隔为本端 Desired Min TX Interval 与对端 Required Min RX Interval 之中的最大值，也就是说比较慢的一方决定了发送频率。

- 检测超时时间为对端 BFD 控制报文中的 Detect Mult 乘以经过协商的对端 BFD 控制报文发送时间间隔。
- 如果加大本端 Desired Min TX Interval，那么本端实际发送 BFD 控制报文的时间间隔必须要等收到对端 F 字段置位的报文后才能改变，这是为了确保在本端加大 BFD 控制报文发送时间间隔前对端已经加大了检测时间，否则可能导致对端检测定时器错误超时。
- 如果减小本端 Required Min RX Interval，那么本端检测超时时间必须要等收到对端 F 字段置位的报文后才能改变，这是为了确保在本端减小检测超时时间前对端已经减小了 BFD 控制报文发送间隔时间，否则可能导致本端检测定时器错误超时。
- 如果减小本端 Desired Min TX Interval，本端 BFD 控制报文发送时间间隔将会立即减小；加大 Required Min RX Interval，本端检测超时时间将会立即加大。

BFD 协商举例



1. Router A 与 Router B 创建 BFD 配置，双方的 Desired Min TX Interval 和 Required Min RX Interval 都为 100ms，Detect Mult 都为 3。根据上面介绍的 BFD 定时器协商规则，Router A 的发送时间间隔为 Router A 的 TX 与 Router B 的 RX 中的最大值：100ms，Router B 的发送时间间隔也是 100ms，双方的检测超时时间都为 100ms * Detect Mult (3) = 300ms。

2. 此时将 Router A 的 TX 和 RX 加大到 150ms。
   - Router A 比较本端的 RX（150ms）和 Router B 的 TX（100ms），从而将本端检测时间改为 450ms。同时向对端发送 P 字段置位的 BFD 控制报文（TX 和 RX 均为 150ms）。
   - Router B 收到报文后，给 Router A 回复 F 字段置位的 BFD 控制报文（TX 和 RX 均为 100ms）。同时将收到报文中的 RX 与本端的 TX 进行比较，由于对端 Router A TX 较大，故 Router B 的发送间隔改为 150ms。经过比较本端 RX 和对端的 TX，从而将检测超时时间也增大到 450ms。
   - Router A 收到对端发来 F 字段置位的控制报文。根据报文中的 RX 与本端的 TX 进行比较计算出新的时间间隔为 150ms。
   - 定时器协商完成，双方的发送间隔和检测超时时间分别为 150ms 和 450ms。

BFD 基本配置

BFD 参数配置

1. BFD 支持基于 FortiGate 的 2 个层面进行参数的配置。
   - FortiGate 全局（基于每 VDOM）：

```bash
config system settings
    set bfd {enable | disable}
    set bfd-desired-min-tx <ms>
    set bfd-required-min-rx <ms>
    set bfd-detect-mult <multiplier>
    set bfd-dont-enforce-src-port {enable | disable}
end
```

     - `enable`：开启所有接口下的 BFD 配置（如果接口下的 BFD 配置为 global）。
     - `disable`：关闭所有接口下的 BFD 配置（如果接口下的 BFD 配置为 global）。
     - `bfd-desired-min-tx`：发送方支持的最小 BFD 控制报文发送时间间隔，单位为毫秒，默认为 250ms。
     - `bfd-required-min-rx`：发送方支持的最小 BFD 控制报文接收时间间隔，单位为毫秒，默认为 250ms。
     - `bfd-detect-mult`：检测时间倍数。
     - `bfd-dont-enforce-src-port`：开启后不强制验证 BFD 数据包的源端口（默认关闭）。
   - FortiGate 接口，接口下的 BFD 开关有三种模式：

```bash
config system interface
    edit <interface-name>
        set bfd {global | enable | disable}
        set bfd-desired-min-tx <ms>
        set bfd-required-min-rx <ms>
        set bfd-detect-mult <multiplier>
    next
end
```

     - `global`：继承 config system settings 下的 bfd 开关（默认配置）。
     - `enable`：覆盖 config system settings 下的 bfd 开关，在此接口下开启 BFD。
     - `disable`：覆盖 config system settings 下的 bfd 开关，在此接口下关闭 BFD。
     - `bfd-desired-min-tx`：发送方支持的最小 BFD 控制报文发送时间间隔，单位为毫秒，默认为 250ms。
     - `set bfd-required-min-rx`：发送方支持的最小 BFD 控制报文接收时间间隔，单位为毫秒，默认为 250ms。
     - bfd-detect-mult：检测时间倍数。
2. 较低层级（接口）的配置可以覆盖较高层级（全局）的配置，例如在特定接口上配置 BFD 为 enable 后，配置的 BFD 参数会优先于全局下配置的 BFD 参数。

> **重要**
>
> FortiGate 的 BFD 目前不支持 BFD Echo 模式。

BFD 邻居配置

当 BFD 被静态路由引用时，需要通过配置 BFD 邻居来让 BFD 协议获取 BFD 邻居。

> **重要**
>
> 当 BFD 被动态路由如 OSPF、BGP 引用时，可以通过 OSPF 和 BGP 的邻居信息获取到 BFD 的邻居信息，不需要配置 BFD 邻居。

```bash
config router {bfd | bfd6}
    config neighbor
        edit <IP-address>
            set interface <interface-name>
        next
```

```
    end
end
```

BFD 多跳配置

重要

该功能在 7.2.0 GA 及以后版本支持。

1. 默认配置下，不论是被 OSPF、BGP 还是静态路由引用，FortiGate 的 BFD 邻居只支持直连网段的设备。如果 FortiGate 的 BFD 邻居不是直连网络的设备，需要在 BFD 下配置 multihop-template，指定 BFD 流量的源（自身 IP）和目的（邻居 IP），这里配置的 BFD 邻居可以是多跳的不同网段的设备 IP。
   - 当 BFD 被静态路由引用时，src 和 dst 需要配置 Host 级别的 netmask，IPv4 为 32 位，IPv6 为 128 位。
   - 当 BFD 被动态路由（如 OSPF 和 BGP）引用时，src 和 dst 可以配置为网段（包含 BFD 本地地址和邻居地址），如 192.168.13.0/24，2100::/64，邻居地址通过动态路由协议获取。这样可以方便的配置多个 BFD 多跳邻居。
2. 同时可以在 multihop-template 中配置 BFD 的相关参数与认证功能，这些 BFD 参数（匹配模板的 BFD 流量）优先于接口及全局级别。
3. 在只有 BFD 多跳邻居的情况下，配置 multihop-template 后，不需要再配置 BFD neighbor。

```bash
config router {bfd | bfd6}
    config multihop-template
        edit <ID>
            set src <IP address/netmask>
            set dst <IP address/netmask>
            set bfd-desired-min-tx <integer>
            set bfd-required-min-rx <integer>
            set bfd-detect-mult <integer>
            set auth-mode {none | md5}
            set md5-key <password>
        next
    end
end
```

BFD 与静态路由

当静态路由开启 BFD 后，BFD 检测到 BFD 邻居故障时，BFD 会从路由表中删除静态路由，如果路由的目的地恢复，则 BFD 将路由在路由表中恢复。

举例：添加两条目的相同但下一跳不同的启用 BFD 的静态路由。如果其中一条路由具有较高优先级，则所有匹配流量都使用该路由。如果 BFD 检测到高优先级路由到网关的链路故障，则将高优先级路由从路由表中删除，所有匹配的流量都使用低优先级路由。如果较高优先级路由到网关的链路恢复正常，BFD 会将该路由重新添加到路由表中，所有匹配的流量都会切换为使用较高优先级路由。

FortiGate 的 BFD 同时支持 IPv4 和 IPv6 的静态路由。

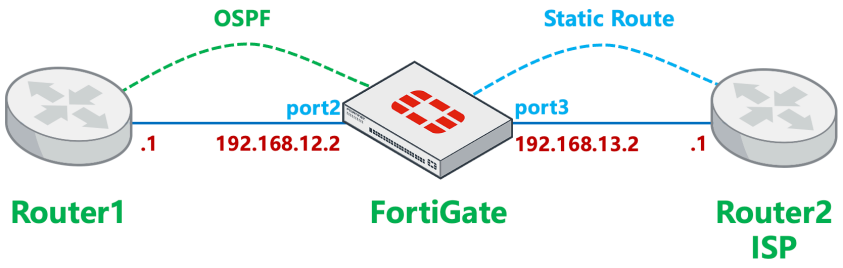
基本配置

当 BFD 被静态路由引用时，需要通过配置 BFD 邻居来让 BFD 协议获取 BFD 邻居信息。

```bash
config router {bfd | bfd6}
    config neighbor
        edit <IP-address>
            set interface <interface-name>
        next
    end
end
```

```bash
config router {static | static6}
    edit <sequence-number>
        set bfd {enable | disable}
        set device <gateway-out-interface>
    next
end
```

网络拓扑



- FortiGate 与 Router1 建立 OSPF 邻居，并重发布静态路由。
- FortiGate 与 Router2（ISP）之间使用静态路由，并开启 BFD，当 BFD 检测 UP 时，静态路由存在于 FortiGate 的路由表，会被重发布到 OSPF；当 BFD 检测 DOWN 时，删除 FortiGate 上开启 BFD 的静态路由，不会重分发到 OSPF。
- Router1 配置与 FortiGate 的 OSPF 邻居。（略）

配置步骤

1. 配置网络接口 IP（略）。

2. 配置 FortiGate 与 Router1 建立 OSPF 邻居，并重发布静态路由。

```bash
config router ospf
    set router-id 192.168.12.2
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 192.168.12.0 255.255.255.0
        next
```

```
    end
    config redistribute "static"
        set status enable
    end
end
```

3. 在 FortiGate 的 port3 上开启 BFD，配置 BFD 邻居为 Router2（ISP），并配置静态路由，去往 ISP 的两条静态路由开启 BFD，去往内部网络的路由不开启 BFD。

```bash
config system interface
    edit "port3"
        set ip 192.168.13.2 255.255.255.0
        set bfd enable
    next
end

config router bfd
    config neighbor
        edit 192.168.13.1
            set interface "port3"
        next
    end
end

config router static
    edit 1
        set dst 223.5.5.5 255.255.255.255
        set gateway 192.168.13.1
        set device "port3"
        set bfd enable
    next
    edit 2
        set dst 114.114.114.114 255.255.255.255
        set gateway 192.168.13.1
        set device "port3"
    next
    edit 3
        set dst 10.10.1.0 255.255.255.0
        set gateway 192.168.12.1
        set device "port2"
    next
end
```

结果验证

1. 在 Router2 还未配置 BFD 的状态下，查看 FortiGate 的 BFD 邻居和静态路由状态。BFD 邻居状态为 DOWN，开启 BFD 的静态路由未放入路由表，未开启 BFD 的静态路由被放入路由表。

> **重要**
>
> 与动态路由（如 OSPF 和 BGP）不同，静态路由引用 BFD 时，静态路由是否放入路由表依赖于相关的 BFD 邻居状态，BFD 邻居状态为 UP 时将开启 BFD 的相关静态路由放入路由表，BFD 邻居状态为 DOWN 时将开启 BFD 的相关静态路由从路由表中移除（具体信息可见本章节最后的注意事项）。

```bash
FortiGate # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.13.2    192.168.13.1    DOWN        port3           1/1

FortiGate # get router info routing-table all
Routing table for VRF=0
S       10.10.1.0/24 [10/0] via 192.168.12.1, port2, [1/0]
S       114.114.114.114/32 [10/0] via 192.168.13.1, port3, [1/0]
C       192.168.12.0/24 is directly connected, port2
C       192.168.13.0/24 is directly connected, port3
```

2. 在 Router2（ISP）上开启 BFD 功能后，再次查看 FortiGate 的 BFD 邻居和静态路由状态。BFD 邻居状态为 UP，开启 BFD 的静态路由被放入路由表。

```bash
FortiGate # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.13.2    192.168.13.1    UP          port3           1/1

FortiGate # get router info routing-table all
Routing table for VRF=0
S       10.10.1.0/24 [10/0] via 192.168.12.1, port2, [1/0]
S       114.114.114.114/32 [10/0] via 192.168.13.1, port3, [1/0]
C       192.168.12.0/24 is directly connected, port2
C       192.168.13.0/24 is directly connected, port3
S       223.5.5.5/32 [10/0] via 192.168.13.1, port3, [1/0]
```

3. FortiGate 上的相关 BFD 日志与抓包如下所示，BFD 邻居状态由 Init 变为 UP。

```bash
date=2023-10-19 time=11:28:21 eventtime=1697686101603684308 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.13.2→192.168.13.1,49152,port3,1]:
state INIT → UP local_diag=0×00"
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-10-19 11:28:11.613310	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
2	2023-10-19 11:28:16.618795	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
3	2023-10-19 11:28:21.602440	192.168.13.1	192.168.13.2	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
4	2023-10-19 11:28:21.603008	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
5	2023-10-19 11:28:21.603554	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
6	2023-10-19 11:28:21.603708	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
7	2023-10-19 11:28:21.834751	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
8	2023-10-19 11:28:21.835450	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
9	2023-10-19 11:28:22.066236	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
10	2023-10-19 11:28:22.066481	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00

4. 模拟 FortiGate 与 Router2 之间的线路异常，查看 FortiGate 的 BFD 邻居和静态路由状态。BFD 邻居状态为 DOWN，开启 BFD 的静态路由未放入路由表，未开启 BFD 的静态路由仍然存在于路由表。

```bash
FortiGate # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.13.2    192.168.13.1    DOWN        port3           1/1

FortiGate # get router info routing-table all
Routing table for VRF=0
S       10.10.1.0/24 [10/0] via 192.168.12.1, port2, [1/0]
S       114.114.114.114/32 [10/0] via 192.168.13.1, port3, [1/0]
C       192.168.12.0/24 is directly connected, port2
C       192.168.13.0/24 is directly connected, port3
```

5. FortiGate 上的相关 BFD 日志与抓包如下所示，BFD 邻居状态由 UP 变为 DWON，local_diag = 0x01（表示 Control Detection Time Expired）。

```bash
date=2023-10-19 time=15:06:50 eventtime=1697699210346884501 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.13.2→192.168.13.1,49152,port3,1]:
state UP → DOWN local_diag=0×01"
```

29	2023-10-19 15:06:49.594358	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
30	2023-10-19 15:06:49.754854	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
31	2023-10-19 15:06:49.985309	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
32	2023-10-19 15:06:50.225972	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
33	2023-10-19 15:06:50.346924	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
34	2023-10-19 15:06:55.353093	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00

6. 恢复 FortiGate 与 Router2 之间的线路，查看 FortiGate 的 BFD 邻居和静态路由状态。BFD 邻居状态变为 UP，开启 BFD 的静态路由被放入路由表。

```bash
FortiGate # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.13.2    192.168.13.1    UP          port3           1/1

FortiGate # get router info routing-table all
Routing table for VRF=0
S       10.10.1.0/24 [10/0] via 192.168.12.1, port2, [1/0]
S       114.114.114.114/32 [10/0] via 192.168.13.1, port3, [1/0]
C       192.168.12.0/24 is directly connected, port2
C       192.168.13.0/24 is directly connected, port3
S       223.5.5.5/32 [10/0] via 192.168.13.1, port3, [1/0]
```

7. FortiGate 上的相关 BFD 日志与抓包如下所示，BFD 邻居状态由 INIT 变为 UP。

```bash
date=2023-10-19 time=15:54:03 eventtime=1697702043275319216 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.13.2→192.168.13.1,49152,port3,1]:
```

```
state INIT → UP local_diag=0×00"
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-10-19 15:53:51.267802	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
2	2023-10-19 15:53:53.389096	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
3	2023-10-19 15:53:58.395090	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
4	2023-10-19 15:54:03.274526	192.168.13.1	192.168.13.2	BFD Control	66	Diag: Control Detection Time Expired, State: Down, Flags: 0x00
5	2023-10-19 15:54:03.274666	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
6	2023-10-19 15:54:03.275247	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
7	2023-10-19 15:54:03.276714	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
8	2023-10-19 15:54:03.477716	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
9	2023-10-19 15:54:03.498376	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00

8. 关闭 Router2（由 FortiGate 模拟）的 BFD 功能，Router2 会向 FortiGate 发送 Administratively Down 类型的 BFD 通知。

57	2023-10-19 16:45:49.942872	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
58	2023-10-19 16:45:49.983121	192.168.13.2	192.168.13.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
59	2023-10-19 16:45:50.153884	192.168.13.1	192.168.13.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
60	2023-10-19 16:45:50.213521	192.168.13.1	192.168.13.2	BFD Control	66	Diag: Administratively Down, State: AdminDown, Flags: 0x00
61	2023-10-19 16:45:50.213708	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Neighbor Signaled Session Down, State: Down, Flags: 0x00
62	2023-10-19 16:45:55.219047	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Neighbor Signaled Session Down, State: Down, Flags: 0x00
63	2023-10-19 16:46:00.224542	192.168.13.2	192.168.13.1	BFD Control	66	Diag: Neighbor Signaled Session Down, State: Down, Flags: 0x00

9. 查看 FortiGate 的 BFD 邻居和静态路由状态。BFD 邻居状态变为 DOWN，但开启 BFD 的静态路由没有从路由表中移除。

```bash
FortiGate # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.13.2    192.168.13.1    DOWN        port3           1/1

FortiGate # get router info routing-table all
Routing table for VRF=0
S       10.10.1.0/24 [10/0] via 192.168.12.1, port2, [1/0]
S       114.114.114.114/32 [10/0] via 192.168.13.1, port3, [1/0]
C       192.168.12.0/24 is directly connected, port2
C       192.168.13.0/24 is directly connected, port3
C       192.168.100.0/24 is directly connected, port1
S       223.5.5.5/32 [10/0] via 192.168.13.1, port3, [1/0]
```

.0. FortiGate 上的相关 BFD 日志与抓包如下所示，BFD 邻居状态由 UP 变为 DOWN，local_diag = 0x23（表示 Remote peer indicates restarting）。

```bash
date=2023-10-19 time=16:45:50 eventtime=1697705150213670782 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.13.2→192.168.13.1,49152,port3,1]:
state UP → DOWN local_diag=0×23"
```

注意事项

1. 与动态路由协议（如 OSPF 和 BGP）不同，静态路由没有对等体邻居发现的方法。因此，当配置了 BFD 时，静态路由网关的可达性完全取决于指定邻居的 BFD 会话的状态。
   - 当 BFD 邻居状态为 DOWN，就认为静态路由的网关不可达，相关静态路由不会被安装在路由表中（不需要 BFD 邻居状态有一个从 UP → DOWN 的过程）。
   - 当 BFD 邻居状态为 UP，就认为静态路由的网关可达，相关静态路由会被安装在路由表中。
   - 这是为了防止在 FortiGate 在 HA 环境下，BFD 邻居状态为 DOWN 时，主备切换后，新主机将原主机已经从路由表中移除的静态路由错误地放入路由表。
2. 由于以上 BFD 与静态路由之间的特性，建议项目实施中配置静态路由与 BFD 时，在对端设备还未配置 BFD 的情况下，先不要在 FortiGate 的静态路由中开启 BFD，等待对端设备完成 BFD 配置后，再将 FortiGate 的静态路由开启 BFD，这样可以防止上述结果验证步骤 1 中初始状态下静态路由无法放入路由表的情况出现。
3. 对于动态路由（如 OSPF 和 BGP），BFD 必须要有一个从 UP → DOWN 的状态转换，才会执行邻居关系的重置。
4. 当开启 BFD 的静态路由已经放入路由表后，邻居不是因为链路故障 DOWN，例如对端设备配置关闭 BFD 功能（向 FortiGate 发送了 Administratively Down 类型的 BFD 通知），或是修改 FortiGate 自身的 BFD 配置导致 BFD 邻居 DOWN，这种情况下 FortiGate 不会将静态路由从路由表中移除。

BFD 与 OSPF

基本配置

> **重要**
>
> 必须配置并建立 OSPF 邻居后，BFD 才能从 OSPF 获取 BFD 邻居信息，从而发送 BFD 报文。

1. 在 OPSF 中开启 BFD。

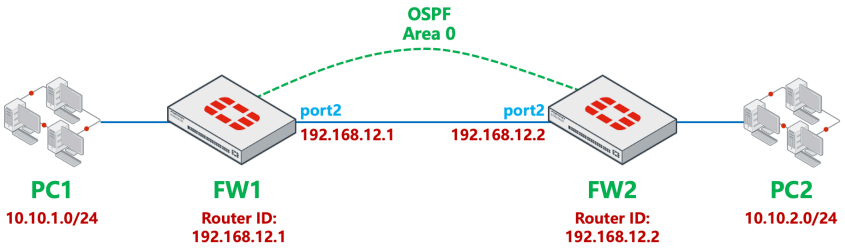
```bash
config router {ospf | ospf6}
    set bfd {enable | disable}
end
```

2. 也可以在特定的 OSPF 接口下开启 BFD。

```bash
config router {ospf | ospf6}
    set bfd enable
    config {ospf-interface | ospf6-interface}
        edit <ID>
            set bfd {global | enable | disable}
            set area-id <IP address>
        next
    end
end
```

- global：继承 config router ospf/ospf6 下的 BFD 开关配置（注意不是继承 config system setting 下的 BFD 开关配置）。
- enable：仅在此接口下开启 BFD，覆盖 config router ospf/ospf6 下的 BFD 开关配置。
- disable：在此接口下关闭 BFD，覆盖 config router ospf/ospf6 下的 BFD 开关配置。

网络拓扑



- FW1 与 FW2 建立 IPv4 的 OSPF 邻居。
- BFD 检测失败时，重置 OSPF 邻居。

配置步骤

1. 基本网络配置（略）。

2. 在 FW1 和 FW2 的 port2 上分别开启 BFD。

```bash
config system interface
    edit "port2"
        set bfd enable
    next
end
```

3. 在 FW1 和 FW2 上分别配置 OSPF，配置 OSPF 邻居，宣告 IPv4 内网网段，在 OSPF 接口 port2 上开启 BFD。

```bash
FW1:
config router ospf
    set router-id 192.168.12.1
    config area
        edit 0.0.0.0
        next
```

```
    end
    config ospf-interface
        edit "port2"
            set interface "port2"
            set bfd enable
        next
    end
    config network
        edit 1
            set prefix 192.168.12.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

FW2:

```
config router ospf
    set router-id 192.168.12.2
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "port2"
            set interface "port2"
            set bfd enable
        next
    end
    config network
        edit 1
            set prefix 192.168.12.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

结果验证

1. 在 FW1 上查看 BFD 邻居状态均为 UP（FW2 略）。

```
FW1 # get router info bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=192.168.12.1     dst=192.168.12.2     ct=04 ifi=4 type=S
```

```
FW1 # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.12.1    192.168.12.2    UP          port2           1/4
```

2. 在 FW1 上查看 BFD 相关日志与抓包（FW2 略），BFD 状态由 INIT 变为 UP。

```
date=2023-10-20 time=15:45:27 eventtime=1697787926746001101 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49152,port2,1]:
state DOWN → UP local_diag=0×00"
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-10-20 15:45:18.856104	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Down, Flags: 0x00
2	2023-10-20 15:45:23.862082	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Down, Flags: 0x00
3	2023-10-20 15:45:26.533192	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Down, Flags: 0x00
4	2023-10-20 15:45:26.533698	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
5	2023-10-20 15:45:26.744988	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
6	2023-10-20 15:45:26.746187	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
7	2023-10-20 15:45:26.747592	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
8	2023-10-20 15:45:26.967934	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
9	2023-10-20 15:45:26.978397	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
10	2023-10-20 15:45:27.160012	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00

3. 在 FW1 上查看 OSPF 邻居与路由状态，邻居状态正常建立，OSPF 路由学习正常。

```bash
FW1 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.12.2      1   Full/DR         00:00:31    192.168.12.2    port2

FW1 # get router info routing-table ospf
Routing table for VRF=0
O       10.10.2.0/24 [110/2] via 192.168.12.2, port2, 00:02:50, [1/0]
```

4. 模拟线路问题 10s（10s 后恢复）导致 BFD 中断，查看 FW1 上 BFD 邻居消失（因为 BFD DOWN 后重置了 OSPF 连接，BFD 邻居信息依赖于 OSPF 已建立的邻居信息）。

```bash
FW1 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
```

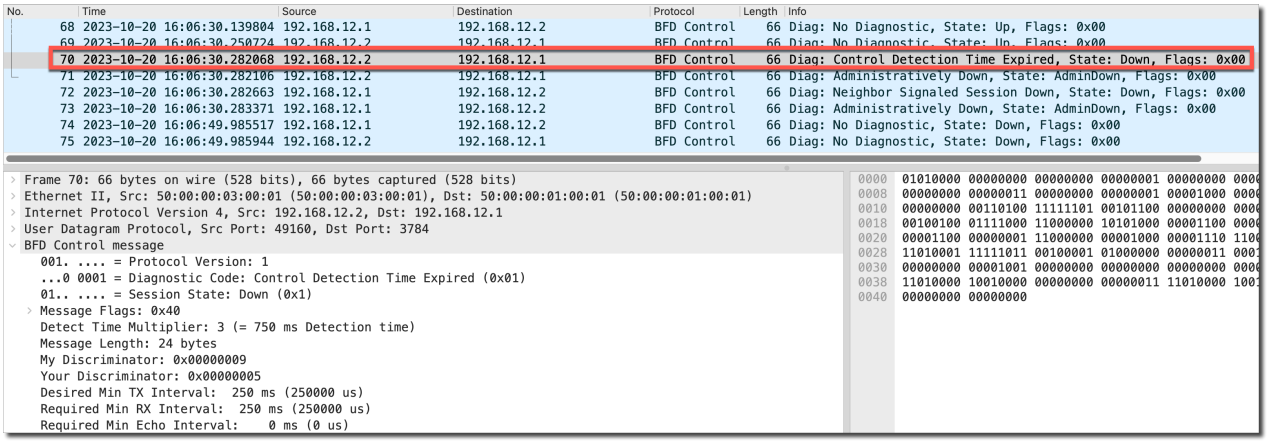
5. 在 FW1 上查看 BFD 相关日志，BFD DOWN 后，对应的 OSPF 邻居关系跟随 DOWN。local_diag = 0x01（表示 Control Detection Time Expired）。

```bash
date=2023-10-20 time=16:06:30 eventtime=1697789190283253338 tz="+0800" logid="0103020302" type="event" subtype="router"
level="warning" vd="root" logdesc="OSPF neighbor status changed" msg="OSPF: %OSPF-5-ADJCHANGE: neighbor
port2:192.168.12.1-192.168.12.2 Down "
date=2023-10-20 time=16:06:30 eventtime=1697789190282633679 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49156,port2,5]:
state UP → DOWN local_diag=0×01"
```

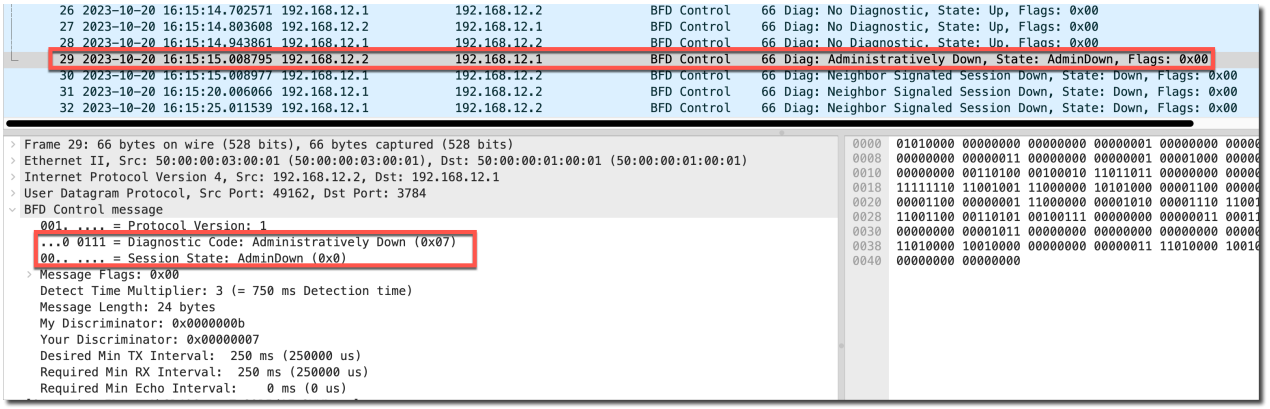
6. 在 FW1 上查看 OSPF 邻居状态，邻居断开，并开始重建 OSPF 邻居。

```bash
FW1 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.12.2      1   Init/DROther    00:00:39    192.168.12.2    port2

FW1 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.12.2      1   Full/Backup     00:00:37    192.168.12.2    port2
```

7. 抓包查看 BFD DOWN 的报文，FW2 发现检测超时，向 FW1 发送 BFD Time Expired 报文，随后两端互相发送 Admin DOWN 的 BFD 控制报文，随后双向 BFD 字段中的 Diagnostic Code 为空，状态为 DOWN。



8. 关闭 FW2（由 FortiGate 模拟）的 BFD 功能，FW2 会向 FW1 发送 Administratively Down 类型的 BFD 通知。



9. 查看 FortiGate 的 BFD 邻居和 OSPF 状态。BFD 邻居状态变为 DOWN，但开启 BFD 的 OSPF 邻居没有重建。

```bash
FW1 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.12.1    192.168.12.2    DOWN        port2           7/11

FW1 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.12.2      1   Full/DR         00:00:35    192.168.12.2    port2
```

0. FW1 上的相关 BFD 日志如下所示，BFD 邻居状态由 UP 变为 DOWN，local_diag = 0x23（表示 Remote peer indicates restarting），路由事件日志中没有 OSPF 邻居断开的日志。

```bash
date=2023-10-20 time=16:15:15 eventtime=1697789715008917984 tz="+0800" logid="0103020304" type="event" subtype="router" level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49158,port2,7]: state UP → DOWN local_diag=0×23"
```

注意事项

1. Admin 类型的 Down 不会导致 OSPF 邻居重建。
2. BFD Down 后触发了 OSPF 邻居重建，如果 BFD 一直没有 UP，则重建后的 OSPF 邻居不会受任何影响，除非再次发生了 BFD 从 UP 到 Down 的状态转换（静态路由不是这样）。

BFD 与 BGP

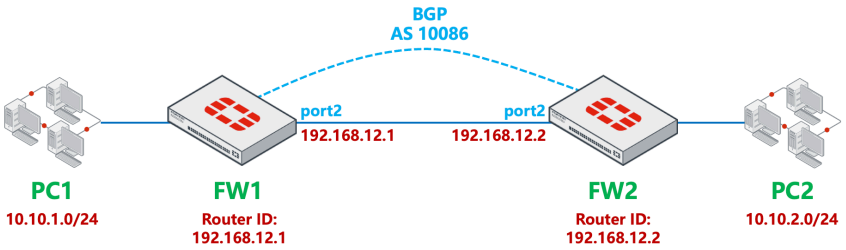
基本配置

> **重要**
>
> 必须配置并建立 BGP 邻居后，BFD 才能从 BGP 获取 BFD 邻居信息，从而发送 BFD 报文。

在 BGP 中开启 BFD。

```bash
config router bgp
    config neighbor
        edit <neighbor-IP-address>
            set bfd {enable | disable}
        next
    end
end
```

网络拓扑



- FW1 与 FW2 建立 IPv4 的 IBGP 邻居。
- BFD 检测失败时，重置 BGP 邻居。

配置步骤

1. 基本网络配置（略）。

2. 在 FW1 和 FW2 的 port2 上分别开启 BFD。

```bash
config system interface
    edit "port2"
        set bfd enable
    next
end
```

3. 在 FW1 和 FW2 上分别配置 BGP，配置 BGP 邻居，宣告 IPv4 内网网段，在 BGP 邻居中开启 BFD。

```bash
FW1:
config router bgp
    set as 10086
    set router-id 192.168.12.1
    config neighbor
        edit "192.168.12.2"
            set bfd enable
            set remote-as 10086
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
end
```

```bash
FW2:
config router bgp
    set as 10086
    set router-id 192.168.12.2
    config neighbor
        edit "192.168.12.1"
            set bfd enable
            set remote-as 10086
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
end
```

结果验证

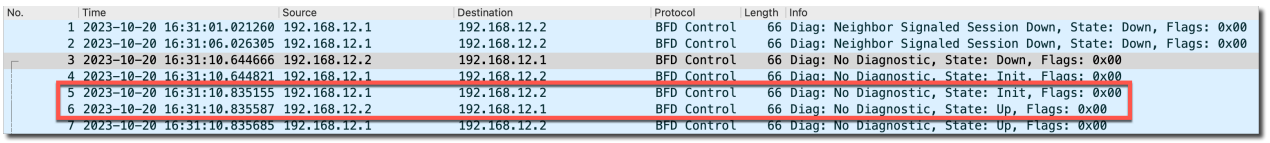
1. 在 FW1 上查看 BFD 邻居状态均为 UP（FW2 略）。

```bash
FW1 # get router info bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=192.168.12.1     dst=192.168.12.2     ct=08 ifi=4 type=SM
```

```bash
FW1 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.12.1    192.168.12.2    UP          port2           8/12
```

2. 在 FW1 上查看 BFD 相关日志与抓包（FW2 略），BFD 状态由 INIT 变为 UP。

```bash
date=2023-10-20 time=16:31:11 eventtime=1697790670835673328 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49159,port2,8]:
state INIT → UP local_diag=0×00"
```



No.	Time	Source	Destination	Protocol	Length	Info
1	2023-10-20 16:31:01.021260	192.168.12.1	192.168.12.2	BFD Control	66	Diag: Neighbor Signaled Session Down, State: Down, Flags: 0x00
2	2023-10-20 16:31:06.026305	192.168.12.1	192.168.12.2	BFD Control	66	Diag: Neighbor Signaled Session Down, State: Down, Flags: 0x00
3	2023-10-20 16:31:10.644666	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Down, Flags: 0x00
4	2023-10-20 16:31:10.644821	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
5	2023-10-20 16:31:10.835155	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Init, Flags: 0x00
6	2023-10-20 16:31:10.835587	192.168.12.2	192.168.12.1	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00
7	2023-10-20 16:31:10.835685	192.168.12.1	192.168.12.2	BFD Control	66	Diag: No Diagnostic, State: Up, Flags: 0x00

3. 在 FW1 上查看 BGP 邻居与路由状态，邻居状态正常建立，BGP 路由学习正常。

```bash
FW1 # get router info bgp summary
VRF 0 BGP router identifier 192.168.12.1, local AS number 10086
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.2 4      10086      11      11        1    0    0 00:06:44        1
Total number of neighbors 1

FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 192.168.12.2 (recursive is directly connected, port2), 00:06:27, [1/0]
```

4. 模拟线路问题 10s（10s 后恢复）导致 BFD 中断，中断过程中查看 FW1 上 BFD 邻居消失（因为 BFD DOWN 后重置了 BGP 连接，BFD 邻居信息依赖于 BGP 已建立的邻居信息）。

```bash
FW1 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
```

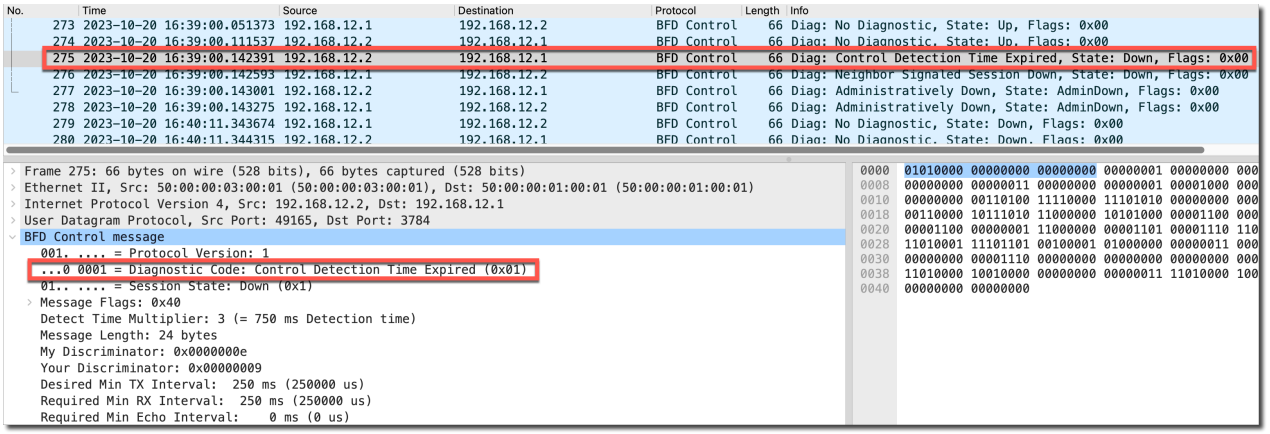
5. 在 FW1 上查看 BFD 相关日志，BFD DOWN 后，对应的 BGP 邻居关系跟随 DOWN。local_diag = 0x01（表示 Control Detection Time Expired）。

```
date=2023-10-20 time=16:39:00 eventtime=1697791140142880643 tz="+0800" logid="0103020300" type="event" subtype="router"
level="warning" vd="root" logdesc="BGP neighbor status changed" msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 192.168.12.2
Down BFD Down; User reset added"
date=2023-10-20 time=16:39:00 eventtime=1697791140142553520 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49160,port2,9]:
state UP → DOWN local_diag=0×01"
```
bash

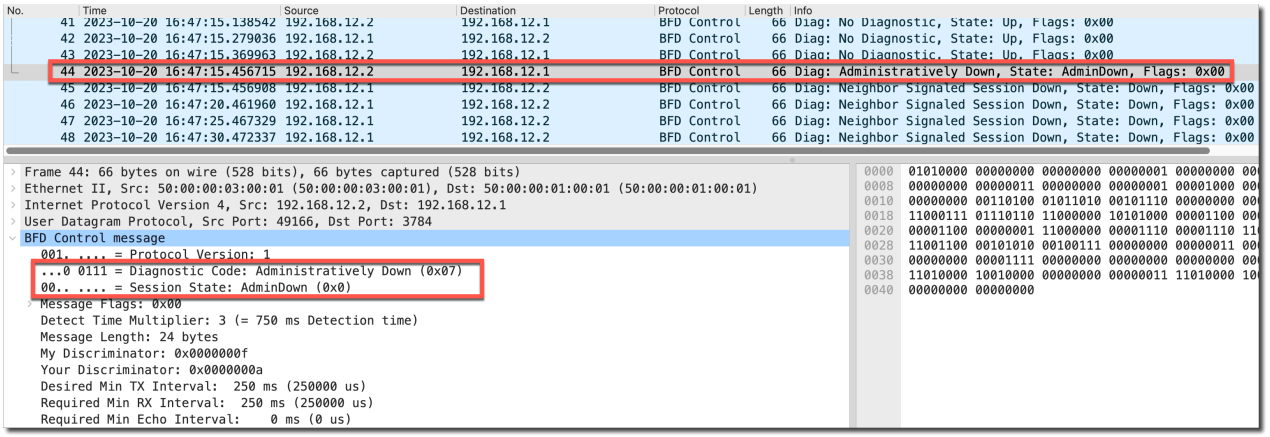
6. 在 FW1 上查看 BGP 邻居状态，邻居断开，并开始重建 BGP 邻居。

```
FW1 # get router info bgp summary
VRF 0 BGP router identifier 192.168.12.1, local AS number 10086
BGP table version is 3
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.2 4      10086      18      19        0    0    0    never Connect
Total number of neighbors 1

FW1 # get router info bgp summary
VRF 0 BGP router identifier 192.168.12.1, local AS number 10086
BGP table version is 3
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.2 4      10086      21      22        3    0    0 00:00:27        1
Total number of neighbors 1
```
bash

7. 抓包查看 BFD DOWN 的报文，FW2 发现检测超时，向 FW1 发送 BFD Time Expired 报文，随后两端互相发送 Admin DOWN 的 BFD 控制报文，随后双向 BFD 字段中的 Diagnostic Code 为空，状态为 DOWN。



8. 关闭 FW2（由 FortiGate 模拟）的 BFD 功能，FW2 会向 FW1 发送 Administratively Down 类型的 BFD 通知。



9. 查看 FortiGate 的 BFD 邻居和 BGP 状态。BFD 邻居状态变为 DOWN，但开启 BFD 的 BGP 邻居没有重建。

```bash
FW1 # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
192.168.12.1    192.168.12.2    DOWN        port2           10/15

FW1 # get router info bgp summary
VRF 0 BGP router identifier 192.168.12.1, local AS number 10086
BGP table version is 3
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.12.2 4      10086      30      31        3    0    0 00:08:37        1
Total number of neighbors 1
```

10. FW1 上的相关 BFD 日志如下所示，BFD 邻居状态由 UP 变为 DOWN，local_diag = 0x23（表示 Remote peer indicates restarting），路由事件日志中没有 BGP 邻居断开的日志。

```bash
date=2023-10-20 time=16:47:15 eventtime=1697791635456872179 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[192.168.12.1→192.168.12.2,49161,port2,A]:
state UP → DOWN local_diag=0×23"
```

注意事项

1. Admin 类型的 Down 不会导致 BGP 邻居重建。
2. BFD Down 后触发了 BGP 邻居重建，如果 BFD 一直没有 UP，则重建后的 BGP 邻居不会受任何影响，除非再次发生了 BFD 从 UP 到 Down 的状态转换（静态路由不是这样）。

BFD 多跳模式

重要

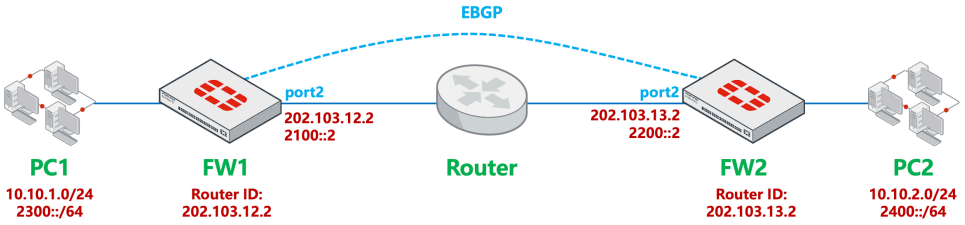
该功能在 7.2.0 GA 及以后版本支持。

功能介绍

1. 默认配置下，不论是被 OSPF、BGP 还是静态路由引用，FortiGate 的 BFD 邻居只支持直连网段的设备。如果 FortiGate 的 BGP 邻居不是直连网络的设备，需要在 BFD 下配置 multihop-template，指定 BFD 流量的源（自身 IP）和目的（邻居 IP），这里配置的 BFD 邻居可以是多跳的不同网段的设备 IP。
   - 当 BFD 被静态路由引用时，src 和 dst 需要配置 Host 级别的 netmask，IPv4 为 32 位，IPv6 为 128 位。
   - 当 BFD 被动态路由（如 OSPF 和 BGP）引用时，src 和 dst 可以配置为网段（包含 BFD 本地地址和邻居地址），如 192.168.13.0/24，2100::/64，邻居地址通过动态路由协议获取。这样可以方便的配置多个 BFD 多跳邻居。
   - 不支持 OSPF 和静态路由的 BFD 多跳模式（OSPF 邻居本来就必须直连建立）。
2. 同时可以在 multihop-template 中配置 BFD 的相关参数与认证功能，这些 BFD 参数（匹配模板的 BFD 流量）优先于接口及全局级别。
3. 在只有 BFD 多跳邻居的情况下，配置 multihop-template 后，不需要再配置 BFD neighbor。

```bash
config router {bfd | bfd6}
    config multihop-template
        edit <ID>
            set src <IP address/netmask>
            set dst <IP address/netmask>
            set bfd-desired-min-tx <integer>
            set bfd-required-min-rx <integer>
            set bfd-detect-mult <integer>
            set auth-mode {none | md5}
            set md5-key <password>
        next
    end
end
```

网络拓扑



- FW1 与 FW2 建立 IPv4 与 IPv6 的 EBGP 邻居，网段非直连。
- FW1 与 FW2 之间的 IPv4/IPv6 EBGP 邻居启用多跳 BFD。

配置步骤

1. 基础网络配置（略）。
2. 在 FW1 和 FW2 的 port2 上分别开启 BFD，并分别配置 IPv4 和 IPv6 的多跳 BFD 模版（多跳环境下必须配置）。

```bash
config system interface
    edit "port2"
        set bfd enable
    next
end
```

```bash
FW1:
config router bfd
    config multihop-template
        edit 1
            set src 202.103.12.0 255.255.255.0
            set dst 202.103.13.0 255.255.255.0
        next
    end
end
config router bfd6
    config multihop-template
        edit 1
            set src 2100::/64
            set dst 2200::/64
        next
    end
end
```

```bash
FW2:
config router bfd
    config multihop-template
        edit 1
            set src 202.103.13.0 255.255.255.0
            set dst 202.103.12.0 255.255.255.0
        next
    end
end
config router bfd6
    config multihop-template
        edit 1
            set src 2200::/64
            set dst 2100::/64
        next
    end
end
```

3. 在 FW1 和 FW2 上分别配置 BGP，配置 EBGP 邻居（包含 IPv4 和 IPv6），宣告 IPv4 和 IPv6 内网网段。

```bash
FW1:
config router bgp
    set as 10086
    set router-id 202.103.12.2
    config neighbor
        edit "202.103.13.2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set remote-as 10010
        next
        edit "2200::2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set remote-as 10010
        next
    end
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 2300::/64
        next
    end
end
```

```bash
FW2:
config router bgp
    set as 10010
    set router-id 202.103.13.2
    config neighbor
        edit "202.103.12.2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set remote-as 10086
        next
        edit "2100::2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set remote-as 10086
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 2400::/64
        next
    end
end
```

结果验证

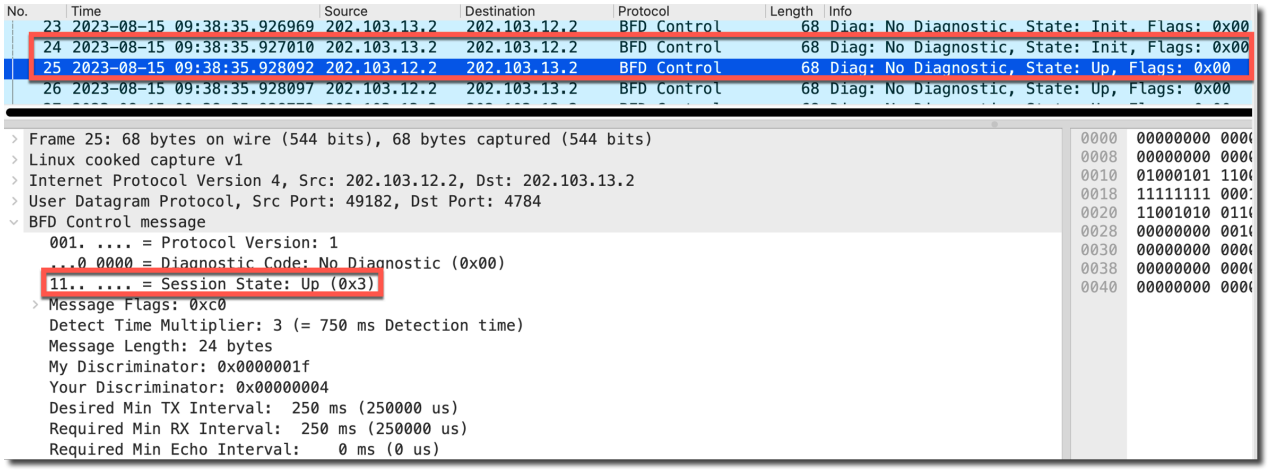
1. 在 FW1 上查看 BFD IPv4 和 IPv6 邻居状态均为 UP（FW2 略）。

```bash
FW1 # get router info bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=202.103.12.2     dst=202.103.13.2     ct=08 ifi=4 type=SM

FW1 # get router info6 bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=2100::2
dst=2200::2
ct=08 ifi=4 type=SM
```

```bash
FW1 # get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
202.103.12.2    202.103.13.2    UP          port2           3/4/M

OurAddress: 2100::2FW1 # get router info6 bfd neighbor
NeighAddress: 2200::2
State: UP Interface: port2 Desc: 4/3 Multi-hop
```

2. 在 FW1 上查看 BFD 相关日志与抓包（FW2 略），IPv4 和 IPv6 的 BFD 状态由 INIT 变为 UP。

```bash
date=2023-08-14 time=17:07:51 eventtime=1692004071014368092 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[202.103.12.2-
>202.103.13.2,49154,port2,3,multi]: state INIT → UP local_diag=0×00"

date=2023-08-14 time=17:07:51 eventtime=1692004070970891127 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[2100::2→2200::2,49155,port2,4,multi]:
state INIT → UP local_diag=0×00"
```



3. 在 FW1 上查看 IPv4 和 IPv6 的 BGP 邻居与 BGP 路由状态，邻居状态正常建立，IPv4 和 IPv6 的 BGP 路由学习正常。

```bash
FW1 # get router info bgp summary
VRF 0 BGP router identifier 202.103.12.2, local AS number 10086
BGP table version is 1
3 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/DOWN  State/PfxRcd
202.103.13.2 4      10010      35      37        1    0    0 00:27:54        1
2200::2      4      10010      36      36        1    0    0 00:26:57        1

Total number of neighbors 2

FW1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [20/0] via 202.103.13.2 (recursive via 202.103.12.1, port2), 00:27:31, [1/0]

FW1 # get router info6 routing-table bgp
Routing table for VRF=0
B       2400::/64 [20/0] via 2200::2 (recursive via 2100::1, port2), 00:27:34, [1024/0]
```

4. 模拟 IPv6 的 BFD 中断，查看 FW1 上 BFD IPv6 邻居 DOWN。

```bash
FW1 # get router info6 bfd neighbor

OurAddress: 2100::2
NeighAddress: 2200::2
State: DOWN Interface: port2 Desc: 10/12 Multi-hop
```

5. 在 FW1 上查看 BFD 相关日志，BFD DOWN 后，对应的 IPv6 BGP 邻居关系跟随 DOWN。

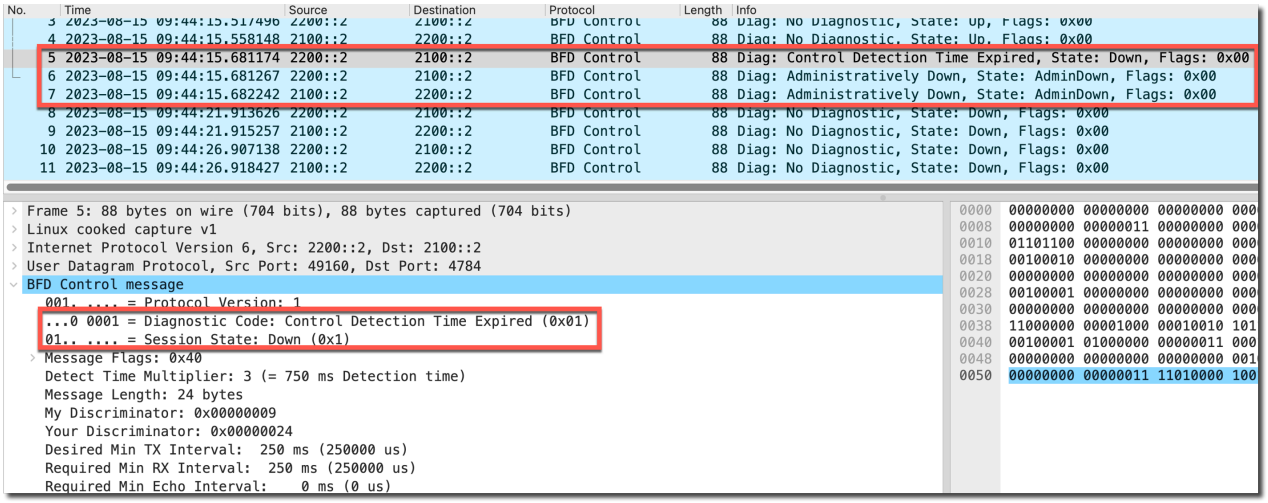
```bash
date=2023-08-14 time=17:33:33 eventtime=1692005613533793725 tz="+0800" logid="0103020300" type="event" subtype="router"
level="warning" vd="root" logdesc="BGP neighbor status changed" msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 2200::2 DOWN
User reset"
date=2023-08-14 time=17:33:33 eventtime=1692005613533476560 tz="+0800" logid="0103020300" type="event" subtype="router"
level="warning" vd="root" logdesc="BGP neighbor status changed" msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 2200::2 DOWN
BGP Notification CEASE"
......
date=2023-08-14 time=17:33:33 eventtime=1692005613533219142 tz="+0800" logid="0103020300" type="event" subtype="router"
level="warning" vd="root" logdesc="BGP neighbor status changed" msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 2200::2 DOWN
BFD DOWN; User reset added"
date=2023-08-14 time=17:33:33 eventtime=1692005613532847101 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[2100::2→2200::2,49155,port2,4,multi]:
state UP → DOWN local_diag=0×01"
```

6. 在 FW1 上查看 IPv4 和 IPv6 的 BGP 邻居状态，IPv6 邻居断开，并开始重建 EBGP 邻居。

```bash
FW1 # get router info bgp summary
VRF 0 BGP router identifier 202.103.12.2, local AS number 10086
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries
```

```
Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/DOWN  State/PfxRcd
202.103.13.2 4      10010     113     114        1    0    0 01:24:53        1
2200::2      4      10010     111     113        0    0    0    never Active
Total number of neighbors 2

FW1 # get router info bgp summary
VRF 0 BGP router identifier 202.103.12.2, local AS number 10086
BGP table version is 2
3 BGP AS-PATH entries
0 BGP community entries
Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/DOWN  State/PfxRcd
202.103.13.2 4      10010     114     115        1    0    0 01:25:49        1
2200::2      4      10010     116     119        2    0    0 00:00:55        1
```

7. 抓包查看 IPv6 BFD DOWN 的报文，FW2 发现检测超时，向 FW1 发送 BFD Time Expired 报文，两端互相发送 AdminDOWN 的 BFD 控制报文，随后双向 BFD 字段中的 Diagnostic Code 为空，状态为 DOWN。



8. 在 FW1 上删除 IPv4 的 BFD 多跳模版，模拟未配置多跳模板情况下的 BFD 多跳状态。

```bash
FW1 # config router bfd
FW1 (bfd) # config multihop-template
FW1 (multihop-template) # delete 1
FW1 (multihop-template) # end
FW1 (bfd) # end
```

9. 查看 FW1 和 FW2 上的 IPv4 BFD 邻居状态，均变为 DOWN。

```bash
FW1 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
202.103.12.2    202.103.13.2    DOWN        port2           16/0

FW2 # get router info bfd neighbor

OurAddress      NeighAddress    State       Interface       LDesc/RDesc
202.103.13.2    202.103.12.2    DOWN        port2           8/17/M
```

0. FW1 和 FW2 的 IPv4 BGP 邻居都没有发生重建。

```bash
FW1 # get router info bgp summary

VRF 0 BGP router identifier 202.103.12.2, local AS number 10086
BGP table version is 2
3 BGP AS-PATH entries
0 BGP community entries
Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.13.2 4      10010     123     124        1    0    0 01:32:59        1
2200::2      4      10010     124     127        2    0    0 00:08:05        1

FW2 # get router info bgp summary

VRF 0 BGP router identifier 202.103.13.2, local AS number 10010
```

```
BGP table version is 6
3 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
202.103.12.2 4      10086     215     216        5    0    0 01:41:32        1
2100::2      4      10086     193     203        6    0    0 00:16:37        1
```

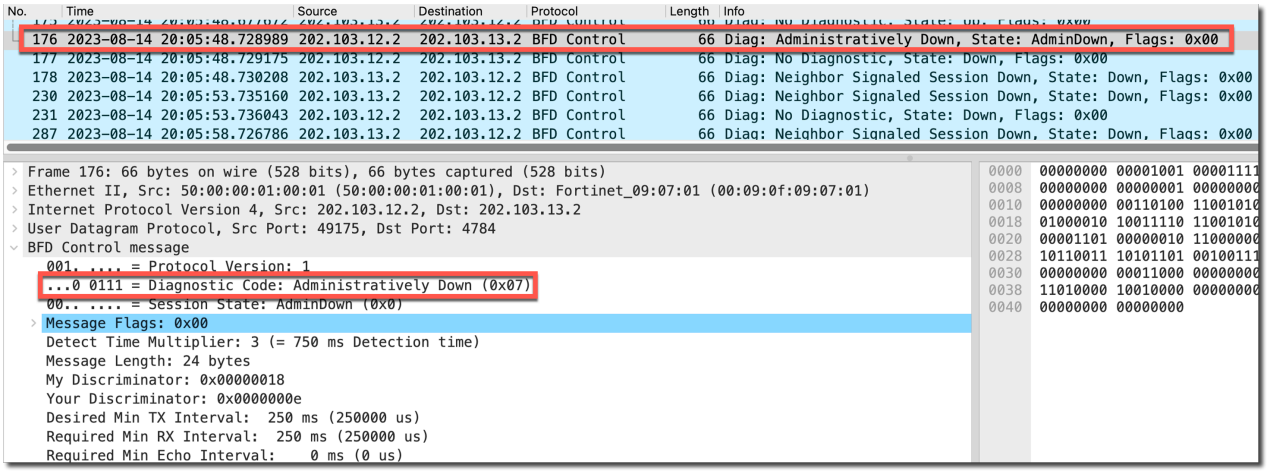
1. 在 FW1 上查看路由相关日志，显示 IPv4 BFD 为 Admin Down，后续没有 IPv4 BGP 断开的日志。

```bash
date=2023-08-14 time=19:40:54 eventtime=1692013254437017862 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[202.103.12.2-
>202.103.13.2,49168,port2,11,multi]: state UP → ADMIN DOWN local_diag=0×27"
```

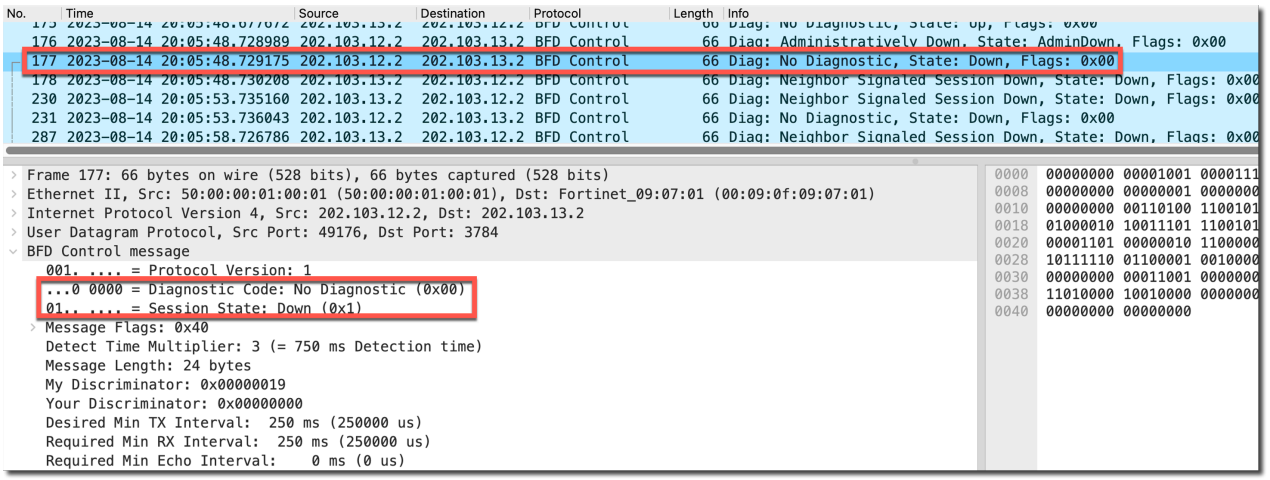
2. 在 FW2 上查看路由相关日志，显示 IPv4 BFD Down，后续没有也 IPv4 BGP 断开的日志。

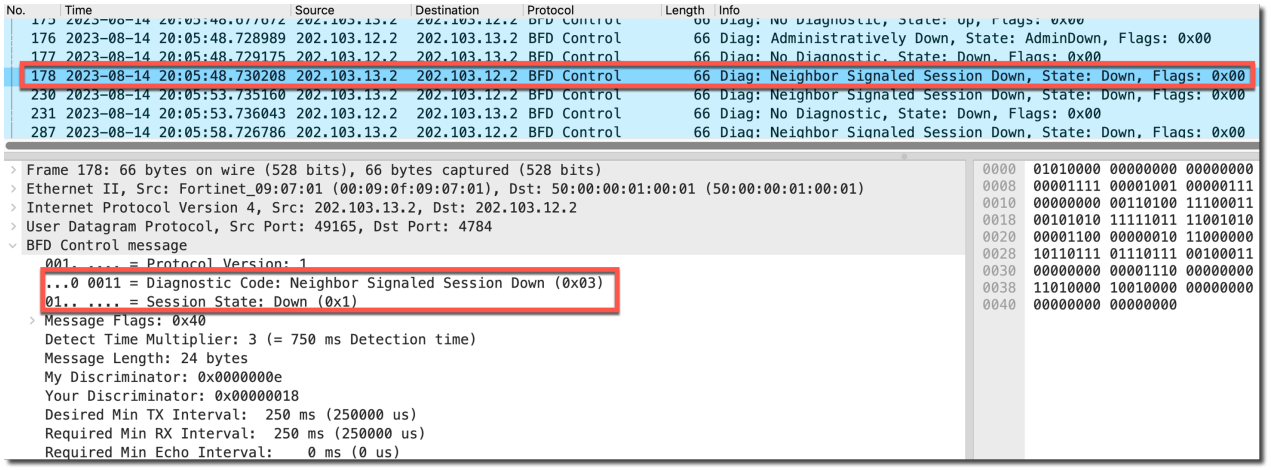
```bash
date=2023-08-14 time=19:40:54 eventtime=1692013254462567015 tz="+0800" logid="0103020304" type="event" subtype="router"
level="warning" vd="root" logdesc="Routing log warning" msg="BFD: BFD session[202.103.13.2-
>202.103.12.2,49159,port2,8,multi]: state UP → DOWN local_diag=0×23"
```

3. 同时抓包查看 IPv4 的 BFD，FW1 向 FW2 发送了 Administratively Down 类型的 BFD。



4. 随后 FW1 → FW2 方向的 BFD Down 中的 Diagnostic Code 变为空，FW2 → FW1 方向的 BFD Down 中的 Diagnostic Code 变为 Neighbor Signaled Session Down。





注意事项

1. Admin 类型的 Down 不会导致 BGP 邻居重建。
2. BFD Down 后触发了 BGP 邻居重建，如果 BFD 一直没有 UP，则重建后的 BGP 邻居不会受任何影响，除非再次发生了 BFD 从 UP 到 Down 的状态转换（静态路由不是这样）。

IPv6

页面开启 IPv6

上网配置

NAT64

- NAT64 映射
- Embedded 模式的 NAT64
- NAT64 + DNS64 上网

NAT46 映射

DHCPv6

- DHCPv6 原理
- 有状态模式
- 无状态 + 有状态模式
- DHCPv6-PD

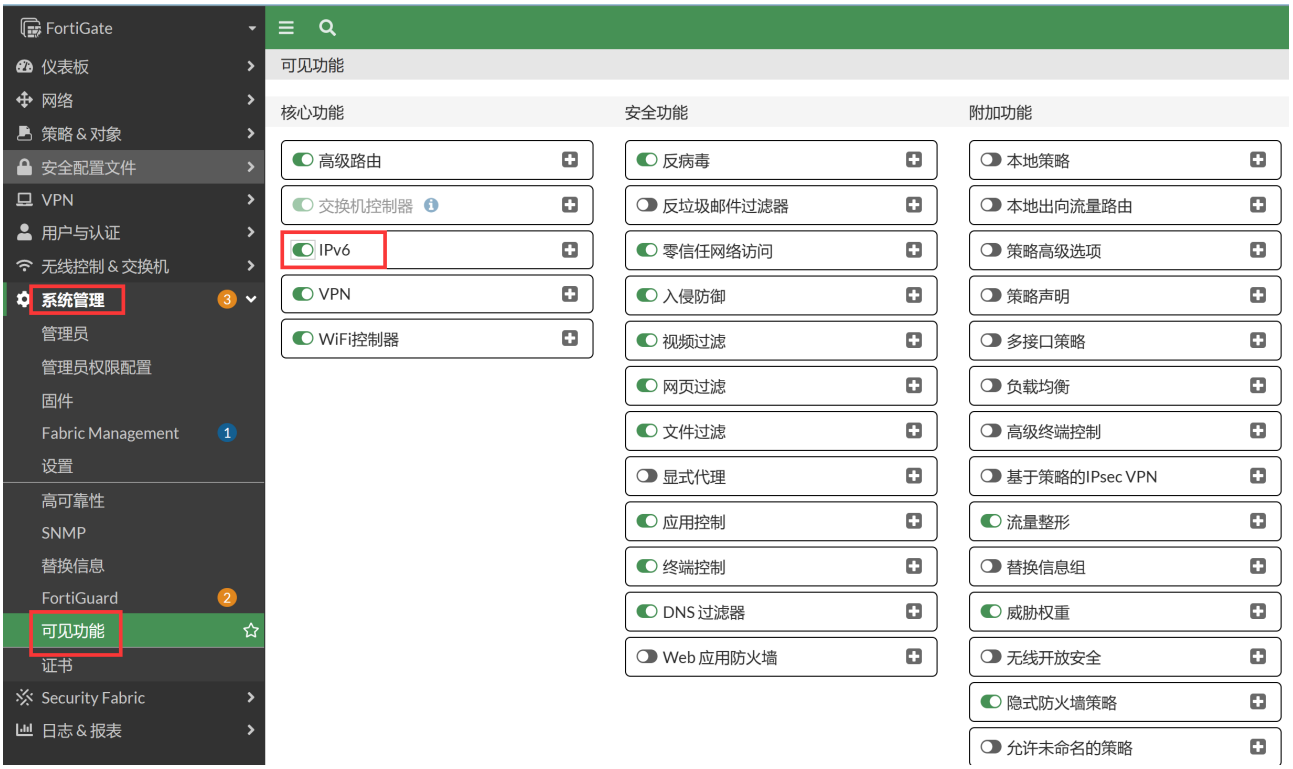
DS-Lite

OSPFv3

常用命令

页面开启 IPv6

需要使用 IPv6 时，需要在系统管理 → 可见功能中开启 IPv6。



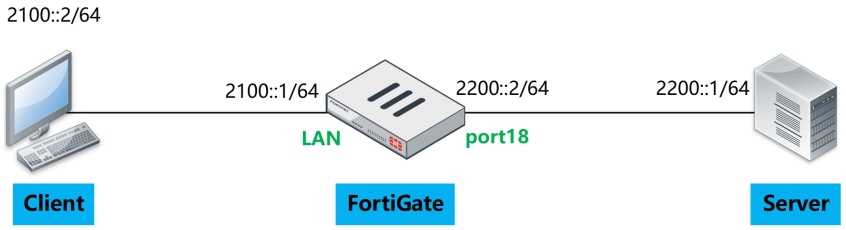

```bash
config system global
    set gui-ipv6 enable
end
```

上网配置

网络需求

1. 用户内部使用 IPv6 网络，FortiGate 作为客户网络的互联网边界介入防火墙，实现对互联网的访问。
2. 外部互联网接入服务商为 IPv6 网络。
3. 内部也为 IPv6 网络。

网络拓扑

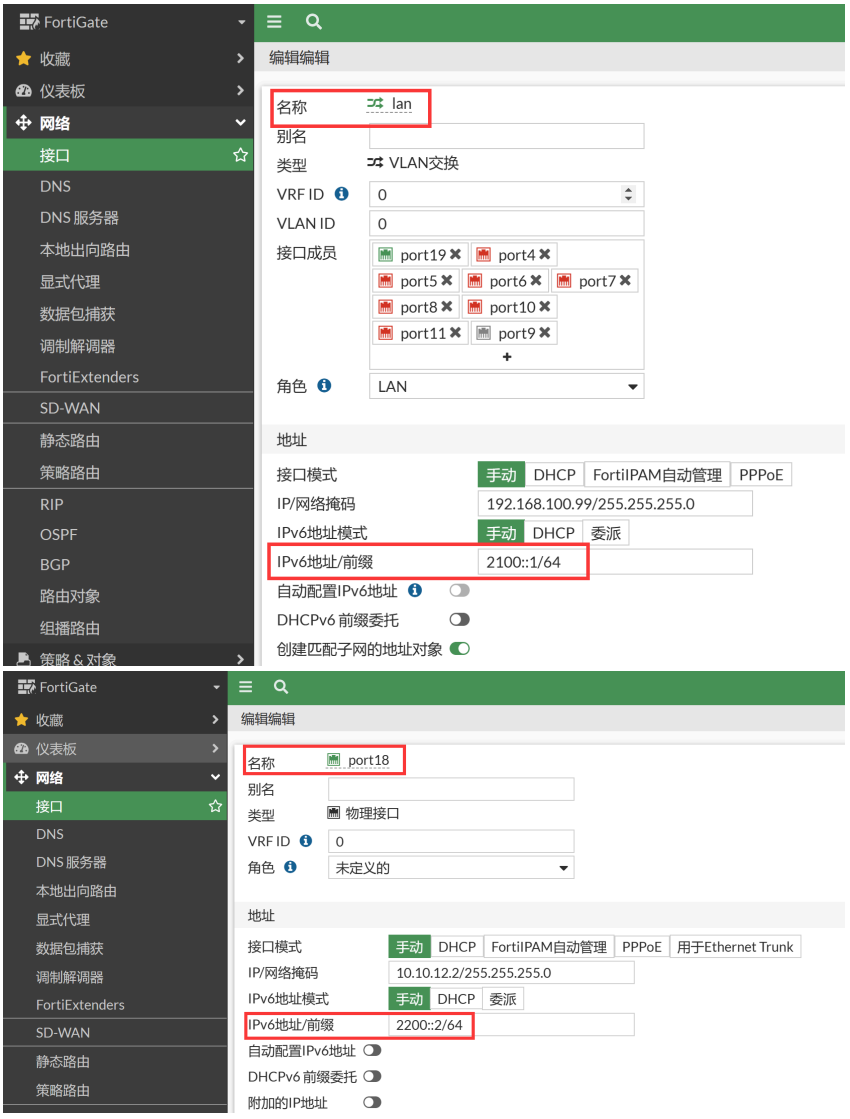


配置要点

- 配置接口 IP
- 配置路由
- 配置策略

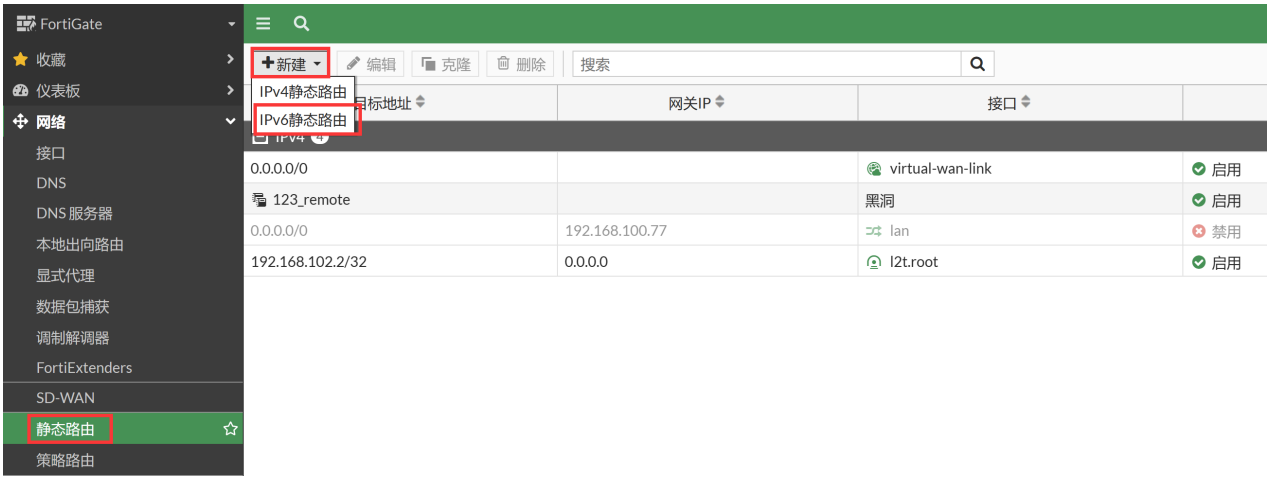
配置步骤

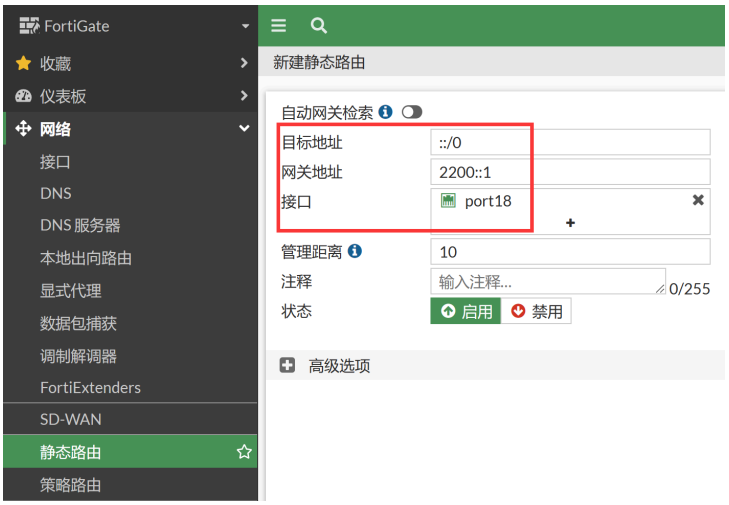
1. 配置接口 IPv6 地址。




```
config system interface
    edit "lan"
        config ipv6
            set ip6-address 2100::1/64
            set ip6-allowaccess ping https ssh snmp http fgfm fabric
        end
    next
    edit "port18"
        config ipv6
            set ip6-address 2200::2/64
            set ip6-allowaccess ping https ssh snmp http fgfm fabric
        end
    next
end
```

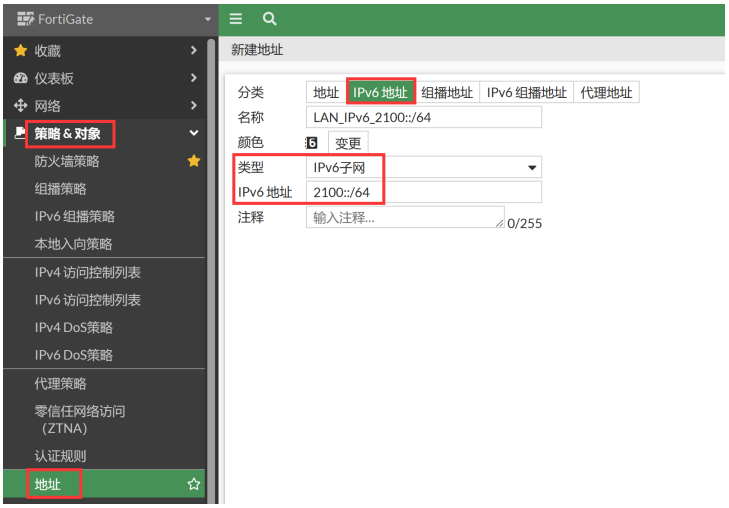
2. 配置 IPv6 路由。






```
config router static6
    edit 1
        set gateway 2200::1
        set device "port18"
    next
end
```

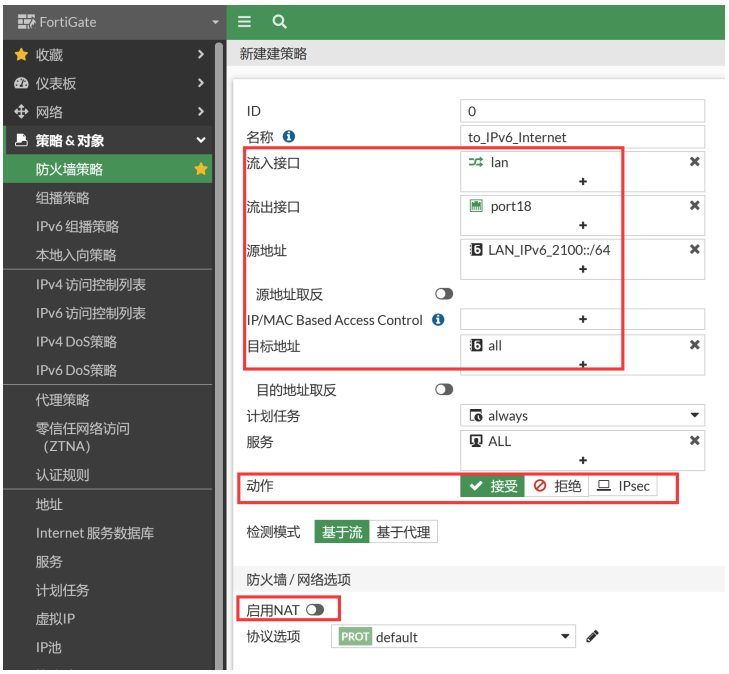
3. 创建内网网段的 IPv6 地址对象，进入策略 & 对象 → 地址，新建内网的 IPv6 地址对象。




```
config firewall address6
    edit "LAN_IPv6_2100::/64"
        set uuid 042a2792-8d96-51ed-39e3-a512a3f5e16d
        set ip6 2100::/64
    next
```

```
end
```

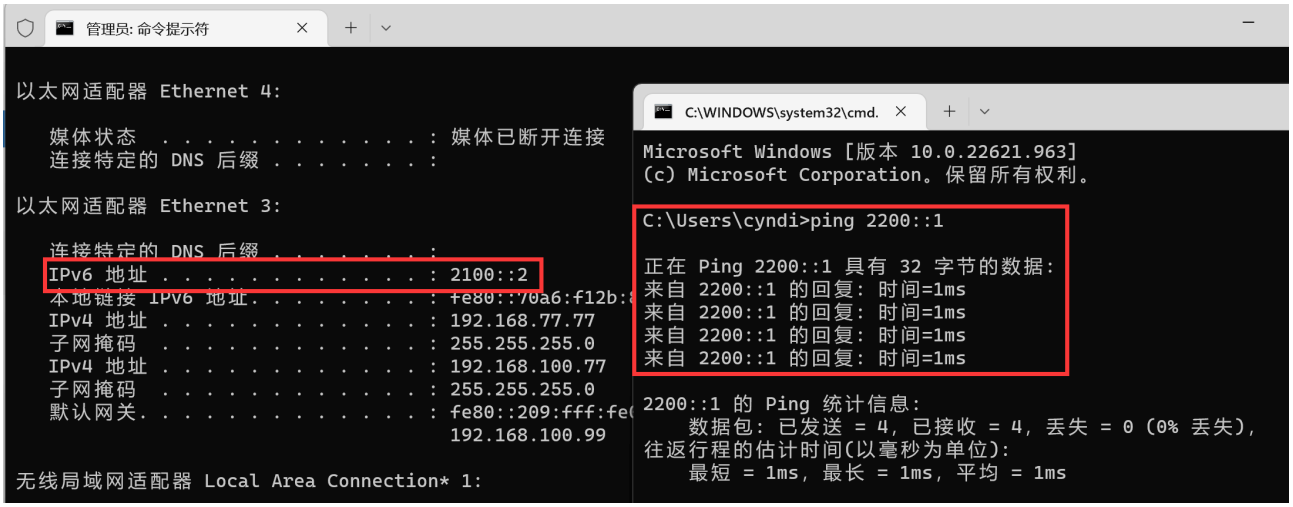
4. 新建防火墙策略，允许内部 IPv6 访问外部 IPv6 网络，由于都是 IPv6 地址，可以直接通信，不需要开启 NAT。




```
config firewall policy
    edit 10
        set name "to_IPv6_Internet"
        set srcintf "lan"
        set dstintf "port18"
        set action accept
        set srcaddr6 "LAN_IPv6_2100::/64"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

结果验证

客户端 PC 可以通过 FortiGate 正常访问 IPv6 Internet。



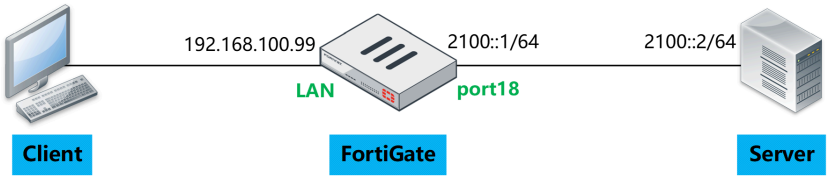
NAT46 映射

网络需求

将 IPv6 内部服务器，发布到 IPv4 网络之中，允许用户通过 IPv4 来对其进行访问。

网络拓扑

192.168.100.77



Client 侧为 IPv4，Server 侧为 IPv6，使用 Client 访问 Server，FortiGate 做 NAT46 转换。

配置步骤

1. 创建 VIP，选择类型为 IPv4，“映射到”填写 IPv6 Server 真实地址，“外部 IP 地址/范围”填写 IPv4 VIP 地址，如果是一个范围，必须填写为一个完整的 subnet（最小 31 位）的单播地址范围，不能填写广播和网络号，否则 GUI 报错。

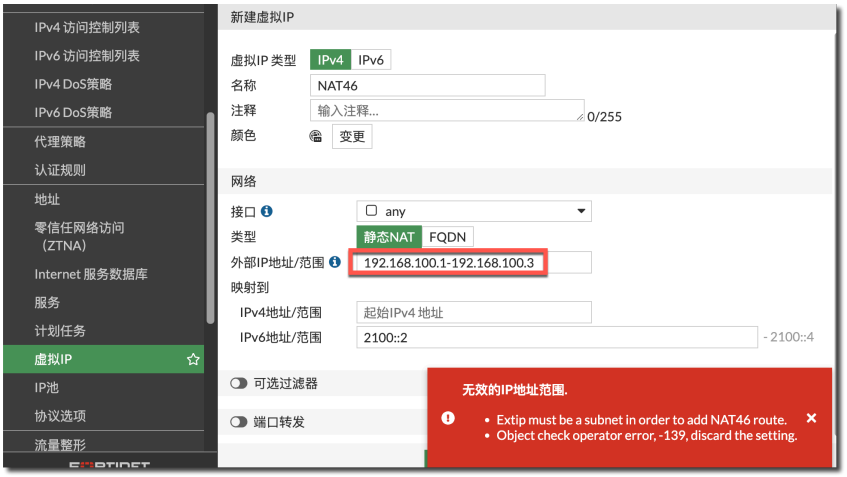



```
config firewall vip
    edit "NAT46"
        set extip 192.168.200.99
        set nat44 disable
        set nat46 enable
        set extintf "lan"
        set ipv6-mappedip 2100::2
        set add-nat46-route enable
    next
end
```

2. 可以看默认存在一个 naf.root 接口，用于处理 NAT46 流量，创建 VIP 完成后，如果开启了 add-nat46-route（默认开启），针对 naf.root 接口生成去往 VIP IPv4 地址的路由，关闭 add-nat46-route 后，该路由会消失。

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
FortiGate-101F # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->192.168.200.99/32 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)
```

3. 如果是 NAT46，“外部 IP 范围”配置的不是一个完整的 subnet，或范围中包含 subnet 的广播地址，则下发时页面或 CLI 会报错，因为无法为 naf 接口下发路由。



4. 创建 IPv6 Pool，配置地址池范围，注意不要和 port18 地址冲突，配置冲突这里并不会提示，后续会讲到为什么不能配成一样的，开启 NAT46 后，须填写为一个完整的 subnet（最小 128 位）的 IPv6 地址范围，否则 GUI 报错。




```
config firewall ippool6
    edit "NAT46"
        set startip 2200::
        set endip 2200::ffff
        set nat46 enable
        set add-nat46-route enable
    next
end
```

5. 创建完成后，如果开启了 add-nat46-route（默认开启），可以看到生成了 IPv6 的 kernel 路由，针对 naf.root 接口生成去往 IPPool IPv6 地址的路由，IPPool6 关闭 NAT46 或关闭 add-nat46-route 后，该路由会消失。

```
FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2200::/112 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

6. 开启 NAT46 后，如果 IPPool6 配置的不是一个完整的 subnet，则下发时页面或 CLI 会报错，因为无法为 naf 接口下发路由。



7. 创建防火墙策略，先在“启用 NAT”选项中开启 NAT46，然后源地址引用 IPv4 的 all，然后在目标地址中引用 NAT46 VIP，在 IPPool 中引用 IPPool6。




```
config firewall policy
    edit 2
        set name "NAT46"
        set srcintf "lan"
        set dstintf "port18"
        set action accept
        set nat46 enable
        set srcaddr "all"
        set dstaddr "NAT46"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
```

```
        set service "ALL"
        set ippool enable
        set poolname6 "NAT46"
    next
end
```

结果验证与注意事项

1. Client 访问 FortiGate 的 NAT46 VIP 地址 192.168.200.99，抓包查看 NAT46 过程，可以看到经过 naf.root 接口，目标地址被转换为 Server IPv6，源 IPv4 地址被转换为 IPPool6 中的 IPv6 地址，返回的流量从 IPv6 被还原为 IPv4。

```
FortiGate-101F # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
3.380842 lan in 192.168.100.77 -> 192.168.200.99: icmp: echo request
3.380968 naf.root out 192.168.100.77 -> 192.168.200.99: icmp: echo request
3.380978 naf.root in 2200::24f6 -> 2100::2: icmp6: echo request seq 2035
3.381071 port18 out 2200::24f6 -> 2100::2: icmp6: echo request seq 2035
3.381213 port18 in 2100::2 -> 2200::24f6: icmp6: echo reply seq 2035 [flowlabel 0x69fcf]
3.381256 naf.root out 2100::2 -> 2200::24f6: icmp6: echo reply seq 2035 [flowlabel 0x69fcf]
3.381264 naf.root in 192.168.200.99 -> 192.168.100.77: icmp: echo reply
3.381304 lan out 192.168.200.99 -> 192.168.100.77: icmp: echo reply
```

2. debug flow IPv4/IPv6，可以看到整个 NAT46 过程都需要经过 naf.root 接口处理。

```
FortiGate-101F # id=20085 trace_id=649 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 192.168.100.77:1->192.168.200.99:2048) tun_id=0.0.0.0 from
lan. type=8, code=0, id=1, seq=2035."
id=20085 trace_id=649 func=init_ip_session_common line=6003 msg="allocate a new session-0013ba16, tun_id=0.0.0.0"
id=20085 trace_id=649 func=get_vip46_addr line=1091 msg="find DNAT46: IP-2100::2, port-128(fixed port)"
id=20085 trace_id=649 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-192.168.200.99 via naf.root"
id=20085 trace_id=649 func=get_new_addr46 line=1154 msg="find SNAT46: IP-2200::24f6(from IPPOOL), port-60417"
id=20085 trace_id=649 func=fw_forward_handler line=874 msg="Allowed by Policy-2: SNAT"
id=20085 trace_id=649 func=ip4_nat_af_input line=261 msg="nat64 ipv4 received a packet proto=1"
id=20085 trace_id=649 func=resolve_ip6_tuple line=4926 msg="allocate a new session-00001c8b"
id=20085 trace_id=649 func=vf_ip6_route_input line=1201 msg="find a route: gw-2100::2 via port18 err 0 flags 01000001"
id=20085 trace_id=649 func=fw6_forward_handler line=435 msg="Check policy between naf.root -> port18"
id=20085 trace_id=649 func=fw6_forward_handler line=567 msg="Allowed by Policy-2:"
id=20085 trace_id=586 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:60417->2200::24f6:129) from port18."
id=20085 trace_id=586 func=resolve_ip6_tuple_fast line=4827 msg="Find an existing session, id-00001c8b, reply direction"
id=20085 trace_id=586 func=vf_ip6_route_input line=1201 msg="find a route: gw-2200::24f6 via naf.root err 0 flags 01000001"
id=20085 trace_id=586 func=fw6_forward_dirty_handler line=371 msg="state=00000204, state2=00000000, npu_state=04040000"
id=20085 trace_id=586 func=ip6_nat_af_input line=298 msg="nat64 ipv6 received a packet proto=58"
id=20085 trace_id=586 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-192.168.100.77 via lan"
id=20085 trace_id=586 func=npu_handle_session64 line=1246 msg="Trying to offloading session from naf.root to lan, skb.npu_flag=00000080 ses.state=00010204 ses.npu_state=0x0
4040000"
id=20085 trace_id=586 func=fw_forward_dirty_handler line=410 msg="state=00010204, state2=00000001, npu_state=04040000"
```

3. 在 FortiGate 查看 IPv4 会话，可以看到源接口为物理入接口 lan，目标接口为 index 37，为 naf.root，naf 被标记为 1，IPv4 流量未被加速，会话中还包含相关的 IPv6 的相关信息。

```
session info: proto=1 proto_state=00 duration=8 expire=51 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=60/1/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 6/0 rx speed(Bps/kbps): 6/0
                                                    dev_index37即为naf.root接口
orgin->sink: org pre->post, reply pre->post dev=41->37/37->41 gwy=192.168.200.99/192.168.100.77
hook=pre dir=org act=noop 192.168.100.77:1->192.168.200.99:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.200.99:1->192.168.100.77:0(0.0.0.0:0)
peer=2200::24f6:60417->2100::2:128 naf=1
hook=pre dir=org act=noop 2200::24f6:60417->2100::2:128(:::0)
hook=post dir=reply act=noop 2100::2:60417->2200::24f6:129(:::0)
misc=0 policy_id=2 pol_uuid_idx=555 auth_info=0 chk_client_info=0 vd=0
serial=0013ba16 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
ofld_fail_reason(kernel, drv): none/not-established, none(0)/none(0)
npu_state_err=00/04
```

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

4. 在 FortiGate 查看 IPv6 会话，可以看到源接口为 index 37，为 naf.root，目标接口为物理出接口 port18，naf 被标记为 2，IPv6 流量未被加速，会话中还包含相关的 IPv4 会话信息。

```
session6 info: proto=58 proto_state=00 duration=15 expire=44 timeout=0 flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty npu
statistic(bytes/packets/allow_err): org=80/1/0 reply=80/1/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=37->30/30->37
hook=pre dir=org act=noop 2200::24f6:60417->2100::2:128(:::0)
hook=post dir=reply act=noop 2100::2:60417->2200::24f6:129(:::0)
peer=192.168.200.99:1->192.168.100.77:0 naf=2
hook=pre dir=org act=noop 192.168.100.77:1->192.168.200.99:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.200.99:1->192.168.100.77:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00001c8b tos=ff/ff ips_view=0 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
ofld_fail_reason(kernel, drv): none/not-established, none(0)/none(0)
npu_state_err=00/04
```

```
FortiGate-101F # diagnose netlink interface list | grep port18
if=port18 family=00 type=1 index=30 mtu=1500 link=0 master=0
```

5. 修改 IPPool6 的地址范围和 IPv6 出接口地址一致，再次测试 NAT46 转换。

```
config firewall ippool6
    edit "NAT46"
        set startip 2100::1
        set endip 2100::1
        set nat46 enable
    next
end
```

6. 查看 kernel 路由中关于 naf.root 的路由，可以看到未产生 IPPool6 的路由，这样返回的流量肯定是被丢掉的，接下来验证一下。

```
FortiGate-101F # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->192.168.200.99/32 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)

FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

7. debug flow IPv4 & IPv6 可以看到，Client 到 Server 方向的 NAT46 的 DNAT 和 SNAT 均正常工作，但返回的 IPv6 流量未匹配到 naf.root 接口的路由，而 gw 查找到的是 root 接口，返回的流量无法到达 Client。

```
FortiGate-101F # id=20085 trace_id=858 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 192.168.100.77:1->192.168.200.99:2048) tun_id=0.0.0.0 from
lan. type=8, code=0, id=1, seq=2051."
id=20085 trace_id=858 func=init_ip_session_common line=6003 msg="allocate a new session-0013c852, tun_id=0.0.0.0"
id=20085 trace_id=858 func=get_vip46_addr line=1091 msg="find DNAT46: IP-2100::2, port-128(fixed port)"
id=20085 trace_id=858 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-192.168.200.99 via naf.root"
id=20085 trace_id=858 func=get_new_addr46 line=1154 msg="find SNAT46: IP-2100::1(from IPPOOL), port-60417"
id=20085 trace_id=858 func=fw_forward_handler line=874 msg="Allowed by Policy-2: SNAT"
id=20085 trace_id=858 func=ip4_nat_af_input line=261 msg="nat64 ipv4 received a packet proto=1"
id=20085 trace_id=858 func=resolve_ip6_tuple line=4926 msg="allocate a new session-00001c92"
id=20085 trace_id=858 func=vf_ip6_route_input line=1201 msg="find a route: gw-2100::2 via port18 err 0 flags 01000001"
id=20085 trace_id=858 func=fw6_forward_handler line=435 msg="Check policy between naf.root -> port18"
id=20085 trace_id=858 func=fw6_forward_handler line=567 msg="Allowed by Policy-2:"
id=20085 trace_id=624 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:60417->2100::1:129) from port18."
id=20085 trace_id=624 func=resolve_ip6_tuple_fast line=4827 msg="Find an existing session, id-00001c92, reply direction"
id=20085 trace_id=624 func=vf_ip6_route_input line=1201 msg="find a route: gw-:: via root err 0 flags 80200001"
```

8.**所以配置 NAT46 时，需要注意，IPPool6 不要配置成和出接口地址相同的地址，配置不会报错，但实际会不通，如果是 IPv6 的第二地址，实际测试也是不行的。**

```
config system interface
    edit "port18"
        config ipv6
            set ip6-address 2100::1/64
            config ip6-extra-addr
                edit 2100::3/64
            end
        end
end
```

9. 还原 IPPool6 的配置，这次修改 VIP 的 External IP 和物理入接口 IP 一致。

```
config firewall vip
    edit "NAT46"
        set extip 192.168.100.99
        set nat44 disable
        set nat46 enable
        set extintf "lan"
        set ipv6-mappedip 2100::2
    next
end
```

10. 查看 kernel 路由中关于 naf.root 的路由，可以看到未产生 naf.root 接口的 VIP 的路由，这样 NAT46 的 Client 发起方流量应该是无法进入 NAT46 流程的，接下来验证一下。

```
FortiGate-101F # get router info kernel | grep naf

FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2100::100/128 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

11. debug flow IPv4 & IPv6 可以看到，Client 到 Server 方向的 NAT46 流量被 FortiGate 路由到 root 接口，未进入 naf.root 接口，由 FortiGate 直接返回了 reply 报文，没有 IPv6 的 debug flow 产生。

```
FortiGate-101F # id=20085 trace_id=442 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 192.168.100.77:1->192.168.100.99:2048) tun_id=0.0.0.0 from lan. type=8, code=0, id=1, seq=1940."
id=20085 trace_id=442 func=init_ip_session_common line=6003 msg="allocate a new session-0011c351, tun_id=0.0.0.0"
id=20085 trace_id=442 func=get_vip46_addr line=1091 msg="find DNAT46: IP-2100::2, port-128(fixed port)"
id=20085 trace_id=442 func=vf_ip_route_input_common line=2604 msg="find a route: flag=84000000 gw-192.168.100.99 via root"
id=20085 trace_id=443 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 192.168.100.99:1->192.168.100.77:0) tun_id=0.0.0.0 from local. type=0, code=0, id=1, seq=1940."
id=20085 trace_id=443 func=resolve_ip_tuple_fast line=5910 msg="Find an existing session, id-0011c351, reply direction"
```

12. **所以配置 NAT46 时，需要注意，IPv4 VIP 的 extip 不要配置成和入接口地址相同的地址，配置不会报错，但实际会不通，如果是 IPv4 的第二地址，实际测试也是不行的。**

CLI 相关

VIP CLI

1. IPv4 类型的 VIP 支持三种组合，NAT44、NAT46、NAT44 + NAT46。

```
config firewall vip
    edit "vip1"
        set extip 30.0.0.110-30.0.0.120
        set extintf "port3"
        set arp-reply disable
        set portforward enable
        set srcintf-filter "port3"
        set nat44 disable    <-------- new option enabled by default
        set nat46 enable    <-------- new option disabled by default
        set ipv6-mappedip 2040::20-2040::30    <-------- new option available when nat46 enabled
        set extport 10
        set ipv6-mappedport 100    <-------- new option available when nat46 enabled
    next
end
```

2. NAT46 也支持 Virtual Server，开启 NAT46 后，RS 必须配置为 IPv6 地址，实现的 NAT 过程和普通的 NAT46 是一样的。

```
config firewall vip
    edit "vip_svr"
        set type server-load-balance
        set extip 30.0.0.130
        set extintf "port3"
        set arp-reply disable
        set server-type http
        set monitor "ping-mon-l"
        set nat46 enable
        set extport 80
            config realservers
                edit 1
                    set ip 2040::20    <-------- server ip must be ipv6 when nat46 is enabled
```

```
                    set port 80
                next
            end
        next
end
```

3. IPv6 类型的 VIP 支持三种组合，NAT66、NAT64、NAT66 + NAT64。

```
config firewall vip6
    edit "vip3"
        set extip 2002::100-2002::200
        set portforward enable
        set nat66 disable    <——— new option enabled by default
        set nat64 enable    <——— new option disabled by default
        set ipv4-mappedip 8.8.8.8-8.8.8.108    <——— new option available when nat64 is enabled
        set ipv4-mappedport 1000    <—— new option available when nat64 is enabled
        set extport 100
    next
end
```

IPPool CLI

1. IPv4 类型的 IPPool，类型为 NAT64，地址池范围必须是一个 subnet 的范围，默认会开启 add-nat64-route。

```
config firewall ippool
    edit "pol64"
        set startip 10.0.0.0    <—— ip range must be a subnet if nat64 enabled
        set endip 10.0.255.255
        set nat64 enable    <—— new option
        set add-nat64-route enable    <——— new option available when nat64 enabled (by default enable)
    next
end
```

2. IPv6 类型的 IPPool，类型为 NAT46，地址池范围必须是一个 subnet 的范围，默认会开启 add-nat46-route。

```
config firewall ippool6
    edit "pol46"
        set startip 2002::    <——— ip range must be a subnet if nat46 enabled
        set endip 2002::ffff
        set nat46 enable
        set add-nat46-route enable    <——— new option available when nat46 enabled (by default enable)
    next
end
```

Firewall Policy & Central SNAT CLI

1. NAT46 Firewall Policy。

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "vip1"    <— vip with nat46 enabled
        set srcaddr6 "all"
        set dstaddr6 "all"    <—— both ipv4 and ipv6 addresses are required when nat46 enable
        set action accept
        set schedule "always"
        set service "ALL"
        set dnsfilter-profile "default"
        set ippool enable
        set poolname6 "ipv6_po1"    <——— ipv6 pool with nat46 enabled
        set nat46 enable    <——— new option to enable nat46
    next
end
```

2. NAT46 Central SNAT。

```
config firewall central-snat-map
    edit 2
        set srcintf "any"
        set dstintf "any"
        set orig-addr "all"
        set dst-addr "all"
        set nat46 enable    ←——— new option for ipv4 rule
        set nat-ippool6 "ipv6_po1"    ←——— available when nat46 is enabled, ipv6 pool with nat46 enabled
    next
end
```

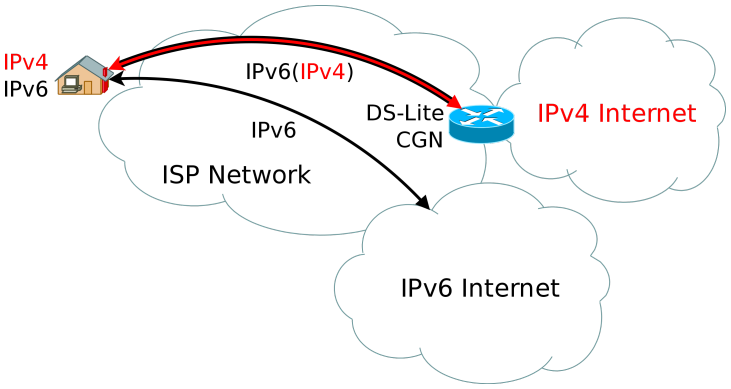
DS-Lite

> **重要**
>
> 该功能在 7.2.0 及以后版本支持。

DS-Lite 简介

Dual-StackLite，简称 DS-Lite，是一种使用 IPv4overIPv6 隧道将 IPv4 数据包发送到运营商来实现 IPv4 私网地址用户穿越 IPv6 网络访问 IPv4 公网的解决方案。

支持此技术的客户端设备（CPE）会将 IPv4 数据包封装到 IPv6 数据包中，并且将数据包发送至运营商的电信级 NAT（CGNAT）。CGNAT 收到数据包后，将其还原为 IPv4 数据包，在进行 NAT 处理后发送到 IPv4 互联网。CGNAT 通过记录 IPv6 源地址、私有 IPv4 地址，以及 TCP 或 UDP 端口号来标识流量。



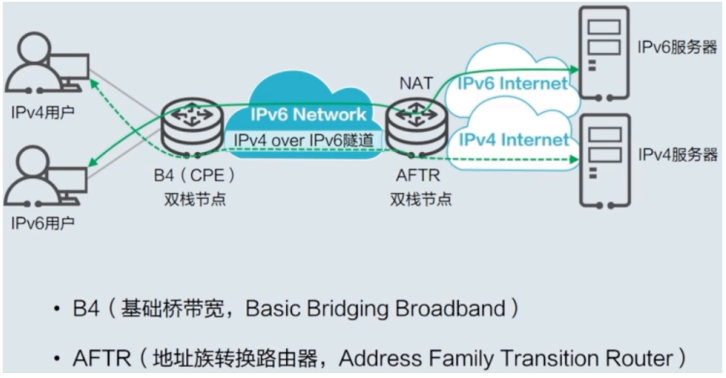
DS-Lite 中的几个角色：

B4（Basic Bridging Broadband）和 AFTR（Address Family Transition Router）都具有双栈能力：

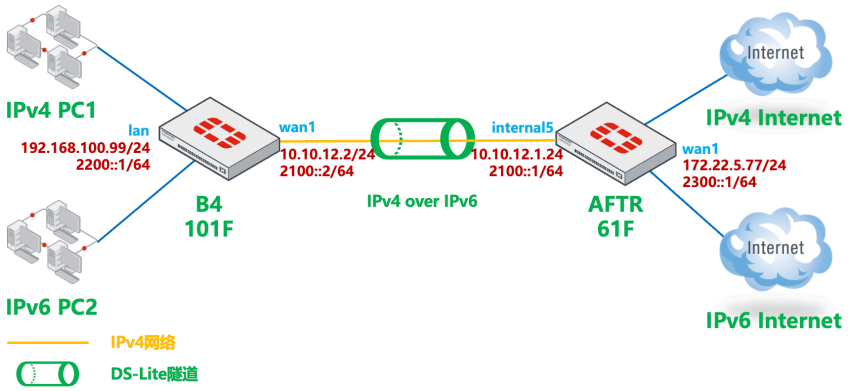
- B4 是 IPv4 over IPv6 的创建者，一般位于用户侧（CPE）。
- AFTR 是 IPv4 over IPv6 的终结者，负责 IPv4 over IPv6 报文的解封装与 NAT，一般部署于 ISP 设备上，也被叫作 BR（Border Relay）。

IPv6 和 IPv4 用户都可以正常访问对应 IP 栈的 Internet：

- IPv6 用户通过 B4 和 AFTR 之间的 IPv6 线路直接访问 IPv6 Internet。
- IPv4 用户访问 IPv4 的 Internet，通过 B4 的 IPv4 over IPv6 隧道封装 IPv6 外层 IP 头，到达 AFTR 后，解封装为原始的 IPv4 流量，通过 NAT 转换后访问 IPv4 Internet。



网络拓扑



1. B4 与 AFTR（BR）设备均为 FortiGate，之间配置 IPv4 over IPv6 隧道（DS-Lite 类型的 VNE 隧道，Virtual Network Enabler）。
2. IPv4 PC1 访问 IPv4 Internet 的流量被 B4 的 VNE 隧道封装为 IPv4 over IPv6 流量送至 AFTR，AFTR 解封装后得到 IPv4 流量，做 SNAT 后送至 IPv4 Internet。
3. IPv6 PC2 访问 IPv6 Internet 的流量直接由 B4 和 AFTR 的 IPv6 协议栈转发至 IPv6 Internet。

配置步骤

1. 基础 IPv4/IPv6 双栈网络配置（根据网络拓扑配置，略）。

2. 在 B4 设备上配置 VNE（Virtual Network Enabler）隧道，隧道类型为 DS-Lite，绑定接口为与 AFTR 的接口 wan1，br 地址为 AFTR 的 IPv6 地址，开启 DS-Lite 隧道的加速。

```bash
config system vne-tunnel
    set status enable
    set interface "wan1"
    set auto-asic-offload enable
    set br "2100::1"    //也可配置FQDN//
    set mode ds-lite
end
```

3. 在 AFTR 设备上配置反向的 VNE（Virtual Network Enabler）隧道，隧道类型为 DS-Lite，绑定接口为与 AFTR 的接口 wan1，br 地址为 B4 的 IPv6 地址，开启 DS-Lite 隧道的 NP 加速。

```bash
config system vne-tunnel
    set status enable
    set interface "internal5"
    set auto-asic-offload enable
    set br "2100::2"    //也可配置FQDN//
    set mode ds-lite
end
```

4. 在 B4 上配置 IPv4 和 IPv6 默认路由，IPv4 默认路由指向 VNE 隧道，IPv6 默认路由指向 AFTR。

```bash
config router static
    edit 5
        set device "vne.root"
    next
end
config router static6
    edit 1
        set gateway 2100::1
        set device "wan1"
    next
end
```

5. 在 AFTR 设备上配置反向的 IPv4 路由和 IPv6 路由，IPv4 路由指向 VNE 隧道，IPv6 路由指向 B4。

```bash
config router static
    edit 12
        set dst 192.168.100.0 255.255.255.0
        set device "vne.root"
    next
end
config router static6
    edit 1
        set dst 2200::/64
        set gateway 2100::2
        set device "internal5"
    next
end
```

6. 在 B4 上配置安全策略，放通 lan 到 VNE 隧道的 IPv4 流量，以及 lan 到 wan1 的 IPv6 流量。

```bash
config firewall policy
    edit 6
        set name "dslite"
        set srcintf "lan"
        set dstintf "vne.root"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    edit 7
        set name "IPv6"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 在 AFTR 设备上配置安全策略，放通 VNE 隧道到 wan1 的 IPv4 流量，开启 SNAT，放通 Internal5 到 wan1 的 IPv6 流量。

```bash
config firewall policy
    edit 11
        set name "dslite"
        set srcintf "vne.root"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 12
        set srcintf "internal5"
        set dstintf "wan1"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
end
```

结果验证

1. 在 B4 上查看 VNE 隧道状态。

```
B4 # diagnose test application vned 1
```

bash

```
vdom: root/0, is master, devname=wan1 link=1 tun=vne.root mode=fixed-ip ssl_cert=Fortinet_Factory
end user ipv6 perfix: ::/0
interface ipv6 addr: 2100::2
config ipv4 perfix: 0.0.0.0/0.0.0.0
config br: 2100::1
tunnel br: 2100::1
tunnel ipv6 addr: 2100::2
tunnel ipv4 addr: 0.0.0.0/0.0.0.0
DS-Lite rule client: state=succeed retries=0 interval=0 expiry=0 reply_code=0
fqdn=2100::1 num=1 cur=0 ttl=4294967295 expiry=0
2100::1
```

2. 在 AFTR 设备上查看 VNE 隧道状态。

```
AFTR # diagnose test application vned 1
```

bash

```
vdom: root/0, is master, devname=internal5 link=1 tun=vne.root mode=fixed-ip ssl_cert=Fortinet_Factory
end user ipv6 perfix: ::/0
interface ipv6 addr: 2100::1
config ipv4 perfix: 0.0.0.0/0.0.0.0
config br: 2100::2
tunnel br: 2100::2
tunnel ipv6 addr: 2100::1
tunnel ipv4 addr: 0.0.0.0/0.0.0.0
DS-Lite rule client: state=succeed retries=0 interval=0 expiry=0 reply_code=0
fqdn=2100::2 num=1 cur=0 ttl=4294967295 expiry=0
2100::2
```

3. 使用 IPv4 PC1 访问 IPv4 Internet，在 B4 和 AFTR 上同时抓包，原始 IPv4 从 B4 的 lan 口收到后，根据 IPv4 默认路由被送入 VNE 接口，VNE 接口封装新的 IPv6 头后，从 wan1 发送给 AFTR，AFTR 收到后，解封装为原始 IPv4，查 IPv4 默认路由从 wan1 发出并做 SNAT，返回流量也是相同的流程。

```
B4 # diagnose sniffer packet any '(host 223.5.5.5 and icmp) or host 2100::1)' 4
```

bash

```
interfaces=[any]
filters=[(host 223.5.5.5 and icmp) or host 2100::1]
9.415112 lan in 192.168.100.77 → 223.5.5.5: icmp: echo request
9.415202 vne.root out 192.168.100.77 → 223.5.5.5: icmp: echo request
9.415216 wan1 out 2100::2 → 2100::1: 192.168.100.77 → 223.5.5.5: icmp: echo request
9.419797 vne.root in 223.5.5.5 → 192.168.100.77: icmp: echo reply
9.419828 lan out 223.5.5.5 → 192.168.100.77: icmp: echo reply

AFTR # diagnose sniffer packet any '(host 223.5.5.5 and icmp) or host 2100::2' 4
interfaces=[any]
filters=[(host 223.5.5.5 and icmp) or host 2100::2]
3.640262 vne.root in 192.168.100.77 → 223.5.5.5: icmp: echo request
3.640354 wan1 out 172.22.5.77 → 223.5.5.5: icmp: echo request
3.644756 wan1 in 223.5.5.5 → 172.22.5.77: icmp: echo reply
3.644785 vne.root out 223.5.5.5 → 192.168.100.77: icmp: echo reply
3.644801 internal5 out 2100::1 → 2100::2: 223.5.5.5 → 192.168.100.77: icmp: echo reply
```

4. 在 B4 上查看 IPv4 流量（VNE 隧道内层）的会话，流量被 NP 加速，无法看到 IPv6 的流量会话。

bash

```
session info: proto=1 proto_state=00 duration=32 expire=28 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=120/2/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=42->52/52->42 gwy=223.5.5.5/192.168.100.77
hook=pre dir=org act=noop 192.168.100.77:1->223.5.5.5:8(0.0.0.0:0)
hook=post dir=reply act=noop 223.5.5.5:1->192.168.100.77:0(0.0.0.0:0)
src_mac=00:e0:4c:b9:97:7c
misc=0 policy_id=6 pol_uuid_idx=606 auth_info=0 chk_client_info=0 vd=0
serial=007e0aab tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
```

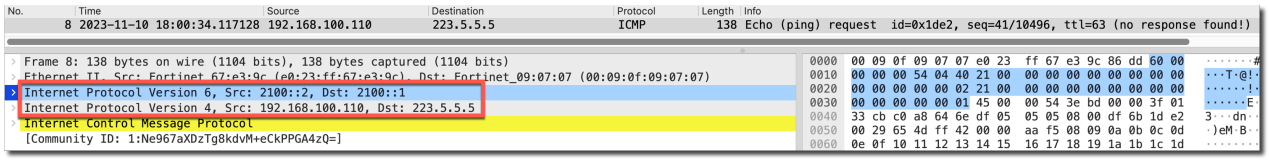
```
npu info: flag=0x84/0x81, offload=8/8, ips_offload=0/0, epid=66/95, ipid=95/66, vlan=0x0000/0x0000
vlifid=95/245, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=1/1

B4 # diagnose sys session6 list
total session6 0:
```

5. 在 AFTR 上查看 IPv4 流量（VNE 隧道内层）的会话，流量被 NP 加速，无法看到 IPv6 的流量会话。

```
session info: proto=1 proto_state=00 duration=238 expire=28 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0
use=4
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=12300/205/1 reply=12300/205/1 tuples=2
tx speed(Bps/kbps): 51/0 rx speed(Bps/kbps): 51/0
orgin->sink: org pre->post, reply pre->post dev=34->5/5->34 gwy=172.22.5.1/192.168.100.77
hook=post dir=org act=snat 192.168.100.77:1->223.5.5.5:8(172.22.5.77:1)
hook=pre dir=reply act=dnat 223.5.5.5:1->172.22.5.77:0(192.168.100.77:1)
misc=0 policy_id=11 pol_uuid_idx=594 auth_info=0 chk_client_info=0 vd=0
serial=005f4689 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x84, offload=8/8, ips_offload=0/0, epid=64/71, ipid=71/64, vlan=0x0000/0x0000
vlifid=247/64, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=1/2

AFTR # diagnose sys session6 list
total session6 0:
```

bash

6. 对应的 DS-Lite IPv4 over IPv6 报文，就是 IPv4 over IPv6 封装，没有其他协议层。

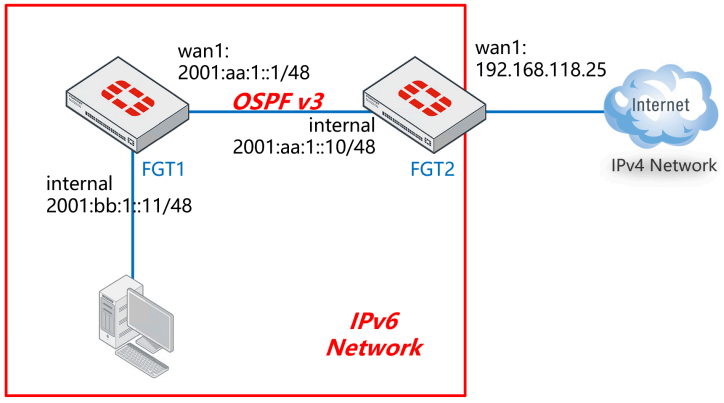


OSPFv3

网络需求

在 IPv6 网络内部使用 OSPFv3。

网络拓扑



配置要点

FGT1：

- 配置基本上网功能
- 配置 OSPFv3

FGT2：

- 配置 NAT64 访问 IPv4 Internet
- 配置 OSPF

配置步骤

FGT1

1. 配置基本上网功能，参考 IPv6 → 上网配置 ↗章节。

```
config system interface
    edit "internal"
        config ipv6
            set ip6-allowaccess ping https http
            set ip6-address 2001:bb:1::1/48
        next
    edit "wan1"
           config ipv6
            set ip6-allowaccess ping https
            set ip6-address 2001:aa:1::1/48
        next
  end
```

2. 配置 OSPFv3。

```
config router ospf6
    set router-id 192.168.1.200    //指定路由ID
    config area
        edit 0.0.0.0    //配置area 0
        next
    end
    config ospf6-interface
        edit "wan1"    // 名字，可自定义
```

```
            set interface "wan1"    //配置wan1口运行OSPFv3
        next
        end
    config redistribute "connected"    //重分发直连路由
        set status enable
    end
end
```

FGT2

1. 配置 NAT64 上网功能（参考 NAT64 + DNS64 上网 ↗章节）。

2. 配置 OSPFv3。

```
config router ospf6
   set default-information-originate always    //向OSPF邻居FGT1发布一条默认路由
   set router-id 192.168.1.99    //配置路由ID
       config area    //配置区域 0.0.0.0
           edit 0.0.0.0
           next
       end
       config ospf6-interface     //指定internal接口运行OSPF协议
           edit "internal"    //名字，可自定义
               set interface "internal"
           next
       end
end
```

结果验证

1. 查看 FGT1 OSPF 邻居。

```
FortiGate # get router info6   ospf  neighbor
OSPFv3 Process (*null*)
Neighbor ID     Pri   State           Dead Time   Interface  Instance ID
192.168.1.99      1   Full/Backup     00:00:34    wan1       0
```

2. 查看 FGT1 路由表。

```
FortiGate # get  router  info6   routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
      IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      I - IS-IS, B - BGP
      * - candidate default
Timers: Uptime
O*E2    ::/0 [110/1] via fe80::a5b:eff:fe6f:f7a6, wan1, 00:20:29    //学到的默认路由
C       ::1/128 via ::, root, 02:21:41
C       2001:aa:1::/48 via ::, wan1, 02:13:38
C       2001:bb:1::/48 via ::, internal, 01:58:51
```

3. FGT1 用户可以通过 FGT2 访问 IPv4 互联网。

常用命令

1. IPv6 抓包。

```
FGT # diagnose sniffer packet any 'host 2001:aa:1::10' 4    //基于ipv6地址
interfaces=[any]
filters=[host 2001:aa:1::10]
1.389573 internal in 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 415
1.389692 wan1 out 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 415
1.389912 wan1 in 2001:aa:1::10 → 2001:bb:1::10: icmp6: echo reply seq 415
1.389983 internal out 2001:aa:1::10 → 2001:bb:1::10: icmp6: echo reply seq 415
2.391299 internal in 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 416
2.391426 wan1 out 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 416
2.391671 wan1 in 2001:aa:1::10 → 2001:bb:1::10: icmp6: echo reply seq 416
2.391735 internal out 2001:aa:1::10 → 2001:bb:1::10: icmp6: echo reply seq 416
8 packets received by filter
0 packets dropped by kernel

FGT # diagnose sniffer packet any icmp6 4 2    //基于icmp6
interfaces=[any]
filters=[icmp6]
1.410860 internal in 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 431
1.410986 wan1 out 2001:bb:1::10 → 2001:aa:1::10: icmp6: echo request seq 431
```

2. IPv6 会话。

```
FGT # diagnose sys session6 list
session6 info: proto=17 proto_state=01 duration=0 expire=179 timeout=0 flags=00000000 sockport=0 sockflag=0 use=3
origin-shaper=shared-1M-pipe prio=2  guarantee 0Bps  max 131072Bps  traffic 787Bps)
reply-shaper=shared-1M-pipe prio=2  guarantee 0Bps  max 131072Bps traffic 787Bps)
per_ip_shaper=
ha_id=0
policy_dir=0 tunnel=/
state=may_dirty os rs
statistic(bytes/packets/allow_err): org=83/1/0 reply=276/1/0 tuples=2
orgin→sink: org pre→post, reply pre→post dev=3→5/5→3
hook=pre dir=org act=noop 2001:bb:1::10:57194 →2001:aa:1::10:53(:::0)
hook=post dir=reply act=noop 2001:aa:1::10:53 →2001:bb:1::10:57194(:::0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0 serial=000003f1
npu_state=00000000

FGT # diagnose sys session6 full-stat
session table:           table_size=131072 max_depth=1 used=188
misc info:       session_count=94 setup_rate=20 exp_count=0 clash=0
       memory_tension_drop=0 ephemeral=0/0 removeable=0
delete=0, flush=0, dev_down=0/0
TCP sessions:
        19 in ESTABLISHED state
```

3. debug flow 命令。

```
diagnose debug enable
diagnose debug flow filter6 proto 1
diagnose debug flow show con enable
diagnose debug flow trace start6 10
```

NAT64

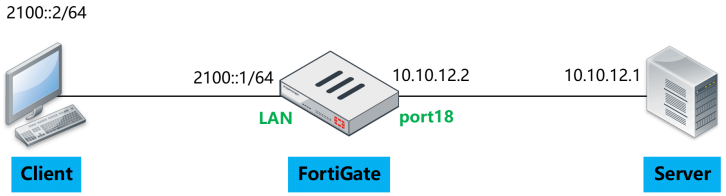
- NAT64 映射
- Embedded 模式的 NAT64
- NAT64 + DNS64 上网

NAT64 映射

网络需求

将 IPv4 内部服务器，发布到 IPv6 网络之中，允许用户通过 IPv6 来对其进行访问。

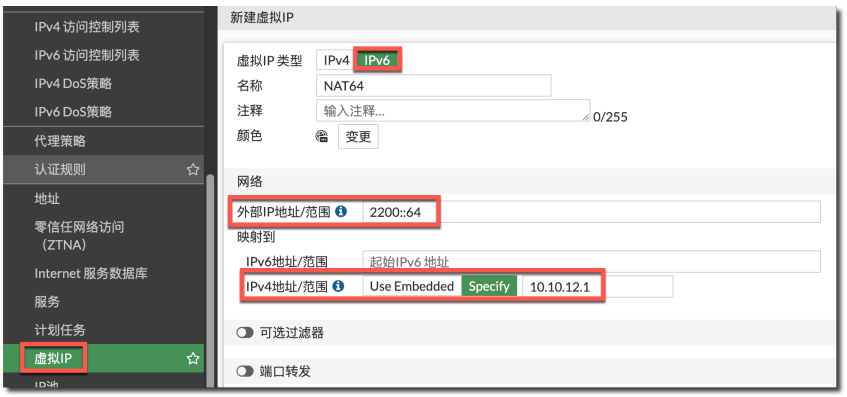
网络拓扑



Client 侧为 IPv6，Server 侧为 IPv4，使用 Client 访问 Server，FortiGate 做 NAT64 转换。

配置步骤

1. 创建 VIP，选择类型为 IPv6，“映射到”填写 IPv4 Server 真实地址，“外部 IP 地址/范围”填写 IPv6 VIP 地址，NAT64 VIP 的外部地址如果是一个范围，必须填写为一个完整的 subnet，否则 GUI 报错。



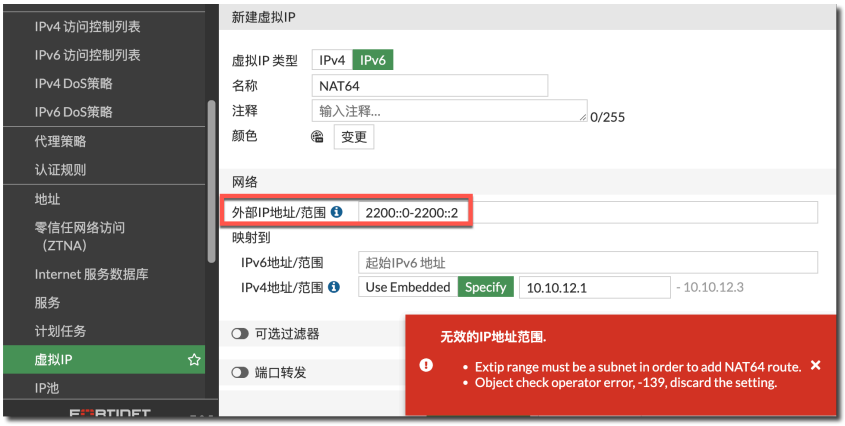

```
config firewall vip6
    edit "NAT64"
        set extip 2200::64
        set nat66 disable
        set nat64 enable
        set ipv4-mappedip 10.10.12.1
        set add-nat64-route enable
    next
end
```

2. 可以看默认存在一个 naf.root 接口，用于处理 NAT64 流量，创建 VIP 完成后，如果开启了 add-nat64-route（默认开启），针对 naf.root 接口生成去往 VIP IPv6 地址的路由，关闭 add-nat64-route 后，该路由会消失。

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

```
FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2200::64/128 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

3. 注意：如果是 NAT64 VIP，“外部 IP 范围”配置的不是一个完整的 subnet，或范围中包含 subnet 的广播地址，则下发时页面或 CLI 会报错，因为无法为 naf 接口下发路由。



4. 创建 IPv4 Pool，配置地址池范围，注意不要和 port18 地址冲突，配置冲突这里并不会提示，后续会讲到为什么不能配成一样的，开启 NAT64 后，须填写为一个完整的 subnet（最小 32 位）的 IPv4 地址范围，否则 GUI 报错。

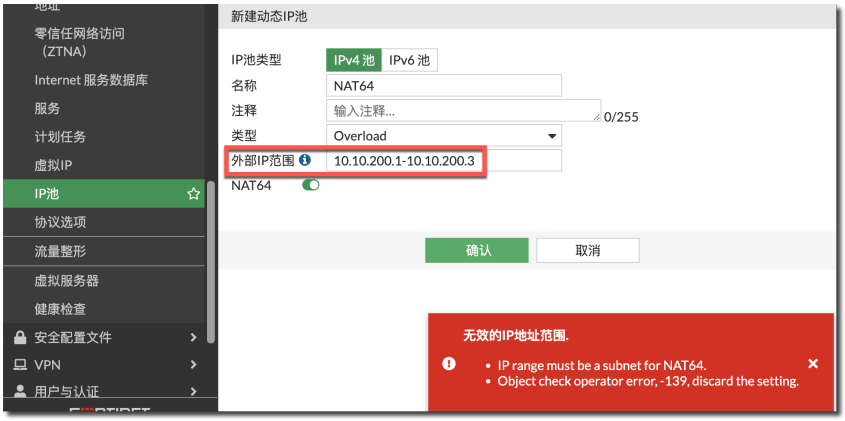



```
config firewall ippool
    edit "NAT64"
        set startip 10.10.200.0
        set endip 10.10.200.127
        set nat64 enable
        set add-nat64-route enable
    next
end
```

5. 创建完成后，如果开启了 add-nat64-route（默认开启），可以看到生成了 IPv4 的 kernel 路由，针对 naf.root 接口生成去往 IPPool IPv4 地址的路由，IPPool 关闭 NAT64 或关闭 add-nat64-route 后，该路由会消失。

```
FortiGate # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0→10.10.200.0/25 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)
```

6. 开启 NAT64 后，如果 IPPool 配置的不是一个完整的 subnet，则下发时页面或 CLI 会报错，因为无法为 naf 接口下发路由。



7. 创建防火墙策略，先在 NAT 选项中开启 NAT64，在 Source 中引用 IPv6 的 all，然后在 Destination 中引用 VIP，在 IPPool 中引用 IPPool。




```
config firewall policy
    edit 2
        set name "NAT64"
        set srcintf "lan"
        set dstintf "port18"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "NAT64"
        set schedule "always"
        set service "ALL"
        set ippool enable
        set poolname "NAT64"
    next
end
```

结果验证与注意事项

1. Client 访问 FortiGate 的 NAT64 VIP 地址 2200::64，抓包查看 NAT64 过程，可以看到经过 naf.root 接口，目标地址被转换为 Server IPv4，源 IPv6 地址被转换为 IPPool 中的 IPv4 地址，返回的流量从 IPv4 被还原为 IPv6。

```
FortiGate-101F # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
2.844158 lan in 2100::2 -> 2200::64: icmp6: echo request seq 78
2.844203 naf.root out 2100::2 -> 2200::64: icmp6: echo request seq 78
2.844210 naf.root in 10.10.200.20 -> 10.10.12.1: icmp: echo request
2.844231 port18 out 10.10.200.20 -> 10.10.12.1: icmp: echo request
2.844401 port18 in 10.10.12.1 -> 10.10.200.20: icmp: echo reply
2.844415 naf.root out 10.10.12.1 -> 10.10.200.20: icmp: echo reply
2.844419 naf.root in 2200::64 -> 2100::2: icmp6: echo reply seq 78
2.844432 lan out 2200::64 -> 2100::2: icmp6: echo reply seq 78
```

2. debug flow IPv4/IPv6，可以看到 DNAT64 整个过程都需要经过 naf.root 接口处理。

```
FortiGate-101F # id=20085 trace_id=876 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:1->2200::64:128) from lan."
id=20085 trace_id=876 func=resolve_ip6_tuple line=4926 msg="allocate a new session-00001d16"
id=20085 trace_id=876 func=get_vip64_addr line=1174 msg="find DNAT64: IP-10.10.12.1, port-8(fixed port)"
id=20085 trace_id=876 func=vf_ip6_route_input line=1201 msg="find a route: gw-2200::64 via naf.root err 0 flags 01000001"
id=20085 trace_id=876 func=fw6_forward_handler line=435 msg="Check policy between lan -> naf.root"
id=20085 trace_id=876 func=get_new_addr64 line=1110 msg="find SNAT64: IP-10.10.200.20(from IPPOOL), port-16716"
id=20085 trace_id=876 func=fw6_forward_handler line=567 msg="Allowed by Policy-2: SNAT"
id=20085 trace_id=876 func=ip6_nat_af_input line=298 msg="nat64 ipv6 received a packet proto=58"
id=20085 trace_id=876 func=init_ip_session_common line=6003 msg="allocate a new session-0014068b, tun_id=0.0.0.0"
id=20085 trace_id=876 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-10.10.12.1 via port18"
id=20085 trace_id=876 func=fw_forward_handler line=874 msg="Allowed by Policy-2:"
id=20085 trace_id=873 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 10.10.12.1:16716->10.10.200.20:0) tun_id=0.0.0.0 from port18. type=0, code=0
, id=16716, seq=87."
id=20085 trace_id=873 func=resolve_ip_tuple_fast line=5910 msg="Find an existing session, id-0014068b, reply direction"
id=20085 trace_id=873 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-10.10.200.20 via naf.root"
id=20085 trace_id=873 func=fw_forward_dirty_handler line=410 msg="state=00000200, state2=00000000, npu_state=04040000"
id=20085 trace_id=873 func=ip4_nat_af_input line=261 msg="nat64 ipv4 received a packet proto=1"
id=20085 trace_id=873 func=vf_ip6_route_input line=1201 msg="find a route: gw-2100::2 via lan err 0 flags 01000001"
id=20085 trace_id=873 func=fw6_forward_dirty_handler line=371 msg="state=00010200, state2=00000000, npu_state=04040000"
```

3. 在 FortiGate 查看 IPv6 会话，可以看到源接口为物理入接口 lan，目标接口为 index 37，为 naf.root，naf 被标记为 1，IPv6 流量未被加速，会话中还包含相关的 IPv4 的相关信息。

```
session6 info: proto=58 proto_state=00 duration=1 expire=58 timeout=0 flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty npu
statistic(bytes/packets/allow_err): org=80/1/0 reply=80/1/0 tuples=2
tx speed(Bps/kbps): 48/0 rx speed(Bps/kbps): 48/0
orgin->sink: org pre->post, reply pre->post dev=41->37/37->41
hook=pre dir=org act=noop 2100::2:1->2200::64:128(:::0)
hook=post dir=reply act=noop 2200::64:1->2100::2:129(:::0)
peer=10.10.200.20:33100->10.10.12.1:8 naf=1
hook=pre dir=org act=noop 10.10.200.20:33100->10.10.12.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.12.1:33100->10.10.200.20:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00001d1f tos=ff/ff ips_view=25718 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
```

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

4. 在 FortiGate 查看 IPv4 会话，可以看到源接口为 index 37，为 naf.root，目标接口为物理出接口 port18，naf 被标记为 2，IPv4 流量未被加速，会话中还包含相关的 IPv6 会话信息。

```
session info: proto=1 proto_state=00 duration=1 expire=58 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=60/1/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 56/0 rx speed(Bps/kbps): 56/0
orgin->sink: org pre->post, reply pre->post dev=37->30/30->37 gwy=10.10.12.1/10.10.200.20
hook=pre dir=org act=noop 10.10.200.20:12620->10.10.12.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.12.1:12620->10.10.200.20:0(0.0.0.0:0)
peer=2200::64:1->2100::2:129 naf=2
hook=pre dir=org act=noop 2100::2:1->2200::64:128(:::0)
hook=post dir=reply act=noop 2200::64:1->2100::2:129(:::0)
misc=0 policy_id=2 pol_uuid_idx=563 auth_info=0 chk_client_info=0 vd=0
serial=0014090b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
```

```
FortiGate-101F # diagnose netlink interface list | grep port18
if=port18 family=00 type=1 index=30 mtu=1500 link=0 master=0
```

5. 修改 IPPool 的地址范围和 IPv4 出接口地址一致，再次测试 NAT64 转换。

```
config firewall ippool
    edit "NAT64"
        set startip 10.10.12.2
        set endip 10.10.12.2
        set nat64 enable
```

```
    next
end
```

6. 查看 kernel 路由中关于 naf.root 的路由，可以看到未产生 IPPool 的路由，这样返回的流量肯定是被丢掉的，接下来验证一下。

```
FortiGate-101F # get router info kernel | grep naf

FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2200::64/128 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

7. debug flow IPv4 & IPv6 可以看到，Client 到 Server 方向的 NAT46 的 DNAT 和 SNAT 均正常工作，但返回的 IPv6 流量未匹配到 naf.root 接口的路由，而 gw 查找到的是 root 接口，返回的流量无法到达 Client。

```
FortiGate-101F # id=20085 trace_id=925 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:1->2200::64:128) from lan."
id=20085 trace_id=925 func=resolve_ip6_tuple line=4926 msg="allocate a new session-00001d49"
id=20085 trace_id=925 func=get_vip64_addr line=1174 msg="find DNAT64: IP-10.10.12.1, port-8(fixed port)"
id=20085 trace_id=925 func=vf_ip6_route_input line=1201 msg="find a route: gw-2200::64 via naf.root err 0 flags 01000001"
id=20085 trace_id=925 func=fw6_forward_handler line=435 msg="Check policy between lan -> naf.root"
id=20085 trace_id=925 func=get_new_addr64 line=1110 msg="find SNAT64: IP-10.10.12.2(from IPPOOL), port-8524"
id=20085 trace_id=925 func=fw6_forward_handler line=567 msg="Allowed by Policy-2: SNAT"
id=20085 trace_id=925 func=ip6_nat_af_input line=298 msg="nat64 ipv6 received a packet proto=58"
id=20085 trace_id=925 func=init_ip_session_common line=6003 msg="allocate a new session-00140e71, tun_id=0.0.0.0"
id=20085 trace_id=925 func=vf_ip_route_input_common line=2604 msg="find a route: flag=00000000 gw-10.10.12.1 via port18"
id=20085 trace_id=925 func=fw_forward_handler line=874 msg="Allowed by Policy-2:"
id=20085 trace_id=876 func=print_pkt_detail line=5824 msg="vd-root:0 received a packet(proto=1, 10.10.12.1:8524->10.10.12.2:0) tun_id=0.0.0.0 from port18. type=0, code=0, id=8524, seq=90."
id=20085 trace_id=876 func=resolve_ip_tuple_fast line=5910 msg="Find an existing session, id-00140e71, reply direction"
id=20085 trace_id=876 func=vf_ip_route_input_common line=2604 msg="find a route: flag=84000000 gw-10.10.12.2 via root"
```

8.**所以配置 NAT64 时，需要注意，IPPool 不要配置成和出接口地址相同的地址，配置不会报错，但实际会不通，如果是 IPv4 的第二地址，实际测试也是不行的。**

9. 还原 IPPool 的配置，这次修改 VIP6 的 External IP 和物理入接口 IP 一致。

```
config firewall vip6
    edit "NAT64"
        set extip 2100::1
        set nat66 disable
        set nat64 enable
        set ipv4-mappedip 10.10.12.1
    next
end
```

0. 查看 kernel 路由中关于 naf.root 的路由，可以看到未产生 naf.root 接口的 VIP6 的路由，这样 NAT64 的 Client 发起方流量应该是无法进入 NAT64 流程的，接下来验证一下。

```
FortiGate-101F # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->10.10.200.0/24 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)

FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

1. debug flow IPv4 & IPv6 可以看到，Client 到 Server 方向的 NAT64 流量被 FortiGate 路由到 root 接口，未进入 naf.root 接口，由 FortiGate 直接返回了 reply 报文，没有 IPv4 的 debug flow 产生。

```
FortiGate-101F # id=20085 trace_id=973 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:1->2100::1:128) from lan."
id=20085 trace_id=973 func=resolve_ip6_tuple line=4926 msg="allocate a new session-00001d70"
id=20085 trace_id=973 func=get_vip64_addr line=1174 msg="find DNAT64: IP-10.10.12.1, port-8(fixed port)"
id=20085 trace_id=973 func=vf_ip6_route_input line=1201 msg="find a route: gw-:: via root err 0 flags 80200001"
id=20085 trace_id=974 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::1:1->2100::2:129) from local."
id=20085 trace_id=974 func=resolve_ip6_tuple_fast line=4827 msg="Find an existing session, id-00001d70, reply direction"
```

12.**所以配置 NAT64 时，需要注意，IPv6 VIP 的 extip 不要配置成和入接口地址相同的地址，配置不会报错，但实际会不通，如果是 IPv6 的第二地址，实际测试也是不行的。**

CLI 相关

VIP CLI

1. IPv4 类型的 VIP 支持三种组合，NAT44、NAT46、NAT44 + NAT46。

```
config firewall vip
    edit "vip1"
        set extip 30.0.0.110-30.0.0.120
        set extintf "port3"
        set arp-reply disable
        set portforward enable
```

```
        set srcintf-filter "port3"
        set nat44 disable    ←──────── new option enabled by default
        set nat46 enable    ←──────── new option disabled by default
        set ipv6-mappedip 2040::20-2040::30    ←──────── new option available when nat46 enabled
        set extport 10
        set ipv6-mappedport 100    ←──────── new option available when nat46 enabled
    next
end
```

2. IPv6 类型的 VIP 支持三种组合，NAT66、NAT64、NAT66 + NAT64。

```
config firewall vip6
    edit "vip3"
        set extip 2002::100-2002::200
        set portforward enable
        set nat66 disable    ←──── new option enabled by default
        set nat64 enable    ←──── new option disabled by default
        set ipv4-mappedip 8.8.8.8-8.8.8.108    ←──────── new option available when nat64 is enabled
        set ipv4-mappedport 1000    ←──── new option available when nat64 is enabled
        set extport 100
    next
end
```

3. NAT64 也支持 Virtual Server，开启 NAT64 后，RS 必须配置为 IPv4 地址，实现的 NAT 过程和普通的 NAT64 是一样的。

```
config firewall vip6
    edit "vip2"
        set type server-load-balance
        set extip 2002::2
        set server-type udp
        set monitor "ping-mon-l"
        set nat64 enable
        set extport 53
        config realservers
            edit 1
                set ip 8.8.8.8    ←──────── server ip must be ipv4 when nat64 is abled
                set port 53
            next
            edit 2
                set ip 8.8.4.4
                set port 53
            next
        end
    next
end
```

IPPool CLI

1. IPv4 类型的 IPPool，类型为 NAT64，地址池范围必须是一个 subnet 的范围，默认会开启 add-nat64-route。

```
config firewall ippool
    edit "pol64"
        set startip 10.0.0.0    ←──── ip range must be a subnet if nat64 enabled
        set endip 10.0.255.255
        set nat64 enable    ←──── new option
        set add-nat64-route enable    ←──────── new option available when nat64 enabled (by default enable)
    next
end
```

2. IPv6 类型的 IPPool，类型为 NAT46，地址池范围必须是一个 subnet 的范围，默认会开启 add-nat46-route。

```
config firewall ippool6
    edit "pol46"
        set startip 2002::    ←──────── ip range must be a subnet if nat46 enabled
        set endip 2002::ffff
        set nat46 enable
        set add-nat46-route enable    ←──────── new option available when nat46 enabled (by default enable)
    next
end
```

Firewall Policy & Central SNAT CLI

1. NAT64 Firewall Policy。

```
config firewall policy
     edit 3
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"    <——— both ipv4 and ipv6 addresses are required when nat64 enable
        set srcaddr6 "all"
        set dstaddr6 "vip3"    <—— vip6 with nat64 enabled
        set action accept
        set schedule "always"
        set service "ALL"
        set ippool enable
        set poolname "po1"    <——— ipv4 pool with nat64 enabled
        set nat64 enable    <—— new option to enable nat64
    next
end
```

2. NAT64 Central SNAT。

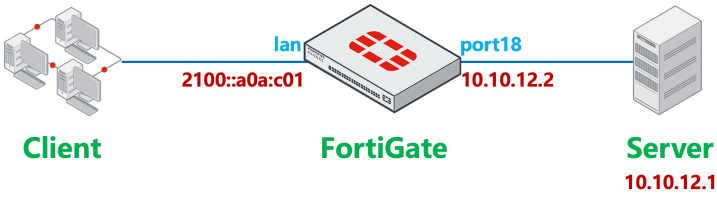
```
config firewall central-snat-map
    edit 3
        set type ipv6
        set srcintf "any"
        set dstintf "any"
        set orig-addr6 "all"
        set dst-addr6 "all"
        set nat64 enable    <——— new option for ipv6 rule
        set nat-ippool "po1"    <——— available when nat64 is enabled, ipv4 pool with nat64 enabled
    next
end
```

Embedded 模式的 NAT64

网络需求

将 IPv4 内部服务器，发布到 IPv6 网络之中，同时将 IPv4 服务器的 IPv4 地址嵌入 IPv6 VIP 地址的最后 32 位中，允许用户通过 IPv6 来对其进行访问。

网络拓扑



Client 侧为 IPv6，Server 侧为 IPv4，FortiGate 做 NAT64 转换，将 IPv4 服务器的 IPv4 地址嵌入 IPv6 VIP 地址的最后 32 位中，使用 Client 访问 Server 的 IPv6 VIP 地址。

配置步骤

1. 创建 VIP，选择类型为 IPv6，“外部 IP 地址/范围”填写 IPv6 VIP 地址，“映射到”选择 Use Embedded，这里会根据 IPv6 外部地址的最后 32 位二进制数字，计算出映射的 IPv4 地址，并显示在提示信息中，这里计算出的是 Server IPv4 10.10.12.1。




```
config firewall vip6
    edit "NAT64_embedded"
        set extip 2100::a0a:c01
        set nat66 disable
        set nat64 enable
        set add-nat64-route enable
        set embedded-ipv4-address enable
    next
end
```

2. 可以看默认存在一个 naf.root 接口，用于处理 NAT64 流量，创建 VIP 完成后，如果开启了 add-nat64-route（默认开启），针对 naf.root 接口生成去往 VIP IPv6 地址的路由，关闭 add-nat64-route 后，该路由会消失。

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

```
FortiGate-101F # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2100::a0a:c02/128 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

3. 创建 IPv4 Pool，配置地址池范围，配置地址池范围，注意不要和 port18 地址冲突，开启 NAT64 后，须填写为一个完整的 subnet（最小 32 位）的 IPv4 地址范围。




```
config firewall ippool
    edit "NAT64_embedded"
        set startip 10.10.200.0
        set endip 10.10.200.127
        set nat64 enable
        set add-nat64-route enable
    next
end
```

4. 创建完成后，如果开启了 add-nat64-route（默认开启），可以看到生成了 IPv4 的 kernel 路由，针对 naf.root 接口生成去往 IPPool IPv4 地址的路由，IPPool 关闭 NAT64 或关闭 add-nat64-route 后，该路由会消失。

```
FortiGate # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->10.10.200.0/25 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)
```

5. 创建防火墙策略，先在 NAT 选项中开启 NAT64，在源地址中引用 IPv6 的 all，然后在目标地址中引用 VIP，在 IP 池配置中引用 IPPool。




```
config firewall policy
    edit 2
        set name "NAT64"
        set srcintf "lan"
        set dstintf "port18"
        set action accept
        set nat64 enable
```

```
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "NAT64"
        set schedule "always"
        set service "ALL"
        set ippool enable
        set poolname "NAT64"
    next
end
```

结果验证

1. Client 访问 FortiGate 的 NAT64 VIP 地址 2100::a0a:c01，抓包查看 NAT64 过程，可以看到经过 naf.root 接口，目标地址被转换为 IPv6 的 extip 的最后 32 位二进制数字的 embedded IP 10.10.12.1。

```
FortiGate-101F # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
2.834516 lan in 2100::2 -> 2100::a0a:c01: icmp6: echo request seq 117
2.834657 naf.root out 2100::2 -> 2100::a0a:c01: icmp6: echo request seq 117
2.834667 naf.root in 10.10.200.20 -> 10.10.12.1: icmp: echo request
2.834751 port18 out 10.10.200.20 -> 10.10.12.1: icmp: echo request
2.834945 port18 in 10.10.12.1 -> 10.10.200.20: icmp: echo reply
2.834971 naf.root out 10.10.12.1 -> 10.10.200.20: icmp: echo reply
2.834978 naf.root in 2100::a0a:c01 -> 2100::2: icmp6: echo reply seq 117
2.835021 lan out 2100::a0a:c01 -> 2100::2: icmp6: echo reply seq 117
```

2. debug flow IPv4/IPv6，可以看到 DNAT64 整个过程都需要经过 naf.root 接口处理。

```
FortiGate-101F # id=20085 trace_id=1375 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::2:1->2100::a0a:c01:128) from lan."
id=20085 trace_id=1375 func=resolve_ip6_tuple line=4926 msg="allocate a new session-000028bd"
id=20085 trace_id=1375 func=get_vip64_addr line=1174 msg="find DNAT64: IP-10.10.12.1 port-8(fixed port)"
id=20085 trace_id=1375 func=vf_ip6_route_input line=1201 msg="find a route: gw-2100::a0a:c01 via naf.root err 0 flags 01000001"
id=20085 trace_id=1375 func=fw6_forward_handler line=435 msg="Check policy between lan -> naf.root"
id=20085 trace_id=1375 func=get_new_addr64 line=1110 msg="find SNAT64: IP-10.10.200.20(from IPPOOL) port-53580"
id=20085 trace_id=1375 func=fw6_forward_handler line=567 msg="Allowed by Policy-2: SNAT"
id=20085 trace_id=1375 func=ip6_nat_af_input line=298 msg="nat64 ipv6 received a packet proto=58"
id=20085 trace_id=1375 func=init_ip_session_common line=6003 msg="allocate a new session-0015ead7, tun_id=0.0.0.0"
id=20085 trace_id=1375 func=vf_ip_route_input_common line=2604 msg="find a route: flag=04000000 gw-10.10.12.1 via port18"
id=20085 trace_id=1375 func=fw_forward_handler line=874 msg="Allowed by Policy-2:"
id=20085 trace_id=1376 func=resolve_ip6_tuple_fast line=4787 msg="vd-root:0 received a packet(proto=58, 2100::a0a:c01:1->2100::2:129) from naf.root."
id=20085 trace_id=1376 func=resolve_ip6_tuple_fast line=4827 msg="find an existing session, id-000028bd, reply direction"
id=20085 trace_id=1376 func=vf_ip6_route_input line=1201 msg="find a route: gw-2100::2 via lan err 0 flags 01000001"
id=20085 trace_id=1376 func=fw6_forward_dirty_handler line=371 msg="state=00010200, state2=00000000, npu_state=04040000"
```

3. 在 FortiGate 查看 IPv6 会话，可以看到源接口为物理入接口 lan，目标接口为 index 37，为 naf.root，naf 被标记为 1，IPv6 流量未被加速，会话中还包含相关的 IPv4 的相关信息。

```
session6 info: proto=58 proto_state=00 duration=7 expire=52 timeout=0 flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty npu
statistic(bytes/packets/allow_err): org=80/1/0 reply=80/1/0 tuples=2
tx speed(Bps/kbps): 10/0 rx speed(Bps/kbps): 10/0
orgin->sink: org pre->post, reply pre->post dev=41->37/37->41
hook=pre dir=org act=noop 2100::2:1->2100::a0a:c01:128(:::0)
hook=post dir=reply act=noop 2100::a0a:c01:1->2100::2:129(:::0)
peer=10.10.200.20:53580->10.10.12.1:8 naf=1
hook=pre dir=org act=noop 10.10.200.20:53580->10.10.12.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.12.1:53580->10.10.200.20:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=000028bd tos=ff/ff ips_view=13872 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0 ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
```

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

4. 在 FortiGate 查看 IPv4 会话，可以看到源接口为 index 37，为 naf.root，目标接口为物理出接口 port18，naf 被标记为 2，IPv4 流量未被加速，会话中还包含相关的 IPv6 会话信息。

```
session info: proto=1 proto_state=00 duration=10 expire=49 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=60/1/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=37->30/30->37 gwy=10.10.12.1/10.10.200.20
hook=pre dir=org act=noop 10.10.200.20:53580->10.10.12.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.12.1:53580->10.10.200.20:0(0.0.0.0:0)
peer=2100::a0a:c01:1->2100::2:129 naf=2
hook=pre dir=org act=noop 2100::2:1->2100::a0a:c01:128(:::0)
hook=post dir=reply act=noop 2100::a0a:c01:1->2100::2:129(:::0)
misc=0 policy_id=2 pol_uuid_idx=567 auth_info=0 chk_client_info=0 vd=0
serial=0015ead7 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4040000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
```

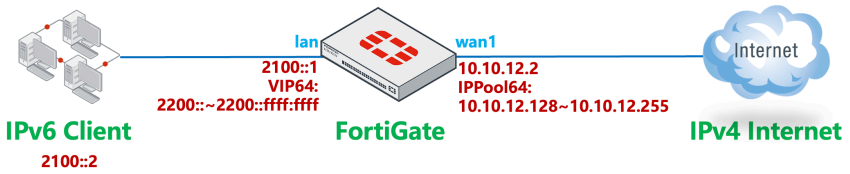
```
FortiGate-101F # diagnose netlink interface list | grep port18
if=port18 family=00 type=1 index=30 mtu=1500 link=0 master=0
```

NAT64 + DNS64 上网

网络需求

内网用户为 IPv6 网络，Internet 网络为 IPv4 网络，内部 IPv6 用户通过 NAT64 + DNS64 访问 IPv4 的 Internet 网络。

网络拓扑



1. Client 侧为 IPv6，Server 侧为 IPv4，FortiGate 做 NAT64 + DNS64 转换，内部 IPv6 用户通过 NAT64 + DNS64 访问 IPv4 的 Internet 网络。
2. 配置 IPv6 Client 的 IPv6 为 21001，FortiGate 内网口 lan 的 IPv6 为 2100::1/64。
3. 配置 FortiGate 外网口 wan1 的 IP 为 10.10.12.2/24，默认路由指向 IPv4 Internet 10.10.12.1。

配置步骤

NAT64 配置

1. 创建 VIP，选择类型为 IPv6，“外部 IP 地址/范围”填写 IPv6 VIP 地址，“映射到”选择 Use Embedded，这里填写的 IPv6 地址范围要写 96 位的 IPv6 前缀（注意不要配置和接口 IPv6 地址相同的网段），最后 32 位范围为 0.0.0.0~255.255.255.255 的 16 进制格式，用于映射所有的 IPv4 地址。




```
config firewall vip6
    edit "NAT64_DNS64"
        set uuid f913176c-48aa-51ee-62dd-5d5b772dede2
        set extip 2200::-2200::ffff:ffff
        set nat66 disable
        set nat64 enable
        set embedded-ipv4-address enable
    next
end
```

2. 可以看默认存在一个 naf.root 接口，用于处理 NAT64 流量，创建 VIP 完成后，如果开启了 add-nat64-route（默认开启），针对 naf.root 接口生成去往 VIP IPv6 地址的路由，关闭 add-nat64-route 后，该路由会消失。

```
FortiGate-101F # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

```
FortiGate # get router info6 kernel | grep naf
vf=0 tbl=254 type=01(unicast) protocol=11(zebra) flag=00000000 prio=1024 dst:2200::/96 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00000000 prio=256 dst:fe80::/64 dev=37(naf.root) pmtu=1500
vf=0 tbl=254 type=01(unicast) protocol=3(boot) flag=00000000 prio=256 dst:ff00::/8 dev=37(naf.root) pmtu=1500
```

3. 创建 IPv4 Pool，配置地址池范围，配置地址池范围，注意不要和 wan1 地址冲突，开启 NAT64 后，须填写为一个完整的 subnet（最小 32 位）的 IPv4 地址范围。




```
config firewall ippool
    edit "NAT64_DNS64"
        set startip 10.10.12.128
        set endip 10.10.12.255
        set nat64 enable
        set add-nat64-route enable
    next
end
```

4. 创建完成后，如果开启了 add-nat64-route（默认开启），可以看到生成了 IPv4 的 kernel 路由，针对 naf.root 接口生成去往 IPPool IPv4 地址的路由，IPPool 关闭 NAT64 或关闭 add-nat64-route 后，该路由会消失。

```
FortiGate # get router info kernel | grep naf
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0→10.10.12.128/25 pref=0.0.0.0 gwy=0.0.0.0 dev=37(naf.root)
```

5. 创建防火墙策略，先在 NAT 选项中开启 NAT64，在源地址中引用 IPv6 的 all，然后在目标地址中引用 VIP，在 IP 池配置中引用 IPPool。



```
config firewall policy
    edit 9
        set name "NAT64_DNS64"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "NAT64_DNS64"
        set schedule "always"
        set service "ALL"
        set ippool enable
        set poolname "NAT64_DNS64"
    next
end
```

DNS64 配置

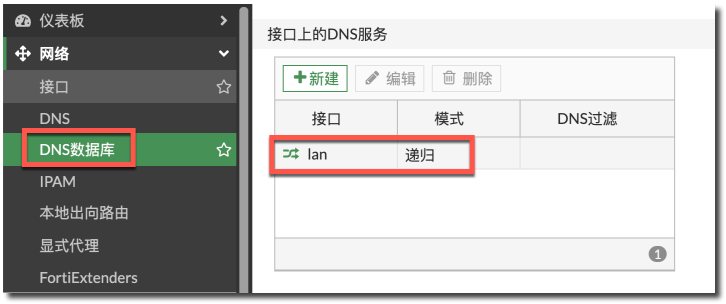
1. 在 CLI 下开启 DNS64，配置 dns64-prefix 为 NAT64 VIP 的前 96 位 2200::/96，确认 always-synthesize-aaaa-record 为开启状态（默认开启）。

```
config system dns64
    set status enable
    set dns64-prefix 2200::/96
    set always-synthesize-aaaa-record enable
end
```

重要

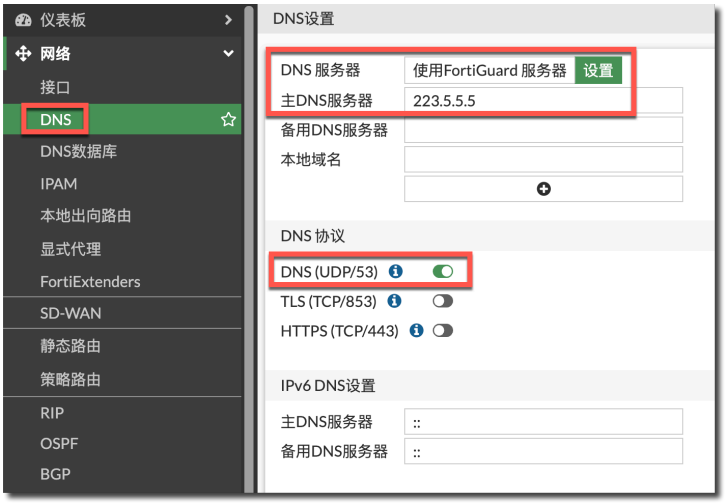
- enable：默认配置，所有从 IPv6 客户端发送的 DNS AAAA Record 请求会被 FortiGate 无条件转换为 A Record 请求，通过 IPv4 DNS 发送至系统配置的 DNS 服务器（不会发送 AAAA 请求），IPv4 DNS 服务器返回 A Record 后，将 A Record 中的 IPv4 地址嵌入 dns64-prefix 的后 32 位中，然后封装为 IPv6 的 AAAA Record 回应，返回给 IPv6 客户端。
- disable：所有从 IPv6 客户端发送的 DNS AAAA Record 请求首先会被 FortiGate 封装在 IPv4 的 DNS 报文中发送至系统配置的 DNS 服务器：
  - 如果 DNS 服务器返回了 AAAA Record，FortiGate 将此 AAAA Record 回应封装为 IPv6 DNS 返回给 IPv6 客户端（这里返回的 AAAA Record 为原服务器响应的 IPv6，FortiGate 不做任何修改）。
  - 如果 DNS 服务器返回的 DNS 报文中没有 AAAA Record 的 IPv6 地址或没有响应 DNS 请求，FortiGate 会执行 always-synthesize-aaaa-record enable 时相同的操作，向 DNS 服务器发送 A Record 请求的 IPv4 DNS 报文，IPv4 DNS 服务器返回 A Record 后，将 A Record 中的 IPv4 地址嵌入 dns64-prefix 的后 32 位中，然后封装为 IPv6 的 AAAA Record 回应，返回给 IPv6 客户端。
- 当 always-synthesize-aaaa-record 配置为 disable 时，DNS 服务器返回的 AAAA Record 会被 FortiGate 不经修改直接返回给 IPv6 客户端，客户端访问此 IPv6 时，可能不在 NAT64 VIP 的 IPv6 地址范围中，导致被 FortiGate 丢弃，所以建议 always-synthesize-aaaa-record 的配置保持默认开启，不要关闭。
- 关于 always-synthesize-aaaa-record 的说明请参考官方文档：https://docs.fortinet.com/document/fortigate/7.99.99/administration-guide/443324/nat64-policy-and-dns64-dns-proxy ↗ 。

2. 在 lan 口上开启 DNS 服务器，IPv6 网络侧用户发起 AAAA Record 查询，FortiGate 的 DNS64 代理请求 IPv4 网络中对应 FQDN 的 A Record，得到 A Record 响应后，DNS64 服务器将 A Record 响应中的 IPv4 地址嵌入 dns64-prefix 的后 32 位中，然后封装为 IPv6 的 AAAA Record 回应，返回给 IPv6 客户端。




```
config system dns-server
    edit "lan"
    next
end
```

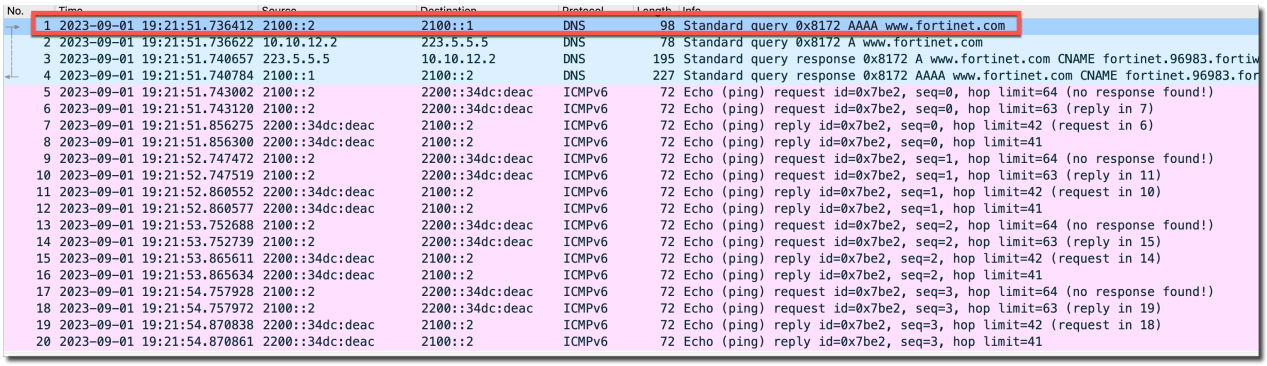
3. 为系统配置 IPv4 DNS 服务器（也可以使用默认服务器）。




```
config system dns
    set primary 223.5.5.5
end
```

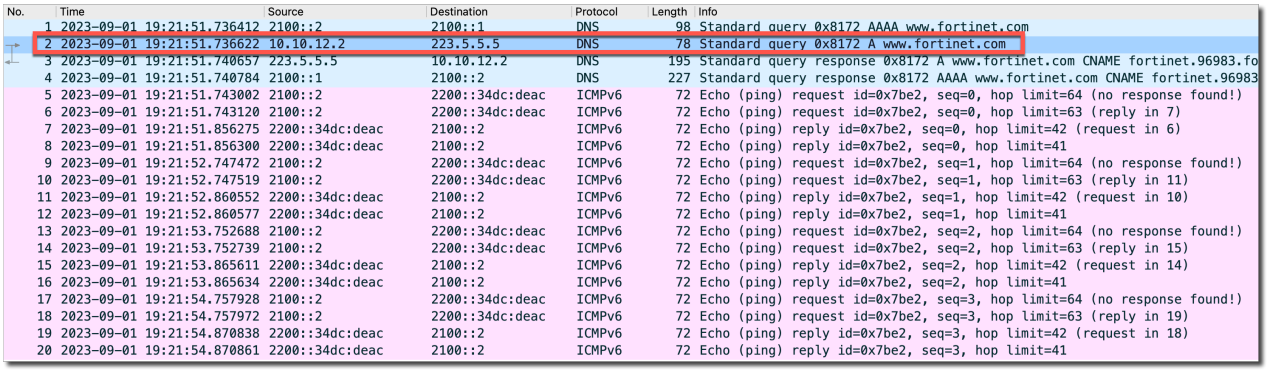
结果验证

1. IPv6 Client 访问 IPv4 Internet 域名，抓包查看 DNS64 过程，客户端发出 IPv6 的 AAAA 请求，请求域名 www.fortinet.com ↗ 的 AAAA Record。



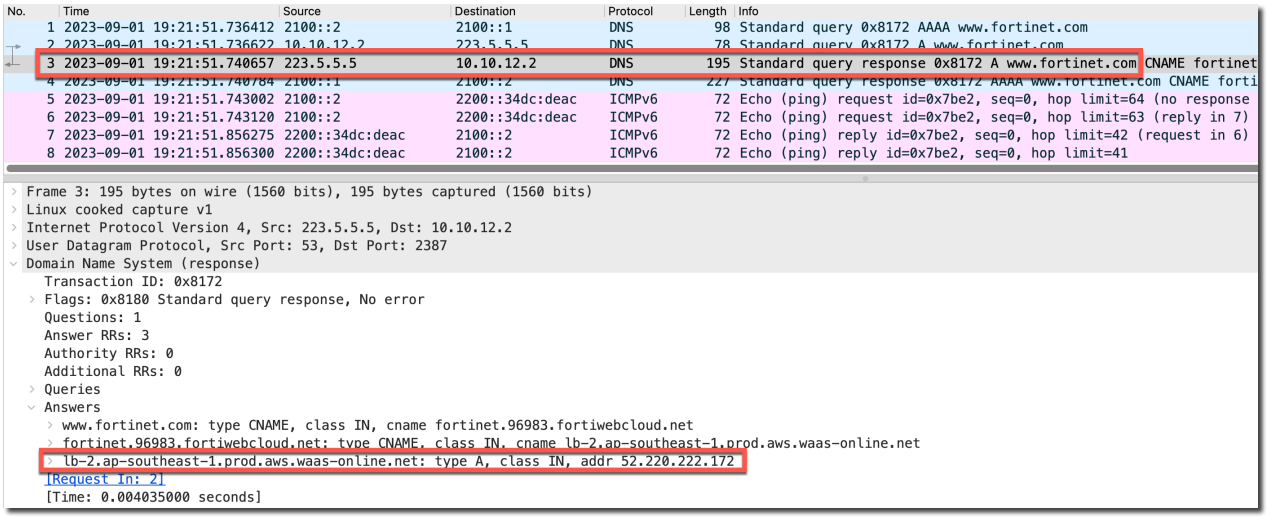
No.	Time	Source	Destination	Protocol	Length	Info
1	2023-09-01 19:21:51.736412	2100::2	2100::1	DNS	98	Standard query 0x8172 AAAA www.fortinet.com
2	2023-09-01 19:21:51.736622	10.10.12.2	223.5.5.5	DNS	78	Standard query 0x8172 A www.fortinet.com
3	2023-09-01 19:21:51.740657	223.5.5.5	10.10.12.2	DNS	195	Standard query response 0x8172 A www.fortinet.com CNAME fortinet.96983.fortiw
4	2023-09-01 19:21:51.740784	2100::1	2100::2	DNS	227	Standard query response 0x8172 AAAA www.fortinet.com CNAME fortinet.96983.for
5	2023-09-01 19:21:51.743002	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=0, hop limit=64 (no response found!)
6	2023-09-01 19:21:51.743120	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=0, hop limit=63 (reply in 7)
7	2023-09-01 19:21:51.856275	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=0, hop limit=42 (request in 6)
8	2023-09-01 19:21:51.856300	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=0, hop limit=41
9	2023-09-01 19:21:52.747472	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=1, hop limit=64 (no response found!)
10	2023-09-01 19:21:52.747519	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=1, hop limit=63 (reply in 11)
11	2023-09-01 19:21:52.860552	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=1, hop limit=42 (request in 10)
12	2023-09-01 19:21:52.860577	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=1, hop limit=41
13	2023-09-01 19:21:53.752688	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=2, hop limit=64 (no response found!)
14	2023-09-01 19:21:53.752739	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=2, hop limit=63 (reply in 15)
15	2023-09-01 19:21:53.865611	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=2, hop limit=42 (request in 14)
16	2023-09-01 19:21:53.865634	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=2, hop limit=41
17	2023-09-01 19:21:54.757928	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=3, hop limit=64 (no response found!)
18	2023-09-01 19:21:54.757972	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=3, hop limit=63 (reply in 19)
19	2023-09-01 19:21:54.870838	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=3, hop limit=42 (request in 18)
20	2023-09-01 19:21:54.870861	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=3, hop limit=41

2. FortiGate 将此 AAAA Record 请求转换为 A Record 请求封装在 IPv4 DNS 中，发送给 DNS 服务器 223.5.5.5。

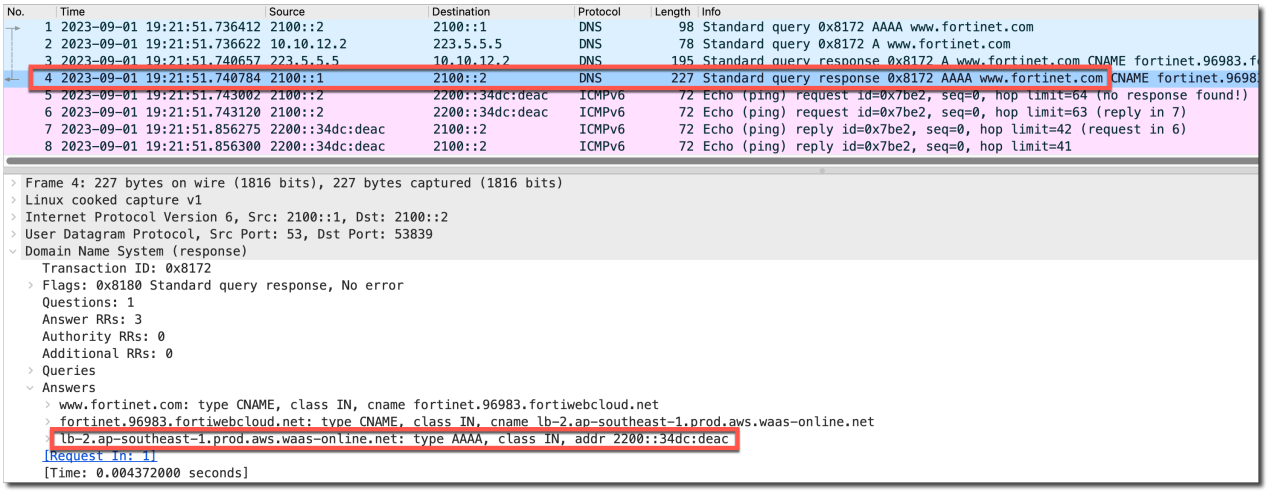


No.	Time	Source	Destination	Protocol	Length	Info
1	2023-09-01 19:21:51.736412	2100::2	2100::1	DNS	98	Standard query 0x8172 AAAA www.fortinet.com
2	2023-09-01 19:21:51.736622	10.10.12.2	223.5.5.5	DNS	78	Standard query 0x8172 A www.fortinet.com
3	2023-09-01 19:21:51.740657	223.5.5.5	10.10.12.2	DNS	195	Standard query response 0x8172 A www.fortinet.com CNAME fortinet.96983.fo
4	2023-09-01 19:21:51.740784	2100::1	2100::2	DNS	227	Standard query response 0x8172 AAAA www.fortinet.com CNAME fortinet.96983
5	2023-09-01 19:21:51.743002	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=0, hop limit=64 (no response found!)
6	2023-09-01 19:21:51.743120	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=0, hop limit=63 (reply in 7)
7	2023-09-01 19:21:51.856275	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=0, hop limit=42 (request in 6)
8	2023-09-01 19:21:51.856300	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=0, hop limit=41
9	2023-09-01 19:21:52.747472	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=1, hop limit=64 (no response found!)
10	2023-09-01 19:21:52.747519	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=1, hop limit=63 (reply in 11)
11	2023-09-01 19:21:52.860552	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=1, hop limit=42 (request in 10)
12	2023-09-01 19:21:52.860577	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=1, hop limit=41
13	2023-09-01 19:21:53.752688	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=2, hop limit=64 (no response found!)
14	2023-09-01 19:21:53.752739	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=2, hop limit=63 (reply in 15)
15	2023-09-01 19:21:53.865611	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=2, hop limit=42 (request in 14)
16	2023-09-01 19:21:53.865634	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=2, hop limit=41
17	2023-09-01 19:21:54.757928	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=3, hop limit=64 (no response found!)
18	2023-09-01 19:21:54.757972	2100::2	2200::34dc:deac	ICMPv6	72	Echo (ping) request id=0x7be2, seq=3, hop limit=63 (reply in 19)
19	2023-09-01 19:21:54.870838	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=3, hop limit=42 (request in 18)
20	2023-09-01 19:21:54.870861	2200::34dc:deac	2100::2	ICMPv6	72	Echo (ping) reply id=0x7be2, seq=3, hop limit=41

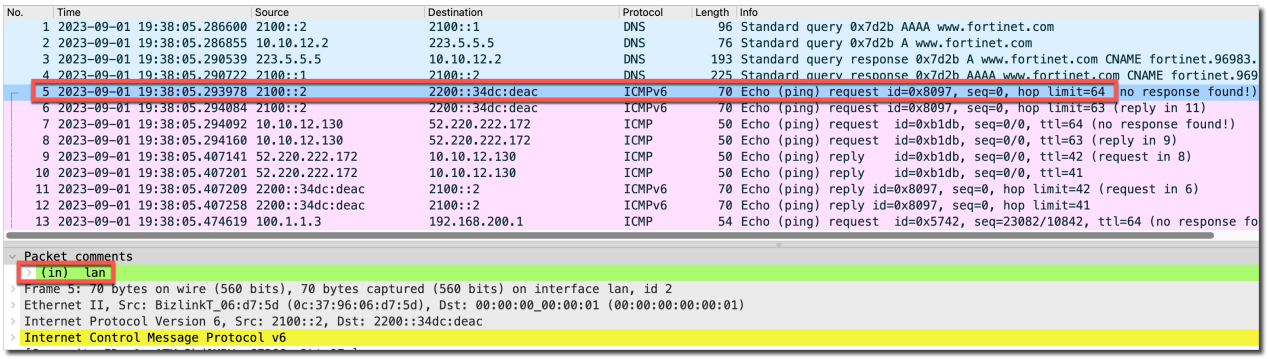
3. DNS 服务器使用 IPv4 DNS 返回该域名的 A Record IPv4 地址 52.220.222.172。



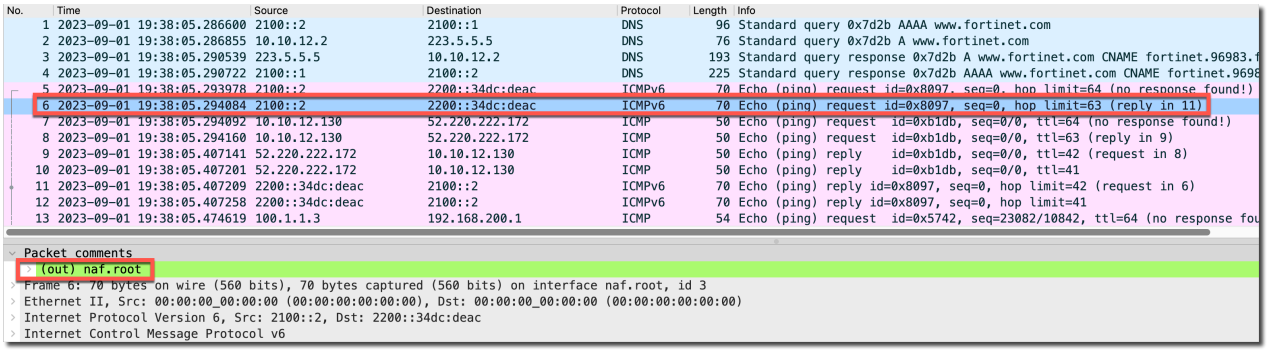
4. FortiGate 收到 DNS 返回的该域名的 A Record IPv4 地址 52.220.222.172 转换为 16 进制后将其嵌入 dns64-prefix 210034dc:deac，将其转换为 AAAA Record，并封装在 IPv6 DNS 报文中返回给 IPv6 Client。



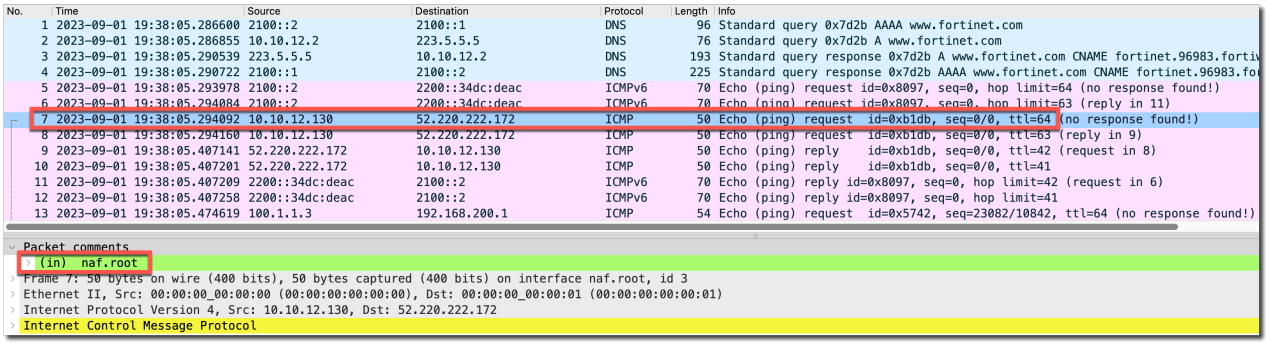
5. 客户端收到 FortiGate 发送的 AAAA Record 回应后，解析到 www.fortinet.com 的 IPv6 IP 为 2200::34dc:deac，所以直接访问该 IPv6 地址，FortiGate 从 lan 口收到该 IPv6 请求包。



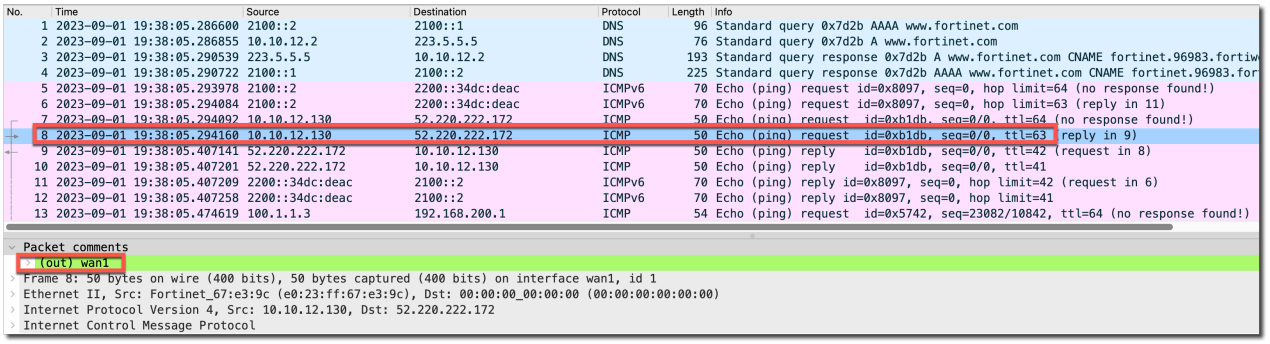
6. 该 IPv6 请求包经由 FortiGate 的 naf 接口，执行 NAT64 转换。



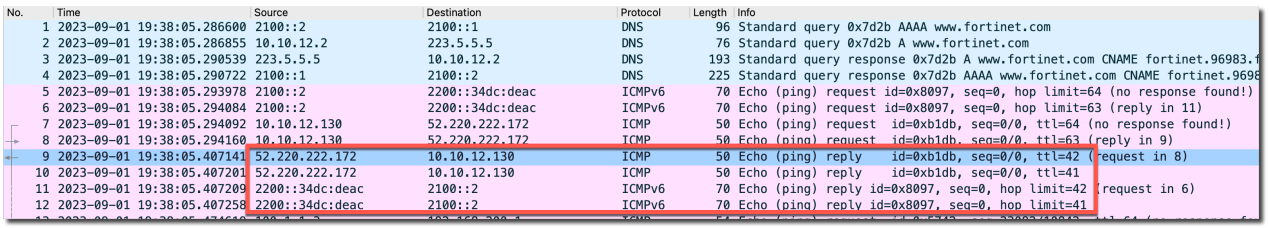
7. 经由 FortiGate 的 naf 接口进行 NAT64 后，转换为 IPv4 报文，源地址为 NAT64 的地址池中的 IPv4 地址，目标地址为原 IPv6 请求包目标 IPv6 地址 2200::34dc:deac 的最后 32 位转换为 10 进制后的 IPv4 地址 52.220.222.172。



8. 最终由 wan1 接口发出送往 Internet 上的目的 IPv4 地址。



9. 服务器 Reply 的 IPv4 报文从 FortiGate 的 wan1 收到，经由 naf 接口做 NAT64 还原后，从 lan 口返回 IPv6 Client。



10. 查看 IPv6 会话，如下所示，会话没有被 NP 加速，流量均经过 naf 接口处理，naf 标记为 1。

```
FortiGate # diagnose sys session6 list

session6 info: proto=58 proto_state=00 duration=110 expire=59 timeout=0 flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty npu npd
statistic(bytes/packets/allow_err): org=6160/110/0 reply=6160/110/0 tuples=2
tx speed(Bps/kbps): 54/0 rx speed(Bps/kbps): 56/0
orgin->sink: org pre->post, reply pre->post dev=45->37/37->45
hook=pre dir=org act=noop 2100::2:33956->2200::34dc:deac:128(:::0)
hook=post dir=reply act=noop 2200::34dc:deac:33956->2100::2:129(:::0)
```

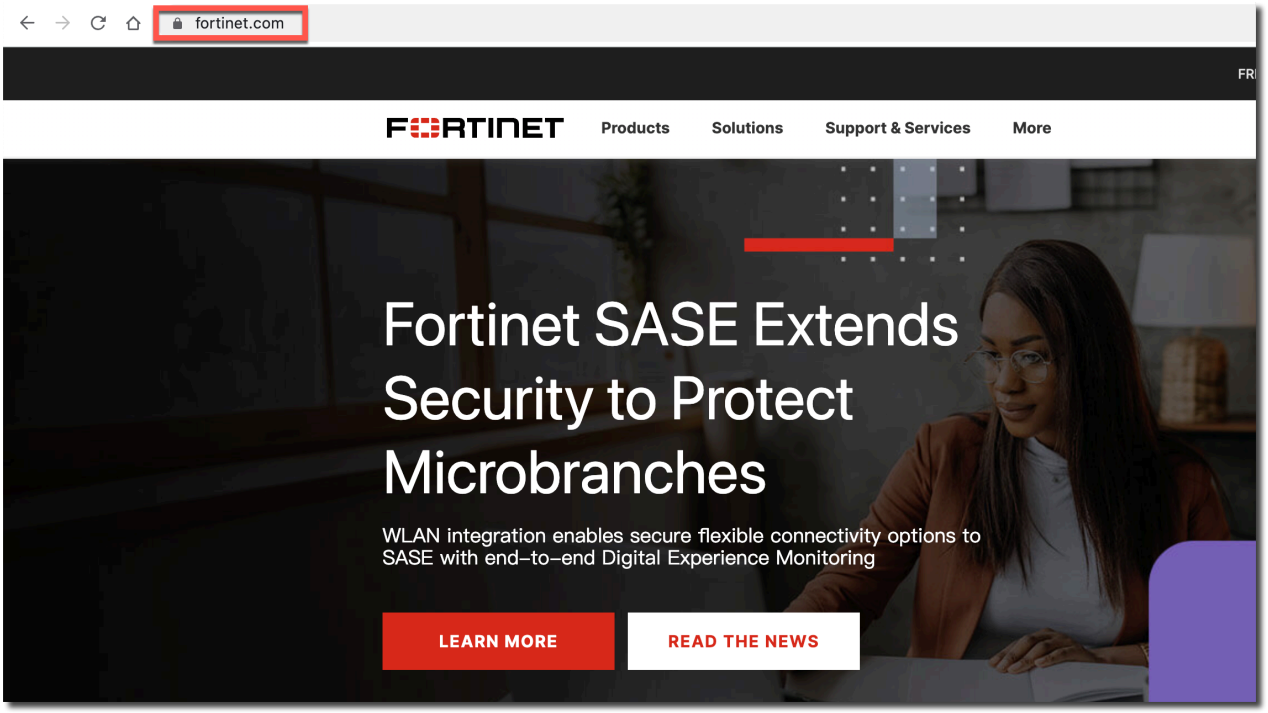
```
peer=10.10.12.130:30184→52.220.222.172:8 naf=1
hook=pre dir=org act=noop 10.10.12.130:30184→52.220.222.172:8(0.0.0.0:0)
hook=post dir=reply act=noop 52.220.222.172:30184→10.10.12.130:0(0.0.0.0:0)
src_mac=0c:37:96:06:d7:5d
misc=0 policy_id=9 pol_uuid_idx=587 auth_info=0 chk_client_info=0 vd=0
serial=0000a1d7 tos=ff/ff ips_view=0 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4040000
npu info: flag=0×81/0×00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=95/0, vlan=0×0000/0×0000
vlifid=0/0, vtag_in=0×0000/0×0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied

FortiGate # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

1. 查看 IPv4 会话，如下所示，会话没有被 NP 加速，流量均经过 naf 接口处理，naf 标记为 2。

```
FortiGate # diagnose sys session list

session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=72/2/1 reply=72/2/1 tuples=2
tx speed(Bps/kbps): 57/0 rx speed(Bps/kbps): 57/0
orgin→sink: org pre→post, reply pre→post dev=37→7/7→37 gwy=10.10.12.1/10.10.12.130
hook=pre dir=org act=noop 10.10.12.130:30184→52.220.222.172:8(0.0.0.0:0)
hook=post dir=reply act=noop 52.220.222.172:30184→10.10.12.130:0(0.0.0.0:0)
peer=2200::34dc:deac:33956→2100::2:129 naf=2
hook=pre dir=org act=noop 2100::2:33956→2200::34dc:deac:128(:::0)
hook=post dir=reply act=noop 2200::34dc:deac:33956→2100::2:129(:::0)
misc=0 policy_id=9 pol_uuid_idx=587 auth_info=0 chk_client_info=0 vd=0
serial=000ecb4f tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4040000
npu info: flag=0×00/0×00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0×0000/0×0000
vlifid=0/0, vtag_in=0×0000/0×0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:

FortiGate # diagnose netlink interface list | grep naf
if=naf.root family=00 type=65535 index=37 mtu=1500 link=0 master=0
```

2. IPv6 客户端可以正常访问 IPv6 的服务器。

```
~ ping6  www.fortinet.com
PING6(56=40+8+8 bytes) 2100::2 --> 2200::34dc:deac
16 bytes from 2200::34dc:deac, icmp_seq=0 hlim=41 time=114.818 ms
16 bytes from 2200::34dc:deac, icmp_seq=1 hlim=41 time=113.894 ms
16 bytes from 2200::34dc:deac, icmp_seq=2 hlim=41 time=114.014 ms
16 bytes from 2200::34dc:deac, icmp_seq=3 hlim=41 time=113.918 ms
16 bytes from 2200::34dc:deac, icmp_seq=4 hlim=41 time=113.714 ms
^C
--- lb-2.ap-southeast-1.prod.aws.waas-online.net ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 113.714/114.072/114.818/0.386 ms
```



附件

附 NAT64 + DNS64 过程抓取的报文：dns64.pcapng ↗。

DHCPv6

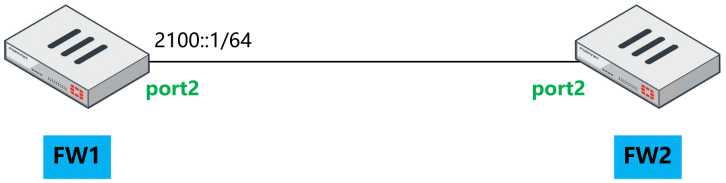
- DHCPv6 原理
- 有状态模式
- 无状态 + 有状态模式
- DHCPv6-PD

有状态模式

网络需求

FortiGate 作为 DHCPv6 服务器，FortiGate 或 PC 为客户端，客户端通过 DHCPv6 获取 IPv6 地址以及 DNS 等信息，不通过 RA 进行自动配置。

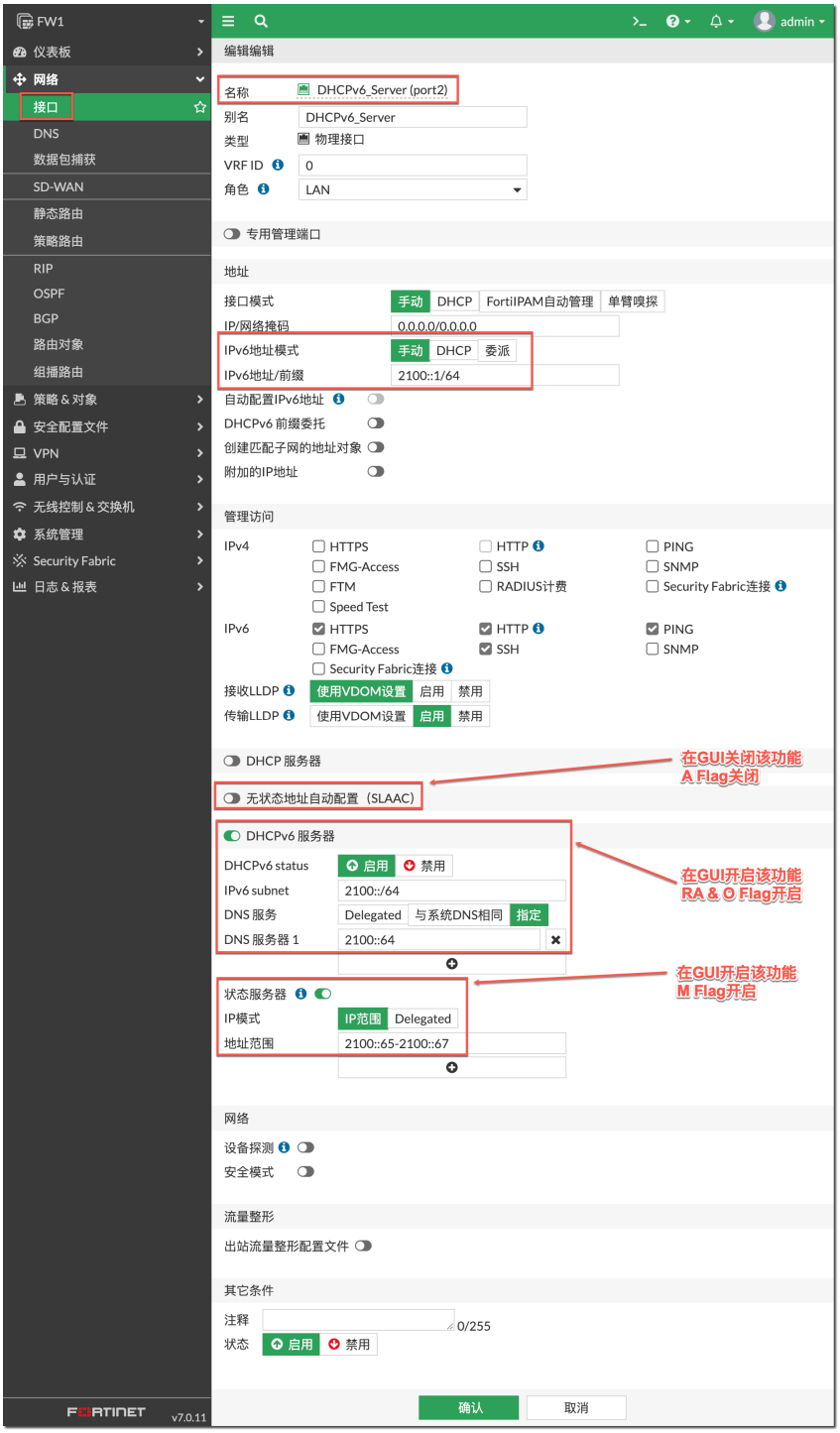
网络拓扑



FW1 和 FW2（或 PC）直连，FW1 作为 DHCPv6 服务器，FW2（或 PC）为客户端，FW2（或 PC）通过 DHCPv6 获取 IPv6 地址以及 DNS 等信息，不通过 RA 进行自动配置。

配置步骤

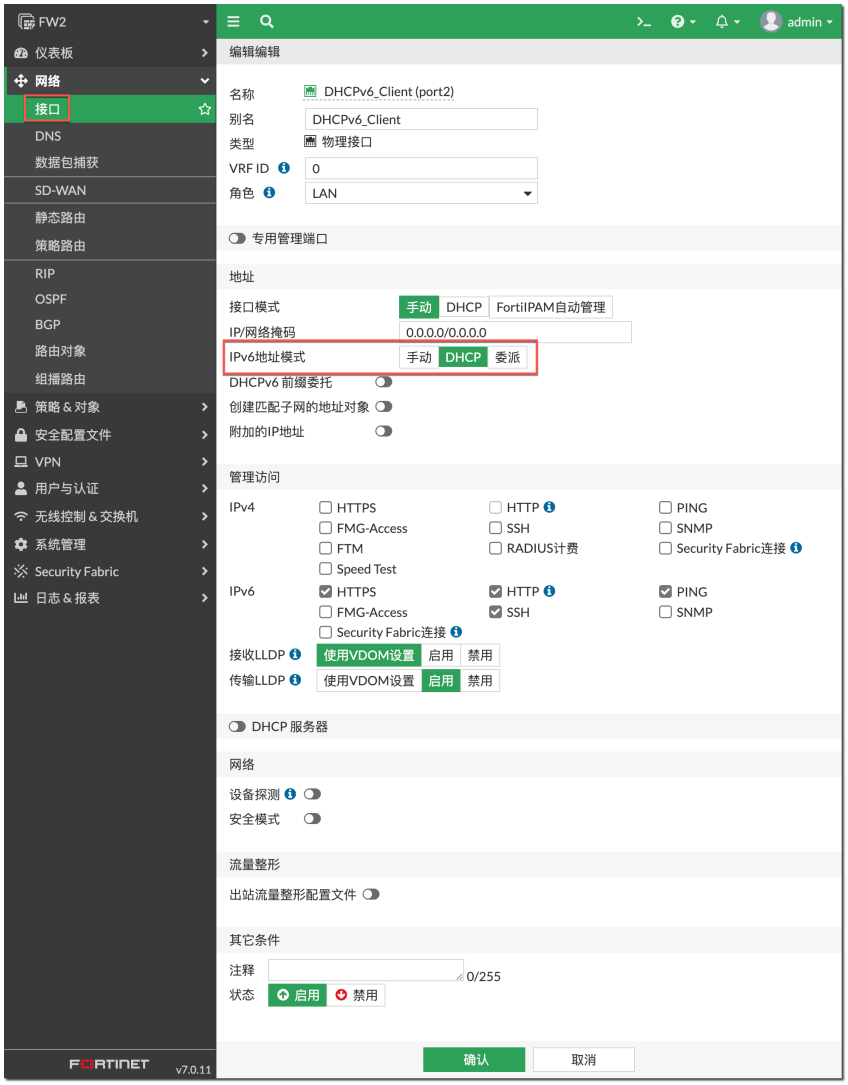
1. FW1 上配置 IPv6 地址 2100::1/64，并关闭“无状态地址自动配置（SLAAC）”，开启“DHCPv6 服务器”，配置 DNS 服务器，开启“状态服务器”，并配置分配 IP 范围。




```
config system interface
    edit "port2"
        config ipv6
            set ip6-address 2100::1/64
            set ip6-allowaccess ping https ssh http
            set ip6-send-adv enable    //开启RA
            set ip6-manage-flag enable    //开启M Flag
            set ip6-other-flag enable    //开启O Flag
        end
    next
en

config system dhcp6 server
    edit 1
        set subnet 2100::/64
        set interface "port2"
        config ip-range
```

```
            edit 1
                set start-ip 2100::65
                set end-ip 2100::67
            next
        end
        set dns-server1 2100::64
    next
end
```

2. FW2 上配置 port2 的“IPv6 地址模式”为“DHCP”。

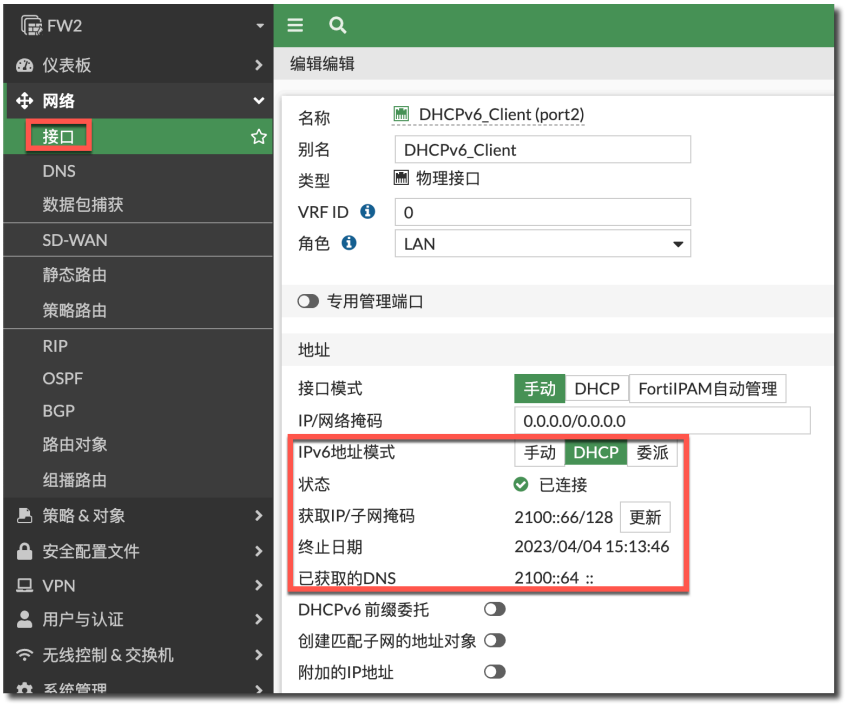



```
config system interface
    edit "port2"
        config ipv6
            set ip6-mode dhcp
        end
    next
end
```

结果验证

1. 在 FW2 上查看 DHCPv6 获取的 IPv6 地址和 DNS 服务器地址，并自动添加了 IPv6 网关为 FW1 的 port2 的 IPv6 本地链路地址。

> **重要**
>
> - FortiGate 作为 IPv6 客户端时，获取 IPv6 地址后，默认路由的添加需要通过 IPv6 路由 kernel 表（get router info6 kernel）中看到。
> - 客户端添加的默认路由的下一跳为服务器端接口的 IPv6 本地地址，而非手工配置的 IPv6 地址。

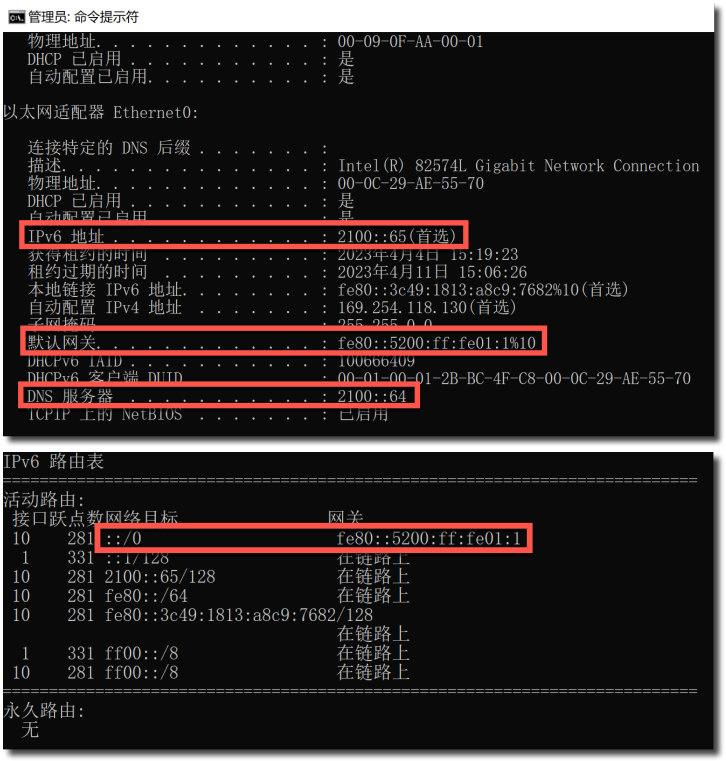



```
FW2 # config system interface
FW2 (interface) # edit port2

FW2 (port2) # get | grep ip6
    ip6-mode            : dhcp
    ip6-address         : 2100::66/128
    ip6-allowaccess     : ping https ssh http
    ip6-reachable-time  : 0
    ip6-retrans-time    : 0
    ip6-hop-limit       : 0
    vrip6_link_local    : ::
    ip6-dns-server-override: enable

FW2 (port2) # get | grep DNS
delegated-DNS1      : ::
delegated-DNS2      : ::
Acquired DNS1       : 2100::64
Acquired DNS2       : ::
Acquired DNS1       : 0.0.0.0
Acquired DNS2       : 0.0.0.0
```

```
FW2 # get router info6 kernel | grep gwy
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00450000 prio=1024 gwy:fe80::5200:ff:fe01:1 dev=4(port2) pmtu=1500

FW1 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag=P scope=0 prefix=64 addr=2100::1 preferred=4294967295 valid=4294967295 cstamp=1790 tstamp=1790
dev=4 devname=port2 flag=P scope=253 prefix=64 addr=fe80::5200:ff:fe01:1 preferred=4294967295 valid=4294967295
cstamp=1790 tstamp=1790
```

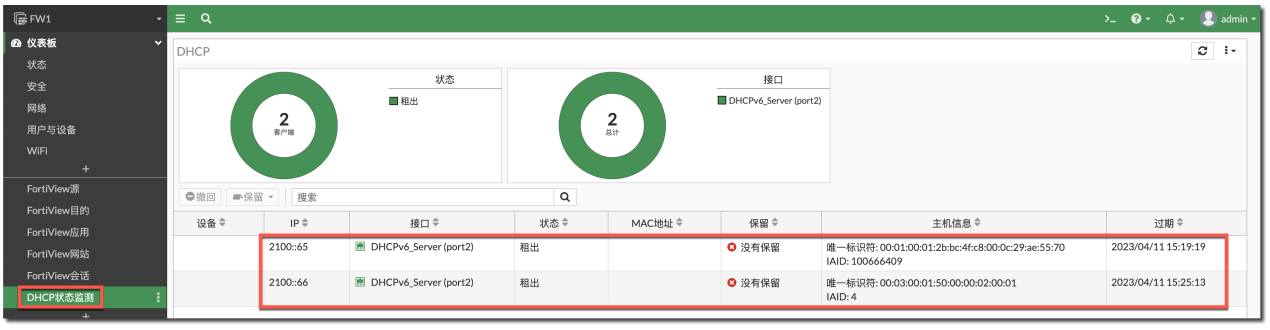
2. 使用 Windows PC 替换 FW2，来获取 IPv6 地址，可以看到 Windows 客户端获取到了 2100::/64 前缀的 IPv6 地址，并将 FW1 的 port2 的 IPv6 本地地址作为 IPv6 网关。





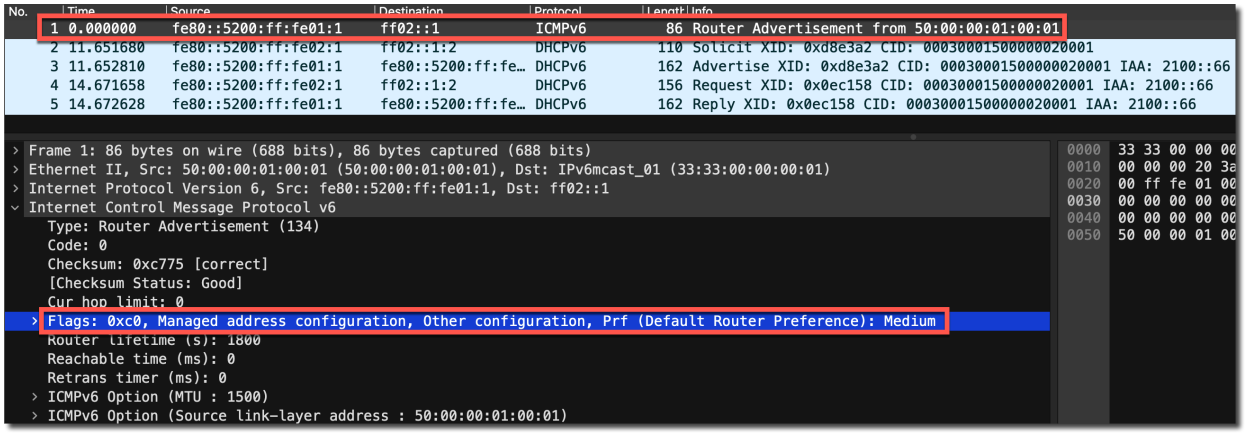

```
FW1 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag=P scope=0 prefix=64 addr=2100::1 preferred=4294967295 valid=4294967295 cstamp=1790 tstamp=1790
dev=4 devname=port2 flag=P scope=253 prefix=64 addr=fe80::5200:ff:fe01:1 preferred=4294967295 valid=4294967295
cstamp=1790 tstamp=1790
```

3. FW1 上查看 DHCPv6 地址分配情况。

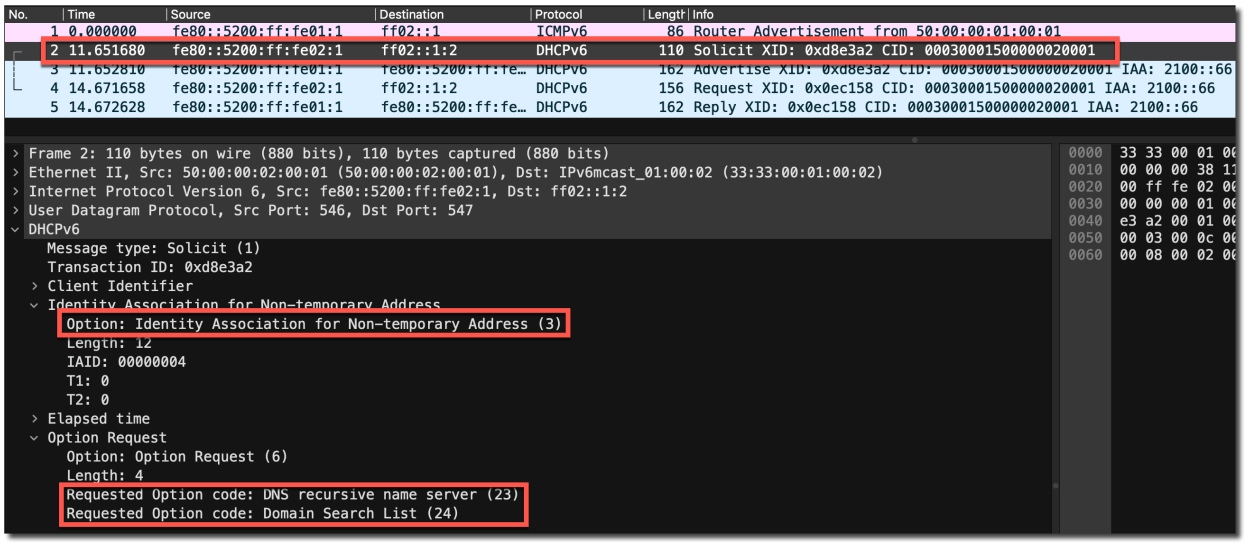


```
FW1 # exe dhcp6 lease-list port2
Interface       DUID                                        IAID       IP/Prefix                    Expiry
port2           00:01:00:01:2b:bc:4f:c8:00:0c:29:ae:55:70   100666409  2100::65                     Tue Apr 11
15:19:19 2023

port2           00:03:00:01:50:00:00:02:00:01               4          2100::66                     Tue Apr 11
15:25:13 2023
```

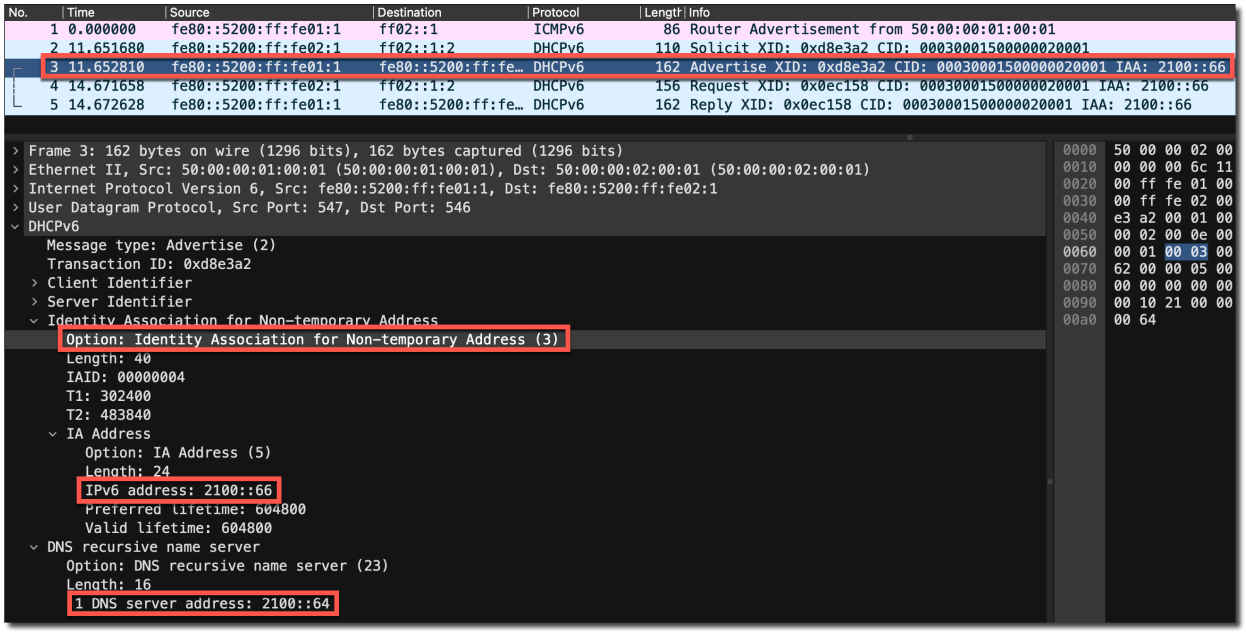
4. 抓包查看整个流程。
   - 查看 FW1 发出的 RA 报文，可以看到 RA 中开启了 M 和 O Flag，未开启 A Flag（没有 SLAAC prefix）。



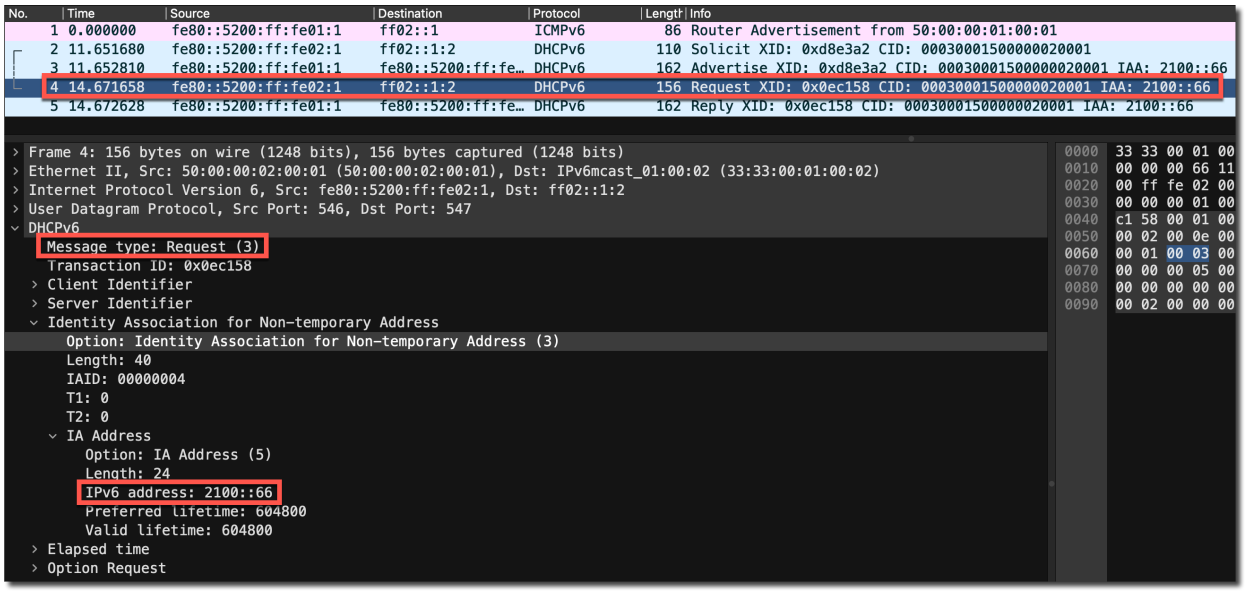
   - FW2 配置接口模式为 DHCPv6 后，发出 DHCPv6 Solicit 报文，请求非临时 IP 地址与 DNS 地址，目标 IP 地址（ff021:2 ）和 MAC 均为组播。



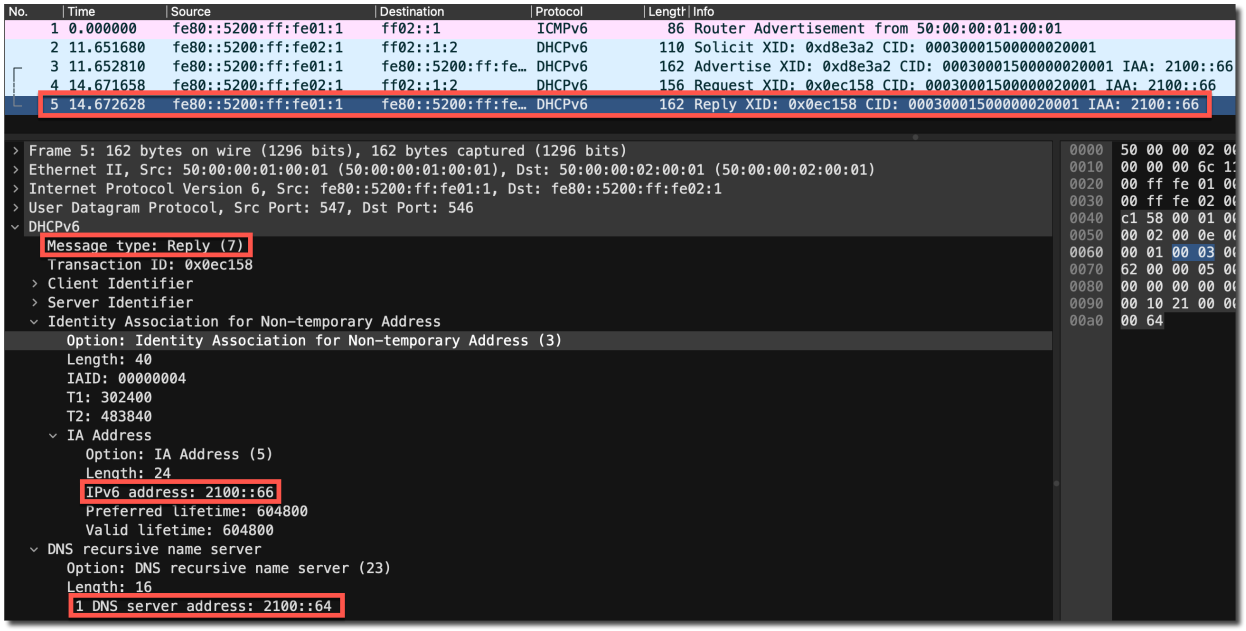
   - FW1 返回 DHCPv6 Advertise 报文，包含可用的 IPv6 地址以及 DNS 服务器地址，报文为单播，目标 IP 地址为 FW2 的本地 IPv6 地址。



- FW2 请求占用 FW1 提供的 IPv6 21001:2， DHCPv6 保留的组播地址 ff02::1:2 ）和 MAC 均为组播。



- FW1 发送 DHCPv6 Reply 报文，确认客户端可以使用该 IPv6 并携带了 DNS 服务器地址，目标 IP 为单播 FW2 的本地 IPv6 地址。



补充说明

1. 服务器端（FW1）A Flag 和 M Flag 同时开启，客户端（FW2）会生成几个地址?
   - 客户端（FW2）的 IPv6 地址配置为“Manual”+“Auto configure IPv6 addres”时，只会发出 RS 报文，然后 FW1 返回 RA 报文，客户端根据 RA 中的 prefix 自动配置地址。
   - 客户端（FW2）的 IPv6 地址配置为“DHCP”时，不会发出 RS 报文，只会发出 DHCPv6 报文，然后通过 DHCPv6 获取 IP 和 DNS 等信息，即使服务器端发出了携带 prefix 的 RA 报文，FW2 依然会通过 DHCPv6 获取 IP 地址和 DNS 等信息。
   - 也就是对于 FortiGate 作为 Client，Server 同时开启 M 和 A Flag，地址生成方式取决于客户端的 IPv6 地址模式。
   - 对于 PC Client 来说，当配置为 DHCP 获取地址时，仍然会发出 RS 报文，通过 Server 端发出的 RA 报文中的 Flag 来确定地址获取的方式，当 RA 中的 A Flag 和 M Flag 同时开启时，PC Client 会同时获取两种地址，一种从 RA 中的 prefix 自动生成，一种从 DHCPv6 Server 获取，如下图所示。



2. 接口下开启 DHCPv6 状态服务器后，请确保 DHCPv6 分配的 IPv6 地址与该接口的 IPv6 地址在同一网段，否则状态服务器不会响应任何 DHCPv6 请求。

DHCPv6 原理

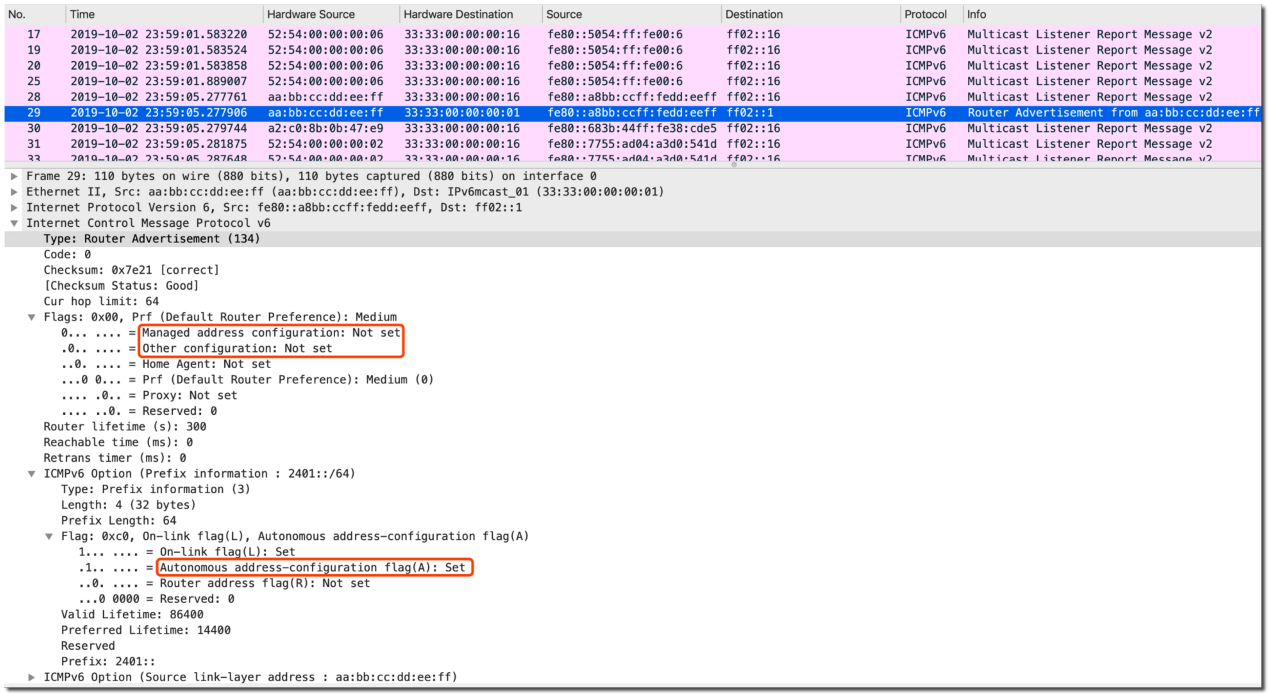
DHCPv6 获取方式

1. DHCPv6 客户端获取 Global Address 的方式有两种：
   - 无状态（Stateless）：根据路由通告报文 RA（Router Advertisement）包含的 prefix 前缀信息自动配置 IPv6 地址，组成方式是 Prefix + (EUI64 or 随机)，Stateless 也可以称为 SLAAC（Stateless Address Auto Configuration）。
   - 有状态（Stateful）：通过 DHCPv6 方式获得 IPv6 地址。
2. 上边的有状态又分为两种方式：
   - 有状态 DHCPv6（Stateful DHCPv6）：IPv6 地址、其他参数（如 DNS）均通过 DHCPv6 获取。
   - 无状态 DHCPv6（Stateless DHCPv6）：IPv6 地址依然通过路由通告 RA 方式生成，其他参数（如 DNS）通过 DHCPv6 获取。

> **重要**
>
> 适用场景：
>
> - 有状态、无状态仅针对于 IPv6 地址分配方式，并不包含其他参数。
> - 有状态：可控、可管理。在网络中存在一个 IP 地址管理者，它能够识别客户端，根据不同的客户端，分配对应的 IPv6 地址，客户端与服务端之间需要维护 IP 地址的租期及续约。目前实现这种效果的，就是 DHCPv6 协议，IP 地址管理者就是 DHCPv6 Server。
> - 无状态：不可控、难管理。在网络中只有网关，没有 IP 地址管理者。因此无人去识别客户端，每个客户端根据网关发送的相同的 RA 报文内容，自行配置 IPv6 地址。

IPv6 RA 中的 Flag

1. DHCPv6 客户端通过网络中路由器/防火墙发送的 IPv6 RA（Router Advertisement）报文中的 Flag 来控制获取 IPv6 和 DNS 等信息的方式。
   - Autonomous flag（自治 flag，简称 A flag）：表示是否配置无状态 IP。在一个 RA 报文中，可存在多个 prefix，比如 2401/64、2403::/64，每个 prefix 都可以独立配置 A flag。
     - 为 on 时（对应 bit 位为 1）：表示客户端应当在该 prefix 范围内自动生成 IPv6 地址（客户端通过 DAD_Duplicate Address Detection 自行保证地址可用），并配置子网路由条目、网关。
     - 为 off 时（对应 bit 位为 0）：表示客户端不应当在该 prefix 范围内自动生成 IPv6 地址，但是可以配置子网路由条目、网关。
   - Managed flag（简称 M flag）：表示是否配置有状态 IP。M flag 是 RA 报文的全局参数，一个 RA 报文只有一个 M flag。
     - 为 on 时（对应 bit 位为 1）：表示在 stateless 流程结束后开始 stateful 流程，也就是告诉客户端可以通过 DHCPv6 来获得 IPv6 地址和其他参数（如 DNS 列表）。
     - 为 off 时（对应 bit 位为 0）：表示不通过 DHCPv6 来获得 IPv6 地址。
   - Other flag（简称 O flag）：表示是否通过 DHCPv6 获得除 IP 以外的其他参数（如 DNS 列表）。O flag 也是 RA 报文中的全局参数，一个 RA 报文只有一个 O flag。注意：仅当 M flag 为 off 时，该参数才会被读取。
     - 为 on 时（对应 bit 位为 1）：当 M flag 为 on，或者 M flag 为 off 且至少有一个 A flag 为 on 时，将通过 DHCPv6 获得其他参数。
     - 为 off 时（对应 bit 位为 0）：当 M flag 为 on 时，依然将通过 DHCPv6 获得其他参数；当 M flag 也为 off 时，将不通过 DHCPv6 获得其他参数。
2. 如下图的 RA 报文， A flag 置为 1，M flag 和 O flag 均置为 0，表示要求客户端仅通过 RA 中的 prefix 自动配置 IPv6 地址，并配置子网路由条目、网关，其他参数不通过 DHCPv6 获取。

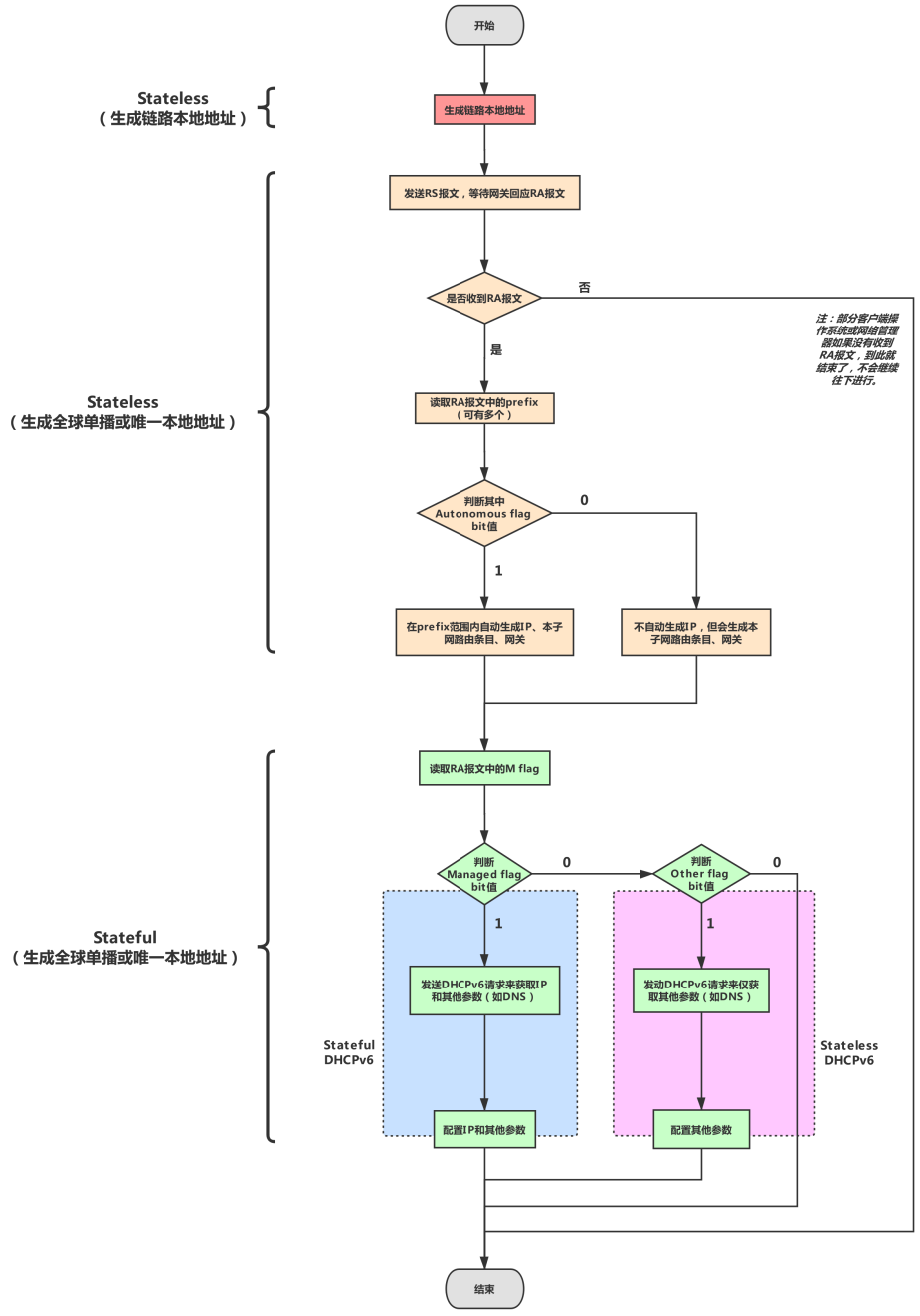


No.	Time	Hardware Source	Hardware Destination	Source	Destination	Protocol	Info
17	2019-10-02 23:59:01.583220	52:54:00:00:00:06	33:33:00:00:00:16	fe80::5054:ff:fe00:6	ff02::16	ICMPv6	Multicast Listener Report Message v2
19	2019-10-02 23:59:01.583524	52:54:00:00:00:06	33:33:00:00:00:16	fe80::5054:ff:fe00:6	ff02::16	ICMPv6	Multicast Listener Report Message v2
20	2019-10-02 23:59:01.583858	52:54:00:00:00:06	33:33:00:00:00:16	fe80::5054:ff:fe00:6	ff02::16	ICMPv6	Multicast Listener Report Message v2
25	2019-10-02 23:59:01.889007	52:54:00:00:00:06	33:33:00:00:00:16	fe80::5054:ff:fe00:6	ff02::16	ICMPv6	Multicast Listener Report Message v2
28	2019-10-02 23:59:05.277761	aa:bb:cc:dd:ee:ff	33:33:00:00:00:16	fe80::a8bb:ccff:fedd:eeff	ff02::16	ICMPv6	Multicast Listener Report Message v2
29	2019-10-02 23:59:05.277906	aa:bb:cc:dd:ee:ff	33:33:00:00:00:01	fe80::a8bb:ccff:fedd:eeff	ff02::1	ICMPv6	Router Advertisement from aa:bb:cc:dd:ee:ff
30	2019-10-02 23:59:05.279744	a2:c0:8b:0b:47:e9	33:33:00:00:00:16	fe80::683b:44ff:fe38:cde5	ff02::16	ICMPv6	Multicast Listener Report Message v2
31	2019-10-02 23:59:05.281875	52:54:00:00:00:02	33:33:00:00:00:16	fe80::7755:ad04:a3d0:541d	ff02::16	ICMPv6	Multicast Listener Report Message v2
33	2019-10-02 23:59:05.287648	52:54:00:00:00:02	33:33:00:00:00:16	fe80::7755:ad04:a3d0:541d	ff02::16	ICMPv6	Multicast Listener Report Message v2

```
▶ Frame 29: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
▶ Ethernet II, Src: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff), Dst: IPv6mcast_01 (33:33:00:00:00:01)
▶ Internet Protocol Version 6, Src: fe80::a8bb:ccff:fedd:eeff, Dst: ff02::1
▼ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x7e21 [correct]
    [Checksum Status: Good]
    Cur hop limit: 64
  ▼ Flags: 0x00, Prf (Default Router Preference): Medium
      0... .... = Managed address configuration: Not set
      .0.. .... = Other configuration: Not set
      ..0. .... = Home Agent: Not set
      ...0 0... = Prf (Default Router Preference): Medium (0)
      .... .0.. = Proxy: Not set
      .... ..0. = Reserved: 0
    Router lifetime (s): 300
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ▼ ICMPv6 Option (Prefix information : 2401::/64)
      Type: Prefix information (3)
      Length: 4 (32 bytes)
      Prefix Length: 64
    ▼ Flag: 0xc0, On-link flag(L), Autonomous address-configuration flag(A)
        1... .... = On-link flag(L): Set
        .1.. .... = Autonomous address-configuration flag(A): Set
        ..0. .... = Router address flag(R): Not set
        ...0 0000 = Reserved: 0
      Valid Lifetime: 86400
      Preferred Lifetime: 14400
      Reserved
      Prefix: 2401::
  ▶ ICMPv6 Option (Source link-layer address : aa:bb:cc:dd:ee:ff)
```

总结

DHCPv6 流程



- Stateless 自动配置“全球地址”（或“唯一本地地址”）。
- Stateful 自动配置“全球地址”（或“唯一本地地址”）和其他参数，其中 Stateful 阶段中存在 Stateful DHCPv6 或 Stateless DHCPv6。

重要

部分客户端操作系统或网络管理器当 Stateless 阶段没有收到 RA 报文后，就到此结束，不会走 Stateful 阶段，比如 CentOS 7、Ubuntu 17 的默认逻辑都是这样，而 Windows Server 就会继续走 Stateful 阶段。

Flag 组合对应客户端动作

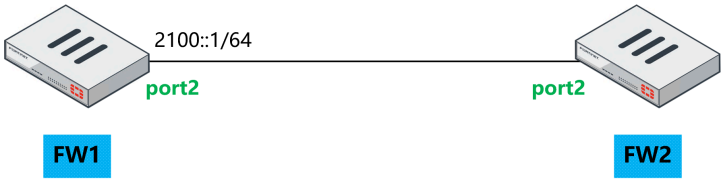
Managed Flag	Other Flag	Autonomous Flag	RA IP	RA 子网路由	RA gateway	DHCPv6 IP	DHCPv6 dns
M on	O on	A on	Y	Y	Y	Y	Y
		A off	N	Y	Y	Y	Y
	O off	A on	Y	Y	Y	Y	Y
		A off	N	Y	Y	Y	Y
M off	O on	A on	Y	Y	Y	N	Y
		A off	N	Y	Y	N	Y
	O off	A on	Y	Y	Y	N	N
		A off	N	Y	Y	N	N

无状态 + 有状态模式

网络需求

FortiGate 作为 DHCPv6 服务器，FortiGate 或 PC 为客户端，客户端从 FortiGate 的 RA 报文中自动配置 IPv6 地址，并使用 DHCPv6 从 FortiGate 获取 DNS 服务器地址。

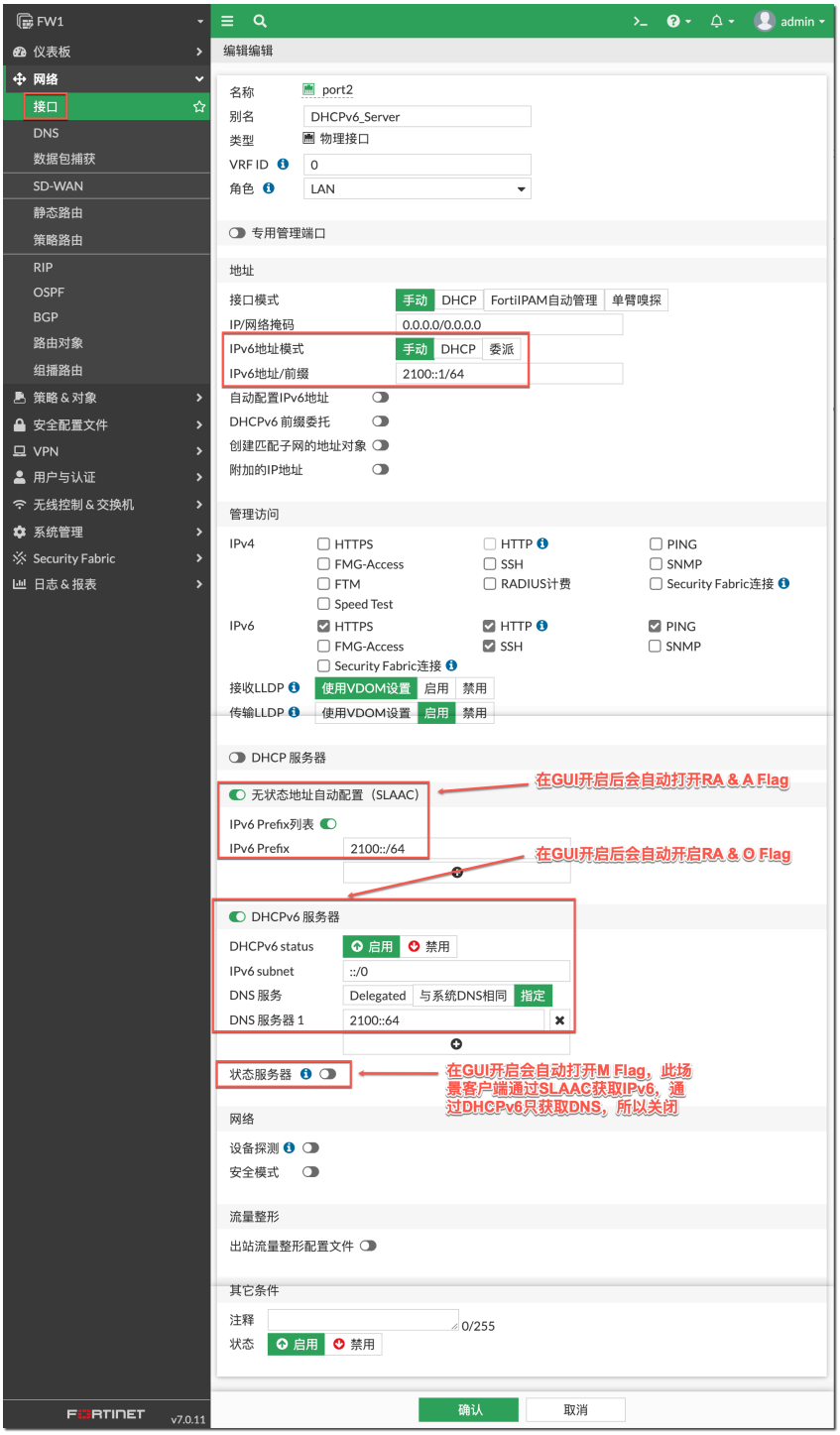
网络拓扑



FW1 和 FW2（或 PC）直连，FW1 作为 DHCPv6 服务器，FW2（或 PC）为客户端，FW2（或 PC）从 FW1 的 RA 报文中自动配置 IPv6 地址，并使用 DHCPv6 从 FW1 获取 DNS 服务器地址。

配置步骤

1. FW1 上配置 IPv6 地址 2100/64，开启“DHCPv6 服务器”，配置 DNS 服务器，不开启“状态服务器”。




```
config system interface
    edit "port2"
        config ipv6
            set ip6-address 2100::1/64
            set ip6-allowaccess ping https ssh http
            set ip6-send-adv enable     //开启RA
            set ip6-other-flag enable     //开启O Flag
            set ip6-manage-flag disable     //关闭M Flag
            config ip6-prefix-list
                edit 2100::/64
                    set autonomous-flag enable     //开启A Flag
                next
            end
        end
    next
end
```

```
config system dhcp6 server
    edit 1
        set subnet 2100::/64
        set interface "port2"
        set dns-server1 2100::64
    next
end
```

2. FW2 上配置 port2 的 IPv6 配置为“手动”模式，开启“自动配置 IPv6 地址”。



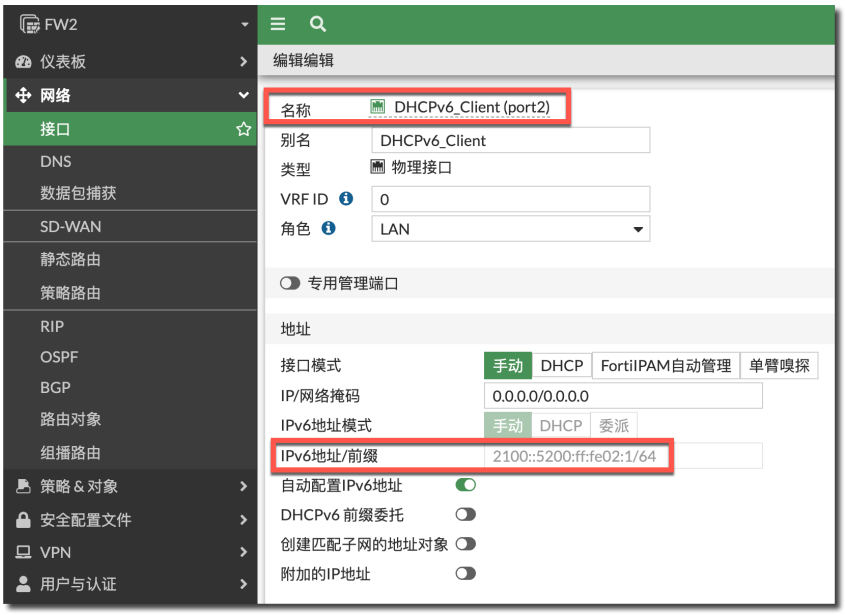
3. FW2 想要从 FW1 获取静态地址后，再通过 DHCPv6 获取 DNS 服务器信息，需要开启 dhcp6-Information-request，该功能只能在 CLI 下开启，用于在 IPv6 自动配置后再通过 DHCPv6 获取 DNS 服务器等信息，最终配置如下。

```
config system interface
    edit "port2"
        config ipv6
            set ip6-allowaccess ping https ssh http
            set dhcp6-information-request enable
            set autoconf enable
        end
    next
end
```

结果验证

1. 在 FW2 上查看自动配置的 IPv6 地址和 DNS 服务器地址，自动配置的 IPv6 地址根据前缀 + EUI 自动生成，DNS 通过 DHCPv6 获取，并自动添加了 IPv6 网关为 FW1 的 port2 的 IPv6 本地链路地址。

> **重要**
>
> - FortiGate 作为 IPv6 客户端时，获取 IPv6 地址后，默认路由的添加需要通过 IPv6 路由 kernel 表（get router info6 kernel）中看到。
> - 客户端添加的默认路由的下一跳为服务器端接口的 IPv6 本地地址，而非手工配置的 IPv6 地址。

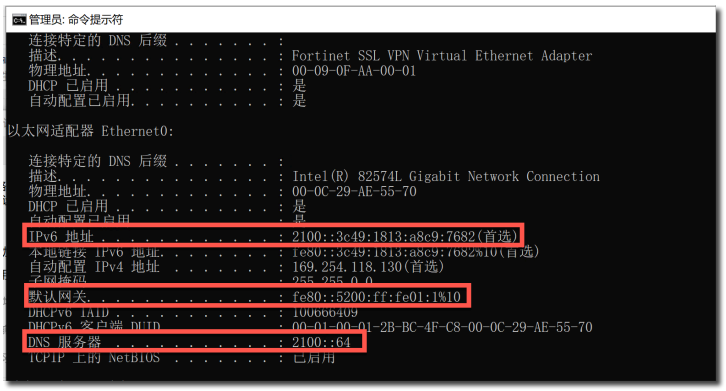



```
FW2 # config system interface
FW2 (interface) # edit port2
FW2 (port2) # get | grep ip6
    ip6-mode            : static
    ip6-address         : 2100::5200:ff:fe02:1/64
    ip6-allowaccess     : ping https ssh http
    ip6-reachable-time  : 0
    ip6-retrans-time    : 0
    ip6-hop-limit       : 0
    vrip6_link_local    : ::
    ip6-dns-server-override: enable
    ip6-send-adv        : disable

FW2 (port2) # get | grep DNS
Acquired DNS1       : 2100::64
Acquired DNS2       : ::
Acquired DNS1       : 0.0.0.0
Acquired DNS2       : 0.0.0.0
```

```
FW2 # get router info6 kernel | grep gwy
vf=0 tbl=254 type=01(unicast) protocol=2(kernel) flag=00450000 prio=1024 gwy:fe80::5200:ff:fe01:1 dev=4(port2) pmtu=1500

FW1 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag=P scope=0 prefix=64 addr=2100::1 preferred=4294967295 valid=4294967295 cstamp=1790 tstamp=1790
dev=4 devname=port2 flag=P scope=253 prefix=64 addr=fe80::5200:ff:fe01:1 preferred=4294967295 valid=4294967295
cstamp=1790 tstamp=1790
```

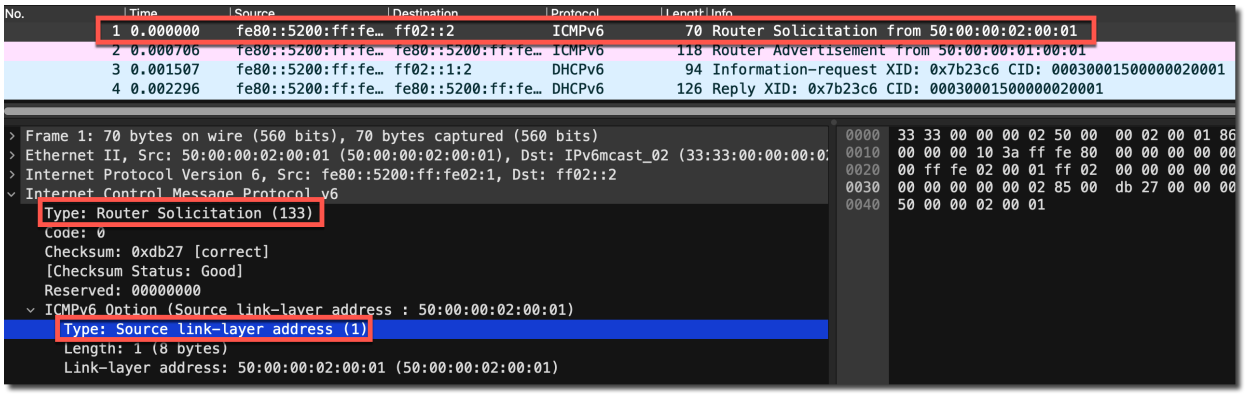
2. 使用 Windows PC 替换 FW2，来获取 IPv6 地址，可以看到 Windows 客户端获取到了 2100::/64 前缀的 IPv6 地址，并将 FW1 的 port2 的 IPv6 本地地址作为 IPv6 网关。



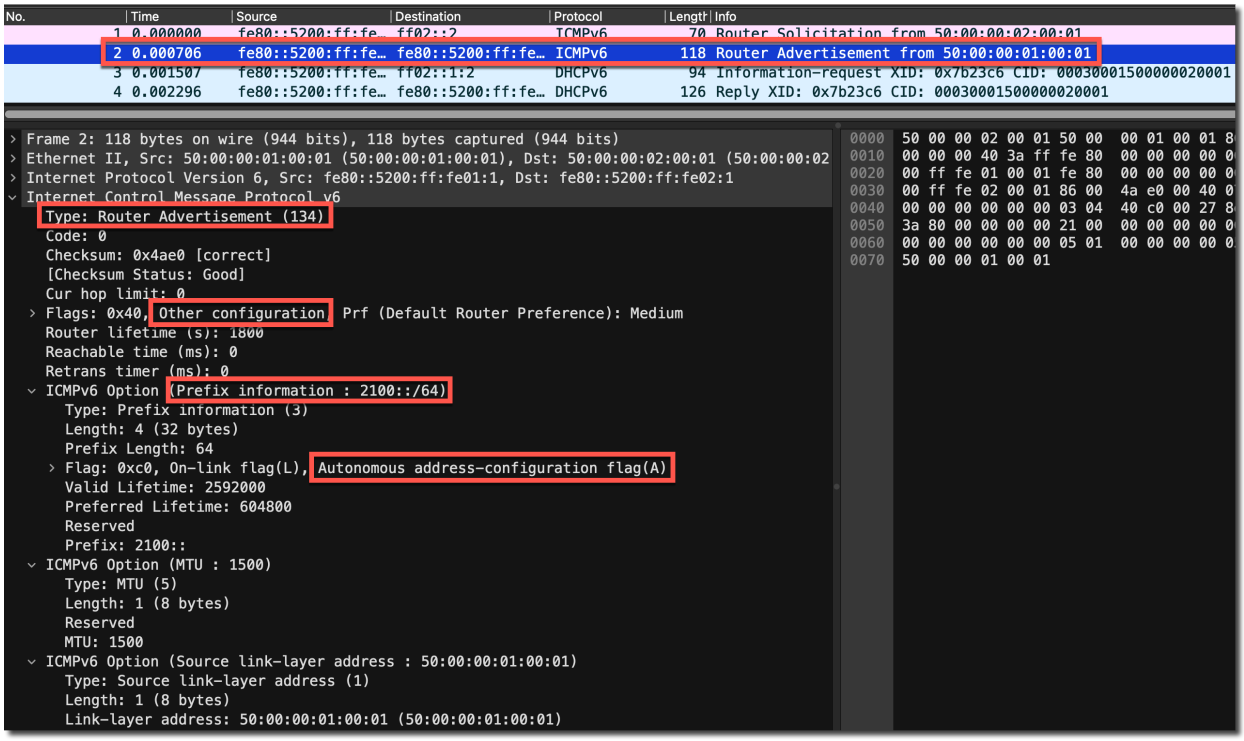



```
FW1 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag=P scope=0 prefix=64 addr=2100::1 preferred=4294967295 valid=4294967295 cstamp=1790 tstamp=1790
dev=4 devname=port2 flag=P scope=253 prefix=64 addr=fe80::5200:ff:fe01:1 preferred=4294967295 valid=4294967295
cstamp=1790 tstamp=1790
```

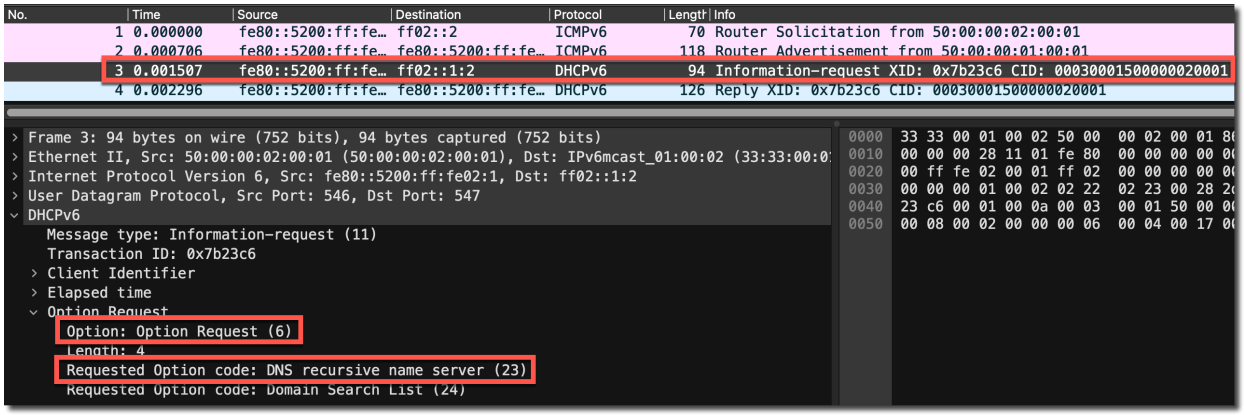
3. 抓包查看整个流程。
   - FW2 配置自动配置后，发出 RS 报文，请求 IP 地址配置，目标 IP（ff02::2，链路本地范围的所有路由器）和 MAC 均为组播。



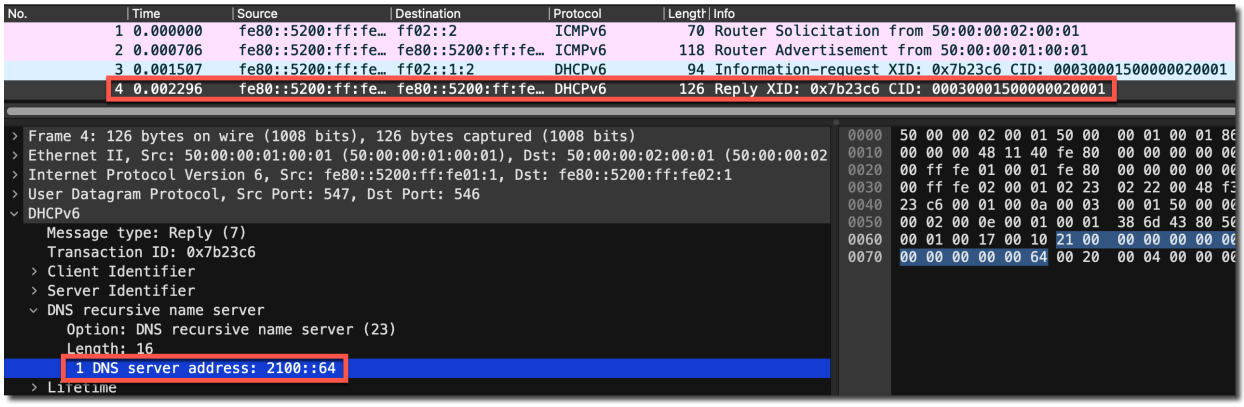
   - FW1 发送 RA 报文，通报自身 IPv6 prefix-list，可以看到前缀网段中标记了 A Flag，RA 中标记了 O Flag，该报文为单播。



- 由于开启了 DHCP6-Information-request，FW2 会继续通过 DHCPv6 请求 DNS 服务器等信息，该报文为组播，目标 IP 为 DHCPv6 保留的组播地址 ff02::1:2。



- FW1 返回 DHCPv6 Reply 给 FW2，携带 DNS 服务器地址，该报文为单播。

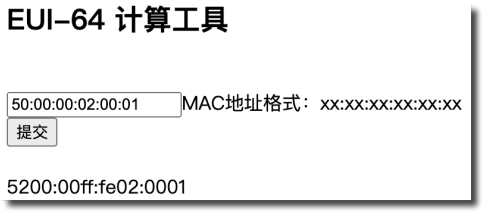


补充说明

1. 客户端获取到 Prefix 后，有两种方式生成 IPv6 地址：

- Prefix+EUI-64（FortiGate 使用）：自动将 48bit 的以太网 MAC 地址扩展成 64bit，再组合一个≤64 位的 IPv6 地址前缀，组成一个 IPv6 地址（Link-Local 地址也是通过此方法生成）。经过测试，如果 Prefix 不足 64 位，则用 0 补齐 Prefix 和 EUI-64 之间的内容（根据 EUI-64 计算器，您可自行查看上述步骤中 FW2 获取的 IPv6 地址是否符合 EUI64 标准）。

```
FW2 # diagnose netlink dstmac list port2
dev=port2 mac=50:00:00:02:00:01 policy6 rx_tcp_mss=0 tx_tcp_mss=0
...
```



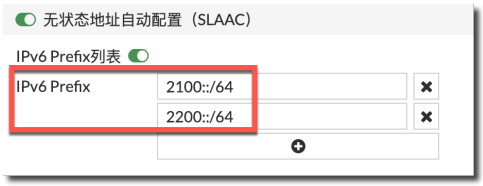
- 随机生成：RFC4941 引入了一种随机地址机制，他包含了用一组随机数字代替由 MAC 地址转换的接口 ID。该地址具有一定的生存周期，随着生存周期的结束，该地址会自动更换，该地址在用户对外发起连接的时候充当连接发起的 IPv6 地址，这一行为的目的在于保证主机在对外通信时候的匿名性。Windows、macOS 和 Linux 默认均为这种方式生成 IPv6，可以使用如下命令关闭随机地址。

```
Windows:
netsh interface IPV6 set global randomizeidentifiers=disabled    //禁用地址随机
netsh interface ipv6 set privacy state=disable    //禁用临时地址

linux下修改sysctl.conf:
net.ipv6.conf.all.use_tempaddr=0
net.ipv6.conf.eth0.use_tempaddr=0
net.ipv6.conf.default.use_tempaddr=0
或sysctl net.ipv6.conf.INTERFACE.use_tempaddr=0

macOS:
sysctl net.inet6.ip6.use_tempaddr=0
```

> **重要**
>
> EUI 的生成过程详见：https://ccie.lol/knowledge-base/ipv6-use-eui-64-convert-link-local-address/ ↗
>
> EUI-64 在线生成器：http://www.wlts.cc/eui64/eui64.php ↗，https://www.vultr.com/zh/resources/mac-converter/ ↗

2. RA 中包含多个 prefix 时，客户端是否会自动配置多个 IPv6?

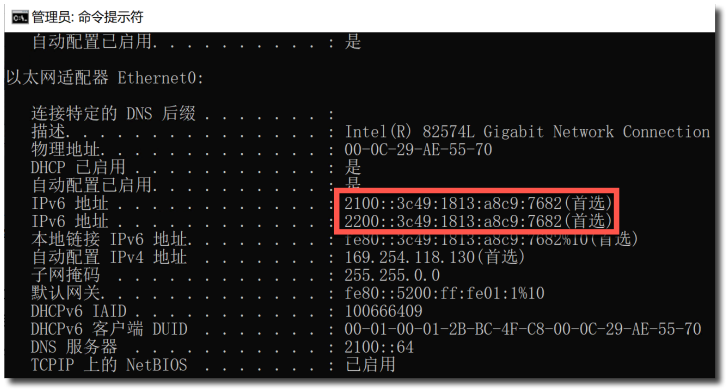
- 服务器端（FW1）的 SLAAC 配置两个 prefix（RA 报文中包含两个 prefix）。



- 客户端（FW2）自动配置 IPv6 地址，可以看到两个 prefix 均生成了 IPv6 地址。

```
FW2 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag= scope=0 prefix=64 addr=2200::5200:ff:fe02:1 preferred=604794 valid=2591994 cstamp=255528
tstamp=258728
dev=4 devname=port2 flag= scope=0 prefix=64 addr=2100::5200:ff:fe02:1 preferred=604794 valid=2591994 cstamp=139822
tstamp=258728
dev=4 devname=port2 flag=P scope=253 prefix=64 addr=fe80::5200:ff:fe02:1 preferred=4294967295 valid=4294967295
cstamp=2901 tstamp=2901
```

- PC 客户端也会生成 2 个 prefix 的 IPv6 地址。


管理员: 命令提示符
自动配置已启用. . . . . : 是
以太网适配器 Ethernet0:
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Intel(R) 82574L Gigabit Network Connection
物理地址. . . . . : 00-0C-29-AE-55-70
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
IPv6 地址 . . . . . : 2100::3c49:1813:a8c9:7682(首选)
IPv6 地址 . . . . . : 2200::3c49:1813:a8c9:7682(首选)
本地链接 IPv6 地址. . . . . : fe80::3c49:1813:a8c9:7682%10(首选)
自动配置 IPv4 地址 . . . . . : 169.254.118.130(首选)
子网掩码 . . . . . : 255.255.0.0
默认网关. . . . . : fe80::5200:ff:fe01:1%10
DHCPv6 IAID . . . . . : 100666409
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-2B-BC-4F-C8-00-0C-29-AE-55-70
DNS 服务器 . . . . . : 2100::64
TCPIP 上的 NetBIOS . . . . . : 已启用

DHCPv6-PD

网络需求

DHCPv6-PD（Prefix Delegation，前缀分配）是 DHCPv6 的扩展，传统的 DHCPv6 注重的是将地址等网络配置参数从服务器端传递给 IPv6 客户主机，而 DHCPv6-PD 注重的是前缀分配，即 PD 服务器端将一个子网范围的前缀地址段以及 DNS 等其他网络配置信息下发给 PD 客户端，这就意味 PD 服务器下发的是一个 “子网地址段”，而不是具体的可以直接供主机使用的 IPv6 地址。

举个例子，PD 服务器可以将 2022:da8::/60 前缀通过某个 IPv6 接口下发给 PD 客户端，同时一般会在 PD 客户端的上联设备上生成相应的路由。PD 客户端收到该前缀后，可以在本地 IPv6 接口上动态使用该前缀的/64 子网地址。

那么 DHCPv6-PD 到底用在什么场景中呢？这样做又有什么好处呢？DHCPv6-PD 在运营商场景中最为常见，当然也可以用在其他较大的分层设计的网络中。

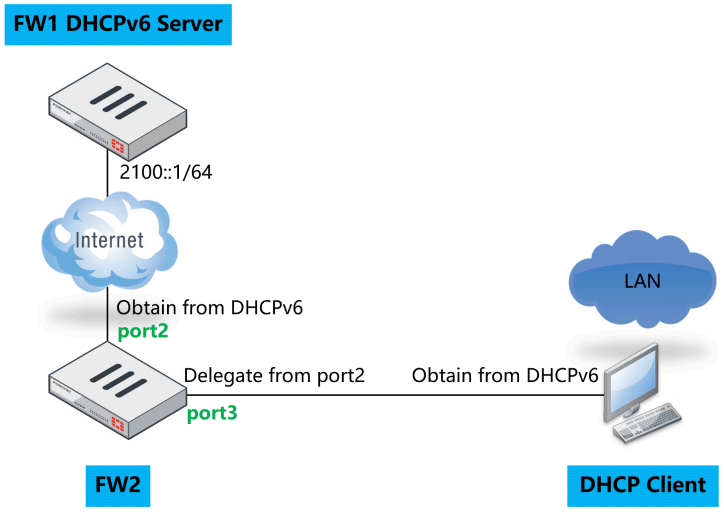
以运营商为例，DHCPv6-PD 服务器往往是供应商边界（Provider Edge，PE）路由器，而 DHCPv6-PD 客户端则是客户前置设备（Customer Premise Equipment，CPE）路由器。首先，PE 路由器作为 PD 服务器，会提前设置好一定范围的前缀分配列表，然后将这些前缀列表通过直连 CPE 的接口自动分配给各个 CPE 路由器。然后，CPE 路由器从上联接口收到 PE 路由器自动分配的前缀后，对此前缀做子网划分，这个子网划分一般是提前规划好并应用在各个下联接口的。待获取到前缀后，CPE 路由器自动就在各个下联接口下生成了 IPv6 地址，并可进一步通过自动地址配置技术下发给最终客户主机使用。

DHCPv6-PD 请求过程

- PD 客户端向 PD 服务器发送 DHCPv6 请求报文，此报文携带 IA_PD 选项，表明自己需要申请 IPv6 前缀。
- PD 服务器收到请求报文后，从自己的前缀列表池中取出可用的前缀，附带在 IA_PD 选项中，回复给 PD 客户端。
- 前缀分配到期后，PD 客户端重新向 PD 服务器发送 DHCPv6 请求报文，请求更新前缀。
- PD 服务器重新为 PD 客户端分配前缀（在原有前缀未被占用的情况下，一般就是续租）。
- PD 客户端需要释放前缀时，向 PD 服务器发送快速请求报文，以释放前缀。
- PD 服务器接收快速请求报文后，回收前缀，并对 PD 客户端的快速请求报文进行回应。

从上述通信过程可以看出，DHCPv6-PD 与 DHCPv6 的主要区别就是前者申请的是前缀，后者申请的是地址。

网络拓扑

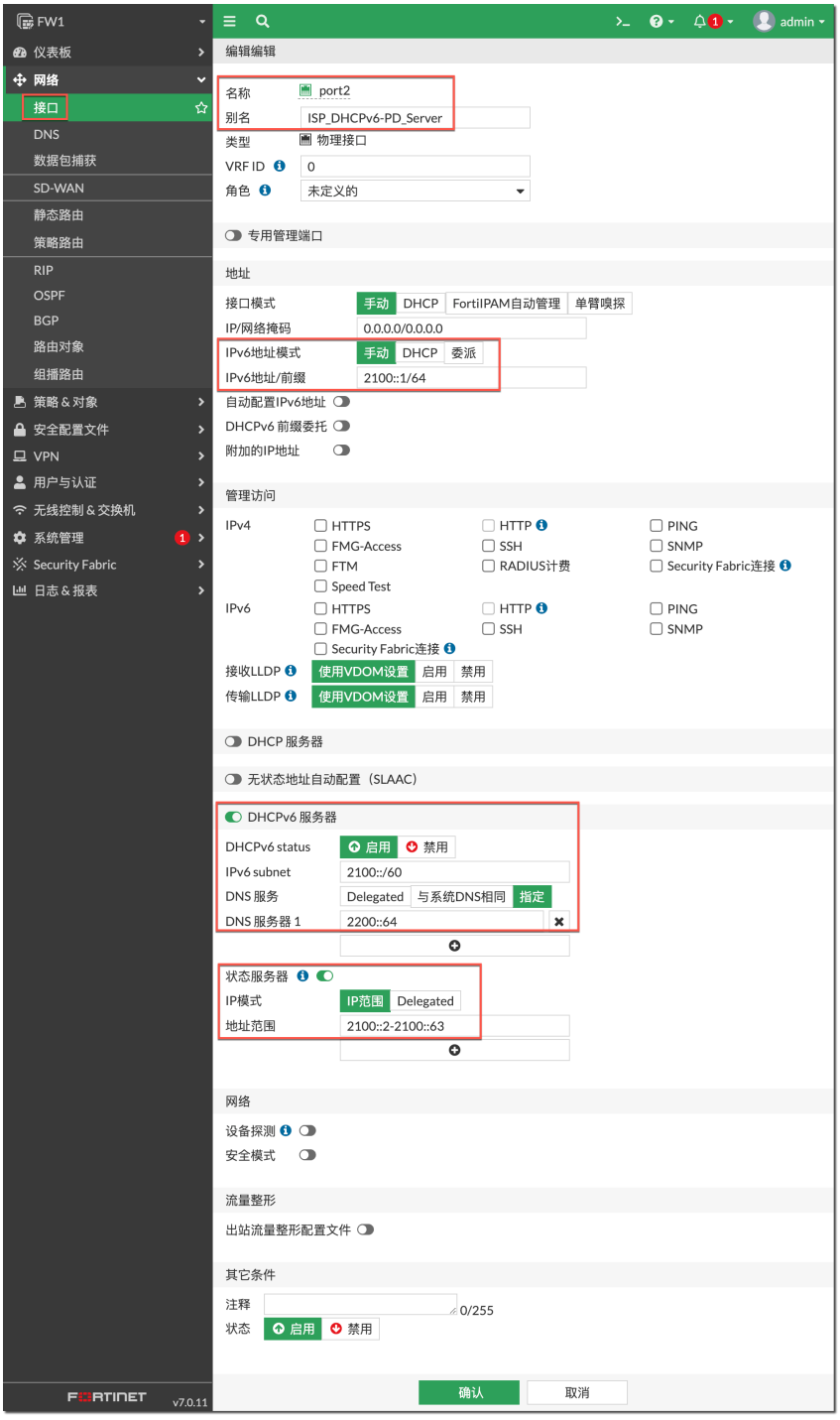


- FW1 的 port2 从 DHCPv6-PD 服务器获取 IPv6 地址和 Prefix，port3 通过 Delegate（继承）的方式从 port2 获取 IPv6 前缀，配合自身配置的 subnet 组成 port3 的 IPv6 地址。
- FW1 的 port3 开启 delegated DHCP Server，为内网 PC 分配 port2 获取的前缀中的 IPv6 地址。

配置步骤

FW1（模拟 ISP）

1. FW1 配置 DHCPv6 Server，模拟 Internet 运营商。




```bash
config system interface
    edit "port2"
        config ipv6
            set ip6-address 2100::1/64
            set ip6-send-adv enable
            set ip6-manage-flag enable
            set ip6-other-flag enable
        end
    next
end
```

2. FW1 开启 DHCPv6-PD 服务器，该功能只能在 CLI 开启。在 config system dhcp6 server 中，config prefix-range 为 PD 的配置，这里的配置就是可以给 FW2 分配的 Prefix 范围。

重要

这个范围不能超过 prefix-length，比如运营商的地址池为 60 位，分配的 prefix 为 64 位，则最多可以分配 $2^{(64-60)}$=16 个 prefix，否则下发配置报错。

```bash
config system dhcp6 server
    edit 1
        set subnet 2100::/60
        set interface "port2"
        config prefix-range
            edit 1
                set start-prefix 2100:0:0:3::
                set end-prefix 2100:0:0:4::
                set prefix-length 64
            next
        end
        config ip-range
            edit 1
                set start-ip 2100::2
                set end-ip 2100::63
            next
        end
        set dns-server1 2200::64
    next
end
```

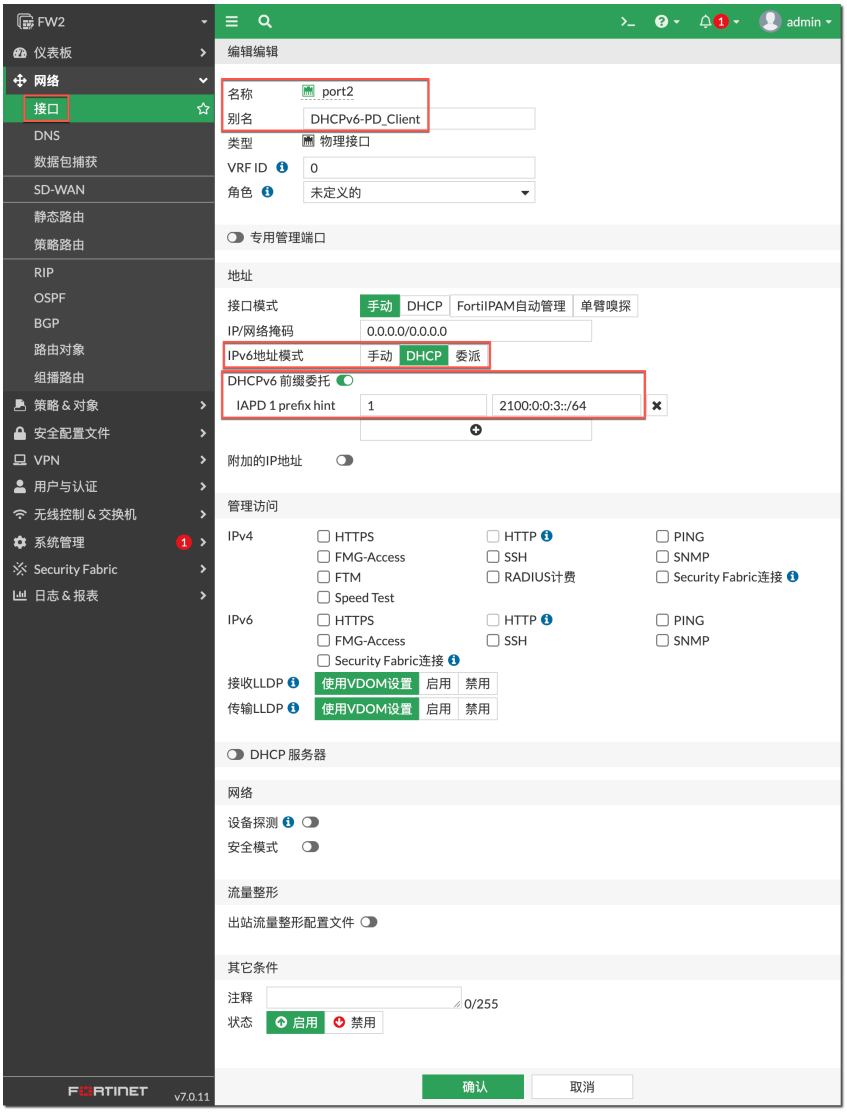
FW2（模拟用户边界设备）

1. FW2 上配置 port2 接口的 IPv6 为 DHCP 方式获取，并开启 DHCPv6 prefix delegation，填写 DHCPv6 prefix hint 为想要从 DHCPv6-PD 服务器获取的 prefix。

重要

这里的 DHCPv6 prefix hint ID（dhcp6-iapd-list ID）最好配置为 1，否则在下步 GUI 配置继承接口时，无法选择到上游接口（CLI 可以配置）。

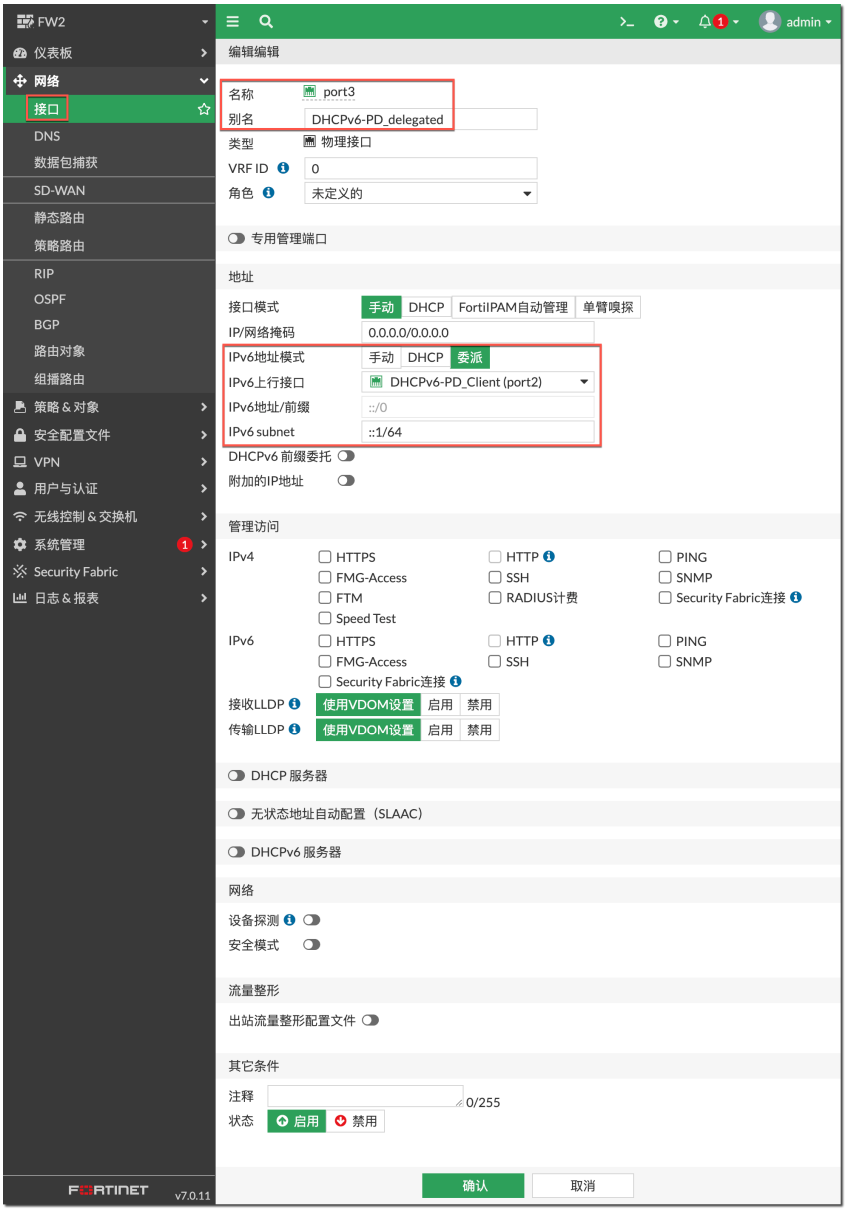
如果不知道需要向运营商申请哪一个 prefix，请填写为::/0，表示不指定请求的 prefix，让运营商选择分配的 prefix。




```bash
config system interface
    edit "port2"
        config ipv6
            set ip6-mode dhcp
            set dhcp6-prefix-delegation enable
            config dhcp6-iapd-list
                edit 1
                    set prefix-hint 2100:0:0:3::/64     //可以填写为::/0或不填写，表示不指定请求的prefix，让运营商选择分配的
prefix//
                next
            end
        end
    next
end
```

2. 在 FW2 的 port3 上，设置 IPv6 的 IP 地址获取模式为“委派”，配置“IPv6 上行接口”为 port2，并配置“IPv6 Subnet”，从 port2 继承 prefix 后，加上配置的 subnet 地址，就组成了 port3 的 IPv6 地址。

> **重要**
>
> 如果 GUI 中无法选择 IPv6 上行接口，可通过 CLI 配置。




```bash
config system interface
    edit "port3"
        config ipv6
            set ip6-mode delegated
            set ip6-delegated-prefix-iaid 1    //与步骤1中port2下配置的dhcp6-iapd序列号对应//
            set ip6-upstream-interface "port2"
            set ip6-subnet ::1/64    //获取prefix后填充的子网IPv6//
        end
    next
end
```

3. 那么 FW2 获取到这个 prefix 之后，除了可以分配给 port3 使用，还有什么作用呢？还可以通过自建 DHCPv6 服务器，将该 Prefix 中的 IPv6 分配给内网的 IPv6 客户端。

4. 在 FW2 上的 port3 开启 DHCPv6 服务器，配置 subnet 为 port3 接口网段，配置 DNS Server，开启 Stateful Server，配置分配 IP mode 为 Delegated，配置 upstream interface 为 port2，从 port2 获取的 prefix 分配 IP 地址给 port3，需要特别注意配置中引用步骤 1 中 port2 下配置的 IAPD。

> **重要**
>
> 该配置需要在 CLI 下配置，在 GUI 下配置会出现无法下发的情况。

```
config system dhcp6 server
    edit 1
        set subnet 2100:0:0:3::/64
        set interface "port3"
        set ip-mode delegated
        set upstream-interface "port2"
        set delegated-prefix-iaid 1    //重要步骤：配置中引用port2中配置的IAPD//
        set dns-server1 2200::64
    next
end
```
bash

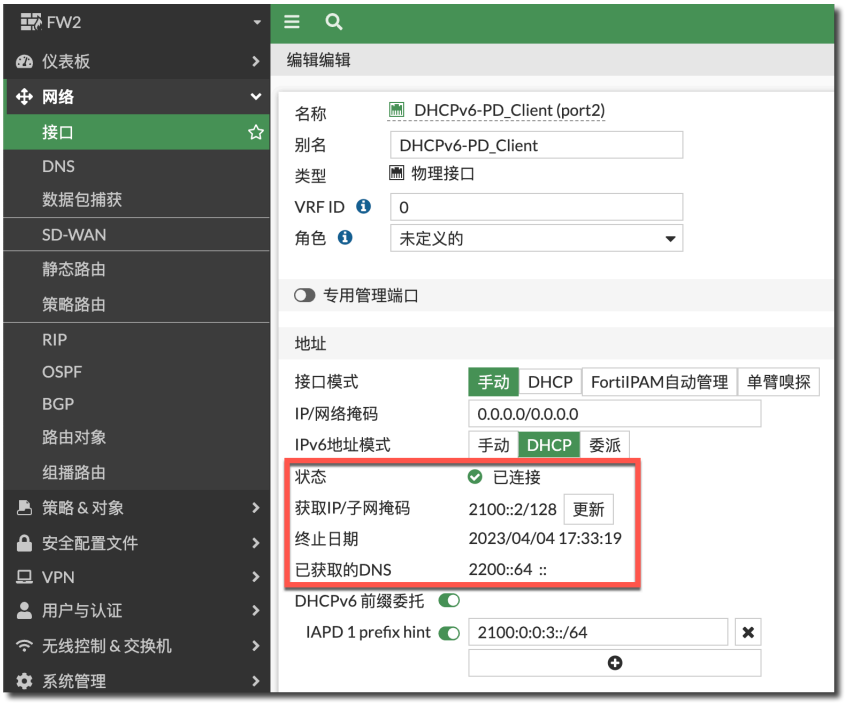
5. 再次在 FW2 查看 port3 的配置，可以看到 RA、M Flag、O Flag 均被开启。

```
config system interface
    edit "port3"
        config ipv6
            set ip6-mode delegated
            set ip6-delegated-prefix-iaid 1
            set ip6-send-adv enable
            set ip6-manage-flag enable
            set ip6-other-flag enable
            set ip6-upstream-interface "port2"
            set ip6-subnet ::1/64
        end
    next
end
```
bash

结果验证

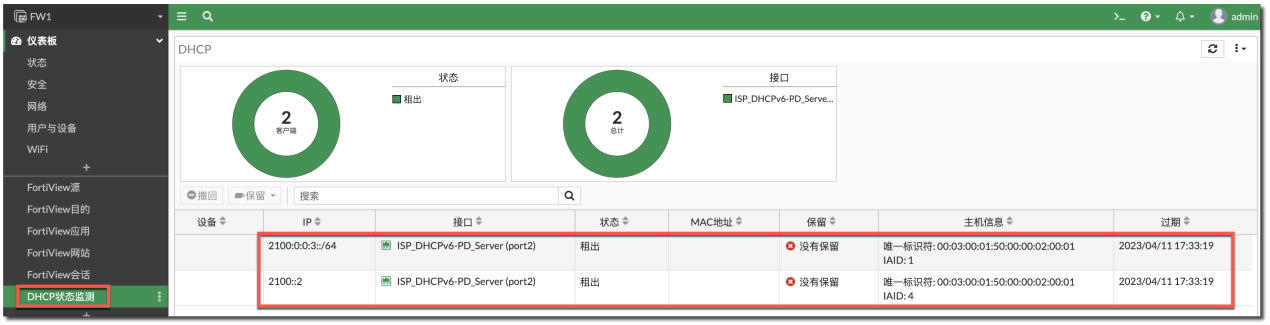
DHCPv6-PD

1. 在 FW2 上查看 port2 获取到的 IPv6 地址。




```
FW2 # diagnose ipv6 address list | grep port2
dev=4 devname=port2 flag=P scope=0 prefix=128 addr=2100::2 preferred=4294967295 valid=4294967295 cstamp=352820
tstamp=352820
...
```
bash

2. 在 FW1 上查看分配的 DHCPv6 客户端，可以看到 FW1 向 FW2 分别分配了一个 IPv6 地址和一个 IPv6 prefix，prefix IAID 为 FW2 的 port2 配置的 IAID 1，DUID（DHCPv6 Unique Identifier）为 FW2 的 DUID。



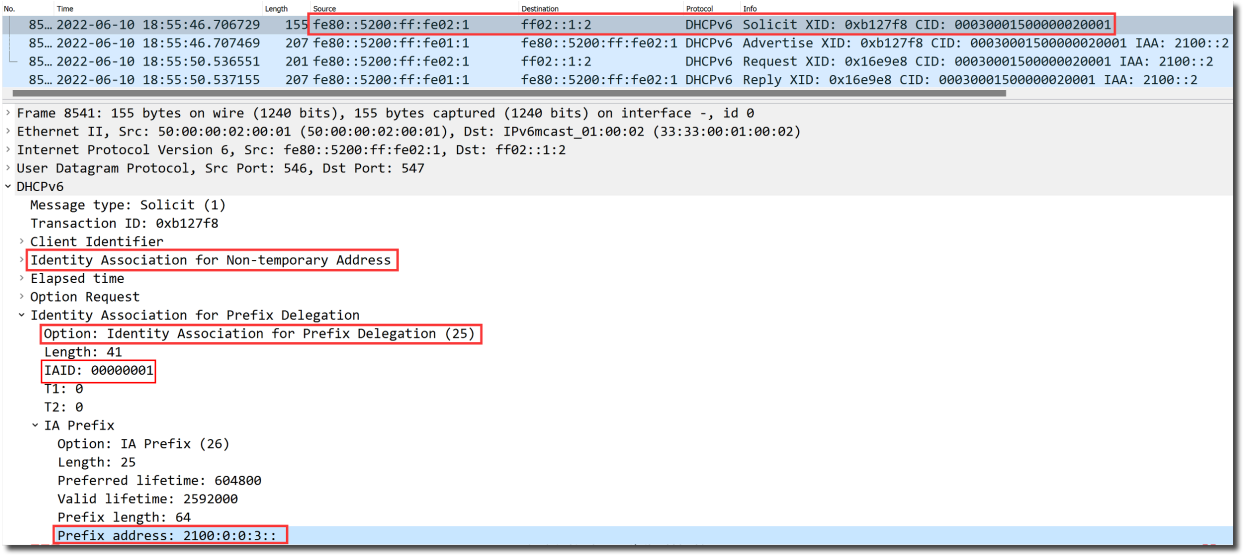

```bash
FW1 # exe dhcp6 lease-list
Interface       DUID                                IAID       IP/Prefix                Expiry
port2           00:03:00:01:50:00:00:02:00:01       1          2100:0:0:3::/64          Tue Apr 11
17:33:19 2023

port2           00:03:00:01:50:00:00:02:00:01       4          2100::2                  Tue Apr 11
17:33:19 2023
```

3. 查看 FW1 IAPD（Identity Association for Prefix Delegation）的 debug 信息（`diagnose debug application dhcp6s -1`）。

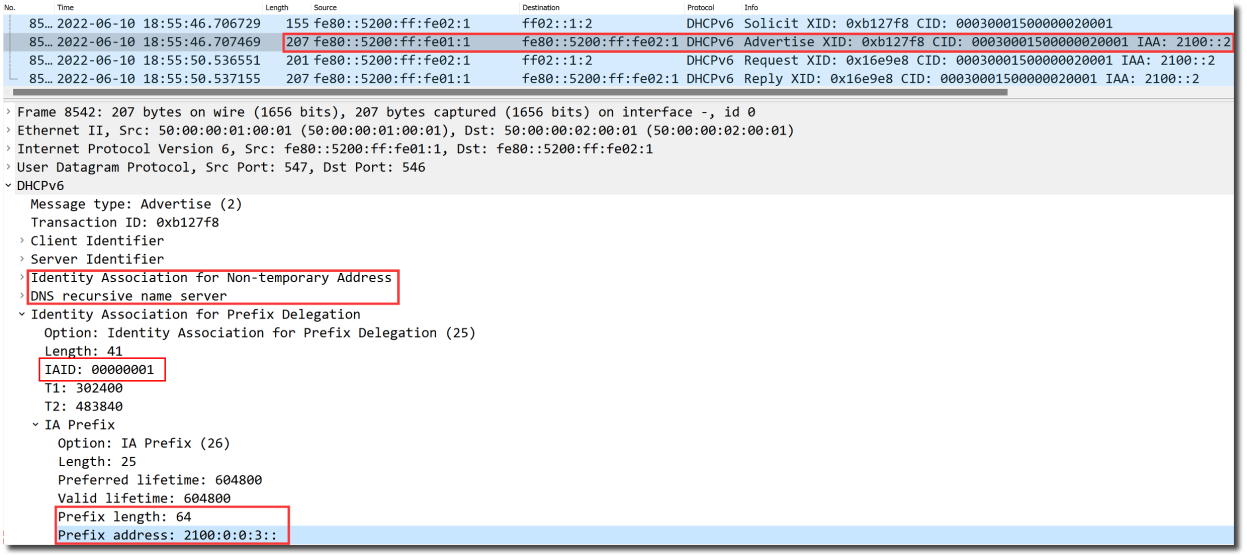
```
[debug]dhcp6_get_options() get DHCP option IA_PD, len 41
[debug]    IA_PD: ID=1, T1=0, T2=0
[debug]copyin_option() get DHCP option IA_PD prefix, len 25
[debug]copyin_option()    IA_PD prefix: 2100:0:0:3::/64 pltime=604800 vltime=604800
[debug]react_request() found a host configuration named 00:03:00:01:50:00:00:02:00:01
```

4. 抓包查看以上获取过程。

   - 客户端 DHCPv6 请求中既包含 IPv6 地址请求，也包含 prefix 的请求（IAPD），请求的 Prefix 与配置的一致，这里看到的 IAID 的值与 dhcp6-iapd-list 的 ID 一致，如果配置了多个 dhcp6-iapd-list，则会携带多个 IAPD 请求。

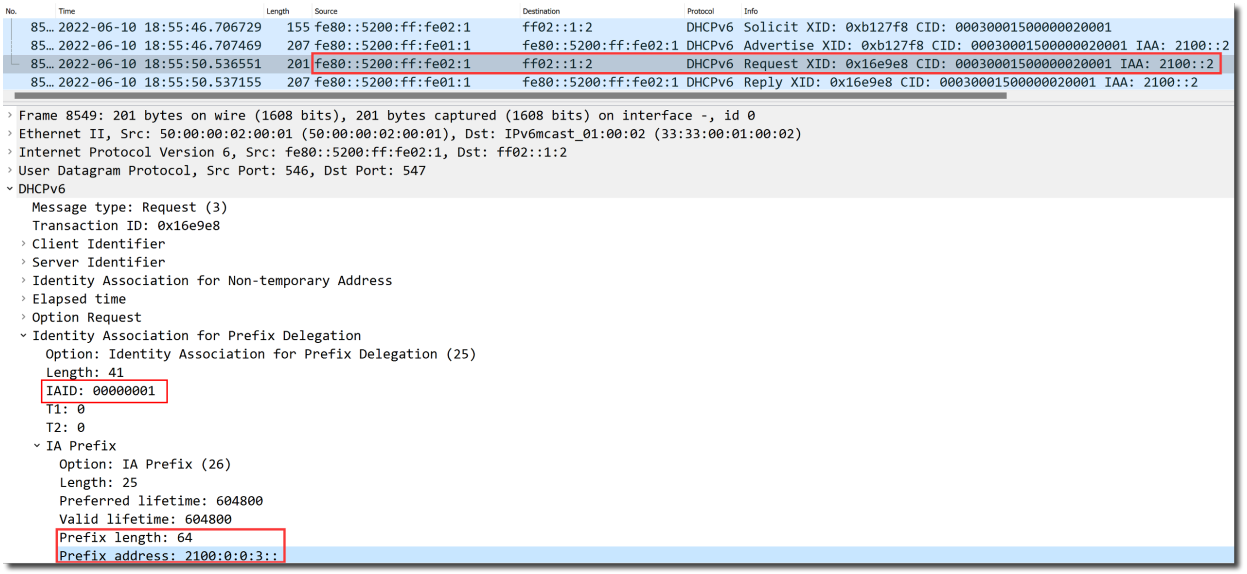


   - 服务器通过 DHCPv6 Advertise 告知可用的 IPv6 地址，并确认该 prefix 可用，返回的 prefix IAID 也为 1。



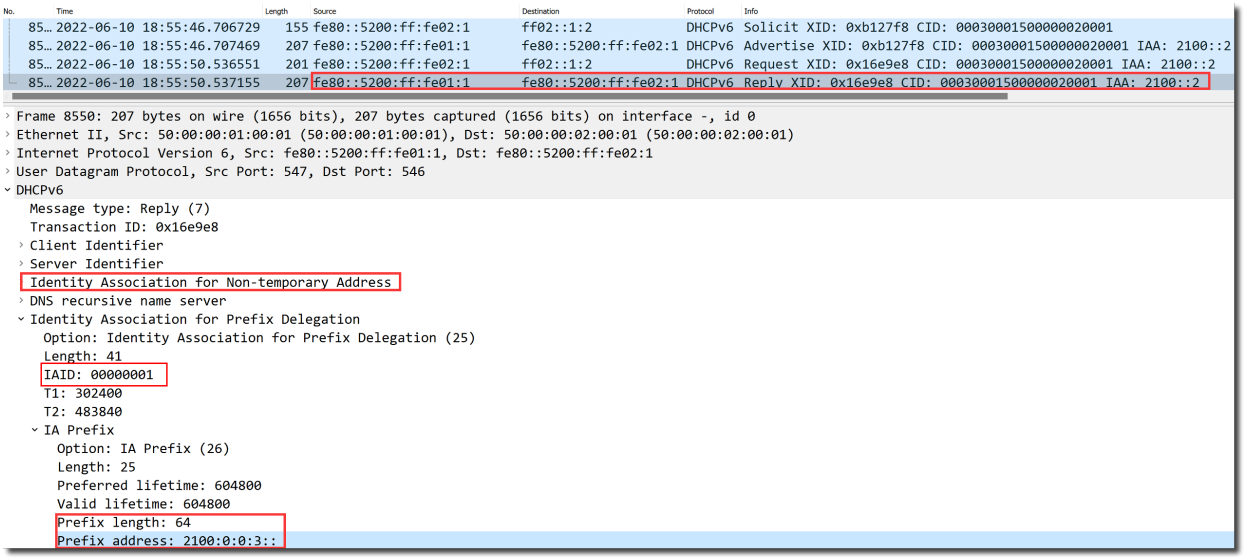
```
No.   Time                        Length Source                Destination           Protocol Info
85…   2022-06-10 18:55:46.706729  155    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Solicit XID: 0xb127f8 CID: 0003000150000020001
85…   2022-06-10 18:55:46.707469  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Advertise XID: 0xb127f8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.536551  201    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Request XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.537155  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Reply XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2

> Frame 8542: 207 bytes on wire (1656 bits), 207 bytes captured (1656 bits) on interface -, id 0
> Ethernet II, Src: 50:00:00:01:00:01 (50:00:00:01:00:01), Dst: 50:00:00:02:00:01 (50:00:00:02:00:01)
> Internet Protocol Version 6, Src: fe80::5200:ff:fe01:1, Dst: fe80::5200:ff:fe02:1
> User Datagram Protocol, Src Port: 547, Dst Port: 546
v DHCPv6
    Message type: Advertise (2)
    Transaction ID: 0xb127f8
  > Client Identifier
  > Server Identifier
  > Identity Association for Non-temporary Address
  > DNS recursive name server
  v Identity Association for Prefix Delegation
      Option: Identity Association for Prefix Delegation (25)
      Length: 41
      IAID: 00000001
      T1: 302400
      T2: 483840
    v IA Prefix
        Option: IA Prefix (26)
        Length: 25
        Preferred lifetime: 604800
        Valid lifetime: 604800
        Prefix length: 64
        Prefix address: 2100:0:0:3::
```

- 客户端请求确认占用该 IPv6 地址和 prefix，IAID 与 dhcp6-iapd-list 的 ID 一致，如果配置了多个 dhcp6-iapd-list，则会携带多个 IAPD 请求。



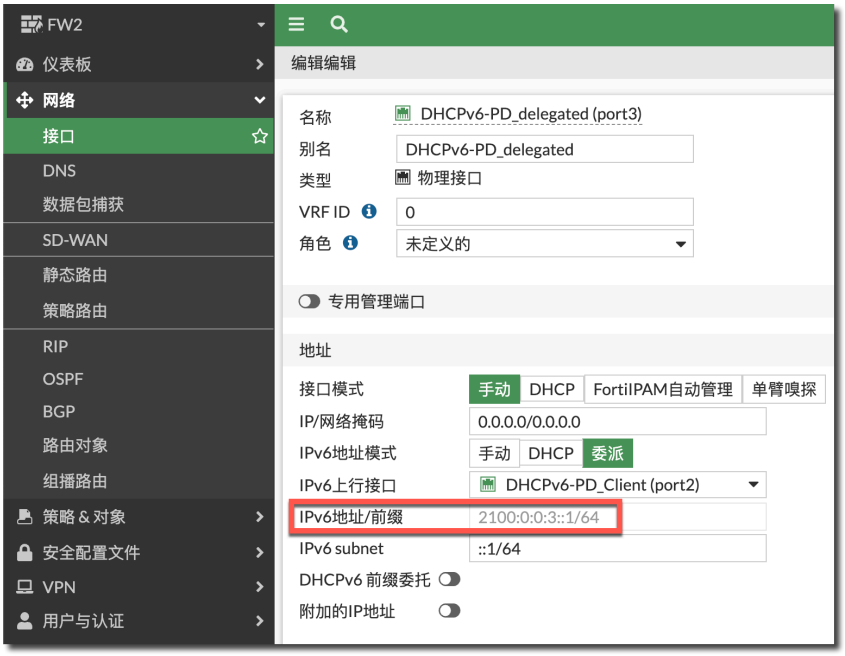
```
No.   Time                        Length Source                Destination           Protocol Info
85…   2022-06-10 18:55:46.706729  155    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Solicit XID: 0xb127f8 CID: 0003000150000020001
85…   2022-06-10 18:55:46.707469  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Advertise XID: 0xb127f8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.536551  201    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Request XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.537155  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Reply XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2

> Frame 8549: 201 bytes on wire (1608 bits), 201 bytes captured (1608 bits) on interface -, id 0
> Ethernet II, Src: 50:00:00:02:00:01 (50:00:00:02:00:01), Dst: IPv6mcast_01:00:02 (33:33:00:01:00:02)
> Internet Protocol Version 6, Src: fe80::5200:ff:fe02:1, Dst: ff02::1:2
> User Datagram Protocol, Src Port: 546, Dst Port: 547
v DHCPv6
    Message type: Request (3)
    Transaction ID: 0x16e9e8
  > Client Identifier
  > Server Identifier
  > Identity Association for Non-temporary Address
  > Elapsed time
  > Option Request
  v Identity Association for Prefix Delegation
      Option: Identity Association for Prefix Delegation (25)
      Length: 41
      IAID: 00000001
      T1: 0
      T2: 0
    v IA Prefix
        Option: IA Prefix (26)
        Length: 25
        Preferred lifetime: 604800
        Valid lifetime: 604800
        Prefix length: 64
        Prefix address: 2100:0:0:3::
```

- 服务器返回 Reply 确认分配 IPv6 与 Prefix，原报文见附件。



```
No.   Time                        Length Source                Destination           Protocol Info
85…   2022-06-10 18:55:46.706729  155    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Solicit XID: 0xb127f8 CID: 0003000150000020001
85…   2022-06-10 18:55:46.707469  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Advertise XID: 0xb127f8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.536551  201    fe80::5200:ff:fe02:1  ff02::1:2             DHCPv6   Request XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2
85…   2022-06-10 18:55:50.537155  207    fe80::5200:ff:fe01:1  fe80::5200:ff:fe02:1  DHCPv6   Reply XID: 0x16e9e8 CID: 0003000150000020001 IAA: 2100::2

> Frame 8550: 207 bytes on wire (1656 bits), 207 bytes captured (1656 bits) on interface -, id 0
> Ethernet II, Src: 50:00:00:01:00:01 (50:00:00:01:00:01), Dst: 50:00:00:02:00:01 (50:00:00:02:00:01)
> Internet Protocol Version 6, Src: fe80::5200:ff:fe01:1, Dst: fe80::5200:ff:fe02:1
> User Datagram Protocol, Src Port: 547, Dst Port: 546
v DHCPv6
    Message type: Reply (7)
    Transaction ID: 0x16e9e8
  > Client Identifier
  > Server Identifier
  > Identity Association for Non-temporary Address
  > DNS recursive name server
  v Identity Association for Prefix Delegation
      Option: Identity Association for Prefix Delegation (25)
      Length: 41
      IAID: 00000001
      T1: 302400
      T2: 483840
    v IA Prefix
        Option: IA Prefix (26)
        Length: 25
        Preferred lifetime: 604800
        Valid lifetime: 604800
        Prefix length: 64
        Prefix address: 2100:0:0:3::
```

5. 查看 FW2 上的 port3 继承获取的 IPv6 地址，可以看到 port3 已经从 port2 的 DHCP - PD Client 获取到了前缀，加上自己的 subnet Address，组成了自己的 IPv6 地址。



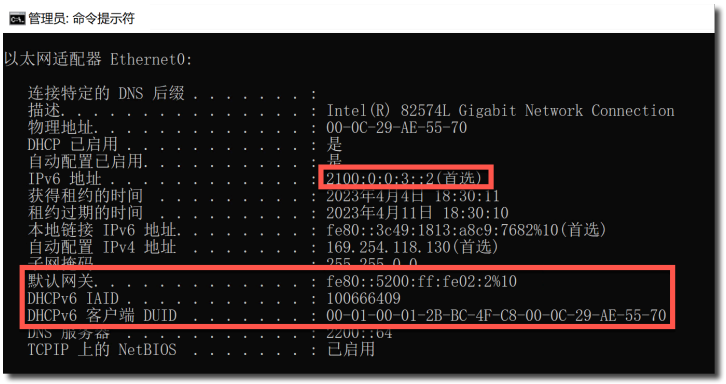

```bash
FW2 # diagnose ipv6 address list | grep port3
dev=5 devname=port3 flag=P scope=0 prefix=64 addr=2100:0:0:3::1 preferred=4294967295 valid=4294967295 cstamp=485929
tstamp=485929
...
```

DHCPv6 Client

1. 配置 PC 客户端的 IPv6 获取方式为自动获取。



2. 查看获取的 IPv6 地址、默认网关与 DNS 服务器，可以看到 PC 成功获取到了 upstream delegated prefix 中的 IPv6 地址。



3. 在 FW2 上查看 DHCP Client 列表，DUID 为 PC 客户端的 DUID，IANA 的 IAID 与客户端申请时的 IAID 一致（100666409）。

```bash
FW2 # exe dhcp6 lease-list
Interface       DUID                                    IAID       IP/Prefix                  Expiry
port3           00:01:00:01:2b:bc:4f:c8:00:0c:29:ae:55:70     100666409  2100:0:0:3::2              Tue Apr 11
18:30:05 2023
```

4. FW2 上 IAPD（Identity Association for Prefix Delegation）的 debug 信息（`diagnose debug application dhcp6s -1`）。

```bash
...
[debug]dhcp6_get_options() get DHCP option option request, len 6
[debug]   requested option: vendor specific info
[debug]   requested option: DNS
[debug]   requested option: domain search list
[debug]react_confirm() found a host configuration named 00:01:00:01:2b:bc:4f:c8:00:0c:29:ae:55:70
...
```

策略与对象

单线路上网配置

- 静态地址线路上网配置
- 单条 ADSL 线路上网配置
- DHCP 线路上网配置

多线路上网配置

- 双线路同运营商上网配置
- 双线路不同运营商上网配置
- 双线路 ADSL

虚拟 IP 映射

- 端口映射应原理
- 地址映射（一对一 IP 映射）
- 端口映射（一对多端口映射）
- 多线路端口映射
- VIP 映射的主机访问 Internet
- 内网主机通过公网 IP 访问 VIP
- VIP46 映射
- VIP64 映射

同一接口进出的三层转发

导出策略为 CSV 或 JSON

配置 Session TTL

会话 dirty 机制

iprope 列表

策略常见问题

- 流量到达 FortiGate 后无法转发
- VIP 匹配顺序
- VIP 流量无法匹配防火墙策略
- NAT 端口耗尽
- 地址池 ARP 刷新
- 策略页面加载慢

本地流量策略

- Local-in 策略
- Local-out 流量控制
- Preferred Source

ISDB

- ISDB 简介
- 预定义 ISDB
- 自定义 ISDB
- 扩展 ISDB
- IP 信誉级别
- 基于地理位置的 ISDB
- ISDB 端口附加配置
- 查询 ISDB/地理位置信息
- 按需模式的 ISDB

DoS 策略

- 什么是 DoS
- FortiGate DoS 策略工作原理
- DoS 策略使用

FortiGate DoS 策略配置　FortiGate DoS 策略调整

- FortiGate DoS 策略排错

流量整形

- 流量整形介绍
- 流量整形策略

  穿越流量　本地流量
- 流量整形配置文件
- 流量整形器

  共享流量整形器　每 IP 整形器
- NP7 的流量整形

  NP7 流量整形介绍　Policing 模式　Shaping 模式

转换 DNS 解析结果

同一接口进出的三层转发

本文介绍了 FortiGate 在数据包进出为同一逻辑接口（三层转发）时的处理机制。

数据包在穿越 FortiGate 做三层转发时，如果 FortiGate 查找路由表后确定从 **同一接口 in**的数据包也应通过**同一接口 out（不更改 VLAN tag、不做 NAT、不进行 IPSec 封装/解封装）**，FortiGate 转发行为与如下内容有关。

> **提示**
>
> 以下内容基于接口配置下 `set icmp-send-redirect enable`（默认配置）为前提。

默认行为

通过如下 CLI 命令修改同一接口进出的三层转发场景下的处理机制。

```bash
config system global
    set allow-traffic-redirect {enable | disable}
    set ipv6-allow-traffic-redirect {enable | disable}
end
```

FortiOS 7.4.10/7.6.5/8.0.0 之前版本

默认为 enable，数据包会直接从同一出接口发出，**不需要相应防火墙策略放通，也不会产生会话和流量日志**。

FortiOS 7.4.10/7.6.5/8.0.0 及之后版本

默认为 disable。必须配置明确的防火墙策略放通，流量才会被转发。

> **警告**
>
> - 在升级到 FortiOS 7.4.10/7.6.5/8.0.0 及之后版本后，无论之前 `allow-traffic-redirect` / `ipv6-allow-traffic-redirect` 的配置是 enable 还是 disable，都会强制将其设置为 disable。
> - 升级到 FortiOS 7.4.10/7.6.5/8.0.0 前请务必注意该变化产生的业务风险，提前配置相关防火墙策略。

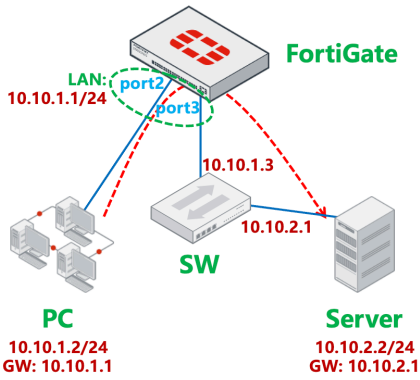
> **提示**
>
> - 对于公有云 FortiGate-VM 的单臂（one-arm）流量，始终建议将 `allow-traffic-redirect` / `ipv6-allow-traffic-redirect` 状态设置为 `disable`，并创建防火墙策略进行控制。
> - 对于需要做 NAT、日志审计的流量，建议将 `allow-traffic-redirect` / `ipv6-allow-traffic-redirect` 状态设置为 `disable`，并创建对应的防火墙策略。

不同流量场景的效果

对于不同类型的流量/版本组合，`allow-traffic-redirect` 的开启和关闭可能导致不同的效果。

源 IP 与转发接口在同一网段

网络拓扑



- FortiGate 的 port2 和 port3 组成一个交换接口 LAN。
- PC 源 IP `10.10.1.2` 与 FortiGate 的转发接口 LAN 的 IP `10.10.1.1` 在同一网段。PC 的网关为 FortiGate 的 LAN 接口 IP，Server 的网关为 SW 10.10.2.1。
- PC 访问 Server，流量从 FortiGate 的 LAN 接口进，查路由表从 LAN 接口出（三层转发），下一跳为 SW 的地址 10.10.1.3。

转发行为

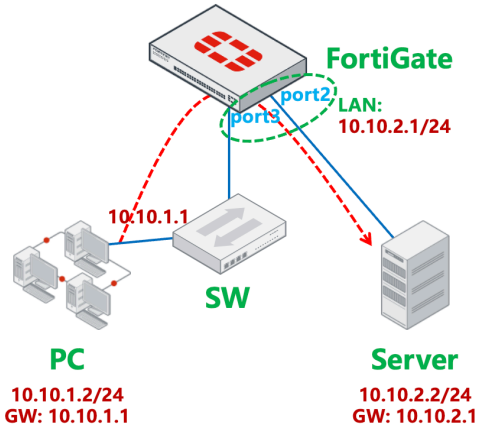
- `set allow-traffic-redirect enable`：数据包就会按照路由（或策略路由）转发，直接从同一出接口发出，**不需要**相应防火墙策略放通，流量穿过 FortiGate 时，**不会创建会话，也不会产生流量日志。**
- `set allow-traffic-redirect disable`： 必须有相应的防火墙策略放通。如果没有策略放通，流量将匹配隐式拒绝策略被丢弃。流量穿过 FortiGate 时，**会创建会话，可以产生流量日志，可以根据策略配置做 NAT。**以上图拓扑为例，需要配置的放通策略如下所示。

> **重要**
>
> 对于 TCP 流量，必须在策略中关闭 `anti-replay`，这样 TCP 流量就不会作为重放流量被丢弃。

```bash
config firewall policy
    edit 0
        set srcintf "LAN"
        set dstintf "LAN"
        set action accept
        set srcaddr "10.10.1.2/32"
        set dstaddr "10.10.2.2/32"
        set schedule "always"
        set service "ALL"
        set anti-replay disable     //针对TCP流量必须关闭//
    next
end
```

源 IP 与转发接口在不同网段

网络拓扑



- FortiGate 的 port2 和 port3 组成一个交换接口 LAN。
- PC 源 IP `10.10.1.2` 与 FortiGate 的转发接口 LAN 的 IP `10.10.2.1` 不在同一网段。PC 的网关为 SW 10.10.1.1，Server 的网关为 FortiGate 的 LAN 接口 10.10.2.1。

- PC 访问 Server，流量从 FortiGate 的 LAN 接口进，查路由表从 LAN 接口出（三层转发），下一跳为 SW 的地址 10.10.1.3。

转发行为

- **FortiOS 版本 < 7.0.16/7.2.11/7.4.4：** 无论 `allow-traffic-redirect` 启用还是禁用，流量都必须与防火墙策略进行匹配。**会创建会话，也可以产生流量日志。**
- **FortiOS 版本 ⩾ 7.0.16/7.2.11/7.4.4：**
  - `set allow-traffic-redirect enable`：数据包就会按照路由（或策略路由）转发，直接从同一出接口发出，**不需要**相应防火墙策略放通，流量穿过 FortiGate 时，**不会创建会话，也不会产生流量日志。**
  - `set allow-traffic-redirect disable`：必须有相应的防火墙策略放通。如果没有策略放通，流量将匹配隐式拒绝策略被丢弃。流量穿过 FortiGate 时，**会创建会话，可以产生流量日志，可以根据策略配置做 NAT。**以上图拓扑为例，需要配置的放通策略如下所示。

    > **重要**
    >
    > 对于 TCP 流量，必须在策略中关闭 `anti-replay`，这样 TCP 流量就不会作为重放流量被丢弃。

    ```bash
    config firewall policy
        edit 0
            set srcintf "LAN"
            set dstintf "LAN"
            set action accept
            set srcaddr "10.10.1.2/32"
            set dstaddr "10.10.2.2/32"
            set schedule "always"
            set service "ALL"
            set anti-replay disable    //针对TCP流量必须关闭//
        next
    end
    ```

导出策略为 CSV 或 JSON

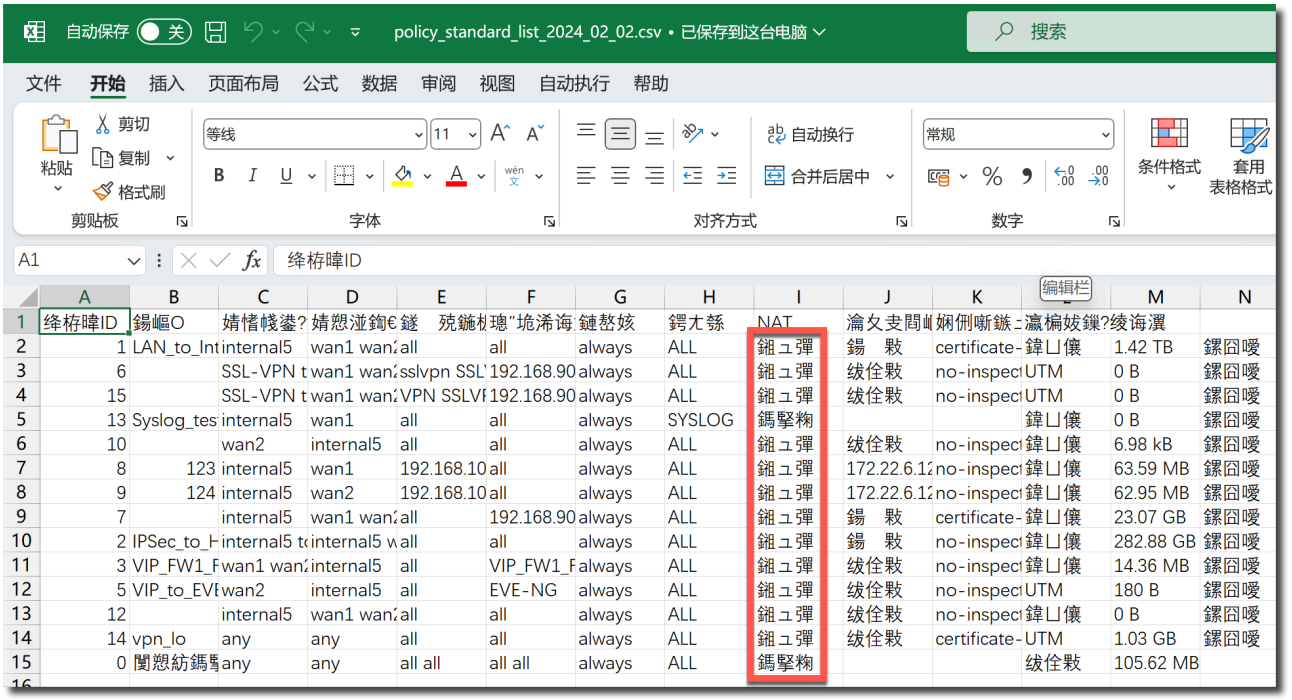
导出方法

在防火墙策略、组播策略、IPv6 组播策略、IPv4 DoS 策略、IPv6 DoS 策略页面，点击页面右上角的导出按钮，可以导出 CSV 或 JSON 格式的策略列表。



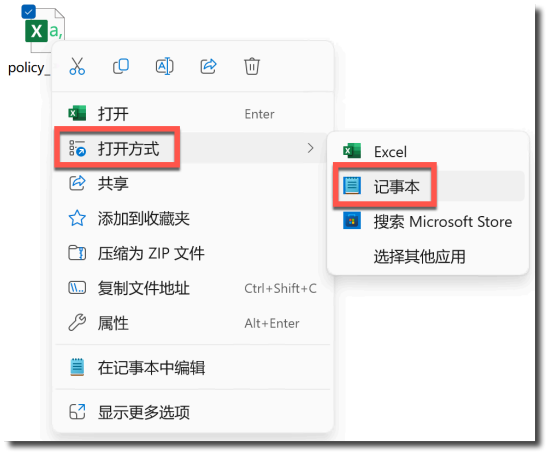
导出 CSV 乱码

导出的策略内容中如果包含中文，导出的 CSV 文件中，中文字符会显示为乱码。这是由于 CSV 文件默认中文支持 ANSI 编码，且没有预留修改编码的选项。

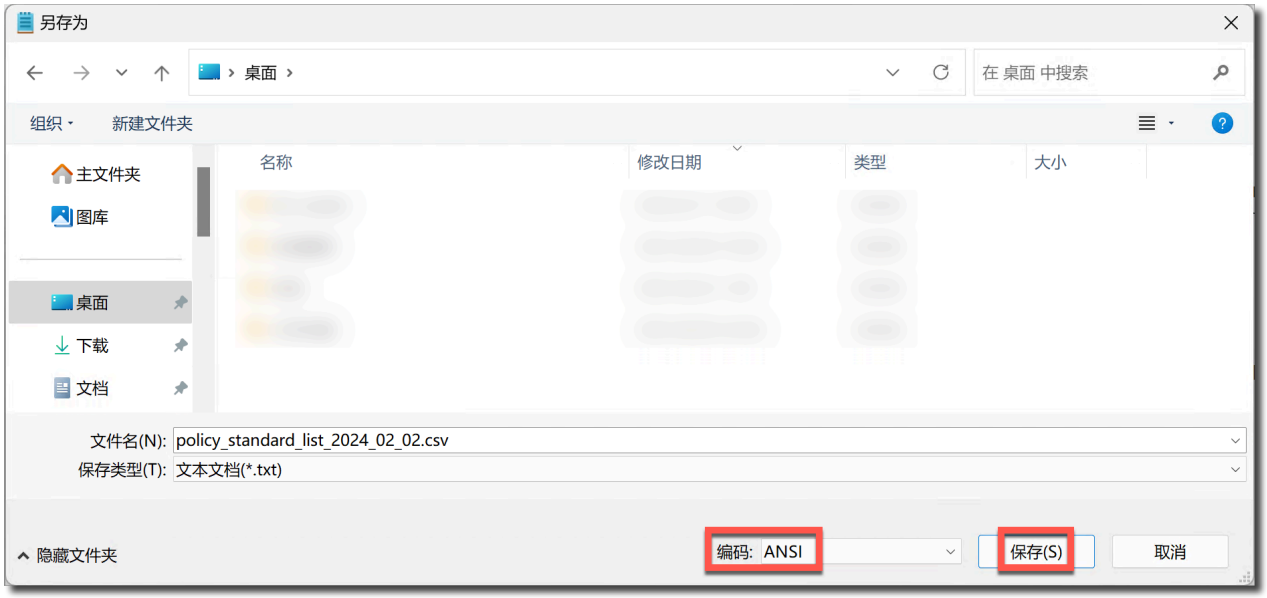


CSV 乱码解决方法

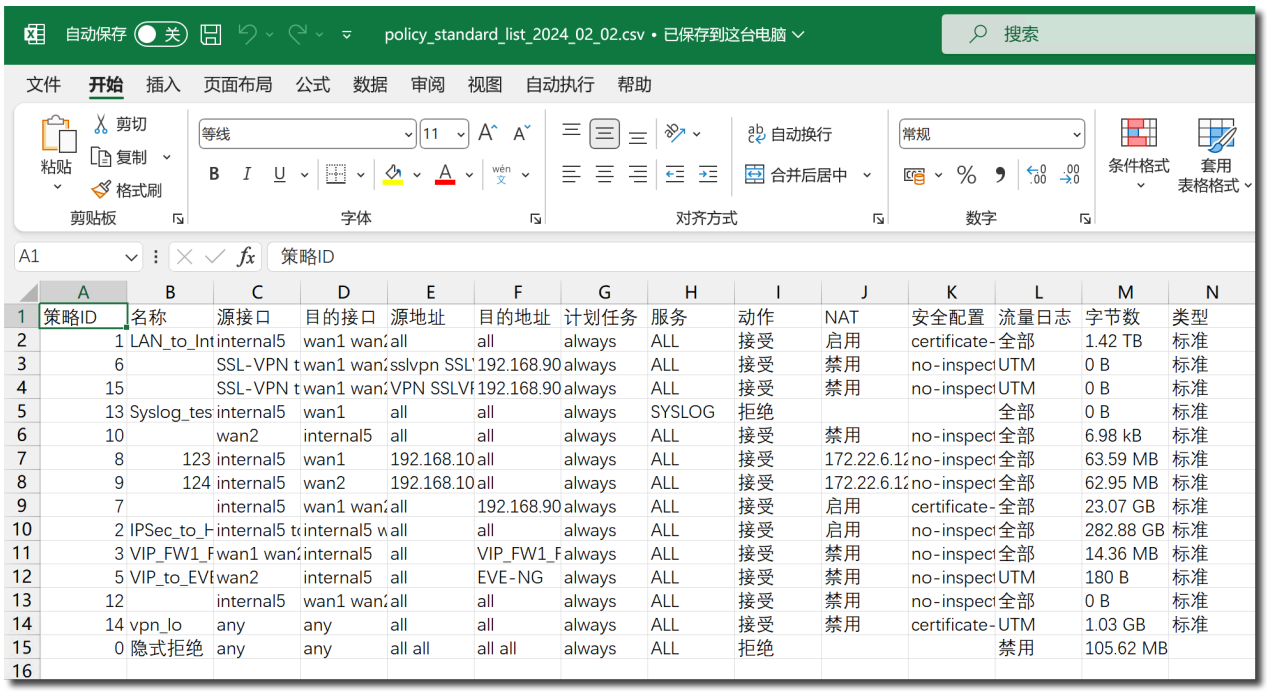
1. 右键点击导出的 CSV 文件，使用记事本打开该文件。



2. 在打开的记事本中点击文件 → 另存为，编码选择 ANSI，并保存文件。



3. 打开保存的 CSV 文件，乱码已经可以正常显示为中文。



策略ID	名称	源接口	目的接口	源地址	目的地址	计划任务	服务	动作	NAT	安全配置	流量日志	字节数	类型
1	LAN_to_Int	internal5	wan1 wan2	all	all	always	ALL	接受	启用	certificate-	全部	1.42 TB	标准
6		SSL-VPN t	wan1 wan2	sslvpn SSL	192.168.90	always	ALL	接受	禁用	no-inspect	UTM	0 B	标准
15		SSL-VPN t	wan1 wan2	VPN SSLVI	192.168.90	always	ALL	接受	禁用	no-inspect	UTM	0 B	标准
13	Syslog_tes	internal5	wan1	all	all	always	SYSLOG	拒绝			全部	0 B	标准
10		wan2	internal5	all	all	always	ALL	接受	禁用	no-inspect	全部	6.98 kB	标准
8	123	internal5	wan1	192.168.10	all	always	ALL	接受	172.22.6.12	no-inspect	全部	63.59 MB	标准
9	124	internal5	wan2	192.168.10	all	always	ALL	接受	172.22.6.12	no-inspect	全部	62.95 MB	标准
7		internal5	wan1 wan2	all	192.168.90	always	ALL	接受	启用	certificate-	全部	23.07 GB	标准
2	IPSec_to_H	internal5 t	internal5 w	all	all	always	ALL	接受	启用	no-inspect	全部	282.88 GB	标准
3	VIP_FW1_F	wan1 wan2	internal5	all	VIP_FW1_F	always	ALL	接受	禁用	no-inspect	全部	14.36 MB	标准
5	VIP_to_EVI	wan2	internal5	all	EVE-NG	always	ALL	接受	禁用	no-inspect	UTM	180 B	标准
12		internal5	wan1 wan2	all	all	always	ALL	接受	禁用	no-inspect	全部	0 B	标准
14	vpn_lo	any	any	all	all	always	ALL	接受	禁用	certificate-	UTM	1.03 GB	标准
0	隐式拒绝	any	any	all all	all all	always	ALL	拒绝			禁用	105.62 MB	

配置 Session TTL

功能介绍

1. 会话生存时间，即会话建立后无任何数据传送情况下的存活时间，默认为 3600 秒，当该会话在超时之前有任何数据匹配该会话，则该会话 ttl 计时器复位到该数值，如 3600 秒。有的服务如数据库。需要长连接，尽可能能将其 TCP 的会话超时时间设置得较大。
2. session-ttl 配置有 4 种不同的类型，分为全局 session-ttl 、全局自定义的 session-ttl、策略 session-ttl 、服务对象 session-ttl。session-ttl 必须在 CLI 下配置。

> **重要**
>
> 配置 session-ttl 时间只对状态为“已建立”的会话 TTL 时间生效。

配置方法

1. 配置全局 session-ttl。

```
config system session-ttl
    set default 3600    //TCP会话的老化时间，300-2764800秒(最大为32天)，默认为3600s
end

config system global
    set udp-idle-timer 180    //UDP会话的老化时间，1-86400秒(最大为1天)，默认为180s
end
```

2. 全局自定义 session-ttl，可自由定义协议、端口与会话超时时间。

```
config system session-ttl
    config port
        edit 1
            set protocol 6
            set timeout 86400
            set start-port 5000
            set end-port 5000
        next
        edit 2
            set protocol 17
            set timeout 360
            set start-port 4500
            set end-port 4500
        next
        edit 3
            set protocol 1
            set timeout 60
        next
    end
end
```

3. 基于策略的 session-ttl，针对匹配该策略的所有协议的会话生效。

```
config firewall policy
    edit 1
        set srcintf lan
        set dstintf wan1
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ANY
        set session-ttl 604800    //针对匹配该策略的所有协议/端口的会话生效
        set nat enable
    next
end
```

4. 基于服务对象的 session-ttl，定义不同的对象，虽然同为 23 端口，可以配置指定不同协议的 session-ttl 时间，被策略引用后才会生效。

```
config firewall service custom
    edit telnet
        set protocol TCP
        set tcp-portrange 23
        set session-ttl 7200
    next
    edit "telnetnew"
        set protocol UDP
        set tcp-portrange 23
        set session-ttl 3600
    next
end
```

Session TTL 优先级

全局 session-ttl（低）< 全局自定义 session-ttl < 策略 session-ttl < 被策略引用的服务对象 session-ttl（高）。优先级高的 session-ttl 优先被使用。

会话 dirty 机制

简介

会话 Dirty 状态

当 FortiGate 接收到新会话的首个数据包（如 TCP SYN）时，设备会执行路由查询，并根据防火墙策略判断是否允许该流量通过，然后创建对应的会话。对首个响应数据包（如 TCP SYN ACK）仅执行路由查找（不检查防火墙策略）。

FortiGate 会将路由查询信息（出接口与网关）、防火墙策略 ID 及其他信息（如地址转换、UTM 检测、认证、日志记录等）保存至会话中，并将其状态标记为 `may_dirty`。其中本文主要关注其中的 `state=` 字段中的会话 dirty 状态，如下所示，该会话的状态为 `may_dirty`。

```bash
session info: proto=6 proto_state=11 duration=2 expire=3597 timeout=3600 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty ndr npu f00     ←——这里的会话状态为may_dirty
statistic(bytes/packets/allow_err): org=116/2/1 reply=60/1/1 tuples=3
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin→sink: org pre→post, reply pre→post dev=40→7/7→40 gwy=10.10.12.1/192.168.100.177
hook=post dir=org act=snat 192.168.100.177:54449→223.5.5.5:443(10.10.12.2:54449)
hook=pre dir=reply act=dnat 223.5.5.5:443→10.10.12.2:54449(192.168.100.177:54449)
hook=post dir=reply act=noop 223.5.5.5:443→192.168.100.177:54449(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=0c:37:96:06:d7:5d
misc=0 policy_id=1 pol_uuid_idx=661 auth_info=0 chk_client_info=0 vd=0
serial=00172c91 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×003c94 ips_offload ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=1/1, epid=16/16, ipid=95/66, vlan=0×0000/0×0000
vlifid=95/66, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=2/0, ha_divert=0/0
```

会话 dirty 状态有如下 2 种：

- `may_dirty`：如果流量被防火墙策略允许，且设备没有发生任何策略、路由等变化，会话保持 `may_dirty` 状态。
- `dirty`：如果防火墙策略中的参数（如源、目的）发生变化，或其他可能触发状态变更的条件（如路由变化），现有会话的 `may_dirty` 标记将被修改为 `dirty` 状态。随后 FortiGate 需要重新评估下一个匹配该会话数据包的路由、策略等信息：
  - 如果流量仍被防火墙策略允许，则移除 `dirty` 标志并恢复 `may_dirty` 标志，更新会话中的 NAT、网关、策略 ID 等信息。
  - 若流量经重新评估后被判定为拒绝，则标记为 `block` 标志，并保留在会话表中直至过期。任何匹配带有 `block` 标志会话的数据包均会被丢弃。
  - 只要会话的状态没有被标记为 `dirty`，FortiGate 就不会对该会话执行重新评估。

会话 dirty 原因

以下是可能导致会话被标记为 `dirty` 的原因：

- 防火墙策略变化。
- 路由变化。
- 网络相关的配置变更。
- 执行 FortiGuard 计划更新（仅当下载新定义且策略关联了相关安全配置文件时）。
- 若会话通过 FGSP 协议进行同步。

会话 dirty 相关配置

firewall-session-dirty

防火墙策略配置发生变化后，相关会话的 dirty 状态变化会根据 `firewall-session-dirty` 配置决定。

- **VDOM 级别配置**

```bash
config system settings
    set firewall-session-dirty { check-all | check-new | check-policy-option }
end
```

- `check-all`（默认）：所有被编辑的防火墙策略影响的会话状态被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。
- `check-new`：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。
- `check-policy-option`：使用防火墙策略级别汇总的配置。

- **策略级别配置**

```bash
config firewall policy
    edit <id>
        set firewall-session-dirty < check-all | check-new >
    next
end
```

  - `check-all`（默认）：所有被编辑的防火墙策略影响的会话状态被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。
  - `check-new`：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。

fqdn-session-check

FQDN 地址对象解析的 IP 发生变化后，相关会话的 dirty 状态变化会根据 `fqdn-session-check` 配置决定。

```bash
config system settings
    set fqdn-session-check { enable | disable }
end

FQDN地址示例:
config firewall address
    edit "www.fortinet.com"
        set type fqdn
        set fqdn "www.fortinet.com"
    next
end
```

- `disable`（默认）：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。
- `enable`：现有相关会话被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。

snat-route-change

该参数决定路由变化后，SNAT 的会话表是否跟随着一起变化更新。

```bash
config system global
    set snat-route-change { enable | disable }
end
```

- `disable`（默认）：会话表项不变化，继续保持该会话，保持原有出口。
- `enable`：会话表项变化，会话状态会置位为 `dirty`，并且后续同一会话的数据将会重新上送 CPU 查询策略/路由等参数，重新进行策略的匹配，一旦路由变化将会匹配新的出口进行数据的转发。

preserve-session-route

该参数基于路由出接口，决定路由变化后，非 SNAT 的会话表是否跟随着一起变化更新。

```
config system interface
    edit "port2"
        set preserve-session-route { enable | disable }
    next
end
```

- `disable`（默认）：不保持会话，立刻将会话标记为 `dirty`，并重新评估后续流量。
- `enable`：强制保持会话，不会主动的更新会话中路由的下一跳地址。
- 这个命令的优先级更高，只要开启了接口的会话保持，无论 `snat-route-change` 是否开启，会话还是为会优先保持，会话/流量不会更变出接口信息。

ext-resource-session-check

外部威胁源对象（如 IP、域名、Web 分类等）的内容发生变化后，相关会话的 dirty 状态变化会根据 `ext-resource-session-check` 配置决定。

> **重要**
>
> 外部威胁源的配置请参考：Security Fabric → 威胁情报源 ↗章节。

```bash
config system settings
    set fqdn-session-check { enable | disable }
end
```

- `disable`（默认）：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。
- `enable`：现有相关会话被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。

dyn-addr-session-check

动态地址（如从 SDN 连接器获取地址）对象中的地址内容发生变化后，相关会话的 dirty 状态变化会根据 `dyn-addr-session-check` 配置决定。

```
config system settings
    set dyn-addr-session-check { enable | disable }
end
```

- `disable`（默认）：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。
- `enable`：现有相关会话被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。

iprope 列表

简介

iprope 表是 FortiGate 内部用于表示管理员所配置防火墙策略的结构化数据形式。 iprope 将所有策略条目按功能划分为多个策略组，每组条目从上到下依次匹配，一旦命中即执行相应动作，不再继续检查同组的后续条目。

匹配依据包括：

- 源/目的地址
- 端口范围
- 协议类型

可能的动作包括：

- drop：丢弃数据包，不建立会话。
- accept：放行数据包，建立会话。
- redirect：将数据包交付内部处理逻辑。

> **提示**
>
> iprope 表中不存在 GUI 防火墙策略的“隐式拒绝”，所有条目均为显式策略。

常用命令

用于查看不同策略组的命令：

```bash
diagnose firewall iprope list 100002   # 静态 SNAT 策略
diagnose firewall iprope list 100000   # VIP（DNAT）策略
diagnose firewall iprope list 100004   # 普通 Forward 策略
diagnose firewall iprope list 10000e   # 系统生成的 Local-In 策略
diagnose firewall iprope list 100015   # Traffic Shaping 策略
diagnose firewall iprope list 100017   # 简单 ZTNA 策略
```

输出示例及字段说明

```bash
FortiGate # diagnose firewall iprope list 00100004

Policy Group 00100004

policy index=1 uuid_idx=14 action=accept
flag (8050108): redir nat master use_src pol_stats
flag2 (4000): resolve_sso
flag3 (20): schedule(always)
cos_fwd=255 cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0

zone(1): 3 -> zone(1): 6
source(1): 10.0.1.0-10.0.1.255, uuid_idx=12,
dest(1): 192.0.2.0-192.0.2.255, uuid_idx=13,
service(1):
[0:0x0:0/(0,65535)->(0,65535)]
helper:auto
```

字段含义说明：

字段	含义
policy index	策略条目索引
action	动作类型
flag/flag2/flag3	启用的处理能力标识，如 NAT、SSL、策略统计等
zone	入/出口 zone 标识
source/dest	源/目的地址范围和 UUID
service	协议与端口匹配范围
helper	ALG（会话助手）配置方式

策略组列表

Group ID	名称	说明
00000003	AUTH_DEFAULT	Authentication 策略
00000005	CAPTIVE_PORTAL	配置 security-mode 的接口
00004e20	SESS_HELPER	Session Helper 相关
00100001	CUST_LOCAL_IN	自定义 Local-In 策略
00100002	STATIC_SNAT	静态 SNAT 或地址池 SNAT
00100003	DEC_FWD	IPsec 解密后策略匹配
00100004	ENC_FWD	所有 Forward 策略
0010000a	MULTICAST	组播策略
0010000c	EP_REDIR	Endpoint Control 相关策略
0010000d	CENTRAL_NAT	Central NAT 策略
0010000e	IMPLICIT_IN	系统生成的 Local-In 策略（例如管理协议）
0010000f	ADMIN_IN	管理面流量的基于接口放行策略
00100011	ZTNA_PROXY	ZTNA 策略
00100015	Traffic Shaper	Forward 整形策略
00100018	Traffic Shaper	Local-In 整形策略
00100019	Traffic Shaper	Local-Out 整形策略

提示

如果某功能在设备中启用，则对应策略组中可能包含默认策略条目。

注意事项

- 策略组具有固定的 Group ID。
- 每条策略使用 policy index 标识。
- 出厂配置中若某功能默认启用，则可能存在预置条目。

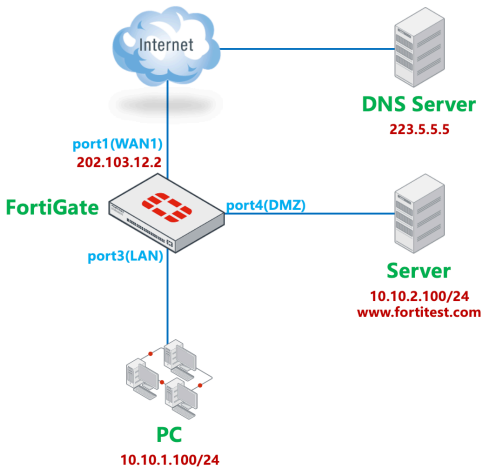
- 所有策略检查在数据平面内部完成，不在 GUI 中显示。

转换 DNS 解析结果

DNS Translation 功能用于将穿越 FortiGate 的 DNS 解析结果替换为指定的结果。

具体适用场景请见如下网络拓扑与说明。

网络拓扑



1. DNS 服务器位于 Internet，内网 PC 填写的 DNS 服务器均为公网 DNS 服务器的 IP 223.5.5.5。
2. FortiGate 通过 VIP 将 Server 的端口映射到 WAN1（port1）的 202.103.12.2。
3. Server 的域名为 `www.fortitest.com`，通过公网 DNS 服务器解析的结果为 202.103.12.2。
4. 需要实现的效果：
   - 外网客户端访问 Server 时，直接通过 `www.fortitest.com` 解析到公网 IP 202.103.12.2，通过 FortiGate 的 VIP 映射访问即可。
   - 内网客户端访问 Server 时，首先通过公网的 DNS 服务器 223.5.5.5 解析 `www.fortitest.com` 的 IP 地址，DNS Server 会回复解析结果为 202.103.12.2，FortiGate 需要将解析结果修改为 Server 的内网 IP 10.10.2.100，然后转发给内网客户端，内网客户端直接通过解析到的 Server 内网 IP 访问服务器。

配置步骤

1. 基础网络配置、上网策略、与 Server 的 VIP 配置（略）。

2. 在 FortiGate 上配置 DNS Translation，将 DNS 请求结果为 202.103.12.2 的 DNS 响应报文中的 IP 地址修改为 Server 的内网 IP 10.10.2.100。

```bash
config firewall dnstranslation
    edit 1
        set src 202.103.12.2
        set dst 10.10.2.100
    next
end
```

结果验证

1. 公网客户端 PC 使用 DNS 服务器 223.5.5.5 解析 Server 的域名（流量不经过 FortiGate），得到结果为 Server 对应的 VIP 地址 202.103.12.2。

```bash
C:\Users\Administrator.SUMMERICE2019>nslookup www.fortitest.com
服务器:  public1.alidns.com
Address:  223.5.5.5

非权威应答:
名称:    www.fortitest.com
Address:  202.103.12.2
```

2. 内网客户端 PC 使用 DNS 服务器 223.5.5.5 解析 Server 的域名（流量经过 FortiGate），得到结果为 Server 内网 IP 10.10.2.100。

```bash
C:\Users\Administrator.SUMMERICE2019>nslookup www.fortitest.com
服务器:  public1.alidns.com
Address:  223.5.5.5

非权威应答:
名称:    www.fortitest.com
Address:  10.10.2.100
```

单线路上网配置

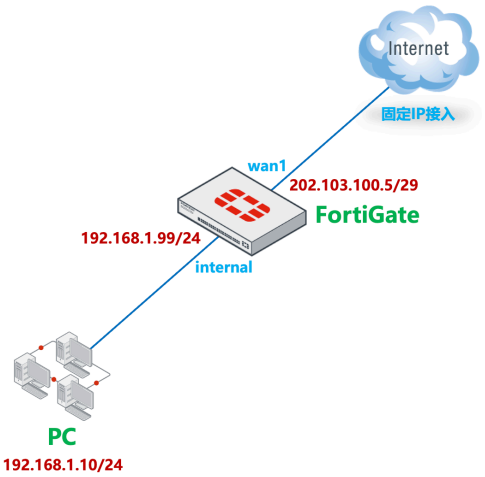
- 静态地址线路上网配置
- 单条 ADSL 线路上网配置
- DHCP 线路上网配置

静态地址线路上网配置

网络需求

外网接口使用专线，由运营商分配指定的静态地址，内网为 192.168.1.0/24 网段，实现基本上网功能。

网络拓扑



假设运营商分配的地址如下：

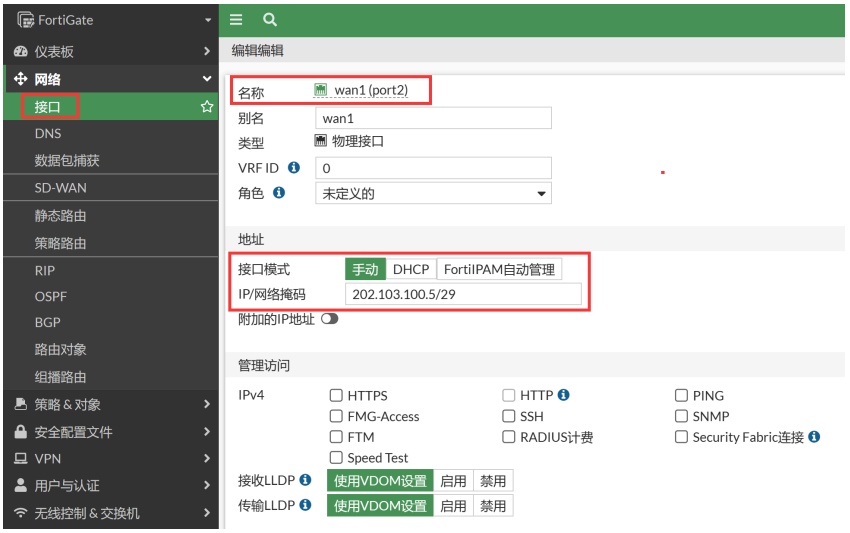
网段	分配 IP	网关地址	DNS
202.103.100.0/29	202.103.100.5	202.103.100.1	202.106.196.115

配置要点

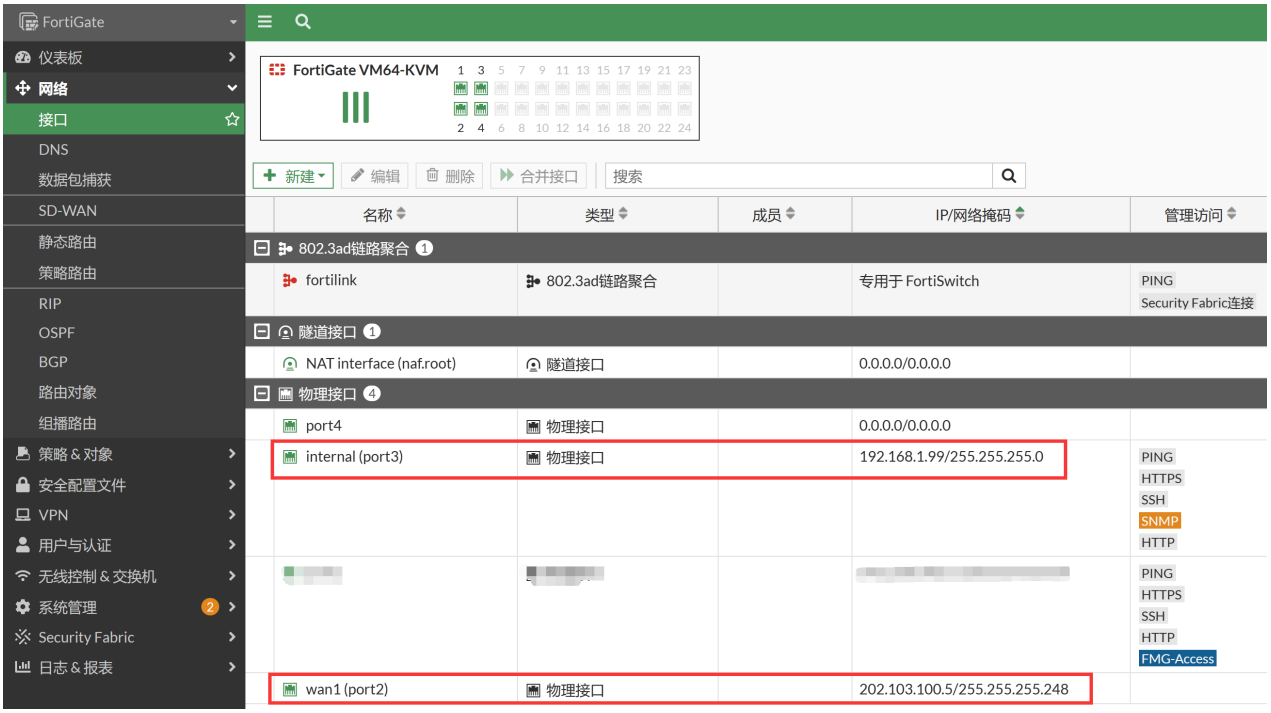
- wan1（port2）口：IP 地址配置为互联网运营商所提供的地址
- internal（port3）口：IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置静态路由表
- 配置地址对象 lan，内容为 192.168.1.0/24
- 配置从 internal（port3）到 wan1（port2）口的策略，并开启 NAT

配置步骤

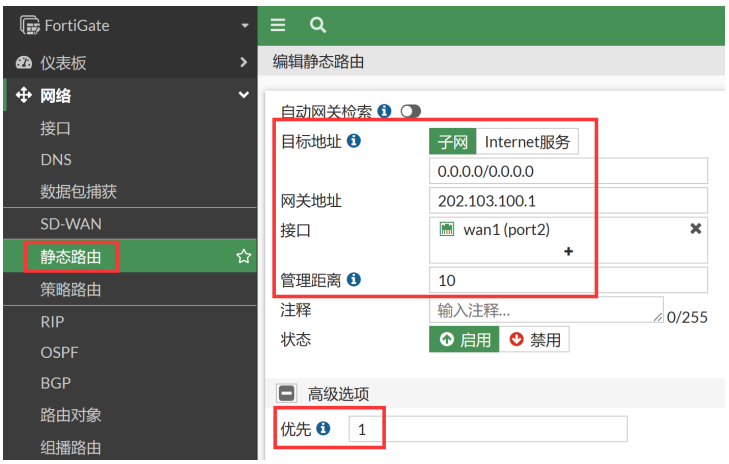
1. 进入网络 → 接口，配置 wan1 口的 IP 地址。对于 202.103.100.0/29 网段，202.103.100.0 网络地址和 202.103.100.7 广播地址均不可用，202.103.100.1 为运营商网关所用，202.103.100.2—202.103.100.6 为可用 IP 地址范围。所以配置给 wan 接口的地址我们选择为 202.103.100.5。



2. 配置 internal（port3）接口的 IP 为 192.168.1.99/24。选择性地开启接口的管理功能，建议内部开启 https ,ssh, ping 服务。



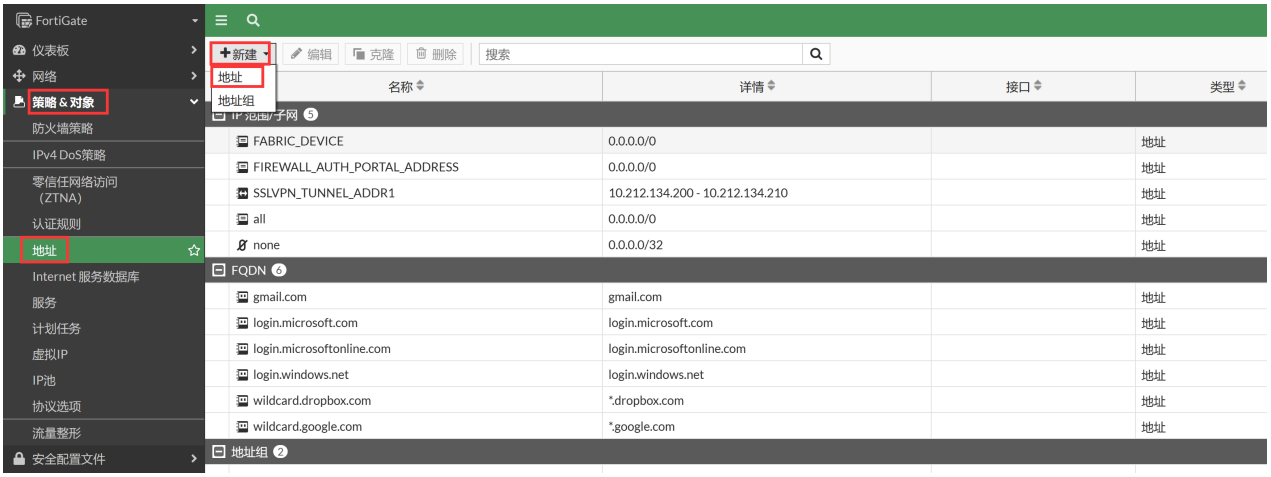
3. 配置静态路由，添加默认路由指向运营商网关。



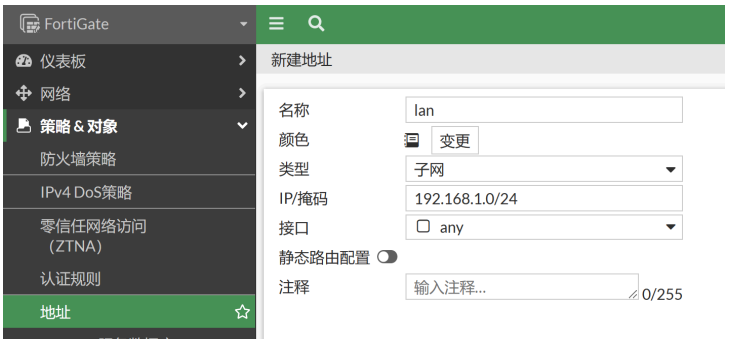
```
config router static
    edit 1
        set gateway 202.103.100.1
        set device "port2"
    next
end
```

- 目的的 IP/子网掩码：由于是默认网关，使用默认的 0.0.0.0/0.0.0.0 即可
- 接口：该路由所关联的接口，wan1 口，必须正确填写，否则该路由无法工作
- 网关：下一跳 IP 地址， 即 wan1 口对端运营商设备接口的 ip 地址
- 管理距离：默认 10
- 高级选项 - 优先级：默认 1

4. 配置地址资源，进入防火墙 & 对象 → 地址，点击新建地址按钮。



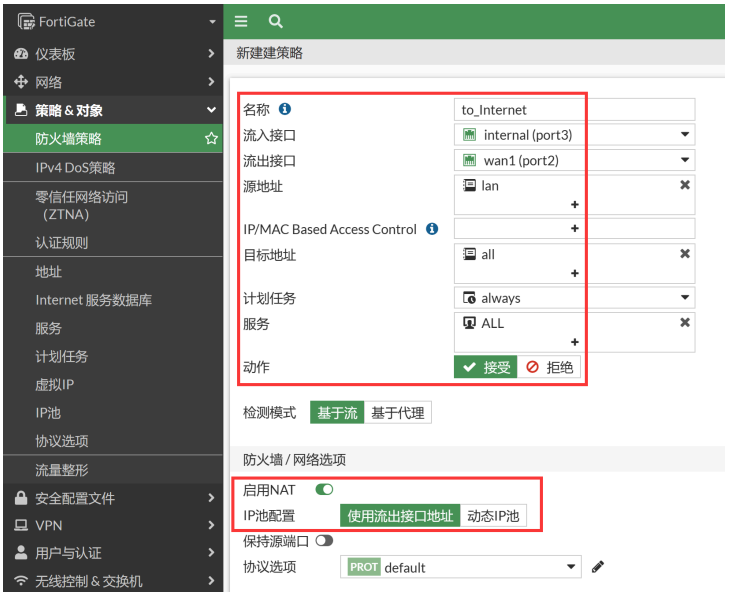
5. 名称配置为 lan，地址节点选择子网：192.168.1.0/24，点击确认。




```
config firewall address
    edit "lan"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，放通内网到 Internet 的流量，并开启 NAT，点击确定按钮后，系统自动保存配置，策略生效。

> **重要**
>
> 如果选择“动态 IP 池”作为“IP 池配置”，所引用 IPPool 配置中的外部 IP 范围不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 IPPool 后，IPPool 中的地址属于 FortiGate 自身的 IP 地址，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。




```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port3"
        set dstintf "port2"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：internal（port3）
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：wan1（port2）
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

结果验证

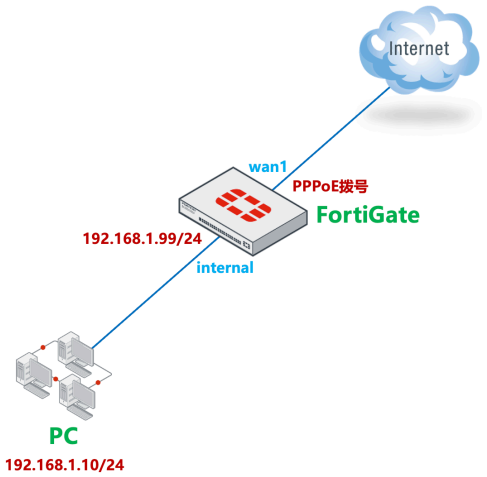
将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 一般设置为当地的 DNS 即可，电脑可正常上网。

单条 ADSL 线路上网配置

网络需求

外网接口使用 ADSL 拨号，内网为 192.168.1.0/24 网段，实现基本上网功能。

网络拓扑



配置要点

- wan1（port2）口: 接 ADSL 的接口，务必开启自动获取网关，这样 ADSL 拨号成功后设备会自动生成默认路由，无需手动配置默认路由
- internal（port3）口: IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置地址对象“lan”，内容为 192.168.1.0/24
- 配置从 internal（port3）到 PPPoE 口的策略，并开启 NAT

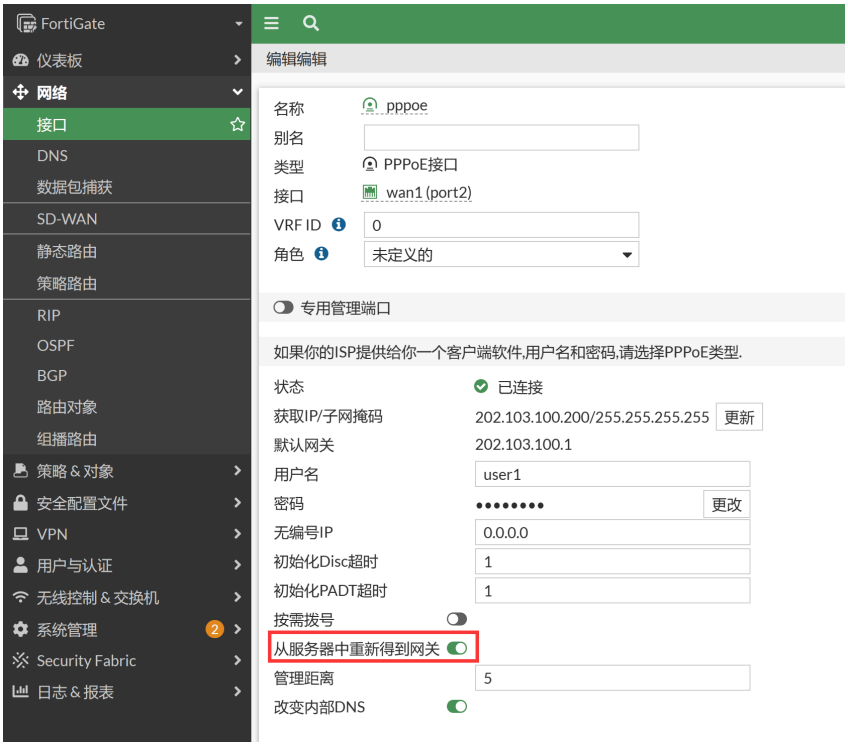
配置步骤

1. 配置 wan1 口的 PPPoE，建议在 CLI 下配置 PPPoE 接口，使用此方式配置获取到 IP 后会自动添加默认路由到运营商。

> **重要**
>
> 建议通过 CLI 的方式新建 PPPoE 虚拟接口，参考：部署前注意事项 → PPPoE 接口 ↗配置章节。

```
config system pppoe-interface
    edit "pppoe"
        set device "port2"
        set username "user1"
        set password xxxxxxxx
    next
end

以上命令执行后，会自动在system interface下创建同名接口，其中包含“从服务器中重新得到网关”的配置
FortiGate # show full sys int pppoe
config system interface
    edit "pppoe"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set defaultgw enable
        set interface "port2"
    next
end
```

2. 以上配置完成后，可以在 GUI 上看到 wan1（port2）下多了一个 PPPoE 接口，并获取到了运营商的 IP。可以看到“从服务器中重新得到网关”已开启。

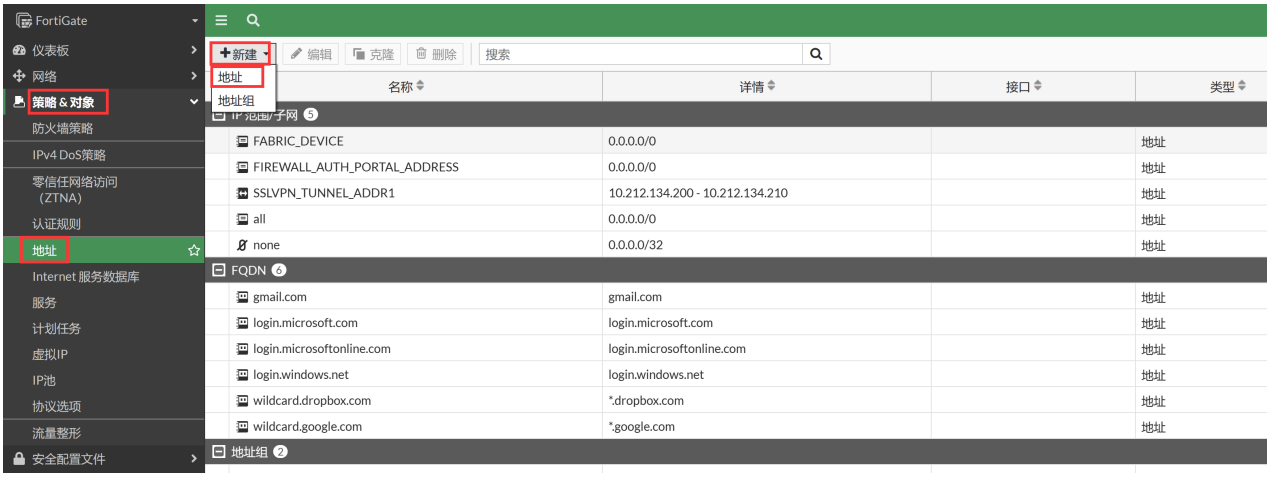



```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 202.103.100.1, pppoe, [1/0]
C       192.168.1.0/24 is directly connected, port3
C       202.103.100.1/32 is directly connected, pppoe
C       202.103.100.200/32 is directly connected, pppoe
```

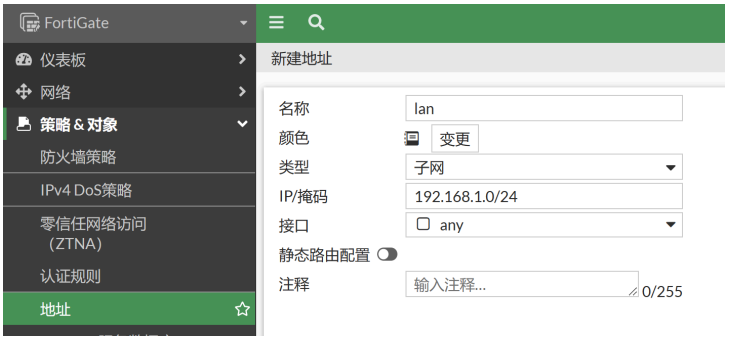
3. 配置 internal（port3）接口的 IP 为 192.168.1.99/24。选择性地开启接口的管理功能，建议内部开启 https ,ssh, ping 服务。



4. 配置地址资源，进入防火墙 & 对象 → 地址，点击新建地址按钮。

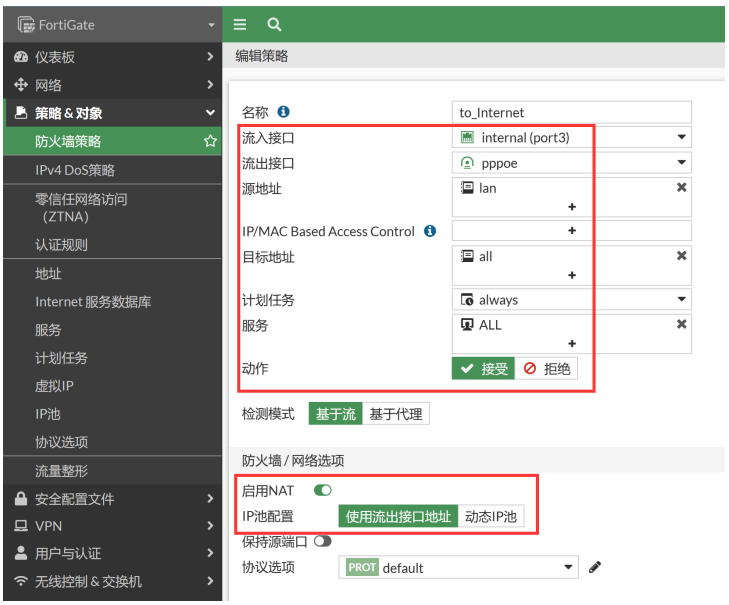


5. 名称配置为 lan，地址节点选择子网：192.168.1.0/24，点击确认。




```
config firewall address
    edit "lan"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，放通内网到 Internet 的流量，并开启 NAT，点击确定按钮后，系统自动保存配置，策略生效。




```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port3"
        set dstintf "port2"
        set action accept
        set srcaddr "lan"
```

```
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：internal（port3）
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：pppoe 接口
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

结果验证

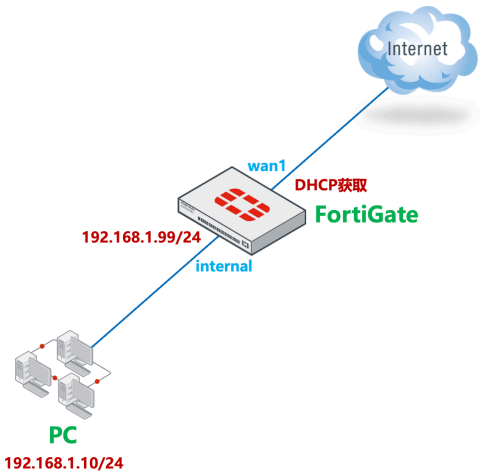
将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 一般设置为当地的 DNS 即可，电脑可正常上网。

DHCP 线路上网配置

网络需求

外网接口使用 DHCP，内网为 192.168.1.0/24 网段，实现基本上网功能。

网络拓扑

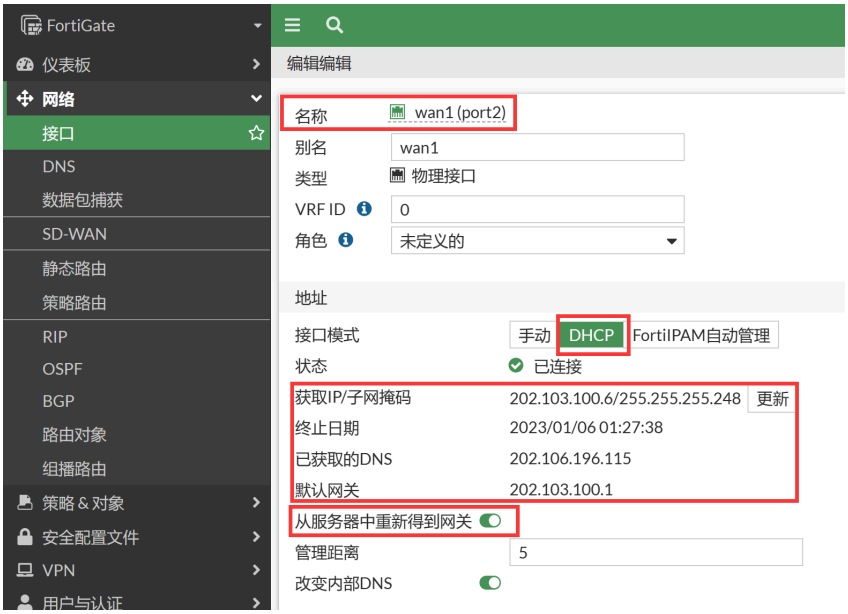


配置要点

- wan1（port2）口：务必勾选“从服务器重新得到网关”，这样 DHCP 地址获取成功后设备会自动生成默认路由，无需手动配置默认路由
- internal（port3）口: IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置地址对象“lan”，内容为 192.168.1.0/24
- 配置从 internal（port3）到 wan1（port2）口的策略，并开启 NAT

配置步骤

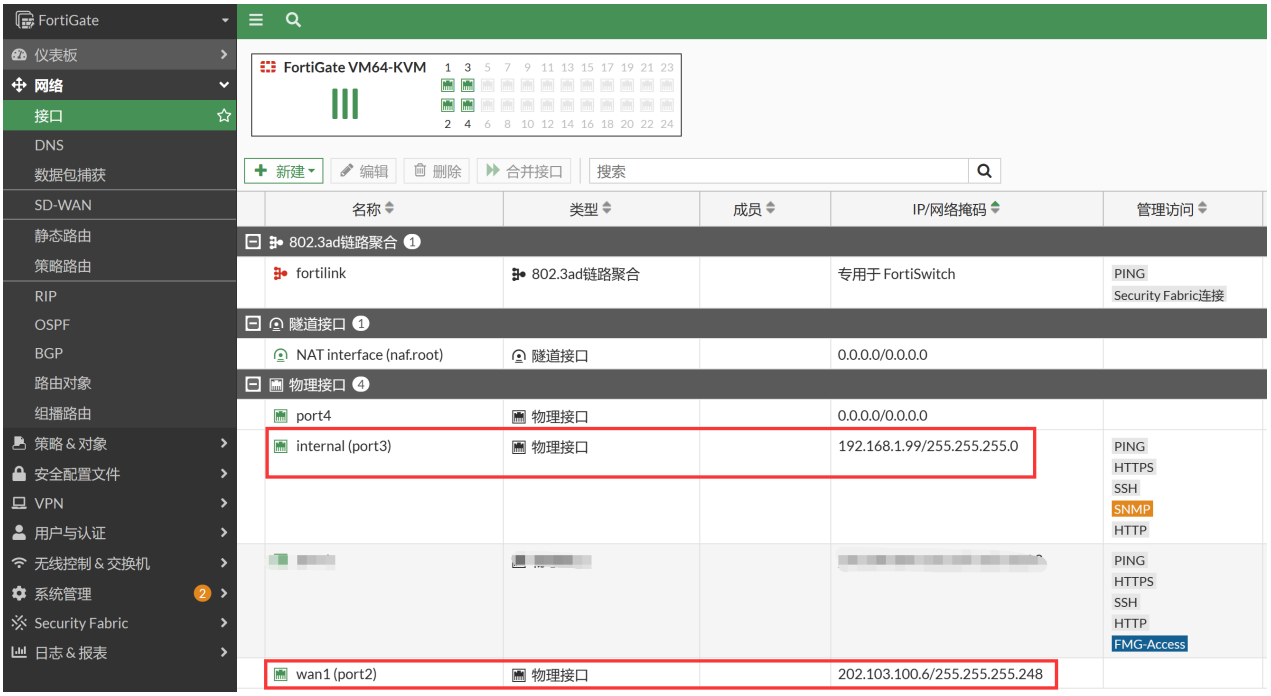
1. 进入网络 → 接口，编辑 wan1（port2）接口，将接口的模式配置为 DHCP，并勾选“从服务器重新得到网关”。



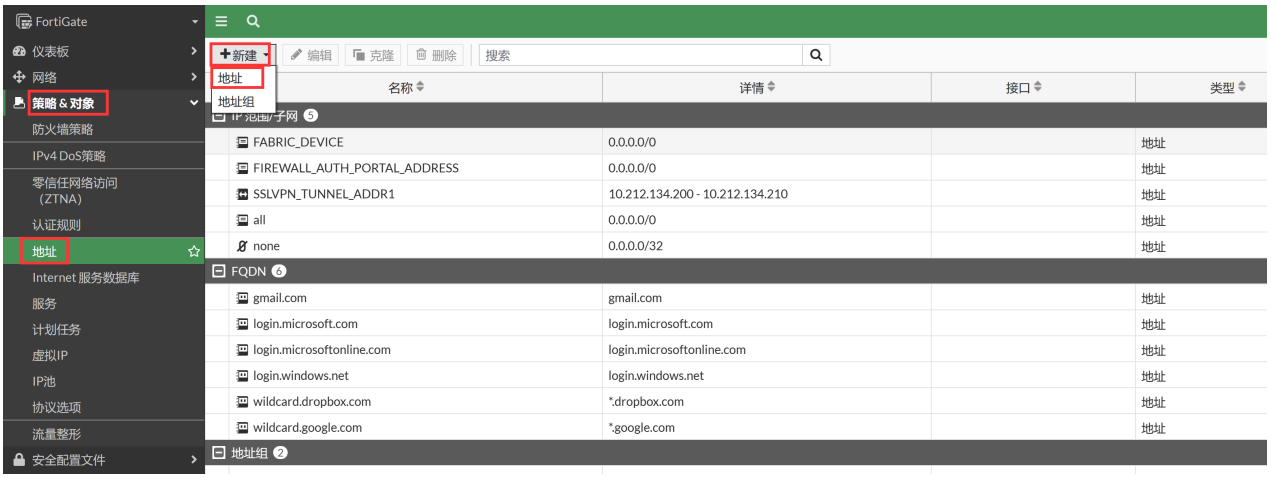
2. 以上配置完成后，等待 DHCP 获取到 IP 地址，然后查看路由表，可以看到通过 DHCP 自动添加了默认路由。

```
FortiGate #  get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 202.103.100.1, port2, [1/0]
C       192.168.1.0/24 is directly connected, port3
C       202.103.100.0/29 is directly connected, port2
```

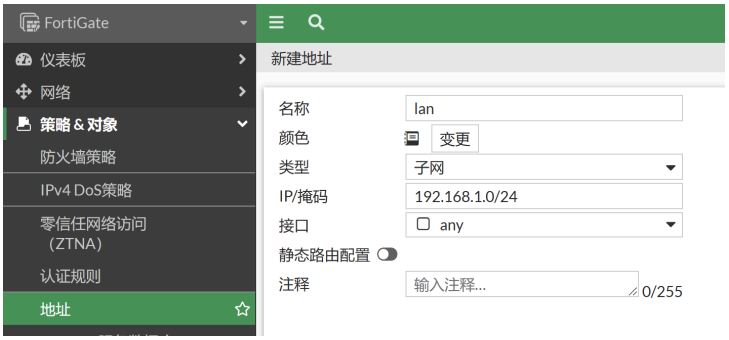
3. 配置 internal（port3）接口的 IP 为 192.168.1.99/24。选择性地开启接口的管理功能，建议内部开启 https ,ssh, ping 服务。



4. 配置地址资源，进入防火墙 & 对象 → 地址，点击新建地址按钮。

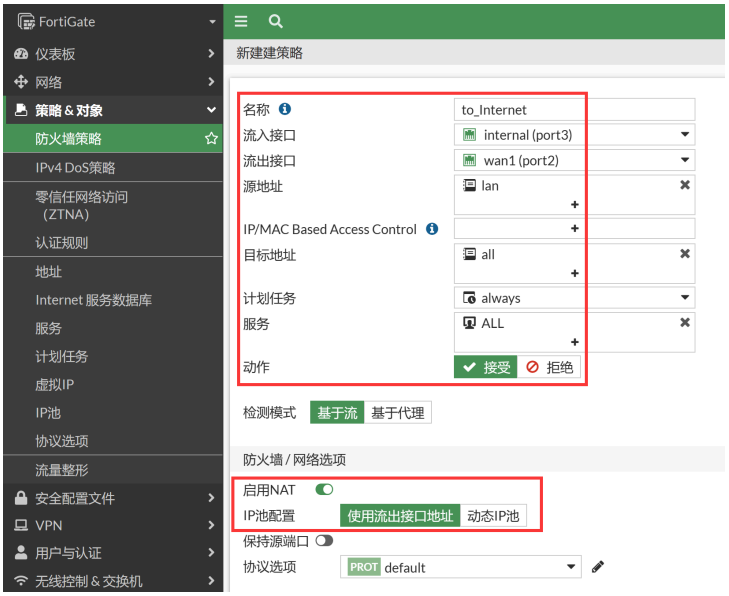


5. 名称配置为 lan，地址节点选择子网：192.168.1.0/24，点击确认。




```
config firewall address
    edit "lan"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，放通内网到 Internet 的流量，并开启 NAT，点击确定按钮后，系统自动保存配置，策略生效。




```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port3"
        set dstintf "port2"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：internal（port3）
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：pppoe 接口
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

结果验证

将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 一般设置为当地的 DNS 即可，电脑可正常上网。

多线路上网配置

- 双线路同运营商上网配置
- 双线路不同运营商上网配置
- 双线路 ADSL

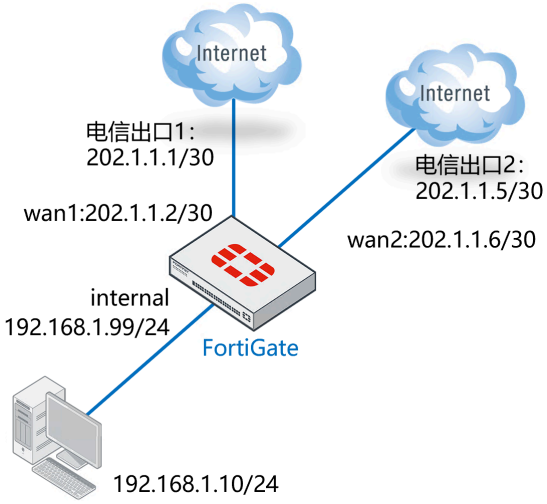
双线路同运营商上网配置

网络需求

现设备上申请了两条电信线路，带宽大小相同 ，通过配置实现负载均衡、冗余备份。

- 电信 1：wan1（port2）口：202.1.1.2/30，网关为 202.1.1.1
- 电信 2：wan2（port3）口：202.1.1.6/30，网关为 202.1.1.5

网络拓扑

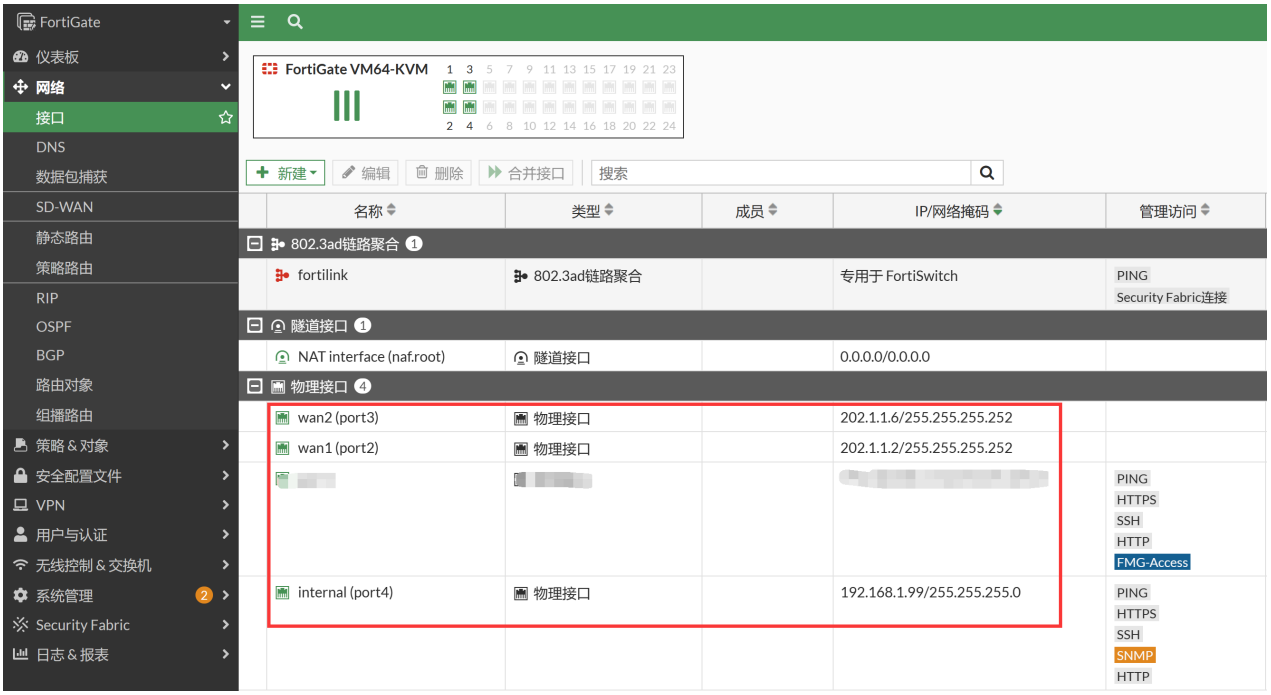


配置要点

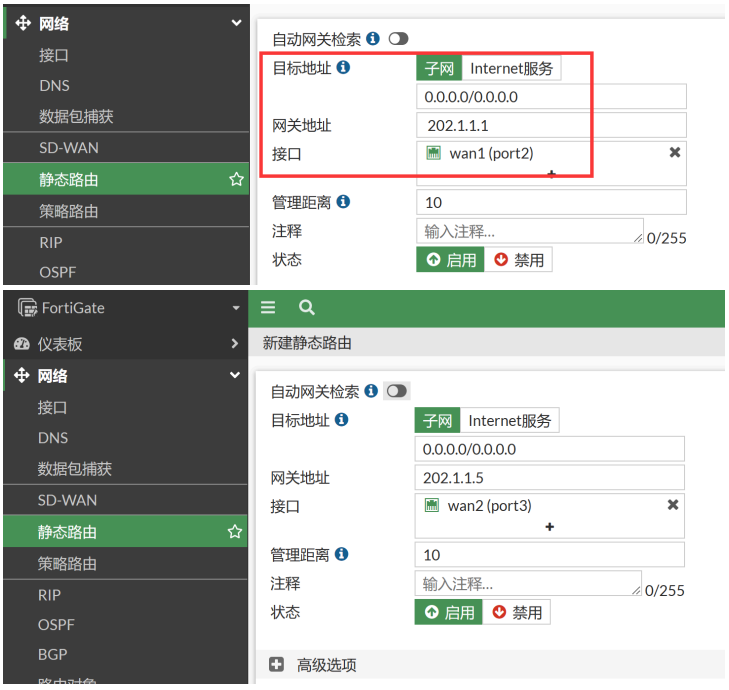
- 配置接口 IP
- 配置路由
- 配置 zone（untrust 和 trust）
- 配置策略
- 配置 ECMP 负载均衡方式

配置步骤

1. 配置接口 IP，进入网络 → 接口页面，编辑 wan1（port2）、wan2（port3）internal（port4），分别配置为 202.1.1.2/30、202.1.1.6/30、192.168.1.99/24。



2. 配置静态路由，创建两条链路的默认路由。




```
config router static
    edit 1
        set gateway 202.1.1.1
        set device "port2"
    next
    edit 2
        set gateway 202.1.1.5
        set device "port3"
    next
end
```

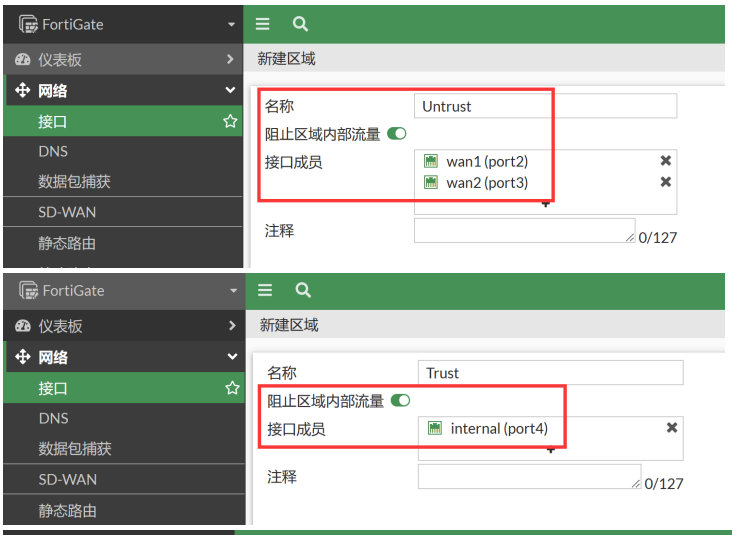
- 目的的 IP/子网掩码：由于是默认网关，使用默认的 0.0.0.0/0.0.0.0 即可
- 接口：该路由所关联的接口，wan1 口，必须正确填写，否则该路由无法工作
- 网关：下一跳 IP 地址， 即 wan1 口对端运营商设备接口的 ip 地址
- 管理距离：默认 10
- 高级选项 - 优先级：默认 1，优先级小的优先使用
- 注意：

- 要想实现两个出口线路同时工作，两条路由表的管理距离必须相同。否则路径长度长的路由条目，不会装入路由表
- 管理距离相等的同时，优先级必须相同。 如果路径相等优先级不同，则虽然两条路由都会被装入路由表，防火墙会优先采用优先级数值较小的路由工作，无法实现两条链路的流量均担

3. 配置区域，采用区域的方式，为了配置更加快捷、精简。如果采用基于物理接口的方式，则可能需要配置多条防火墙策略。进入网络 → 接口，点击新建 → 所属区域按钮。



4. 分别创建 Untrust 和 Trust 区域，



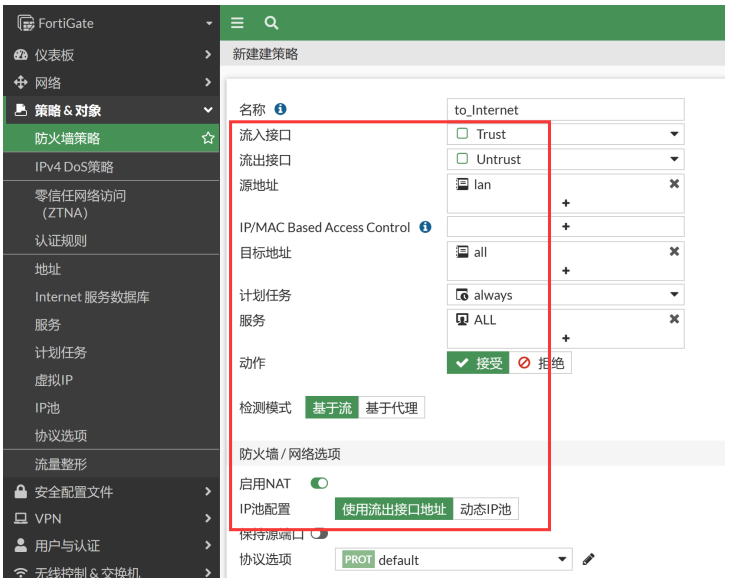



```
config system zone
    edit "Untrust"
        set interface "port2" "port3"
    next
    edit "Trust"
        set interface "port4"
    next
end
```

5. 配置上网放通策略，进入策略 & 对象 → 防火墙策略，新建安全策略。

重要

如果选择“动态 IP 池”作为“IP 池配置”，所引用 IPPool 配置中的外部 IP 范围不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 IPPool 后，IPPool 中的地址属于 FortiGate 自身的 IP 地址，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。




```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "Trust"
        set dstintf "Untrust"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：trust
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：untrust
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT： 选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 untrust 的接口组里的 wan1/wan2 接口地址，进行互联网访问
- 点击 "确定" 按钮后，系统自动保存配置，策略生效
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

6. 配置 ECMP 负载均衡方式（具体可参考 https://community.fortinet.com/t5/FortiGate/Technical-Tip-ECMP-Load-balancing-algorithms-for-IPv4-and-IPv6/ta-p/191149 ）。

```
FortiGate # config system settings
FortiGate (settings) # set v4-ecmp-mode ?
source-ip-based         Select next hop based on source IP.
weight-based            Select next hop based on weight.
usage-based             Select next hop based on usage.
source-dest-ip-based    Select next hop based on both source and destination IPs.
```

- 基于源 IP：默认配置，根据源地址的不同选择不同的下一跳
- 权重负载均衡： 根据权重值来选择路由（在默认路由的 CLI 下配置）
  - 比如 wan1 50， wan2 50，其他为 0 则 2 条链路会 1:1 的比例分配流量
  - 比如 wan1 50， wan2 100，则流量会按照 1:2 的比例分配
- 用量：根据线路带宽用量，选择用量较少的线路
- 源目 IP：根据源地址 - 目的地址对的不同选择不同的下一跳

- 推荐采用“基于源 IP”的负载均衡方式（默认）。因为例如网银、网游会进行源 IP 的验证，如果存在不同 IP 的多条流的交互（同时通过 wan1/wan2 做不同的 NAT 转换成不同的公网 IP，一个网银 user 使用了多个公网 IP 访问网银业务），将会导致网银业务交互失败、游戏掉线等问题。

结果验证

不同的源 IP 的客户端的上网流量被分配到了不通的链路。

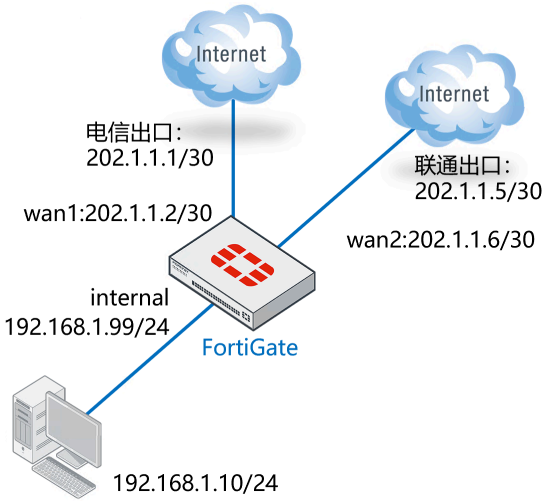
双线路不同运营商上网配置

网络需求

防火墙到电信和联通分别有一条链路，去往电信的 IP，通过 wan1（port2）口；去往联通的 IP，通过 wan2（port3）口。

- 电信：wan1（port2）口：202.1.1.2/30，网关为 202.1.1.1，NAT 地址池：202.1.1.1-10
- 联通：wan2（port3）口：102.1.1.2/30，网关为 102.1.1.1，NAT 地址池：200.0.0.1-10

网络拓扑

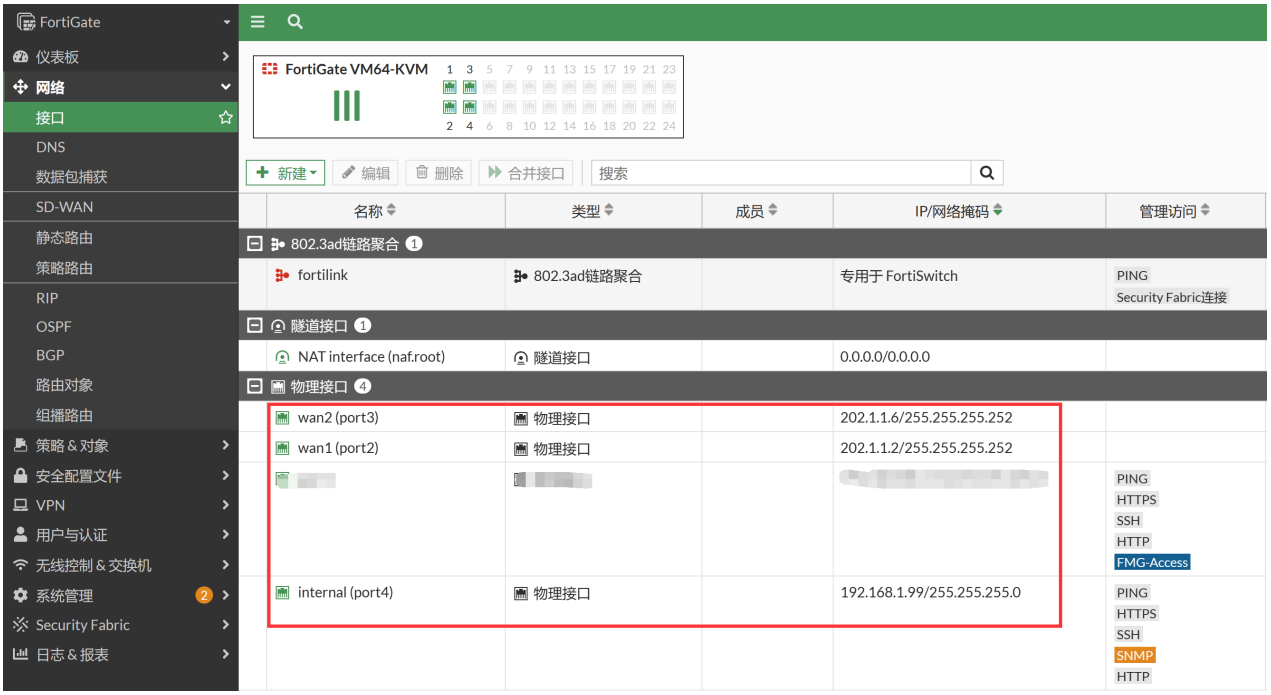


配置要点

- 配置接口 IP
- 配置路由
- 配置地址池
- 配置策略
- 注意：部分低端型号路由表规格有限，不适用导入运营商路由表的方法

配置步骤

1. 配置接口 IP，进入网络 → 接口页面，编辑 wan1（port2）、wan2（port3）internal（port4），分别配置为 202.1.1.2/30、202.1.1.6/30、192.168.1.99/24。



2. 配置静态路由，创建电信链路（wan1）的默认路由。




```
config router static
    edit 1
        set gateway 202.1.1.1
        set device "port2"
    next
end
```

- 目的的 IP/子网掩码：由于是默认网关，使用默认的 0.0.0.0/0.0.0.0 即可
- 接口：该路由所关联的接口，wan1 口，必须正确填写，否则该路由无法工作
- 网关：下一跳 IP 地址， 即 wan1 口对端运营商设备接口的 ip 地址
- 管理距离：默认 10
- 高级选项 - 优先级：默认 1，优先级小的优先使用
- 注意：
  - 要想实现两个出口线路同时工作，两条路由表的管理距离必须相同。否则路径长度长的路由条目，不会装入路由表
  - 管理距离相等的同时，优先级必须相同。 如果路径相等优先级不同，则虽然两条路由都会被装入路由表，防火墙会优先采用优先级数值较小的路由工作，无法实现两条链路的流量均担

3. 参照附件 → 命令行脚本批量转换工具说明↗和附件 → FortiGate 通用多运营商路由导入脚本↗附件章节，将联通明细路由导入 FortiGate（注意脚本中要开启允许静态路由引用地址对象）。

4. 配置地址池，进入策略 & 对象 → IP 池，新建两个运营商的地址池。

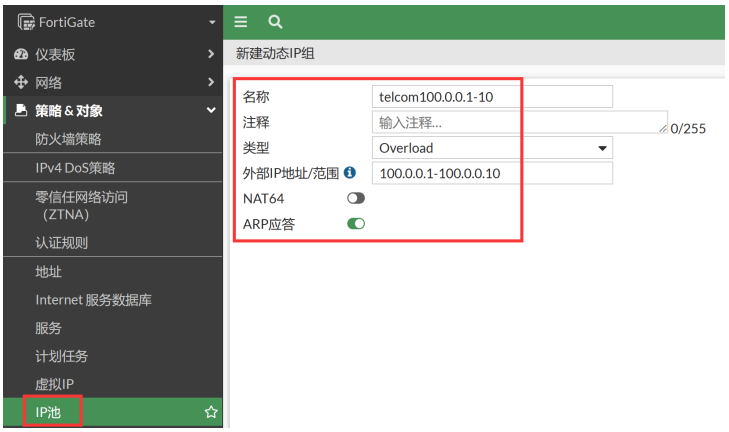
重要

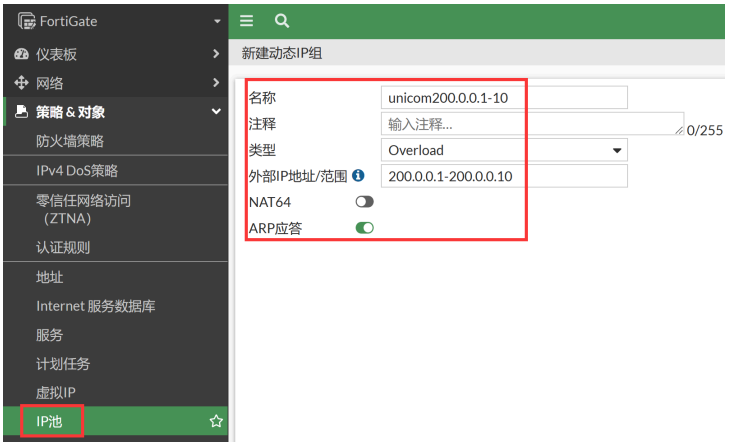
如果在 SD-WAN 中使用 ippool 的方式做 SNAT，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。

```
config firewall ippool
    edit xxxx
        set associated-interface port2
    next
end
```

重要

IPPool 配置中的外部 IP 范围不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 IPPool 后，IPPool 中的地址属于 FortiGate 自身的 IP 地址，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。

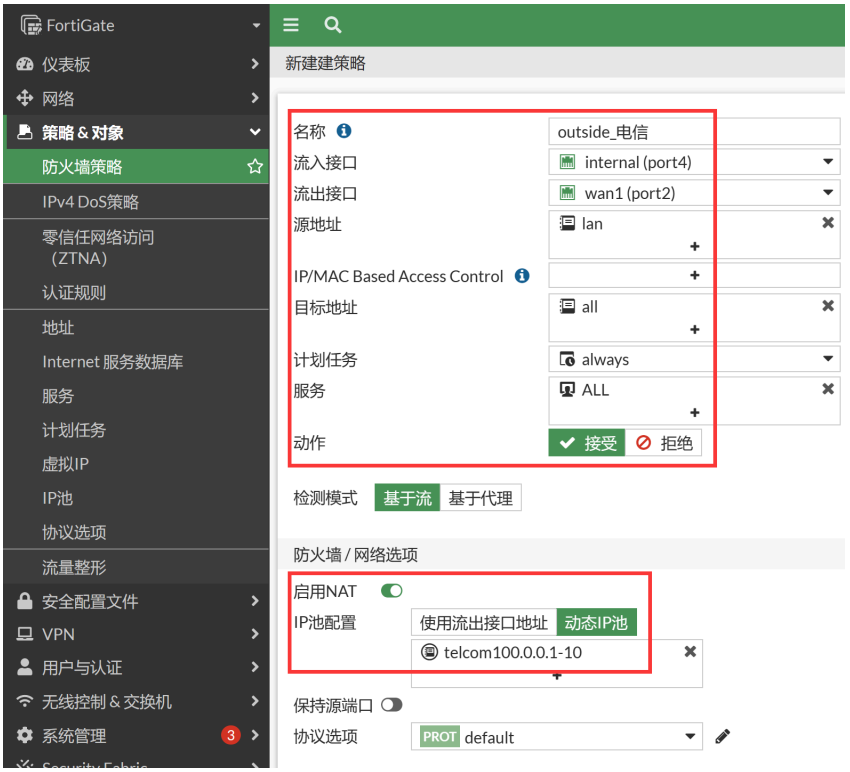


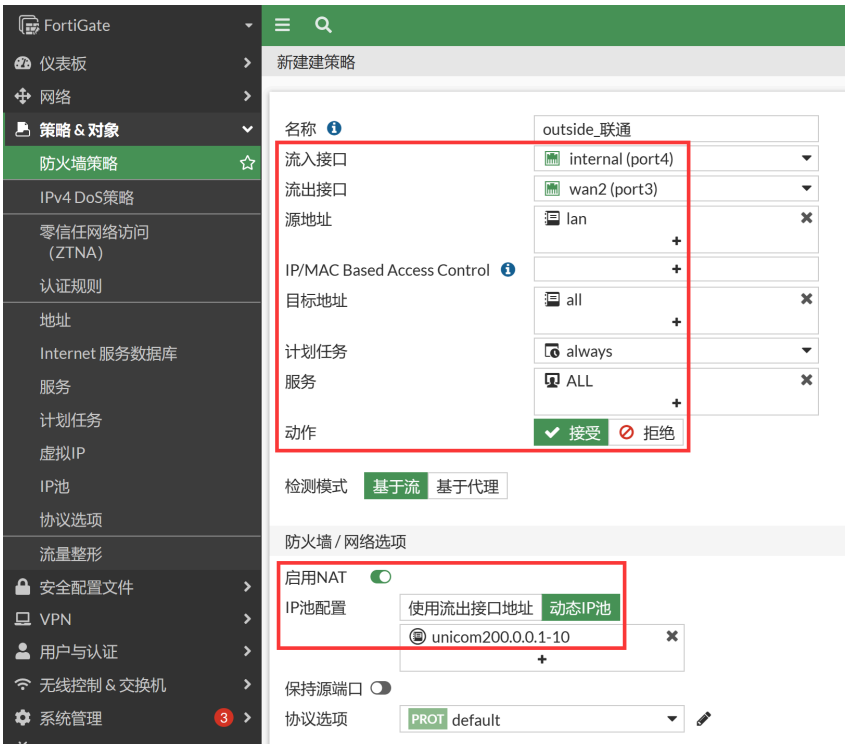


- 类型："超载"，动态的从地址池中分配
- 外部 IP 范围/子网：NAT 转换后的源地址池
- 地址解析协议回复：启用，地址池会回应 ARP 请求

```
config firewall ippool
    edit "telcom100.0.0.1-10"
        set startip 100.0.0.1
        set endip 100.0.0.10
    next
    edit "unicom200.0.0.1-10"
        set startip 200.0.0.1
        set endip 200.0.0.10
    next
end
```

5. 配置上网放通策略，进入策略 & 对象 → 防火墙策略，新建安全策略。分别创建 internal 到 wan1 和 internal 到 wan2 的上网策略。






```
config firewall policy
    edit 1
        set name "outside_电信"
        set srcintf "port4"
        set dstintf "port2"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set ippool enable
        set poolname "telcom100.0.0.1-10"
    next
    edit 2
```

```
        set name "outside_联通"
        set srcintf "port4"
        set dstintf "port3"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set ippool enable
        set poolname "unicom200.0.0.1-10"
    next
end
```

- 流入接口：选择 internal
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：wan1 或 wan2
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 勾选 "动态 IP 地址池"，并选择相应的地址池进行互联网访问
- 点击确定按钮后，系统自动保存配置，策略生效
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

结果验证

客户端上网测试，通过 tracert 命令，检测去往不同运营 IP 的路径是否正确。

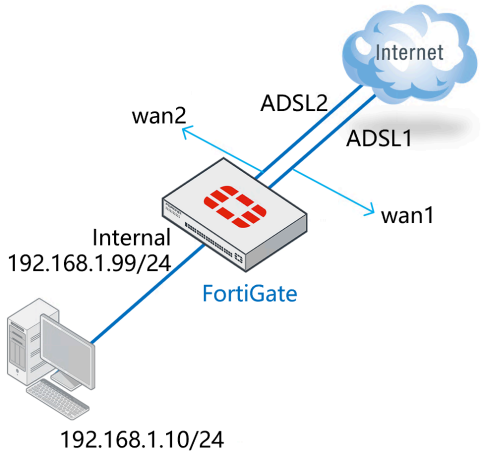
注意事项

1.**IPPool 配置中的外部 IP 范围不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 IPPool 后，IPPool 中的地址属于 FortiGate 自身的 IP 地址，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。** 2. 如果在 SD-WAN 中使用 ippool 的方式做 SNAT，一定要在 ippool 下配置绑定接口，防止 ippool 在 SD-WAN 成员中转换错误。

双线路 ADSL

1. 使用两条 ADSL 拨号接入互联网，内网为 192.168.1.0/24 网段。
2. 通过策略路由，192.168.1.128/25 网段单独使用一条 ADSL2 线路上网，其他人在 ADSL1 和 ADSL2 之间负载。

网络拓扑



配置要点

- wan1（port2）口、wan2（port3）口：接 ADSL 的接口，务必开启“从服务器重新获得网关”，这样 ADSL 拨号成功后设备会自动生成默认路由，无需手动配置默认路由
- internal（port4）口: IP 地址设置格式为：192.168.1.99/24，可选择性地开启接口的管理功能
- 配置地址对象 "lan" 内容为：192.168.1.0/24，"lan1" 内容为: 192.168.1.128/25。
- 配置从 internal 到 wan1 口的策略，并开启 NAT
- 配置从 internal 到 wan2 口的策略，并开启 NAT
- 配置策略路由，强制 lan1 地址段使用 wan2 口访问互联网

配置步骤

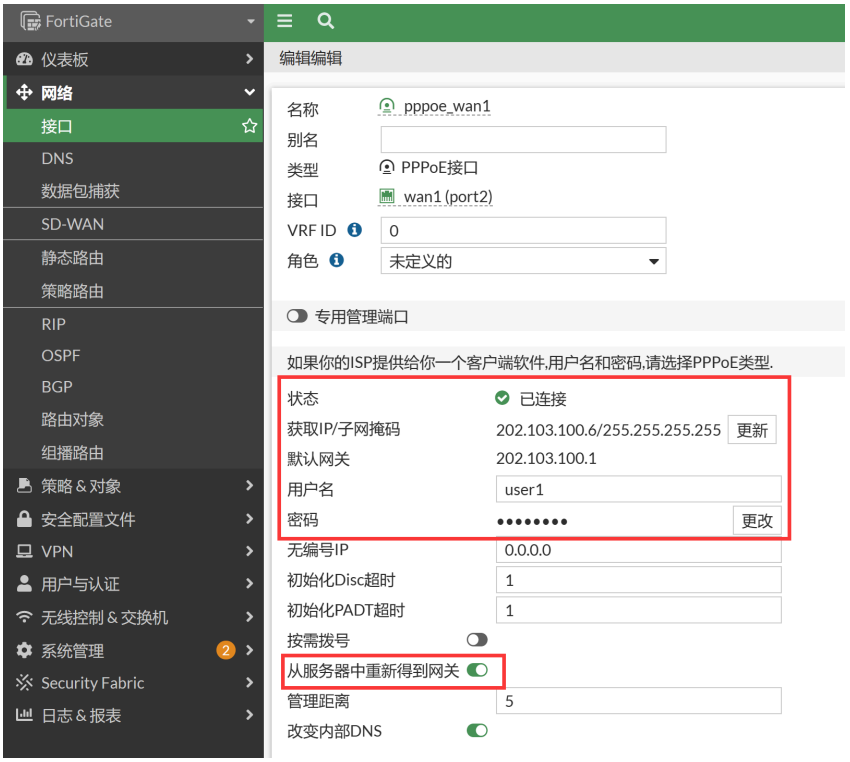
1. 配置 wan1 口和 wan2 口的 PPPoE，建议在 CLI 下配置 PPPoE 接口，使用此方式配置获取到 IP 后会自动添加默认路由到运营商。

> **重要**
>
> 建议通过 CLI 的方式新建 PPPoE 虚拟接口，参考：部署前注意事项 → PPPoE 接口 ↗配置章节。

```
config system pppoe-interface
    edit "pppoe_wan1"
        set device "port2"
        set username "user1"
        set password ENC
iPUwfhZvvT1T1+v3UPEiPkxLa7RArisivypVqKeSgnPpEGbxLuddF05fEn962jPWkP+PyJBPoKRJcigI+8qvIuzFdBwFW2Mn/EF3Dq6zkcQqTfBT=
    next
    edit "pppoe_wan2"
        set device "port3"
        set username "user2"
        set password ENC
1ZwM5cq6bHbQbaIYXw89RBaYTCvlGH3TVghDsO0OMj1WdIEi/jCtnSTdvG3WfL9Iy9/r0C+D/CGI4hplaN2I/ga3Fp2x/dXfeQnQ2Xf82A6FGZ1R=
    next
end

以上命令执行后，会自动在system interface下创建同名接口，其中包含“从服务器中重新得到网关”的配置
FortiGate # show full sys int pppoe
config system interface
    edit "pppoe_wan1"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set defaultgw enable
```

```
        set interface "port2"
    next
        edit "pppoe_wan2"
        set vdom "root"
        set mode pppoe
        set type tunnel
        set defaultgw enable
        set interface "port3"
    next
end
```

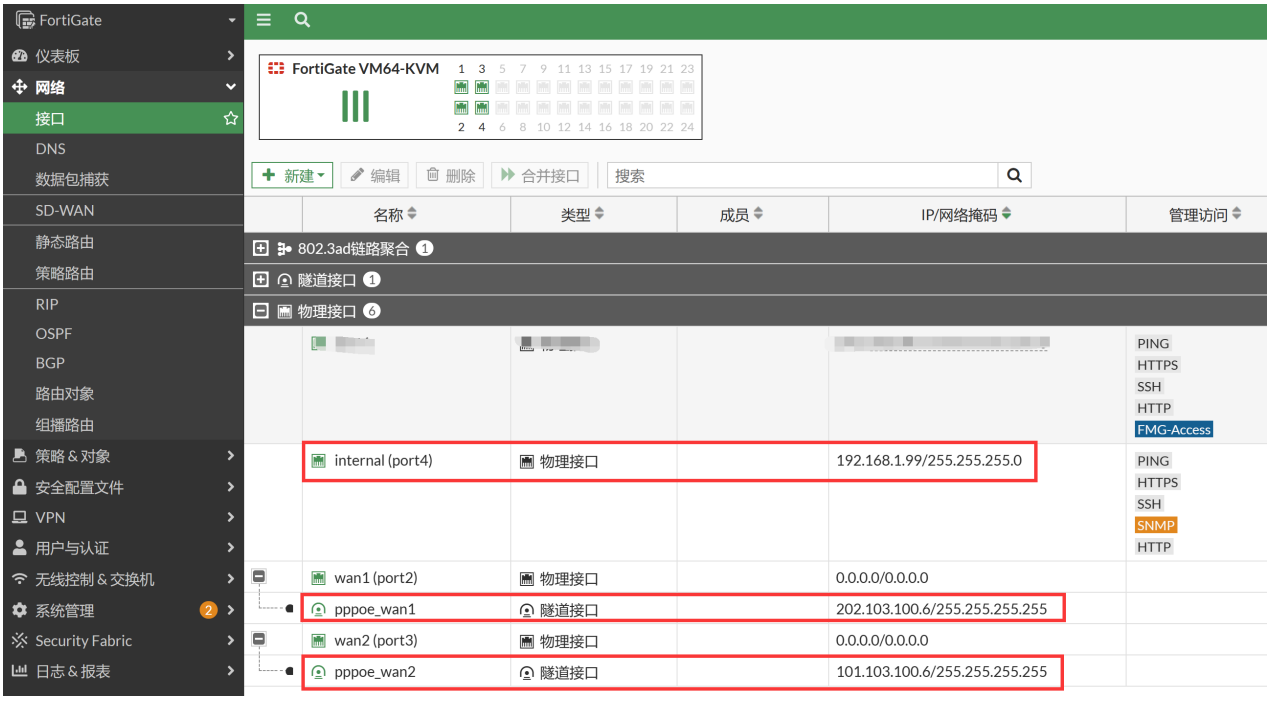
2. 以上配置完成后，可以在 GUI 上看到 wan1（port2）、wan2（port3）下分别多了个 PPPoE 接口，并获取到了运营商的 IP。可以看到“从服务器中重新得到网关”已开启，路由表中两个运营商的默认路由等价负载。



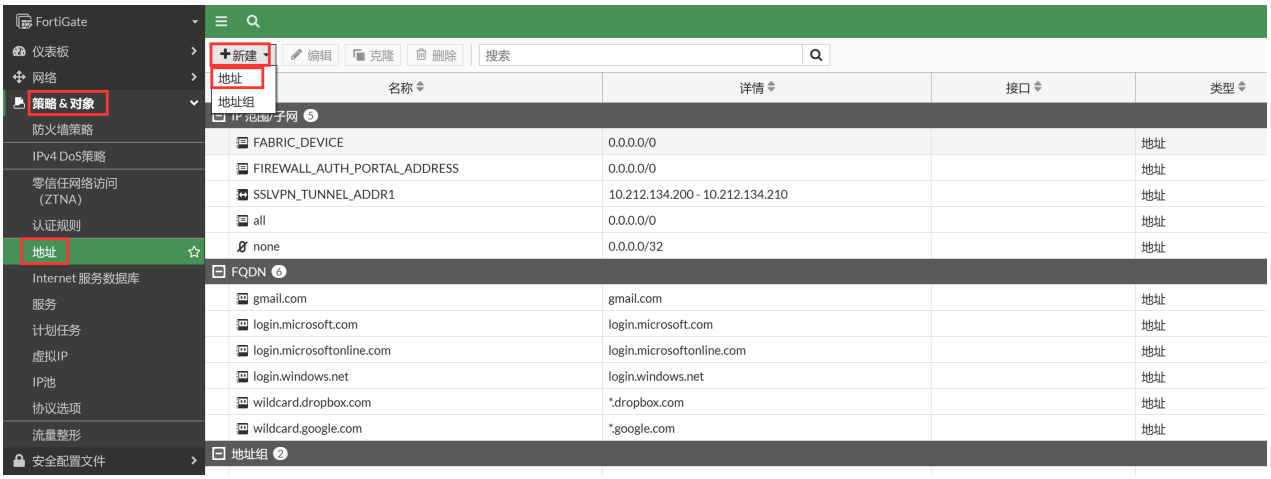


```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 101.203.100.1, pppoe_wan2, [1/0]
                  [5/0] via 202.103.100.1, pppoe_wan1, [1/0]
C       101.103.100.6/32 is directly connected, pppoe_wan2
C       101.203.100.1/32 is directly connected, pppoe_wan2
C       202.103.100.1/32 is directly connected, pppoe_wan1
C       202.103.100.6/32 is directly connected, pppoe_wan1
```

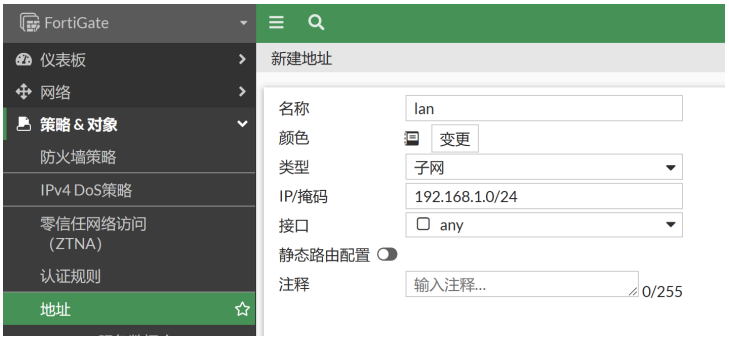
3. 配置 internal（port4）接口的 IP 为 192.168.1.99/24。选择性地开启接口的管理功能，建议内部开启 https、ssh、ping 服务。



4. 配置地址资源，进入防火墙 & 对象 → 地址，点击新建地址按钮。

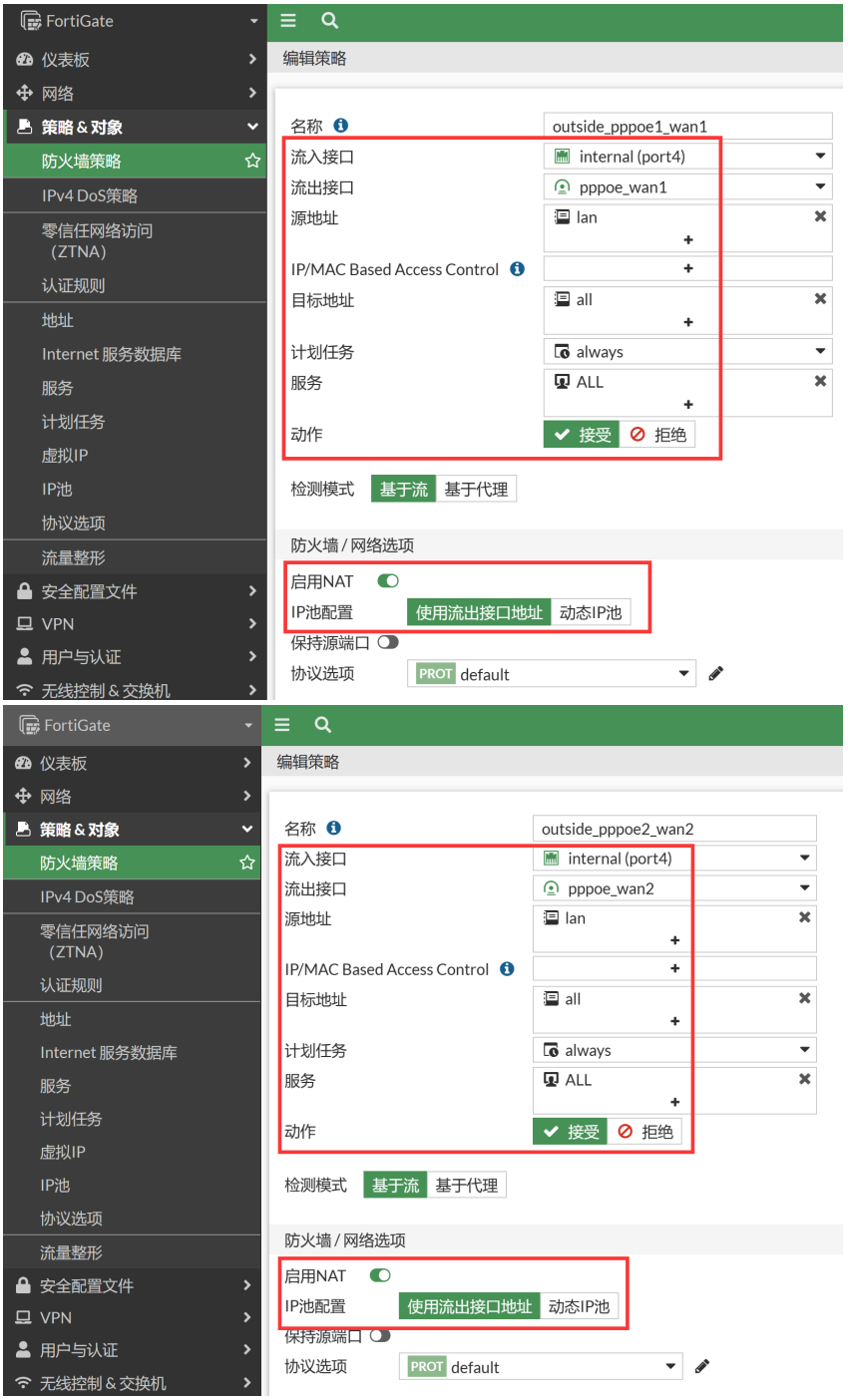


5. 名称配置为 lan，地址节点选择子网：192.168.1.0/24，点击确认。




```
config firewall address
    edit "lan"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

6. 分别为 wan1 和 wan2 配置防火墙策略，放通内网到 Internet 的流量，并开启 NAT，点击确定按钮后，系统自动保存配置，策略生效。




```
config firewall policy
    edit 1
        set name "outside_pppoe1_wan1"
        set srcintf "port4"
        set dstintf "pppoe_wan1"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set name "outside_pppoe2_wan2"
        set srcintf "port4"
        set dstintf "pppoe_wan2"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
```

```
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

- 流入接口：internal（port4）
- 源地址：选择刚才定义的地址资源 lan 上网网段
- 流出接口：pppoe 接口
- 目的地址选择: all，代表所有的地址
- 时间表：always
- 服务: ALL
- 动作：ACCEPT
- NAT：选择 "启用 NAT"， 系统会自动将内网的 lan 地址段 ip，转换为 wan1 接口地址，进行互联网访问
- 注意：启用 "记录允许流量（记录流日志）" 将会给系统带来额外的资源消耗，所以非必要情况下请不要启用记录日志

7. 配置策略路由，强制 192.168.1.128/25 网段单独使用一条 ADSL2 线路上网（需要在系统管理 → 可见功能中开启高级路由）。其他地址可以在 ADSL1 和 ADSL 两条链路上负载。




```
config router policy
    edit 1
        set input-device "port4"
        set src "192.168.1.128/255.255.255.128"
        set dstaddr "all"
        set output-device "pppoe_wan2"
    next
end
```

- 协议端口：协议类型，0 为所有任何协议，可以指定 6 tcp,17 udp,132 sctp 等
- 流入接口：流量进入接口 ，internal
- 源地址掩码：数据包的源地址，192.168.1.128/25
- 目的地址掩码：数据包的目的地址, 默认全部
- 目的端口：目的端口，默认为所有。从 1-65536
- 动作：流量采用的动作
- 流出接口：数据包的流出接口，pppoe_wan2

- 网关地址：流出接口的下一条网关地址， 由于是 PPPoE 拨号，不写下一跳地址，防火墙会根据静态路由表中的条目，自动使用 PPPoE 生成的静态路由的下一跳 IP

结果验证

1. 将电脑 IP 地址设置为 192.168.1.10/24，网关设置地为 192.168.1.99，DNS 配置为当地的 DNS 即可。电脑可通过 wan1 以及 wan2 访问互联网。

2. 更改主机 IP 为 192.168.1.168/24，主机只能通过 wan2 口访问互联网。

3. 排障时可使用 tracert 命令，或者抓包命令来查看数据包流出接口。

```
FortiGate#diagnose sniffer packet any 'icmp and host x.x.x.x' 4    //x.x.x.x为测试目标地址
```

虚拟 IP 映射

- 端口映射应原理
- 地址映射（一对一 IP 映射）
- 端口映射（一对多端口映射）
- 多线路端口映射
- VIP 映射的主机访问 Internet
- 内网主机通过公网 IP 访问 VIP
- VIP46 映射
- VIP64 映射

端口映射应原理

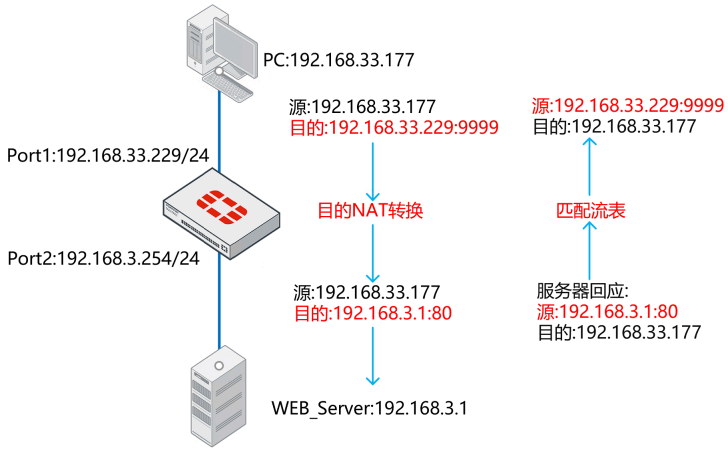
功能原理

数据从 FortiGate 通过时的处理流程：

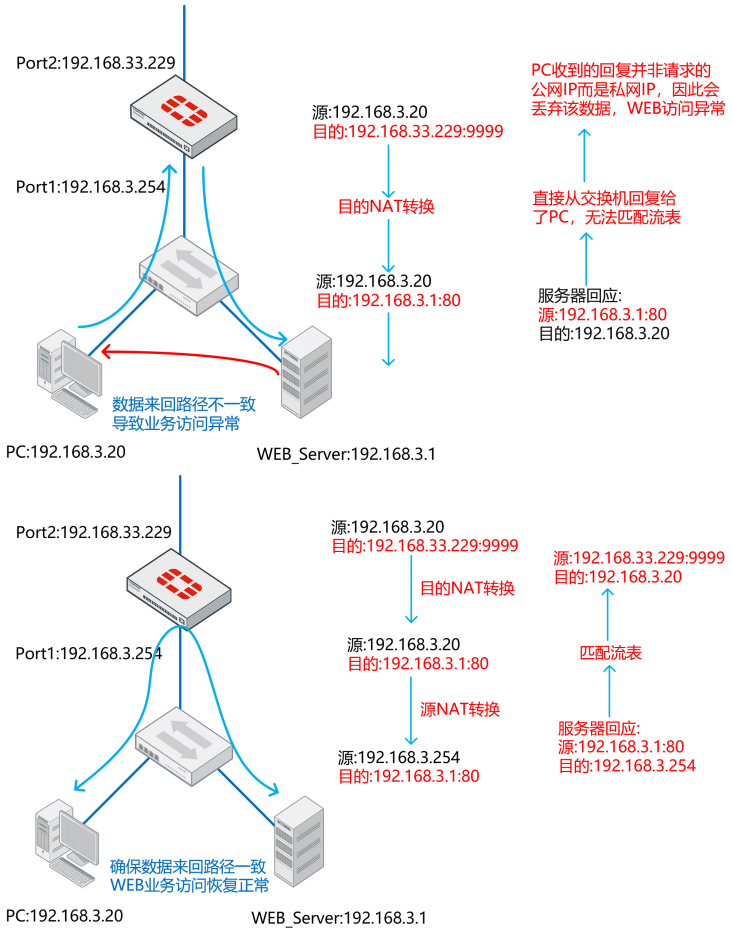
1. 入接口收到数据包 → 新建流表或匹配流表 → 匹配目的 NAT → 查路由表 → 匹配安全策略 → 数据向出口转发 → 匹配源 NAT → 出接口转发数据包。
2. 源 NAT 转换通俗地讲即将源地址转换为另一个地址，目的地址不变。局域网的私网地址需要将源地址转换为公网地址才能访问外网。
3. 目的 NAT 转换通俗地讲即将目的地址转换为另一个地址，源地址不变。公网地址访问私网地址的服务器首先是将数据发给服务器所在的局域网的公网地址，再由设备将目的地址转换为服务器地址。

端口映射流程举例

1. 外网地址 192.168.33.177 访问服务器 192.168.3.1 的过程为：
   - 外网访问服务器的源 IP 为 192.168.33.177，目的地址为 192.168.33.229，目的端口为 9999，经过 FortiGate 的目的 NAT 转换，将目的 IP 转换为 192.168.3.1，目的端口转换为 80。
   - 服务器回应时源 IP 为 192.168.3.1，源端口为 80，目的地址为 192.168.33.177，匹配刚才目的 NAT 转换所建的流表将源 IP 地址转换为 192.168.33.229，源端口转换为 9999。

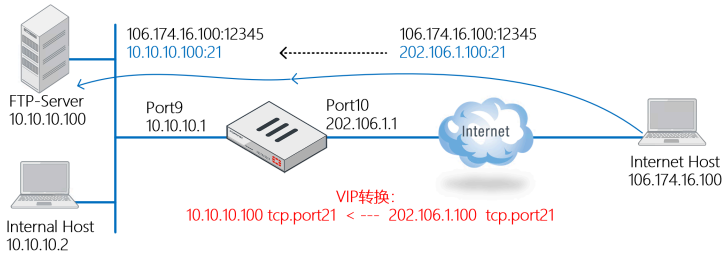


2. 内网用户 192.168.3.20 通过公网地址 192.168.33.229 访问服务器的数据过程需保证来回路径一致，所以需要配置源 NAT 转换也需要配置目的 NAT 转换：
   - 内网 PC 发出的源 IP 为 192.168.3.20，目的 IP 为 192.168.33.229，数据到 FortiGate 后，将源 IP 转换为内网口 IP 地址 192.168.3.254，目的 IP 转换为 192.168.3.1（注意目的端口也由 9999 变为 80）。
   - 服务器收到数据后回应的源 IP 为 192.168.3.1，目的 IP 为 192.168.3.254，数据到 FortiGate 后，匹配刚才源 NAT 转换和目的 NAT 转换的流表，将 192.168.3.1 变为 192.168.33.229，192.168.3.254 变为 192.168.3.20，则此时源 IP 地址变为 192.168.33.229（注意源端口也由 80 变为 9999），目的 IP 地址变为 192.168.3.20，发送给内网 PC。



配置举例

网络拓扑



配置步骤

1. 新建 VIP 映射，将公网地址从 port10 映射到内部的 FTP 服务器私网 IP。

```
config firewall vip
    edit VIP-FTP-server-port10
        set extip 202.106.1.100
        set extintf "port10"
        set portforward enable
        set mappedip 10.10.10.100
        set extport 21
        set mappedport 21
end
```

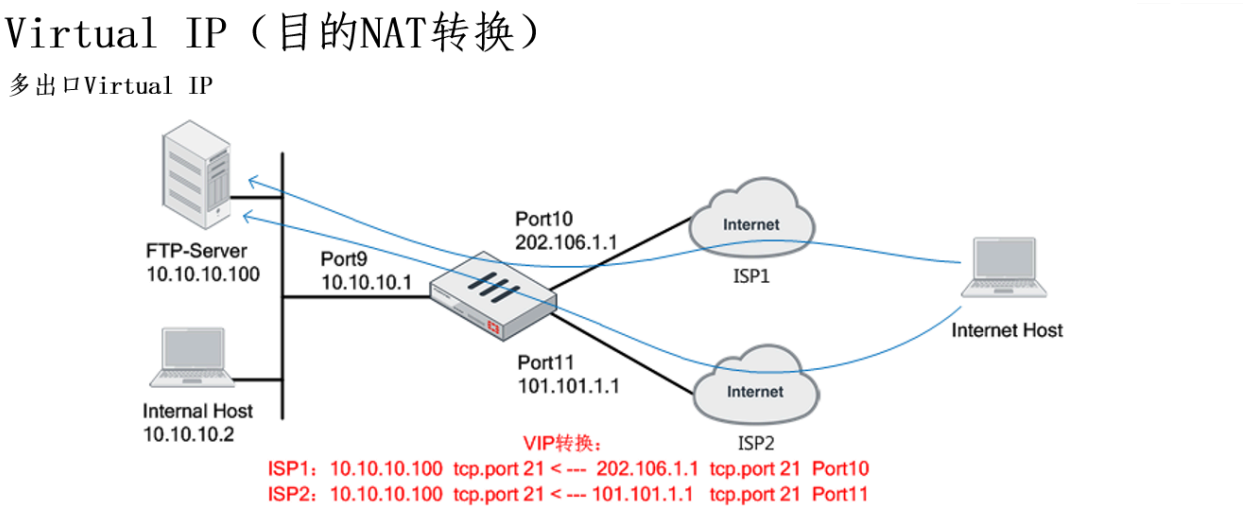
2. 创建安全策略引用该 VIP。

```
config firewall policy
    edit 0
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr VIP-ETP-Server-port10
        set action accept
        set schedule "always"
        set service "ALL"
end
```

3. 使用 Internet PC 访问 VIP 的 FTP 服务，查看会话表项。

```
FG1000D # get system session list
PROTO   EXPIRE SOURCE           SOURCE-NAT          DESTINATION       DESTINATION-NAT
tcp     3551   106.174.16.100:4063 -                202.106.1.100:21  10.10.10.100:21  //Dnat转换session
icmp    59     10.10.10.100:512 202.106.1.1:62464  202.106.1.2:8     -                // 服务器OUT 走普通Snat
icmp    60     10.10.10.100:512 202.106.1.1:62464  106.174.16.100:8  -
FG1000D # diagnose sys session list
session info: proto=6 proto_state=01 duration=285 expire=3315 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty
statistic(bytes/packets/allow_err): org=128/3/1 reply=130/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=17->16/16->17 gwy=10.10.10.100/202.106.1.2
hook=pre dir=org act=dnat 106.174.16.100:4063->202.106.1.100:21(10.10.10.100:21)
hook=post dir=reply act=snat 10.10.10.100:21->106.174.16.100:4063(202.106.1.100:21)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=015d9e27 tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
npu_state=00000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0
```

4. 多出口 Virtual IP 的场景。



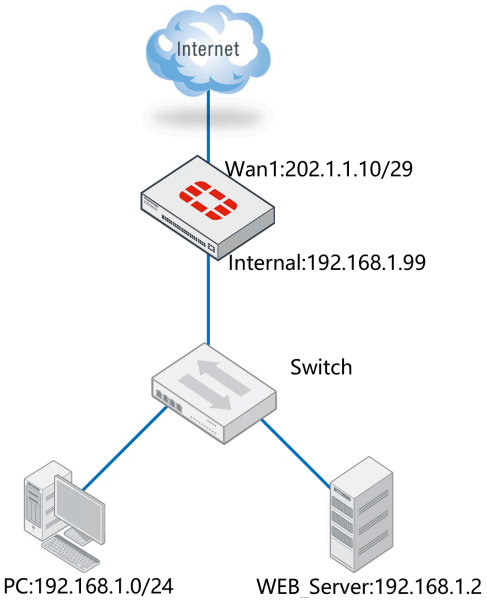
双出口的VIP配置只需正常添加另外一条VIP条目和另外一个策略即可，但要特别主要在双出口环境下的RPF（反向路由检查）问题，如果双出口做负载均衡的情况下无需注意RPF检查问题，但是如果想要做浮动路由主备，需修改路由的优先级大小而非管理距离大小，以保证两条默认路由同时存在于路由表中。

地址映射（一对一 IP 映射）

网络需求

用户已完成了防火墙基础配置，现在需要将内网的一台 web 服务器需要将 HTTPS 服务全映射到外网口的地址，让外网的用户能访问到此服务器。

网络拓扑

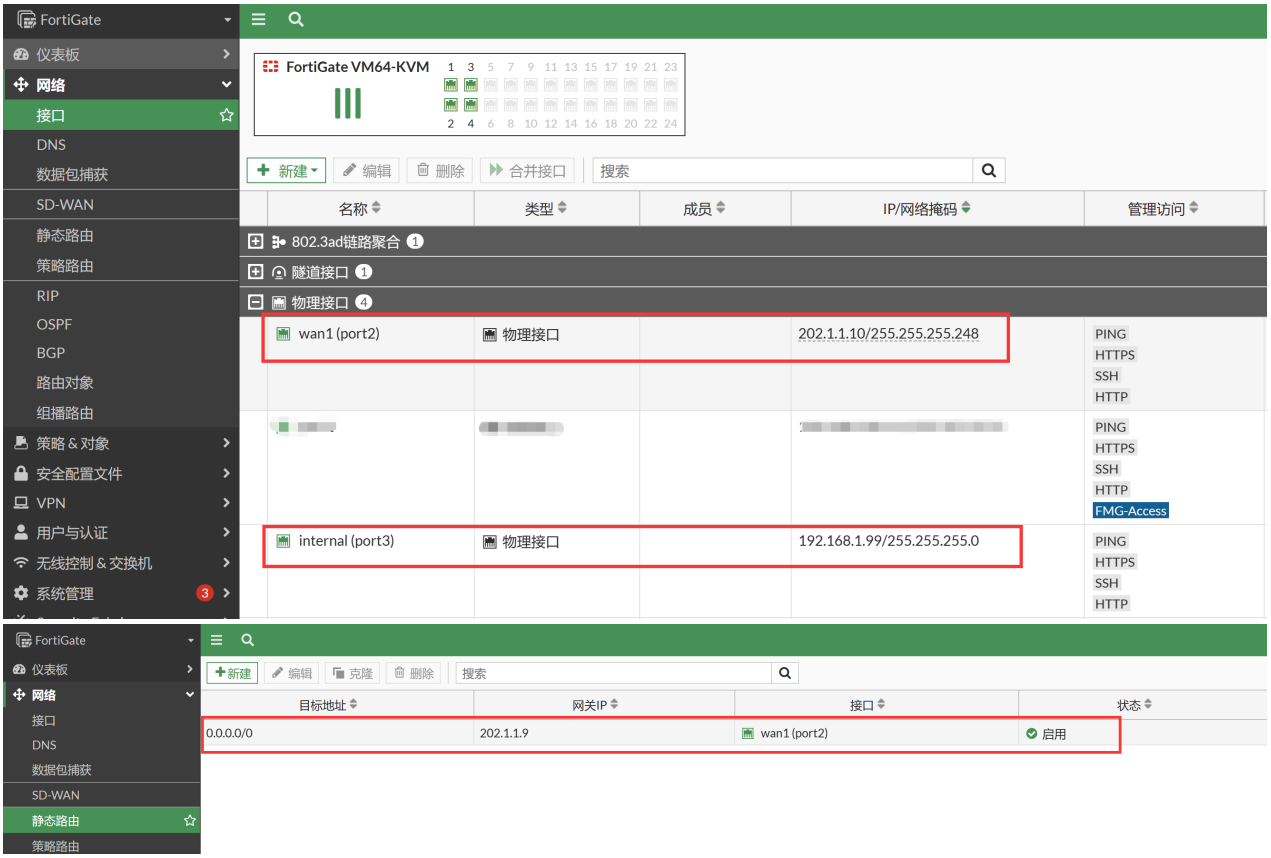


配置要点

- 基础上网配置
- 配置虚拟 IP（DNAT）
- 配置安全策略

配置步骤

1. 基础上网配置，请参照网络管理 → 路由模式 → 静态地址上网方式↗章节，接口 IP、静态路由配置如下。



2. 进入策略 & 对象 → 虚拟 IP，新建虚拟 IP。

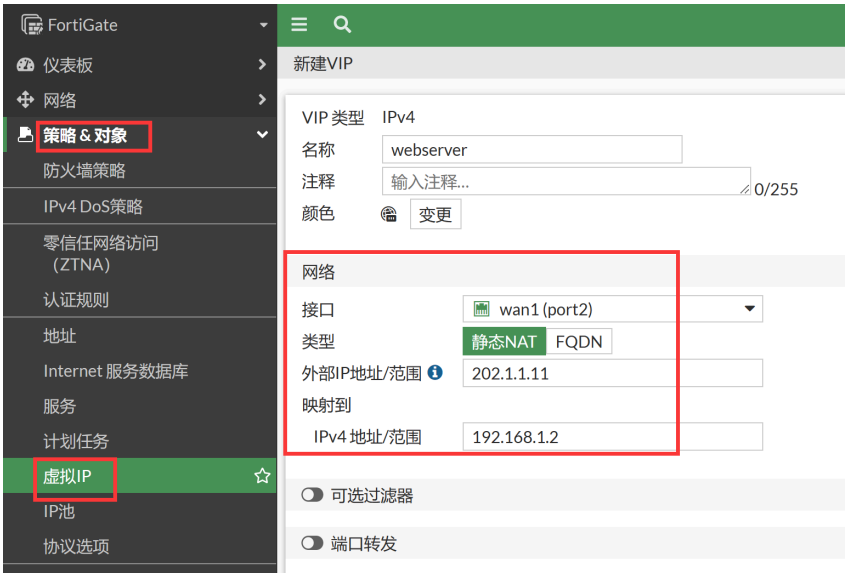
3. 配置 VIP，名称为 webserver，用于 wan1 接口。

重要

VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。

重要

强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。

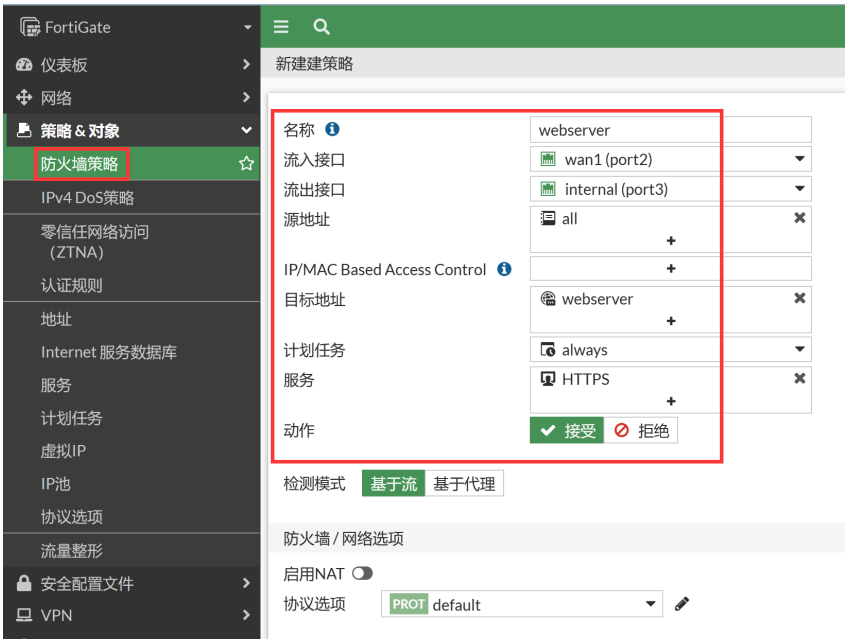


```
config firewall vip
    edit "webserver"
        set extip 202.1.1.11
        set mappedip "192.168.1.2"
        set extintf "port2"
    next
end
```

重要

“外部的 IP 地址/范围”和“映射的 IPv4 地址/范围”数量是对应，一对一映射，可以只写一个 IP 对一个 IP 的映射关系，也可以写一个映射范围。比如 202.1.1.3-202.1.1.10，内部映射必须是 192.168.1.2-192.168.1.9，对应映射关系也是对应的，202.1.1.3 对应 192.168.1.2，202.1.14 对应 192.168.1.3，依此类推。

4. 配置安全策略引用 VIP。




```
config firewall policy
    edit 1
        set name "webserver"
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "webserver"
        set schedule "always"
        set service "HTTPS"
    next
end
```

- 流入接口：wan1 //如内网用户也需通过虚拟 IP 访问，需将此处设置为 any
- 源地址：all
- 流出接口： internal
- 目的地址: webserver //定义好的虚拟 IP 映射对象
- 服务: HTTPS //只允许进行 HTTPS 服务访问
- 注意：如内网用户也需通过虚拟 IP 访问，有 2 种方式可实现：
  - 需将原策略的【流入接口】设置为 any，并在该策略上开启 NAT
  - 或者添加一条【流入接口】为 internal 的内到内策略，并在该策略上开启 NAT
- NAT：根据需求开启，如果内网服务器到源客户端公网 IP 没有路由可达，仅有到达 FortiGate 内网口的路由，则需要开启 NAT，在做 DNAT 之后再做一次源 NAT，将源 IP 转换为 FortiGate 内网口 IP
- 动作：接受

结果验证

从外部进行访问 https://202.1.1.11 ↗ ，如果需要测试映射是否有效可以在策略的服务临时添加 ping 服务进行测试。

注意事项

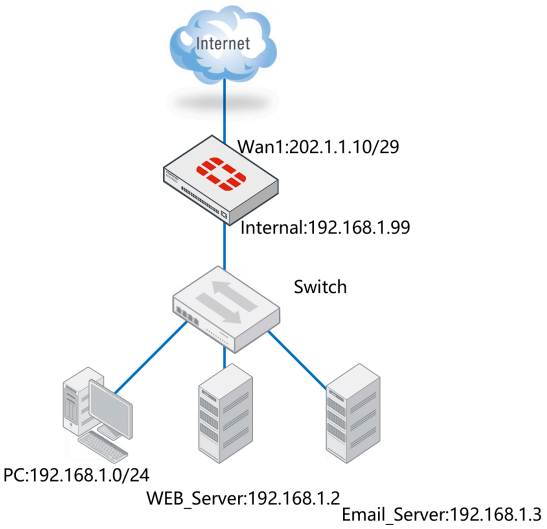
1.**VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。** 2. 强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。

端口映射（一对多端口映射）

网络需求

1. 现在需要将内网的一台 web 服务器（192.168.1.2）的 80 端口映射到外网口的地址（202.1.1.11）的 8080 端口（内网端口与外网映射端口不一致）。
2. 将内网的一台 email 服务器（192.168.1.3）的 25 端口映射到外网口的地址（202.1.1.11）的 25 端口。
3. 案例意义：了解 FortiGate 的关键功能处理模块的匹配顺序：DNAT → 路由 → 安全策略 → 源 NAT。

网络拓扑

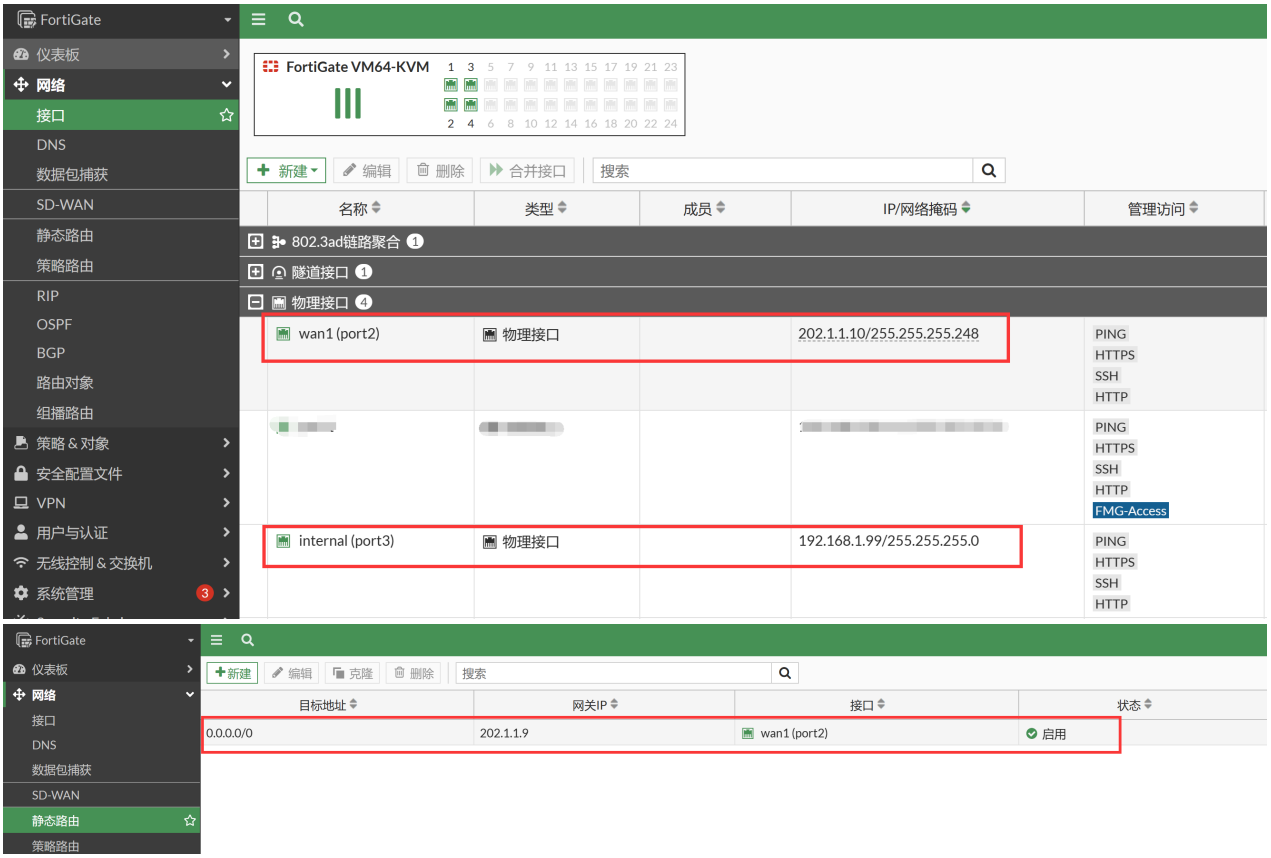


配置要点

- 基础上网配置
- 配置虚拟 IP（DNAT）
- 配置安全策略

配置步骤

1. 基础上网配置，请参照网络管理 → 路由模式 → 静态地址上网方式 ↗章节，接口 IP、静态路由配置如下。



2. 进入策略 & 对象 → 虚拟 IP，新建虚拟 IP。

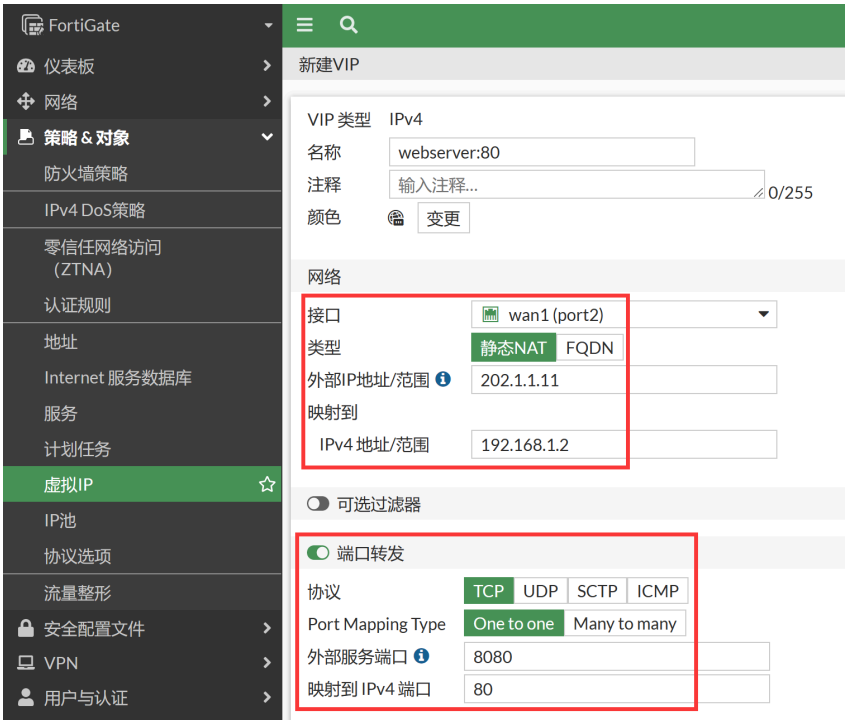
3. 配置 VIP1，名称为 webserver:80，对 HTTP 服务器进行映射，用于 wan1 接口。

重要

VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。

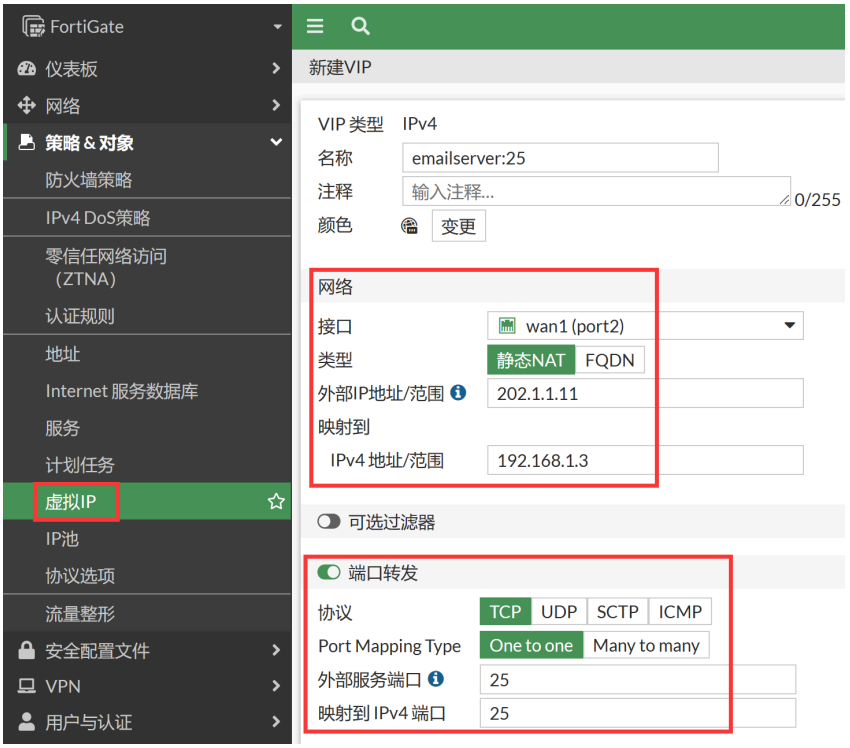
重要

强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。



4. 配置 VIP2，名称为 emailserver:25，对 smtp 服务器进行映射，用于 wan1 接口。

> **重要**
>
> 建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。



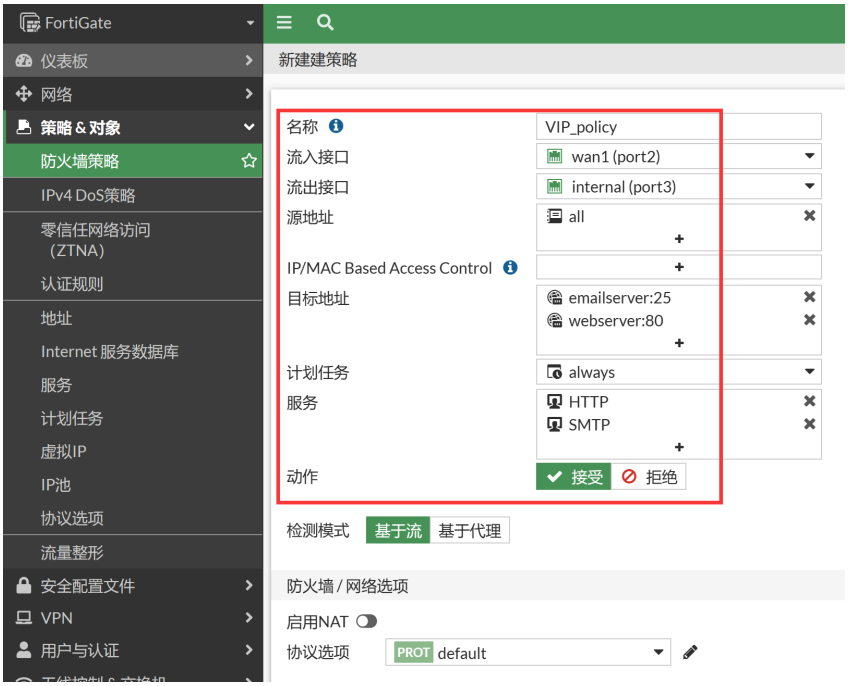

```
config firewall vip
    edit "webserver:80"
        set extip 202.1.1.11
        set mappedip "192.168.1.2"
        set extintf "port2"
        set portforward enable
        set extport 8080
        set mappedport 80
    next
```

```
    edit "emailserver:25"
        set extip 202.1.1.11
        set mappedip "192.168.1.3"
        set extintf "port2"
        set portforward enable
        set extport 25
        set mappedport 25
    next
end
```

重要

“外部的 IP 地址/范围”和“映射的 IPv4 地址/范围”数量是对应，一对一映射，可以只写一个 IP 对一个 IP 的映射关系，也可以写一个映射范围。比如 202.1.1.3-202.1.1.10，内部映射必须是 192.168.1.2-192.168.1.9，对应映射关系也是对应的，202.1.1.3 对应 192.168.1.2，202.1.1.14 对应 192.168.1.3，依此类推

5. 配置安全策略引用两个 VIP。




```
FortiGate # show firewall policy
config firewall policy
    edit 1
        set name "VIP_policy"
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "emailserver:25" "webserver:80"
        set schedule "always"
        set service "HTTP" "SMTP"
    next
end
```

- 流入接口：wan1 //如内网用户也需通过虚拟 IP 访问，需将此处设置为 any
- 源地址：all
- 流出接口： internal
- 目的地址: webserver:80 和 emailserver:25 //定义好的虚拟 IP 映射对象
- 服务: HTTP 和 SMTP //只允许进行 HTTP 和 SMTP 服务访问
- 注意：如内网用户也需通过虚拟 IP 访问，有 2 种方式可实现：
  - 需将原策略的【流入接口】设置为 any，并在该策略上开启 NAT
  - 或者添加一条【流入接口】为 internal 的内到内策略，并在该策略上开启 NAT
- NAT：根据需求开启，如果内网服务器到源客户端公网 IP 没有路由可达，仅有到达 FortiGate 内网口的路由，则需要开启 NAT，在做 DNAT 之后再做一次源 NAT，将源 IP 转换为 FortiGate 内网口 IP
- 动作：接受

结果验证

1. 从外部进行访问 http://202.1.1.11 ↗ 。
2. 从外部访问 202.1.1.11 的 SMTP 服务。

注意事项

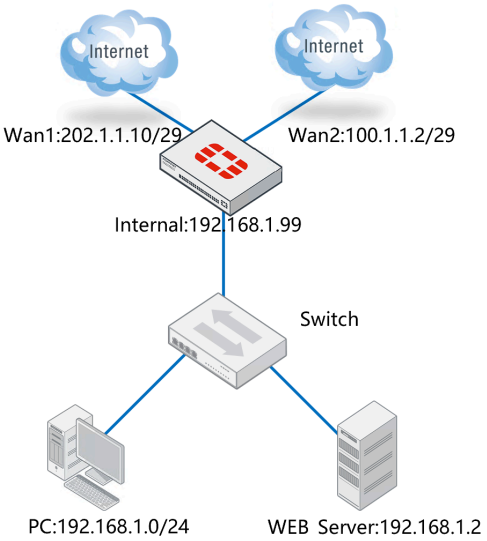
1.**VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。** 2. 强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP、VIP 映射的服务器在做 SNAT 时源地址转换错误等网络问题。

多线路端口映射

网络需求

- 将内网中的一台 WEB 服务器，分别映射到电信和联通 2 个互联网出口的公网 IP 上：
  - WEB 服务器地址： 192.168.1.2/24 网关：192.168.1.99
  - 电信出口: 202.1.1.2/29 , 网关 202.1.1.1，服务器公网地址 202.1.1.3
  - 联通出口: 100.1.1.2/29 , 网关 100.1.1.1，服务器公网地址 100.1.1.3
  - 内部网段 192.168.1.0/24 需要上互联网
- 案例意义：
  - FortiGate 支持数据流源进源出
  - 防火墙会对会话进行跟踪，从电信接口进入的访问，会优先从电信接口返回
  - 从联通接口进入的访问，会优先从联通接口返回
  - 条件是防火墙的路由表内，分别拥有能够匹配返回数据流的路由表，所以只需要配置电信和网通的两条默认路由即可

网络拓扑

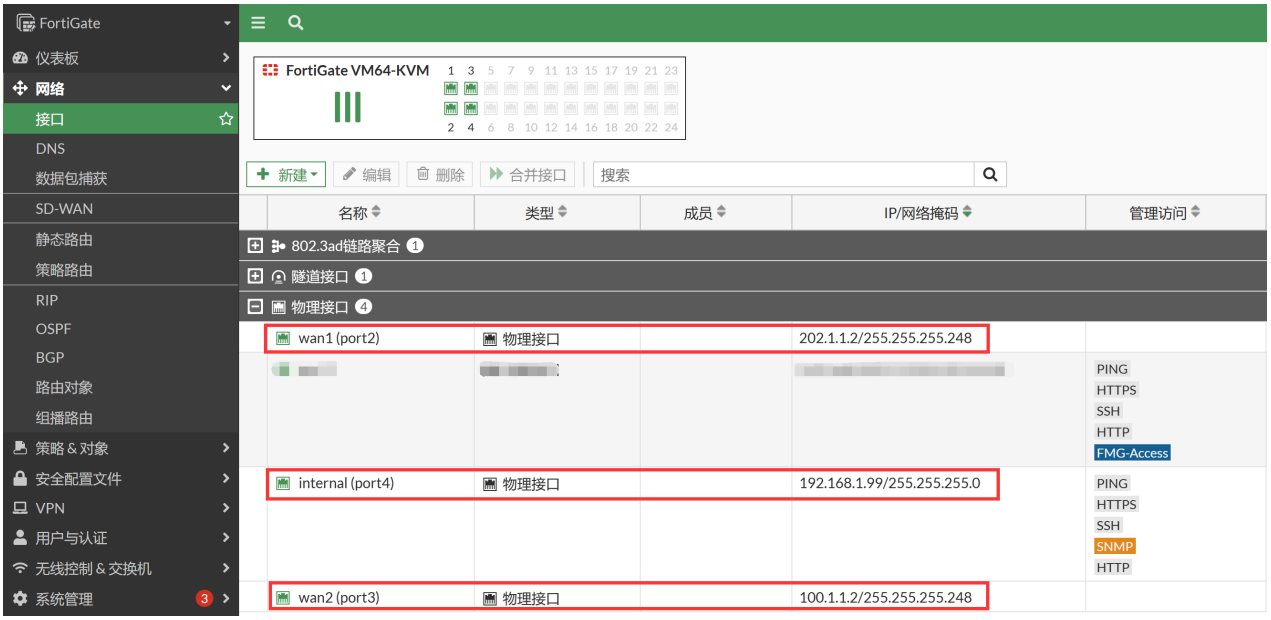


配置要点

- 配置接口 IP
- 配置虚拟 IP（DNAT）
- 配置安全策略

配置步骤

1. 基础上网配置，请参照策略与对象 → 多线路上网配置 → 双线路同运营商上网配置 ↗章节，接口 IP、静态路由配置如下。





> **重要**
>
> 防火墙会对会话进行跟踪，从电信接口进入的访问，会优先从电信接口返回；从联通接口进入的访问，会优先从联通接口返回。条件是防火墙的路由表内，分别拥有能够匹配返回数据流的路由表。所以只需要配置电信和网通的两条默认路由即可。

2. 查看路由表，存在两条等价默认路由。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port3, [1/0]
                  [10/0] via 202.1.1.1, port2, [1/0]
C       100.1.1.0/29 is directly connected, port3
C       192.168.1.0/24 is directly connected, port4
C       202.1.1.0/29 is directly connected, port2
```

3. 进入策略 & 对象 → 虚拟 IP，新建虚拟 IP。

> **重要**
>
> 建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。

4. 配置 VIP1，名称为 web1，对 HTTP 服务器进行映射，用于 wan1（电信）接口。

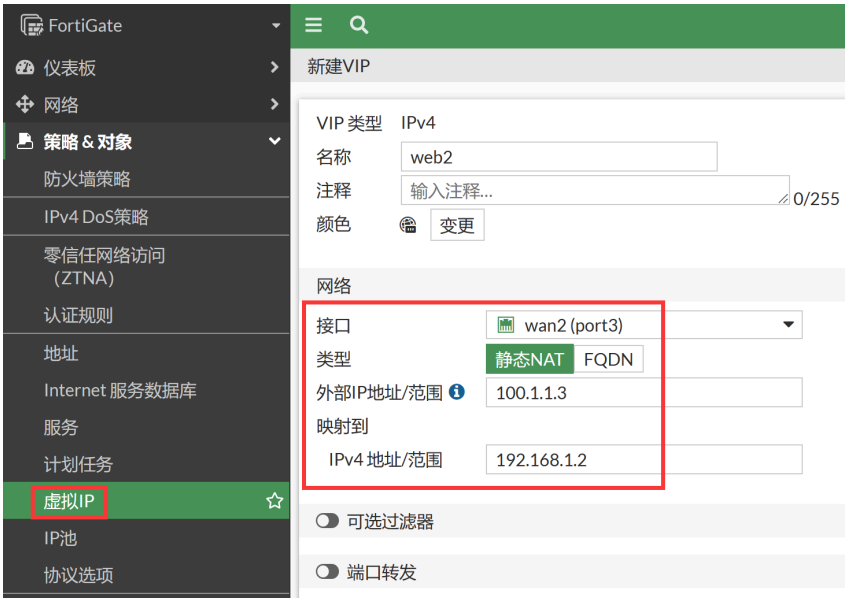
> **重要**
>
> VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。

重要

强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。



5. 配置 VIP2，名称为 web2，对 HTTP 服务器进行映射，用于 wan2（联通）接口。

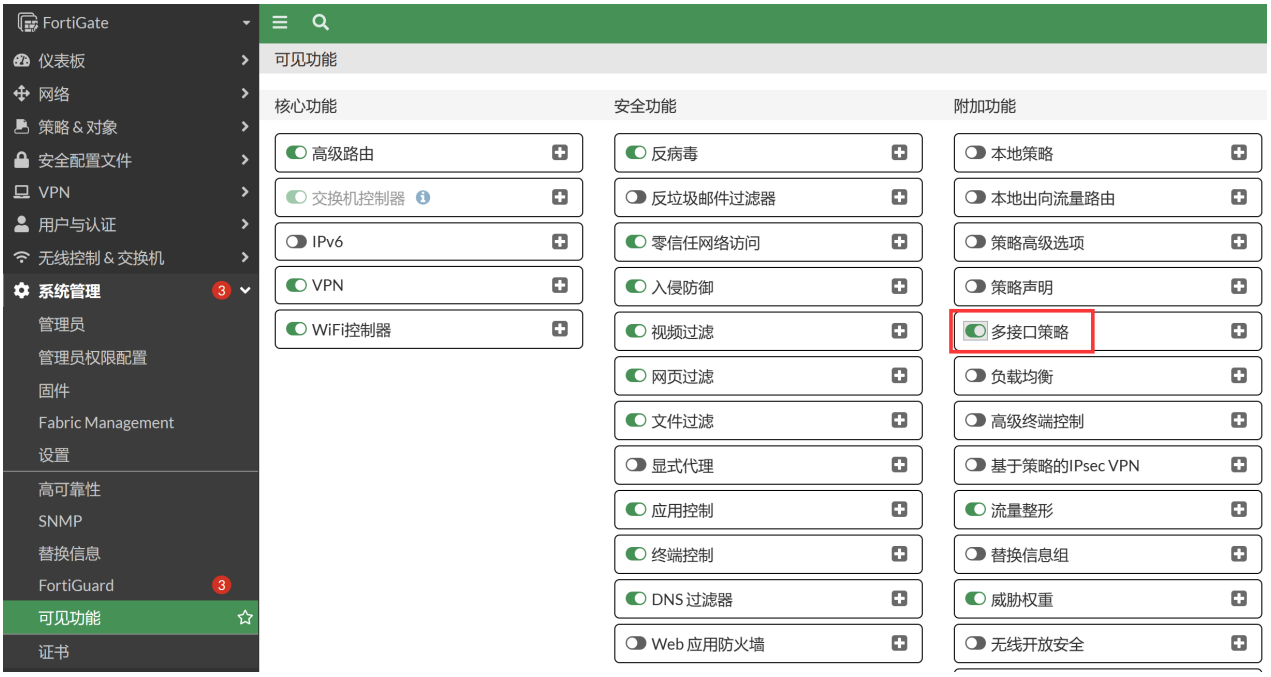



```
config firewall vip
    edit "web1"
        set extip 202.1.1.3
        set mappedip "192.168.1.2"
        set extintf "port2"
    next
    edit "web2"
        set extip 100.1.1.3
        set mappedip "192.168.1.2"
        set extintf "port3"
    next
end
```

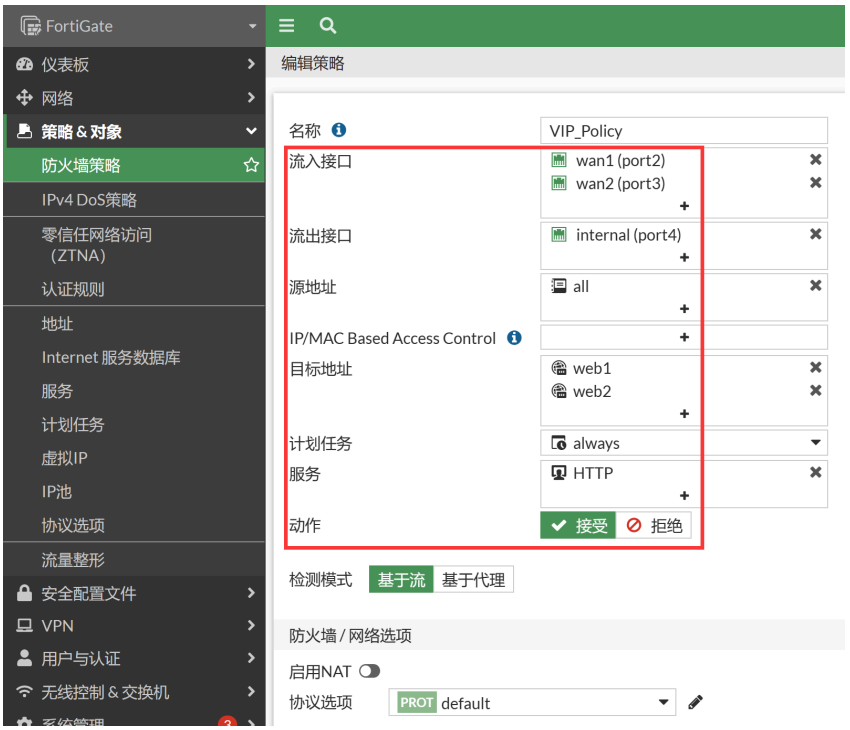
重要

“外部的 IP 地址/范围”和“映射的 IPv4 地址/范围”数量是对应，一对一映射，可以只写一个 IP 对一个 IP 的映射关系，也可以写一个映射范围。比如 202.1.1.3-202.1.1.10，内部映射必须是 192.168.1.2-192.168.1.9，对应映射关系也是对应的，202.1.1.3 对应 192.168.1.2，202.1.14 对应 192.168.1.3，依此类推。

6. 配置安全策略引用两个 VIP，默认配置下，两条链路需要针对两个 VIP 配置两条防火墙策略，这里我们可以通过一条策略实现，打开系统管理 → 可见功能，开启“多接口策略”功能，这样我们就可以在防火墙策略的流入接口和流出接口中选择多个接口。



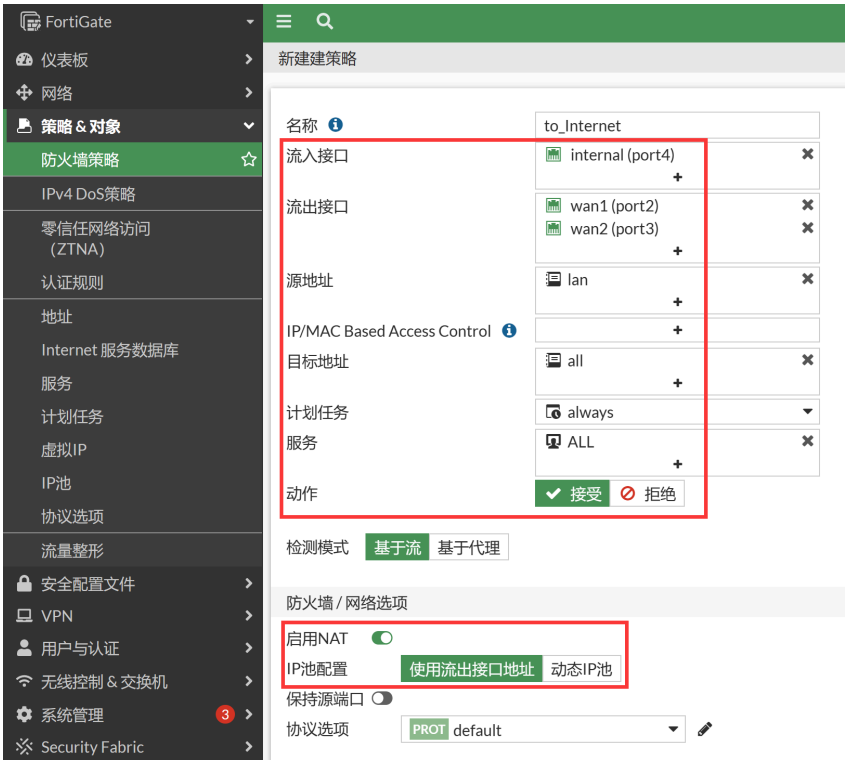
7. 配置安全策略引用两个 VIP，入接口可以选择多个接口，同时选择 wan1 和 wan2。




```
config firewall policy
    edit 1
        set name "VIP_Policy"
        set srcintf "port2" "port3"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "web1" "web2"
        set schedule "always"
        set service "HTTP"
    next
end
```

- 流入接口：wan1、wan2 //如内网用户也需通过虚拟 IP 访问，需将此处设置为 any
- 源地址：all
- 流出接口： internal
- 目的地址: web1 和 web2 //定义好的虚拟 IP 映射对象
- 服务: HTTP //只允许进行 HTTP 服务访问
- 注意：如内网用户也需通过虚拟 IP 访问，有 2 种方式可实现：
  - 需将原策略的【流入接口】设置为 any，并在该策略上开启 NAT
  - 或者添加一条【流入接口】为 internal 的内到内策略，并在该策略上开启 NAT
- NAT：根据需求开启，如果内网服务器到源客户端公网 IP 没有路由可达，仅有到达 FortiGate 内网口的路由，则需要开启 NAT，在做 DNAT 之后再做一次源 NAT，将源 IP 转换为 FortiGate 内网口 IP
- 动作：接受

8. 配置内网网段访问两条互联网线路的放通策略。




```
config firewall policy
    edit 2
        set name "to_Internet"
        set srcintf "port4"
        set dstintf "port2" "port3"
        set action accept
        set srcaddr "lan"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

1. 分别从 2 个运营商接口访问 202.1.1.3 和 100.1.1.3 两个 IP 的 80 端口。
2. 内网用户可以通过两条线路访问 Internet。

注意事项

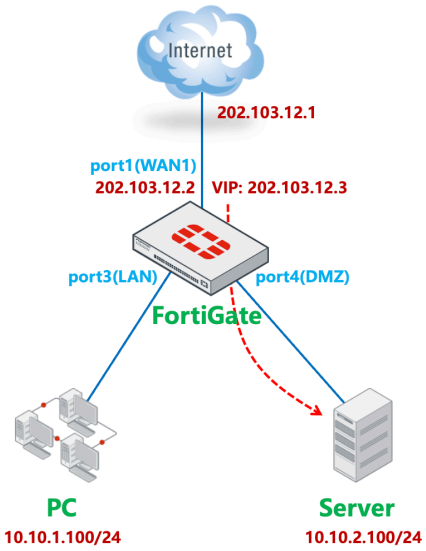
1. VIP 配置中的外部 IP 不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 后，extip 属于 FortiGate 自身的 IP 地址，即使 VIP 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。
2. 强烈建议在配置 VIP 时，选择绑定接口，这样可以防止 VIP 在其他接口响应 ARP 导致的网络问题。

VIP 映射的主机访问 Internet

网络需求

某用户使用 VIP 一对一映射了一个公网 IP 到内网服务器，同时又需要此服务器通过 FortiGate 的 SNAT 访问 Internet。

网络拓扑



1. 在 port1（WAN1）上通过 VIP 将 202.103.12.3 映射到内网 Server 10.10.2.100。
2. 内网 PC 和 Server 都通过 port1（WAN1）做 SNAT 访问 Internet，SNAT 策略配置转换的源 IP 为借用出接口地址 202.103.12.2。

配置步骤

1. 配置 VIP 映射，将 202.103.12.3 映射到内网 Server 10.10.2.100。

> **重要**
>
> 建议在配置 VIP 时绑定对应的接口，详情见“结果验证”的第 2 步。




```bash
config firewall vip
    edit "VIP_to_Server"
        set extip 202.103.12.3
        set mappedip "10.10.2.100"
        set extintf "port1"
```

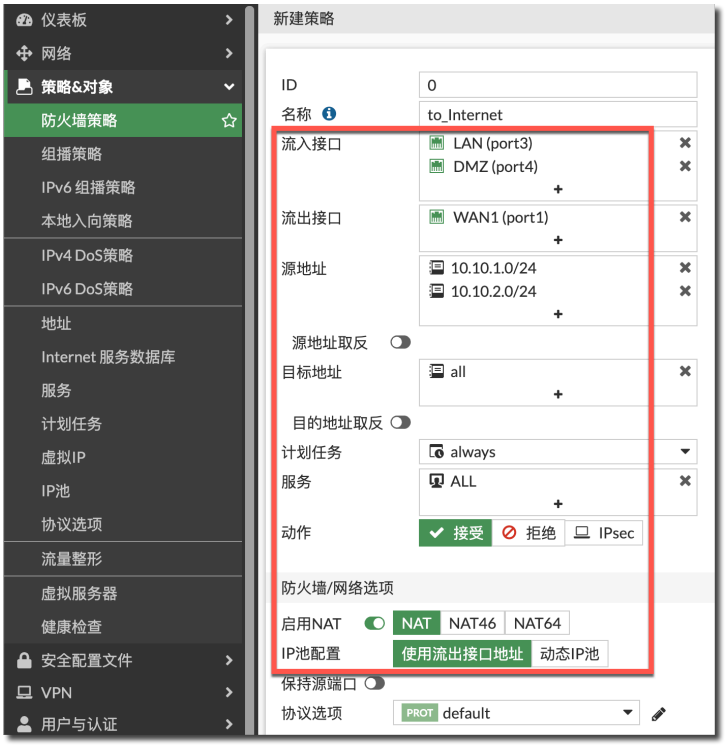
```
    next
end
```

2. 使用安全策略放通 Internet 到 VIP 的访问。




```bash
config firewall policy
    edit 1
        set name "VIP_to_Server"
        set srcintf "port1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_to_Server"
        set schedule "always"
        set service "ALL"
    next
end
```

3. 配置上网策略，允许内网 PC 和 Server 访问 Internet，并开启 SNAT，SNAT IP 池选择使用流出接口地址。




```bash
config firewall policy
    edit 2
        set name "to_Internet"
        set srcintf "port3" "port4"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24" "10.10.2.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

1. 内网 PC 可以访问 Internet 时，SNAT 转换的源地址正确地转换为出接口 port1（WAN1）的 IP 地址 202.103.12.2。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
10.314895 port3 in 10.10.1.100 -> 110.242.68.3: icmp: echo request
10.314955 port1 out 202.103.12.2 -> 110.242.68.3: icmp: echo request
10.329378 port1 in 110.242.68.3 -> 202.103.12.2: icmp: echo reply
10.329411 port3 out 110.242.68.3 -> 10.10.1.100: icmp: echo reply
11.329278 port3 in 10.10.1.100 -> 110.242.68.3: icmp: echo request
11.329294 port1 out 202.103.12.2 -> 110.242.68.3: icmp: echo request
11.342086 port1 in 110.242.68.3 -> 202.103.12.2: icmp: echo reply
11.342098 port3 out 110.242.68.3 -> 10.10.1.100: icmp: echo reply
```

2. Server 也可以正常访问 Internet，但是经过 SNAT 后，转换的源地址不是出接口 port1（WAN1）的 IP 地址 202.103.12.2，而是 VIP 配置的 extip 地址 202.103.12.3。

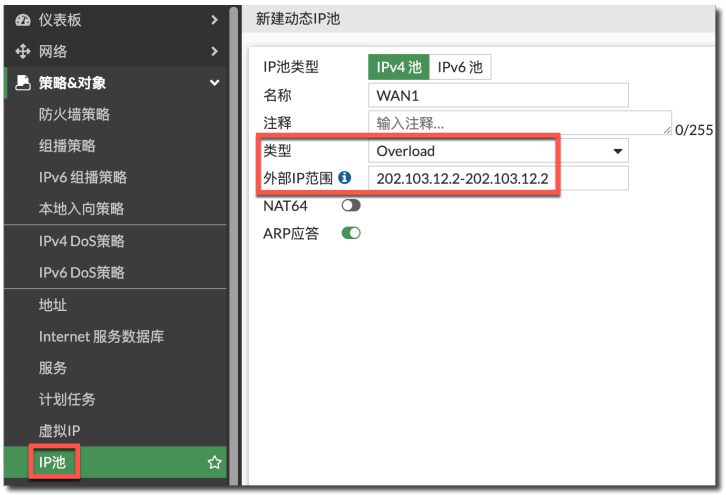
> **重要**
>
> 在多公网出口环境中，VIP 配置必须按照上述配置步骤绑定对应的公网接口，否则映射的 Server 主动访问 Internet 并执行 SNAT 时，可能会选错 SNAT 源地址。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
6.055136 port4 in 10.10.2.100 -> 110.242.68.4: icmp: echo request
6.055191 port1 out 202.103.12.3 -> 110.242.68.4: icmp: echo request
6.068436 port1 in 110.242.68.4 -> 202.103.12.3: icmp: echo reply
6.068471 port4 out 110.242.68.4 -> 10.10.2.100: icmp: echo reply
7.063159 port4 in 10.10.2.100 -> 110.242.68.4: icmp: echo request
7.063176 port1 out 202.103.12.3 -> 110.242.68.4: icmp: echo request
7.075904 port1 in 110.242.68.4 -> 202.103.12.3: icmp: echo reply
7.075913 port4 out 110.242.68.4 -> 10.10.2.100: icmp: echo reply
```

3. 默认配置下，当访问 Internet 的 SNAT 策略使用流出接口地址作为 IP 池配置时，有 VIP 映射的服务器访问 Internet，会优先使用 VIP 的 extip（202.103.12.3）作为 SNAT 的源地址，而不是流出接口地址（202.103.12.2）。

解决方法

1. 如果要求 VIP 映射的服务器访问 Internet 时，按照上网策略的配置使用流出接口地址作为 SNAT 的源地址，则需要将上网策略的 SNAT 池改为“动态 IP 池”方式。新建 IP 池，范围配置为出接口 port1（WAN1）的 IP。




```bash
config firewall ippool
    edit "WAN1"
        set startip 202.103.12.2
        set endip 202.103.12.2
    next
end
```

2. 编辑上网策略，将“IP 池配置”修改为“动态 IP 池”，选择上步创建的 IP 池。




```bash
config firewall policy
    edit 2
        set name "to_Internet"
        set srcintf "port3" "port4"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24" "10.10.2.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set ippool enable
        set poolname "WAN1"
    next
end
```

3. 再次使用 Server 访问 Internet，可以看到 FortiGate 使用出接口 port1（WAN1）作为 SNAT 的源地址。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
1.358499 port4 in 10.10.2.100 -> 110.242.68.3: icmp: echo request
1.358517 port1 out 202.103.12.2 -> 110.242.68.3: icmp: echo request
1.372310 port1 in 110.242.68.3 -> 202.103.12.2: icmp: echo reply
1.372331 port4 out 110.242.68.3 -> 10.10.2.100: icmp: echo reply
2.364802 port4 in 10.10.2.100 -> 110.242.68.3: icmp: echo request
2.364819 port1 out 202.103.12.2 -> 110.242.68.3: icmp: echo request
2.377127 port1 in 110.242.68.3 -> 202.103.12.2: icmp: echo reply
2.377137 port4 out 110.242.68.3 -> 10.10.2.100: icmp: echo reply
```

4. 如果上网策略配置了“动态 IP 池”，但 VIP 映射的 Server 又想使用 VIP 的 extip 作为 SNAT 的源地址，那么可以通过如下配置实现。在 VIP 的配置中开启 nat-source-vip 后，即使上网策略的 SNAT 使用了“动态 IP 池”，VIP 映射的主机依然会使用 VIP 的 extip 作为 SNAT 的源地址。

```bash
config firewall vip
    edit "VIP_to_Server"
        set nat-source-vip enable
    next
end
```

5. 开启 nat-source-vip 后，Server 再次访问 Internet，SNAT 转换的源地址是 VIP 配置的 extip 地址 202.103.12.3。

```
FortiGate # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
2.769121 port4 in 10.10.2.100 -> 110.242.68.3: icmp: echo request
2.769275 port1 out 202.103.12.3 -> 110.242.68.3: icmp: echo request
2.782631 port1 in 110.242.68.3 -> 202.103.12.3: icmp: echo reply
2.782670 port4 out 110.242.68.3 -> 10.10.2.100: icmp: echo reply
3.781464 port4 in 10.10.2.100 -> 110.242.68.3: icmp: echo request
3.781469 port1 out 202.103.12.3 -> 110.242.68.3: icmp: echo request
3.795056 port1 in 110.242.68.3 -> 202.103.12.3: icmp: echo reply
3.795062 port4 out 110.242.68.3 -> 10.10.2.100: icmp: echo reply
```

总结

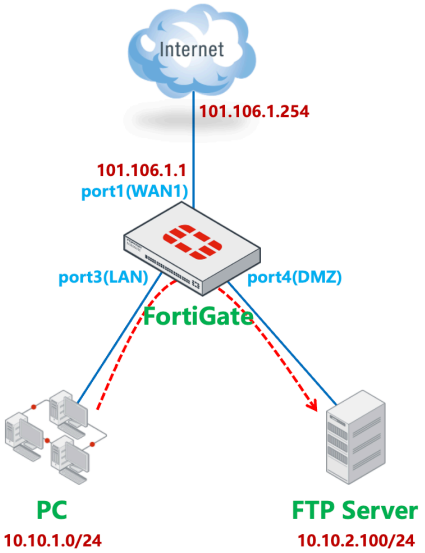
1.**上网策略的 SNAT 使用“流出接口地址”作为“IP 池配置”时：**VIP 映射的内网 Server 访问 Internet，FortiGate 使用 VIP 的 extip 作为 SNAT 的源 IP，而不是上网策略的出接口地址（nat-source-vip 不起任何作用）。 2.**上网策略的 SNAT 使用“动态 IP 池”作为“IP 池配置”时：**

- 默认情况下，VIP 映射的内网 Server 访问 Internet，FortiGate 使用 IP 池范围中的地址作为 SNAT 的源 IP，而不是 VIP 的 extip。
- 如果想使用 VIP 的 extip 作为 SNAT 的源 IP，则需要在 VIP 配置中开启 nat-source-vip。

内网主机通过公网 IP 访问 VIP

需求场景一

- *PC 与 Server 对于 FortiGate 属于不同的接口、相同的网段。*
- *PC 与 Server 属于不同网段。*



1. FortiGate 的 port1（WAN1）连接 ISP。默认路由 1 指向 port1（WAN1）。

> **重要**
>
> 如果通过策略路由来控制内网 PC 访问 Internet，请参考本文注意事项章节中的说明。

```bash
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.106.1.254, port1, [1/0]
C       10.10.1.0/24 is directly connected, port3
C       10.10.2.0/24 is directly connected, port4
C       101.106.1.0/24 is directly connected, port1
```

2. FortiGate 通过配置 VIP，在 port1（WAN1）的公网 IP 101.106.1.1 上映射了 FTP Server 的 21 端口。

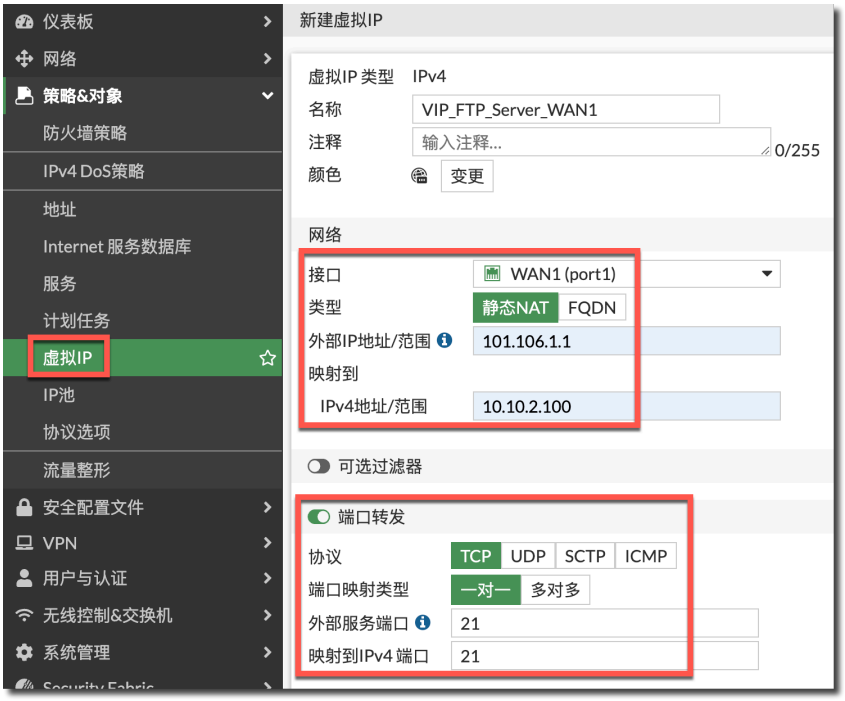
3. 内网 PC 通过 port1（WAN1）做 SNAT 访问互联网。

4. 内网 PC 和 FTP Server 位于不同的 FortiGate 内网口下，属于不同网段。

5. 内网 PC 需要通过 FortiGate 在 port1（WAN1）上映射的公网 IP 101.106.1.1（或通过域名解析得到此公网 IP）的 21 端口访问 FTP Server。

配置步骤

1. 正常配置属于外网接口 port1（WAN1）的 VIP 条目，注意这里必须配置 VIP 的关联接口（此例中为 port1），否则无法实现此功能。




```bash
config firewall vip
    edit "VIP_FTP_Server_WAN1"
        set extip 101.106.1.1
        set mappedip "10.10.2.100"
        set extintf "port1"
        set portforward enable
        set extport 21
        set mappedport 21
    next
end
```

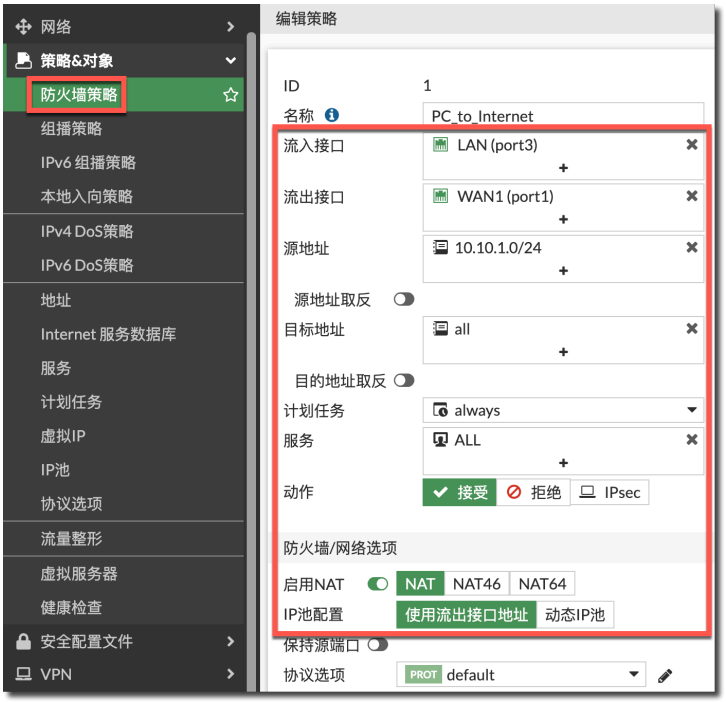
2. 配置安全策略放通外网 port1（WAN1）到内网 port4（DMZ）且目的 IP 为 VIP 条目的策略，注意这里不用开启 SNAT。

> **重要**
>
> VIP 策略的“源地址”一定要包含 PC 所在网段，这里以“all”为例，既能包含外网访问 VIP 的流量，也包含 PC 所在网段。



```bash
config firewall policy
    edit 2
        set name "VIP_FTP_Server_WAN1"
        set srcintf "port1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_FTP_Server_WAN1"
        set schedule "always"
        set service "FTP"
    next
end
```

3. 正常配置放通内网 port3（LAN）到外网 port1（WAN1）的上网策略。




```bash
config firewall policy
    edit 1
        set name "PC_to_Internet"
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

内网 PC 就可以通过 FTP 服务器的 VIP 公网 IP 访问到 DMZ 区域的 FTP 服务器。

```
管理员: C:\WINDOWS\system32\cmd.exe - ftp  101.106.1.1

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . . . . . . . . : 10.10.1.200
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.10.1.1

以太网适配器 以太网:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 蓝牙网络连接:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

C:\Users\Administrator.SUMMERICE2019>ftp 101.106.1.1
连接到 101.106.1.1。
220 3Com 3CDaemon FTP 服务器版本 2.0
530 没有登录
用户(101.106.1.1:(none)): user1
331 用户名正确，需要口令
密码:
230 用户已登录
ftp> ls
200 PORT 命令成功
150 文件状态正常；准备启动数据连接
.
..
3CDaemon.EXE
3CDaemon.ini
desktop.ini
Ftpd.log
MiniWebserver
MiniWebserver.zip
NoAccess.ftp
Profiles.ftp
putty.exe
Tftpd.log
TFTPHost.ini
226 正在关闭数据连接
ftp: 收到 159 字节，用时 0.00秒 159000.00千字节/秒。
```

抓包分析

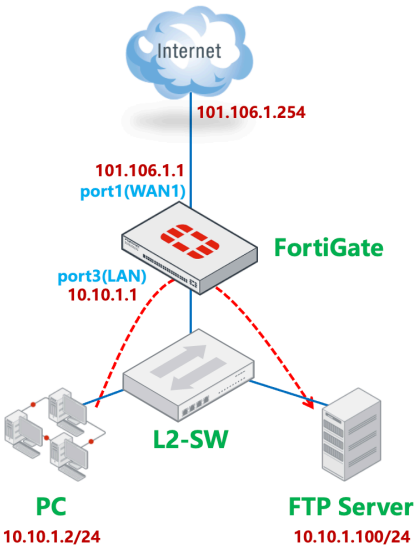
```
FortiGate # diagnose sniffer packet any 'port 21' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 21]
1.637679 port3 in 10.10.1.200.58661 -> 101.106.1.1.21: syn 1988247018
1.637770 port4 out 10.10.1.200.58661 -> 10.10.2.100.21: syn 1988247018
内网主机10.10.1.200发起Syn请求；FortiGate在内网口port3自动做了一次DNAT，将服务器的公网IP 101.106.1.1转成私网地址10.10.2.100

1.638900 port4 in 10.10.2.100.21 -> 10.10.1.200.58661: syn 2530392497 ack 1988247019
1.638920 port3 out 101.106.1.1.21 -> 10.10.1.200.58661: syn 2530392497 ack 1988247019
FTP服务器回复Syn ack；做DNAT还原，此过程对于Client和FTP服务器都是透明的，用户无法感知
1.639630 port3 in 10.10.1.200.58661 -> 101.106.1.1.21: ack 2530392498
1.639638 port4 out 10.10.1.200.58661 -> 10.10.2.100.21: ack 2530392498
1.644298 port4 in 10.10.2.100.21 -> 10.10.1.200.58661: psh 2530392498 ack 1988247019
1.644318 port3 out 101.106.1.1.21 -> 10.10.1.200.58661: psh 2530392498 ack 1988247019
1.697043 port3 in 10.10.1.200.58661 -> 101.106.1.1.21: ack 2530392540
1.697071 port4 out 10.10.1.200.58661 -> 10.10.2.100.21: ack 2530392540
```

Debug Flow 分析

```
FortiGate # id=65308 trace_id=28 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.1.200:58759->101.106.1.1:21) tu
n_id=0.0.0.0 from port3. flag [S], seq 507603746, ack 0, win 64240"
id=65308 trace_id=28 func=init_ip_session_common line=6043 msg="allocate a new session-00003b2f, tun_id=0.0.0.0"
id=65308 trace_id=28 func=get_new_addr line=1239 msg="find DNAT: IP-10.10.2.100, port-21"
id=65308 trace_id=28 func=get_new_addr line=1239 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-58759"
id=65308 trace_id=28 func=fw_pre_route_handler line=184 msg="VIP-10.10.2.100:21, outdev-unknown"
id=65308 trace_id=28 func=__ip_session_run_tuple line=3445 msg="DNAT 101.106.1.1:21->10.10.2.100:21"
id=65308 trace_id=28 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port4"
id=65308 trace_id=28 func=fw_forward_handler line=1000 msg="Allowed by Policy-2:"
id=65308 trace_id=28 func=ip_session_confirm_final line=3090 msg="npu_state=0x100, hook=4"
id=65308 trace_id=28 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=original)"
id=65308 trace_id=29 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.2.100:21->10.10.1.200:58759) tun_id=0.0.0.0
 from port4. flag [S.], seq 51973133, ack 507603747, win 65535"
id=65308 trace_id=29 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00003b2f, reply direction"
id=65308 trace_id=29 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=29 func=npu_handle_session44 line=1322 msg="Trying to offloading session from port4 to port3, skb.npu_flag=00000400 ses.state=
00000200 ses.npu_state=0x00000100"
id=65308 trace_id=29 func=fw_forward_dirty_handler line=438 msg="state=00000200, state2=00000000, npu_state=00000100"
id=65308 trace_id=29 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.2.100->101.106.1.1:21"
id=65308 trace_id=29 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=reply)"
```

需求场景二

- *PC 与 Server 对于 FortiGate 属于相同的接口、相同的网段。*
- *PC 与 Server 属于相同网段。*



1. FortiGate 的 port1（WAN1）连接 ISP。默认路由 1 指向 port1（WAN1）。

> **重要**
>
> 如果通过策略路由来控制内网 PC 访问 Internet，请参考本文注意事项章节中的说明。

```bash
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.106.1.254, port1, [1/0]
C       10.10.1.0/24 is directly connected, port3
C       101.106.1.0/24 is directly connected, port1
```

2. FortiGate 通过配置 VIP，在 port1（WAN1）的公网 IP 101.106.1.1 上映射了 FTP Server 的 21 端口。

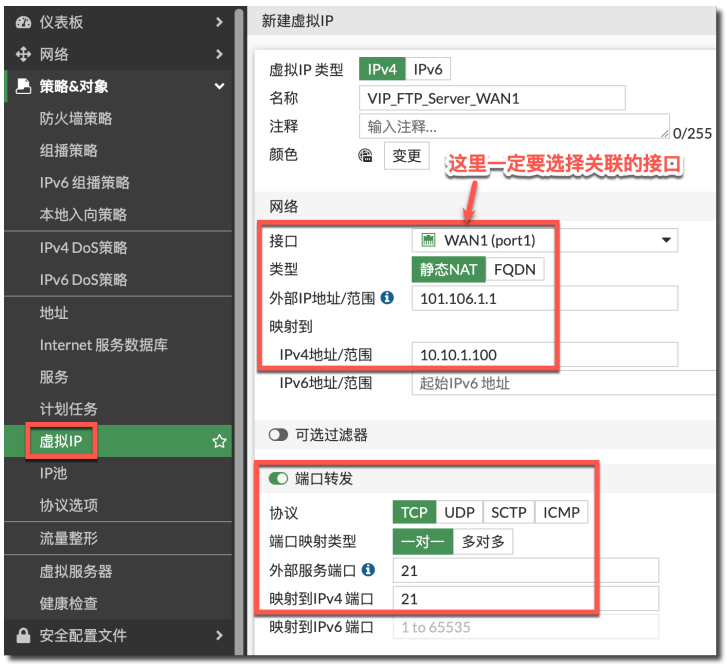
3. 内网 PC 通过 port1（WAN1）做 SNAT 访问互联网。

4. 内网 PC 和 FTP Server 位于相同的二层交换机下，与 FortiGate 的 port3（LAN）接口都属于同一网段。

5. 内网 PC 需要通过 FortiGate 在 port1（WAN1）上映射的公网 IP 101.106.1.1（或通过域名解析得到此公网 IP）的 21 端口访问 FTP Server。

配置步骤

1. 正常配置属于外网接口 port1（WAN1）的 VIP 条目，注意这里必须配置 VIP 的关联接口（此例中为 port1），否则无法实现此功能。



```bash
config firewall vip
    edit "VIP_FTP_Server_WAN1"
        set extip 101.106.1.1
        set mappedip "10.10.1.100"
        set extintf "port1"
        set portforward enable
        set extport 21
        set mappedport 21
    next
end
```

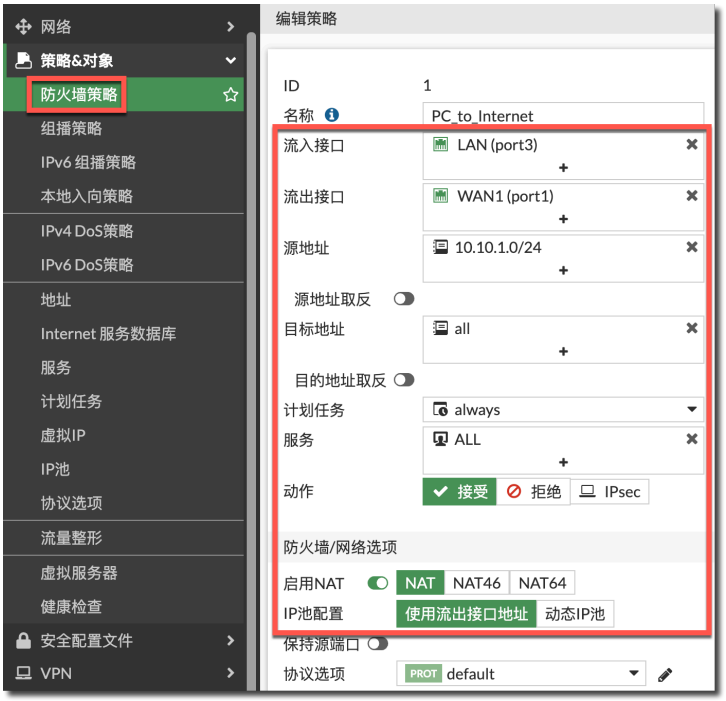
2. 配置安全策略放通外网 port1（WAN1）到内网 port3（LAN）且目的 IP 为 VIP 条目的策略，注意这里不用开启 SNAT。

> **重要**
>
> VIP 策略的“源地址”一定要包含 PC 所在网段，这里以“all”为例，既能包含外网访问 VIP 的流量，也包含 PC 所在网段。




```bash
config firewall policy
    edit 2
        set name "VIP_FTP_Server_WAN1"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_FTP_Server_WAN1"
        set schedule "always"
        set service "FTP"
    next
end
```

3. 正常配置放通内网 port3（LAN）到外网 port1（WAN1）的上网策略。




```bash
config firewall policy
    edit 1
        set name "PC_to_Internet"
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

内网 PC 就可以通过 FTP 服务器的 VIP 公网 IP 访问到 LAN 区域的 FTP 服务器。

```
以太网适配器 Ethernet0:

   连接特定的 DNS 后缀 . . . . . :
   本地链接 IPv6 地址. . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . . . . . . . . : 10.10.1.2
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.10.1.1

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :

C:\Users\Administrator.SUMMERICE2019>ftp 101.106.1.1
连接到 101.106.1.1。
220 3Com 3CDaemon FTP 服务器版本 2.0
530 没有登录
用户(101.106.1.1:(none)): user1
331 用户名正确，需要口令
密码:
230 用户已登录
ftp> ls
200 PORT 命令成功
150 文件状态正常；准备启动数据连接
.
..
3CDaemon.EXE
3CDaemon.ini
desktop.ini
Ftpd.log
MiniWebserver
MiniWebserver.zip
NoAccess.ftp
Profiles.ftp
putty.exe
Tftpd.log
TFTPHost.ini
226 正在关闭数据连接
ftp: 收到 159 字节，用时 0.03秒 5.13千字节/秒。
```

抓包分析

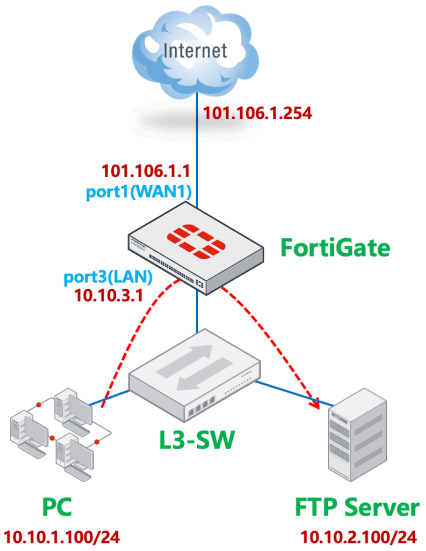
```
FortiGate # diagnose sniffer packet any 'port 21' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 21]
2.240199 port3 in 10.10.1.2.58933 -> 101.106.1.1.21: syn 4029466216
2.240316 port3 out 10.10.1.1.58933 -> 10.10.1.100.21: syn 4029466216
内网主机10.10.1.2发起Syn请求；FortiGate在内网口port3自动做了一次DNAT，将公网IP 101.106.1.1转换成私网IP 10.10.1.100
同时做了一次SNAT，将源IP 10.10.1.2转换为内网口port3的IP 10.10.1.1
2.241016 port3 in 10.10.1.100.21 -> 10.10.1.1.58933: syn 3261134504 ack 4029466217
2.241037 port3 out 101.106.1.1.21 -> 10.10.1.2.58933: syn 3261134504 ack 4029466217
FTP服务器回复Syn ack；DNAT还原+SNAT还原，此过程对于Client和FTP服务器都是透明的，用户无法感知
2.241390 port3 in 10.10.1.2.58933 -> 101.106.1.1.21: ack 3261134505
2.241400 port3 out 10.10.1.1.58933 -> 10.10.1.100.21: ack 3261134505
2.245253 port3 in 10.10.1.100.21 -> 10.10.1.1.58933: psh 3261134505 ack 4029466217
2.245272 port3 out 101.106.1.1.21 -> 10.10.1.2.58933: psh 3261134505 ack 4029466217
2.296049 port3 in 10.10.1.2.58933 -> 101.106.1.1.21: ack 3261134547
2.296111 port3 out 10.10.1.1.58933 -> 10.10.1.100.21: ack 3261134547
2.304163 port3 in 10.10.1.2.58933 -> 101.106.1.1.21: psh 4029466217 ack 3261134547
2.304198 port3 out 10.10.1.1.58933 -> 10.10.1.100.21: psh 4029466217 ack 3261134547
2.307333 port3 in 10.10.1.100.21 -> 10.10.1.1.58933: psh 3261134547 ack 4029466231
2.307357 port3 out 101.106.1.1.21 -> 10.10.1.2.58933: psh 3261134547 ack 4029466231
2.357734 port3 in 10.10.1.2.58933 -> 101.106.1.1.21: ack 3261134566
2.357766 port3 out 10.10.1.1.58933 -> 10.10.1.100.21: ack 3261134566
```

Debug Flow 分析

```
FortiGate # id=65308 trace_id=86 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.1.2:58945->101.106.1.1:21) tun_
id=0.0.0.0 from port3. flag [S], seq 2448638206, ack 0, win 8192"
id=65308 trace_id=86 func=init_ip_session_common line=6043 msg="allocate a new session-00004bdf, tun_id=0.0.0.0"
id=65308 trace_id=86 func=get_new_addr line=1239 msg="find DNAT: IP-10.10.1.100, port-21"
id=65308 trace_id=86 func=get_new_addr line=1239 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-58945"
id=65308 trace_id=86 func=fw_pre_route_handler line=184 msg="VIP-10.10.1.100:21, outdev-unknown"
id=65308 trace_id=86 func=__ip_session_run_tuple line=3445 msg="DNAT 101.106.1.1:21->10.10.1.100:21"
id=65308 trace_id=86 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=86 func=fw_forward_handler line=1000 msg="Allowed by Policy-2: SNAT"
id=65308 trace_id=86 func=ip_session_confirm_final line=3000 msg="npu_state=0x100, hook=4"
id=65308 trace_id=86 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.1.2->10.10.1.1:58945"
id=65308 trace_id=86 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=original)"
id=65308 trace_id=87 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.1.100:21->10.10.1.1:58945) tun_id=0.0.0.0 f
rom port3. flag [S.], seq 2750902401, ack 2448638207, win 65535"
id=65308 trace_id=87 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00004bdf, reply direction"
id=65308 trace_id=87 func=__ip_session_run_tuple line=3445 msg="DNAT 10.10.1.1:58945->10.10.1.2:58945"
id=65308 trace_id=87 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=87 func=npu_handle_session44 line=1322 msg="Trying to offloading session from port3 to port3, skb.npu_flag=00000400 ses.state=
00000200 ses.npu_state=0x00000101"
id=65308 trace_id=87 func=fw_forward_dirty_handler line=438 msg="state=00000200, state2=00000000, npu_state=00000101"
id=65308 trace_id=87 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.1.100->101.106.1.1:21"
id=65308 trace_id=87 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=reply)"
```

需求场景三

- *PC 与 Server 对于 FortiGate 属于相同的接口、不同的网段。*
- *PC 与 Server 属于不同网段。*



1. FortiGate 的 port1（WAN1）连接 ISP。默认路由 1 指向 port1（WAN1）。

> **重要**
>
> 如果通过策略路由来控制内网 PC 访问 Internet，请参考本文注意事项章节中的说明。

```bash
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.106.1.254, port1, [1/0]
C       10.10.3.0/24 is directly connected, port3
C       101.106.1.0/24 is directly connected, port1
```

2. FortiGate 通过配置 VIP，在 port1（WAN1）的公网 IP 101.106.1.1 上映射了 FTP Server 的 21 端口。

3. 内网 PC 通过 port1（WAN1）做 SNAT 访问互联网。

4. 内网 PC 和 FTP Server 属于不同网段，网关位于同一个三层交换机下。

5. 内网 PC、FTP Server 与 FortiGate 的 port3（LAN）接口属于不同网段。

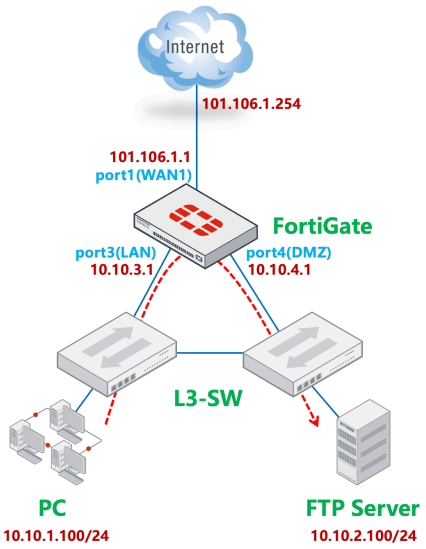
6. 内网 PC 需要通过 FortiGate 在 port1（WAN1）上映射的公网 IP 101.106.1.1（或通过域名解析得到此公网 IP）的 21 端口访问 FTP Server。

配置步骤

参考需求场景二的配置即可。

需求场景四

- *PC 与 Server 对于 FortiGate 属于不同的接口、不同的网段。*
- *PC 与 Server 属于不同网段。*



1. FortiGate 的 port1（WAN1）连接 ISP。默认路由 1 指向 port1（WAN1）。

> **重要**
>
> 如果通过策略路由来控制内网 PC 访问 Internet，请参考本文注意事项章节中的说明。

```bash
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.106.1.254, port1, [1/0]
S       10.10.1.0/24 [10/0] via 10.10.3.2, port3, [1/0]
S       10.10.2.0/24 [10/0] via 10.10.4.2, port4, [1/0]
C       10.10.3.0/24 is directly connected, port3
C       10.10.4.0/24 is directly connected, port4
C       101.106.1.0/24 is directly connected, port1
```

2. FortiGate 通过配置 VIP，在 port1（WAN1）的公网 IP 101.106.1.1 上映射了 FTP Server 的 21 端口。

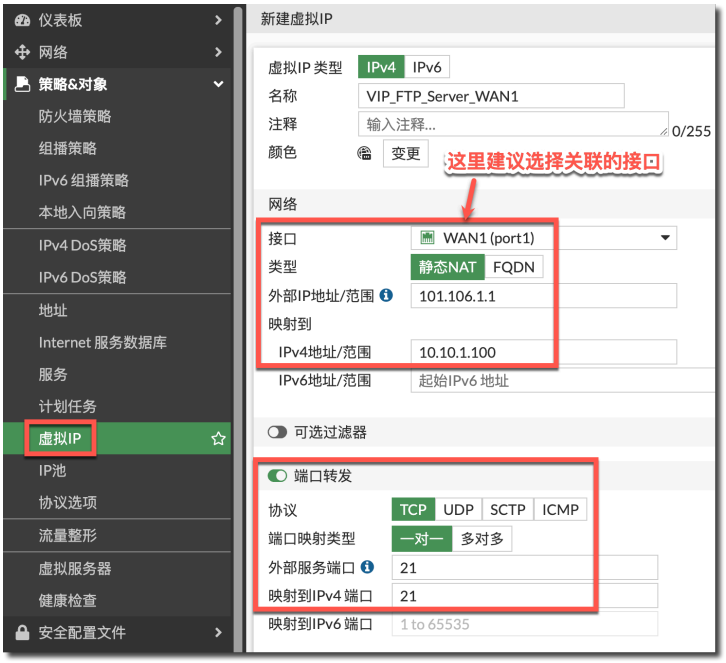
3. 内网 PC 通过 port1（WAN1）做 SNAT 访问互联网。

4. 内网 PC 和 FTP Server 位于不同的 3 层交换机下，属于不同网段，网关为各自相连的 L3-SW，且 PC 可以和 FTP Server 通过两台 L3-SW 直接通信。两台 L3-SW 与 FortiGate 的 2 个接口分别三层互联，两台 L3-SW 的网关都指向 FortiGate。

5. 内网 PC 需要通过 FortiGate 在 port1（WAN1）上映射的公网 IP 101.106.1.1（或通过域名解析得到此公网 IP）的 21 端口访问 FTP Server。

配置步骤

1. 正常配置属于外网接口 port1（WAN1）的 VIP 条目，强烈建议配置 VIP 的关联接口（此例中为 port1，不要配置为 any），否则可能产生其他问题。




```bash
config firewall vip
    edit "VIP_FTP_Server_WAN1"
        set extip 101.106.1.1
        set mappedip "10.10.2.100"
        set extintf "port1"
        set portforward enable
        set extport 21
        set mappedport 21
    next
end
```

2. 配置安全策略：公网 port1（WAN1）到 port4（DMZ），源地址为内网 PC 网段 10.10.1.0/24，且目的 IP 为 VIP 条目的策略，注意这条策略开启 SNAT，用于内网 PC 对 VIP 的访问。




```bash
config firewall policy
    edit 3
        set name "LAN_to_VIP_FTP_Server"
        set srcintf "port1"
        set dstintf "port4"
```

```
        set action accept
        set srcaddr "10.10.1.0/24"
        set dstaddr "VIP_FTP_Server_WAN1"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

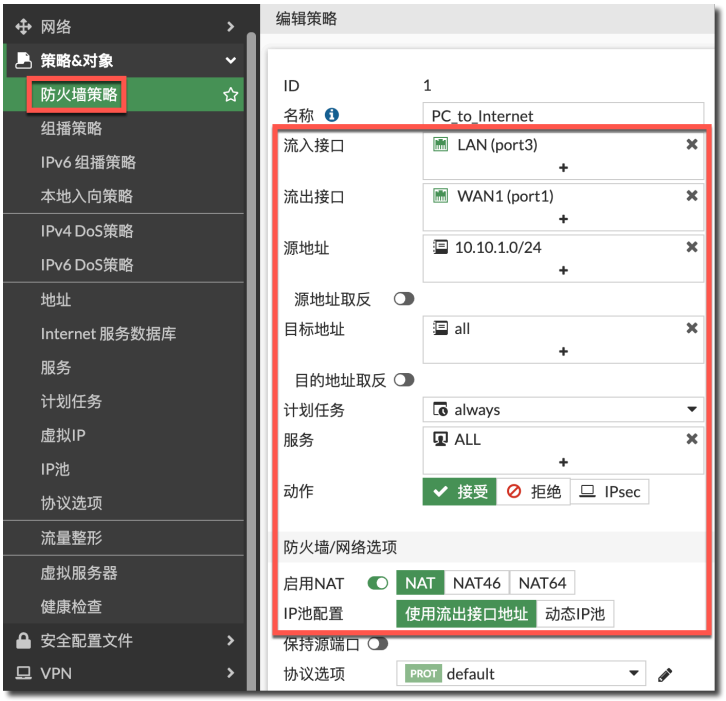
3. 配置安全策略放通外网 port1（WAN1）到内网 port4（DMZ）且目的 IP 为 VIP 条目的策略，注意这里不用开启 SNAT，用于外网客户端对 VIP 的访问。

> **重要**
>
> VIP 策略的“源地址”一定要包含 PC 所在网段，这里以“all”为例，既能包含外网访问 VIP 的流量，也包含 PC 所在网段。




```
config firewall policy
    edit 2
        set name "VIP_FTP_Server_WAN1"
        set srcintf "port1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_FTP_Server_WAN1"
        set schedule "always"
        set service "FTP"
    next
end
```
bash

4. 正常配置放通内网 port3（LAN）到外网 port1（WAN1）的上网策略。




```bash
config firewall policy
    edit 1
        set name "PC_to_Internet"
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

5. 注意策略顺序如下。



结果验证

内网 PC 就可以通过 FTP 服务器的 VIP 公网 IP 访问到 DMZ 区域的 FTP 服务器。

```
管理员: C:\WINDOWS\system32\cmd.exe - ftp  101.106.1.1

   连接特定的 DNS 后缀 . . . . . :
   本地链接 IPv6 地址. . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . . . . . . . . : 10.10.1.100
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.10.1.1

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :

C:\Users\Administrator.SUMMERICE2019>ftp 101.106.1.1
连接到 101.106.1.1。
220 3Com 3CDaemon FTP 服务器版本 2.0
530 没有登录
用户(101.106.1.1:(none)): user1
331 用户名正确，需要口令
密码:
230 用户已登录
ftp> ls
200 PORT 命令成功
150 文件状态正常；准备启动数据连接
.
..
3CDaemon.EXE
3CDaemon.ini
desktop.ini
Ftpd.log
MiniWebserver
MiniWebserver.zip
NoAccess.ftp
Profiles.ftp
putty.exe
Tftpd.log
TFTPHost.ini
226 正在关闭数据连接
ftp: 收到 159 字节，用时 0.05秒 3.46千字节/秒。
```

抓包分析

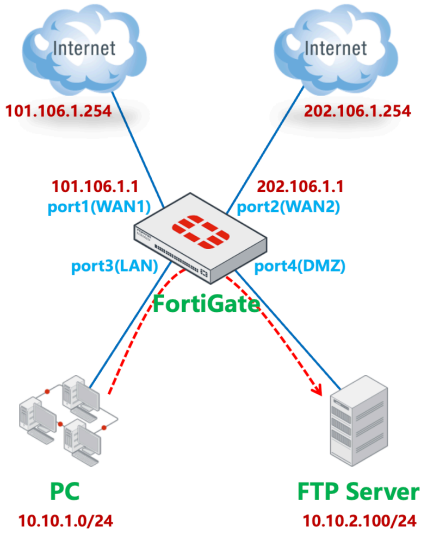
```
FortiGate # diagnose sniffer packet any 'port 21' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 21]
1.311118 port3 in 10.10.1.100.63731 -> 101.106.1.1.21: syn 742153841
1.311238 port4 out 10.10.4.1.63731 -> 10.10.2.100.21: syn 742153841
内网主机10.10.1.100发起Syn请求：FortiGate在内网口port3自动做了一次DNAT，将公网IP 101.106.1.1转换为10.10.2.100
同时做了一次SNAT，将源IP 10.10.1.100转换为DMZ接口port4的IP 10.10.4.1
1.311953 port4 in 10.10.2.100.21 -> 10.10.4.1.63731: syn 2547098373 ack 742153842
1.311967 port3 out 101.106.1.1.21 -> 10.10.1.100.63731: syn 2547098373 ack 742153842
FTP服务器回复Syn ack：DNAT还原+SNAT还原，此过程对于Client和FTP服务器是透明的，用户无法感知
1.312845 port3 in 10.10.1.100.63731 -> 101.106.1.1.21: ack 2547098374
1.312854 port4 out 10.10.4.1.63731 -> 10.10.2.100.21: ack 2547098374
1.317986 port4 in 10.10.2.100.21 -> 10.10.4.1.63731: psh 2547098374 ack 742153842
1.317994 port3 out 101.106.1.1.21 -> 10.10.1.100.63731: psh 2547098374 ack 742153842
1.358954 port3 in 10.10.1.100.63731 -> 101.106.1.1.21: psh 742153842 ack 2547098416
1.359010 port4 out 10.10.4.1.63731 -> 10.10.2.100.21: psh 742153842 ack 2547098416
1.363297 port4 in 10.10.2.100.21 -> 10.10.4.1.63731: psh 2547098416 ack 742153856
1.363321 port3 out 101.106.1.1.21 -> 10.10.1.100.63731: psh 2547098416 ack 742153856
1.427952 port3 in 10.10.1.100.63731 -> 101.106.1.1.21: ack 2547098435
1.428009 port4 out 10.10.4.1.63731 -> 10.10.2.100.21: ack 2547098435
```

Debug Flow 分析

```
FortiGate # id=65308 trace_id=1 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.1.100:63737->101.106.1.1:21) tun
_id=0.0.0.0 from port3. flag [S], seq 1515674373, ack 0, win 8192"
id=65308 trace_id=1 func=init_ip_session_common line=6043 msg="allocate a new session-00000811, tun_id=0.0.0.0"
id=65308 trace_id=1 func=get_new_addr line=1239 msg="find DNAT: IP-10.10.2.100, port-21"
id=65308 trace_id=1 func=get_new_addr line=1239 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-63737"
id=65308 trace_id=1 func=fw_pre_route_handler line=184 msg="VIP-10.10.2.100:21, outdev-unknown"
id=65308 trace_id=1 func=__ip_session_run_tuple line=3445 msg="DNAT 101.106.1.1:21->10.10.2.100:21"
id=65308 trace_id=1 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-10.10.4.2 via port4"
id=65308 trace_id=1 func=get_new_addr line=1239 msg="find SNAT: IP-10.10.4.1(from IPPOOL), port-63737"
id=65308 trace_id=1 func=fw_forward_handler line=1000 msg="Allowed by Policy-3: SNAT"
id=65308 trace_id=1 func=ip_session_confirm_final line=3090 msg="npu_state=0x100, hook=4"
id=65308 trace_id=1 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.1.100->10.10.4.1:63737"
id=65308 trace_id=1 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=original)"
id=65308 trace_id=2 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.2.100:21->10.10.4.1:63737) tun_id=0.0.0.0 fr
om port4. flag [S.], seq 3648687917, ack 1515674374, win 65535"
id=65308 trace_id=2 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-00000811, reply direction"
id=65308 trace_id=2 func=__ip_session_run_tuple line=3445 msg="DNAT 10.10.4.1:63737->10.10.1.100:63737"
id=65308 trace_id=2 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-10.10.3.2 via port3"
id=65308 trace_id=2 func=npu_handle_session44 line=1322 msg="Trying to offloading session from port4 to port3, skb.npu_flag=00000400 ses.state=0
0000200 ses.npu_state=0x00000101"
id=65308 trace_id=2 func=fw_forward_dirty_handler line=438 msg="state=00000200, state2=00000000, npu_state=00000101"
id=65308 trace_id=2 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.2.100->101.106.1.1:21"
id=65308 trace_id=2 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=reply)"
```

需求场景五

- *PC 与 Server 对于 FortiGate 属于不同的接口、相同的网段。*
- *PC 与 Server 属于不同网段。*



1. FortiGate 的 port1（WAN1）和 port2（WAN2）分别连接两个 ISP。默认路由 1 指向 port1（WAN1），路由优先级为 10，默认路由 2 指向 port2（WAN2），路由优先级为 20。

> **重要**
>
> 如果通过策略路由来控制内网 PC 访问 Internet，请参考本文注意事项章节中的说明。

```bash
FortiGate # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.106.1.254, port1, [10/0]
                  [10/0] via 202.106.1.254, port2, [20/0]
C       10.10.1.0/24 is directly connected, port3
C       10.10.2.0/24 is directly connected, port4
C       101.106.1.0/24 is directly connected, port1
C       202.106.1.0/24 is directly connected, port2
```

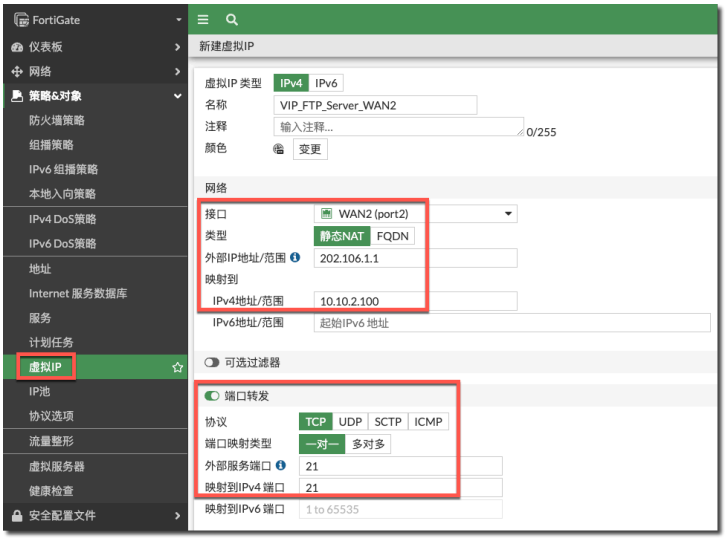
2. FortiGate 通过配置 VIP，在 port2（WAN2）的公网 IP 202.106.1.1 上映射了 FTP Server 的 21 端口。

3. 内网 PC 通过 port1（WAN1）做 SNAT 访问互联网（备用通过 port2 上网）。

4. 内网 PC 需要通过 FortiGate 在 port2（WAN2）上映射的公网 IP 202.106.1.1（或通过域名解析得到此公网 IP）的 21 端口访问 FTP Server。

配置步骤

1. 正常配置属于外网接口 port2（WAN2）的 VIP 条目，强烈建议配置 VIP 的关联接口（此例中为 port2，不要配置为 any），否则可能引起其他问题。




```bash
config firewall vip
    edit "VIP_FTP_Server_WAN2"
        set extip 202.106.1.1
        set mappedip "10.10.2.100"
        set extintf "port2"
        set portforward enable
        set extport 21
        set mappedport 21
    next
end
```

2. 配置安全策略放通外网 port2（WAN2）到内网 port4（DMZ）且目的 IP 为 VIP 条目的策略，注意这里不用开启 SNAT。

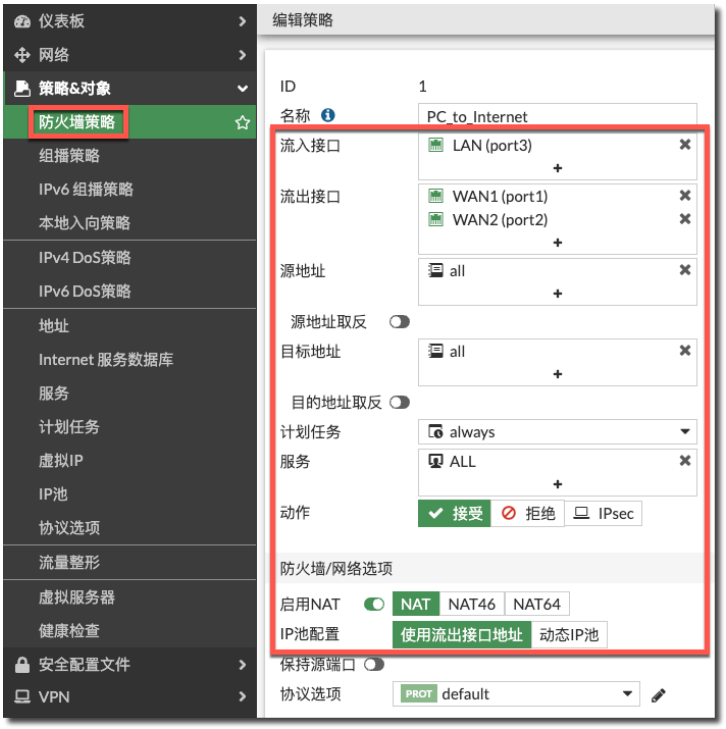
> **重要**
>
> VIP 策略的“源地址”一定要包含 PC 所在网段，这里以“all”为例，既能包含外网访问 VIP 的流量，也包含 PC 所在网段。




```bash
config firewall policy
    edit 2
        set name "VIP_FTP_Server_WAN1"
        set srcintf "port2"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
```

```
        set dstaddr "VIP_FTP_Server_WAN2"
        set schedule "always"
        set service "FTP"
    next
end
```

3. 正常配置放通内网 port3（LAN）到外网 port1（WAN1）、port2（WAN2）的上网策略。




```bash
config firewall policy
    edit 1
        set name "PC_to_Internet"
        set srcintf "port3"
        set dstintf "port1" "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

内网 PC 就可以通过 FTP 服务器的 VIP 公网 IP 访问到 DMZ 区域的 FTP 服务器。

```
管理员: 命令提示符 - ftp  202.106.1.1

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . . . . . . . . : 10.10.1.100
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.10.1.1

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

C:\Users\Administrator.SUMMERICE2019>ftp 202.106.1.1
连接到 202.106.1.1。
220 3Com 3CDaemon FTP 服务器版本 2.0
530 没有登录
用户(202.106.1.1:(none)): user1
331 用户名正确，需要口令
密码:
230 用户已登录
ftp> ls
200 PORT 命令成功
150 文件状态正常；准备启动数据连接
.
..
3CDaemon.EXE
3CDaemon.ini
desktop.ini
Ftpd.log
MiniWebserver
MiniWebserver.zip
NoAccess.ftp
Profiles.ftp
putty.exe
Tftpd.log
TFTPHost.ini
226 正在关闭数据连接
ftp: 收到 159 字节，用时 0.03秒 5.13千字节/秒。
ftp>
```

抓包分析

```
FortiGate # diagnose sniffer packet any 'port 21' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[port 21]
7.999455 port3 in 10.10.1.100.52079 -> 202.106.1.1.21: syn 227023947
7.999583 port4 out 10.10.1.100.52079 -> 10.10.2.100.21: syn 227023947
内网主机10.10.1.100发起Syn请求；FortiGate在内网口port3自动做了一次DNAT，将服务器的公网IP 202.106.1.1转成私网IP 10.10.2.100
8.000233 port4 in 10.10.2.100.21 -> 10.10.1.100.52079: syn 4073107371 ack 227023948
8.000270 port3 out 202.106.1.1.21 -> 10.10.1.100.52079: syn 4073107371 ack 227023948
FTP服务器回复Syn,ack；做DNAT还原，此过程对于Client和FTP服务器都是透明的，用户无法感知
8.000688 port3 in 10.10.1.100.52079 -> 202.106.1.1.21: ack 4073107372
8.000700 port4 out 10.10.1.100.52079 -> 10.10.2.100.21: ack 4073107372
8.004699 port4 in 10.10.2.100.21 -> 10.10.1.100.52079: psh 4073107372 ack 227023948
8.004711 port3 out 202.106.1.1.21 -> 10.10.1.100.52079: psh 4073107372 ack 227023948
8.047736 port3 in 10.10.1.100.52079 -> 202.106.1.1.21: psh 227023948 ack 4073107414
8.047858 port4 out 10.10.1.100.52079 -> 10.10.2.100.21: psh 227023948 ack 4073107414
8.054411 port4 in 10.10.2.100.21 -> 10.10.1.100.52079: psh 4073107414 ack 227023962
8.054468 port3 out 202.106.1.1.21 -> 10.10.1.100.52079: psh 4073107414 ack 227023962
8.098313 port3 in 10.10.1.100.52079 -> 202.106.1.1.21: ack 4073107433
8.098342 port4 out 10.10.1.100.52079 -> 10.10.2.100.21: ack 4073107433
```

Debug Flow 分析

```
FortiGate # id=65308 trace_id=60 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.1.100:52128->202.106.1.1:21) tu
n_id=0.0.0.0 from port3. flag [S], seq 3753854968, ack 0, win 8192"
id=65308 trace_id=60 func=init_ip_session_common line=6043 msg="allocate a new session-0001857c, tun_id=0.0.0.0"
id=65308 trace_id=60 func=get_new_addr line=1239 msg="find DNAT: IP-10.10.2.100, port-21"
id=65308 trace_id=60 func=get_new_addr line=1239 msg="find SNAT: IP-202.106.1.1(from IPPOOL), port-52128"
id=65308 trace_id=60 func=fw_pre_route_handler line=184 msg="VIP-10.10.2.100:21, outdev-unknown"
id=65308 trace_id=60 func=__ip_session_run_tuple line=3445 msg="DNAT 202.106.1.1:21->10.10.2.100:21"
id=65308 trace_id=60 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port4"
id=65308 trace_id=60 func=fw_forward_handler line=1000 msg="Allowed by Policy-2:"
id=65308 trace_id=60 func=ip_session_confirm_final line=3090 msg="npu_state=0x100, hook=4"
id=65308 trace_id=60 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=original)"
id=65308 trace_id=61 func=print_pkt_detail line=5857 msg="vd-root:0 received a packet(proto=6, 10.10.2.100:21->10.10.1.100:52128) tun_id=0.0.0.0
 from port4. flag [S.], seq 3548123384, ack 3753854969, win 65535"
id=65308 trace_id=61 func=resolve_ip_tuple_fast line=5945 msg="Find an existing session, id-0001857c, reply direction"
id=65308 trace_id=61 func=__vf_ip_route_input_rcu line=2001 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=61 func=npu_handle_session44 line=1322 msg="Trying to offloading session from port4 to port3, skb.npu_flag=00000400 ses.state=
00000200 ses.npu_state=0x00000100"
id=65308 trace_id=61 func=fw_forward_dirty_handler line=438 msg="state=00000200, state2=00000000, npu_state=00000100"
id=65308 trace_id=61 func=__ip_session_run_tuple line=3432 msg="SNAT 10.10.2.100->202.106.1.1:21"
id=65308 trace_id=61 func=__ip_session_run_tuple line=3486 msg="run helper-ftp(dir=reply)"
```

总结

内网 PC 通过 VIP 的公网地址访问内部服务器时，对于 FortiGate 来说：

- 如果发起访问的内网 PC 和 VIP 映射的真实服务器在 FortiGate 的同一个内网口下（与网段无关），FortiGate 会自动做 DNAT + SNAT。
- 如果发起访问的内网 PC 和 VIP 映射的真实服务器不在 FortiGate 的同一个内网口下（与网段无关），FortiGate 会自动做 DNAT，但不会做 SNAT。如果内网服务器和 PC 的内网 IP 可以不通过 FortiGate 直接访问，则需要像场景三的步骤 2 那样额外配置一条源为内网 PC 网段，目标为 VIP 的开启 SNAT 的策略，否则服务器返回的 Syn ack 会从下游交换机直接返回，而不会经过 FortiGate。

注意事项

如果通过策略路由来控制内网 PC 访问 Internet，则需要额外的策略路由配置。

问题现象

1. 在需求场景 2 的配置基础上，FortiGate 通过配置策略路由来控制 PC 网段访问 Internet。

```bash
config router policy
    edit 1
        set input-device "port3"
        set src "10.10.1.0/255.255.255.0"
        set gateway 101.106.1.254
        set output-device "port1"
    next
end

FortiGate # diagnose firewall proute list
list route policy info(vf=root):
id=1(0x01) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-0 iif=5(port3) dport=0-65535 path(1)
oif=3(port1) gwy=101.106.1.254
source wildcard(1): 10.10.1.0/255.255.255.0
destination wildcard(1): 0.0.0.0/0.0.0.0
hit_count=2649 last_used=2024-05-07 09:35:17
```

2. 此时使用内网 PC 就无法通过 FTP 服务器的 VIP 公网 IP 访问到 LAN 区域的 FTP 服务器。抓包查看内网 PC 访问 FTP 服务器的 VIP 公网 IP 的流量只有 in，没有 out。

```bash
FortiGate # diagnose sniffer packet any 'port 21' 4 0 l

2024-05-06 14:57:40.855316 port3 in 10.10.1.100.51646 → 101.106.1.1.21: syn 1097509905
2024-05-06 14:57:41.872411 port3 in 10.10.1.100.51646 → 101.106.1.1.21: syn 1097509905
2024-05-06 14:57:43.872568 port3 in 10.10.1.100.51646 → 101.106.1.1.21: syn 1097509905
```

3. 相关 Debug Flow 显示 TCP SYN 报文经过 DNAT 转换后，匹配上网的策略路由，出接口为 port1（WAN1），下一跳为运营商网关。但没有可以匹配的防火墙策略，被隐式策略（0）丢弃。

```
FortiGate # id=65308 trace_id=1 func=print_pkt_detail line=5894 msg="vd-root:0 received a packet(proto=6, 10.10.1.2:63185->101.106.1.1:21) tun_id
=0.0.0.0 from port3. flag [S], seq 2162673907, ack 0, win 8192"
id=65308 trace_id=1 func=init_ip_session_common line=6080 msg="allocate a new session-00000b70, tun_id=0.0.0.0"
id=65308 trace_id=1 func=get_new_addr line=1213 msg="find DNAT: IP-10.10.1.100, port-21"
id=65308 trace_id=1 func=__iprope_tree_check line=535 msg="gnum-100004, use addr/intf hash, len=2"
id=65308 trace_id=1 func=get_new_addr line=1213 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-63185"
id=65308 trace_id=1 func=fw_pre_route_handler line=184 msg="VIP-10.10.1.100:21, outdev-unknown"
id=65308 trace_id=1 func=__ip_session_run_tuple line=3471 msg="DNAT 101.106.1.1:21->10.10.1.100:21"
id=65308 trace_id=1 func=rpdb_srv_match_input line=1040 msg="Match policy routing id=1: to 101.106.1.254 via ifindex-3"
id=65308 trace_id=1 func=__vf_ip_route_input_rcu line=1990 msg="find a route: flag=00000000 gw-101.106.1.254 via port1"
id=65308 trace_id=1 func=__iprope_tree_check line=524 msg="gnum-100004, use int hash, slot=27, len=1"
id=65308 trace_id=1 func=fw_forward_handler line=837 msg="Denied by forward policy check (policy 0)"
```

解决方法 1

1. 在上网的策略路由 之前添加一条新的策略路由（如下配置中的 ID 2）。流入接口为 port3（LAN），源地址为内网 PC 网段（10.10.1.0/24），目的地址为 FTP Server 的内网 IP，动作为停止策略路由（deny）。

```bash
config router policy
    edit 2
        set input-device "port3"
        set src "10.10.1.0/255.255.255.0"
        set dst "10.10.1.100/255.255.255.255"
        set action deny
    next
end
    edit 1
        set input-device "port3"
```

```
        set src "10.10.1.0/255.255.255.0"
        set gateway 101.106.1.254
        set output-device "port1"
    next
end
```

2. 此时使用内网 PC 可以正常通过 FTP 服务器的 VIP 公网 IP 访问到 LAN 区域的 FTP 服务器，Debug Flow 信息显示不会匹配策略路由，流程与场景二中的 Debug Flow 一致：

```
FortiGate # id=65308 trace_id=43 func=print_pkt_detail line=5894 msg="vd-root:0 received a packet(proto=6, 10.10.1.2:63205->101.106.1.1:21) tun_"
id=65308 trace_id=43 func=init_ip_session_common line=6080 msg="allocate a new session-00000dcd, tun_id=0.0.0.0"
id=65308 trace_id=43 func=get_new_addr line=1213 msg="find DNAT: IP-10.10.1.100, port-21"
id=65308 trace_id=43 func=__iprope_tree_check line=535 msg="gnum-100004, use addr/intf hash, len=2"
id=65308 trace_id=43 func=get_new_addr line=1213 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-63205"
id=65308 trace_id=43 func=fw_pre_route_handler line=184 msg="VIP-10.10.1.100:21, outdev-unknown"
id=65308 trace_id=43 func=__ip_session_run_tuple line=3471 msg="DNAT 101.106.1.1:21->10.10.1.100:21"
id=65308 trace_id=43 func=__vf_ip_route_input_rcu line=1990 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=43 func=__iprope_tree_check line=524 msg="gnum-100004, use int hash, slot=29, len=2"
id=65308 trace_id=43 func=fw_forward_handler line=989 msg="Allowed by Policy-2: SNAT"
id=65308 trace_id=43 func=ip_session_confirm_final line=3113 msg="npu_state=0x100, hook=4"
id=65308 trace_id=43 func=__ip_session_run_tuple line=3458 msg="SNAT 10.10.1.2->10.10.1.1:63205"
id=65308 trace_id=43 func=__ip_session_run_tuple line=3512 msg="run helper-ftp(dir=original)"
id=65308 trace_id=44 func=print_pkt_detail line=5894 msg="vd-root:0 received a packet(proto=6, 10.10.1.100:21->10.10.1.1:63205) tun_id=0.0.0.0 f"
id=65308 trace_id=44 func=resolve_ip_tuple_fast line=5982 msg="Find an existing session, id-00000dcd, reply direction"
id=65308 trace_id=44 func=__ip_session_run_tuple line=3471 msg="DNAT 10.10.1.1:63205->10.10.1.2:63205"
id=65308 trace_id=44 func=__vf_ip_route_input_rcu line=1990 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=44 func=npu_handle_session44 line=1327 msg="Trying to offloading session from port3 to port3, skb.npu_flag=00000400 ses.state="
id=65308 trace_id=44 func=fw_forward_dirty_handler line=439 msg="state=00000200, state2=00000000, npu_state=00000101"
id=65308 trace_id=44 func=__ip_session_run_tuple line=3458 msg="SNAT 10.10.1.100->101.106.1.1:21"
id=65308 trace_id=44 func=__ip_session_run_tuple line=3512 msg="run helper-ftp(dir=reply)"
```

3. 其他场景下使用策略路由上网时，均可以参考此配置。

解决方法 2

1. 在上网的策略路由 之前添加一条新的策略路由（如下配置中的 ID 2）。流入接口为 port3（LAN），源地址为内网 PC 网段（10.10.1.0/24），目的地址为 FTP Server 的内网 IP。流出接口为内网接口 port3（LAN），网关地址为去往 FTP Server 的下一跳 IP（此例需求场景 2 中的下一跳 IP 就是 FTP Server 的内网 IP）。

> **重要**
>
> 在此例（需求场景 2）中，策略路由 ID 2 的目的地址一定要填写具体的 Server 内网 IP，防止 Server 返回的流量匹配到策略路由。

```bash
config router policy
    edit 2
        set input-device "port3"
        set src "10.10.1.0/255.255.255.0"
        set dst "10.10.1.100/255.255.255.255"
        set gateway 10.10.1.100
        set output-device "port3"
    next
    edit 1
        set input-device "port3"
        set src "10.10.1.0/255.255.255.0"
        set gateway 101.106.1.254
        set output-device "port1"
    next
end
```

2. 此时使用内网 PC 可以正常通过 FTP 服务器的 VIP 公网 IP 访问到 LAN 区域的 FTP 服务器，Debug Flow 信息显示如下：

```
FortiGate # id=65308 trace_id=32 func=print_pkt_detail line=5894 msg="vd-root:0 received a packet(proto=6, 10.10.1.2:63197->101.106.1.1:21) tun_"
id=65308 trace_id=32 func=init_ip_session_common line=6080 msg="allocate a new session-00000ce2, tun_id=0.0.0.0"
id=65308 trace_id=32 func=get_new_addr line=1213 msg="find DNAT: IP-10.10.1.100, port-21"
id=65308 trace_id=32 func=__iprope_tree_check line=535 msg="gnum-100004, use addr/intf hash, len=2"
id=65308 trace_id=32 func=get_new_addr line=1213 msg="find SNAT: IP-101.106.1.1(from IPPOOL), port-63197"
id=65308 trace_id=32 func=fw_pre_route_handler line=184 msg="VIP-10.10.1.100:21, outdev-unknown"
id=65308 trace_id=32 func=__ip_session_run_tuple line=3471 msg="DNAT 101.106.1.1:21->10.10.1.100:21"
id=65308 trace_id=32 func=rpdb_srv_match_input line=1040 msg="Match policy routing id=2: to 10.10.1.100 via ifindex-5"
id=65308 trace_id=32 func=__vf_ip_route_input_rcu line=1990 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=32 func=__iprope_tree_check line=524 msg="gnum-100004, use int hash, slot=29, len=2"
id=65308 trace_id=32 func=fw_forward_handler line=989 msg="Allowed by Policy-2: SNAT"
id=65308 trace_id=32 func=ip_session_confirm_final line=3113 msg="npu_state=0x100, hook=4"
id=65308 trace_id=32 func=__ip_session_run_tuple line=3458 msg="SNAT 10.10.1.2->10.10.1.1:63197"
id=65308 trace_id=32 func=__ip_session_run_tuple line=3512 msg="run helper-ftp(dir=original)"
id=65308 trace_id=33 func=print_pkt_detail line=5894 msg="vd-root:0 received a packet(proto=6, 10.10.1.100:21->10.10.1.1:63197) tun_id=0.0.0.0 f"
id=65308 trace_id=33 func=resolve_ip_tuple_fast line=5982 msg="Find an existing session, id-00000ce2, reply direction"
id=65308 trace_id=33 func=__ip_session_run_tuple line=3471 msg="DNAT 10.10.1.1:63197->10.10.1.2:63197"
id=65308 trace_id=33 func=__vf_ip_route_input_rcu line=1990 msg="find a route: flag=00000000 gw-0.0.0.0 via port3"
id=65308 trace_id=33 func=npu_handle_session44 line=1327 msg="Trying to offloading session from port3 to port3, skb.npu_flag=00000400 ses.state="
id=65308 trace_id=33 func=fw_forward_dirty_handler line=439 msg="state=00000200, state2=00000000, npu_state=00000101"
id=65308 trace_id=33 func=__ip_session_run_tuple line=3458 msg="SNAT 10.10.1.100->101.106.1.1:21"
id=65308 trace_id=33 func=__ip_session_run_tuple line=3512 msg="run helper-ftp(dir=reply)"
```

- TCP SYN 到达 FortiGate 后先做 DNAT 转换（目的地址转换为 FTP Server IP），然后匹配策略路由 ID 2（步骤 4 新增的策略路由），出接口为 port3（LAN）下一跳为 FTP Server，随后执行 SNAT，将源地址转换为 port3（LAN）接口 IP。
- FTP Server 返回 TCP SYN ACK 时，做 DNAT 和 SNAT 还原，返回客户端 PC。

3. 其他场景下使用策略路由上网时，均可以参考此配置。

特殊情况

如果 FortiGate 开启了异步路由（asymroute），且针对内网 PC 和服务器配置了用于上网的策略路由。客户端访问服务器后，服务器返回的流量也会查找策略路由，有可能匹配上上网的策略路由从公网口发出，从而导致返回的流量无法到达客户端。

所以这种情况下一定要针对服务器 → 客户端反向的流量配置策略路由，出接口配置为内网口，并放置在上网策略路由的前边。

VIP46 映射

请参考：IPv6 → NAT46 映射章节。

VIP64 映射

请参考：IPv6 → NAT64 → NAT64 映射、IPv6 → NAT64 → Embedded 模式的 NAT64 和 IPv6 → NAT64 → NAT64 + DNS64 上网章节。

策略常见问题

- 流量到达 FortiGate 后无法转发
- VIP 匹配顺序
- VIP 流量无法匹配防火墙策略
- NAT 端口耗尽
- 地址池 ARP 刷新
- 策略页面加载慢

流量到达 FortiGate 后无法转发

问题现象

1. 某用户通过 FortiGate 访问 Internet，但发现部分 Internet 服务无法访问，对应的 IP 也无法 Ping 通。

2. 检查防火墙上网策略配置全部正确，路由也无异常。

3. 抓包发现流量到达 FortiGate 后没有转发到 Internet（只有 in，没有 out）。

```bash
FortiGate # diagnose sniffer packet any 'host 223.5.5.5' 4
interfaces=[any]
filters=[host 223.5.5.5]
0.400011 lan in 192.168.100.178 → 223.5.5.5: icmp: echo request
1.405165 lan in 192.168.100.178 → 223.5.5.5: icmp: echo request
2.410579 lan in 192.168.100.178 → 223.5.5.5: icmp: echo request
3.412497 lan in 192.168.100.178 → 223.5.5.5: icmp: echo request
```

4. Debug Flow 看到该流量查找路由表的出接口为 root，随后被丢弃。

```bash
id=65308 trace_id=1 func=print_pkt_detail line=5795 msg="vd-root:0 received a packet(proto=1, 192.168.100.178:65060-
>223.5.5.5:2048) tun_id=0.0.0.0 from lan. type=8, code=0, id=65060, seq=744."
id=65308 trace_id=1 func=init_ip_session_common line=5980 msg="allocate a new session-0000139d, tun_id=0.0.0.0"
id=65308 trace_id=1 func=vf_ip_route_input_common line=2611 msg="find a route: flag=80000000 gw-223.5.5.5 via root"
id=65308 trace_id=1 func=fw_local_in_handler line=606 msg="iprope_in_check() check failed on policy 0, drop"
```

问题原因

1. 用户错误的配置了 IPPool 的地址范围为客户端要正常访问的 IP 地址，不论是否为防火墙策略引用。

```bash
config firewall ippool
    edit "IPPool"
        set startip 223.5.5.5
        set endip 223.5.5.5
    next
end
```

2. 或是用户错误的配置了 VIP 的 extip 地址为客户端要正常访问的 IP 地址，不论是否为防火墙策略引用。

```bash
config firewall vip
    edit "VIP"
        set extip 223.5.5.5
        set mappedip "10.10.2.1"
        set extintf "wan1"
    next
end
```

3. VIP 配置中的外部 IP 和 IPPool 中配置的地址范围不能配置为现有网络中其他设备的 IP、FortiGate 的网关 IP，也不能配置为内网客户端正常上网可能访问的 IP 地址。这是由于 FortiGate 配置 VIP 或 IPPool 后，extip、IPPool 地址范围属于 FortiGate 自身的 IP 地址，即使 VIP 或 IPPool 没有被防火墙策略引用，所有访问该 IP 的流量经过 FortiGate 时都会被 FortiGate 引流到自身，影响正常的业务。

解决方法

1. 遇到以上问题时首先使用 Sniffer 抓包和 Debug Flow 排查，如果显示流量被路由到 root 接口，则有可能是以上配置导致。
2. 修改 VIP 中的 extip 或 IPPool 中的地址范围为正确的地址，或删除错误配置的 VIP、IPPool。

VIP 流量无法匹配防火墙策略

问题描述

1. 用户在防火墙策略中配置了一条引用 VIP 作为目标的防火墙策略（ID 6），源接口为 wan1，目标接口为 lan。

```bash
config firewall vip
    edit "VIP_192.168.100.77"
        set extip 10.10.12.5
        set mappedip "192.168.100.77"
        set extintf "wan1"
    next
end

config firewall policy
    edit 6
        set name "VIP_ADC"
        set srcintf "wan1"
        set dstintf "lan"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_192.168.100.77"
        set schedule "always"
        set service "ALL"
    next
end
```

2. 用户想要暂时禁止从 wan1 到 lan 的流量（包括 VIP 流量），在上述策略前创建了一条从 wan1 到 lan 禁止所有流量的策略（ID 8）。

```bash
config firewall policy
    edit 8
        set name "Deny_wan1_to_lan"
        set srcintf "wan1"
        set dstintf "lan"
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic disable
    next
    edit 6
        set name "VIP_ADC"
        set srcintf "wan1"
        set dstintf "lan"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP_192.168.100.77"
        set schedule "always"
        set service "ALL"
    next
end
```

3. 此时外部流量访问 VIP 时，仍然可以通过 VIP 映射正常访问到内网服务器，流量会话匹配的 Policy ID 为 6，拒绝策略的 Policy ID 8 在 Policy ID 6 之上，但仍然匹配了 Policy ID 6。

```bash
session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 flags=00000000 socktype=0 sockport=0 av_idx=0
use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty npu npd
statistic(bytes/packets/allow_err): org=164/3/1 reply=154/2/1 tuples=2
tx speed(Bps/kbps): 10/0 rx speed(Bps/kbps): 10/0
orgin->sink: org pre->post, reply pre->post dev=7->48/48->7 gwy=192.168.100.77/10.10.12.1
hook=pre dir=org act=dnat 10.10.12.1:13034->10.10.12.5:21(192.168.100.77:21)
hook=post dir=reply act=snat 192.168.100.77:21->10.10.12.1:13034(10.10.12.5:21)
pos/(before,after) 0/(0,0), 0/(0,0)
```

```
dst_mac=00:e0:4c:b9:97:7c
misc=0 //policy_id=6// pol_uuid_idx=590 auth_info=0 chk_client_info=0 vd=0
serial=00002ee8 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied helper
```

解决方法

1. 默认配置下（FortiOS 7.2.4 之前的版本），仅当防火墙策略引用的目标地址为 VIP 时，访问 VIP 的流量才会匹配该策略。如果目标为“all”，则流量不会匹配该策略，所以上述场景下，访问 VIP 的流量只会匹配 Policy ID 6。
2. 通过在 Policy ID 8 下开启 `match-vip`，可以使访问 VIP 的流量匹配目标为“all”的策略 Policy ID 8，从而被丢弃。

```bash
config firewall policy
    edit 8
        set match-vip enable
    next
end
```

3. 外部流量访再次问 VIP 时，虽然也执行了 VIP 的 DNAT 转换，但最后会匹配 Policy ID 8 被丢弃。

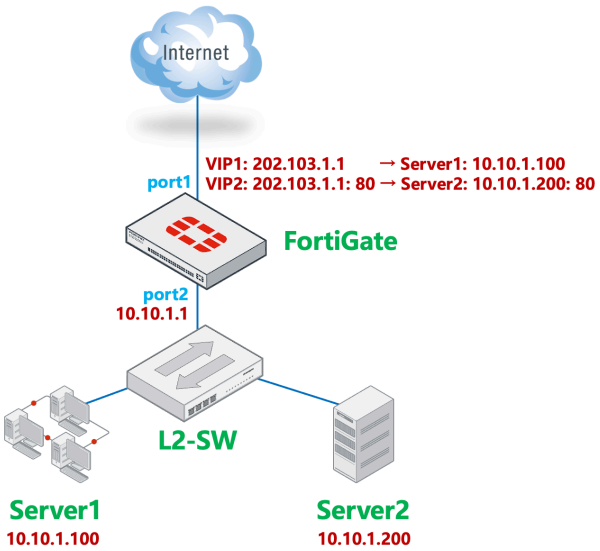
```bash
id=65308 trace_id=13 func=print_pkt_detail line=5795 msg="vd-root:0 received a packet(proto=6, 10.10.12.1:14317->10.10.12.5:21) tun_id=0.0.0.0 from wan1. flag [S], seq 987682832, ack 0, win 65535"
id=65308 trace_id=13 func=init_ip_session_common line=5980 msg="allocate a new session-00003270, tun_id=0.0.0.0"
id=65308 trace_id=13 func=get_new_addr line=1231 msg="find DNAT: IP-192.168.100.77, port-0(fixed port)"
id=65308 trace_id=13 func=fw_pre_route_handler line=180 msg="VIP-192.168.100.77:21, outdev-wan1"
id=65308 trace_id=13 func=__ip_session_run_tuple line=3402 msg="DNAT 10.10.12.5:21→192.168.100.77:21"
id=65308 trace_id=13 func=vf_ip_route_input_common line=2611 msg="find a route: flag=04000000 gw-192.168.100.77 via lan"
id=65308 trace_id=13 func=__iprope_tree_check line=539 msg="gnum-100004, use addr/intf hash, len=4"
id=65308 trace_id=13 func=fw_forward_handler line=825 msg="Denied by forward policy check (policy 8)"
```

注意事项

- 在 FortiOS 7.2.4 及以后版本中，防火墙策略中的 `match-vip` 选项默认为 `enable` 状态。
- 在 FortiOS 7.2.4 之前的版本中，防火墙策略中的 `match-vip` 选项默认为 `disable` 状态。

VIP 匹配顺序

网络拓扑



- 初始状态下，FortiGate 的公网接口 port1 上配置了 1 个 VIP：从公网 IP 202.103.1.1 全映射到 Server1 10.10.1.100（VIP1）。

```bash
config firewall vip
    edit "VIP1"
        set extip 202.103.1.1
        set mappedip "10.10.1.100"
        set extintf "port1"
    next
end
```

- 创建 1 条防火墙策略（VIP1_Policy）引用 VIP1，公网客户端通过 VIP1_Policy 策略访问 Server1。

```bash
config firewall policy
    edit 1
        set name "VIP1_Policy"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP1"
        set schedule "always"
        set service "ALL"
    next
end
```

问题现象

1. 由于业务需求，需要增加与 VIP1 相同的公网 IP 202.103.1.1 的 80 端口映射到 Server2 10.10.1.100 的 80 端口（VIP2），创建后 VIP2 位于 VIP1 后边。

```bash
config firewall vip
    edit "VIP1"
        set extip 202.103.1.1
        set mappedip "10.10.1.100"
        set extintf "port1"
    next
    edit "VIP2"
        set extip 202.103.1.1
        set mappedip "10.10.1.200"
        set extintf "port1"
```

```
        set portforward enable
        set extport 80
        set mappedport 80
    next
end
```

2. 创建防火墙策略 VIP2_Policy，引用 VIP2，将 VIP2_Policy 移动到 VIP1_Policy 之前。当前的需求是，按照防火墙策略的顺序：

   - 访问 VIP 202.103.1.1 的 80 端口的流量应该优先匹配 VIP2_Policy。
   - 访问 VIP 202.103.1.1 的其他端口的流量应该匹配 VIP1_Policy。

```bash
config firewall policy
    edit 2
        set name "VIP2_Policy"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP2"
        set schedule "always"
        set service "ALL"
    next
    edit 1
        set name "VIP1_Policy"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "VIP1"
        set schedule "always"
        set service "ALL"
    next
end
```

3. 但实际情况是：公网客户端（202.103.1.200）访问 VIP（202.103.1.1）的 80 端口（或其他任意端口），流量经过 DNAT 后，目标地址仍然被转换为 VIP1 对应的 Server1 IP 10.10.1.100，会话匹配策略为 VIP1_Policy（ID 1）。这不符合预期。

```bash
session info: proto=6 proto_state=01 duration=10 expire=3589 timeout=3600 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
......
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 202.103.1.200:50883->202.103.1.1:80(10.10.1.100:80) <———DNAT的地址仍为VIP1映射的Server1
hook=post dir=reply act=snat 10.10.1.100:80->202.103.1.200:50883(202.103.1.1:80)
......
misc=0 policy_id=1 pol_uuid_idx=15851 auth_info=0 chk_client_info=0 vd=0
......
```

4. 查看策略 VIP2_Policy（ID 2）的匹配计数，始终为 0。

```bash
FortiGate # diagnose firewall iprope show 100004 2
idx:2
pkts:0 (0 0 0 0 0 0 0 0)
bytes:0 (0 0 0 0 0 0 0 0)
asic_pkts:0 (0 0 0 0 0 0 0 0)
asic_bytes:0 (0 0 0 0 0 0 0 0)
flag:0x0
```

5. 查看策略 VIP1_Policy（ID 1）的匹配计数，持续增长。

```bash
FortiGate # diagnose firewall iprope show 100004 1
idx:1
pkts:282 (282 0 0 0 0 0 0 0)
bytes:15921 (15921 0 0 0 0 0 0 0)
asic_pkts:0 (0 0 0 0 0 0 0 0)
asic_bytes:0 (0 0 0 0 0 0 0 0)
flag:0x0
hit count:54 (54 0 0 0 0 0 0 0)
    first hit:2025-11-05 14:52:19 last hit:2025-11-05 16:27:38
established session count:0
    first est:2025-11-05 15:02:00 last est:2025-11-05 15:53:55
```

6. 通过如下 CLI 命令查看 iprope 表特定流量的防火墙策略匹配情况，可以看到以上访问 VIP 80 端口的流量匹配的就是防火墙策略 VIP1_Policy（ID 1）。

```bash
diagnose firewall iprope lookup <src ip> <src port> <dst ip> <dst port> <protocol> <device>

FortiGate # diagnose firewall iprope lookup 202.103.1.200 0 202.103.1.1 80 6 port1
<src [202.103.1.200-0] dst [202.103.1.1-80] proto 6 dev port1> matches policy id: 1
```

问题原因

- 由于 FortiGate 内核设计，VIP 转换始终优于防火墙策略进行处理：
  - 首先流量到达 FortiGate 后，VIP 按照顺序从上向下依次匹配，并执行 DNAT 转换。
  - 然后防火墙策略才会按照 DNAT 转换后的流量特征，顺序从上向下依次匹配。
- 当全映射 VIP（VIP1）排在 80 端口映射 VIP（VIP2）上方时，所有目标为该公网 IP 的流量均命中全映射 VIP（VIP1），并按照 VIP1 进行 DNAT 转换，导致转换后的流量无法匹配 VIP2_Policy。
- 通过观察防火墙 iprope 的 VIP 策略组（100000）也可以看到，VIP1 在 VIP2 之前：

> **相关信息**
>
> iprope 的相关解释请参考：策略与对象 → iprope 列表 ↗章节。

```bash
FortiGate # diagnose firewall iprope list 100000

policy index=15849 uuid_idx=15849 action=accept ←——VIP1在前
flag (8000104): f_p nat pol_stats
schedule()
cos_fwd=0  cos_rev=0
group=00100000 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 202.103.1.1-202.103.1.1, uuid_idx=15849,
service(1):
        [0:0x0:0/(0,0)->(0,0)] flags:0 helper:auto
nat(1): flag=0 base=202.103.1.1:0 10.10.1.100-10.10.1.100(0:0)

policy index=15850 uuid_idx=15850 action=accept ←——VIP2在后
flag (8000100): nat pol_stats
schedule()
cos_fwd=0  cos_rev=0
group=00100000 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 202.103.1.1-202.103.1.1, uuid_idx=15850,
service(1):
        [6:0x0:0/(0,65535)->(80,80)] flags:0 helper:auto
nat(1): flag=0 base=202.103.1.1:80 10.10.1.200-10.10.1.200(80:80)
......
```

解决方法

1. 调整 VIP 顺序，让映射 80 端口的 VIP（VIP2）排在全映射的 VIP （VIP1）之前，防火墙策略的顺序不需要调整。

```bash
config firewall vip
    move VIP2 before VIP1
end

config firewall vip
    edit "VIP2"
        set extip 202.103.1.1
        set mappedip "10.10.1.200"
        set extintf "port1"
```

```
        set portforward enable
        set extport 80
        set mappedport 80
    next
    edit "VIP1"
        set extip 202.103.1.1
        set mappedip "10.10.1.100"
        set extintf "port1"
    next
end
```

2. 公网客户端（202.103.1.200）再次访问 VIP（202.103.1.1）的 80 端口，流量经过 DNAT 后，目标地址被转换为 VIP2 对应的 Server2 IP 10.10.2.100，会话匹配策略为 VIP2_Policy（ID 2）。符合预期。

```bash
session info: proto=6 proto_state=01 duration=4 expire=3595 timeout=3600 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
......
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 202.103.1.200:58372->202.103.1.1:80(10.10.1.200:80) <———DNAT的地址变为VIP2映射的Server2
hook=post dir=reply act=snat 10.10.1.200:80->202.103.1.200:58372(202.103.1.1:80)
......
misc=0 policy_id=2 pol_uuid_idx=15852 auth_info=0 chk_client_info=0 vd=0
......
```

3. 公网客户端（202.103.1.200）再次访问 VIP（202.103.1.1）的其他端口（如 443），流量经过 DNAT 后，目标地址被转换为 VIP1 对应的 Server1 IP 10.10.1.100，会话匹配策略为 VIP1_Policy（ID 1）。符合预期。

```bash
session info: proto=6 proto_state=06 duration=0 expire=4 timeout=3600 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
......
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 202.103.1.200:58617->202.103.1.1:443(10.10.1.100:443) <———DNAT的地址为VIP1映射的Server1
hook=post dir=reply act=snat 10.10.1.100:443->202.103.1.200:58617(202.103.1.1:443)
......
misc=0 policy_id=1 pol_uuid_idx=15851 auth_info=0 chk_client_info=0 vd=0
......
```

4. 查看策略 VIP2_Policy（ID 2）已经开始有匹配计数。

```bash
FortiGate # diagnose firewall iprope show 100004 2
idx:2
pkts:46 (46 0 0 0 0 0 0 0)
bytes:2029 (2029 0 0 0 0 0 0 0)
asic_pkts:0 (0 0 0 0 0 0 0 0)
asic_bytes:0 (0 0 0 0 0 0 0 0)
flag:0x0
hit count:3 (3 0 0 0 0 0 0 0)
    first hit:2025-11-05 16:44:31 last hit:2025-11-05 16:50:40
established session count:1
    first est:2025-11-05 16:44:31 last est:2025-11-05 16:50:40
```

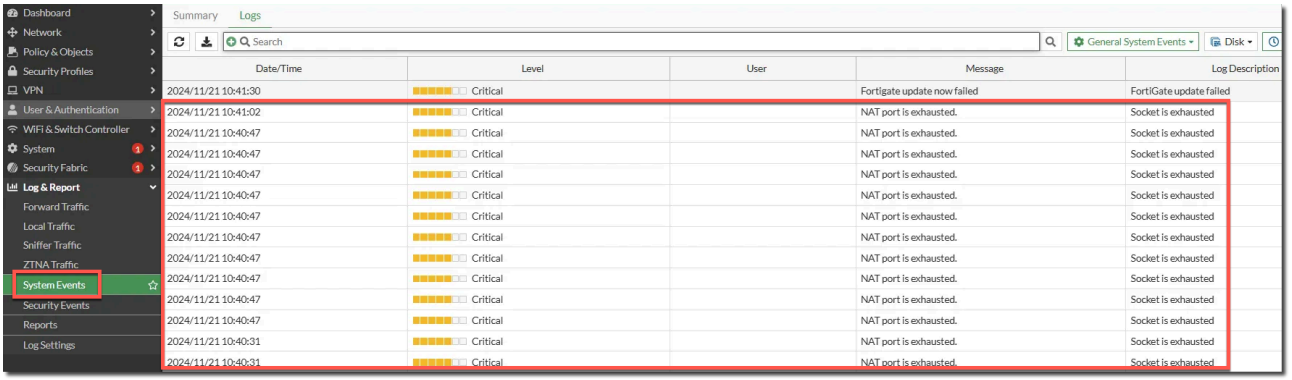
5. 查询 iprope 表特定流量的防火墙策略匹配情况，可以看到访问 VIP 80 端口的流量匹配的是防火墙策略 VIP2_Policy（ID 2）。

```bash
diagnose firewall iprope lookup <src ip> <src port> <dst ip> <dst port> <protocol> <device>

FortiGate # diagnose firewall iprope lookup 202.103.1.200 0 202.103.1.1 80 6 port1
<src [202.103.1.200-0] dst [202.103.1.1-80] proto 6 dev port1> matches policy id: 2
```

NAT 端口耗尽

问题现象

FortiGate 部署在 Internet 出口，并开启 NAT 功能实现内部用户上网。在使用过程中 FortiGate 产生了 `NAT port is exhausted. Socket is exhausted` 的系统事件日志。




```bash
date="2024-11-27" time="11:20:32" devid="FG101FTK2000XXXX" vd="root" type="event" subtype="system" bid=540114651
devname="FG101FTK2000XXXX" dstepid=3 dsteuid=3 dstip="8.8.8.8" dstport=53 dvid=2147 epid=3 euid=3
eventtime=1732492832315428126 id=7242716162855799960 level="critical" logdesc="Socket is exhausted" logid="0100020007"
logver=604065868 msg="NAT port is exhausted." nat="33.43.111.222" proto=17 service="kernel" srcip="10.200.100.50"
srcport=15651 status="failure" tz="-0500" vrf=0
```

问题原因

在 FortiGate 上使用 `同一SNAT IP + 同一目的IP + 同一协议 + 同一目的端口`（以下简称 **四元组**）时，支持 60416 个连接。如果同一 **四元组**已经存在了 60416 个连接，此时需要多一个连接，那么就会出现 NAT 端口耗尽的情况，新的连接无法建立，并产生以上日志。

对于其他连接，只要 **四元组**没有达到 60416 个连接，就不会产生以上问题。所以只要其他流量中的 SNAT IP、目的 IP、协议、目的端口有任意一项与已耗尽的 **四元组**不同，业务就不会受到影响。

FortiGate 端口使用规则

1. FortiOS 使用以上 **四元组**来确定 NAT 可用池的原因是 FortiGate 是基于会话的防火墙。 流量通过 FortiGate 设备进行连接时，会创建一个会话，并为该会话创建两个索引。 FortiGate 设备使用这些索引将匹配流量引导至匹配的会话。

2. 使用这些属性来确定池的原因是 FortiOS 防火墙基于会话的设计。 通过 FortiGate 设备进行 TCP 连接时，会创建一个会话，并为该会话创建 2 个索引， FortiGate 设备使用这 2 个索引将匹配流量引导至会话：

   -**索引 1**，用于流量发起方向的会话标识。

   ```
   src-ip: 10.10.1.1
   dst-ip: 202.1.1.1
   proto: TCP
   src-port: 10240
   dst-port: 443
   ```

   -**索引 2**，用于返回流量的会话标识。

   ```
   src-ip: 202.1.1.1
   dst-ip: 101.1.1.1
   proto: TCP
   src-port: 80
   dst-port: 46372
   ```

   - 其中 46372 是发起方向流量 SNAT 后 FortiGate 分配的端口。有了这 2 个索引，当流量与其中任一索引匹配时，流量所属的会话就能被唯一匹配。

3. FortiGate 在分配 NAT 端口时，只需确保所选 NAT 端口与 **四元组**相结合是唯一的，以便唯一标识会话。

4. 举例，在以上流量已经创建会话的情况下，内网同一客户端 10.10.1.1 再与同一公网服务器 202.1.1.1 的 443 端口建立连接，此时 FortiGate 将创建另一个会话，返回方向上的索引如下。

```
src-ip: 202.1.1.1
dst-ip: 10.10.1.1
proto: TCP
src-port: 443
dst-port: NP
```

5. NP 的值可以是任何值，只要五个值加起来是唯一的。 例如，FortiGate 可以再次选择 46372，因为 `src-port` 不一样。

```
src-ip: 202.1.1.1
dst-ip: 10.10.1.1
proto: TCP
src-port: 443
dst-port: 46372
```

6. 所以同一 **四元组**的情况下 FortiGate 可以建立的连接数，取决于 `src-port` 的数量，FortiGate 支持的最大值为 60416。

7. 举例，如果访问的目的 IP 有 3 个，那么 `同一SNAT IP + 同一目的端口` 最多可以在 FortiGate 上产生多少个连接？答案是 `60416 * 3 = 181248`。

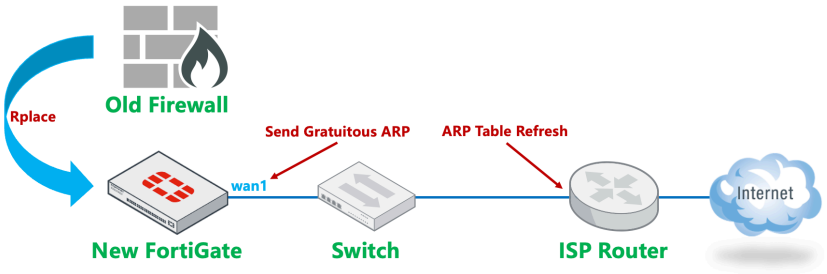
地址池 ARP 刷新

背景

在网络设备替换场景中（例如原厂商设备被 FortiGate 替换），如果设备的 SNAT 使用了 IP Pool（地址池），可能会出现如下问题：

- 新防火墙上线后，IP Pool 不会主动发送 Gratuitous ARP（免费 ARP）。
- 上游路由器或交换机的 ARP 缓存仍保留旧防火墙的 MAC。
- 因此下行流量仍发往旧 MAC，新防火墙收到后直接丢弃。表现为：上行能通，下行不通。
- 割接时无法联系运营商关闭路由器接口，导致无法更新运营商路由器的 ARP 表。

网络环境



- 使用 FortiGate 替换原有旧防火墙。
- FortiGate 上线后，上游 ISP 路由器的 ARP 缓存仍保留旧防火墙的 MAC。
- 新 FortiGate 通过 IP Pool 配置 SNAT，使内网客户端访问 Internet。

```bash
config system interface
    edit wan1
        set ip 192.168.1.2/24
        set allowaccess ping
    next
end

config firewall ippool
    edit "pool1"
        set startip 192.168.1.11
        set endip   192.168.1.20
        set associated-interface wan1
    next
end

config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr "LAN"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
        set ippool enable
        set poolname "pool1"
    next
end
```

方法 1 - HA 切换

如果设备为 HA 组网，HA 主备切换会触发 ARP 更新，FortiGate 会对所有 VIP、接口 IP、IP Pool 地址广播 Gratuitous ARP。

1. 确认 HA 已启用 Gratuitous ARP（默认即启用）。

```bash
config system ha
    set gratuitous-arps enable
    set arps 5
    set arps-interval 8
end
```

2. 在当前 HA 主设备上手动触发一次 HA 切换。

```bash
diagnose sys ha reset-uptime
```

3. 切换完成后，在新的 HA 主设备上抓包验证，可以看到设备根据 IP Pool 中的地址发送了对应的 Gratuitous ARP。

```bash
FortiGate # diagnose sniffer packet port3 "arp" 4 0 l

2025-10-13 20:57:17.298321 port3 -- arp who-has 192.168.1.11 tell 192.168.1.11
2025-10-13 20:57:17.298331 port3 -- arp who-has 192.168.1.12 tell 192.168.1.12
2025-10-13 20:57:17.298340 port3 -- arp who-has 192.168.1.13 tell 192.168.1.13
...
2025-10-13 20:57:49.206136 port3 -- arp who-has 192.168.1.18 tell 192.168.1.18
2025-10-13 20:57:49.206143 port3 -- arp who-has 192.168.1.19 tell 192.168.1.19
2025-10-13 20:57:49.206149 port3 -- arp who-has 192.168.1.20 tell 192.168.1.20
```

方法 2 - 临时创建 VIP

FortiGate 的 VIP 支持参数 `gratuitous-arp-interval`，可主动周期性广播 Gratuitous ARP。可临时将 IP Pool 地址段配置为 VIP 对象，以此强制发送 Gratuitous ARP 来刷新上游路由器 ARP 表项。

1. 创建 VIP 对象，地址范围包含 IP Pool 中的地址（可根据 IP Pool 数量配置多个 VIP）。

重要

- `extip`：填写 IP Pool 中的地址。
- `mappedip`：可随意填写。
- `extintf`：填写连接运营商侧的接口。
- `gratuitous-arp-interval`：Gratuitous ARP 的更新间隔，范围 5~86400s，设置为 0 表示不发送 Gratuitous ARP。注意间隔设置过小可能触发上游安全告警。

```bash
config firewall vip
    edit arp-refresh
        set extip 192.168.1.11-192.168.1.20
        set mappedip 192.168.3.11-192.168.3.20
        set extintf wan1
        set portforward enable
        set gratuitous-arp-interval 5
        set extport 12345
        set mappedport 12345
    next
end
```

2. 配置完成后抓包验证，设备根据 VIP 中的地址发送了对应的 Gratuitous ARP。

```bash
FortiGate # diagnose sniffer packet wan1 'arp' 4 0 l
interfaces=[port3]
filters=[arp]
2025-10-13 21:50:46.230249 port3 -- arp who-has 192.168.1.11 tell 192.168.1.11
2025-10-13 21:50:46.230257 port3 -- arp who-has 192.168.1.12 tell 192.168.1.12
2025-10-13 21:50:46.230262 port3 -- arp who-has 192.168.1.13 tell 192.168.1.13
...
```

3. 确认上游路由器 ARP 更新完成后，删除该 VIP。

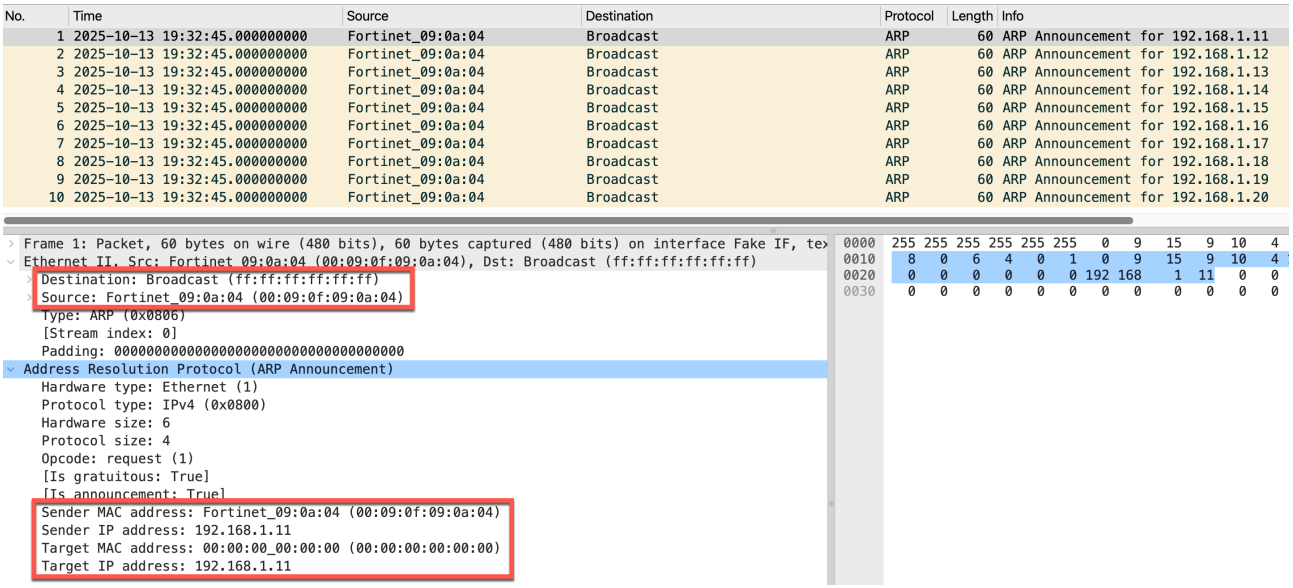
```bash
config firewall vip
    delete arp-refresh
end
```

Gratuitous ARP 原理

报文示例

Gratuitous ARP 的报文示例（gratuitous_arp.pcap ↗）。

报文特征：Gratuitous ARP = Sender IP == Target IP 且 Destination MAC == FF:FF:FF:FF:FF:FF。



主要作用

- 刷新上游交换机或路由器的 ARP 表。
- 通知邻居设备该 IP 地址的 MAC 已变更。
- 检测 IP 冲突（同网段若有重复地址，会收到响应）。

字段	特征	说明
操作码（Opcode）	1 (request) 或 2 (reply)	GratuitousARP 可为 Request 或 Reply 形式
Sender IP (SPA)	本机 IP	表示自己拥有的 IP 地址
Target IP (TPA)	与 Sender IP 相同	这是 Gratuitous ARP 的关键特征之一
目标 MAC (THA)	全 0 (00:00:00:00:00:00)	因为不是响应任何主机，所以目标 MAC 为空
源 MAC (SHA)	本机 MAC 地址	通常是接口的真实 MAC
以太网目标地址	FF:FF:FF:FF:FF:FF（广播）	告知整个二层网段的所有设备
以太网类型 (Ethertype)	0x0806	表示这是一个 ARP 报文

Gratuitous ARP 的两种类型

类型	Opcode	特点	主要用途
ARP Request 型	1	“谁有我的地址？”	通知邻居更新 ARP 表
ARP Reply 型	2	“我就是这个地址”	某些厂商采用这种形式广播自己的存在

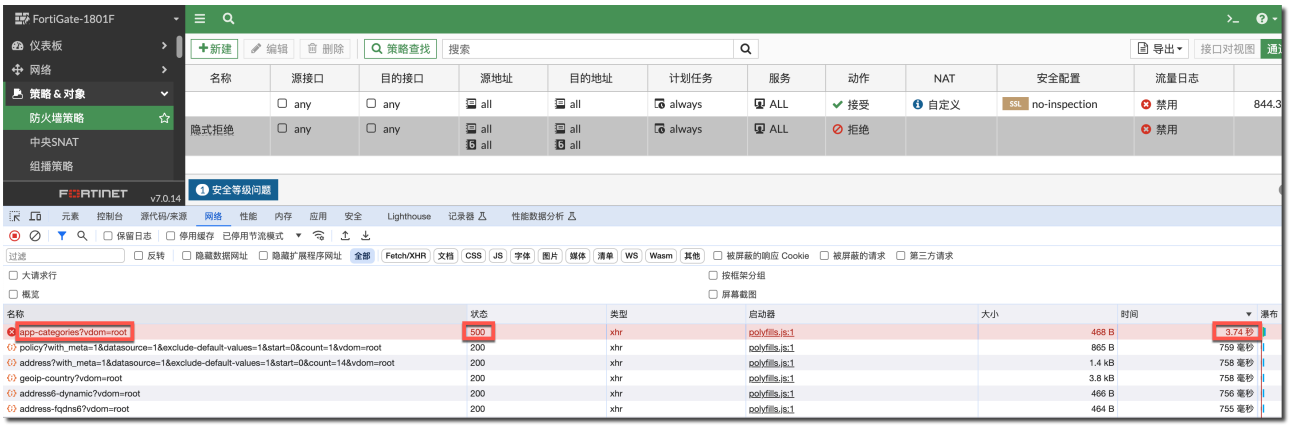
策略页面加载慢

问题现象

> **重要**
>
> 该问题仅存在于 FortiOS 7.0.X 版本，FortiOS 7.2 和 7.4 版本不存在此问题。

访问 FortiGate 的防火墙策略（Firewall Policy）、中央 SNAT（Central SNAT）、IPv4 访问控制列表（IPv4 Access Control List）、零信任网络（ZTNA）、组播策略（Multicast Policy）、代理策略（Proxy Policy）、流量整形策略（Traffic Shaping Policies）、应用签名（Application Signatures）等策略相关页面时加载较慢，需要 3~5 秒才可以加载出来。

使用浏览器的开发者工具可以看到导致页面加载缓慢的资源是 `/api/v2/monitor/fortiguard/app-categories?vdom=root`，并返回 HTTP 500 错误。



问题原因

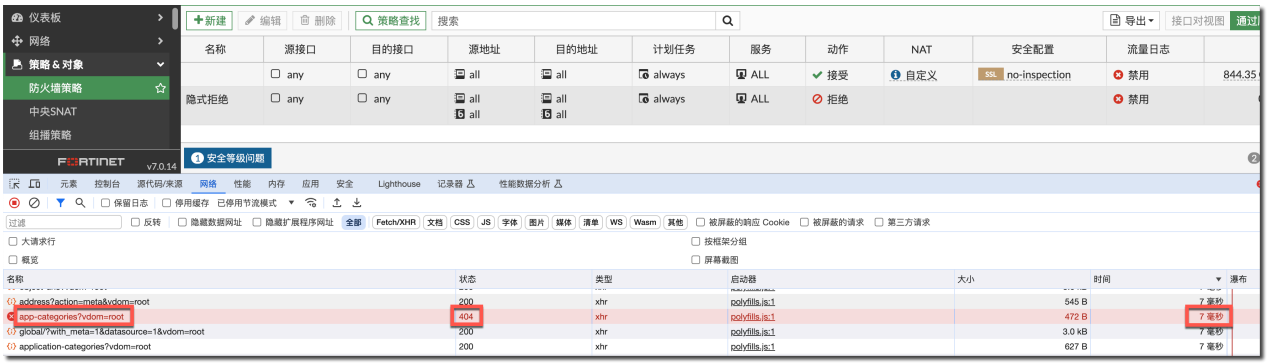
这个问题是由于通过 Web 方式访问这些页面时，需要从 FortiGuard 加载 `app-categories` 资源，如果 FortiGate 此时无法访问 FortiGuard，则会花费 3~5 秒等待回应，直至返回 HTTP 500（Internal Server Error）错误后才能显示页面。

解决方法

1. 确认 FortiGate 可以正常访问 Internet，解决 FortiGate 与 FortiGuard 之间的连通性。
2. 如果 FortiGate 无法访问 Internet 或与 FortiGuard 通讯存在问题，可以在 CLI 下配置如下功能，关闭 `gui-fortiguard-resource-fetch` 后，访问这些页面时 FortiGate 不再会向 FortiGuard 请求 `app-categories` 资源，直接显示相关页面。

```bash
config system global
    set gui-fortiguard-resource-fetch disable
end
```

3. 再次访问 FortiGate 的防火墙策略页面，`app-categories` 资源可以迅速加载（直接返回 HTTP 404）。



本地流量策略

- Local-in 策略
- Local-out 流量控制
- Preferred Source

Local-in 策略

Local-in Policy 介绍

普通的防火墙策略仅能针对穿越防火墙接口的流量生效，针对目标是 FortiGate 自身地址（且未穿越 FortiGate 接口）的流量是不起作用的。Local-in Policy 可以控制发送到 FortiGate 接口的入方向（未穿越 FortiGate 接口）流量。

> **重要**
>
> 即使流量的目标是 FortiGate 自身接口的地址，但穿越了 FortiGate 的接口，就不算 Local-in 流量，可以被普通防火墙策略控制，例如穿越 FortiGate 的 WAN 接口访问设备的内网口的流量。

管理流量（如 HTTPS、PING、SSH 等）可以在接口中配置允许或拒绝。受信任的管理主机可以在管理员选项中配置，以限制可以访问管理服务的主机。Local-in Policy 可以自定义 Local-in 流量的源地址、目的地址、接口和服务。以控制以上功能无法控制的入方向本地流量：

- 可用于限制管理员访问或其他服务，例如 SSL VPN 服务，定义源地址或地址组来限制访问。例如，通过使用地理位置类型的地址，限制某个特定地理区域的 IP 访问 FortiGate。也可以将“Security Fabric → 外部连接器”中配置的“IP 地址威胁来源”用作源地址或目的地址。
- 可以使用虚拟补丁（Virtual Patch）来减少针对 FortiGate 的已知漏洞的攻击。漏洞规则会在指定接口上对本地入方向流量进行扫描，并丢弃所有匹配攻击特征的流量。关于虚拟补丁（Virtual Patch）的使用方法，请参考虚拟补丁↗章节。

Local-in Policy 配置

> **重要**
>
> - Local-in Policy 不像普通的防火墙策略，普通防火墙策略默认隐含策略的动作为 deny all，而 Local-in Policy 默认隐含策略的动作为 permit all。
> - 在 FortiOS 7.4.6、7.6.1 及以后版本中，如果接口加入了 SD-WAN 区域，Local-in Policy 只能引用接口所属的 SD-WAN 区域，不能引用独立的接口；如果之前的版本已经配置了 Local-in Policy 并引用属于 SD-WAN 区域的某个成员接口，升级到 FortiOS 7.4.6、7.6.1 及以后版本后，需要重新配置 Local-in Policy 才能生效。

```bash
config firewall {local-in-policy | local-in-policy6}
    edit <policy_number>
        set intf <interface>
        set srcaddr <source_address> [source_address] ...
        set srcaddr-negate {enable | disable}    //enable表示排除选择的源地址，默认为disable//
        set dstaddr <destination_address> [destination_address] ...
        set dstaddr-negate {enable | disable}    //enable表示排除选择的目标地址，默认为disable//
        set action {accept | deny}
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set virtual-patch {enable | disable}
        set comments <string>
    next
end
```

配置步骤

举例 1

用户在公网口 port1 上开启了 SSL VPN 功能。发现某些异常 IP 地址段经常尝试登录 SSL VPN，但登录失败，并产生 SSL VPN 相关日志。用户想要禁止这些 IP 地址段尝试登录 SSL VPN，不要产生 SSL VPN 登录日志。

1. 配置异常 IP 地址段的地址对象和地址组对象，例如尝试登录的异常源 IP 段为 117.53.1.0/24、27.1.5.0/24 和 113.25.0.0/16，并使用地址组引用。

```bash
config firewall address
    edit "117.53.1.0/24"
        set subnet 117.53.1.0 255.255.255.0
    next
    edit "27.1.5.0/24"
        set subnet 27.1.5.0 255.255.255.0
    next
    edit "113.25.0.0/16"
        set subnet 113.25.0.0 255.255.0.0
    next
```

```
end

config firewall addrgrp
    edit "SSL_VPN_Deny_IP"
        set member "117.53.1.0/24" "27.1.5.0/24" "113.25.0.0/16"
    next
end
```

2. 配置用户使用的 SSL VPN 端口的服务对象，例如用户配置的 SSL VPN 端口为 10443。

```
config firewall service custom
    edit "SSL_VPN_10443"
        set tcp-portrange 10443
    next
end
```
bash

3. 配置 Local-in Policy 禁止这些异常 IP 访问 FortiGate 的 SSL VPN 服务。

```
config firewall local-in-policy
    edit 1
        set intf "port1"
        set srcaddr "SSL_VPN_Deny_IP"
        set dstaddr "all"
        set action deny
        set service "SSL_VPN_10443"
        set schedule "always"
    next
end
```
bash

4. 使用 Debug Flow 查看限制网段发起的 SSL VPN 访问，流量被 Local-in Policy（ID 1）阻断，且不再触发 SSL VPN 日志生成。

```
id=65308 trace_id=12 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 117.53.1.37:16008-
>202.103.12.2:10443) tun_id=0.0.0.0 from port1. flag [S], seq 4019479492, ack 0, win 29200"
id=65308 trace_id=12 func=init_ip_session_common line=6073 msg="allocate a new session-0000004e"
id=65308 trace_id=12 func=__vf_ip_route_input_rcu line=1999 msg="find a route: flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=12 func=fw_local_in_handler line=618 msg="iprope_in_check() check failed on policy 1, drop"
```
bash

5. 使用 Debug Flow 查看非限制网段的 IP 访问 SSL VPN 服务，可以正常访问 TCP 10443 端口。

```
id=65308 trace_id=39 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 202.103.12.1:16204-
>202.103.12.2:10443) tun_id=0.0.0.0 from port1. flag [S], seq 2461158975, ack 0, win 29200"
id=65308 trace_id=39 func=init_ip_session_common line=6073 msg="allocate a new session-0000003d"
id=65308 trace_id=39 func=__vf_ip_route_input_rcu line=1999 msg="find a route: flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=39 func=ip_session_confirm_final line=3111 msg="npu_state=0×0, hook=1"
id=65308 trace_id=40 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 202.103.12.2:10443-
>202.103.12.1:16204) tun_id=0.0.0.0 from local. flag [S.], seq 1646480095, ack 2461158976, win 28960"
id=65308 trace_id=40 func=resolve_ip_tuple_fast line=5976 msg="Find an existing session, id-0000003d, reply direction"
...
```
bash

6. 查看相关 Local-in 会话。

```
session info: proto=6 proto_state=01 duration=2 expire=3597 timeout=3600 refresh_dir=both flags=00000000 socktype=3
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 50/0 rx speed(Bps/kbps): 27/0
orgin->sink: org pre->in, reply out->post dev=3->9/9->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 202.103.12.1:16208->202.103.12.2:10443(0.0.0.0:0)
hook=post dir=reply act=noop 202.103.12.2:10443->202.103.12.1:16208(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00000042 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```
bash

举例 2

用户在公网口 port1 上开启了 SSH 访问功能。发现经常有国外 IP 尝试通过 SSH 登录 FortiGate，但登录失败。用户想要禁止国外的 IP 访问 SSH，但允许国内 IP 访问 SSH。

1. 配置基于地理位置的国内 IP 地址对象。

```bash
config firewall address
    edit "China"
        set type geography
        set country "CN"
    next
end
```

2. 配置 Local-in Policy 允许国内 IP 的客户端访问 FortiGate 的 SSH 服务，但不允许其他 IP 访问 FortiGate 的 SSH 服务。

```bash
config firewall local-in-policy
    edit 1
        set intf "port1"
        set srcaddr "China"
        set dstaddr "all"
        set action accept
        set service "SSH"
        set schedule "always"
    next
    edit 2
        set intf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action deny
        set service "SSH"
        set schedule "always"
    next
end
```

3. 使用 Debug Flow 查看国内 IP 的 SSH 访问，可以正常访问。

```bash
id=65308 trace_id=1 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 202.103.12.1:18768->202.103.12.2:22) tun_id=0.0.0.0 from port1. flag [S], seq 3436667351, ack 0, win 29200"
id=65308 trace_id=1 func=init_ip_session_common line=6073 msg="allocate a new session-0000005a"
id=65308 trace_id=1 func=__vf_ip_route_input_rcu line=1999 msg="find a route: flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=1 func=ip_session_confirm_final line=3111 msg="npu_state=0×0, hook=1"
id=65308 trace_id=2 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 202.103.12.2:22->202.103.12.1:18768) tun_id=0.0.0.0 from local. flag [S.], seq 4214370010, ack 3436667352, win 28960"
id=65308 trace_id=2 func=resolve_ip_tuple_fast line=5976 msg="Find an existing session, id-0000005a, reply direction"
...
```

4. 查看相关 Local-in 会话。

```bash
session info: proto=6 proto_state=01 duration=9 expire=3590 timeout=3600 refresh_dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=164/3/1 reply=127/2/1 tuples=2
tx speed(Bps/kbps): 17/0 rx speed(Bps/kbps): 13/0
orgin->sink: org pre->in, reply out->post dev=3->9/9->3 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 202.103.12.1:18782->202.103.12.2:22(0.0.0.0:0)
hook=post dir=reply act=noop 202.103.12.2:22->202.103.12.1:18782(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=15846 auth_info=0 chk_client_info=0 vd=0
serial=00000092 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

5. 使用 Debug Flow 查看国外 IP 的 SSH 访问，流量被 Local-in Policy（ID 2）阻断。

```
id=65308 trace_id=1 func=print_pkt_detail line=5888 msg="vd-root:0 received a packet(proto=6, 35.1.2.3:10648->35.1.2.1:22) tun_id=0.0.0.0 from port1. flag [S], seq 891389291, ack 0, win 29200"
id=65308 trace_id=1 func=init_ip_session_common line=6073 msg="allocate a new session-0000008c"
id=65308 trace_id=1 func=__vf_ip_route_input_rcu line=1999 msg="find a route: flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=1 func=fw_local_in_handler line=618 msg="iprope_in_check() check failed on policy 2, drop"
```

bash

举例 3

在 HA 环境下，用户开启并配置了 HA 独立管理口，需要在 HA 独立管理口下配置 Local-in Policy。

1. 查看 FortiGate HA 的独立管理口配置，独立管理口为 internal1。

```
config system ha
    config ha-mgmt-interfaces
        edit 1
            set interface "internal1"
        next
    end
end
```

bash

2. 在 internal1 上配置 Local-in Policy，入接口无法选择 HA 独立管理口 internal1。

```
FortiGate # config firewall local-in-policy
FortiGate (local-in-policy) # edit 1
new entry '1' added
FortiGate (1) # set intf ?
<string>    Please input string value.
any             Match any interface in the virtual domain.
dmz     interface
fortilink       interface
internal        interface
internal4       interface
internal5(to_FW2_90D_internal_12_13)    interface
l2t.root        interface
naf.root        interface
ssl.root(SSL VPN interface)     interface
to_Home_WAN1    interface
to_Home_WAN2    interface
wan1    interface
wan2    interface
```

bash

3. 在 HA 独立管理口上配置 Local-in Policy 时，需要开启 `ha-mgmt-intf-only` 选项，配置为 enable 后，Local-in Policy 可以在 HA 独立管理口上正常下发。

```
config firewall local-in-policy
    edit 1
        set ha-mgmt-intf-only enable
        set intf "internal1"
        set srcaddr "Beijing_LAB_88"
        set dstaddr "all"
        set service "HTTPS"
        set schedule "always"
    next
end
```

bash

TTL Policy

通过配置 TTL Policy，可以阻止具有较高 TTL 的攻击流量。此功能与 Local-in Policy 一样仅用于本地入向流量，不适用于穿越 FortiGate 接口的流量，使用 srcintf 设置本地入向流量命中的接口（不能用于 HA 独立管理口）。

```
config firewall ttl-policy
    edit <id>
        set status {enable | disable}
        set action {accept | deny}
        set srcintf <interface>
        set srcaddr <source_address> [source_address] ...
```

bash

```
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set ttl <value/range>
    next
end
```

Local-out 流量控制

Local-out 流量指的是源自 FortiGate 并发往外部目标地址的流量。这种流量可能来自 Syslog、FortiAnalyzer 日志记录、FortiGuard 服务、远程认证等。默认情况下，Local-out 流量根据当前路由表来查找用于流量的出接口。但有些情况下，查询路由表的结果可能是去往目标 IP 的路径有多个出接口负载，但只有其中一个接口可以正确的到达目标地址。这种情况下如果负载到了错误的出接口，那么业务就会出现问题。

很多类型的 Local-out 流量支持根据 SD-WAN 规则或手工指定的接口作为出接口。在手工指定出口接口时，也可以指定源 IP 地址。支持设置出接口的 Local-out 流量类型如下所示：

1. Security Fabric → 外部连接器（External Resources）中的终端/身份（Endpoint/Identity，可通过 CLI 配置）、威胁来源等（Threat Feeds，可通过 GUI 和 CLI 配置）。
2. 日志中的 FortiAnalyzer 服务器、FortiAnalyzer Cloud 服务器、FortiGate Cloud 日志服务器、Syslog 服务器等。（可通过 GUI 和 CLI 配置）
3. 系统 DNS 服务器、FortiGuard 服务器、FortiSandbox 服务器等。（可通过 GUI 和 CLI 配置）
4. LDAP 服务器、TACACS + 服务器、Radius 服务器等。（可通过 CLI 配置）

> **重要**
>
> 如果 FortiGate 开启了多 VDOM 功能，以上内容中 1、2、3 需要在 Global 视图下配置，4 需要在 VDOM 视图下配置。

Local-out 流量配置

通过 GUI 配置

DNS 流量

用户 FortiGate 设备路由表中有两条等价的默认路由，出接口分别为 wan1 和 wan2，只有 wan1 可以正常到达 DNS 服务器，DNS 流量默认情况下在 wan1 和 wan2 上负载，当负载到 wan2 时，DNS 解析会失败。

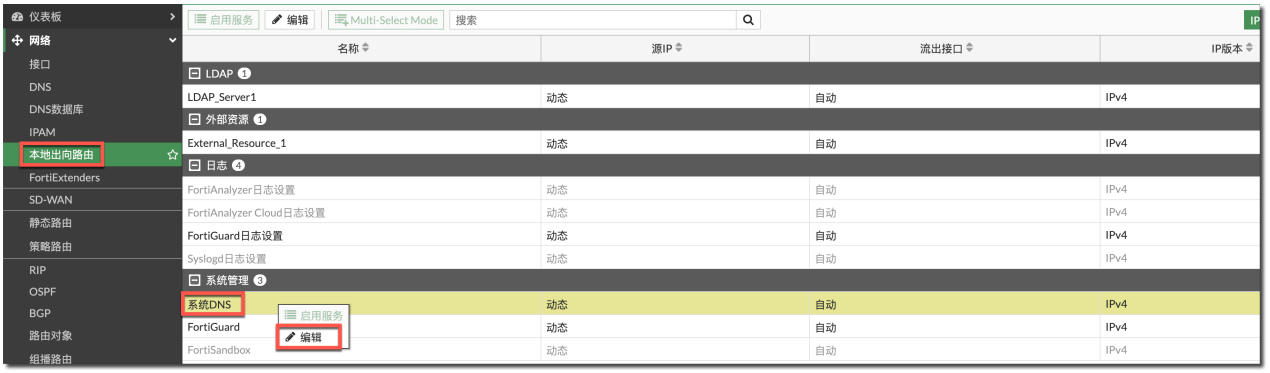
```bash
FortiGate # get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.22.5.1, wan2, [1/0]
                 [10/0] via 172.22.6.1, wan1, [1/0]
...
```

1. 在 GUI 上配置 Local-out 流量控制，首先要在“系统管理 → 可见功能”中开启“本地出向流量路由”。



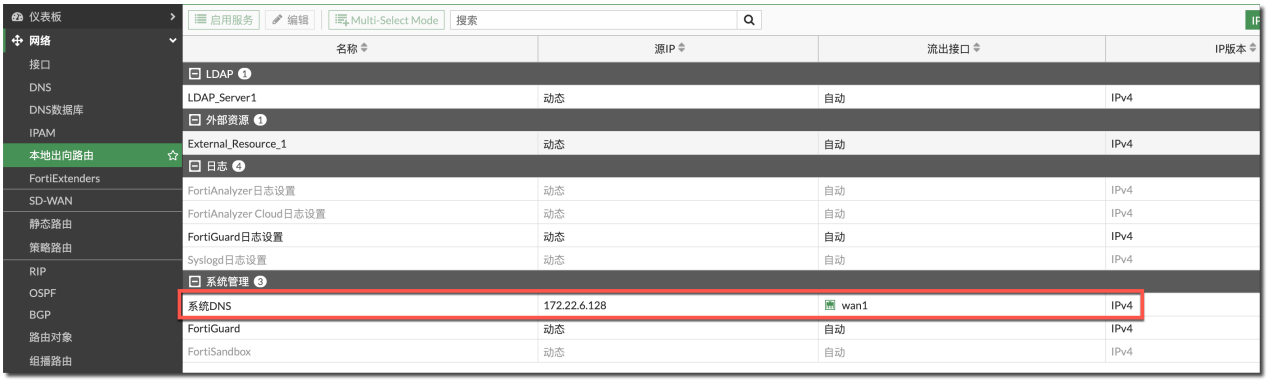

```bash
config system global
    set gui-local-out enable
end
```

2. 进入网络 → 本地出向路由页面，编辑系统 DNS 选项。



3. 配置系统 DNS 的流出接口为指定模式，并选择 wan1 作为出接口（如果 wan1 上有多个 IP，也可以修改源地址），点击确认下发配置。






```bash
config system dns
    set interface-select-method specify
    set interface "wan1"
end
```

LDAP 流量

用户 FortiGate 设备路由表中有两条去往 LDAP 服务器所在网段的路由，出接口分别为 internal4 和 internal5，只有 internal5 可以正常到达 LDAP 服务器，LDAP 流量默认情况下在 internal4 和 internal5 上负载，当负载到 internal4 时，LDAP 会无法访问。

```bash
FortiGate # get router info routing-table static

S       192.168.100.0/24 [10/0] via 10.10.12.2, internal4, [1/0]
                         [10/0] via 10.10.13.2, internal5, [1/0]
...
```

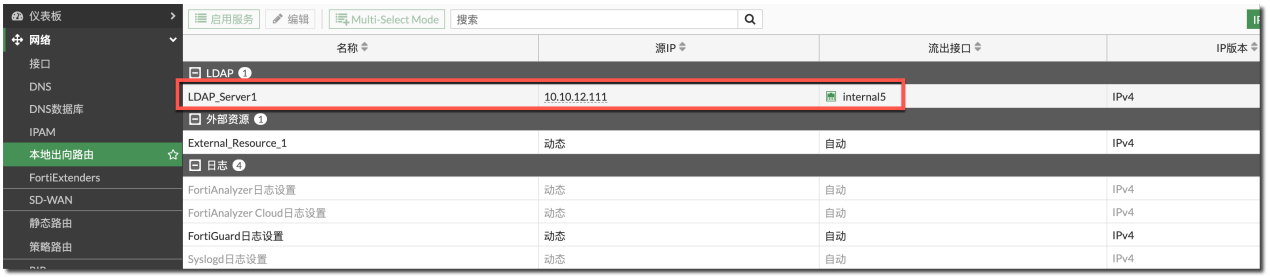
同时，LDAP 服务器只允许 FortiGate 以 10.10.12.111 作为源 IP 访问。

1. 进入网络 → 本地出向路由页面，编辑系统 LDAP 服务器选项（只有配置了 LDAP/Radius/Tacacs + 服务器后，才会在这里显示）。



2. 配置系统 LDAP 服务器的流出接口为指定模式，并选择 internal5 作为出接口，选择源 IP 为手动模式，选择 internal5 接口的第二地址 10.10.12.111，点击确认下发配置。






```bash
config user ldap
    edit "LDAP_Server1"
        set source-ip "10.10.12.111"
        set interface-select-method specify
        set interface "internal5"
    next
end
```

通过 CLI 配置

PING & Traceroute

ping 和 traceroute 命令除了可以指定接口和源地址，也可以指定是否使用 SD-WAN 规则发送。

```bash
execute ping-options use-sdwan {yes | no}
execute ping6-options use-sd-wan {yes | no}
execute traceroute-options use-sdwan {yes | no}
```

集中管理

连接 FortiManager 进行集中管理的流量。

```bash
config system central-management
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
end
```

NTP 服务器

```bash
config system ntp
    config ntpserver
        edit <id>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

- `auto`：查找路由表进行转发。
- `sdwan`：查找 SD-WAN 规则或策略路由进行转发。
- `specify`：手工指定接口转发。
- `interface`：`interface-select-method` 配置为 `specify` 时可用，用于手工指定流量的出接口。

DHCP 代理

```bash
config system settings
    set dhcp-proxy-interface-select-method {auto | sdwan | specify}
    set dhcp-proxy-interface <interface>
end
```

DHCP 中继

```bash
config system interface
    edit <interface>
        set dhcp-relay-interface-select-method {auto | sdwan | specify}
        set dhcp-relay-interface <interface>
    next
end
```

证书 SCEP

```bash
config vpn certificate setting
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
    set source-ip <IP address>
end
```

IPS TLS 协议探测

```bash
config ips global
    config tls-active-probe
        set interface-selection-method {auto | sdwan | specify}
        set interface <interface>
        set vdom <VDOM>
        set source-ip <IPv4 address>
        set source-ip6 <IPv6 address>
    end
end
```

- `VDOM`：当 `interface-selection-method` 设置为 `sdwan` 或 `specify` 时，可以指定流量的 VDOM。

NetFlow/sFlow

```bash
NetFlow:
config system {netflow | vdom-netflow}
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
    set source-ip <IP address>
end

sFlow:
config system {sflow | vdom-sflow}
    config collectors
        edit <id>
            set interface-select-method {auto | sdwan | specify}
```

```
            set interface <interface>
            set source-ip <IP address>
        next
    end
end
```

FortiClient EMS

```bash
config endpoint-control fctems
    edit fctems1
        set interface-select-method {auto | sdwan | specify}
        set interface <interface>
        set source-ip <IP address>
    end
end
```

TACACS+

```bash
config log tacacs+accounting setting
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
    set source-ip <IP address>
end
```

Preferred Source

功能简介

当 FortiGate 产生本地发起（local-out）的流量时，默认会采用出接口的主 IP 作为源地址。如：

- FortiGuard 查询
- FortiGate Cloud 隧道
- DNS 查询
- BGP 对等会话
- 远程 RADIUS/LDAP 流量
- 设备自身发起的 ping、traceroute

从 FortiOS v7.4.0 及之后版本，引入了 Preferred Source 功能，可在静态路由、SD-WAN 成员或 BGP route-map 中指定 Local-out 流量应优先使用的源地址，以实现更精细的源 IP 管控。

配置方法

静态路由

在静态路由条目中使用 `preferred-source`，可以让匹配该路由的 Local-out 流量使用指定的源地址。

```bash
config router static
    edit <id>
        set dst 10.10.12.0 255.255.255.0
        set gateway 10.255.2.2
        set preferred-source 192.168.100.1
        set device "port2"
    next
end
```

SD-WAN

在 SD-WAN 成员配置中使用 `preferred-source`，可让 Local-out 流量在经过该 SD-WAN 成员时使用自定义的源地址。

> **重要**
>
> - SD-WAN 健康检查的探测流量需要使用 `set source <ip_address>`，而不是 `preferred-source`。
> - SD-WAN 成员中配置的 `preferred-source` 优先于静态路由中配置的 `preferred-source`。

```bash
config system sdwan
    config members
        edit <id>
            set preferred-source 192.168.100.1
        next
    end
end
```

BGP Route-map

在 route-map 中可通过 `set-ip-prefsrc` 设置源地址，然后将该 route-map 作为 route-map-in 应用到邻居。BGP 邻居建立、BGP 邻居间通信等 Local-out 流量会采用该源地址。

```bash
config router route-map
    edit "<route-map_name>"
        config rule
            edit <id>
                set set-ip-prefsrc <ip_address>
            next
        end
    next
end
```

```
config router bgp
    config neighbor
        edit 169.254.10.1
            set route-map-in <route-map_name>
        next
    end
end
```

注意事项

源地址优先级

如果系统其他位置（例如 `config system dns`）显式配置了 `source-ip`，则该 `source-ip` 会覆盖 `preferred-source`。

> **相关信息**
>
> 参考：策略与对象 → 本地流量策略 → Local-out 流量控制 ↗章节。

如下所示，即使路由设置了 `preferred-source`，DNS 查询仍会使用 `192.168.100.99`。

```
config router static                                                    bash
    edit 1
        set dst 223.5.5.5 255.255.255.255
        set gateway 192.168.100.254
        set preferred-source 192.168.100.1
        set device "port2"
    next
end

config system dns
    set primary 223.5.5.5
    set source-ip 192.168.100.99
end
```

本地地址要求

FortiGate 不强制 Preferred Source 必须是本地接口 IP，但若该地址不属于设备的任一接口，则回复流量可能无法返回，最终导致流量丢失。 如果使用的源地址不是本地接口地址，可通过以下方式确保回程可达：

- 使用 Loopback 接口：

```
config system interface                                                 bash
    edit "loopback1"
        set ip 100.1.1.1 255.255.255.255
        set type loopback
    next
end
```

- 使用 Secondary IP：

```
config system interface                                                 bash
    edit "port2"
        set secondary-IP enable
        config secondaryip
            edit <id>
                set ip 100.1.1.1 255.255.255.255
            next
        end
    next
end
```

- 使用开启 `arp-reply` 的 IP Pool：

```bash
config firewall ippool
    edit "pool1"
        set startip 100.1.1.1
        set endip 100.1.1.1
        set arp-reply enable
    next
end
```

IPsec 相关流量

IPsec 隧道协商与 ESP 流量不会使用静态路由或 SD-WAN 成员中配置的 `preferred-source`，而是使用：

- 已配置的 local-gw。
- 若未配置 local-gw，则使用接口主 IP。

```bash
config vpn ipsec phase1-interface
    edit "example"
        set interface "port2"
        set local-gw 202.103.2.1
        set remote-gw 101.202.1.1
    next
end
```

ISDB

- ISDB 简介
- 预定义 ISDB
- 自定义 ISDB
- 扩展 ISDB
- IP 信誉级别
- 基于地理位置的 ISDB
- ISDB 端口附加配置
- 查询 ISDB/地理位置信息
- 按需模式的 ISDB

ISDB 简介

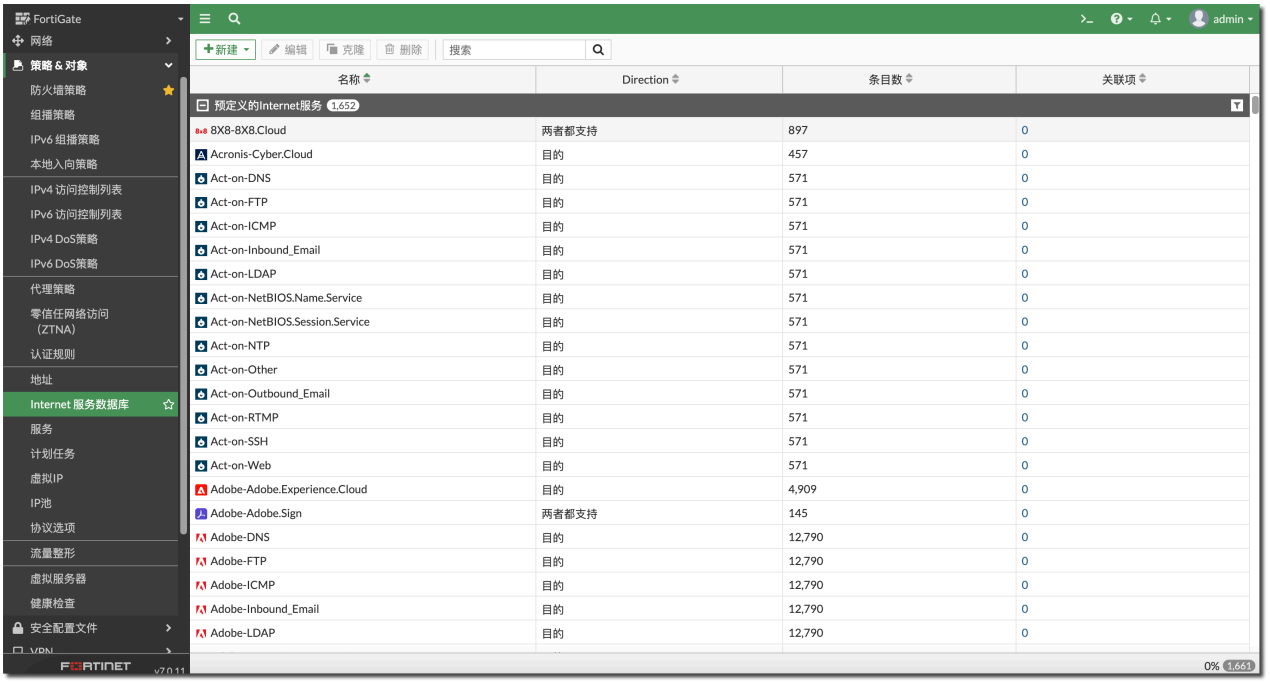
Internet Service Database（ISDB）是综合了 IP 地址范围、IP 所有者、服务端口号、IP 安全可信度的公网 IP/服务数据库。该数据库来自 FortiGuard 系统，会定期更新，例如地理位置、IP 声誉、知名度和 DNS 等。所有这些信息都有助于用户更有效地定义 Internet 安全，您可以使用数据库的内容作为策略中包含或排除的标准。

从 FortiOS 5.6 开始，ISDB 可以在防火墙策略中使用，但只能作为目标对象应用于策略。

从 FortiOS 6.0 开始，ISDB 可以作为策略中的源对象和目标对象使用。您还可以将 Internet 服务应用于限速策略或 SD-WAN 策略。

您可以将三种类型的 Internet 服务应用于策略、静态路由、策略路由、SD-WAN 规则：

- 预定义 ISDB，进入 FortiGate 管理页面 → 策略 & 对象 → Internet 服务数据库，可以看到所有的预定义 ISDB 内容。



- 基于地址位置的 ISDB，可在上述页面手动新建
- 自定义 ISDB，只能通过 CLI 配置
- 扩展 ISDB，只能通过 CLI 配置

预定义 ISDB

本文主要介绍如何将预定义的 ISDB 条目应用到防火墙策略中。

配置步骤

> **重要**
>
> ISDB 是一系列 IP、协议、端口号的集合，所以配置后进行测试时，请使用 ISDB 中包含的协议、端口号进行测试，并不是只匹配到目标 IP 就会匹配到相关的 ISDB。

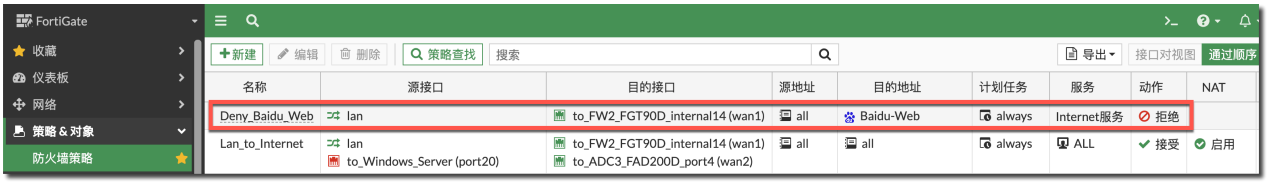
1. 通过 Web 方式访问防火墙的管理页面，进入“策略 & 对象 → 防火墙策略”页面，新建防火墙策略。
2. 配置流入接口和源地址，在“目标地址”选项中选择“Internet 服务”，搜索“baidu”，选择“Baidu-Web”到目标地址中（该 ISDB 包含了所有百度的 Web 页面资源），动作选择“拒绝”，开启“记录拒绝流量”。





```
config firewall policy
    edit 8
        set name "Deny_Baidu_Web"
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Baidu-Web"
        set schedule "always"
        set logtraffic all
    next
end
```

3. 将该条策略移动至上网策略之前。

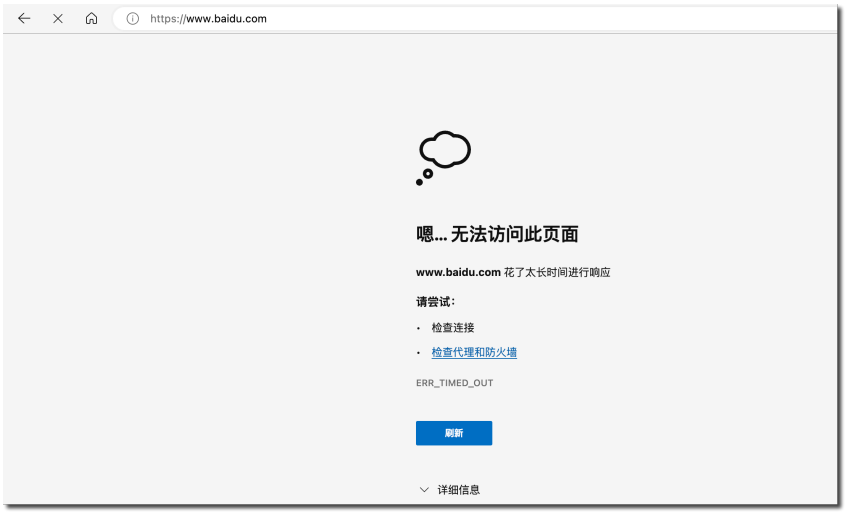


4. 使用“diagnose internet-service id 65646”可以看到该 ISDB 的详细信息，“diagnose internet-service id-summary 65646”可以看到该 ISDB 的简要信息。

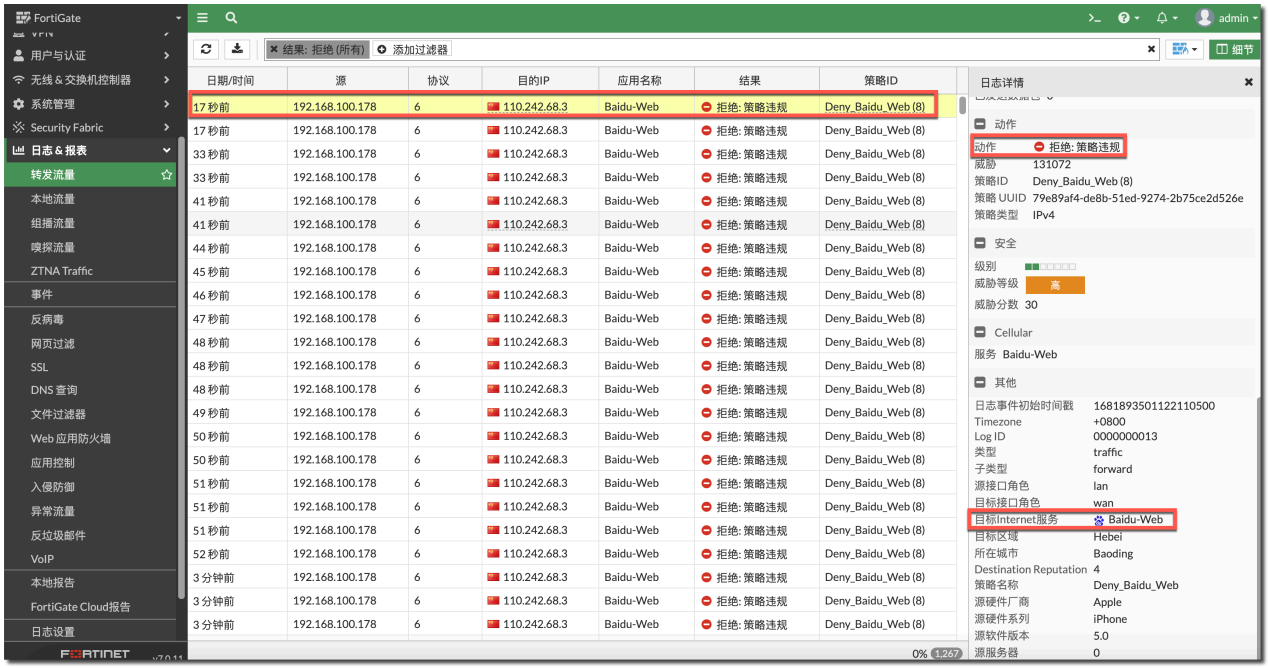
```
diagnose internet-service id 65646
diagnose internet-service id-summary 65646
```

结果验证

1. 使用内网 PC 经过 FortiGate 访问网页 https://www.baidu.com ↗ ，无法打开页面。



2. 查看防火墙的流量日志，可以看到客户端访问 https://www.baidu.com ↗ 的流量被阻断，目标 ISDB 识别正确。



3. 由于策略中引用的 ISDB Baidu-Web 不包含 ICMP 服务，所以客户端仍然可以 Ping 通 www.baidu.com ↗ 。

```
~ ifconfig en7
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6467<RXCSUM,TXCSUM,VLAN_MTU,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 0c:37:96:06:d7:5d
        inet6 fe80::c9:55b5:f9f6:7b31%en7 prefixlen 64 secured scopeid 0xd
        inet 192.168.100.178 netmask 0xffffff00 broadcast 192.168.100.255
        inet 192.168.77.253 netmask 0xffffff00 broadcast 192.168.77.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (1000baseT <full-duplex>)
        status: active
~ ping www.baidu.com -c 5
PING www.a.shifen.com (110.242.68.4): 56 data bytes
64 bytes from 110.242.68.4: icmp_seq=0 ttl=51 time=10.831 ms
64 bytes from 110.242.68.4: icmp_seq=1 ttl=51 time=14.927 ms
64 bytes from 110.242.68.4: icmp_seq=2 ttl=51 time=10.963 ms
64 bytes from 110.242.68.4: icmp_seq=3 ttl=51 time=11.974 ms
64 bytes from 110.242.68.4: icmp_seq=4 ttl=51 time=10.716 ms

--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 10.716/11.882/14.927/1.587 ms
```

自定义 ISDB

功能介绍

当预定义的 ISDB 中无法找到自己需要的 ISDB 时，可以使用自定义的 ISDB（只能在 CLI 下配置），并应用在防火墙策略、限速策略、SD-WAN 策略的目标中。

创建自定义的 ISDB 时，必须包含以下参数：

- IP 地址或 IP 地址范围
- 协议号
- 端口号或端口号范围（如果是没有端口号的协议，则不需要填写，如 ESP）
- 声誉级别（默认为 3）

配置示例

```
config firewall internet-service-custom
    edit <name>
       set comment <comment>
       set reputation {1 | 2 | 3 | 4 | 5}
       config entry
           edit <ID>
               set protocol <protocol #>
               set dst <object_name>
               config port-range
                   edit <ID>
                       set start-port <port #>
                       set end-port <port #>
                   next
               end
           next
       end
    end
end
```

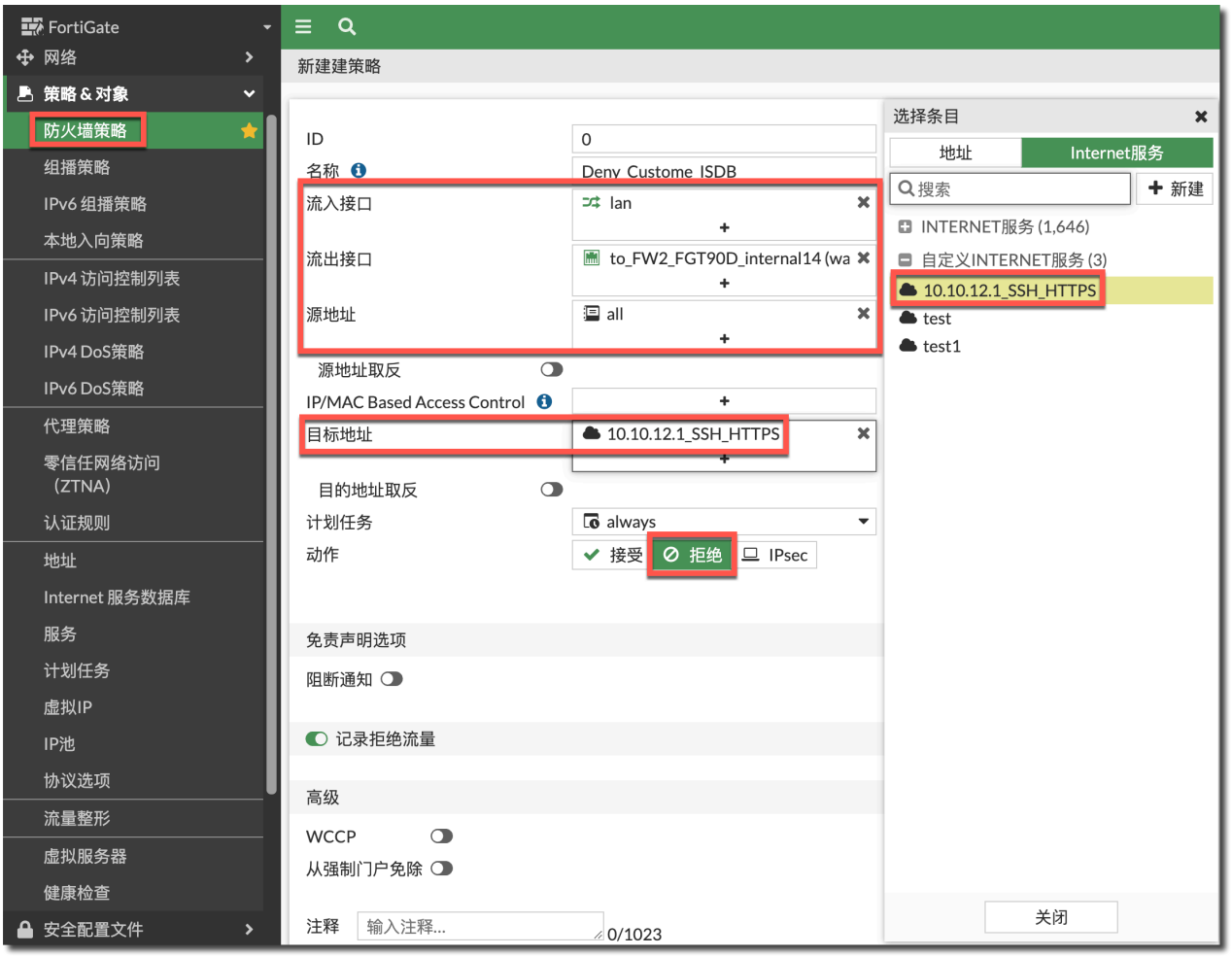
配置步骤

1. 配置目标的地址对象，并配置一条自定义的 ISDB 引用它们，协议为 TCP 的 22 端口和 TCP 的 443 端口。

```bash
config firewall address
    edit "10.10.12.1/32"
        set subnet 10.10.12.1 255.255.255.255
    next
end

config firewall internet-service-custom
    edit "10.10.12.1_SSH_HTTPS"
        set comment ''
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 22
                        set end-port 22
                    next
                    edit 2
                        set start-port 443
                        set end-port 443
                    next
                end
```

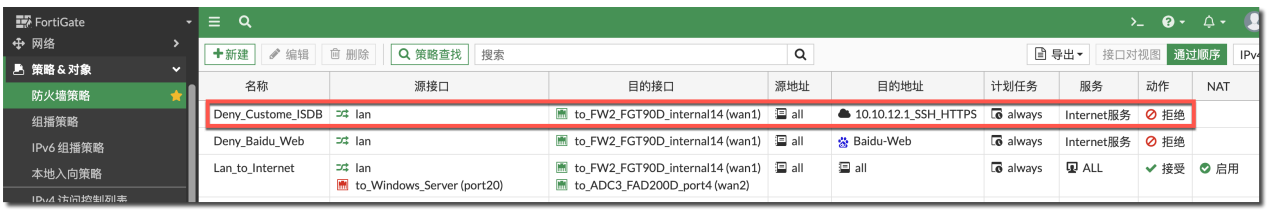
```
                set dst "10.10.12.1/32"
            next
        end
    next
end
```

2. 配置流入接口和源地址，在“目标地址”选项中选择“Internet 服务”，在“自定义 INTERNET 服务”分类中，选择上步创建的自定义 ISDB 到目标地址中，动作选择“拒绝”，开启“记录拒绝流量”。




```
config firewall policy
    edit 9
        set name "Deny_Custome_ISDB"
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-custom "10.10.12.1_SSH_HTTPS"
        set schedule "always"
        set logtraffic all
    next
end
```

3. 将该条策略移动至上网策略之前。

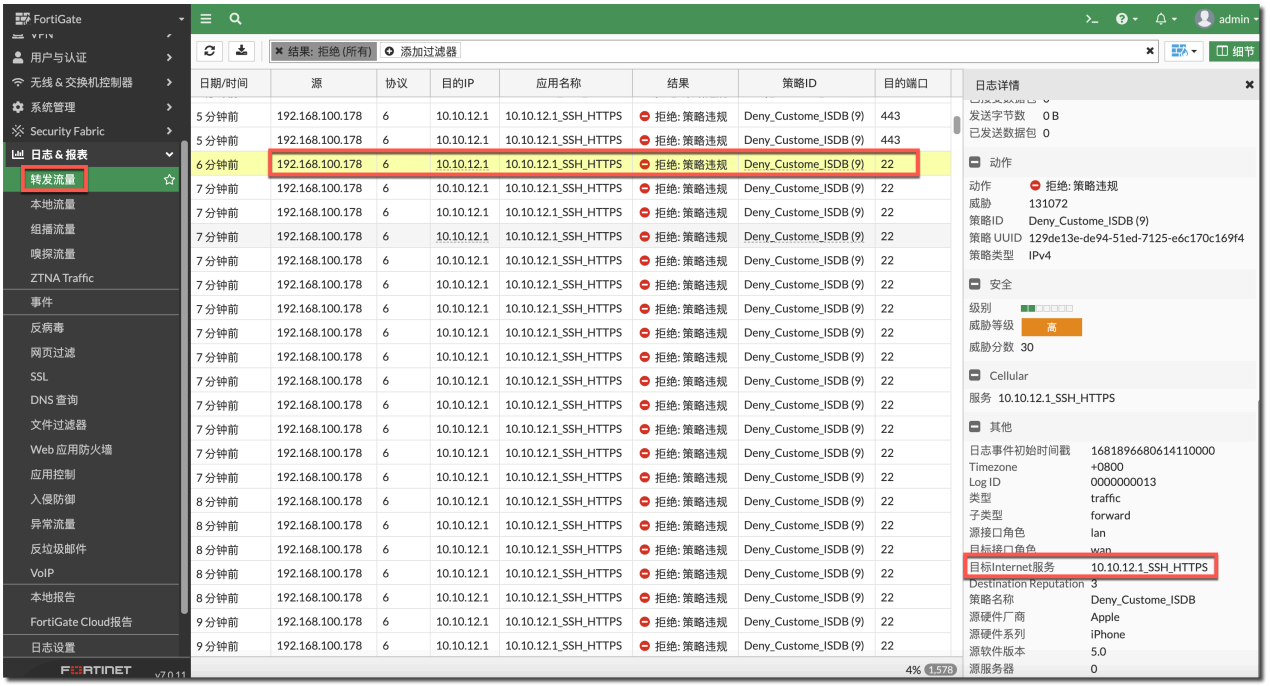


结果验证

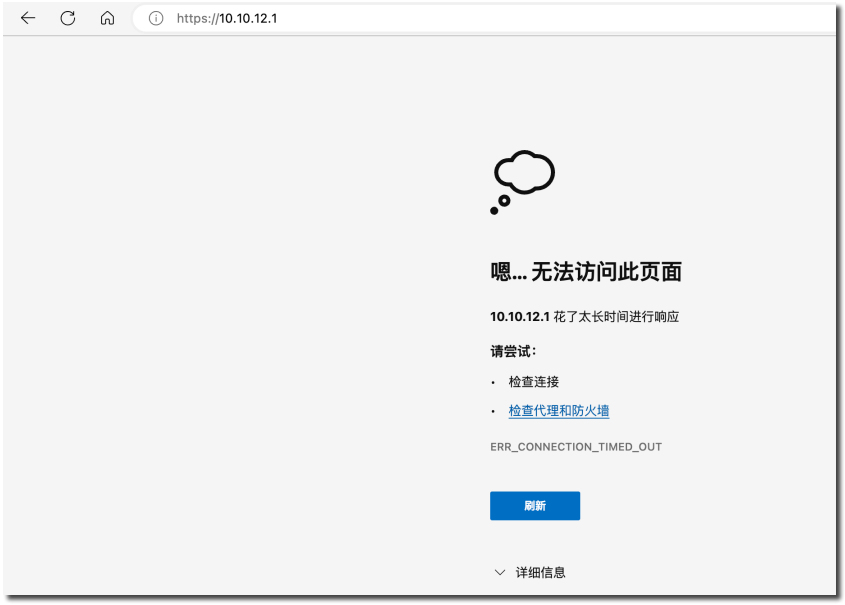
1. 使用内网 PC 经过 FortiGate 访问 10.10.12.1 的 SSH（TCP port22）服务，无法访问。

```
~ ifconfig en7
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6467<RXCSUM,TXCSUM,VLAN_MTU,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 0c:37:96:06:d7:5d
        inet6 fe80::c9:55b5:f9f6:7b31%en7 prefixlen 64 secured scopeid 0xd
        inet 192.168.100.178 netmask 0xffffff00 broadcast 192.168.100.255
        inet 192.168.77.253 netmask 0xffffff00 broadcast 192.168.77.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (1000baseT <full-duplex>)
        status: active
~ ssh admin@10.10.12.1
ssh: connect to host 10.10.12.1 port 22: Operation timed out
```

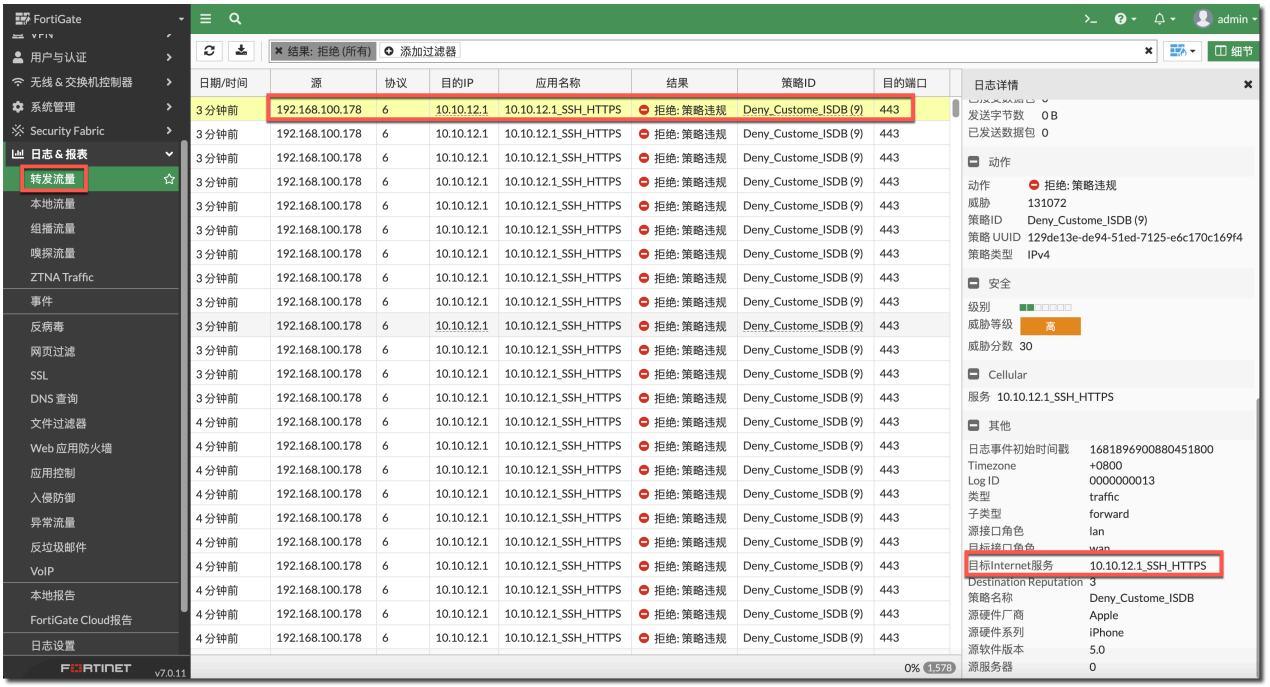
2. 查看防火墙的流量日志，可以看到客户端访问 10.10.12.1 的 SSH 流量被阻断，目标 ISDB 识别正确。



3. 使用内网 PC 经过 FortiGate 访问 10.10.12.1 的 HTTPS（TCP port 443）服务，被阻断。



4. 查看防火墙的流量日志，可以看到客户端访问 10.10.12.1 的 HTTPS 流量被阻断，目标 ISDB 识别正确。


FortiGate
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
转发流量
本地流量
组播流量
嗅探流量
ZTNA Traffic
事件
反病毒
网页过滤
SSL
DNS 查询
文件过滤器
Web 应用防火墙
应用控制
入侵防御
异常流量
反垃圾邮件
VoIP
本地报告
FortiGate Cloud报告
日志设置
FORTINET
v7.0.11
admin
结果: 拒绝 (所有)
添加过滤器
细节
日期/时间 源 协议 目的IP 应用名称 结果 策略ID 目的端口
3 分钟前 192.168.100.178 6 10.10.12.1 10.10.12.1_SSH_HTTPS 拒绝: 策略违规 Deny_Custome_ISDB (9) 443
4 分钟前 192.168.100.178 6 10.10.12.1 10.10.12.1_SSH_HTTPS 拒绝: 策略违规 Deny_Custome_ISDB (9) 443
0% 1,578
日志详情
发送字节数 0 B
已发送数据包 0
动作
动作 拒绝: 策略违规
威胁 131072
策略ID Deny_Custome_ISDB (9)
策略 UUID 129de13e-de94-51ed-7125-e6c170c169f4
策略类型 IPv4
安全
级别
威胁等级 高
威胁分数 30
Cellular
服务 10.10.12.1_SSH_HTTPS
其他
日志事件初始时间戳 1681896900880451800
Timezone +0800
Log ID 0000000013
类型 traffic
子类型 forward
源接口角色 lan
目标接口角色 wan
目标Internet服务 10.10.12.1_SSH_HTTPS
Destination Reputation 3
策略名称 Deny_Custome_ISDB
源硬件厂商 Apple
源硬件系列 iPhone
源软件版本 5.0
源服务器 0

扩展 ISDB

功能介绍

扩展 ISDB 允许您对现有的预定义 ISDB 中配置的 IP 和端口范围进行增加，实际上是编辑预定义类型的 ISDB 条目，并向其添加 IP 地址和端口范围。

扩展 ISDB 必须在 CLI 下配置，创建扩展的 ISDB 时，必须包含以下参数：

- IP 地址或 IP 地址范围
- 协议号
- 端口号或端口号范围（如果是没有端口号的协议，则不需要填写，如 ESP）

删除时必须从 GUI 的预定义 ISDB 中删除相关条目。

配置示例

```
config firewall internet-service-extension
    edit <ID #>
        set comment <comment>
        config entry
            edit <ID #>
                set protocol <number #>
                set dst <object_name>
                config port-range
                    edit <ID #>
                        set start-port <number #>
                        set end-port <number #>
                    next
                end
            next
        end
    end
end
```

网络需求

1. 用户想禁止内网访问阿里云相关资源的流量，配置了安全策略阻断去往预定义 ISDB“Alibaba-Alibaba.Cloud”的流量。
2. 但由于阿里云域名对应的 IP 一直在变化，预定义的 ISDB 还未来得及更新最新的 IP 地址和端口号，导致阿里云的个别资源仍然可以正常访问（如 www.aliyun.com ↗ ）。



3. 通过域名解析，找到仍然可以访问的资源为 42.59.1.232、42.59.1.234、42.59.1.231、42.59.1.233、42.59.1.229、42.59.1.236、42.59.1.230、42.59.1.235 的 443 端口。

```
~ nslookup www.aliyun.com
Server:         172.22.40.20
Address:        172.22.40.20#53

Non-authoritative answer:
www.aliyun.com  canonical name = www-jp-de-intl-adns.aliyun.com.
www-jp-de-intl-adns.aliyun.com  canonical name = www-jp-de-intl-adns.aliyun.com.
gds.alibabadns.com.
www-jp-de-intl-adns.aliyun.com.gds.alibabadns.com       canonical name = www.ali
yun.com.w.cdngslb.com.
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.232
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.234
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.231
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.233
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.229
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.236
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.230
Name:   www.aliyun.com.w.cdngslb.com
Address: 42.59.1.235
```

4. 查看解析到的 IP 所属的预定义 ISDB，可以看到没有属于任何一个预定义的 ISDB。

```
FortiGate # diagnose internet-service info root 6 443 42.59.1.235
Can not find Internet Service ID and name. ret=-1
```

5. 解决方法是通过在预定义的 ISDB 上使用扩展 ISDB，以扩充预定义的 ISDB。

配置步骤

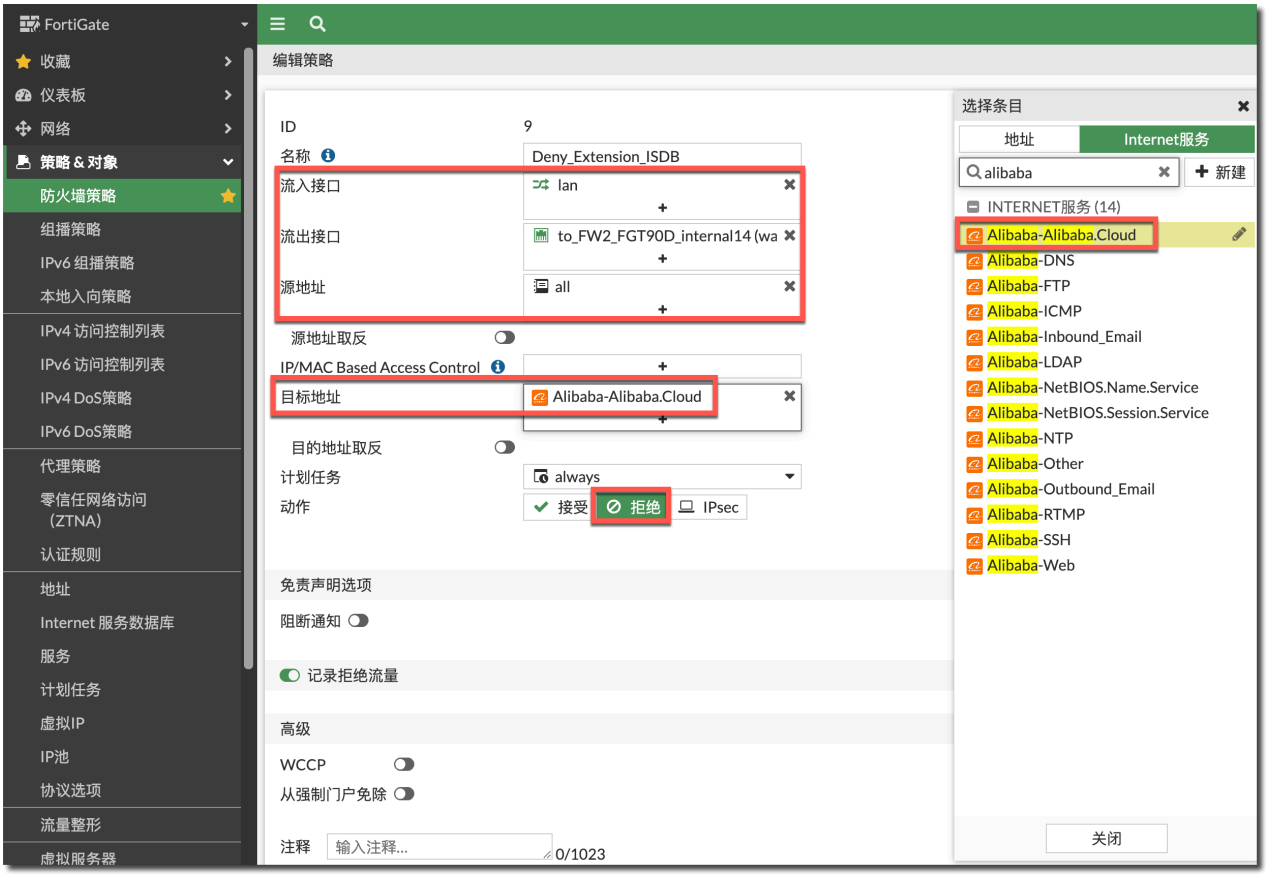
1. 配置需要在预定义的 ISDB 上扩展的 IP 资源的地址和地址组对象。

```
config firewall address
    edit "AliCloud_1"
        set subnet 42.59.1.232 255.255.255.255
    next
    edit "AliCloud_2"
        set subnet 42.59.1.234 255.255.255.255
    next
    edit "AliCloud_3"
        set subnet 42.59.1.231 255.255.255.255
    next
    edit "AliCloud_4"
        set subnet 42.59.1.233 255.255.255.255
    next
    edit "AliCloud_5"
        set subnet 42.59.1.229 255.255.255.255
    next
    edit "AliCloud_6"
        set subnet 42.59.1.236 255.255.255.255
    next
    edit "AliCloud_7"
        set subnet 42.59.1.230 255.255.255.255
    next
    edit "AliCloud_8"
        set subnet 42.59.1.235 255.255.255.255
    next
end
config firewall addrgrp
    edit "AliCloud_extension"
        set member "AliCloud_1" "AliCloud_2" "AliCloud_3" "AliCloud_4" "AliCloud_5" "AliCloud_6" "AliCloud_7"
"AliCloud_8"
    next
end
```

2. 配置一条扩展的 ISDB，edit 的 ID 填写预定义的 ISDB“Alibaba-Alibaba.Cloud”的 ID 号（该 ID 号可以在 GUI 中查询），引用上步配置的地址组，协议为 TCP，端口号为 443 端口。

```
config firewall internet-service-extension
    edit 6881402
        set comment ''
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 443
                        set end-port 443
                    next
                end
                set dst "AliCloud_extension"
            next
        end
    next
end
```

3. 防火墙策略中配置流入接口和源地址，在“目标地址”选项中选择“Internet 服务”，在“INTERNET 服务”分类中，选择预定义的 ISDB“Alibaba-Alibaba.Cloud”，动作选择“拒绝”，开启“记录拒绝流量”。




```
config firewall policy
    edit 9
        set name "Deny_Extension_ISDB"
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Alibaba-Alibaba.Cloud"
        set schedule "always"
        set logtraffic all
    next
end
```

4. 将该条策略移动至上网策略之前。

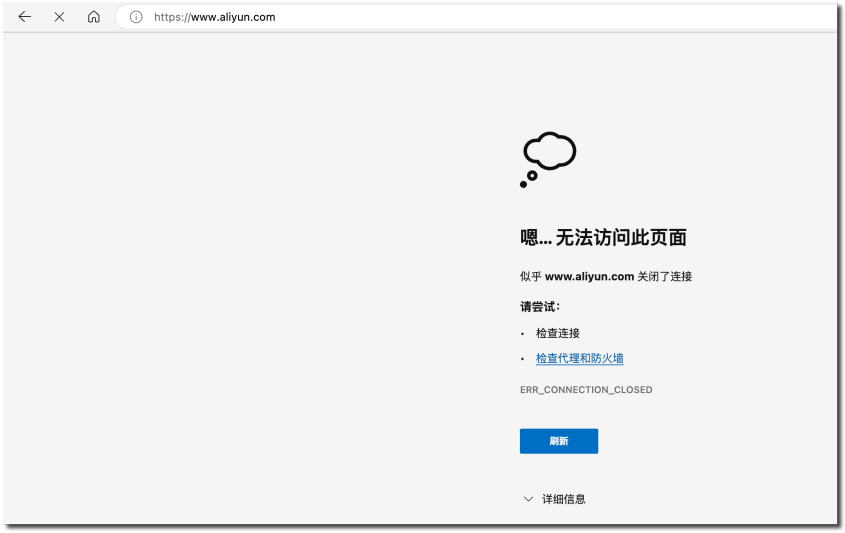


结果验证

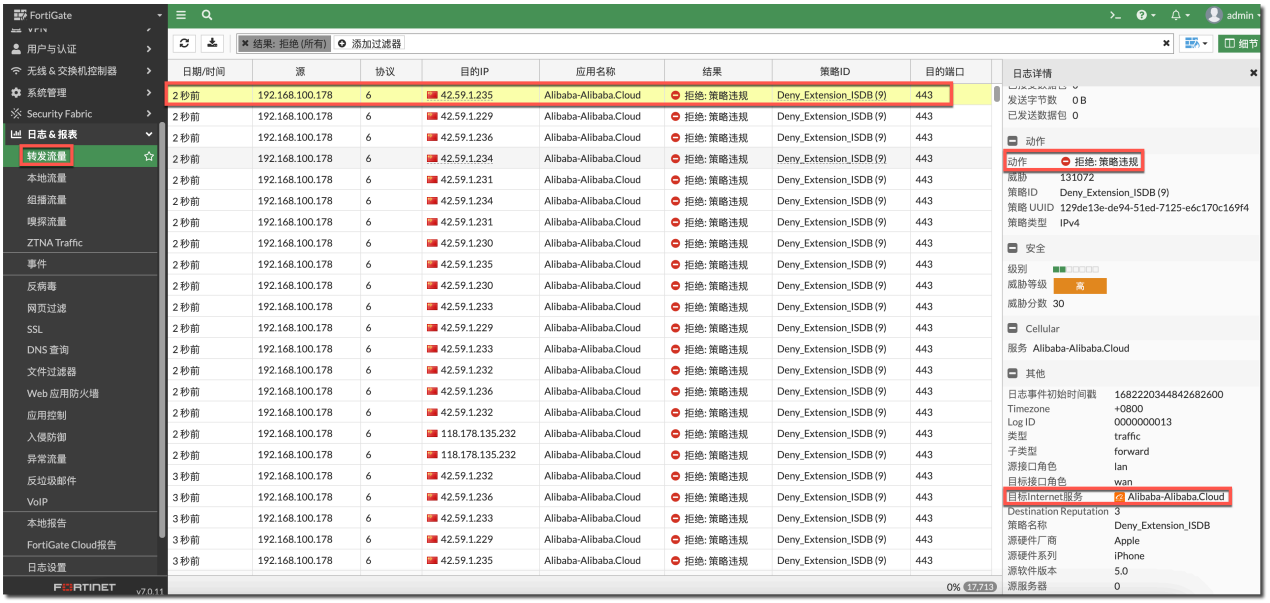
1. 查询手动扩展的 IP 和端口所属的 ISDB，可以看到已被归类到预定义的 ISDB“Alibaba-Alibaba.Cloud”中。

```
FortiGate # diagnose internet-service info root 6 443 42.59.1.235
Internet Service: 6881402(Alibaba-Alibaba.Cloud) country(0 Any) region(0 Any) city(0 Any)
```

2. 使用内网 PC 经过 FortiGate 访问 www.aliyun.com，被阻断。



3. 查看防火墙的流量日志，可以看到客户端访问 10.10.12.1 的 HTTPS 流量被阻断，目标 ISDB 识别正确。

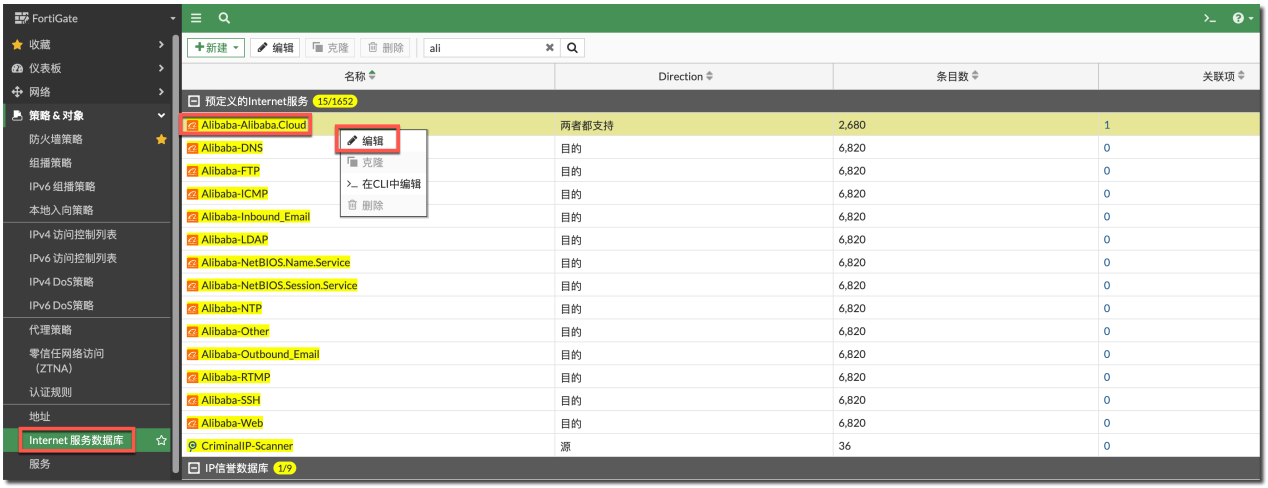


禁用预定义 ISDB 中的 IP/端口

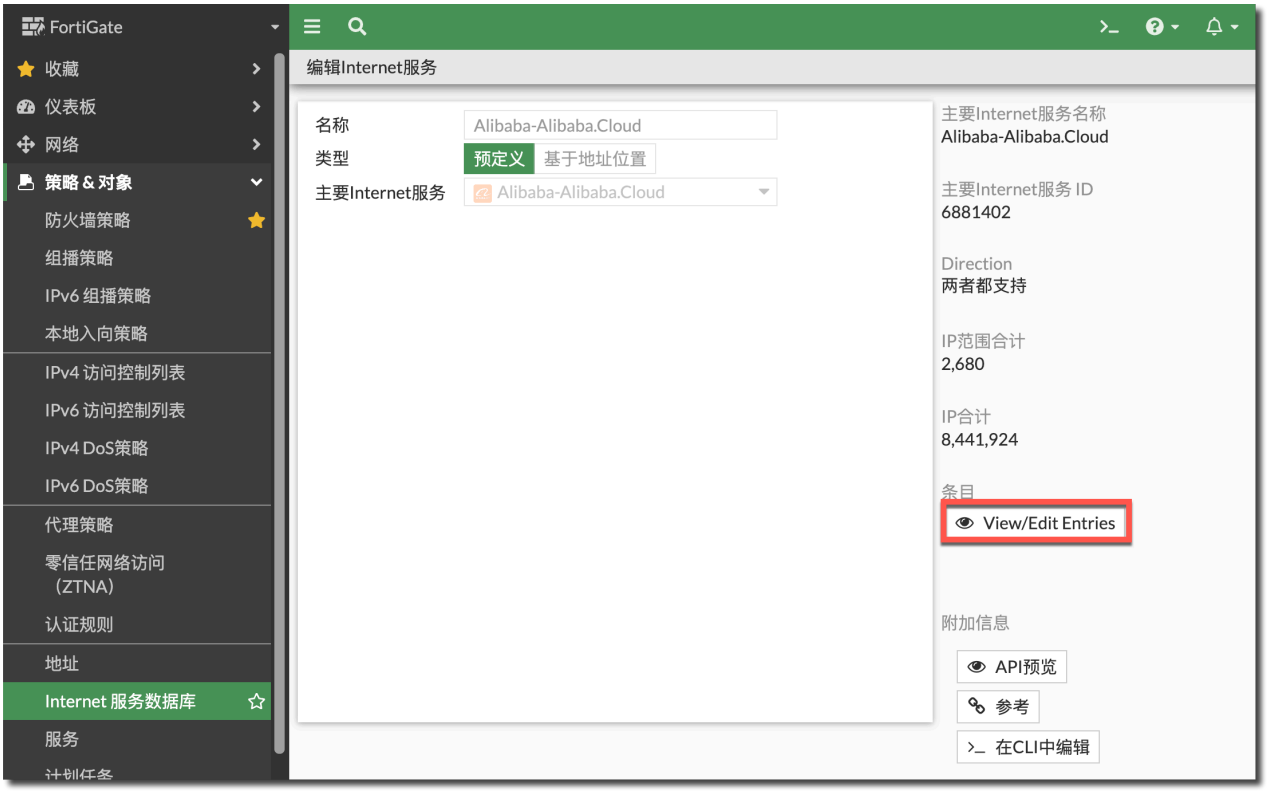
1. 首先查询某个属于预定义 ISDB“Alibaba-Alibaba.Cloud”的 IP/端口。

```
FortiGate # diagnose internet-service info root 6 443 223.109.11.27
Internet Service: 6881402(Alibaba-Alibaba.Cloud) country(156 China) region(801 Jiangsu) city(26800 Wuxi)
```

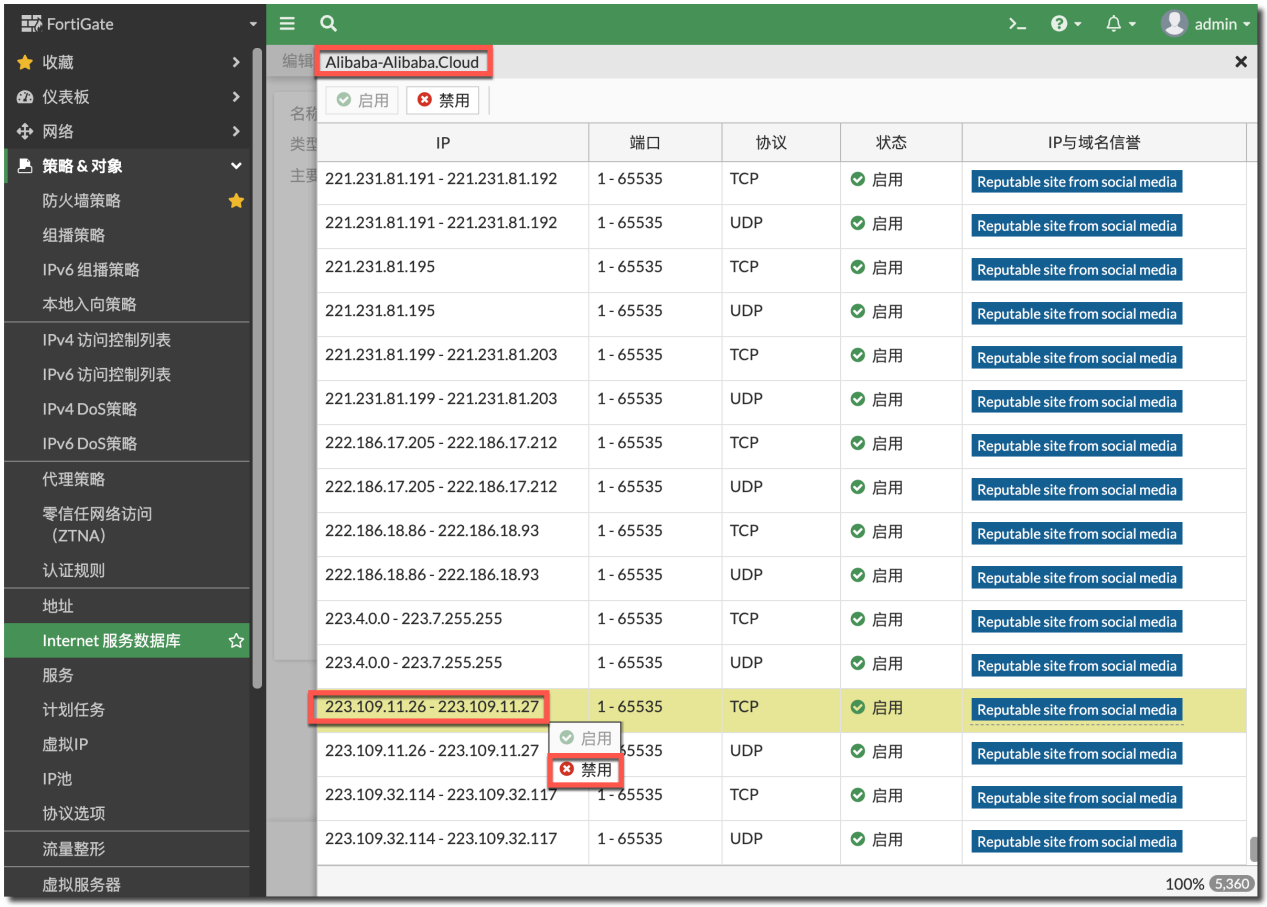
2. 进入“策略 & 对象 → Internet 服务数据库”，编辑要删除内容的预定义 ISDB，这里以“Alibaba-Alibaba.Cloud”为例，双击或邮件选择“编辑”。

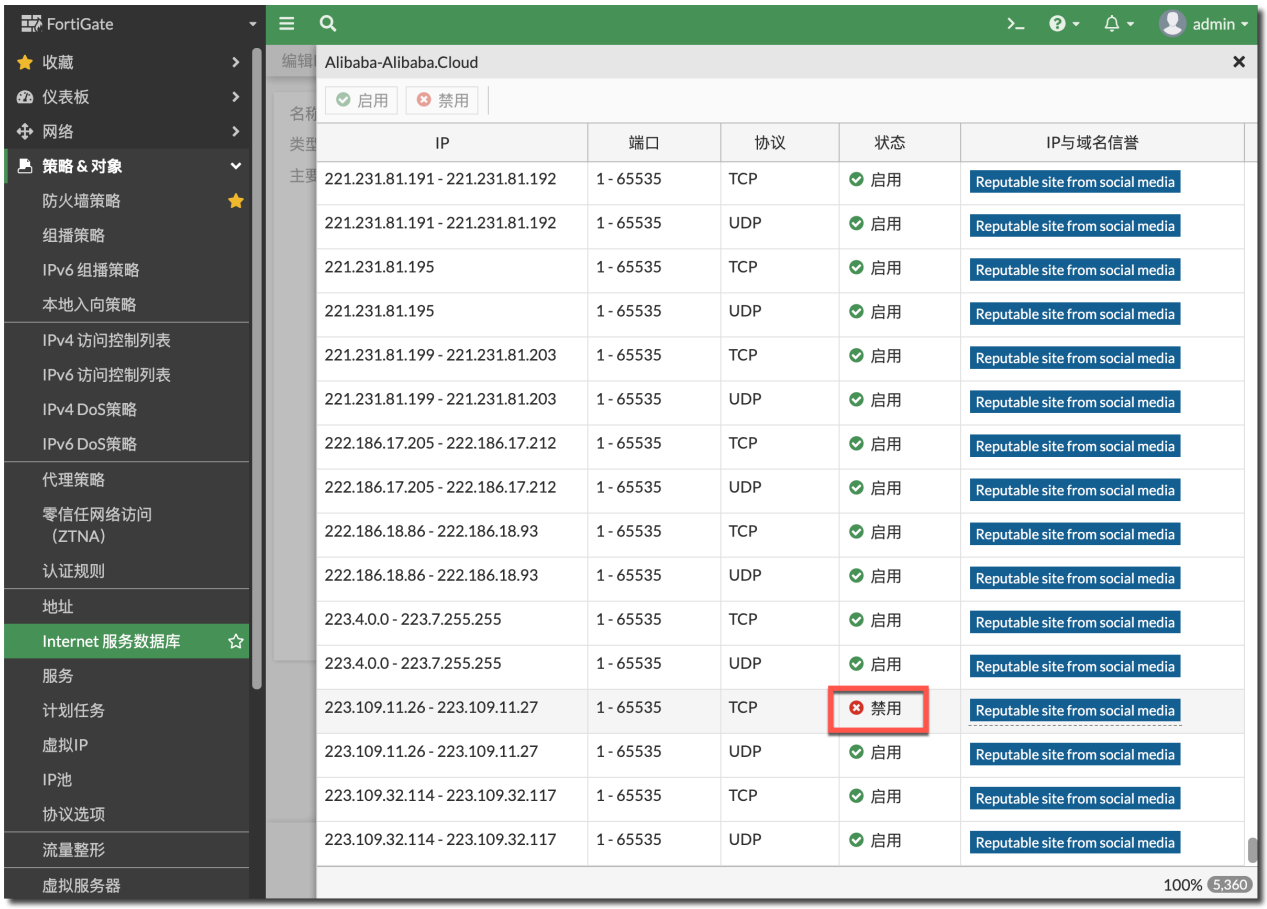


3. 点击页面右侧的“View/Edit Entries”，



4. 右键禁用 IP/端口所属的条目。





5. 再次查询该 IP 地址/端口所属的 ISDB，可以看到没有属于任何 ISDB。

```
FortiGate # diagnose internet-service info root 6 443 223.109.11.27
Can not find Internet Service ID and name. ret=-1
```

IP 信誉级别

功能介绍

ISDB 中目前有五个信誉级别，可以在自定义 ISDB 中定义信誉级别。您可以配置防火墙策略以根据所需的信誉级别过滤流量。如果 IP 地址的信誉级别 ≥ 策略中设置的级别，则会转发数据包，否则将丢弃数据包。

五个信誉级别如下所示：

信誉级别	
1	已知的恶意站点，例如网络钓鱼站点或与僵尸网络服务器相关的站点
2	高风险服务站点，例如 TOR、代理和 P2P
3	未经证实的网站
4	信誉良好的社交媒体网站，如 Facebook 和 Twitter
5	已知和经过验证的安全网站，例如 Gmail、Amazon、eBay 和淘宝

重要

防火墙策略中，默认最低信誉级别为 0，表示未启用信誉过滤器。

对于未包含在 ISDB 中的 IP 地址，默认信誉级别为 3。

默认信誉方向是基于目标 IP 的（可在策略中修改为基于源地址）。

配置举例

举例一

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "port1"
        set srcaddr “all”
        set dstaddr "all"
        set reputation-minimum 3
        set reputation-direction source
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

重要

来自信誉级别为 3、4、5 的源 IP 地址的数据包将匹配此策略。

举例二

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "wan2"
        set srcaddr “all”
        set dstaddr "all"
        set reputation-minimum 4
```

```
        set reputation-direction destination
        set action accept
        set schedule "always"
        set service "FTP"
        set nat enable
    next
end
```

重要

如果流量目标的 IP 信誉级别为 4 或 5，且服务为 FTP，则会匹配此策略。

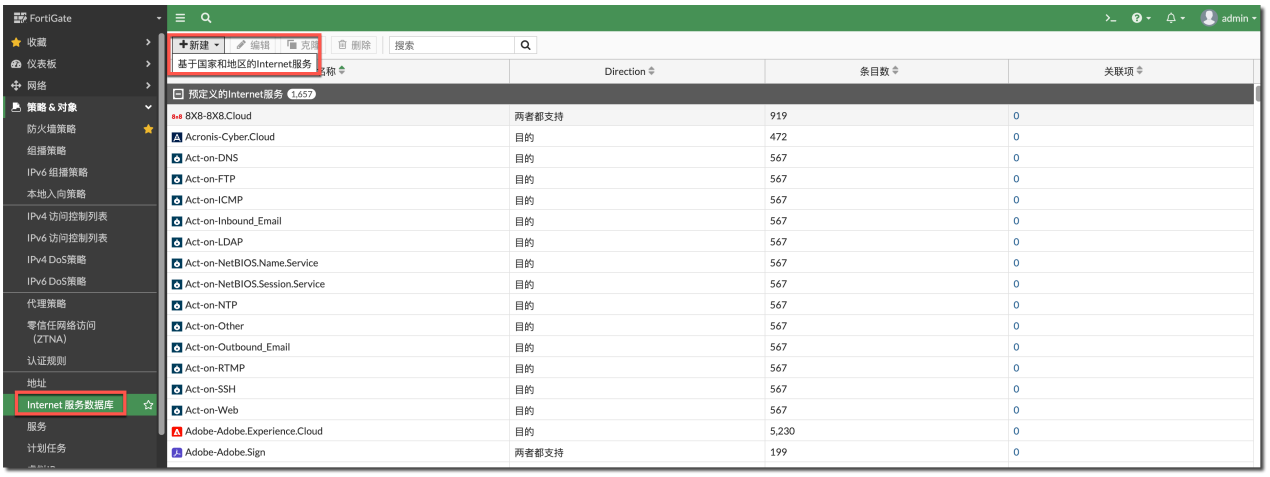
基于地理位置的 ISDB

功能介绍

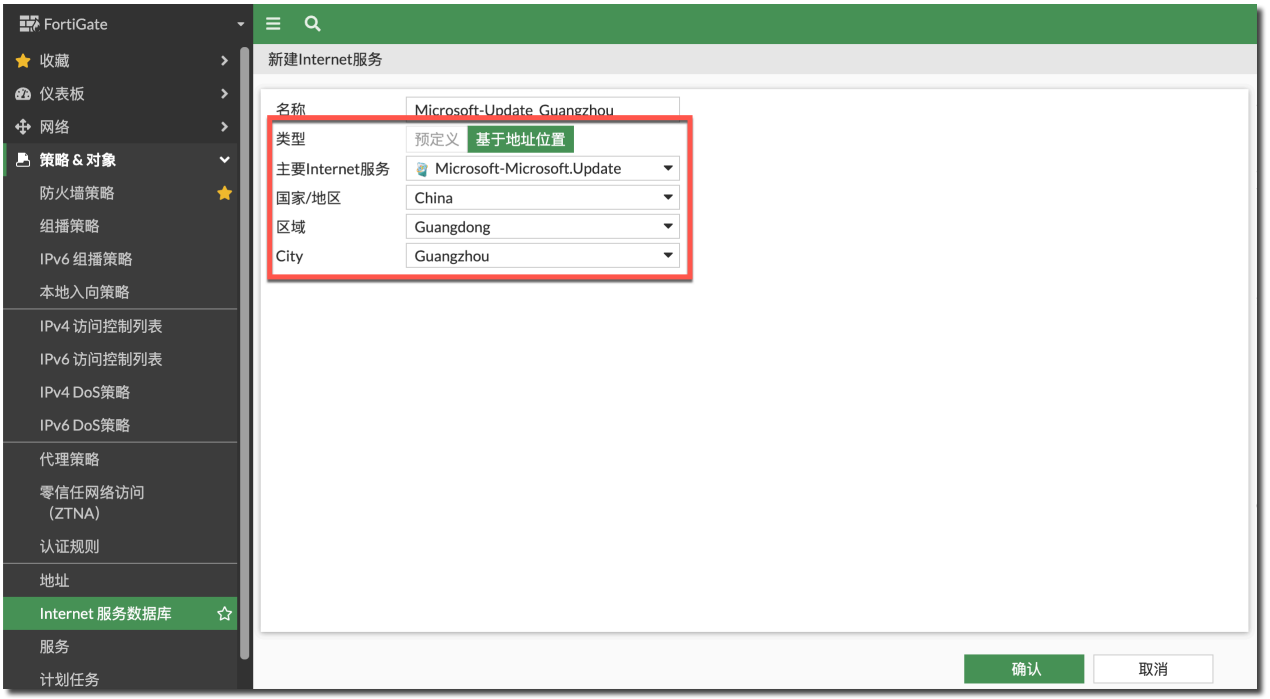
基于地理位置的 ISDB 允许用户定义国家/地区和城市。这些对象可以在防火墙策略、限速策略、SD-WAN 规则中使用，以便对 ISDB 的位置进行更精细的控制。

配置步骤

1. 进入“策略 & 对象 → Internet 服务数据库”，点击“新建”按钮，选择“基于国家和地区的 Internet”服务。



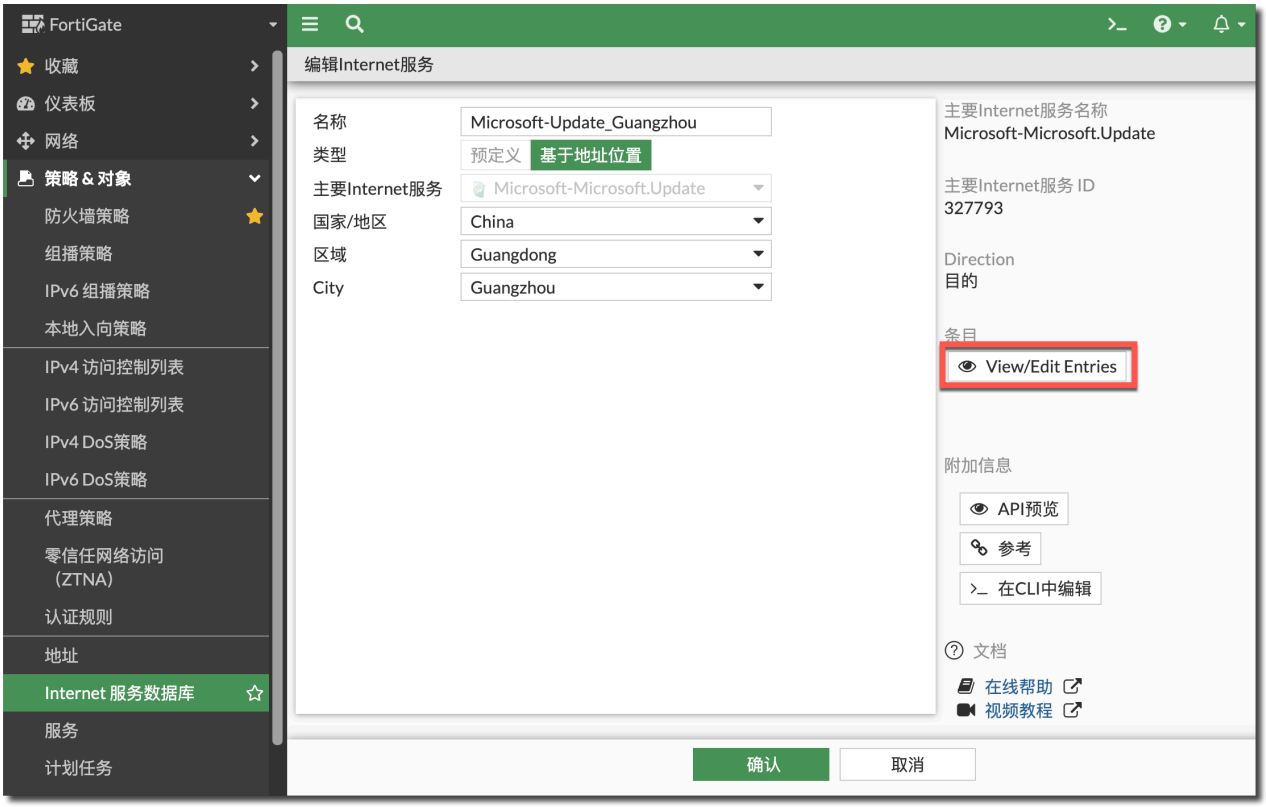
2. 按需配置主要的 Internet 服务，以及限定的国家/地区、区域和城市，下发配置。如下图所示为微软的更新服务 ISDB，限定 IP 的地理位置在中国广东省广州市。

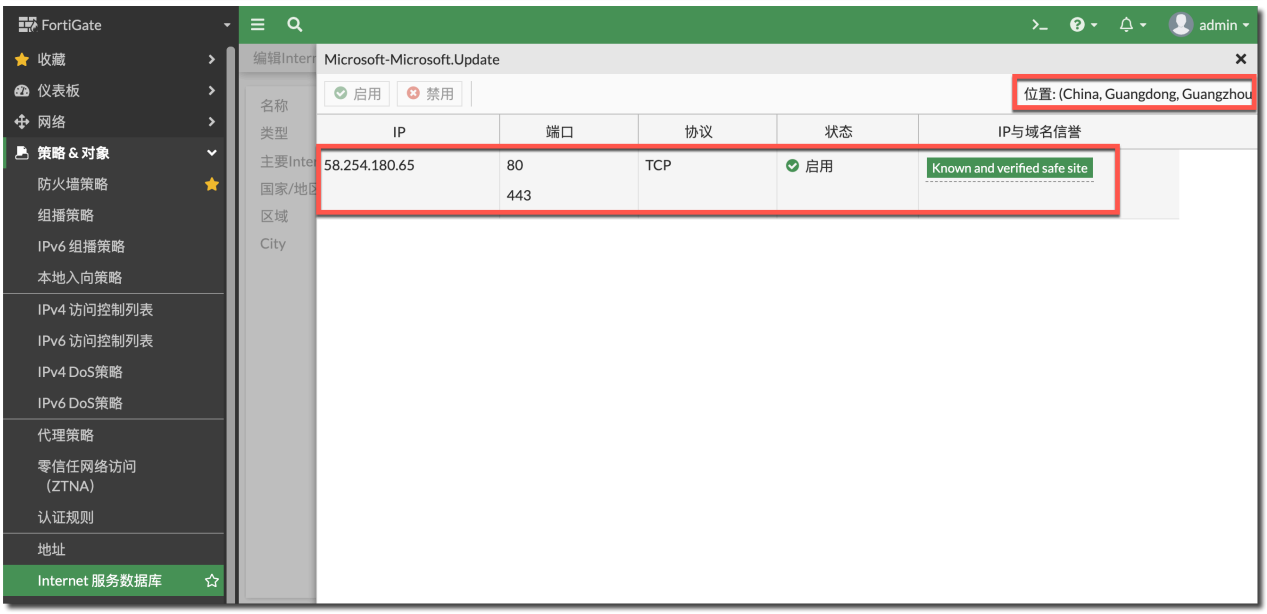



```
config firewall internet-service-name
    edit "Microsoft-Update_Guangzhou"
        set type location
        set internet-service-id 327793
        set country-id 156
```

```
        set region-id 628
        set city-id 9028
    next
end
```

3. 编辑上步配置的基于地理位置的 ISDB，点击右侧的“View/Edit Entries”，可以查看该 ISDB 包含的 IP/端口库。

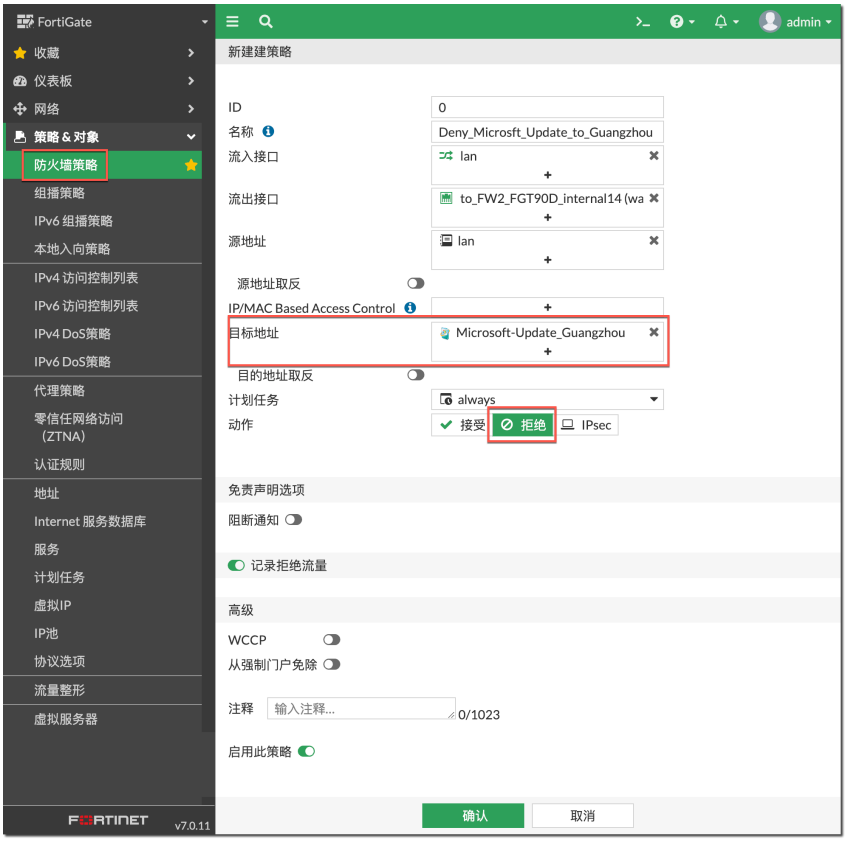




4. CLI 下使用调试命令查看该 ISDB 属于广州的 IP 条目。

```
FortiGate # diagnose internet-service id 327793 | grep "country(156) region(628) city(9028)"
58.254.180.65-58.254.180.65 country(156) region(628) city(9028) blocklist(0×0) reputation(5), popularity(5) domain(0)
botnet(0) proto(6) port(80 443)
```

5. 新建防火墙安全策略，在目的中引用该 ISDB，禁止访问该 ISDB 所属资源的流量。




```
config firewall policy
    edit 6
        set name "Deny_Microsft_Update_to_Guangzhou"
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "lan"
        set internet-service enable
        set internet-service-name "Microsoft-Update_Guangzhou"
        set schedule "always"
        set logtraffic all
    next
end
```

ISDB 端口附加配置

功能介绍

1. 通过在 CLI 下配置“internet-service-append”，我们可以将包含指定端口的预定义 ISDB 服务，自动附加其他指定端口。
2. 举例：客户需要将所有包含 443 端口的 ISDB 中自动附加端口 10443。

配置步骤

1. 配置前，查询某个 IP + 端口的所属 ISDB，可以看到 110.242.68.4 + Port 443 属于“Baidu-Web”，而 110.242.68.4 + Port 10443 属于“Baidu-Other”。

```
FortiGate # diagnose internet-service info root 6 443 110.242.68.4
Internet Service: 2883585(Baidu-Web) country(156 China) region(688 Hebei) city(1692 Baoding)

FortiGate # diagnose internet-service info root 6 10443 110.242.68.4
Internet Service: 2883584(Baidu-Other) country(156 China) region(688 Hebei) city(1692 Baoding)
```

2. 在 CLI 下配置“internet-service-append”，一下配置表示只要某个预定义的 ISDB 中包含的端口匹配“match-port”配置的端口 443，则会将“append-port”配置的端口 10443 附加在这个 ISDB 中，该配置在全局的预定义 ISDB 中生效。

> **重要**
>
> 不包含具体端口的 ISDB 不会匹配“match-port”配置的端口。

```
config firewall internet-service-append
    set match-port 443
    set append-port 10443
end
```

3. 需要使用如下命令重载 ISDB。

```
execute internet-service refresh
```

4. 再次查询 110.242.68.4 + Port 10443 端口所属的 ISDB，已经变为“Baidu-Web”，但不再属于“Baidu-Other”。

```
FortiGate # diagnose internet-service info root 6 10443 110.242.68.4
Internet Service: 2883585(Baidu-Web) country(156 China) region(688 Hebei) city(1692 Baoding)
```

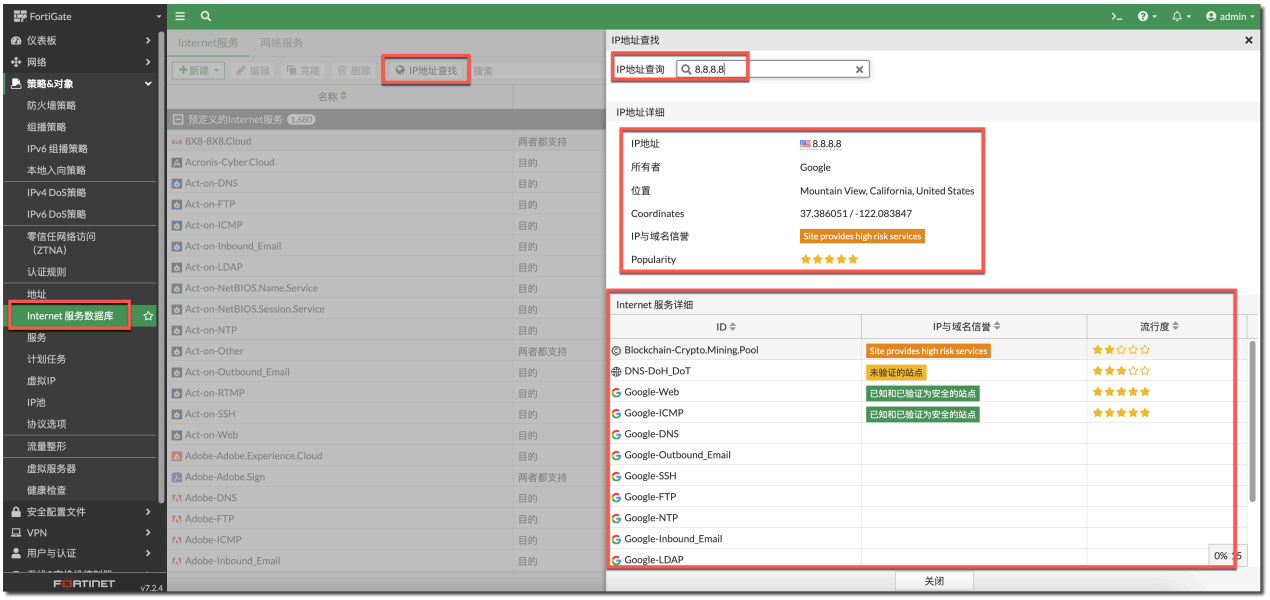
查询 ISDB/地理位置信息

用户可以从 ISDB 库/GEO 库中查找 IP 地址所属 ISDB 以及地理位置信息。返回的 IP 地址信息包括所属 ISDB、位置、信誉和其他服务信息。

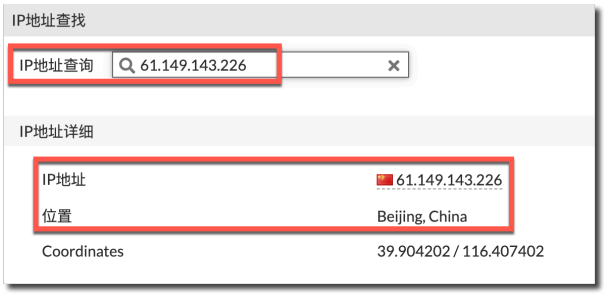
通过 GUI 查询

> **重要**
>
> GUI 方式查询需 FortiOS 7.2.0 及更高版本支持。

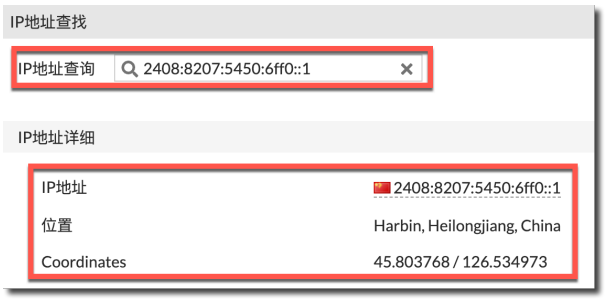
1. 进入“策略 & 对象 → Internet 服务数据库”，点击“IP 地址查找”按钮，填写 IPv4 地址，可以查找到该 IP 所属的 ISDB 信息以及地理位置信息。



2. 查找一个不属于 ISDB 库的 IP 地址，也可以显示地理位置信息。



3. 同时也支持 IPv6 的查询。



通过 CLI 查询

1. 查询具体流量所属的 ISDB，可以查询更精细的协议和端口。

```
diagnose internet-service info <vdname> <protocol> <port> <ip>
```

```
FortiGate # diagnose internet-service info root 6 443 110.242.68.4
Internet Service: 2883585(Baidu-Web) country(156 China) region(688 Hebei) city(1692 Baoding)
```

2. 查询所有匹配查询 IP 地址的 ISDB 条目，包括具有不同协议和端口号的条目，可以指定掩码（模糊查找）。

```bash
FortiGate # diagnose internet-service match root 96.45.46.46 255.255.255.0
Internet Service: 1245324(Fortinet-FortiGuard), matched entry num: 18, matched num: 18
Internet Service: 1245454(Fortinet-FortiGuard.Secure.DNS), matched entry num: 1, matched num: 1
Internet Service: 1245185(Fortinet-Web), matched entry num: 4, matched num: 1024
Internet Service: 1245186(Fortinet-ICMP), matched entry num: 1, matched num: 256
Internet Service: 1245187(Fortinet-DNS), matched entry num: 2, matched num: 512
Internet Service: 1245188(Fortinet-Outbound_Email), matched entry num: 4, matched num: 1024
Internet Service: 1245190(Fortinet-SSH), matched entry num: 1, matched num: 256
Internet Service: 1245191(Fortinet-FTP), matched entry num: 2, matched num: 512
Internet Service: 1245192(Fortinet-NTP), matched entry num: 2, matched num: 512
Internet Service: 1245193(Fortinet-Inbound_Email), matched entry num: 4, matched num: 1024
Internet Service: 1245198(Fortinet-LDAP), matched entry num: 4, matched num: 1024
Internet Service: 1245199(Fortinet-NetBIOS.Session.Service), matched entry num: 2, matched num: 512
Internet Service: 1245200(Fortinet-RTMP), matched entry num: 2, matched num: 512
Internet Service: 1245208(Fortinet-NetBIOS.Name.Service), matched entry num: 1, matched num: 256
Internet Service: 1245184(Fortinet-Other), matched entry num: 2, matched num: 512
```

3. 查询某个 ISDB ID 包含的资源库具体列表，这里以 Google.Web ↗（65537）为例。

```
diagnose internet-service id 65537
```

4. 查询某个 ISDB 的汇总信息，这里以 Google.Web ↗（65537）为例。

```bash
FortiGate # diagnose internet-service id-summary 65537
Version: 00007.03214
Timestamp: 202305221646
Total number of IP ranges: 1192387
Number of Groups: 24
Group(0), Singularity(90), Number of IP ranges(2877)
Group(1), Singularity(89), Number of IP ranges(273689)
Group(2), Singularity(88), Number of IP ranges(49676)
Group(3), Singularity(87), Number of IP ranges(22743)
Group(4), Singularity(86), Number of IP ranges(16913)
Group(5), Singularity(85), Number of IP ranges(29777)
Group(6), Singularity(84), Number of IP ranges(5120)
Group(7), Singularity(83), Number of IP ranges(1290)
Group(8), Singularity(82), Number of IP ranges(266)
Group(9), Singularity(80), Number of IP ranges(93)
Group(10), Singularity(20), Number of IP ranges(886)
Group(11), Singularity(19), Number of IP ranges(990)
Group(12), Singularity(18), Number of IP ranges(7498)
Group(13), Singularity(17), Number of IP ranges(5855)
Group(14), Singularity(16), Number of IP ranges(10700)
Group(15), Singularity(15), Number of IP ranges(129919)
Group(16), Singularity(11), Number of IP ranges(412)
Group(17), Singularity(10), Number of IP ranges(197733)
Group(18), Singularity(9), Number of IP ranges(8129)
Group(19), Singularity(8), Number of IP ranges(3857)
Group(20), Singularity(7), Number of IP ranges(46816)
Group(21), Singularity(6), Number of IP ranges(183359)
Group(22), Singularity(5), Number of IP ranges(184276)
Group(23), Singularity(4), Number of IP ranges(9513)
Internet Service: 65537(Google-Web)
Number of IP ranges: 22984
Number of IP addresses: 19527487
Singularity: 6
Icon Id: 1
Direction: dst
Data source: isdb
```

5. 通过本地 GEO 库查询某个 IP 的地理位置信息。

```bash
FortiGate # diagnose internet-service id 65537 | grep 8.8.8.8

FortiGate # diagnose geoip ip2country 61.149.143.226
61.149.143.226 - China, is not anycast ip
```

6. 通过 FortiGuard GEO 库查询某个 IP 的地理位置信息（需要 FortiGate 可以连接到 Internet），查询到的地理位置信息更为详细。

```bash
FortiGate # diagnose geoip geoip-query 61.149.143.226
IP:61.149.143.226
{
  "city":{
    "geoname_id":10175057,
    "names":{
      "en":"Beijing"
    }
  },
  "continent":{
    "code":"AS",
    "names":{
      "en":"Asia"
    }
  },
  "country":{
    "iso_code":"CN",
    "names":{
      "en":"China"
    }
  },
  "subdivisions":[
    {
      "names":{
        "en":"Beijing"
      }
    }
  ],
  "location":{
    "latitude":39.904202,
    "longitude":116.407402,
    "time_zone":"Asia\/Shanghai"
  }
}
```

按需模式的 ISDB

功能介绍

默认情况下，FortiGate 会使用全尺寸 ISDB 库文件放置在 Flash 盘分区中。在某些较低端的型号中（如 FortiGate-30E、FortiGate-50E 等），由于硬件 Flash 容量较小，使用全尺寸 ISDB 库可能会导致 Flash 分区异常。

ISDB 按需模式将全尺寸 ISDB 文件替换为小得多的文件。此文件仅包含互联网服务的基本条目。在防火墙策略/限速策略/SD-WAN 规则中使用某个 ISDB 服务时，FortiGate 会查询 FortiGuard 以下载对应 ISDB 条目的 IP 地址/端口信息并将其存储在 Flash 驱动器上。防火墙策略中使用的 ISDB 条目的内容在重新启动后仍然存在。

> **重要**
>
> 该功能在 FortiOS 7.2.4 GA 及更新版本支持。

配置步骤

1. 在 CLI 下开启 ISDB 的 on-demand 模式，默认为 standard 模式，注意修改为 on-demand 模式后，已从 FortiGuard 下载的 ISDB 条目将会从 Flash 分区中删除，配置完成后，可以使用“execute update-ffdb-ondemand”下载 on-demand 模式的 ISDB 库，或者等待 FortiGate 的自动更新。

```
FortiGate # config sys global

FortiGate (global) # set internet-service-database ?
mini         Small sized Internet Service database with very limited IP addresses.
standard     Medium sized Internet Service database with most IP addresses.
full         Full sized Internet Service database with all IP addresses.
on-demand    Internet Service database with customer selected IP addresses.

FortiGate (global) # set internet-service-database on-demand
FortiGate (global) # end

Warning: Changing Internet Service database update mode will lead to the removal of all downloaded Internet Service
files.
Do you want to continue? (y/n)y

Please run "execute update-ffdb-ondemand" to do explicit download or wait for automatic schedule update for
configuration of Internet Service database
```

2. 此时查看 FortiGate 的 ISDB 库版本，Flash 中的 ISDB 库已变为 On-Demand 模式，文件为空。

```
FortiGate # diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
————————
Version: 0.00000
Contract Expiry Date: n/a
Last Updated using scheduled update on Wed May 24 10:53:22 2023
Last Update Attempt: Wed May 24 17:23:10 2023
Result: No Updates
```

3. 使用“execute update-ffdb-ondemand”下载 on-demand 模式的 ISDB 库，或者等待 FortiGate 的自动更新完成，再次查看 FortiGate 的 ISDB 库版本，可以看到 FortiGate 已经下载了 On-Demand 模式的 ISDB 文件。

```
FortiGate # diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
————————
Version: 7.03216
Contract Expiry Date: n/a
Last Updated using manual update on Wed May 24 17:27:11 2023
Last Update Attempt: Wed May 24 17:27:11 2023
Result: Updates Installed
```

4. 由于尚未将任何 ISDB 条目应用于防火墙策略/限速策略/SD-WAN 策略，因此 ISDB Summary 信息中的 IP 范围数量和 IP 地址数量为空，这里以 Google.Web ↗（65537）为例。

```
FortiGate # diagnose internet-service id-summary 65537
Version: 00007.03216
Timestamp: 202305231647
Total number of IP ranges: 2858
Number of Groups: 1
Group(0), Singularity(90), Number of IP ranges(2858)
Internet Service: 65537(Google-Web)
Number of IP ranges: 0
Number of IP addresses: 0
Singularity: 0
Icon Id: 1
Direction: dst
Data source: isdb
Country:
Region:
City:
```

5. 将该 ISDB 应用于一条防火墙策略中。

```
config firewall policy
    edit 5
        set name "Google-Web"
        set srcintf "lan"
        set dstintf "wan"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Google-Web"
        set schedule "always"
        set nat enable
    next
end
```

6. 再次查看 Google.Web ↗（65537）Summary 信息，可以看到在防火墙策略引用该 ISDB 后，FortiGate 自动从 FortiGuard 下载了该 ISDB 库。

```
FortiGate # diagnose internet-service id-summary 65537
Version: 00007.03216
Timestamp: 202305231647
Total number of IP ranges: 25767
Number of Groups: 2
Group(0), Singularity(90), Number of IP ranges(2858)
Group(1), Singularity(6), Number of IP ranges(22909)
Internet Service: 65537(Google-Web)
Number of IP ranges: 22909
Number of IP addresses: 19527416
Singularity: 6
Icon Id: 1
Direction: dst
Data source: isdb
Country: 4 8 10 12 16 20 24 28 31 32 36 40 44 48 50 51 52 56 60 64 68 70
        72 76 84 86 90 96 100 104 108 112 116 120 124 132 136 144 148 152 156 158 162
        166 170 175 178 180 184 188 191 192 196 203 204 208 212 214 218 222 226 231 232 233
        234 238 239 242 246 248 250 254 258 260 262 266 268 270 275 276 288 292 296 300 308
...
```

7. 查看其他 ISDB 的 Summary 信息，仍然为空。

```
FortiGate # diagnose internet-service id-summary 1245187
Version: 00007.03216
Timestamp: 202305231647
Total number of IP ranges: 25767
Number of Groups: 2
Group(0), Singularity(90), Number of IP ranges(2858)
Group(1), Singularity(6), Number of IP ranges(22909)
Internet Service: 1245187(Fortinet-DNS)
Number of IP ranges: 0
Number of IP addresses: 0
Singularity: 0
Icon Id: 19
```

```
Direction: dst
Data source: isdb
Country:
Region:
City:
```

DoS 策略

什么是 DoS

FortiGate DoS 策略工作原理

DoS 策略使用

- FortiGate DoS 策略配置
- FortiGate DoS 策略调整

FortiGate DoS 策略排错

什么是 DoS

DoS&DDoS

1. 拒绝服务攻击（英语：denial-of-service attack，简称 DoS 攻击）亦称洪水攻击，是一种网络攻击手法，其目的在于使目标电脑的网络或系统资源耗尽，使服务暂时中断或停止，导致其正常用户无法访问。

2. 当黑客使用网络上两个或以上被攻陷的电脑作为“僵尸”向特定的目标发动“拒绝服务”式攻击时，称为分布式拒绝服务攻击（distributed denial-of-service attack，简称 DDoS 攻击）。

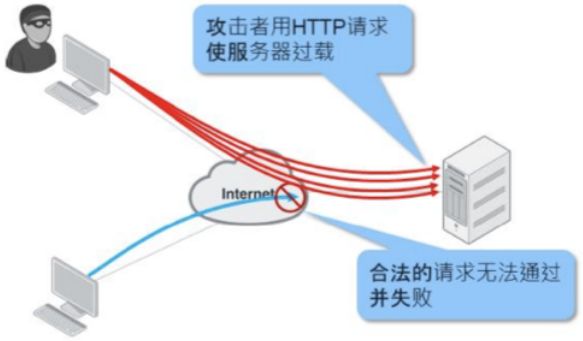
3. DoS 侧重于通过对主机特定漏洞的利用攻击导致网络栈失效、系统崩溃、主机死机而无法提供正常的网络服务功能，从而造成拒绝服务，常见的 DOS 攻击手段有 TearDrop、Land、Jolt、IGMP Nuker、Boink、Smurf、Bonk、OOB 等；扫描也是网络攻击的一种，攻击者在发起网络攻击之前，通常会试图确定目标上开放的 TCP/UDP 端口，而一个开放的端口通常意味着某种应用。

4. DDoS 的表现形式主要有两种，一种为流量攻击，主要是针对网络带宽的攻击，即大量攻击包导致网络带宽被阻塞，合法网络包被虚假的攻击包淹没而无法到达主机；另一种为资源耗尽攻击，主要是针对服务器主机的攻击，即通过大量攻击包导致主机的内存被耗尽或 CPU 被内核及应用程序占完而造成无法提供网络服务。 与 DoS 攻击相比，DDoS 攻击更难以预防，因为黑客可能来自世界各地，DDoS 攻击可以分为两个阶段：

    - 感染阶段：黑客将尝试开发并感染尽可能多的系统，受到感染和破坏的系统将称为 Bot 或 Zombie，这些僵尸和僵尸分散在世界各地，所有这些僵尸或僵尸都将监听黑客的命令。
    - 攻击阶段：黑客向机器人或僵尸发出命令，以攻击目标系统，当这些僵尸程序或僵尸程序获得命令后，它们将开始攻击目标系统。从目标的角度来看，攻击将无处不在。

5. 拒绝服务的攻击也可能会导致目标计算机同一网络中的其他计算机被攻击，互联网和局域网之间的带宽会被攻击导致大量消耗，不但影响目标计算机，同时也影响局域网中的其他电脑。如果攻击的规模较大，整个地区的网络连接都可能会受到影响。

6. 攻击者的会话回消耗所有资源 ——RAM、CPU、端口号，减慢或使目标功能失效，直到不能响应合法的请求。



攻击现象

- 网络异常缓慢（打开文件或访问网站）
- 特定网站无法访问
- 无法访问任何网站
- 垃圾邮件的数量急剧增加
- 无线或有线网络连接异常断开
- 长时间尝试访问网站或任何互联网服务时被拒绝
- 服务器容易断线、卡顿、lag

攻击方式

DDoS 攻击可分为 3 类：带宽消耗、资源匮乏、应用程序编程缺陷。

1. 带宽消耗：
    - 这种攻击试图使系统的带宽饱和。如果黑客可以用尽所有带宽，则黑客可以阻止合法用户访问系统。
    - 例如：客户从 ISP 订阅了 10 Mbps 的互联网带宽。如果黑客能够充分利用 10 Mbps 的 Internet 链接，则合法用户将不会留有带宽。由于没有足够的带宽供合法用户使用，因此他们将无法再访问系统。攻击的示例包括 Smurf 攻击，Fraggle 攻击，Chargen 攻击和放大攻击。
    - 使用 FortiGate 或 FortiDDoS 不能有效地缓解此类 DDoS 攻击。
2. 资源匮乏：
    - 该攻击试图使系统上的资源过载，使其无法使用，资源可以是会话表，CPU 处理时间，磁盘空间等。
    - 攻击示例为 TCP SYN Flood，SlowLoris 等。
    - 可以使用 Fortigate 和 FortiDDoS 有效地缓解此类 DDoS 攻击。与 Fortigate 相比，FortiDDoS 具有更先进和有效的缓解技术。
3. 应用程序编程缺陷：

- 该攻击试图导致系统严重错误。黑客试图将格式错误的数据发送到系统，并且系统崩溃。通过对系统造成严重错误，这使系统无法使用。
- 攻击示例包括缓冲区溢出和编程逻辑错误。
- 使用 FortiGate 和 FortiDDoS 可以有效地缓解此类 DDoS 攻击。与 FortiGate 相比，FortiDDos 具有更先进和有效的缓解技术。

攻击与防护方法

1. 防 DoS 攻击：
   - SYN Flood 攻击：如果到一个目标 IP 地址的新 TCP 连接（包括重传）的 SYN 数据包速率超过配置的阈值，则执行配置的操作。
   - TCP Session 攻击：如果来自一个源/去往一个目的 IP 地址的并发 TCP 连接数超过配置的阈值，则将执行配置的操作。
   - UDP Flood 攻击：如果到一个目标 IP 地址的 UDP 会话速率超过配置的阈值，则将执行配置的操作。
   - UDP Session 攻击：如果来自一个源/去往一个目的 IP 地址的并发 UDP 连接数超过配置的阈值，则将执行配置的操作。
   - ICMP Flood 攻击：如果发送到一个目标 IP 地址的 ICMP 数据包的速率超过配置的阈值，则将执行配置的操作。
   - ICMP Session 攻击：如果来自一个源/去往一个目的 IP 地址的并发 ICMP 连接数超过配置的阈值，则将执行配置的操作。
   - SCTP Flood 攻击：如果发送到一个目标 IP 地址的 SCTP 数据包的速率超过配置的阈值，则将执行配置的操作。
   - CTP Session 攻击：如果来自一个源/去往一个目的 IP 地址的并发 SCTP 连接数超过配置的阈值，则将执行配置的操作。
2. 抗扫描攻击：
   - TCP 协议扫描：根据实际网络情况，当受到 TCP 扫描攻击时，可以配置防 TCP 扫描。 当一个源 IP 地址在 1 秒内将含有 TCP SYN 片段的 IP 封包发送给位于相同目标 IP 地址的不同端口数量大于配置的门限值时，即认为其进行了端口扫描，系统将其标记为 TCP SCAN，并在配置的阻断时间内拒绝来自于该台源主机的所有其它 TCP SYN 包。 启用防 TCP 扫描，可能会占用比较多的内存。
   - UDP 协议扫描：根据实际网络情况，当受到 UDP 扫描攻击时，可以配置防 UDP SCAN 扫描。 当一个源 IP 地址在 1 秒内将含有 UDP 的 IP 封包发送给位于相同目标 IP 地址的不同端口数量大于配置的门限值时，即进行了一次端口扫描，系统将其标记为 UDP SCAN，并在配置的阻断时间内拒绝来自于该台源主机的所有其它 UDP 包。 启用防 UDP 扫描，可能会占用比较多的内存。
   - PING 扫描：根据实际网络情况，当受到 PING 扫描攻击时，可以配置防 PING 扫描。 当一个源 IP 地址在 1 秒内发送给不同主机的 ICMP 封包超过门限值时，即进行了一次地址扫描。此方案的目的是将 ICMP 封包 (通常是应答请求) 发送给各个主机，以期获得至少一个回复，从而查明目标地址。NGFW 设备在内部记录从某一远程源地点发往不同地址的 ICMP 封包数目。当某个源 IP 被标记为地址扫描攻击，则系统在配置的阻断时间内拒绝来自该主机的其它更多 ICMP 封包。 启用防 PING 扫描，可能会占用比较多的内存。
   - 主机抑制时长：设置防扫描功能的阻断时间，当系统检测到扫描攻击时，在配置的时长内拒绝来自于该台源主机的所有其它攻击包，缺省配置为 20 秒。

FortiGate DoS 策略工作原理

DoS 防护

1. 拒绝服务（DoS）策略会检查到达 FortiGate 接口的网络流量是否存在异常模式，这通常表明存在攻击。当攻击者启动与目标的异常大量会话时（穿过或到达 FortiGate），将发生拒绝服务。大量会话会减慢或禁用目标系统，从而阻止合法用户使用它。
2. 在安全策略之前先检查 DoS 策略，以防止攻击触发更多的资源密集型安全保护并减慢 FortiGate 的速度。

DoS 异常

FortiGate 针对特定的异常流量设置了预定义的异常种类，用户无法添加自定义的 DoS 异常，以下是每个异常阈值的解释。

1. L3 异常：
   - ip_src_session（并发）：如果来自一个源 IP 地址的并发 IP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - ip_dst_session（并发）：如果到一个目标 IP 地址的并发 IP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
2. L4 异常
   - tcp_syn_flood（速率）：如果到一个目标 IP 地址的新 TCP 连接（包括重传）的 SYN 数据包速率超过配置的阈值，则执行配置的操作。建议值：2000/s。
   - tcp_port_scan（速率）：如果来自一个源 IP 地址的新 TCP 连接的 SYN 包速率（包括重传）超过了配置的阈值，则将执行配置的操作。建议值：1000/s。
   - tcp_src_session（并发）：如果来自一个源 IP 地址的并发 TCP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - tcp_dst_session（并发）：如果到一个目标 IP 地址的并发 TCP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - udp_flood（速率）：如果到一个目标 IP 地址的 UDP 会话速率超过配置的阈值，则将执行配置的操作。建议值：2000/s。
   - udp_scan（速率）：如果来自一个源 IP 地址的 UDP 会话速率超过配置的阈值，则将执行配置的操作。建议值：2000/s。
   - udp_src_session（并发）：如果来自一个源 IP 地址的并发 UDP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - udp_dst_session（并发）：如果到一个目标 IP 地址的并发 UDP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - icmp_flood（速率）：如果发送到一个目标 IP 地址的 ICMP 数据包的速率超过配置的阈值，则将执行配置的操作。建议值：250/s。
   - icmp_sweep（速率）：如果来自一个源 IP 地址的 ICMP 数据包的速率超过配置的阈值，则将执行配置的操作。建议值：100/s。
   - icmp_src_session（并发）：如果来自一个源 IP 地址的并发 ICMP 连接数超过配置的阈值，则将执行配置的操作。建议值：300。
   - icmp_dst_session（并发）：如果到一个目标 IP 地址的并发 ICMP 连接数超过配置的阈值，则将执行配置的操作。 建议值：1000。
   - sctp_flood（速率）：如果发送到一个目标 IP 地址的 SCTP 数据包的速率超过配置的阈值，则将执行配置的操作。建议值：2000/s。
   - sctp_scan（速率）：如果来自一个源 IP 地址的 SCTP 会话速率超过配置的阈值，则将执行配置的操作。建议值：1000/s。
   - sctp_src_session（并发）：如果来自一个源 IP 地址的并发 SCTP 连接数超过配置的阈值，则将执行配置的操作。建议值：5000。
   - sctp_dst_session（并发）：如果到一个目标 IP 地址的并发 SCTP 连接数超过配置的阈值，则执行配置的操作。 建议值：5000。

基于并发会话数的阈值

对于基于并发会话数的阈值，DoS 策略会将超过设置阈值的连接阻断。这里举个例子：

1. 在 FortiGate 的 DoS 策略中，将 icmp_src_session 的阈值设置为 10，攻击者发送了 10 条匹配异常条件的连接通过 FortiGate。
2. 这 10 条连接将被放通，但第 11 条及以后匹配的连接将被阻断。此时，匹配异常条件的允许通过会话计数器变为 10。
3. 之后，这 10 条连接没有后续报文传输，FortiGate 针对 ICMP 流量的的老化时间为 60s，那么 60s 的计时器到期后，匹配异常条件的允许通过会话计数器被重置为 0。
4. 所以，如果在阻止之前允许进行 10 个会话，则在 60 秒后，将允许再进行 10 个会话。

基于速率的阈值

对于基于速率的阈值，其中阈值以每秒通过或到达的数据包为单位进行度量，“阻止”操作可通过两种方式防止异常流量瘫痪目标系统：

1. continuous：检测到异常后立即阻止数据包，并在速率高于阈值时继续阻止数据包。此为默认配置。

2. periodical：检测到异常后，允许每秒配置的数据包数量。

3. 这里举个例子：

   - 在 FortiGate 的 DoS 策略中，将 icmp_flood 的阈值设置为 10/s，攻击者以 20pps 的速率启动 1000 个数据包的连续 ping 通过 FortiGate。

   - 在 continuous 模式下，如果前 10 个数据包触发了 DoS 策略，则前 10 个数据包通过，阻断剩余的 990 个数据包。

   - 在 periodical 模式下，每秒钟允许通过 10 个数据包，所以会有 500 个数据包被阻断。（注：由于发包速率可能会发生波动，因此通过和阻止的数据包的实际数量可能不准确，但数量应接近定义的阈值。）

   - periodical 模式需要在 CLI 下启动：

```bash
config ips global
    set anomaly-mode {continuous | periodical}
end
```

Quarantine

1. Quarantine 用于阻止来自被认为是恶意行为者的源 IP 地址或对网络有害的流量源的任何其他流量。在隔离期间，将阻止来自源 IP 地址的流量，并将源 IP 地址添加到被禁止的用户列表中。禁止的用户列表保留在内核中，并由防病毒，数据泄漏防护（DLP），DoS 和入侵防护系统（IPS）使用。使用任何这些功能的任何策略都将阻止来自攻击者 IP 地址的流量。该功能仅在 CLI 下可以配置，请参考“FortiGate DoS 策略配置”。

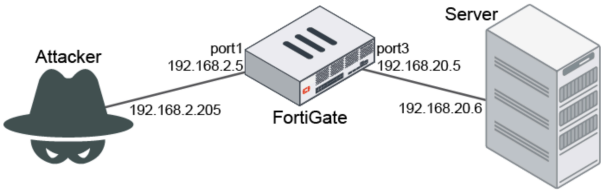
2. 查看 Quarantine 用户列表：

```bash
# diagnose user quarantine list
src-ip-addr       created                 expires                 cause
192.168.118.77     Fri Jan 15 18:47:54 2021 Fri Jan 15 18:57:54 2021 DOS
```

FortiGate DoS 策略排错

DoS 策略排错最好的办法就是查看分析异常日志与 IPS 异常信息。

网络拓扑



icmp_flood 测试

1. Attacker 向 FortiGate 以 50pps 的速率发送总共 3000 个 icmp 报文。

2. 在 FortiGate 上配置阈值为 30pps 的 icmp_flood 的 DoS 策略。

```
config firewall DoS-policy
    edit 1
        set name icmpFlood
        set interface "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set threshold 30
            next
        end
    next
end
```

3. 配置 debug filter，过滤 icmp_flood 攻击的 debug 信息。

```
# diagnose ips anomaly config
DoS sensors in kernel vd 0:
DoS id 1 proxy 0
  0 tcp_syn_flood status 0 log 0 nac 0 action 0 threshold 2000
  ...
  7 udp_dst_session status 0 log 0 nac 0 action 0 threshold 5000
  8 icmp_flood status 1 log 1 nac 0 action 7 threshold 30
  9 icmp_sweep status 0 log 0 nac 0 action 0 threshold 100
  ...
total # DoS sensors: 1.
# diagnose ips anomaly filter id 8
```

4. 使用 Mac OS 或 Linux 发起 icmp_flood 攻击，这里使用 hping3 工具为例（需要自行安装）。

```
summerice@Summerice ~ % sudo hping3 --icmp -i u20000（发包间隔20000微秒，即50pps） -c 3000 192.168.2.50

HPING 192.168.2.50 (en11 192.168.2.50): icmp mode set, 28 headers + 0 data bytes

...

--- 192.168.2.50 hping statistic ---

3000 packets tramitted, 30 packets received, 99% packet loss

round-trip min/avg/max = 0.4/0.5/0.8 ms
```

5. 由以上 hping3 发包结果可知，只有 30 个包通过，剩余 icmp 包被阻断，符合 icmp_flood 防护工作原理。

6. 在攻击期间，在 FortiGate 查看异常列表。

```
# diagnose ips anomaly list

list nids meter:

id=icmp_flood         ip=192.168.2.50 dos_id=1 exp=1000 pps=46 freq=50

total # of nids meters: 1.
```

参数	说明
id = icmp_flood	攻击名称。
ip = 192.168.2.50	触发异常的被攻击对象的 IP 地址，可能是服务器，也可能是客户端。 对于 icmp_flood 攻击，IP 地址为攻击目标 IP 地址。如果是 icmp_sweep 攻击，该 IP 应为攻击源 IP。
dos_id = 1	匹配的 DoS 策略 ID。
exp = 998	剩余过期时间，单位为秒。
pps = 46	命令执行时，已经收到的数据包 pps。
freq = 50	对于基于并发会话数的异常，freq 为并发会话数。 对于基于速率的异常（flood、scan）： continuous 模式：freq 为 pps 中的最大数值，或者时最后一秒收到的数据包的总数。 periodic 模式：freq 为 pps 数值。

日志分析

1. 在 GUI 中访问“日志 & 报告” → 异常，下载 DoS 的异常日志，这里截取攻击刚开始的两条日志举例。

2. 第一条日志中的字段分析。

```
date=2021-1-15 time=18:38:39 eventtime=1605911919824184594 tz="-0800" logid="0720018433" type="utm" subtype="anomaly"
eventtype="anomaly" level="alert" vd="root" severity="critical" srcip=192.168.2.205 srccountry="Reserved"
dstip=192.168.2.50 srcintf="port1" srcintfrole="undefined" sessionid=0 action="clear_session" proto=1 service="PING"
count=1307 attack="icmp_flood" icmpid="0×2195" icmptype="0×08" icmpcode="0×00" attackid=16777316 policyid=1
policytype="DoS-policy" ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly: icmp_flood, 31 > threshold 30,
repeats 28 times" crscore=50 craction=4096 crlevel="critical"
```

参数	说明
msg="anomaly: icmp_flood, 31 > threshold 30	在攻击的刚开始，第一条日志记录了攻击流量超过了 30pps 的阈值。
repeats 28 times	自上次记录日志以来超过阈值的数据包数。
srcip = 192.168.2.205 dstip = 192.168.2.50	攻击流量的源目 IP。
action="clear_session"	清除相关会话，即阻断动作。 如果动作设置为“监视器”且“logging”开启，这里会显示“detected”。

3. 第二条日志中的字段分析。

```
date=2021-1-15 time=18:39:09 eventtime=1605911949826224056 tz="-0800" logid="0720018433" type="utm" subtype="anomaly"
eventtype="anomaly" level="alert" vd="root" severity="critical" srcip=192.168.2.205 srccountry="Reserved"
dstip=192.168.2.50 srcintf="port1" srcintfrole="undefined" sessionid=0 action="clear_session" proto=1 service="PING"
count=1497 attack="icmp_flood" icmpid="0×2195" icmptype="0×08" icmpcode="0×00" attackid=16777316 policyid=1
policytype="DoS-policy" ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly: icmp_flood, 50 > threshold 30, repeats
1497 times" crscore=50 craction=4096 crlevel="critical"
```

参数	说明
msg="anomaly: icmp_flood, 50 > threshold 30	在日志生成的前一秒，检测到了 50 个匹配异常的数据包，超过了配置的阈值 30。
repeats 1497 times	自上次记录日志以来超过阈值的数据包数。

- 由于是持续的攻击流量，FortiGate 为多个数据包每 30s 生成一次日志（注意第二条日志和第一条日志的间隔为 30s）。
- 以下情况 FortiGate 不会再次生成 DoS 异常日志：
  - 相同的攻击 ID 在 5s 内发生了多次。
  - 相同的攻击 ID 在 30s 内发生了多次，且匹配的异常策略动作一致，并且拥有相同的源目 IP。

DoS 策略使用

- FortiGate DoS 策略配置
- FortiGate DoS 策略调整

FortiGate DoS 策略配置

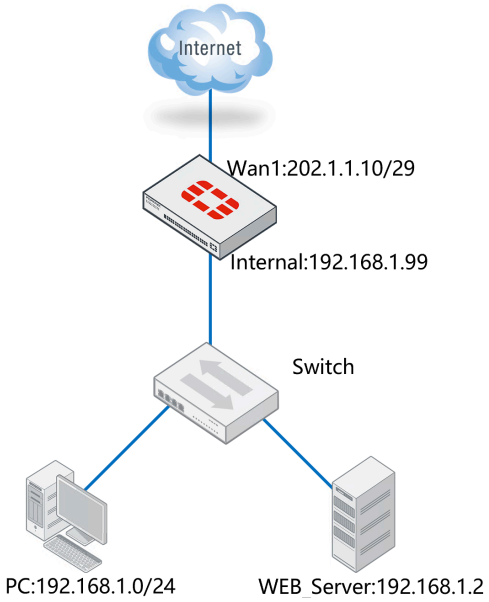
重要

要阻止 DoS 攻击，在位于攻击者和希望保护的所有资源之间的 FortiGate 设备上应用 DoS 策略。 DoS 过滤比包处理过程更早完成，由内核处理。FortiGate 的防 SYN Flood 攻击采用了业界最新的 syncookie 技术，在很少占用系统资源的情况下，可以有效地抵御 SYN Flood 等 DDOS 攻击对受保护服务器的攻击。从而阻止外部的恶意攻击，保护设备和内网，当检测到此类扫描探测时，向用户进行报警提示。

网络需求

Web 服务器 192.168.1.2，映射到外网地址 202.1.1.11，对互联网开放 HTTP 服务，从而增加了服务器被攻击的风险，需要开启 DDoS 功能对服务器进行保护。对所有来自互联网的访问进行 DDOS 防护。

网络拓扑



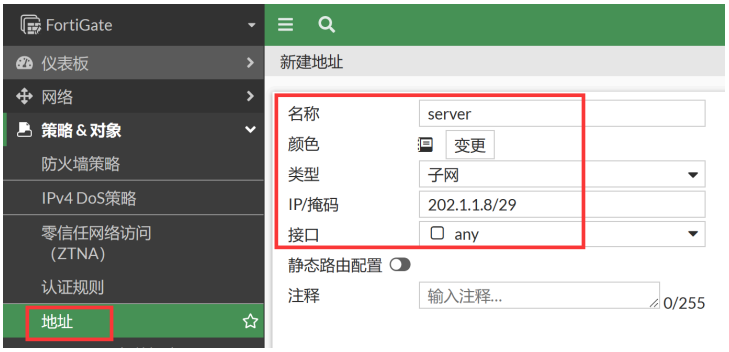
配置要点

- 定义服务器地址
- 定义 DDoS 策略

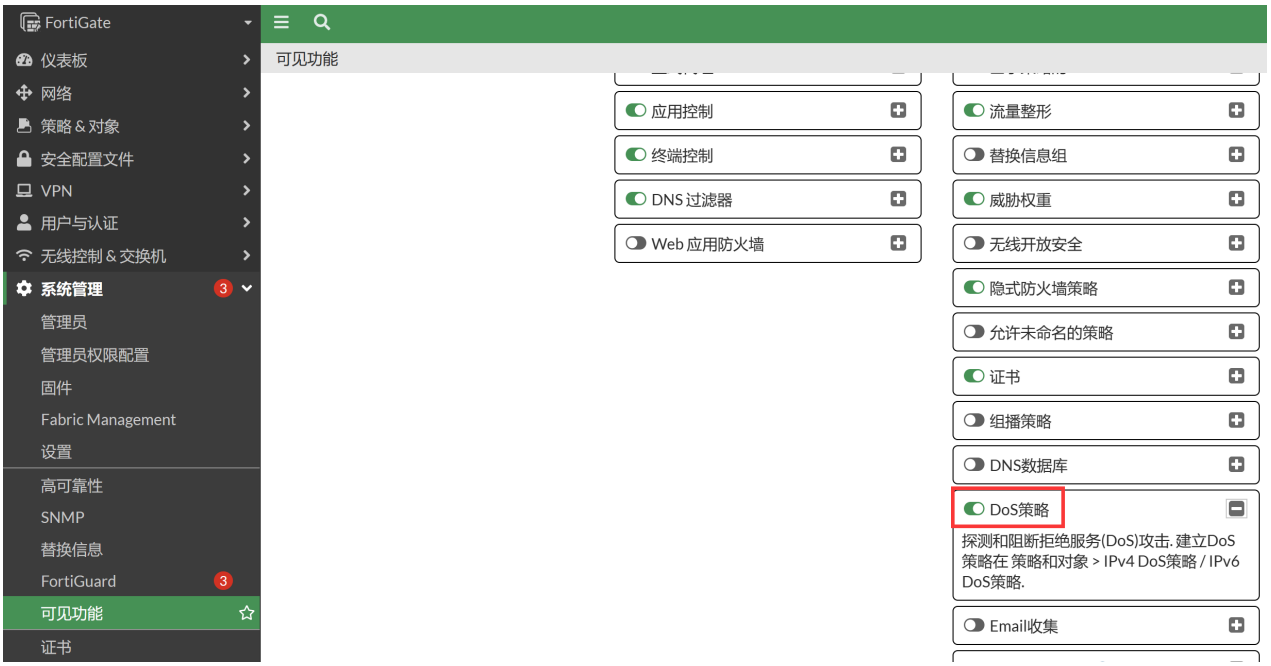
配置步骤

通过 GUI 配置

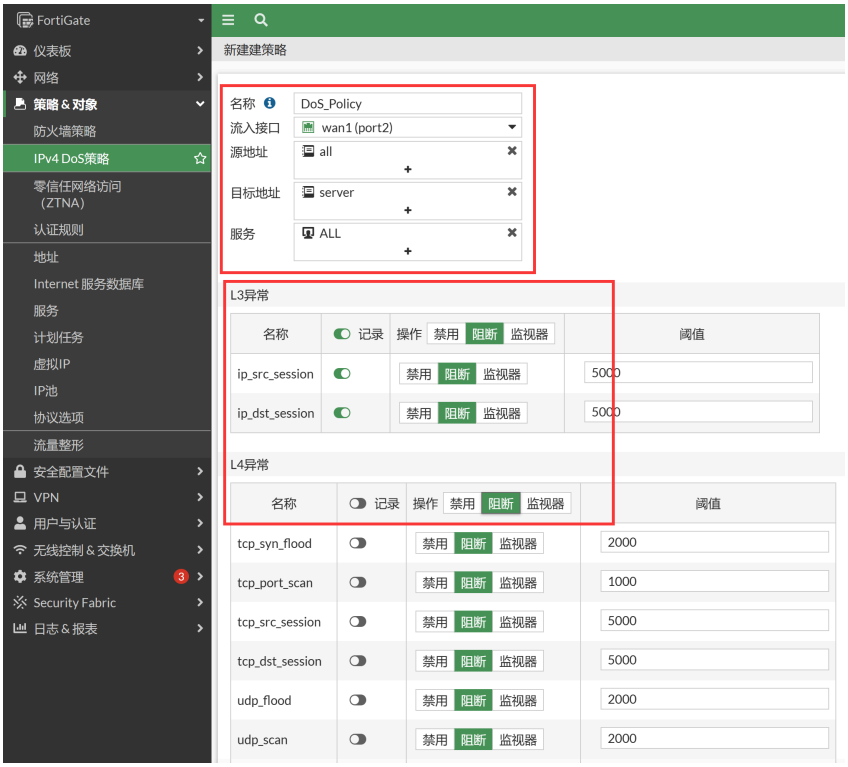
1. 定义服务器地址，进入策略 & 对象 → 地址，新建地址对象，名称配置为“server”，地址类型选择子网：ip/掩码为“202.1.1.8/29”，点击“确认”，该地址范围既包括了服务器地址（202.1.1.11）也包括了防火墙的外网口地址（202.1.1.10）。



2. 配置 DoS 策略齐前，需要确保在系统配置 → 可见功能中开启了 DoS 策略。



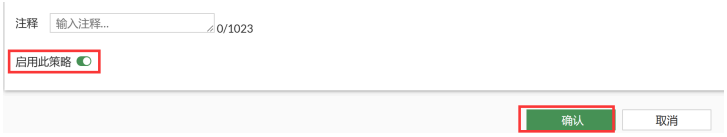
3. 一条 DoS 策略可以包含一个或多个异常配置，通过访问 GUI 的“策略 & 对象” → “IPv4 DoS 策略”或“IPv6 DoS 策略”页面，点击“新建”按钮，并配置参数。



4. 具体参数可以参考如下表格。

参数	说明
名称	输入策略名称。
流入接口	选择 DoS 策略工作的接口。
源地址	选择攻击流量的源地址。
目标地址	选择攻击流量的目标地址。 此处的地址应为与 FortiGate 接口相关联的地址，例如接口地址、接口第二地址、VIP 地址。
服务	选择攻击流量的目标服务或服务组。 可以配置为 ALL，或为了节省 FortiGate 资源，只选择 FortiGate 接口会应答的服务。
L3 异常 L4 异常	配置异常: Logging：启用/禁用特定异常或所有异常的日志记录。当操作为“阻断”或“监视器”时，将记录异常流量。 动作：选择达到阈值时需要执行的操作： 禁用：不检查异常流量。 阻断：阻断异常流量。 监视器：允许异常流量，但是如果启用了 logging，则记录日志信息。 Threshold：触发异常动作需要达到的流量速率或并发数阈值。
注释	可选项。

5. 启用此策略，然后点击确认。



通过 CLI 配置

1. 相关 CLI 命令。

```
config firewall DoS-policy
    edit 1
        set name "DoS_Policy"
        set interface "port2"
        set srcaddr "all"
        set dstaddr "server"
        set service "ALL"
        config anomaly
            edit "tcp_syn_flood"
                set status enable
                set action block
                set threshold 2000
            next
            edit "tcp_port_scan"
                set status enable
                set action block
                set threshold 1000
            next
            edit "tcp_src_session"
                set status enable
                set action block
                set threshold 5000
            next
            edit "tcp_dst_session"
                set status enable
                set action block
```

```
            set threshold 5000
        next
        edit "udp_flood"
            set status enable
            set action block
            set threshold 2000
        next
        edit "udp_scan"
            set status enable
            set action block
            set threshold 2000
        next
        edit "udp_src_session"
            set status enable
            set action block
            set threshold 5000
        next
        edit "udp_dst_session"
            set status enable
            set action block
            set threshold 5000
        next
        edit "icmp_flood"
            set status enable
            set action block
            set threshold 250
        next
        edit "icmp_sweep"
            set status enable
            set action block
            set threshold 100
        next
        edit "icmp_src_session"
            set status enable
            set action block
            set threshold 300
        next
        edit "icmp_dst_session"
            set status enable
            set action block
            set threshold 1000
        next
        edit "ip_src_session"
            set status enable
            set log enable
            set action block
            set threshold 5000
        next
        edit "ip_dst_session"
            set status enable
            set log enable
            set action block
            set threshold 5000
        next
        edit "sctp_flood"
            set status enable
            set action block
            set threshold 2000
        next
        edit "sctp_scan"
            set status enable
            set action block
            set threshold 1000
        next
        edit "sctp_src_session"
            set status enable
            set action block
            set threshold 5000
        next
        edit "sctp_dst_session"
            set status enable
            set action block
```

```
                set threshold 5000
            next
        end
    next
end
```

2. 具体参数可以参考如下表格。

参数	说明
`name <string>`	策略名称。
`interface <string>`	DoS 策略工作的接口。
`srcaddr <string>`	攻击流量的源地址。
`dstaddr <string>`	攻击流量的目标地址。 此处的地址应为与 FortiGate 接口相关联的地址，例如接口地址、接口第二地址、VIP 地址。
`service <string>`	攻击流量的目标服务或服务组。 可以配置为 ALL，或为了节省 FortiGate 资源，只选择 FortiGate 接口会应答的服务。
status {enable or disable}	启用或关闭该异常。
log {enable or disable}	启用/禁用特定异常或所有异常的日志记录。当操作为“阻断”或“监视器”时，将记录异常流量。
action {pass or block}	选择达到阈值时需要执行的操作： pass：允许异常流量，但是如果启用了 logging，则记录日志信息。 block：阻断异常流量。
quarantine {none or attacker}	Quarantine 方法： none：关闭 Quarantine 功能。 attacker：阻断所有来自攻击者 IP 的流量，然后将攻击者 IP 加入用户黑名单。
quarantine-expiry <###d##h##m>	设置 Quarantine 的时长，格式为天、分钟、分钟（###d##h##m），范围 1m - 364d23h59m, 默认为 5m）。该配置仅在 Quarantine 被配置为 attacker 时可用。
quarantine-log {enable or disable}	启用或禁用 Quarantine 日志（默认禁用），该配置仅在 Quarantine 被配置为 attacker 时可用。
`threshold <integer>`	触发异常动作需要达到的流量速率或并发数阈值。

FortiGate DoS 策略调整

1. 应该如何调整 DoS 策略以防误报或阻止正常流量？DoS 策略不依赖于会话状态，所以 FortiGate 的出站和入站流量都会被处理。例如如下流量抓包：

```
# di sniff pack Vlan_11 'port 80 and tcp[tcpflags] == tcp-syn' 1 25
interfaces=[Vlan_11]
filters=[port 80 and tcp[tcpflags] == tcp-syn]
0.876904 10.95.13.204.24088 → 10.95.136.204.80: syn 3585436935
0.894848 10.95.4.223.7273 → 10.95.128.117.80: syn 2436279189
0.947586 10.95.10.90.5871 → 10.95.132.100.80: syn 1124757321
0.972220 10.95.5.29.26781 → 10.95.128.217.80: syn 2500978264
```

2. 基于这一点，你需要使用如下推荐的配置。

从 Internet 保护内网服务器

1. 确保仅将内网服务器映射的 IP（比如 VIP）配置为 DoS 策略的目的地址。
2. 在被保护的服务器上开启需要的服务。在 DoS 策略的服务中仅配置服务器对外开启的服务。
3. DoS 策略中仅配置与服务器开启的服务有关的异常。例如服务器只开启了 SMTP 服务，那么在配置 DoS 策略时，只配置 tcp_syn_flood、tcp_src_session 或 ip_src_session 异常。
4. 将这些异常配置到一个 DoS 策略中，根据实际使用情况，阈值设置为正常的值。例如配置 tcp_src_session 异常，对于单个源客户端来说，最大的并发会话数为 100，则使用如下配置（对于每一个异常，都有一个“正常”的阈值，需要根据实际情况进行配置）：

```
config firewall DoS-policy
    edit 1
        set interface "wan1"
        set srcaddr "all"
        set dstaddr "200.201.202.1"
        set service "SMTP"
            config anomaly
                edit "tcp_src_session"
                    set status enable
                    set action block
                    set quarantine attacker
                    set quarantine-expiry 10
                    set quarantine-log enable
                    set threshold 100
                next
            end
    next
end
```

5. 如果使用 tcp_dst_session 或 ip_dst_session，则 DoS 策略将限制服务器处理的并发会话数，不建议这么使用。

从内网保护 FortiGate

1. 将 DoS 策略的源配置为本地网络的网段（不要使用 any）。
2. 将 DoS 策略的目的配置为 any（攻击流量会通过默认路由到达防火墙），服务配置为 ALL。
3. 开启这些异常：tcp_syn_flood、tcp_dst_session、ip_dst_session、tcp_port_scan，并根据平时使用情况配置正常的阈值。
4. 需要注意的是，在 UDP 相关异常和 tcp_src 异常的情况下，某些流量可能是由 torrent（P2P）或 Skype 软件导致的。

流量整形

流量整形介绍

流量整形策略

- 穿越流量
- 本地流量

流量整形配置文件

流量整形器

- 共享流量整形器
- 每 IP 整形器

NP7 的流量整形

- NP7 流量整形介绍
- Policing 模式
- Shaping 模式

流量整形介绍

功能介绍

FortiGate 通过对网络流量进行带宽限制和优先级排序来提供服务质量（QoS）。流量整形是 FortiGate 用来提供 QoS 的一种技术。流量整形的方法是在流量拥堵期间，将高优先级的流量优先转发。为重要业务流量保证稳定效果，同时对不太重要的流量进行限制。

FortiGate 的 NP6 流量整形可配置为 policing 和 queuing 两种模式。两者的区别如下：

- Policing：当流量超过配置的带宽限制时，流量将被丢弃。
- Quening：当流量超过配置的带宽限制时，流量会延迟传输并存放在队列中，直到带宽释放。如果队列已满，流量会被丢弃。

Policing 和 Quening 模式都可以通过设置带宽限制来确定流量的优先级，并提供带宽保证和最大带宽服务。但实现方式不同，因为 Quening 使用队列，而 Policing 不使用队列：

- 在 Quening 模式下，数据包从接口发送之前，要使用 RED 或 FIFO 等算法将其排入队列。内核根据 HTB 算法对数据包进行去队列，然后再将其发送出去。
- 在 Policing 模式下，如果流量超过了分配的带宽，会被直接丢弃。

配置方法

在 FortiGate 上配置流量整形有 3 种方法。如下列出了 3 种方法及其功能。

> **重要**
>
> 如果 3 种方法都配置了，则第 1 种优先于第 2 种，第 2 种优先于第 3 种。

方法	Policing		Quening
	流量优先级	带宽保证保证与最大带宽	流量队列
Traffic shaping profile **流量整形配置文件**	支持	支持，基于出接口带宽的百分比	支持
Traffic shaper **流量整形器**	支持	支持，基于速率	不支持
Global traffic prioritization **全局流量优先级**	支持	不支持	不支持

其中“Traffic shaping profile”方式可配置为 Policing 或 Quening 模式。Quening 模式可以在配置流量整形时提供队列相关选项。每种方法的实现方式略有不同，下文主要介绍 Policing 模式的功能和使用方法。

流量优先级

FortiGate 可以将数据包划分为不同的优先级，以便优先处理某些流量，不同的流量整形方法，划分优先级的方法如下：

- Traffic shaping profile（流量整形配置文件）：不同的流量可以配置被分到不同的本地 Class，最多可以配置 30 个 Class。每个 Class 的流量可配置为 5 个优先级。
- Traffic shaper（流量整形器）：流量优先级可分为 high（2）、medium（3）或 low（4）级别。被带宽保证的流量会自动使用 critical（1）级别。
- Global traffic prioritization（全局流量优先级）：根据流量报文 IP 头中携带的 ToS（服务类型）或 DSCP 值，流量优先级分为 high（2）、medium（3）或 low（4）。

带宽保证与最大带宽

配置带宽保证的目的是分配总带宽的一部分优先给特定流量，以保证特定流量优先传输。每种流量整形的带宽保证方法的配置方法和处理方式都不同：

- Traffic shaping profile（流量整形配置文件）应用于设备流量出接口。如上文所属，Traffic shaping profile 通过为不同的流量分配不同的 Class 来区分流量优先级，每个 Class 配置的保证带宽为出接口总带宽（需要配置）的百分比（而不是具体带宽数值），所有 Class 共同使用接口最大配置带宽的 100%（也可小于 100%，但不能大于 100%）。每个 Class 中配置的优先级并不重要，保证带宽配置始终会生效。
- Traffic shaper（流量整形器）对带宽保证没有硬性限制。管理员需要了解分配给所有流量整形器的保证带宽，以免超过出接口的总带宽。被流量整形器带宽保证的流量优先级为 1。如果优先级为 1 的总流量超过了总出口带宽，流量就会被丢弃。

- 配置接口最大带宽限定了出接口可使用的最大带宽。它被配置为流量整形配置文件中外带宽的百分比。它被配置为流量整形器的速率。

配置出接口带宽

流量整形通常是为离开 FortiGate 的出口流量配置的。因此，在所有流量整形配置方法中，都必须定义出接口带宽，以便进行流量优先级排序。在 Traffic shaping profile 中定义保证带宽和最大带宽时，也需要定义出接口带宽。

对于 Traffic shaper，配置出接口带宽不是应用最大带宽限制的必要条件；但对于保证带宽，配置外带宽则是必要条件。流量整形器上保证带宽限制下的流量优先级为 1。如果未配置外带宽，则不会进行流量优先级排序，优先级也毫无意义。

Traffic shaping policy

流量整形配置文件和流量整形器是监控流量的方法。流量整形策略用于将流量映射到流量整形器或将其分配到一个类。 流量整形策略是根据特定 IP 头字段和/或上层标准匹配流量的规则。例如，它可以根据源和目标 IP、服务、应用程序和 URL 类别来匹配流量。一种常见的用例是根据 IP 标头中的 ToS 或 DS（差异化服务）字段匹配流量。这允许从下游设备的流量中读取服务类型或差异化服务（DiffServ）标签，并在 FortiGate 上进行相应的优先级排序。

DSCP matching and DSCP marking

DSCP 匹配和 DSCP 标记可在防火墙整形策略和常规防火墙策略上执行。DSCP 匹配用于匹配入口流量的 DSCP 标记，DSCP 标记用于更改出口流量的 DSCP 标记。

在防火墙整形策略和常规防火墙策略中，使用 `tos` 和 `tos-mask` 字段执行 DSCP 匹配。使用 `diffserv-forward` 和 `diffserv-reverse` 字段执行 DSCP 标记。

流量整形配置文件

施工中……

流量整形策略

- 穿越流量
- 本地流量

穿越流量

施工中……

本地流量

施工中……

流量整形器

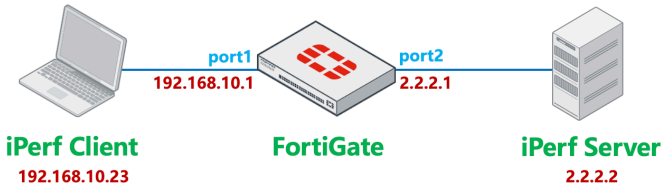
- 共享流量整形器
- 每 IP 整形器

共享流量整形器

网络需求

基于多 IP 共享进行限速，限制 Iperf Client 传输速度，上行和下行速度分别为 5Mbps。

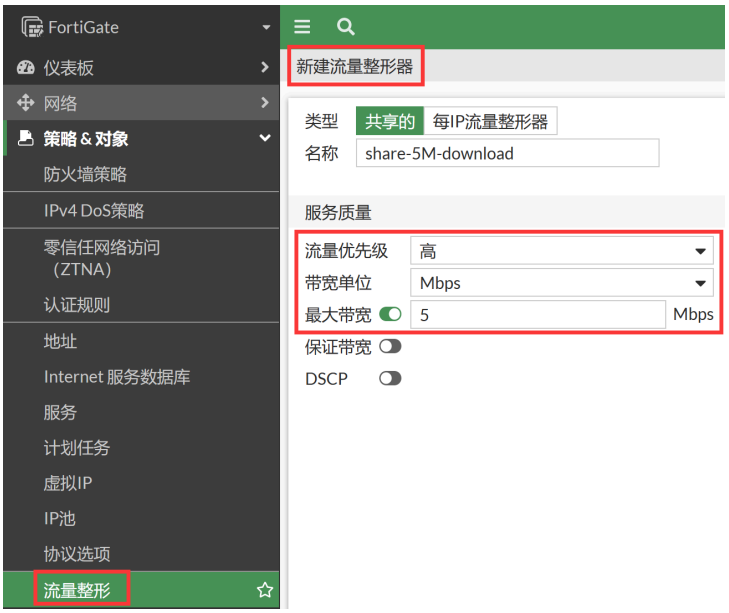
网络拓扑



配置步骤

1. 进入策略 & 对象 → 流量整形 → 流量整形器，分别新建上传和下载的流量整形器，选择类型为共享的，最大带宽配置为 5Mbps。





```
config firewall shaper traffic-shaper
    edit "share-5M-upload"
        set maximum-bandwidth 5
        set bandwidth-unit mbps
    next
    edit "share-5M-download"
        set maximum-bandwidth 5
        set bandwidth-unit mbps
    next
end
```

2. 进入策略 & 对象 → 流量整形 → 流量整策略，新建流量整形策略，源接口为内网口 port1，出接口为 port2，服务配置为 iperf 打流使用的端口，这里以 UDP 5001 为例。开启 Apply Shaper，开启共享流量整形器，引用上传限速 5M，开启反向整形器，引用下载 5M。




```
config firewall shaping-policy
    edit 1
        set name "5001"
        set service "udp_5001"
        set srcintf "port1"
        set dstintf "port2"
        set traffic-shaper "share-5M-upload"
        set traffic-shaper-reverse "share-5M-download"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

结果验证

1. 客户端 IPerf 打流测试打 30M 流（上传）。

2. 客户端端观察。

```
C:\iperf-3.1.3-win64>iperf3.exe -c 2.2.2.2 -p 5001 -u -b 30M -l 1K
Connecting to host 2.2.2.2, port 5001
[  4] local 192.168.10.23 port 49590 connected to 2.2.2.2 port 5001
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[ 4]   0.00-1.00   sec 3.30 MBytes  27.7 Mbits/sec  3380
[ 4]   1.00-2.00   sec 3.61 MBytes  30.2 Mbits/sec  3692
[ 4]   2.00-3.00   sec 3.69 MBytes  30.9 Mbits/sec  3774
```

```
[ 4]   3.00-4.00   sec 3.66 MBytes  30.7 Mbits/sec  3748
[ 4]   4.00-5.00   sec 3.46 MBytes  29.0 Mbits/sec  3546
[ 4]   5.00-6.00   sec 3.66 MBytes  30.7 Mbits/sec  3745
[ 4]   6.00-7.00   sec 3.39 MBytes  28.4 Mbits/sec  3468
[ 4]   7.00-8.00   sec 3.53 MBytes  29.6 Mbits/sec  3612
[ 4]   8.00-9.00   sec 3.62 MBytes  30.3 Mbits/sec  3705
[ 4]   9.00-10.00  sec 3.66 MBytes  30.7 Mbits/sec  3743
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 4]   0.00-10.00  sec 35.6 MBytes  29.8 Mbits/sec  0.196 ms 29694/36412 (82%)
[  4] Sent 36412 datagrams
iperf Done.
```

3. 服务器端观察，接收速率被限制在 5Mbps 左右。

```
C:\iperf-3.1.3-win64>iperf3.exe -s -p 5001
-----------------------------------------------------------
Server listening on 5001
-----------------------------------------------------------
Accepted connection from 2.2.2.1, port 61655
[ 5] local 2.2.2.2 port 5001 connected to 2.2.2.1 port 49590
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 5]   0.00-1.00  sec  1.08 MBytes  9.05 Mbits/sec  0.205 ms 2219/3325 (67%)
[ 5]   1.00-2.00   sec  628 KBytes  5.14 Mbits/sec  0.155 ms 3072/3700 (83%)
[ 5]   2.00-3.00   sec  641 KBytes  5.25 Mbits/sec  0.154 ms 3118/3759 (83%)
[ 5]   3.00-4.00   sec  627 KBytes  5.13 Mbits/sec  0.175 ms 2952/3579 (82%)
[ 5]   4.00-5.00   sec  610 KBytes  5.00 Mbits/sec  0.132 ms 3093/3703 (84%)
[ 5]   5.00-6.00   sec  635 KBytes  5.20 Mbits/sec  0.199 ms 3044/3679 (83%)
[ 5]   6.00-7.00  sec   591 KBytes  4.84 Mbits/sec  0.192 ms 2924/3515 (83%)
[ 5]   7.00-8.00   sec  616 KBytes  5.04 Mbits/sec  0.190 ms 3060/3676 (83%)
[ 5]   8.00-9.00   sec  628 KBytes  5.15 Mbits/sec  0.202 ms 3041/3669 (83%)
[ 5]   9.00-10.00  sec  635 KBytes  5.20 Mbits/sec  0.172 ms 3077/3712 (83%)
[ 5]  10.00-10.03  sec 1.00 KBytes   286 Kbits/sec  0.196 ms 94/95 (99%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 5]   0.00-10.03  sec 0.00 Bytes  0.00 bits/sec  0.196 ms 29694/36412 (82%)
-----------------------------------------------------------
```

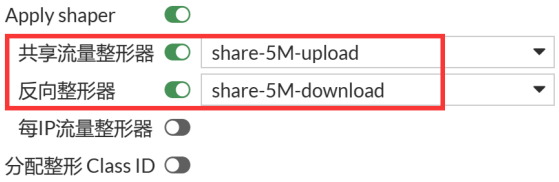
4. 查看会话状态，可以看到共享限速的标识信息。

```
session info: proto=17 proto_state=01 duration=11 expire=179 timeout=0 flags=00000000 sockflag=00000000 sockport=0
av_idx=0 use=4
origin-shaper=share-5M-upload    //应用的共享限速
prio=2 guarantee 0Bps max 655360Bps traffic 655739Bps drops 36587508B
reply-shaper=
per_ip_shaper=
class_id=0 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty os route_preserve
statistic(bytes/packets/allow_err): org=8138304/7737/1 reply=32/1/1 tuples=2
tx speed(Bps/kbps): 694987/5559 rx speed(Bps/kbps): 2/0
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 192.168.10.23:49768->2.2.2.2:5001(2.2.2.1:49768)
hook=pre dir=reply act=dnat 2.2.2.2:5001->2.2.2.1:49768(192.168.10.23:49768)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=00001102 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0

FGT-1 # diagnose  firewall shaper  traffic-shaper  list
name share-5M-upload
maximum-bandwidth 640 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 640 KB/sec
priority 2
overhead 0
tos ff
packets dropped 965353
bytes dropped 1015551356

FGT-1 # diagnose  netlink interface list  port2
if=port2 family=00 type=1 index=4 mtu=1500 link=0 master=0
```

```
ref=17 state=off start fw_flags=0 flags=up broadcast run multicast
Qdisc=pfifo_fast hw_addr=00:0c:29:e1:7d:41 broadcast_addr=ff:ff:ff:ff:ff:ff
stat: rxp=21438 txp=465340 rxb=12605783 txb=478598536 rxe=0 txe=0 rxd=0 txd=0 mc=0 collision=0
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=17

FGT-1 # diagnose netlink qlen get port2
TX queue length for interface port2 is 1000
```

注意事项

共享带宽限速上行和下行限速需要分别调用，彼此之间独立工作，特别注意别只限制了上传速度，而没有配置下载速度，然后测速的时候发现限速不准确，标准的上行、下行速度限制为下图所示。

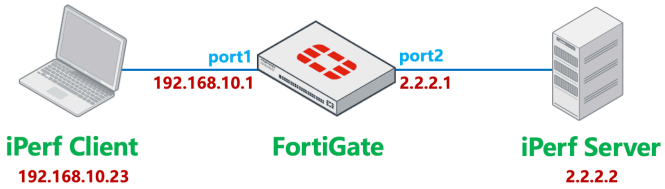


每 IP 整形器

网络需求

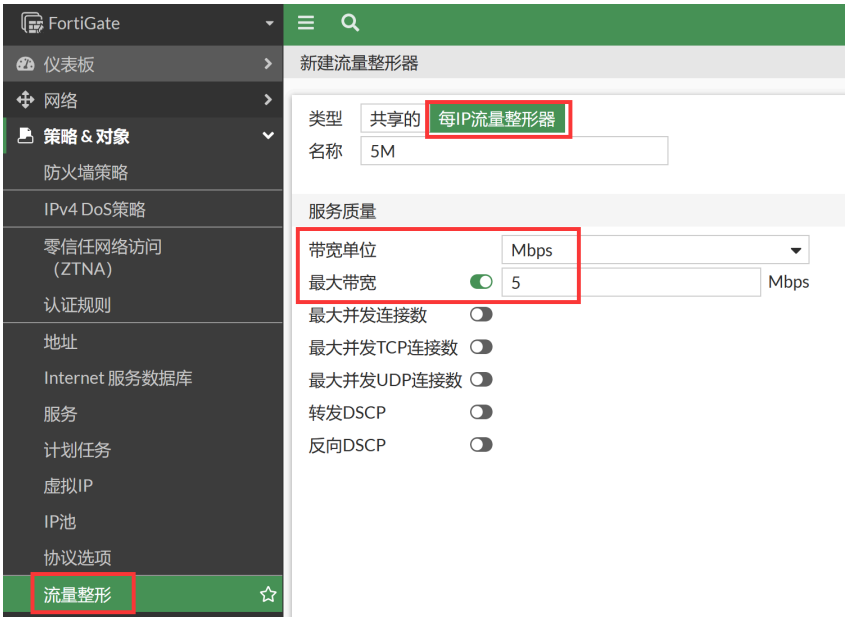
基于每 IP 对客户端进行限速，限制 Iperf Client 传输速度（包括上行和下行速度总和）为 5Mbps。

网络拓扑



配置步骤

1. 进入策略 & 对象 → 流量整形 → 流量整形器，新建流量整形器，选择类型为每 IP 流量整形器，最大带宽配置为 5Mbps。




```
config firewall shaper per-ip-shaper
    edit "5M"
        set max-bandwidth 5
        set bandwidth-unit mbps
    next
end
```

2. 进入策略 & 对象 → 流量整形 → 流量整策略，新建流量整形策略，源接口为内网口 port1，出接口为 port2，服务配置为 iperf 打流使用的端口，这里以 UDP 5001 为例。开启 Apply Shaper，并引用上步配置的每 IP 限速整形器。




```
config firewall shaping-policy
    edit 1
        set name "5M"
        set service "udp_5001"
        set srcintf "port1"
        set dstintf "port2"
        set per-ip-shaper "5M"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

结果验证

1. 客户端 IPerf 打流测试打 30M 流（上传）。

2. 客户端端观察。

```
C:\iperf-3.1.3-win64>iperf3.exe -c 2.2.2.2 -p 5001 -u -b 30M -l 1K
Connecting to host 2.2.2.2, port 5001
[  4] local 192.168.10.23 port 54998 connected to 2.2.2.2 port 5001
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[ 4]   0.00-1.00   sec 3.27 MBytes  27.4 Mbits/sec  3350
[ 4]   1.00-2.00   sec 3.53 MBytes  29.6 Mbits/sec  3619
[ 4]   2.00-3.00   sec 3.64 MBytes  30.6 Mbits/sec  3732
[ 4]   3.00-4.00   sec 3.77 MBytes  31.6 Mbits/sec  3860
[ 4]   4.00-5.00   sec 3.42 MBytes  28.7 Mbits/sec  3501
[ 4]   5.00-6.00   sec 3.57 MBytes  30.0 Mbits/sec  3660
[ 4]   6.00-7.00   sec 3.67 MBytes  30.8 Mbits/sec  3763
[  4]   7.00-8.00  sec  3.68 MBytes  30.8 Mbits/sec  3768
[ 4]   8.00-9.00   sec 3.34 MBytes  28.1 Mbits/sec  3421
[ 4]   9.00-10.00  sec 3.56 MBytes  29.8 Mbits/sec  3641
```

```
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 4]   0.00-10.00  sec 35.5 MBytes  29.7 Mbits/sec  0.180 ms 29610/36310 (82%)
[  4] Sent 36310 datagrams
iperf Done.
```

3. 服务器端观察，接收速率被限制在 5Mbps 左右。

```
C:\iperf-3.1.3-win64>iperf3.exe -s -p 5001
-----------------------------------------------------------
Server listening on 5001
-----------------------------------------------------------
Accepted connection from 2.2.2.1, port 61145
[ 5] local 2.2.2.2 port 5001 connected to 2.2.2.1 port 54998
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 5]  0.00-1.00   sec  1.07 MBytes 9.00 Mbits/sec  0.159 ms  2162/3262 (66%)
[ 5]   1.00-2.00   sec  616 KBytes  5.05 Mbits/sec  0.186 ms 3071/3687 (83%)
[ 5]   2.00-3.00   sec  635 KBytes  5.21 Mbits/sec  0.153 ms 3072/3707 (83%)
[ 5]   3.00-4.00   sec  654 KBytes  5.36 Mbits/sec  0.138 ms 3127/3781 (83%)
[ 5]   4.00-5.00   sec  597 KBytes  4.89 Mbits/sec  0.160 ms 2989/3586 (83%)
[ 5]   5.00-6.00   sec  621 KBytes  5.09 Mbits/sec  0.117 ms 3049/3670 (83%)
[ 5]   6.00-7.00   sec  642 KBytes  5.25 Mbits/sec  0.141 ms 3148/3790 (83%)
[ 5]   7.00-8.00   sec  632 KBytes  5.18 Mbits/sec  0.107 ms 3074/3706 (83%)
[ 5]   8.00-9.00   sec  587 KBytes  4.80 Mbits/sec  0.160 ms 2851/3438 (83%)
[ 5]   9.00-10.00  sec  615 KBytes  5.04 Mbits/sec  0.187 ms 2961/3576 (83%)
[ 5]  10.00-10.02  sec 1.00 KBytes   409 Kbits/sec  0.180 ms 106/107 (99%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[ 5]   0.00-10.02  sec 0.00 Bytes  0.00 bits/sec  0.180 ms 29610/36310 (82%)
-----------------------------------------------------------
```

4. 查看会话状态，可以看到每 IP 限速的标识信息。

```
session info: proto=17 proto_state=01 duration=33 expire=160 timeout=0 flags=00000000 sockflag=00000000 sockport=0
av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=5M    //被每IP限速标识
class_id=0 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty per_ip route_preserve
statistic(bytes/packets/allow_err): org=9691056/9213/1 reply=32/1/1 tuples=2
tx speed(Bps/kbps): 291285/2330 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=3->4/4->3 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 192.168.10.23:49631->2.2.2.2:5001(2.2.2.1:49631)
hook=pre dir=reply act=dnat 2.2.2.2:5001->2.2.2.1:49631(192.168.10.23:49631)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=00000e61 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0

diagnose  firewall shaper  per-ip-shaper list
name 5M
maximum-bandwidth 640 KB/sec
maximum-concurrent-session 0
tos ff/ff
packets dropped 94884
bytes dropped 99717192
addr=192.168.10.23 status: bps=5250320 ses=3
```

注意事项

每 IP 带宽限速对上行和下行同时生效。只需要配置为 5M，则上行和下行速度总和限速为 5M。

重要

With per-IP traffic shaping, you can limit each IP address's behavior to avoid a situation where one user uses all of the available bandwidth. In addition to controlling the maximum bandwidth used per IP address, you can also define the maximum number of concurrent sessions for an IP address. For example, if you apply a per-IP shaper of 1 Mbps to your entire network, FortiOS allocates each user/IP address 1 Mbps of bandwidth. Even if the network consists of a single user, FortiOS allocates them 1 Mbps. If there are ten users, each user gets 1 Mbps of bandwidth, totaling 10 Mbps of outgoing traffic.

For shared shapers, all users share the set guaranteed and maximum bandwidths. For example, if you set a shared shaper for all PCs using an FTP service to 10 Mbps, all users uploading to the FTP server share the 10 Mbps.

Shared shapers affect upload speed. If you want to limit the download speed from the FTP server in the example, you must configure the shared shaper as a reverse shaper. Per-IP shapers apply the speed limit on both upload and download operations.

The following example shows how to apply a per-IP shaper to a traffic shaping policy. This shaper assigns each user a maximum bandwidth of 1 Mbps and allows each user to have a maximum of ten concurrent connections to the FTP server. In the example, FortiOS communicates with users using port10 and the FTP server using port9.

NP7 的流量整形

- NP7 流量整形介绍
- Policing 模式
- Shaping 模式

NP7 流量整形介绍

施工中……

Policing 模式

施工中……

Shaping 模式

施工中……

UTM

入侵防御

- 入侵防御简介
- 入侵防御配置
- 虚拟补丁

病毒防护

- 反病毒简介
- 反病毒配置
- 基于文件 Hash 的 AV 豁免

Web 过滤功能

- Web 过滤常见问题

  Chrome/Edge 访问网页无法被拦截
- Web 过滤简介
- Web 过滤配置
- URL 过滤
- URL 过滤表达式

应用控制

- 应用控制简介
- 配置应用控制

DNS 过滤

- 透明 DNS 转发器

SSL/SSH 检查

- 检测模式
- SSL/SSH 检测配置文件
- certificate-inspection
- deep-inspection
- 保护 SSL 服务器

入侵防御

- 入侵防御简介
- 入侵防御配置
- 虚拟补丁

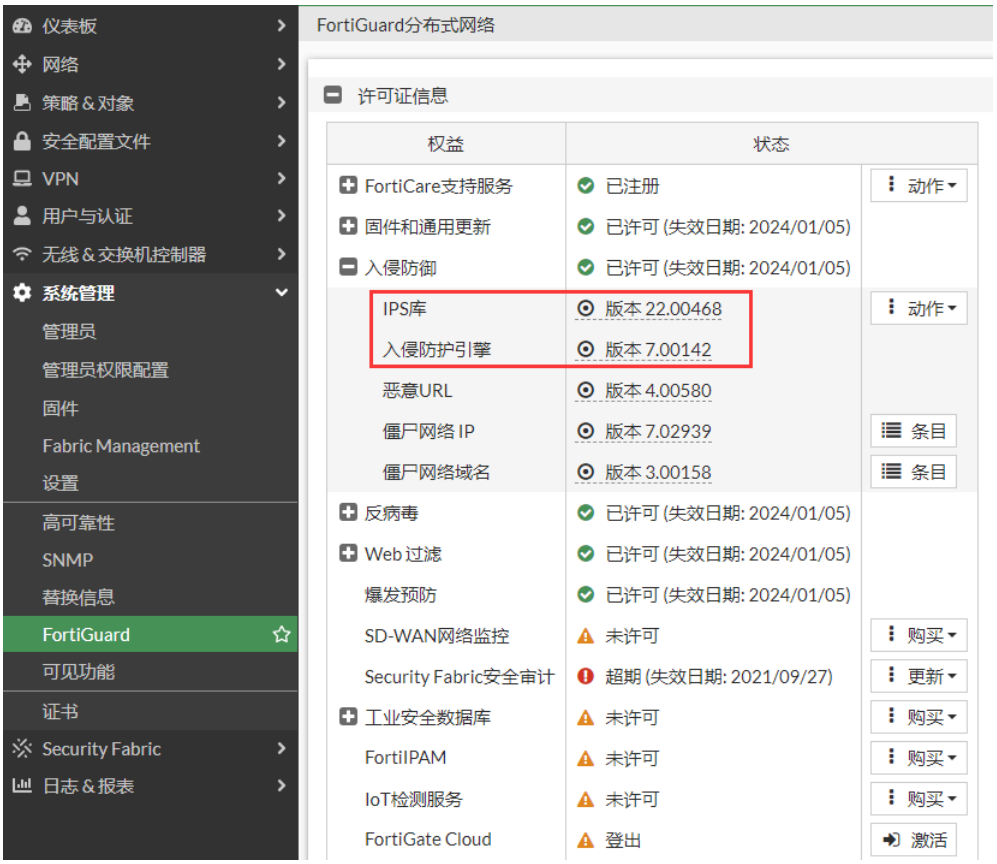
入侵防御简介

什么是 IPS 入侵防御系统

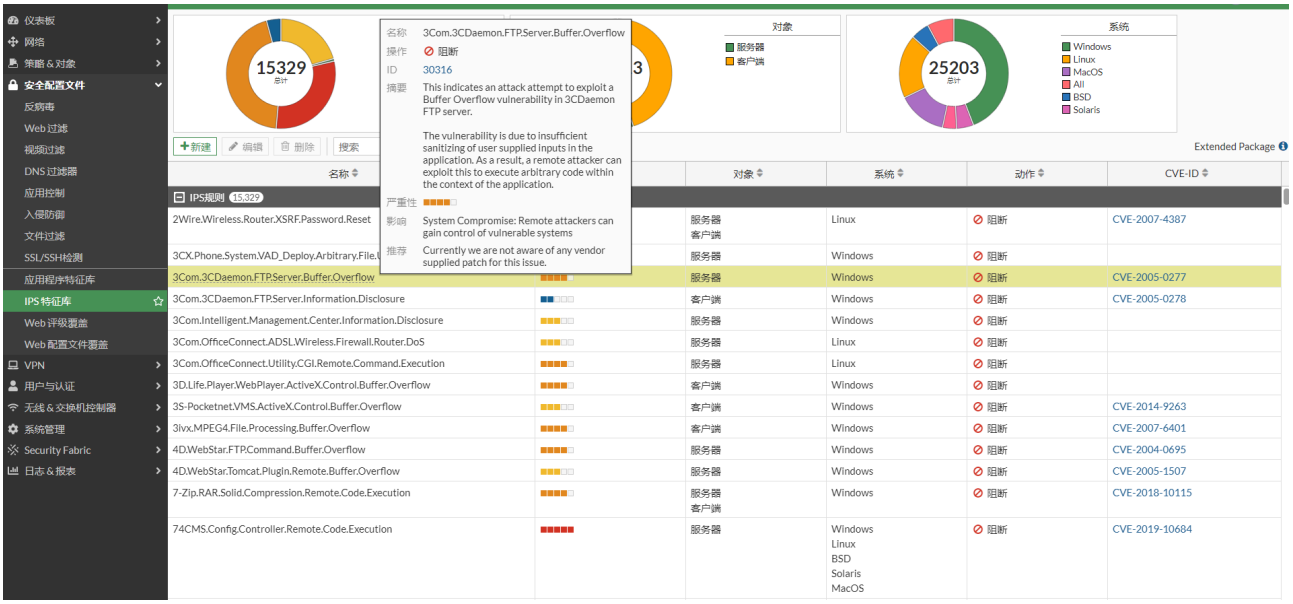
入侵防御系统 (IPS) 可帮助企业识别恶意流量并主动拦截此类流量进入企业网络。IPS 通过特征库检测来识别流量中是否存在漏洞和漏洞利用程序；如果检测到威胁，则按照安全防御策略采取适当的措施，比如阻止访问。

入侵防御特征库

选择“系统管理”-->“FortiGuard”-->“许可证信息”，在“入侵防御”菜单下可以查看 IPS 特种库的版本及 IPS 引擎的版本。

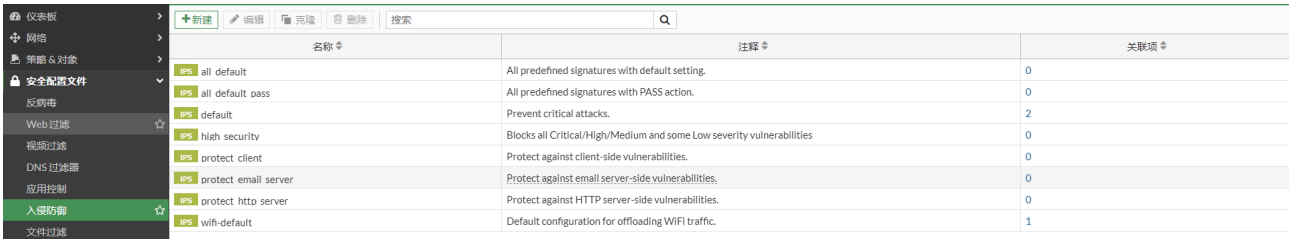


选择“安全配置文件”-->“IPS 特征库”，可以查看具体的特征，将鼠标悬停在某一个特征上，可以查看该特征的详细信息。



入侵防御配置文件

入侵防御默认有一些配置文件，这些配置文件应用的目标各有不同，也可以根据自身的业务需求创建新的配置文件。



在 high_security 配置文件中，IPS 阻断严重级别是 Critical/High/Medium 的特征库，严重级别是 Low 的则是保持特征的默认动作。



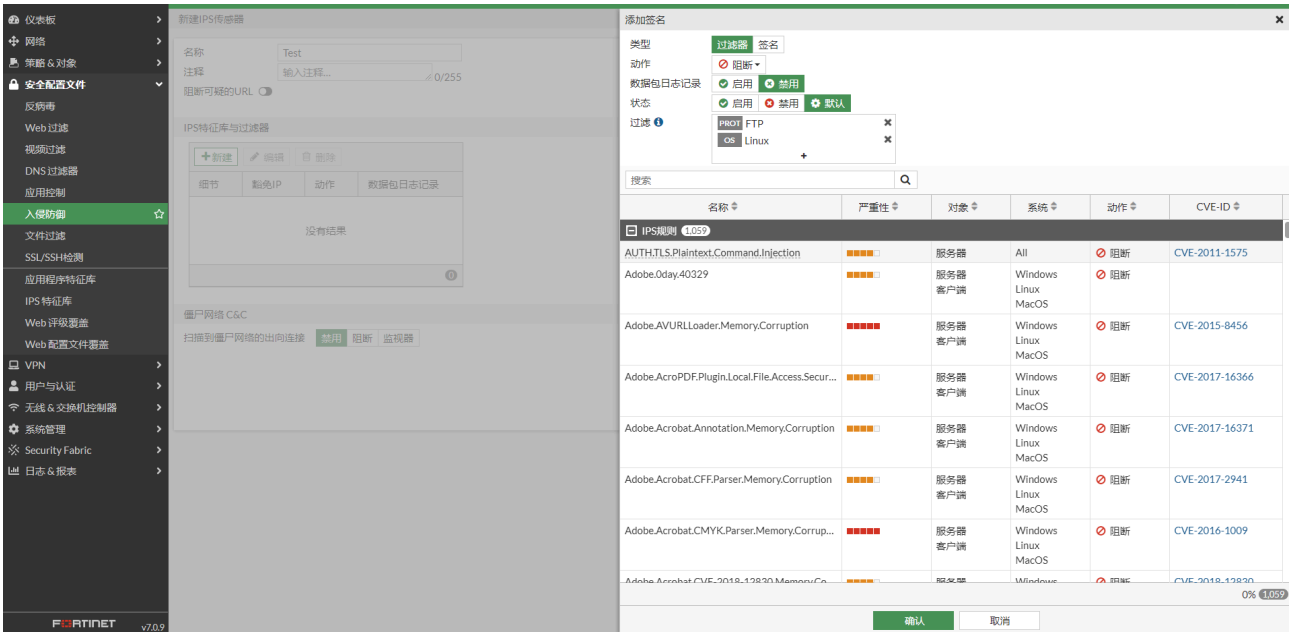
在 protect_http_server 配置文件中，IPS 将对象过滤器的属性设置为服务器，协议过滤器的属性设置为 HTTP，因此过滤器仅包含适用于服务器的 HTTP 协议的签名，动作是使用特征默认的动作。



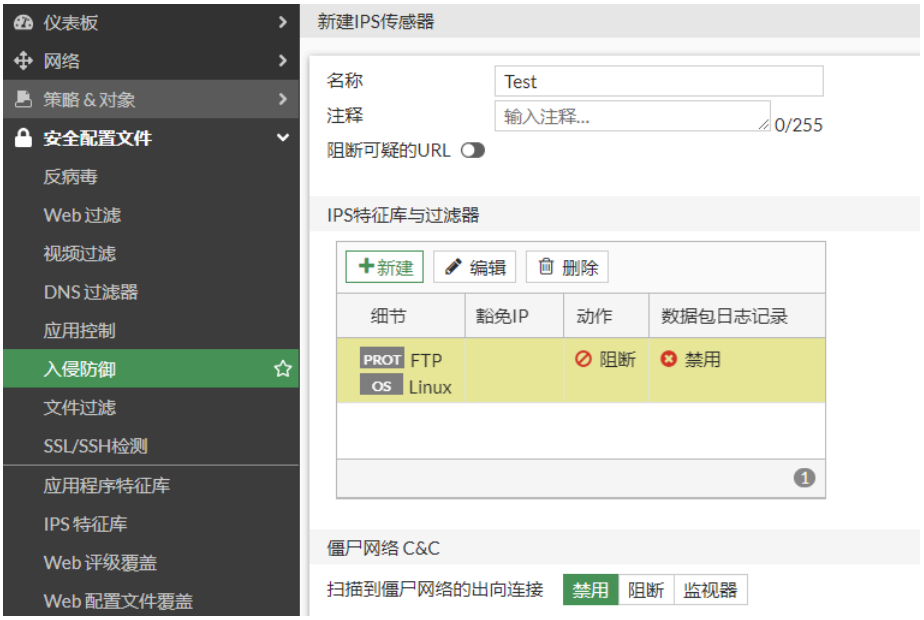
用户也可以根据自身的需求创建配置文件，选择“安全配置文件”-->“入侵防御”，点击“新建”，这里保护 linux 服务器的 HTTP 服务。

1.创建 IPS 过滤器

在” IPS 特征库与过滤器“选项下，点击“新建”，类型选择过滤器，动作设置为阻断，协议过滤器设置为 FTP，OS 过滤器设置为 Linux。

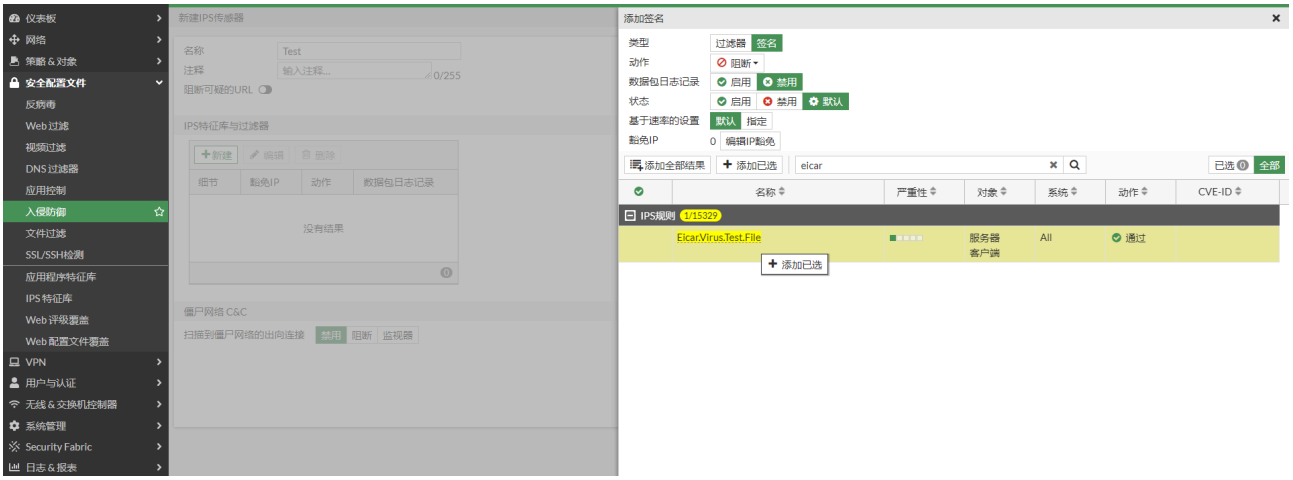


确定保存。

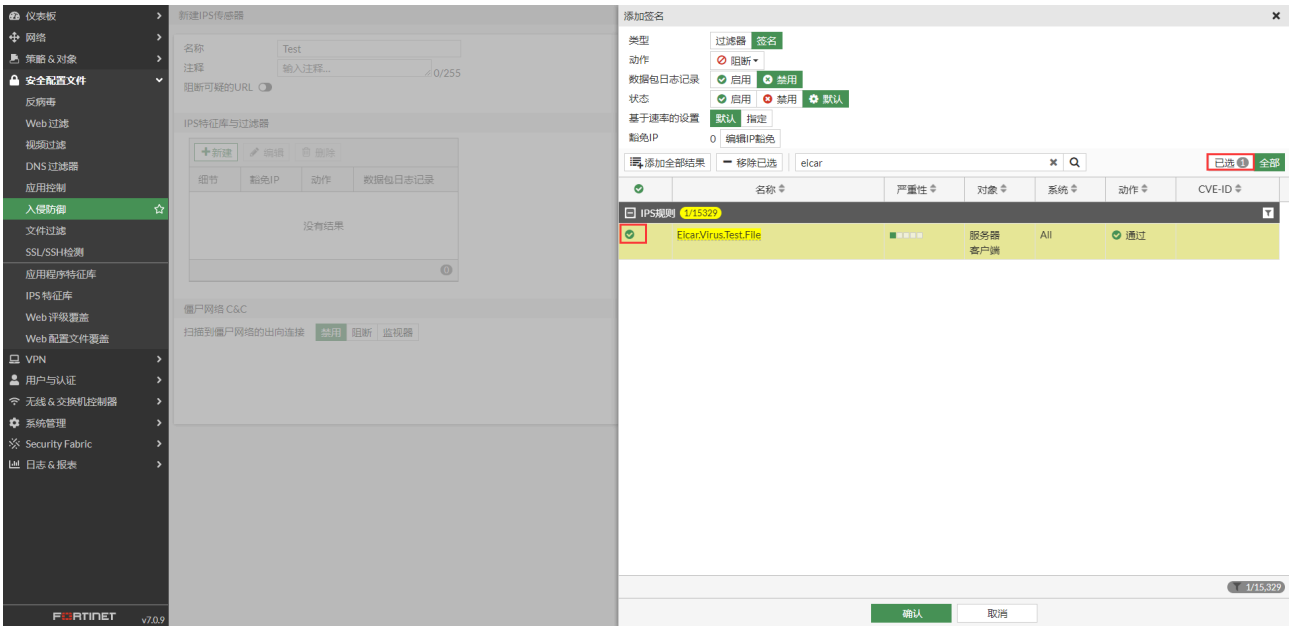


2.指定某一个 IPS 具体的特征

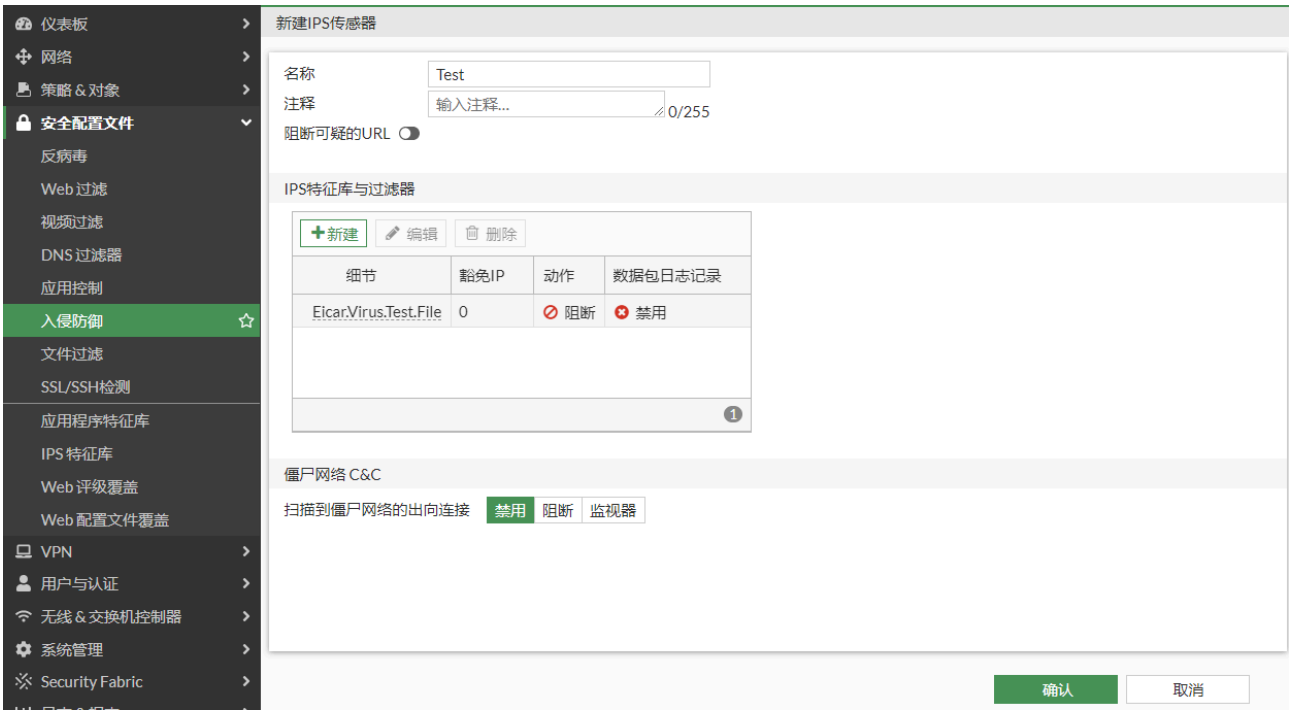
这里以 eicar 做测试。类型选择签名，动作设置为阻断，在搜索框中搜索 eicar，将过滤出 eicar 的特征：Eicar.Virus.Test.File，然后右击该特征，点击“添加已选”，表示选中该特征。



特征选中后，会有绿色勾选的标识，以及“已选”标签会标识出特征的数量，然后点击确认。

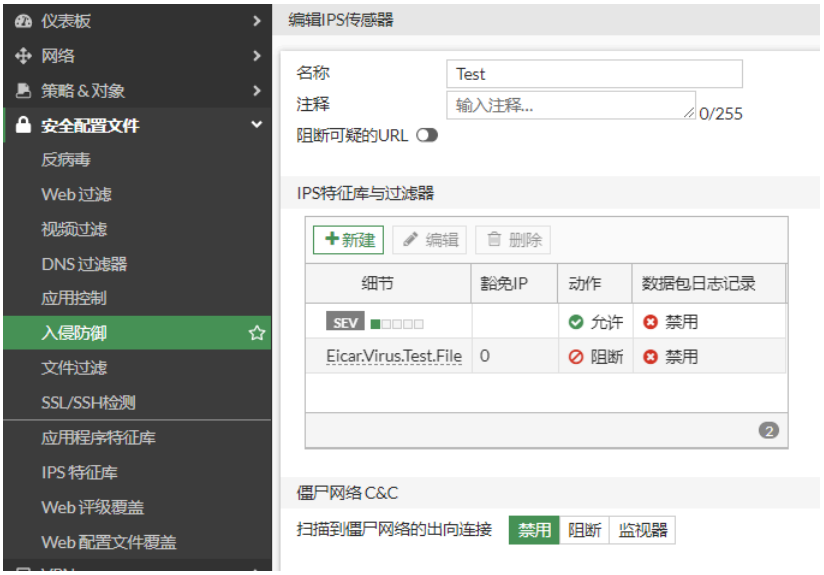


确定保存。



3.IPS 规则的顺序

如下图，严重级别过滤器设置为信息，该过滤器包含了 eicar，动作是允许；单独设置 eicar 特征的动作是阻断；那么最终 eicar 的动作是允许。IPS 规则是由上到下匹配的。

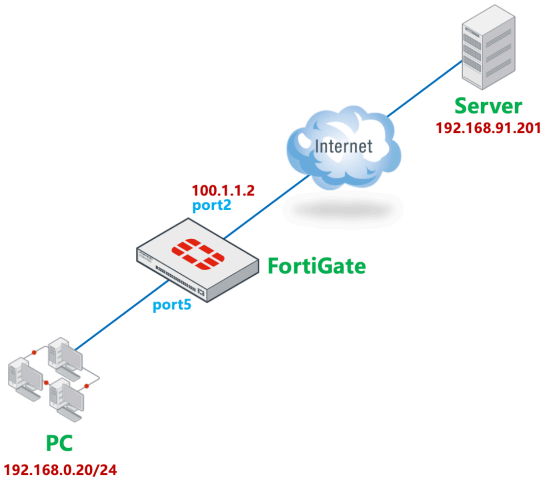


入侵防御配置

网络需求

客户使用 IPS 保护用户上网的流量。

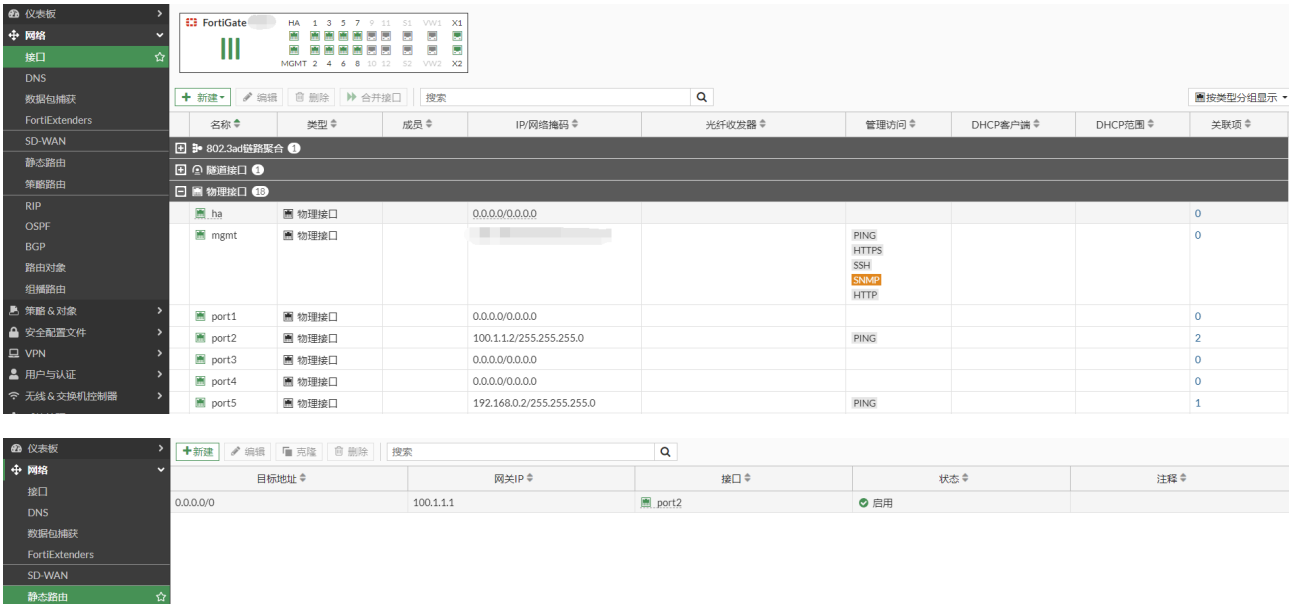
##网络拓扑



配置步骤

1.基本配置

配置接口 IP 和路由。



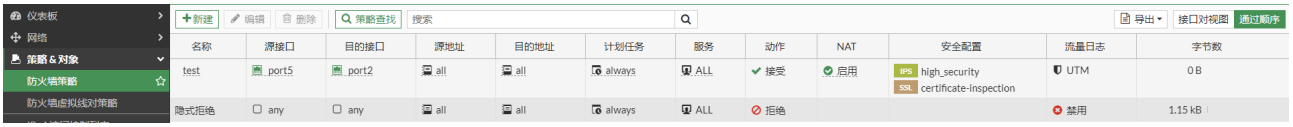
2.配置 IPS 配置文件

这里直接使用默认的 high_security 配置文件。



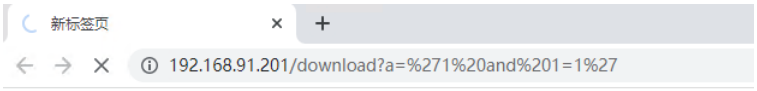
3.配置策略

在策略中调用 IPS 配置文件。

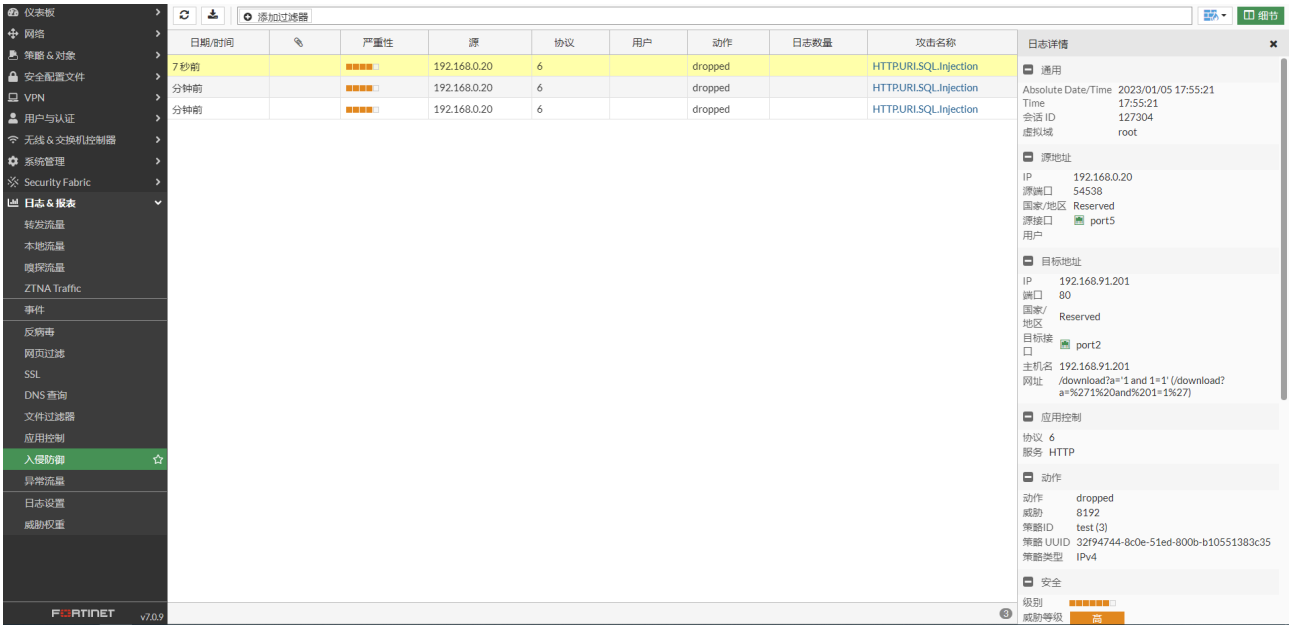


入侵防御测试

发起 SQL 注入攻击被 IPS 拦截。



查看 IPS 日志。



安全

级别

威胁等级 高

威胁分数 30

Cellular

服务 HTTP

入侵防御

配置名称	high_security
攻击名称	HTTP.URI.SQL.Injection
攻击ID	15621
参考	http://www.fortinet.com/ids/VID15621
事件序列号	121635773
方向	outgoing
严重性	
消息	web_misc: HTTP.URI.SQL.Injection

其他

日志事件初始时间戳	1672912521315240700
Timezone	+0800
Log ID	0419016384
类型	utm
子类型	ips
事件类型	signature
源接口角色	undefined
目标接口角色	undefined

虚拟补丁

重要

该功能在 FortiOS 7.2.4 及以后版本支持。

虚拟补丁介绍

虚拟补丁（Virtual Patch）是通过使用 FortiGate 的 IPS Engine 阻止已知漏洞来减轻漏洞攻击的方法。可以用在攻击目标为 FortiGate 自身的情况下，方法是使用 FortiGate 入接口上的 Local-in Policy 将 FMWP（Firmware Virtual Patch）数据库应用到接口上。例如可以使用从 FortiGuard 推送的 FMWP 数据库来减轻针对 GUI 和 SSL VPN 的攻击，从而暂时地修补这些漏洞。

适用场景

适用于 FortiGate 自身存在漏洞，但暂时无法升级固件的情况（比如承载重要业务，短时间内不能升级重启）。当然，虚拟补丁只是权宜之计，最终仍然是推荐升级版本来解决漏洞问题。

实现原理

当在 Local-in Policy 中启用虚拟补丁时，IPS Engine 会查询 FortiGuard 服务器，并执行如下动作：

- 获取针对当前 FortiGate 固件版本的漏洞列表。
- 根据 Local-in 会话中的服务（协议和端口号），确定流向 FortiGate 上本地入接口的会话是否需要扫描。目前，仅可以检查 SSL VPN 和 Web GUI 的本地流量。
- 如果经过检查，匹配 Local-in Policy 的会话所属功能没有包含在该 FortiOS 版本漏洞的功能类型中，则 IPS Engine 会绕过该会话（例如 IPS Engine 从 FortiGuard 获取的当前固件版本的漏洞只有关于 SSL VPN 的漏洞，没有关于 GUI 的漏洞，那么 IPS Engine 会自动 Bypass 所有 GUI 访问的 Local-in 会话流量）。IPS Engine 仅扫描和丢弃可能正在利用漏洞的会话来优化性能。

配置方法

```bash
config firewall local-in-policy
    edit <id>
        set action accept
        set virtual-patch {enable | disable}
    next
end
```

配置步骤

举例 1

虚拟补丁查询 FortiGuard，此 FortiOS 版本只存在 SSL VPN 漏洞。IPS Engine 会丢弃到 FortiGate 本地接口的 SSL VPN 的攻击流量，并绕过 Web GUI 流量。

1. 首先需要 FortiGate 具有 FMWP License。

```bash
FortiGate # diagnose autoupdate versions | grep FMWP -A 6
FMWP Definitions
————————
Version: 24.00013 signed
Contract Expiry Date: Fri Feb  2 2024
Last Updated using scheduled update on Thu Feb  1 02:18:21 2024
Last Update Attempt: Thu Feb  1 16:48:12 2024
Result: No Updates
```

2. 在 FortiGate 的公网接口上，配置 Local-in Policy，动作为 accept，开启虚拟补丁，Service 选择 HTTPS 和 SSL VPN 的端口（Service 请根据您的 FortiGate 配置的 HTTPS 和 SSL VPN 端口来配置）。

```
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "HTTPS" "SSL_VPN_10443"
        set schedule "always"
        set virtual-patch enable
    next
end
```
bash

3. 查看当前虚拟补丁从 FortiGuard 获取的漏洞信息，只包含 SSL VPN 漏洞，不包含 GUI 访问漏洞（7.4 版本的 CLI 为 `diagnose ips vpatch fmwp-status`）。

```
FortiGate # diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 1

  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.
```

4. 针对此漏洞进行攻击，攻击流量被 IPS Engine 拦截，如下为攻击的 IPS 拦截日志，GUI 访问流量被 IPS Engine bypass，不会产生 IPS 日志。

```
1: date=2023-11-07 time=14:53:44 eventtime=1699325624346021995 tz="+1200" logid="0419016384" type="utm" subtype="ips"
eventtype="signature" level="alert" vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root" dstintfrole="undefined"
sessionid=284 action="dropped" proto=6 service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
srcport=53250 dstport=11443 hostname="myfortigate.example" url="/error" httpmethod="POST" direction="outgoing"
attackid=10002887 ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=99614721 msg="vPatch: FortiOS.SSL-
VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"
```
bash

举例 2

虚拟补丁查询 FortiGuard，此 FortiOS 版本只存在 GUI 访问漏洞。IPS Engine 会丢弃到 FortiGate 本地接口的 GUI 访问的攻击流量，并绕过 SSL VPN 流量。

1. 在 FortiGate 的公网接口上，配置 Local-in Policy，动作为 accept，开启虚拟补丁，Service 选择 HTTPS 和 SSL VPN 的端口（Service 请根据您的 FortiGate 配置的 HTTPS 和 SSL VPN 端口来配置）。

```
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "HTTPS" "SSL_VPN_10443"
        set schedule "always"
        set virtual-patch enable
    next
end
```
bash

2. 查看当前虚拟补丁从 FortiGuard 获取的漏洞信息，只包含 SSL VPN 漏洞，不包含 GUI 访问漏洞（7.4 版本的 CLI 为 `diagnose ips vpatch fmwp-status`）。

```
FortiGate # diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 2

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.
```
bash

3. 针对此漏洞进行攻击，攻击流量被 IPS Engine 拦截，如下为攻击的 IPS 拦截日志，SSL VPN 访问流量被 IPS Engine bypass，不会产生 IPS 日志。

```
1: date=2023-11-07 time=14:55:15 eventtime=1699325715311370215 tz="+1200" logid="0419016384" type="utm" subtype="ips"
eventtype="signature" level="alert" vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root" dstintfrole="undefined"
sessionid=304 action="dropped" proto=6 service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=53622 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin" agent="Node.js" httpmethod="GET"
direction="outgoing" attackid=10002156 ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=99614722
msg="vPatch: FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096 crlevel="critical"
```
bash

举例 3

虚拟补丁查询 FortiGuard，此 FortiOS 版本同时存在 GUI 访问漏洞和 SSL VPN 漏洞。IPS Engine 会丢弃到 FortiGate 本地接口的 GUI 访问的攻击流量，并绕过 SSL VPN 流量。

1. 在 FortiGate 的公网接口上，配置 Local-in Policy，动作为 accept，开启虚拟补丁，Service 选择 HTTPS 和 SSL VPN 的端口（Service 请根据您的 FortiGate 配置的 HTTPS 和 SSL VPN 端口来配置）。

```bash
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "HTTPS" "SSL_VPN_10443"
        set schedule "always"
        set virtual-patch enable
    next
end
```

2. 查看当前虚拟补丁从 FortiGuard 获取的漏洞信息，同时包含 SSL VPN 漏洞和 GUI 访问漏洞（7.4 版本的 CLI 为 `diagnose ips vpatch fmwp-status`）。

```bash
FortiGate # diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.
```

3. 针对这些漏洞进行攻击，攻击流量被 IPS Engine 拦截，如下为攻击的 IPS 拦截日志，SSL VPN 和 GUI 访问流量均会产生 IPS 日志。

```bash
1: date=2023-11-07 time=06:42:44 eventtime=1699296164649894963 tz="+1200" logid="0419016384" type="utm" subtype="ips"
eventtype="signature" level="alert" vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root" dstintfrole="undefined"
sessionid=1094 action="dropped" proto=6 service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
srcport=44164 dstport=10443 hostname="myfortigate.example" url="/error" httpmethod="POST" direction="outgoing"
attackid=10002887 ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=116392250 msg="vPatch: FortiOS.SSL-
VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"

2: date=2023-11-07 time=06:42:09 eventtime=1699296129458704870 tz="+1200" logid="0419016384" type="utm" subtype="ips"
eventtype="signature" level="alert" vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root" dstintfrole="undefined"
sessionid=1066 action="dropped" proto=6 service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=42352 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin" agent="Node.js" httpmethod="GET"
direction="outgoing" attackid=10002156 ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=116392236
msg="vPatch: FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096 crlevel="critical"
```

其他 CLI 命令

通过 CLI 查询特定版本的漏洞信息。

```bash
FortiGate # diagnose wad dev-vuln query vendor=fortinet&version=7.2.5&product=fortios
Dev-Vuln Lookup result: success, cache: found, fgd: unknown, item: 0x7fb474e0b4a0
Vulnerability details:
info entry (1):
        'vendor' = fortinet
       'product' = fortios
         'model' = N/A
   'version.min' = 7.2.0
   'version.max' = 7.2.5
      'firmware' = N/A
         'build' = N/A
    'date_added' = 2023-08-22T13:09:11
```

```
  'date_updated' = 2023-08-22T13:09:11
       'sig_id' = 10004065
      'vuln_id' = 918630
     'severity' = 3
...
```

病毒防护

- 反病毒简介
- 反病毒配置
- 基于文件 Hash 的 AV 豁免

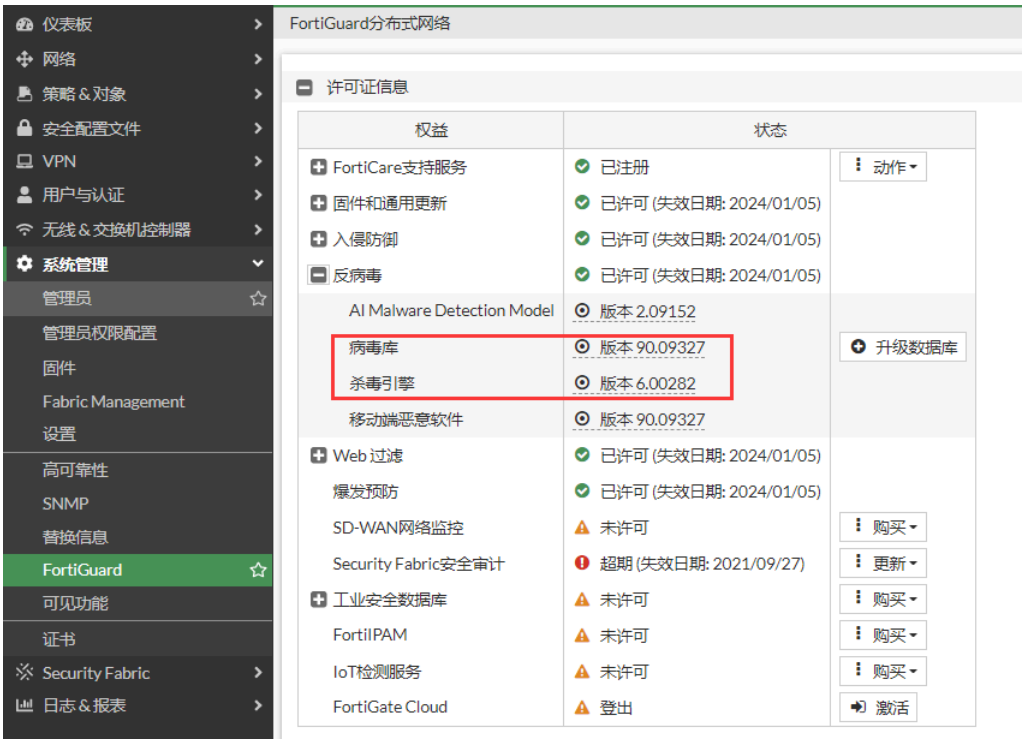
反病毒简介

什么是 AV 反病毒

AV 反病毒功能凭借庞大且不断更新的病毒特征库有效保护企业内网用户和服务器免受病毒文件侵害。AV 通过特征库检测来识别流量中的病毒文件，如果检测到威胁，则按照安全防御策略采取适当的措施，比如阻止访问，避免由病毒文件而引起的数据破坏、权限更改和系统崩溃等情况的发生，保证网络的安全。

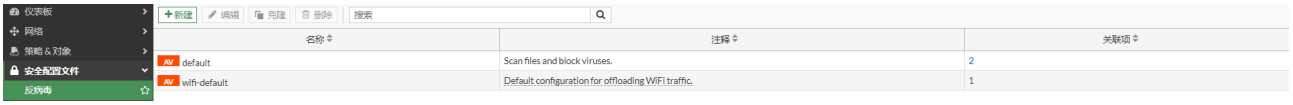
反病毒特征库

选择“系统管理”-->“FortiGuard”-->“许可证信息”，在“反病毒菜单”下可以查看 AV 特种库的版本及 AV 引擎的版本。



反病毒配置文件

反病毒默认自带两个配置文件，这两个配置文件设置是一样，只是描述性信息不一样。



默认使用 default 配置文件即可，动作默认是阻断，可根据实际情况调整。



用户也可以根据自身的需求创建配置文件，选择“安全配置文件”-->“反病毒”，点击“新建”，这里保护 HTTP 服务。

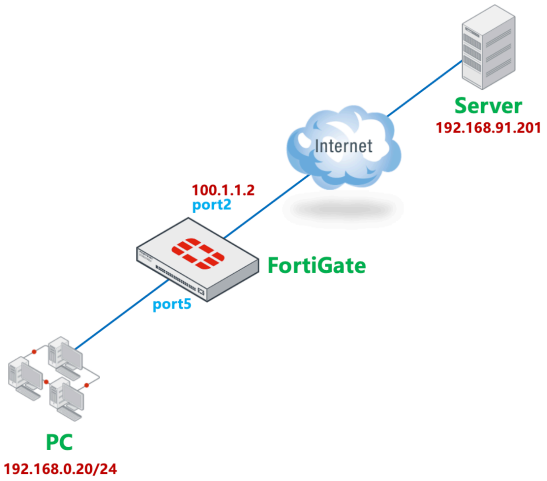


反病毒配置

网络需求

客户使用 AV 保护用户上网的流量。

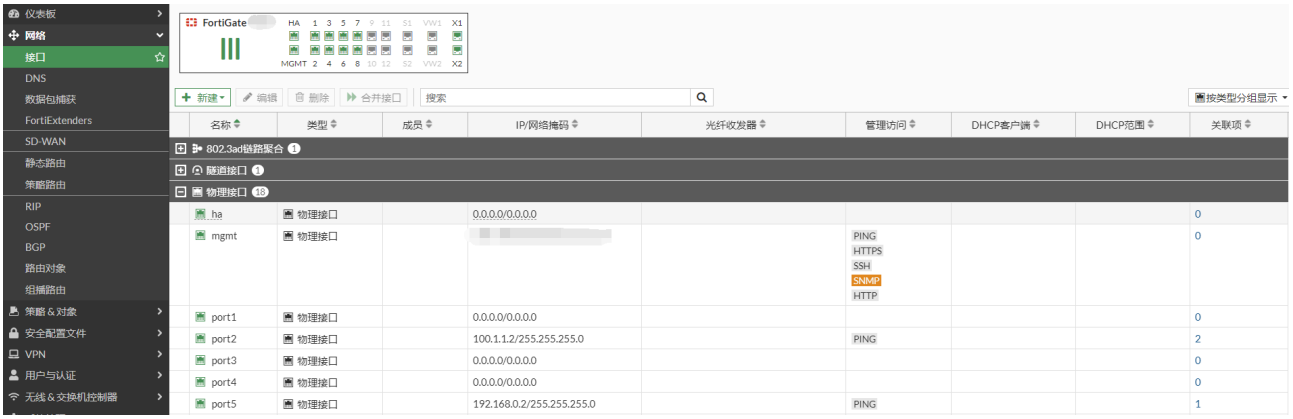
网络拓扑



配置步骤

1.基本配置

配置接口 IP 和路由。





2.配置 IPS 配置文件

这里使用默认的 default 配置文件。测试环境中使用 eicar 做测试。



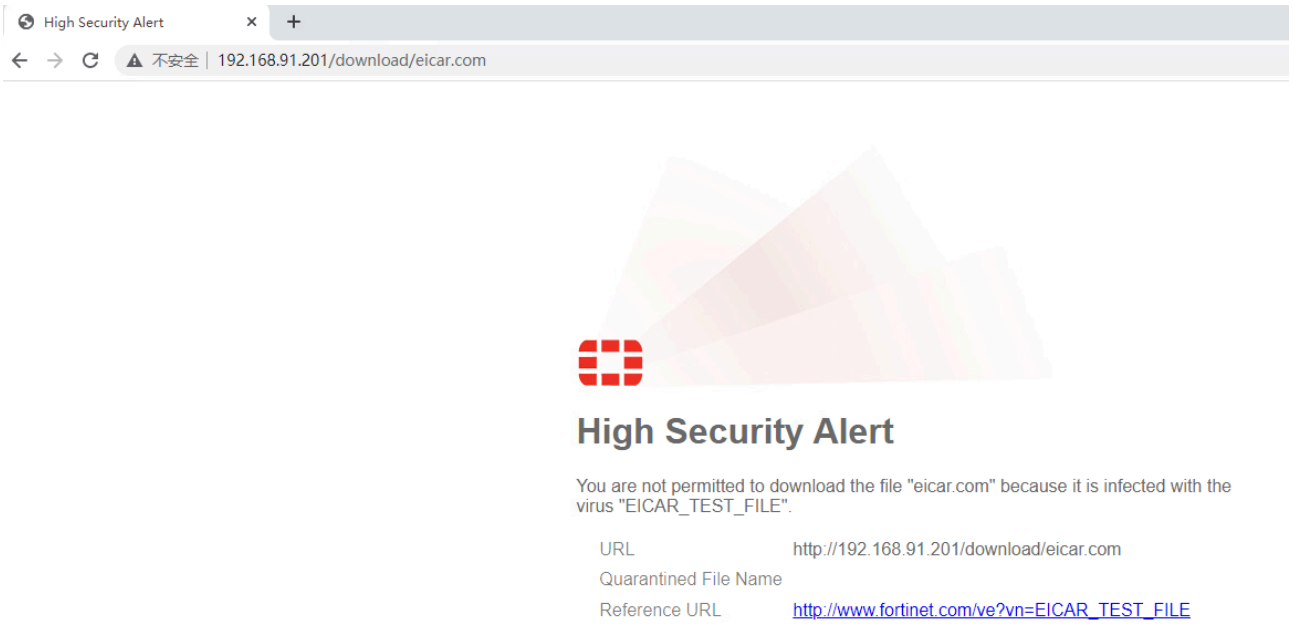
3.配置策略



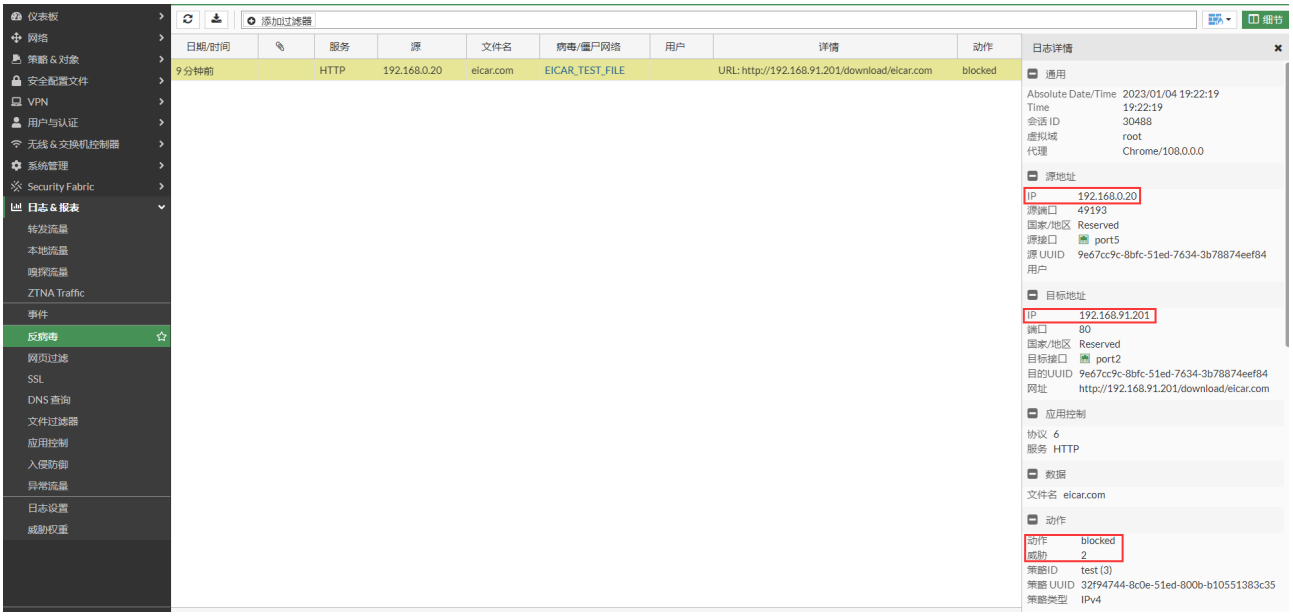
反病毒测试

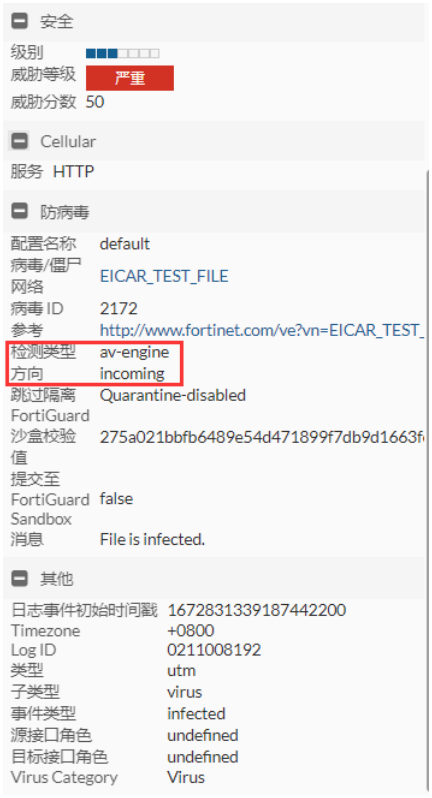
1.使用 PC 下载 eicar.com ↗ 文件

PC 下载 eicar.com ↗ 时，被 AV 检测到并拦截。



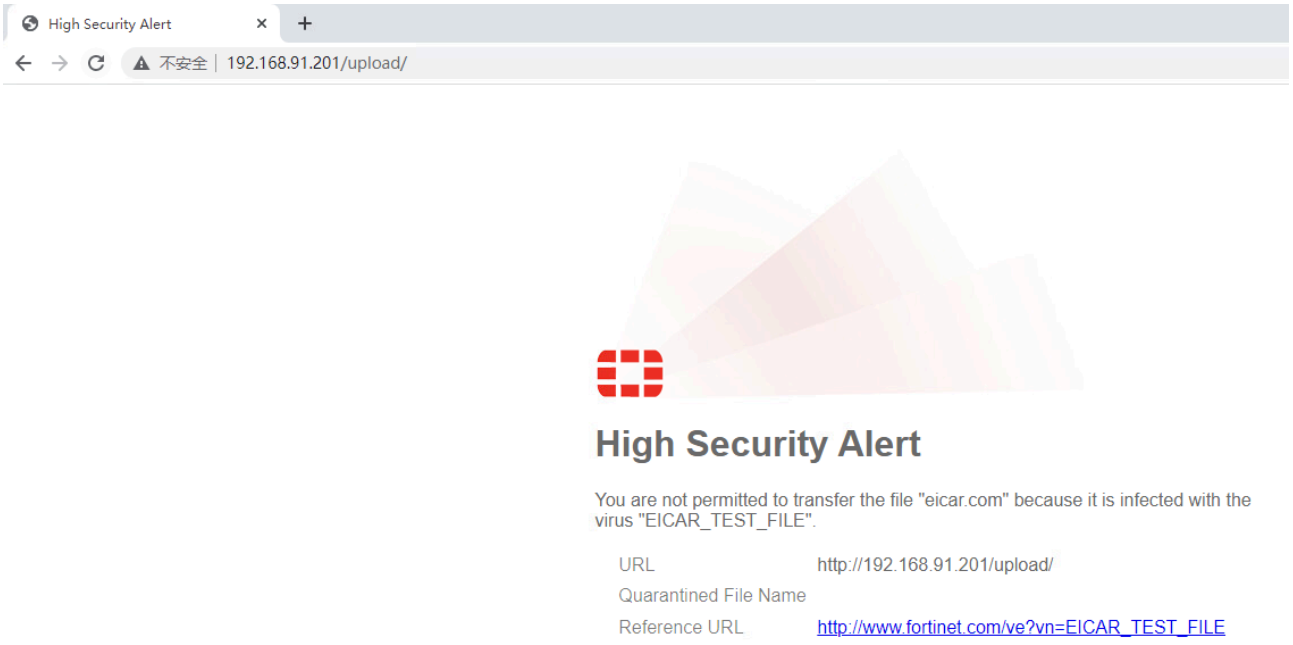
查看反病毒日志。对于发起方 192.168.0.20 而言，攻击是从 192.168.91.201 到 192.168.0.20，因此方向是 incoming。



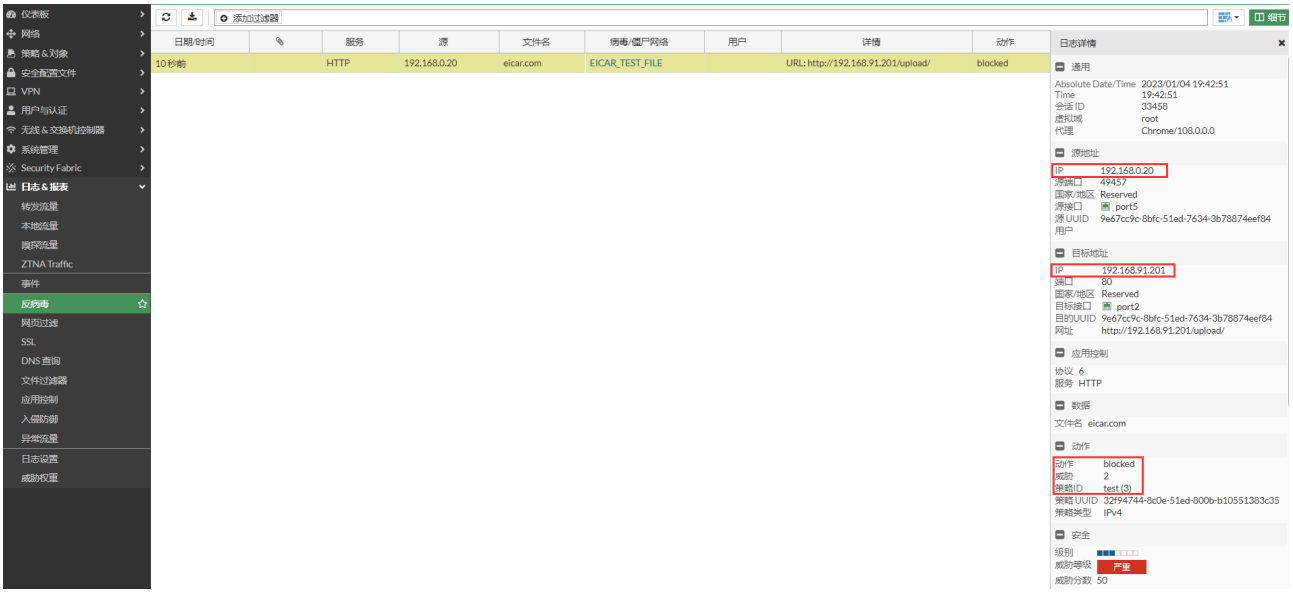


2.**使用 PC 上传 eicar.com** ↗

PC 上传 eicar.com ↗ 时，被 AV 检测到并拦截。



查看反病毒日志。对于发起方 192.168.0.20 而言，攻击是从 192.168.0.20 到 192.168.91.201，因此方向是 outgoing。



Cellular

服务	HTTP

防病毒

配置名称	default
病毒/僵尸网络	EICAR_TEST_FILE
病毒 ID	2172
参考	http://www.fortinet.com/ve?vn=EICAR_TEST_
检测类型	av-engine
方向	outgoing
跳过隔离	Quarantine-disabled
FortiGuard 沙盒校验值	275a021bbfb6489e54d471899f7db9d1663f
提交至 FortiGuard Sandbox	false
消息	File is infected.

其他

日志事件初始时间戳	1672832572227783400
Timezone	+0800
Log ID	0211008192
类型	utm
子类型	virus
事件类型	infected
源接口角色	undefined
目标接口角色	undefined
Virus Category	Virus

基于文件 Hash 的 AV 豁免

> **重要**
>
> 该功能在 FortiOS 7.2.4 及以上版本支持。

网络需求

在反病毒功能的使用中，如果有安全的文件被反病毒功能误判并阻止，用户需要向 FortiGuard 提交误报信息，然后等待官方确认和修复，需要等待一定的时间。需要可以对误判文件进行豁免操作，以允许即时传输该文件。

功能介绍

FortiGate 允许对反病毒误判的文件进行豁免。用户可以指定文件的 MD5、SHA1 或 SHA256 哈希值进行匹配，当匹配 FortiGate AV 的白名单时，该文件会被 Bypass。

> **重要**
>
> 豁免列表功能不适用于 Outbreak Prevention、Machine-Learning、FortiNDR 和 FortiSandbox inline scans。

配置方法

此功能仅支持在 CLI 下配置。

```bash
config antivirus exempt-list
    edit <name>
        set hash-type {md5 | sha1 | sha256}
        set hash <string>
        set status {enable | disable}
    next
end
```

配置步骤与配置验证

1. 获取被误判文件的 Hash 值，这里以一个 Eicar 文件 `csm-eicar.com` 为例，该文件的 MD5 值为 `44d88612fea8a8f36de82e1278abb02f`，该文件位于 `www.csm-testcenter.org` 网站。
2. 按需在“安全配置文件 → 反病毒”页面中配置反病毒配置文件，这里以“default”配置文件为例。

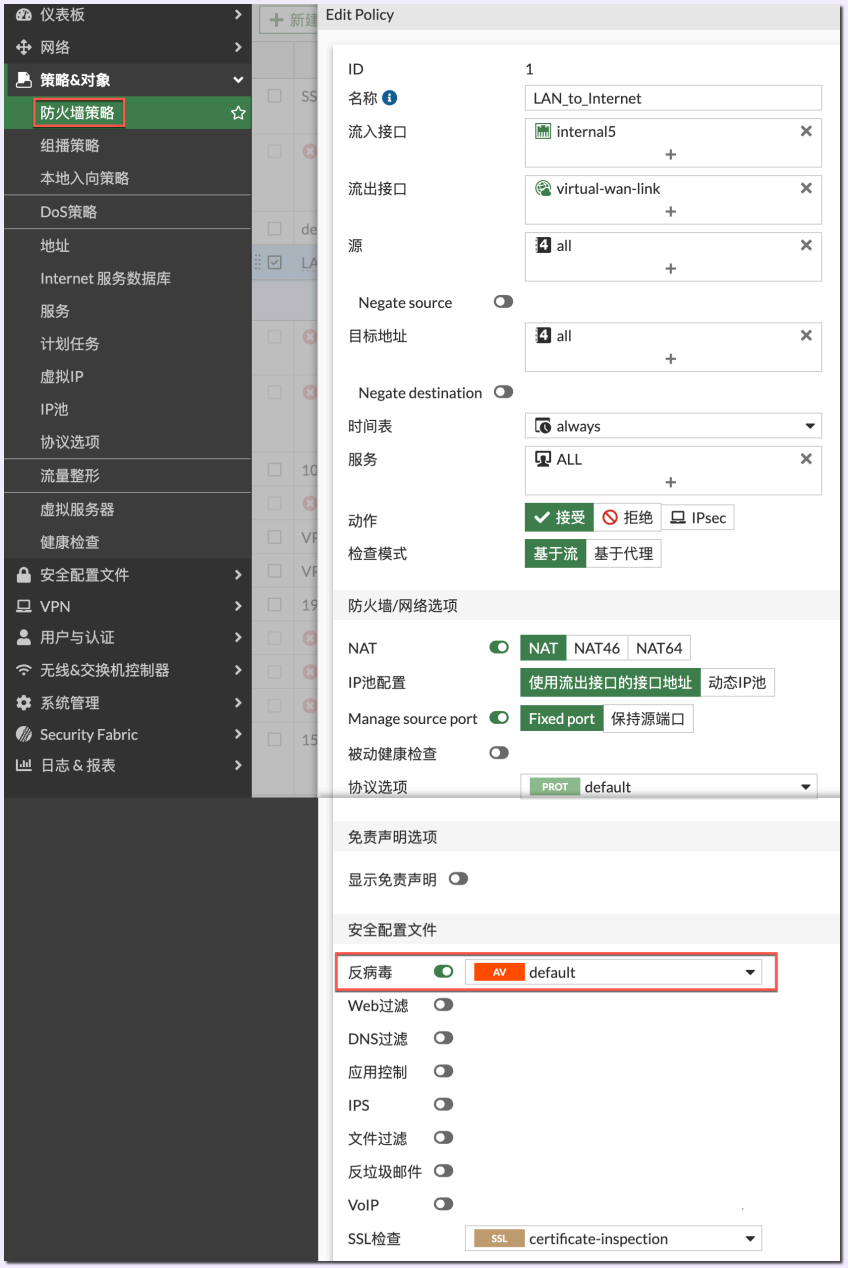


3. 在防火墙策略中引用该反病毒配置文件。

重要

- FortiOS 7.2.X 版本需要防火墙策略的“检测模式”使用 `proxy`（基于代理）模式才能生效。
- FortiOS 7.4.X 版本的防火墙策略“检测模式”没有要求。

:::

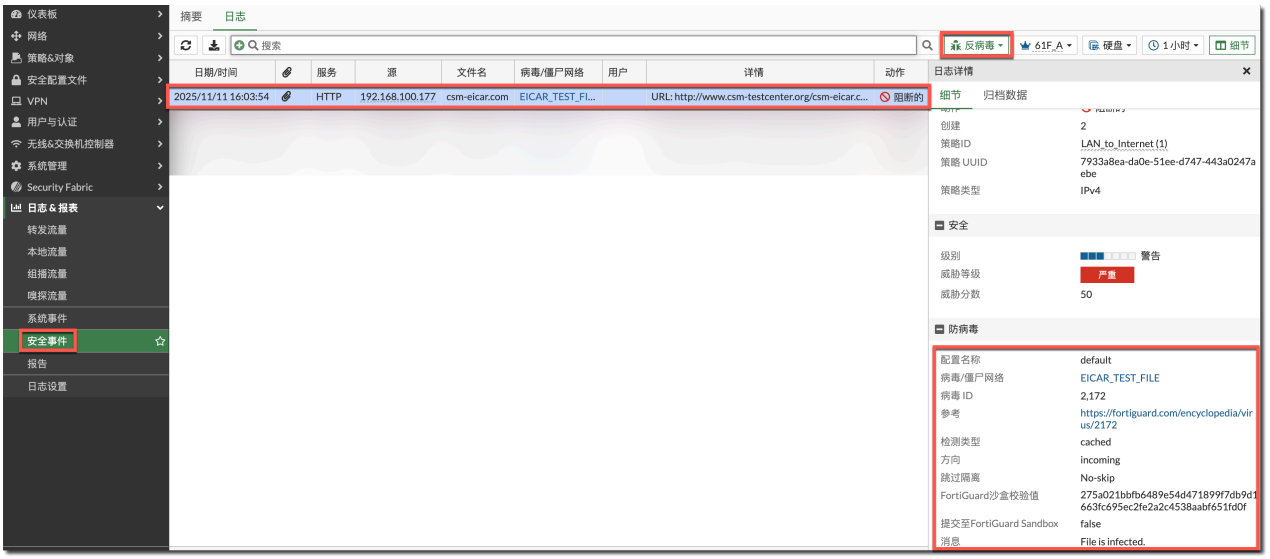



```bash
config firewall policy
    edit 1
        set name "LAN_to_Internet"
        set srcintf "internal5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set nat enable
    next
end
```

4. 此时使用客户端 PC 使用 HTTP 穿过 FortiGate 下载该文件，下载失败。

```bash
root@Ubuntu # wget http://www.csm-testcenter.org/csm-eicar.com
--2025-11-11 16:03:51--  http://www.csm-testcenter.org/csm-eicar.com
正在解析主机 www.csm-testcenter.org (www.csm-testcenter.org)... 85.215.35.144
正在连接 www.csm-testcenter.org (www.csm-testcenter.org)|85.215.35.144|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：68 [application/octet-stream]
正在保存至："csm-eicar.com.1"
csm-eicar.com.1                0%[                                                 ]       0  --.-KB/s  用时 0s
2025-11-11 16:03:52 (0.00 B/s) - 在 0/68 字节处发生读取错误 (Connection reset by peer)。重试中。
--2025-11-11 16:03:53--  (尝试次数： 2)  http://www.csm-testcenter.org/csm-eicar.com
正在连接 www.csm-testcenter.org (www.csm-testcenter.org)|85.215.35.144|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 403 Forbidden
2025-11-11 16:03:53 错误 403：Forbidden。
```

5. 该文件下载时被 FortiGate 的反病毒模块拦截，如下所示 AV 拦截日志。




```bash
date=2025-11-11 time=16:03:54 eventtime=1762848233896917415 tz="+0800" logid="0211008192" type="utm" subtype="virus"
eventtype="infected" level="warning" vd="root" policyid=1 poluuid="7933a8ea-da0e-51ee-d747-443a0247aebe"
policytype="policy" msg="File is infected." action="blocked" service="HTTP" sessionid=48151480 srcip=192.168.100.177
dstip=85.215.35.144 srcport=60122 dstport=80 srccountry="Reserved" dstcountry="Germany" srcintf="internal5"
srcintfrole="undefined" dstintf="wan1" dstintfrole="undefined" srcuuid="477325da-a1dc-51ed-ecdd-9add8d1533e2"
dstuuid="477325da-a1dc-51ed-ecdd-9add8d1533e2" proto=6 direction="incoming" filename="csm-eicar.com" checksum="6851cf3c"
quarskip="No-skip" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="cached"
ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172 url="http://www.csm-testcenter.org/csm-eicar.com"
profile="default" agent="Wget/1.21.4" httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f" analyticssubmit="false" crscore=50
craction=2 crlevel="critical"
```

6. 在 CLI 下配置反病毒豁免列表，Hash 类别为 MD5，将该文件的 MD5 值填入其中。

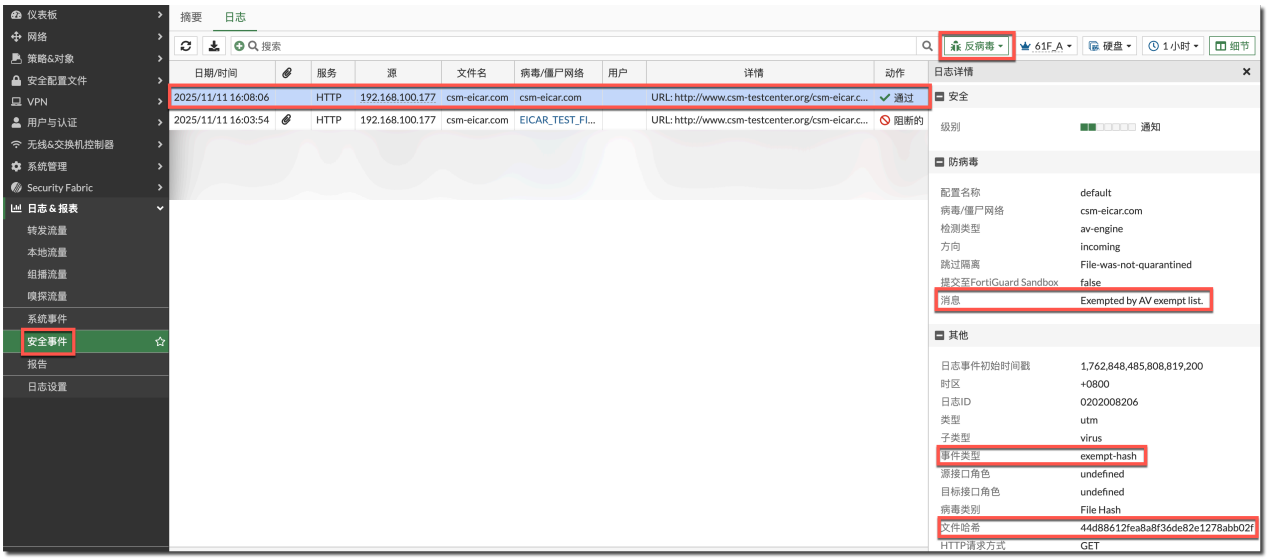
```bash
config antivirus exempt-list
    edit "csm-eicar.com"
        set hash-type md5
        set hash "44d88612fea8a8f36de82e1278abb02f"
    next
end
```

7. 再次使用客户端 PC 通过 FortiGate 下载该文件，下载成功。

```bash
root@Ubuntu # wget http://www.csm-testcenter.org/csm-eicar.com
--2025-11-11 16:08:04--  http://www.csm-testcenter.org/csm-eicar.com
正在解析主机 www.csm-testcenter.org (www.csm-testcenter.org)... 85.215.35.144
正在连接 www.csm-testcenter.org (www.csm-testcenter.org)|85.215.35.144|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：68 [application/octet-stream]
```

```
正在保存至: “csm-eicar.com.2”

csm-eicar.com.2                100%[═══════════════════════════════════════════════⇒]      68   331 B/s  用时 0.2s

2025-11-11 16:08:05 (331 B/s) - 已保存 “csm-eicar.com.2” [68/68])
```

8. 该文件下载时被 FortiGate 的反病毒模块放通，如下所示 AV 拦截日志，日志中提示被该文件“Exempted by AV exempt list”。




```
date=2025-11-11 time=16:08:06 eventtime=1762848485808819095 tz="+0800" logid="0202008206" type="utm" subtype="virus"   bash
eventtype="exempt-hash" level="notice" vd="root" policyid=1 poluuid="7933a8ea-da0e-51ee-d747-443a0247aebe"
policytype="policy" msg="Exempted by AV exempt list." action="passthrough" service="HTTP" sessionid=48156818
srcip=192.168.100.177 dstip=85.215.35.144 srcport=62065 dstport=80 srccountry="Reserved" dstcountry="Germany"
srcintf="internal5" srcintfrole="undefined" dstintf="wan1" dstintfrole="undefined" srcuuid="477325da-a1dc-51ed-ecdd-
9add8d1533e2" dstuuid="477325da-a1dc-51ed-ecdd-9add8d1533e2" proto=6 direction="incoming" filename="csm-eicar.com"
quarskip="File-was-not-quarantined" virus="csm-eicar.com" viruscat="File Hash" dtype="av-engine"
filehash="44d88612fea8a8f36de82e1278abb02f" url="http://www.csm-testcenter.org/csm-eicar.com" profile="default"
agent="Wget/1.21.4" httpmethod="GET" analyticssubmit="false"
```

Web 过滤功能

Web 过滤常见问题

- Chrome/Edge 访问网页无法被拦截

Web 过滤简介

Web 过滤配置

URL 过滤

URL 过滤表达式

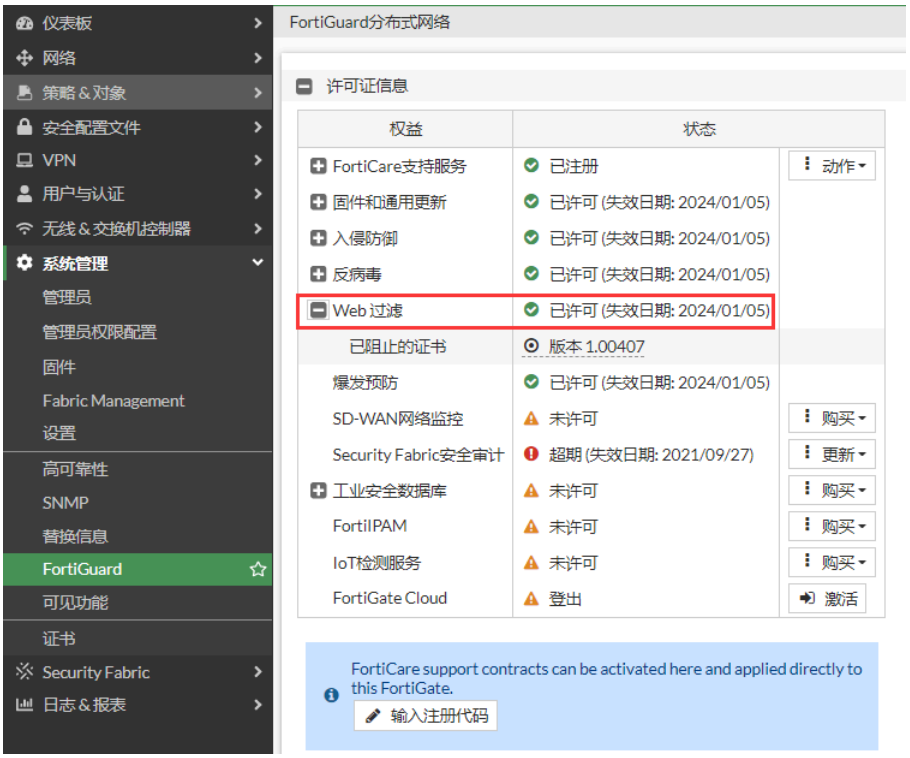
Web 过滤简介

什么是 Web 过滤

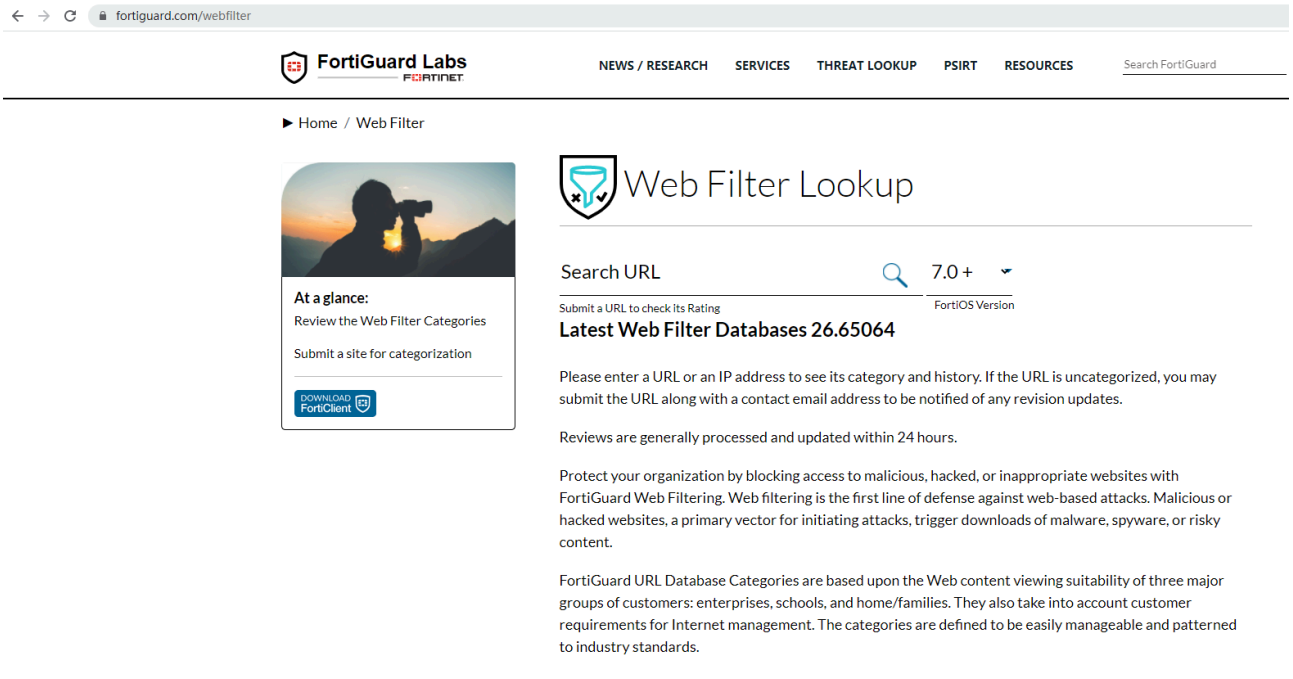
Web 过滤是 FortiGate 对用户访问 Web 资源进行限制或控制的功能。

Web 过滤特征库

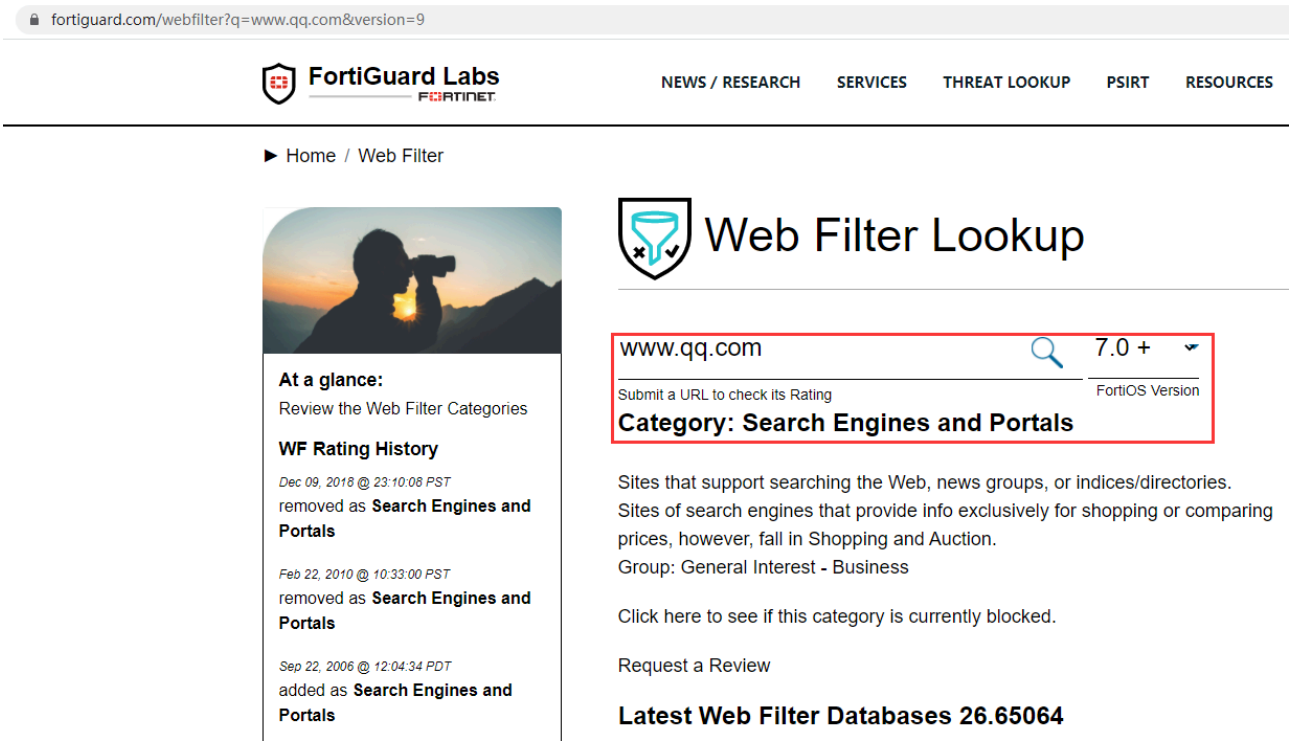
选择“系统管理”-->“FortiGuard”-->“许可证信息”，查看 Web 过滤的状态是有 license 的。Web 过滤的特征库在云端，不在 FortiGate 本地。



在 FortiGuard 上查看 Webfilter 特征库。

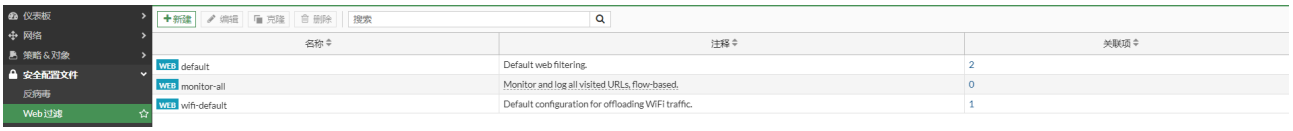


输入网站，可以查询该 URL 属于哪个分类。



web 过滤配置文件

web 过滤默认有一些配置文件，这些配置文件应用的目标各有不同，也可以根据自身的业务需求创建新的配置文件。



default 配置文件默认对成人内容，不安全的内容等进行阻断，对其他的内容进行监控。



monitor-all 配置文件对所有的内容都进行监控。



用户也可以根据自身的需求创建配置文件，选择“安全配置文件”-->“Web 过滤”，点击“新建”。如这里通过“URL 过滤器”只允许用户访问 support.fortinet.com.cn ↗，其他网站都拒绝。



URL 过滤与 WebFilter 的执行顺序

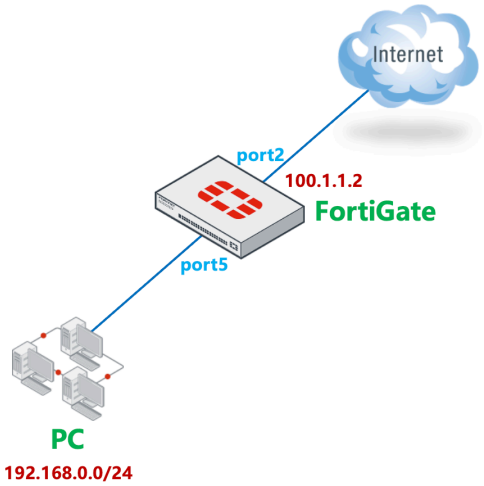
1. 如果 URL 过滤的动作是允许或监视器，那么将继续执行 WebFilter 分类过滤（如果 WebFilter 通过查询 FortiGuard 的分类的匹配动作是拒绝，那么该 URL 仍然会被拒绝）。
2. 如果 URL 过滤的动作是拒绝，那么该 URL 将被直接拒绝。
3. 如果 URL 过滤的动作是例外，那么该 URL 直接放通，不会执行 WebFilter 分类过滤。

Web 过滤配置

网络需求

客户使用 Webfilter 限制用户访问 Web 资源。

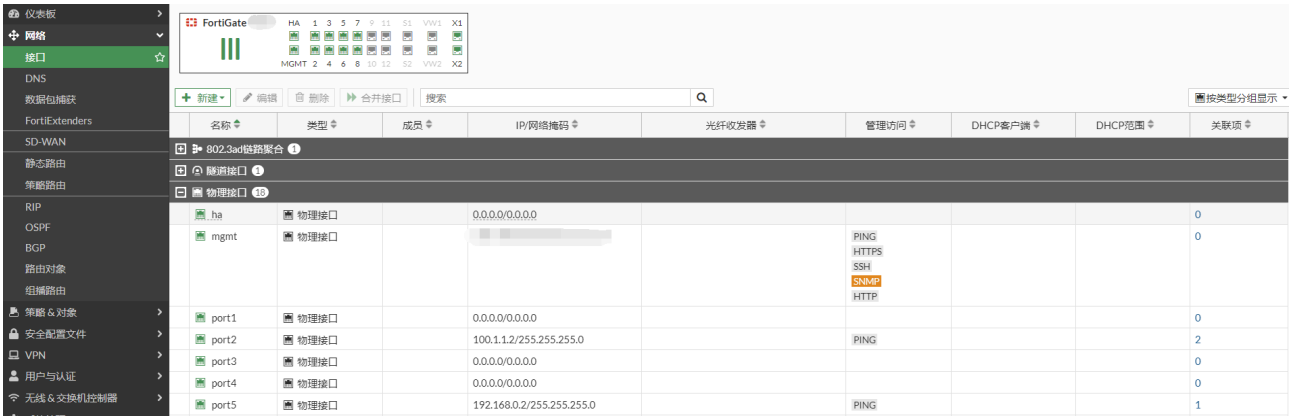
网络拓扑



配置步骤

1.基本配置

配置接口 IP 和路由。





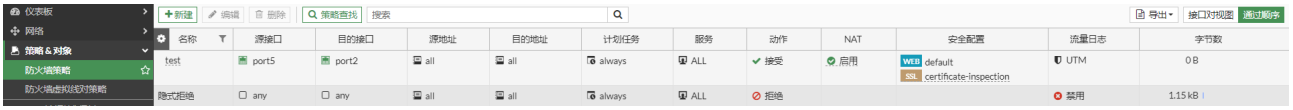
2.配置 Web 过滤配置文件

这里直接使用 Default 配置文件，为了便于测试，这里阻断了用户访问”搜索引擎和门户网站“的 URL 分类。

请勾选”当发生分级错误时允许的网站“。这是为了防止因网络延迟，丢包等原因导致 FortiGate 不能与 FortiGuard webfilter 通信而引起 web 站点无法评估，进而导致该站点不能访问；勾选后如果是由于无法评估引起的错误，将允许访问该站点。

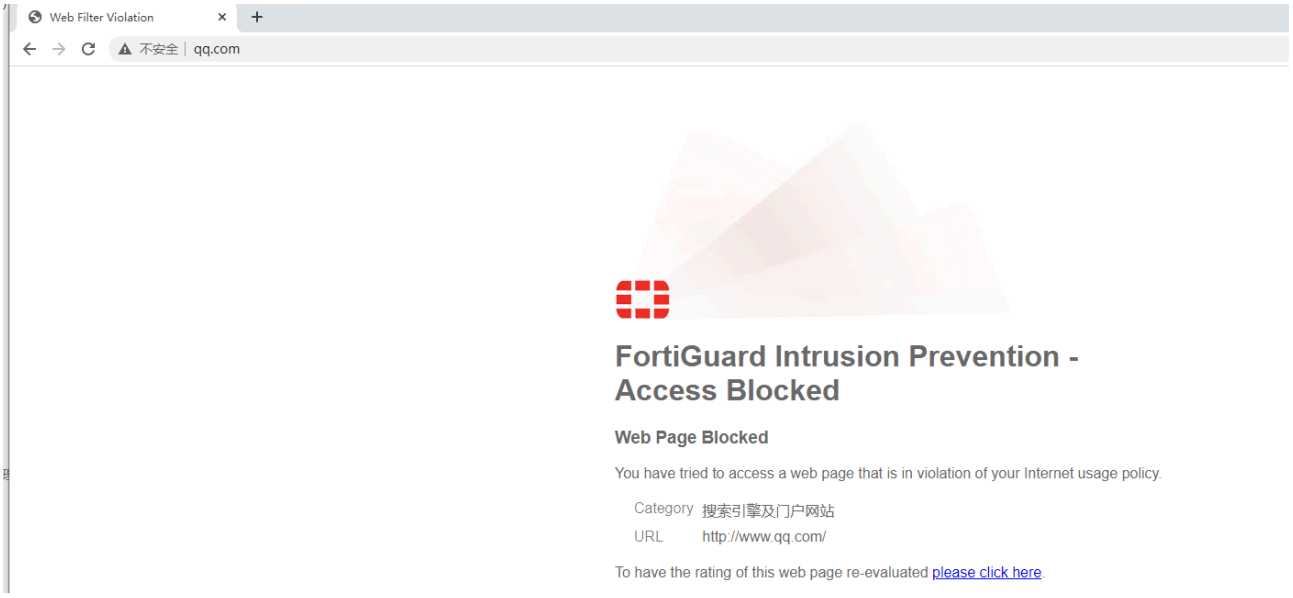


3.配置策略

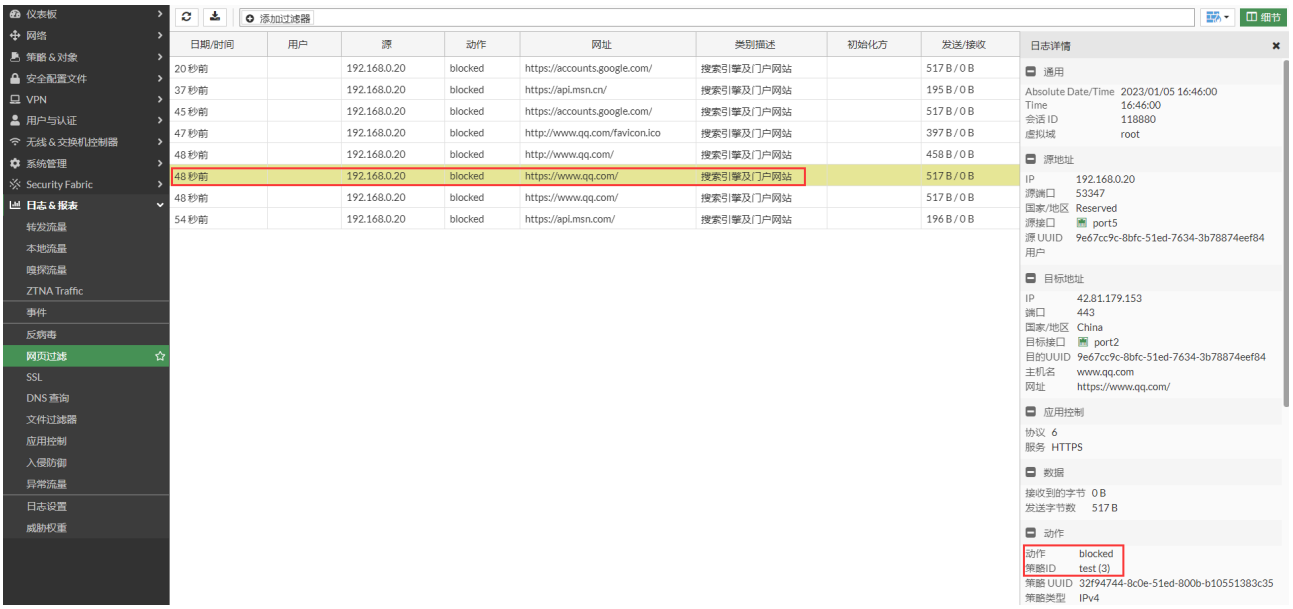


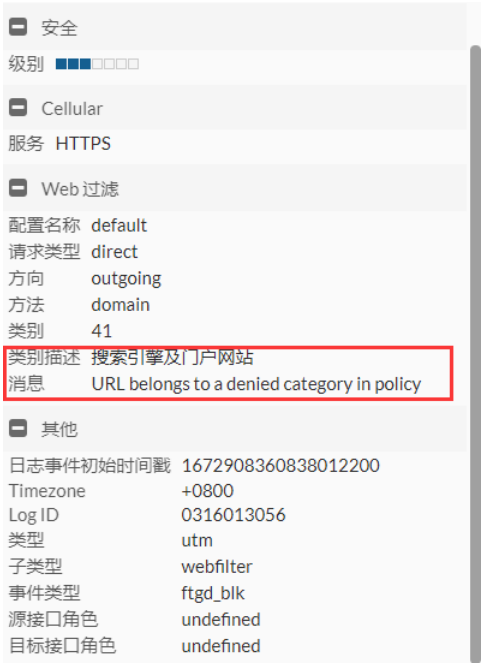
Web 过滤测试

使用 PC 访问 www.qq.com ↗ 被 web 过滤阻断。


Web Filter Violation
不安全 | qq.com
FortiGuard Intrusion Prevention - Access Blocked
Web Page Blocked
You have tried to access a web page that is in violation of your Internet usage policy.
Category 搜索引擎及门户网站
URL http://www.qq.com/
To have the rating of this web page re-evaluated please click here.

查看 Web 过滤日志


仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
转发流量
本地流量
嗅探流量
ZTNA Traffic
事件
反病毒
网页过滤
SSL
DNS 查询
文件过滤器
应用控制
入侵防御
异常流量
日志设置
威胁权重
添加过滤器
细节
日期/时间 用户 源 动作 网址 类别描述 初始化方 发送/接收
20 秒前 192.168.0.20 blocked https://accounts.google.com/ 搜索引擎及门户网站 517 B / 0 B
37 秒前 192.168.0.20 blocked https://api.msn.cn/ 搜索引擎及门户网站 195 B / 0 B
45 秒前 192.168.0.20 blocked https://accounts.google.com/ 搜索引擎及门户网站 517 B / 0 B
47 秒前 192.168.0.20 blocked http://www.qq.com/favicon.ico 搜索引擎及门户网站 397 B / 0 B
48 秒前 192.168.0.20 blocked http://www.qq.com/ 搜索引擎及门户网站 458 B / 0 B
48 秒前 192.168.0.20 blocked https://www.qq.com/ 搜索引擎及门户网站 517 B / 0 B
48 秒前 192.168.0.20 blocked https://www.qq.com/ 搜索引擎及门户网站 517 B / 0 B
54 秒前 192.168.0.20 blocked https://api.msn.com/ 搜索引擎及门户网站 196 B / 0 B
日志详情
通用
Absolute Date/Time 2023/01/05 16:46:00
Time 16:46:00
会话 ID 118880
虚拟域 root
源地址
IP 192.168.0.20
源端口 53347
国家/地区 Reserved
源接口 port5
源 UUID 9e67cc9c-8bfc-51ed-7634-3b78874eef84
用户
目标地址
IP 42.81.179.153
端口 443
国家/地区 China
目标接口 port2
目的UUID 9e67cc9c-8bfc-51ed-7634-3b78874eef84
主机名 www.qq.com
网址 https://www.qq.com/
应用控制
协议 6
服务 HTTPS
数据
接收到的字节 0 B
发送字节数 517 B
动作
动作 blocked
策略ID test (3)
策略 UUID 32f94744-8c0e-51ed-800b-b10551383c35
策略类型 IPv4


安全
级别
Cellular
服务 HTTPS
Web 过滤
配置名称 default
请求类型 direct
方向 outgoing
方法 domain
类别 41
类别描述 搜索引擎及门户网站
消息 URL belongs to a denied category in policy
其他
日志事件初始时间戳 1672908360838012200
Timezone +0800
Log ID 0316013056
类型 utm
子类型 webfilter
事件类型 ftgd_blk
源接口角色 undefined
目标接口角色 undefined

URL 过滤

什么是 URL 过滤

在 Web 过滤中，除了根据 FortiGuard 分类执行监控或阻断动作外，还可以使用 URL 过滤来手动指定哪些网站被允许，哪些网站被阻断。



URL 过滤与 WebFilter 的执行顺序

1. 如果 URL 过滤的动作是允许或监视器，那么将继续执行 WebFilter 分类过滤（如果 WebFilter 通过查询 FortiGuard 的分类的匹配动作是拒绝，那么该 URL 仍然会被拒绝）。
2. 如果 URL 过滤的动作是拒绝，那么该 URL 将被直接拒绝。
3. 如果 URL 过滤的动作是例外，那么该 URL 直接放通，不会执行 WebFilter 分类过滤。

URL 过滤类型

简单：URL 过滤器将执行完整的匹配；

如果网址设置为 support.fortinet.com.cn ↗，那么 URL 过滤器将匹配 support.fortinet.com.cn ↗，不会匹配 www.fortinet.com.cn ↗；

如果网址设置为 fortinet.com.cn ↗，那么 URL 过滤器将匹配 support.fortinet.com.cn ↗，www.fortinet.com.cn ↗，fortinet.com.cn ↗，不会匹配 fortinetxxx.com.cn ↗。

正则表达式：URL 过滤器根据正则表达式执行匹配；

如果网址设置为 fortinet（即.*fortinet.*），那么 URL 过滤器将匹配包含 fortinet 字符的网址，如 www.fortinet.com ↗,www.xxxfortinetyyy.com ↗ 等；

如果网址设置为 forti.*\.com，那么 URL 过滤器将匹配 support.fortinet.com ↗，www.fortiguard.com ↗ 等。

通配符：URL 过滤器根据通配符执行匹配；

如果网址设置为 *，那么 URL 过滤器将匹配所有网址；

如果网址设置为 *.fortinet.com，那么 URL 过滤器将匹配 support.fortinet.com ↗，www.fortinet.com ↗ 等；



URL 过滤动作

例外：该 UR 直接被 URL 过滤器允许，不再执行后续的 FortiGate 过滤器，内容过滤器等；

阻断：该 URL 被 URL 过滤器阻断；

允许：该 URL 被 URL 过滤器允许，但仍然要执行 FortiGuard 过滤器，内容过滤器等；

监视器：动作和允许一样，但记录日志。



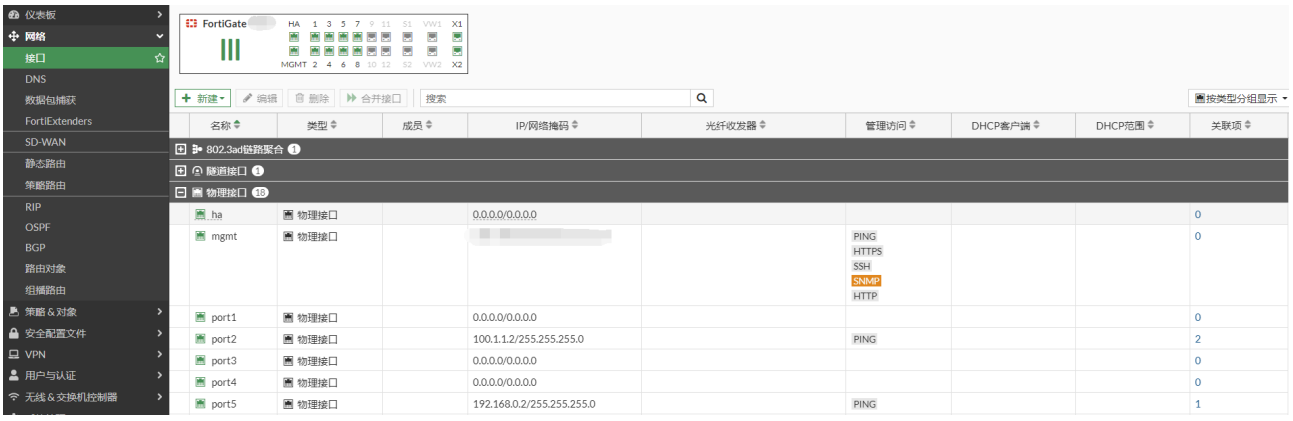
##网络拓扑

PC----------(port5:192.168.0.2/24)FGT(port2:100.1.1.2)----------Internet

配置步骤

1.基本配置

配置接口 IP 和路由。





2.配置 Web 过滤配置文件

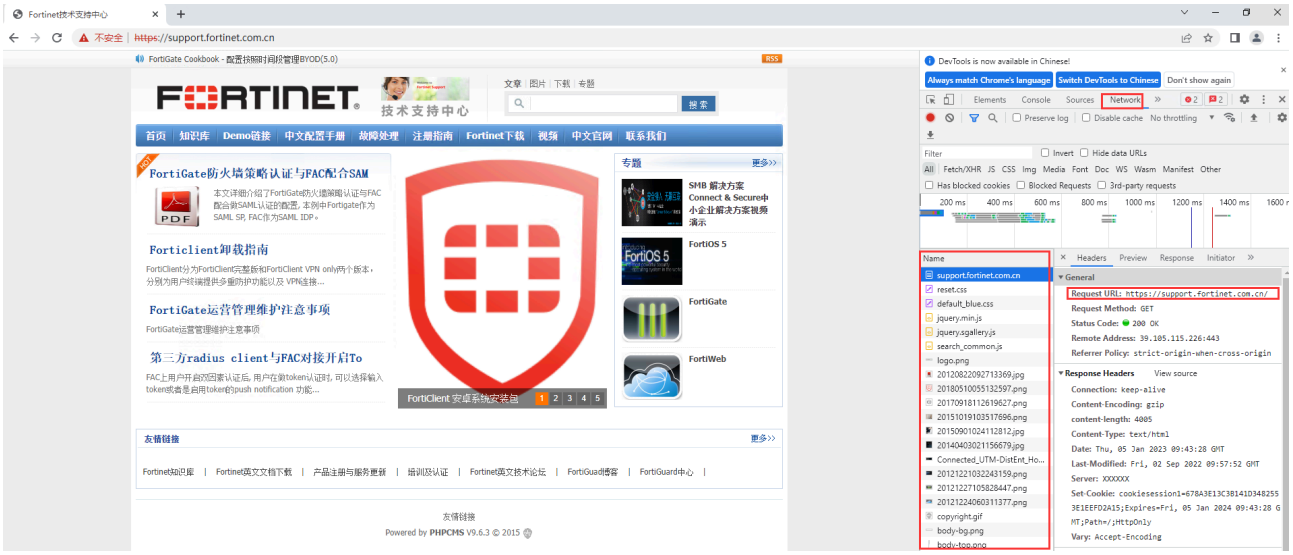
这里直接使用 Default 配置文件，并使用 URL 过滤器只允许 fortinet.com.cn ↗ 域名的网站，阻止其他所有网站。

请勾选”当发生分级错误时允许的网站“。这是为了防止因网络延迟，丢包等原因导致 FortiGate 不能与 FortiGuard webfilter 通信而引起 web 站点无法评估，进而导致该站点不能访问；勾选后如果是由于无法评估引起的错误，将允许访问该站点。



重要

在浏览器访问一个网站中，浏览器不仅仅只是访问该域名本身。网站中有很多图片，js 脚本等等都是需要加载的，因此在 URL 放通网站的时候，需要连那些浏览器内部访问的请求都要放通。浏览器访问一个域名时，通过 F12，点击 network 可以查看到该域名加载了哪些内容。

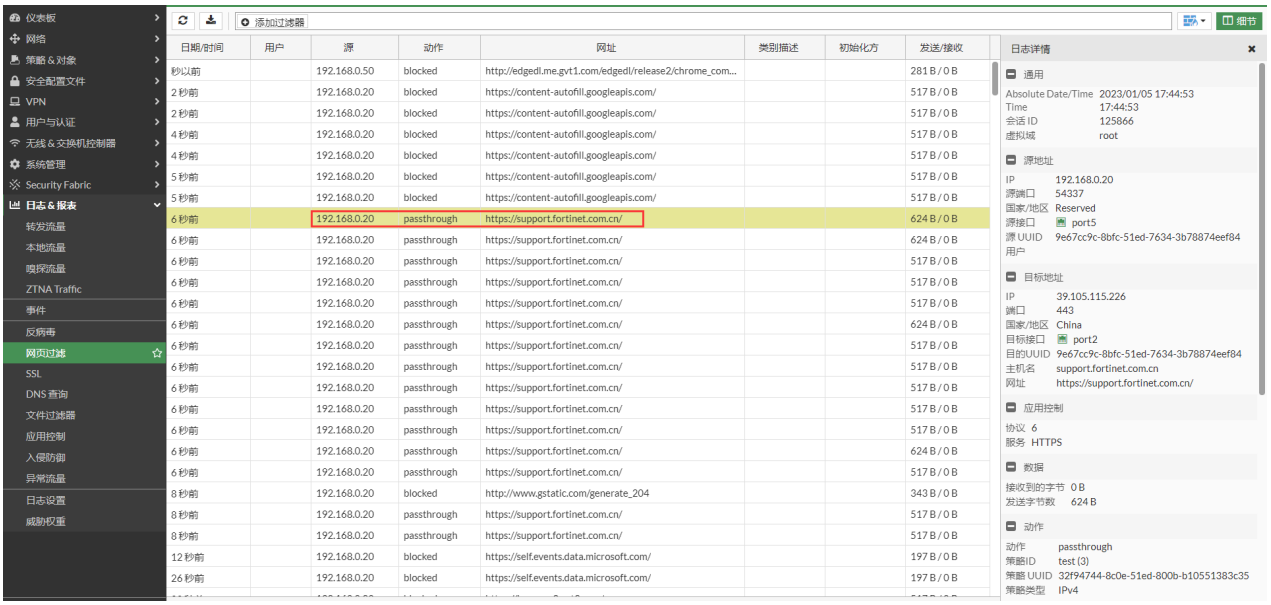


URL 过滤测试

1. PC 访问 support.fortinet.com.cn ↗ 是允许的。

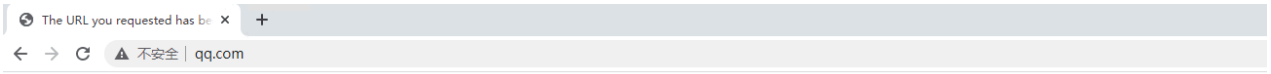


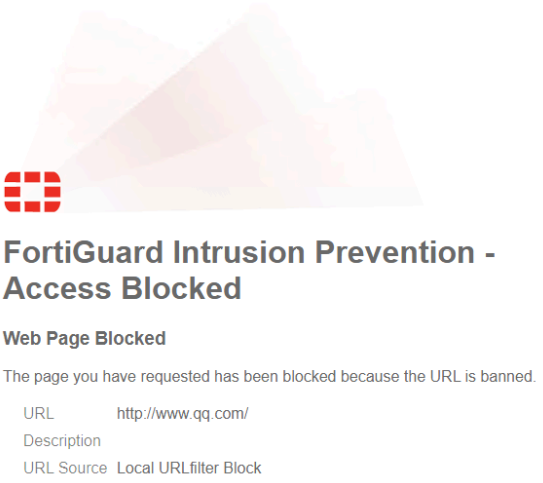
查看 web 过滤日志。



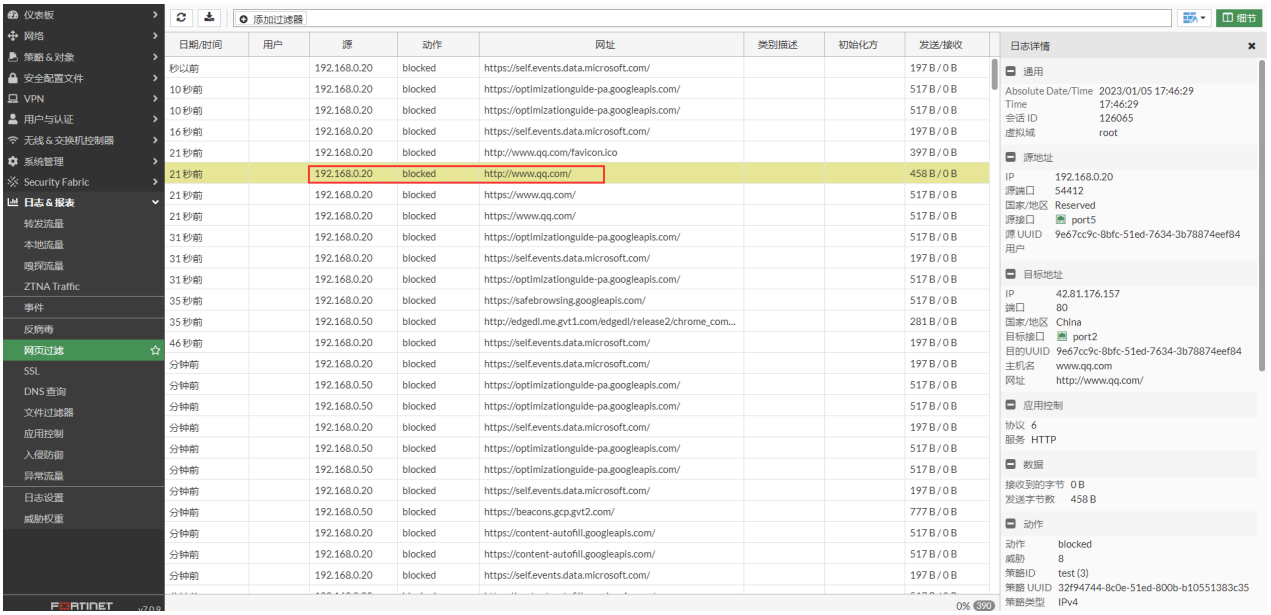


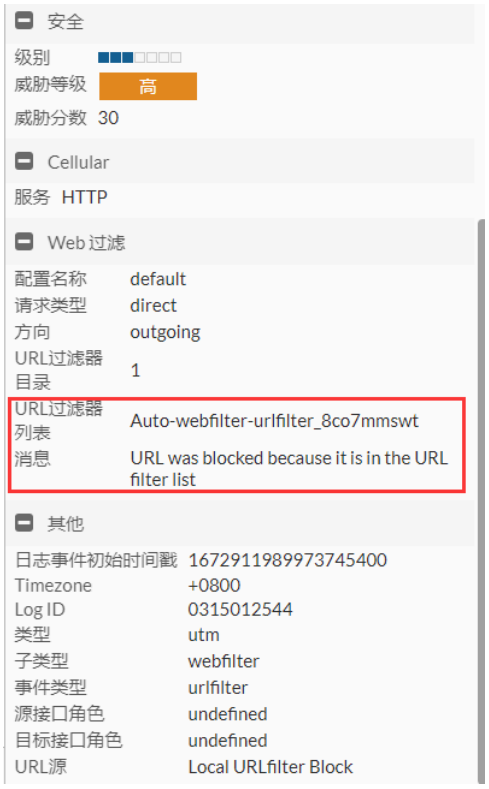
2. PC 访问 www.qq.com ↗ 被 URL 过滤器阻止。





查看 webfilter 日志。


仪表板
网络
策略 & 对象
安全配置文件
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
转发流量
本地流量
嗅探流量
ZTNA Traffic
事件
反病毒
网页过滤
SSL
DNS 查询
文件过滤器
应用控制
入侵防御
异常流量
日志设置
威胁权重
添加过滤器
细节
日期/时间 用户 源 动作 网址 类别描述 初始化方 发送/接收
秒以前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
10 秒前 192.168.0.20 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
10 秒前 192.168.0.20 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
16 秒前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
21 秒前 192.168.0.20 blocked http://www.qq.com/favicon.ico 397 B / 0 B
21 秒前 192.168.0.20 blocked http://www.qq.com/ 458 B / 0 B
21 秒前 192.168.0.20 blocked https://www.qq.com/ 517 B / 0 B
21 秒前 192.168.0.20 blocked https://www.qq.com/ 517 B / 0 B
31 秒前 192.168.0.20 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
31 秒前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
31 秒前 192.168.0.20 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
35 秒前 192.168.0.20 blocked https://safebrowsing.googleapis.com/ 517 B / 0 B
35 秒前 192.168.0.50 blocked http://edgedl.me.gvt1.com/edgedl/release2/chrome_com... 281 B / 0 B
46 秒前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
分钟前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
分钟前 192.168.0.50 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.50 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
分钟前 192.168.0.50 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.50 blocked https://optimizationguide-pa.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
分钟前 192.168.0.50 blocked https://beacons.gcp.gvt2.com/ 777 B / 0 B
分钟前 192.168.0.20 blocked https://content-autofill.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.20 blocked https://content-autofill.googleapis.com/ 517 B / 0 B
分钟前 192.168.0.20 blocked https://self.events.data.microsoft.com/ 197 B / 0 B
日志详情
通用
Absolute Date/Time 2023/01/05 17:46:29
Time 17:46:29
会话 ID 126065
虚拟域 root
源地址
IP 192.168.0.20
源端口 54412
国家/地区 Reserved
源接口 port5
源 UUID 9e67cc9c-8bfc-51ed-7634-3b78874eef84
用户
目标地址
IP 42.81.176.157
端口 80
国家/地区 China
目标接口 port2
目的UUID 9e67cc9c-8bfc-51ed-7634-3b78874eef84
主机名 www.qq.com
网址 http://www.qq.com/
应用控制
协议 6
服务 HTTP
数据
接收到的字节 0 B
发送字节数 458 B
动作
动作 blocked
威胁 8
策略ID test (3)
策略 UUID 32f94744-8c0e-51ed-800b-b10551383c35
策略类型 IPv4
0%


安全
级别
威胁等级 高
威胁分数 30
Cellular
服务 HTTP
Web 过滤
配置名称 default
请求类型 direct
方向 outgoing
URL过滤器目录 1
URL过滤器列表 Auto-webfilter-urlfilter_8co7mmswt
消息 URL was blocked because it is in the URL filter list
其他
日志事件初始时间戳 1672911989973745400
Timezone +0800
Log ID 0315012544
类型 utm
子类型 webfilter
事件类型 urlfilter
源接口角色 undefined
目标接口角色 undefined
URL源 Local URLfilter Block

URL 过滤表达式

简介

本文介绍 WebFilter 在 URL 过滤时的三种表达式类型：

- 简单（Simple）
- 正则表达式（Regular Expression）
- 通配符（Wildcard）

> **提示**
>
> 以下表达式的匹配机制也适用于 DNS Filter 中的 URL 表达式。

简单（Simple）

简单（Simple）的 URL 过滤器条目必须采用标准 URL 格式，可以包含子域名和路径。

举例：`fortinet.com`，`fortinet.com/support`、`handbook.fortinet.com.cn`、`net.com` 等。

当 FortiGate 的防火墙策略处于不同的检测模式（inspection-mode）时，Simple 表达式的匹配方法会有所不同：

- 当防火墙策略配置为流模式（`set inspection-mode flow`，默认配置）时：只要 simple 表达式的字符串出现在实际访问的 URL 中（哪怕只是其中一部分），就会判定为匹配。
- 当防火墙策略配置为代理模式（`set inspection-mode proxy`）时：只有当实际访问的域名与 simple 表达式完全一致，或是 simple 表达式的子域名时，才会匹配。

防火墙策略检测模式	匹配规则说明	Simple 表达式	匹配示例✅	不匹配示例❌

flow（默认）	只要 Simple 表达式的字符串出现在实际 URL 中任意位置，即判定为匹配	`net.com`	`net.com`
			`fortinet.com`
			`www.net.com`
			`www.fortinet.com`
				`net1.com`

		`net.com/prod`	`www.fortinet.com/products/doc`	

proxy	只有当访问的域名与 Simple 表达式完全一致，或是其子域名时，才算匹配（不检查后边）	`net.com`	`net.com`
			`www.net.com`
			`net.com.cn`
			`www.net.com.cn`
				`forinet.com`
				`www.fortinet.com`

		`net.com/prod`	`www.net.com/products/doc`	`www.fortinet.com/products/doc`

include-subdomains 选项

从 FortiOS 7.6.3 版本开始，当防火墙策略的检测模式配置为代理模式（`set inspection-mode proxy`）时，可以在使用 simple 条目时控制子域名匹配的选项。

```
config webfilter urlfilter                                                    bash
    edit <id>
        set include-subdomains {enable/disable}
    next
end
```

- `enable`（默认）：simple 表达式会匹配子域名。例如，`net.com` 会匹配 `www.net.com`、`www.net.com.cn`。
- `disable`：仅匹配确切的域名。例如，`net.com` 只会会匹配 `net.com`、`net.com.cn`，不会匹配 `www.net.com`、`www.net.com.cn`。

通配符（Wildcard）

通配符（Wildcard）表达式可以根据指定的关键字灵活匹配多个 URL，不需要严格的格式，比 simple 类型更灵活。

匹配类型	匹配规则说明	Wildcard 表达式	匹配示例✅	不匹配示例❌
基本匹配	URL 开头左到右扫描，只要任意位置符合规则即匹配成功	`for*guard`	`fortiguard.com`	
中间匹配	若要匹配 URL 中间的字符，需要在开头加 `*`	`*or*guard`	`www.fortiguard.com`	
		`or*guard`		`www.fortiguard.com`

结尾匹配	条目末尾不需要 `*`，自动匹配末尾任意字符，`*fortinet` 等效于 `*fortinet*`	`*fortinet`	`www.fortinet.com`
			`support.fortinet.com`
			`handbook.fortinet.com.cn`	

子域名匹配	条目以 `*.` 开头，会匹配该域名和所有子域名	`*.fortinet.com`	`fortinet.com`
			`www.fortinet.com`
			`support.fortinet.com`	

正则表达式（Regex）

正则表达式（Regex）是一种类似 Perl 语法的规则，用来匹配一个或多个特定的 URL，也可以用来排除不符合条件的 URL。

正则符号	表达式含义	正则表达式	匹配示例✅	不匹配示例❌
`*`	匹配前一个字符出现 0 次或多次	`fortinet*.com`	`fortinetttttt.com`	`fortinetsupport.com`
`/i`	匹配时区分大小写	`/FORTINET/i`	`FORTINET`	`fortinet`
`^`	必须以特定字符串开头	`^fo`	`fortinet.com`	`support.fortinet.com` `notfortinet.com`
`$`	必须以特定字符串结尾	`\.com$`	`fortinet.com`	`fortinet.com.cn`

`.`	匹配任意单个字符	`fortinet.com`	`fortinetacom`
			`fortinetbcom`
			`fortinet.com`
				`fortinetabccom`

`.*`	匹配任意字符	`forti.\*com`	`fortinet.com`
			`fortinetabccom`
			`fortinet-xyz-123com`	

`\`	转义特殊字符	`fortinet\.com`	`fortinet.com`	`fortinetacom` `fortinetbcom`
组合用法		`^fortinet\.com$`	`fortinet.com`	`www.fortinet.com` `fortinet.com.cn`

		`^forti.*\.com`	`fortiguard.com`
			`fortinet.com`
			`fortinet.com.cn`
				`www.fortinet.com`
				`fortinet.net`
				`fortizcom`

附：

> **重要**
>
> [!NOTE]
>
> 正则表达式解释文档：https://perldoc.perl.org/perlre ↗ 正则表达式测试网站：https://regex101.com ↗

注意事项

带路径的 URL 识别

使用 URL 过滤功能时，如果需要识别的 HTTPS 网页携带目录（例如客户端访问的网页为 `https://www.baidu.com/more/`），由于 FQDN 后的网站目录 `/more/` 是在 TLS 协商结束后客户端才会发起 GET 请求，访问 `/more/` 目录的 HTTP GET 请求被 TLS 加密，FortiGate 需要开启防火墙策略中 SSL 检测模式需要使用 `deep-inspection` 模式才能识别到请求目录 `/more` 的 HTTP 报文。

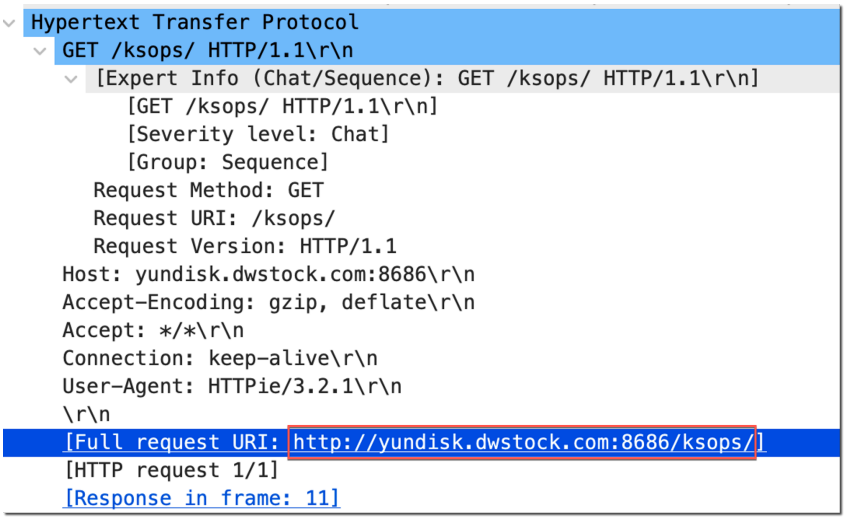
HTTP 网页不需要使用 `deep-inspection` 模式。

URL 中携带 #符号

当需要识别的 URL 中携带 `#` 字符，例如需要拦截的 url 为 `http://yundisk.dwstock.com:8686/ksops/#/login`。

- 配置为 Simple 格式 `http://yundisk.dwstock.com:8686/ksops/#/login` 时无法保存 ULR Filter 配置，提示错误。
- 当使用通配符格式 `http://yundisk.dwstock.com:8686/ksops/*/login` 时无法实现拦截。

实际上客户端在请求 URL `http://yundisk.dwstock.com:8686/ksops/#/login` 时，`#` 及之后的部分并不会发送给 HTTP Server，如下抓包所示，客户端请求的 URI 并不包含 `#` 以及之后的字段 `/login`。



`#` 后的部分被称为 URI fragment，仅在客户端本地浏览器生效，不会发送给 Server（参考：https://en.wikipedia.org/wiki/URI_fragment ↗）。

Web 过滤常见问题

- Chrome/Edge 访问网页无法被拦截

Chrome/Edge 访问网页无法被拦截

问题现象

Web 过滤针对部分网页配置了阻断动作，但 Chrome/Edge 浏览器发起的网页访问无法被阻断，Firefox 浏览器发起的相同网页访问可以正确被 Web 过滤阻断。

使用基于 FortiGuard 的 Web 过滤或 URL 过滤均会出现此问题。

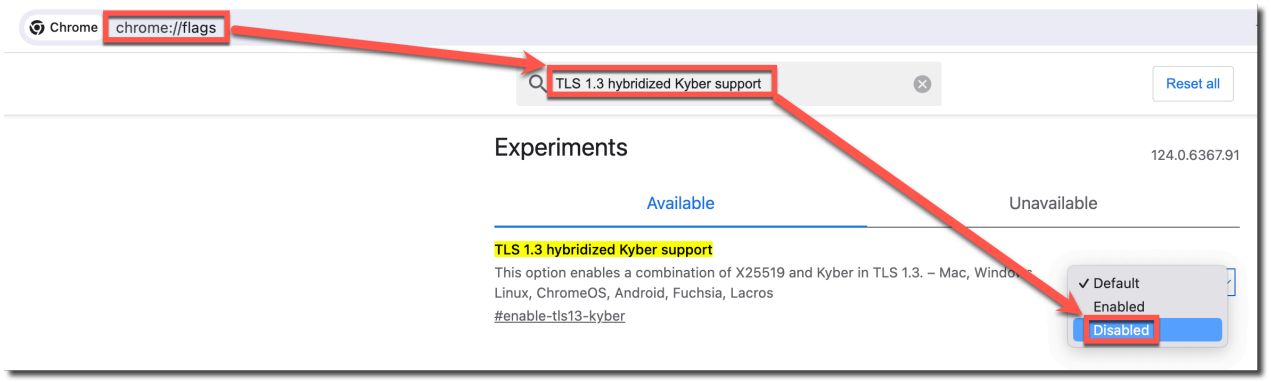
问题原因

从 Chrome 116 版本开始，引入了新特性“TLS 1.3 hybridized Kyber”。该特性默认开启，导致 FortiGate 的 IPS Engine 无法解密 TLS 交互流量，所以 Web 过滤功能失效。

解决方法 1 - 修改 Chrome 配置

1. 在 Chrome 浏览器的地址栏，输入 `chrome://flags`（Edge 为 `edge://flags`），进入 Experiments 页面。

2. 在页面上方的搜索框中搜索 `TLS 1.3 hybridized Kyber support`，并将其修改为 Disabled。



3. 随后清除 Chrome 网页缓存，即可正常拦截 Chrome 的网页访问。

解决方法 2 - 升级 IPS Engine

升级 FortiGate 的 IPS Engine 到 6.0.15:0171、IPSE 7.1.8:0178、IPSE 7.2.6:0331、IPSE 7.4.2:0519 或更高版本后，即可正常拦截 Chrome 的网页访问。

应用控制

- 应用控制简介
- 配置应用控制

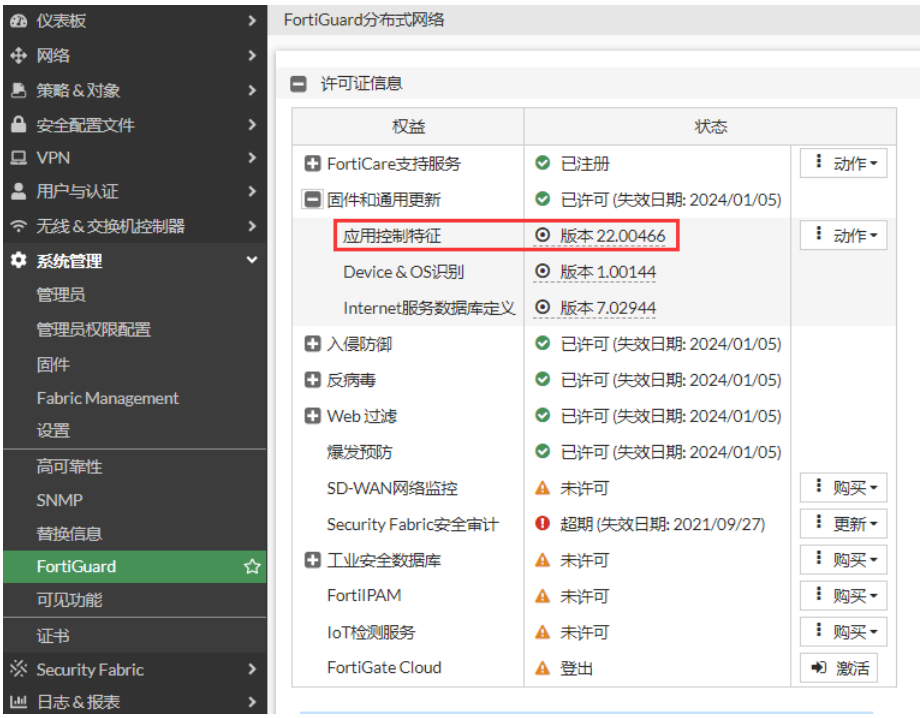
应用控制简介

什么是应用控制

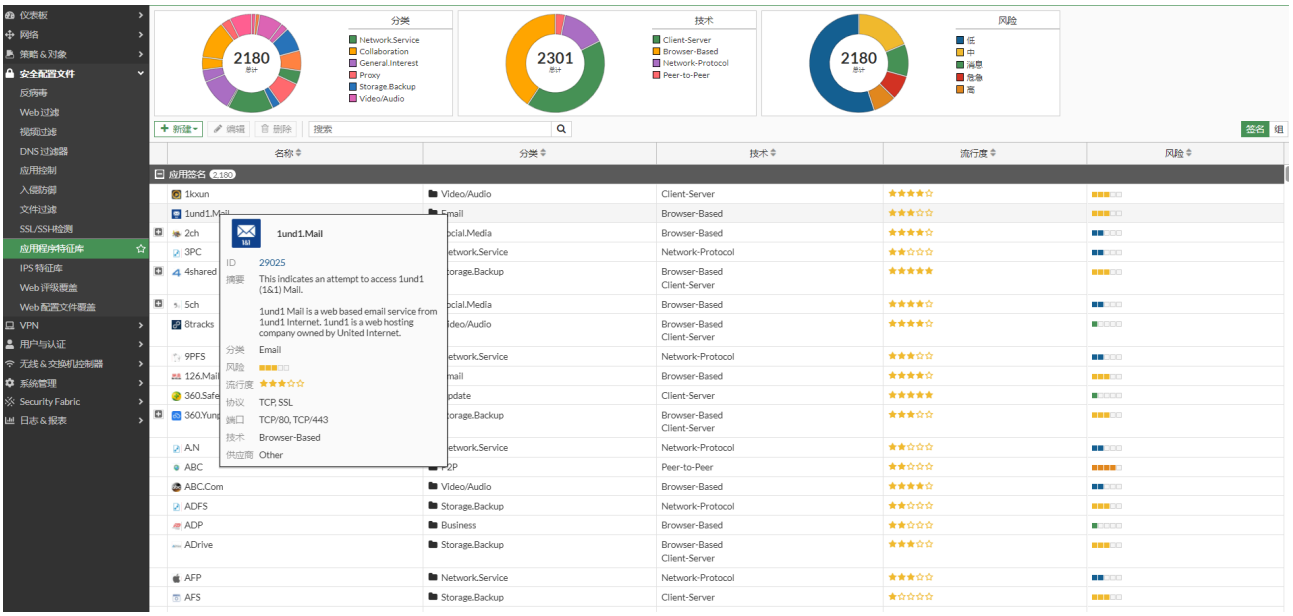
FortiGate 通过扫描流量中的内容来检测应用程序，FortiGate 应用控制可以检测二千多种不同的 Web 应用、软件程序、网络服务及通讯协议等特征，使企业重拾对应用程序的可视化和控制能力。

应用控制特征库

选择“系统管理”-->“FortiGuard”-->“许可证信息”，在”固件和通用更新“菜单下可以查看应用控制特征库的版本。

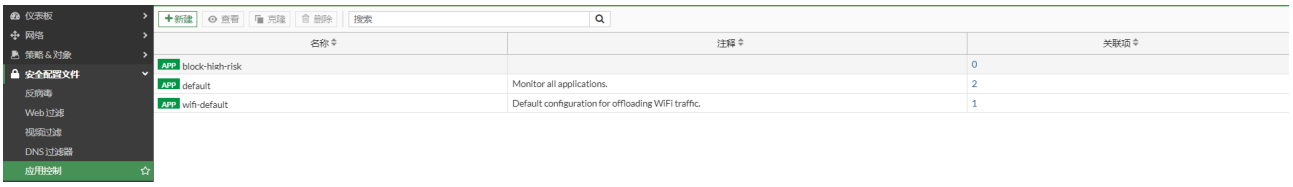


选择“安全配置文件”-->“应用控制特征库”，可以查看具体的特征，将鼠标悬停在某一个特征上，可以查看该特征的详细信息。



应用控制配置文件

应用控制默认有一些配置文件，这些配置文件应用的目标各有不同，也可以根据自身的业务需求创建新的配置文件。



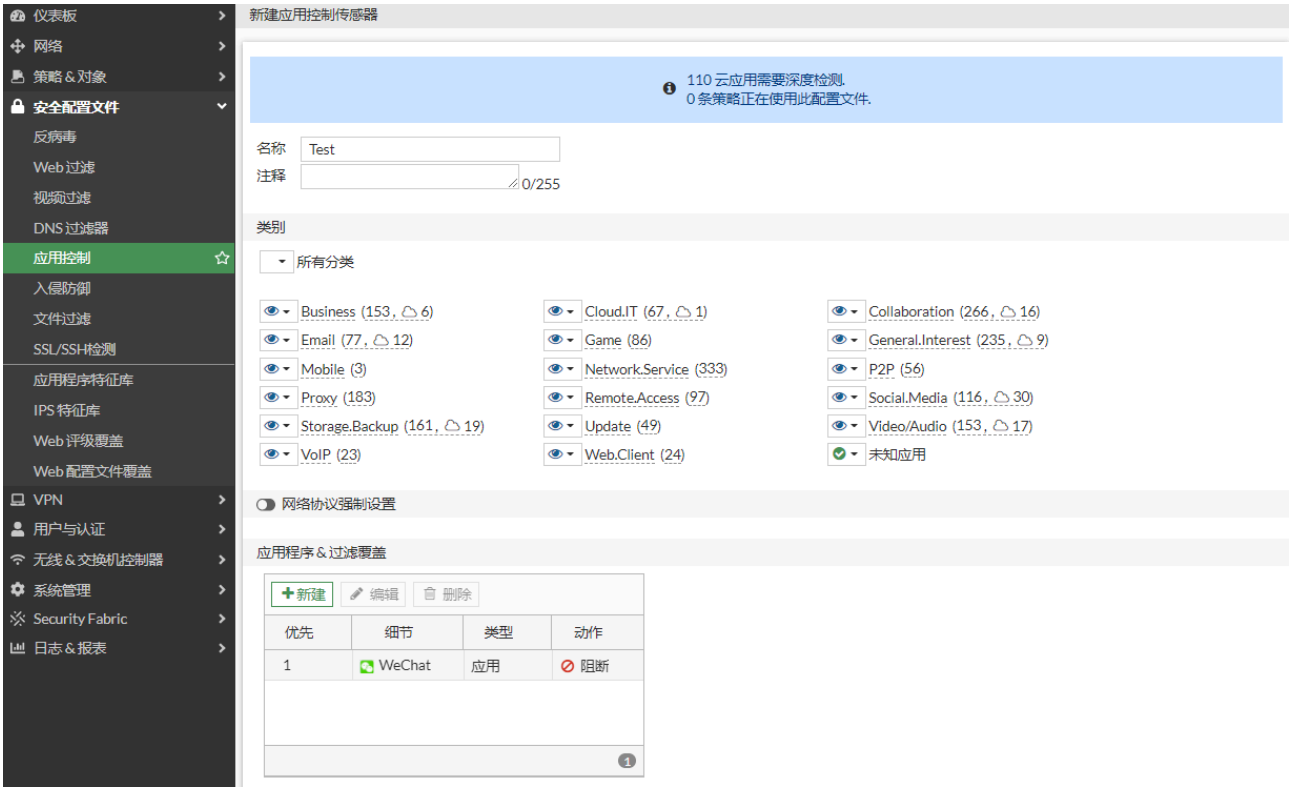
default 配置文件，默认监控所有的应用。



block-high-risk 配置文件，阻断了 Proxy 和 P2P 类别的应用。



用户也可以根据自身的需求创建配置文件，选择“安全配置文件”-->“应用控制”，点击“新建”，如阻断微信。

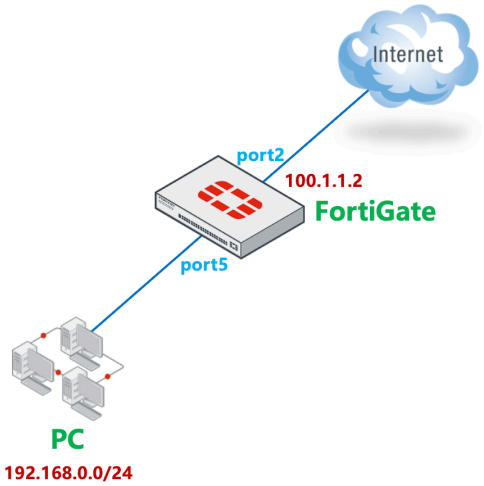

仪表板
网络
策略 & 对象
安全配置文件
反病毒
Web 过滤
视频过滤
DNS 过滤器
应用控制
入侵防御
文件过滤
SSL/SSH检测
应用程序特征库
IPS 特征库
Web 评级覆盖
Web 配置文件覆盖
VPN
用户与认证
无线 & 交换机控制器
系统管理
Security Fabric
日志 & 报表
新建应用控制传感器
110 云应用需要深度检测.
0 条策略正在使用此配置文件.
名称 Test
注释
0/255
类别
所有分类
Business (153, 6)
Cloud.IT (67, 1)
Collaboration (266, 16)
Email (77, 12)
Game (86)
General.Interest (235, 9)
Mobile (3)
Network.Service (333)
P2P (56)
Proxy (183)
Remote.Access (97)
Social.Media (116, 30)
Storage.Backup (161, 19)
Update (49)
Video/Audio (153, 17)
VoIP (23)
Web.Client (24)
未知应用
网络协议强制设置
应用程序 & 过滤覆盖
新建
编辑
删除
优先
细节
类型
动作
1
WeChat
应用
阻断

配置应用控制

网络需求

客户使用应用控监控和限制用户上网的流量。

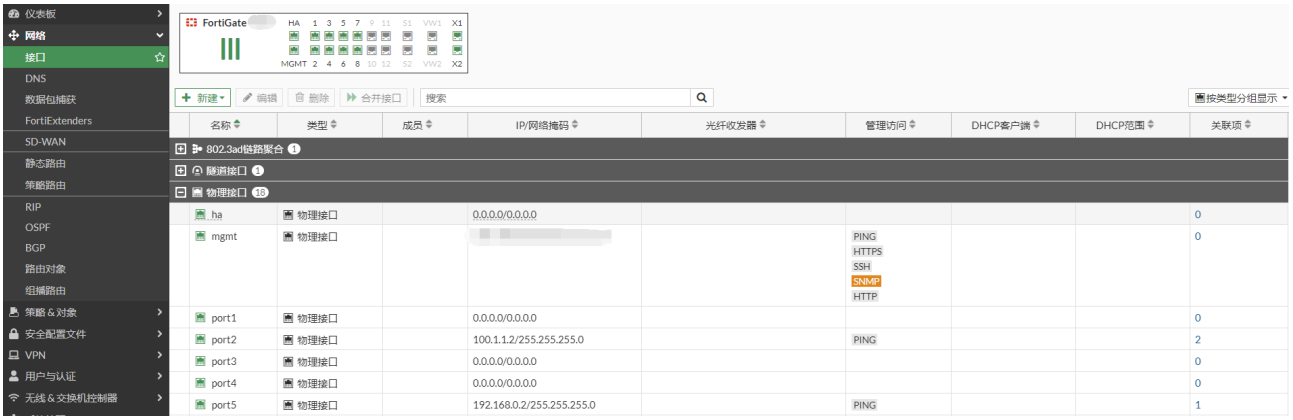
网络拓扑



配置步骤

1.基本配置

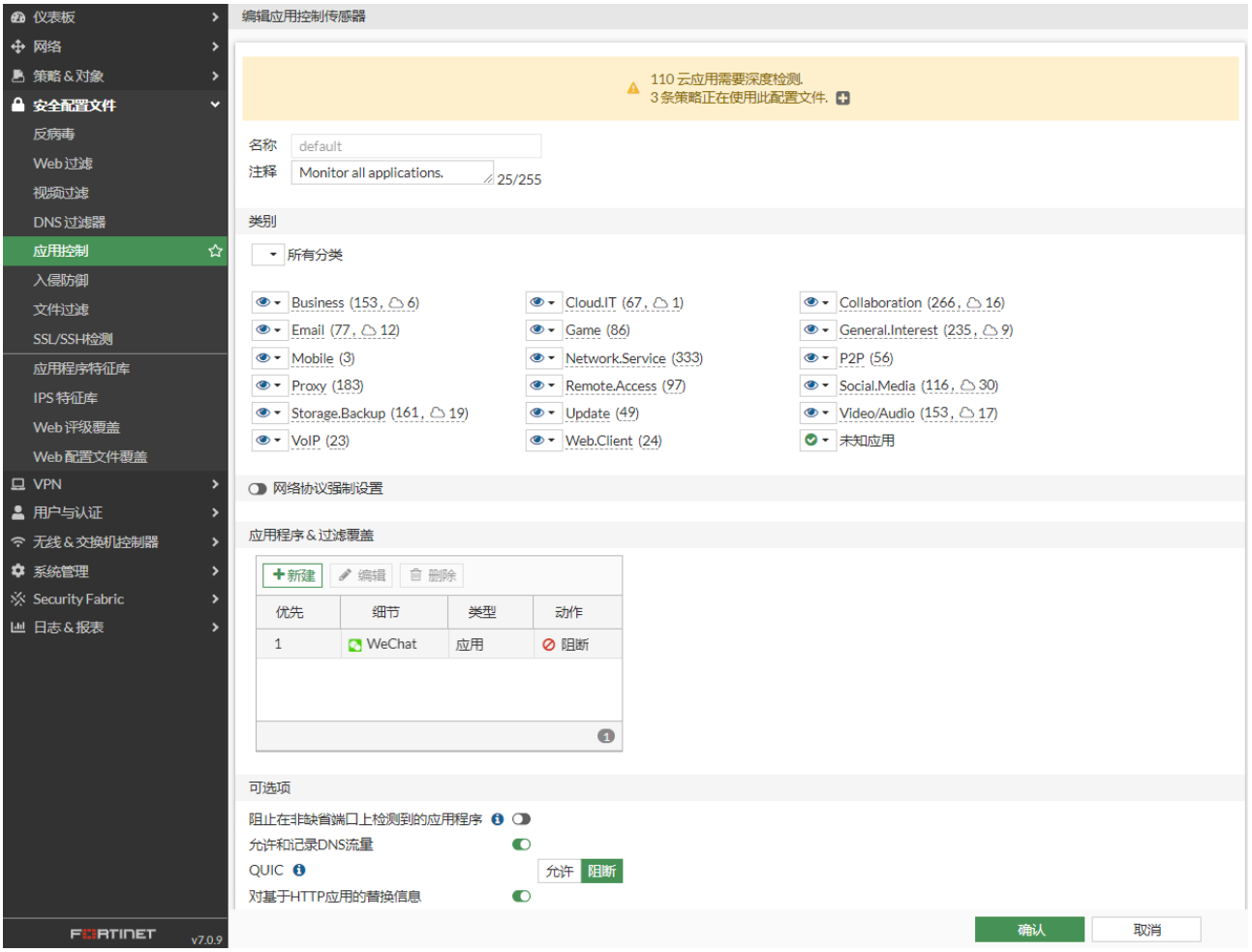
配置接口 IP 和路由。





2.配置应用控制配置文件

这里直接使用 Default 配置文件，并阻断了用户使用微信。



3.配置策略

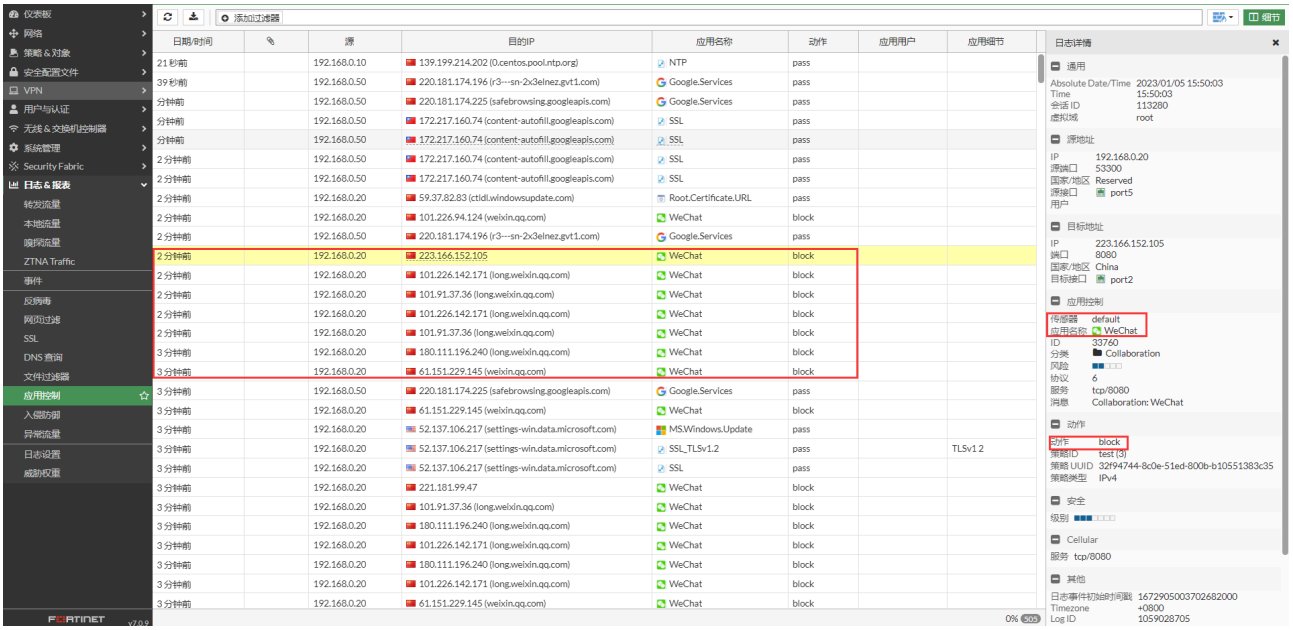


应用控制测试

PC 登录微信被应用控制阻断。



查看应用控制日志。



注意事项

有时会遇到开启应用控制后，腾讯相关的应用都是允许状态，但是微信或企业微信的图片无法发送（或发送缓慢），可能是由于微信/企业微信应用用到了 QUIC 协议，该协议在应用控制配置文件中默认是禁止的，修改为允许后，可以解决此问题。

> **重要**
>
> 在 7.2 及以后版本中，此开关被移除，请在应用控制过滤覆盖中对 QUIC 应用进行控制（QUIC 位于 Network.Service 分类中，默认放通）。




```
config application list
    edit "default"
        set comment "Monitor all applications."
        set options allow-dns allow-quic
        config entries
```

```
            edit 1
                set action pass
            next
        end
    next
end
```

DNS 过滤

- 透明 DNS 转发器

透明 DNS 转发器

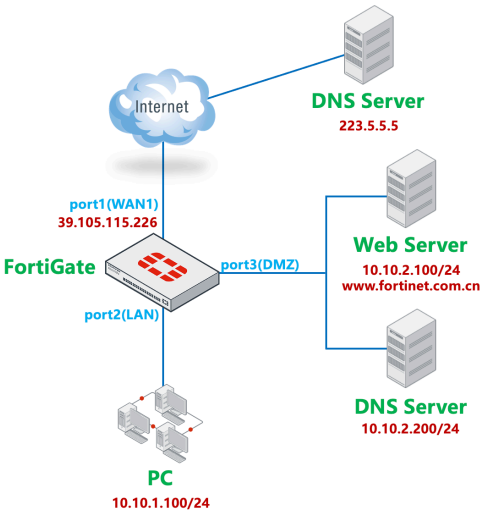
> **重要**
>
> 此功能在 FortiOS 7.4.0 及更新版本上支持。

功能介绍

FortiGate 可以使用透明条件 DNS 转发器（Transparent conditional DNS forwarder）拦截特定域名的 DNS 查询并转发到特定 DNS 服务器，或是直接替换查询结果并回复给客户端。例如当客户端配置的 DNS 服务器位于公网时，为了将目标地址解析到本地的服务器 IP，FortiGate 可以拦截请求并将其重新路由到本地 DNS 服务器进行解析（如果配置了 DNS 数据库条目，也可以直接返回 DNS 结果）。

这样可以更好地控制 DNS 请求，尤其是管理员无法管理客户端设置的 DNS 服务器配置时。通过解析服务器的本地 IP 而不是公网 IP 来提高网络效率，也可以用于特定域名在公网 DNS 服务器的解析结果不符合用户预期的情况。

网络拓扑



1. 内网 PC 填写的 DNS 服务器均为公网 DNS 服务器的 IP `223.5.5.5`。
2. FortiGate 通过 VIP 将 Web Server 的端口映射到 WAN1（port1）上。
3. Web Server 的域名为 `www.fortinet.com.cn`，通过公网 DNS 服务器解析的结果为 FortiGate 配置的 VIP 地址。
4. 需要实现的效果：
   - 外网客户端访问 Web Server 时，直接通过 `www.fortinet.com.cn` 解析到 FortiGate 配置的 VIP IP，通过 FortiGate 的 VIP 映射访问即可。
   - 内网客户端访问 Server 时：
     - 通过公网的 DNS 服务器 223.5.5.5 解析 `www.fortinet.com.cn` 的 IP 地址时，FortiGate 拦截 DNS 请求，并直接回复 DNS 结果为 Web Server 的内网 IP `10.10.2.100`，内网客户端直接通过解析到的 Server 内网 IP 访问服务器。
     - 其他 `fortinet.com.cn` 域的主机名（如 `support.fortinet.com.cn`）的 DNS 请求也会被 FortiGate 拦截，但转发到内网 DNS Server `10.10.2.200` 进行解析，不直接回复 DNS 请求，也不通过外网的 DNS 服务器解析。

配置步骤

1. 基础网络配置、上网策略、与 Server 的 VIP 配置（略）。

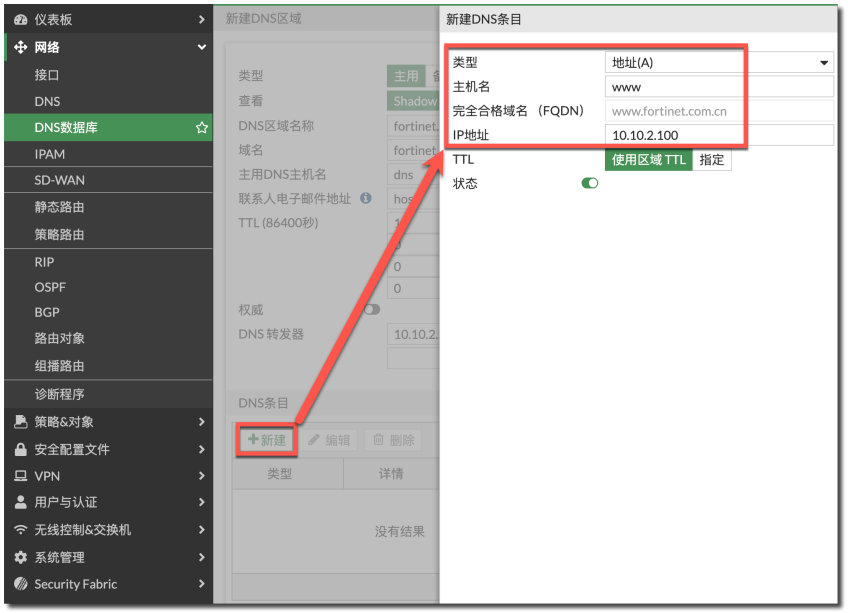
2. 在 FortiGate 上配置 DNS 数据库（需要先在“系统管理 → 可见功能”中开启“DNS 数据库”）。进入“网络 → DNS 数据库”页面，新建 DNS 数据库。



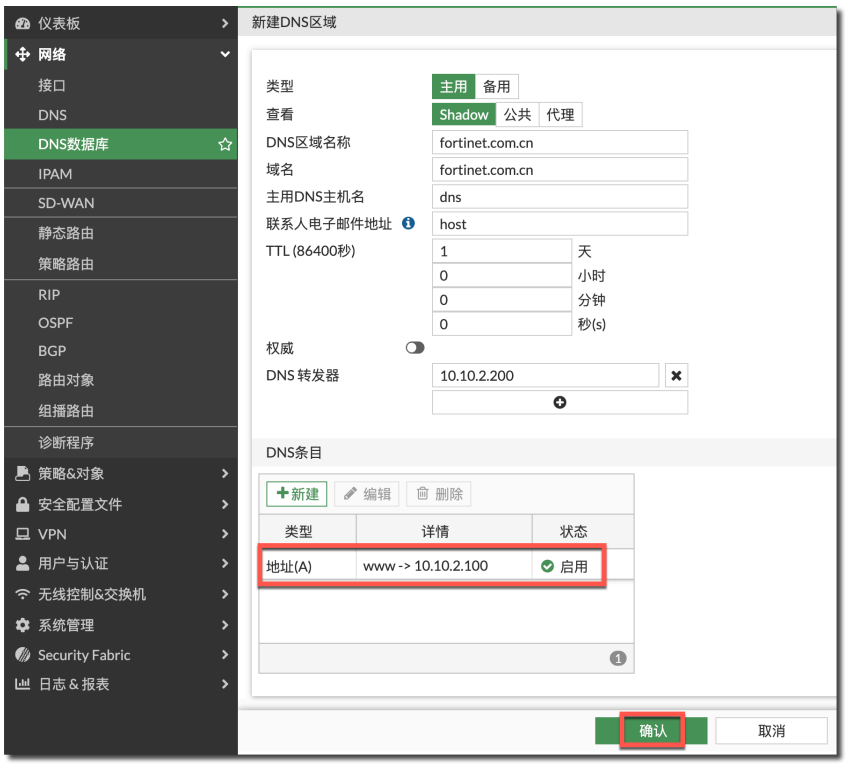
3. 在弹出的新建窗口中，配置区域名称，域名填写为 `fortinet.com.cn`，关闭“权威”选项，DNS 转发器填写内网的 DNS 服务器 `10.10.2.200`（没有匹配上 DNS 数据库条目的主机名会被转发至此 DNS 服务器进行解析）。



4. 在 DNS 条目中点击新建按钮，配置主机名为 `www`，IP 地址为 Web Server 的内网 IP `10.10.2.100`。



5. 点击确认按钮下发 DNS 数据库配置。




```bash
config system dns-database
    edit "fortinet.com.cn"
        set domain "fortinet.com.cn"
        set authoritative disable
        set forwarder "10.10.2.200"
        config dns-entry
            edit 1
                set hostname "www"
                set ip 10.10.2.100
            next
        end
    next
end
```

```bash
FortiGate # diagnose test application dnsproxy 8    //查看DNS数据库信息//
worker idx: 0
vfid=0 name=fortinet.com.cn domain=fortinet.com.cn ttl=86400 authoritative=0 view=shadow type=primary serial=172498465
refresh=0
forwarder(s): 10.10.2.200
source-ip(s): 0.0.0.0 ::
    SOA: fortinet.com.cn (primary: dns.fortinet.com.cn, contact: host@fortinet.com.cn, serial: 172498465)(86400)
      A: www.fortinet.com.cn⟶10.10.2.100(86400)
```

6. 如果需要指定 DNS Database 使用的源 IP，例如在 IPSec 环境中，需要指定 DNS 的源 IP 为 Tunnel 接口的 IP，需要在 CLI 中加入如下配置（此例中不需要配置）。

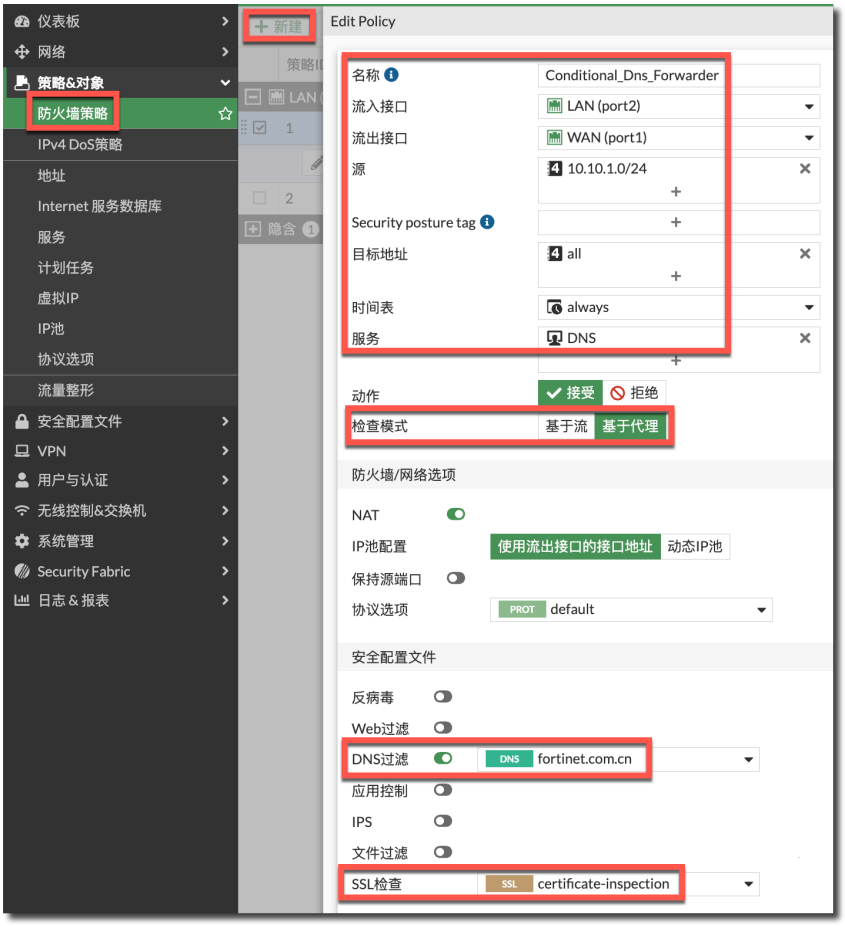
```bash
config system dns-database
    edit "fortinet.com.cn"
        set source-ip <IPv4_address>
    next
end
```

7. 新建 DNS Filter 配置文件，并通过 `transparent-dns-database` 功能引用上述配置的 DNS Database（在此例中，我们仅使用 `transparent-dns-database` 功能，所以关闭了 DNS Filter 的其他过滤功能）。

> **重要**
>
> DNS Filter 配置文件中的 transparent-dns-database 可以引用多个 DNS Database。后续删除 DNS Database 时，需要先取消此处的引用。

```bash
config dnsfilter profile
    edit "fortinet.com.cn"
        config ftgd-dns
            set options ftgd-disable
        end
        set transparent-dns-database "fortinet.com.cn"
    next
end
```

8. 在 FortiGate 上新建防火墙策略，源地址为内网 PC 网段，服务为 DNS 流量，**检查模式需要配置为“基于代理”**，DNS 过滤器选择上步配置的 DNS Filter 配置文件，SSL 检查选择“certificate-inspection”。注意此条策略放置在上网策略之前）。




```bash
config firewall policy
    edit 1
        set name "Conditional_Dns_Forwarder"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "10.10.1.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "DNS"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "certificate-inspection"
        set dnsfilter-profile "fortinet.com.cn"
        set nat enable
    next
end
```



结果验证

1. 公网客户端 PC 使用 DNS 服务器 `223.5.5.5` 解析 Web Server 的域名 `www.fortinet.com.cn`（流量不经过 FortiGate），得到结果为 Server 对应的 VIP 地址 `39.105.115.226`。

```bash
::: important
nslookup www.fortinet.com.cn
:::
Server:     223.5.5.5
Address:    223.5.5.5#53

Non-authoritative answer:
Name:   www.fortinet.com.cn
Address: 39.105.115.226
```

2. 内网客户端 PC 使用 DNS 服务器 `223.5.5.5` 解析 `www.fortinet.com.cn` 的域名（流量经过 FortiGate），得到结果为 Web Server 内网 IP `10.10.2.100`。DNS 流量被 FortiGate 拦截，由 FortiGate 代替公网 DNS 服务器回复 DNS 查询结果，这个过程对于客户端是透明的。

```bash
::: important
nslookup -type=a www.fortinet.com.cn 223.5.5.5
:::
服务器:  public1.alidns.com
Address:  223.5.5.5

非权威应答:
名称:    www.fortinet.com.cn
Address:  10.10.2.100
```

3. 内网客户端 PC 使用 DNS 服务器 `223.5.5.5` 解析 `support.fortinet.com.cn` 的域名（DNS 数据库条目中不存在的主机名）：

   - DNS 流量被 FortiGate 拦截，由 FortiGate 转发至内网 DNS 服务器 `10.10.2.200` 进行解析，这个过程对于客户端是透明的。

     ```bash
     FortiGate # diagnose sniffer packet any 'host 223.5.5.5 or host 10.10.2.200' 4
     Using Original Sniffing Mode
     interfaces=[any]
     filters=[host 223.5.5.5 or host 10.10.2.200]
     ...
     7.354793 port2 in 10.10.1.100.65195 → 223.5.5.5.53: udp 41
     7.355069 port3 out 10.10.2.1.3313 → 10.10.2.200.53: udp 41
     7.355655 port3 in 10.10.2.200.53 → 10.10.2.1.3313: udp 57
     7.355764 port2 out 223.5.5.5.53 → 10.10.1.100.65195: udp 57
     ...
     ```

   - 解析结果为内网 DNS 服务器返回的结果 `10.10.2.130`，这个结果是内网 DNS 服务器返回的。

     ```bash
     C:\Users\Administrator.SUMMERICE2019>nslookup -type=a support.fortinet.com.cn
     服务器:  public1.alidns.com
     Address:  223.5.5.5

     名称:    support.fortinet.com.cn
     Address:  10.10.2.130
     ```

   - 查看 FortiGate 的 DNS 缓存，已经记录了 `support.fortinet.com.cn` 的 DNS 缓存信息，在 TTL（最后一个 `:` 符号后的数字）结束前，客户端再次请求该域名时，FortiGate 会直接响应 DNS 请求。

     ```bash
     FortiGate # diagnose test application dnsproxy 7
     worker idx: 0
     ...
     vfid=0, name=support.fortinet.com.cn, ttl=86400:86250:1650
              10.10.2.130 (ttl=86400)
     ...
     ```

SSL/SSH 检查

- 检测模式
- SSL/SSH 检测配置文件
- certificate-inspection
- deep-inspection
- 保护 SSL 服务器

检测模式

SSL（Secure Sockets Layer）内容扫描与检查功能用于对加密流量执行安全检测，使系统能够在 SSL/TLS 加密通信中应用 IPS、AV、Web Filter 等安全功能。

SSL/SSH 检测配置文件可直接应用于防火墙策略中。FortiOS 默认内置四种 SSL/SSH 检测配置文件，其中三种为只读配置，可被克隆并修改使用：

- certificate-inspection（只读，可克隆）
- deep-inspection（只读，可克隆）
- no-inspection（只读，可克隆）
- custom-deep-inspection（可编辑）

SSL/SSH 检测配置文件通常应用于出口防火墙策略，适用于目标地址不明确的流量场景。根据策略需求，可配置以下检查参数：

- 用于解密 SSL 加密流量的 CA 证书
- 参与检查的 SSL 协议类型
- 各 SSL 协议对应的检测端口
- 是否允许无效 SSL 证书
- 是否对 SSH 流量执行检查
- 可绕过 SSL 检查的地址或 Web 分类白名单

检测模式对比

在实际策略应用中，主要关注以下三种模式：no-inspection、certificate-inspection、deep-inspection。它们的差异核心在于：是否解密流量、可获取的信息深度、对业务/性能的影响。

certificate-inspection

- 工作方式：不解密业务内容，仅检查 TLS 握手阶段的证书与部分协议特征，防火墙策略中的默认配置。
- 可见信息：服务器证书、证书链、有效期、签发者、以及部分 TLS 握手信息（例如 SNI 等）。
- 能力与限制：
  - 可基于证书可信性/异常证书进行阻断或告警（例如过期、吊销、校验失败等）。
  - 相比 deep-inspection，对性能影响更小、对业务兼容性更高。
  - 由于不解密载荷，无法对加密内容（如 HTTPS）执行深度的 AV/IPS/Web Filter 等检测。
- 典型场景：大多数通用出口流量的“轻量检测”、对业务敏感但仍希望拦截恶意/异常证书的网站。

no-inspection

- 工作方式：不对 SSL/TLS 流量进行检查与解密，流量保持端到端加密直通。
- 可见信息：仅检查/记录最基础的连接层信息（如源/目的 IP、端口、会话信息等）。
- 能力与限制：
  - 无法对加密内容执行 AV/IPS/Web Filter 等基于内容的检测。
  - 适用于对隐私/合规要求极高或无法安装证书的场景。
- 典型场景：银行/医疗等敏感业务、第三方应用不允许解密、故障排查期间临时放行等。

deep-inspection

- 工作方式：对 SSL/TLS 流量进行“中间人解密”与重签名，解密后将明文交给安全模块（如 IPS/AV/Web Filter）检测，再重新加密转发。通常需要在终端侧部署并信任 FortiGate（或自颁发） 的 CA 证书。
- 可见信息：完整的应用层内容（HTTP/HTTPS 内容、文件下载内容、URL/Host 等更完整的特征）。
- 能力与限制：
  - 可对加密流量应用 AV/IPS/Web Filter/DLP 等内容级检测，安全能力最强。
  - 对性能消耗更大，并可能影响某些对证书绑定/校验严格的应用（需通过白名单/例外策略保障兼容性）。
  - 需要每个终端信任 CA 证书，否则可能出现证书告警或连接失败。
  - 要求“中间人解密”与重签名的证书必须是 CA 证书或 Sub CA 证书，不能使用用户证书。
- 典型场景：对外网访问需要强安全管控的办公终端、重点部门出网、对下载/恶意网站拦截要求高的环境。

配置建议

- certificate-inspection：当希望兼顾安全与兼容性/性能，且主要关注证书异常与基础风险控制时。大部分情况下推荐此配置（防火墙策略中的默认配置）。
- no-inspection：当业务明确要求不解密、或对解密高度敏感/不具备终端证书部署条件时。

- deep-inspection：当组织允许终端部署证书、且需要最大化加密流量可视化与防护能力时。

SSL/SSH 检测配置文件

配置方法

GUI

进入安全配置文件 → SSL/SSH 检测，点击新建按钮。

相关信息

预置的 custom-deep-inspection 配置文件也可直接编辑使用。



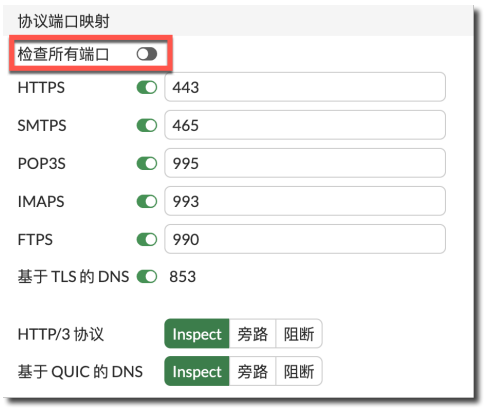
- SSL 检测选项：
  - 启用 SSL 检测：
    - 多个客户端连接到多个服务器：用于目标未知的通用策略，常用于出站互联网流量。
    - 保护 SSL 服务器：用于特定 SSL 服务器，需在“服务证书”配置处指定服务器证书。
  - 检查方法：
    - SSL 证书检查（certificate-inspection）：仅检查证书（到 SSL/TLS 层的头部信息），不检查加密内容。可对 ECH 做允许或阻断。
    - 完整 SSL 检查（deep-inspection）：检查 SSL/TLS 加密载荷内容。
  - CA 证书：从已安装证书中选择用于检查/解密的 CA 证书。可点击 Download 下载证书。
  - 恶意证书：配置阻断或允许潜在恶意证书。可通过“查看恶意证书定义”查看被阻断证书列表（含原因、SHA1 指纹与日期）。
  - 未受信任 SSL 证书：配置服务器证书非受信任 CA 签发时的动作。
    - 允许（默认）。
    - 阻断。

        - 忽略：仅对“deep-inspection”有效，重新签名为受信任。说明：在 GUI 中对证书检查模式配置 Ignore 不生效且不会保存。
        - 可通过“查看信任 CA 列表”查看 FortiGate 信任的内置与导入 CA。
    - 服务器证书 SNI 检查：检查 SSL Client Hello 消息中的 SNI 与服务器证书 CN 或 SAN 是否匹配。
        - 启用（默认）：不匹配时使用证书 CN 进行 URL 过滤。
        - 严格：不匹配时关闭连接。
        - 禁用：禁用检查。
    - 强制 SSL 加密算法合规：启用/禁用 SSL 密码套件合规性检查，仅“deep-inspection”可用。
    - 强制 SSL 协商过程合规：启用/禁用 SSL 协商合规性检查，仅“deep-inspection”可用。
    - 基于 HTTPS 的 RPC：启用/禁用 RPC over HTTPS 流量检查，仅“deep-inspection”可用。
    - 基于 HTTPS 的 MAPI：启用/禁用 MAPI over HTTPS 流量检查。
- 协议端口映射：
    - 检查所有端口：启用后使用 IPS 引擎检查所有端口。可选择是否禁用 DNS over TLS。
    - HTTPS/SMTPS/POP3S/IMAPS/FTPS：若未启用“检查所有端口”，可在对应协议旁指定需要检查的端口。
    - 基于 TLS 的 DNS：启用/禁用 DNS over TLS 检查。
    - 加密的客户端 Hello：允许或阻止使用加密 Client Hello（ECH）的 TLS 连接，仅在“检查方法”为 certificate-inspection 时可用。
    - HTTP/3 协议：
        - 当检查方法为“SSL 证书检查”或 HTTPS 被禁用时，HTTP/3 固定为 Bypass 且不可修改。
        - 当检查方法为“deep-inspection”时，可选 Inspect、旁路、阻断。
    - 基于 QUIC 的 DNS：当检查方法为检查方法 deep-inspection 时可用。可选 Inspect、旁路或阻断。
- 免除 SSL 检查：
    - 信誉良好的网站：仅“deep-inspection”可用。用于指定信誉网站、FortiGuard Web 分类或地址对象的豁免。可启用日志记录豁免。
- SSH 检测选项：
    - SSH 深度检测：启用/禁用 SSH 协议深度扫描；启用后 SSH 端口可配置。
    - SSH 端口：定义要检测的 SSH 协议数据包的端口：
        - 任何：检测所有 SSH 流量。
        - 指定：仅检测指定 TCP 端口。
- 通用选项：
    - 无效的 SSL 证书：配置无效证书的放行/阻断策略。设为自定义时可分别配置过期、回收（吊销）、验证超时、验证失败等场景。深度检查下可选“保持不受信任 & 允许”、“阻断”、“信任 & 允许”。
    - 记录 SSL 异常日志：启用后记录包含无效证书的会话日志，默认启用。日志生成在安全事件日志类型下的 SSL 子类型。

CLI

```bash
config firewall ssl-ssh-profile
    edit <name>
        set allowlist [enable|disable]
        set block-blocklisted-certificates [disable|enable]
        set caname {string}
        set comment {var-string}
        config dot
            Description: Configure DNS over TLS options.
            set cert-validation-failure [allow|block| ... ]
            set cert-validation-timeout [allow|block| ... ]
            set client-certificate [bypass|inspect| ... ]
            set expired-server-cert [allow|block| ... ]
            set proxy-after-tcp-handshake [enable|disable]
            set quic [inspect|bypass| ... ]
            set revoked-server-cert [allow|block| ... ]
            set sni-server-cert-check [enable|strict| ... ]
```

检测所有端口



为优化系统资源，FortiGate 支持针对特定协议启用或禁用协议端口映射。每种协议都有默认的 TCP 端口，管理员可以根据实际需要修改端口配置，以便对非标准端口上的加密流量进行检测。FortiGate 会根据数据包头部信息识别协议类型。支持的协议包括：

- HTTPS
- SMTPS
- POP3S
- IMAPS
- FTPS
- DNS over TLS（DoT）

“检测所有端口”的实际效果，取决于防火墙检测模式（flow/proxy）和 SSL 检查深度（certificate-inspection/deep-inspection）的配置。

相关信息

防火墙策略的检测模式介绍可参考：故障排查 → 数据包处理流程 → 安全检测模式 ↗章节。

策略检测模式/证书检测模式	检测所有端口状态	行为
Proxy + deep-inspection	禁用	仅扫描“协议端口映射”中指定端口
Proxy + deep-inspection	启用	扫描所有端口
Flow + deep-inspection	启用/禁用	扫描所有端口
certificate-inspection	禁用	仅扫描“协议端口映射”中指定端口（Flow/Proxy 一致）
certificate-inspection	启用	扫描所有端口（Flow/Proxy 一致）

certificate-inspection

> **注意**
>
> 当希望兼顾安全与兼容性/性能，且主要关注证书异常与基础风险控制时。大部分情况下推荐此配置（防火墙策略中的默认配置），而不是 deep-inspection。

功能简介

FortiGate 支持 SSL/TLS 证书检查（certificate-inspection）。与 deep-inspection 相比，certificate-inspection 模式仅检查至 SSL/TLS 证书层级的报文头信息，不会解密或检查加密后的实际内容。

该模式主要适用于以下场景：

- 不希望对 HTTPS 流量进行深度解密（出于隐私或合规考虑）。
- 仍需要基于证书信息对网站访问进行控制。
- 结合 Web Filter、认证、告警等安全策略使用。
- 若需要内容级安全检测，应使用深度检查模式（参考：UTM → SSL/SSH 检查 → certificate-inspection ↗ 章节）。

FortiGate 默认提供预置的 certificate-inspection 配置文件，可直接使用（但无法修改配置），也可以克隆或新建 SSL/SSH 配置文件。

工作模式

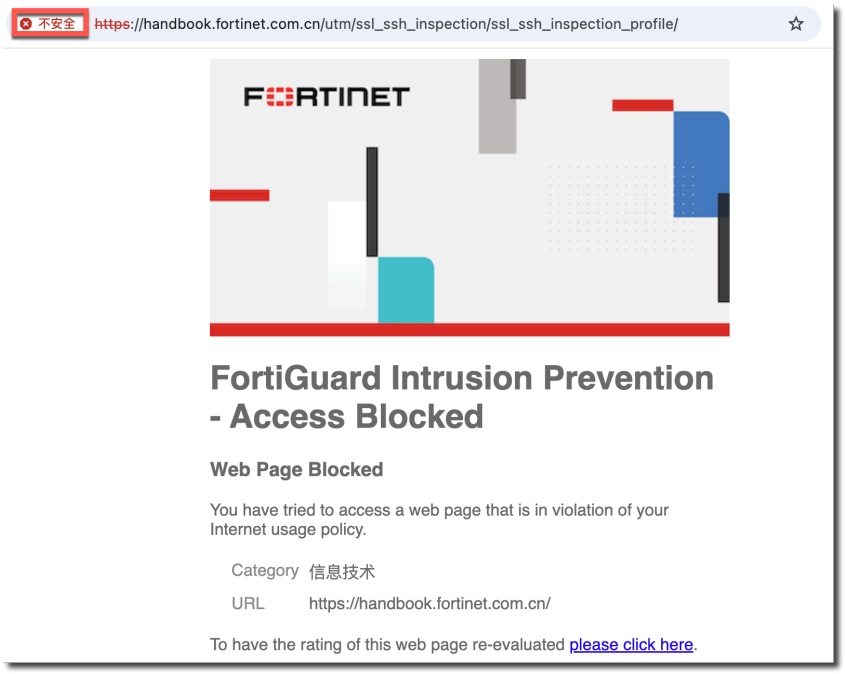
在应用 certificate-inspection 的防火墙策略中：

```bash
config firewall policy
    edit 1
        set name "to_Internet1"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "default"
        set nat enable
    next
end
```

- FortiGate 会解析 HTTPS 连接的 SSL/TLS 证书信息。
- 根据证书信息触发 Web Filter、认证或阻断动作。
- 不解密 HTTPS 会话内容。
- 当访问被拦截或触发替换页面时，返回的 HTTPS 页面由 FortiGate 生成，并使用 FortiGate CA 证书进行签名。

替换页面与证书提示

在以下情况下，用户访问 HTTPS 网站时可能会看到替换页面：



- 网站被 Web Filter 分类阻断。
- Web Filter 分类触发告警页面。
- 访问需要用户认证的页面。
- 其他策略触发的 HTTPS 拦截行为。

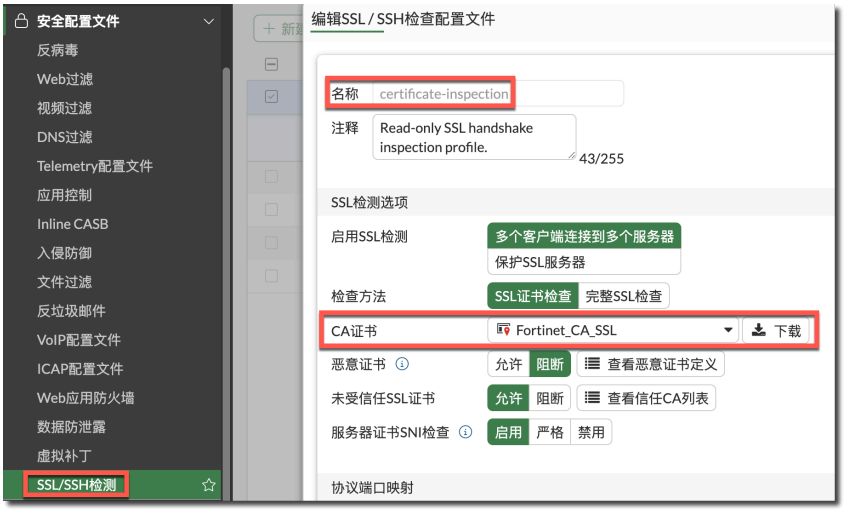
替换页面默认使用 FortiGate CA 证书签名（也可以替换成导入的自定义 CA 证书），因此：



- 若客户端未信任该 CA，会提示证书不受信任（仅针对 FortiGate 替换页面，不会影响真实业务访问的目标网站）。
- 属于正常现象，不代表配置错误。

客户端证书信任配置

为避免客户端出现 HTTPS 证书警告，需要在终端设备上安装 FortiGate 使用的 CA 证书并信任。获取 CA 证书步骤如下：



1. 进入安全配置文件 → SSL/SSH 检测。

2. 编辑当前防火墙策略所使用的 SSL/SSH 检测配置文件，如上所示为系统预置的 certificate-inspection 配置文件。

3. 在 CA 证书右侧点击“下载”按钮。

> **提示**
>
> - 内置的 certificate-inspection 配置文件默认使用的 CA 为 Fortinet_CA_SSL，无法修改。
> - 可在设备证书库中导入自定义 CA 证书（参考：系统管理 → 系统设置 → 证书管理 ↗章节，必须导入带私钥的 CA 证书或 SubCA 证书，不能使用用户证书），并克隆或新建 SSL/SSH 检测配置文件调用自定义 CA 证书。
> - 导入的自定义 CA 证书位于“本地证书”分类中。

4. 将下载的 CA 证书加入客户端系统或者浏览器的信任列表。

> **重要**
>
> - Chrome/Edge 依赖系统的证书信任列表。
> - 而 Firefox 有自己的信任列表，需要在浏览器设置中添加，或在浏览器的证书设置中开启“允许 Firefox 自动信任您安装的第三方证书”，开启后 Firefox 也会信任系统的证书信任列表。
>
> 
>

Windows

1. 双击下载的 CA 证书，点击“安装证书 (I)...”。



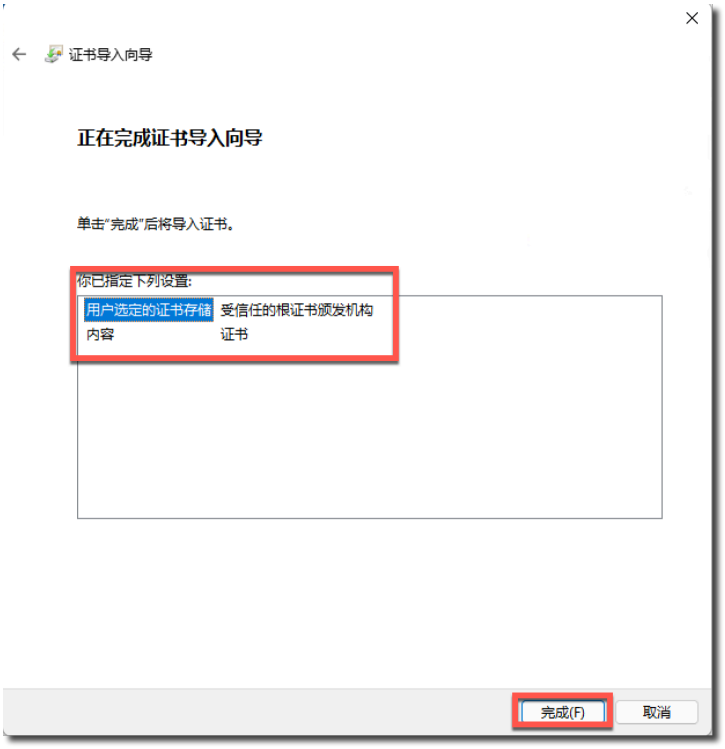
2. 存储位置选择“本地计算机 (L)”，然后点击下一步。



3. 选择“将所有的证书都放入下列存储 (P)”，点击“浏览 (R)...”按钮，选择“受信任的根证书颁发机构”。



4. 完成证书导入。



macOS

1. 双击下载的 CA 证书，会打开“钥匙串访问”应用。

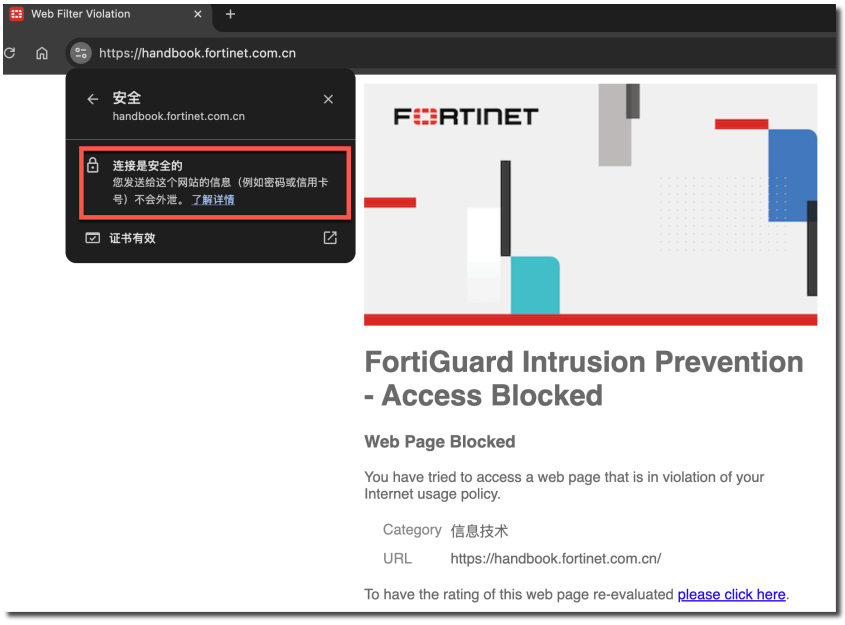
2. 进入“系统 → 证书”页面，可以看到刚刚导入的 CA 证书，然后双击该证书。



3. 在“使用此证书时”选项中选择“始终信任”，关闭窗口（会提示输入系统密码）。



5. 客户端再次访问该网站时，会直接显示出 FortiGate 的阻断页面（或登录页面），不会弹出证书告警页面。

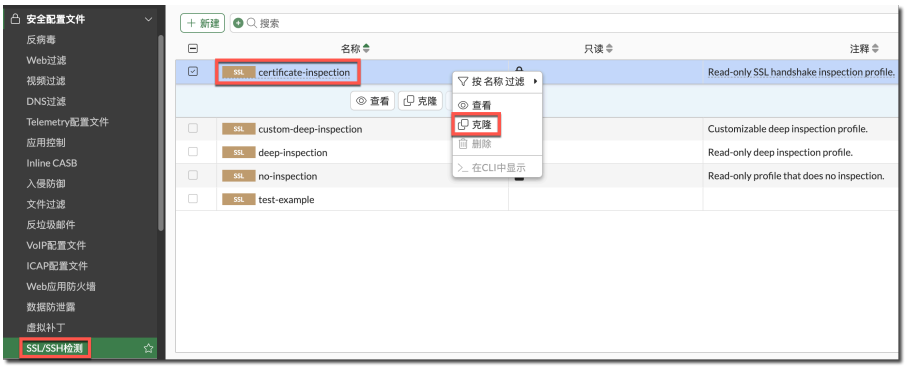


自定义 certificate-inspection 配置文件

默认的 certificate-inspection 配置文件为只读，无法修改任何配置（仅监听 TCP 443 端口，使用出厂自带的 Fortinet_CA_SSL 作为 CA 证书等），当有如下配置需求（包含但不限），必须新建或克隆一个证书检查配置文件。

- 需要检查非标准 HTTPS 端口（例如 8443）。
- 需要使用自定义的 CA 证书。
- 需要自定义网站证书无效时的动作。
- 其他自定义参数，如针对无效 SSL 证书配置处理方式，例如允许、阻断、根据不同证书错误类型设置不同动作等，请参考：UTM → SSL/SSH 检查 → SSL/SSH 检测配置文件↗章节。

1. 进入安全配置文件 → SSL/SSH 检测，新建或克隆 certificate-inspection 配置文件。



2. 在“协议端口映射”中配置 HTTPS 端口，如 443、8443，在“SSL 检测选项”中配置自定义的 CA 证书，根据需要配置其他选项。

> **提示**
>
> - 内置的 certificate-inspection 配置文件默认使用的 CA 为 Fortinet_CA_SSL，无法修改。
> - 可在设备证书库中导入自定义 CA 证书（参考：系统管理 → 系统设置 → 证书管理↗章节，必须导入带私钥的 CA 证书或 SubCA 证书，不能使用用户证书），并克隆或新建 SSL/SSH 检测配置文件调用自定义 CA 证书。
> - 导入的自定义 CA 证书位于“本地证书”分类中。



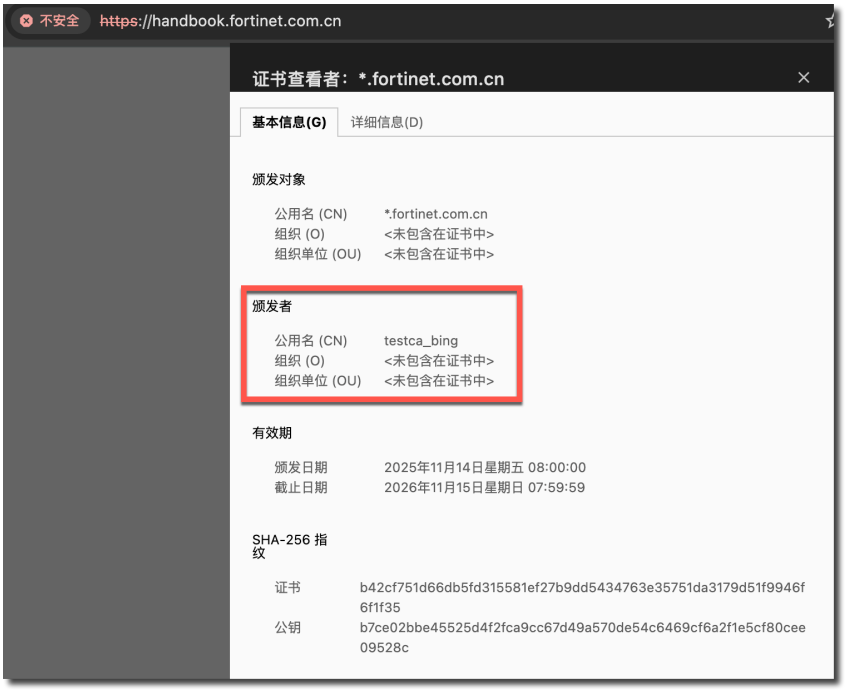

```bash
config firewall ssl-ssh-profile
    edit "test-certificate-inspection"
        config https
            set ports 443 8443
            set status certificate-inspection
            set quic bypass
```

```
            set encrypted-client-hello allow
        end
        config ftps
            set status disable
        end
        config imaps
            set status disable
        end
        config pop3s
            set status disable
        end
        config smtps
            set status disable
        end
        config ssh
            set ports 22
            set status disable
        end
        config dot
            set status disable
            set quic inspect
        end
        set caname "testca_bing"
    next
end
```

3. 保存配置，并在防火墙策略中调用该 SSL/SSH 检测配置文件。

```bash
config firewall policy
    edit 1
        set name "to_Internet1"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "test-certificate-inspection"
        set webfilter-profile "default"
        set nat enable
    next
end
```

4. 随后客户端访问 HTTPS 443 或 8443 端口的网站时，FortiGate 的阻断页面证书会被替换成自定义 CA 证书。



其他选项

不支持 SSL 版本的处理策略

当客户端或服务器尝试使用 FortiGate 无法识别的 SSL/TLS 版本协商时的动作，默认为 block。

```shell
config firewall ssl-ssh-profile
    edit <name>
        config {ssl | https | ftps | imaps | pop3s | smtps | dot}
            set unsupported-ssl-version {allow | block}
        end
    next
end
```

检测所有端口



为优化系统资源，FortiGate 支持针对特定协议启用或禁用协议端口映射。每种协议都有默认的 TCP 端口，管理员可以根据实际需要修改端口配置，以便对非标准端口上的加密流量进行检测。FortiGate 会根据数据包头部信息识别协议类型。支持的协议包括：

- HTTPS
- SMTPS
- POP3S
- IMAPS
- FTPS
- DNS over TLS（DoT）

“检测所有端口”的实际效果，取决于防火墙检测模式（flow/proxy）和 SSL 检查深度（certificate-inspection/deep-inspection）的配置。

相关信息

- 防火墙策略的检测模式介绍可参考：故障排查 → 数据包处理流程 → 安全检测模式 ↗章节。
- 在 Flow-based + deep-inspection 检测模式下，FortiGate 会对所有端口进行检测，而不受端口映射配置影响。

策略检测模式/证书检测模式	检测所有端口状态	行为
Proxy + deep-inspection	禁用	仅扫描“协议端口映射”中指定端口
Proxy + deep-inspection	启用	扫描所有端口
Flow + deep-inspection	启用/禁用	扫描所有端口
certificate-inspection	禁用	仅扫描“协议端口映射”中指定端口（Flow/Proxy 一致）
certificate-inspection	启用	扫描所有端口（Flow/Proxy 一致）

deep-inspection

> 注意
>
> 基于以下限制，大部分常见场景下更推荐使用 certificate-inspection 模式，而不是 deep-inspection：
>
> - 对性能消耗更大，并可能影响某些对证书绑定/校验严格的应用（需通过白名单/例外策略保障兼容性）。
> - 需要每个终端信任 CA 证书，否则可能出现证书告警或连接失败。
> - 要求“中间人解密”与重签名的证书必须是 CA 证书或 Sub CA 证书，不能使用用户证书。

功能简介

深度检测（deep-inspection）是 FortiGate 在 SSL/TLS 加密流量场景下提供的一种安全检测机制。通过对加密流量进行解密、检测和重新加密，FortiGate 能够识别并阻断隐藏在加密通道中的恶意行为，从而避免攻击绕过传统的安全防护措施。

深度检测支持 HTTPS 以及多种基于 SSL/TLS 的常见协议，例如 SMTPS、POP3S、IMAPS 和 FTPS。

能力与限制：

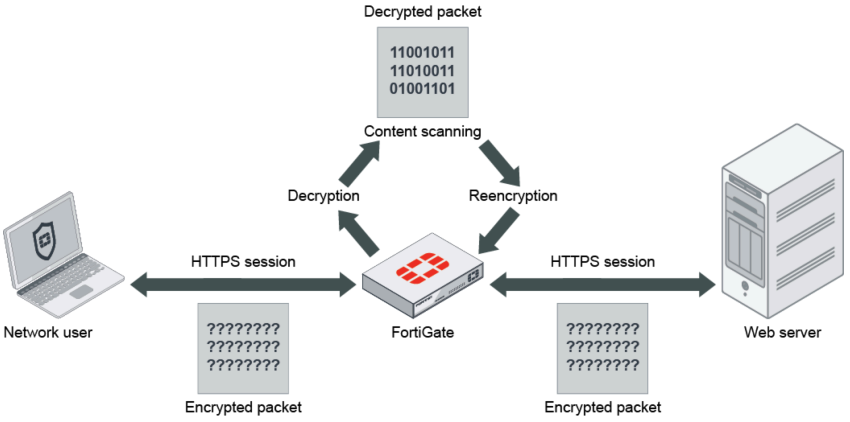
- 可对加密流量应用 AV/IPS/Web Filter/DLP 等内容级检测，安全能力最强。
- 对性能消耗更大，并可能影响某些对证书绑定/校验严格的应用（需通过白名单/例外策略保障兼容性）。
- 需要每个终端信任 CA 证书，否则可能出现证书告警或连接失败。
- 要求“中间人解密”与重签名的证书必须是 CA 证书或 Sub CA 证书，不能使用用户证书。

工作模式

虽然 HTTPS 等加密协议为互联网通信提供了隐私保护，但攻击者同样可以利用加密流量隐藏恶意内容，例如：

- 在加密的电子商务会话中下载携带病毒的文件。
- 钓鱼邮件中的“看似正常”附件，在本地执行后建立到 C & C（Command and Control）服务器的加密连接。
- 通过加密通道下载或回传恶意程序数据。

由于上述通信过程均处于加密状态，如果不进行解密检测，相关攻击可能绕过网络安全设备的防护。



启用深度检测后，FortiGate 会执行以下操作：

- 模拟目标服务器，与客户端建立 SSL/TLS 会话。
- 解密并检查流量内容，识别威胁并执行放通/阻断。
- 重新加密流量，再转发至真实的目标服务器。

证书告警提示

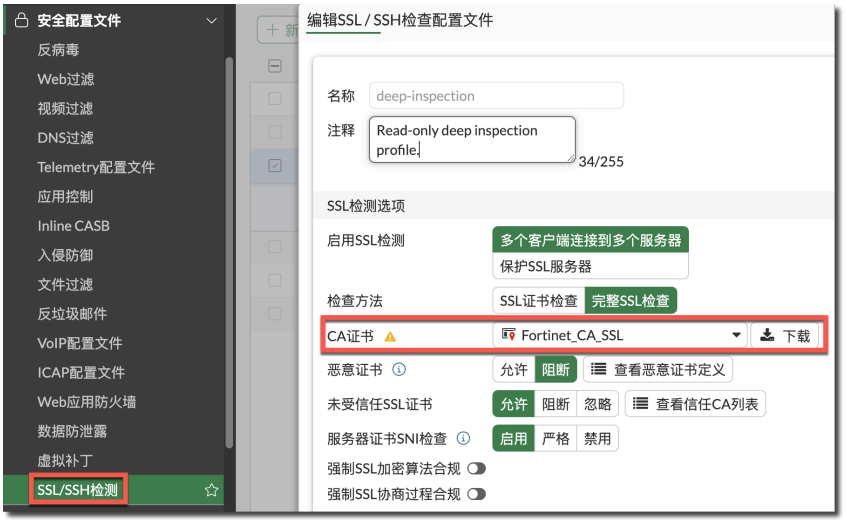
在重新加密流量时，FortiGate 会使用本地存储的 CA 证书进行签名，例如：Fortinet_CA_SSL、Fortinet_CA_Untrusted、管理员自行上传的 CA 证书。

在应用 deep-inspection 的防火墙策略中，由于客户端浏览器默认不信任这些内置/自定义 CA 证书，客户端在访问时会收到“证书不受信任”的安全警告（不代表配置错误，点击“继续前往”仍可以显示目标网站）。



消除证书告警

为避免客户端出现访问网站时的 HTTPS 证书警告，需要在终端设备上安装 deep-inspection 配置文件使用的 CA 证书并信任。获取 CA 证书步骤如下：



1. 进入安全配置文件 → SSL/SSH 检测。
2. 编辑当前防火墙策略所使用的 SSL/SSH 检测配置文件，如上所示为系统预置的 deep-inspection 配置文件。
3. 在 CA 证书右侧点击“下载”按钮。

> **提示**
>
> - 内置的 certificate-inspection 配置文件默认使用的 CA 为 Fortinet_CA_SSL，无法修改。
> - 可在设备证书库中导入自定义 CA 证书（参考：系统管理 → 系统设置 → 证书管理↗章节，必须导入带私钥的 CA 证书或 SubCA 证书，不能使用用户证书），并克隆或新建 SSL/SSH 检测配置文件调用自定义 CA 证书。
> - 导入的自定义 CA 证书位于“本地证书”分类中。

4. 将下载的 CA 证书加入客户端系统或者浏览器的信任列表。

重要

- Chrome/Edge 依赖系统的证书信任列表。
- 而 Firefox 有自己的信任列表，需要在浏览器设置中添加，或在浏览器的证书设置中开启“允许 Firefox 自动信任您安装的第三方证书”，开启后 Firefox 也会信任系统的证书信任列表。



Windows

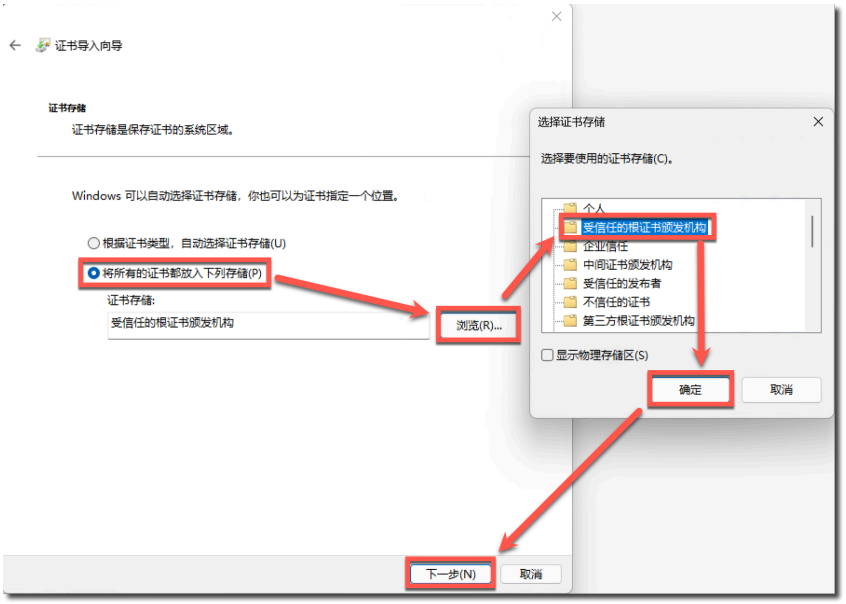
1. 双击下载的 CA 证书，点击“安装证书 (I)...”。



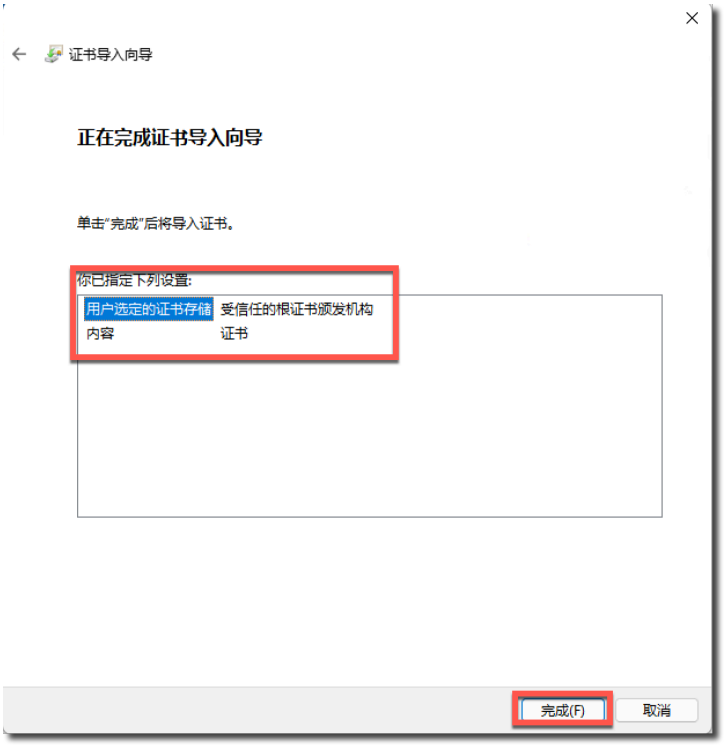
2. 存储位置选择“本地计算机 (L)”，然后点击下一步。



3. 选择“将所有的证书都放入下列存储 (P)”，点击“浏览 (R)...”按钮，选择“受信任的根证书颁发机构”。



4. 完成证书导入。



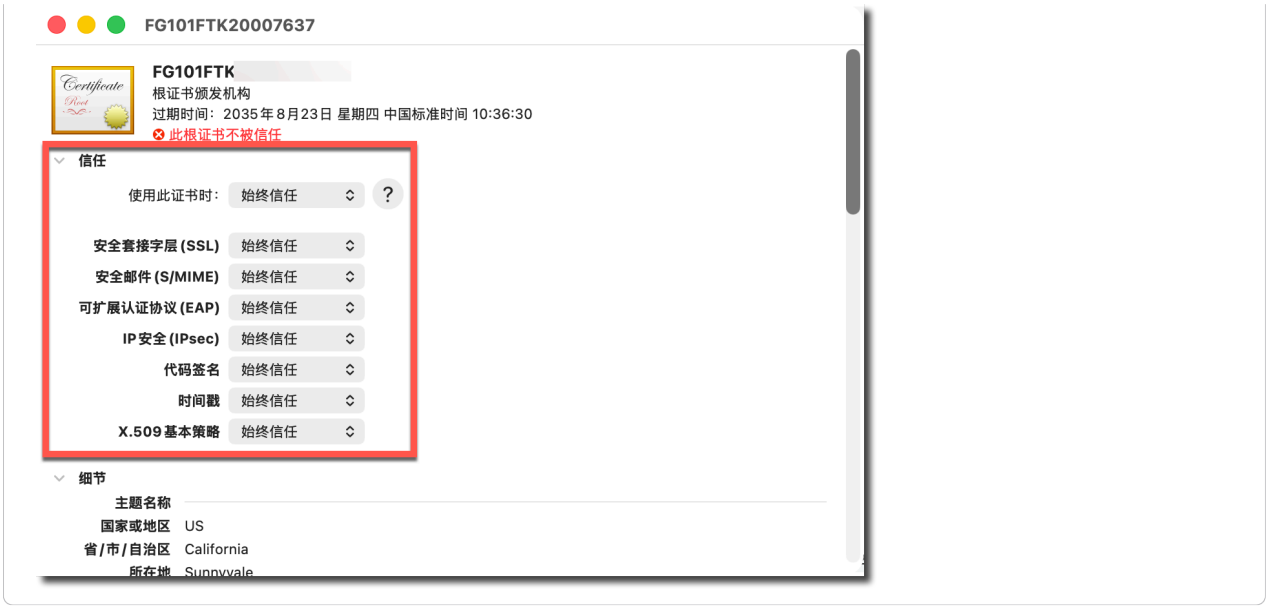
macOS

1. 双击下载的 CA 证书，会打开“钥匙串访问”应用。

2. 进入“系统 → 证书”页面，可以看到刚刚导入的 CA 证书，然后双击该证书。



3. 在“使用此证书时”选项中选择“始终信任”，关闭窗口（会提示输入系统密码）。



5. 客户端再次访问该网站时，会直接显示出目标网页，页面显示连接安全，不会弹出证书告警页面。

> **提示**
>
> 同时诸如 Web Filter 阻断页面的 FortiGate 替换页面的证书告警也会消失，参考：UTM → SSL/SSH 检查 → certificate-inspection ↗ 章节。

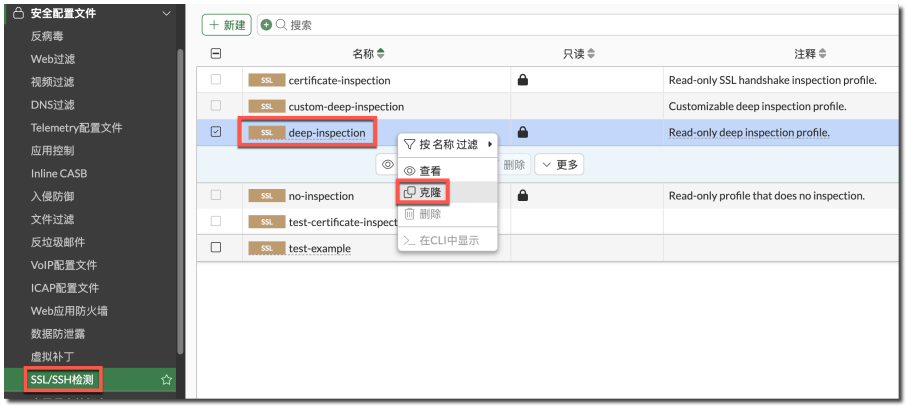


自定义 deep-inspection 配置文件

默认的 deep-inspection 配置文件为只读，无法修改任何配置（使用出厂自带的 Fortinet_CA_SSL 作为 CA 证书等），当有如下配置需求（包含但不限），必须新建或克隆一个证书检查配置文件：

- 需要使用自定义的 CA 证书。
- 其他自定义参数，如针对无效 SSL 证书配置处理方式，例如允许、阻断、根据不同证书错误类型设置不同动作等，请参考：UTM → SSL/SSH 检查 → SSL/SSH 检测配置文件 ↗ 章节。

1. 进入安全配置文件 → SSL/SSH 检测，新建或克隆 deep-inspection 配置文件。



2. 在“SSL 检测选项”中配置自定义的 CA 证书，保存配置。

> **提示**
>
> - 可在设备证书库中导入自定义 CA 证书（参考：系统管理 → 系统设置 → 证书管理 ↗章节，必须导入带私钥的 CA 证书或 SubCA 证书，不能使用用户证书），并克隆或新建 SSL/SSH 检测配置文件调用自定义 CA 证书。
> - 导入的自定义 CA 证书位于“本地证书”分类中。



3. 对应的 SSL/SSH 检查配置文件 CLI 命令如下。

```bash
config firewall ssl-ssh-profile
    edit "test-deep-inspection"
        config https
            set ports 443
            set status deep-inspection
            set quic inspect
        end
        config ftps
            set ports 990
            set status deep-inspection
        end
        config imaps
            set ports 993
            set status deep-inspection
        end
        config pop3s
            set ports 995
            set status deep-inspection
```

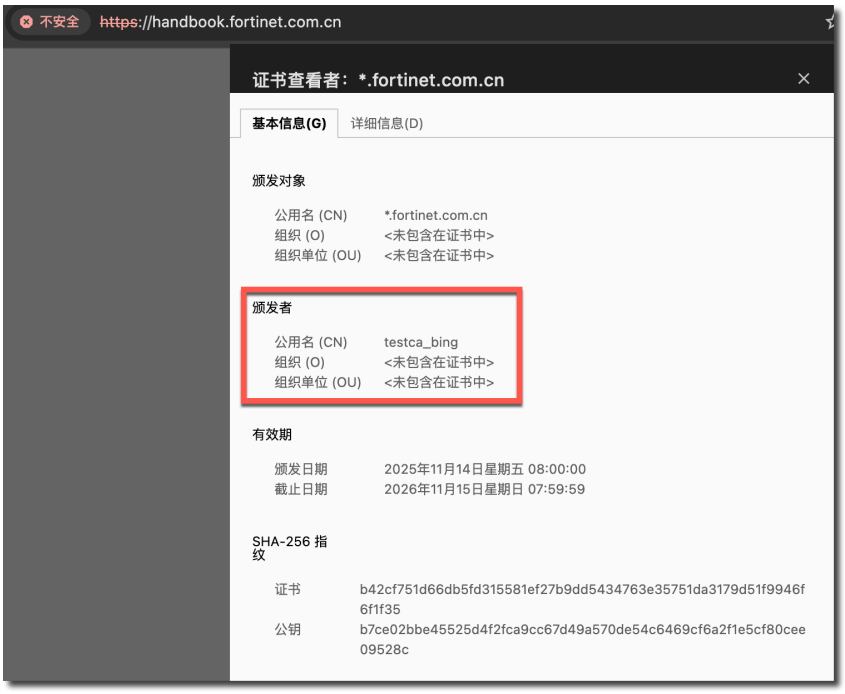
```
        end
        config smtps
            set ports 465
            set status deep-inspection
        end
        config ssh
            set ports 22
            set status disable
        end
        config dot
            set status disable
            set quic inspect
        end
        set caname "testca_bing"
    next
end
```

4. 在防火墙策略中调用该 SSL/SSH 检测配置文件。

> **提示**
>
> 使用 deep-inspection 类型的配置文件时，必须在防火墙策略中至少调用一个 UTM 配置文件，否则深度检测功能不会生效。

```bash
config firewall policy
    edit 1
        set name "to_Internet1"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "test-deep-inspection"
        set webfilter-profile "default"
        set nat enable
    next
end
```

5. 随后客户端访问 HTTPS 443 或 8443 端口的网站时，FortiGate 的阻断页面证书会被替换成自定义 CA 证书。



其他选项

免除 SSL 检查

> **提示**
>
> 此功能仅支持 deep-inspection 模式。

在某些场景下，可能因隐私或合规要求需要跳过深度检测。

在“免除 SSL 检查”中配置需要跳过深度检测的内容，可免除以下类型的目标地址（按需开启“记录 SSL 免除日志”，相关的日志可以在“日志 & 报表 → 安全事件 → 日志”下的 SSL 分类中查看）：



- Web Filter 分类。

  > **提示**
  >
  > - 默认配置下，在执行“免除 SSL 检查”功能时，网站域名与其 IP 地址会同时被 FortiGuard 评级，且返回的分类结果不一致时，FortiGate 会优先采用 IP 地址的分类结果。
  > - 可以通过如下 CLI 关闭 IP 地址评级，在执行“免除 SSL 检查”功能时，FortiGate 不会再使用 IP 分类结果，仅基于域名的 FortiGuard 分类进行判断。
  > - 该命令只针对“免除 SSL 检查”功能生效。
  >
  > ```bash
  > config firewall ssl-ssh-profile
  >     edit <profile-name>
  >         set ssl-exemption-ip-rating disable
  >     next
  > end
  > ```

- IP/IP 范围/IP 子网/地址位置/FQDN/通配符 FQDN。
- 启用“信誉良好的网站”，可跳过 FortiGuard 认定的常见可信网站。

```bash
config firewall ssl-ssh-profile
    edit "test-deep-inspection"
        config ssl-exempt
            edit 1
                set type address
                set address "10.10.1.1-10.10.1.100"
            next
            edit 2
                set type address
                set address "192.168.100.0/24"
            next
            edit 3
                set type address
                set address "China"
            next
```

```
            edit 4
                set type address
                set address "handbook.fortinet.com.cn"
            next
            edit 5
                set type wildcard-fqdn
                set wildcard-fqdn "*.fortilab.com"
            next
            edit 6
                set fortiguard-category 33
            next
            edit 7
                set fortiguard-category 31
            next
        end
        set ssl-exemption-log enable
    next
end
```

SSL/TLS 版本检测

> **提示**
>
> 此功能仅支持 deep-inspection 模式。

FortiGate 提供两种方式限制深度检测适用的 SSL/TLS 版本。

- 全局配置：启用后，系统默认将最低允许的 SSL/TLS 版本设置为 TLS 1.1（默认开启）。

```bash
config system global
    set strong-crypto enable
end
```

- 深度检测配置文件配置：当客户端或服务器尝试使用低于最低允许版本的 SSL/TLS 时阻断会话（默认为 tls-1.0）。

> **提示**
>
> 当检测模式为 certificate-inspection 时，无法设置 `min-allowed-ssl-version`，因为此模式下 FortiGate 不会代理与服务器协商具体的 SSL/TLS 版本。

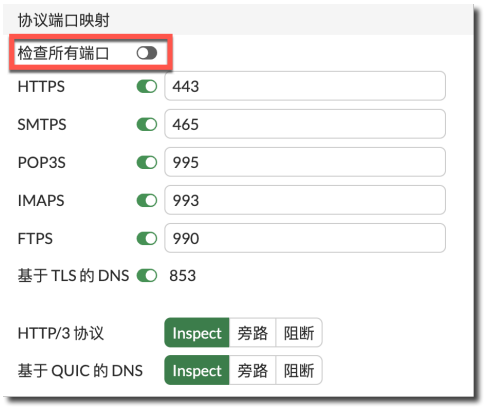
```bash
config firewall ssl-ssh-profile
    edit <name>
        config {ssl | https | ftps}
            set min-allowed-ssl-version {ssl-3.0 | tls-1.0 | tls-1.1 | tls-1.2 | tls-1.3}
        end
    next
end
```

不支持 SSL 版本的处理策略

当客户端或服务器尝试使用 FortiGate 无法识别的 SSL/TLS 版本协商时的动作，默认为 block。

```shell
config firewall ssl-ssh-profile
    edit <name>
        config {ssl | https | ftps | imaps | pop3s | smtps | dot}
            set unsupported-ssl-version {allow | block}
        end
    next
end
```

检测所有端口



为优化系统资源，FortiGate 支持针对特定协议启用或禁用协议端口映射。每种协议都有默认的 TCP 端口，管理员可以根据实际需要修改端口配置，以便对非标准端口上的加密流量进行检测。FortiGate 会根据数据包头部信息识别协议类型。支持的协议包括：

- HTTPS
- SMTPS
- POP3S
- IMAPS
- FTPS
- DNS over TLS（DoT）

“检测所有端口”的实际效果，取决于防火墙检测模式（flow/proxy）和 SSL 检查深度（certificate-inspection/deep-inspection）的配置。

相关信息

- 防火墙策略的检测模式介绍可参考：故障排查 → 数据包处理流程 → 安全检测模式 ↗章节。
- 在 Flow-based + deep-inspection 检测模式下，FortiGate 会对所有端口进行检测，而不受端口映射配置影响。

策略检测模式/证书检测模式	检测所有端口状态	行为
Proxy + deep-inspection	禁用	仅扫描“协议端口映射”中指定端口
Proxy + deep-inspection	启用	扫描所有端口
Flow + deep-inspection	启用/禁用	扫描所有端口
certificate-inspection	禁用	仅扫描“协议端口映射”中指定端口（Flow/Proxy 一致）
certificate-inspection	启用	扫描所有端口（Flow/Proxy 一致）

保护 SSL 服务器

功能简介

保护 SSL 服务器（Protecting SSL Server）是 SSL/SSH 检测配置文件中的一种工作模式，主要用于入站（Inbound）HTTPS 流量的解密与安全检测。

该功能通常应用在互联网客户端访问位于 FortiGate 后端的真实服务器的场景中（反向代理），通常在映射真实服务器的 VIP 策略中调用，例如：

- 公网用户访问内网 Web Server。
- 反向代理/发布服务器场景。
- 对外提供 HTTPS 服务的业务系统。

工作模式

在启用“保护 SSL 服务器”模式后，FortiGate 的工作方式如下：

1. FortiGate 使用真实服务器的 SSL 证书。
2. FortiGate 模拟真实服务器，与客户端建立 SSL 会话：
   - FortiGate 并不使用自签名 CA 证书，而是直接向客户端发送真实服务器的证书。
   - 如果使用的真实服务器证书是公网权威 CA 颁发的证书（或客户端手动信任了该证书），客户端不会出现证书不信任提示。
   - 真实服务器可以是 HTTP（实现 SSL 卸载），也可以是 HTTPS。
3. FortiGate 解密 HTTPS 流量。
4. 对解密后的流量执行安全检测（如 IPS、AV 等）。
5. 再将流量转发给真实服务器（反向代理）。

适用场景

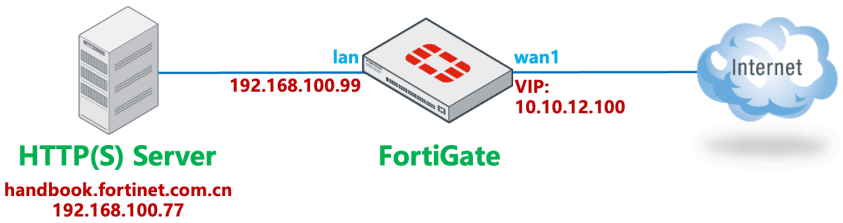
“保护 SSL 服务器”模式适用于以下场景：

- 需要对入站 HTTPS 流量进行深度安全检测（如 IPS、AV 等）。
- 服务器证书（需要携带私钥，如 PFX/PKCS#12 格式）可以导出并部署在 FortiGate 上。
- 希望客户端看到的证书与服务器真实证书一致。
- 业务允许 FortiGate 作为 SSL 解密终端。

不适用场景：

- 服务器证书无法导出。
- 不允许中间设备接触服务器私钥。
- 仅需基础证书检查、不希望解密流量。

网络拓扑



- FortiGate 通过 VIP 策略发布 HTTPS 网站 `https://handbook.fortinet.com.cn`。
- HTTPS 网站的 SSL 证书安装在 FortiGate 上，并对访问 VIP 的 HTTPS 流量做 IPS 检查。
- 客户端的 DNS 服务器将 `handbook.fortinet.com.cn` 解析到 FortiGate 的 VIP 地址。
- 客户端通过 Internet 访问 `handbook.fortinet.com.cn`，看到的网站 SSL 证书为 FortiGate 提供。

配置方法

前置条件

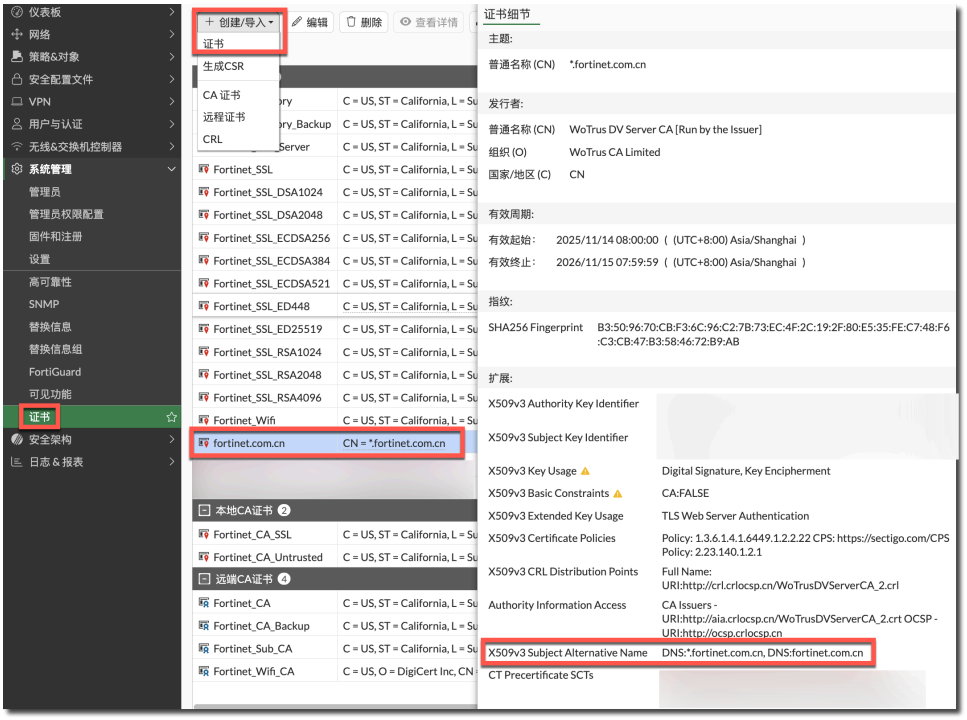
- 已获取服务器的 SSL 证书文件，通常包含：
  - 证书（Certificate）。
  - 私钥（Private Key），PFX/PKCS#12 格式的公钥和私钥在一个文件中。

- 可能还包含中间证书链。
- FortiGate 具备导入本地证书的权限。
- 该证书 SAN 与服务器域名匹配。

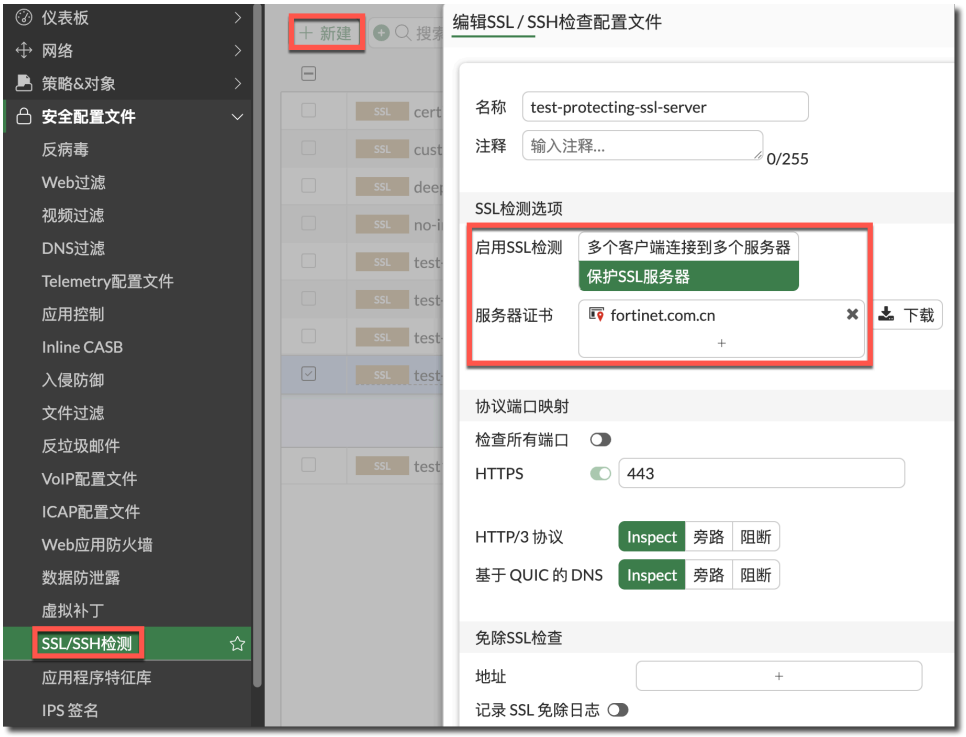
配置步骤

1. 导入服务器的 SSL 证书（含私钥，如 PFX/PKCS#12 格式，步骤参考：系统管理 → 系统设置 → 证书管理 ↗章节）。

> **重要**
>
> - 网站证书的 SAN（Subject Alternative Name）字段需要是网站真实的 FQDN（也可以是通配符方式）。否则访问时会产生证书告警。如下所示，网站 `handbook.fortinet.com.cn` 所使用的网站 SSL 证书的 SAN 为 `*.fortinet.com.cn`（通配符方式）。
> - 网站的 SSL 证书需要是公网权威 CA 颁发的证书（或客户端手动信任了该证书），否则客户端访问时会出现证书不信任提示。



2. 创建或编辑 SSL/SSH 检测配置文件，在“启用 SSL 检测”处选择“保护 SSL 服务器”，在“服务器证书”处选择上步导入的服务器证书，保存配置。




```bash
config firewall ssl-ssh-profile
    edit "test-protecting-ssl-server"
        config https
            set ports 443
            set status deep-inspection
            set quic inspect
        end
        config ftps
            set status disable
        end
        config imaps
            set status disable
        end
        config pop3s
            set status disable
        end
        config smtps
            set status disable
        end
        config ssh
            set ports 22
            set status disable
        end
        config dot
            set status disable
            set quic inspect
        end
        set server-cert-mode replace
        set server-cert "fortinet.com.cn"
    next
end
```

3. 在 Web 服务器的 VIP 映射策略中应用上步创建的 SSL/SSH 检测配置文件，

```bash
config firewall vip
    edit "Web_Server_VIP"
        set extip 10.10.12.100
        set mappedip 192.168.100.77
        set extintf wan1
        set portforward enable
        set extport 443
        set mappedport 443
    next
```

```
end

config firewall policy
    edit 7
        set name "Web_Server_SSL_Protect"
        set srcintf "wan1"
        set dstintf "lan"
        set action accept
        set srcaddr "all"
        set dstaddr "Web_Server_VIP"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "test-protecting-ssl-server"
        set ips-sensor "default"
    next
end
```

4. 客户端通过 FortiGate 的 VIP 访问 `https://handbook.fortinet.com.cn`，网站 SSL 证书由 FortiGate 提供，未出现证书告警。



5. 相关流量日志在“本地流量”分类中显示，源 IP 为 FortiGate 的 wan1 接口 IP。

> **提示**
>
> - 上步中引用 VIP 的防火墙策略 ID 为 7，从配置上看客户端是通过防火墙策略 + VIP 的方式访问 `handbook.fortinet.com.cn`，直觉上相关日志应该在“转发流量”分类中显示。
> - 但实际上相关流量日志在“本地流量”分类中显示，这说明 FortiGate 是使用反向代理的方式访问真实服务器。

```bash
date=2026-01-26 time=16:31:42 eventtime=1769416301197951804 tz="+0800" logid="0001000014" type="traffic" subtype="local"
level="notice" vd="root" srcip=192.168.100.99 srcport=23235 srcintf="root" srcintfrole="undefined" dstip=192.168.100.77
dstport=443 dstintf="lan" dstintfrole="lan" srccountry="China" dstcountry="China" sessionid=3629352 proto=6
action="close" policyid=0 service="HTTPS" trandisp="noop" app="HTTPS" duration=5 sentbyte=3076 rcvdbyte=19676 sentpkt=28
rcvdpkt=24
```

常见问题

1. 客户端是否需要安装 CA 证书。

   在“保护 SSL 服务器”模式下，SSL/SSH 检测配置文使用的是服务器真实证书，不是自签名 CA 证书。如果使用的真实服务器证书是公网权威 CA 颁发的证书（或客户端手动信任了该证书所属的 CA 证书），客户端不会出现证书不信任提示。

2. 会不会影响服务器端 SSL 配置?

   FortiGate 只是作为中间解密设备，不会修改服务器自身的 SSL 配置。

3. 私钥安全风险?

   是的，这是需要重点评估的风险点：

- 服务器私钥需要存放在 FortiGate 上。
- 建议限制 FortiGate 管理权限、定期更换证书、明确安全合规要求后再启用。

4. 与其他 SSL 检测模式有什么区别?

模式	是否解密	使用证书	典型场景
certificate-inspection	否	不涉及	仅做证书校验
deep-inspection	是	FortiGate CA 或自定义 CA	出站 HTTPS 解密
保护 SSL 服务器	是	服务器真实 SSL 证书	入站 HTTPS 保护

- 保护 SSL 服务器：对外发布 HTTPS 服务（如 VIP 策略），且需要对加密内容做安全检测。
- 多个客户端连接到多个服务器：
  - certificate-inspection：只需基础证书检查、不想解密。
  - deep-inspection：内部用户访问互联网，需要对加密内容做安全检测。

VPN

IPSec VPN

- IPSec VPN 原理
- IPSec DPD 模式
- IPSec 重协商机制
- 自动协商与 keepalive
- 点到点 VPN
  - IPSec 的两种配置模式
  - 接口模式
  - PKI 证书认证
  - VPN 隧道上运行 OSPF 或 BGP
  - 多路负载/主备 IPSec VPN
  - FortiGuard DDNS 方式建立 IPSec VPN
  - GRE over IPSec
  - 修改 IPSec 协商端口
- 拨号 VPN
  - Hub-and-spoke 模式
  - 拨号 VPN 运行动态路由
  - ADVPN 模式
  - 野蛮模式/network-id 适用场景
  - FortiClient 拨号
  - FortiClient 使用证书认证连接 IPSec
  - 基于 TCP 的 FortiClient 拨号
  - IPSec 多用户组访问控制
- 透明模式下的 IPSec VPN
- 重命名 IPSec 隧道
- 与友商进行 IPSec VPN 对接
  - 与友商对接 IPSec 的注意事项
  - 与阿里云建立 IPSec VPN
  - 与思科路由器建立 IPSec VPN (IKE v1)
  - 与思科路由器建立 IPSec VPN (IKE v2)
  - 与思科 ASA 防火墙建立 IPSec VPN (IKE v1)
  - 与思科 ASA 防火墙建立 IPSec VPN (IKE v2)
  - 与华为 USG6000V 防火墙建立 IPSec VPN
  - 与 H3C SecPath 防火墙建立 IPSec VPN
  - 与山石 SG6000 防火墙建立 IPSec VPN
  - 与 Palo Alto 防火墙建立 IPSec VPN
  - 与 Check Point 防火墙建立 IPSec VPN
  - 与 Juniper SRX 防火墙建立 IPSec VPN
  - 与 Openswan 建立 IPSec VPN
  - 与 strongSwan 建立 IPsec VPN
- IPSec VPN 排错
  - IPSec 相关诊断命令
  - IPSEC IKEv1 主模式 Debug 分析
  - IPSec IKEv1 野蛮模式 Debug 分析
  - IPSEC IKEv2 Debug 分析
  - IPSEC 协商失败举例
  - 使用附加 IP 时隧道无法建立
  - 多个拨号隧道使用相同 IP 池
  - IPSec 命名注意事项
  - IPSec 业务不通问题
  - ESP 分片丢包
  - IPSec 无法修改隧道类型
  - Hub-Spoke 双线路隧道抖动
  - Hub-Spoke 静态路由问题
  - Hub 反向流量异步
  - IPsec 绑定 Loopback 时无法建立
  - IPSec 解密

SSL VPN

- SSL VPN 的两种模式
- 开启 SSL VPN
- SSL VPN FortiClient 隧道模式
- SSL VPN 分配 IP 的方式
- SSL VPN WEB 代理模式配置
- SSL VPN 数字证书认证
- SSL VPN 双因子认证
  - FortiToken 双因子认证
- SSL VPN 源地址连接控制
- 解锁 SSL VPN 用户
- DTLS
- SSL VPN 常见问题
  - SSL VPN 连接后断开

SSL VPN 迁移到 IPSec VPN

- 介绍
- 迁移背景
  - 安全性对比
  - IKEv1 还是 IKEv2?
  - 隧道协议与封装方式
- 迁移基础
  - 设计考虑事项
  - FortiOS SSL VPN 到拨号 VPN 的迁移
  - FortiClient 终端配置迁移

FortiClient VPN

- FortiClient 下载
- 系统兼容性列表
- Windows FortiClient 安装及使用
- mac FortiClient 安装及使用
- iPhone FortiClient 安装及使用
- Android FortiClient 安装及使用
- FortiClient 连接问题
- FortiClient 排错与卸载

系统原生客户端接入

- iOS

  iOS 原生客户端接入 IPSec（IKEv1） IKEv2
- Android 系统

  Android12 以下系统（IKEv1） Android12 或以上系统（IKEv2）
- macOS

  macOS 原生客户端接入 IPSec（IKEv1） IKEv2
- Windows 系统

  Windows 原生客户端接入 IPSec IKEv1（L2TP over IPSec） Windows 原生客户端接入 IPSec（IKEv2）

  使用 Windows 系统防火墙与 FortiGate 建立 IPSec VPN L2TP VPN PPTP VPN

其他 VPN 技术

- GRE

IPSec VPN

IPSec VPN 原理

IPSec DPD 模式

IPSec 重协商机制

自动协商与 keepalive

点到点 VPN

- IPSec 的两种配置模式
- 接口摸式
- PKI 证书认证
- VPN 隧道上运行 OSPF 或 BGP
- 多路负载/主备 IPSec VPN
- FortiGuard DDNS 方式建立 IPSec VPN
- GRE over IPSec
- 修改 IPSec 协商端口

拨号 VPN

- Hub-and-spoke 模式
- 拨号 VPN 运行动态路由
- ADVPN 模式
- 野蛮模式/network-id 适用场景
- FortiClient 拨号
- FortiClient 使用证书认证连接 IPSec
- 基于 TCP 的 FortiClient 拨号
- IPSec 多用户组访问控制

透明模式下的 IPSec VPN

重命名 IPSec 隧道

与友商进行 IPSec VPN 对接

- 与友商对接 IPSec 的注意事项
- 与阿里云建立 IPSec VPN
- 与思科路由器建立 IPSec VPN (IKE v1)
- 与思科路由器建立 IPSec VPN (IKE v2)
- 与思科 ASA 防火墙建立 IPSec VPN (IKE v1)
- 与思科 ASA 防火墙建立 IPSec VPN (IKE v2)
- 与华为 USG6000V 防火墙建立 IPSec VPN
- 与 H3C SecPath 防火墙建立 IPSec VPN
- 与山石 SG6000 防火墙建立 IPSec VPN
- 与 Palo Alto 防火墙建立 IPSec VPN
- 与 Check Point 防火墙建立 IPSec VPN
- 与 Juniper SRX 防火墙建立 IPSec VPN
- 与 Openswan 建立 IPSec VPN
- 与 strongSwan 建立 IPsec VPN

IPSec VPN 排错

- IPSec 相关诊断命令
- IPSEC IKEv1 主模式 Debug 分析
- IPSec IKEv1 野蛮模式 Debug 分析
- IPSEC IKEv2 Debug 分析
- IPSEC 协商失败举例
- 使用附加 IP 时隧道无法建立
- 多个拨号隧道使用相同 IP 池
- IPSec 命名注意事项
- IPSec 业务不通问题

- ESP 分片丢包
- IPSec 无法修改隧道类型
- Hub-Spoke 双线路隧道抖动
- Hub-Spoke 静态路由问题
- Hub 反向流量异步
- IPsec 绑定 Loopback 时无法建立
- IPSec 解密

IPSec VPN 原理

什么是 IPSEC VPN

IPSec 全称为 Internet Protocol Security，是由 Internet Engineering Task Force (IETF) 定义的安全标准框架，通过在公网上为两个或多个私有网络之间建立 IPsec 隧道，并通过加密和验证算法保证 VPN 连接的安全。IPSEC 协议工作在 IP 层，在 IP 层对数据包进行加密和验证：

- 数据的机密性：IPSec 的发送方对发给对端的数据进行加密；
- 数据的完整性：IPSec 的接收方对接收到的数据进行验证以保证数据在传送的过程中没有被修改；
- 数据来源的认证：IPSec 接收方验证数据的起源；
- 抗重播：IPSec 的接收方可以检测并拒绝到重播的 IP 包；

SA 安全关联

SA（security association）安全关联分为两种：ISAKMP/IKE SA 和 IPsec SA。

ISAKMP/IKE SA：对等体之间的 ISAKMP/IKE SA 用于控制 IPsec 协商数据流，如协商对 IKE 数据流进行加密以及对对等体进行验证的算法，确保协商过程中的安全。

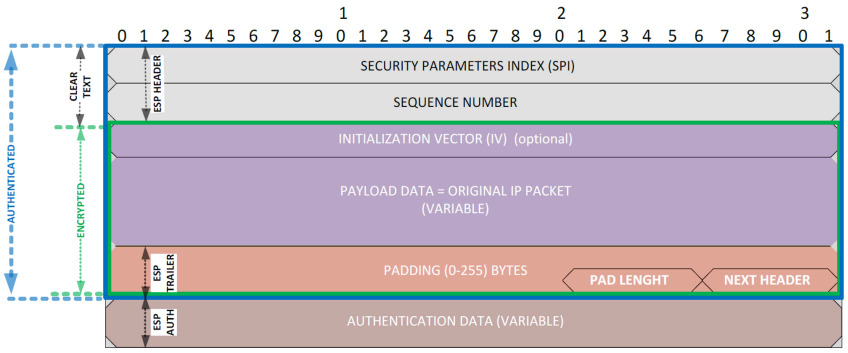
IPsec SA：IPSec SA 用于协商对等体之间真正需要通信的 IP 数据流进行加密的算法，IPSec SA 是单向的，因此至少需要两个: 一个用于入站数据流，另一个用于出站数据流。

IPSEC 安全协议

FortiGate IPSEC VPN 只使用 ESP，不支持 AH。

ESP：ESP（Encapsulating Security Payload），协议号 50，支持认证和加密功能。ESP 在每一个数据包的标准 IP 报头后面添加一个 ESP 报文头，并在数据包后面追加一个 ESP 尾（ESP Trailer 和 ESP Auth data）。

ESP 报文结构如下图：



- **SPI**：安全参数索引，用于标识 IPSEC SA；
- **SEQUENCE NUMBER**：序列号，长度 32bit，从 1 开始的递增的计数器，唯一地标识每一个数据包，用于防止重放攻击；
- **INITIALIZATION VECTOR (IV)**：初始向量，一些算法如 AES - GCM、AES - CCM 和 ChaCha20-Poly1305，采用 IV 生成一个随机数，用作加密和解密的输入参数；
- **PAYLOAD DATA**：负载数据，原始包中可变长的数据内容；
- **PADDING**：填充字段，需要加密的明文长度（也就是 ESP 报文的结构中，IV + Payload Data + Padding + Pad Length + Next Header 的长度）必须是块长度的整数倍，事实是无法保证最后一个块的大小正好为块长度要求，因此信息必须填充至满足要求。只要是使用对称密钥的块密码工作模式，均可能需要 Padding 字段进行填充；
- **PAD LENGHT**：填充字段的长度，0 表示没有填充；
- **NEXT HEADER**：下一头部，标识 ESP 报文头后的负载类型。传输模式下，是被保护的上层协议（TCP 或 UDP）的编号；隧道模式下，是 IP 协议的编号；
- **AUTHENTICATION DATA**：认证数据，包含数据完整性校验值 ICV，用于接收方进行完整性校验；

ESP 报文抓包示例：

```
No.  Time       Source          Destination     Protocol  Length  Info
     8.644528   192.168.89.34   192.168.89.35   ESP          166  ESP (SPI=0xe68f324b)
     8.645259   192.168.89.35   192.168.89.34   ESP          166  ESP (SPI=0x51b93fc6)

> Frame 10: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits)
> Ethernet II, Src: Fortinet_ee:e2:fa (e8:1c:ba:ee:e2:fa), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
v Internet Protocol Version 4, Src: 192.168.89.34, Dst: 192.168.89.35
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 152
    Identification: 0x0300 (768)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 63
    Protocol: Encap Security Payload (50)
    Header Checksum: 0x449e [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.89.34
    Destination Address: 192.168.89.35
v Encapsulating Security Payload
    ESP SPI: 0xe68f324b (3868144203)
    ESP Sequence: 2
```

AH：AH（Authentication Header），协议号 51，只提供认证功能，不提供加密功能。

封装模式

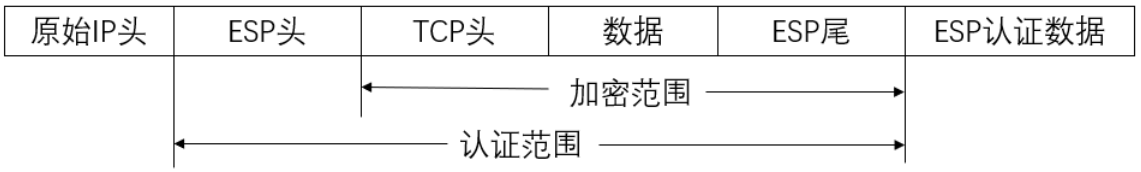
IPSEC 隧道模式

在隧道模式下，ESP 头被插到原始 IP 头之前，重新生成一个新的报文头放到 ESP 头之前，保护 IP 头和负载。隧道模式主要应用于 VPN 网关之间或主机与 VPN 网关之间的通信，用来保护 VPN 网关后面的网络。



IPSEC 传输模式

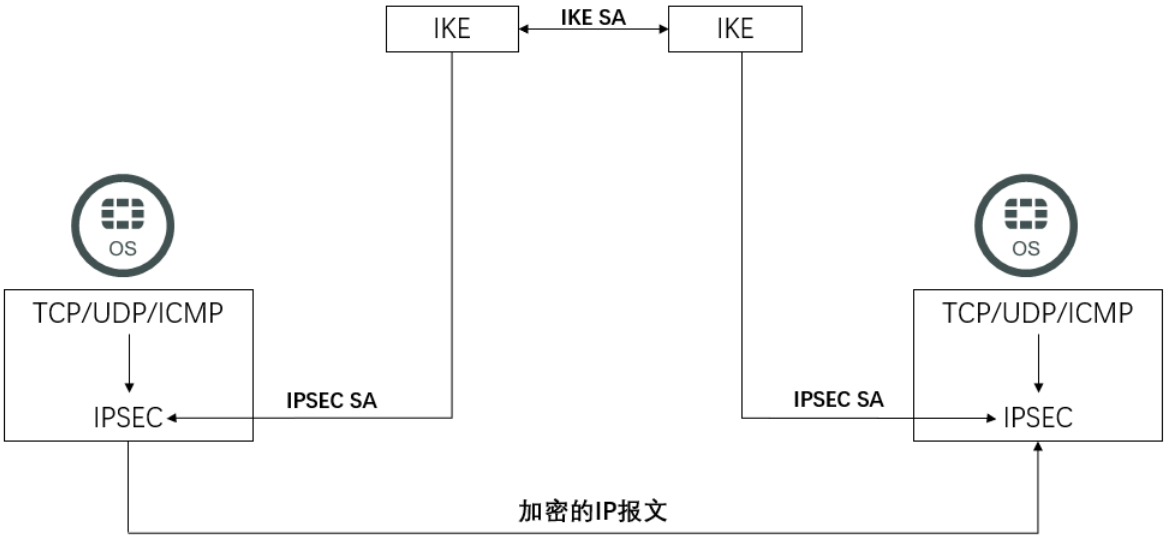
在传输模式下，ESP 头被插入到 IP 头与传输层协议头之间，保护 TCP/UDP/ICMP 的负载。传输模式没有添加额外的 IP 头，因此原始报文中的 IP 地址在加密后报文的 IP 头中是可见的。一般用于 IPsec 加密点等于 IPsec 通信点的环境下或 GRE over IPsec 的环境中。



IKE 协议

IKE 协议与 IPSEC 的关系

因特网密钥交换 IKE（Internet Key Exchange）协议建立在 Internet 安全关联和密钥管理协议 ISAKMP 定义的框架上，是基于 UDP 的应用层协议，分 IKEv1 和 IKEv2 两个版本。它为 IPsec 协商建立 SA，并把协商好的参数交给 IPsec，IPsec 使用 IKE 建立的 SA 对 IP 报文加密或认证。



IKE 的安全机制

- **DH 秘钥交换算法**：DH 是一种公共密钥交换算法，它用于产生密钥材料，并通过 ISAKMP 消息在发送和接收端之间进行密钥材料交换。然后，两端各自计算出完全相同的对称密钥，该对称密钥用于计算加密和验证的密钥。DH 使用密钥组定义自己产生密钥的长度。

DH Group	Description
1	More Modular Exponential (MODP) DH Group with a 768-bit modulus
2	MODP with a 1024-bit modulus
5	MODP with a 1536-bit modulus
14	MODP with a 2048-bit modulus
15	MODP with a 3027-bit modulus
16	MODP with a 4096-bit modulus
17	MODP with a 6144-bit modulus
18	MODP with a 8192-bit modulus
19	256-bit random elliptic curve group
20	384-bit random elliptic curve group
21	521-bit random elliptic curve group
27	Brainpool 224-bit elliptic curve group
28	Brainpool 256-bit elliptic curve group
29	Brainpool 384-bit elliptic curve group
30	Brainpool 512-bit elliptic curve group
31	Curve25519 128-bit elliptic curve group

- **身份认证**

  通过 IP 地址或者名称确定双方的身份，有两种身份认证方式：预共享秘钥和数字证书认证。

  - 在预共享密钥认证中，通信双方采用共享的密钥对报文进行 Hash 计算，判断双方的计算结果是否相同。如果相同，则认证通过；否则认证失败。

- 在数字证书认证中，通信双方使用 CA 证书进行数字证书合法性验证，双方各有自己的公钥（网络上传输）和私钥（本地存储）。发送方对报文进行 Hash 计算，并用自己的私钥对报文计算结果进行加密，生成数字签名。接收方使用发送方的公钥对数字签名进行解密，并对报文进行 Hash 计算，判断计算结果与解密后的结果是否相同。如果相同，则认证通过；否则认证失败。

IKE 支持的认证算法有：MD5、SHA1、SHA2-256、SHA2-384、SHA2-512。（MD5 和 SHA1 认证算法不安全）

- **数据加密**

  身份数据在密钥产生之后加密传送，实现了对身份数据的保护。

  IKE 支持的加密算法有：DES、3DES、AES-128、AES-192、AES-256。（DES 和 3DES 加密算法不安全）

- **PFS**

  PFS（Perfect Forward Secrecy），即完善的前向安全性，是指一个密钥被破解，并不影响其他密钥的安全性，因为这些密钥间没有派生关系。IPSec SA 的密钥是从 IKE SA 的密钥导出的，由于一个 IKE SA 协商生成一对或多对 IPSec SA，当 IKE 的密钥被窃取后，攻击者将可能收集到足够的信息来导出 IPSec SA 的密钥，PFS 通过执行一次额外的 DH 交换，保证 IPSec SA 密钥的安全。

IKEv1 秘钥协商过程

IKEv1 秘钥协商分为两个阶段：

第一阶段：通信双方协商和建立 IKE 协议本身的使用的安全通道，即建立 IKE SA。

第二阶段：利用第一阶段建立的安全通道建立一对用于数据安全传输的 IPSEC SA。

###**IKEv1 第一阶段**

IKEv1 第一阶段协商支持两种协商模式：主模式（Main Mode）和野蛮模式（Aggressive Mode）。

主模式

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.89…	192.168.89.…	ISAKMP	614	Identity Protection (Main Mode)
2	0.000096	192.168.89…	192.168.89.…	ISAKMP	234	Identity Protection (Main Mode)
3	0.000148	192.168.89…	192.168.89.…	ISAKMP	422	Identity Protection (Main Mode)
4	0.000424	192.168.89…	192.168.89.…	ISAKMP	422	Identity Protection (Main Mode)
5	0.000676	192.168.89…	192.168.89.…	ISAKMP	150	Identity Protection (Main Mode)
6	0.000746	192.168.89…	192.168.89.…	ISAKMP	134	Identity Protection (Main Mode)

野蛮模式

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.89.35	192.168.89.34	ISAKMP	698	Aggressive
2	0.000228	192.168.89.34	192.168.89.35	ISAKMP	570	Aggressive
3	0.000458	192.168.89.35	192.168.89.34	ISAKMP	214	Aggressive

主模式

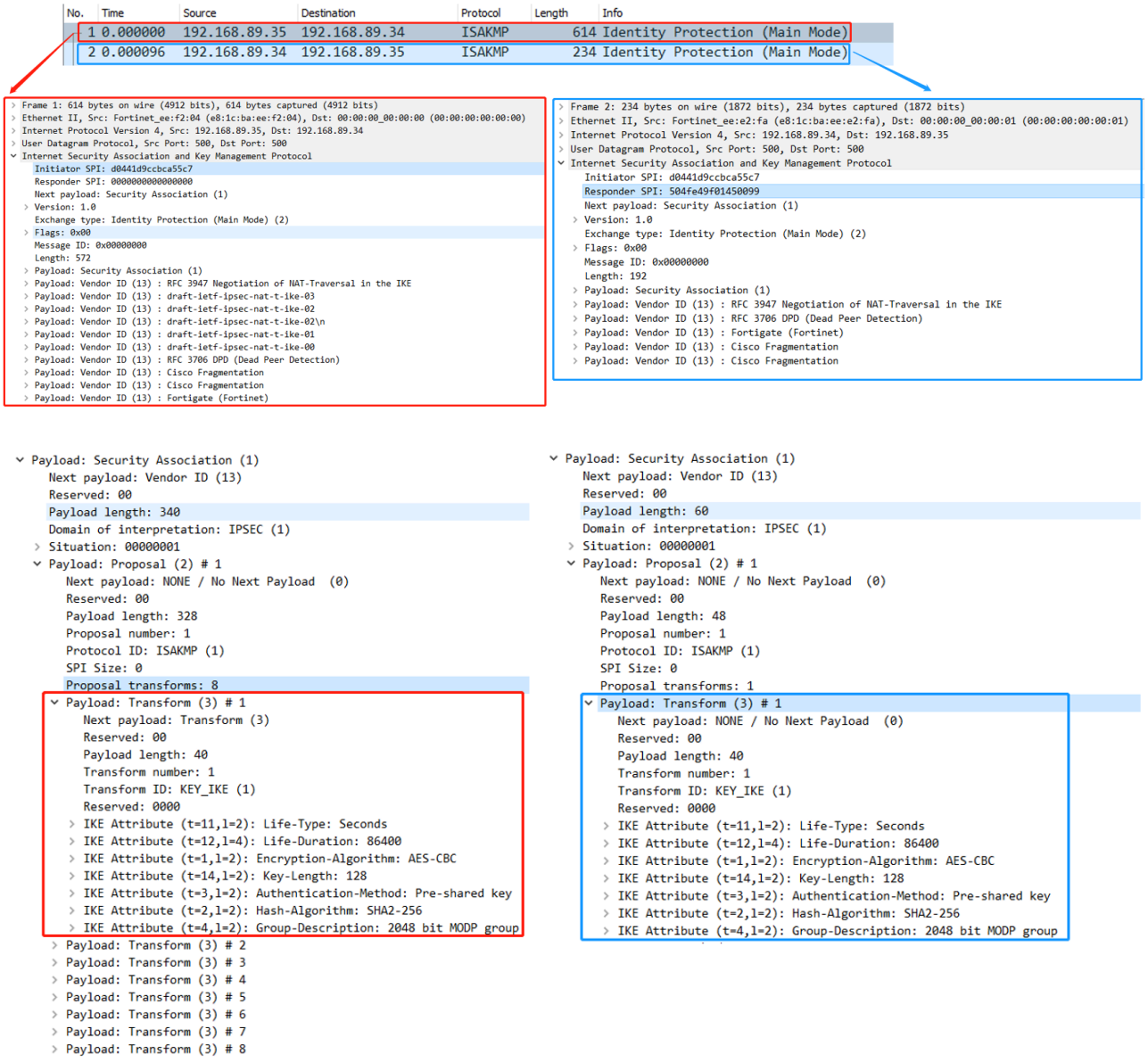
主模式包含 3 次双向交换，用到了 6 条 ISAKMP 信息。

1.**第 1 和 2 个包用于安全提议交换。**

a. 在第 1 个包中，发起方 SPI 设置为随机值，而响应方 SPI 设置为 0；在第 2 个数据包中，响应方 SPI 必须用新值回复，并且整个协商保持相同的 SPI 值。

b. 对 Vendor ID 进行处理，以确定对等体是否支持 NAT-T、DPD 等。

c. 协商安全协商加密和验证的算法。



2.第 3 和 4 个包用于密钥信息交换

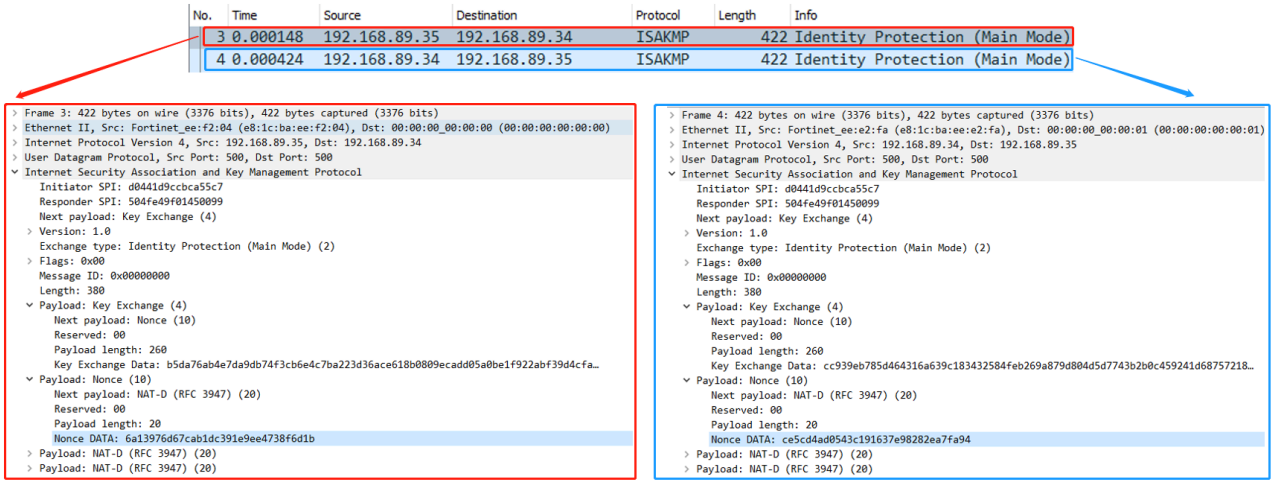
通过 DH 公钥值和 nonce 值，两个对等体将生成一个名为 SKEYID 的种子密钥。为了不同的目的，将使用该种子密钥生成另外 3 个会话密钥：

SKEYID_d：作为阶段 2 生成 KEY 的材料；

SKEYID_a：用来 ISAKMP 包完整性用的 key；

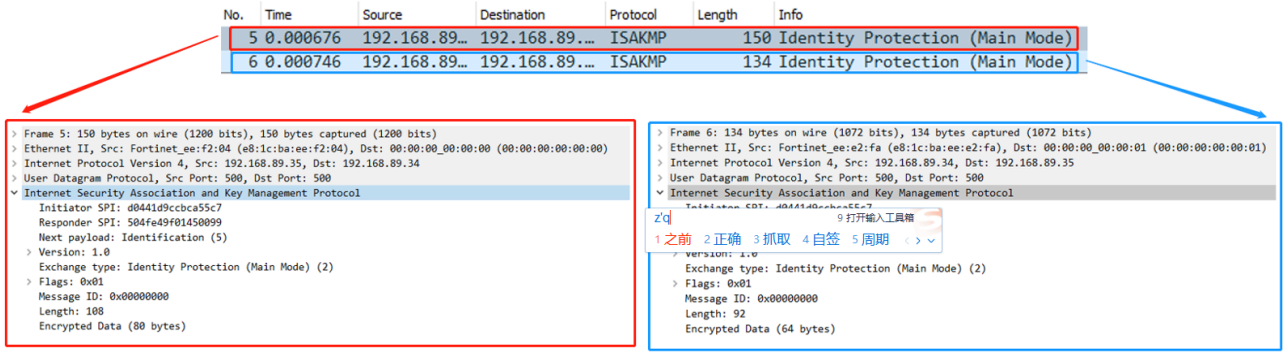
SKEYID_e：用来加密 ISAKMP 包的 key；

nonce：随机生成的数字，用于防重放攻击。



3.第 5 和 6 个包用于身份和认证信息交换

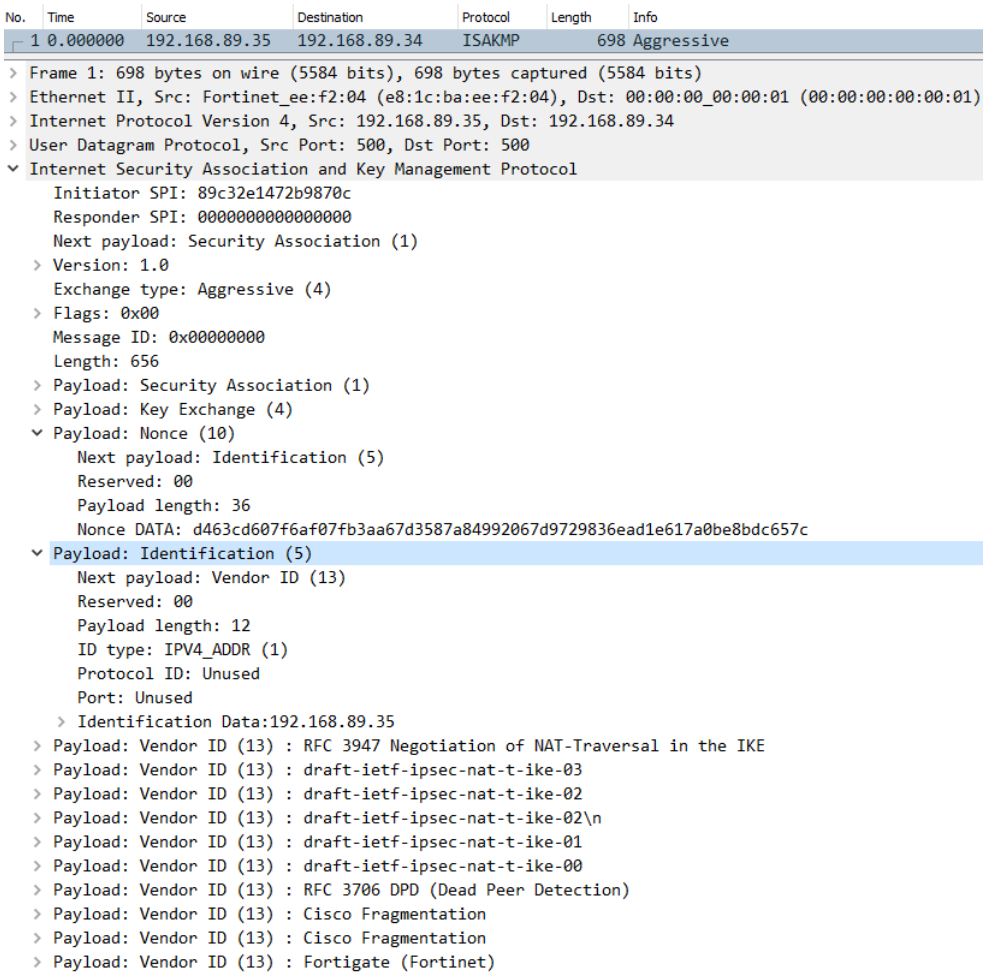
第 5 和 6 个包是加密的，目的是确认对方的身份。如果是使用预共享密钥，那么验证包括检查双方是否有相同的预共享密钥。如果是证书，则对等体交换证书，并假设签署双方的 CA 是受信任的，则验证成功。



野蛮模式

野蛮模式只用到 3 个包。

1. 第 1 个包，发起方发送安全提议、DH 公开值、临时值 (nonce)，以及身份 ID 供响应方验证。


No. Time Source Destination Protocol Length Info
1 0.000000 192.168.89.35 192.168.89.34 ISAKMP 698 Aggressive
Frame 1: 698 bytes on wire (5584 bits), 698 bytes captured (5584 bits)
Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.89.35, Dst: 192.168.89.34
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
Initiator SPI: 89c32e1472b9870c
Responder SPI: 0000000000000000
Next payload: Security Association (1)
Version: 1.0
Exchange type: Aggressive (4)
Flags: 0x00
Message ID: 0x00000000
Length: 656
Payload: Security Association (1)
Payload: Key Exchange (4)
Payload: Nonce (10)
Next payload: Identification (5)
Reserved: 00
Payload length: 36
Nonce DATA: d463cd607f6af07fb3aa67d3587a84992067d9729836ead1e617a0be8bdc657c
Payload: Identification (5)
Next payload: Vendor ID (13)
Reserved: 00
Payload length: 12
ID type: IPV4_ADDR (1)
Protocol ID: Unused
Port: Unused
Identification Data:192.168.89.35
Payload: Vendor ID (13) : RFC 3947 Negotiation of NAT-Traversal in the IKE
Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-03
Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-02
Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-02\n
Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-01
Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-00
Payload: Vendor ID (13) : RFC 3706 DPD (Dead Peer Detection)
Payload: Vendor ID (13) : Cisco Fragmentation
Payload: Vendor ID (13) : Cisco Fragmentation
Payload: Vendor ID (13) : Fortigate (Fortinet)

2. 第 2 个包，响应方用选定提议的所有参数，DH 公开值进行应答，以及身份 ID 供发起方验证。

```
No.  Time      Source          Destination     Protocol  Length  Info
   2 0.000228  192.168.89.34   192.168.89.35   ISAKMP         570 Aggressive

> Frame 2: 570 bytes on wire (4560 bits), 570 bytes captured (4560 bits)
> Ethernet II, Src: Fortinet_ee:e2:fa (e8:1c:ba:ee:e2:fa), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.89.34, Dst: 192.168.89.35
> User Datagram Protocol, Src Port: 500, Dst Port: 500
v Internet Security Association and Key Management Protocol
    Initiator SPI: 89c32e1472b9870c
    Responder SPI: 6f11e242abe9057c
    Next payload: Security Association (1)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x00
    Message ID: 0x00000000
    Length: 528
  > Payload: Security Association (1)
  > Payload: Key Exchange (4)
  v Payload: Nonce (10)
      Next payload: Identification (5)
      Reserved: 00
      Payload length: 20
      Nonce DATA: f457954ca60fb9d9c24d8b312807db57
  v Payload: Identification (5)
      Next payload: Hash (8)
      Reserved: 00
      Payload length: 12
      ID type: IPV4_ADDR (1)
      Protocol ID: Unused
      Port: Unused
    > Identification Data:192.168.89.34
  v Payload: Hash (8)
      Next payload: Vendor ID (13)
      Reserved: 00
      Payload length: 36
      Hash DATA: 29df11090eeecadcb17e74059df78cf5e9bda51bad43e7b0b2e340fde2b2deba
  > Payload: Vendor ID (13) : RFC 3947 Negotiation of NAT-Traversal in the IKE
  > Payload: NAT-D (RFC 3947) (20)
  > Payload: NAT-D (RFC 3947) (20)
  > Payload: Vendor ID (13) : RFC 3706 DPD (Dead Peer Detection)
  > Payload: Vendor ID (13) : Fortigate (Fortinet)
  > Payload: Vendor ID (13) : Cisco Fragmentation
  > Payload: Vendor ID (13) : Cisco Fragmentation
```

3. 第 3 个包，发起方发送验证数据。

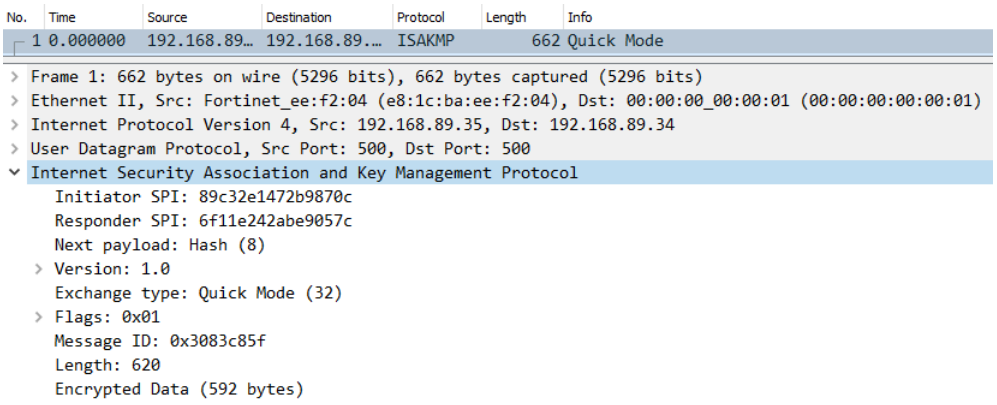
```
No.  Time      Source          Destination     Protocol  Length  Info
   3 0.000458  192.168.89.35   192.168.89.34   ISAKMP         214 Aggressive

> Frame 3: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.89.35, Dst: 192.168.89.34
> User Datagram Protocol, Src Port: 500, Dst Port: 500
v Internet Security Association and Key Management Protocol
    Initiator SPI: 89c32e1472b9870c
    Responder SPI: 6f11e242abe9057c
    Next payload: Hash (8)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x01
    Message ID: 0x00000000
    Length: 172
    Encrypted Data (144 bytes)
```

IKEv1 第二阶段

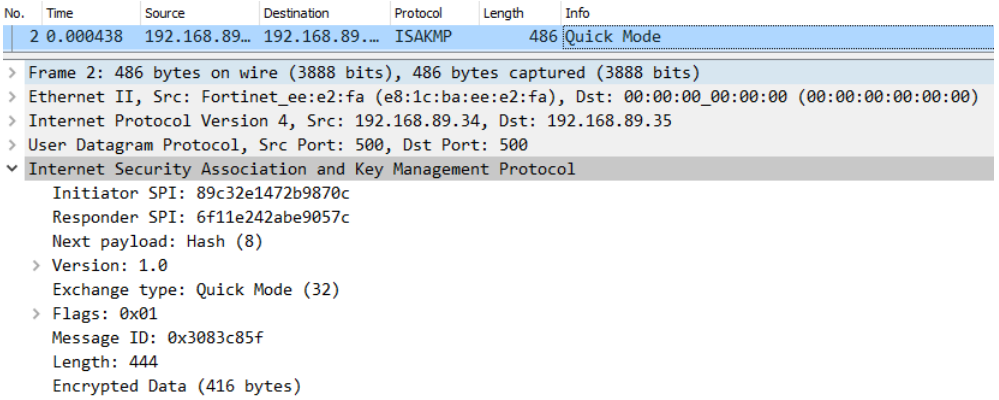
IKEv1 协商阶段二的目的就是建立用来安全传输数据的 IPSec SA，并为数据传输衍生出密钥。该采用快速模式（Quick Mode），使用 IKEv1 协商阶段一中生成的密钥对 ISAKMP 消息的完整性和身份进行验证，并对 ISAKMP 消息进行加密，从而保证了数据交换的安全性。

IKEv1 第二阶段有 3 个包：

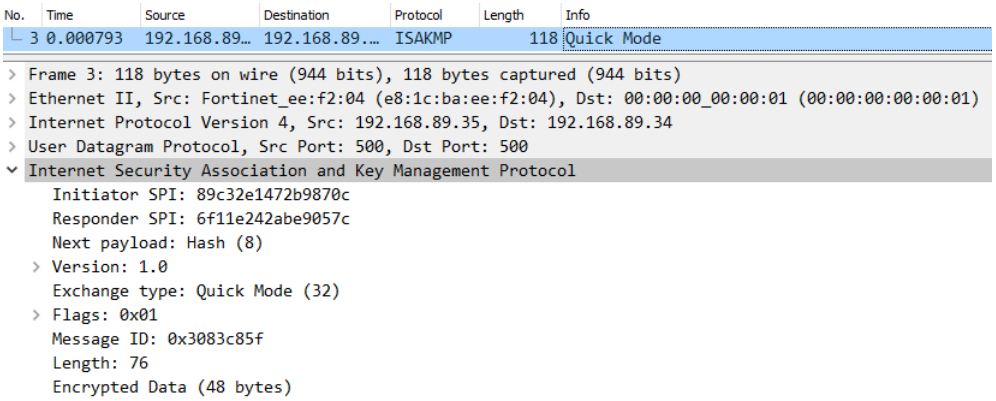
1.第 1 个包，发起方发送本端的安全参数：被保护的数据流和 IPSec 安全提议等，以及身份认证信息。



2.第 2 个包，响应方发送确认的安全参数和身份认证信息并生成 IPSEC 数据传输密钥。



3.发起方发送确认信息，确认与响应方可以通信，协商结束。



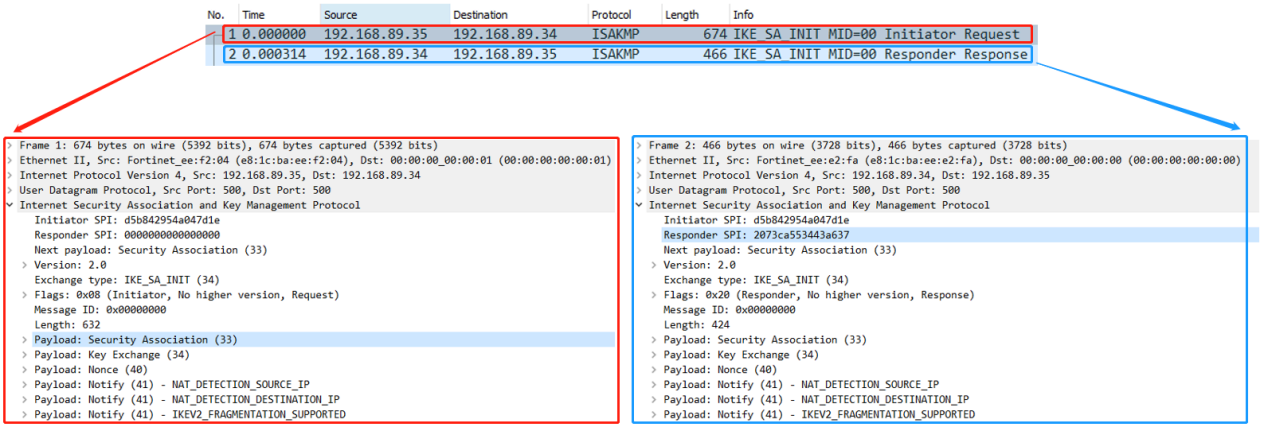
IKEv2 秘钥协商过程

IKEv2 使用 2 次交换共 4 个包就可以完成一对 IPSec SA 的建立，如果建立的 IPSec SA 大于一对时，每一对 IPSec SA 只需额外增加 1 次 CREATE_CHILD_SA 交换就可以完成。IKEv2 定义了 4 种交换：初始交换（Initial Exchanges）、认证交换（IKE_AUTH）、创建子 SA 交换（Create_Child_SA Exchange）以及通知交换（Informational Exchange）。

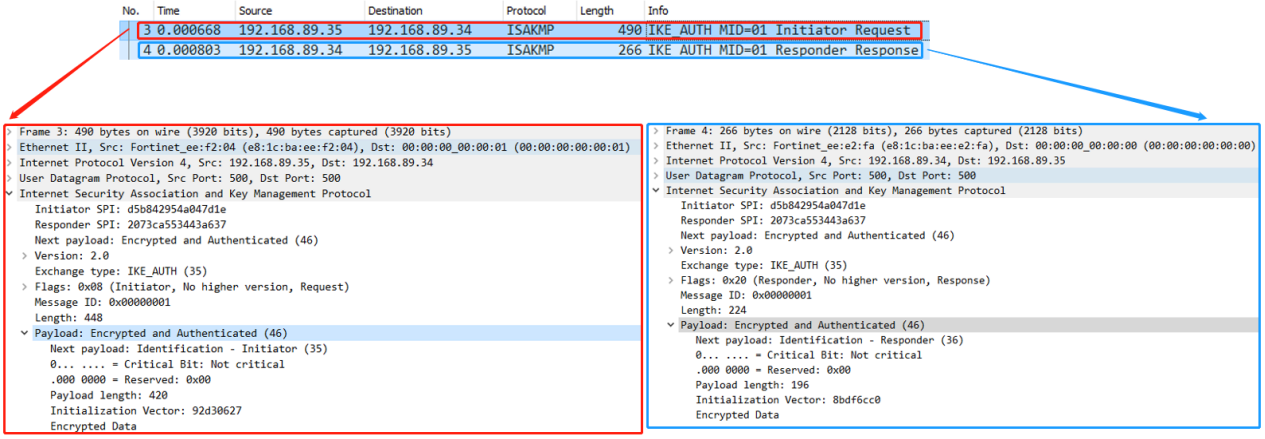
初始交换

IKEv2 通过初始交换就可以完成第一对 IPSec SA 的协商建立。

1. 第 1 和 2 个包，IKE_SA_INIT 交换；协商 IKE SA 加密和验证算法，交换临时随机数和 DH 交换，然后生成一个共享密钥材料，通过这个共享密钥材料可以衍生出 IPSec SA 的所有密钥。

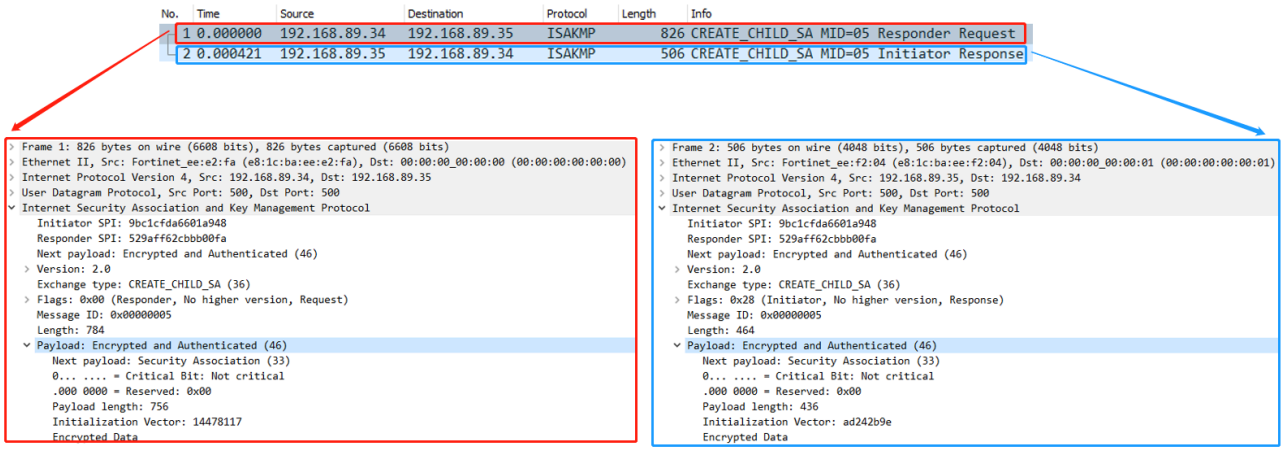


2. 第 3 和 4 个包，IKE_AUTH 交换；完成身份认证、对前两条信息的认证和 IPSec SA 的参数协商，第 3 和 4 个包是加密的，其中可以包含一对用于 ESP 封装的 IPSec SA。



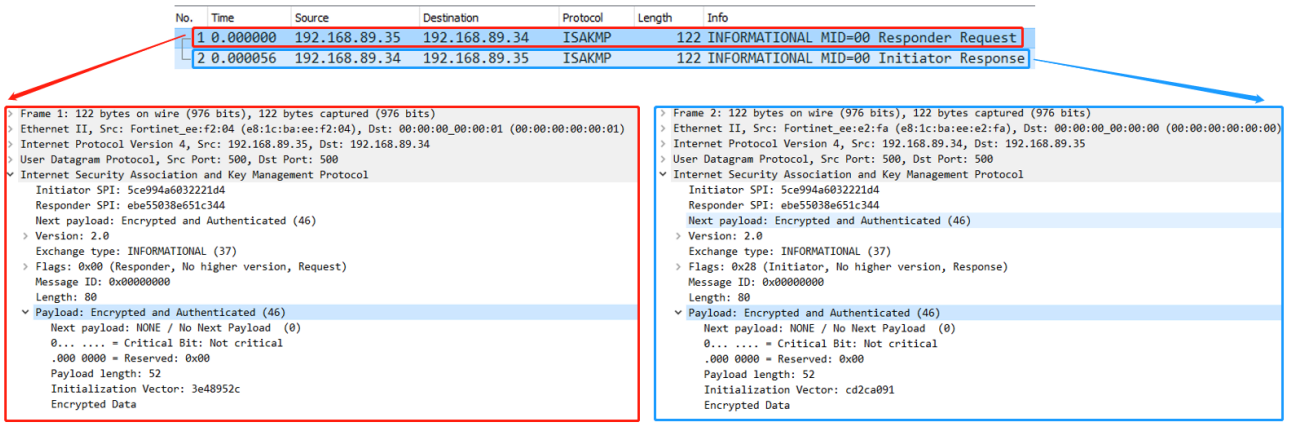
创建子 SA 交换

当一个 IKE SA 需要创建多对 IPSEC SA 时，则创建子 SA 交换来协商后续其他的 IPSec SA。创建子 SA 交换还可以用于 IKE SA 的重协商。



通知交换

IPSEC 两端有时会传递一些控制信息，例如错误信息或者通告信息，这些信息在 IKEv2 中是通过通知交换完成的。



NAT 穿越

疑问：ESP 隧道模式默认只支持一对一 NAT 转换，不支持 PAT；ESP 传输模式不支持 NAT。如果 FortiGate 位于企业内部，运营商出口有单独的设备做 NAT，那么 IPSEC 流量如何穿越 NAT?

解决方案：

NAT-T 技术，即在 IP 和 ESP 报文之间插入一个 8 个字节 UDP 头部 (端口号默认为 4500)，FortiGate 默认 enable NAT-T

新IP头	UDP头	ESP头	原始IP头	TCP头	数据	ESP尾	ESP认证数据

NAT-T 由三部分组成：

1. 首先是判断远程对等体是否支持 NAT-T；

2. 其次是检测对等体之间的路径上是否有 NAT；

3. 最后决定如何使用 UDP 封装来处理 NAT。

IKEv1 穿越 NAT

IKEv1 第一阶段协商支持两种协商模式：主模式（Main Mode）和野蛮模式（Aggressive Mode）。

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
1	0.000…	100.1.1.2	200.1.1.2	ISAKMP	614	500	500	Identity Protection (Main Mode)
2	-5662…	200.1.1.2	100.1.1.2	ISAKMP	234	500	500	Identity Protection (Main Mode)
3	-5662…	100.1.1.2	200.1.1.2	ISAKMP	422	500	500	Identity Protection (Main Mode)
4	-5662…	200.1.1.2	100.1.1.2	ISAKMP	422	500	500	Identity Protection (Main Mode)
5	-5662…	100.1.1.2	200.1.1.2	ISAKMP	154	4500	4500	Identity Protection (Main Mode)
6	-5662…	200.1.1.2	100.1.1.2	ISAKMP	138	4500	4500	Identity Protection (Main Mode)
7	-5662…	100.1.1.2	200.1.1.2	ISAKMP	666	4500	4500	Quick Mode
8	-5662…	200.1.1.2	100.1.1.2	ISAKMP	490	4500	4500	Quick Mode
9	-5662…	100.1.1.2	200.1.1.2	ISAKMP	122	4500	4500	Quick Mode

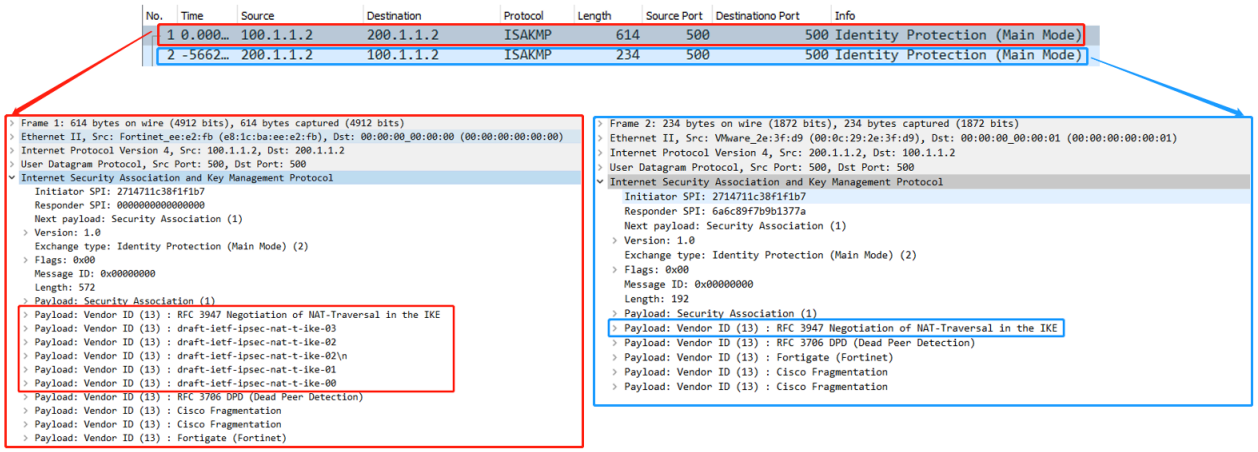
主模式

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
1	0.000…	100.1.1.2	200.1.1.2	ISAKMP	762	500	500	Aggressive
2	0.000…	200.1.1.2	100.1.1.2	ISAKMP	634	500	500	Aggressive
3	0.001…	100.1.1.2	200.1.1.2	ISAKMP	218	4500	4500	Aggressive
4	0.001…	100.1.1.2	200.1.1.2	ISAKMP	666	4500	4500	Quick Mode
5	0.002…	200.1.1.2	100.1.1.2	ISAKMP	490	4500	4500	Quick Mode
6	0.002…	100.1.1.2	200.1.1.2	ISAKMP	122	4500	4500	Quick Mode

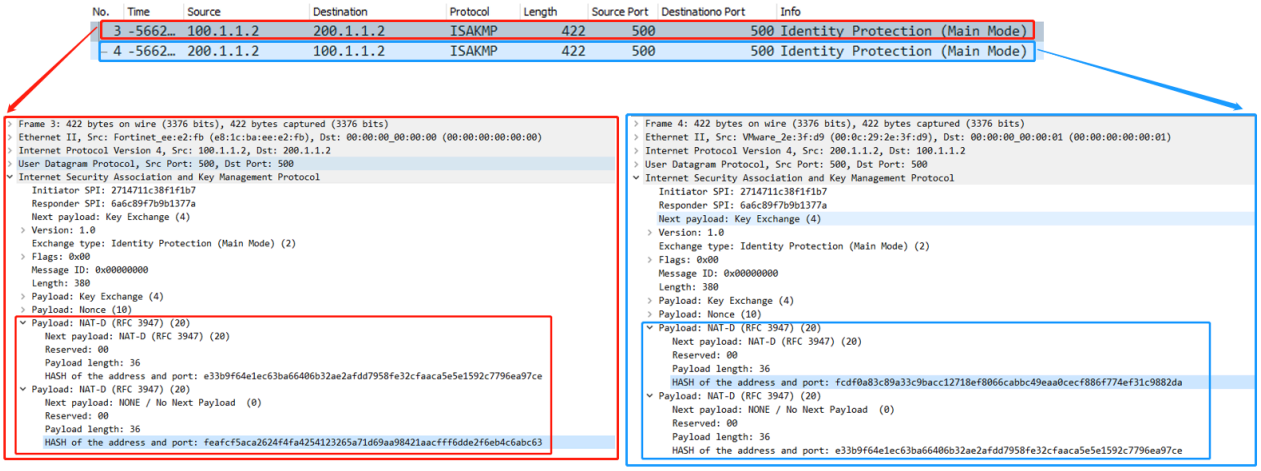
野蛮模式

主模式

1. 第 1 和 2 个包协商是否支持 NAT-T



2. 第 3 和 4 个包，发起方插入了两个 NAT - D 载荷。第一个 NAT - D 载荷包含 IKE 对等体的 IP 地址和端口的 Hash 值，第二个 NAT - D 载荷包含本端的 IP 地址和端口的 Hash 值。响应方也计算这两个 Hash 值，两方计算的哪个 Hash 值不相等，表明哪个设备在 NAT 网关后面。



3. 完成 NAT-T 检测后，如果发现 NAT 网关，则后续 UDP 报文端口号修改为 4500。

   当 UDP 封装的是 ISAKMP 消息，会增加一个 non-ESP marker（为 4 个值为 0 的字节），以示跟封装 ESP 报文有区别。

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
5	-5662…	100.1.1.2	200.1.1.2	ISAKMP	154	4500	4500	Identity Protection (Main Mode)
6	-5662…	200.1.1.2	100.1.1.2	ISAKMP	138	4500	4500	Identity Protection (Main Mode)
7	-5662…	100.1.1.2	200.1.1.2	ISAKMP	666	4500	4500	Quick Mode
8	-5662…	200.1.1.2	100.1.1.2	ISAKMP	490	4500	4500	Quick Mode
9	-5662…	100.1.1.2	200.1.1.2	ISAKMP	122	4500	4500	Quick Mode

```
> Frame 5: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
> Ethernet II, Src: Fortinet_ee:e2:fb (e8:1c:ba:ee:e2:fb), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 100.1.1.2, Dst: 200.1.1.2
> User Datagram Protocol, Src Port: 4500, Dst Port: 4500
v UDP Encapsulation of IPsec Packets
    Non-ESP Marker
v Internet Security Association and Key Management Protocol
    Initiator SPI: 2714711c38f1f1b7
    Responder SPI: 6a6c89f7b9b1377a
    Next payload: Identification (5)
  > Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  > Flags: 0x01
    Message ID: 0x00000000
    Length: 108
    Encrypted Data (80 bytes)
```

```
0000  00 00 00 00 00 00 e8 1c  ba ee e2 fb 08 00 45 00   ........ ......E·
0010  00 8c 21 3f 00 00 40 11  2b 1c 64 01 01 02 c8 01   ··!?··@· +·d·····
0020  01 02 11 94 11 94 00 78  53 7c 00 00 00 00 27 14   ·······x S|····'·
0030  71 1c 38 f1 f1 b7 6a 6c  89 f7 b9 b1 37 7a 05 10   q·8···jl ····7z··
0040  02 01 00 00 00 00 00 00  00 6c db f6 30 43 06 5c   ········ ·l··0C·\
0050  aa eb 05 bd 1f 4f 15 1f  92 35 3a ed 71 b3 f1 06   ·····O·· ·5:·q···
0060  22 c5 36 43 cf 32 5e 78  4e 12 31 6c 4a 78 22 df   "·6C·2^x N·1lJx"·
0070  02 1e 87 cb ba fd 6e be  33 f0 e8 2d 6a 33 04 0d   ······n· 3··-j3··
0080  f8 4c f0 29 42 61 e9 ae  01 fe 44 61 95 f4 25 c4   ·L·)Ba·· ··Da··%·
0090  59 a1 49 fa d8 e5 d7 ae  66 85                     Y·I····· f·
```

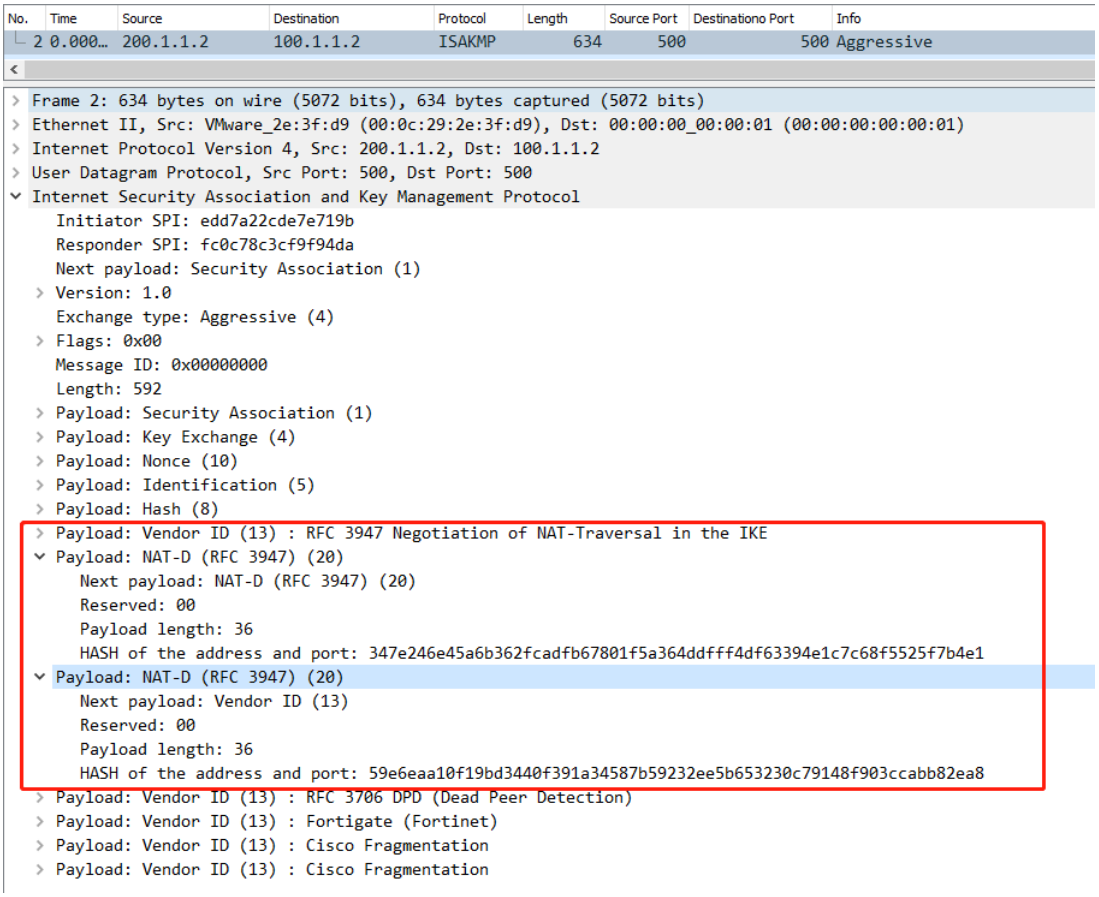
野蛮模式

1. 第 1 个包，发起方在 IKE 消息中插入 Vendor ID 来告知对方自己支持 NAT 穿越。

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
1	0.000…	100.1.1.2	200.1.1.2	ISAKMP	762	500	500	Aggressive

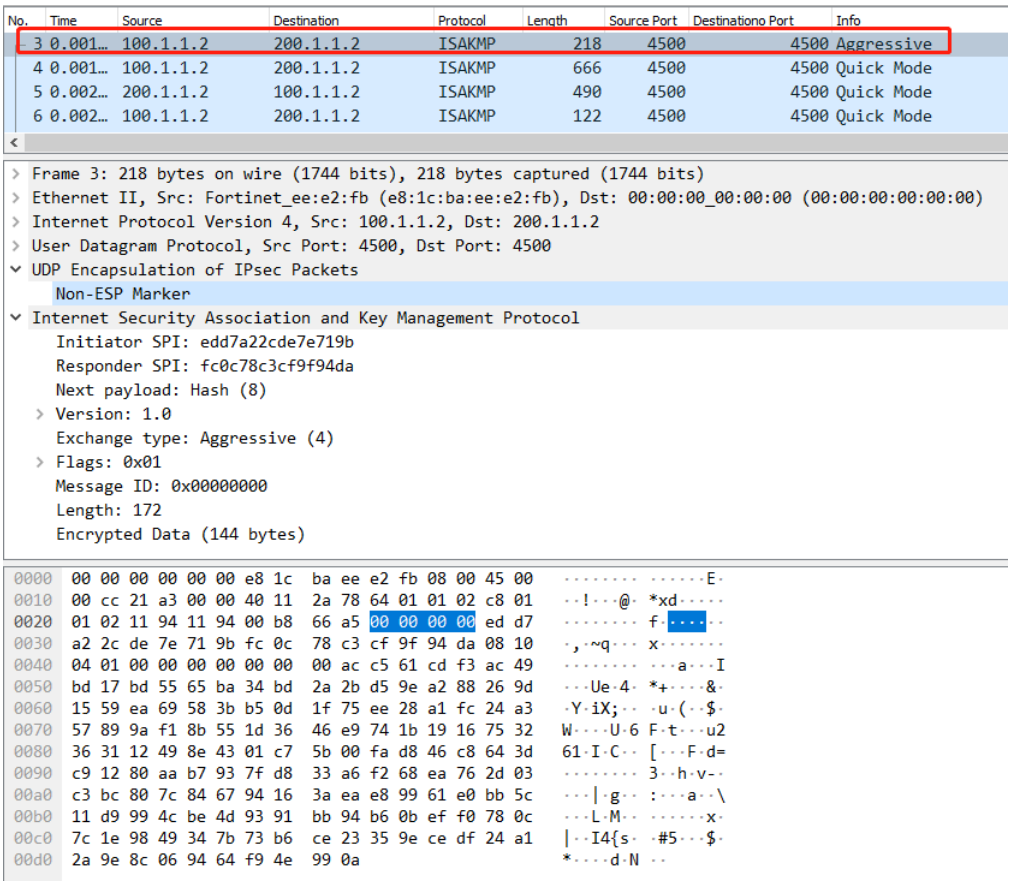
```
> Frame 1: 762 bytes on wire (6096 bits), 762 bytes captured (6096 bits)
> Ethernet II, Src: Fortinet_ee:e2:fb (e8:1c:ba:ee:e2:fb), Dst: 00:00:00_00:00:00 (00:00:00:00:00:
> Internet Protocol Version 4, Src: 100.1.1.2, Dst: 200.1.1.2
> User Datagram Protocol, Src Port: 500, Dst Port: 500
v Internet Security Association and Key Management Protocol
    Initiator SPI: edd7a22cde7e719b
    Responder SPI: 0000000000000000
    Next payload: Security Association (1)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x00
    Message ID: 0x00000000
    Length: 720
  > Payload: Security Association (1)
  > Payload: Key Exchange (4)
  > Payload: Nonce (10)
  > Payload: Identification (5)
  > Payload: Vendor ID (13) : RFC 3947 Negotiation of NAT-Traversal in the IKE
  > Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-03
  > Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-02
  > Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-02\n
  > Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-01
  > Payload: Vendor ID (13) : draft-ietf-ipsec-nat-t-ike-00
  > Payload: Vendor ID (13) : RFC 3706 DPD (Dead Peer Detection)
  > Payload: Vendor ID (13) : Cisco Fragmentation
  > Payload: Vendor ID (13) : Cisco Fragmentation
  > Payload: Vendor ID (13) : Fortigate (Fortinet)
```

2. 第 2 个包，响应方确认支持 NAT-T，并携带两个 NAT - D 载荷。第一个 NAT - D 载荷包含 IKE 对等体的 IP 地址和端口的 Hash 值，第二个 NAT - D 载荷包含本端的 IP 地址和端口的 Hash 值。



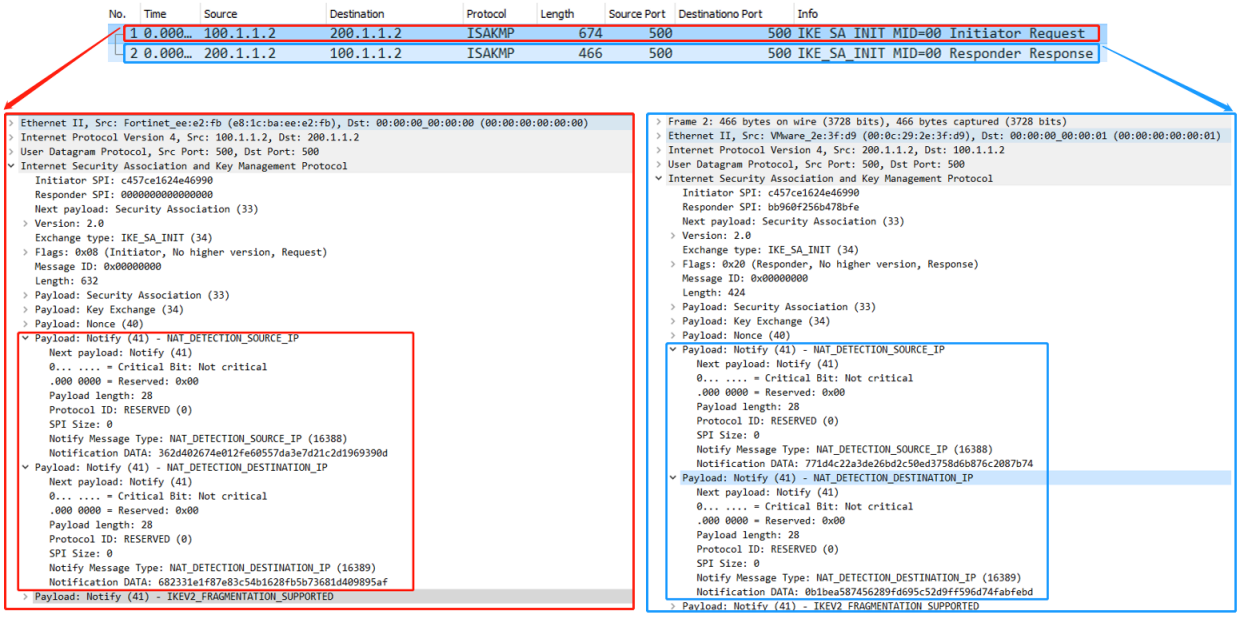
3. 第 3 个包，发起方同样携带自己的两个 NAT-T 载荷，响应方也计算这两个 Hash 值，两方计算的哪个 Hash 值不相等，表明哪个设备在 NAT 网关后面，但这个包的数据内容是加密的。完成 NAT-T 检测后，如果发现 NAT 网关，则从此 UDP 报文开始端口号修改为 4500。

   当 UDP 封装的是 ISAKMP 消息，会增加一个 non-ESP marker（为 4 个值为 0 的字节），以示跟封装 ESP 报文有区别。



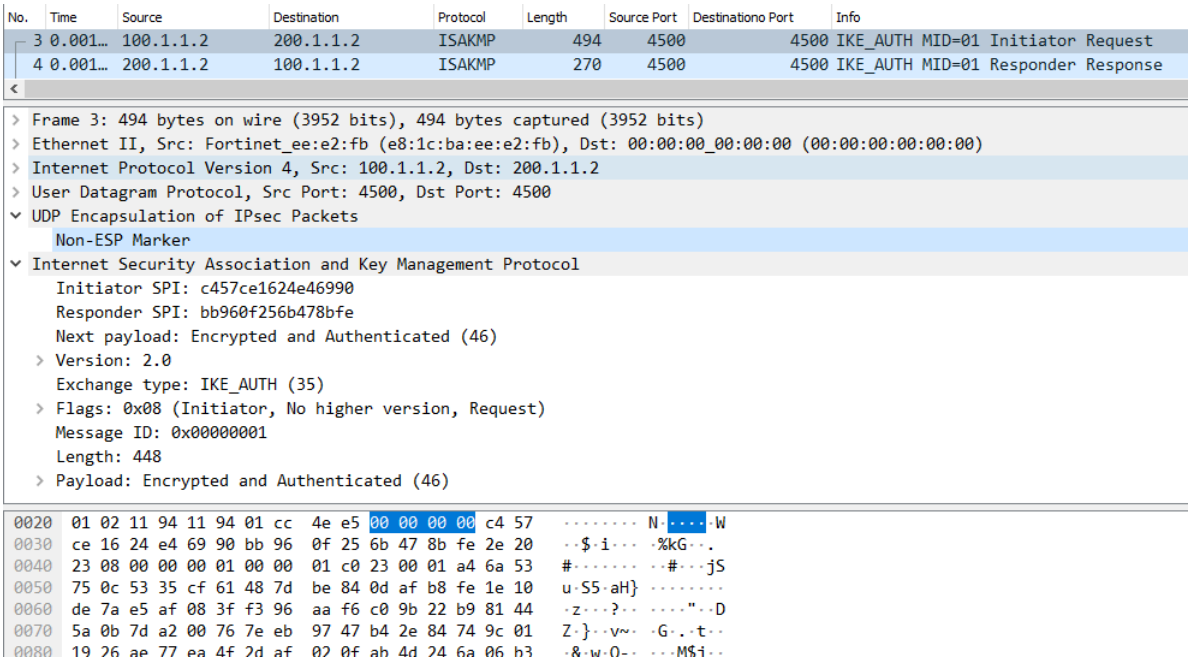
IKEv2 穿越 NAT

1. 第 1 和 2 个包，在 IKE 消息中插入两个载荷，第一个载荷 NAT_DETECTION_SOURCE_IP 包含本端的 IP 地址和端口的 Hash 值，第二个载荷 NAT_DETECTION_DESTINATION_IP 包含对端的 IP 地址和端口的 Hash 值。这两个通知载荷用于检测在将要建立 IPSec 隧道的两个网关之间是否存在 NAT，如果接收到的 NAT_DETECTION_SOURCE 通知载荷没有匹配数据包 IP 头中的源 IP 和端口的 Hash 值，则说明对端位于 NAT 网关后面。如果接收到的 NAT_DETECTION_DESTINATION_IP 通知载荷没有匹配数据包 IP 头中的目的 IP 和端口的 Hash 值，则意味着本端位于 NAT 网关之后。



2. 第 3 和 4 个包，完成 IKE_SA_INIT 后，发现 NAT 设备，后续 UDP 报文端口号修改为 4500。

   当 UDP 封装的是 ISAKMP 消息，会增加一个 non-ESP marker（为 4 个值为 0 的字节），以示跟封装 ESP 报文有区别。



IPSec DPD 模式

DPD 简介

1. Dead Peer Detection（DPD）是一种机制，用于检测远程对等方（peer）是否仍然活跃。在 IPsec VPN 中，两个端点之间的连接可能会由于网络问题或其他原因中断，为了及时发现并处理这种情况，就引入了 DPD。
2. DPD 通过在 IPsec SA（安全关联）之间定期交换检测消息来实现。这些消息通常是在 IKE（Internet Key Exchange）阶段协商的。如果一端在一定时间内没有收到来自对等方的 DPD 消息，就会认为对等方可能已经不再活跃，从而触发重新建立 IKE 通道或采取其他措施来恢复连接。
3. DPD 参数不参与 IKE 协商，仅本地有效，本地的 DPD 配置决定自身的 DPD 探测机制。

DPD 模式

FortiGate 的 DPD 配置有 3 种模式：

- 禁用（disable）：任何情况下都不发送 DPD 探测报文。**一般情况下不建议选择此模式。**
- 空闲（on-idle）：IPSec 隧道内一个方向上没有流量（即 idle）时，在这个方向发送 DPD 探测报文；只有当 IPSec 隧道双向都有流量时，才不发 DPD 探测报文。**推荐使用，可以较快地发现隧道中断。**
- 按需（on-demand）：只有 IPSec 隧道内单向有流量时，在这个方向上发送 DPD 探测报文，其他情况（双向都有流量或双向都没有流量）下都不发送 DPD 探测报文。**较少的带宽占用和更少的 CPU 中断影响，但检测到隧道中断的速度较慢。**

隧道建立阶段	set dpd on-idle	set dpd on-demand
Tunnel 建立前 （阶段 1 已建立，阶段 2 未建立）	有 DPD probe	没有 DPD probe
Tunnel 建立后 （阶段 2 已建立，但没有 IPSec 流量）	有 DPD probe	没有 DPD probe
Ping over tunnel （Reques/Reply，IPSec 双向有流量）	没有 DPD probe	没有 DPD probe
Ping over tunnel （Reques/no reply，IPSec 单向流量）	Ping reply 方向有 DPD probe	Ping request 方向有 DPD probe

IPSec 重协商机制

SA 到期时间

FortiGate 的 IPSec 关于 SA 的到期时间，有两个相关的时间概念：

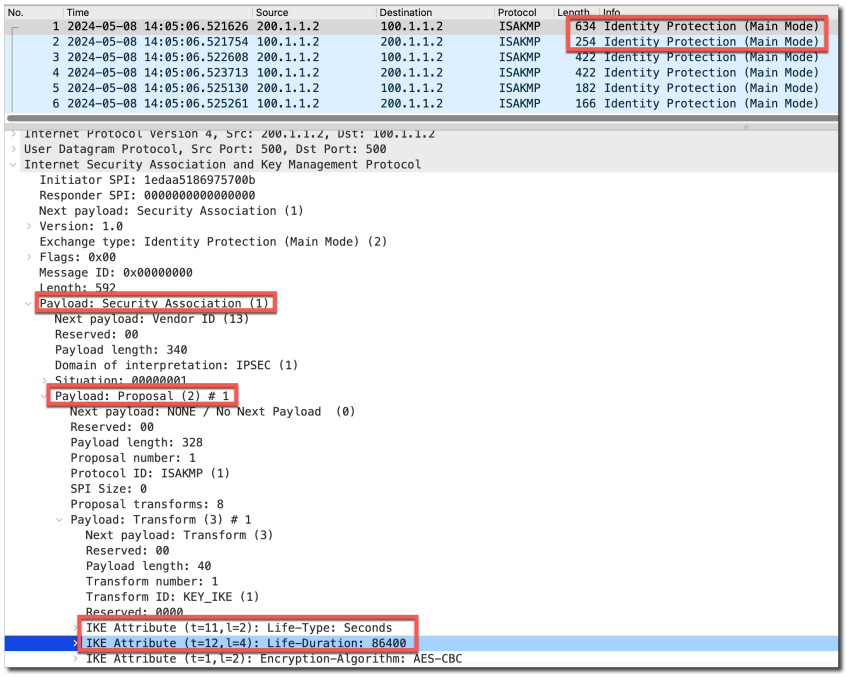
- **hard-timeout**：在 IPSec 的一阶段和二阶段中，均可以配置重协商超时时间（lifetime，下文我们用 hard-timeout 来表示配置的 lifetime）。当 hard-timeout 到期后，FortiGate 将结束当前一阶段/二阶段的 SA。hard-timeout 表示每一个一阶段/二阶段 SA 在本地的最大存在时间。默认配置下，FortiGate 的 IPSec 一阶段重协商（hard-timeout）超时时间为 86400s（24 小时），二阶段重协商（hard-timeout）超时时间为 43200s（12 小时）。
- **soft-timeout**：实际的重协商间隔时间会小于 hard-timeout，被称为 soft-timeout。soft-timeout 总是小于 hard-timeout。在 hard-timeout 到期前，FortiGate 会发起新的一阶段/二阶段协商并形成一对新的 SA。同时，旧的一阶段/二阶段 SA 直到 hard-timeout 到期时才会消失，在（hard-timeout - soft-timeout）这个时间差（rekey-margin）内，会存在两对 SA。soft-timeout 是一种平滑机制，保证在旧 SA 的 hard-timeout 到期前协商出新的 SA。在旧 SA 与新 SA 交替的过程中，IPSec 业务不会中断。如果在 SA 的 hard-timeout 到期时才重协商新的 SA，那么会导致 IPSec 业务在重协商期间中断。

> **重要**
>
> soft-timeout rekey 机制需要在 IPSec 二阶段中开启 `keepalive` 功能才能生效，详情请参考：VPN → IPSec VPN → 自动协商与 keepalive ↗ 章节。

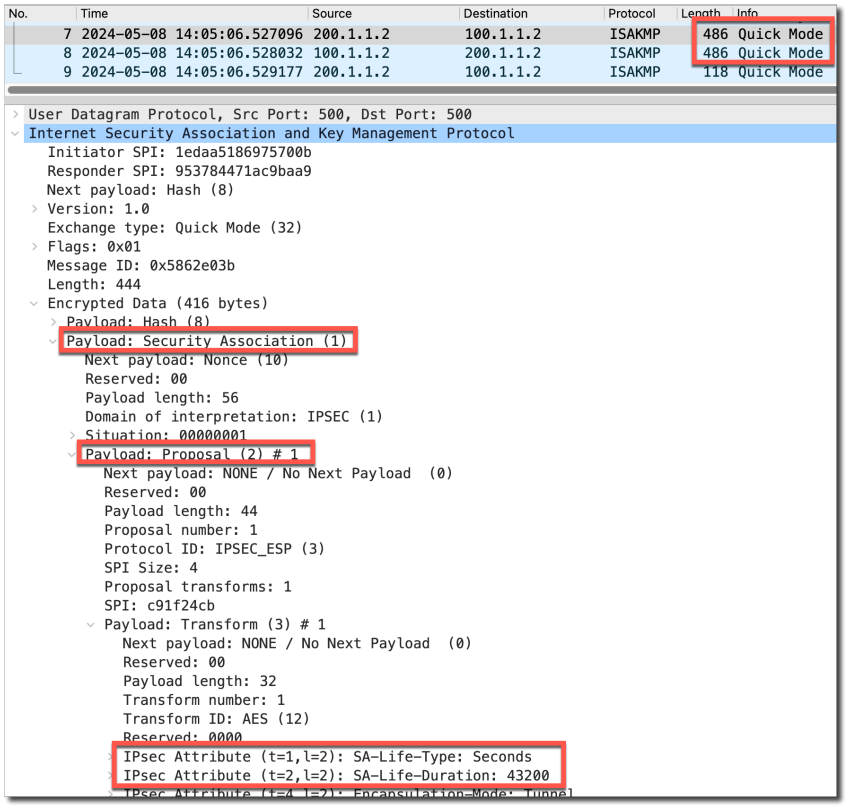
hard-timeout 交互

在 IPSec 一阶段/二阶段协商交互报文中，会携带已配置的重协商时间：

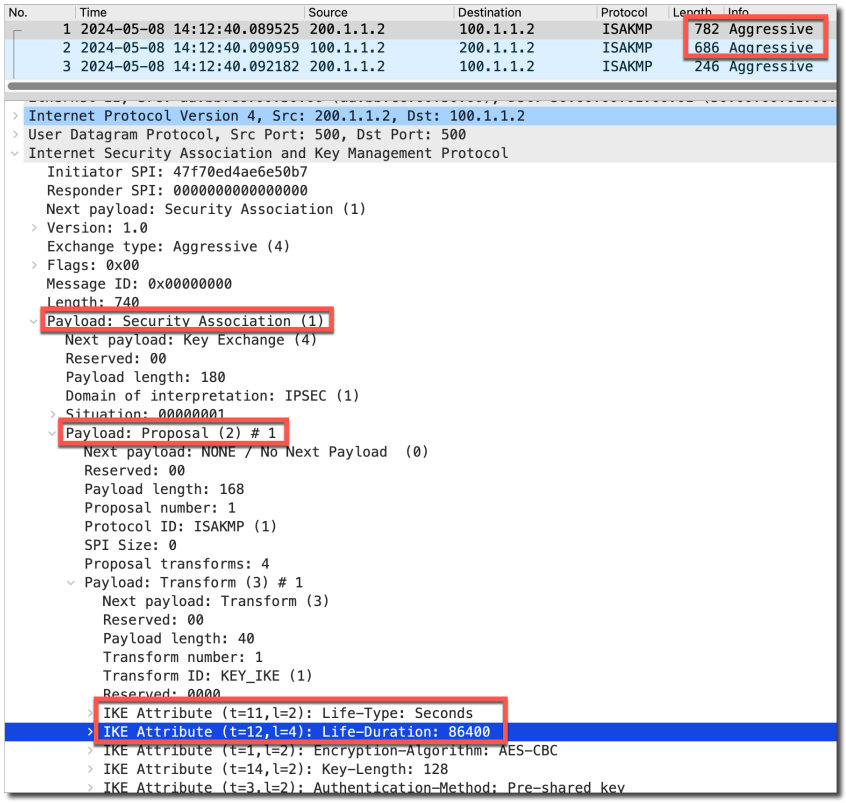
- IKEv1 主模式：
  - 一阶段，位于第 1、2 个协商报文中：



  - 二阶段，位于第 1、2 个协商报文中：



- IKEv1 野蛮模式：
  - 一阶段，位于第 1、2 个协商报文中：



  - 二阶段：与 IKEv1 主模式相同，位于第 1、2 个协商报文中。
- IKEv2：FortiGate 一阶段/二阶段协商时不携带 hard-timeout 参数。

hard-timeout 协商

IKEv1

如 hard-timeout 交互章节所示，IPSec IKEv1 协商的双方会交互 hard-timeout 参数。但并不会因为双方配置的 hard-timeout 不一致而导致协商失败。**当 IPSec IKEv1 协商的双方配置的一阶段/二阶段 hard-timeout 不一致时，两端 IPSec 应以 IPSec 协商双方配置的较小的 hard-timeout 为准。**如下示例：

1. FW1 与 FW2 之间建立 IPSec IKEv1（主模式或野蛮模式）连接，FW1 配置 IPSec 的一阶段/二阶段协商时间为 3600s/1800s，FW2 配置 IPSec 的一阶段/二阶段协商时间为 1800s/3600s。

```bash
FW1:
config vpn ipsec phase1-interface
    edit "to_FW2"
        set interface "port2"
        set keylife 3600    ←——一阶段hard-timeout
        set net-device disable
        set remote-gw 201.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW2"
        set phase1name "to_FW2"
        set keylifeseconds 1800    ←——二阶段hard-timeout
    next
end
```

```bash
FW2:
config vpn ipsec phase1-interface
    edit "to_FW1"
        set interface "port2"
        set keylife 1800    ←——一阶段hard-timeout
        set net-device disable
        set remote-gw 200.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW1"
        set phase1name "to_FW1"
        set keylifeseconds 3600    ←——二阶段hard-timeout
    next
end
```

2. 两端 IPSec 建立后，查看 IPSec 一阶段的 hard-timeout，可以看到 FW1 和 FW2 协商成功的一阶段 hard-timeout 以 FW2 配置（较小的 hard-timeout）为准（1800s）。

```bash
FW1 # diagnose vpn ike gateway list name to_FW2 | grep "lifetime\|name"
name: to_FW2
  lifetime/rekey: 1800/1778    ←——“/”前为协商完成后生效的一阶段hard-timeout

FW2 # diagnose vpn ike gateway list name to_FW1 | grep "lifetime\|name"
name: to_FW1
  lifetime/rekey: 1800/1751    ←——“/”前为协商完成后生效的一阶段hard-timeout
```

3. 查看两端 IPSec 二阶段的 hard-timeout，可以看到 FW1 和 FW2 协商成功的二阶段 hard-timeout 以 FW1 配置（较小的 hard-timeout）为准（1800s）。

```bash
FW1 # get vpn ipsec tunnel name to_FW2 | grep "name\|lifetime"
  name: 'to_FW2'
    name: 'to_FW2'
      lifetime/rekey: 1800/1618    ←——“/”前为协商完成后生效的二阶段hard-timeout
```

```
FW2 # get vpn ipsec tunnel name to_FW1 | grep "name\|lifetime"
  name: 'to_FW1'
    name: 'to_FW1'
      lifetime/rekey: 1800/1564     <———“/”前为协商完成后生效的二阶段hard-timeout
```

IKEv2

IKEv2 的协商报文中，不包含任何 hard-timeout 参数，所以 **IKEv2 配置下 IPSec 双方的 hard-timeout 以本地配置为准，不会进行交互**。如下示例：

1. FW1 与 FW2 之间建立 IPSec IKEv2 连接，FW1 配置 IPSec 的一阶段/二阶段协商时间为 3600s/1800s，FW2 配置 IPSec 的一阶段/二阶段协商时间为 1800s/3600s。

```bash
FW1:
config vpn ipsec phase1-interface
    edit "to_FW2"
        set interface "port2"
        set ike-version 2
        set keylife 3600     <———一阶段hard-timeout
        set net-device disable
        set remote-gw 201.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW2"
        set phase1name "to_FW2"
        set keylifeseconds 1800     <———二阶段hard-timeout
    next
end
```

```bash
FW2:
config vpn ipsec phase1-interface
    edit "to_FW1"
        set interface "port2"
        set ike-version 2
        set keylife 1800     <———一阶段hard-timeout
        set net-device disable
        set remote-gw 200.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW1"
        set phase1name "to_FW1"
        set auto-negotiate enable
        set keylifeseconds 3600     <———二阶段hard-timeout
    next
end
```

2. 两端 IPSec 建立后，查看 IPSec 一阶段的 hard-timeout，可以看到 FW1 和 FW2 的一阶段 hard-timeout 以本地配置的为准，FW1 为 3600s，FW2 为 1800s。

```bash
FW1 # diagnose vpn ike gateway list name to_FW2 | grep "hard-timeout\|name"
name: to_FW2
  lifetime/rekey: 3600/3313     <———“/”前为协商完成后生效的一阶段hard-timeout

FW2 # diagnose vpn ike gateway list name to_FW1 | grep "hard-timeout\|name"
name: to_FW1
  lifetime/rekey: 1800/1725     <———“/”前为协商完成后生效的一阶段hard-timeout
```

3. 查看两端 IPSec 二阶段的 hard-timeout，可以看到 FW1 和 FW2 的二阶段 hard-timeout 以本地配置的为准，FW1 为 1800s，FW2 为 3600s。

```bash
FW1 # get vpn ipsec tunnel name to_FW2 | grep "name\|lifetime"
  name: 'to_FW2'
    name: 'to_FW2'
      lifetime/rekey: 1800/1648     <———“/”前为协商完成后生效的二阶段hard-timeout
```

```
FW2 # get vpn ipsec tunnel name to_FW1 | grep "name\|lifetime"
  name: 'to_FW1'
    name: 'to_FW1'
      lifetime/rekey: 3600/3299     <——"/"前为协商完成后生效的二阶段hard-timeout
```

soft-timeout 数值

如上文介绍，soft-timeout 会比 hard-timeout 减小一定的数值（rekey-margin），根据 hard-timeout 设置的大小以及 IPSec 连接建立的会话方向是发起方（initiator）还是响应方（responder），这个减去的值会相应变化，总结如下：

重要

- initiator：IPSec 一阶段建立成功的发起方，也就是 IKE 成功协商过程中第 1 个 IKE 报文的发起者。
- responder：IPSec 一阶段建立成功的响应方，也就是 IKE 成功协商过程中第 2 个 IKE 报文的响应者。

hard-timeout 配置	发起方 soft-timeout - hard-timeout (rekey-margin)	响应方 soft-timeout - hard-timeout (rekey-margin)
hard-timeout < 3600s	-30s	-20s
hard-timeout ⩾ 3600s	-300s	-270s

无论是发起方（initiator）还是响应方（responder），当 soft-timeout 到期时（hard-timeout 还未到期），会主动与对端 IPSec 协商出一个新的一阶段/二阶段 SA，在（hard-timeout - soft-timeout = rekey-margin）这个时间差内，会与旧的 SA 共存。如下示例：

1. FW1 与 FW2 之间建立 IPSec IKEv1（主模式或野蛮模式）连接，FW1 配置 IPSec 的一阶段/二阶段协商时间为 120s/120s，FW2 配置 IPSec 的一阶段/二阶段协商时间为 120s/120s。为了测试时使 FW1 一直为响应方（Responder），开启 FW1 IPSec 一阶段中的 `passive-mode` 功能（不会主动发起 IKE 协商）。

```bash
FW1:
config vpn ipsec phase1-interface
    edit "to_FW2"
        set interface "port2"
        set keylife 120     <——一阶段hard-timeout
        set net-device disable
        set passive-mode enable     <——开启passive-mode，不会主动发起IKE协商
        set remote-gw 201.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW2"
        set phase1name "to_FW2"
        set keylifeseconds 120     <——二阶段hard-timeout
    next
end
```

```bash
FW2:
config vpn ipsec phase1-interface
    edit "to_FW1"
        set interface "port2"
        set keylife 120     <——一阶段hard-timeout
        set net-device disable
        set remote-gw 200.1.1.2
        set psksecret xxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "to_FW1"
```

```
        set phase1name "to_FW1"
        set auto-negotiate enable    ←——开启二阶段自动发起协商
        set keylifeseconds 120    ←——二阶段hard-timeout
    next
end
```

2. 两端 IPSec 建立后，立即查看 IPSec 一阶段的 hard-timeout 与 soft-timeout，可以看到响应方（responder）FW1 的 soft-timeout 比 hard-timeout 小 20s，发起方（initiator）FW2 的 soft-timeout 比 hard-timeout 小 30s，符合本章节开头的总结。

```bash
FW1 # diagnose vpn ike gateway list name to_FW2 | grep "name\|lifetime\|status\|direction"
name: to_FW2
  direction: responder    ←——FW1为IKE会话的响应方
  status: established 0-0s ago = 0ms    ←——SA已建立时间
  lifetime/rekey: 120/100    ←——作为Responder，soft-timeout比hard-timeout小20s

FW2 # diagnose vpn ike gateway list name to_FW1 | grep "name\|lifetime\|status\|direction"
name: to_FW1
  direction: initiator    ←——FW2为IKE会话的发起方
  status: established 0-0s ago = 0ms    ←——SA已建立时间
  lifetime/rekey: 120/90    ←——作为Initiator，soft-timeout比hard-timeout小30s
```

3. 在连接刚建立时，立即查看 IPSec 二阶段的 hard-timeout 与 soft-timeout，可以看到响应方（responder）FW1 的 soft-timeout 比 hard-timeout 小 20s，发起方（initiator）FW2 的 soft-timeout 比 hard-timeout 小 30s，符合本章节开头的总结。

```bash
FW1 # get vpn ipsec tunnel name to_FW2 | grep "name\|lifetime"
  name: 'to_FW2'
    name: 'to_FW2'
      lifetime/rekey: 120/100    ←——作为Responder，soft-timeout比hard-timeout小20s

FW2 # get vpn ipsec tunnel name to_FW1 | grep "name\|lifetime"
  name: 'to_FW1'
    name: 'to_FW1'
      lifetime/rekey: 120/90    ←——作为Initiator，soft-timeout比hard-timeout小30s
```

4. 所以理论上，发起方 FW2（initiator）的 soft-timeout 总是比响应方 FW1（responder）的 soft-timeout 先到期。

5. 等待发起方 FW2（initiator）的一阶段 soft-timeout 到期，发起方 FW2（initiator）会主动与 FW1 协商一个新的 SA（soft-timeout 从 90s 开始倒计时），旧 SA 的 soft-timeout 虽然变为 0，但会重新从 30s（hard-timeout - soft-timeout）开始倒计时，旧 SA 的倒计时参数从 `rekey` 变为 `expiry`。

```bash
FW2 # diagnose vpn ike gateway list name to_FW1 | grep "name\|lifetime\|status\|direction"
name: to_FW1
  direction: initiator
  lifetime/rekey: 120/0    ←——FW2(initiator)的soft-timeout倒计时结束

FW2 # diagnose vpn ike gateway list name to_FW1 | grep "name\|lifetime\|status\|direction"
name: to_FW1
  direction: initiator
  lifetime/rekey: 120/90    ←——新的SA生成，rekey时间从90s开始倒计时
  direction: initiator
  hard-timeout/expiry: 120/30    ←——FW2(initiator)的旧SA，soft-timeout到期后，剩余的hard-timeout倒计时
```

6. 30s 后，FW2 旧 SA 的 `expiry` 倒计时结束，旧 SA 彻底删除，只剩下新的 SA。

```bash
FW2 # diagnose vpn ike gateway list name to_FW1 | grep "name\|lifetime\|status\|direction"
name: to_FW1
  direction: initiator
  lifetime/rekey: 120/59    ←——新SA的soft-timeout倒计时，旧SA删除
```

7. 二阶段的情况与一阶段一致，所以在这种 soft-timeout 机制下，由于旧 SA 与新 SA 有共同存在时间差（hard-timeout - soft-timeout），只要网络状态正常，IPSec 业务不会产生中断。

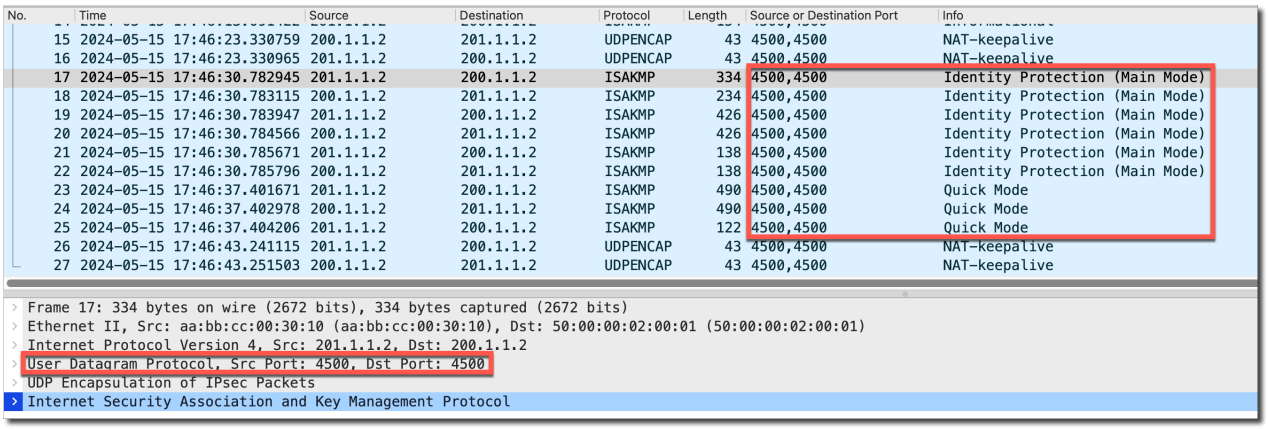
Rekey 报文交互

IKEv1 主模式

FW2 的 SA soft-timeout 到期时发起的新的 SA 协商，新的 SA 协商正常使用 IKEv1 主模式报文（一阶段 6 个包，二阶段 3 个包）交互，根据是否为 NAT-T 环境，Rekey 协商报文所使用的端口有所不同：

- NAT-T：使用 Rekey 方式协商的 IKEv1 报文，所有协商报文将使用 UDP 4500 端口进行协商，报文内容不变。

> **重要**
>
> 而新发起（非 Rekey 方式）的 IKEv1 主模式协商第一阶段前 4 个报文使用 UDP 500 端口，后 2 个报文才开始使用 UDP 4500 端口）。



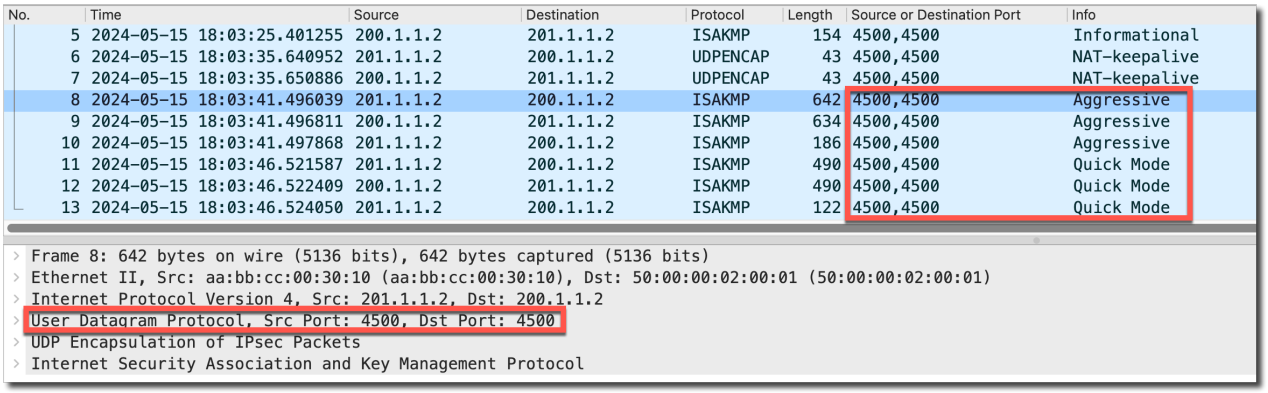
- 非 NAT-T：无论是 Rekey 方式还是新发起的 IKEv1 主模式协商，都是用 UDP 500 端口。

IKEv1 野蛮模式

FW2 的 SA soft-timeout 到期时发起的新的 SA 协商，新的 SA 协商正常使用 IKEv1 野蛮模式报文（一阶段 3 个包，二阶段 3 个包）交互，根据是否为 NAT-T 环境，Rekey 协商报文所使用的端口有所不同：

- NAT-T：使用 Rekey 方式协商的 IKEv1 报文，所有协商报文将使用 UDP 4500 端口进行协商，报文内容不变。

> **重要**
>
> 而新发起（非 Rekey 方式）的 IKEv1 野蛮模式协商第一阶段前 2 个报文使用 UDP 500 端口，最后 1 个报文才开始使用 UDP 4500 端口）。



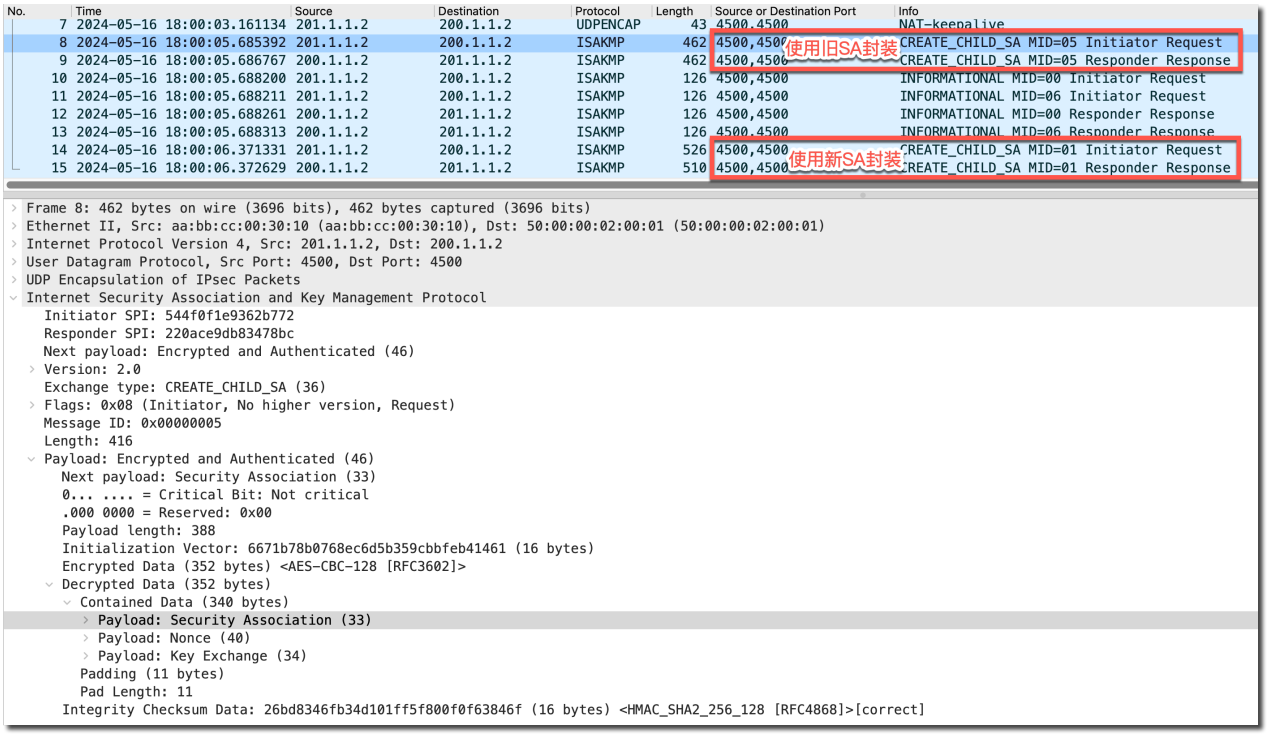
- 非 NAT-T：无论是 Rekey 方式还是新发起的 IKEv1 野蛮模式协商，都是用 UDP 500 端口。

IKEv2

FW2 的 SA soft-timeout 到期时发起的新的 SA 协商，新的一阶段 SA 协商包（原 IKE_SA_INIT）使用旧的一阶段 SA 进行加密/验证，使用 CREATE_CHILD_SA 封装发送，新的二阶段 SA 协商包（原 IKE_AUTH）使用新的一阶段 SA 进行加密/验证，使用 CREATE_CHILD_SA 封装发送。所以对于 IKEv2 的 Rekey 过程，均是被加密/验证保护的：

- NAT-T：使用 Rekey 方式协商的 IKEv2 报文，所有协商报文将使用 UDP 4500 端口进行协商，报文内容不变。

> **重要**
>
> 而新发起（非 Rekey 方式）的 IKEv2 协商 IKE_SA_INIT 的 2 个报文使用 UDP 500 端口，IKE_AUTH 的 2 个报文才开始使用 UDP 4500 端口）。



- 非 NAT-T：无论是 Rekey 方式还是新发起的 IKEv2 协商，都是用 UDP 500 端口。

自动协商与 keepalive

提示

- 本文主要讨论 IPSec 二阶段中的 `auto-negotiate` 和 `keepalive`。
- IPSec 一阶段中的 `keepalive` 配置是用于保持 NAT-T 环境下的 UDP 4500 会话，不属于本文讨论范围。

配置方法

提示

若 IPSec 一阶段开启了 `passive-mode`，二阶段的 `auto-negotiate` 与 `keepalive` 将无法开启。

`auto-negotiate` 与 `keepalive` 是 IPsec 二阶段配置中用于隧道保持在线的机制。二者独立工作，又可搭配使用。

- `auto-negotiate`：二阶段 SA 不存在时主动发起协商，默认关闭（dynamic 模式的隧道无法配置），无需流量触发。

```bash
config vpn ipsec phase2-interface
    edit <phase2_interface_name>
        set phase1name <phase1_interface_name>
        set auto-negotiate [enable | disable]
    next
end
```

- `keepalive`：在已经有二阶段 SA 存在的情况下，决定二阶段 SA 达到 soft-timeout 时是否立即 rekey，防止隧道在 hard-timeout 到期时再进行 rekey，导致隧道中断。

相关信息

soft-timeout = hard-timeout – rekey-margin

soft-timeout、hard-timeout、rekey-margin 的相关信息请参考 VPN → IPSec VPN → IPSec 重协商机制 ↗章节。

重要

`auto-negotiate` 开启后，`keepalive` 会被强制开启，且 `keepalive` 相关命令消失。只有在 `auto-negotiate` 关闭时才可以控制 `keepalive` 的开关。

```bash
config vpn ipsec phase2-interface
    edit <phase2_interface_name>
        set phase1name <phase1_interface_name>
        set auto-negotiate disable
        set keepalive [enable | disable]
    next
end
```

工作机制

不能将 `auto-negotiate` 和 `keepalive` 混淆，这是两个完全独立的功能。简单来说：

- `auto-negotiate`：没有二阶段隧道时 → 主动协商建立，无需流量触发。
- `keepalive`：有二阶段隧道的前提下 → 在 soft-timeout 到期时发起 rekey 防止隧道中断。

功能	生效条件	开启行为	关闭行为
`auto-negotiate`	二阶段 SA 不存在	持续主动发起二阶段协商，直到 SA 建立，无需流量触发	不会主动发起协商，需流量触发或对端发起
`keepalive`	二阶段 SA 达到 soft-timeout	立即发起二阶段 rekey，需要已建立的二阶段 SA	若最近 2 分钟内无流量，在 hard-timeout 到期时删除 SA，但不会 rekey 若最近 2 分钟内有流量，在 hard-timeout 到期时 rekey

组合行为

`auto-negotiate`	`keepalive`	对应行为
开启	强制开启	二阶段隧道会自动建立，无需流量触发 二阶段 soft-timeout 到期时发起 rekey
关闭	开启	二阶段隧道不会自动建立，需流量触发或对端发起 二阶段 soft-timeout 到期时发起 rekey
关闭	关闭	二阶段隧道不会自动建立，需流量触发 若最近 2 分钟内无流量，在 hard-timeout 到期时删除 SA，但不会 rekey 若最近 2 分钟内有流量，在 hard-timeout 到期时 rekey

配置建议

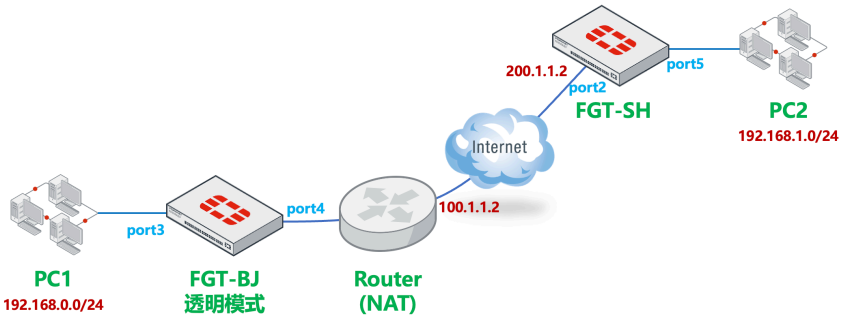
- `auto-negotiate`：通常建议开启，如有流量付费或性能考虑可设置为关闭。
- `keepalive`：始终建议保持开启，防止二阶段隧道在 hard-timeout 到期后发生中断。

透明模式下的 IPSec VPN

网络需求

在 FortiGate 透明模式下，通过 IPsec VPN（策略模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。

网络拓扑



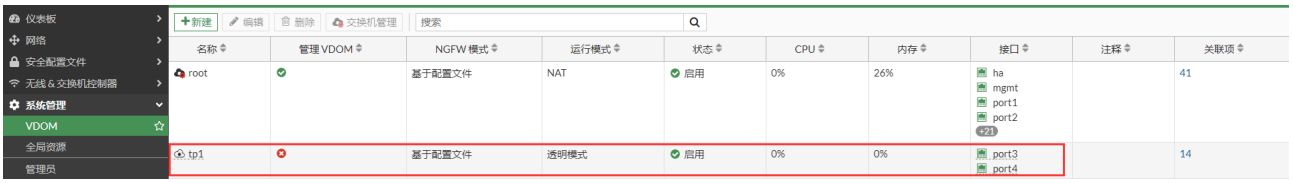
- FGT - BJ 是透明模式，PC1 的网关指向 Router。
- FGT - SH 是 NAT/路由模式。

配置步骤

FGT - BJ 配置

1.基本配置

为了便于测试，单独建立一个透明模式的 vdom，将 port3 和 port4 划入 tp1 vdom。





将设备修改为透明模式

> **重要**
>
> IPSec 将使用透明模式配置的管理 IP（manageip）作为 IPSec 的本地网关对 IKE 或 ESP 进行封装。
>
> 在此例中，FGT - BJ 的 IPSec 本地网关地址为透明模式的管理 IP 192.168.0.5，需要确保透明模式的管理 IP 与 Router（NAT）之间的连通性。

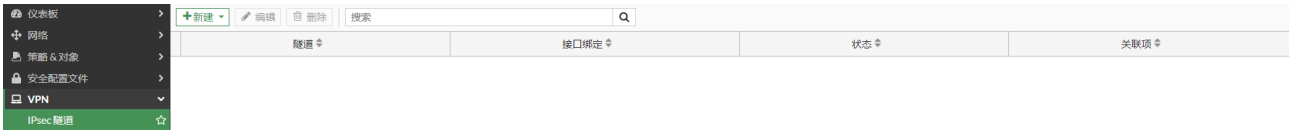
```
config system settings
    set opmode transparent
    set manageip 192.168.0.5/255.255.255.0
    set gateway 192.168.0.2
end
```

2.打开基于策略的 IPSEC



3.配置 IPSEC

“选择 VPN”-->"IPSEC 隧道"，点击“创建”



配置 IPSEC VPN 阶段一



配置 IPSEC VPN 阶段二

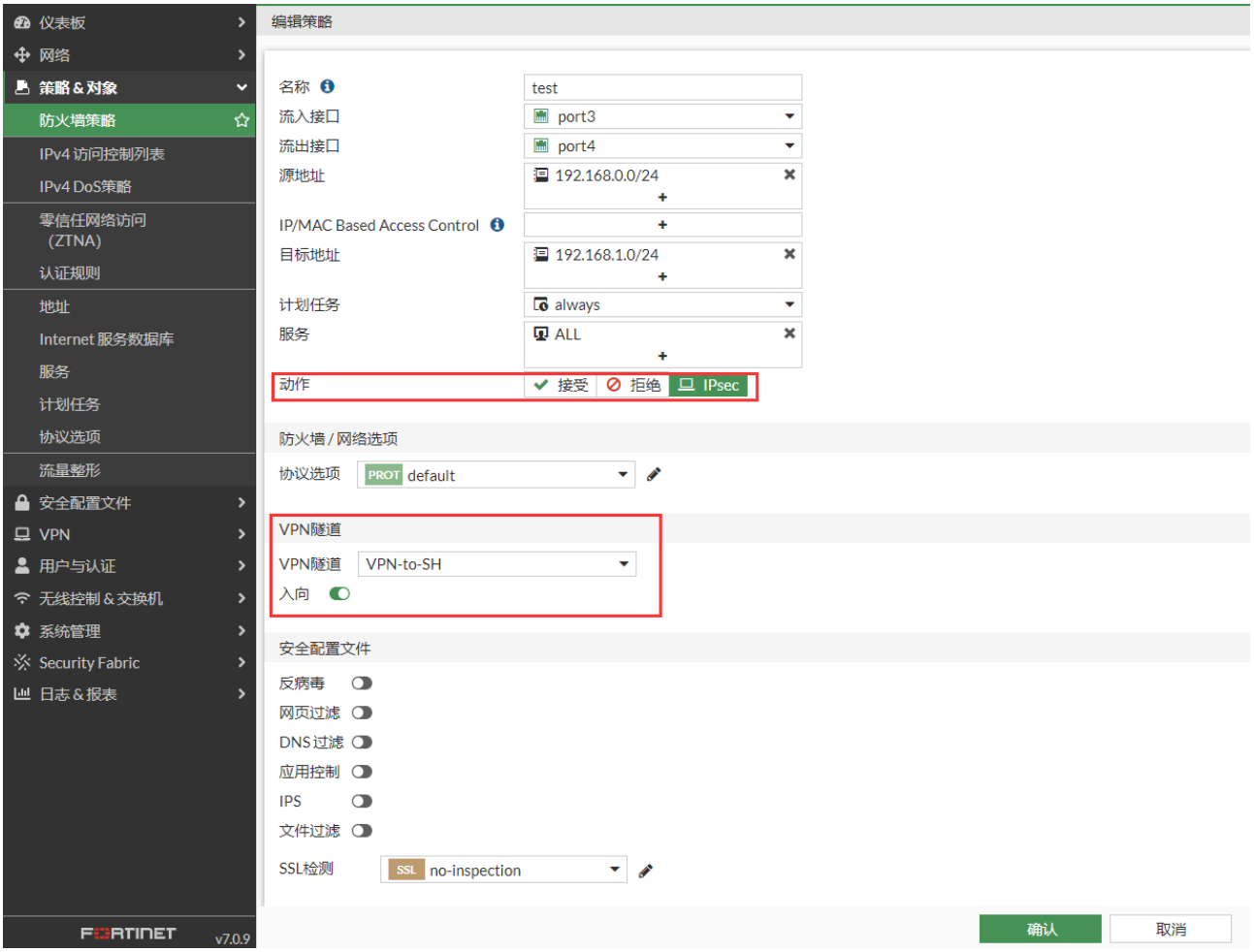


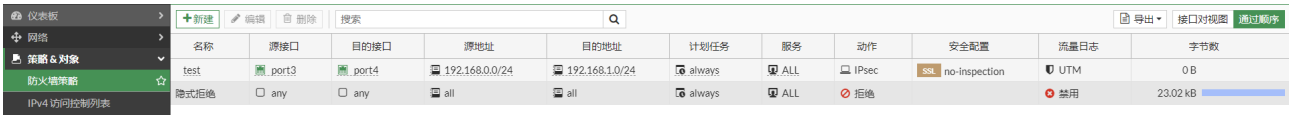
配置完成。



4.配置策略

新建策略时，动作要选择 IPSEC，指定 IPSEC 隧道，入向表示允许对端发起 IPSEC 流量，如果不勾选，对端发起的 IPSEC 流量将被拒绝。





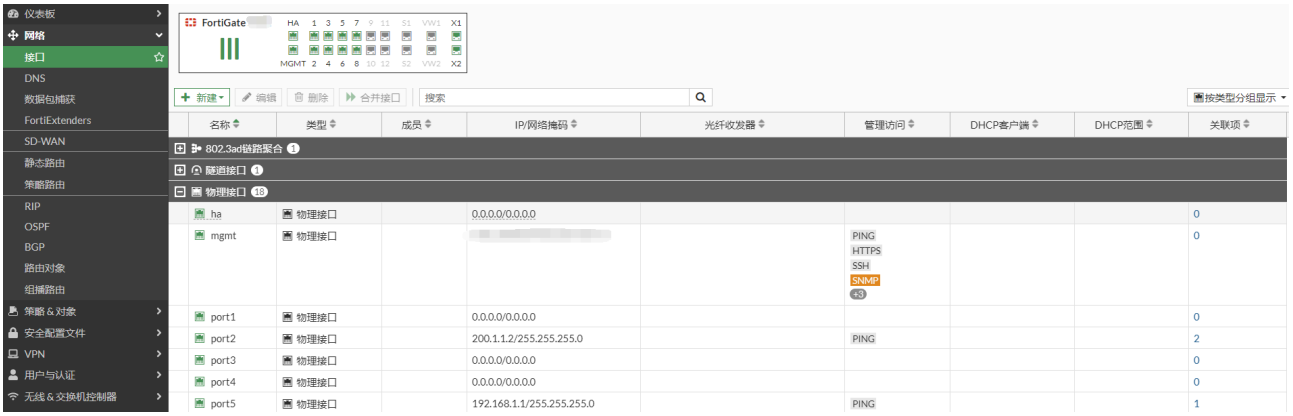
5. IPSEC 配置对应的命令行

```
config vpn ipsec phase1
    edit "VPN-to-SH"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 200.1.1.2
        set psksecret ENC
51lISs3KQ4m+sXPI9G+hwY8ysQEZUNApjT3dLeudcNwE3/ESIDyHJMeUAIBaMZ6aL6lrBiwleBYoEKnP+8gUusyE2nKqVwGYSdzsKHbycQyezu8a9zESIu3C
oVZA+FUiM+iUSDN0GHwZh6nMkl4g1rXYcLle5qzF0ybsKyYOIwmwovD0v9hBsw5HWMl06WcYnA5hVA==
        set dpd-retryinterval 10
    next
end

# show vpn ipsec phase2
config vpn ipsec phase2
    edit "VPN-to-SH"
        set phase1name "VPN-to-SH"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.0.0 255.255.255.0
        set dst-subnet 192.168.1.0 255.255.255.0
    next
end
```

FGT - SH 配置

1.基本配置

配置接口 ip 和路由。



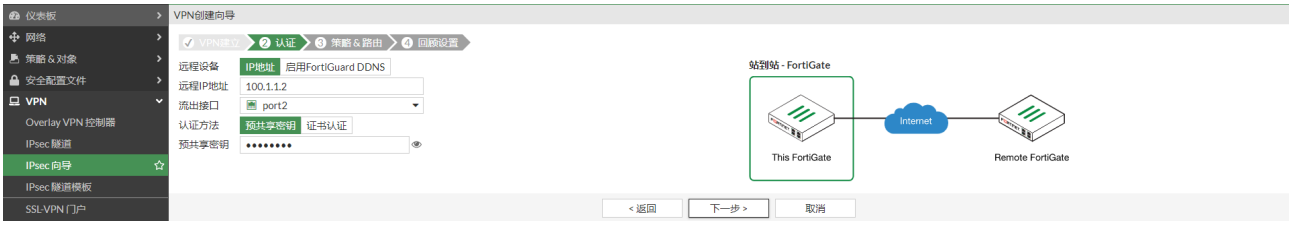


2.配置 IPSEC

选择“VPN”-->“IPSEC 隧道”，点击“创建”。



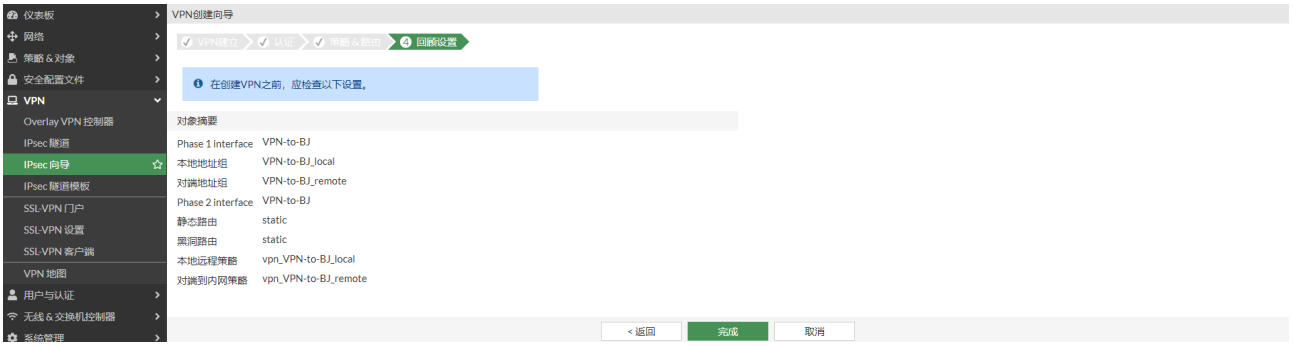
输入对端 FortiGate 的公网 IP（在本文拓扑中为 Router NAT 后的公网地址 100.1.1.2）及预共享秘钥。



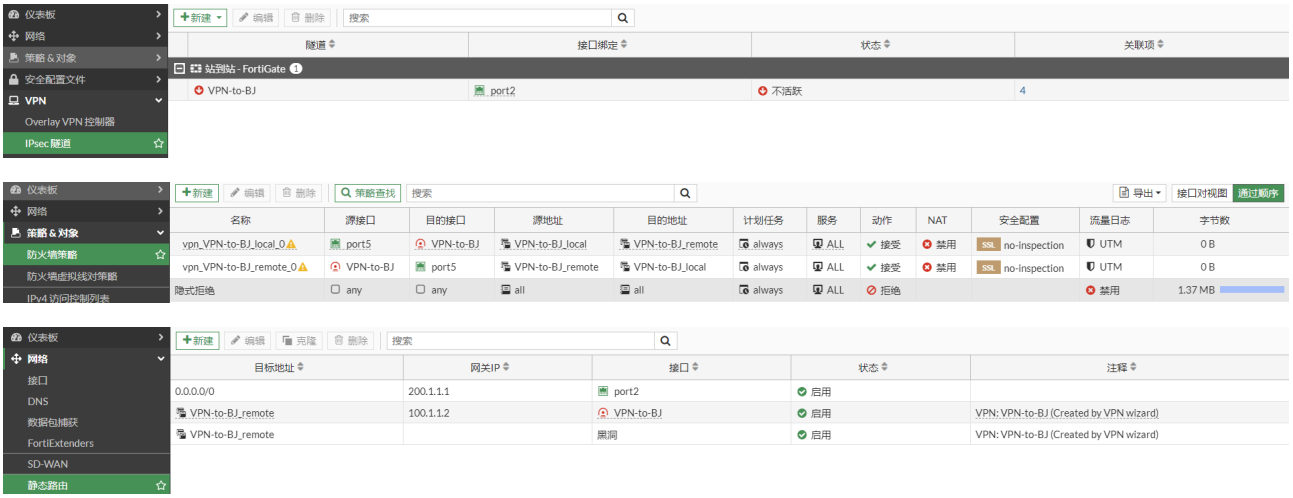
输入需要保护的网段和接口。



即将创建的内容。



IPSEC 模板创建 IPSEC VPN 隧道，策略，路由成功。



3. 针对向导配置的优化建议



4. IPSEC 配置对应的命令行

```
# show vpn ipsec phase1-interface
config vpn ipsec phase1-interface
    edit "VPN-to-BJ"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set comments "VPN: VPN-to-BJ (Created by VPN wizard)"
        set remote-gw 100.1.1.2
        set psksecret ENC
y6uhBx2lgYaxVJ4mkMu+Jf1M4a0a3XnIHULRXDaRBcvy6SVVyFksSsqg6ZdfXAp7okjT3btI8WATWijaxbM1T/1YHfU9gtI0vXFuBoc6Jij41ztBkohof+Ow
hJnb+WILWrNIGrgThqN/Su3RtyNk/ruiiW4hs3INa5ku0ZlT7StMlJBRoZ4zQ2XjMXETmOgdZnQR9Q==
        set dpd-retryinterval 10
    next
end

wangxiang-used2 # show vpn ipsec phase2-interface
config vpn ipsec phase2-interface
    edit "VPN-to-BJ"
        set phase1name "VPN-to-BJ"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "VPN-to-BJ_local"
        set dst-name "VPN-to-BJ_remote"
    next
end
```

查看 IPSEC 状态

FGT - BJ IPSEC 状态

GUI 查看 IPSEC 状态。



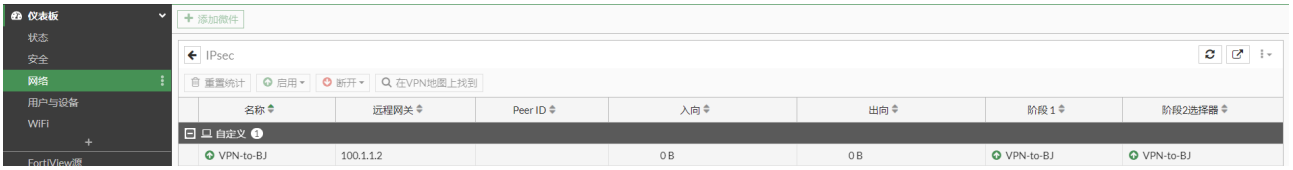
CLI 查看 IPSEC 状态

```
# diagnose vpn ike gateway list
vd: tp1/3
name: VPN-to-SH
version: 1
interface:  0
addr: 192.168.0.5:4500 → 200.1.1.2:4500
tun_id: 200.1.1.2/::200.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 522s ago
nat: me
IKE SA: created 1/1  established 1/1  time 9000/9000/9000 ms
IPsec SA: created 1/1  established 1/1  time 9000/9000/9000 ms

  id/spi: 88 df473084a6cd160a/fd52b18f1451abed
  direction: initiator
  status: established 522-513s ago = 9000ms
  proposal: aes128-sha256
  key: dbd02be2254a30c4-383ba553458819b3
  lifetime/rekey: 86400/85586
  DPD sent/recv: 00000009/0000002c

# diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
```

```
name=VPN-to-SH ver=1 serial=1 192.168.0.5:4500→200.1.1.2:4500 tun_id=200.1.1.2 tun_id6=::200.1.1.2 dst_mtu=1500 dpd-
link=on weight=0
bound_if=0 lgwy=static/1 tun=tunnel mode=auto/1 encap=none/8 options[0008]=npu  run_state=0 role=primary accept_traffic=1
overlay_id=0

proxyid_num=1 child_num=0 refcnt=7 ilast=4 olast=4 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=9
natt: mode=keepalive draft=32 interval=10 remote_port=4500
proxyid=VPN-to-SH proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:192.168.0.0-192.168.0.255:0
  dst: 0:192.168.1.0-192.168.1.255:0
  SA:  ref=3 options=18227 type=00 soft=0 mtu=1422 expire=42386/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=f77af5f3 esp=aes key=16 51994a2f09e97e1d834fba1454afce40
       ah=sha1 key=20 f75607ee1dde5d6a97d79d125fa2fbe77033e9d7
  enc: spi=e9cb7b26 esp=aes key=16 4c3df88982eb657c0ac2e851e26727b2
       ah=sha1 key=20 24392681766e619bcee614fb6e3b8318ae37d40e
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=192.168.0.5 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

FGT - SH IPSEC 状态

GUI 查看 IPSEC 状态。



CLI 查看 IPSEC 状态。

```
# diagnose vpn ike gateway list

vd: root/0
name: VPN-to-BJ
version: 1
interface: port2 10
addr: 200.1.1.2:4500 → 100.1.1.2:64916
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 563s ago
nat: peer
IKE SA: created 1/2  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 14 df473084a6cd160a/fd52b18f1451abed
  direction: responder
  status: established 544-544s ago = 0ms
  proposal: aes128-sha256
  key: dbd02be2254a30c4-383ba553458819b3
  lifetime/rekey: 86400/85585
  DPD sent/recv: 0000002f/00000009

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-BJ ver=1 serial=3 200.1.1.2:4500→100.1.1.2:64916 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-
link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=4 olast=4 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=47
natt: mode=silent draft=32 interval=10 remote_port=64916
```

```
proxyid=VPN-to-BJ proto=0 sa=1 ref=2 serial=1
  src: 0:192.168.1.0-192.168.1.255:0
  dst: 0:192.168.0.0-192.168.0.255:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1422 expire=42386/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42933/43200
  dec: spi=e9cb7b26 esp=aes key=16 4c3df88982eb657c0ac2e851e26727b2
       ah=sha1 key=20 24392681766e619bcee614fb6e3b8318ae37d40e
  enc: spi=f77af5f3 esp=aes key=16 51994a2f09e97e1d834fba1454afce40
       ah=sha1 key=20 f75607ee1dde5d6a97d79d125fa2fbe77033e9d7
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=200.1.1.2 npu_selid=2 dec_npuid=0 enc_npuid=0
run_tally=0
```

业务测试

1.从 FGT - BJ 端 ping FGT - SH 端

```
# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6352519  bytes 1854154972 (1.7 GiB)
        RX errors 0  dropped 3749690  overruns 0  frame 0
        TX packets 130448  bytes 8786148 (8.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC1# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=63 time=0.717 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=63 time=0.853 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=63 time=0.720 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=63 time=0.752 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.717/0.760/0.853/0.061 ms
```

2.从 FGT - SH 端 ping FGT - BJ 端

```
# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::1a6c:e61:d2b9:a415  prefixlen 64  scopeid 0×20<link>
        inet6 2001::2  prefixlen 64  scopeid 0×0<global>
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 6040640  bytes 490806558 (468.0 MiB)
        RX errors 0  dropped 145  overruns 0  frame 0
        TX packets 3246663  bytes 205412788614 (191.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=63 time=0.790 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=63 time=0.879 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=63 time=0.686 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=63 time=0.855 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.686/0.802/0.879/0.079 ms
```

重命名 IPSec 隧道

功能简介

本文介绍如何在 FortiGate 上重命名已创建的 IPSec 隧道（`phase1-interface`）。

- FortiOS 7.4.2 之前：已创建的 IPSec 隧道不支持直接重命名，需要删除关联配置及自身配置后，使用新名称重建。
- FortiOS 7.4.2 及之后：支持通过 GUI 和 CLI 直接重命名 IPSec 隧道，与隧道关联的配置会自动修改。

配置步骤

GUI

进入“VPN → VPN 隧道”，编辑目标隧道后修改`名称`字段并保存即可。



CLI

```bash
config vpn ipsec phase1-interface
    rename <old-tunnel-name> to <new-tunnel-name>
end

config vpn ipsec phase1-interface
    rename SiteA to SiteB
end
```

注意事项

1. 修改 IPSec VPN 名称后，对应的 IPSec Tunnel 接口名称也会对应修改。
2. 与隧道关联的配置，如防火墙策略中的入/出接口、静态路由或策略路由中的出接口、SD-WAN、监控等对应配置中的接口名称等均会自动变化。

点到点 VPN

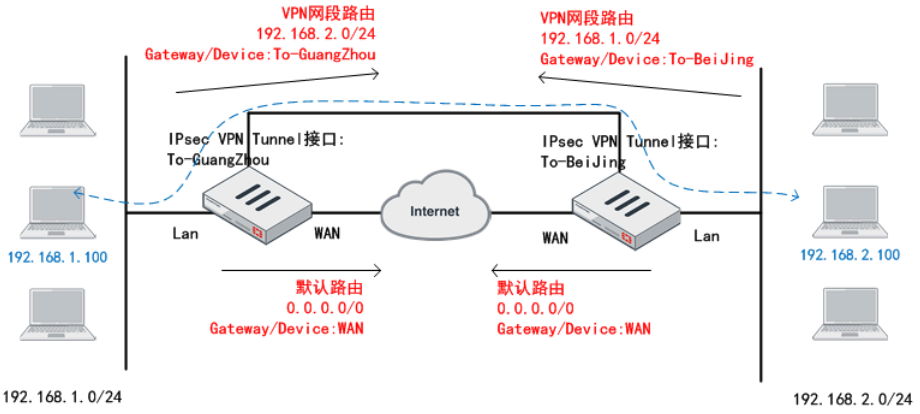
- IPSec 的两种配置模式
- 接口摸式
- PKI 证书认证
- VPN 隧道上运行 OSPF 或 BGP
- 多路负载/主备 IPSec VPN
- FortiGuard DDNS 方式建立 IPSec VPN
- GRE over IPSec
- 修改 IPSec 协商端口

IPSec 的两种配置模式

FortiGate IPsec VPN 的两种配置模式

接口模式（默认，推荐使用）：

接口模式作为默认的 IPsec VPN 方式，FortiGate 会自动创建一个 IPsec VPN Tunnel 接口，然后通过路由将感兴趣流指向 IPsec VPN Tunnel，什么样的感兴趣流走 IPsec VPN，由路由来决定，配置简单易懂，使用非常灵活，推荐使用。

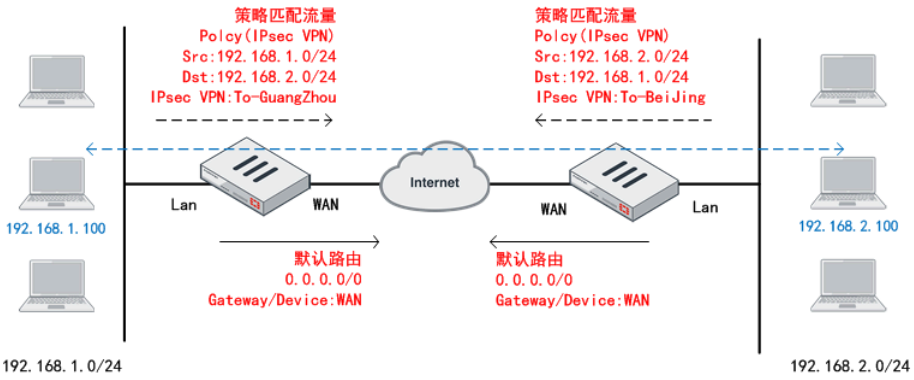


LAN 和 To-GuangZhou 的接口之间通信需满足：

- IPsec VPN 隧道 UP
- 路由正确
- 策略放通

策略模式（默认隐藏，需要在 Features 中开启才能使用，不推荐使用）：

通过策略匹配出感兴趣流，然后再策略中调用 IPsec VPN，是一种较为传统的 IPsec VPN 连接方式，一般较少使用，通常推荐使用接口模式 IPsec VPN。



IPsec VPN 建立起来之后，通过策略将流量匹配，只有感兴趣流才走 IPsec VPN，而其他普通数据则走普通的路由从 WAN 口出去，不会自动生成 IPsec tunnel 接口，因此不再需要也不能配置指向 VPN tunnel 的路由。

接口模式 VS 策略模式

为什么 FGT 默认把策略模式隐藏，而推荐大家使用接口模式呢？

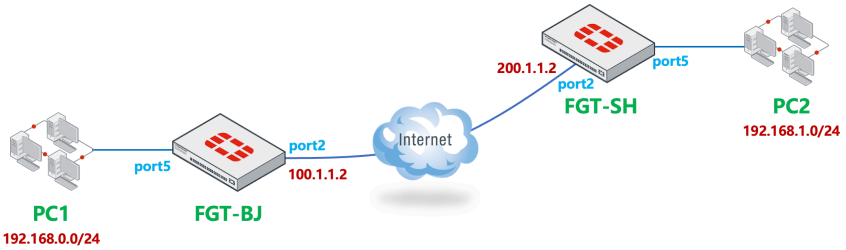
- 策略模式能实现的，几乎接口模式都能实现；
- 接口模式能实现的，策略模式不一定能够实现，例如：GRE OVER IPSEC、隧道上直接跑动态路由协议（RIP/OSPF/BGP）；
- 对于安全设备来说，生成一个 VPN-Tunnel 接口，策略配置起来的时候，思路更加简单清晰；
- 第二阶段感兴趣流可以配置为 any_to_any，感兴趣流量通过路由来引导，而不需要定义与配置复杂的第二阶段感兴趣流；
- 接口模式支持 ADVPN，更加适应各种复杂的组网环境；
- 接口模式支持 IPv4 和 IPv6，而策略模式仅支持 IPv4。

接口模式

网络需求

通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。

网络拓扑



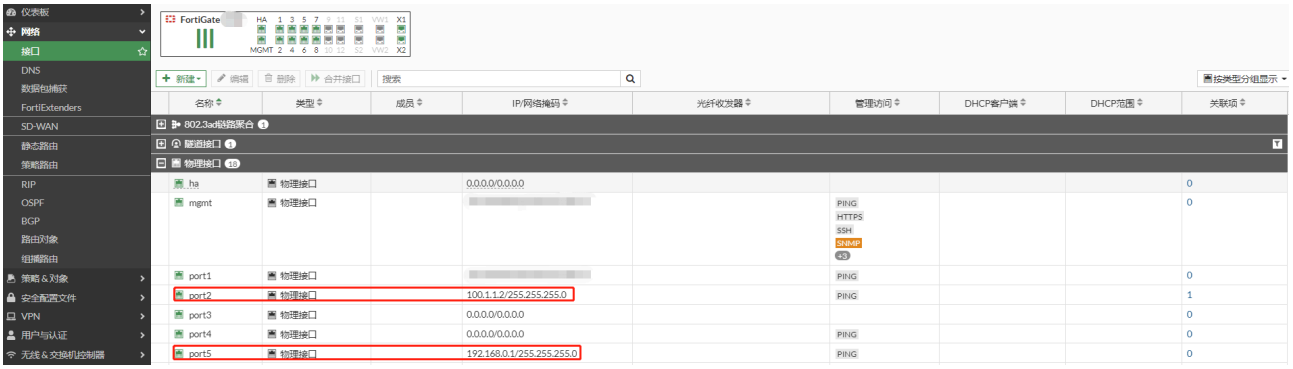
配置步骤

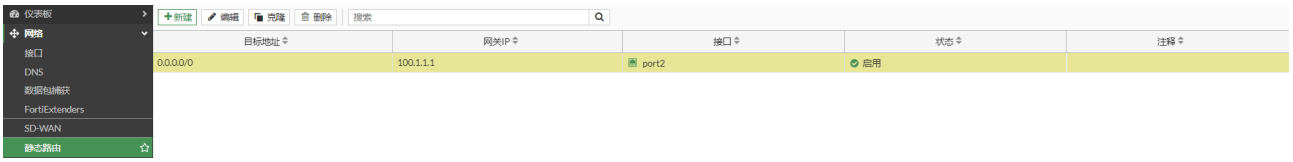
IPSec 配置

以 FGT - BJ 配置步骤为例，FGT - SH 同理。

1.基本配置

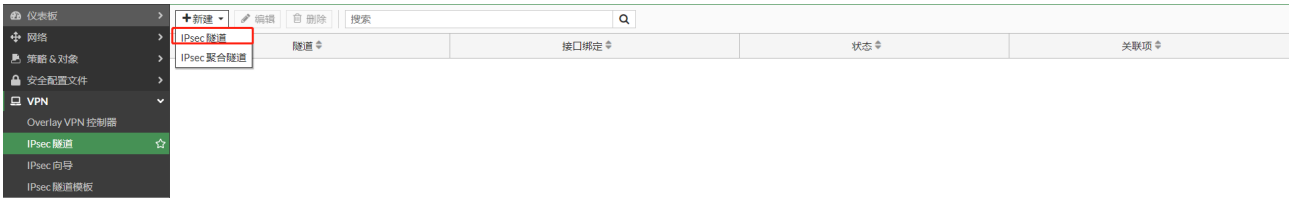
配置接口 IP 和路由



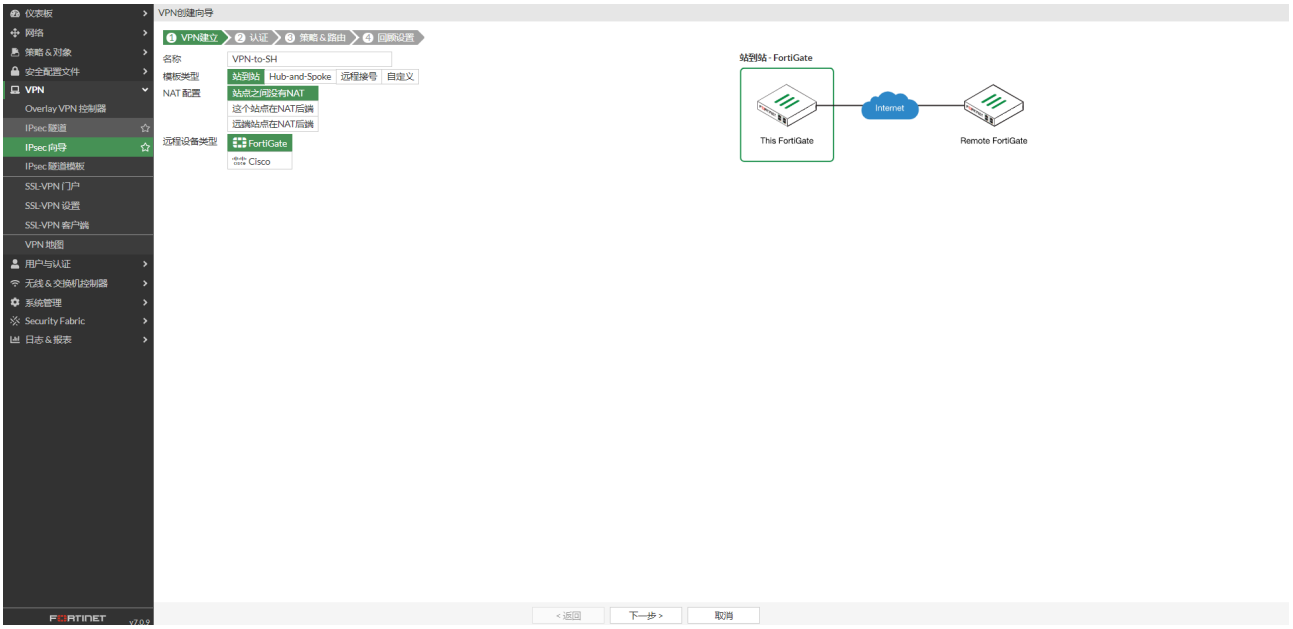


2.配置 IPSEC VPN

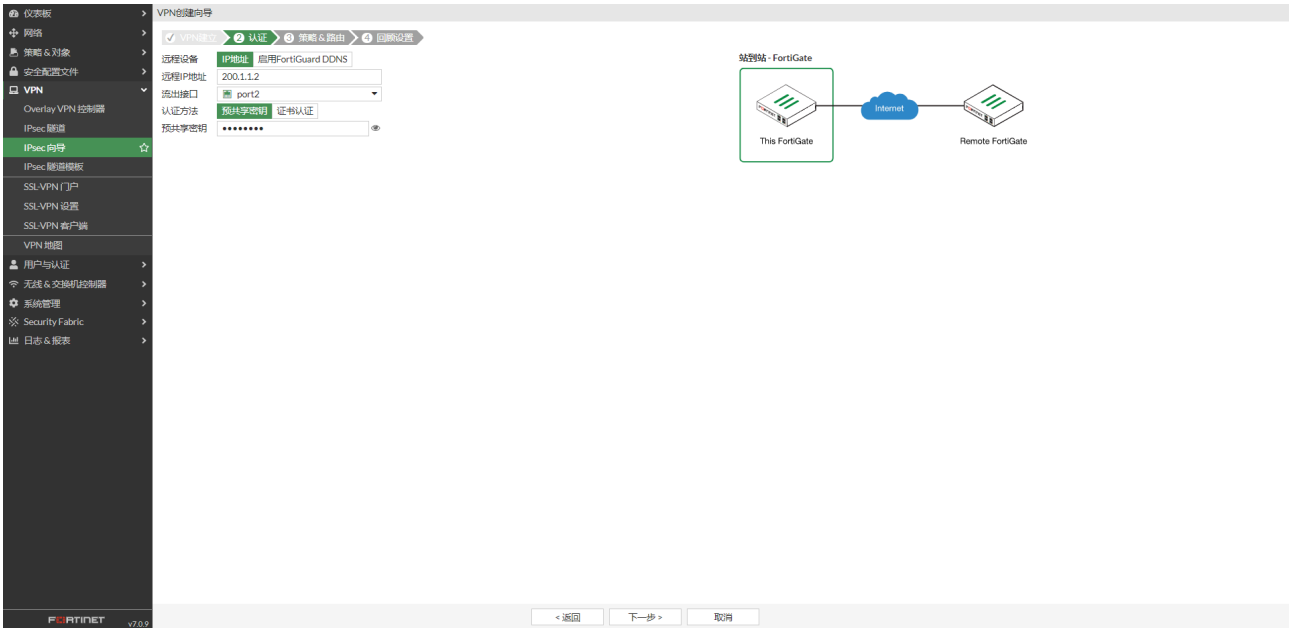
选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



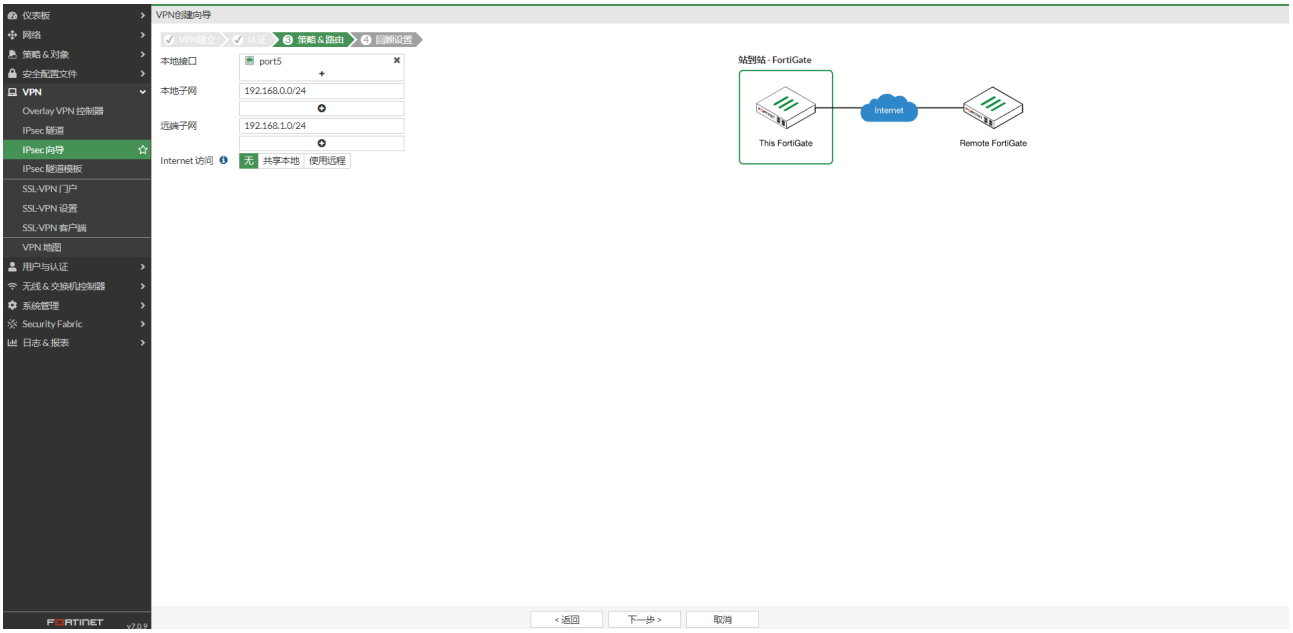
根据“VPN 创建向导”进行 VPN 模板配置，输入名称，拓扑中没有 NAT，这里选择“站点之间没有 NAT”，并点击下一步。



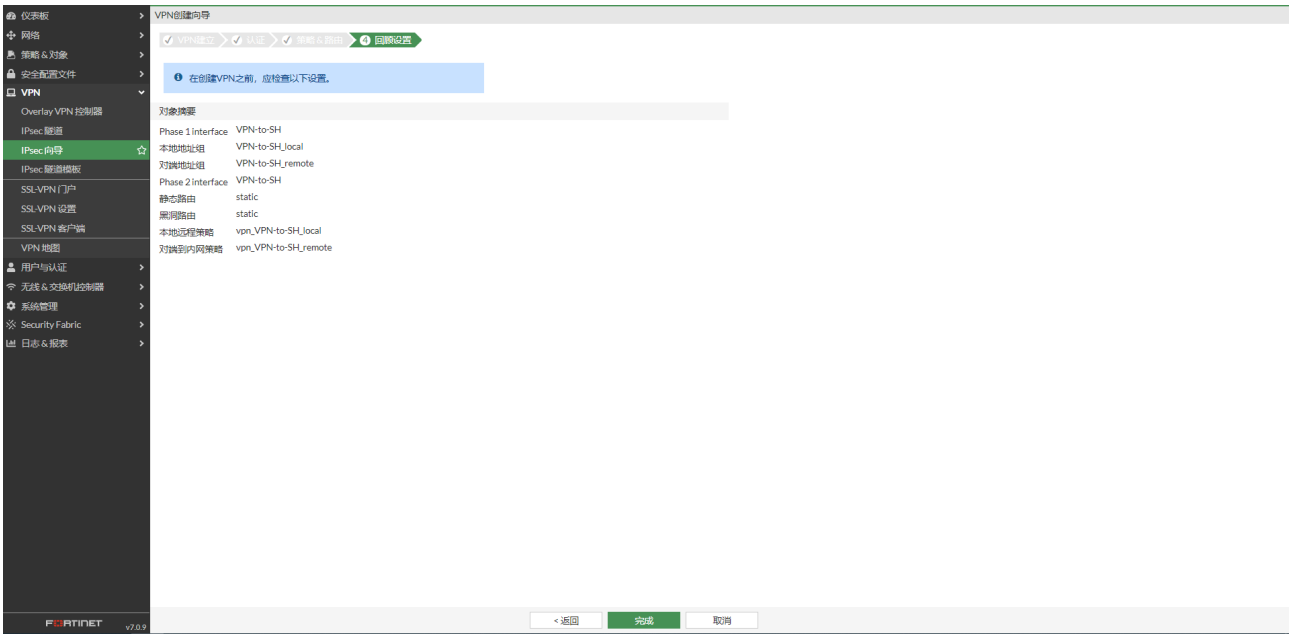
输入对端设备的 IP 地址（流出接口会根据路由自动选择），以及预共享秘钥。



选择“本地接口”，即需要被 IPSEC 保护的内网的接口，这里是 port5 接口，本地子网会自动写为该接口的 IP 所在的 IP 网段（如果被保护的是其他网段，可以根据实际网络需求修改），输入“远端子网”，即远端需要被保护的网络。然后点击“下一步”。



VPN 创建向导提示即将创建的内容，然后点击完成。



VPN 创建成功。



重要

如果最后完成模板部署的时候报错，那么需要注意，不要有旧的配置存在，比如相关 VPN 的策略、VPN 的路由、VPN 的 IP 地址对象，因为会冲突，所以需要把这些内容删除干净才可以顺利的部署向导模板，一般第一次部署不会遇到报错。

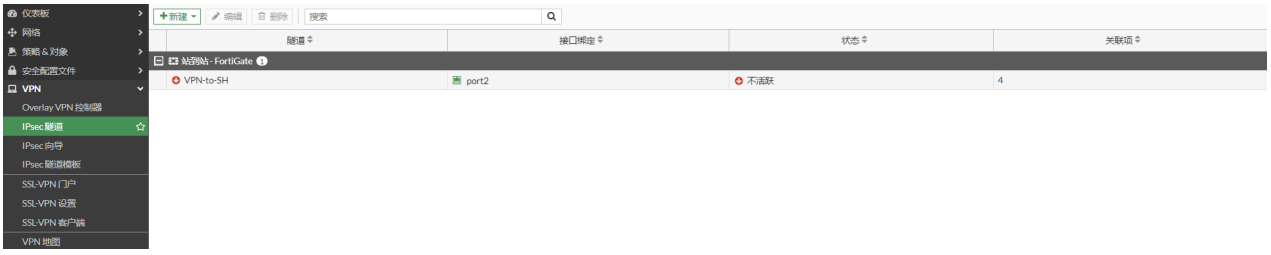
查看 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1. 创建地址对象和地址对象组



2. 创建 IPSEC 隧道



对应的命令行如下：

```bash
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: VPN-to-SH (Created by VPN wizard)"
        set wizard-type static-fortigate
        set remote-gw 200.1.1.2
        set psksecret ENC
feA2LOY0w7mUVHbFI06AZkRjLX80Di1eIIk+FiRIezwPpWU/5oSuf/2LUOIRtrDxOIoyTN+kj5TYADnJOL/K6zQ82ektaS146SvvvZQ9umHQTJJSW7HtDfDq
ic1tZAImYJ+APdWC090tUIyc3z5qgcarmPUALfbDSZkmpzY6AtSLMJGHcGVkY4rVTl0GPNekUB/gUg==
    next
```

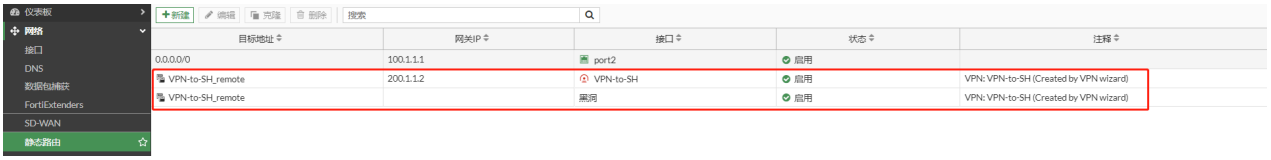
```
end
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set phase1name "VPN-to-SH"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: VPN-to-SH (Created by VPN wizard)"
        set src-addr-type name
        set dst-addr-type name
        set src-name "VPN-to-SH_local"
        set dst-name "VPN-to-SH_remote"
    next
end
```

3. 创建路由

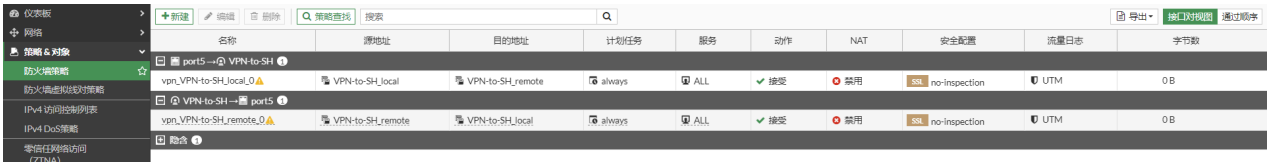
这里到对端的子网创建了两条路由：一条下一跳是 VPN-to-SH 接口的路由，一条黑洞路由。

> **重要**
>
> IPsec VPN 隧道的路由下一跳只是一个隧道 ID 标识，并不是真正的路由下一跳 IP，这和我们普通的路由下一跳不同，如这里的 200.1.1.2 是隧道的 ID。

黑洞路由的目的：VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量会因为查询到了默认路由而走向了 port2（Internet），从而产生了不期望的会话，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，业务流量会匹配不期望的会话而走向 port2，从而引起该业务异常。因此有了这条黑洞路由就不会产生这条不期望的会话。

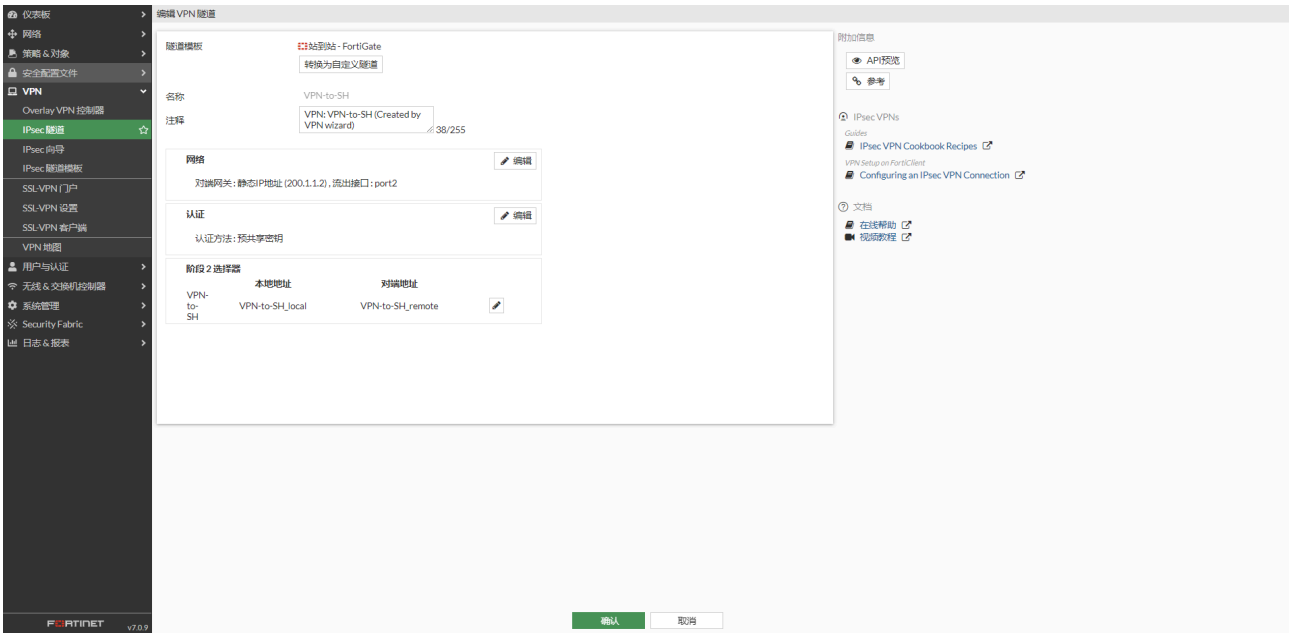


4. 创建策略



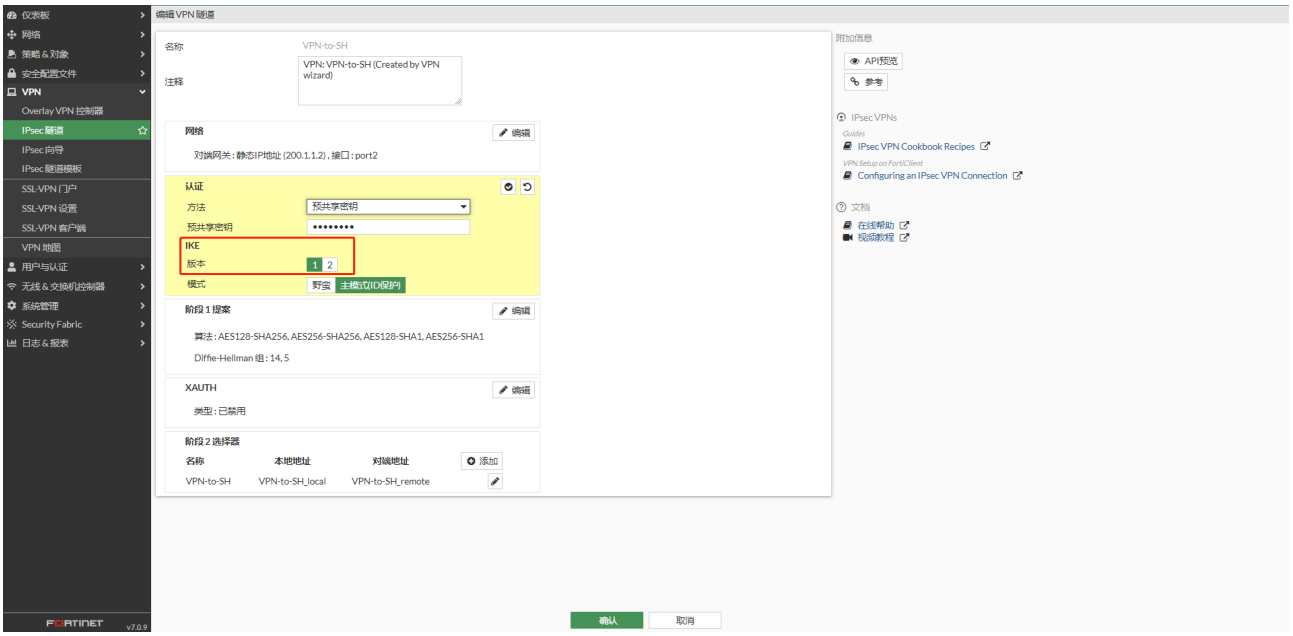
修改 IPSEC VPN 配置

编辑 VPN 隧道如下图

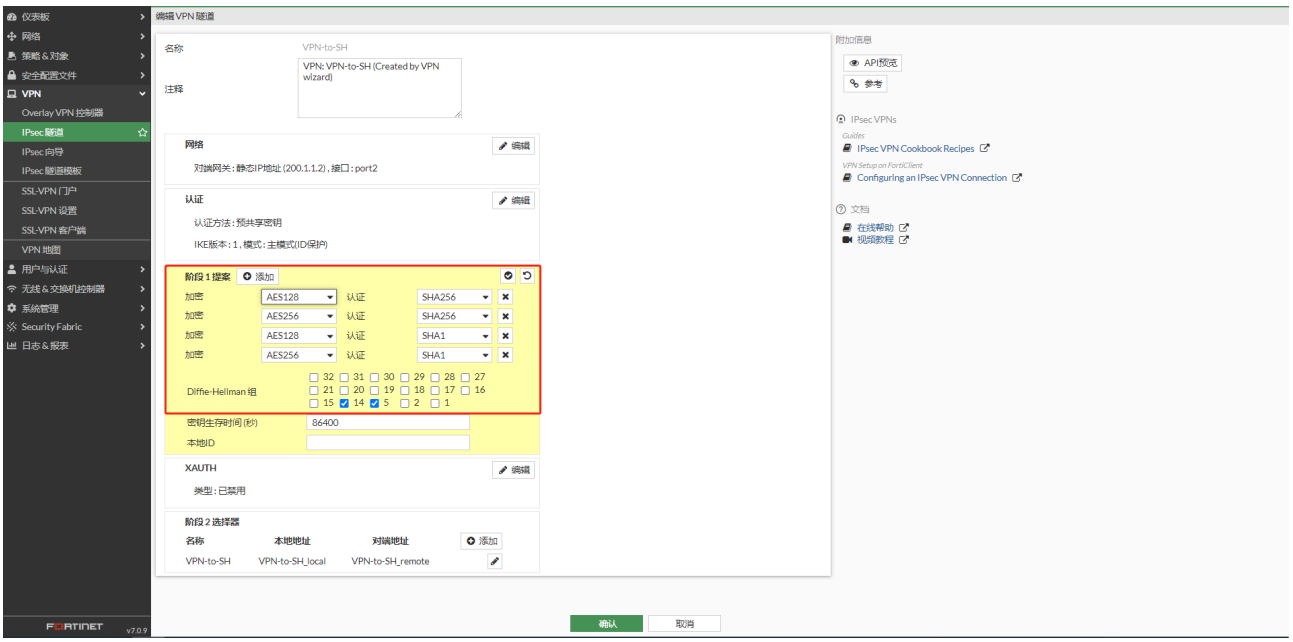


通过点击“转换为自定义隧道”可以在 GUI 界面设置的更多，是否需要修改可以根据实际网络需求决定。

如 IKE 版本（默认使用 IKEv1）



如修改阶段一加密和 HASH 算法



针对向导配置的优化建议

1. 在第一阶段中配置 DPD 模式为“空闲”（on-idle）（IPSec 隧道内没有流量时发 DPD 探测，只有当 IPSec 隧道内双向有流量时，才不发 DPD 探测），实现快速的检测并切换 VPN 隧道的目的。



对应的命令行

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

2. 开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

IPSEC VPN 阶段一自动协商是默认开启的。

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

查看 IPSEC 状态

GUI 查看 IPSEC 状态



CLI 查看 IPSEC 状态

```
# diagnose vpn ike gateway list

vd: root/0
name: VPN-to-SH
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::200.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 1547s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/1  time 0/0/0 ms

  id/spi: 1428 5264c8b1b5ea006d/587ec20848be628f
  direction: initiator
  status: established 1547-1547s ago = 0ms
  proposal: aes128-sha256
  key: 7646944159f3ebe9-5eb6656064941ede
  lifetime/rekey: 86400/84552
  DPD sent/recv: 000000ec/00000000

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-SH ver=1 serial=3 100.1.1.2:0→200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::200.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=9 olast=9 ad=/0
stat: rxp=1 txp=13 rxb=84 txb=17290
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=236
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-SH proto=0 sa=1 ref=2 serial=3 auto-negotiate
  src: 0:192.168.0.0-192.168.0.255:0
  dst: 0:192.168.1.0-192.168.1.255:0
  SA:  ref=6 options=18227 type=00 soft=0 mtu=1438 expire=41368/0B replaywin=2048
       seqno=e esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=0df9349a esp=aes key=16 6d4d64a051d4c6e4b1dce3c019e9c6c7
       ah=sha1 key=20 f307e4af948aed97e217c89295f50edd78166092
  enc: spi=58e500e1 esp=aes key=16 e751f770f16484e877dd63dd86576d76
       ah=sha1 key=20 3a38ffe6cacee7cae62d04c70aeeb82c9b6a91ff
  dec:pkts/bytes=2/168, enc:pkts/bytes=26/19266
  npu_flag=03 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=9 dec_npuid=1 enc_npuid=1
run_tally=0
```

查看路由

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
S       192.168.1.0/24 [10/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]        ←——到192.168.1.0/24走IPSEC VPN
```

指定 IPSec 本端的协商 IP

如果 IPSec 绑定在 WAN1 接口上，但是想使用 Loopback 接口的 IP 与对端建立 IPSec，应该怎么办呢？这种需求可以通过配置 local-gw 来实现。如下示例中：

1. FortiGate 创建的 IPSec 绑定在 WAN1 口上，但 wan1 口的地址无法和 IPSec 对端设备直接通信，只有设备上的 Loopback 接口可以与对端设备通信。

```bash
config system interface
    edit "wan1"
        set vdom "root"
        set ip 192.168.1.100 255.255.255.255
        set allowaccess ping
    next
    edit "Loopback1"
        set vdom "root"
        set ip 172.22.6.155 255.255.255.255
        set allowaccess ping
        set type loopback
    next
end
```

2. 配置 IPSec 隧道时，通过 local-gw 来指定本端协商 IPSec 时所使用的源 IP 地址。

> **重要**
>
> 在硬件 FortiGate 上，接口（Interface）不要选择 Loopback 接口，会导致 IPSec 流量无法被 NP 加速，本地网关（local-gw）可以选择 Loopback 接口。




```bash
config vpn ipsec phase1-interface
    edit "to_Home_WAN1"
        set interface "wan1"
        set local-gw 172.22.6.129
    next
end
```

3. IPSec 协商与业务流量仍然从 wan1 接口发出，但源 IP 会使用 Loopback1 接口的 IP。

```bash
FortiGate # diagnose sniffer packet any 'host 123.112.245.210 and udp' 4
interfaces=[any]
filters=[host 123.112.245.210 and udp]
18.804385 wan1 out 172.22.6.129.500 → 123.112.245.210.500: udp 649
18.816603 wan1 in 123.112.245.210.500 → 172.22.6.129.500: udp 432
18.819713 wan1 out 172.22.6.129.4500 → 123.112.245.210.4500: udp 276
18.828528 wan1 in 123.112.245.210.4500 → 172.22.6.129.4500: udp 244
```

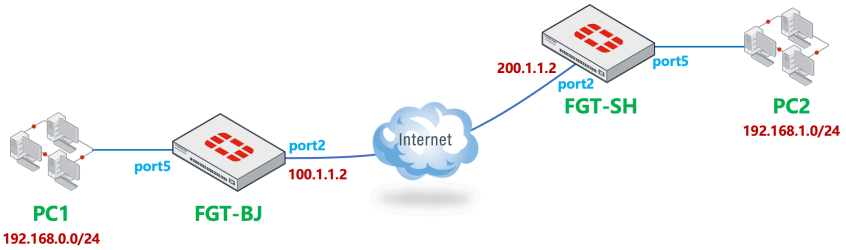
```
18.974296 wan1 out 172.22.6.129.4500 → 123.112.245.210.4500: udp 120
21.877118 wan1 in 123.112.245.210.4500 → 172.22.6.129.4500: udp 104
21.877318 wan1 out 172.22.6.129.4500 → 123.112.245.210.4500: udp 104
21.882354 wan1 in 123.112.245.210.4500 → 172.22.6.129.4500: udp 104
...
```

PKI 证书认证

网络需求

通过 IPsec VPN 将 2 个局域网连接起来，IPsec VPN 隧道需要采用证书认证方式来协商，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。

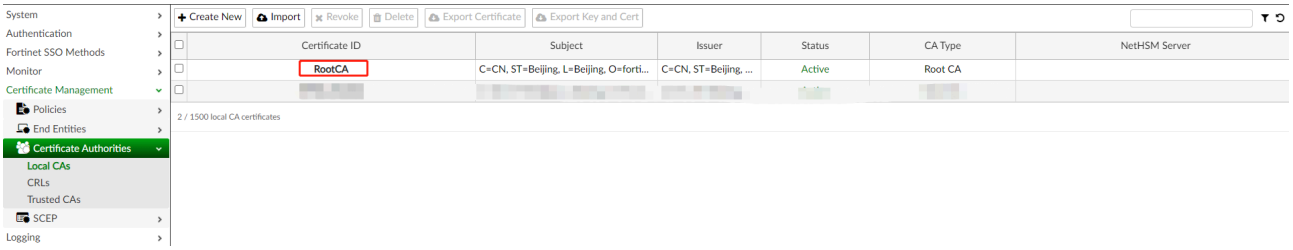
网络拓扑



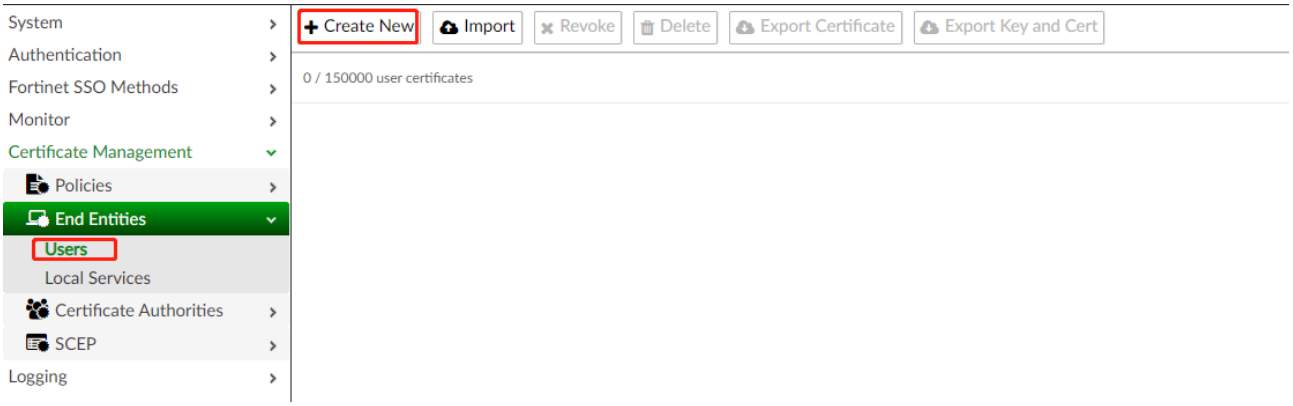
配置步骤

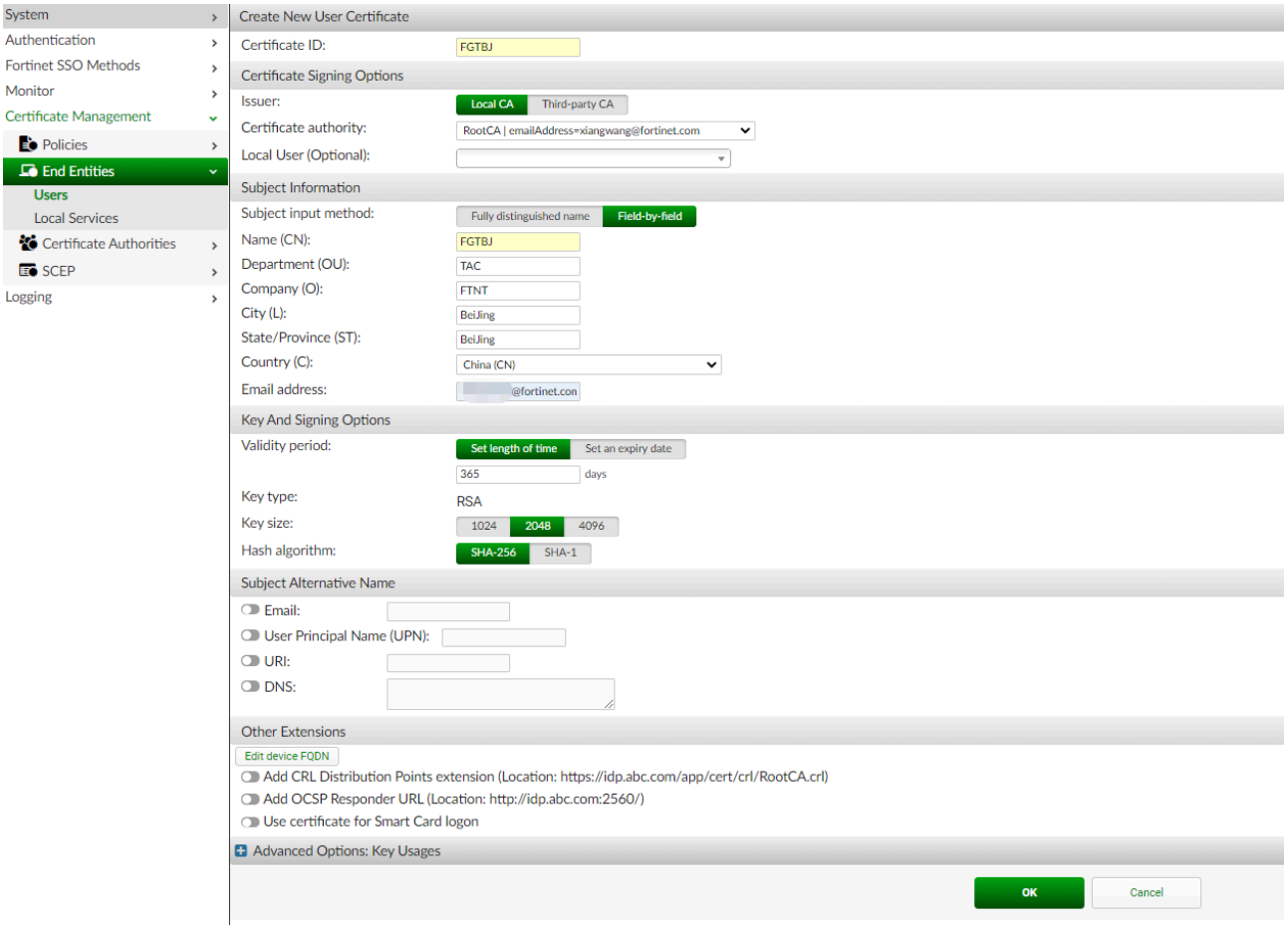
使用 FortiAuthenticator 为 FGT - BJ 和 FGT - SH 颁发证书

FortiAuthenticator 可以做为证书服务器，RootCA 是已创建好的根证书，这里使用 RootCA 为这两台 FortiGate 签发证书。（也可以使用其他证书服务器）

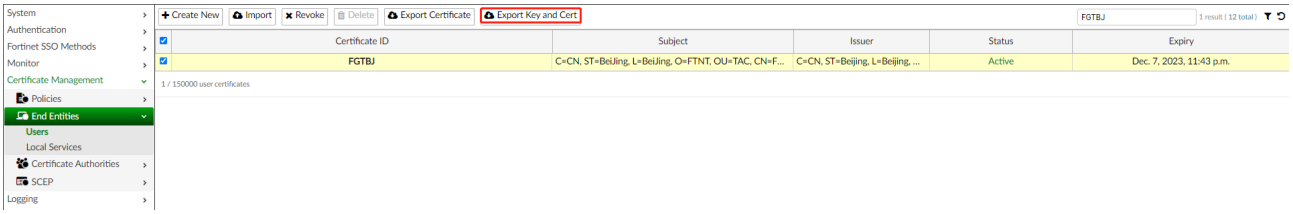


为 FGT - BJ 创建证书，FGT - SH 同理。选择“End Entities”-->"User"，点击“Create New”，输入证书相关信息，并点击完成。

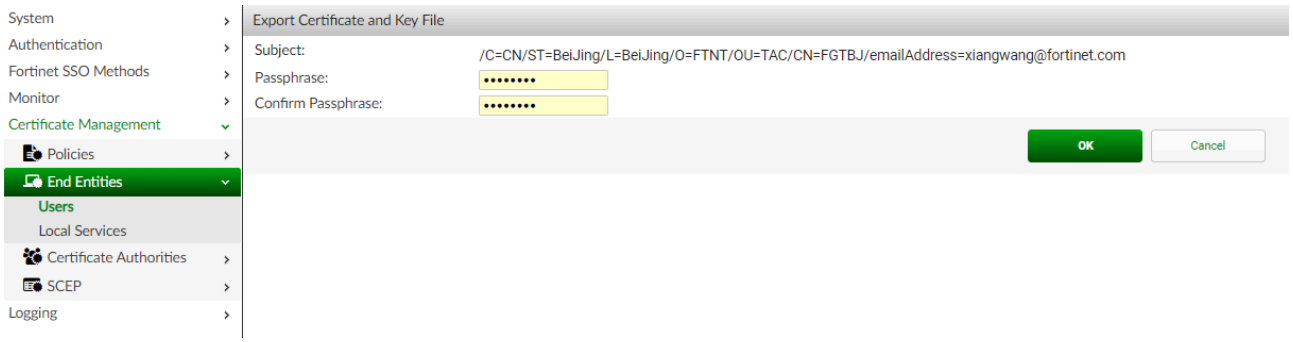




点击“Export Key and Cert”导出 FGTBJ 证书的私钥和公钥。



输入一个密码保护私钥



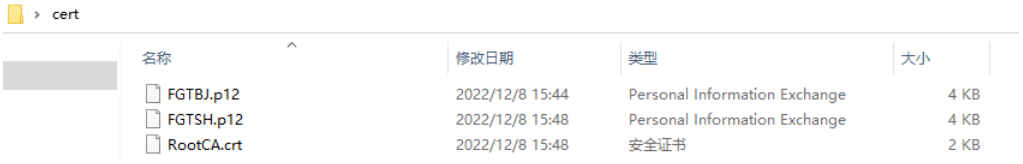
下载证书



FortiGate 导入证书

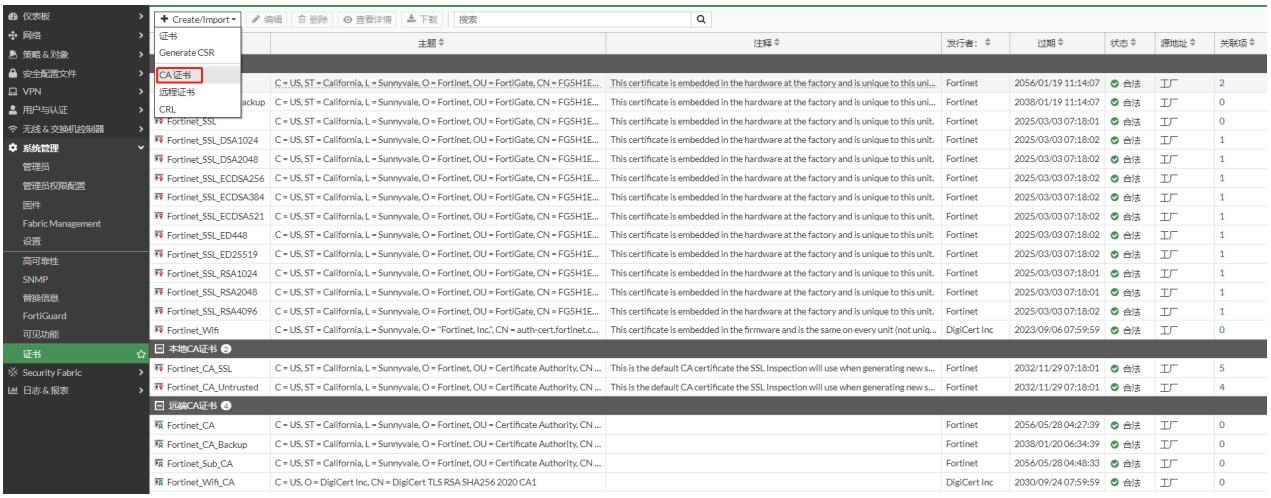
将根证书 RootCA.crt 和证书 FGTBJ.p12 导入到 FGT_BJ。同理将根证书 RootCA.crt 和证书 FGTSH.p12 导入到 FGT_SH。

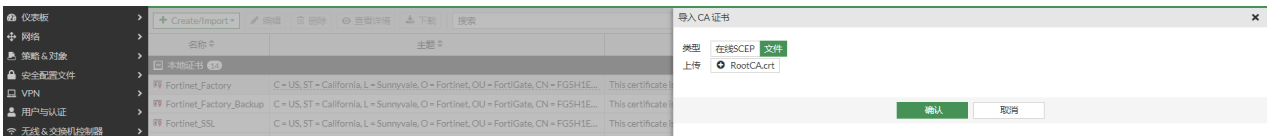
在数字证书认证中，通信双方使用 CA 证书进行数字证书合法性验证，双方各有自己的公钥（网络上传输）和私钥（本地存储）。发送方对报文进行 Hash 计算，并用自己的私钥对报文计算结果进行加密，生成数字签名。接收方使用发送方的公钥对数字签名进行解密，并对报文进行 Hash 计算，判断计算结果与解密后的结果是否相同。如果相同，则认证通过；否则认证失败。



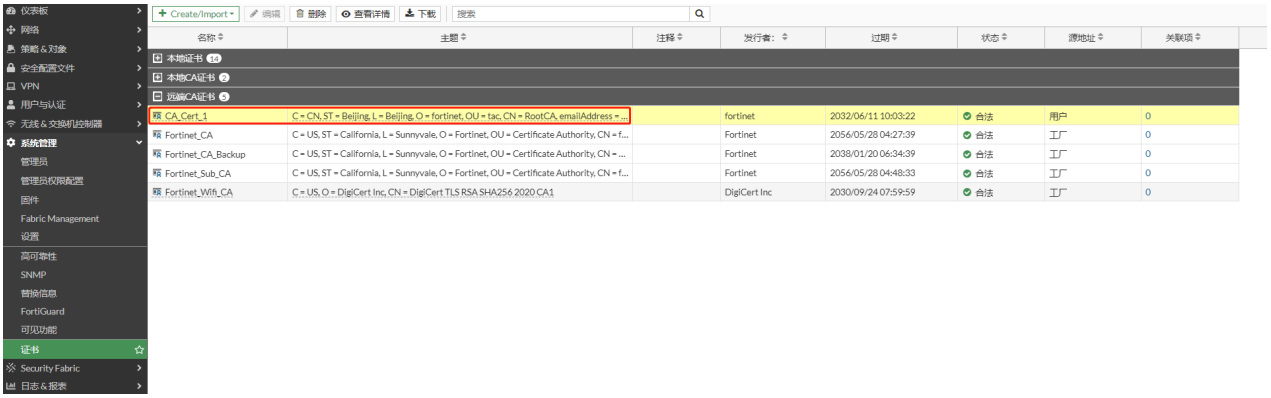
1. 导入 CA 证书

选择“系统管理”-->“证书”，点击“Create/Import”，选择 CA 证书。



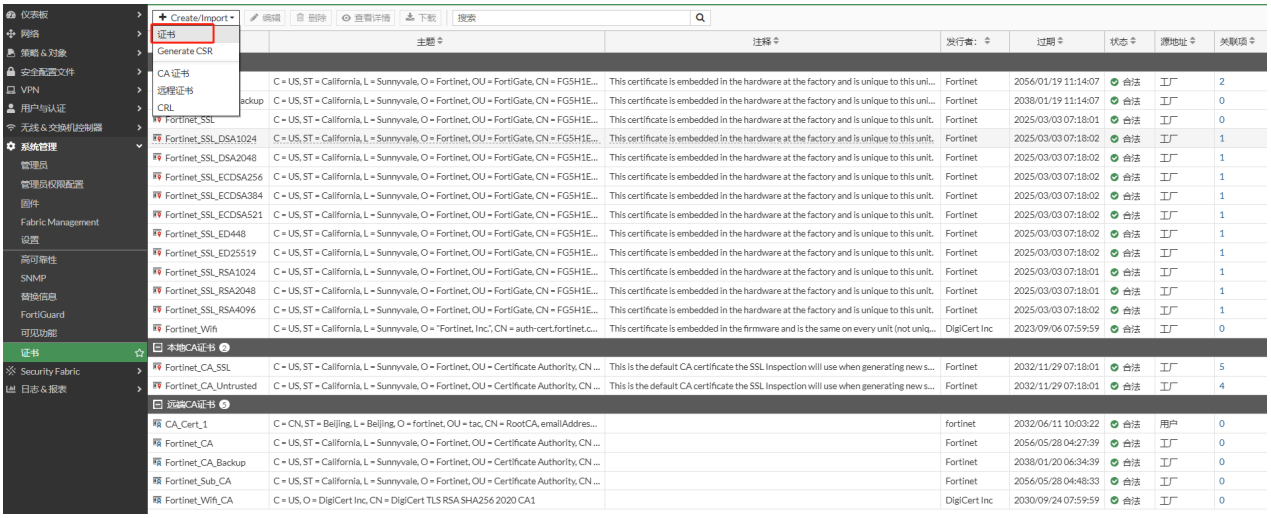


查看导入的 CA 证书

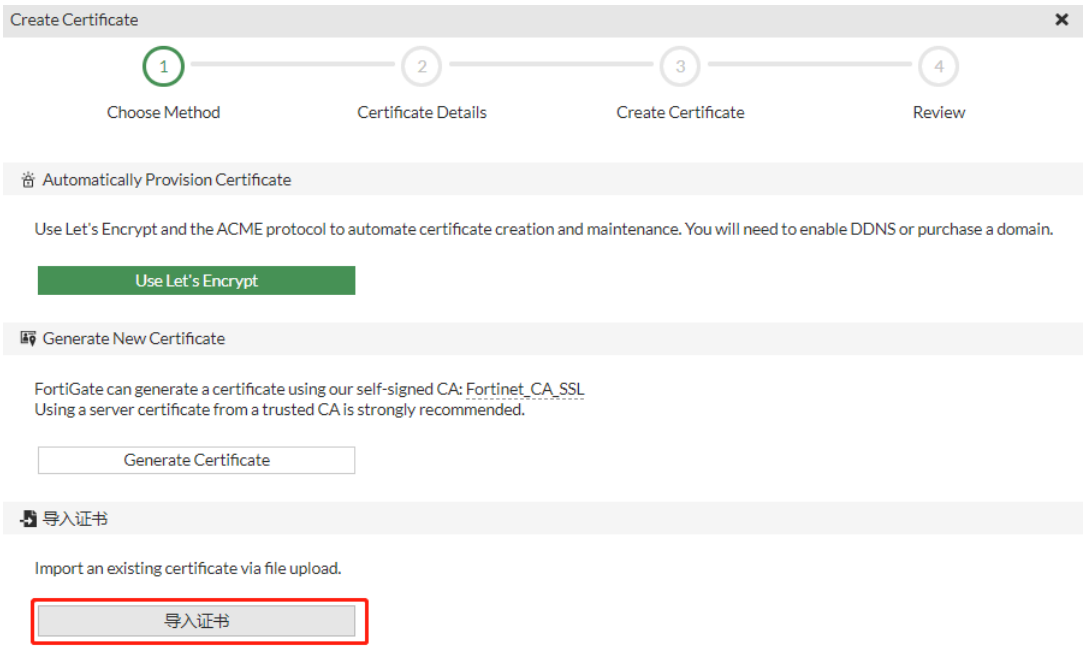


2. 导入本地证书

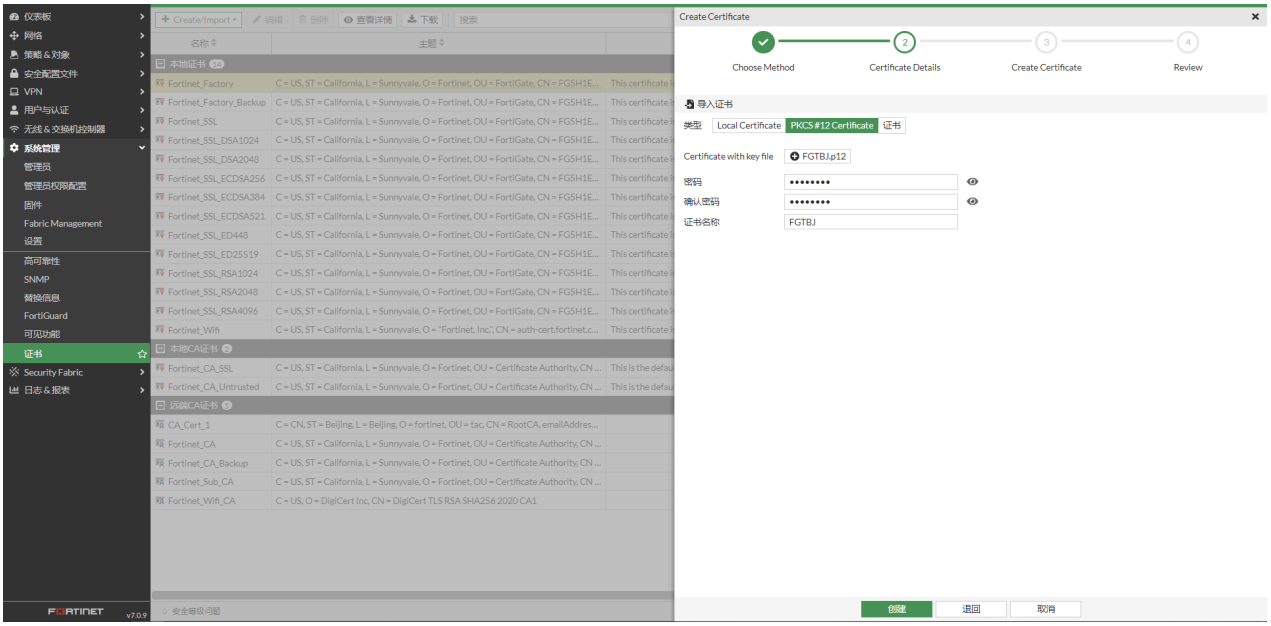
选择“系统管理”-->“证书”，点击“Create/Import”，选择“证书”。



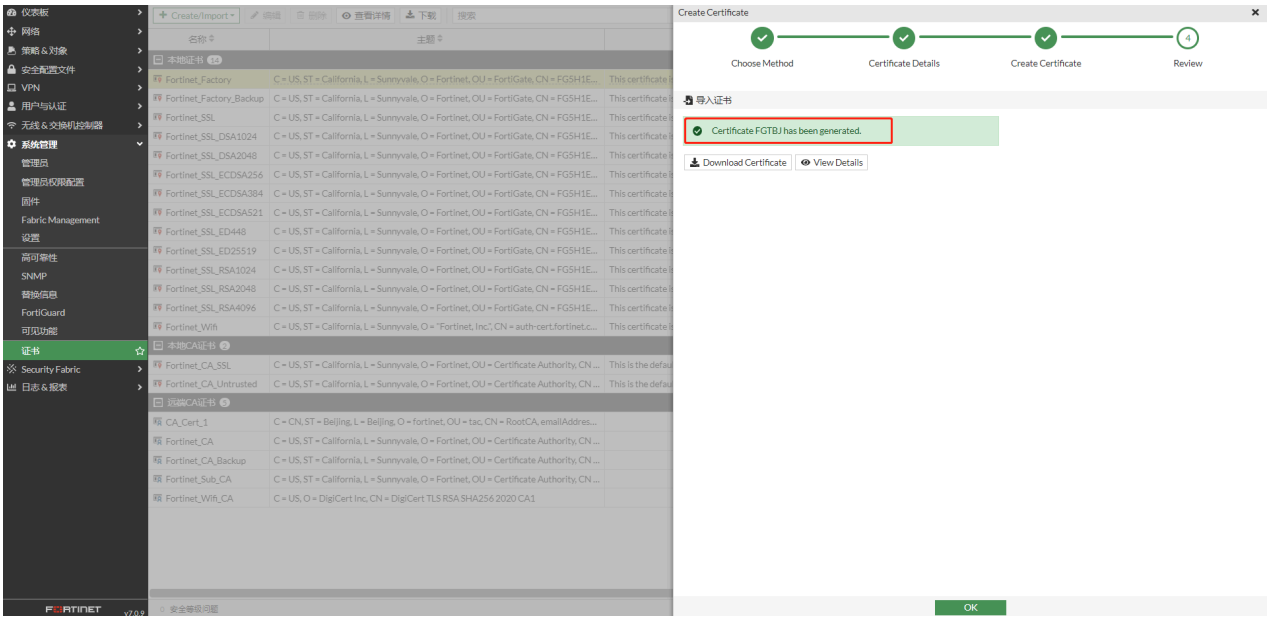
点击导入证书



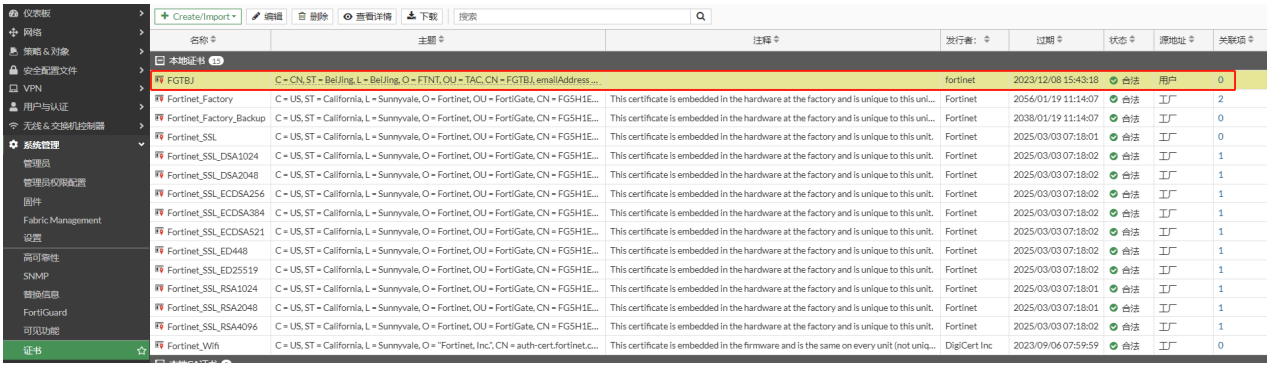
输入证书的密码，点击“创建”



提示证书已成功导入



查看导入的本地证书

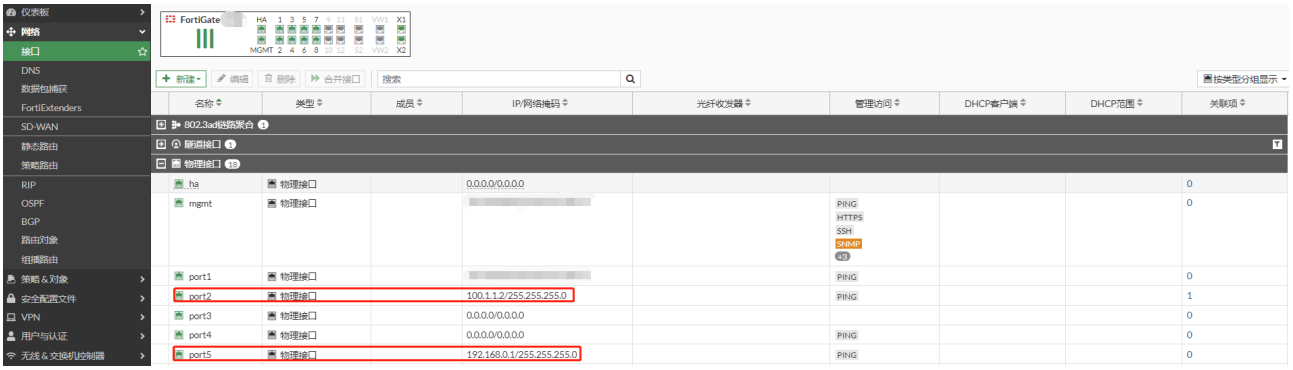


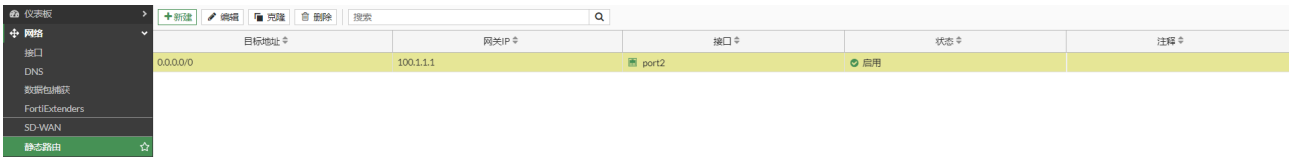
配置 IPSec

以 FGT - BJ 配置步骤为例，FGT - SH 同理。

1.基本配置

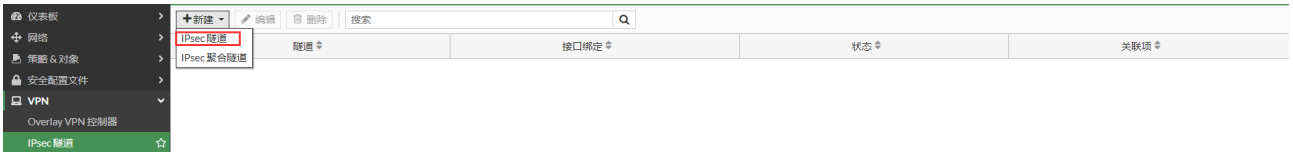
配置接口 IP 和路由



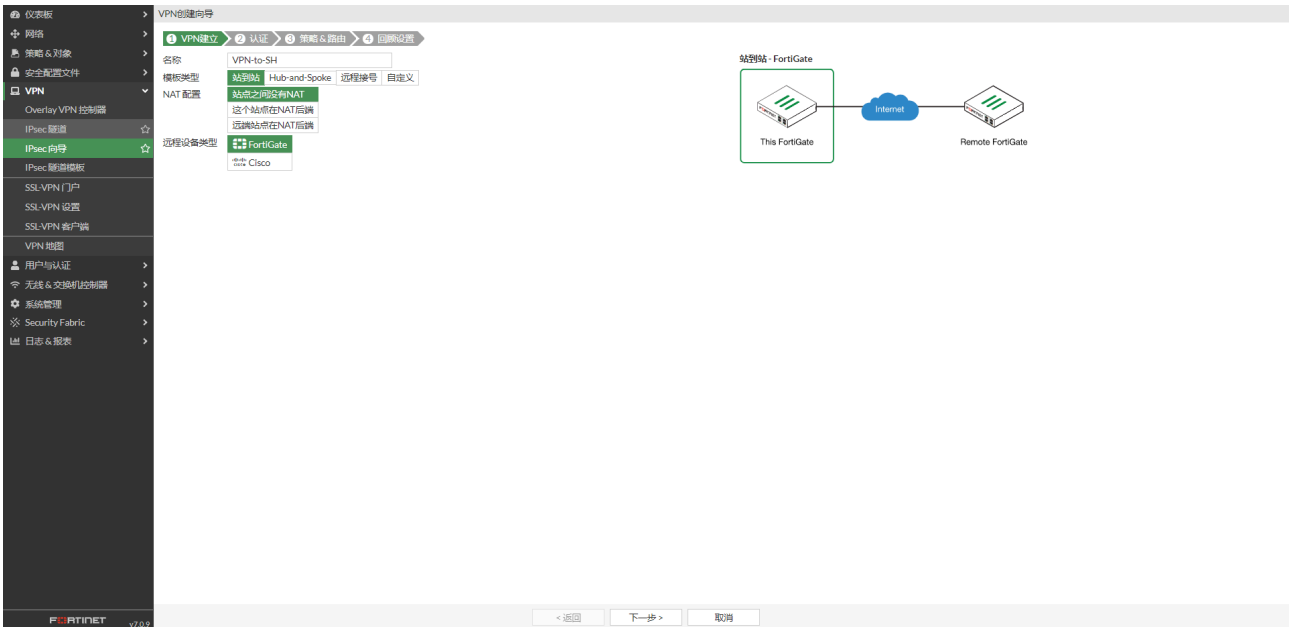


2.配置 IPSEC VPN

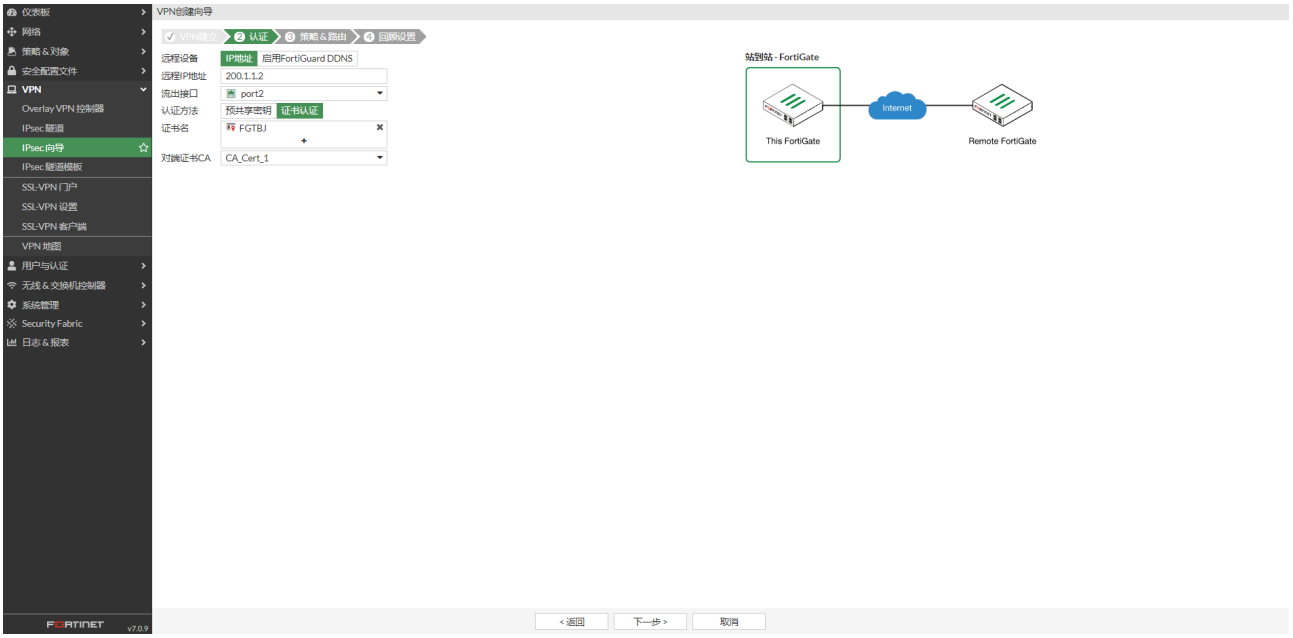
选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



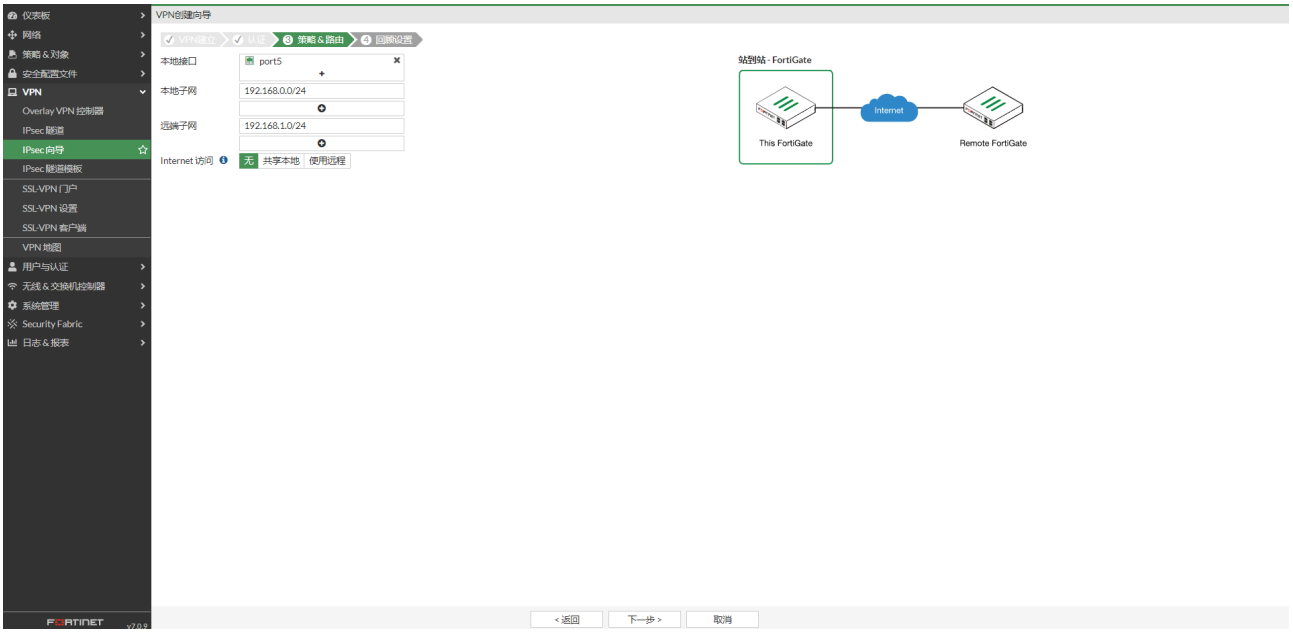
根据“VPN 创建向导”进行 VPN 模板配置，输入名称，拓扑中没有 NAT，这里选择“站点之间没有 NAT”，并点击下一步。



输入对端设备的 IP 地址（流出接口会根据路由自动选择），在证书认证中选择导入的本地证书“FGTBJ”，对端证书 CA 选择导入的 CA 证书“CA_Cert_1”。



选择“本地接口”，即需要被 IPSEC 保护的内网的接口，这里是 port5 接口，本地子网会自动写为该接口的 IP 所在的 IP 网段（如果被保护的是其他网段，可以根据实际网络需求修改），输入“远端子网”，即远端需要被保护的网络。然后点击“下一步”。



VPN 创建向导提示即将创建的内容，然后点击完成。



VPN 创建成功。



重要

如果最后完成模板部署的时候报错，那么需要注意，不要有旧的配置存在，比如相关 VPN 的策略、VPN 的路由、VPN 的 IP 地址对象，因为会冲突，所以需要把这些内容删除干净才可以顺利的部署向导模板，一般第一次部署不会遇到报错。

查看 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

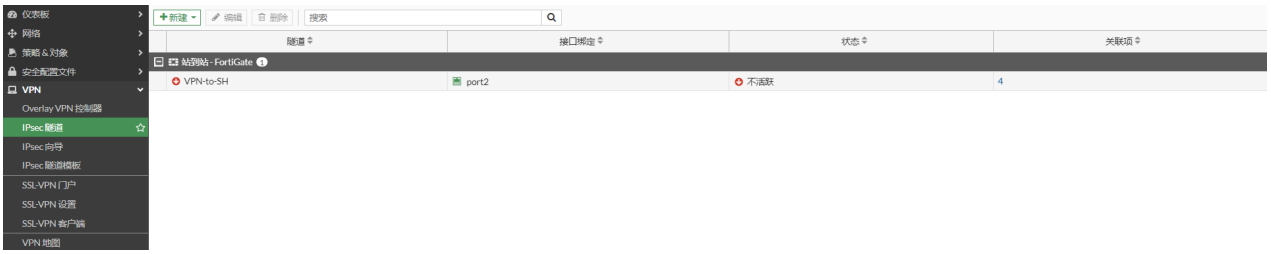
1. 创建地址对象和地址对象组



2. 创建 PKI 用户

```
config user peer
    edit "VPN-to-SH_peer"
        set ca "CA_Cert_1"
    next
end
```

3. 创建 IPSEC 隧道



对应的命令行如下：

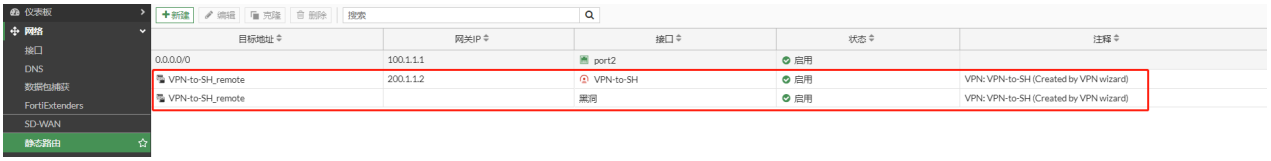
```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set interface "port2"
        set authmethod signature
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: VPN-to-SH (Created by VPN wizard)"
        set wizard-type static-fortigate
        set remote-gw 200.1.1.2
        set certificate "FGTBJ"
        set peer "VPN-to-SH_peer"
    next
end
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set phase1name "VPN-to-SH"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: VPN-to-SH (Created by VPN wizard)"
        set src-addr-type name
        set dst-addr-type name
        set src-name "VPN-to-SH_local"
        set dst-name "VPN-to-SH_remote"
    next
end
```

4. 创建路由

这里到对端的子网创建了两条路由：一条下一跳是 VPN-to-SH 接口的路由，一条黑洞路由。

> **重要**
>
> IPsec VPN 隧道的路由下一跳只是一个隧道 ID 标识，并不是真正的路由下一跳 IP，这和我们普通的路由下一跳不同，如这里的 200.1.1.2 是隧道的 ID。

黑洞路由的目的：VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量会因为查询到了默认路由而走向了 port2（Internet），从而产生了不期望的会话，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，业务流量会匹配不期望的会话而走向 port2，从而引起该业务异常。因此有了这条黑洞路由就不会产生这条不期望的会话。

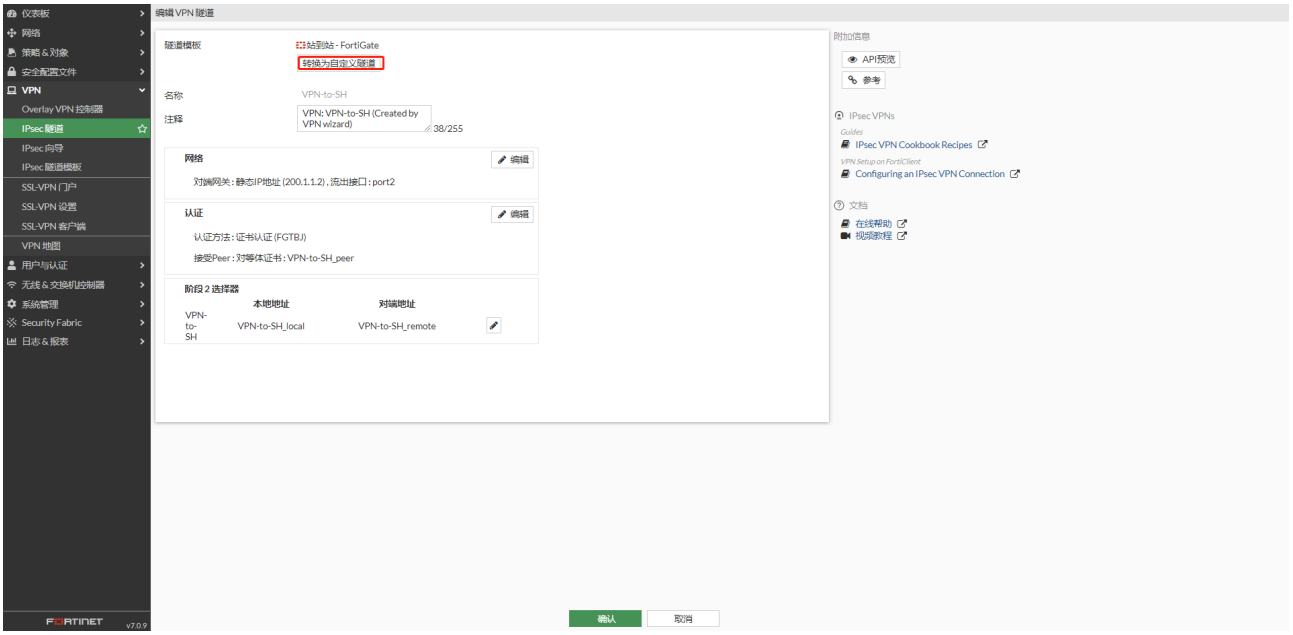


5. 创建策略



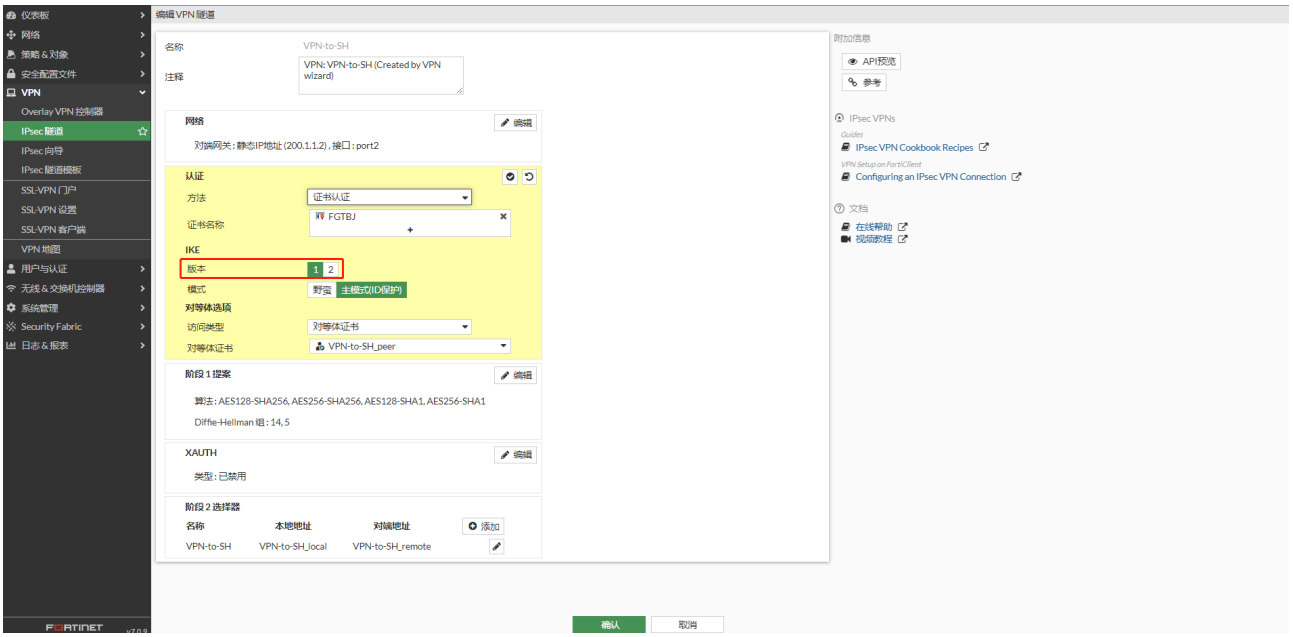
修改 IPSEC VPN 配置

编辑 VPN 隧道如下图

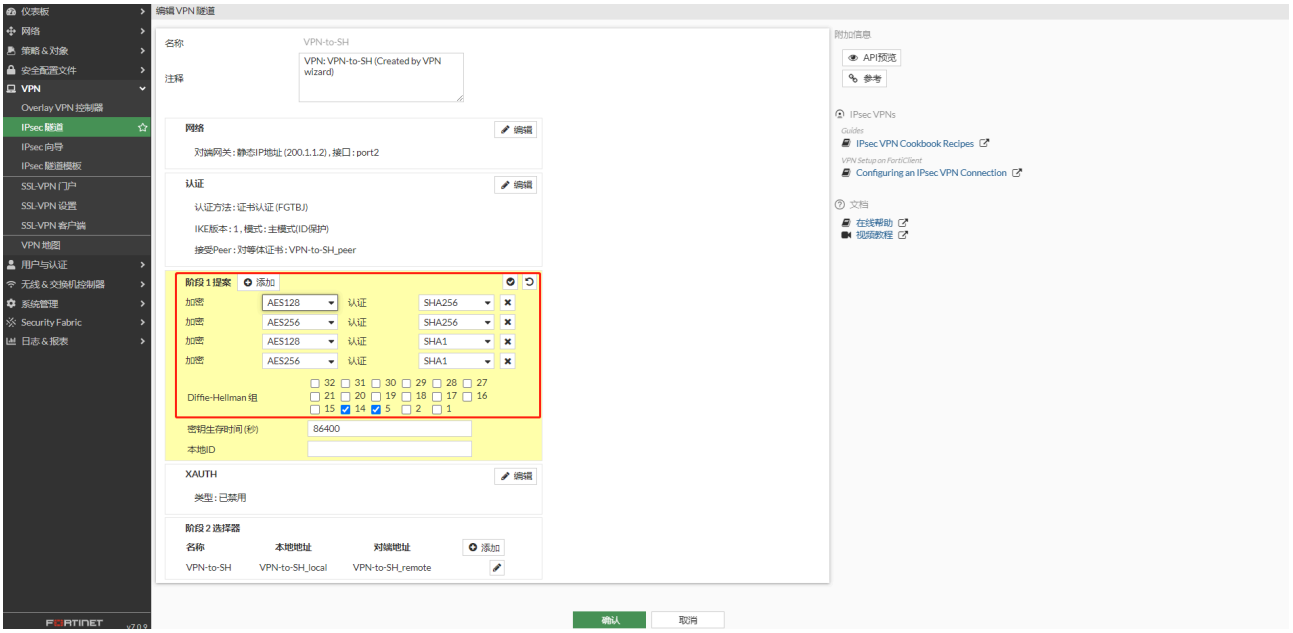


通过点击“转换为自定义隧道”可以在 GUI 界面设置的更多，是否需要修改可以根据实际网络需求决定。

如 IKE 版本（默认使用 IKEv1）

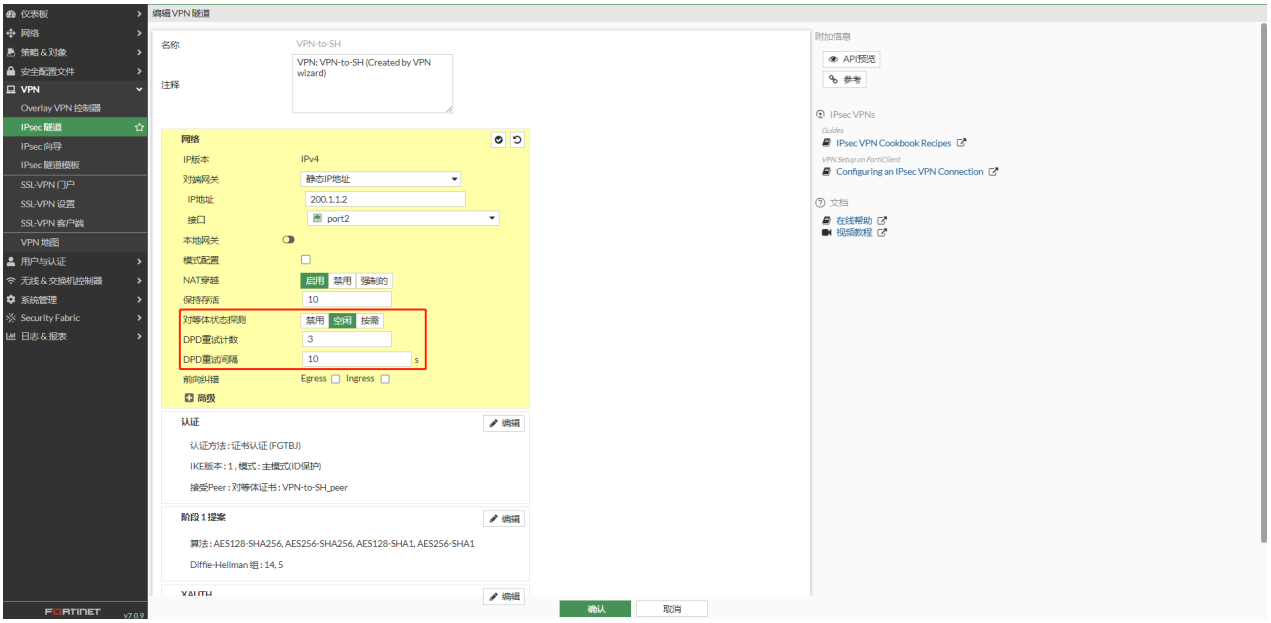


如修改阶段一加密和 HASH 算法



针对向导配置的优化建议

1. 在第一阶段中配置 DPD 模式为“空闲”（on-idle）（IPSec 隧道内没有流量时发 DPD 探测，只有当 IPSec 隧道内双向有流量时，才不发 DPD 探测），实现快速的检测并切换 VPN 隧道的目的。



对应的命令行

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

2. 开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

   IPSEC VPN 阶段一自动协商是默认开启的。

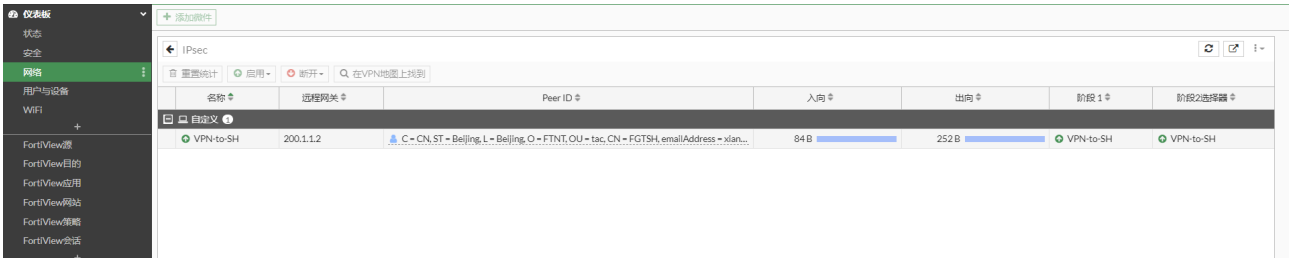
```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

查看 IPSEC 状态

GUI 查看 IPSEC 状态



CLI 查看 IPSEC 状态

```
# diagnose vpn ike gateway list

vd: root/0
name: VPN-to-SH
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::200.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 95s ago
peer-id: C = CN, ST = Beijing, L = Beijing, O = FTNT, OU = tac, CN = FGTSH, emailAddress = xxxxxx@fortinet.com
peer-id-auth: yes
IKE SA: created 2/2  established 2/2  time 0/10500/21000 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 261 36a40a780f6e5a26/5c8a58382858d1c5
  direction: responder
  status: established 81-81s ago = 0ms
  proposal: aes128-sha256
  key: 4d17bfee50a67591-97e10ab6c738e5c1
  lifetime/rekey: 86400/86048
  DPD sent/recv: 00000000/00000000
  peer-id: C = CN, ST = Beijing, L = Beijing, O = FTNT, OU = tac, CN = FGTSH, emailAddress = xxxxxx@fortinet.com

  id/spi: 259 6fdf66aa8c4d7016/d499a5bf9b7784e0
  direction: initiator
  status: established 95-74s ago = 21000ms
  proposal: aes128-sha256
  key: fb35b9a59b37285f-0f48bef12a92b99a
  lifetime/rekey: 86400/86025
  DPD sent/recv: 00000008/00000000
  peer-id: C = CN, ST = Beijing, L = Beijing, O = FTNT, OU = tac, CN = FGTSH, emailAddress = xxxxxx@fortinet.com

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-SH ver=1 serial=4 100.1.1.2:0→200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::200.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
```

```
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=4 olast=4 ad=/0
stat: rxp=1 txp=3 rxb=84 txb=252
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=8
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-SH proto=0 sa=1 ref=3 serial=1
  src: 0:192.168.0.0-192.168.0.255:0
  dst: 0:192.168.1.0-192.168.1.255:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42837/0B replaywin=2048
       seqno=4 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=0e81b327 esp=aes key=16 71d5090574a275127b9127ecc28d36a7
       ah=sha1 key=20 6389a6da9137a9d1b26d272d3919f2d218e6c0ce
  enc: spi=58e500e2 esp=aes key=16 447dd0e3a482ed893fea4aa549597dc4
       ah=sha1 key=20 1895ccc38badbf93caa77c9cbae424f4bd26e483
  dec:pkts/bytes=2/168, enc:pkts/bytes=6/708
  npu_flag=03 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=a dec_npuid=1 enc_npuid=1
```

查看路由

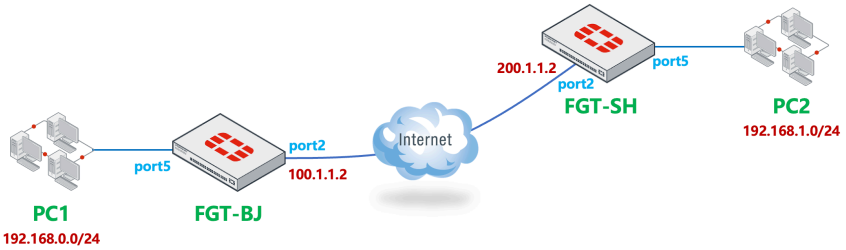
```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
S       192.168.1.0/24 [10/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]        ←——到192.168.1.0/24走IPSEC VPN
```

VPN 隧道上运行 OSPF 或 BGP

网络需求

通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。FGT - BJ 和 FGT - SH 通过 OSPF 或者 BGP 协议动态的学习到对方的路由。

网络拓扑

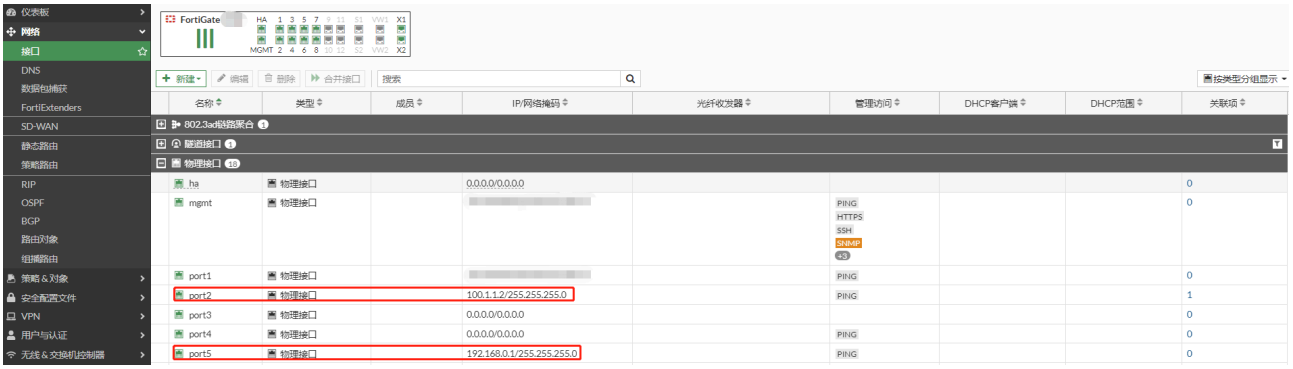


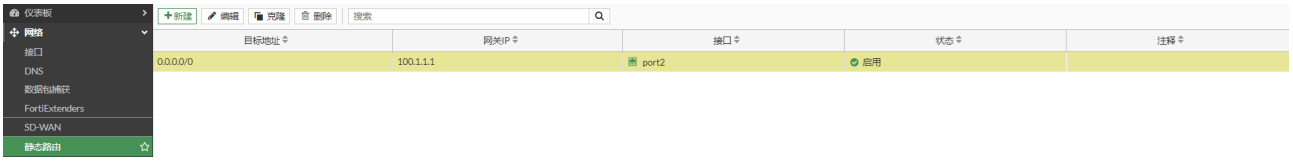
配置步骤

以 FGT - BJ 配置步骤为例，FGT - SH 同理。

1.基本配置

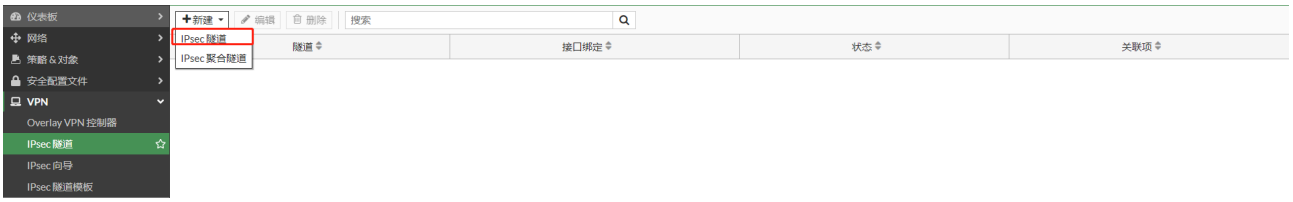
配置接口 IP 和路由





2.配置 IPSEC VPN

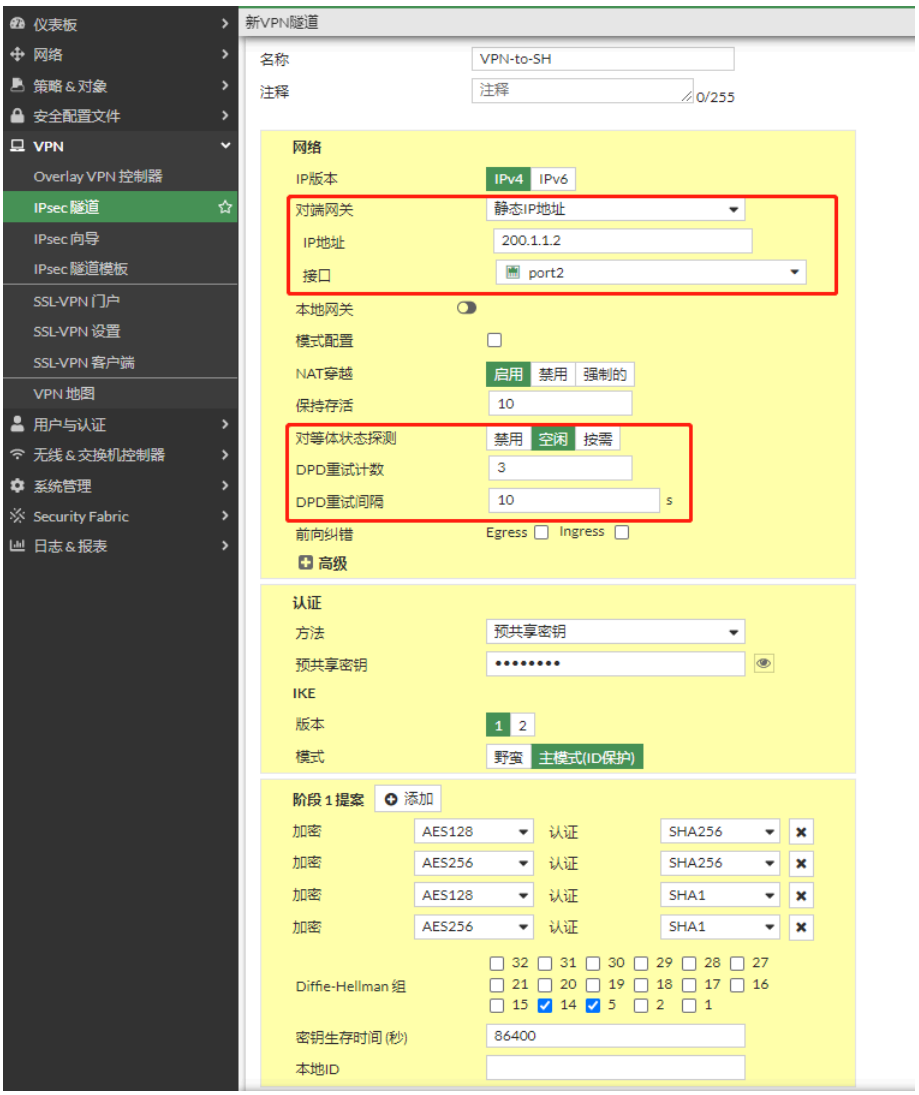
选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



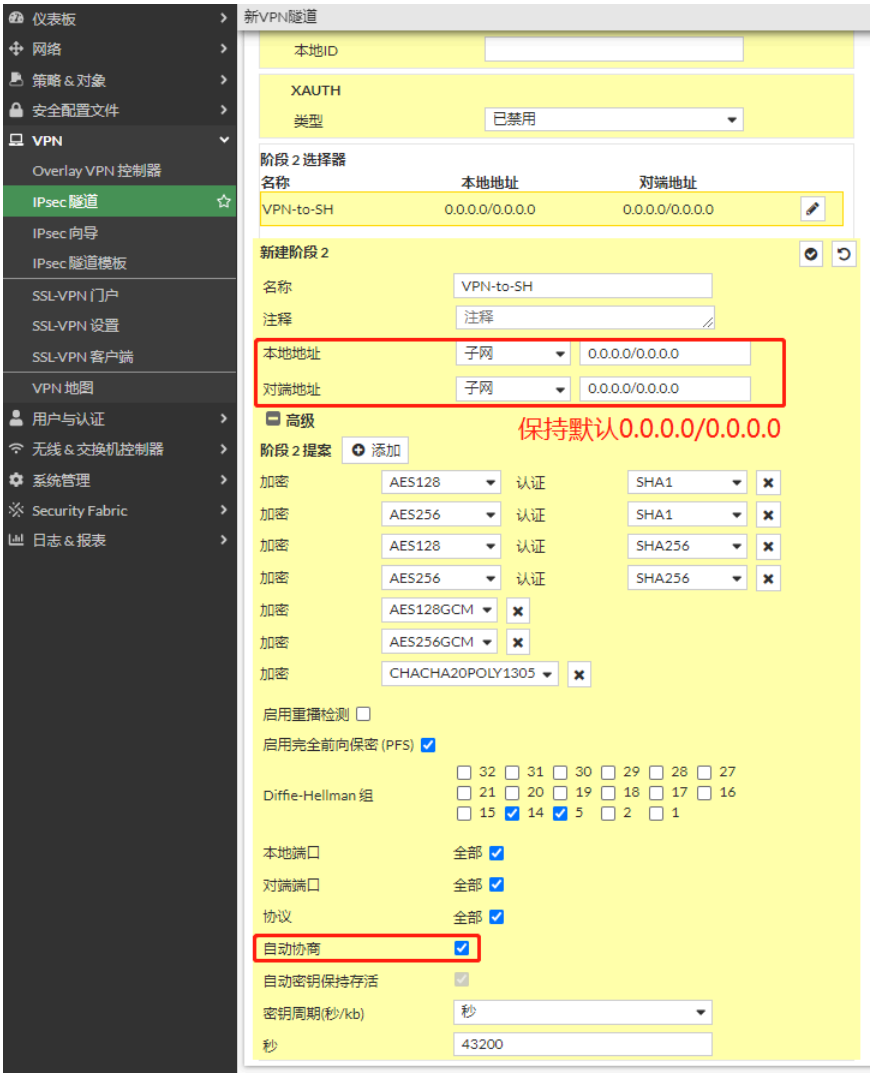
选择“自定义”



IPSEC VPN 阶段一配置

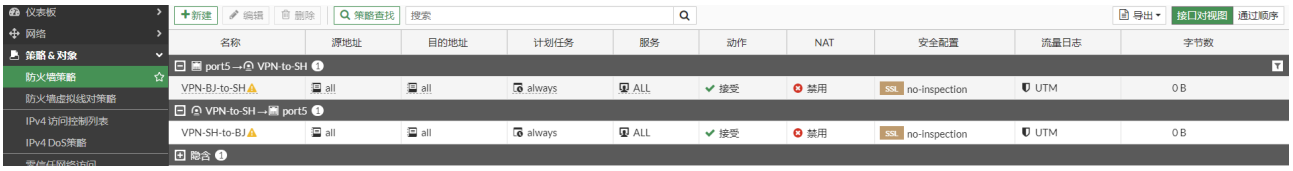


IPSEC VPN 阶段二配置





3.配置 IPSEC 策略



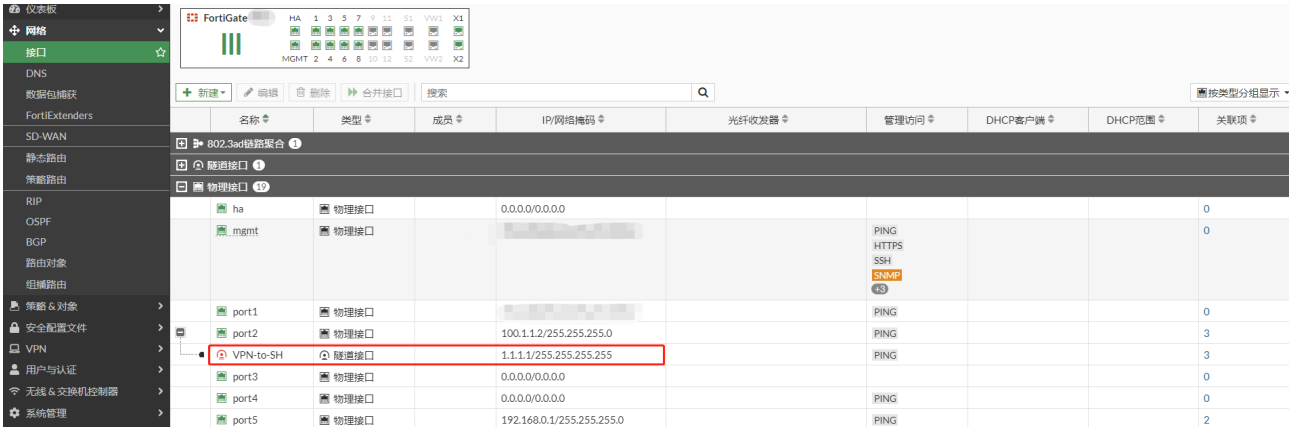
4.查看 IPSEC 状态

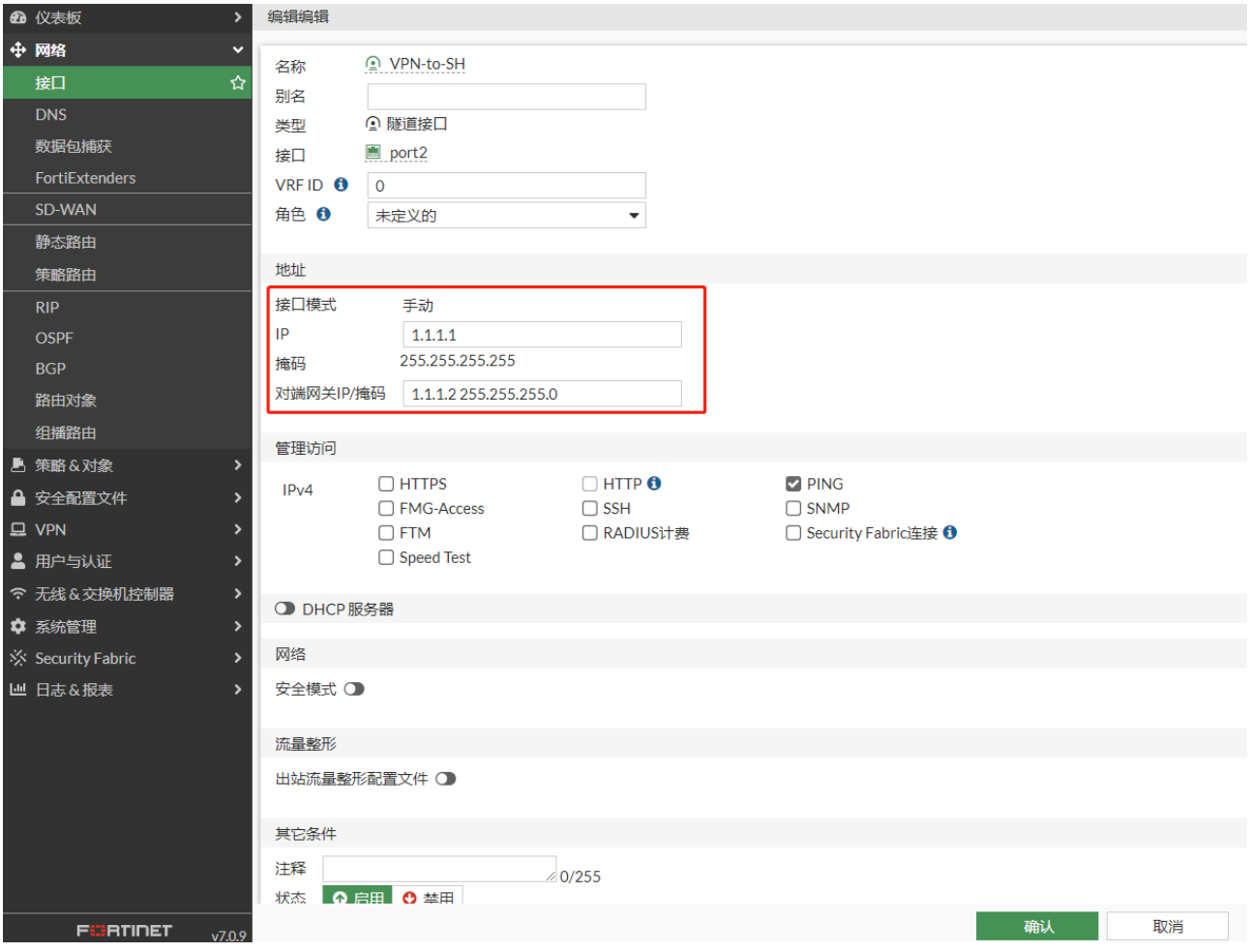
如果对端配置已配置完成，此时 IPSEC 状态是 UP 的。



5.配置 IPSEC 接口 IP

为隧道接口分配 IP 用于动态路由协议





对应的命令行

```
config system interface
    edit "VPN-to-SH"
        set vdom "root"
        set ip 1.1.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 1.1.1.2 255.255.255.0
        set interface "port2"
    next
end
```

查看路由表

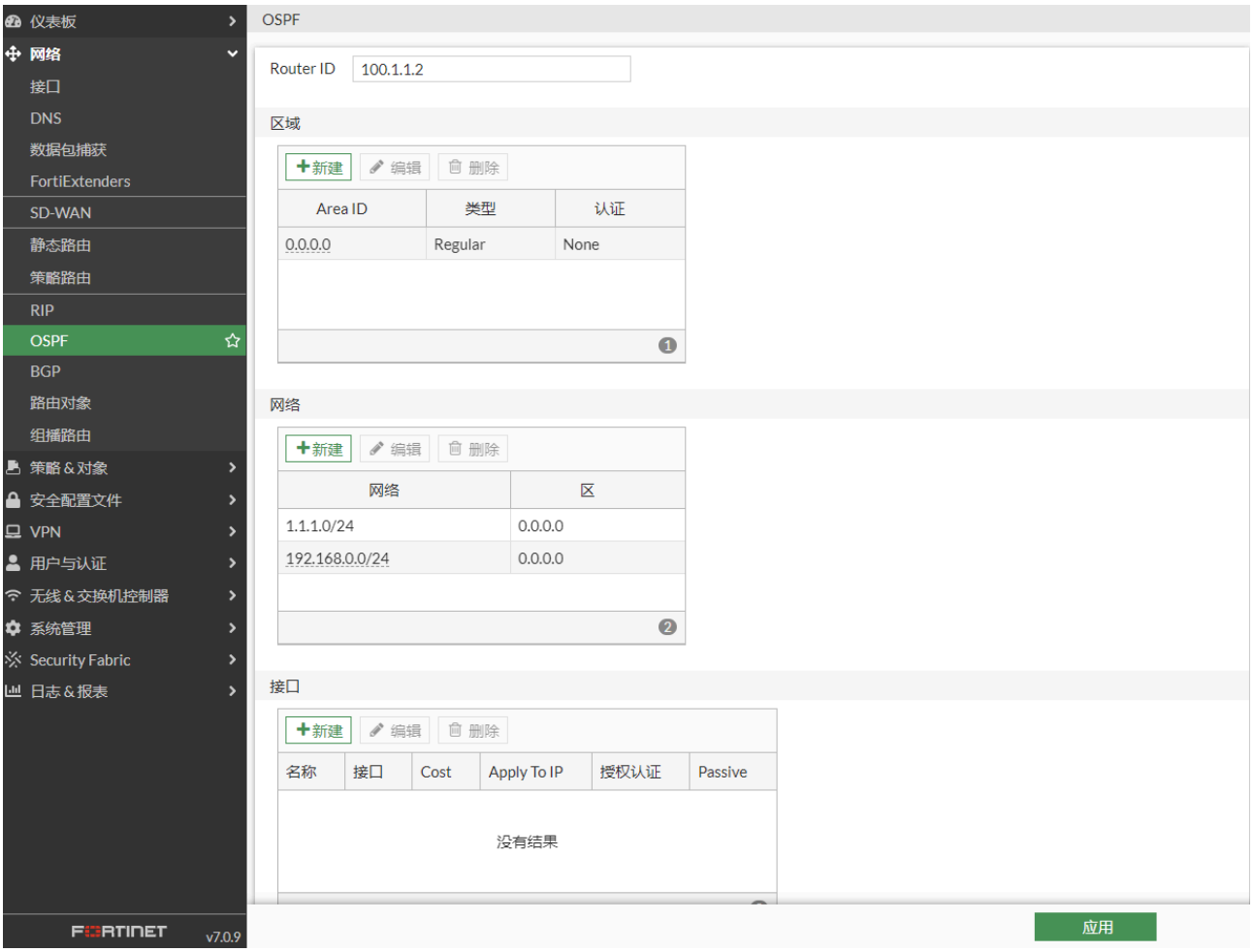
```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
S       1.1.1.0/24 [5/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]
C       1.1.1.1/32 is directly connected, VPN-to-SH
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
```

ping 测试，能够正常 ping 通对端的 IPSEC 接口地址

```
# execute ping-options source 1.1.1.1
# execute ping 1.1.1.2
PING 1.1.1.2 (1.1.1.2): 56 data bytes
64 bytes from 1.1.1.2: icmp_seq=0 ttl=255 time=0.5 ms
64 bytes from 1.1.1.2: icmp_seq=1 ttl=255 time=0.4 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=255 time=0.5 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=255 time=0.6 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=255 time=0.5 ms

--- 1.1.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.5/0.6 ms
```

6.使用 OSPF 或者 BGP 学习路由

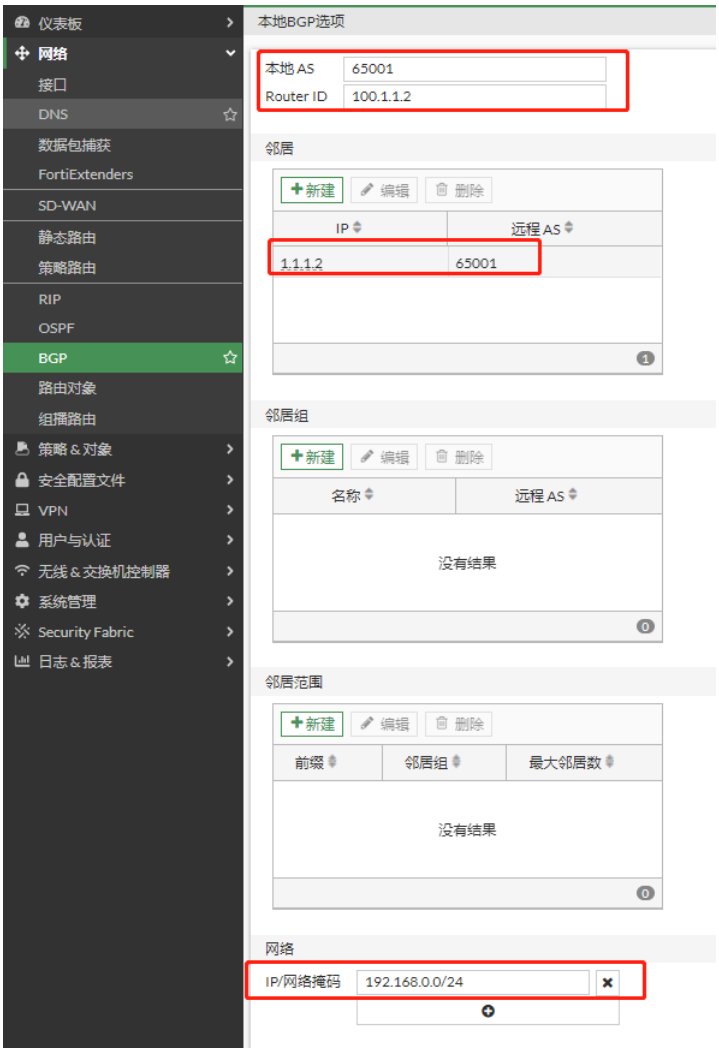
配置 OSPF



当对端也配置完成后，则可以建立 OSPF 邻居，并学习到对端的路由。

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
200.1.1.2         1   Full/ -         00:00:39    1.1.1.2         VPN-to-SH(tun-id:200.1.1.2)

# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
S       1.1.1.0/24 [5/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]
C       1.1.1.1/32 is directly connected, VPN-to-SH
O       1.1.1.2/32 [110/100] via VPN-to-SH tunnel 200.1.1.2, 00:01:44
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
O       192.168.1.0/24 [110/101] via VPN-to-SH tunnel 200.1.1.2, 00:01:4
```

配置 BGP



当对端也配置完成后，则可以建立 BGP 邻居，并学习到对端的路由。

```
# get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 1.1.1.2, remote AS 65001, local AS 65001, internal link
  BGP version 4, remote router ID 200.1.1.2
  BGP state = Established, up for 00:08:37
  Last read 00:00:03, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 12 messages, 0 notifications, 0 in queue
  Sent 12 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds
  Update source is VPN-to-SH
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  1 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
 Connections established 1; dropped 0
Local host: 1.1.1.1, Local port: 179
Foreign host: 1.1.1.2, Foreign port: 15455
Egress interface: 41
Nexthop: 1.1.1.1
Nexthop interface: VPN-to-SH
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
```

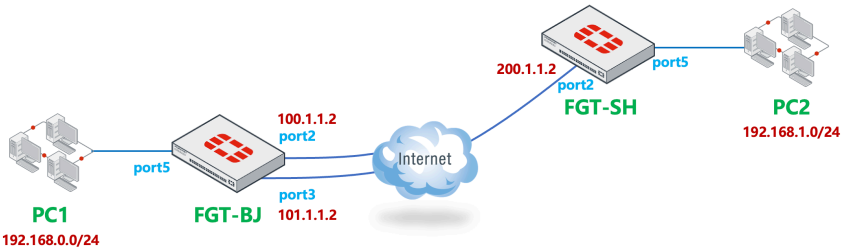
```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
S       1.1.1.0/24 [5/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]
C       1.1.1.1/32 is directly connected, VPN-to-SH
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
```

多路负载/主备 IPSec VPN

网络需求

通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信，并实现多路负载或者主备切换。

网络拓扑



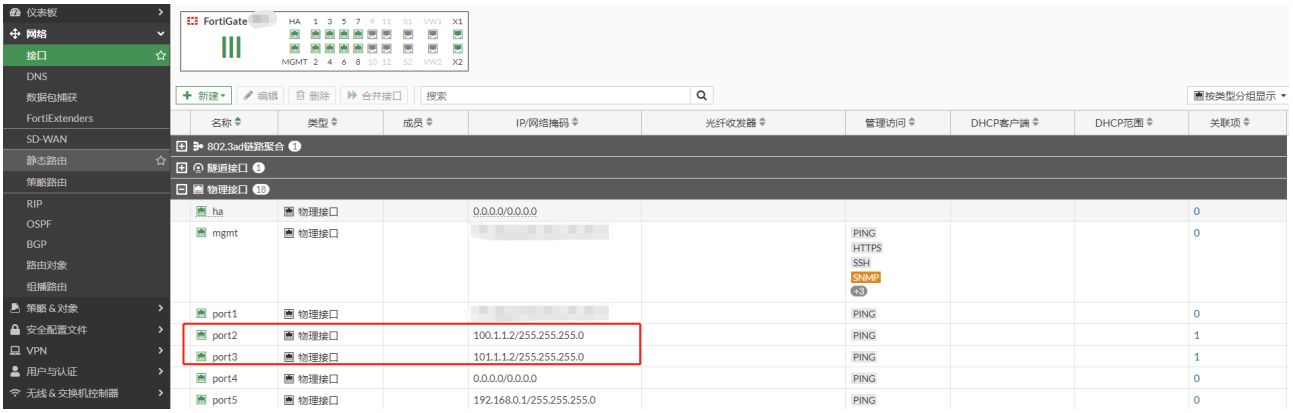
1. FGT - BJ port2 和 port3 接口分别与 FGT - SH port2 200.1.1.2 建立两条 IPSec VPN。
2. FGT - SH 配置同理，FGT - SH 使用 port2 接口分别与 FGT - BJ 100.1.1.2 和 101.1.1.2 建立两条 IPSec VPN。

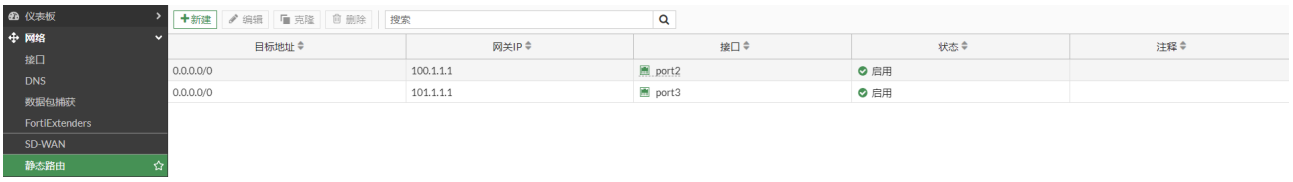
配置步骤

配置 IPSec

1.基本配置

配置接口 IP 和路由

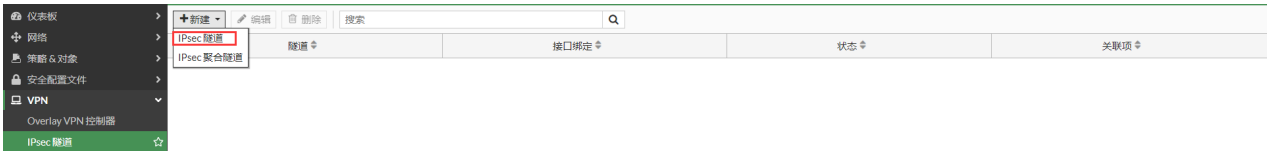




2. 配置 IPSEC VPN

FGT - BJ 使用 port2 接口与 FGT - SH port2 200.1.1.2 建立 IPSEC VPN

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



根据“VPN 创建向导”进行 VPN 模板配置，输入名称，拓扑中没有 NAT，这里选择“站点之间没有 NAT”，并点击下一步。



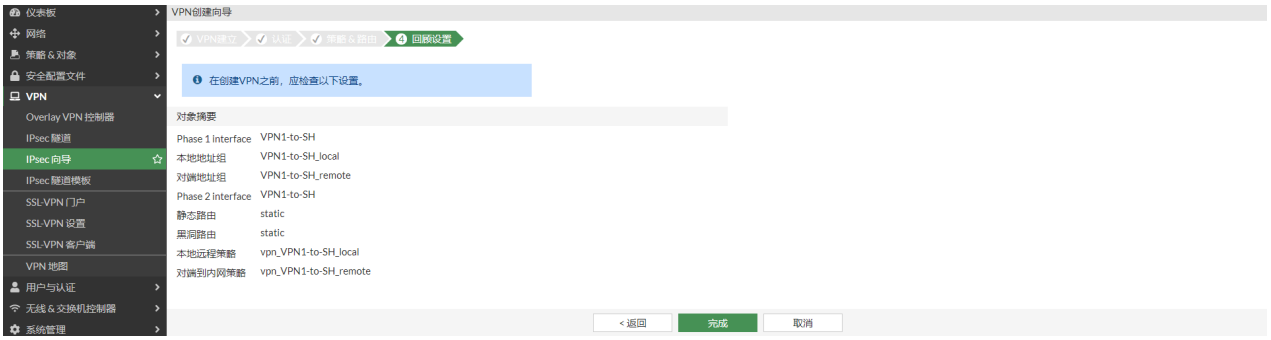
输入对端设备的 IP 地址（流出接口会根据路由自动选择），以及预共享秘钥。



选择“本地接口”，即需要被 IPSEC 保护的内网的接口，这里是 port5 接口，本地子网会自动写为该接口的 IP 所在的 IP 网段（如果被保护的是其他网段，可以根据实际网络需求修改），输入“远端子网”，即远端需要被保护的网络。然后点击“下一步”。



VPN 创建向导提示即将创建的内容，然后点击完成。

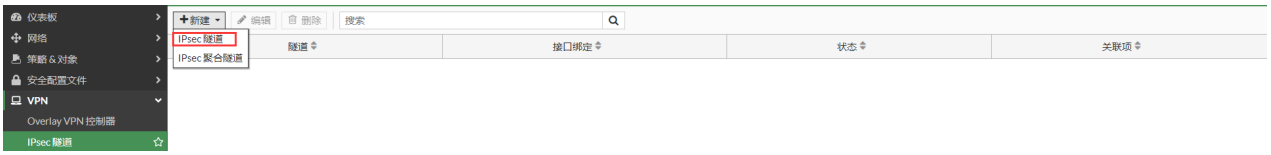


VPN 创建成功。



FGT - BJ 使用 port3 接口与 FGT - SH port2 200.1.1.2 建立 IPSEC VPN

选择“VPN” --> “IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



根据“VPN 创建向导”进行 VPN 模板配置，输入名称，拓扑中没有 NAT，这里选择“站点之间没有 NAT”，并点击下一步。



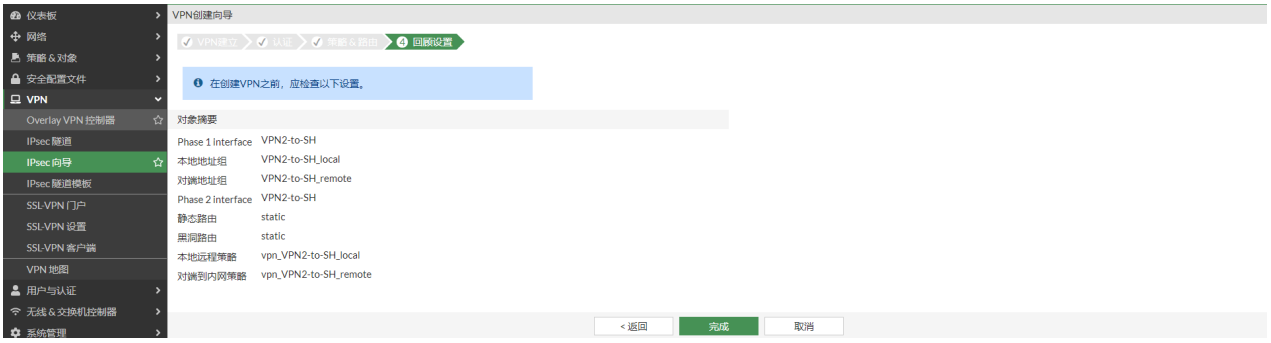
输入对端设备的 IP 地址（流出接口会根据路由自动选择），以及预共享秘钥。



选择“本地接口”，即需要被 IPSEC 保护的内网的接口，这里是 port5 接口，本地子网会自动写为该接口的 IP 所在的 IP 网段（如果被保护的是其他网段，可以根据实际网络需求修改），输入“远端子网”，即远端需要被保护的网络。然后点击“下一步”。



VPN 创建向导提示即将创建的内容，然后点击完成。



VPN 创建成功。

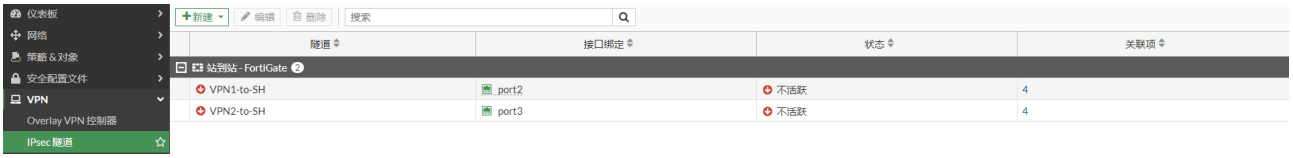


重要

如果最后完成模板部署的时候报错，那么需要注意，不要有旧的配置存在，比如相关 VPN 的策略、VPN 的路由、VPN 的 IP 地址对象，因为会冲突，所以需要把这些内容删除干净才可以顺利的部署向导模板，一般第一次部署不会遇到报错。

查看 IPSEC 配置

已创建两条 IPSEC VPN。

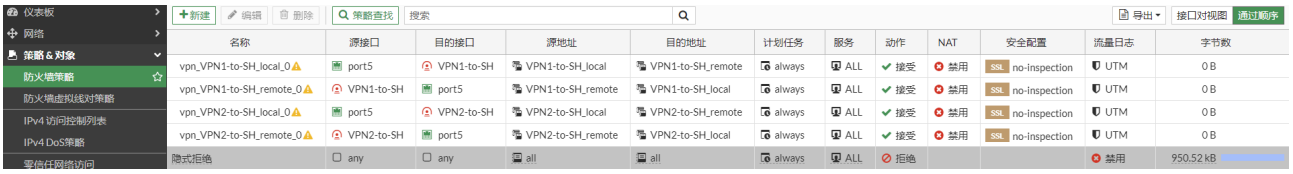


已创建 IPSEC 路由。

> **重要**
>
> IPsec VPN 隧道的路由下一跳只是一个隧道 ID 标识，并不是真正的路由下一跳 IP，这和我们普通的路由下一跳不同，如这里的 200.1.1.2 是隧道的 ID；隧道 ID 通常是隧道的远程网关，对于具有相同远程网关的隧道，其他隧道 id 将随机分配（10.0.0.x），如第 VPN2-to-SH 隧道 ID 10.0.0.1。

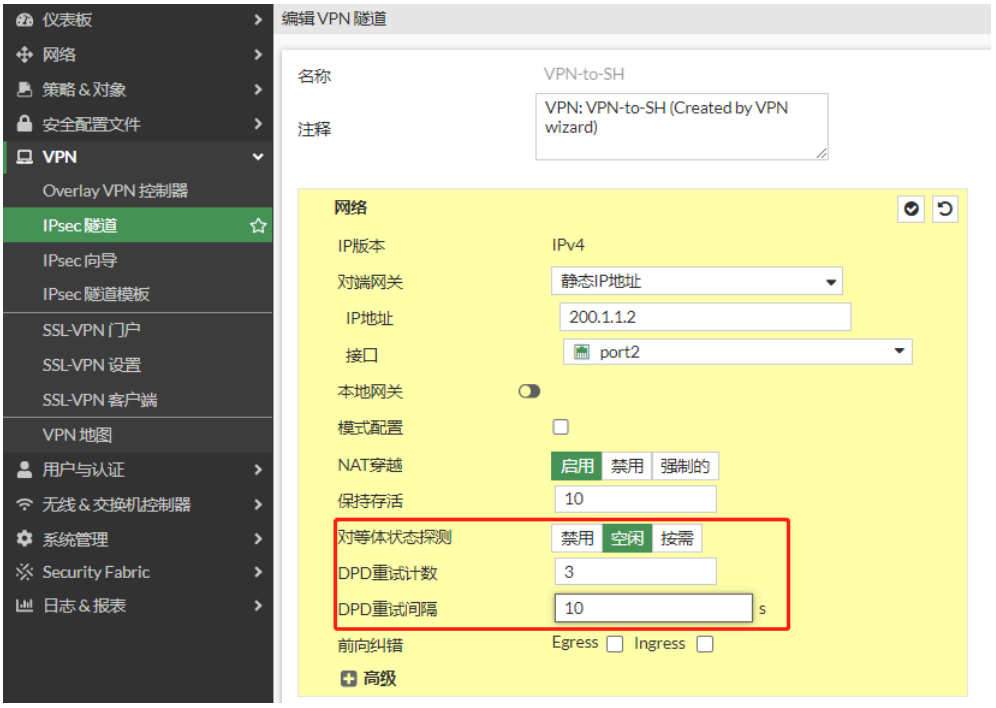


已创建防火墙策略。



针对向导配置的优化建议

1. 在第一阶段中配置 DPD 模式为“空闲”（on-idle）（IPSec 隧道内没有流量时发 DPD 探测，只有当 IPSec 隧道内双向有流量时，才不发 DPD 探测），实现快速的检测并切换 VPN 隧道的目的。



对应的命令行

```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
    edit "VPN2-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

2. 开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

   IPSEC VPN 阶段一自动协商是默认开启的。

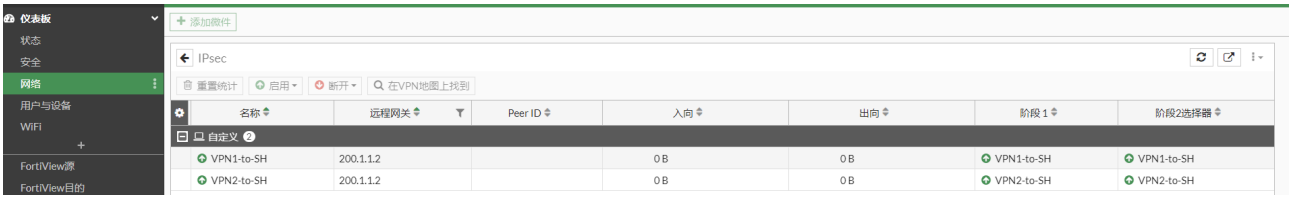
```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set auto-negotiate enable
    next
    edit "VPN2-to-SH"
        set auto-negotiate enable
    next
end
```

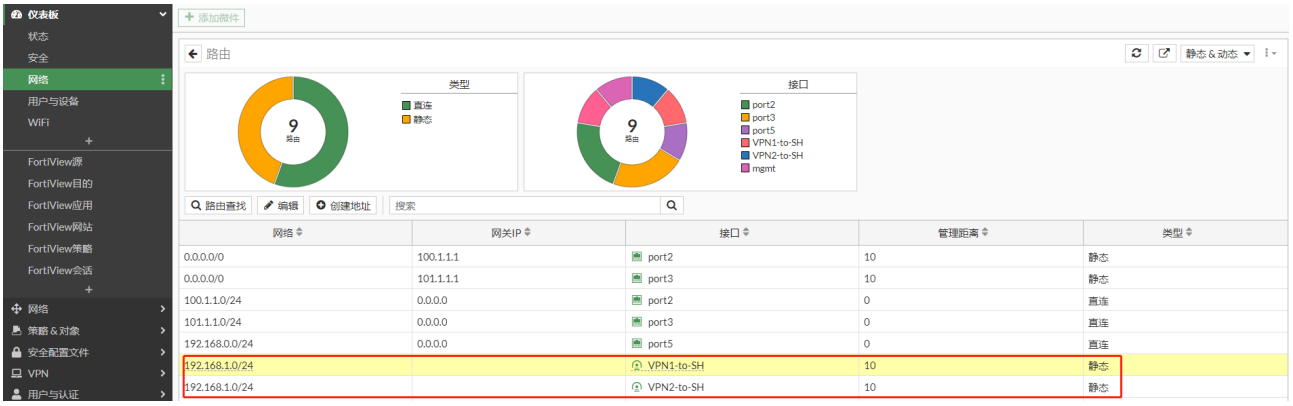
   IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "VPN1-to-SH"
        set auto-negotiate enable
    next
    edit "VPN2-to-SH"
        set auto-negotiate enable
    next
end
```

多路负载 IPSEC VPN

按照上述方式配置完成后，两条 IPSEC VPN 隧道的会 UP 起来，并且已通过静态路由实现负载均衡。





主备 IPSEC VPN

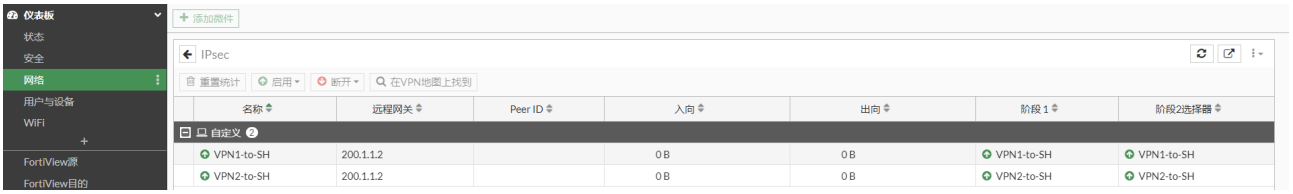
有两种方法实现主备 IPSEC VPN，这里 VPN1-to-SH 作为主，VPN2-to-SH 作为备。

1.使用路由的方法。通过静态路由的优先级控制 IPSEC VPN 主备。（动态路由也可以）

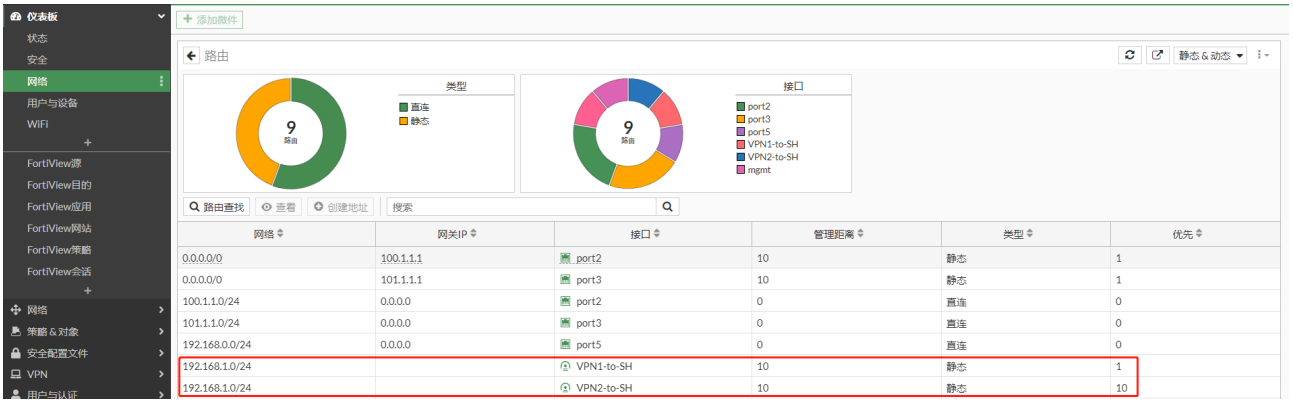
将 VPN2-to-SH 路由的优先级设置为 10，默认是 1，值越低优先级越高。



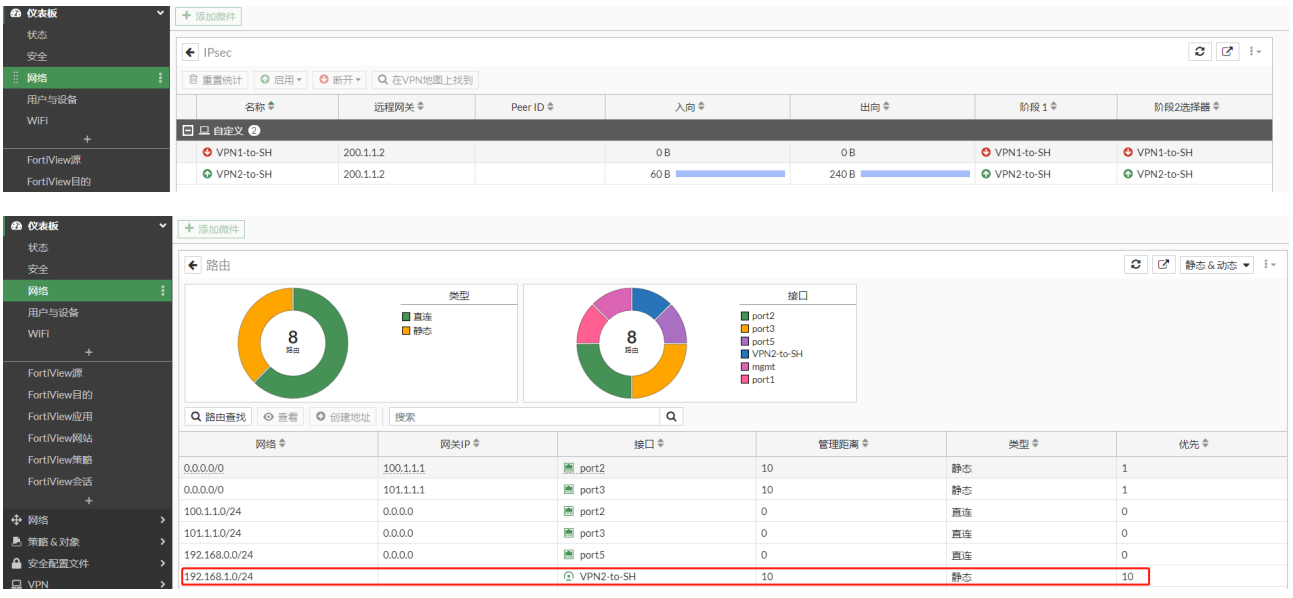
两条 IPSEC VPN 都能 UP。



从路由表可以看出，VPN1-to-SH 优先。



当 VPN1-to-SH DOWN 后，路由指向 VPN2-to-SH。



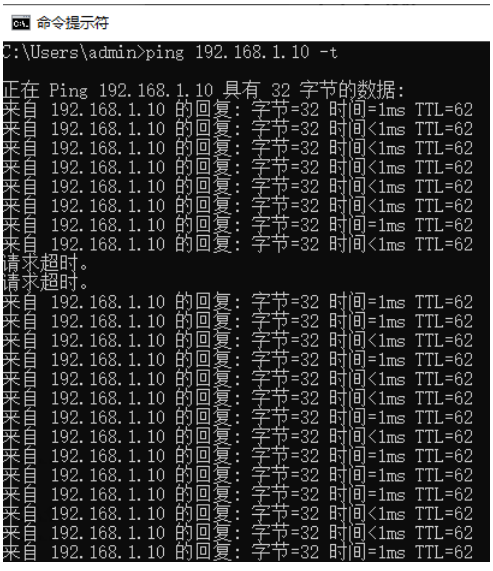
当 dpd 设置重试次数为 3，重试时间间隔为 10s 时，主备切换丢 10 个包。

```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```



当 dpd 设置重试次数为 3，重试时间间隔为 1s 时，主备切换丢 2 个包。

```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 1
    next
end
```



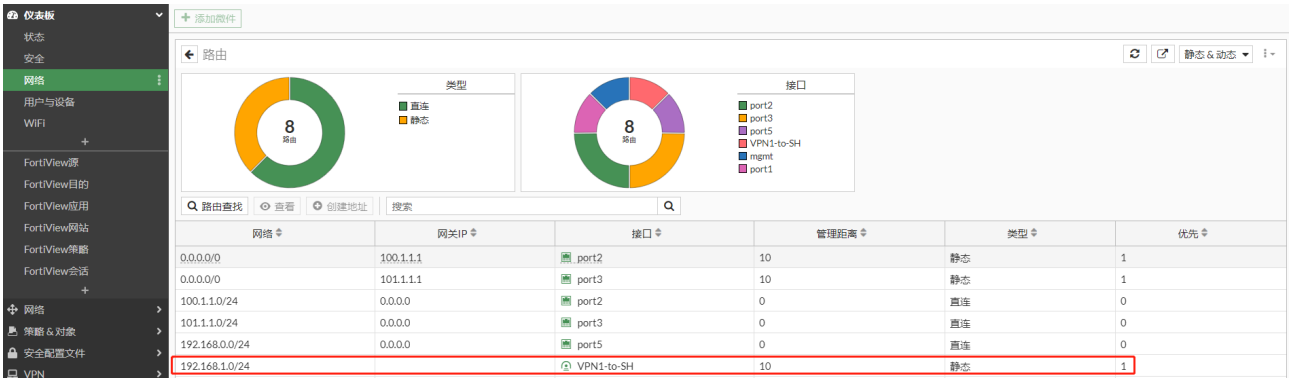
2.使用 IPSEC monitor 的方法

VPN1-to-SH 作为主 VPN UP 起来，VPN2-to-SH 作为备 VPN 则处于监听模式，一旦检测到主 VPN 隧道故障，则将备份的 VPN 协商起来，接替主 VPN 的工作。

```
config vpn ipsec phase1-interface
    edit "VPN2-to-SH"
        set monitor "VPN1-to-SH"
    next
end
```

配置完成后，VPN1-to-SH 是 UP 的，VPN2-to-SH 是 down 的。





当 port2 的链路故障，VPN1-to-SH DOWN 后，VPN2-to-SH 会协商 UP 起来。



当 dpd 设置重试次数为 3，重试时间间隔为 10s 时，主备切换丢 10 个包。

```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

命令提示符

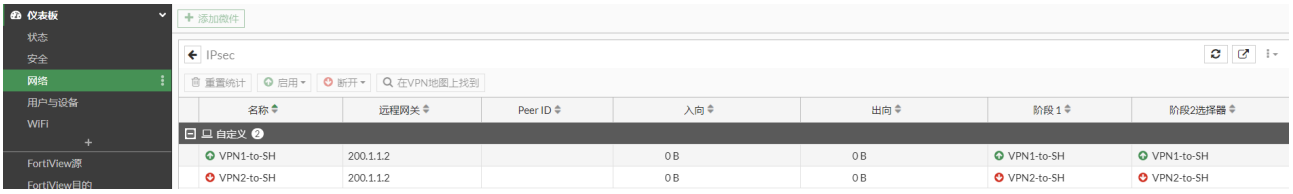
```
C:\Users\admin>ping 192.168.1.10 -t

正在 Ping 192.168.1.10 具有 32 字节的数据:
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
```

当 dpd 设置重试次数为 3，重试时间间隔为 1s 时，主备切换丢 2 个包。

```
config vpn ipsec phase1-interface
    edit "VPN1-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 1
    next
end
```

```
C:\Users\admin>ping 192.168.1.10 -t

正在 Ping 192.168.1.10 具有 32 字节的数据:
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
请求超时。
请求超时。
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=62
来自 192.168.1.10 的回复: 字节=32 时间=1ms TTL=62
```

当 port2 的链路恢复后，VPN1-to-SH 会抢回来，VPN1-to-SH 协商 UP 起来，VPN2-to-SH DOWN。

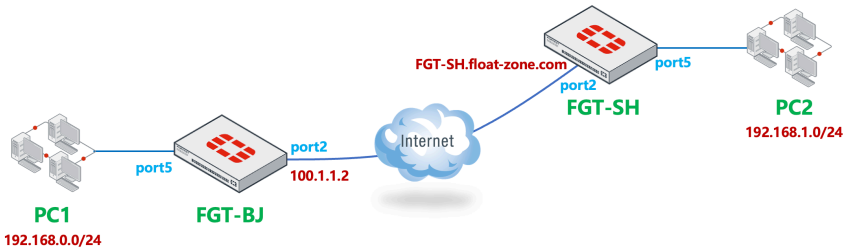


FortiGuard DDNS 方式建立 IPSec VPN

网络需求

FGT - BJ port2 为静态固定 IP 地址，FGT - SH port2 为 PPPOE/DHCP 动态拨号获取的 IP 地址，需要通过 IPsec VPN 将两个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。

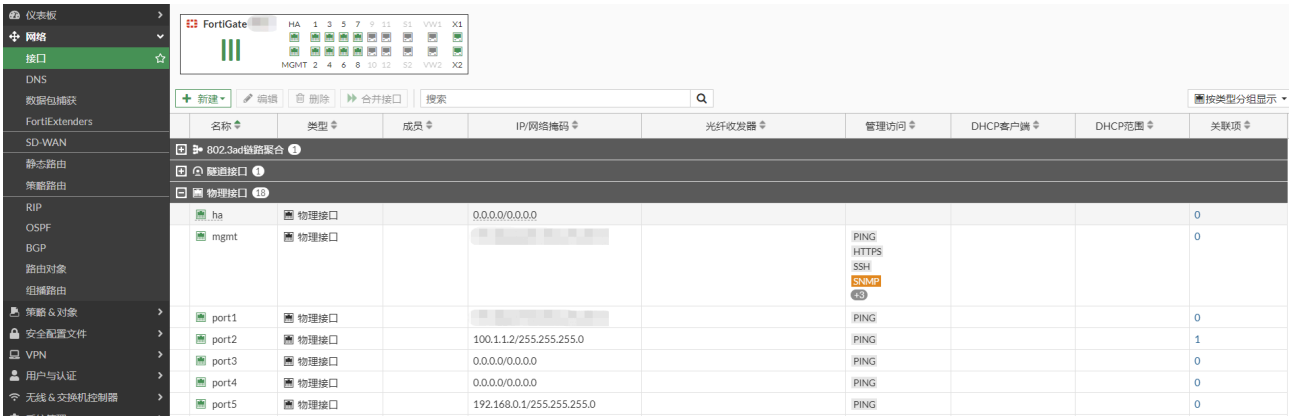
网络拓扑



配置步骤

FGT - BJ 配置

1.基本配置



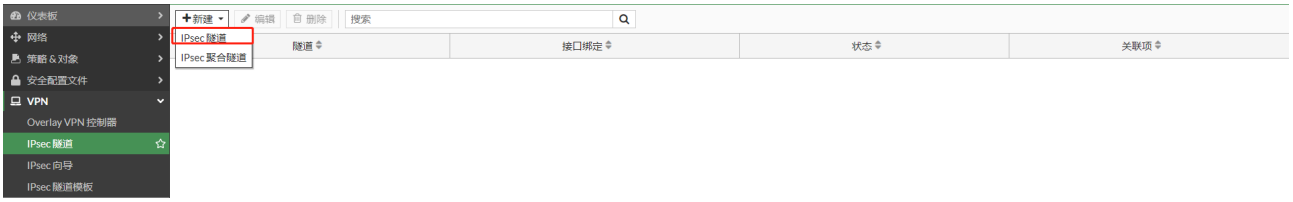


2.配置 IPSEC VPN

FGT - BJ 使用域名 FGT - SH.float-zone.com ↗ 与对端发起连接，在配置 IPSEC VPN 前，先从 FGT - BJ 上测试是否能解析该域名。

```
# execute ping FGT-SH.float-zone.com
PING FGT-SH.float-zone.com (200.1.1.2): 56 data bytes
64 bytes from 200.1.1.2: icmp_seq=0 ttl=254 time=0.4 ms
64 bytes from 200.1.1.2: icmp_seq=1 ttl=254 time=0.3 ms
64 bytes from 200.1.1.2: icmp_seq=2 ttl=254 time=0.6 ms
64 bytes from 200.1.1.2: icmp_seq=3 ttl=254 time=0.4 ms
64 bytes from 200.1.1.2: icmp_seq=4 ttl=254 time=0.5 ms

--- FGT-SH.float-zone.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.6 ms
```

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



选择“自定义”



IPSEC VPN 阶段一配置

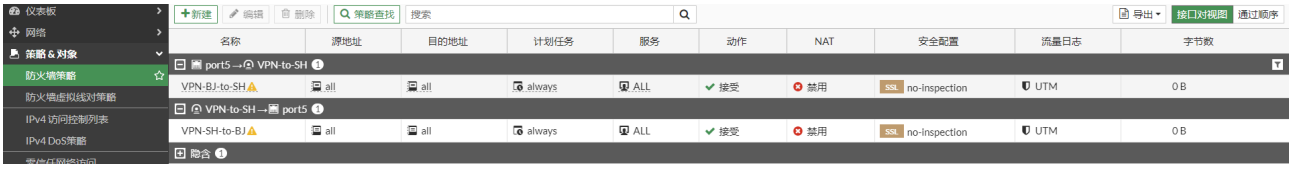


IPSEC VPN 阶段二配置

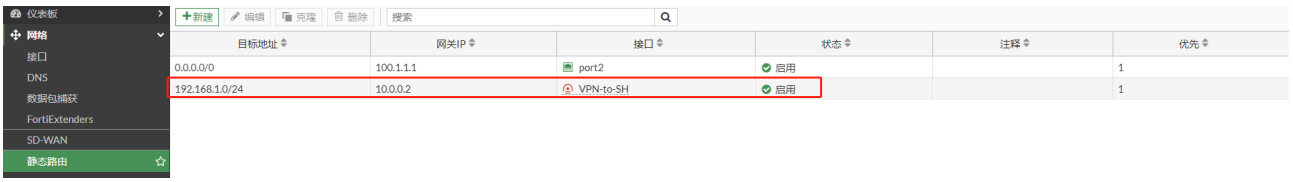




3.配置 IPSEC 策略

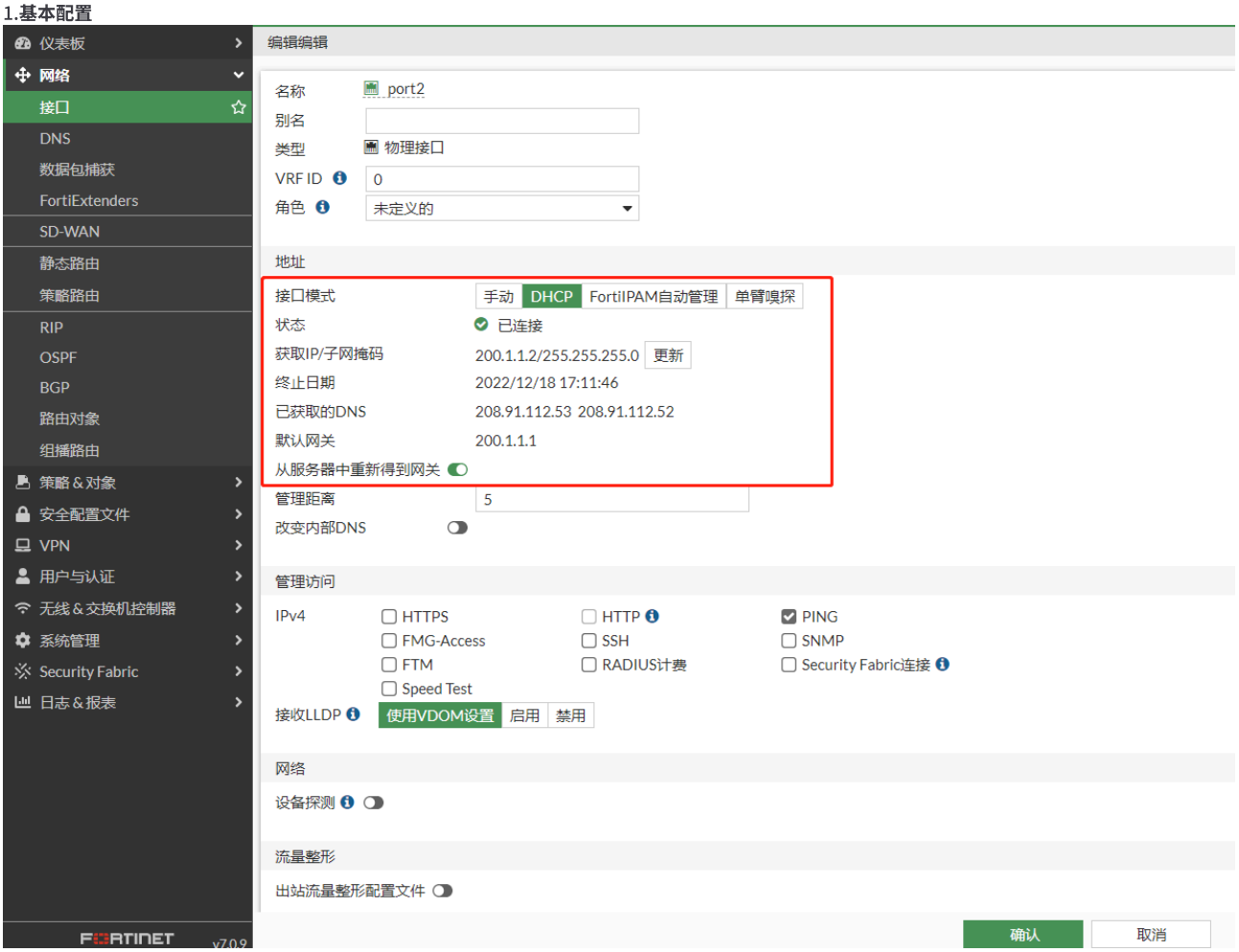


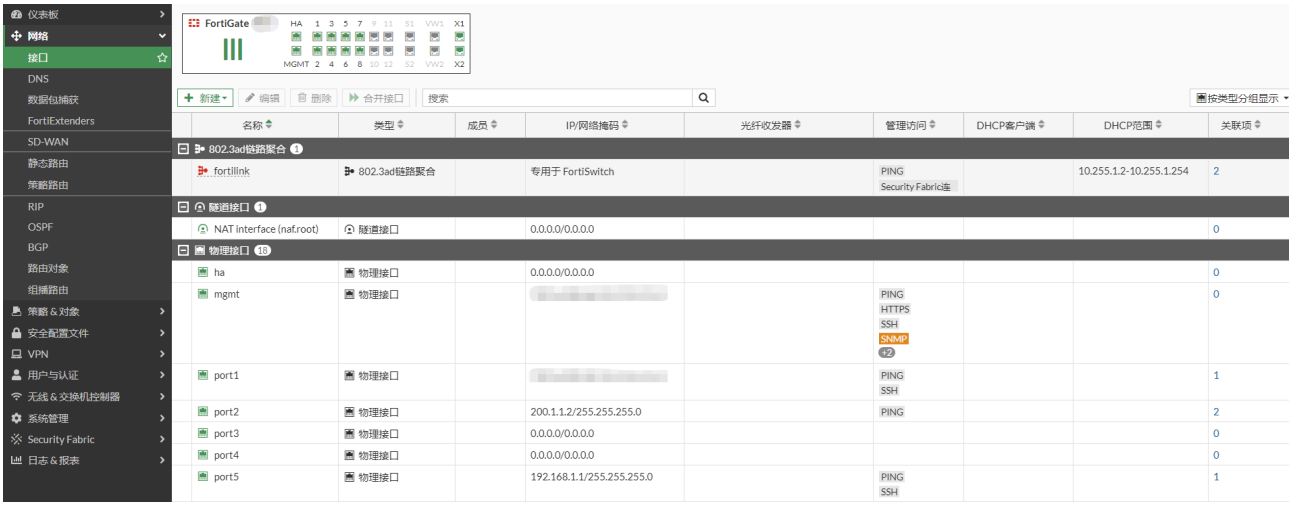
4.配置路由



FGT - SH 配置

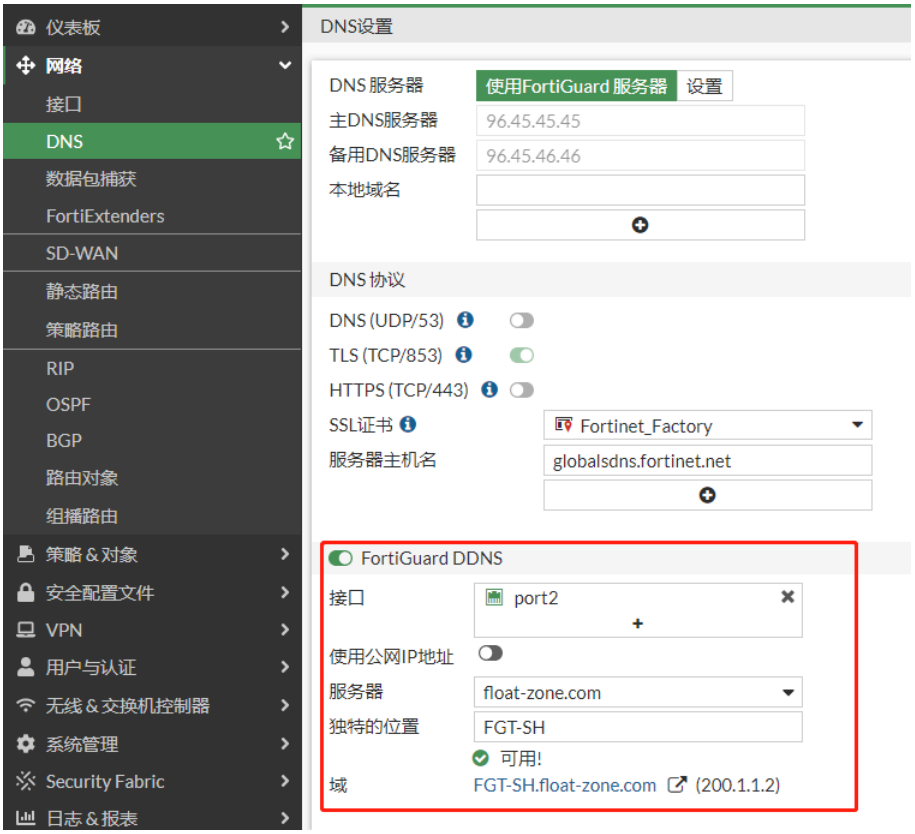
1.基本配置





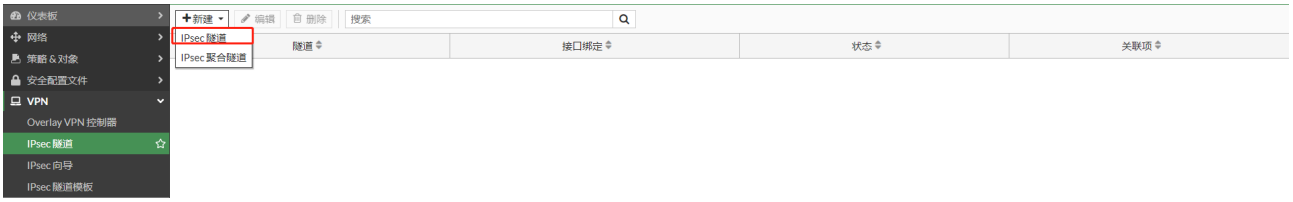
2.配置 FortiGuard DDNS

FGT - SH port2 接口使用 DHCP 获取地址，地址是会变化的，但建立 IPSEC VPN 时，对端需要一个固定的地址，因此使用域名替代 port2 接口的 IP。



3.配置 IPSEC VPN

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



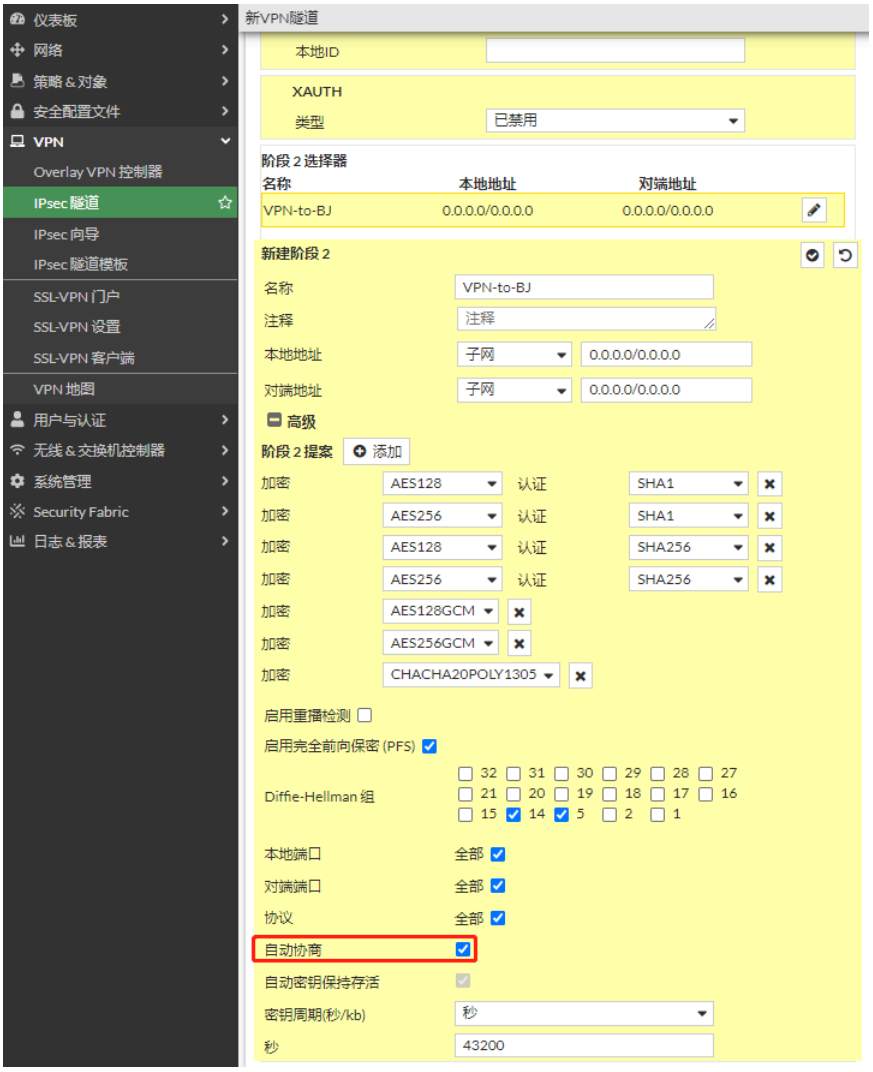
这里依然使用自定义



IPSEC VPN 阶段一配置



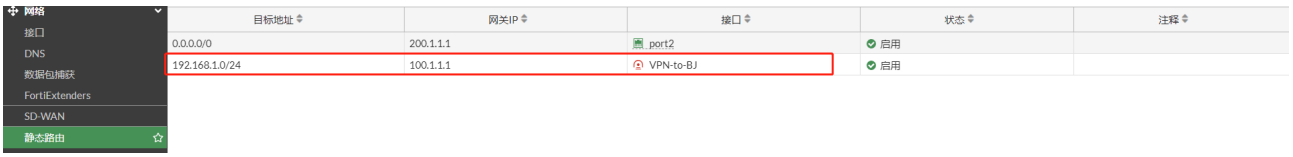
IPSEC VPN 阶段二配置



4.配置防火墙策略



5.配置路由



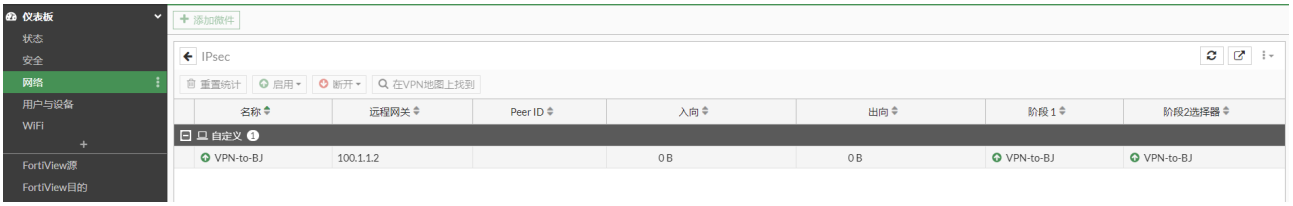
查看 IPSEC VPN 状态

1.FGT - BJ IPSEC VPN 状态



```
# diagnose vpn ike gateway list

vd: root/0
name: VPN-to-SH
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 10.0.0.2/::10.0.0.2
remote_location: 0.0.0.0
network-id: 0
created: 195s ago
IKE SA: created 2/2  established 2/2  time 0/10500/21000 ms
IPsec SA: created 1/2  established 1/1  time 0/0/0 ms

  id/spi: 10 e3498ed0f017b22b/a19990e821f3733e
  direction: responder
  status: established 177-177s ago = 0ms
  proposal: aes128-sha256
  key: c9c4dd7d0bdea34b-e9425ad0fe0a17bc
  lifetime/rekey: 86400/85952
  DPD sent/recv: 00000000/00000000

  id/spi: 9 d46c45f0f50d9ce8/fcaf35fab405c65a
  direction: initiator
  status: established 195-174s ago = 21000ms
  proposal: aes128-sha256
  key: 27f09a3f144bf8da-85d9c19874786059
  lifetime/rekey: 86400/85925
  DPD sent/recv: 00000000/00000012

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-SH ver=1 serial=b 100.1.1.2:0→200.1.1.2:0 tun_id=10.0.0.2 tun_id6=::10.0.0.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=3 olast=3 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=10000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-SH proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=18225 type=00 soft=0 mtu=1438 expire=42750/0B replaywin=0
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42933/43200
  dec: spi=03ff1e97 esp=aes key=16 e44c72247f93456bb0191fb24eee8a8f
       ah=sha1 key=20 884f93d1f1212a08eb7e99fe6153408554d1e7f2
  enc: spi=a7236c3f esp=aes key=16 712b50eed56e1d231956ed11c595220b
       ah=sha1 key=20 0bede372f1a608d05df1c793d25c8939afb995a4
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=16 dec_npuid=0 enc_npuid=0
run_tally=0
```

2.FGT - SH IPSEC VPN 状态




```
# diagnose vpn ike gateway  list

vd: root/0
name: VPN-to-BJ
version: 1
```

```
interface: port2 10
addr: 200.1.1.2:500 -> 100.1.1.2:500
tun_id: 100.1.1.1/::100.1.1.1
remote_location: 0.0.0.0
network-id: 0
created: 261s ago
IKE SA: created 2/2  established 2/2  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 59 d46c45f0f50d9ce8/fcaf35fab405c65a
  direction: responder
  status: established 258-258s ago = 0ms
  proposal: aes128-sha256
  key: 27f09a3f144bf8da-85d9c19874786059
  lifetime/rekey: 86400/85871
  DPD sent/recv: 0000001b/00000000

  id/spi: 58 e3498ed0f017b22b/a19990e821f3733e
  direction: initiator
  status: established 261-261s ago = 0ms
  proposal: aes128-sha256
  key: c9c4dd7d0bdea34b-e9425ad0fe0a17bc
  lifetime/rekey: 86400/85838
  DPD sent/recv: 00000000/00000000

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-BJ ver=1 serial=d 200.1.1.2:0->100.1.1.2:0 tun_id=100.1.1.1 tun_id6=::100.1.1.1 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=3 olast=3 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=27
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-BJ proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=18225 type=00 soft=0 mtu=1438 expire=42637/0B replaywin=0
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=a7236c3f esp=aes key=16 712b50eed56e1d231956ed11c595220b
       ah=sha1 key=20 0bede372f1a608d05df1c793d25c8939afb995a4
  enc: spi=03ff1e97 esp=aes key=16 e44c72247f93456bb0191fb24eee8a8f
       ah=sha1 key=20 884f93d1f1212a08eb7e99fe6153408554d1e7f2
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=200.1.1.2 npu_selid=f dec_npuid=0 enc_npuid=0
run_tally=0
```

GRE over IPSec

应用场景

传统的策略模式 IPSec VPN 无法运行组播、广播等数据，因此像 OSPF/RIP 等协议无法通过 IPSec VPN 直接运行，而需要借用 GRE 协议的辅助，GRE 协议可封装组播、广播等数据，而 IPSec 又可将 GRE 协议封装在里面并且安全的加密，这样实现了通过互联网 VPN 安全的跑组播、广播等数据。

> **重要**
>
> 飞塔的接口模式 IPSec VPN 可以直接运行 OSPF/RIP 协议，因此如果都是 FortiGate 设备则无需使用 GRE over IPSec，而直接使用接口模式的 IPSec VPN 即可实现 GRE over IPSec 一样的功能。与友商对接的时候也应该优先选择接口模式，部分友商有类似 FortiGate 接口模式的 IPSec VPN（例如思科叫 VTI，Virtual Tunnel Interface 模式的 IPSec VPN），而不需要再选择的 GRE over IPSec，可以被接口模式替代。

那么什么时候需要用到 GRE over IPSec 呢?

1. 替换友商设备运行了 GRE over IPSec 的设备

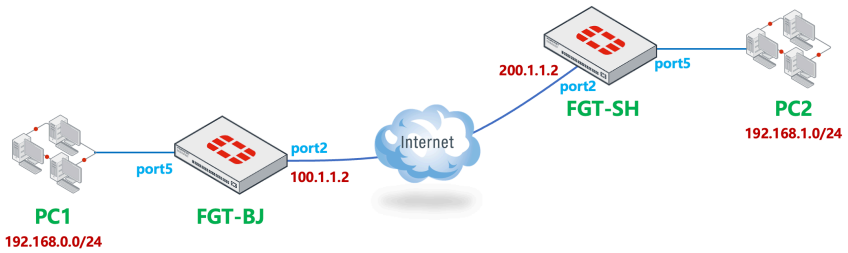
2. 对端设备为友商设备，且只能用 GRE over IPSec 进行对接

如果全部都是飞塔防火墙，则推荐使用接口模式 IPSec VPN 直接运行路由协议即可。如果存在友商设备场景下，且必须用 GRE over IPSec，那就选择此方式进行对接和部署。

网络需求

通过 GRE over IPSec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信，FGT - BJ 和 FGT - SH 之间运行 OSPF 协议。

网络拓扑



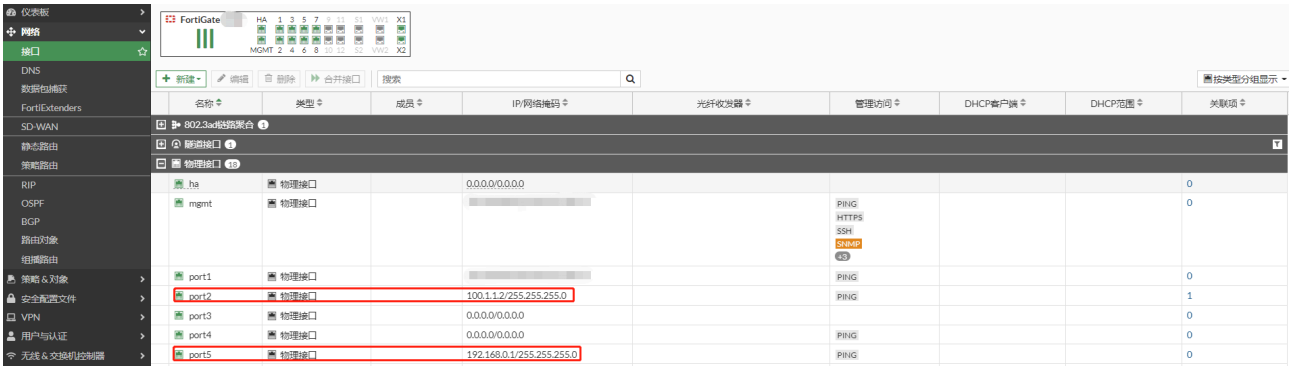
配置步骤

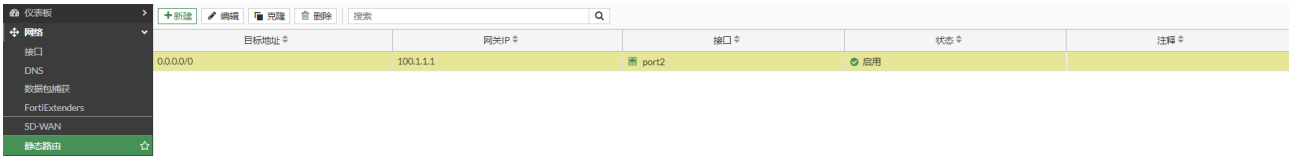
以 FGT - BJ 配置步骤为例，FGT - SH 同理。

> **重要**
>
> 以下配置方法为 Fortinet 独有，不需要手动创建 GRE tunnel 接口，而是通过在 IPSec tunnel 接口上直接封装 GRE 的方式实现 GRE over IPSec，配置较为方便。而官方 Docs 文档介绍的 GRE over IPSec 配置为较旧的方法（不推荐），推荐使用本文方法进行配置（如果对端不是 Fortinet 的设备，请在对端设备上使用传输模式，并配置第二阶段的感兴趣流为 GRE Tunnel 的源目 IP，协议号为 47）。旧版配置方法可参考：
> https://docs.fortinet.com/document/fortigate/7.0.11/administration-guide/799752/gre-over-ipsec ↗

1.基本配置

配置接口 IP 和路由





2.配置 GRE over IPSEC VPN

GRE over IPSEC VPN 只能通过命令行配置（一阶段中需要配置“set encapsulation gre”，二阶段封装模式选择传输模式：“set encapsulation transport-mode”）

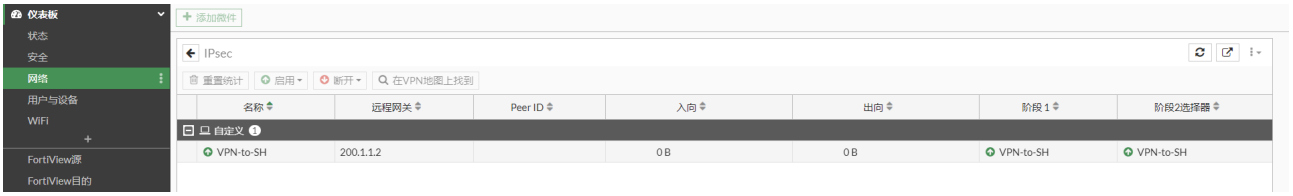
```
VPN第一阶段配置：
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 200.1.1.2
        set psksecret fortinet
        set dpd-retrycount 3
        set dpd-retryinterval 10
        set encapsulation gre
    next
end

VPN第二阶段的配置：
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set phase1name "VPN-to-SH"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set encapsulation transport-mode
        set protocol 47
    next
end
```

3.配置防火墙策略



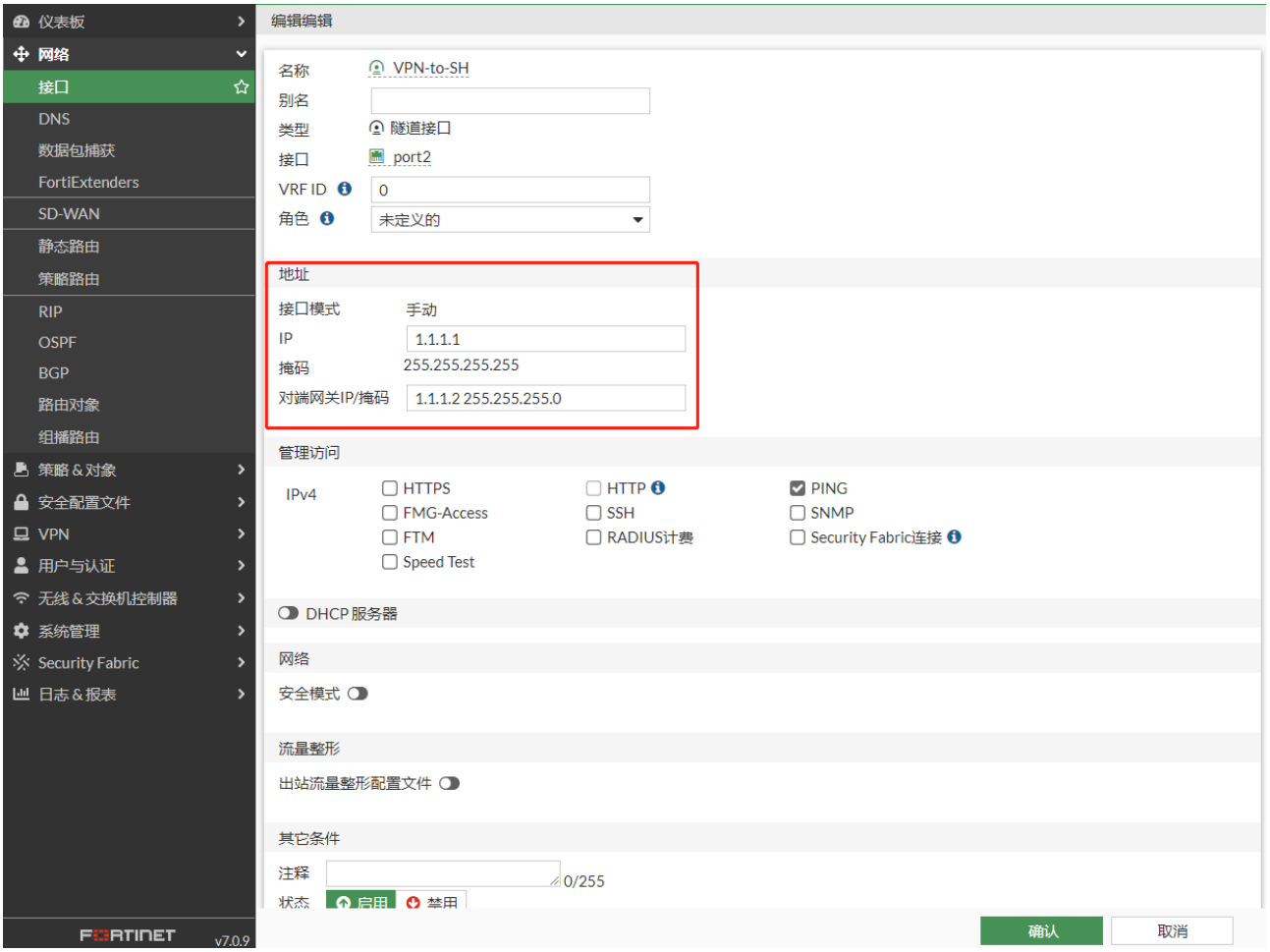
4.查看 IPSEC VPN 状态

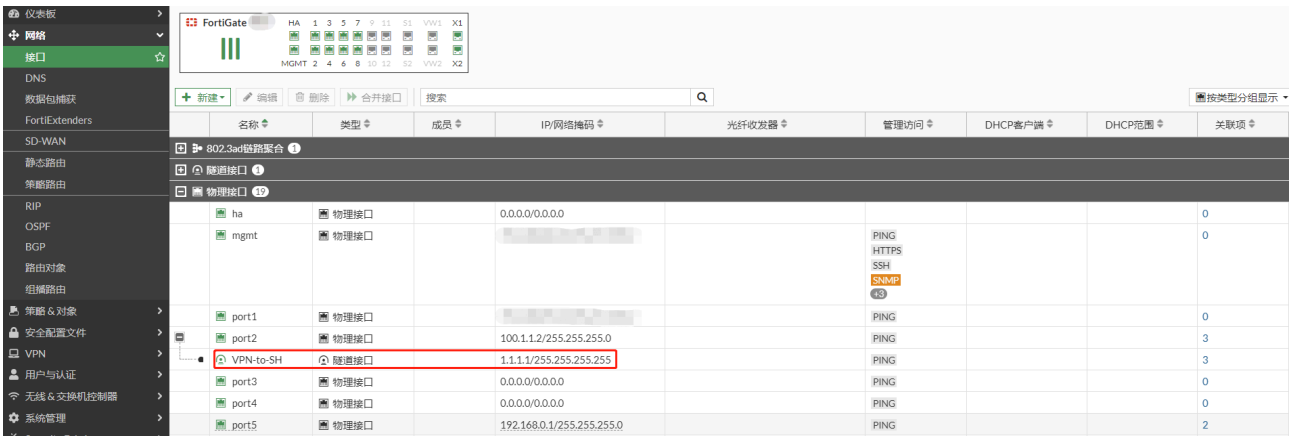
此时 VPN 状态是 UP 的



```
# diagnose vpn ike gateway list

vd: root/0
name: VPN-to-SH
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::200.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 5s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPSec SA: created 2/2  established 2/2  time 0/0/0 ms

  id/spi: 35 5aefc0a3874aef73/d1fa063796f38bbb
  direction: initiator
  status: established 5-5s ago = 0ms
  proposal: aes128-sha256
  key: 35cc029e92fafcd2-1eb88678a2dd3d93
  lifetime/rekey: 86400/86094
  DPD sent/recv: 00000000/00000000

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-SH ver=1 serial=9 100.1.1.2:0→200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::200.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=GRE/549 options[0225]=edg frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

encap-addr: 100.1.1.2→200.1.1.2
proxyid_num=1 child_num=0 refcnt=4 ilast=6 olast=6 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=42
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-SH proto=47 sa=2 ref=3 serial=1 auto-negotiate transport-mode
  src: 47:0.0.0.0-255.255.255.255:0
  dst: 47:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=8327 type=00 soft=0 mtu=1438 expire=42890/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42899/43200
  dec: spi=d4fb6a41 esp=aes key=16 a8da8882351e6d6f0a07ca6a8e9990fa
       ah=sha1 key=20 30e87d9cdb0ec7883005cd21c70cb75b41333a74
  enc: spi=9265e023 esp=aes key=16 8255103778ec791deb6dd576decb6859
       ah=sha1 key=20 479dd7e310e013112dcb9b4f9367c9686389efc5
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=14 dec_npuid=0 enc_npuid=0
  SA:  ref=3 options=8327 type=00 soft=0 mtu=1438 expire=42918/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42927/43200
  dec: spi=d4fb6a42 esp=aes key=16 3ef3be2fdb220fc740c3df3eb6955714
       ah=sha1 key=20 e1b9972a81d688cc32d160b0fc6060b57c42ccac
  enc: spi=9265e022 esp=aes key=16 a247b6207b2244b0ff7822fae3fab091
       ah=sha1 key=20 1c89fa4c89e197efa0886b23eaf722760c63b574
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=14 dec_npuid=0 enc_npuid=0
run_tally=0
```

5.配置 IPSEC VPN 接口地址（此环境中，IPSec 接口将充当 GRE Tunnel 接口使用）





接口地址通过 GRE over IPSEC 能通互 ping 成功。

> **重要**
>
> 配置 IPSec 接口的 remote-gw 后，路由表中会产生 remote-gw 对应的直连路由，该路由可以将 GRE 封装后的流量送入 IPSec 隧道中。

```
# execute ping-options source 1.1.1.1
# execute ping 1.1.1.2
PING 1.1.1.2 (1.1.1.2): 56 data bytes
64 bytes from 1.1.1.2: icmp_seq=0 ttl=255 time=0.7 ms
64 bytes from 1.1.1.2: icmp_seq=1 ttl=255 time=0.3 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=255 time=0.5 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=255 time=0.4 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=255 time=0.6 ms

--- 1.1.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0.3/0.5/0.7 ms

# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
S       1.1.1.0/24 [5/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]
C       1.1.1.1/32 is directly connected, VPN-to-SH
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
C       192.168.88.0/24 is directly connected, mgmt
C       192.168.89.0/24 is directly connected, port1
```

6.配置 OSPF



查看 OSPF 邻居和路由

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
200.1.1.2         1   Full/ -         00:00:34    1.1.1.2         VPN-to-SH(tun-id:200.1.1.2)

# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
S       1.1.1.0/24 [5/0] via VPN-to-SH tunnel 200.1.1.2, [1/0]
```

```
C       1.1.1.1/32 is directly connected, VPN-to-SH
O       1.1.1.2/32 [110/100] via VPN-to-SH tunnel 200.1.1.2, 00:00:01
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
O       192.168.1.0/24 [110/101] via VPN-to-SH tunnel 200.1.1.2, 00:00:01
```

修改 IPSec 协商端口

1. 某些 ISP 会阻断 UDP 端口 500 或 UDP 端口 4500，导致 IPsec VPN 无法协商和建立。FortiGate 支持修改 IPsec 的协商端口来避免该问题。

```
config system settings
    set ike-port 10000    //默认udp port 500，端口范围udp port <1024> to <65535>
end
```

2. 修改后所有的隧道将使用指定的端口来协商。

3. 对于业务层面：

   - 非 NAT-T 环境下仍然使用 ESP 封装。
   - NAT-T 下使用上述配置的 IKE port 来封装 UDP Header。

拨号 VPN

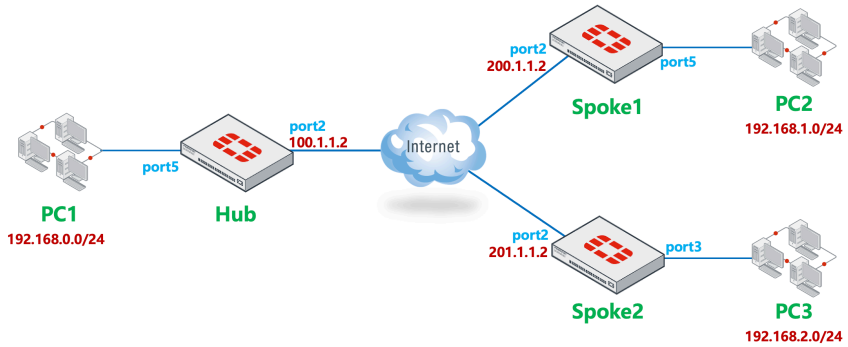
- Hub-and-spoke 模式
- 拨号 VPN 运行动态路由
- ADVPN 模式
- 野蛮模式/network-id 适用场景
- FortiClient 拨号
- FortiClient 使用证书认证连接 IPSec
- 基于 TCP 的 FortiClient 拨号
- IPSec 多用户组访问控制

Hub-and-spoke 模式

网络需求

某公司总部内部有一台 OA 服务器，其余分 2 个支机构都需要通过 vpn 拨入总部内网对 OA 服务器进行访问，为了方便配置，总部不想有太多的配置，总部只建立一条 vpn 隧道，实现所有分支机构和总部的通讯，SPOKE 之间的流量通过 HUB 进行中转。

网络拓扑

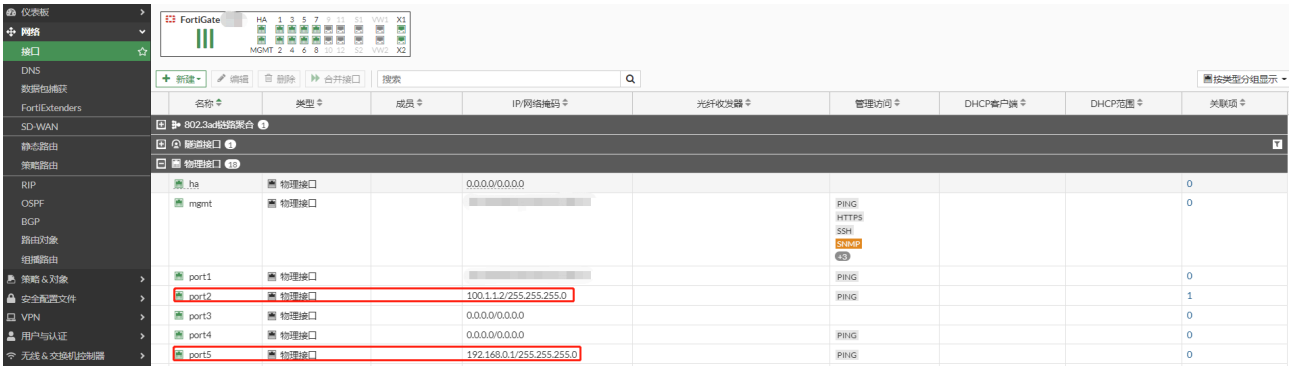


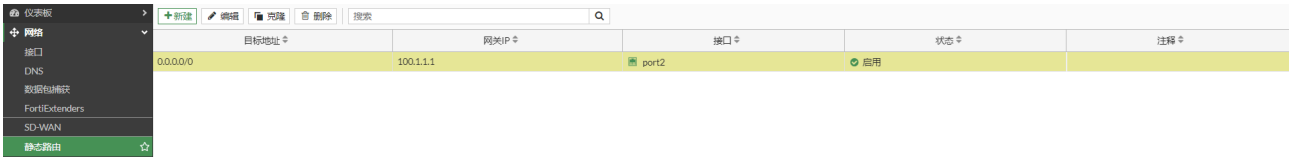
配置步骤

###HUB 配置

1.基本配置

配置接口 IP 和路由





2.配置 IPSEC VPN

总部（HUB）IPsec VPN 配置说明：

- HUB 使用动态拨号的 VPN 方式。 因为分支可能来自任意地方的 IP，另外分支可能使用 PPPOE 拨号，公网 IP 随时会变更，因此没有办法确认对方的公网 IP，只能动态拨号方式的 IPsec VPN。
- HUB 使用主模式。

  其实 HUB 的阶段 1 使用主模式或野蛮模式二者都是可以工作的，只要 hub 和 spoke 之间使用相同的阶段 1 模式即可。

  如果 HUB 相同接口下（如 wan1 口）只有一条动态拨号的 IPsec VPN，可以简单的直接使用主模式，接收 Peer ID 为 any 即可。

  如果 HUB 相同接口下（如 wan1 口）有两条或以上 type 是 dynamic 的阶段 1，就必须使用野蛮模式，因为阶段 1 type 是 dynamic 的 remote-gw 为 0.0.0.0，存在多个时会导致冲突，结果匹配到非预期的阶段 1，这时候必须使用野蛮模式的 Peer ID 来区分/匹配）。

一条动态拨号 VPN 的场景下我们 HUB 选择最简单的主动模式 + 任意 Peer ID + 预共享密钥。这种选择优势是在 SPOKE 看来感觉就像是建立一个很普通的 LAN - TO - LAN 的 IPsec VPN。

- HUB 第二阶段感兴趣流保持默认的 0.0.0.0/0.0.0.0↔0.0.0.0/0.0.0.0。

> **重要**
>
> 如果使用静态路由来转发 Hub 与 Spoke 之间的流量，由于 Hub 是被动的接受 Spoke 的协商，在没有协商之前 Hub 不知道 Spoke 的具体网段/感兴趣流，Hub 保持默认 0.0.0.0/0↔0.0.0.0/0 即可，会自动学习分支的源保护网段，同时自动产生反向的去往感兴趣流的回程静态路由（一阶段开启了 add-route 功能），不要在 Hub 上手动配置去往 Spoke 内网的静态路由，否则会出现 VPN 技术 → IPSec VPN → IPSec VPN 排错 → Hub-Spoke 静态路由问题 ↗章节的问题。
>
> 而 Spoke 是需要配置明细的源保护网段的，如 192.168.1.0/24↔0.0.0.0/0，Hub 通过 add-route 功能自动添加去往 Spoke 源保护网段（192.168.1.0/24）的静态路由。

创建 IPSEC VPN。



IPSEC VPN 阶段一配置。



IPSEC VPN 阶段二配置。



VPN 创建完成。



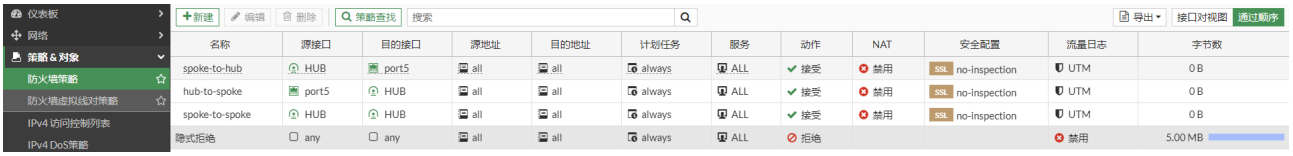
对应的命令行

```bash
config vpn ipsec phase1-interface
    edit "HUB"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route enable    //确保Hub的add-route开启，默认开启//
        set dpd on-idle
        set psksecret ENC
o00sB0EFtnTN9mJd5/xShv4v22vbn3NpgGIepVJSXGup1iQ++HB0zjDaOuj8jIyKjxZ3yhYqjtQXL9HJ5aoTQ4rlVRiBlw7jcxbSB8lzbU0As8YhUob3Z6T2St
vFeRC1VUDac8wmjsPbkUkOq0YlRZ22DoQMe2t+lG6orUAgDlY6iko6mlodLfmMiRxOFNIASP5UNg==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "HUB"
```

```
        set phase1name "HUB"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set keepalive enable
    next
end
```

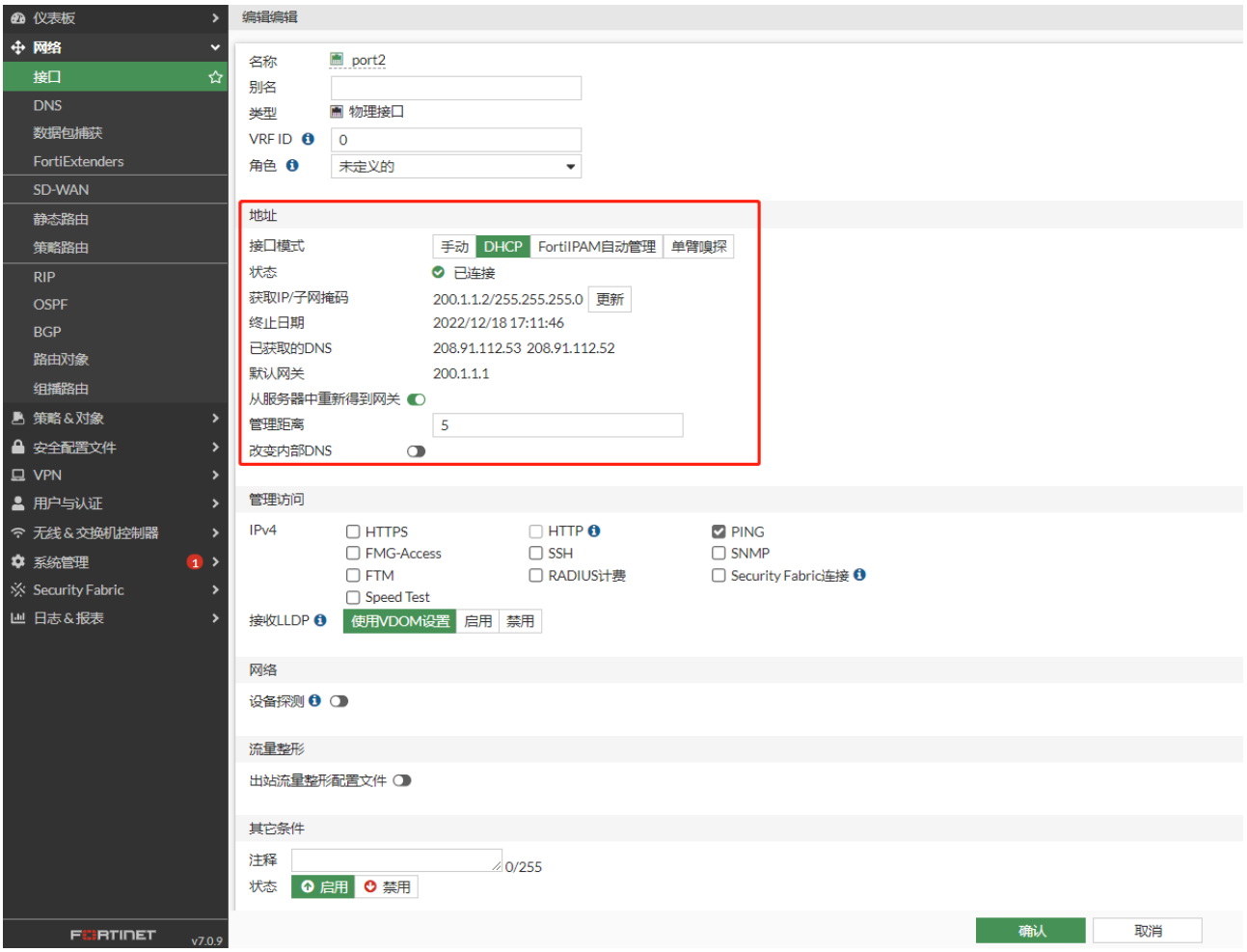
3.配置策略

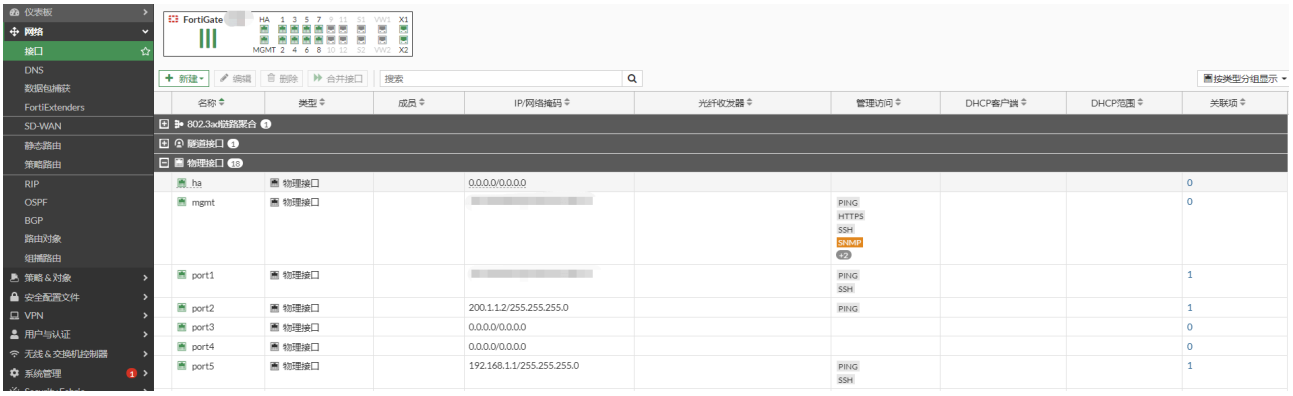
创建 3 条策略，从上到下分别是：分支到总部的放通策略；总部到分支的放通策略；分支到分支的放通策略。



SPOKE1 配置

1.基本配置



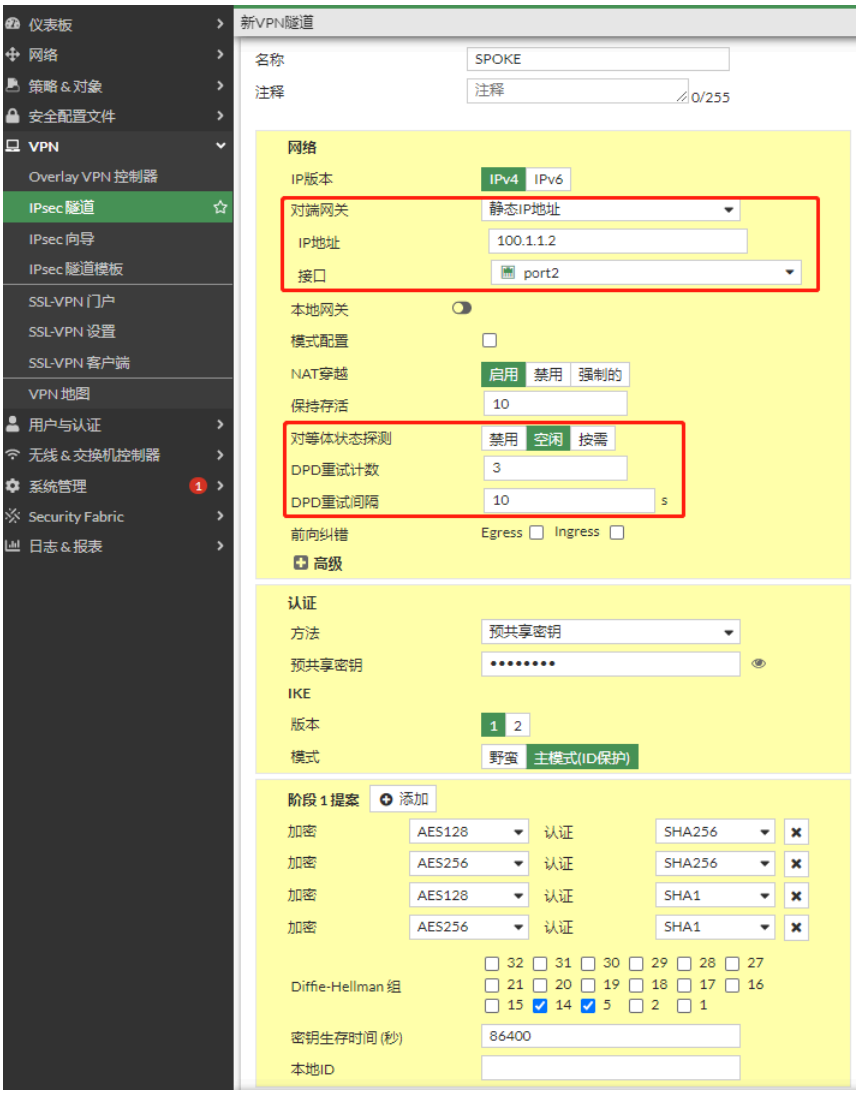


2.配置 IPSEC VPN

创建 IPSEC VPN。



IPSEC VPN 阶段一配置

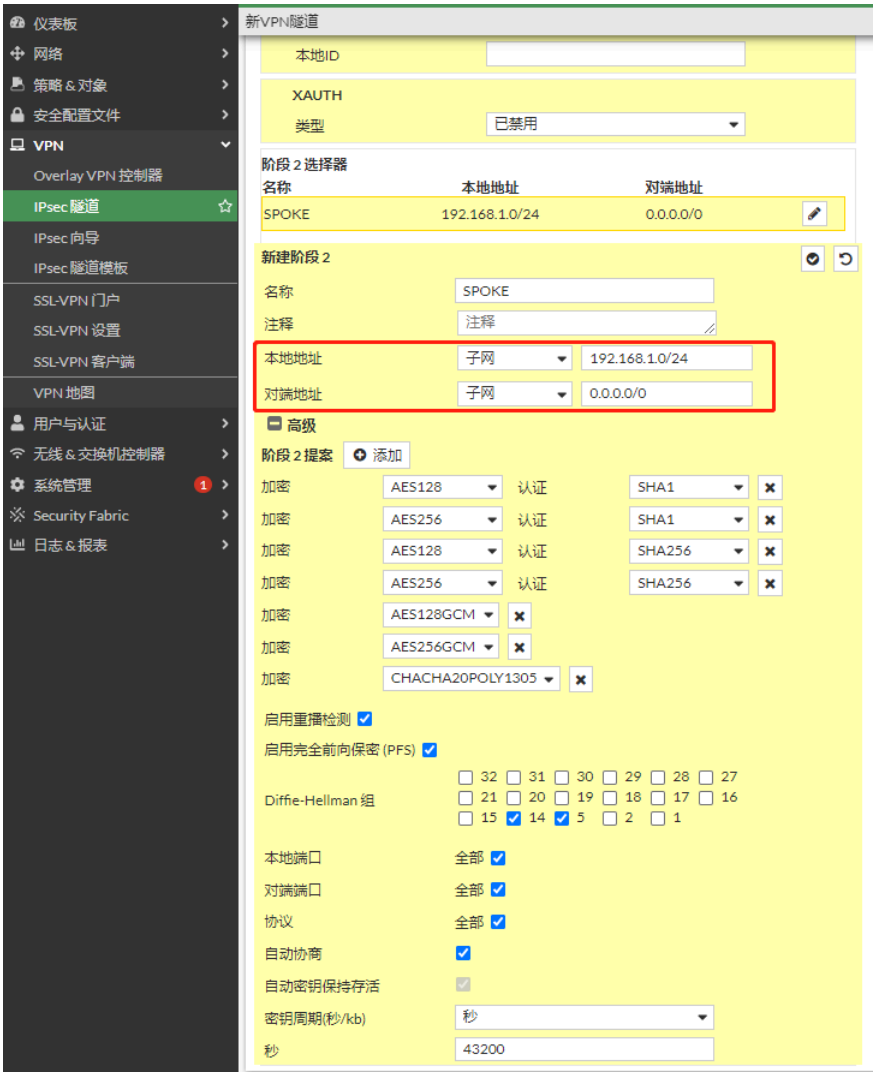


IPSEC VPN 阶段二配置。

重要

如果使用静态路由来转发 Hub 与 Spoke 之间的流量，由于 Hub 是被动的接受 Spoke 的协商，在没有协商之前 Hub 不知道 Spoke 的具体网段/感兴趣流，Hub 保持默认 0.0.0.0/0↔0.0.0.0/0 即可，会自动学习分支的源保护网段，同时自动产生反向的去往感兴趣流的回程静态路由（一阶段开启了 add-route 功能），不要在 Hub 上手动配置去往 Spoke 内网的静态路由，否则会出现 VPN 技术 → IPSec VPN → IPSec VPN 排错 → Hub-Spoke 静态路由问题 ↗章节的问题。

而 Spoke 是需要配置明细的源保护网段的，如 192.168.1.0/24↔0.0.0.0/0，Hub 通过 add-route 功能自动添加去往 Spoke 源保护网段（192.168.1.0/24）的静态路由。



VPN 创建完成。



对应的命令行

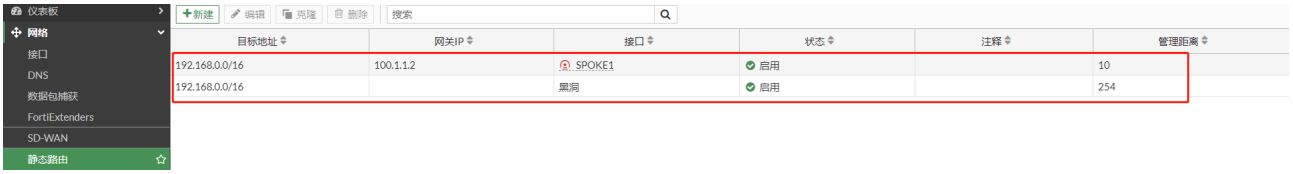
```
config vpn ipsec phase1-interface
    edit "SPOKE1"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 100.1.1.2
        set psksecret ENC
/nQqtlGc19JE/tKd0G4lHD1r7279qcDTpVAlNPW1djHDvxMhIuZctGGr1lw1l/Odi3WkvMrhEDydpVxEtvSWPBUp2+pbhB8fXowEkkoYtaGvvmA0ohj+Bk7NOE
THZKUQw9D7b8ibhF5HjbbEphvWb6QwDgI+pjwhp56AZp6sF05L3RLR2nLMEqJKCf6CBog5Se/MbA==
```

```
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE1"
        set phase1name "SPOKE1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.1.0 255.255.255.0
    next
end
```

3.配置路由

这里路由的目标是 192.168.0.0/16，是 HUB 和其他 SPOKE 的业务网段汇总（分支之间没有单独建立 VPN，从 HUB 互通）。如果只需要访问总部，则写总部的明细网段即可，即 192.168.0.0/24。

管理距离为 254 的黑洞路由是为了 VPN 中断的时候防止产生错误的 NAT 会话。



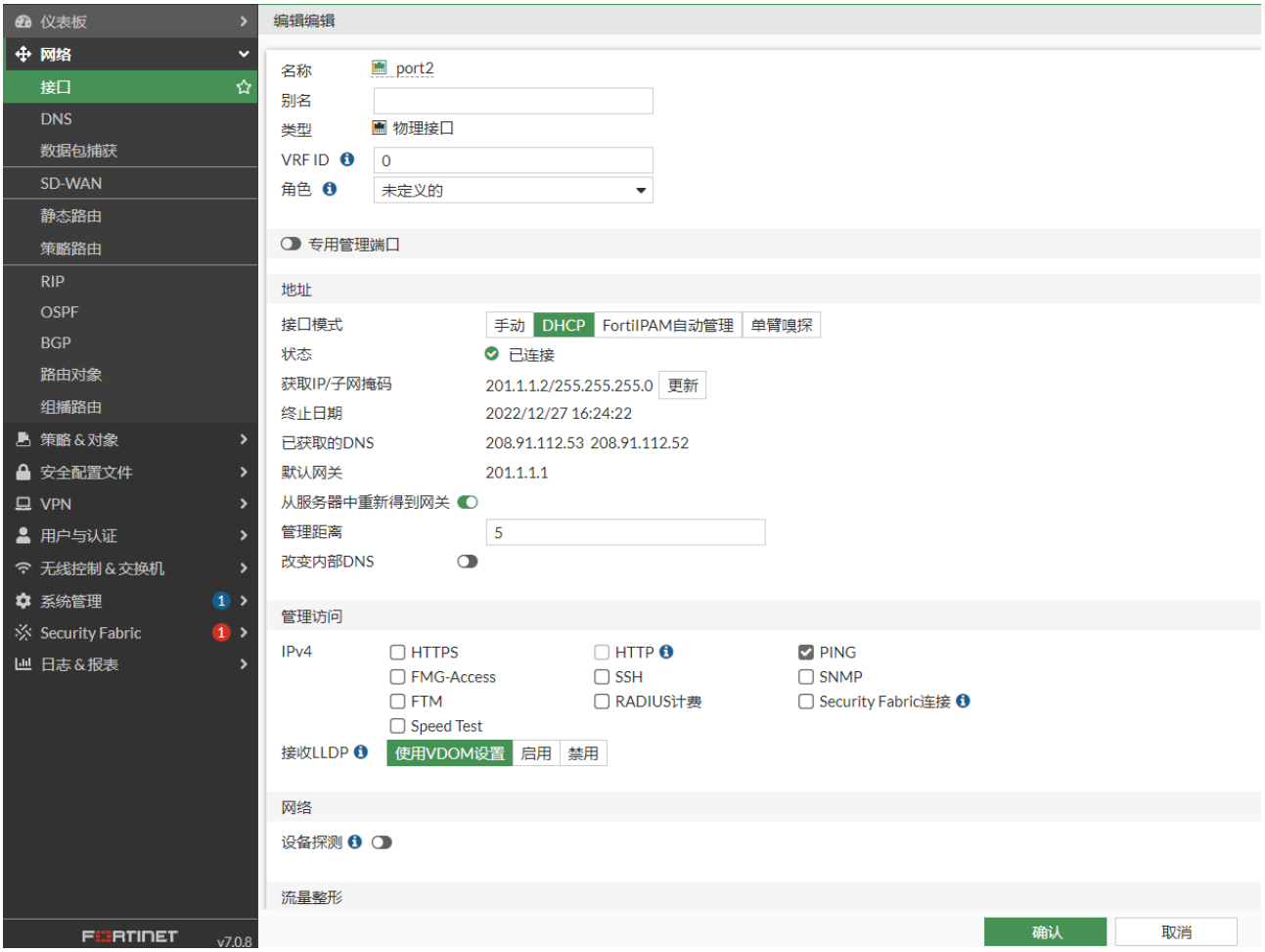
4.配置策略

由于 SPOKE1 开启了自动连接，当配置策略后，就会主动与 HUB 协商，策略中的接口已经是绿色的，说明 VPN 接口已经 UP。



SPOKE2 配置

1.基本配置



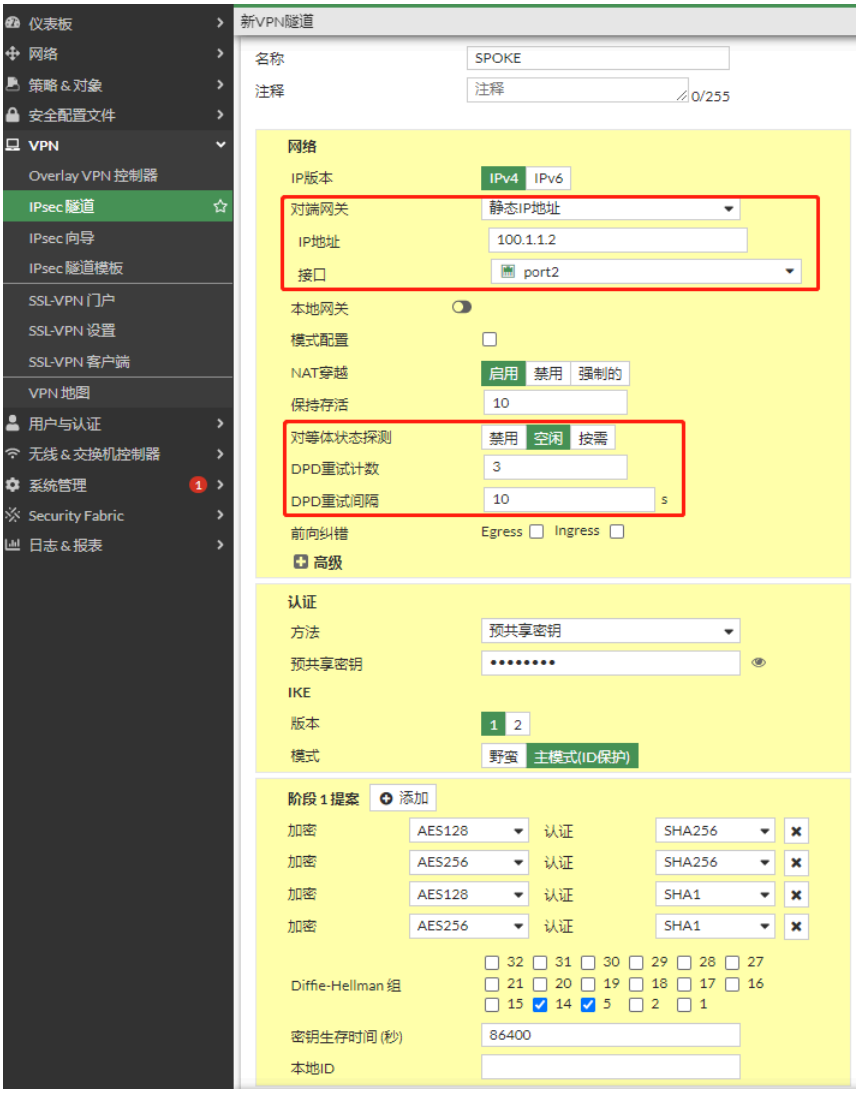


2.配置 IPSEC VPN

创建 IPSEC VPN。



IPSEC VPN 阶段一配置

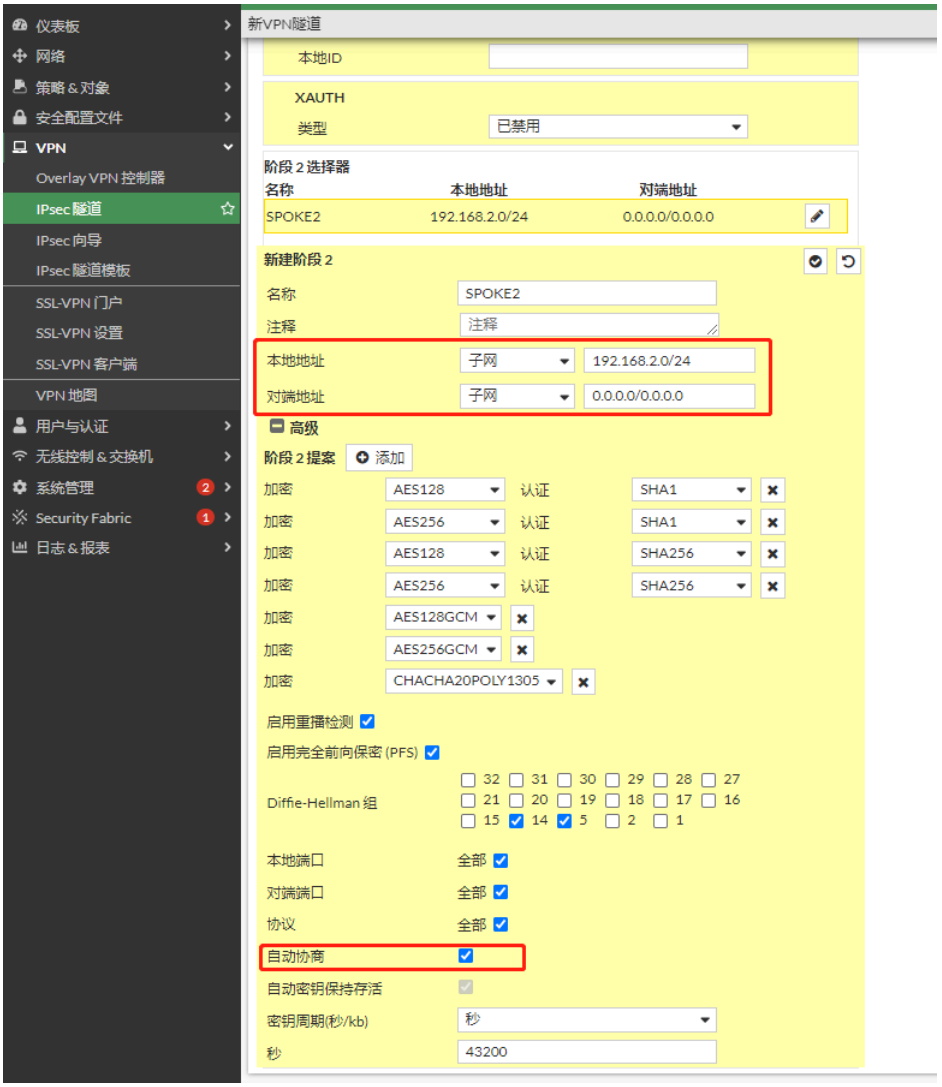


IPSEC VPN 阶段二配置。

重要

如果使用静态路由来转发 Hub 与 Spoke 之间的流量，由于 Hub 是被动的接受 Spoke 的协商，在没有协商之前 Hub 不知道 Spoke 的具体网段/感兴趣流，Hub 保持默认 0.0.0.0/0↔0.0.0.0/0 即可，会自动学习分支的源保护网段，同时自动产生反向的去往感兴趣流的回程静态路由（一阶段开启了 add-route 功能），不要在 Hub 上手动配置去往 Spoke 内网的静态路由，否则会出现 VPN 技术 → IPSec VPN → IPSec VPN 排错 → Hub-Spoke 静态路由问题 ↗章节的问题。

而 Spoke 是需要配置明细的源保护网段的，如 192.168.1.0/24↔0.0.0.0/0，Hub 通过 add-route 功能自动添加去往 Spoke 源保护网段（192.168.1.0/24）的静态路由。



完成 VPN 创建。



对应的命令行

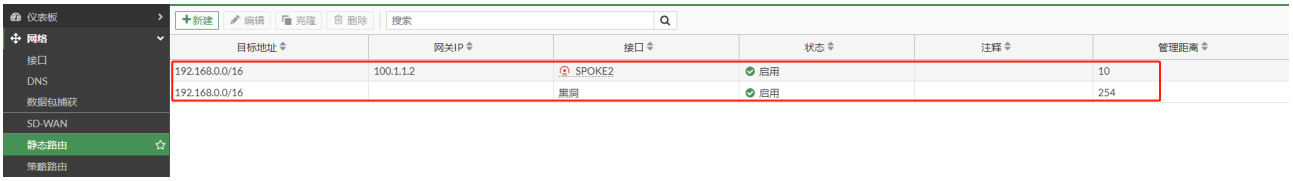
```
config vpn ipsec phase1-interface
    edit "SPOKE2"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 100.1.1.2
        set psksecret ENC
tW+gF8+AfteBwl7HXJwMXlNTH3LGBMv0D0d1krXTNETF4/AFPUWOQvD96qoK7aQNOGOben9RUCqQ7BX2GT9lisno5ilp1zp1HlFaqkTe2l6BPL3rM1BLMVYF7k
CW/ozSPHSXU14OIlZ+sNQioiskUhMk7RFHbkXALqDBDVMIIfM/odiP+kpHZmya3+mQDDYCjnEA3Q==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE2"
        set phase1name "SPOKE2"
```

```
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.2.0 255.255.255.0
    next
end
```

3.配置路由

这里路由的目标是 192.168.0.0/16，是 HUB 和其他 SPOKE 的业务网段汇总（分支之间没有单独建立 VPN，从 HUB 互通）。如果只需要访问总部，则写总部的明细网段即可，即 192.168.0.0/24。

管理距离为 254 的黑洞路由是为了 VPN 中断的时候防止产生错误的 NAT 会话。



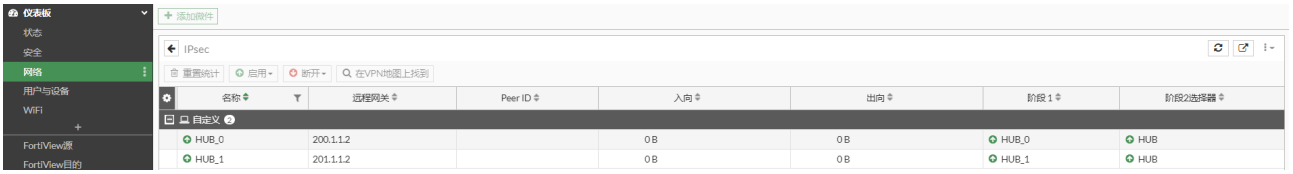
4.配置策略

由于 SPOKE2 开启了自动连接，当配置策略后，就会主动与 HUB 协商，策略中的接口已经是绿色的，说明 VPN 接口已经 UP。



查看 VPN 状态

1.HUB 端 VPN 状态




```
# diagnose vpn  ike gateway list

vd: root/0
name: HUB_0
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::10.0.0.38
remote_location: 0.0.0.0
network-id: 0
created: 222s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 5 9e99b894d5de25e7/f6f50471f8504b5c
  direction: responder
  status: established 222-222s ago = 0ms
  proposal: aes128-sha256
  key: 12b4b577207df3b1-72d0c8ab630bca42
  lifetime/rekey: 86400/85907
  DPD sent/recv: 00000017/00000000

vd: root/0
name: HUB_1
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 201.1.1.2:500
tun_id: 201.1.1.2/::10.0.0.39
remote_location: 0.0.0.0
```

```
network-id: 0
created: 115s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 6 61a95053c4a2a7bc/283be8ae94d730cc
  direction: responder
  status: established 115-115s ago = 0ms
  proposal: aes128-sha256
  key: 29af7f5a8312e219-5b96fb7006b7bab6
  lifetime/rekey: 86400/86014
  DPD sent/recv: 0000000a/00000005

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=HUB_0 ver=1 serial=41 100.1.1.2:0->200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::10.0.0.38 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=4 olast=4 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=23
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:192.168.1.0-192.168.1.255:0
  SA:  ref=3 options=2a6 type=00 soft=0 mtu=1438 expire=42969/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43191/43200
  dec: spi=14b71563 esp=aes key=16 8c3637326e7c38dbefc9bac5333125ca
       ah=sha1 key=20 a8fe40c940d8f4321d72ff9f0b0018c5b52d4f1e
  enc: spi=bcdbb5b1 esp=aes key=16 d24189aae387a6a13930e4998c7fc653
       ah=sha1 key=20 58cbd4a72ea1ad459c322ab483fe0c78f483c170
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=3e dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=HUB_1 ver=1 serial=42 100.1.1.2:0->201.1.1.2:0 tun_id=201.1.1.2 tun_id6=::10.0.0.39 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=7 olast=7 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=10
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:192.168.2.0-192.168.2.255:0
  SA:  ref=3 options=2a6 type=00 soft=0 mtu=1438 expire=43069/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43187/43200
  dec: spi=14b71564 esp=aes key=16 2a26ef846d7b8174388e862859ea7680
       ah=sha1 key=20 8915e7b5a2f18de14e3f239a7921653949a2c205
  enc: spi=1719f302 esp=aes key=16 1b704defe11ffdf2f26f4f2482f65eab
       ah=sha1 key=20 0b6a0e5d527eed61ca2b002ea8797b270a61eb87
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=201.1.1.2 npu_lgwy=100.1.1.2 npu_selid=3f dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=HUB ver=1 serial=3f 100.1.1.2:0->0.0.0.0:0 tun_id=10.0.0.15 tun_id6=::10.0.0.15 dst_mtu=0 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=44854899 olast=44854899 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=10000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

查看 HUB 端生成的路由

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
S       192.168.1.0/24 [15/0] via HUB tunnel 200.1.1.2, [1/0]
S       192.168.2.0/24 [15/0] via HUB tunnel 201.1.1.2, [1/0]
```

2.SPOKE1 端 VPN 状态




```
# diagnose vpn ike gateway  list

vd: root/0
name: SPOKE1
version: 1
interface: port2 10
addr: 200.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 293s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 0 9e99b894d5de25e7/f6f50471f8504b5c
  direction: initiator
  status: established 293-293s ago = 0ms
  proposal: aes128-sha256
  key: 12b4b577207df3b1-72d0c8ab630bca42
  lifetime/rekey: 86400/85806
  DPD sent/recv: 00000000/0000001e

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE1 ver=1 serial=17 200.1.1.2:0→100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=5 olast=5 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=10000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE1 proto=0 sa=1 ref=2 serial=2 auto-negotiate
  src: 0:192.168.1.0-192.168.1.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=18227 type=00 soft=0 mtu=1438 expire=42607/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=bcdbb5b1 esp=aes key=16 d24189aae387a6a13930e4998c7fc653
       ah=sha1 key=20 58cbd4a72ea1ad459c322ab483fe0c78f483c170
```

```
  enc: spi=14b71563 esp=aes key=16 8c3637326e7c38dbefc9bac5333125ca
       ah=sha1 key=20 a8fe40c940d8f4321d72ff9f0b0018c5b52d4f1e
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=200.1.1.2 npu_selid=18 dec_npuid=0 enc_npuid=0
run_tally=0
```

查看 SPOKE1 端的路由

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       192.168.0.0/16 [10/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
C       192.168.1.0/24 is directly connected, port5
C       200.1.1.0/24 is directly connected, port2
```

3.SPOKE2 端 VPN 状态




```
# diagnose vpn  ike gateway list

vd: root/0
name: SPOKE2
version: 1
interface: port2 6
addr: 201.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 302s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 0 61a95053c4a2a7bc/283be8ae94d730cc
  direction: initiator
  status: established 302-302s ago = 10ms
  proposal: aes128-sha256
  key: 29af7f5a8312e219-5b96fb7006b7bab6
  lifetime/rekey: 86400/85797
  DPD sent/recv: 00000018/0000000a

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE2 ver=1 serial=d 201.1.1.2:0→100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=4 olast=4 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=24
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:192.168.2.0-192.168.2.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=18203 type=00 soft=0 mtu=1438 expire=42596/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=1719f302 esp=aes key=16 1b704defe11ffdf2f26f4f2482f65eab
```

```
      ah=sha1 key=20 0b6a0e5d527eed61ca2b002ea8797b270a61eb87
  enc: spi=14b71564 esp=aes key=16 2a26ef846d7b8174388e862859ea7680
      ah=sha1 key=20 8915e7b5a2f18de14e3f239a7921653949a2c205
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=201.1.1.2 npu_selid=a dec_npuid=0 enc_npuid=0
run_tally=0
```

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 201.1.1.1, port2, [1/0]
S       192.168.0.0/16 [10/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
```

业务测试

1. HUB 端业务测试

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6140132  bytes 1760274153 (1.6 GiB)
        RX errors 0  dropped 3749603  overruns 0  frame 0
        TX packets 51686  bytes 3833219 (3.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问SPOKE1
PC1# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=62 time=1.42 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=62 time=0.788 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=62 time=0.751 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=62 time=0.688 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.688/0.913/1.427/0.300 ms

访问SPOKE2
PC1# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=62 time=1.09 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.935 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.991 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=1.06 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.935/1.021/1.098/0.070 ms
```

2. SPOKE1 端业务测试

```
PC2# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::1a6c:e61:d2b9:a415  prefixlen 64  scopeid 0×20<link>
        inet6 2001::2  prefixlen 64  scopeid 0×0<global>
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 5904889  bytes 459674552 (438.3 MiB)
        RX errors 0  dropped 58  overruns 0  frame 0
        TX packets 3224540  bytes 205411564402 (191.3 GiB)
```

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问HUB
PC2# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.746 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.765 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.862 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=0.677 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.677/0.762/0.862/0.071 ms

访问SPOKE2
PC2# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=61 time=1.14 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=61 time=0.952 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=61 time=1.51 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.886 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.886/1.124/1.519/0.248 ms
```

3. SPOKE2 端业务测试

```
PC3# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::2652:4dd7:5d0e:941d  prefixlen 64  scopeid 0x20<link>
        inet6 240e:604:109:39::216  prefixlen 64  scopeid 0x0<global>
        ether 00:0c:29:37:f0:ac  txqueuelen 1000  (Ethernet)
        RX packets 9638867  bytes 205844622412 (191.7 GiB)
        RX errors 0  dropped 118  overruns 0  frame 0
        TX packets 5737914  bytes 378901312 (361.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问HUB
PC2# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.848 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.935 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.899 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=1.04 ms
--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.848/0.931/1.045/0.081 ms

访问SPOKE1
PC3# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=61 time=1.13 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=61 time=1.22 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=61 time=1.21 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=61 time=1.04 ms
--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.043/1.151/1.220/0.075 ms
```

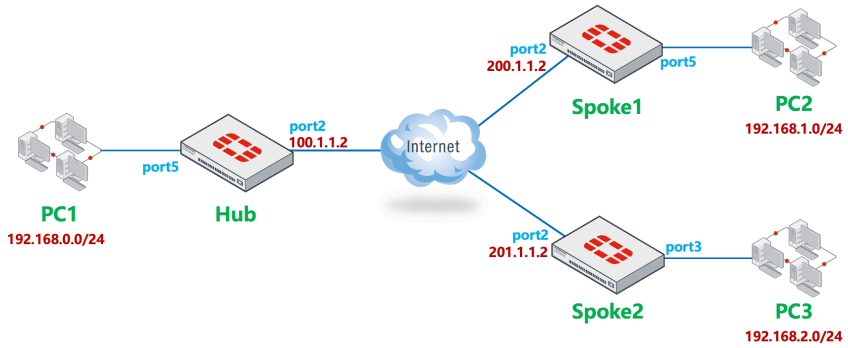
拨号 VPN 运行动态路由

网络需求

某公司总部内部有一台 OA 服务器，其余分 2 个支机构都需要通过 vpn 拨入总部内网对 OA 服务器进行访问，为了方便配置，总部不想有太多的配置，总部只建立一条 vpn 隧道，实现所有分支机构和总部的通讯。同时需要运行动态路由协议将总部和分支的路由全部打通，SPOKE 之间的流量通过 HUB 进行中转。

##网络拓扑

HUB and SPOKE 通过动态路由学习到整网的业务网段路由，SPOKE 之间相互的流量通过 HUB 进行中转。



VPN Tunnel IP 地址分配，以及 BGP 的规划：

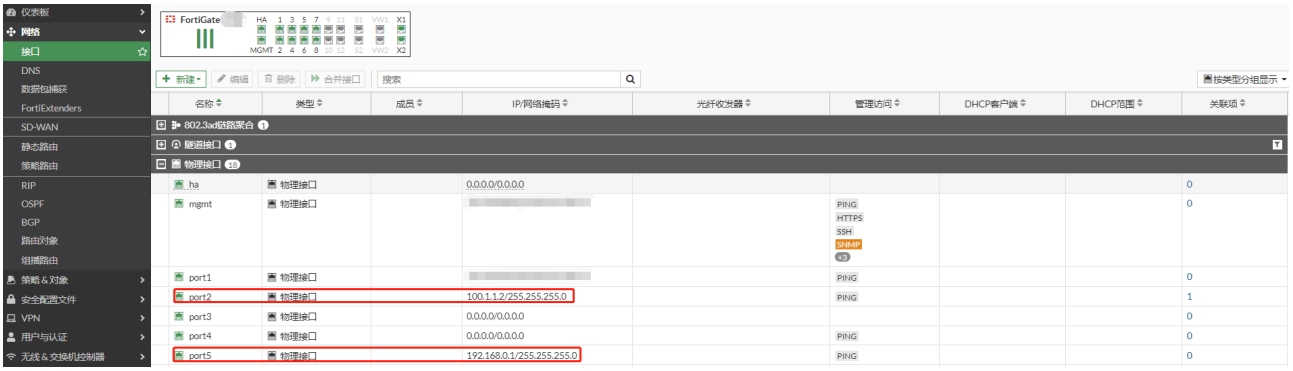
角色	公网 IP	私网网段	VPN 隧道 IP	BGP 信息
HUB	100.1.1.2	192.168.0.1/24	10.10.10.1	AS 65001 RR
SPOKE1	200.1.1.2	192.168.1.1/24	10.10.10.2	AS 65001 RR Client
SPOKE2	201.1.1.2	192.168.2.1/24	10.10.10.3	AS 65001 RR Client
SPOKE3	...	...	...	...
SPOKEX	...	...	...	...

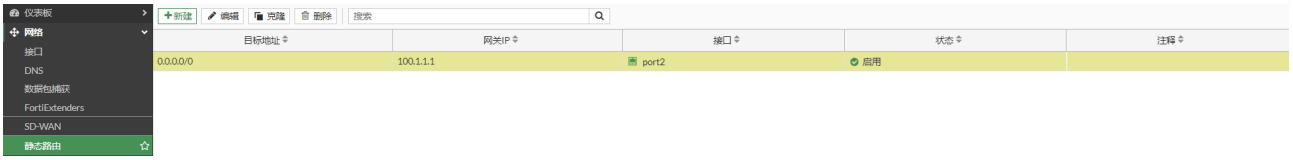
配置步骤

HUB 端配置

1.基本配置

配置接口 IP 和路由





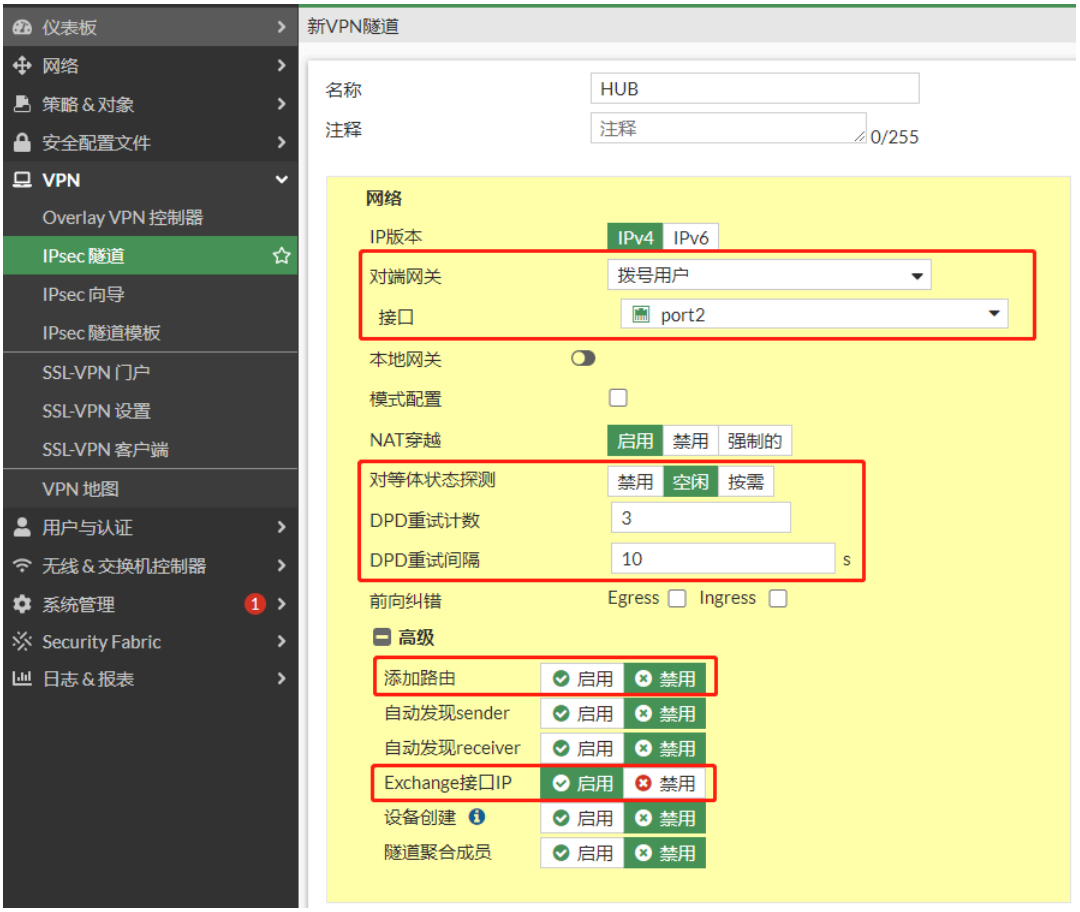
2.配置 IPSEC VPN

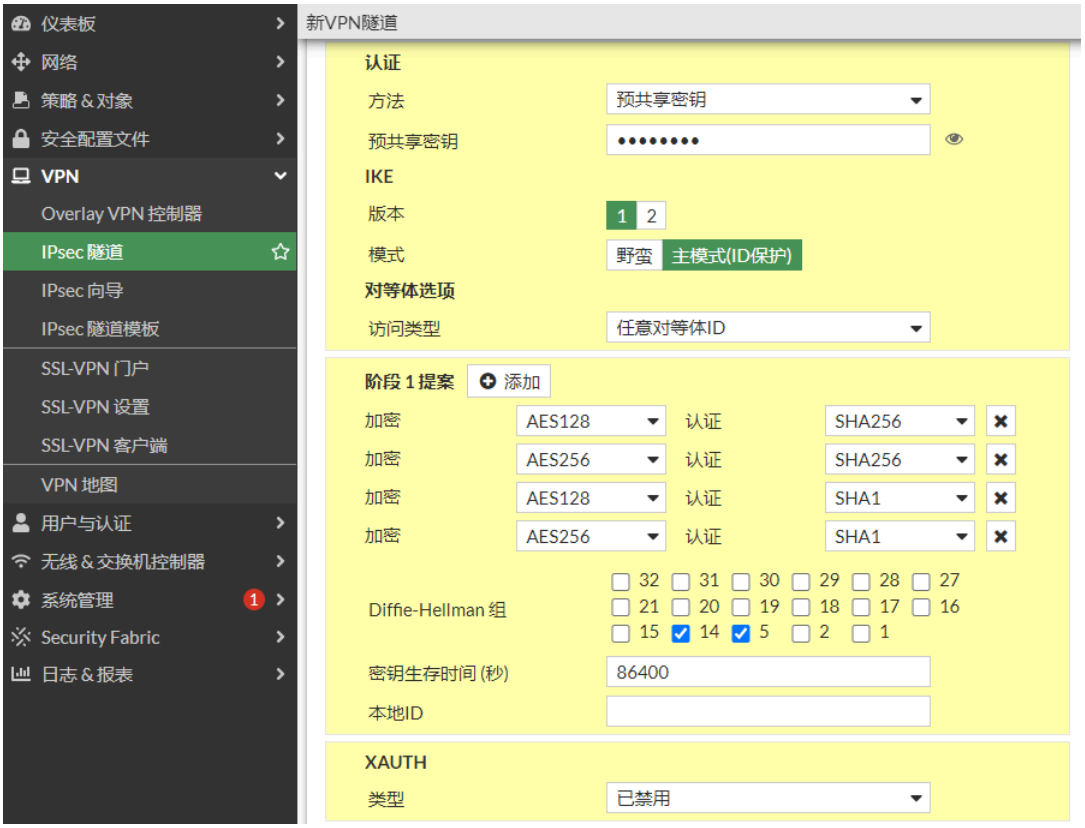
创建 IPSEC VPN。



IPSEC VPN 阶段一配置。

由于 HUB 和 SPOKE 之间是通过动态路由学习，因此“添加路由”选项禁用，不用 HUB 自动学习 SPOKE 的路由；Exchange 接口 IP 打开，需要交换 HUB 和 SPOKE 之间 VPN 接口的 IP。





IPSEC VPN 阶段二配置。



创建完成。



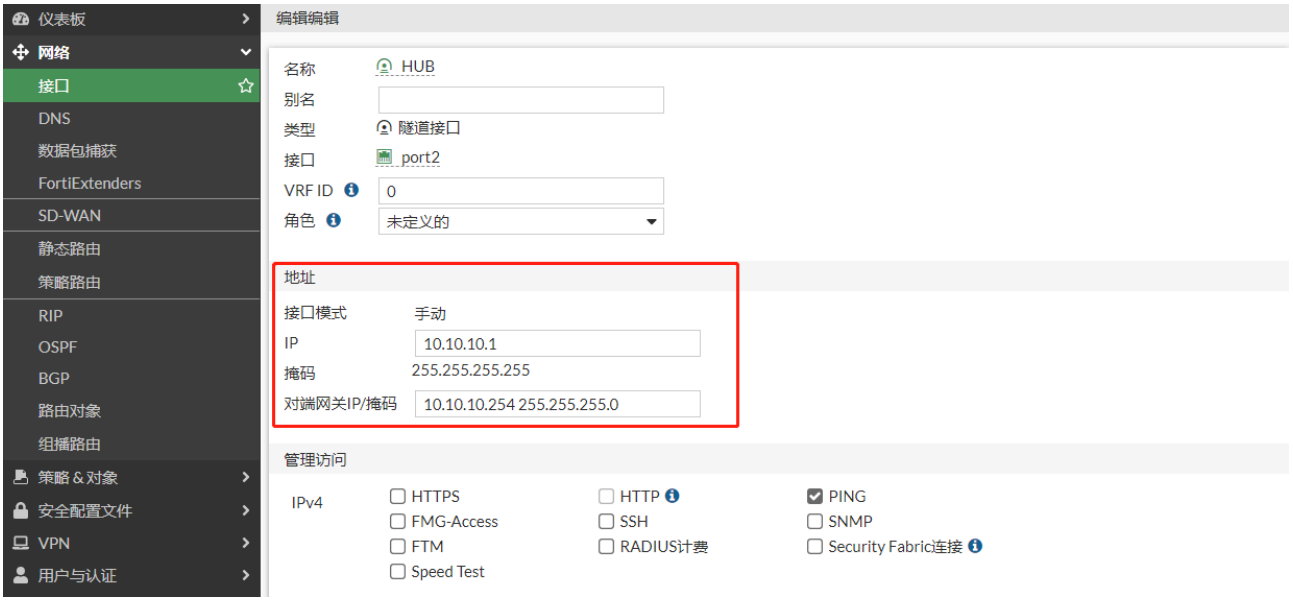
对应的命令行

```
config vpn ipsec phase1-interface
    edit "HUB"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set exchange-interface-ip enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set psksecret ENC
m9c6Qk4uhwWLRTRDI2EzYQym02Uma0709h07KB2pmS0Ek+8ZUOpRTH6OZdFEZPdlZktXajf/KRkR6SNO7UwW27pYAWoRygwcWhvJrO67tIWl5AksoOX3uy2YTj
FX+kVbrw5WTVoLogqtJ4yPk7coZpMKUvqJkAv8flqIIi3EfY4TgY2avzSOkBVm8ZLgokTkmt3PMQ==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "HUB"
```

```
        set phase1name "HUB"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set keepalive enable
    next
end
```

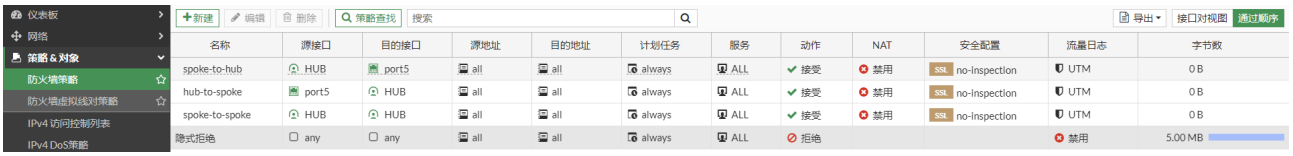
3.配置 VPN 接口 IP

HUB 端的”对端网关 IP“ 10.10.10.254 是不被 Spoke 所使用的预留 IP，IPsec Tunnel 是一个点对点的隧道，但是 ADVPN 中这条隧道需要同时对应多个 SPOKE，因此不能将 Remote IP 写成一个存在的 SPOKE 端 IP。



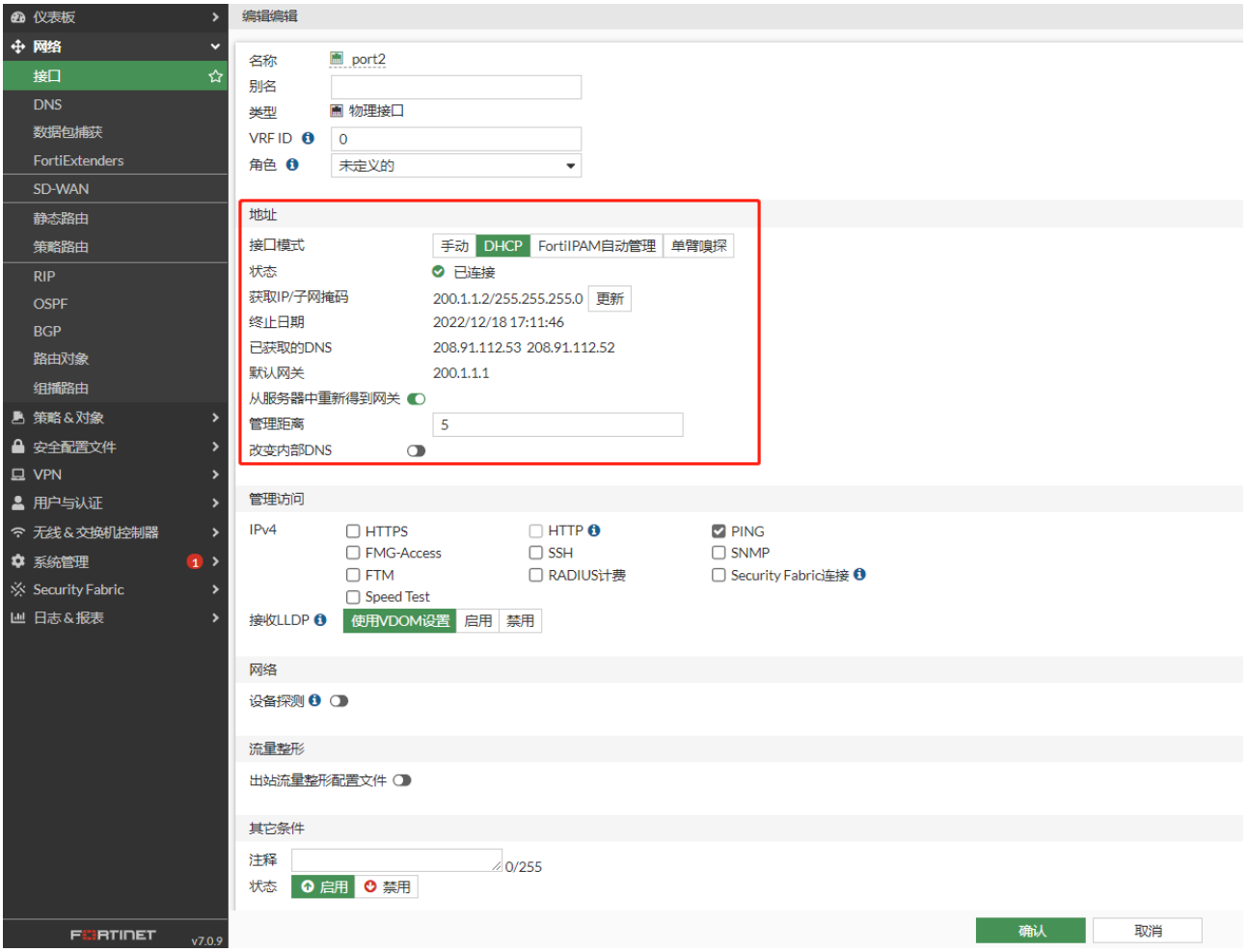
4.配置策略

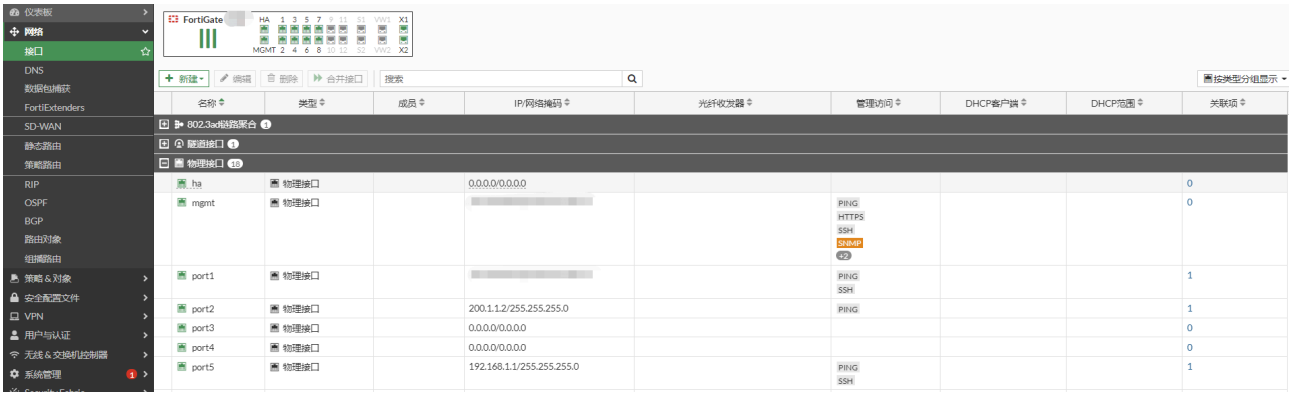
创建 3 条策略，从上到下分别是：分支到总部的放通策略；总部到分支的放通策略；分支到分支的放通策略。



SPOKE1 配置

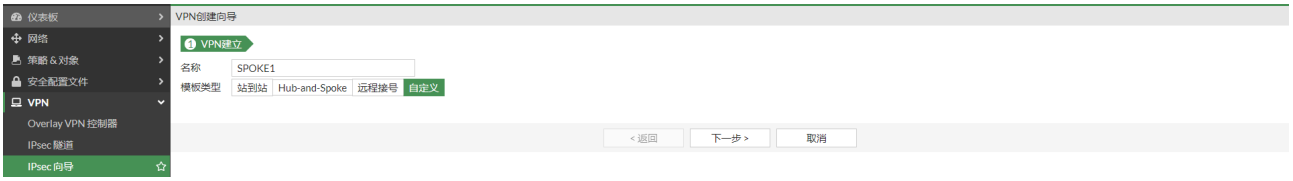
1.基本配置





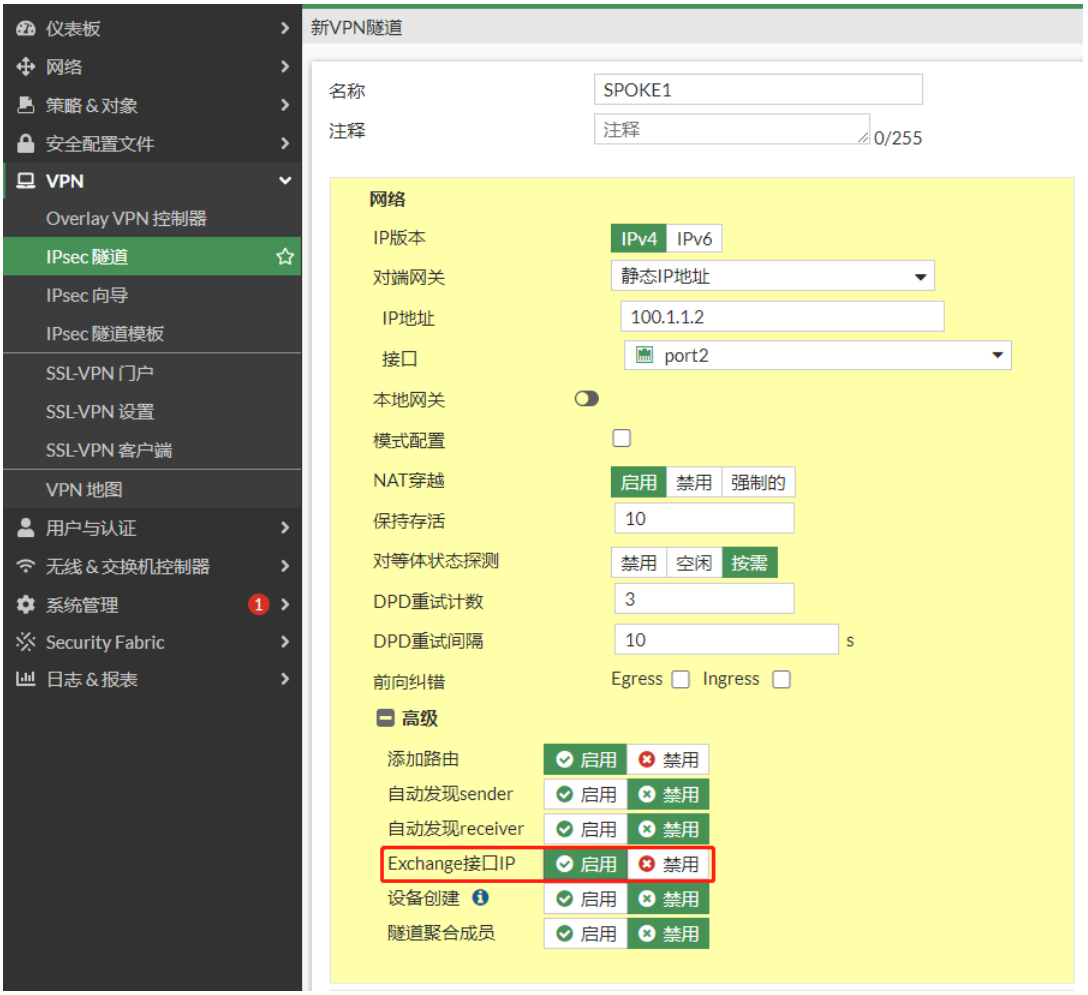
2.配置 IPSEC VPN

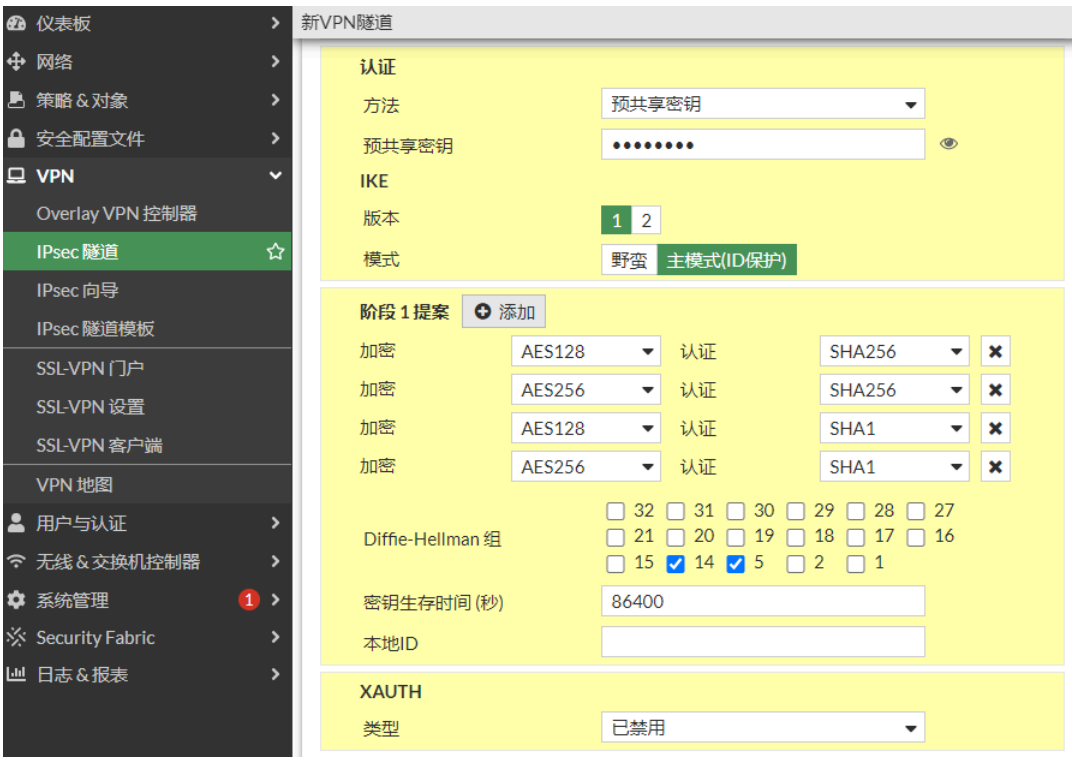
创建 IPSEC VPN。



IPSEC VPN 阶段一配置。

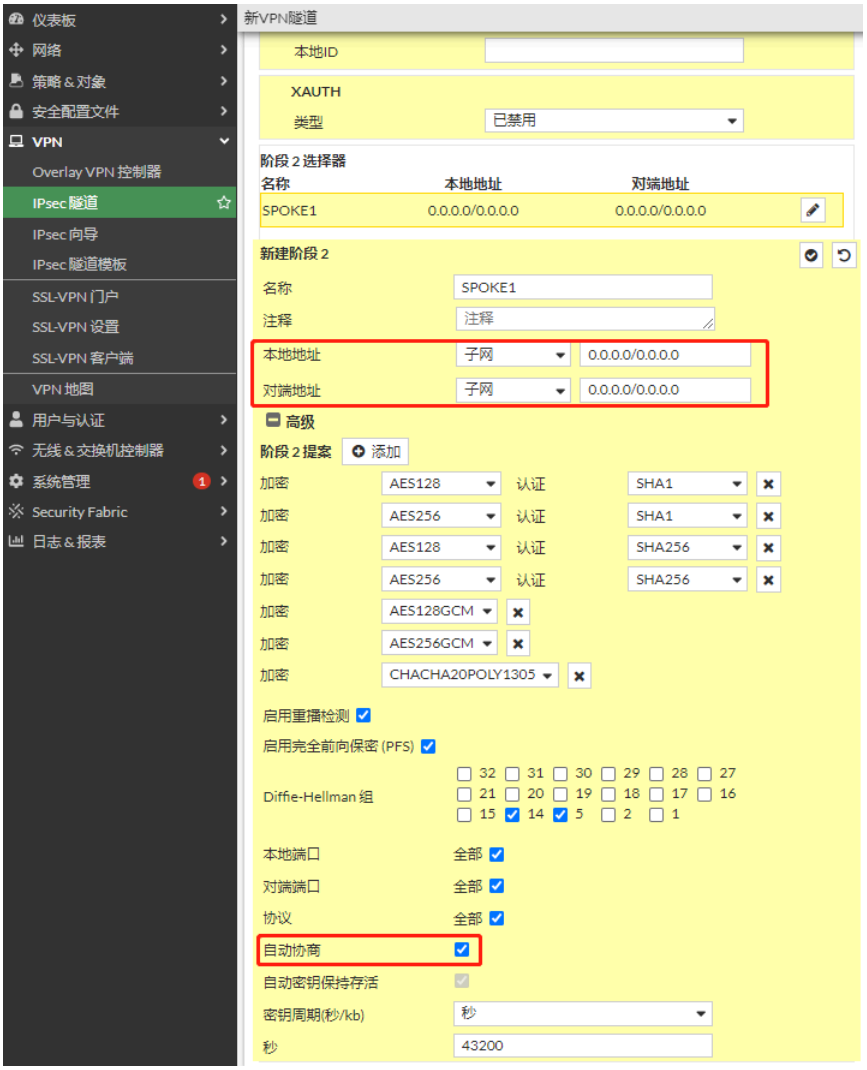
由于 HUB 和 SPOKE 之间是通过动态路由学习，因此 Exchange 接口 IP 打开，需要交换 HUB 和 SPOKE 之间 VPN 接口的 IP；SPOKE1 是静态 IPSEC，所以“添加路由”选项对 SPOKE1 没有意义，不用关注。





IPSEC VPN 阶段二配置。

第二阶段感兴趣流配置为 0.0.0.0/0.0.0.0，后续通过动态路由将流量指向 VPN 隧道。



VPN 创建完成。

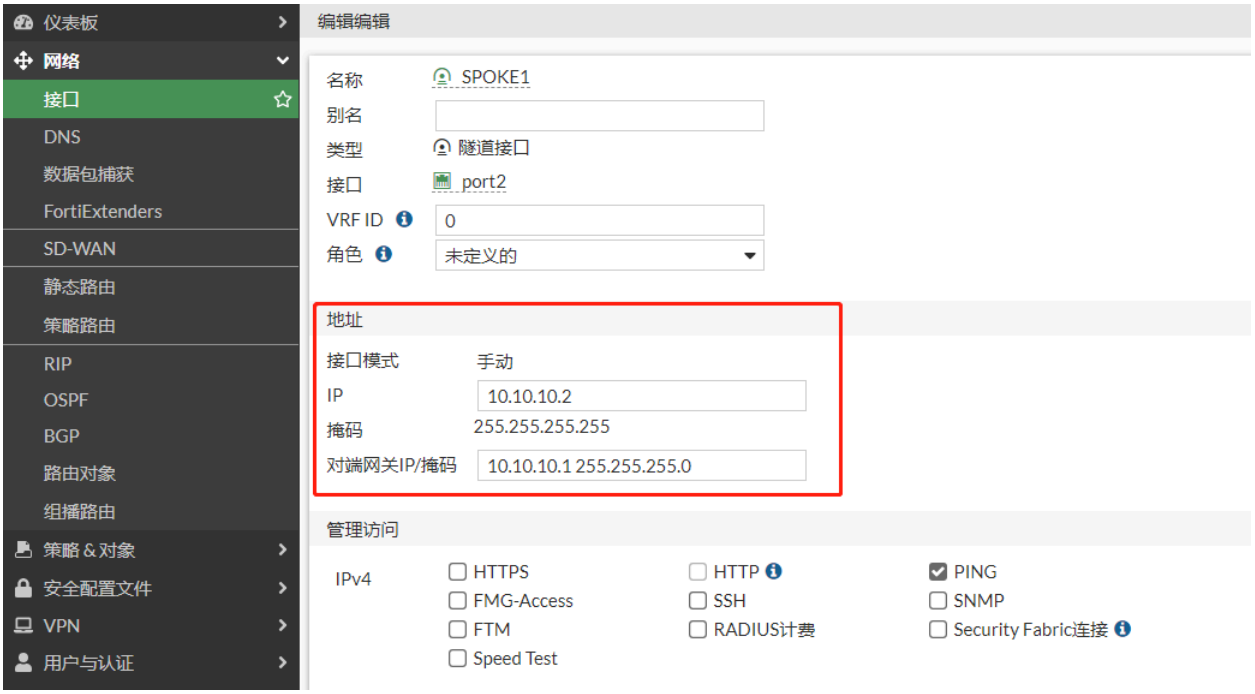


对应的命令行

```
config vpn ipsec phase1-interface
    edit "SPOKE1"
        set interface "port2"
        set peertype any
        set net-device disable
        set exchange-interface-ip enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 100.1.1.2
        set psksecret ENC
SasMccP3qWqglxEV1I2Z8ESNcmdXyPqkxyIoMpEM3haEKdLyi6qcOoc9WtdukEH2zXuV/bcVW6HyYqi8rh+KZ533z7cH6zqNpXCrOicxBYDXrEPVU6NXtp4UgL
+bqiG6Vm1FzA62V+YN4Rfchkuk49CuipVrer1xFqqIgXQVxM5GVuq+5wFvtWf5VBeuFbw+GHJlPQ==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE1"
        set phase1name "SPOKE1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

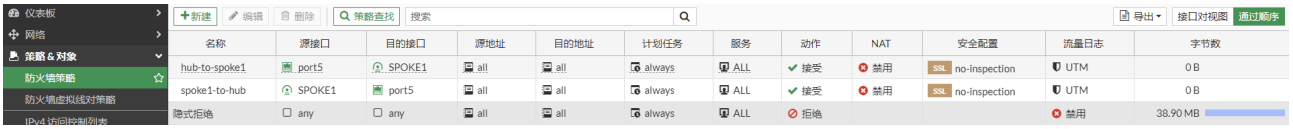
3.配置 VPN 接口 IP

SPOKE1 的“对端网关 IP”是指向 HUB 的。



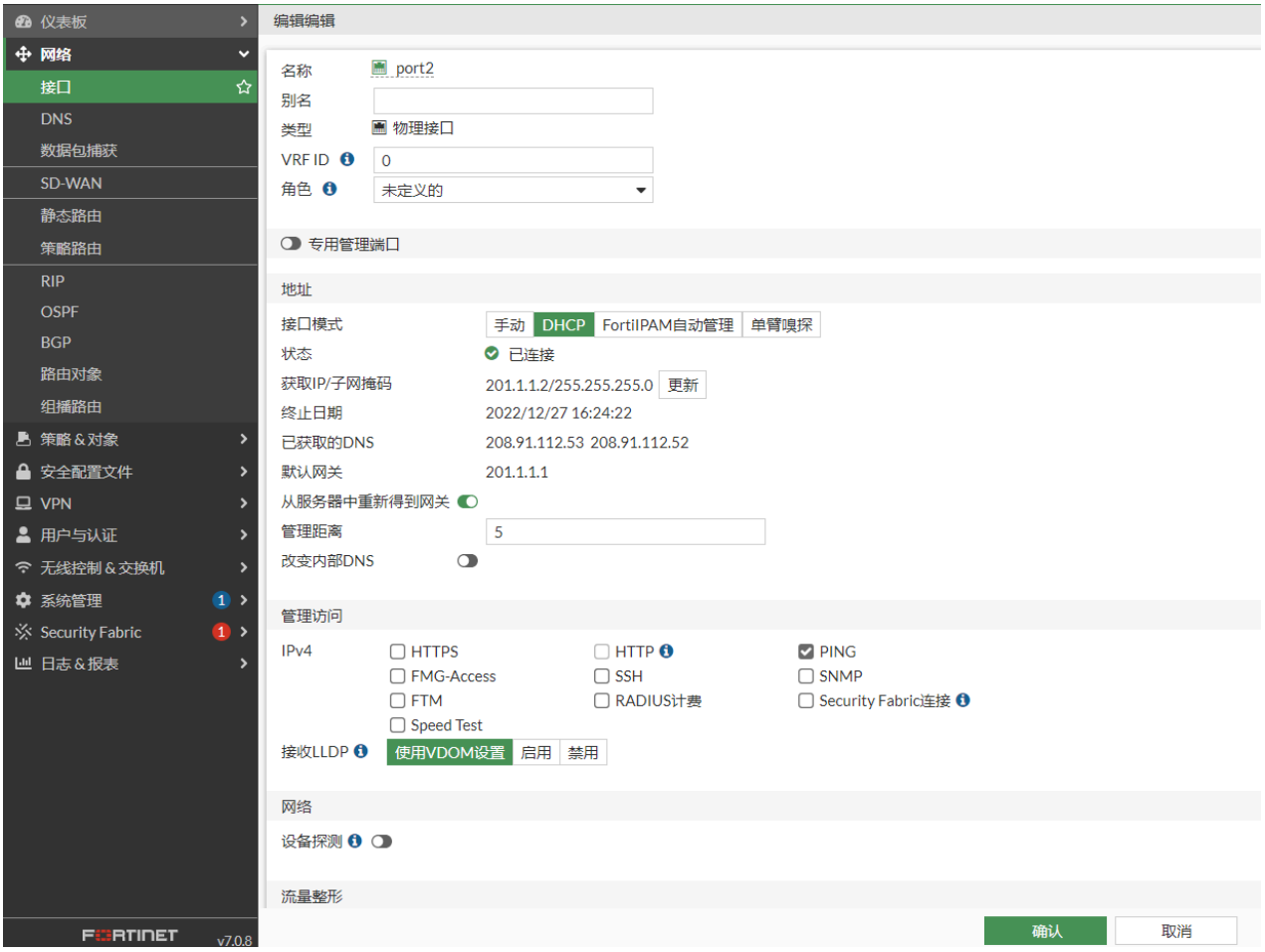
4.配置策略

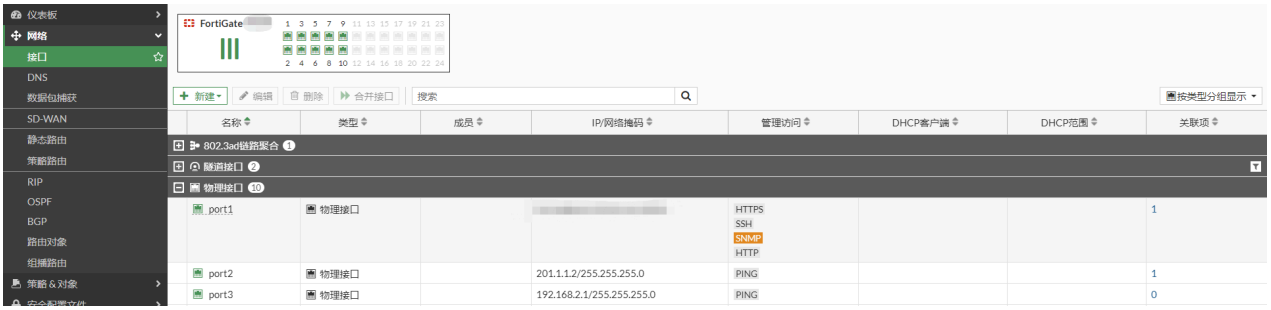
由于 SPOKE1 开启了自动连接，当配置策略后，就会主动与 HUB 协商，策略中的接口已经是绿色的，说明 VPN 接口已经 UP。



SPOKE2 配置

1. 基本配置





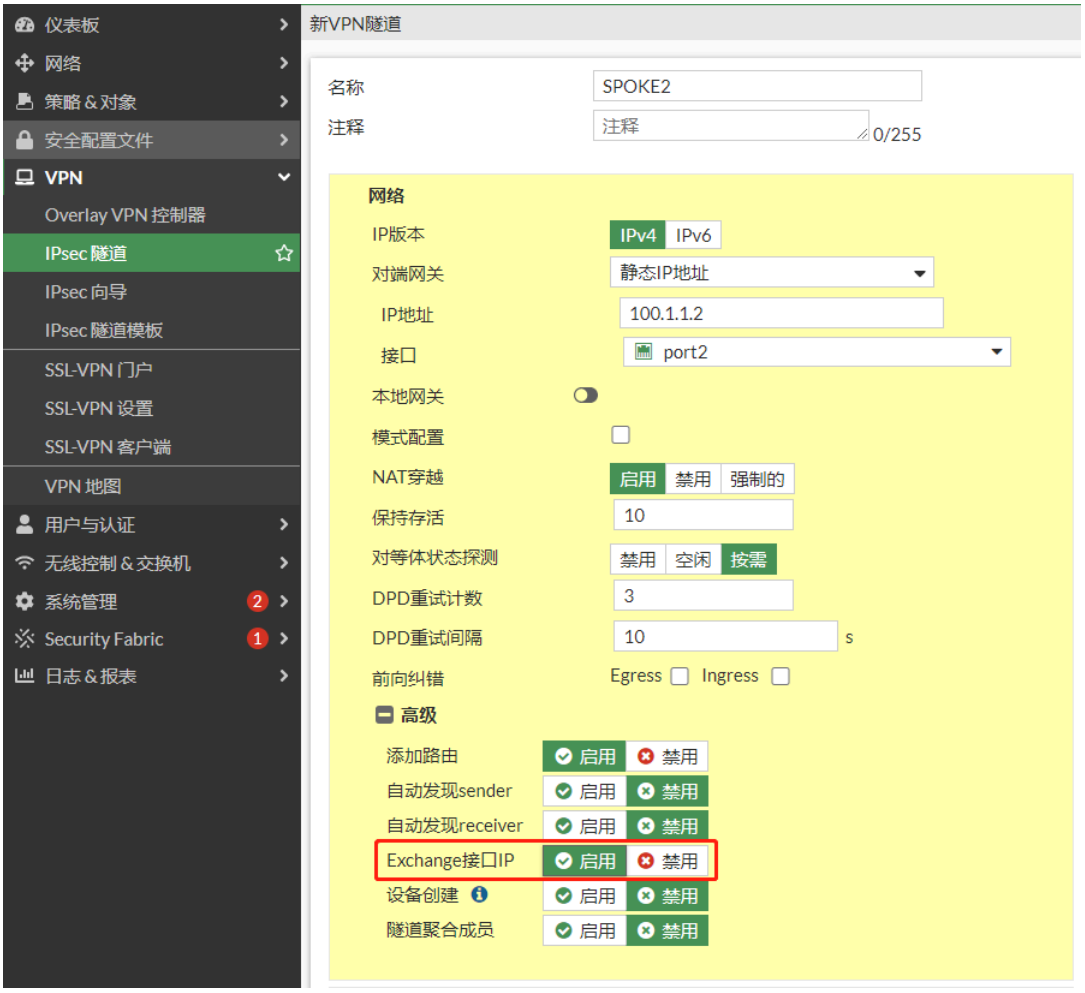
2.配置 IPSEC VPN

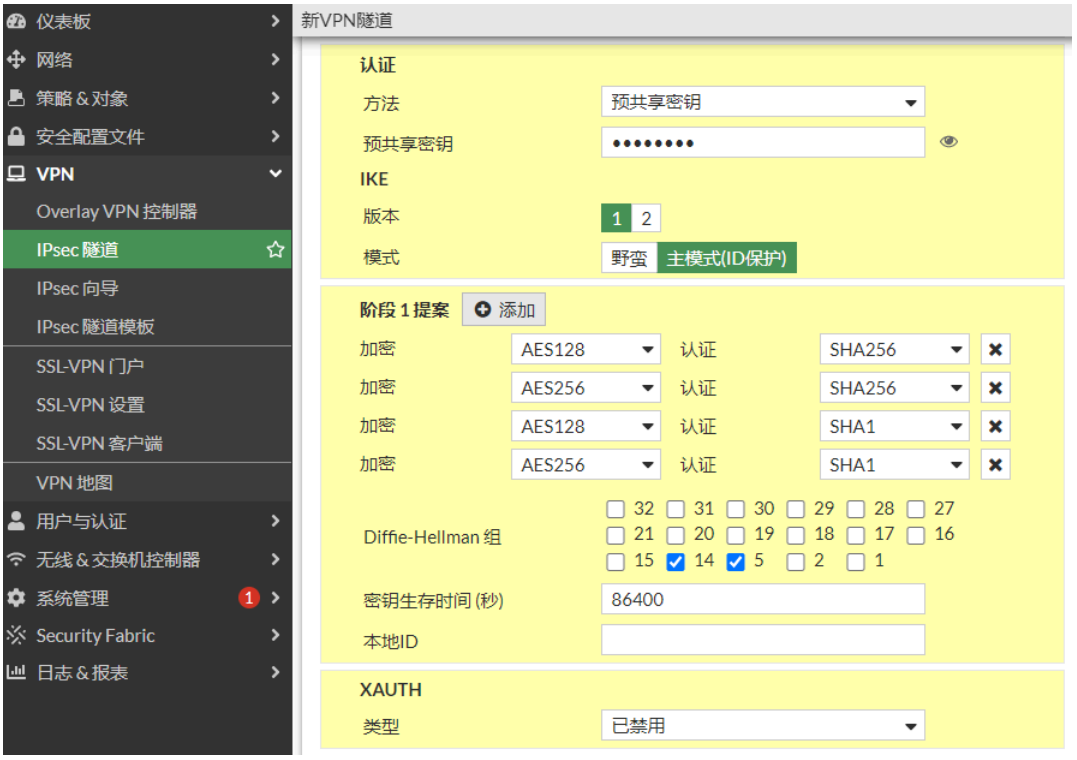
创建 IPSEC VPN。



IPSEC VPN 阶段一配置。

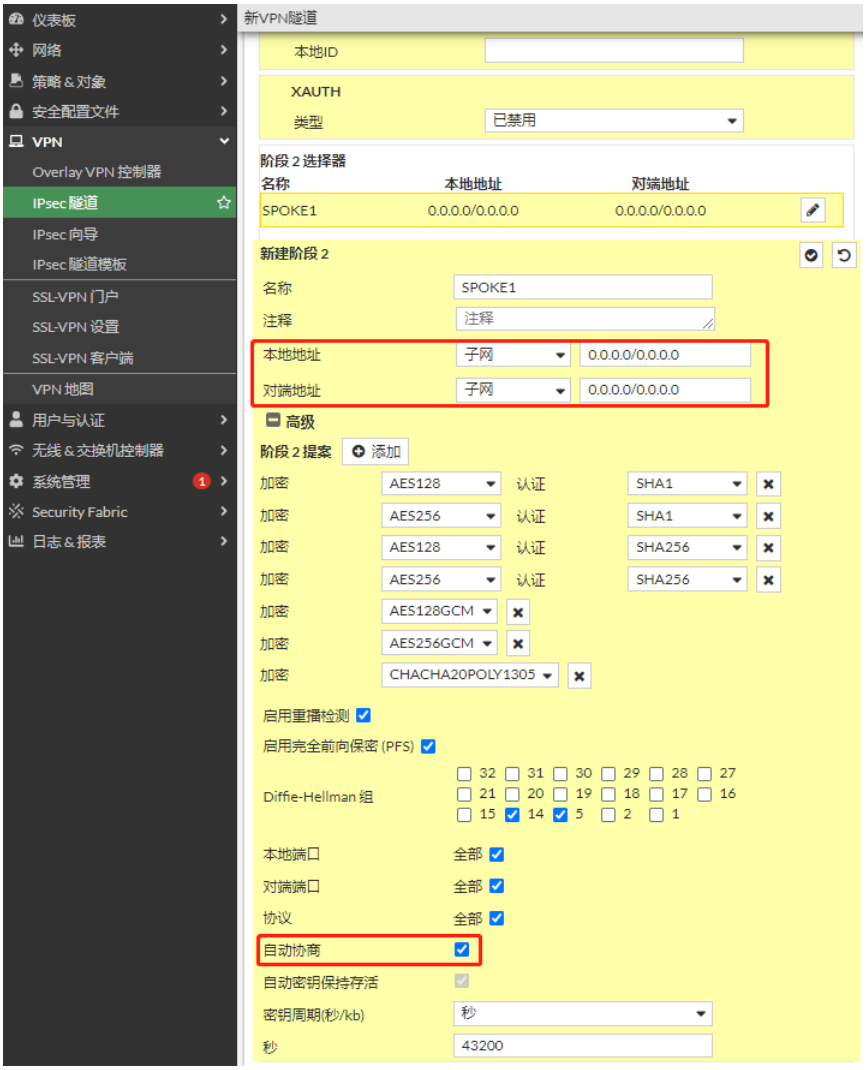
由于 HUB 和 SPOKE 之间是通过动态路由学习，因此 Exchange 接口 IP 打开，需要交换 HUB 和 SPOKE 之间 VPN 接口的 IP；SPOKE2 是静态 IPSEC，所以“添加路由”选项对 SPOKE2 没有意义，不用关注。





IPSEC VPN 阶段二配置。

第二阶段感兴趣流配置为 0.0.0.0/0.0.0.0，后续通过动态路由将流量指向 VPN 隧道。



VPN 创建完成。

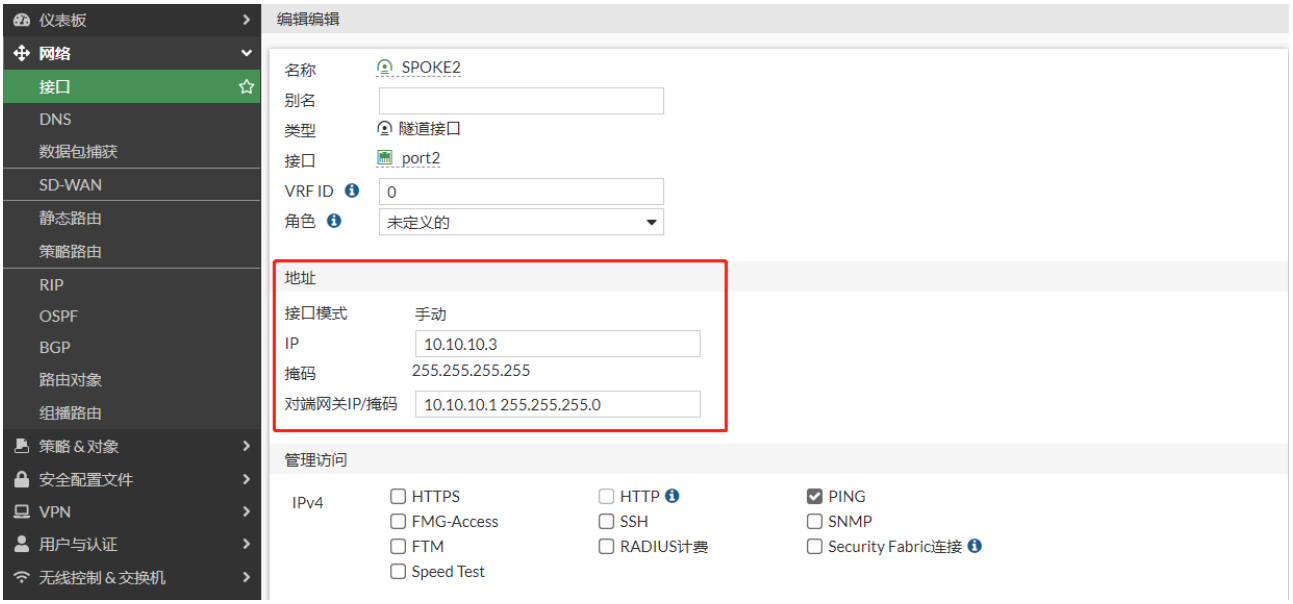


对应的命令行

```
config vpn ipsec phase1-interface
    edit "SPOKE2"
        set interface "port2"
        set peertype any
        set net-device disable
        set exchange-interface-ip enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 100.1.1.2
        set psksecret ENC
H3nAayIN5AWs3FomZpZri+tl6IAqg3iVu66h3qNXTjw1oSkiHauyAa/IUNw4xeJxD5dmiLfcAd/zzDb6BVai19ekg0Qvejqy39Mxr9UaRniPKE4dvoaoVLWy4O
vamxRzf82+JAUYKlF0NvVlMQ/+0JUYOunFeUlLLtht4dW69X9VTXfEBdwav496AzfYgI4gRTFFSg==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE2"
        set phase1name "SPOKE2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

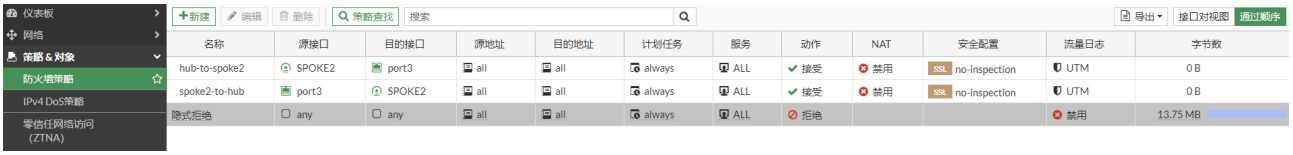
3.配置 VPN 接口 IP

SPOKE2 的“对端网关 IP”是指向 HUB 的。



4.配置策略

由于 SPOKE2 开启了自动连接，当配置策略后，就会主动与 HUB 协商，策略中的接口已经是绿色的，说明 VPN 接口已经 UP。



查看 VPN 状态和路由表

1.HUB 端 VPN 状态和路由表




```
# diagnose vpn ike gateway  list

vd: root/0
name: HUB_0
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::10.0.0.4
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.1 → 0.0.0.0
created: 33s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 b114b317daa486b7/29ede80ef4398500
  direction: responder
  status: established 33-33s ago = 0ms
  proposal: aes128-sha256
  key: 770b2cdeba34d096-8edcf542d0c07ef5
  lifetime/rekey: 86400/86096
  DPD sent/recv: 00000004/00000000
```

```
vd: root/0
name: HUB_1
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 201.1.1.2:500
tun_id: 201.1.1.2/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.1 → 0.0.0.0
created: 22s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 3 a0d49273057cb2d8/146f94f98305de8d
  direction: responder
  status: established 22-22s ago = 0ms
  proposal: aes128-sha256
  key: 4dc2b68b5795fd1c-0a400a810f439a19
  lifetime/rekey: 86400/86107
  DPD sent/recv: 00000000/00000008

FGT-wangxiang-used # diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=HUB_0 ver=1 serial=4 100.1.1.2:0→200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::10.0.0.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=5 olast=5 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=226 type=00 soft=0 mtu=1438 expire=43155/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43191/43200
  dec: spi=d87e91b4 esp=aes key=16 f598636b2f6feb532e32f883300f4595
       ah=sha1 key=20 663bcb60b48c072daebdc011e27c35836e2f45eb
  enc: spi=d56f5619 esp=aes key=16 f3059875bf7c1915eb711a5355a603a8
       ah=sha1 key=20 f8d348a6abdb169781f2e1b3aae099c70d16ea92
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=2 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=HUB_1 ver=1 serial=5 100.1.1.2:0→201.1.1.2:0 tun_id=201.1.1.2 tun_id6=::10.0.0.5 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=4 olast=4 ad=/0
stat: rxp=1 txp=0 rxb=60 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=2
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=4 options=226 type=00 soft=0 mtu=1438 expire=43162/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43187/43200
  dec: spi=d87e91b5 esp=aes key=16 f7229ae8b4fe965ea7346f41ffae201a
       ah=sha1 key=20 8ada15a3ef8a71fdb3037ee1f7014ca930009cdb
  enc: spi=c7b86106 esp=aes key=16 a6346a22cf40fba236f0cfaacc720261
       ah=sha1 key=20 e093aed38364b32892da5ae4f99413be54a7c635
  dec:pkts/bytes=2/120, enc:pkts/bytes=0/0
  npu_flag=02 npu_rgwy=201.1.1.2 npu_lgwy=100.1.1.2 npu_selid=3 dec_npuid=1 enc_npuid=0
------------------------------------------------------
name=HUB ver=1 serial=1 100.1.1.2:0→0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1 dst_mtu=0 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
```

```
proxyid_num=0 child_num=2 refcnt=4 ilast=42949600 olast=42949600 ad=/0
stat: rxp=3 txp=0 rxb=180 txb=0
dpd: mode=on-idle on=0 idle=10000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

查看路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       10.10.10.0/24 is directly connected, HUB
C       10.10.10.1/32 is directly connected, HUB
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
```

从 HUB 端分别 ping SPOKE1 和 SPOKE2 的隧道接口地址。

```
# execute ping-options source 10.10.10.1
# execute ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=255 time=0.7 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=255 time=0.4 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=255 time=0.3 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=255 time=0.3 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=255 time=0.4 ms

--- 10.10.10.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.7 ms

# execute ping 10.10.10.3
PING 10.10.10.3 (10.10.10.3): 56 data bytes
64 bytes from 10.10.10.3: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 10.10.10.3: icmp_seq=1 ttl=255 time=0.4 ms
64 bytes from 10.10.10.3: icmp_seq=2 ttl=255 time=0.4 ms
64 bytes from 10.10.10.3: icmp_seq=3 ttl=255 time=0.5 ms
64 bytes from 10.10.10.3: icmp_seq=4 ttl=255 time=0.2 ms

--- 10.10.10.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.3/0.5 ms
```

2.SPOKE1 端 VPN 状态




```
# diagnose vpn  ike gateway  list

vd: root/0
name: SPOKE1
version: 1
interface: port2 10
addr: 200.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.2 → 10.10.10.1
created: 139s ago
```

```
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 0 33b5335c7174437e/c89906c6196f3cd7
  direction: initiator
  status: established 139-139s ago = 0ms
  proposal: aes128-sha256
  key: ad3e41d87eb3b8b8-c97c621b25fe9a48
  lifetime/rekey: 86400/85960
  DPD sent/recv: 00000004/0000000a

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE1 ver=1 serial=1 200.1.1.2:0->100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=12 olast=12 ad=/0
stat: rxp=0 txp=6 rxb=0 txb=360
dpd: mode=on-demand on=1 idle=10000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE1 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=5 options=18227 type=00 soft=0 mtu=1438 expire=42762/0B replaywin=2048
       seqno=7 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=1a45fdb4 esp=aes key=16 373c729e4164750638f8ff0d37dbdd12
       ah=sha1 key=20 323ffd35fb089657505202639ad92371df146881
  enc: spi=a7d1cc77 esp=aes key=16 6da00eca8e15572377040c982e206526
       ah=sha1 key=20 3ea0959e65417277e75bdc5f52bdbc768f33d9b0
  dec:pkts/bytes=0/0, enc:pkts/bytes=12/1080
  npu_flag=01 npu_rgwy=100.1.1.2 npu_lgwy=200.1.1.2 npu_selid=0 dec_npuid=0 enc_npuid=1
run_tally=0
```

查看路由表

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
C       10.10.10.2/32 is directly connected, SPOKE1
C       192.168.1.0/24 is directly connected, port5
C       200.1.1.0/24 is directly connected, port2
```

从 SPOKE1 端 ping HUB 的隧道地址。

```
# execute ping-options source 10.10.10.2
# execute ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1): 56 data bytes
64 bytes from 10.10.10.1: icmp_seq=0 ttl=255 time=0.7 ms
64 bytes from 10.10.10.1: icmp_seq=1 ttl=255 time=0.4 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=255 time=0.6 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=255 time=0.4 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=255 time=0.6 ms

--- 10.10.10.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.5/0.7 ms
```

3.SPOKE2 端 VPN 状态




```
# diagnose vpn ike gateway list

vd: root/0
name: SPOKE2
version: 1
interface: port2 6
addr: 201.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.3 → 10.10.10.1
created: 166s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 0 712a042d8d25877d/437eb372be68f67e
  direction: initiator
  status: established 166-166s ago = 10ms
  proposal: aes128-sha256
  key: 48fb4b35b9a67aa3-f2f62d8949523c2d
  lifetime/rekey: 86400/85933
  DPD sent/recv: 0000000f/00000004

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE2 ver=1 serial=1 201.1.1.2:0→100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=8 olast=8 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=15
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=18203 type=00 soft=0 mtu=1438 expire=42730/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42899/43200
  dec: spi=99b97fa0 esp=aes key=16 6da0cd14244e96c9224fdfc526104c88
       ah=sha1 key=20 c15decb70b3de9678d1313b564ec2e6e912b9540
  enc: spi=a7d1cc78 esp=aes key=16 023a05cc8cd9224c781ca640b044c5b3
       ah=sha1 key=20 96da3fa1c8207a267200aac94db3718944574a1b
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=201.1.1.2 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

查看路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 201.1.1.1, port2, [1/0]
```

```
S       10.10.10.0/24 [5/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
C       10.10.10.3/32 is directly connected, SPOKE2
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
```

从 SPOKE2 端 ping HUB 隧道接口地址。

```
# execute ping-options source 10.10.10.3
# execute ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1): 56 data bytes
64 bytes from 10.10.10.1: icmp_seq=0 ttl=255 time=0.5 ms
64 bytes from 10.10.10.1: icmp_seq=1 ttl=255 time=0.6 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=255 time=0.6 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=255 time=0.6 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=255 time=0.4 ms

--- 10.10.10.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.5/0.6 ms
```

配置动态路由

1.HUB 端 BGP 配置

利用 BGP neighbor-group 特性，只要匹配前缀列表 10.10.10.0/24 的 BGP-Peer 均可以和 HUB 建立 BGP 邻居。

将邻居设置为 Route-Reflector-Client，自己即为路由反射器（Route-Reflector）,通过路由反射器原理实时同步整网路由。

```
config router bgp
    set as 65001
    set router-id 100.1.1.2
    config neighbor-group
        edit "group1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set interface "HUB"
            set remote-as 65001
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.10.0 255.255.255.0
            set neighbor-group "group1"
        next
    end
    config network
        edit 1
            set prefix 192.168.0.0 255.255.255.0
        next
    end
end
```

2.SPOKE1 端 BGP 配置

与 HUB 端建立 BGP 邻居

```
config router bgp
    set as 65001
    set router-id 200.1.1.2
    config neighbor
        edit "10.10.10.1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set interface "SPOKE1"
            set remote-as 65001
        next
    end
    config network
```

```
        edit 1
            set prefix 192.168.1.0 255.255.255.0
        next
    end
end
```

3.SPOKE2 端 BGP 配置

与 HUB 端建立 BGP 邻居

```
config router bgp
    set as 65001
    set router-id 201.1.1.2
    config neighbor
        edit "10.10.10.1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set interface "SPOKE2"
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 192.168.2.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 ::/128
        next
    end
end
```

查看路由表

1.HUB 端路由表

```
FGT-wangxiang-used # get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       10.10.10.0/24 is directly connected, HUB
C       10.10.10.1/32 is directly connected, HUB
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
B       192.168.1.0/24 [200/0] via 10.10.10.2 (recursive is directly connected, HUB), 00:21:21
B       192.168.2.0/24 [200/0] via 10.10.10.3 (recursive is directly connected, HUB), 00:21:21
```

2.SPOKE1 端路由表

```
FGT-wangxiang-used # get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
```

```
C       10.10.10.2/32 is directly connected, SPOKE1
C       200.1.1.0/24 is directly connected, port2
C       192.168.1.0/24 is directly connected, port5
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE1 tunnel 100.1.1.2), 00:21:41
B       192.168.2.0/24 [200/0] via 10.10.10.3 (recursive via SPOKE1 tunnel 100.1.1.2), 00:21:18
```

3.SPOKE2 端路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
C       10.10.10.3/32 is directly connected, SPOKE2
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE2 tunnel 100.1.1.2), 00:22:27
B       192.168.1.0/24 [200/0] via 10.10.10.2 (recursive via SPOKE2 tunnel 100.1.1.2), 00:22:27
```

业务测试

1.HUB 端业务测试

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6140132  bytes 1760274153 (1.6 GiB)
        RX errors 0  dropped 3749603  overruns 0  frame 0
        TX packets 51686  bytes 3833219 (3.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问SPOKE1
PC1# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=62 time=1.42 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=62 time=0.788 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=62 time=0.751 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=62 time=0.688 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.688/0.913/1.427/0.300 ms

访问SPOKE2
PC1# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=62 time=1.09 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.935 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.991 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=1.06 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.935/1.021/1.098/0.070 ms
```

2.SPOKE1 端业务测试

```
PC2# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::1a6c:e61:d2b9:a415  prefixlen 64  scopeid 0×20<link>
        inet6 2001::2  prefixlen 64  scopeid 0×0<global>
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 5904889  bytes 459674552 (438.3 MiB)
        RX errors 0  dropped 58  overruns 0  frame 0
        TX packets 3224540  bytes 205411564402 (191.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问HUB
PC2# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.746 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.765 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.862 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=0.677 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.677/0.762/0.862/0.071 ms

访问SPOKE2
PC2# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=61 time=1.14 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=61 time=0.952 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=61 time=1.51 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.886 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.886/1.124/1.519/0.248 ms
```

3.SPOKE2 端业务测试

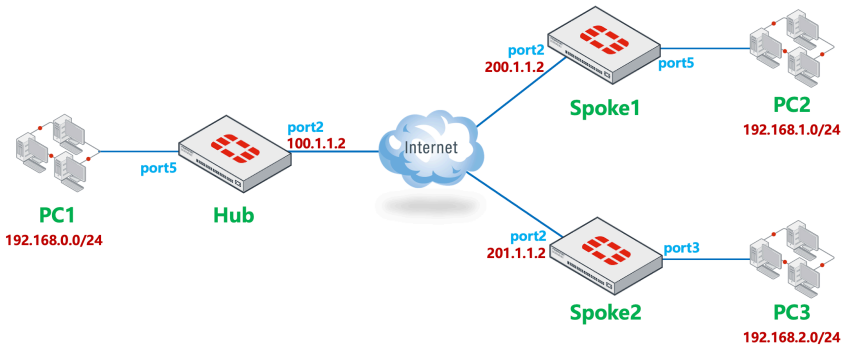
```
PC3# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::2652:4dd7:5d0e:941d  prefixlen 64  scopeid 0×20<link>
        inet6 240e:604:109:39::216  prefixlen 64  scopeid 0×0<global>
        ether 00:0c:29:37:f0:ac  txqueuelen 1000  (Ethernet)
        RX packets 9638867  bytes 205844622412 (191.7 GiB)
        RX errors 0  dropped 118  overruns 0  frame 0
        TX packets 5737914  bytes 378901312 (361.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

访问HUB
PC3# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.848 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.935 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.899 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=1.04 ms
--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.848/0.931/1.045/0.081 ms

访问SPOKE1
PC3# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=61 time=1.13 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=61 time=1.22 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=61 time=1.21 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=61 time=1.04 ms
--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.043/1.151/1.220/0.075 ms
```

ADVPN 模式

网络需求

传统的 Hub-Spoke 方式中，Spoke 只能和 Hub 建立永久隧道，Spoke 之间的流量需要通过 Hub 来转发，这种方式减轻了 Spoke 的负担，增加了 Hub 的性能要求，同时利于总部对分支间流量的监控；使用 ADVPN 技术实现的 Full-Mesh 方式中，Spoke 之间可以建立动态直连隧道，分支间的流量可以直接转发。相比而言，Hub 负担减轻，同时减少分支间流量的延迟，更有利于 SPOKE 之间的流量传输，在实际使用的过程中可按照自身需求进行选择。

网络拓扑



VPN Tunnel IP 地址分配，以及 BGP 的规划：

角色	公网 IP	私网网段	VPN 隧道 IP	BGP 信息
HUB	100.1.1.2	192.168.0.1/24	10.10.10.1	AS 65001 RR 反射器
SPOKE1	200.1.1.2	192.168.1.1/24	10.10.10.2	AS 65001 RR Client
SPOKE2	201.1.1.2	192.168.2.1/24	10.10.10.3	AS 65001 RR Client
SPOKE3	...	...	...	...
SPOKEX	...	...	...	...

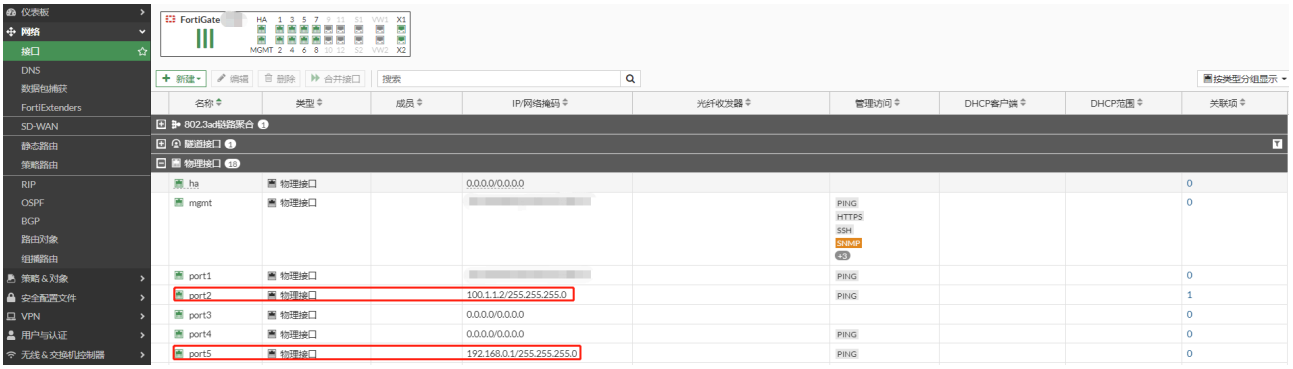
配置步骤

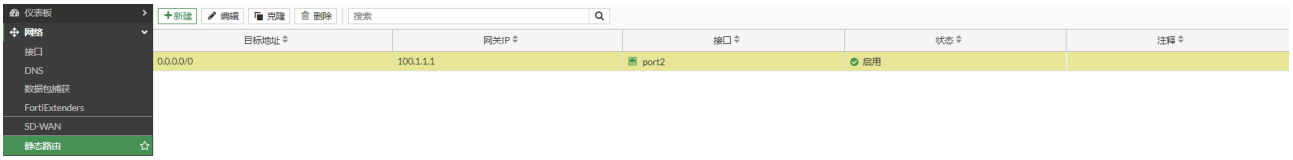
这里使用 IPSEC “Hub and Spoke”模板创建 ADVPN。

HUB 端配置

1.基本配置

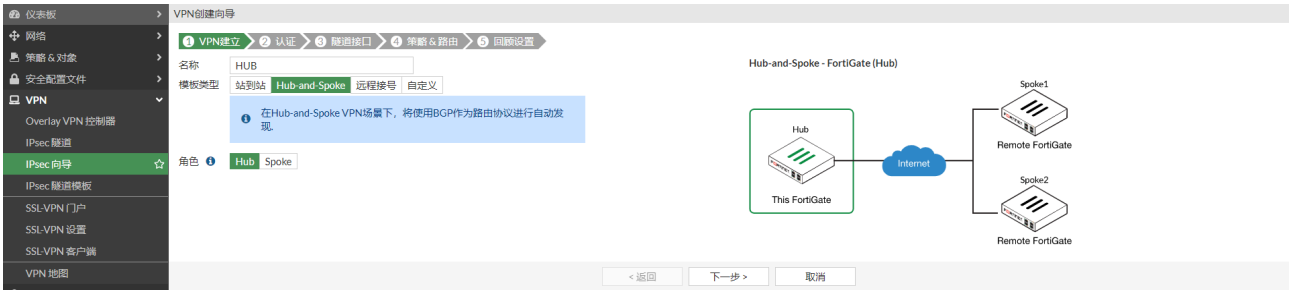
配置接口 IP 和路由



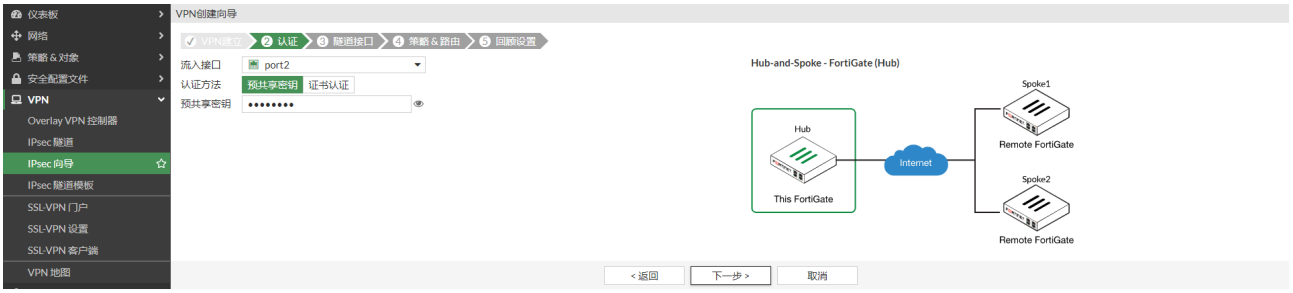


2.配置 IPSEC VPN

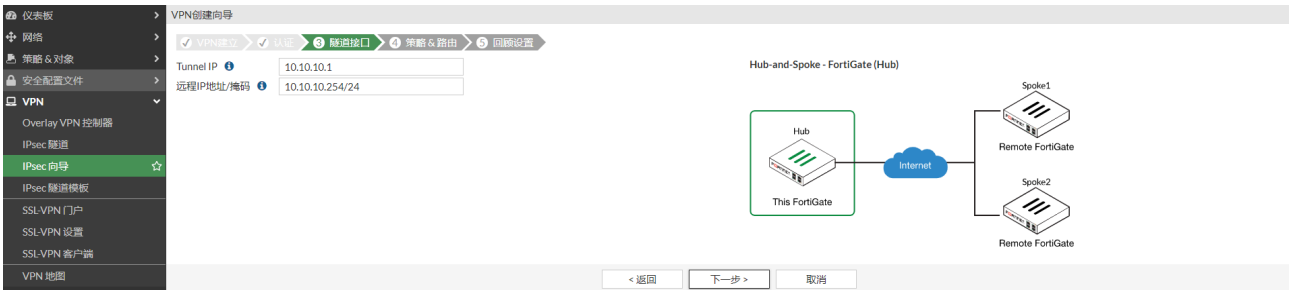
选择“Hub and Spoke”模板



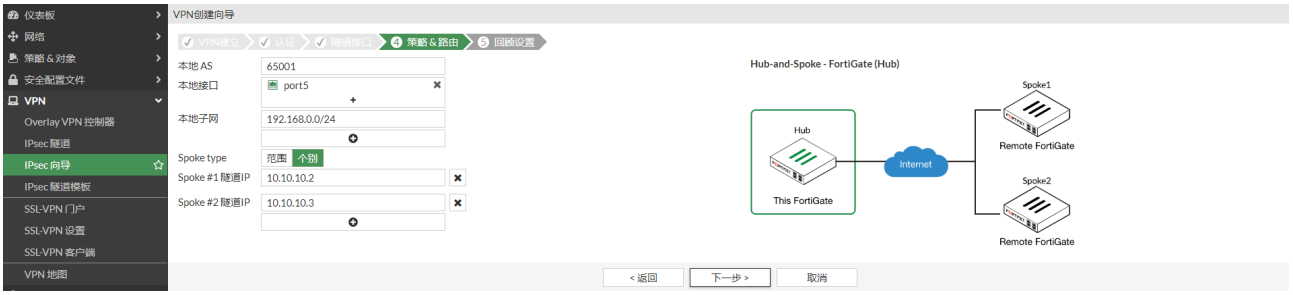
选择创建 IPSEC 的接口及预共享秘钥。



HUB 端的 tunnel ip 是 10.10.10.1，远程 IP 地址/掩码是 10.10.10.254/24。10.10.10.254 是不被 Spoke 所使用的预留 IP，IPsec Tunnel 是一个点对点的隧道，但是 ADVPN 中这条隧道需要同时对应多个 SPOKE，因此不能将 Remote IP 写成一个存在的 SPOKE 端 IP。



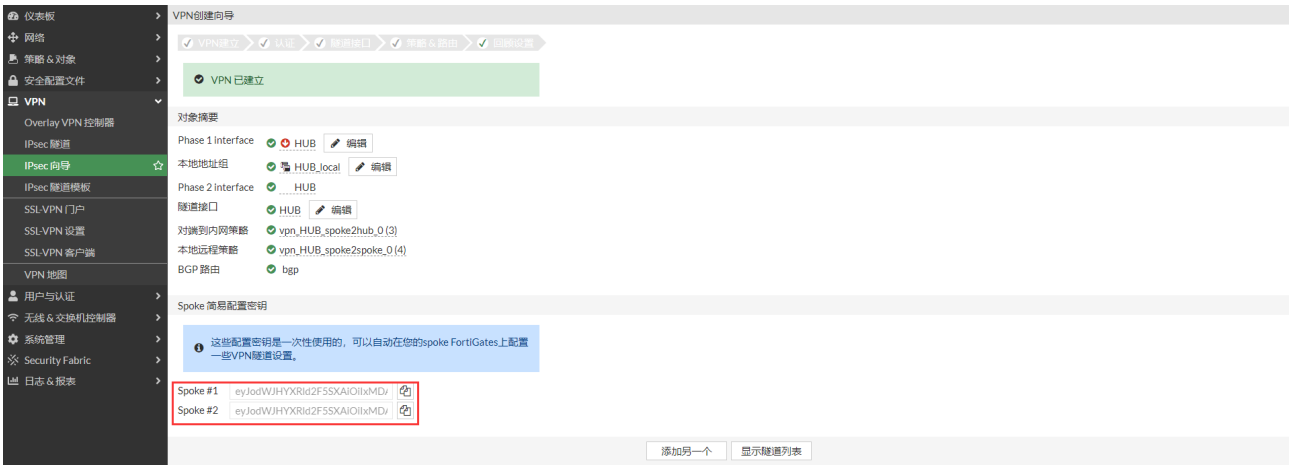
输入 BGP AS 号，本地的内网接口及本地的内网网段，这里有两个 spoke，分别设置对应隧道 IP。



VPN 向导即将创建的内容。



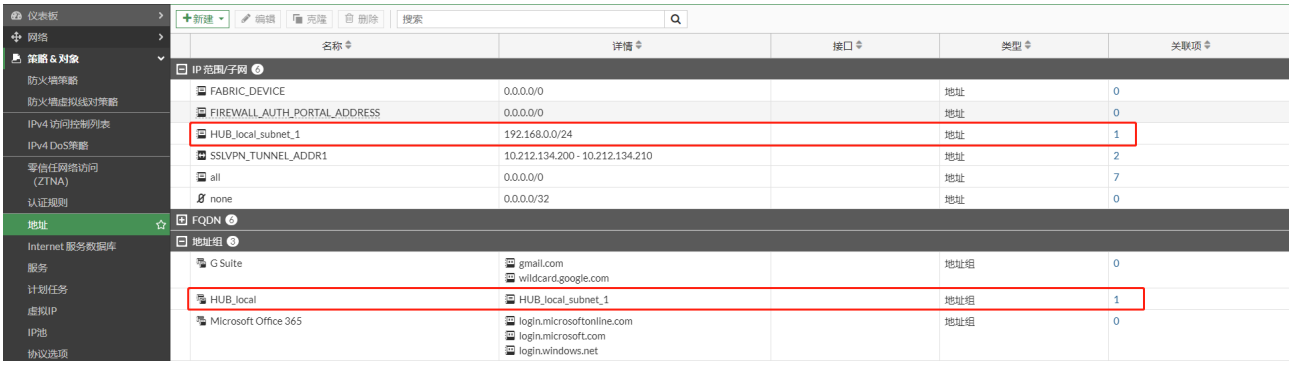
VPN 创建完成，Spoke#1 和 Spoke#2 需要复制下来用于 SPOKE 端创建 VPN。



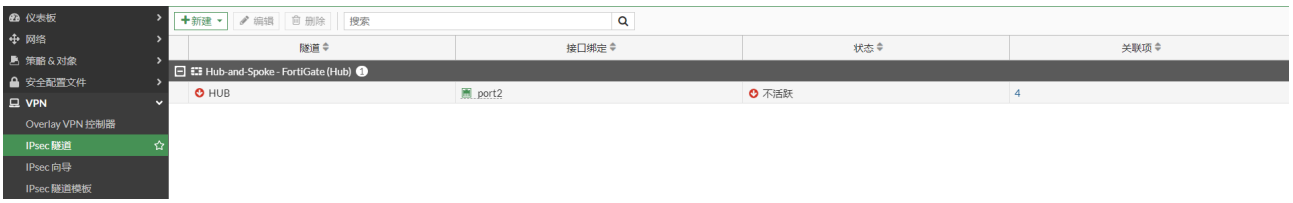
查看 HUB 端 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建地址对象和地址对象组



2.创建 IPSEC VPN



对应的命令行

```
config vpn ipsec phase1-interface
    edit "HUB"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
```

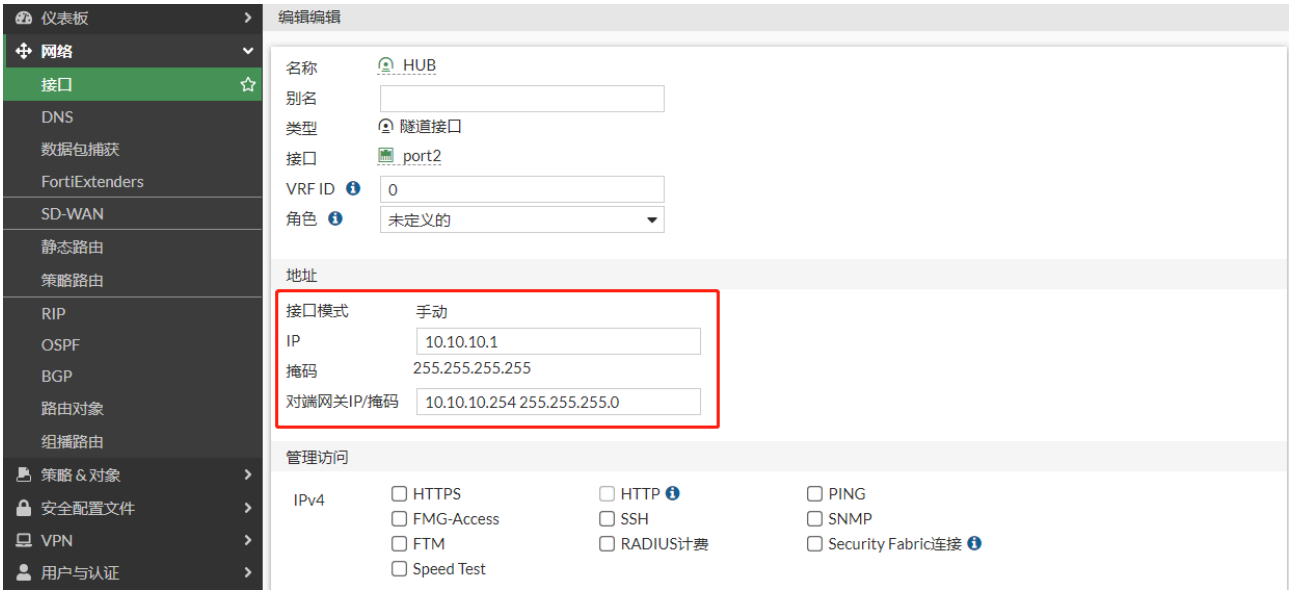
```
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: HUB (Created by VPN wizard)"
        set wizard-type hub-fortigate-auto-discovery
        set auto-discovery-sender enable
        set psksecret ENC
SKVqWAT1K2iJB8U5T26IVmwtHjbLPes7VloWfh1ARXPGobKTYHruFIVEe/RjNKZrNj3j1GdXrvp1GlPLt7DExzLGKKyIoQ/1q5owciyhpLn753JL0kRW86eR/C
2h0aDcGZVKI8U9MagbU1pEy7RC71rtRr3dJjSd95eu6oeyq9FssYpAq9jCdyDbiSpbZoEVoYEwiw==
    next
end
config vpn ipsec phase2-interface
    edit "HUB"
        set phase1name "HUB"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: HUB (Created by VPN wizard)"
    next
end
```

3.创建策略

> **重要**
>
> 模板中只创建了 SPOKE 到 HUB 端的策略以及 SPOKE 到 SPOKE 端的策略，没有创建 HUB 到 SPOKE 端的策略，可以根据业务的需求添加。



4.创建 VPN 接口 IP



5.创建 BGP



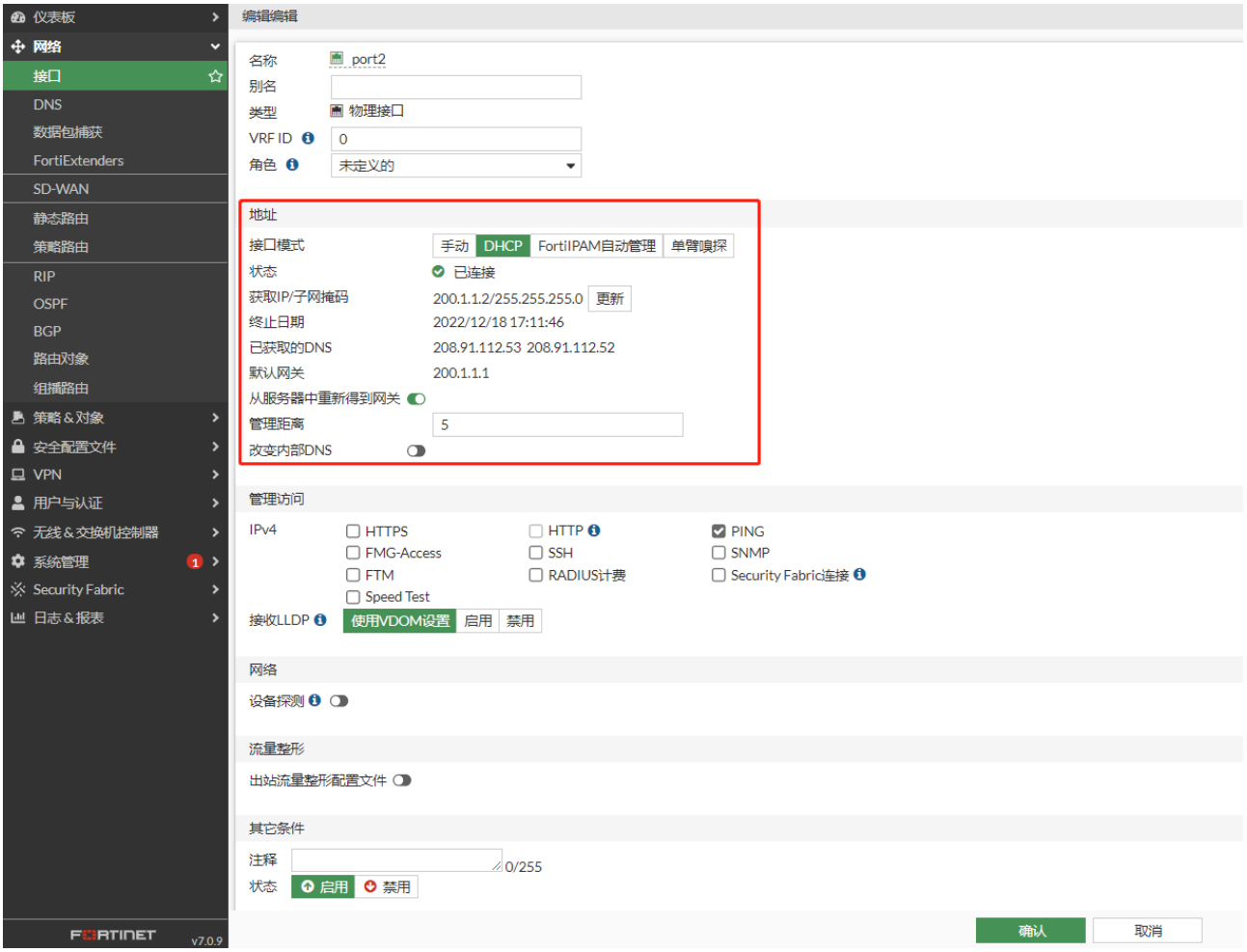
针对向导配置的优化建议

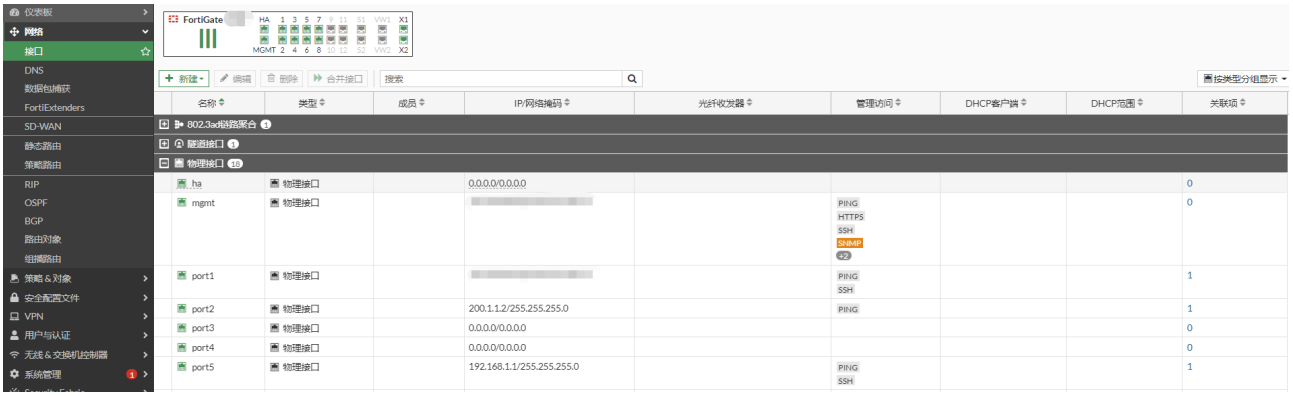
1. 在第一阶段中开启 DPD 周期性检测（每隔 10s 检测一次 Peer 状态），实现快速的检测并切换 VPN 隧道的目的。

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

SPOKE1 端配置

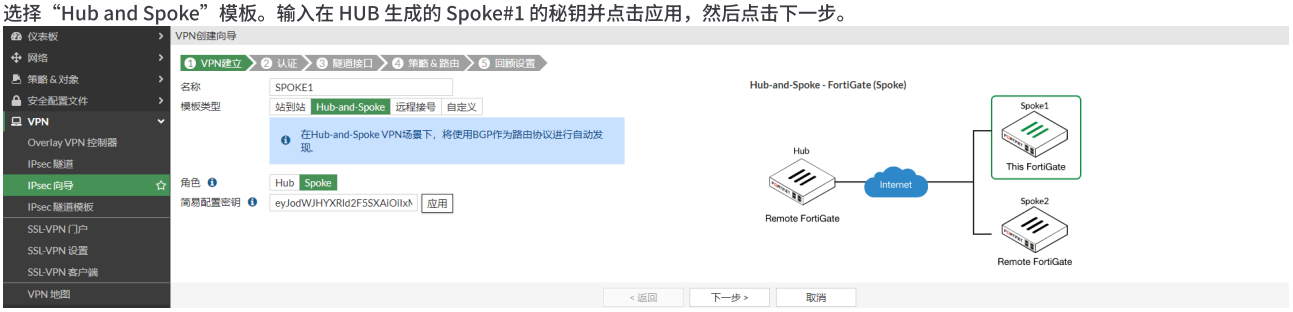
1.基本配置



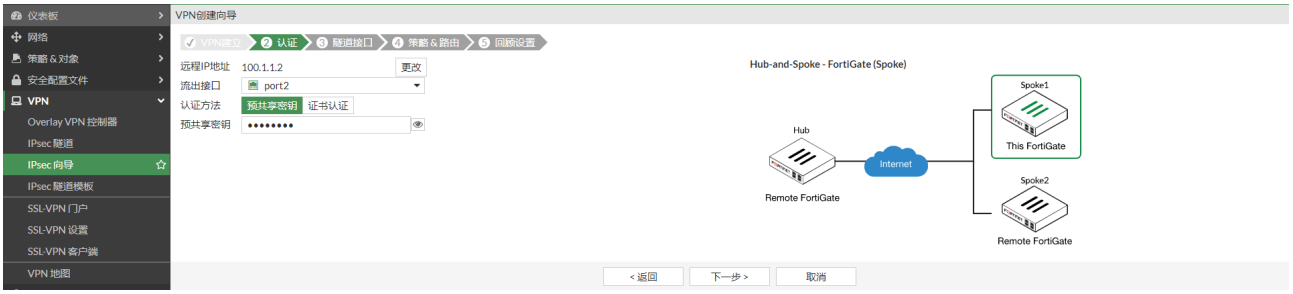


2.配置 IPSEC VPN

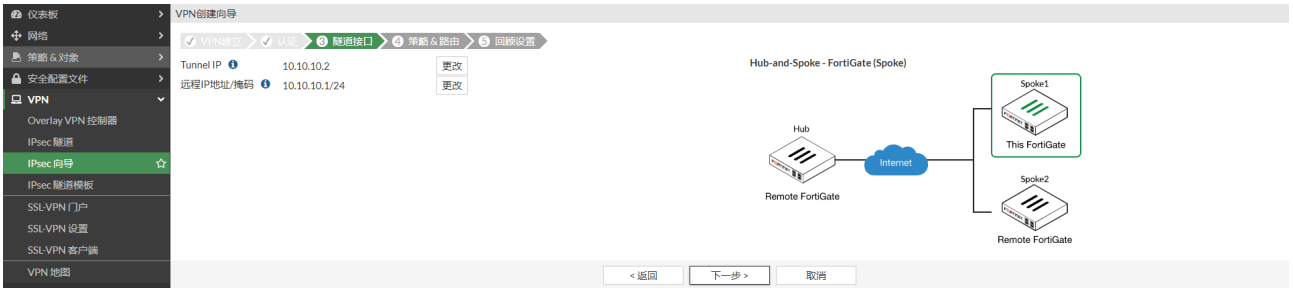
选择“Hub and Spoke”模板。输入在 HUB 生成的 Spoke#1 的秘钥并点击应用，然后点击下一步。



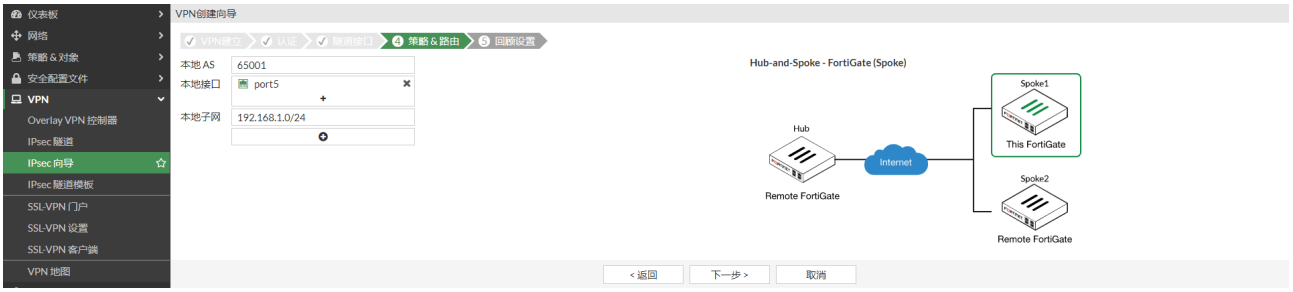
远程 IP 地址和流出接口会自动生成，需要输入预共享秘钥。点击下一步。



自动生成隧道 IP，点击下一步。



自动生成本地 AS 号，需要生成 SPOKE1 本地的内网接口及需要保护的内网网段，点击下一步。



VPN 向导即将创建的内容，点击完成。



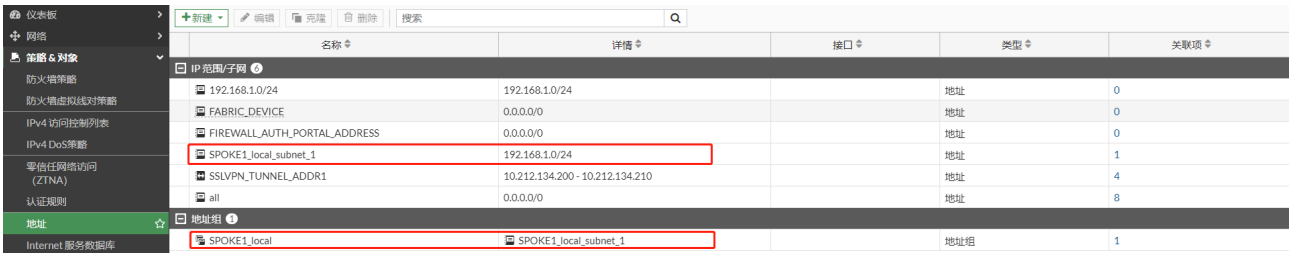
VPN 创建完成。



查看 SPOKE1 端 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建地址对象和地址对象组



2.创建 VPN



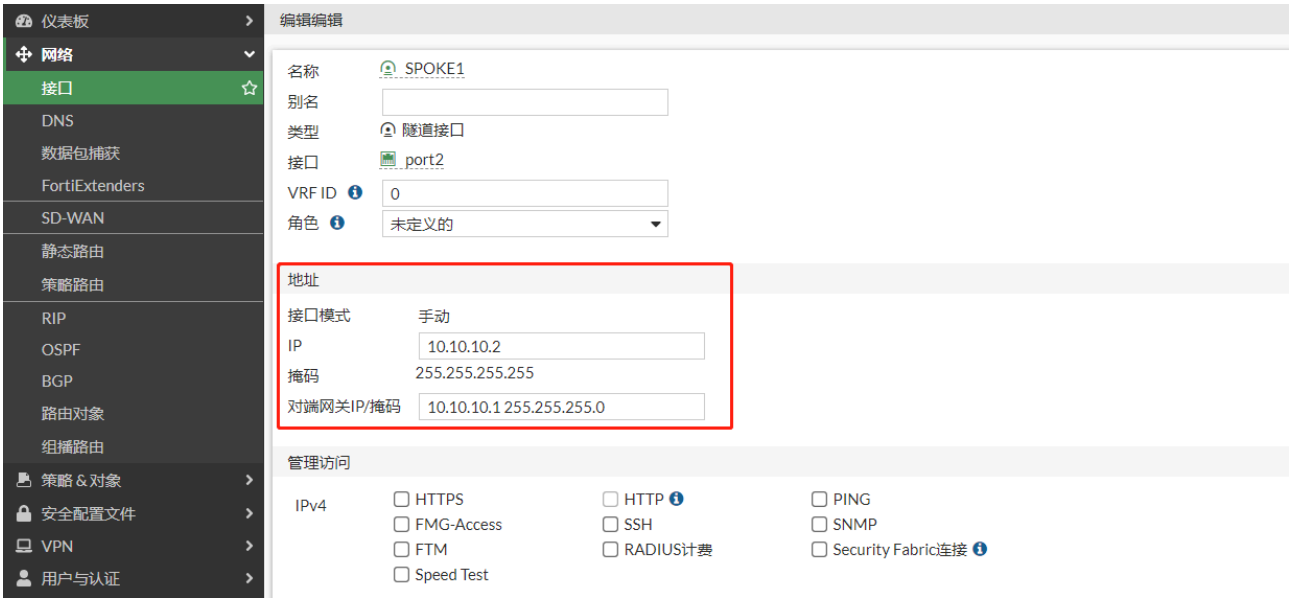
对应的命令行

```bash
config vpn ipsec phase1-interface
    edit "SPOKE1"
        set interface "port2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: SPOKE1 (Created by VPN wizard)"
        set wizard-type spoke-fortigate-auto-discovery
        set auto-discovery-receiver enable
        set remote-gw 100.1.1.2
        set psksecret ENC
zVXgULIXvCQPfG0ubZ8R36jhME7KanJ9V/NyV8zt5tXA5jlwPAli6alNN6g26Udtb04sWU/veKHuorIFmj9fO0J9vBi6Da6+pRlhnfSZ/f3fcxG7hj7ydbk72P
yXw2mDdDKsMty+27VVqTNG8OEyxdX+/vCG82iT5NjaLM84V/P1/YNUlIWqrnte2PEjuw9tzzbbDQ==
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE1"
        set phase1name "SPOKE1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: SPOKE1 (Created by VPN wizard)"
    next
end
```

3.创建策略略



4.创建 VPN 接口 IP



5.创建 BGP



针对向导配置的优化建议

1. 在第一阶段中开启 DPD 周期性检测（每隔 10s 检测一次 Peer 状态），实现快速的检测并切换 VPN 隧道的目的。

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

2. 开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

   IPSEC VPN 阶段一自动协商是默认开启的。

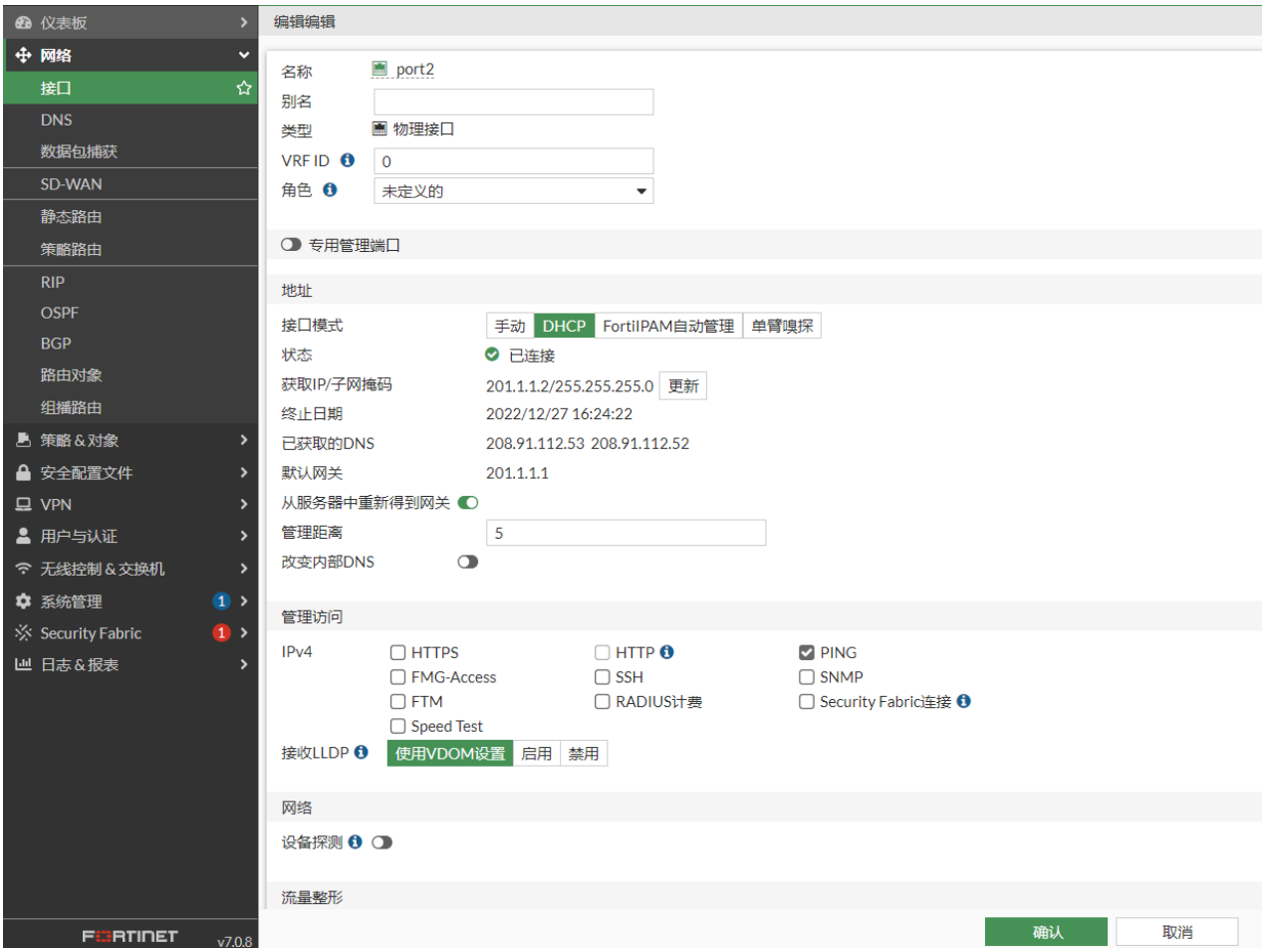
```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

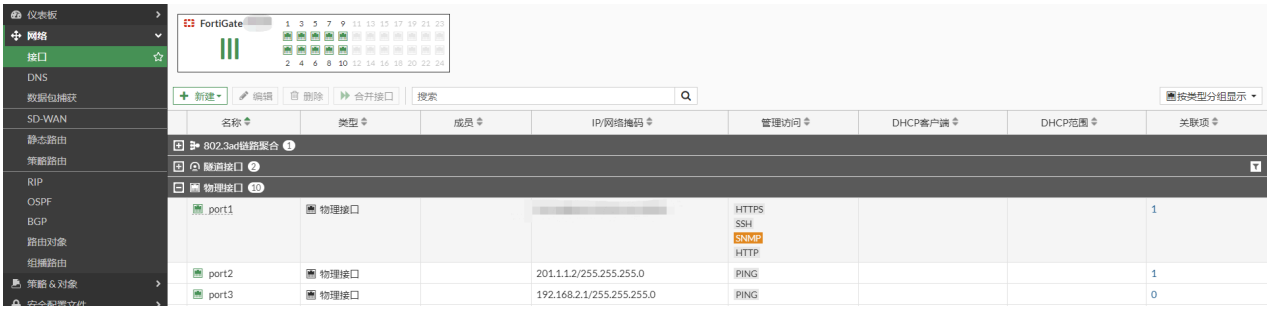
   IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

SPOKE2 配置

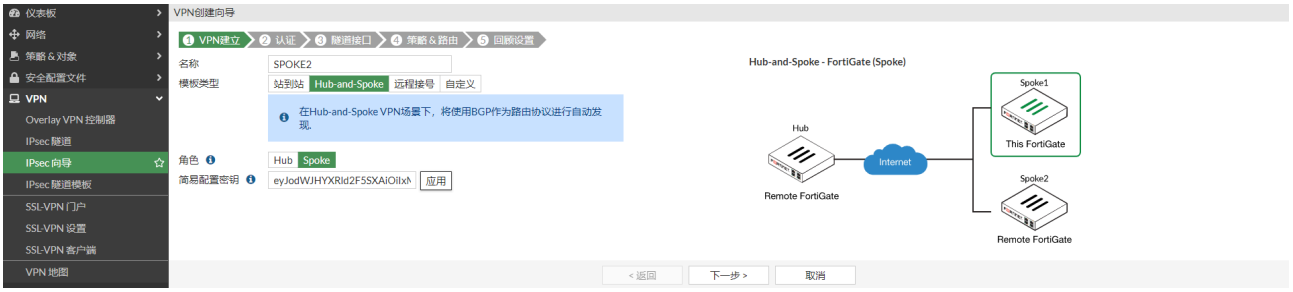
1. 基本配置



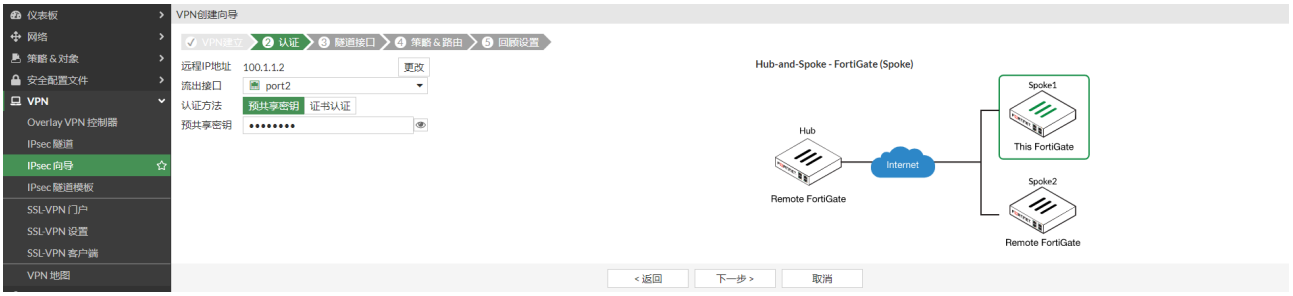


2.配置 IPSEC VPN

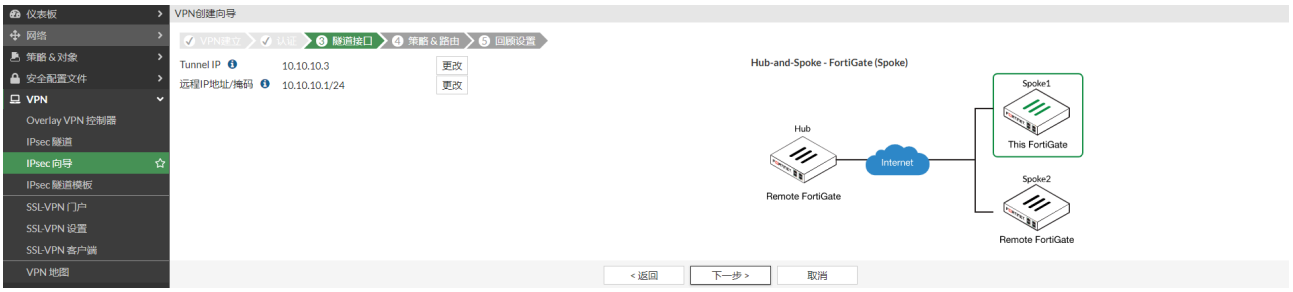
选择“Hub and Spoke”模板。输入在 HUB 生成的 Spoke#2 的秘钥并点击应用，然后点击下一步。



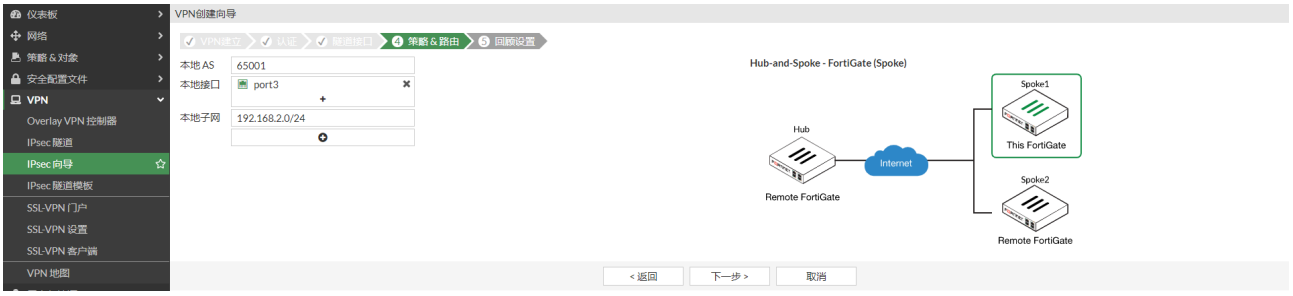
远程 IP 地址和流出接口会自动生成，需要输入预共享秘钥。点击下一步。



自动生成隧道 IP，点击下一步。



自动生成本地 AS 号，需要生成 SPOKE1 本地的内网接口及需要保护的内网网段，点击下一步。



VPN 向导即将创建的内容，点击完成。



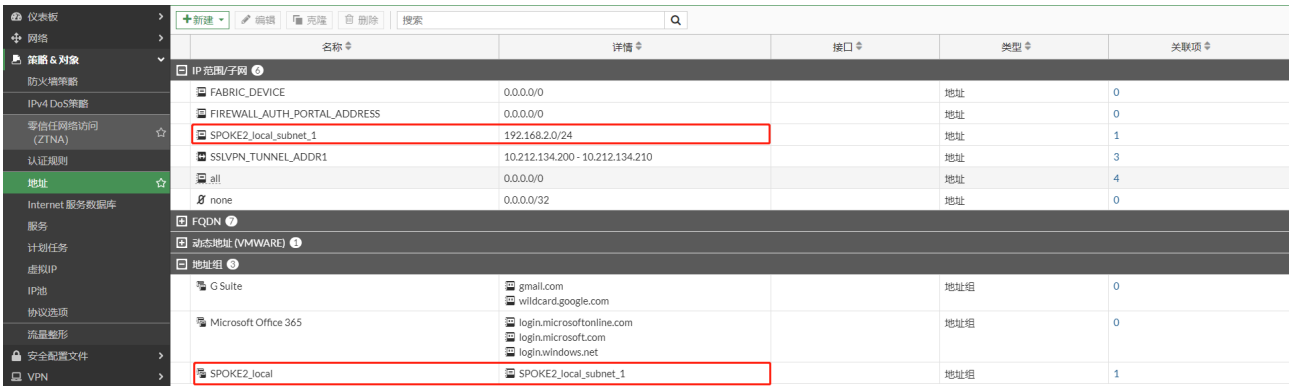
VPN 创建成功。



查看 SPOKE2 端 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建地址对象和地址对象组



2.创建 VPN



对应的命令行

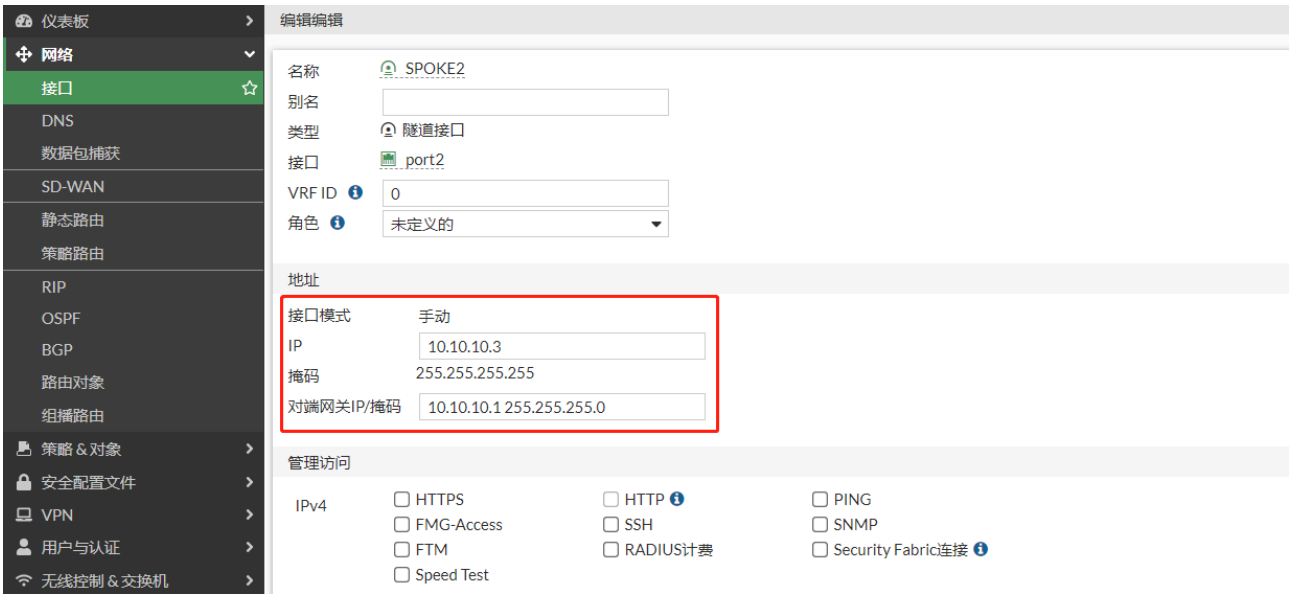
```
config vpn ipsec phase1-interface
    edit "SPOKE2"
        set interface "port2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set comments "VPN: SPOKE2 (Created by VPN wizard)"
        set wizard-type spoke-fortigate-auto-discovery
        set auto-discovery-receiver enable
        set remote-gw 100.1.1.2
```

```
        set psksecret ENC
Hhob5itx47wX/q8zqk2vgdQqbTGPVR3Ks4Bti+eH4AOK/4vS+a1JTnh4IwhyrQ8J8APUDv6ttVJDUV00lROhiXPy3XNqtZR1Vw8mlAv1I3KQTvEWV9/AUZRWRq
oEKN8uo98qbPR97LC5NPQ7OIiZYvsh8T5XQhHGzRNiQpL4sSWBRbJDTvKrxarnuJr8UIdzHqO7xA==
    next
end
config vpn ipsec phase2-interface
    edit "SPOKE2"
        set phase1name "SPOKE2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: SPOKE2 (Created by VPN wizard)"
    next
end
```

3.创建策略



4.创建 VPN 接口 IP



5.创建 BGP



针对向导配置的优化建议

1. 在第一阶段中开启 DPD 周期性检测（隧道没有流量时，每隔 10s 检测一次 Peer 状态），实现快速的检测并切换 VPN 隧道的目的。

```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

2. 开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

   IPSEC VPN 阶段一自动协商是默认开启的。

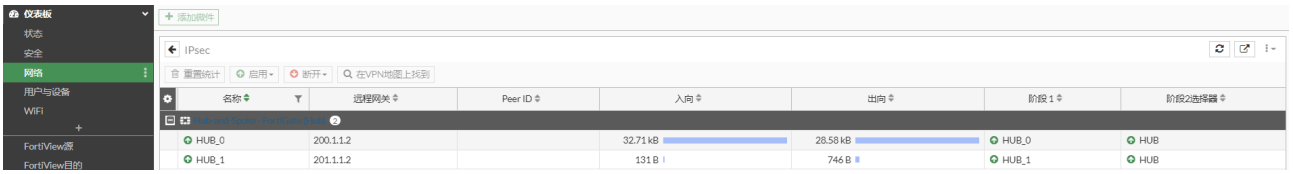
```
config vpn ipsec phase1-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

   IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "VPN-to-SH"
        set auto-negotiate enable
    next
end
```

查看 VPN 和路由状态

1.HUB 端 VPN 和路由状态




```
# diagnose vpn  ike gateway list

vd: root/0
name: HUB_0
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 10.10.10.2/::10.0.0.4
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.1 → 10.10.10.2
created: 212s ago
auto-discovery: 1 sender
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 0 1f454ba48eb31ef1/779d7de61a925ba5
  direction: responder
  status: established 212-212s ago = 0ms
  proposal: aes128-sha256
  key: 26b761298dff1684-03a1e6994809ae6d
  lifetime/rekey: 86400/85917
  DPD sent/recv: 00000000/00000000

vd: root/0
name: HUB_1
version: 1
interface: port2 10
addr: 100.1.1.2:500 → 201.1.1.2:500
tun_id: 10.10.10.3/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.1 → 10.10.10.3
created: 155s ago
auto-discovery: 1 sender
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 1 4b28344cdc4862a2/c0f31569b4326560
  direction: responder
  status: established 155-155s ago = 0ms
  proposal: aes128-sha256
  key: 5e6660c76dfafef8-062dfbfdc7c83d2d
  lifetime/rekey: 86400/85974
  DPD sent/recv: 00000000/00000008

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=HUB_0 ver=1 serial=5 100.1.1.2:0→200.1.1.2:0 tun_id=10.10.10.2 tun_id6=::10.0.0.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=s/1
stat: rxp=2 txp=482 rxb=32710 txb=34637
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=1
```

```
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=4 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=a26 type=00 soft=0 mtu=1438 expire=43082/0B replaywin=2048
       seqno=f2 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
  dec: spi=9fbaba90 esp=aes key=16 a779f017a79a7e25bebcd73a8729ca4d
       ah=sha1 key=20 032444a21009bb1ed477c1b1c9fff594761031d5
  enc: spi=54b85815 esp=aes key=16 c4dcfc64ac2175b63e9e35198755e4ac
       ah=sha1 key=20 9b19cfd19c928ae8d12de4c6a76a7040e9baf274
  dec:pkts/bytes=2/16392, enc:pkts/bytes=482/47064
  npu_flag=03 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=2 dec_npuid=1 enc_npuid=1
--------------------------------------------------------------
name=HUB_1 ver=1 serial=6 100.1.1.2:0->201.1.1.2:0 tun_id=10.10.10.3 tun_id6=::10.0.0.5 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=HUB index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=s/1
stat: rxp=2 txp=12 rxb=131 txb=798
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=2
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=HUB proto=0 sa=1 ref=3 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=a26 type=00 soft=0 mtu=1438 expire=43120/0B replaywin=2048
       seqno=4 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43190/43200
  dec: spi=9fbaba91 esp=aes key=16 2670729ad842dd91c879db020170554b
       ah=sha1 key=20 1b6eb76171659fc431e7ffaed7e90bb9d7daa1fe
  enc: spi=f01e0450 esp=aes key=16 e4ed2f14faac5e5ddf7dec37ac8d6843
       ah=sha1 key=20 12f6d432f0a1ebef51de6e32d811dba066f54741
  dec:pkts/bytes=2/142, enc:pkts/bytes=6/551
  npu_flag=03 npu_rgwy=201.1.1.2 npu_lgwy=100.1.1.2 npu_selid=3 dec_npuid=1 enc_npuid=1
--------------------------------------------------------------
name=HUB ver=1 serial=4 100.1.1.2:0->0.0.0.0:0 tun_id=10.0.0.2 tun_id6=::10.0.0.2 dst_mtu=0 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=42953655 olast=42953655 ad=/0
stat: rxp=4 txp=494 rxb=32841 txb=35435
dpd: mode=on-idle on=0 idle=10000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

查看路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port2, [1/0]
C       10.10.10.0/24 is directly connected, HUB
C       10.10.10.1/32 is directly connected, HUB
C       100.1.1.0/24 is directly connected, port2
C       192.168.0.0/24 is directly connected, port5
B       192.168.1.0/24 [200/0] via 10.10.10.2 (recursive is directly connected, HUB), 00:50:01
B       192.168.2.0/24 [200/0] via 10.10.10.3 (recursive is directly connected, HUB), 00:29:13
```

2.SPOKE1 端 VPN 和路由状态



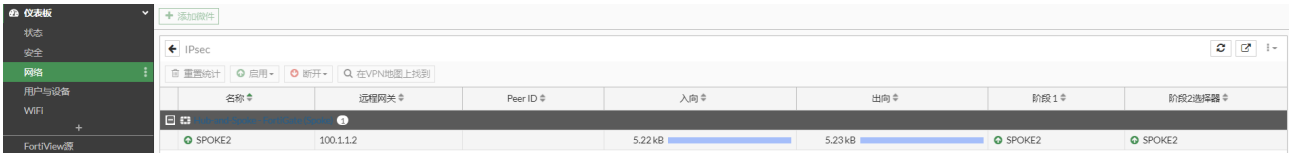

```
# diagnose vpn ike gateway list

vd: root/0
name: SPOKE1
version: 1
interface: port2 10
addr: 200.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.2 → 10.10.10.1
created: 397s ago
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 0 1f454ba48eb31ef1/779d7de61a925ba5
  direction: initiator
  status: established 397-397s ago = 0ms
  proposal: aes128-sha256
  key: 26b761298dff1684-03a1e6994809ae6d
  lifetime/rekey: 86400/85702
  DPD sent/recv: 00000004/00000000

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE1 ver=1 serial=1 200.1.1.2:0→100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=auto/1 encap=none/568 options[0238]=npu create_dev frag-rfc  role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=11 olast=11 ad=r/2
stat: rxp=2 txp=649 rxb=32678 txb=45413
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE1 proto=0 sa=1 ref=3 serial=2 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=42643/0B replaywin=2048
       seqno=1bf esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42933/43200
  dec: spi=54b85815 esp=aes key=16 c4dcfc64ac2175b63e9e35198755e4ac
       ah=sha1 key=20 9b19cfd19c928ae8d12de4c6a76a7040e9baf274
  enc: spi=9fbaba90 esp=aes key=16 a779f017a79a7e25bebcd73a8729ca4d
       ah=sha1 key=20 032444a21009bb1ed477c1b1c9fff594761031d5
  dec:pkts/bytes=2/16342, enc:pkts/bytes=892/90448
  npu_flag=03 npu_rgwy=100.1.1.2 npu_lgwy=200.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
```

查看路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
```

```
C       10.10.10.2/32 is directly connected, SPOKE1
C       192.168.1.0/24 is directly connected, port5
C       200.1.1.0/24 is directly connected, port2
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE1 tunnel 100.1.1.2), 00:52:00
B       192.168.2.0/24 [200/0] via 10.10.10.3 (recursive via SPOKE1 tunnel 100.1.1.2), 00:30:43
```

3.SPOKE2 端 VPN 和路由状态




```
# diagnose vpn ike gateway list

vd: root/0
name: SPOKE2
version: 1
interface: port2 6
addr: 201.1.1.2:500 → 100.1.1.2:500
tun_id: 100.1.1.2/::100.1.1.2
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.3 → 10.10.10.1
created: 429s ago
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 0 4b28344cdc4862a2/c0f31569b4326560
  direction: initiator
  status: established 429-429s ago = 0ms
  proposal: aes128-sha256
  key: 5e6660c76dfafef8-062dfbfdc7c83d2d
  lifetime/rekey: 86400/85671
  DPD sent/recv: 00000017/00000003

# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=SPOKE2 ver=1 serial=2 201.1.1.2:0→100.1.1.2:0 tun_id=100.1.1.2 tun_id6=::100.1.1.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/568 options[0238]=npu create_dev frag-rfc  role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=9 olast=9 ad=r/2
stat: rxp=27 txp=24 rxb=1863 txb=1536
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=23
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE2 proto=0 sa=1 ref=3 serial=2 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=1a203 type=00 soft=0 mtu=1438 expire=42552/0B replaywin=2048
       seqno=f esn=0 replaywin_lastseq=00000013 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=f01e0450 esp=aes key=16 e4ed2f14faac5e5ddf7dec37ac8d6843
       ah=sha1 key=20 12f6d432f0a1ebef51de6e32d811dba066f54741
  enc: spi=9fbaba91 esp=aes key=16 2670729ad842dd91c879db020170554b
       ah=sha1 key=20 1b6eb76171659fc431e7ffaed7e90bb9d7daa1fe
  dec:pkts/bytes=36/2480, enc:pkts/bytes=28/2653
  npu_flag=00 npu_rgwy=100.1.1.2 npu_lgwy=201.1.1.2 npu_selid=2 dec_npuid=0 enc_npuid=0
```

查看路由表

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
```

```
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0
S       10.10.10.0/24 [5/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
C       10.10.10.3/32 is directly connected, SPOKE2
C       192.168.2.0/24 is directly connected, port3
C       200.1.1.0/24 is directly connected, port2
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE2 tunnel 100.1.1.2), 00:34:25
B       192.168.1.0/24 [200/0] via 10.10.10.2 (recursive via SPOKE2 tunnel 100.1.1.2), 00:34:25
```

业务测试

1.SPOKE1 端访问 HUB

```
PC2# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::1a6c:e61:d2b9:a415  prefixlen 64  scopeid 0x20<link>
        inet6 2001::2  prefixlen 64  scopeid 0x0<global>
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 5904889  bytes 459674552 (438.3 MiB)
        RX errors 0  dropped 58  overruns 0  frame 0
        TX packets 3224540  bytes 205411564402 (191.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC2# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.746 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.765 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.862 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=0.677 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.677/0.762/0.862/0.071 ms
```

2.SPOKE2 端访问 HUB

```
PC3# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::2652:4dd7:5d0e:941d  prefixlen 64  scopeid 0x20<link>
        inet6 240e:604:109:39::216  prefixlen 64  scopeid 0x0<global>
        ether 00:0c:29:37:f0:ac  txqueuelen 1000  (Ethernet)
        RX packets 9638867  bytes 205844622412 (191.7 GiB)
        RX errors 0  dropped 118  overruns 0  frame 0
        TX packets 5737914  bytes 378901312 (361.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC3# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.848 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.935 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.899 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=1.04 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.848/0.931/1.045/0.081 ms
```

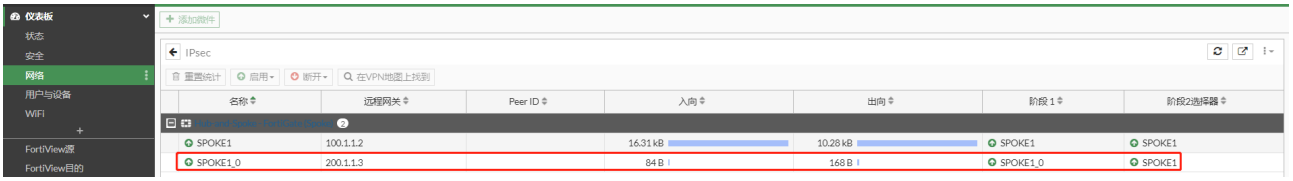
3.SPOKE 之间访问

```
PC2# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::1a6c:e61:d2b9:a415  prefixlen 64  scopeid 0x20<link>
        inet6 2001::2  prefixlen 64  scopeid 0x0<global>
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 5908666  bytes 460902918 (439.5 MiB)
```

```
        RX errors 0  dropped 58  overruns 0  frame 0
        TX packets 3225276  bytes 205411619294 (191.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

SPOKE1访问SPOKE2
PC2# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=61 time=1.27 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.925 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.812 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.812/1.004/1.276/0.199 ms
```

SPOKE 之间访问时，会触发 SPOKE 之间创建单独的 VPN。

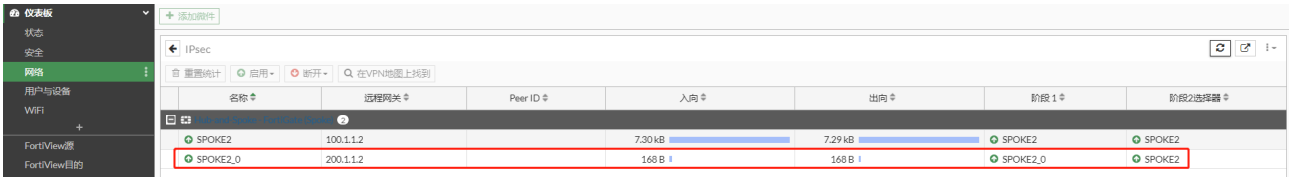
在 SPOKE1 上查看与 SPOKE2 之间的 VPN 和路由。




```
# diagnose vpn ike gateway list name SPOKE1_0

vd: root/0
name: SPOKE1_0
version: 1
interface: port2 10
addr: 200.1.1.2:500 → 201.1.1.2:500
tun_id: 201.1.1.2/::201.1.1.2
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.2 → 10.10.10.3
created: 30s ago
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 c9a21426384683c3/9ec0d3e19cd425e1
  direction: initiator
  status: established 30-30s ago = 0ms
  proposal: aes128-sha256
  key: 8caa0db523816e6c-c1c1d43ab5961a5a
  lifetime/rekey: 86400/86069
  DPD sent/recv: 00000003/00000000

# diagnose vpn tunnel list name SPOKE1_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=SPOKE1_0 ver=1 serial=3 200.1.1.2:0→201.1.1.2:0 tun_id=201.1.1.2 tun_id6=::201.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/760 options[02f8]=npu create_dev no-sysctl rgwy-chg frag-
rfc  role=primary accept_traffic=1 overlay_id=0

parent=SPOKE1 index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=2 olast=2 ad=r/2
stat: rxp=1 txp=2 rxb=84 txb=168
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=SPOKE1 proto=0 sa=1 ref=3 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=42855/0B replaywin=2048
       seqno=3 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=54b85816 esp=aes key=16 0478ea93933324f29cf01524e7d28ec4
       ah=sha1 key=20 5fb7754757facf499e85b1bdb314520d3cf10faa
```

```
  enc: spi=f01e0451 esp=aes key=16 183a97d565d45c86096969afeacd60ec
       ah=sha1 key=20 7f1cf1f622146d60453f68f35ed9804ab4566d46
  dec:pkts/bytes=2/168, enc:pkts/bytes=4/472
  npu_flag=03 npu_rgwy=201.1.1.2 npu_lgwy=200.1.1.2 npu_selid=3 dec_npuid=1 enc_npuid=1

  # get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 200.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE1 tunnel 100.1.1.2, [1/0]
C       10.10.10.2/32 is directly connected, SPOKE1
                      is directly connected, SPOKE1_0
C       10.10.10.3/32 is directly connected, SPOKE1_0
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE1 tunnel 100.1.1.2), 00:07:22
B       192.168.2.0/24 [200/0] via 10.10.10.3 (recursive is directly connected, SPOKE1_0), 00:00:26
C       200.1.1.0/24 is directly connected, port2
C       192.168.1.0/24 is directly connected, port5
```

在 SPOKE2 上查看与 SPOKE1 之间的 VPN。




```
# diagnose vpn ike gateway list name SPOKE2_0

vd: root/0
name: SPOKE2_0
version: 1
interface: port2 6
addr: 201.1.1.2:500 → 200.1.1.2:500
tun_id: 10.10.10.2/::10.0.0.1
remote_location: 0.0.0.0
network-id: 0
virtual-interface-addr: 10.10.10.3 → 10.10.10.2
created: 150s ago
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 c9a21426384683c3/9ec0d3e19cd425e1
  direction: responder
  status: established 150-150s ago = 0ms
  proposal: aes128-sha256
  key: 8caa0db523816e6c-c1c1d43ab5961a5a
  lifetime/rekey: 86400/85979
  DPD sent/recv: 00000000/0000000f

# diagnose vpn  tunnel list name SPOKE2_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=SPOKE2_0 ver=1 serial=3 201.1.1.2:0→200.1.1.2:0 tun_id=10.10.10.2 tun_id6=::10.0.0.1 dst_mtu=1500 dpd-link=on
weight=1
bound_if=6 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/760 options[02f8]=npu create_dev no-sysctl rgwy-chg frag-rfc
role=primary accept_traffic=1 overlay_id=0

parent=SPOKE2 index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=6 olast=6 ad=r/2
stat: rxp=2 txp=2 rxb=168 txb=168
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
```

```
proxyid=SPOKE2 proto=0 sa=1 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=1a203 type=00 soft=0 mtu=1438 expire=43034/0B replaywin=2048
       seqno=3 esn=0 replaywin_lastseq=00000003 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43191/43200
  dec: spi=f01e0451 esp=aes key=16 183a97d565d45c86096969afeacd60ec
       ah=sha1 key=20 7f1cf1f622146d60453f68f35ed9804ab4566d46
  enc: spi=54b85816 esp=aes key=16 0478ea93933324f29cf01524e7d28ec4
       ah=sha1 key=20 5fb7754757facf499e85b1bdb314520d3cf10faa
  dec:pkts/bytes=4/336, enc:pkts/bytes=4/472
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=201.1.1.2 npu_selid=3 dec_npuid=0 enc_npuid=0

# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 201.1.1.1, port2, [1/0]
S       10.10.10.0/24 [5/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
S       10.10.10.1/32 [15/0] via SPOKE2 tunnel 100.1.1.2, [1/0]
C       10.10.10.2/32 is directly connected, SPOKE2_0
C       10.10.10.3/32 is directly connected, SPOKE2
                      is directly connected, SPOKE2_0
B       192.168.0.0/24 [200/0] via 10.10.10.1 (recursive via SPOKE2 tunnel 100.1.1.2), 00:13:13
B       192.168.1.0/24 [200/0] via 10.10.10.2 (recursive is directly connected, SPOKE2_0), 00:01:48
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
```

野蛮模式/network-id 适用场景

关于主模式中的 Peer ID 问题

1. 在主模式中，第 3、4 个包交互之后（DH 公钥和随机数交换），需要使用到 Pre-shared key（管理员配置的预共享密钥）值用于生成 SHEYID（预共享密钥方式认证：SKEYID = hash (pre-shared key ,Ni | Nr) ），此时是无法通过 Peer-ID 查找到，因为 Peer-ID 参数在第 5、6 个包中，而且还是被加密的，解密需要使用到 SHEYID 为材料生成的密钥 SHEYID_e，因此这个时候只能通过对方发起 IKE 协商的公网 IP 地址去查找 Peer 的 Pre-shared key 以确认其身份。
2. 如此将引出一个问题，在对方（分部）为 PPPoE/DHCP 等动态 IP 场景中时，总部无法配置明确的对方 IP 和 Pre-shared key，而只能配置动态拨号方式的 IPsec VPN，如果我们选择第一阶段为主模式，因为 Peer-ID 在第 5、6 包携带，并且是加密的，因此 Hub 在主模式中是无法提前获知到 Peer-ID 信息的，也没办法根据 Peer-ID 信息去查找到底用哪一个 Pre-shared key，一般动态 IP 场景中主模式使用较少，而且就算使用主模式的话的 Peer-ID 配置为 any 的较多。
3. IKEv1 主模式拨号连接的匹配条件顺序（所有匹配条件都一致时，Spoke 发起的连接匹配 Hub 配置顺序的第一个拨号连接）：
   - local-gw：默认为拨号连接使用的接口 Primary IP，可以手动更改为设备上其他接口的 IP。
   - IKEv1 Main mode + PSK。
   - Proposal + DH group。

什么时候需要野蛮模式

1. Hub 端同一个接口 IP 上存在多条动态拨号的 IKEv1 主模式 IPsec VPN，且使用的算法、预共享密钥、DH 组一致。此时由于拨号 VPN 的目的 IP 都是 any（0.0.0.0），此时根本无法区分这两条 IPsec VPN 隧道，不知道拨号用户到底是匹配 Dia_VPN1 还是匹配 Dia_VPN2（FGT 的处理是优先匹配靠前的拨号 VPN），因此为了区分本地的多条动态拨号 VPN，则需要配置 Peer ID 参数，以便区分这本地的多条动态 IPsec VPN 隧道，例如：
   - Dia_VPN1（动态拨号 VPN 野蛮模式）：
     - Local ID：BJ
     - Peer ID：South-FGT
     - Dia_VPN2（动态拨号 VPN 野蛮模式）：
       - Local ID：BJ
       - Peer ID：North-FGT
2. 如果对方拨号用户 1 填写的 Local ID 是 South-FGT 则连接到 Dia_VPN1，如果对方拨号用户 2 填写的 Local ID 是 North-FGT 则连接到 Dia_VPN2。这样就可以将不通的拨号用户进行区分和分离，方便进行独立的 VPN 隧道、独立的策略进行控制。在这种需求的情况下：首先需要舍弃掉主模式，用野蛮模式，同时使用不同的 Peer-ID 来区分不同的动态 VPN 隧道。
3. IKEv1 野蛮模式拨号连接的匹配条件顺序（所有匹配条件都一致时，Spoke 发起的连接匹配 Hub 配置顺序的第一个拨号连接）：
   - local-gw：默认为拨号连接使用的接口 Primary IP，可以手动更改为设备上其他接口的 IP。
   - IKEv1 Main mode + PSK。
   - Proposal + DH group。
   - Peer ID（开启客户端用户认证时使用 user.local.name ↗）。

什么时候需要 network-id

1. 在 IKEv2 的协商报文中，ID 信息包含在 IKE_AUTH 报文中，且被加密。所以会出现和 IKEv1 主模式一样的问题（同一个接口 IP 上存在多条动态拨号的 IKEv1 主模式 IPsec VPN，且使用的算法、预共享密钥、DH 组一致）。
2. Fortinet 使用 network-id 属性来解决此问题，该字段包含在 IKEv2 协商的第 1、2 个报文（IKE_SA_INIT）中，没有被加密，**network-id 为 Fortinet 私有**。

```bash
Hub:
config vpn ipsec phase1-interface
    edit "Hub_1"
        set ike-version 2
        set network-overlay enable
        set network-id 1
    next
end
config vpn ipsec phase1-interface
    edit "Spoke1"
```

```
        set ike-version 2
        set network-overlay enable
        set network-id 1
    next
end
```

3. Hub 在同一个接口 IP 上存在多条动态拨号的 IKEv2 IPsec VPN，且使用的算法、预共享密钥、DH 组一致，可以配置不同的 network-id 来区分不同的 Spoke 拨入连接。

```bash
config vpn ipsec phase1-interface
    edit "Hub_1"
        set type dynamic
        set interface "wan1"
        set ike-version 2
        set network-overlay enable
        set network-id 1
    next
        set type dynamic
        edit "Hub_2"
        set interface "wan1"
        set ike-version 2
        set network-overlay enable
        set network-id 2
    next
end
```

4. 每个 Spoke 配置不同的 network-id 来连接 Hub 上对应的拨号连接。

```bash
Spoke1:
config vpn ipsec phase1-interface
    edit "Spoke1"
        set interface "port2"
        set ike-version 2
        set network-overlay enable
        set network-id 1
    next
end

Spoke2:
config vpn ipsec phase1-interface
    edit "Spoke1"
        set interface "port2"
        set ike-version 2
        set network-overlay enable
        set network-id 2
    next
end
```

5. IKEv2 拨号连接的匹配条件顺序（所有匹配条件都一致时，Spoke 发起的连接匹配 Hub 配置顺序的第一个拨号连接）：

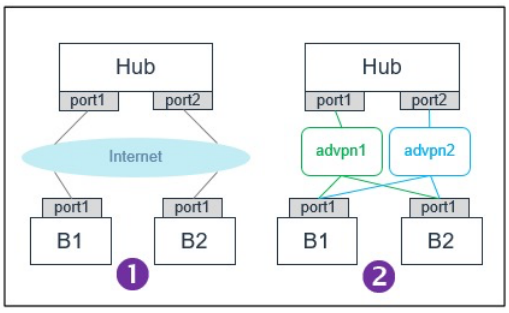
   - local-gw：默认为拨号连接使用的接口 Primary IP，可以手动更改为设备上其他接口的 IP。
   - IKEv1 Main mode + PSK。
   - Proposal + DH group。
   - network-id。

ADVPN 下的 network-id 或野蛮模式 ID

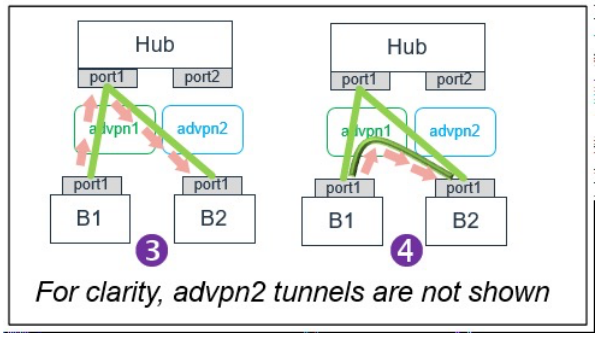
在 ADVPN 部署模式下，如果每个 Spoke 与 Hub 建立两条 IPSec 隧道，Spoke 与 Spoke 创建 shortcut 隧道前，需要注意配置 IKEv2 的 network-id 或野蛮模式 ID。如下举例的 ADVPN 环境，所有 IPSec 连接使用 IKEv2 建立。

1. ADVPN 部署，B1 和 B2 只有一条运营商线路接入，Hub 有两条运营商线路。

2. B1 和 B2 的单线路与 Hub 的两个线路接口分别创建 overlay IPSec 隧道 advpn1 和 advpn2。

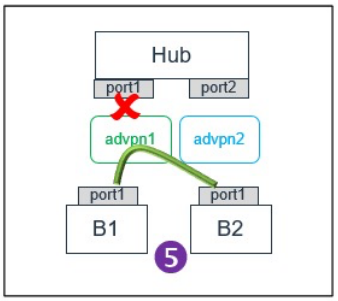


3. 初始状态下，B1 访问 B2 的流量将经过 B1 → advpn1 → Hub → advpn1 → B2。然后 Hub 将通过 advpn1 协助 B1 和 B2 之间协商建立 Shortcut 隧道。



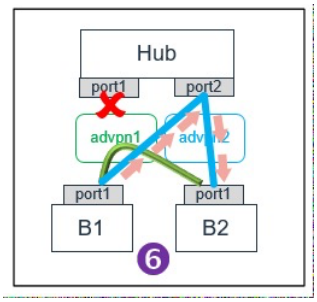
- B1 和 B2 之间建立基于 advpn1 上的 Shortcut。
- 随后从 B1 访问 B2 的流量将经过 Shortcut _advpn1。

4. 如果 Hub 的 port1 线路出现故障。



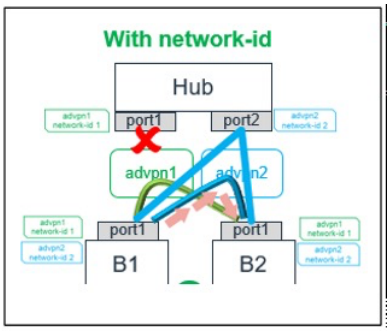
- Hub（port1）和 B1 之间的父隧道将关闭，Hub（port1) 和 B2 之间也会发生同样的情况。
- 同时，B1 和 B2 之间创建的 Shortcut_advpn1 会维持建立状态，因为 ADVPN Shortcut 的 lifetime 与其父隧道（advpn1）的 lifetime 无关。
- B1↔Hub 和 B2↔Hub 通过 advpn1 隧道建立的 BGP 邻居中断。

5. 之后 B1 和 B2 之间的 BGP 路由通过 Hub 的 advpn2 收敛。从 B1 主动访问 B2 的流量会先经过 Hub 转发，因为 B1 和 B2 之间还没有通过 advpn2 创建 Shortcut。



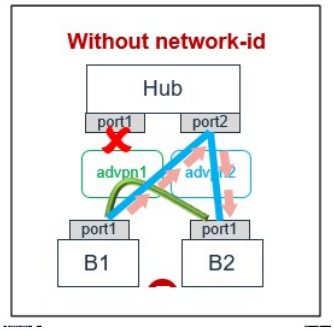
6. Hub 将尝试通过 Advpn2 在 B1 和 B2 之间建立 Shortcut 隧道。

-**如果 B1 上的两条 IPSec 隧道配置了不同的 Network-id**：将在 Advpn2 上的 B1 和 B2 之间建立 advpn2 上的 Shortcut。



- advpn2 和 advpn1 上的 Shortcut 隧道都建立在相同 Underlay 线路 IP 上：B1 (port1) ↔ B2 (port1)。
- 这两个相同 Underlay 源目 IP 的 Shortcut 隧道可以同时建立，因为每个 Overlay IPSec 隧道都配置了不同的 network-id，B2 可以根据不同的 network-id 来区分隧道并与之建立。
- BGP 路由收敛后，B1 访问 B2 的流量流经 advpn2 上的 Shortcut 传输。

-**如果 B1 上的两条 IPSec 隧道没有配置 Network-id**：B2 忽略 B1 在 advpn2 上的 Shortcut 隧道建立请求。



- 因为在相同的 Underlay IP 地址 B1 (port1) ↔ B2 (port1) 上已经存在 Shortcut 隧道（基于 advpn1）。如果不为每个 Overlay 隧道配置不同的 Network-id，则 B2 无法区分两个 advpn 上的 Shortcut 并同时建立两个相同 Underlay 的 Shortcut 隧道。
- 只要 B1 对 B2 Advpn1 上的 Shortcut 隧道建立后，B1 通过 Advpn2 向 B2 发送任何流量，都会通过 Hub 进行转发，因为此时 advpn2 上的 Shortcut 隧道无法建立。

7. 在 Hub 和 Spoke 上都要配置 network-id，且每个线路上 Hub 和 Spoke 的 network-id 保持一致。

```bash
Hub:
config vpn ipsec phase1-interface
    edit "advpn1"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set network-overlay enable
        set network-id 1
        ...
    next
    edit "advpn2"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set network-overlay enable
        set network-id 2
        ...
    next
end

B1&B2:
config vpn ipsec phase1-interface
    edit "advpn1"
        set ike-version 2
        set interface "port1"
        set remote-gw x.x.x.x
        set network-overlay enable
        set network-id 1
        ...
    next
 edit "advpn2"
        set ike-version 2
```

```
        set interface "port1"
        set remote-gw y.y.y.y
        set network-overlay enable
        set network-id 2
        ...
    next
end
```

8. 对于 IKEv1，IKEv1 并不支持 network-id。但也可以通过如下两种方式部署上述的 ADVPN 场景：
   - ADVPN 隧道可以部署为 IKEv1 的野蛮模式，Spoke 之间相同 Underlay IP 的 Shortcut 可以通过配置不同的野蛮模式 ID 来进行区分。
   - 使用 IKEv1 的主模式/野蛮模式，并不配置 ID。通过开启 Shortcut 隧道与父隧道的绑定关系，当 Shortcut 隧道的父隧道中断后，所有关联的 Shortcut 隧道会一起关闭。这样就不会出现上述 Spoke 之间同时存在两个 Shortcut 需要建立的情况，这个配置需要在 Spoke 上的 IPSec 一阶段配置（`set auto-discovery-shortcuts dependent`）：

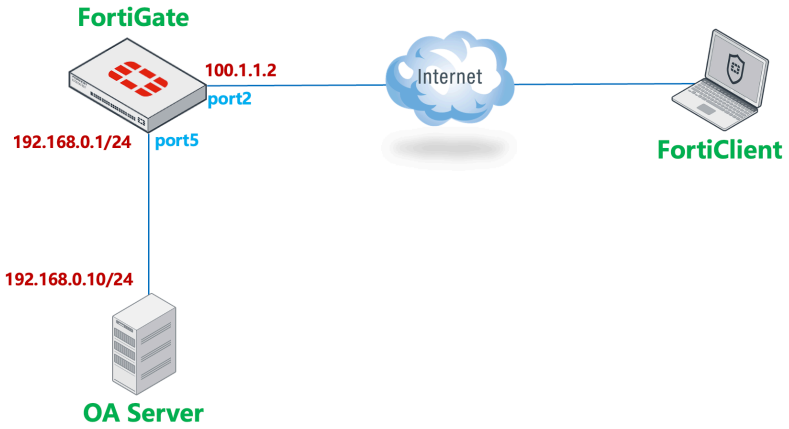
```bash
config vpn ipsec phase1-interface
    edit "advpn1"
        set ike-version 1
        set interface "port1"
        set remote-gw x.x.x.x
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent    //Shortcut隧道状态依赖父隧道状态，默认为independent//
        ...
    next
 edit "advpn2"
        set ike-version 1
        set interface "port1"
        set remote-gw y.y.y.y
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent    //Shortcut隧道状态依赖父隧道状态，默认为independent//
        ...
    next
end
```

FortiClient 拨号

网络需求

某公司内部有一台 OA 服务器，在外移动办公的工作人员需要通过 IPSEC VPN 拨入到公司内网来对内网服 OA 服务器进行访问。

网络拓扑

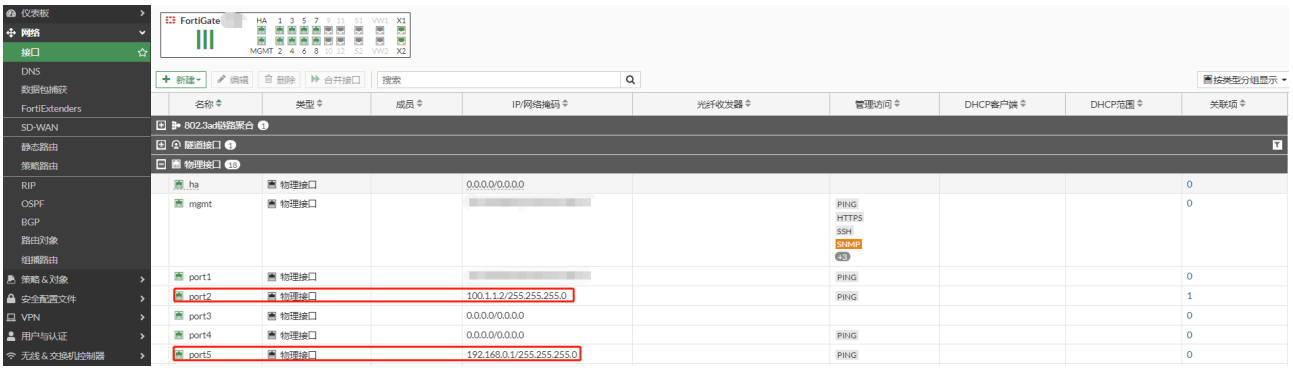


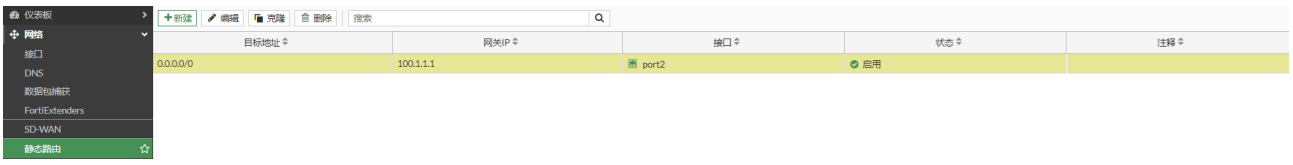
配置步骤

配置 IPSec

1.基本配置

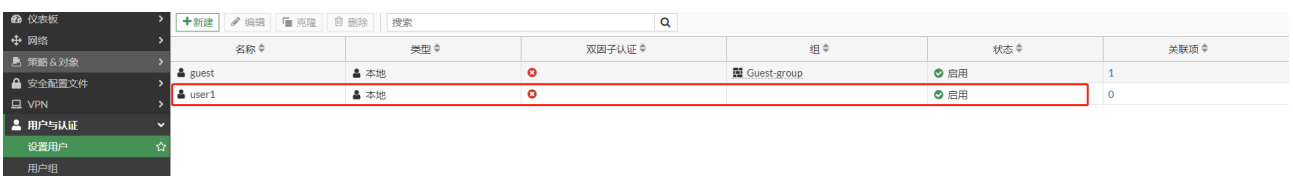
配置接口 IP 和路由





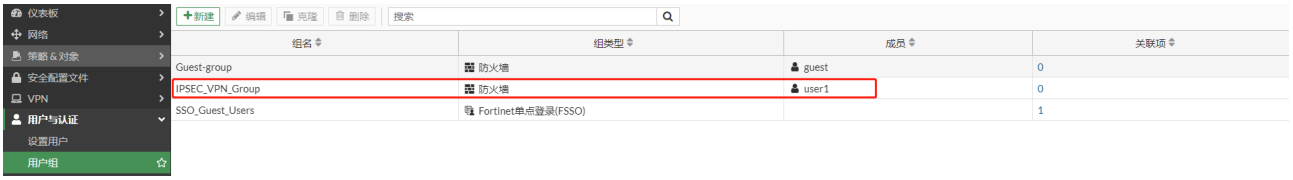
2.创建用户

选择“用户与认证”-->“设置用户”，创建本地用户。

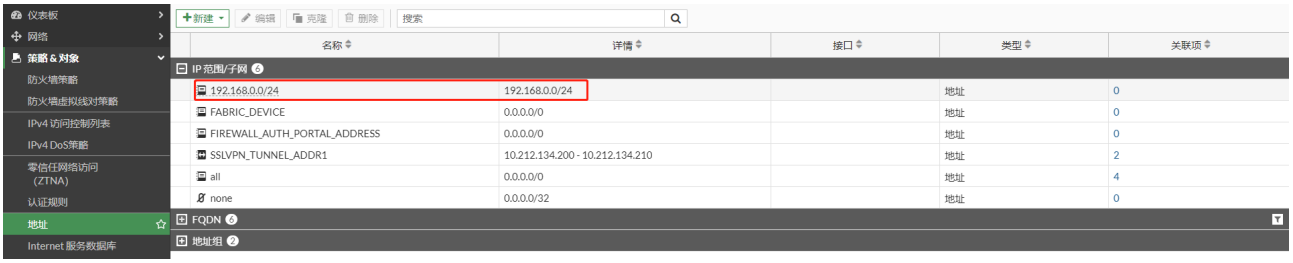


3.创建用户组

选择“用户与认证”-->“用户组”，点击“新建”，创建本地用户组。

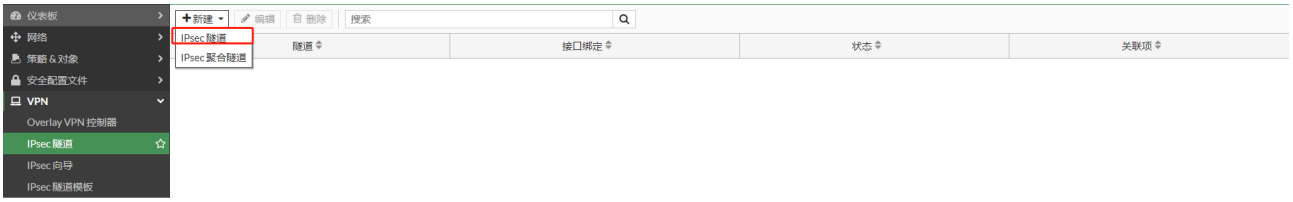


4.创建子网



5.配置 IPSEC VPN

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



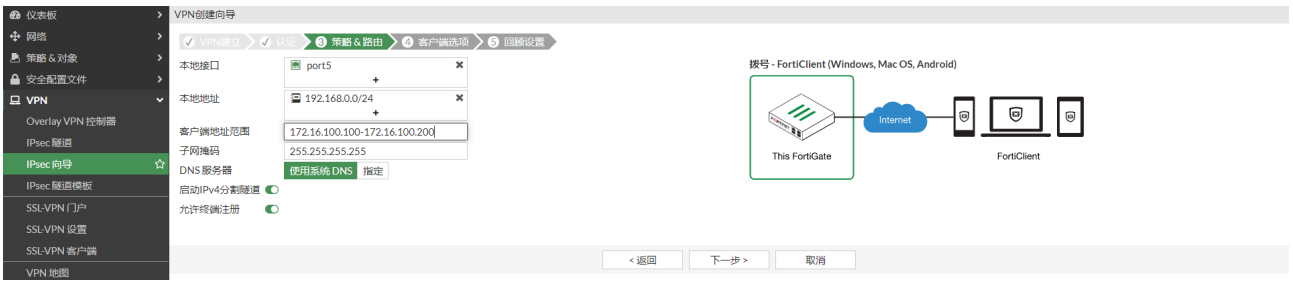
根据“VPN 创建向导”进行 VPN 模板配置，输入名称，选择”远程接“-->"基于客户端"-->“FortiClient”，并点击下一步。



选择对外的接口，设置预共享秘钥和用户组，并点击下一步。



输入内网接口，允许访问的本地地址段，客户端拨号后获取的地址范围，并点击下一步。



保存密码，FortiClient 第一次拨号成功后，FortiClient 会显示“保存密码”的选项，默认勾选。

自动连接，FortiClient 第一次拨号成功后，FortiClient 会显示“自动连接”的选项，该功能是运行 FortiClient 会拨入该 IPSEC VPN。

保持存活，FortiClient 第一次拨号成功后，FortiClient 会显示“保持连接”的选项，该功能是 IPSEC VPN 由于网络原因中断后，会自动重拨。

免费的 FortiClient 版本不支持自动连接和保持连接，需要完整版的 FortiClient，需要购买 EMS license。



VPN 创建向导提示即将创建的内容，然后点击完成。



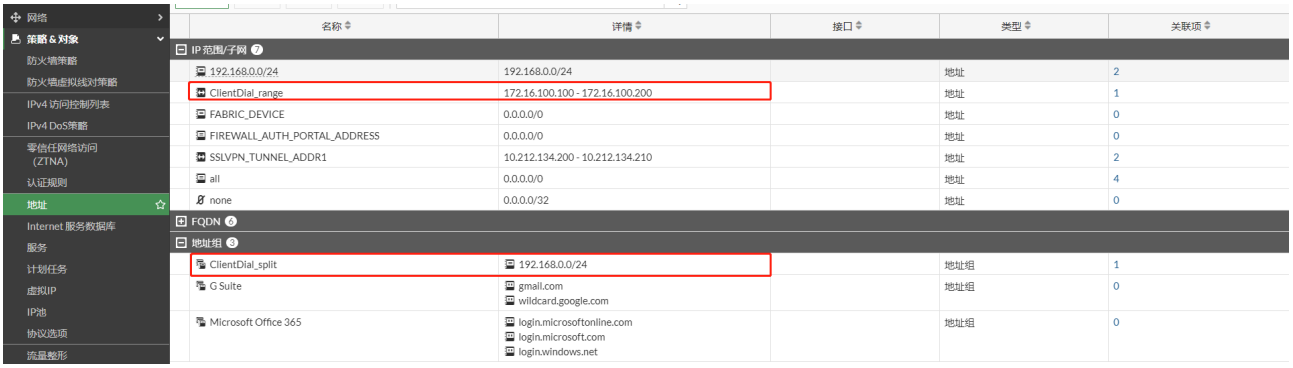
VPN 创建成功。



查看 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建地址对象和地址对象组。



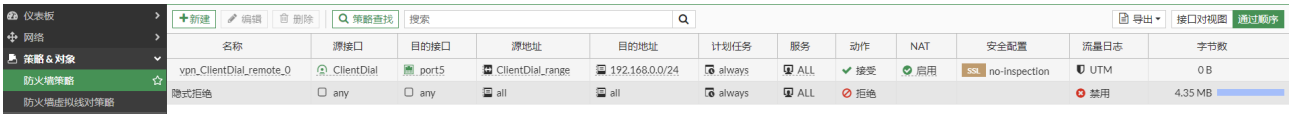
2.创建 IPSEC VPN



对应的命令行

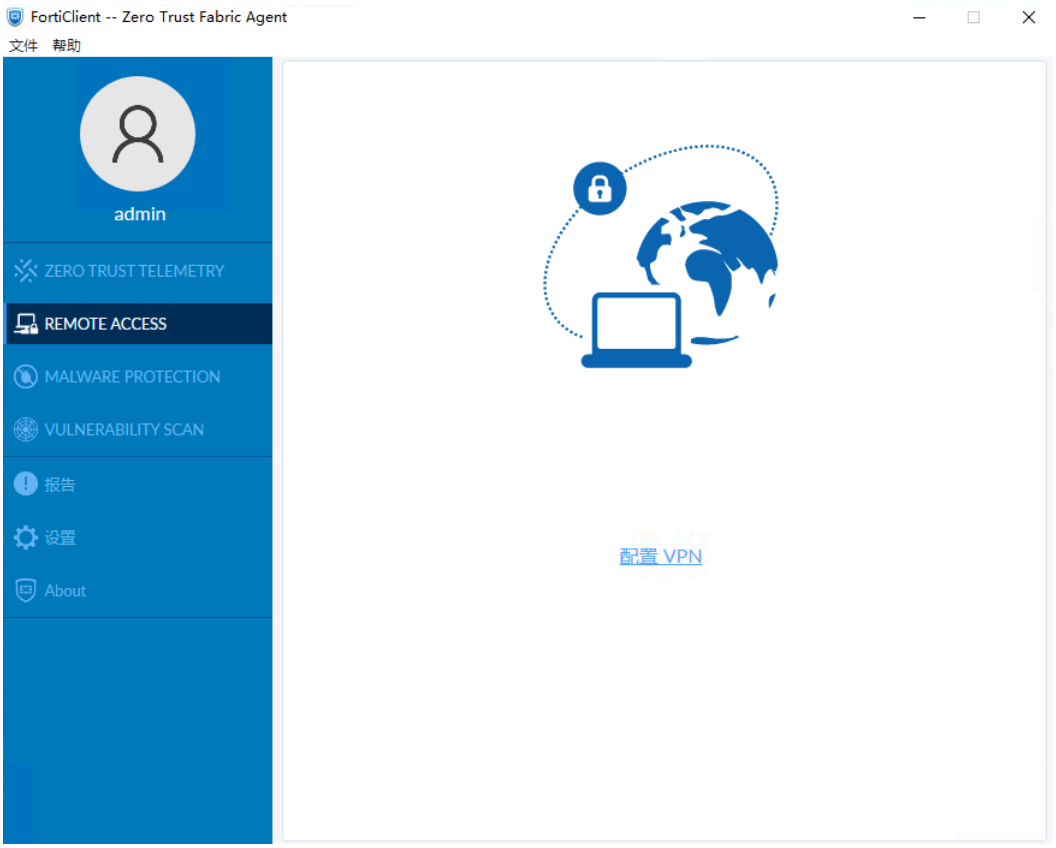
```
config vpn ipsec phase1-interface
    edit "ClientDial"
        set type dynamic
        set interface "port2"
        set mode aggressive
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: ClientDial (Created by VPN wizard)"
        set wizard-type dialup-forticlient
        set xauthtype auto
        set authusrgrp "IPSEC_VPN_Group"
        set ipv4-start-ip 172.16.100.100
        set ipv4-end-ip 172.16.100.200
        set dns-mode auto
        set ipv4-split-include "ClientDial_split"
        set save-password enable
        set psksecret ENC
cTt0jnn1i1wx0Elky0oa0pRS3I7Kb8UAUHRndju/6cI7nbcQ1IBcmXqdj9aJXjE/ZNUZgElJLwKARXoP3iJPBAkWJaYtQrEC+XfZC2QGr/Xsw72UQSL0Rll8wI
KAk7vPDIQXbn/eQSYPs7CH+2r64HiKJ91+zt3LzAlASNooXPKea8EK7f9qaUTfj/p/kxqrk9aN6A==
    next
end

config vpn ipsec phase2-interface
    edit "ClientDial"
        set phase1name "ClientDial"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: ClientDial (Created by VPN wizard)"
    next
end
```

2.**创建策略。**



配置 FortClient

1. 选择“Remote Access”，点击“配置 VPN”。

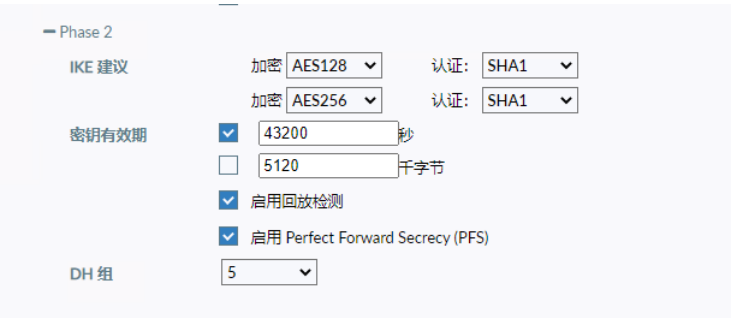


2. 输入连接名，远程网关，共享秘钥，以及用户名，然后点击“保存”。

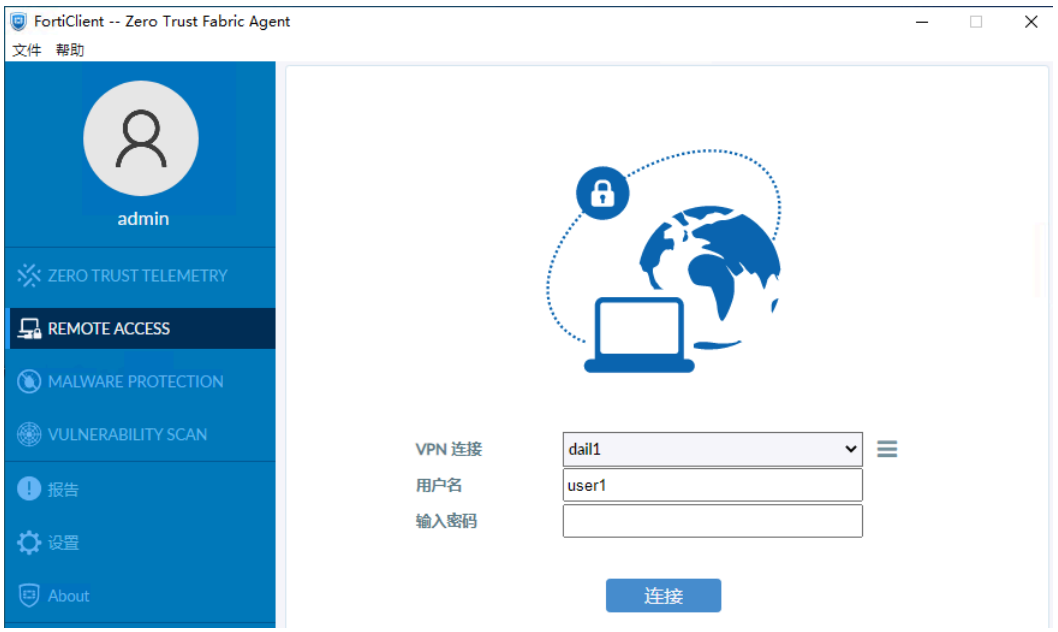


在”高级设置“有更多的选项，如 ike 版本，IKE 阶段一和阶段二的加密集等，可以根据 FortiGate 的设置做对应的修改。





3. VPN 配置完成

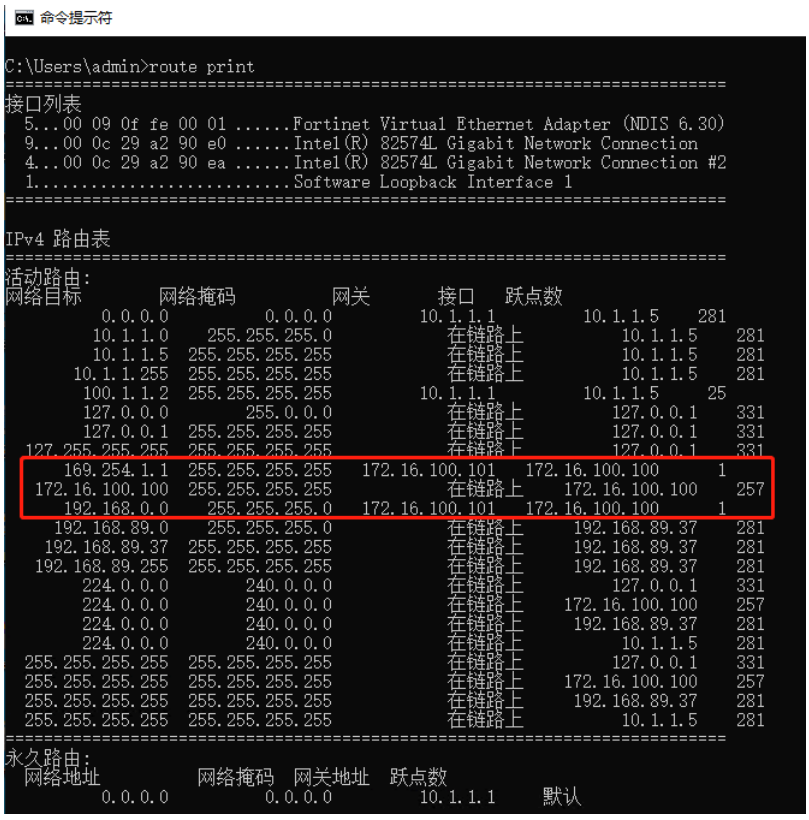


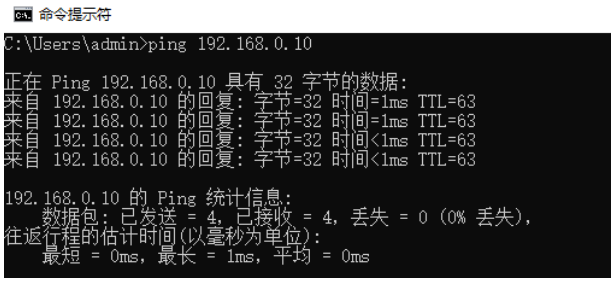
FortClient 拨号测试

1. 在 FortiClient 中输入账号密码，点击“连接”，FortiClient 拨号成功。

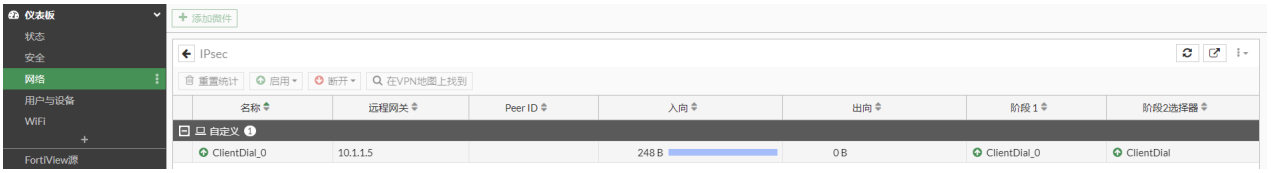


2. 查看终端路由表，192.168.0.0/24 指向 IPSEC VPN，并能成功访问 OA Server。

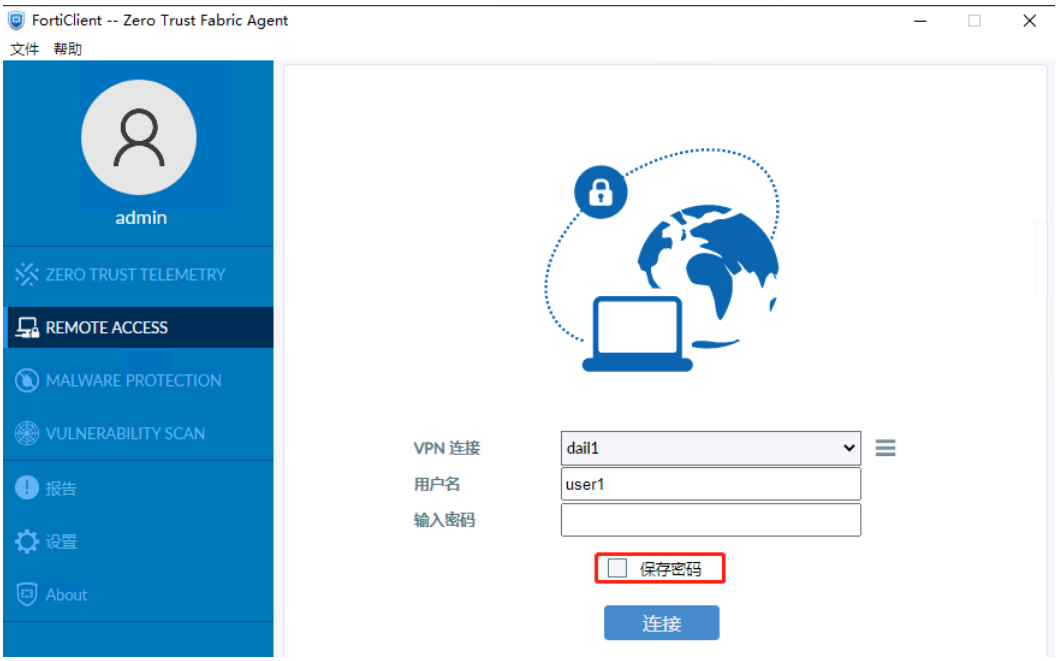




3. FortiGate 查看 client 连接。



4. 点击“中断连接”，FortiClient 就会显示“保存密码”选项。

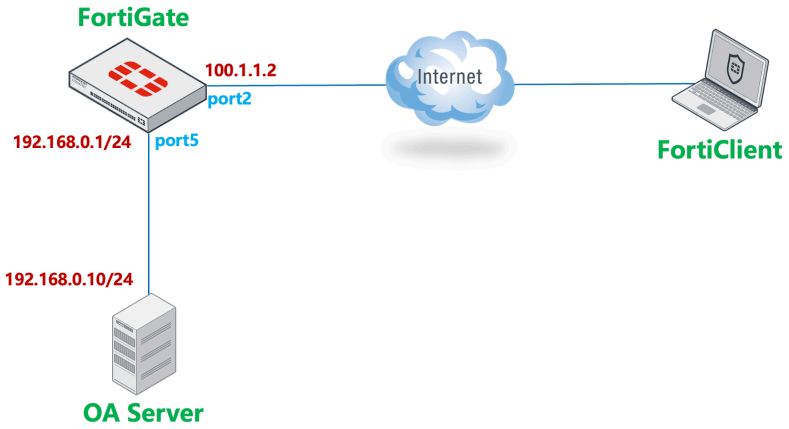


FortiClient 使用证书认证连接 IPSec

网络需求

某公司内部有一台 OA 服务器，在外移动办公的工作人员需要通过 vpn, 拨入到公司内网来对内网服 OA 服务器进行访问，采用证书认证的方式。

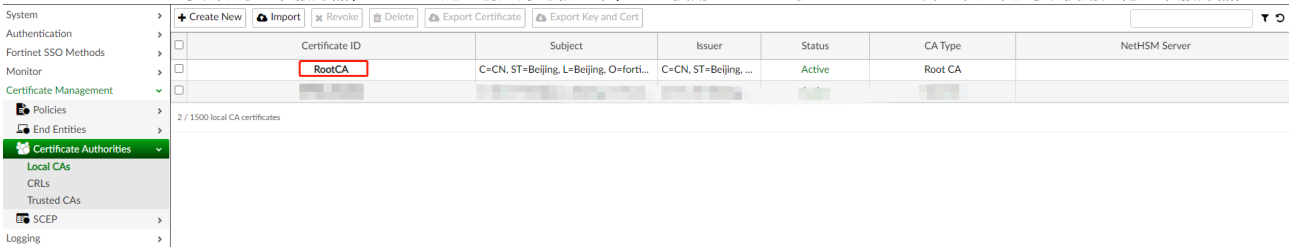
网络拓扑



配置步骤

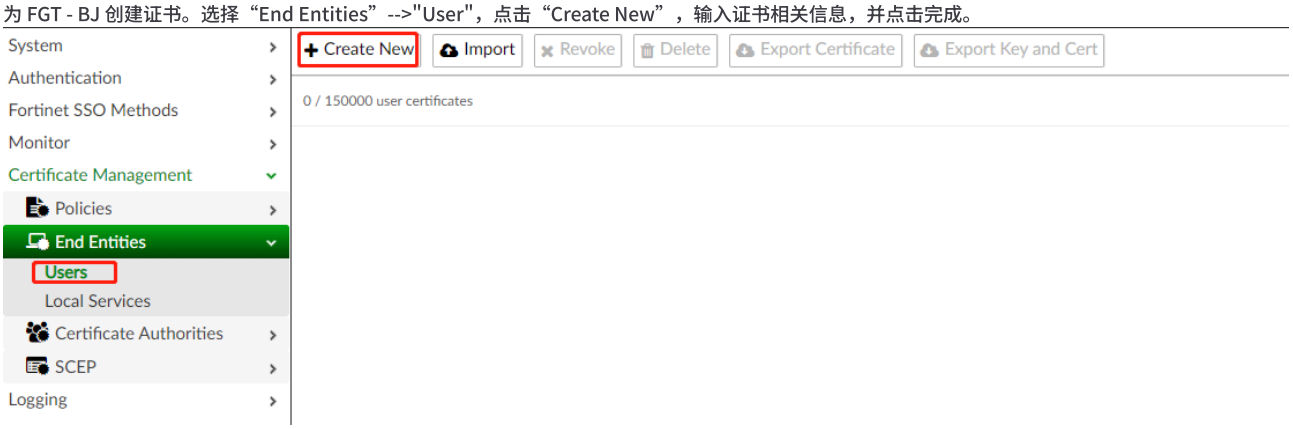
使用 FortiAuthenticator 为 FGT - BJ 和终端颁发证书

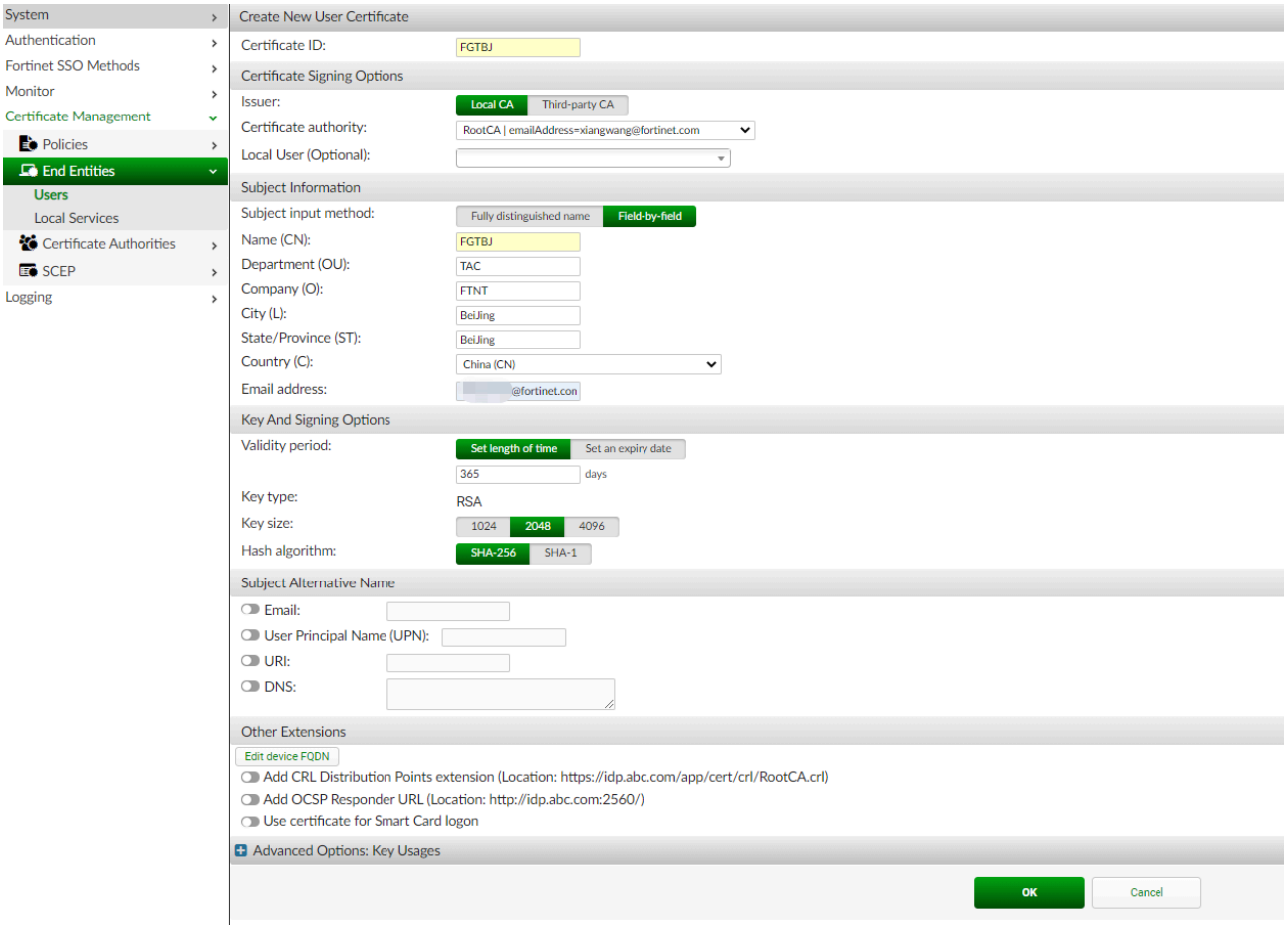
FortiAuthenticator 可以做为证书服务器，RootCA 是已创建好的根证书，这里使用 RootCA 为 FortiGate 签发证书。（也可以使用其他证书服务器）



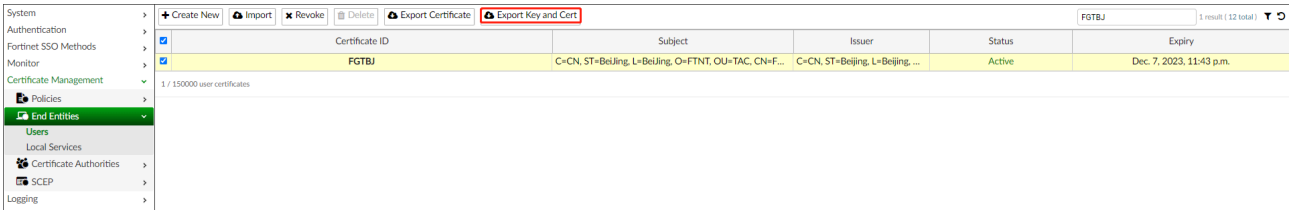
1.为 FGT - BJ 颁发证书

为 FGT - BJ 创建证书。选择“End Entities”-->"User"，点击“Create New”，输入证书相关信息，并点击完成。

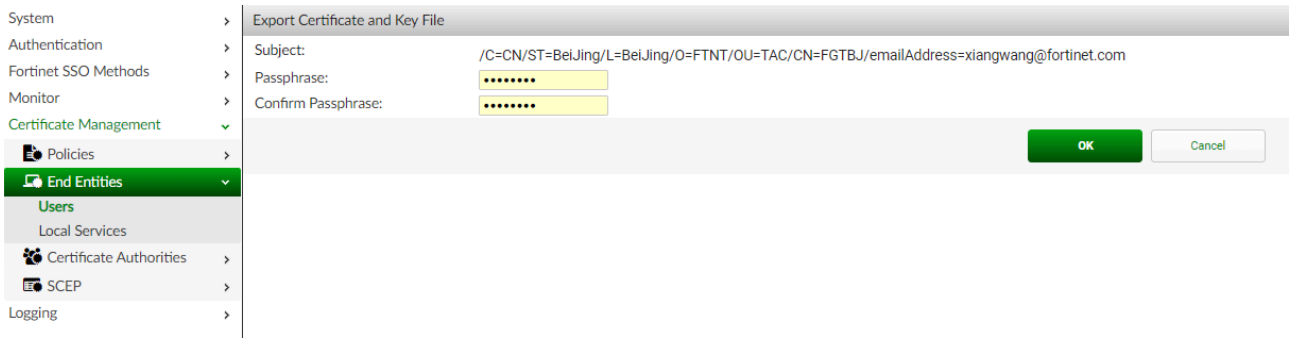




点击“Export Key and Cert”导出 FGTBJ 证书的私钥和公钥。



输入一个密码保护私钥

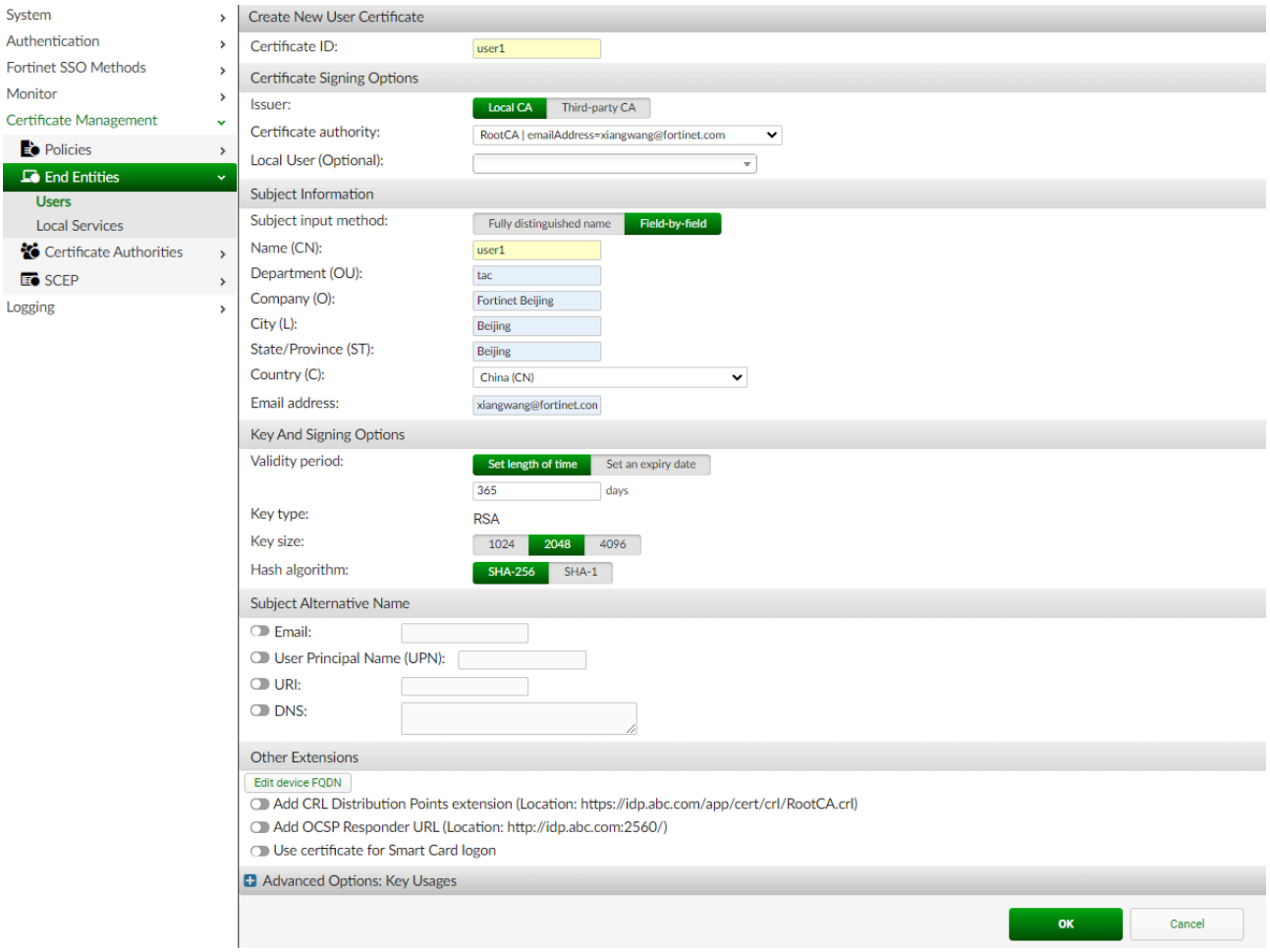


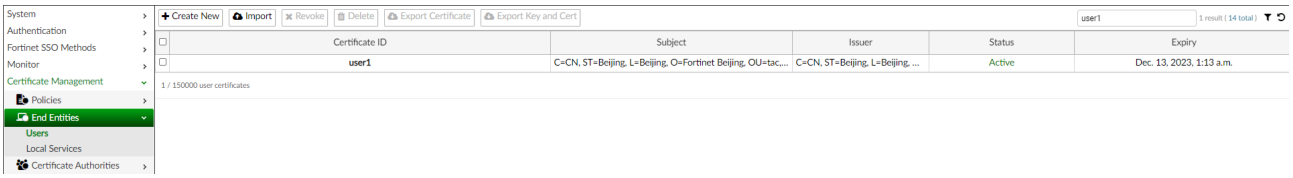
下载证书



2.为终端颁发证书

为终端颁发证书和上述步骤是一样的。

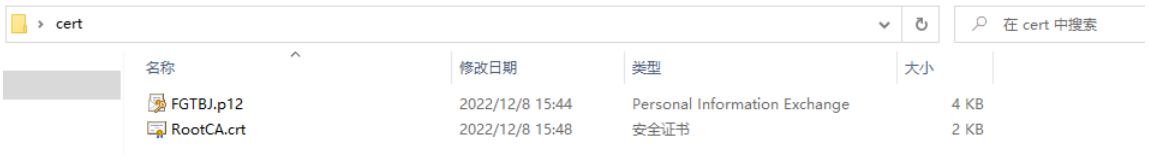




FortiGate 导入证书

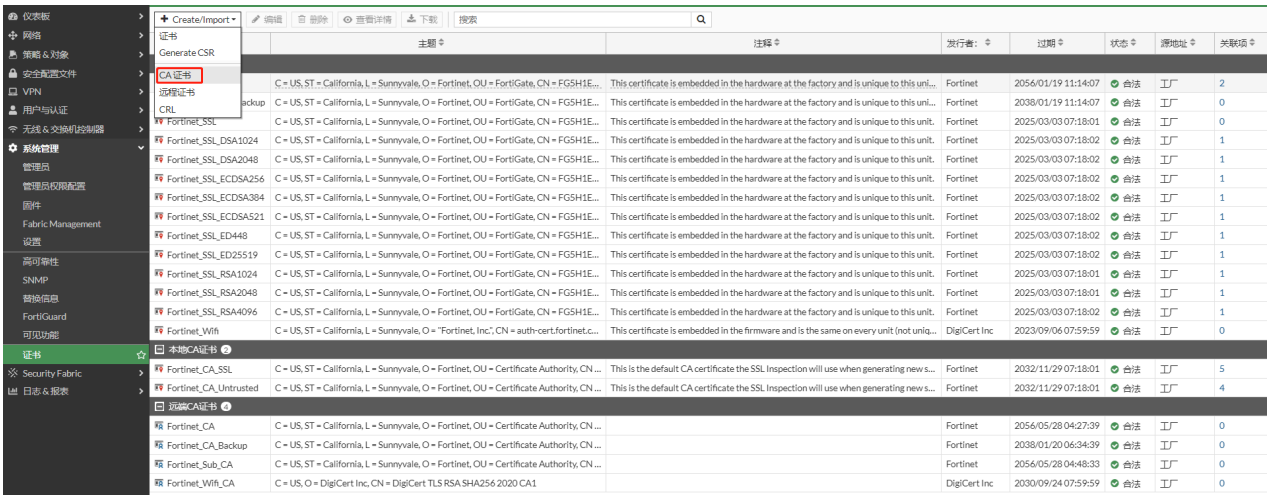
将根证书 RootCA.crt 和证书 FGTBJ.p12 导入到 FGT_BJ。

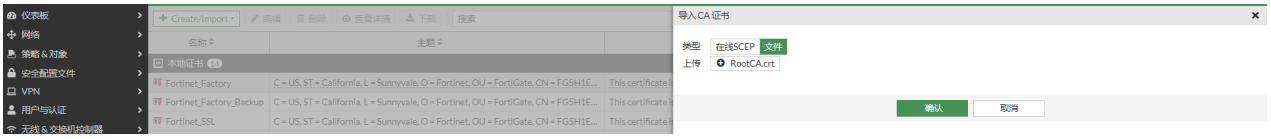
在数字证书认证中，通信双方使用 CA 证书进行数字证书合法性验证，双方各有自己的公钥（网络上传输）和私钥（本地存储）。发送方对报文进行 Hash 计算，并用自己的私钥对报文计算结果进行加密，生成数字签名。接收方使用发送方的公钥对数字签名进行解密，并对报文进行 Hash 计算，判断计算结果与解密后的结果是否相同。如果相同，则认证通过；否则认证失败。



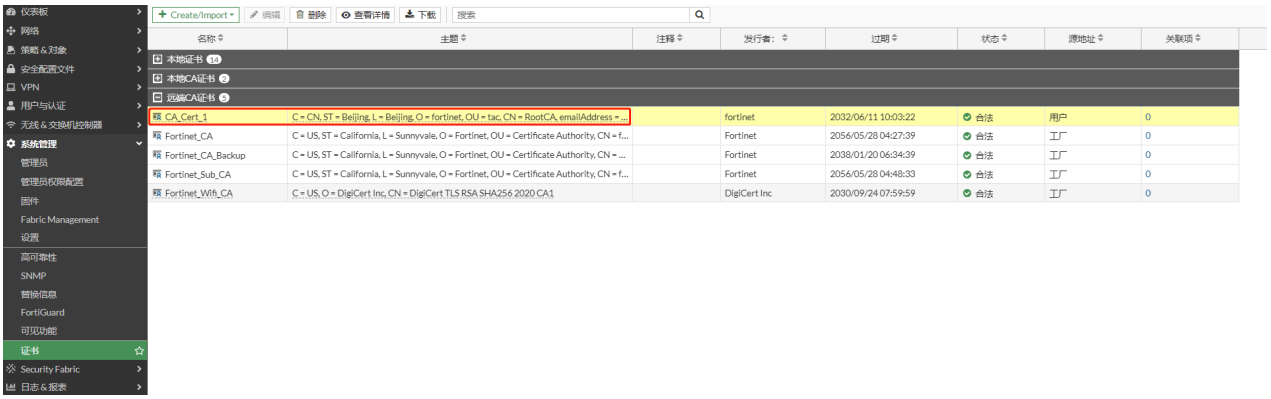
1. 导入 CA 证书

选择“系统管理”-->“证书”，点击“Create/Import”，选择 CA 证书。



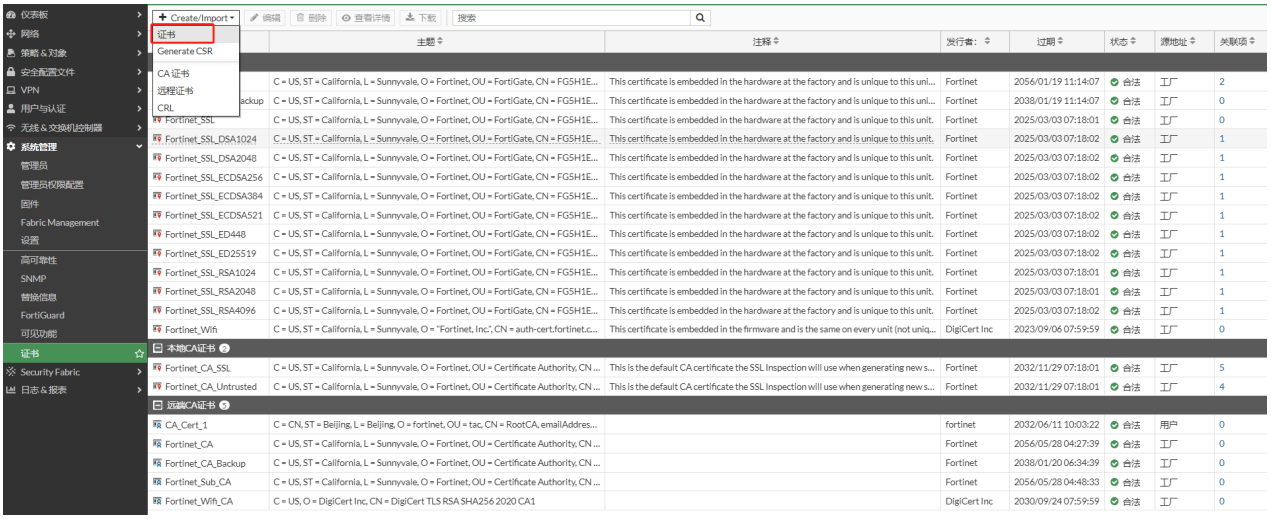


查看导入的 CA 证书

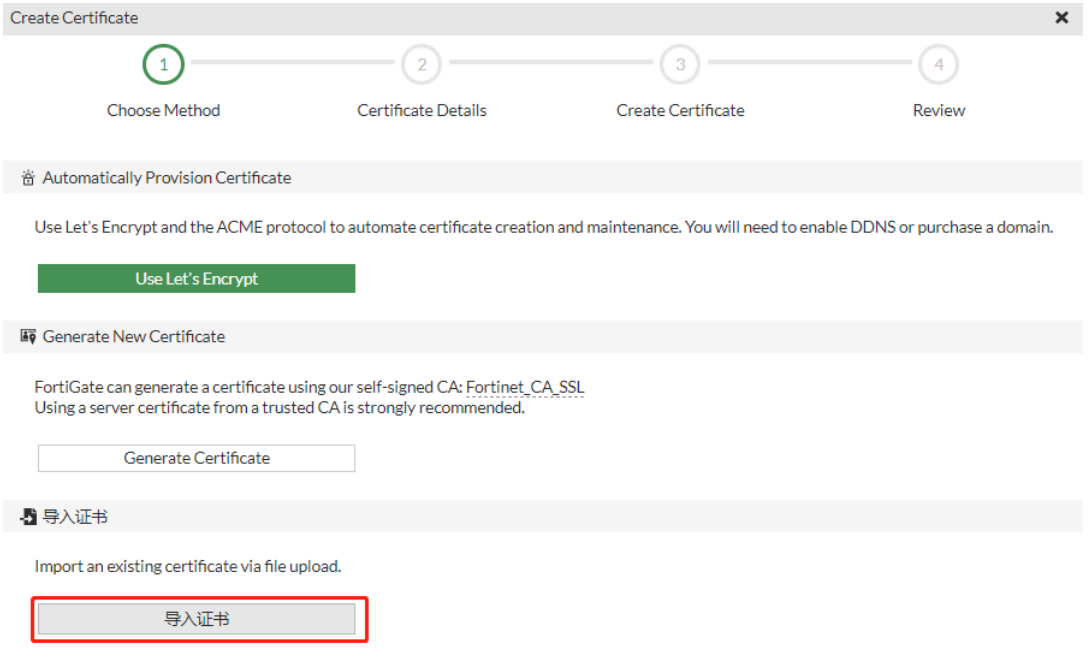


2. 导入本地证书

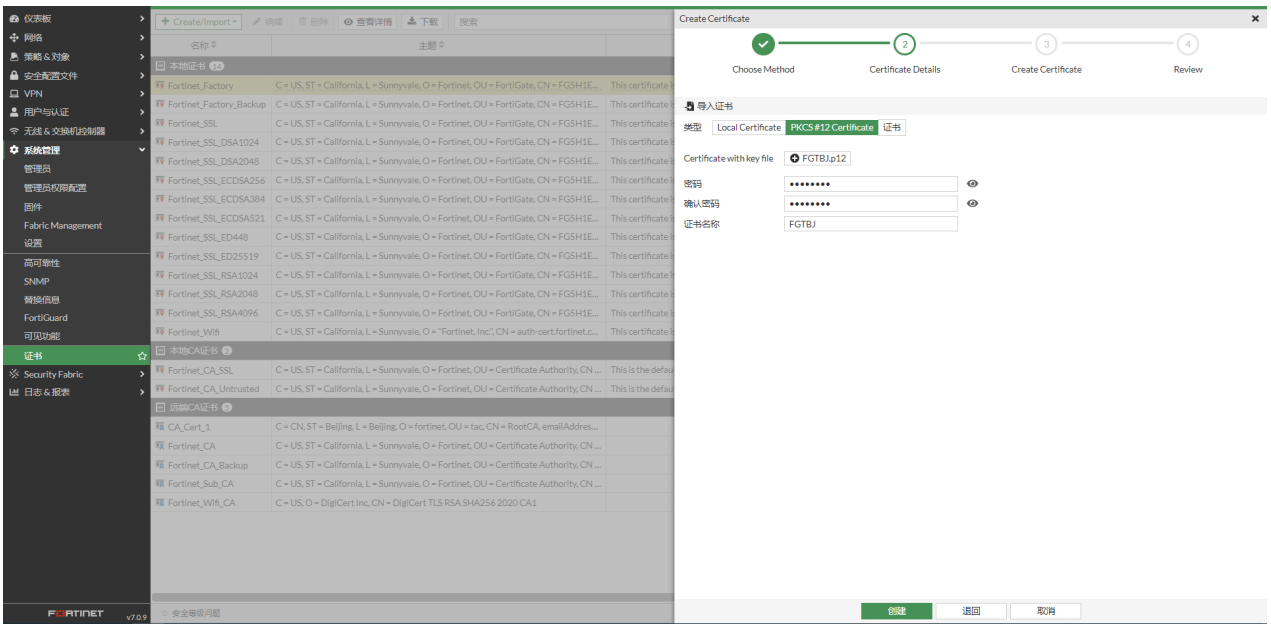
选择“系统管理”-->“证书”，点击“Create/Import”，选择“证书”。



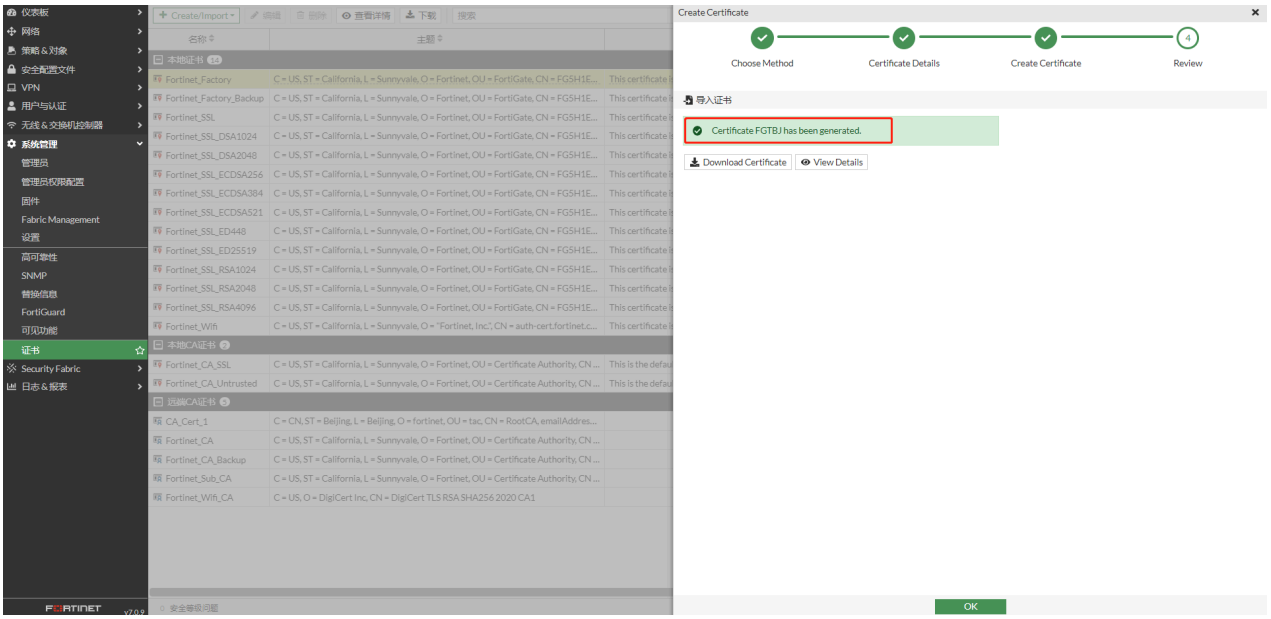
点击导入证书



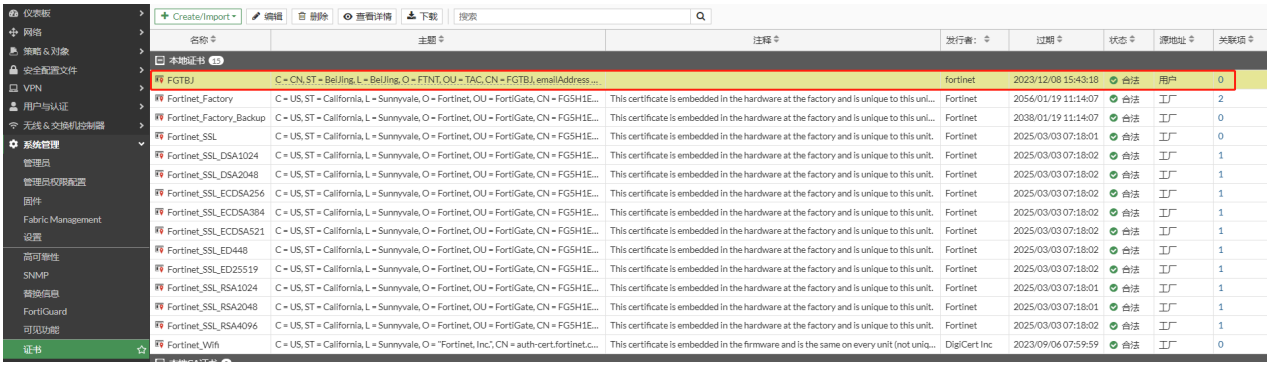
输入证书的密码，点击“创建”



提示证书已成功导入



查看导入的本地证书



终端导入证书

将根证书 RootCA.crt 和证书 user1.p12 导入到终端。



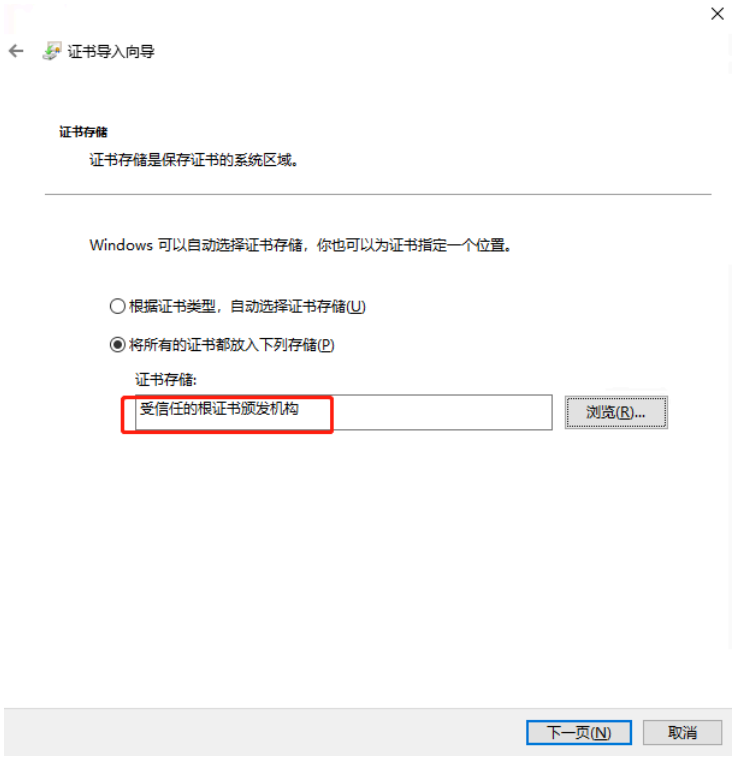
1. 导入根证书 RootCA.crt

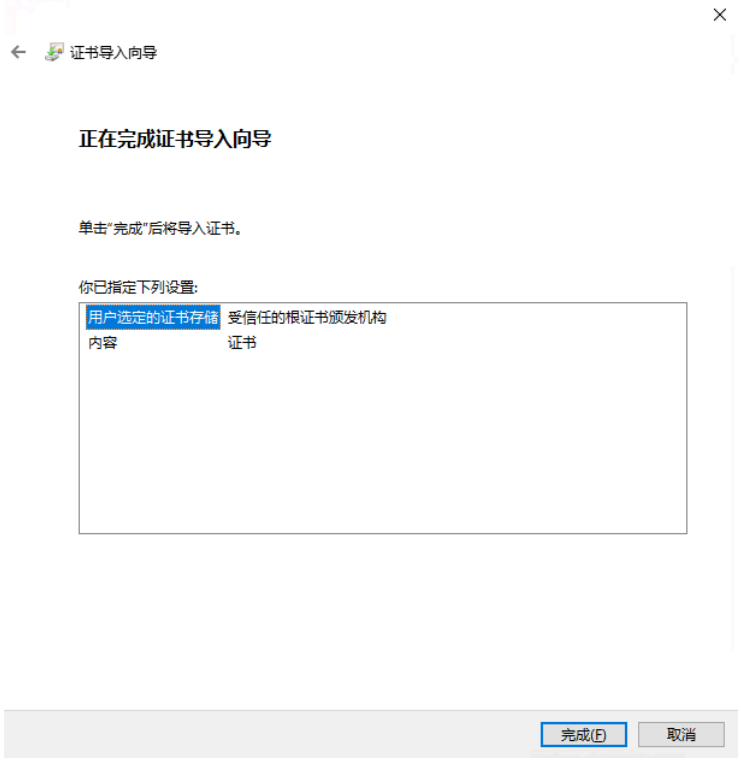
   双击 RootCA.crt，点击“安装证书”。





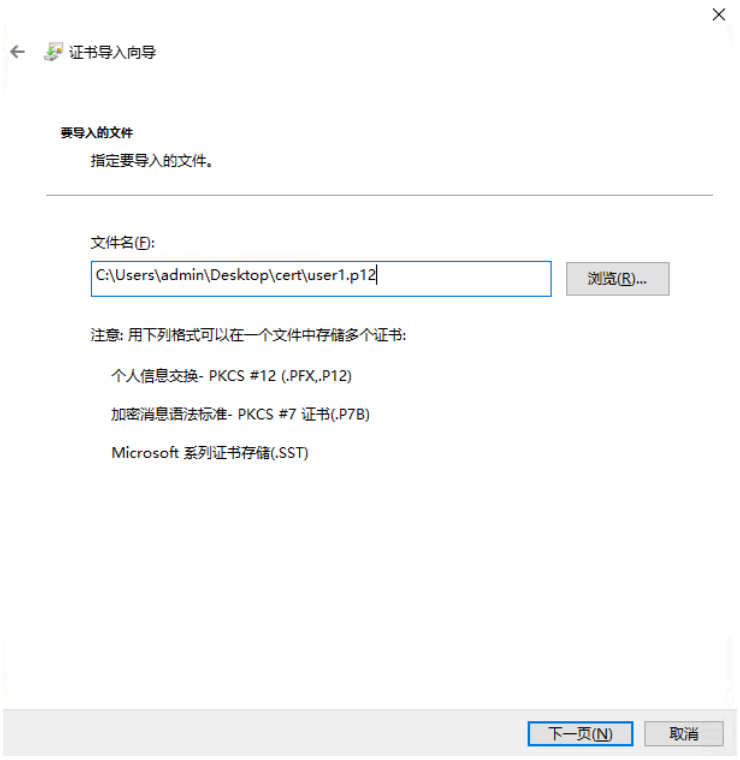
证书存入“受信任的根证书颁发机构”





2. 导入个人证书 user1.p12

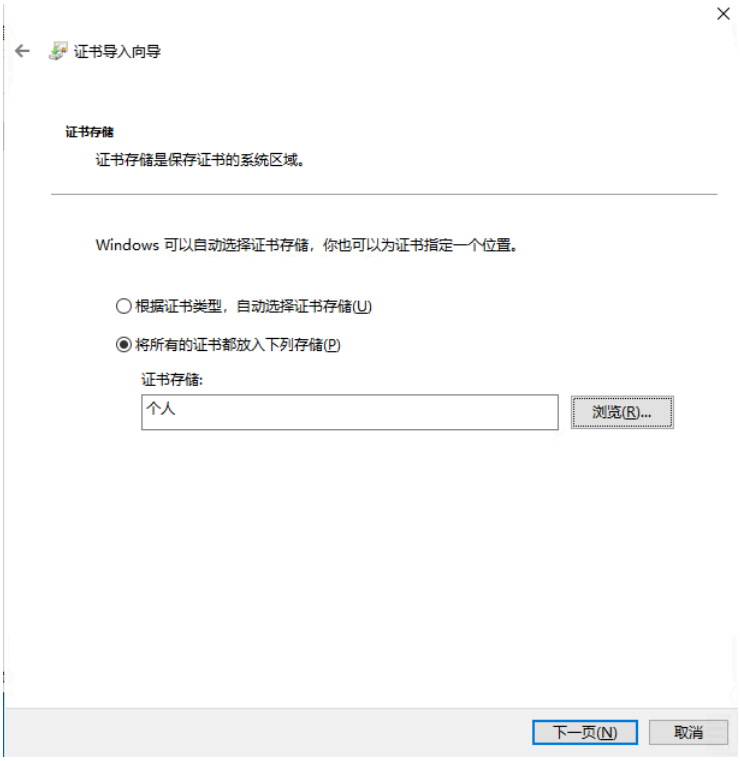
   双击 user1.p12，然后安装证书

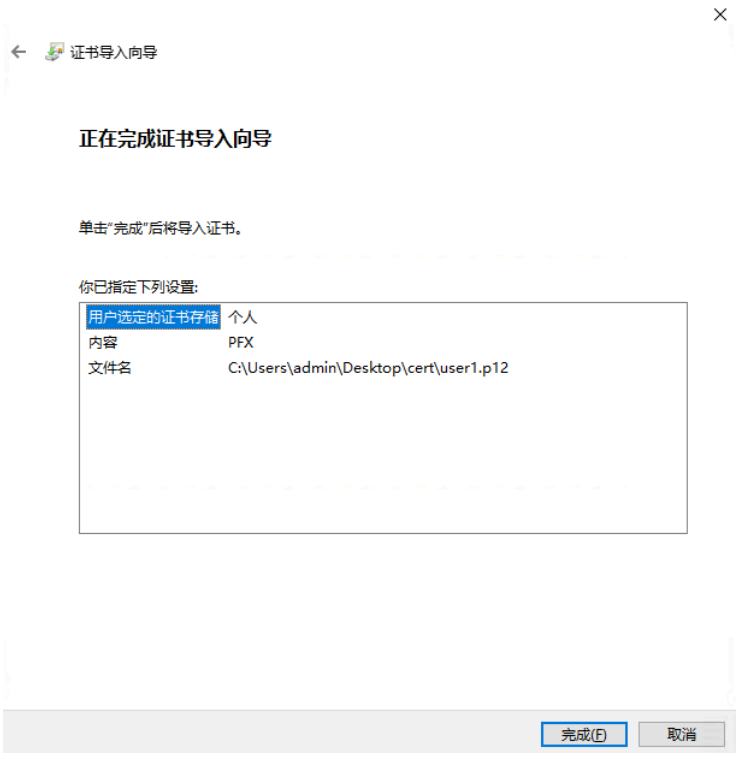


输入证书的密码



证书存入“个人”

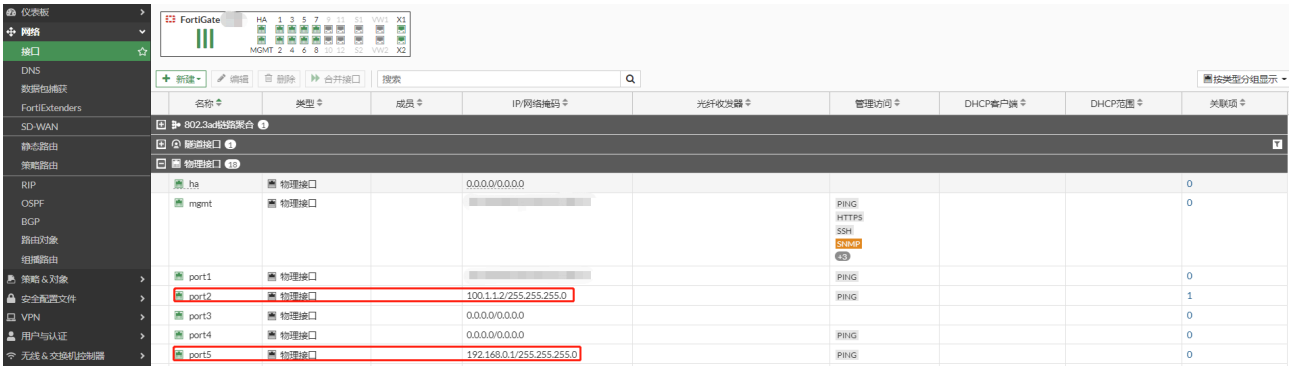


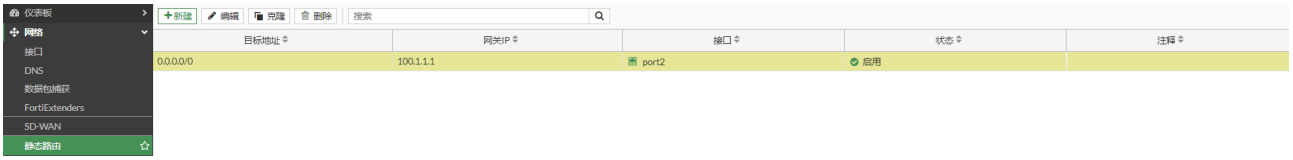


IPSEC VPN 配置步骤

1.基本配置

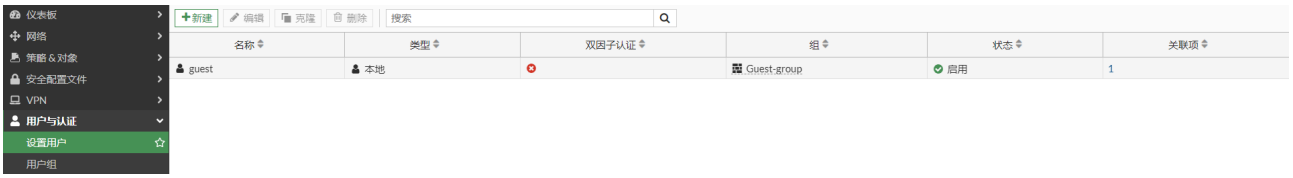
配置接口 IP 和路由





2.创建用户

选择“用户与认证”-->“设置用户”，点击“新建”。



选择“本地用户”，点击“下一步”。



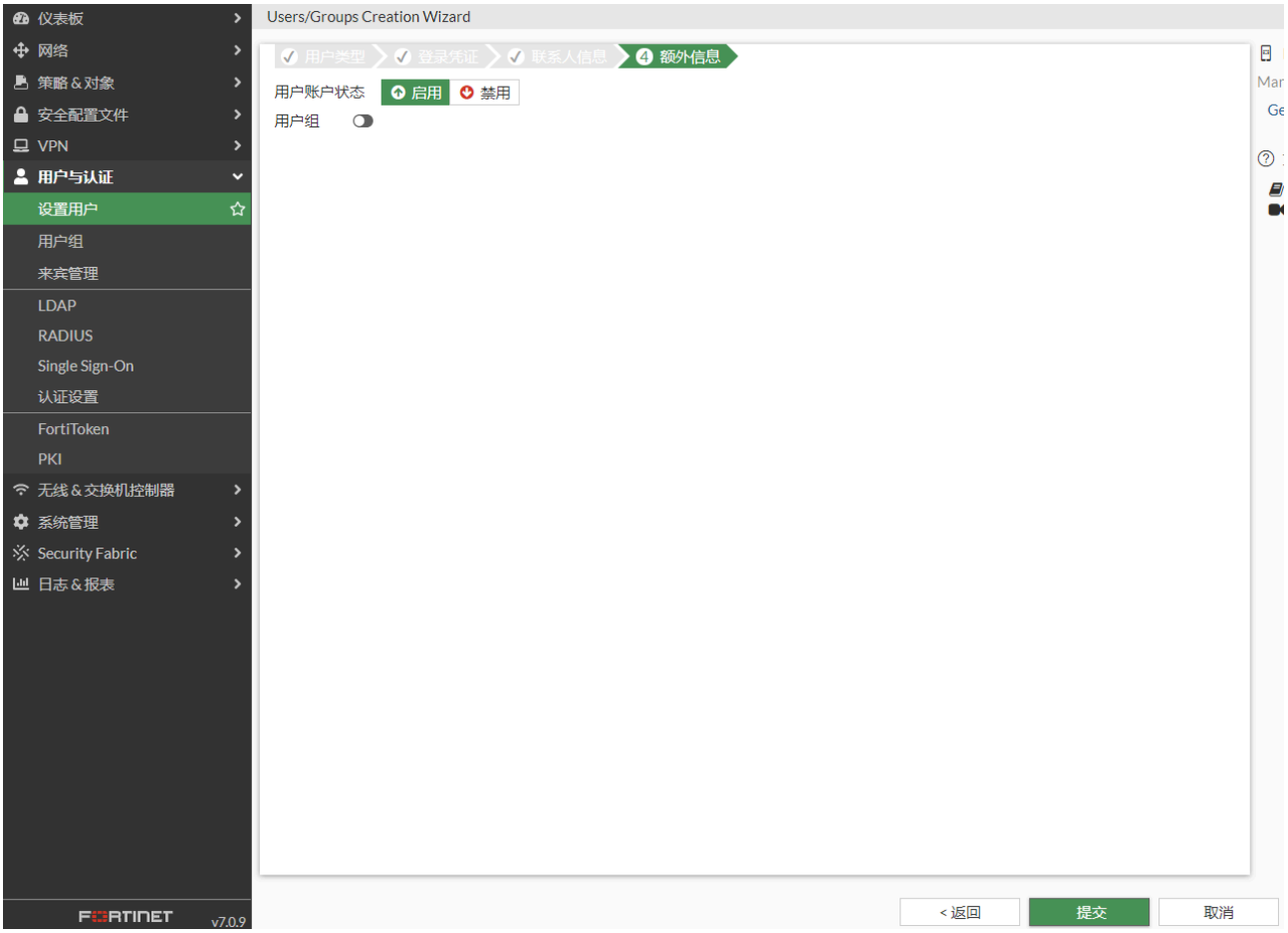
输入用户名和密码，点击“下一步”。



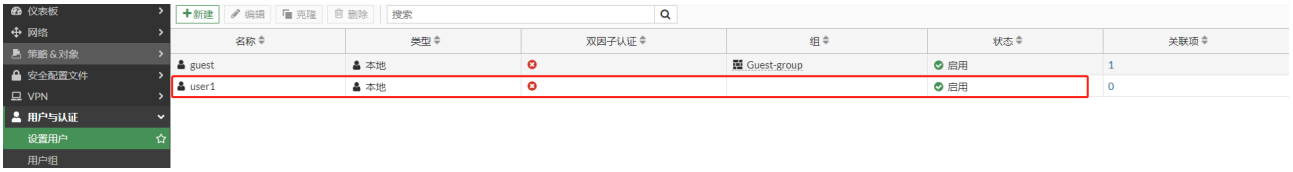
可根据需求选择启用。这里不启用，点击“下一步”。



点击“提交”。



完成用户创建。

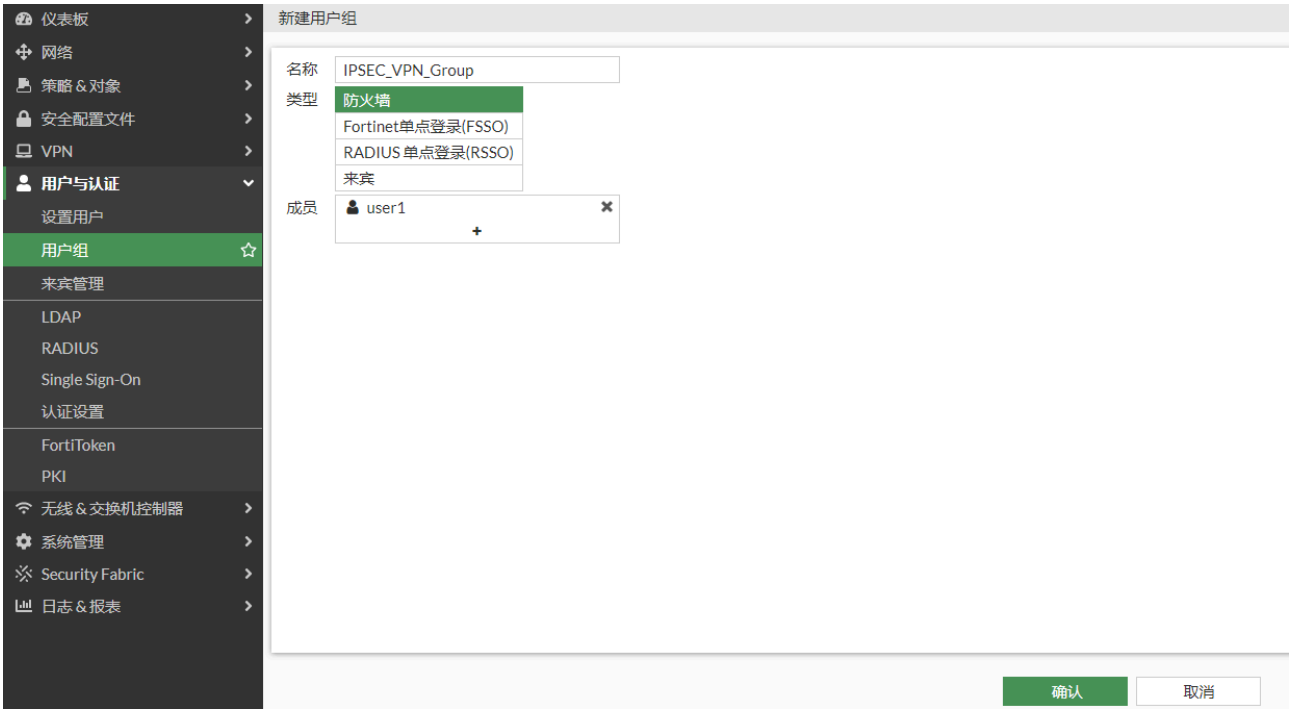


3.创建用户组

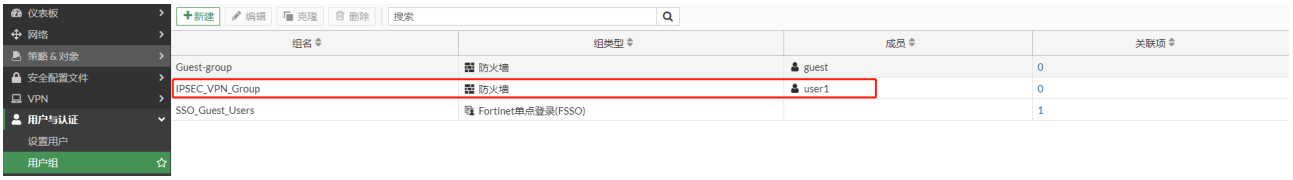
选择“用户与认证”-->“用户组”，点击“新建”。



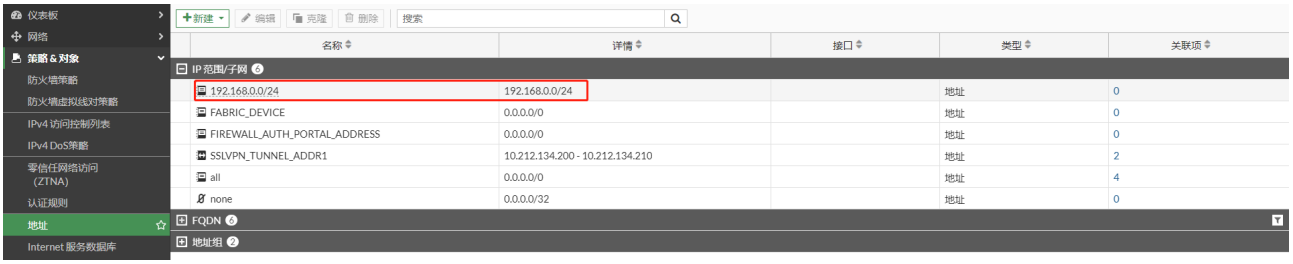
输入名称，即组名，并将用户加入用户组，点击“确认”。



完成用户组创建。

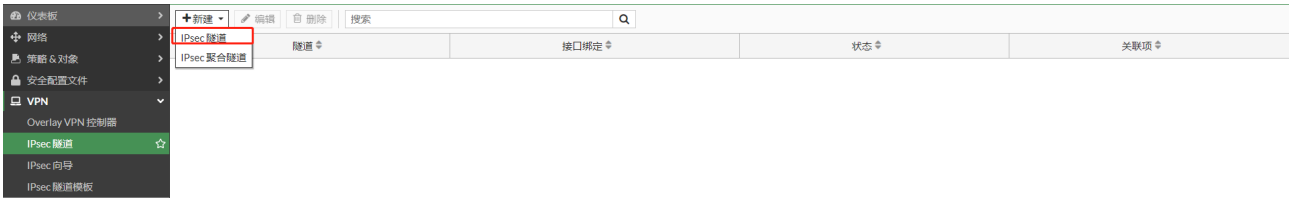


4.创建子网



5.配置 IPSEC VPN

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



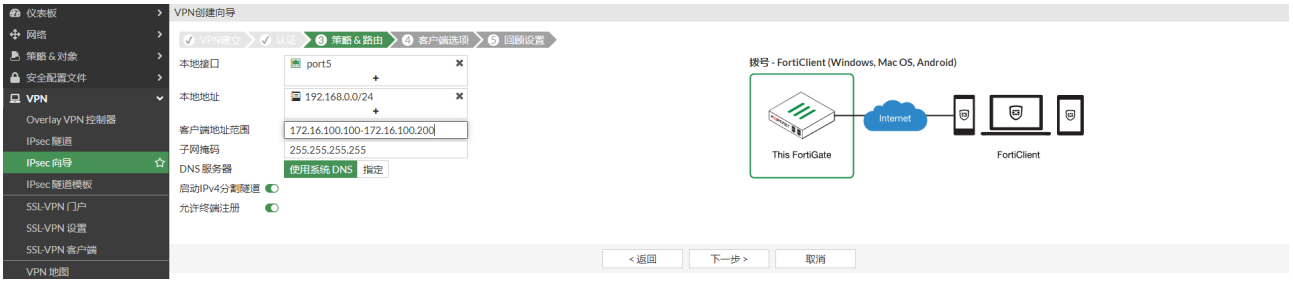
根据“VPN 创建向导”进行 VPN 模板配置，输入名称，选择”远程接“-->"基于客户端"-->“FortiClient”，并点击下一步。



选择对外的接口，用户组，证书名是本地的证书 FGTBJ，对端证书 CA 是导入的 RootCA，即 CA_Cert_1，并点击下一步。



输入内网接口，允许访问的本地地址段，客户端拨号后获取的地址范围，并点击下一步。



保存密码，FortiClient 第一次拨号成功后，FortiClient 会显示“保存密码”的选项，默认勾选。

自动连接，FortiClient 第一次拨号成功后，FortiClient 会显示“自动连接”的选项，该功能是运行 FortiClient 会拨入该 IPSEC VPN。

保持存活，FortiClient 第一次拨号成功后，FortiClient 会显示“保持连接”的选项，该功能是 IPSEC VPN 由于网络原因中断后，会自动重拨。

免费的 FortiClient 版本不支持自动连接和保持连接，需要完整版的 FortiClient，需要购买 EMS license。



VPN 创建向导提示即将创建的内容，然后点击完成。



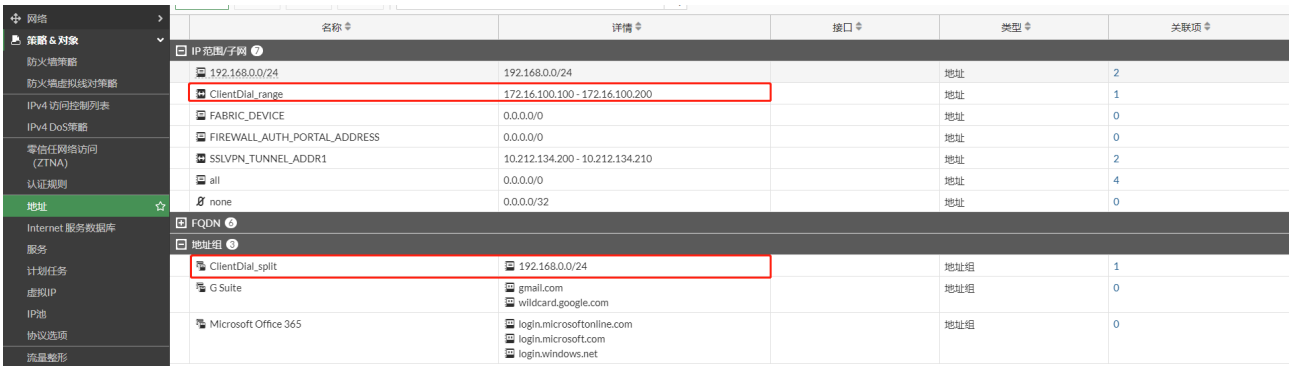
VPN 创建成功。



查看 IPSEC 向导创建的配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建地址对象和地址对象组。



2.创建 PKI 对象

```
config user peer
    edit "ClientDial_peer"
        set ca "CA_Cert_1"
    next
end
```

3.创建 IPSEC VPN



对应的命令行

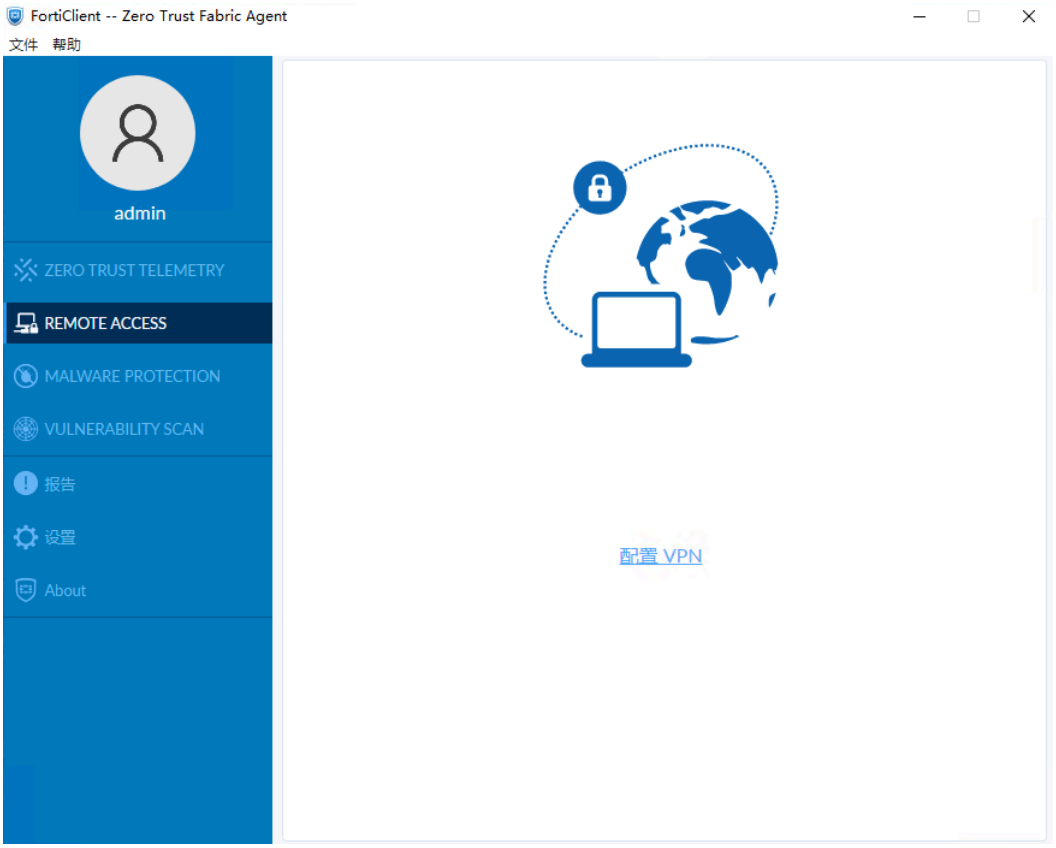
```
config vpn ipsec phase1-interface
    edit "ClientDial"
        set type dynamic
        set interface "port2"
        set authmethod signature
        set mode aggressive
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: ClientDial (Created by VPN wizard)"
        set wizard-type dialup-forticlient
        set xauthtype auto
        set authusrgrp "IPSEC_VPN_Group"
        set certificate "FGTBJ"
        set peer "ClientDial_peer"
        set ipv4-start-ip 172.16.100.100
        set ipv4-end-ip 172.16.100.200
        set dns-mode auto
        set ipv4-split-include "ClientDial_split"
        set save-password enable
    next
end
config vpn ipsec phase2-interface
    edit "ClientDial"
        set phase1name "ClientDial"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set comments "VPN: ClientDial (Created by VPN wizard)"
    next
end
```

4.创建策略



配置 FortClient

1. 选择“Remote Access”，点击“配置 VPN”。

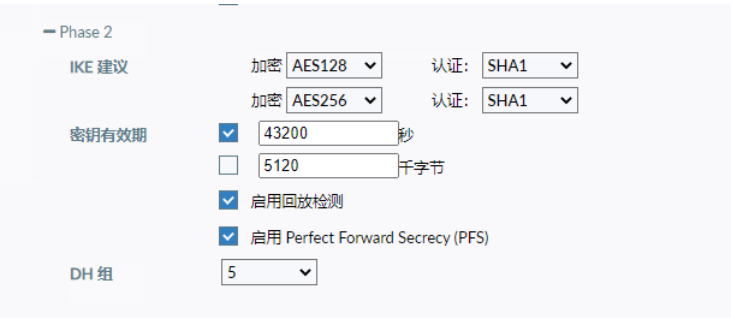


2. 选择 IPSEC VPN，输入连接名，远程网关，个人证书，以及用户名，然后点击“保存”。

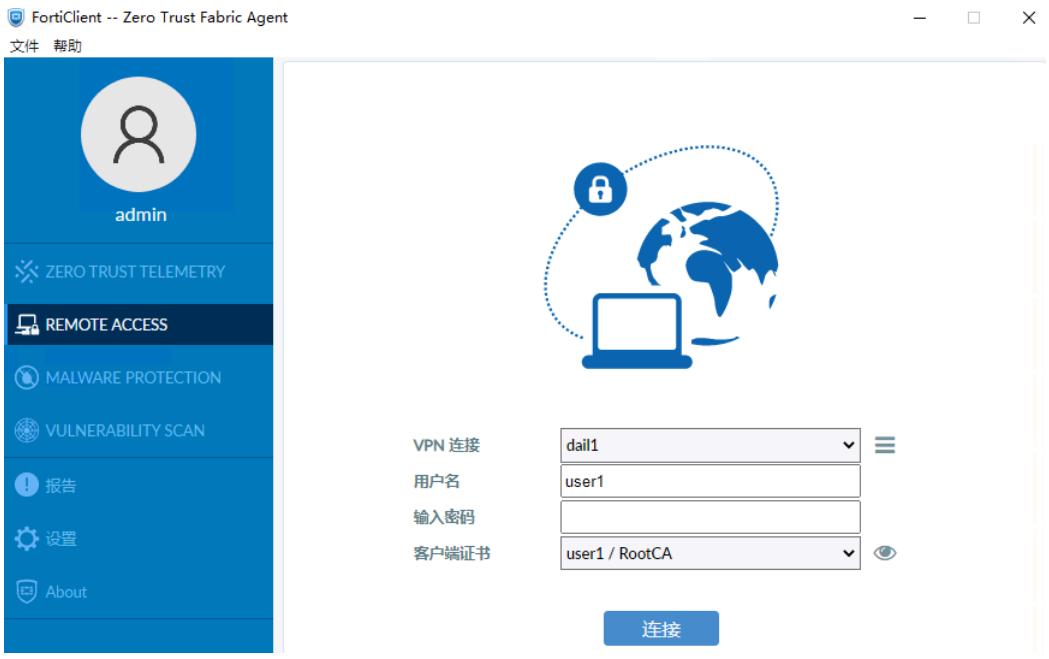


3. 在”高级设置“有更多的选项，如 ike 版本，IKE 阶段一和阶段二的加密集等，可以根据 FortiGate 的设置做对应的修改。





3. VPN 配置完成

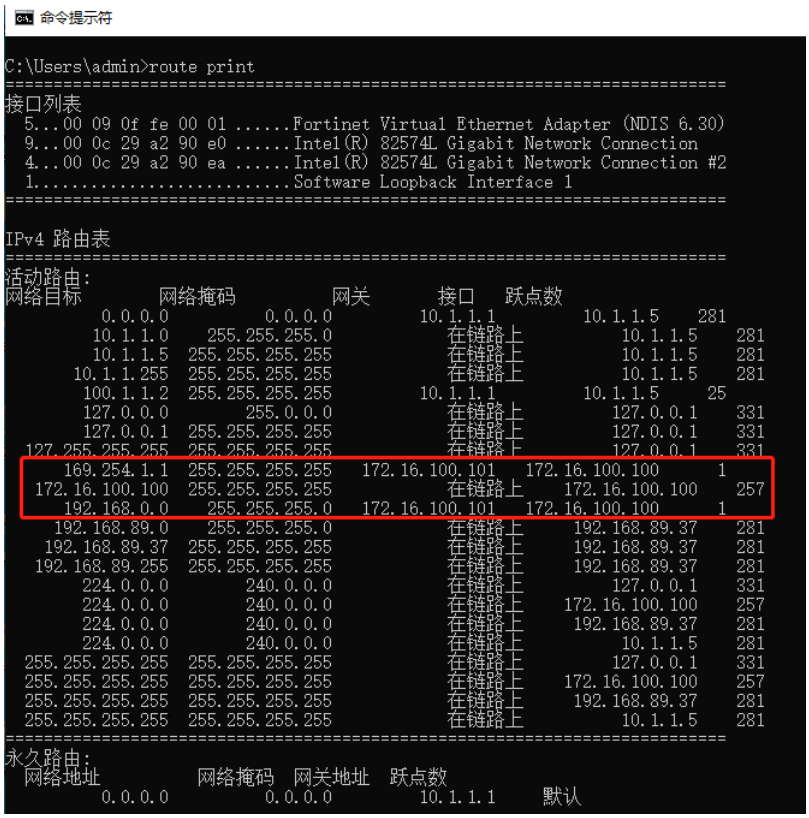


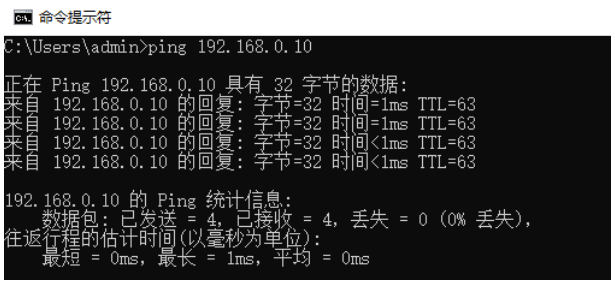
FortClient 拨号测试

1. 在 FortiClient 中输入账号密码，点击“连接”，FortiClient 拨号成功。

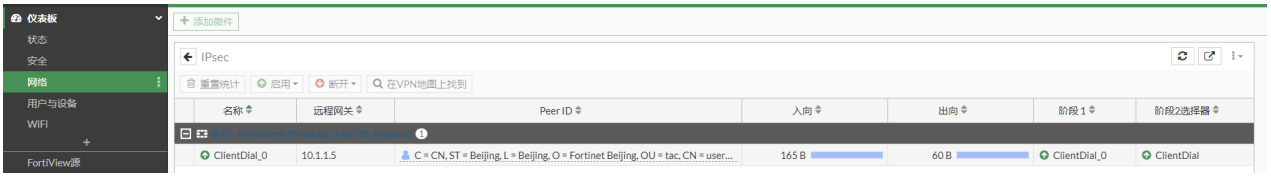


2. 查看终端路由表，192.168.0.0/24 指向 IPSEC VPN，并能成功访问 OA Server。

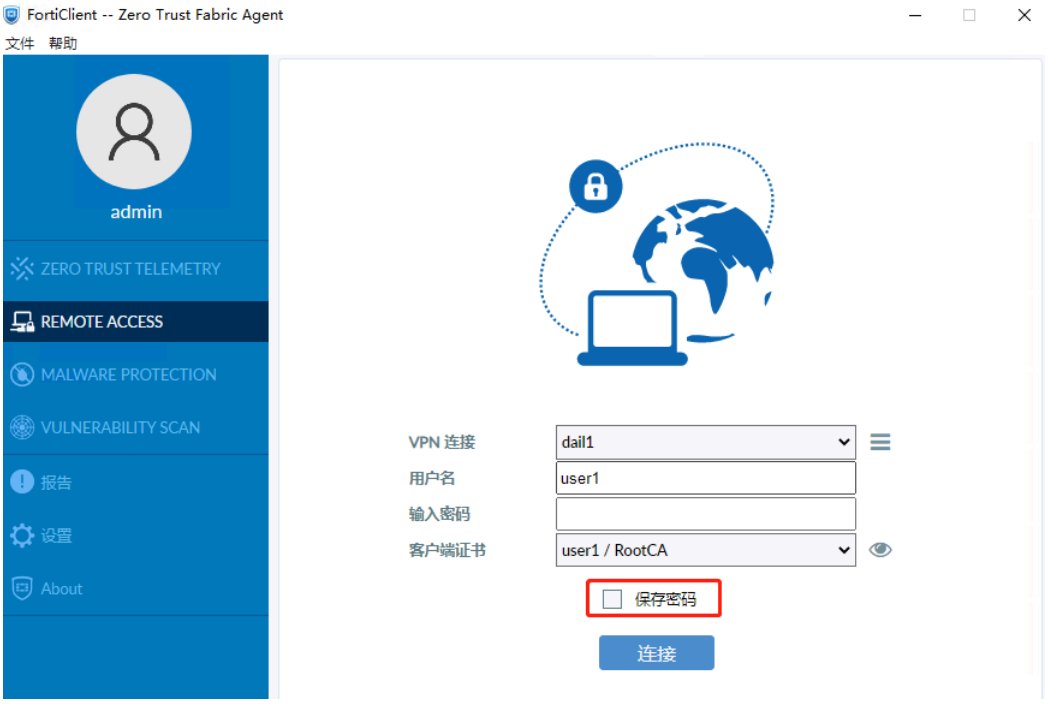




3. FortiGate 查看 Client 的连接。



4. 点击“中断连接”，FortiClient 就会显示“保存密码”选项。



基于 TCP 的 FortiClient 拨号

重要

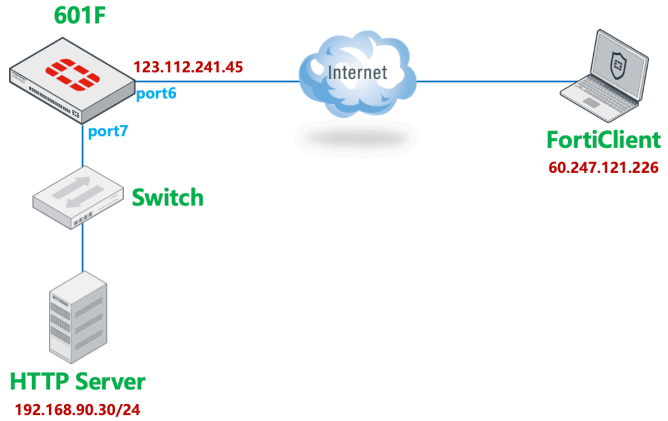
- 该功能在 FortiOS 7.4.2、7.6.0 及以上版本支持。
- 该功能要求 FortiClient 版本≥7.4.1。
- 该功能仅支持 IKEv2。
- 使用基于 TCP 的 FortiClient IPSec 拨号模式，FortiGate 暂不支持 ESP 流量的 NP 加速（但可以被 CP 芯片加速）。
- 在 FortiClient 7.4.4 及以后版本中，FortiClient VPN only 版本（免费）将不再支持此功能，FortiClient ZTNA 版本（需要 EMS License）仍然支持此功能。

功能介绍

传统拨号 IPsec VPN 流量使用 UDP 或 ESP 协议，在 FortiOS 7.4.3、7.6.0 及以上版本可以通过 TCP 运行。出于以下考量，我们推荐将 FortiClient 拨号模式的 SSL VPN/IPSec over UDP 逐渐迁移至 IPSec over TCP。

- 从 FortiOS 7.6.3 开始，FortiGate 在所有型号上移除了 SSL VPN Tunnel 模式，需要使用 IPSec VPN 拨号模式替代 SSL VPN 的 Tunnel 模式（详情参考：VPN 技术 → SSL VPN → 开启 SSL VPN ↗ 中的“SSL VPN 移除情况”部分）。
- 在高延迟或拥塞的网络中，基于 UDP 的 IPSec VPN 连接可能会出现数据包丢失或性能下降。TCP 具有错误检测和重传机制，可提高 IPSec VPN 连接在此类环境中的可靠性和稳定性。
- 部分运营商可能对 UDP 500/4500、ESP 流量实施策略性限速（如降低 QoS 优先级或设置带宽阈值）或阻断，尤其在高峰时段或跨境链路上。通过将 IPSec VPN 切换至 TCP 传输，可规避此类 UDP 限速策略，利用 TCP 的标准化流量特征获得更稳定的带宽分配，从而保障 VPN 隧道的吞吐量。

网络拓扑



- FortiGate-601F（NP7）配置基于 TCP 的 IKEv2 拨号 IPsec 服务器，自定义 TCP 端口为 4443。
- IKEv2 使用 EAP 进行用户身份验证。
- 客户端使用 FortiClient 7.4.3 版本，配置 TCP 目标端口 4443 连接 FortiGate 的 IPsec。

基础配置

用户配置

1. 创建用于 IPSec 拨号认证的用户和用户组。

```bash
config user local
    edit "user1"
        set type password
        set passwd 1q2w3e4r
    next
    edit "user2"
        set type password
        set passwd 1q2w3e4r
    next
end
```

```
config user group
    edit "VPN"
        set member "user1" "user2"
    next
end
```

2. 配置用于隧道分割的地址对象，可以配置多个。

```
config firewall address
    edit "192.168.90.0/24"
        set subnet 192.168.90.0 255.255.255.0
    next
end
```
bash

指定基于 TCP 的 IPSec 端口

1. 指定基于 TCP 的 IPSec VPN 使用的端口（`ike-tcp-port`），这里配置为 TCP 4443。

> **重要**
>
> - 默认 `ike-tcp-port` 配置：
>   - FortiOS 7.4：4500。
>   - FortiOS 7.6：443。
> - 修改 `ike-tcp-port` 选项是基于当前 VDOM 生效的，该 VDOM 内所有基于 TCP 的 IPSec 连接都将使用指定端口封装 IKE 协商包和 IPSec ESP 报文。若需同时支持其他使用不同 TCP 端口的 IPSec 连接（如与其他 FortiGate 之间），建议通过新建 VDOM 实现。
> - 修改 `ike-tcp-port` 选项不会影响基于 UDP 的 IPSec 连接端口（`ike-port`，默认 500）。

```
config system settings
    set ike-tcp-port 4443
end
```
bash

2. 如果使用 TCP 443 端口，可能会与 FortiGate 的 HTTPS 管理端口或 SSL VPN 端口冲突，需要修改 HTTPS 管理端口/SSL VPN 为其他端口，IPSec 的 TCP 443 端口才能生效。

```
HTTPS管理端口：
config system global
    set admin-sport <port>
end

SSL VPN端口：
config vpn ssl settings
    set port <port>
end
```
bash

FortiGate IPSec 配置

> **重要**
>
> 通过 GUI 的配置步骤基于 FortiOS 7.6.3，如果是 7.4.X 版本，请参考通过 CLI 命令配置的部分。

通过 GUI 配置（FortiOS 7.6）

1. 在 FortiGate 进入“VPN → VPN 隧道”页面，点击“新建”按钮，选择“来自模板的 IPSec 隧道”。



2. 在弹出的“VPN 向导”中，填写隧道名称，模板选择“远程访问”类型，然后点击“开始”按钮。



3. 在“VPN 隧道”页面填写“预共享密钥”，传输模式选择“TCP 封装”，用户身份验证方法选择“阶段 1 接口”，选择已创建的 VPN 用户组，其他保持默认，点击“下一个”按钮。

重要

可选配置：

- 传输：
  - TCP 封装：强制使用 TCP 封装 IKE 协商报文与 IPSec ESP 报文，TCP 端口会使用已配置的 `ike-tcp-port`。
  - 自动：优先使用 UDP 封装 IKE 协商报文，如果 UDP 连接超时（默认超时时间为 15s）。
- 使用 Fortinet 封装：
  - 开启：使用 Fake TCP 头部封装 ESP（RFC8229），仅适用于 FortiGate 之间建立的 IPSec。
  - 关闭：使用标准 TCP 头部封装 ESP（RFC9293），使用 FortiClient 连接时，一定要关闭此选项。
- 用户身份验证方法：
  - 阶段 1 接口：在 IPSec 阶段 1 配置中手动选择需要认证的用户组，只能选择一个用户组。
  - 从策略继承：在 IPSec 相关的防火墙策略中指定需要认证的用户组，适用于有多个用户组的情况。
- DNS 服务器：
  - 使用系统 DNS：分配系统配置的 DNS 服务器给 FortiClient。
  - 指定：手工指定给 FortiClient 分配的 DNS 服务器。 :::



4. 在“远程端点”页面填写给 FortiClient 分配的地址池，关闭“EMS SN 验证”，点击“下一个”按钮。



5. 在“本地 FortiGate”页面选择用于接入 IPSec 连接的接口 port6，IPSec 拨号访问的目的接口 port7，“本地地址”选择需要隧道分离的地址对象“192.168.90.0/24”（可选多个），点击“下一个”按钮。



6. Comment 页面提示会自动下发的地址对象（分配的地址池）、地址组（包含隧道分割的地址对象）、策略（远程到本地放通）的名称，可以点右侧的笔修改名称。



7. IPSec VPN 连接创建完成。



8. 查看通过向导自动创建的地址和地址组对象。

```bash
地址（分配的IP地址池）：
config firewall address
    edit "TCP_Dialup_range"
        set type iprange
        set comment "VPN: TCP_Dialup -- Created by VPN wizard"
        set start-ip 10.100.1.100
```

```
        set end-ip 10.100.1.200
    next
end

地址组（隧道分割网段）：
config firewall addrgrp
    edit "TCP_Dialup_split"
        set member "192.168.90.0/24"
        set comment "VPN: TCP_Dialup -- Created by VPN wizard"
    next
end
```

9. 查看通过向导创建的 IPSec 隧道一阶段和二阶段 CLI 配置。

```bash
config vpn ipsec phase1-interface
    edit "TCP_Dialup"
        set type dynamic
        set interface "port6"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set comments "VPN: TCP_Dialup -- Created by VPN wizard"
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set authusrgrp "VPN"
        set transport tcp
        set ipv4-start-ip 10.100.1.100
        set ipv4-end-ip 10.100.1.200
        set dns-mode auto
        set ipv4-split-include "TCP_Dialup_split"
        set save-password enable
        set psksecret ENC
iEMCQpp6y6RY3YqsG30txAPygrAppFQYI9fCj5fb/3SVQKJv4rQ4AkexsTC+z3KnXxUpDygDUCc8znjJQnx06jI0CKMHOR++uVgfEOAoMwwfnNQCHrCBO+I4
79zBKhDC11kWOGp7IfOX2vWEsZ4LH/MXIuP+vzvD70RqPthkGGz+bnKMgjsr0fkVUwUghtV6PryzeFlmMjY3dkVA
    next
end

config vpn ipsec phase2-interface
    edit "TCP_Dialup"
        set phase1name "TCP_Dialup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: TCP_Dialup -- Created by VPN wizard"
    next
end
```

.0. 查看通过向导自动创建的防火墙策略，默认开启 SNAT，可根据实际需求调整配置。

```bash
config firewall policy
    edit 22
        set name "vpn_TCP_Dialup_local_allow"
        set srcintf "TCP_Dialup"
        set dstintf "prot7"
        set action accept
        set srcaddr "TCP_Dialup_range"
        set dstaddr "192.168.90.0/24"
        set schedule "always"
        set service "ALL"
        set nat enable
        set comments "VPN: TCP_Dialup -- Created by VPN wizard"
    next
end
```

通过 CLI 配置（FortiOS 7.4）

1. 创建用于分配 IP 的地址对象和用于隧道分割的地址组对象。

```bash
地址（分配的IP地址池）：
config firewall address
    edit "TCP_Dialup_range"
        set type iprange
        set start-ip 10.100.1.100
        set end-ip 10.100.1.200
    next
end

地址组（隧道分割网段）：
config firewall address
    edit "192.168.90.0/24"
        set subnet 192.168.90.0 255.255.255.0
    next
end

config firewall addrgrp
    edit "TCP_Dialup_split"
        set member "192.168.90.0/24"
    next
end
```

2. 配置 IPSec VPN 一阶段。

- `type`：dynamic，拨号模式。
- `interface`：port6，选择用于接入 IPSec 连接的接口。
- `ike-verison`：2，只有 IKEv2 支持基于 TCP 的 IPSec 模式。
- `mode-cfg`：enable，开启为客户端分配 IP 和 DNS 功能。
- `proposal` / `dhgrp`：一阶段安全提议，与 FortiClient 配置一致即可。
- `eap`：enable，IKEv2 客户端认证使用 EAP 模式。
- `authusrgrp`：VPN，认证使用已创建的 VPN 用户组。
- `transport`：tcp，使用 TCP 封装 IKE 协商报文和 ESP 报文。
- `ipv4-start-ip` / `ipv4-end-ip`：为 FortiClient 分配 IP 的地址池。
- `dns-mode`：auto，使用系统 DNS 分配给 FortiClient。
- `ipv4-split-include`：选择需要隧道分离的地址组对象“TCP_Dialup_split”，这里既可以选择地址组，也可以选择地址对象，但只能选择一个，所以这里以地址组为例，可以包含多个地址对象。
- `save-password`：enable，允许 FortiClient 保存密码。
- `psksecret`：预共享密钥。

```bash
config vpn ipsec phase1-interface
    edit "TCP_Dialup"
        set type dynamic
        set interface "port6"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "VPN"
        set transport tcp
        set ipv4-start-ip 10.100.1.100
        set ipv4-end-ip 10.100.1.200
        set dns-mode auto
        set ipv4-split-include "TCP_Dialup_split"
        set save-password enable
        set psksecret 1q2w3e4r
    next
end
```

重要

可选配置：

- `transport`：
  - tcp：强制使用 TCP 封装 IKE 协商报文与 IPSec ESP 报文，TCP 端口会使用已配置的 `ike-tcp-port`。
  - auto：优先使用 UDP 封装 IKE 协商报文，如果 UDP 连接超时（默认超时时间为 15s）。
- `fortinet-esp`：
  - enable：使用 Fake TCP 头部封装 ESP（RFC8229），仅适用于 FortiGate 之间建立的 IPSec。
  - disable：使用标准 TCP 头部封装 ESP（RFC9293），使用 FortiClient 连接时，一定要关闭此选项。
- `authusrgrp`：
  - 配置用户组：在 IPSec 阶段 1 配置中手动选择需要认证的用户组，只能选择一个用户组。
  - 不配置：在 IPSec 相关的防火墙策略中指定需要认证的用户组，适用于有多个用户组的情况。
- `dns-mode`：
  - auto：分配系统配置的 DNS 服务器给 FortiClient。
  - manual：手工指定给 FortiClient 分配的 DNS 服务器，然后通过 `set ipv4-dns-server1 <ipv4_address>`、`set ipv4-dns-server2 <ipv4_address>` 等命令设置手工指定的 DNS 服务器。 :::

3. 配置 IPSec VPN 二阶段。

- `phase1name`：绑定的一阶段名称。
- `proposal` / `dhgrp`：二阶段安全提议，与 FortiClient 配置一致即可。

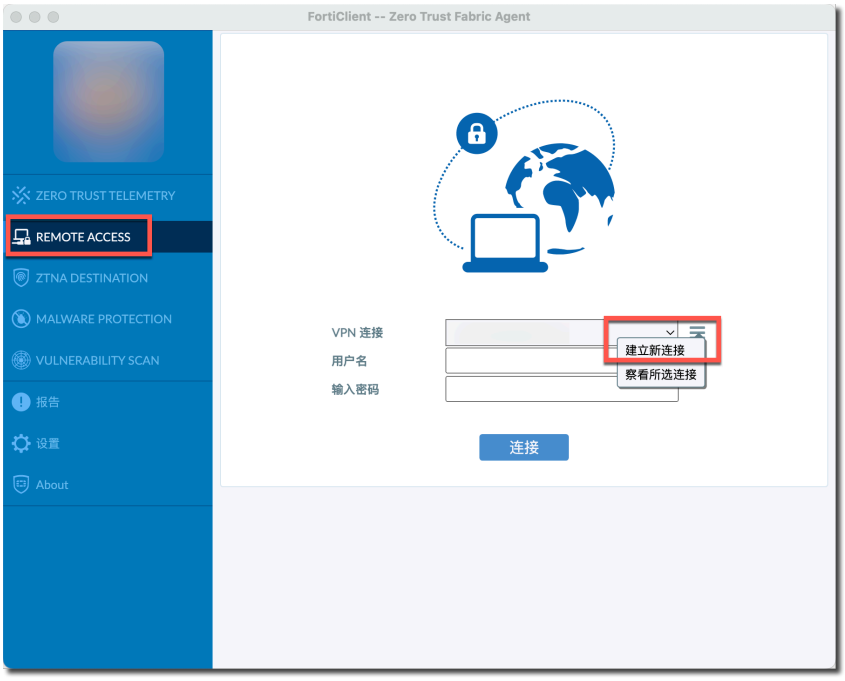
```bash
config vpn ipsec phase2-interface
    edit "TCP_Dialup"
        set phase1name "TCP_Dialup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

4. 创建防火墙策略，放通 FortiClient 到 VPN 资源的访问，可根据实际需求调整配置。

```bash
config firewall policy
    edit 22
        set name "vpn_TCP_Dialup_local_allow"
        set srcintf "TCP_Dialup"
        set dstintf "port7"
        set action accept
        set srcaddr "TCP_Dialup_range"
        set dstaddr "192.168.90.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

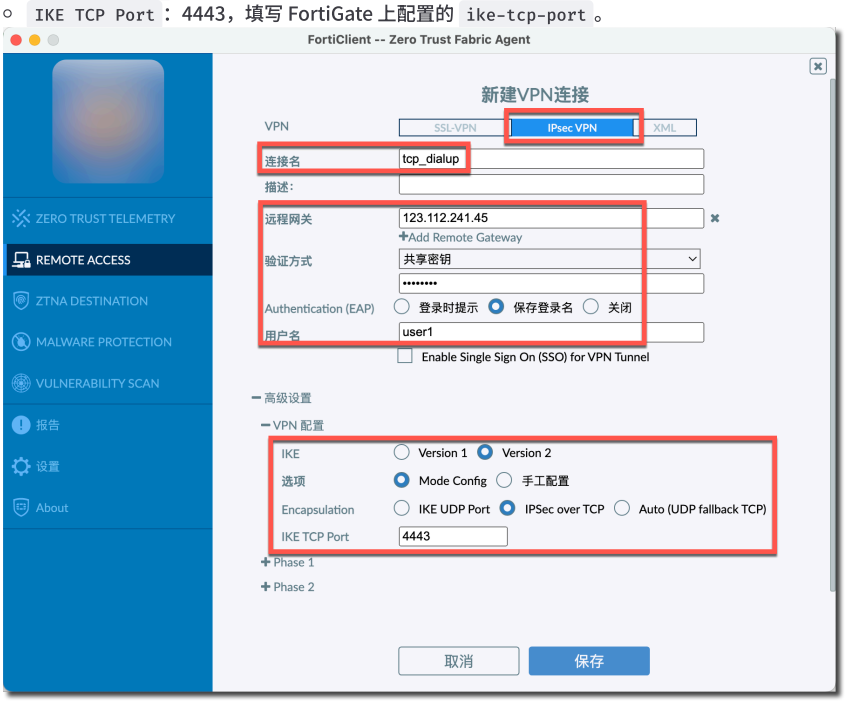
FortiClient 配置

1. 在 FortiClient 的“Remote Access”界面新建连接。



2. 配置 VPN 连接的基本信息和“高级设置”中的“VPN 配置”。

- `VPN类型`：IPSec VPN。
- 远程网关：填写 FortiGate 侧接入 IPSec 的 IP。
- 验证方式：共享密钥，并填入 FortiGate 配置的 IPSec 预共享密钥。
- `Authentication` / 用户名：保存登录名，填写用于 EAP 认证的用户名。
- `IKE`：Version2。
- 选项：Mode Config。
- `Encapsulation`：IPSec over TCP。
- `IKE TCP Port`：4443，填写 FortiGate 上配置的 `ike-tcp-port`。

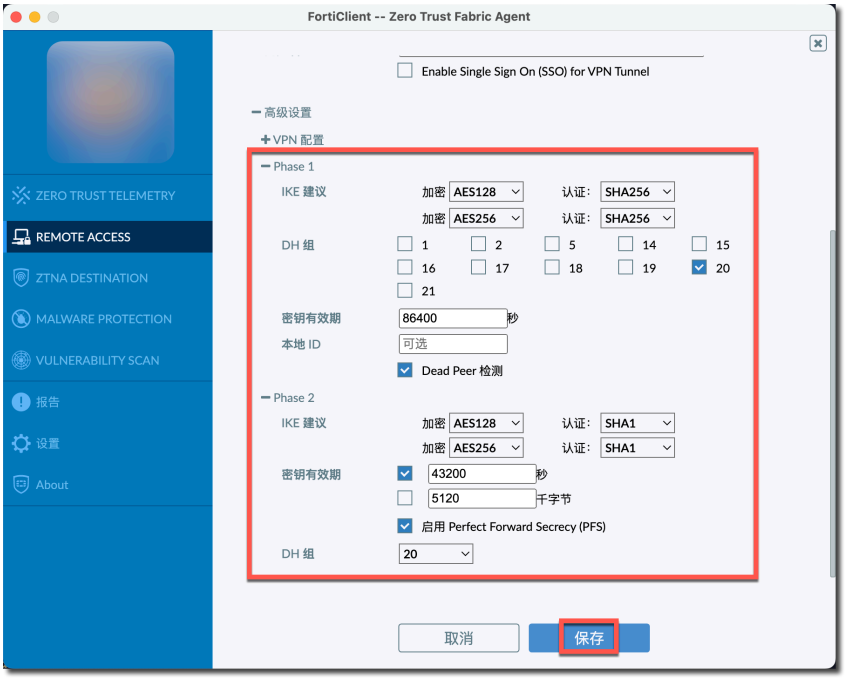


重要

可选配置：

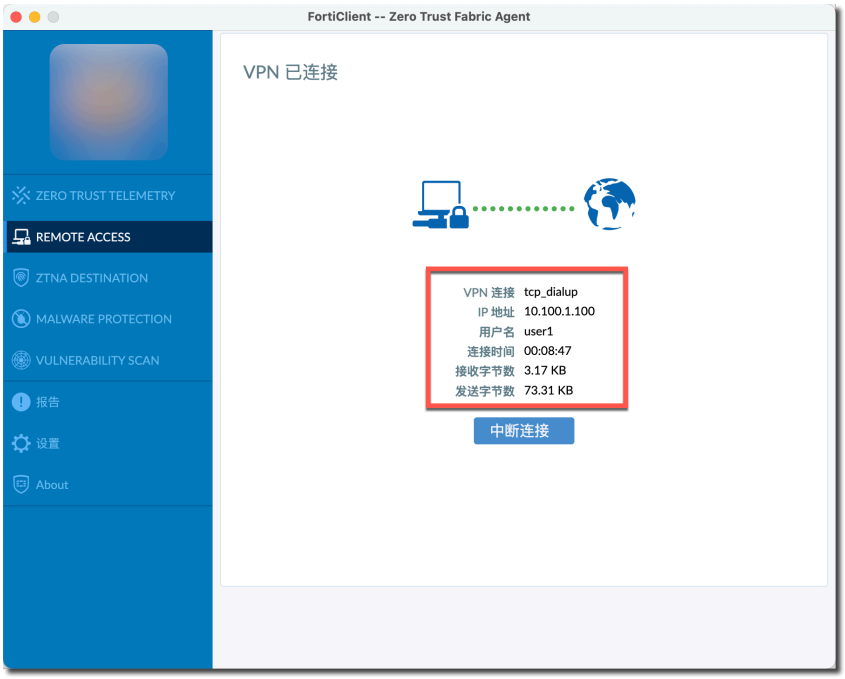
- `Encapsulation`：
  - Auto (UDP fallback TCP)：优先使用 UDP 封装 IKE 协商报文，如果 UDP 连接超时（默认超时时间为 15s），需要 FortiGate 的 IPSec 一阶段中的 `transport` 也设置为 `auto` 才能使用。 :::

3. 配置 IPSec 一阶段与二阶段，配置的算法包含在 FortiGate 配置的安全提议中即可。点击“保存”按钮创建 IPSec 连接。

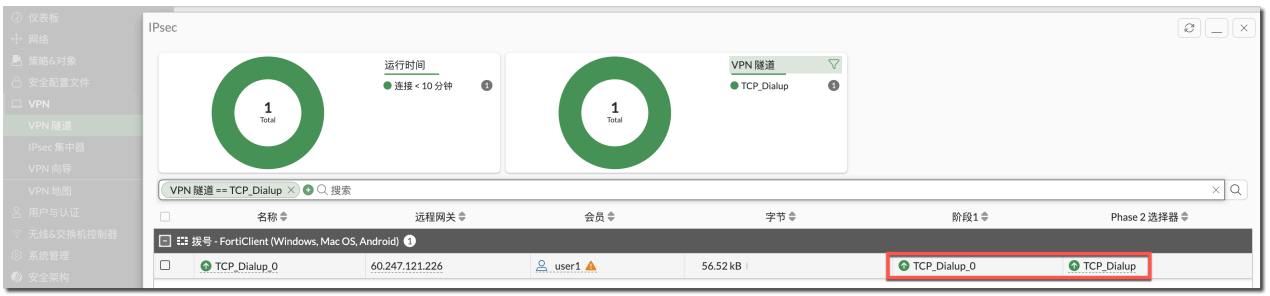


结果验证

1. 在 FortiClient 中输入账号/密码，点击“连接”，FortiClient 拨号成功。



2. FortiGate GUI 显示 IPSec 一二阶段均建立成功。



3. 在 FortiGate 上查看 IPSec 一阶段状态，可以看到传输模式为 TCP。

```bash
FortiGate # diagnose vpn ike gateway list name TCP_Dialup_0

vd: root/0
name: TCP_Dialup_0
version: 2
interface: port6 12
addr: 123.112.241.45:4443 → 60.247.121.226:59961
tun_id: 10.100.1.100/::10.0.0.15
remote_location: 0.0.0.0
network-id: 0
transport: TCP    //传输协议为TCP//
created: 7s ago
eap-user: user1    //EAP认证的用户名//
2FA: no
peer-id: Summerices-MacBook-Pro.local
peer-id-auth: no
FortiClient UID: 20835405F599455B9842348F3C690201
assigned IPv4 address: 10.100.1.100/255.255.255.255    //分配的IP地址//
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 120/120/120 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 70 de71e2a0d62b8736/24365703d2c80d80
  direction: responder
  status: established 7-7s ago = 120ms
  proposal: aes128-sha256
  child: no
  SK_ei: 9e8bbde335eec3a5-885365f407bc0db2
  SK_er: 00eb55dc186eebfe-fba2d0f60debd94e
  SK_ai: 73c01df8a38dacbb-4d79c9eb83842232-affb78dcf6a39117-996de329f3049a45
  SK_ar: dd98c48c5201681a-b979d55de179487c-184da864964c75fa-8464b07fdc107016
  PPK: no
  message-id sent/recv: 0/6
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/86122
  DPD sent/recv: 00000000/00000000
  peer-id: Summerices-MacBook-Pro.local
```

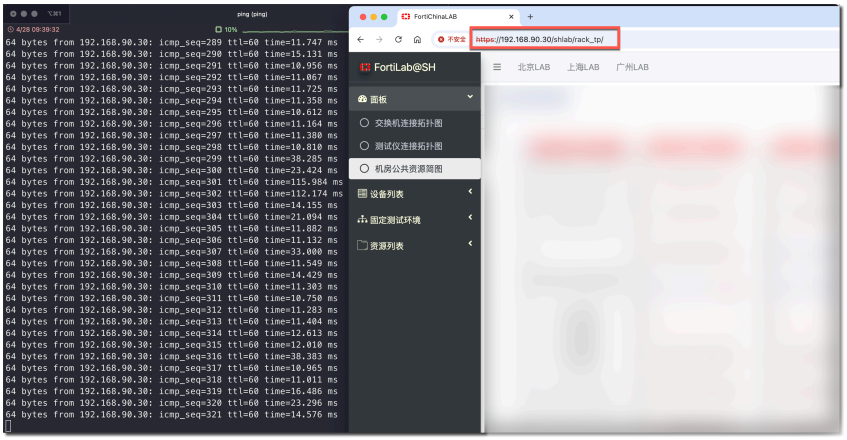
4. 在 FortiGate 上查看 IPSec 二阶段状态。

```bash
FortiGate # diagnose vpn tunnel list name TCP_Dialup_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=TCP_Dialup_0 ver=2 serial=1d 123.112.241.45:4443→60.247.121.226:59961 nexthop=123.112.241.1 tun_id=10.100.1.100
tun_id6=::10.0.0.15 status=up dst_mtu=1500 weight=1
bound_if=5 real_if=5 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74664 options[123a8]=npu rgwy-chg rport-chg
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=TCP_Dialup index=0
proxyid_num=1 child_num=0 refcnt=7 ilast=45 olast=231 ad=/0
stat: rxp=670 txp=519 rxb=58809 txb=43596
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=TCP_Dialup proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.100.1.100-10.100.1.100:0    //目标感兴趣流是为FortiClient分配的IP地址//
  SA:  ref=3 options=6a6 type=00 soft=0 mtu=1438 expire=42398/0B replaywin=2048
       seqno=208 esn=0 replaywin_lastseq=0000029e qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43190/43200
  dec: spi=1a0ea8ca esp=aes key=16 04866c80174edb4312e10f6b793f94c9
       ah=sha1 key=20 7d7024df2395a98611823438 65a6d00d636cedbe
  enc: spi=0ac10c0a esp=aes key=16 0abce89fea1f16aaf2b6e5b7411f5bcc
       ah=sha1 key=20 195e92517ac09ed37e7de8ab851fd69d3ba61802
  dec:pkts/bytes=670/58809, enc:pkts/bytes=519/78888
  npu_flag=00 npu_rgwy=0.0.0.0 npu_lgwy=0.0.0.0 npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

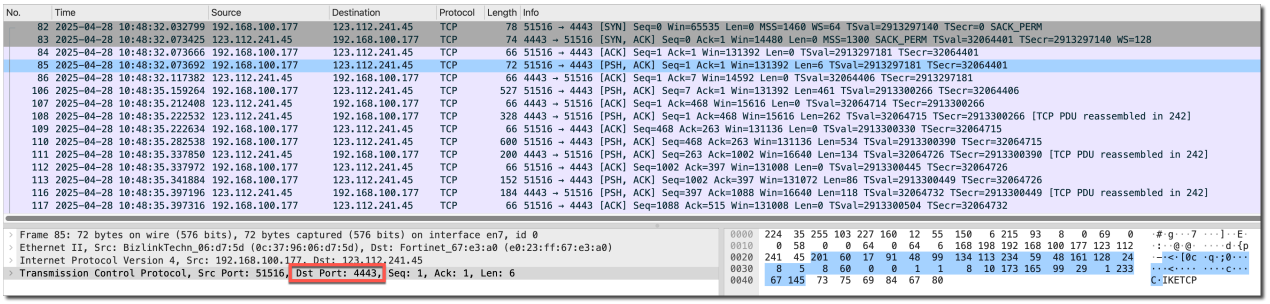
5. 查看 FortiClient 所在终端 IPv4 路由表（以下为 macOS 查看 IPv4 路由的命令，Windows 可以使用 `route print -4` 查看），192.168.90.0/24 指向 FortiClient 获取的地址 10.100.1.100。

```bash
::: important
netstat -rn -f inet
:::
Routing tables

Internet:
Destination        Gateway            Flags               Netif Expire
......
10.100.1.100       10.100.1.100       UHr                 utun8
10.100.1.100/32    link#34            UCS                 utun8
192.168.90         10.100.1.100       UGSc                utun8
......
```

6. 使用 FortiClient 所在终端访问 192.168.90.30，可以通过 IPSec 正常访问。



7. 抓包查看 IPSec over TCP 的协商过程与业务流量，可以看到 IKEv2 协商与 ESP 流量都被 TCP 4500 封装（RFC 9293）。



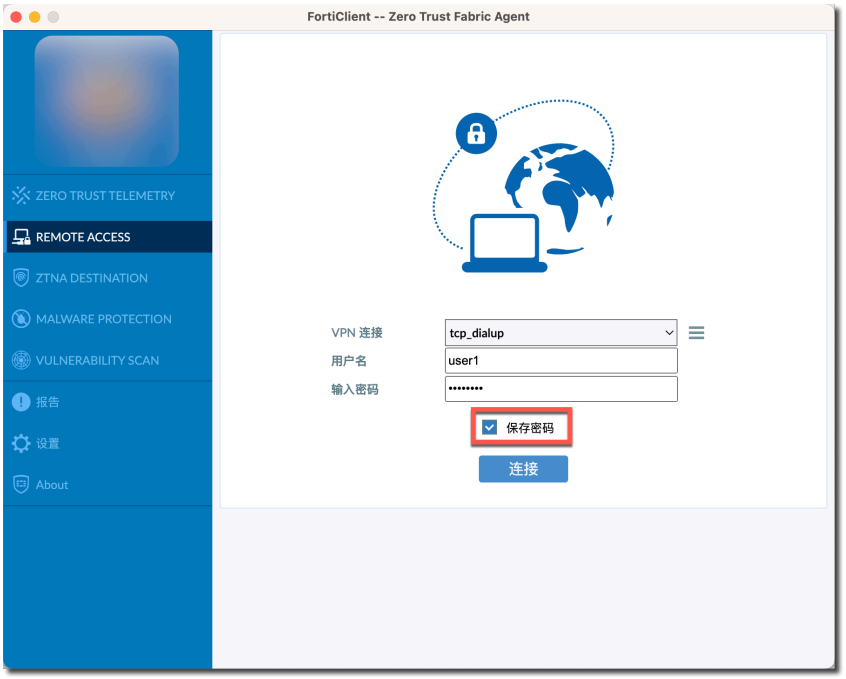
8. 查看 FortiClient 访问 VPN 资源的会话，基于 TCP 的 IPSec 流量（未开启 `fortinet-esp`）无法被 NP 加速。

```bash
session info: proto=1 proto_state=00 duration=14 expire=59 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/10.100.1.100 vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=1260/15/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 83/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=53->13/13->53 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.100.1.100:44090->192.168.90.30:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.90.30:44090->10.100.1.100:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=1904 auth_info=0 chk_client_info=0 vd=0
serial=0000034d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000100
no_ofld_reason:  npu-flag-off    //流量无法被NP加速//
```

9. 查看 FortiGate 的硬件/软件加解密统计，可以看到基于 TCP 的 IPSec 流量（未开启 `fortinet-esp`）虽然无法被 NP 加速，但可以被 CP 加速，CP 芯片统计计数一直在增长，而 SOFTWARE（CPU）统计计数一直为 0。

```bash
FortiGate # diagnose vpn ipsec status
All ipsec crypto devices in use:
np7_0:    //NP统计计数始终为0//
    Encryption (encrypted/decrypted)
        null             : 0                0
        des              : 0                0
        3des             : 0                0
        aes              : 0                0
        aes-gcm          : 0                0
        aria             : 0                0
        seed             : 0                0
        chacha20poly1305 : 0                0
    Integrity (generated/validated)
        null             : 0                0
        md5              : 0                0
        sha1             : 0                0
        sha256           : 0                0
        sha384           : 0                0
        sha512           : 0                0

......

CP9:    //CP统计计数一直在增长//
    Encryption (encrypted/decrypted)
        null             : 0                0
        des              : 0                0
        3des             : 0                0
        aes              : 47               464
        aes-gcm          : 0                0
        aria             : 0                0
        seed             : 0                0
        chacha20poly1305 : 0                0
    Integrity (generated/validated)
        null             : 0                0
        md5              : 0                0
        sha1             : 47               464
        sha256           : 0                0
        sha384           : 0                0
        sha512           : 0                0

......

SOFTWARE:    //CPU统计计数始终为0//
    Encryption (encrypted/decrypted)
        null             : 0                0
        des              : 0                0
        3des             : 0                0
        aes              : 0                0
        aes-gcm          : 0                0
        aria             : 0                0
        seed             : 0                0
        chacha20poly1305 : 0                0
    Integrity (generated/validated)
        null             : 0                0
        md5              : 0                0
        sha1             : 0                0
        sha256           : 0                0
        sha384           : 0                0
        sha512           : 0                0
```

.0. FortiClient 第一次连接成功后点击“中断连接”，就会显示“保存密码”选项。


FortiClient -- Zero Trust Fabric Agent
ZERO TRUST TELEMETRY
REMOTE ACCESS
ZTNA DESTINATION
MALWARE PROTECTION
VULNERABILITY SCAN
报告
设置
About
VPN 连接
tcp_dialup
用户名
user1
输入密码
保存密码
连接

IPSec 多用户组访问控制

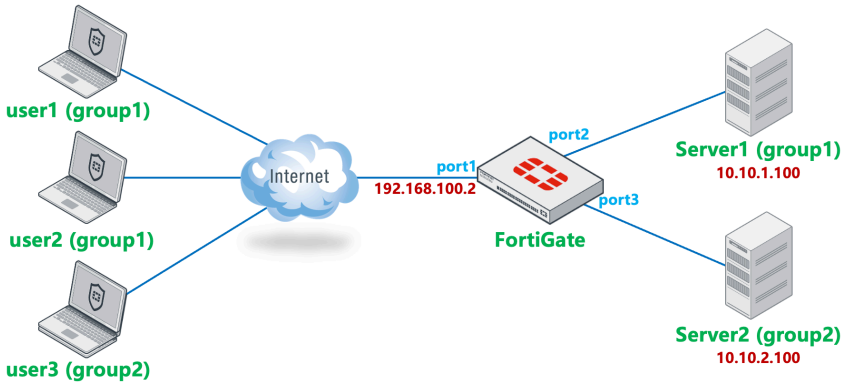
功能简介

在 IPsec VPN 远程接入场景中，用户组用于在建立隧道前对远程用户进行身份认证。一个 IPsec VPN 隧道可以关联单个用户组或多个用户组。

用户组在 IPSec VPN 隧道中的配置位置可以存在于 2 处位置：

- 配置在 IPSec Phase 1（P1）中：只能使用一个用户组。
- 配置在防火墙策略（Policy）中：
  - 同一条防火墙策略可以同时配置多个用户组。
  - 不同的防火墙策略可以配置不同的用户组，为不同的用户组分配不同的访问权限。
  - 类似 SSL VPN 的 Portal（门户）功能，这在将 SSL VPN 迁移到 IPSec 时非常有用。

网络拓扑



- 用户 user1、user2 属于用户组 group1，用户 user3 属于用户组 group2。
- user1/user2/user3 分别通过 FortiClient IPSec VPN 拨号连接 FortiGate，访问 Server1/Server2。
- 在防火墙策略中控制：
  - 用户组 group1（user1/user2）拨号后只能访问 Server1，无法访问 Server2。
  - 用户组 group2（user3）拨号后只能访问 Server2，无法访问 Server1。

配置方法

基础网络配置略。

IPSec 一阶段配置用户组

只能使用一个用户组，配置方法请参考：VPN → IPSec VPN → 拨号 VPN → FortiClient 拨号↗章节。参考配置如下：

```bash
config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "group1"
        set transport udp
        set ipv4-start-ip 10.100.1.100
        set ipv4-end-ip 10.100.1.200
        set dns-mode auto
        set ipv4-split-include "Server"
        set save-password enable
        set psksecret xxxxxx
```

```
    next
end

config vpn ipsec phase2-interface
    edit "IPSec_Dialup"
        set phase1name "IPSec_Dialup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

防火墙策略配置用户组

1. 根据网络拓扑中的内容创建用户与用户组。

```bash
config user local
    edit "guest"
        set type password
        set xxxxxx
    next
    edit "user1"
        set type password
        set passwd xxxxxx
    next
    edit "user2"
        set type password
        set passwd xxxxxx
    next
    edit "user3"
        set type password
        set xxxxxx
    next
end

config user group
    edit "group1"
        set member "user1" "user2"
    next
    edit "group2"
        set member "user3"
    next
end
```

2. 进入“VPN → VPN 向导”页面，模板选择“远程访问”，填写隧道名称，并点击“开始”按钮。



3. 配置 VPN 隧道相关信息。



- 认证方法：根据需求选择，这里以预共享密钥为例。
- IKE：由于 FortiClient 在 7.4.4 版本开始不再支持 IKEv1，这里推荐使用 IKEv2。
- 传输：使用 UDP，如使用 UDP 协商失败，再尝试使用自动方式。
- EAP 对等体识别：使用默认的 EAP 认证。
- 用户身份验证方法：选择“从策略继承”，并选择用户组 group1 和 group2。如果选择“阶段 1 接口”，则只能选择一个用户组。

提示

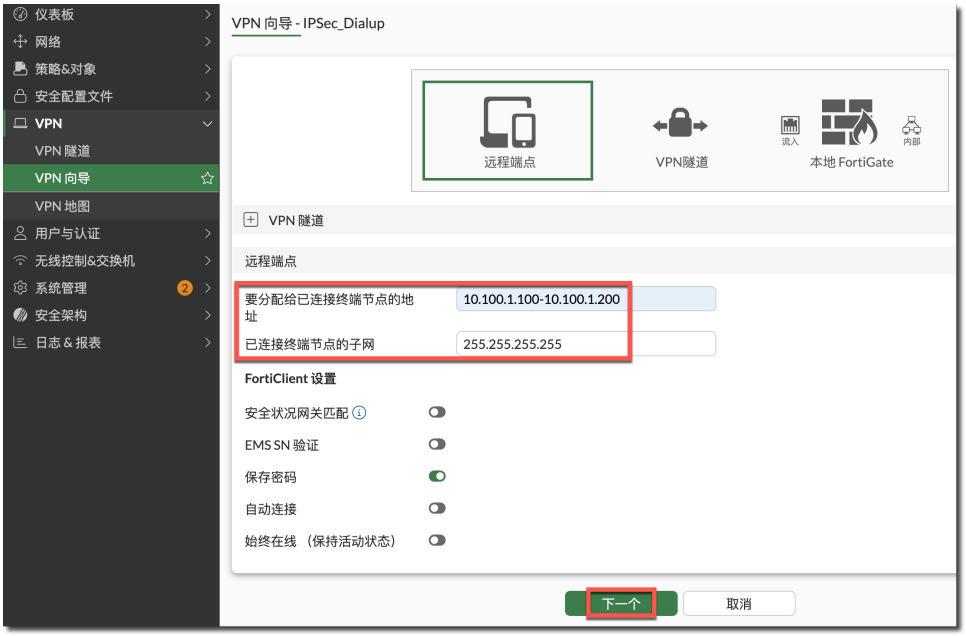
- FortiOS 7.6 以下版本的 GUI 向导没有“从策略继承”选项，可以在 GUI 向导配置时先选择一个用户组，在向导配置结束后，修改 IPSec 隧道配置，在 XAUTH 处将用户组的模式修改为“从策略继承”。



- 也可以在 IPSec 一阶段的 CLI 配置中取消引用用户组，即可达到“从策略继承”相同的配置效果。

```bash
config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        unset authusrgrp "group1"
    next
end
```

4. 配置给客户端分配的 IP 地址范围。

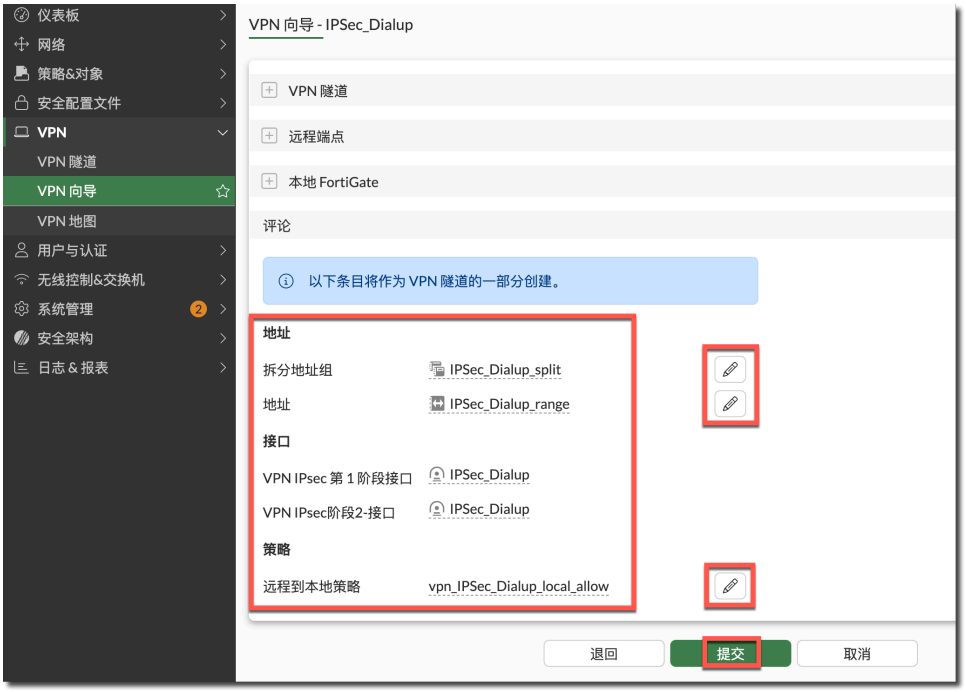


5. 配置本地 FortiGate 相关信息。



- 与隧道绑定的输入接口：选择用于接入 VPN 的互联网接口。
- 本地接口：面向内网服务器 Server1 和 Server2 的接口。
- 本地地址：客户端拨号成功后访问的内网资源（隧道分割），这里以 Server1 和 Server2 所在的网段为例。

6. 检查配置无误后，提交配置（可以修改自动生成的隧道分割地址/地址组、防火墙策略名称）。



7. 向导自动生成的隧道分割地址组与 IPSec 一阶段/二阶段配置 CLI 命令。

```bash
config firewall addrgrp
    edit "IPSec_Dialup_split"
        set member "Server1_10.10.1.0/24" "Server2_10.10.2.0/24"
        set comment "VPN: IPSec_Dialup -- Created by VPN wizard"
        set uuid be3f392e-fdaf-51f0-431c-681b8197a201
    next
end

config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set comments "VPN: IPSec_Dialup -- Created by VPN wizard"
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set transport udp
        set ipv4-start-ip 10.100.1.100
        set ipv4-end-ip 10.100.1.200
        set dns-mode auto
        set ipv4-split-include "Server"
        set save-password enable
        set psksecret ENC
aZFKK834W6UFV39Tkp9enmmHi5ke2PbXeCsnrExTfzHtYHqaUgcirBnxASvBdeCVQjph+8WW2MJXd59ZWSr2OcfpbNK5k3AGqNBzbXvSY/Is4Yo5nUKmhBEH
9oPkuUlhyp4+ibXcYd4Eq1Di0aIfloxfPCFCn5MNPQDOQ0YH6BTu+gCTcU8kIaiysXW7gytAEa0HLllmMjY3dkVA
    next
end

config vpn ipsec phase2-interface
    edit "IPSec_Dialup"
        set phase1name "IPSec_Dialup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: IPSec_Dialup -- Created by VPN wizard"
    next
end
```

8. 查看自动生成的防火墙策略，只有 1 条。




```bash
config firewall policy
    edit 1
        set name "vpn_IPSec_Dialup_local_allow"
        set uuid be44468a-fdaf-51f0-0db3-180ea9115c6d
        set srcintf "IPSec_Dialup"
        set dstintf "port2"
        set action accept
        set srcaddr "IPSec_Dialup_range"
        set dstaddr "Server1_10.10.1.0/24" "Server2_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "group1" "group2"
        set comments "VPN: IPSec_Dialup -- Created by VPN wizard"
    next
end
```

9. 为了实现网络拓扑中的需求，需要将其拆分成 2 条策略。

> **提示**
>
> - VPN 向导自动创建的防火墙策略开启了 NAT，在此拓扑下防火墙策略不需要 NAT（Server 配置的网关均为 FortiGate），所以这里关闭了 2 条防火墙策略的 NAT。
> - 为了后续验证结果，开启了防火墙策略的日志记录功能，与隐含策略的日志记录功能。

> **重要**
>
> 不要在 IPSec 一阶段（`config vpn ipsec phase1-interface`）和防火墙策略（`config firewall policy`）中同时配置用户组，可能会产生匹配逻辑问题。

- 用户组 group1（user1/user2）拨号后只能访问 Server1，无法访问 Server2。
- 用户组 group2（user3）拨号后只能访问 Server2，无法访问 Server1。



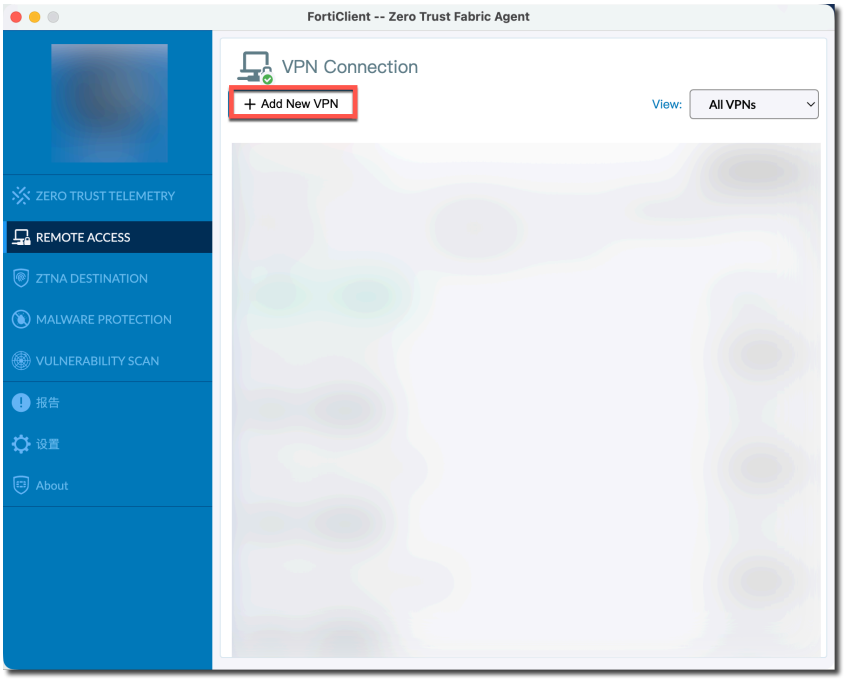

```bash
config firewall policy
    edit 1
        set name "IPSec_Dialup_Server1"
        set srcintf "IPSec_Dialup"
        set dstintf "port2"
        set action accept
        set srcaddr "IPSec_Dialup_range"
        set dstaddr "Server1_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set groups "group1" "group2"
    next
    edit 2
        set name "IPSec_Dialup_Server2"
        set srcintf "IPSec_Dialup"
        set dstintf "port3"
        set action accept
        set srcaddr "IPSec_Dialup_range"
        set dstaddr "Server2_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set groups "group2"
```

```
    next
end

config log setting
    set fwpolicy-implicit-log enable
end
```

结果验证

用户组 group1（user1/user2）拨号后只能访问 Server1，无法访问 Server2

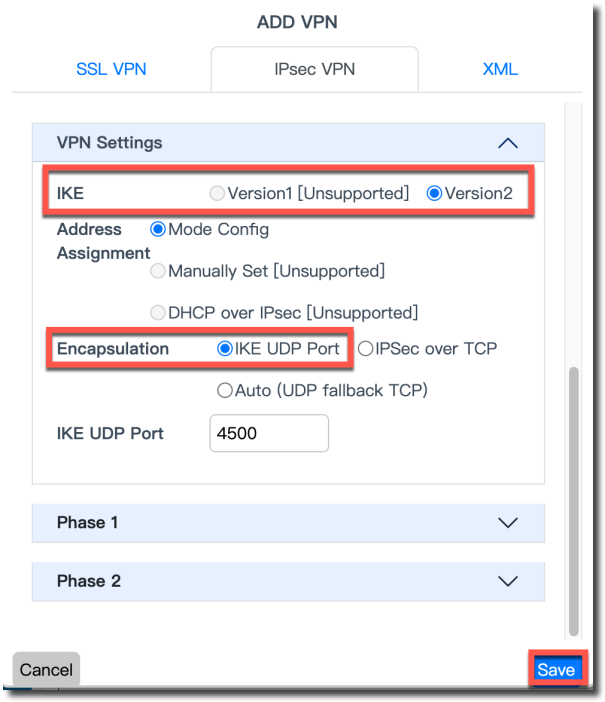
1. user1 在 FortiClient 客户端（这里以 FortiClient 7.4.5 为例）创建新的 VPN 连接。



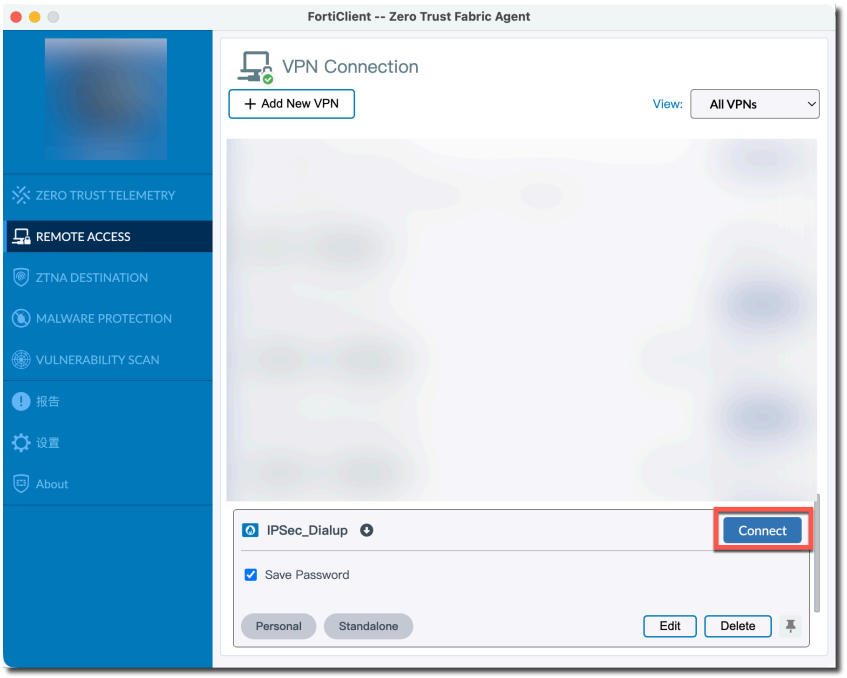
2. 配置 IPSec VPN 服务器 IP、预共享密钥、用户名等信息。



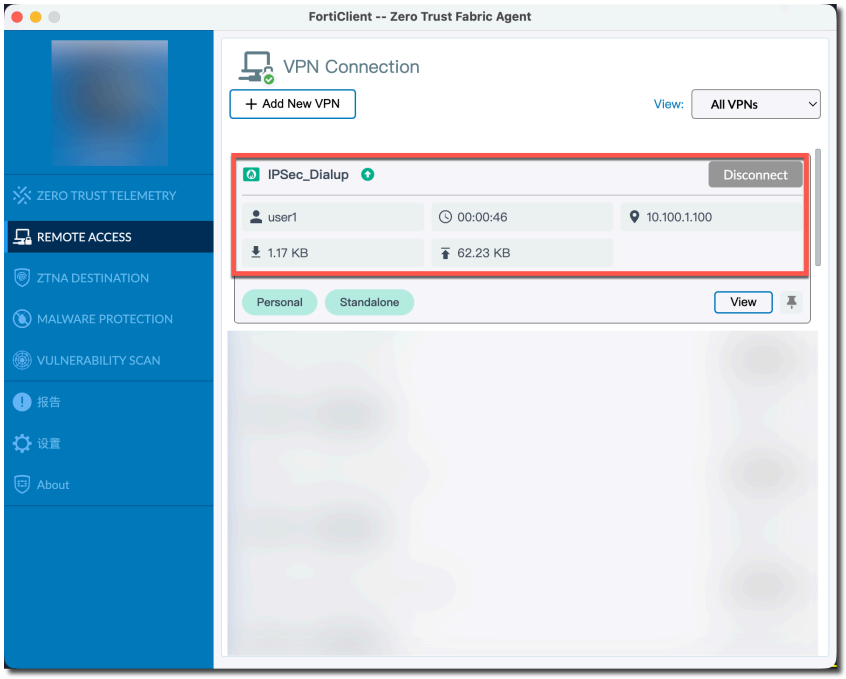
3. 配置 IKE 版本为 IKEv2，封装模式选择 IKE UDP Port，保存配置。



4. 在已创建的 IPSec VPN 连接处点击“Connect”按钮。



5. 填写账号与密码信息后，连接成功，客户端获取到 IP 地址。



6. 在 FortiGate 查看 IPSec VPN 连接状态，可以看到用户、分配 IP 等信息。




```bash
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: IPSec_Dialup_0
version: 2
interface: port1 3
addr: 192.168.100.2:4500 → 192.168.100.177:60054
tun_id: 10.100.1.100/::10.0.0.3
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 226s ago
eap-user: user1
2FA: no
groups:
  group1 2
```

7. 在 user1 的 PC 查看路由表，可以看到 FortiGate 给 user1 的 PC 推送了 Server1 和 Server2 两个网段的路由。

> **提示**
>
> 这里以 macOS 为例，Windows 请在 CMD 下使用 `route print -4` 命令查看。

```bash
user1 # netstat -nr -f inet
Routing tables

Internet:
Destination        Gateway            Flags               Netif Expire
......
10.10.1/24         10.100.1.100       UGSc                utun8
10.10.2/24         10.100.1.100       UGSc                utun8
......
```

8. user1 的 IPSec 拨号成功后，访问 Server1，可以正常访问，相关流量日志显示匹配防火墙策略 ID 1。

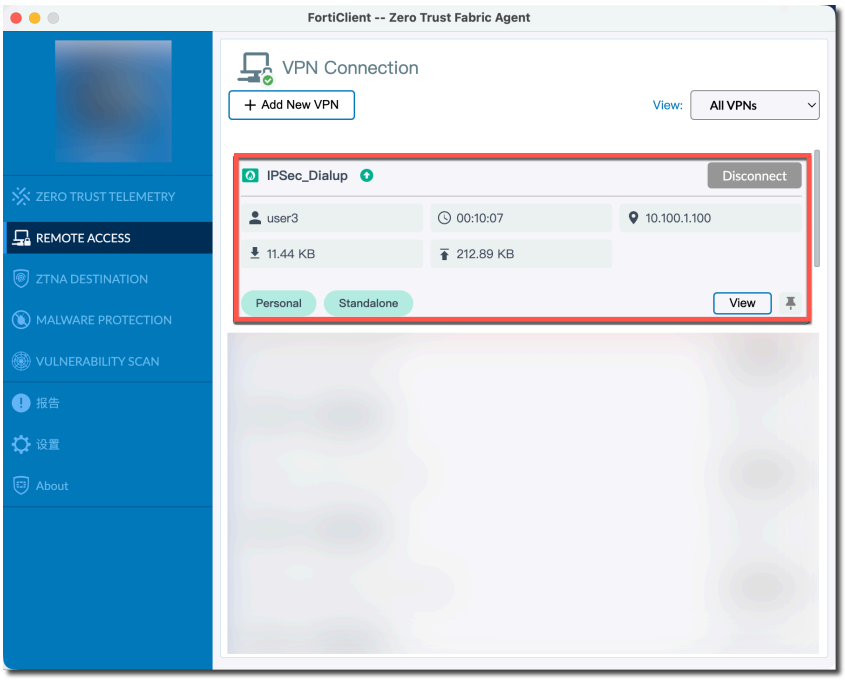
```bash
user1 # ping 10.10.1.100 -c 3
PING 10.10.1.100 (10.10.1.100): 56 data bytes
64 bytes from 10.10.1.100: icmp_seq=0 ttl=127 time=4.712 ms
64 bytes from 10.10.1.100: icmp_seq=1 ttl=127 time=4.943 ms
64 bytes from 10.10.1.100: icmp_seq=2 ttl=127 time=3.157 ms

--- 10.10.1.100 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.157/4.271/4.943/0.793 ms

date=2026-01-30 time=16:36:20 eventtime=1769762179464495516 tz="+0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" srcip=10.100.1.100 identifier=55135 srcintf="IPSec_Dialup"
srcintfrole="undefined" dstip=10.10.1.100 dstintf="port2" dstintfrole="undefined" srccountry="Reserved"
dstcountry="Reserved" sessionid=557 proto=1 action="accept" policyid=1 policytype="policy" poluuid="be44468a-fdaf-51f0-
0db3-180ea9115c6d" policyname="IPSec_Dialup_Server1" user="user1" group="group1" authserver="user1" service="PING"
trandisp="noop" appcat="unscanned" duration=62 sentbyte=252 rcvdbyte=252 sentpkt=3 rcvdpkt=3 vpntype="ipsecvpn"
```

9. user1 的 IPSec 拨号成功后，访问 Server2，无法访问，相关流量日志显示匹配防火墙隐含策略（ID 0）被丢弃。

```bash
user1 # ping 10.10.2.100 -c 3
PING 10.10.2.100 (10.10.2.100): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2

--- 10.10.2.100 ping statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss

date=2026-01-30 time=16:38:30 eventtime=1769762310115699448 tz="+0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" srcip=10.100.1.100 identifier=40802 srcintf="IPSec_Dialup"
srcintfrole="undefined" dstip=10.10.2.100 dstintf="port3" dstintfrole="undefined" srccountry="Reserved"
dstcountry="Reserved" sessionid=718 proto=1 action="deny" policyid=0 policytype="policy" user="user1" authserver="user1"
service="PING" trandisp="noop" appcat="unscanned" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0
vpntype="ipsecvpn" crscore=30 craction=131072 crlevel="high"
```

用户组 group2（user3）拨号后只能访问 Server2，无法访问 Server1

1. 使用 user3 的 FortiClient 连接 FortiGate 的 IPSec 。



2. user3 的 IPSec 拨号成功后，访问 Server1，无法访问，相关流量日志显示匹配防火墙隐含策略（ID 0）被丢弃。

```
user3 # ping 10.10.1.100 -c 3                                                                                    bash
PING 10.10.1.100 (10.10.1.100): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1

--- 10.10.1.100 ping statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss

date=2026-01-30 time=16:54:21 eventtime=1769763260787974122 tz="+0800" logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="root" srcip=10.100.1.100 identifier=64112 srcintf="IPSec_Dialup" srcintfrole="undefined"
dstip=10.10.1.100 dstintf="port2" dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=2190
proto=1 action="deny" policyid=0 policytype="policy" user="user3" authserver="user3" service="PING" trandisp="noop"
appcat="unscanned" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 vpntype="ipsecvpn" crscore=30 craction=131072
crlevel="high"
```

3. user3 的 IPSec 拨号成功后，访问 Server2，可以正常访问，相关流量日志显示匹配防火墙策略 ID 2。

```
user3 # ping 10.10.2.100 -c 3                                                                                    bash
PING 10.10.2.100 (10.10.2.100): 56 data bytes
64 bytes from 10.10.2.100: icmp_seq=0 ttl=127 time=3.327 ms
64 bytes from 10.10.2.100: icmp_seq=1 ttl=127 time=3.262 ms
64 bytes from 10.10.2.100: icmp_seq=2 ttl=127 time=3.159 ms

--- 10.10.2.100 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.159/3.249/3.327/0.069 ms

date=2026-01-30 time=16:55:15 eventtime=1769763315534392580 tz="+0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" srcip=10.100.1.100 identifier=55664 srcintf="IPSec_Dialup"
srcintfrole="undefined" dstip=10.10.2.100 dstintf="port3" dstintfrole="undefined" srccountry="Reserved"
dstcountry="Reserved" sessionid=2181 proto=1 action="accept" policyid=2 policytype="policy" poluuid="642d9c36-fdb0-51f0-
e2ae-322fc0df415e" policyname="IPSec_Dialup_Server2" user="user3" group="group2" authserver="user3" service="PING"
trandisp="noop" appcat="unscanned" duration=60 sentbyte=84 rcvdbyte=84 sentpkt=1 rcvdpkt=1 vpntype="ipsecvpn"
```

与友商进行 IPSec VPN 对接

- 与友商对接 IPSec 的注意事项
- 与阿里云建立 IPSec VPN
- 与思科路由器建立 IPSec VPN (IKE v1)
- 与思科路由器建立 IPSec VPN (IKE v2)
- 与思科 ASA 防火墙建立 IPSec VPN (IKE v1)
- 与思科 ASA 防火墙建立 IPSec VPN (IKE v2)
- 与华为 USG6000V 防火墙建立 IPSec VPN
- 与 H3C SecPath 防火墙建立 IPSec VPN
- 与山石 SG6000 防火墙建立 IPSec VPN
- 与 Palo Alto 防火墙建立 IPSec VPN
- 与 Check Point 防火墙建立 IPSec VPN
- 与 Juniper SRX 防火墙建立 IPSec VPN
- 与 Openswan 建立 IPSec VPN
- 与 strongSwan 建立 IPsec VPN

与友商对接 IPSec 的注意事项

1. 由于每个厂商的地址组在 IPSec 协商中封装的格式可能不一致（FortiGate 使用地址组时，二阶段只使用一个 SA 包含多个网段，不会切割为多个 SA），可能导致协商失败或隧道翻转等现象，二阶段感兴趣流如果存在多对网段，请不要使用地址组作为第二阶段感兴趣流。请使用多个 Subnet 的作为二阶段感兴趣流的模式。
2. 由于每个厂商对两端 SA 重协商时间不一致时的处理方法不一致（FortiGate 会使用双方 SA 重协商较小的时间作为自己的 SA 重协商时间），两端的一、二阶段 SA 重协商时间请尽量配置为相同的时间。
3. 如果 IPSec 与友商对接时选择证书认证，可能会出现连接无法建立的情况。
   - 在 FortiGate 的 IPSec Debug 信息中，如果看到如下报错：

     ```bash
     FortiGate # diagnose debug application ike -1
     FortiGate # diagnose debug enable

     RenHangNewVPN: 7913020: received p1 notify type INITIAL-CONTACT
     //RenHangNewVPN: 7913020: peer identifier IPV4_ADDR 54.121.33.6//
     RenHangNewVPN: 7913020: Validating X.509 certificate
     //RenHangNewVPN: 7913020: peer cert, subject='IPSECVPN', issuer='CFIT'//
     //RenHangNewVPN: 7913020: peer ID does not match cert//
     RenHangNewVPN: 7913020: certificate validation failed
     RenHangNewVPN: 7913020: info_send_n1, type 20
     ......
     ```

   - 这是由于 FortiGate 选择证书认证时，默认会开启证书 ID（subject 或称为 CN）与对端 ID 的一致性检查（符合 RFC 4945）。当对端 IPSec 设备发送的 ID（此例中为 IPv4 54.121.33.6）与对端发送的证书的 subject（此例中为“IPSECVPN”）不一致时，会认证失败。

     ```bash
     config vpn ipsec phase1-interface
         edit "FGT110"
             set cert-id-validation enable    //使用证书认证后默认开启//
         next
     end
     ```

   - 当然，FortiGate 在开启 IPSec 证书认证时，会自动将本端 ID 修改为“asn1dn”类型，并将本端证书的 subject 复制到本端 ID 中，不会使用默认的本端 IPv 地址作为本端 ID（符合 RFC 4945）。

     ```bash
     config vpn ipsec phase1-interface
         edit "FGT110"
             set interface "port8"
             set authmethod signature
             set peertype any
             set net-device disable
             set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
             set localid "FGT200"    //配置locaid-type asn1dn后自动从本端证书的subject中读取后配置//
             set localid-type asn1dn    //开启证书认证后自动配置//
             set comments "VPN: FGT110 (Created by VPN wizard)"
             set remote-gw 192.168.8.111
             set certificate "FGT200"
         next
     end
     ```

   - 所以当 IPSec 两端均为 FortiGate 时，如果没有手工更改 ID 配置，连接是可以正常建立的。但如果对端设备为友商设备，可能不会遵守 RFC 4945，仍然使用 IP 地址作为 ID 发送至 FortiGate，这样就会导致协商失败。
   - 解决办法：
     - 是在 FortiGate 上关闭证书 ID 与对端 ID 的一致性检查。

       ```bash
       config vpn ipsec phase1-interface
           edit "FGT110"
               set cert-id-validation disable
           next
       end
       ```

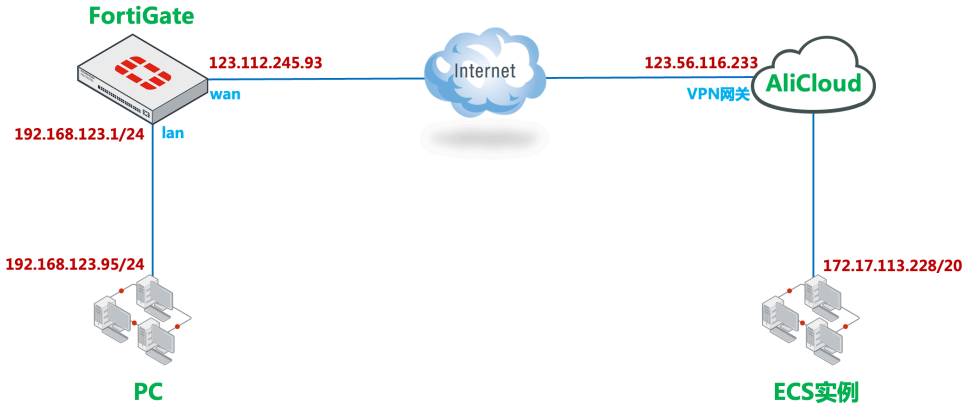
     - 或是在对端友商设备上配置 ID 类型为 asn1dn，ID 内容为其证书的 subject（如果友商设备支持此配置）。

与阿里云建立 IPSec VPN

网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 172.17.112.0/20 两个网段的通信。与阿里云进行 IPsec VPN（IKE v1）对接。

组网拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置阿里云 VPN
  - 配置 VPN 网关
  - 配置 IPsec VPN 连接
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略

注意事项

1. 在阿里云 IPSec 的工作机制中，阿里云 IPSec 一二阶段并没有关联，当阿里云的一阶段到期后，并不会重新发起一阶段的重协商，也不会删除对应的二阶段，而是需要满足一二阶段同时到期后才会发起一阶段的重协商。而对于 FortiGate 来说，只要一阶段的 SA 到期，就会删除二阶段的 SA 连接（FortiGate 作为安全设备，这是正常的操作）。如下图为阿里云工程师对于阿里云 IPSec 协商工作机制的解释。



2. 这样就会导致当 FortiGate 为 IPSec 协商的响应方，而阿里云为 IPSec 协商的发起方时，可能会出现一阶段协商到期后，FortiGate 将自己的一阶段和关联的二阶段连接删除，阿里云 IPSec 仅将自己的一阶段删除，保留二阶段直到二阶段倒计时结束，才会重新发起一阶段连接。

3. 此时 FortiGate 重新发起一阶段协商，阿里云是不响应的。会出现 FortiGate 端 IPSec 连接已经断开，而阿里云端仍然显示连接存在的情况，此时业务是不通的，需要等到阿里云残留的二阶段到期后，才会重新发起协商，如果两端二阶段配置的时间较长，则会中断较长的时间。

4.**想要规避此问题，需要 FortiGate 始终作为 IPSec VPN 的发起方，阿里云作为响应方，这样在一阶段到期前（30s 或 300s），FortiGate 总能主动重新发起新的一阶段和二阶段连接。**

5. 由于阿里云端的 IPSec 协商机制无法配置修改，**Workaround 是在 FortiGate 侧配置 Local-in Policy**，在没有 IKE 会话建立的情况下，丢弃对端发来的 IKE 协商包（另外，FortiGate 在 IPSec 连接建立后会删除 IKE 会话，也会丢弃在连接建立后对端发送的 IKE 协商包，避免自己成为响应方）。详情见配置步骤。

配置步骤

配置 FortiGate

1. 基本上网配置。





2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 配置网络、认证、第一阶段。

> **重要**
>
> 这里的安全提议（阶段 1 提案）建议只保留一组，与阿里云测配置一致，不要选择多个算法组合。

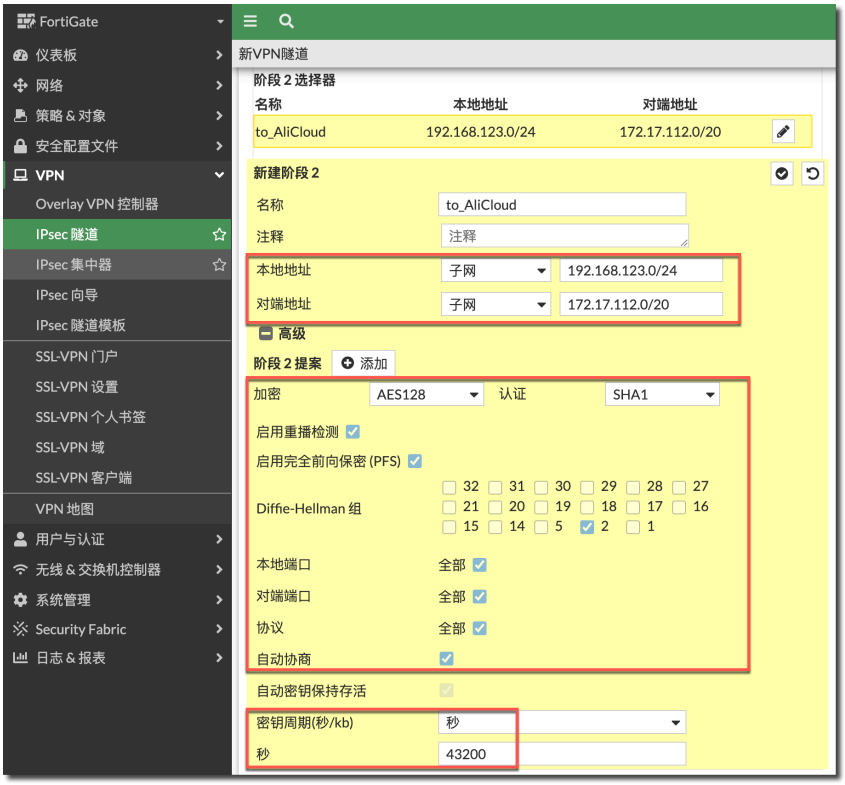



```
config vpn ipsec phase1-interface
    edit "to_AliCloud"
        set interface "wan"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
```

```
        set dhgrp 2
        set remote-gw 123.56.116.233
        set psksecret xxxxxx
    next
end
```

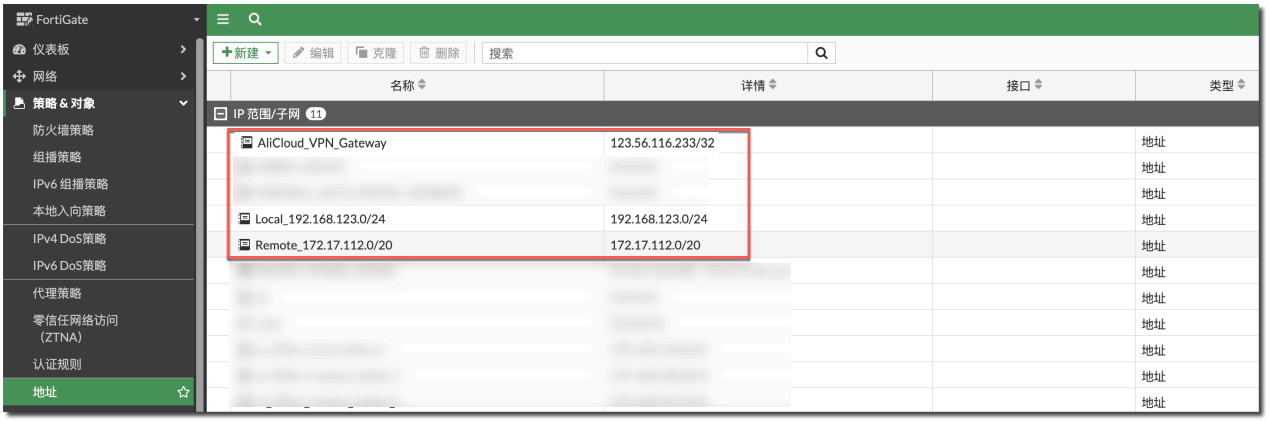
5. 配置第二阶段。

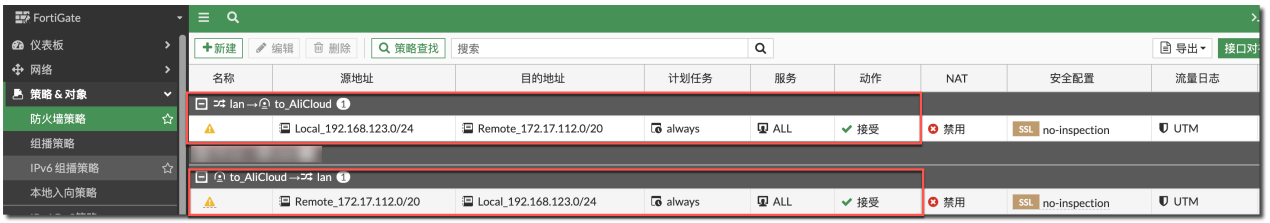
> **重要**
>
> 这里的安全提议（阶段 2 提案）建议只保留一组，与阿里云测配置一致，不要选择多个算法组合。




```
config vpn ipsec phase2-interface
    edit "to_AliCloud"
        set phase1name "to_AliCloud"
        set proposal aes128-sha1
        set dhgrp 2
        set auto-negotiate enable
        set src-subnet 192.168.123.0 255.255.255.0
        set dst-subnet 172.17.112.0 255.255.240.0
    next
end
```

6. 配置 VPN 相关的网段地址对象、阿里云 VPN 网关地址对象和防火墙策略。






```
config firewall address
    edit "Local_192.168.123.0/24"
        set subnet 192.168.123.0 255.255.255.0
    next
    edit "Remote_172.17.112.0/20"
        set subnet 172.17.112.0 255.255.240.0
    next
    edit "AliCloud_VPN_Gateway"
        set subnet 123.56.116.233 255.255.255.255
    next
end

config firewall policy
    edit 4
        set srcintf "lan"
        set dstintf "to_AliCloud"
        set action accept
        set srcaddr "Local_192.168.123.0/24"
        set dstaddr "Remote_172.17.112.0/20"
        set schedule "always"
        set service "ALL"
    next
    edit 5
        set srcintf "to_AliCloud"
        set dstintf "lan"
        set action accept
        set srcaddr "Remote_172.17.112.0/20"
        set dstaddr "Local_192.168.123.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 配置 VPN 业务网段的静态路由。



```
config router static
    edit 3
        set dst 172.17.112.0 255.255.240.0
        set device "to_AliCloud"
    next
    edit 4
        set dst 172.17.112.0 255.255.240.0
        set distance 254
        set blackhole enable
    next
end
```

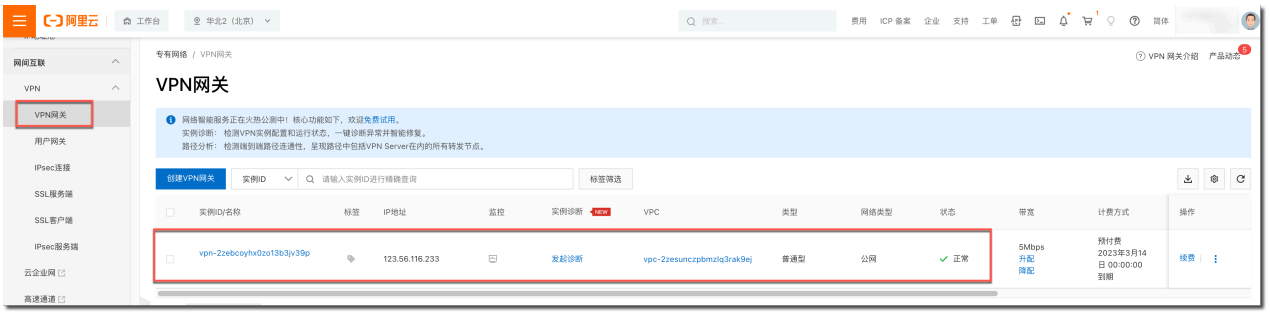
8.**为 FortiGate 配置 Local-in Policy，在没有 IKE 会话建立的情况下，丢弃阿里云侧发来的 IKE 协商包。**

> **重要**
>
> 此步非常重要，不配置很可能出现与阿里云 IPSec 连接中断后无法重新建立的情况。

```
config firewall local-in-policy
    edit 1
        set intf "wan"
        set srcaddr "AliCloud_VPN_Gateway"
        set dstaddr "all"
        set service "IKE"
        set schedule "always"
        set action deny
    next
end
```

配置阿里云 IPSec VPN

1. 进入网间互联 → VPN → VPN 网关页面，创建 VPN 网关，注意选择正确的与实例互通的交换机。



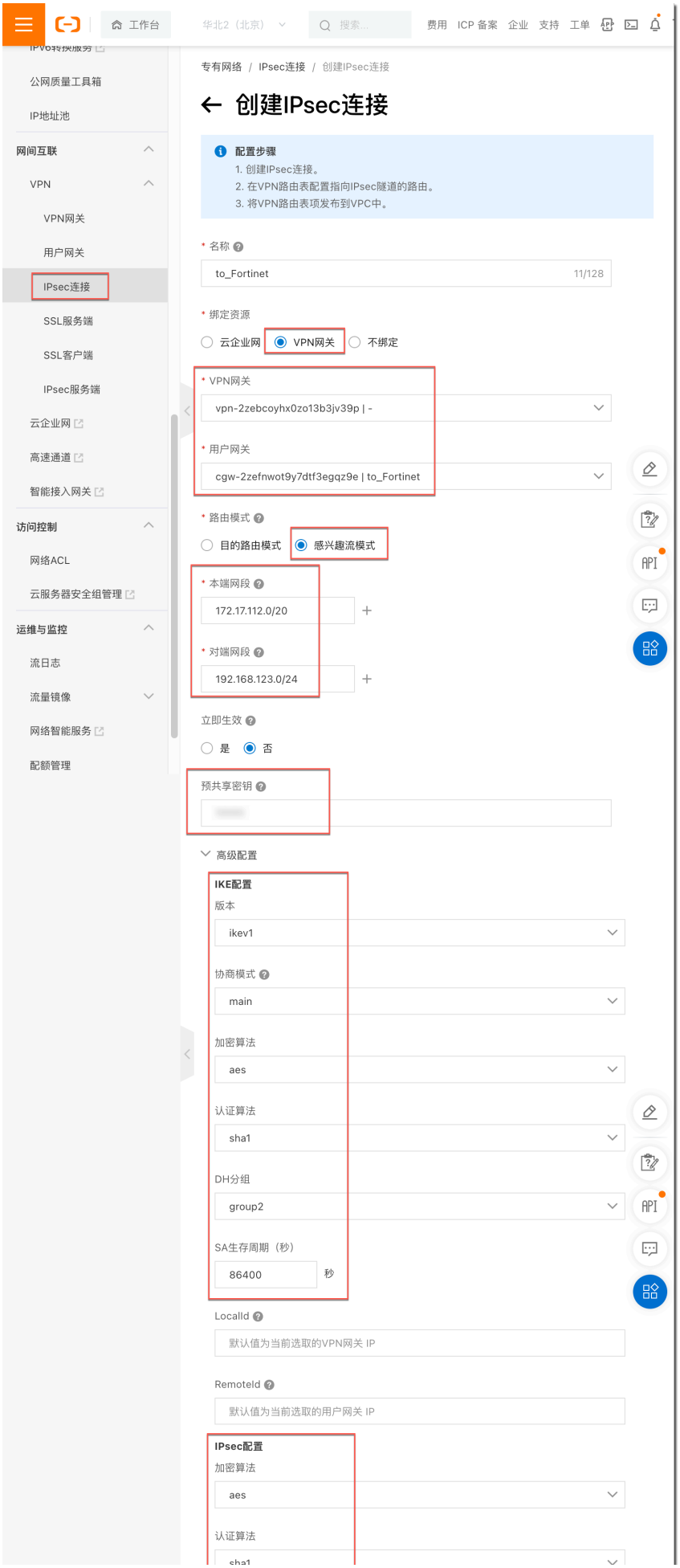
2. 进入用户网关页面，创建用户网关，即 FortiGate 的公网地址。

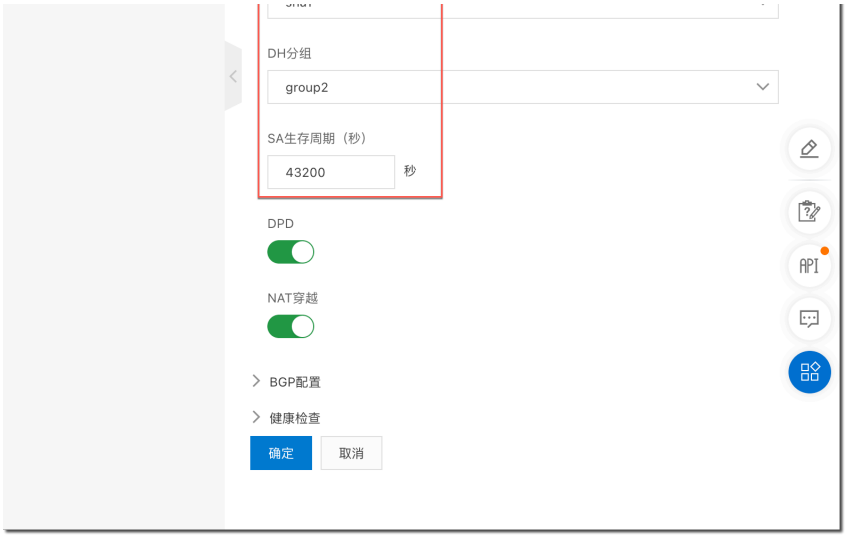


3. 进入 IPSec 连接页面，创建 IPSec 连接。



4. 选择绑定资源为 VPN 网关，分别选择第 2 步和第 3 步创建的 VPN 网关和用户网关，路由模式选择感兴趣流，填写本端和对端的感兴趣流网段，一二阶段算法与重协商时间与 FortiGate 保持一致。


工作台
华北2（北京）
搜索...
费用
ICP 备案
企业
支持
工单
公网质量工具箱
IP地址池
网间互联
VPN
VPN网关
用户网关
IPsec连接
SSL服务端
SSL客户端
IPsec服务端
云企业网
高速通道
智能接入网关
访问控制
网络ACL
云服务器安全组管理
运维与监控
流日志
流量镜像
网络智能服务
配额管理
专有网络 / IPsec连接 / 创建IPsec连接
创建IPsec连接
配置步骤
1. 创建IPsec连接。
2. 在VPN路由表配置指向IPsec隧道的路由。
3. 将VPN路由表项发布到VPC中。
名称
to_Fortinet
11/128
绑定资源
云企业网
VPN网关
不绑定
VPN网关
vpn-2zebcoyhx0zo13b3jv39p | -
用户网关
cgw-2zefnwot9y7dtf3egqz9e | to_Fortinet
路由模式
目的路由模式
感兴趣流模式
本端网段
172.17.112.0/20
对端网段
192.168.123.0/24
立即生效
是
否
预共享密钥
高级配置
IKE配置
版本
ikev1
协商模式
main
加密算法
aes
认证算法
sha1
DH分组
group2
SA生存周期（秒）
86400
秒
LocalId
默认值为当前选取的VPN网关 IP
RemoteId
默认值为当前选取的用户网关 IP
IPsec配置
加密算法
aes
认证算法
sha1



5. 点击确定按钮，会弹出是否去 VPN 网关中发布路由的提示，选择确定，则会根据感兴趣流自动添加路由到 VPN 网关。

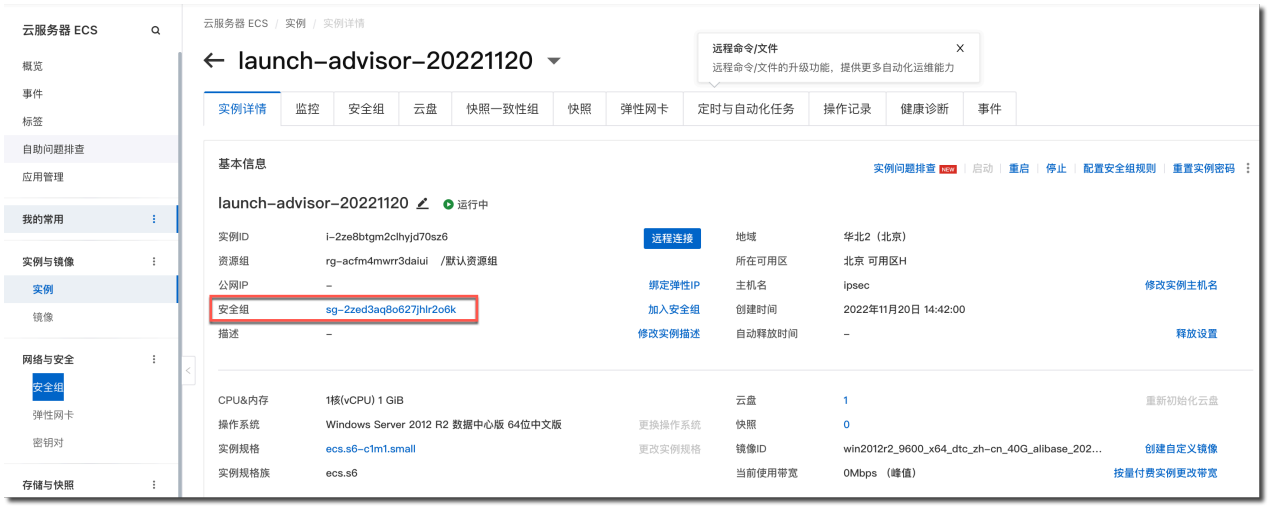


6. 进入 VPN 网关，编辑第 1 步中创建的 VPN 网关，在下方的策略路由表中可以看到针对 VPN 感兴趣流的策略路由，点击发布按钮发布路由，使其状态变为已发布。





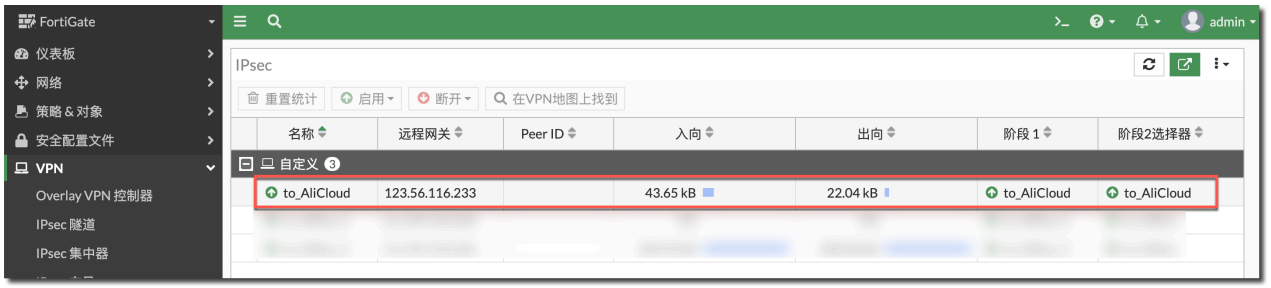
7. 配置 ECS 实例的安全组，允许出方向和入方向的流量（根据实际需求配置）。





结果验证

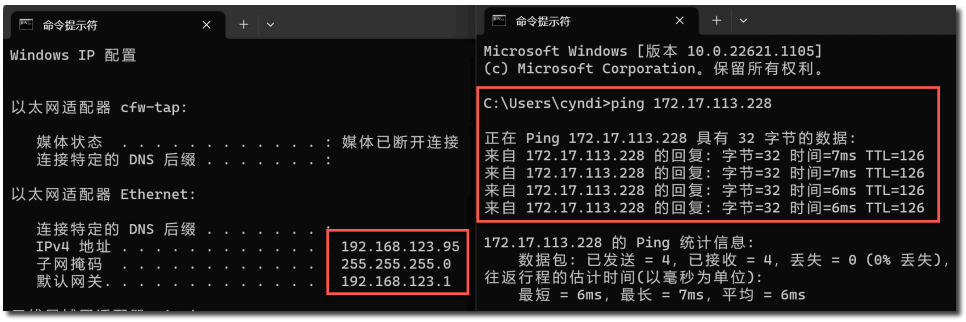
1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 123.112.245.1, wan, [1/0]
C       192.168.123.0/24 is directly connected, lan
S       172.17.112.0/20 [10/0] via to_AliCloud tunnel 123.56.116.233, [1/0]
C       123.112.245.0/24 is directly connected, wan
```

3. FortiGate 侧 PC 业务测试。



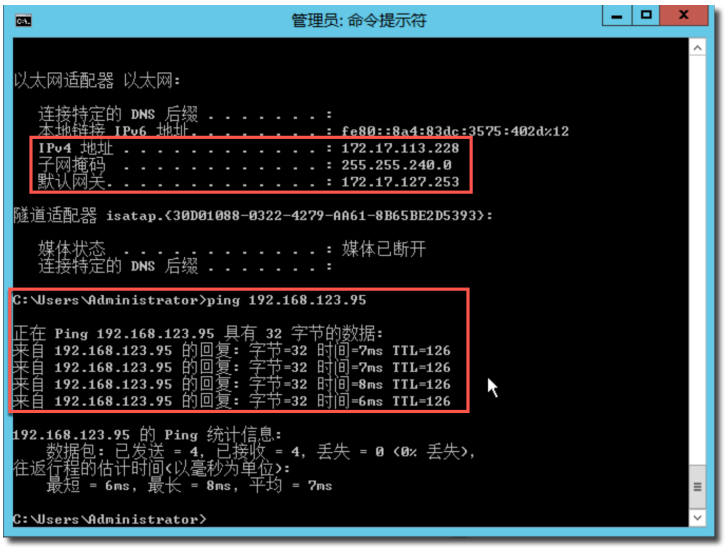
4. FortiGate 查看隧道建立状态，可以看到，FortiGate 成为了 initiator，这是正确的状态，如果成为 responder，则会出现注意事项中提到的问题（配置 Local-in Policy 就是为了防止 FortiGate 成为 responder）。

```
FortiGate # diagnose vpn ike gateway list name to_AliCloud

vd: root/0
name: to_AliCloud
version: 1
interface: wan 5
addr: 123.112.245.93:500 -> 123.56.116.233:500
tun_id: 123.56.116.233/::123.56.116.233
remote_location: 0.0.0.0
network-id: 0
created: 9s ago
nat: me peer
IKE SA: created 1/1  established 1/1  time 40/40/40 ms
IPsec SA: created 1/1  established 1/1  time 70/70/70 ms

  id/spi: 1544 dd588fcaa7cdae73/f47afb7c48f8e129
  direction: initiator
  status: established 9-9s ago = 40ms
  proposal: aes128-sha1
  key: b12f333e43d7d263-03a98c2d39015ba5
  lifetime/rekey: 86400/86090
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list name to_AliCloud
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=to_AliCloud ver=1 serial=18 123.112.245.93:500->123.56.116.233:500 tun_id=123.56.116.233 tun_id6=::123.56.116.233
dst_mtu=1500 dpd-link=on weight=1
bound_if=5 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=6 ilast=7 olast=1 ad=/0
stat: rxp=116 txp=129 rxb=16390 txb=9310
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=5
natt: mode=keepalive draft=32 interval=10 remote_port=500
proxyid=to_AliCloud proto=0 sa=1 ref=3 serial=1 auto-negotiate
  src: 0:192.168.123.0-192.168.123.255:0
  dst: 0:172.17.112.0-172.17.127.255:0
  SA:  ref=6 options=18227 type=00 soft=0 mtu=1422 expire=42774/0B replaywin=2048
       seqno=80 esn=0 replaywin_lastseq=00000073 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=688c1754 esp=aes key=16 2d9485697de2a31671bffbd55a97d91d
       ah=sha1 key=20 ccd0cdfea070180dabba59888f50d6470aaef88d
  enc: spi=4cba32a0 esp=aes key=16 495332b764e20e44523f5de593c14806
       ah=sha1 key=20 a3880ddf07c940ccc18ec827d0f1329094389c16
  dec:pkts/bytes=117/16450, enc:pkts/bytes=131/9566
  npu_flag=03 npu_rgwy=123.56.116.233 npu_lgwy=192.168.123.175 npu_selid=19 dec_npuid=1 enc_npuid=1
run_tally=0
```

5. 阿里云侧查看 IPSec 隧道建立状态，可以看到第二阶段已经建立成功。



6. 阿里云侧 ECS 实例业务测试。



7. FortiGate 侧抓包查看。

```
FortiGate # dia sni pa any "host 172.17.113.228 or host 123.56.116.223" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 172.17.113.228 or host 123.56.116.223]
2.729115 to_AliCloud out 192.168.123.95 → 172.17.113.228: icmp: echo request
2.729136 wan out 123.112.245.93 → 123.56.116.223: ESP(spi=0×4cba32a0,seq=0×7a)
2.730083 wan in 123.56.116.223 → 123.112.245.93: ESP(spi=0×688c1754,seq=0×74)
2.730098 to_AliCloud in 172.17.113.228 → 192.168.123.95: icmp: echo reply
3.729410 to_AliCloud out 192.168.123.95 → 172.17.113.228: icmp: echo request
3.729430 wan out 123.112.245.93 → 123.56.116.223: ESP(spi=0×4cba32a0,seq=0×7b)
3.730364 wan in 123.56.116.223 → 123.112.245.93: ESP(spi=0×688c1754,seq=0×75)
3.730382 to_AliCloud in 172.17.113.228 → 192.168.123.95: icmp: echo reply
4.729580 to_AliCloud out 192.168.123.95 → 172.17.113.228: icmp: echo request
4.729600 wan out 123.112.245.93 → 123.56.116.223: ESP(spi=0×4cba32a0,seq=0×7c)
4.730377 wan in 123.56.116.223 → 123.112.245.93: ESP(spi=0×688c1754,seq=0×76)
4.730397 to_AliCloud in 172.17.113.228 → 192.168.123.95: icmp: echo reply
5.729921 to_AliCloud out 192.168.123.95 → 172.17.113.228: icmp: echo request
5.729941 wan out 123.112.245.93 → 123.56.116.223: ESP(spi=0×4cba32a0,seq=0×7d)
5.731016 wan in 123.56.116.223 → 123.112.245.93: ESP(spi=0×688c1754,seq=0×77)
5.731031 to_AliCloud in 172.17.113.228 → 192.168.123.95: icmp: echo reply
6.730170 to_AliCloud out 192.168.123.95 → 172.17.113.228: icmp: echo request
6.730191 wan out 123.112.245.93 → 123.56.116.223: ESP(spi=0×4cba32a0,seq=0×7e)
6.731200 wan in 123.56.116.223 → 123.112.245.93: ESP(spi=0×688c1754,seq=0×78)
6.731219 to_AliCloud in 172.17.113.228 → 192.168.123.95: icmp: echo reply
```

其他说明

1. 请注意与阿里云建立 IPSec VPN 时，如果 FortiGate 位于 NAT 后（接口为私网地址），则需要在 FortiGate 的 IPSec 一阶段中指定 Local-id，id 类型为 address，地址为 FortiGateNAT 后的公网地址。

```
config vpn ipsec phase1-interface
        edit to_AliCloud
                set localid-type address
                set localid 123.112.245.93     //此处填写NAT后的公网地址
    next
```

```
end
```

2. 如果要使用 IKEv2，只需要将两端一阶段配置中的 IKE 版本修改为 IKEv2 即可，同样的，需要保证两端的安全算法一致。

3. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 172.17.113.228 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭（VPN 流量较大的情况下谨慎开启）：

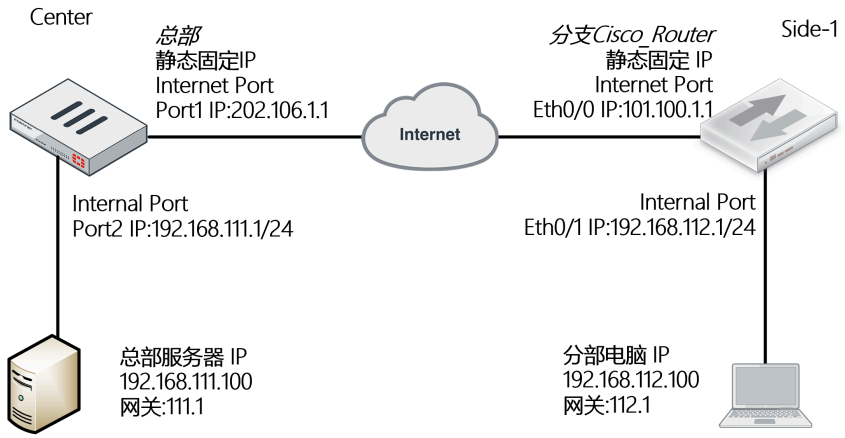
     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit to_AliCloud
     FortiGate (to_AliCloud) # set npu-offload disable
     FortiGate (to_AliCloud) # end
     ```

与思科路由器建立 IPSec VPN (IKE v1)

网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.111.0/24 与 192.168.112.0/24 两个网段的通信。与思科路由器进行 IPsec VPN（IKE v1）对接。

网络拓扑



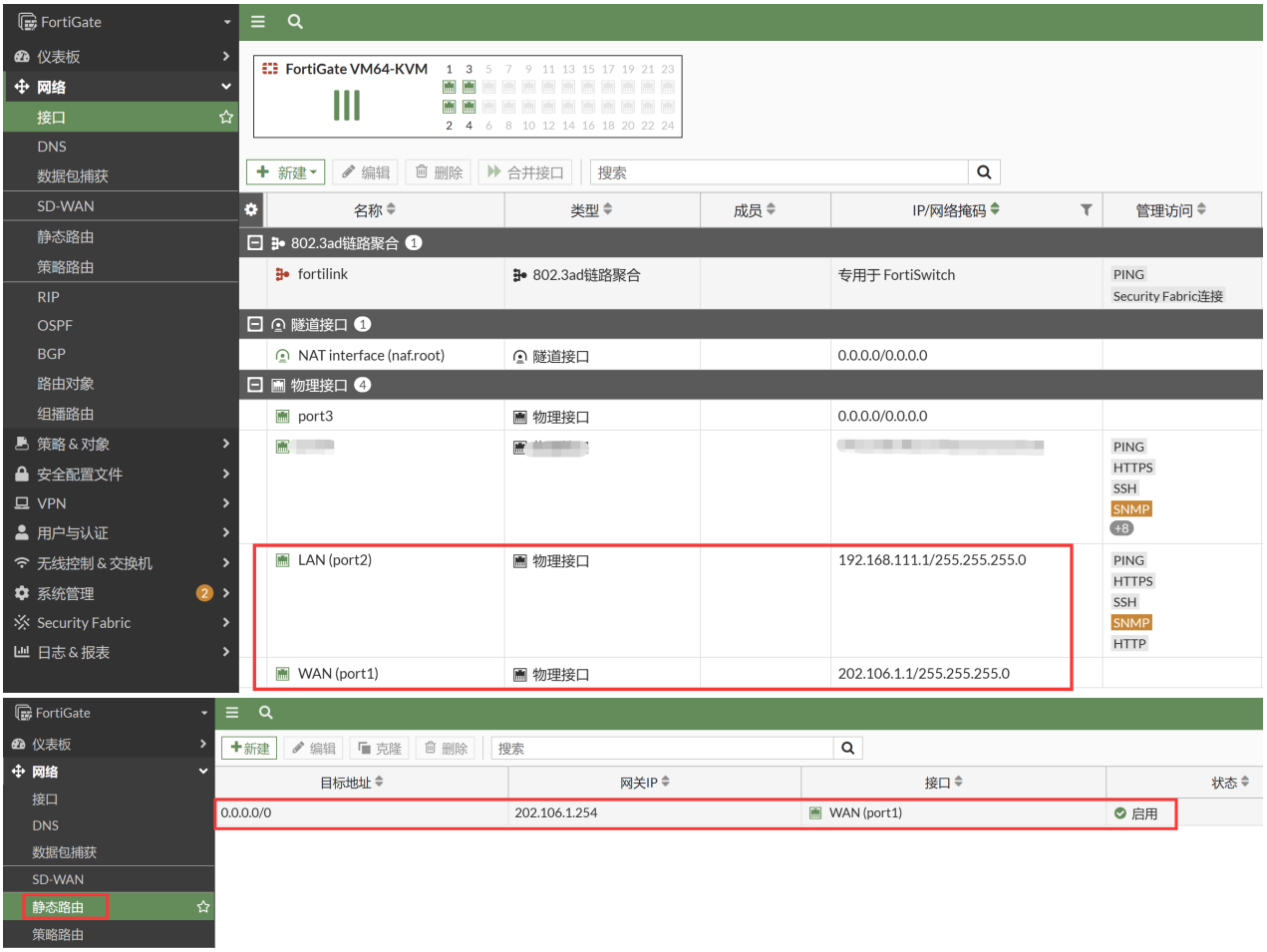
配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Cisco_Router
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略

配置步骤

FortiGate

1. 基本上网配置。



2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。

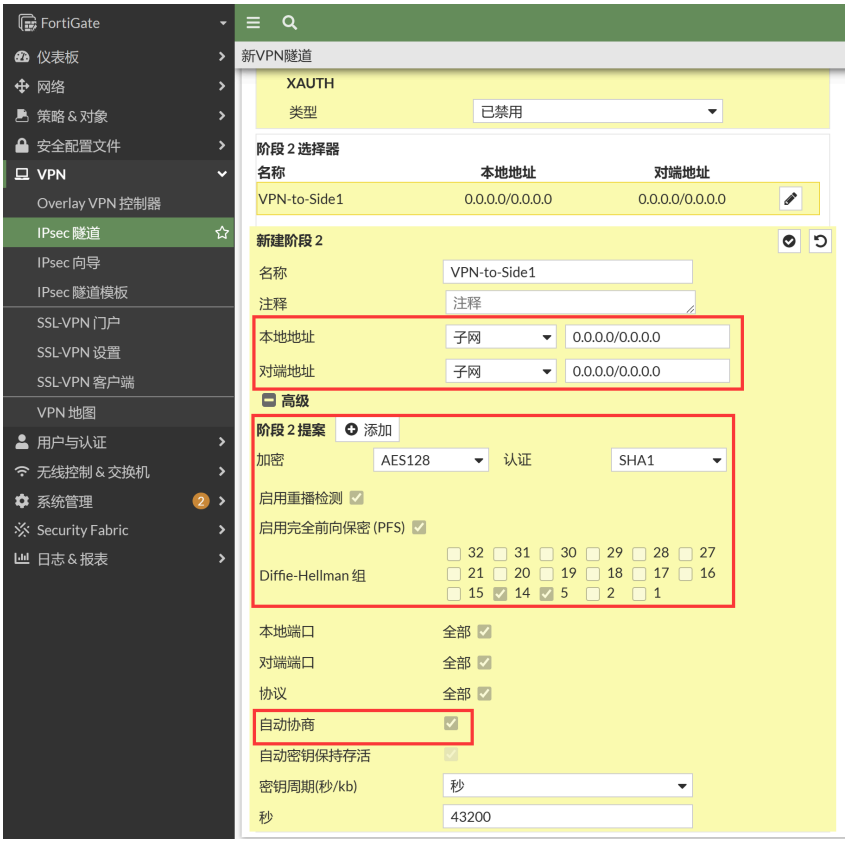


4. 配置网络、认证、第一阶段。



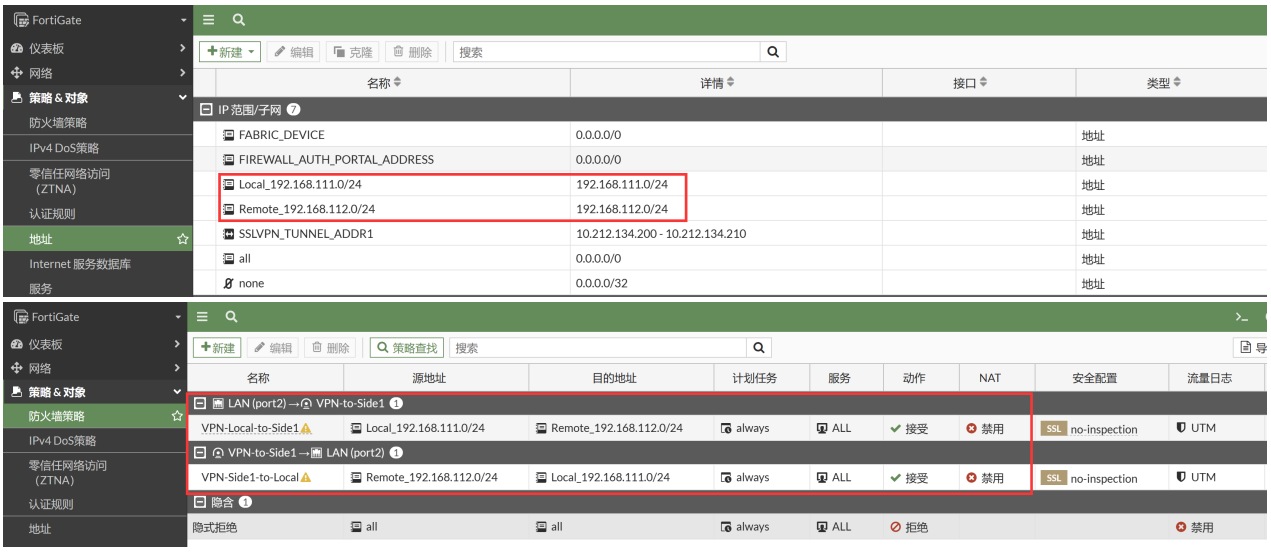

```
config vpn ipsec phase1-interface
    edit "VPN-to-Side1"
        set interface "port1"
        set peertype any
        set net-device disable
        set proposal aes256-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set remote-gw 101.100.1.1
        set psksecret ENC
/HISpjoxUQ1RoRXahwT2OpZmuyYsDY818arB4Q93PztlW0oynlDWuKIk1WIAXFLNldoVFlgOOjBk9Bf9tc6LdI/BeD6CmRvgokNIKa5fVnVgxXLzHM9fPGvt
RXb362jpxZkMxuKHcNGVrJCimmjVGAS+gixHgh8mRLj5kaMAbyhNX1odwNNgxYcxqap5iAwncL3ZzQ==
    next
end
```

5. 配置第二阶段。




```
config vpn ipsec phase2-interface
    edit "VPN-to-Side1"
        set phase1name "VPN-to-Side1"
        set proposal aes128-sha1
        set auto-negotiate enable
    next
end
```

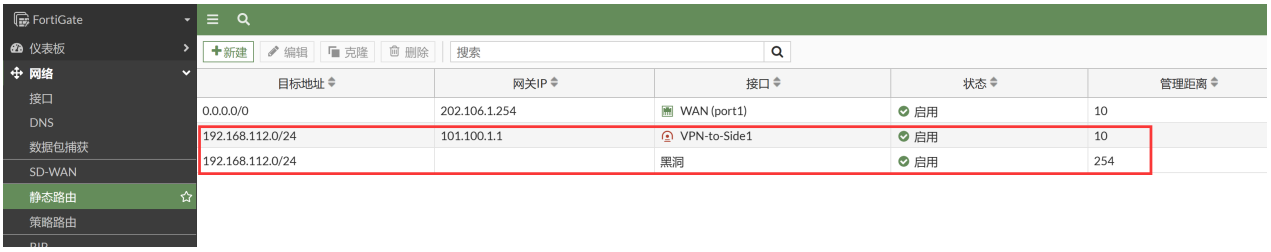
6. 配置 VPN 相关的网段地址对象和防火墙策略。




```
config firewall address
    edit "Local_192.168.111.0/24"
        set uuid 3bff7c90-8e7b-51ed-5311-ca9abc900a91
        set subnet 192.168.111.0 255.255.255.0
    next
    edit "Remote_192.168.112.0/24"
        set uuid 4647b500-8e7b-51ed-ea6f-c0ec2647bb42
        set subnet 192.168.112.0 255.255.255.0
```

```
    next
end

config firewall policy
    edit 1
        set name "VPN-Local-to-Side1"
        set uuid 7d1e7ce4-8e7b-51ed-19ea-001e373a03f3
        set srcintf "port2"
        set dstintf "VPN-to-Side1"
        set action accept
        set srcaddr "Local_192.168.111.0/24"
        set dstaddr "Remote_192.168.112.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "VPN-Side1-to-Local"
        set uuid 96c924d2-8e7b-51ed-f2d1-a61215d677f8
        set srcintf "VPN-to-Side1"
        set dstintf "port2"
        set action accept
        set srcaddr "Remote_192.168.112.0/24"
        set dstaddr "Local_192.168.111.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 配置 VPN 业务网段的静态路由。




```
config router static
    edit 1
        set gateway 202.106.1.254
        set device "port1"
    next
    edit 2
        set dst 192.168.112.0 255.255.255.0
        set device "VPN-to-Side1"
    next
    edit 3
        set dst 192.168.112.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Cisco Router

1. 基本上网配置。

```
 interface Ethernet0/0
 ip address 101.100.1.1 255.255.255.0
 ip nat outside
 no shutdown
 !
 interface Ethernet0/1
 ip address 192.168.112.1 255.255.255.0
 ip nat inside
 no shutdown
```

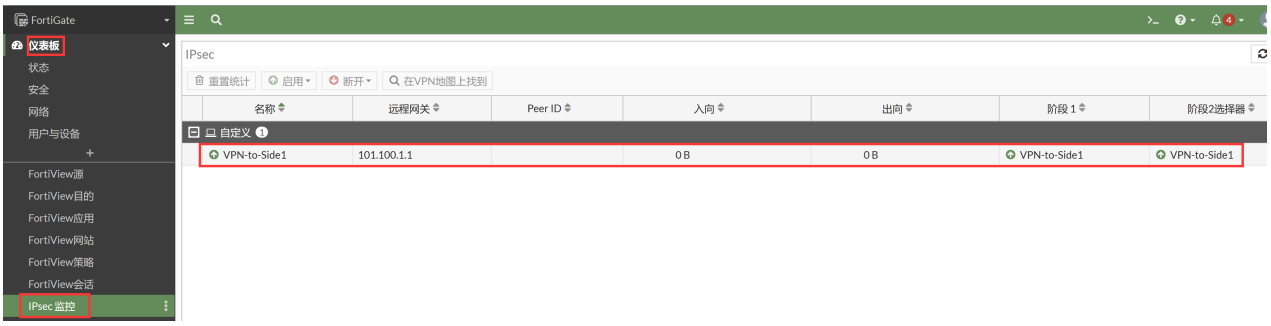
```
 !
 ip nat inside source list 101 interface Ethernet0/0 overload
 ip route 0.0.0.0 0.0.0.0 101.100.1.254
 !
 access-list 101 permit ip any any
```

2. IPsec IKE v1 的配置。

```
crypto isakmp policy 10
 encr aes 256
 hash sha
 authentication pre-share
 group 5
 !
 crypto isakmp key Fortinet123# address 202.106.1.1
 crypto isakmp keepalive 10 periodic
 !
 crypto ipsec transform-set MY-Trans esp-aes esp-sha-hmac
  mode tunnel
 !
 crypto ipsec profile MY-IPsec-Pro
  set transform-set MY-Trans
  set pfs group5
 !
 interface Tunnel0
  ip address 1.1.1.2 255.255.255.0
  ip mtu 1400
  tunnel source Ethernet0/0
  tunnel mode ipsec ipv4
  tunnel destination 202.106.1.1
  tunnel protection ipsec profile MY-IPsec-Pro
 !
 ip route 192.168.111.0 255.255.255.0 Tunnel0 1.1.1.1
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.106.1.254, port1, [1/0]
C       192.168.111.0/24 is directly connected, port2
S       192.168.112.0/24 [10/0] via VPN-to-Side1 tunnel 101.100.1.1, [1/0]
C       202.106.1.0/24 is directly connected, port1
```

3. FortiGate 侧业务测试。

```
FortiGate # execute ping-options source 192.168.111.1
FortiGate # execute ping 192.168.112.1
PING 192.168.112.1 (192.168.112.1): 56 data bytes
64 bytes from 192.168.112.1: icmp_seq=0 ttl=255 time=1.0 ms
64 bytes from 192.168.112.1: icmp_seq=1 ttl=255 time=1.2 ms
64 bytes from 192.168.112.1: icmp_seq=2 ttl=255 time=0.9 ms
64 bytes from 192.168.112.1: icmp_seq=3 ttl=255 time=1.1 ms
64 bytes from 192.168.112.1: icmp_seq=4 ttl=255 time=1.0 ms
--- 192.168.112.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.0/1.2 ms
```

4. FortiGate 查看隧道建立状态。

```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Side1
version: 1
interface: port1 3
addr: 202.106.1.1:500 → 101.100.1.1:500
tun_id: 101.100.1.1/::101.100.1.1
remote_location: 0.0.0.0
network-id: 0
created: 449s ago
IKE SA: created 1/2  established 1/1  time 0/0/0 ms
IPsec SA: created 2/2  established 1/1  time 0/0/0 ms
  id/spi: 58 584e6c9052795a25/a433e7123ece6706
  direction: responder
  status: established 423-423s ago = 0ms
  proposal: aes256-sha1
  key: 447062de7f9b419b-37148f8d51b0e3be-dab2d73c40ed2fcc-48619811c9f95b96
  lifetime/rekey: 86400/85706
  DPD sent/recv: 00000000/6f07eaf6

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Side1 ver=1 serial=1 202.106.1.1:0→101.100.1.1:0 tun_id=101.100.1.1 tun_id6=::101.100.1.1 dst_mtu=1500 dpd-
link=on weight=1
bound_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overla0
proxyid_num=1 child_num=0 refcnt=4 ilast=5 olast=5 ad=/0
stat: rxp=110 txp=115 rxb=10840 txb=11260
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=2
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Side1 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=2905/0B replaywin=2048
       seqno=74 esn=0 replaywin_lastseq=0000006e qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3333/3600
  dec: spi=148a81f0 esp=aes key=16 810f57e8981275c305be8c7fe3e6231d
       ah=sha1 key=20 7f9b004e1340ab3614dc359e50bf1ba5110ca6b6
  enc: spi=41a02d8a esp=aes key=16 81e238bdfb77856910326d4b40ac4bef
       ah=sha1 key=20 8785688516cc98c13b79d0008333db7b0a6f6dc7
  dec:pkts/bytes=220/21680, enc:pkts/bytes=230/30340
  npu_flag=00 npu_rgwy=101.100.1.1 npu_lgwy=202.106.1.1 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

5. Cisco_Router 侧业务测试。

```
Router#ping 192.168.111.1 source 192.168.112.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.111.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.112.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/4/7 ms
```

```
Router#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
202.106.1.1     101.100.1.1     QM_IDLE           1002 ACTIVE
```

6. FortiGate 侧抓包查看。

```
FortiGate # dia sni pa any "host 192.168.112.1 or host 101.100.1.1" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.112.1 or host 101.100.1.1]
2.729115 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
2.729136 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×41a02d8a,seq=0×7a)
2.730083 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a81f0,seq=0×74)
2.730098 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
3.729410 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
3.729430 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×41a02d8a,seq=0×7b)
3.730364 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a81f0,seq=0×75)
3.730382 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
4.729580 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
4.729600 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×41a02d8a,seq=0×7c)
4.730377 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a81f0,seq=0×76)
4.730397 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
5.729921 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
5.729941 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×41a02d8a,seq=0×7d)
5.731016 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a81f0,seq=0×77)
5.731031 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
6.730170 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
6.730191 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×41a02d8a,seq=0×7e)
6.731200 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a81f0,seq=0×78)
6.731219 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
     FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
     FortiGate (BJ-OSPF-TO-SH) # end
     ```

2. 附 Cisco Router IPsec VPN 传统的 Crypto MAP 的配置方式（上述为 VTI 方式的 IPsec VPN）。

```
FortiGate二阶段配置：

config vpn ipsec phase2-interface
    edit "VPN-TO-Side1"
        set phase1name "VPN-TO-Side1"
        set proposal 3des-md5 aes128-sha1
        set auto-negotiate enable
        set src-subnet 192.168.111.0 255.255.255.0
        set dst-subnet 192.168.112.0 255.255.255.0
    next
```

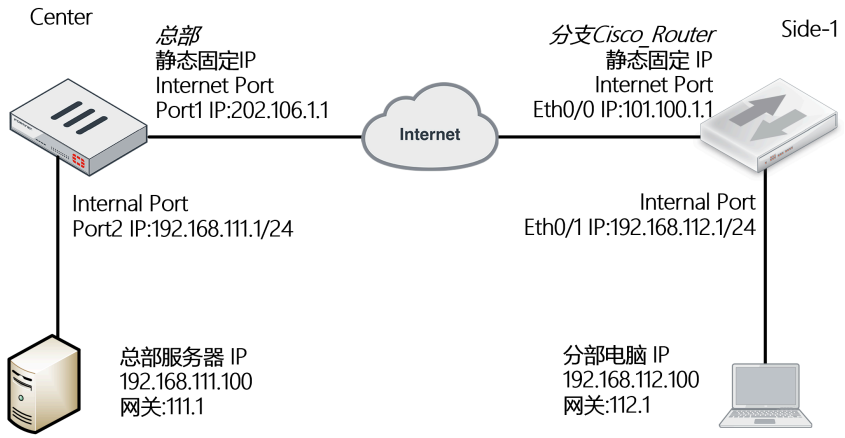
```
end

Cisco_Router 配置:

hostname Side_Router
!
interface Ethernet0/0
ip address 101.100.1.1 255.255.255.0
ip nat outside
crypto map MY_MAP
!
interface Ethernet0/1
ip address 192.168.112.1 255.255.255.0
ip nat inside
!
ip route 0.0.0.0 0.0.0.0 101.100.1.254

access-list 101 deny ip 192.168.112.0 0.0.0.255 192.168.111.0 0.0.0.255  //NAT Bypass掉VPN的流量
access-list 101 permit ip any any

ip nat inside source list 101 interface Ethernet0/0 overload
!
!
crypto isakmp policy 10
 encr aes 256
 hash sha
 authentication pre-share
 group 5

crypto isakmp key Fortinet123# address 202.106.1.1
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec transform-set MY-Trans esp-aes esp-sha-hmac
 mode tunnel
!
access-list 102 permit ip 192.168.112.0 0.0.0.255 192.168.111.0 0.0.0.255
!
crypto map MY_MAP 10 ipsec-isakmp
 set peer 202.106.1.1
 set transform-set MY-Trans
 set pfs group5
 match address 102
```

与思科路由器建立 IPSec VPN (IKE v2)

网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.111.0/24 与 192.168.112.0/24 两个网段的通信。与思科路由器进行 IPsec VPN（IKE v2）对接。

网络拓扑



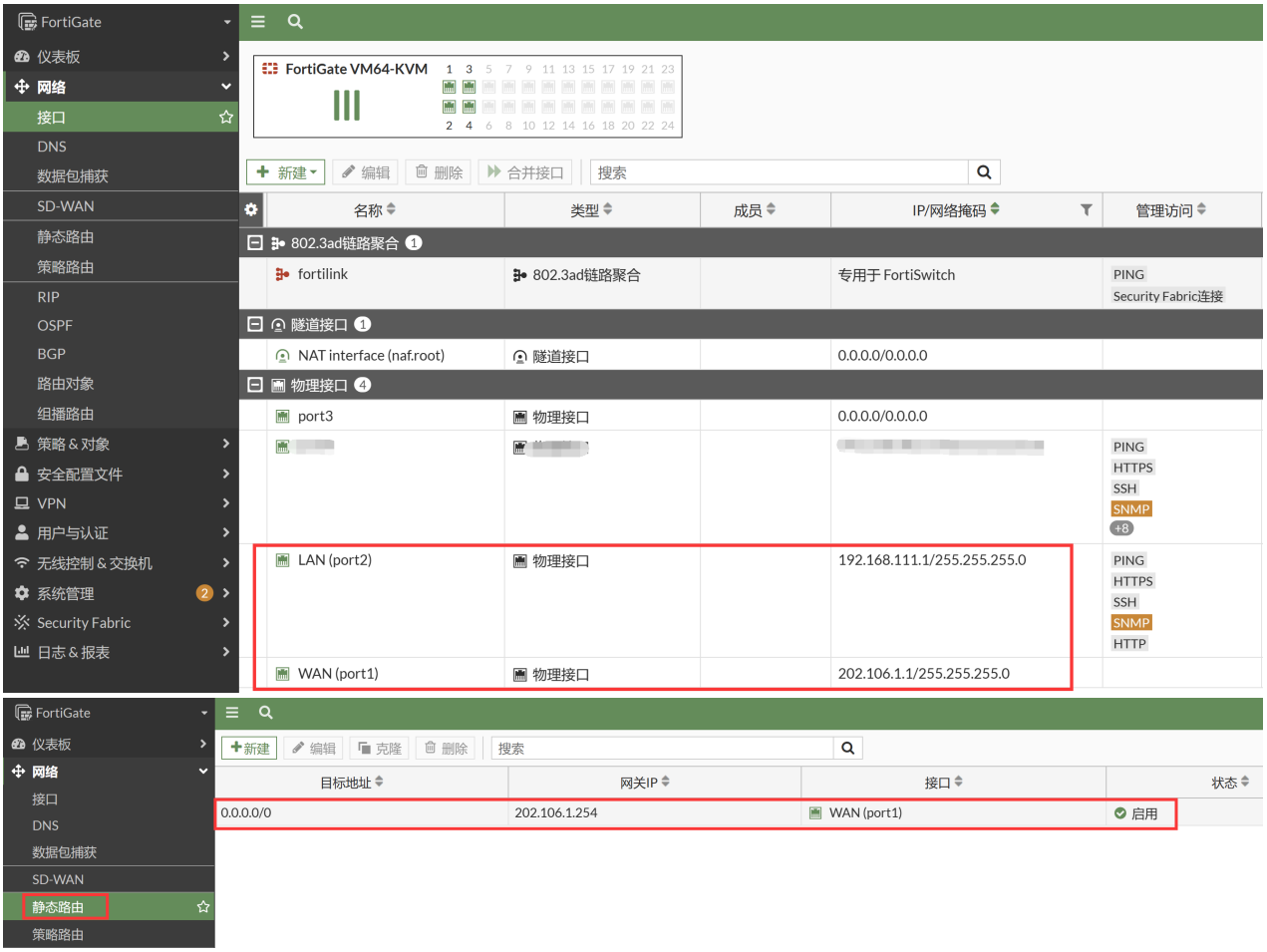
配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Cisco_Router
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略

配置步骤

FortiGate

1. 基本上网配置。



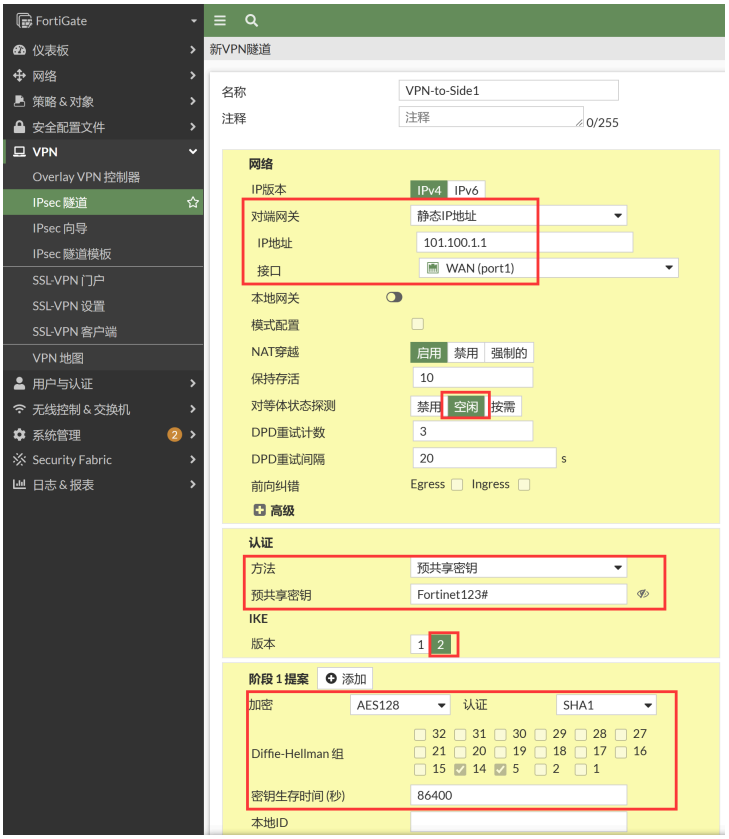
2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。

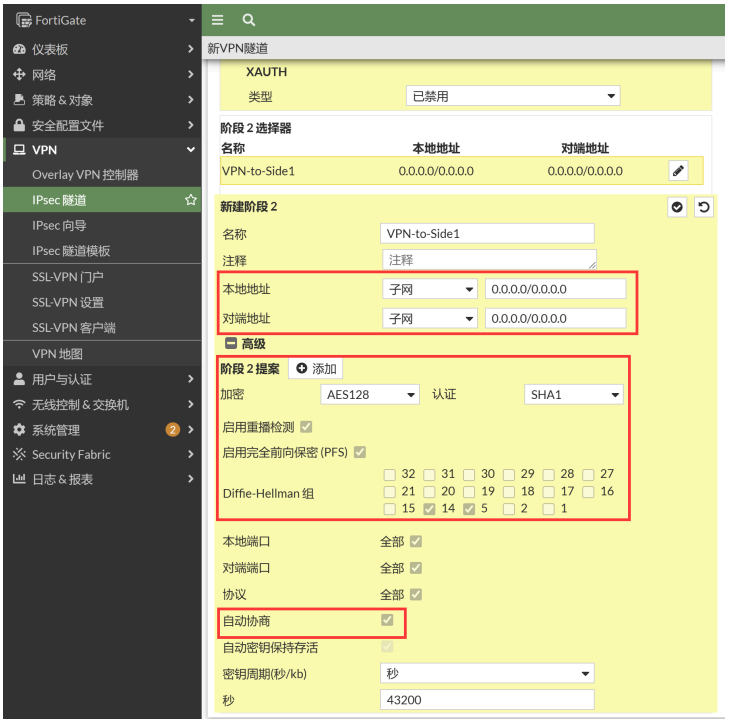


4. 配置网络、认证、第一阶段。



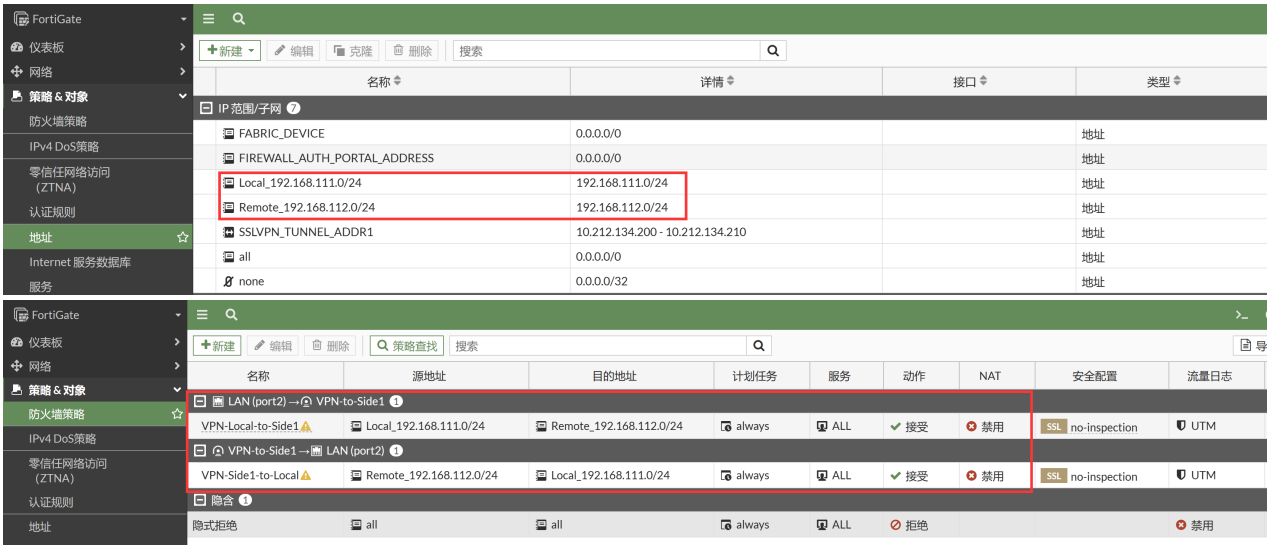

```
config vpn ipsec phase1-interface
    edit "VPN-to-Side1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
        set remote-gw 101.100.1.1
        set psksecret ENC
/HISpjoxUQ1RoRXahwT2OpZmuyYsDY818arB4Q93PztlW0oynlDWuKIk1WIAXFLNldoVFlgOOjBk9Bf9tc6LdI/BeD6CmRvgokNIKa5fVnVgxXLzHM9fPGvt
RXb362jpxZkMxuKHcNGVrJCimmjVGAS+gixHgh8mRLj5kaMAbyhNX1odwNNgxYcxqap5iAwncL3ZzQ==
    next
end
```

5. 配置第二阶段。




```
config vpn ipsec phase2-interface
    edit "VPN-to-Side1"
        set phase1name "VPN-to-Side1"
        set proposal aes128-sha1
        set auto-negotiate enable
    next
end
```

6. 配置 VPN 相关的网段地址对象和防火墙策略。




```
config firewall address
    edit "Local_192.168.111.0/24"
        set uuid 3bff7c90-8e7b-51ed-5311-ca9abc900a91
        set subnet 192.168.111.0 255.255.255.0
    next
    edit "Remote_192.168.112.0/24"
        set uuid 4647b500-8e7b-51ed-ea6f-c0ec2647bb42
        set subnet 192.168.112.0 255.255.255.0
    next
end

config firewall policy
    edit 1
```

```
        set name "VPN-Local-to-Side1"
        set uuid 7d1e7ce4-8e7b-51ed-19ea-001e373a03f3
        set srcintf "port2"
        set dstintf "VPN-to-Side1"
        set action accept
        set srcaddr "Local_192.168.111.0/24"
        set dstaddr "Remote_192.168.112.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "VPN-Side1-to-Local"
        set uuid 96c924d2-8e7b-51ed-f2d1-a61215d677f8
        set srcintf "VPN-to-Side1"
        set dstintf "port2"
        set action accept
        set srcaddr "Remote_192.168.112.0/24"
        set dstaddr "Local_192.168.111.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 配置 VPN 业务网段的静态路由。




```
config router static
    edit 1
        set gateway 202.106.1.254
        set device "port1"
    next
    edit 2
        set dst 192.168.112.0 255.255.255.0
        set device "VPN-to-Side1"
    next
    edit 3
        set dst 192.168.112.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Cisco Router

1. 基本上网配置。

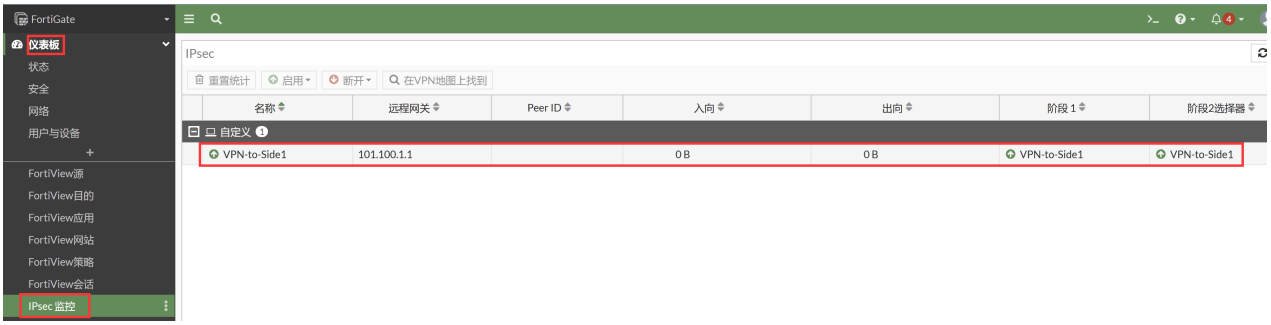
```
interface Ethernet0/0
ip address 101.100.1.1 255.255.255.0
ip nat outside
no shutdown
!
interface Ethernet0/1
ip address 192.168.112.1 255.255.255.0
ip nat inside
no shutdown
!
ip nat inside source list 101 interface Ethernet0/0 overload
ip route 0.0.0.0 0.0.0.0 101.100.1.254
!
access-list 101 permit ip any any
```

2. IPsec IKE v2 的配置。

```
crypto ikev2 proposal MY-IKEV2-Proposal
 encryption aes-cbc-128
 integrity sha1
 group 5 14
 !
 crypto ikev2 policy MY_IKEV2_Policy
 proposal MY-IKEV2-Proposal
 !
 crypto ikev2 keyring MY_Keyring
 peer Center
   address 202.106.1.1
   pre-shared-key Fortinet123#
 !
 crypto ikev2 profile MY_IKEV2_Pro
 match identity remote address 202.106.1.1 255.255.255.255
 identity local address 101.100.1.1
 authentication remote pre-share
 authentication local pre-share
 keyring local MY_Keyring
 !
 crypto isakmp keepalive 10 periodic
 !
 crypto ipsec transform-set MY_Trans esp-aes esp-sha-hmac
 mode tunnel
 !
 crypto ipsec profile MY_IPsec_Pro
 set transform-set MY_Trans
 set pfs group5
 set ikev2-profile MY_IKEV2_Pro
 !
 interface Tunnel0
 ip address 1.1.1.2 255.255.255.0
 ip mtu 1400
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 202.106.1.1
 tunnel protection ipsec profile MY_IPsec_Pro
 !
 ip route 192.168.111.0 255.255.255.0 Tunnel0 1.1.1.1
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
```

```
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.106.1.254, port1, [1/0]
C       192.168.111.0/24 is directly connected, port2
S       192.168.112.0/24 [10/0] via VPN-to-Side1 tunnel 101.100.1.1, [1/0]
C       202.106.1.0/24 is directly connected, port1
```

3. FortiGate 侧业务测试。

```
FortiGate # execute ping-options source 192.168.111.1

FortiGate # execute ping 192.168.112.1
PING 192.168.112.1 (192.168.112.1): 56 data bytes
64 bytes from 192.168.112.1: icmp_seq=0 ttl=255 time=1.0 ms
64 bytes from 192.168.112.1: icmp_seq=1 ttl=255 time=1.2 ms
64 bytes from 192.168.112.1: icmp_seq=2 ttl=255 time=0.9 ms
64 bytes from 192.168.112.1: icmp_seq=3 ttl=255 time=1.1 ms
64 bytes from 192.168.112.1: icmp_seq=4 ttl=255 time=1.0 ms

--- 192.168.112.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.0/1.2 ms
```

4. 查看 FortiGate 隧道状态。

```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Side1
version: 2
interface: port1 3
addr: 202.106.1.1:500 → 101.100.1.1:500
tun_id: 101.100.1.1/::101.100.1.1
remote_location: 0.0.0.0
network-id: 0
created: 189s ago
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms
  id/spi: 371 c9520b1c2bc0e6c8/6aade7b829009948
  direction: initiator
  status: established 189-189s ago = 10ms
  proposal: aes128-sha1
  child: no
  SK_ei: 75237437cff25754-cf7db509c4681115
  SK_er: b32740a409713752-dee1a9e800220c20
  SK_ai: a51d7711574f8d4a-44ccb29646bb5227-6ea8727f
  SK_ar: 4e8c3e430012d7d7-bc227ae5d10bf045-205b82e5
  PPK: no
  message-id sent/recv: 11/0
  lifetime/rekey: 86400/85910
  DPD sent/recv: 0000000c/0000000c

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Side1 ver=2 serial=1 202.106.1.1:0→101.100.1.1:0 tun_id=101.100.1.1 tun_id6=::101.100.1.1 dst_mtu=1500 dpd-
link=on weight=1
bound_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overla0
proxyid_num=1 child_num=0 refcnt=4 ilast=3 olast=3 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=13
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Side1 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=42694/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=148a8320 esp=aes key=16 2c299d148b78322997ffe3948ea8a426
       ah=sha1 key=20 c713f9ec7cecb6b6ffc80ce6fc407e8dc565a56f
```

```
  enc: spi=2e03e37b esp=aes key=16 ef6e6e604f710a27393927db2be5c431
       ah=sha1 key=20 d80fc71d9b6947366693bff2a3c205b6b8b3b95d
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=101.100.1.1 npu_lgwy=202.106.1.1 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

5. Cisco_Router 侧业务测试。

```
Router#ping 192.168.111.1 source 192.168.112.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.111.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.112.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/4/7 ms

Router#show crypto ikev2 sa
 IPv4 Crypto IKEv2  SA

Tunnel-id Local                 Remote                fvrf/ivrf            Status
2         101.100.1.1/500       202.106.1.1/500       none/none            READY
      Encr: AES-CBC, keysize: 128, PRF: SHA1, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/261 sec
```

6. FortiGate 侧抓包查看。

```
FortiGate # dia sni pa any "host 192.168.112.1 or host 101.100.1.1" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.112.1 or host 101.100.1.1]
1.980972 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
1.981265 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×2e03e37b,seq=0×2)
1.982549 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a8320,seq=0×1)
1.982619 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
2.981743 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
2.981768 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×2e03e37b,seq=0×3)
2.982852 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a8320,seq=0×2)
2.982869 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
3.982291 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
3.982312 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×2e03e37b,seq=0×4)
3.983316 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a8320,seq=0×3)
3.983333 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
4.983018 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
4.983039 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×2e03e37b,seq=0×5)
4.984193 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a8320,seq=0×4)
4.984212 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
5.983838 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
5.983860 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×2e03e37b,seq=0×6)
5.984866 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×148a8320,seq=0×5)
5.984884 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

```
FortiGate # config vpn ipsec phase1-interface
FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
FortiGate (BJ-OSPF-TO-SH) # end
```

2. Cisco Router IPsec VPN 传统的 Crypto MAP 的配置方式（上述为 VTI 方式的 IPsec VPN）。

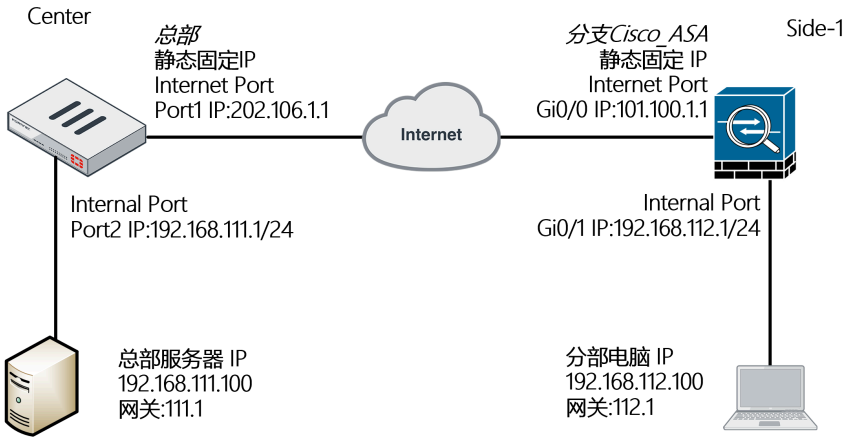
```
FortiGate 只需要修改IPsec VPN的第二阶段（添加明确的感兴趣流即可），其他的所有配置不变：
config vpn ipsec phase2-interface
    edit "VPN-TO-Side1"
        set phase1name "VPN-TO-Side1"
        set proposal 3des-md5 aes128-sha1
        set auto-negotiate enable
        set src-subnet 192.168.111.0 255.255.255.0
        set dst-subnet 192.168.112.0 255.255.255.0
    next
end

Cisco_Router配置：
hostname Side_Router
!
interface Ethernet0/0
ip address 101.100.1.1 255.255.255.0
ip nat outside
crypto map MY_MAP
!
interface Ethernet0/1
ip address 192.168.112.1 255.255.255.0
ip nat inside
!
ip route 0.0.0.0 0.0.0.0 101.100.1.254
access-list 101 deny ip 192.168.112.0 0.0.0.255 192.168.111.0 0.0.0.255 //NAT需bypass掉VPN流量
access-list 101 permit ip any any
ip nat inside source list 101 interface Ethernet0/0 overload
crypto ikev2 proposal MY-IKEV2-Proposal
encryption aes-cbc-128
integrity sha1
group 5 14
!
crypto ikev2 policy MY_IKEV2_Policy
proposal MY-IKEV2-Proposal
!
crypto ikev2 keyring MY_Keyring
peer Center
  address 202.106.1.1
  pre-shared-key Fortinet123#
!
crypto ikev2 profile MY_IKEV2_Pro
match identity remote address 202.106.1.1 255.255.255.255
identity local address 101.100.1.1
authentication remote pre-share
authentication local pre-share
keyring local MY_Keyring
!
crypto isakmp keepalive 10 periodic
!
crypto ipsec transform-set MY_Trans esp-aes esp-sha-hmac
mode tunnel
!
access-list 102 permit ip 192.168.112.0 0.0.0.255 192.168.111.0 0.0.0.255
!
crypto map MY_MAP 10 ipsec-isakmp
set peer 202.106.1.1
set transform-set MY_Trans
set pfs group5
set ikev2-profile MY_IKEV2_Pro
match address 102
```

与思科 ASA 防火墙建立 IPSec VPN (IKE v1)

网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.111.0/24 与 192.168.112.0/24 两个网段的通信。与思科 ASA 防火墙进行 IPsec VPN（IKE v1）对接。

网络拓扑



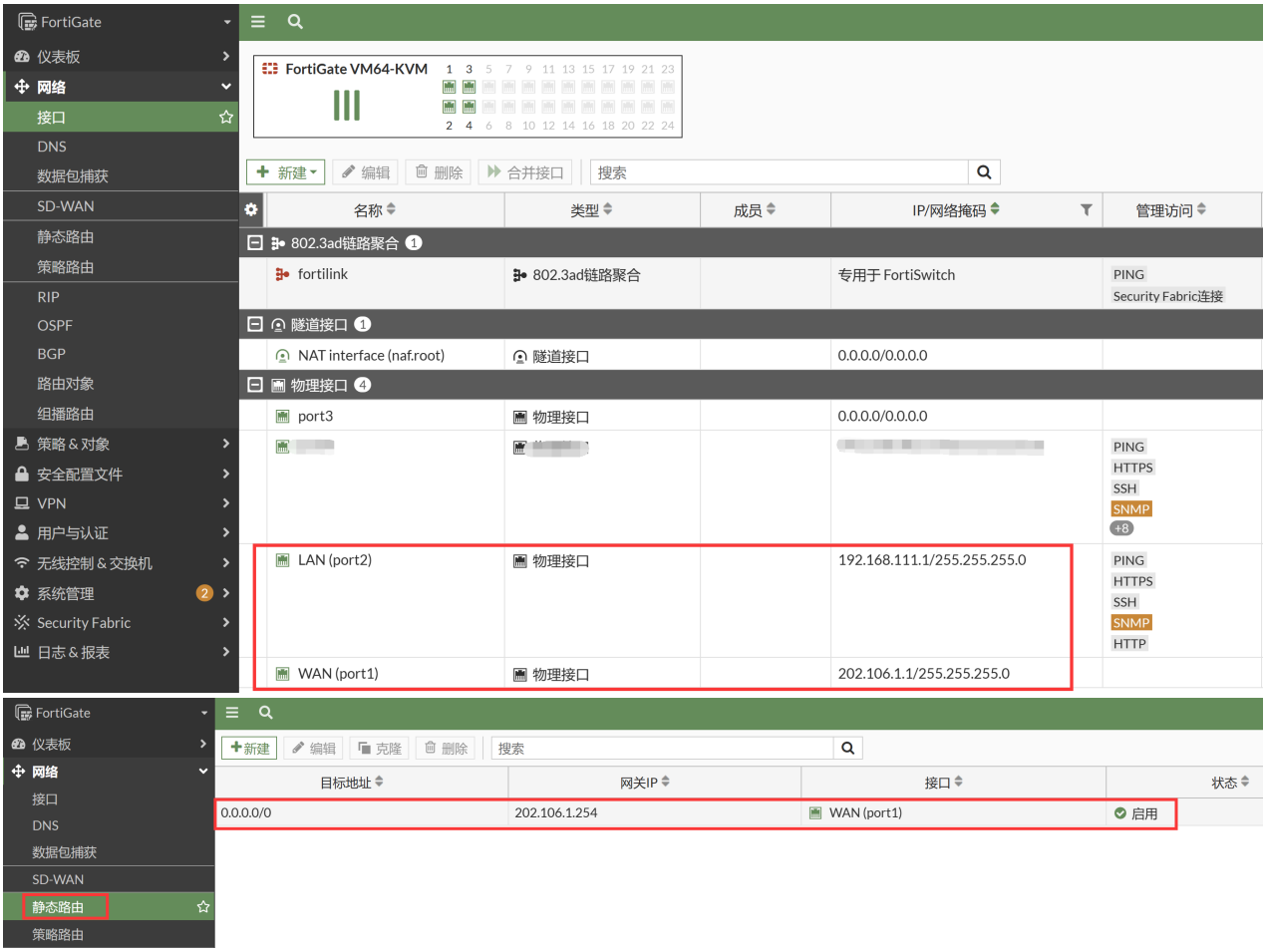
配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Cisco_ASA 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略

配置步骤

FortiGate

1. 基本上网配置。



2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。

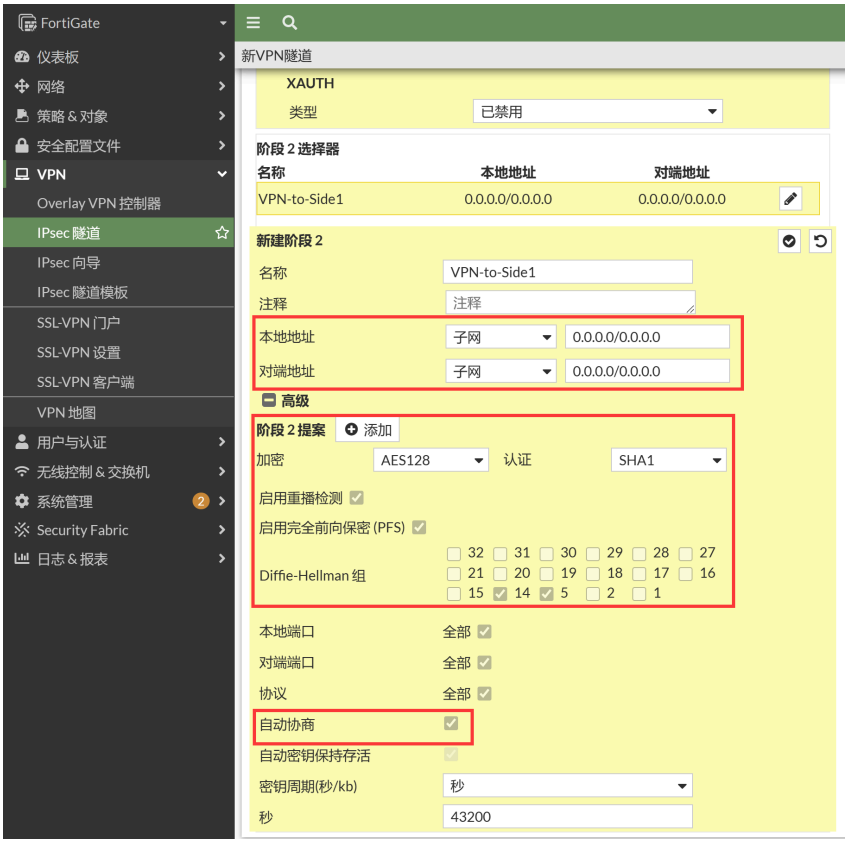


4. 配置网络、认证、第一阶段。



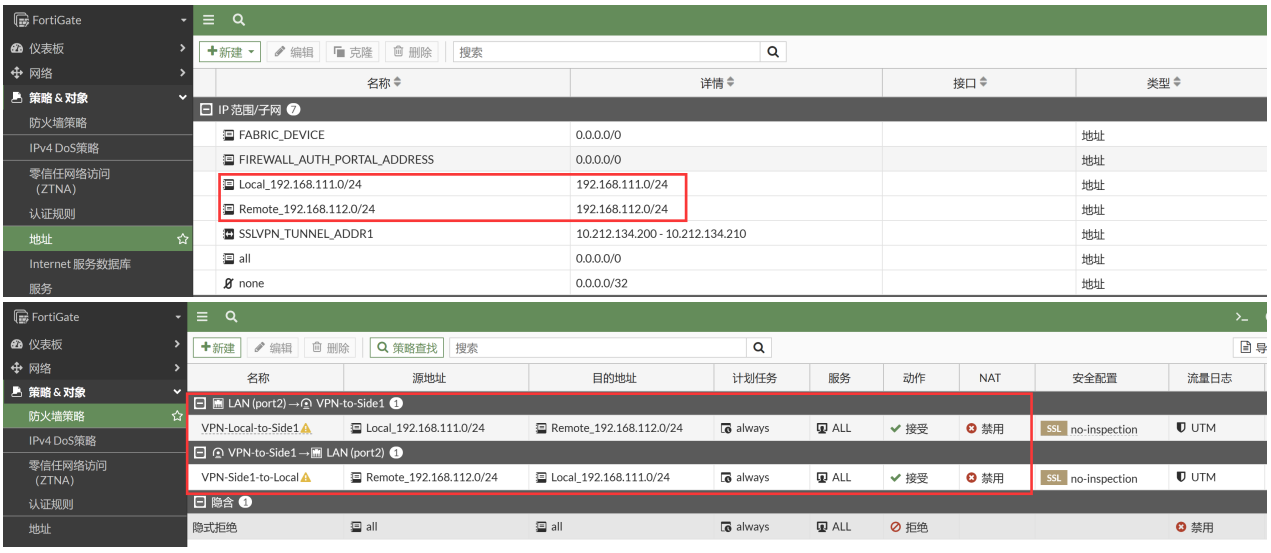

```
config vpn ipsec phase1-interface
    edit "VPN-to-Side1"
        set interface "port1"
        set peertype any
        set net-device disable
        set proposal aes256-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set remote-gw 101.100.1.1
        set psksecret ENC
/HISpjoxUQ1RoRXahwT2OpZmuyYsDY818arB4Q93PztlW0oynlDWuKIk1WIAXFLNldoVFlgOOjBk9Bf9tc6LdI/BeD6CmRvgokNIKa5fVnVgxXLzHM9fPGvt
RXb362jpxZkMxuKHcNGVrJCimmjVGAS+gixHgh8mRLj5kaMAbyhNX1odwNNgxYcxqap5iAwncL3ZzQ==
    next
end
```

5. 配置第二阶段。




```
config vpn ipsec phase2-interface
    edit "VPN-to-Side1"
        set phase1name "VPN-to-Side1"
        set proposal aes128-sha1
        set auto-negotiate enable
    next
end
```

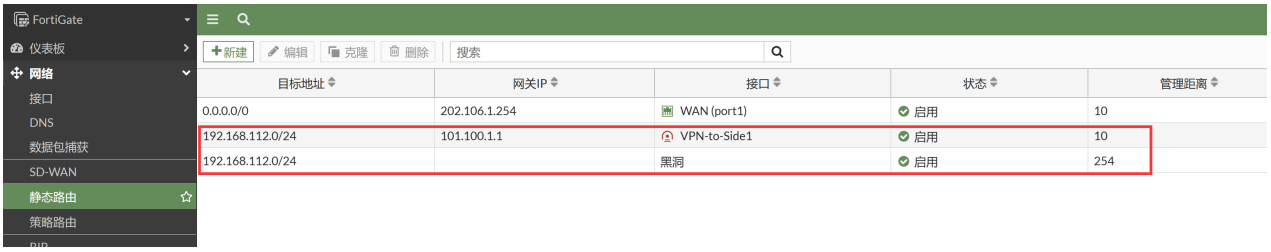
6. 配置 VPN 相关的网段地址对象和防火墙策略。




```
config firewall address
    edit "Local_192.168.111.0/24"
        set uuid 3bff7c90-8e7b-51ed-5311-ca9abc900a91
        set subnet 192.168.111.0 255.255.255.0
    next
    edit "Remote_192.168.112.0/24"
        set uuid 4647b500-8e7b-51ed-ea6f-c0ec2647bb42
        set subnet 192.168.112.0 255.255.255.0
```

```
    next
end

config firewall policy
    edit 1
        set name "VPN-Local-to-Side1"
        set uuid 7d1e7ce4-8e7b-51ed-19ea-001e373a03f3
        set srcintf "port2"
        set dstintf "VPN-to-Side1"
        set action accept
        set srcaddr "Local_192.168.111.0/24"
        set dstaddr "Remote_192.168.112.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "VPN-Side1-to-Local"
        set uuid 96c924d2-8e7b-51ed-f2d1-a61215d677f8
        set srcintf "VPN-to-Side1"
        set dstintf "port2"
        set action accept
        set srcaddr "Remote_192.168.112.0/24"
        set dstaddr "Local_192.168.111.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 配置 VPN 业务网段的静态路由。




```
config router static
    edit 1
        set gateway 202.106.1.254
        set device "port1"
    next
    edit 2
        set dst 192.168.112.0 255.255.255.0
        set device "VPN-to-Side1"
    next
    edit 3
        set dst 192.168.112.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Cisco Router

1. 基本上网配置。

```
 interface GigabitEthernet0/0
 nameif Outside
 security-level 0
 ip address 101.100.1.1 255.255.255.0
 no shutdown
 !
 interface GigabitEthernet0/1
 nameif Inside
 security-level 100
 ip address 192.168.112.1 255.255.255.0
```

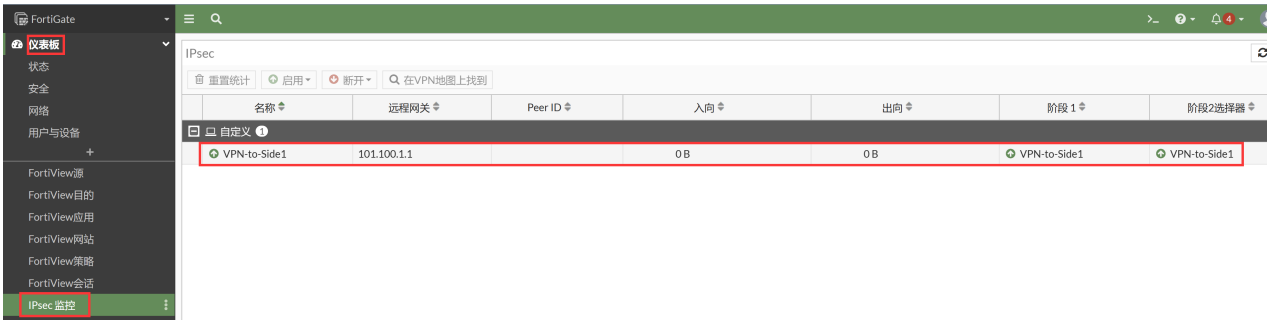
```
 no shutdown
 !
 route Outside 0.0.0.0 0.0.0.0 101.100.1.254

 object network Inside_Net_192.168.112.0
 subnet 192.168.112.0 255.255.255.0

 object network Inside_Net_192.168.112.0
 nat (Inside,Outside) dynamic interface

 policy-map global_policy
 class inspection_default
   inspect icmp
   inspect icmp error
```

2. IPsec IKE v1 的配置。

```
crypto ikev1 enable Outside

 crypto ikev1 policy 10
 authentication pre-share
 encryption aes-256
 hash sha
 group 5
 lifetime 86400

 tunnel-group 202.106.1.1 type ipsec-l2l
 tunnel-group 202.106.1.1 ipsec-attributes
 ikev1 pre-shared-key Fortinet123#
 !
 crypto ipsec ikev1 transform-set MY_Trans esp-aes esp-sha-hmac

 crypto ipsec profile MY_IPsec_Pro
 set ikev1 transform-set MY_Trans
 set security-association lifetime seconds 43200
 set security-association lifetime kilobytes unlimited
 set pfs group5
 !
 interface Tunnel0
 nameif VPN
 ip address 1.1.1.2 255.255.255.0
 tunnel source interface Outside
 tunnel destination 202.106.1.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MY_IPsec_Pro
 !
 management-access Inside    //为了可以让Inside接口可以被ping通
 !
 route VPN 192.168.111.0 255.255.255.0 1.1.1.1
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.106.1.254, port1, [1/0]
C       192.168.111.0/24 is directly connected, port2
S       192.168.112.0/24 [10/0] via VPN-to-Side1 tunnel 101.100.1.1, [1/0]
C       202.106.1.0/24 is directly connected, port1
```

3. FortiGate 侧业务测试。

```
FortiGate # execute ping-options source 192.168.111.1

FortiGate # execute ping 192.168.112.1
PING 192.168.112.1 (192.168.112.1): 56 data bytes
64 bytes from 192.168.112.1: icmp_seq=0 ttl=255 time=1.0 ms
64 bytes from 192.168.112.1: icmp_seq=1 ttl=255 time=1.2 ms
64 bytes from 192.168.112.1: icmp_seq=2 ttl=255 time=0.9 ms
64 bytes from 192.168.112.1: icmp_seq=3 ttl=255 time=1.1 ms
64 bytes from 192.168.112.1: icmp_seq=4 ttl=255 time=1.0 ms

--- 192.168.112.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.0/1.2 ms
```

4. 查看 FortiGate 隧道状态。

```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Side1
version: 1
interface: port1 3
addr: 202.106.1.1:500 → 101.100.1.1:500
tun_id: 101.100.1.1/::101.100.1.1
remote_location: 0.0.0.0
network-id: 0
created: 142s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 71 e412d996a3f9731b/a23fe38afa532c1b
  direction: initiator
  status: established 142-142s ago = 10ms
  proposal: aes256-sha1
  key: 608f0ca220d02d6c-cd0af89676672e4a-c534a6a6c28ef467-15adfbe008693366
  lifetime/rekey: 86400/85957
  DPD sent/recv: 00000000/702e146a

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Side1 ver=1 serial=1 202.106.1.1:0→101.100.1.1:0 tun_id=101.100.1.1 tun_id6=::101.100.1.1 dst_mtu=1500 dpd-
link=on weight=1
bound_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0
proxyid_num=1 child_num=0 refcnt=4 ilast=3 olast=3 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=2
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Side1 proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=28338/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=28497/28800
  dec: spi=24c2aad2 esp=aes key=16 5568087b376e909df121ee5f0c0bcf6c
       ah=sha1 key=20 c423673ee77d191b93bea09089d120ab841bf14c
```

```
  enc: spi=cd21cdff esp=aes key=16 60e173bbd4bff20493566a2e0209abfd
       ah=sha1 key=20 50122d2f90bbb2c02e875084e81d5e1960995b8a
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=101.100.1.1 npu_lgwy=202.106.1.1 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

5. Cisco_Router 侧业务测试。

```
Router#ping 192.168.111.1 source 192.168.112.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.111.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.112.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/4/7 ms

ciscoasa# show crypto ikev1 sa
IKEv1 SAs:
   Active SA: 1
    Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)
Total IKE SA: 1
1   IKE Peer: 202.106.1.1
    Type    : L2L             Role    : responder
    Rekey   : no              State   : MM_ACTIVE
```

6. FortiGate 侧抓包查看。

```
FortiGate # dia sni pa any "host 192.168.112.1 or host 101.100.1.1" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.112.1 or host 101.100.1.1]
2.692454 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
2.692475 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×cd21cdff,seq=0×16)
2.694379 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2aad2,seq=0×15)
2.694394 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
3.692845 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
3.692865 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×cd21cdff,seq=0×17)
3.694116 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2aad2,seq=0×16)
3.694130 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
4.693332 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
4.693356 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×cd21cdff,seq=0×18)
4.695841 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2aad2,seq=0×17)
4.695863 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
5.693612 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
5.693634 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×cd21cdff,seq=0×19)
5.695531 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2aad2,seq=0×18)
5.695548 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
6.694338 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
6.694359 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×cd21cdff,seq=0×1a)
6.696289 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2aad2,seq=0×19)
6.696307 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

- 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

```
FortiGate # config vpn ipsec phase1-interface
FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
FortiGate (BJ-OSPF-TO-SH) # end
```

2. 附 Cisco ASA IPsec VPN 传统的 Crypto MAP 的配置方式（上述为 VTI 方式的 IPsec VPN）。

```
FortiGate 只需要修改IPsec VPN的第二阶段（添加明确的感兴趣流即可），其他的所有配置不变：
config vpn ipsec phase2-interface
    edit "VPN-TO-Side1"
        set phase1name "VPN-TO-Side1"
        set proposal 3des-md5 aes128-sha1
        set auto-negotiate enable
        set src-subnet 192.168.111.0 255.255.255.0
        set dst-subnet 192.168.112.0 255.255.255.0
    next
end

Cisco_ASA配置：
hostname ASA-FW
!
interface GigabitEthernet0/0
nameif Outside
security-level 0
ip address 101.100.1.1 255.255.255.0
!
interface GigabitEthernet0/1
nameif Inside
security-level 100
ip address 192.168.112.1 255.255.255.0
!
route Outside 0.0.0.0 0.0.0.0 101.100.1.254 1

object network Inside_Net_192.168.112.0
subnet 192.168.112.0 255.255.255.0

object network Remote_VPN_Net_192.168.111.0
subnet 192.168.111.0 255.255.255.0

object network Inside_Net_192.168.112.0
nat (Inside,Outside) dynamic interface

policy-map global_policy
class inspection_default
  inspect icmp
  inspect icmp error

crypto ikev1 enable Outside

crypto ikev1 policy 10
 authentication pre-share
 encryption aes-256
 hash sha
 group 5
 lifetime 86400

tunnel-group 202.106.1.1 type ipsec-l2l
tunnel-group 202.106.1.1 ipsec-attributes
 ikev1 pre-shared-key Fortinet123#

crypto ipsec ikev1 transform-set MY_Trans_1 esp-3des esp-md5-hmac
crypto ipsec ikev1 transform-set MY_Trans_2 esp-aes esp-sha-hmac
crypto ipsec ikev1 transform-set MY_Trans_3 esp-aes-192 esp-sha-hmac

access-list VPN-Traffic extended permit ip object Inside_Net_192.168.112.0 object Remote_VPN_Net_192.168.111.0

crypto map MY_IPsec_MAP 10 match address VPN-Traffic
crypto map MY_IPsec_MAP 10 set pfs group5
```

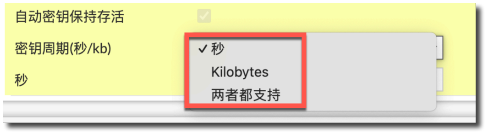
```
crypto map MY_IPsec_MAP 10 set peer 202.106.1.1
crypto map MY_IPsec_MAP 10 set ikev1 transform-set MY_Trans_1 MY_Trans_2 MY_Trans_3
crypto map MY_IPsec_MAP interface Outside

nat (Inside,Outside) source static Inside_Net_192.168.112.0 Inside_Net_192.168.112.0 destination static
Remote_VPN_Net_192.168.111.0 Remote_VPN_Net_192.168.111.0 //使用Twice NAT将VPN流量在NAT中Bypass掉
```

注意事项

1. 默认配置下，FortiGate 只使用时间到期作为二阶段 rekey 的条件，当二阶段配置的时间倒计时结束，则发起二阶段的 rekey。



2. FortiGate 可以使用流量作为 rekey 的条件，也可以同时使用时间（秒）和流量（KB）同时作为 rekey 的条件，当其中一个条件达到配置的数量时，则发起二阶段 rekey。



3. 对于 Cisco ASA 防火墙，默认配置下同时使用时间（秒）和流量（KB）同时作为 rekey 的条件，默认的配置为 4608000 KB 和 3600 秒。当其中一个条件达到配置的数量时，则发起二阶段 rekey。这样可能会导致 IPSec 隧道二阶段频繁的 rekey。

4. 通过如下命令配置 Cisco ASA 只使用时间作为二阶段 rekey 条件。

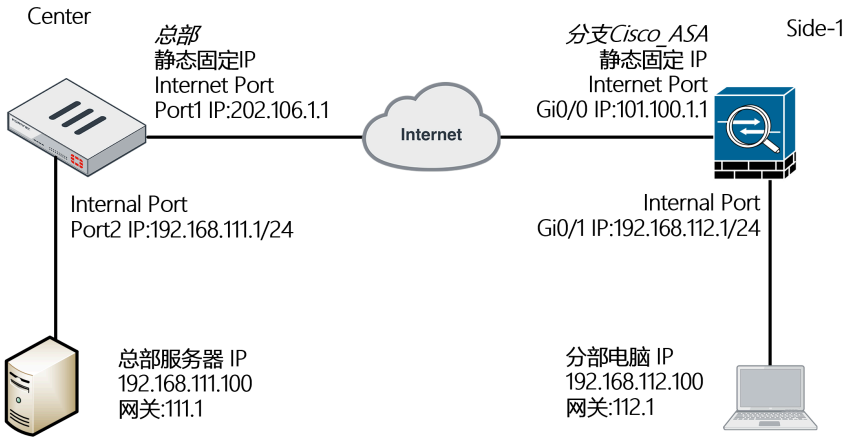
```
crypto ipsec profile MY_IPsec_Pro
set security-association lifetime kilobytes unlimited
```

与思科 ASA 防火墙建立 IPSec VPN (IKE v2)

网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.111.0/24 与 192.168.112.0/24 两个网段的通信。与思科 ASA 防火墙进行 IPsec VPN（IKE v2）对接。

网络拓扑



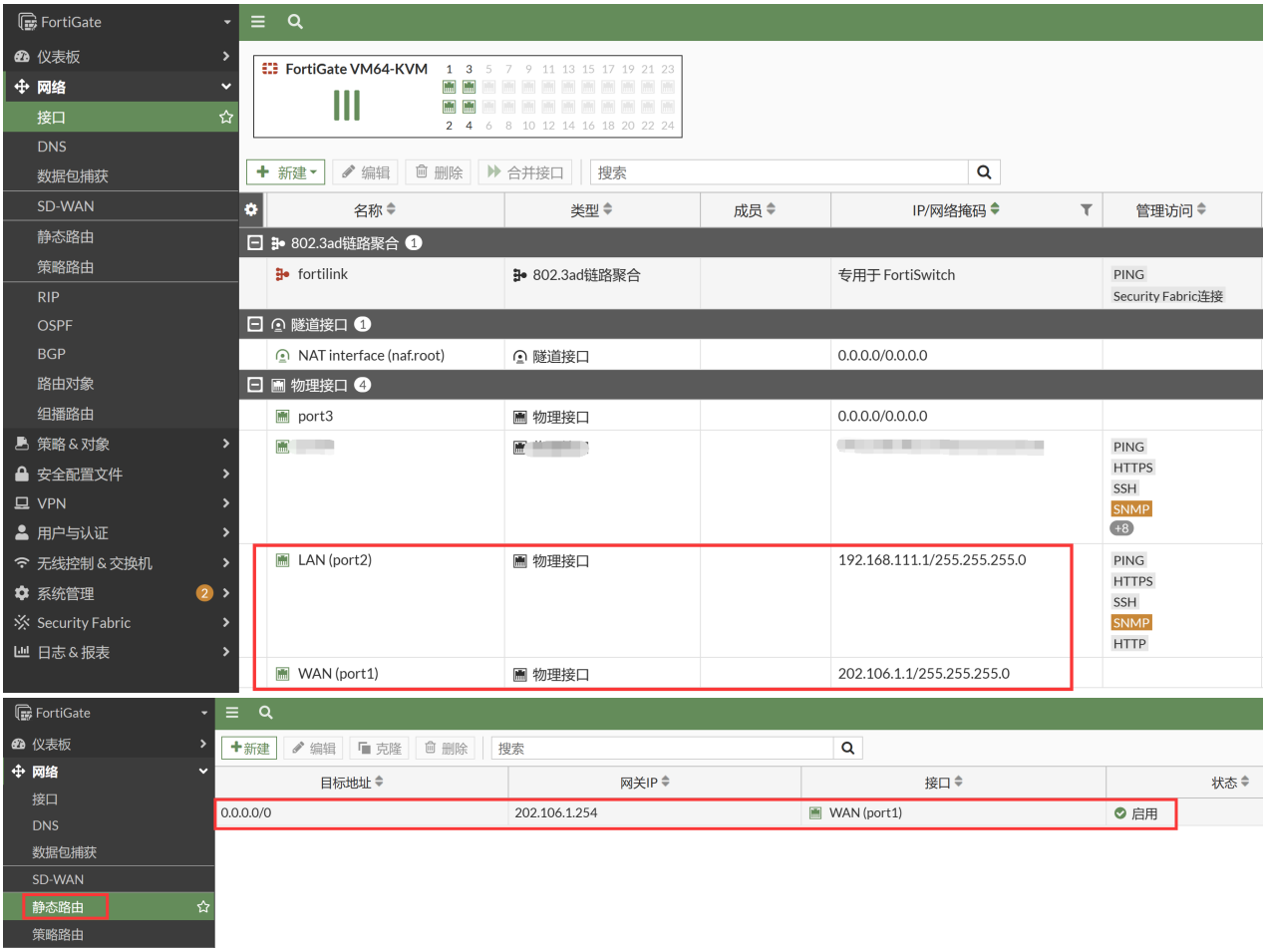
配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Cisco_ASA 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略

配置步骤

FortiGate

1. 基本上网配置。



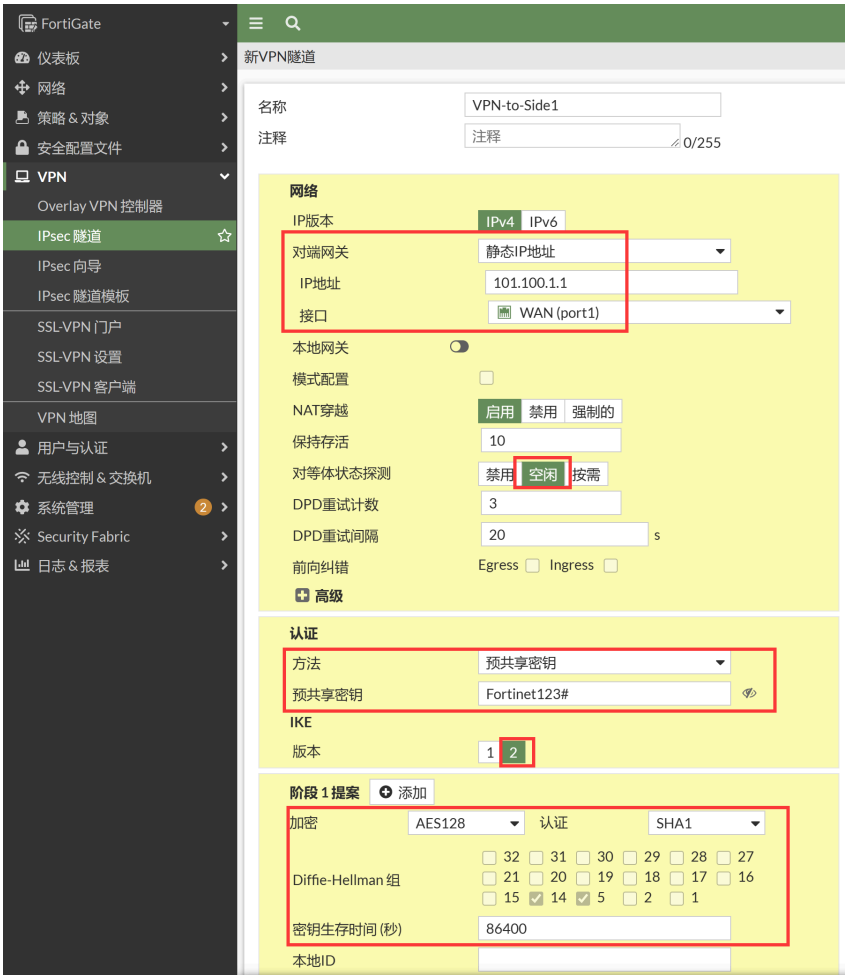
2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。

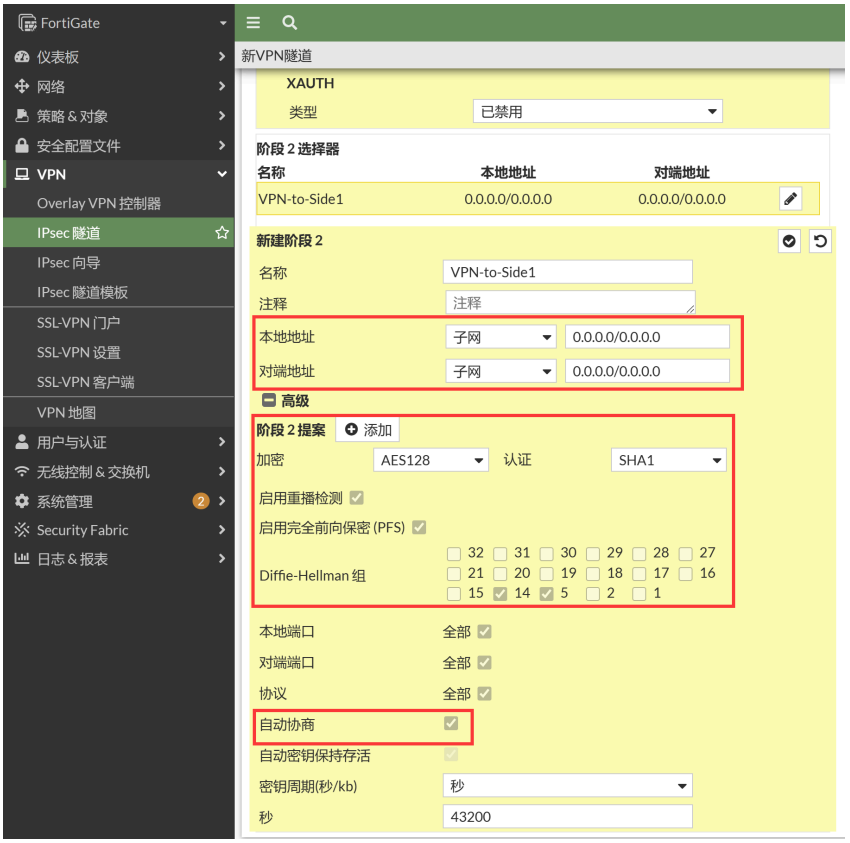


4. 配置网络、认证、第一阶段。



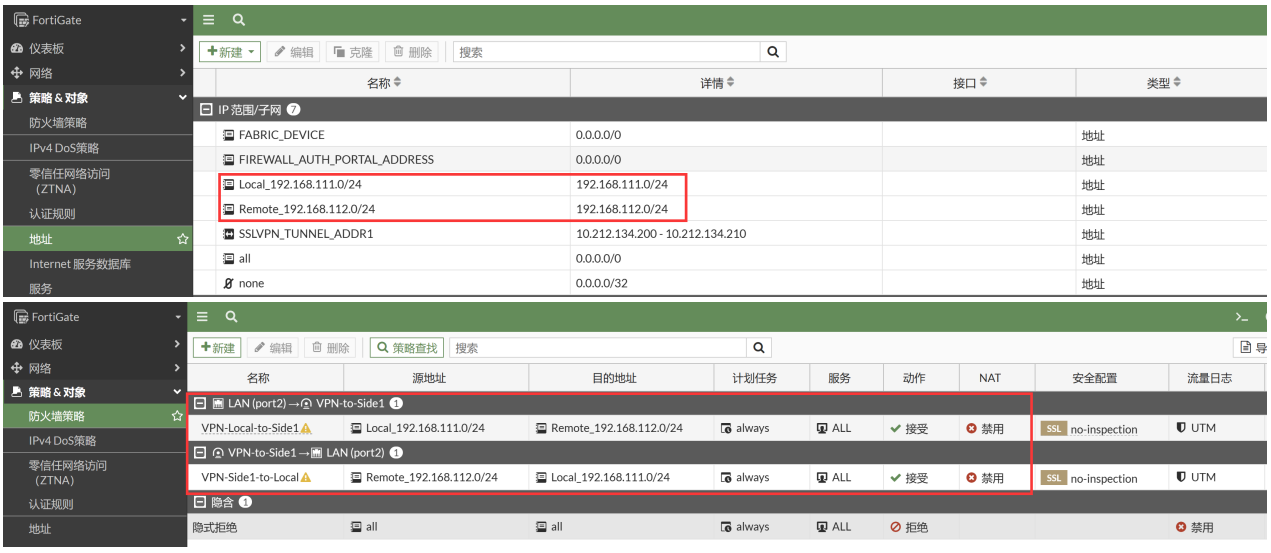

```
config vpn ipsec phase1-interface
    edit "VPN-to-Side1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
        set remote-gw 101.100.1.1
        set psksecret ENC
/HISpjoxUQ1RoRXahwT2OpZmuyYsDY818arB4Q93PztlW0oynlDWuKIk1WIAXFLNldoVFlgOOjBk9Bf9tc6LdI/BeD6CmRvgokNIKa5fVnVgxXLzHM9fPGvt
RXb362jpxZkMxuKHcNGVrJCimmjVGAS+gixHgh8mRLj5kaMAbyhNX1odwNNgxYcxqap5iAwncL3ZzQ==
    next
end
```

5. 配置第二阶段。




```
config vpn ipsec phase2-interface
    edit "VPN-to-Side1"
        set phase1name "VPN-to-Side1"
        set proposal aes128-sha1
        set auto-negotiate enable
    next
end
```

6. 配置 VPN 相关的网段地址对象和防火墙策略。




```
config firewall address
    edit "Local_192.168.111.0/24"
        set subnet 192.168.111.0 255.255.255.0
    next
    edit "Remote_192.168.112.0/24"
        set subnet 192.168.112.0 255.255.255.0
    next
end
```

```
config firewall policy
    edit 1
        set name "VPN-Local-to-Side1"
        set srcintf "port2"
        set dstintf "VPN-to-Side1"
        set action accept
        set srcaddr "Local_192.168.111.0/24"
        set dstaddr "Remote_192.168.112.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "VPN-Side1-to-Local"
        set srcintf "VPN-to-Side1"
        set dstintf "port2"
        set action accept
        set srcaddr "Remote_192.168.112.0/24"
        set dstaddr "Local_192.168.111.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

7. 配置 VPN 业务网段的静态路由。




```
config router static
    edit 1
        set gateway 202.106.1.254
        set device "port1"
    next
    edit 2
        set dst 192.168.112.0 255.255.255.0
        set device "VPN-to-Side1"
    next
    edit 3
        set dst 192.168.112.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Cisco Router

1. 基本上网配置。

```
interface GigabitEthernet0/0
nameif Outside
security-level 0
ip address 101.100.1.1 255.255.255.0
no shutdown
!
interface GigabitEthernet0/1
nameif Inside
security-level 100
ip address 192.168.112.1 255.255.255.0
no shutdown
!
route Outside 0.0.0.0 0.0.0.0 101.100.1.254
```

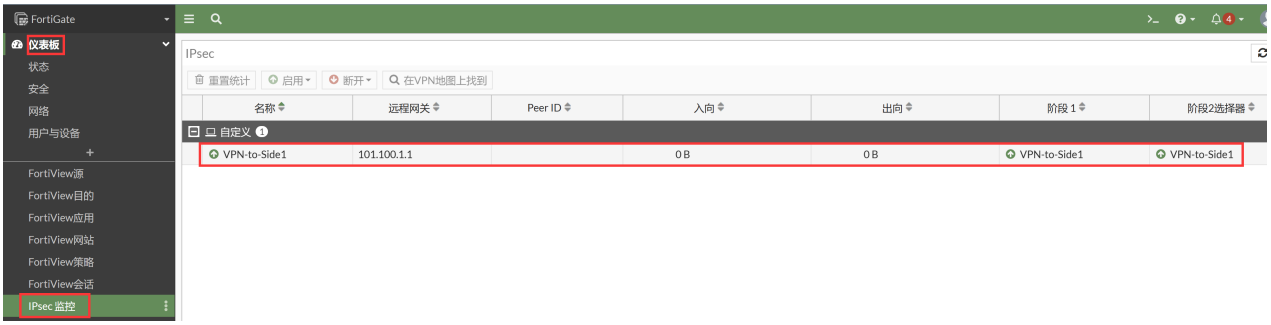
```
object network Inside_Net_192.168.112.0
subnet 192.168.112.0 255.255.255.0

object network Inside_Net_192.168.112.0
nat (Inside,Outside) dynamic interface

policy-map global_policy
class inspection_default
  inspect icmp
  inspect icmp error
```

2. IPsec IKEv2 的配置。

```
crypto ikev2 enable Outside

crypto ikev2 policy 10
 encryption aes-256 aes-192 aes 3des
 integrity sha256 sha md5
 group 14 5
 prf sha256 sha md5
 lifetime seconds 86400
!
tunnel-group 202.106.1.1 type ipsec-l2l
tunnel-group 202.106.1.1 ipsec-attributes
 ikev2 remote-authentication pre-shared-key Fortinet123#
 ikev2 local-authentication pre-shared-key Fortinet123#
!
crypto ipsec ikev2 ipsec-proposal MY_IPsec_Proposal
 protocol esp encryption aes-256 aes-192 aes 3des
 protocol esp integrity sha-256 sha-1 md5

crypto ipsec profile MY_IPsec_Profile
 set ikev2 ipsec-proposal MY_IPsec_Proposal
 set security-association lifetime seconds 43200
 set security-association lifetime kilobytes unlimited
 set pfs group5

interface Tunnel0
nameif VPN
ip address 1.1.1.2 255.255.255.0
tunnel source interface Outside
tunnel destination 202.106.1.1
tunnel mode ipsec ipv4
tunnel protection ipsec profile MY_IPsec_Profile
!
management-access Inside     //为了可以让Inside接口可以被ping通。
!
route VPN 192.168.111.0 255.255.255.0 1.1.1.1
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.106.1.254, port1, [1/0]
C       192.168.111.0/24 is directly connected, port2
S       192.168.112.0/24 [10/0] via VPN-to-Side1 tunnel 101.100.1.1, [1/0]
C       202.106.1.0/24 is directly connected, port1
```

3. FortiGate 侧业务测试。

```
FortiGate # execute ping-options source 192.168.111.1

FortiGate # execute ping 192.168.112.1
PING 192.168.112.1 (192.168.112.1): 56 data bytes
64 bytes from 192.168.112.1: icmp_seq=0 ttl=255 time=1.0 ms
64 bytes from 192.168.112.1: icmp_seq=1 ttl=255 time=1.2 ms
64 bytes from 192.168.112.1: icmp_seq=2 ttl=255 time=0.9 ms
64 bytes from 192.168.112.1: icmp_seq=3 ttl=255 time=1.1 ms
64 bytes from 192.168.112.1: icmp_seq=4 ttl=255 time=1.0 ms

--- 192.168.112.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.0/1.2 ms
```

4. 查看 FortiGate 隧道状态。

```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Side1
version: 2
interface: port1 3
addr: 202.106.1.1:500 → 101.100.1.1:500
tun_id: 101.100.1.1/::101.100.1.1
remote_location: 0.0.0.0
network-id: 0
created: 36s ago
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 134 11aa3d39fcd93c44/852e6b5523a3345d
  direction: initiator
  status: established 36-36s ago = 0ms
  proposal: aes256-sha1
  child: no
  SK_ei: 88ce24e4a2611994-bffb25c2d0ba869b-57e6a55b89508df1-b570966836823c48
  SK_er: 0dadc56cac99ad39-b507dff27d037bb1-33c8ef3555410c03-7e8383322010cb35
  SK_ai: 7481bc53c4d20b62-c9770a0695f2a2a1-0209ff70
  SK_ar: 150cadb3c206c29d-49f33c62d553fb0c-f87d3bf1
  PPK: no
  message-id sent/recv: 2/3
  lifetime/rekey: 86400/86063
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Side1 ver=2 serial=1 202.106.1.1:0→101.100.1.1:0 tun_id=101.100.1.1 tun_id6=::101.100.1.1 dst_mtu=1500 dpd-
link=on weight=1
bound_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0
proxyid_num=1 child_num=0 refcnt=4 ilast=12 olast=12 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=37
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Side1 proto=0 sa=1 ref=2 serial=1 auto-negotiate
```

```
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=42847/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=24c2ab68 esp=aes key=16 3de80d9ea806f7d02167116ef729d135
       ah=sha1 key=20 f05476e4cee8fab25811105c92ea6e5ccf8fd1e4
  enc: spi=0dd3fc4c esp=aes key=16 def29e57dd9b54801a3f31b9a6a920cb
       ah=sha1 key=20 adbfb39e5fe95fb385215da6a73113902bed6c4f
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=101.100.1.1 npu_lgwy=202.106.1.1 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

5. Cisco_Router 侧业务测试。

```
ciscoasa#ping 192.168.111.1 source 192.168.112.1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.111.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.112.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/4/7 ms

ciscoasa(config)# show crypto ikev2 sa
IKEv2 SAs:
Session-id:2, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id Local                                         Remote
Status         Role
 38557965 101.100.1.1/500                               202.106.1.1/500
READY    RESPONDER
      Encr: AES-CBC, keysize: 256, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/73 sec
Child sa: local selector  0.0.0.0/0 - 255.255.255.255/65535
          remote selector 0.0.0.0/0 - 255.255.255.255/65535
          ESP spi in/out: 0×dd3fc4c/0×24c2ab68
```

6. FortiGate 侧抓包查看。

```
FortiGate # dia sni pa any "host 192.168.112.1 or host 101.100.1.1" 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 192.168.112.1 or host 101.100.1.1]
9.068586 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
9.068748 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×0dd3fc4c,seq=0×2)
9.072619 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2ab68,seq=0×1)
9.072669 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
10.069075 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
10.069097 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×0dd3fc4c,seq=0×3)
10.071135 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2ab68,seq=0×2)
10.071155 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
11.069398 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
11.069418 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×0dd3fc4c,seq=0×4)
11.071456 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2ab68,seq=0×3)
11.071483 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
12.069674 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
12.069695 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×0dd3fc4c,seq=0×5)
12.071430 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2ab68,seq=0×4)
12.071450 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
13.069841 VPN-to-Side1 out 192.168.111.1 → 192.168.112.1: icmp: echo request
13.069861 port1 out 202.106.1.1 → 101.100.1.1: ESP(spi=0×0dd3fc4c,seq=0×6)
13.071790 port1 in 101.100.1.1 → 202.106.1.1: ESP(spi=0×24c2ab68,seq=0×5)
13.071809 VPN-to-Side1 in 192.168.112.1 → 192.168.111.1: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

- 抓取 IPsec VPN 的 ESP 加密数据包：

```
diagnose sniffer packet any "host 101.1.1.1 and esp" 4
```

- 抓取 IPsec VPN 的明文业务数据包：

```
diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
```

- 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

```
FortiGate # config vpn ipsec phase1-interface
FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
FortiGate (BJ-OSPF-TO-SH) # end
```

2. Cisco Router IPsec VPN 传统的 Crypto MAP 的配置方式（上述为 VTI 方式的 IPsec VPN）。

```
FortiGate 只需要修改IPsec VPN的第二阶段（添加明确的感兴趣流即可），其他的所有配置不变：
config vpn ipsec phase2-interface
    edit "VPN-TO-Side1"
        set phase1name "VPN-TO-Side1"
        set proposal 3des-md5 aes128-sha1
        set auto-negotiate enable
        set src-subnet 192.168.111.0 255.255.255.0
        set dst-subnet 192.168.112.0 255.255.255.0
    next
end

Cisco_ASA配置：
hostname ASA-FW
!
interface GigabitEthernet0/0
nameif Outside
security-level 0
ip address 101.100.1.1 255.255.255.0
!
interface GigabitEthernet0/1
nameif Inside
security-level 100
ip address 192.168.112.1 255.255.255.0
!
route Outside 0.0.0.0 0.0.0.0 101.100.1.254 1
object network Inside_Net_192.168.112.0
subnet 192.168.112.0 255.255.255.0
object network Inside_Net_192.168.112.0
nat (Inside,Outside) dynamic interface
object network Remote_VPN_Net_192.168.111.0
subnet 192.168.111.0 255.255.255.0
policy-map global_policy
class inspection_default
  inspect icmp
  inspect icmp error
crypto ikev2 enable Outside
crypto ikev2 policy 10
 encryption aes-256 aes-192 aes 3des
 integrity sha256 sha md5
 group 14 5
 prf sha256 sha md5
 lifetime seconds 86400
tunnel-group 202.106.1.1 type ipsec-l2l
tunnel-group 202.106.1.1 ipsec-attributes
 ikev2 remote-authentication pre-shared-key Fortinet123#
 ikev2 local-authentication pre-shared-key Fortinet123#
crypto ipsec ikev2 ipsec-proposal MY_IPsec_Proposal
 protocol esp encryption aes-256 aes-192 aes 3des
 protocol esp integrity sha-256 sha-1 md5
access-list VPN-Traffic extended permit ip object Inside_Net_192.168.112.0 object Remote_VPN_Net_192.168.111.0
crypto map MY_IPsec_MAP 10 match address VPN-Traffic
crypto map MY_IPsec_MAP 10 set pfs group5
```

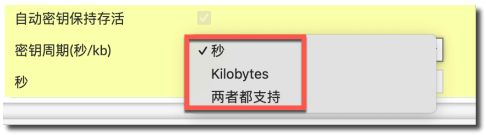
```
crypto map MY_IPsec_MAP 10 set peer 202.106.1.1
crypto map MY_IPsec_MAP 10 set ikev2 ipsec-proposal MY_IPsec_Proposal
crypto map MY_IPsec_MAP interface Outside
nat (Inside,Outside) source static Inside_Net_192.168.112.0 Inside_Net_192.168.112.0 destination static
Remote_VPN_Net_192.168.111.0 Remote_VPN_Net_192.168.111.0 //使用Twice NAT将VPN流量在NAT中Bypass掉
```

注意事项

1. 默认配置下，FortiGate 只使用时间到期作为二阶段 rekey 的条件，当二阶段配置的时间倒计时结束，则发起二阶段的 rekey。



2. FortiGate 可以使用流量作为 rekey 的条件，也可以同时使用时间（秒）和流量（KB）同时作为 rekey 的条件，当其中一个条件达到配置的数量时，则发起二阶段 rekey。



3. 对于 Cisco ASA 防火墙，默认配置下同时使用时间（秒）和流量（KB）同时作为 rekey 的条件，默认的配置为 4608000 KB 和 3600 秒。当其中一个条件达到配置的数量时，则发起二阶段 rekey。这样可能会导致 IPSec 隧道二阶段频繁的 rekey。

4. 通过如下命令配置 Cisco ASA 只使用时间作为二阶段 rekey 条件。

```
crypto ipsec profile MY_IPsec_Pro
set security-association lifetime kilobytes unlimited
```

与华为 USG6000V 防火墙建立 IPSec VPN

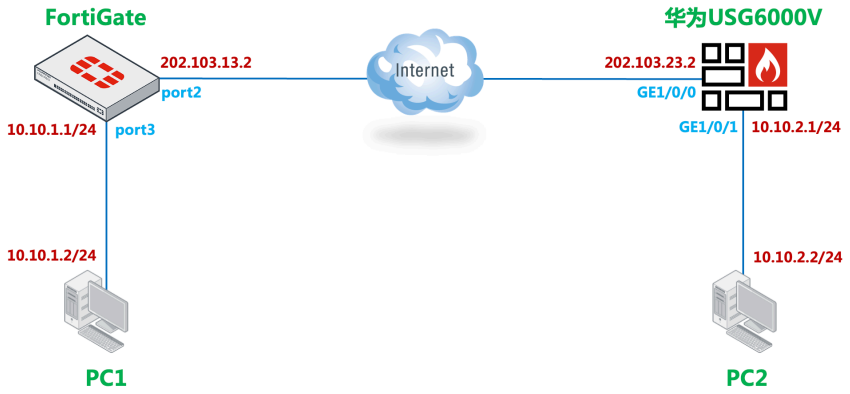
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与华为 USG6000V 防火墙进行 IPsec VPN（IKE v1）对接。

华为 USG6000V 防火墙版本：

```bash
<USG6000V2>display version
2023-04-06 17:43:15.520 +08:00
Huawei Versatile Routing Platform Software
VRP (R) Software, Version 5.170 (USG6000V2 V500R005C00SPC100)
Copyright (C) 2014-2018 Huawei Technologies Co., Ltd.
USG6000V2 uptime is 0 week, 0 day, 2 hours, 40 minutes
IPS Signature Database Version   : 2018070605
IPS Engine Version               : V200R005C00SPC037
AV Signature Database Version    :
SA Signature Database Version    : 2018062700
C&C Domain Name Database Version :
FILE Reputation Database Version :
Location Database Version        : 2014010414
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置华为 USG6000V 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

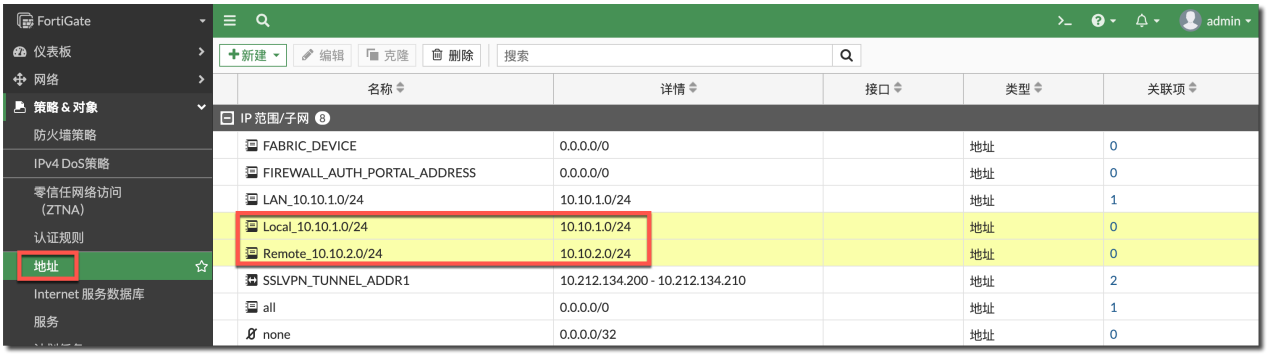
重要

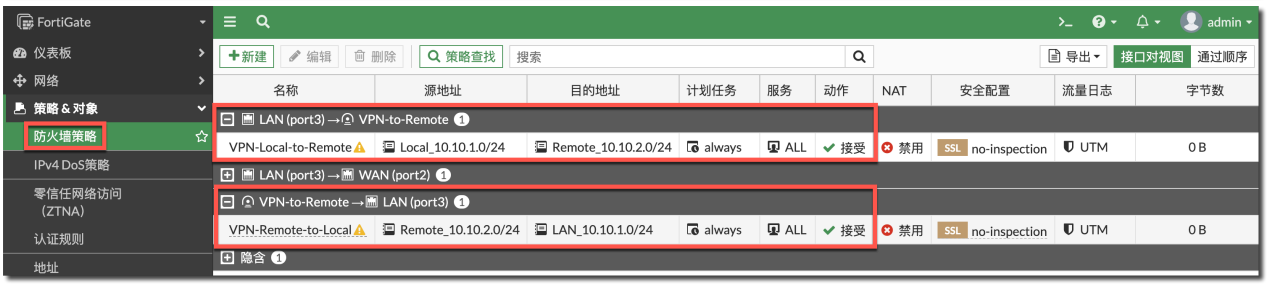
不像和 Cisco/PA/Juniper/山石使用虚拟接口创建 IPSec，感兴趣流是全 0。华为防火墙通常都是使用物理口创建 IPSec，需要写明细的感兴趣流，所以 FortiGate 也要对应的写明细感兴趣数据流，不能写全 0。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
10.10.1.0/24
10.10.2.0/24
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
10.10.1.0/24
对端地址
子网
10.10.2.0/24
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```bash
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 10.10.1.0 255.255.255.0
        set dst-subnet 10.10.2.0 255.255.255.0
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。






```bash
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
        set srcintf "VPN-to-Remote"
```

```
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

重要

相关 VPN 的路由配置中“黑洞路由的意义：

你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?

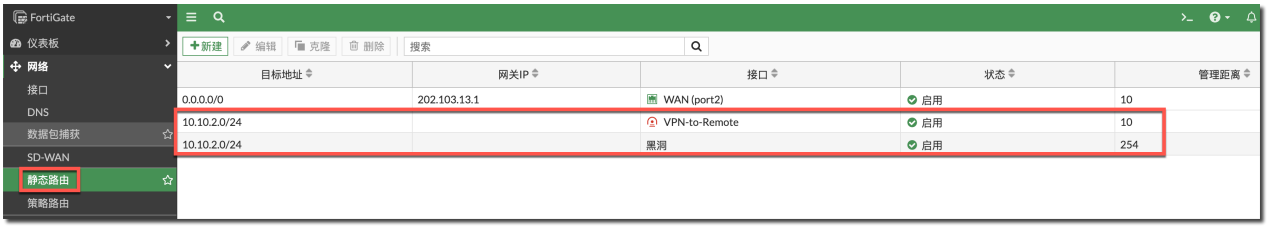
VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。

解决方法：

- 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
- 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
- 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。

```
FortiGate # config system global
FortiGate (global) # set snat-route-change enable
FortiGate (global) # end
```

三种方式任意选择一种即可。推荐使用黑洞路由方式。



bash

```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

华为 USG6000V

1. 基本 IP 与路由配置。






```bash
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 202.103.23.2 255.255.255.0
#
interface GigabitEthernet1/0/1
 undo shutdown
 ip address 10.10.2.1 255.255.255.0
#
firewall zone trust
 set priority 85
 add interface GigabitEthernet0/0/0
 add interface GigabitEthernet1/0/1
#
firewall zone untrust
 set priority 5
 add interface GigabitEthernet1/0/0
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet1/0/0 202.103.23.1
```

2. 地址对象、策略、NAT 配置。








```bash
ip address-set Local_10.10.2.0/24 type object
 address 0 10.10.2.0 mask 24
#
ip address-set Remote_10.10.1.0/24 type object
 address 0 10.10.1.0 mask 24
#
security-policy
 default action permit
 rule name to_Internet
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  action permit
#
nat-policy
 rule name to_Internet
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  action source-nat easy-ip
```

3. 配置 IPSec，新建 IPSec VPN 策略。



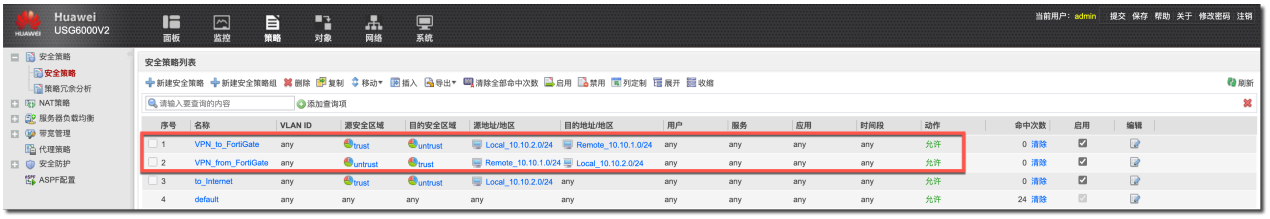
4. 如下图参数配置，注意与 FortiGate 端的安全提议要一致（注意加密数据流不要选择地址对象，需要手动输入，否则第二阶段可能无法建立），开启反向路由注入。




```bash
acl number 3000
 rule 5 permit ip source 10.10.2.0 0.0.0.255 destination 10.10.1.0 0.0.0.255
#
ike proposal 1
 encryption-algorithm aes-256
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
```

```
#
ike peer ike64165343931
 pre-shared-key %^%#GFRw6k/f}&DP2(A}Sh*9",ym@xVr%!NO~3!N]5u!%^%#
 ike-proposal 1
 remote-id-type none
 dpd type periodic
 remote-address 202.103.13.2
#
ipsec policy ipsec6416534390 1 isakmp
 security acl 3000
 pfs dh-group14
 ike-peer ike64165343931
 proposal prop64165343931
 tunnel local applied-interface
 alias to_FortiGate
 sa trigger-mode auto
 sa duration traffic-based 10485760
 sa duration time-based 43200
 route inject dynamic
```

5. 配置 IPSec VPN 相关网段放通的安全策略，并将他们移动至上网策略之前。




```
security-policy
 default action permit
 rule name VPN_to_FortiGate
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  destination-address address-set Remote_10.10.1.0/24
  action permit
 rule name VPN_from_FortiGate
  source-zone untrust
  destination-zone trust
  source-address address-set Remote_10.10.1.0/24
  destination-address address-set Local_10.10.2.0/24
  action permit
 rule name to_Internet
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  action permit
```

6. 华为防火墙的转发流程是先做 SNAT，再做 IPsec VPN，也就是说，如果按照我们之前配置的 SNAT 规则，VPN 的流量 10.10.2.2 去访问 10.10.1.2 的流量会被 SNAT 成公网 IP 202.103.23.2，然后再进入到 IPsec VPN 流程，这个时候是无法匹配到感兴趣流的，因此流量将无法成功被 IPSec 加密。所以需要再配置一个 VPN 流量 bypass 的 SNAT 规则，让 VPN 流量不做 SNAT 转换，然后将其放置到最前面，优先匹配即可。

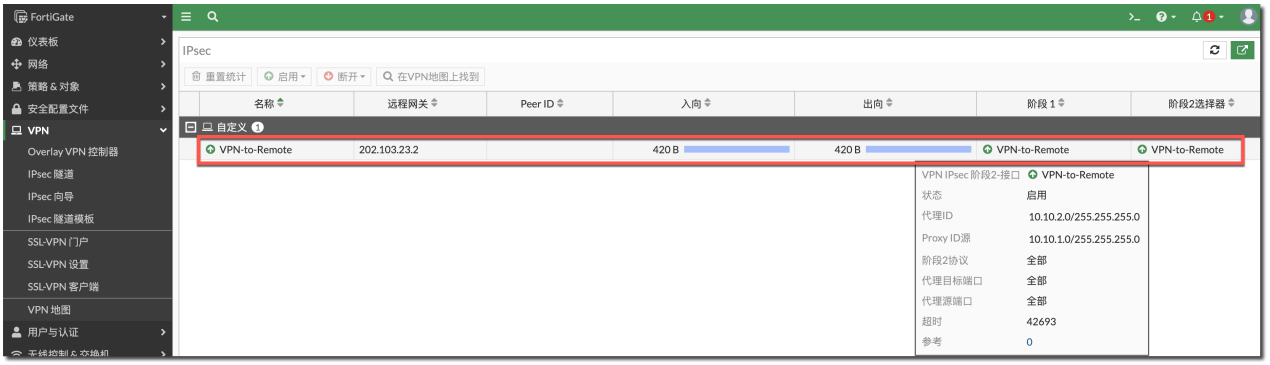



```
nat-policy
 rule name VPN_to_FortiGate
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  destination-address address-set Remote_10.10.1.0/24
  action no-nat
```

```
 rule name to_Internet
  source-zone trust
  destination-zone untrust
  source-address address-set Local_10.10.2.0/24
  action source-nat easy-ip
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。




```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 490s ago
IKE SA: created 1/1  established 1/1  time 210/210/210 ms
IPsec SA: created 1/2  established 1/1  time 390/390/390 ms
  id/spi: 257 05b470e04dc4363a/7f25ce9c15e94de3
  direction: initiator
  status: established 490-490s ago = 210ms
  proposal: aes256-sha256
  key: 9b5842bda3877c51-ebbc39ba4a960fde-f76d2defdebe67bd-de35bd7bc5720ec2
  lifetime/rekey: 86400/85609
  DPD sent/recv: 0000004d/135cb4c9

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=1 202.103.13.2:0->202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=10
proxyid_num=1 child_num=0 refcnt=4 ilast=15 olast=15 ad=/0
stat: rxp=12 txp=12 rxb=1008 txb=1008
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=77
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Remote proto=0 sa=1 ref=4 serial=1 auto-negotiate
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.2.0-10.10.2.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=42436/0B replaywin=2048
       seqno=d esn=0 replaywin_lastseq=0000000d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42929/43200
  dec: spi=34dc8977 esp=aes key=32 947441e7c4d219ad7c0cb909942e17d2c8e25b42346ea53f301ba37161d2fdef
       ah=sha256 key=32 df58345781028ba4ac18cc037ce667e974941e4ff544b7368f523307c83d4ea4
  enc: spi=0b11f21b esp=aes key=32 d5dcfee82b332abf9bb62c620c3e42ee42c0f6c990fa380204b02b83f0acdbff
       ah=sha256 key=32 16b2afbcec0b0feb10cf4dedba31d99e3c59ce020823a88b9b99122b02bedf65
  dec:pkts/bytes=24/2016, enc:pkts/bytes=24/2880
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

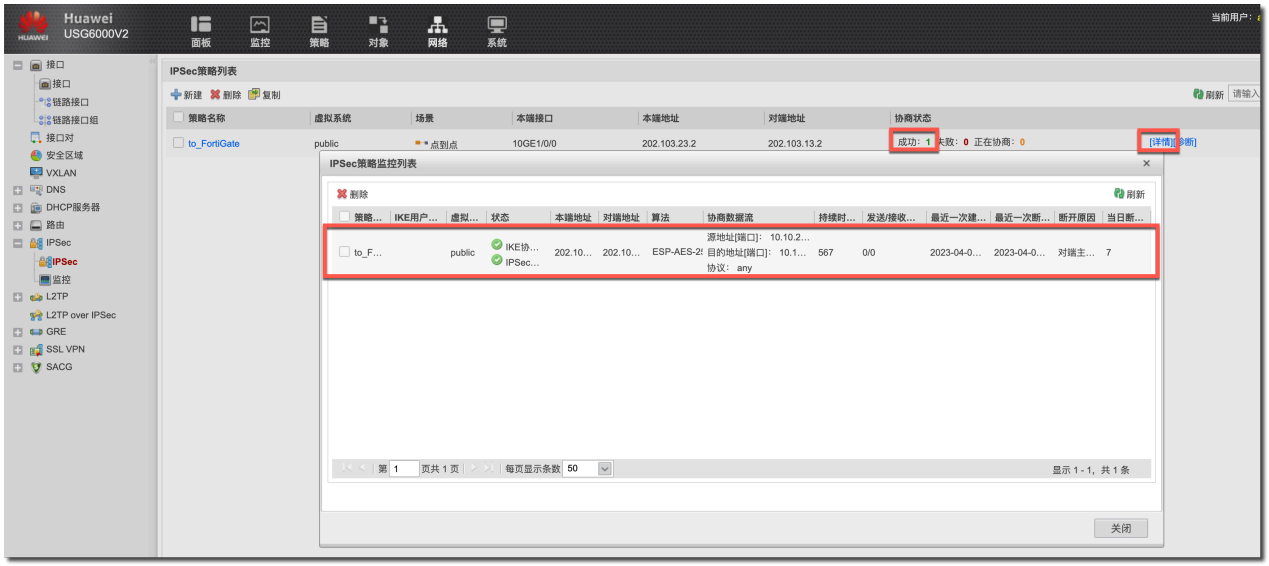
2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

3. FortiGate 侧 PC1 业务测试。

```
VPCS> show
NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6804/64

VPCS> ping 10.10.2.2
84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=5.413 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=2.511 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=2.242 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=1.902 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=2.080 ms
```

4. 华为防火墙查看 IPSec 连接状态。




```
<USG6000V2>display ike sa verbose remote 202.103.13.2
2023-04-06 17:27:29.590 +08:00
Ike sa verbose information :
-------------------------------------------------------------------------
Ike Sa phase   : 1
Establish Time : 2023-04-06 17:16:56
PortCfg Index  : 0×6
IKE Peer Name  : ike64165343931
Connection Id  : 347
Version        : v1
Exchange Mode  : Main
Flow VPN       :
Peer VPN       :
-------------------------------------------------------------------------
Intiator Cookie                        : 0×05b470e04dc4363a
Responder Cookie                       : 0×7f25ce9c15e94de3
```

```
Local Address                           : 202.103.23.2
Remote Address                          : 202.103.13.2:500
Encryption Algorithm                    : AES-256
Authentication Algorithm                : SHA2-256
Authentication Method                   : Pre-Shared key
DPD Capability                          : Yes
DPD Enable                              : Yes
Remaining Duration                      : 85766
Reference Counter                       : 1
Flags                                   : RD|A
Remote Id Type                          : IP
Remote Id                               : 202.103.13.2
DH Group                                : 14
NAT Traversal Version                   : RFC3947
ModeCfg IP                              : -
-------------------------------------------------------------------------

  Number of IKE SA : 1
```

```
<USG6000V2> display ipsec sa remote 202.103.13.2
2023-04-06 17:28:30.260 +08:00
ipsec sa information:
=============================
Interface: GigabitEthernet1/0/0
=============================
  -----------------------------
  IPSec policy name: "ipsec6416534390"
  Sequence number  : 1
  Acl group        : 3000
  Acl rule         : 5
  Mode             : ISAKMP
  -----------------------------
    Connection ID     : 350
    Encapsulation mode: Tunnel
    Holding time      : 0d 0h 11m 32s
    Tunnel local      : 202.103.23.2:500
    Tunnel remote     : 202.103.13.2:500
    Flow source       : 10.10.2.0/255.255.255.0 0/0-65535
    Flow destination  : 10.10.1.0/255.255.255.0 0/0-65535
    [Outbound ESP SAs]
      SPI: 886868343 (0×34dc8977)
      Proposal: ESP-ENCRYPT-AES-256 ESP-AUTH-SHA2-256-128
      SA remaining key duration (kilobytes/sec): 10485759/42508
      Max sent sequence-number: 18
      UDP encapsulation used for NAT traversal: N
      SA encrypted packets (number/bytes): 17/1428
    [Inbound ESP SAs]
      SPI: 185725467 (0×b11f21b)
      Proposal: ESP-ENCRYPT-AES-256 ESP-AUTH-SHA2-256-128
      SA remaining key duration (kilobytes/sec): 10485759/42508
      Max received sequence-number: 1
      UDP encapsulation used for NAT traversal: N
      SA decrypted packets (number/bytes): 17/1428
      Anti-replay : Enable
      Anti-replay window size: 1024
```

5. 查看华为防火墙的路由表，可以看到自动添加了到 FortiGate 内网的路由。

```
[USG6000V2]display ip routing-table
2023-04-06 17:32:20.280 +08:00
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 8        Routes : 8
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0   Static  60   0           D   202.103.23.1    GigabitEthernet1/0/0
      10.10.1.0/24  Unr     70   0           D   202.103.13.2    GigabitEthernet1/0/0
      10.10.2.0/24  Direct  0    0           D   10.10.2.1       GigabitEthernet1/0/1
```

```
      10.10.2.1/32  Direct  0    0           D   127.0.0.1       GigabitEthernet1/0/1
      127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
   202.103.23.0/24  Direct  0    0           D   202.103.23.2    GigabitEthernet1/0/0
   202.103.23.2/32  Direct  0    0           D   127.0.0.1       GigabitEthernet1/0/0
```

6. 华为防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=3.139 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=2.126 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=2.508 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.257 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=1.767 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-06 17:30:20.440445 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:20.441055 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:20.441296 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:20.441308 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:21.444661 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:21.444753 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:21.444962 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:21.444969 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:22.447197 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:22.447332 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:22.447520 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:22.447527 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:23.451289 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:23.451454 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:23.451667 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:23.451676 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:24.453906 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:24.453939 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-06 17:30:24.454165 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-06 17:30:24.454176 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

关于 sniffer 抓 VPN 业务和 ESP 的包：

- 抓取 IPsec VPN 的 IKE 协商包：

```
diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
```

- 抓取 IPsec VPN 的 ESP 加密数据包：

```
diagnose sniffer packet any "host 101.1.1.1 and esp" 4
```

- 抓取 IPsec VPN 的明文业务数据包：

```
diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
```

- 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

```
FortiGate # config vpn ipsec phase1-interface
FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
FortiGate (BJ-OSPF-TO-SH) # end
```

与 H3C SecPath 防火墙建立 IPSec VPN

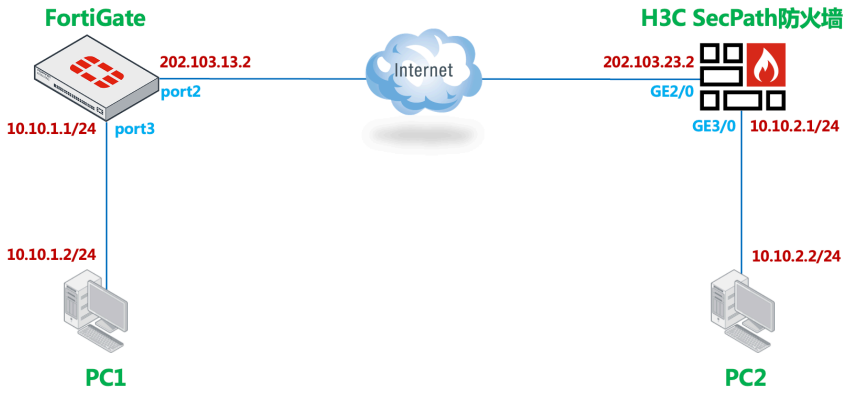
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与 H3C SecPath 防火墙进行 IPsec VPN（IKE v1）对接。

H3C SecPath 防火墙版本：

```
<H3C>display version
H3C Comware Software, Version 7.1.064, ESS 1185P29
Copyright (c) 2004-2022 New H3C Technologies Co., Ltd. All rights reserved.
H3C SecPath vFW1000 uptime is 0 weeks, 0 days, 0 hours, 7 minutes
Last reboot reason : Power on
Boot image: flash:/vFW1000-CMW710-BOOT-E1185P29-X64.bin
Boot image version: 7.1.064, ESS 1185P29
  Compiled Oct 31 2022 14:00:00
System image: flash:/vFW1000-CMW710-SYSTEM-E1185P29-X64.bin
System image version: 7.1.064, ESS 1185P29
  Compiled Oct 31 2022 14:00:00
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 H3C SecPath 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

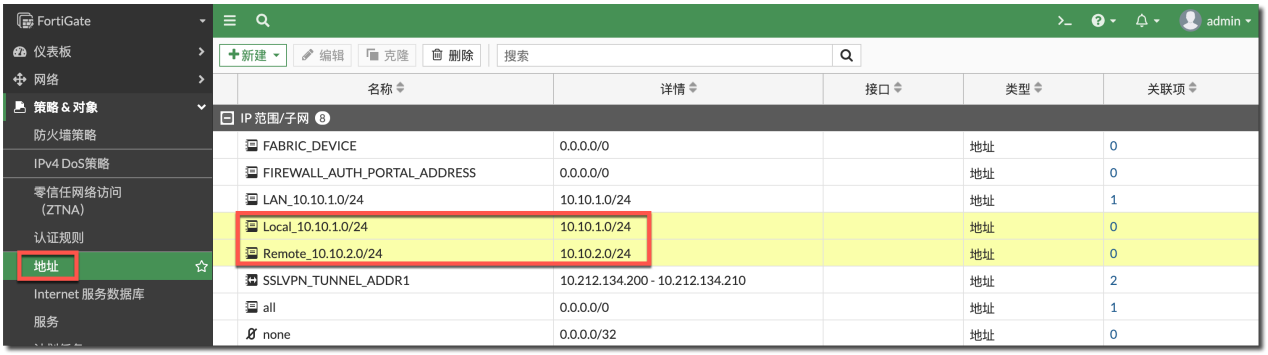
重要

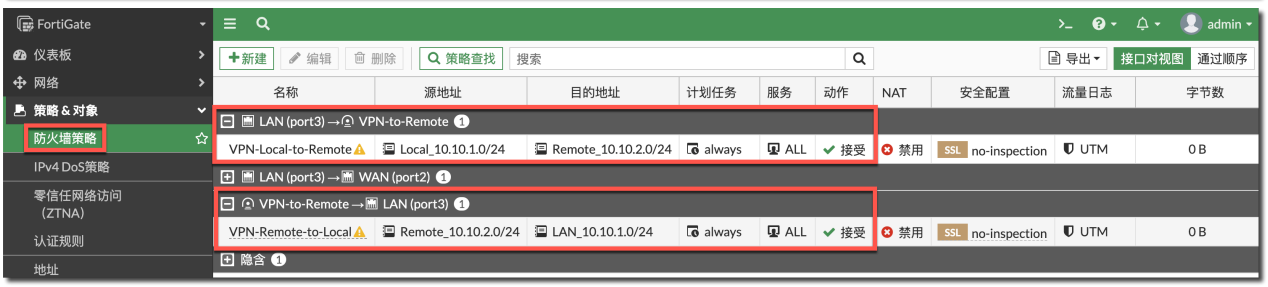
不像和 Cisco/PA/Juniper/山石使用虚拟接口创建 IPSec，感兴趣流是全 0。H3C 防火墙通常都是使用物理口创建 IPSec，需要写明细的感兴趣流，所以 FortiGate 也要对应的写明细感兴趣数据流，不能写全 0。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
10.10.1.0/24
10.10.2.0/24
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
10.10.1.0/24
对端地址
子网
10.10.2.0/24
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 10.10.1.0 255.255.255.0
        set dst-subnet 10.10.2.0 255.255.255.0
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。






```
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
        set srcintf "VPN-to-Remote"
```

```
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

重要

相关 VPN 的路由配置中“黑洞路由的意义：

你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?

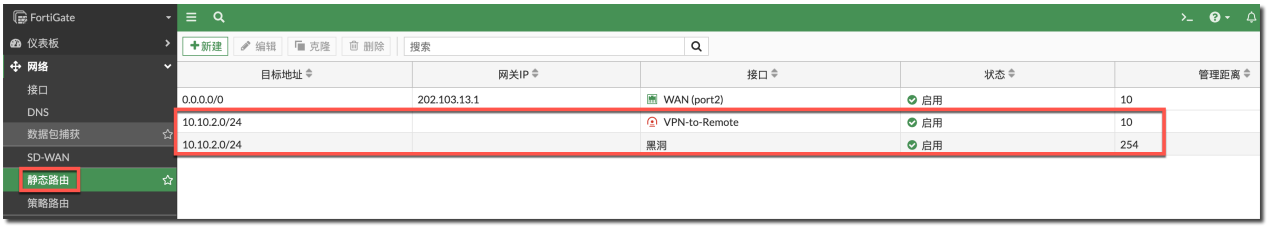
VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。

解决方法：

- 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
- 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
- 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。

```
FortiGate # config system global
FortiGate (global) # set snat-route-change enable
FortiGate (global) # end
```

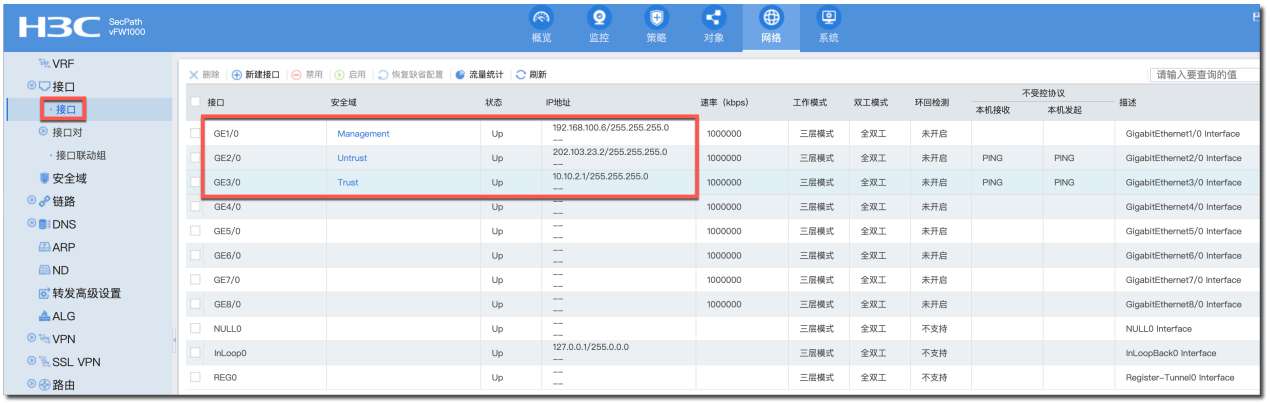
三种方式任意选择一种即可。推荐使用黑洞路由方式。

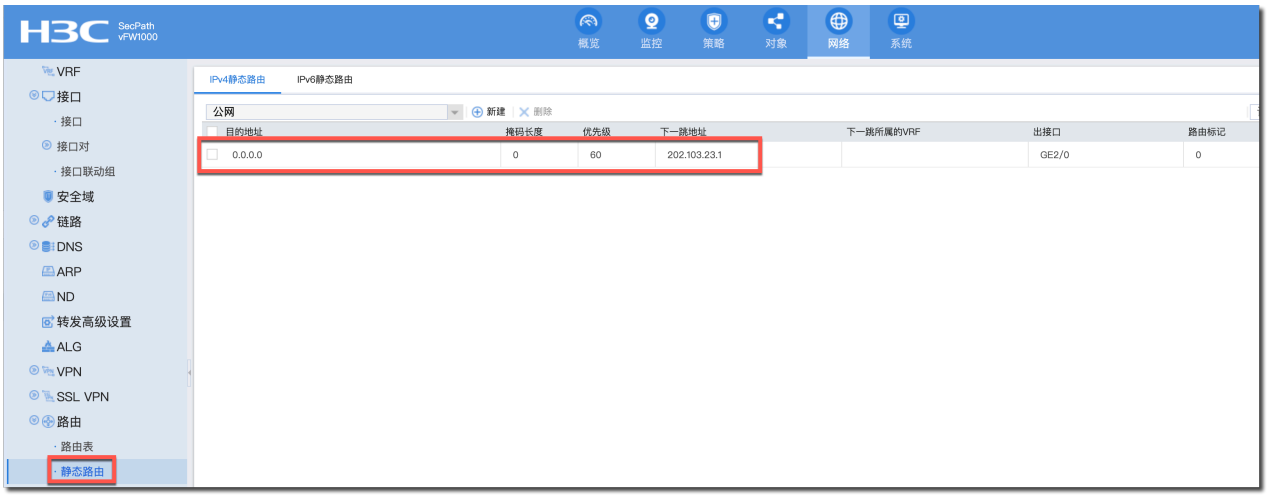



```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

H3C SecPath

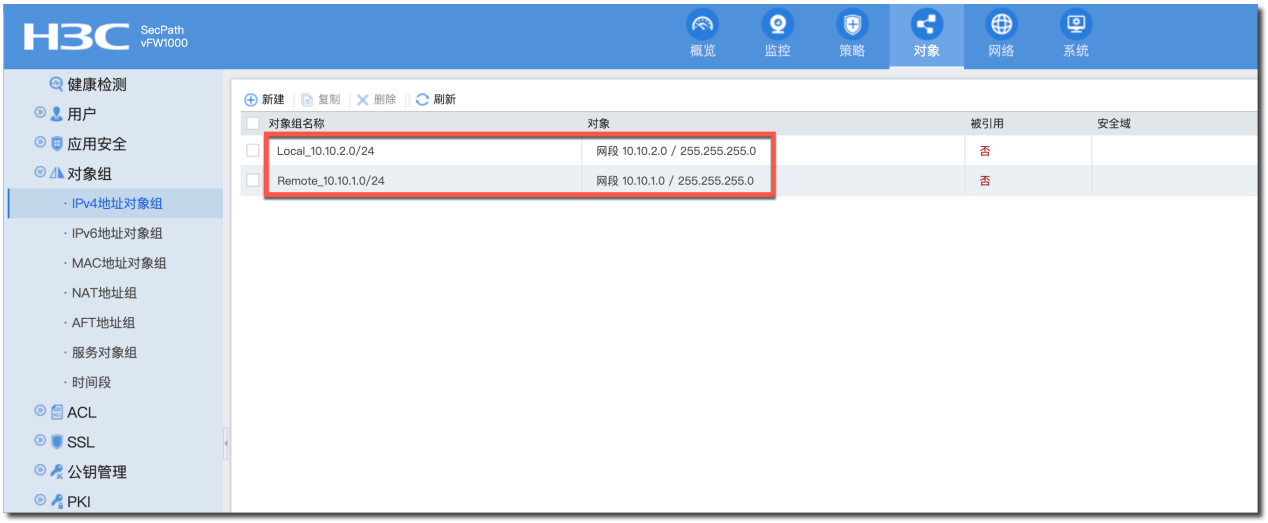
1. 基本 IP 与路由配置。



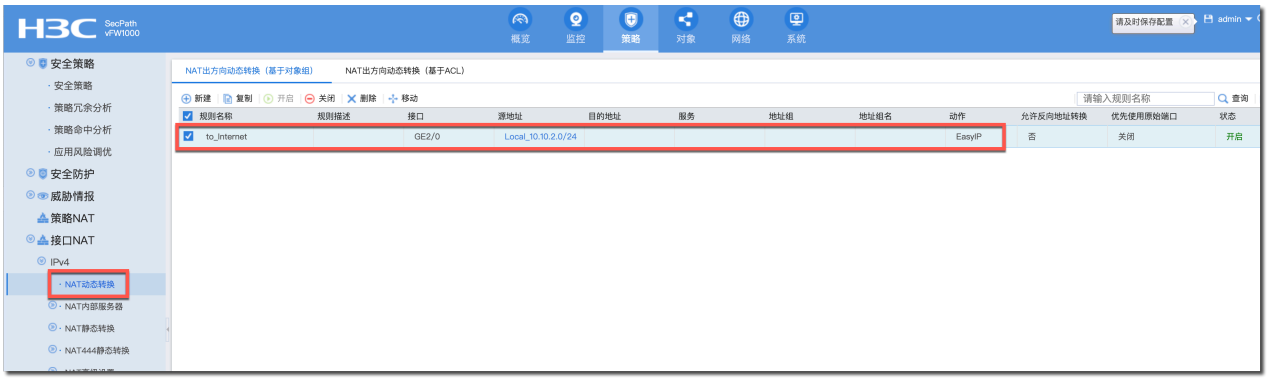



```
interface GigabitEthernet2/0
 port link-mode route
 ip address 202.103.23.2 255.255.255.0
 manage ping inbound
 manage ping outbound
#
interface GigabitEthernet3/0
 port link-mode route
 ip address 10.10.2.1 255.255.255.0
 manage ping inbound
 manage ping outbound
#
security-zone name Trust
 import interface GigabitEthernet3/0
#
security-zone name Untrust
 import interface GigabitEthernet2/0
#
 ip route-static 0.0.0.0 0 GigabitEthernet2/0 202.103.23.1
```

2. 地址对象、策略、NAT 配置。

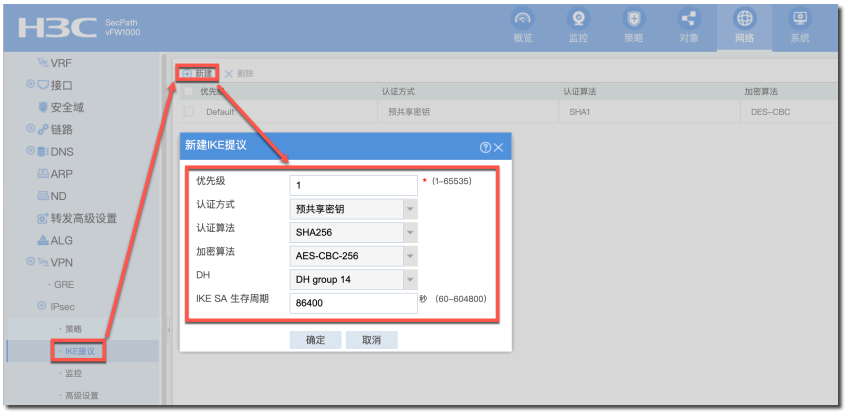







```
object-group ip address Local_10.10.2.0/24
 0 network subnet 10.10.2.0 255.255.255.0
#
object-group ip address Remote_10.10.1.0/24
 0 network subnet 10.10.1.0 255.255.255.0
#
security-policy ip
 rule 2 name to_Internet
  action pass
  source-zone Trust
  destination-zone Untrust
  source-ip Local_10.10.2.0/24
#
nat policy
 rule name to_Internet
  source-ip Local_10.10.2.0/24
  outbound-interface GigabitEthernet2/0
  action easy-ip
```

3. 配置 IPSec，新建 IKE 提议，按照如图安全提议配置，点击确定下发配置。



4. 新建 IPSec 策略，如下图参数配置，注意与 FortiGate 端的安全提议要一致，注意对端 ID 选择“IPv4 地址”，内容填写为 FortiGate 的公网口 IP 地址，如果 FortiGate 的地址是动态的，可以选择 FQDN 形式的 ID，FortiGate 侧本地 ID 也要对应修改为 FQDN 形式，开启“自动触发”和“DPD”，勾选“自动生成安全策略”（这里自动创建的策略是允许 IPSec 协商的发起和接受）。


H3C SecPath vFW1000
概览
监控
策略
对象
网络
系统
VRF
接口
安全域
链路
DNS
ARP
ND
转发高级设置
ALG
VPN
GRE
IPsec
策略
IKE提议
监控
高级设置
ADVPN
L2TP
SSL VPN
路由
组播
DHCP
服务
安全接入
基本配置
策略名称 to_FortiGate (1-46字符)
优先级 1 (1-65535)
设备角色 对等/分支节点 中心节点
IP地址类型 IPv4 IPv6
智能选路 开启
接口 GE2/0 [配置]
本端地址 202.103.23.2
不进行Nat转换 开启
对端IP地址/主机名 202.103.13.2 (1-253字符)
描述 (1-80字符)
IKE策略
协商模式 主模式 野蛮模式 国密主模式
认证方式 预共享密钥 数字认证
预共享密钥 ••••••••• (1-128字符)
IKE提议 优先级（认证方式；认证算法；加密算法；DH）
本端ID IPv4 地址 1.2.3.4
对端ID IPv4 地址 202.103.13.2
对端ID填写FortiGate的公网口IP，如果FortiGate的公网IP为动态的，可以使用FQDN方式，FortiGate侧的本地ID也需要对应修改为FQDN
保护的数据流
源IP地址 目的IP地址 VRF 协议 源端口 目的端口 动作 编辑
10.10.2.0/255.25... 10.10.1.0/255.25... 公网 any any any 保护
共 1 条
需要配置受保护数据流目的地址的路由信息，如果没有请进行配置。
需要为本端访问对端业务的流量，配置放通动作的安全策略，如果没有请进行配置。
需要为对端访问本端业务的流量，配置放通动作的安全策略，如果没有请进行配置。
触发模式 流量触发 自动触发
高级配置
IPsec参数
封装模式 隧道模式 传输模式
安全协议 ESP AH AH-ESP
ESP认证算法 SHA256
ESP加密算法 AES-CBC-256
PFS Group_14
IPsec SA生存时间
基于时间 43200 秒 (180-604800)
基于流量 千字节 (2560-4294967295)
IPsec SA 空闲超时时间 60 秒 (60-86400)
DPD检测 开启
检测方式 按需检测 按时检测
检测时间间隔 20 秒 (1-300)
重传时间间隔 5 秒 (1-60)
内网VRF 公网
QoS预分类 开启
安全策略
自动生成安全策略
配置对端访问的安全策略
名称 IPsec_to_FortiGate_1_20230407151324_IN
源安全域 Any [多选]
目的安全域 Local [多选]
源地址对象组 Any [多选]
目的地址对象组 Any [多选]
服务 ike, nat-t-ipsec, ipsec-ah, ipsec-esp [多选]
配置本端访问的安全策略
名称 IPsec_to_FortiGate_1_20230407151324_OUT
源安全域 Local [多选]
目的安全域 Any [多选]
源地址对象组 Any [多选]
目的地址对象组 Any [多选]
服务 ike, nat-t-ipsec, ipsec-ah, ipsec-esp [多选]
确定
取消

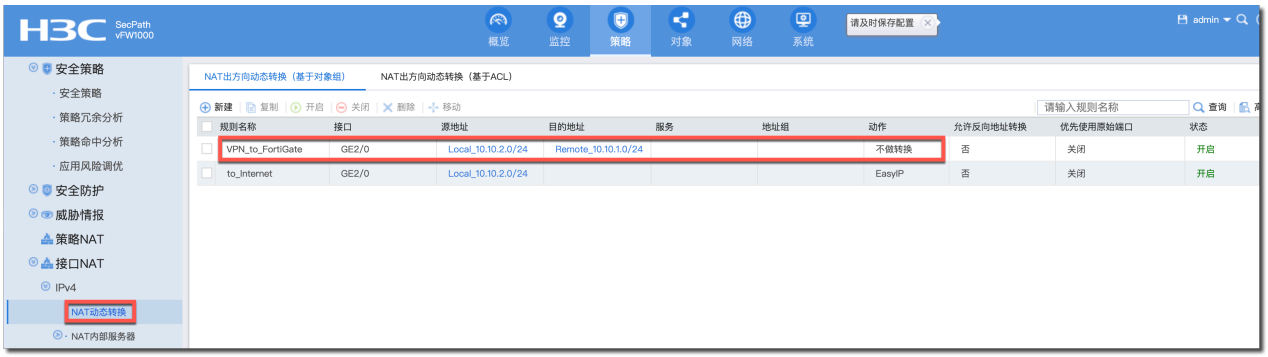
```
acl advanced name IPsec_to_FortiGate_IPv4_1
 rule 0 permit ip source 10.10.2.0 0.0.0.255 destination 10.10.1.0 0.0.0.255
#
ipsec transform-set to_FortiGate_IPv4_1
 esp encryption-algorithm aes-cbc-256
 esp authentication-algorithm sha256
 pfs dh-group14
#
ipsec policy to_FortiGate 1 isakmp
 transform-set to_FortiGate_IPv4_1
 security acl name IPsec_to_FortiGate_IPv4_1
 local-address 202.103.23.2
 remote-address 202.103.13.2
 ike-profile to_FortiGate_IPv4_1
 sa trigger-mode auto
 sa duration time-based 43200
 sa idle-time 60
#
ike profile to_FortiGate_IPv4_1
 keychain to_FortiGate_IPv4_1
 dpd interval 20 on-demand
 match remote identity address 202.103.13.2 255.255.255.255
 match local address GigabitEthernet2/0
#
ike proposal 1
 encryption-algorithm aes-cbc-256
 dh group14
 authentication-algorithm sha256
#
ike keychain to_FortiGate_IPv4_1
 match local address GigabitEthernet2/0
 pre-shared-key address 202.103.13.2 255.255.255.255 key cipher $c$3$OAC3tjUo2G+AOc7w0iW8jeQvV2wpRj96iHSZ
#
interface GigabitEthernet2/0
 ipsec apply policy to_FortiGate
#
security-policy ip
 rule 4 name IPsec_to_FortiGate_1_20230407151324_IN
  action pass
  destination-zone Local
  service ike
  service nat-t-ipsec
  service ipsec-ah
  service ipsec-esp
 rule 3 name IPsec_to_FortiGate_1_20230407151324_OUT
  action pass
  source-zone Local
  service ike
  service nat-t-ipsec
  service ipsec-ah
  service ipsec-esp
```

5. 配置 IPSec VPN 相关网段放通的安全策略，并将他们移动至上网策略之前。



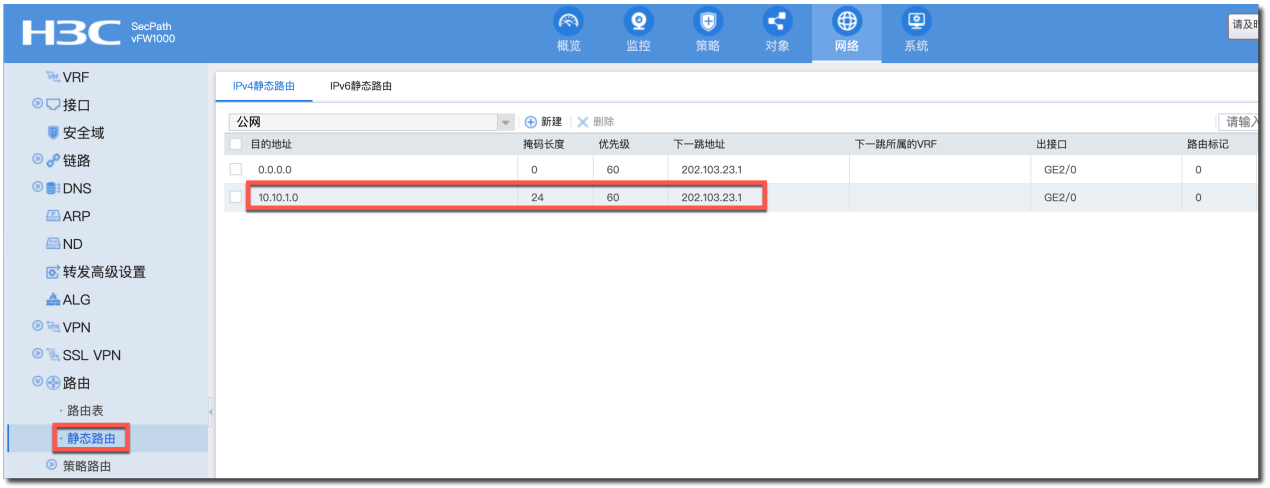
```
security-policy ip
 rule 6 name VPN_from_FortiGate
  action pass
  source-zone Untrust
  destination-zone Trust
  source-ip Remote_10.10.1.0/24
  destination-ip Local_10.10.2.0/24
 rule 5 name VPN_to_FortiGate
  action pass
  source-zone Trust
  destination-zone Untrust
  source-ip Local_10.10.2.0/24
  destination-ip Remote_10.10.1.0/24
 rule 4 name IPsec_to_FortiGate_1_20230407151324_IN
  action pass
  destination-zone Local
  service ike
  service nat-t-ipsec
  service ipsec-ah
  service ipsec-esp
 rule 3 name IPsec_to_FortiGate_1_20230407151324_OUT
  action pass
  source-zone Local
  service ike
  service nat-t-ipsec
  service ipsec-ah
  service ipsec-esp
 rule 2 name to_Internet
  action pass
  source-zone Trust
  destination-zone Untrust
  source-ip Local_10.10.2.0/24
```

6. H3C 防火墙的转发流程是先做 SNAT，再做 IPsec VPN，也就是说，如果按照我们之前配置的 SNAT 规则，VPN 的流量 10.10.2.2 去访问 10.10.1.2 的流量会被 SNAT 成公网 IP 202.103.23.2，然后再进入到 IPsec VPN 流程，这个时候是无法匹配到感兴趣流的，因此流量将无法成功被 IPSec 加密。所以需要再配置一个 VPN 流量 bypass 的 SNAT 规则，让 VPN 流量不做 SNAT 转换，然后将其放置到最前面，优先匹配即可。




```
nat policy
 rule name VPN_to_FortiGate
  source-ip Local_10.10.2.0/24
  destination-ip Remote_10.10.1.0/24
  outbound-interface GigabitEthernet2/0
  action no-nat
 rule name to_Internet
  source-ip Local_10.10.2.0/24
  outbound-interface GigabitEthernet2/0
  action easy-ip
```

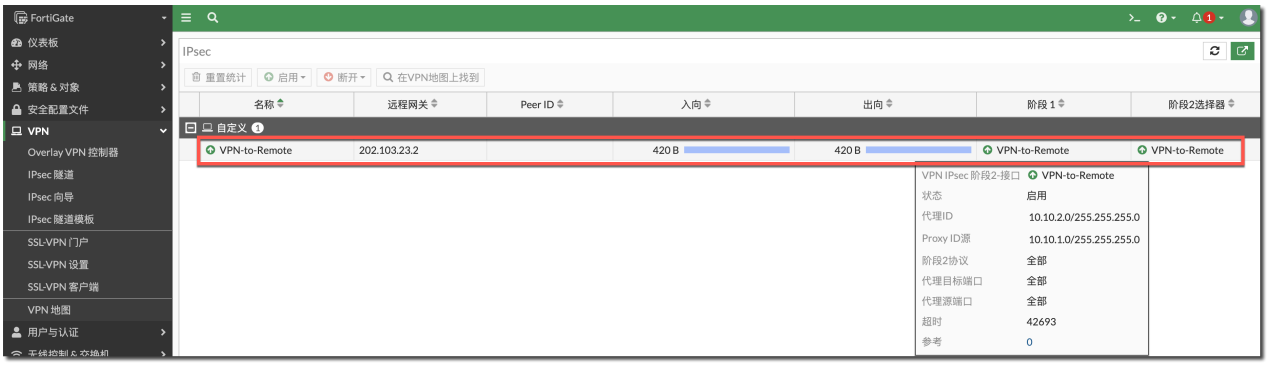
7. 添加去往 FortiGate 内网 VPN 网段的静态路由（由于已经配置了默认路由从 IPSec 绑定的物理口出去，其实这里不需要配置，但如果没有默认路由或默认路由是其他接口，可以参考此配置）。




```
ip route-static 10.10.1.0 24 GigabitEthernet2/0 202.103.23.1
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。




```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 2435s ago
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/38  established 1/38  time 20/75/2030 ms
  id/spi: 365 3f178bf07066db5b/4bc6621ed12c3030
  direction: initiator
  status: established 2435-2435s ago = 20ms
  proposal: aes256-sha256
  key: 2f13f4b81d3eb4c1-575e1272457cc854-d37978c764f9a778-cd5724cdc244f07a
  lifetime/rekey: 86400/83664
  DPD sent/recv: 00000a61/000030dd

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=1 202.103.13.2:0→202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0
proxyid_num=1 child_num=0 refcnt=4 ilast=13 olast=13 ad=/0
stat: rxp=10 txp=15 rxb=840 txb=1260
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=2658
natt: mode=none draft=0 interval=0 remote_port=0
```

```
proxyid=VPN-to-Remote proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.2.0-10.10.2.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=42881/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42901/43200
  dec: spi=34dc89a0 esp=aes key=32 9ebf3ec4affd4b4a57b18fe15c1c505184622c2cdcba87d788887f569cfaa23e
       ah=sha256 key=32 c3efca48ffbcd7fb0e8013fef745576010a71ec3c9a02a32f333c380de01e2a6
  enc: spi=bc07e7b7 esp=aes key=32 660dedaa1a6d145b94073c031a3e9f1587dcb44d3b50b9683382b27ecd5562ce
       ah=sha256 key=32 308adc9899b8775255c59053ed4a5edc3df8231135007d2a26614097619361de
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=0
```

2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

3. FortiGate 侧 PC1 业务测试。

```
VPCS> show
NAME   IP/MASK              GATEWAY                          GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6804/64

VPCS> ping 10.10.2.2
84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=4.769 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=2.769 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=2.194 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=3.881 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=2.335 ms
```

4. H3C 防火墙查看 IPSec 连接状态。




```
<H3C>display ike sa verbose remote 202.103.13.2
   ____________________________________________

   Connection ID: 2
   Outside VPN:
   Inside VPN:
```

```
  Profile: to_FortiGate_IPv4_1
  Transmitting entity: Responder
  Initiator cookie: 3f178bf07066db5b
  Responder cookie: 4bc6621ed12c3030
  ------------------------------------------------

  Local IP/port: 202.103.23.2/500
  Local ID type: IPV4_ADDR
  Local ID: 202.103.23.2

  Remote IP/port: 202.103.13.2/500
  Remote ID type: IPV4_ADDR
  Remote ID: 202.103.13.2

  Authentication-method: PRE-SHARED-KEY
  Authentication-algorithm: SHA256
  Encryption-algorithm: AES-CBC-256

  Life duration(sec): 86400
  Remaining key duration(sec): 83791
  Exchange-mode: Main
  Diffie-Hellman group: Group 14
  NAT traversal: Not detected

  Extend authentication: Disabled
  Assigned IP address:
  Vendor ID index:0×ffffffff
  Vendor ID sequence number:0×0
```

```
<H3C> display ipsec sa remote 202.103.13.2
-----------------------------
Interface: GigabitEthernet2/0
-----------------------------

  -----------------------------
  IPsec policy: to_FortiGate
  Sequence number: 1
  Mode: ISAKMP
  -----------------------------
    Tunnel id: 0
    Encapsulation mode: tunnel
    Perfect Forward Secrecy: dh-group14
    Inside VPN:
    Extended Sequence Numbers enable: N
    Traffic Flow Confidentiality enable: N
    Transmitting entity: Responder
    Path MTU: 1424
    Tunnel:
        local  address: 202.103.23.2
        remote address: 202.103.13.2
    Flow:
        sour addr: 10.10.2.0/255.255.255.0  port: 0  protocol: ip
        dest addr: 10.10.1.0/255.255.255.0  port: 0  protocol: ip

    [Inbound ESP SAs]
      SPI: 1271313254 (0×4bc6b366)
      Connection ID: 347892350977
      Transform set: ESP-ENCRYPT-AES-CBC-256 ESP-AUTH-SHA256
      SA idle time: 60
      SA duration (kilobytes/sec): 1843200/43200
      SA remaining duration (kilobytes/sec): 1843200/43177
      Max received sequence-number: 0
      Anti-replay check enable: Y
      Anti-replay window size: 64
      UDP encapsulation used for NAT traversal: N
      Status: Active

    [Outbound ESP SAs]
      SPI: 886868385 (0×34dc89a1)
      Connection ID: 347892350976
      Transform set: ESP-ENCRYPT-AES-CBC-256 ESP-AUTH-SHA256
      SA idle time: 60
```

```
      SA duration (kilobytes/sec): 1843200/43200
      SA remaining duration (kilobytes/sec): 1843200/43177
      Max sent sequence-number: 0
      UDP encapsulation used for NAT traversal: N
      Status: Active
```

5. 查看 H3C 防火墙的路由表，可以看到包含了手动配置的到 FortiGate 内网的路由。

```
<H3C>display ip routing-table
Destinations : 18       Routes : 18
Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/0          Static  60  0           202.103.23.1    GE2/0
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
10.10.1.0/24       Static  60  0           202.103.23.1    GE2/0
10.10.2.0/24       Direct  0   0           10.10.2.1       GE3/0
10.10.2.1/32       Direct  0   0           127.0.0.1       InLoop0
10.10.2.255/32     Direct  0   0           10.10.2.1       GE3/0
127.0.0.0/8        Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32       Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.100.0/24   Direct  0   0           192.168.100.6   GE1/0
192.168.100.6/32   Direct  0   0           127.0.0.1       InLoop0
192.168.100.255/32 Direct  0   0           192.168.100.6   GE1/0
202.103.23.0/24    Direct  0   0           202.103.23.2    GE2/0
202.103.23.2/32    Direct  0   0           127.0.0.1       InLoop0
202.103.23.255/32  Direct  0   0           202.103.23.2    GE2/0
224.0.0.0/4        Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24       Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

6. H3C 防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=3.529 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=2.741 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=2.472 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.250 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=2.193 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-07 16:07:08.123477 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:08.123667 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:08.123902 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:08.123914 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:09.127191 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:09.127380 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:09.127691 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:09.127700 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:10.131278 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:10.131445 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:10.131618 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:10.131624 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:11.134468 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:11.134502 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:11.134737 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

```
2023-04-07 16:07:11.134744 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:12.137190 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:12.137220 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-07 16:07:12.137409 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-07 16:07:12.137416 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

关于 sniffer 抓 VPN 业务和 ESP 的包：

- 抓取 IPsec VPN 的 IKE 协商包：

```
diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
```

- 抓取 IPsec VPN 的 ESP 加密数据包：

```
diagnose sniffer packet any "host 101.1.1.1 and esp" 4
```

- 抓取 IPsec VPN 的明文业务数据包：

```
diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
```

- 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

```
FortiGate # config vpn ipsec phase1-interface
FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
FortiGate (BJ-OSPF-TO-SH) # end
```

与山石 SG6000 防火墙建立 IPSec VPN

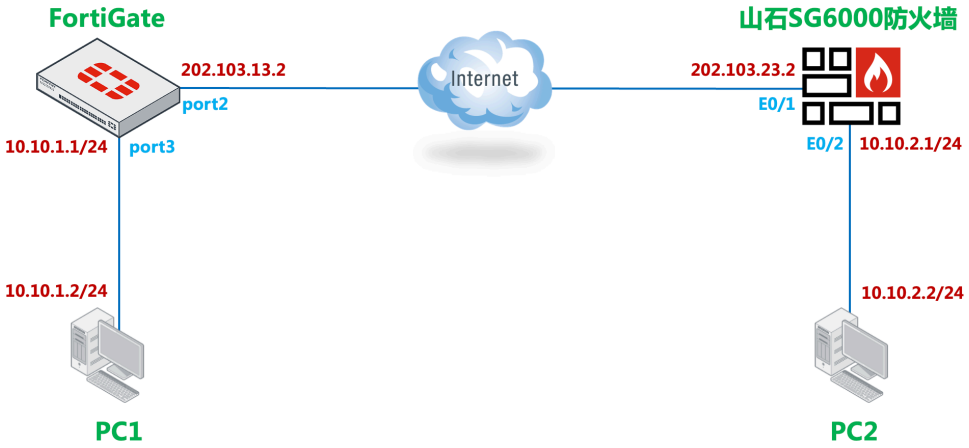
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与山石 SG6000-CloudEdge 防火墙进行 IPsec VPN（IKE v1）对接。

山石防火墙版本：

```
SG-6000# show version
Hillstone Networks StoneOS software, Version 5.5
Copyright (c) 2009-2023 by Hillstone Networks

Product name: SG-6000-VM00 S/N: 0010082687152785 Assembly number: 0000
Boot file is SG6000-CloudEdge-5.5R10P1-v6
Storage UUID is 2a771694-c137-483a-a7e2-ff93722c32ee
Instance UUID is iid-datasource-none
Update magic: 002300130532f202d5
Built by buildmaster8 2023/03/14 07:04:14
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置山石 SG6000 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

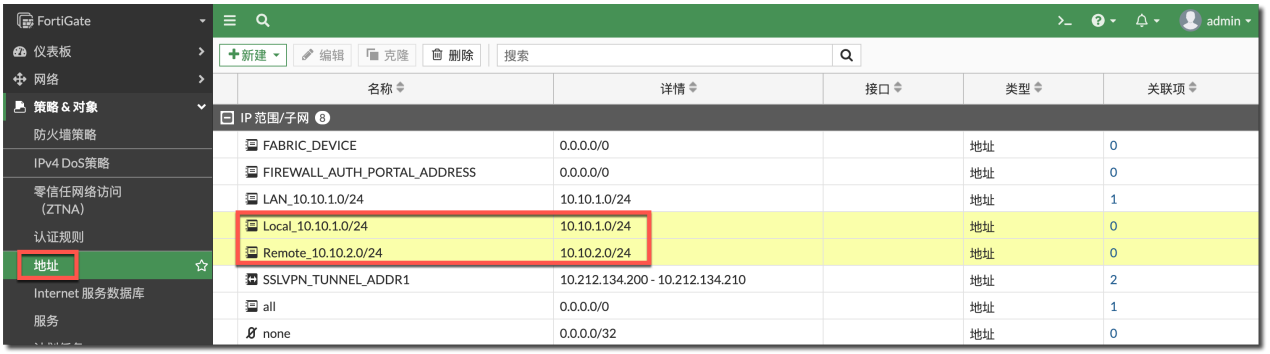
重要

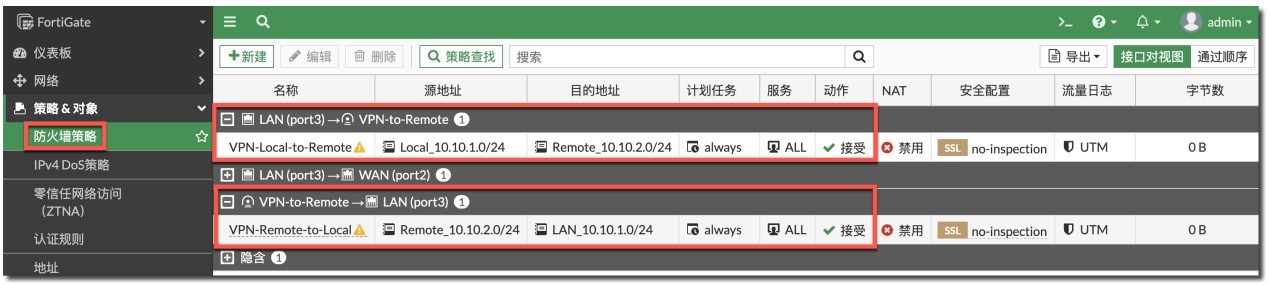
和山石/Juniper/Palo Alto 防火墙对接 IPSec 推荐使用全 0 的感兴趣流，这些厂商设备与 FortiGate 一致，创建 IPSec 时会创建虚拟的 IPSec 接口，通过路由对 VPN 流量进行引流，将需要加密的数据送到 IPSec 接口（当然写细化的感兴趣流也是可以的）。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
0.0.0.0/0.0.0.0
0.0.0.0/0.0.0.0
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
0.0.0.0/0.0.0.0
对端地址
子网
0.0.0.0/0.0.0.0
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 14 5 2
        set auto-negotiate enable
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。






```
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
```

```
        set srcintf "VPN-to-Remote"
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

重要

相关 VPN 的路由配置中“黑洞路由的意义：

你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?

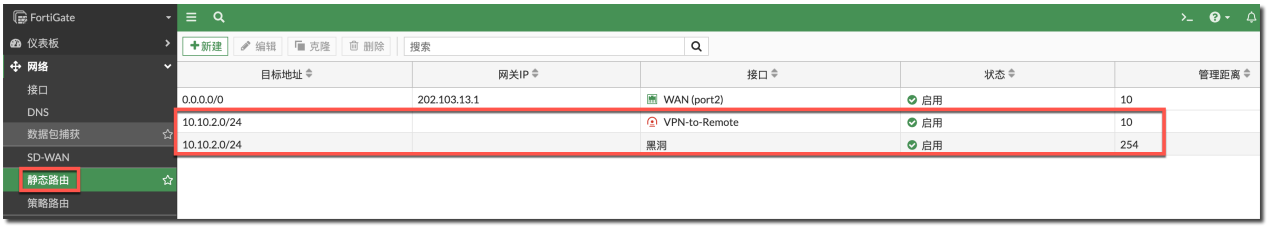
VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。

解决方法：

- 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
- 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
- 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。

```
FortiGate # config system global
FortiGate (global) # set snat-route-change enable
FortiGate (global) # end
```

三种方式任意选择一种即可。推荐使用黑洞路由方式。

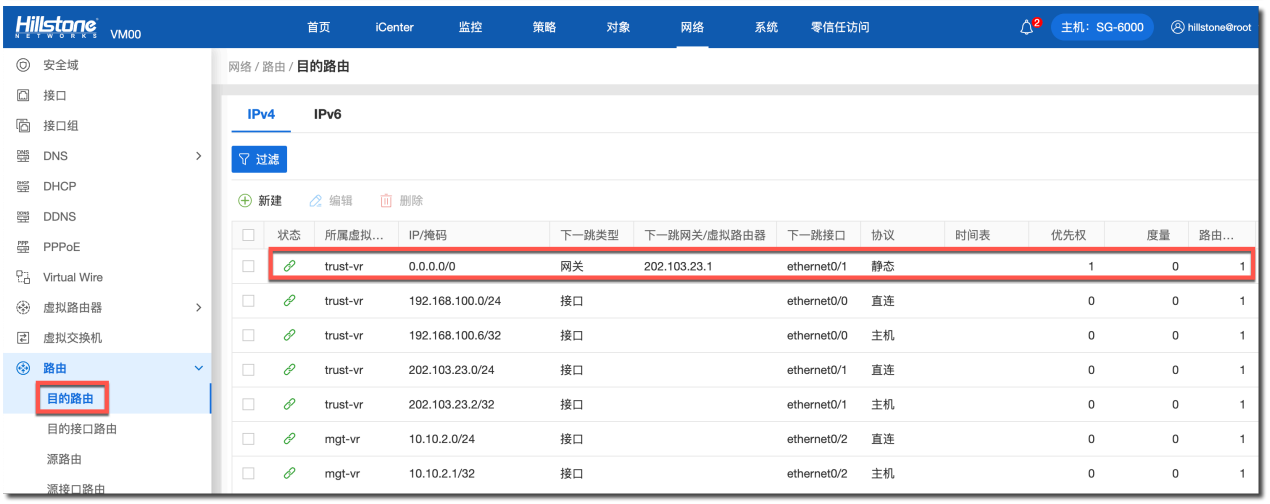



```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

山石 SG6000

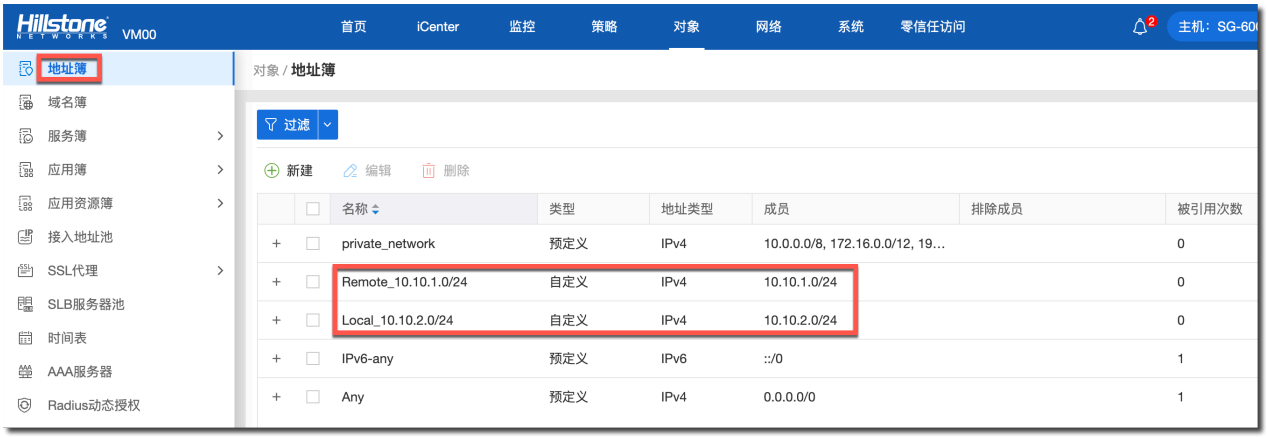
1. 基本 IP 与路由配置。

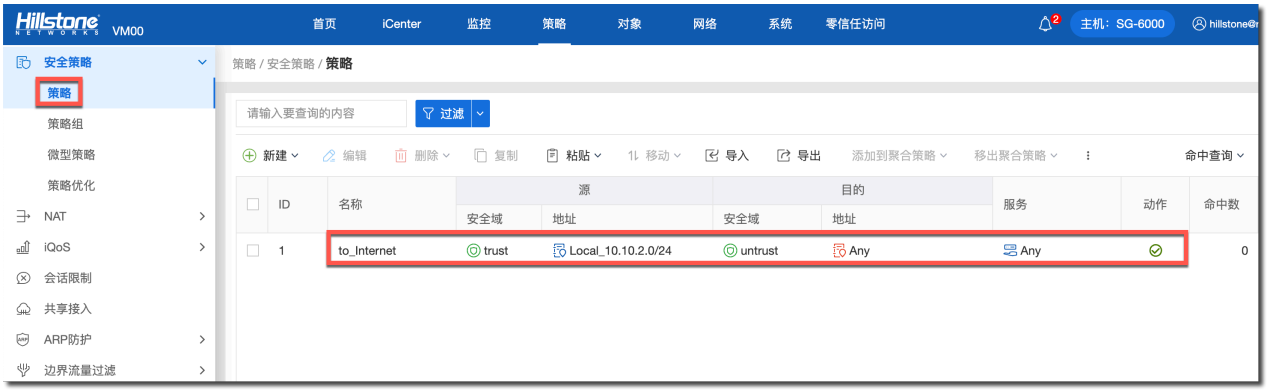


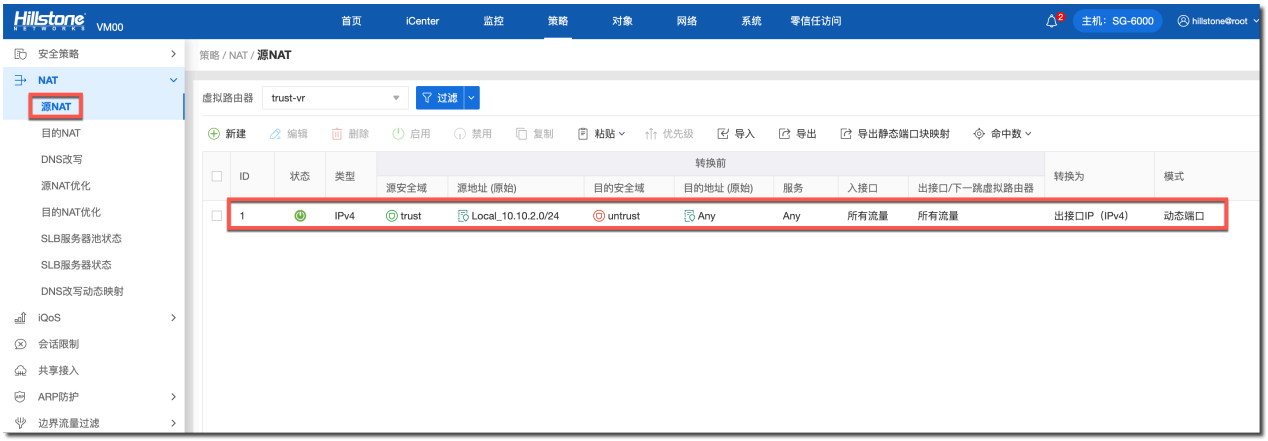



```
interface ethernet0/1
  zone  "untrust"
  ip address 202.103.23.2 255.255.255.0
  manage ping
exit
interface ethernet0/2
  zone  "trust"
  ip address 10.10.2.1 255.255.255.0
  manage ping
exit

ip vrouter "trust-vr"
  ip route 0.0.0.0/0 202.103.23.1
exit
```

2. 地址对象、策略、NAT 配置。





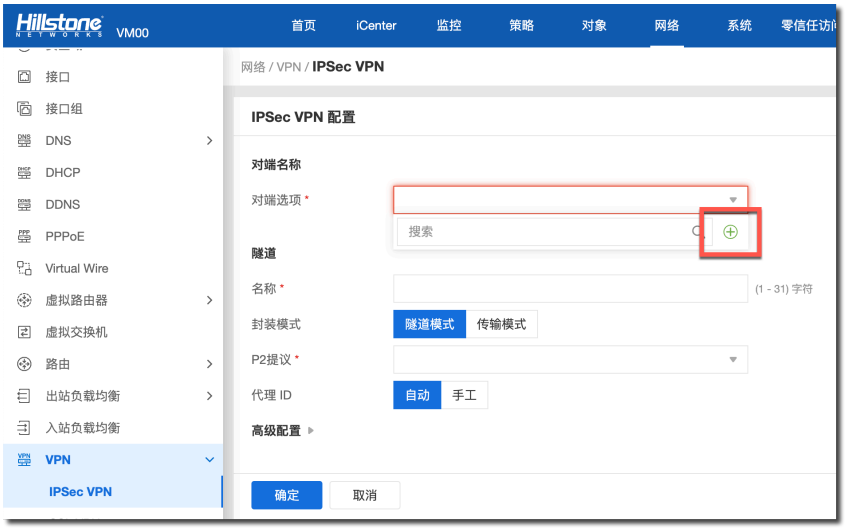



```
address "Local_10.10.2.0/24"
  ip 10.10.2.0/24
exit
address "Remote_10.10.1.0/24"
  ip 10.10.1.0/24
exit

rule id 1
  action permit
  src-zone "trust"
  dst-zone "untrust"
  src-addr "Local_10.10.2.0/24"
  dst-addr "Any"
  service "Any"
  name "to_Internet"
exit

ip vrouter "trust-vr"
  snatrule id 1 from-zone "trust" to-zone "untrust" from address-book "Local_10.10.2.0/24" to address-book "Any" service
"Any" trans-to eif-ip mode dynamicport
exit
```

3. 在 IPSec VPN 页面点击新建按钮。



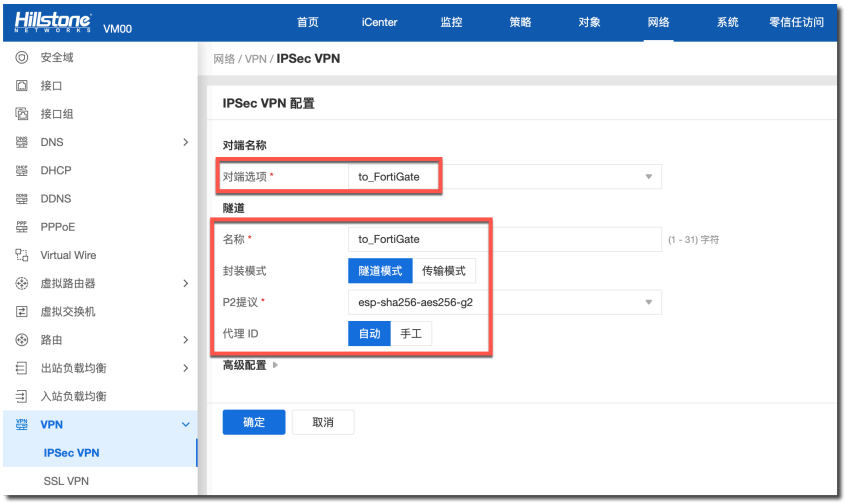
4. 点击对端选项，点击右侧的“+”，创建新的对端选项，如下图配置 VPN 对端的相关配置，这里配置的是第一阶段安全提议，注意与 FortiGate 配置一致，并开启 DPD，点击确定下发对端配置。





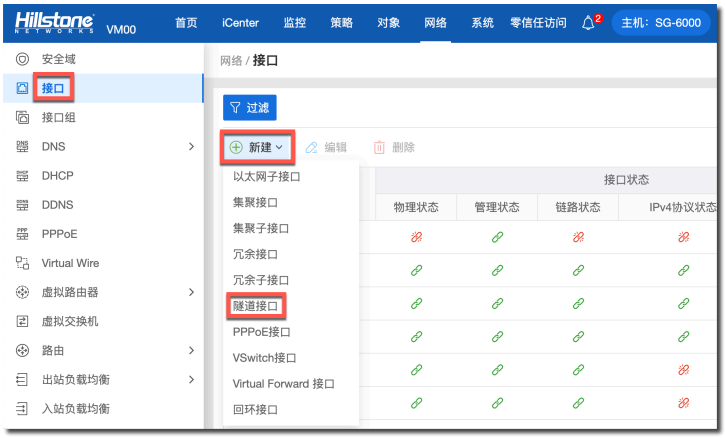

```
isakmp peer "to_FortiGate"
  isakmp-proposal "psk-sha256-aes256-g2"
  pre-share "$01011NMr$XE3F0ui+whYFfilc+Fx6mDJ7I8Y="
  peer 202.103.13.2
  accept-all-peer-id
  dpd interval 10 retry 3 periodic
  interface ethernet0/1
exit
```

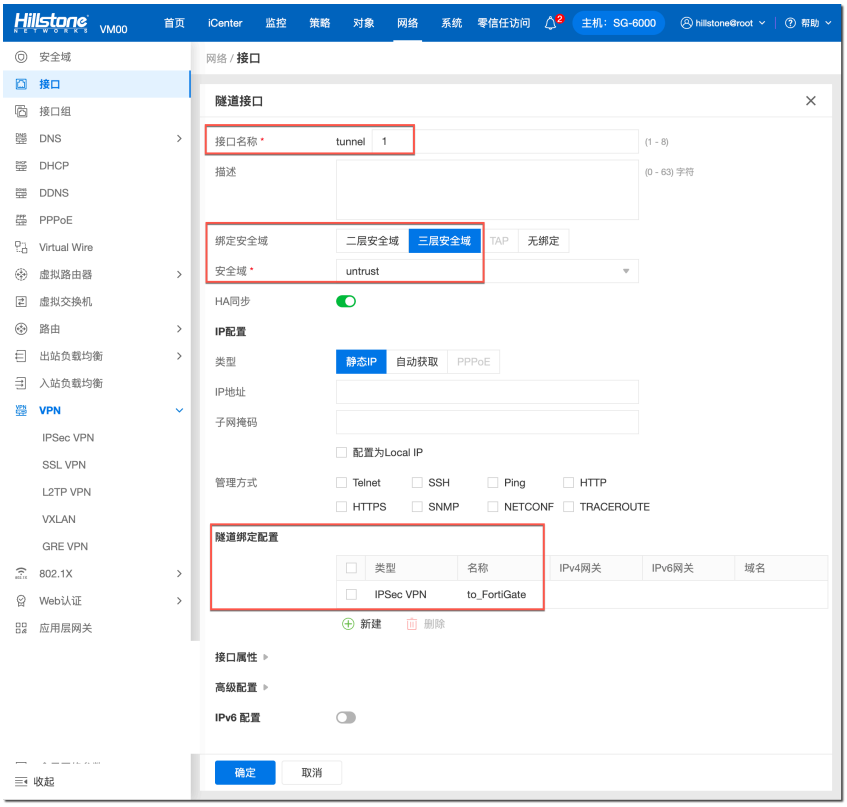
5. “VPN 对端配置”创建完成后，继续配置 IPSec VPN，对端选项选择上步创建的“to_FortiGate”，配置二阶段安全提议与 FortiGate 端一致，高级配置中无需配置。




```
tunnel ipsec "to_FortiGate" auto sa-index 2
  ipsec-proposal "esp-sha256-aes128-g2"
  isakmp-peer "to_FortiGate"
exit
```

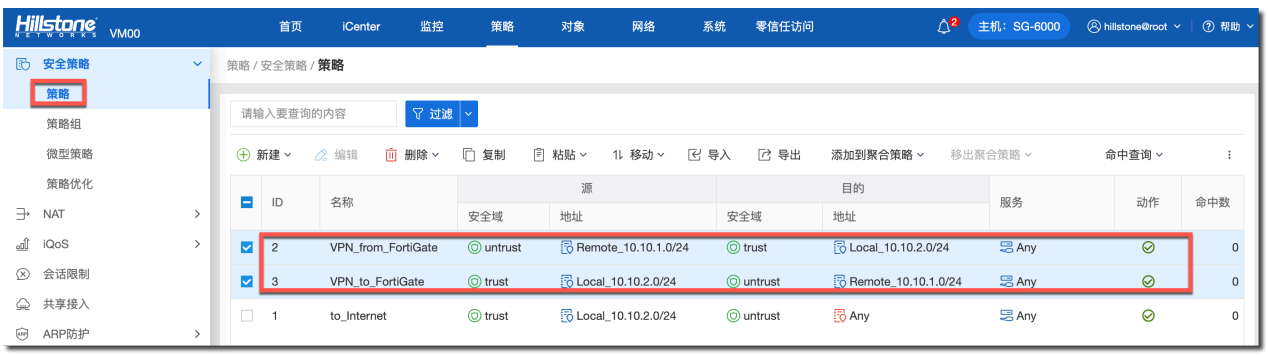
6. 创建隧道接口，并将 IPSec VPN 连接绑定至 IPSec 接口。





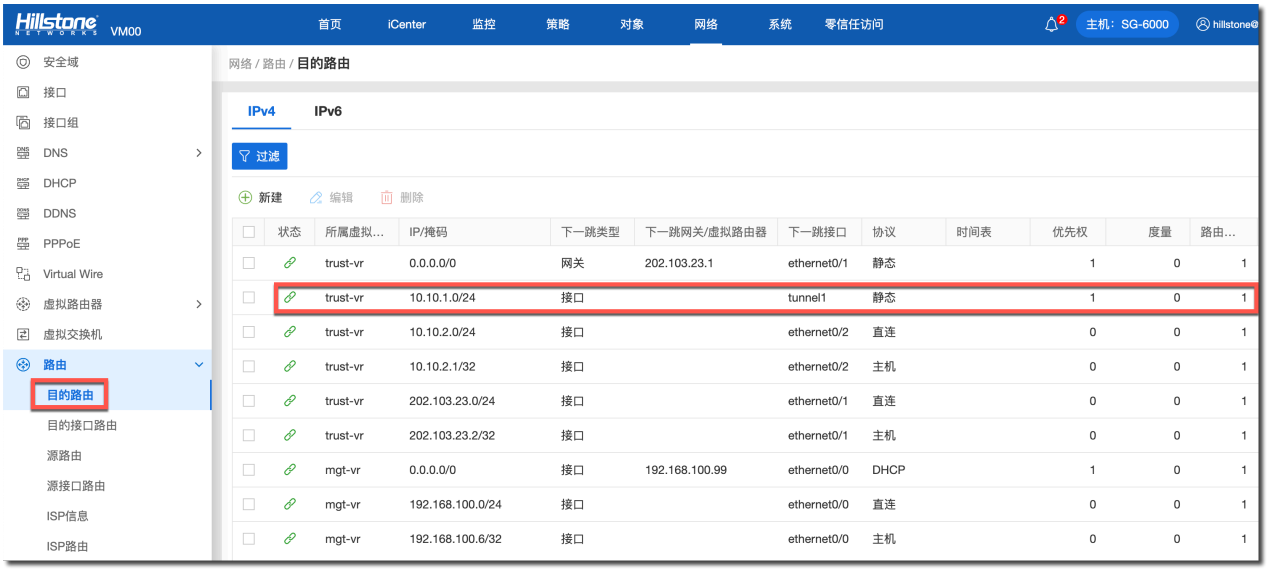

```
interface tunnel1
  zone  "untrust"
  tunnel ipsec "to_FortiGate"
exit
```

7. 配置 IPSec VPN 相关网段放通的安全策略，并将他们移动至上网策略之前。



```
rule id 2
  action permit
  src-zone "untrust"
  dst-zone "trust"
  src-addr "Remote_10.10.1.0/24"
  dst-addr "Local_10.10.2.0/24"
  service "Any"
  name "VPN_from_FortiGate"
exit
rule id 3
  action permit
  src-zone "trust"
  dst-zone "untrust"
  src-addr "Local_10.10.2.0/24"
  dst-addr "Remote_10.10.1.0/24"
  service "Any"
  name "VPN_to_FortiGate"
exit
rule id 1
  action permit
  src-zone "trust"
  dst-zone "untrust"
  src-addr "Local_10.10.2.0/24"
  dst-addr "Any"
  service "Any"
  name "to_Internet"
exit
```

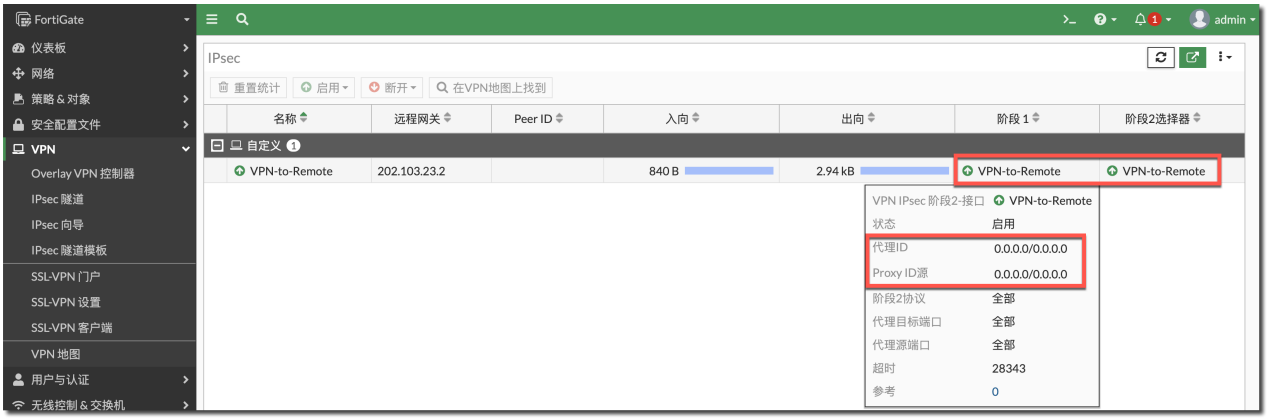
8. 添加去往 FortiGate 内网 VPN 网段的静态路由，出接口为 tunnel 接口。




```
ip vrouter "trust-vr"
  ip route 10.10.1.0/24 "tunnel1"
exit
```

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。




```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 1254s ago
IKE SA: created 1/1  established 1/1  time 120/120/120 ms
IPsec SA: created 1/7  established 1/2  time 0/65/130 ms
  id/spi: 6 682ffbe3bdcd7130/bca06b5547f0a164
  direction: initiator
  status: established 1254-1254s ago = 120ms
  proposal: aes256-sha256
  key: a45c9a51ac96f767-98ce117bc21328d4-431c2a6eeceb933e-af19239a1116dd28
  lifetime/rekey: 86400/84845
  DPD sent/recv: 00000000/000003bd

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=2 202.103.13.2:0→202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=on weig1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overla0
proxyid_num=1 child_num=0 refcnt=4 ilast=6 olast=6 ad=/0
stat: rxp=10 txp=35 rxb=840 txb=2940
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Remote proto=0 sa=1 ref=2 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=28252/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=28503/28800
  dec: spi=92c2ead0 esp=aes key=32 22c824606a1f39ec9f9ce853b7e78f0bceacf61695360474d5365d4df8448f0b
       ah=sha256 key=32 f501d221d046169419833ad4c6f6aa6d90155700048590023024a6e48dc12e3b
  enc: spi=2008ef27 esp=aes key=32 fbc3d717c22ec085dac575010867f28b798f13b0f95d92f37ebe1eb9baf9d997
       ah=sha256 key=32 4dc43f2f80f5b212cb399272a4fc2b765351b7c835bce4fe56370a06692b6b67
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=1 dec_npuid=0 enc_npuid=0
run_tally=0
```

2. 查看 FortiGate 的路由表。

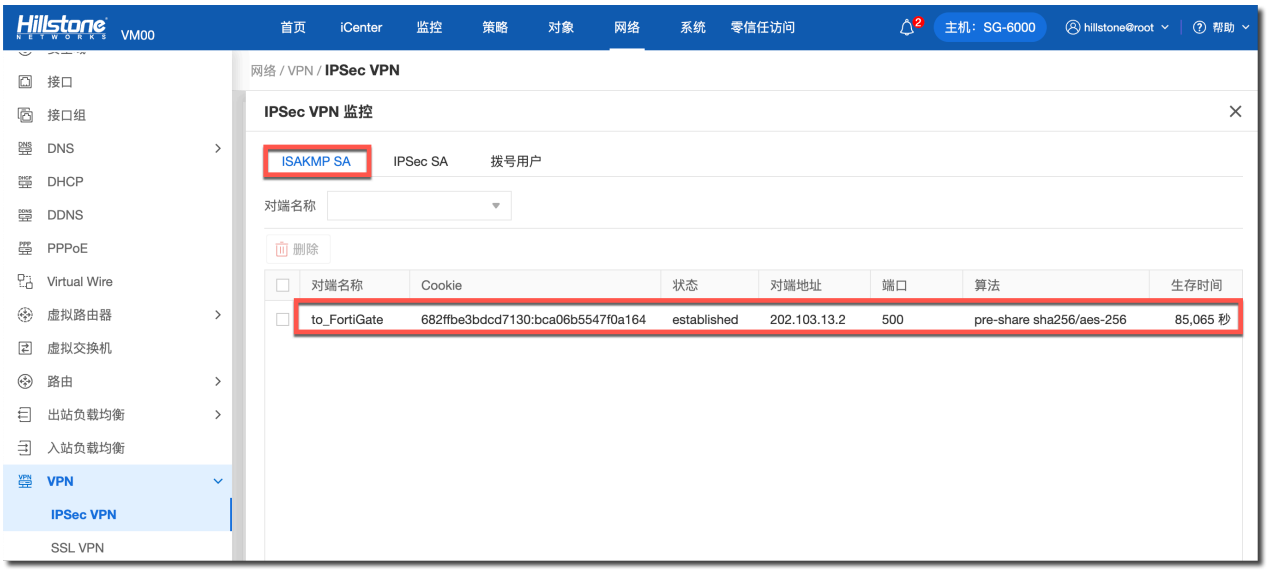
```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
```

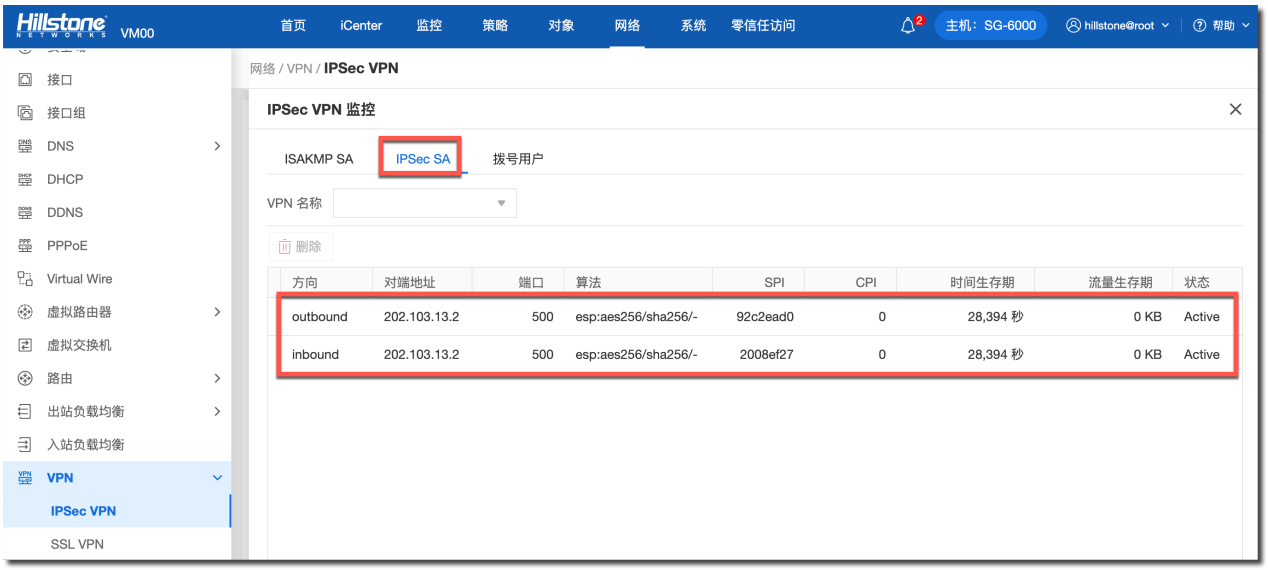
```
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

3. FortiGate 侧 PC1 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.2.2

84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=3.442 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=4.599 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=1.291 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=1.419 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=1.397 ms
```

4. 山石防火墙查看 IPSec 连接状态。





```
SG-6000(config)# show isakmp sa
Total: 1
═══════════════════════════════════════════════════════════════════════════════════════
Cookies                            Gateway              Port      Algorithms                Lifetime
───────────────────────────────────────────────────────────────────────────────────────
682ffbe3bdcd7130:bca06b5547f0a164  202.103.13.2         500       pre-share  sha256/aes-256 84848
═══════════════════════════════════════════════════════════════════════════════════════

SG-6000(config)# show ipsec sa
Total: 1
S - Status, I - Inactive, A - Active;
═══════════════════════════════════════════════════════════════════════════════════════
Id    VPN                             Peer IP           Port Algorithms           SPI      Life(s) S
───────────────────────────────────────────────────────────────────────────────────────
3     to_FortiGate                    >202.103.13.2      500 esp:aes256/sha256/-  92c2ead0 28243   A
3     to_FortiGate                    <202.103.13.2      500 esp:aes256/sha256/-  2008ef27 28243   A
═══════════════════════════════════════════════════════════════════════════════════════
```

5. 查看山石防火墙的路由表，可以看到包含了手动配置的到 FortiGate 内网的路由。

```
SG-6000(config)# show ip route
Codes: K - kernel route, C - connected, S - static, N - Domain, Z - ISP, R - RIP, O - OSPF,
       B - BGP, D - DHCP, P - PPPoE, W - wireless, H - HOST, G - SCVPN, V - VPN, M - IMPORT,
       I - ISIS, Y - SYNC, L - llb outbound, > - selected first nexthop, * - FIB route, b - BFD enable

Routing Table for Virtual Router <trust-vr>
═══════════════════════════════════════════════════════════════════════
S>* 0.0.0.0/0 [1/0/1] via 202.103.23.1, ethernet0/1
S>* 10.10.1.0/24 [1/0/1] is directly connected, tunnel1
C>* 10.10.2.0/24 is directly connected, ethernet0/2
H>* 10.10.2.1/32 [0/0/1] is local address, ethernet0/2
C>* 202.103.23.0/24 is directly connected, ethernet0/1
H>* 202.103.23.2/32 [0/0/1] is local address, ethernet0/1
```

6. 山石防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6806/64

VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=1.715 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=1.710 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=7.731 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.330 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=1.684 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-10 16:27:56.023973 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:56.024029 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:56.026679 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:56.026698 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:57.029816 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:57.029855 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:57.030097 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:57.030109 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:58.031820 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:58.031856 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:58.032037 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:58.032049 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:59.033955 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:27:59.033988 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
```

```
2023-04-10 16:27:59.034145 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:27:59.034151 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:28:00.036548 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:28:00.036586 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-10 16:28:00.036803 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-10 16:28:00.036816 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：
   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
     FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
     FortiGate (BJ-OSPF-TO-SH) # end
     ```

2. 山石 SG6000 防火墙 IPsec VPN Debug：

```
SG-6000[DBG]# debug vpn
SG-6000[DBG]# show logging debug
```

与 Palo Alto 防火墙建立 IPSec VPN

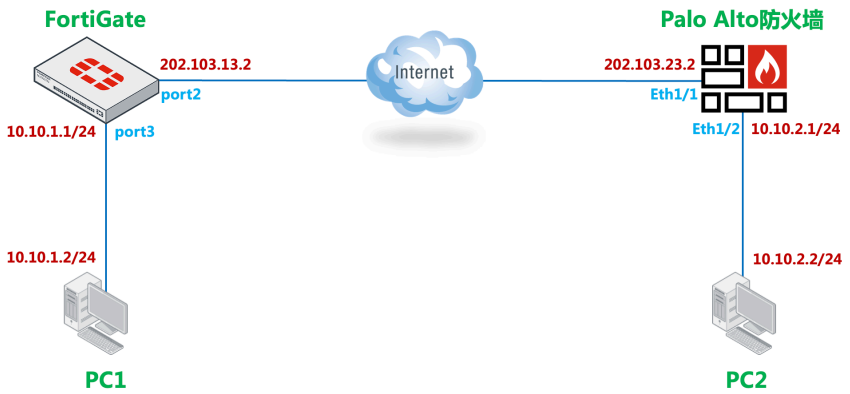
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与 Palo Alto 防火墙进行 IPsec VPN（IKE v1）对接。

Palo Alto 防火墙版本信息：

```
admin@PA-VM> show system info

hostname: PA-VM
ip-address: 192.168.100.3
public-ip-address: unknown
netmask: 255.255.255.0
default-gateway: 192.168.100.99
ip-assignment: dhcp
ipv6-address: unknown
ipv6-link-local-address: fe80::5200:ff:fe01:0/64
ipv6-default-gateway:
mac-address: 50:00:00:01:00:00
time: Tue Apr 11 19:13:23 2023
uptime: 0 days, 0:10:21
family: vm
model: PA-VM
serial: unknown
vm-mac-base: BA:DB:EE:FB:AD:00
vm-mac-count: 255
vm-uuid: A42F5C9E-B1D9-4A19-B229-D959AF933A2B
vm-cpuid: KVM:63060000FDFB8B07
vm-license: none
vm-cap-tier: unknown
vm-cpu-count: 2
vm-memory: 4031324
vm-mode: KVM
cloud-mode: non-cloud
sw-version: 10.1.0
global-protect-client-package-version: 0.0.0
device-dictionary-version: 1-211
device-dictionary-release-date:
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Palo Alto 防火墙
  - 基本上网配置
  - 配置 IPsec VPN

- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

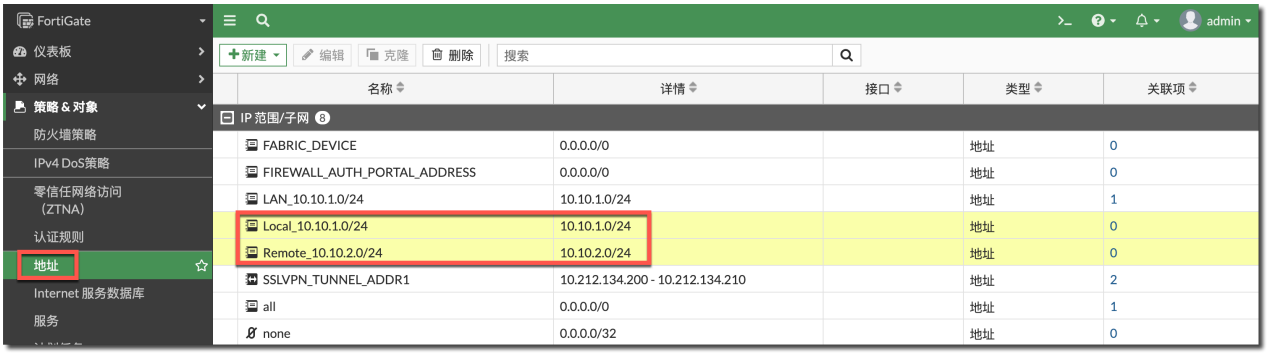
重要

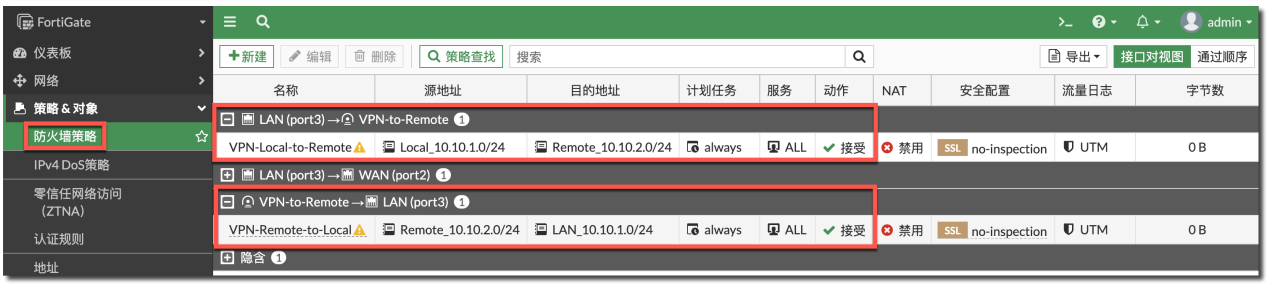
和山石/Palo Alto 防火墙对接 IPSec 推荐使用全 0 的感兴趣流，这些厂商设备与 FortiGate 一致，创建 IPSec 时会创建虚拟的 IPSec 接口，通过路由对 VPN 流量进行引流，将需要加密的数据送到 IPSec 接口（当然写细化的感兴趣流也是可以的）。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
0.0.0.0/0.0.0.0
0.0.0.0/0.0.0.0
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
0.0.0.0/0.0.0.0
对端地址
子网
0.0.0.0/0.0.0.0
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 14 5 2
        set auto-negotiate enable
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。






```
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
```

```
        set srcintf "VPN-to-Remote"
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

重要

相关 VPN 的路由配置中“黑洞路由的意义：

你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?

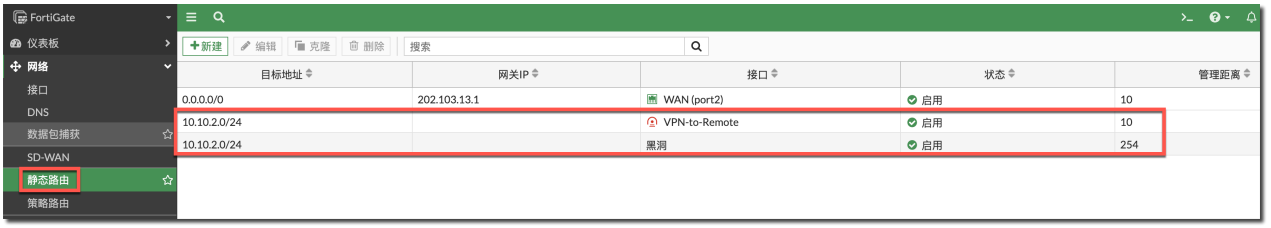
VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。

解决方法：

- 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
- 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
- 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。

```
FortiGate # config system global
FortiGate (global) # set snat-route-change enable
FortiGate (global) # end
```

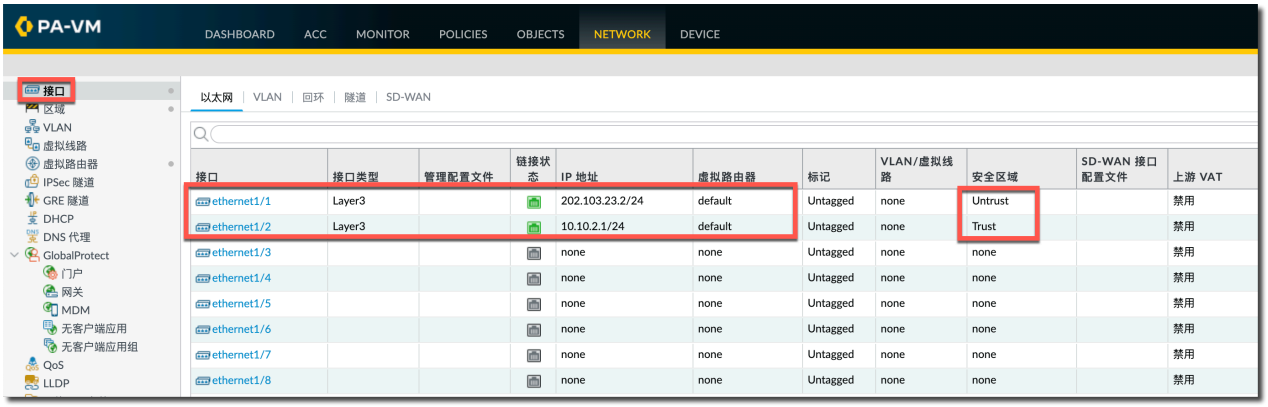
三种方式任意选择一种即可。推荐使用黑洞路由方式。

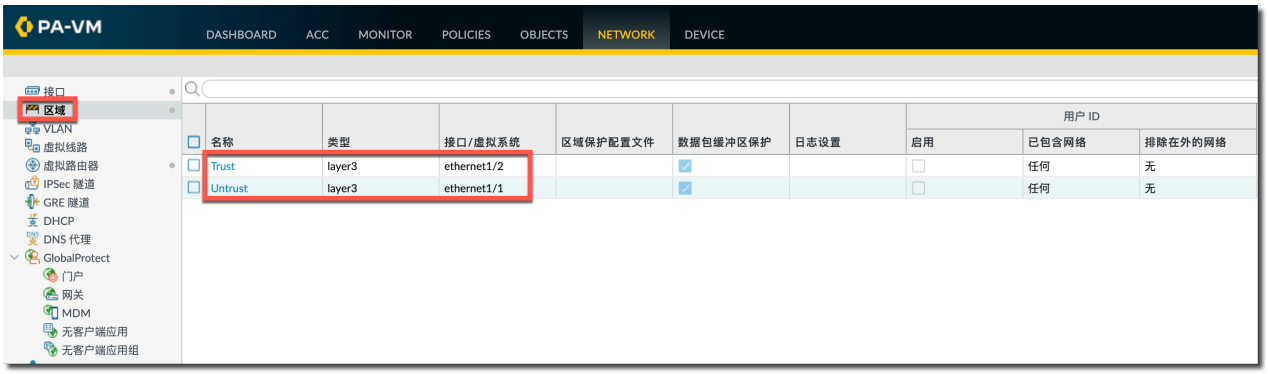


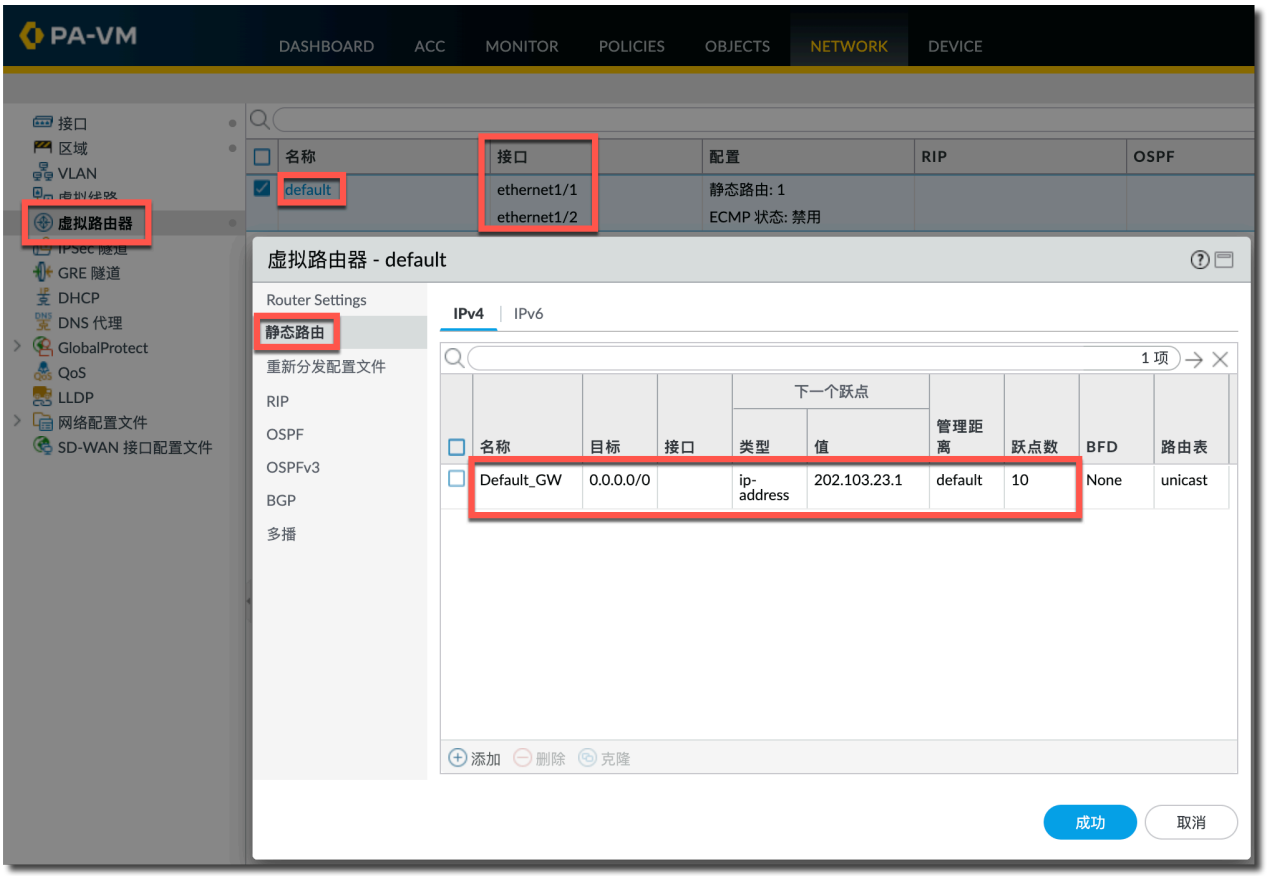

```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Palo Alto

1. 基本 IP 与路由配置，请注意一定要将接口加入安全域和虚拟路由器，否则接口无法通信。



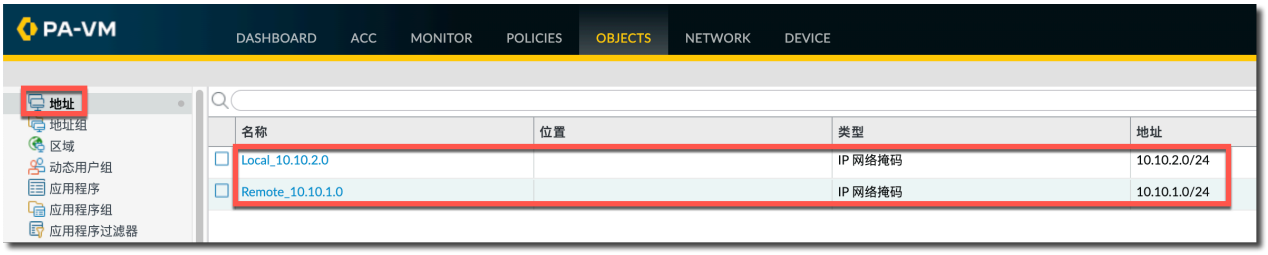




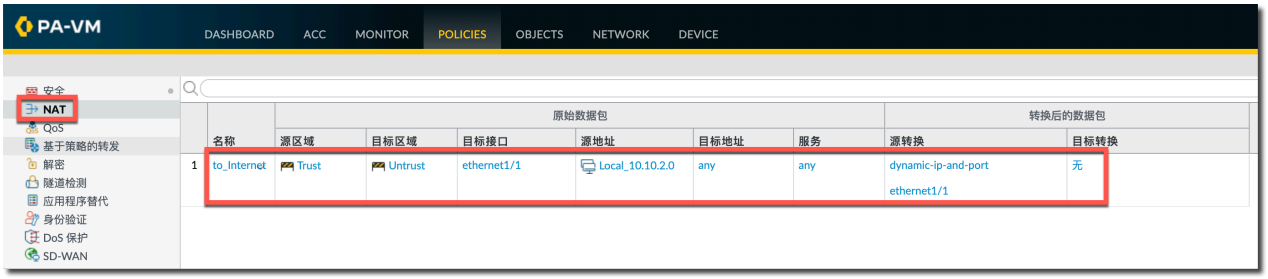

```
network {
  interface {
    ethernet {
      ethernet1/1 {
        layer3 {
          ip {
            202.103.23.2/24;
          }
        }
```

```
      }
      ethernet1/2 {
        layer3 {
          ip {
            10.10.2.1/24;
          }
        }
      }
    }
  }
  virtual-router {
    default {
      routing-table {
        ip {
          static-route {
            Default_GW {
              nexthop {
                ip-address 202.103.23.1;
              }
              metric 10;
              destination 0.0.0.0/0;
              route-table {
                unicast;
              }
            }
          }
        }
      }
      interface [ ethernet1/1 ethernet1/2];
    }
  }
}
```

2. 地址对象、策略、NAT 配置。



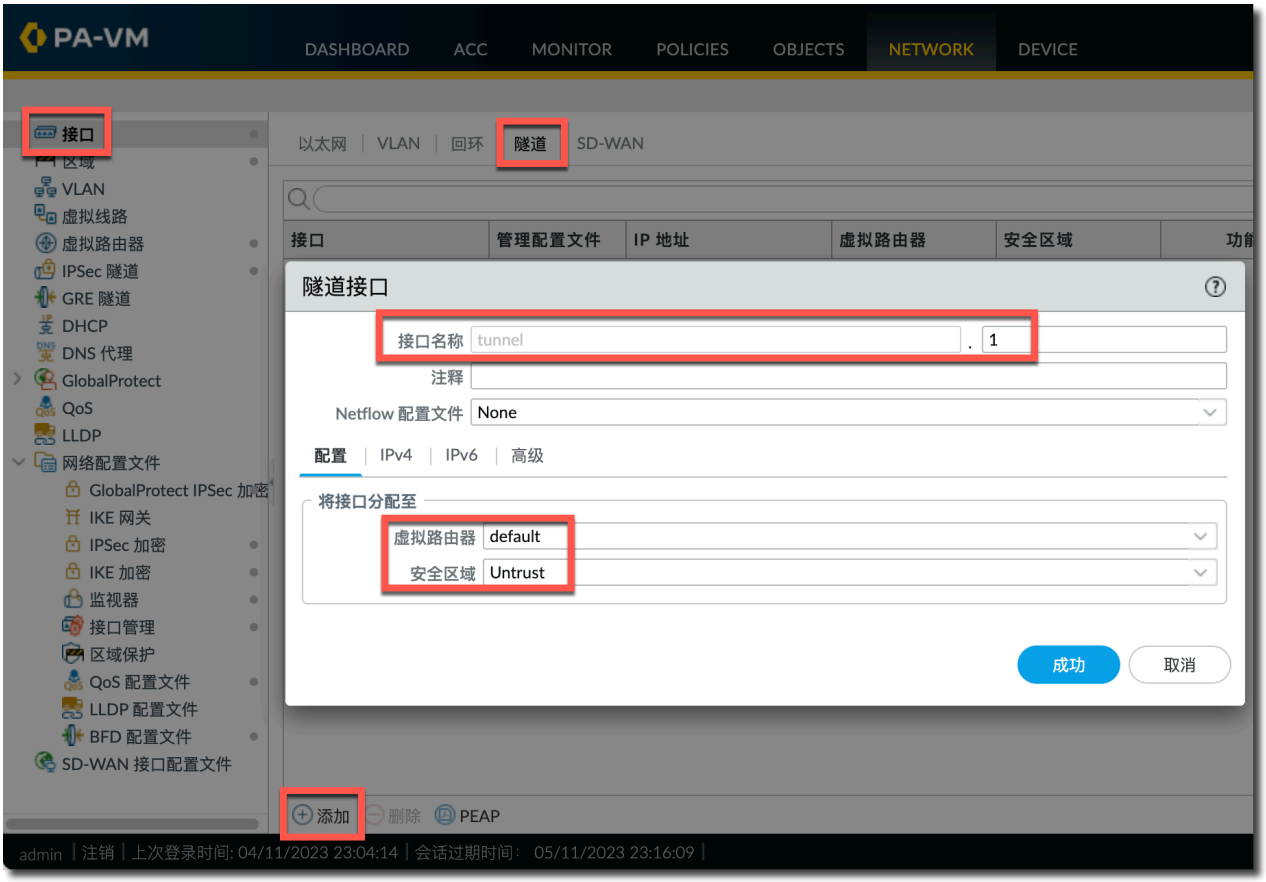





```
address {
  Local_10.10.2.0 {
    ip-netmask 10.10.2.0/24;
  }
  Remote_10.10.1.0 {
    ip-netmask 10.10.1.0/24;
  }
```

```
}

rulebase {
  security {
    rules {
      to_Internet {
        to Untrust;
        from Trust;
        source Local_10.10.2.0;
        destination any;
        source-user any;
        category any;
        application any;
        service application-default;
        source-hip any;
        destination-hip any;
        action allow;
      }
    }
  }
  nat {
    rules {
      to_Internet {
        source-translation {
          dynamic-ip-and-port {
            interface-address {
              interface ethernet1/1;
            }
          }
        }
        to Untrust;
        from Trust;
        source Local_10.10.2.0;
        destination any;
        service any;
        to-interface ethernet1/1;
      }
    }
  }
}
```

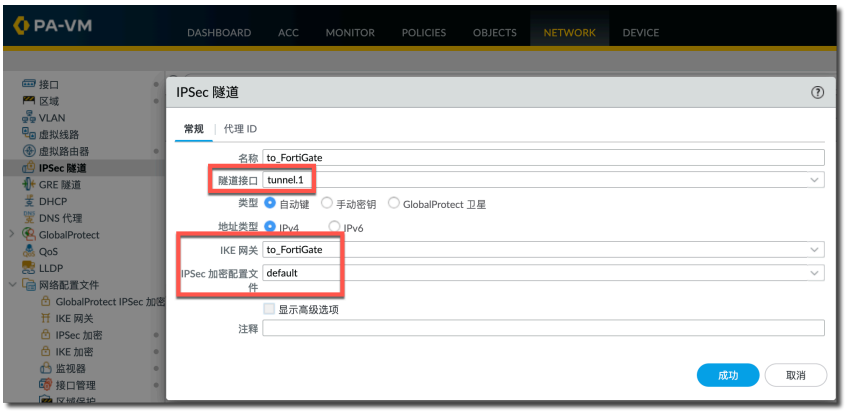
3. 创建 IPSec 隧道接口，并加入安全域和虚拟路由器。



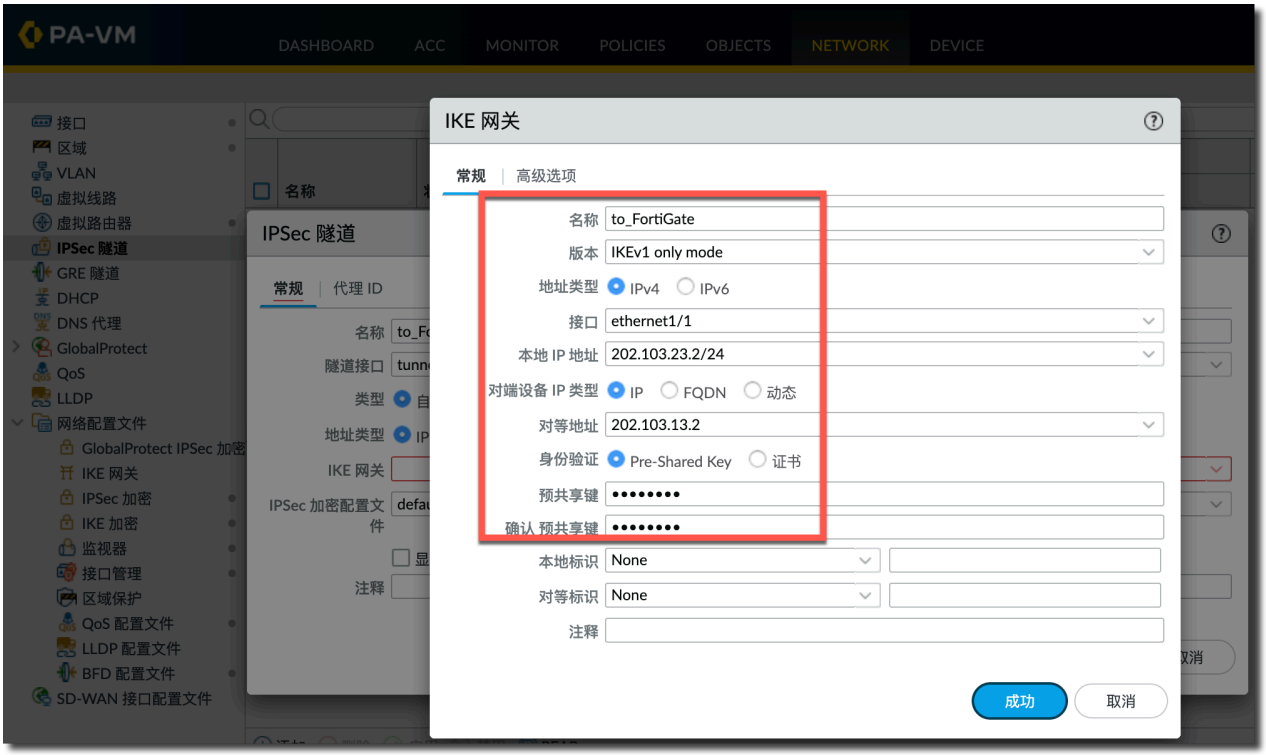


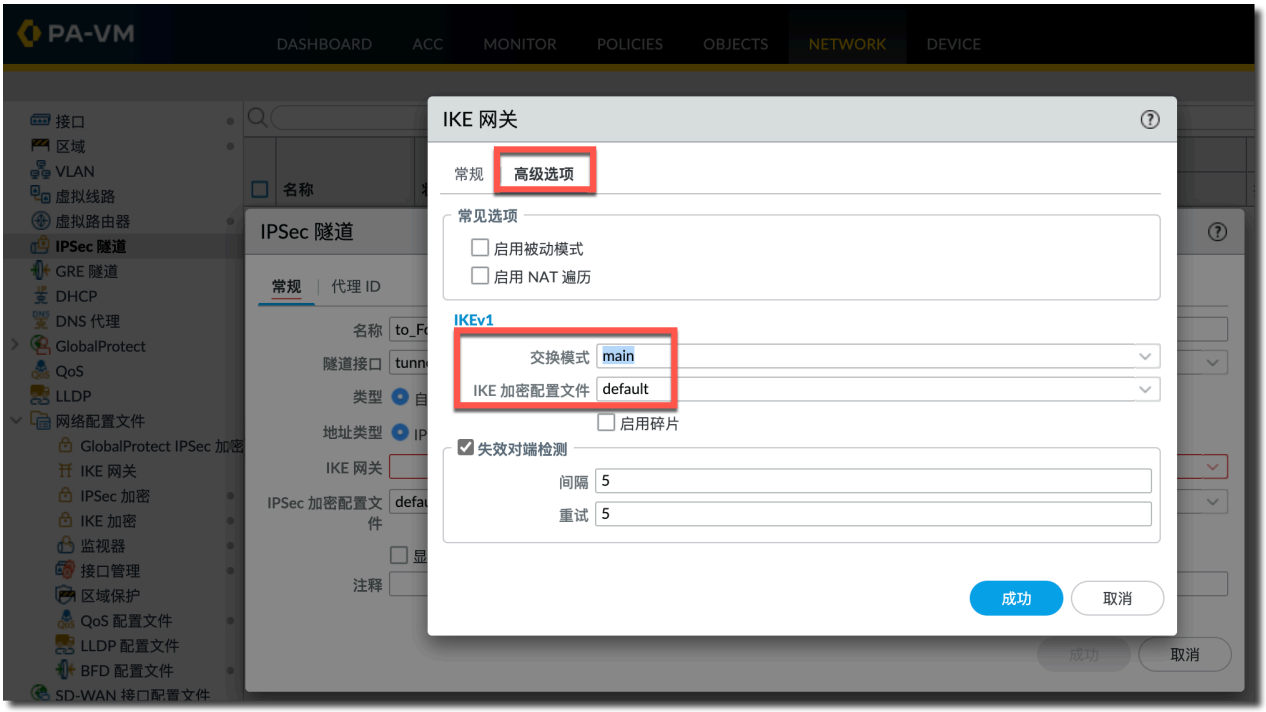

```
network {
  interface {
    tunnel {
      units {
        tunnel.1;
      }
    }
  }
  virtual-router {
    default {
      interface [ ethernet1/1 ethernet1/2 tunnel tunnel.1];
    }
  }
}

zone {
  Untrust {
    network {
      layer3 [ ethernet1/1 tunnel tunnel.1];
    }
  }
}
```

4. 在 IPSec 隧道页面点击添加按钮，输入隧道名称，隧道接口选择上步创建的“tunnel.1”。在“IKE 网关”中选择“新 IKE 网关”。

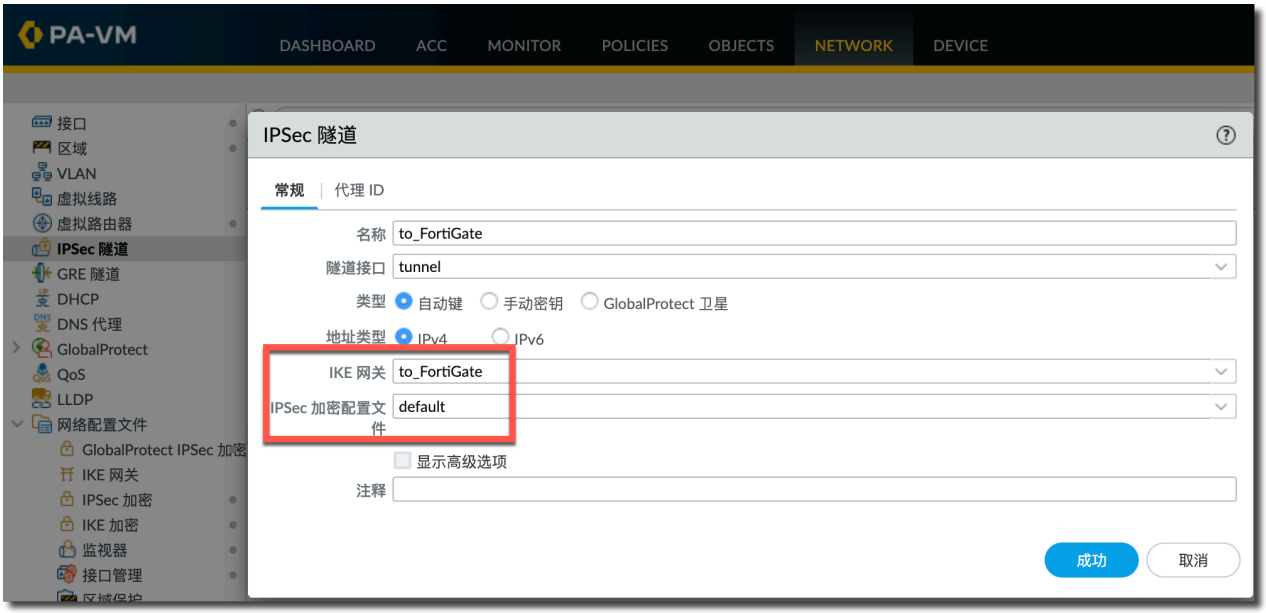


5. 在弹出的“IKE 网关”页面中，填写名称，版本选择 IKEv1，接口选择自身用于建立 IPSec 的接口，对端 VPN 网关地址，预共享密钥。在“高级选项”中，配置交换模式为 main，可以修改默认的一阶段安全提议，注意要和 FortiGate 端保持一致，此例中 FortiGate 的安全算法已配置为和 Palo Alto 中“default”包含的算法一致，所以不再进行修改（自定义的一阶段安全提议可以到“NETWORK → 网络配置文件 → IKE 加密”中创建）。随后下发“IKE 网关”的配置。





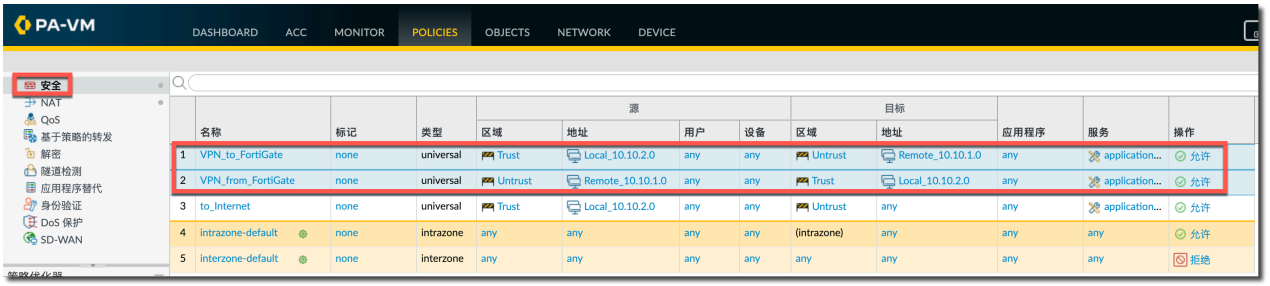
6. “IKE 网关”配置下发后，返回“IPSec 隧道”窗口，确认“IKE 网关”选择为上步创建的 IKE 网关，IPSec 加密配置文件（二阶段安全提议）选择“default”（注意要和 FortiGate 端保持一致，此例中 FortiGate 的安全算法已配置为和 Palo Alto 中“default”包含的算法一致，所以不再进行修改，自定义的二阶段安全提议可以到“NETWORK → 网络配置文件 → IPSec 加密”中创建）。“代理 ID”选项卡无需配置（默认为 0.0.0.0/0↔0.0.0.0/0）。




```
network {
  ike {
    crypto-profiles {
      ike-crypto-profiles {
        default {
          encryption [ aes-128-cbc 3des];
          hash sha1;
          dh-group group2;
          lifetime {
            hours 8;
          }
        }
      }
      ipsec-crypto-profiles {
        default {
          esp {
            encryption [ aes-128-cbc 3des];
            authentication sha1;
          }
          dh-group group2;
          lifetime {
            hours 1;
          }
        }
      }
    }
    gateway {
      to_FortiGate {
        authentication {
          pre-shared-key {
            key -AQ==pkKner19T1G/kibOr4kfy7Wymbg=sg4ErcrrUBiaoIjlTToVEA==;
          }
        }
        protocol {
          ikev1 {
            dpd {
              enable yes;
            }
            exchange-mode main;
          }
        }
        local-address {
          ip 202.103.23.2/24;
          interface ethernet1/1;
        }
        protocol-common {
          nat-traversal {
            enable no;
```

```
        }
        fragmentation {
          enable no;
        }
      }
      peer-address {
        ip 202.103.13.2;
      }
    }
  }
}
tunnel {
  ipsec {
    to_FortiGate {
      auto-key {
        ike-gateway {
          to_FortiGate;
        }
      }
      tunnel-monitor {
        enable no;
      }
      tunnel-interface tunnel;
    }
  }
}
```

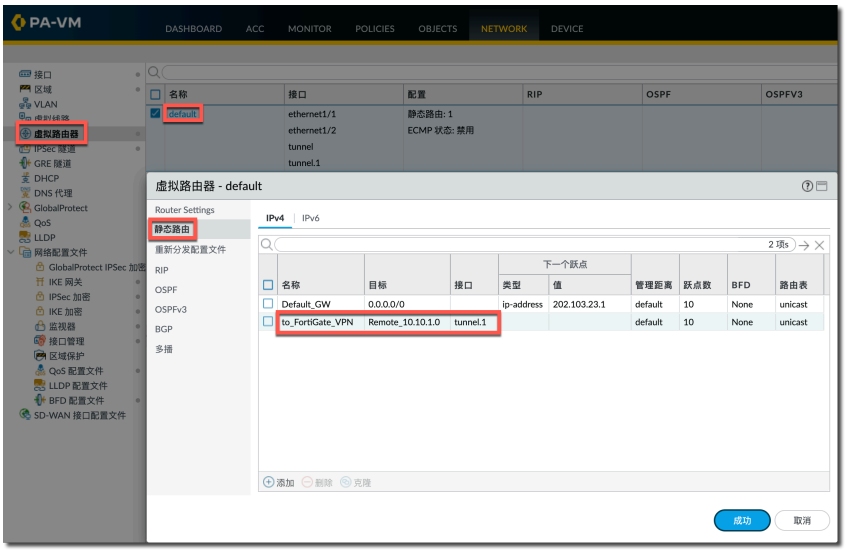
7. 配置 IPSec VPN 相关网段放通的安全策略，并将他们移动至上网策略之前。




```
rulebase {
  security {
    rules {
      VPN_to_FortiGate {
        to Untrust;
        from Trust;
        source Local_10.10.2.0;
        destination Remote_10.10.1.0;
        source-user any;
        category any;
        application any;
        service application-default;
        source-hip any;
        destination-hip any;
        action allow;
      }
      VPN_from_FortiGate {
        to Trust;
        from Untrust;
        source Remote_10.10.1.0;
        destination Local_10.10.2.0;
        source-user any;
        category any;
        application any;
        service application-default;
        source-hip any;
        destination-hip any;
        action allow;
      }
      to_Internet {
```

```
        to Untrust;
        from Trust;
        source Local_10.10.2.0;
        destination any;
        source-user any;
        category any;
        application any;
        service application-default;
        source-hip any;
        destination-hip any;
        action allow;
      }
    }
  }
}
```

8. 添加去往 FortiGate 内网 VPN 网段的静态路由，出接口为 tunnel.1 接口，下一跳配置为“None”即可。

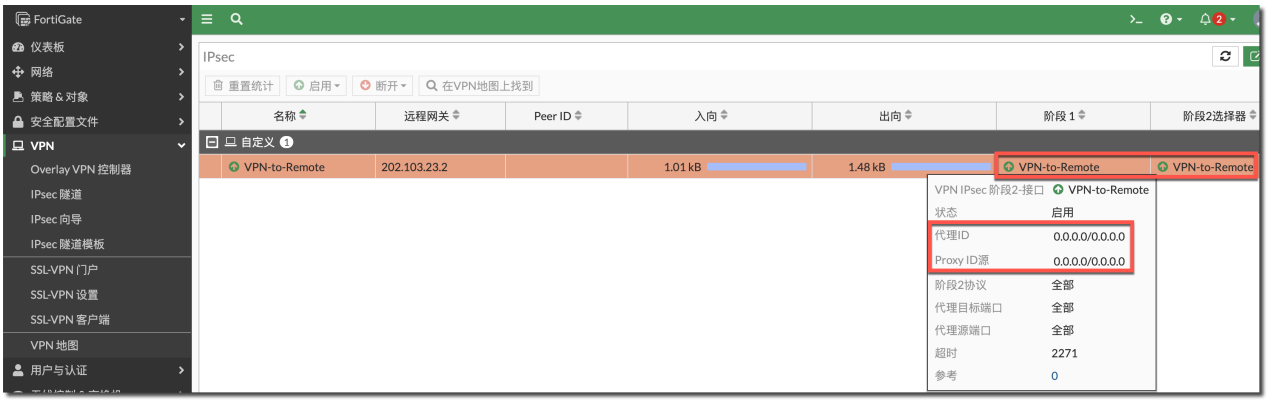



```
network {
  virtual-router {
    default {
      routing-table {
        ip {
          static-route {
            to_FortiGate_VPN {
              path-monitor {
                enable no;
                failure-condition any;
                hold-time 2;
              }
              interface tunnel.1;
              metric 10;
              destination Remote_10.10.1.0;
              route-table {
                unicast;
              }
            }
          }
        }
      }
      interface [ ethernet1/1 ethernet1/2 tunnel tunnel.1];
    }
  }
}
```

9. 最后，点击右上角的 commit 按钮提交所有配置。

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。




```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 1403s ago
IKE SA: created 1/1  established 1/1  time 120/120/120 ms
IPsec SA: created 1/3  established 1/1  time 140/140/140 ms
  id/spi: 0 d73a6ee8d0baf991/db995f50650cc6b5
  direction: initiator
  status: established 1403-1403s ago = 120ms
  proposal: aes128-sha1
  key: fd7f588b849be1d4-d63993f074d6b8f8
  lifetime/rekey: 86400/84697
  DPD sent/recv: 0000003e/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=3 202.103.13.2:0→202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=on weig1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overla0
proxyid_num=1 child_num=0 refcnt=4 ilast=2 olast=2 ad=/0
stat: rxp=12 txp=17 rxb=1008 txb=1484
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=63
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Remote proto=0 sa=1 ref=2 serial=2 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=2205/0B replaywin=2048
       seqno=12 esn=0 replaywin_lastseq=0000000c qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3300/3600
  dec: spi=35f11fb3 esp=aes key=16 2aba5fd6c7959ed2ee4dbd4653f2e42f
       ah=sha1 key=20 78b303176925e02007e758e1b4f3b2a43fcaab7e
  enc: spi=b5d1ae56 esp=aes key=16 a5d0f8ace26a19dd6c5e1089c0b4f755
       ah=sha1 key=20 c99af8e83adcd0b435fca71d0b25302ca84cc4f1
  dec:pkts/bytes=24/2016, enc:pkts/bytes=34/4132
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=3 dec_npuid=0 enc_npuid=0
run_tally=0
```

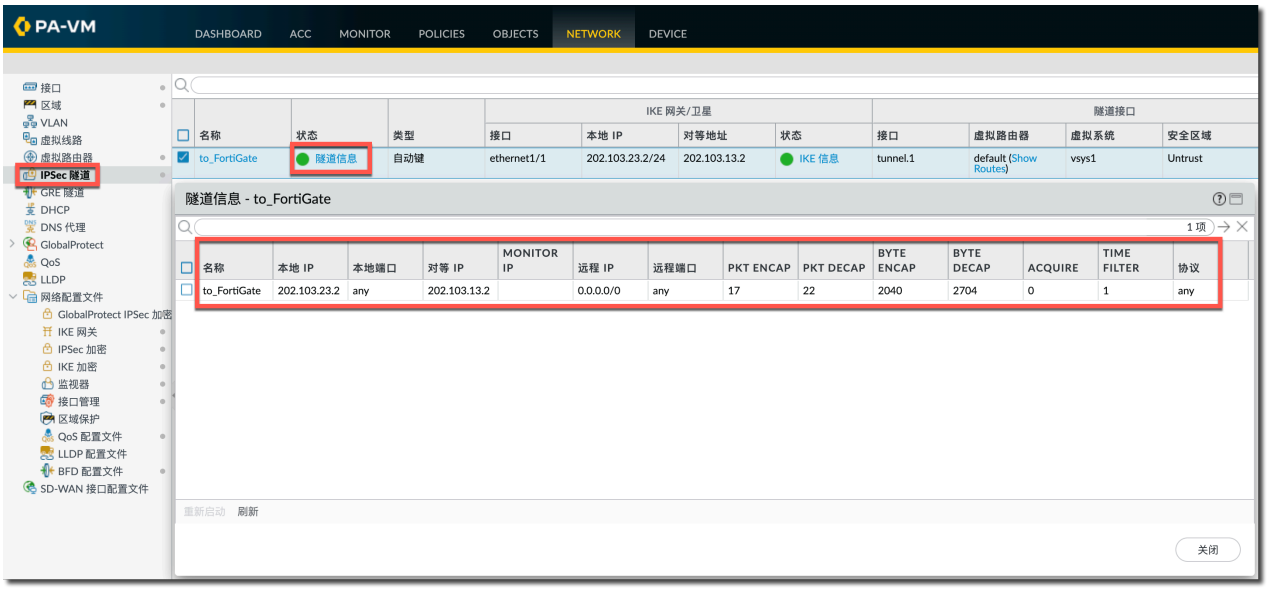
2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

3. FortiGate 侧 PC1 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6804/64

VPCS> ping 10.10.2.2

84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=4.250 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=2.122 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=2.039 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=1.824 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=2.243 ms
```

4. Palo Alto 防火墙查看 IPSec 连接状态。




```
admin@PA-VM> show vpn ike-sa
IKEv1 phase-1 SAs
GwID/client IP  Peer-Address          Gateway Name                                                  Role Mode
Algorithm             Established     Expiration      V  ST Xt Phase2
_____________  ___________          ___________                                                   ____ ____ _______
--             __________     __________      -  -- -- ______
1              202.103.13.2          to_FortiGate                                                  Resp Main PSK/
DH2/A128/SHA1    Apr.12 00:11:29 Apr.12 08:11:29 v1 13 1  1
Show IKEv1 IKE SA: Total 1 gateways found. 1 ike sa found.
IKEv1 phase-2 SAs
Gateway Name                                                  TnID     Tunnel                GwID/IP          Role
Algorithm          SPI(in)  SPI(out) MsgID    ST Xt
___________                                                   ____     ______                _______          ____ --
_______          _______  ________ _____    -- --
to_FortiGate                                                  1        to_FortiGate          1                Resp
```

```
ESP/ DH2/tunl/SHA1 B5D1AE56 35F11FB3 689E6AC0 9  1
Show IKEv1 phase2 SA: Total 1 gateways found. 1 ike sa found.

admin@PA-VM> show vpn ipsec-sa
GwID/client IP  TnID   Peer-Address           Tunnel(Gateway)
Algorithm          SPI(in)  SPI(out) life(Sec/KB)             remain-time(Sec)
———————————   ———   ———————————           ———————————————
———————————          ———————  ———————— ———————————           ———————————————
1               1      202.103.13.2           to_FortiGate(to_FortiGate)
ESP/A128/SHA1      B5D1AE56 35F11FB3 3600/Unlimited           2152
Show IPSec SA: Total 1 tunnels found. 1 ipsec sa found.

admin@PA-VM> show vpn flow
total tunnels configured:                                  1
filter - type IPSec, state any
total IPSec tunnel configured:                             1
total IPSec tunnel shown:                                  1
id    name                                                         state   monitor local-ip
peer-ip                      tunnel-i/f                                   ————    ———————  ————————                        ———
--    ———————————————                                                    ————    ———————  ————————                        ———
———                        ——————————
1     to_FortiGate                                                 active  off     202.103.23.2
202.103.13.2                 tunnel.1
```

5. 查看 Palo Alto 防火墙的路由表，可以看到包含了手动配置的到 FortiGate 内网的路由。

```
admin@PA-VM> show routing route

flags: A:active, ?:loose, C:connect, H:host, S:static, ~:internal, R:rip, O:ospf, B:bgp,
       Oi:ospf intra-area, Oo:ospf inter-area, O1:ospf ext-type-1, O2:ospf ext-type-2, E:ecmp, M:multicast
VIRTUAL ROUTER: default (id 1)
  ==========
destination                              nexthop                              metric flags      age   interface
next-AS
0.0.0.0/0                                202.103.23.1                         10     A S              ethernet1/1
10.10.1.0/24                             0.0.0.0                              10     A S              tunnel.1
10.10.2.0/24                             10.10.2.1                            0      A C              ethernet1/2
10.10.2.1/32                             0.0.0.0                              0      A H
202.103.23.0/24                          202.103.23.2                         0      A C              ethernet1/1
202.103.23.2/32                          0.0.0.0                              0      A H
total routes shown: 6
```

6. Palo Alto 防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=2.099 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=2.617 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=2.074 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.024 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=1.944 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-12 15:38:12.130447 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:12.130504 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:12.130732 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:12.130746 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:13.134398 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:13.134471 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:13.134728 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

```
2023-04-12 15:38:13.134745 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:14.137358 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:14.137391 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:14.137592 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:14.137600 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:15.140135 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:15.140174 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:15.140321 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:15.140334 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:16.143326 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:16.143362 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-12 15:38:16.143547 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-12 15:38:16.143560 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：
   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
     FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
     FortiGate (BJ-OSPF-TO-SH) # end
     ```

2. Palo Alto 防火墙 IPsec VPN Debug：

```
debug ike global on debug
less mp-log ikemgr.log
debug ike pcap on
view-pcap no-dns-lookup yes no-port-lookup yes debug-pcap ikemgr.pcap
debug ike pcap off
```

与 Check Point 防火墙建立 IPSec VPN

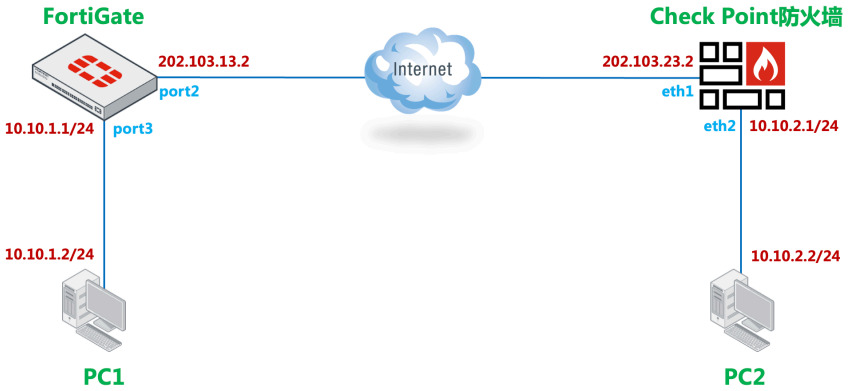
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与 Check Point 防火墙进行 IPsec VPN（IKE v1）对接。

Check Point 防火墙版本：

```
gw-000100> ver
Product version Check Point Gaia R81.10
OS build 335
OS kernel version 3.10.0-957.21.3cpx86_64
OS edition 64-bit
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Check Point 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

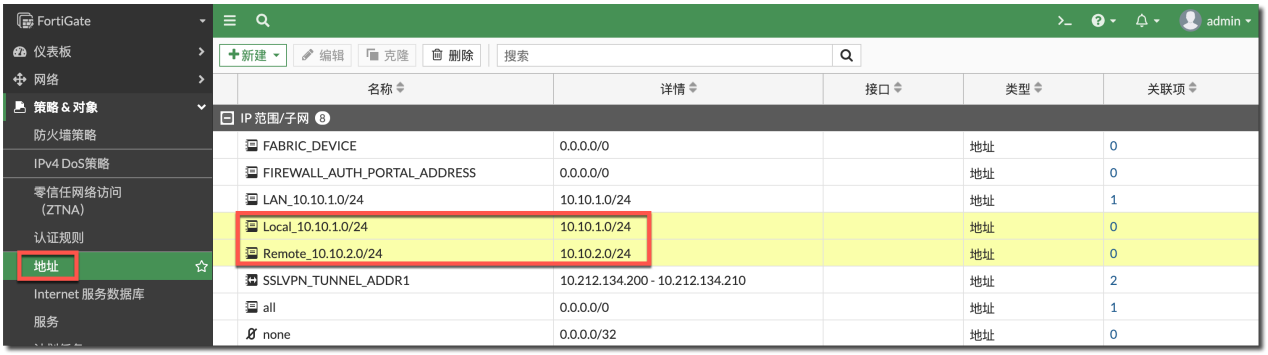
重要

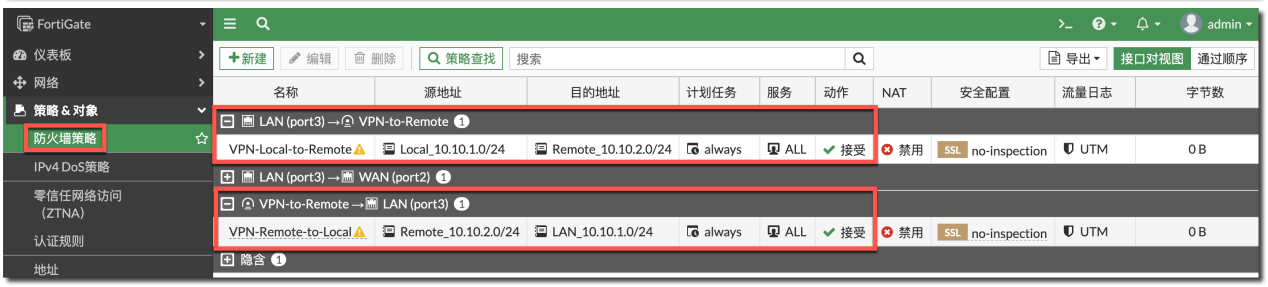
由于 Check Point 的 IPSec 配置需要配置明细的感兴趣流，FortiGate 与 Check Point 防火墙对接 IPSec 需要对应使用明细的感兴趣流。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
10.10.1.0/24
10.10.2.0/24
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
10.10.1.0/24
对端地址
子网
10.10.2.0/24
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 14 5 2
        set auto-negotiate enable
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。






```
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
        set srcintf "VPN-to-Remote"
```

```
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

重要

相关 VPN 的路由配置中“黑洞路由的意义：

你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?

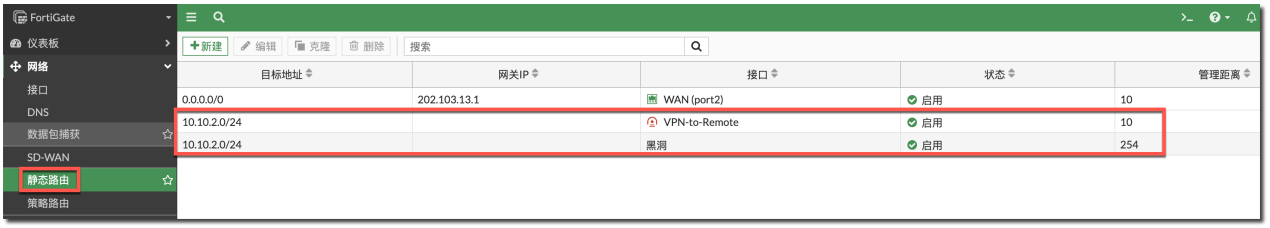
VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。

解决方法：

- 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
- 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
- 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。

```
FortiGate # config system global
FortiGate (global) # set snat-route-change enable
FortiGate (global) # end
```

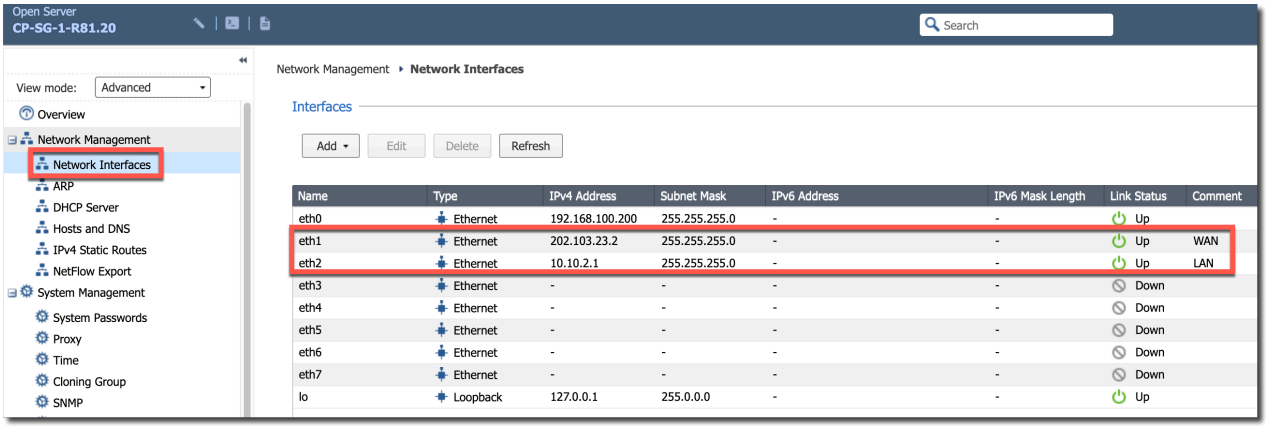
三种方式任意选择一种即可。推荐使用黑洞路由方式。

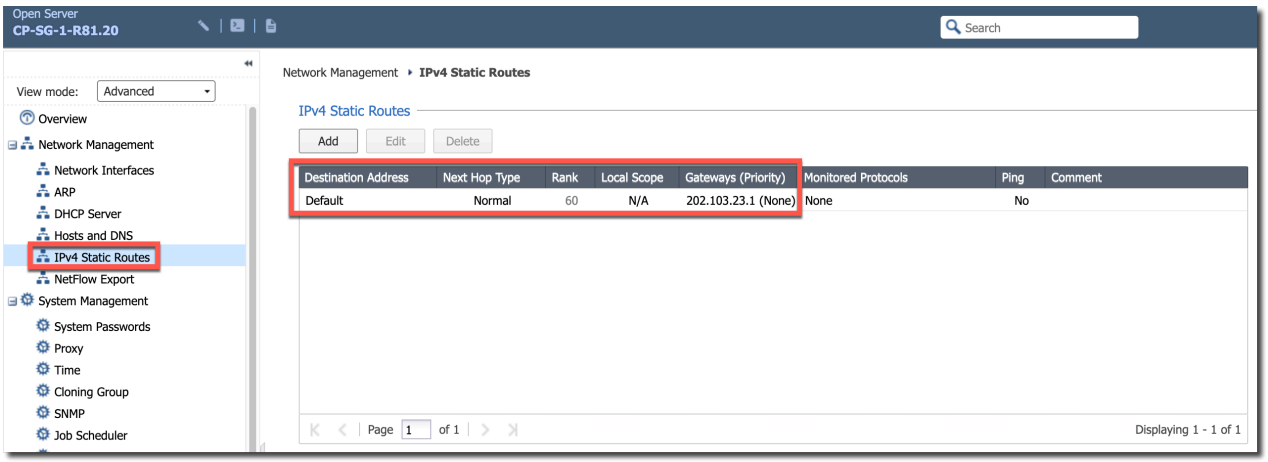



```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Check Point 防火墙

1. 进入 Check Point 防火墙的 Web 管理页面，配置基本 IP 与路由。

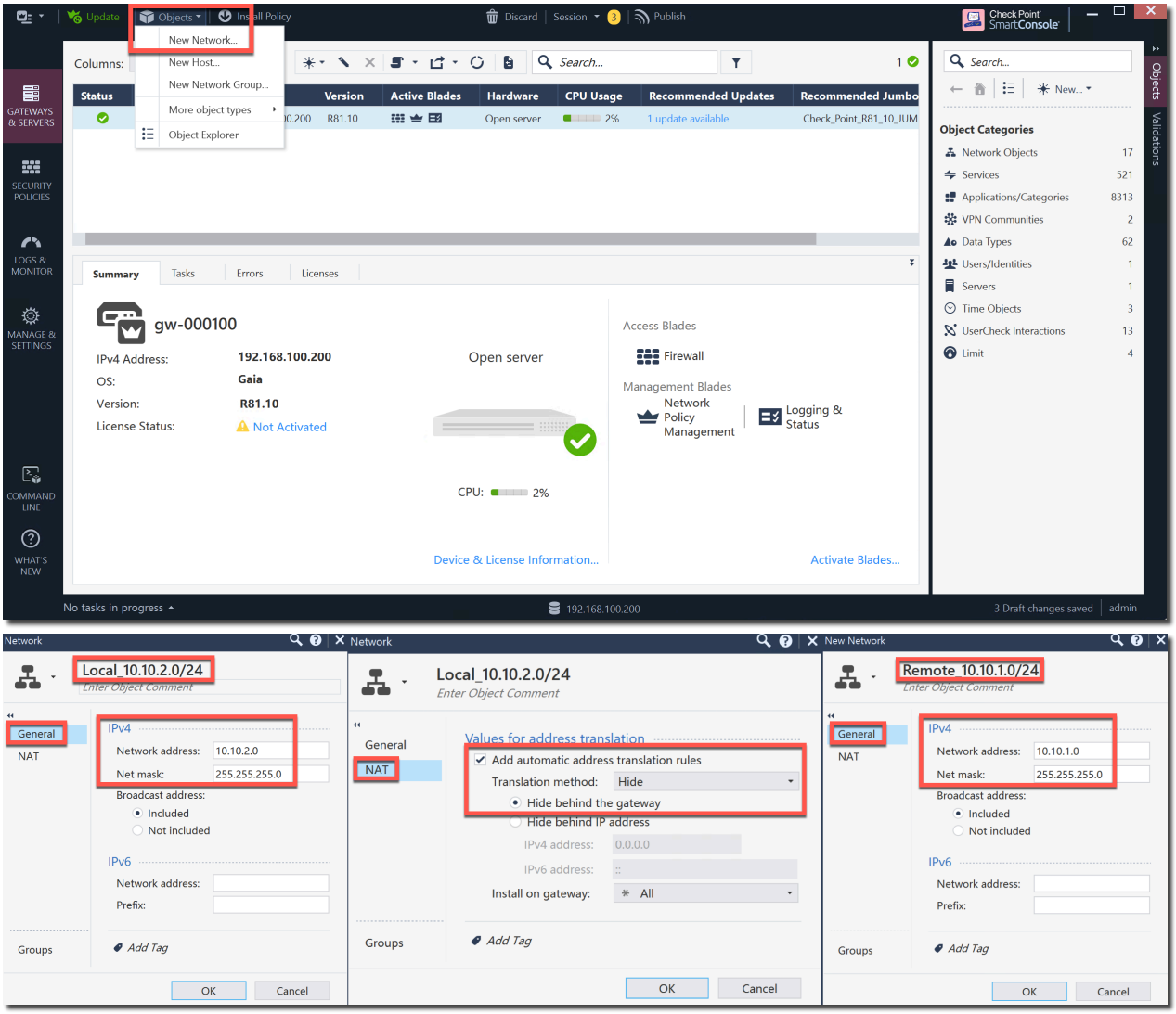




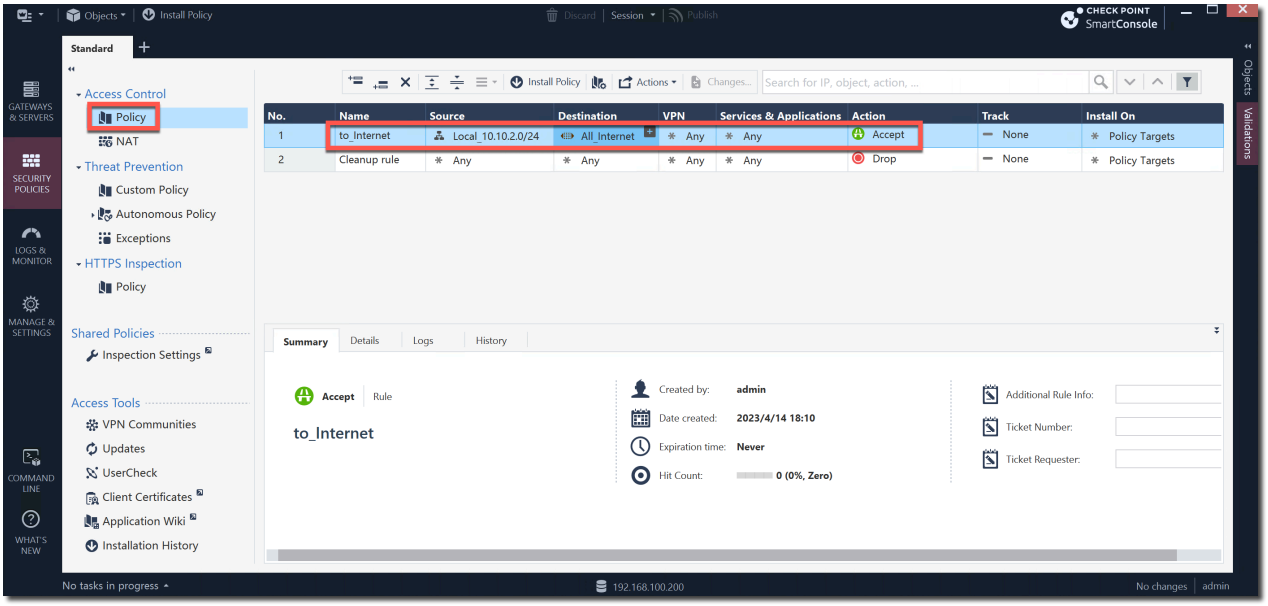

```
set interface eth1 comments "WAN"
set interface eth1 link-speed 1000M/full
set interface eth1 state on
set interface eth1 auto-negotiation on
set interface eth1 mtu 1500
set interface eth1 ipv4-address 202.103.23.2 mask-length 24
set interface eth2 comments "LAN"
set interface eth2 link-speed 1000M/full
set interface eth2 state on
set interface eth2 auto-negotiation on
set interface eth2 mtu 1500
set interface eth2 ipv4-address 10.10.2.1 mask-length 24

set static-route default nexthop gateway address 202.103.23.1 on
```

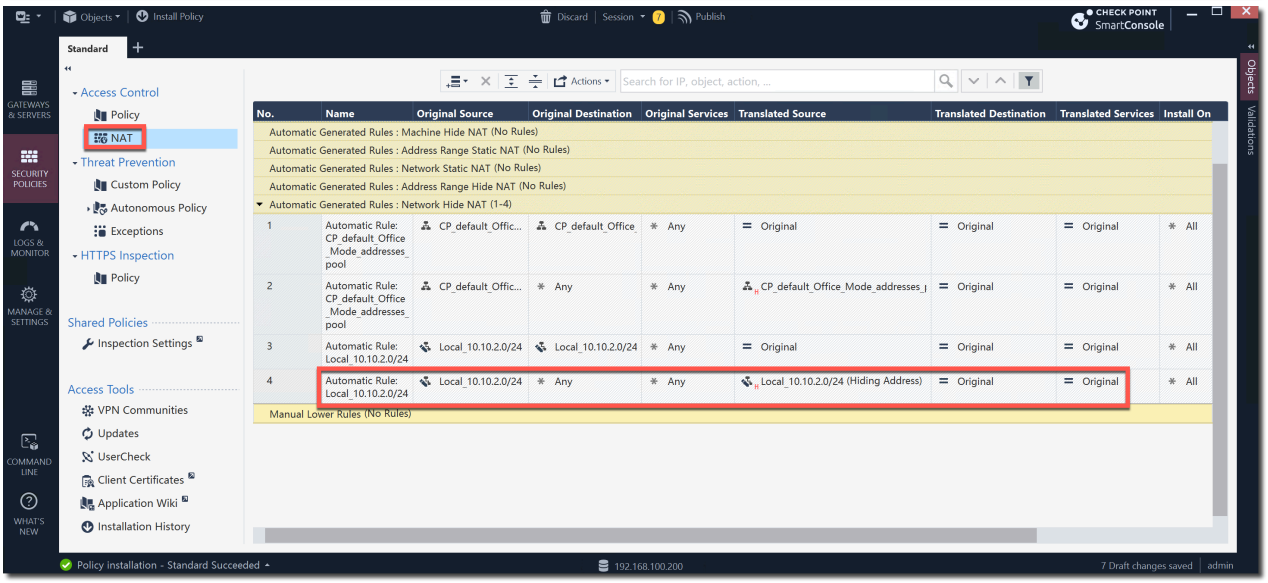
2. 在 Check Point 防火墙的 CLI 下使用“save config”命令保存配置。

3. 使用 SmartConsole 连接 Check Point 防火墙，配置地址对象（在本地网段的地址对象中开启 NAT，并选择 Hide 模式的 NAT，地址对象创建完成后会自动创建对应的 SNAT 策略），创建上网的安全策略。

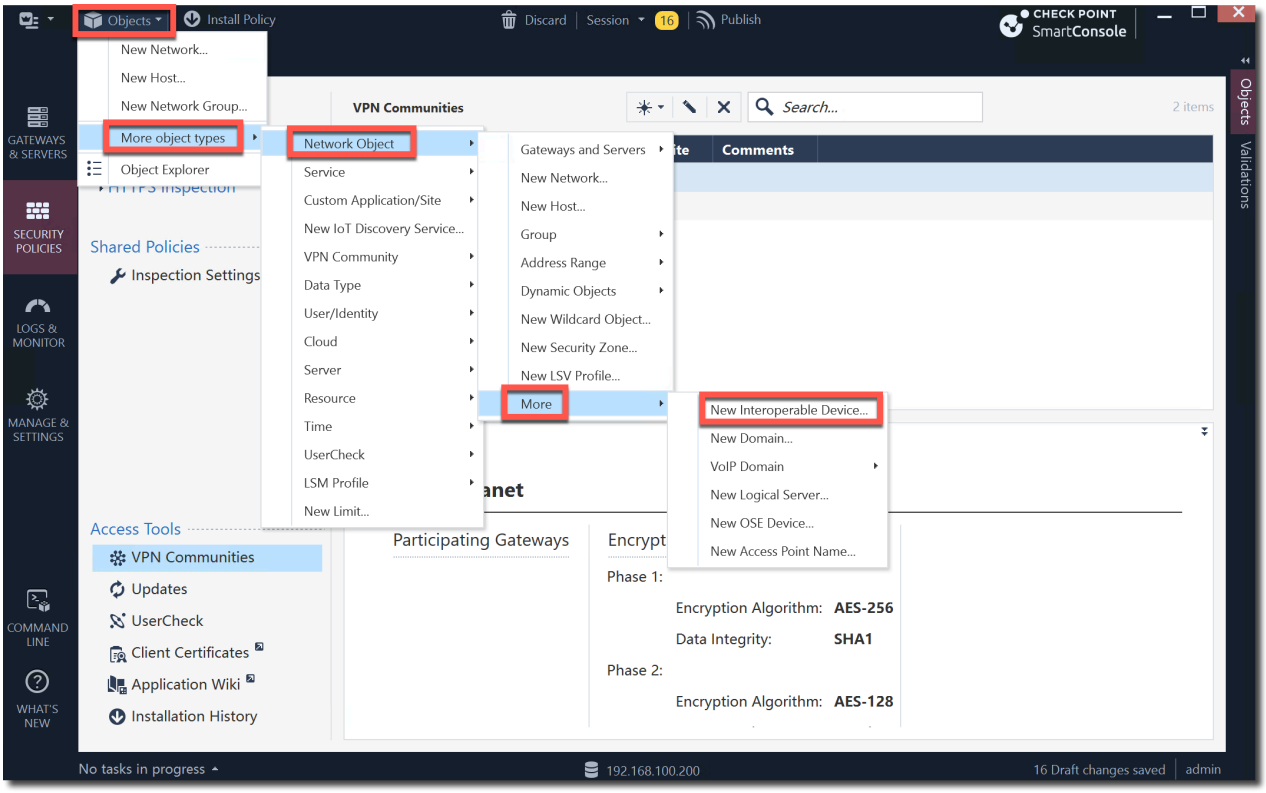

Update
Objects
Install Policy
New Network...
New Host...
New Network Group...
More object types
Object Explorer
Discard
Session
Publish
Check Point SmartConsole
Columns:
Search...
Status
Version
Active Blades
Hardware
CPU Usage
Recommended Updates
Recommended Jumbo
R81.10
Open server
2%
1 update available
Check_Point_R81_10_JUM
GATEWAYS & SERVERS
SECURITY POLICIES
LOGS & MONITOR
MANAGE & SETTINGS
COMMAND LINE
WHAT'S NEW
Summary
Tasks
Errors
Licenses
gw-000100
IPv4 Address: 192.168.100.200
OS: Gaia
Version: R81.10
License Status: Not Activated
Open server
CPU: 2%
Device & License Information...
Access Blades
Firewall
Management Blades
Network Policy Management
Logging & Status
Activate Blades...
Object Categories
Network Objects 17
Services 521
Applications/Categories 8313
VPN Communities 2
Data Types 62
Users/Identities 1
Servers 1
Time Objects 3
UserCheck Interactions 13
Limit 4
Objects
Validations
No tasks in progress
192.168.100.200
3 Draft changes saved
admin

Network
Local_10.10.2.0/24
Enter Object Comment
General
NAT
IPv4
Network address: 10.10.2.0
Net mask: 255.255.255.0
Broadcast address:
Included
Not included
IPv6
Network address:
Prefix:
Groups
Add Tag
OK
Cancel
Network
Local_10.10.2.0/24
Enter Object Comment
General
NAT
Values for address translation
Add automatic address translation rules
Translation method: Hide
Hide behind the gateway
Hide behind IP address
IPv4 address: 0.0.0.0
IPv6 address: ::
Install on gateway: All
Groups
Add Tag
OK
Cancel
New Network
Remote_10.10.1.0/24
Enter Object Comment
General
NAT
IPv4
Network address: 10.10.1.0
Net mask: 255.255.255.0
Broadcast address:
Included
Not included
IPv6
Network address:
Prefix:
Groups
Add Tag
OK
Cancel

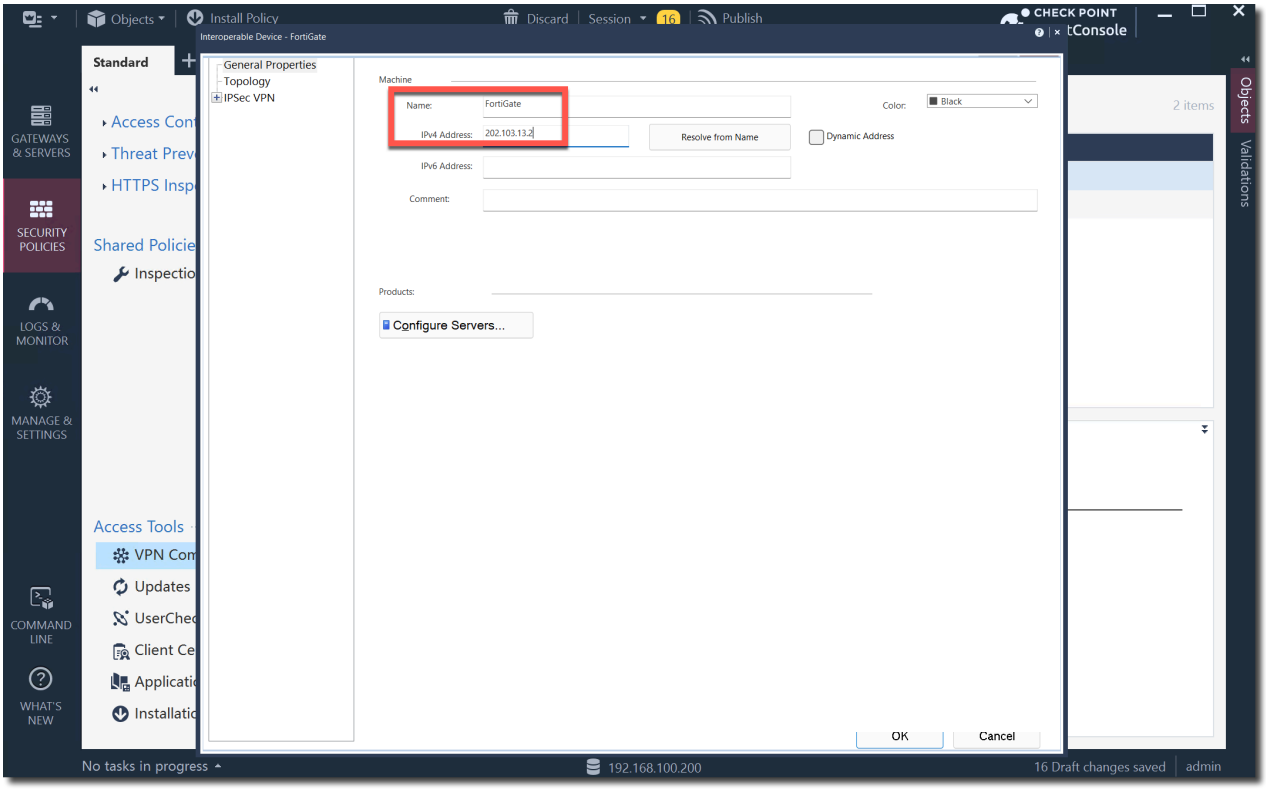




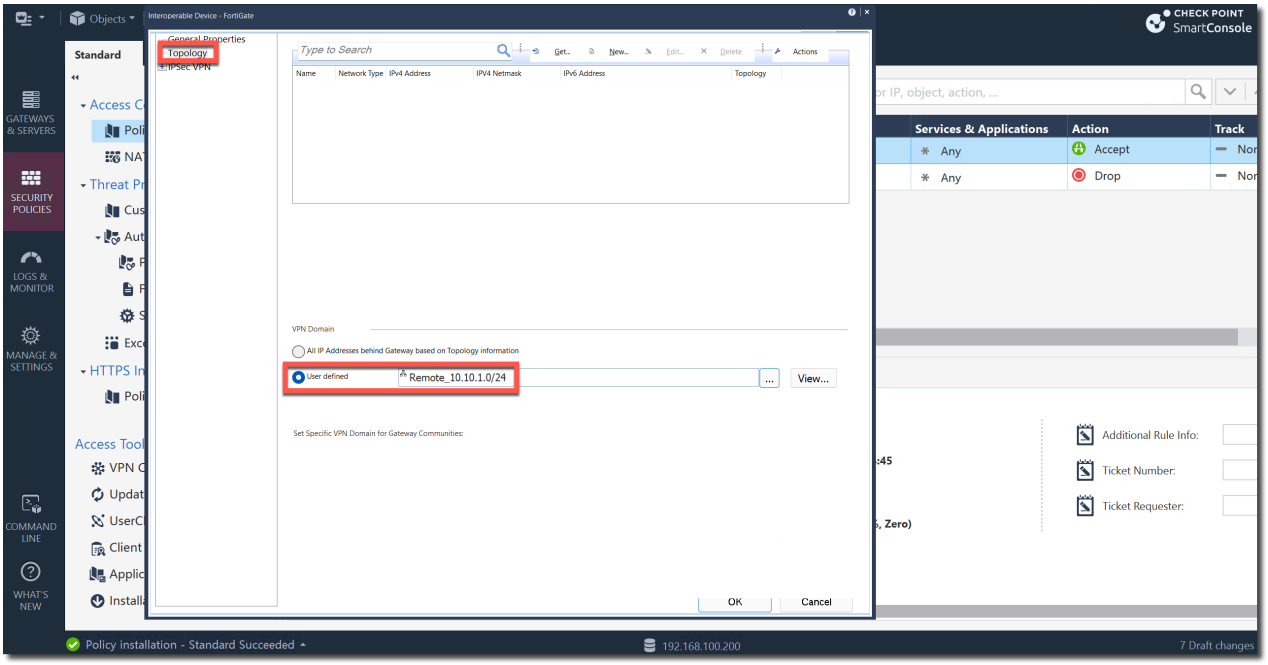
4. 创建“Interoperable Device”对象（IPSec 远程网关）。



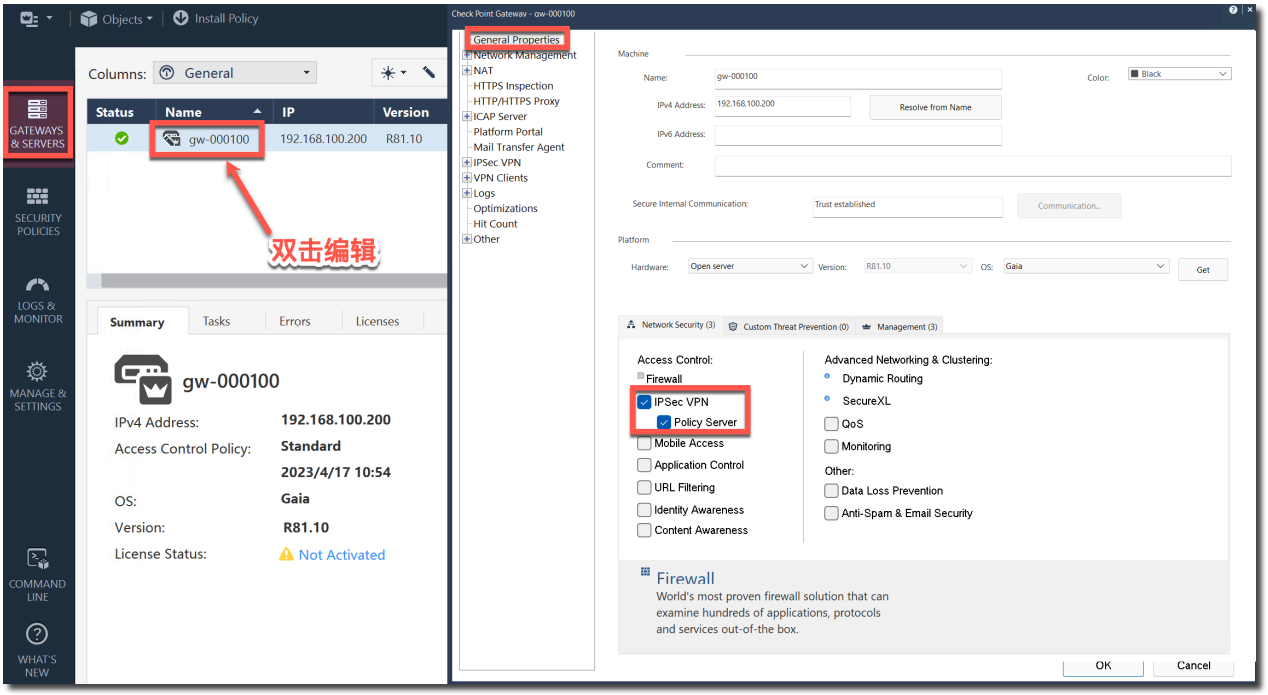
5. 在弹出的“Interoperable Device”的“General Properties”中的“Machine”处，填入 FortiGate 的 VPN 网关地址。



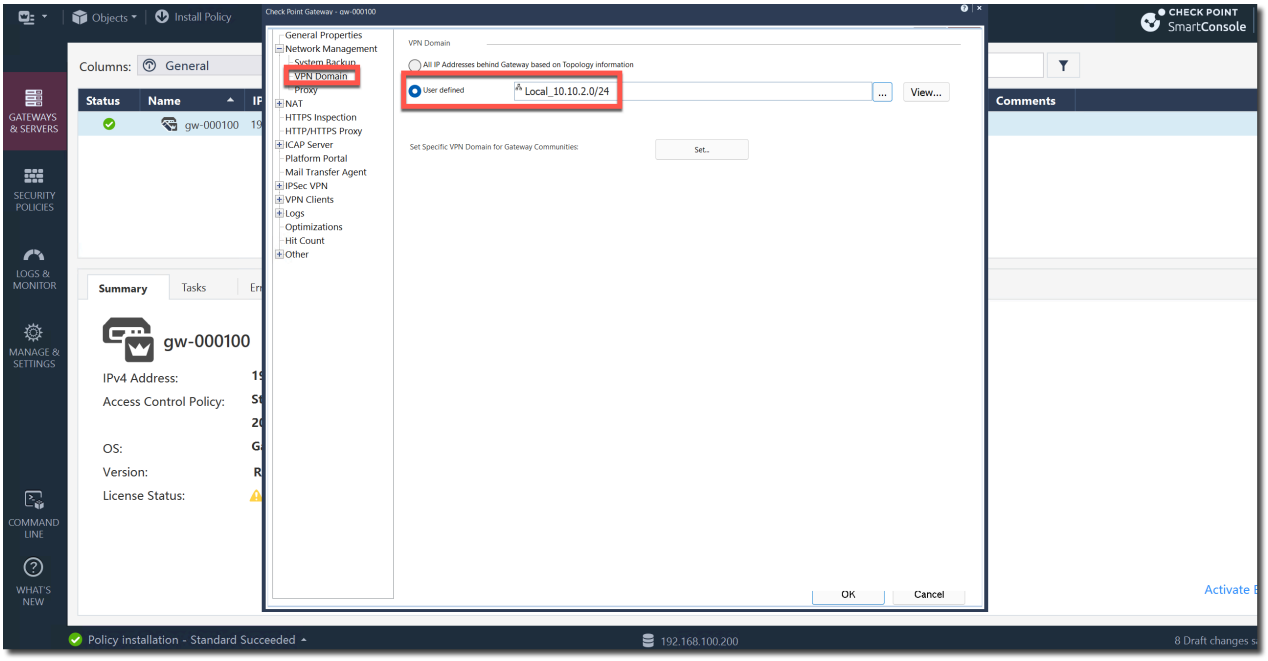
6. 在“Topology”中配置“VPN Domain”为“User defined”，并选择对端（FortiGate 侧）VPN 感兴趣流的地址对象，点击 OK 完成创建。



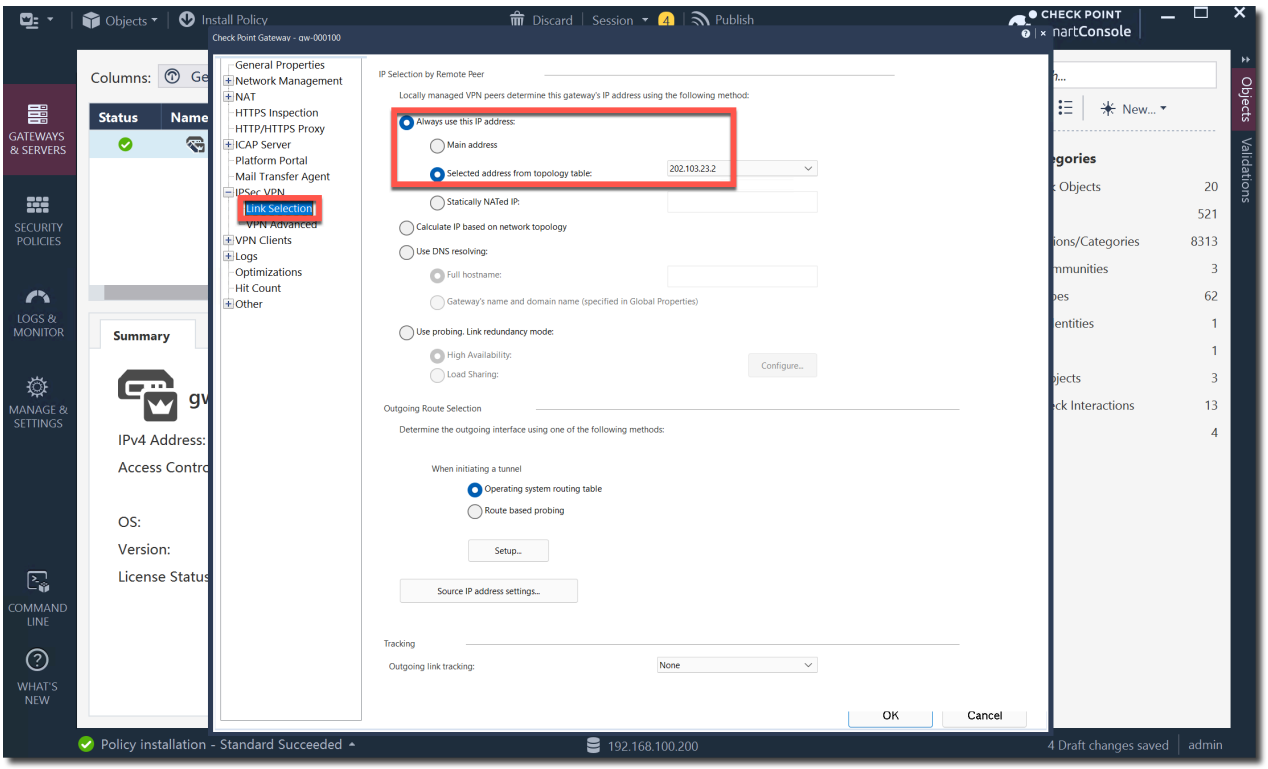
7. 进入“GATEWAYS & SERVERS”选项卡，双击管理的防火墙设备，在弹出的页面中的“General Properties”中开启 IPSec VPN 服务。



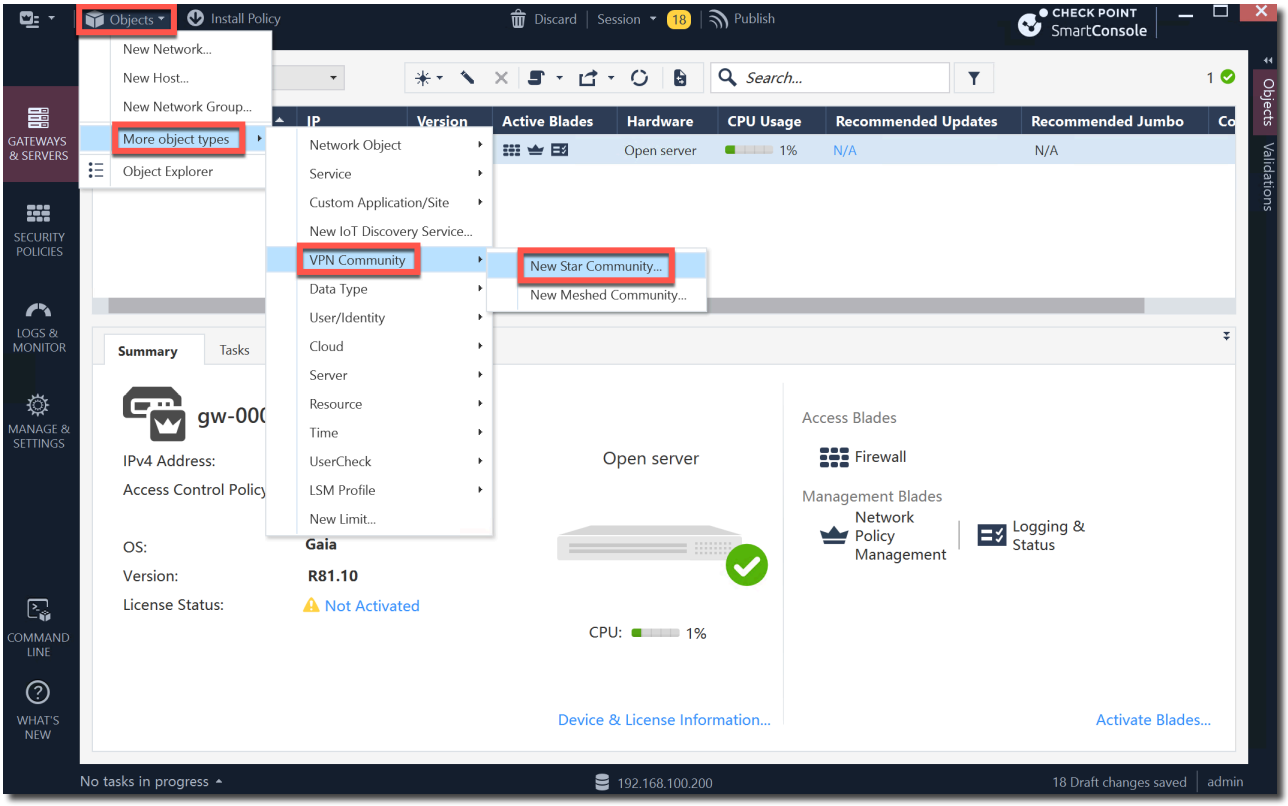
8. 在“Network Domain → VPN Domain”中，配置“User defined”为本地 VPN 保护网段的地址对象。



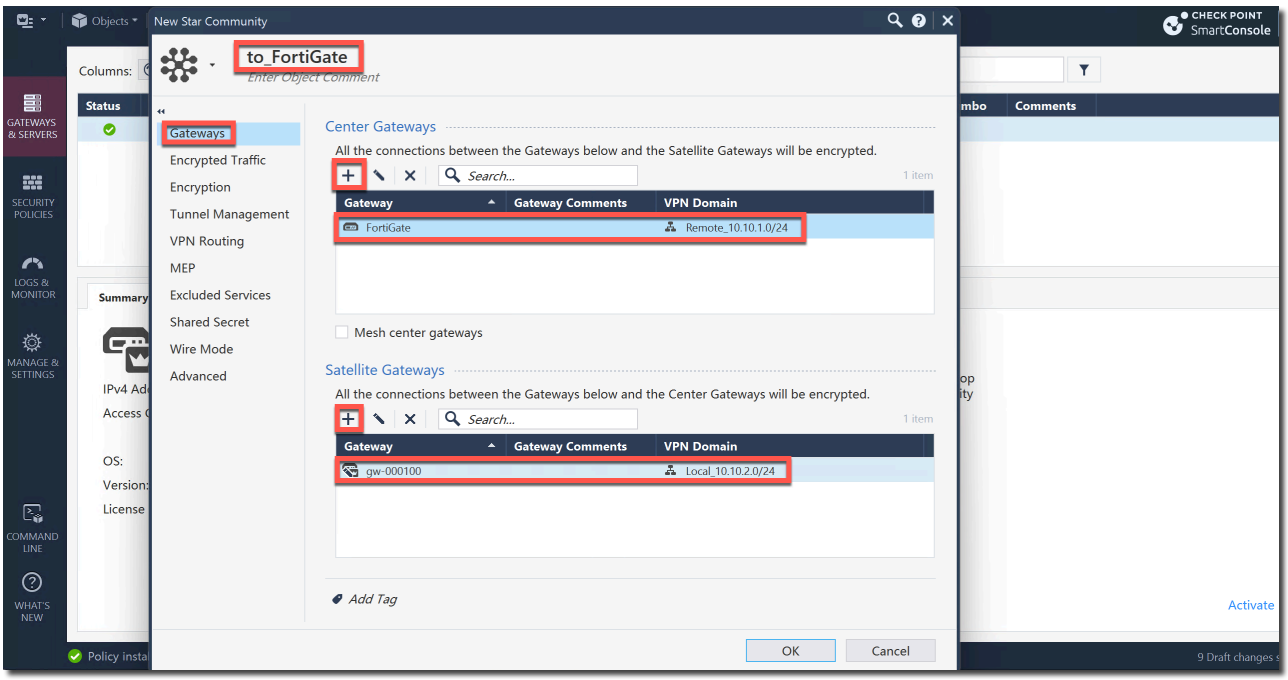
9. 在“IPSec VPN → Link Selection”中，配置“Always use this IP address”为“Selected address from topology table”，并选择 eth1（本地 VPN 网关）的接口 IP，点击 OK 下发配置。



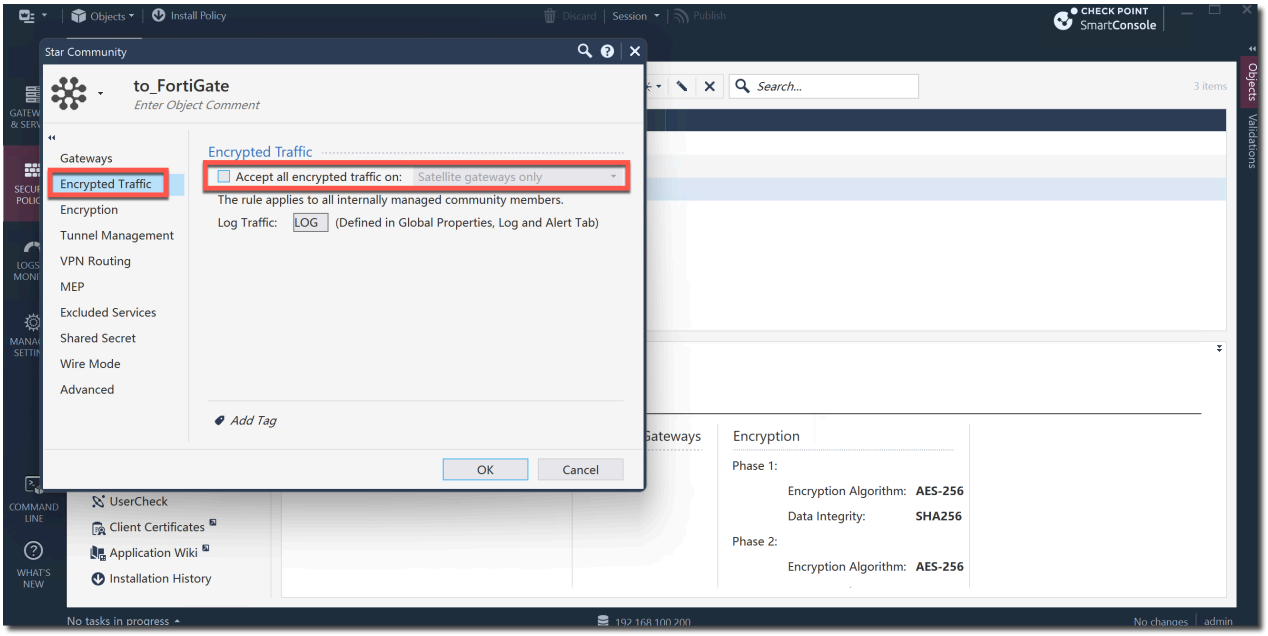
0. 创建“VPNCommunity”对象。



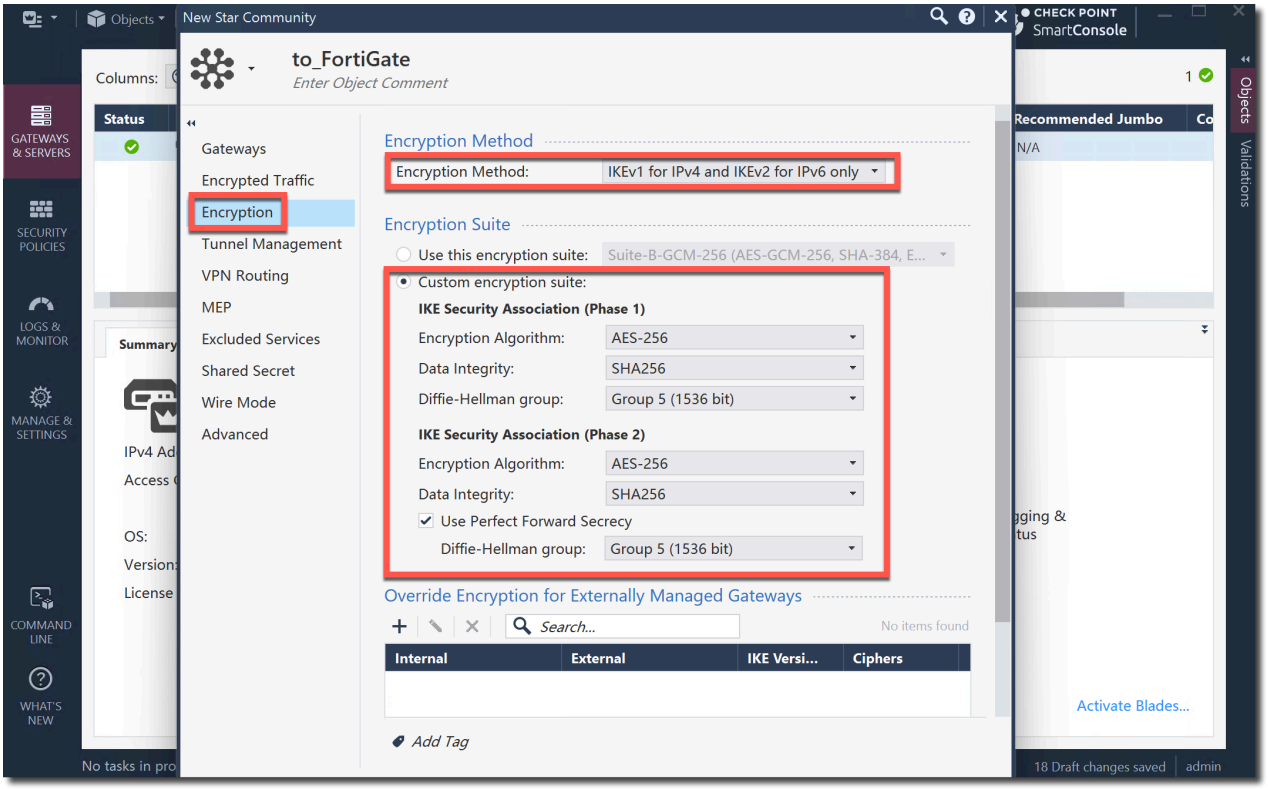
1. 在弹出的页面中，配置 Gateway，“Center Gateways”添加上步创建的“Interoperable Device”，“Satellite Gateways”添加本地 Gateway。



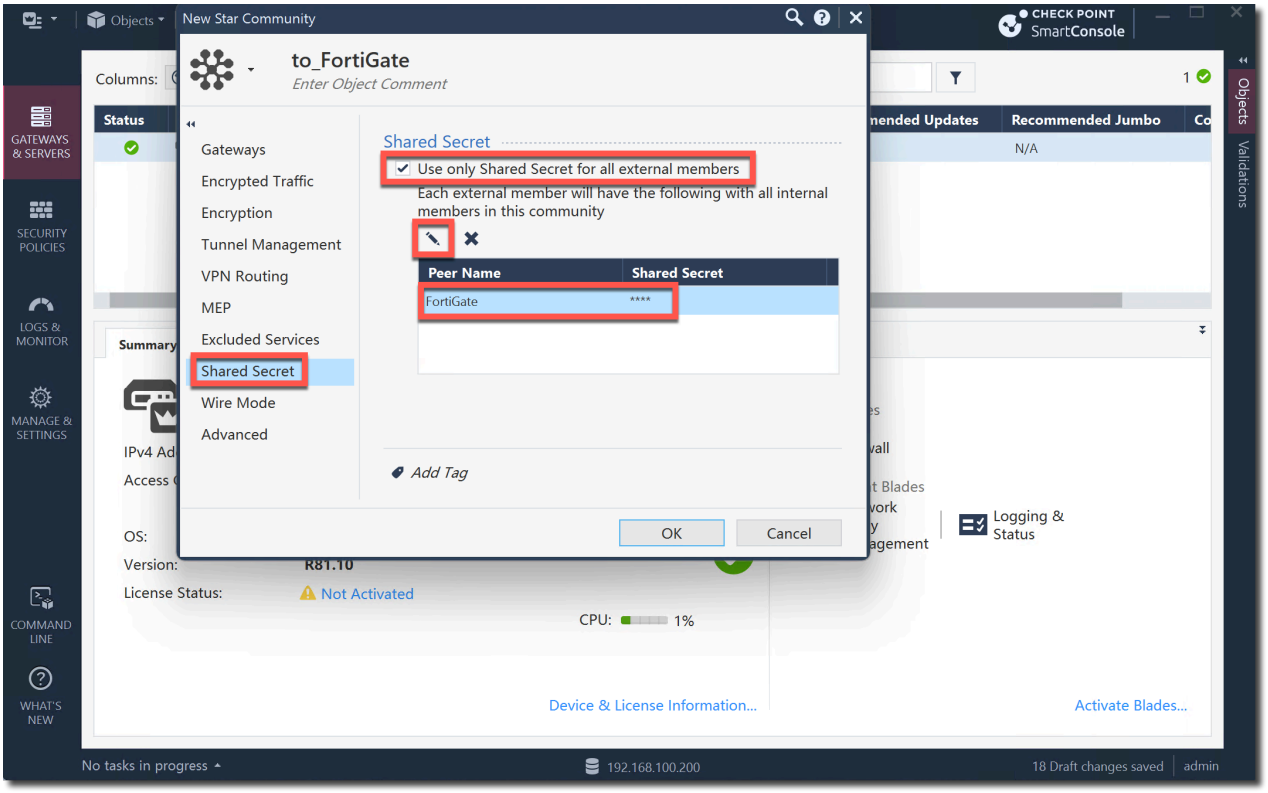
1. 在“Encrypted Traffic”中，不勾选“Accept all encrypted Traffic on”。



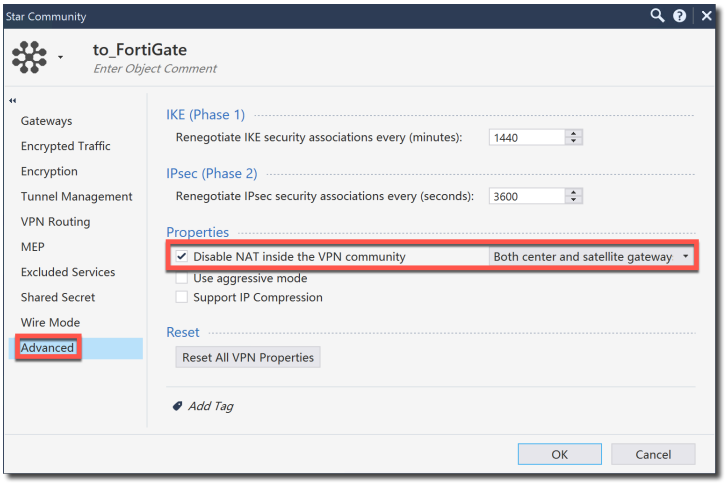
2. 在“Encryption”中配置一阶段和二阶段的安全提议，注意与 FortiGate 端保持一致。



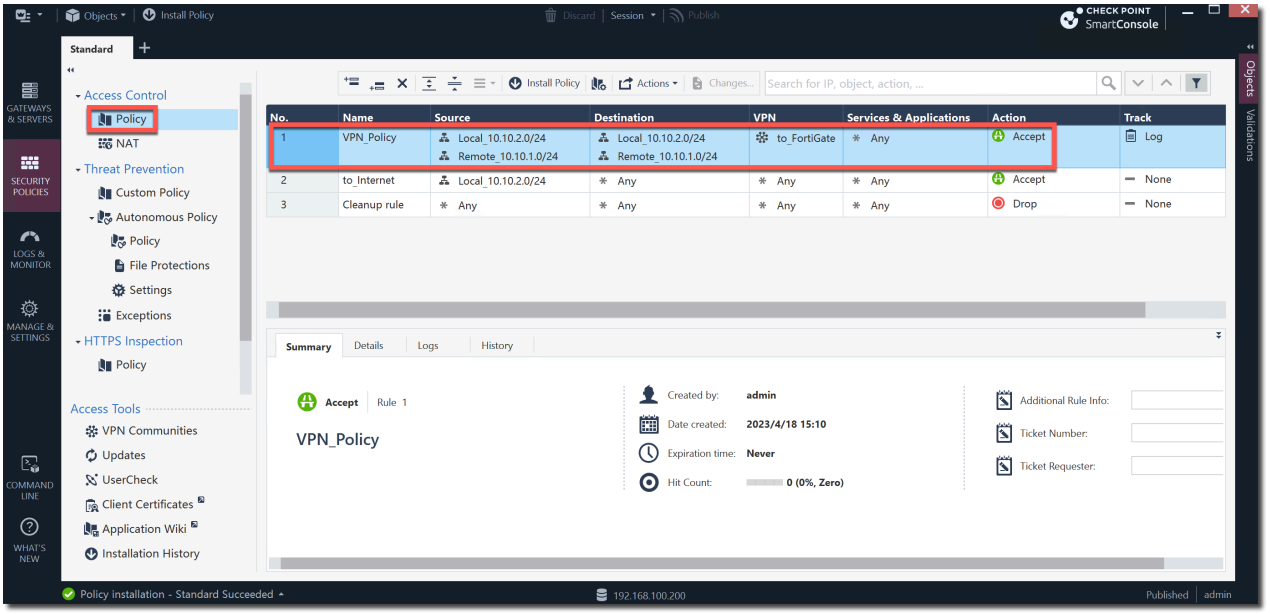
3. 在“Shared Secret”中开启“Use only Shared Secret for all external members”，编辑预共享密钥与 FortiGate 端一致。



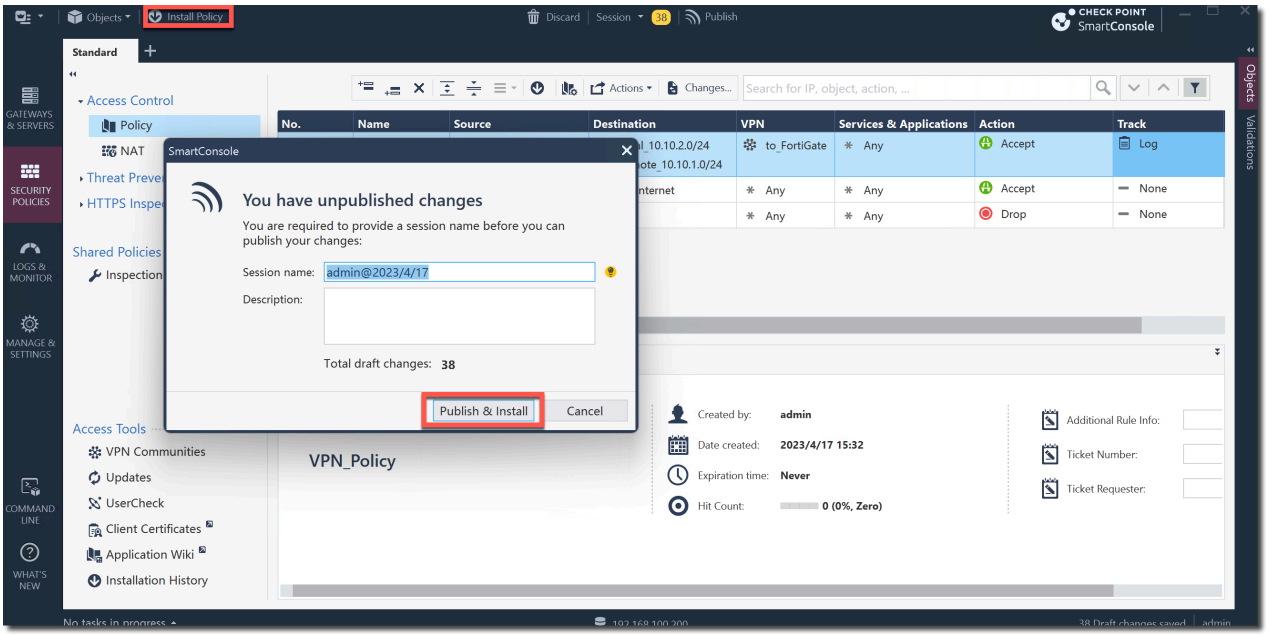
4. 由于 Check Point 的 IPSec 绑定在实际链路口，而不是虚拟 tunnel 接口，在出方向上，SNAT 动作优先于 IPSec 封装，VPN 流量经过 SNAT 后会匹配不上感兴趣流，所以需要配置 IPSec 流量不被 SNAT 处理。在“Advanced”选项卡中，勾选“Disable NAT inside the VPN Community”，开启后，进入 VPN 隧道的流量不会匹配 NAT 策略。点击 OK 下发配置。



5. 在上网策略之前配置 IPSec VPN 相关网段放通的安全策略。



6. 最后，点击“Install Policy”安装所有策略。



结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。



```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 127s ago
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 508 d0481e88ec264101/908a37e3aa3e43be
  direction: initiator
  status: established 127-127s ago = 0ms
  proposal: aes256-sha256
  key: 39c34495a79b54ca-cdd33b755c2150e9-2b31e3f2ace9c0a2-5b9307e2ee57ee63
  lifetime/rekey: 1400/1242
  DPD sent/recv: 0000006e/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=1 202.103.13.2:0→202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=on1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 o0
proxyid_num=1 child_num=0 refcnt=4 ilast=31 olast=31 ad=/0
stat: rxp=10 txp=10 rxb=840 txb=840
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=110
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Remote proto=0 sa=1 ref=10 serial=2 auto-negotiate
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.2.0-10.10.2.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=3166/0B replaywin=2048
       seqno=b esn=0 replaywin_lastseq=0000000a qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3297/3600
  dec: spi=daad030c esp=aes key=32 71ddf0ef357e7828acaffcbb4678daa04020591c56c2e6b3371643d4d5c0d1f5
       ah=sha256 key=32 cc7d15cad39ce576d9a1656e51c1b0e870aec41cf5d5e39c7ffcead1750363bd
  enc: spi=ccc51a59 esp=aes key=32 43d629dbc1e0bdb045772b7d5f42f6400a9ae5544be14b52312d7d7543bfb96b
       ah=sha256 key=32 5fdff7bce9562ae578488bbb78e9f050f91cfc137871c947056232114e017b93
  dec:pkts/bytes=20/1680, enc:pkts/bytes=20/2400
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=1 dec_npuid=0 enc_npuid=0
run_tally=0
```

2. 查看 FortiGate 的路由表。

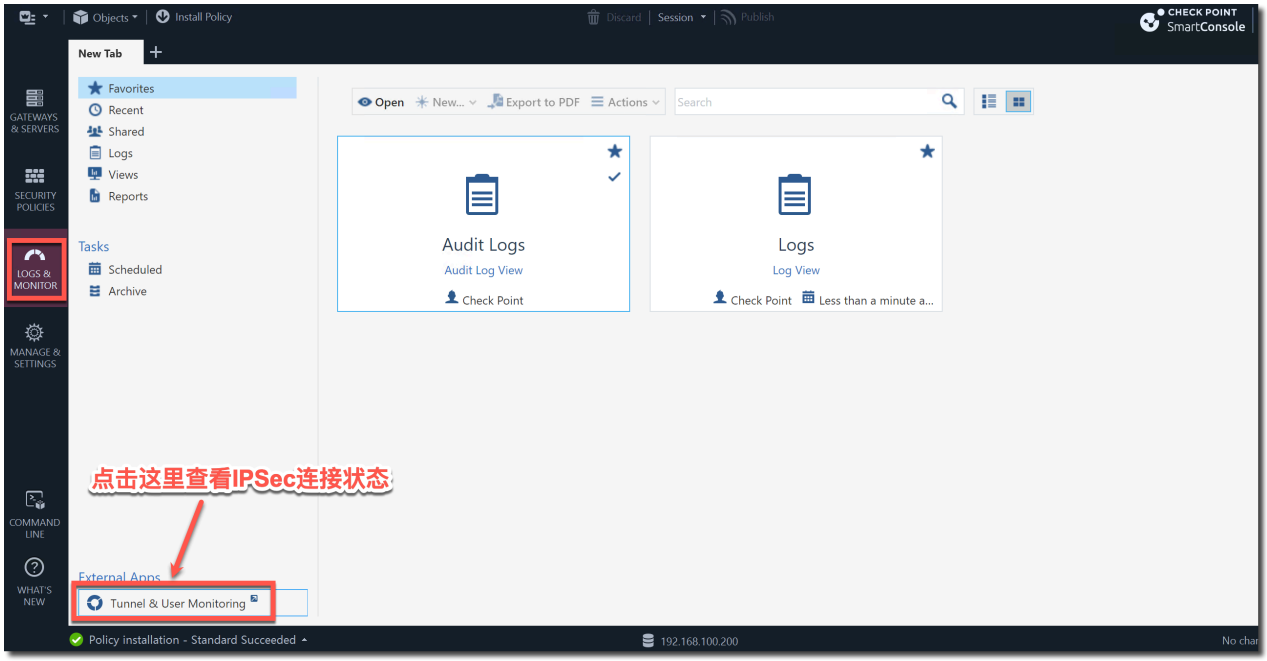
```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

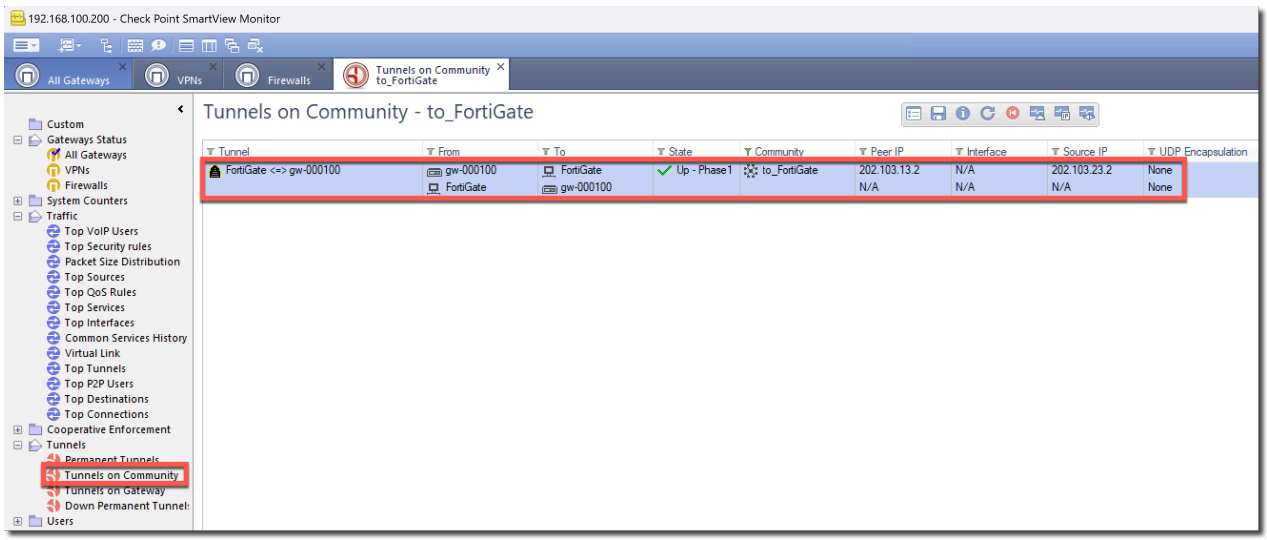
3. FortiGate 侧 PC1 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.2.2
```

```
84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=5.727 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=2.354 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=4.145 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=2.402 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=3.445 ms
```

4. Check Point 防火墙查看 IPSec 连接状态。





5. 查看 Check Point 防火墙的路由表，可以看到 VPN 流量匹配了默认路由，送往本地 VPN 网关接口 eth1（如果默认路由指向的不是本地 VPN 网关接口，请自行配置细化的 VPN 路由指向 VPN 网关接口）。

```
gw-000100> show route
Codes: C - Connected, S - Static, R - RIP, B - BGP (D - Default),
       O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA),
       A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed,
       NP - NAT Pool, U - Unreachable, i - Inactive

S               0.0.0.0/0           via 202.103.23.1, eth1, cost 0, age 12890
C               10.10.2.0/24        is directly connected, eth2
C               127.0.0.0/8         is directly connected, lo
C               192.168.100.0/24    is directly connected, eth0
C               202.103.23.0/24     is directly connected, eth1
```

6. Check Point 防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6806/64

VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=2.687 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=2.235 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=2.130 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.225 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=2.170 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-18 15:18:07.196766 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:07.196805 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:07.197221 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:07.197240 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:08.200924 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:08.200964 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:08.201134 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:08.201143 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:09.204488 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:09.204523 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:09.204736 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:09.204749 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:10.208678 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:10.208715 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:10.209058 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:10.209069 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:11.212295 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:11.212330 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-18 15:18:11.212521 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-18 15:18:11.212534 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：
   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
     FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
     FortiGate (BJ-OSPF-TO-SH) # end
     ```

2. Check Point 防火墙 IPsec VPN Debug 请参考：https://sc1.checkpoint.com/documents/R81/WebAdminGuides/EN/CP_R81_SitetoSiteVPN_AdminGuide/Topics-VPNSG/CLI/vpn-debug.htm ↗ 。

```
vpn debug
      on [<Debug_Topic>=<Debug_Level>]
      off
      ikeon [-s <Size_in_MB>]
      ikeoff
      trunc [<Debug_Topic>=<Debug_Level>]
      truncon [<Debug_Topic>=<Debug_Level>]
      truncoff
      timeon [<Seconds>]
      timeoff
      ikefail [-s <Size_in_MB>]
      mon
      moff
      say ["String"]
      tunnel [<Level>]
```

与 Juniper SRX 防火墙建立 IPSec VPN

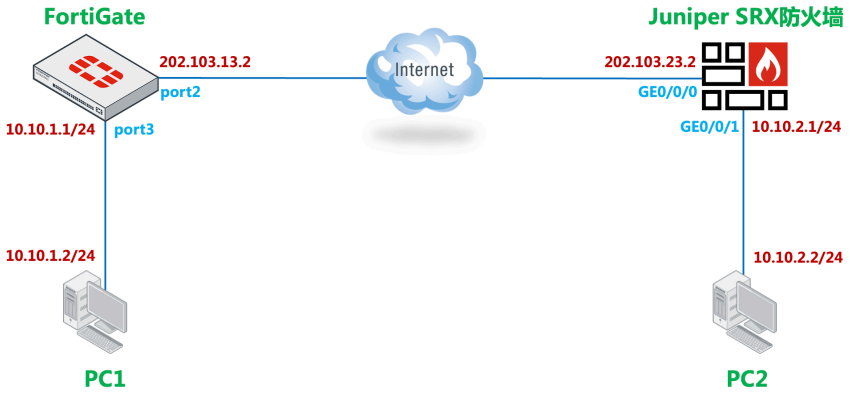
网络需求

如图所示，通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 10.10.1.0/24 与 10.10.2.0/24 两个网段的通信。与 Juniper SRX 防火墙进行 IPsec VPN（IKE v1）对接。

Juniper SRX 防火墙版本：

```
root# show version
## Last changed: 2023-04-11 03:02:08 UTC
version 23.1R1.8;
```

网络拓扑



配置要点

- 配置 FortiGate
  - 基本上网配置
  - 配置 IPsec VPN
- 配置 Juniper SRX 防火墙
  - 基本上网配置
  - 配置 IPsec VPN
- 说明：如果要删除 IPSEC VPN 第一阶段、第二阶段时，需要先删除被调用的路由与防火墙安全策略。

配置步骤

FortiGate

1. 基本上网配置。







2. 配置 IPSec VPN，进入 VPN → IPSec 隧道，点击新建 → IPSec 隧道按钮。



3. 选择 IPsec VPN 自定义模板进行配置，点击下一步。



4. 如图配置网络、认证、第一阶段、第二阶段。

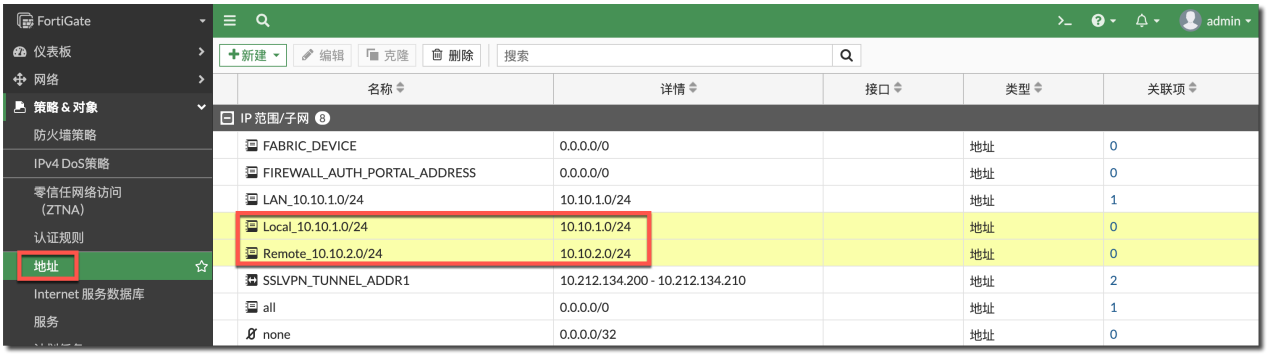
重要

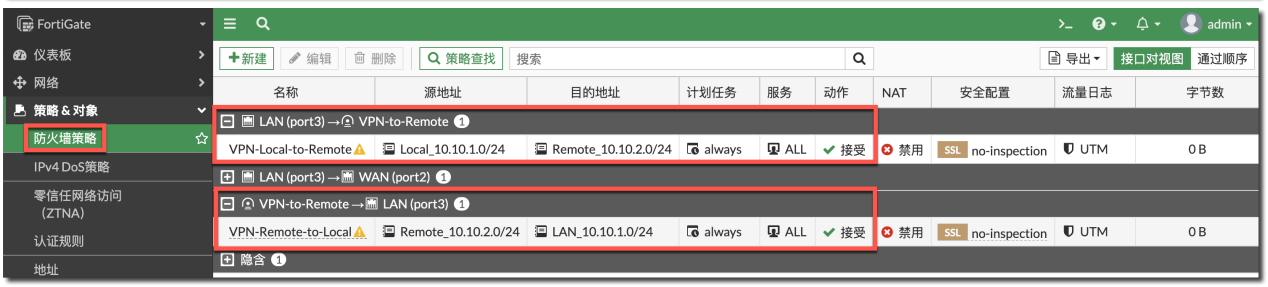
由于 Juniper 的 IPSec 配置向导会创建明细的感兴趣流，FortiGate 与 Juniper 防火墙对接 IPSec 推荐对应使用明细的感兴趣流。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.11
新VPN隧道
名称
VPN-to-Remote
注释
注释
0/255
网络
IP版本
IPv4
IPv6
对端网关
静态IP地址
IP地址
202.103.23.2
接口
WAN (port2)
本地网关
模式配置
NAT穿越
启用
禁用
强制的
保持存活
10
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
20
s
前向纠错
Egress
Ingress
高级
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
模式
野蛮
主模式(ID保护)
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
密钥生存时间 (秒)
86400
本地ID
XAUTH
类型
已禁用
阶段 2 选择器
名称
本地地址
对端地址
VPN-to-Remote
10.10.1.0/24
10.10.2.0/24
新建阶段 2
名称
VPN-to-Remote
注释
注释
本地地址
子网
10.10.1.0/24
对端地址
子网
10.10.2.0/24
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
加密
AES128GCM
加密
AES256GCM
加密
CHACHA20POLY1305
启用重播检测
启用完全前向保密 (PFS)
Diffie-Hellman 组
32
31
30
29
28
27
21
20
19
18
17
16
15
14
5
2
1
本地端口
全部
对端端口
全部
协议
全部
自动协商
自动密钥保持存活
密钥周期(秒/kb)
秒
秒
43200
确认
取消

```
config vpn ipsec phase1-interface
    edit "VPN-to-Remote"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set remote-gw 202.103.23.2
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "VPN-to-Remote"
        set phase1name "VPN-to-Remote"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 10.10.1.0 255.255.255.0
        set dst-subnet 10.10.2.0 255.255.255.0
    next
end
```

5. 配置 VPN 相关的网段地址对象和防火墙策略。



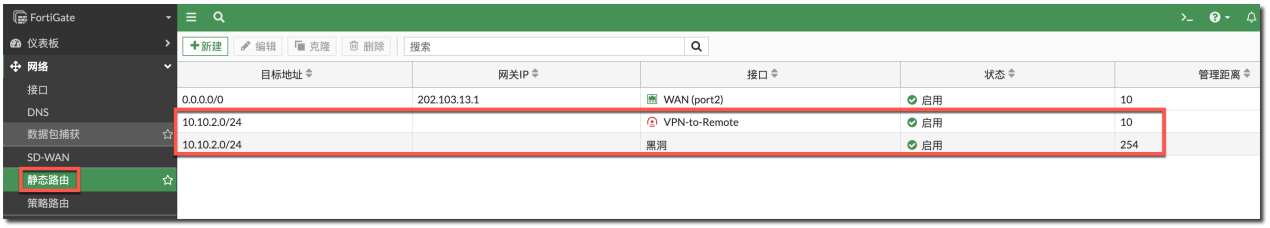



```
config firewall address
    edit "Local_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "Remote_10.10.2.0/24"
        set subnet 10.10.2.0 255.255.255.0
    next
end

config firewall policy
    edit 2
        set name "VPN-Local-to-Remote"
        set srcintf "port3"
        set dstintf "VPN-to-Remote"
        set action accept
        set srcaddr "Local_10.10.1.0/24"
        set dstaddr "Remote_10.10.2.0/24"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "VPN-Remote-to-Local"
        set srcintf "VPN-to-Remote"
```

```
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_10.10.2.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

6. 配置 VPN 业务网段的静态路由。

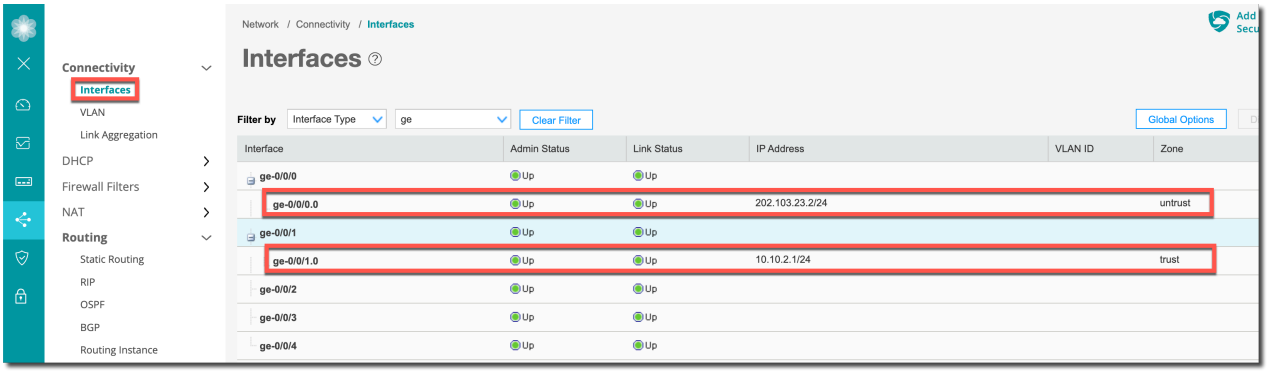
> **重要**
>
> 相关 VPN 的路由配置中“黑洞路由的意义：
>
> 你可能会遇到以下的 VPN 业务问题：通过 VPN 的 SIP 电话，时不时中断，无法向服务器成功注册？ 通过 VPN 的 Radius 认证无法经常会无法认证成功？ 通过 VPN 的 AP 注册到总部时不时会中断？持续的 PRGT 监控 ping 总部的服务器，时不时会显示中断?
>
> VPN 有时候会因为各种原因重新连接，比如互联网不稳定，PPPOE 重新连接更换公网 IP 地址等，VPN tunnel 此时会出现短暂的 DOWN，而去往对方的 VPN 业务网段的路由也会短暂消失，此时 VPN 的业务流量（SIP 注册请求/Radius/Capwap/ICMP）会因为查询到了默认路由而走向了 WAN（Internet），从而产生了错误的 UDP-NAT-Seesion，此时即便 VPN tunnel 再次 UP，VPN 业务网段的路由再次恢复，SIP 等 VPN 旧的业务流量依旧会走到错误的 Session 上去，从而引起该业务异常。
>
> 解决方法：
>
> - 方法一：配置去往 VPN 业务网段的“黑洞路由”，管理距离为 254，VPN 正常的情况下此黑洞路由不生效，而当 VPN 中断的时候，黑洞路由浮起来并且生效，将 VPN 中断时候的 VPN 流量丢弃，避免将流量转发到互联网而产生错误的 session。配置 VPN 的备份黑洞路由的意义在此。
> - 方法二：配置源接口：LAN，目的接口：WAN，源 IP：本地内网网段，目的 IP：对端内网网段，动作：丢包的策略。将此去往 Internet 的私网（无用的）流量丢弃掉，避免 FGT 产生这种错误的 session，从而避免了 UDP 业务时不时中断的问题。
> - 方法三：在全局下开启“set snat-route-change enable”，一旦路由发生变化，将相关的会话 flag 置位为“Dirty”，清除路由缓存并重新查找目的 IP 的下一跳地址，这样 VPN 隧道恢复的时候，流量也会相应的切换到 VPN 隧道里面来。
>
> ```
> FortiGate # config system global
> FortiGate (global) # set snat-route-change enable
> FortiGate (global) # end
> ```
>
> 三种方式任意选择一种即可。推荐使用黑洞路由方式。

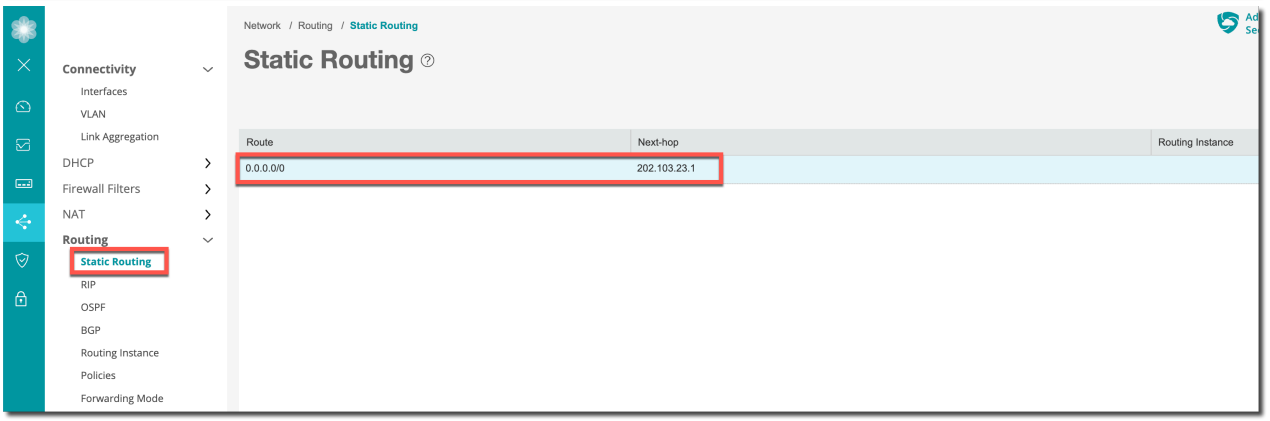



```
config router static
    edit 1
        set gateway 202.103.12.1
        set device "port2"
    next
    edit 2
        set dst 10.10.2.0 255.255.255.0
        set device "VPN-to-Remote"
    next
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 254
        set blackhole enable
    next
end
```

Juniper SRX

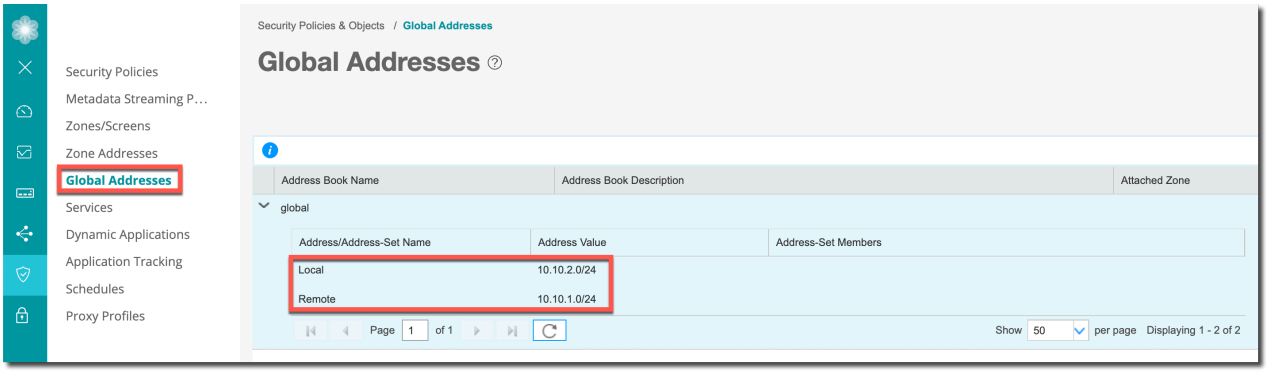
1. 基本 IP 与路由配置。

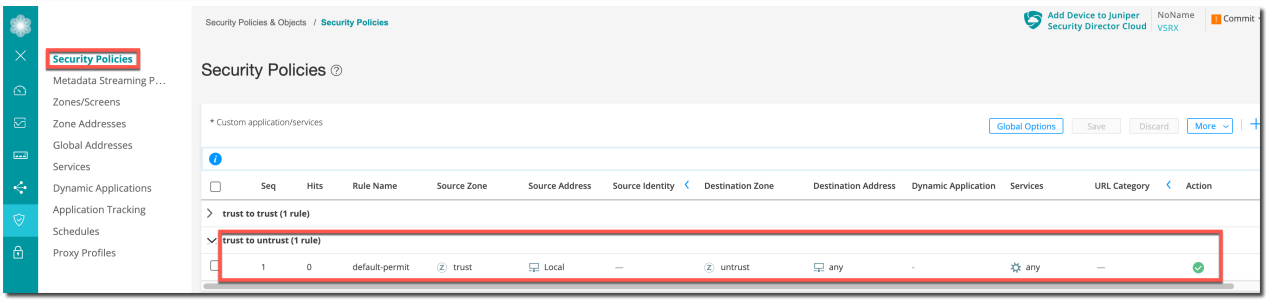


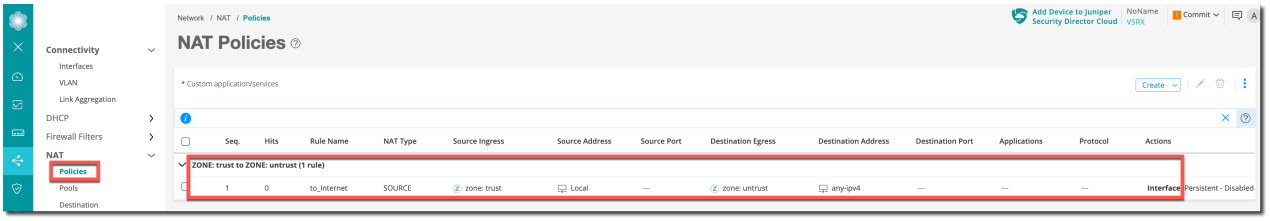



```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 202.103.23.2/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.10.2.1/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 202.103.23.1;
    }
}
```

2. 地址对象、上网策略（默认存在）、上网 NAT 配置。



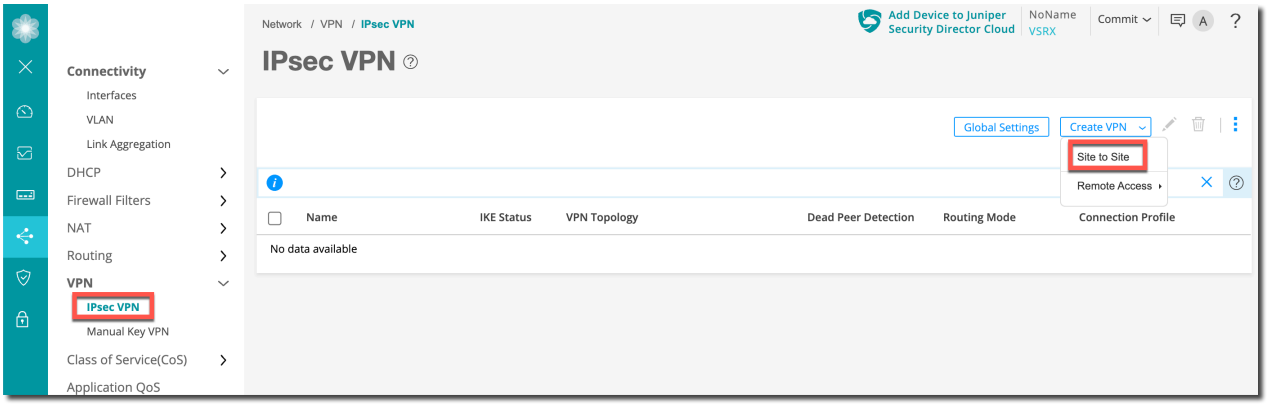




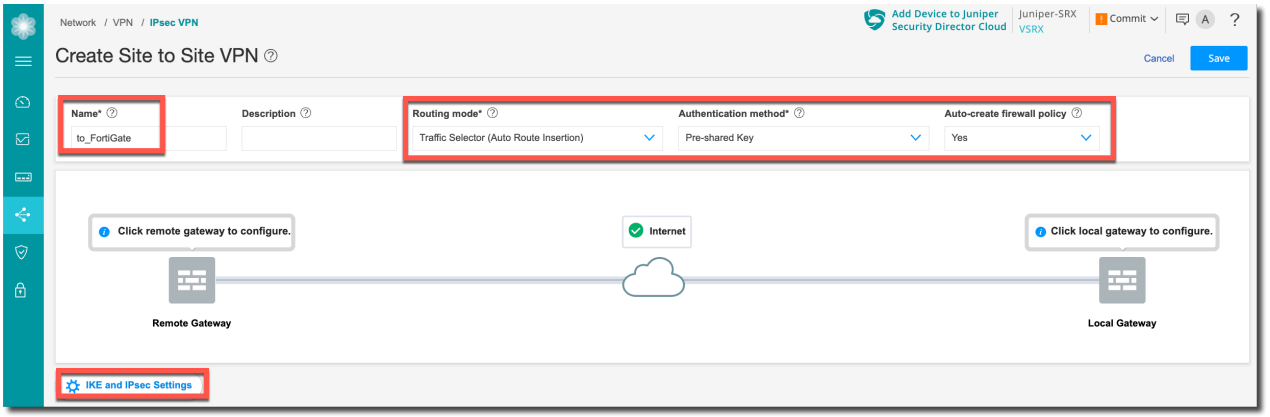

```
security {
    address-book {
        global {
            address Local 10.10.2.0/24;
            address Remote 10.10.1.0/24;
        }
    }
    nat {
        source {
            rule-set Zone_trust-Zone_untrust {
                from zone trust;
                to zone untrust;
                rule to_Internet {
                    match {
                        source-address-name Local;
                        destination-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            interface;
                        }
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy default-permit {
                match {
                    source-address Local;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
```

```
                }
            }
        }
    }
}
```

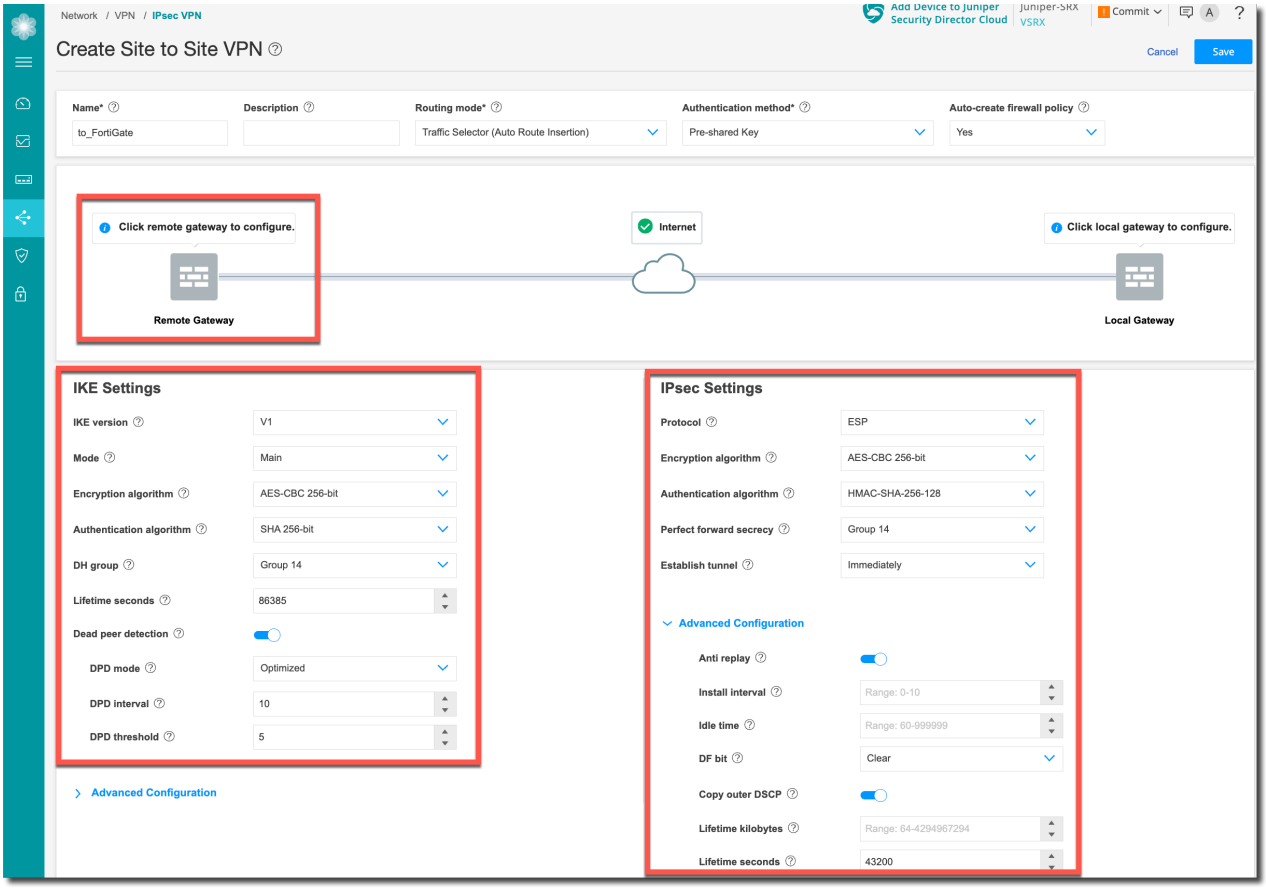
3. 在 IPSec VPN 页面点击“Create VPN”按钮，并选择“Site to Site”模式。



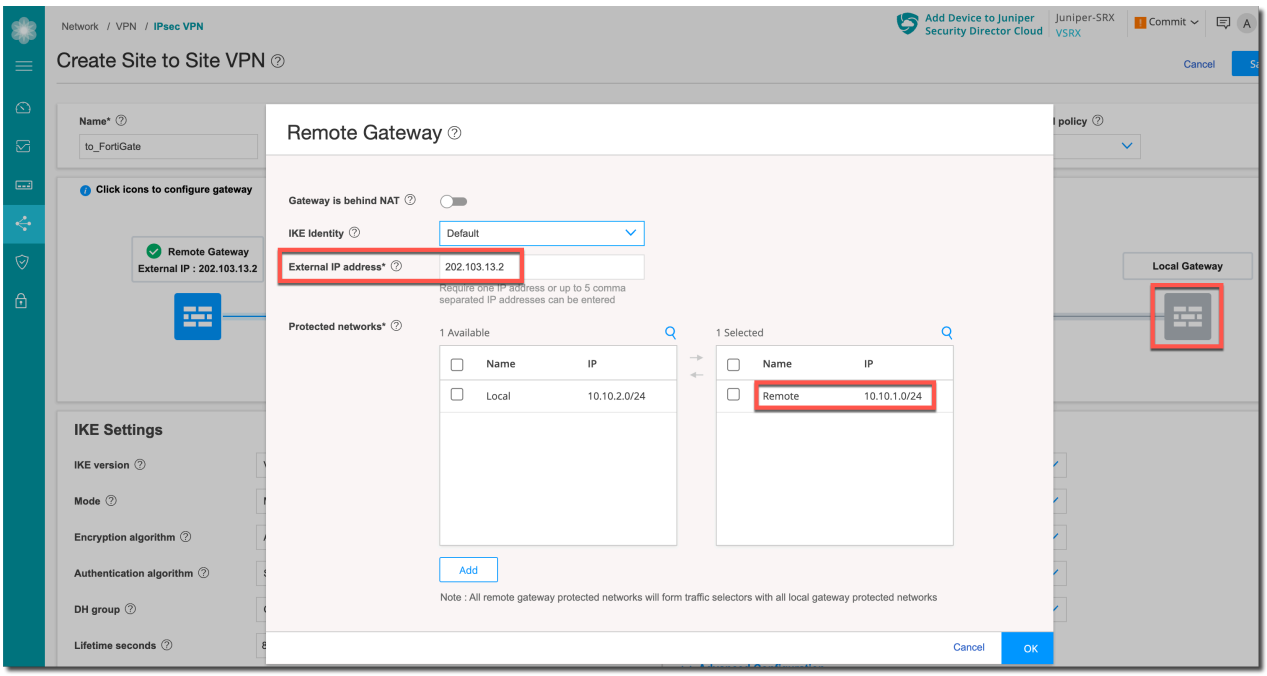
4. 填写 IPSec VPN 名称，“Routing mode”选择“Traffic Selector”（明细感兴趣流，并自动添加到对端 VPN 网段的路由），“Authentication method”选择“Pre-shared Key”，“Auto-create firewall policy”选择“Yes”（自动添加 VPN 网段之间的放通策略）。随后点击左下角的“IKE and IPSec Settings”。



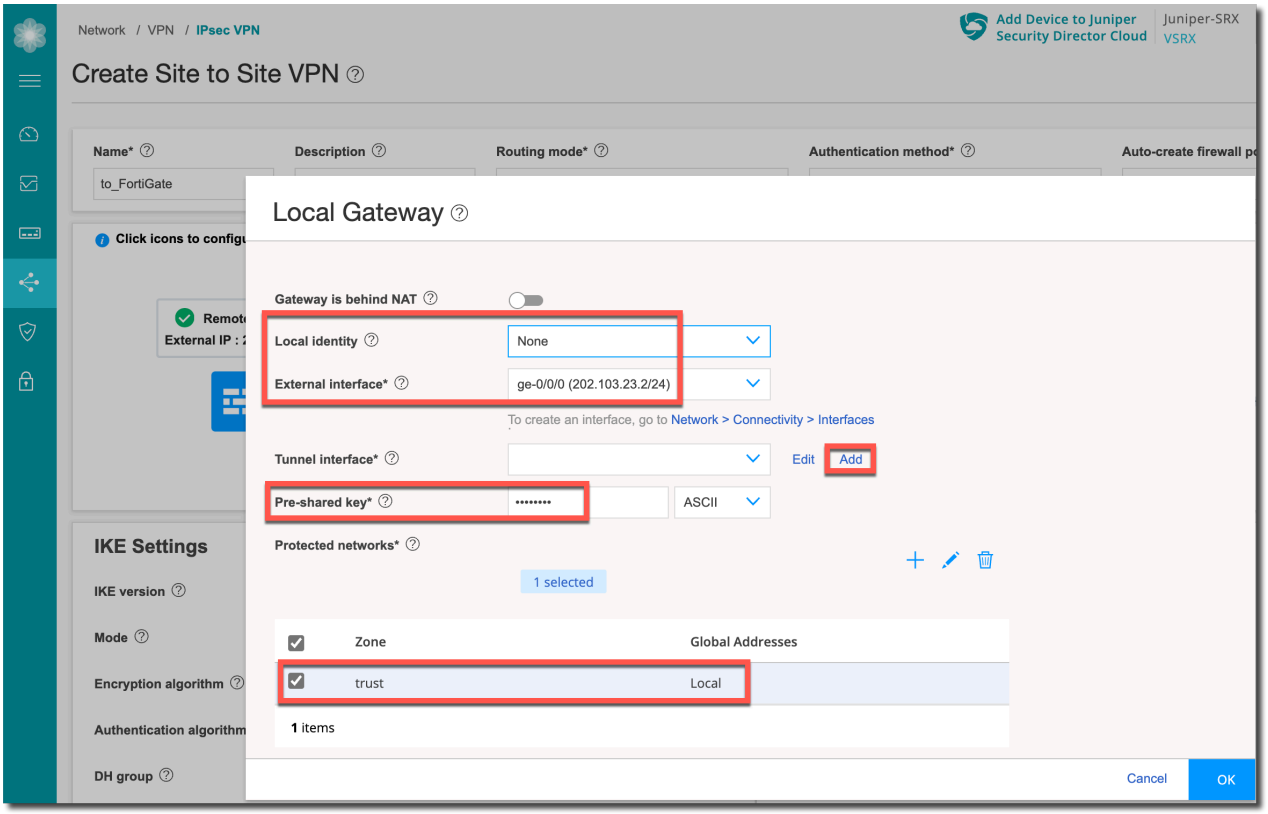
5. 配置 IKE 与 IPSec 协商参数，与 FortiGate 配置为一致，配置完成后，点击“Remote Gateway”配置远程网关。



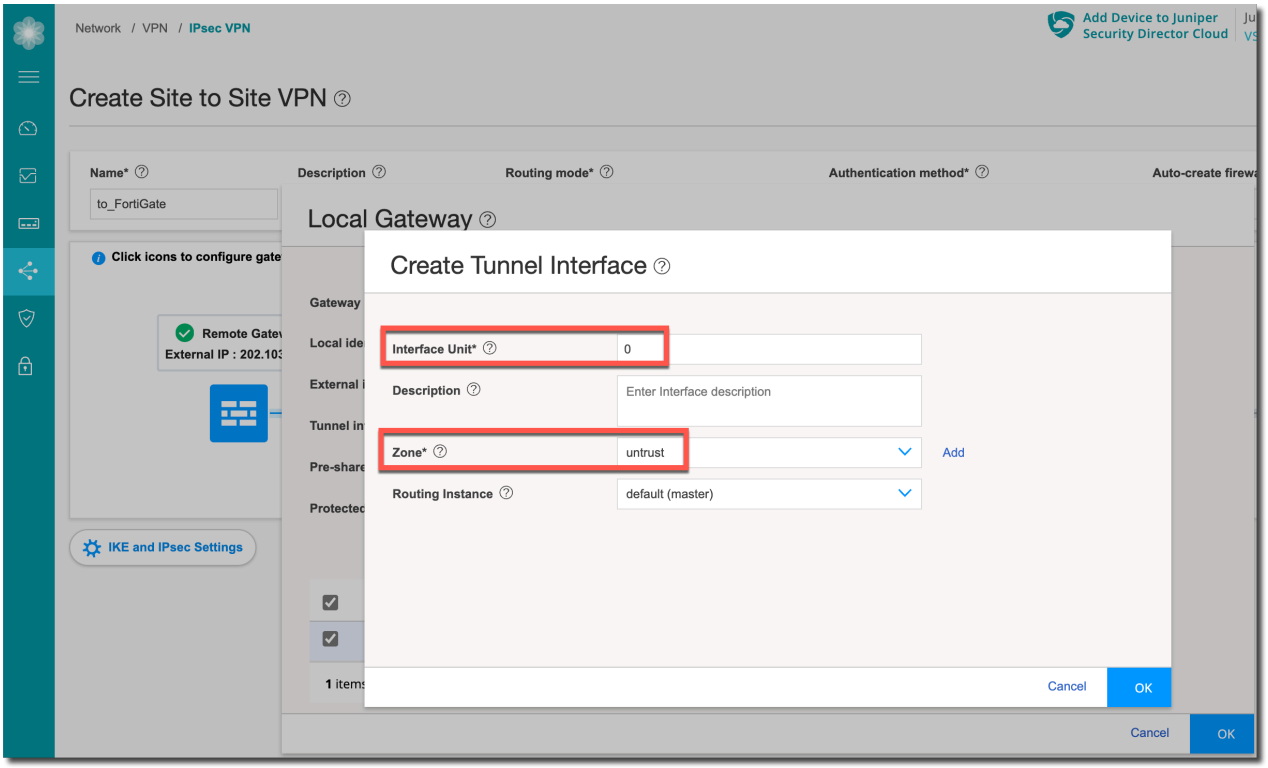
6. 根据实际情况选择是否修改“Gateway is behind NAT”以及 ID 配置。在“External IP address”中配置 FortiGate 侧的 VPN 网关地址，“Protected networks”选中 FortiGate 侧的 VPN 网段，点击 OK 下发配置。随后点击“Local Gateway”配置本地网关。



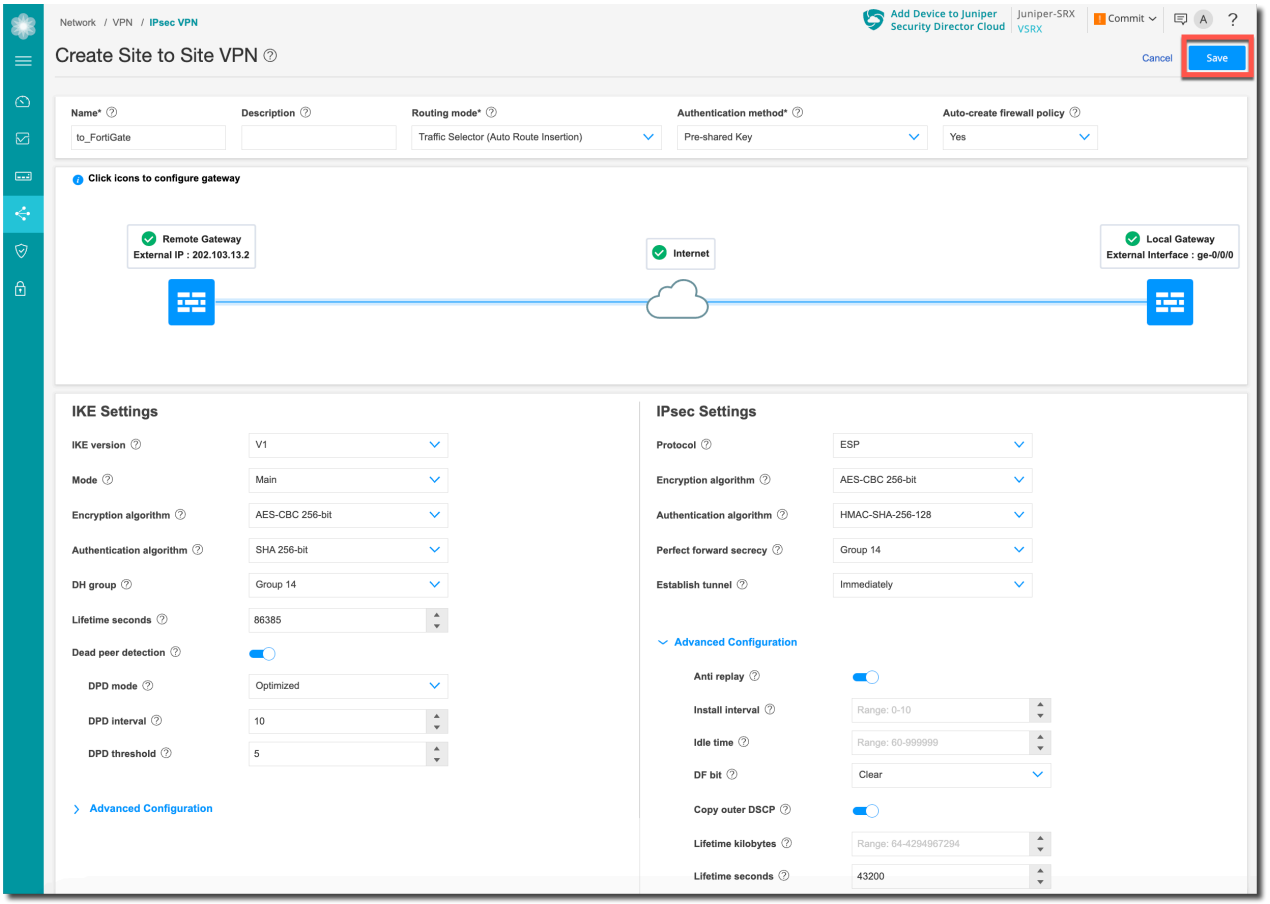
7. 点击“Local Gateway”配置本地网关，选择“External interface”为本地网关公网接口，填写预共享密钥，“Protected networks”选择本地的 VPN 网段，随后在“Tunnel interface”右侧点击“Add”。



8. 新建 Tunnel 接口，配置 Unit 号与安全域信息，随后下发本地网关的配置。



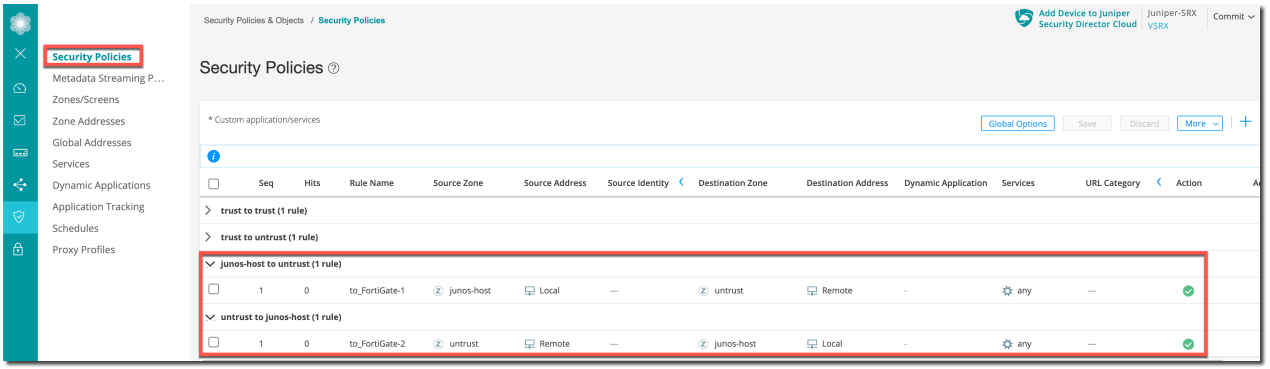
9. 点击“Save”保存 IPSec VPN 配置。




```
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
security {
    ike {
        proposal to_FortiGate {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 86385;
        }
        policy to_FortiGate {
            mode main;
            proposals to_FortiGate;
            pre-shared-key ascii-text "$9$KnSMxNdbs2oJx7Dikqf51REcev"; ## SECRET-DATA
        }
        gateway to_FortiGate {
            ike-policy to_FortiGate;
            address 202.103.13.2;
            dead-peer-detection {
                optimized;
                interval 10;
                threshold 5;
            }
            external-interface ge-0/0/0;
            local-address 202.103.23.2;
            version v1-only;
        }
    }
    ipsec {
        proposal to_FortiGate {
            protocol esp;
```

```
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 43200;
        }
        policy to_FortiGate {
            perfect-forward-secrecy {
                keys group14;
            }
            proposals to_FortiGate;
        }
        vpn to_FortiGate {
            bind-interface st0.0;
            df-bit clear;
            copy-outer-dscp;
            ike {
                gateway to_FortiGate;
                ipsec-policy to_FortiGate;
            }
            traffic-selector ts-1 {
                local-ip 10.10.2.0/24;
                remote-ip 10.10.1.0/24;
            }
            establish-tunnels immediately;
        }
    }
}
```

.0. 查看自动添加的 VPN 网段安全策略。



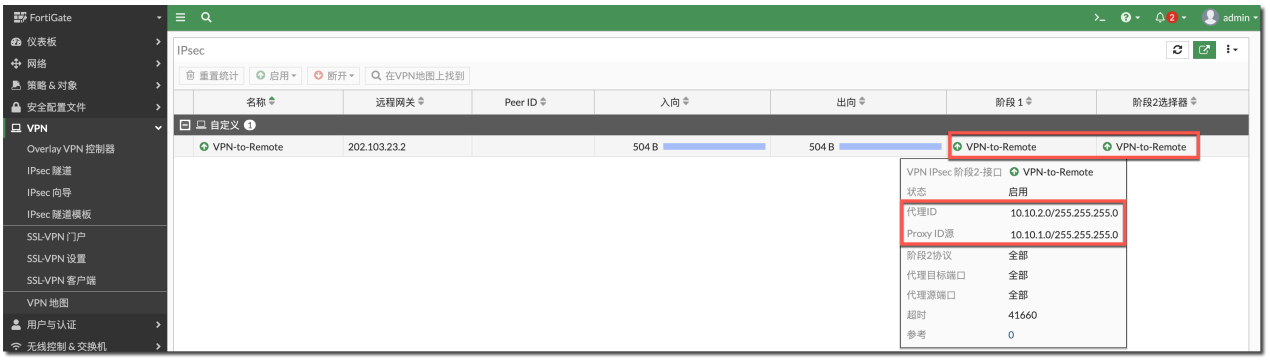

```
security {
    policies {
            policy to_FortiGate-1 {
                match {
                    source-address Local;
                    destination-address Remote;
                    application any;
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
        }
        from-zone junos-host to-zone untrust {
            policy to_FortiGate-1 {
                match {
                    source-address Local;
                    destination-address Remote;
                    application any;
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
```

```
                }
            }
        }
        from-zone untrust to-zone junos-host {
            policy to_FortiGate-2 {
                match {
                    source-address Remote;
                    destination-address Local;
                    application any;
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
        }
        from-zone untrust to-zone trust {
            policy to_FortiGate-2 {
                match {
                    source-address Remote;
                    destination-address Local;
                    application any;
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
        }
    }
}
```

1. 最后，点击页面右上角的 Commit 按钮提交所有配置。

结果验证

1. FortiGate 上查看 IPSec 隧道建立，在仪表盘新建 IPSec 监控，可以看到 IPSec 建立成功。




```
FortiGate # diagnose vpn ike gateway list
vd: root/0
name: VPN-to-Remote
version: 1
interface: port2 4
addr: 202.103.13.2:500 → 202.103.23.2:500
tun_id: 202.103.23.2/::202.103.23.2
remote_location: 0.0.0.0
network-id: 0
created: 3s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms
  id/spi: 84 81f7935c958dc231/0d50274a3603ed0c
  direction: initiator
```

```
  status: established 3-3s ago = 10ms
  proposal: aes256-sha256
  key: 7617c0e9088a38f0-ba9b72527df54344-75a4f0b475a5c89a-ea60570f9f6a7f36
  lifetime/rekey: 86400/86096
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=VPN-to-Remote ver=1 serial=1 202.103.13.2:0→202.103.23.2:0 tun_id=202.103.23.2 tun_id6=::202.103.23.2 dst_mtu=1500
dpd-link=on weig1
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overla0
proxyid_num=1 child_num=0 refcnt=4 ilast=2 olast=2 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=65
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=VPN-to-Remote proto=0 sa=1 ref=2 serial=2 auto-negotiate
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.2.0-10.10.2.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1438 expire=42888/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=0f3295e9 esp=aes key=32 1ef676bbce325c056b2bd2e7bdcc04b53d5493123ffff37734b92d7f0be650dc
       ah=sha256 key=32 1e0dd4317b3cf00168dad8528b820aadae59a09dac665d094324210b22abcadf
  enc: spi=06cf5a4d esp=aes key=32 06d4be5946563a9b7db6ad62c12625e61c18f83807abf62dbdf96d62f4dd2856
       ah=sha256 key=32 a11616829e581b70f3ec1e4fbecde7efb85a6da8cdb1718e34b40e12fbc5779f
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.13.2 npu_selid=1 dec_npuid=0 enc_npuid=0
run_tally=0
```

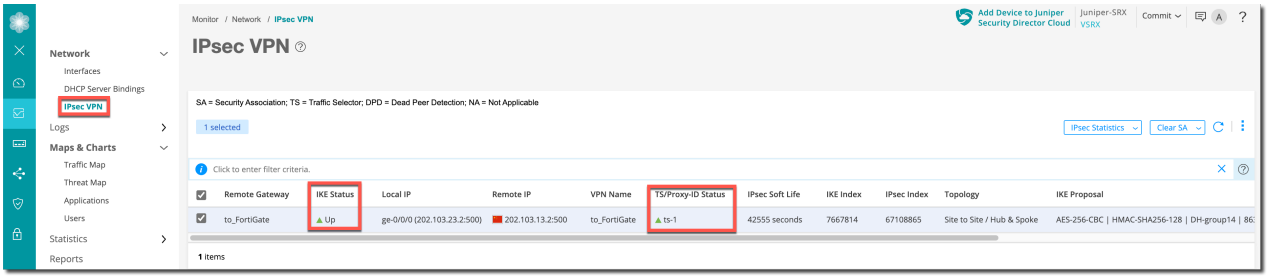
2. 查看 FortiGate 的路由表。

```
FortiGate # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 202.103.13.1, port2, [1/0]
C       10.10.1.0/24 is directly connected, port3
S       10.10.2.0/24 [10/0] via VPN-to-Remote tunnel 202.103.23.2, [1/0]
C       192.168.100.0/24 is directly connected, port1
C       202.103.13.0/24 is directly connected, port2
```

3. FortiGate 侧 PC1 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.1.2/24         10.10.1.1
       fe80::250:79ff:fe66:6805/64

VPCS> ping 10.10.2.2

84 bytes from 10.10.2.2 icmp_seq=1 ttl=62 time=4.768 ms
84 bytes from 10.10.2.2 icmp_seq=2 ttl=62 time=1.770 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=62 time=1.698 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=62 time=2.116 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=62 time=1.780 ms
```

4. Juniper 防火墙查看 IPSec 连接状态。




```
root@Juniper-SRX> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
7667814 UP     81f7935c958dc231  0d50274a3603ed0c  Main           202.103.13.2

root@Juniper-SRX>  show security ike security-associations 202.103.13.2 detail
IKE peer 202.103.13.2, Index 7667814, Gateway Name: to_FortiGate
  Role: Responder, State: UP
  Initiator cookie: 81f7935c958dc231, Responder cookie: 0d50274a3603ed0c
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 202.103.23.2:500, Remote: 202.103.13.2:500
  Lifetime: Expires in 86162 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 202.103.13.2
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2996
   Output bytes  :                 2196
   Input  packets:                   15
   Output packets:                   14
   Input  fragmentated packets:       0
   Output fragmentated packets:       0
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 1
    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 202.103.23.2:500, Remote: 202.103.13.2:500
    Local identity: 202.103.23.2
    Remote identity: 202.103.13.2
    Flags: IKE SA is created
```

```
root@Juniper-SRX> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108865 ESP:aes-cbc-256/sha256 6cf5a4d 42899/ unlim - root 500 202.103.13.2
  ::: important
  67108865 ESP:aes-cbc-256/sha256 f3295e9 42899/ unlim - root 500 202.103.13.2
  :::

root@Juniper-SRX> show security ipsec security-associations vpn-name to_FortiGate detail
ID: 67108865 Virtual-system: root, VPN Name: to_FortiGate
  Local Gateway: 202.103.23.2, Remote Gateway: 202.103.13.2
  Traffic Selector Name: ts-1
  Local Identity: ipv4(10.10.2.0-10.10.2.255)
  Remote Identity: ipv4(10.10.1.0-10.10.1.255)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Enabled, Bind-interface: st0.0
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x2c608b29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Tue Apr 11 2023 15:42:30
    : IPSec SA negotiation successfully completed          (1 times)
    Tue Apr 11 2023 15:42:30
    : IKE SA negotiation successfully completed            (4 times)
    Tue Apr 11 2023 15:42:29
```

```
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Tue Apr 11 2023 15:42:25
    : IPSec SA negotiation successfully completed          (1 times)
    Tue Apr 11 2023 15:42:25
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Tue Apr 11 2023 15:20:00
    : IPSec SA negotiation successfully completed          (1 times)
    Tue Apr 11 2023 15:20:00
    : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
  Direction: inbound, SPI: 6cf5a4d, AUX-SPI: 0
                             , VPN Monitoring: -
    Hard lifetime: Expires in 42865 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 42302 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: f3295e9, AUX-SPI: 0
                             , VPN Monitoring: -
    Hard lifetime: Expires in 42865 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 42302 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

5. 查看 Juniper 防火墙的路由表，可以看到包含了自动注入的到 FortiGate 内网的路由，出接口为 Tunnel 接口。

```
root@Juniper-SRX> show route

inet.0: 8 destinations, 9 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 00:58:27
                    ::: important
                     to 202.103.23.1 via ge-0/0/0.0
                    :::
                    [Access-internal/12] 00:59:24, metric 0
                    ::: important
                     to 192.168.100.99 via fxp0.0
                    :::
10.10.1.0/24       *[Static/5] 00:19:34
                    ::: important
                     via st0.0
                    :::
10.10.2.0/24       *[Direct/0] 00:58:27
                    ::: important
                     via ge-0/0/1.0
                    :::
10.10.2.1/32       *[Local/0] 00:58:27
                        Local via ge-0/0/1.0
192.168.100.0/24   *[Direct/0] 00:59:24
                    ::: important
                     via fxp0.0
                    :::
192.168.100.4/32   *[Local/0] 00:59:24
                        Local via fxp0.0
202.103.23.0/24    *[Direct/0] 00:58:27
                    ::: important
                     via ge-0/0/0.0
                    :::
202.103.23.2/32    *[Local/0] 00:58:27
                        Local via ge-0/0/0.0
```

6. Juniper 防火墙侧 PC2 业务测试。

```
VPCS> show

NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  10.10.2.2/24         10.10.2.1
       fe80::250:79ff:fe66:6806/64
```

```
VPCS> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=62 time=1.410 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=62 time=1.547 ms
84 bytes from 10.10.1.2 icmp_seq=3 ttl=62 time=1.624 ms
84 bytes from 10.10.1.2 icmp_seq=4 ttl=62 time=2.176 ms
84 bytes from 10.10.1.2 icmp_seq=5 ttl=62 time=1.661 ms
```

7. FortiGate 侧抓包查看。

```
FortiGate # diagnose sniffer packet any 'host 10.10.2.2 and host 10.10.1.2 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.2 and host 10.10.1.2 and icmp]
2023-04-11 15:49:30.598552 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:30.598585 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:30.598754 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:30.598761 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:31.601426 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:31.601457 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:31.601769 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:31.601774 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:32.604474 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:32.604508 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:32.604694 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:32.604701 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:33.607589 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:33.607647 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:33.608105 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:33.608124 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:34.609922 VPN-to-Remote in 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:34.609955 port3 out 10.10.2.2 → 10.10.1.2: icmp: echo request
2023-04-11 15:49:34.610140 port3 in 10.10.1.2 → 10.10.2.2: icmp: echo reply
2023-04-11 15:49:34.610153 VPN-to-Remote out 10.10.1.2 → 10.10.2.2: icmp: echo reply
```

说明

1. 关于 sniffer 抓 VPN 业务和 ESP 的包：

   - 抓取 IPsec VPN 的 IKE 协商包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and (port 500 or port 4500)" 4
     ```

   - 抓取 IPsec VPN 的 ESP 加密数据包：

     ```
     diagnose sniffer packet any "host 101.1.1.1 and esp" 4
     ```

   - 抓取 IPsec VPN 的明文业务数据包：

     ```
     diagnose sniffer packet any "host 192.168.112.100 and icmp" 4
     ```

   - 注意：由于硬件设备存在 IPsec VPN 芯片加速，因此可能数据包会抓不完全，主要指“ESP 数据和明文业务数据”抓不全，因此有时候需要将 VPN 隧道的 NP 加速关闭：

     ```
     FortiGate # config vpn ipsec phase1-interface
     FortiGate (phase1-interface) # edit BJ-OSPF-TO-SH
     FortiGate (BJ-OSPF-TO-SH) # set npu-offload disable
     FortiGate (BJ-OSPF-TO-SH) # end
     ```

2. juniper 防火墙相关：

   - juniper 防火墙 IPsec VPN Debug：

     ```
     # set system syslog file kmd-logs daemon info
     # set system syslog file kmd-logs match KMD
     # commit
     show log kmd-logs
     ```

   - 参考文档：

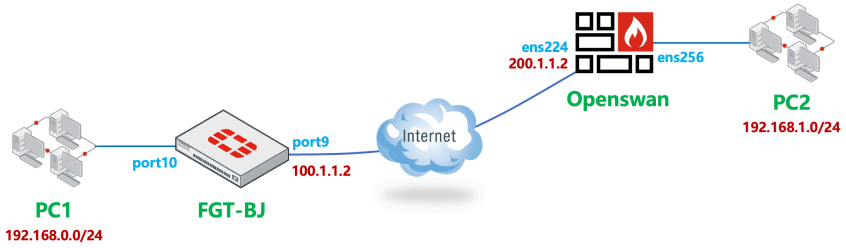
- https://supportportal.juniper.net/s/article/SRX-Resolution-Guide-How-to-troubleshoot-Problem-Scenarios-in-VPN-tunnels ↗
- https://supportportal.juniper.net/s/article/SRX-How-to-configure-syslog-to-display-VPN-status-messages ↗
- https://supportportal.juniper.net/s/article/SRX-IKE-Phase-1-VPN-status-messages ↗
- https://supportportal.juniper.net/s/article/SRX-How-to-troubleshoot-IKE-Phase-2-VPN-connection-issues ↗

与 Openswan 建立 IPSec VPN

网络需求

通过 IPsec VPN（接口模式）将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.1.0/24 两个网段的通信。

网络拓扑



配置步骤

FortiGate IPSec 配置

1.基本配置

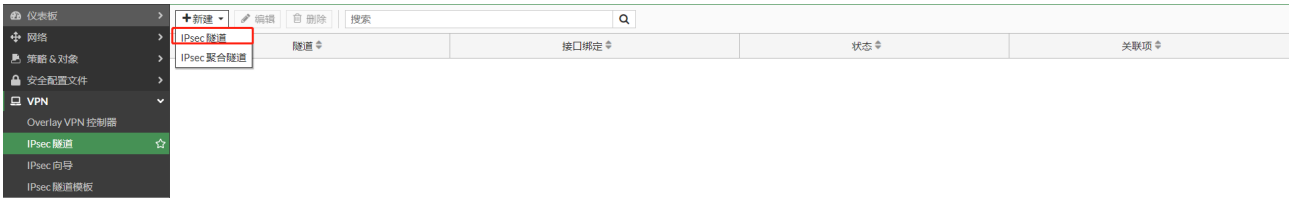
配置接口 IP 和路由。





2.配置 IPSEC

选择“VPN”-->“IPsec 隧道”，点击“新建”，选择“IPsec 隧道”。



根据“VPN 创建向导”进行 VPN 模板配置，输入名称，拓扑中没有 NAT，这里选择“站点之间没有 NAT”，并点击下一步。



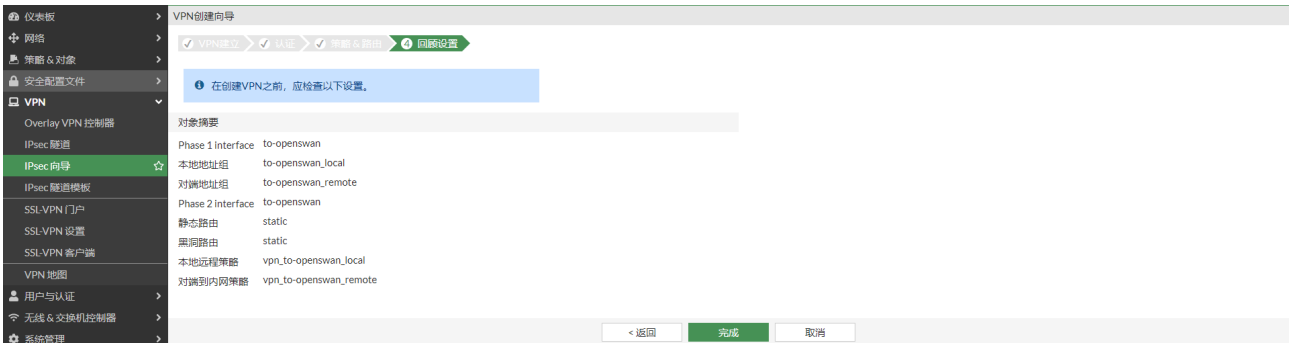
输入对端设备的 IP 地址（流出接口会根据路由自动选择），以及预共享秘钥。



选择“本地接口”，即需要被 IPSEC 保护的内网的接口，这里是 port10 接口，本地子网会自动写为该接口的 IP 所在的 IP 网段（如果被保护的是其他网段，可以根据实际网络需求修改），输入“远端子网”，即远端需要被保护的网络。然后点击“下一步”。



VPN 创建向导提示即将创建的内容，然后点击完成。



VPN 创建成功。



3.设置 dpd 和自动连接

在第一阶段中开启 DPD 周期性检测，每隔 10s 检测一次 Peer 状态。

```
config vpn ipsec phase1-interface
    edit "to-openswan"
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 10
    next
end
```

开启自动协商，主动让隧道 UP 起来，而非使用 VPN 业务的时候再去触发 VPN 的协商，这样可以减少业务的丢包。在 VPN 主动发起方开启即可。

IPSEC VPN 阶段一自动协商是默认开启的。

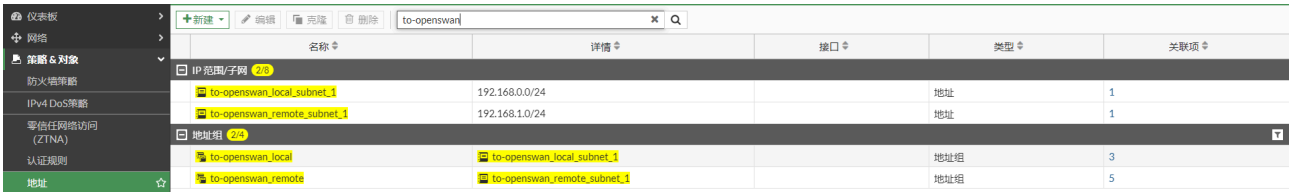
```
config vpn ipsec phase1-interface
    edit "to-openswan"
        set auto-negotiate enable
    next
end
```

IPSEC VPN 阶段二自动协商默认关闭，需要开启。

```
config vpn ipsec phase2-interface
    edit "to-openswan"
        set auto-negotiate enable
    next
end
```

###查看 IPSEC 向导所作的配置

1.地址对象配置

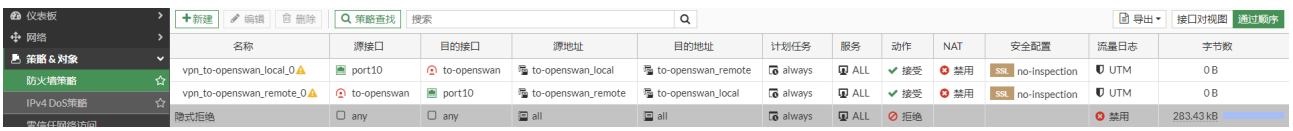


2.IPSEC 配置

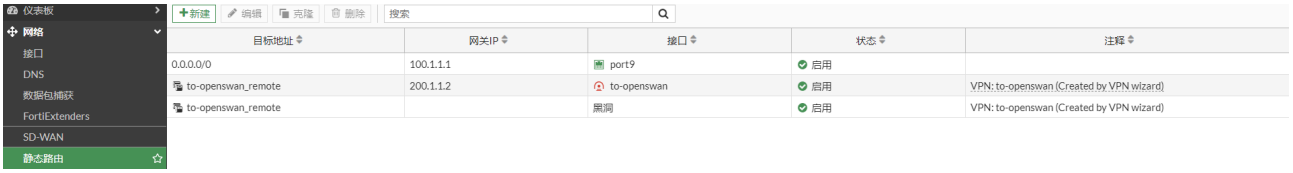



```
config vpn ipsec phase1-interface
    edit "to-openswan"
        set interface "port9"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set comments "VPN: to-openswan (Created by VPN wizard)"
        set wizard-type static-fortigate
        set remote-gw 200.1.1.2
        set psksecret ENC
p62NOSnHRHf0H/FDMjbdenLf9XFZ8AWBxxp4ztydCj8wqSRTiYDcbEBov4ZKf/5xUwzXl3tE3mUPLyRP9h6gq1YJGR2kocyO1hqc/Iy3hQa+LYORLlznaMEKcx
m6bKsNPeJHCe+FiOBatOW6pw9y8Vxwbp1yGvTkkKmyzfW3FWWzwbFHsa1AxigVRlZCsEKIzbW3dw==
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "to-openswan"
        set phase1name "to-openswan"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
        set comments "VPN: to-openswan (Created by VPN wizard)"
        set src-addr-type name
        set dst-addr-type name
        set src-name "to-openswan_local"
        set dst-name "to-openswan_remote"
    next
end
```

3.策略配置



4.路由配置



配置 openswan

1.接口及路由配置

```
# ifconfig
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 200.1.1.2  netmask 255.255.255.0  broadcast 200.1.1.255
        inet6 fe80::2652:4dd7:5d0e:941d  prefixlen 64  scopeid 0x20<link>
        inet6 240e:604:109:39::216  prefixlen 64  scopeid 0x0<global>
        ether 00:0c:29:37:f0:ac  txqueuelen 1000  (Ethernet)
        RX packets 10  bytes 2266 (2.2 KiB)
        RX errors 0  dropped 500  overruns 0  frame 0
        TX packets 18  bytes 2284 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
ens256: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::227:594:d82:e098  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:37:f0:b6  txqueuelen 1000  (Ethernet)
        RX packets 2519  bytes 389134 (380.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 150  bytes 10928 (10.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         200.1.1.1       0.0.0.0         UG    106    0        0 ens224
192.168.1.0     0.0.0.0         255.255.255.0   U     105    0        0 ens256
200.1.1.0       0.0.0.0         255.255.255.0   U     106    0        0 ens224
```

2.安装 openswan

测试环境中 centos 版本

```
# rpm --query centos-release
centos-release-7-9.2009.1.el7.centos.x86_64

# uname -a
Linux centos7-3 3.10.0-1160.49.1.el7.x86_64 #1 SMP Tue Nov 30 15:51:32 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

安装 openswan

```
yum install -y openswan
```

查看 openswan 安装的配置文件，libreswan 即 openswan。

```
# rpm -qc libreswan
/etc/ipsec.conf
/etc/ipsec.d/policies/block
/etc/ipsec.d/policies/clear
/etc/ipsec.d/policies/clear-or-private
/etc/ipsec.d/policies/portexcludes.conf
/etc/ipsec.d/policies/private
/etc/ipsec.d/policies/private-or-clear
/etc/ipsec.secrets
/etc/pam.d/pluto
```

3.开启 ipv4 转发，关闭 ICMP 重定向

```
cat >> /etc/sysctl.conf << EOF
echo net.ipv4.ip_forward = 1
EOF

sysctl -a | egrep "ipv4.*(accept|send)_redirects" | awk -F "=" '{print$1"= 0"}'
sysctl -a | egrep "ipv4.*(accept|send)_redirects" | awk -F "=" '{print$1"= 0"}' >> /etc/sysctl.conf
sysctl -p /etc/sysctl.conf
```

4.关闭 SELINUX

```
临时关闭：setenforce 0

永久关闭：
#vi /etc/selinux/config
SELINUX=disabled
```

5.关闭防火墙或者开启 IPSEC 所需的端口 UDP 500，UDP4500，ESP

这里关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

6.配置 openswan

```
在/etc/ipsec.conf中默认包含如下配置，建议将IPsec连接作为单独的文件添加到/etc/ipsec.d/
include /etc/ipsec.d/*.conf
```

```
在/etc/ipsec.secrets中默认包含如下配置，建议将IPsec共享密钥作为单独的文件添加到/etc/ipsec.d/
include /etc/ipsec.d/*.secrets
```

配置 IPSEC 预共享密钥

```
vim /etc/ipsec.d/ipsec.secrets
100.1.1.2  : psk "ipsec-key"

格式：本地用于连接的IP+空格+远端网关IP+空格+英文冒号+空格+PSK+预共享密钥，冒号的两边都有空格，密钥用英文双引号。
```

配置 IPSEC 连接

```
vim /etc/ipsec.d/ipsec.conf
#ipsec连接名称
conn ipsec1

#phase1
#共享密钥
  authby=secret
#start表示ipsec服务启动后会主动发起IPSEC连接;add只表示添加，服务启动不会发起连接,使用ipsec auto --up ipsec名称发起连接
  auto=start
#阶段1密钥集
  ike=aes128-sha1;modp1536
#ike密钥交换方式
  keyexchange=ike
#阶段1生命周期
  ikelifetime=86400s
#默认为no主模式，野蛮模式为yes
  aggrmode=no

#phase2
#段2传输格式
  phase2=esp
#阶段2密钥集
  phase2alg=aes128-sha1;modp1536
#关闭压缩
  compress=no
#开启PFS
  pfs=yes
#阶段2生命周期
  salifetime=3600s
#隧道模式
```

```
  type=tunnel
#本地IP
  left=200.1.1.2
#本地子网
  leftsubnet=192.168.1.0/24
#远端VPN网关IP
  right=100.1.1.2
#远端子网
  rightsubnet=192.168.0.0/24
#远端路由按缺省配置
  rightnexthop=%defaultroute
#开启dpd检查，每10s发起一次dpd检查，30s没有收到dpd响应，则清楚该ipsec连接
  dpddelay=10
  dpdtimeout=30
  dpdaction=clear
```

启动 IPSEC 服务

```
systemctl start ipsec
systemctl enable ipsec
```

通过 ipsec verify 进行配置项校验。如果回显信息全部为 OK 时，表示配置成功。

```
# ipsec verify
Verifying installed system and configuration files
Version check and ipsec on-path                         [OK]
Libreswan 3.25 (netkey) on 3.10.0-1160.49.1.el7.x86_64
Checking for IPsec support in kernel                    [OK]
 NETKEY: Testing XFRM related proc values
         ICMP default/send_redirects                    [OK]
         ICMP default/accept_redirects                  [OK]
         XFRM larval drop                               [OK]
Pluto ipsec.conf syntax                                 [OK]
Two or more interfaces found, checking IP forwarding    [OK]
Checking rp_filter                                      [OK]
Checking that pluto is running                          [OK]
 Pluto listening for IKE on udp 500                     [OK]
 Pluto listening for IKE/NAT-T on udp 4500              [OK]
 Pluto ipsec.secret syntax                              [OK]
Checking 'ip' command                                   [OK]
Checking 'iptables' command                             [OK]
Checking 'prelink' command does not interfere with FIPS [OK]
Checking for obsolete ipsec.conf options                [OK]
```

如果报如下错误：

```
Checking rp_filter                                      [ENABLED]
 /proc/sys/net/ipv4/conf/all/rp_filter                  [ENABLED]
 /proc/sys/net/ipv4/conf/ens192/rp_filter               [ENABLED]
 /proc/sys/net/ipv4/conf/ens224/rp_filter               [ENABLED]
 /proc/sys/net/ipv4/conf/ens256/rp_filter               [ENABLED]
 /proc/sys/net/ipv4/conf/ip_vti0/rp_filter              [ENABLED]
```

则通过如下命令解决：

```
 关闭源路由检查
 echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/ens192/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/ens224/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/ens256/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/ip_vti0/rp_filter
```

查看 IPSEC 状态

1.查看 FortiGate IPSEC 状态




```
# diagnose vpn ike gateway  list

vd: root/0
name: to-openswan
version: 1
interface: port9 38
addr: 100.1.1.2:500 → 200.1.1.2:500
tun_id: 200.1.1.2/::200.1.1.2
remote_location: 0.0.0.0
network-id: 0
created: 19s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 50/50/50 ms
  id/spi: 137 d69345dd761e35e1/0f7cdc8405b9cb8d
  direction: responder
  status: established 19-19s ago = 10ms
  proposal: aes128-sha1
  key: 82d396f4422ed9b5-2031250ed717f7f3
  lifetime/rekey: 86400/86110
  DPD sent/recv: 00000000/00003e85

# diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=to-openswan ver=1 serial=2 100.1.1.2:0→200.1.1.2:0 tun_id=200.1.1.2 tun_id6=::200.1.1.2 dst_mtu=1500 dpd-link=on
weight=1
bound_if=38 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=7 olast=7 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=10000ms retry=3 count=0 seqno=158
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to-openswan proto=0 sa=1 ref=2 serial=3
  src: 0:192.168.0.0-192.168.0.255:0
  dst: 0:192.168.1.0-192.168.1.255:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1438 expire=3312/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3330/3600
  dec: spi=dc629039 esp=aes key=16 257e01bac6998199e46bb81fcfea8ea7
       ah=sha1 key=20 ae77b59ee36fc4412db7c998b00513655ce05e44
  enc: spi=2d0f59d9 esp=aes key=16 3abe273778a50dd2e45708eaca1af809
       ah=sha1 key=20 5710a515c5a7d1a4bb857e1b62bce8a7ed8f0622
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=200.1.1.2 npu_lgwy=100.1.1.2 npu_selid=3 dec_npuid=0 enc_npuid=0
run_tally=0
```

2.查看 openswan IPSEC 状态

```
# ipsec status
000 using kernel interface: netkey
000 interface lo/lo ::1@500
000 interface ens192/ens192 2022::231@500
000 interface ens224/ens224 240e:604:109:39::216@500
000 interface lo/lo 127.0.0.1@4500
000 interface lo/lo 127.0.0.1@500
000 interface ens192/ens192 192.168.88.231@4500
000 interface ens192/ens192 192.168.88.231@500
000 interface ens224/ens224 200.1.1.2@4500
000 interface ens224/ens224 200.1.1.2@500
000 interface ens256/ens256 192.168.1.1@4500
000 interface ens256/ens256 192.168.1.1@500
000
000
```

```
000 fips mode=disabled;
000 SElinux=disabled
000 seccomp=disabled
000
000 config setup options:
000
000 configdir=/etc, configfile=/etc/ipsec.conf, secrets=/etc/ipsec.secrets, ipsecdir=/etc/ipsec.d
000 nssdir=/etc/ipsec.d, dumpdir=/run/pluto, statsbin=unset
000 dnssec-rootkey-file=/var/lib/unbound/root.key, dnssec-trusted=<unset>
000 sbindir=/usr/sbin, libexecdir=/usr/libexec/ipsec
000 pluto_version=3.25, pluto_vendorid=OE-Libreswan-3.25
000 nhelpers=-1, uniqueids=yes, dnssec-enable=yes, perpeerlog=no, logappend=yes, logip=yes, shuntlifetime=900s,
xfrmlifetime=300s
000 ddos-cookies-threshold=50000, ddos-max-halfopen=25000, ddos-mode=auto
000 ikeport=500, ikebuf=0, msg_errqueue=yes, strictcrlpolicy=no, crlcheckinterval=0, listen=<any>, nflog-all=0
000 ocsp-enable=no, ocsp-strict=no, ocsp-timeout=2, ocsp-uri=<unset>
000 ocsp-trust-name=<unset>
000 ocsp-cache-size=1000, ocsp-cache-min-age=3600, ocsp-cache-max-age=86400, ocsp-method=get
000 secctx-attr-type=32001
000 debug: control
000
000 nat-traversal=yes, keep-alive=20, nat-ikeport=4500
000 virtual-private (%priv):
000 - allowed subnets: 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 25.0.0.0/8, 100.64.0.0/10, fd00::/8, fe80::/10
000
000 ESP algorithms supported:
000
000 algorithm ESP encrypt: id=3, name=ESP_3DES, ivlen=8, keysizemin=192, keysizemax=192
000 algorithm ESP encrypt: id=6, name=ESP_CAST, ivlen=8, keysizemin=128, keysizemax=128
000 algorithm ESP encrypt: id=11, name=ESP_NULL, ivlen=0, keysizemin=0, keysizemax=0
000 algorithm ESP encrypt: id=12, name=ESP_AES, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=13, name=ESP_AES_CTR, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=14, name=ESP_AES_CCM_A, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=15, name=ESP_AES_CCM_B, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=16, name=ESP_AES_CCM_C, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=18, name=ESP_AES_GCM_A, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=19, name=ESP_AES_GCM_B, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=20, name=ESP_AES_GCM_C, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=22, name=ESP_CAMELLIA, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=23, name=ESP_NULL_AUTH_AES_GMAC, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=252, name=ESP_SERPENT, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm ESP encrypt: id=253, name=ESP_TWOFISH, ivlen=8, keysizemin=128, keysizemax=256
000 algorithm AH/ESP auth: id=1, name=AUTH_ALGORITHM_HMAC_MD5, keysizemin=128, keysizemax=128
000 algorithm AH/ESP auth: id=2, name=AUTH_ALGORITHM_HMAC_SHA1, keysizemin=160, keysizemax=160
000 algorithm AH/ESP auth: id=5, name=AUTH_ALGORITHM_HMAC_SHA2_256, keysizemin=256, keysizemax=256
000 algorithm AH/ESP auth: id=6, name=AUTH_ALGORITHM_HMAC_SHA2_384, keysizemin=384, keysizemax=384
000 algorithm AH/ESP auth: id=7, name=AUTH_ALGORITHM_HMAC_SHA2_512, keysizemin=512, keysizemax=512
000 algorithm AH/ESP auth: id=8, name=AUTH_ALGORITHM_HMAC_RIPEMD, keysizemin=160, keysizemax=160
000 algorithm AH/ESP auth: id=9, name=AUTH_ALGORITHM_AES_XCBC, keysizemin=128, keysizemax=128
000 algorithm AH/ESP auth: id=250, name=AUTH_ALGORITHM_AES_CMAC_96, keysizemin=128, keysizemax=128
000 algorithm AH/ESP auth: id=251, name=AUTH_ALGORITHM_NULL_KAME, keysizemin=0, keysizemax=0
000
000 IKE algorithms supported:
000
000 algorithm IKE encrypt: v1id=5, v1name=OAKLEY_3DES_CBC, v2id=3, v2name=3DES, blocksize=8, keydeflen=192
000 algorithm IKE encrypt: v1id=8, v1name=OAKLEY_CAMELLIA_CBC, v2id=23, v2name=CAMELLIA_CBC, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=-1, v1name=n/a, v2id=20, v2name=AES_GCM_C, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=-1, v1name=n/a, v2id=19, v2name=AES_GCM_B, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=-1, v1name=n/a, v2id=18, v2name=AES_GCM_A, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=13, v1name=OAKLEY_AES_CTR, v2id=13, v2name=AES_CTR, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=7, v1name=OAKLEY_AES_CBC, v2id=12, v2name=AES_CBC, blocksize=16, keydeflen=128
000 algorithm IKE encrypt: v1id=65004, v1name=OAKLEY_SERPENT_CBC, v2id=65004, v2name=SERPENT_CBC, blocksize=16,
keydeflen=128
000 algorithm IKE encrypt: v1id=65005, v1name=OAKLEY_TWOFISH_CBC, v2id=65005, v2name=TWOFISH_CBC, blocksize=16,
keydeflen=128
000 algorithm IKE encrypt: v1id=65289, v1name=OAKLEY_TWOFISH_CBC_SSH, v2id=65289, v2name=TWOFISH_CBC_SSH, blocksize=16,
keydeflen=128
000 algorithm IKE PRF: name=HMAC_MD5, hashlen=16
000 algorithm IKE PRF: name=HMAC_SHA1, hashlen=20
000 algorithm IKE PRF: name=HMAC_SHA2_256, hashlen=32
000 algorithm IKE PRF: name=HMAC_SHA2_384, hashlen=48
000 algorithm IKE PRF: name=HMAC_SHA2_512, hashlen=64
```

```
000 algorithm IKE PRF: name=AES_XCBC, hashlen=16
000 algorithm IKE DH Key Exchange: name=MODP1024, bits=1024
000 algorithm IKE DH Key Exchange: name=MODP1536, bits=1536
000 algorithm IKE DH Key Exchange: name=MODP2048, bits=2048
000 algorithm IKE DH Key Exchange: name=MODP3072, bits=3072
000 algorithm IKE DH Key Exchange: name=MODP4096, bits=4096
000 algorithm IKE DH Key Exchange: name=MODP6144, bits=6144
000 algorithm IKE DH Key Exchange: name=MODP8192, bits=8192
000 algorithm IKE DH Key Exchange: name=DH19, bits=512
000 algorithm IKE DH Key Exchange: name=DH20, bits=768
000 algorithm IKE DH Key Exchange: name=DH21, bits=1056
000 algorithm IKE DH Key Exchange: name=DH22, bits=1024
000 algorithm IKE DH Key Exchange: name=DH23, bits=2048
000 algorithm IKE DH Key Exchange: name=DH24, bits=2048
000
000 stats db_ops: {curr_cnt, total_cnt, maxsz} :context={0,3,64} trans={0,3,6936} attrs={0,3,4624}
000
000 Connection list:
000
000 "ipsec1": 192.168.1.0/24===200.1.1.2<200.1.1.2>...100.1.1.2<100.1.1.2>===192.168.0.0/24; erouted; eroute owner: #3
000 "ipsec1":     oriented; my_ip=unset; their_ip=unset; my_updown=ipsec _updown;
000 "ipsec1":   xauth us:none, xauth them:none,  my_username=[any]; their_username=[any]
000 "ipsec1":   our auth:secret, their auth:secret
000 "ipsec1":   modecfg info: us:none, them:none, modecfg policy:push, dns:unset, domains:unset, banner:unset, cat:unset;
000 "ipsec1":   labeled_ipsec:no;
000 "ipsec1":   policy_label:unset;
000 "ipsec1":   ike_life: 86400s; ipsec_life: 3600s; replay_window: 32; rekey_margin: 540s; rekey_fuzz: 100%; keyingtries:
0;
000 "ipsec1":   retransmit-interval: 500ms; retransmit-timeout: 60s;
000 "ipsec1":   initial-contact:no; cisco-unity:no; fake-strongswan:no; send-vendorid:no; send-no-esp-tfc:no;
000 "ipsec1":   policy: PSK+ENCRYPT+TUNNEL+PFS+UP+IKEV1_ALLOW+IKEV2_ALLOW+SAREF_TRACK+IKE_FRAG_ALLOW+ESN_NO;
000 "ipsec1":   conn_prio: 24,24; interface: ens224; metric: 0; mtu: unset; sa_prio:auto; sa_tfc:none;
000 "ipsec1":   nflog-group: unset; mark: unset; vti-iface:unset; vti-routing:no; vti-shared:no; nic-offload:auto;
000 "ipsec1":   our idtype: ID_IPV4_ADDR; our id=200.1.1.2; their idtype: ID_IPV4_ADDR; their id=100.1.1.2
000 "ipsec1":   dpd: action:clear; delay:10; timeout:30; nat-t: encaps:auto; nat_keepalive:yes; ikev1_natt:both
000 "ipsec1":   newest ISAKMP SA: #2; newest IPsec SA: #3;
000 "ipsec1":   IKE algorithms: AES_CBC_128-HMAC_SHA1-MODP1536
000 "ipsec1":   IKE algorithm newest: AES_CBC_128-HMAC_SHA1-MODP1536
000 "ipsec1":   ESP algorithms: AES_CBC_128-HMAC_SHA1_96-MODP1536
000 "ipsec1":   ESP algorithm newest: AES_CBC_128-HMAC_SHA1_96; pfsgroup=MODP1536
000
000 Total IPsec connections: loaded 1, active 1
000
000 State Information: DDoS cookies not required, Accepting new IKE connections
000 IKE SAs: total(1), half-open(0), open(0), authenticated(1), anonymous(0)
000 IPsec SAs: total(1), authenticated(1), anonymous(0)
000
000 #2: "ipsec1":500 STATE_MAIN_I4 (ISAKMP SA established); EVENT_SA_REPLACE in 85343s; newest ISAKMP; lastdpd=3s(seq
in:16011 out:0); idle; import:admin initiate
000 #3: "ipsec1":500 STATE_QUICK_I2 (sent QI2, IPsec SA established); EVENT_SA_REPLACE in 2784s; newest IPSEC; eroute
owner; isakmp#2; idle; import:admin initiate
000 #3: "ipsec1" esp.dc629039@100.1.1.2 esp.2d0f59d9@200.1.1.2 tun.0@100.1.1.2 tun.0@200.1.1.2 ref=0 refhim=0 Traffic:
ESPin=0B ESPout=0B! ESPmax=4194303B
000
000 Bare Shunt list:
000
```

业务测试

1.从 FortiGate 端 PC ping OPENSWAN 端的 PC

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 00:0c:29:0e:4e:c5  txqueuelen 1000  (Ethernet)
        RX packets 58117456  bytes 4943397966 (4.6 GiB)
        RX errors 0  dropped 183  overruns 0  frame 0
        TX packets 3346784  bytes 205418084392 (191.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
PC1# ping 192.168.1.10 -c 4
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=62 time=1.13 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=62 time=0.852 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=62 time=0.750 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=62 time=0.775 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.750/0.877/1.132/0.153 ms
```

2.从 OPENSWAN 端的 PC ping FortiGate 端 PC

```
PC2# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6476393  bytes 1884675006 (1.7 GiB)
        RX errors 0  dropped 3749817  overruns 0  frame 0
        TX packets 184443  bytes 12303642 (11.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC2# ping 192.168.0.10 -c 4
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=62 time=0.681 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=62 time=0.715 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=62 time=0.801 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=62 time=0.771 ms

--- 192.168.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.681/0.742/0.801/0.046 ms
```

OPENSWAN debug 查看

开启 openswan debug

```
# /etc/ipsec.conf - Libreswan IPsec configuration file
# see 'man ipsec.conf' and 'man pluto' for more information
# For example configurations and documentation, see https://libreswan.org/wiki/
config setup
        # Normally, pluto logs via syslog.
        logfile=/var/log/pluto.log
        #
        # Do not enable debug options to debug configuration issues!
        #
        # plutodebug="control parsing"
        # plutodebug="all crypt"
        plutodebug=control
```

1. 阶段 1 密钥机不匹配时，openswan debug 显示如下：

```
Feb  7 17:58:48.108027: | *received 332 bytes from 100.1.1.2:500 on ens224 (port=500)
Feb  7 17:58:48.108082: | processing: start from 100.1.1.2:500 (in process_md() at demux.c:392)
Feb  7 17:58:48.108103: |  processing version=1.0 packet with exchange type=ISAKMP_XCHG_IDPROT (2)
Feb  7 17:58:48.108119: | icookie table: hash icookie fc 65 ed f1  76 0e d2 08 to 15842219016969466431 slot
0x558680eded00
Feb  7 17:58:48.108126: | v1 state object not found
Feb  7 17:58:48.108148: | received Vendor ID payload [RFC 3947]
Feb  7 17:58:48.108160: | ignoring Vendor ID payload [draft-ietf-ipsec-nat-t-ike-03]
Feb  7 17:58:48.108171: | ignoring Vendor ID payload [draft-ietf-ipsec-nat-t-ike-02]
Feb  7 17:58:48.108179: | ignoring Vendor ID payload [draft-ietf-ipsec-nat-t-ike-02_n]
Feb  7 17:58:48.108186: | ignoring Vendor ID payload [draft-ietf-ipsec-nat-t-ike-01]
Feb  7 17:58:48.108195: | ignoring Vendor ID payload [draft-ietf-ipsec-nat-t-ike-00]
Feb  7 17:58:48.108205: | received Vendor ID payload [Dead Peer Detection]
Feb  7 17:58:48.108213: | received Vendor ID payload [FRAGMENTATION]
Feb  7 17:58:48.108224: | received Vendor ID payload [FRAGMENTATION c0000000]
Feb  7 17:58:48.108237: packet from 100.1.1.2:500: ignoring unknown Vendor ID payload [8299031757a36082c6a621de00000000]
Feb  7 17:58:48.108251: | creating state object #2 at 0x558682efea98
```

```
Feb  7 17:58:48.108265: | processing: start state #2 connection "ipsec1" 100.1.1.2:500 (in main_inI1_outR1() at
ikev1_main.c:757)
Feb  7 17:58:48.108313: | inserting state object #2
Feb  7 17:58:48.108338: "ipsec1" #2: responding to Main Mode
Feb  7 17:58:48.108372: | started looking for secret for 200.1.1.2→100.1.1.2 of kind PKK_PSK
Feb  7 17:58:48.108384: | actually looking for secret for 200.1.1.2→100.1.1.2 of kind PKK_PSK
Feb  7 17:58:48.108397: | 1: compared key 100.1.1.2 to 200.1.1.2 / 100.1.1.2 → 4
Feb  7 17:58:48.108406: | 2: compared key 200.1.1.2 to 200.1.1.2 / 100.1.1.2 → 12
Feb  7 17:58:48.108413: | line 1: match=12
Feb  7 17:58:48.108422: | best_match 0>12 best=0×558682efd7c8 (line=1)
Feb  7 17:58:48.108430: | concluding with best_match=12 best=0×558682efd7c8 (lineno=1)
Feb  7 17:58:48.108441: "ipsec1" #2: WARNING: connection ipsec1 PSK length of 8 bytes is too short for sha2_256 PRF in
FIPS mode (16 bytes required)
Feb  7 17:58:48.108461: "ipsec1" #2: Oakley Transform [AES_CBC (256), HMAC_SHA2_256, MODP2048] refused
Feb  7 17:58:48.108471: | started looking for secret for 200.1.1.2→100.1.1.2 of kind PKK_PSK
Feb  7 17:58:48.108478: | actually looking for secret for 200.1.1.2→100.1.1.2 of kind PKK_PSK
Feb  7 17:58:48.108486: | 1: compared key 100.1.1.2 to 200.1.1.2 / 100.1.1.2 → 4
Feb  7 17:58:48.108495: | 2: compared key 200.1.1.2 to 200.1.1.2 / 100.1.1.2 → 12
Feb  7 17:58:48.108505: | line 1: match=12
Feb  7 17:58:48.108512: | best_match 0>12 best=0×558682efd7c8 (line=1)
Feb  7 17:58:48.108522: | concluding with best_match=12 best=0×558682efd7c8 (lineno=1)
Feb  7 17:58:48.108533: "ipsec1" #2: WARNING: connection ipsec1 PSK length of 8 bytes is too short for sha2_256 PRF in
FIPS mode (16 bytes required)
Feb  7 17:58:48.108541: "ipsec1" #2: Oakley Transform [AES_CBC (256), HMAC_SHA2_256, MODP1536] refused
Feb  7 17:58:48.108548: "ipsec1" #2: no acceptable Oakley Transform
Feb  7 17:58:48.108556: | complete v1 state transition with NO_PROPOSAL_CHOSEN
```

2. 当共享密钥错误时，openswan debug 显示如下：

```
Feb  7 17:57:28.726151: | *received 108 bytes from 100.1.1.2:500 on ens224 (port=500)
Feb  7 17:57:28.726176: | processing: start from 100.1.1.2:500 (in process_md() at demux.c:392)
Feb  7 17:57:28.726186: |  processing version=1.0 packet with exchange type=ISAKMP_XCHG_IDPROT (2)
Feb  7 17:57:28.726197: | cookies table: hash icookie 2b 08 9f f0  ed 75 b2 1a rcookie 2e fa 5e f5  8a 50 a4 93 to
1021963185563737028 4 slot 0×558680ed9da0
Feb  7 17:57:28.726217: | v1 peer and cookies match on #1, provided msgid 00000000 == 00000000
Feb  7 17:57:28.726224: | v1 state object #1 found, in STATE_MAIN_R2
Feb  7 17:57:28.726232: | processing: start state #1 connection "ipsec1" 100.1.1.2:500 (in process_v1_packet() at
ikev1.c:1117)
Feb  7 17:57:28.726270: "ipsec1" #1: byte 2 of ISAKMP Identification Payload should have been zero, but was not
(ignored)
Feb  7 17:57:28.726283: "ipsec1" #1: length of ISAKMP Identification Payload is larger than can fit
Feb  7 17:57:28.726290: "ipsec1" #1: probable authentication failure (mismatch of preshared secrets?): malformed payload
in packet
Feb  7 17:57:28.726313: | processing: stop from 100.1.1.2:500 (BACKGROUND) (in process_md() at demux.c:394)
Feb  7 17:57:28.726322: | processing: stop state #1 connection "ipsec1" 100.1.1.2:500 (in process_md() at demux.c:396)
```

与 strongSwan 建立 IPsec VPN

配置请参考链接：FortiGate 与 strongSwan 建立 IPSec VPN ↗。

IPSec VPN 排错

- IPSec 相关诊断命令
- IPSEC IKEv1 主模式 Debug 分析
- IPSec IKEv1 野蛮模式 Debug 分析
- IPSEC IKEv2 Debug 分析
- IPSEC 协商失败举例
- 使用附加 IP 时隧道无法建立
- 多个拨号隧道使用相同 IP 池
- IPSec 命名注意事项
- IPSec 业务不通问题
- ESP 分片丢包
- IPSec 无法修改隧道类型
- Hub-Spoke 双线路隧道抖动
- Hub-Spoke 静态路由问题
- Hub 反向流量异步
- IPsec 绑定 Loopback 时无法建立
- IPSec 解密

IPSec 相关诊断命令

本文介绍 FortiGate 上常用的 IPSec VPN 诊断命令，包括查看 IKE 状态摘要、Phase1/Phase2 隧道状态，以及如何使用日志过滤器进行针对性排错。

diagnose vpn ike status

该命令用于查看 IKE 守护进程的摘要信息，包括当前连接数、IKE SA 和 IPsec SA 的统计。

```bash
FortiGate # diagnose vpn ike status

connection: 2/50
IKE SA: created 2/51  established 2/9  times 0/13/40 ms
IPsec SA: created 1/13  established 1/7  times 0/8/30 ms
```

字段	含义
`connection: 2/50`	当前活跃连接数/历史累计连接数
`IKE SA: created 2/51`	当前已创建的 IKE SA 数/历史累计创建数
`established 2/9`	当前已建立的 IKE SA 数/历史累计建立数
`times 0/13/40 ms`	IKE SA 建立耗时：最小值/平均值/最大值（毫秒）
`IPsec SA: created 1/13`	当前已创建的 IPsec SA 数/历史累计创建数
`established 1/7`	当前已建立的 IPsec SA 数/历史累计建立数
`times 0/8/30 ms`	IPsec SA 建立耗时：最小值/平均值/最大值（毫秒）

diagnose vpn ike gateway list

该命令用于查看 IPsec Phase1（IKE SA）的详细状态，对应 VPN 向导中的 Phase1 接口配置。

```bash
# 查看所有 Phase1 网关状态
diagnose vpn ike gateway list

# 查看指定 Phase1 的网关状态
diagnose vpn ike gateway list name <phase1-name>
```

输出示例：

```bash
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: to_Hub
version: 2
interface: wan1 5
addr: 172.22.6.64:4500 → 123.112.245.136:4500
tun_id: 10.0.0.1/::10.0.0.1
remote_location: 0.0.0.0
network-id: 1
transport: UDP
created: 1613s ago
peer-id: 192.168.123.175
peer-id-auth: no
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 30/30/30 ms
IPsec SA: created 1/1  established 1/1  time 30/30/30 ms

  id/spi: 280 b3e059d8b13c10b4/0282c25b49016f77
```

```
  direction: initiator
  status: established 1613-1613s ago = 30ms
  proposal: aes128-sha256
  child: no
  SK_ei: 563a54d670e8b165-f9c32703918e1298
  SK_er: 8e22948050818d3f-d2492b127bf87596
  SK_ai: f2852d83e816f6c1-00a57a7dee98a9f2-ff7f6ca67414b4ba-3bb9e10545d4d001
  SK_ar: 2c68aa001decc746-c22fb11c6cbe0acd-0fbd93f99d8771b1-b6d2f846405d4463
  PPK: no
  message-id sent/recv: 219/0
  QKD: no
  lifetime/rekey: 86400/84486
  DPD sent/recv: 00021ff7/00021ff7
  peer-id: 192.168.123.175
```

字段	含义
`vd`	所属虚拟域（VDOM）及索引号
`name`	Phase1 接口名称
`version`	IKE 版本号（1 或 2）
`interface`	绑定的物理接口及索引
`addr`	本端 IP: 端口 -> 对端 IP: 端口
`tun_id`	隧道标识（默认为 Remote Gateway IP）
`direction`	角色：initiator（发起方）或 responder（响应方）
`status`	SA 当前状态及存活时间
`proposal`	协商使用的加密/认证算法组合
`lifetime/rekey`	SA 生命周期/重协商时间（秒）
`DPD`	DPD 检测配置（模式、空闲超时、重试次数等）

diagnose vpn tunnel list

该命令用于查看 IPsec Phase2（IPsec SA）的详细状态，包括加密算法、流量选择器（Selector）、数据传输统计等。

```bash
# 查看所有 Phase2 隧道
diagnose vpn tunnel list

# 查看指定隧道名称的 Phase2
diagnose vpn tunnel list name <phase2-name>
```

输出示例：

```bash
FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=dhcp ver=2 serial=2 10.10.12.1:0->0.0.0.0:0 nexthop= tun_id=10.0.0.2 tun_id6=::10.0.0.2 status=up dst_mtu=0 weight=1
bound_if=12 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary
accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=0 refcnt=2 ilast=44589822 olast=44589822 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=to_Hub ver=2 serial=1 172.22.6.64:4500->123.112.245.136:4500 nexthop=172.22.6.1 tun_id=10.0.0.1 tun_id6=::10.0.0.1
status=up dst_mtu=1500 weight=1
bound_if=5 real_if=5 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=sync-
```

```
primary accept_traffic=1 overlay_id=1

proxyid_num=1 child_num=0 refcnt=5 ilast=3 olast=3 ad=/0
stat: rxp=64 txp=1093 rxb=9762 txb=4955
dpd: mode=on-idle on=1 status=ok idle=5000ms retry=3 count=0 seqno=139271
natt: mode=keepalive draft=0 interval=10 remote_port=4500
fec: egress=0 ingress=0
proxyid=to_Home_WAN1 proto=0 sa=1 ref=3 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=10224 type=00 soft=0 mtu=1422 expire=41170/0B replaywin=0
       seqno=444 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42899/43200
  dec: spi=3ad09099 esp=aes key=32 7aa05571013eb92215965d8fefc41eb3a5a8f49143dcbcf19780184cdf9e266e
       ah=sha256 key=32 70ee0bdd2aecad1179137c570b9dddd8761c9ac7403c47f60186fdc699684605
  enc: spi=afeaa692 esp=aes key=32 757de8c4d6c5650bd892208dd98e88d78e930858f95daabaa6cf35f290ef8dd7
       ah=sha256 key=32 b33303fca9192130b15b669fd8cd222bf515c58732e2070a6e4312ca8369b139
  dec:pkts/bytes=65/9814, enc:pkts/bytes=1159/14195
  npu_flag=03 npu_lgwy=172.22.6.64:4500 npu_rgwy=123.112.245.136:4500
npu_selid=0 dec_npuid=1 enc_npuid=1 npu_isaidx=40 npu_osaidx=4
```

字段	含义
`name`	Phase2 名称
`ver`	IKE 版本
`bound_if`	绑定的接口索引
`mode`	隧道模式
`options`	NPU 加速等选项标志
`stat: rxp/txp`	接收/发送的数据包数量
`stat: rxb/txb`	接收/发送的字节数
`dpd`	DPD 配置及状态
`natt`	NAT-T（NAT 穿越）配置
`proxyid`	流量选择器（代理 ID）
`src` / `dst`	源/目标子网范围（本端/对端感兴趣流）
`sa`	IPsec SA 数量（0 = 未建立，1 = 已建立）
`dec: spi` / `enc: spi`	解密/加密的 SPI 值及算法
`mtu`	隧道有效 MTU
`expire`	SA 剩余过期时间（秒）
`replaywin`	重放攻击检测窗口大小
`npu_flag`	NPU（硬件加速）标志及参数

diagnose vpn ipsec status

该命令用于查看所有 IPsec 加密设备（Crypto Device）的加密/解密计数器统计信息，帮助确认数据是否正确通过 VPN 硬件/软件加密通道传输。

```bash
FortiGate # diagnose vpn ipsec status
All ipsec crypto devices in use:
np7_0:
    Encryption (encrypted/decrypted)
        null             : 0                0
        des              : 0                0
```

```
        3des              : 0               0
        aes               : 9359            6696
        aes-gcm           : 0               0
        aria              : 0               0
        seed              : 0               0
        chacha20poly1305  : 0               0
    Integrity (generated/validated)
        null              : 0               0
        md5               : 0               0
        sha1              : 9359            6696
```

输出中可能出现的加密设备包括：`NP7_0`、`NP7_1`（NPU 硬件加速）、`CP8` / `CP9`（内容处理器）、`NPU Host Offloading` 以及 `SOFTWARE`（软件加密）。

diagnose vpn ike log filter

当设备上存在多条 IPsec 隧道时，Debug 输出可能非常庞大。使用 IKE 日志过滤器可以只显示与特定隧道或对端相关的 Debug 信息，大大简化排错过程。

> **命令语法变更**
>
> 从 FortiOS v7.4.1 开始，`diagnose vpn ike log-filter` 的语法已变更为 `diagnose vpn ike log filter`（去掉了连字符）。同时部分子命令也有变化，例如 `src-addr4` 更名为 `loc-addr4`，`dst-addr4` 更名为 `rem-addr4`。请根据实际固件版本使用对应语法。

查看与清除过滤器

查看当前过滤器设置：

```bash
FortiGate # diagnose vpn ike log filter list
        vd: any
        name: any
        interface index: any
        IPv4 local gw: 10.10.12.2
        multiple IPv4 local gw: any
        IPv4 remote gw: 1.1.1.1
        multiple IPv4 remote gw: any
        IPv6 local gw: any
        multiple IPv6 local gw: any
        IPv6 remote gw: any
        multiple IPv6 remote gw: any
        dest port: any
```

清除所有过滤条件：

```bash
diagnose vpn ike log filter clear
```

过滤选项

子命令	说明
`list`	显示当前过滤器设置
`clear`	清除所有过滤条件
`name <phase1-name>`	按 Phase1 名称过滤
`interface <index>`	按接口索引过滤（0 表示匹配所有接口）
`vd <index>`	按 VDOM 索引过滤（-1 表示匹配所有 VDOM）
`loc-addr4 <IP>`	按本端 IPv4 网关地址过滤（v7.4.1+ 语法，旧版为 `src-addr4`）
`rem-addr4 <IP>`	按对端 IPv4 网关地址过滤（v7.4.1+ 语法，旧版为 `dst-addr4`）
`loc-addr6 <IPv6>`	按本端 IPv6 网关地址过滤
`rem-addr6 <IPv6>`	按对端 IPv6 网关地址过滤
`mloc-addr4 <IP1> [IP2] ...`	按多个本端 IPv4 地址过滤（最多 6 个）
`mrem-addr4 <IP1> [IP2] ...`	按多个对端 IPv4 地址过滤（最多 6 个）
`dst-port <port>`	按目标端口过滤
`negate <filter-name>`	取反指定的过滤条件（例如 `negate dst-port`）

常用过滤示例

- 按对端网关 IP 过滤：

```bash
diagnose vpn ike log filter rem-addr4 198.51.100.1
```

- 按 Phase1 名称过滤：

```bash
diagnose vpn ike log filter name to_Hub
```

- 按本端网关 IP 过滤：

```bash
diagnose vpn ike log filter loc-addr4 203.0.113.1
```

- 按接口索引过滤：

```bash
diagnose vpn ike log filter interface 10
```

- 同时过滤多个对端地址：

```bash
diagnose vpn ike log filter mrem-addr4 198.51.100.1 198.51.100.2 198.51.100.3
```

IKE Debug

> **注意**
>
> Debug 输出会消耗系统资源，排错完成后务必及时关闭。在生产环境中，务必先配置好过滤器再开启 Debug，避免大量无关日志输出影响设备性能。

开启 Debug

```bash
# 第一步：清除旧的过滤条件
diagnose vpn ike log filter clear

# 第二步：设置过滤器（推荐按对端 IP 或隧道名称过滤）
diagnose vpn ike log filter rem-addr4 <对端IP>

# 第三步：开启 IKE debug
diagnose debug application ike -1

# 第四步：启用时间戳，便于与 VPN 事件日志交叉对比
diagnose debug console timestamp enable

# 第五步：开启 debug 输出
diagnose debug enable
```

关闭 Debug

```bash
# 关闭 debug 输出
diagnose debug disable

# 重置所有 debug 应用设置
diagnose debug reset

# 清除过滤条件
diagnose vpn ike log filter clear
```

其他命令

查看隧道摘要

以简洁的方式列出所有 IPsec 隧道的 UP/DOWN 状态和流量统计。

```bash
FortiGate # get vpn ipsec tunnel summary
'to_Hub' 123.112.245.136:4500  selectors(total,up): 1/1  rx(pkt,err): 47/0  tx(pkt,err): 1074/0
```

查看隧道详情

以详细的格式列出隧道信息，包括 Phase1/Phase2 参数。显示完整 Phase2 信息，包括本端/远端网关、选择器、SPI 值、加密算法和 NPU 加速状态。

```
# 查看所有隧道详情
get vpn ipsec tunnel details

# 查看指定隧道详情
get vpn ipsec tunnel name <tunnel-name>
```

输出示例：

```bash
FortiGate # get vpn ipsec tunnel details

gateway
  name: 'to_Hub'
  local-gateway: 172.22.6.64:4500 (static)
  remote-gateway: 123.112.245.136:4500 (static)
  status: up
  mode: ike-v2
  interface: 'wan1' (5) vrf:0
  rx  packets: 50  bytes: 7606  errors: 0
  tx  packets: 1078  bytes: 3827  errors: 0
  dpd: on-idle/negotiated status:ok idle: 5000ms  retry: 3  count: 0
  nat traversal mode: keep-alive      interval: 10
  selectors
```

```
    name: 'to_Hub'
    auto-negotiate: disable
    mode: tunnel
    src: 0:0.0.0.0/0.0.0.0:0
    dst: 0:0.0.0.0/0.0.0.0:0
    SA
      lifetime/rekey: 43200/41562
      mtu: 1422
      tx-esp-seq: 435
      replay: disabled
      qat: 0
      inbound
        spi: 3ad09099
        enc:  aes-cb  7aa05571013eb92215965d8fefc41eb3a5a8f49143dcbcf19780184cdf9e266e
        auth: sha256  70ee0bdd2aecad1179137c570b9dddd8761c9ac7403c47f60186fdc699684605
      outbound
        spi: afeaa692
        enc:  aes-cb  757de8c4d6c5650bd892208dd98e88d78e930858f95daabaa6cf35f290ef8dd7
        auth: sha256  b33303fca9192130b15b669fd8cd222bf515c58732e2070a6e4312ca8369b139
      NPU acceleration: encryption(outbound) decryption(inbound)
```

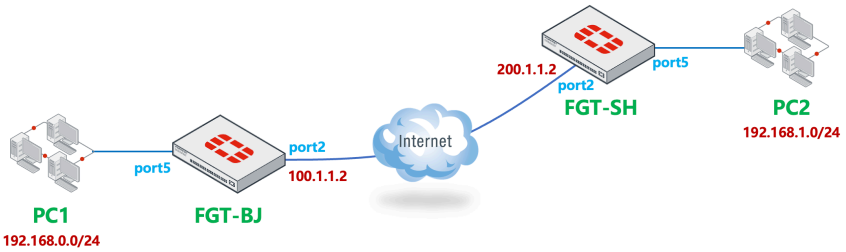
重启隧道

> **注意**
>
> 该命令会断开并重新协商指定隧道，生产环境请谨慎使用。

```bash
# 重启指定隧道
diagnose vpn ike gateway flush name <phase1-name>

# 重启所有隧道（谨慎执行）
diagnose vpn ike gateway flush
```

IPSEC IKEv1 主模式 Debug 分析

网络拓扑



IPSEC Debug 分析

FGT - SH 发起协商，FGT - BJ 进行 Debug 查看，下面是 FGT - BJ 的 debug 输出

```
# diagnose debug application ike -1
# diagnose debug  enable

收到发起方的数据包，主模式第1个包
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=2f013665b867de80/0000000000000000 len=572 vrf=0
ike 0: in
2F013665B867DE800000000000000000011002000000000000000023C0D000154000000010000000100000148010100080300002801010000800B000100
0C000400015180800100078 00E0080800300018002000480040 00E0300002802010000800B0001000C000400015180800100078 00E0080800300018002
000480040005030000280301000080 0B0001000C000400015180800100078 00E0100800300018002000480040 00E0300002804010000800B0001000C00
0400015180800100078 00E0100800300018002000480040005030000280501000080 0B0001000C000400015180800100078 00E0080800300018002000 2
8004000E0300002806010000800B0001000C000400015180800100078 00E0080800300018002000280040005030000280701000080 0B0001000C000400
015180800100078 00E0100800300018002000280040 00E0000002808010000800B0001000C000400015180800100078 00E0100800300018002000280 04
00050D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C179D9215529D560D000014CD60464335DF21F87CFDB2FC68B6A4
480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D83821A0F0AEAA8620D0000144485152D18B6BBCD0BE8A8469579DDCC
0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE7F00D6C2D30D0000184048B7D56EBCE88525E7DE7F00D6C2D3C0
0000000000000148299031757A36082C6A621DE00000000
ike 0:2f013665b867de80/0000000000000000:2: responder: main mode get 1st message...
ike 0:2f013665b867de80/0000000000000000:2: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:2f013665b867de80/0000000000000000:2: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:2f013665b867de80/0000000000000000:2: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:2f013665b867de80/0000000000000000:2: VID draft-ietf-ipsec-nat-t-ike-02\n 90CB80913EBB696E086381B5EC427B1F
ike 0:2f013665b867de80/0000000000000000:2: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:2f013665b867de80/0000000000000000:2: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:2f013665b867de80/0000000000000000:2: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:2f013665b867de80/0000000000000000:2: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:2f013665b867de80/0000000000000000:2: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:2f013665b867de80/0000000000000000:2: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done

VPN-to-SH有匹配的proposal
ike 0:2f013665b867de80/0000000000000000:2: negotiation result
ike 0:2f013665b867de80/0000000000000000:2: proposal id = 1:
ike 0:2f013665b867de80/0000000000000000:2:   protocol id = ISAKMP:
ike 0:2f013665b867de80/0000000000000000:2:      trans_id = KEY_IKE.
ike 0:2f013665b867de80/0000000000000000:2:      encapsulation = IKE/none
ike 0:2f013665b867de80/0000000000000000:2:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:2f013665b867de80/0000000000000000:2:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:2f013665b867de80/0000000000000000:2:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:2f013665b867de80/0000000000000000:2:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:2f013665b867de80/0000000000000000:2: ISAKMP SA lifetime=86400
ike 0:2f013665b867de80/0000000000000000:2: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0×8272220 10 100.1.1.2→200.1.1.2:500.
ike 0:VPN-to-SH:2: DPD negotiated
ike 0:VPN-to-SH:2: peer is FortiGate/FortiOS (v0 b0)
ike 0:VPN-to-SH:2: selected NAT-T version: RFC 3947
ike 0:VPN-to-SH:2: cookie 2f013665b867de80/4878dfc8fee0cc14
```

响应方回复的数据包，主模式第2个包，回复选定的proposal

```
ike 0:VPN-to-SH:2: out
2F013665B867DE804878DFC8FEE0CC1401100200000000000000000C00D00003C0000000100000001000000300101000100000028010100000800B0001000C000400015180800100078 00E00808003000180020004800 4000E0D0000144A131C81070358455C5728F20E95452F0D000014AFCAD71368A1F1C96B8696FC775701000D0000148299031757A36082C6A621DE000000000D0000144048B7D56EBCE88525E7DE7F00D6C2D3000000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:VPN-to-SH:2: sent IKE msg (ident_r1send): 100.1.1.2:500→200.1.1.2:500, len=192, vrf=0,
id=2f013665b867de80/4878dfc8fee0cc14
```

收到发起方的数据包，主模式第3个包，DH秘钥交换

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=2f013665b867de80/4878dfc8fee0cc14 len=380 vrf=0
ike 0: in
2F013665B867DE804878DFC8FEE0CC1404100200000000000000017C0A0001045369DFD050FE4B83832486EB52226B86D1577173ACAF314808BCCC85503E79B107B6A4A84FF35AAC33CD21A34B92B0F5DB386B7C1CA6EB8CA960C9F10BF634422EFB65A1F10C990178FF50F399D3F7C100739FD07180370CD8E8F453C3820719E6E1A93B05A82547D9C006171B4662D5F6957AA4EAF646E951149C7932F857FCFF65C3A08C33BAB87CCA8EFFEB772C3173FA598322DF3985D685274538422F7F6A8DEE06E7DD82D20903CA20351BA0ED72759E472228CD1F3CA1B4B3EFAE928024A9EEDD2518F558B4642DBD8F6104C3C575D60EC18C42708D5BD62139AD4BEACA3EFAEF9C2D20F1E08ACC6383B5BBA423372457D1276AEB92EA44A578B12F0314000014D72D6C2B0293D53F5100169A9B9FCD2C140000243D49DDDC234A0ECB8B6FB311024C90A47BA01F6F651CB35C0EFAD8EA92C118660000002486DF949862FBE5DACC6A16AF1D52082976C458B245B28165A65B4A6C8047F396
ike 0:VPN-to-SH:2: responder:main mode get 2nd message...
ike 0:VPN-to-SH:2: received NAT-D payload type 20
ike 0:VPN-to-SH:2: received NAT-D payload type 20
ike 0:VPN-to-SH:2: NAT not detected
ike 0:VPN-to-SH:2: generate DH public value request queued
ike 0:VPN-to-SH:2: compute DH shared secret request queued
```

响应方回复的数据包，主模式第4个包，DH秘钥交换

```
ike 0:VPN-to-SH:2: out
2F013665B867DE804878DFC8FEE0CC1404100200000000000000017C0A000104C7C1AE93B18487170D87BCDE7C5CADB6EB6ECEAB1C79D43B51793B52005A64A8D481779209039910A037E5EC9E8601AD675F6FE02A9B17C534A68DF48B0290C898E47A0A5E64660501E1FB35BB995E337DC257B14F89B09CC5708EF7522767D911B69A99D20EAC47A1A95CF8E21887BEFCB2CEB0952D1F193108454325C82F43311DF030E4F48F830007A19C77AA9B77EF98834B427DE9E1136EDF6C4F9FF48D5772C165E105B6278DBCBC220796E509FC0397DEBC694B7E975307B5B4505C9AC48F73E26EA60DB5AD9705D9830347667D05BC59CE6E3662CA054EB1A01F9CB075CB3BCC292B982E931042EC6F6692BF624215179E6663F3B319E98E56CD59EA1400001440B940F425F78C908561DB60981F79C71400002486DF949862FBE5DACC6A16AF1D52082976C458B245B28165A65B4A6C8047F396000000243D49DDDC234A0ECB8B6FB311024C90A47BA01F6F651CB35C0EFAD8EA92C11866
ike 0:VPN-to-SH:2: sent IKE msg (ident_r2send): 100.1.1.2:500→200.1.1.2:500, len=380, vrf=0,
id=2f013665b867de80/4878dfc8fee0cc14
ike 0:VPN-to-SH:2: ISAKMP SA 2f013665b867de80/4878dfc8fee0cc14 key 16:161F81D2B495911EF63B98D14AD53392
```

收到发起方的数据包，主模式第5个数据包

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=2f013665b867de80/4878dfc8fee0cc14 len=108 vrf=0
ike 0: in
2F013665B867DE804878DFC8FEE0CC1405100201000000000000006C300DBBB94A52BF9EBD32DAD36637AF3D01E12599D17FA31C11C0E84769FF94BBB2E2B495567DA5BC9742A711CE425345CA3172FC40E6F780C6D7B5BA7C1335289E087ADE06F7BE1D7049A827E902B778
ike 0:VPN-to-SH:2: responder: main mode get 3rd message...
ike 0:VPN-to-SH:2: dec
2F013665B867DE804878DFC8FEE0CC1405100201000000000000006C0800000C01000000C80101020B00002428BFF5F8227E569BC4138008FEFD2446057302EEE6662B5CED771411A71FFB930000001C00000001011060022F013665B867DE804878DFC8FEE0CC14FE5CDC03
ike 0:VPN-to-SH:2: received p1 notify type INITIAL-CONTACT
```

主模式使用IPV4地址作为身份ID，预共享秘钥认证成功

```
ike 0:VPN-to-SH:2: peer identifier IPV4_ADDR 200.1.1.2
ike 0:VPN-to-SH:2: PSK authentication succeeded
ike 0:VPN-to-SH:2: authentication OK
ike 0:VPN-to-SH:2: enc
2F013665B867DE804878DFC8FEE0CC1405100201000000000000004C0800000C010000006401010200000024 1328931A3FDD6ABABDF48A93D4C3937D6151EA202121C6FAEAE55698A6F85A57
```

响应方回复的数据包，主模式第6个包，身份信息和认证信息确认

```
ike 0:VPN-to-SH:2: out
2F013665B867DE804878DFC8FEE0CC1405100201000000000000005C74BC93E1AB30646D580EC9F228CEAFA928ADC9EC53A454D6C7E3FA36CB5BFCF1EEE6F13590EC2BFDA2299E25C9815B2AF528D0E8A6F441D22D4D5B70F98E3BA8
ike 0:VPN-to-SH:2: sent IKE msg (ident_r3send): 100.1.1.2:500→200.1.1.2:500, len=92, vrf=0,
id=2f013665b867de80/4878dfc8fee0cc14
```

第一阶段建立成功，生成IKE SA

```
ike 0:VPN-to-SH:2: established IKE SA 2f013665b867de80/4878dfc8fee0cc14
ike 0:VPN-to-SH:2: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike 0:VPN-to-SH:2: processing INITIAL-CONTACT
ike 0:VPN-to-SH: flushing
ike 0:VPN-to-SH: flushed
ike 0:VPN-to-SH:2: processed INITIAL-CONTACT
ike 0:VPN-to-SH: set oper up
ike 0:VPN-to-SH:2: no pending Quick-Mode negotiations
```

收到发起方的数据包，快速模式第1个包

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=2f013665b867de80/4878dfc8fee0cc14:31008365 len=620 vrf=0
ike 0: in
2F013665B867DE804878DFC8FEE0CC1408102001310083650000026C9DE34950B723760C49E4FC1522F510BF59FFD6880B3F2331E0948E049ED7594508
887D23676A0F646530D7BA6CFB3AEDD0C299727D1637970406D9A3092D2AE86918E3D126BCB320EC2E8B602D97A7FE10CF5AE7497746CF0A56E8AA444F
3CD5237C7AC30606E2E279E5E4271B4D2141907A99821BEDECA31CE8B5D36EDB61478B6E673853D952B6958DB25F2BA7EA95A67BC7E259A0E823ACBA17
C8C6F38003C4F79191F876DBC95B78A92A2473EF158A4CA04BABE1AB83BA5D3E026928B4CB9E68B884DF08D45C5EBC789B41ACBD5106F9E7EF8A83DE6E
9C5970D6158B88E2228B7424BE7020DE0920E9740531A00F1104517ECC0273E2BC19F201AC0487A7D05AF9D29FBB3C9167565774CF17ACB103694B5EB9
6057B514C26B849773F0888333A8D28D70950BD54CA2980F6B80FB85990BC5154AB5A26D6F0C267B18F0AE6BBC54FD1D94A2B66FC51F83247C2183050D
E99262E6B829D7AAB266F4877F7AF64F0340DBE7D2BF3DD4E06201D0C54059471963AB9DD6C53DDE84BD5144F281EEE1BD5B06B1A2C8E9700AA29BD150
DAB64AD02C92D3711AB0EA27549C7FEEA9DD88C7F5D6C51CF6B4B8553F2BD90F1176D4F3B89A244F95DAF7B2F060B78F0255AE0A74D1444F9A484B9093
0BA66159EF203A8E476681C45A859B716D8C0BB17D1C24890A4624E72ECB5C4557BF807D59352BF0B4ADA324DF70CF7537AD43D9138BCD7AE2705C0559
0828E3D1B1736BD348BE39C32D4EC50F790AB3C69EC66D7D95098DA090EABE77C45CCB4E91510569033E985919B8873237DE59323B6B4DBBE391C220A0
16130C42EDF4681897FD
ike 0:VPN-to-SH:2:0: responder received first quick-mode message
ike 0:VPN-to-SH:2: dec
2F013665B867DE804878DFC8FEE0CC1408102001310083650000026C01000024C7EFFF7C6D036D232FFF690088223379 28B03AFD50F33CED859B59D9AC
9412950A0000EC0000000100000001000000E001030407DB38E2B303000020010C0000800100018002A8C0800400018006008080050002 8003000E0300
0020020C0000800100018002A8C0800400018006010080050002 8003000E03000020030C0000800100018002A8C0800400018006008080050005800300
0E03000020040C0000800100018002A8C0800400018006010080050005 8003000E0300001C0514000080010001 8002A8C0800400018006008080030 00E
0300001C0614000080010001 8002A8C0800400018006010080030 00E0000001C071C0000800100018002A8C0800400018006010080030 00E040000147D
2BEB602046C003CC7C8078DB63B5A805000104712E6A9987254B0247F457E224710813BC36723BD07FDC6D2420A3F8C1A6612EE12F0CB8FE7F18EA8318
C03EB5B25AC803317615EAFA6600B18F97D90FB8D16928CA3A95AAE106C87A4E368B460882AE763AE31D4484B18CFD50595144BD0BF7D748635590A894
1EB4CCA7C6A25ADB532CE4807E08F3BFC6D781C1B3A4DF1CE59849CFEEF7A71E92FDB318C37EBDCBAFAEFE35670D1A9FCF77C818CD2708017E0482BAA4
2BF9951BD52E55C6CBD6AECF6985510D388FD04931532D5C932696885C6492C0F0CDD3C3965DB1D5EA477A0416CA98692CC6EBA52142719D965CACBD1F
B78574D33056E5EF1A5D7D5878CA209BF6C2DBA82526AE34503FEB58C1F77705000010040000000C0A80100FFFFFF0000000010040000000C0A80000FFFF
FF00B09C916D1B008707
```

第二阶段有匹配的感兴趣流和proposal

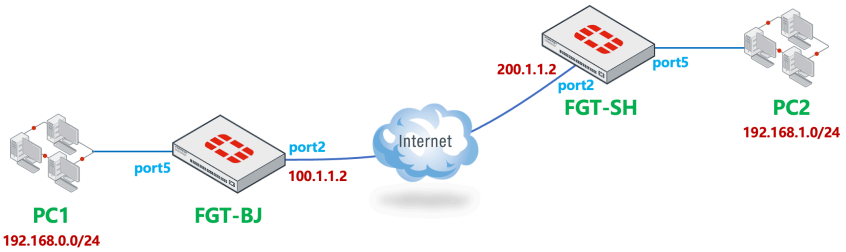
```
ike 0:VPN-to-SH:2:0: peer proposal is: peer:0:192.168.1.0-192.168.1.255:0, me:0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:2:VPN-to-SH:0: trying
ike 0:VPN-to-SH:2:VPN-to-SH:0: matched phase2
ike 0:VPN-to-SH:2:VPN-to-SH:0: autokey
ike 0:VPN-to-SH:2:VPN-to-SH:0: my proposal:
ike 0:VPN-to-SH:2:VPN-to-SH:0: proposal id = 1:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   PFS DH group = 14
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0: proposal id = 2:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   protocol id = IPSEC_ESP:
```

```
ike 0:VPN-to-SH:2:VPN-to-SH:0:   PFS DH group = 5
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0: incoming proposal:
ike 0:VPN-to-SH:2:VPN-to-SH:0: proposal id = 1:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   PFS DH group = 14
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:2:VPN-to-SH:0: negotiation result
ike 0:VPN-to-SH:2:VPN-to-SH:0: proposal id = 1:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:2:VPN-to-SH:0:   PFS DH group = 14
ike 0:VPN-to-SH:2:VPN-to-SH:0:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:2:VPN-to-SH:0:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:2:VPN-to-SH:0:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:2:VPN-to-SH:0: set pfs=MODP2048
ike 0:VPN-to-SH:2:VPN-to-SH:0: using tunnel mode.
ike 0:VPN-to-SH:2:VPN-to-SH:0: generate DH public value request queued
ike 0:VPN-to-SH:2:VPN-to-SH:0: compute DH shared secret request queued
ike 0:VPN-to-SH:2:VPN-to-SH:0: replay protection enabled
ike 0:VPN-to-SH:2:VPN-to-SH:0: SA life soft seconds=42929.
ike 0:VPN-to-SH:2:VPN-to-SH:0: SA life hard seconds=43200.
ike 0:VPN-to-SH:2:VPN-to-SH:0: IPsec SA selectors #src=1 #dst=1
ike 0:VPN-to-SH:2:VPN-to-SH:0: src 0 4 0:192.168.0.0/255.255.255.0:0
ike 0:VPN-to-SH:2:VPN-to-SH:0: dst 0 4 0:192.168.1.0/255.255.255.0:0


生成IN和OUT方向的IPSEC SA
ike 0:VPN-to-SH:2:VPN-to-SH:0: add IPsec SA: SPIs=fe02f809/db38e2b3
ike 0:VPN-to-SH:2:VPN-to-SH:0: IPsec SA dec spi fe02f809 key 16:95A30E49CBCB4EFCBAA65C8CA9B0074C auth
20:C5E5A450DABB70B398CD151C67D3B30E2BC938EA
ike 0:VPN-to-SH:2:VPN-to-SH:0: IPsec SA enc spi db38e2b3 key 16:A4DDB950CCE7BAD17A48AAB32F61514A auth
20:10B16B53C56103A2036BF5308EB3DB30885DB220
```

```
ike 0:VPN-to-SH:2:VPN-to-SH:0: added IPsec SA: SPIs=fe02f809/db38e2b3

隧道UP
ike 0:VPN-to-SH:2:VPN-to-SH:0: sending SNMP tunnel UP trap
ike 0:VPN-to-SH:2: enc
2F013665B867DE804878DFC8FEE0CC1408102001310083650000001B001000024C308502AF541F30479F8C453637BADED887799A67B446206E205F558A5
BC077D0A000038000000010000000010000002C01030401FE02F80900000020010C0000800100018002A8C0800400018006008080050002800300 0E0400
00141451040EC380E31C646516A89096C12505000104890DB8BA0888142ACD3C99CBD3B57B71186093383DA57E60CE60308E0E9BEB6B27C751FF683DC4
FC56EAB5DECE51B3A52C085A0C477DEF0BAE3ACCA11764D4C49C0ADBF6E12A033929ADC01A1C6D7587C5BE571C7E63781906FA124705430DBDDF477C23
8A27E9358A1A8C8FEF923702229F4C8F790847066853CF54E7AD9ED0C3B847C3ADB01A262FA03C3B36EEDDE140E5975C509DB991A65764CD8F2E8569FE
870118494C3735784B8F91AF3F82D296E04FA2C061D0BCC2B713115077AC2223402FEAA2DBA0E8530C10D033FEF9D150B30F77A26241B542A613E19159
0DB77FD0A9A9432C20B526FEAE06EE70920F382F9C49928D39118013ED6EC7697BB20500001004000000C0A80100FFFFFF0000000010040000 00C0A800
00FFFFFF00

响应方回复的数据包，快速模式第2个包，发送确认的安全参数和身份认证信息
ike 0:VPN-to-SH:2: out
2F013665B867DE804878DFC8FEE0CC1408102001310083650000001BC5BF94023269A223BC47B8445F7F1DB5707F939EDAFF54CF75D4A3B1CFCF2C256BF
F1ACC1E94729AA344342A8D3C38849D3E7AD8827053E70BEF72D069A331301EBE135F7674951DB0D1B568F0C47D778E992FC25D61D0690EB27AC0481CF
C80317A30A7CF06A20635F53AA4EDD2B830C92E9EF5EAA3D1E3272ECEAB88CBC187A242D81B1F9DA053482D05FF4ACDB571F8F9B9D675C4BFEC401E9E1
3AEEFF6A17F8689BDDAE83CFE363AFF24BA4D6C7D1617D0A7AA4C5A63E0C60BA40A7B92D020E494A35C7242308C437436DAD7C015D43BF517F55FDD120
7D9C4719B48D77DED72DA81289957600914 1D1F9E2307A279523EAF0850B7A3968BA3AD4E7EE48761EF8E11D046BCB54BC4114E2E6A69E7B3A0CBE071D
1A0A23BCA262AD92A9FD6BD193A935819B5BEE6067FBAA1FC9B087F84CA3641DA372AA27FE6A17B751A72F263CE3D2FDED093809CB7CE6E7D3D2A6556E
720147AD12D2D8861490A9E78EC213F5F155B93D6075AD26FE1177EDA152551EB1867B3B7C65EFA2CB7387D780427B0662F9EC667139E17B9692F36060
F0ECB8F8620EA1DEAB55488A5A6D4BD3AB
ike 0:VPN-to-SH:2: sent IKE msg (quick_r1send): 100.1.1.2:500→200.1.1.2:500, len=444, vrf=0,
id=2f013665b867de80/4878dfc8fee0cc14:31008365

收到发起方的数据包，快速模式第3个包，发起方的确认报文
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=2f013665b867de80/4878dfc8fee0cc14:31008365 len=76 vrf=0
ike 0: in
2F013665B867DE804878DFC8FEE0CC1408102001310083650000004CC37A67EFDA1D1385BAD16B7F1462D26C8510BEA49AD6681DA24219B2C2B9B3CF5E
C6577F1AE0051F2DBEC65F7BB65CD7
ike 0:VPN-to-SH:2: dec
2F013665B867DE804878DFC8FEE0CC1408102001310083650000004C000000249C86B516C45350E8D2254EE877DBFB8CF7B461D3454C9FD93BCEFB1770
35802712E75AF0009C454FDB097C0B
ike 0:VPN-to-SH:VPN-to-SH:0: send SA_DONE SPI 0×db38e2b3
ike 0:VPN-to-SH: link is idle 10 100.1.1.2→200.1.1.2:0 dpd=1 seqno=1 rr=0


# diagnose debug  disable
```

IPSec IKEv1 野蛮模式 Debug 分析

网络拓扑



IPSEC Debug 分析

FGT - SH 发起协商，FGT - BJ 进行 Debug 查看，下面是 FGT - BJ 的 debug 输出

```
# diagnose debug application ike -1
# diagnose debug  enable

收到对端的数据包，野蛮模式第1个包
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=d281703dcfc7fc74/0000000000000000 len=720 vrf=0
ike 0: in
D281703DCFC7FC7400000000000000000001100400000000000000002D0040000B400000001000000010000000A80101000403000028010100008000B000100
0C000400015180800100078000E00808003000180020004800400 0E0300002803010000800B0001000C000400015180800100078000E0100800300018002
00048004000E0300002805010000800B0001000C000400015180800100078000E00808003000180020002800400 0E0000002807010000800B0001000C00
040001518080010007800E0100800300018002000280040 00E0A00010453E290339EEA9422A4B4B1034821162393F25DC77AE0D24864206F06B8E78267
33B2FD2DF5C70E86310C29F83A77057D630FF9F8C0CE6DBA3FB571D5D3960C9AAC7260FE6D8019E523508632E1423CF657A462DCA2F4EFB96268E48003
194A77D9D85C6A864F1366B7FDA693D1798DFBADCDF3797116C551F260BC03DFA9B7ADF795EAD145DE2ADD68EDF20E87E67D8F2E90D3245D3A8ED9AB63
0134CA5F23158BFDA4A6DBC1E5F6C03FB6C3278A3209295A8C7558E93BBB66F115383B11CF7F4644608A5E448B53F4B2BC34470646445575684CD0850
FE9D6B47A981BBAEDD24D60F8424618B0F45E34B8C20FF7A48AF12192B44EED4DF2F333533849321F505000024D612B8B99DFE002B24E032D5DA3AB863
E9AC517913DE2F80658C3C0B3554A9190D00000C01000000C80101020D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C
179D9215529D560D000014CD60464335DF21F87CFDB2FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D8382
1A0F0AEAA8620D0000144485152D18B6BBCD0BE8A8469579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE
7F00D6C2D30D0000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000000000148299031757A36082C6A621DE00000000
ike 0:d281703dcfc7fc74/0000000000000000:3: responder: aggressive mode get 1st message ...
ike 0:d281703dcfc7fc74/0000000000000000:3: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:d281703dcfc7fc74/0000000000000000:3: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:d281703dcfc7fc74/0000000000000000:3: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:d281703dcfc7fc74/0000000000000000:3: VID draft-ietf-ipsec-nat-t-ike-02 90CB80913EBB696E086381B5EC427B1F
ike 0:d281703dcfc7fc74/0000000000000000:3: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:d281703dcfc7fc74/0000000000000000:3: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:d281703dcfc7fc74/0000000000000000:3: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:d281703dcfc7fc74/0000000000000000:3: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:d281703dcfc7fc74/0000000000000000:3: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:d281703dcfc7fc74/0000000000000000:3: VID FORTIGATE 8299031757A36082C6A621DE00000000

发起方使用IP地址作为身份ID
ike 0::3: peer identifier IPV4_ADDR 200.1.1.2
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.

VPN-to-SH有匹配的Proposal
ike 0: cache rebuild done
ike 0:d281703dcfc7fc74/0000000000000000:3: negotiation result
ike 0:d281703dcfc7fc74/0000000000000000:3: proposal id = 1:
ike 0:d281703dcfc7fc74/0000000000000000:3:   protocol id = ISAKMP:
ike 0:d281703dcfc7fc74/0000000000000000:3:      trans_id = KEY_IKE.
ike 0:d281703dcfc7fc74/0000000000000000:3:      encapsulation = IKE/none
ike 0:d281703dcfc7fc74/0000000000000000:3:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:d281703dcfc7fc74/0000000000000000:3:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:d281703dcfc7fc74/0000000000000000:3:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:d281703dcfc7fc74/0000000000000000:3:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:d281703dcfc7fc74/0000000000000000:3: ISAKMP SA lifetime=86400
```

```
ike 0:d281703dcfc7fc74/0000000000000000:3: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0×8291690 10 100.1.1.2→200.1.1.2:500.
ike 0:VPN-to-SH:3: DPD negotiated
ike 0:VPN-to-SH:3: peer is FortiGate/FortiOS (v0 b0)
ike 0:VPN-to-SH:3: selected NAT-T version: RFC 3947
ike 0:VPN-to-SH:3: generate DH public value request queued
ike 0:VPN-to-SH:3: compute DH shared secret request queued
ike 0:VPN-to-SH:3: cookie d281703dcfc7fc74/d07504953e9d8606
ike 0:VPN-to-SH:3: ISAKMP SA d281703dcfc7fc74/d07504953e9d8606 key 16:9F8B9F21828B2667E9F48941FAE1411C
```

响应方回复的数据包，野蛮模式第2个报文，响应方用选定提议的所有参数，DH公开值进行应答，以及身份ID供发起方验证

```
ike 0:VPN-to-SH:3: out
D281703DCFC7FC74D07504953E9D8606011004000000000000002500400003C0000000100000001000000300101000100000028010100008000B000100
0C0004000151808001000780 0E0080803000180020004800 4000E0A000104797A5665CD5074E082FADF617796BCD6769F2E25B6A03AB6FDF9AF091A20
F4695EE09AD2266A3F99D70C8EC1B470EFA5D2B768626C412AD2F6F3FB6D69FDD77C07FB9A275DC405328907BCF8D587D375C914378C28788E4978BE85
654DBBA2E9E057968A72A7B4B5FC460588F20130C54B4A041261E4DC38B01F10B4118C243ECC4BD50F2E1034879B7ED428F2003DAE42C74A7479A20B3A
2CEB082321F640F4BEC18FF6BA01E3E34600D713DEBD218A350ABE0ACC9F61904F2F43949BFDC43D0A6F3D656B1522B16050D04205DF5E60118758106E
F1852CF801163810164D3B7B2064C4A54185660E20493F81102B12FF4B88B1A447F34749178AE6CDB39185050000140DA682516661FD4466ADF8F415C3
423C0800000C01000000640101020D0000240A3F41A59BC6E3354F353985366B8D7148AF7B96F560E33CBED1D86B8E28CB4C140000144A131C81070358
455C5728F20E95452F14000024D3141A6A7D9320C4105437E65A9B4E60E653C68D9A71A5AFFC24B06A49CED2EE0D0000244B1C68D360542FC9E16DC786
7E8FC6CA8FC90FBDC591A2AC1F0B7B37CE0A6BA60D000014AFCAD71368A1F1C96B8696FC775701000D0000148299031757A36082C6A621DE000000000D
0000144048B7D56EBCE88525E7DE7F00D6C2D3000000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:VPN-to-SH:3: sent IKE msg (agg_r1send): 100.1.1.2:500→200.1.1.2:500, len=592, vrf=0,
id=d281703dcfc7fc74/d07504953e9d8606
```

收到发起方的数据包，野蛮模式第3个报文，发起方的确认信息，确认与响应方可以通信

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=d281703dcfc7fc74/d07504953e9d8606 len=172 vrf=0
ike 0: in
D281703DCFC7FC74D07504953E9D860608100401000000000000000AC07B95FFAA22F64125C7C69E4BE2CDAA9CF69581F048D1D9FF7526A143818E59F1D
59314B926251FAF7B80060D7317500A859CB66E64DE016BFC7E1C9D52DE71246F25B9959A6CE700EF95E28746F045B41AAE04299D7F39A4C8C98BAB85A
F3D842971CD2CDB280743677706CCD7A2D1D334D7F6A9CC59946388B50734290E8ECBBBFAA9E8198207007CADFB101184CBD
ike 0:VPN-to-SH:3: responder: aggressive mode get 2nd response …
ike 0:VPN-to-SH:3: dec
D281703DCFC7FC74D07504953E9D860608100401000000000000000AC14000024D90A7BBC0962B9E6192FA19BDAFDD607B0FDFBC19A05A019F2259FC9D0
EE6371140000244B1C68D360542FC9E16DC7867E8FC6CA8FC90FBDC591A2AC1F0B7B37CE0A6BA60B000024D3141A6A7D9320C4105437E65A9B4E60E653
C68D9A71A5AFFC24B06A49CED2EE0000001C0000000101106002D281703DCFC7FC74D07504953E9D860628CB58CA69594207
ike 0:VPN-to-SH:3: received NAT-D payload type 20
ike 0:VPN-to-SH:3: received NAT-D payload type 20
ike 0:VPN-to-SH:3: received p1 notify type INITIAL-CONTACT
```

预共享秘钥认证成功

```
ike 0:VPN-to-SH:3: PSK authentication succeeded
ike 0:VPN-to-SH:3: authentication OK
ike 0:VPN-to-SH:3: NAT not detected
```

生成IKE SA

```
ike 0:VPN-to-SH:3: established IKE SA d281703dcfc7fc74/d07504953e9d8606
ike 0:VPN-to-SH:3: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike 0:VPN-to-SH:3: processing INITIAL-CONTACT
ike 0:VPN-to-SH: flushing
ike 0:VPN-to-SH: flushed
ike 0:VPN-to-SH:3: processed INITIAL-CONTACT
ike 0:VPN-to-SH: set oper up
ike 0:VPN-to-SH:3: no pending Quick-Mode negotiations
```

收到发起的数据包，快速模式第1个包

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=d281703dcfc7fc74/d07504953e9d8606:03b7f8e6 len=620 vrf=0
ike 0: in
D281703DCFC7FC74D07504953E9D86060810200103B7F8E60000026C407ACC496B7A31B516DCEBFC25D4C6C668B271C7969F0B1785DF58417ED399C37B
D28D42CC804C33CDE204D57588C4354E4A4A421804B7160B020EEF914F7BC2368EBD8EB6D322359574E37DEFA2C255E7BAE94FF245859DE7912F4A0C20
F1594833582B0A40CE6D9D267ADE8CCDFB47F06B360CA48256F67D0834B6657C3A46C9DE785F0117A336F83810DF0F2F7CC2926BD5757F093C453BB0AB
32ACE53DF72C096045C5C5E0D0FE3E79D2DE5108D2AE0AE916E364D152A131AC2E8C980EEC1CCA7BAD7175739A4B09FF95BA46C482D8149A9B3A7A2D0A
1DDABEE65391070FEDB3497CF5436B0B9B8F9ED64F087CB601C6A871338BF756D28D8FA1B703D92BCC51643BF78BCF156DF38046951B9D71108FD12F8D
1A9EAFED83291E95BE22C2CAB810A10E00AA9B82EB4789270EF1CC3AA442A946A21E7FFDCBD85336AD8C8773F231EB9669992CE37A8F9B069189103D59
B587441485370D7260E8F61C1A94E9527C15FDEB26AE4D3AB69477F95892BB4429E0DBF2CA55A7543664A11BC2643ABE7008D97B0D6A12F3827E0F9F2F
D98DB4A92B846D8D366702F77523EA5F4CA6D4B6621A267623758C9C401C1A9C7FA595EDC789D0EAACBAB88FF66E6532381135B859609595B877D7661F
DAF1B684733451FC63FCED3C1911FE14EB3906267CC24ECE199CB0DF7405C7D4C9D838AB44A0FD78AE099670A5724CDA917FD2333912E8012FC9BD51D3
```

```
93517FA78184BC5351B2883ABA8F6AA876D1327D6E79ACD13A921C56EF904E29073200647A1DFA2228B0334AC8BF881814A612AF98637D691383A2D9FF
3901FAF6ADF04FCEC55B
ike 0:VPN-to-SH:3:230: responder received first quick-mode message
ike 0:VPN-to-SH:3: dec
D281703DCFC7FC74D07504953E9D86060810200103B7F8E60000026C01000024F703EDB480571505460F14635B22E25FDFF7DF6B9E191863AA7EB50C0E
12D04F0A0000EC0000000100000001000000E001030407DB38E2B403000020010C0000800100018002A8C0800400018006008080050002800300 0E0300
0020020C0000800100018002A8C0800400018006010080050002800300 0E0300002003 0C0000800100018002A8C0800400018006008080050005800300
0E0300002004 0C0000800100018002A8C0800400018006010080050005800300 0E0300001C0514000080010001800 2A8C0800400018006008080030000E
0300001C0614000080010001800 2A8C0800400018006010080030000E0000001C071C0000800100018002A8C0800400018006010080030000E040000149F
D42C6D9B3DFD82CB79295C580709A305000104A2533113195DAEB243659EAF23432B20E32C34FDA0642C25C1EE2B76EB5E2E6C2548A3B88F58251C8056
76AC034AB82D49AEE335AFA2F81D2CD91B85B7BA20A614C30C1A18BBFAD91A0096377FCD2B23EE36457D337899F356C65A95EFC1510AB13F053C29A411
0A046F7875F15CB6D9D7EE6839883A6BF2BAA63F50C121FAC79535E3473F8923779BEB886F401A298ADDE0D158CA543089CDCF1724714916CD066A8135
AB09BC90B19B3C5B11AC81A18CCD85A3FF06B205D0A053659596D46CD57809980095C4BEC4F9256453E2F65857DD22E05A558466F54DB7A8F65DF67F6F
7372CA2C9EBC88ACEBE89140D4B52538F79B73BA7B3EEE86AB669A455D059005000010040000000C0A80100FFFFFF0000000010040000000C0A80000FFFF
FF0042BE359222C53C07
```

第二阶段有匹配的感兴趣流和proposal

```
ike 0:VPN-to-SH:3:230: peer proposal is: peer:0:192.168.1.0-192.168.1.255:0, me:0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:3:VPN-to-SH:230: trying
ike 0:VPN-to-SH:3:VPN-to-SH:230: matched phase2
ike 0:VPN-to-SH:3:VPN-to-SH:230: autokey
ike 0:VPN-to-SH:3:VPN-to-SH:230: my proposal:
ike 0:VPN-to-SH:3:VPN-to-SH:230: proposal id = 1:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   PFS DH group = 14
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230: proposal id = 2:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   PFS DH group = 5
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230: incoming proposal:
ike 0:VPN-to-SH:3:VPN-to-SH:230: proposal id = 1:
```

```
ike 0:VPN-to-SH:3:VPN-to-SH:230:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   PFS DH group = 14
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=NULL
ike 0:VPN-to-SH:3:VPN-to-SH:230: negotiation result
ike 0:VPN-to-SH:3:VPN-to-SH:230: proposal id = 1:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:3:VPN-to-SH:230:   PFS DH group = 14
ike 0:VPN-to-SH:3:VPN-to-SH:230:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:VPN-to-SH:3:VPN-to-SH:230:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:3:VPN-to-SH:230:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:3:VPN-to-SH:230: set pfs=MODP2048
ike 0:VPN-to-SH:3:VPN-to-SH:230: using tunnel mode.
ike 0:VPN-to-SH:3:VPN-to-SH:230: generate DH public value request queued
ike 0:VPN-to-SH:3:VPN-to-SH:230: compute DH shared secret request queued
ike 0:VPN-to-SH:3:VPN-to-SH:230: replay protection enabled
ike 0:VPN-to-SH:3:VPN-to-SH:230: SA life soft seconds=42933.
ike 0:VPN-to-SH:3:VPN-to-SH:230: SA life hard seconds=43200.
ike 0:VPN-to-SH:3:VPN-to-SH:230: IPsec SA selectors #src=1 #dst=1
ike 0:VPN-to-SH:3:VPN-to-SH:230: src 0 4 0:192.168.0.0/255.255.255.0:0
ike 0:VPN-to-SH:3:VPN-to-SH:230: dst 0 4 0:192.168.1.0/255.255.255.0:0
```

生成IN和OUT方向的IPSEC SA

```
ike 0:VPN-to-SH:3:VPN-to-SH:230: add IPsec SA: SPIs=fe02f80a/db38e2b4
ike 0:VPN-to-SH:3:VPN-to-SH:230: IPsec SA dec spi fe02f80a key 16:EC236EB218FDFDB618A4F3F9365BC2A6 auth
20:87C419704D9037DF765191533D14EAD75985138F
ike 0:VPN-to-SH:3:VPN-to-SH:230: IPsec SA enc spi db38e2b4 key 16:DD0E65226AEB9DC29360DACD7874D604 auth
20:4E1E6ADE3A2BF7F38507D2FA5202121C9C8959D6
ike 0:VPN-to-SH:3:VPN-to-SH:230: added IPsec SA: SPIs=fe02f80a/db38e2b4
```

隧道UP

```
ike 0:VPN-to-SH:3:VPN-to-SH:230: sending SNMP tunnel UP trap
ike 0:VPN-to-SH:3: enc
D281703DCFC7FC74D07504953E9D86060810200103B7F8E6000001B001000024413D8E541E90DE2DD722E84C5DDB7A0305D221EA20A2D53D6762ED25D7
4F6E930A000038000000010000000010000002C01030401FE02F80A00000020010C0000800100018002A8C0800400018006008080050002800300 0E0400
0014C56F0A2B71A119C8EAB36D97BAC4EEA305000104DEA2BDF7CD3559834DE23B8726D4DA4B2D898E9764AB0E2EAF160F491FE06241136B6F7AFACDC1
D95C11A410F81873854CB58C6F01E57007899C8117E5A195E82A1026164307EDE5755191B9D04DABBC68F8ECD89C647AADD7930F8F2D0C8192BAD7B897
2E909A799255B91DF75FC3A2520918C10D2FE3520BB242299E22B16E8640BD323AB7508B1F8FD29D10BCB164841CC8D93B4C9688280998F26DA9C19B0C
5299A58AFBC55E8C56171FC17DA0ED6231550F37D9468CE6DC3D0FEDAAAC946916CEC1F2D81FC1FBBA2664810466BED343E7D5E441B63367B74CB7E6B3
3074BBBD538FE9DB46659F4D93E28A387CF33AF6B0448BE4F81D6FB01296E4616CD705000010040000000C0A80100FFFFFF0000000010040000000C0A800
00FFFFFF00
```

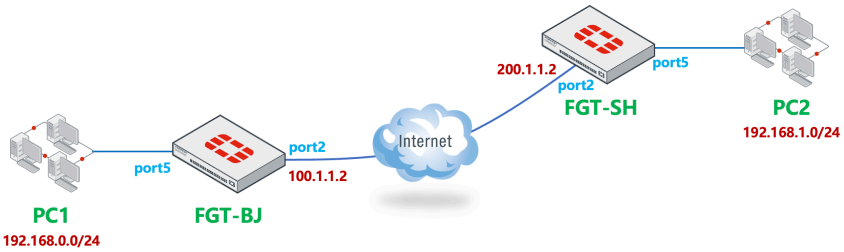
响应方回复的数据包，快速模式第2个包，发送确认的安全参数和身份认证信息

```
ike 0:VPN-to-SH:3: out
D281703DCFC7FC74D07504953E9D86060810200103B7F8E6000001BC10B04C4F733E257D73525EBEDB01FA1A25FC3BD684DD58F70CDA141AAC4F133E03
D65A9D3C6E8CE741F647E0342CEF07F0DF809D58566CFF017720EA9F27B873C28BBC35235E5B1B9F3A53EB0EC357AFAD729E146B7C371228FDE746E94E
B35CB1D5D046C5CC736E5F193AFB4451B0D0FD7FD6D142BFF53BA3443A16B8FEA41FDD9BA2596CEA16FFEA245EA0B0860F565062C6DEC2024E7E1B0BC2
BB0FE7ECF49F5F5C721AD5E9E4C26107D1A3A6B2FCCC777E9D6B9DD5DF8A920417EC102E90E7CD7008437520568E9A0E5C67E3D918F7EC16166B8E6092
EE47811FFAF75A795A97C3625E23E6D94A17538C2B4645102E0058229D4C5F6B5656AEF34B231A8E5A3D9422C534BC23F6D2CDAD5D1A17EE968D0C1B2A
DD0EC1022EA1CFDAF758728C606C60CE27A4BB4DC54D76C6DBA1B208301D52F8E17759DED5F91A24A6C6328D581FC6110E17FD6166C9EA1FC21791AB18
3E9A1F3068B3551949E5C8A3F15DFC0348EDD2485F428598BED1B3DE1BF2E7017989E03A5F37167F64648EED3AF3F30447D5054A0AB7AD192BEA4AF625
706283F673B2D93EA6D6B17537D86C1905
```

```
ike 0:VPN-to-SH:3: sent IKE msg (quick_r1send): 100.1.1.2:500→200.1.1.2:500, len=444, vrf=0,
id=d281703dcfc7fc74/d07504953e9d8606:03b7f8e6


收到发起方的数据包，快速模式第3个包，发起方的确认报文
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=d281703dcfc7fc74/d07504953e9d8606:03b7f8e6 len=76 vrf=0
ike 0: in
D281703DCFC7FC74D07504953E9D86060810200103B7F8E60000004CBFE7E9BC7B60005BC3C8AB90CACB6C5412DACDAAE36B845C34C802177DAD5E1D0A
243A6B8BA8770B920A75A6D68626C7
ike 0:VPN-to-SH:3: dec
D281703DCFC7FC74D07504953E9D86060810200103B7F8E60000004C0000002421BD4F317C5DD58943C142AE4186BE1BBE36FBC91C8600E924DC42BD95
107512E61AC8842070C9A16C994E0B
ike 0:VPN-to-SH:VPN-to-SH:230: send SA_DONE SPI 0×db38e2b4
ike 0:VPN-to-SH: link is idle 10 100.1.1.2→200.1.1.2:0 dpd=1 seqno=e5 rr=0

# diagnose debug  disable
```

IPSEC IKEv2 Debug 分析

网络拓扑



IPSEC Debug 分析

1. FGT - SH 发起协商，FGT - BJ 进行 Debug 查看，下面是 FGT - BJ 的 debug 输出。

```bash
diagnose debug application ike -1
diagnose debug  enable
```

2. 收到发起方的数据包，SA_INIT 交互的第 1 个包。

```bash
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv2 exchange=SA_INIT id=0640897749c18520/0000000000000000 len=632
ike 0: in
0640897749C1852000000000000000000212022080000000000000278220000F00200003401010005030000C0100000C800E00800300000802000005
030000080300000C030000080400000500000008040000E0200003402010005030000C0100000C800E010003000008020000050300000803000000C
030000080400000500000008040000E0200002C030100040300000C01000014800E008003000008020000050300000804000005000000080400000E
0200002C040100040300000C01000014800E010003000008020000060300000804000005000000080400000E0000002C050100040300000C0100001C
800E010003000008020000050300000804000005000000080400000E28000108000E0000D7B68B7EF380848D233FF08CD57789166A356E53E982392A
157953AE8939C833D28C94436514956B08E168503B92D5B09E97FBA36CBCC3B3CBCB5DC9640C6E6F073FFC88B01DC668D8AF3B61AC2D46B71524B91E
CD8CE68F1529986A43BB2E357CFDEE972FFBFCB2B5BC92A38BE6B9E8BEB47A923216AA3753346BB764CF7760E526B1FD010A8FB51E7B1BB7F80782D5
3B8C6690852F4D03969CE4CFE8F2252360A4F5672A5C76BA16E45053259BF8E0A154E9BFF1709B7A72ECF15ED1936E512288D639935CF8096D9F4B85
27444CE4C13657F6B751371B6E183B688870C466D5E439DD1AA319306DBC2C6F5EDC76A1F4EF5223F306E09B3F990FDFF6675B1D290000242941FAF7
F9BCFBFA6A308624D8327BB7BDBA4C72BAAA79E27F5FE76580056BE82900001C000040046248FDD09A0E5B87EFEE4EBE51267EFAA176C9692900001C
00004005B85DCDB85885019864A547D5DB8E60946F538368000000080000402E
```

3. VPN-to-SH 有匹配的 proposal。

```bash
ike 0:0640897749c18520/0000000000000000:8: responder received SA_INIT msg
ike 0:0640897749c18520/0000000000000000:8: received notify type NAT_DETECTION_SOURCE_IP
ike 0:0640897749c18520/0000000000000000:8: received notify type NAT_DETECTION_DESTINATION_IP
ike 0:0640897749c18520/0000000000000000:8: received notify type FRAGMENTATION_SUPPORTED
ike 0:0640897749c18520/0000000000000000:8: incoming proposal:
ike 0:0640897749c18520/0000000000000000:8: proposal id = 1:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=AES_CBC (key_len = 128)
ike 0:0640897749c18520/0000000000000000:8:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP1536.
ike 0:0640897749c18520/0000000000000000:8: proposal id = 2:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=AES_CBC (key_len = 256)
ike 0:0640897749c18520/0000000000000000:8:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP1536.
ike 0:0640897749c18520/0000000000000000:8: proposal id = 3:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=AES_GCM_16 (key_len = 128)
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
```

```
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP1536.
ike 0:0640897749c18520/0000000000000000:8: proposal id = 4:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=AES_GCM_16 (key_len = 256)
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_384
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP1536.
ike 0:0640897749c18520/0000000000000000:8: proposal id = 5:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=CHACHA20_POLY1305 (key_len = 256)
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP1536.
ike 0:0640897749c18520/0000000000000000:8: matched proposal id 1
ike 0:0640897749c18520/0000000000000000:8: proposal id = 1:
ike 0:0640897749c18520/0000000000000000:8:   protocol = IKEv2:
ike 0:0640897749c18520/0000000000000000:8:      encapsulation = IKEv2/none
ike 0:0640897749c18520/0000000000000000:8:         type=ENCR, val=AES_CBC (key_len = 128)
ike 0:0640897749c18520/0000000000000000:8:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike 0:0640897749c18520/0000000000000000:8:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:0640897749c18520/0000000000000000:8:         type=DH_GROUP, val=MODP2048.
ike 0:0640897749c18520/0000000000000000:8: lifetime=86400
ike 0:0640897749c18520/0000000000000000:8: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0×829cec0 10 100.1.1.2→200.1.1.2:500.
ike 0:VPN-to-SH:8: processing notify type NAT_DETECTION_SOURCE_IP
ike 0:VPN-to-SH:8: processing NAT-D payload
ike 0:VPN-to-SH:8: NAT not detected
ike 0:VPN-to-SH:8: process NAT-D
ike 0:VPN-to-SH:8: processing notify type NAT_DETECTION_DESTINATION_IP
ike 0:VPN-to-SH:8: processing NAT-D payload
ike 0:VPN-to-SH:8: NAT not detected
ike 0:VPN-to-SH:8: process NAT-D
ike 0:VPN-to-SH:8: processing notify type FRAGMENTATION_SUPPORTED
ike 0:VPN-to-SH:8: responder preparing SA_INIT msg
ike 0:VPN-to-SH:8: generate DH public value request queued
ike 0:VPN-to-SH:8: responder preparing SA_INIT msg
ike 0:VPN-to-SH:8: compute DH shared secret request queued
ike 0:VPN-to-SH:8: responder preparing SA_INIT msg
```

4. 响应方回复的数据包，SA_INIT 交互的第 2 个包，回复选定的 proposal，临时随机数。

```bash
ike 0:VPN-to-SH:8: out
0640897749C1852079D668275E8FAB8521202220000000000000001A8220000300000002C0101000403000000C0100000C800E0080030000080200000503000080300000C000000080400000E28000108000E000032F9A820F465C00102EAE1056B3A4DF6DE5B01C6A412F62CFF70E4A8C0657841624ADE35189F3859E91FA1D4EB8AD2079B2672CAE23B4EB3483F687782E7DD184617255D9D3CF5987140EFB5E4CE13CC8791F17848D159153FDEE42249BA69969FD50497EE35CECBD972655A69B5B754395D7CEB23D3594484A61E9D73DBA6DB2F174C96C4378F68BDDE672F8D58DDABC40414048E78E9B443A4A61D4FF49F286E19EE15FF0F4F1BF1FA65E247EAEAB3F3E8DA4DDB4629B4117A26D001F31511C33FDDB6B6FCA71459CE367717A6A31C45AFC519A1FBC4F7785D60831FF06B926E1B03FB978B4D668E4CF129FA9334E6D3CE4B4C42653AEB779F484257A57F6F29000014C40061FEA012E49F9283F72CE763FF882900001C000040046697454FC958F7F6395BF1C4E736670A949CC29F2900001C00004005BF5FEB5E231B18D249058CB0FACC069964298451000000080000402E
ike 0:VPN-to-SH:8: sent IKE msg (SA_INIT_RESPONSE): 100.1.1.2:500→200.1.1.2:500, len=424, vrf=0,
```

5. 生成 IKE SA 和会话秘钥。

```bash
id=0640897749c18520/79d668275e8fab85
ike 0:VPN-to-SH:8: IKE SA 0640897749c18520/79d668275e8fab85 SK_ei 16:45B41A6D8E1483FD7AFBE6FAC7AD5404
ike 0:VPN-to-SH:8: IKE SA 0640897749c18520/79d668275e8fab85 SK_er 16:A49B75CD8B161C9B2BF811E347C84897
ike 0:VPN-to-SH:8: IKE SA 0640897749c18520/79d668275e8fab85 SK_ai
32:9C8219F27844C989B0FBF7DDF69965A779FF2871503BCCA63B8915E0556E0729
ike 0:VPN-to-SH:8: IKE SA 0640897749c18520/79d668275e8fab85 SK_ar
32:C3B60A61FC4AA1C9660568AC39CC2957AC01101FAB71C1D44AFCA83A2303E14F
```

6. 收到发起方的数据包，IKE_AUTH 交互的第 1 个包。

```bash
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv2 exchange=AUTH id=0640897749c18520/79d668275e8fab85:00000001 len=448
ike 0: in
0640897749C1852079D668275E8FAB852E2023080000000100000 1C0230001A4E1AC49EE91ACC84D384E33FC743CD8013E9CAC48143D241030DFAF21A7349A62273BA50FB9FF55C8EA7DAC579275E067A8AF56F7982C4B09BC7838438CDA3A7E1922135D0025DFD7F875743BF4EFC02F139D07A0DBE919EC
```

```
DBCE30D2C703943CFCC76A587FC78A45CA0BC4AF9EFA50BFEA9E0BB9142944F32F0447B60648788C76A52C81F8EA155BAF403E2B6E9CFED650E38EBD
52F56C33D131ED95A1C812BAC7DE00FEC27E85BA265268086E144ADF658829BA388745E16C256E988A9F86C4B6206DBC8F354C6DB2D708D2487174F4
1E3D5E4BF856E1461104858CCFE4FA217592DD616E15431ACB1E9DC0016A4F5D61A7E77B44425DE471512E9C42CF92B8642E0C5DF0F2B5C605DAAF1B
B2CCCA507F275BC4F68E6F263634FF8B66D7F5879329736025E32B14AE0E06C0D4483625F2C462DE381FF0234A65E5E2E1F0A2D048EF0496B949F535
7C76CC8496BF54E65D61011E069E8DD4038B18981DFD7D1BE41A70D76305361B72DB470F287C99DE1ADC38B51796FF1E5E678387D66E8042BE112EF7
EBE5847EB61EAAAAD105C0DF10257BEA8229AEF457366958F9181AE3
ike 0:VPN-to-SH:8: dec
0640897749C1852079D668275E8FAB852E2023080000000100000198230000042900000C01000000C80101022700000800004000290000280200000
4C6DA8BCBDEB2583A8A9810615789489134006FEBCA456E407D70393DB45904821000008000040242C000104020000280103040 3DB38E2B90300000C
0100000C800E0080030000080300000200000008050000000200002802030403DB38E2B90300000C0100000C800E0100030000080300000200000008
0500000002000028030403DB38E2B90300000C0100000C800E0080030000080300000C000000080500000002000028040304 03DB38E2B90300000C
0100000C800E0100030000080300000C000000080500000002000020050304 02DB38E2B90300000C0100001400E0080000000080500000002000020
06030402DB38E2B90300000C0100001400E0100000000080500000000000002007030402DB38E2B90300000C0100001C800E0100000000805000000
2D000018010000000700001000 00FFFFC0A80100C0A801FF000000180100000007000010000 0FFFFC0A80000C0A800FF
ike 0:VPN-to-SH:8: responder received AUTH msg
ike 0:VPN-to-SH:8: processing notify type INITIAL_CONTACT
ike 0:VPN-to-SH:8: processing notify type MESSAGE_ID_SYNC_SUPPORTED
```

7. 发起方使用 IP 作为自身的 ID，认证成功。

```
ike 0:VPN-to-SH:8: peer identifier IPV4_ADDR 200.1.1.2
ike 0:VPN-to-SH:8: auth verify done
ike 0:VPN-to-SH:8: responder AUTH continuation
ike 0:VPN-to-SH:8: authentication succeeded
ike 0:VPN-to-SH:8: responder creating new child
```

8. 找到匹配的 proposal 和感兴趣流。

```bash
ike 0:VPN-to-SH:8:423: peer proposal:
ike 0:VPN-to-SH:8:423: TSi_0 0:192.168.1.0-192.168.1.255:0
ike 0:VPN-to-SH:8:423: TSr_0 0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:8:VPN-to-SH:423: comparing selectors
ike 0:VPN-to-SH:8:VPN-to-SH:423: matched by rfc-rule-2
ike 0:VPN-to-SH:8:VPN-to-SH:423: phase2 matched by subset
ike 0:VPN-to-SH:8:VPN-to-SH:423: accepted proposal:
ike 0:VPN-to-SH:8:VPN-to-SH:423: TSi_0 0:192.168.1.0-192.168.1.255:0
ike 0:VPN-to-SH:8:VPN-to-SH:423: TSr_0 0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:8:VPN-to-SH:423: autokey
ike 0:VPN-to-SH:8:VPN-to-SH:423: incoming child SA proposal:
ike 0:VPN-to-SH:8:VPN-to-SH:423: proposal id = 1:
ike 0:VPN-to-SH:8:VPN-to-SH:423:   protocol = ESP:
ike 0:VPN-to-SH:8:VPN-to-SH:423:      encapsulation = TUNNEL
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=ENCR, val=AES_CBC (key_len = 128)
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=INTEGR, val=SHA
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=ESN, val=NO
ike 0:VPN-to-SH:8:VPN-to-SH:423:         PFS is disabled
ike 0:VPN-to-SH:8:VPN-to-SH:423: matched proposal id 1
ike 0:VPN-to-SH:8:VPN-to-SH:423: proposal id = 1:
ike 0:VPN-to-SH:8:VPN-to-SH:423:   protocol = ESP:
ike 0:VPN-to-SH:8:VPN-to-SH:423:      encapsulation = TUNNEL
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=ENCR, val=AES_CBC (key_len = 128)
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=INTEGR, val=SHA
ike 0:VPN-to-SH:8:VPN-to-SH:423:         type=ESN, val=NO
ike 0:VPN-to-SH:8:VPN-to-SH:423:         PFS is disabled
ike 0:VPN-to-SH:8:VPN-to-SH:423: lifetime=43200
ike 0:VPN-to-SH:8: responder preparing AUTH msg
ike 0:VPN-to-SH:8: established IKE SA 0640897749c18520/79d668275e8fab85
ike 0:VPN-to-SH:8: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike 0:VPN-to-SH:8: processing INITIAL-CONTACT
ike 0:VPN-to-SH: flushing
ike 0:VPN-to-SH: flushed
ike 0:VPN-to-SH:8: processed INITIAL-CONTACT
ike 0:VPN-to-SH: set oper up
ike 0:VPN-to-SH:8:VPN-to-SH:423: replay protection enabled
ike 0:VPN-to-SH:8:VPN-to-SH:423: set sa life soft seconds=42930.
ike 0:VPN-to-SH:8:VPN-to-SH:423: set sa life hard seconds=43200.
ike 0:VPN-to-SH:8:VPN-to-SH:423: IPsec SA selectors #src=1 #dst=1
ike 0:VPN-to-SH:8:VPN-to-SH:423: src 0 7 0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:8:VPN-to-SH:423: dst 0 7 0:192.168.1.0-192.168.1.255:0
ike 0:VPN-to-SH:8:VPN-to-SH:423: add IPsec SA: SPIs=fe02f80f/db38e2b9
```

9. 生成 IN 和 OUT 方向的 IPSEC SA。

```bash
ike 0:VPN-to-SH:8:VPN-to-SH:423: IPsec SA dec spi fe02f80f key 16:81DC7F6F04EF90B86ACD5DE389290829 auth
20:6ED062D0343F9A3905CB3DF496846DDE03A3F5ED
ike 0:VPN-to-SH:8:VPN-to-SH:423: IPsec SA enc spi db38e2b9 key 16:7AC6134E6A36AE792428800B79CC2DC2 auth
20:50EEC2F6F23A0AC711679F61DC3EE6928FC967B9
ike 0:VPN-to-SH:8:VPN-to-SH:423: added IPsec SA: SPIs=fe02f80f/db38e2b9
```

.0. 隧道 UP。

```bash
ike 0:VPN-to-SH:8:VPN-to-SH:423: sending SNMP tunnel UP trap
ike 0:VPN-to-SH:8: enc
2700000C01000000640101022900002802000000C55D3BB2AD28F1C48AD025D070255A48E114948725D28D20941363F40BBCAFA62100000800004024
2C00002C0000002801030403FE02F80F0300000C0100000C800E008003000008030000020000000805000000 2D00001801000000070000100000FFFF
C0A80100C0A801FF0000001801000000070000100000FFFFC0A80000C0A800FF0706050403020107
```

.1. 响应方回复的数据包，IKE_AUTH 交互的第 2 个包，回复身份信息，感兴趣流，选定的 proposal。

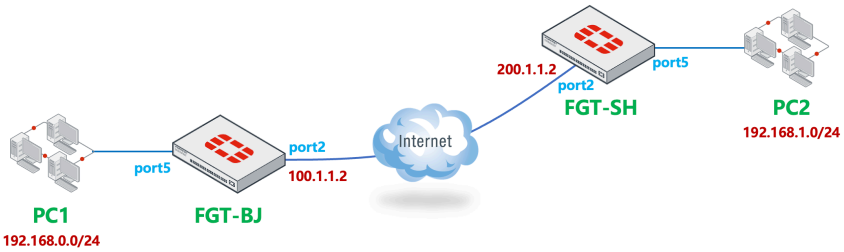
```bash
ike 0:VPN-to-SH:8: out
0640897749C1852079D668275E8FAB852E2023200000000 1000000E0240000C44AD9A518699109640D1EA67962E73585E17D1269A94910B3672BAAB5
706F353F950D18963DB076624C38DCF0C3D3CDF8D3EC991CF4CEE93D0EFA461E9E23C9362FD0F06B42E6251BDB1C2064DC0E17A8E99FE31AF55CB0E5
A6786EA694DADE5CBBFA5037A0297F77F38B19891910F33430D50350FAD2A716BF45BD0493A7F2DD0E2A3FC09074C464DC2DA2EFCB7CE00D94D4822D
8C81C5A15841DEB46B7017E379643FF00BA34247C836CC8F32565A082E1226E3ABB10379865E18FBACC661FF
ike 0:VPN-to-SH:8: sent IKE msg (AUTH_RESPONSE): 100.1.1.2:500→200.1.1.2:500, len=224, vrf=0,
id=0640897749c18520/79d668275e8fab85:00000001
ike 0:VPN-to-SH: link is idle 10 100.1.1.2→200.1.1.2:0 dpd=1 seqno=16d rr=0
```

.2. 关闭 Debug。

```bash
diagnose debug disable
```

IPSEC 协商失败举例

网络拓扑



收不到对端 IPSEC UDP port 500 的协商报文

在 FGT - BJ 端抓包可以看出 FGT - BJ 已经发送 IPSEC 协商报文给 FGT - SH，但是 FGT - BJ 没有收到 FGT - SH 的协商报文，因此问题可能是运营商阻断了 UDP port 500 端口的通信，也可能是 FGT - SH 的配置问题。同理可在 FGT - SH 端抓包查看。

```bash
FGT-BJ# diagnose sniffer packet any 'host 200.1.1.2 and udp port 500' 4
interfaces=[any]
filters=[host 200.1.1.2 and udp port 500]
8.239752 port2 out 100.1.1.2.500 → 200.1.1.2.500: udp 192
8.240249 port2 out 100.1.1.2.500 → 200.1.1.2.500: udp 380
8.240778 port2 out 100.1.1.2.500 → 200.1.1.2.500: udp 92
```

NAT 环境下收不到对端 IPSEC UDP port 4500 的协商报文

在 FGT - BJ 端抓包可以看出两端有 UDP port 500 的 IPSEC 报文交互，但是 FGT - BJ 没有收到 FGT - SH IPSEC UDP port 4500 的协商报文，因此问题可能是运营商阻断了 UDP port 4500 端口的通信，也可能是 FGT - SH 的配置问题。同理可在 FGT - SH 端抓包查看。

```bash
FGT-BJ# diagnose sniffer packet any 'host 200.1.1.2 and (udp port 500 or udp port 4500)' 4
interfaces=[any]
filters=[host 200.1.1.2 and (udp port 500 or udp port 4500)]
6.097481 port2 out 100.1.1.2.500 → 200.1.1.2.500: udp 380
6.098041 port2 in 200.1.1.2.500 → 100.1.1.2.500: udp 380
6.098294 port2 out 100.1.1.2.4500 → 200.1.1.2.4500: udp 112
6.098780 port2 out 100.1.1.2.4500 → 200.1.1.2.4500: udp 624
```

阶段一 Proposal 秘钥集不匹配

从 Debug 显示 ike Negotiate ISAKMP SA Error: ike 0:f0d8d95cbf3e993e/0000000000000000:13: no SA proposal chosen，对比 incoming proposal 和 my proposal 可以看出 IPSEC 阶段一（ike Negotiate ISAKMP SA）没有匹配的加密算法。

```bash
FGT-BJ# diagnose debug application ike -1
FGT-BJ# diagnose debug  enable

ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=f0d8d95cbf3e993e/0000000000000000 len=332 vrf=0
ike 0: in
F0D8D95CBF3E993E0000000000000000011002000000000000000014C0D0000640000000100000001000000580101000203000028010100000800B000100
0C0004000151808001000780 0E0080800300018002000680040 00E0000002802010000800B0001000C0004000151808001000780 0E0080800300018002
0006800400050D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C179D9215529D560D000014CD60464335DF21F87CFDB2
FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D83821A0F0AEAA8620D0000144485152D18B6BBCD0BE8A846
9579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE7F00D6C2D30D0000184048B7D56EBCE88525E7DE7F00
D6C2D3C0000000000000148299031757A36082C6A621DE00000000
ike 0:f0d8d95cbf3e993e/0000000000000000:13: responder: main mode get 1st message...
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID draft-ietf-ipsec-nat-t-ike-02 90CB80913EBB696E086381B5EC427B1F
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID DPD AFCAD71368A1F1C96B8696FC77570100
```

```
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:f0d8d95cbf3e993e/0000000000000000:13: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done
ike 0:f0d8d95cbf3e993e/0000000000000000:13: incoming proposal:
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 0:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_512.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 0:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_512.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: my proposal, gw VPN-to-SH:
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=256
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=256
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
```

```
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=256
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: proposal id = 1:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:   protocol id = ISAKMP:
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      trans_id = KEY_IKE.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:      encapsulation = IKE/none
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=256
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_HASH_ALG, val=SHA.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f0d8d95cbf3e993e/0000000000000000:13:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f0d8d95cbf3e993e/0000000000000000:13: ISAKMP SA lifetime=86400
ike 0:f0d8d95cbf3e993e/0000000000000000:13: negotiation failure
ike Negotiate ISAKMP SA Error: ike 0:f0d8d95cbf3e993e/0000000000000000:13: no SA proposal chosen
```

预共享秘钥不匹配

从 Debug 显示 probable pre-shared secret mismatch，可以判断是预共享秘钥不匹配

bash

```
FGT-BJ# diagnose debug application ike -1
FGT-BJ# diagnose debug  enable

ike shrank heap by 163840 bytes
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=dd4a4831edc1e725/0000000000000000 len=332 vrf=0
ike 0: in
DD4A4831EDC1E725000000000000000000011002000000000000000014C0D0000640000000100000001000000580101000203000028010100008 00B000100
0C0004000151808001000780 0E0080800300018002000480040 00E0000028020100008 00B0001000C000400015180800100078 00E0080800300018002
0004800400050D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C179D9215529D560D000014CD60464335DF21F87CFDB2
FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D83821A0F0AEAA8620D0000144485152D18B6BBCD0BE8A846
9579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE7F00D6C2D30D0000184048B7D56EBCE88525E7DE7F00
D6C2D3C000000000000014829903175 7A36082C6A621DE00000000
ike 0:dd4a4831edc1e725/0000000000000000:17: responder: main mode get 1st message ...
ike 0:dd4a4831edc1e725/0000000000000000:17: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:dd4a4831edc1e725/0000000000000000:17: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:dd4a4831edc1e725/0000000000000000:17: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:dd4a4831edc1e725/0000000000000000:17: VID draft-ietf-ipsec-nat-t-ike-02\n 90CB80913EBB696E086381B5EC427B1F
ike 0:dd4a4831edc1e725/0000000000000000:17: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:dd4a4831edc1e725/0000000000000000:17: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:dd4a4831edc1e725/0000000000000000:17: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:dd4a4831edc1e725/0000000000000000:17: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:dd4a4831edc1e725/0000000000000000:17: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:dd4a4831edc1e725/0000000000000000:17: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done
ike 0:dd4a4831edc1e725/0000000000000000:17: negotiation result
ike 0:dd4a4831edc1e725/0000000000000000:17: proposal id = 1:
ike 0:dd4a4831edc1e725/0000000000000000:17:   protocol id = ISAKMP:
ike 0:dd4a4831edc1e725/0000000000000000:17:      trans_id = KEY_IKE.
ike 0:dd4a4831edc1e725/0000000000000000:17:      encapsulation = IKE/none
ike 0:dd4a4831edc1e725/0000000000000000:17:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:dd4a4831edc1e725/0000000000000000:17:         type=OAKLEY_HASH_ALG, val=SHA2_256.
```

```
ike 0:dd4a4831edc1e725/0000000000000000:17:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:dd4a4831edc1e725/0000000000000000:17:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:dd4a4831edc1e725/0000000000000000:17: ISAKMP SA lifetime=86400
ike 0:dd4a4831edc1e725/0000000000000000:17: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0x9b57be0 10 100.1.1.2->200.1.1.2:500.
ike 0:VPN-to-SH:17: DPD negotiated
ike 0:VPN-to-SH:17: peer is FortiGate/FortiOS (v0 b0)
ike 0:VPN-to-SH:17: selected NAT-T version: RFC 3947
ike 0:VPN-to-SH:17: cookie dd4a4831edc1e725/7d0c610fb77157ce
ike 0:VPN-to-SH:17: out
DD4A4831EDC1E7257D0C610FB77157CE011002000000000000000C00D00003C000000010000000100000030010100010000002801010000800B000100
0C0004000151808001000780 0E0080803000180020004800 4000E0D0000144A131C81070358455C5728F20E95452F0D000014AFCAD71368A1F1C96B86
96FC775701000D0000148299031757A36082C6A621DE000000000D0000144048B7D56EBCE88525E7DE7F00D6C2D3000000184048B7D56EBCE88525E7DE
7F00D6C2D3C0000000
ike 0:VPN-to-SH:17: sent IKE msg (ident_r1send): 100.1.1.2:500->200.1.1.2:500, len=192, vrf=0,
id=dd4a4831edc1e725/7d0c610fb77157ce
ike 0: comes 200.1.1.2:500->100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=dd4a4831edc1e725/7d0c610fb77157ce len=380 vrf=0
ike 0: in
DD4A4831EDC1E7257D0C610FB77157CE041002000000000000000017C0A0001044FF0C6534555C243272FB0833E95A874D13A865B705BD1C94ED0D0543A
52972D5512DB656C442276401DC9FCFA3AA119208499DF00C5043FA108FE0BE3D38F2BF896E927F0E96454BC828886CC7CD8AB1022D19D60A9811B25CD
571A7E10188C6FD7E61CB9E64DE5D33A38F9BCF0B2C2E41C611D9AC5C047E2A03938BC6798683CE6208F43A6CA894407182B9CEE9A3B41FB58BC303932
6C180335E5044F7FA4D6A027426B609933E4B02BF09AA693AA64518B8050FD0ED70BC7AE87C0C88C09D58E68F4474F21E8D6C3F7D66D3C7A423E91278F
42D6B1BB014A53F215E8D1BEAFA2E1D1CD717DFA02A896BD83C525E5400993BA860A61A4207B66437C4EA603140000014657893D41BEBDBEA198635BC6A
09EF9914000024A66B51CDA1E16E10A0B69659421DF4143EF4394DF3E6A08E9E5BAB9DEBC5292C0000002406045FC70A8214BED832E183B81236F416DF
5770C32F118AD2F38B9F3CD54606
ike 0:VPN-to-SH:17: responder:main mode get 2nd message...
ike 0:VPN-to-SH:17: received NAT-D payload type 20
ike 0:VPN-to-SH:17: received NAT-D payload type 20
ike 0:VPN-to-SH:17: NAT not detected
ike 0:VPN-to-SH:17: generate DH public value request queued
ike 0:VPN-to-SH:17: compute DH shared secret request queued
ike 0:VPN-to-SH:17: out
DD4A4831EDC1E7257D0C610FB77157CE041002000000000000000017C0A0001042354F77BA3782481818EFCA017482C82BE52A8F85A6E36AF8C311C80EA
75EA9C3CAD3AB41C069BE71EFF4DFC076C72E0369E7C0B7668EF7A53B674B8E5CAE1755DEF61CB21927A45A1304EF0895714D3CB622F37B435C9E7EA0C
2D7980951C20CAA41CAEF5A44F61D30304B0C24EA82BB1FBBB20FF2C5245FBDCCC0D9102DAE56B2C8AA4E41DB5473AF7ADDA0CA5C1EE43CAE6BB1D8073
79CECF73F8D70849EFB353D51805FE16EA186604E87CAC595040E42F65DF18CAF08373C3C2A81D7F0BB8025139B5AD3F94239748EBB9FF337AE10C4FE4
15B5B93017A42D66B7574E406398BCBB7C235F88B382B94FC03638CB9BDB5F8A8D43764004B4B41EA86E381C14000014BDA49AFE0D6450572E7C41AE2B
C3CEF51400002406045FC70A8214BED832E183B81236F416DF5770C32F118AD2F38B9F3CD5460600000024A66B51CDA1E16E10A0B69659421DF4143EF4
394DF3E6A08E9E5BAB9DEBC5292C
ike 0:VPN-to-SH:17: sent IKE msg (ident_r2send): 100.1.1.2:500->200.1.1.2:500, len=380, vrf=0,
id=dd4a4831edc1e725/7d0c610fb77157ce
ike 0:VPN-to-SH:17: ISAKMP SA dd4a4831edc1e725/7d0c610fb77157ce key 16:308FD68288AC676140E7228F0D38783A
ike 0: comes 200.1.1.2:500->100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Identity Protection id=dd4a4831edc1e725/7d0c610fb77157ce len=108 vrf=0
ike 0: in
DD4A4831EDC1E7257D0C610FB77157CE05100201000000000000006C8FE002AE3B8049C7B1913FC10F2D97F44D646E05224E064DEC03481780B7A23820
5D754F3E9D56715677EE1755456CB2A7065507E04FDCFB454ED3A7C7AC0A77E146380BED18595E4EC503A89EDB11C2
ike 0:VPN-to-SH:17: responder: main mode get 3rd message...
ike 0:VPN-to-SH:17: dec
DD4A4831EDC1E7257D0C610FB77157CE05100201000000000000006C553BDCD748E60ACBE00D989D77561C67934BDA836A49273085DDDE950D5F328C84
5CF726C579351BED1BC286ED98D115D683ED3B0FB5FF6A97809437417B29D0B722E56F95A3EA9268565E3303B3C75D
ike 0:VPN-to-SH:17: parse error
ike 0:VPN-to-SH:17: probable pre-shared secret mismatch
```

Peer ID 不匹配

从 Debug 显示 ignoring IKE request, incorrect ID，可以判断 FGT - BJ 设置了对端的身份 peer ID，由于对端不匹配，因此协商失败。

```bash
FGT-BJ# diagnose debug application ike -1
FGT-BJ# diagnose debug  enable

ike 0: comes 200.1.1.2:500->100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=e6280452ac6300c3/0000000000000000 len=720 vrf=0
ike 0: in
E6280452AC6300C3000000000000000000011004000000000000002D0040000B400000001000000010000000A801010004030000280101000080 0B000100
0C0004000151808001000780 0E0080803000180020004800 4000E030000280301000080 0B0001000C0004000151808001000780 0E0100800300018002
00048004000E030000280501000080 0B0001000C0004000151808001000780 0E0080803000180020002800 4000E0000002807010000800B0001000C00
0400015180800100078 00E0100800300018002000280 04000E0A000104B6A00CB09789C8BC6146763CE20B649D18F4923FD7C80543ED2EF0EF97667D1C
FA0684F40AAE49A6734172BFDEB8594A7B8DF0D5BC9141C613DFC8D822AC73EAC167AB6412A1F22AE72A3B1328224F5377C57F094B6D2A952A483A4B8C
```

```
44787707C387C33EB7240CF5B3FB5B4C2144EF94B39E830D3EF2F2980C403A3817E0FBB2FC18425CD0498A73419AFA2BF3CADCB520CA2583DE90F55E72
E686898BF3177D3AEE60C7BC0F7AB35D7D7E06E2E319C8FAAD0E8822D81AAB7780EC837664E6A1772420F20B3C80760169600CEBDDF5A2EAAC57413912
18D4C104892F7823A400F837375AC6E86D811BCB68AA5D54C6463C4E609113C567E3CDE1E489A5A874050000245345D5C44232C8EA75FA163A9605835C
B173327C00AA6B3C79B3D087F9BE7F410D00000C01000000C80101020D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C
179D9215529D560D000014CD60464335DF21F87CFDB2FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D8382
1A0F0AEAA8620D0000144485152D18B6BBCD0BE8A8469579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE
7F00D6C2D30D0000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000000000148299031757A36082C6A621DE00000000
ike 0:e6280452ac6300c3/0000000000000000:24: responder: aggressive mode get 1st message ...
ike 0:e6280452ac6300c3/0000000000000000:24: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:e6280452ac6300c3/0000000000000000:24: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:e6280452ac6300c3/0000000000000000:24: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:e6280452ac6300c3/0000000000000000:24: VID draft-ietf-ipsec-nat-t-ike-02\n 90CB80913EBB696E086381B5EC427B1F
ike 0:e6280452ac6300c3/0000000000000000:24: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:e6280452ac6300c3/0000000000000000:24: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:e6280452ac6300c3/0000000000000000:24: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:e6280452ac6300c3/0000000000000000:24: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:e6280452ac6300c3/0000000000000000:24: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:e6280452ac6300c3/0000000000000000:24: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0::24: peer identifier IPV4_ADDR 200.1.1.2
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done
ike 0:VPN-to-SH: ignoring IKE request, incorrect ID.
ike 0:e6280452ac6300c3/0000000000000000:24: negotiation failure
ike Negotiate ISAKMP SA Error: ike 0:e6280452ac6300c3/0000000000000000:24: no SA proposal chosen
```

阶段二 Proposal 秘钥集不匹配

从 Debug 显示 ike Negotiate IPsec SA Error: ike 0:VPN-to-SH:28:23: no SA proposal chosen，对比 incoming proposal 和 my proposal 可以看出 IPSEC 阶段二（ike Negotiate IPsec SA Error）没有匹配的加密算法。

```bash
FGT-BJ# diagnose debug application ike -1
FGT-BJ# diagnose debug  enable

ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=f7204652fa935744/0000000000000000 len=656 vrf=0
ike 0: in
F7204652FA935744000000000000000001100400000000000000029004000B400000001000000010000000A801010004030000280201000080 0B000100
0C0004000151808001000780 0E0080803000180020004800400050300002804010000800B0001000C0004000151808001000780 0E0100800300018002
0004800400050300002806010000800B0001000C0004000151808001000780 0E0080803000180020002800400050000002808010000800B0001000C00
0400015180800100078 00E0100803000180020002800400050A0000C412447085117ADAC1FEBBD542C368680FEA237164357F20442AFC40C2A589DBA7
F32F33ED514F890D0E4FF2F6C7BEE902AE01CBB5CD541119097BB21863BADF1518CB892AA4A7F18FA33895136C38CF1714E4CD91A84C885C3C4F5B6236
07469BFF4370460765FEF16583F0A53838A32668DD758252861C401EDAD583A882849DFAD20E37FC7446DF38A7EB5D2A643726C99BDD7008293879D362
3F8317E15C2334995A051081FB9626802B5C179F22EAFF570FCA3C09E1BD616BCE8C8DF58719050000244E01F8ABA71441ECB0510C71D87D85DCEE37B3
927DE27397AE61CA69B36E242D0D00000C01000000C80101020D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C179D92
15529D560D000014CD60464335DF21F87CFDB2FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D83821A0F0A
EAA8620D0000144485152D18B6BBCD0BE8A8469579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE7F00D6
C2D30D0000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000000000148299031757A36082C6A621DE00000000
ike 0:f7204652fa935744/0000000000000000:28: responder: aggressive mode get 1st message ...
ike 0:f7204652fa935744/0000000000000000:28: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:f7204652fa935744/0000000000000000:28: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:f7204652fa935744/0000000000000000:28: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:f7204652fa935744/0000000000000000:28: VID draft-ietf-ipsec-nat-t-ike-02\n 90CB80913EBB696E086381B5EC427B1F
ike 0:f7204652fa935744/0000000000000000:28: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:f7204652fa935744/0000000000000000:28: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:f7204652fa935744/0000000000000000:28: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:f7204652fa935744/0000000000000000:28: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:f7204652fa935744/0000000000000000:28: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:f7204652fa935744/0000000000000000:28: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0::28: peer identifier IPV4_ADDR 200.1.1.2
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done
ike 0:f7204652fa935744/0000000000000000:28: negotiation result
ike 0:f7204652fa935744/0000000000000000:28: proposal id = 1:
ike 0:f7204652fa935744/0000000000000000:28:   protocol id = ISAKMP:
ike 0:f7204652fa935744/0000000000000000:28:      trans_id = KEY_IKE.
ike 0:f7204652fa935744/0000000000000000:28:      encapsulation = IKE/none
```

```
ike 0:f7204652fa935744/0000000000000000:28:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:f7204652fa935744/0000000000000000:28:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:f7204652fa935744/0000000000000000:28:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:f7204652fa935744/0000000000000000:28:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:f7204652fa935744/0000000000000000:28: ISAKMP SA lifetime=86400
ike 0:f7204652fa935744/0000000000000000:28: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0×9b57be0 10 100.1.1.2→200.1.1.2:500.
ike 0:VPN-to-SH:28: DPD negotiated
ike 0:VPN-to-SH:28: peer is FortiGate/FortiOS (v0 b0)
ike 0:VPN-to-SH:28: selected NAT-T version: RFC 3947
ike 0:VPN-to-SH:28: generate DH public value request queued
ike 0:VPN-to-SH:28: compute DH shared secret request queued
ike 0:VPN-to-SH:28: cookie f7204652fa935744/5b61d63b74cbb9f6
ike 0:VPN-to-SH:28: ISAKMP SA f7204652fa935744/5b61d63b74cbb9f6 key 16:0F91D8F5D8A2C9FA3A9088B189E03E25
ike 0:VPN-to-SH:28: out
F7204652FA9357445B61D63B74CBB9F60110040000000000000002110400003C0000000100000001000000300101000100000028020100008000B000100
0C000400015180800100078000E00808003000180020004800400050A0000C4F4B3253B46BD6F0E6565ECEED05034971D80ECF4F3CDC60FFDA3FEDB48C1
7FECCE8FF822AE90A5CCB6AF42639776B96DC37937512BC0A3F9BFFF2E928675A2C809A7EC994FD0EB20B67BE20D9B3BE4B6714191F4F0C7367C5CD642
6D044D32CAC8846133F49E58D54971E9A0AC21E87A84D89AE61272B054C00647076F653AF6C6C14169F85D90EA436DD94F0C74BACAD6DBB6A8B1C3891F
792351C7EB7DA6833B6B05882779827B8624D94F3B8CC9175C8CCE725AFAEB326D800797781699FE0500001448B6599BE05DE8169526D5559B9F3C9308
00000D020000007465737431 0D0000240A540CEA1B861ABB4EE4E8A7BF1241860187DB76D929109FC58DD68ED39DB45F140000144A131C81070358455C
5728F20E95452F140000249A6E313BA678326493465137F6BDA361E9B2381E17CE5CF19EE808B22346E67C0D000024ACC52C4EFEC900F50AD8EEB8FE3C
08EA6E18437072859ECC217AD6E29C5964A00D000014AFCAD71368A1F1C96B8696FC775701000D0000148299031757A36082C6A621DE000000000D0000
144048B7D56EBCE88525E7DE7F00D6C2D3000000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:VPN-to-SH:28: sent IKE msg (agg_r1send): 100.1.1.2:500→200.1.1.2:500, len=529, vrf=0,
id=f7204652fa935744/5b61d63b74cbb9f6
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=f7204652fa935744/5b61d63b74cbb9f6 len=172 vrf=0
ike 0: in
F7204652FA9357445B61D63B74CBB9F6081004010000000000000AC4368B8C00750D9E5D4364D4C8C3E8741F5D66ADB8F2716593C3DB039D13088AB5A
2E4D38FEA304CAB222731BC854E4506D50ABAFC2D456FDEB620E7927D161A6FA9709F473EA4EB963799383B67C3F1E82076A68ED37501BCC42E3BD01A2
11E2E5821A7785CE35C0527663C430D35266AF1BD1D8496ECFB32F5420C795F386B8C7F8D08611A2ECC3CF020C7D4D43D93E
ike 0:VPN-to-SH:28: responder: aggressive mode get 2nd response...
ike 0:VPN-to-SH:28: dec
F7204652FA9357445B61D63B74CBB9F6081004010000000000000AC140000246A0154E97988DFCFFB1D314FF8E895AE5C9ED93CB3B6FE44833A8B423F
639ED514000024ACC52C4EFEC900F50AD8EEB8FE3C08EA6E18437072859ECC217AD6E29C5964A00B0000249A6E313BA678326493465137F6BDA361E9B2
381E17CE5CF19EE808B22346E67C0000001C0000000101106002F7204652FA9357445B61D63B74CBB9F639536A88D14FCC07
ike 0:VPN-to-SH:28: received NAT-D payload type 20
ike 0:VPN-to-SH:28: received NAT-D payload type 20
ike 0:VPN-to-SH:28: received p1 notify type INITIAL-CONTACT
ike 0:VPN-to-SH:28: PSK authentication succeeded
ike 0:VPN-to-SH:28: authentication OK
ike 0:VPN-to-SH:28: NAT not detected
ike 0:VPN-to-SH:28: established IKE SA f7204652fa935744/5b61d63b74cbb9f6
ike 0:VPN-to-SH:28: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike 0:VPN-to-SH:28: processing INITIAL-CONTACT
ike 0:VPN-to-SH: flushing
ike 0:VPN-to-SH: flushed
ike 0:VPN-to-SH:28: processed INITIAL-CONTACT
ike 0:VPN-to-SH: set oper up
ike 0:VPN-to-SH:28: no pending Quick-Mode negotiations
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=f7204652fa935744/5b61d63b74cbb9f6:854d6010 len=476 vrf=0
ike 0: in
F7204652FA9357445B61D63B74CBB9F608102001854D6010000001DCD35083E7B123E3B5C7C5ECCE5C291BF2AB24505BA8045D3712FD03B3C6921B8879
E69C4F89C41ED3AA8F4D8C67F3CF1ED801F07862C20EE784387EF43372115C4C7ED2DE2269431C4D69AD61BE03C918555C69280EF4D9DE81C058FBAC44
0D4FAF0B507CDE4779CE282A0D12E7CA0AC0EF1C13D9F100C6C4DD849A1E12E11AEB8141356C9231AA43667F07C93232668531971F3BA330804BD63D96
3699AB7DB0B046B938470F1082B30E1E52D5C1B2F3E23ECC8C19F0B7281FF51E7A06F39C83D1897A91E3D2BABA0233D1154DAEAD2C5E15D7363063CB4B
BB1125536368DD508F6CA3230FF80980D26A696B970B3CF5846974FC7F6AF48891A118EA8DAA9A5022FA69958095DDF55BAC8A30676A780B99B398E9DD
CC84A2E8DC79E68240437D7C28D7D6B76A4155BE29F3018195D8CF8B7CA339308F09C1744459F751A837DB85A3BD7E332374577376EAA9662E5692439C
FCB0AF2D06833CC88D9CC371854AE33272D9469971B48A4C62BD2413A6F814DAD70C4680B0942B67E5EF8CC1F9158573B40B4B6B9BC5E8D7EC7B32285E
FFE51935D5F011493CA822378ED42D1F04FEF702FB8371DAF7902BE30CFA6C0185A42145D0DE9AA73C5ACB5C111CA8C9BB
ike 0:VPN-to-SH:28:23: responder received first quick-mode message
ike 0:VPN-to-SH:28: dec
F7204652FA9357445B61D63B74CBB9F608102001854D6010000001DC01000024EA4C99A2785B85A08F8AD5DAEDD8227E2E23F7157394766515BE32574D
F3324A0A000058000000010000000100000004C0103040236826FEB03000020010C0000800100018002A8C0800400018006010080050002800300 0E0000
0020020C0000800100018002A8C0800400018006010080050005800300 0E04000014153BB1F0FE1EC2D16347F6E59D9279EE050001044B6A54A5DE7C5B
F9DD546C32A28508FEFBC9056F465563F366A20AA5318FFC514DF0800CD028DC7422634E749A1B85A4CA414501B87DE19FDF9F6674BFAB35E224D5460F
DD6A26D11A800DF0386403F850876F93BE32971442E8E4F235F307FED3C53AF74294B6989BDF38438A5595A509423235B2B820D57B344870F03F999C71
2458DD13A2A36047370CB5917097D54D6B7E8D320BADD3B292FF2D089AD0923DE2AE9116B7CB98DFCAED4A97D7DF7FA9391EAD63F121C4C24BA8E11FBD
590F2459296C6B750745C67B165CE6A0D16A3A9E6CDC2621470412544C4C29BF6912EA9BFEB8811E08459CF098E6C6401228568B632BB267E10593E6AF
1A9272724A0500001004000000C0A80100FFFFFF00000000 1004000000C0A80000FFFFFF002BDD3013B2168B798A40790B
```

```
ike 0:VPN-to-SH:28:23: peer proposal is: peer:0:192.168.1.0-192.168.1.255:0, me:0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:28:VPN-to-SH:23: trying
ike 0:VPN-to-SH:28:VPN-to-SH:23: matched phase2
ike 0:VPN-to-SH:28:VPN-to-SH:23: autokey
ike 0:VPN-to-SH:28:VPN-to-SH:23: my proposal:
ike 0:VPN-to-SH:28:VPN-to-SH:23: proposal id = 1:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   PFS DH group = 14
ike 0:VPN-to-SH:28:VPN-to-SH:23:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:28:VPN-to-SH:23:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:28:VPN-to-SH:23:         type = AUTH_ALG, val=SHA2_512
ike 0:VPN-to-SH:28:VPN-to-SH:23: proposal id = 2:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   PFS DH group = 5
ike 0:VPN-to-SH:28:VPN-to-SH:23:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:28:VPN-to-SH:23:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:28:VPN-to-SH:23:         type = AUTH_ALG, val=SHA2_512
ike 0:VPN-to-SH:28:VPN-to-SH:23: incoming proposal:
ike 0:VPN-to-SH:28:VPN-to-SH:23: proposal id = 1:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   protocol id = IPSEC_ESP:
ike 0:VPN-to-SH:28:VPN-to-SH:23:   PFS DH group = 14
ike 0:VPN-to-SH:28:VPN-to-SH:23:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:28:VPN-to-SH:23:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:28:VPN-to-SH:23:         type = AUTH_ALG, val=SHA1
ike 0:VPN-to-SH:28:VPN-to-SH:23:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:VPN-to-SH:28:VPN-to-SH:23:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:VPN-to-SH:28:VPN-to-SH:23:         type = AUTH_ALG, val=SHA2_256
ike 0:VPN-to-SH:28:VPN-to-SH:23: negotiation failure
ike Negotiate IPsec SA Error: ike 0:VPN-to-SH:28:23: no SA proposal chosen
ike 0:VPN-to-SH:23: info_send_n2, type 14, peer SPI 36826feb
ike 0:VPN-to-SH:28: enc
F7204652FA9357445B61D63B74CBB9F608100501D001250C000000500B000024DA181A96FDB1E05A810C664047DA00EE6AC66F194C9549253F322B804C
59869000000010000000010304000E36826FEB
ike 0:VPN-to-SH:28: out
F7204652FA9357445B61D63B74CBB9F608100501D001250C0000005C3A5150D6C379B62DDE5EB654F552CE2C5FB1022E78DE45FD65FA1FDE284B57EA3A
C4DD593068FECBE6A60DBADB418C54B61E2C9C2C415BB5323CB41283CFFA87
ike 0:VPN-to-SH:28: sent IKE msg (p2_notify_14): 100.1.1.2:500→200.1.1.2:500, len=92, vrf=0,
id=f7204652fa935744/5b61d63b74cbb9f6:d001250c
ike 0:VPN-to-SH:28: error processing quick-mode message from 200.1.1.2 as responder
ike shrank heap by 159744 bytes
```

感兴趣流不匹配

从 Debug 显示 specified selectors mismatch，对比两端感兴趣流 peer: type = 7/7, local = 0:192.168.0.0-192.168.0.255:0, remote = 0:192.168.1.0-192.168.1.255:0 和 mine: type = 7/7, local = 0:192.168.0.0-192.168.0.255:0, remote = 0:192.168.10.0-192.168.10.255:0 是不匹配的，因此协商失败。

bash

```
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=9bbf928209f22769/0000000000000000 len=656 vrf=0
ike 0: in
9BBF928209F22769000000000000000000011004000000000000000290040000B400000001000000010000000A801010004030000280101000080B000100
0C000400015180800100078 00E00808003000180020004800400050300002802010000800B0001000C000400015180800100078 00E0100800300018002
000480040005030000280301000800B0001000C000400015180800100078 00E00808003000180020002800400050000002804010000800B0001000C00
040001518080010007800E0100800300018002000280040005 0A0000C499A094C58E0633FF797B3EF770CB35722E912E0533C5C6657BDC1883DB5EB8B8
D704EE6D5405EAD5CAE35B54698AAB9E4B2B18D285BE7A0F3C4BF05DB4E86E8F080A723FE26C667B17D52F1C95F1E8960A94377D36CBDBADC70D355E7B
B3557D2BAF27572B634183ACA7914686E2B09871469BAF9DA137915F60C17DC1FABCA750E380641AB53665BFA7D3CFB3037B8DD8600E92551B6733CE4A
2DE9D528A39E495D23BEB48455FFD88570E68DA43E9318D348675FBB93AF3362B6DFBAC499C405000024EA3AFD1354F3F8794D04364A1DA75C09290F12
1FD2A23F751D01BE3635C3C1980D00000C01000000C80101020D0000144A131C81070358455C5728F20E95452F0D0000147D9419A65310CA6F2C179D92
15529D560D000014CD60464335DF21F87CFDB2FC68B6A4480D00001490CB80913EBB696E086381B5EC427B1F0D00001416F6CA16E4A4066D83821A0F0A
EAA8620D0000144485152D18B6BBCD0BE8A8469579DDCC0D000014AFCAD71368A1F1C96B8696FC775701000D0000144048B7D56EBCE88525E7DE7F00D6
C2D30D0000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000000000148299031757A36082C6A621DE00000000
ike 0:9bbf928209f22769/0000000000000000:29: responder: aggressive mode get 1st message...
ike 0:9bbf928209f22769/0000000000000000:29: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:9bbf928209f22769/0000000000000000:29: VID draft-ietf-ipsec-nat-t-ike-03 7D9419A65310CA6F2C179D9215529D56
ike 0:9bbf928209f22769/0000000000000000:29: VID draft-ietf-ipsec-nat-t-ike-02 CD60464335DF21F87CFDB2FC68B6A448
ike 0:9bbf928209f22769/0000000000000000:29: VID draft-ietf-ipsec-nat-t-ike-02\n 90CB80913EBB696E086381B5EC427B1F
ike 0:9bbf928209f22769/0000000000000000:29: VID draft-ietf-ipsec-nat-t-ike-01 16F6CA16E4A4066D83821A0F0AEAA862
ike 0:9bbf928209f22769/0000000000000000:29: VID draft-ietf-ipsec-nat-t-ike-00 4485152D18B6BBCD0BE8A8469579DDCC
ike 0:9bbf928209f22769/0000000000000000:29: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:9bbf928209f22769/0000000000000000:29: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
```

```
ike 0:9bbf928209f22769/0000000000000000:29: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:9bbf928209f22769/0000000000000000:29: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0::29: peer identifier IPV4_ADDR 200.1.1.2
ike 0: cache rebuild start
ike 0:VPN-to-SH: local:100.1.1.2, remote:200.1.1.2
ike 0:VPN-to-SH: cached as static-ddns.
ike 0: cache rebuild done
ike 0:9bbf928209f22769/0000000000000000:29: negotiation result
ike 0:9bbf928209f22769/0000000000000000:29: proposal id = 1:
ike 0:9bbf928209f22769/0000000000000000:29:   protocol id = ISAKMP:
ike 0:9bbf928209f22769/0000000000000000:29:      trans_id = KEY_IKE.
ike 0:9bbf928209f22769/0000000000000000:29:      encapsulation = IKE/none
ike 0:9bbf928209f22769/0000000000000000:29:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:9bbf928209f22769/0000000000000000:29:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:9bbf928209f22769/0000000000000000:29:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:9bbf928209f22769/0000000000000000:29:         type=OAKLEY_GROUP, val=MODP1536.
ike 0:9bbf928209f22769/0000000000000000:29: ISAKMP SA lifetime=86400
ike 0:9bbf928209f22769/0000000000000000:29: SA proposal chosen, matched gateway VPN-to-SH
ike 0:VPN-to-SH: created connection: 0×9b57be0 10 100.1.1.2→200.1.1.2:500.
ike 0:VPN-to-SH:29: DPD negotiated
ike 0:VPN-to-SH:29: peer is FortiGate/FortiOS (v0 b0)
ike 0:VPN-to-SH:29: selected NAT-T version: RFC 3947
ike 0:VPN-to-SH:29: generate DH public value request queued
ike 0:VPN-to-SH:29: compute DH shared secret request queued
ike 0:VPN-to-SH:29: cookie 9bbf928209f22769/1912fc364606cdf2
ike 0:VPN-to-SH:29: ISAKMP SA 9bbf928209f22769/1912fc364606cdf2 key 16:6F0A5C5D7ABE1862A0754D4F67F6C309
ike 0:VPN-to-SH:29: out
9BBF928209F227691912FC364606CDF2011004000000000000000211040000 3C000000100000001000000300101000100000028010100008 00B000100
0C0004000151808001000780 0E0080030001800200048040005 0A0000C4789B20D8E6155283C87B6C929A2CE7FC886FDB69F26FE13BB5BCD92C73B0
F2BDC7A495FB095CC5CED439F41DB332A22594A9128635D392BB8D511BA38FF3984BCD092FBC7AF4E22FC8CD25BF91FA6D0334B12D4ED2FAC4FD061A7C
199BFC6C19613645D9729D57D80003CF347B5B51D545EE9B5560904DC8A071A9291F42F948A61CDDB99915C0757FE13411153AF6CB42FB8BE04873F606
280B3AFDCB7B817A1D4106364F934F64936DBB595681EEF360CC887FCBD456657FA8497FCA18471A05000014FAFD06AA2EB8122B83A734FAF066A39F08
00000D020000007 4657374310D0000249471A719861E95E1CBF1723784350CC0876E57B07E6FA74D06A0B3F4F81AD3F2140000144A131C81070358455C
5728F20E95452F1400002493E1BF9E618E0E5225FEE63AB09074A2ED8DADEDE66B1D79F009C96C006B13E60D000024CB2EC39A82D1FEE194284F072DA1
FE064954BC8059215C114AE94739631B63C40D000014AFCAD71368A1F1C96B8696FC775701000D0000148299031757A36082C6A621DE000000000D0000
144048B7D56EBCE88525E7DE7F00D6C2D3000000184048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:VPN-to-SH:29: sent IKE msg (agg_r1send): 100.1.1.2:500→200.1.1.2:500, len=529, vrf=0,
id=9bbf928209f22769/1912fc364606cdf2
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Aggressive id=9bbf928209f22769/1912fc364606cdf2 len=172 vrf=0
ike 0: in
9BBF928209F227691912FC364606CDF2081004010000000000000000 ACF2D87D3D3C57A87AA9FBEEFD034859AFC9AF00BEBEB1BB1FA731127B48EAD6C7AA
5B0FA8D84E6446D502E242937B83F222A23BD25C31095752FEFF3C97C3095B698EE63122D4E47925CD95D7F1FF2B793BA587D4EC882D3D10804C395C24
CB12AEFC208B3C7DB4793889FB5CDAD64831290EAA5AE65FD7DE10AE60D80109162F1F9C769F1F57059FCED85A10BCA12E70
ike 0:VPN-to-SH:29: responder: aggressive mode get 2nd response...
ike 0:VPN-to-SH:29: dec
9BBF928209F227691912FC364606CDF2081004010000000000000000 AC14000024A31F2CDD20D257FD8CB06D9FEA96472FD20E34599130F73928E0E24B80
D81FE914000024CB2EC39A82D1FEE194284F072DA1FE064954BC8059215C114AE94739631B63C40B00002493E1BF9E618E0E5225FEE63AB09074A2ED8D
ADEDE66B1D79F009C96C006B13E60000001C0000000 1011060029BBF928209F227691912FC364606CDF279B09DF11828DE07
ike 0:VPN-to-SH:29: received NAT-D payload type 20
ike 0:VPN-to-SH:29: received NAT-D payload type 20
ike 0:VPN-to-SH:29: received p1 notify type INITIAL-CONTACT
ike 0:VPN-to-SH:29: PSK authentication succeeded
ike 0:VPN-to-SH:29: authentication OK
ike 0:VPN-to-SH:29: NAT not detected
ike 0:VPN-to-SH:29: established IKE SA 9bbf928209f22769/1912fc364606cdf2
ike 0:VPN-to-SH:29: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike 0:VPN-to-SH:29: processing INITIAL-CONTACT
ike 0:VPN-to-SH: flushing
ike 0:VPN-to-SH: flushed
ike 0:VPN-to-SH:29: processed INITIAL-CONTACT
ike 0:VPN-to-SH: set oper up
ike 0:VPN-to-SH:29: no pending Quick-Mode negotiations
ike 0: comes 200.1.1.2:500→100.1.1.2:500,ifindex=10,vrf=0....
ike 0: IKEv1 exchange=Quick id=9bbf928209f22769/1912fc364606cdf2:29a1bf3f len=412 vrf=0
ike 0: in
9BBF928209F227691912FC364606CDF20810200129A1BF3F0000019CA256FBF54CF315B0F0D3EBA8234DEF95082641935CBC5CF69FD677D3657FB02E59
1720692F37064413173465 55CA0F5FC750F4202A8459C159C3892443B4A1537AA863AD33883432E73F63C1406D12043F142C79E1BAAA3310248865F52B
E530725EA7DBF2A00E9E5E5A1C3942D96BD979F2FD4C43071680651F142C015A47351772636C667AD010EAA6A9971A9D52B2553E954C9590FC03C7D252
600CDBEB16E4CB8E913AA3992A2986EE68D44774CD5F1F253BF07FB7AF649189E2D8EF982FEC6EBFD89C0F2B6CBD5EDFDD65A93353A3497AA73932C546
DBDD94F900977EDCBF058144FA0B69508000E017A97BB075968D854547FC55ECFA841B356563E32D41413B3813B0DB32B3E4C8CDE3DF159F96F6B887D2
AB5480D0A36FBFD657A1F1D89868DC78860903A1CDA173B655022112B15F5DE6A14430D998F230F041355C61D20829D9BA9CAF3CE57D58DD8425CD37E0
```

```
A9BEE5514FDB7159AFB8398FDAFCB077499CA00FA4FCA9F1BC9A5B2208DFE69B2C041C85442669A6A1D23E8133AD
ike 0:VPN-to-SH:29:25: responder received first quick-mode message
ike 0:VPN-to-SH:29: dec
9BBF928209F227691912FC364606CDF20810200129A1BF3F0000019C0100002458D9086AFB973BD12DB26A7870A43555F16C84F66FDA552401520C9283
2D79BE0A000058000000001000000010000004C0103040236826FEC03000020010C0000800100018002A8C0800400018006010080050002800300050000
0020020C0000800100018002A8C0800400018006010080050005800300050400001 4062F996DD325B8E32EB8B8EDDE3D48C4050000C4D0A802E07811F1
5FB57E6A8DC7C46D9E4EC6BB5580F531B24787DCC36DB6A9B2D04B13A0FE462CC4348B4C3434CBFC3E887FCF6BD982ABF9E06E24AB4B3A88C6416FCE8B
C58A45F97226F2F59C63758AFC7112483C3C4BF357F35DF1351267ED3004C6A8CAE54F7FFAEC67482CD3FDC9CC4C53A176B4A3C3DC6D62A2948C08071A
8D5B616A55284CD82A1D89E021E67A5A05AFDD3B3F2FEB0DC8251B8E70D4ED71E83655B7E6E8CFB5BFF09E433594BF8D4CE68CDC3656BAC89C067E84A3
BA9A0500001004000000C0A80100FFFFFF00000000 1004000000C0A80000FFFFFF005F66F91E52BA59DB99E4F90B
ike 0:VPN-to-SH:29:25: peer proposal is: peer:0:192.168.1.0-192.168.1.255:0, me:0:192.168.0.0-192.168.0.255:0
ike 0:VPN-to-SH:29:VPN-to-SH:25: trying
ike 0:VPN-to-SH:29:25: specified selectors mismatch
ike 0:VPN-to-SH:29:25: peer: type=7/7, local=0:192.168.0.0-192.168.0.255:0, remote=0:192.168.1.0-192.168.1.255:0
ike 0:VPN-to-SH:29:25: mine: type=7/7, local=0:192.168.0.0-192.168.0.255:0, remote=0:192.168.10.0-192.168.10.255:0
ike 0:VPN-to-SH:29:25: no matching phase2 found
ike 0:VPN-to-SH:29:25: failed to get responder proposal
ike 0:VPN-to-SH:29: error processing quick-mode message from 200.1.1.2 as responder
```

IKEv2 重协商失败

1. 使用 IKEv2，且感兴趣流只有一对保护网段时，一定要注意两端二阶段的 PFS 配置是否一致，如果不一致，可能会导致第一次 IPSec 协商成功，但后续重协商时失败。
2. 这是由于 IKEv2 在第一次协商时，二阶段的 SA 包含在 IKE_AUTH 报文中，而 IKE_AUTH 中携带的 SA 是无法使用 PFS 的，所以即使两端 PFS 配置不一致（例如一端开启另一端不开启，或配置的 PFS DH 组不一致），第一次协商连接也是可以正常建立的。
3. 但重协商（rekey）时，使用的是 Create_Child_SA 报文携带 SA 信息，Create_Child_SA 会按照用户配置携带 PFS 信息，这时如果两端的 PFS 配置不一致，会导致 IKEv2 重协商时协商失败。
4. 使用如下命令（childless-ike）可以强制 FortiGate 使用 Child SA 携带二阶段 SA（IKE_AUTH 中不携带二阶段 SA，所有二阶段 SA 必须通过 Create_Child_SA 协商，注意使用此功能时对端设备也必须支持并开启此功能）。

```bash
config vpn ipsec phase1-interface
    edit xxxx
        set ike-version 2
        set childless-ike enable
    next
end
```

使用附加 IP 时隧道无法建立

问题现象

1. 在本端 FortiGate 的接口 wan1（`202.1.1.2`）上配置了附加 IP（Secondary IP）`202.1.1.3`。

```bash
config system interface
    edit "wan1"
        set vdom "root"
        set ip 202.1.1.2 255.255.255.0
        config secondaryip
            edit 1
                set ip 202.1.1.3 255.255.255.0
            next
        end
    next
end
```

2. IPSec 连接绑定接口为 wan1。

```bash
config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        set type dynamic
        set interface "wan1"
        set ike-version 2
......
    next
end

config vpn ipsec phase2-interface
    edit "IPSec_Dialup"
        set phase1name "IPSec_Dialup"
        set replay disable
    next
end
```

3. 对端设备将 VPN 隧道的 Remote Gateway 配置为该附加 IP（`202.1.1.3`）时，隧道无法正常建立，开启 IKE Debug 后出现 `no proposal chosen` 报错（即使对端发起的安全提议和本端配置的安全提议完全一致）：

```bash
comes 39.183.172.30:64916→202.1.1.3:500,ifindex=41,vrf=0,len=373....
......
ike V=root:0:93bce19daab14410/0000000000000000:3056: incoming proposal:
ike V=root:0:93bce19daab14410/0000000000000000:3056: proposal id = 1:
ike V=root:0:93bce19daab14410/0000000000000000:3056:   protocol = IKEv2:
ike V=root:0:93bce19daab14410/0000000000000000:3056:      encapsulation = IKEv2/none
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=INTEGR, val=AUTH_HMAC_SHA_96
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=DH_GROUP, val=ECP384.
......
ike V=root:0:93bce19daab14410/0000000000000000:3056: my proposal, gw IPSec_Dialup:
ike V=root:0:93bce19daab14410/0000000000000000:3056: proposal id = 1:
ike V=root:0:93bce19daab14410/0000000000000000:3056:   protocol = IKEv2:
ike V=root:0:93bce19daab14410/0000000000000000:3056:      encapsulation = IKEv2/none
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=INTEGR, val=AUTH_HMAC_SHA_96
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:93bce19daab14410/0000000000000000:3056:         type=DH_GROUP, val=ECP384.
ike V=root:0:93bce19daab14410/0000000000000000:3056: lifetime=86400
......
ike V=root:0:93bce19daab14410/0000000000000000:3056: no proposal chosen
```

问题原因

- FortiGate 默认使用接口的主 IP 进行 IKE 协商，当需要使用附加 IP 建立 VPN 隧道时，必须在 Phase 1 配置中通过 `local-gw` 参数显式指定。
- 当对端发起到附加 IP 的 VPN 连接时，本端 FortiGate 仍会尝试用主 IP 进行协商，导致 IP 地址不匹配，最终报 `no proposal chosen` 错误。

解决方法

1. 在 IPSec 一阶段配置中，通过 `local-gw` 显式指定附加 IP，使 FortiGate 使用该 IP 进行 IKE 协商：

> **重要**
>
> `local-gw` 指定的 IP 必须已在对应接口上完成配置（主 IP 或附加 IP 均可）。




```bash
config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        set interface "wan1"
        set local-gw 202.1.1.3
    next
end
```

2. 配置完成后，对端与 FortiGate 建立 IPSec 时，IKE Debug 中显示协商成功：

```bash
......
ike V=root: SA proposal chosen, matched gateway IPSec_Dialup
ike V=root:0:Tunnel1:Tunnel1: created connection: 0×b5be700 9 39.183.172.30→202.1.1.3:1012
```

多个拨号隧道使用相同 IP 池

问题描述

当多个拨号（dial-up）IPsec 隧道配置了相同的 IP 地址池时，即使每个隧道配置了不同的用户组，同一时间也只能有一个 VPN 用户保持连接。第一个连接的 VPN 用户会在第二个用户连接后立即被断开。

原因分析

这是 FortiOS 的设计行为：相同的 IP 池不能在 FortiGate 上的多个 IPsec 隧道中同时使用。

如果在不同的拨号 IPsec 隧道上配置了相同的 IP 池，如下所示：

```bash
config vpn ipsec phase1-interface
    edit <tunnel_A>
        set type dynamic
        set authusrgrp <group_A>
        set ipv4-start-ip 10.180.1.1
        set ipv4-end-ip 10.180.1.10
    next
    edit <tunnel_B>
        set type dynamic
        set authusrgrp <group_B>
        set ipv4-start-ip 10.180.1.1
        set ipv4-end-ip 10.180.1.10
    next
end
```

在 IKE 协商过程中，IKE 守护进程会根据 phase1 中配置的用户组将 IPsec 隧道映射到用户。用户认证成功后，IP 池中第一个可用的 IP 会被分配给该用户。当第二个用户连接 VPN 并认证成功后，IKE 守护进程会尝试将该隧道 IP 池中的第一个 IP 分配给第二个用户，此时会释放第一个已连接 VPN 用户的 IP 地址。

结果就是：第一个已连接的 VPN 用户被断开，第二个 VPN 用户成功连接。

问题配置举例

以下配置示例使用两个不同的用户组（本地用户和 LDAP 组），debug 输出展示了当两个隧道配置相同 IP 池时，第一个 VPN 用户在第二个用户连接后被断开的过程。

配置用户组

为每个拨号 IPsec 隧道配置不同的用户组：

```bash
config user local
    edit "tester"
        set type password
        set passwd *****
    next
end

config user group
    edit "ldap.users"
        set member "fortiserver.ldap" "zilan"
        config match
            edit 1
                set server-name "fortiserver.ldap"
                set group-name "CN=ldap.group,DC=fortiserver,DC=com"
            next
        end
    next
    edit "local.users"
        set member "tester"
    next
end
```

配置拨号 IPsec 隧道

为不同用户组配置两个拨号 IPsec 隧道（phase1 和 phase2）：

```bash
config vpn ipsec phase1-interface
    edit "fct.1"
        set type dynamic
        set interface "port4"
        set mode aggressive
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256
        set xauthtype auto
        set authusrgrp "local.users"        # ← 本地用户组
        set ipv4-start-ip 10.180.1.1
        set ipv4-end-ip 10.180.1.10
        set psksecret ENC *****
    next
    edit "fct.2"
        set type dynamic
        set interface "port4"
        set mode aggressive
        set mode-cfg enable
        set proposal aes128-sha1 aes256-sha1
        set xauthtype auto
        set authusrgrp "ldap.users"         # ← LDAP 用户组
        set ipv4-start-ip 10.180.1.1
        set ipv4-end-ip 10.180.1.10
        set psksecret ENC *****
    next
end

config vpn ipsec phase2-interface
    edit "fct.1"
        set phase1name "fct.1"
        set proposal aes128-sha256
        set dhgrp 2
    next
    edit "fct.2"
        set phase1name "fct.2"
        set proposal aes128-sha256
        set dhgrp 5
    next
end
```

Debug 分析

第一个用户连接：本地用户 `tester` 连接 VPN，IKE 守护进程从 IP 池中分配第一个 IP 10.180.1.1：

```bash
2025-06-30 13:59:39.847399 ike V=root:0:fct.1:1720: add route 10.180.1.1/255.255.255.255 gw 10.180.1.1 oif fct.1(1364)
metric 15 priority 1
2025-06-30 13:59:39.849412 ike V=root:0:fct.1_0:1065:fct:1720: tunnel 1 of VDOM limit 0/0
2025-06-30 13:59:39.850493 ike V=root:0:fct.1_0:1065:fct:1720: add IPsec SA: SPIs=14b059d8/08b06a3f
```

通过 `diagnose vpn ike gateway list` 确认用户 `tester` 已连接并被分配 IP 10.180.1.1：

```bash
# diagnose vpn ike gateway list

vd: root/0
name: fct.1_0
...
xauth-user: tester                                  # ← 本地用户
assigned IPv4 address: 10.180.1.1/255.255.255.255  # ← 分配的 IP
...
```

第二个用户连接：LDAP 用户 `zilan` 连接 VPN 后，IKE 守护进程将路由从 fct.1 移动到 fct.2，释放第一个用户的 IP 并重新分配给第二个用户：

```bash
2025-06-30 13:59:54.415801 ike V=root:0:fct.1:1720: moving route 10.180.1.1/255.255.255.255 oif fct.1(1364) metric 15 priority 1 to 0:fct.2:1727
                                                  # ↑ 将路由从 fct.1 移动到 fct.2

2025-06-30 13:59:54.417456 ike V=fct.1:0:fct.1:1720: del route 10.180.1.1/255.255.255.255 tunnel 10.180.1.1 oif fct.1(1364) metric 15 priority 1
                                                   # ↑ 删除 fct.1 的路由

2025-06-30 13:59:54.419296 ike V=root:0:fct.1_0: going to be deleted
                                                   # ↑ fct.1 隧道准备删除

2025-06-30 13:59:54.423117 ike V=root:0:fct.1_0: deleting IPsec SA with SPI 08b06a3f
                                                   # ↑ 删除 fct.1 的 IPsec SA

2025-06-30 13:59:54.464700 ike V=root:0:fct.1_0: mode-cfg release 10.180.1.1/255.255.255.255
                                                   # ↑ 释放第一个用户的 IP 地址

2025-06-30 13:59:54.466634 ike V=root:0:fct.2:1727: add route 10.180.1.1/255.255.255.255 gw 10.0.0.12 oif fct.2(1362) metric 15 priority 1
                                                   # ↑ 为第二个用户在 fct.2 上添加路由
```

通过 `diagnose vpn ike gateway list` 确认用户 `zilan` 已连接并使用了同一个 IP 10.180.1.1：

```bash
# diagnose vpn ike gateway list

vd: root/0
name: fct.2_0
...
xauth-user: zilan                                  # ← LDAP 用户
assigned IPv4 address: 10.180.1.1/255.255.255.255  # ← 使用了同一个 IP
...
```

解决方法

为每个拨号 IPsec 隧道配置不同的、不重叠的 IP 地址池：

```bash
config vpn ipsec phase1-interface
    edit "fct.1"
        set type dynamic
        set authusrgrp "local.users"
        set ipv4-start-ip 10.180.1.1
        set ipv4-end-ip 10.180.1.10       # ← IP 池 1: 10.180.1.1 - 10.180.1.10
    next
    edit "fct.2"
        set type dynamic
        set authusrgrp "ldap.users"
        set ipv4-start-ip 10.180.2.1
        set ipv4-end-ip 10.180.2.10       # ← IP 池 2: 10.180.2.1 - 10.180.2.10
    next
end
```

IPSec 命名注意事项

注意事项 1

ADVPN 环境下，有时候会遇到所有 Spoke 与 Hub 的 IPSec 连接建立成功，但 Spoke 之间在流量触发后，Shortcut 连接无法建立的情况。

问题现象

1. 检查 Spoke 与 Hub 间的 IPSec 连接状态是否正常。
2. 在 Spoke 上执行 IKE Debug，检查输出日志，如下所示，Spoke 已经收到了 Hub 返回的 shortcut-offer：

```
# diagnose debug application ike -1
# diagnose debug enable
```

```
ike 0:FGT-ADVPN-HUB-1:17: notify msg received: SHORTCUT-OFFER
ike 0:FGT-ADVPN-HUB-1: shortcut-offer 172.26.22.22→172.10.10.1 psk 64 ppk 0 ver 1 mode 0
ike 0 looking up shortcut by addr 172.10.10.1, name FGT-ADVPN-HUB-1
```

3. 问题出现在另一个 Spoke 发送的 shortcut-reply 后，当 Spoke 收到另一个 Spoke 发送的 shortcut-reply（经由 Hub 转发）后，准备与其建立 Shortcut 连接，但由于 Shortcut 连接的名字过长（Shortcut 连接的名字是在与 Hub 的连接名称基础上加上字符“_[number]”，FGT - ADVPN - HUB - 1_0），导致无法发起 Shortcut 连接。

```bash
ike 0:FGT-ADVPN-HUB-1:17: notify msg received: SHORTCUT-REPLY
ike 0:FGT-ADVPN-HUB-1: recv shortcut-reply 17386919350863387436 3d31b000f3a4cb20/918e5c255a8451ca 10.10.50.1 to
172.26.22.22 psk 64 ppk 0
 ver 1 mode 0 ext-mapping 0.0.0.0:0
ike 0:FGT-ADVPN-HUB-1: iif 28 172.10.10.1→172.26.22.22 route lookup oif 12 root
ike 0:FGT-ADVPN-HUB-1: shortcut-reply received from 10.10.50.1:500, local-nat=no, peer-nat=no
ike 0:FGT-ADVPN-HUB-1: created connection: 0×dd3cdb0 5 10.10.10.2→10.10.50.1:500.
ike 0:FGT-ADVPN-HUB-1: adding new dynamic tunnel for 10.10.50.1:500
ike 0:FGT-ADVPN-HUB-1: could not create dialup name FGT-ADVPN-HUB-1_0, too long
ike 0:FGT-ADVPN-HUB-1: deleting
ike 0:FGT-ADVPN-HUB-1: deleted
ike 0:FGT-ADVPN-HUB-1: schedule auto-negotiate
```

4. 检查 Spoke 端的 IPSec 一阶段配置，名称为 FGT - ADVPN-HUB-1，总计 15 个字符。

```
# config vpn ipsec phase1-interface
    edit "FGT-ADVPN-HUB-1"
        set interface "wan1"
        set keylife 28800
        set peertype any
        set net-device enable
        set proposal aes256-sha256
        set add-route disable
        set dpd on-idle
        set dhgrp 14
        set auto-discovery-receiver enable
        set remote-gw x.x.x.x
        set psksecret ENC
        set dpd-retryinterval 5
    next
```

解决方法

1. 当配置 IPSec VPN 时，对名字的长度限制为 15 个字符，超过 15 个字符是无法配置的。
2. 但如果存在 ADVPN 的配置，当 Spoke 与 Spoke 之间触发 Shortcut 连接建立时，Shortcut 连接的名称会在与 Hub 的连接名称的基础上加上字符 `_` + 数字（从 0 开始），例如以上示例中，Shortcut 连接的名称为“FGT - ADVPN - HUB - 1_0”，这已经超过了 15 个字符，这就导致 Shortcut 连接由于名字过长无法发起连接建立。
3. 如果是 ADVPN 部署环境，建议在 Spoke 上配置与 Hub 的 IPSec 连接时，名称长度建议小于 12 个字符（根据最多可以建立多少个 Shortcut 决定，例如需要建立 2 位数数量的 Shortcut 连接，使用 13 个字符的 IPSec 连接名称，从第 10 个 Shortcut 连接开始将无法建立）。

注意事项 2

在同一个 FortiGate 上配置了一个 IPSec 连接的名称后，新增其他 IPSec 连接配置时，名称不要使用已创建的 IPSec 连接名称 + `_<number>` 的格式，例如：

- 第 1 个 IPSec 连接（拨号模式）：名称为 `CT`。
- 第 2 个 IPSec 连接：不要配置名称为 `CT_0`（或 `CT_1`、`CT_2`、`CT_3` ……）。

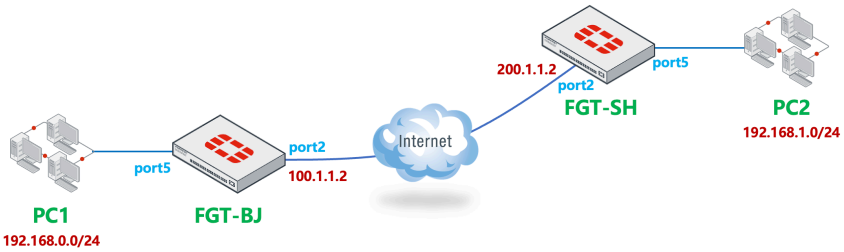
这是由于第 1 个 IPSec 拨号连接有客户端拨号成功后，生成的动态 IPSec 隧道名称会被命名为 `CT_0`、`CT_1`、`CT_2`、`CT_3` …… 这些名称中的 `CT_0` 与第 2 个 IPSec 连接的名称冲突，会导致第 2 个 IPSec 连接无法正常通信，即使两个 IPSec 连接在不同的 VDOM（即使开启了 VDOM，任何接口的名称也具有全局唯一性）。

出现此问题时，受影响的 IPSec 接口的状态会出现异常，使用 `diagnose netlink interface list` 可以看到该接口（CT_1）的 link≠0（接口的 link 只有等于 0 时才正常）。

```bash
Hub (VDOM2) # diagnose netlink interface list

......
if=CT_1 family=00 type=768 index=23 mtu=1420 link=24 master=0    //受影响的IPSec隧道名称，link≠0//
flags=up p2p run noarp multicast

if=CT family=00 type=768 index=24 mtu=1420 link=0 master=0
flags=up p2p run noarp multicast
......
```

IPSec 业务不通问题

网络拓扑



隧道 UP 了，但是业务不通，如何排查

1. 抓包查看数据是否通过 IPSEC 接口转发

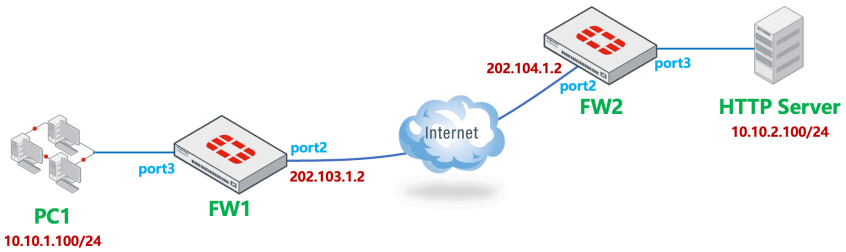
```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
2.615008 port5 in 192.168.0.10 → 192.168.1.10: icmp: echo request
2.615030 VPN-to-SH out 192.168.0.10 → 192.168.1.10: icmp: echo request     #VPN-to-SH是IPSEC接口
2.615891 VPN-to-SH in 192.168.1.10 → 192.168.0.10: icmp: echo reply
2.616067 port5 out 192.168.1.10 → 192.168.0.10: icmp: echo reply
```

2. 通过 Debug Flow 查看匹配的策略和路由是否正确

```
#diagnose debug flow filter addr 192.168.1.10
#diagnose debug flow show function-name enable
#diagnose debug console timestamp enable
#diagnose debug flow trace start 5
#diagnose debug enable

从PC1 192.168.0.10 ping PC2 192.168.1.10 第1个数据包 echo request
2022-12-30 17:50:12 id=20085 trace_id=1 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.0.10:23728→192.168.1.10:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=23728, seq=1."
2022-12-30 17:50:12 id=20085 trace_id=1 func=init_ip_session_common line=6024 msg="allocate a new session-00003fab,
tun_id=0.0.0.0"
2022-12-30 17:50:12 id=20085 trace_id=1 func=vf_ip_route_input_common line=2605 msg="find a route: flag=04000000 gw-
200.1.1.2 via VPN-to-SH"
2022-12-30 17:50:12 id=20085 trace_id=1 func=fw_forward_handler line=881 msg="Allowed by Policy-4:"
2022-12-30 17:50:12 id=20085 trace_id=1 func=ipsecdev_hard_start_xmit line=669 msg="enter IPSec interface VPN-to-SH,
tun_id=0.0.0.0"
2022-12-30 17:50:12 id=20085 trace_id=1 func=_do_ipsecdev_hard_start_xmit line=229 msg="output to IPSec tunnel VPN-to-
SH"
2022-12-30 17:50:12 id=20085 trace_id=1 func=esp_output4 line=844 msg="IPsec encrypt/auth"
2022-12-30 17:50:12 id=20085 trace_id=1 func=ipsec_output_finish line=544 msg="send to 100.1.1.1 via intf-port2"

PC2到PC1的响应包 echo reply
2022-12-30 17:50:12 id=20085 trace_id=2 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:23728→192.168.0.10:0) tun_id=200.1.1.2 from VPN-to-SH. type=0, code=0, id=23728, seq=1."
2022-12-30 17:50:12 id=20085 trace_id=2 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-00003fab,
reply direction"
2022-12-30 17:50:12 id=20085 trace_id=2 func=vf_ip_route_input_common line=2605 msg="find a route: flag=00000000 gw-
192.168.0.10 via port5"
2022-12-30 17:50:12 id=20085 trace_id=2 func=npu_handle_session44 line=1183 msg="Trying to offloading session from VPN-
to-SH to port5, skb.npu_flag=00000000 ses.state=00010200 ses.npu_state=0x05040000"
2022-12-30 17:50:12 id=20085 trace_id=2 func=fw_forward_dirty_handler line=410 msg="state=00010200, state2=00000000,
npu_state=05040000"
```

ESP 分片丢包

网络拓扑



网络配置

1. 基础网络配置（略）。

2. FW1 的 IPSec 配置，静态模式的 IPSec IKEv2 连接，中间没有 NAT 穿越。

> **重要**
>
> 默认配置下 IPSec 的分片方式为 post-encapsulation（后封装），IPSec 隧道在收到任何需要 ESP 封装的报文时，不考虑 IPSec Tunnel 接口的 MTU，先将明文封装到 ESP 中（如果明文已经是分片包，则会先重组明文分片），直接发送至公网接口。从公网接口发出时，再根据公网接口的 MTU（默认为 1500 Bytes）对 ESP 报文进行分片。这种 post-encapsulation（后封装）方式符合 RFC 标准。

```bash
config vpn ipsec phase1-interface
    edit "VPN_to_FW2"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set ip-fragmentation post-encapsulation    //默认配置//
        set dpd on-idle
        set remote-gw 202.103.13.2
        set psksecret ENC
A/oFySlyF2xkGInFX2pcoKpiQAgFb7xyVO4YaJ01IKQzYWPNOClbem+A/JpdLQUUaw+936tj6UC4LjW3Zivxgzl/9N3QZk8WlEBJUO/HrdWrUL=
    next
end

config vpn ipsec phase2-interface
    edit "VPN_to_FW2"
        set phase1name "VPN_to_FW2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

3. FW2 的 IPSec 配置。

```bash
config vpn ipsec phase1-interface
    edit "VPN_to_FW1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set ip-fragmentation post-encapsulation    //默认配置//
        set dpd on-idle
        set remote-gw 202.103.12.2
        set psksecret ENC
mBPuOPm3FhObt60JaaO6ac0T0YsD/8sYru2PKnbY5V1y/3ljX+fA22/WCKbqTAHzhd45dz9yTlZ/qQ/ajGemn+UuMcHnrdeVPEwTrbDi6oP0om=
    next
end

config vpn ipsec phase2-interface
```

```
    edit "VPN_to_FW1"
        set phase1name "VPN_to_FW1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

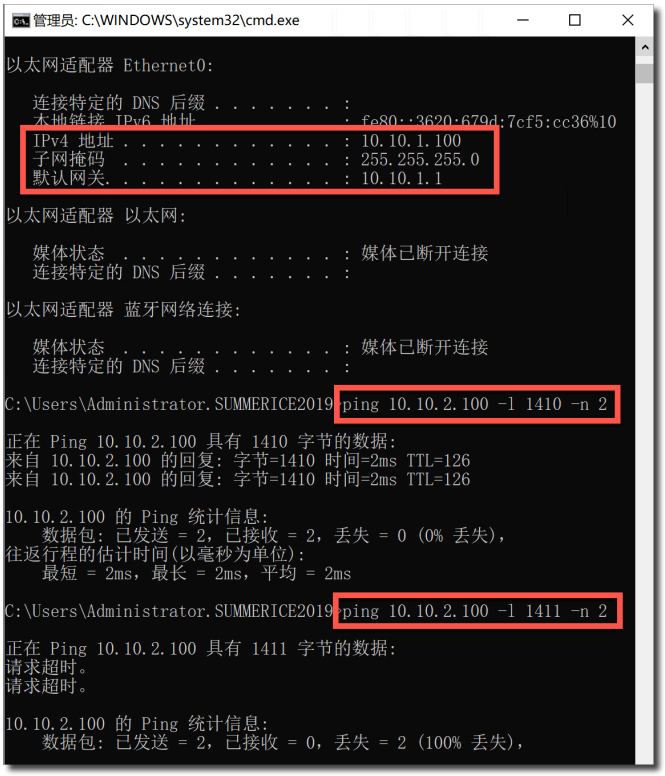
4. 查看 IPSec 二阶段状态，可以看到 options 中有 `frag-rfc` 标记，表示分片方式为 RFC 标准（post-encapsulation，后封装）。

```bash
FW1 # diagnose vpn tunnel list | grep options
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1230 expire=42024/0B replaywin=2048
```

问题现象

1. PC1 与 HTTP Server 可以互相 Ping 通，通过 SMB 传输文件也没有问题。

2. 当 PC1 和 HTTP Server 使用较大的 data-size（当前 IPSec 算法 + 非 NAT-T 环境下，超过 1410 Bytes）互 Ping 时，无法 Ping 通。



3. 较大的 UDP 报文（产生了分片）无法传输。

问题原因

1. 当较大的 ICMP、UDP 等报文无法通过 IPSec 隧道传输时，就要考虑运营商是否有丢弃 ESP 分片报文的动作了。

2. 以 ICMP 包为例，PC1 Ping HTTP Server，data-size=1600，无法 Ping 通。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
Windows IP 配置
以太网适配器 Ethernet0:
   连接特定的 DNS 后缀 . . . . . :
   本地链接 IPv6 地址. . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . : 10.10.1.100
   子网掩码 . . . . . : 255.255.255.0
   默认网关. . . . . : 10.10.1.1

C:\Users\Administrator.SUMMERICE2019>ping 10.10.2.100 -t -l 1600 -n 1
```

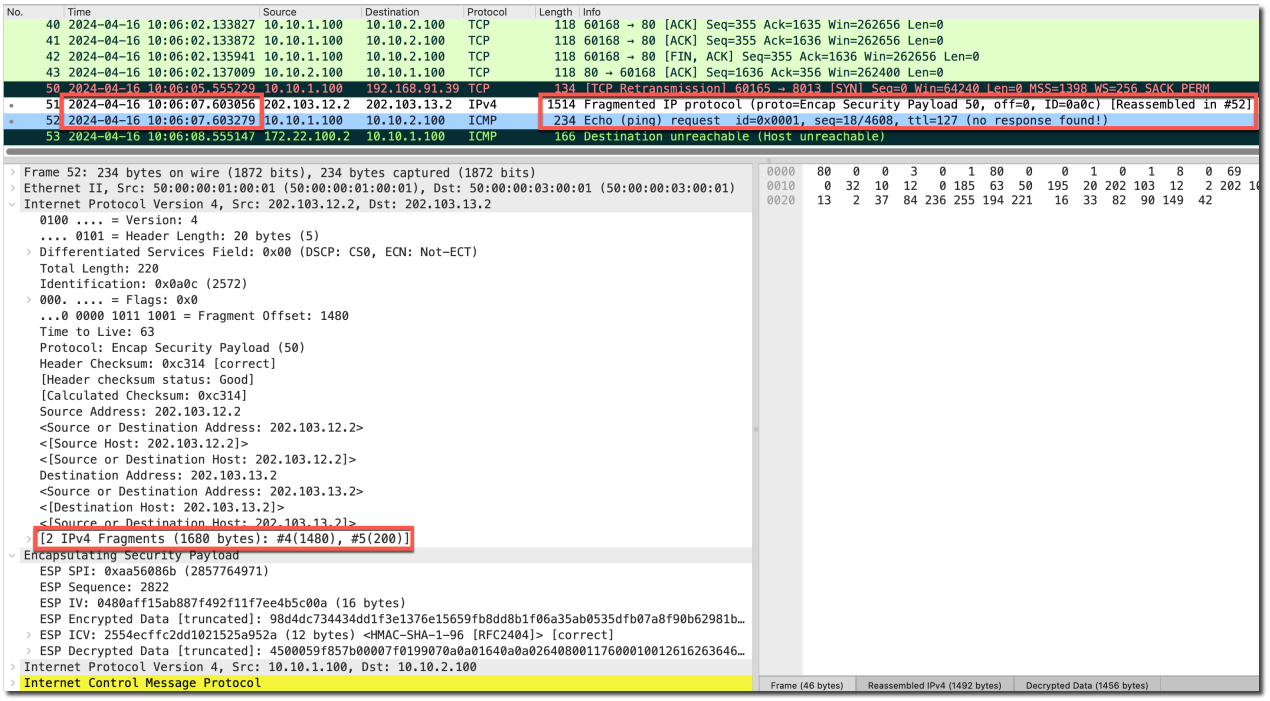
```
正在 Ping 10.10.2.100 具有 1600 字节的数据:
请求超时。

10.10.2.100 的 Ping 统计信息:
    数据包: 已发送 = 1, 已接收 = 0, 丢失 = 1 (100% 丢失)
```

3. 在 FW1 上抓取目标地址为 10.10.2.100 的数据包，PC1 发出的 Ping request 报文被分为两片（第 1 片去掉 IP 头的长度为 1480 Bytes，第 2 片去掉 IP 头的长度为 128 Bytes），进入 IPSec 接口后，被重组为一个包（加上 IP 头部的总长度为 1628 Bytes）。

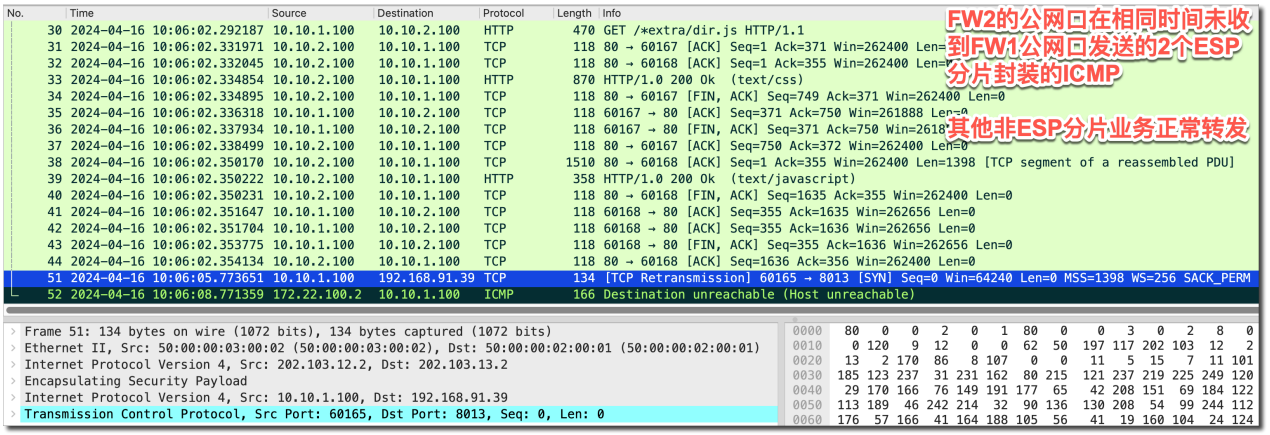
> **重要**
>
> FortiGate 默认的 IP 分片方式为后封装，即先重组分片报文，再封装 ESP，然后从业务接口发出时根据接口 MTU 对 ESP 报文进行分片。

```bash
FW1 # diagnose sniffer packet any 'host 10.10.2.100' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.100]
2024-04-12 14:08:00.664087 port3 in 10.10.1.100 → 10.10.2.100: icmp: echo request (frag 36352:1480@0+)
2024-04-12 14:08:00.664107 port3 in 10.10.1.100 → 10.10.2.100:  ip-proto-1 (frag 36352:128@1480)
2024-04-12 14:08:00.664179 VPN_to_FW2 out 10.10.1.100 → 10.10.2.100: icmp: echo request
```

4. 同时在 FW1 的公网接口（port2）上抓取向 FW2 发送的 ESP 报文（如果是 NAT-T 环境，则需要抓取 UDP 4500 报文）并解密（抓取 ESP 及解密步骤请参考 VPN 技术 → IPSec VPN → IPSec VPN 排错 → IPSec 解密↗章节），可以看到步骤 3 中进入 IPSec 隧道并重组的 ICMP 报文被 ESP 封装，封装 ESP 后的总大小为 1688 Bytes（包含 IP 头部），从 port2 发送时，由于 port2 的 MTU 为 1500 Bytes（默认配置），需要分成 2 片后发出，这样就产生了 ESP 分片。其他的业务没有产生 ESP 分片。



5. 在 FW2 的公网接口（port2）抓取 ESP 报文（如果是 NAT-T 环境，则需要抓取 UDP 4500 报文）并解密（解密步骤请参考 VPN 技术 → IPSec VPN → IPSec VPN 排错 → IPSec 解密↗章节），没有收到步骤 4 中抓到的两片 ESP 报文（封装 ICMP Request 报文），只能收到没有分片的 ESP 报文（如下所示的被 ESP 封装且未分片的 FTP 业务正常转发和使用，但没有抓到步骤 4 中的两片封装 ICMP 的 ESP）。



6. 基本可以确定 FW1 发出的 ESP 分片在 Internet 上被丢弃，与 FortiGate 无关。

解决方法

- 联系运营商检查线路上的 ESP 分片丢包。
- 开启 IPSec 的预封装功能（pre-encapsulation），避免在 Internet 线路上出现 ESP 分片报文，本文着重介绍这种方法。

IPSec 预封装

1. 在 FW1 和 FW2 的 IPSec 一阶段中将 IP 分片模式修改为预封装模式（pre-encapsulation），修改该配置后，隧道会发生重建。

> **重要**
>
> 将 IPSec 的分片方式修改为 pre-encapsulation（预封装）后，IPSec 隧道在收到任何需要 ESP 封装的报文时，会根据当前 IPSec Tunnel 接口的 MTU，先将明文根据 Tunnel 接口的 MTU 进行分片（如果明文已经是分片包，则会先重组明文分片），对每一片明文分片封装 ESP，然后从公网接口发出。
>
> 在 IPSec 绑定的公网接口 MTU 为 1500（默认）的情况下，IPSec Tunnel 接口的 MTU 被设置为 1420 Bytes，确保封装 ESP 后的报文从 IPSec 绑定的公网接口发出时，不会超过 1500 Bytes，从而避免了在公网线路上产生 ESP 分片（如果是 AES256/SHA512 算法 + NAT-T 环境的组合，默认的 1420 是不行的，请参考这里 ↗）。
>
> 如果修改了公网接口的 MTU 值，IPSec Tunnel 接口上的 MTU 值不会自动跟随修改。需要在 Tunnel 接口上手动跟随修改 MTU 值：
>
> - 例如使用 AES128/SHA1 算法组合，物理接口的 MTU 为 1500，IPSec Tunnel 接口的 MTU 为 1420。
> - 客户因为网络需要将物理接口的 MTU 修改为 1400，那么需要同步地将 IPSec Tunnel 接口的 MTU 修改为 1320。
>
> 本文仅介绍物理口没有更改 MTU 的情况。

```bash
FW1:
config vpn ipsec phase1-interface
    edit "VPN_to_FW2"
        set ip-fragmentation pre-encapsulation
    next
end

FW2:
config vpn ipsec phase1-interface
    edit "VPN_to_FW1"
        set ip-fragmentation pre-encapsulation
    next
end
```

> **重要**
>
> 在“拨号用户（dynamic）”类型的 IPSec 一阶段配置中设置 `pre-encapsulation` 时：
>
> - 预封装功能不会立刻生效，必须重启设备或删除重配才能生效（在 FortiOS 7.X 上均存在此问题）。
> - 在新配置的 IPSec 一阶段中直接配置 `set ip-fragmentation pre-encapsulation` 可以直接生效。
>
> “静态 IP 地址（static）”模式的 IPSec 隧道不存在此问题。

2. 查看 IPSec Tunnel 接口的 MTU 值，IPSec Tunnel 接口的 MTU 值为 1420。

```bash
FW1 # diagnose netlink interface list name VPN_to_FW2 | grep mtu
if=VPN_to_FW2 family=00 type=768 index=17 mtu=1420 link=0 master=0

FW2 # diagnose netlink interface list name VPN_to_FW1 | grep mtu
if=VPN_to_FW1 family=00 type=768 index=16 mtu=1420 link=0 master=0
```

3. 再次查看 IPSec 二阶段状态，可以看到 options 中的 `frag-rfc` 标记消失，表示分片方式为预封装（pre-encapsulation）。

```bash
FW1 # diagnose vpn tunnel list | grep options
bound_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/40 options[0028]=npu  run_state=0 role=primary accept_traffic=1
overlay_id=0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1230 expire=42899/0B replaywin=2048
```

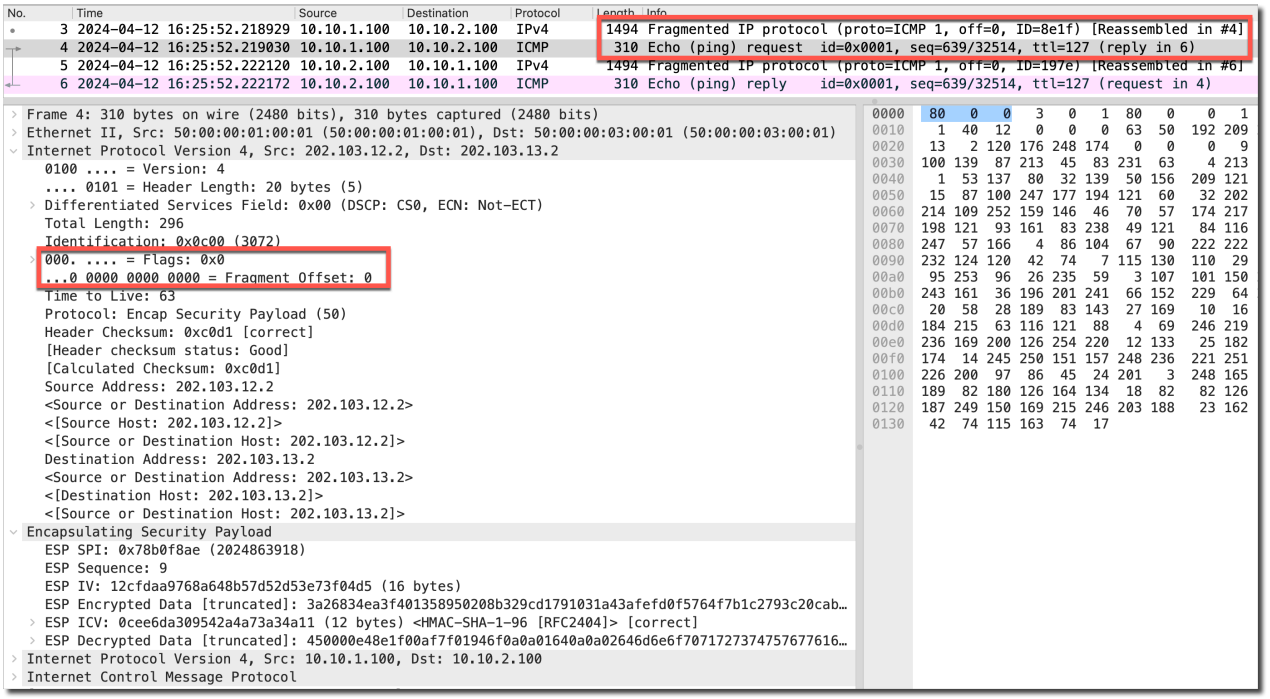
4. 使用 PC1 Ping HTTP Server，data-size=1600，这次可以 Ping 通了。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
Windows IP 配置
以太网适配器 Ethernet0:
   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::3620:679d:7cf5:cc36%10
   IPv4 地址 . . . . . . . . . . . . : 10.10.1.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.10.1.1

C:\Users\Administrator.SUMMERICE2019>ping 10.10.2.100 -t -l 1600 -n 1
正在 Ping 10.10.2.100 具有 1600 字节的数据:
来自 10.10.2.100 的回复: 字节=1600 时间=2ms TTL=126

10.10.2.100 的 Ping 统计信息:
    数据包: 已发送 = 1, 已接收 = 1, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 2ms, 最长 = 2ms, 平均 = 2ms
```

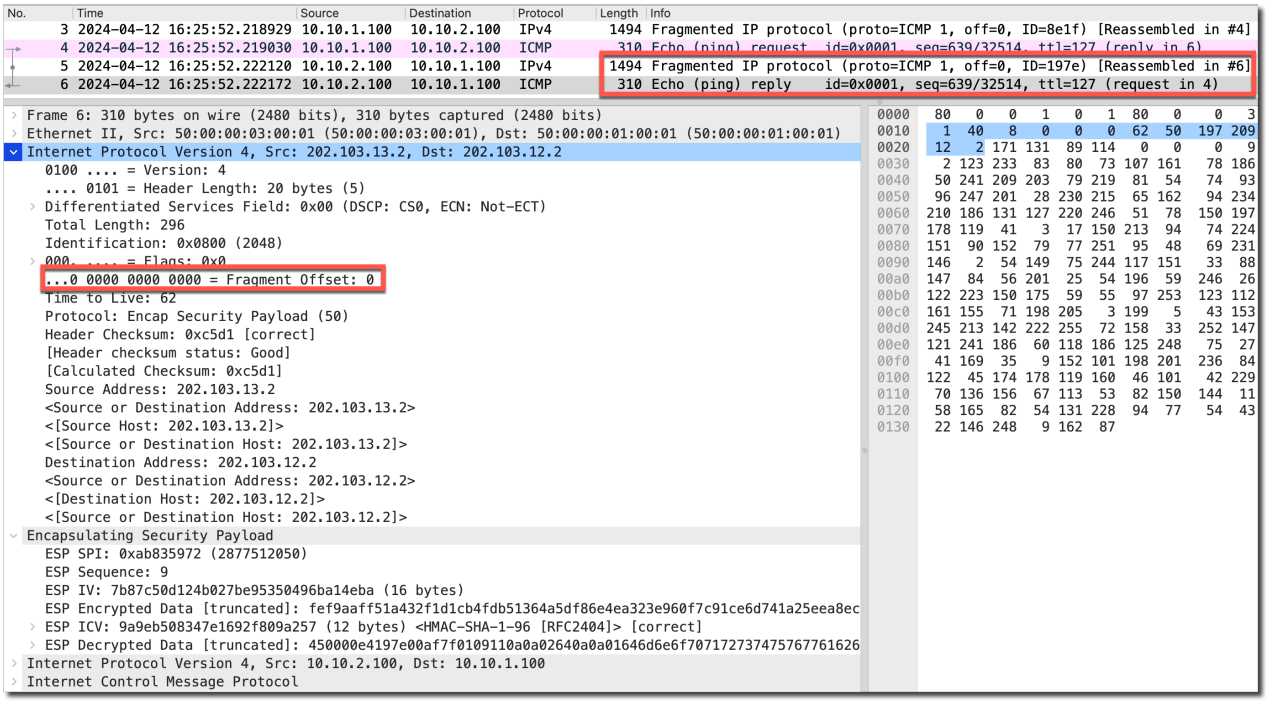
5. 在 FW1 上抓取目标地址为 10.10.2.100 的数据包，PC1 发出的 Ping request 报文被分为两片（第 1 片去掉 IP 头的长度为 1480 Bytes，第 2 片去掉 IP 头的长度为 128 Bytes），进入 IPSec 接口后，被重组为一个包（加上 IP 头部的总长度为 1628 Bytes），随后根据 IPSec Tunnel 接口的 MTU = 1420 Bytes 进行预分片，被分为两片（第 1 片去掉 IP 头的长度为 1400 Bytes，第 2 片去掉 IP 头的长度为 208 Bytes）。

```bash
FW1 # diagnose sniffer packet any 'host 10.10.2.100' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.100]
//Request方向//
2024-04-12 16:12:38.796118 port3 in 10.10.1.100 → 10.10.2.100: icmp: echo request (frag 36382:1480@0+)
2024-04-12 16:12:38.796139 port3 in 10.10.1.100 → 10.10.2.100:  ip-proto-1 (frag 36382:128@1480)
2024-04-12 16:12:38.796191 VPN_to_FW2 out 10.10.1.100 → 10.10.2.100: icmp: echo request (frag 36382:1400@0+)
2024-04-12 16:12:38.796271 VPN_to_FW2 out 10.10.1.100 → 10.10.2.100:  ip-proto-1 (frag 36382:208@1400)

//Reply方向//
2024-04-12 16:12:38.798245 VPN_to_FW2 in 10.10.2.100 → 10.10.1.100: icmp: echo reply (frag 6525:1400@0+)
2024-04-12 16:12:38.798297 VPN_to_FW2 in 10.10.2.100 → 10.10.1.100:  ip-proto-1 (frag 6525:208@1400)
2024-04-12 16:12:38.798317 port3 out 10.10.2.100 → 10.10.1.100: icmp: echo reply (frag 6525:1400@0+)
2024-04-12 16:12:38.798478 port3 out 10.10.2.100 → 10.10.1.100:  ip-proto-1 (frag 6525:208@1400)
```

6. 同时在 FW1 的公网接口（port2）上抓取向 FW2 发送的 ESP 报文（如果是 NAT-T 环境，则需要抓取 UDP 4500 报文）并解密（解密步骤请参考 VPN 技术 → IPSec VPN → IPSec VPN 排错 → IPSec 解密 ↗章节），可以看到步骤 5 中被预分片的 2 个 ICMP 分片报文分别被 ESP 封装，加密后的两个 ESP 包大小均未超过 1500 Bytes（包含 IP 头部），从 port2 发送时，由于 port2 的 MTU 为 1500 Bytes（默认配置），不需要进行分片，这样就不会产生 ESP 分片发送到 Internet。



7. 返回的 ICMP Reply 也一样被 FW2 的 IPSec 进行预封装分片，没有在 Internet 线路上产生 ESP 分片。

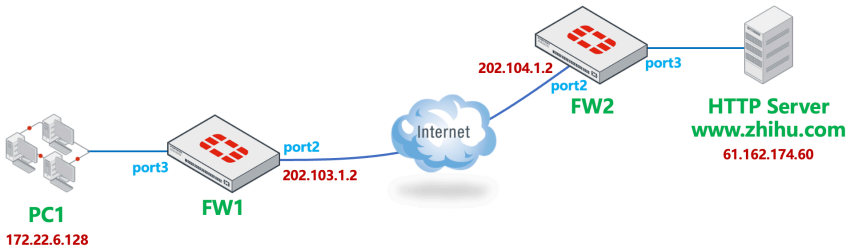


8. 这样就规避了 Internet 线路上丢弃 ESP 分片包导致的业务不通问题。

IPSec 预封装对 DF 置位 IP 包的影响

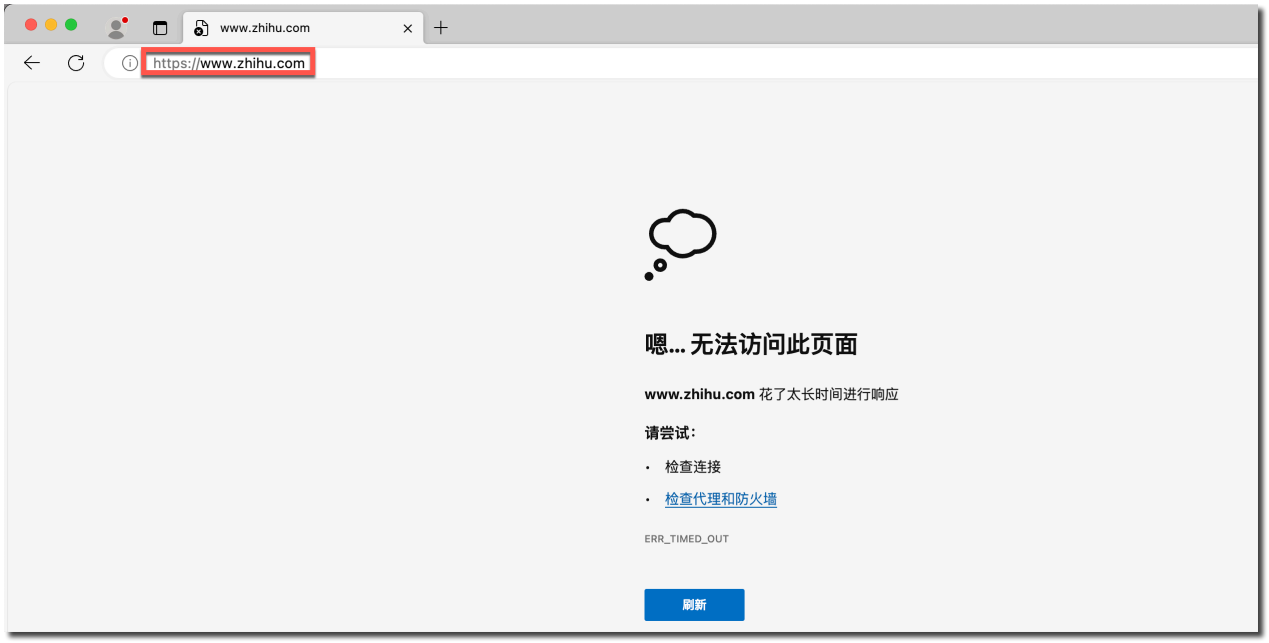
问题现象

1. 修改拓扑如下，PC1 通过 FW1 与 FW2 之间的 IPSec 隧道，从 FW2 的 port3 访问 Internet，FW1 和 FW2 的 IPSec 均开启预封装（pre-encapsulation）功能。

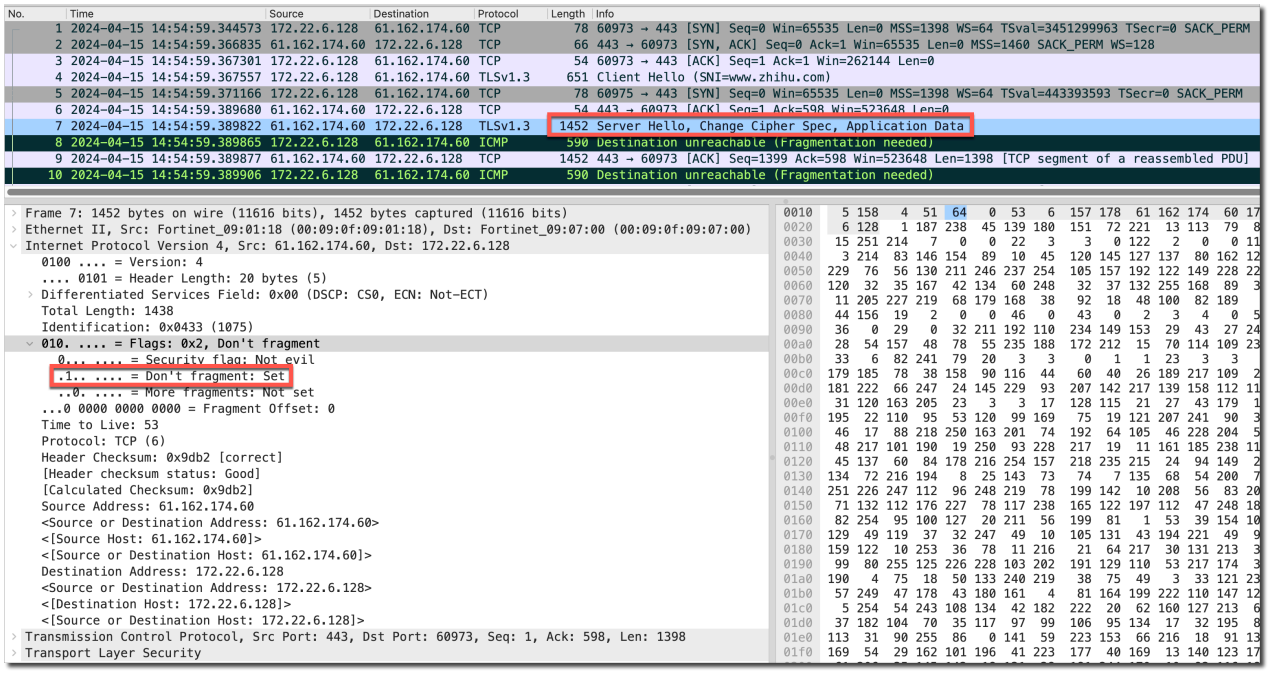


2. PC 在日常访问 HTTP 应用时，TCP SYN 中的 TCP - MSS 一般都为 1460 Bytes。大部分 HTTP 服务器返回的 TCP 业务报文，IP 头部中的 DF（Don't fragment）选项都会置位，表示不允许传输路径上的设备对此报文进行分片。

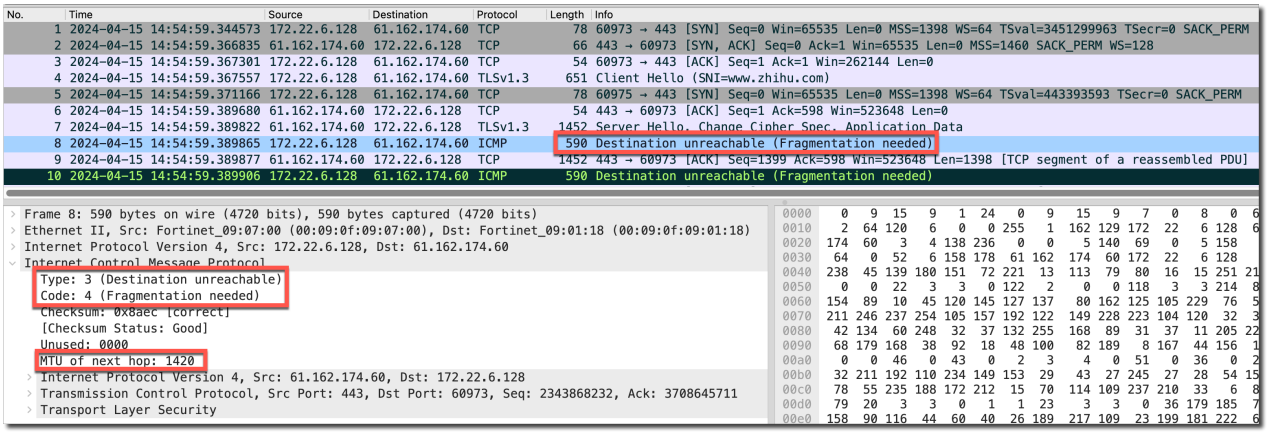
3. 如上述拓扑所示 PC1（10.10.1.100）通过 IPSec 隧道访问 HTTP 服务器（`www.zhihu.com`），无法打开该网站。



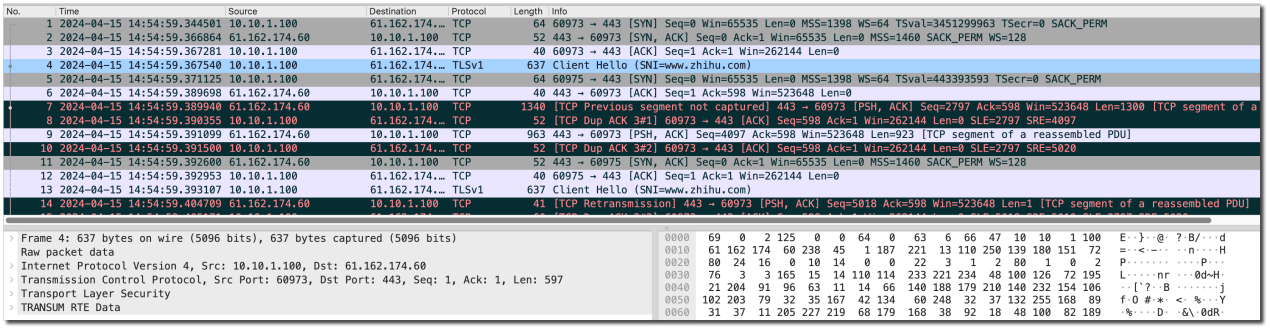
4. 在 FW2 的 port3 接口抓包，HTTP 服务器向客户端返回的 Server Hello 报文较大，去除二层头部帧后的大小为 1438 Bytes，这超过了此例中 IPSec Tunnel 默认的 MTU 1420 Bytes。



5. IPSec 开启了预封装（pre-encapsulation）功能，需要根据此例中 IPSec Tunnel 接口的 MTU 1420 Bytes 进行预分片，但由于该 Server Hello 报文的 IP 头部的 DF（Don't fragment）选项被置位，该报文无法被 IPSec 预分片，所以被 FortiGate 丢弃。FortiGate 也会返回一个 `fragmentation needed and DF set (Type3, Code4)` 的 ICMP 消息给服务器端，携带了 FortiGate IPSec Tunnel 接口的 MTU 值 1420 Bytes。



6. 同时在 FW2 的 IPSec Tunnel 接口上抓包，没有看到这个 Server Hello 报文，说明被 FortiGate 丢弃。



解决方法 1 - 修改 TCP - MSS

1. 在 FW2 或 FW1 的 IPSec Tunnel 接口或 VPN 策略中配置调低 TCP 握手时协商的 TCP - MSS，修改 TCP - MSS 的大小根据 IPSec 接口的 MTU 决定，默认的 IPSec Tunnel 接口 MTU 为 1420 Bytes，那么 TCP - MSS 建议修改为 IPSec Tunnel 接口的 MTU 减去 20 Bytes 的 IP 头部和 20 Bytes 的 TCP 头部，不同二阶段加密/验证算法和 NAT-T 环境下的建议 IPSec Tunnel 接口 MTU 和 TCP - MSS 如下：

> **重要**
>
> 一些特殊情况下 TCP 头部可能达到 60 Bytes，如果修改 TCP - MSS 为 1380 Bytes 后仍然出现由于 TCP 头部超过 20 Bytes 导致的以上问题，请修改 TCP - MSS 为 1340 Bytes。
>
> 以下 IPSec Tunnel 接口的 MTU 均建立在其绑定的业务接口的 MTU 为 1500 Bytes 的前提下，如果调整了业务接口的 MTU（如此例中的 port2），IPSec Tunnel 接口的 MTU 以及 TCP - MSS 都要做相应修改。

算法组合	非 NAT-T		NAT-T	
	建议 Tunnel MTU	建议 TCP - MSS	建议 Tunnel MTU	建议 TCP - MSS
3DES-SHA1	1420（默认，无需修改）	1380	1420（默认，无需修改）	1380
AES256-SHA256/AES256-SHA384/AES192-SHA256/AES192-SHA384/AES128-SHA1/AES128-SHA256	1420（默认，无需修改）	1380	1420（默认，无需修改）	1380
AES256-SHA512	1420（默认，无需修改）	1380	1406（需手工修改）	1366

2. 在 FW2 或 FW1 上修改 IPSec Tunnel 接口或 VPN 策略的 TCP - MSS（仅需要在一台设备上修改，这里以 FW2 为例）为 1380。

```bash
config system interface
    edit "VPN_to_FW1"
        set tcp-mss 1380
    next
end
```

```
修改后可以使用如下CLI命令查看Tunnel接口的TCP-MSS：
FW2 # diagnose netlink dstmac list VPN_to_FW1
dev=VPN_to_FW1 mac=00:00:00:00:00:00 rx_tcp_mss=1380 tx_tcp_mss=1380 egress_overspill_threshold=0 egress_bytes=0
egress_over_bps=0 ingress_overspill_threshold=0 ingress_bytes=0 ingress_over_bps=0 sampler_rate=0
```

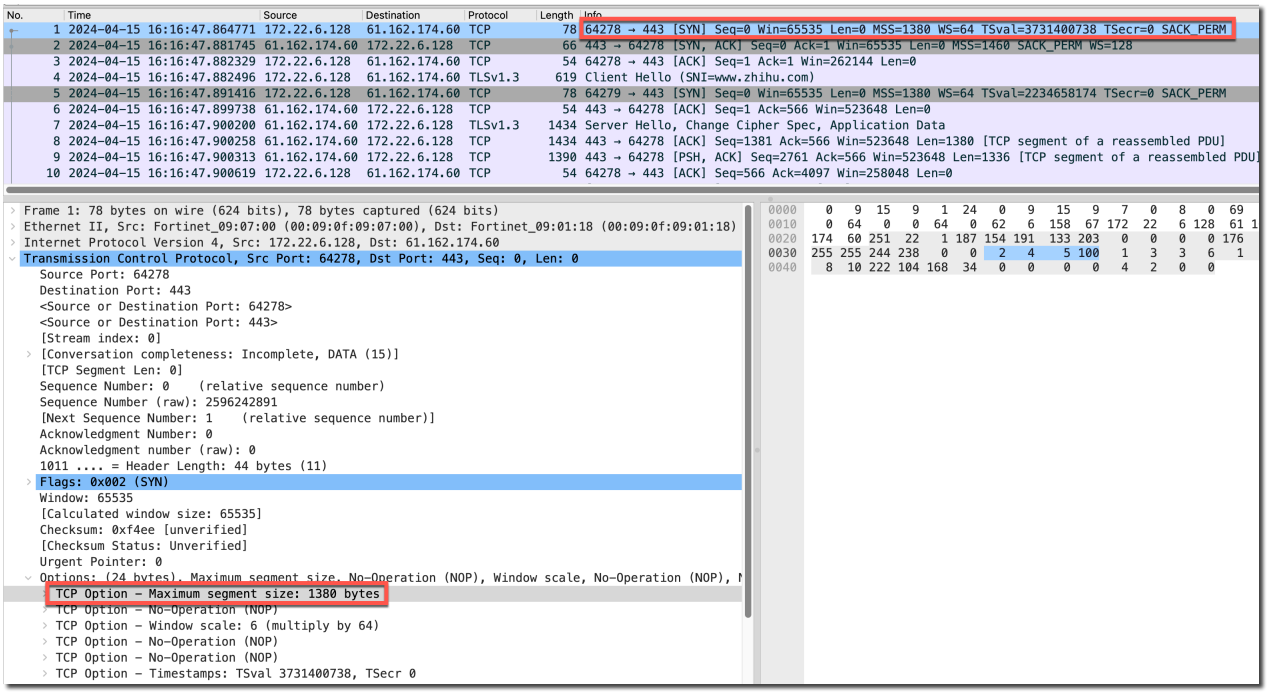
或基于策略的修改（两种方法取其一）：

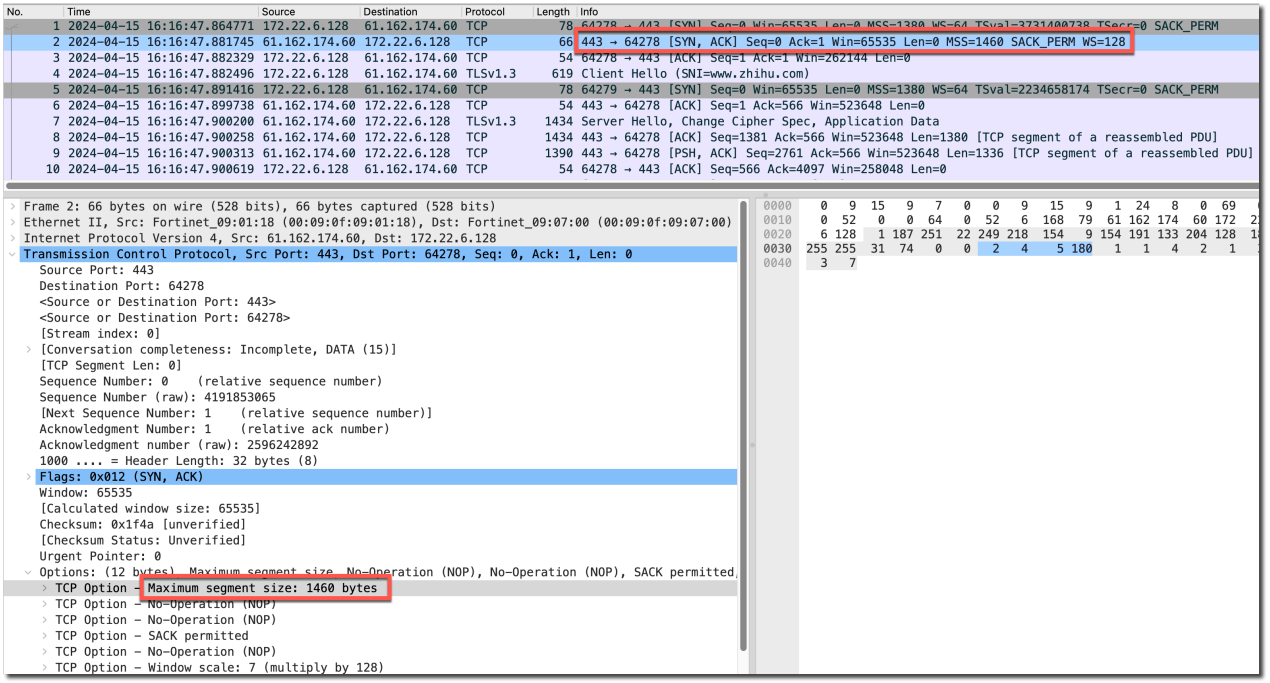
```bash
config firewall policy
    edit 2
        set name "VPN_to_FW1"
        set srcintf "port3" "VPN_to_FW1"
        set dstintf "VPN_to_FW1" "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set tcp-mss-sender 1380      //发送方向的TCP SYN报文修改TCP-MSS为1380//
        set tcp-mss-receiver 1380    //接收方向的TCP SYN报文修改TCP-MSS为1380//
    next
end
```

3. 再次使用 PC1（10.10.1.100）通过 IPSec 隧道访问 HTTP 服务器（`www.zhihu.com`），可以正常打开该网站。

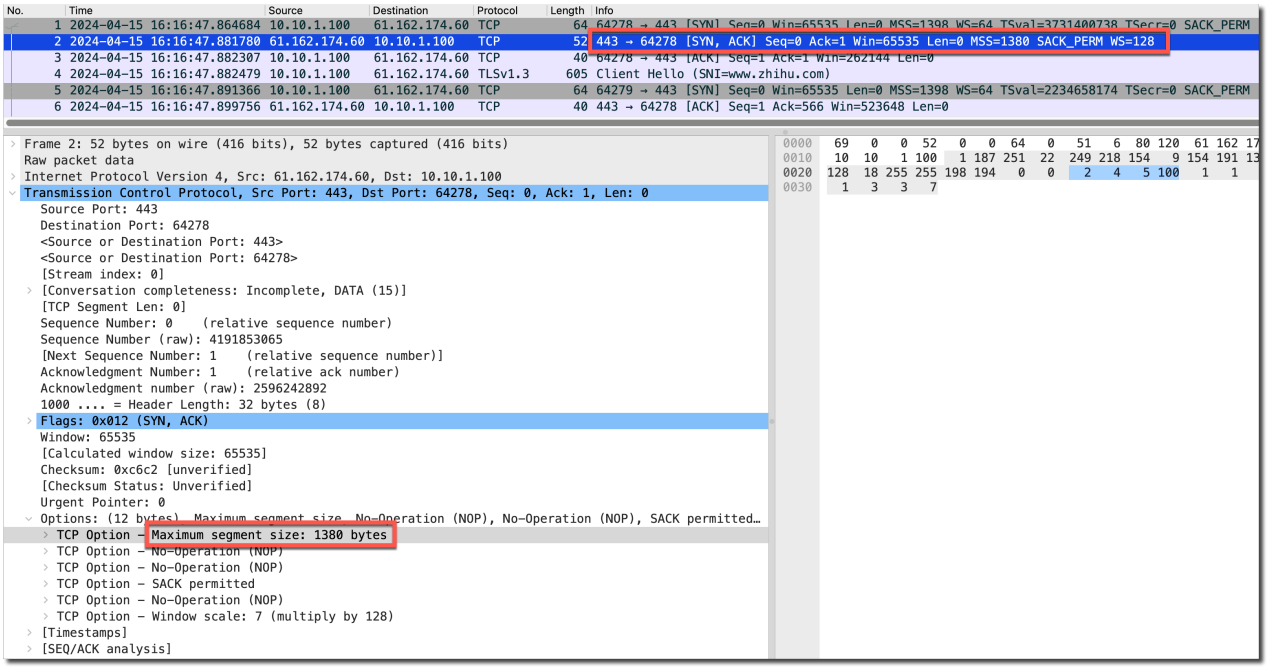


4. 在 FW2 的 port3 接口抓包，可以看到 PC1 访问 HTTP 服务器发出的 TCP SYN 报文中的 TCP - MSS 被 FortiGate 修改为了 1380 Bytes。HTTP 服务器返回的 TCP - MSS 为 1460 Bytes。

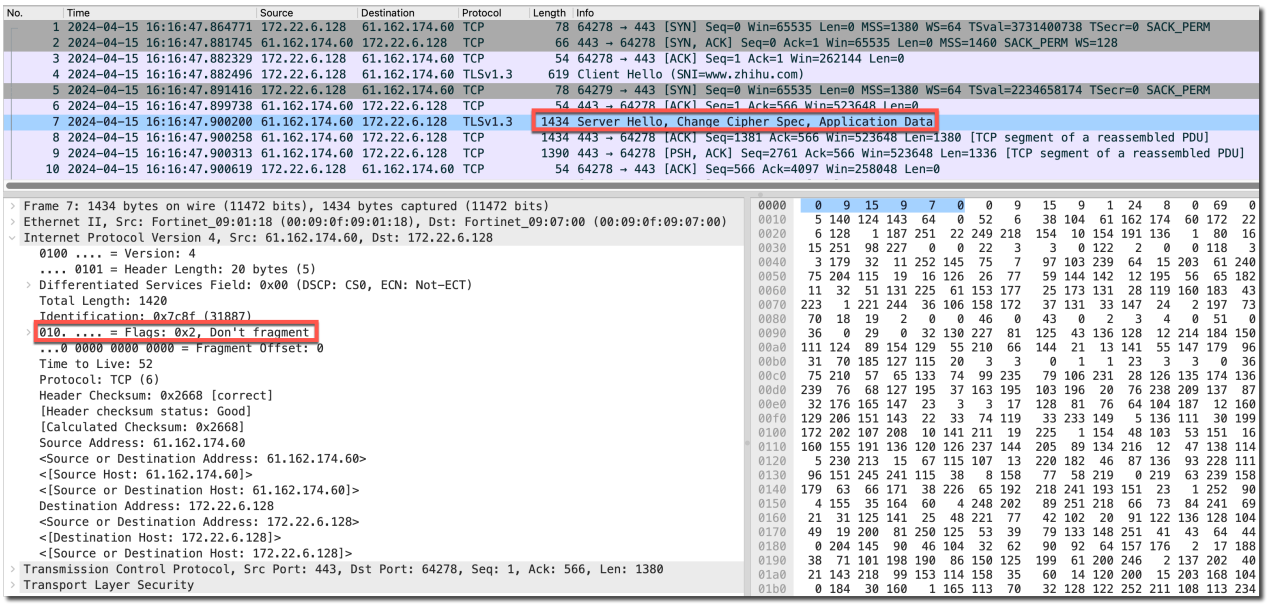




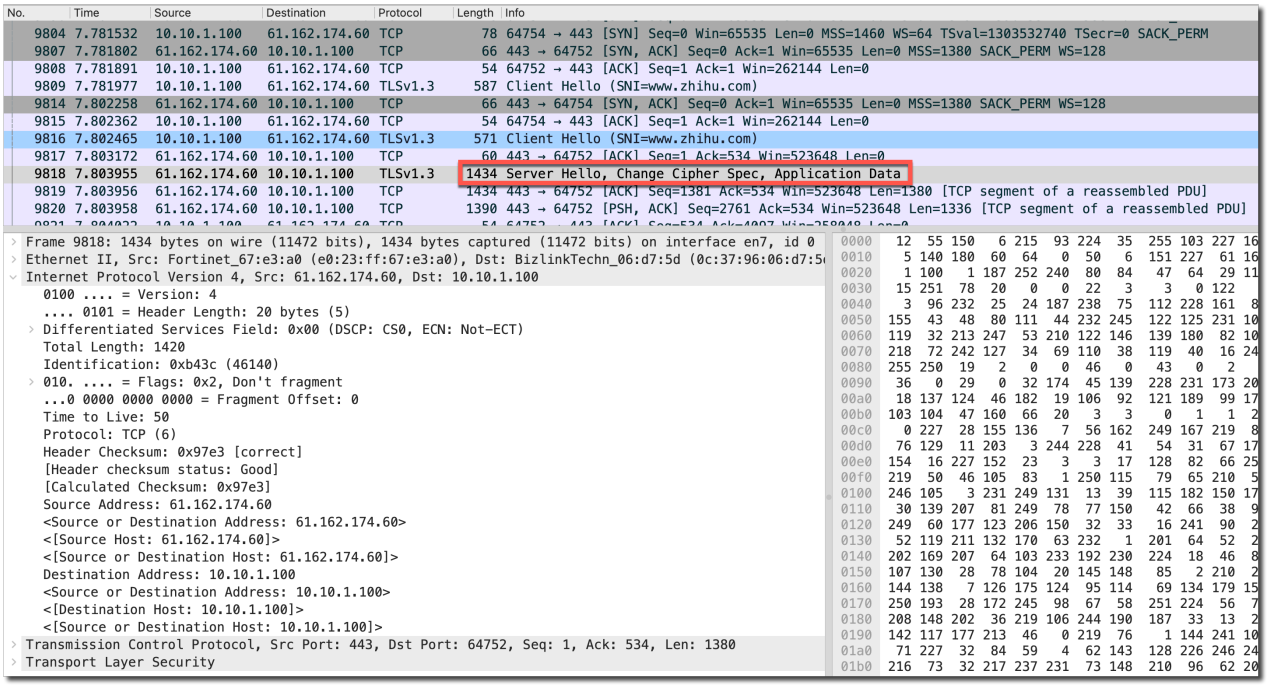
5. 在 FW2 的 IPSec Tunnel 接口抓包，可以看到 HTTP 服务器返回的 SYN ACK 的 TCP - MSS 被修改为 1380 Bytes。



6. 在 FW2 的 port3 接口抓包，HTTP 服务器向客户端返回的 Server Hello 这次去除二层头部帧后的大小为 1420 Bytes，没有超过此例中 IPSec Tunnel 默认的 MTU 1420 Bytes，所以即使 IP 头中仍然有 DF 置位，但不会被 IPSec 预分片。



7. 同时在 PC1 的网卡上抓包，可以收到这个 Server Hello 报文，说明没有被 FortiGate 丢弃，问题解决。



8. Internet 传输路径上未产生任何 ESP 分片。

解决方法 2 - 关闭 honor-df

1. 在 FW1 和 FW2 上的全局配置下关闭 `honor-df`（默认开启），关闭后，将忽略任何 IP 头部中的 DF 位，对超过接口 MTU 的报文执行分片操作。

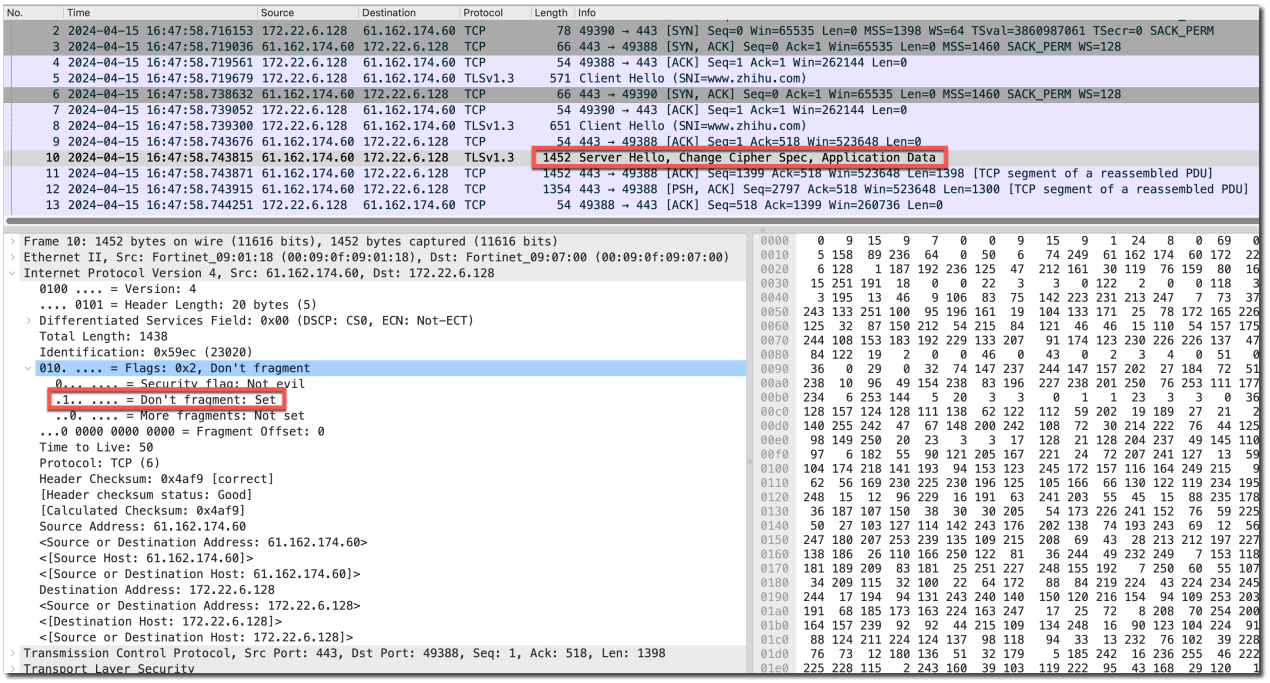
> **重要**
>
> 该配置不仅影响预封装模式的 IPSec，也影响其他非 VPN 流量，请谨慎使用。

```bash
config system global
    set honor-df disable
end
```

2. 再次使用 PC1（10.10.1.100）通过 IPSec 隧道访问 HTTP 服务器（`www.zhihu.com`），可以正常打开该网站。

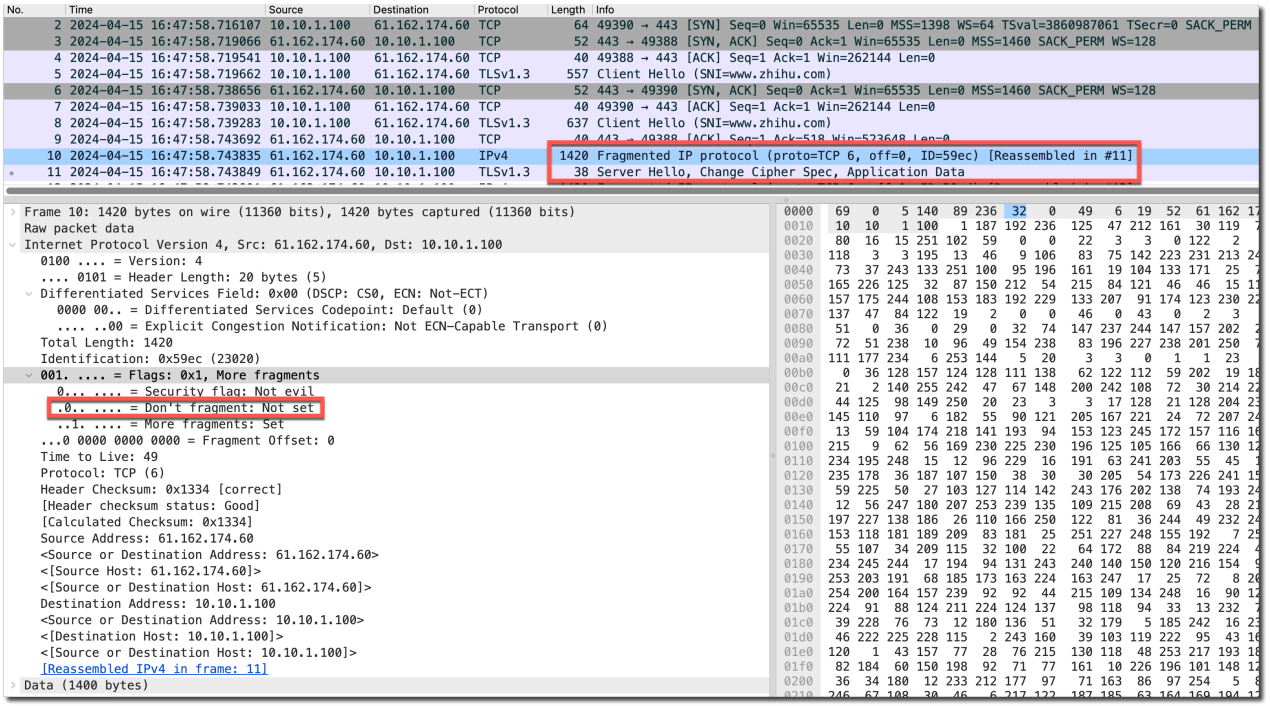


3. 在 FW2 的 port3 接口抓包，HTTP 服务器向客户端返回的 Server Hello 报文较大，去除二层头部帧后的大小为 1438 Bytes，这超过了此例中 IPSec Tunnel 默认的 MTU 1420 Bytes，且 IP 头部中的 DF 置位。

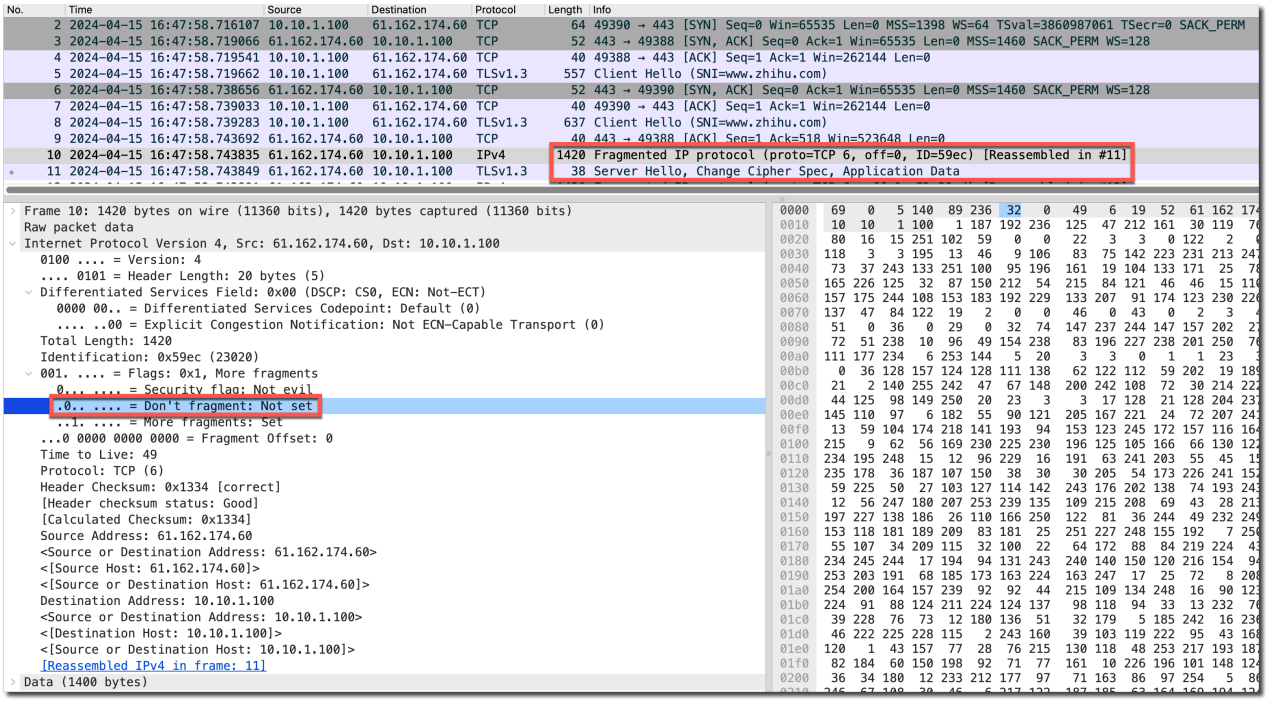


4. 同时在 FW2 的 IPSec Tunnel 接口抓包，可以看到 FortiGate 无视了这个 Server Hello 报文 IP 头部的 DF 标志位，执行了 IPSec 预封装功能，根据 IPSec Tunnel 接口的 MTU 1420 Bytes 将 Server Hello 报文分为 2 片，这两个分片 IP 包的 DF 选项均没有置位。

> **重要**
>
> 在 Tunnel 接口上抓取的报文没有 2 层帧头部，所以两个分片的大小为 1420 Bytes 和 38 Bytes。



5. 同时在 PC1 的网卡上抓包，可以收到这个 Server Hello 报文的两个分片，问题解决。



6. Internet 传输路径上未产生任何 ESP 分片。

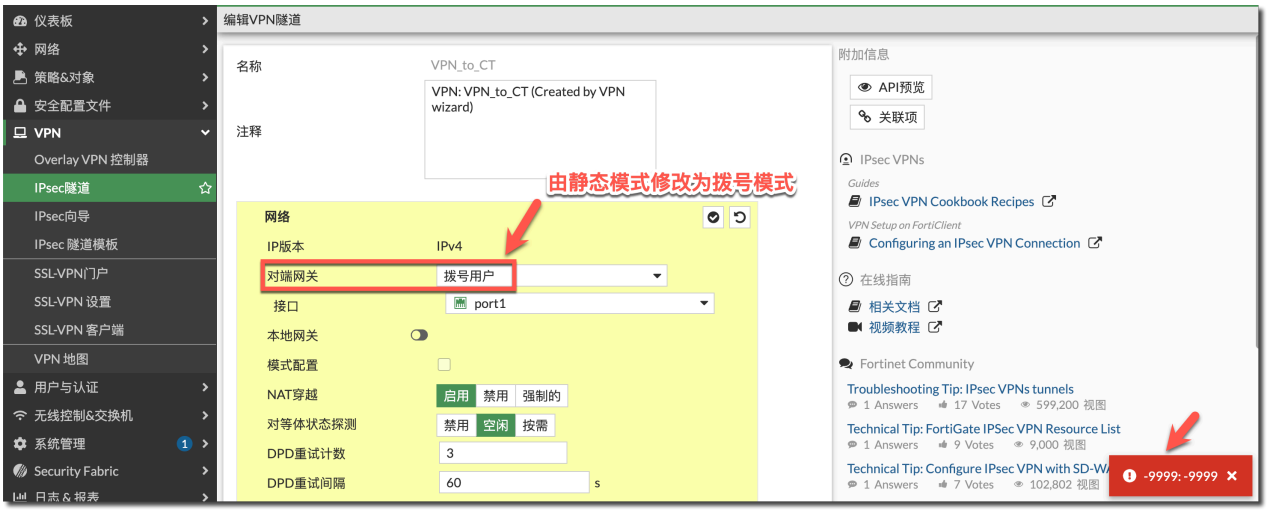
总结

1. 当遇到 Internet 线路上丢 ESP 分片的情况，可以使用 FortiGate IPSec 的预封装（pre-encapsulation）功能规避此问题：
   - IPSec 后封装（post-encapsulation）：默认配置。IPSec 隧道在收到任何需要 ESP 封装的报文时，不考虑 IPSec Tunnel 接口的 MTU，先将明文封装到 ESP 中（如果明文已经是分片包，则会先重组明文分片），直接发送至公网接口。从公网接口发出时，再根据公网接口的 MTU（默认为 1500 Bytes）对 ESP 报文进行分片。这种 post-encapsulation（后封装）方式符合 RFC 标准。
   - IPSec 预封装（pre-encapsulation）：IPSec 隧道在收到任何需要 ESP 封装的报文时，会根据当前 IPSec Tunnel 接口的 MTU，先将明文根据 Tunnel 接口的 MTU 进行分片（如果明文已经是分片包，则会先重组明文分片），对每一片明文分片封装 ESP，然后从公网接口发出。
2. 当使用 IPSec 预封装（pre-encapsulation）模式时，DF 置位且超过 IPSec Tunnel MTU 的 IP 报文在 IPSec 隧道中传输会被丢弃，可以通过以下 2 种方法解决：
   - 解决方法 1 - 修改 TCP - MSS（推荐）。根据当前 IPSec 接口绑定的业务接口的 MTU 值决定 IPSec Tunnel 接口的 MTU，根据 IPSec Tunnel 接口的 MTU 决定 TCP - MSS 的大小，适用于 TCP 报文。
   - 解决方法 2 - 关闭 honor-df。忽略任何 IP 头部中的 DF 置位，影响全局。对于 DF 置位的 UDP 或其他非 TCP 报文可以使用此种方法。
3. 更多关于 IPSec 分片的知识请参考：https://summerice.notion.site/376bdd5796914707b0560a8a0e429bde?v ↗ = 033f688f25f34b5cacc6df3f21a7560b & pvs = 4 。

IPSec 无法修改隧道类型

问题现象

1. 当需要修改已创建的 IPSec 隧道类型时，例如已创建的 IPSec 隧道类型为静态模式，需要修改为拨号模式，页面会提示 `-9999:-9999` 的错误信息。



2. 通过 CLI 更改时也会失败，提示 `Cannot change tunnel type once configured.`。

```bash
Spoke1 # config vpn ipsec phase1-interface
Spoke1 (phase1-interface) # edit VPN_to_CT
Spoke1 (VPN_to_CT) # set type dynamic
Spoke1 (VPN_to_CT) # next
Cannot change tunnel type once configured.
object set operator error, -9999, roll back the setting
Command fail. Return code 1
```

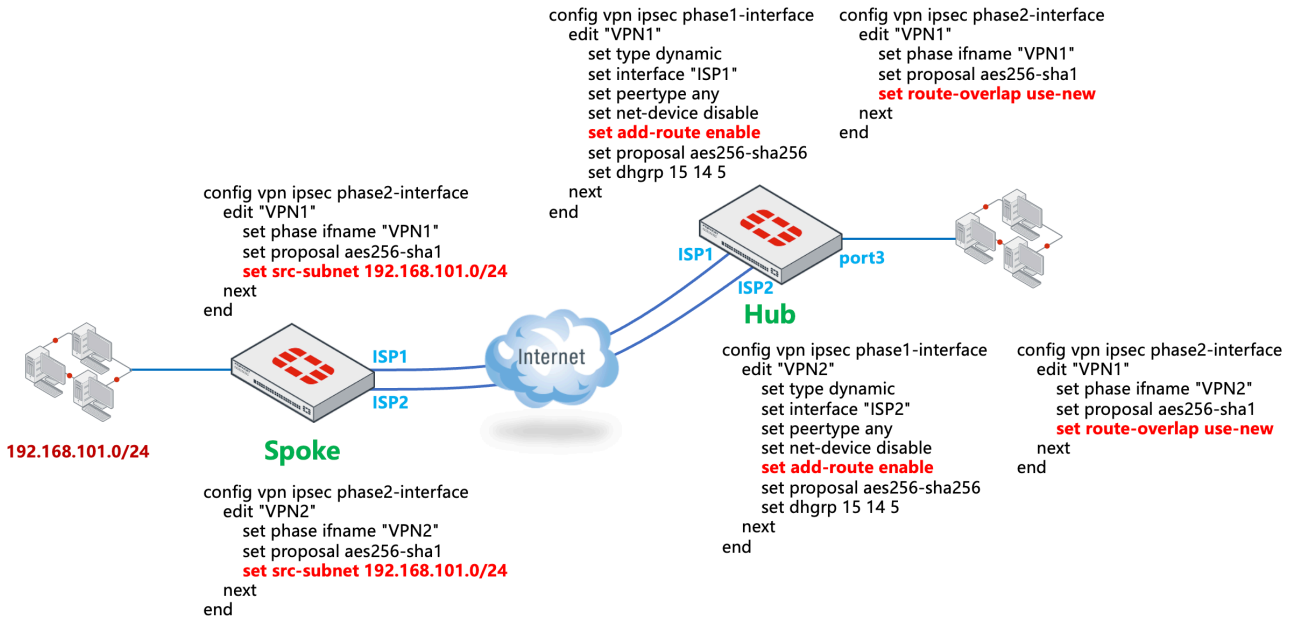
问题原因

配置 IPSec 隧道后会生成“tun_id”，用于 IPsec 隧道的路由会使用“tun_id”进行匹配。因此无法修改 IPSec 隧道的类型，在静态、拨号、DDNS 类型之间都无法进行修改。

解决方法为删除已有的 IPSec 隧道（以及相关联配置），并重新创建其他类型的 IPSec 隧道。

Hub-Spoke 双线路隧道抖动

网络拓扑



1. Spoke 与 Hub 均有 2 条线路接入 Internet。
2. Hub 的 ISP1 和 ISP2 线路上分别创建一个拨号类型的 IPSec 连接：
   - Hub 上的两条拨号 IPSec 隧道一阶段配置中，开启 `add-route` 功能，根据 Spoke 的源保护网段自动添加去往 Spoke 的路由。
   - 二阶段源目保护网段均为 0.0.0.0/0，默认配置下，`route-overlap` 状态为 `use-new` 。
3. Spoke 的 ISP1 线路与 Hub1 的 ISP1 线路建立 IPSec 连接，Spoke 的 ISP2 线路与 Hub1 的 ISP1 线路建立 IPSec 连接。
   - Spoke 上的两条 IPSec 隧道二阶段配置中，源保护网段均配置为明细的 192.168.101.0/24，目的保护网段配置为 0.0.0.0/0。
4. 由于 Spoke 两条线路的上的 IPSec 二阶段配置的源保护网段一样（192.168.101.0/24），所以理论上 Hub 在实现 `add-route` 时，应该添加 2 条目标为 192.168.101.0/24 的路由，并等价负载到两条拨号隧道上创建的子隧道。这也是用户想要实现的需求。

问题现象

> **重要**
>
> 在升级 FortiOS 7.0.13、7.2.6、7.4.1 及更新版本后，会出现此问题，这是正常情况，功能机制随版本发生了变化。

1. Spoke 上的 SD-WAN 健康检查检测 Hub 端的资源，出现丢包现象。

```bash
Spoke # diagnose sys sdwan health-check
Health Check(Spoke1):
Seq(4 VPN1): state(dead), packet-loss(95.000%) sla_map=0x0
Seq(5 VPN2): state(alive), packet-loss(11.000%) latency(9.479), jitter(4.353), bandwidth-up(19999998), bandwidth-dw(19999998), bandwidth-bi(39999996) sla_map=0x0
```

2. 在 Hub 上查看 IPSec 一阶段（IKE）状态，多次执行查看命令，已创建时间一直处于很短的状态。

```bash
Hub # diagnose vpn ike gateway list | grep "name:\|created"
name: VPN2_0
//created: 1s ago//
IKE SA: created 1/1 established 1/1 time 10/10/10 ms
IPsec SA: created 2/2 established 2/2 time 0/0/0 ms
name: VPN1_0
//created: 1s ago//
IKE SA: created 1/1 established 1/1 time 10/10/10 ms
IPsec SA: created 1/1 established 1/1 time 10/10/10 ms
```

3. 查看 Hub 上通过 `add-route` 功能添加的路由，只有一条隧道上添加了去往 Spoke 内网的路由。

```bash
Hub # get router info routing-table details 192.168.101.0

Routing table for VRF=0
Routing entry for 192.168.101.0/24
Known via "static", distance 15, metric 0, best
* via VPN1 tunnel 200.52.10.17, tun_id
```

问题原因

1. 通过 `diagnose debug application ike -1` 查看 IPSec Debug 信息，可以看到 IPSec 不停地在两条隧道上添加和删除去往 Spoke 的路由。

```bash
Hub # diagnose debug application ike -1
Hub # diagnose debug enable

ike 0:VPN1_0:562:1753: peer proposal is: peer:0:192.168.101.0-192.168.101.255:0, me:0:0.0.0.0-255.255.255.255:0
ike 0:VPN1_0:562:Spoke1LAN:1753: dst 0 7 0:192.168.101.0-192.168.101.255:0
ike 0:VPN2:1748: moving route 192.168.101.0/255.255.255.0 oif VPN2(32) metric 15 priority 1 to 0:VPN1:1753
ike 0:VPN2:1748: del route 192.168.101.0/255.255.255.0 tunnel 20.0.0.3 oif VPN2(32) metric 15 priority 1
ike 0:VPN1:1753: add route 192.168.101.0/255.255.255.0 gw 200.52.10.17 oif VPN1(31) metric 15 priority 1
ike 0:VPN2:563:1755: TSi_0 0:192.168.101.0-192.168.101.255:0
ike 0:VPN2:563:Spoke1-LAN_E:1755: TSi_0 0:192.168.101.0-192.168.101.255:0
ike 0:VPN2_0:563:Spoke1-LAN_E:1755: dst 0 7 0:192.168.101.0-192.168.101.255:0
ike 0:VPN1:1753: moving route 192.168.101.0/255.255.255.0 oif VPN1(31) metric 15 priority 1 to 0:VPN2:1755
ike 0:VPN1:1753: del route 192.168.101.0/255.255.255.0 tunnel 200.52.10.17 oif VPN1(31) metric 15 priority 1
ike 0:VPN2:1755: add route 192.168.101.0/255.255.255.0 gw 20.0.0.3 oif VPN2(32) metric 15 priority 1
ike 0:VPN1_0:564:1761: peer proposal is: peer:0:192.168.101.0-192.168.101.255:0, me:0:0.0.0.0-255.255.255.255:0
ike 0:VPN1_0:564:Spoke1LAN:1761: dst 0 7 0:192.168.101.0-192.168.101.255:0
ike 0:VPN2:1755: moving route 192.168.101.0/255.255.255.0 oif VPN2(32) metric 15 priority 1 to 0:VPN1:1761
ike 0:VPN2:1755: del route 192.168.101.0/255.255.255.0 tunnel 20.0.0.3 oif VPN2(32) metric 15 priority 1
ike 0:VPN1:1761: add route 192.168.101.0/255.255.255.0 gw 200.52.10.17 oif VPN1(31) metric 15 priority 1
```

2. 这是由于 Hub 端在执行 `add-route` 功能时，由于两条 Spoke 隧道使用的源保护网段一样，Hub 认为这两个保护网段冲突：

   - 当 Hub 的 IPSec 二阶段中的 `route-overlap` 状态为 `use-new`（默认配置）时：每个 Spoke 网段只能添加一条路由到路由表，后建立的 IPSec 隧道会导致前边已建立的 IPSec 隧道及其添加的相同路由被移除。这样就会出现两条隧道不停翻转（overlap）的情况。
   - 当 Hub 的 IPSec 二阶段中的 `route-overlap` 状态为 `allow` 时：允许同时在两个隧道上添加去往 Spoke 内网的路由。

解决方法

1. 在 Hub 的两条 IPSec 拨号连接的二阶段配置中，将 `route-overlap` 状态配置为 `allow`。

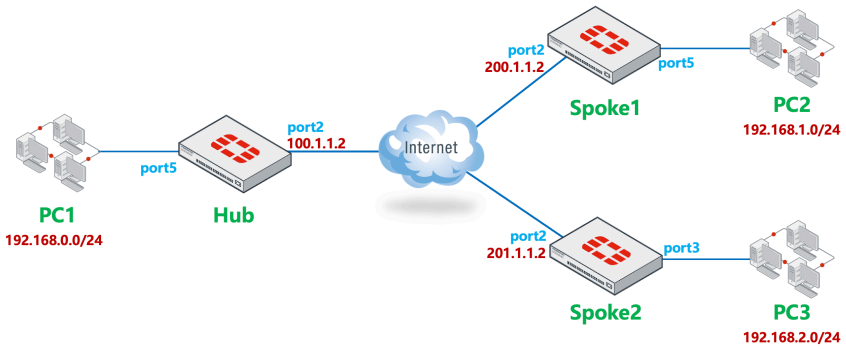
```bash
config vpn ipsec phase2-interface
    edit "VPN1"
        set route-overlap allow
    next
    edit "VPN2"
        set route-overlap allow
    next
end
```

2. 查看 Hub 上通过 `add-route` 功能添加的路由，两条隧道上都添加了去往 Spoke 内网的路由，并等价负载。

```bash
Hub # get router info routing-table details 192.168.101.0

Routing table for VRF=0
Routing entry for 192.168.101.0/24
Known via "static", distance 15, metric 0, best
* via VPN2 tunnel 115.38.6.12, tun_id
* via VPN1 tunnel 200.52.10.17, tun_id
```

Hub-Spoke 静态路由问题

网络拓扑



1. Hub 设备位于数据中心，配在 port2 配置“拨号用户（Dynamic）”模式的 IPSec 连接。
2. Spoke1 与 Spoke2 均在 port2 上配置“静态 IP 地址（Static）”类型的 IPSec 连接，与 Hub 建立 IPSec 隧道。
3. Hub 与 Spoke 之间通过静态路由进行通信。

配置信息

仅以 Hub 和 Spoke1 为例。基础网络 IP、路由配置略。

Hub

1. IPSec 隧道配置，隧道类型为 dynamic，第二阶段保护网段为 0.0.0.0/0↔0.0.0.0/0，`add-route` 功能为关闭状态。

```bash
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set add-route disable
        set dpd on-idle
        set psksecret ENC
YsyLk34z2H6ffoT8QYhgsX5aS5FN4Vq3g9IpjKgh/7uBJdl+W3IMHDuXxXpT1a+ZyxR37cOfpB1Uzf0Vnv8D2Q0E+3259DPYr556hZRDGYo46f=
        set dpd-retryinterval 60
    next
end

config vpn ipsec phase2-interface
    edit "Hub"
        set phase1name "Hub"
        set proposal aes128-sha1
        set keepalive enable
    next
end
```

2. 配置 IPSec Tunnel IP。

```bash
config system interface
    edit "Hub"
        set ip 10.10.10.1 255.255.255.255
        set type tunnel
        set remote-ip 10.10.10.254 255.255.255.0
        set interface "port2"
    next
end
```

3. 配置防火墙策略放通 IPSec 隧道与内网口 port3 之间的流量。

```bash
config firewall policy
    edit 1
        set name "VPN_Hub"
        set srcintf "Hub" "port3"
        set dstintf "Hub" "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. 配置去往 Spoke1 内网网段（192.168.1.0/24）和 Spoke2 内网网段（192.168.2.0/24）的静态路由，出接口选择 IPSec Tunnel 接口。

```bash
config router static
    edit 2
        set dst 192.168.1.0 255.255.255.0
        set device "Hub"
    next
    edit 3
        set dst 192.168.2.0 255.255.255.0
        set device "Hub"
    next
end
```

Spoke1

1. IPSec 隧道配置，隧道类型为 static，第二阶段保护网段为 0.0.0.0/0↔0.0.0.0/0。

```bash
config vpn ipsec phase1-interface
    edit "Spoke1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 100.1.1.2
        set dpd on-idle
        set psksecret ENC
fDnGerNZb+ogGLimNoAdp/nQj8tOZRBwu91bMs1MqK4oB/t1Szc8C1Ma43Pga8cW8Qxdew7TgFECbvwq1nYxPqVU6pnbj/DkpT/nuGzvTietA0mjCBubl/1j
PU8wkFWABfIXsivn0YOSyjHmQx5LSdbhjvM9FTI6CKjixMszGuC8T8kpczpL2h1FAX1SkytiKeztJg==
    next
end

config vpn ipsec phase2-interface
    edit "Spoke1"
        set phase1name "Spoke1"
        set proposal aes128-sha1
        set auto-negotiate enable
    next
end
```

2. 配置 IPSec Tunnel IP。

```bash
config system interface
    edit "Spoke1"
        set ip 10.10.10.2 255.255.255.255
        set type tunnel
        set remote-ip 10.10.10.254 255.255.255.0
        set interface "port2"
    next
end
```

3. 配置防火墙策略放通 IPSec 隧道与内网口 port3 之间的流量。

```bash
config firewall policy
    edit 1
        set name "VPN_Spoke1"
        set srcintf "port3" "Spoke1"
        set dstintf "port3" "Spoke1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. 配置去往 Hub 内网网段（192.168.0.0/24）的静态路由，出接口选择 IPSec Tunnel 接口。

```bash
config router static
    edit 2
        set dst 192.168.0.0 255.255.255.0
        set device "Spoke1"
    next
end
```

Spoke2

略，与 Spoke1 一致。

问题现象

1. IPSec 隧道建立后，查看 Hub 和 Spoke 的 IPSec 一阶段状态中的 `tun_id`，可以看到 Hub 建立了两个动态子 IPSec 连接（Hub_0 和 Hub_1），Hub 和 Spoke 显示已建立的隧道 ID（`tun_id`）均为对端公网 IP 地址。

```bash
Hub # diagnose vpn ike gateway list | grep "name\|tun_id"
name: Hub_0
tun_id: 200.1.1.2/::10.0.0.5    <——隧道ID为Spoke1的公网IP
name: Hub_1
tun_id: 201.1.1.2/::10.0.0.7    <——隧道ID为Spoke2的公网IP
```

```bash
Spoke1 # diagnose vpn ike gateway list | grep "name\|tun_id"
name: Spoke1
tun_id: 100.1.1.2/::100.1.1.2    <——隧道ID为Hub的公网IP
```

```bash
Spoke2 # diagnose vpn ike gateway list | grep "name\|tun_id"
name: Spoke2
tun_id: 100.1.1.2/::100.1.1.2    <——隧道ID为Hub的公网IP
```

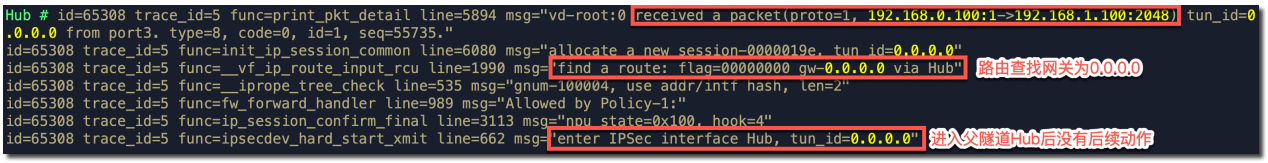
2. 使用 PC2（Spoke1 内网）访问 PC1（Hub 内网），可以正常通信。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
Windows IP 配置
以太网适配器 Ethernet0:
   IPv4 地址 . . . . . . . . . . . . : 192.168.1.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.1.1

C:\Users\Administrator.SUMMERICE2019>ping 192.168.0.100
正在 Ping 192.168.0.100 具有 32 字节的数据:
来自 192.168.0.100 的回复: 字节=32 时间=3ms TTL=126
来自 192.168.0.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.0.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.0.100 的回复: 字节=32 时间=2ms TTL=126
192.168.0.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 3ms, 平均 = 1ms
```

3. 使用 PC1（Hub 内网）访问 PC2（Spoke1 内网）或 PC3（Spoke2 内网），无法正常通信。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
Windows IP 配置
以太网适配器 Ethernet0:
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1

C:\Users\Administrator.SUMMERICE2019>ping 192.168.1.100
正在 Ping 192.168.1.100 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

4. 在 Hub 上抓包显示 ICMP Request 报文已被送入 IPSec Tunnel 接口。Debug Flow 虽然显示 ICMP Request 报文已被送入 IPSec Tunnel 接口，但路由查找的 `gw`（网关）为 0.0.0.0，正确的 `gw` 应为步骤 1 中看到的 Hub 与 Spoke1 的 IPSec 子连接 `Hub_0` 的 `tun_id` 200.1.1.1。



5. 在 Hub 上查看 IPSec Tunnel 接口的统计信息，可以看到 txe（发送错误）计数随着 PC1 访问 PC2 的流量在一直增长。

```bash
Hub # diagnose netlink interface list name Hub | grep "txe=\|if="
if=Hub family=00 type=768 index=16 mtu=1420 link=0 master=0
stat: rxp=15 txp=15 rxb=900 txb=900 rxe=0 txe=102 rxd=0 txd=0 mc=0 collision=0 @ time=1715074667

Hub # diagnose netlink interface list name Hub | grep "txe=\|^if="
if=Hub family=00 type=768 index=16 mtu=1420 link=0 master=0
stat: rxp=15 txp=15 rxb=900 txb=900 rxe=0 txe=108 <———txe计数增长 rxd=0 txd=0 mc=0 collision=0 @ time=1715074676
```

问题原因

1. 这是由于 FortiOS 7.0 新的 IPSec 内核设计导致的。FortiOS 7.0 之前版本的 Dynamic 模式 IPSec 使用“route tree”功能查找流量应该匹配的子隧道。FortiOS 7.0 之后将“route tree”功能移除，使用“tunn-id”用于标识 Dynamic 模式 IPSec 连接生成的每个子连接。当 Dynamic 模式的 IPSec 连接生成子连接（如 Hub_0，Hub_1……）时，将自动分配一个 `tunn-id` 给这个子连接，默认配置下，使用对端的公网 IP 作为隧道的 `tunn-id`。

2. 在 Spoke1（Static 类型的 IPSec）上查看去往 Hub 内网网段（192.168.0.0/24）的路由，可以看到路由信息中有对应的 `tun_id`，为 Hub 的公网地址 100.1.1.2，所以 Spoke 内网 PC 去往 Hub 内外 PC 的流量可以正确送入对应 `tun_id` 的 IPSec 隧道。

```bash
Spoke1 # get router info routing-table all | grep 192.168.0.0
S       192.168.0.0/24 [10/0] via Spoke1 tunnel 100.1.1.2, [1/0]
```

3. 在 Hub 上查看路由表，可以看到已配置的 Hub 去往 Spoke1 和 Spoke2 的静态路由在路由表中显示为 Tunnel 接口的直连路由，并未显示对应的 `tun_id`（这里应为 Spoke1 和 Spoke2 的公网 IP）。这是由于在 Hub 上配置的去往 Spoke 内网的静态路由并未指定出接口和下一跳为哪个 IPSec 子连接（Hub_0 或 Hub_1）。

```bash
Hub # get router info routing-table all | grep 192.168
C       192.168.0.0/24 is directly connected, port3
S       192.168.1.0/24 [10/0] is directly connected, Hub, [1/0]
S       192.168.2.0/24 [10/0] is directly connected, Hub, [1/0]
```

4. 当 PC1（Hub 内网）访问 PC2（Spoke1 内网）或 PC3（Spoke2 内网）时，发起流量进入 Hub 的 IPSec 隧道后，由于路由中没有 `tun_id` 标识，IPSec 不知道应该将其送往哪个子 IPSec 连接（Hub_0 或 Hub_1）。

解决方法 1 - 使用 add-route

1. 在 Spoke 上的 IPSec 二阶段配置中，修改源保护网段为明细保护网段，目标保护网段仍为 0.0.0.0/0。

```bash
Spoke1:
config vpn ipsec phase2-interface
    edit "Spoke1"
        set src-subnet 192.168.1.0 255.255.255.0
    next
end

Spoke2:
config vpn ipsec phase2-interface
    edit "Spoke1"
        set src-subnet 192.168.2.0 255.255.255.0
    next
end
```

2. 将 Hub 上配置的去往 Spoke 内外的静态路由禁用或删除。

```bash
config router static
    edit 2
        set status disable
        set dst 192.168.1.0 255.255.255.0
        set device "Hub"
    next
    edit 3
        set status disable
        set dst 192.168.2.0 255.255.255.0
        set device "Hub"
    next
end
```

3. 在 Hub 上的 IPSec 一阶段配置中，将 `add-route` 功能开启，根据 Spoke 协商的源保护网段自动添加去往 Spoke 内网的路由。

```bash
config vpn ipsec phase1-interface
    edit "Hub"
        set add-route enable
    next
end
```

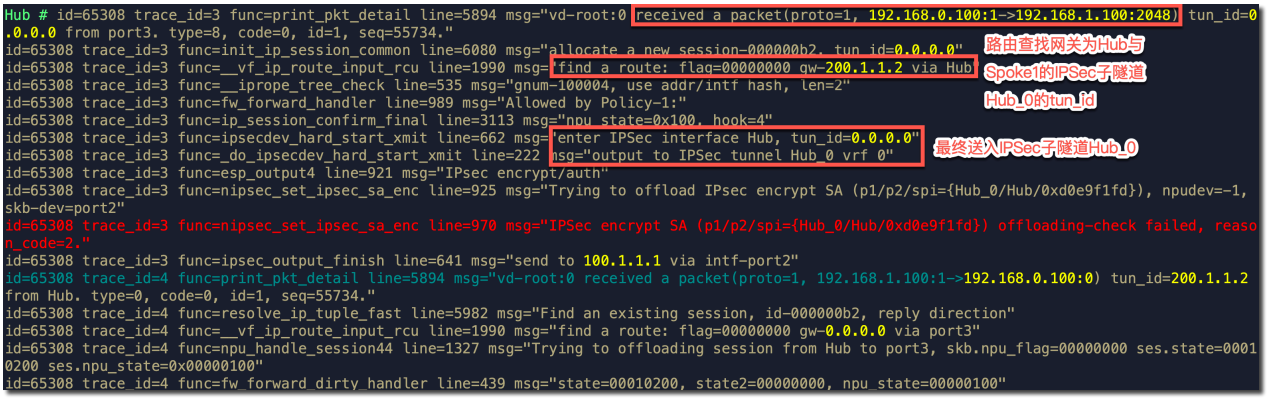
4. 在 Hub 上查看路由表，可以看到通过 `add-route` 功能添加的去往 Spoke 内网的路由自动添加了 `tunn-id` 标记，去往 Spoke1 内网的路由指定的是 IPSec 子连接 Hub_0 的 `tunn-id`，去往 Spoke2 内网的路由指定的是 IPSec 子连接 Hub_1 的 `tunn-id`。

```bash
Hub # get router info routing-table all | grep 192.168
C       192.168.0.0/24 is directly connected, port3
S       192.168.1.0/24 [15/0] via Hub tunnel 200.1.1.2, [1/0]
S       192.168.2.0/24 [15/0] via Hub tunnel 201.1.1.2, [1/0]
```

5. 使用 PC1（Hub 内网）访问 PC2（Spoke1 内网）或 PC3（Spoke2 内网），可以正常通信。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
以太网适配器 Ethernet0:
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1

C:\Users\Administrator.SUMMERICE2019>ping 192.168.1.100
正在 Ping 192.168.1.100 具有 32 字节的数据:
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=3ms TTL=126
192.168.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 3ms, 平均 = 1ms
```

6. Debug Flow 显示 ICMP Request 报文到达 FortiGate 后，路由查找网关为 Hub 与 Spoke1 的 IPSec 子隧道 Hub_0 的 `tun_id` 200.1.1.2。该流量最终可以被送入 IPSec 子隧道 Hub_0 送往 Spoke1。



解决方法 2 - 配置 VPN 路由下一跳

> **重要**
>
> 此方法要求 Spoke 的公网 IP 不会变化。

1. Spoke IPSec 二阶段仍然配置 0.0.0.0↔0.0.0.0 的保护网段，Hub 关闭 `add-route` 功能。

2. 在 Hub 上修改去往 Spoke 的静态路由，配置下一跳为 Spoke 对应的公网 IP（tun_id）。

> **重要**
>
> 出接口为 IPSec Tunnel 时，相关路由的下一跳（gateway）只能在 CLI 下配置。

```bash
config router static
    edit 2
        set dst 192.168.1.0 255.255.255.0
        set gateway 200.1.1.2
        set device "Hub"
    next
    edit 3
        set dst 192.168.2.0 255.255.255.0
        set gateway 201.1.1.2
        set device "Hub"
    next
end
```

3. 在 Hub 查看路由表，去往 Spoke 内网的路由下一跳变为 Spoke1 和 Spoke2 的公网 IP，对应相关已建立的 IPSec 子隧道 Hub_0、Hub_1 的 `tun_id`。

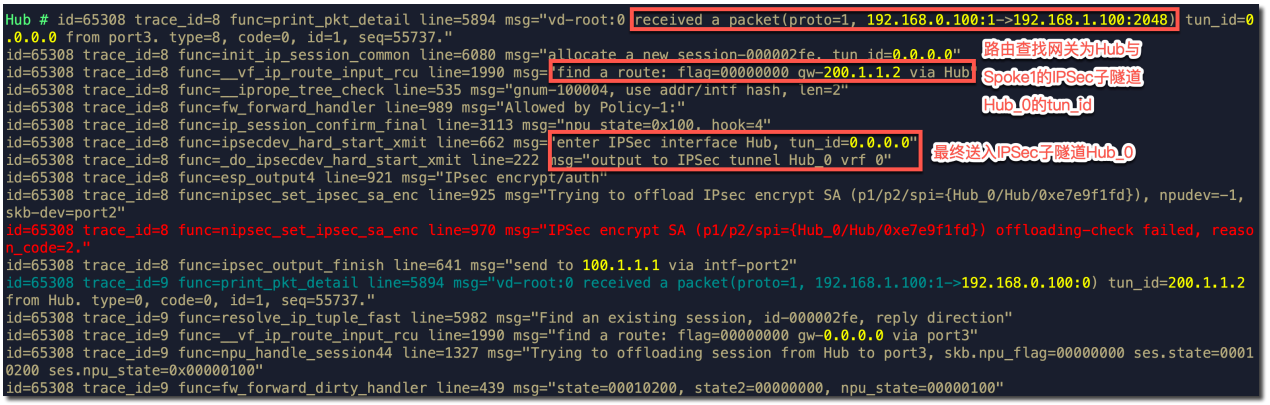
```bash
Hub # get router info routing-table all | grep 192.168
C       192.168.0.0/24 is directly connected, port3
S       192.168.1.0/24 [10/0] via Hub tunnel 200.1.1.2, [1/0]
S       192.168.2.0/24 [10/0] via Hub tunnel 201.1.1.2, [1/0]
```

4. 使用 PC1（Hub 内网）访问 PC2（Spoke1 内网）或 PC3（Spoke2 内网），可以正常通信。

```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
以太网适配器 Ethernet0:
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1

C:\Users\Administrator.SUMMERICE2019>ping 192.168.1.100
正在 Ping 192.168.1.100 具有 32 字节的数据:
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
```

```
来自 192.168.1.100 的回复: 字节=32 时间=3ms TTL=126
192.168.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 3ms, 平均 = 1ms
```

5. Debug Flow 显示 ICMP Request 报文到达 FortiGate 后，路由查找网关为 Hub 与 Spoke1 的 IPSec 子隧道 Hub_0 的 `tun_id` 200.1.1.2。该流量最终可以被送入 IPSec 子隧道 Hub_0 送往 Spoke1。



解决方法 3 - 使用 exchange-ip

1. Spoke IPSec 二阶段仍然配置 0.0.0.0↔0.0.0.0 的保护网段，Hub 关闭 `add-route` 功能。

2. 在所有 Spoke 上修改 IPSec 一阶段配置，开启 `exchange-ip` 功能，Spoke 与 Hub 进行 IKE 协商时会发送对应隧道已配置的 IPSec Tunnel IP 地址，隧道建立完成后，Hub 将使用 Spoke 发送的 IPSec Tunnel IP 作为对应的隧道 ID（`tun_id`）。

```bash
Spoke1:
config vpn ipsec phase1-interface
    edit "Spoke1"
        set exchange-interface-ip enable
    next
end

Spoke2:
config vpn ipsec phase1-interface
    edit "Spoke2"
        set exchange-interface-ip enable
    next
end
```

重要

在本文“配置信息”章节中已为所有 IPSec Tunnel 接口配置了 IP。

如果 Spoke 的 IPSec Tunnel 接口没有手动配置 IP 或没有通过 mode-cfg 获取 IP，则需要手工指定 `exchange-ip` 要交换的 IP 地址（该地址可自定义，不需要是设备上已存在的地址）：

```bash
config vpn ipsec phase1-interface
    edit "Spoke2"
        set exchange-interface-ip enable
        set exchange-ip-addr4 10.10.10.2
    next
end
```

3. IPSec 隧道建立后，查看 Hub 和 Spoke 的 IPSec 一阶段状态中的 `tun_id`，可以看到 Hub 建立了两个动态子 IPSec 连接（Hub_0 和 Hub_1），Hub 和 Spoke 显示已建立的隧道 ID（tun_id）均为对端公网 IP 地址。

```bash
Hub # diagnose vpn ike gateway list | grep "name\|tun_id"
name: Hub_0
tun_id: 10.10.10.2/::10.0.0.50     <——隧道ID为Spoke1的IPSec Tunnel IP
name: Hub_1
tun_id: 10.10.10.3/::10.0.0.53     <——隧道ID为Spoke2的IPSec Tunnel IP
```

4. 在 Hub 上修改去往 Spoke 的静态路由，配置下一跳为 Spoke 对应的 IPSec Tunnel 接口 IP。

> **重要**
>
> 出接口为 IPSec Tunnel 时，相关路由的下一跳（gateway）只能在 CLI 下配置。

```bash
config router static
    edit 2
        set dst 192.168.1.0 255.255.255.0
        set gateway 10.10.10.2
        set device "Hub"
    next
    edit 3
        set dst 192.168.2.0 255.255.255.0
        set gateway 10.10.10.3
        set device "Hub"
    next
end
```

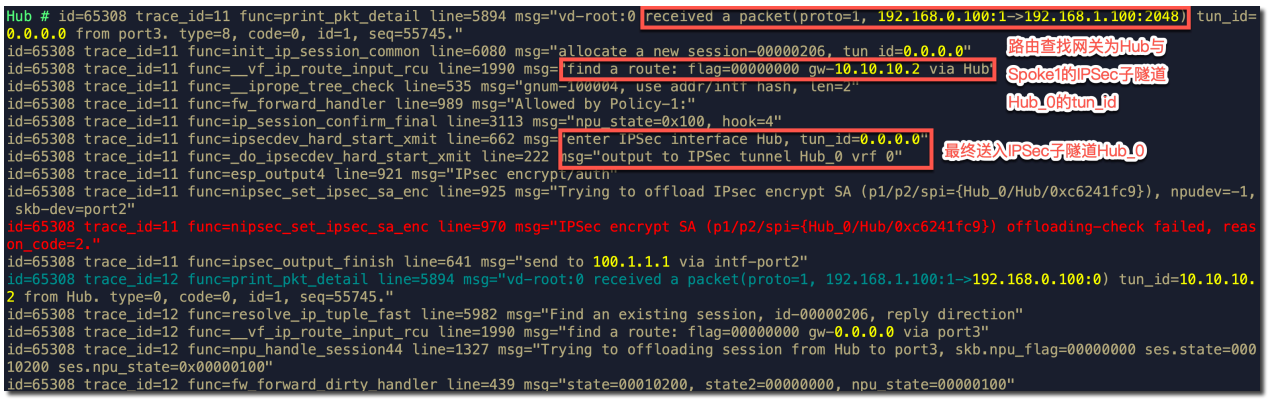
5. 在 Hub 查看路由表，去往 Spoke 内网的路由下一跳变为 Spoke1 和 Spoke2 的 IPSec Tunnel IP，对应相关已建立的 IPSec 子隧道 Hub_0、Hub_1 的 `tun_id`。

```bash
Hub # get router info routing-table all | grep 192.168
C       192.168.0.0/24 is directly connected, port3
S       192.168.1.0/24 [10/0] via Hub tunnel 10.10.10.2, [1/0]
S       192.168.2.0/24 [10/0] via Hub tunnel 10.10.10.3, [1/0]
```

6. 使用 PC1（Hub 内网）访问 PC2（Spoke1 内网）或 PC3（Spoke2 内网），可以正常通信。

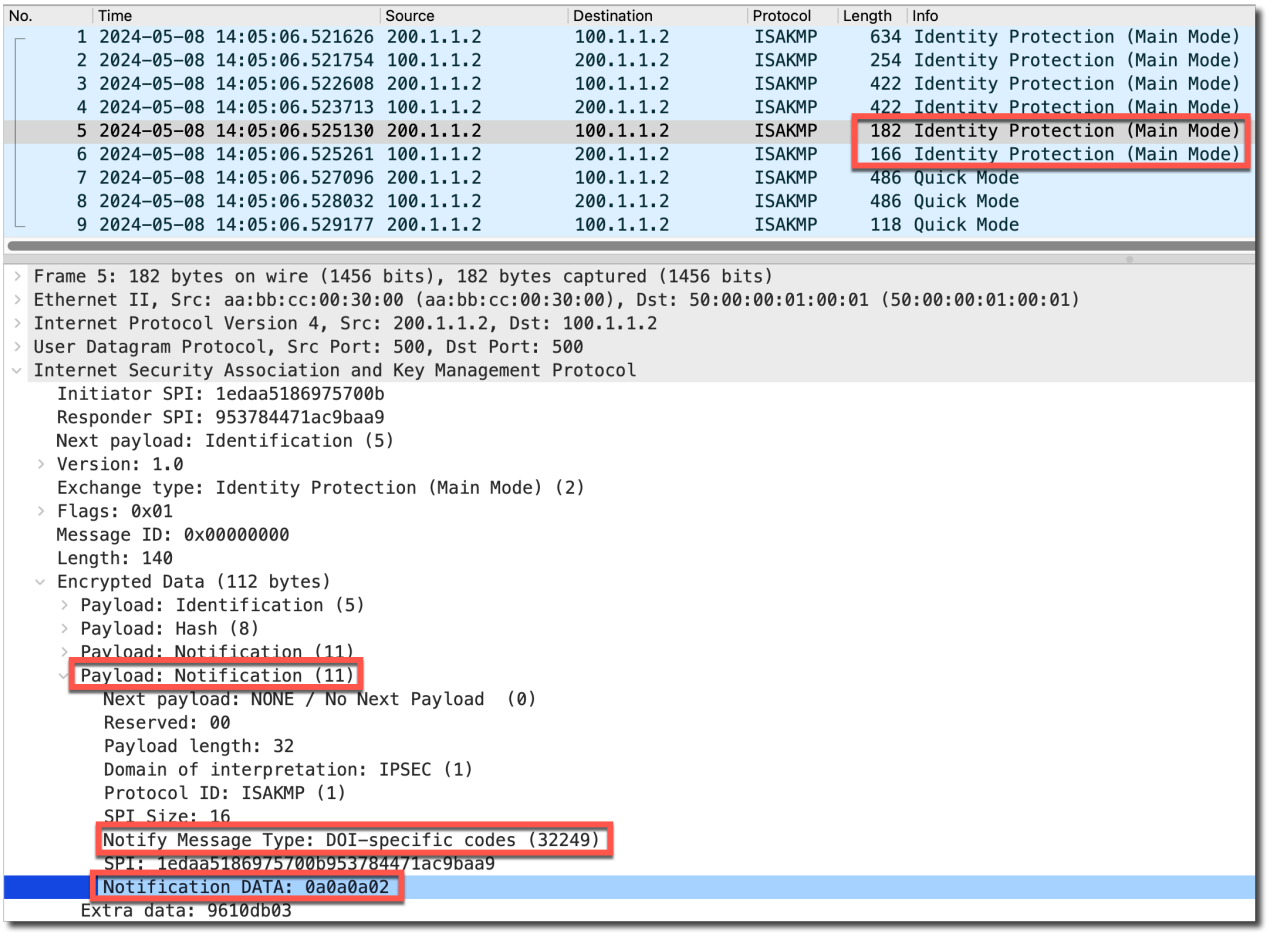
```bash
C:\Users\Administrator.SUMMERICE2019>ipconfig
以太网适配器 Ethernet0:
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.100
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1

C:\Users\Administrator.SUMMERICE2019>ping 192.168.1.100
正在 Ping 192.168.1.100 具有 32 字节的数据:
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.1.100 的回复: 字节=32 时间=3ms TTL=126
192.168.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 3ms, 平均 = 1ms
```

7. Debug Flow 显示 ICMP Request 报文到达 FortiGate 后，路由查找网关为 Hub 与 Spoke1 的 IPSec 子隧道 Hub_0 的 `tun_id` 200.1.1.2。该流量最终可以被送入 IPSec 子隧道 Hub_0 送往 Spoke1。

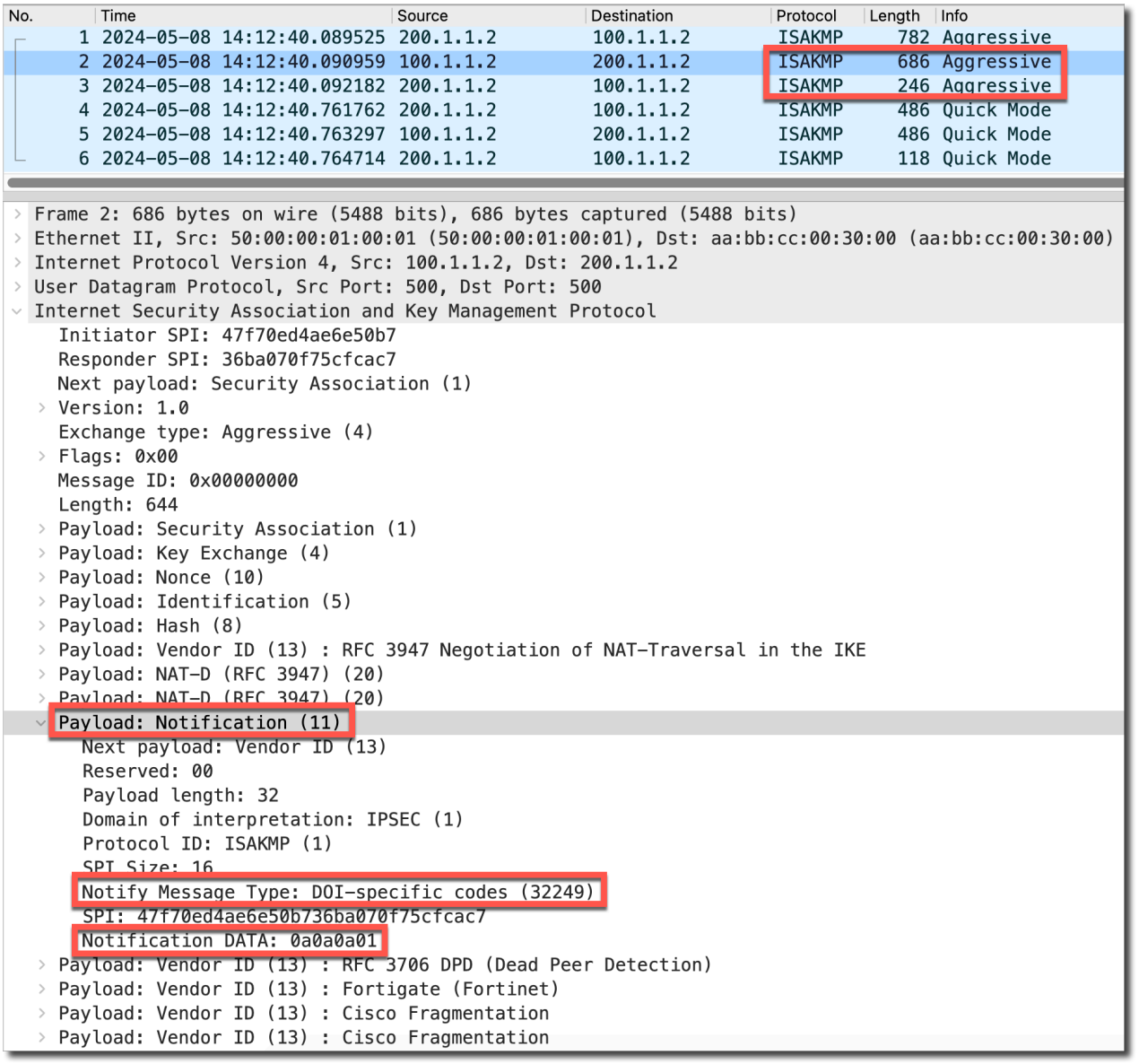


Exchange IP 交互报文

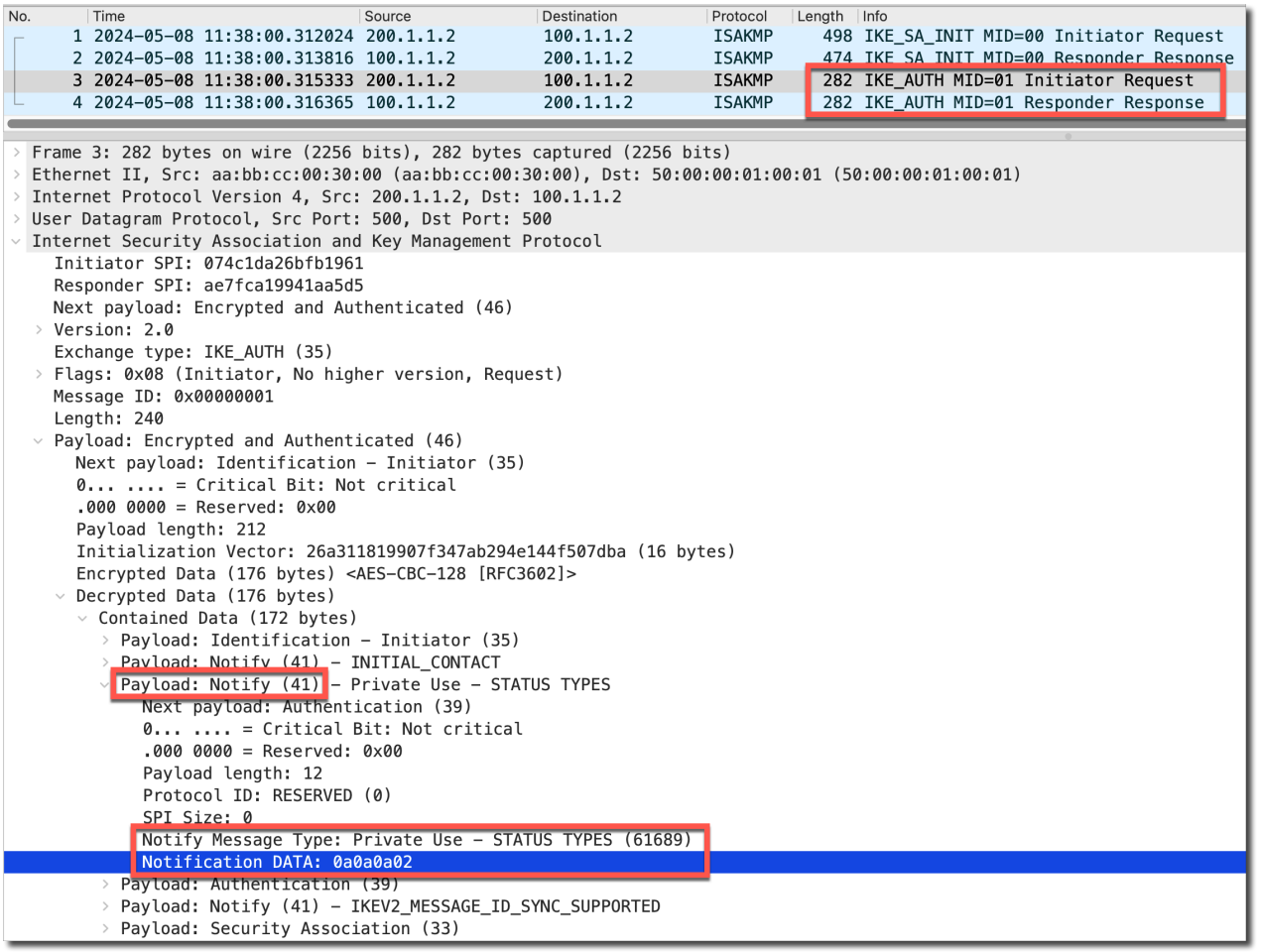
- IKEv1 主模式：位于第一阶段第 5、6 个报文中 `Payload: Notification (11)`（`Notify Message Type` 为 `DOI-specific codes (32249)`）的 `Notification DATA`。



- IKEv1 野蛮模式：位于第一阶段第 2、3 个报文中 `Payload: Notification (11)`（`Notify Message Type` 为 `DOI-specific codes (32249)`）的 `Notification DATA`。



- IKEv2：位于 IKE_AUTH 报文中 `Payload: Notify (41)`（`Notify Message Type` 为 `Private Use - STATUS TYPES (61689)`）的 `Notification DATA`。

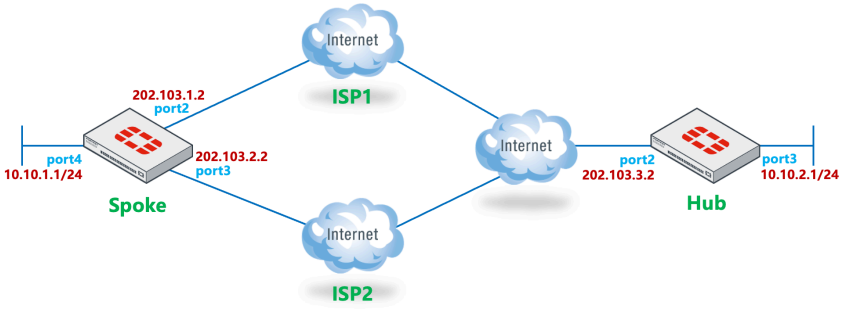


解决方法 4 - 使用动态路由

使用 BGP 或 OSPF 等动态路由，自动学习去往 Spoke 内网网段的路由会携带对应的 `tun_id`。

Hub 反向流量异步

网络拓扑



1. Spoke 有两条运营商线路 ISP1、ISP2，Hub 只有一条运营商线路。
2. Hub 创建 dynamic 模式的 IPSec VPN，Spoke 配置两条静态隧道与 Hub 建立 IPSec 连接。
3. Spoke 配置 SD-WAN，成员为两条 IPSec 隧道，并配置 SD-WAN 健康检查，检查源为 Spoke 内网口 port4 的 IP 地址，检查目标为 Hub 的内网口 port3 的 IP 地址。
4. 路由层面：
   - Spoke → Hub：Spoke 配置静态路由指向 SD-WAN 区域，IPSec 保护网段的源配置为明细网段 10.10.1.0/24。
   - Hub → Spoke：Hub 通过 IPSec 一阶段中的 `add-route` / 二阶段中的 `set route-overlap allow` 配置自动添加去往 Spoke 内网的路由，并在 Spoke 的两条隧道中负载。

配置信息

基础网络 IP、路由配置略。

Hub

1. 在公网接口 port2 上配置 IPSec 隧道，隧道类型为 dynamic，一阶段开启 `add-route`，二阶段保护网段为 0.0.0.0/0↔0.0.0.0/0，`route-overlap` 配置为 `allow`，`net-device` 为默认关闭状态。

> **相关信息**
>
> - `add-route` 的相关信息请参考：VPN → IPSec VPN → IPSec VPN 排错 → Hub-Spoke 静态路由问题 ↗。
> - `route-overlap` 的相关信息请参考：VPN → IPSec VPN → IPSec VPN 排错 → Hub-Spoke 双线路隧道抖动 ↗。

```bash
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set dpd on-idle
        set psksecret fortinet
        set dpd-retryinterval 60
    next
end

config vpn ipsec phase2-interface
    edit "Hub"
        set phase1name "Hub"
        set proposal aes128-sha1
        set route-overlap allow
    next
end
```

2. 配置防火墙策略放通 IPSec 隧道与内网口 port3 之间的流量。

```bash
config firewall policy
    edit 1
        set name "VPN_Hub"
        set srcintf "Hub" "port3"
        set dstintf "Hub" "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

Spoke

1. 在两条 ISP 线路接口 port2 和 port3 分别创建 IPSec 隧道配置，隧道类型为 static，第二阶段保护网段本端均使用明细的内网网段：10.10.1.0/24↔0.0.0.0/0。

```bash
config vpn ipsec phase1-interface
    edit "vpn_line1"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set dpd on-idle
        set remote-gw 202.103.3.3
        set psksecret fortinet
    next
    edit "vpn_line2"
        set interface "port3"
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set dpd on-idle
        set remote-gw 202.103.3.3
        set psksecret fortinet
    next
end

config vpn ipsec phase2-interface
    edit "vpn_line1"
        set phase1name "vpn_line1"
        set proposal aes128-sha1
        set auto-negotiate enable
        set src-subnet 10.10.1.0 255.255.255.0
    next
    edit "vpn_line2"
        set phase1name "vpn_line2"
        set proposal aes128-sha1
        set auto-negotiate enable
        set src-subnet 10.10.1.0 255.255.255.0
    next
end
```

2. 将两个 IPSec 接口加入 SD-WAN 区域，并配置 SD-WAN 健康检查，检查源为 Spoke 内网口 port4 的 IP 地址，检查目标为 Hub 的内网口 port3 的 IP 地址。

```bash
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "vpn_line1"
        next
        edit 2
```

```
            set interface "vpn_line2"
        next
    end
    config health-check
        edit "to_Hub"
            set server "10.10.2.1"
            set update-static-route disable
            set source 10.10.1.1
            set members 0
        next
    end
end
```

3. 配置防火墙策略放通 2 条 IPSec 隧道（SD-WAN 区域）与内网口 port4 之间的流量。

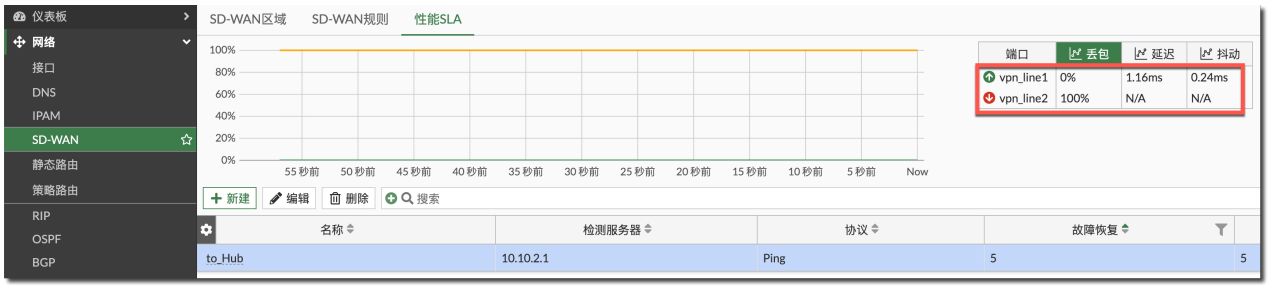
```bash
config firewall policy
    edit 1
        set name "all"
        set srcintf "virtual-wan-link" "port4"
        set dstintf "port4" "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. 配置去往 Hub 内网网段（192.168.0.0/24）的静态路由，出接口选择 SD-WAN 区域。

```bash
config router static
    edit 3
        set dst 10.10.2.0 255.255.255.0
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

问题现象

1. Spoke 与 Hub 的两条 IPSec 隧道建立后，SD-WAN 健康检查工作异常，在 vpn_line1 隧道上显示 `alive`，vpn_line2 隧道显示 `dead`。




```bash
Spoke # diagnose sys sdwan health-check status
Health Check(to_Hub):
Seq(1 vpn_line1): state(alive), packet-loss(0.000%) latency(1.092), jitter(0.172), mos(4.404), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0x0
Seq(2 vpn_line2): state(dead), packet-loss(100.000%) sla_map=0x0
```

2. 在 Spoke 上抓包，可以看到健康检查从 Spoke 两条隧道 vpn_line1、vpn_line2 分别发送了 ICMP Request，但 ICMP Reply 都是从 vpn_line1 返回的。这说明 Hub 在返回 ICMP Reply 报文时送错了隧道，所以 Spoke 认定 vpn_line2 为 `dead` 状态。

```bash
Spoke # diagnose sniffer packet any 'host 10.10.2.1 and icmp' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.1 and icmp]
0.277013 vpn_line1 out 10.10.1.1 → 10.10.2.1: icmp: echo request
```

```
0.277148 vpn_line2 out 10.10.1.1 → 10.10.2.1: icmp: echo request
0.278120 vpn_line1 in 10.10.2.1 → 10.10.1.1: icmp: echo reply
0.278160 vpn_line1 in 10.10.2.1 → 10.10.1.1: icmp: echo reply
```

3. 在 Hub 上查看路由表，可以看到去往 Spoke 内网（10.10.1.0/24）的路由等价负载到了两条 IPSec 隧道（Hub_0、Hub_1）上。

```bash
Hub # get router info routing-table all
......
Routing table for VRF=0
S       10.10.1.0/24 [15/0] via Hub_1 tunnel 202.103.1.2, [1/0]
                     [15/0] via Hub_0 tunnel 202.103.2.2, [1/0]
......
```

4. 在 Hub 上查看 Spoke 发起的健康检查的 2 条会话，由于 Hub 未开启 `net-device`，所以两条会话的入接口均为 dynamic 模式的 IPSec 接口（Hub，index = 16）。

```bash
Hub # diagnose netlink interface list Hub | grep index
if=Hub family=00 type=768 index=16 mtu=1420 link=0 master=0

session info: proto=1 proto_state=00 duration=6288 expire=59 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/202.103.1.2 vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=501840/12546/1 reply=501840/12546/1 tuples=2
tx speed(Bps/kbps): 79/0 rx speed(Bps/kbps): 79/0
orgin->sink: org pre->in, reply out->post dev=16->8/8->16 <-----入接口为Hub接口 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.10.1.1:31400->10.10.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.2.1:31400->10.10.1.1:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=15849 auth_info=0 chk_client_info=0 vd=0
serial=0000a1ef tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local

session info: proto=1 proto_state=00 duration=3401 expire=59 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/202.103.2.2 vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=271160/6779/1 reply=271160/6779/1 tuples=2
tx speed(Bps/kbps): 79/0 rx speed(Bps/kbps): 79/0
orgin->sink: org pre->in, reply out->post dev=16->8/8->16 <-----入接口为Hub接口  gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.10.1.1:32554->10.10.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.2.1:32554->10.10.1.1:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=15849 auth_info=0 chk_client_info=0 vd=0
serial=0000a704 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

5. Hub 上显示 Spoke 与本端建立了两条 IPSec 隧道 Hub_0 和 Hub_1。Hub 的 Debug Flow 显示所有的 ICMP Reply 都被送进了 Hub_0 隧道，从 Hub_1 隧道收到的 ICMP Request 对应的 ICMP Reply 从 Hub_0 返回，流量没有源进源出。

```bash
Hub # get vpn ipsec tunnel summary
'Hub_0' 202.103.1.2:0  selectors(total,up): 1/1  rx(pkt,err): 158890/0  tx(pkt,err): 317780/0
'Hub_1' 202.103.2.2:0  selectors(total,up): 1/1  rx(pkt,err): 158890/0  tx(pkt,err): 0/0

Hub # diagnose debug flow filter daddr 10.10.1.1
Hub # diagnose debug flow filter proto 1
Hub # diagnose debug enable
Hub # diagnose debug flow trace start 200

↓↓↓↓第1个ICMP Reply（ICMP Request从Hub_0接收），从Hub_0返回↓↓↓↓
id=65308 trace_id=20 func=print_pkt_detail line=6005 msg="vd-root:0 received a packet(proto=1, 10.10.2.1:31400-
>10.10.1.1:0) tun_id=0.0.0.0 from local. type=0, code=0, id=31400, seq=981."
id=65308 trace_id=20 func=resolve_ip_tuple_fast line=6107 msg="Find an existing session, id-0000a1ef, reply direction"
```

```
id=65308 trace_id=20 func=ip_session_core_in line=6732 msg="dir=1, tun_id=202.103.1.2"    ←——针对Hub_0收到的ICMP Request
返回的Reply，tun_id为Spoke的vpn_line1公网IP
id=65308 trace_id=20 func=ipsecdev_hard_start_xmit line=662 msg="enter IPSec interface Hub, tun_id=202.103.1.2"
id=65308 trace_id=20 func=_do_ipsecdev_hard_start_xmit line=222 msg="output to IPSec tunnel Hub_0, tun_id=202.103.1.2,
vrf 0"    ←——送到了Hub_0，这个送对了
id=65308 trace_id=20 func=esp_output4 line=917 msg="IPsec encrypt/auth"
id=65308 trace_id=20 func=ipsec_output_finish line=676 msg="send to 202.103.3.1 via intf-port2"

↓↓↓↓第2个ICMP Reply（ICMP Request从Hub_1接收），从Hub_0返回↓↓↓↓
id=65308 trace_id=19 func=print_pkt_detail line=6005 msg="vd-root:0 received a packet(proto=1, 10.10.2.1:31596-
>10.10.1.1:0) tun_id=0.0.0.0 from local. type=0, code=0, id=31596, seq=981."
id=65308 trace_id=19 func=resolve_ip_tuple_fast line=6107 msg="Find an existing session, id-0000a2bd, reply direction"
id=65308 trace_id=19 func=ip_session_core_in line=6732 msg="dir=1, tun_id=202.103.2.2"    ←——针对Hub_1收到的ICMP Request
返回的Reply，tun_id为Spoke的vpn_line2公网IP
id=65308 trace_id=19 func=ipsecdev_hard_start_xmit line=662 msg="enter IPSec interface Hub, tun_id=202.103.2.2"
id=65308 trace_id=19 func=_do_ipsecdev_hard_start_xmit line=222 msg="output to IPSec tunnel Hub_0, tun_id=202.103.1.2,
vrf 0"    ←——送到了Hub_0，这个送错了
id=65308 trace_id=19 func=esp_output4 line=917 msg="IPsec encrypt/auth"
id=65308 trace_id=19 func=ipsec_output_finish line=676 msg="send to 202.103.3.1 via intf-port2"
```

问题原因

1. 默认配置下，Hub 的 dynamic 模式的 IPSec 连接的 `net-device` 配置为 `disable` 状态，来自同一 Spoke 的多条隧道共享一个 phase1-interface，Hub 做回包决策是：
   - 查路由（通常是 ECMP，指向同一 phase1）。
   - 匹配 selector/proxy-id，也就是感兴趣流。
2. 如果多条隧道的 selector（感兴趣流）相同，多条 SA 会同时匹配。而 Hub 没有“入向隧道”的绑定关系，于是回包会落到“优先级更高/先匹配”的那条路由对应的隧道，形成不对称回流。
3. 典型触发条件：
   - Hub 为 dynamic 模式隧道，`net-device` 为 `disable` 状态。
   - 同一 Spoke 多条隧道。
   - Spoke 多条隧道的 selector/proxy-id（感兴趣流）重复。
   - Hub 回程为 ECMP/聚合隧道（多条候选路径同权重）。

解决方法 1-Spoke 配置 location-id

重要

- 配置要求：
  - 为每台 Spoke 配置唯一的 `location-id`。这样 Hub 会把来自同一 Spoke 的多条拨号隧道分为一组，并 **缓存“原始入向隧道”**。回包优先在该组里选入向那条隧道，保持对称。
  - `location-id` 只是用于 Hub 识别同一 Spoke 多条隧道的标识，并不是真实的 IP 地址，但建议配置为能标识该 Spoke 的 IP 地址，如 loopback 接口地址。
- 工作原理：
  - 分组：Hub 接收报文时，依据远端上报的 `location-id`，将相同 `location-id`、挂在同一 phase1-interface 的拨号隧道划为同一站点组。
  - 隧道绑定：Hub 会缓存流量的原始入向隧道，形成一个“隧道组键”，基于 `phase1 + location-id + SPI` 的内部标识。
  - 回包选路：回包时优先在该组内直接选回入向那条隧道。若不可用，再在同组内切换，不会跨到其它 Spoke。

1. 在 Spoke 上配置 `location-id`（同一台 Spoke 的多条隧道用相同的 `location-id`）。

```bash
config system settings
    set location-id 10.10.1.1
end
```

2. 重建 Spoke 与 Hub 间的隧道，然后在 Hub 查看 IPSec 一阶段状态。可以看到 Spoke 将 `location-id` 传递给了 Hub，Hub 将 Spoke 建立的两条隧道标记了相同的 `remote_location: 10.10.1.1`。

提示

`location-id` 通过 IKE 协商传递：

- IKEv1：`Main Mode` 的第 5/6 个报文中的 Fortinet 私有字段。
- IKEv2：`IKE_AUTH` 报文中的 Fortinet 私有字段。

```bash
Hub # diagnose vpn ike gateway list

vd: root/0
name: Hub_0
version: 1
interface: port2 4
addr: 202.103.3.3:500 → 202.103.1.2:500
tun_id: 202.103.1.2/::10.0.0.62
remote_location: 10.10.1.1    ←——Spoke配置的location-id
network-id: 0
transport: UDP
created: 3s ago
peer-id: 202.103.1.2
peer-id-auth: no
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 72 d1e24bba9ed24c5b/d31a2eaa6a2cec9f
  direction: responder
  status: established 3-3s ago = 0ms
  proposal: aes128-sha256
  key: 827f2d4a98e8918f-77fd9d748ea39e48
  QKD: no
  lifetime/rekey: 86400/86126
  DPD sent/recv: 00000000/00000000
  peer-id: 202.103.1.2

vd: root/0
name: Hub_1
version: 1
interface: port2 4
addr: 202.103.3.3:500 → 202.103.2.2:500
tun_id: 202.103.2.2/::10.0.0.63
remote_location: 10.10.1.1    ←——Spoke配置的location-id
network-id: 0
transport: UDP
created: 2s ago
peer-id: 202.103.2.2
peer-id-auth: no
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 73 4324ed445892ef2b/c7efb986832df49b
  direction: responder
  status: established 2-2s ago = 0ms
  proposal: aes128-sha256
  key: b40c73f8b0deb162-7144c5911025e8fe
  QKD: no
  lifetime/rekey: 86400/86127
  DPD sent/recv: 00000000/00000000
  peer-id: 202.103.2.2
```

3. 在 Spoke 查看 SD-WAN 健康检查状态，两条隧道均为 `alive` 状态。

```bash
Spoke # diagnose sys sdwan health-check status
Health Check(to_Hub):
Seq(1 vpn_line1): state(alive), packet-loss(0.000%) latency(1.358), jitter(0.377), mos(4.403), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0×0
Seq(2 vpn_line2): state(alive), packet-loss(0.000%) latency(1.194), jitter(0.249), mos(4.404), bandwidth-up(9999999),
bandwidth-dw(9999999), bandwidth-bi(19999998) sla_map=0×0
```

4. 在 Spoke 上抓包，可以看到 ICMP Reply 可以被 Hub 从正确的隧道送回。

```bash
Spoke # diagnose sniffer packet any 'host 10.10.2.1 and icmp' 4
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.2.1 and icmp]
0.241158 vpn_line1 out 10.10.1.1 → 10.10.2.1: icmp: echo request
0.241283 vpn_line2 out 10.10.1.1 → 10.10.2.1: icmp: echo request
0.242357 vpn_line1 in 10.10.2.1 → 10.10.1.1: icmp: echo reply
0.242594 vpn_line2 in 10.10.2.1 → 10.10.1.1: icmp: echo reply
```

解决方法 2-Hub 开启 net-device

> **注意**
>
> 由于会带来 Hub 的性能问题，在 Spoke 数量较多的场景下不推荐此方法。

1. Hub 的 IPSec 一阶段开启 `net-device`，隧道会自动重建。

```bash
config vpn ipsec phase1-interface
    edit "Hub"
        set net-device enable
    next
end
```

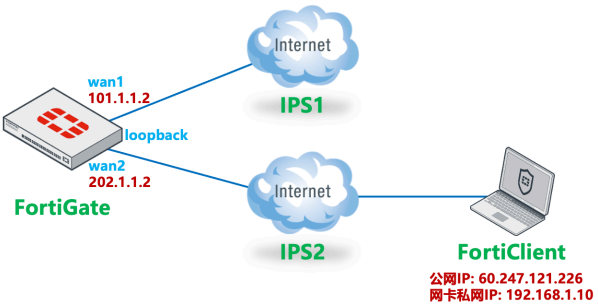
2. 在 Hub 上查看 Spoke 发起的健康检查的 2 条会话，由于 Hub 开启了 `net-device`，所以两条会话的入接口为动态生成的 tunnel 接口（Hub_0：index = 16，Hub_1：index = 17）。此时源进源出特性可以直接根据会话生效。

```bash
Hub # diagnose netlink interface list | grep Hub
if=Hub family=00 type=768 index=16 mtu=1420 link=0 master=0
if=Hub_0 family=00 type=768 index=17 mtu=1420 link=16 master=0
if=Hub_1 family=00 type=768 index=18 mtu=1420 link=16 master=0

session info: proto=1 proto_state=00 duration=6836 expire=59 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/202.103.1.2 vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=545680/13642/1 reply=545680/13642/1 tuples=2
tx speed(Bps/kbps): 80/0 rx speed(Bps/kbps): 80/0
orgin->sink: org pre->in, reply out->post dev=17->8/8->17 <——入接口为Hub_0接口 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.10.1.1:31400->10.10.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.2.1:31400->10.10.1.1:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=15849 auth_info=0 chk_client_info=0 vd=0
serial=0000a1ef tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local

session info: proto=1 proto_state=00 duration=3949 expire=59 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/202.103.2.2 vlan_cos=0/0
state=local may_dirty
statistic(bytes/packets/allow_err): org=315000/7875/1 reply=315000/7875/1 tuples=2
tx speed(Bps/kbps): 80/0 rx speed(Bps/kbps): 80/0
orgin->sink: org pre->in, reply out->post dev=18->8/8->18 <——入接口为Hub_1接口 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.10.1.1:32554->10.10.2.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.10.2.1:32554->10.10.1.1:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=15849 auth_info=0 chk_client_info=0 vd=0
serial=0000a704 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

IPsec 绑定 Loopback 时无法建立

问题现象

IPSec 隧道绑定到 loopback 接口，且公网出口存在多 ISP 的 ECMP 场景时，IPSec 隧道可能无法建立。



1. FortiGate 有两个 ISP 出口，默认路由在两个 ISP 出口上等价负载。

```bash
FortiGate #   get router info routing-table all
......
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 101.1.1.1, wan1, [1/0]
                  [10/0] via 202.1.1.1, wan2, [1/0]
......
```

2. IPSec 连接绑定在 loopback 接口，`local-gw` 配置为 wan2 的 IP 地址，NAT-T 配置为默认的开启状态（`set nattraversal enable`）。

```bash
config vpn ipsec phase1-interface
    edit "IPSec_Dialup"
        set type dynamic
        set interface "loopback1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set dpd on-idle
        set eap enable
        set eap-identity send-request
        set authusrgrp "IPSec-Group"
        set ipv4-start-ip 10.13.0.129
        set ipv4-end-ip 10.13.0.254
        set dns-mode auto
        set ipv4-split-include "intranet"
        set nattraversal enable
        set psksecret xxxxxxxx
    next
end
```

3. FortiClient 连接 FortiGate 的 IPSec，拨号连接失败。在 FortiClient 侧抓包可以看到一直在重发 IKEv2 第一个协商包 `IKE_SA_INIT (Initiator Request)`，但没有收到 FortiGate 的任何响应。

```bash
1 16:10:20.828351600 192.168.1.10 202.1.1.2 a69f9452f9fc9bc4 IKE_SA_INIT MID=00 Initiator Request
2 16:10:23.506134300 192.168.1.10 202.1.1.2 a69f9452f9fc9bc4 IKE_SA_INIT MID=00 Initiator Request
3 16:10:26.510410100 192.168.1.10 202.1.1.2 a69f9452f9fc9bc4 IKE_SA_INIT MID=00 Initiator Request
4 16:10:29.511326300 192.168.1.10 202.1.1.2 a69f9452f9fc9bc4 IKE_SA_INIT MID=00 Initiator Request
......
```

4. 从 FortiGate 处抓包可以看到 FortiGate 从 wan2 接收了 FortiClient 发送的 `IKE_SA_INIT (Initiator Request)`，但从 wan1 接口返回了 IKEv2 第 2 个包 `IKE_SA_INIT (Responder Response)`，没有源进源出。

```bash
R2-601F # diagnose sniffer packet any "host 60.247.121.226" 4 0 l
interfaces=[any]
filters=[host 60.247.121.226]
2026-03-13 11:14:58.393925 wan2 in 60.247.121.226.500 → 202.1.1.2.500: udp 328
2026-03-13 11:14:58.393985 wan1 out 202.1.1.2.500 → 60.247.121.226.500: udp 268
```

```
2026-03-13 11:15:04.403999 wan2 in 60.247.121.226.500 → 202.1.1.2.500: udp 328
2026-03-13 11:15:04.404026 wan1 out 202.1.1.2.500 → 60.247.121.226.500: udp 268
......
```

5. 在 FortiGate 上查看 IKEv2 会话，reply 没有遵循 session 优先的原则源进源出，而是独立进行了 RIB 的查找。

```bash
session info: proto=17 proto_state=01 duration=0 expire=179 timeout=0 refresh_dir=both flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log local may_dirty
statistic(bytes/packets/allow_err): org=401/1/1 reply=288/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->in, reply out->post dev=41->46/46->39 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 60.247.121.226:21733->202.1.1.2:500(0.0.0.0:0)
hook=post dir=reply act=noop 202.1.1.2:500->60.247.121.226:21733(0.0.0.0:0)
misc=0 policy_id=101 pol_uuid_idx=16314 auth_info=0 chk_client_info=0 vd=0
serial=7a971a9d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x001000
no_ofld_reason:  local
```

问题原因

在未开启强制 NAT-T（`set nattraversal forced`）的情况下，IKE（IPSec 一阶段）与 ESP（IPSec 二阶段）是独立会话。隧道绑定到 loopback 且存在冗余 ISP 时，IKE 与 ESP 流量可能走不同路径，入向与出向 IKE 协商可能经过不同接口，导致部分客户端无法建立隧道。

解决方法

在 IPSec 一阶段配置中开启 `loopback-asymroute` 控制 IKE 流量是否允许异步路由，该命令仅在一阶段绑定接口为 loopback 时可见。

```bash
config vpn ipsec phase1-interface
    edit <name>
        set interface <loopback interface name>
        set loopback-asymroute {enable | disable}
    next
end
```

- `enable`（默认）：允许入向/出向 IKE 流量走不同接口（允许异步）。
- `disable`：要求入向/出向 IKE 流量走同一接口（源进源出）。

IPSec 解密

加密 IPSec 报文类型

IKE 协商阶段

1. IKE 协商阶段，部分 IKE 报文会被加密：
   - IKEv1：
     - 主模式（Main）：Phase 1 第 5、6 个包，Phase 2 全部报文。
     - 野蛮模式（Aggressive）：Phase 1 第 3 个包，Phase 2 全部报文。
   - IKEv2：
     - IKE_AUTH 阶段的全部报文。
     - CREATE_CHILD_SA 的全部报文。
2. 报文封装格式（默认端口）：
   - 没有 NAT-T：以上报文为 UDP 500 封装。
   - 有 NAT-T：以上报文为 UDP 4500 封装。
3. 如果在 config system settings 下修改了 ike-port 为非 500，例如修改为 5555，不论有没有 NAT-T，IKE 协商报文全部用 UDP 5555 来封装。
4. 不仅包含 IKE 协商报文，IPSec 协商成功后或协商中发送的被加密的 Information 报文也属于此类型。
5. 当遇到 IKE 协商失败的情况时，如果 IKE Debug 无法判断具体的原因，可能需要解密 IKE 报文来进行判断。

IPSec 业务阶段

1. IPSec 业务阶段，所有的业务流量会被 ESP 封装并加密。
2. 报文封装格式：
   - 没有 NAT-T：所有的加密报文为 IP Protocol 50 的 ESP 包。
   - 有 NAT-T：所有的加密报文为 UDP 4500 封装（默认端口）的 ESP 包。
3. 如果在 config system settings 下修改了 ike-port 为非 500，例如修改为 5555：
   - 没有 NAT-T，所有的加密报文为 IP Protocol 50 的 ESP 包。
   - 有 NAT-T：所有的加密报文为 UDP 5555 封装的 ESP 包。
4. 当遇到 IPSec 隧道两端内网的 PC 无法互相通信时，在 FortiGate 上抓取明文，报文都已送入隧道接口，但不确定 ESP 报文是否正确发出和接收，此时可以抓取 ESP 并解密后判断业务流量丢失是发生在 FortiGate 还是外部网络。

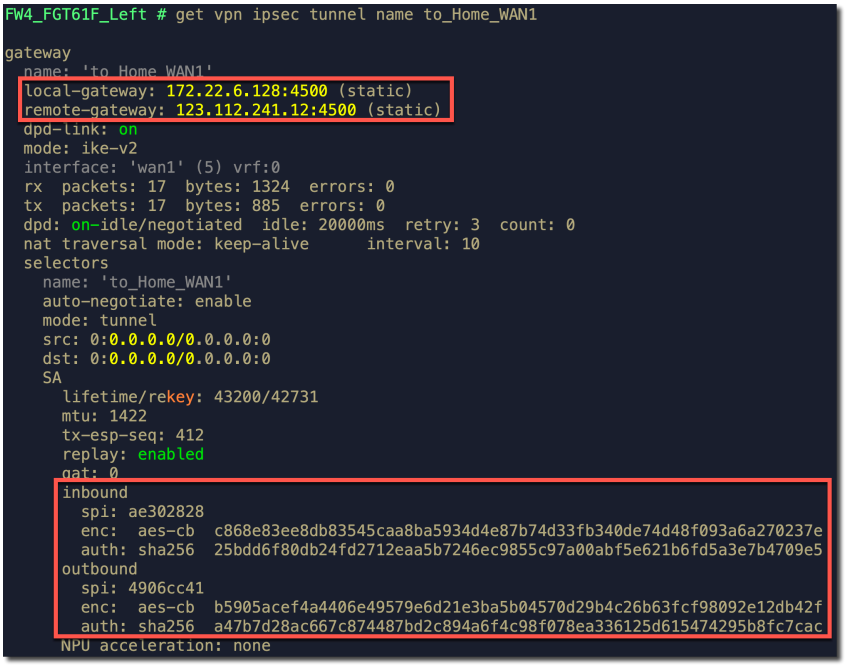
ESP 解密步骤

1. 关闭 FortiGate IPSec 一阶段配置中的 npu-offload（VM 虚机不需要执行此步骤），否则抓出的 ESP 包解密后无法看到正确的明文。

> **重要**
>
> 执行此操作后，IPSec 隧道会新进行协商，且之后 IPSec 业务流量无法被 NP 加速，请在非业务窗口期执行。
>
> 或不执行此步，使用中间可抓包的设备（如交换机的端口镜像）执行抓包操作。

```bash
config vpn ipsec phase1-interface
    edit "to_Home_WAN1"
        set npu-offload disable
    next
end
```

2. 执行如下 `get vpn ipsec tunnel name <Tunnel_name>` 命令收集当前 IPSec 隧道的解密信息：



- local-gateway：本地 IPSec 加密点 IP 为 172.22.6.128
- remote-gateway：对端 IPSec 加密点 123.112.241.12
- inbound（解密方向）：
  - SPI：ae302828
  - enc：解密算法为 AES - CBC，解密密钥为 c868e83ee8db83545caa8ba5934d4e87b74d33fb340de74d48f093a6a270237e
  - auth：验证算法为 sha256，验证密钥为 25bdd6f80db24fd2712eaa5b7246ec9855c97a00abf5e621b6fd5a3e7b4709e5
- outbound（加密方向）：
  - SPI：4906cc41
  - enc：解密算法为 AES - CBC，解密密钥为 b5905acef4a4406e49579e6d21e3ba5b04570d29b4c26b63fcf98092e12db42f
  - auth：验证算法为 sha256，验证密钥为 a47b7d28ac667c874487bd2c894a6f4c98f078ea336125d615474295b8fc7cac

3. 以上密钥信息也可以通过 `diagnose debug application ike -1` 的输出结果中找到。

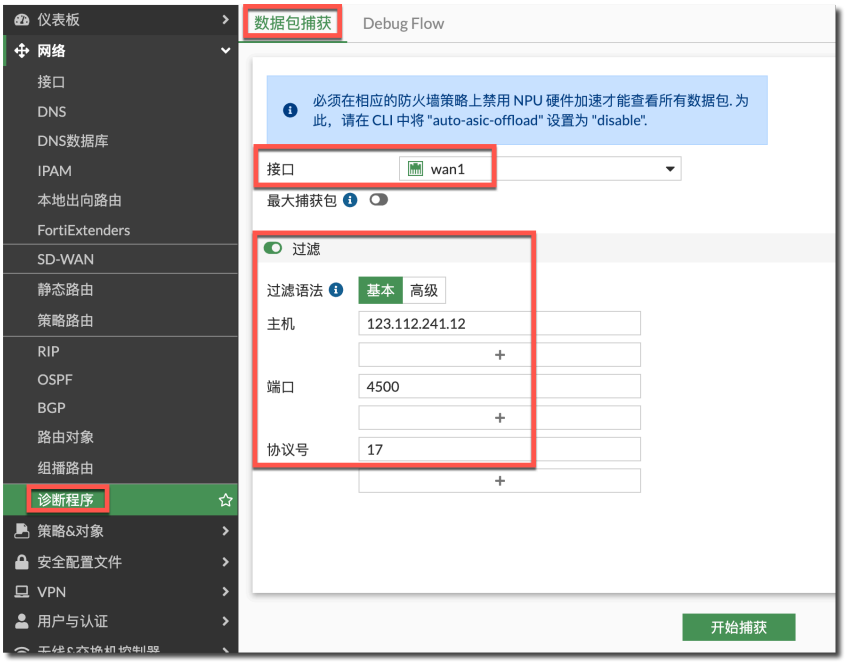
```bash
FortiGate # diagnose debug application ike -1
FortiGate # diagnose debug enable
......
ike 0:to_Home_WAN1:221:to_Home_WAN1:18115: IPsec SA dec spi ae302828 key
32:C868E83EE8DB83545CAA8BA5934D4E87B74D33FB340DE74D48F093A6A270237E auth
32:25BDD6F80DB24FD2712EAA5B7246EC9855C97A00ABF5E621B6FD5A3E7B4709E5
ike 0:to_Home_WAN1:221:to_Home_WAN1:18115: IPsec SA enc spi 4906cc41 key
32:B5905ACEF4A4406E49579E6D21E3BA5B04570D29B4C26B63FCF98092E12DB42F auth
32:A47B7D28AC667C874487BD2C894A6F4C98F078EA336125D615474295B8FC7CAC
ike 0:to_Home_WAN1:221:to_Home_WAN1:18115: added IPsec SA: SPIs=ae302828/4906cc41
ike 0:to_Home_WAN1: HA send IKE connection add 172.22.6.128→123.112.241.12
......
```

4. 在 IPSec 通讯接口上抓取 ESP 包：

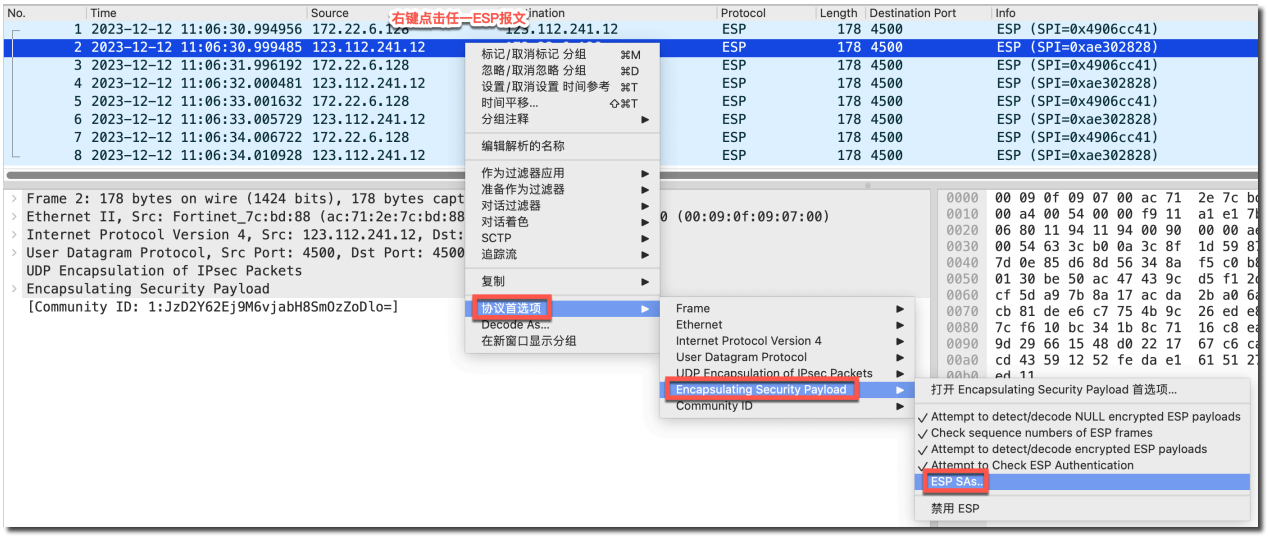
- 没有 NAT-T：抓取 IP Protocol 50 的报文。
- 有 NAT-T：抓取 UDP 4500 的报文（默认端口，如果在 config system settings 下修改了 ike-port，则抓取 UDP 对应的端口号，详见 VPN 技术 → IPSec VPN → 点到点 VPN → 修改 IPSec 协商端口 ↗章节）。

5. 这里我们以有 NAT-T 环境为例，在 ESP 报文的出接口 wan1 上抓取 UDP 4500 的 ESP 加密报文，如果没有 NAT-T，则抓取 IP Protocol 50 的报文（GUI 方式抓包方法详见故障排查 → Sniffer 工具 → 通过 GUI 界面抓包 ↗章节）。

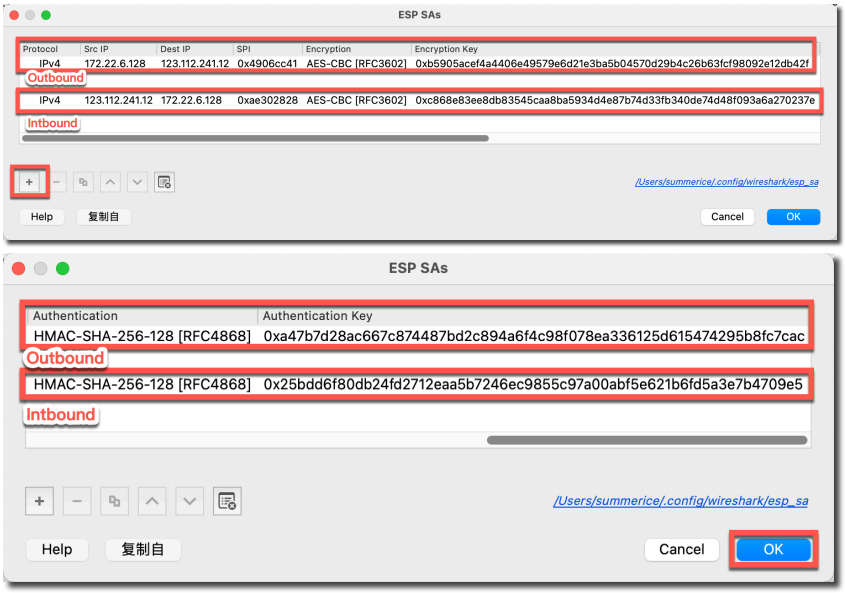
> **重要**
>
> 也可以在 CLI 下使用 sniffer 抓取格式 6 的 ESP 报文，再转换为 Wireshark 可以打开的格式，详见如下章节：故障排查 → Sniffer 工具 → 使用 wireshark 打开抓包文件 ↗。



6. 执行两个 IPSec 站点之间的流量访问后，停止抓包后下载抓包文件，使用 Wireshark 打开下载的抓包文件，右键点击任意一个 ESP 报文，依次选择“协议首选项 → Encapsulating Security Payload → ESP SAs...”。

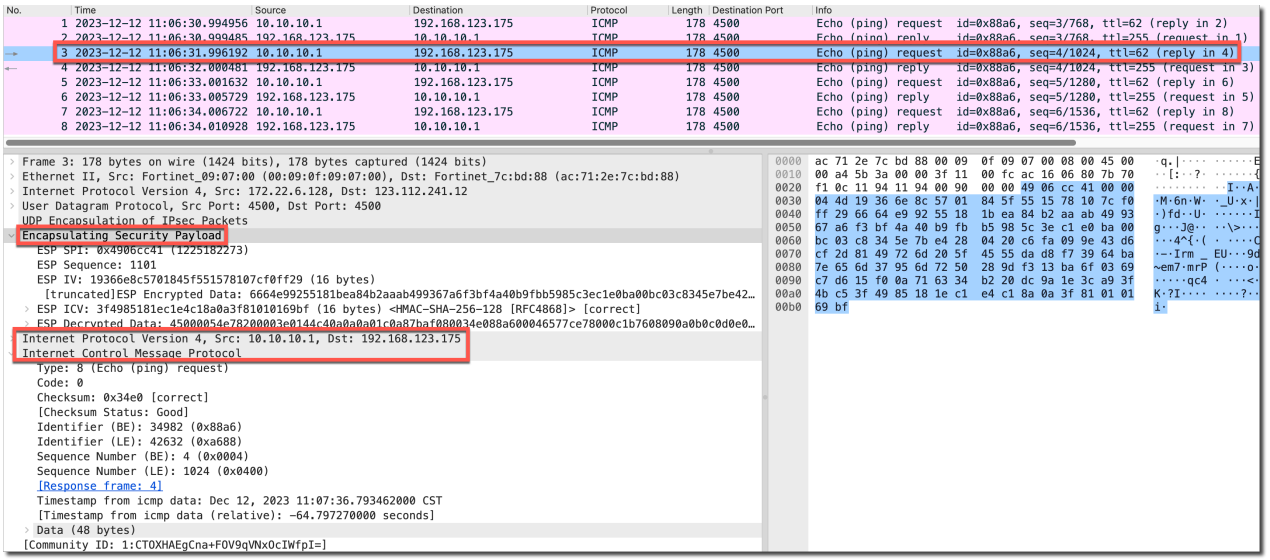


7. 点击左下角的✚，分别新建 outbound 方向和 Inbound 方向的 SA，根据步骤 2 或 3 收集到的网关地址、SPI、算法及密钥信息，填入相应的配置中，填写完成后，点击 OK 按钮。



- 所有的 SPI、加密密钥、验证密钥填入时前边都要加入字符 0x，如 outbound 方向的 SPI 应填写为 0x4906cc41。
- 加密算法和验证算法根据步骤 2 或 3 收集到的算法选择，这里选择为 AES - CBC [RFC3602] 和 HMAC-SHA-256-128 [RFC4868]。

8. 此时即可以看到 ESP 中加密的明文内容。

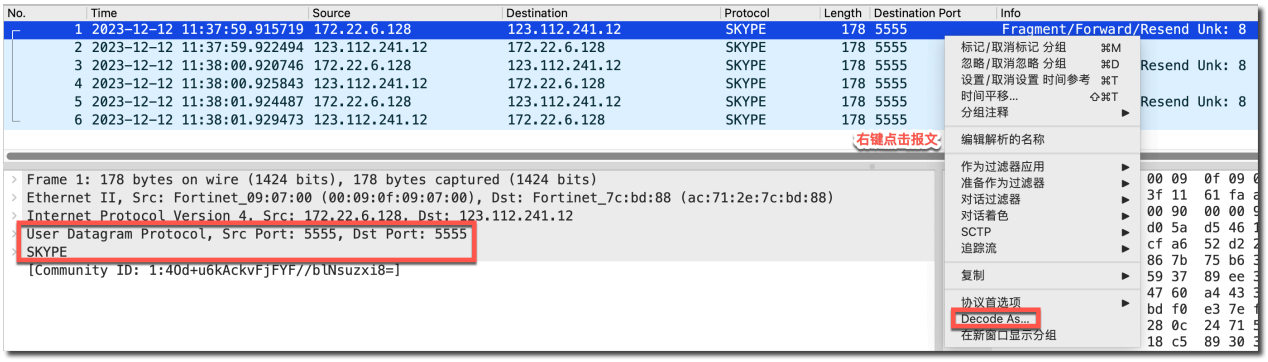


注意事项

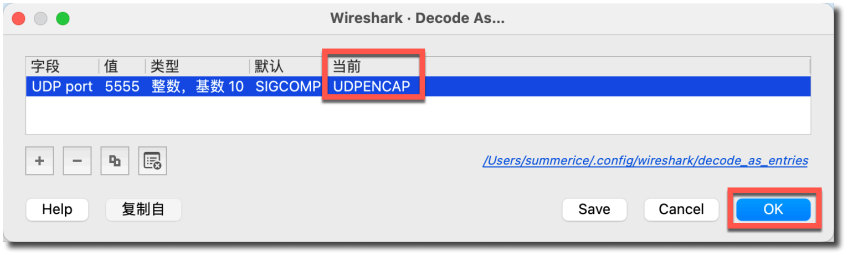
1. 如果在 `config system settings` 下修改了 ike-port，且存在 NAT-T 环境，则抓取 ESP 报文时过滤 UDP 端口为设置的 ike-port 端口号（详见 VPN 技术 → IPSec VPN → 点到点 VPN → 修改 IPSec 协商端口 ↗章节），如下所示，将两端 FortiGate 的 ike-port 均修改为 5555。

```bash
config system settings
    set ike-port 5555
end
```

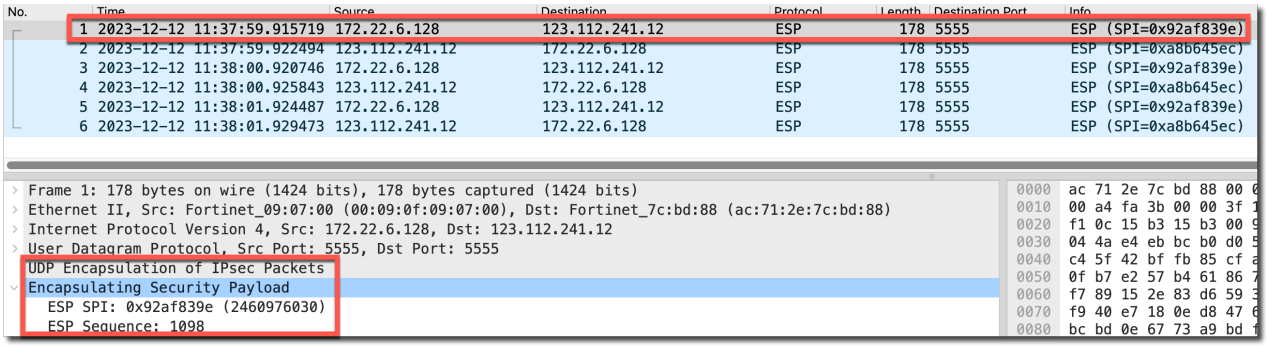
2. ESP 报文抓取后并使用 Wireshark 打开，由于 ESP 报文被 UDP 5555 封装，Wireshark 无法直接解析出这是 ESP 报文，而是识别为其他协议。这时我们需要右键点击抓到的 UDP 5555 报文，选择“Decode As...”按钮。



3. 在弹出的“Decode As...”窗口中，选择 UDP 5555 的当前列为“UDPENCAP”，即 UDP 封装的 ESP。



4. 再次查看抓到的 ESP 包，Wireshark 可以解析到 ESP 包头，随后继续使用本章节介绍的 ESP 解密方法进行解密即可。

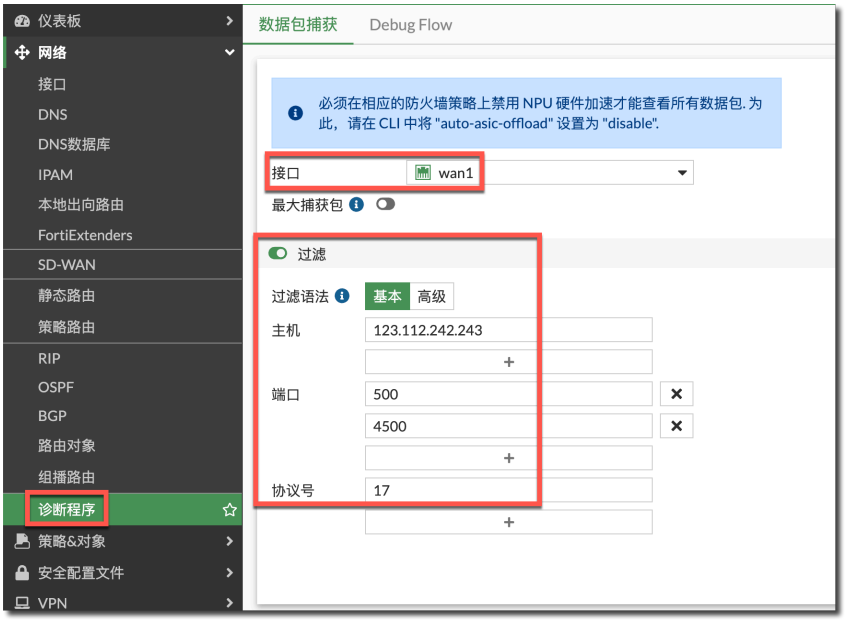


IKE 解密步骤

IKEv1

1. 抓取 IKE 报文不需要关闭 npu-offload。首先在 IKE 协商前开启抓包，这里我们以有 NAT-T + IKEv1 主模式的环境为例，抓取 UDP 500 或 UDP 4500 的数据包。如果没有 NAT-T，则只需抓取 UDP 500 的数据包（GUI 方式抓包方法详见故障排查 → Sniffer 工具 → 通过 GUI 界面抓包 ↗章节）。

> **重要**
>
> 也可以在 CLI 下使用 sniffer 抓取格式 6 的 IKE 报文，再转换为 Wireshark 可以打开的格式，详见如下章节：故障排查 → Sniffer 工具 → 使用 wireshark 打开抓包文件 ↗。



2. 发起 IKE 协商或重协商，并保存抓取的 IKE 协商报文（ISAKMP）。

3. 协商完成后，在 FortiGate 上执行 `get vpn ike gateway <tunnel_name>` 获取 IKE 密钥信息，需要获取 2 个如下信息。

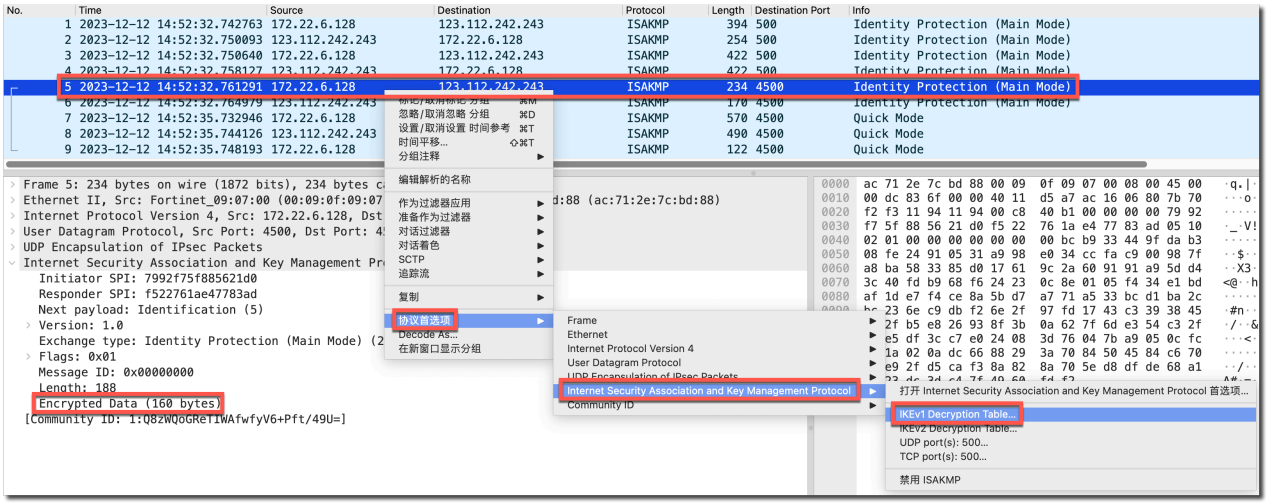
```
FW4_FGT61F_Left # get vpn ike gateway to_Home_WAN1

vd: root/0
name: to_Home_WAN1
version: 1
interface: wan1 5
addr: 172.22.6.128:4500 -> 123.112.242.243:4500
created: 478s ago
peer-id: 192.168.123.175
peer-auth: no
IKE SA  created: 1/1  established: 1/1  time: 20/20/20 ms
IPsec SA  created: 1/1  established: 1/1  time: 10/10/10 ms

  id/spi: 291 7992f75f885621d0/f522761ae47783ad
  direction: initiator
  status: established 478-478s ago = 20ms
  proposal: aes-256-sha256
  key: 54e29ac67129db6d-96c7e03b20b877f2-bdb4b15288826e8e-13cb202b0bb2cccf
  lifetime/rekey: 86400/85621
  DPD sent/recv: 0000440c/00000000
```

   - 发起方的 SPI（Cookie）：这个值从“id/spi”项的值中“/”前边的 16 进制字符获取，此例中为 7992f75f885621d0（发起方的 Cookie 永远在“/”符号之前）。
   - key：54e29ac67129db6d96c7e03b20b877f2bdb4b15288826e8e13cb202b0bb2cccf（需要将中间的“-”字符删除）。

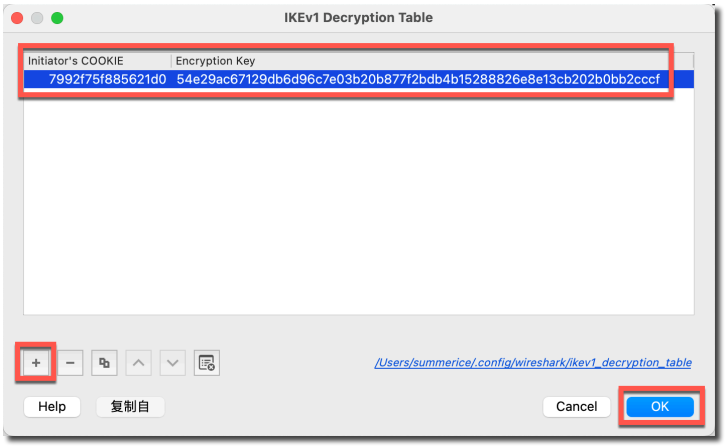
4. 以上解密材料也可以通过 `diagnose debug application ike -1` 的协商过程输出结果中找到（如果 IKE 协商过程中失败，或协商成功后就中断，来不及使用步骤 3 的命令查看密钥信息，可以使用此方法代替）。

```bash
FortiGate # diagnose debug application ike -1
FortiGate # diagnose debug enable
......
ike 0:to_Home_WAN1:291: ISAKMP SA 7992f75f885621d0/f522761ae47783ad key
32:54E29AC67129DB6D96C7E03B20B877F2BDB4B15288826E8E13CB202B0BB2CCCF
......
```

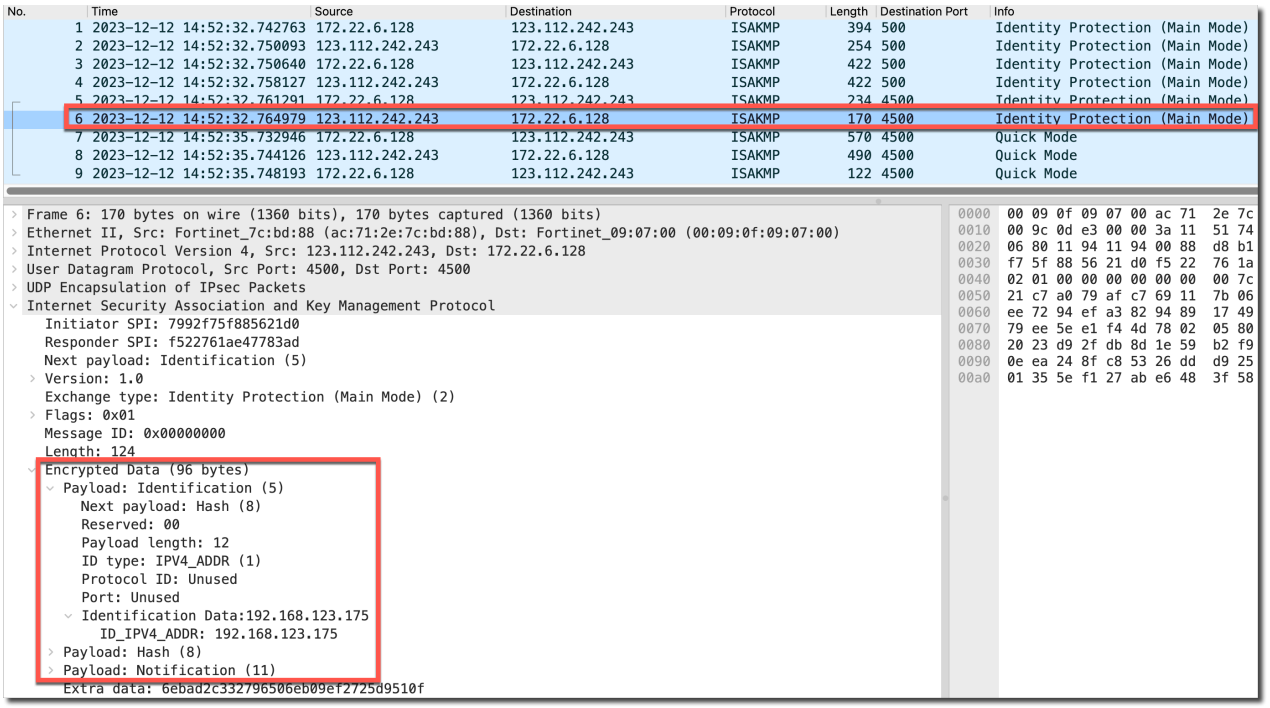
5. 使用 Wireshark 打开步骤 2 中抓取的 IKE 协商报文（ISAKMP）。可以看到 IKEv1 主模式从第 5 个报文开始加密。右键点击任意一个 ISAKMP 报文，依次选择“协议首选项 → Internet Security Association and Key Management Protocol → IKEv1 Decryption Table...”。



6. 在弹出的“IKEv1 Decryption Table”窗口中，点击左下角的✚ 按钮，在“Initiator's COOKIE”中填写步骤 3 或 4 获取的发起方的 SPI（Cookie）：7992f75f885621d0；在“Encryption Key”中填写步骤 3 或 4 获取的 key 54e29ac67129db6d96c7e03b20b877f2bdb4b15288826e8e13cb202b0bb2cccf。随后点击 OK 按钮。



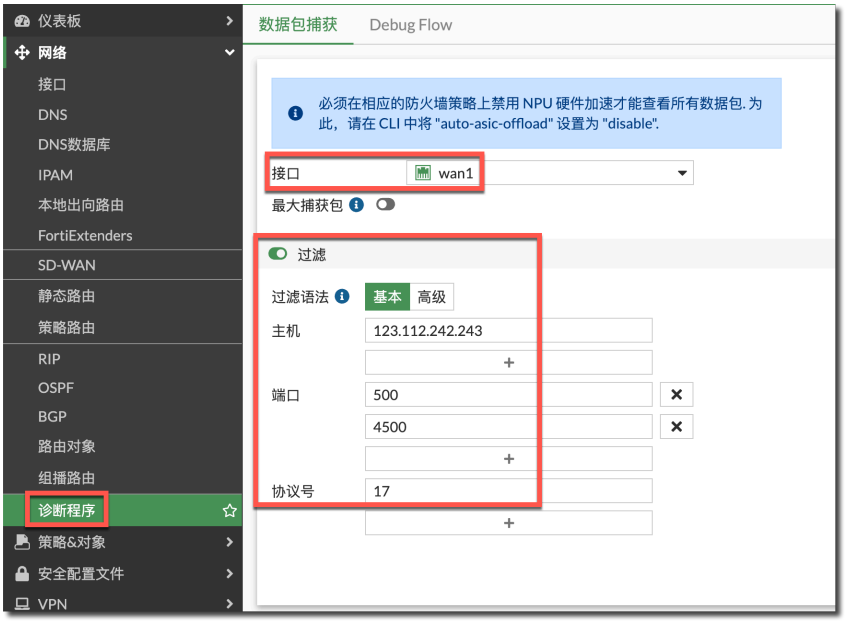
7. 此时即可以看到 IKEv1 协商报文中加密的明文内容。



IKEv2

1. 抓取 IKE 报文不需要关闭 npu-offload。首先在 IKE 协商前开启抓包，这里我们以有 NAT-T + IKEv2 的环境为例，抓取 UDP 500 或 UDP 4500 的数据包。如果没有 NAT-T，则只需抓取 UDP 500 的数据包（GUI 方式抓包方法详见故障排查 → Sniffer 工具 → 通过 GUI 界面抓包 ↗章节）。

> **重要**
>
> 也可以在 CLI 下使用 sniffer 抓取格式 6 的 IKE 报文，再转换为 Wireshark 可以打开的格式，详见如下章节：故障排查 → Sniffer 工具 → 使用 wireshark 打开抓包文件 ↗。



2. 发起 IKE 协商或重协商，并保存抓取的 IKE 协商报文（ISAKMP）。

3. 协商完成后，在 FortiGate 上执行 `get vpn ike gateway <Tunnel_name>` 获取 IKE 密钥信息，需要获取 8 个如下信息。

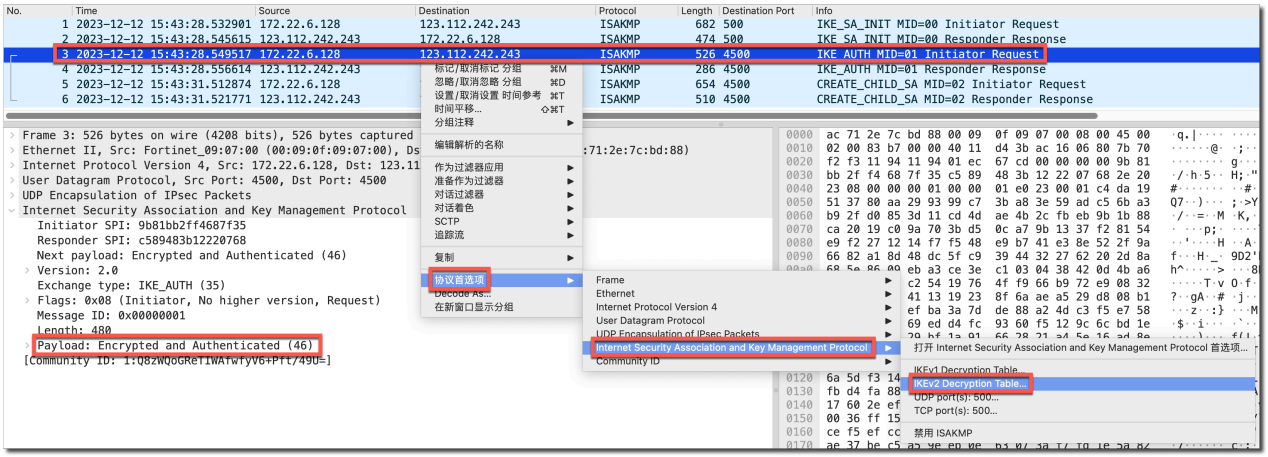
```
FW4_FGT61F_Left # get vpn ike gateway to_Home_WAN1

vd: root/0
name: to_Home_WAN1
version: 2
interface: wan1 5
addr: 172.22.6.128:4500 -> 123.112.242.243:4500
created: 186s ago
peer-id: 192.168.123.175
peer-auth: no
IKE SA  created: 1/1  established: 1/1  time: 20/20/20 ms
IPsec SA  created: 2/2  established: 2/2  time: 0/10/20 ms

  id/spi: 297 9b81bb2ff4687f35/c589483b12220768
  direction: initiator
  status: established 186-186s ago = 20ms
  proposal: aes-128-sha256
  SK_ei: 44655c0119f850aa-413ac8a3382e9cf1
  SK_er: 85b64b08dd028d1d-3142238395fcdb00
  SK_ai: b416e2d3d84af464-966a40d04798a805-eac288448d2940d2-f56f42ef13c5c7af
  SK_ar: cb2308de4db789ea-2ced21363bf1cc3f-d3fa379a1391a7f4-30376dd12c06ca87
  lifetime/rekey: 86400/85913
  DPD sent/recv: 00004425/00004425
```

- 发起方的 SPI（Cookie）：这个值从“id/spi”项的值中“/”前边的 16 进制字符获取，此例中为 9b81bb2ff4687f35（发起方的 Cookie 永远在“/”符号之前）。
- 响应方的 SPI（Cookie）：这个值从“id/spi”项的值中“/”后边的 16 进制字符获取，此例中为 c589483b12220768（响应方的 Cookie 永远在“/”符号之后）。
- 加密算法：从“proposal”项中获取，此例为 AES-128。
- 验证算法：从“proposal”项中获取，此例为 SHA-256。
- SK_ei：44655c0119f850aa413ac8a3382e9cf1（需要将中间的“-”字符删除）。
- SK_er：85b64b08dd028d1d-3142238395fcdb00（需要将中间的“-”字符删除）。
- SK_ai：b416e2d3d84af464966a40d04798a805eac288448d2940d2f56f42ef13c5c7af（需要将中间的“-”字符删除）。
- SK_ar：cb2308de4db789ea2ced21363bf1cc3fd3fa379a1391a7f430376dd12c06ca87（需要将中间的“-”字符删除）。

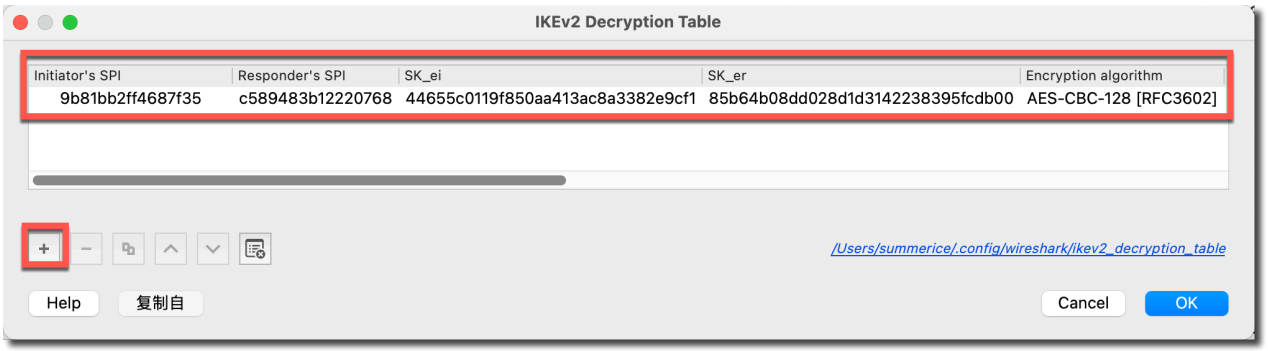
4. 以上解密材料也可以通过 `diagnose debug application ike -1` 的协商过程输出结果中找到（如果 IKE 协商过程中失败，或协商成功后就中断，来不及使用步骤 3 的命令查看密钥信息，可以使用此方法代替）。

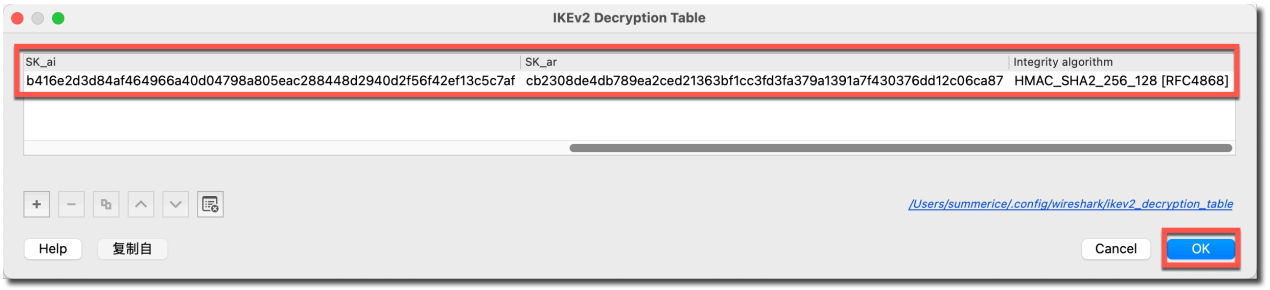
```bash
FortiGate # diagnose debug application ike -1
FortiGate # diagnose debug enable
......
ike 0: comes 123.112.242.243:500→172.22.6.128:500,ifindex=5,vrf=0....
ike 0: IKEv2 exchange=SA_INIT_RESPONSE id=9b81bb2ff4687f35/c589483b12220768 len=432
......
ike 0:to_Home_WAN1:297: matched proposal id 1
ike 0:to_Home_WAN1:297: proposal id = 1:
ike 0:to_Home_WAN1:297:   protocol = IKEv2:
ike 0:to_Home_WAN1:297:      encapsulation = IKEv2/none
ike 0:to_Home_WAN1:297:         type=ENCR, val=AES_CBC (key_len = 128)
ike 0:to_Home_WAN1:297:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike 0:to_Home_WAN1:297:         type=PRF, val=PRF_HMAC_SHA2_256
ike 0:to_Home_WAN1:297:         type=DH_GROUP, val=MODP2048.
ike 0:to_Home_WAN1:297: lifetime=86400
......
ike 0:to_Home_WAN1:297: IKE SA 9b81bb2ff4687f35/c589483b12220768 SK_ei 16:44655C0119F850AA413AC8A3382E9CF1
ike 0:to_Home_WAN1:297: IKE SA 9b81bb2ff4687f35/c589483b12220768 SK_er 16:85B64B08DD028D1D3142238395FCDB00
ike 0:to_Home_WAN1:297: IKE SA 9b81bb2ff4687f35/c589483b12220768 SK_ai
32:B416E2D3D84AF464966A40D04798A805EAC288448D2940D2F56F42EF13C5C7AF
ike 0:to_Home_WAN1:297: IKE SA 9b81bb2ff4687f35/c589483b12220768 SK_ar
32:CB2308DE4DB789EA2CED21363BF1CC3FD3FA379A1391A7F430376DD12C06CA87
......
```

5. 使用 Wireshark 打开步骤 2 中抓取的 IKE 协商报文（ISAKMP）。可以看到 IKEv2 从 IKE_AUTH Initiator Request 报文开始加密。右键点击任意一个 ISAKMP 报文，依次选择“协议首选项 → Internet Security Association and Key Management Protocol → IKEv2 Decryption Table...”。

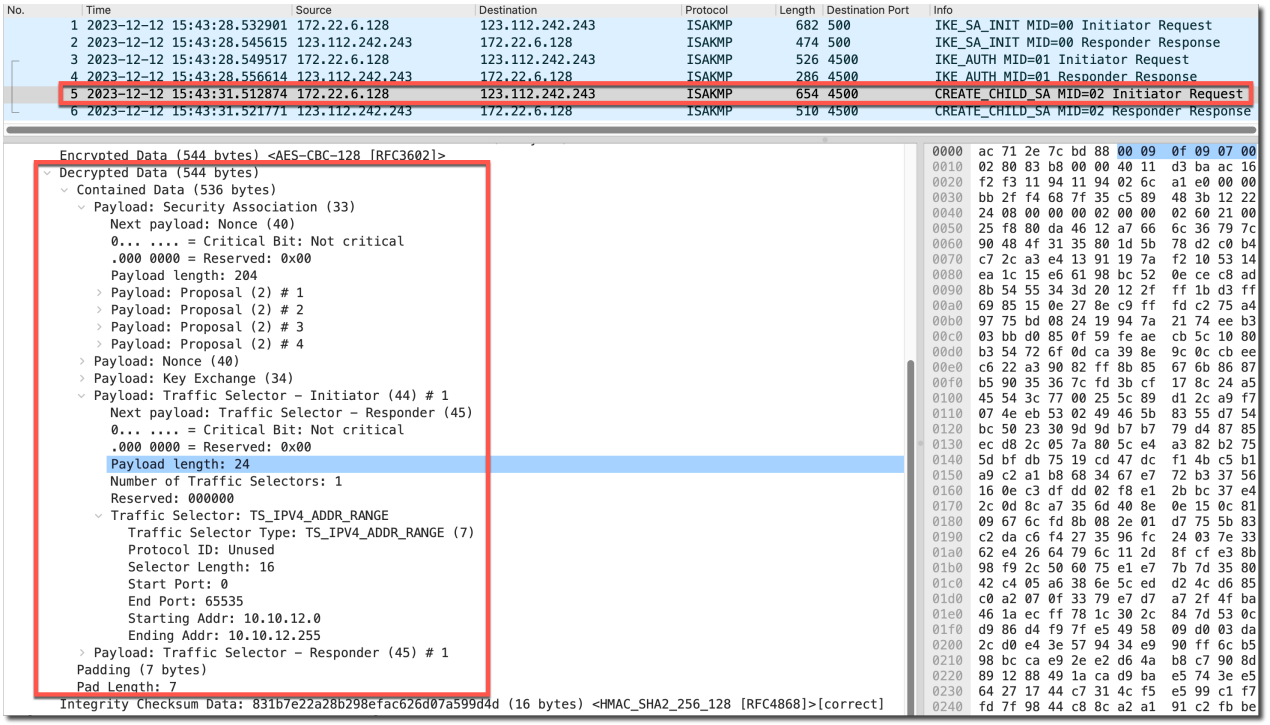


6. 在弹出的“IKEv2 Decryption Table”窗口中，点击左下角的✚ 按钮，依次填写步骤 3 或 4 获取的解密材料信息。随后点击 OK 按钮。





7. 此时即可以看到 IKEv2 协商报文中加密的明文内容。

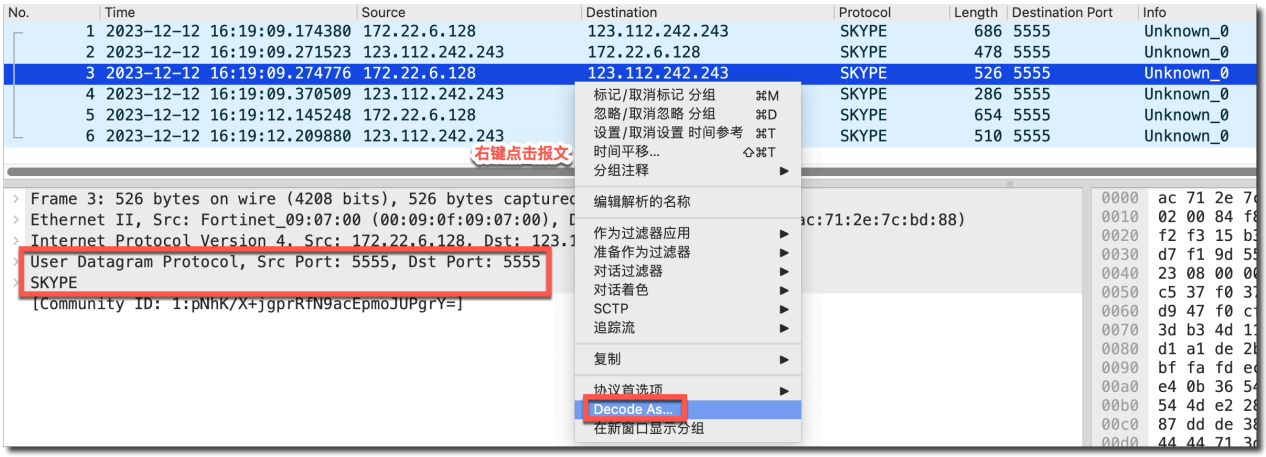


注意事项

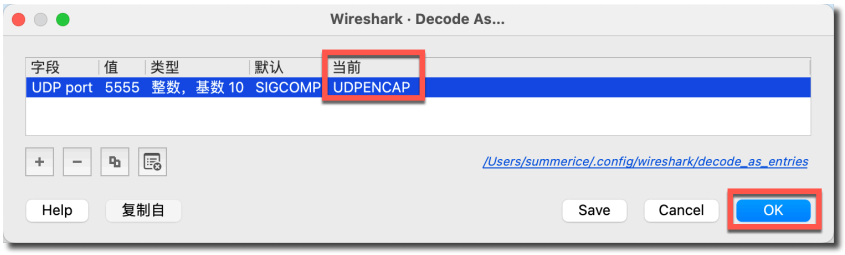
1. 如果在 `config system settings` 下修改了 ike-port，无论是否存在 NAT-T 环境，抓取 IKE 协商报文时要过滤 UDP 对应设置的 ike-port 端口号（详见 VPN 技术 → IPSec VPN → 点到点 VPN → 修改 IPSec 协商端口 ↗章节），如下所示，将两端 FortiGate 的 ike-port 均修改为 5555，所有的 IKE 协商包和 Information 报文会被 UDP 5555 封装。

```bash
config system settings
    set ike-port 5555
end
```

2. IKE 报文抓取后并使用 Wireshark 打开，由于 IKE 报文被 UDP 5555 封装，Wireshark 无法直接解析出这是 ISAKMP 报文，而是识别为其他协议。这时我们需要右键点击抓到的 UDP 5555 报文，选择“Decode As...”按钮。

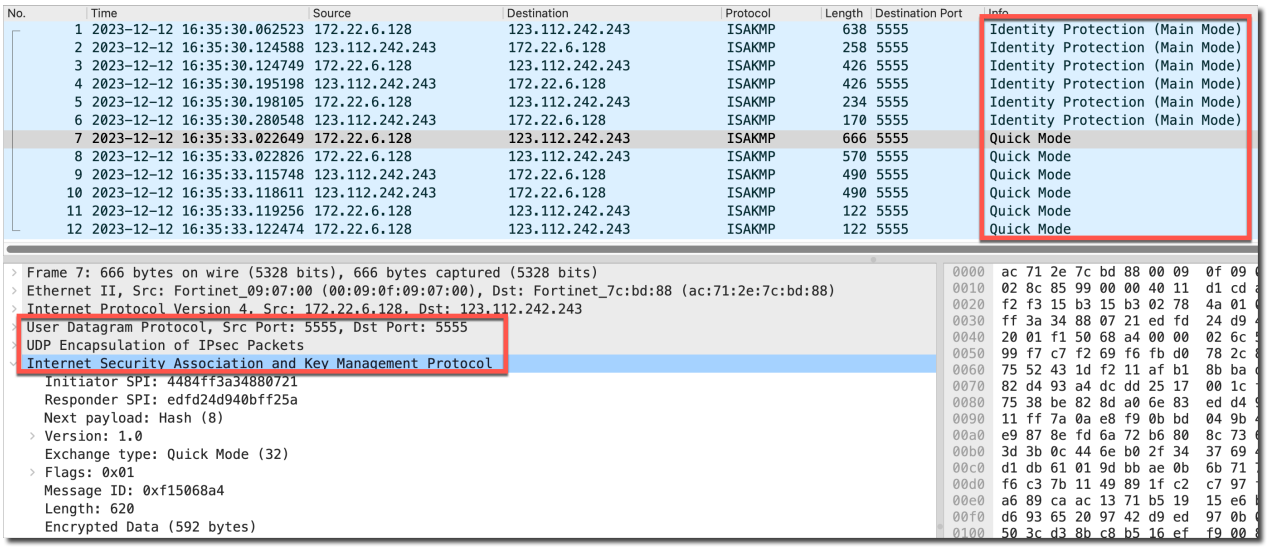


3. 在弹出的“Decode As...”窗口中，选择 UDP 5555 的“当前”列为“UDPENCAP”，然后点击 OK 按钮后（注意在修改了 ike-port 后，这里要选择 UDPENCAP，而不是 ISAKMP）。

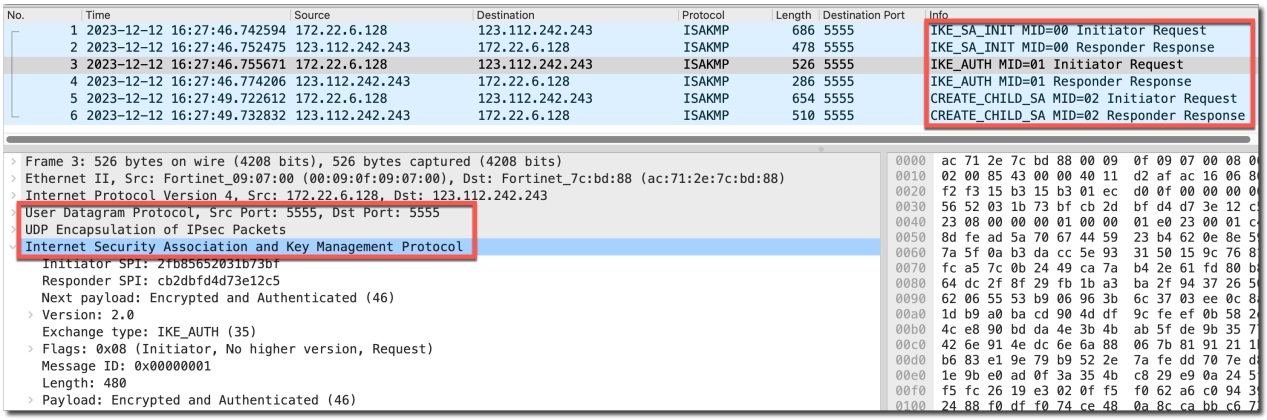


4. 可以看到 Wireshark 可以解析到 IKEv1 和 IKEv2 的 ISAKMP 协商报文。随后继续使用本章节介绍的 IKE 解密方法进行解密即可。

- IKEv1：



- IKEv2：


No. Time Source Destination Protocol Length Destination Port Info
1 2023-12-12 16:27:46.742594 172.22.6.128 123.112.242.243 ISAKMP 686 5555 IKE_SA_INIT MID=00 Initiator Request
2 2023-12-12 16:27:46.752475 123.112.242.243 172.22.6.128 ISAKMP 478 5555 IKE_SA_INIT MID=00 Responder Response
3 2023-12-12 16:27:46.755671 172.22.6.128 123.112.242.243 ISAKMP 526 5555 IKE_AUTH MID=01 Initiator Request
4 2023-12-12 16:27:46.774206 123.112.242.243 172.22.6.128 ISAKMP 286 5555 IKE_AUTH MID=01 Responder Response
5 2023-12-12 16:27:49.722612 172.22.6.128 123.112.242.243 ISAKMP 654 5555 CREATE_CHILD_SA MID=02 Initiator Request
6 2023-12-12 16:27:49.732832 123.112.242.243 172.22.6.128 ISAKMP 510 5555 CREATE_CHILD_SA MID=02 Responder Response
Frame 3: 526 bytes on wire (4208 bits), 526 bytes captured (4208 bits)
Ethernet II, Src: Fortinet_09:07:00 (00:09:0f:09:07:00), Dst: Fortinet_7c:bd:88 (ac:71:2e:7c:bd:88)
Internet Protocol Version 4, Src: 172.22.6.128, Dst: 123.112.242.243
User Datagram Protocol, Src Port: 5555, Dst Port: 5555
UDP Encapsulation of IPsec Packets
Internet Security Association and Key Management Protocol
Initiator SPI: 2fb85652031b73bf
Responder SPI: cb2dbfd4d73e12c5
Next payload: Encrypted and Authenticated (46)
Version: 2.0
Exchange type: IKE_AUTH (35)
Flags: 0x08 (Initiator, No higher version, Request)
Message ID: 0x00000001
Length: 480
Payload: Encrypted and Authenticated (46)
0000 ac 71 2e 7c bd 88 00 09 0f 09 07 00 08 0
0010 02 00 85 43 00 00 40 11 d2 af ac 16 06 8
0020 f2 f3 15 b3 15 b3 01 ec d0 0f 00 00 00 0
0030 56 52 03 1b 73 bf cb 2d bf d4 d7 3e 12 c
0040 23 08 00 00 00 01 00 00 01 e0 23 00 01 c
0050 8d fe ad 5a 70 67 44 59 23 b4 62 0e 8e 5
0060 7a 5f 0a b3 da cc 5e 93 31 50 15 9c 76 8
0070 fc a5 7c 0b 24 49 ca 7a b4 2e 61 fd 80 b
0080 64 dc 2f 8f 29 fb 1b a3 ba 2f 94 37 26 5
0090 62 06 55 53 b9 06 96 3b 6c 37 03 ee 0c 8
00a0 1d b9 a0 ba cd 90 4d df 9c fe ef 0b 58 2
00b0 4c e8 90 bd da 4e 3b 4b ab 5f de 9b 35 7
00c0 42 6e 91 4e dc 6e 6a 88 06 7b 81 91 21 1
00d0 b6 83 e1 9e 79 b9 52 2e 7a fe dd 70 7e d
00e0 1e 9b e0 ad 0f 3a 35 4b c8 29 e9 0a 24 5
00f0 f5 fc 26 19 e3 02 0f f5 f0 62 a6 c0 94 3
0100 24 88 f0 df f0 74 ce 48 0a 8c ca bb c6 7

SSL VPN

SSL VPN 的两种模式

开启 SSL VPN

SSL VPN FortiClient 隧道模式

SSL VPN 分配 IP 的方式

SSL VPN WEB 代理模式配置

SSL VPN 数字证书认证

SSL VPN 双因子认证

- FortiToken 双因子认证

SSL VPN 源地址连接控制

解锁 SSL VPN 用户

DTLS

SSL VPN 常见问题

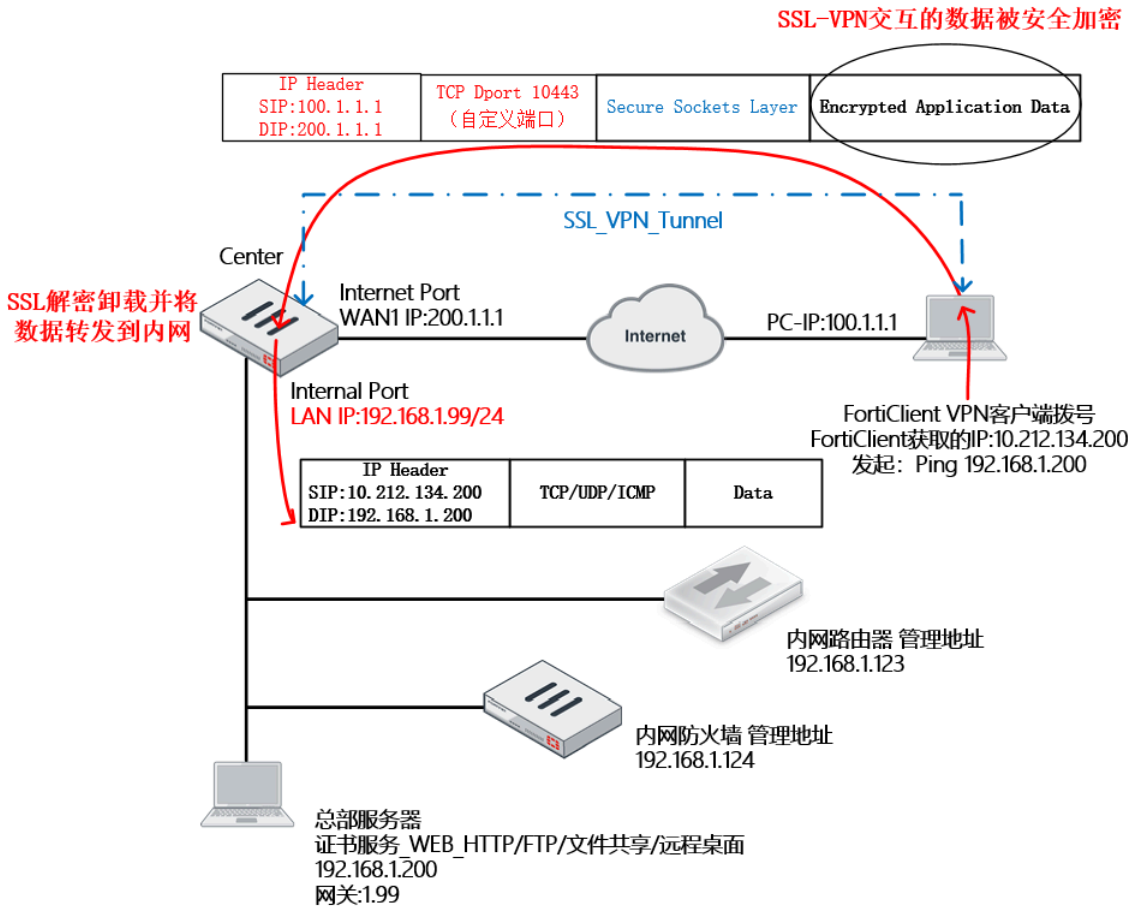
- SSL VPN 连接后断开

SSL VPN 的两种模式

> **重要**
>
> 由于部分用户 Web 服务器的网页代码不够规范，可能与 SSL VPN Web 代理模式不兼容，导致访问异常，强烈建议使用 SSL VPN 隧道模式。

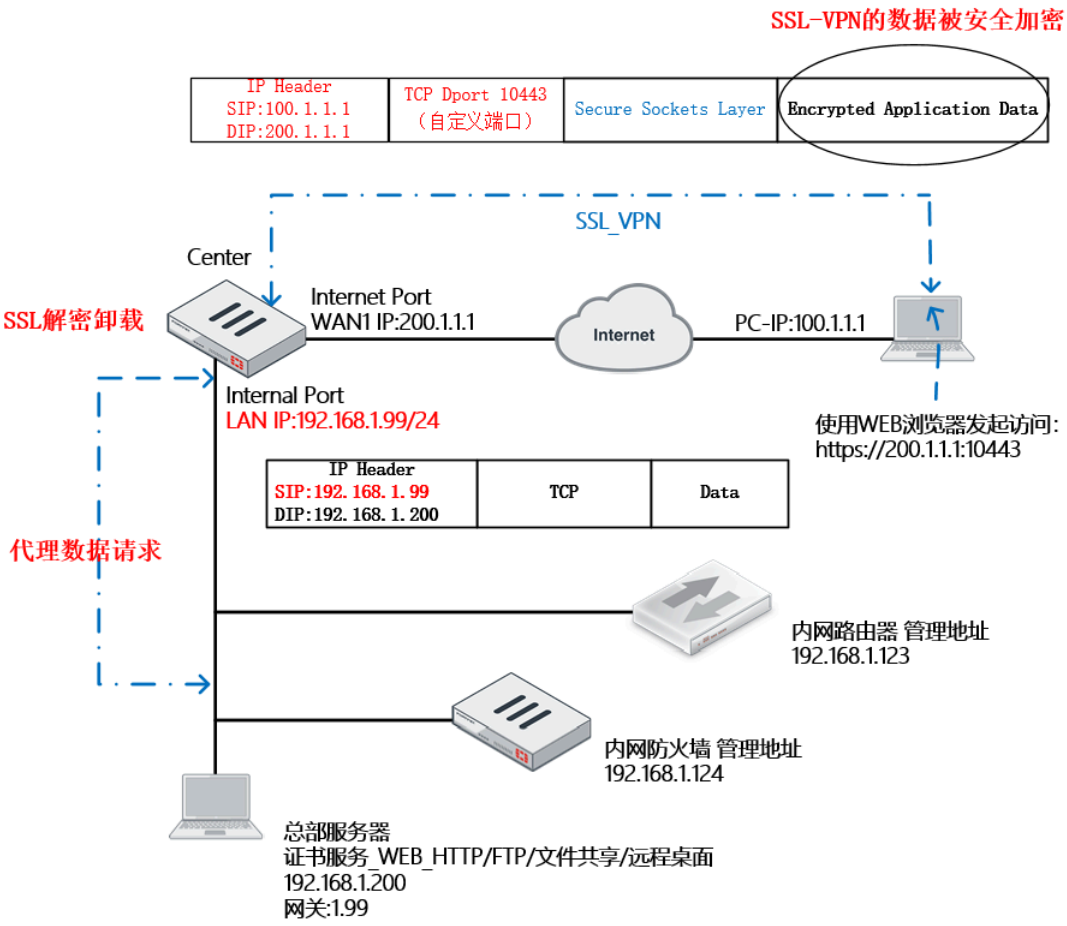
##SSL VPN 隧道模式

Tunnel 模式，需要下载运行的客户端支持。客户端和 FortiGate 设备建立 SSL 隧道后，FortiGate 为客户端分配 IP，客户端通过建立的虚接口直接通过 SSL 隧道连接到内部网络，推荐使用。



##WEB 代理模式

Web 代理模式，FortiGate 将来自远端浏览器的页面请求转发给 Web 服务器，然后将服务器的响应回传给终端用户。由于有时用户 Web 服务器的网页代码不够规范，可能与 SSLVPN 代理模式不兼容，会导致访问异常。


SSL-VPN的数据被安全加密
IP Header
SIP:100.1.1.1
DIP:200.1.1.1
TCP Dport 10443
（自定义端口）
Secure Sockets Layer
Encrypted Application Data
SSL_VPN
Center
SSL解密卸载
Internet Port
WAN1 IP:200.1.1.1
Internet
PC-IP:100.1.1.1
Internal Port
LAN IP:192.168.1.99/24
使用WEB浏览器发起访问:
https://200.1.1.1:10443
IP Header
SIP:192.168.1.99
DIP:192.168.1.200
TCP
Data
代理数据请求
内网路由器 管理地址
192.168.1.123
内网防火墙 管理地址
192.168.1.124
总部服务器
证书服务_WEB_HTTP/FTP/文件共享/远程桌面
192.168.1.200
网关:1.99

开启 SSL VPN

GUI 开启 SSL VPN

从 FortiOS 7.4.1 和 7.6.0 开始，默认配置下，“VPN → SSL - VPN”相关菜单在 GUI 界面中被隐藏（但仍可以通过 CLI 命令配置 SSL VPN 的相关功能）。

如果需要在 GUI 启用 SSL VPN 功能的可见性，需要在 CLI 下执行以下命令：

```bash
config system settings
    set gui-sslvpn enable    //隧道模式//
end
config system global
    set sslvpn-web-mode enable    //Web模式//
end
```

SSL VPN 移除情况

在以下版本的特定型号上完全移除了 SSL VPN 功能（**即使在 CLI 下也无法配置和使用**）。

IPSec 可以实现与 SSL VPN 基本相同的接入功能，且支持 TCP 封装，推荐使用 IPSec 进行远程接入（请参考 VPN 技术 → IPSec VPN → 拨号 VPN → 基于 TCP 的 FortiClient 拨号 ↗ 章节。

> **重要**
>
> 以下表格中未列出的版本/型号组合仍然可以正常使用 SSL VPN 功能。

移除 SSL VPN 的版本	移除 SSL VPN 的型号
7.4.8	所有 G 系列（如FortiGate-70G）
7.6.0	FortiGate-40F/40F-3G4G
	FortiWiFi-40F/40F-3G4G
	FortiGate-60F/61F
	FortiWiFi-60F/61F
	FortiGate-Rugged-60F/60F-3G4G
	FortiGate-80E/81E
	FortiGate-90E/91E
7.6.3(移除 Tunnel 模式，仅支持 Web 模式)	其他型号

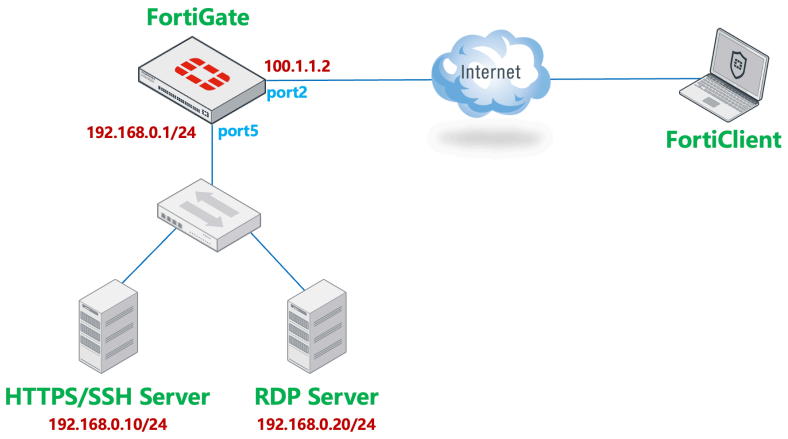
SSL VPN FortiClient 隧道模式

> **重要**
>
> 由于部分用户 Web 服务器的网页代码不够规范，可能与 SSL VPN Web 代理模式不兼容，导致访问异常，强烈建议使用隧道模式，而不是 Web 代理模式。

网络需求

在外移动办公的工作人员需要通过 SSL VPN 隧道模式拨入到公司内网来对内网主机进行访问。

网络拓扑

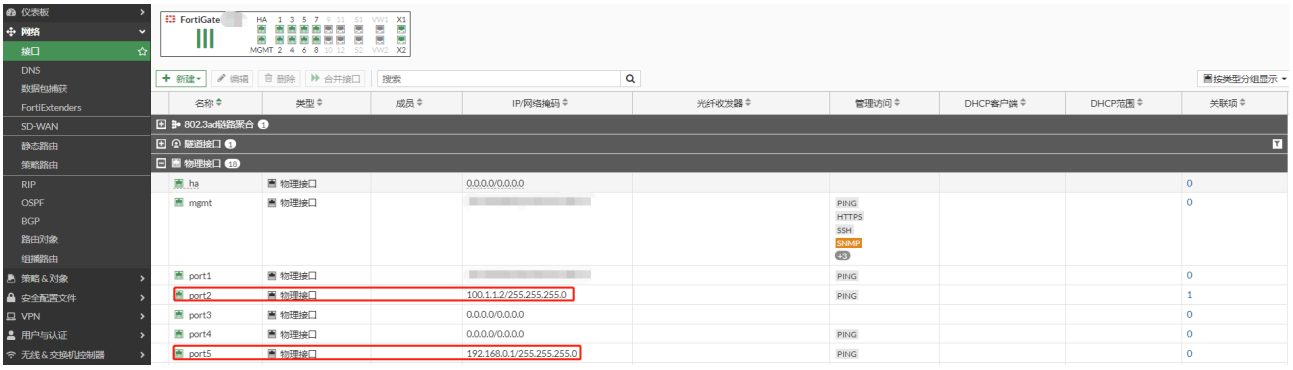


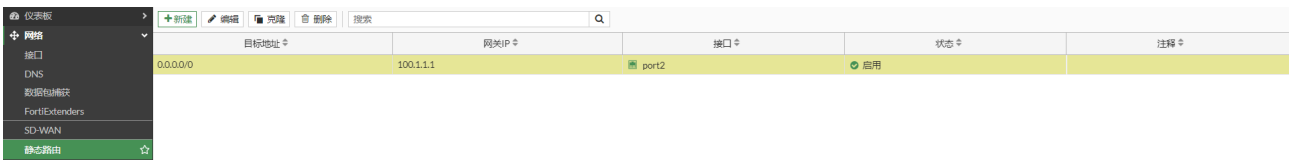
配置步骤

配置 SSL VPN

1.基本配置

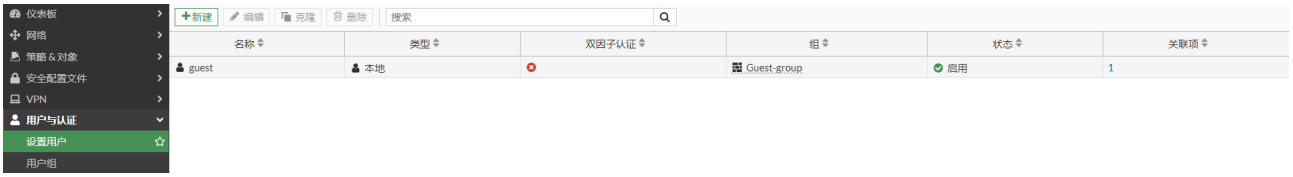
配置接口 IP 和路由





2.创建用户

选择“用户与认证”-->“设置用户”，点击“新建”。



选择“本地用户”，点击“下一步”。



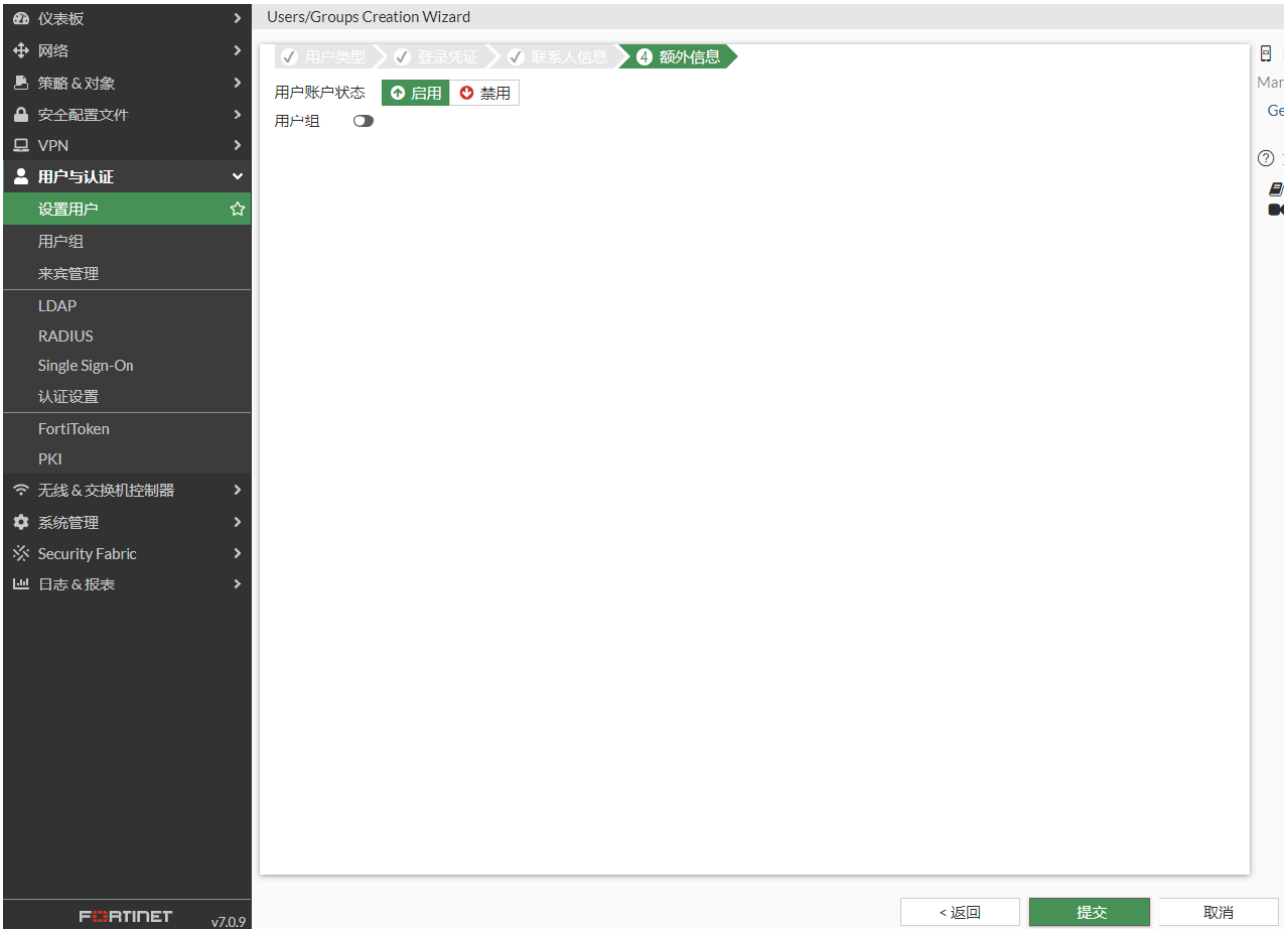
输入用户名和密码，点击“下一步”。



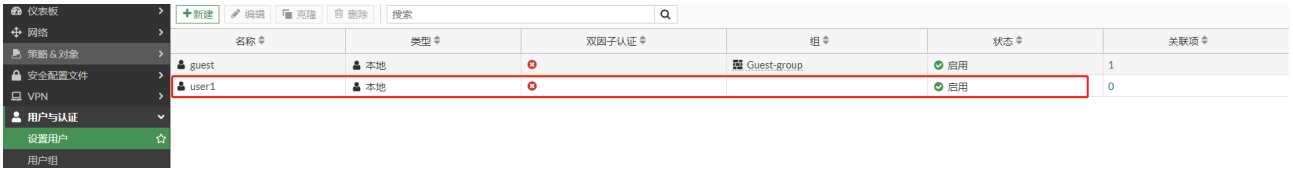
可根据需求选择启用。这里不启用，点击“下一步”。



点击“提交”。



完成用户创建。

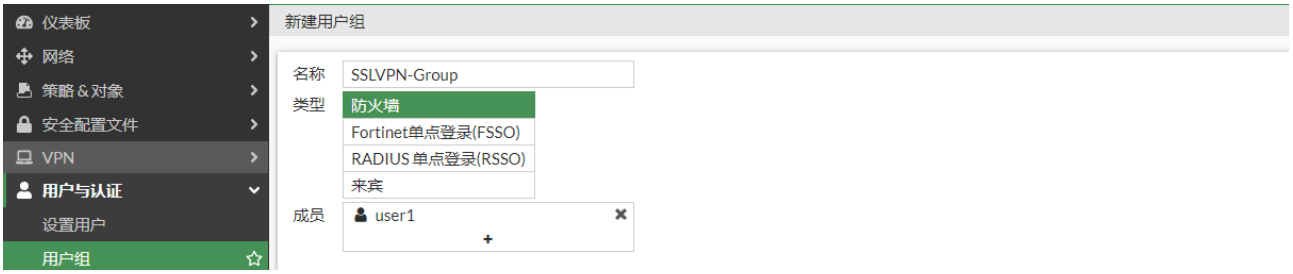


3.创建用户组

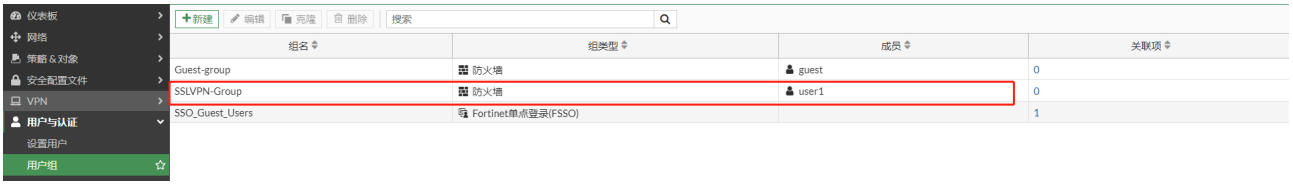
选择“用户与认证”-->“用户组”，点击“新建”。



输入名称，即组名，并将用户加入用户组，点击“确认”。

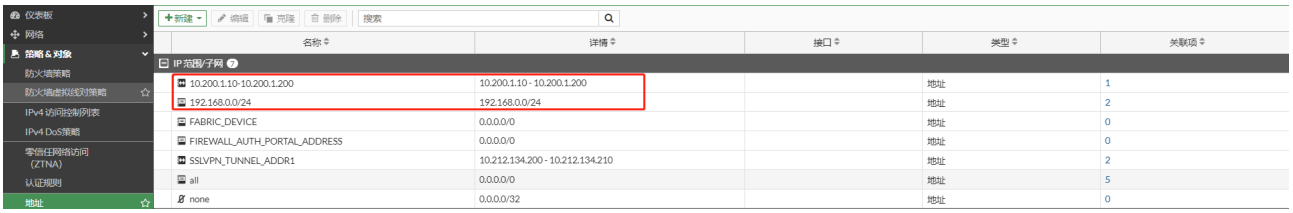


完成用户组创建。



4.配置地址对象

10.200.1.10-10.200.1.200 作为 SSLVPN 地址池；192.168.0.0/24 作为需要访问的内网地址段。



5.配置 SSL - VPN 门户

系统默认创建 3 个 SSLVPN 门户：full-access: 开启了隧道模式和 web 代理模式；tunnel-access 只开启了隧道模式；web-access：只开启 web 代理模式。可以根据实际需求进行修改，也可以新建新的 SSL - VPN 门户。这里使用 tunnel-access。



隧道分割的三个选项：

-**禁用**：所有客户端流量都将走 SSLVPN 隧道； -**基于策略目标启用**：只有在 SSL VPN 防火墙策略中的目标地址的流量走 SSLVPN；

```
::: important
在 SSL VPN 安全策略中配置的目标 IP 地址对象的相关路由将会下发到客户端的 VPN 路由表，不支持目标为 ISDB 条目。

另外，如果在下方配置了“隧道分割地址”并选择了地址对象，则会覆盖安全策略配置的目标地址作为 VPN 路由下发到客户端，安全策略将仅做流量控制使用。
:::
```

-**基于可信目标启用**：访问除 192.168.0.0/24 之外的流量走 SSLVPN。

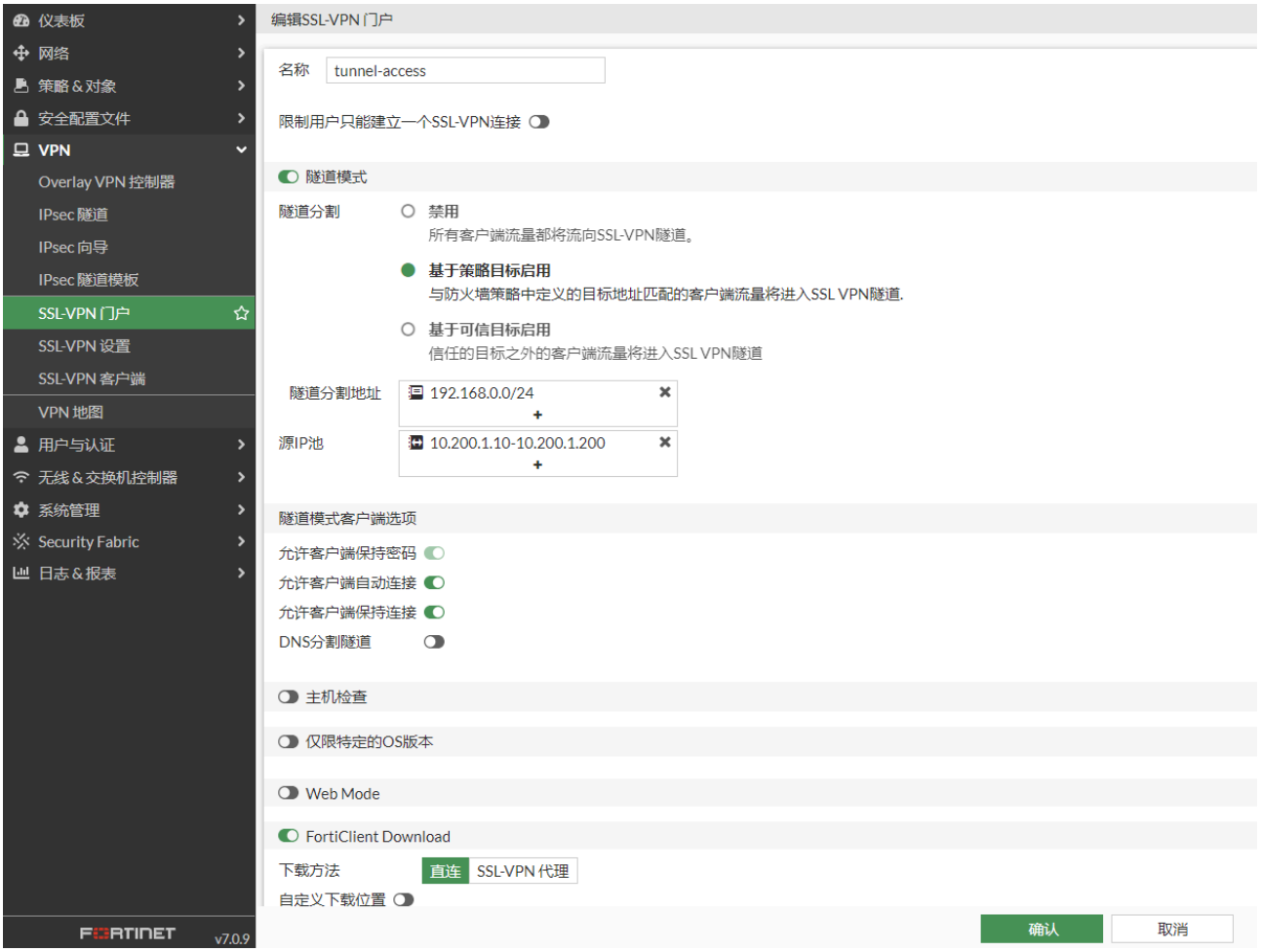
允许客户端保持密码：FortiClient 第一次拨号成功后，FortiClient 会显示“保存密码”的选项。

允许客户端自动连接：FortiClient 第一次拨号成功后，FortiClient 会显示“自动连接”的选项，该功能是运行 FortiClient 会拨入该 SSLVPN。

允许客户端保持连接：FortiClient 第一次拨号成功后，FortiClient 会显示“保持连接”的选项，该功能是 SSLVPN 由于网络原因中断后，会自动重拨。

免费的 FortiClient 版本不支持自动连接和保持连接，需要完整版的 FortiClient，需要购买 EMS license。

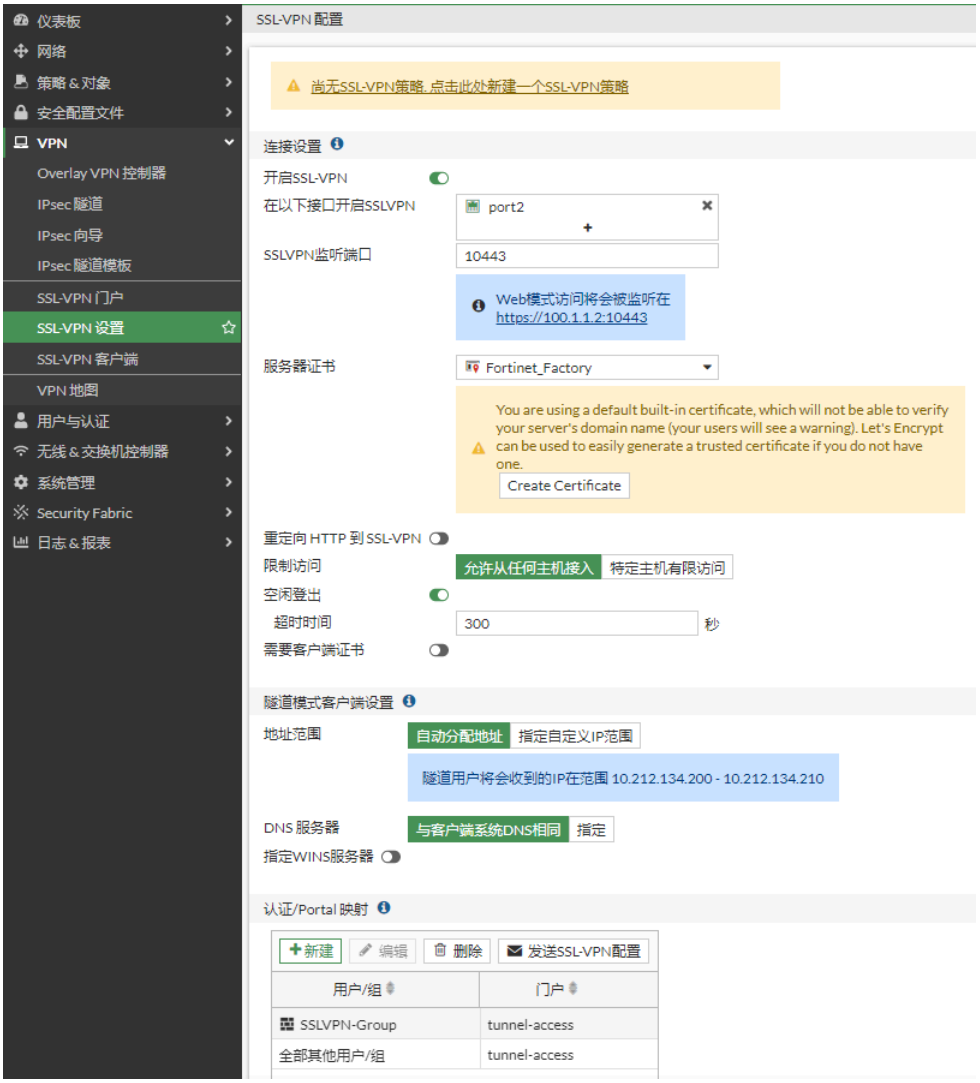
源 IP 池：客户端 SSLVPN 拨号成功后获取的地址。



6.配置 SSLVPN

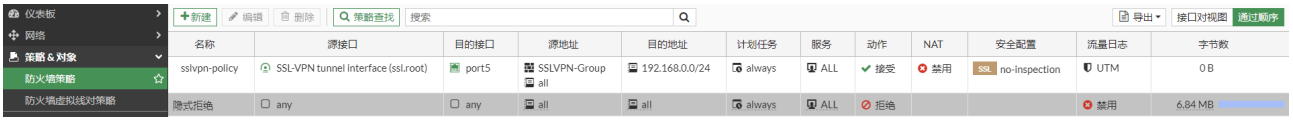
设置提供 SSLVPN 服务的接口和端口，将 SSLVPN-Group 用户组和 tunnel-access 门户关联；全部其他用户/组也要关联一个门户，用于没有配置关联的用户/用户组访问。

tunnel-access 门户中的 源 IP 池 优先于 SSL - VPN 设置中的 地址范围。



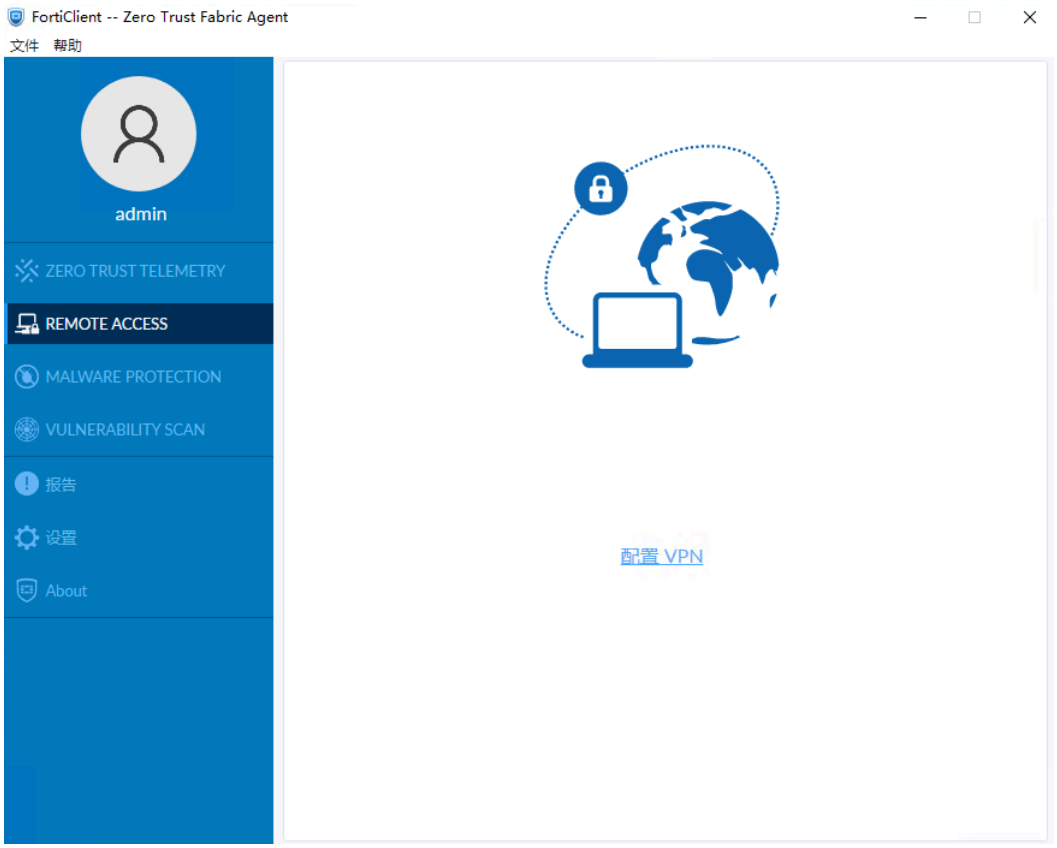
7.创建策略

当客户端 SSLVPN 拨号成功后，将会使用获取的地址（10.200.1.10-10.200.1.200）访问内部主机，因此内部网络需要增加到 10.200.1.0/24 网段的回程路由指向 FortiGate 或者 可以在策略中开启 NAT，那么源地址将被转换为 FortiGate 接口地址，则不用考虑回程路由。

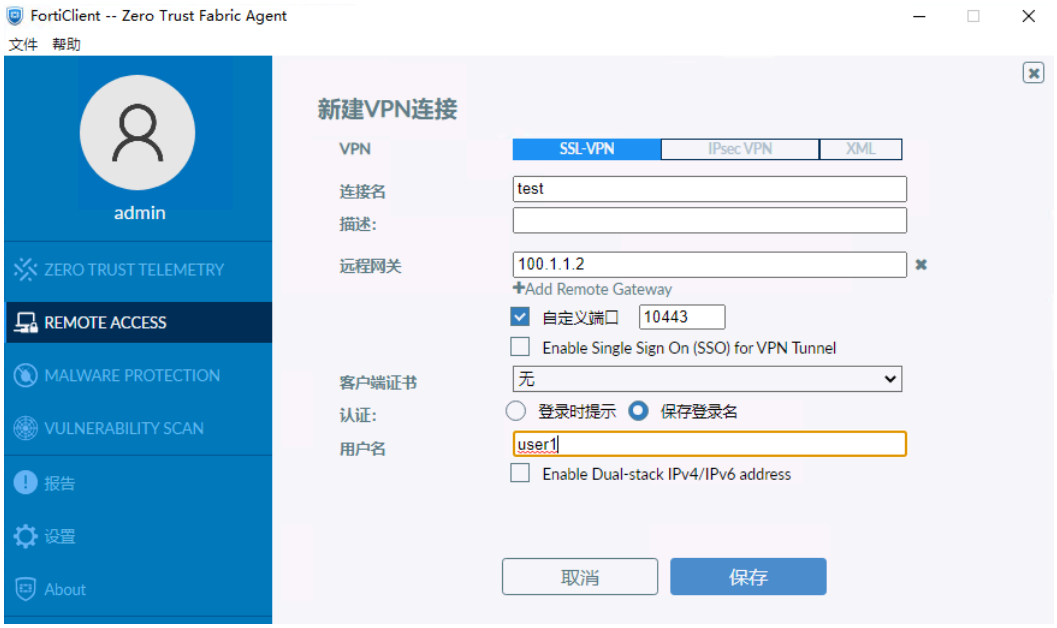


配置 FortiClient

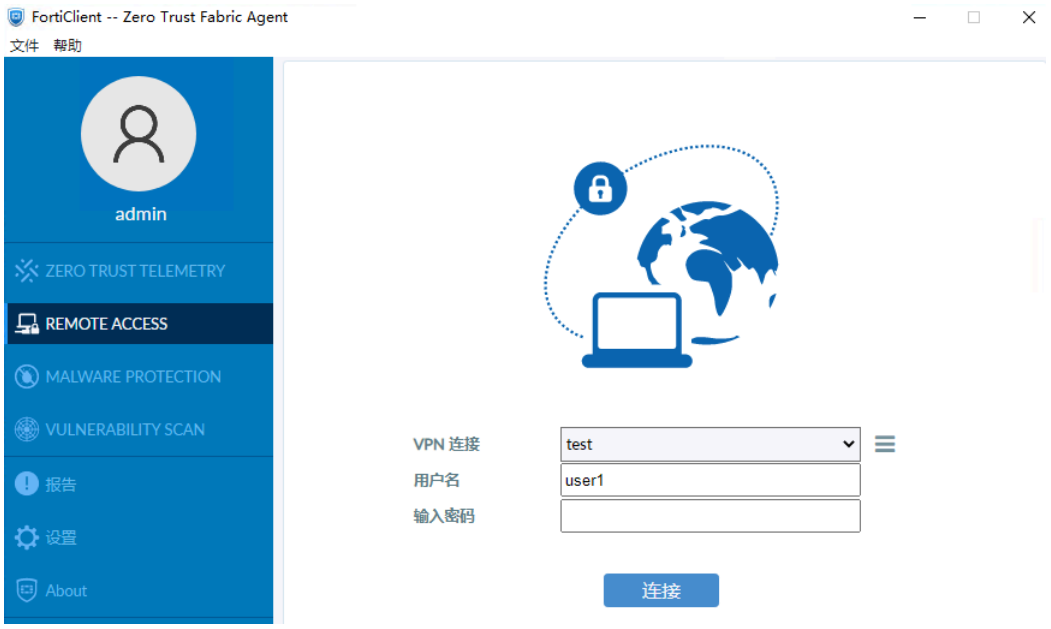
1. 选择“Remote Access”，点击“配置 VPN”。



2. 选择 SSL VPN，设置连接名，远程网关，SSLVPN 端口以及用户名，然后点击保存。



3. 配置完成。



业务测试

1. 在 FortiClient 中输入密码，并点击连接，在弹出的窗口中，点击“是”，信任 SSLVPN 证书



2. SSLVPN 拨号成功，获取的地址是 10.200.1.10



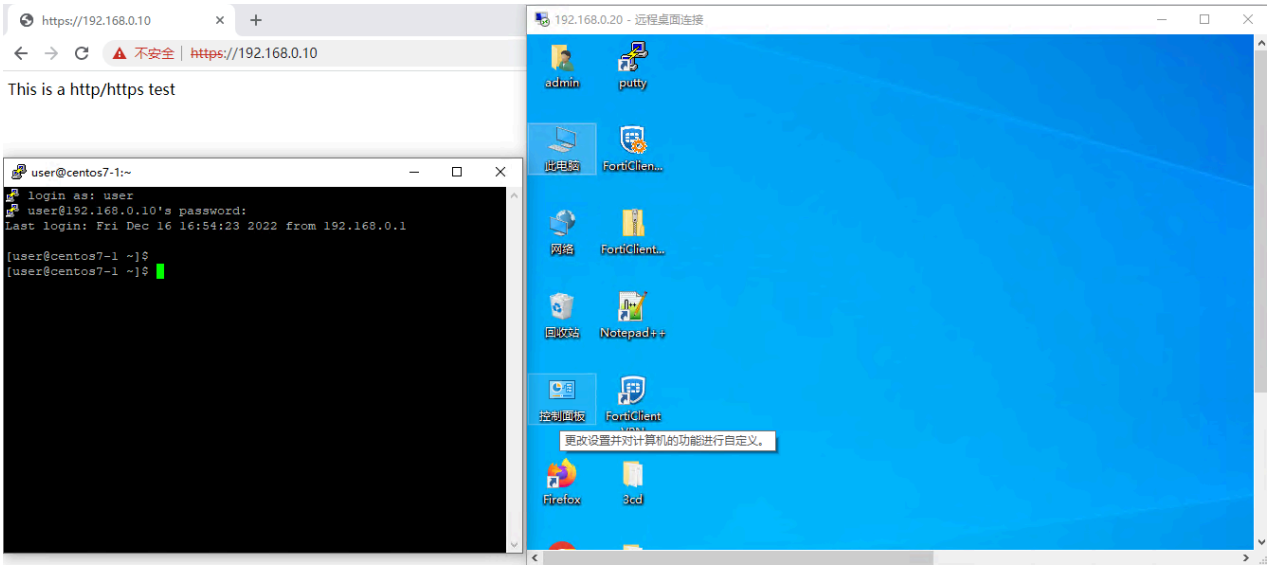
查看终端获取到的地址

```
C:\Users\admin>ipconfig/all
以太网适配器 以太网 2:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Fortinet SSL VPN Virtual Ethernet Adapter
   物理地址. . . . . . . . . . . . . : 00-09-0F-AA-00-01
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::9124:e3aa:1ace:74d2%7(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.200.1.10(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.255
   默认网关. . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 100665615
   DHCPv6 客户端 DUID . . . . . . . . : 00-01-00-01-29-1C-CC-4A-00-0C-29-A2-90-E0
   DNS 服务器 . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS . . . . . . . . : 已启用
```

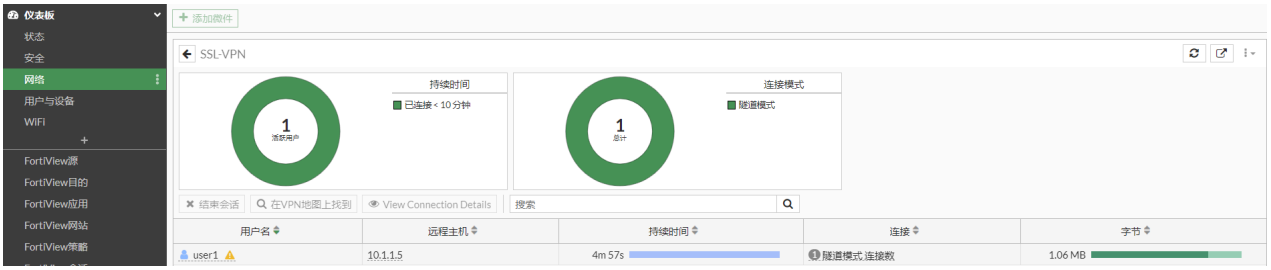
查看终端获取到的路由

```
C:\Users\admin>route print
===========================================================================
接口列表
  6...00 09 0f fe 00 01 ......Fortinet Virtual Ethernet Adapter (NDIS 6.30)
  7...00 09 0f aa 00 01 ......Fortinet SSL VPN Virtual Ethernet Adapter
 10...00 0c 29 a2 90 e0 ......Intel(R) 82574L Gigabit Network Connection
  5...00 0c 29 a2 90 ea ......Intel(R) 82574L Gigabit Network Connection #2
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0         10.1.1.1         10.1.1.5    281
         10.1.1.0    255.255.255.0            在链路上          10.1.1.5    281
         10.1.1.5  255.255.255.255            在链路上          10.1.1.5    281
       10.1.1.255  255.255.255.255            在链路上          10.1.1.5    281
      10.200.1.10  255.255.255.255            在链路上       10.200.1.10    257
        100.1.1.2  255.255.255.255         10.1.1.1         10.1.1.5     25
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
      192.168.0.0    255.255.255.0      10.200.1.11      10.200.1.10      1
```

3. 访问内网主机 HTTPS，SSH，RDP 服务都正常



4. FortiGate 查看 SSLVPN 连接




```
# get vpn ssl monitor
SSL-VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out    HTTPS in/out  Two-factor
Auth
 0       user1   SSLVPN-Group   1(1)       276             28434        10.1.1.5       0/0            0/0          0

SSL-VPN sessions:
 Index   User    Group          Source IP      Duration        I/O Bytes       Tunnel/Dest IP
 0       user1   SSLVPN-Group   10.1.1.5         366           236265/838218     10.200.1.10
```

SSL VPN 分配 IP 的方式

网络需求

如果在“SSL - VPN 配置”（`config vpn ssl settings`）中与“SSL - VPN 门户”（`config vpn ssl web portal`）中同时引用了两个不同的地址池，那么对应 Portal 用户的 FortiClient 拨号后获取的 IP 地址应该是“SSL - VPN 配置”还是“SSL - VPN 门户”引用的地址池中的呢?

SSL VPN Tunnel 模式分配 IP 的模式有两种：

- 默认配置下，FortiClient 拨号后获取的 IP 为“SSL - VPN 门户”引用的地址池中的 IP。且分配 IP 的方式为 `first-available`，即永远使用用户对应 Portal 引用的地址池中可用的最小地址来分配 IP。
- 如果想要使用轮询（`round-robin`）方式，当已经从地址池分配过一个 IP 后，即使该 IP 已经释放，也使用该 IP 后的可用 IP 进行分配（类似于 PPPoE 的 IP 分配方式）。使用此种方式时，IP 将从“SSL - VPN 配置”（`config vpn ssl settings`）的地址池中分配。

配置方法

```bash
FortiGate # config vpn ssl settings
FortiGate (settings) # set tunnel-addr-assigned-method ?
first-available    Assign the first available address from the pools.    //默认最小可用优先方式，使用config vpn ssl web portal中引用的地址池//
round-robin        Assign the available address from the pool with a round robin fashion.    //轮询方式，使用config vpn ssl settings中引用的地址池//
```

配置步骤

1. `config vpn ssl settings` 配置，引用地址池为 `sslvpn_pool`，用户组 `VPN` 对应的 Portal 为 `full-access`，使用默认选项 `first-available` 作为 IP 分配方式（相关用户、策略配置略）。

```bash
config vpn ssl settings
    set servercert "Fortinet_Factory"
    set tunnel-ip-pools "sslvpn_pool"
    set source-interface "wan1"
    set source-address "all"
    set source-address6 "all"
    set default-portal "web-access"
    config authentication-rule
        edit 2
            set groups "VPN"
            set portal "full-access"
        next
    end
    set tunnel-addr-assigned-method first-available    //默认配置//
end

config firewall address
    edit "sslvpn_pool"
        set type iprange
        set start-ip 3.3.3.1
        set end-ip 3.3.3.100
    next
end
```

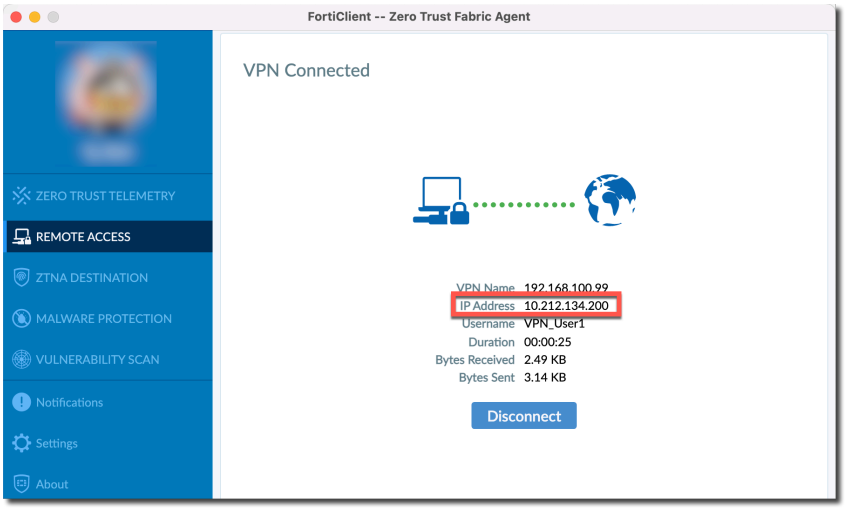
2. `config vpn ssl web portal` 配置，引用地址池为 `SSLVPN_TUNNEL_ADDR1`。

```bash
config vpn ssl web portal
    edit "full-access"
        set tunnel-mode enable
        set ip-pools "SSLVPN_TUNNEL_ADDR1"
    next
end

config firewall address
    edit "SSLVPN_TUNNEL_ADDR1"
```

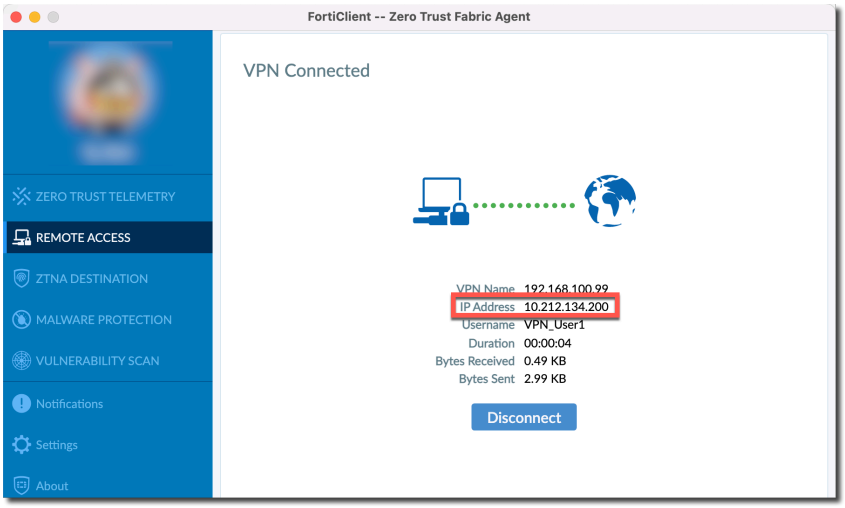
```
        set type iprange
        set start-ip 10.212.134.200
        set end-ip 10.212.134.210
    next
end
```

结果验证

1. FortiClient 使用用户组 VPN 中的用户连接 FortiGate 的 SSL VPN，连接成功后，获取的 IP 为 `config vpn ssl web portal` 引用的地址池 `SSLVPN_TUNNEL_ADDR1` 中的 IP 10.212.134.200。



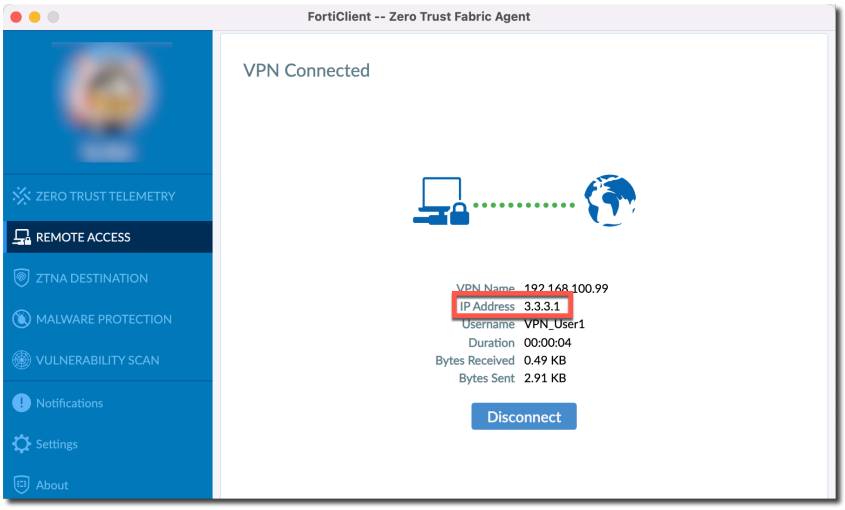
2. 断开 FortiClient 的 SSL VPN 连接，再次连接 FortiGate 的 SSL VPN，会获取到地址池 `SSLVPN_TUNNEL_ADDR1` 最小可用的地址 10.212.134.200，与步骤 1 中获取的地址一样（该地址释放后没有其他客户端获取）。



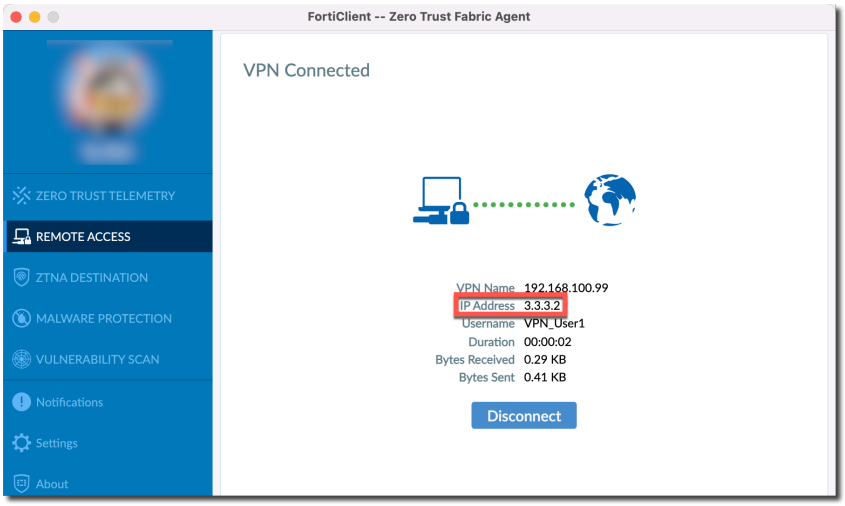
3. 在 `config vpn ssl settings` 中修改 IP 分配方式为 `round-robin`。

```bash
config vpn ssl settings
    set tunnel-addr-assigned-method round-robin
end
```

4. 再次使用 FortiClient 连接 FortiGate 的 SSL VPN，获取的 IP 为 `config vpn ssl settings` 引用的地址池 `sslvpn_pool` 中的 IP 3.3.3.1。



5. 断开 FortiClient 的 SSL VPN 连接，再次连接 FortiGate 的 SSL VPN，会获取到地址池 `sslvpn_pool` 的下一个可用 IP 3.3.3.2（即使上一个 IP 3.3.3.1 已经释放）。



SSL VPN WEB 代理模式配置

重要

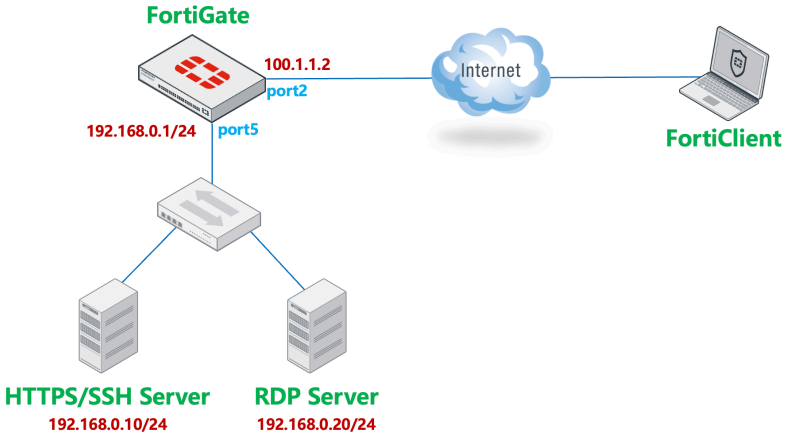
由于部分用户 Web 服务器的网页代码不够规范，可能与 SSL VPN Web 代理模式不兼容，导致访问异常，强烈建议使用 SSL VPN 隧道模式。

SSL VPN Web 代理模式目前不支持 HTML5 格式。

网络需求

在外移动办公的工作人员需要通过 SSL VPN WEB 模式拨入到公司内网来对内网主机进行访问。

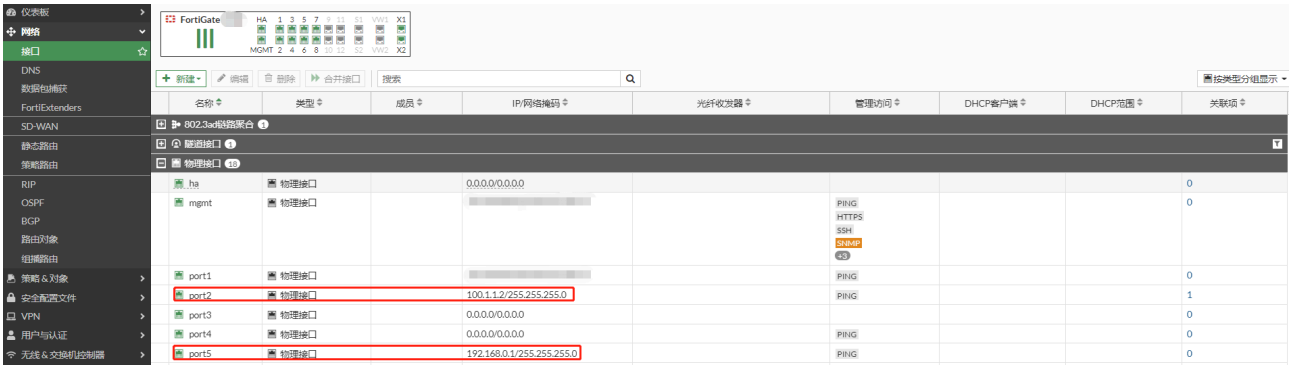
网络拓扑

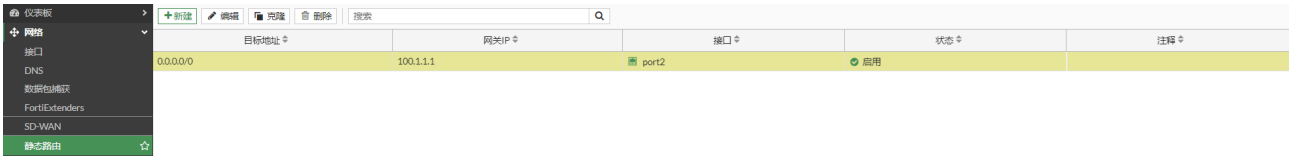


配置步骤

1.基本配置

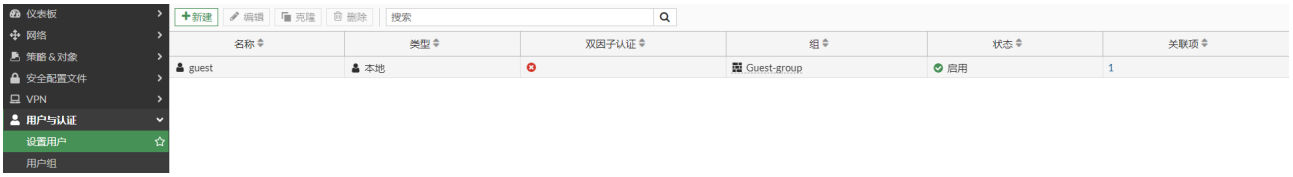
配置接口 IP 和路由





2.创建用户

选择“用户与认证”-->“设置用户”，点击“新建”。



选择“本地用户”，点击“下一步”。



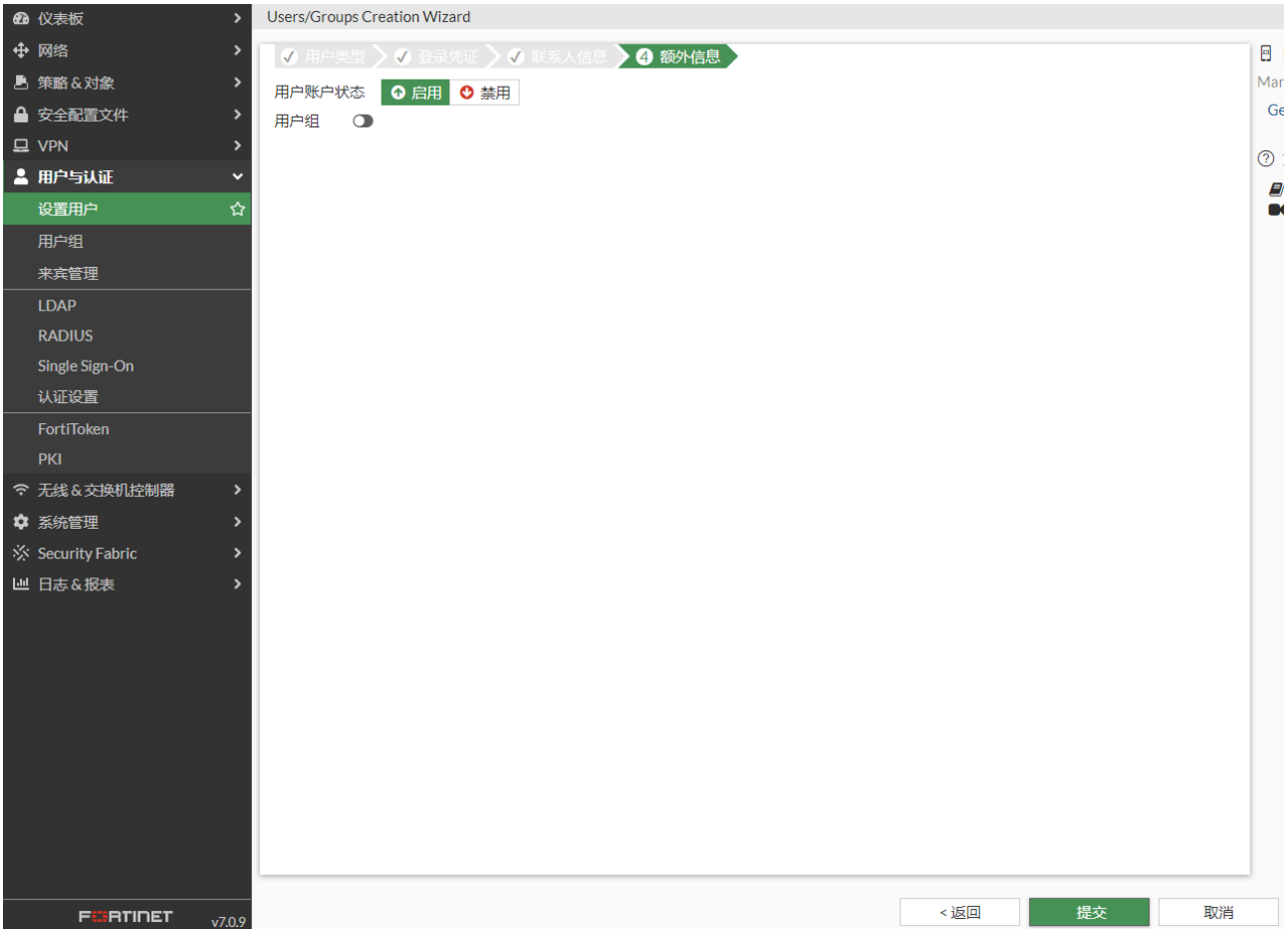
输入用户名和密码，点击“下一步”。



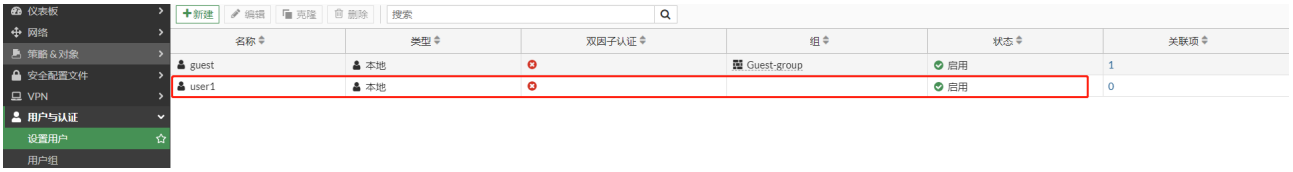
可根据需求选择启用。这里不启用，点击“下一步”。



点击“提交”。



完成用户创建。

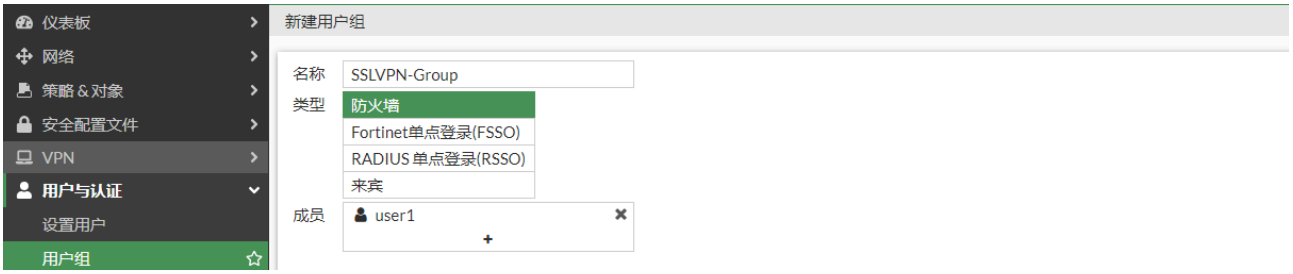


3.创建用户组

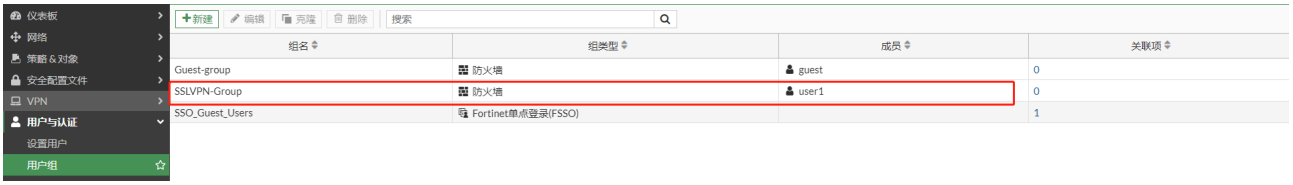
选择“用户与认证”-->“用户组”，点击“新建”。



输入名称，即组名，并将用户加入用户组，点击“确认”。

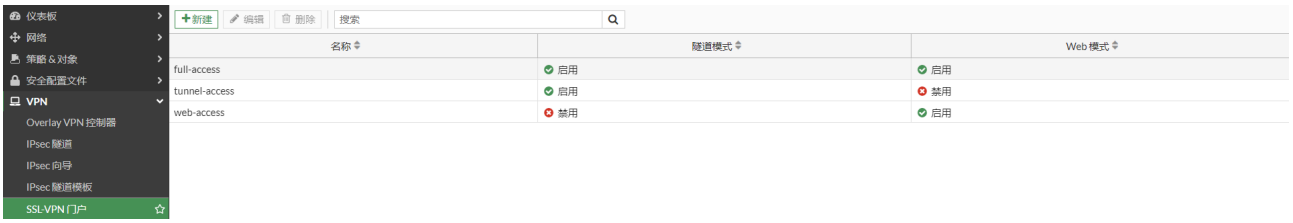


完成用户组创建。

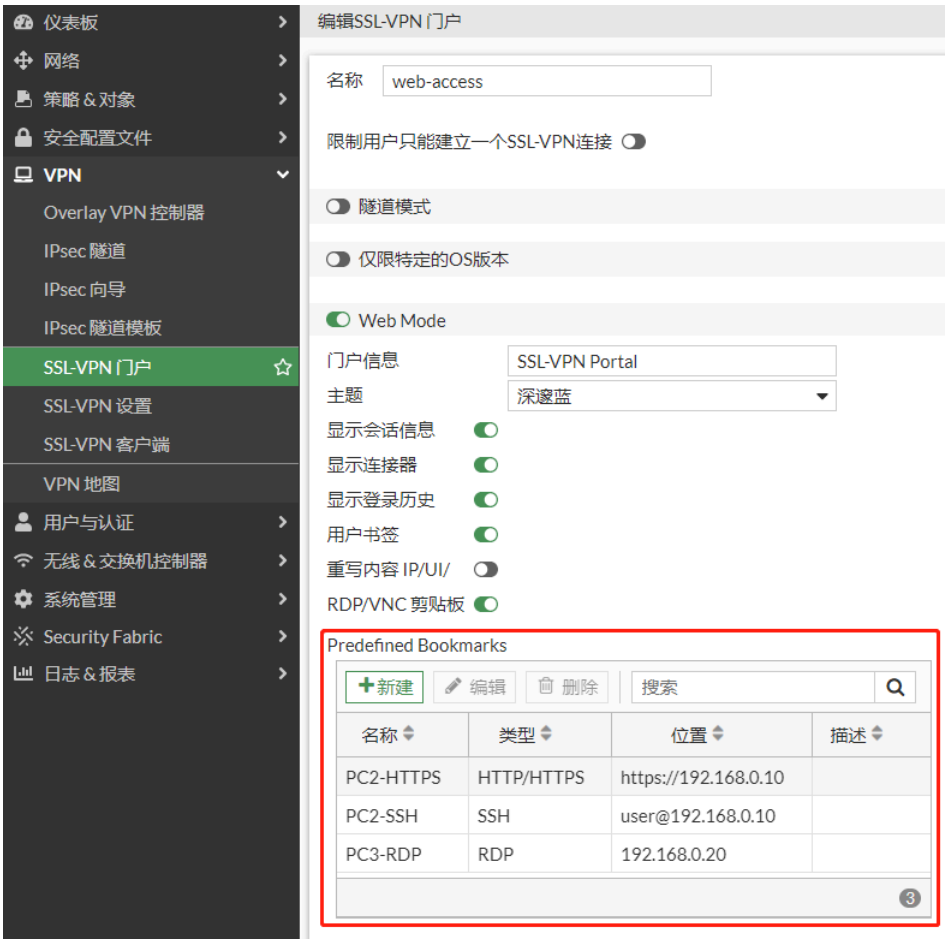


4.配置 SSL - VPN 门户

系统默认创建 3 个 SSLVPN 门户：full-access: 开启了隧道模式和 web 代理模式；tunnel-access 只开启了隧道模式；web-access：只开启 web 代理模式。可以根据实际需求进行修改，也可以新建新的 SSL - VPN 门户。



这里使用 web-access 门户，新建 HTTPS，SSH，RPD 3 个资源供客户端拨号后访问。



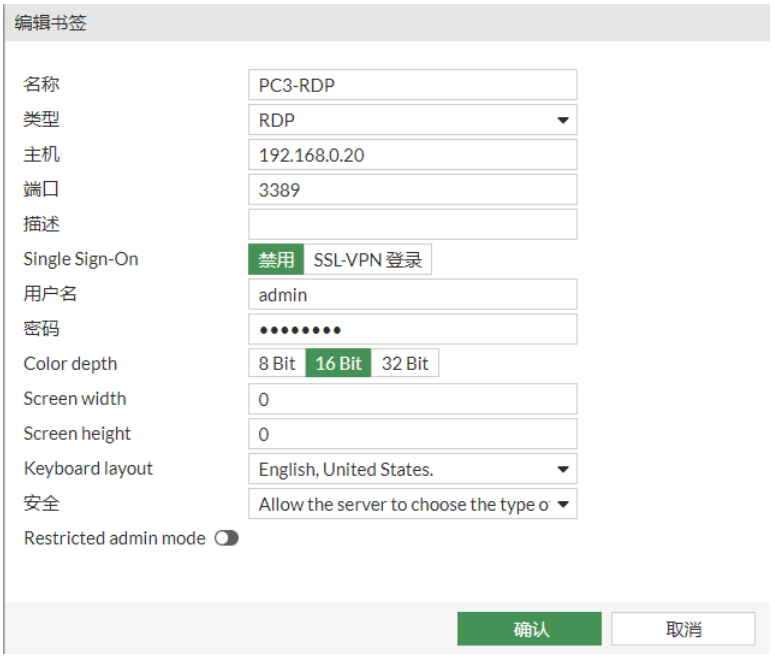
PC2-HTTPS 资源配置：



PC2-SSH 资源配置：

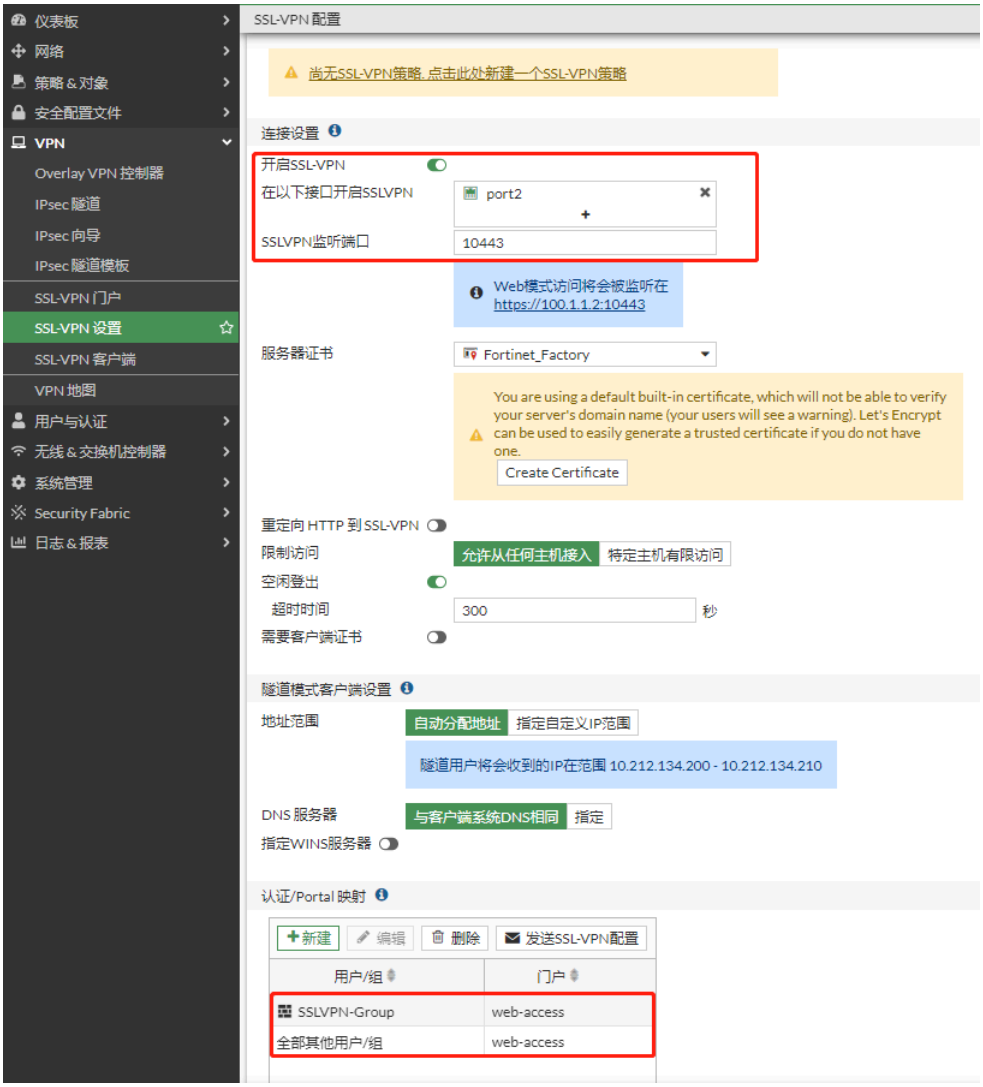


PC3-RDP 资源配置：

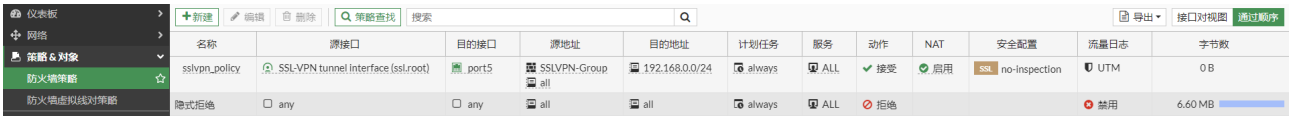


3.配置 SSLVPN

设置提供 SSLVPN 服务的接口和端口，将 SSLVPN-Group 用户组和 web-access 门户关联，全部其他用户/组也要关联一个门户，用于没有配置关联的用户/用户组访问。

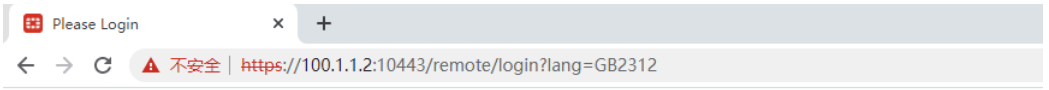


6.创建策略



业务测试

1. PC1 使用 chrome 访问 SSLVPN，并输入用户名和密码。



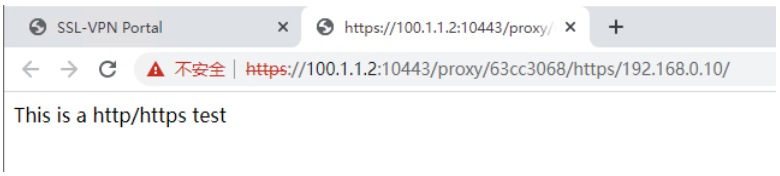


2. 登录成功后，可以看到之前创建好的 3 个资源。

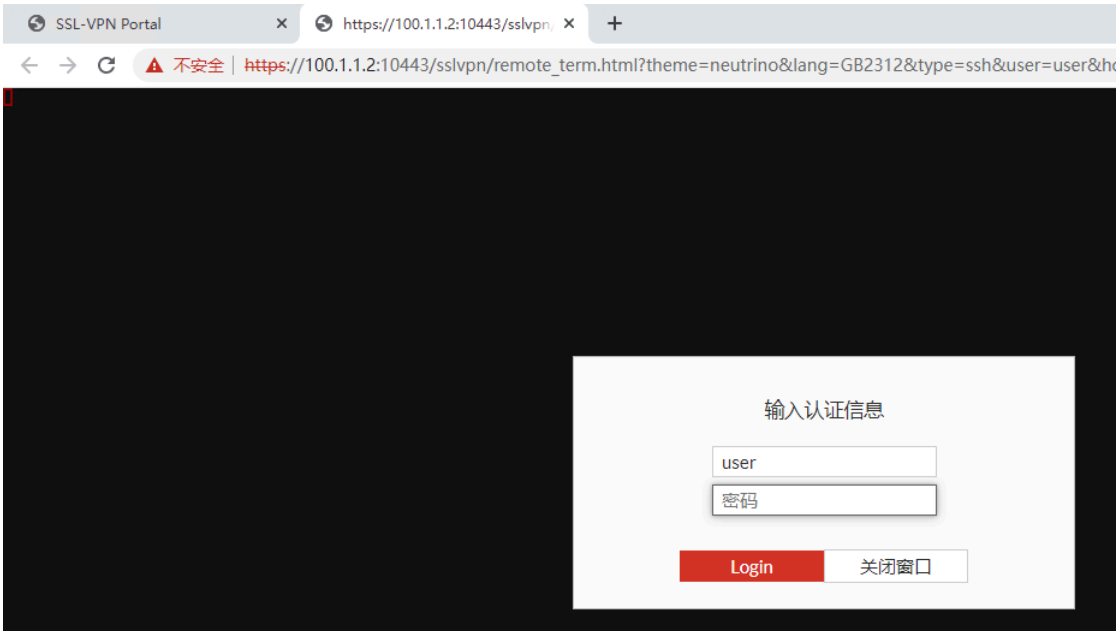


3. 业务访问

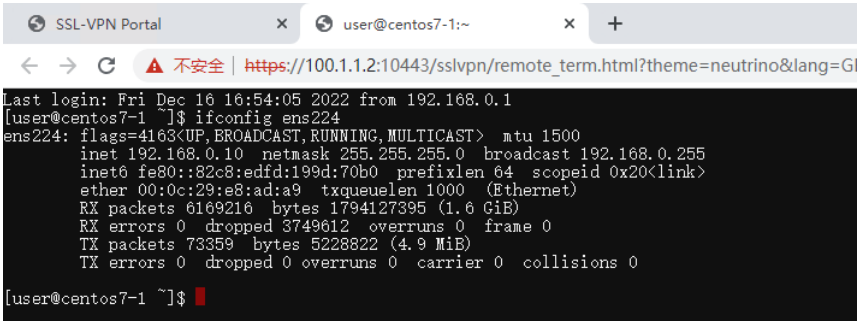
点击 PC2-HTTPS，访问成功。



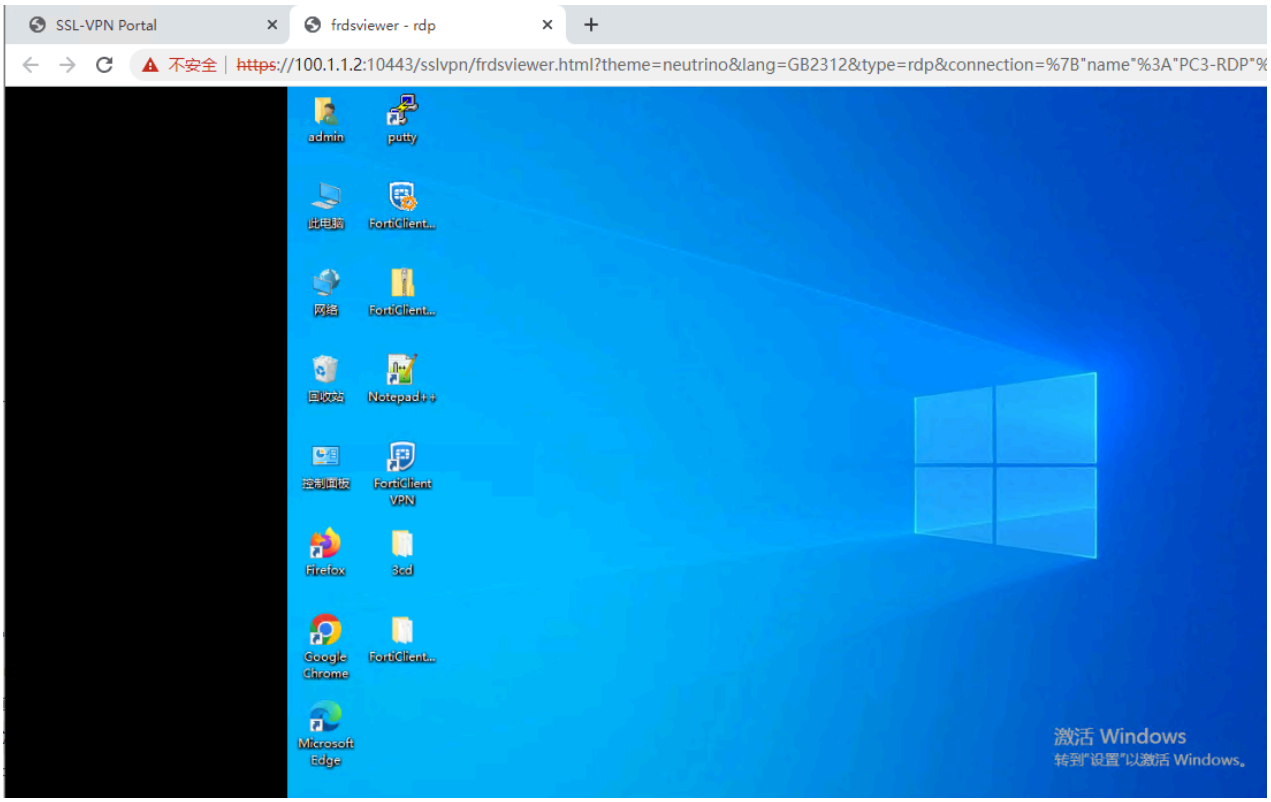
点击 PC2-SSH，访问成功。



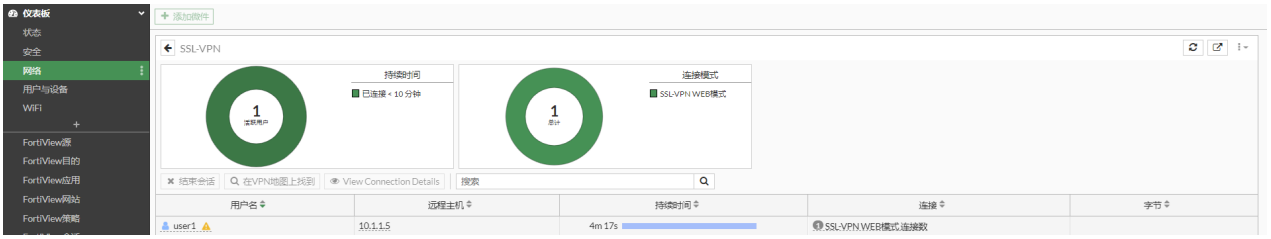
输入 PC2 SSH 的用户的密码，登录成功。



点击 PC3-RDP，访问成功。



4. FortiGate 查看 SSLVPN 用户状态。




```
 # get vpn ssl monitor
SSL-VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out    HTTPS in/out  Two-factor
Auth
 0       user1   SSLVPN-Group   1(1)      289               28788    10.1.1.5       0/0              0/0          0

SSL-VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Web 代理模式自定义代理 IP

默认情况下，使用 Web 代理模式的 SSL VPN 时，客户端访问内部服务器，是通过 FortiGate 上流量的出接口 IP 作为代理 IP 访问内部服务器的，如果想要修改这个 IP 为其他 IP 地址，可以使用如下方法实现。

1. 配置 IPPool，内容为自定义的代理 IP 地址，例如默认出接口 IP 为 172.16.1.254 作为 SSL VPN Web 模式的代理 IP，但用户想使用 172.16.1.1 作为代理 IP，则配置该 IP 为地址池的范围。

```
config firewall ippool
    edit "SSL-VPN-IP-Pool"
        set type overload
        set startip 172.16.1.1
        set endip 172.16.1.1
    next
end
```

2. 在 SSL VPN 的访问策略（从 SSL VPN 接口到内网口方向）中，开启 SNAT，并引用上述地址池。

```
config firewall policy
    edit 1
        set name "SSL-VPN Web Mode"
        set srcintf "ssl.root"
        set dstintf "LAN"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set ippool enable
        set poolname "SSL-VPN-IP-Pool"
    end
```

3. 在 SSL VPN 配置中开启 Web 模式的 SNAT。

```
config vpn ssl settings
    set web-mode-snat enable
end
```

4.**此步很重要：**在 SSL VPN 绑定的接口上，开启第二地址，并配置第二地址为上述地址池中的 IP（32 位）。

```
config system interface
    edit "wan1"
        set ip 172.16.1.254 255.255.255.0
        set secondary-IP enable
        set interface "port2"
        config secondaryip
            edit 1
```

```
                set ip 172.16.1.1 255.255.255.255
            next
        end
    next
end
```

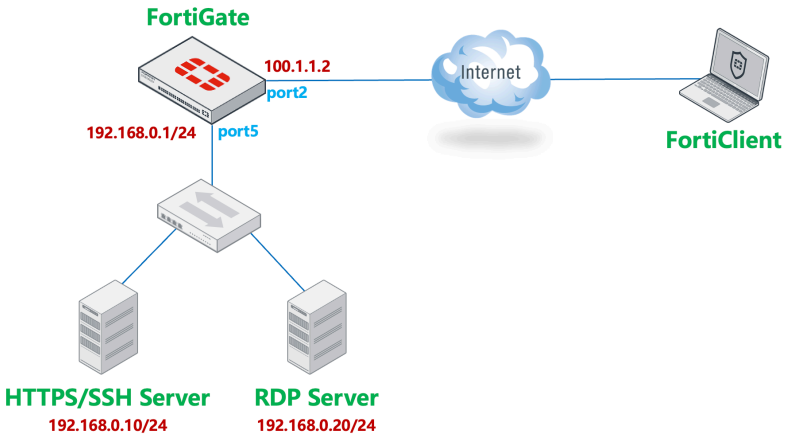
5. 随后，SSL VPN 客户端通过 Web 模式访问内网 Web 服务器，将通过地址池中配置的 IP 作为代理源 IP 访问。

SSL VPN 数字证书认证

网络需求

在外移动办公的工作人员需要通过 SSL VPN 并且使用数字证书认证的方式，拨入到公司内网来对内网主机进行访问。

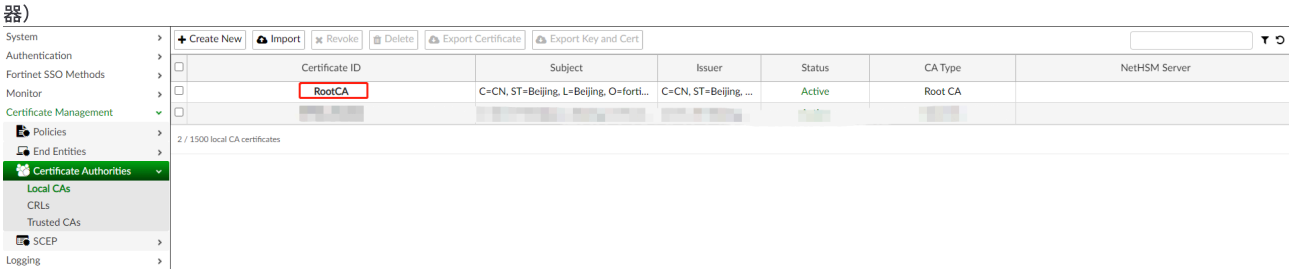
##网络拓扑



配置步骤

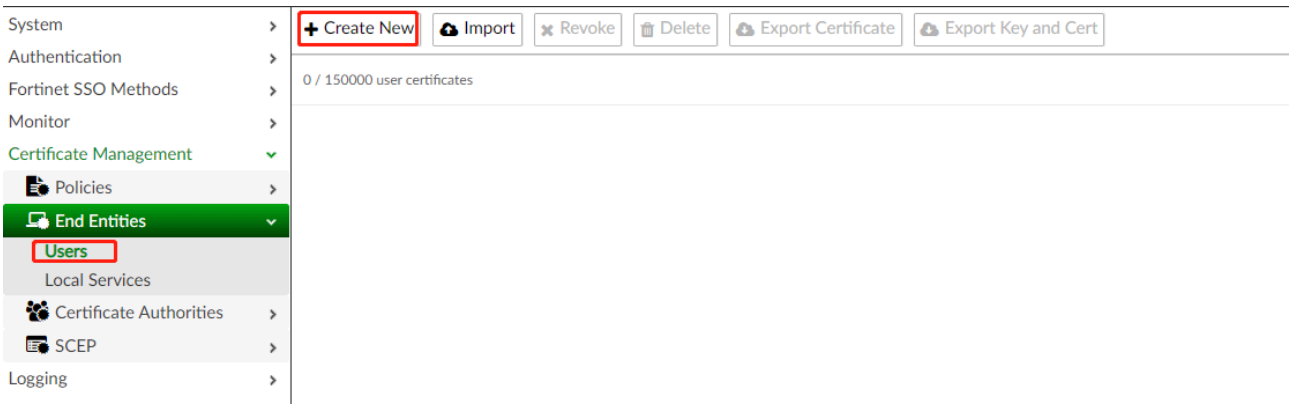
使用 FortiAuthenticator 为 FGT - BJ 和终端颁发证书

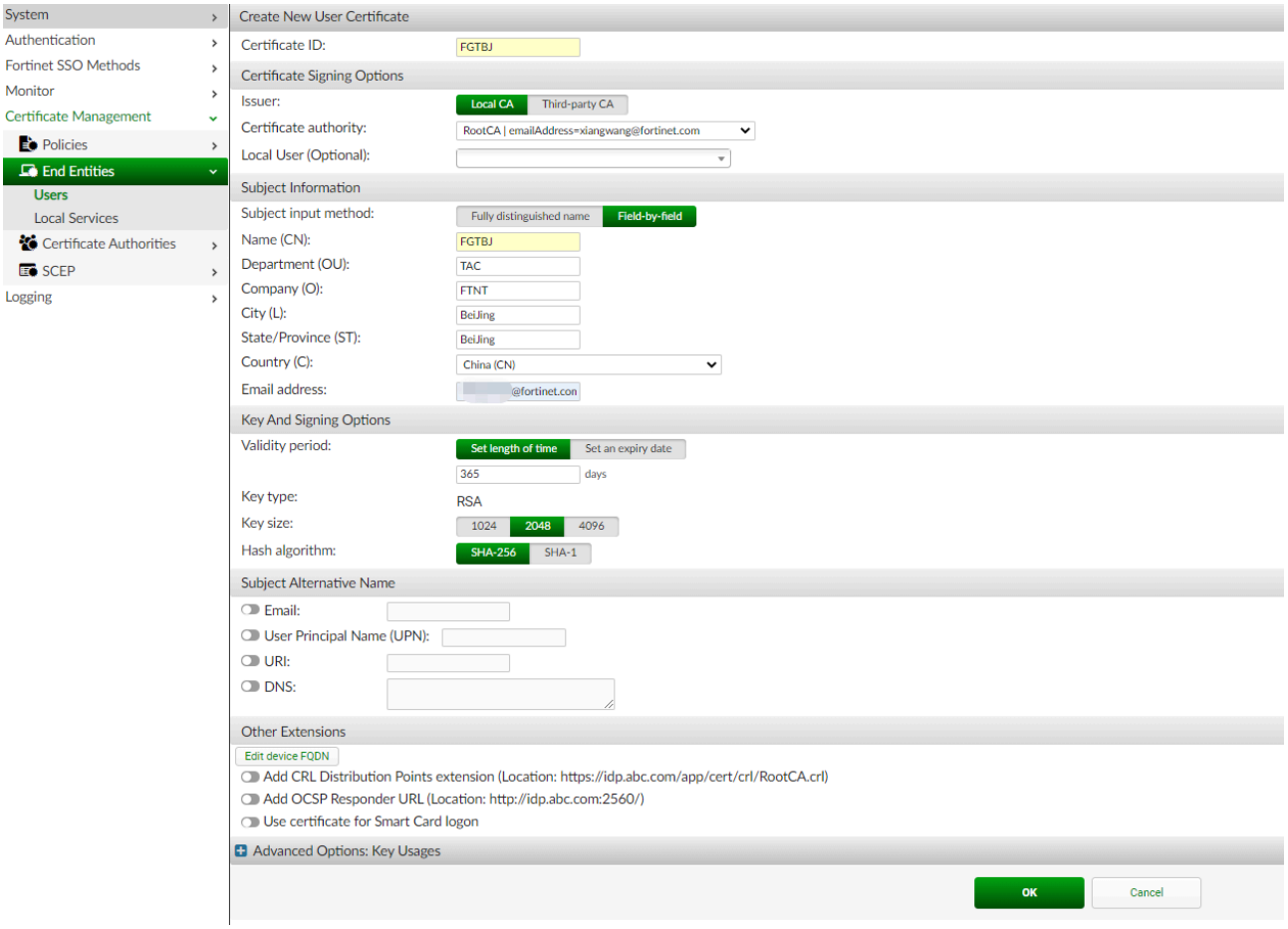
FortiAuthenticator 可以做为证书服务器，RootCA 是已创建好的根证书，这里使用 RootCA 为这两台 FortiGate 签发证书。（也可以使用其他证书服务器）



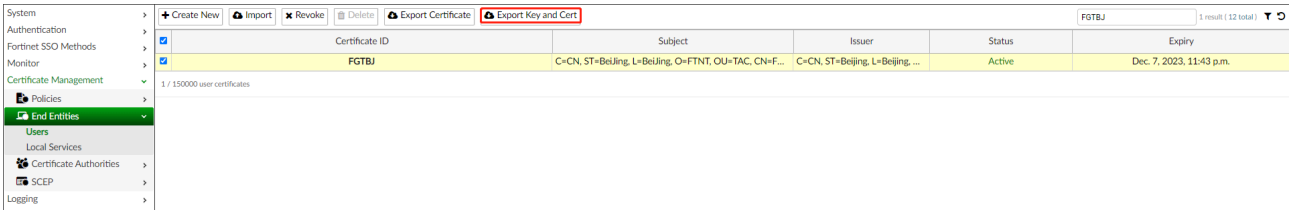
1.为 FGT - BJ 颁发证书

为 FGT - BJ 创建证书。选择“End Entities”-->"User"，点击“Create New”，输入证书相关信息，并点击完成。

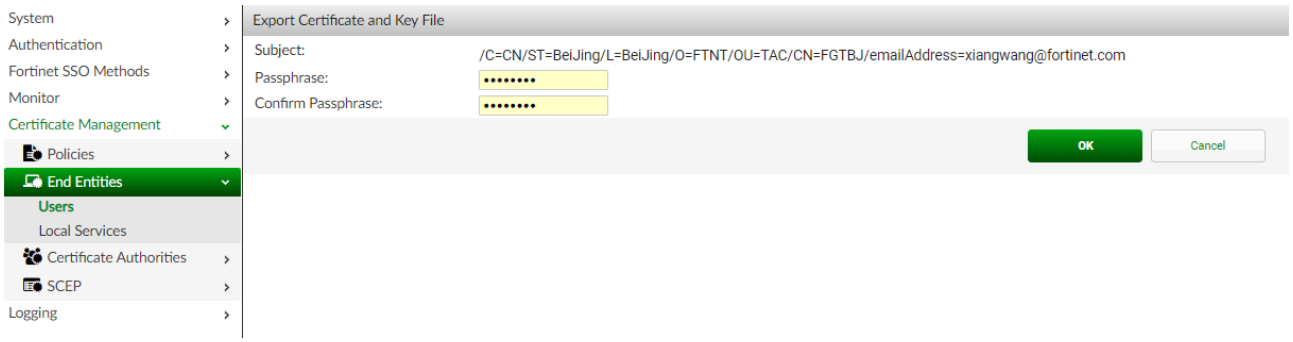




点击“Export Key and Cert”导出 FGTBJ 证书的私钥和公钥。



输入一个密码保护私钥

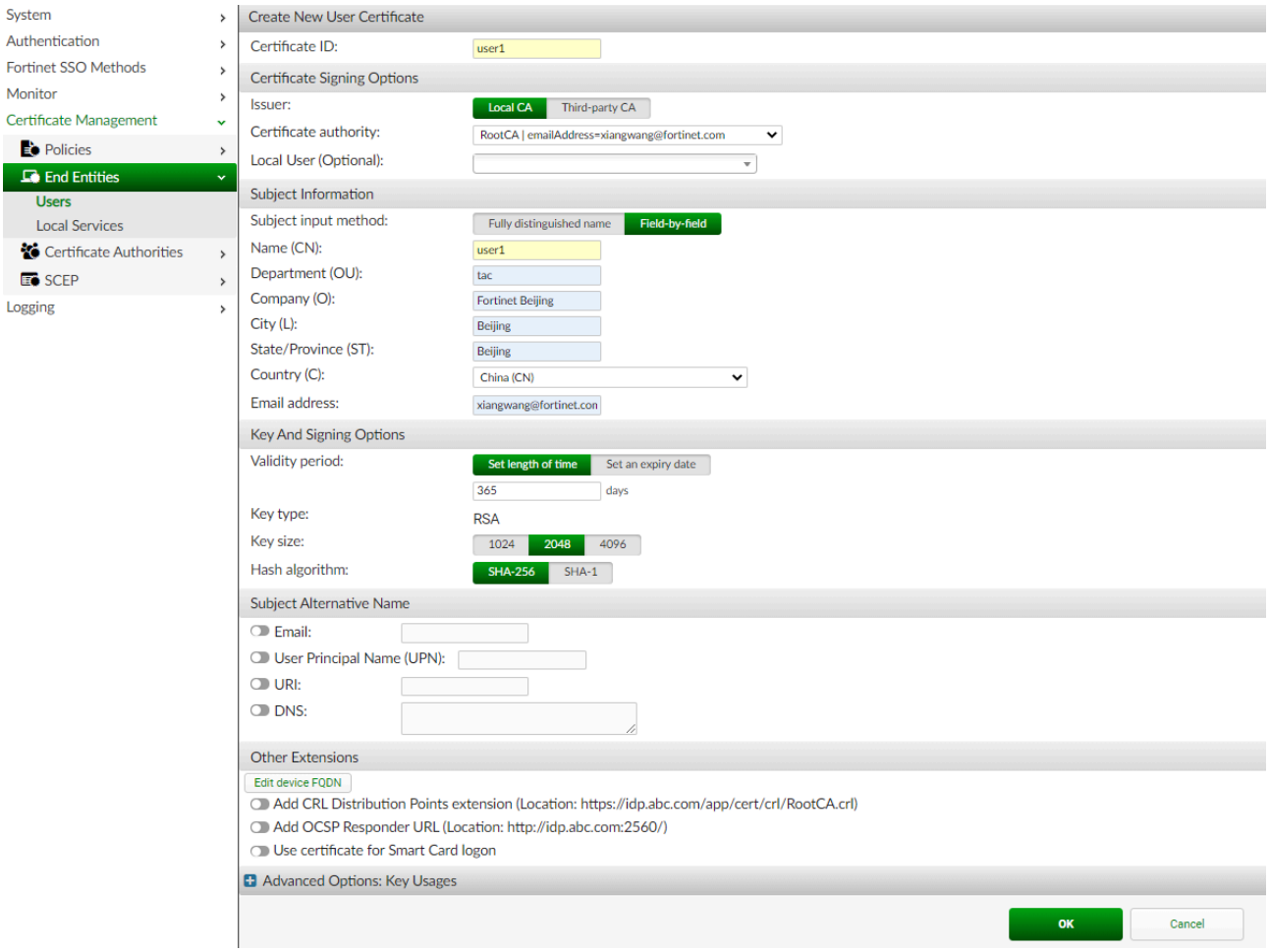


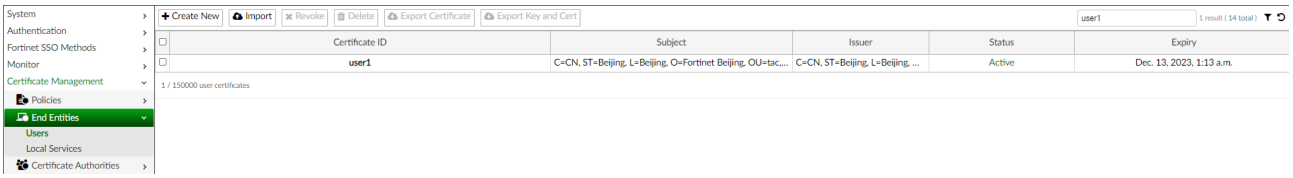
下载证书



2.为终端颁发证书

为终端颁发证书和上述步骤是一样的。

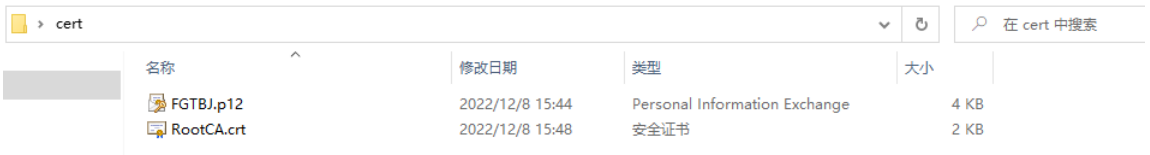




FortiGate 导入证书

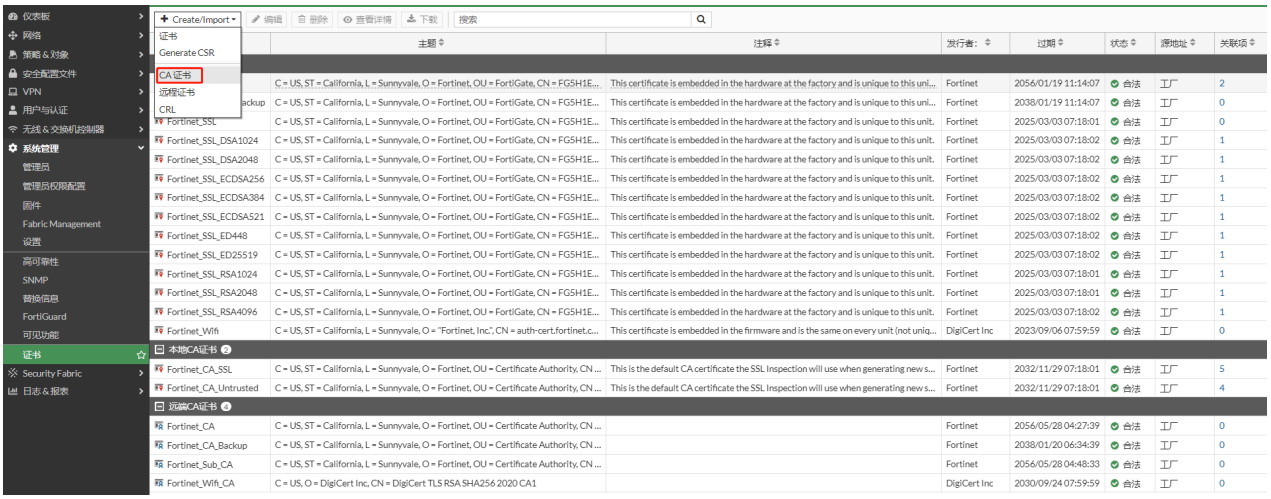
将根证书 RootCA.crt 和证书 FGTBJ.p12 导入到 FGT_BJ。

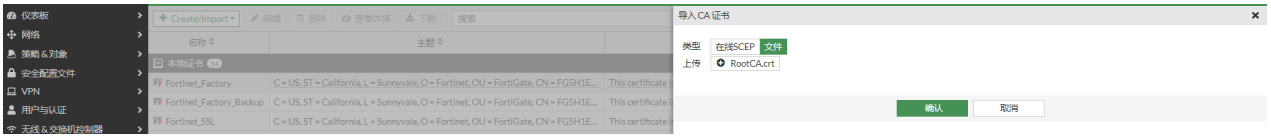
在数字证书认证中，通信双方使用 CA 证书进行数字证书合法性验证，双方各有自己的公钥（网络上传输）和私钥（本地存储）。发送方对报文进行 Hash 计算，并用自己的私钥对报文计算结果进行加密，生成数字签名。接收方使用发送方的公钥对数字签名进行解密，并对报文进行 Hash 计算，判断计算结果与解密后的结果是否相同。如果相同，则认证通过；否则认证失败。



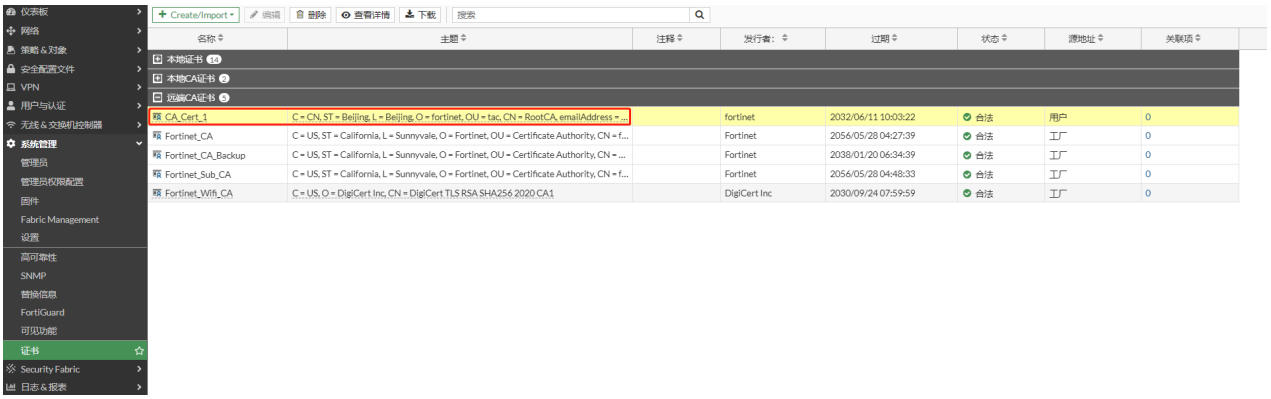
1. 导入 CA 证书

选择“系统管理”-->“证书”，点击“Create/Import”，选择 CA 证书。



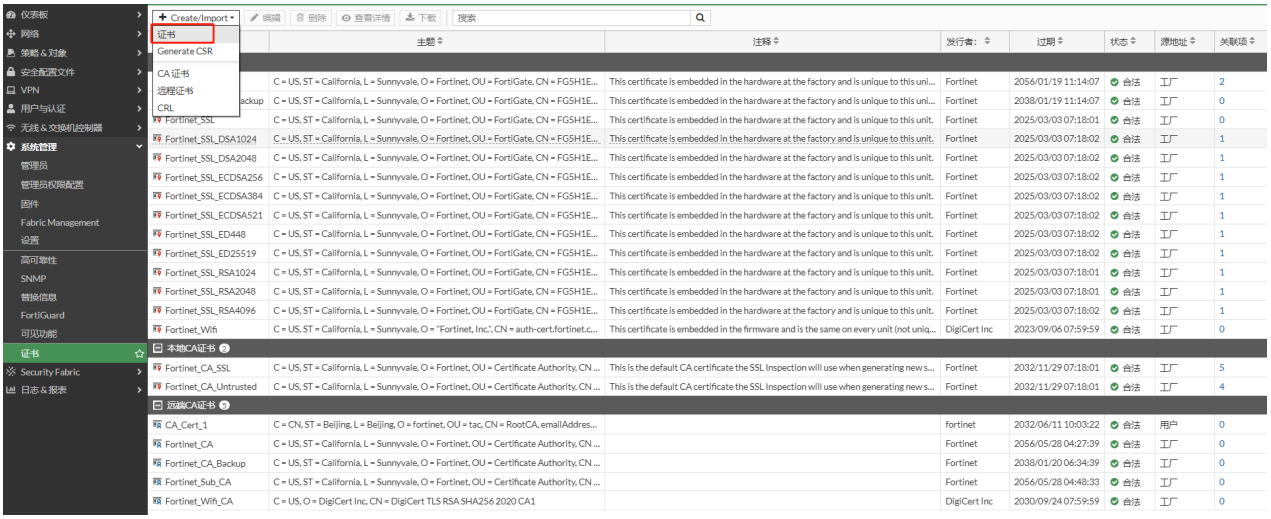


查看导入的 CA 证书

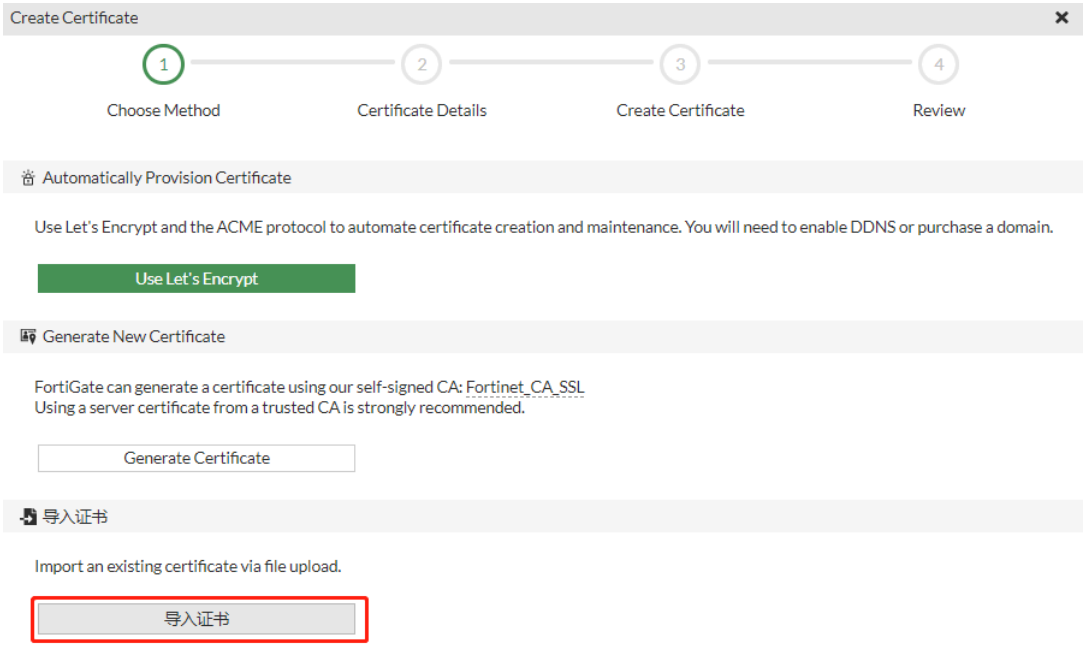


2. 导入本地证书

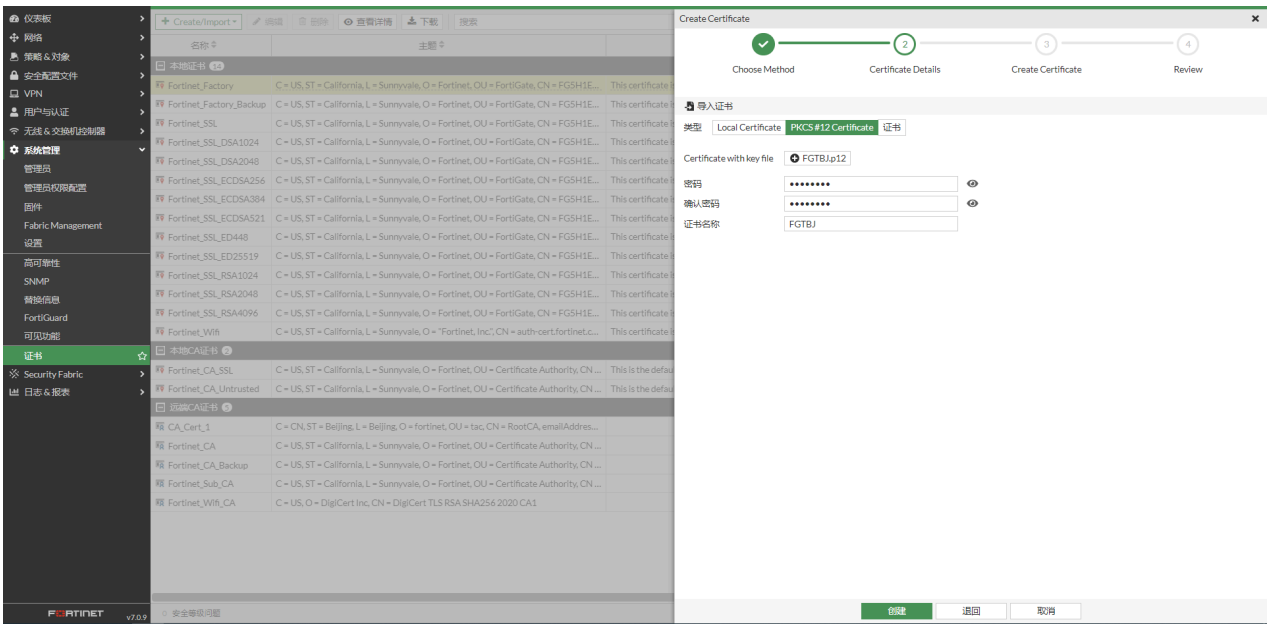
选择“系统管理”-->“证书”，点击“Create/Import”，选择“证书”。



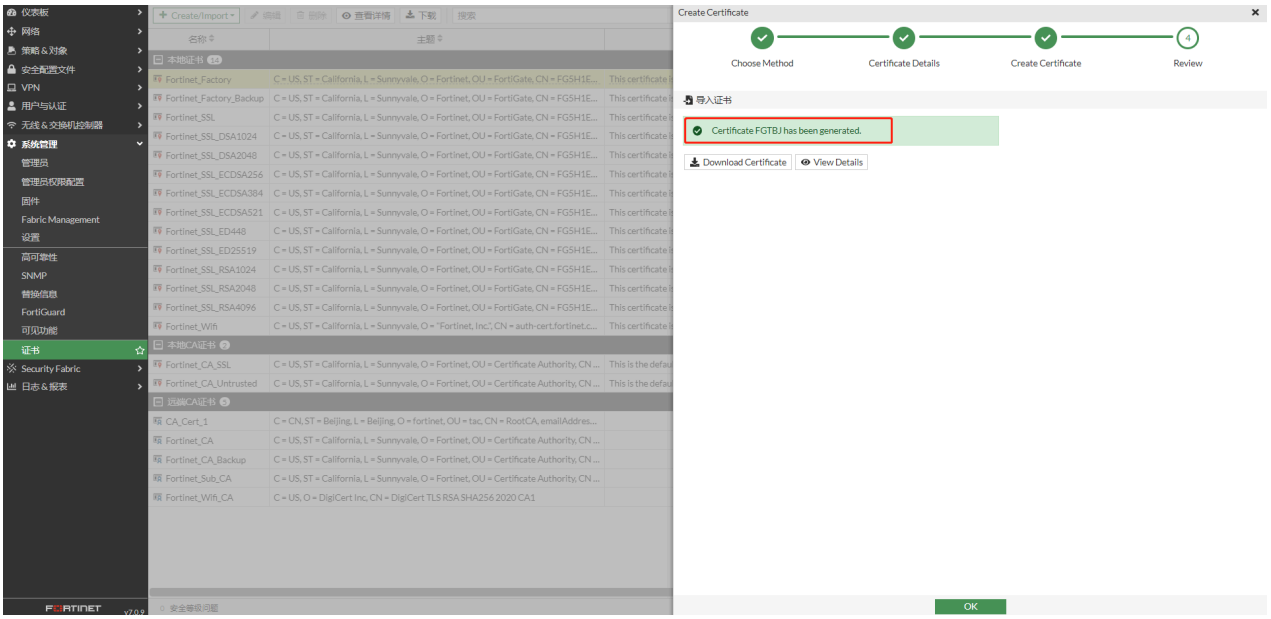
点击导入证书



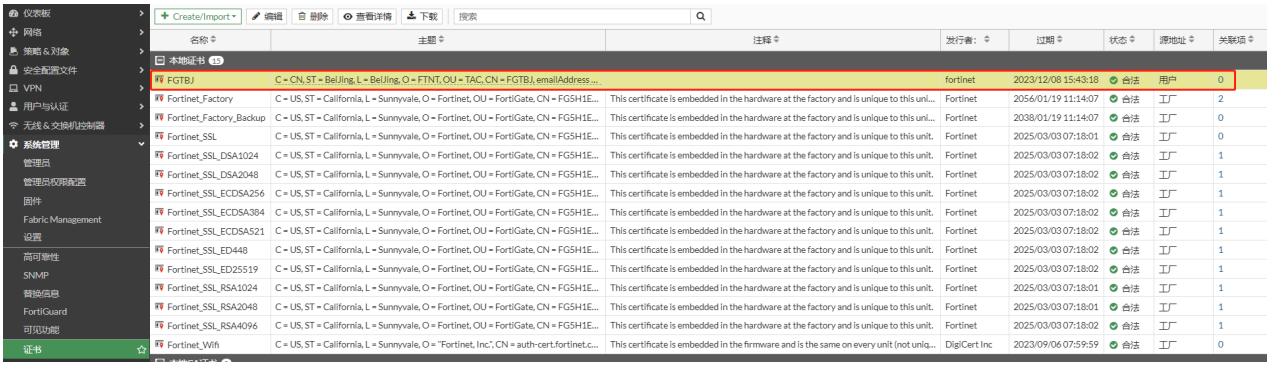
输入证书的密码，点击“创建”



提示证书已成功导入



查看导入的本地证书



终端导入证书

将根证书 RootCA.crt 和证书 user1.p12 导入到终端。



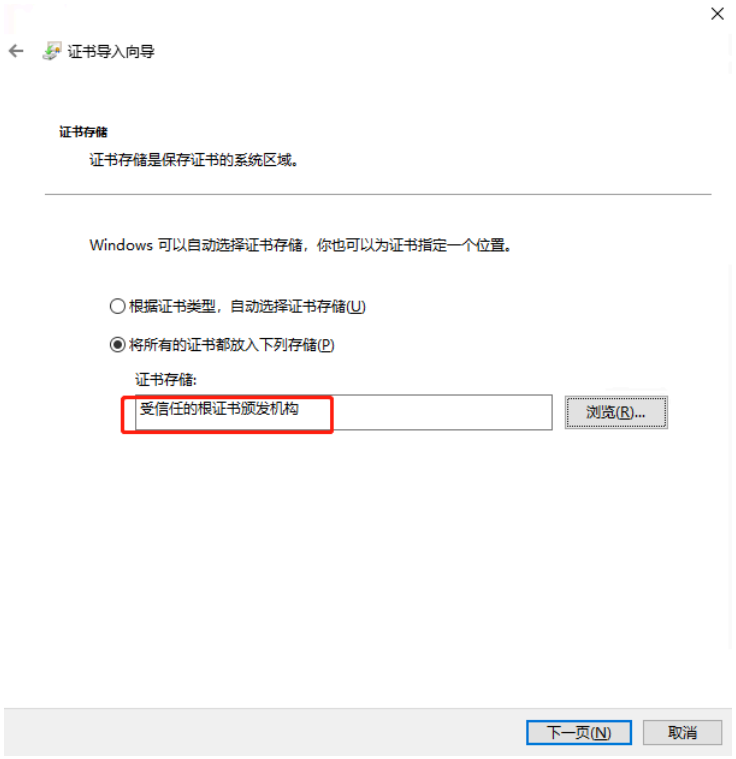
1. 导入根证书 RootCA.crt

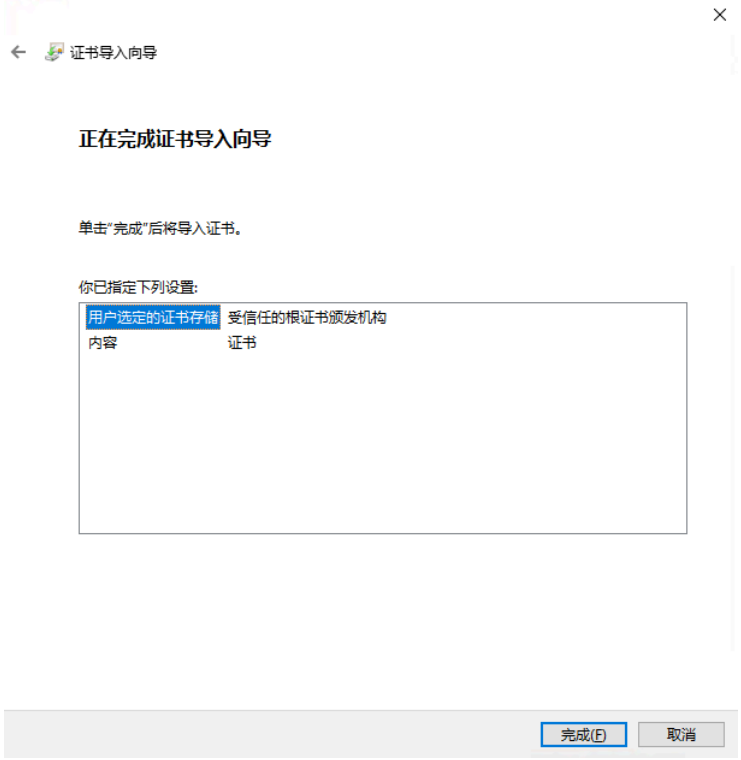
   双击 RootCA.crt，点击“安装证书”。





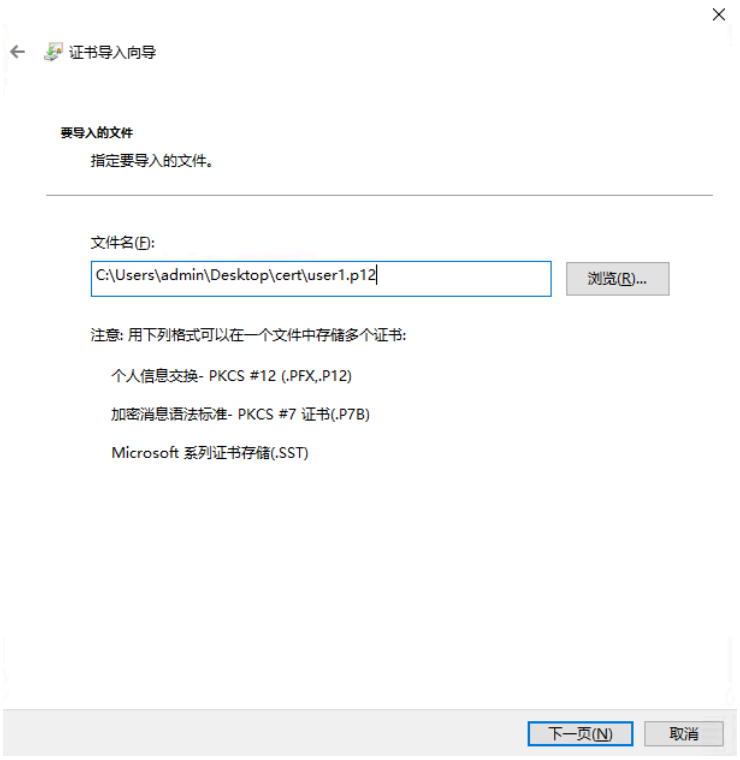
证书存入“受信任的根证书颁发机构”





2. 导入个人证书 user1.p12

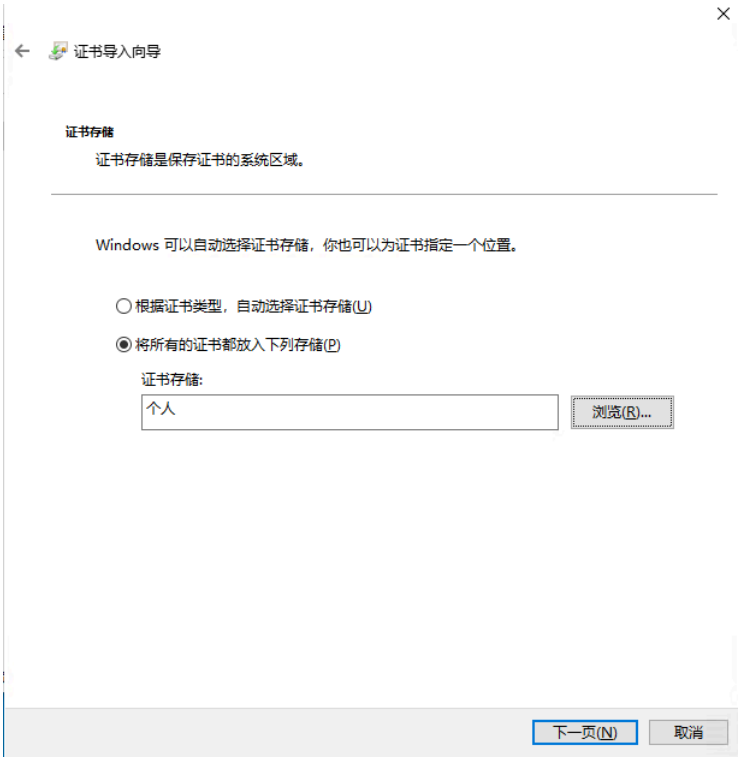
   双击 user1.p12，然后安装证书

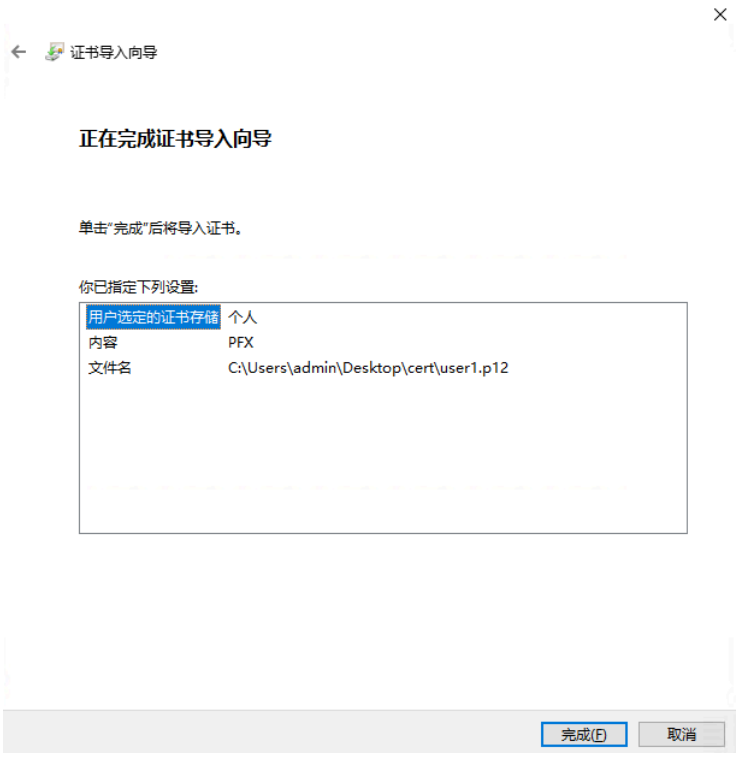


输入证书的密码



证书存入“个人”

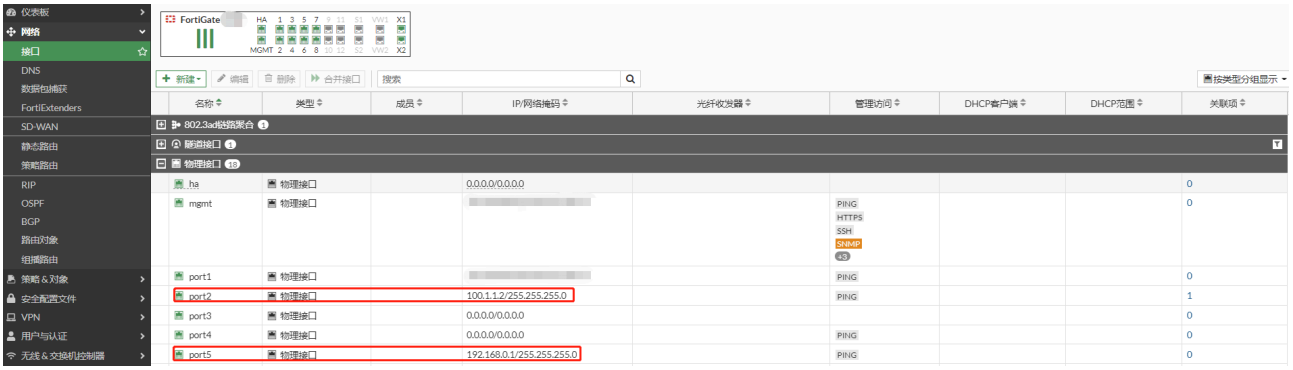


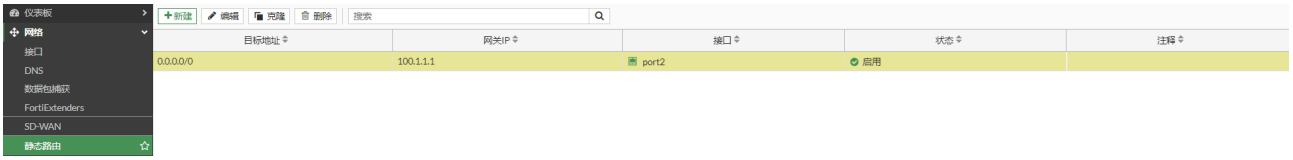


配置 SSL VPN

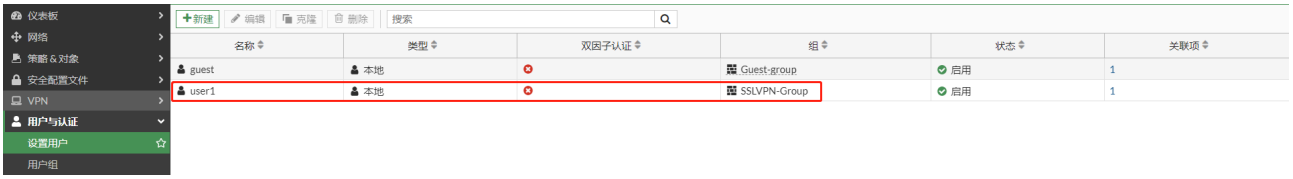
1.基本配置

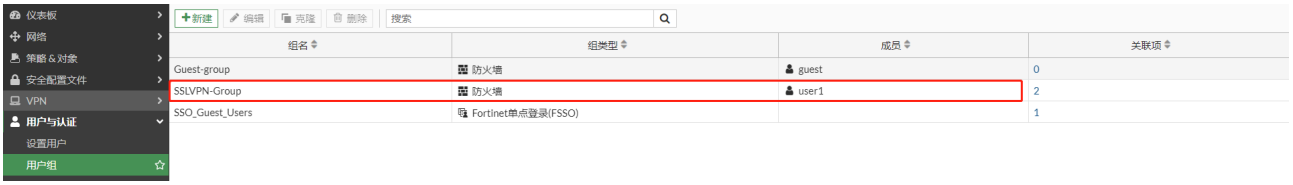
配置接口 IP 和路由





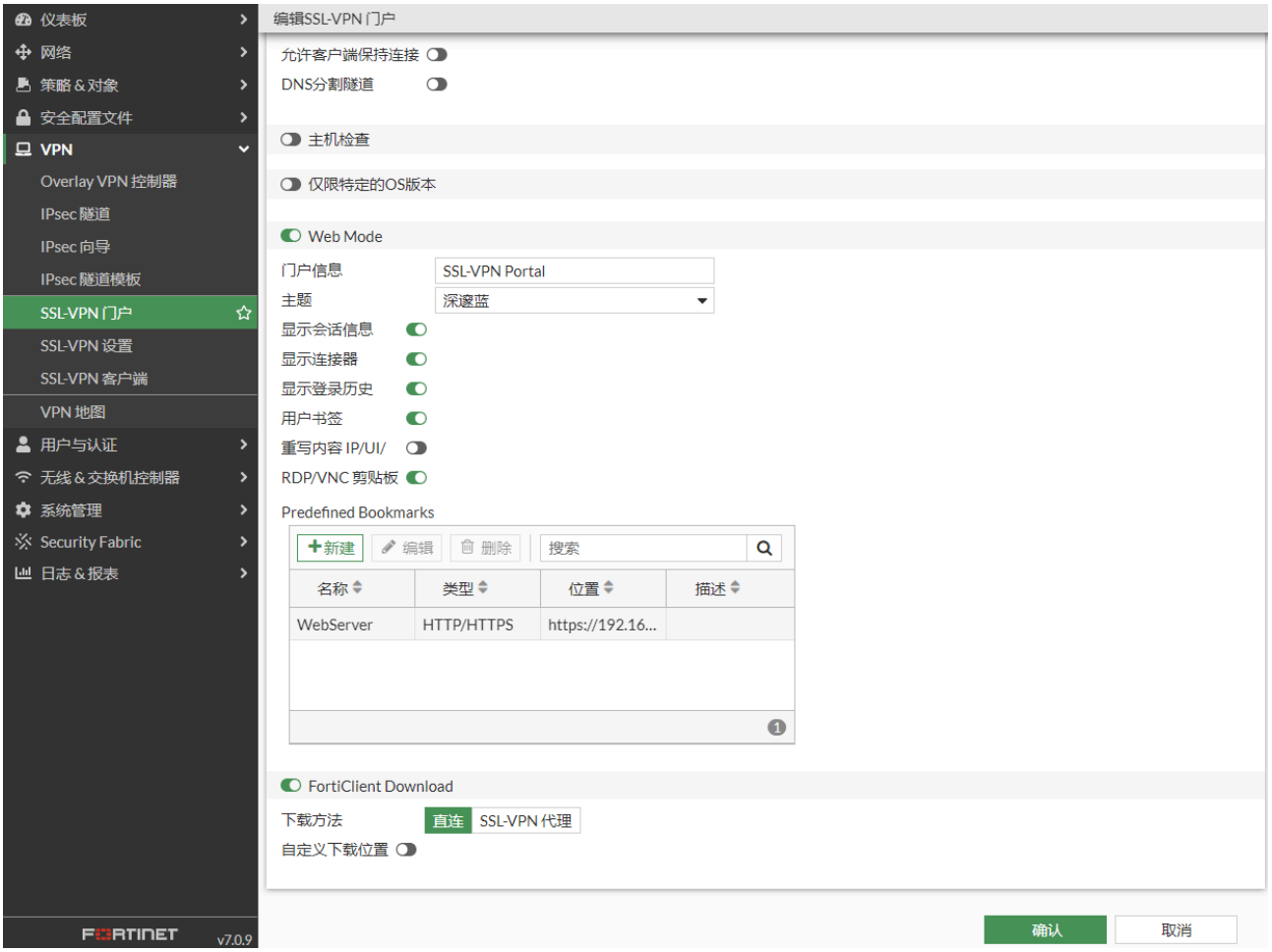
2.创建用户，并将用户加入到用户组。





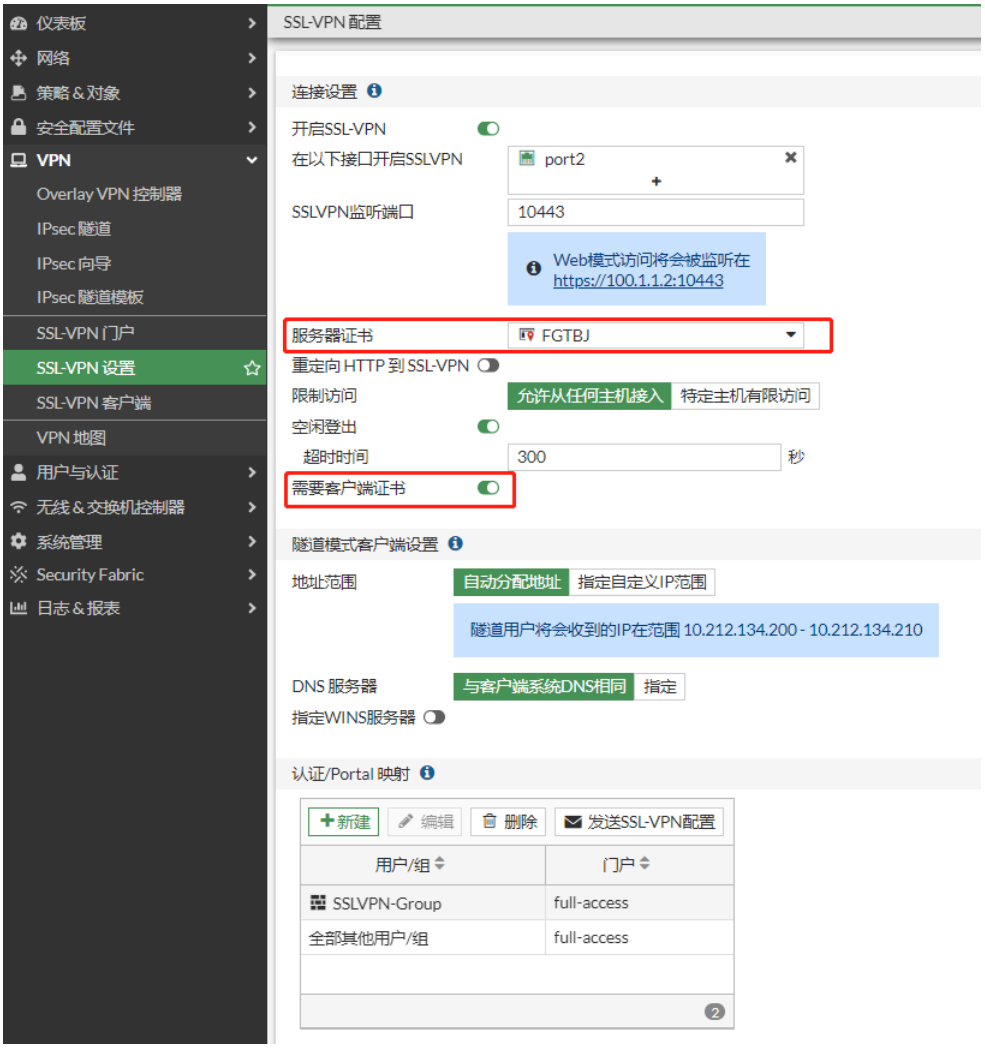
3.配置 SSL - VPN 门户





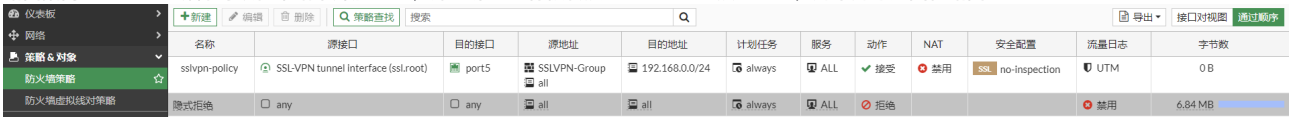
4.配置 SSLVPN

服务器证书选择“FGTBJ”，FortiGate 会使用 FGTBJ 证书与终端交互；勾选“需要客户端证书”，那么认证过程中 FortiGate 会要求客户端提供证书。



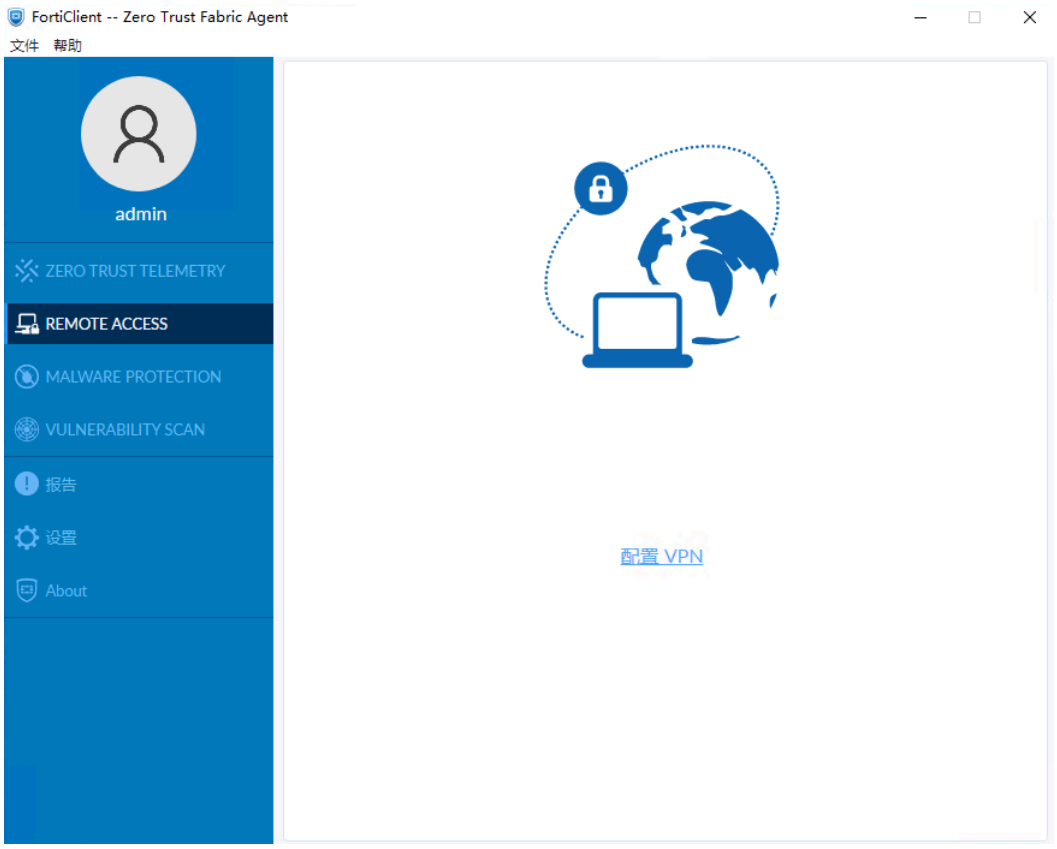
5.创建策略

当客户端 SSLVPN 拨号成功后，将会使用获取的地址（10.200.1.10-10.200.1.200）访问内部主机，因此内部网络需要增加到 10.200.1.0/24 网段的回程路由指向 FortiGate 或者 可以在策略中开启 NAT，那么源地址将被转换为 FortiGate 接口地址，则不用考虑回程路由。



配置 FortiClient

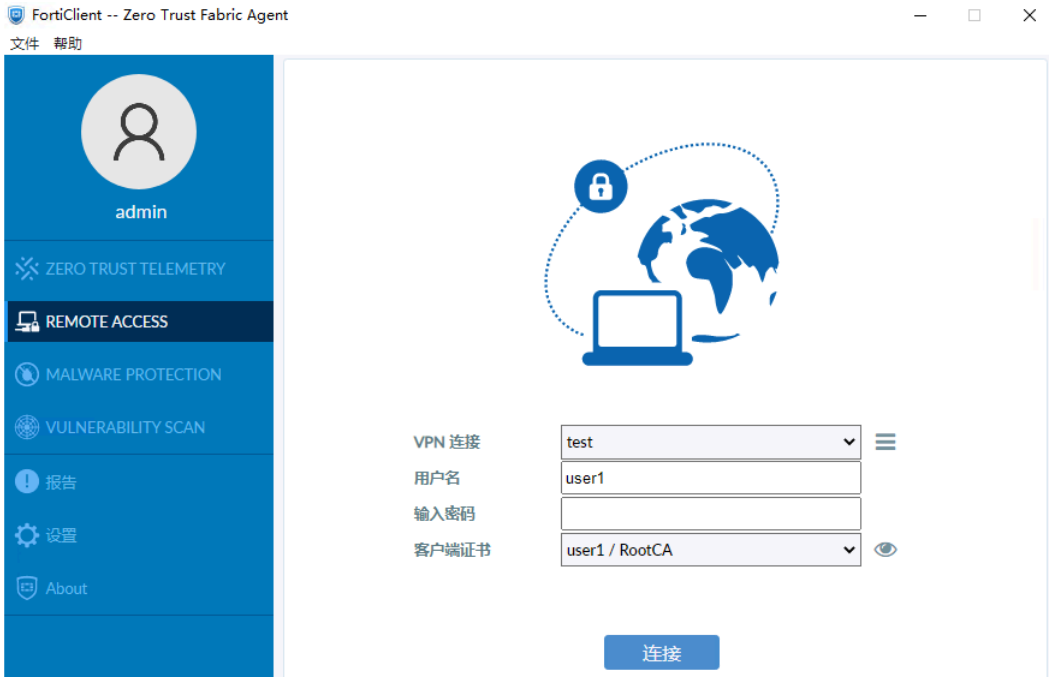
1. 选择“Remote Access”，点击“配置 VPN”。



2. 选择 SSL VPN，设置连接名，远程网关，SSLVPN 端口，用户名以及选择终端证书，然后点击保存。



3. 配置完成。

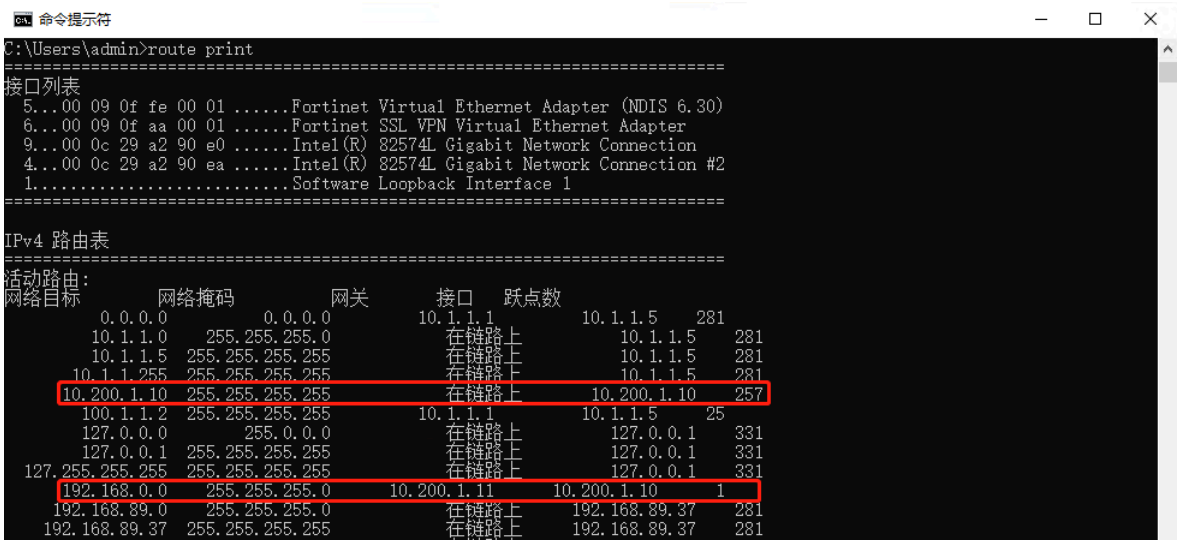


FortiClient 拨号测试

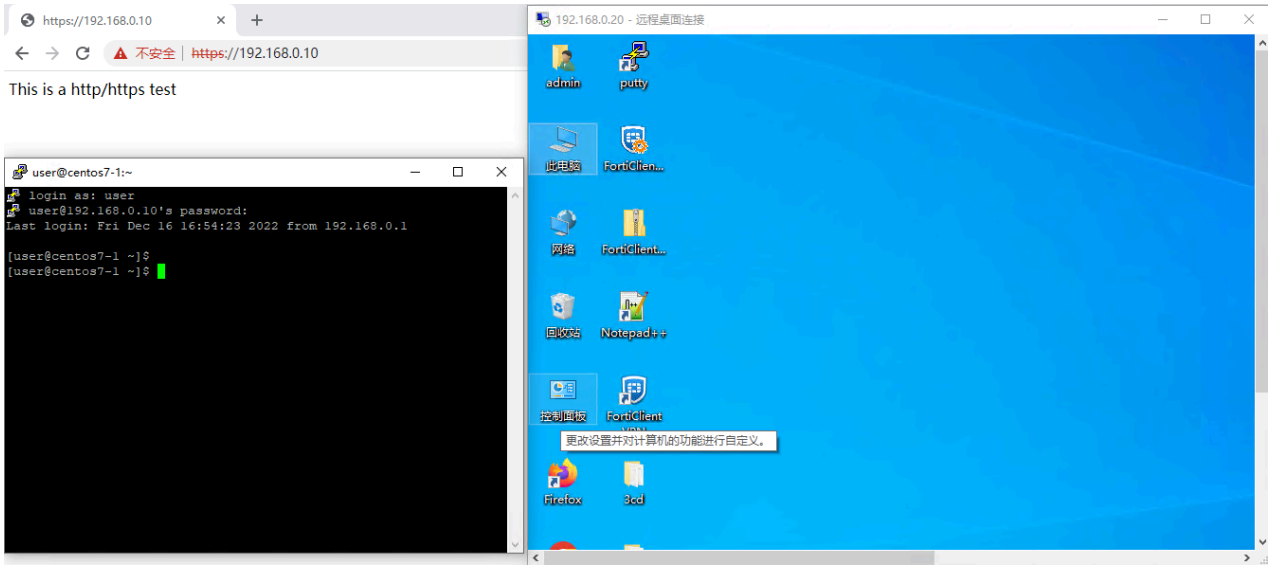
1. 输入密码，点击连接。



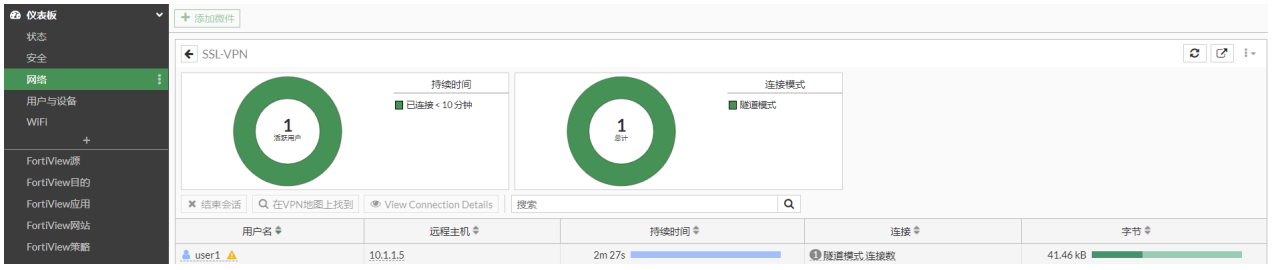
查看终端获取到的路由



2. 访问内网主机 HTTPS，SSH，RDP 服务都正常



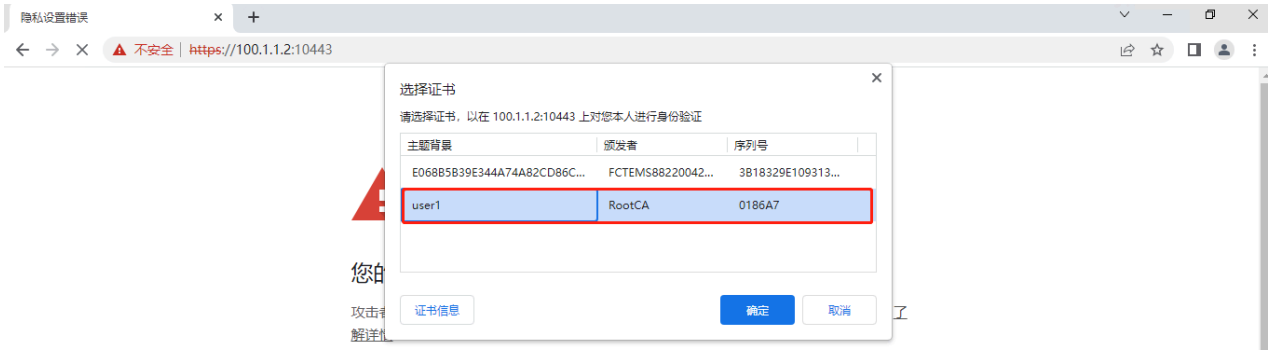
3. FortiGate 查看 SSLVPN 连接



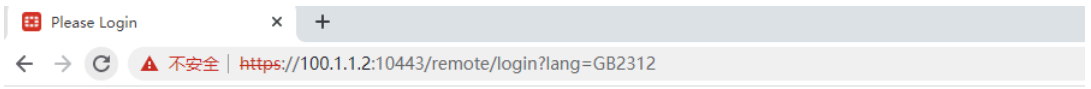

```
# get vpn ssl monitor
SSL-VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out    HTTPS in/out    Two-
factor Auth
 0       user1   SSLVPN-Group   1(1)            189    28625    10.1.1.5       0/0     0/0     0

SSL-VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
 0       user1   SSLVPN-Group   10.1.1.5         175     27388/14075    10.200.1.10
```

浏览器拨号测试

1. PC1 使用 chrome 访问 SSLVPN，在弹出的证书选择框中，使用 user1 证书。



2. 跳转到输入账号密码界面。

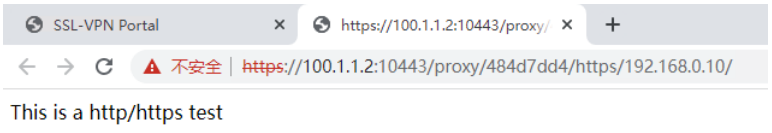




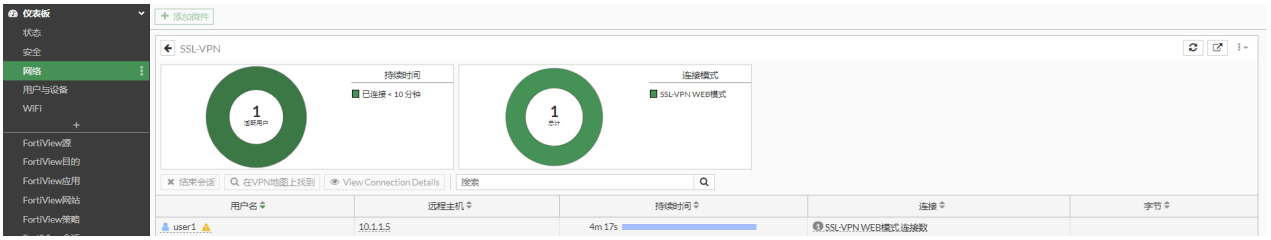
3. 登录成功。



4. 访问 webserver 成功。



5. FortiGate 查看 SSLVPN 用户状态。




```
 # get vpn ssl monitor
SSL-VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out    HTTPS in/out  Two-factor
Auth
 0       user1   SSLVPN-Group   1(1)      289               28788    10.1.1.5       0/0              0/0          0

SSL-VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

SSL VPN 源地址连接控制

简介

SSL VPN 可以在功能层面对访问的客户端源地址进行限制，分为两种方式：

- 白名单：仅允许配置的 IP 范围访问 SSL VPN 服务。
- 黑名单：禁止特定的 IP 范围访问 SSL VPN 服务，其他源 IP 允许正常访问。

配置步骤

白名单

需求：只允许中国地区的 IP 访问 FortiGate 的 SSL VPN 服务。

1. 在“VPN → SSL - VPN 设置”页面中，配置“限制访问”的方式为“特定主机有限访问”，点击下方的+ 按钮，在选择条目窗口中点击“新建”按钮。



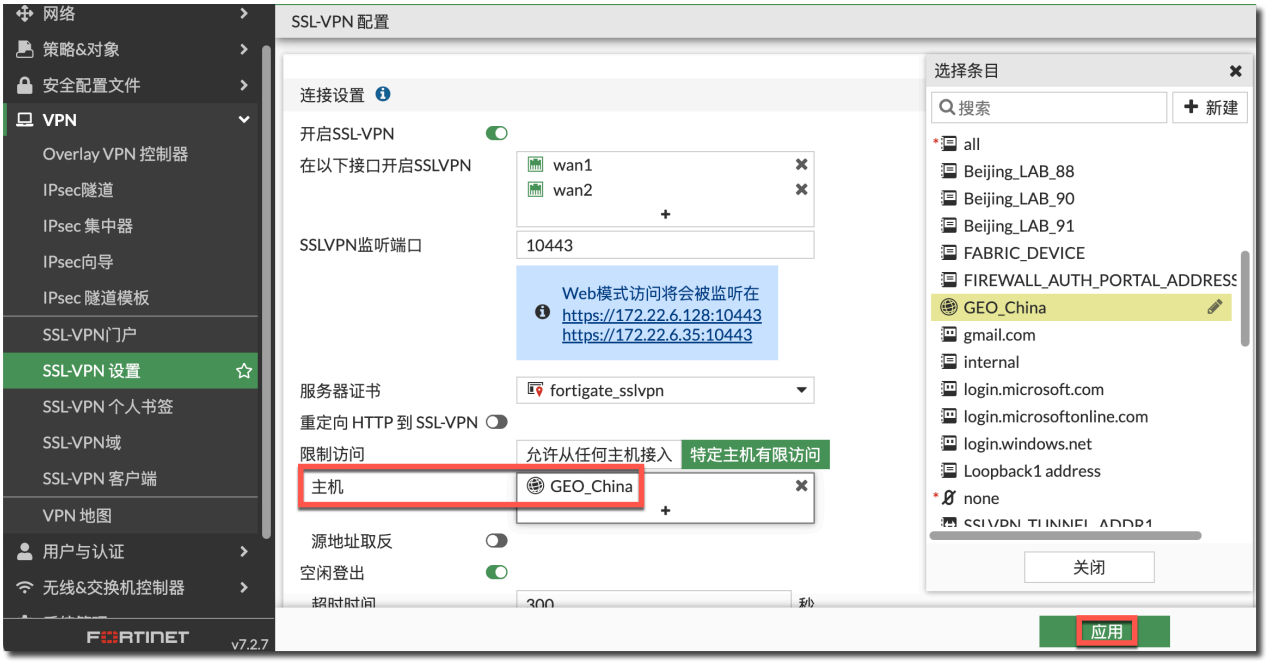
2. 选择地址类型。



3. 配置地址名称，类型选择“地理”，国家选择 China，点击确认下发配置。



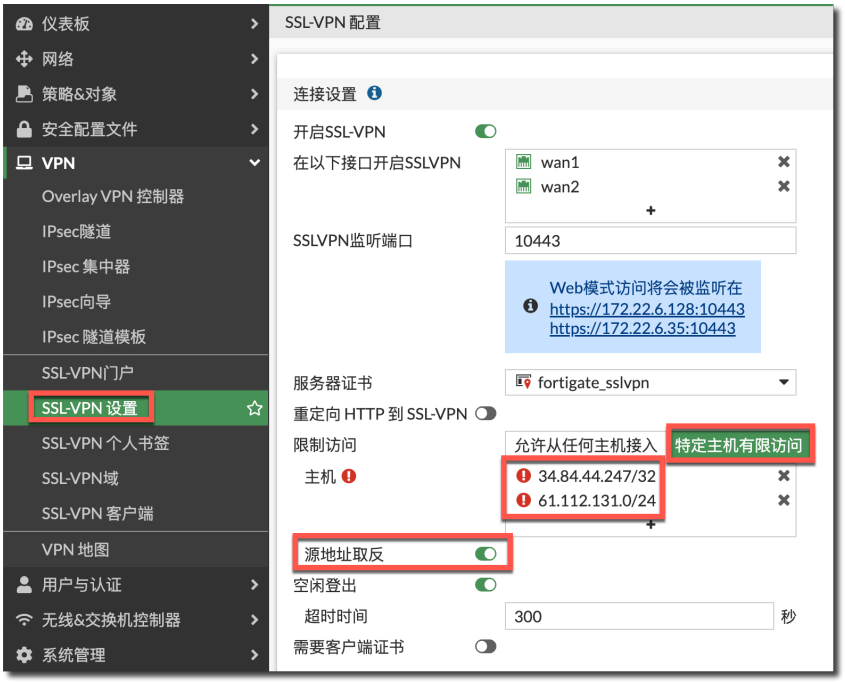
4. 选择上步创建的地理地址对象为限制访问的主机，点击页面下方的应用按钮下发配置。




```bash
config firewall address
    edit "GEO_China"
        set type geography
        set country "CN"
    next
end

config vpn ssl settings
    set source-address "GEO_China"
end
```

黑名单

在“VPN → SSL - VPN 设置”页面中，配置“限制访问”的方式为“特定主机有限访问”，开启“源地址取反”功能，并配置不允许访问的客户端网段，如下图所示，除了“34.84.44.247/32”和“61.112.131.0/24”网段不允许访问 SSL VPN 服务外，其他客户端可以正常访问 SSL VPN 服务。




```bash
config vpn ssl settings
    set source-address "34.84.44.247/32" "61.112.131.0/24"
    set source-address-negate enable
end
```

解锁 SSL VPN 用户

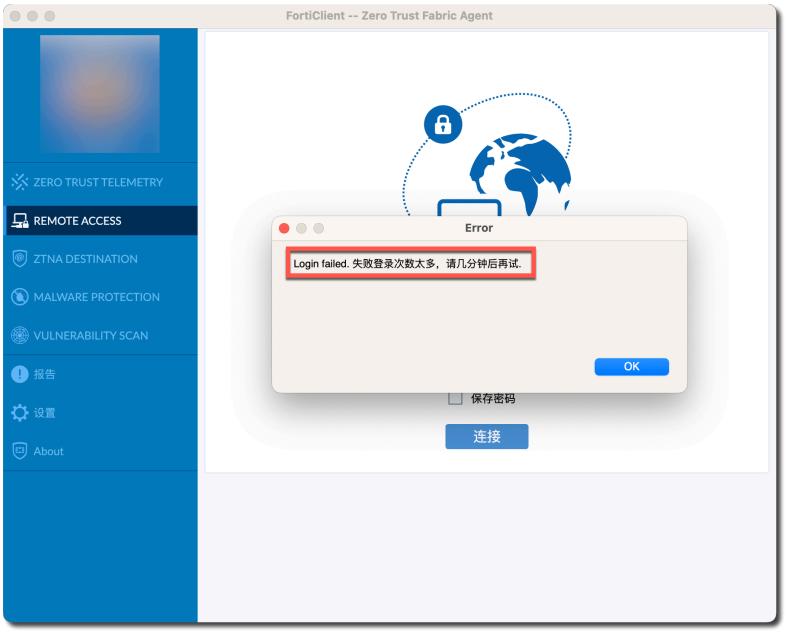
> **重要**
>
> 此功能在 FortiOS 7.2.6 及以上版本、7.4.1 及以上版本、7.6.0 及以上版本支持。

功能介绍

1. 默认配置下，SSL VPN 用户登录失败 2 次后会被锁定 60s。

```bash
config vpn ssl setting
    set login-attempt-limit 2
    set login-block-time 60
end
```

2. SSL VPN 用户被锁定后，FortiClient 登录会有如下提示。



3. FortiGate 的 SSL VPN Debug 会有如下提示。

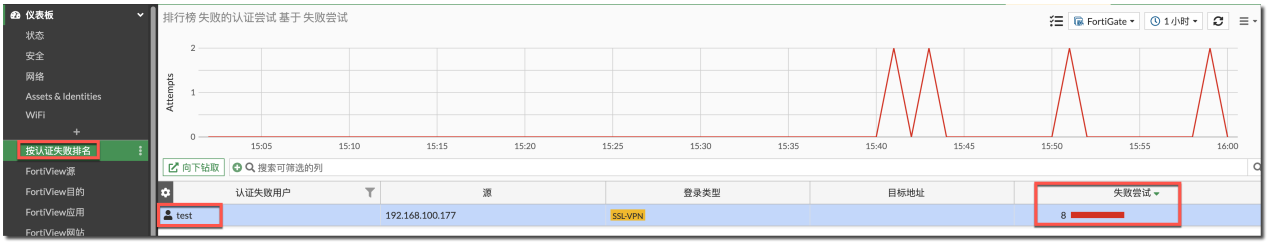
```bash
diagnose debug application sslvpn -1
diagnose debug enable

[15356:root:6]req: /remote/logincheck
[15356:root:6]fsv_blocklist_check:65 locked: rowid=1,host=192.168.100.177
[15356:root:6]Transfer-Encoding n/a
[15356:root:6]Content-Length 75
[15356:root:6]readPostEnter:19 Post Data length 0.
[15356:root:6]sslConnGotoNextState:318 error (last state: 1, closeOp: 0)
[15356:root:6]Destroy sconn 0x7fb0ea2800, connSize=0. (root)
[15356:root:6]SSL state:warning close notify (192.168.100.177)
```

解锁方法

1. 查看当前 SSL VPN 被锁定的用户列表，其中包含第一次尝试登录时间、最后一次尝试登录时间、登录失败次数、所属 VDOM ID。

```bash
FortiGate # diagnose vpn ssl blocklist list
|id|addr|status|first-attempt|last-attempt|fail-count|vfid|
|1|192.168.100.177|locked|2024-12-27 15:51:07|2024-12-27 15:51:11|2|0|
Total 1 entry.
```

2. 通过 FortiView 页面“按认证失败排名”可以看到每个用户在特定时间范围内的尝试登录失败次数。



3. 通过如下 CLI 命令可以解锁已被锁定的 SSL VPN 用户。

```bash
FortiGate # diagnose vpn ssl blocklist del <all|vfid|addr>
```

- `all`：解锁所有被锁定的 SSL VPN 用户。
- `vfid`：解锁特定 VDOM ID 下被锁定的所有 SSL VPN 用户。
- `addr`：解锁特定 IP 被锁定的 SSL VPN 用户。

4. 通过 IP 地址的形式解锁 SSL VPN 用户，可以看到 IP 192.168.100.177 被锁定的用户被解锁。

```bash
FortiGate # diagnose vpn ssl blocklist del 192.168.100.177

FortiGate # diagnose vpn ssl blocklist list
|id|addr|status|first-attempt|last-attempt|fail-count|vfid|
```

DTLS

默认配置下，SSL VPN 使用 TCP 来封装 TLS，当 SSL VPN 封装的业务数据为 TCP 时，就形成了 TCP-in-TCP 流量。这种情况下性能可能比预期的要慢。

Datagram Transport Layer Security (DTLS) 允许 SSL VPN 使用 TLS 加密流量，并使用 UDP 作为传输层，而不是 TCP。DTLS 具有与 SSL 相同的安全性，但使用 UDP 而不是 TCP。这可以显著提高性能，避免了在 TCP-in-TCP 中可能发生的重传问题。

配置步骤

FortiGate

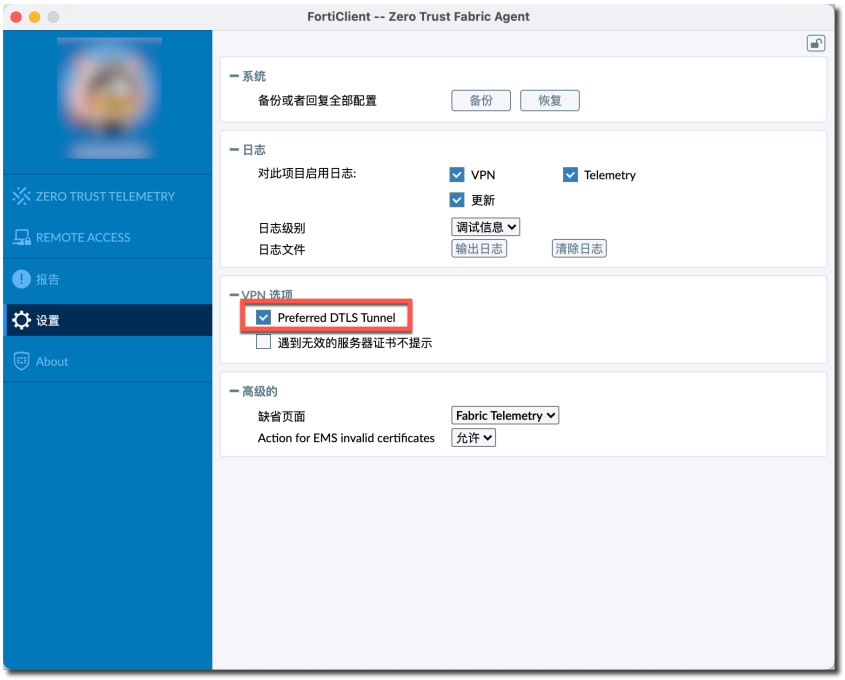
通过如下 CLI 开启 SSL VPN 的 DTLS。

```
config vpn ssl settings
    set dtls-tunnel enable
end
```

FortiClient

对应的 FortiClient 也需要开启 DTLS 功能。

> **重要**
>
> 如果 FortiClient 连接到了 EMS Server，这里的配置是无法修改的，需要管理员在 EMS Server 端进行修改。如果 FortiClient 未连接 EMS Server，或使用的是 FortiClient VPN only 版本，需要点击 FortiClient 界面的🔒图标解锁后，才可以继续配置。



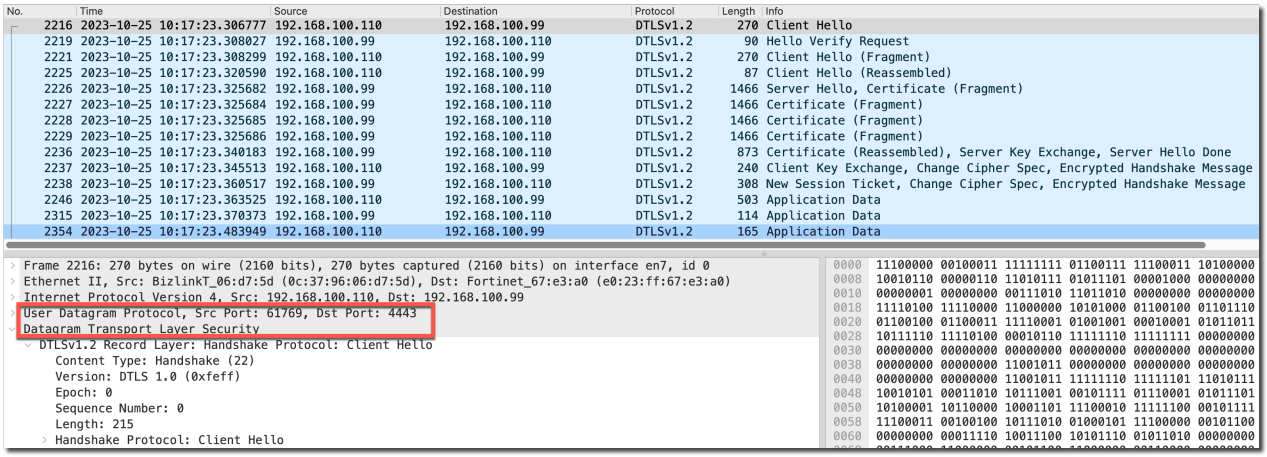
- 开启“Preferred DTLS Tunnel”：FortiClient 将在 FortiGate 上启用并成功建立隧道时使用 DTLS。如果在 FortiGate 上未启用或隧道建立不成功，则使用 TLS。DTLS 隧道使用 UDP 而不是 TCP，可以提高 VPN 的吞吐量。
- 关闭“Preferred DTLS Tunnel”：当禁用时，FortiClient 将使用 TLS，即使在 FortiGate 上启用了 DTLS。

结果验证

1. FortiClient 连接 FortiGate 的 SSL VPN 服务，查看 FortiGate 的 sslvpnd 进程 debug 信息，可以看到 SSL VPN 连接使用 DTLS 建立。

```
[321:root:5]DTLS established: DTLSv1 ECDHE-RSA-AES256-GCM-SHA384 from 192.168.100.110
```

2. 抓包查看 SSL VPN 协商与业务流量，可以看到 SSL VPN 协商与后续业务流量均使用 DTLS 封装。


No. Time Source Destination Protocol Length Info
2216 2023-10-25 10:17:23.306777 192.168.100.110 192.168.100.99 DTLSv1.2 270 Client Hello
2219 2023-10-25 10:17:23.308027 192.168.100.99 192.168.100.110 DTLSv1.2 90 Hello Verify Request
2221 2023-10-25 10:17:23.308299 192.168.100.110 192.168.100.99 DTLSv1.2 270 Client Hello (Fragment)
2225 2023-10-25 10:17:23.320590 192.168.100.110 192.168.100.99 DTLSv1.2 87 Client Hello (Reassembled)
2226 2023-10-25 10:17:23.325682 192.168.100.99 192.168.100.110 DTLSv1.2 1466 Server Hello, Certificate (Fragment)
2227 2023-10-25 10:17:23.325684 192.168.100.99 192.168.100.110 DTLSv1.2 1466 Certificate (Fragment)
2228 2023-10-25 10:17:23.325685 192.168.100.99 192.168.100.110 DTLSv1.2 1466 Certificate (Fragment)
2229 2023-10-25 10:17:23.325686 192.168.100.99 192.168.100.110 DTLSv1.2 1466 Certificate (Fragment)
2236 2023-10-25 10:17:23.340183 192.168.100.99 192.168.100.110 DTLSv1.2 873 Certificate (Reassembled), Server Key Exchange, Server Hello Done
2237 2023-10-25 10:17:23.345513 192.168.100.110 192.168.100.99 DTLSv1.2 240 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2238 2023-10-25 10:17:23.360517 192.168.100.99 192.168.100.110 DTLSv1.2 308 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2246 2023-10-25 10:17:23.363525 192.168.100.110 192.168.100.99 DTLSv1.2 503 Application Data
2315 2023-10-25 10:17:23.370373 192.168.100.99 192.168.100.110 DTLSv1.2 114 Application Data
2354 2023-10-25 10:17:23.483949 192.168.100.110 192.168.100.99 DTLSv1.2 165 Application Data
Frame 2216: 270 bytes on wire (2160 bits), 270 bytes captured (2160 bits) on interface en7, id 0
Ethernet II, Src: BizlinkT_06:d7:5d (0c:37:96:06:d7:5d), Dst: Fortinet_67:e3:a0 (e0:23:ff:67:e3:a0)
Internet Protocol Version 4, Src: 192.168.100.110, Dst: 192.168.100.99
User Datagram Protocol, Src Port: 61769, Dst Port: 4443
Datagram Transport Layer Security
DTLSv1.2 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: DTLS 1.0 (0xfeff)
Epoch: 0
Sequence Number: 0
Length: 215
Handshake Protocol: Client Hello
0000 11100000 00100011 11111111 01100111 11100011 10100000
0008 10010110 00000110 11010111 01011101 00001000 00000000
0010 00000001 00000000 00111010 11011010 00000000 00000000
0018 11110100 11110000 11000000 10101000 01100100 01101110
0020 01100100 01100011 11110001 01001001 00010001 01011011
0028 10111110 11110100 00010110 11111110 11111111 00000000
0030 00000000 00000000 00000000 00000000 00000000 00000000
0038 00000000 00000000 11001011 00000000 00000000 00000000
0040 00000000 00000000 11001011 11111110 11111101 11010111
0048 10010101 00011010 10111001 00101111 01110001 01011101
0050 10100001 10110000 10001101 11100010 11111100 00101111
0058 11100011 00100100 10111010 01000101 11100000 00101100
0060 00000000 00011110 10011100 10101110 01011010 00000000

SSL VPN 双因子认证

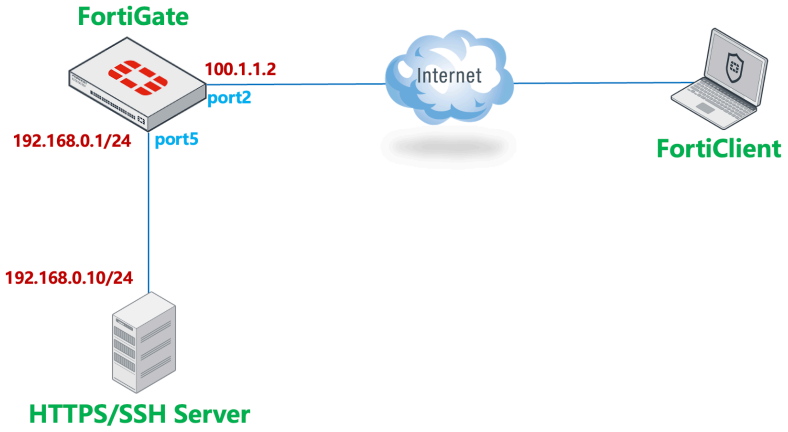
- FortiToken 双因子认证

FortiToken 双因子认证

网络需求

在外移动办公的工作人员需要通过 SSL VPN 隧道模式拨入到公司内网来对内网主机进行访问。

网络拓扑



配置步骤

FortiToken 配置

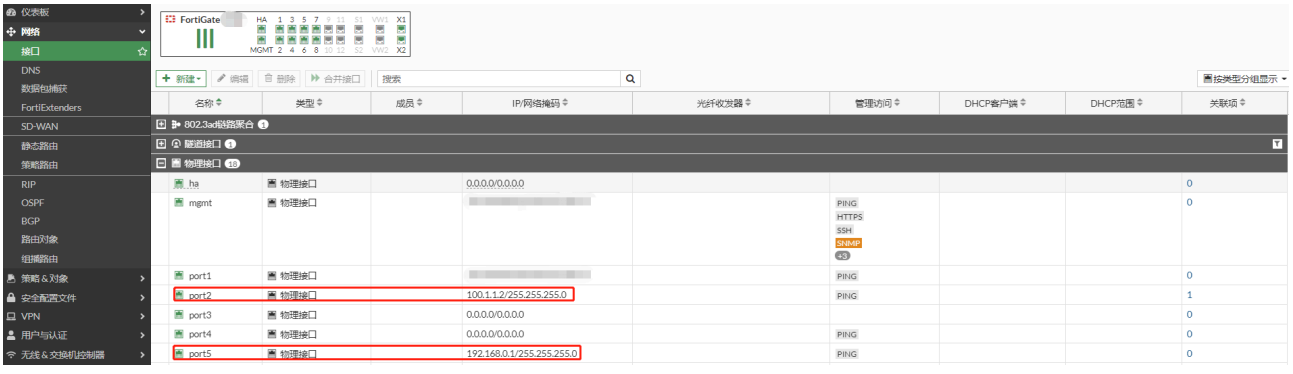
请参考如下章节：

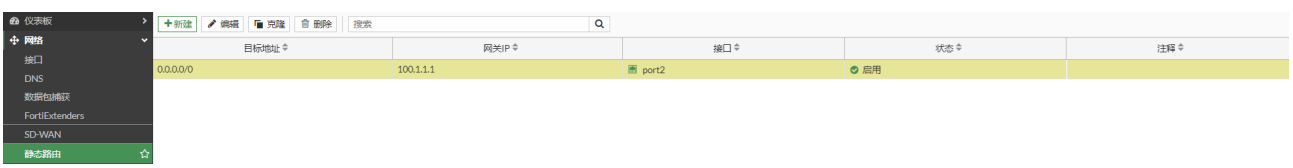
- 用户与认证 → FortiToken 认证 → FortiToken Mobile 在 FortiGate 上的激活 ↗
- 用户与认证 → FortiToken 认证 → FortiToken Mobile APP 软件下载与安装 ↗
- 用户与认证 → FortiToken 认证 → FortiToken Mobile 与用户进行绑定 ↗

配置 SSL VPN

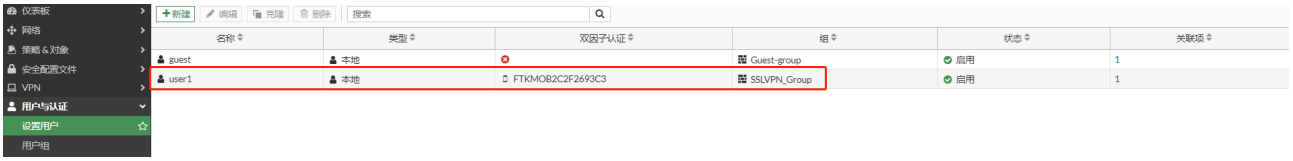
1.基本配置

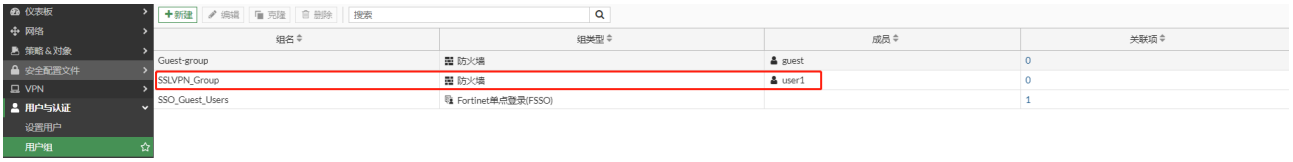
配置接口 IP 和路由





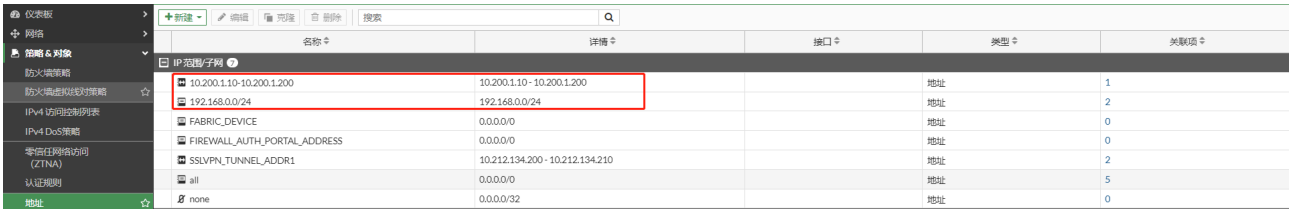
2.创建用户，开启双因子认证，并将用户加入到用户组



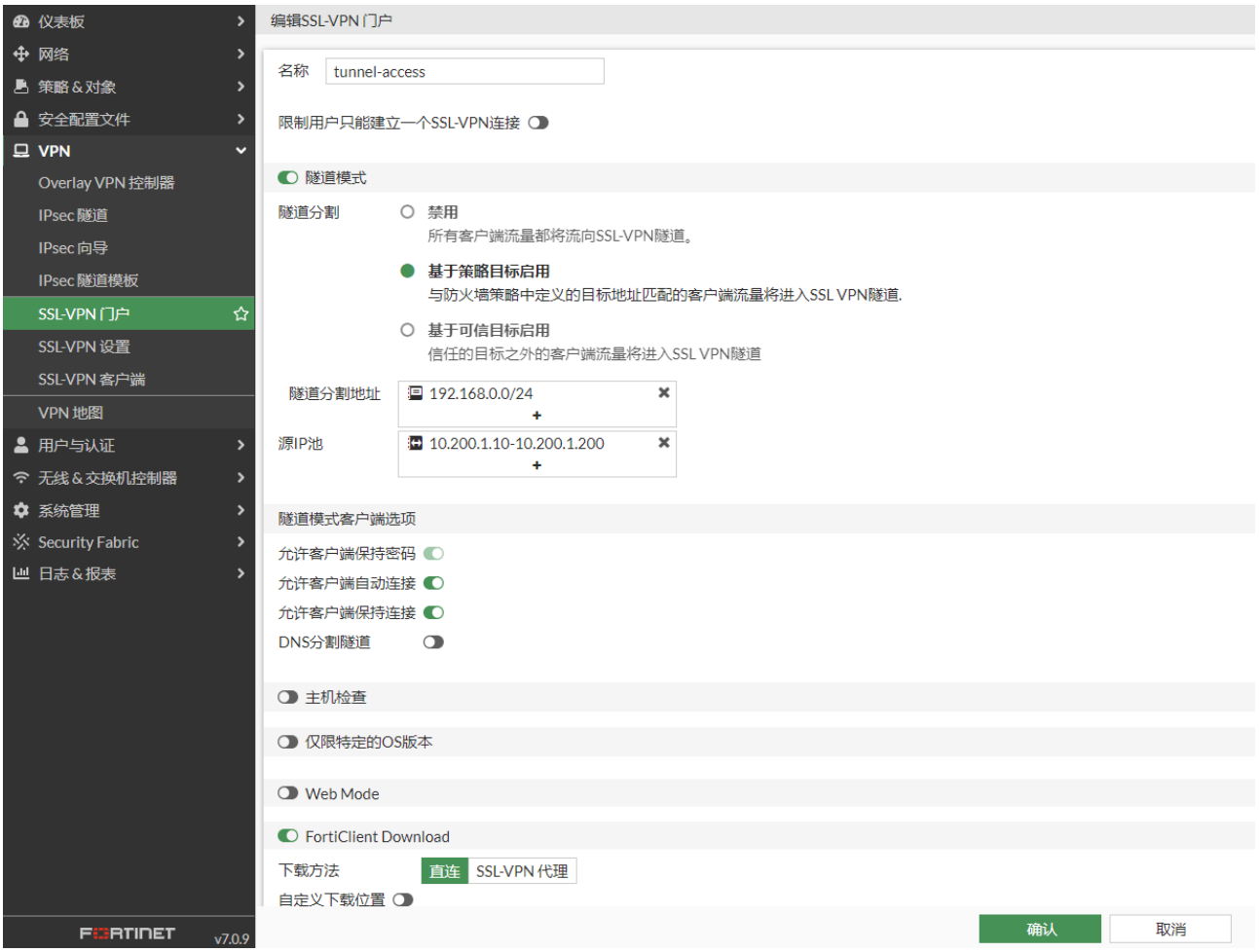


3.配置地址对象

10.200.1.10-10.200.1.200 作为 SSLVPN 地址池；192.168.0.0/24 作为需要访问的内网地址段。

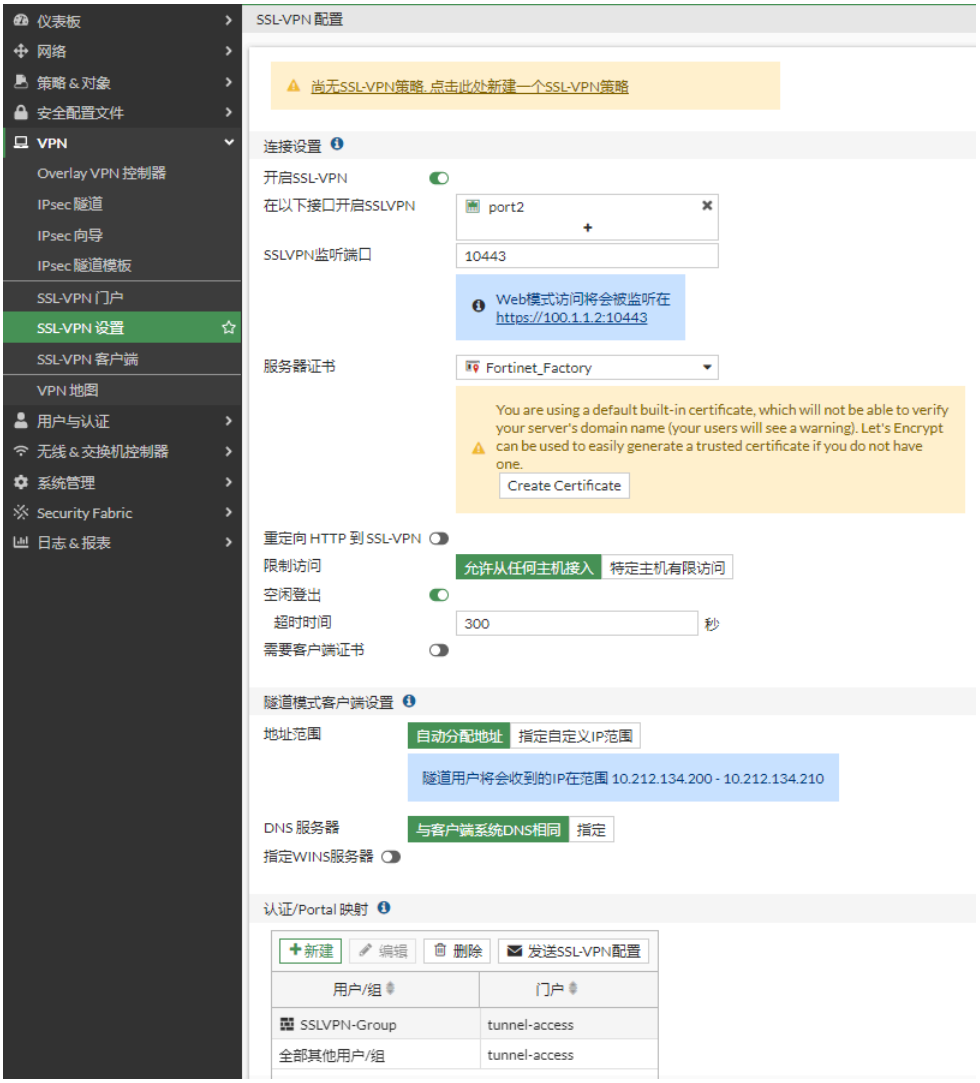


4.配置 SSL - VPN 门户



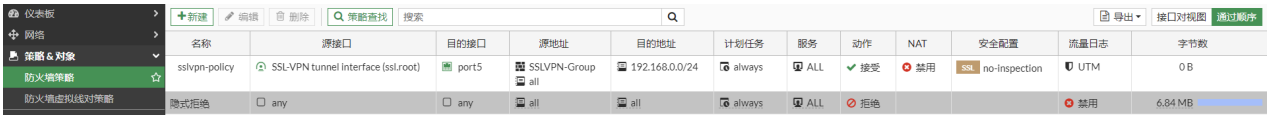
5.配置 SSLVPN

设置提供 SSLVPN 服务的接口和端口，将 SSLVPN-Group 用户组和 tunnel-access 门户关联；全部其他用户/组也要关联一个门户，用于没有配置关联的用户/用户组访问。



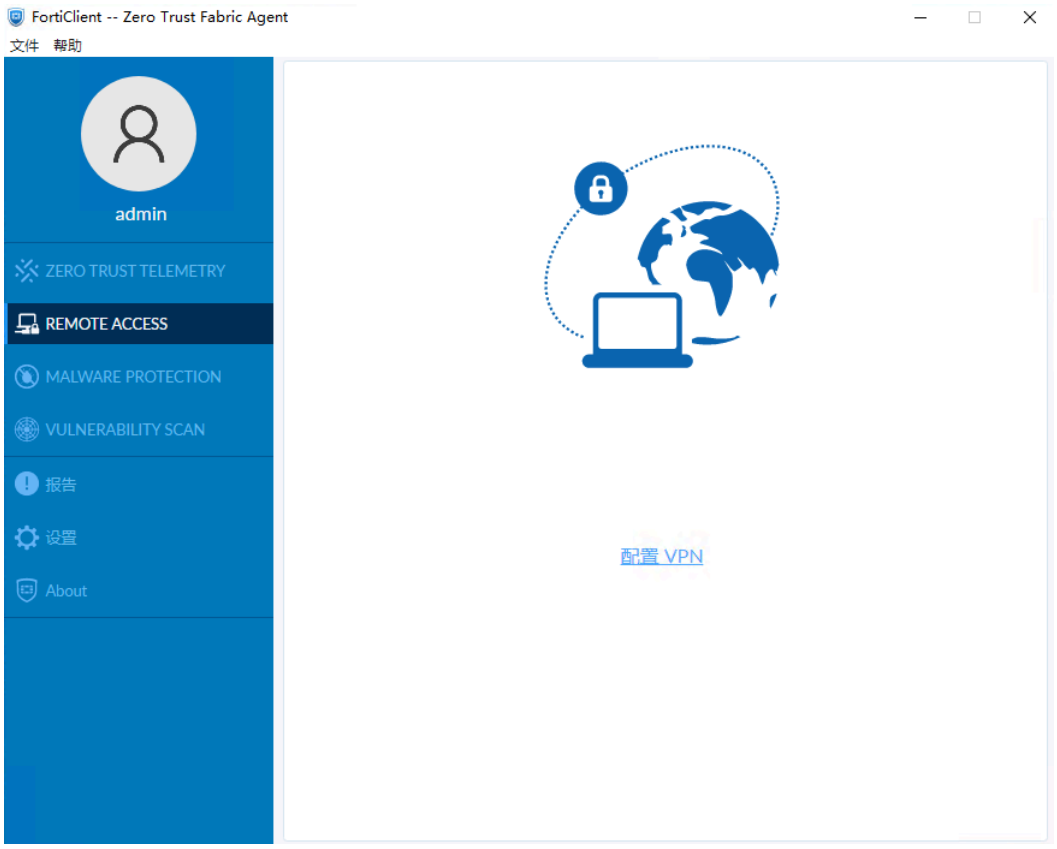
6. 创建策略

当客户端 SSLVPN 拨号成功后，将会使用获取的地址（10.200.1.10-10.200.1.200）访问内部主机，因此内部网络需要增加到 10.200.1.0/24 网段的回程路由指向 FortiGate 或者 可以在策略中开启 NAT，那么源地址将被转换为 FortiGate 接口地址，则不用考虑回程路由。

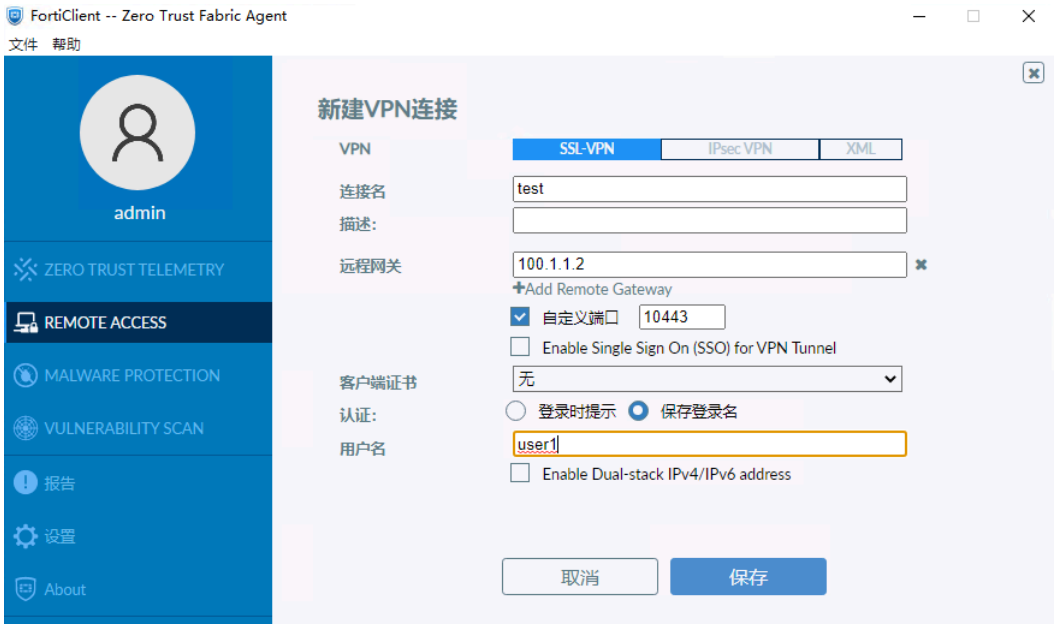


配置 FortiClient

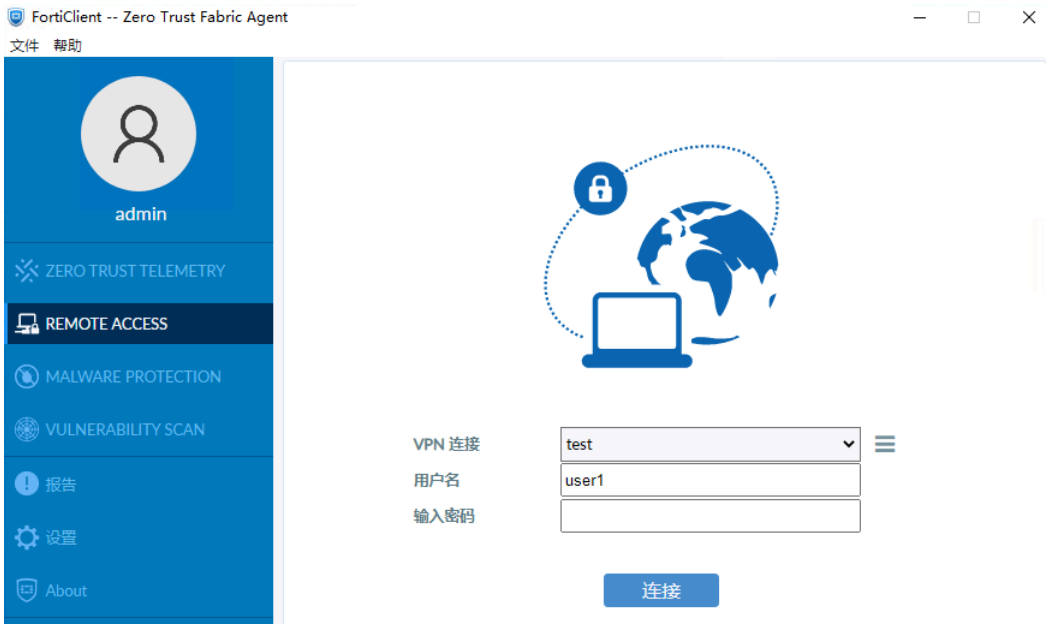
1. 选择“Remote Access”，点击“配置 VPN”。



2. 选择 SSL VPN，设置连接名，远程网关，SSLVPN 端口以及用户名，然后点击保存。



3. 配置完成。

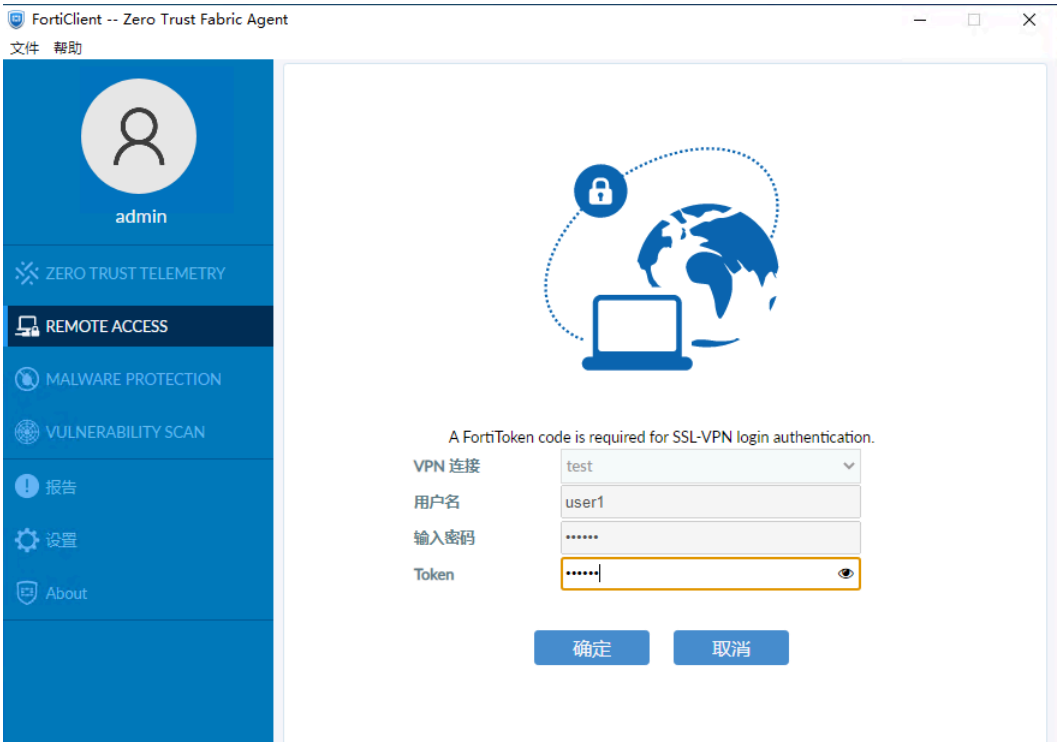


业务测试

1. 在 FortiClient 中输入密码，并点击连接，在弹出的窗口中，点击“是”，信任 SSLVPN 证书



2. 输入 token 码



3. 连接成功



查看终端获取到的地址

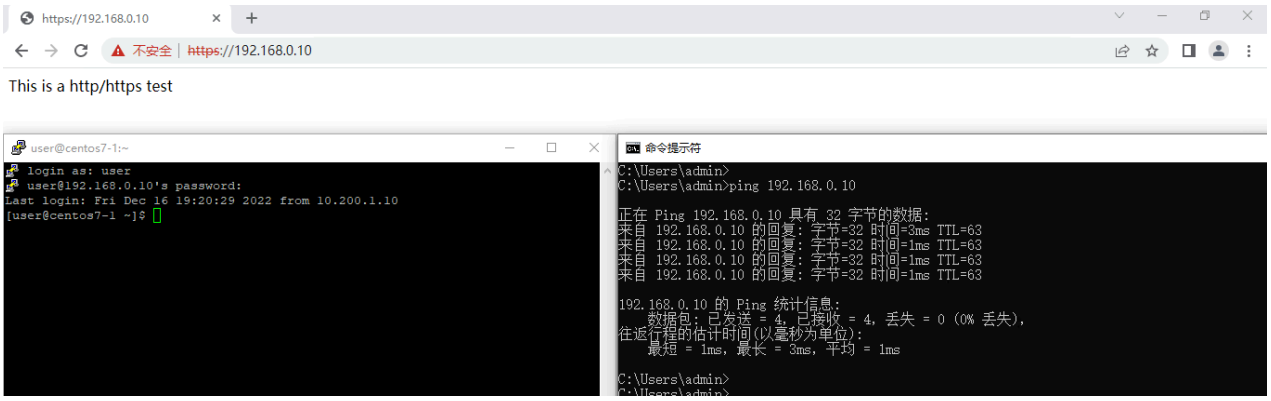
```
C:\Users\admin>ipconfig/all
以太网适配器 以太网 2:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Fortinet SSL VPN Virtual Ethernet Adapter
   物理地址. . . . . . . . . . . . . : 00-09-0F-AA-00-01
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::9124:e3aa:1ace:74d2%7(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.200.1.10(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.255
```

```
   默认网关. . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 100665615
   DHCPv6 客户端 DUID  . . . . . . . . : 00-01-00-01-29-1C-CC-4A-00-0C-29-A2-90-E0
   DNS 服务器  . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . . : 已启用
```

查看终端获取到的路由

```
C:\Users\admin>route print
===========================================================================
接口列表
  6...00 09 0f fe 00 01 ......Fortinet Virtual Ethernet Adapter (NDIS 6.30)
  7...00 09 0f aa 00 01 ......Fortinet SSL VPN Virtual Ethernet Adapter
 10...00 0c 29 a2 90 e0 ......Intel(R) 82574L Gigabit Network Connection
  5...00 0c 29 a2 90 ea ......Intel(R) 82574L Gigabit Network Connection #2
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0         10.1.1.1         10.1.1.5    281
         10.1.1.0    255.255.255.0            在链路上          10.1.1.5    281
         10.1.1.5  255.255.255.255            在链路上          10.1.1.5    281
       10.1.1.255  255.255.255.255            在链路上          10.1.1.5    281
      10.200.1.10  255.255.255.255            在链路上        10.200.1.10    257
        100.1.1.2  255.255.255.255         10.1.1.1         10.1.1.5     25
        127.0.0.0        255.0.0.0            在链路上          127.0.0.1    331
        127.0.0.1  255.255.255.255            在链路上          127.0.0.1    331
  127.255.255.255  255.255.255.255            在链路上          127.0.0.1    331
      192.168.0.0    255.255.255.0      10.200.1.11      10.200.1.10      1
```

4. 访问内网主机 PING，HTTPS，SSH 服务都正常



SSL VPN 常见问题

- SSL VPN 连接后断开

SSL VPN 连接后断开

问题现象

1. FortiClient 在连接 SSL VPN 成功后，马上断开连接。

2. FortiClient 所在 PC 的真实公网 IP 为 121.211.6.3。

3. 但在 `diagnose debug application sslvpn -1` 的 debug 信息中，FortiClient 的公网 IP 却显示显示为 221.211.135.16（`host[221.211.135.16]`）。

```bash
diagnose debug application sslvpn -1
diagnose debug

[12260:root:27]deconstruct_session_id:494 decode session id ok, user=[jszn], group=[sslvpn-group],authserver=[],portal=
[tunnel-access],host[221.211.135.16],realm=[],csrf_token=[467291DF2B61CBF51DD42F8B423FFB99],idx=1,auth=1,
sid=3471a060,login=1755833636,access=1755833636,saml_logout_url=no,pip=201.218.89.162,grp_info=[QOa1Sc],rmt_grp_info=[]
[12260:root:27]session removed s: 0×7f8b9fd000 (root)
```

4. 正常的 debug 信息应为 `host[121.211.6.3]`：

```
[12260:root:27]deconstruct_session_id:494 decode session id ok, user=[jszn], group=[sslvpn-group],authserver=[],portal=
[tunnel-access],host[121.211.6.3],realm=[],csrf_token=[467291DF2B61CBF51DD42F8B423FFB99],idx=1,auth=1,
sid=3471a060,login=1755833636,access=1755833636,saml_logout_url=no,pip=205.218.89.162,grp_info=[QOa1Sc],rmt_grp_info=[]
[12260:root:27]session removed s: 0×7f8b9fd000 (root)
```

5. `diagnose debug application sslvpn -1` 的 debug 信息中，SSL VPN 连接中断的原因是 `source ip check failed`。

```bash
diagnose debug application sslvpn -1
diagnose debug

......
[12261:root:27]get_shm_session:1368 source ip check failed
......
```

问题原因

SSL VPN 的 `source ip check failed` 会导致身份验证失败，并在成功连接后几秒内终止 SSL VPN 连接。此情况可能出现在客户端使用 NAT、代理服务或负载均衡等技术，这些技术可能会在 SSL VPN 会话建立过程中更改客户端的实际公网 IP。

解决方法

允许 SSL VPN 认证会话令牌中的源 IP 与客户端实际建立连接的源 IP 地址不同的情况下，仍应能成功认证。执行以下命令禁用 SSL VPN 连接的源 IP 地址验证：

```bash
config vpn ssl settings
    set auth-session-check-source-ip disable
end
```

SSL VPN 迁移到 IPSec VPN

重要

Fortinet 建议在升级到 FortiOS 7.6.3 及以上版本之前，将远程接入从 SSL VPN Tunnel 模式迁移到 IPSec VPN。

介绍

迁移背景

- 安全性对比
- IKEv1 还是 IKEv2?
- 隧道协议与封装方式

迁移基础

- 设计考虑事项
  - 认证方式
  - 多用户组
  - 全隧道与分离隧道
  - 客户端地址分配
  - Split DNS 与 DNS 后缀
  - 策略配置
  - FortiClient 或终端配置
  - 先迁移 VPN 还是先升级?
- FortiOS SSL VPN 到拨号 VPN 的迁移
  - 网络拓扑
  - 第一部分：识别用户认证方式
  - 第二部分：使用 VPN Wizard 配置 IPSec 隧道
  - 自定义 IPSec 隧道设置
- FortiClient 终端配置迁移

介绍

VPN 技术用于让用户、设备和站点在不安全的互联网环境中安全互联。SSL VPN 和 IPSec VPN 都是使用广泛、配置和部署相对简单的 VPN 技术。

不同 VPN 技术有各自的优势和常见使用场景。SSL VPN 通常用于单个用户和终端的安全远程接入。它一般易于部署，并且连接基于 TLS 的 TCP/443，运营商较少限制这类连接。SSL VPN 同时提供 Tunnel 模式和 Web 模式，可以分别用于安装客户端和无客户端访问的部署方式。

IPSec VPN 通常用于站点到站点连接，尤其适合通过 ADVPN 搭建多站点 Hub-Spoke 架构。ADVPN 可以在站点之间自动建立隧道，结合动态路由，并通过 FortiManager 等编排工具批量下发配置，从而简化复杂的多站点部署。

在较小规模的远程用户接入场景中，IPSec VPN 同样可以通过拨号 VPN 支持远程用户。与 SSL VPN 类似，当 IPSec 使用 IKEv2 时，可以配置 TCP 作为传输方式，通过 TCP/443 进行 IKE 协商，并将 ESP 报文封装在 TCP 头部中。协议、加密算法和认证方式都可以按企业需求定制。

作为 VPN 的替代方案，尤其是替代基于 Web 的 SSL VPN 访问，也可以使用 ZTNA。ZTNA 在终端和 Zero Trust Application Gateway 之间通过 TLS 建立连接。零信任方式默认不信任设备，直到设备通过必要的安全状态检查，例如客户端证书校验和漏洞扫描。

本文讨论 SSL VPN 和 IPSec VPN 的差异，以及从 SSL VPN 迁移到 IPSec VPN 时需要考虑的事项，并给出常见 SSL VPN 使用场景迁移到 IPSec VPN 的步骤。

迁移背景

- 安全性对比
- IKEv1 还是 IKEv2?
- 隧道协议与封装方式

安全性对比

SSL VPN 通过 TLS 提供安全能力，主要包括：

- 加密客户端与 VPN 网关之间传输的数据，确保传输中的数据未被篡改。
- 提供客户端和服务器之间相互验证身份的认证机制。
- 使用 Diffie-Hellman 等安全密钥交换方式，在客户端与服务器之间建立共享密钥。
- 使用 X.509 证书认证服务器，必要时也可以认证客户端。

IPSec 通过 ISAKMP（Internet Security Association and Key Exchange Management Protocol）框架提供安全能力，主要包括：

- 使用 IKE（Internet Key Exchange）协议协商安全通信参数，生成和管理密钥，并在通信双方之间建立 SA（Security Association）。
- 使用 IKE 协商出的对称加密算法加密数据包，例如 AES、3DES、CHACHA。密钥由 IKE 生成。
- 使用 HMAC（Hash-based Message Authentication Code）验证消息完整性，确保传输中的数据未被篡改。
- 在 SA 中指定密钥生命周期和其他安全参数。

与 SSL VPN 由客户端和服务器协商 Cipher Suite 不同，IPSec 可以更灵活地选择加密算法、哈希算法以及密钥生命周期。

IKEv1 还是 IKEv2?

FortiGate 同时支持 IKEv1 和 IKEv2，两者配置方式相近。IKEv2 底层协议更简洁，协商 SA 时需要的消息交互比 IKEv1 更少。

两者的主要差异：

- IKEv1 使用 XAuth（Extended Authentication）进行用户认证。
- IKEv2 使用 EAP（Extensible Authentication Protocol）进行用户认证。
- IKEv2 支持使用 TCP 作为传输方式，使 IPSec 可以通过 TCP 协商，并将 ESP 报文封装在 TCP 中，例如使用 TCP/443。

IKEv1 使用广泛、行为固定，通常更容易排障。IKEv2 提供更多灵活性，但也会带来更多配置组合，排障时变量更多。

> **重要**
>
> 从 FortiClient 7.4.4 开始，客户端不再支持 IKEv1。如果计划部署 FortiClient 7.4.4 及以上版本，应选择 IKEv2。

隧道协议与封装方式

SSL VPN 使用 TLS 协议建立隧道。

Fortinet IPSec VPN 支持以下隧道和封装方式：

- 原生 ESP。
- UDP 封装。
- 使用 Fortinet 私有扩展的 TCP 封装。
- 使用 RFC 8229 的 TCP 封装。

ESP 不封装时直接使用 IP Protocol 50。ESP 封装在 UDP 中时，通常使用 UDP/500 和 UDP/4500 进行 NAT Traversal，这也是拨号 IPSec VPN 的常见方式。

在远程接入 VPN 中，FortiGate 作为拨号 IPSec Server，FortiClient 作为终端客户端时，建议使用 IKEv2，并将 IPSec 隧道配置为使用 TCP 传输，通常使用 TCP/443。这样可以让 ESP 报文封装在 TCP 中，穿越直接 IPSec 流量被运营商网络阻断或被 CGNAT 影响的环境。

该能力要求：

- IPSec 使用 IKEv2。
- FortiClient 版本为 7.4.1 或更高。
- FortiClient 侧也配置为使用 TCP 传输。

站点到站点 IPSec 中，UDP 端口可以自定义。IKEv2 站点到站点 IPSec 中，TCP 端口也可以自定义，并支持将 ESP 报文封装到 TCP 头部中。

迁移基础

理解 SSL VPN 和 IPSec VPN 的差异后，即可开始规划迁移。本部分包含：

- 设计考虑事项。
- FortiOS SSL VPN 到拨号 VPN 的迁移。
- FortiClient 终端配置迁移。

设计考虑事项

- 认证方式
- 多用户组
- 全隧道与分离隧道
- 客户端地址分配
- Split DNS 与 DNS 后缀
- 策略配置
- FortiClient 或终端配置
- 先迁移 VPN 还是先升级?

FortiOS SSL VPN 到拨号 VPN 的迁移

- 网络拓扑
- 第一部分：识别用户认证方式

  本地用户认证　基于 LDAP 的用户认证　基于 RADIUS 的用户认证　基于 SAML 的用户认证
- 第二部分：使用 VPN Wizard 配置 IPSec 隧道
- 自定义 IPSec 隧道设置

FortiClient 终端配置迁移

FortiClient 终端配置迁移

FortiClient EMS 上从 SSL VPN 迁移到 IPSec，需要与 FortiGate 配置同步进行，因为 IPSec 设置必须在 FortiGate（VPN Server）和 FortiClient（VPN Client）两侧匹配。

FortiClient EMS 通过 Remote Access Endpoint Profile 配置 VPN，可以配置 SSL VPN、IPSec VPN 或两者同时存在。示例使用 FortiClient EMS 和 FortiClient 7.4.4。

开始时，在 FortiClient EMS 的 Endpoint Profiles 下添加 Remote Access Profile。添加新的 Remote Access Profile 后，在同一个 Remote Access Profile 中的 VPN Tunnels 下添加隧道。

通过 FortiClient EMS Remote Access Endpoint Profile 迁移

1. 在 FortiClient EMS 中，进入 Endpoint Profiles。
2. 选择需要的 Profile 类型，点击 Add。
3. 点击 Add Profile，创建 Windows、macOS 和 Linux Profile。
4. 在 VPN Tunnel 下，点击 Add Tunnel → Manual，并完成 Basic Settings。

基础设置

配置项	说明
Name	隧道名称
Type	选择 IPSec VPN。FortiClient EMS 7.4.3 及以上版本中，默认类型为 IPSec
Remote Gateway	FortiClient 用于连接 FortiGate 的 IP 地址或 FQDN；如果 FortiGate 使用 VPN Wizard 创建，该项对应 Wizard Authentication 步骤中的 Incoming interface 地址，通常与 SSL VPN Remote Gateway 相同
Authentication Method	可选 Local Certificate、Pre Shared Key、Smart Card Certificate 或 Local Store Certificate

FortiGate VPN Wizard 允许使用预共享密钥或 Signature。FortiGate 侧使用预共享密钥时，FortiClient EMS 的 Pre Shared Key 必须填写相同值。

如需使用 Signature 认证，应选择符合企业要求的证书方式，并确保证书 CA 与 FortiGate VPN Wizard Authentication 步骤中配置的 Peer Certificate CA 匹配。

VPN 设置

在 Basic Settings 的 VPN Settings 中配置 IKE version、Mode 和 Options。这些设置必须与 FortiGate 一致。

配置项	说明
IKE	FortiGate VPN Wizard 默认使用 Version 2。通常建议使用 IKEv2，因为可以使用 TCP/443；FortiClient 7.4.4 及以后不再支持 IKEv1
Address Mode（IKEv1）	IKE 为 Version 1 时，可选择 Aggressive 或 Main mode
Address Assignment	默认和建议使用 Mode Config，这也是 FortiGate IPSec Wizard 默认方式
Encapsulation	可选择 IKE UDP Port 或 IPSec over TCP。Auto 推荐使用，因为 UDP 无法连接时会自动从 UDP 回退到 TCP

Phase 1

在 FortiGate 上可通过以下命令查看 VPN Wizard 自动配置的 Phase 1 参数：

```bash
show full vpn ipsec phase1-interface <tunnel-name>
```

FortiClient EMS 的 Phase 1 设置必须与 FortiGate 上的 Phase 1 设置匹配。

配置项	说明
IKE Proposal	选择用于保护 FortiClient 与 FortiGate 协商的加密和认证算法。至少有一组加密/认证算法与 FortiGate 匹配
DH Groups	必须与 FortiGate 上选择的 DH Group 匹配
Key Life	IKE 密钥生命周期，范围 120 到 172800 秒。FortiGate VPN Wizard 默认 86400 秒
Local ID	对应 FortiGate VPN 的 Peer ID。FortiGate VPN Wizard 默认留空
Network ID	FortiGate 启用 Network Overlay 时，对应 FortiGate VPN 的 Network ID
EAP Authentication Method	RADIUS 认证使用 EAP-MSCHAPv2；LDAP 认证使用 EAP-TTLS。该设置在 FortiClient EMS 7.4.4 中新增
Enable IKE Fragmentation	建议启用。当证书认证、Proposal 较多或 IKE 参数较多时，IKE 消息可能超过路径 MTU

FortiGate VPN Wizard 默认 Phase 1 Proposal 包括：

- AES128-SHA256。
- AES256-SHA256。
- AES128-SHA1。
- AES256-SHA1。

> **重要**
>
> 不建议使用 SHA1 作为认证算法，建议使用 SHA256。
>
> DH Group 5 和 14 不推荐，建议考虑使用 20 或 21。

Phase 1 也可以配置 DPD。FortiClient EMS GUI 中不提供 DPD 配置，但可以通过 XML 配置。

Phase 2

在 FortiGate 上可通过以下命令查看 VPN Wizard 自动配置的 Phase 2 参数：

```bash
show full vpn ipsec phase2-interface <tunnel-name>
```

FortiClient EMS 的 Phase 2 设置必须与 FortiGate 上的 Phase 2 设置匹配。

配置项	说明
IPSec Proposal	选择用于保护 IPSec Peer 之间数据传输的加密和认证算法。至少有一组与 FortiClient 和 FortiGate 同时匹配
DH groups	与 FortiGate 上的 DH Group 匹配
Key Life	Phase 2 密钥生命周期。默认按秒，也可以按 KB 或两者同时配置；两者同时配置时，先达到的限制生效。默认值为 43200 秒，与 FortiGate VPN Wizard 默认值一致
Replay Detection	启用后，FortiGate 检查已接收数据包并丢弃乱序到达的重复报文。FortiClient EMS 和 FortiGate 默认启用
PFS	启用后，隧道建立和 Phase 2 密钥过期后会强制进行新的 DH 密钥交换。FortiClient EMS 和 FortiGate 默认启用

FortiGate VPN Wizard 默认 Phase 2 Proposal 包括：

- AES128-SHA1。
- AES256-SHA1。
- AES128-SHA256。
- AES256-SHA256。
- AES128GCM。
- AES256GCM。

不建议使用 SHA1 作为认证算法，建议使用 SHA256。DH Group 5 和 14 不推荐，建议考虑使用 20 或 21。

高级设置

Advanced Settings 中可以配置 Save Password、Auto-Connect 和 Always Up。这些选项会显示在 FortiClient GUI 中，用于自动连接 VPN 隧道并在网络中断后恢复连接。

如果在 FortiClient EMS 中包含这些设置，也应在 FortiGate VPN Wizard 的 Client Options 步骤中配置对应选项。

完成后：

1. 点击 Save 保存更改。
2. 将 Profile 推送到 FortiClient 终端。
3. 在终端打开 FortiClient，进入 Remote Access 页面确认配置已经下发。

> **注**
>
> 用户必须在 FortiClient GUI 中选择 Save Password、Auto-Connect 和 Always Up 后，这些选项才会实际启用。

设计考虑事项

典型 SSL VPN Tunnel 模式中，远程用户从互联网访问 FortiGate WAN 接口的 TCP/443，认证成功后建立 SSL VPN Tunnel。终端与 FortiGate WAN 接口之间的流量通过 TLS 加密，用户可以访问哪些内部资源由对应用户组的防火墙策略决定。

迁移到拨号 IPSec VPN 后，拓扑通常保持一致。远程用户仍从互联网访问 FortiGate WAN 接口的 TCP/443，但使用 IKEv2 进行 IPSec 接入，并通过所选认证方式完成身份认证。隧道建立后，流量通过 ISAKMP/IPSec 保护，ESP 报文封装在 TCP 头部中。用户访问内部资源的权限仍由用户组对应的防火墙策略控制。

因此，从 SSL VPN 迁移到 IPSec VPN 时，通常不需要改变网络拓扑，也可以继续使用 TCP/443。

> **重要**
>
> 通常建议使用 IKEv2，因为 IKEv2 可以使用 TCP/443。
>
> 从 FortiClient 7.4.4 开始，客户端不再支持 IKEv1。只有在不需要 IKE over TCP，且不计划部署 FortiClient 7.4.4 及以上版本时，才考虑继续使用 IKEv1。
>
> FortiClient 7.4.4 不支持 IPv6。如需 IPv6，请使用 FortiClient 7.4.6 或更高版本。

- 认证方式
- 多用户组
- 全隧道与分离隧道
- 客户端地址分配
- Split DNS 与 DNS 后缀
- 策略配置
- FortiClient 或终端配置
- 先迁移 VPN 还是先升级?

认证方式

建立 SSL VPN 隧道时，用户必须认证到与 SSL VPN User Group to Portal Mapping 关联的用户组。FortiGate 支持的 SSL VPN 认证方式包括：

SSL VPN 认证方式	要求
PKI	至少配置一种用户认证方式
Local	至少配置一种用户认证方式
LDAP	至少配置一种用户认证方式
RADIUS	至少配置一种用户认证方式
SAML	至少配置一种用户认证方式
Two-factor authentication	可选

FortiToken 双因素认证可以与 Local、LDAP、RADIUS 或 SAML 组合使用。客户端证书双因素认证也支持。

IPSec 隧道中，用户可以通过预共享密钥、证书、IKEv1 XAuth 或 IKEv2 EAP 认证。FortiGate 支持的 IPSec 认证方式包括：

IPSec 认证方式	IKE 版本	要求
Pre-shared key	IKEv1/IKEv2	必须配置一种网关认证方式
PKI（Signature）	IKEv1/IKEv2	必须配置一种网关认证方式
LDAP	IKEv1/IKEv2，IKEv2 需要 EAP-TTLS	可选用户认证方式
Local	IKEv1/IKEv2	可选用户认证方式
RADIUS	IKEv1/IKEv2	可选用户认证方式
SAML	IKEv2	可选用户认证方式
Two-factor authentication	IKEv1/IKEv2	可选

预共享密钥和 PKI 可以与其他用户认证方式组合。FortiToken 双因素认证可以与 Local、LDAP、RADIUS 或 SAML 组合使用，客户端证书双因素认证也支持。

迁移时，原 SSL VPN 的认证方式都可以迁移到 IPSec VPN。用户和用户组可以复用到新的 IPSec 配置中。管理员在创建 IPSec 隧道时必须选择预共享密钥或 PKI 证书，因为这是 IPSec 隧道的必选配置。

多用户组

SSL VPN 只使用一个 SSL VPN Settings 页面和一个 SSL VPN 接口。多个用户组可以映射到不同 Portal，并通过防火墙策略控制精细化访问权限。

IPSec VPN 中，一个拨号 VPN 隧道可以通过在 VPN 设置中定义用户组，或从防火墙策略继承用户组的方式支持一个或多个用户组。与 SSL VPN 不同，管理员也可以创建多个独立拨号 IPSec 隧道，用于承载当前不同 SSL VPN Tunnel Mode Web Portal 为各用户组提供的差异化功能。

如果在同一个 WAN 接口上配置多个拨号 IPSec 隧道，且隧道参数相近，则每个隧道都需要唯一标识，让拨号客户端发起连接时匹配到正确隧道：

- IKEv1 建议使用 Aggressive Mode，并通过 Phase 1 中的 Peer ID 区分。
- IKEv2 建议使用 Phase 1 中的 Network ID 区分。

> **重要**
>
> Network ID 不能在未纳管的独立 FortiClient 中配置。通过 FortiClient EMS 管理时，FortiClient EMS 7.2.6 及以上或 7.4.1 及以上支持配置 Network ID。

从 SSL VPN 迁移到 IPSec VPN 时，应根据用户组数量和原 Portal 差异选择用户组定义方式。

全隧道与分离隧道

Full tunneling 会强制所有远程用户流量经过 VPN。

Split tunneling 允许管理员指定哪些目的流量经过 VPN。

SSL VPN 和 IPSec VPN 都支持 Split tunneling。SSL VPN 默认根据防火墙策略中的目的地址启用 Split tunneling。IPSec 自定义配置默认关闭 Split tunneling，但通过 VPN Wizard 创建时默认启用。

启用 Split tunneling 后，必须配置需要包含或排除在隧道路由中的网络。

客户端地址分配

SSL VPN 从预定义或自定义 IP 地址范围中为客户端分配地址。

拨号 IPSec VPN 支持多种地址分配方式。建议使用 Mode Config，让 FortiGate 作为 IP 地址分配服务器。Mode Config 提供多种分配方式，例如手工地址范围，也可以与 DHCP Server 集成。

迁移到 IPSec VPN 后，客户端地址分配方式更灵活。建议使用 Mode Config，并根据实际需求选择对应的地址分配方式。

Split DNS 与 DNS 后缀

SSL VPN Tunnel 模式支持 Split DNS 和 DNS suffix 配置。

拨号 IPSec 隧道是否支持这些能力取决于 IKE 版本：

- IKEv1：支持 DNS 后缀配置，但需要在 Phase 1 中启用 `unity-support`。
- IKEv2：支持 Split DNS。

规划迁移时，应检查现有 SSL VPN 设置，确认是否依赖 Split DNS 或 DNS suffix，再选择满足需求的 IKE 版本。

策略配置

SSL VPN 使用单一 `ssl.root` 隧道接口作为防火墙策略源接口，控制终端客户端到内部资源的访问。用户组必须定义在策略中，用于控制允许访问内部资源的用户组。

IPSec VPN 会为每个 IPSec 隧道创建一个同名虚拟 VPN 接口。迁移后，应在防火墙策略中选择对应虚拟隧道接口作为源接口，控制终端客户端访问内部资源。

用户组可以定义在策略中，并由 VPN 隧道配置继承；也可以直接定义在各 IPSec 隧道配置中。

迁移时需要检查并调整防火墙策略，将原 `ssl.root` 相关策略迁移为 IPSec 隧道接口相关策略，并确认用户组配置位置与隧道认证方式一致。

FortiClient 或终端配置

SSL VPN Tunnel 模式需要终端安装 FortiClient。IPSec 隧道同样需要 FortiClient。

FortiClient 可以单独安装在终端上，也可以通过 FortiClient EMS 管理。建议使用 FortiClient EMS，因为它可以集中管理客户端配置，并支持更大规模部署。

基本 FortiClient SSL VPN 配置包括：

配置项	说明
Connection name	用于标识隧道的本地名称
Remote Gateway	FortiGate SSL VPN 接口地址
Port	FortiGate 监听端口，默认 TCP/443，可改为其他端口
Authentication	支持每次手工输入用户名密码，或保存登录信息
Single Sign-On	启用后，可使用内置浏览器或外部浏览器进行 SAML 认证
Client Certificate	SSL VPN Server 要求客户端证书时，FortiClient 必须提供证书

基本 FortiClient IPSec VPN 配置包括：

配置项	说明
Connection name	用于标识隧道的本地名称
Remote Gateway	FortiGate IPSec VPN 网关地址
Authentication Method	预共享密钥或 X.509 客户端证书
Authentication（XAuth or EAP）	支持每次手工输入用户名密码，或保存登录信息
Failover SSL VPN	仅 SSL VPN 冗余场景相关，否则设置为 None
Single Sign On	启用 SAML 认证，FortiClient 7.2.4 及以上支持
Advanced Settings	IKE 版本、IKEv1 Main/Aggressive、地址模式、Phase 1、Phase 2 等

高级设置包含以下关键项：

区域	参数
VPN Settings	IKE version、Main/Aggressive mode（IKEv1）、Encapsulation and IKE TCP port（IKEv2）、Addressing mode、Phase 1 options、Phase 2 options
Phase 1	IKE Proposal、DH Group、Key Life、Local ID、Dead Peer Detection、NAT Traversal、Local LAN
Phase 2	IPSec Proposal、Key Life、Replay Detection、Perfect Forward Secrecy（PFS）、DH Group

这些设置必须与 FortiGate 上的 VPN 设置匹配。例如，当 FortiGate 上配置多个拨号隧道并启用 Peer ID 时，客户端必须配置匹配的 Local ID。FortiClient 中 Local ID 位于 Phase 1 options。

如果可行，VPN 设置应通过 FortiClient EMS 集中配置和推送。从 FortiClient EMS 创建新的 Remote Access Profile，让 IPSec 隧道参数与 FortiGate 隧道设置一致。

先迁移 VPN 还是先升级?

先迁移 VPN 还是先升级 FortiOS，应由管理员根据公司策略、最佳实践和业务影响决定。需要评估两种方式对远程用户造成的潜在停机影响。

另一个考虑因素是当前固件是否影响安全性。如果存在关键安全补丁需求，管理员可能会决定先升级，再迁移 VPN。

同时，迁移需要时间完成设计评估、计划制定、配置实施、受控测试和用户部署。应预留足够时间进行规划，并在确定方案后再安排升级和维护窗口。

FortiOS SSL VPN 到拨号 VPN 的迁移

理解设计考虑事项后，可以根据实际偏好迁移配置。建议分两部分进行：

- 首先分析当前 SSL VPN 使用的用户认证方式，并理解哪些条件会影响 IPSec VPN 实现方式选择。
- 然后使用 VPN Wizard 配置 IPSec 隧道。某些场景可能需要进一步自定义配置。

本部分包含：

- 网络拓扑。
- 第一部分：识别用户认证方式。
- 第二部分：使用 VPN Wizard 配置 IPSec 隧道。

网络拓扑

第一部分：识别用户认证方式

- 本地用户认证
- 基于 LDAP 的用户认证
- 基于 RADIUS 的用户认证
- 基于 SAML 的用户认证

第二部分：使用 VPN Wizard 配置 IPSec 隧道

自定义 IPSec 隧道设置

网络拓扑

本文场景假设 SSL VPN 已经预先配置在 FortiGate WAN 接口上，并使用 TCP/443。远程用户连接到 FortiGate WAN 接口后，访问 FortiGate LAN 接口后的内部资源。

该 SSL VPN 配置将迁移到 IPSec，并保持相同的基础拓扑。

第二部分：使用 VPN Wizard 配置 IPSec 隧道

完成现有 SSL VPN 用户认证方式识别并确认 IPSec 支持要求后，可以开始迁移到 IPSec VPN。

IPSec 隧道可以通过 GUI VPN Wizard、GUI 自定义 IPSec 配置或 CLI 配置。这里使用 VPN Wizard 创建 IPSec 隧道。通过 VPN Wizard 指定的隧道参数后续仍可自定义，例如修改 IKE 版本、IKE 模式、安全关联（SA）和其他细粒度设置。

配置 VPN Wizard

1. 在 FortiGate 进入“VPN → VPN Wizard”。
2. 配置以下选项，并点击 Begin：

配置项	说明
Tunnel name	输入 VPN 隧道名称，最长 15 个字符
Select a template	选择 Remote Access

VPN 隧道

配置以下选项并点击 Next：

配置项	说明
VPN client type	选择 FortiClient；如果终端使用其他 VPN 客户端，可选择 iOS、Android/Windows 或 Cisco
Authentication method	选择 Pre-shared key 或 Signature
IKE	按需求选择 IKEv2 或 IKEv1；如需 TCP 传输，必须选择 IKEv2
Transport	UDP、Auto 或 TCP encapsulation；TCP encapsulation 仅 IKEv2 可用
Use Fortinet encapsulation	FortiClient 不支持 Fortinet 封装，应保持 Disabled
NAT traversal	中间存在 NAT 设备时启用；IKEv1 下可选择 Forced
Keepalive frequency	NAT traversal 为 Enabled 或 Forced 时可用，默认 10，用于隧道空闲时维持中间 NAT 会话
EAP peer identification	IKEv2 下使用 EAP 认证时选择 EAP identity request
User authentication method	选择用户组，或从防火墙策略继承
DNS Server	使用系统 DNS 或手工指定 DNS
Enable IPv4 Split Tunnel	启用后只有 Local address 中的流量进入隧道；关闭后所有远程用户流量进入隧道

Authentication method：

- Pre-shared key：创建唯一预共享密钥，所有 FortiClient 终端使用该密钥连接 VPN。
- Signature：使用用户证书认证远程用户。需要设置 Certificate Name 作为 VPN Gateway 身份证书，并设置 Peer Certificate CA 为签发 FortiClient 终端证书的 CA。服务器证书和 Peer CA 都必须提前导入 FortiGate。

User authentication method：

- 单用户组：如果当前 SSL VPN Authentication/Portal Mapping 使用单一用户组，选择 Phase 1 interface，并从下拉框中选择该用户组。
- 多用户组：如果当前 SSL VPN Authentication/Portal Mapping 使用多个用户组，选择 Inherit from policy，并选择这些用户组。VPN Wizard 会自动把这些用户组加入对应 IPSec 防火墙策略的 User/Group 字段。

如果启用 Split tunneling，并且指定的 DNS Server 位于 FortiGate 后方，需要确保该 DNS Server 可以通过 Local interface 访问，并且地址属于 Local Address 范围。

远程端点

配置以下选项并点击 Next：

配置项	说明
Addresses to assign to connected endpoints	分配给拨号客户端的地址范围
Subnet for connected endpoints	建议配置为 `255.255.255.255`，因为地址分配给单个客户端
Security posture gateway matching	可选，选择所需 Security posture tag
Save Password	允许 VPN 客户端保存 XAuth 用户名和密码，默认启用，对应 `set save-password enable`
Auto Connect	允许客户端在无流量时拉起隧道，默认关闭，对应 `set client-auto-negotiate disable`
Always up（keepalive）	允许客户端在无流量时保持隧道，默认关闭，对应 `set client-keep-alive disable`

VPN Wizard 只会使用 Mode Config 和地址范围创建隧道。如果需要其他地址分配方式，可以在 Wizard 创建后自定义。

可以使用不同于原 SSL VPN 的地址范围，避免 IP 地址重叠。

本地 FortiGate

配置以下选项并点击 Next：

配置项	说明
Incoming interface that binds to tunnel	与 SSL VPN Settings 中的 Listen on interface 对应
Create and add interface to zone	如需将接入接口划入 Zone 则启用，否则关闭
Local interface	VPN 用户访问的内部接口，对应原 `ssl.root` 到目的接口的防火墙策略
Local address	VPN 用户允许访问的内部网段，对应原 `ssl.root` 到目的地址的防火墙策略

如果在 VPN tunnel 部分启用了 IPv4 Split Tunnel，Local Address 中的网段会用于分离隧道。

配置确认

提交前检查 VPN Wizard 将创建的对象，然后点击 Submit：

类型	对象
Addresses	Split address group，作为允许访问的目的地址组，并用于 Split tunneling
Addresses	Address，作为 VPN 客户端地址范围
Interfaces	VPN IPsec Phase 1 interface
Interfaces	VPN IPsec Phase 2 interface
Interfaces	Zone，如启用了 Zone 创建
Policies	Remote to local policies
Peer	如果认证方式为 Signature，会创建或引用 PKI Peer

VPN Wizard 会生成所需配置、对象和策略。可进入“VPN → IPsec tunnels”查看新建隧道。

使用 TCP 作为传输方式

拨号 IPSec VPN 传统上依赖 UDP，现在可以使用默认 TCP/443。该能力可以让 FortiClient 的 VPN 流量穿越只允许 TCP 流量的限制性防火墙。

该功能要求：

- IPSec 隧道使用 IKEv2。
- FortiClient 也配置为使用 TCP 传输。
- FortiClient 版本为 7.4.1 或更高。

通过 GUI 配置 TCP 传输：

1. 进入“VPN → IPsec Tunnels”，编辑对应 IPSec 隧道。
2. 确认 IKE 为 Version 2。
3. 将 Transport 设置为 TCP encapsulation。
4. 点击 OK。

通过 CLI 配置 TCP 传输：

```bash
config vpn ipsec phase1-interface
    edit <tunnel-name>
        set transport tcp
    next
end
```

查看 FortiGate 默认 IKEv2 TCP 端口：

```bash
show full-configuration system settings | grep ike-tcp
```

示例输出：

```bash
set ike-tcp-port 443
```

可选：将默认 TCP 端口改为 5500：

```bash
config system settings
    set ike-tcp-port 5500
end
```

`ike-tcp-port <port>` 用于设置 IKE/IPSec 流量的 TCP 端口，范围 1-65535，默认 443。

重要

当 IKE/IPSec 使用 TCP/443，且 GUI HTTPS 管理端口也使用 443 时，绑定 IPSec 隧道的接口可能影响 GUI 管理访问。此时应修改 IKE/IPSec 端口或管理访问端口。

```bash
config system global
    set admin-sport <port>
end
```

如果与 ZTNA 或 SSL VPN 端口冲突，ZTNA 和 SSL VPN 优先。应检查 FortiGate 上其他入站服务端口，避免端口冲突。

后续步骤

根据需求，可能还需要编辑 VPN Wizard 创建的 IPSec 隧道设置。常见自定义包括 Peer ID/Network ID，以及 Phase 1 和 Phase 2 Proposal。

自定义 IPSec 隧道设置

VPN Wizard 创建隧道后，可能需要根据实际需求继续调整配置。常见自定义包括：

- 使用 Peer ID。
- 修改 Phase 1 和 Phase 2 Proposal。

使用 Peer ID

如果同一物理 WAN 接口上配置多个拨号 IPSec 隧道，FortiGate 需要通过 Peer ID 或 Network ID 区分传入的 IPSec 连接请求，并将连接关联到正确的 IPSec 隧道。

IKEv1 Aggressive Mode 可以使用 Peer ID 区分多个拨号 IPSec 隧道。因此，IKEv1 建议使用 Aggressive Mode，以便在 Phase 1 中使用 Peer ID 字段。

IKEv1 Aggressive Mode 协商过程中，FortiClient 会发送本地配置的 Local ID，FortiGate 将其与定义好的 Peer ID 匹配，从而识别对应隧道。因此，FortiClient 上的 Local ID 必须与 FortiGate 上对应隧道的 Peer ID 一致。

IKEv2 中，Peer ID 不包含在初始 IKE 消息中。因此 FortiGate 作为 IPSec 拨号 Server 时，无法准确地在多个拨号 Phase 1 配置中匹配正确隧道。IKEv2 建议改用 Phase 1 中的 Network ID。

Network ID 不能在未纳管的独立 FortiClient 中配置。通过 FortiClient EMS 管理时，FortiClient EMS 7.2.6 及以上或 7.4.1 及以上支持配置 Network ID。

IKEv2 中，FortiGate 使用 Network ID 作为唯一标识，区分同一 WAN 接口上的多个拨号隧道。协商过程中，FortiClient 发送 Network ID，FortiGate 将其与已定义的 Network ID 匹配，以识别正确隧道。FortiClient 上配置的 Network ID 必须与 FortiGate 上对应隧道的 Network ID 一致。

> **重要**
>
> 不同 IPSec 隧道必须配置唯一 Peer ID 或 Network ID。

IKEv1 Peer ID 配置思路：

1. 进入“VPN → IPsec Tunnels”，编辑对应 IPSec 隧道。
2. 确认 IKE 为 Version 1，Mode 为 Aggressive。
3. 在 Authentication 下将 Accept Peer ID 改为 Specific peer ID。
4. 配置唯一 Peer ID。
5. 保存。

IKEv2 Network ID 通过 CLI 配置：

```bash
config vpn ipsec phase1-interface
    edit <vpn-tunnel-name>
        set network-overlay enable
        set network-id <ID>
    next
end
```

修改 Phase 1 和 Phase 2 Proposal

如需修改 IPSec Phase 1 和 Phase 2 的 SA 参数：

1. 进入“VPN → VPN Tunnels”，编辑 IPSec 隧道。
2. 在 Phase 1 proposal 中选择需要的自定义配置。
3. 在 Phase 2 Selectors 中选择 Phase 2 隧道并点击 Edit。
4. 选择所需自定义配置，并点击 OK 保存 Phase 2 Selectors。
5. 点击 OK 保存 IPSec 隧道配置。

第一部分：识别用户认证方式

Part 1 用于识别当前 SSL VPN 配置使用的用户认证方式。针对每种方式，需要确认迁移到 IPSec 时是否存在限制，以及是否有 IKE 版本要求。

FortiGate 的用户认证方式通常依赖用户或用户组配置。用户组中可能引用 RADIUS、LDAP、SAML 等认证服务器。这些现有对象通常可以直接复用到 IPSec VPN 配置中。

识别当前 SSL VPN 使用的认证方式

1. 找出 SSL VPN 防火墙策略中使用的用户组。
   - 进入“策略&对象 → 防火墙策略”。
   - 编辑 Incoming interface 为 `ssl.root` 的策略。
   - 记录 Source 字段中的用户组。
   - 对所有 SSL VPN 防火墙策略重复该步骤，得到 SSL VPN 用户认证使用的用户组列表。
2. 识别 SSL VPN 配置的认证方式。
   - 进入“用户与认证 → 用户组”，编辑上一步识别出的用户组。
   - 根据以下表现判断认证方式。

配置表现	认证方式
Member 下配置本地用户，Remote Groups > Remote Server 无配置	Local user authentication
Remote Groups > Remote Server 使用 LDAP Server	LDAP-based user authentication
Remote Groups > Remote Server 使用 RADIUS Server	RADIUS-based user authentication
Remote Groups > Remote Server 使用 SAML SSO Server	SAML-based user authentication
Member 下配置 PKI 用户，同时 Remote Group 使用 LDAP Server 或 RADIUS Server	Certificate-based user authentication，并使用 LDAP 或 RADIUS 作为双因素认证

> 注
>
> 本文不演示证书型用户认证迁移。证书认证迁移应参考 IKEv1/IKEv2 相关配置说明。

识别认证方式后，进入对应章节查看迁移要求。确认认证方式后，再进入 Part 2，使用 VPN Wizard 配置 IPSec 隧道，并复用 Part 1 中确认的用户组。

- 本地用户认证
- 基于 LDAP 的用户认证
- 基于 RADIUS 的用户认证
- 基于 SAML 的用户认证

本地用户认证

本地用户认证中，用户名和密码直接在 FortiGate 本地为每个用户配置。可以通过本地用户组包含多个本地用户。

示例中，本地用户 `johnlocus` 加入名为 `Local user group` 的本地用户组。

查看 CLI 配置：

```bash
config user group
    edit "Local user group"
        set member "johnlocus"
    next
end
```

应用用户组

如果只有一个用户组，可以将 `Local user group` 直接用于 IPSec 隧道配置中。如果有多个用户组，应在 IPSec 隧道中选择 Inherit from policy，并在防火墙策略中使用这些用户组。

基于 LDAP 的用户认证

IPSec IKEv1 使用 XAUTH 进行用户认证，IPSec IKEv2 使用 EAP 进行用户认证。只有 EAP-TTLS 可以与 LDAP 互通。因此，如果 LDAP 用户认证使用 IKEv2，必须使用 EAP-TTLS。

> **重要**
>
> EAP-TTLS 支持要求 FortiClient EMS 和 FortiClient 版本为 7.4.3 或更高。

LDAP 认证中，LDAP Server 是集中认证服务器，用户名和密码直接在 LDAP Server 上管理。要在 IPSec 中使用该认证方式，FortiGate 必须配置 LDAP Server，并创建使用 LDAP Server 的用户组。

如需基于 LDAP 组成员关系区分用户组，可以配置多个用户组并使用 Group Name 选项进行匹配。

示例中，LDAP Server 名为 `LDAP Connector`，用户组名为 `LDAP user group`。Group Name 只匹配 LDAP Server 上属于 `Domain Users` 的用户，只有这些用户允许连接 IPSec 隧道。

查看 CLI 配置：

```bash
config user group
    edit "LDAP user group"
        set member "LDAP Connector"
        config match
            edit 1
                set server-name "LDAP Connector"
                set group-name "CN=Domain Users,CN=Users,DC=financial,DC=local"
            next
        end
    next
end
```

应用用户组

如果只有一个用户组，可以将 `LDAP user group` 直接用于 IPSec 隧道配置中。如果有多个用户组，应在 IPSec 隧道中选择 Inherit from policy，并在防火墙策略中使用这些用户组。

使用 LDAP + IKEv2 时，需要将 IKE version 改为 Version 2。

基于 RADIUS 的用户认证

RADIUS 认证中，RADIUS Server 是集中认证服务器，用户名和密码直接在 RADIUS Server 上管理。要在 IPSec 中使用该认证方式，FortiGate 必须配置 RADIUS Server，并创建引用该 RADIUS Server 的用户组。

如需基于 RADIUS Server 上的用户组成员关系区分用户组，可以使用 Group Name。FortiGate 期望 RADIUS Server 在 RADIUS 响应报文中正确返回 `Fortinet-Group-Name` VSA。

示例中，FortiGate 配置了名为 `Radius Connector` 的 RADIUS Server，用户组 `Radius user group` 引用该 RADIUS Server。Group Name 配置为 `Domain Users`，表示只有 RADIUS Server 在响应中返回该组名时，用户才允许连接 IPSec 隧道。

查看 CLI 配置：

```bash
config user group
    edit "Radius user group"
        set member "Radius Connector"
        config match
            edit 1
                set server-name "Radius Connector"
                set group-name "Domain Users"
            next
        end
    next
end
```

应用用户组

如果只有一个用户组，可以将 `Radius user group` 直接用于 IPSec 隧道配置中。如果有多个用户组，应在 IPSec 隧道中选择 Inherit from policy，并在防火墙策略中使用这些用户组。

基于 SAML 的用户认证

IPSec 支持基于 SAML 的用户认证，但要求 FortiClient 版本为 7.2.4 或更高。SAML 认证只支持 IPSec IKEv2，不支持 IKEv1。

使用 SAML 时，建议先升级 FortiClient 到 7.2.4 或更高版本。

VPN Wizard 默认可能创建 IKEv1 配置，后续需要将配置调整为 IKEv2，并启用 EAP 用户认证。

SAML 与 IPSec 配合时，还需要额外配置：

- `auth-ike-saml-port`。
- SAML Server 证书。
- IPSec VPN 网关接口和 SAML Server 之间的接口绑定。

配置 IKE SAML 认证端口：

```bash
config system global
    set auth-ike-saml-port 9443
end
```

查看 SAML Server 使用的证书：

```bash
config user setting
    set auth-cert "SAML_Server_Certificate"
end
```

示例中，SAML 用户组 `SAML Usergroup` 使用名为 `SAML-FAC` 的 SAML SSO Server，并匹配组名 `Corporate`：

```bash
config user group
    edit "SAML Usergroup"
        set member "SAML-FAC"
        config match
            edit 1
                set server-name "SAML-FAC"
                set group-name "Corporate"
            next
        end
    next
end
```

将 SAML Server 与 IPSec 网关所在接口绑定，示例接口为 `WAN`：

```bash
config system interface
    edit "WAN"
        set ike-saml-server "SAML-FAC"
    next
end
```

应用用户组

如果只有一个用户组，可以将 `SAML Usergroup` 直接用于 IPSec 隧道配置中。如果有多个用户组，应在 IPSec 隧道中选择 Inherit from policy，并在防火墙策略中使用这些用户组。

FortiClient VPN

- FortiClient 下载
- 系统兼容性列表
- Windows FortiClient 安装及使用
- mac FortiClient 安装及使用
- iPhone FortiClient 安装及使用
- Android FortiClient 安装及使用
- FortiClient 连接问题
- FortiClient 排错与卸载

FortiClient 下载

本地下载

Windows (v7.2.14.1296)

- 完整版（需要购买 EMS License）

  FortiClientSetup_7.2.14.1296_x64.zip ↗
- VPN Only 版（7.2.14.1296，免费使用，仅包含 VPN 功能，不提供技术支持）

  FortiClientVPNSetup_7.2.14.1296_x64.exe ↗

macOS (v7.2.14.1132)

- 完整版（需要购买 EMS License）

  FortiClient_7.2.14.1132_macosx.dmg ↗
- VPN Only 版（7.2.14.1132，免费使用，仅包含 VPN 功能，不提供技术支持）

  FortiClientVPNSetup_7.2.14.1132_macosx.dmg ↗

Android (v7.2.4-build0138)

- 完整版（需要购买 EMS License）

  FortiClient_v7.2.4-build-release.apk ↗
- VPN Only 版（7.2.4-build0138，免费使用，仅包含 VPN 功能，不提供技术支持）

  FortiClient_vpn-v7.2.4-build-release.apk ↗

Windows (v7.4.7.2003)

- 完整版（需要购买 EMS License）

  FortiClientSetup_7.4.7.2003.M_x64.zip ↗
- VPN Only 版（7.4.3.4726，免费使用，仅包含 VPN 功能，不提供技术支持）

  FortiClientVPNSetup_7.4.3.4726_x64.exe ↗

macOS (v7.4.7.1928)

- 完整版（需要购买 EMS License）

  FortiClient_7.4.7.1928.M_macosx.dmg ↗
- VPN Only 版（7.4.3.4323，免费使用，仅包含 VPN 功能，不提供技术支持）

  FortiClientVPNSetup_7.4.3.4323_macosx.dmg ↗

Android (v6.2.3-build0332)

- 完整版（需要购买 EMS License）

  FortiClient_android.apk ↗
- VPN Only 版（6.2.3-build0332，免费使用，仅包含 VPN 功能，不提供技术支持）

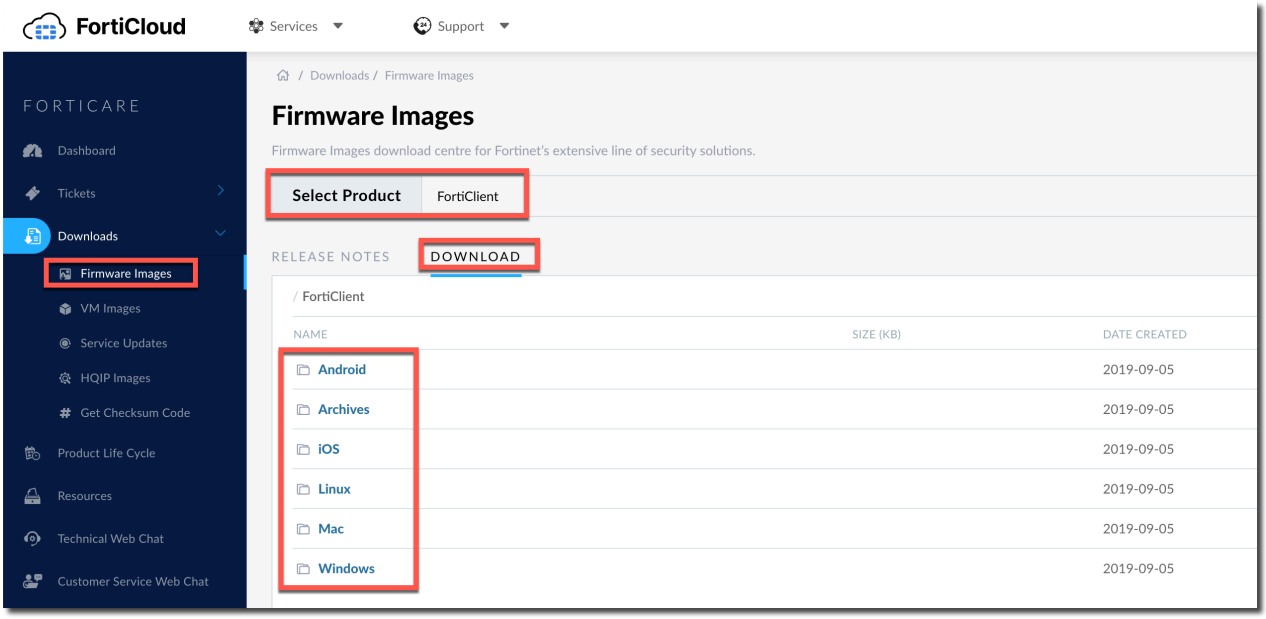
  FortiClientVPN_android.apk ↗

FortiToken Android (v6.2.0-build0013)

FortiTokenAndroid.apk ↗

官网下载

1. 直接打开并登录 https://support.fortinet.com/support/#/downloads/firmware ↗ ，选择产品为 FortiClient，点击 Download 模块，选择对应系统与版本下载。



2. iOS 版本请参考 VPN 技术 → FortiClient VPN → iPhone FortiClient 安装及使用 ↗章节。

注意事项

1. 完整版 FortiClient 需要 EMS License，可根据 EMS 序列号提供技术支持。免费的 FortiClient VPN 仅支持 SSL VPN 和 IPSec VPN，且不提供技术支持服务。



2. 免费版 FortiClient（FortiClient VPN Only）不支持以下 VPN 功能：

   - VPN 自动连接、保持在线。
   - 在系统登录前登录 VPN。

- EMS 集中管理。
- 调试工具。

3. FortiClient 与操作系统兼容性请参考：

- FortiClient iOS 客户端与系统兼容性：https://docs.fortinet.com/document/forticlient/7.2.99/ios-administration-guide/394867/supported-platforms ↗
- FortiClient Android 客户端与系统兼容性：https://docs.fortinet.com/document/forticlient/7.2.99/android-administration-guide/191370/product-integration-and-support ↗
- FortiClient Windows 客户端与系统兼容性：https://docs.fortinet.com/document/forticlient/7.2.99/windows-release-notes/549781/product-integration-and-support ↗
- FortiClient macOS 客户端与系统兼容性：https://docs.fortinet.com/document/forticlient/7.2.99/macos-release-notes/471180/product-integration-and-support ↗
- FortiClient EMS 与 Fortinet 其他产品兼容性：https://fortinetweb.s3.amazonaws.com/docs.fortinet.com/v2/attachments/afec3249-ed3f-11ea-96b9-00505692583a/FortiClient_ems-compatibility-matrix.pdf ↗

系统兼容性列表

提示

- ❓ 表示该版本不支持 ZTNA，未填项表示 FCT 版本不支持该 OS 版本。
- 🧪 表示该版本是 Beta 版本，支持有限的功能。
- Metro 表示 Windows Metro 应用， 需要通过 Window Store 下载：https://apps.microsoft.com/detail/9wzdncrdh6mc?hl=zh-CN&gl = CN
- 不推荐使用 7.0.0 - 7.0.7，所以此表暂未收录，详细信息请参考 Release Note：https://docs.fortinet.com/product/forticlient/7.0

OS/FCT 版本		6.4.7	6.4.8	6.4.9	6.4.10	7.0.7	7.0.8	7.0.9	7.0.10	7.0.11	7.0.12	7.0.13	7.0.14	7.2.0	7.2.1	7.2.2	7.2.3	7.2.4	7.2.5
Windows	Winows 7	✅	✅	✅	✅	❓	❓	❓						❓	❓				
	Windows 8																		
	Windows 8.1	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅						
	Windows 10	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
	Windows 11	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Windows ARM	Windows RT 8																		
	Windows RT 8.1																		
	Windows 10 ARM																		
	Windows 11 ARM																		
Windows Server	Windows Server 2003																		
	Windows Server 2008																		
	Windows Server 2008 R2	✅	✅	✅	✅	✅	✅												
	Windows Server 2012	✅	✅	✅	✅	✅	✅	✅											
	Windows Server 2012 R2	✅	✅	✅	✅	✅	✅	✅											
	Windows Server 2016	✅	✅	✅	✅	✅	✅	✅											
	Windows Server 2019	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
	Windows Server 2022	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
macOS	macOS 10.14 (Mojave)	✅	✅		✅														
	macOS 10.15 (Catalina)	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅						
	macOS 11 (Big Sur)	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅			
	macOS 12 (Monterey)	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
	macOS 13 (Ventura)			✅		✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
	macOS 14 (Sonoma)								✅	✅	✅	✅	✅			✅	✅	✅	✅
	macOS 15 (Sequoia)											✅	✅						✅
	macOS 26 (Tahoe)																		

Windows FortiClient 安装及使用

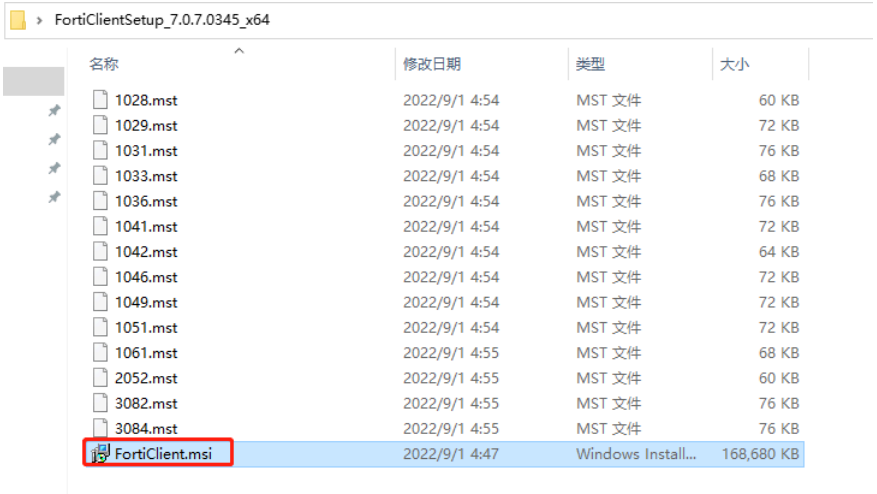
Windows FortiClient 完整版安装

1.下载 FortiClient

参考：VPN 技术 → FortiClient VPN → FortiClient 下载 ↗。

2.安装 FortiClient

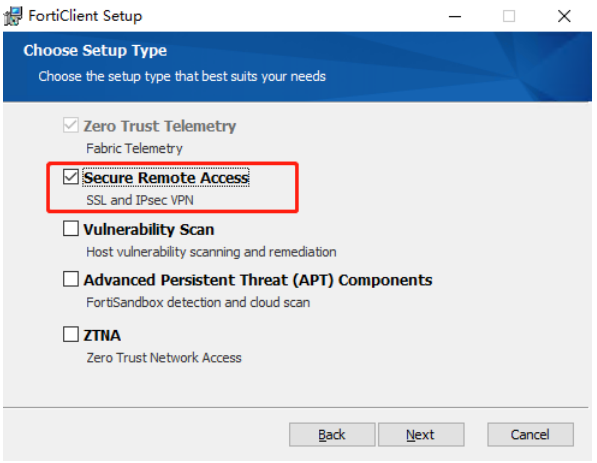
解压 FortiClient 下载包，然后点击“FortiClient.msi”安装（版本号仅举例使用，请根据实际下载的版本文件执行）。



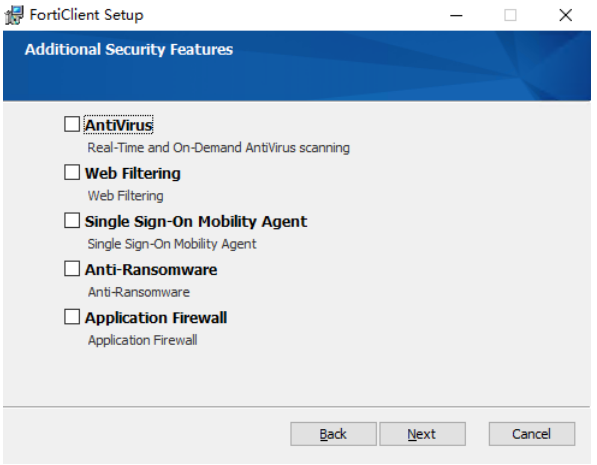
勾选“Yes,I have read an accept the ...”，点击 Next。



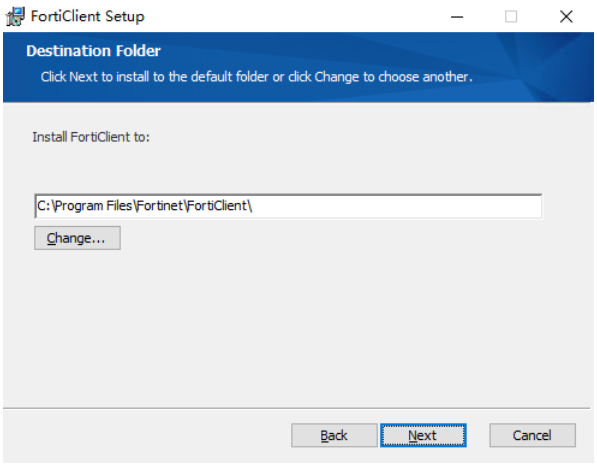
选择“Secure Remote Access”，即 SSLVPN 和 IPSEC VPN，点击 Next。



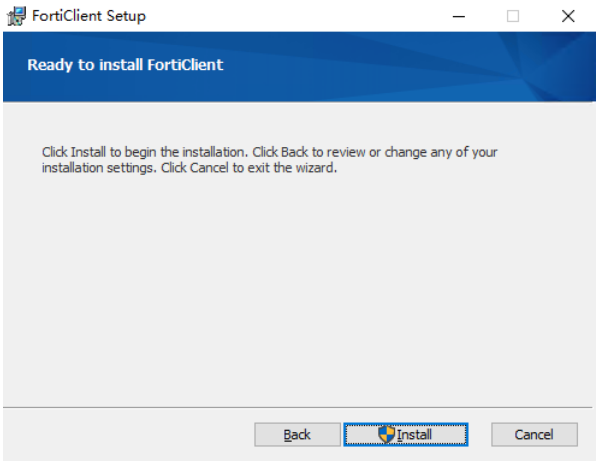
点击 Next。

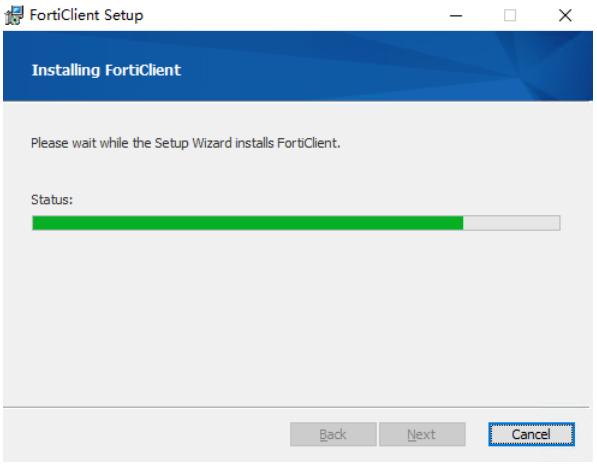


点击 Next。

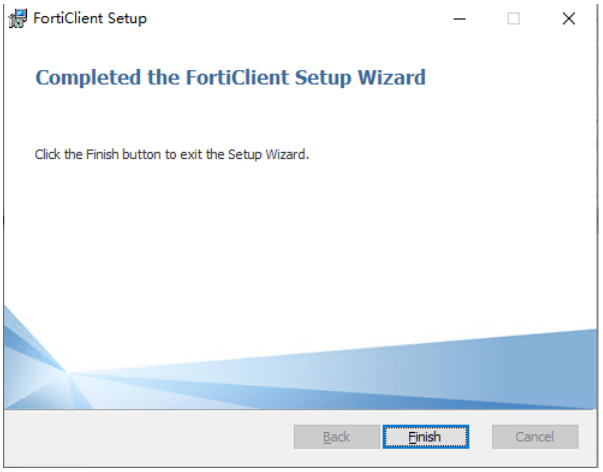


点击 Install。



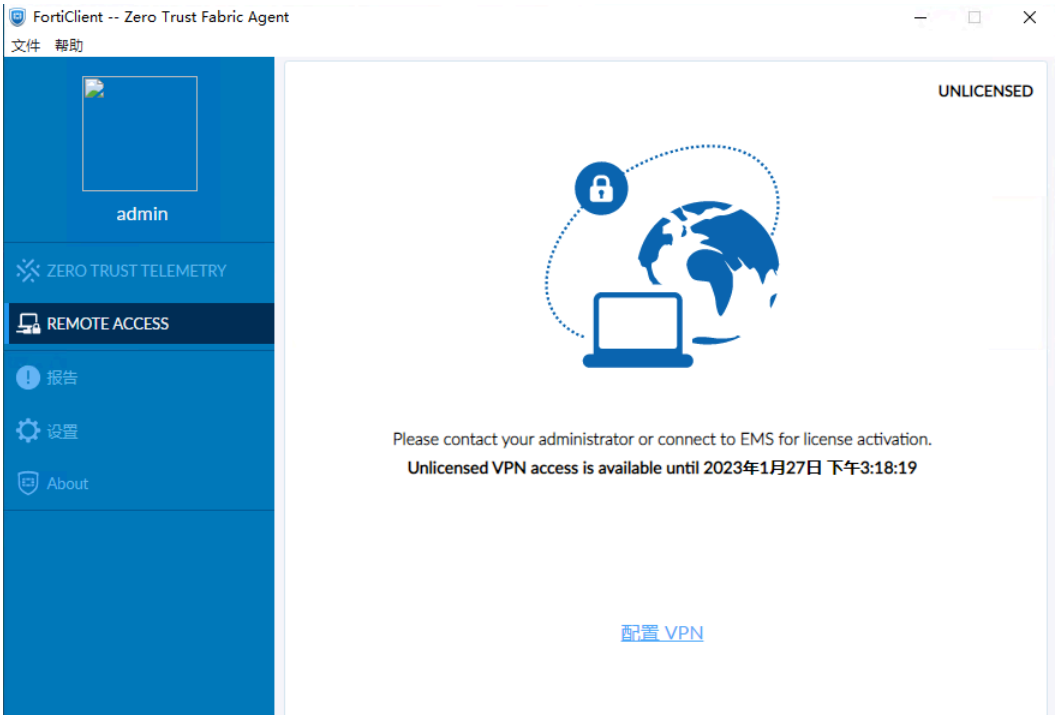


点击“Finish”，安装完成。

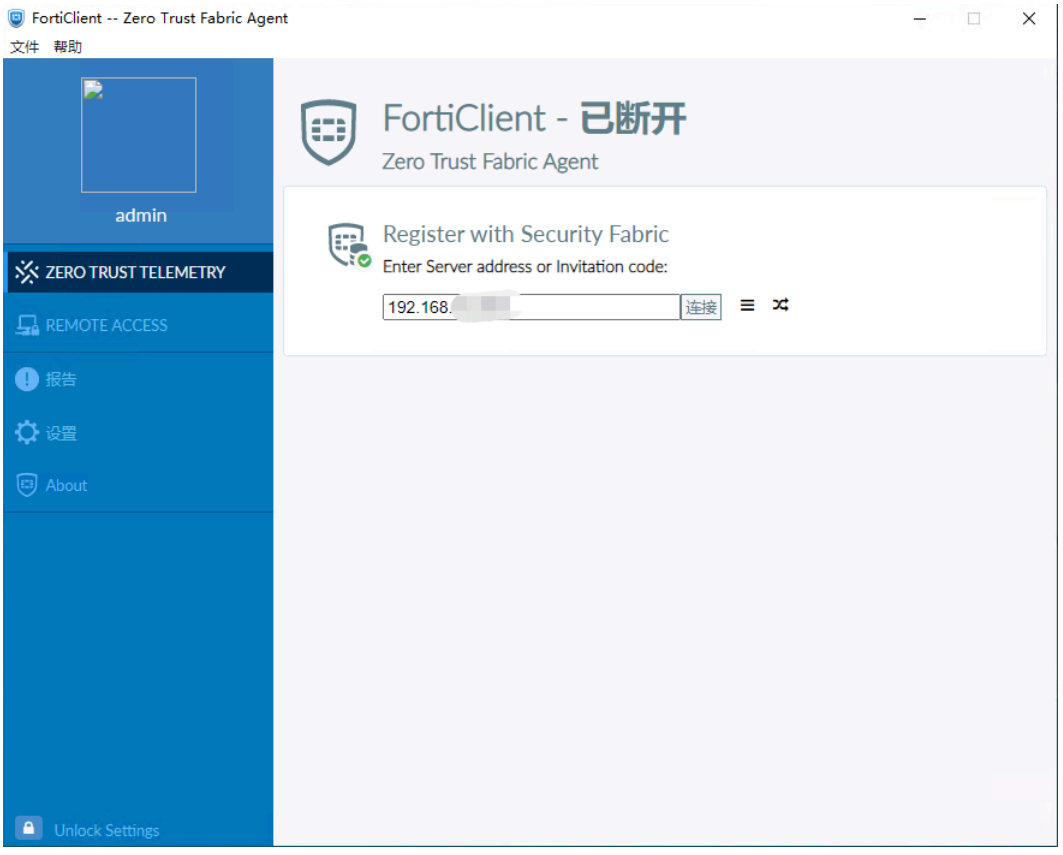


3.连接 EMS

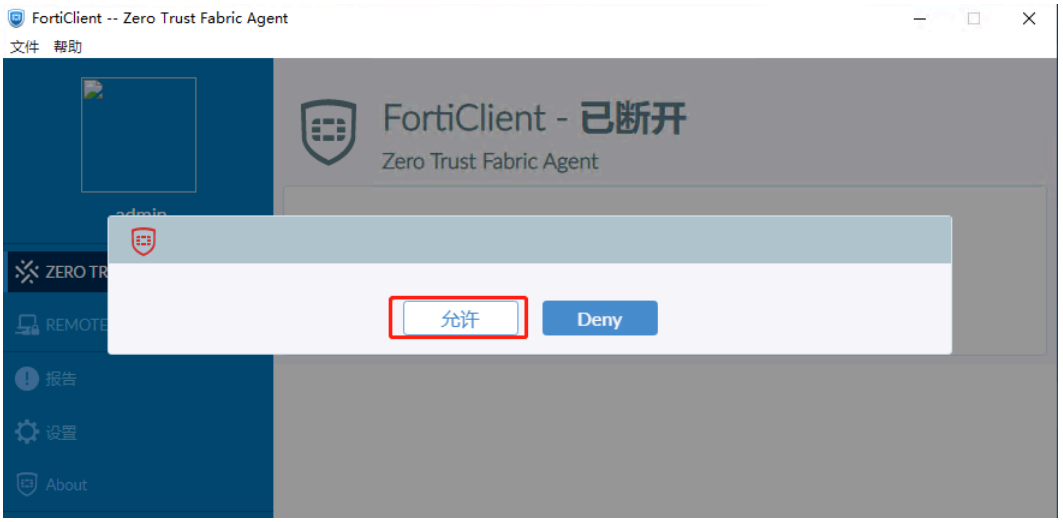
打开 FortiClient，在没有连接 EMS 之前，FortiClient 会显示 UNLICNESED，并提示试用时间。



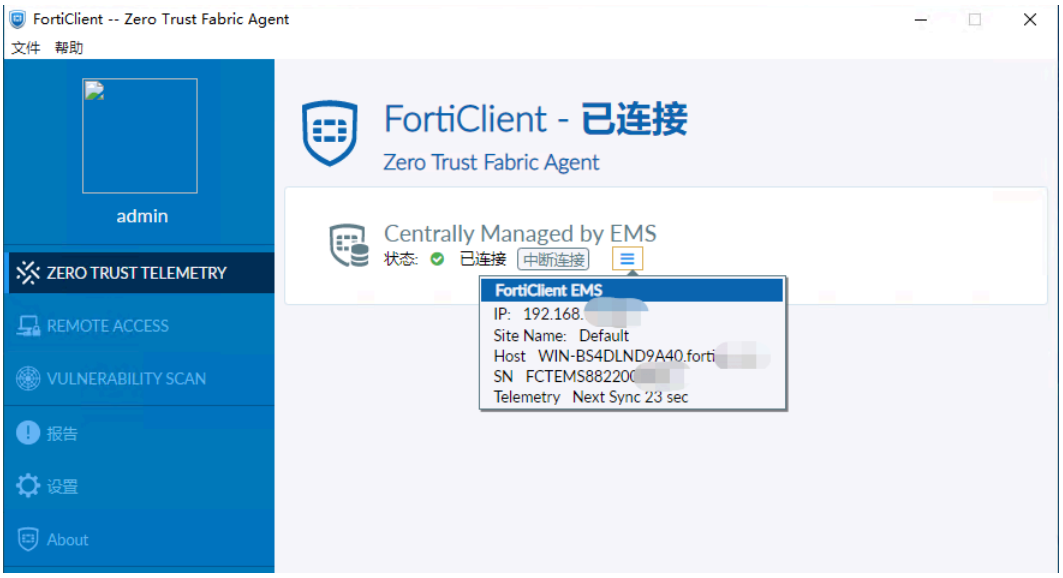
输入 EMS 服务器的地址或者域名。



选择”允许“



已连接上 EMS，并且不再提示 UNLICNESED。





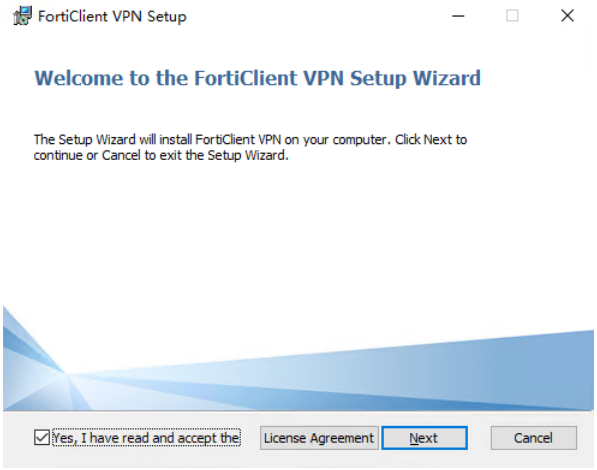
Windows 免费版 FortiClient VPN 安装

1.下载 FortiClient

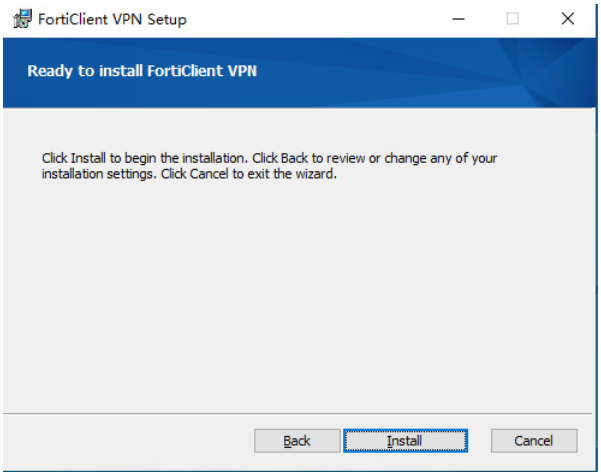
参考：VPN 技术 → FortiClient VPN → FortiClient 下载 ↗。

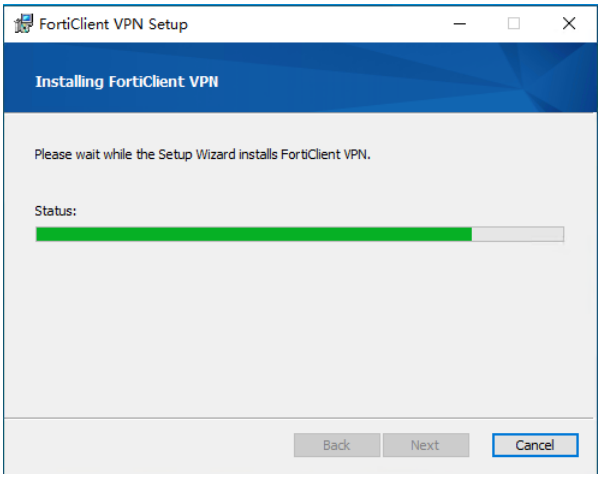
2.安装 FortiClient

双击 FortiClientVPNSetup_7.0.7.0345_x64.exe（版本号仅举例使用，请根据实际下载的版本文件执行）安装，勾选“Yes,I have read an accept the ...”，点击 Next。

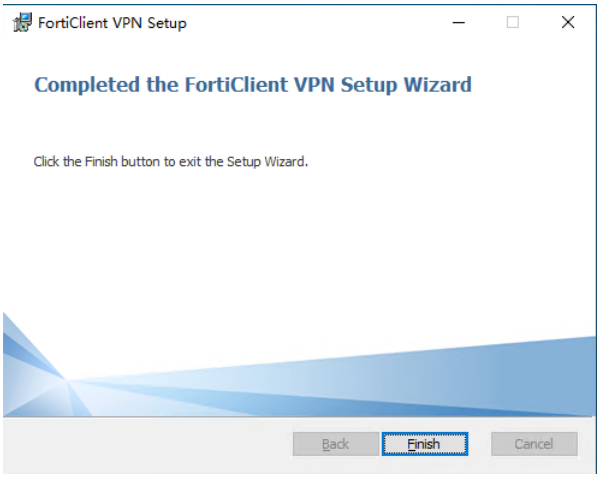


点击 Install。



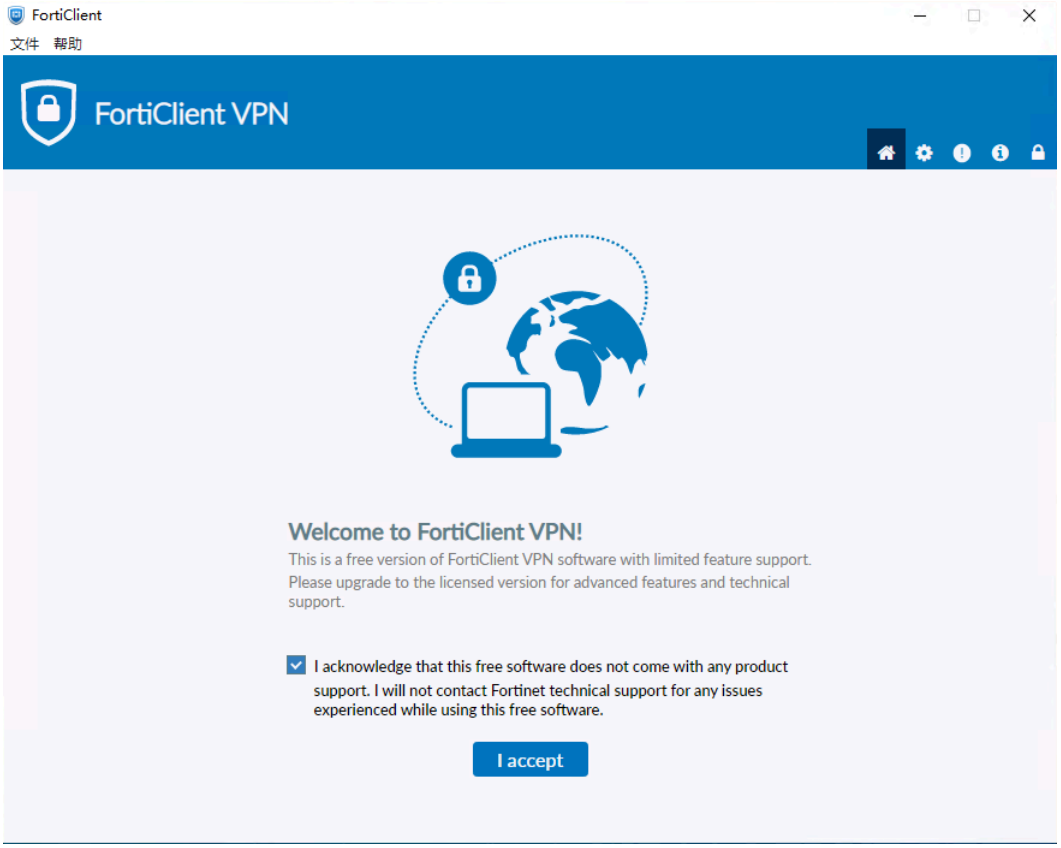


安装完成，点击“Finish”。

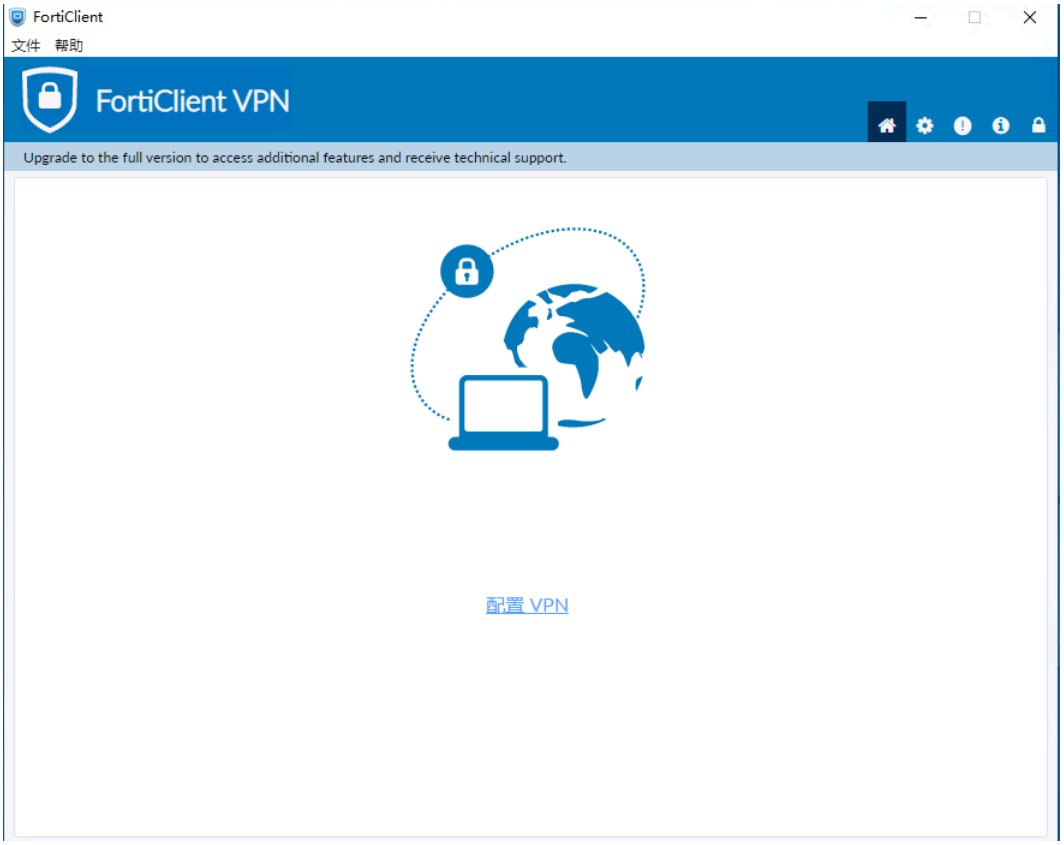


3.打开 FortiClient

勾选“I acknowledge that this free software...”，点击 I accept。免费的 FortiClient 是不提供技术支持服务的。



同意后显示配置界面。



Windows 设置 SSLVPN

1.点击“配置 VPN”

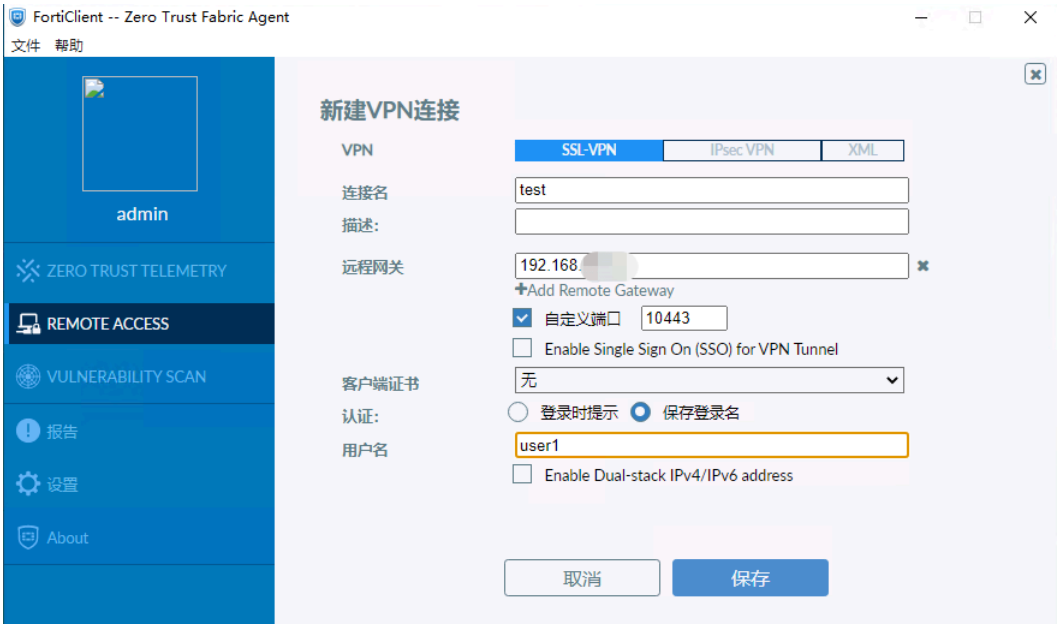


2.设置 SSLVPN 连接

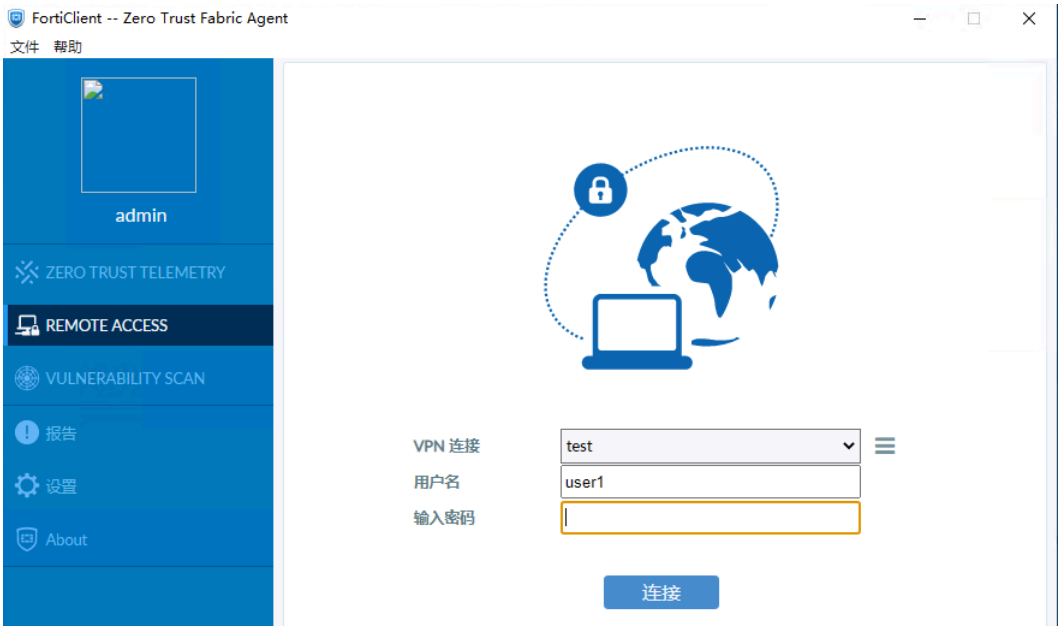
连接名：SSLVPN 名称

远程网关：FortiGate 开启 SSLVPN 服务的接口地址；

自定义端口：SSLVPN 服务的端口号；

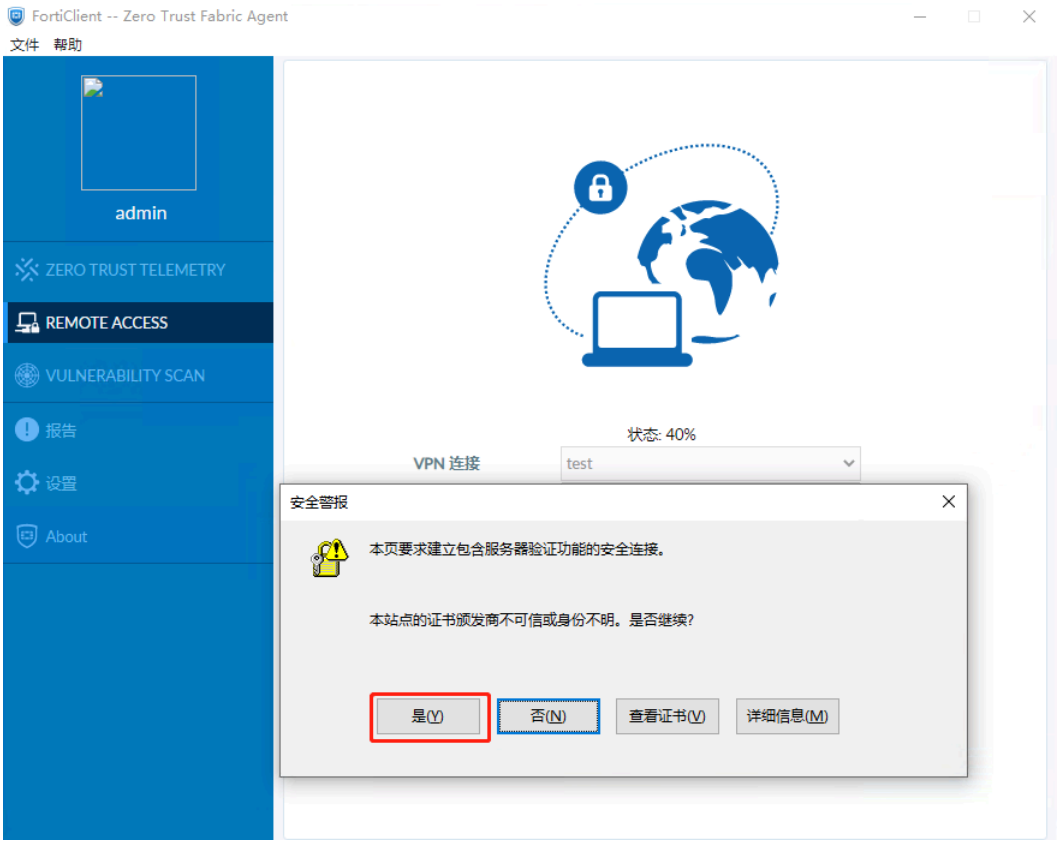


3.配置完成



4.SSLVPN 连接

输入账号和密码后，点击连接，再弹出窗口选择是，接受 FortiGate 的证书。



连接成功。



通过 CLI 方式安装 FortiClient

1. 您可以使用 CLI 方式安装 FortiClient，从而实现安装时的参数设置。下表总结了使用 CLI 时可用的安装选项。 无论您使用.exe 文件还是.msi 文件和.mst 文件，这些选项都可用：

安装参数	说明
/quiet	安装处于静默模式，不需要用户交互。
/passive	安装处于无人值守模式，仅显示进度条。
/norestart	安装完成后不重启系统。
/promptrestart	如有必要，提示您重新启动系统。
/forcerestart	安装后强制重启系统。
/uninstallfamilys	安装前卸载已安装的 FortiClient。使用此选项，FortiClient 安装程序会检测当前系统已安装的 FortiClient 并卸载 FortiClient。例如，FortiClient 7.0.8 安装前可以检测并卸载之前已安装的 FortiClient 7.0.7。

2. 以下示例以静默模式安装 FortiClient build 1131，安装后不重启系统，并在 c:\temp 目录中输出名为“example.log”的安装日志文件。

```
FortiClientSetup_7.0.8.1131_x64.exe /quiet /norestart /log c:\temp\example.log
```

3. 以下示例使用.msi 和.mst 文件安装 FortiClient，并创建一个名为“output”的日志文件。

```
msiexec.exe /i "FortiClient.msi" TRANSFORMS="FortiClient.mst" /log output.log
```

mac FortiClient 安装及使用

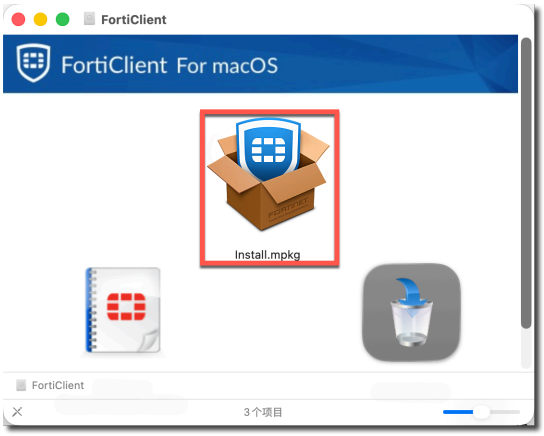
FortiClient 完整版本安装

1.下载 FortiClient

参考：VPN 技术 → FortiClient VPN → FortiClient 下载 ↗。

2.安装 FortiClient

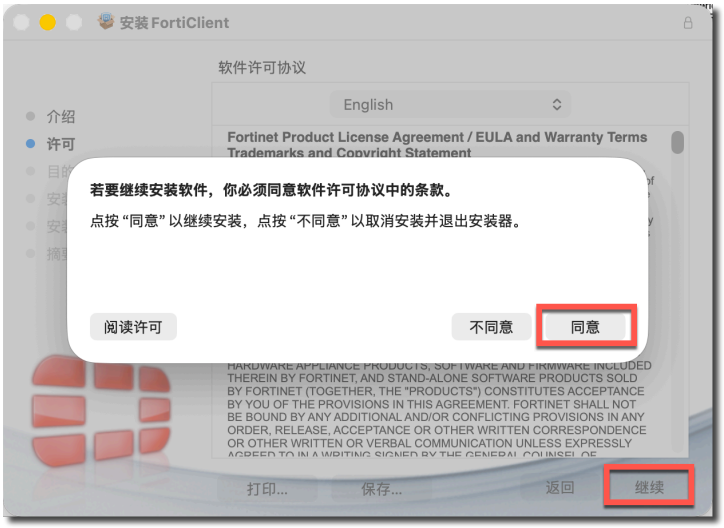
双击 FortiClientVPNSetup_7.4.3.1761_macosx.dmg（版本号仅举例使用，请根据实际下载的版本文件执行）安装，双击“Install.mpkg”。



点击“继续”。



点击继续并在弹出窗口中点击“同意”。



点击“继续”。



点击“安装”，在弹出窗口中输入系统用户名和密码授权安装。



安装完成后，关闭安装程序。



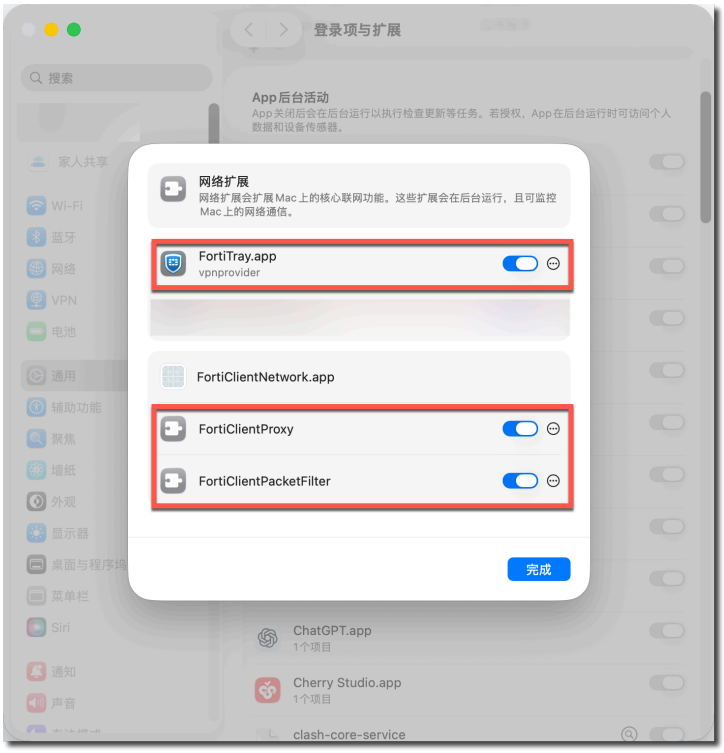
在弹出的“添加 VPN 配置”窗口中点击允许。



在弹出的网络扩展使用请求窗口中点击“打开系统设置”（点击其中一个即可）。



打开授权以下权限（需要系统账户验证）。

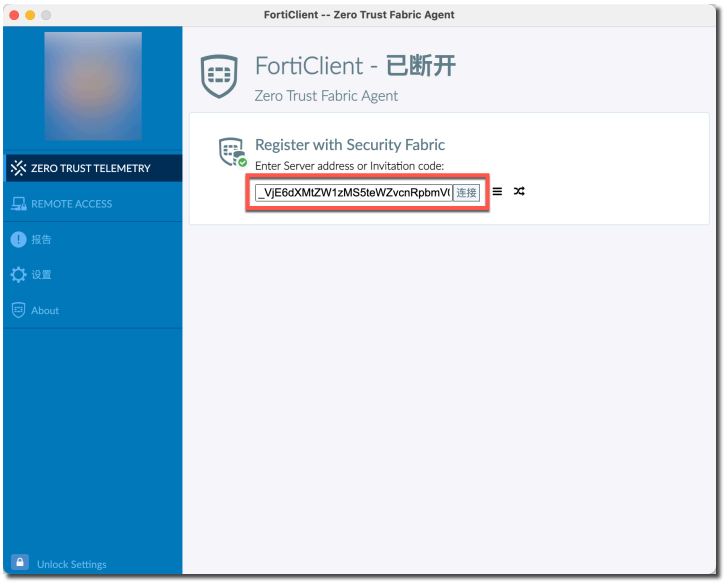


在弹出的“过滤网络内容”窗口中点击允许。

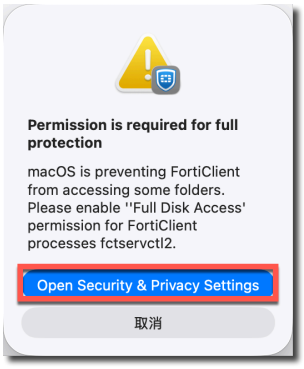


3. 连接 EMS

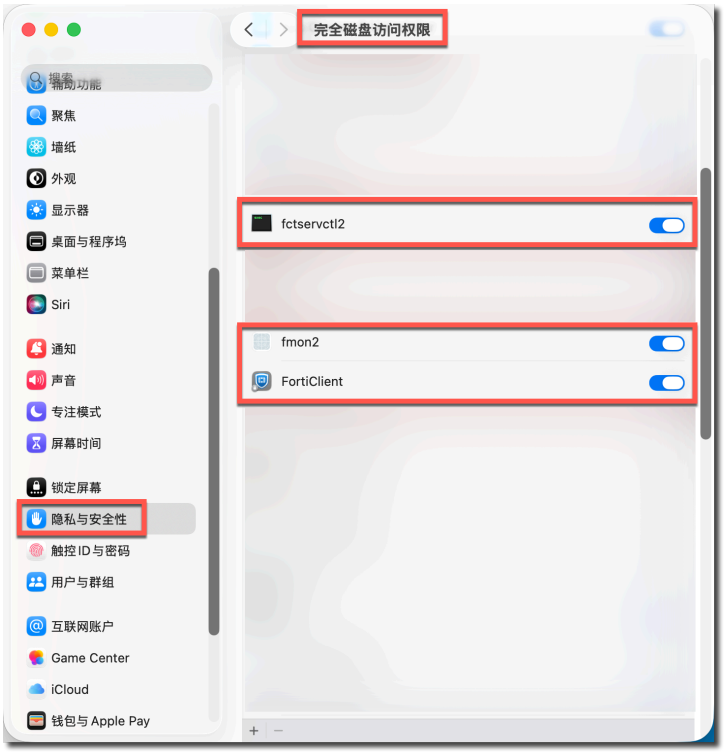
打开 FortiClient 输入 EMS IP 地址或邀请码，点击“连接”。



在弹出的磁盘权限窗口中点击“Open Security & Private Settings”。



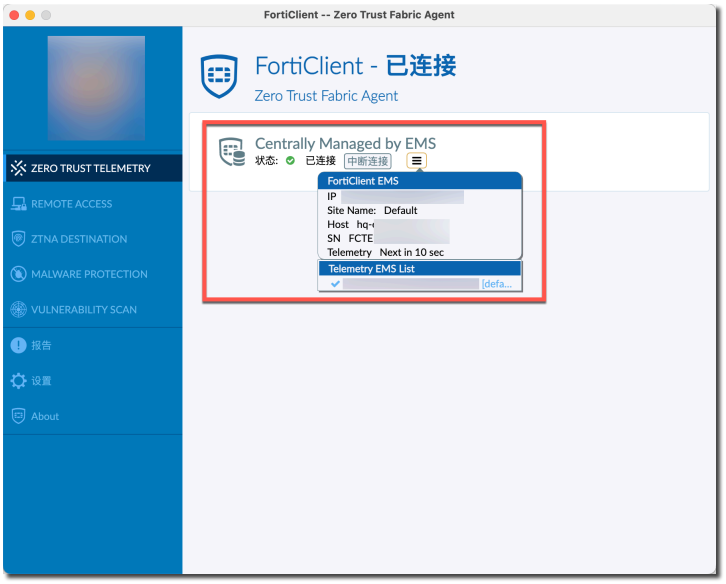
在弹出的“隐私与安全性 → 完全磁盘访问权限”窗口中授予 fctservctl2、fmon2、FortiClient 的磁盘访问权限（需要系统账户验证）。



在弹出的“过滤网络内容”窗口中点击“允许”。



EMS 连接成功。



FortiClient VPN Only 版本安装

1.下载 FortiClient

参考：VPN 技术 → FortiClient VPN → FortiClient 下载 。

2.安装 FortiClient

双击 FortiClientVPNSetup_7.4.3.1761_macosx.dmg（版本号仅举例使用，请根据实际下载的版本文件执行），双击“Install.mpkg”。



点击“继续”。



点击继续并在弹出窗口中点击“同意”。



选择安装位置，并点击“继续”。



点击“安装”，在弹出窗口中输入系统用户名和密码授权安装。



安装完成后，关闭安装程序。



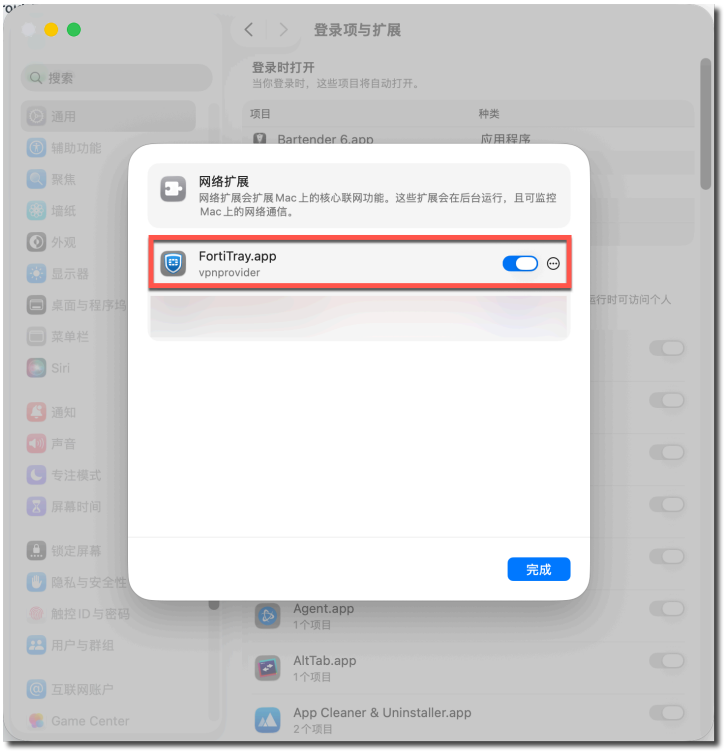
在弹出的“添加 VPN 配置”窗口中点击允许。



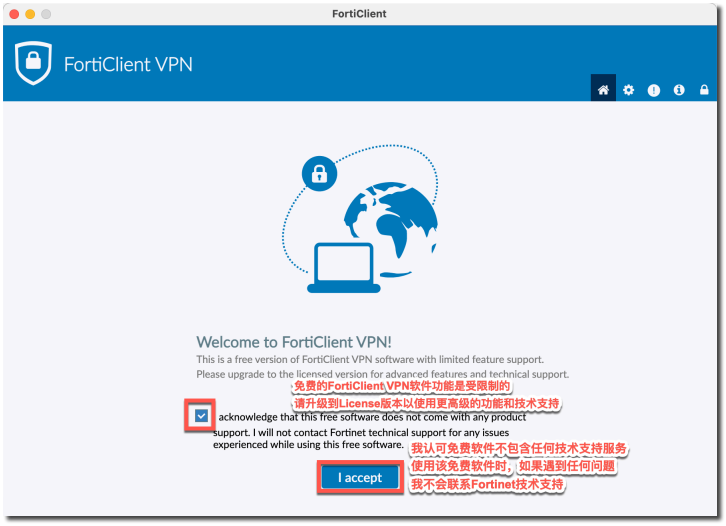
在弹出的网络扩展使用请求窗口中点击“打开系统设置”（点击其中一个即可）。



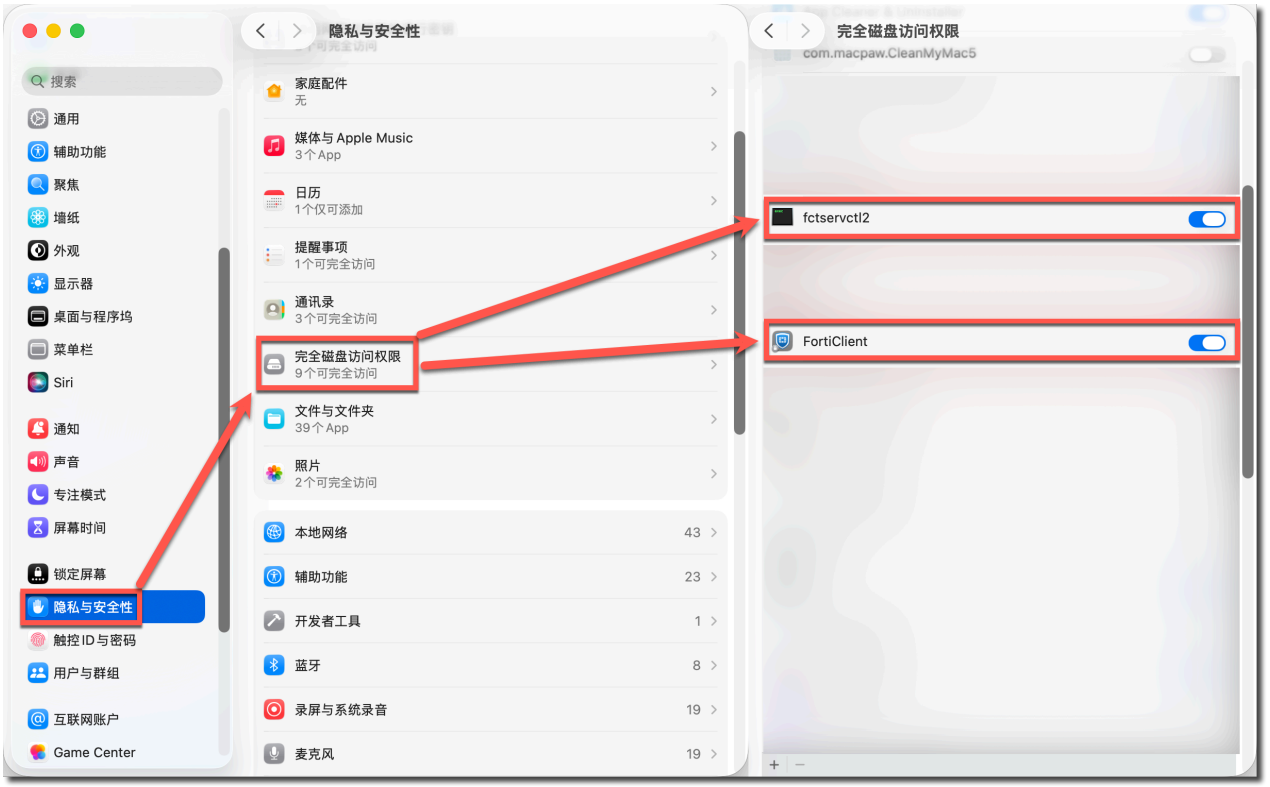
打开授权以下权限（需要系统账户验证）。



3. 打开 FortiClient，勾选“I acknowledge that this free software...”，点击 I accept。请注意免费的 FortiClient 不提供技术支持服务。

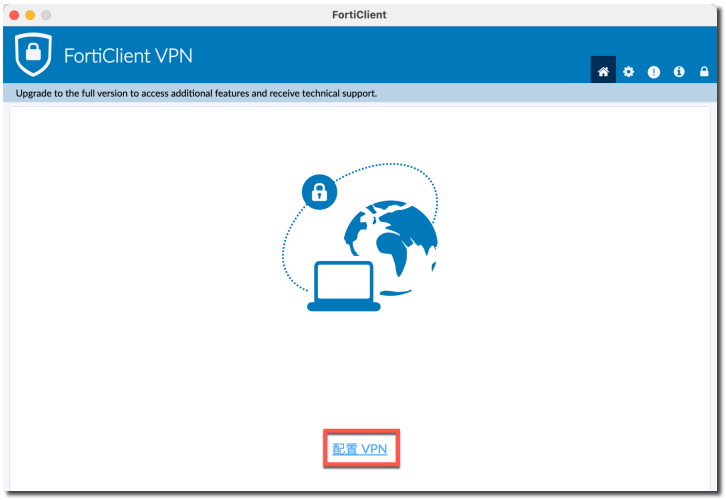


确认在“隐私与安全性 → 完全磁盘访问权限”窗口中授予 FortiClient 的磁盘访问权限（需要系统账户验证）。



设置 SSLVPN

1. 点击“配置 VPN”：



2. 配置 SSLVPN：

   VPN：VPN 类型

   连接名：VPN 名称

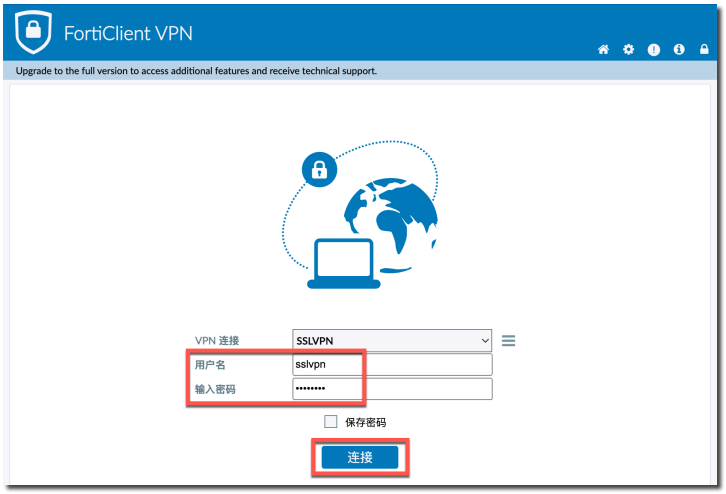
   远程网关：FortiGate 开启 VPN 服务的接口地址

   自定义端口：SSLVPN 服务的端口号

   用户名：VPN 用户名称



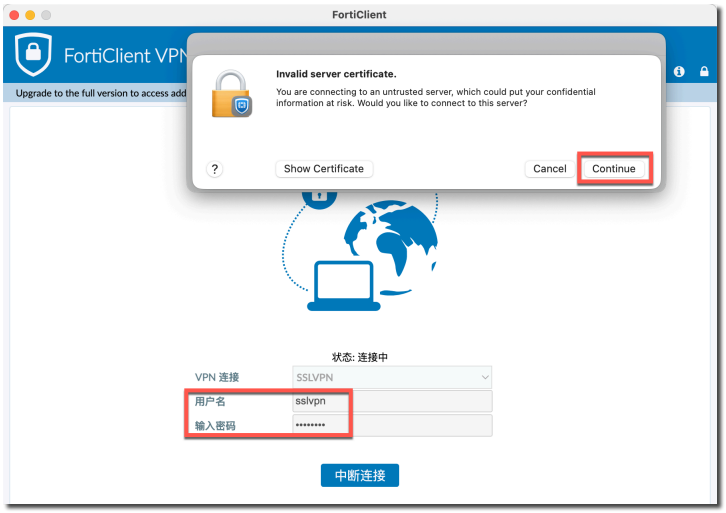
3. 配置完成后，输入用户名和密码，点击连接。



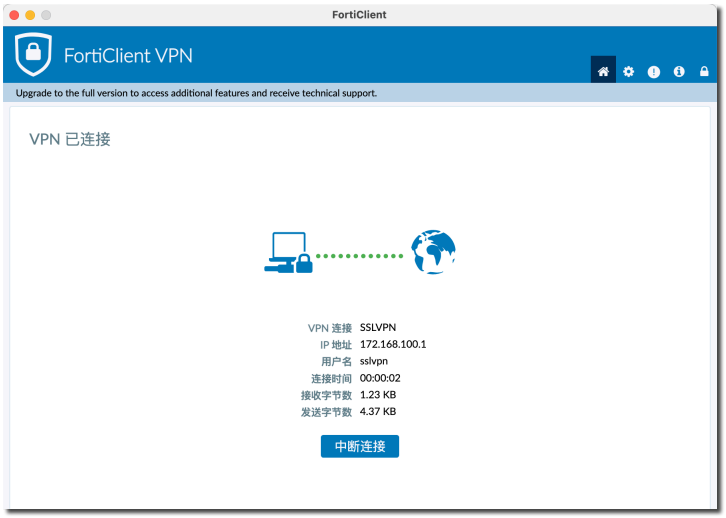
4. 如弹出如下查找本地网络设备的窗口，请点击允许。



5. 在弹出窗口中点击 Continue，接受 FortiGate 的证书。



6. SSLVPN 连接成功。

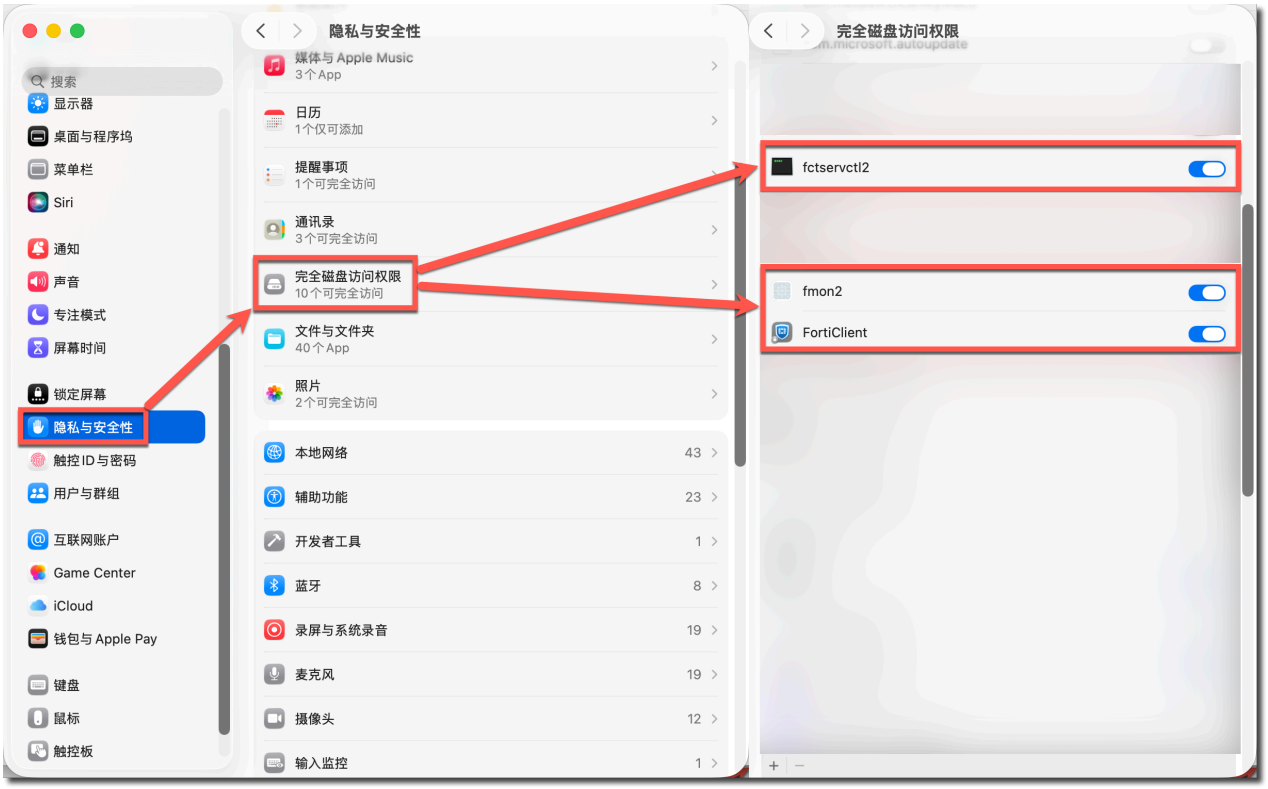


macOS 权限要求

在 macOS 系统下，FortiClient 需要允许一些系统权限才能正常使用。如果在安装过程中没有允许相关权限，可以通过如下方式允许 FortiClient 要求的相关权限。

FortiClient 完整版本

1. 设置 → 隐私与安全性 → 完全磁盘访问权限，开启 fctservctl2、fmon2、FortiClient 的权限。



2. 设置 → 通用 → 登录项与扩展，开启 ForticlientAgent.app 与 Fortinet, Inc 的权限。

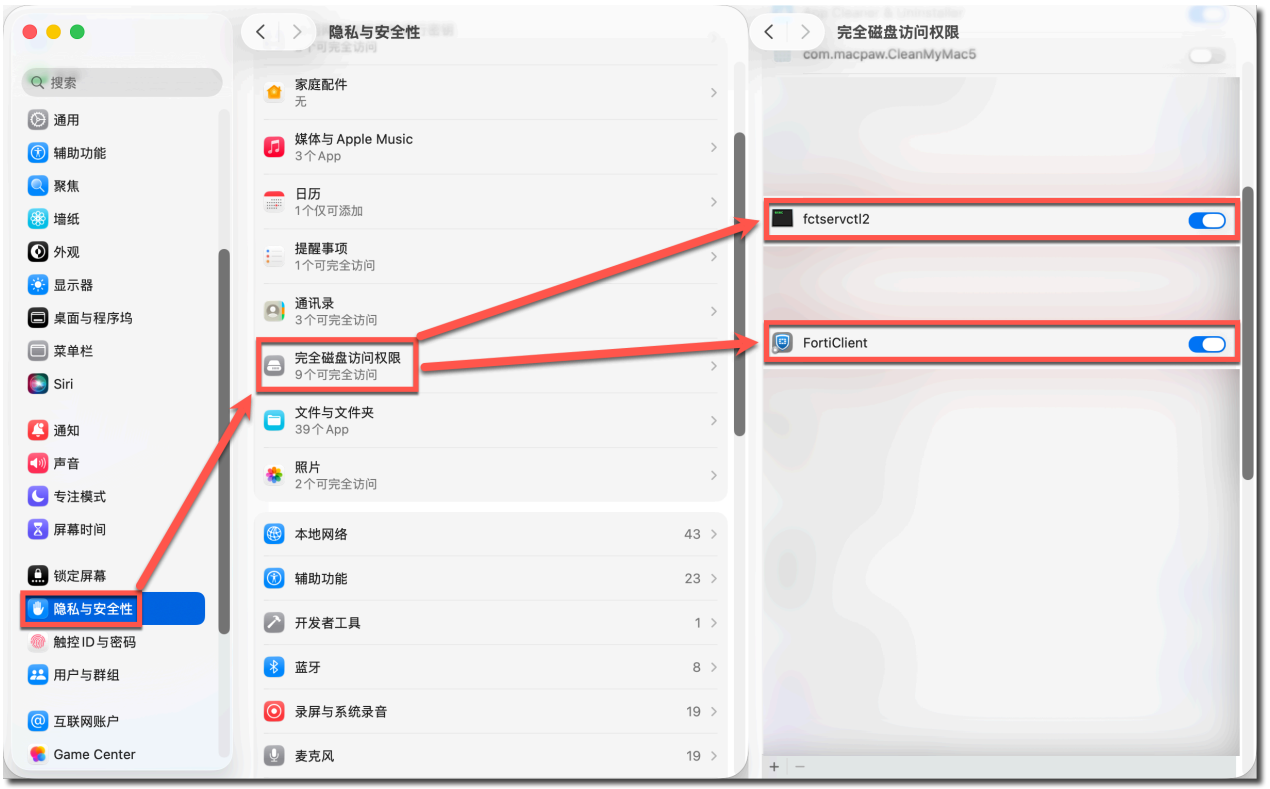


3. 设置 → 通用 → 登录项与扩展 → 拓展 → 按类别 → 网络扩展，开启 FortiTray.app、FortiClientNetwork.app、FortiClientProxy、FortiClientPacketFilter 的权限。



FortiClient VPN Only 版本

1. 设置 → 隐私与安全性 → 完全磁盘访问权限，开启 fctservctl2、FortiClient 的权限。



2. 设置 → 通用 → 登录项与扩展，开启 ForticlientAgent.app 与 Fortinet, Inc 的权限。



3. 设置 → 通用 → 登录项与扩展 → 拓展 → 按类别 → 网络扩展，开启 FortiTray.app 的权限。



iPhone FortiClient 安装及使用

说明

FortiExplorer VPN 是 Fortinet 公司在 App Store 中国区上线的 SSL VPN 连接客户端。它可以方便的下载到 iPhone 和 iPad 上，然后连接 FortiGate SSL VPN 网关，实现远程办公。

> **重要**
>
> 如果您的 Apple ID 属于非中国区，也可以从 Apple Store 直接下载 FortiClient 或 FortiClient VPN 使用。

配置文档请参考：FortiExplorer VPN 使用指南 v1.0 ↗。

Android FortiClient 安装及使用

Android FortiClient 完整版安装

1.下载 FortiClient：

参考：VPN 技术 → FortiClient VPN → FortiClient 下载 ↗。

2.安装 FortiClient

可以通过邮件，QQ 等方式将 FortiClient 发送给 Android 手机，点击安装（版本号仅举例使用，请根据实际下载的版本文件执行）。



允许从 QQ 安装。


17:40
94
允许“QQ”安装应用吗?
外部来源安装的应用未经过系统安全检测，是否允许安装此来源的应用?允许后安装时会联网检测。
允许本次安装
以后都允许
取消

点击继续安装。


17:40
FortiClient
版本：7.0.6.0049 大小：20.5 MB
来自"QQ" | 权限
未发现风险
继续安装
取消安装

安装完成。

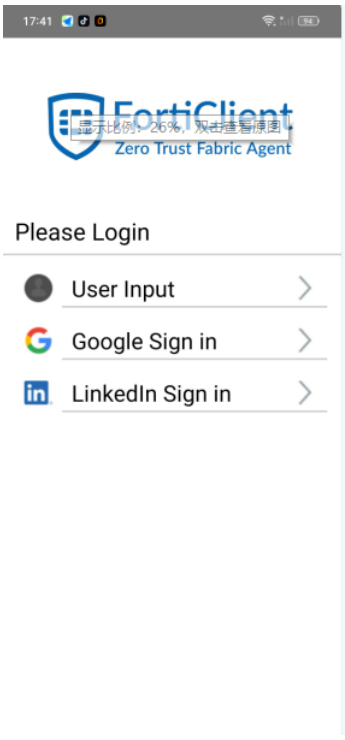


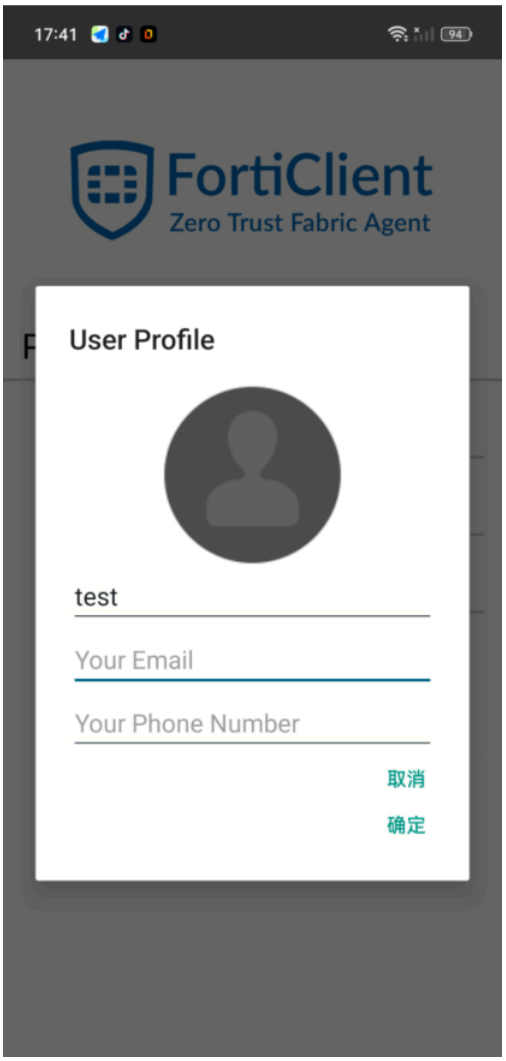
3.连接 EMS

打开 FortiClient，允许所需的权限。

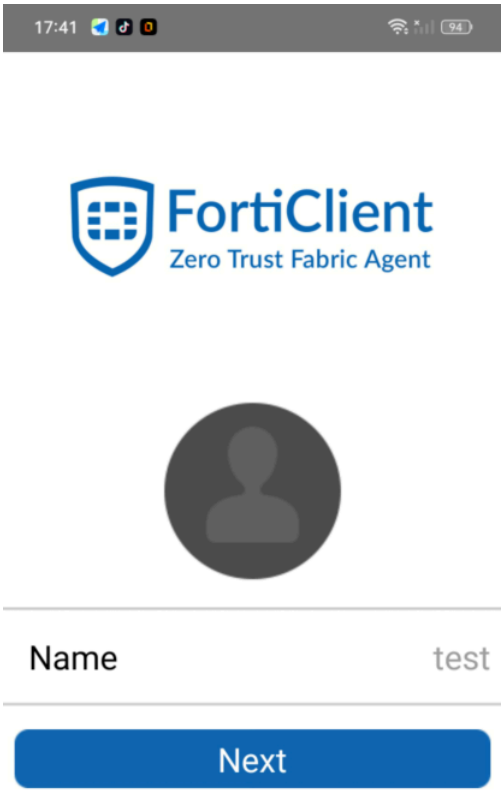


点击 User Input，输入 FortiClient 名称。

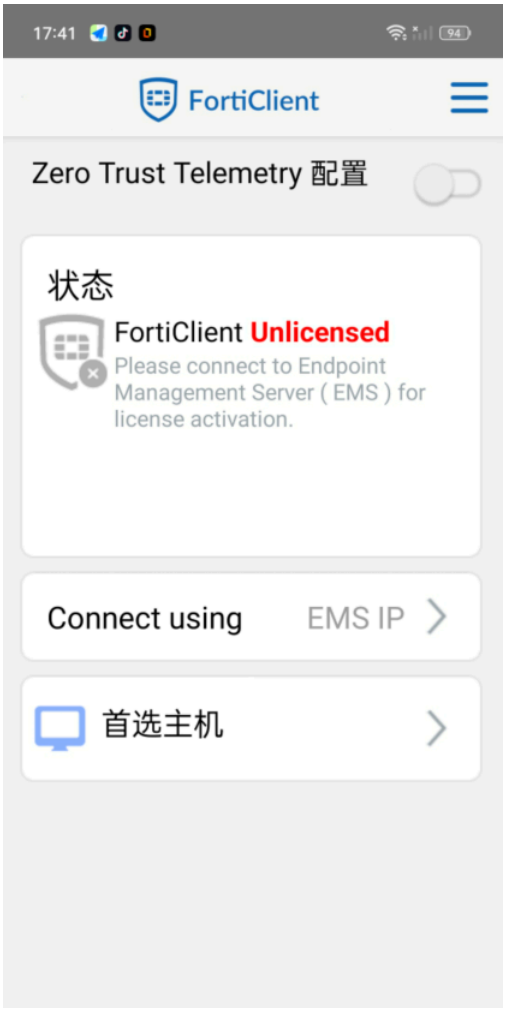



17:41
94
FortiClient
Zero Trust Fabric Agent
User Profile
test
Your Email
Your Phone Number
取消
确定

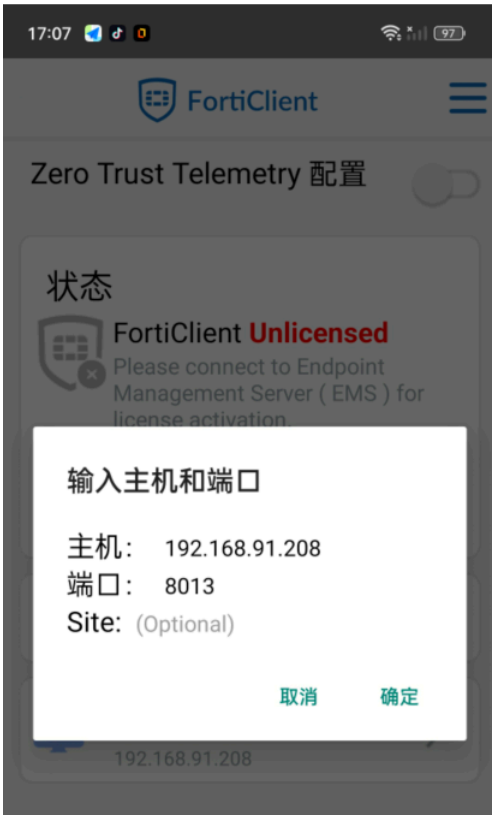
点击确定。



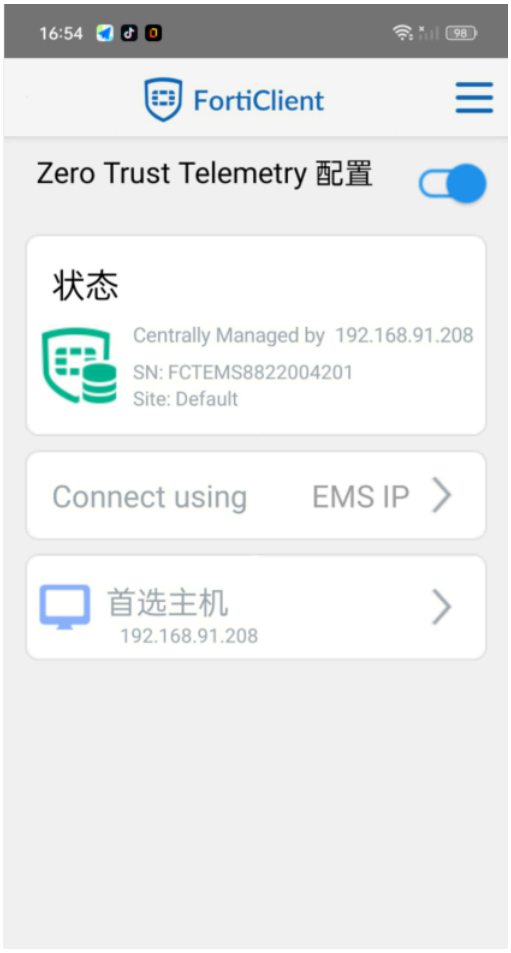
点击 Next。



点击“首选主机”，在“主机”中输入 EMS 的 IP 地址，端口默认是 8013。



EMS 已连接。



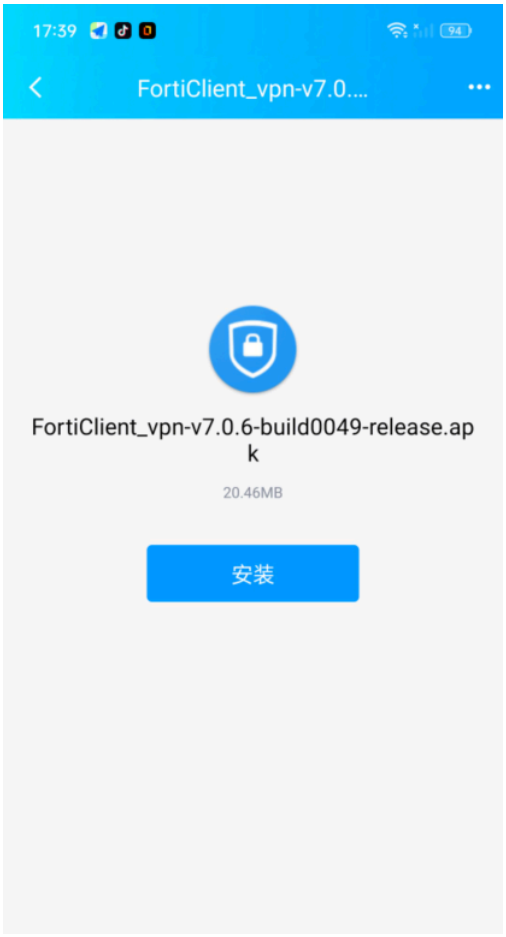
免费版 FortiClient VPN 安装

1.下载 FortiClient

参考：VPN 技术 → FortiClient VPN → FortiClient 下载 ↗。

2. 安装 FortiClient

通过邮件、QQ 等方式将 forticlient 发送给手机，点击安装（版本号仅举例使用，请根据实际下载的版本文件执行）。



允许 FortiClient 从 qq 安装。



点击继续安装。


17:40
FortiClient
版本：7.0.6.0049 大小：20.5 MB
来自"QQ" | 权限
未发现风险
继续安装
取消安装

安装成功。



设置 SSLVPN

1.配置 SSLVPN

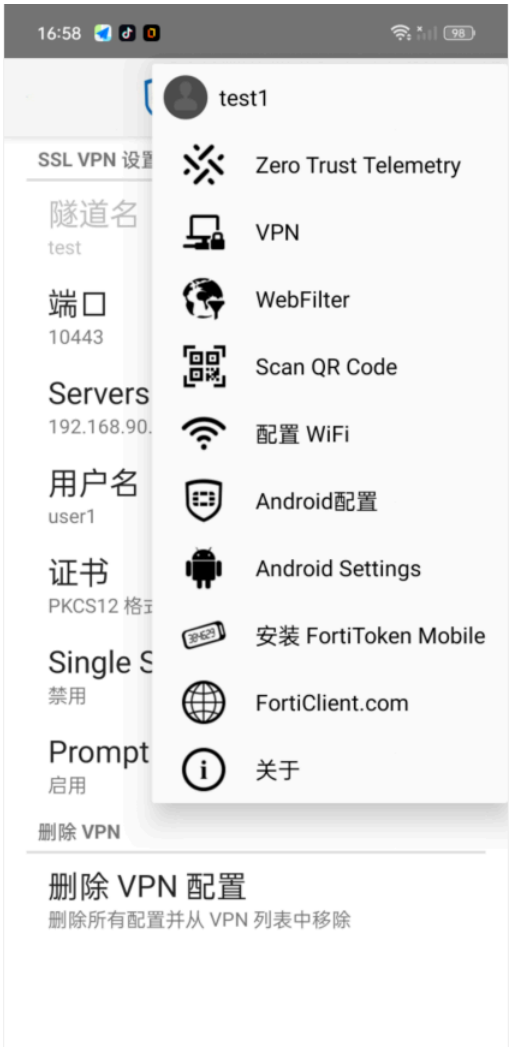
输入 SSLVPN 名称，选择 SSL VPN。



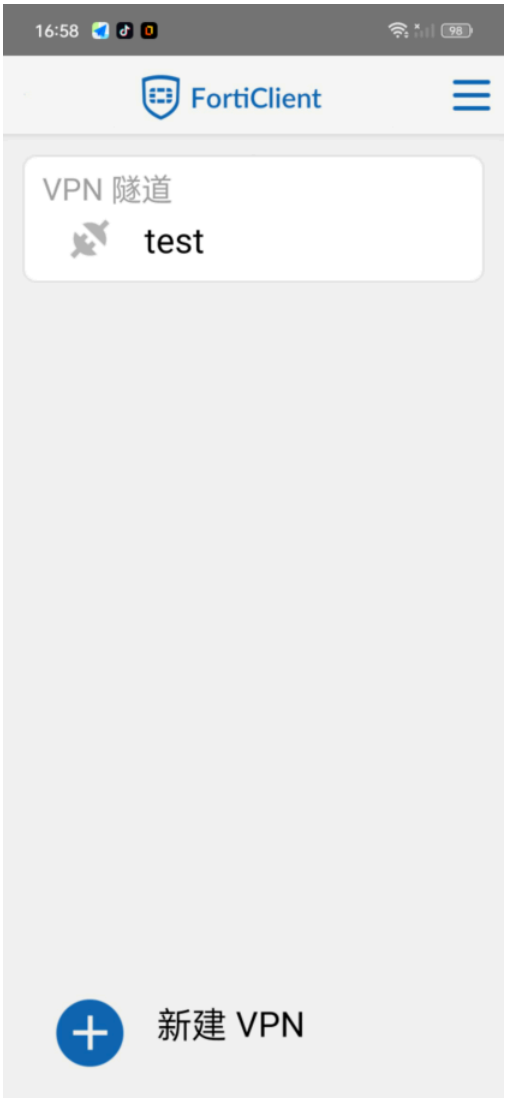
在 Servers 中输入 FortiGate 的 SSLVPN 侦听的 IP，端口中输入 SSLVPN 端口，输入用户名。



点击右上角三横杠，选择 VPN，切换到 VPN 连接。


16:58
test1
SSL VPN 设置
隧道名
test
端口
10443
Servers
192.168.90.
用户名
user1
证书
PKCS12 格式
Single S
禁用
Prompt
启用
删除 VPN
删除 VPN 配置
删除所有配置并从 VPN 列表中移除
Zero Trust Telemetry
VPN
WebFilter
Scan QR Code
配置 WiFi
Android配置
Android Settings
安装 FortiToken Mobile
FortiClient.com
关于

配置完成。



2.SSLVPN 连接

选择 SSLVPN 名称 test，点击“连接”。



证书由 FortiGate 自签发，选择“继续”。



输入密码



连接成功。


17:06
FortiClient
SSL VPN
test
Username: user1
当前会话： 0:08
远端： 192.168.90.31
本地： 10.212.134.200
断开

FortiClient 连接问题

本文介绍 SSL VPN 以特定百分比停止协商的常见错误原因与解决办法。

10% 断开连接

这种情况下 FortiClient 通常会弹出“Unable to establish the VPN connection. The VPN server may be unreachable.”的提示信息，通常是由于网络连接问题引起的。

1. 请检查 FortiClient 到 FortiGate 的 SSL VPN 端口的网络联通性（不只是使用 Ping 测试，而是需要确保 FortiClient 和 FortiGate 之间 TCP 的 SSL VPN 端口是否正常连通，可以尝试使用客户端 telnet SSL VPN 服务器的 IP 和端口，或通过浏览器访问 SSL VPN 的 HTTPS 地址以确认联通性）。
2. 检查 FortiClient 是否配置了正确的 SSL VPN IP 和端口号，并启用了 SSL VPN 功能。



3. 检查 FortiGate 是否配置了服务器证书。
4. 检查 FortiGate 是否配置了以 `ssl.<vdom_name>` 为入接口的防火墙策略。
5. 检查 FortiGate 的 VIP 是否使用了 SSL VPN 的 IP/端口。
6. 检查基于地理位置/源 IP 的限制设置，这些设置可能位于 SSL VPN 配置（参考“VPN 技术 → SSL VPN → SSL VPN 源地址连接控制↗”章节）或 local-in 策略配置（参考“策略与对象 → 本地流量策略 → Local-in 策略↗”章节）中。
7. 如果在 FortiGate 之前还有其他路由/NAT 设备，请检查路由器的端口转发是否正确映射了 SSL VPN 的 IP/端口。
8. 检查 `ssl.<vdom_name>`（如 ssl.root）接口是否为 down 状态，如果被认为关闭，请重新启用。

```bash
config system interface
    edit "ssl.root"
        set vdom "root"
        set status up
    next
end
```

9. 抓取 FortiClient 与 FortiGate 之间的 SSL VPN 报文

```bash
diagnose sniffer packet any ' host <source public IP address> and port <SSL VPN port> ' 4 0 l
```

   - 如果 FortiGate 响应了客户端请求，当 TCP 三次握手未完成或客户端在 TCP 三次握手完成后 reset 连接时，连接将终止，FortiClient 将显示连接在 10% 停止。
   - 如果流量到达 FortiGate，但 FortiGate 未做出响应，可以使用 debug flow 来判断是否匹配了拒绝策略，或因未匹配任何允许策略而被隐式拒绝。

```bash
diagnose debug disable
diagnose debug reset
diagnose debug flow show iprope en
diagnose debug flow show func en
diagnose debug flow filter addr <source public ip addr>
diagnose debug flow filter port <SSL VPN port>
diagnose debug flow trace start 300
diagnose debug enable
```

31% 断开连接

通常 FortiClient 会弹出“Unable to establish the VPN connection. The VPN server may be unreachable or your identity certificate is not trusted. (-5029)”的提示信息。这是由于 TLS 版本协商失败导致，请检查 FortiGate 配置的 SSL VPN TLS 版本是否被 FortiClient 所在的 PC 使用。

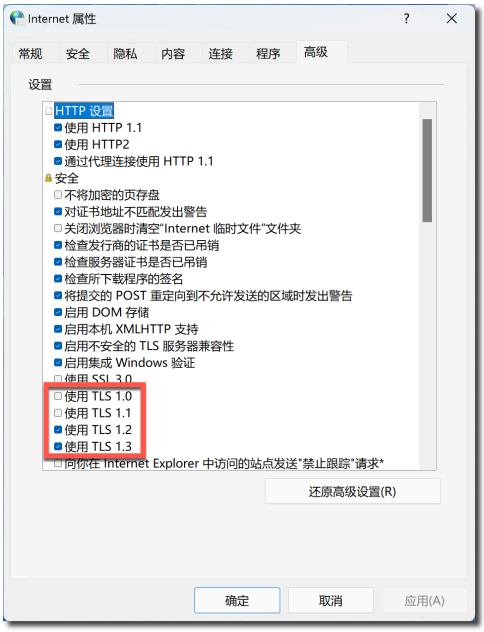
1. 检查 FortiGate 配置的 SSL VPN TLS 版本，如下所示，SSL VPN 使用的 TLS 版本最大为 1.3（ssl-max-proto-ver），最小为 1.2（ssl-min-proto-ver），也就是只接受 TLS 1.2 ~ 1.3 的版本（如果开启了 DTLS，需要参考 dtls-max-proto-ver 和 dtls-min-proto-ver）：

```bash
FortiGate # config vpn ssl settings
FortiGate (settings) # show full-configuration | grep proto
    set ssl-max-proto-ver tls1-3
    set ssl-min-proto-ver tls1-2
    set dtls-max-proto-ver dtls1-2
    set dtls-min-proto-ver dtls1-0
```

2. 可以根据实际情况修改 FortiGate 的 TLS 设置：

```bash
FortiGate # config vpn ssl setting
FortiGate (settings) # set ssl-max-proto-ver ?
tls1-0    TLS version 1.0.
tls1-1    TLS version 1.1.
tls1-2    TLS version 1.2.
tls1-3    TLS version 1.3.

FortiGate (settings) # set ssl-min-proto-ver ?
tls1-0 TLS version 1.0.
tls1-1 TLS version 1.1.
tls1-2 TLS version 1.2.
tls1-3 TLS version 1.3.
```

3. 检查 FortiClient 所在 PC 的 TLS 版本设置，这里以 Windows 为例，打开“Internet 选项”，进入高级选项卡，在“安全”部分中找到 TLS 版本相关选项，如下图所示，选取的 TLS 版本为 1.2 和 1.3，可根据 FortiGate 端的配置进行修改。



40% 断开连接

1. 可能是由于 FortiClient 在连接时弹出了是否希望继续使用非受信任的 TLS/SSL 证书的弹窗，但用户未选择或选择了取消（有时候这个弹窗没有弹出，需要在任务栏点开）。



2. 与 31% 中断类似，FortiClient 会弹出“Unable to establish the VPN connection. The VPN server may be unreachable or your identity certificate is not trusted. (-5029)”的提示信息，这由 TLS 版本协商失败导致。可参考 31% 中断的解决方法。
3. 可能是因为 SSL VPN 使用的服务器证书已过期。请更换服务器证书，或选择“Fortinet_Factory”证书作为 Workaround。
4. 如果使用 SAML 认证，请确保从 IDP 下载了正确的证书，并将其导入到 SP（FortiGate）中。
5. 如果出现“The VPN server may be unreachable. (-5)' errors”错误，可能是由于 FortiClient 所在 PC 安装的某些安全应用阻断了 FortiClient 与 FortiGate 的 SSL 协商，或是外部安全设备进行了阻断。

42% 或 43% 断开连接

通常是用户权限问题（sslvpn_login_permission_denied）。

- 对于本地用户，可能是用户名/密码输入错误。
- 对于远程用户，在 `diagnose debug application sslvpn -1` 和 `diagnose debug application fnbamd -1` 调试中可能有以下错误：“find_matched_usr_grps - Failed group matching”。以下 debug 信息可用于验证用户是否属于正确的 LDAP 用户组，将用户添加到正确的组中可解决此问题。

```
diagnose debug application sslvpn -1
diagnose debug application fnbamd -1
diagnose debug enable
```

45% 断开连接

1. 对于本地用户，可能是用户名/密码输入错误。
2. 对于远程认证（如 LDAP）用户，可能的原因包括：FortiGate 无法连接到远程服务器、远程服务器返回身份验证失败，或 LDAPS 的加密连接失败。举例：如果连接在 45% 停止并显示错误“ldap connection timeout”，可以尝试调整 LDAP 认证超时设置。

```bash
config system global
    set ldapconntimeout 3000
end
```

3. 远程认证也可以参考“48% 断开连接”中 SAML 认证相关的解决方法。

48% 断开连接

1. 可能是由于 `ssl.root`（`ssl.<vdom>`）接口被手动禁用（admin down），但在 `config vpn ssl settings` 配置中 `set status enable`，请检查 SSL VPN 接口和 SSL VPN 功能都处于开启状态。
2. 可能是由于双因子（two-factor）认证失败导致断开，检查双因子认证的配置与 Token 信息输入是否正确。
3. 如果 FortiClient 弹出提示“Credential or SSL VPN configuration is wrong (-7200)”，在 FortiClient 连接过程中通过 Debug 信息来判断问题原因：
    - 如果 Debug 信息中提示 `error:0A000126:SSL routines::unexpected eof while reading`。

```bash
diagnose debug application sslvpn -1
diagnose debug application fnbamd -1
diagnose debug enable

[2612:root:1b]deconstruct_session_id:716 decode session id ok, user=[guest], group=[],authserver=[],portal=[full-access],host[10.200.20.10],realm=[],csrf_token=[D3D4129C5AB9CB25CDCE01CCF8E40],idx=0,auth=1,sid=2d772154,login=1686904099,access=1686904099,saml_logout_url=no,pip=no,grp_info=[4xAcoJ],rmt_grp_info=[]
[2612:root:1b]rmt_web_auth_info_parser_common:557 authentication required
[2612:root:1b]rmt_web_access_check:776 access failed, uri=[/remote/logout],ret=4103,
[2612:root:1b]SSL state:fatal decode error (10.200.20.10)
[2612:root:0]ap_read,105, error=1, errno=0 ssl 0x7f3bb1bb6000 Success. error:0A000126:SSL routines::unexpected eof while reading
//[2612:root:1b]sslvpn_read_request_common,684, ret=-1 error=-1, sconn=0x7f3bb2854800.//
[2612:root:1b]Destroy sconn 0x7f3bb2854800, connSize=0. (root)
```

- 首先确认 FortiClient 使用的用户名和密码完全正确。
- 也可能是由于 PC 的 Internet 选项中的安全等级设置为“高”导致，通过调整 Internet 选项中的安全等级为“中 - 高”或“中”可以修复此问题。



- 如果 Debug 信息中提示 `failed [sslvpn_login_cert_checked_error]`。

```bash
diagnose debug application sslvpn -1
diagnose debug enable

2023-11-02 14:15:59 [11178:root:34]req: /remote/logincheck
2023-11-02 14:15:59 [11178:root:34]rmt_web_auth_info_parser_common:470 no session id in auth info
2023-11-02 14:15:59 [11178:root:34]rmt_web_access_check:723 access failed, uri=[/remote/logincheck],ret=4103,
2023-11-02 14:15:59 [11178:root:34]User Agent: FortiSSLVPN (Windows NT; SV1 [SV{v=02.01; f=07;}])
2023-11-02 14:15:59 [11178:root:34]rmt_logincheck_cb_handler:1189 user 'test15' has a matched local entry.
//2023-11-02 14:15:59 [11178:root:34]login_failed:384 user[test15],auth_type=32768 failed [sslvpn_login_cert_checked_error]//
2023-11-02 14:15:59 [11178:root:0]dump_one_blocklist:84 status=1;host=15.15.15.15;fails=1;logintime=1698730139
```

- 这可能是由于误开启了 FortiGate SSL VPN 的“需要客户端证书”功能，而实际不需要证书认证客户端证书。需要手动关闭该功能。




```bash
config vpn ssl settings
    set reqclientcert disable
end
```

- 如果 FortiGate 使用了 LDAP 作为 SSL VPN 认证，请参考用户与认证 → LDAP 认证 → LDAP 常见错误代码 ↗章节排查。

4. 如果出现“Permission denied. (-455)”错误，检查用户是否包含在“SSL VPN 认证/门户配置”中的用户组中。

5. 使用 SAML 认证时，FortiClient 会弹出登录窗口，此登录窗口存在远程认证超时时间。如果在此时间内没有完成认证（例如 SAML 认证同时开启了双因子认证，默认的计时器时间可能不够），FortiClient 将在 48% 时断开连接。`remoteauthtimeout` 默认为 5s，可通过以下 CLI 命令将其适当增加。

```bash
config system global
    set remoteauthtimeout 120    ←——FortiGate等待远程身份验证服务器响应的秒数，1~480s，默认5s
end
```

- 请注意 `remoteauthtimeout` 的设置时间并非 SSL VPN 在关闭连接前实际等待的时间，实际等待时间 = 10 * remoteauthtimeout + 30s。
- 如果同时开启了双因子认证，双因子认证也有一个认证超时时间 `two-factor-email-expiry`。

```bash
config system global
    set two-factor-email-expiry 60    ←——FortiGate等待双因子认证的超时秒数，30~300s，默认60s
end
```

- 在 SSL VPN 同时使用 SAML + 双因子认证时，SSL VPN 在关闭连接前实际等待的时间以 10 × remoteauthtimeout + 30s 为准。即使 `two-factor-email-expiry` 更长。
- 示例：如果将 `two-factor-email-expiry` 设置为 240s，则 `remoteauthtimeout` 必须大于或等于 21s。 240 = 10 × remoteauthtimeout (21) + 30。
- 对于使用 Azure SAML 的 SSL VPN 身份验证，`remoteauthtimeout` 时间将翻倍。例如，当设置为 30s 时，实际的等待时间将变为 30 × 2 × 10 + 30 = 630s 。

6. 如果出现“Host check requirement error (-455)”错误。如果 SSL VPN 没有开启主机检查（host check），请检查防火墙是否发生了“sslvpnd”进程崩溃。

```
diagnose debug crashlog read
```

7. 检查 SSL VPN 是否开启了 MAC 地址检查，如果开启了 MAC 地址检查，请检查连接的 FortiClient 所在的 PC 的 MAC 地址是否在 FortiGate 配置的允许列表中。

```bash
config vpn ssl web portal
    edit <portal_name>
        set mac-addr-check enable
         config mac-addr-check-rule
            edit <rule_name>
                set mac-addr-list <address> [address]
                set mac-addr-mask <mask between 1-48>
```

```
            next
        end
        set mac-addr-action {allow | deny}
    next
end
```

80% 断开连接

1. 检查 FortiGate 的 SSL VPN 中用于匹配用户的门户（portal，如 full-aacess）是否开启了隧道模式（tunnel mode）。
2. 如果弹出提示“Unable to log on to the server. Your username or password may not be configured properly for this connection. (-14)”，检查 FortiClient 输入的用户名和密码是否与 FortiGate 上配置的一致。
3. 在 FortiGate 检查用户和用户组的关系，可能是用户未加入正确的用户组。
4. 检查是否配置了对应用户/用户组的防火墙策略。
5. 检查 FortiGate 中 SSL VPN 设置的“认证/门户配置（Authentication/Portal Mapping）”配置中的用户/用户组是否与防火墙策略中 SSL VPN 相关策略中引用的用户/用户组完全一致，如果不一致，FortiClient 会出现以下提示（英文系统下会提示“Unable to log on to the server. Your username or password may not be configured properly for this connection. (-14)”）：



重要

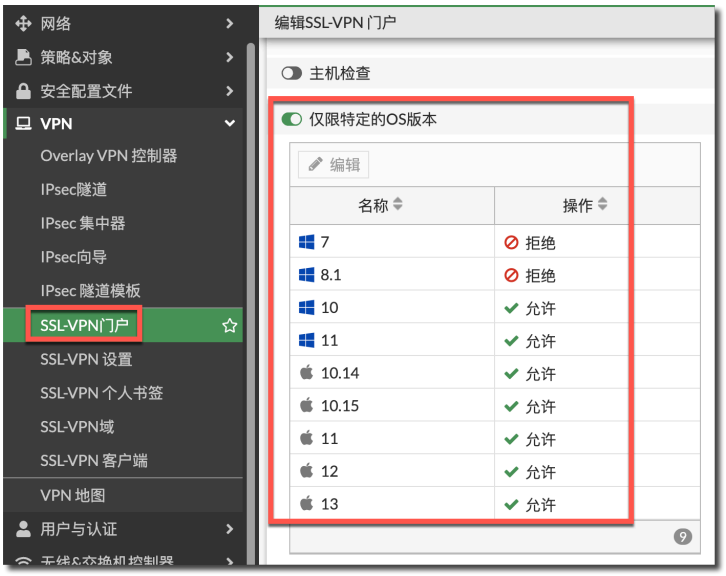
- 如果 SSL VPN 设置的“认证/门户配置”选择了用户组，那么防火墙策略中也要引用用户组，不要引用该用户组中的用户成员。
- 如果 SSL VPN 设置的“认证/门户配置”选择了用户，那么防火墙策略中也要引用用户，不要引用该用户所属的用户组。
- 如果有多条 SSL VPN 相关的防火墙策略，无论策略的顺序如何，只要有策略中配置了“认证/门户配置”中不一致的用户/用户组，都可能导致这个问题的出现。
- 如果不按照上述规则配置，会出现 FortiClient 连接 SSL VPN 时，无法匹配自定义配置的“认证/门户配置（Authentication/Portal Mapping）”，而匹配到默认的 Portal Mapping 时：
  - 如果默认的 Portal Mapping 为“web-access”（只开启了 Web 模式的访问，没有开启 Tunnel 模式的访问），那么就会连接失败。
  - 即使默认的 Portal Mapping 选择了开启 Tunnel 模式的 Portal，FortiClient 可以成功连接 SSL VPN，但也有可能匹配到非用户预期的 Portal，所以还是要按照上述规则进行配置。



6. 检查用户/用户组是否匹配了正确的门户（Portal）。

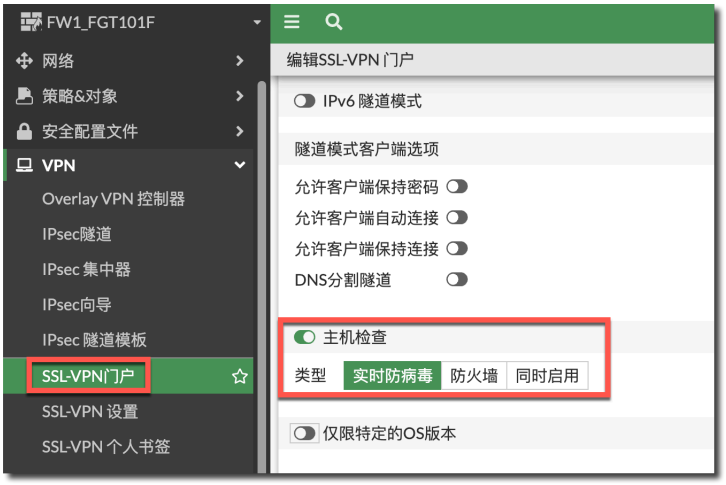
7. 如果在 SSL VPN 设置中为用户/用户组配置了域（Realm），请检查 FortiClient 是否连接了正确的域（Realm）。

8. 如果 FortiGate 上用户/用户组所属的门户（Portal）开启了 OS 版本检查，请确认客户端的 OS 类型与版本是否满足 FortiGate 上配置。




```bash
config vpn ssl web portal
    edit "full-access"
        set os-check enable
        config os-check-list "windows-7"
            set action deny
        end
        config os-check-list "windows-8.1"
            set action deny
        end
        config os-check-list "windows-10"
        end
        config os-check-list "windows-11"
        end
        config os-check-list "macos-mojave-10.14"
        end
        config os-check-list "macos-catalina-10.15"
        end
        config os-check-list "macos-bigsur-11"
        end
        config os-check-list "macos-monterey-12"
        end
        config os-check-list "macos-ventura-13"
        end
    next
end
```

9. 如果 FortiGate 上用户/用户组所属的门户（Portal）开启了主机检查，请确认客户端的主机是否通过了 AV/防火墙/MAC 地址检查。




```bash
config vpn ssl web portal
    edit "full-access"
        set host-check av
        set mac-addr-check enable
        config mac-addr-check-rule
            edit "BingBai"
                set mac-addr-list 0c:37:96:06:d7:5d
            next
        end
    next
end
```

.0. FortiGate 固件降级可能会导致 SSL - VPN 门户（portal）设置初始化，因此请尽量避免版本降级，或确认降级后 portal 配置是否保留。

89% 断开连接

1. 在此阶段，FortiTray.exe 将尝试向位于 `%localappdata%\FortiClient` 目录下的 `fortisslvpn_xml.txt` 文件写入数据。
2. 如果无法访问 `%localappdata%\FortiClient`，将显示错误代码 `-5052`。
3. 如果 `%localappdata%\FortiClient` 目录可访问，但无法修改 `fortisslvpn_xml.txt` 文件，会出现错误代码 `-5053`。
4. 在 CMD 中运行如下命令可以显示为何 `fortitray.exe` 无法写入 `fortisslvpn_xml.txt` 文件。

```bash
notepad %localappdata%\FortiClient\fortisslvpn_xml.txt
```

98% 断开连接

此阶段断开连接标识身份验证已完成，但无法建立隧道。这种情况下通常是由于 FortiClient 所在主机的 OS 问题导致。

1. 尝试在主机上重装 FortiClient 程序（先卸载现有 FortiClient 程序，根据提示信息重启后再重装）。
2. 检查 FortiClient、FortiGate、EMS Server 之间的版本兼容性。请参考“VPN 技术 → FortiClient VPN → 系统兼容性列表 ↗”章节。
3. 可能由网络问题引起，例如 IPv6 到 IPv4 的连接（不支持）、高网络延迟、被阻塞的流量。

其他百分比断开连接

请参考 FortiClient 维护手册 ↗。

FortiClient 排错与卸载

FortiClient 排错

FortiClient 排错请参考链接：https://handbook.fortinet.com.cn/files/Forticlient_maintenance_manual.pdf ↗

FortiClient 卸载

FortiClient 卸载请参考链接：https://handbook.fortinet.com.cn/files/Forticlient卸载指南.pdf ↗

系统原生客户端接入

iOS

- iOS 原生客户端接入 IPSec（IKEv1）
- IKEv2

  iOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）　iOS 原生客户端接入 IPSec（IKEv2 用户名/密码 + 预共享密钥认证）

Android 系统

- Android12 以下系统（IKEv1）
- Android12 或以上系统（IKEv2）

macOS

- macOS 原生客户端接入 IPSec（IKEv1）
- IKEv2

  macOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）　macOS 原生客户端接入 IPSec（用户名/密码 + 预共享密钥认证）

Windows 系统

- Windows 原生客户端接入 IPSec IKEv1（L2TP over IPSec）
- Windows 原生客户端接入 IPSec（IKEv2）
- 使用 Windows 系统防火墙与 FortiGate 建立 IPSec VPN
- L2TP VPN
- PPTP VPN

iOS

iOS 原生客户端接入 IPSec（IKEv1）

IKEv2

- iOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）
- iOS 原生客户端接入 IPSec（IKEv2 用户名/密码 + 预共享密钥认证）

iOS 原生客户端接入 IPSec（IKEv1）

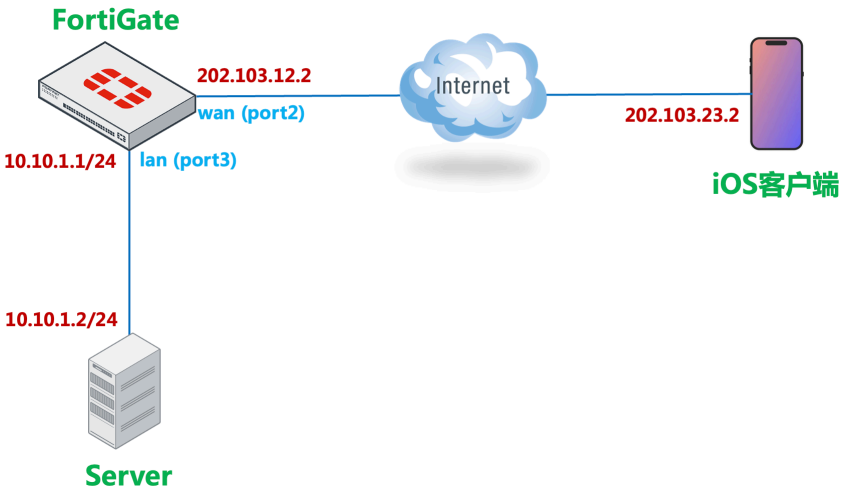
重要

本文基于 iOS 16.3 编写。

网络需求

通过 iOS 原生客户端（IKEv1）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

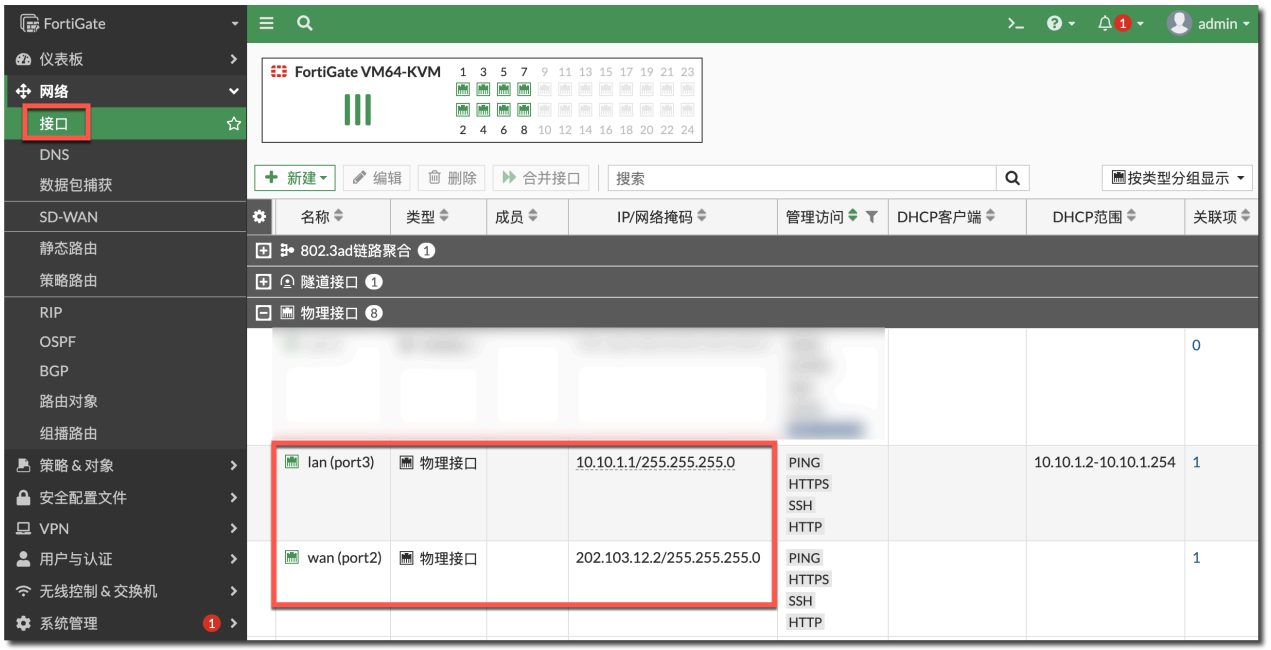
网络拓扑



配置步骤

FortiGate 配置

1. 配置 FortiGate 的接口 IP。



2. 创建本地 VPN 用户和用户组，参考用户与认证 → 本地用户和用户组 ↗章节（如需使用远程认证，请参考用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗或用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。

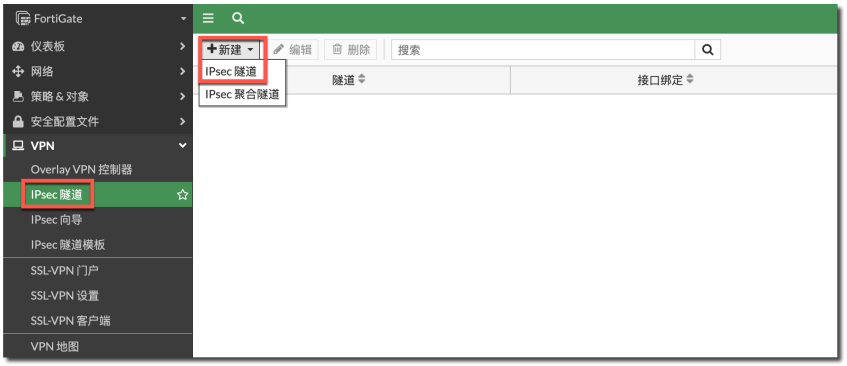



```
config user local
    edit "vpn_user"
        set type password
        set passwd xxxxxxxx
    next
end
```




```
config user group
    edit "VPN"
        set member "vpn_user"
    next
end
```

3. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



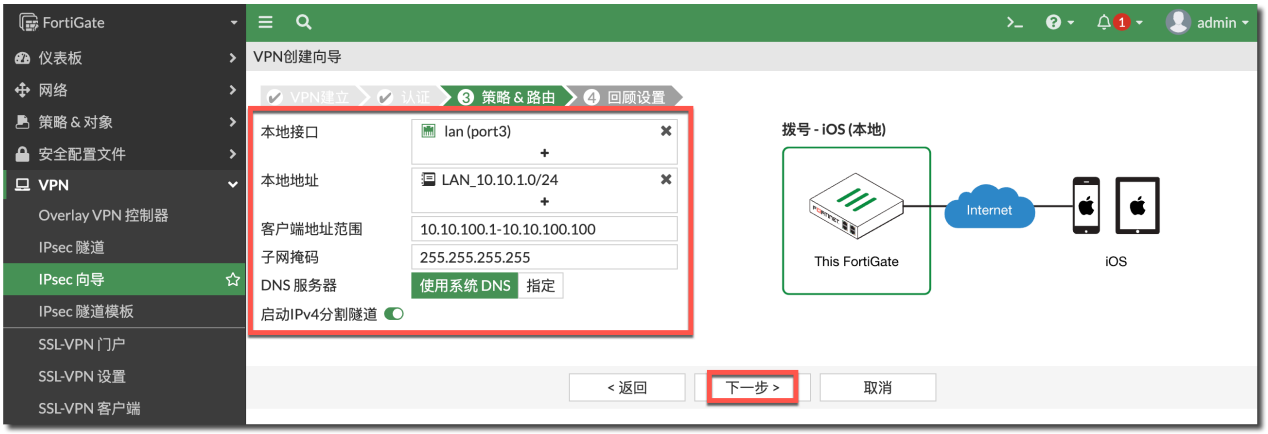
4. 填写 VPN 名称，模版选择远程拨号，远程设备选择 Native，然后选择 iOS Native，点击下一步。



5. 配置流入接口为 wan 口，填入预共享密钥，选择 VPN 用户组，点击下一步。



6. 配置本地接口（VPN 客户端要访问的网络所属出接口），本地地址新建并选择内网的 VPN 网段地址对象，分配客户端地址范围（注意不要和现有网段冲突），DNS 可选择系统 DNS 或指定 DNS，默认开启隧道分割，只有本地地址中的网段会被客户端转发至 VPN 隧道（关闭隧道分割的话，客户端会将自身发起的所有流量转发至 VPN 隧道）。点击下一步。



> **重要**
>
> 如果开启了隧道分割，本地地址不要填写为 0.0.0.0/0，否则会导致客户端连接失败。

7. 检查配置无误后，点击完成按钮，完成隧道建立。





8. 向导配置完成后，FortiGate 会自动创建 IPSec 一阶段和二阶段、内网地址对象、客户端到内网的防火墙访问策略，对应的 CLI 如下：

```
第一阶段和第二阶段VPN配置：
config vpn ipsec phase1-interface
    edit "iOS_IKEv1"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256 aes256-md5 aes256-sha1
        set comments "VPN: iOS_IKEv1 (Created by VPN wizard)"
        set dhgrp 14 5 2
        set wizard-type dialup-ios
        set xauthtype auto
        set authusrgrp "VPN"
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set dns-mode auto
        set ipv4-split-include "iOS_IKEv1_split"
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "iOS_IKEv1"
        set phase1name "iOS_IKEv1"
        set proposal aes256-sha256 aes256-md5 aes256-sha1
        set pfs disable
        set keepalive enable
        set comments "VPN: iOS_IKEv1 (Created by VPN wizard)"
    next
end
```

地址和地址组：

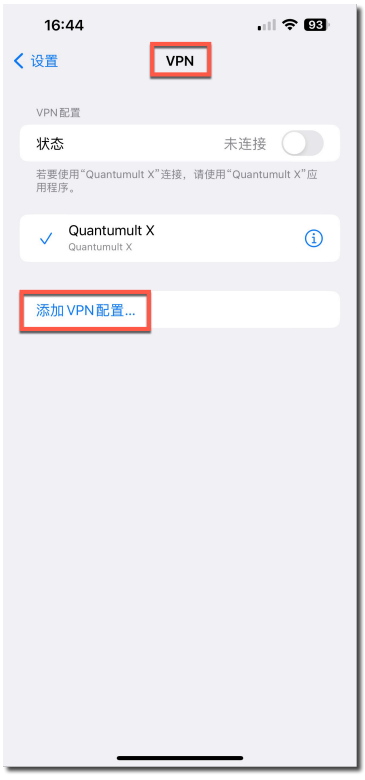
```
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "iOS_IKEv1_range"
        set type iprange
        set comment "VPN: iOS_IKEv1 (Created by VPN wizard)"
        set start-ip 10.10.100.1
        set end-ip 10.10.100.100
    next
end

config firewall addrgrp
    edit "iOS_IKEv1_split"
        set member "LAN_10.10.1.0/24"
        set comment "VPN: iOS_IKEv1 (Created by VPN wizard)"
    next
end
```

允许客户端访问内网VPN资源的防火墙策略：

```
config firewall policy
    edit 1
        set name "vpn_iOS_IKEv1_remote_0"
        set srcintf "iOS_IKEv1"
        set dstintf "port3"
        set action accept
        set srcaddr "iOS_IKEv1_range"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
        set nat enable
        set comments "VPN: iOS_IKEv1 (Created by VPN wizard)"
    next
end
```

客户端配置

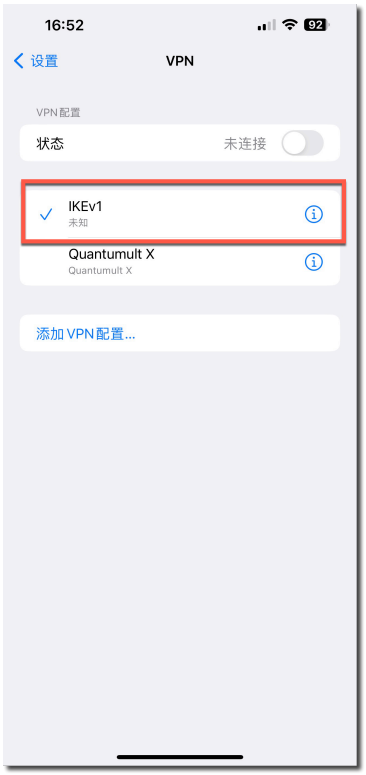
1. 打开设置 → VPN，点击最下方的“添加 VPN 配置…”按钮。



2. 类型选择 IPSec，填写 VPN 名称、服务器 IP、用户名、用户密码、预共享密钥，其他不填写（如果想要使用群组名称，请在 FortiGate 侧配置），点击完成按钮。



3. 点击创建的 VPN 连接，确保该连接被选中（连接前有一个✅）。

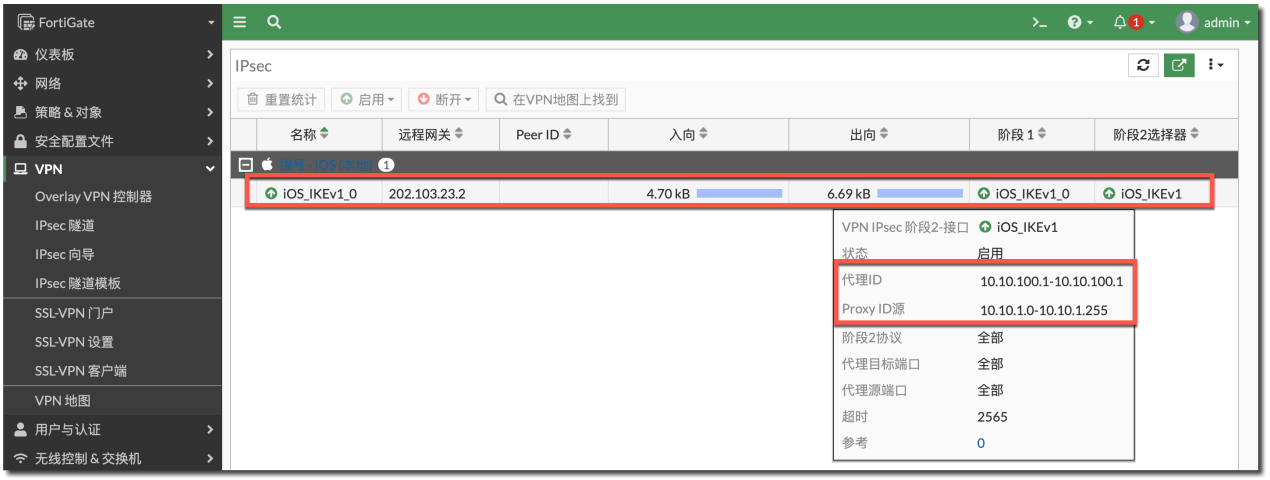


结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。



2. 在 FortiGate 侧查看连接状态。



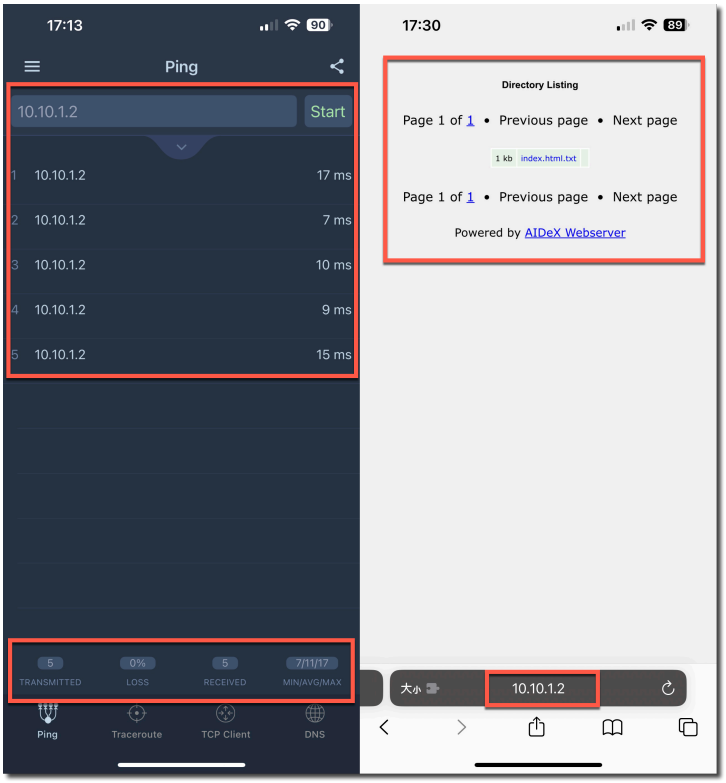

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: iOS_IKEv1_0
version: 1
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:60417
tun_id: 10.10.100.1/::10.0.0.4
remote_location: 0.0.0.0
network-id: 0
created: 1144s ago
xauth-user: vpn_user
2FA: no
assigned IPv4 address: 10.10.100.1/255.255.255.255
nat: peer
IKE SA: created 1/1  established 1/1  time 60/60/60 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms
  id/spi: 2 1ab45dd9b8ad53a1/229987eaaab4142e
  direction: responder
  status: established 1144-1144s ago = 60ms
  proposal: aes256-sha256
  key: 26d033d72c0d55fe-6652a7159cf151dd-1113c3dd29c29cf9-74dabccc059f96ac
```

```
  lifetime/rekey: 3600/2185
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=iOS_IKEv1 ver=1 serial=1 202.103.12.2:0->0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=42957570 olast=42957570 ad=/0
stat: rxp=115 txp=96 rxb=10525 txb=16656
dpd: mode=on-demand on=0 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=iOS_IKEv1_0 ver=1 serial=4 202.103.12.2:4500->202.103.23.2:60417 tun_id=10.10.100.1 tun_id6=::10.0.0.4 dst_mtu=1500
dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=iOS_IKEv1 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=873 olast=873 ad=/0
stat: rxp=51 txp=44 rxb=4704 txb=6692
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=silent draft=32 interval=10 remote_port=60417
proxyid=iOS_IKEv1 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20083 type=00 soft=0 mtu=1422 expire=2440/0B replaywin=2048
       seqno=2d esn=0 replaywin_lastseq=00000033 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3588/3600
  dec: spi=f3a58344 esp=aes key=32 a8ff490c9fb2e89be964d1a6993b11eedf71191e520ca99ed294c8eb4ea26301
       ah=sha256 key=32 7be5011c5d5a80768497ea9a4421bf8016797480cf6e31bd887d7367db35f30b
  enc: spi=05a9cf2e esp=aes key=32 83392937fdf5872826b529822a07a81c22e076d7bd648c8887451a2ce51b8336
       ah=sha256 key=32 a1897271122f887c59149acb755abc2959d1916a2c3340a35ef40f7172809d9b
  dec:pkts/bytes=102/9408, enc:pkts/bytes=88/16868
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=2 dec_npuid=0 enc_npuid=0
```

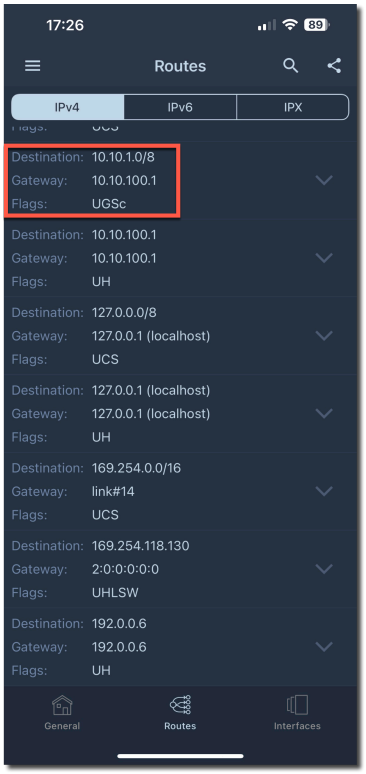
3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=25(iOS_IKEv1)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



5. 查看客户端的路由表，可以看到隧道分割的路由。



IKEv2

- iOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）
- iOS 原生客户端接入 IPSec（IKEv2 用户名/密码 + 预共享密钥认证）

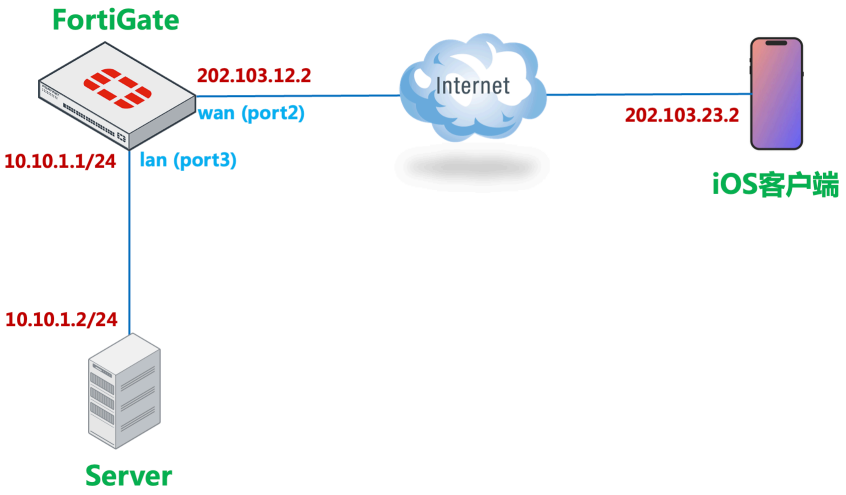
iOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）

> **重要**
>
> 本文基于 iOS 16.3 编写。

网络需求

通过 iOS 原生客户端（IKEv2 预共享密钥认证）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

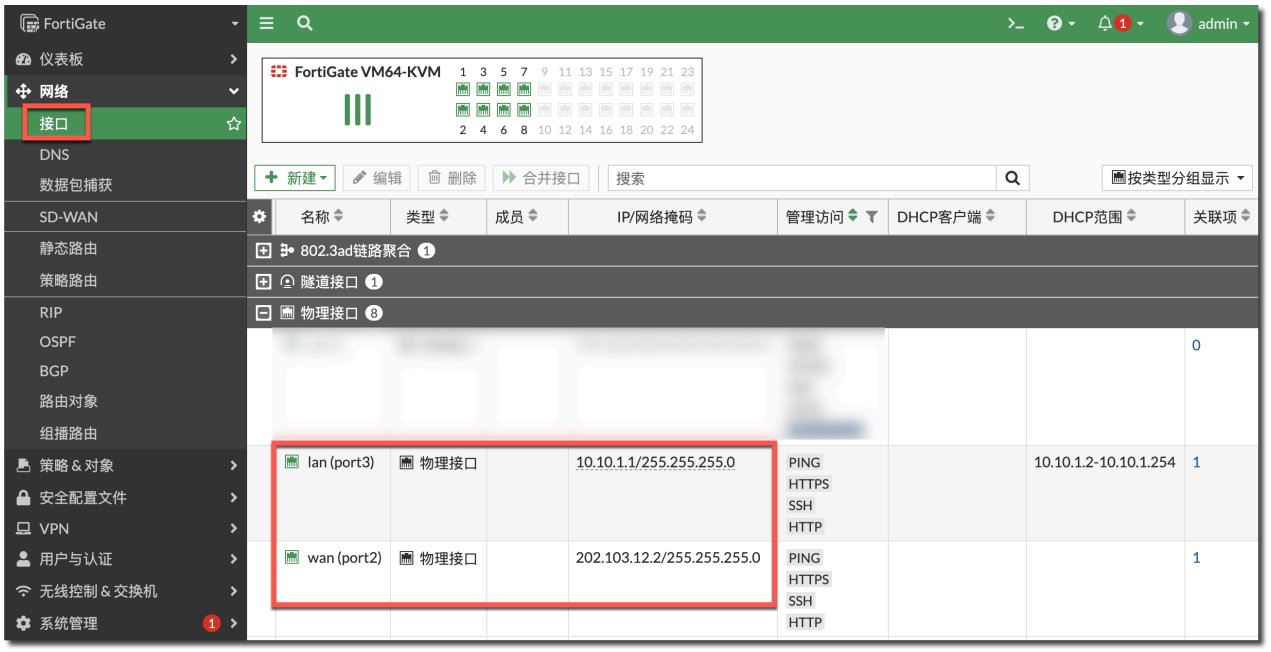
网络拓扑



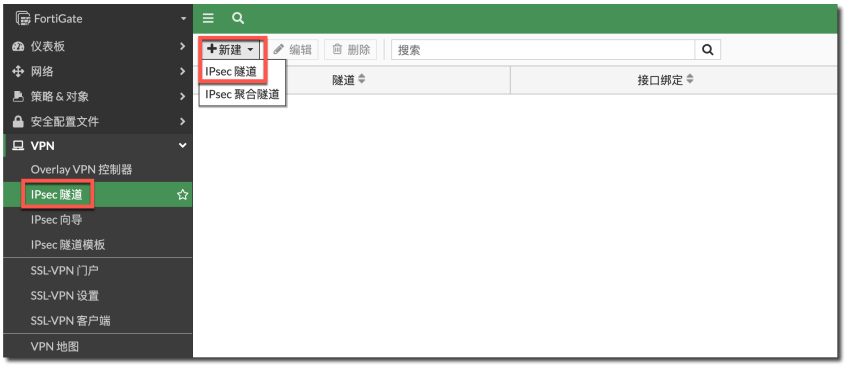
配置步骤

FortiGate 配置

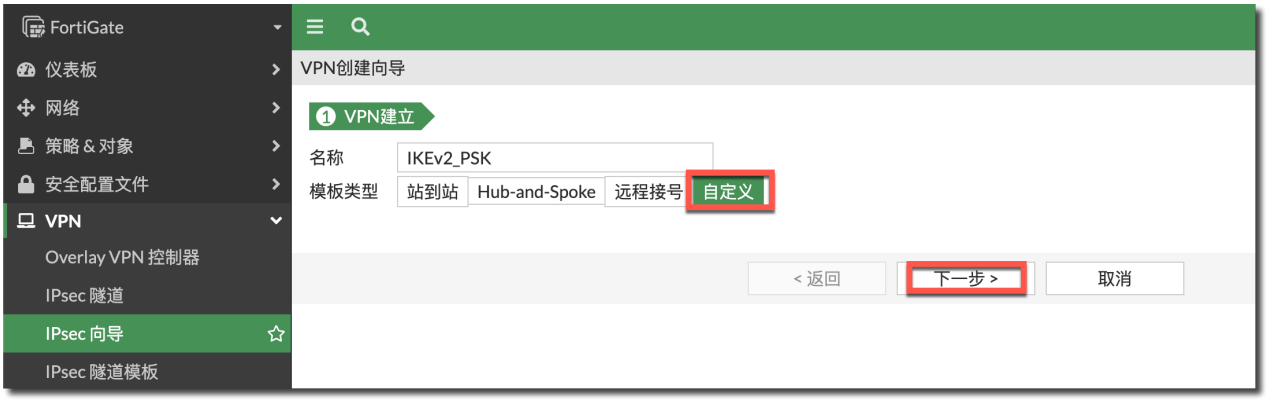
1. 配置 FortiGate 的接口 IP。



2. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



3. 填写 VPN 名称，模版选择自定义，点击下一步。



4. 按照下图修改配置，其他配置保持默认即可，点击确认下发配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.10
新VPN隧道
名称
IKEv2_PSK
注释
0/255
网络
选择为拨号用户模式
接口为wan口
IP版本
IPv4
IPv6
对端网关
拨号用户
接口
wan (port2)
本地网关
模式配置
开启模式配置
在模式配置中使用系统DNS
分配 IP从
范围
为客户端分配的IP地址范围和DNS地址
IPv4 模式配置
客户端地址范围
10.10.100.1-10.10.100.100
子网掩码
255.255.255.255
DNS 服务器
10.10.1.2
启动IPv4分割隧道
开启隧道分割，只有地址组中的地址会被客户端转发至VPN隧道
访问可达网络
LAN_10.10.1.0/24
IPv6 模式配置
客户端地址范围
::-::
前缀长度
128
DNS 服务器
::
启用IPv6隧道分割
NAT穿越
启用
禁用
强制的
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
60
s
前向纠错
Egress
Ingress
高级
填写预共享密钥
IKE版本选择2
访问类型修改为"任意对等体ID"
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
对等体选项
访问类型
任意对等体ID
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
一阶段算法
Diffie-Hellman 组
32 31 30 29 28 27
21 20 19 18 17 16
15 14 5 2 1
密钥生存时间 (秒)
86400
本地ID
阶段 2 选择器
名称
本地地址
对端地址
IKEv2_PSK
0.0.0.0/0.0.0.0
0.0.0.0/0.0.0.0
新建阶段 2
名称
IKEv2_PSK
注释
本地地址
子网
0.0.0.0/0.0.0.0
对端地址
子网
0.0.0.0/0.0.0.0
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
二阶段算法
启用重播检测
启用完全前向保密 (PFS)
关闭PFS
本地端口
全部
对端端口
全部
协议
全部
自动密钥保持存活
开启"自动密钥保持存活"
密钥周期(秒/kb)
秒
秒
43200
确认
取消



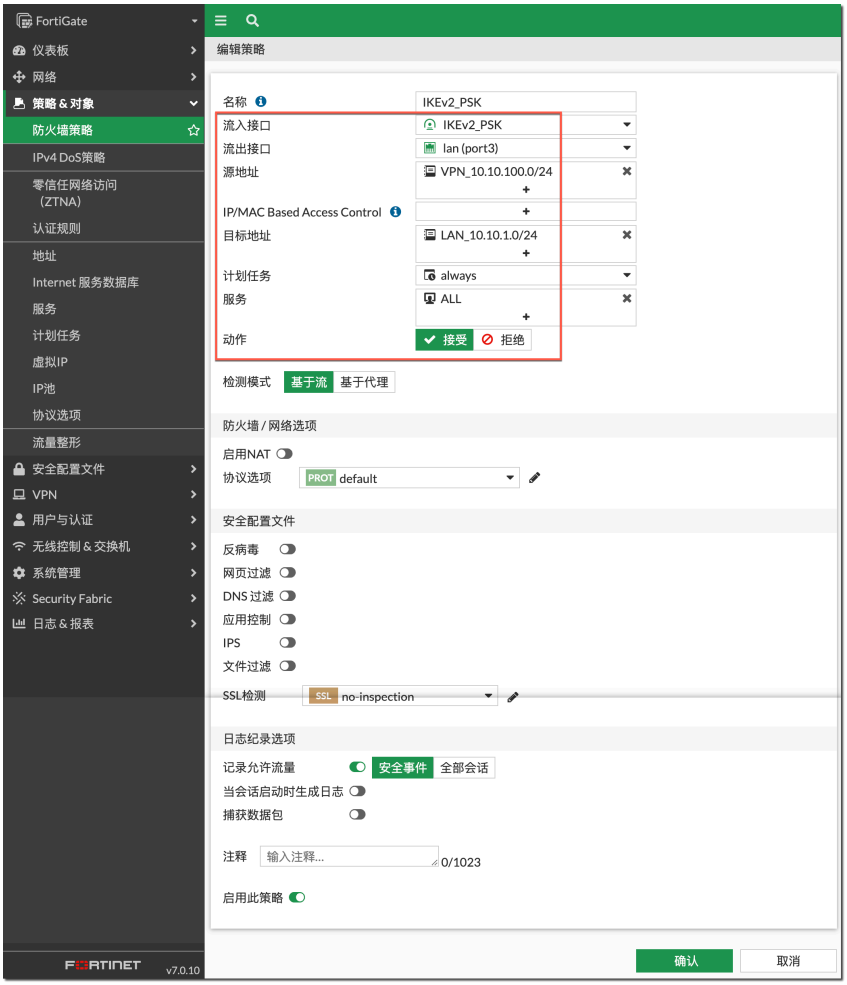
> **重要**
>
> 如果开启了隧道分割，本地地址不要填写为 0.0.0.0/0，否则会导致客户端连接失败。

5. 对应的 CLI 如下：

```
第一阶段配置：
config vpn ipsec phase1-interface
    edit "IKEv2_PSK"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 10.10.1.2
        set proposal aes128-sha256 aes256-sha256
        set dpd on-idle
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set ipv4-split-include "LAN_10.10.1.0/24"
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

第二阶段配置：
config vpn ipsec phase2-interface
    edit "IKEv2_PSK"
        set phase1name "IKEv2_PSK"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256
        set pfs disable
        set keepalive enable
    next
end
```

```
地址对象：
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，允许客户端访问内网 VPN 资源。



客户端地址对象：

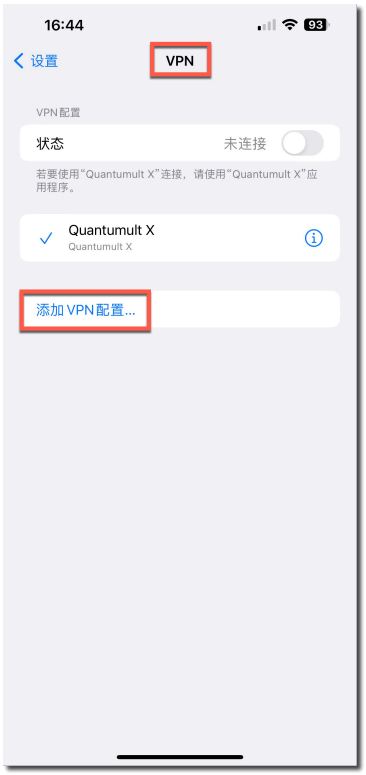
```
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end
```

防火墙策略：

```
config firewall policy
    edit 1
        set name "IKEv2_PSK"
        set srcintf "IKEv2_PSK"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

客户端配置

1. 打开设置 → VPN，点击最下方的“添加 VPN 配置…”按钮。



2. 类型选择 IKEv2，填写 VPN 名称、服务器 IP，远程 ID 随意填写，用户鉴定选择“无”，关闭“使用证书”，填写预共享密钥，点击完成按钮。



3. 点击创建的 VPN 连接，确保该连接被选中（连接前有一个✅）。

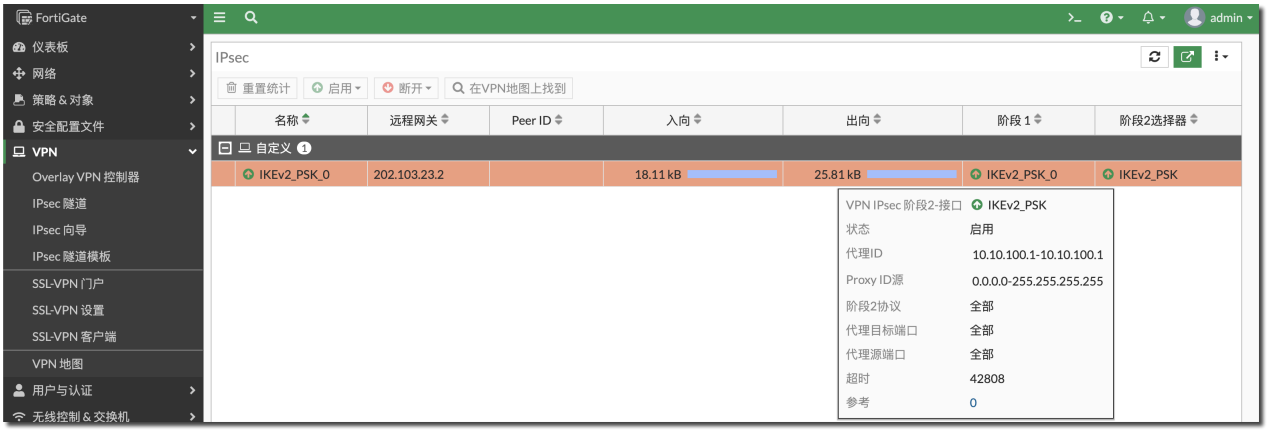


结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。



2. 在 FortiGate 侧查看连接状态。



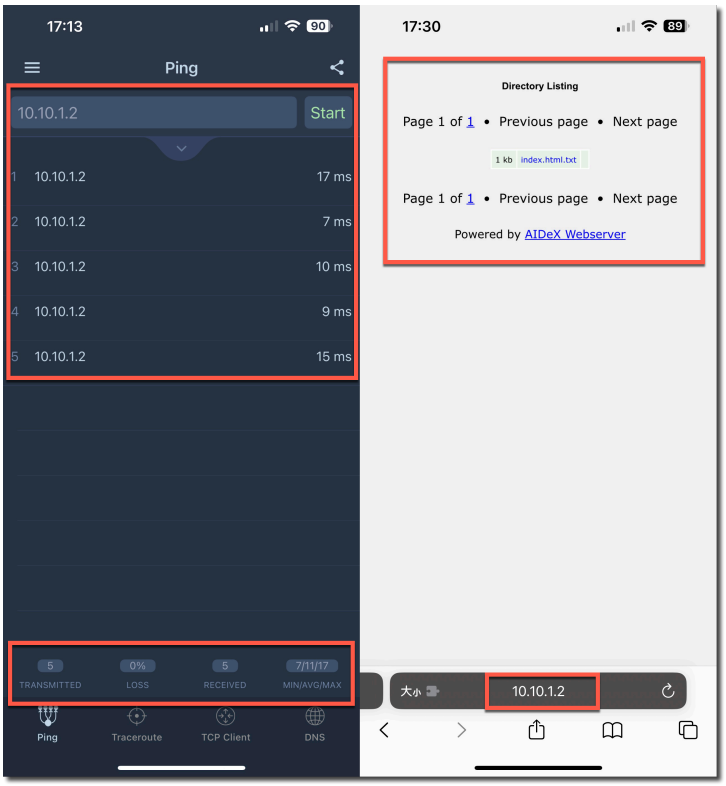

```
diagnose vpn ike gateway list

vd: root/0
name: IKEv2_PSK_0
version: 2
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:64916
tun_id: 10.10.100.1/::10.0.0.9
remote_location: 0.0.0.0
network-id: 0
created: 421s ago
assigned IPv4 address: 10.10.100.1/255.255.255.255
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 450/450/450 ms
IPsec SA: created 1/1  established 1/1  time 140/140/140 ms
  id/spi: 10 5198d872487677ea/31cdf17287e5560c
  direction: responder
  status: established 421-420s ago = 450ms
  proposal: aes256-sha256
  child: no
  SK_ei: dc94f8538f947677-a09ddef1a1bec336-c65701c8f563da84-1812eb919f080c9c
  SK_er: f845db38cae914bf-f45903f0d16af51b-756ce4808bcfedde-f28ff9ec28b3dfe0
  SK_ai: d733f1548b7a4e66-0b7bc5e25539c612-7d9b58309bf6b7ac-539bbdbf51610ccd
  SK_ar: b4b85aecdc194e94-3b04954ddb321f23-fec94182c35dc0d9-ee6c3f8edaab46f4
  PPK: no
  message-id sent/recv: 0/3
  lifetime/rekey: 86400/85709
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=IKEv2_PSK_0 ver=2 serial=f 202.103.12.2:4500->202.103.23.2:64916 tun_id=10.10.100.1 tun_id6=::10.0.0.9 dst_mtu=1500
dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=IKEv2_PSK index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=347 txp=338 rxb=23044 txb=31852
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=64916
proxyid=IKEv2_PSK proto=0 sa=1 ref=24 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20483 type=00 soft=0 mtu=1422 expire=42739/0B replaywin=2048
       seqno=153 esn=0 replaywin_lastseq=0000015b qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43185/43200
  dec: spi=4d836866 esp=aes key=16 8ae2318c0e96f4a41b454135917178a9
       ah=sha1 key=20 87cc5f5d5de49187019ea267a812df1873b9cf6f
  enc: spi=0c26fd39 esp=aes key=16 08bfbe6a97eb166634337de47be3925f
       ah=sha1 key=20 d670dba9af5500ae408f62052c5c192c1b45feff
  dec:pkts/bytes=694/46088, enc:pkts/bytes=676/88844
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=7 dec_npuid=0 enc_npuid=0
------------------------------------------------------
```

```
name=IKEv2_PSK ver=2 serial=e 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.7 tun_id6=::10.0.0.7 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=43032051 olast=43032051 ad=/0
stat: rxp=347 txp=338 rxb=23044 txb=31852
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=25(IKEv2_PSK)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 iOS/macOS 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

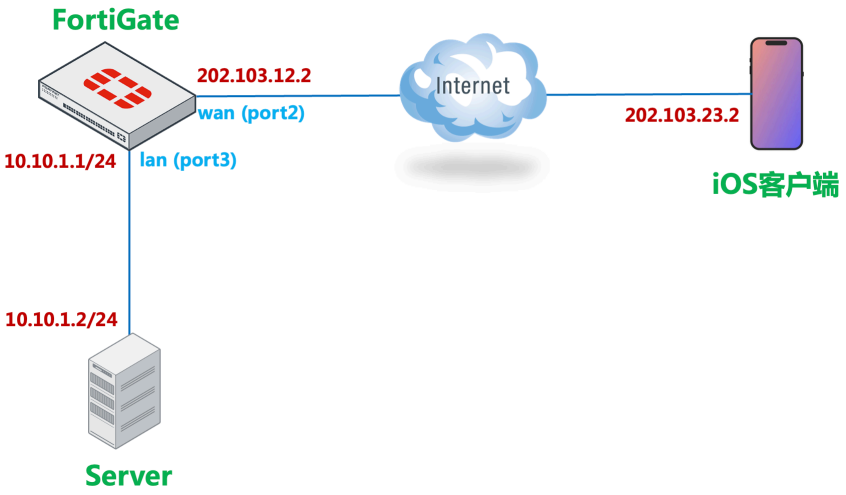
iOS 原生客户端接入 IPSec（IKEv2 用户名/密码 + 预共享密钥认证）

> **重要**
>
> 本文基于 iOS 16.3 编写。

网络需求

通过 iOS 原生客户端（IKEv2 用户名/密码 + 预共享密钥认证）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

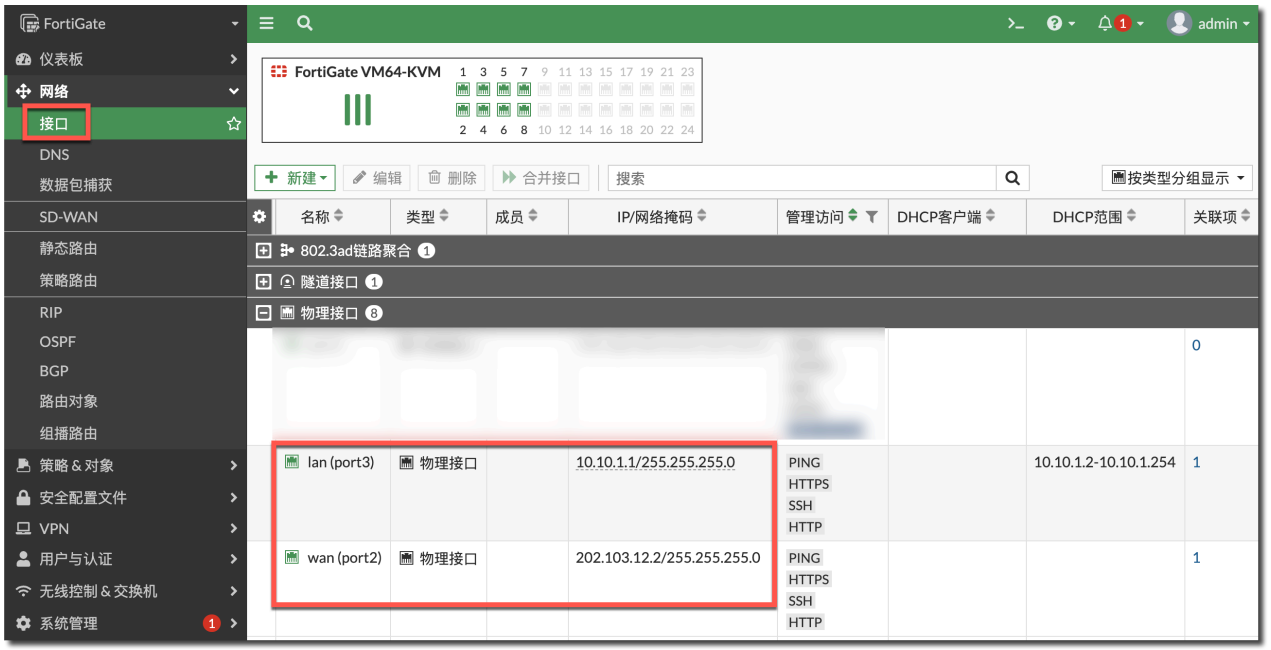
网络拓扑



配置步骤

FortiGate 配置

1. 配置 FortiGate 的接口 IP。



2. 创建本地 VPN 用户和用户组，参考用户与认证 → 本地用户和用户组 ↗章节（如需使用远程认证，请参考用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗或用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。




```
config user local
    edit "vpn_user"
        set type password
        set passwd xxxxxxxx
    next
end
```



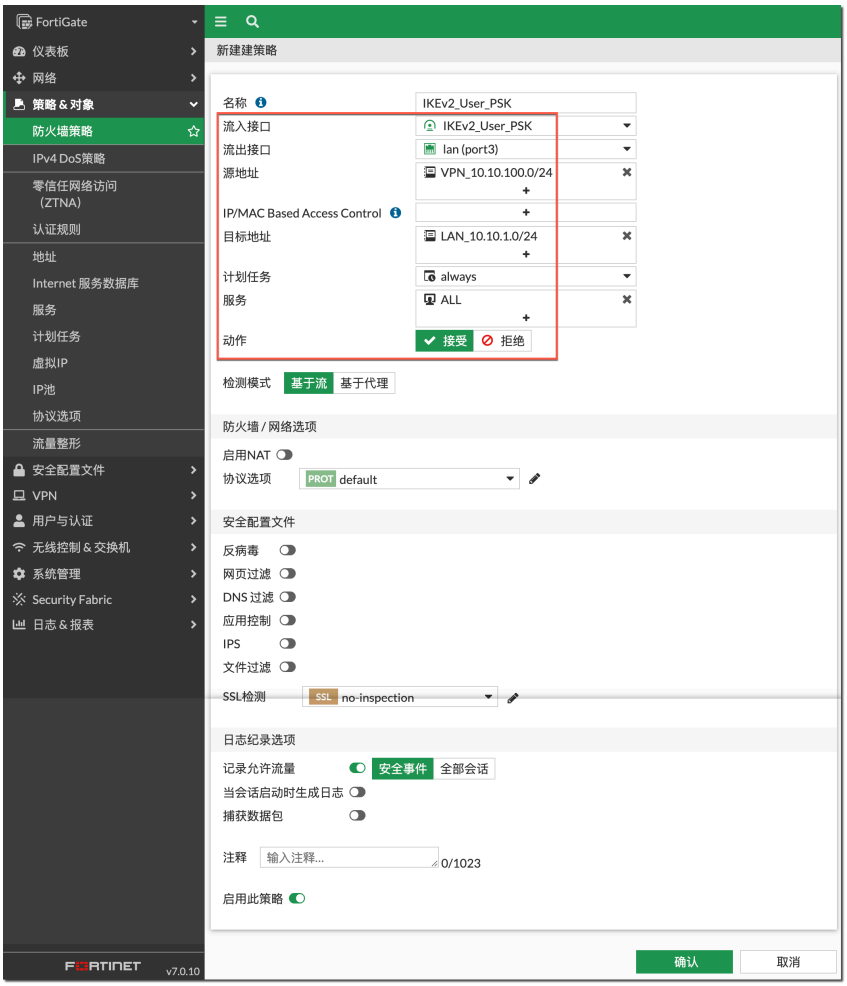

```
config user group
    edit "VPN"
        set member "vpn_user"
    next
end
```

3. 该种方式的 IPSec 需要使用 CLI 方式配置，分别配置 IPSec 一阶段和二阶段。

```
IPSec一阶段配置:
config vpn ipsec phase1-interface
    edit "IKEv2_User_PSK"
        set type dynamic
        set interface "port2"
        set ike-version 2    //IKE协议2
        set peertype any
        set net-device disable
        set mode-cfg enable    //开启为客户端分配IP
        set ipv4-dns-server1 10.10.1.2  //为客户端分配的DNS地址
        set proposal aes256-sha256
        set negotiate-timeout 300
        set dpd on-idle
        set dhgrp 14 5 2
        set eap enable    //需要开启EAP做用户密码认证
        set eap-identity send-request    //开启EAP主动发送认证请求
        set authusrgrp "VPN"    //选择认证的用户组
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100    //客户端分配地址范围
        set ipv4-netmask 255.255.255.0
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

IPSec二阶段配置:
config vpn ipsec phase2-interface
    edit "IKEv2_User_PSK"
```

```
        set phase1name "IKEv2_User_PSK"
        set proposal aes256-sha256
        set keepalive enable
    next
end
```

4. 配置防火墙策略，允许客户端访问内网 VPN 资源。



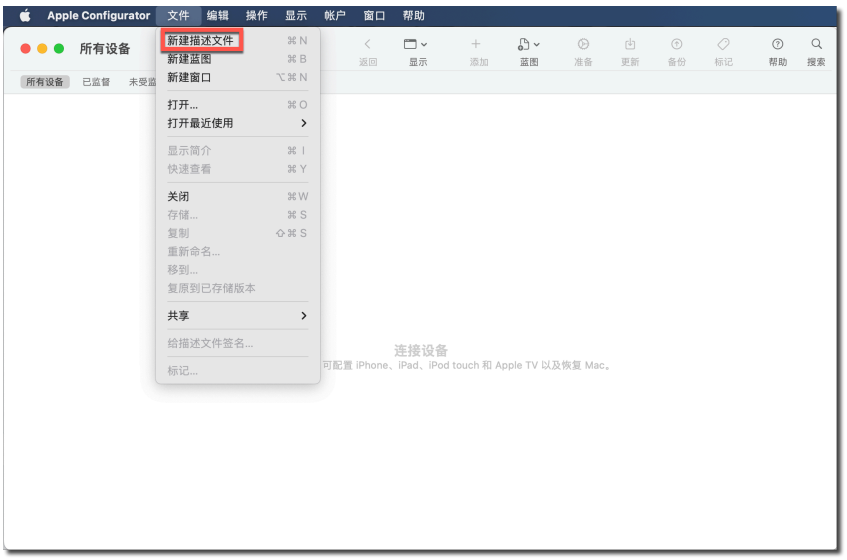

```
VPN资源地址对象:
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end

客户端地址对象:
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end

防火墙策略:
config firewall policy
    edit 1
        set name "IKEv2_User_PSK"
        set srcintf "IKEv2_User_PSK"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
```

```
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

制作和安装描述文件

> **重要**
>
> 想要使用 IKEv2 用户名/密码 + 预共享密钥方式的 VPN 认证，需要使用 Apple Configurator 工具为 iOS/macOS 制作描述文件，该工具可以在 macOS 下的 App Store 中下载。

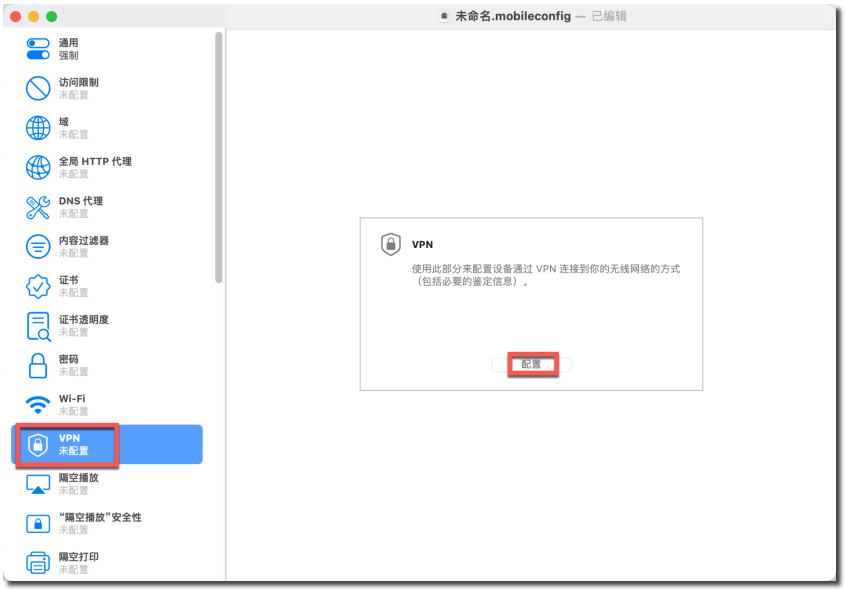
1. 打开 Apple Configurator，点击菜单栏的“文件 → 新建描述文件”（或使用快捷键 Command + N）。



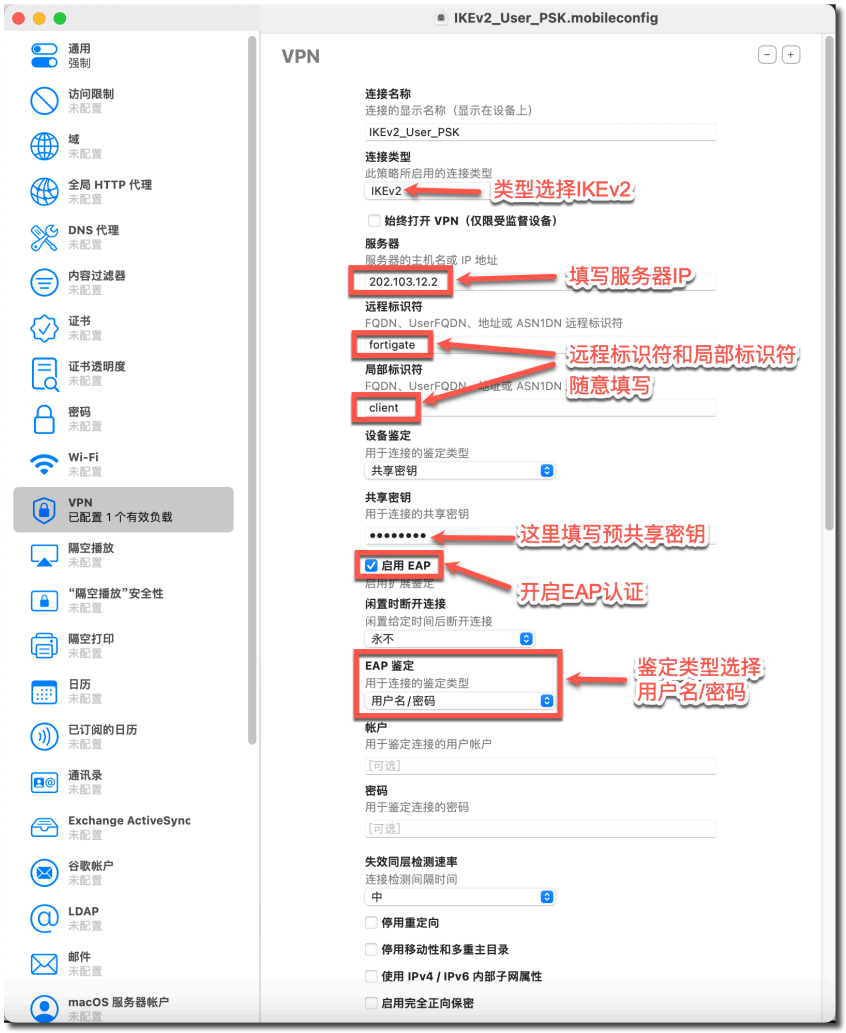
2. 在弹出的配置文件编辑器中，配置描述文件的名称。



3. 选择 VPN 选项卡，点击配置按钮。



4. 配置描述文件，选择连接类型为 IKEv2，填写服务器 IP 地址，远程标识符和局部标识符随意填写，填写预共享密钥，开启 EAP，选择 EAP 鉴定为用户名/密码模式，其他配置保持默认即可（或根据实际需求更改）。


IKEv2_User_PSK.mobileconfig
通用
强制
访问限制
未配置
域
未配置
全局 HTTP 代理
未配置
DNS 代理
未配置
内容过滤器
未配置
证书
未配置
证书透明度
未配置
密码
未配置
Wi-Fi
未配置
VPN
已配置 1 个有效负载
隔空播放
未配置
“隔空播放”安全性
未配置
隔空打印
未配置
日历
未配置
已订阅的日历
未配置
通讯录
未配置
Exchange ActiveSync
未配置
谷歌帐户
未配置
LDAP
未配置
邮件
未配置
macOS 服务器帐户
未配置
VPN
连接名称
连接的显示名称（显示在设备上）
IKEv2_User_PSK
连接类型
此策略所启用的连接类型
IKEv2
类型选择IKEv2
始终打开 VPN（仅限受监督设备）
服务器
服务器的主机名或 IP 地址
202.103.12.2
填写服务器IP
远程标识符
FQDN、UserFQDN、地址或 ASN1DN 远程标识符
fortigate
远程标识符和局部标识符
随意填写
局部标识符
FQDN、UserFQDN、地址或 ASN1DN
client
设备鉴定
用于连接的鉴定类型
共享密钥
共享密钥
用于连接的共享密钥
••••••••
这里填写预共享密钥
启用 EAP
开启EAP认证
闲置时断开连接
闲置给定时间后断开连接
永不
EAP 鉴定
用于连接的鉴定类型
用户名/密码
鉴定类型选择
用户名/密码
帐户
用于鉴定连接的用户帐户
[可选]
密码
用于鉴定连接的密码
[可选]
失效同层检测速率
连接检测间隔时间
中
停用重定向
停用移动性和多重主目录
使用 IPv4 / IPv6 内部子网属性
启用完全正向保密



5. 选择菜单栏的“文件 → 存储”按钮（或使用快捷键 Command + S），保存该描述文件到本地。



6. 通过邮件等方式，将描述文件发送至 iPhone。

7. iPhone 通过邮件 App 收取该描述文件后，点击该描述文件，会弹出提示进入设置中查看描述文件。



8. 进入设置，点击“已下载描述文件”。



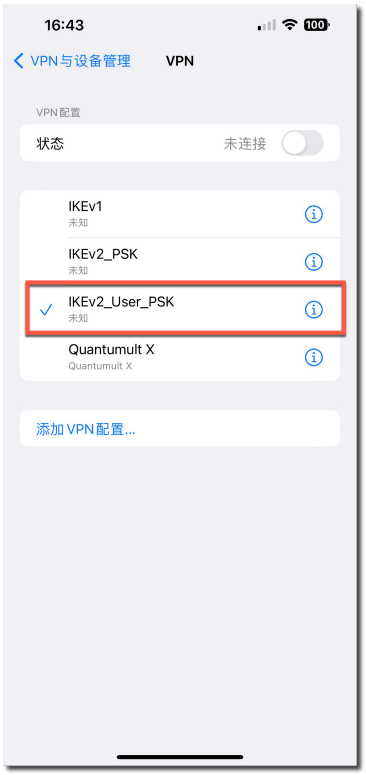
9. 选择制作的描述文件并安装（需要输入手机密码）。



.0. 根据提示输入用户名和密码，点击完成按钮安装描述文件。

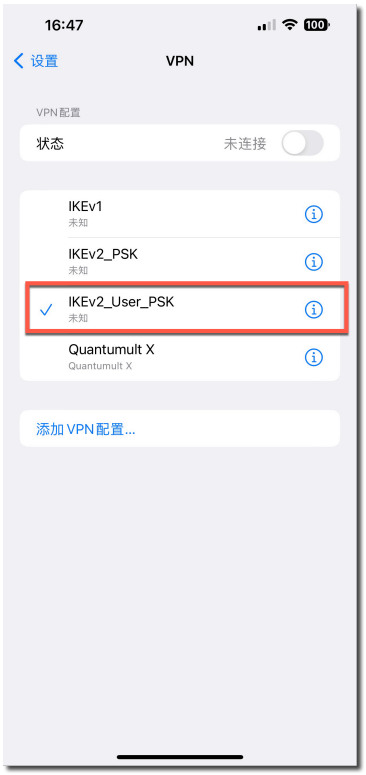


.1. VPN 列表中自动生成 VPN 配置。



客户端配置

1. 打开设置 → VPN，点击创建的 VPN 连接，确保该连接被选中（连接前有一个✅）。



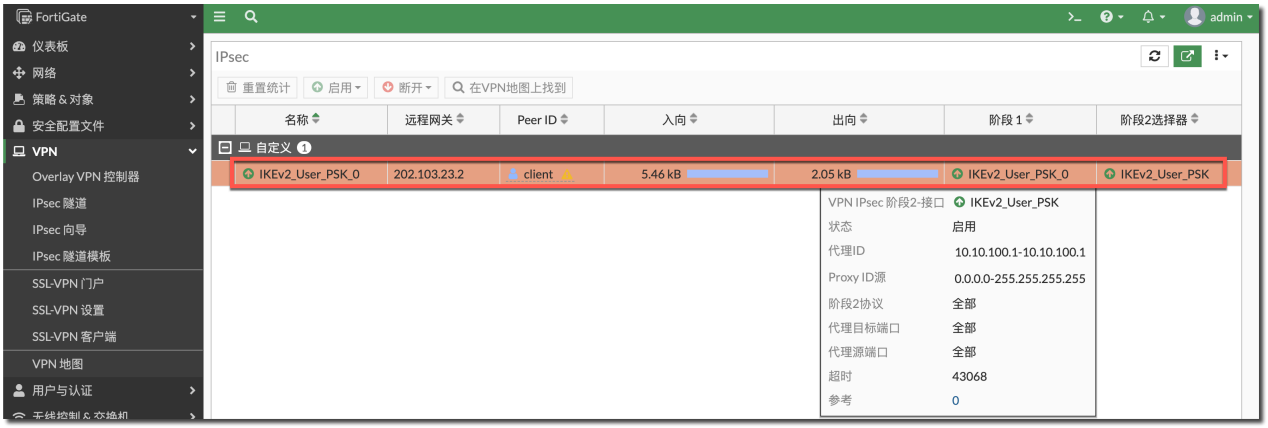
2. 可根据实际需要修改 VPN 连接中的用户名或密码。

结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。



2. 在 FortiGate 侧查看连接状态。



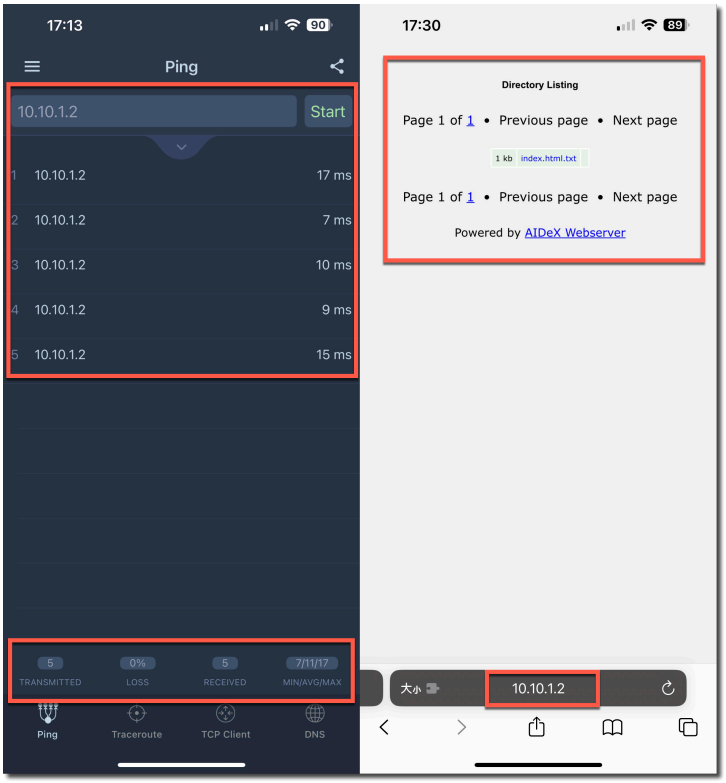

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: IKEv2_User_PSK_0
version: 2
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:64916
tun_id: 10.10.100.1/::10.0.0.17
remote_location: 0.0.0.0
network-id: 0
created: 194s ago
eap-user: vpn_user
2FA: no
peer-id: client
peer-id-auth: no
assigned IPv4 address: 10.10.100.1/255.255.255.0
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 90/90/90 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms
  id/spi: 13 d5ae11369c1b3d95/68eb5a2b96838488
  direction: responder
  status: established 194-194s ago = 90ms
```

```
  proposal: aes256-sha256
  child: no
  SK_ei: 5f2946151d12c26b-0ca7ea2a894ba835-a6a5c9a24d44caa8-e11178ab9a2cd6f5
  SK_er: fe330d4fbd3f7753-77748964618e4a94-4bf4f7baeedcca70-ba645a3fc31f2265
  SK_ai: 488d6b7d9df75b01-d2c8bdc7049e552a-0c17d452a8b4ebe8-631caa5abacaa0c6
  SK_ar: 782bdd0189841d1c-fc202aa4598d009e-056e04417ae4ad9b-da9ede3c92250546
  PPK: no
  message-id sent/recv: 0/6
  lifetime/rekey: 86400/85935
  DPD sent/recv: 00000000/00000000
  peer-id: client

FortiGate # diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=IKEv2_User_PSK ver=2 serial=14 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.9 tun_id6=::10.0.0.9 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=43039746 olast=43039746 ad=/0
stat: rxp=1723 txp=337 rxb=127170 txb=41184
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=IKEv2_User_PSK_0 ver=2 serial=1a 202.103.12.2:4500→202.103.23.2:64916 tun_id=10.10.100.1 tun_id6=::10.0.0.17
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=IKEv2_User_PSK index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=12 olast=12 ad=/0
stat: rxp=113 txp=38 rxb=7546 txb=3581
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=64916
proxyid=IKEv2_User_PSK proto=0 sa=1 ref=10 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20683 type=00 soft=0 mtu=1422 expire=42991/0B replaywin=2048
       seqno=27 esn=0 replaywin_lastseq=00000071 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43190/43200
  dec: spi=7b8ba378 esp=aes key=32 e2651b177effa8747d930eacb965a527f5758f5f0a6503225f2d21eac7d549fe
       ah=sha256 key=32 1f86d1b2efb0dbe0511f2c0b79257ab6b76c3f42aa9ddbafde6f8f4c7b13eb27
  enc: spi=0846436d esp=aes key=32 87d133ec91eccd6c9b10671111bd07242f9560ba62c4735c5a8dffc52ededf59
       ah=sha256 key=32 792ce83e7dce6fdfdb8df9edf286a206023b236a263fa9ab09cdd3b819064386
  dec:pkts/bytes=226/15092, enc:pkts/bytes=76/10133
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=f dec_npuid=0 enc_npuid=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=27(IKEv2_User_PSK)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 iOS/macOS 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

Android 系统

- Android12 以下系统（IKEv1）
- Android12 或以上系统（IKEv2）

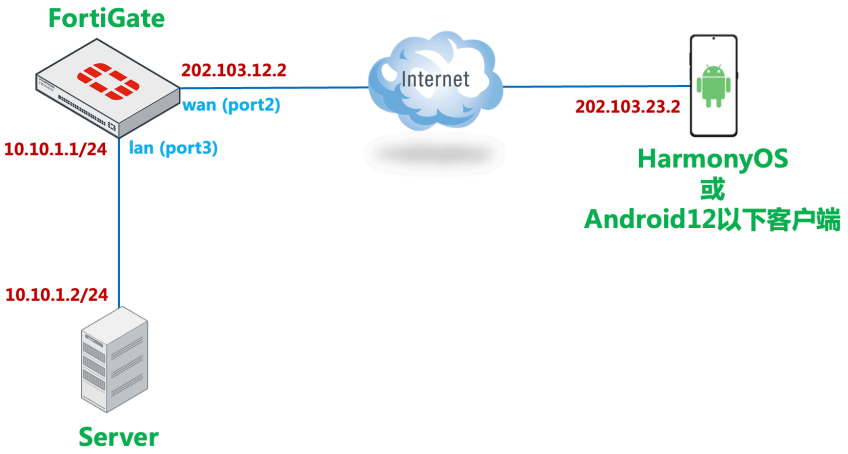
Android12 以下系统（IKEv1）

网络需求

通过 Android12 以下系统客户端（IKEv1）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

> **重要**
>
> 通过 IKEv1 用户名/密码 + 预共享密钥的方式认证（不涉及 L2TP）。

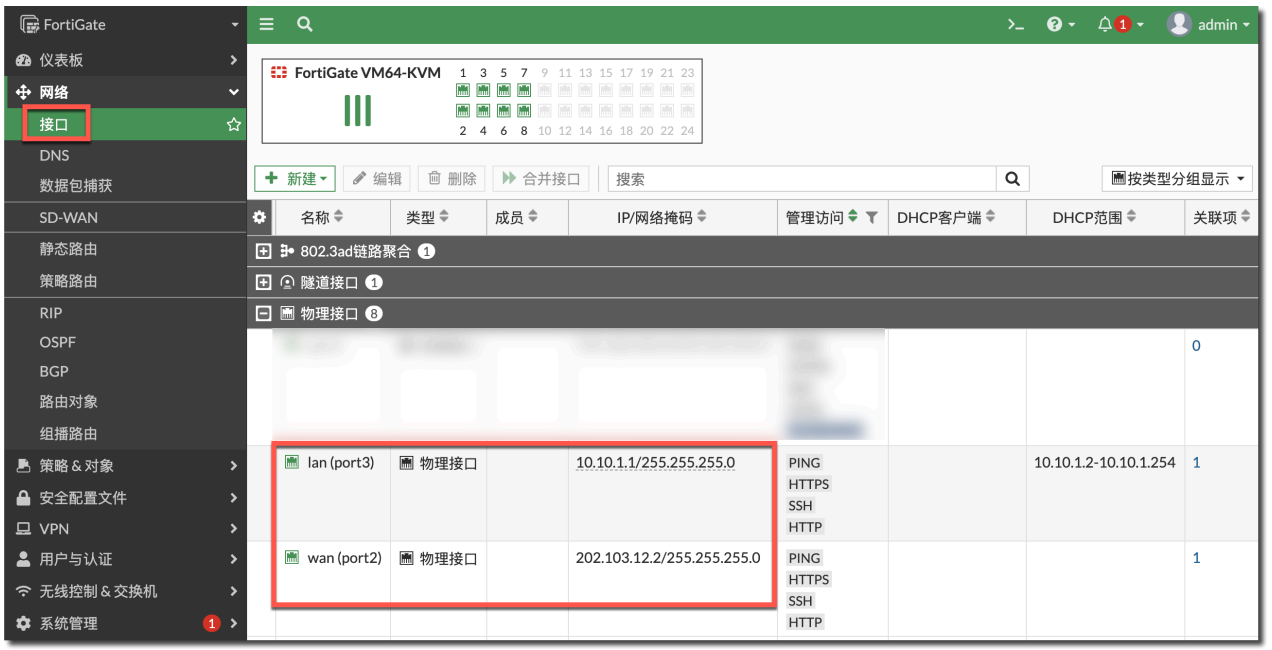
网络拓扑



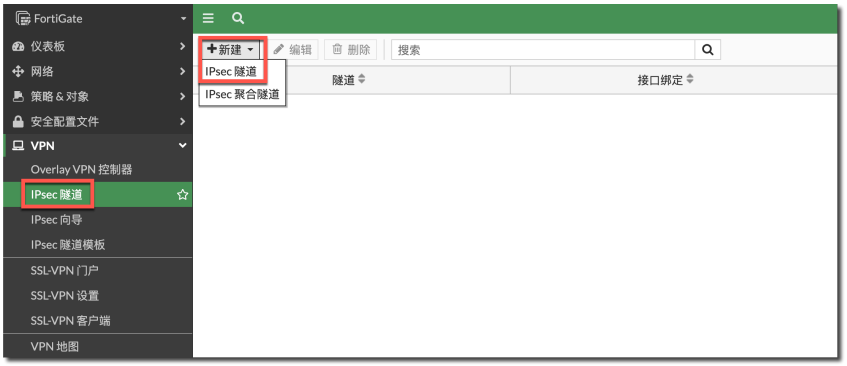
配置步骤

FortiGate 配置

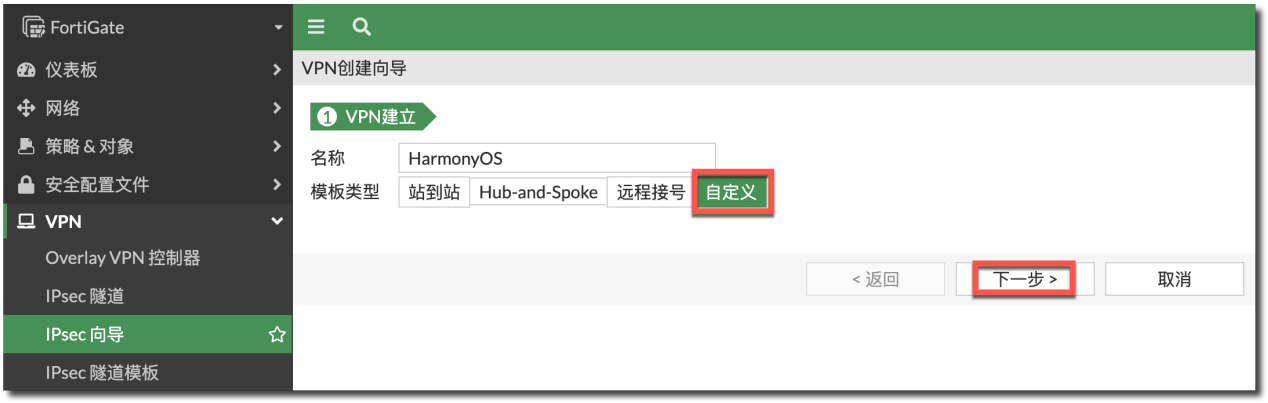
1. 配置 FortiGate 的接口 IP。



2. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



3. 填写 VPN 名称，模版选择自定义，点击下一步。



4. 按照下图修改配置，其他配置保持默认即可，点击确认下发配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
新VPN隧道
名称 HarmonyOS
注释
选择为拨号用户模式
接口为wan口
网络
IP版本 IPv4 IPv6
对端网关 拨号用户
接口 wan (port2)
本地网关
开启模式配置
模式配置
DNS
分配 IP从 范围
为客户端分配的IP地址范围和DNS地址
IPv4 模式配置
客户端地址范围 10.10.1.1-10.10.1.100
子网掩码 255.255.255.255
DNS 服务器 10.10.1.2
启动IPv4分割隧道
IPv6 模式配置
客户端地址范围 ::-::
前缀长度 128
DNS 服务器 ::
启用IPv6隧道分割
NAT穿越 启用 禁用 强制的
对等体状态探测 禁用 空闲 按需
DPD重试计数 3
DPD重试间隔 60 s
前向纠错 Egress Ingress
高级
认证
方法 预共享密钥
预共享密钥
IKE
版本 1 2
模式 野蛮 主模式(ID保护)
填写预共享密钥
访问类型选择“任意对等体ID”
对等体选项
访问类型 任意对等体ID
阶段 1 提案 添加
一阶段算法
加密 AES256 认证 SHA256
加密 AES128 认证 SHA1
Diffie-Hellman 组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
密钥生存时间 (秒) 86400
本地ID
XAUTH
开启XAUTH
类型 Auto 服务器
用户组 从策略继承 选择
类型选择“Auto”服务器
VPN
用户组选择VPN用户组
阶段 2 选择器
名称 本地地址 对端地址
HarmonyOS 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
新建阶段 2
名称 HarmonyOS
注释
本地地址 子网 0.0.0.0/0.0.0.0
对端地址 子网 0.0.0.0/0.0.0.0
高级
二阶段算法
阶段 2 提案 添加
加密 AES256 认证 SHA256
加密 AES128 认证 SHA1
启用重播检测
启用完全前向保密 (PFS)
关闭PFS
本地端口 全部
对端端口 全部
协议 全部
自动密钥保持存活
开启“自动密钥保持存活”



5. 对应的 CLI 如下：

```
第一阶段配置：
config vpn ipsec phase1-interface
    edit "HarmonyOS"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 10.10.1.2
        set proposal aes256-sha256 aes128-sha1
        set dpd on-idle
        set dhgrp 14 5 2
        set xauthtype auto
        set authusrgrp "VPN"
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

第二阶段配置：
config vpn ipsec phase2-interface
    edit "HarmonyOS"
        set phase1name "HarmonyOS"
        set proposal aes256-sha256 aes128-sha1
        set pfs disable
        set keepalive enable
    next
end
```

```
地址对象：
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，允许客户端访问内网 VPN 资源。



客户端地址对象：

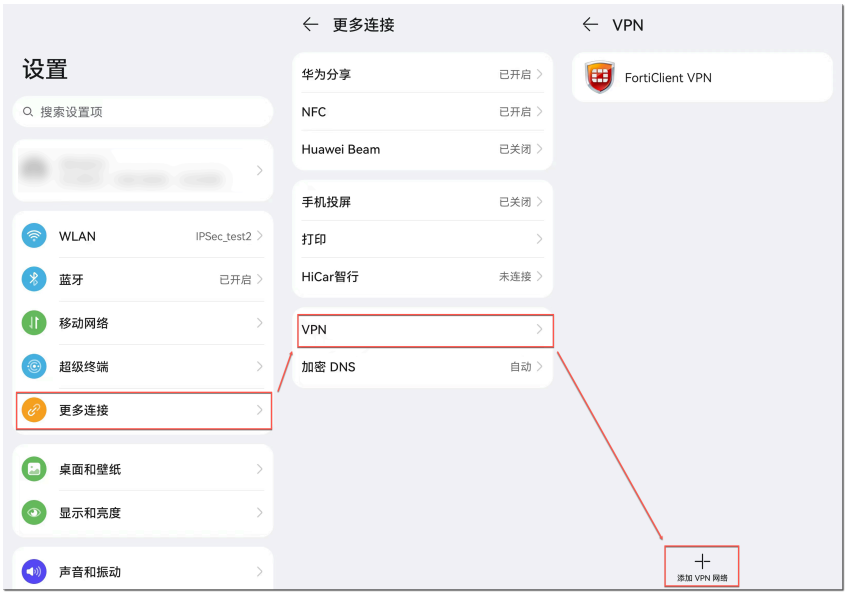
```
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end
```

防火墙策略：

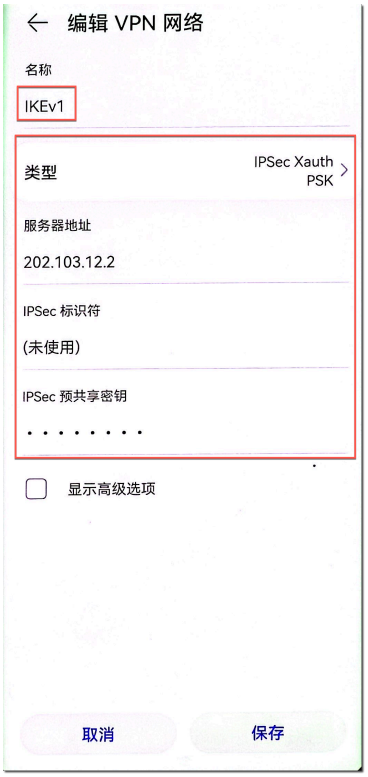
```
config firewall policy
    edit 1
        set name "HarmonyOS"
        set srcintf "HarmonyOS"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

客户端配置

1. 打开设置 → 更多连接 → VPN，点击下方的“添加 VPN 网络”。



2. 类型选择 IPSec Xauth PSK，填写名称、服务器地址、IPSec 预共享密钥，点击保存按钮保存连接（由于华为手机在 VPN 连接配置界面无法截图，所以下图为拍照）。



结果验证

1. 点击创建的 VPN 连接，输入用户名与密码，按需求勾选“保存账户信息”（由于华为手机在 VPN 连接配置界面无法截图，所以下图为拍照），点击连接按钮。连接成功后，点击查看连接状态，可以看到连接的具体信息。



2. 在 FortiGate 侧查看连接状态。



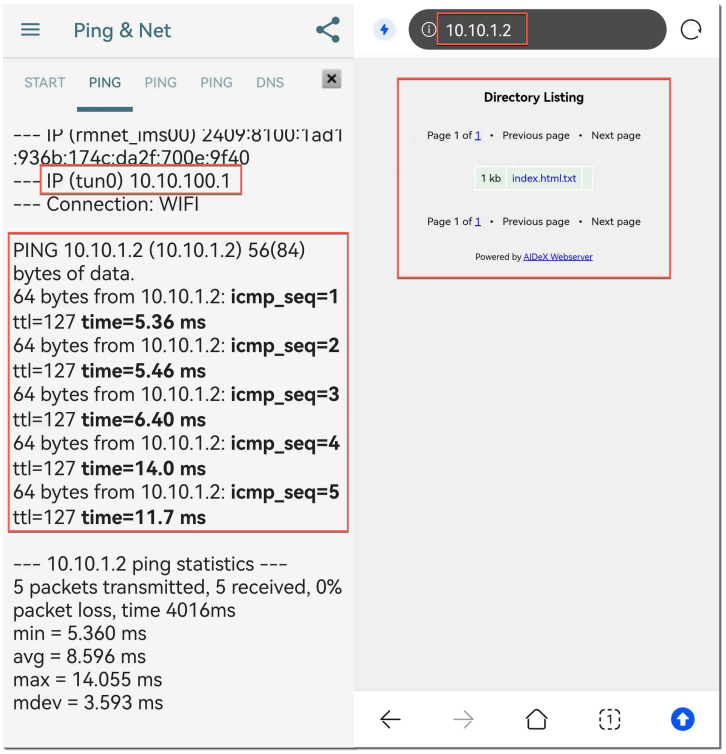

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: HarmonyOS_0
version: 1
interface: port2 4
addr: 202.103.12.2:4500 -> 202.103.23.2:64916
tun_id: 10.10.100.1/::10.0.0.19
remote_location: 0.0.0.0
network-id: 0
created: 26s ago
xauth-user: vpn_user
2FA: no
assigned IPv4 address: 10.10.100.1/255.255.255.255
nat: peer
IKE SA: created 1/1  established 1/1  time 240/240/240 ms
IPsec SA: created 1/1  established 1/1  time 440/440/440 ms
  id/spi: 73 39a070180abf8f20/7c90c7a59d942c95
  direction: responder
  status: established 26-26s ago = 240ms
  proposal: aes256-sha256
  key: 888ae90f86ef043e-a160f4d6b782c66d-c6d006f30ed9f5d7-923c853d34fa1e06
  lifetime/rekey: 28800/28503
```

```
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=HarmonyOS_0 ver=1 serial=17 202.103.12.2:4500->202.103.23.2:64916 tun_id=10.10.100.1 tun_id6=::10.0.0.19
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=HarmonyOS index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=7 olast=7 ad=/0
stat: rxp=5 txp=3 rxb=331 txb=289
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=32 interval=10 remote_port=64916
proxyid=HarmonyOS proto=0 sa=1 ref=5 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20083 type=00 soft=0 mtu=1422 expire=28757/0B replaywin=2048
       seqno=4 esn=0 replaywin_lastseq=00000005 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=28786/28800
  dec: spi=a06d669d esp=aes key=32 bb01635fcbcfec11f8970f8b58a23c6cadc3c12e6b506a799b8563c73501e5db
       ah=sha256 key=32 d8eb7b5dea988ea7b2d0509a6160b7f285827c4f56d6293aa77230759034518d
  enc: spi=02c359e0 esp=aes key=32 d8e3495f33ee0a4b7f851a4eaa1756f6ab16bddb6ba9d9c07b34fcc4c63d6e55
       ah=sha256 key=32 39e4c2b7c0b060c6d27a605c55821e27276b780699081975f1362a13580bd012
  dec:pkts/bytes=10/662, enc:pkts/bytes=6/813
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=11 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=HarmonyOS ver=1 serial=13 202.103.12.2:0->0.0.0.0:0 tun_id=10.0.0.5 tun_id6=::10.0.0.5 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=42980186 olast=42980186 ad=/0
stat: rxp=7 txp=5 rxb=459 txb=473
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=23(HarmonyOS)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 Android 使用 IKEv1，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

Android12 或以上系统（IKEv2）

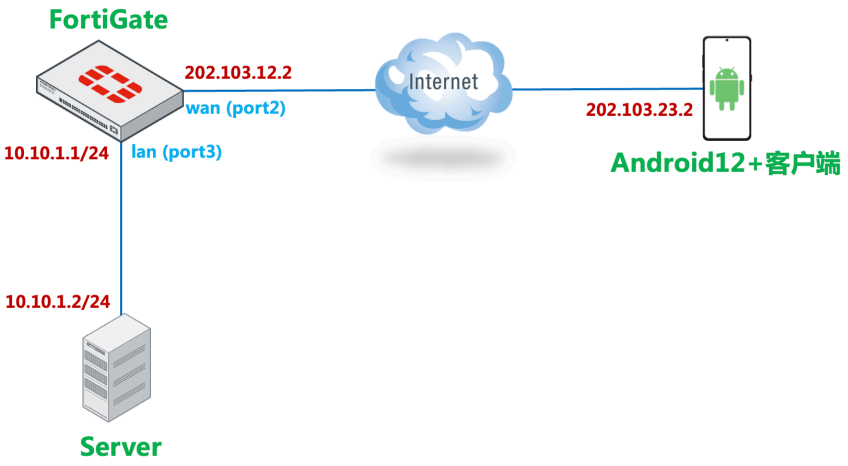
> **重要**
>
> 本文基于 Sumsung One UI 5.1 (Android13) 系统编写。

网络需求

通过 Android12 或以上系统客户端（IKEv2）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

> **重要**
>
> 通过 IKEv2 预共享密钥的方式认证（无用户密码认证）。

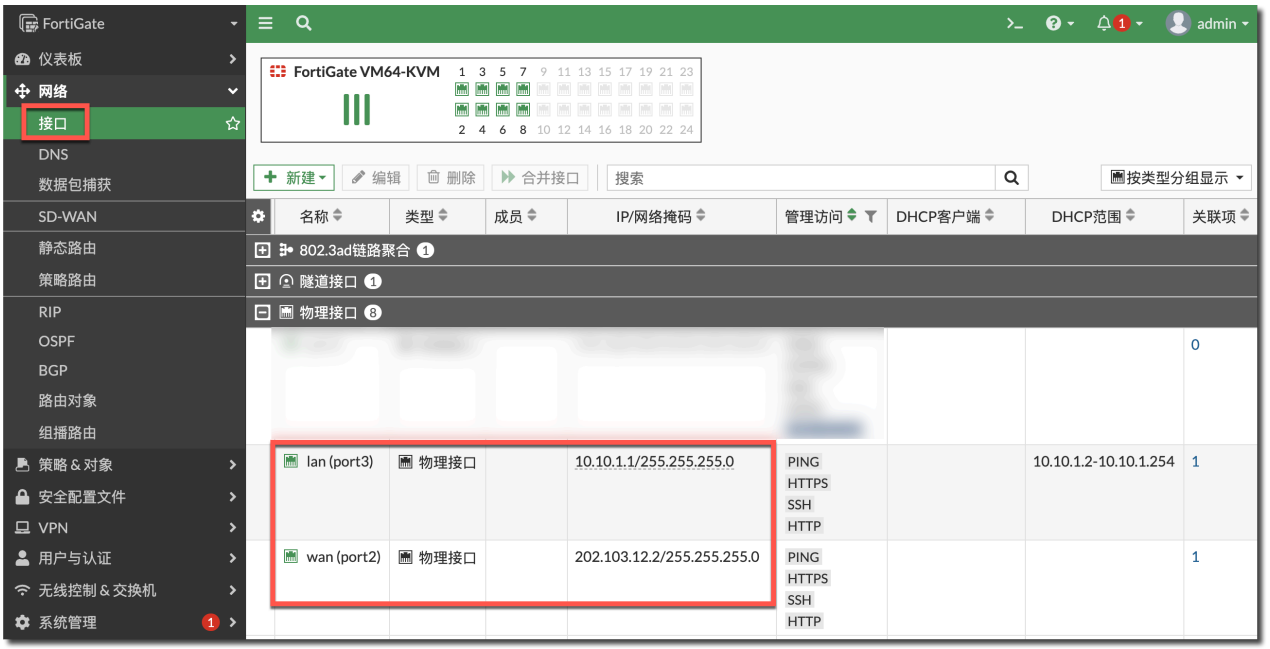
网络拓扑



配置步骤

FortiGate 配置

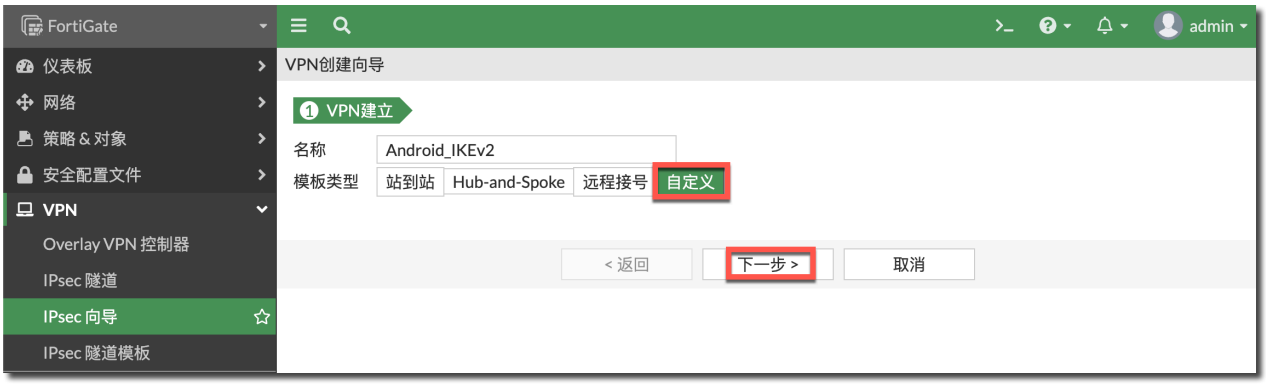
1. 配置 FortiGate 的接口 IP。



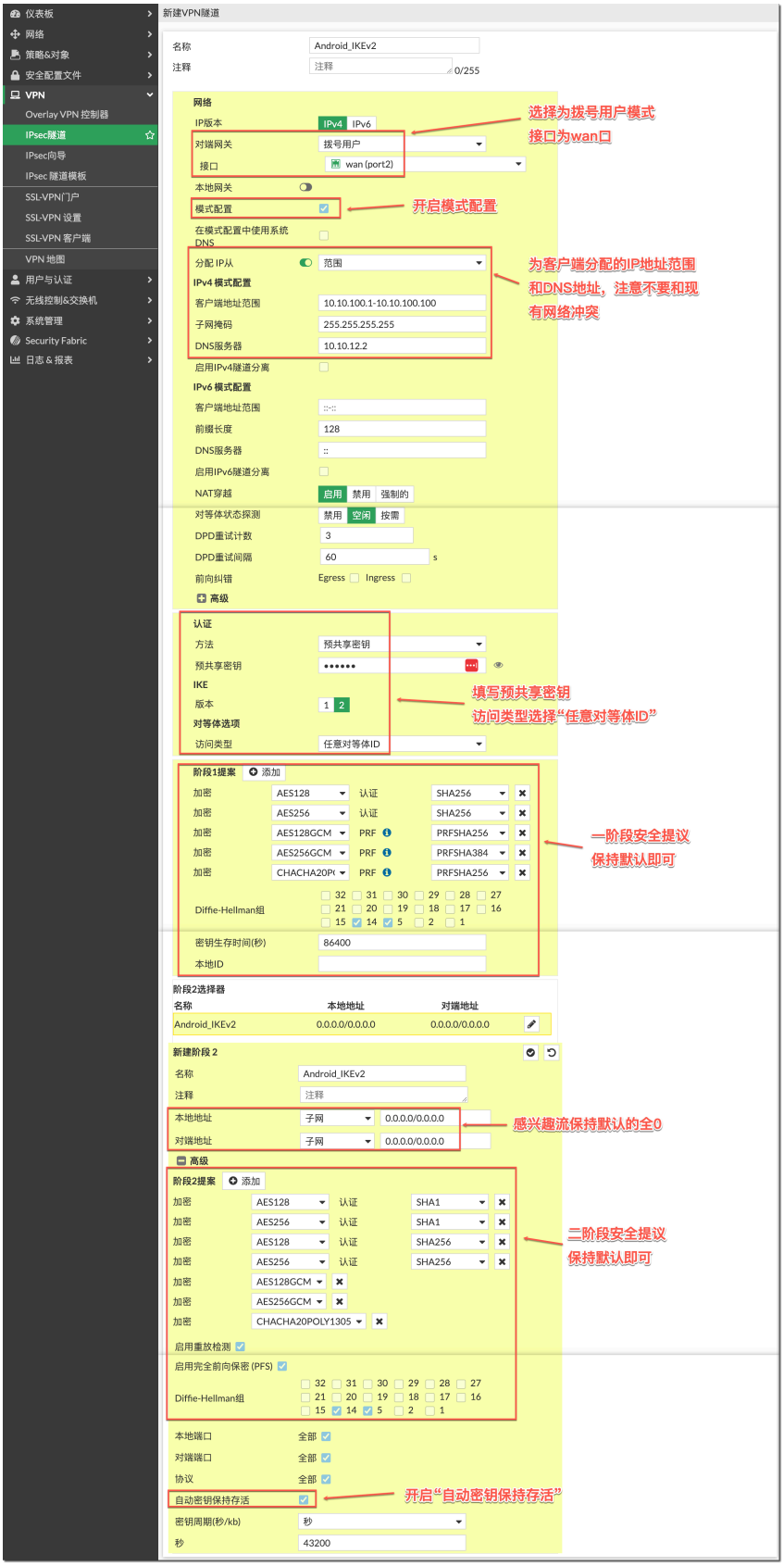
2. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



3. 填写 VPN 名称，模版选择自定义，点击下一步。



4. 按照下图修改配置，其他配置保持默认即可，点击确认下发配置。

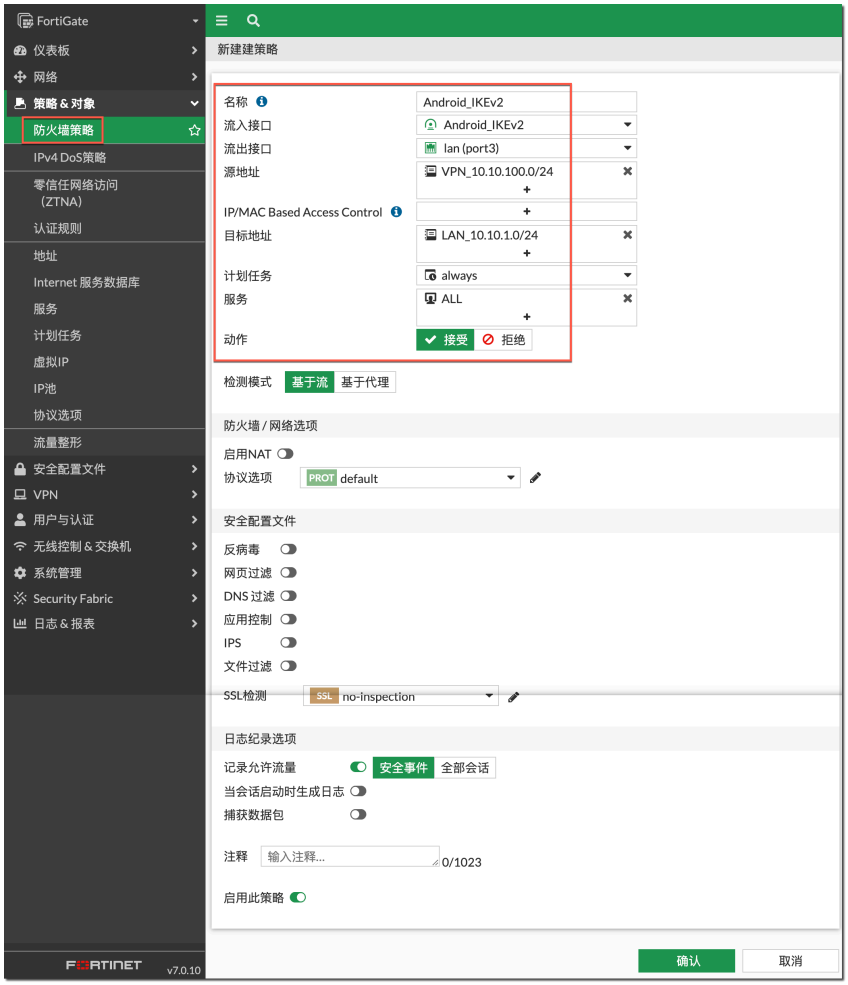

仪表板
网络
策略&对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec隧道
IPsec向导
IPsec 隧道模板
SSL-VPN门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制&交换机
系统管理
Security Fabric
日志 & 报表
新建VPN隧道
名称 Android_IKEv2
注释 0/255
网络
IP版本 IPv4 IPv6
对端网关 拨号用户
接口 wan (port2)
选择为拨号用户模式
接口为wan口
本地网关
模式配置
开启模式配置
在模式配置中使用系统DNS
分配 IP从 范围
IPv4 模式配置
客户端地址范围 10.10.100.1-10.10.100.100
子网掩码 255.255.255.255
DNS服务器 10.10.12.2
为客户端分配的IP地址范围
和DNS地址，注意不要和现
有网络冲突
启用IPv4隧道分离
IPv6 模式配置
客户端地址范围 ::-::
前缀长度 128
DNS服务器 ::
启用IPv6隧道分离
NAT穿越 启用 禁用 强制的
对等体状态探测 禁用 空闲 按需
DPD重试计数 3
DPD重试间隔 60 s
前向纠错 Egress Ingress
高级
认证
方法 预共享密钥
预共享密钥 ••••••
IKE
版本 1 2
对等体选项
访问类型 任意对等体ID
填写预共享密钥
访问类型选择“任意对等体ID”
阶段1提案 添加
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM PRF PRFSHA256
加密 AES256GCM PRF PRFSHA384
加密 CHACHA20P PRF PRFSHA256
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
一阶段安全提议
保持默认即可
密钥生存时间(秒) 86400
本地ID
阶段2选择器
名称 本地地址 对端地址
Android_IKEv2 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
新建阶段 2
名称 Android_IKEv2
注释
本地地址 子网 0.0.0.0/0.0.0.0
对端地址 子网 0.0.0.0/0.0.0.0
感兴趣流保持默认的全0
高级
阶段2提案 添加
加密 AES128 认证 SHA1
加密 AES256 认证 SHA1
加密 AES128 认证 SHA256
加密 AES256 认证 SHA256
加密 AES128GCM
加密 AES256GCM
加密 CHACHA20POLY1305
二阶段安全提议
保持默认即可
启用重放检测
启用完全前向保密 (PFS)
Diffie-Hellman组 32 31 30 29 28 27 21 20 19 18 17 16 15 14 5 2 1
本地端口 全部
对端端口 全部
协议 全部
自动密钥保持存活
开启“自动密钥保持存活”
密钥周期(秒/kb) 秒
秒 43200



5. 对应的 CLI 如下：

```
第一阶段配置：
config vpn ipsec phase1-interface
    edit "Android_IKEv2"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 10.10.12.2
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

第二阶段配置：
config vpn ipsec phase2-interface
    edit "Android_IKEv2"
        set phase1name "Android_IKEv2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set keepalive enable
    next
end
```

6. 配置防火墙策略，允许客户端访问内网 VPN 资源。



内网VPN资源地址对象：

```
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
 next
end
```

客户端地址对象：

```
config firewall address
       edit "VPN_10.10.100.0/24"
           set subnet 10.10.100.0 255.255.255.0
       next
   end
```

防火墙策略：

```
config firewall policy
    edit 1
        set name "Android_IKEv2"
        set srcintf "Android_IKEv2"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

客户端配置

7. 打开设置 → 连接 → 更多连接设置 → VPN（找不到可以使用搜索功能），点击“添加 VPN 配置文件”。



8. 类型选择 IKEv2/IPSec PSK，填写名称、服务器地址、IPSec 预共享密钥，注意必须填写 IPSec 标识符（随意填写即可），点击保存按钮保存连接。

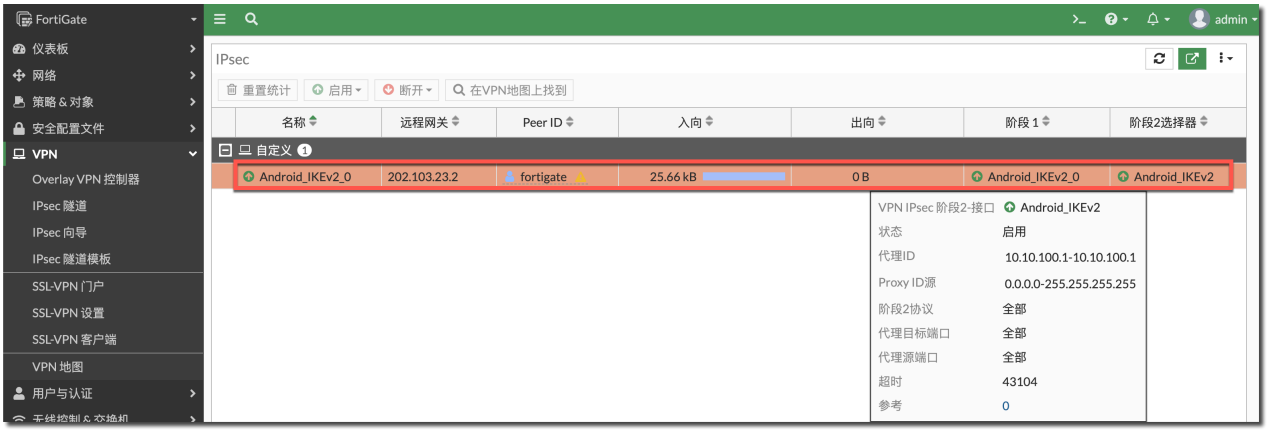


结果验证

1. 点击创建的 VPN 连接，点击连接按钮。连接成功后，点击查看连接状态，可以看到连接的具体信息。



2. 在 FortiGate 侧查看连接状态。



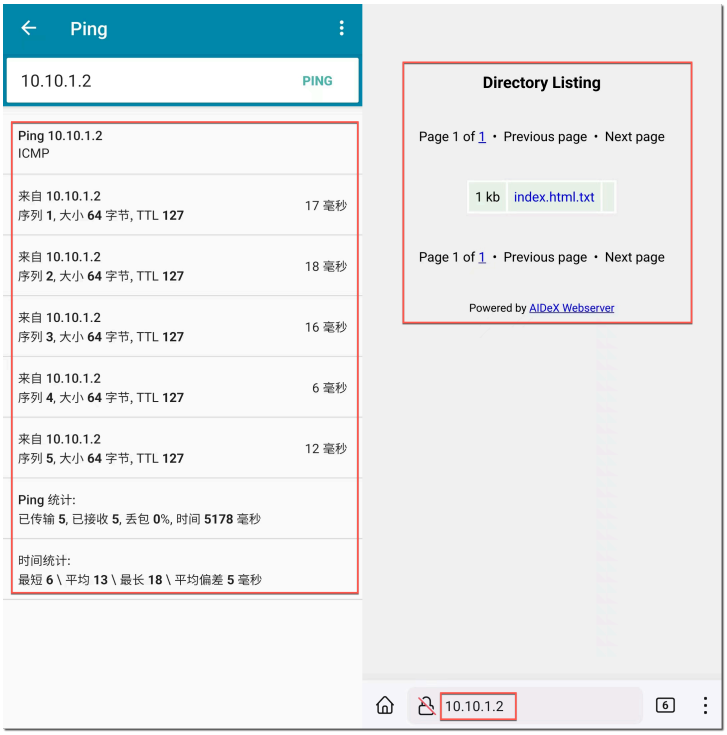

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: Android_IKEv2_0
version: 2
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:35653
tun_id: 10.10.100.1/::10.0.0.29
remote_location: 0.0.0.0
network-id: 0
created: 8s ago
peer-id: fortigate
peer-id-auth: no
assigned IPv4 address: 10.10.100.1/0.0.0.0
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 5 24e28229fbb0aabf/af590cf99666b224
  direction: responder
  status: established 8-8s ago = 20ms
  proposal: aes128-sha256
  child: no
  SK_ei: ebe7fafddd9a3afa-d25d578817491d46
  SK_er: d498cfd72bc78306-5d96c828c054dc49
```

```
  SK_ai: db2fe8a3c377d068-043cb17abe4a50dc-0b6a7406d9136909-d8f071b76a03d411
  SK_ar: 0816650a93fb45f2-620c1f63caa107c3-bdf507a33957e29e-e22ec78cdf616f11
  PPK: no
  message-id sent/recv: 0/2
  lifetime/rekey: 86400/86121
  DPD sent/recv: 00000000/00000000
  peer-id: fortigate

FortiGate # diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=Android_IKEv2_0 ver=2 serial=25 202.103.12.2:4500→202.103.23.2:35653 tun_id=10.10.100.1 tun_id6=::10.0.0.29
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=Android_IKEv2 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=11 ad=/0
stat: rxp=63 txp=0 rxb=3644 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=35653
proxyid=Android_IKEv2 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20683 type=00 soft=0 mtu=1422 expire=43175/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=0000003f qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43187/43200
  dec: spi=02cf6ed7 esp=aes key=16 e9a35403758eddeaa21370e37e36ea8d
       ah=sha256 key=32 54a6e676edae65bb37a0f4a6605b8fcfde743af1f229d29624abbbd4cc8bf8ec
  enc: spi=cbf93f37 esp=aes key=16 4e475164fbf9ac1be9c81497cd8ff98f
       ah=sha256 key=32 682b3b42a8f121a227f70a8952cc92110371f06f32309e5a0c7bb43a48677667
  dec:pkts/bytes=126/7288, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=1b dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=Android_IKEv2 ver=2 serial=22 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.9 tun_id6=::10.0.0.9 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=43054304 olast=43054304 ad=/0
stat: rxp=7209 txp=12 rxb=427332 txb=832
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
 ...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=27(Android_IKEv2)
 ...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 Android 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

macOS

macOS 原生客户端接入 IPSec（IKEv1）

IKEv2

- macOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）
- macOS 原生客户端接入 IPSec（用户名/密码 + 预共享密钥认证）

macOS 原生客户端接入 IPSec（IKEv1）

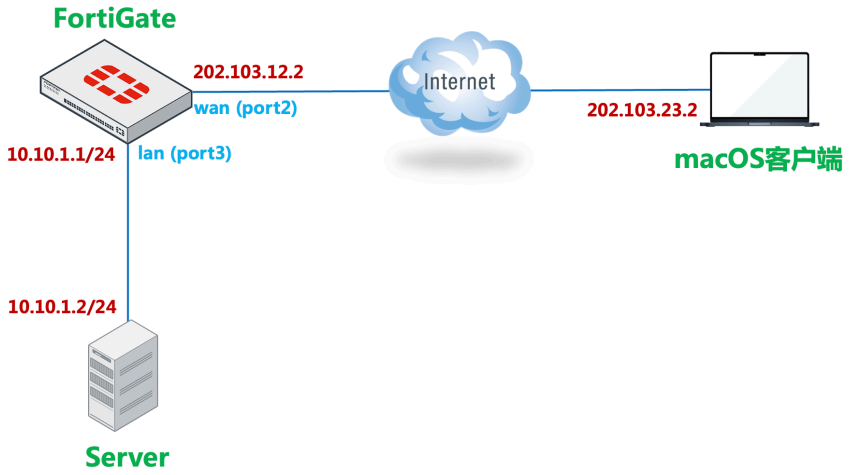
重要

本文基于 macOS 13.2 编写。

网络需求

通过 macOS 原生客户端（IKEv1）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

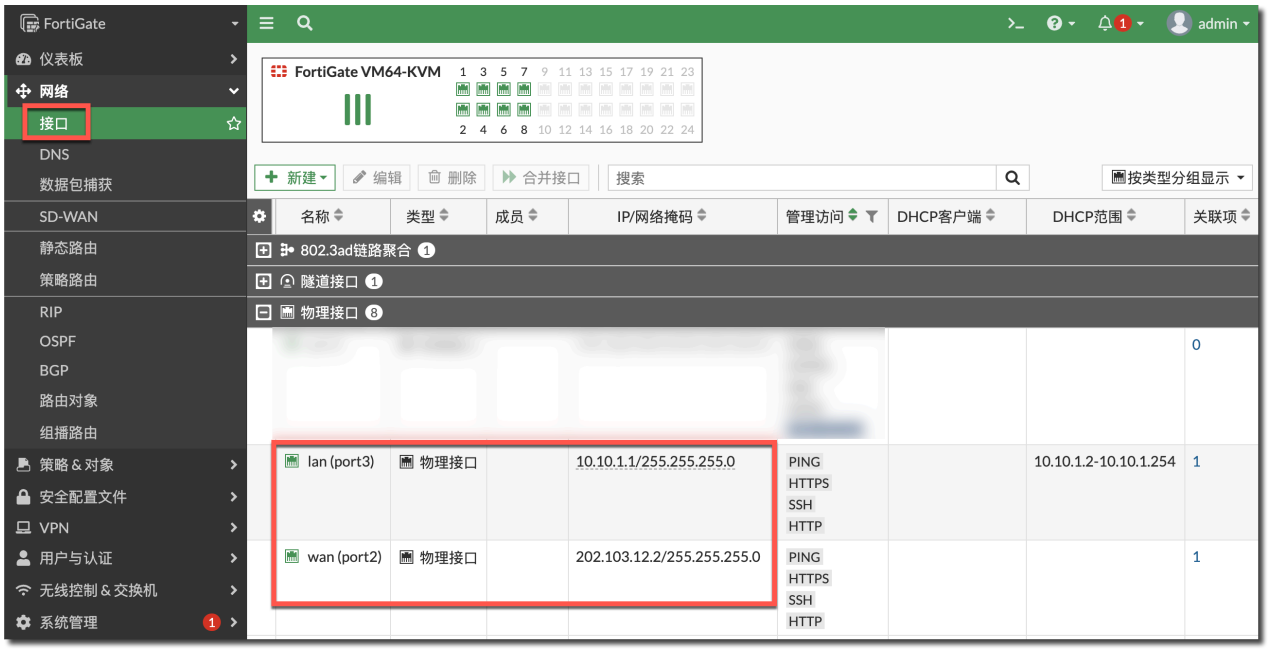
网络拓扑



配置步骤

FortiGate 配置

1. 配置 FortiGate 的接口 IP。



2. 创建本地 VPN 用户和用户组，参考用户与认证 → 本地用户和用户组 ↗章节（如需使用远程认证，请参考用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗或用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。




```
config user local
    edit "vpn_user"
        set type password
        set passwd xxxxxxxx
    next
end
```




```
config user group
    edit "VPN"
        set member "vpn_user"
    next
end
```

3. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



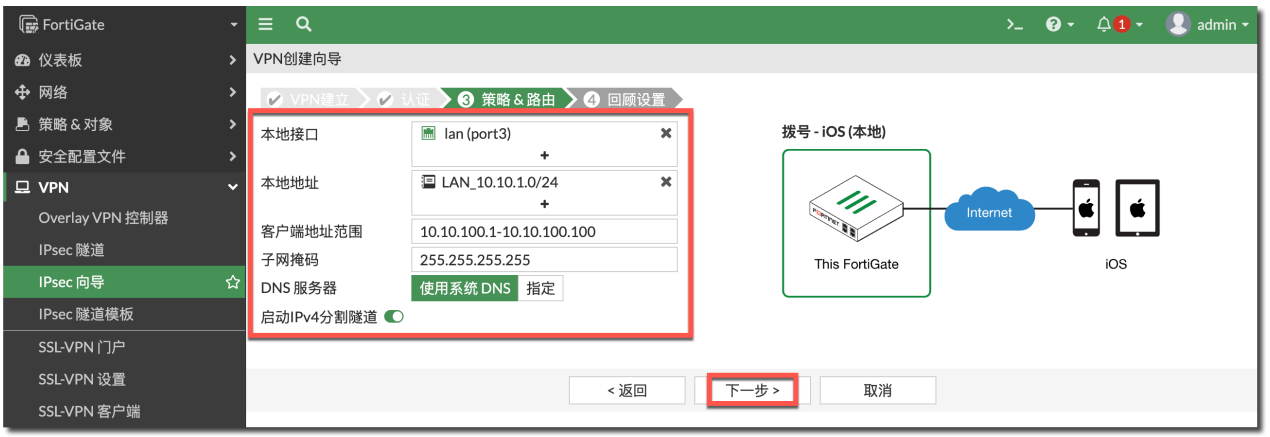
4. 填写 VPN 名称，模版选择远程拨号，远程设备选择 Native，然后选择 iOS Native，点击下一步。



5. 配置流入接口为 wan 口，填入预共享密钥，选择 VPN 用户组，点击下一步。



6. 配置本地接口（VPN 客户端要访问的网络所属出接口），本地地址新建并选择内网的 VPN 网段地址对象，分配客户端地址范围（注意不要和现有网段冲突），DNS 可选择系统 DNS 或指定 DNS，默认开启隧道分割，只有本地地址中的网段会被客户端转发至 VPN 隧道（关闭隧道分割的话，客户端会将自身发起的所有流量转发至 VPN 隧道）。点击下一步。



> **重要**
>
> 如果开启了隧道分割，本地地址不要填写为 0.0.0.0/0，否则会导致客户端连接失败。

7. 检查配置无误后，点击完成按钮，完成隧道建立。





8. 向导配置完成后，FortiGate 会自动创建 IPSec 一阶段和二阶段、内网地址对象、客户端到内网的防火墙访问策略，对应的 CLI 如下：

```
第一阶段和第二阶段VPN配置:
config vpn ipsec phase1-interface
    edit "macOS_IKEv1"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256 aes256-md5 aes256-sha1
        set comments "VPN: macOS_IKEv1 (Created by VPN wizard)"
        set dhgrp 14 5 2
        set wizard-type dialup-ios
        set xauthtype auto
        set authusrgrp "VPN"
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set dns-mode auto
        set ipv4-split-include "macOS_IKEv1_split"
        set psksecret xxxxxxxx
    next
end

config vpn ipsec phase2-interface
    edit "macOS_IKEv1"
        set phase1name "macOS_IKEv1"
        set proposal aes256-sha256 aes256-md5 aes256-sha1
        set pfs disable
        set keepalive enable
        set comments "VPN: macOS_IKEv1 (Created by VPN wizard)"
    next
end
```

地址和地址组：

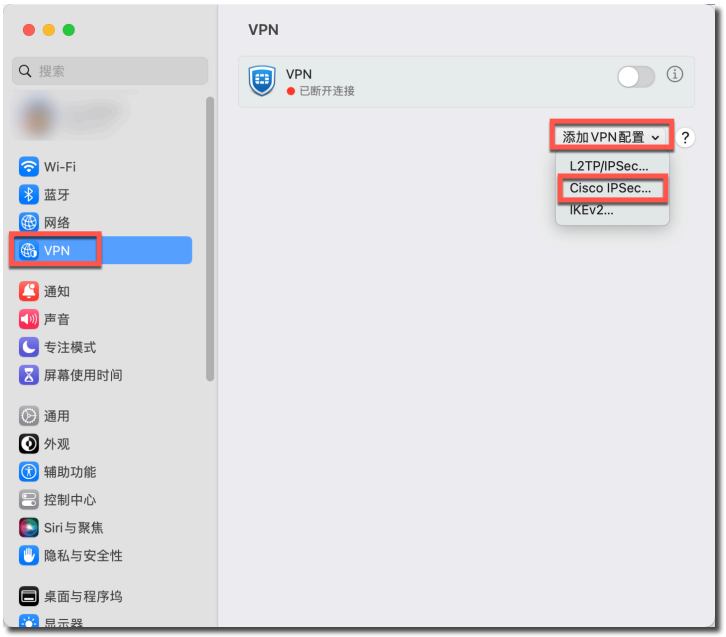
```
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
    edit "macOS_IKEv1_range"
        set type iprange
        set comment "VPN: macOS_IKEv1 (Created by VPN wizard)"
        set start-ip 10.10.100.1
        set end-ip 10.10.100.100
    next
end

config firewall addrgrp
    edit "macOS_IKEv1_split"
        set member "LAN_10.10.1.0/24"
        set comment "VPN: macOS_IKEv1 (Created by VPN wizard)"
    next
end
```

允许客户端访问内网VPN资源的防火墙策略：

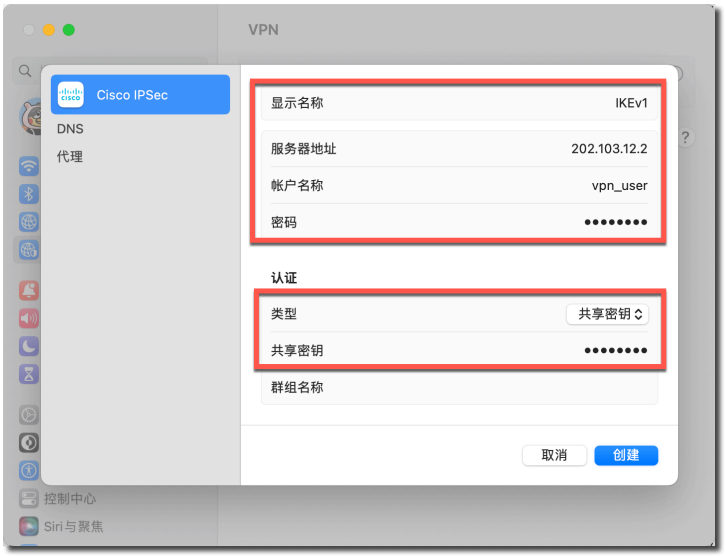
```
config firewall policy
    edit 1
        set name "vpn_macOS_IKEv1_remote_0"
        set srcintf "macOS_IKEv1"
        set dstintf "port3"
        set action accept
        set srcaddr "macOS_IKEv1_range"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
        set nat enable
        set comments "VPN: macOS_IKEv1 (Created by VPN wizard)"
    next
end
```

客户端配置

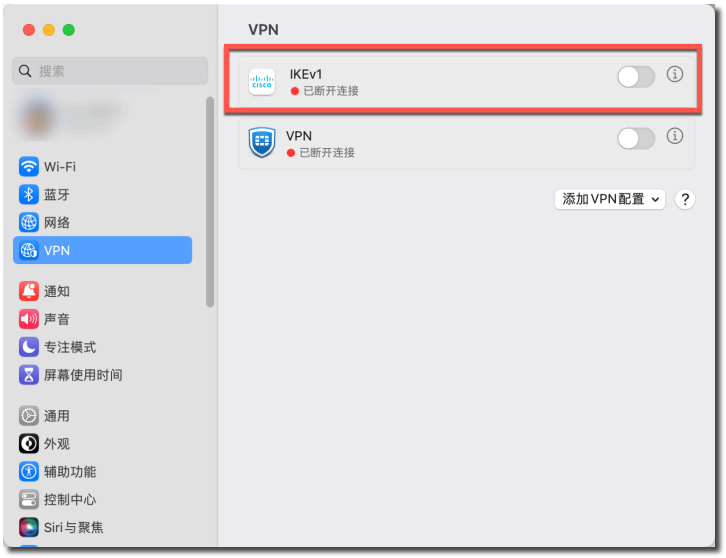
1. 打开系统设置 → VPN，点击“添加 VPN 配置”，选择 Cisco IPSec VPN。



2. 填写 VPN 名称、服务器 IP、用户名、用户密码，认证类型选择共享密钥，填写预共享密钥，其他不填写（如果想要使用群组名称，请在 FortiGate 侧配置），点击创建按钮建立 IPSec。

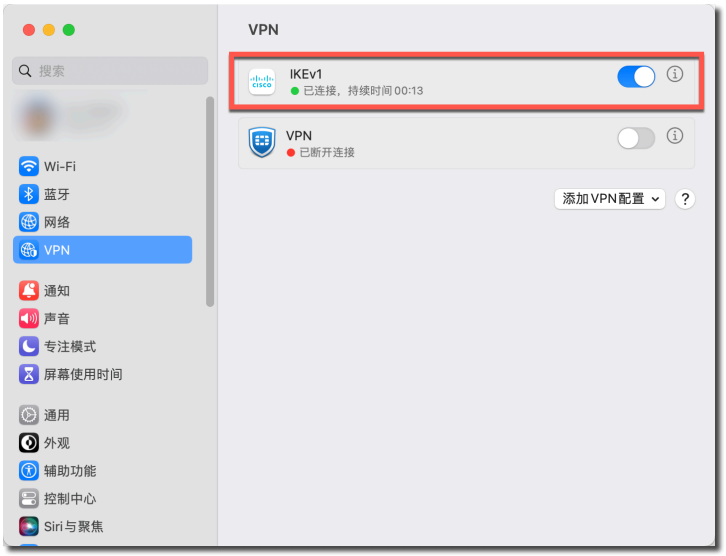


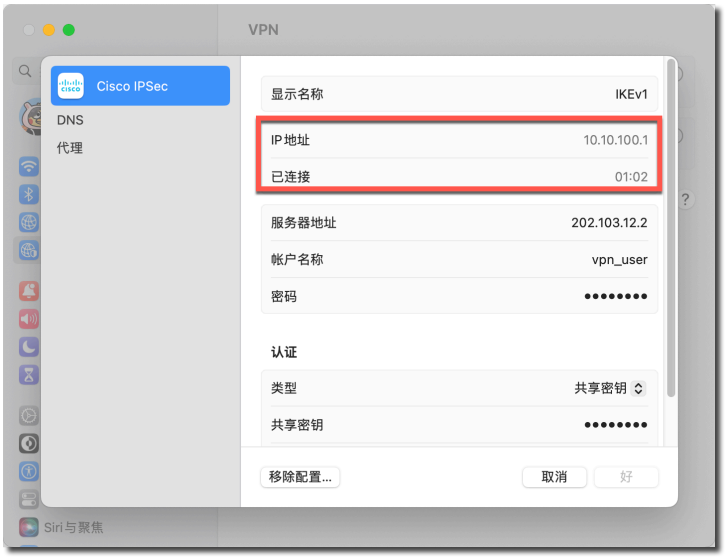
3. VPN 连接配置完成。



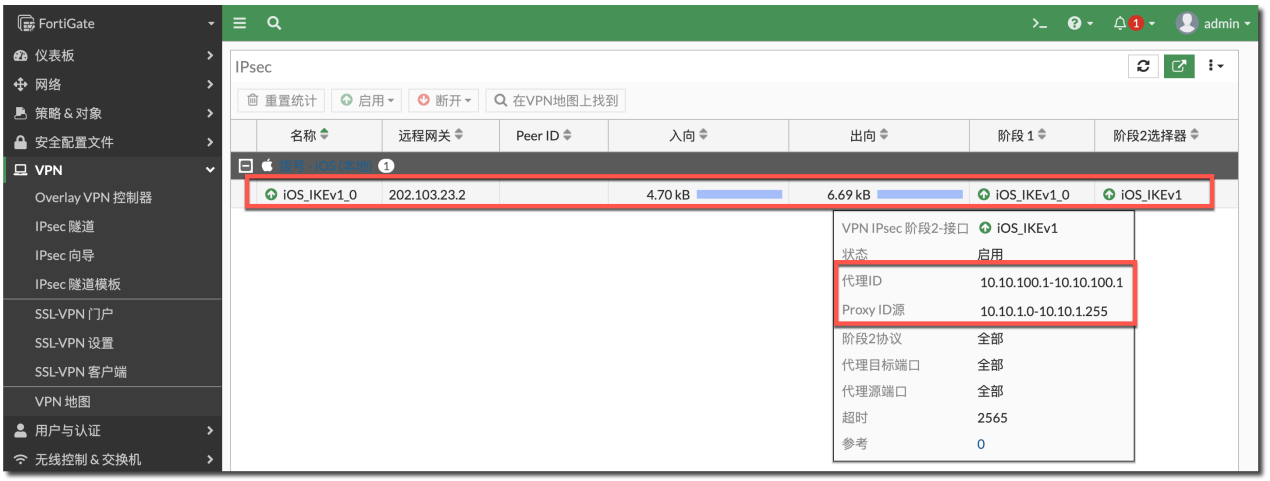
结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。





2. 在 FortiGate 侧查看 IPSec 连接状态。



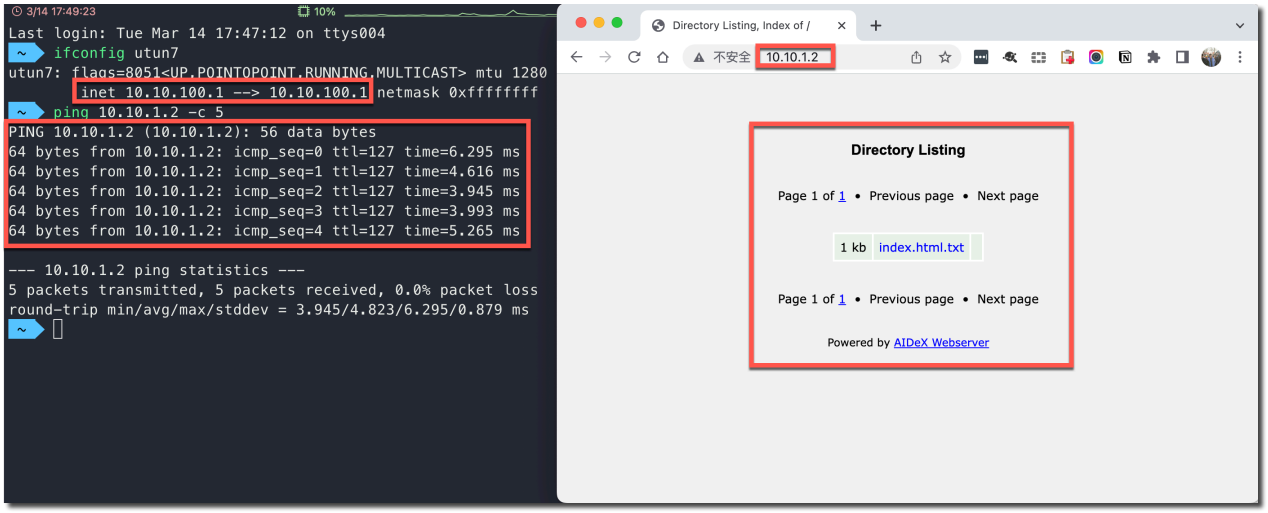

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: macOS_IKEv1_0
version: 1
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:60417
tun_id: 10.10.100.1/::10.0.0.4
remote_location: 0.0.0.0
```

```
network-id: 0
created: 1144s ago
xauth-user: vpn_user
2FA: no
assigned IPv4 address: 10.10.100.1/255.255.255.255
nat: peer
IKE SA: created 1/1  established 1/1  time 60/60/60 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms
  id/spi: 2 1ab45dd9b8ad53a1/229987eaaab4142e
  direction: responder
  status: established 1144-1144s ago = 60ms
  proposal: aes256-sha256
  key: 26d033d72c0d55fe-6652a7159cf151dd-1113c3dd29c29cf9-74dabccc059f96ac
  lifetime/rekey: 3600/2185
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=macOS_IKEv1 ver=1 serial=1 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=42957570 olast=42957570 ad=/0
stat: rxp=115 txp=96 rxb=10525 txb=16656
dpd: mode=on-demand on=0 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=macOS_IKEv1_0 ver=1 serial=4 202.103.12.2:4500→202.103.23.2:60417 tun_id=10.10.100.1 tun_id6=::10.0.0.4
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=macOS_IKEv1 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=873 olast=873 ad=/0
stat: rxp=51 txp=44 rxb=4704 txb=6692
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=silent draft=32 interval=10 remote_port=60417
proxyid=macOS_IKEv1 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:10.10.1.0-10.10.1.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20083 type=00 soft=0 mtu=1422 expire=2440/0B replaywin=2048
       seqno=2d esn=0 replaywin_lastseq=00000033 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3588/3600
  dec: spi=f3a58344 esp=aes key=32 a8ff490c9fb2e89be964d1a6993b11eedf71191e520ca99ed294c8eb4ea26301
       ah=sha256 key=32 7be5011c5d5a80768497ea9a4421bf8016797480cf6e31bd887d7367db35f30b
  enc: spi=05a9cf2e esp=aes key=32 83392937fdf5872826b529822a07a81c22e076d7bd648c8887451a2ce51b8336
       ah=sha256 key=32 a1897271122f887c59149acb755abc2959d1916a2c3340a35ef40f7172809d9b
  dec:pkts/bytes=102/9408, enc:pkts/bytes=88/16868
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=2 dec_npuid=0 enc_npuid=0
```

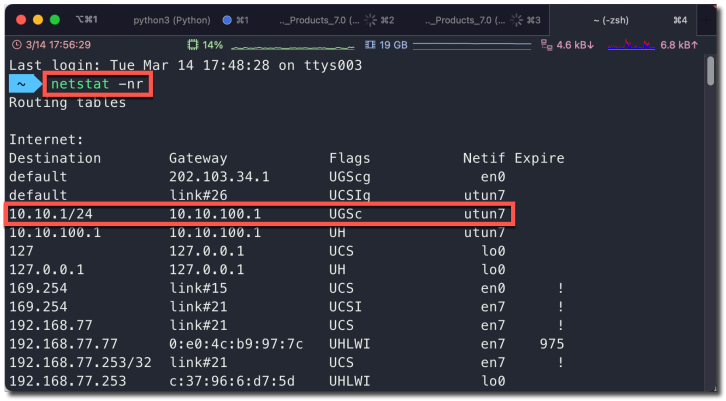
3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=25(macOS_IKEv1)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



5. 查看客户端的路由表（netstat -nr），可以看到隧道分割的路由。



IKEv2

- macOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）
- macOS 原生客户端接入 IPSec（用户名/密码 + 预共享密钥认证）

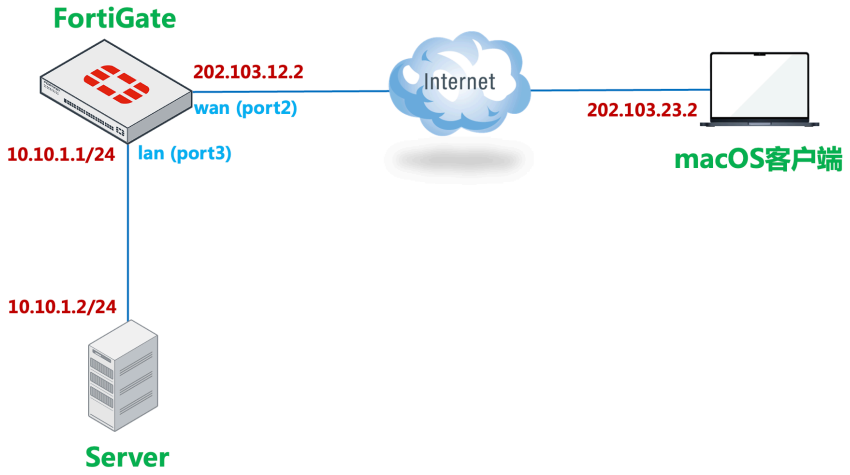
macOS 原生客户端接入 IPSec（IKEv2 预共享密钥认证）

> **重要**
>
> 本文基于 macOS 13.2 编写。

网络需求

通过 macOS 原生客户端（IKEv2 预共享密钥认证）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

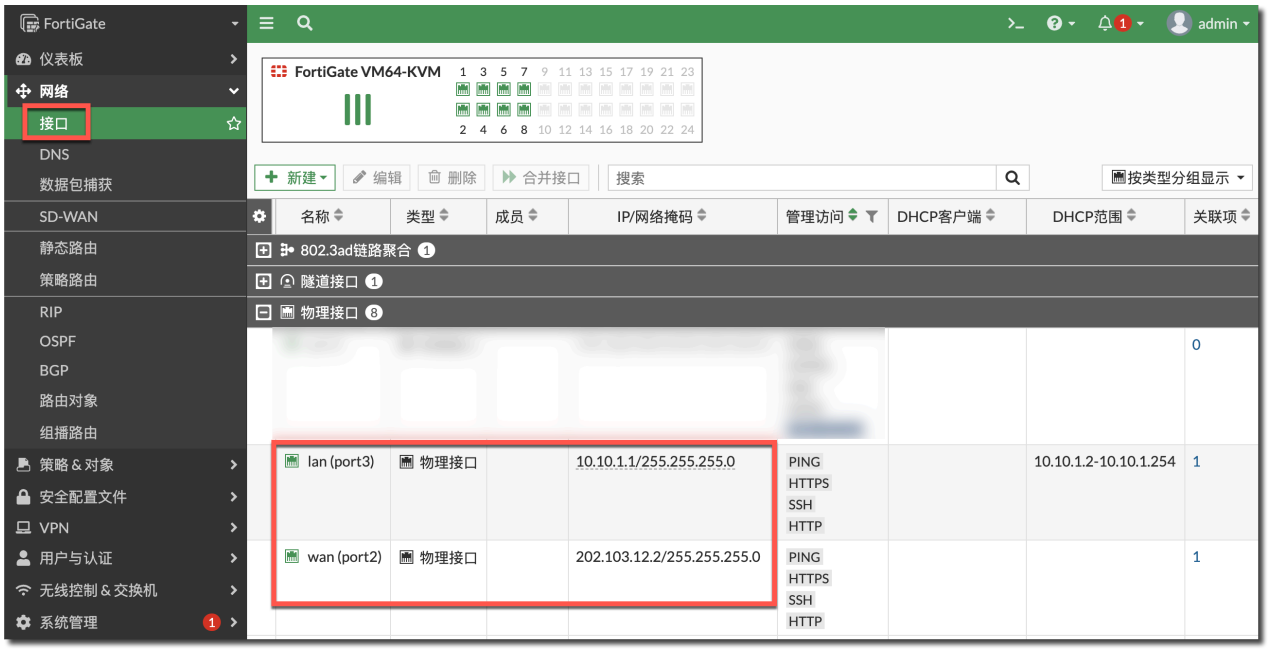
网络拓扑



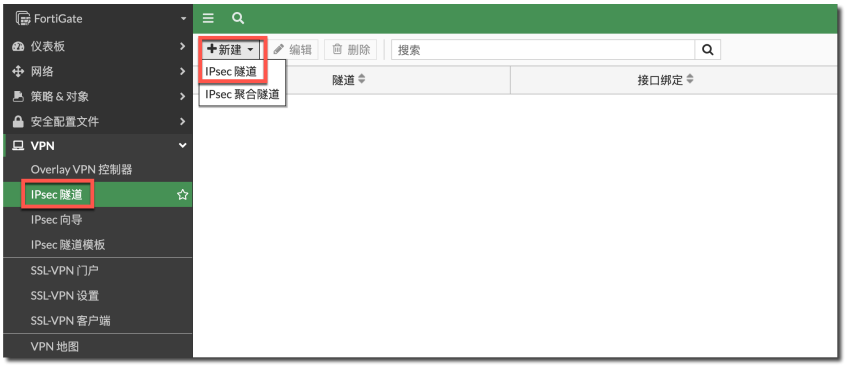
配置步骤

FortiGate 配置

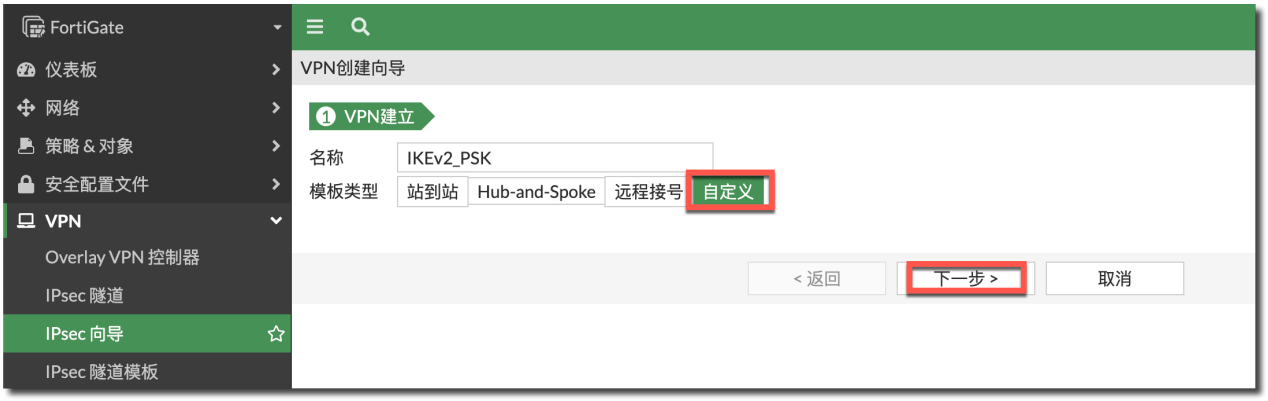
1. 配置 FortiGate 的接口 IP。



2. 进入 FortiGate 的 VPN → IPSec 隧道页面，点击新建按钮，新建 IPSec 隧道。



3. 填写 VPN 名称，模版选择自定义，点击下一步。



4. 按照下图修改配置，其他配置保持默认即可，点击确认下发配置。


FortiGate
仪表板
网络
策略 & 对象
安全配置文件
VPN
Overlay VPN 控制器
IPsec 隧道
IPsec 向导
IPsec 隧道模板
SSL-VPN 门户
SSL-VPN 设置
SSL-VPN 客户端
VPN 地图
用户与认证
无线控制 & 交换机
系统管理
Security Fabric
日志 & 报表
FORTINET
v7.0.10
新VPN隧道
名称
IKEv2_PSK
注释
0/255
网络
选择为拨号用户模式
接口为wan口
IP版本
IPv4
IPv6
对端网关
拨号用户
接口
wan (port2)
本地网关
模式配置
开启模式配置
在模式配置中使用系统DNS
分配 IP从
范围
为客户端分配的IP地址范围和DNS地址
IPv4 模式配置
客户端地址范围
10.10.100.1-10.10.100.100
子网掩码
255.255.255.255
DNS 服务器
10.10.1.2
启动IPv4分割隧道
开启隧道分割，只有地址组中的地址会被客户端转发至VPN隧道
访问可达网络
LAN_10.10.1.0/24
IPv6 模式配置
客户端地址范围
::-::
前缀长度
128
DNS 服务器
::
启用IPv6隧道分割
NAT穿越
启用
禁用
强制的
对等体状态探测
禁用
空闲
按需
DPD重试计数
3
DPD重试间隔
60
s
前向纠错
Egress
Ingress
高级
填写预共享密钥
IKE版本选择2
访问类型修改为"任意对等体ID"
认证
方法
预共享密钥
预共享密钥
IKE
版本
1
2
对等体选项
访问类型
任意对等体ID
阶段 1 提案
添加
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
一阶段算法
Diffie-Hellman 组
32 31 30 29 28 27
21 20 19 18 17 16
15 14 5 2 1
密钥生存时间 (秒)
86400
本地ID
阶段 2 选择器
名称
本地地址
对端地址
IKEv2_PSK
0.0.0.0/0.0.0.0
0.0.0.0/0.0.0.0
新建阶段 2
名称
IKEv2_PSK
注释
本地地址
子网
0.0.0.0/0.0.0.0
对端地址
子网
0.0.0.0/0.0.0.0
高级
阶段 2 提案
添加
加密
AES128
认证
SHA1
加密
AES256
认证
SHA1
加密
AES128
认证
SHA256
加密
AES256
认证
SHA256
二阶段算法
启用重播检测
启用完全前向保密 (PFS)
关闭PFS
本地端口
全部
对端端口
全部
协议
全部
自动密钥保持存活
开启"自动密钥保持存活"
密钥周期(秒/kb)
秒
秒
43200
确认
取消



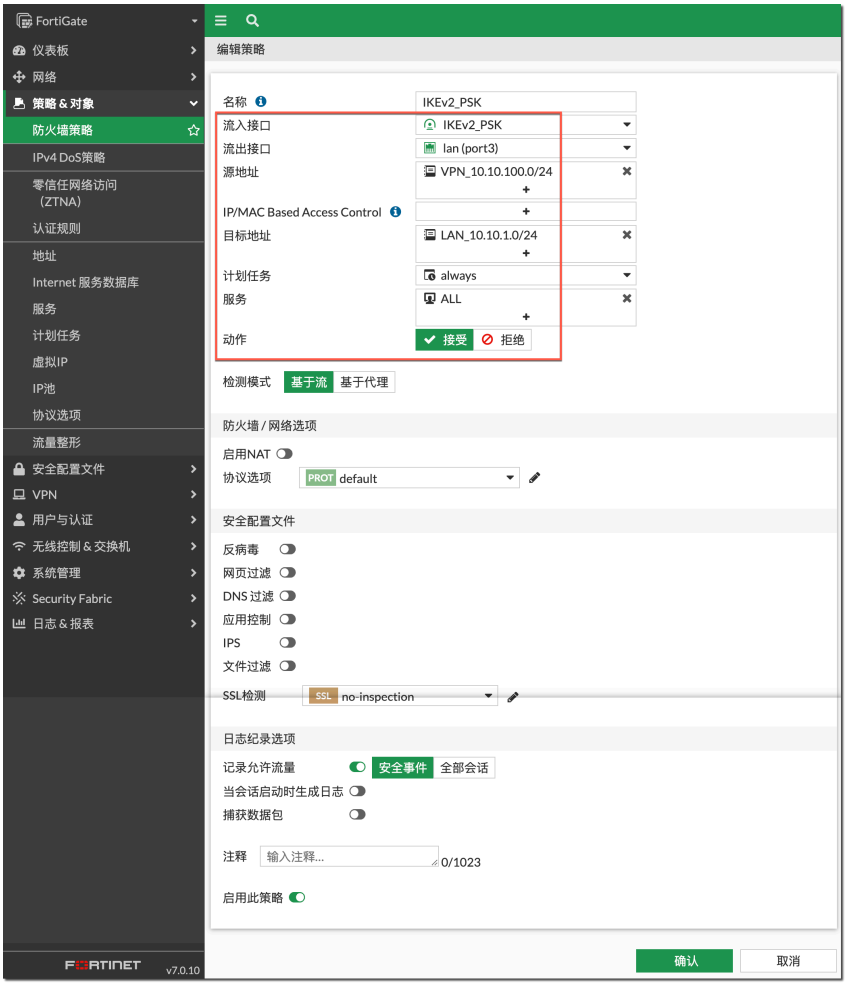
> **重要**
>
> 如果开启了隧道分割，本地地址不要填写为 0.0.0.0/0，否则会导致客户端连接失败。

5. 对应的 CLI 如下：

```
第一阶段配置：
config vpn ipsec phase1-interface
    edit "IKEv2_PSK"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 10.10.1.2
        set proposal aes128-sha256 aes256-sha256
        set dpd on-idle
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100
        set ipv4-split-include "LAN_10.10.1.0/24"
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

第二阶段配置：
config vpn ipsec phase2-interface
    edit "IKEv2_PSK"
        set phase1name "IKEv2_PSK"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256
        set pfs disable
        set keepalive enable
    next
end
```

```
地址对象：
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end
```

6. 配置防火墙策略，允许客户端访问内网 VPN 资源。



客户端地址对象：

```
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end
```

防火墙策略：

```
config firewall policy
    edit 1
        set name "IKEv2_PSK"
        set srcintf "IKEv2_PSK"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

客户端配置

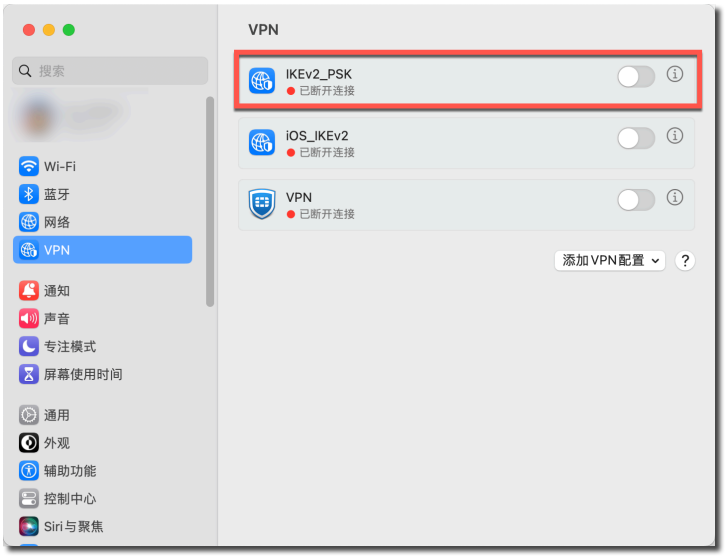
1. 打开系统设置 → VPN，点击“添加 VPN 配置”，选择 IKEv2。



2. 填写 VPN 名称、服务器 IP，远程 ID 随意填写，用户认证选择“无”，机器认证选择“共享密钥”，填写预共享密钥，点击创建完成客户端 VPN 配置。



3. VPN 连接配置完成。

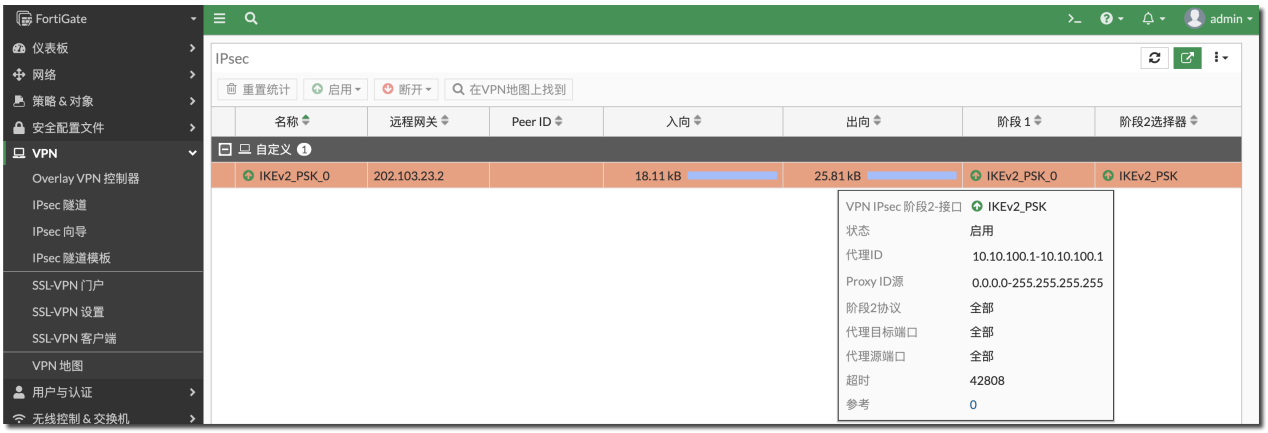


结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。



2. 在 FortiGate 侧查看连接状态。



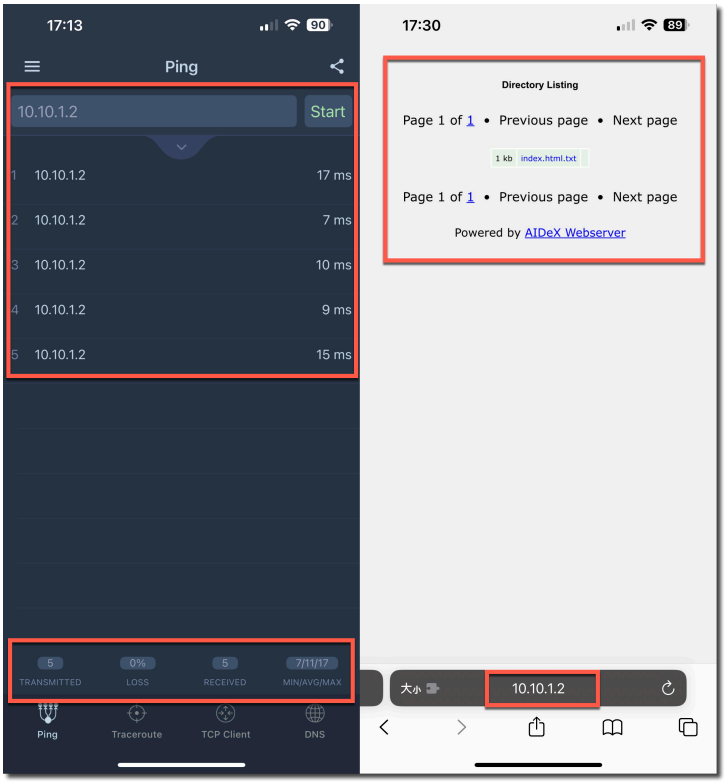

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: IKEv2_PSK_0
version: 2
```

```
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:64916
tun_id: 10.10.100.1/::10.0.0.12
remote_location: 0.0.0.0
network-id: 0
created: 27s ago
assigned IPv4 address: 10.10.100.1/255.255.255.255
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 370/370/370 ms
IPsec SA: created 1/1  established 1/1  time 110/110/110 ms
  id/spi: 16 675a02dbbef8cc4a/096820e48922db26
  direction: responder
  status: established 27-27s ago = 370ms
  proposal: aes256-sha256
  child: no
  SK_ei: d5113c70c4d78f54-f34f30091bf79475-aa35a97d71c0a5aa-2c9859861946d414
  SK_er: 9414d48bd22ecbc1-709aa14d0de11cd6-64b0829447435a9f-3b31a85afdeeff38
  SK_ai: 36305c3ad8a415b5-bee89c242a7faf32-04dee1ec7526fd4d-d952bf68c9014731
  SK_ar: 13adcd000ee30fde-d6c84d005c113924-1c3866f0d5b8a973-224225fb19d7e865
  PPK: no
  message-id sent/recv: 0/2
  lifetime/rekey: 86400/86102
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=IKEv2_PSK_0 ver=2 serial=13 202.103.12.2:4500→202.103.23.2:64916 tun_id=10.10.100.1 tun_id6=::10.0.0.12
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=IKEv2_PSK index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=533 txp=28 rxb=48088 txb=2804
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=64916
proxyid=IKEv2_PSK proto=0 sa=1 ref=15 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20483 type=00 soft=0 mtu=1422 expire=43160/0B replaywin=2048
       seqno=1d esn=0 replaywin_lastseq=00000215 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43191/43200
  dec: spi=4d83686c esp=aes key=32 ccab803299265af9f609977a5bb055b3c66537ac6500719ba9c78d4fa04b6738
       ah=sha256 key=32 b808b44ed4d288c2a2d518f1ff6bf6380c20247cbb1ed2bd3c3ffb3779854d34
  enc: spi=093170d2 esp=aes key=32 63423c52af9b3e35dcccfd7e9d0d1ceaca390652a45506419d8a339e89a5b714
       ah=sha256 key=32 a66edb55e69ac116d48c073458319388b2b413966a273f38898e22580b93adfe
  dec:pkts/bytes=1066/96176, enc:pkts/bytes=56/7812
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=a dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=IKEv2_PSK ver=2 serial=e 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.7 tun_id6=::10.0.0.7 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=43033783 olast=43033783 ad=/0
stat: rxp=3330 txp=1051 rxb=236299 txb=103007
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
 ...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=25(IKEv2_PSK)
 ...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 iOS/macOS 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

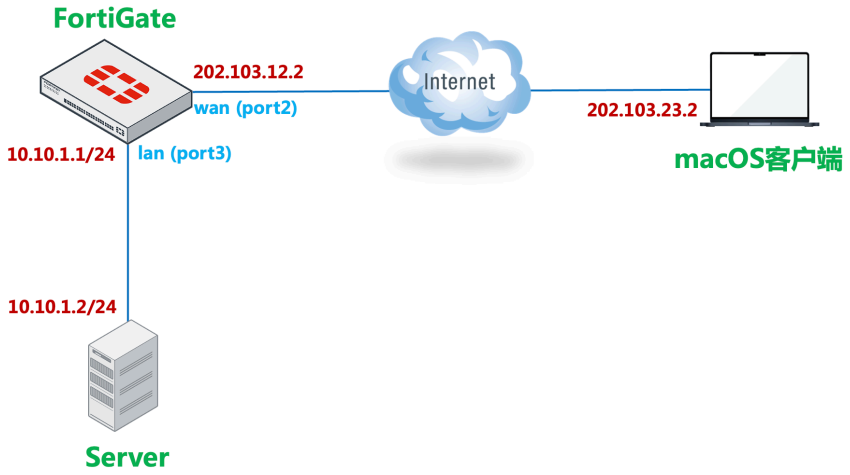
macOS 原生客户端接入 IPSec（用户名/密码 + 预共享密钥认证）

> **重要**
>
> 本文基于 macOS 13.2 编写。

网络需求

通过 macOS 原生客户端（IKEv2 用户名/密码 + 预共享密钥认证）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

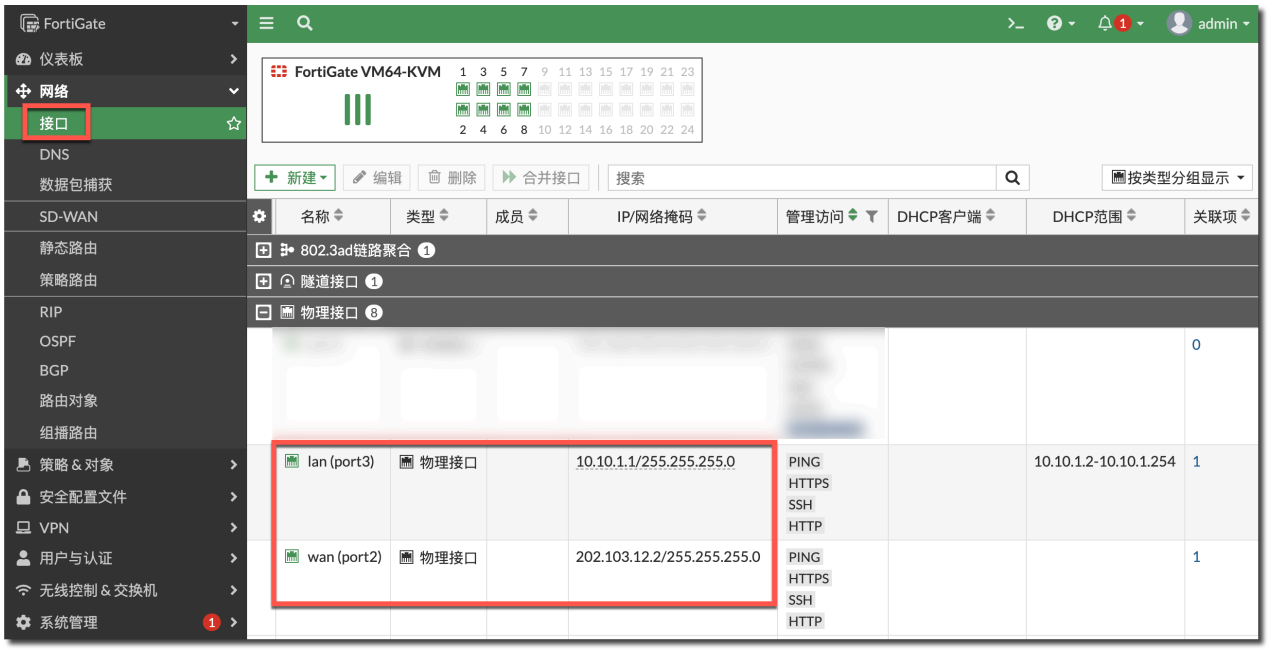
网络拓扑



配置步骤

FortiGate 配置

1. 配置 FortiGate 的接口 IP。



2. 创建本地 VPN 用户和用户组，参考用户与认证 → 本地用户和用户组 ↗章节（如需使用远程认证，请参考用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗或用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。




```
config user local
    edit "vpn_user"
        set type password
        set passwd xxxxxxxx
    next
end
```



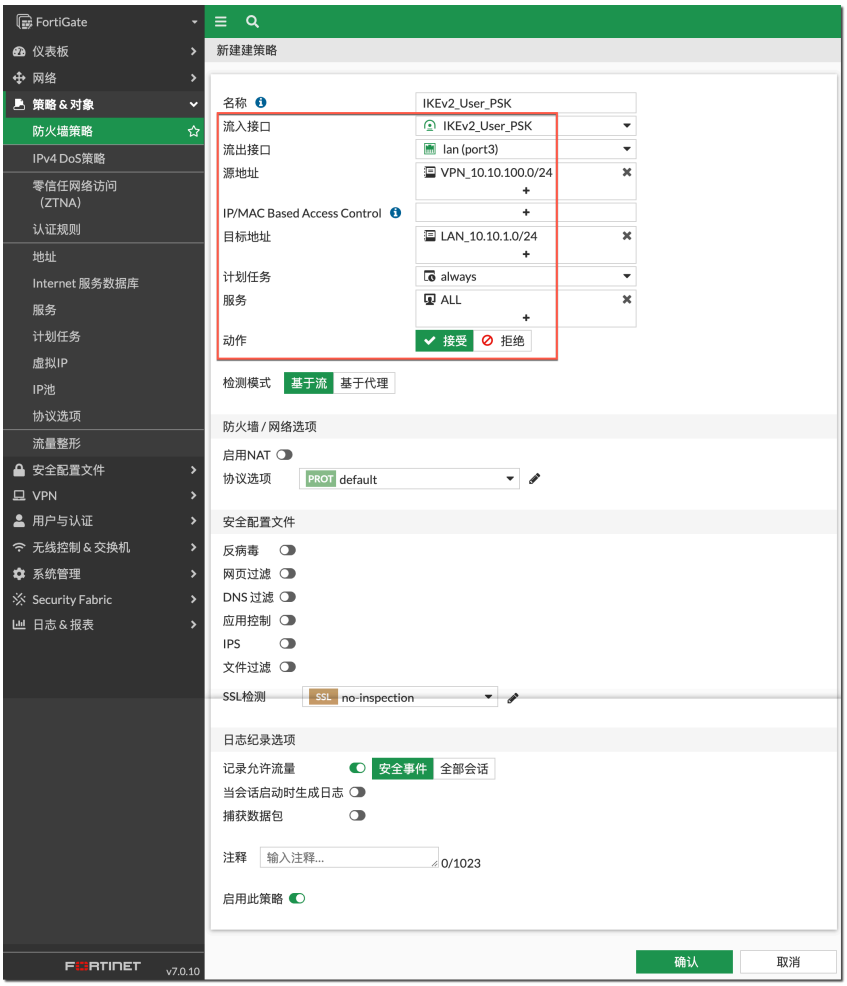

```
config user group
    edit "VPN"
        set member "vpn_user"
    next
end
```

3. 该种方式的 IPSec 需要使用 CLI 方式配置，分别配置 IPSec 一阶段和二阶段。

```
IPSec一阶段配置:
config vpn ipsec phase1-interface
    edit "IKEv2_User_PSK"
        set type dynamic
        set interface "port2"
        set ike-version 2    //IKE协议2
        set peertype any
        set net-device disable
        set mode-cfg enable    //开启为客户端分配IP
        set ipv4-dns-server1 10.10.1.2  //为客户端分配的DNS地址
        set proposal aes256-sha256
        set negotiate-timeout 300
        set dpd on-idle
        set dhgrp 14 5 2
        set eap enable    //需要开启EAP做用户密码认证
        set eap-identity send-request    //开启EAP主动发送认证请求
        set authusrgrp "VPN"    //选择认证的用户组
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100    //客户端分配地址范围
        set ipv4-netmask 255.255.255.0
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end

IPSec二阶段配置:
config vpn ipsec phase2-interface
    edit "IKEv2_User_PSK"
```

```
        set phase1name "IKEv2_User_PSK"
        set proposal aes256-sha256
        set keepalive enable
    next
end
```

4. 配置防火墙策略，允许客户端访问内网 VPN 资源。



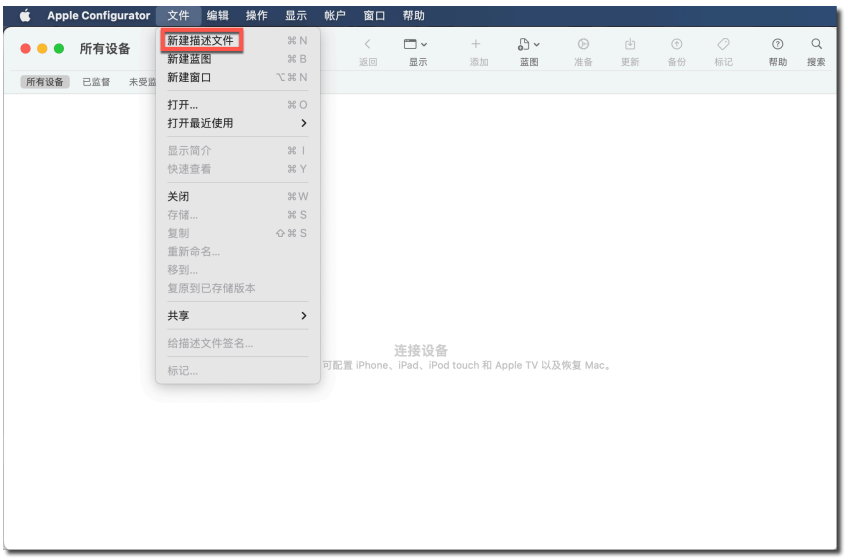

```
VPN资源地址对象:
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end

客户端地址对象:
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end

防火墙策略:
config firewall policy
    edit 1
        set name "IKEv2_User_PSK"
        set srcintf "IKEv2_User_PSK"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
```

```
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

制作和安装描述文件

> **重要**
>
> 想要使用 IKEv2 用户名/密码 + 预共享密钥方式的 VPN 认证，需要使用 Apple Configurator 工具为 iOS/macOS 制作描述文件，该工具可以在 macOS 下的 App Store 中下载。

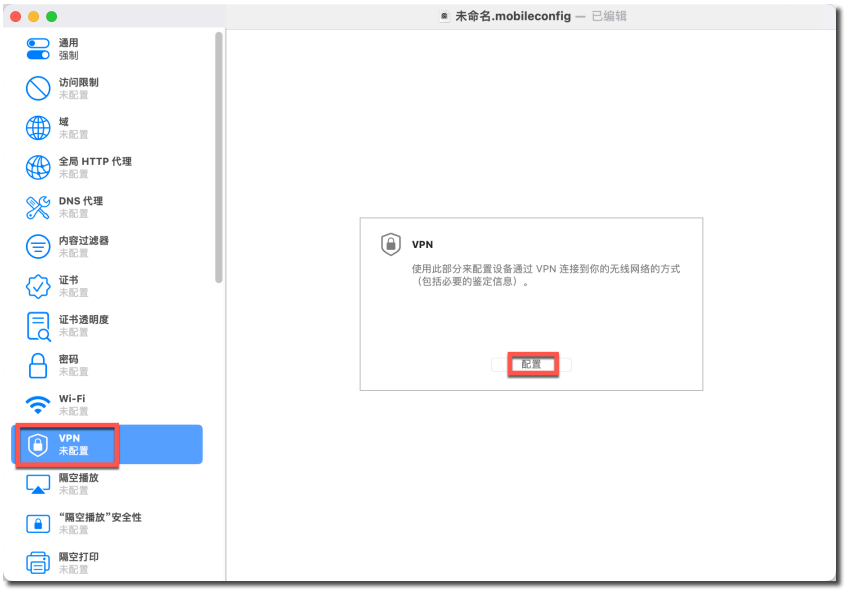
1. 打开 Apple Configurator，点击菜单栏的“文件 → 新建描述文件”（或使用快捷键 Command + N）。



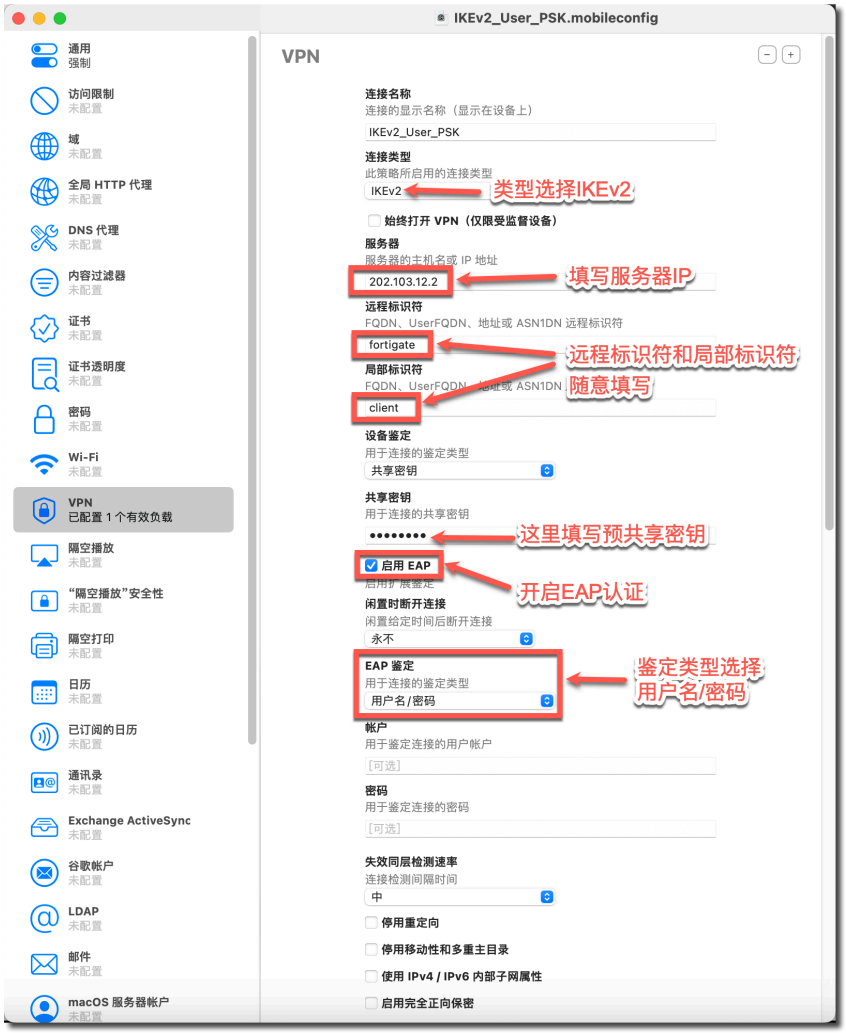
2. 在弹出的配置文件编辑器中，配置描述文件的名称。



3. 选择 VPN 选项卡，点击配置按钮。



4. 配置描述文件，选择连接类型为 IKEv2，填写服务器 IP 地址，远程标识符和局部标识符随意填写，填写预共享密钥，开启 EAP，选择 EAP 鉴定为用户名/密码模式，其他配置保持默认即可（或根据实际需求更改）。


IKEv2_User_PSK.mobileconfig
通用
强制
访问限制
未配置
域
未配置
全局 HTTP 代理
未配置
DNS 代理
未配置
内容过滤器
未配置
证书
未配置
证书透明度
未配置
密码
未配置
Wi-Fi
未配置
VPN
已配置 1 个有效负载
隔空播放
未配置
“隔空播放”安全性
未配置
隔空打印
未配置
日历
未配置
已订阅的日历
未配置
通讯录
未配置
Exchange ActiveSync
未配置
谷歌帐户
未配置
LDAP
未配置
邮件
未配置
macOS 服务器帐户
未配置
VPN
连接名称
连接的显示名称（显示在设备上）
IKEv2_User_PSK
连接类型
此策略所启用的连接类型
IKEv2
类型选择IKEv2
始终打开 VPN（仅限受监督设备）
服务器
服务器的主机名或 IP 地址
202.103.12.2
填写服务器IP
远程标识符
FQDN、UserFQDN、地址或 ASN1DN 远程标识符
fortigate
远程标识符和局部标识符
随意填写
局部标识符
FQDN、UserFQDN、地址或 ASN1DN
client
设备鉴定
用于连接的鉴定类型
共享密钥
共享密钥
用于连接的共享密钥
••••••••
这里填写预共享密钥
启用 EAP
开启EAP认证
闲置时断开连接
闲置给定时间后断开连接
永不
EAP 鉴定
用于连接的鉴定类型
用户名/密码
鉴定类型选择
用户名/密码
帐户
用于鉴定连接的用户帐户
[可选]
密码
用于鉴定连接的密码
[可选]
失效同层检测速率
连接检测间隔时间
中
停用重定向
停用移动性和多重主目录
使用 IPv4 / IPv6 内部子网属性
启用完全正向保密



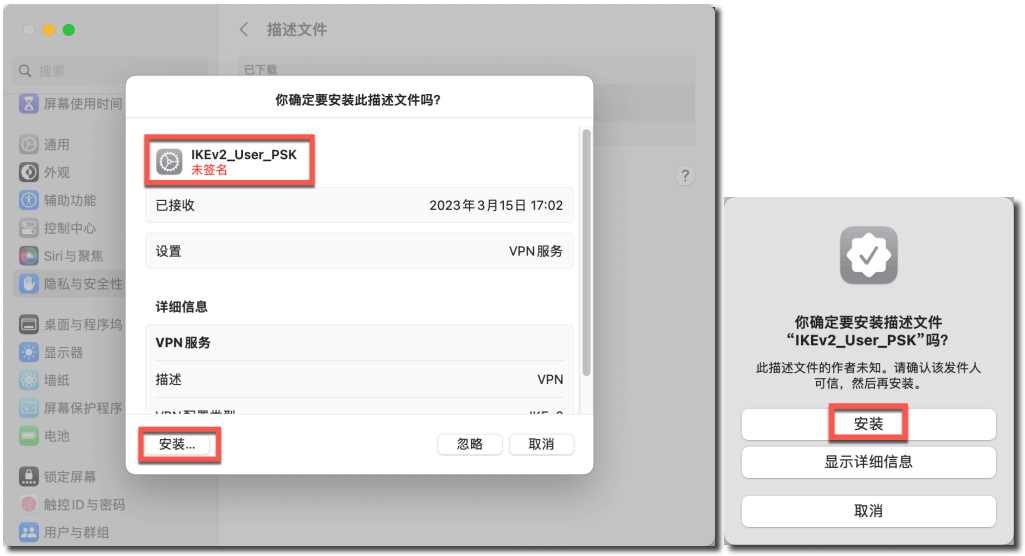
5. 选择菜单栏的“文件 → 存储”按钮（或使用快捷键 Command + S），保存该描述文件到本地。



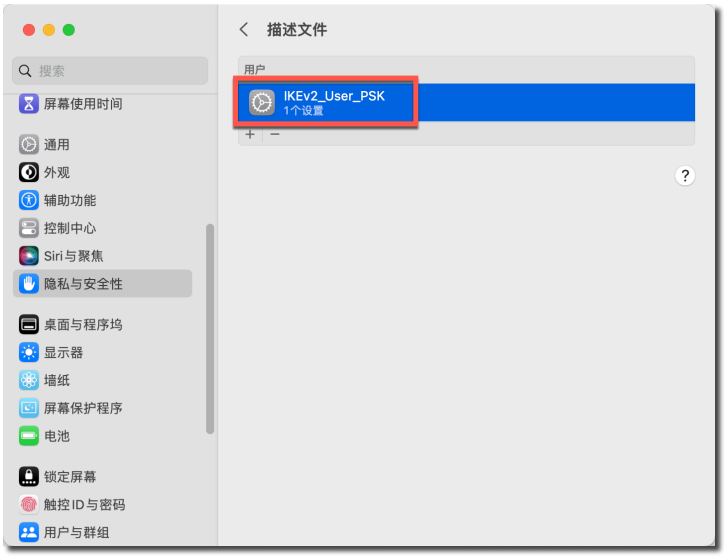
6. 将描述文件发送至客户端 macOS 电脑中，macOS 下直接双击该描述文件安装，会弹出系统提示，在系统设置中查看该描述文件。



7. 进入系统设置 → 隐私与安全 → 其他 → 描述文件，双击“已下载”中的描述文件，确认描述文件无误后，点击安装按钮，在弹出的确认框中点击“安装”（随后需要系统密码验证）。

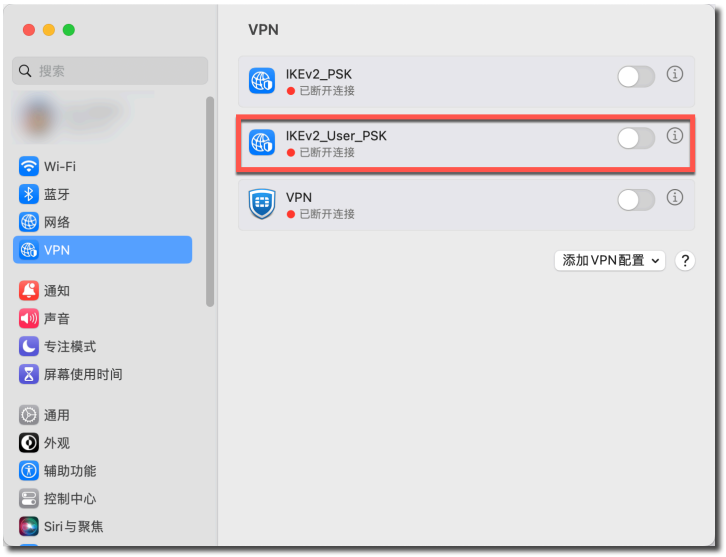


8. 描述文件安装完成。

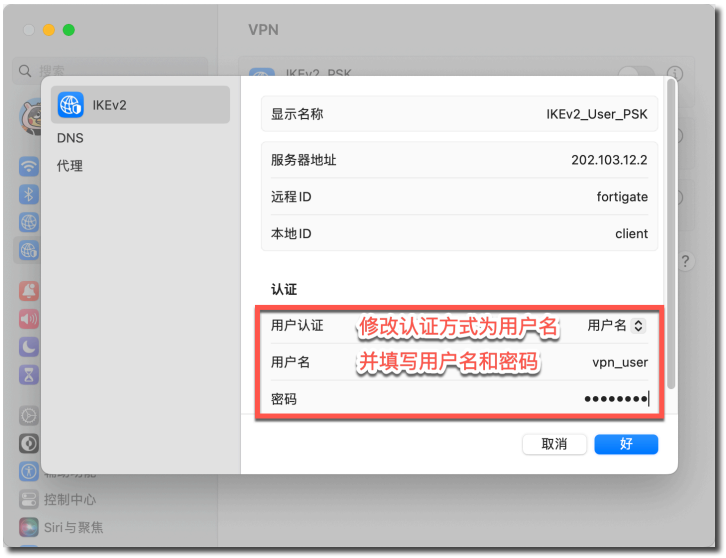


客户端配置

1. 打开系统设置 → VPN，可以看到描述文件中的 VPN 连接已经自动添加。

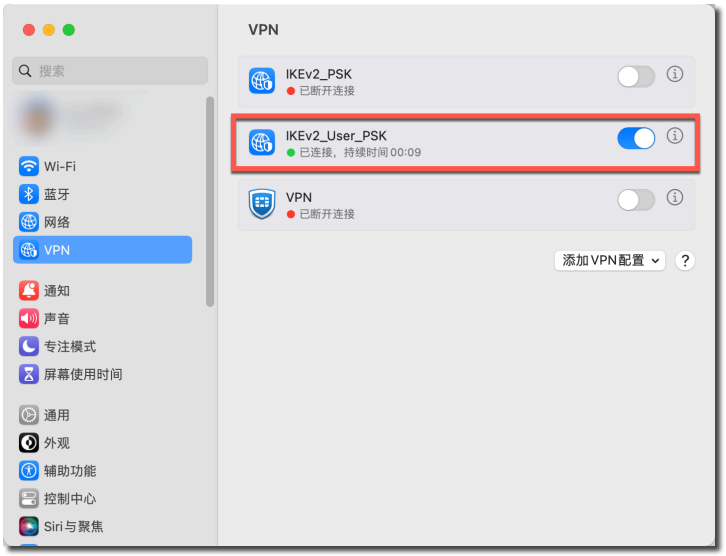


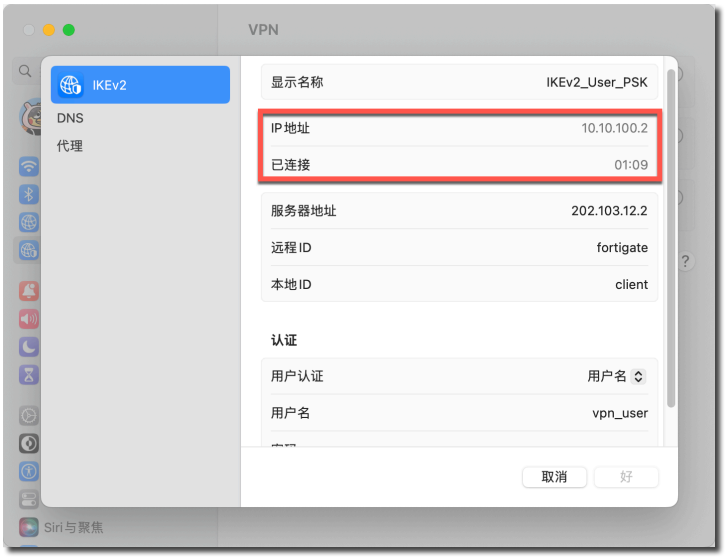
2. 编辑该 VPN 连接，将“用户认证”修改为“用户名”，并填写用户名和密码，点击“好”保存配置。



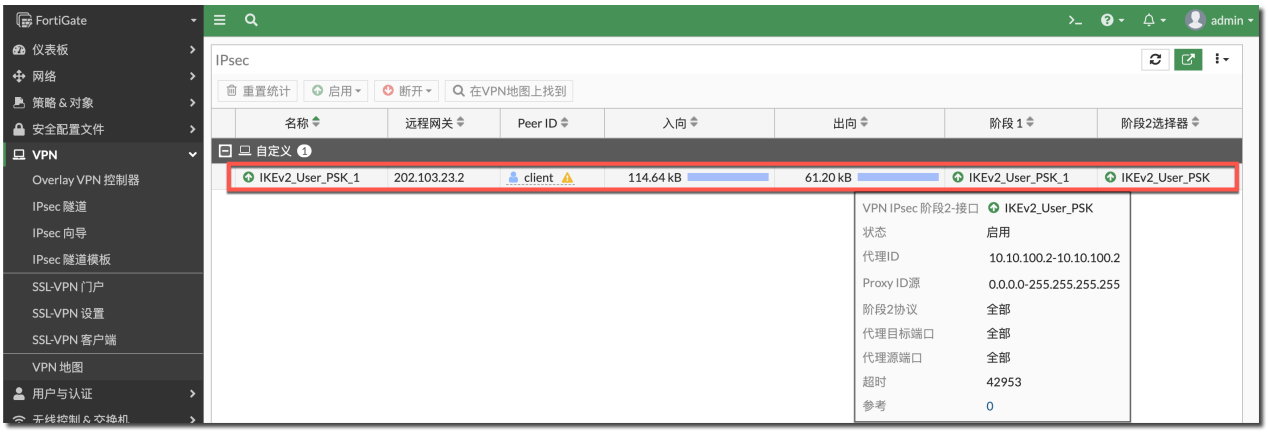
结果验证

1. 在客户端点击连接按钮，查看连接状态，点击 VPN 连接右侧的信息按钮，可以看到连接的具体信息，获取的 IP 为 FortiGate 配置的网段。





2. 在 FortiGate 侧查看连接状态。



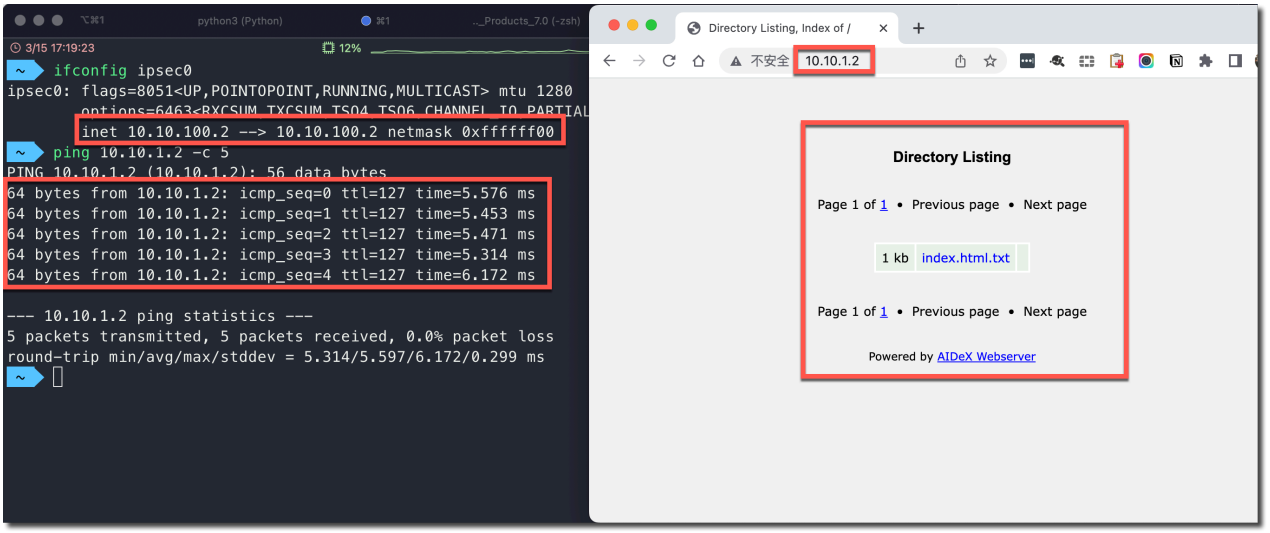

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: IKEv2_User_PSK_1
version: 2
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:60422
tun_id: 10.10.100.2/::10.0.0.18
remote_location: 0.0.0.0
network-id: 0
created: 304s ago
```

```
eap-user: vpn_user
2FA: no
peer-id: client
peer-id-auth: no
assigned IPv4 address: 10.10.100.2/255.255.255.0
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 80/80/80 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 15 4d29f0750955cfd8/2910aaac36020ad6
  direction: responder
  status: established 304-304s ago = 80ms
  proposal: aes256-sha256
  child: no
  SK_ei: c447820d037114b1-50c54fe2851c9be3-0a85e0a1dd660bda-44fb37b6cdfac588
  SK_er: e674721b86f9e390-bdb97e39262e21de-08c0ceeb85f8f4e9-4cd4237c6e4b45a2
  SK_ai: 49d92f240fce91f9-933fa239f76db314-21b8279f552a327f-fbdbed66fbe4a30f
  SK_ar: 68e4bee505827ac5-72af90407c973fe0-34ef0b85985cea0e-a596b75344157af4
  PPK: no
  message-id sent/recv: 0/6
  lifetime/rekey: 86400/85825
  DPD sent/recv: 00000000/00000000
  peer-id: client

FortiGate # diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=IKEv2_User_PSK ver=2 serial=14 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.9 tun_id6=::10.0.0.9 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=43041062 olast=43041062 ad=/0
stat: rxp=5440 txp=1378 rxb=374506 txb=142425
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=IKEv2_User_PSK_1 ver=2 serial=1b 202.103.12.2:4500→202.103.23.2:60422 tun_id=10.10.100.2 tun_id6=::10.0.0.18
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=IKEv2_User_PSK index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=2638 txp=815 rxb=176380 txb=80140
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=60422
proxyid=IKEv2_User_PSK proto=0 sa=1 ref=97 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.2-10.10.100.2:0
  SA:  ref=3 options=20683 type=00 soft=0 mtu=1422 expire=42881/0B replaywin=2048
       seqno=330 esn=0 replaywin_lastseq=00000a4e qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43188/43200
  dec: spi=7b8ba379 esp=aes key=32 7ba541f88e4abc0197580faa828f71aac8f501399ff5962c96bc5ae43c037ece
       ah=sha256 key=32 e2082800cf7dce1c6796d3080969319ec9e52b536e1ba4d4edeefc165f76ff4e
  enc: spi=011f2d7b esp=aes key=32 01b2eb5d0ad5876f35f60382599e97fe84eed38d6023fb8bb45f42893a3da0dd
       ah=sha256 key=32 30d5bed3a56098c277f1213e94bb792bffff86074edfe64dc8761bee1c0403fa
  dec:pkts/bytes=5276/352760, enc:pkts/bytes=1630/223528
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=10 dec_npuid=0 enc_npuid=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.2/32 pref=0.0.0.0 gwy=10.10.100.2
dev=27(IKEv2_User_PSK)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



重要

实际测试 iOS/macOS 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

Windows 系统

- Windows 原生客户端接入 IPSec IKEv1（L2TP over IPSec）
- Windows 原生客户端接入 IPSec（IKEv2）
- 使用 Windows 系统防火墙与 FortiGate 建立 IPSec VPN
- L2TP VPN
- PPTP VPN

Windows 原生客户端接入 IPSec IKEv1（L2TP over IPSec）

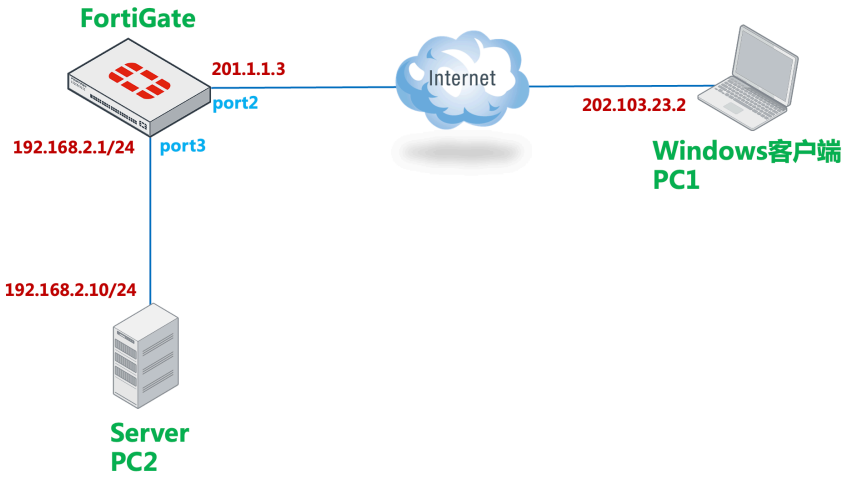
> **重要**
>
> 本文基于 Windows 10 专业版编写。

网络需求

通过 Windows 原生客户端（IKEv1）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

> **重要**
>
> Windows 系统使用 IKEv1，必须配合 L2TP 使用，也就是 L2TP over IPSec。

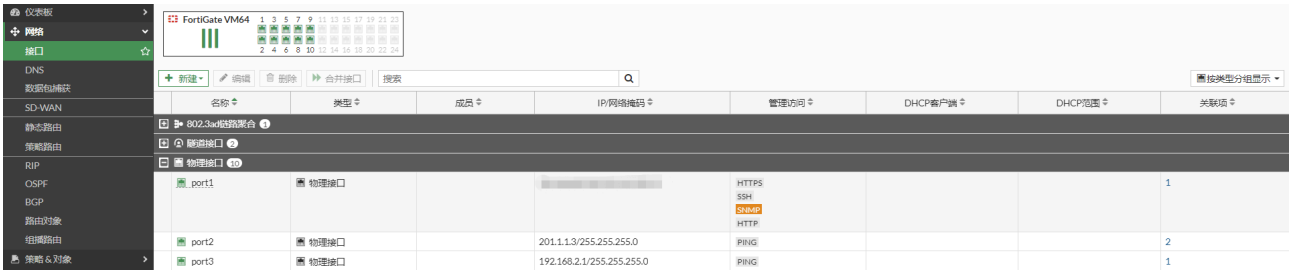
网络拓扑

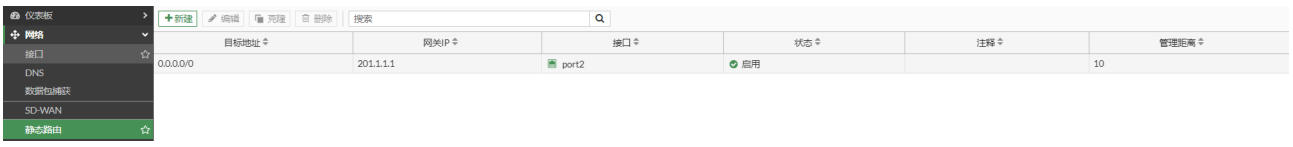


配置步骤

配置 FortiGate

1.基础配置







2.创建用户并将用户加入到用户组





3.创建地址对象



4.配置 L2TP over IPSEC VPN

选择“IPsec 隧道”，点击“新建”，选择“远程拨号”-->"Native"-->“Windows Native”，然后点击下一步。



设置流入接口，预共享秘钥及用户组。



设置本地接口，地址地址以及客户端地址范围。



VPN 向导即将创建的对象。



创建完成。



查看配置

通过“VPN 创建向导”可以很方便的配置 VPN，但我们需要知道向导具体做了哪些配置。

1.创建 IPSEC VPN



对应的命令行

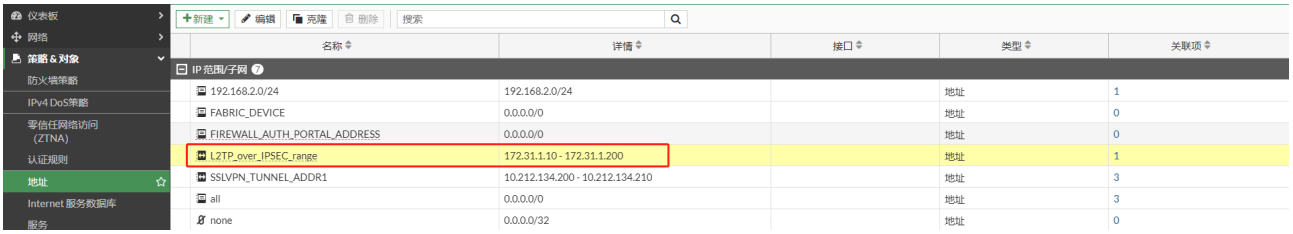
```
config vpn ipsec phase1-interface
    edit "L2TP_over_IPSEC"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes256-md5 3des-sha1 aes192-sha1
        set comments "VPN: L2TP_over_IPSEC (Created by VPN wizard)"
        set dhgrp 2
        set wizard-type dialup-windows
        set psksecret ENC
DCMMcDpdFzv7KXC1lwF5xK3SdOBK0RXf8G8UhEXU+wFRq58XOOoZOcDVaKxLd/mBg1CgkwMqEEXV+IwHs1kpFOTfoGAG428fkKO8BmwlfmDerRWCu5W8IcI6Dw
sDF1PllXIefGl+3H2tjySQ8gm6SeBCKD6XLFjRVlHIW+EwJiTfC1c+bPIEBq60ii9Puckspiec0g==
    next
end
config vpn ipsec phase2-interface
    edit "L2TP_over_IPSEC"
        set phase1name "L2TP_over_IPSEC"
        set proposal aes256-md5 3des-sha1 aes192-sha1
```

```
        set pfs disable
        set encapsulation transport-mode
        set l2tp enable
        set comments "VPN: L2TP_over_IPSEC (Created by VPN wizard)"
        set keylifeseconds 3600
    next
end
```

2.创建 L2TP

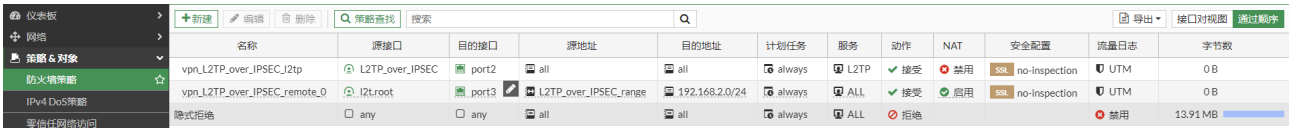
```
config vpn l2tp
    set status enable
    set eip 172.31.1.200
    set sip 172.31.1.10
    set usrgrp "L2TP_over_IPSEC_Group"
end
```

3.创建地址对象

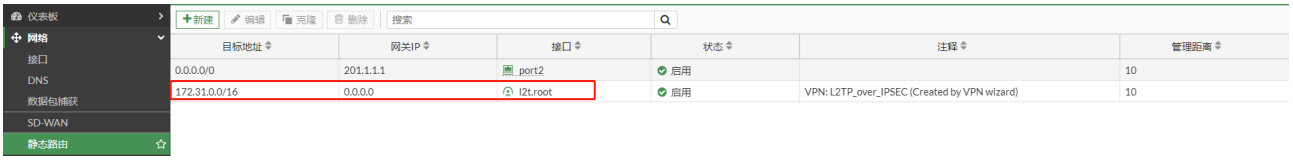


4.创建策略

第 1 条是 L2TP over IPSEC 的策略；第 2 条是 L2TP 到内网的策略。模板没有创建到 Internet 的策略，如果有需求，则要单独创建。

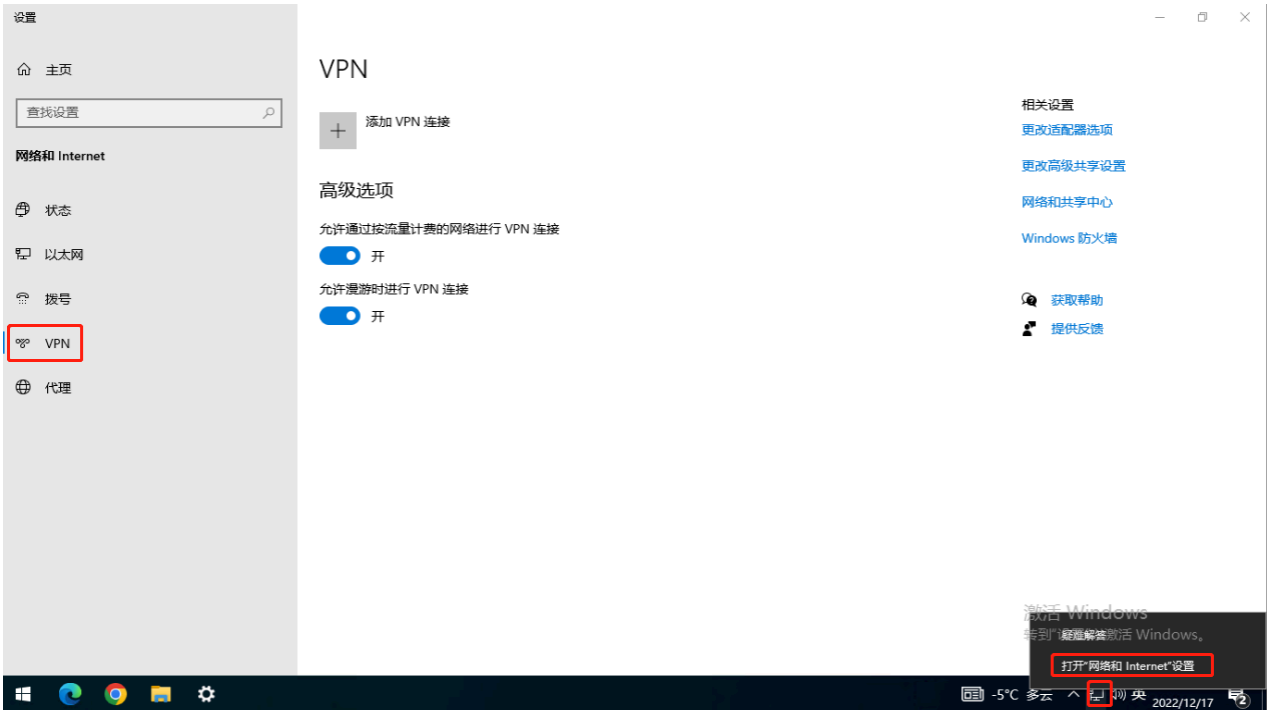


5.创建路由



配置 Windows 客户端

1. 右击本地连接图标，选择“打开网络和 Internet 设置”，选择 VPN。



2. 点击“添加 VPN 连接”，设置 L2TP over IPSEC，然后点击保存。



3. L2TP over IPSEC 创建完成。



结果验证

1. 点击创建的“L2TP_Over_IPSEC”，点击连接



2. 连接成功



查看 L2TP_Over_IPSEC 获取的地址

```
C:\Users\admin>ipconfig/all
以太网适配器 Ethernet1:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   物理地址. . . . . . . . . . . . . : 00-0C-29-91-0B-35
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::bc4e:191a:510a:62ba%7(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.1.1.6(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.1.1.1
   DHCPv6 IAID . . . . . . . . . . . : 402656297
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-29-1C-6E-ED-00-0C-29-91-0B-2B
   DNS 服务器  . . . . . . . . . . . : 192.168.91.208
                                       114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

PPP 适配器 L2TP_Over_IPSEC:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : L2TP_Over_IPSEC
   物理地址. . . . . . . . . . . . . :
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . : 172.31.1.11(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.255
   默认网关. . . . . . . . . . . . . : 0.0.0.0
   DNS 服务器  . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

查看路由表

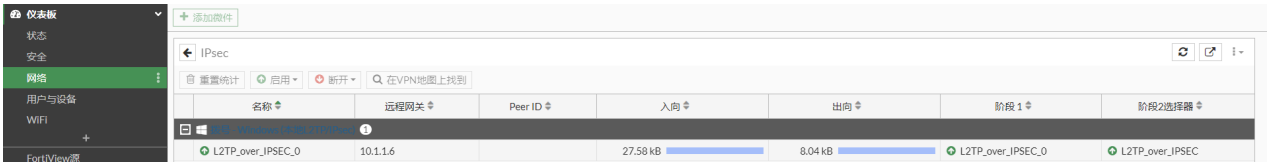
```
C:\Users\admin>route print
===========================================================================
接口列表
 10...00 09 0f fe 00 01 ......Fortinet Virtual Ethernet Adapter (NDIS 6.30)
  5...00 09 0f aa 00 01 ......Fortinet SSL VPN Virtual Ethernet Adapter
 13...00 0c 29 91 0b 2b ......Intel(R) 82574L Gigabit Network Connection
  7...00 0c 29 91 0b 35 ......Intel(R) 82574L Gigabit Network Connection #2
 43...........................L2TP_Over_IPSEC
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
```

```
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0         10.1.1.1         10.1.1.6   4506
          0.0.0.0          0.0.0.0            在链路上       172.31.1.11     26
         10.1.1.0    255.255.255.0            在链路上          10.1.1.6   4506
         10.1.1.6  255.255.255.255            在链路上          10.1.1.6   4506
       10.1.1.255  255.255.255.255            在链路上          10.1.1.6   4506
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1   4556
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1   4556
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1   4556
      172.31.1.11  255.255.255.255            在链路上       172.31.1.11    281
        201.1.1.3  255.255.255.255         10.1.1.1         10.1.1.6   4251
```

3. 访问内部主机

```
C:\Users\admin>ping 192.168.2.10

正在 Ping 192.168.2.10 具有 32 字节的数据:
来自 192.168.2.10 的回复: 字节=32 时间=1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间=1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间=1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63

192.168.2.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

4. FortiGate 查看 L2TP over IPSEC 连接




```
# diagnose vpn ike gateway list

vd: root/0
name: L2TP_over_IPSEC_0
version: 1
interface: port2 6
addr: 201.1.1.3:500 → 10.1.1.6:500
tun_id: 10.1.1.6/::10.0.0.2
remote_location: 0.0.0.0
network-id: 0
created: 667s ago
IKE SA: created 1/1  established 1/1  time 40/40/40 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 0 3074e3c6042e3bd7/445ef45c6bf46ef9
  direction: responder
  status: established 667-667s ago = 40ms
  proposal: 3des-sha1
  key: b894656df6d73056-3c8f6e0801b4977e-f932994583512431
  lifetime/rekey: 28800/27862
  DPD sent/recv: 00000000/00000000

# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=L2TP_over_IPSEC ver=1 serial=8 201.1.1.3:0→0.0.0.0:0 tun_id=10.0.0.3 tun_id6=::10.0.0.3 dst_mtu=0 dpd-link=on
weight=1
bound_if=6 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=44533053 olast=44533053 ad=/0
stat: rxp=403 txp=180 rxb=30057 txb=8731
dpd: mode=on-demand on=0 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
```

```
run_tally=0
--------------------------------------------------------------------
name=L2TP_over_IPSEC_0 ver=1 serial=9 201.1.1.3:0->10.1.1.6:0 tun_id=10.1.1.6 tun_id6=::10.0.0.2 dst_mtu=1500 dpd-
link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/8872 options[22a8]=npu rgwy-chg frag-rfc  run_state=0
role=primary accept_traffic=1 overlay_id=0

parent=L2TP_over_IPSEC index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=2 ad=/0
stat: rxp=403 txp=180 rxb=30057 txb=8731
dpd: mode=on-demand on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=L2TP_over_IPSEC proto=17 sa=1 ref=3 serial=1 transport-mode add-route
  src: 17:201.1.1.3-201.1.1.3:1701
  dst: 17:10.1.1.6-10.1.1.6:0
  SA:  ref=3 options=182 type=00 soft=0 mtu=1470 expire=2915/0B replaywin=2048
       seqno=b5 esn=0 replaywin_lastseq=00000193 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3590/3600
  dec: spi=4b06e27a esp=3des key=24 b4765186ea5f3ac392a6009a3e9173907f2b1872715a317a
       ah=sha1 key=20 75f29e67a8a100e5d7a58b84451675d318b36ee5
  enc: spi=ba90d928 esp=3des key=24 7cbbc629743b42ac00b0ab328129364de3b157c67dfd1a39
       ah=sha1 key=20 e8fd28ceaf0d55c30643495a95af9e5b562d2443
  dec:pkts/bytes=806/60114, enc:pkts/bytes=360/23491
  npu_flag=00 npu_rgwy=10.1.1.6 npu_lgwy=201.1.1.3 npu_selid=6 dec_npuid=0 enc_npuid=0
```

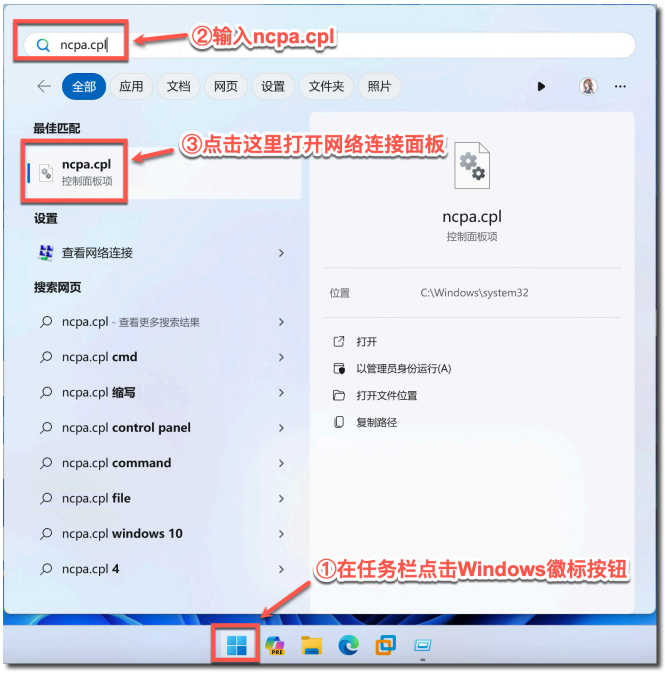
注意事项

如果 FortiGate 上配置的 L2TP 用户为 LDAP 认证用户，则需要确保客户端 L2TP 连接的认证方法为 PAP。

重要

- LDAP 用户身份验证支持 PPTP、L2TP、IPsec VPN 和防火墙身份验证，但由于点对点协议的限制，只支持 PAP（Packet Authentication Protocol）。
- 由于 Windows 本机客户端默认使用 MSCHAP 或 MSCHAPv2，LDAP 认证场景下，您需要将其更改为 PAP。

1. 在 Windows 系统中按照下图方式打开网络连接面板（或使用 Win + R 快捷键，然后运行 ncpa.cpl）。



2. 在弹出的网络连接面板中，右键点击已创建的 L2TP 连接，然后点击属性按钮。



3. 在弹出的 L2TP 连接属性中，选择“安全”面板，然后勾选“允许使用这些协议”，勾选“未加密的密码 (PAP)(H)”，取消勾选 CHAP 和 MS - CHAP v2，然后点击确定按钮。之后即可正常连接 L2TP。



Windows 原生客户端接入 IPSec（IKEv2）

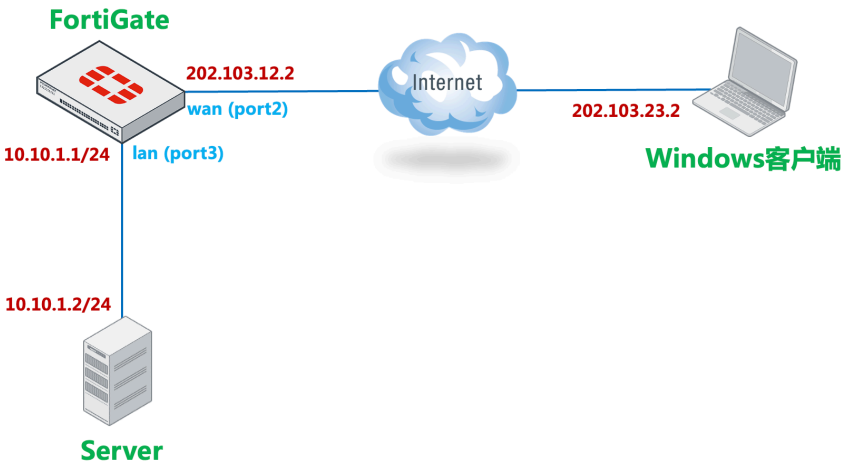
> **重要**
>
> 本文基于 Windows 11 专业版 22H2 编写。

网络需求

通过 Windows 原生客户端（IKEv2）连接 FortiGate 的 IPSec，并访问 VPN 内部资源。

> **重要**
>
> Windows 系统使用 IKEv2，必须配合证书认证（客户端验证服务器的证书，服务器不认证客户端的证书）。

网络拓扑



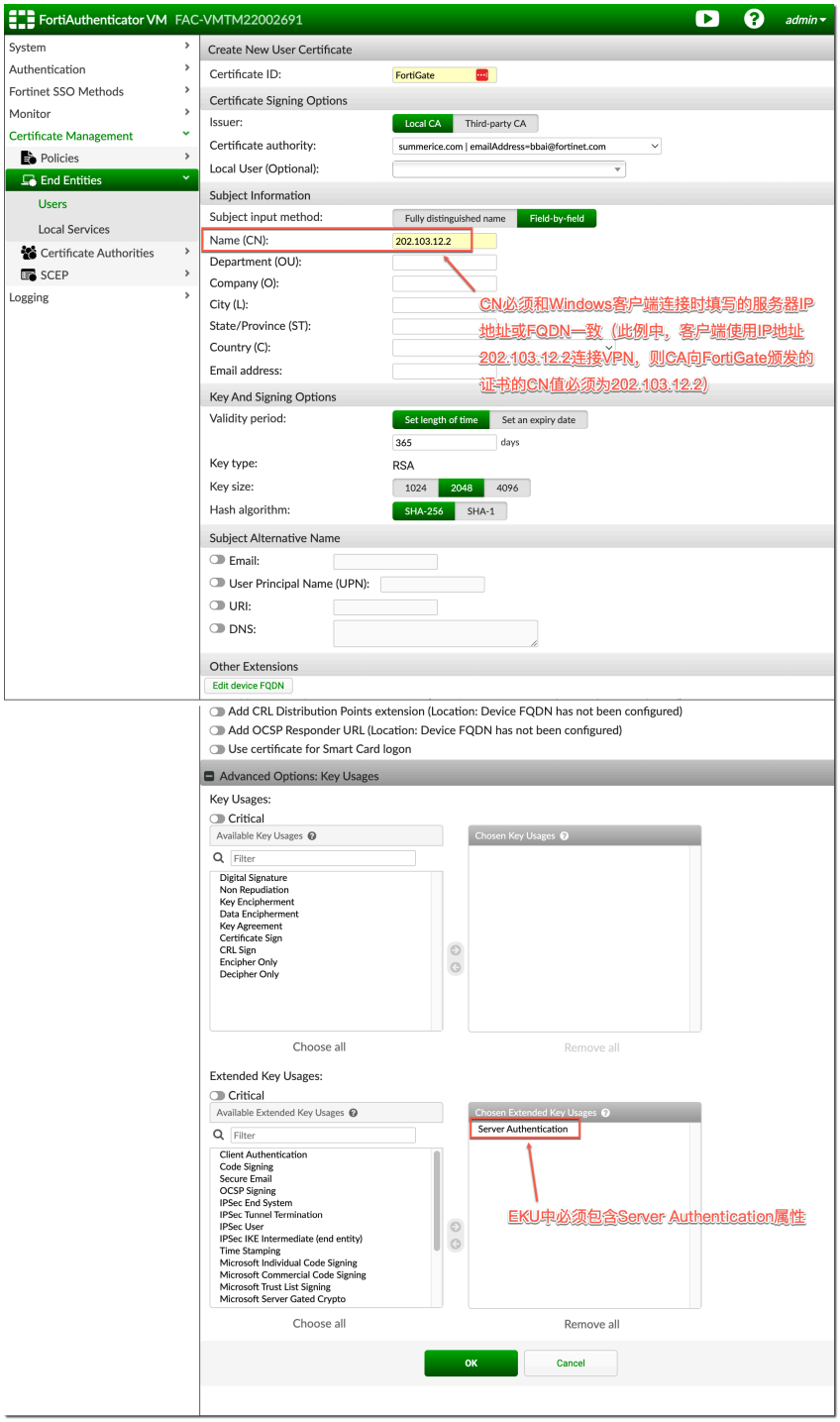
配置步骤

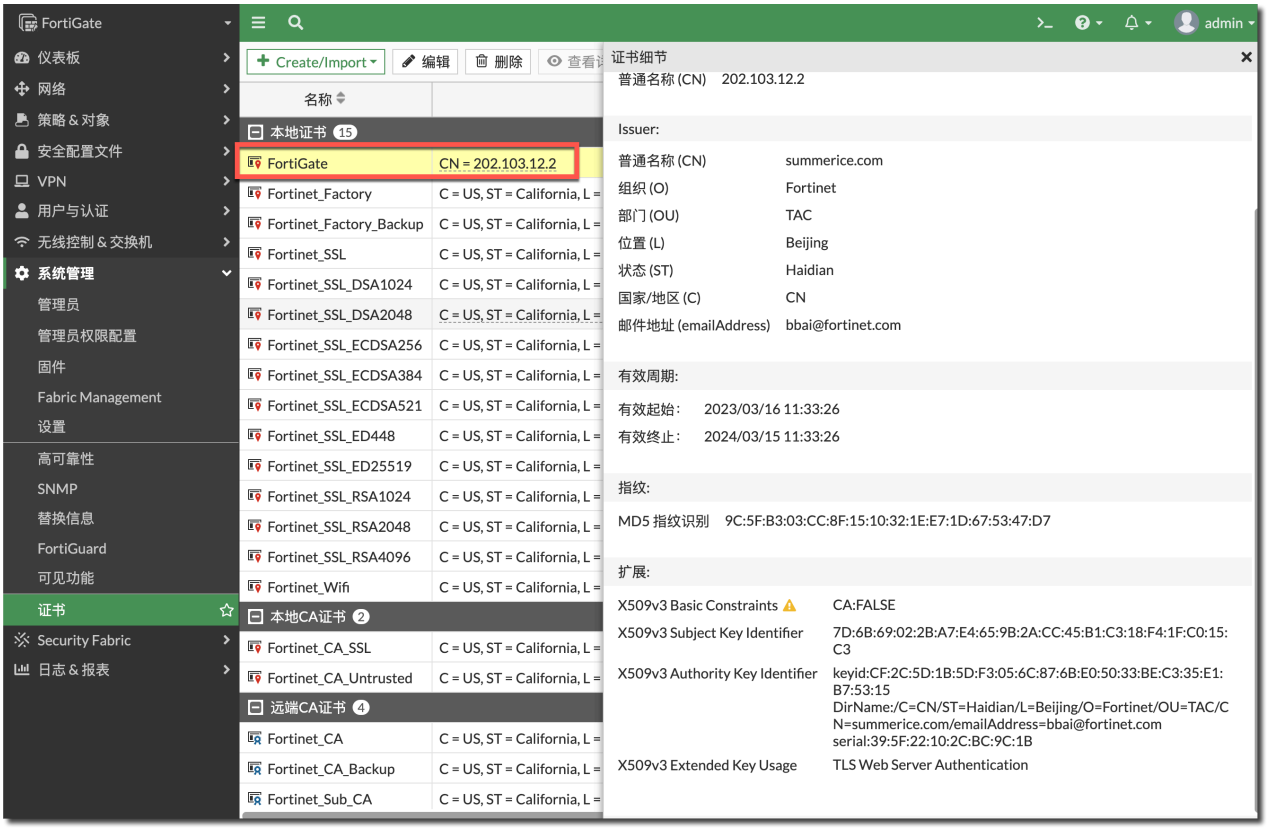
证书颁发

1. 通过证书颁发机构为 FortiGate 颁发证书（这里我们使用 FortiAuthenticator 为例）的注意事项。

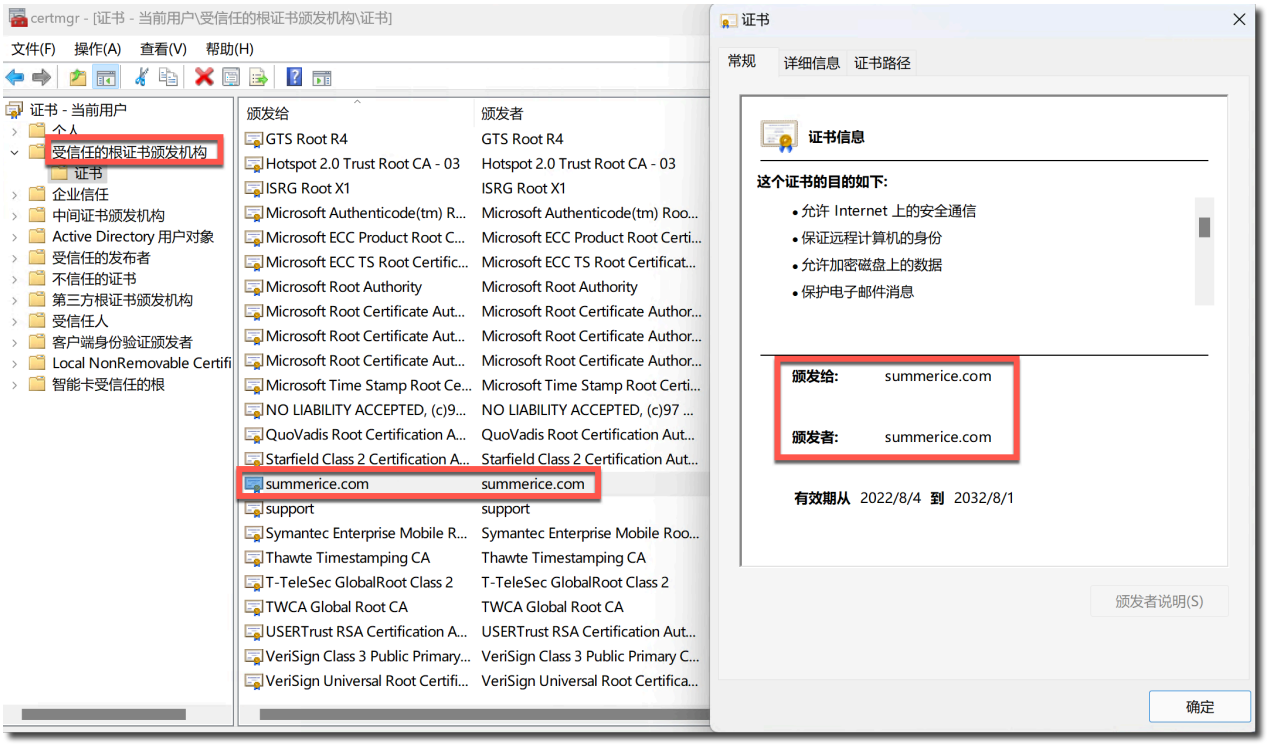
> **重要**
>
> Windows 系统使用 IKEv2，必须配合证书认证（客户端验证服务器的证书，服务器不认证客户端的证书），而且对颁发的证书有严格要求：
>
> - CA 向 FortiGate 颁发的证书的 CN 必须和 Windows 客户端连接时填写的服务器 IP 地址或 FQDN 一致（例如此例中，客户端使用 IP 地址 202.103.12.2 连接 VPN，则 CA 向 FortiGate 颁发的证书的 CN 值必须为 202.103.12.2）。
> - CA 向 FortiGate 颁发的证书的 EKU（Extended Key Usages）必须包含 Server Authentication 属性。
> - 客户端必须导入 CA 证书并加入受信任的根证书颁发机构。

2. 为 FortiGate 颁发证书，并导入 FortiGate（导入过程可参考系统管理 → 系统设置 → 证书管理 ↗章节）。


FortiAuthenticator VM FAC-VMTM22002691
admin
System
Authentication
Fortinet SSO Methods
Monitor
Certificate Management
Policies
End Entities
Users
Local Services
Certificate Authorities
SCEP
Logging
Create New User Certificate
Certificate ID:
FortiGate
Certificate Signing Options
Issuer:
Local CA
Third-party CA
Certificate authority:
summerice.com | emailAddress=bbai@fortinet.com
Local User (Optional):
Subject Information
Subject input method:
Fully distinguished name
Field-by-field
Name (CN):
202.103.12.2
Department (OU):
Company (O):
City (L):
State/Province (ST):
Country (C):
Email address:
CN必须和Windows客户端连接时填写的服务器IP地址或FQDN一致（此例中，客户端使用IP地址202.103.12.2连接VPN，则CA向FortiGate颁发的证书的CN值必须为202.103.12.2）
Key And Signing Options
Validity period:
Set length of time
Set an expiry date
365
days
Key type:
RSA
Key size:
1024
2048
4096
Hash algorithm:
SHA-256
SHA-1
Subject Alternative Name
Email:
User Principal Name (UPN):
URI:
DNS:
Other Extensions
Edit device FQDN
Add CRL Distribution Points extension (Location: Device FQDN has not been configured)
Add OCSP Responder URL (Location: Device FQDN has not been configured)
Use certificate for Smart Card logon
Advanced Options: Key Usages
Key Usages:
Critical
Available Key Usages
Filter
Digital Signature
Non Repudiation
Key Encipherment
Data Encipherment
Key Agreement
Certificate Sign
CRL Sign
Encipher Only
Decipher Only
Chosen Key Usages
Choose all
Remove all
Extended Key Usages:
Critical
Available Extended Key Usages
Filter
Client Authentication
Code Signing
Secure Email
OCSP Signing
IPSec End System
IPSec Tunnel Termination
IPSec User
IPSec IKE Intermediate (end entity)
Time Stamping
Microsoft Individual Code Signing
Microsoft Commercial Code Signing
Microsoft Trust List Signing
Microsoft Server Gated Crypto
Chosen Extended Key Usages
Server Authentication
EKU中必须包含Server Authentication属性
Choose all
Remove all
OK
Cancel

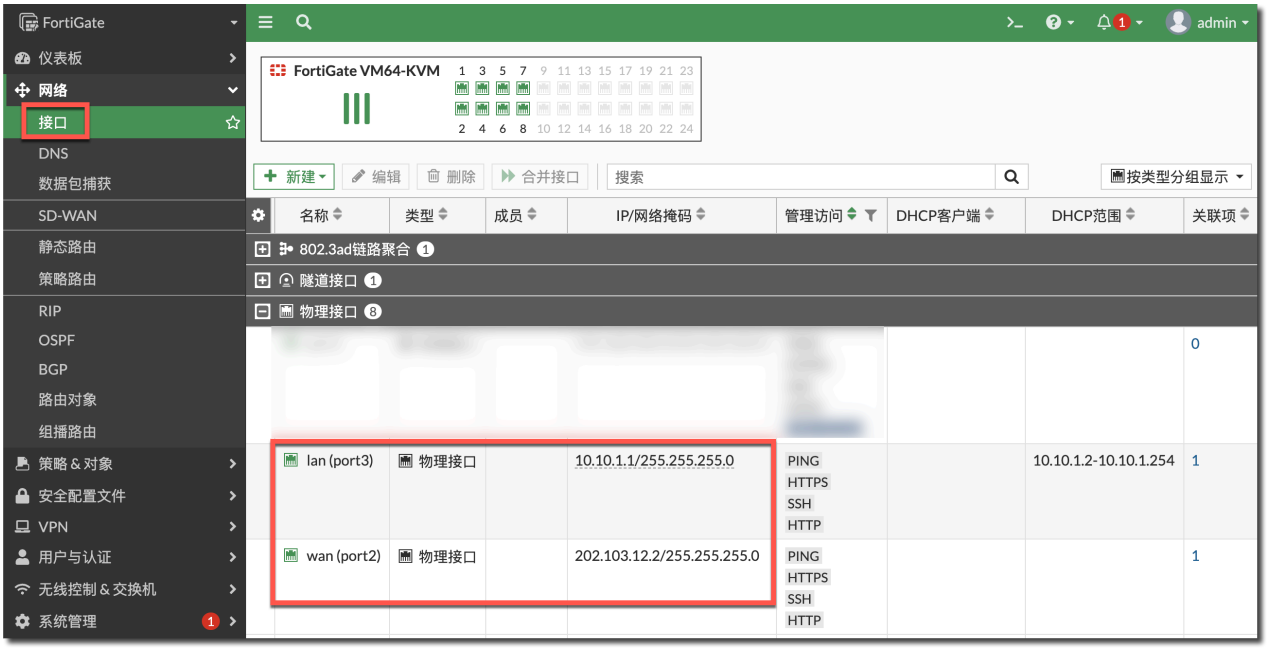


3. 下载 CA 证书，并导入 Windows PC 的受信任根证书颁发机构列表中。



FortiGate 配置

1. 配置 FortiGate 的接口 IP。



2. 创建本地 VPN 用户和用户组，参考用户与认证 → 本地用户和用户组 ↗章节（如需使用远程认证，请参考用户与认证 → LDAP 认证 → LDAP 认证配置与排错 ↗或用户与认证 → Radius 认证 → Radius 认证配置与排错 ↗章节）。




```
config user local
    edit "vpn_user"
        set type password
        set passwd xxxxxxxx
    next
end
```

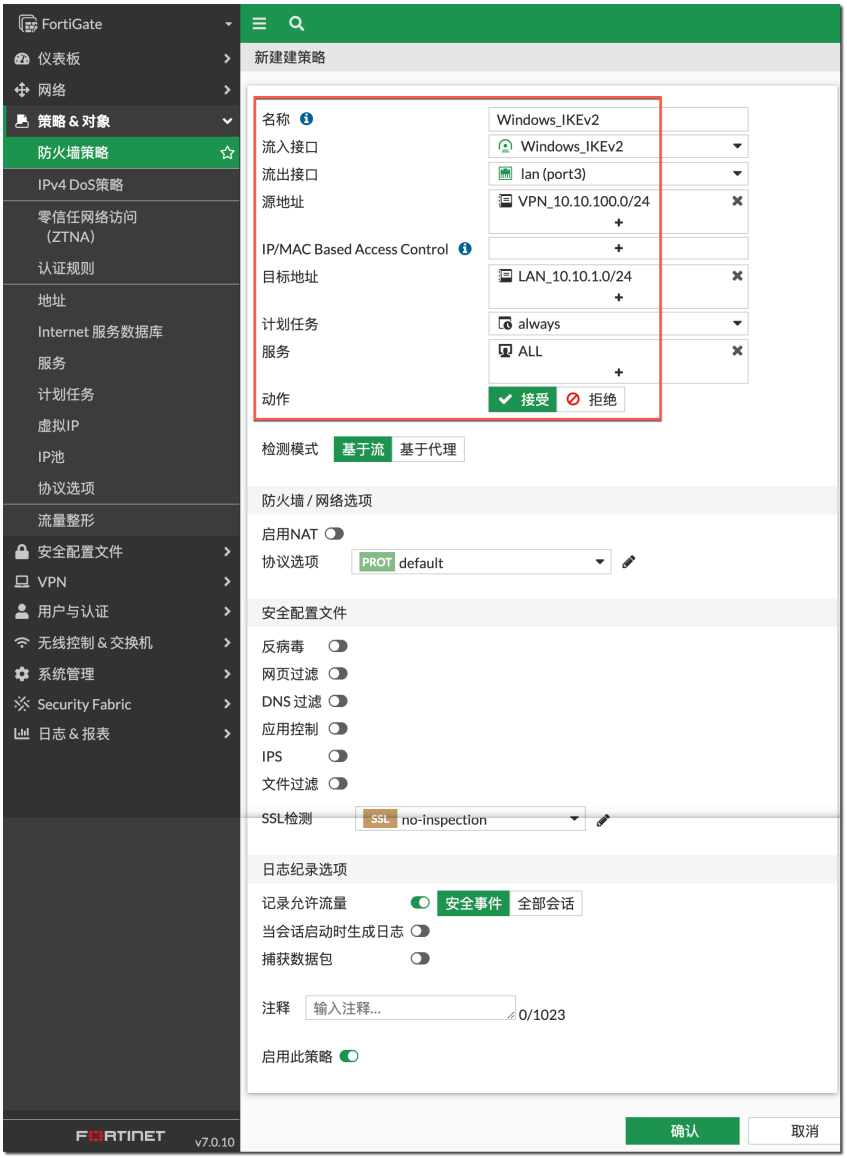



```
config user group
    edit "VPN"
        set member "vpn_user"
    next
end
```

3. 该种方式的 IPSec 需要使用 CLI 方式配置，分别配置 IPSec 一阶段和二阶段。

```
IPSec一阶段配置:
config vpn ipsec phase1-interface
    edit "Windows_IKEv2"
        set type dynamic
        set interface "port2"
        set ike-version 2    //IKE协议2
        set authmethod signature    //认证模式为证书认证
        set peertype any
        set net-device disable
        set mode-cfg enable    //开启为客户端分配IP
        set ipv4-dns-server1 10.10.1.2  //为客户端分配的DNS地址
        set proposal aes128-sha1 aes256-sha256
        set dhgrp 2
        set eap enable    //需要开启EAP做用户密码认证
        set eap-identity send-request    //开启EAP主动发送认证请求
        set authusrgrp "VPN"    //选择认证的用户组
        set certificate "FortiGate"    //选择服务器证书为导入的FortiGate证书
        set ipv4-start-ip 10.10.100.1
        set ipv4-end-ip 10.10.100.100    //客户端分配地址范围
    next
end

IPSec二阶段配置:
config vpn ipsec phase2-interface
    edit "Windows_IKEv2"
        set phase1name "Windows_IKEv2"
        set proposal aes128-sha1 aes256-sha1
        set pfs disable
        set keepalive enable
    next
end
```

4. 配置防火墙策略，允许客户端访问内网 VPN 资源。



VPN资源地址对象：

```
config firewall address
    edit "LAN_10.10.1.0/24"
        set subnet 10.10.1.0 255.255.255.0
    next
end
```

客户端地址对象：

```
config firewall address
    edit "VPN_10.10.100.0/24"
        set subnet 10.10.100.0 255.255.255.0
    next
end
```

防火墙策略：

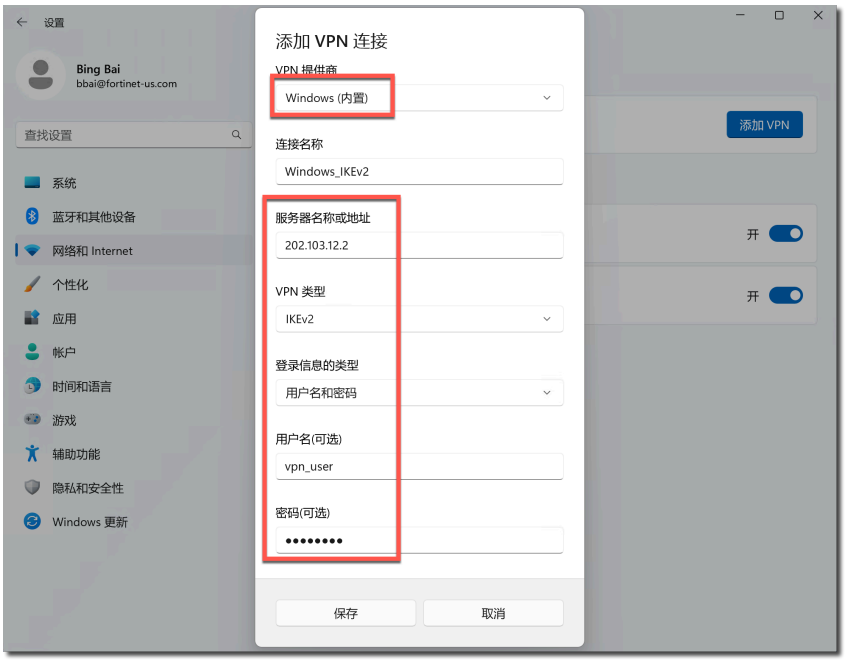
```
config firewall policy
    edit 1
        set name "Windows_IKEv2"
        set srcintf "Windows_IKEv2"
        set dstintf "port3"
        set action accept
        set srcaddr "VPN_10.10.100.0/24"
        set dstaddr "LAN_10.10.1.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

客户端配置

1. 打开设置 → 网络和 Internet → VPN，点击新建按钮。



2. 配置 VPN 提供商为 Windows，配置连接名称和服务器 IP（或 FQDN），“VPN 类型”选择“IKEv2”，“登录信息的类型”选择“用户名和密码”，填写用户名和密码信息，点击保存按钮。



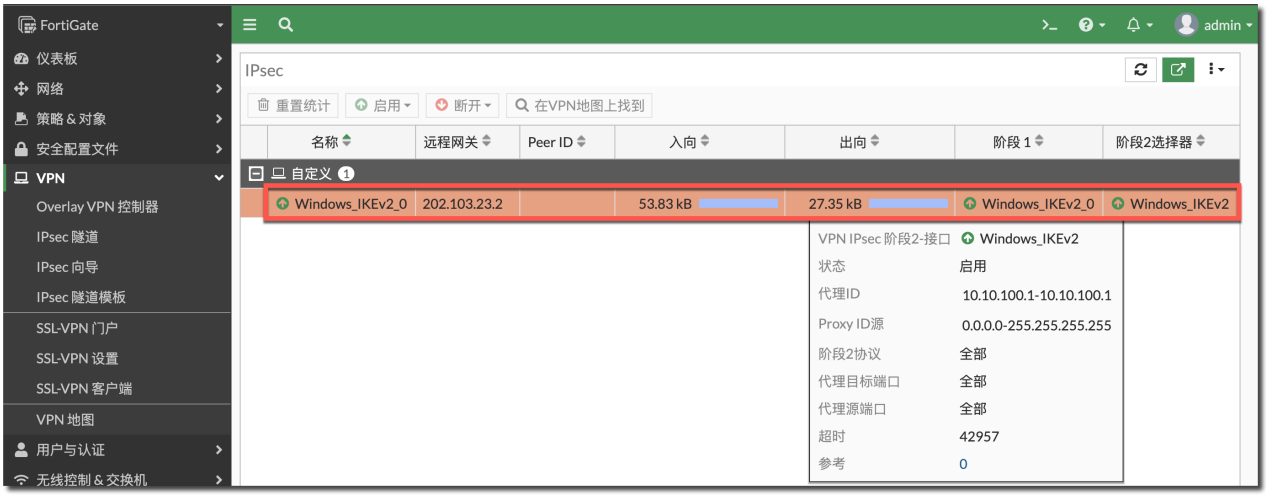
结果验证

1. 在客户端 VPN 点击连接按钮，然后查看连接状态，可以看到 VPN 连接已建立。





2. 在 FortiGate 侧查看连接状态。



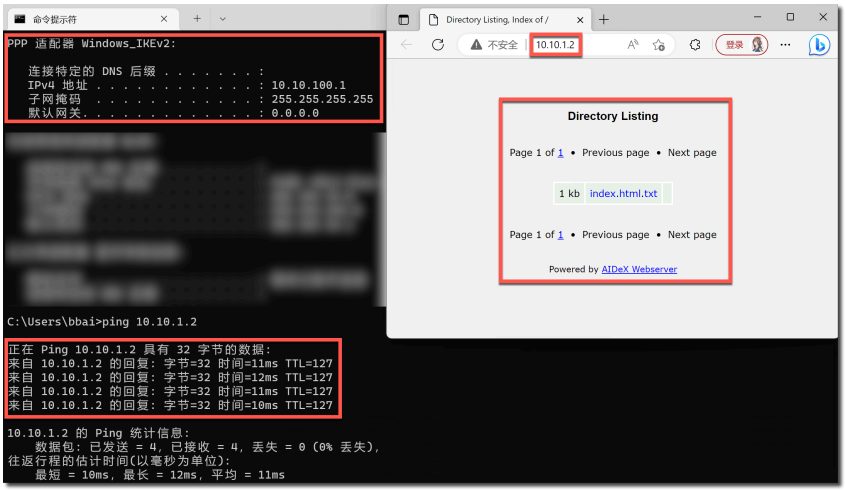

```
FortiGate # diagnose vpn ike gateway list

vd: root/0
name: Windows_IKEv2_0
version: 2
interface: port2 4
addr: 202.103.12.2:4500 → 202.103.23.2:64916
tun_id: 10.10.100.1/::10.0.0.15
```

```
remote_location: 0.0.0.0
network-id: 0
created: 281s ago
eap-user: vpn_user
2FA: no
assigned IPv4 address: 10.10.100.1/0.0.0.0
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 50/50/50 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
  id/spi: 28 1a1ac0eb795737c0/45eb750afe25ce3f
  direction: responder
  status: established 281-281s ago = 50ms
  proposal: aes128-sha1
  child: no
  SK_ei: d4088e142220d3f2-24c73e8bd23cc361
  SK_er: 23dad2bdacf52ae7-ea3e3f1ad62e48cd
  SK_ai: 2db36c765a403317-66950b37f90fc178-3ff9a0a4
  SK_ar: 919125e418fb94dc-d1978545c2da767a-e5f32bb9
  PPK: no
  message-id sent/recv: 0/6
  lifetime/rekey: 86400/85848
  DPD sent/recv: 00000000/00000000

FortiGate # diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=Windows_IKEv2_0 ver=2 serial=10 202.103.12.2:4500→202.103.23.2:64916 tun_id=10.10.100.1 tun_id6=::10.0.0.15
dst_mtu=1500 dpd-link=on weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/9128 options[23a8]=npu rgwy-chg rport-chg frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=Windows_IKEv2 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=813 txp=280 rxb=61158 txb=30402
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=silent draft=0 interval=10 remote_port=64916
proxyid=Windows_IKEv2 proto=0 sa=1 ref=74 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.100.1-10.10.100.1:0
  SA:  ref=3 options=20483 type=00 soft=0 mtu=1422 expire=42903/0B replaywin=2048
       seqno=119 esn=0 replaywin_lastseq=0000032d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43188/43200
  dec: spi=24832a57 esp=aes key=16 7b7328d6fe3ee6230a5072fea98c57d0
       ah=sha1 key=20 1fa4bee8e25fd477e8e79ea28291695d84c4d783
  enc: spi=60f3ab06 esp=aes key=16 aaaf1aca419bf8dd8743f2c267f97b70
       ah=sha1 key=20 50b086fddc0810e3dc5f204c8889f236e18d5663
  dec:pkts/bytes=1626/122316, enc:pkts/bytes=560/81074
  npu_flag=00 npu_rgwy=202.103.23.2 npu_lgwy=202.103.12.2 npu_selid=f dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=Windows_IKEv2 ver=2 serial=2 202.103.12.2:0→0.0.0.0:0 tun_id=10.0.0.2 tun_id6=::10.0.0.2 dst_mtu=0 dpd-link=on
weight=1
bound_if=4 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc  role=primary accept_traffic=1
overlay_id=0
proxyid_num=0 child_num=1 refcnt=3 ilast=42973370 olast=42973370 ad=/0
stat: rxp=3292 txp=280 rxb=241855 txb=30402
dpd: mode=on-idle on=0 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
```

3. 在 FortiGate 侧查看路由 kernel 表，可以看到 FortiGate 已经自动添加了到客户端的路由。

```
FortiGate # get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0→10.10.100.1/32 pref=0.0.0.0 gwy=10.10.100.1
dev=20(Windows_IKEv2)
...
```

4. 使用客户端访问 VPN 内网资源（Ping 和 HTTP 访问）。



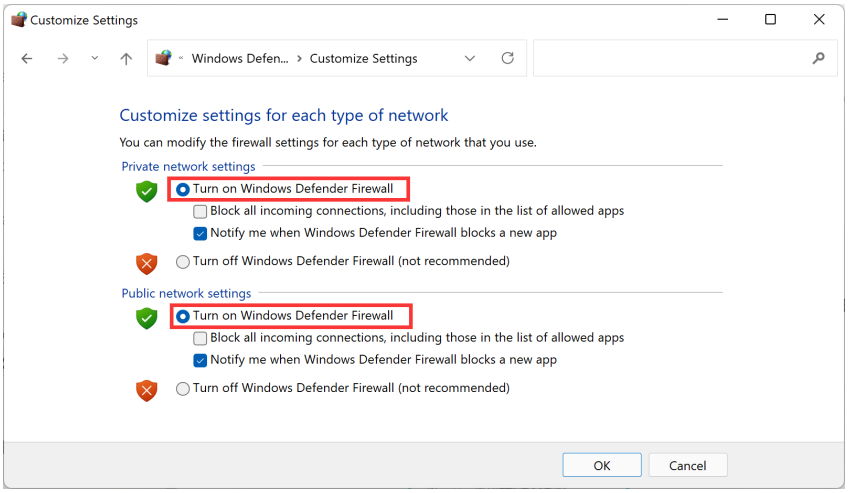
重要

实际测试 Windows 使用 IKEv2，无法使用隧道分割功能，VPN 建立后，所有流量被送入 VPN 隧道。

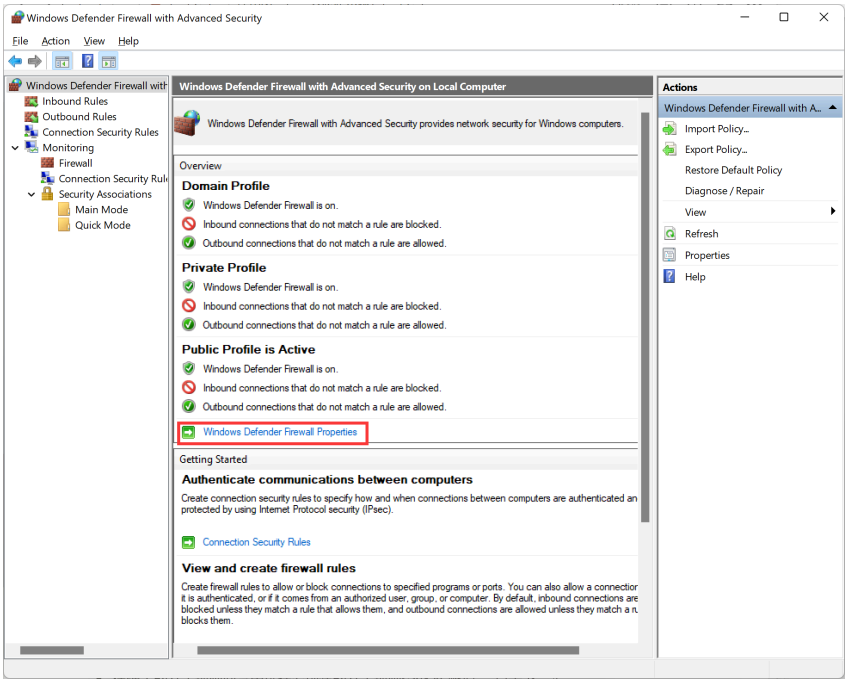
使用 Windows 系统防火墙与 FortiGate 建立 IPSec VPN

Windows Firewall IPSec 配置步骤

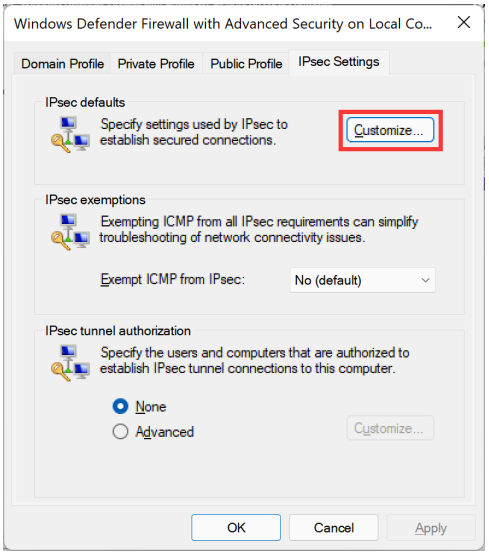
1. 在使用 Windows 防火墙的 IPSec 功能前，需要将 Windows 防火墙功能打开。点击键盘 Win（徽标）键，输入“Windows Defender Firewall”，点击左侧的“Turn Windows Defender Firewall on or off”，将需要使用 VPN 的网络的防火墙功能打开。



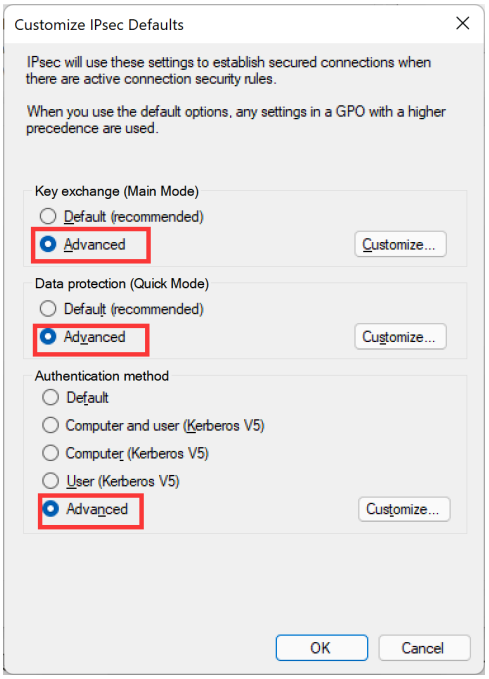
2. 配置 Windows Firewall IPSec 的自定义安全提议。上图中点击返回按钮，然后点击左侧的“Advanced Settings”，点击“Windows Defender Firewall Properties”。



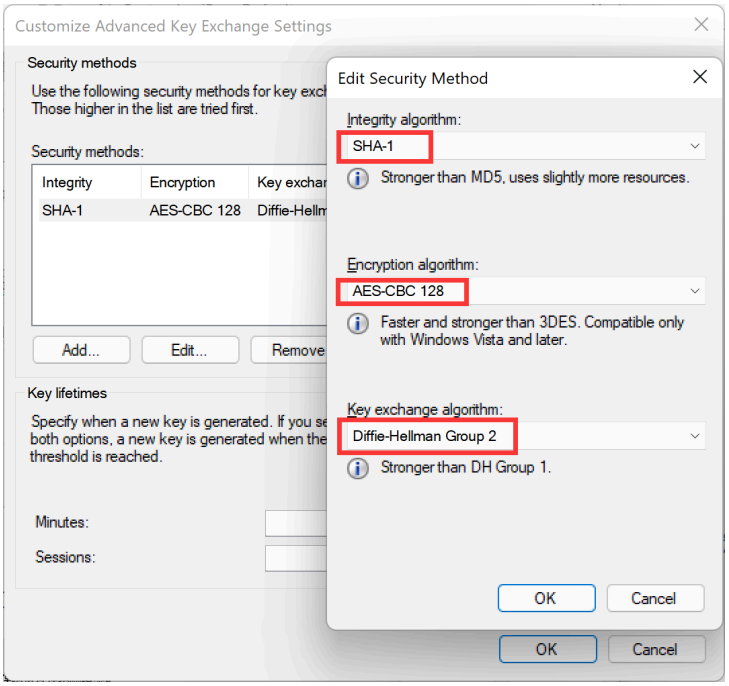
3. 在弹出的属性界面中点击“IPSec Settings”选项卡，在“IPSec defaults”中点击“Customize...”按钮。



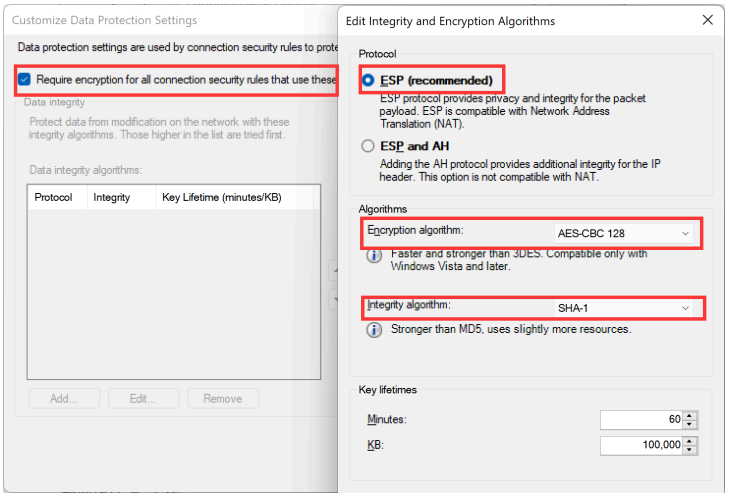
4. 配置“Key exchange (Main Mode)”、“Data protection (Quick Mode)”、“Authentication method”均为“Advanced”。



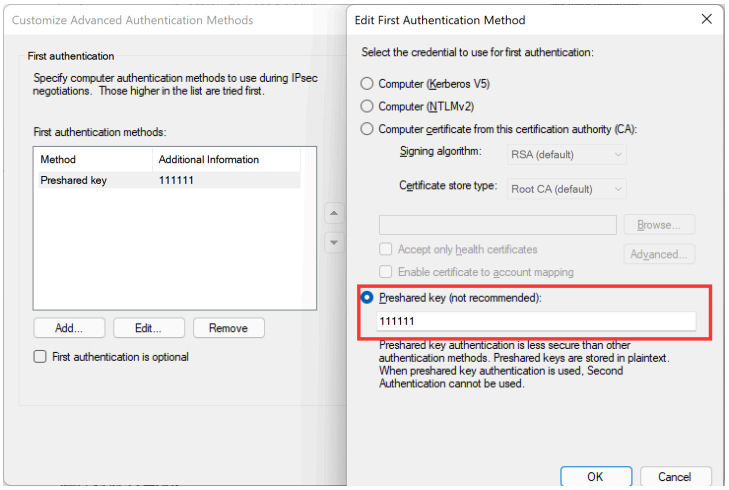
5. 配置第一阶段自定义安全提议。点击“Key exchange (Main Mode)”中的“Customize...”按钮。配置完整性算法、加密算法和 DH 组。



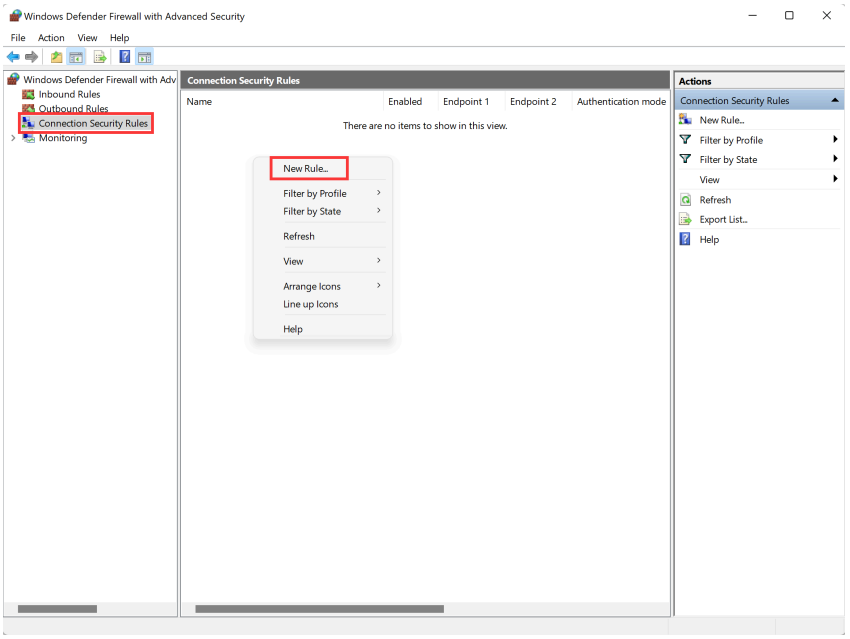
6. 配置第二阶段自定义安全提议。点击“Data protection (Quick Mode)”中的“Customize...”按钮。开启“Require encryption for all connection security rules that use these settings”，配置完整性算法、加密算法。



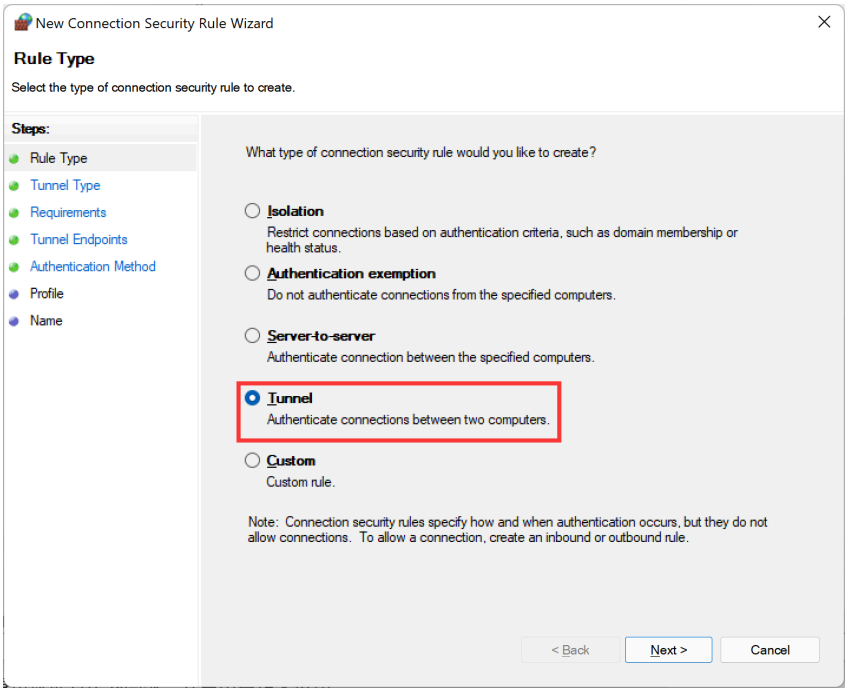
7. 配置自定义的认证方式。点击“Authentication method”中的“Customize...”按钮，按需求配置认证方法，这里以“预共享密钥”为例。



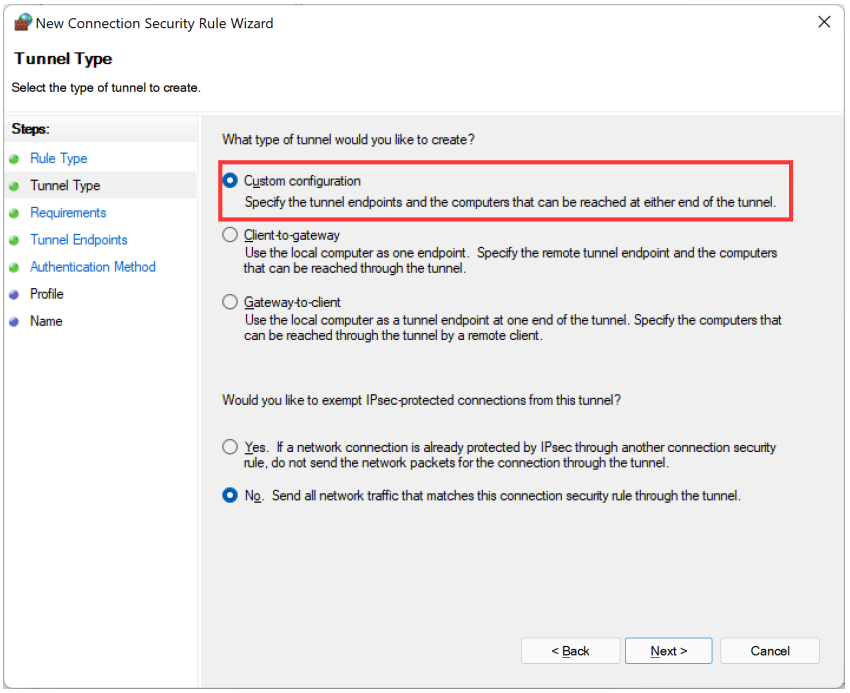
8. 在 Windows 高级防火墙中创建 IPSec 连接。进入 Windows Defender Firewall with Advanced Security → Connection Security Rules，右键点击后选择“New Rules”。



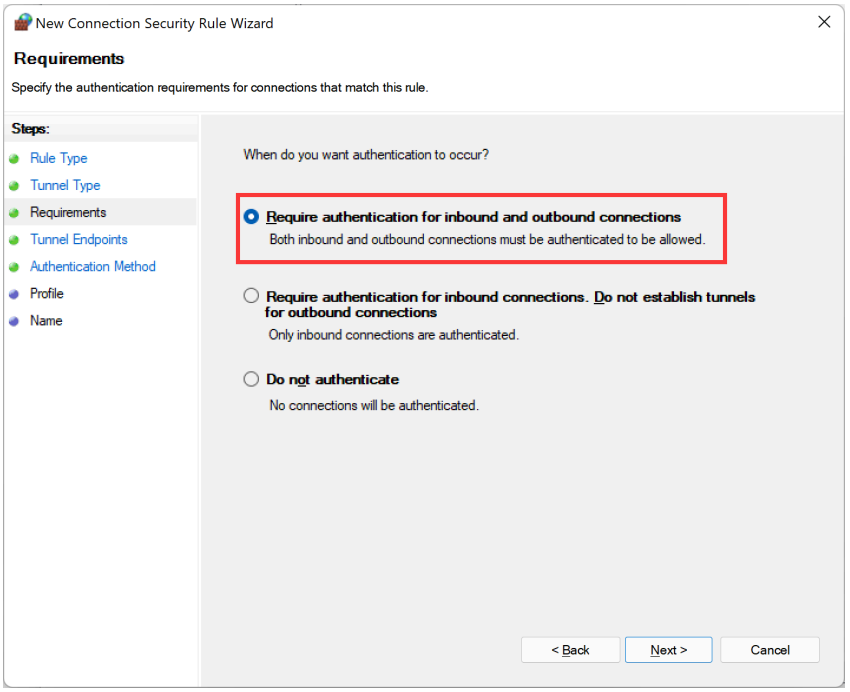
9. 在弹出的安全规则向导中选择要创建的安全规则类型为“Tunnel”，点击下一步。



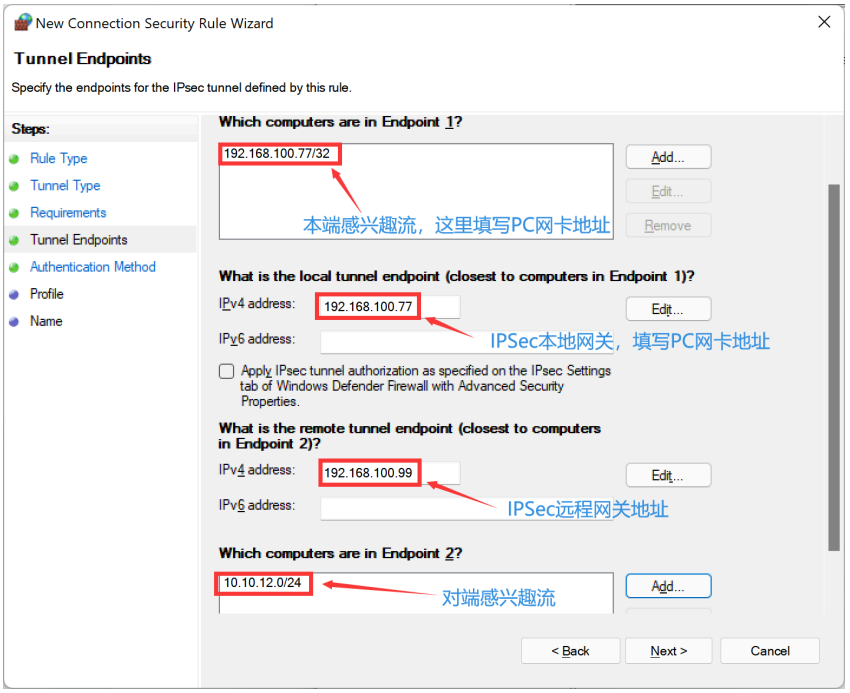
.0. 隧道类型选择“Custom configuration”，其余选项默认，点击下一步。



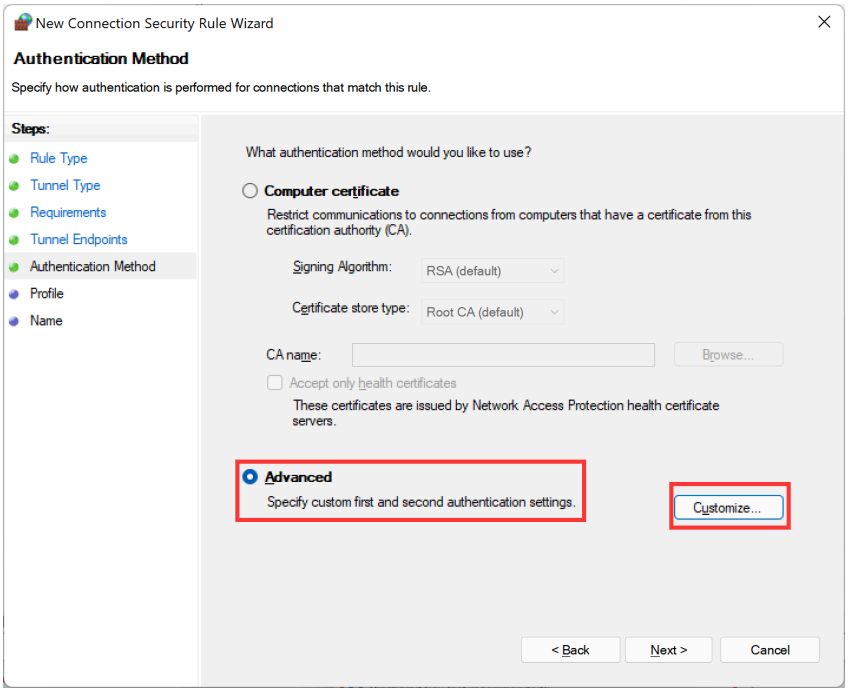
1. 身份验证时间选择“Require authentication for inbound and outbound connections”，点击下一步。



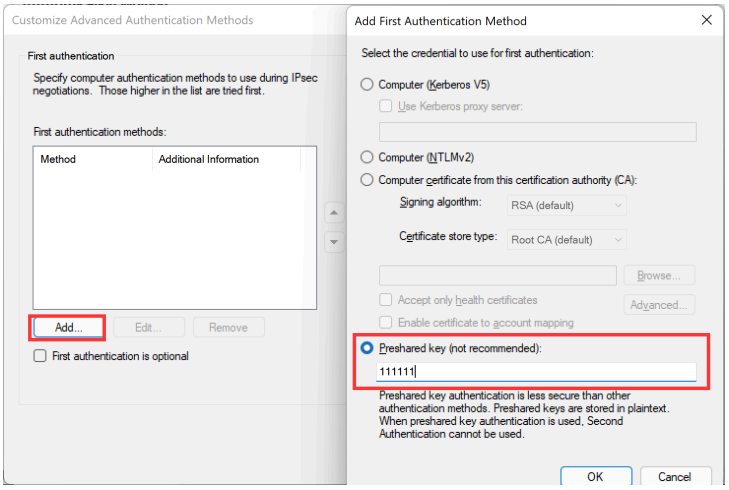
2. 配置终结点 1 中的计算机，这里填写 Windows 本端的保护网段，一般情况下为 PC 的通信地址；配置终结点 2 中的计算机，这里填写对端的保护网段，注意只有 PC 发起了向该保护网段的流量，协商才能被触发；配置本地隧道和远程隧道终结点，这里填写 Windows 本端和对端 IPSec 服务器的协商 IP 地址。注意只有 PC 发起了向该保护网段的流量，协商才能被触发。点击下一步。



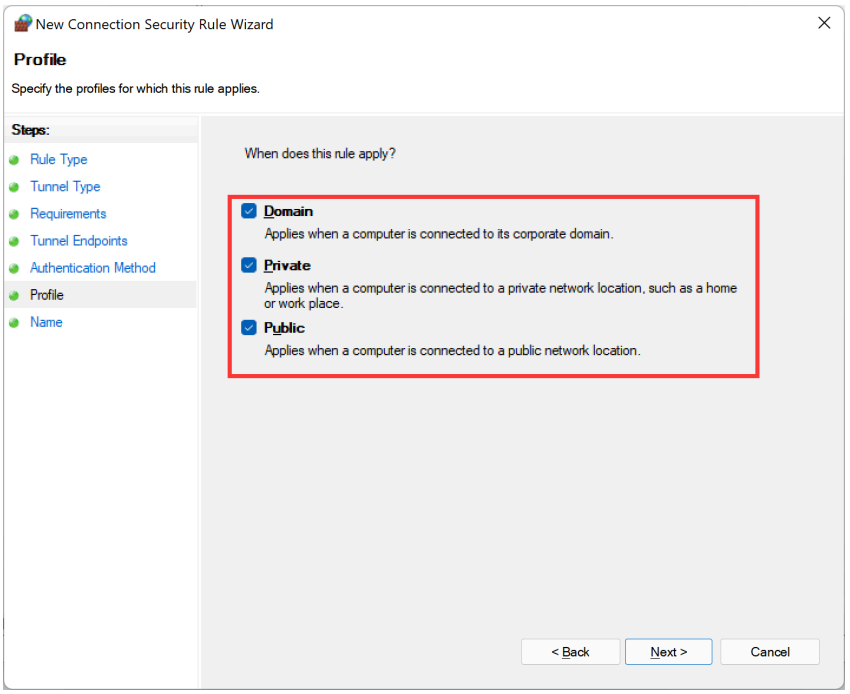
3. 选择身份认证方法为“Advanced”，并点击“Customize...”按钮。



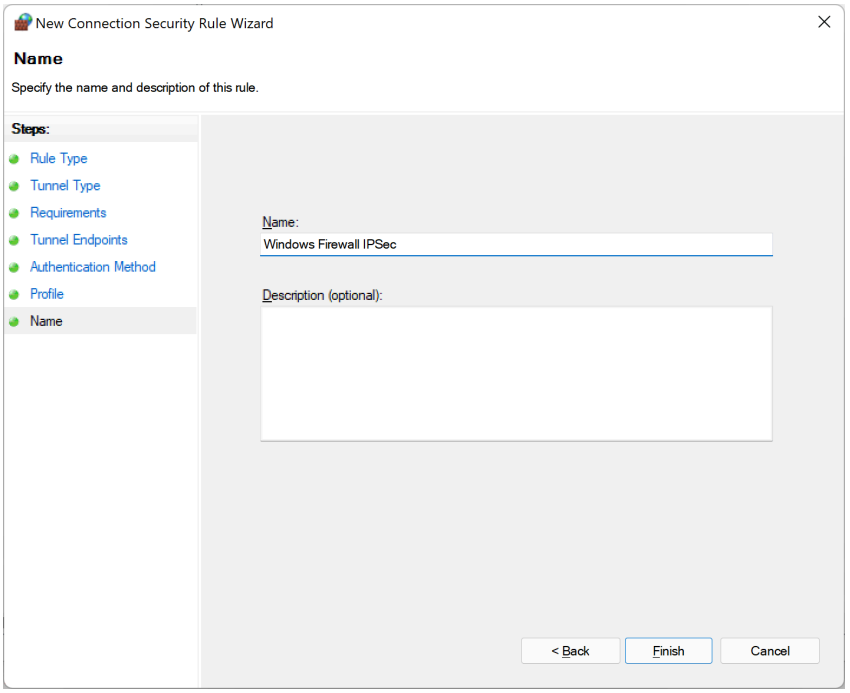
4. 按需求配置认证算法，这里以预共享密钥为例。保存后点击下一步。



5. “When does this rule apply”选择默认即可。点击下一步。



6. 为 IPSec 连接配置名称，完成。



7. 补充：高级防火墙 IPSec 配置的特点是每条连接只用简单的配置本、对端的公网 IP 以及保护网段，安全提议是一个全局配置，所有连接都会受此安全提议影响。这样非常简单快速但也缺少很多灵活性，比如需要这些连接的安全提议不一致时就无法满足。其实高级防火墙是做了很多东西的。在后台的命令行，使用 netsh advfirewall 命令可以对高级防火墙并进行灵活配置，以下是一些简单的配置实例。

```bash
- cmd>netsh advfirewall（进入高级防火墙配置）
- netsh advfirewall conse（更改防火墙配置）
- netsh advfirewall conse [add|delete|set] rule name=连接名 new option=value（增、删、改一条连接。具体选项和可用参数可以在[add|delete|set] rule help中可以查看，内容非常详细的）
- netsh advfirewall consec>set rule name=v4_ipsec new qmsecmethods（第二阶段的安全提议，如果需要配置PFS必须先配置安全提议）=esp:md5-3des+600000min+100000kb qmpfs=dhgroup2（开启PFS，并且使用DH2，当ESP加密算法为空时开启PFS不生效）
```

FortiGate 配置

IPSec 配置

```bash
config vpn ipsec phase1-interface
    edit "WindowsFw"
        set type dynamic
        set interface "lan"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
        set dhgrp 2
        set psksecret xxxxxxxx
        set dpd-retryinterval 60
    next
end
```

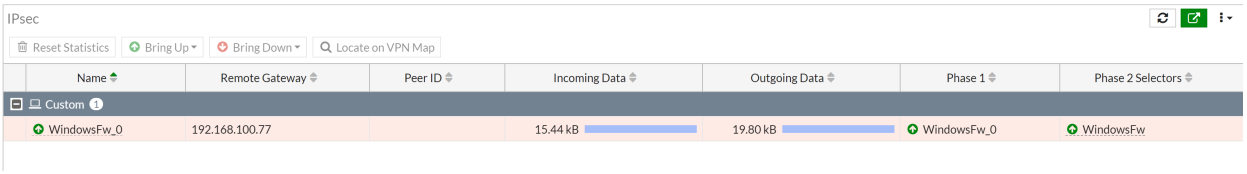
```bash
config vpn ipsec phase2-interface
    edit "WindowsFw"
        set phase1name "WindowsFw"
        set proposal aes128-sha1
        set pfs disable
        set src-subnet 10.10.12.0 255.255.255.0
        set dst-subnet 192.168.100.77 255.255.255.255
    next
end
```

防火墙策略配置

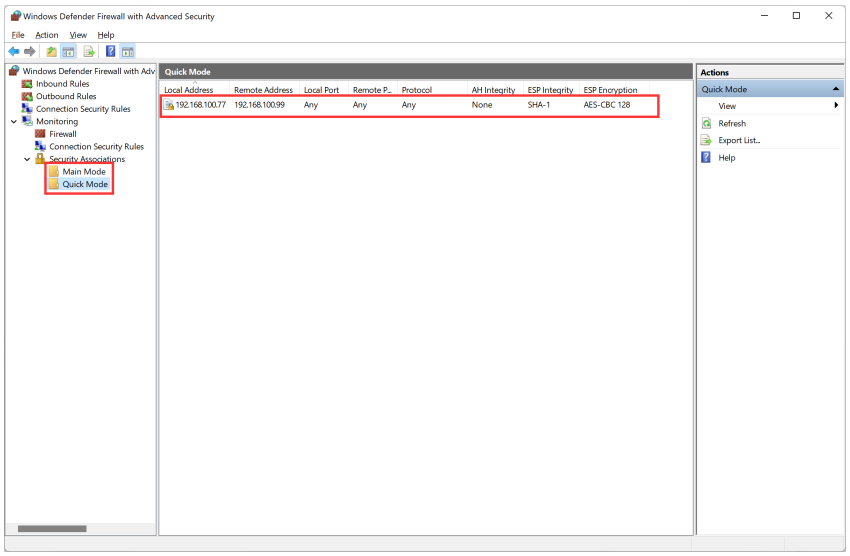
```bash
VPN资源地址对象:
config firewall address
    edit "Local_10.10.12.0/24"
        set subnet 10.10.12.0 255.255.255.0
    next
end

客户端地址对象:
config firewall address
    edit "Remote_192.168.100.77/32"
        set subnet 192.168.100.77 255.255.255.255
    next
end

config firewall policy
    edit 1
        set name "WindowsFw"
        set srcintf "WindowsFw"
        set dstintf "port3"
        set action accept
        set srcaddr "Remote_192.168.100.77/32"
        set dstaddr "Local_10.10.12.0/24"
        set schedule "always"
        set service "ALL"
    next
end
```

结果验证

1. 使用 PC ping FortiGate 端的感兴趣流，触发 IPSec 协商，查看 IPSec 连接建立状态：



2. Windows Firewall 连接监控：



3. 客户端抓包：

No.	Time	Source	Destination	Protocol	Length	Info
84…	1921.031377	192.168.100.77	192.168.100.99	ISAKMP	298	Identity Protection (Main Mode)
84…	1921.032873	192.168.100.99	192.168.100.77	ISAKMP	234	Identity Protection (Main Mode)
84…	1921.033321	192.168.100.77	192.168.100.99	ISAKMP	302	Identity Protection (Main Mode)
84…	1921.035152	192.168.100.99	192.168.100.77	ISAKMP	270	Identity Protection (Main Mode)
84…	1921.035496	192.168.100.77	192.168.100.99	ISAKMP	118	Identity Protection (Main Mode)
84…	1921.036783	192.168.100.99	192.168.100.77	ISAKMP	118	Identity Protection (Main Mode)
84…	1921.037072	192.168.100.77	192.168.100.99	ISAKMP	246	Quick Mode
84…	1921.039256	192.168.100.99	192.168.100.77	ISAKMP	214	Quick Mode
84…	1921.039531	192.168.100.77	192.168.100.99	ISAKMP	102	Quick Mode
84…	1921.039558	192.168.100.77	192.168.100.99	ESP	134	ESP (SPI=0x2b5bf512)
84…	1921.039573	192.168.100.77	192.168.100.99	ESP	134	ESP (SPI=0x2b5bf512)
84…	1921.041155	192.168.100.99	192.168.100.77	ESP	134	ESP (SPI=0x440a2fac)
84…	1921.041242	192.168.100.77	192.168.100.99	ESP	118	ESP (SPI=0x2b5bf512)
84…	1921.041251	192.168.100.99	192.168.100.77	ESP	134	ESP (SPI=0x440a2fac)
84…	1921.041283	192.168.100.77	192.168.100.99	ESP	118	ESP (SPI=0x2b5bf512)
84…	1921.041409	192.168.100.77	192.168.100.99	ESP	630	ESP (SPI=0x2b5bf512)
84…	1921.041505	192.168.100.77	192.168.100.99	ESP	630	ESP (SPI=0x2b5bf512)

L2TP VPN

L2TP VPN 简介

L2TP（Layer 2 Tunneling Protocol） VPN 是一种用于承载 PPP 报文的隧道技术，基于 UDP，端口 1701。该技术主要应用在远程办公场景中为出差员工远程访问企业内网资源提供接入服务。

> **重要**
>
> 针对 FortiGate，L2TP 相比较 SSL VPN/IPSec VPN，更推荐用户使用 SSL VPN/IPSec VPN，一方面是安全性更高（数据加密），另一方面是性能上 SSL VPN/IPSec VPN 比 L2TP 强得多（SSL VPN/IPSec VPN 可以被 NP 芯片加速，而 L2TP 只能由 CPU 处理）。而且 IPSec VPN 也支持各种系统原生客户端接入。

L2TP 数据报文

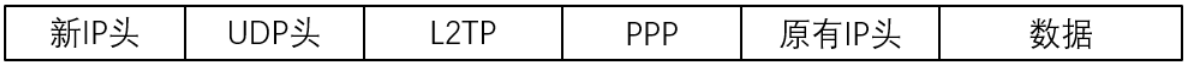
L2TP 数据报文分为两种：控制报文和数据报文；

控制报文：控制消息用于隧道和会话连接的建立、维护以及传输控制；位于隧道和会话建立过程中。控制消息的传输是可靠传输，并且支持对控制消息的流量控制和拥塞控制；主要的控制消息包括控制报文、会话报文等。




```
No.  Time    Source       Destination  Protocol  Length  Source Port  Destinationo Port  Info
  7  0.001…  10.1.1.5     201.1.1.3    L2TP          90         1701              1701  Control Message - ICCN (tunnel id=1, session id=1)

> Frame 7: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{0FB48074-675E-4FD5-8D41-A0C4ACC8F9F7}, id 0
> Ethernet II, Src: VMware_a2:90:ea (00:0c:29:a2:90:ea), Dst: VMware_2e:3f:cf (00:0c:29:2e:3f:cf)
> Internet Protocol Version 4, Src: 10.1.1.5, Dst: 201.1.1.3
> User Datagram Protocol, Src Port: 1701, Dst Port: 1701
v Layer 2 Tunneling Protocol
  > Packet Type: Control Message Tunnel Id=1 Session Id=1
    Length: 48
    Tunnel ID: 1
    Session ID: 1
    Ns: 3
    Nr: 2
  > Control Message AVP
  > Connect Speed AVP
  > Framing Type AVP
  > Proxy Authen Type AVP
```

数据报文：用于承载用户的 PPP 连接数据报文，并在隧道上进行传输。数据消息的传输是不可靠传输，若数据报文丢失，不予重传。不支持对数据消息的流量控制和拥塞控制。




```
No.  Time    Source        Destination   Protocol  Length  Source Port  Destinationo Port  Info
196  4.799…  172.31.1.11   192.168.2.10  ICMP         111         1701              1701  Echo (ping) request  id=0x0001, seq=69/17664, ttl=128

> Frame 196: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface \Device\NPF_{0FB48074-675E-4FD5-8D41-A0C4ACC8F9F7}, id 0
> Ethernet II, Src: VMware_a2:90:ea (00:0c:29:a2:90:ea), Dst: VMware_2e:3f:cf (00:0c:29:2e:3f:cf)
> Internet Protocol Version 4, Src: 10.1.1.5, Dst: 201.1.1.3
v User Datagram Protocol, Src Port: 1701, Dst Port: 1701
    Source Port: 1701
    Destination Port: 1701
    Length: 77
    Checksum: 0xbbb6 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (69 bytes)
v Layer 2 Tunneling Protocol
  > Packet Type: Data    Message Tunnel Id=1 Session Id=1
    Length: 69
    Tunnel ID: 1
    Session ID: 1
v Point-to-Point Protocol
    Protocol: Internet Protocol version 4 (0x0021)
> Internet Protocol Version 4, Src: 172.31.1.11, Dst: 192.168.2.10
> Internet Control Message Protocol
```

L2TP 协议交互过程

1. 建立 L2TP 隧道

   SCCRQ（Start-Control-Connection-Request）：终端向 FortiGate 发起控制连接请求；

   SCCRP（Start-Control-Connection-Reply）：FortiGate 接受了终端的连接请求，隧道的建立过程可以继续；

   SCCCN（Start-Control-Connection-Connected）：对 SCCRP 的回应，完成隧道的建立。

2. 建立 L2TP 会话

   ICRQ（Incoming-Call-Request）：终端请求在已经建立的隧道中建立会话；

   ICRP（Incoming-Call-Reply）：FortiGate 回应 ICRQ，表示 ICRQ 成功，也会在 ICRP 中标识 L2TP 会话必要的参数；

   ICCN（Incoming-Call-Connected）：回应 ICRP，L2TP 会话建立完成。

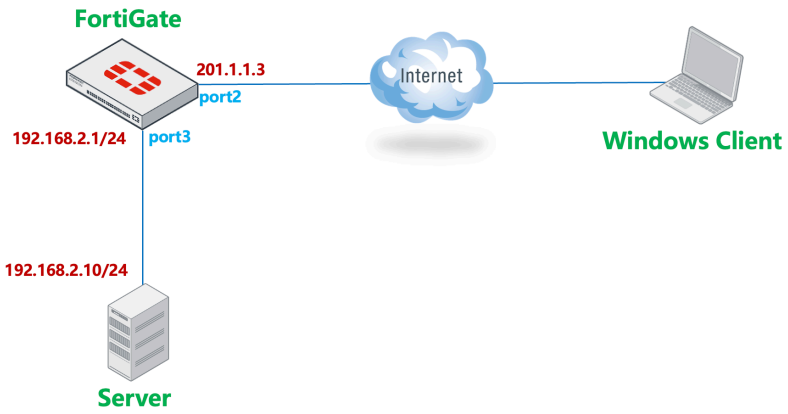
3. 对用户认证，认证通过，建立 PPP 连接

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
1	0.000…	10.1.1.5	201.1.1.3	L2TP	150	1701	1701	Control Message - SCCRQ (tunnel id=0, session id=0)
2	0.000…	201.1.1.3	10.1.1.5	L2TP	121	1701	1701	Control Message - SCCRP (tunnel id=8, session id=0)
3	0.000…	10.1.1.5	201.1.1.3	L2TP	62	1701	1701	Control Message - SCCCN (tunnel id=1, session id=0)
4	0.000…	10.1.1.5	201.1.1.3	L2TP	90	1701	1701	Control Message - ICRQ (tunnel id=1, session id=0)
5	0.000…	201.1.1.3	10.1.1.5	L2TP	60	1701	1701	Control Message - ZLB (tunnel id=8, session id=0)
6	0.001…	201.1.1.3	10.1.1.5	L2TP	70	1701	1701	Control Message - ICRP (tunnel id=8, session id=1)
7	0.001…	10.1.1.5	201.1.1.3	L2TP	90	1701	1701	Control Message - ICCN (tunnel id=1, session id=1)
8	0.001…	201.1.1.3	10.1.1.5	L2TP	60	1701	1701	Control Message - ZLB (tunnel id=8, session id=1)
9	0.025…	10.1.1.5	201.1.1.3	PPP LCP	75	1701	1701	Configuration Request
10	0.026…	201.1.1.3	10.1.1.5	PPP LCP	81	1701	1701	Configuration Request
11	0.026…	201.1.1.3	10.1.1.5	PPP LCP	60	1701	1701	Configuration Reject
12	0.026…	10.1.1.5	201.1.1.3	PPP LCP	83	1701	1701	Configuration Ack
13	0.026…	10.1.1.5	201.1.1.3	PPP LCP	72	1701	1701	Configuration Request
14	0.026…	201.1.1.3	10.1.1.5	PPP LCP	70	1701	1701	Configuration Ack
15	0.026…	201.1.1.3	10.1.1.5	PPP LCP	60	1701	1701	Echo Request
16	0.026…	201.1.1.3	10.1.1.5	PPP CHAP	89	1701	1701	Challenge (NAME='FGVM08[redacted]', VALUE=0xc749f167d96cb0f3e4ee9486389fb3ee)
17	0.027…	10.1.1.5	201.1.1.3	PPP LCP	70	1701	1701	Identification
18	0.027…	10.1.1.5	201.1.1.3	PPP LCP	83	1701	1701	Identification
19	0.027…	10.1.1.5	201.1.1.3	PPP LCP	76	1701	1701	Identification
20	0.027…	201.1.1.3	10.1.1.5	PPP LCP	74	1701	1701	Code Reject
21	0.027…	201.1.1.3	10.1.1.5	PPP LCP	87	1701	1701	Code Reject
22	0.027…	201.1.1.3	10.1.1.5	PPP LCP	80	1701	1701	Code Reject
23	0.028…	10.1.1.5	201.1.1.3	PPP LCP	60	1701	1701	Echo Reply
24	0.028…	10.1.1.5	201.1.1.3	PPP CHAP	111	1701	1701	Response (NAME='user1', VALUE=0xbf896f17a3dde3d72967a5a5ce1161da000000000000000000b8108a806e593f9977b2ee4d…)
25	0.029…	201.1.1.3	10.1.1.5	PPP CHAP	98	1701	1701	Success (MESSAGE='S=F5DE5686706850292F4C41C76EABCA14DF7D71C6')

网络需求

在外移动办公的工作人员需要通过 L2TP VPN 隧道模式拨入到公司内网来对内网主机进行访问。

网络拓扑

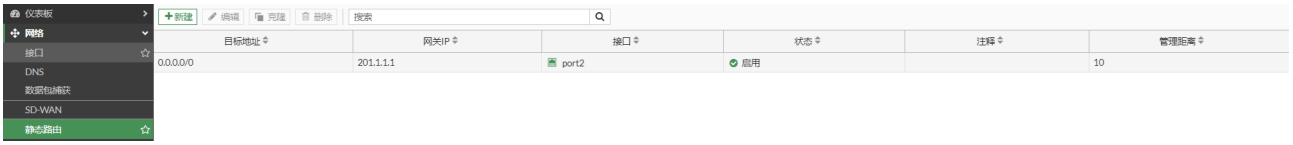


配置步骤

配置 FortiGate L2TP VPN

1.基础配置

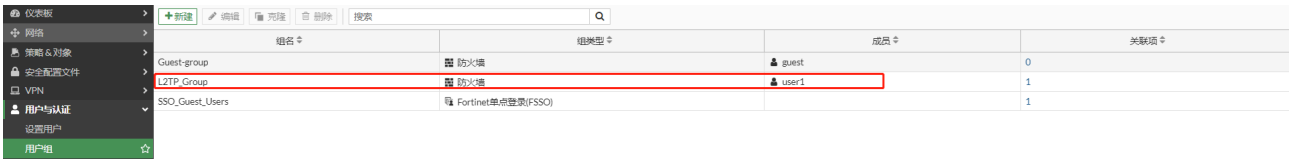






2.创建用户并将用户加入到用户组

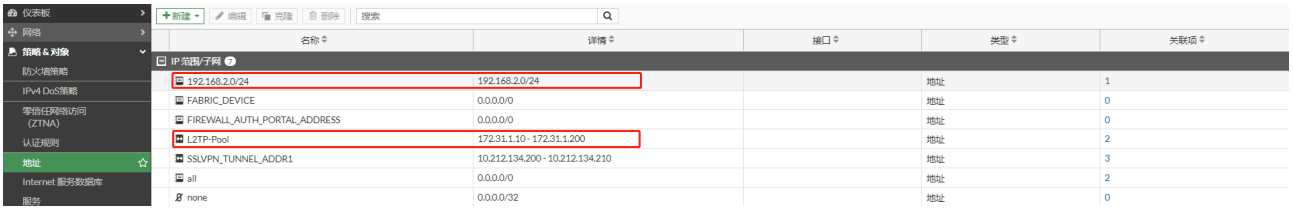




3.配置 L2TP VPN

```
config vpn l2tp
    set status enable
    set eip 172.31.1.200
    set sip 172.31.1.10
    set usrgrp "L2TP_Group"
end
```

3.配置地址对象



4.配置防火墙策略

FortiGate 会自动创建 L2TP 的接口 l2t.root。



第 1 条策略用于客户端 L2TP VPN 拨号后访问内网；第 2 条策略用于客户端 L2TP VPN 拨号后上网。

当客户端 L2TP VPN 拨号成功后，将会使用获取的地址（172.31.1.10-172.31.1.200）访问内部主机，因此内部网络需要增加到 172.31.1.0/24 网段的回程路由指向 FortiGate 或者 可以在策略中开启 NAT，那么源地址将被转换为 FortiGate 接口地址，则不用考虑回程路由。



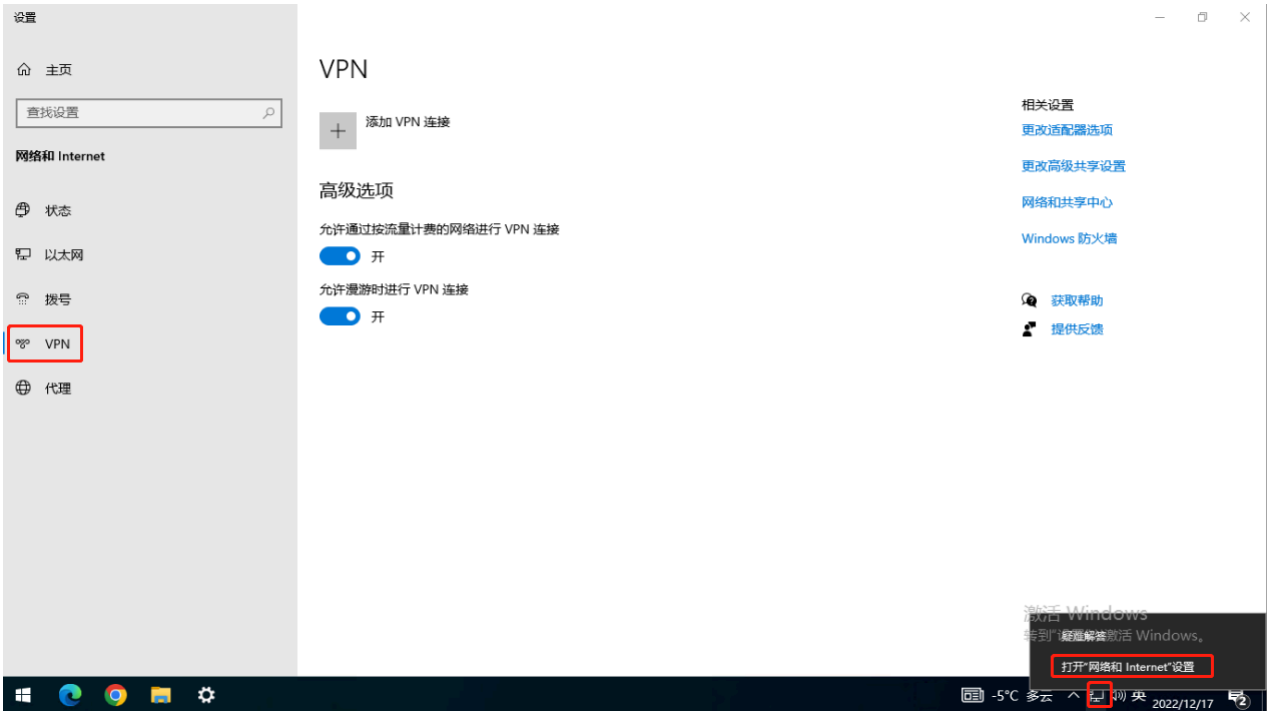
5.配置路由

将 L2TP 地址池的地址段指向 l2t.root 接口。

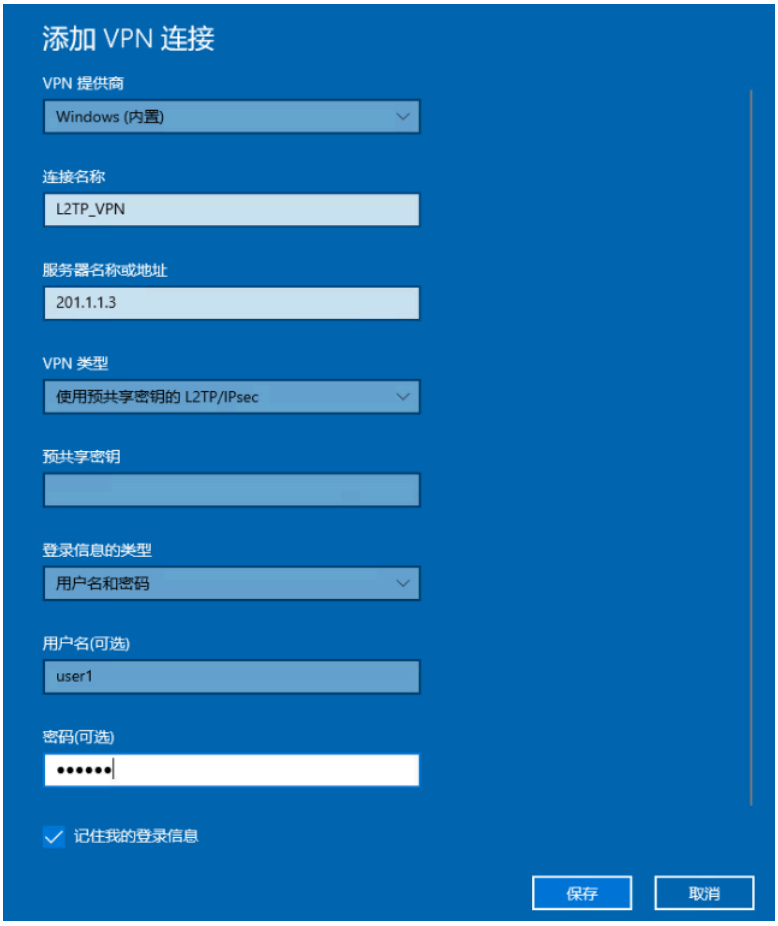
```
config router static
    edit 0
        set dst 172.31.1.0 255.255.255.0
        set device "l2t.root"
    next
end
```

配置终端 L2TP VPN

1. 右击本地连接图标，选择“打开网络和 Internet 设置”，选择 VPN。



2. 点击“添加 VPN 连接”，设置 L2TP，然后点击保存。



3. L2TP VPN 创建完整。



4. 修改注册表

Windows 系统默认使用 L2TP over IPSec 的拨号方式，因此首先会发起 IKE 的协商，并不是纯 L2TP 的协商，需要修改电脑的注册表，让电脑发起纯 L2TP 的协商，而非 IKE 的协商。

> **重要**
>
> 修改注册表后需要重启计算机才能生效。

修改方法：

- 打开注册表编辑器，找到注册表路径 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Parameters` 下的 `ProhibitIpSec` 选项（若没有则在 `Parameters` 目录下创建 DWORD (32 位) 值 `ProhibitIpSec` 即可）的值修改为 1。

- 也可以通过如下 CMD 命令执行（需要管理员权限）。

```bash
reg add "HKLM\SYSTEM\CurrentControlSet\Services\RasMan\Parameters" /v prohibiteIPsec /t REG_DWORD /d 1 /f
```

L2TP VPN 拨号测试

1. 点击创建的“L2TP_VPN”，点击“连接”。



2. 连接成功。



查看 L2TP_VPN 获取的地址

```
C:\Users\admin>ipconfig/all
以太网适配器 Ethernet1:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   物理地址. . . . . . . . . . . . . : 00-0C-29-A2-90-EA
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::65c4:a7eb:1876:83ac%5(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.1.1.5(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.1.1.1
   DHCPv6 IAID . . . . . . . . . . . : 402656297
```

```
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-29-1C-CC-4A-00-0C-29-A2-90-E0
   DNS 服务器  . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

PPP 适配器 L2TP_VPN:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : L2TP_VPN
   物理地址. . . . . . . . . . . . . :
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . : 172.31.1.12(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.255
   默认网关. . . . . . . . . . . . . : 0.0.0.0
   DNS 服务器  . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

查看终端路由表。

```
C:\Users\admin>route print
===========================================================================
接口列表
  6...00 09 0f fe 00 01 ......Fortinet Virtual Ethernet Adapter (NDIS 6.30)
 10...00 0c 29 a2 90 e0 ......Intel(R) 82574L Gigabit Network Connection
  5...00 0c 29 a2 90 ea ......Intel(R) 82574L Gigabit Network Connection #2
 46...........................L2TP_VPN
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0         10.1.1.1         10.1.1.5   4506
          0.0.0.0          0.0.0.0            在链路上       172.31.1.12     26
         10.1.1.0    255.255.255.0            在链路上          10.1.1.5   4506
         10.1.1.5  255.255.255.255            在链路上          10.1.1.5   4506
       10.1.1.255  255.255.255.255            在链路上          10.1.1.5   4506
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1   4556
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1   4556
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1   4556
      172.31.1.12  255.255.255.255            在链路上       172.31.1.12    281
```

3. 访问内部主机。

```
C:\Users\admin>ping 192.168.2.10

正在 Ping 192.168.2.10 具有 32 字节的数据:
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63

192.168.2.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

访问 Internet



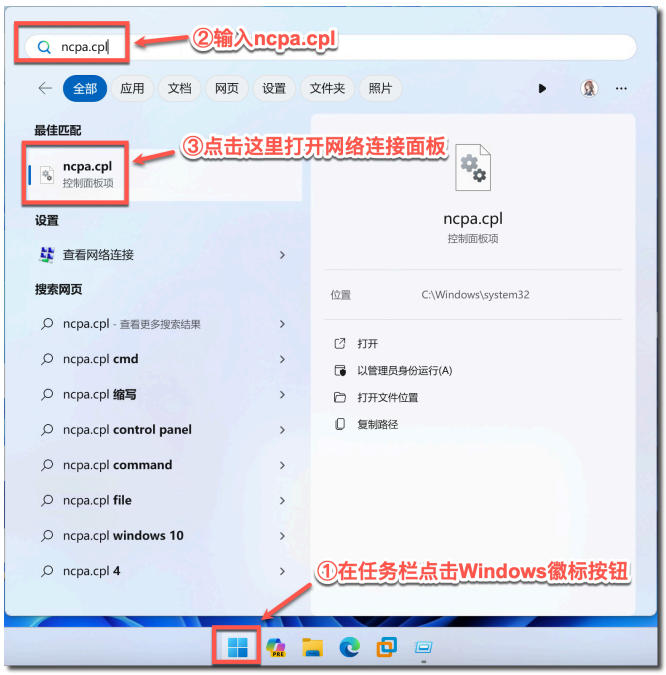
注意事项

如果 FortiGate 上配置的 L2TP 用户为 LDAP 认证用户，则需要确保客户端 L2TP 连接的认证方法为 PAP。

重要

- LDAP 用户身份验证支持 PPTP、L2TP、IPsec VPN 和防火墙身份验证，但由于点对点协议的限制，只支持 PAP（Packet Authentication Protocol）。
- 由于 Windows 本机客户端默认使用 MSCHAP 或 MSCHAPv2，LDAP 认证场景下，您需要将其更改为 PAP。

1. 在 Windows 系统中按照下图方式打开网络连接面板。



2. 在弹出的网络连接面板中，右键点击已创建的 L2TP 连接，然后点击属性按钮。



3. 在弹出的 L2TP 连接属性中，选择“安全”面板，然后勾选“允许使用这些协议”，勾选“未加密的密码 (PAP)(H)”，取消勾选 CHAP 和 MS - CHAP v2，然后点击确定按钮。之后即可正常连接 L2TP。



PPTP VPN

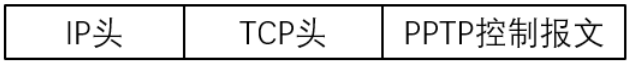
PPTP VPN 简介

PPTP（Point to Point Tunneling Protocol），即点对点隧道协议，默认端口号：TCP 1723。该协议是在 PPP 协议的基础上开发的一种新的增强型安全协议，可以通过密码验证协议（PAP）、可扩展认证协议（EAP）等方法增强安全性，远程用户可以通过 PPTP VPN 拨号访问公司内网资源。

PPTP 数据报文

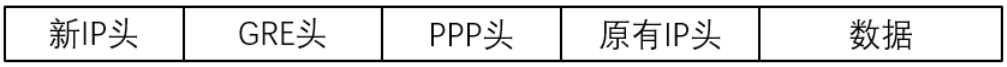
PPTP 数据报文分为两种：

控制报文：用于创建、维护、终止 PPTP 连接；




```
No.  Time    Source      Destination  Protocol  Length  Source Port  Destinationo Port  Info
  9  0.009…  10.1.1.5    201.1.1.3    PPTP          78        53956               1723  Set-Link-Info

> Frame 9: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{0FB48074-675E-4FD5-8D41-A0C4ACC8F9F7}, id 0
> Ethernet II, Src: VMware_a2:90:ea (00:0c:29:a2:90:ea), Dst: VMware_2e:3f:cf (00:0c:29:2e:3f:cf)
> Internet Protocol Version 4, Src: 10.1.1.5, Dst: 201.1.1.3
> Transmission Control Protocol, Src Port: 53956, Dst Port: 1723, Seq: 2729513262, Ack: 3409293323, Len: 24
v Point-to-Point Tunnelling Protocol
    Length: 24
    Message type: Control Message (1)
    Magic Cookie: 0x1a2b3c4d (correct)
    Control Message Type: Set-Link-Info (15)
    Reserved: 0000
    Peer Call ID: 48890
    Reserved: 0000
    Send ACCM: 0xffffffff
    Receive ACCM: 0xffffffff
```

数据报文：用来传送 PPP 数据包，PPP 数据包封装在 GRE 里进行传送。




```
No.  Time    Source      Destination  Protocol  Length  Source Port  Destinationo Port  Info
 73  2.226…  10.1.1.5    201.1.1.3    PPP Comp      91                                 Compressed data

> Frame 73: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface \Device\NPF_{0FB48074-675E-4FD5-8D41-A0C4ACC8F9F7}, id 0
> Ethernet II, Src: VMware_a2:90:ea (00:0c:29:a2:90:ea), Dst: VMware_2e:3f:cf (00:0c:29:2e:3f:cf)
v Internet Protocol Version 4, Src: 10.1.1.5, Dst: 201.1.1.3
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 77
    Identification: 0x2f02 (12034)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: Generic Routing Encapsulation (47)
    Header Checksum: 0x3676 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.1.1.5
    Destination Address: 201.1.1.3
v Generic Routing Encapsulation (PPP)
  > Flags and Version: 0x3001
    Protocol Type: PPP (0x880b)
    Payload Length: 45
    Call ID: 48890
    Sequence Number: 23
v Point-to-Point Protocol
    Protocol: Compressed datagram (0x00fd)
  PPP Compressed Datagram
```

PPTP 协议交互过程

1. 建立 TCP 三次握手；

2. 建立 PPTP 控制层连接：

   Start Control Connection Request：Client 发起控制链接请求，用于初始化 PPTP Client 和 Server 之间的 Tunnel, 开始 Tunnel 的建立过程；

   Start Control Connection Reply：Server 控制链接响应，表示接受了对端的连接请求；

   Outgoing Call Request：创建隧道，选择一个调用 ID 用于标识客户端到服务器发送数据的 PPTP 隧道；

   Outgoing Call Reply：隧道创建成功应答，选择一个调用 ID 用于标识从服务器到客户端发送数据的 PPTP 隧道；

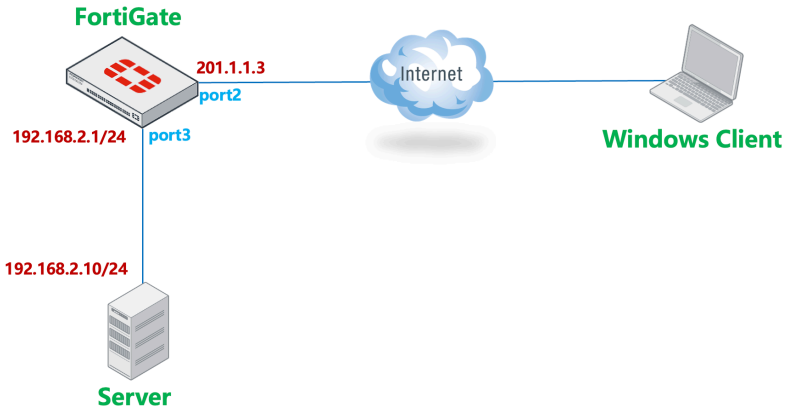
   Set-Link-Info：指定 PPP 协商选项。

No.	Time	Source	Destination	Protocol	Length	Source Port	Destinationo Port	Info
1	0.000…	10.1.1.5	201.1.1.3	TCP	66	53956	1723	53956 → 1723 [SYN] Seq=2729512937 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000…	201.1.1.3	10.1.1.5	TCP	66	1723	53956	1723 → 53956 [SYN, ACK] Seq=3409293134 Ack=2729512938 Win=65535 Len=0 MSS=1460 SACK_PERM=1 WS=16384
3	0.000…	10.1.1.5	201.1.1.3	TCP	54	53956	1723	53956 → 1723 [ACK] Seq=2729512938 Ack=3409293135 Win=2102272 Len=0
4	0.001…	10.1.1.5	201.1.1.3	PPTP	210	53956	1723	Start-Control-Connection-Request
5	0.002…	201.1.1.3	10.1.1.5	TCP	60	1723	53956	1723 → 53956 [ACK] Seq=3409293135 Ack=2729513094 Win=180224 Len=0
6	0.002…	201.1.1.3	10.1.1.5	PPTP	210	1723	53956	Start-Control-Connection-Reply
7	0.002…	10.1.1.5	201.1.1.3	PPTP	222	53956	1723	Outgoing-Call-Request
8	0.003…	201.1.1.3	10.1.1.5	PPTP	86	1723	53956	Outgoing-Call-Reply
9	0.009…	10.1.1.5	201.1.1.3	PPTP	78	53956	1723	Set-Link-Info
10	0.012…	10.1.1.5	201.1.1.3	PPP LCP	71			Configuration Request
11	0.017…	201.1.1.3	10.1.1.5	PPP LCP	75			Configuration Request
12	0.017…	10.1.1.5	201.1.1.3	PPP LCP	79			Configuration Ack
13	0.017…	201.1.1.3	10.1.1.5	GRE	60			Encapsulated PPP
14	0.047…	201.1.1.3	10.1.1.5	TCP	60	1723	53956	1723 → 53956 [ACK] Seq=3409293323 Ack=2729513286 Win=180224 Len=0
15	2.009…	10.1.1.5	201.1.1.3	PPP LCP	71			Configuration Request
16	2.009…	201.1.1.3	10.1.1.5	GRE	60			Encapsulated PPP

网络需求

在外移动办公的工作人员需要通过 PPTP VPN 隧道模式拨入到公司内网来对内网主机进行访问。

网络拓扑

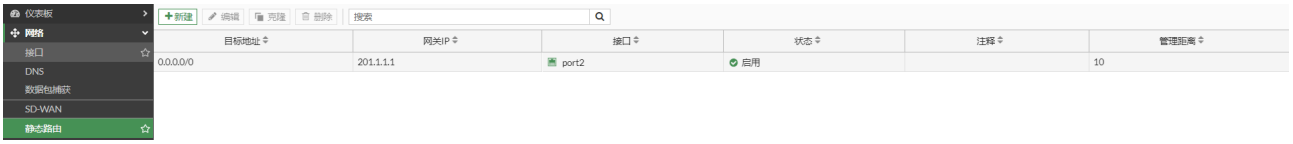


配置步骤

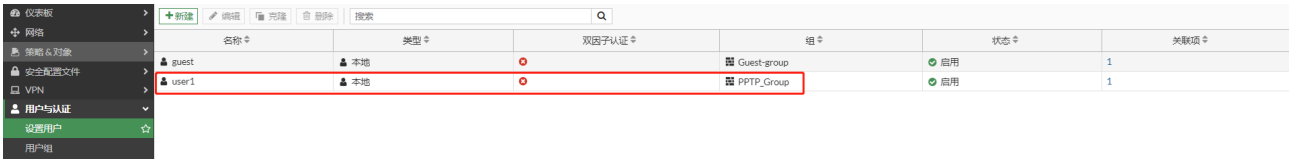
配置 FortiGate PPTP VPN

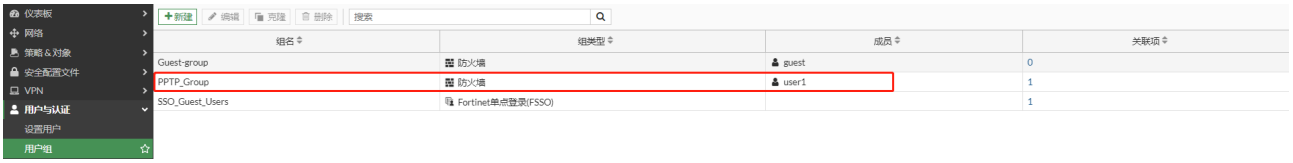
1.基础配置





2.创建用户并将用户加入到用户组

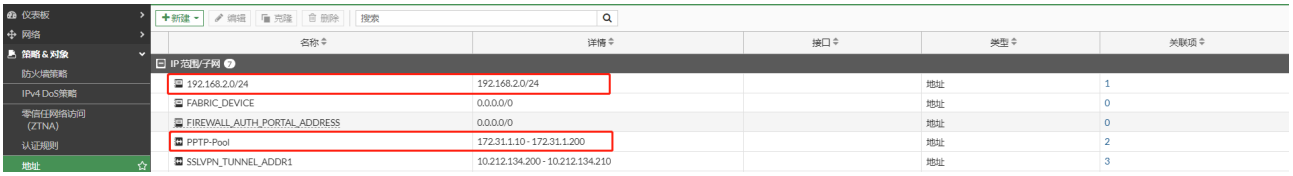




3.配置 PPTP VPN

```
config vpn pptp
    set status enable
    set eip 172.31.1.200
    set sip 172.31.1.10
    set usrgrp "PPTP_Group"
end
```

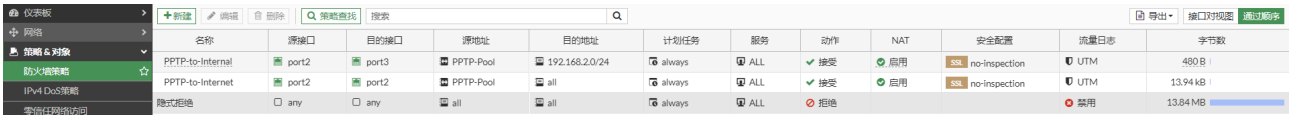
4.配置地址对象



5.配置防火墙策略

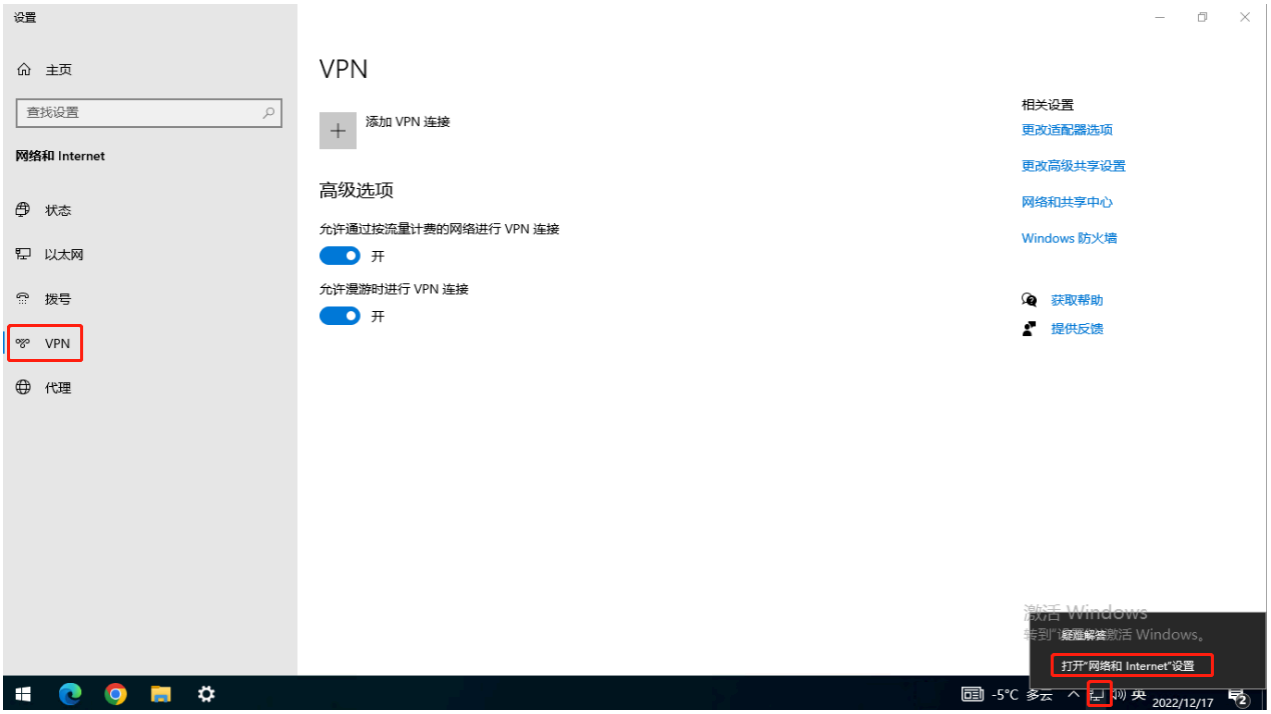
第 1 条策略用于客户端 PPTP VPN 拨号后访问内网；第 2 条策略用于客户端 PPTP VPN 拨号后上网。

当客户端 PPTP VPN 拨号成功后，将会使用获取的地址（172.31.1.10-172.31.1.200）访问内部主机，因此内部网络需要增加到 172.31.1.0/24 网段的回程路由指向 FortiGate 或者 可以在策略中开启 NAT，那么源地址将被转换为 FortiGate 接口地址，则不用考虑回程路由。

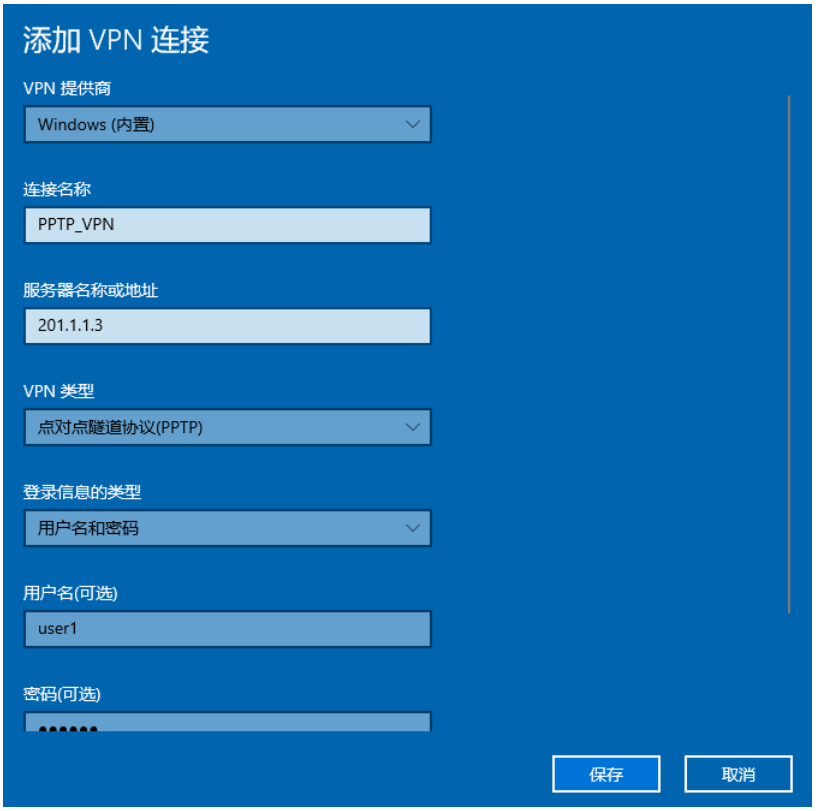


###配置终端 PPTP VPN

1. 右击本地连接图标，选择“打开网络和 Internet 设置”，选择 VPN。



2. 点击“添加 VPN 连接”，设置 PPTP，然后点击保存。



3. PPTP VPN 创建完成。



PPTP VPN 拨号测试

1. 点击创建的“PPTP_VPN”，点击“连接”。



2. 连接成功。



查看 PPTP VPN 获取的地址，可以看到 FortiGate 无法推送 DNS 给 PPTP 客户端，如果 PPTP 客户端需要使用 DNS，则需要手动编辑客户端的 PPTP 连接

```
C:\Users\admin>ipconfig/all
以太网适配器 Ethernet1:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   物理地址. . . . . . . . . . . . . : 00-0C-29-A2-90-EA
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::65c4:a7eb:1876:83ac%5(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.1.1.5(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 10.1.1.1
   DHCPv6 IAID . . . . . . . . . . . : 402656297
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-29-1C-CC-4A-00-0C-29-A2-90-E0
   DNS 服务器  . . . . . . . . . . . : 114.114.114.114
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

PPP 适配器 PPTP_VPN:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : PPTP_VPN
   物理地址. . . . . . . . . . . . . :
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . : 172.31.1.11(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.255
   默认网关. . . . . . . . . . . . . : 0.0.0.0
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

查看终端路由表。

```
C:\Users\admin>route print
===========================================================================
接口列表
  6...00 09 0f fe 00 01 ......Fortinet Virtual Ethernet Adapter (NDIS 6.30)
 10...00 0c 29 a2 90 e0 ......Intel(R) 82574L Gigabit Network Connection
  5...00 0c 29 a2 90 ea ......Intel(R) 82574L Gigabit Network Connection #2
 46...........................PPTP
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
```

```
          0.0.0.0          0.0.0.0         10.1.1.1         10.1.1.5   4506
          0.0.0.0          0.0.0.0            在链路上      172.31.1.11     26
         10.1.1.0    255.255.255.0            在链路上         10.1.1.5   4506
         10.1.1.5  255.255.255.255            在链路上         10.1.1.5   4506
       10.1.1.255  255.255.255.255            在链路上         10.1.1.5   4506
        127.0.0.0        255.0.0.0            在链路上       127.0.0.1   4556
        127.0.0.1  255.255.255.255            在链路上       127.0.0.1   4556
  127.255.255.255  255.255.255.255            在链路上       127.0.0.1   4556
      172.31.1.11  255.255.255.255            在链路上      172.31.1.11    281
```

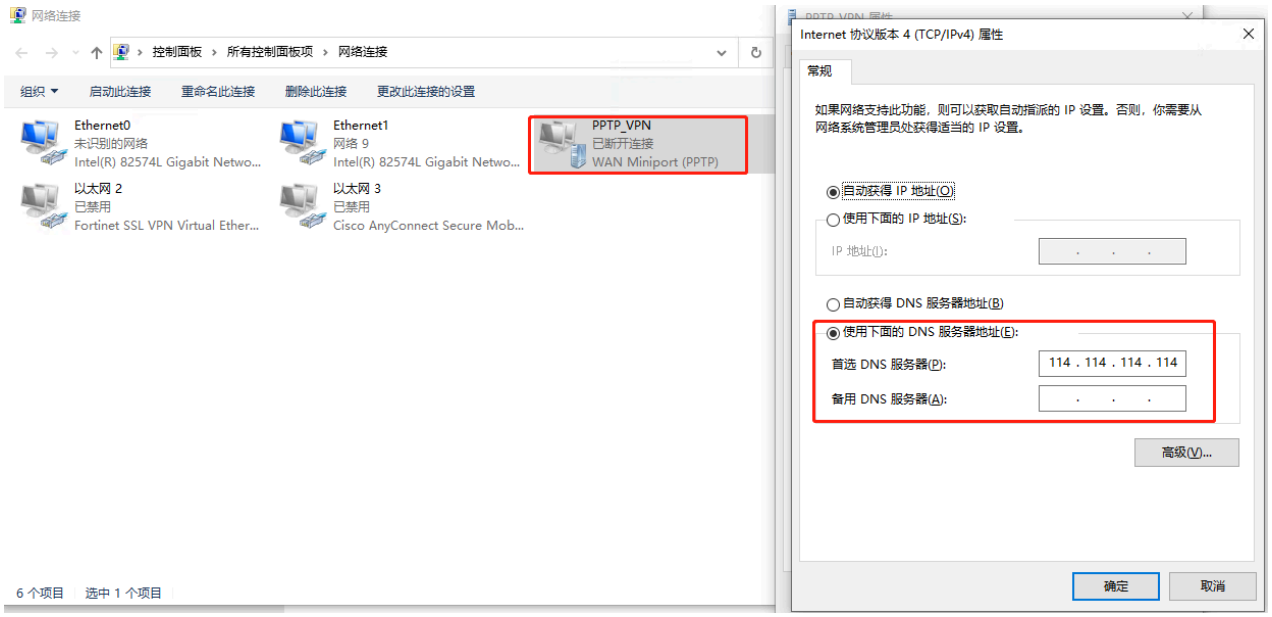
3. 访问内部主机。

```
C:\Users\admin>ping 192.168.2.10

正在 Ping 192.168.2.10 具有 32 字节的数据:
来自 192.168.2.10 的回复: 字节=32 时间=1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间<1ms TTL=63
来自 192.168.2.10 的回复: 字节=32 时间=1ms TTL=63

192.168.2.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

4. 访问 Internet

```
C:\Users\admin>ping 114.114.114.114

正在 Ping 114.114.114.114 具有 32 字节的数据:
来自 114.114.114.114 的回复: 字节=32 时间=23ms TTL=82
来自 114.114.114.114 的回复: 字节=32 时间=21ms TTL=71
来自 114.114.114.114 的回复: 字节=32 时间=21ms TTL=89
来自 114.114.114.114 的回复: 字节=32 时间=20ms TTL=75

114.114.114.114 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 20ms, 最长 = 23ms, 平均 = 21ms
```

5. 如果需要通过 PPTP 访问域名，则需要手动设置 PPTP 的 DNS。先中断 PPTP 连接，手动设置 DNS，然后再次拨号。



6. 访问 `www.baidu.com` 正常。


百度一下，你就知道
baidu.com
新闻
hao123
地图
贴吧
视频
图片
网盘
更多
Bai du 百度
百度一下

其他 VPN 技术

- GRE

GRE

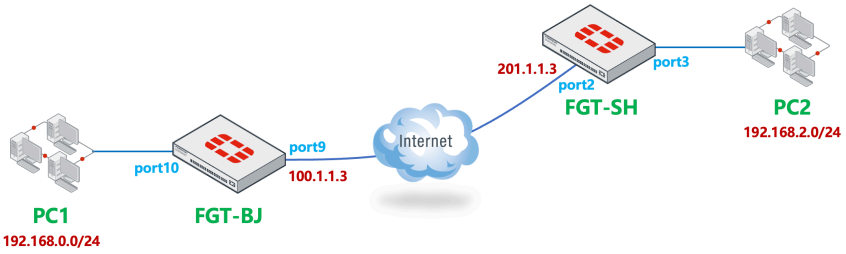
GRE 简介

GRE（Generic Routing Encapsulation，通用路由封装）协议是对某些网络层协议（如 IPv4，IPv6，IPX）的数据报文进行封装，使这些被封装的数据报文能够在另一个网络层协议中传输。

网络需求

通过 GRE 将 2 个局域网连接起来，实现 192.168.0.0/24 与 192.168.2.0/24 两个网段的通信。

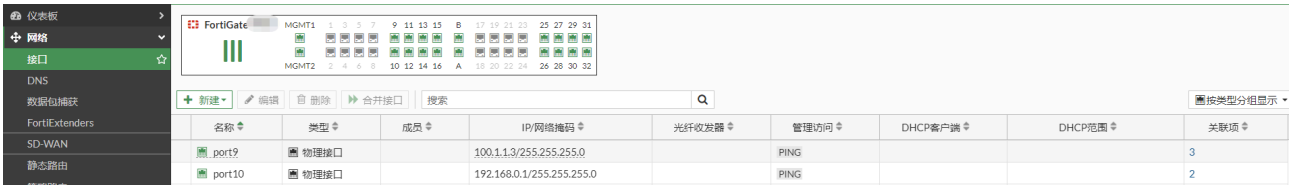
网络拓扑



配置步骤

FGT - BJ GRE 配置

1.基础配置





2.配置 GRE

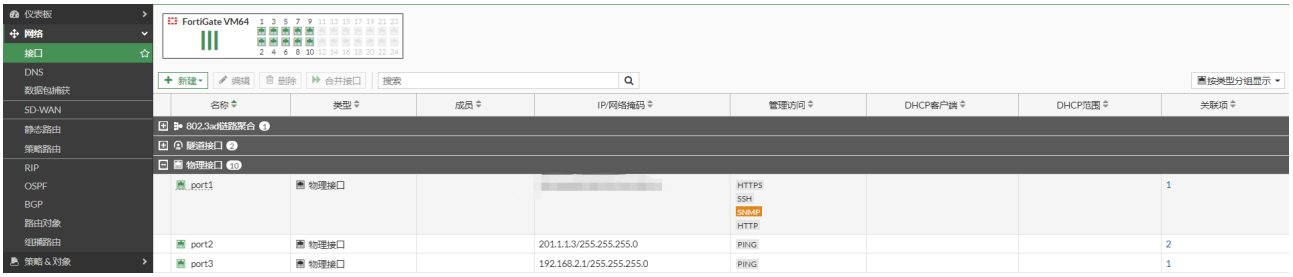
```
config system gre-tunnel
    edit "to-fgt_sh"
        set interface "port9"
        set remote-gw 201.1.1.3
        set local-gw 100.1.1.3
    next
end
```

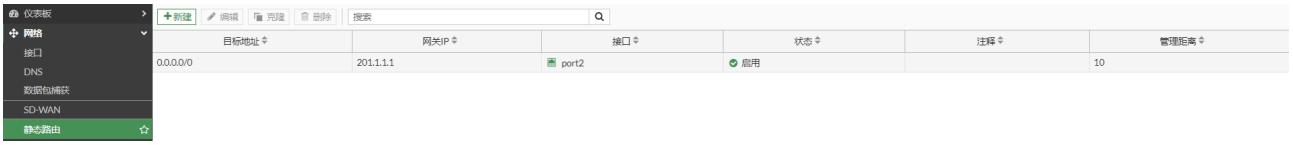
3.配置策略



FGT - SH GRE 配置

1.基础配置





2.配置 GRE

```
config system gre-tunnel
    edit "to-fgt_bj"
        set interface "port2"
        set remote-gw 100.1.1.3
        set local-gw 201.1.1.3
    next
end
```

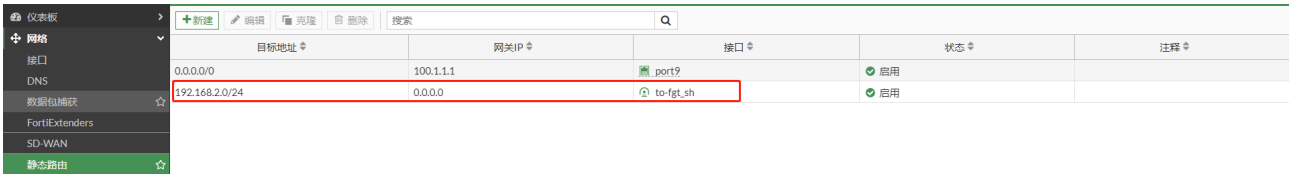
3.配置策略



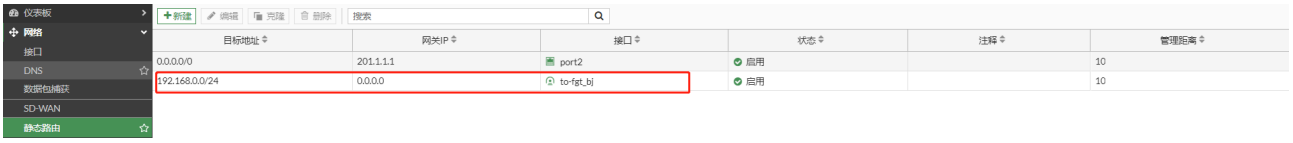
路由及业务测试

静态路由互通

1.FGT - BJ 配置静态路由



2.FGT - SH 配置静态路由



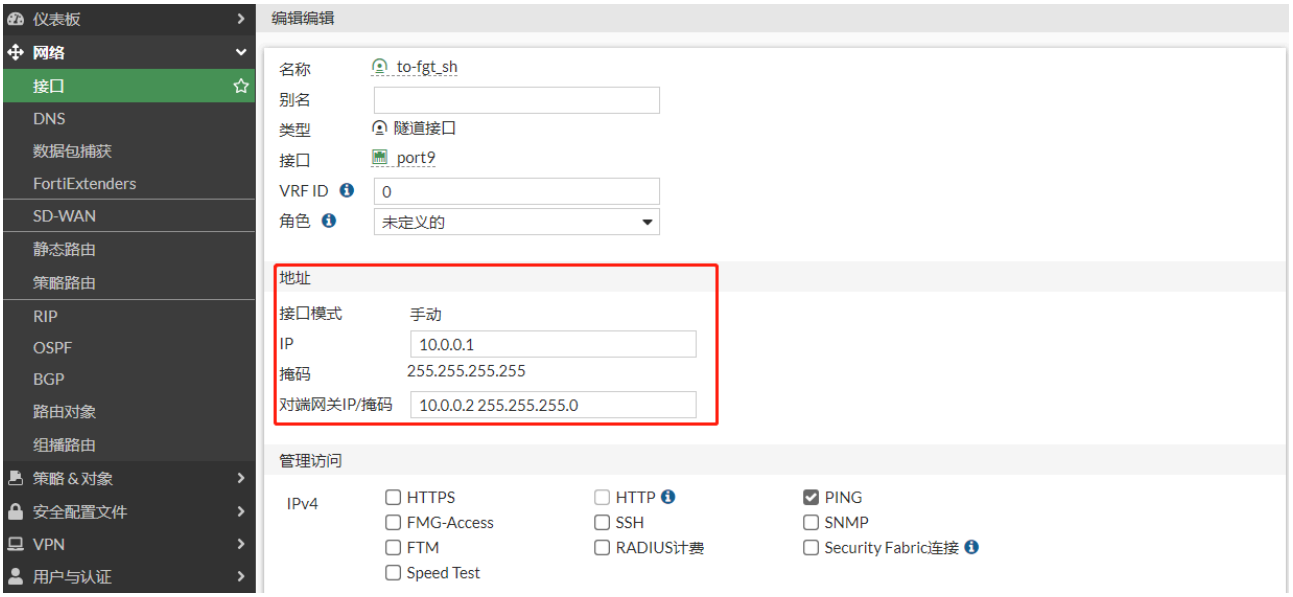
3.业务测试

从 FGT - BJ 内网能 ping 通 FGT - SH 的内网

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6225014  bytes 1799326210 (1.6 GiB)
        RX errors 0  dropped 3749612  overruns 0  frame 0
        TX packets 75758  bytes 5398290 (5.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC1# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.931 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.994 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.673 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.673/0.938/1.155/0.174 ms
```

动态路由互通

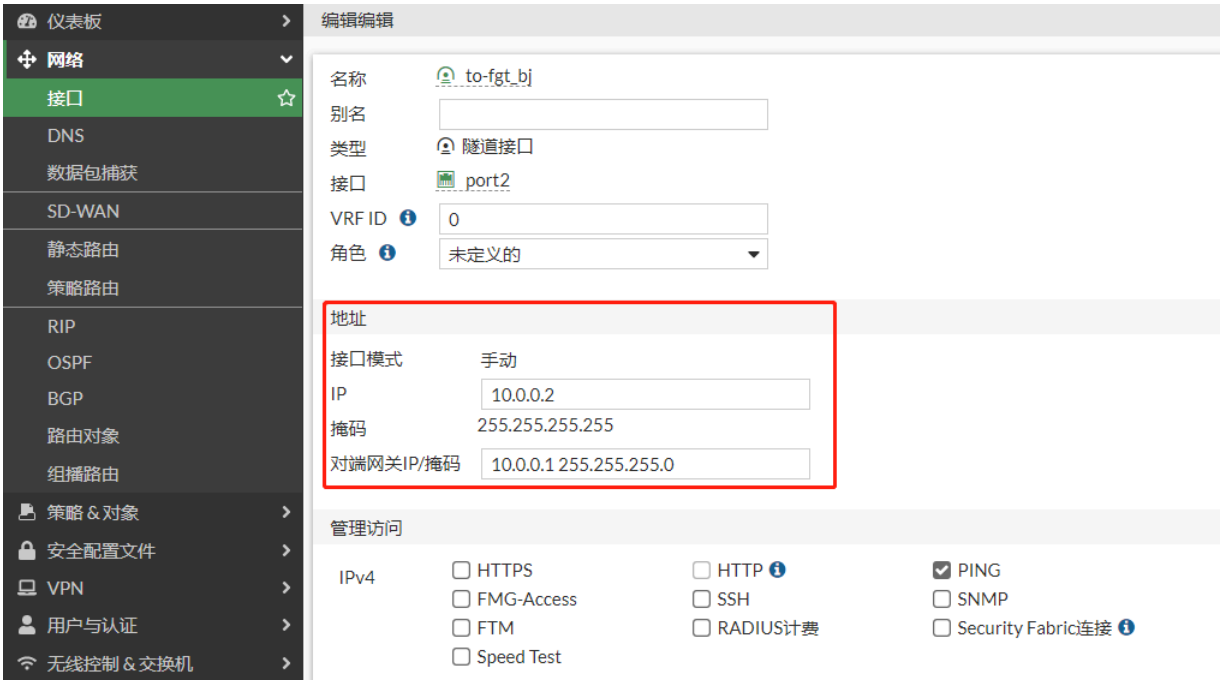
1.FGT - BJ 配置 GRE 隧道接口 IP



查看路由表

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port9, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_sh
C       10.0.0.1/32 is directly connected, to-fgt_sh
C       100.1.1.0/24 is directly connected, port9
C       192.168.0.0/24 is directly connected, port10
```

2.FGT - SH 配置 GRE 隧道接口 IP



查看路由表

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 201.1.1.1, port2, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_bj
C       10.0.0.2/32 is directly connected, to-fgt_bj
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected
```

3.GRE 接口连通性测试

```
从FGT-BJ的GRE接口地址 ping FGT-SH的GRE接口地址
# execute ping-options source 10.0.0.1
# execute ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=0.5 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=0.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=0.5 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=0.4 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=255 time=0.1 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.4/0.5 ms
```

4.配置动态路由

BGP 互通

1.配置 BGP

FGT - BJ 配置 BGP

```
config router bgp
    set as 65001
    set router-id 100.1.1.3
    config neighbor
        edit "10.0.0.2"
            set next-hop-self enable
            set soft-reconfiguration enable
            set interface "to-fgt_sh"
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 192.168.0.0 255.255.255.0
        next
    end
end
```

FGT - SH 配置 BGP

```
config router bgp
    set as 65001
    set router-id 201.1.1.3
    config neighbor
        edit "10.0.0.1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 192.168.2.0 255.255.255.0
        next
    end
end
```

2.查看 BGP 邻居和路由

FGT - BJ 查看 BGP 邻居和路由

```
# get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 10.0.0.2, remote AS 65001, local AS 65001, internal link
  BGP version 4, remote router ID 201.1.1.3
  BGP state = Established, up for 04:22:48
  Last read 00:00:13, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 302 messages, 0 notifications, 0 in queue
  Sent 302 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 2, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  Community attribute sent to this neighbor (both)
  1 accepted prefixes, 1 prefixes in rib
  1 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
```

```
 Connections established 1; dropped 0
Local host: 10.0.0.1, Local port: 179
Foreign host: 10.0.0.2, Foreign port: 14233
Egress interface: 55
Nexthop: 10.0.0.1
Nexthop interface: to-fgt_sh
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network

# get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port9, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_sh
C       10.0.0.1/32 is directly connected, to-fgt_sh
C       100.1.1.0/24 is directly connected, port9
C       192.168.0.0/24 is directly connected, port10
B       192.168.2.0/24 [200/0] via 10.0.0.2 (recursive is directly connected, to-fgt_sh), 04:22:34
```

FGT - SH 查看 BGP 邻居和路由

```
# get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 10.0.0.1, remote AS 65001, local AS 65001, internal link
  BGP version 4, remote router ID 100.1.1.3
  BGP state = Established, up for 04:24:18
  Last read 00:00:30, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 303 messages, 0 notifications, 0 in queue
  Sent 304 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 8, neighbor version 7
  Index 1, Offset 0, Mask 0x2
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  Community attribute sent to this neighbor (both)
  1 accepted prefixes, 1 prefixes in rib
  1 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
 Connections established 1; dropped 0
Local host: 10.0.0.2, Local port: 14233
Foreign host: 10.0.0.1, Foreign port: 179
Egress interface: 34
Nexthop: 10.0.0.2
Nexthop interface: to-fgt_bj
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network

FGVM08TM22000410 # get router info routing-table  all
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 201.1.1.1, port2, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_bj
C       10.0.0.2/32 is directly connected, to-fgt_bj
B       192.168.0.0/24 [200/0] via 10.0.0.1 (recursive is directly connected, to-fgt_bj), 04:23:24
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
```

3.业务测试

从 FGT - BJ 内网能 ping 通 FGT - SH 的内网

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6225014  bytes 1799326210 (1.6 GiB)
        RX errors 0  dropped 3749612  overruns 0  frame 0
        TX packets 75758  bytes 5398290 (5.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC1# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.931 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.994 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.673 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.673/0.938/1.155/0.174 ms
```

OSPF 互通

1.配置 OSPF

FGT - BJ 配置 OSPF

```
config router ospf
    set router-id 100.1.1.3
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.0.0.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.0.0 255.255.255.0
        next
    end
end
```

FGT - SH 配置 OSPF

```
config router ospf
    set router-id 201.1.1.3
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.0.0.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.2.0 255.255.255.0
        next
    end
en
```

2.查看 OSPF 邻居和路由

FGT - BJ 查看 OSPF 邻居和路由

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
201.1.1.3         1   Full/ -         00:00:33    10.0.0.2        to-fgt_sh

# get router info routing-table  all
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 100.1.1.1, port9, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_sh
C       10.0.0.1/32 is directly connected, to-fgt_sh
C       100.1.1.0/24 is directly connected, port9
C       192.168.0.0/24 is directly connected, port10
O       192.168.2.0/24 [110/101] via 10.0.0.2, to-fgt_sh, 00:01:12
```

FGT - SH 查看 OSPF 邻居和路由

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
100.1.1.3         1   Full/ -         00:00:34    10.0.0.1        to-fgt_bj

# get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 201.1.1.1, port2, [1/0]
C       10.0.0.0/24 is directly connected, to-fgt_bj
C       10.0.0.2/32 is directly connected, to-fgt_bj
O       192.168.0.0/24 [110/101] via 10.0.0.1, to-fgt_bj, 00:01:43
C       192.168.2.0/24 is directly connected, port3
C       201.1.1.0/24 is directly connected, port2
```

3.业务测试

从 FGT - BJ 内网能 ping 通 FGT - SH 的内网

```
PC1# ifconfig ens224
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::82c8:edfd:199d:70b0  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e8:ad:a9  txqueuelen 1000  (Ethernet)
        RX packets 6225014  bytes 1799326210 (1.6 GiB)
        RX errors 0  dropped 3749612  overruns 0  frame 0
        TX packets 75758  bytes 5398290 (5.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

PC1# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.931 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.994 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.673 ms
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.673/0.938/1.155/0.174 ms
```

用户与认证

认证相关参数设置

LDAP

本地用户和用户组

本地用户密码策略

LDAP 认证

- LDAP 认证原理
- LDAP 认证配置与排错
- LDAP 常见错误代码
- LDAPS CA 强制认证

Radius 认证

- Radius 认证原理
- Radius 认证配置与排错

SAML 认证

- 防火墙用户与 SSL VPN SAML 认证（Windows ADFS）
- 显式代理 SAML 认证（Windows ADFS）

限制认证并发源 IP 数

Radius

FortiToken 认证

- FortiToken Mobile 在 FortiGate 上的激活
- FortiToken Mobile APP 软件下载与安装
- FortiToken Mobile 与用户进行绑定
- 免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法

FSSO 认证

- FSSO 认证
- FSSO Agent SSL 配置

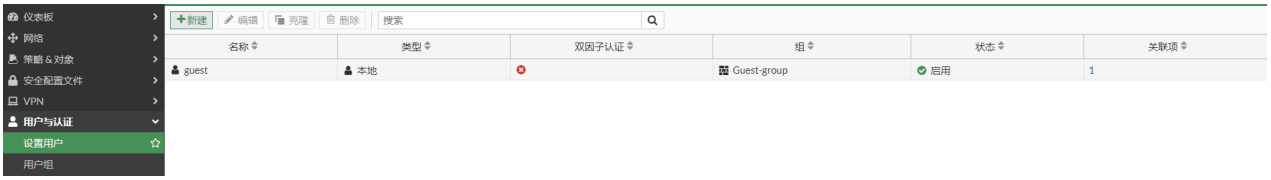
认证常见问题

- 认证与策略顺序的关系
- Web 认证弹出 2 个页面
- 升级版本后 Radius 连接失败

本地用户和用户组

创建本地用户

1. 选择“用户与认证”-->“设置用户”，点击“新建”



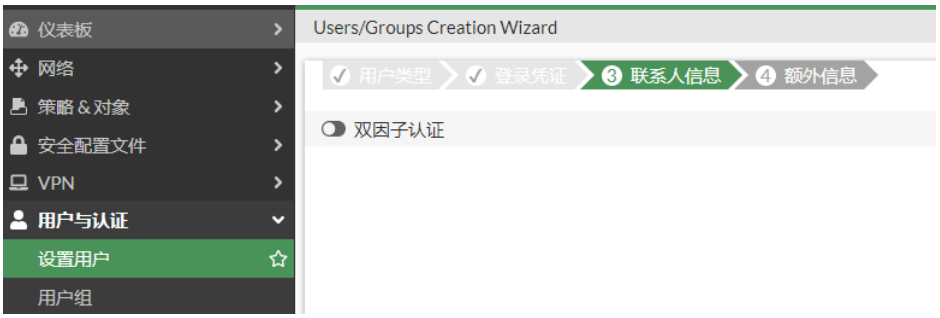
2. 选择“本地用户”，点击“下一步”



3. 输入用户名和密码，点击“下一步”



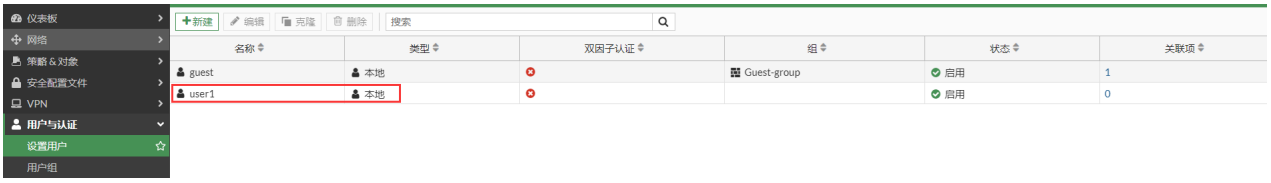
4. 如果本地用户需要绑定 FortiGate 本地的 Fortitoken，则勾选“双因子认证”。这里不需要，直接点击“下一步”



5. 如果需要现在为本地用户指定一个用户组，可以勾选“用户组”来指定，也可以等创建完用户后，在用户组菜单中添加。这里不勾选，直接点击“提交”

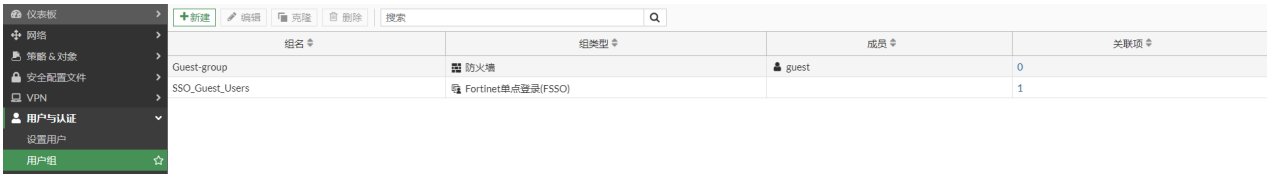


6. 配置完成

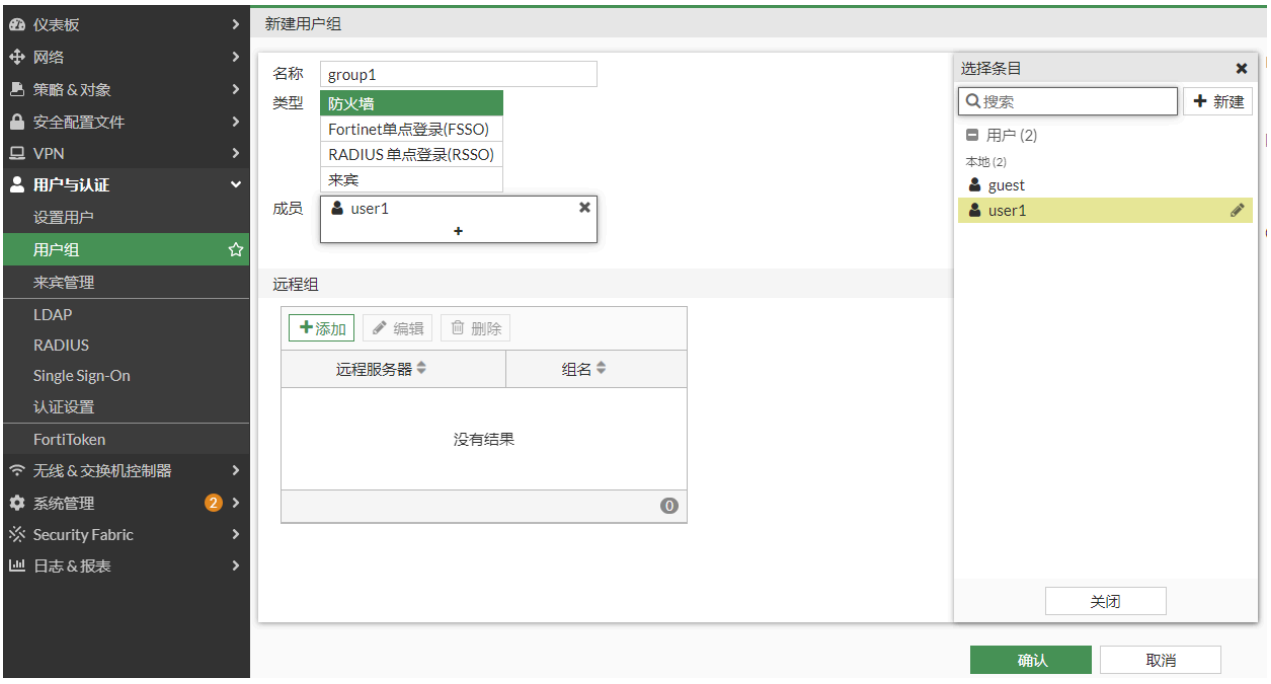


创建本地用户组

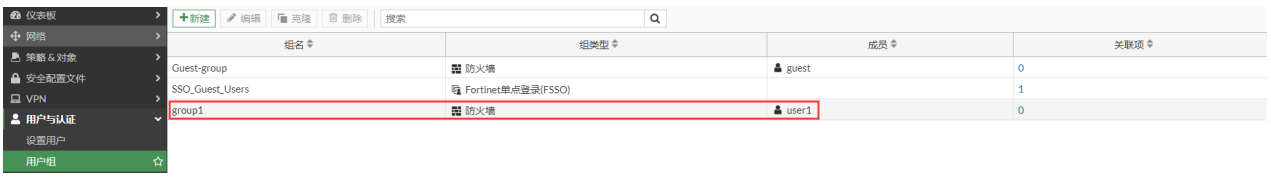
1. 选择“用户与认证”-->“用户组”，点击“新建”



2. 输入用户组名称，在“成员”中选择已创建好的用户，如 user1，可以添加多个用户，点击“确认”



3. 配置完成



本地用户密码策略

功能简介

本地用户密码策略用于约束 FortiGate 防火墙本地用户的密码复杂度、密码有效期和密码复用行为。相比早期版本只控制过期时间的本地用户密码策略，FortiOS 7.4.1 及后续版本可以进一步限制最小长度、大小写字母、数字、特殊字符、与旧密码不同的字符数量，以及是否允许重复使用旧密码。

该功能适用于使用 `config user local` 创建的本地防火墙用户，例如防火墙策略认证、Captive Portal、SSL VPN 或其他引用本地用户/用户组的认证场景。它不等同于系统管理员密码策略，管理员密码策略仍然通过 `config system password-policy` 配置。

配置命令

创建密码策略

```bash
config user password-policy
    edit <policy_name>
        set minimum-length <8-128>
        set min-lower-case-letter <0-128>
        set min-upper-case-letter <0-128>
        set min-non-alphanumeric <0-128>
        set min-number <0-128>
        set min-change-characters <0-128>
        set expire-status {enable | disable}
        set expire-days <0-999>
        set warn-days <0-30>
        set expired-password-renewal {enable | disable}
        set reuse-password {enable | disable}
        set reuse-password-limit <0-20>
    next
end
```

绑定到本地用户

密码策略创建后，需要在本地用户上通过 `passwd-policy` 关联，才会对该用户生效。

```bash
config user local
    edit <user_name>
        set type password
        set passwd-policy <policy_name>
    next
end
```

参数说明

- `minimum-length`：密码最小长度，范围为 8~128，默认值为 8。
- `min-lower-case-letter`：密码中至少包含的小写字母数量，范围为 0~128，默认值为 0。
- `min-upper-case-letter`：密码中至少包含的大写字母数量，范围为 0~128，默认值为 0。
- `min-non-alphanumeric`：密码中至少包含的非字母数字字符数量，范围为 0~128，默认值为 0。
- `min-number`：密码中至少包含的数字数量，范围为 0~128，默认值为 0。
- `min-change-characters`：新密码中至少包含多少个旧密码中不存在的字符，范围为 0~128，默认值为 0。如果同时配置了 `reuse-password`，该参数优先级更高。
- `expire-status`：是否启用密码过期机制，默认值为 `disable`。
- `expire-days`：密码有效期，单位为天，范围为 0~999，默认值为 180。只有启用 `expire-status` 后，才会按该参数强制用户修改密码。
- `warn-days`：密码到期前多少天提示用户修改密码，范围为 0~30，默认值为 15。
- `expired-password-renewal`：是否允许用户在密码已经过期后自行更新密码，默认值为 `disable`。
- `reuse-password`：是否允许重复使用旧密码，默认值为 `enable`。
- `reuse-password-limit`：允许重复使用旧密码的次数，范围为 0~20，默认值为 0。设置为 0 时表示不限制重复使用次数。

配置示例

以下示例创建一个名为 `strict-local-user` 的本地用户密码策略，并将其绑定到用户 `pwd-test1`。

```bash
config user password-policy
    edit "strict-local-user"
        set minimum-length 8
        set min-lower-case-letter 1
        set min-upper-case-letter 1
        set min-non-alphanumeric 3
        set min-number 3
        set min-change-characters 2
        set expire-status enable
        set expire-days 90
        set warn-days 7
        set expired-password-renewal enable
        set reuse-password disable
    next
end

config user local
    edit "bing"
        set type password
        set passwd-policy "strict-local-user"
    next
end
```

当新密码满足策略时，可以正常提交：

```bash
config user local
    edit "bing"
        set passwd CCbcset123!!!
    next
end
```

当新密码不满足策略时，CLI 会拒绝提交并提示缺少的复杂度条件。例如以下密码只有 2 个数字，不满足 `min-number 3`：

```bash
config user local
    edit "bing"
        set passwd CCbXsetp23!!!
New password must conform to the password policy enforced on this user:
...
Command fail. Return code -49
```

实际提示中会列出未满足的条件，例如最小长度、大小写字母数量、特殊字符数量、数字数量、与旧密码不同的字符数量，以及是否重复使用了旧密码。

升级注意事项

从 FortiOS 7.4.0 升级到 7.4.1 或后续版本后，原有 `config user password-policy` 中的 `expire-days`、`warn-days`、`expired-password-renewal` 配置会保留，但需要通过 CLI 启用 `expire-status`，密码过期机制才会真正生效。

可以在升级后检查已有策略：

```bash
config user password-policy
    edit <policy_name>
        get
    next
end
```

如果看到 `expire-status: disable`，即使 `expire-days` 已经配置，FortiGate 也不会按照该天数强制本地用户修改密码。需要显式启用：

```bash
config user password-policy
    edit <policy_name>
        set expire-status enable
    next
end
```

GUI 行为

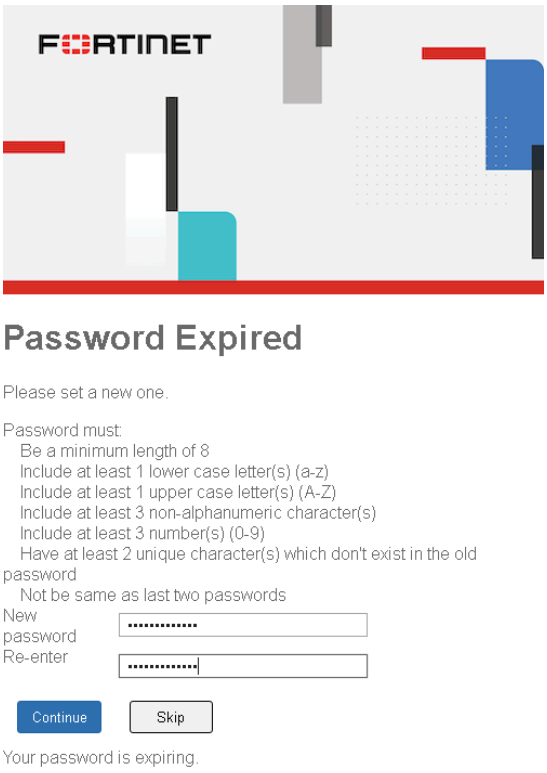
管理员修改本地用户密码

在 GUI 中进入“用户与认证 → 设置用户”，编辑本地用户并点击“修改密码”时，如果新密码不符合该用户绑定的密码策略，界面会显示对应的复杂度告警，并指出需要修正的条件。例如密码中特殊字符数量不足时，会提示需要满足 `min-non-alphanumeric` 的要求。



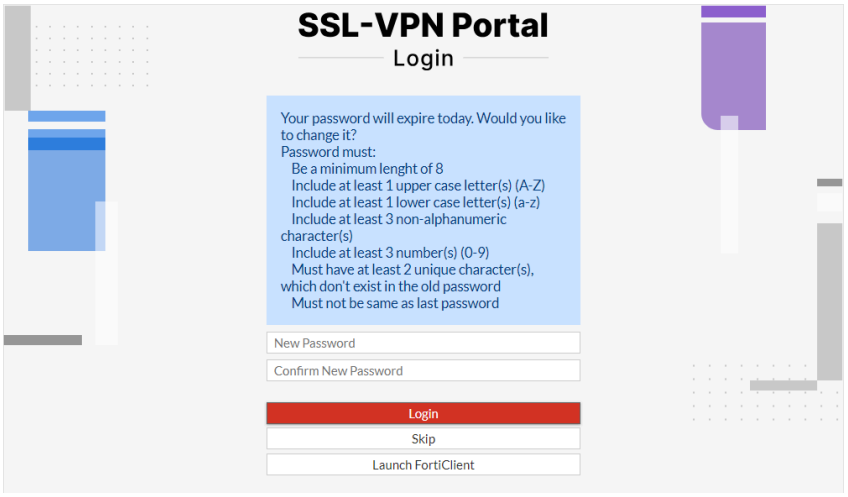
防火墙认证用户修改过期密码

用户通过防火墙策略认证/Captive Portal 认证时，如果本地用户密码已经过期，FortiGate 会提示用户修改密码。新密码提交时同样会按该用户绑定的密码策略检查。



SSL VPN 用户修改过期密码

用户通过 SSL VPN 门户登录时，如果本地用户密码已经过期，也会进入密码更新流程。新密码满足策略后，用户才能继续登录 SSL VPN 门户。


SSL-VPN Portal
Login
Your password will expire today. Would you like to change it?
Password must:
Be a minimum lenght of 8
Include at least 1 upper case letter(s) (A-Z)
Include at least 1 lower case letter(s) (a-z)
Include at least 3 non-alphanumeric character(s)
Include at least 3 number(s) (0-9)
Must have at least 2 unique character(s), which don't exist in the old password
Must not be same as last password
New Password
Confirm New Password
Login
Skip
Launch FortiClient

认证相关参数设置

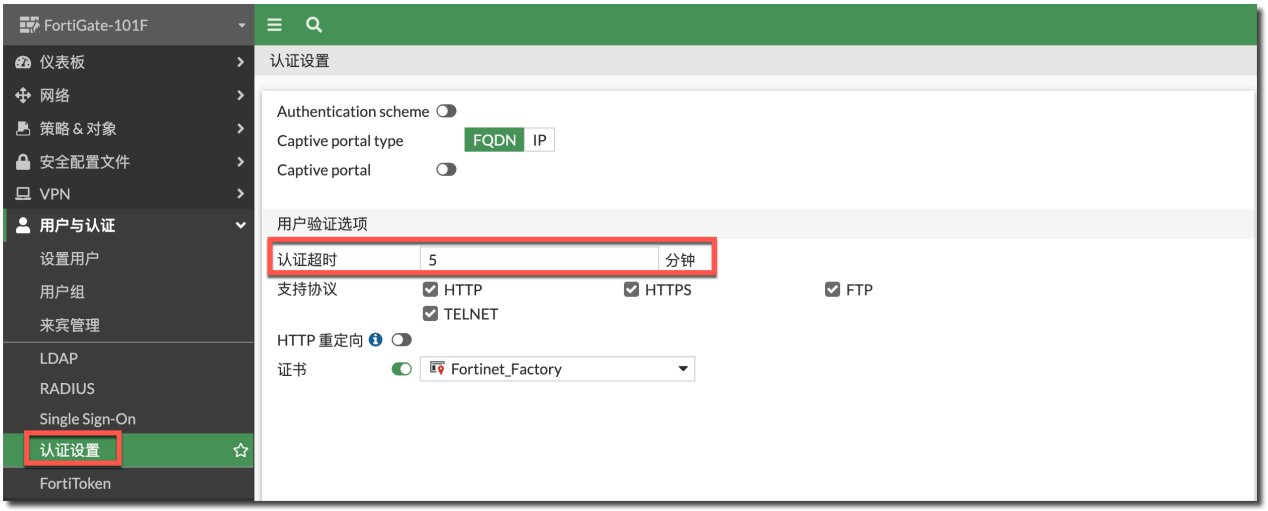
远程认证超时时间

FortiGate 远程认证（如 Radius、LDAP）超时的时间指的是与服务器认证过程的超时时间，认证过程的时间超过这个配置的值，FortiGate 就认为认证超时。默认是 5s，为了避免 Radius、LDAP 等远程服务器响应的数据大或者网络延时的影响，通常需要将远端超时时间调大。

```bash
config system global
    set remoteauthtimeout 120  <1-300s>
end
```

防火墙用户认证超时时间

1. 用户认证超时时间规定了经过防火墙身份验证的连接在空闲多久后需要必须重新进行身份验证。默认值为 5 分钟。您可以使用 1 到 1440（24 小时）之间的数字。




```bash
config user setting
    set auth-timeout ?
        auth-timeout    Enter an integer value from <1> to <1440> (default = <5>).
end
```

2. 防火墙用户认证超时时间支持以下三种方式：

```bash
config user setting
    set auth-timeout-type ?
          idle-timeout    Idle timeout.
          hard-timeout    Hard timeout.
          new-session     New session timeout.
```

- idle-timeout（默认配置）：如果 FortiGate 在配置的空闲时间内未收到该用户的流量，则清除用户在线状态。
- hard-timeout：不管用户是否有流量，都会在设定的超时时间后清除用户的在线状态。
- new-session：在配置的 auth-timeout 计时器后，将提示用户对新会话进行身份验证，已经存在的旧会话不用被重新认证。

防火墙用户认证锁定

默认情况下，防火墙用户密码连续输错，账户不会被锁定。

如果需要提高设备管理面的安全性，可以通过 CLI 调整防火墙密码的重试次数与锁定时长。

```bash
config user setting
    set auth-lockout-threshold <failed_attempts>
    set auth-lockout-duration <seconds>
end
```

- `auth-lockout-threshold`：管理员登录失败后触发锁定的阈值，默认值为 `3`，范围 1~10。

- `auth-lockout-duration`：触发锁定后的等待时间，单位为秒，默认值为 `0`（表示不锁定），范围 0~2147483647。

> **提示**
>
> - `auth-lockout-threshold` 设置得越大，防火墙用户密码被暴力猜测成功的风险也越高。
> - 如果从第一次登录失败开始，到达到锁定阈值的那次失败为止，时间间隔小于 `auth-lockout-duration`，则会触发锁定。

例如，设置管理员密码输错 1 次即锁定，锁定时间为 5 分钟。

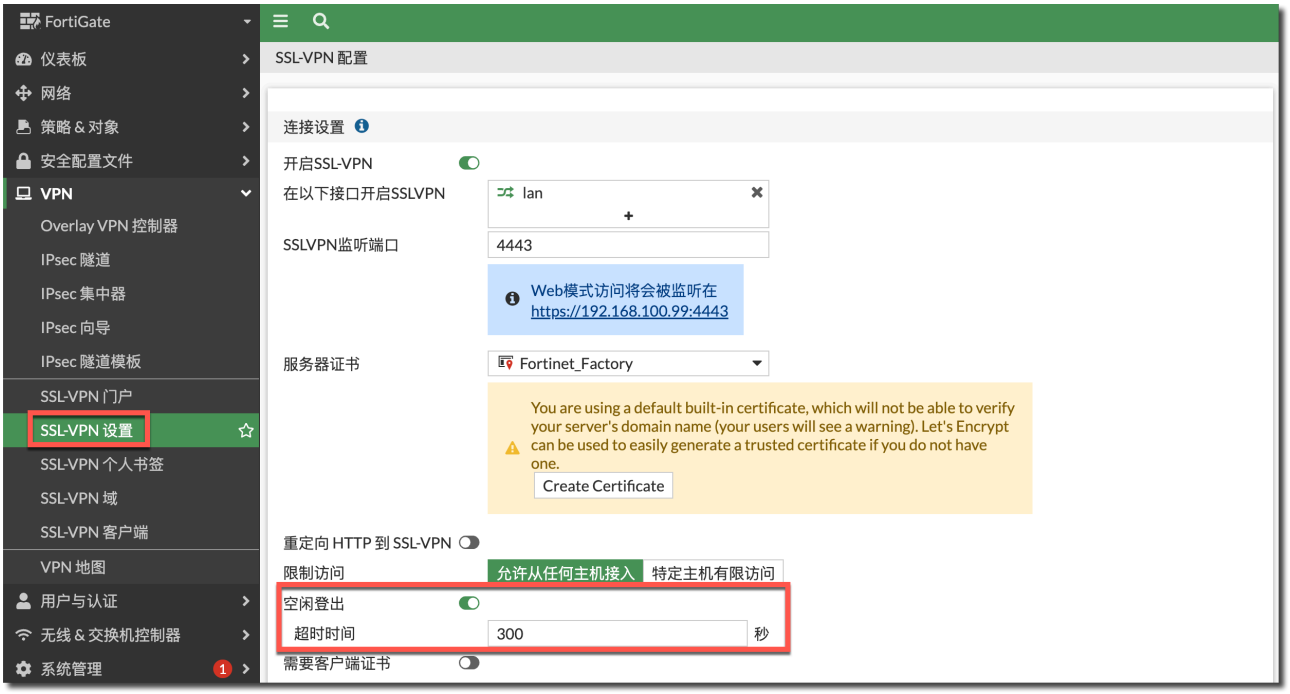
```bash
config user setting
    set auth-lockout-threshold 1
    set auth-lockout-duration 300
end
```

SSL VPN 超时时间

SSL VPN 空闲超时时间

您可以设置 SSL VPN 用户空闲超时来控制 SSL VPN 用户在登录后空闲多久后需要再次认证，最大时间为 259200 秒（3 天），默认为 300 秒。如果 FortiGate 在配置的 SSL VPN 空闲时间内未收到该 SSL VPN 用户的流量，则清除 SSL VPN 用户在线状态。

> **重要**
>
> SSL VPN 空闲超时计时期间内，有任何的流量进入 SSL VPN 隧道，空闲超时都会重置。
>
> 有时候即使客户端系统没有任何的人为操作，也可能产生一些系统流量进入 SSL VPN 隧道，导致空闲超时重置。具体请参考：https://community.fortinet.com/t5/FortiClient/Technical-Tip-SSL-VPN-Idle-timeout-not-working/ta-p/194568




```bash
config vpn ssl settings
    set idle-timeout ?
        idle-timeout    Enter an integer value from <0> to <259200> (default = <300>).
```

SSL VPN 强制重认证超时时间

SSL VPN 用户在登录后，强制重新认证的超时时间（即使客户端没有变为空闲状态）。可以将值设置为 1-259200（默认为 28800s，8 小时），设置为 0 表示不会超时。

```bash
config vpn ssl settings
    set auth-timeout ?
        auth-timeout    Enter an integer value from <0> to <259200> (default = <28800>).
```

SSL VPN 登录失败超时时间

```bash
config vpn ssl settings
    set login-attempt-limit ?
        <login_attempt_limit_integer>    The login-attempt-limit range is 0-10, 0 for no limit
    set login-block-time ?
        <login_block_time_integer>    The login-block-time range is 0-86400(seconds)
    set login-timeout ?
        login-timeout    Enter an integer value from <10> to <180> (default = <30>).
end
```

- login-attempt-limit：设置用户在账户失败锁定之前可以尝试的失败次数。请注意，此值是包容性的（默认值为 2，用户将在第 2 次登录尝试失败后被锁定），当设置为 0 时，永远不会因为登陆失败而锁定用户。
- login-block-time：设置用户在登录失败锁定后被锁定的时间量（以秒为单位，默认为 60s），设置为 0 时表示不锁定（相当于 login-attempt-limit 设置为 0 的效果）。
- login-timeout：设置多次失败的登录被认为连续的时间窗口，默认为 30s。例如，如果进行了一次失败登录尝试，然后在第 20 秒进行第 2 次失败的登录尝试，这两次将被视为连续的，因为它们在登录超时窗口内（即两次尝试之间的时间少于 30 秒），随后会触发 FortiGate 的用户锁定动作。但是，如果第二次尝试是在 31 秒后进行的，则不会被 FortiGate 视为连续尝试，也不会发生锁定。

防火墙认证不绑定 MAC 地址

默认用户通过防火墙认证后，FortiGate 会记录客户端的 MAC 地址，并将客户端认证信息与 MAC 地址绑定，当客户端 MAC 地址变化后，会导致认证失败，如果想关闭防火墙认证绑定 MAC 地址的功能（例如认证时的源 MAC 与认证成功后实际流量的源 MAC 不一致的场景），可以通过如下 CLI 实现（默认为开启状态）：

```bash
config user setting
    set auth-src-mac disable
end
```

限制认证并发源 IP 数

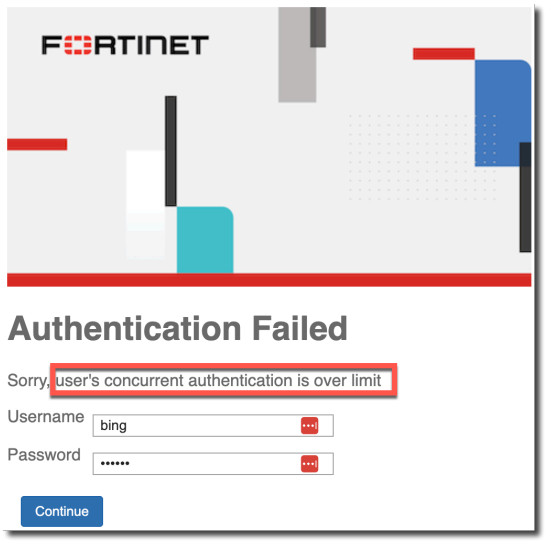
全局配置

`policy-auth-concurrent` 是 FortiGate 全局配置（`config system global`）下的一个参数，用于控制同一个用户是否允许在多个来源（Source IP）上同时保持认证状态（即同一账号从多个源 IP 同时进行防火墙策略认证/Portal 认证）。

```shell
config system global
    set policy-auth-concurrent (0-100)
end
```

- `0`（默认）：表示不限制一个用户可关联的源 IP 数量。同一用户可以同时在多个源 IP 地址上被认证（对防火墙策略认证/Captiveportal 认证而言）。
- `1` ~ `100`：
  - 一旦某用户已达到允许的源 IP 并发数上限，该用户从新的源 IP 发起的后续认证请求会被拒绝。
  - 在防火墙策略认证/Captive Portal 认证场景下，通常会看到浏览器告警/提示。
  - 举例：同一个用户如果已经在 5 个源 IP 上完成认证，再从其他源 IP 发起认证会被拒绝。

```bash
config system global
    set policy-auth-concurrent 5
end
```



覆盖全局配置

可以在用户或用户组级别覆盖全局值。当在用户或用户组级别下启用 `auth-concurrent-override` 后，`auth-concurrent-value` 的配置会覆盖 `config system global` 下的 `policy-auth-concurrent`）配置。

> **重要**
>
> 优先级：用户组 > 用户 > 全局配置

用户级别

```shell
config user local
    edit <user_name>
        set auth-concurrent-override enable
        set auth-concurrent-value (1-100)
    next
end
```

用户组级别

```shell
config user group
    edit <user_group_name>
        set auth-concurrent-override enable
        set auth-concurrent-value (1-100)
    next
end
```

注意事项

- 配置的并发源 IP 限制数是基于每 VDOM 生效的。
- 仅适用于防火墙策略认证/Captiveportal 认证，不适用于 VPN 用户、系统管理员。
- IPsec VPN 不支持限制同一用户的并发连接数（即无法用该思路实现“只允许一个 IPsec 会话”）。
- SSL VPN 可以限制同一用户同时在线会话数，但属于 SSL VPN 的单独机制（`config vpn ssl web portal` 下的 `limit-user-logins`）。
- 如果用户经常出现“认证后换网/换出口/多设备登录导致被拒绝”，请检查以上配置。
- 需要允许同用户多设备/多网络同时认证时，将其设为 `0`（无限制）或适当调大。

LDAP

LDAP 简介

LDAP（Light Directory Access Protocol），轻量级目录访问协议，是一个为查询、浏览和搜索而优化的专业分布式数据库，它成树状结构组织数据。目录服务是由目录数据库和一套访问协议组成的系统。

LDAP 组织结构

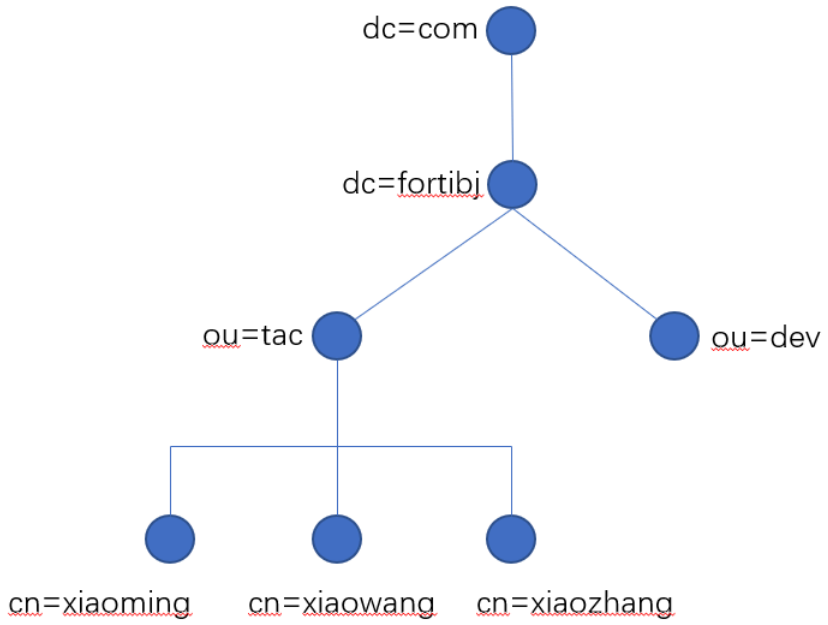
在 LDAP 中目录是按树形结构组织，而 DN（Distinguished Name）唯一可区别的名称，记录了一条记录的位置。DN 有三个属性，分别是 DC（Domain Component）、OU（Organizational Unit）与 CN（Common Name）。

DC：域控制器，类似于文件系统目录；

OU：组织单元，类似部门；

CN：区分身份的属性，类似名称。

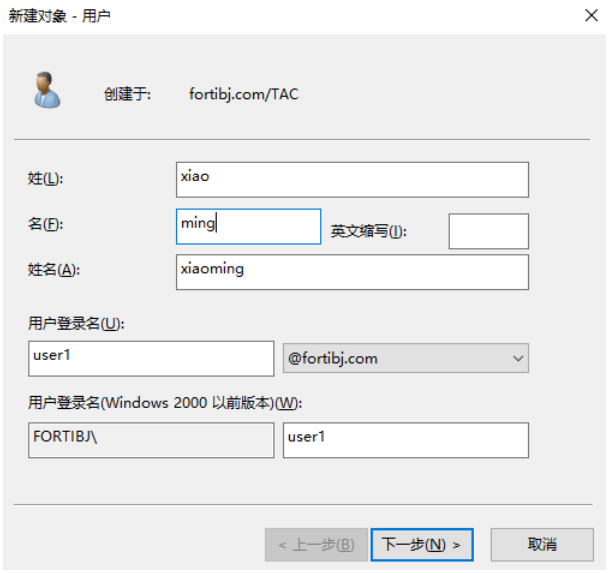
如下图所示，dn 是从树的叶子节点逐层向根节点推导出的一条唯一路径，表明了这个数据结构中的唯一一条记录。一个完整的 dn：cn = xiaoming,ou = tac,dc = fortibj,dc = com，查询得到一条完整的记录，然后再取出所需属性 (Attribute) 的值。



Windows AD 域中几个属性

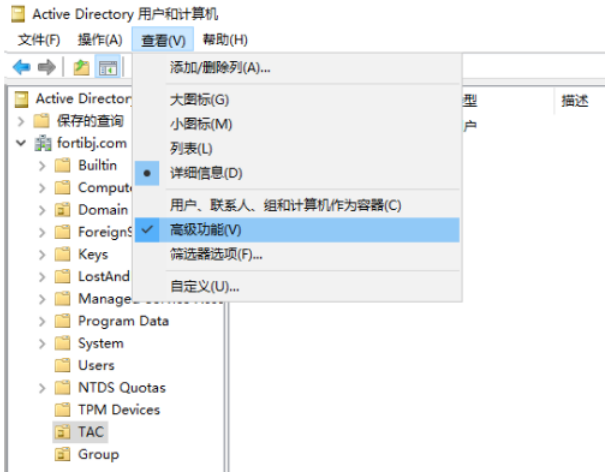
1.CN 和 sAMAccountName

Windows AD 域中创建用户时，输入信息如下：

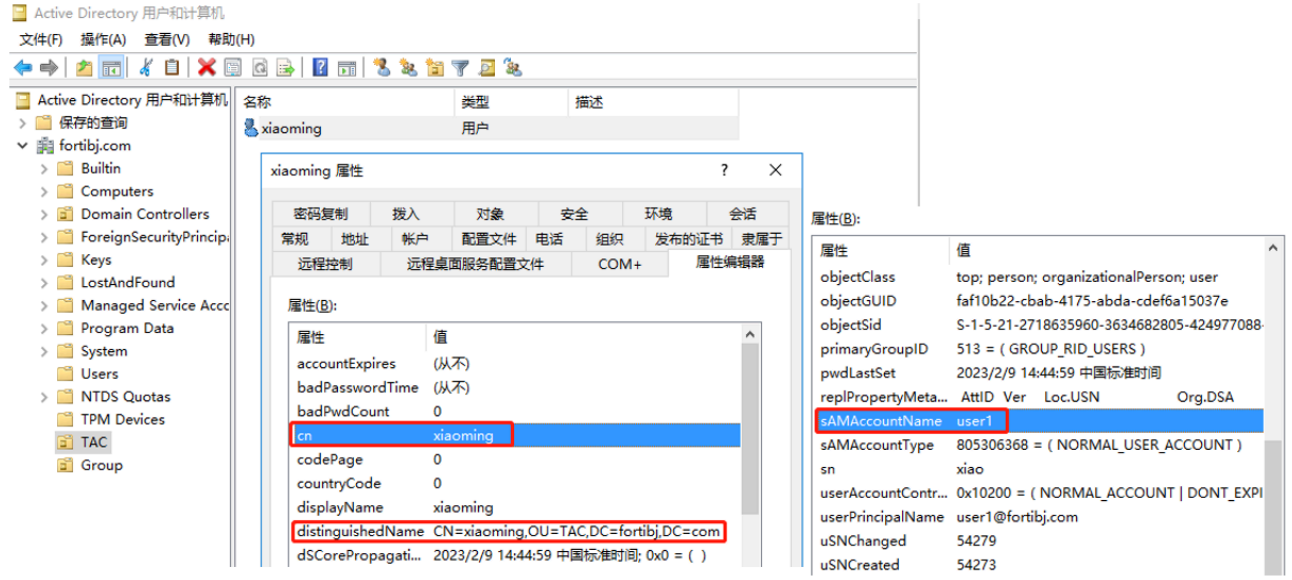


cn 表示姓名（full user name），即 xiaoming，sAMAccountName 表示用户登录名（login user name），即 user1。

在 windows Active Directory 用户和计算机中，点击”查看“，选择”高级功能“，然后可以查看更多的用户属性。



通过用户属性查询到用户 xiaoming 的 CN 是 xiaoming，sAMAccountName 是 user1，dn 是 CN = xiaoming,OU = TAC,DC = fortibj,DC = com



windows CLI 查询 DN

-根据 full name 查询

```bash
C:\Users\Administrator>dsquery user -name xiaoming
"CN=xiaoming,OU=TAC,DC=fortibj,DC=com"
```

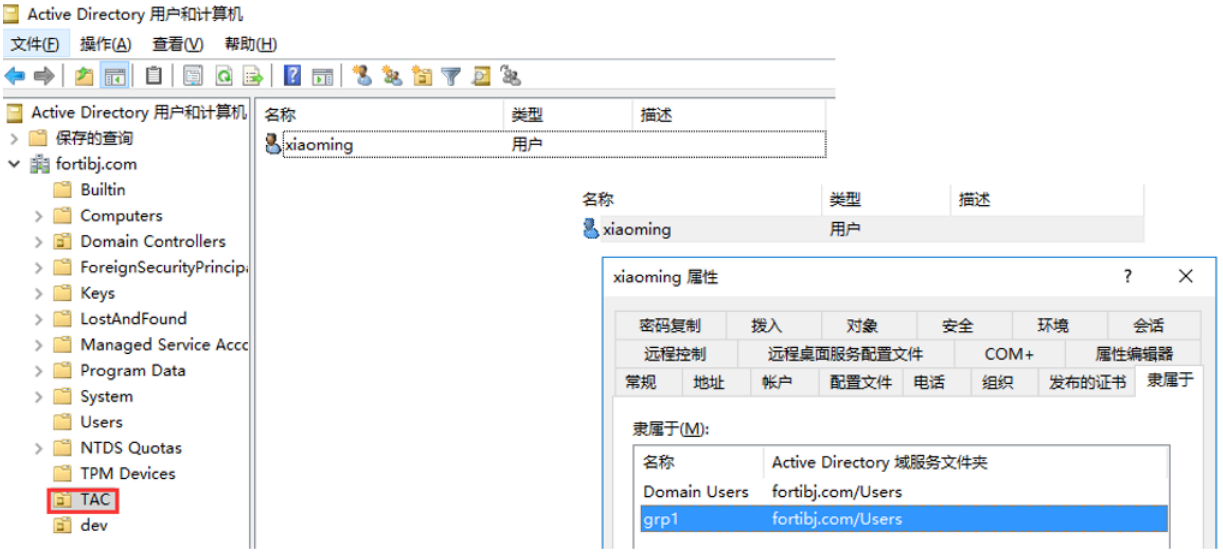
-根据 login name 查询

```bash
C:\Users\Administrator>dsquery user -samid user1
"CN=xiaoming,OU=TAC,DC=fortibj,DC=com"
```

2.OU 和 Group

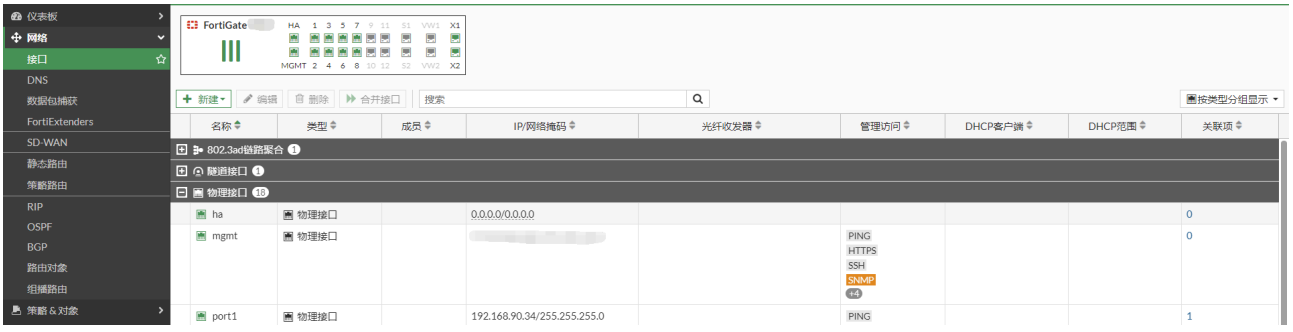
OU 是组织单元，Group 是用户组，两者不一样。

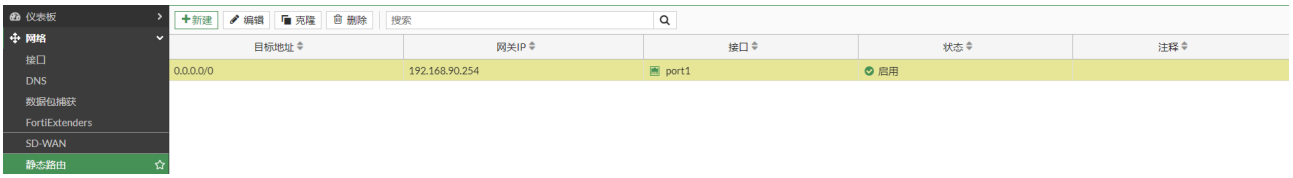
从下图可以看出用户 xiaoming 在组织单元 OU TAC 下，属于用户组 Domain Users 和 grp1。



配置 LDAP 认证

1.配置接口地址和路由





ping 测试 FortiGate 与 LDAP 服务器之间的连通性。测试环境使用 Windows AD 作为 LDAP 服务器，地址是 192.168.91.208。

```bash
# execute ping 192.168.91.208
PING 192.168.91.208 (192.168.91.208): 56 data bytes
64 bytes from 192.168.91.208: icmp_seq=0 ttl=127 time=0.3 ms
64 bytes from 192.168.91.208: icmp_seq=1 ttl=127 time=0.2 ms
64 bytes from 192.168.91.208: icmp_seq=2 ttl=127 time=0.2 ms
64 bytes from 192.168.91.208: icmp_seq=3 ttl=127 time=0.2 ms
```

```
64 bytes from 192.168.91.208: icmp_seq=4 ttl=127 time=0.2 ms

--- 192.168.91.208 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

2.配置 LDAP

选择”用户与认证“-->LDAP，点击”新建“。



配置参数：

名称：输入一个名称，不冲突即可；

服务器 IP/名称：LDAP 服务器的 IP 地址；

服务器端口：LDAP 通信端口，默认 TCP 389；

Common Name 标识符：FortiGate 用于识别连接用户的 LDAP 中对象的属性字段。Windows AD 域使用 sAMAccountName，OpenLDAP 使用 uid。这里使用 Windows AD 域；

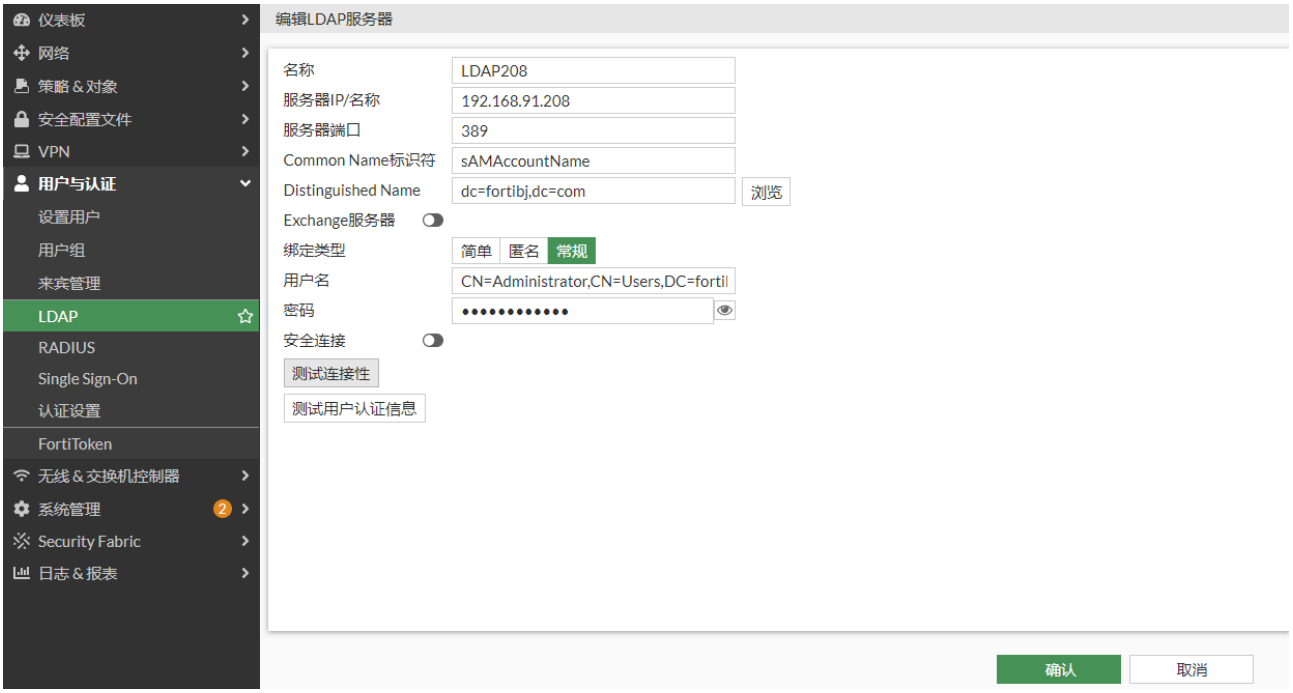
Distinguished Name：用于在 LDAP 服务器上查找用户帐户条目，反映了查找的 CN 标识符之上的 LDAP 的层次结构； 输入 dc = fortibj,dc = com 指定包含所有对象的域根； 输入 ou = tac,dc = fortibj,dc = com 查询指定组织单元下的用户；

绑定类型：选择常规，使用提供的用户名和密码进行绑定，并从 DN 开始搜索并递归遍历子树；

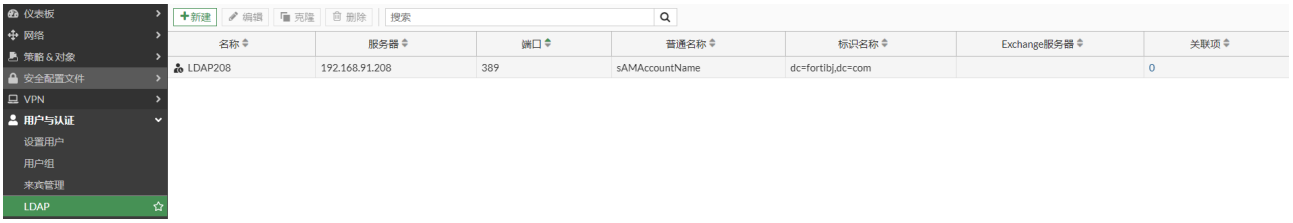
用户名：用户名支持的格式：格式 1：CN = Administrator,CN = Users,DC = fortibj,DC = com；格式 2：Administrator@fortibj.com。

```bash
C:\Users\Administrator>dsquery user -name administrator
"CN=Administrator,CN=Users,DC=fortibj,DC=com"
```

密码：用户的密码；



点击”确定“，配置完成。

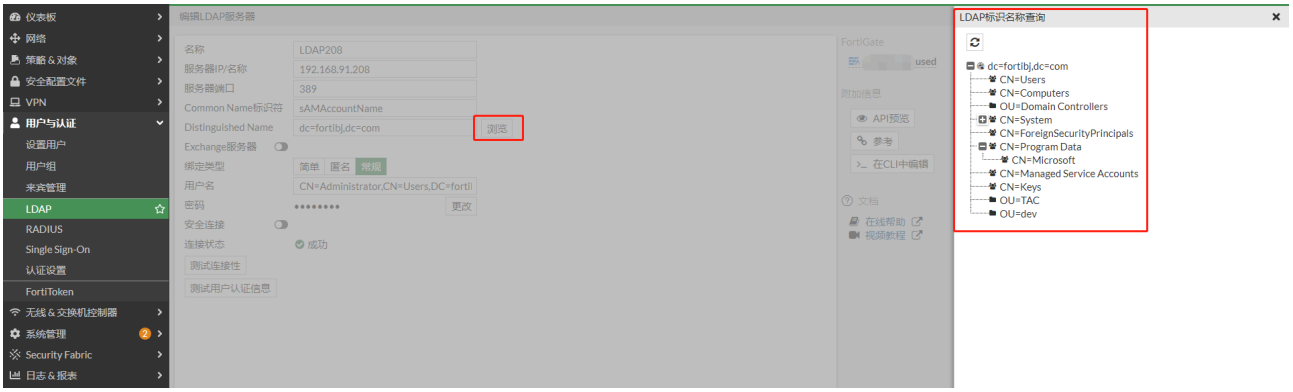


对应的命令行如下：

```bash
config user ldap
    edit "LDAP208"
        set server "192.168.91.208"
        set cnid "sAMAccountName"
        set dn "dc=fortibj,dc=com"
        set type regular
        set username "CN=Administrator,CN=Users,DC=fortibj,DC=com"
        set password xxxxxxxx
    next
end
```

3.查看 LDAP 目录结构

当 LDAP 配置正确时，可以点击”浏览“，查看 LDAP 目录结构。



LDAP 服务器测试

1.FortiGate 与 LDAP 服务器之间连通性测试

GUI 编辑 LDAP 服务器的配置界面，点击”测试连接性“。

CLI 语法测试连通性：

```bash
# diagnose test authserver ldap-direct
[IP] [port number]
```

CLI 语法测试用户和密码是否正确：

```bash
# diagnose test authserver ldap
<server_name> <username> <password>
```

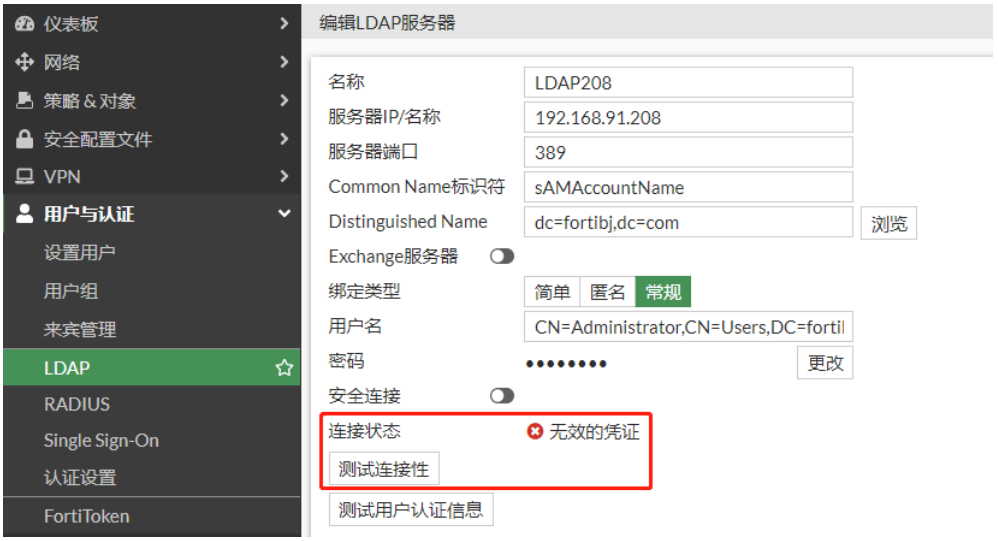
- 如果连接状态是”成功“，则 FortiGate 与 LDAP 服务器之间通信正常。



对应 CLI 的测试结果：

```bash
# diagnose test authserver ldap-direct 192.168.91.208 389
LDAP server '192.168.91.208' status is OK
```

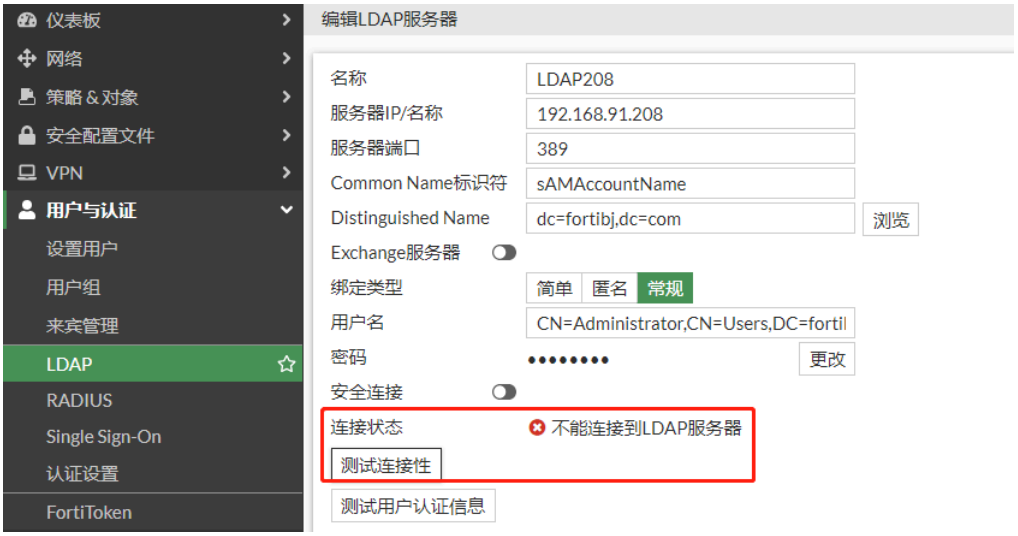
- 如果连接状态是”无效的凭证“，则说明 LDAP 服务器的账号或者密码不正确。



对应的命令行：

```bash
# diagnose test authserver ldap LDAP208 administrator Test@123456
authenticate 'administrator' against 'LDAP208' succeeded!
Group membership(s) - CN=Group Policy Creator Owners,CN=Users,DC=fortibj,DC=com
                      CN=Domain Admins,CN=Users,DC=fortibj,DC=com
                      CN=Enterprise Admins,CN=Users,DC=fortibj,DC=com
                      CN=Schema Admins,CN=Users,DC=fortibj,DC=com
                      CN=Remote Desktop Users,CN=Builtin,DC=fortibj,DC=com
                      CN=Administrators,CN=Builtin,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

- 如果连接状态是”不能连接到 LDAP 服务器“，则说明到 LDAP 服务器不可达。



对应 CLI 的测试结果：

```bash
# diagnose test authserver ldap-direct 192.168.91.208 389
LDAP server '192.168.91.208' status is Server unreachable
```

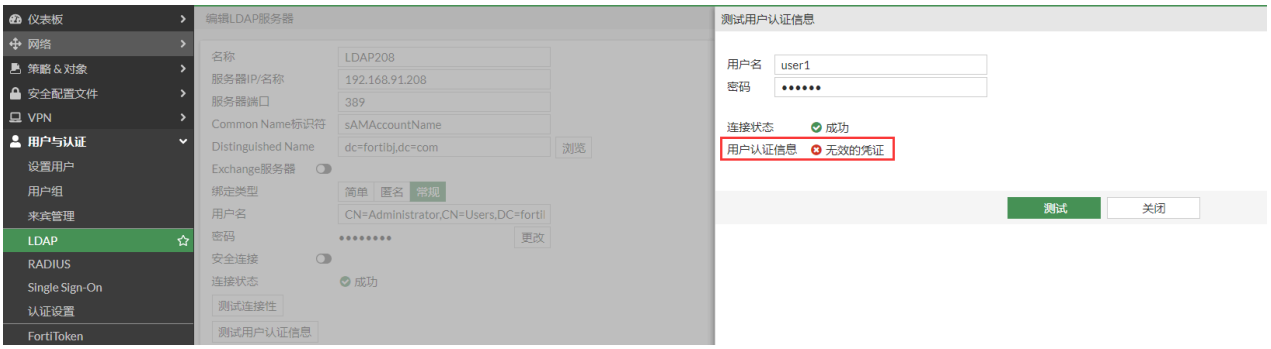
2.测试 LDAP 服务器中的用户及密码

GUI 编辑 Radius 服务器的配置界面，点击”测试用户认证信息“。

CLI 的语法：

```bash
# diagnose test authserver ldap
<server_name> <username> <password>
```

- 用户名或者密码不正确



对应的 CLI 测试结果如下：

```bash
# diagnose test authserver ldap LDAP208 user1 123456
authenticate 'user1' against 'LDAP208' failed!
```

- 用户名和密码正确



对应的 CLI 测试结果如下，CLI 显示了用户 user1 所在的用户组信息。

```bash
# diagnose test authserver ldap LDAP208 user1 Pass@123456
authenticate 'user1' against 'LDAP208' succeeded!
Group membership(s) - CN=grp1,CN=Users,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

LDAP 交互过程

1.LDAP 交互过程



2.报文交互过程分析

FortiGate 执行如下命令验证用户名和密码，获取用户组属性。

```
# diagnose test authserver ldap LDAP208 user1 Pass@123456
authenticate 'user1' against 'LDAP208' succeeded!
Group membership(s) - CN=grp1,CN=Users,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

报文交互过程如下：

- 建立 TCP 连接

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.208	TCP	74	4996 → 389 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=15603975 TSecr=0 WS=16384
2	0.000000000	192.168.91.208	192.168.90.34	TCP	74	389 → 4996 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM TSval=246458301 TSecr=15603975
3	0.000000000	192.168.90.34	192.168.91.208	TCP	66	4996 → 389 [ACK] Seq=1 Ack=1 Win=180224 Len=0 TSval=15603975 TSecr=246458301

- FortiGate 以管理员 DN 和密码为参数向 LDAP 服务器发送管理员绑定请求报文（Bind Request）获得查询权限

```
No.  Time        Source          Destination     Protocol  Length  Info
   4 0.000000000 192.168.90.34   192.168.91.208  LDAP       135    bindRequest(1) "CN=Administrator,CN=Users,DC=fortibj,DC=com" simple

> Frame 4: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 1, Ack: 1, Len: 69
v Lightweight Directory Access Protocol
  v LDAPMessage bindRequest(1) "CN=Administrator,CN=Users,DC=fortibj,DC=com" simple
      messageID: 1
    v protocolOp: bindRequest (0)
      v bindRequest
          version: 3
          name: CN=Administrator,CN=Users,DC=fortibj,DC=com
        v authentication: simple (0)
            simple: Test@123456
```

- LDAP 服务器进行绑定请求报文的处理。如果绑定成功，则向 FortiGate 发送绑定成功的回应报文

```
No.  Time        Source          Destination     Protocol  Length  Info
   5 0.000000000 192.168.91.208  192.168.90.34   LDAP        88    bindResponse(1) success

> Frame 5: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
> Transmission Control Protocol, Src Port: 389, Dst Port: 18762, Seq: 1, Ack: 70, Len: 22
v Lightweight Directory Access Protocol
  v LDAPMessage bindResponse(1) success
      messageID: 1
    v protocolOp: bindResponse (1)
      v bindResponse
          resultCode: success (0)
          matchedDN:
          errorMessage:
```

- FortiGate 以输入的用户名为参数，向 LDAP 服务器发送用户 DN 查询请求报文（Search Request）

```
No.  Time        Source          Destination     Protocol  Length  Info
   7 0.000000000 192.168.90.34   192.168.91.208  LDAP       139    searchRequest(2) "dc=fortibj,dc=com" wholeSubtree

> Frame 7: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 70, Ack: 23, Len: 73
v Lightweight Directory Access Protocol
  v LDAPMessage searchRequest(2) "dc=fortibj,dc=com" wholeSubtree
      messageID: 2
    v protocolOp: searchRequest (3)
      v searchRequest
          baseObject: dc=fortibj,dc=com
          scope: wholeSubtree (2)
          derefAliases: neverDerefAliases (0)
          sizeLimit: 0
          timeLimit: 0
          typesOnly: True
        v Filter: (sAMAccountName=user1)
          v filter: equalityMatch (3)
            v equalityMatch
                attributeDesc: sAMAccountName
                assertionValue: user1
        v attributes: 1 item
            AttributeDescription: 1.1
```

- LDAP 服务器收到查询请求报文后，对用户 DN 进行查找。如果查询成功，则向 FortiGate 发送查询成功的回应报文，返回用户的 DN

```
![image-20230210145716681](/images/image-20230210145716681.png)
```

- FortiGate 以查询得到的用户 DN 和用户输入的密码为参数，向 LDAP 服务器发送用户 DN 绑定请求报文（Bind Request），检查用户密码是否正确

```
No.   Time          Source          Destination       Protocol  Length  Info
    9 0.000000000   192.168.90.34   192.168.91.208    LDAP       159 bindRequest(3) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" simple

> Frame 9: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 143, Ack: 346, Len: 93
v Lightweight Directory Access Protocol
  v LDAPMessage bindRequest(3) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" simple
      messageID: 3
    v protocolOp: bindRequest (0)
      v bindRequest
          version: 3
          name: CN=xiaoming,OU=TAC,DC=fortibj,DC=com
        v authentication: simple (0)
            simple: Pass@123456
```

- LDAP 服务器进行绑定请求报文的处理，如果绑定成功，则向 FortiGate 发送绑定成功的回应报文；如果绑定失败，则向 FortiGate 发送绑定失败的回应报文

```
No.   Time          Source          Destination       Protocol  Length  Info
   10 0.000000000   192.168.91.208  192.168.90.34     LDAP        88 bindResponse(3) success

> Frame 10: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
> Transmission Control Protocol, Src Port: 389, Dst Port: 18762, Seq: 346, Ack: 236, Len: 22
v Lightweight Directory Access Protocol
  v LDAPMessage bindResponse(3) success
      messageID: 3
    v protocolOp: bindResponse (1)
      v bindResponse
          resultCode: success (0)
          matchedDN:
          errorMessage:
```

- 如果绑定成功，FortiGate 向 LDAP 服务器发送用户属性查询请求报文（Search Request），获取用户所属的用户组

  memberof：用户所属的组的 DN。

  primaryGroupID：指定用户的主组的相对标识符 (RID)。默认情况下是 Domain Users 组的 RID，用户是其主组的成员，Domain User 没有列在用户的 memberOf 属性中。因此需要单独查询主组的 DN。

  objectSid：指定安全主体对象的安全标识符 (SID)。SID 是用于标识安全主体对象的唯一值。

```
No.   Time          Source          Destination       Protocol  Length  Info
   11 0.000000000   192.168.90.34   192.168.91.208    LDAP       178 searchRequest(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" baseObject

> Frame 11: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 236, Ack: 368, Len: 112
v Lightweight Directory Access Protocol
  v LDAPMessage searchRequest(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" baseObject
      messageID: 4
    v protocolOp: searchRequest (3)
      v searchRequest
          baseObject: CN=xiaoming,OU=TAC,DC=fortibj,DC=com
          scope: baseObject (0)
          derefAliases: neverDerefAliases (0)
          sizeLimit: 0
          timeLimit: 0
          typesOnly: False
        v Filter: (objectclass=*)
          v filter: present (7)
              present: objectclass
        v attributes: 3 items
            AttributeDescription: memberOf
            AttributeDescription: primaryGroupID
            AttributeDescription: objectSid
```

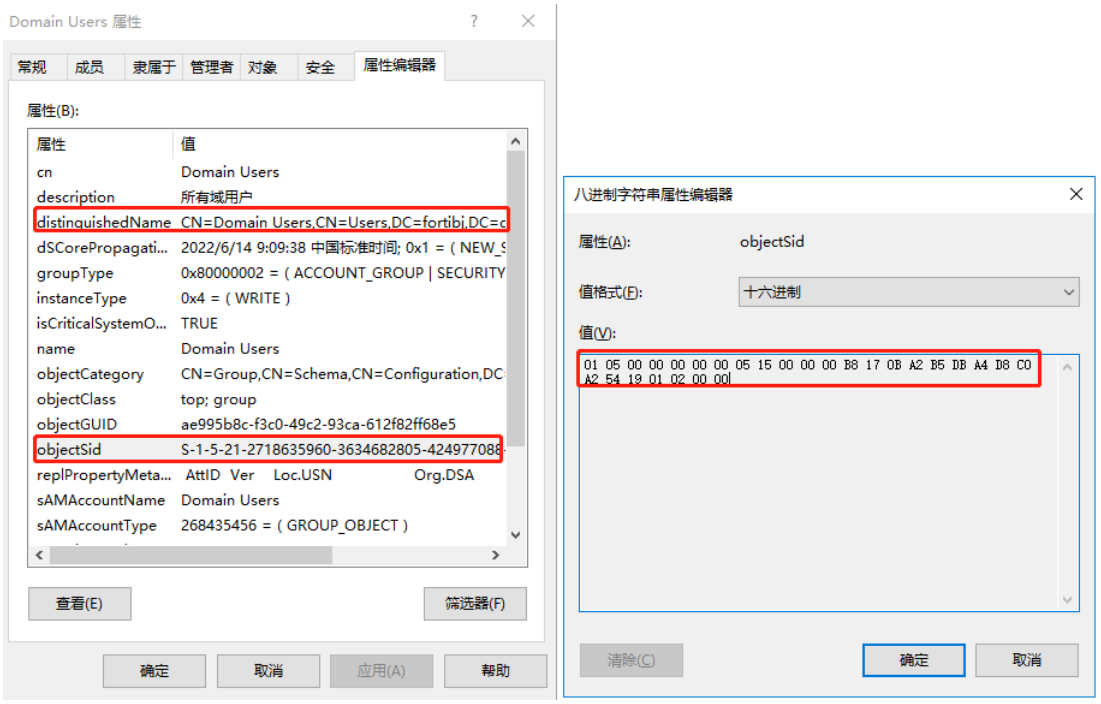
- LDAP 服务器对查询请求报文进行处理，返回用户所在的用户组

```
No.  Time          Source          Destination     Protocol  Length  Info
  12 0.000000000  192.168.91.208  192.168.90.34   LDAP         291  searchResEntry(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" | searchResDone(4) success [3 results]

> Frame 12: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
> Transmission Control Protocol, Src Port: 389, Dst Port: 18762, Seq: 368, Ack: 348, Len: 225
v Lightweight Directory Access Protocol
  v LDAPMessage searchResEntry(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" [3 results]
      messageID: 4
    v protocolOp: searchResEntry (4)
      v searchResEntry
          objectName: CN=xiaoming,OU=TAC,DC=fortibj,DC=com
        v attributes: 3 items
          v PartialAttributeList item memberOf
              type: memberOf
            v vals: 1 item
                AttributeValue: CN=grp1,CN=Users,DC=fortibj,DC=com
          v PartialAttributeList item primaryGroupID
              type: primaryGroupID
            v vals: 1 item
                AttributeValue: 513
          v PartialAttributeList item objectSid
              type: objectSid
            v vals: 1 item
              v SID: S-1-5-21-2718635960-3634682805-424977088-2104  (Domain SID-Domain RID)
                  Revision: 1
                  Num Auth: 5
                  Authority: 5
                  Subauthorities: 21-2718635960-3634682805-424977088-2104
                  RID: 2104  (Domain RID)
      [Response To: 11]
      [Time: 0.000000000 seconds]
v Lightweight Directory Access Protocol
  v LDAPMessage searchResDone(4) success [3 results]
      messageID: 4
    v protocolOp: searchResDone (5)
      v searchResDone
          resultCode: success (0)
          matchedDN:
          errorMessage:
```

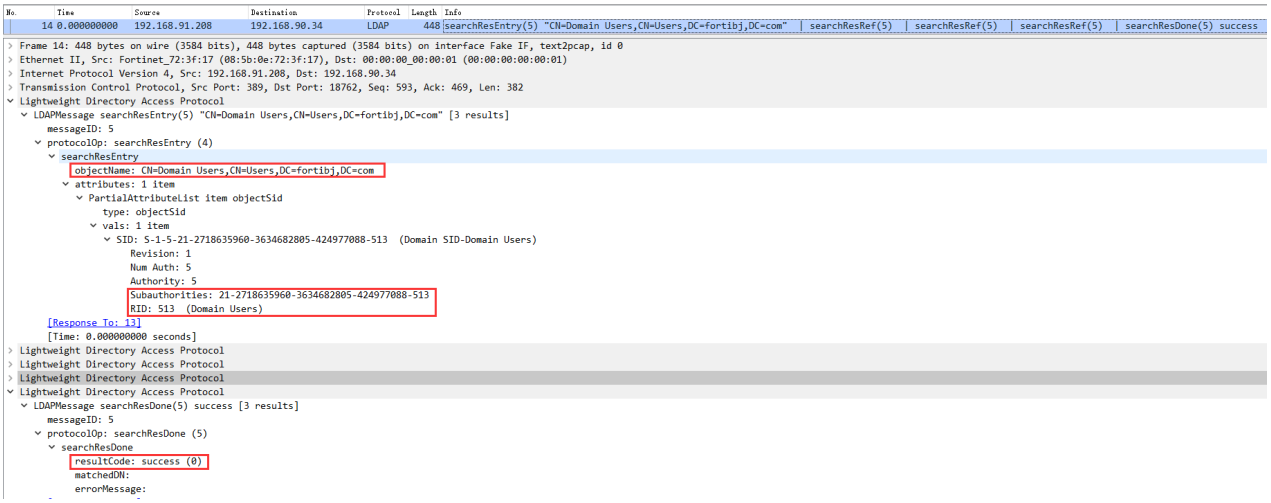
- FortiGate 向 LDAP 服务器发送用户属性查询请求报文（Search Request），获取用户所在的主组

```
No.  Time          Source          Destination     Protocol  Length  Info
  13 0.000000000  192.168.90.34   192.168.91.208  LDAP         187  searchRequest(5) "dc=fortibj,dc=com" wholeSubtree

> Frame 13: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 348, Ack: 593, Len: 121
v Lightweight Directory Access Protocol
  v LDAPMessage searchRequest(5) "dc=fortibj,dc=com" wholeSubtree
      messageID: 5
    v protocolOp: searchRequest (3)
      v searchRequest
          baseObject: dc=fortibj,dc=com
          scope: wholeSubtree (2)
          derefAliases: neverDerefAliases (0)
          sizeLimit: 0
          timeLimit: 0
          typesOnly: False
        v Filter: (&(objectclass=group)(objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…))
          v filter: and (0)
            v and: (&(objectclass=group)(objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…))
              v and: 2 items
                v Filter: (objectclass=group)
                  v and item: equalityMatch (3)
                    v equalityMatch
                        attributeDesc: objectclass
                        assertionValue: group
                v Filter: (objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…)
                  v and item: equalityMatch (3)
                    v equalityMatch
                        attributeDesc: objectSid
                        assertionValue: 01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…
        v attributes: 1 item
            AttributeDescription: objectSid

0000  00 00 00 00 00 00 e8 1c  ba ee f2 04 08 00 45 00   ........ ......E·
0010  00 ad 25 15 00 00 40 06  1d f3 c0 a8 5a 22 c0 a8   ··%···@· ····Z"··
0020  5b d0 49 4a 01 85 92 0d  7b 26 66 83 9a 91 80 18   [·IJ···· {&f·····
0030  00 0b e6 55 00 00 01 01  08 0a 00 8a d7 ac 0a d0   ···U···· ········
0040  16 f0 30 77 02 01 05 63  72 04 11 64 63 3d 66 6f   ··0w···c r··dc=fo
0050  72 74 69 62 6a 2c 64 63  3d 63 6f 6d 0a 01 02 0a   rtibj,dc =com····
0060  01 00 02 01 00 02 01 00  01 01 00 a0 41 a3 14 04   ........ ····A···
0070  0b 6f 62 6a 65 63 74 63  6c 61 73 73 04 05 67 72   ·objectc lass··gr
0080  6f 75 70 a3 29 04 09 6f  62 6a 65 63 74 53 69 64   oup·)··o bjectSid
0090  04 1c 01 05 00 00 00 00  00 05 15 00 00 00 b8 17   ·····... ........
00a0  0b a2 b5 db a4 d8 c0 a2  54 19 01 02 00 00 30 0b   ........ T·····0·
00b0  04 09 6f 62 6a 65 63 74  53 69 64                 ··object Sid
```

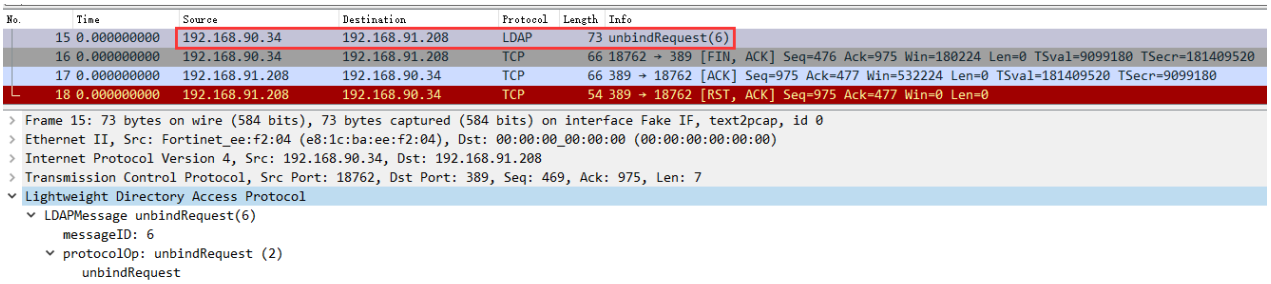
LDAP 服务器中 Domain User objectSid 的值与上图 assertionValue 的值一致。



- LDAP 服务器对查询请求报文进行处理，返回用户所在主组的 DN



- FortiGate 解除绑定



常见的标准的 LDAP 响应码

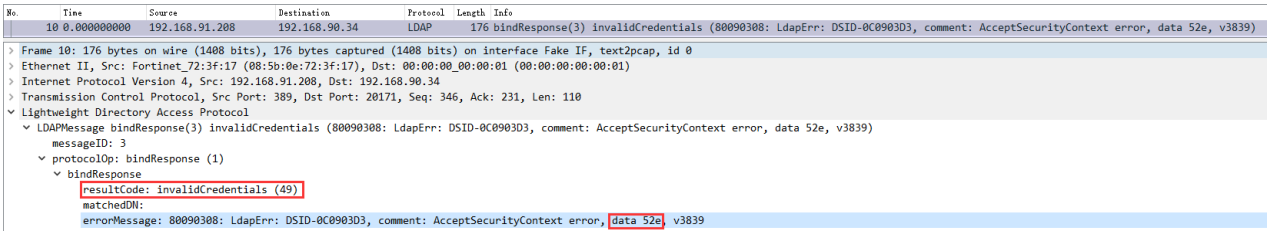
1.常见的标准的 LDAP 响应码

LDAP Result Code	Description
0	SUCCESS
2	PROTOCOL ERROR
7	AUTH METHOD NOT SUPPORTED
16	NO SUCH ATTRIBUTE
21	INVALID SYNTAX
32	NO SUCH OBJECT
34	INVALID DN SYNTAX
49	INVALID CREDENTIALS/ACCOUNT DISABLED
50	LDAP INSUFFICIENT ACCESS

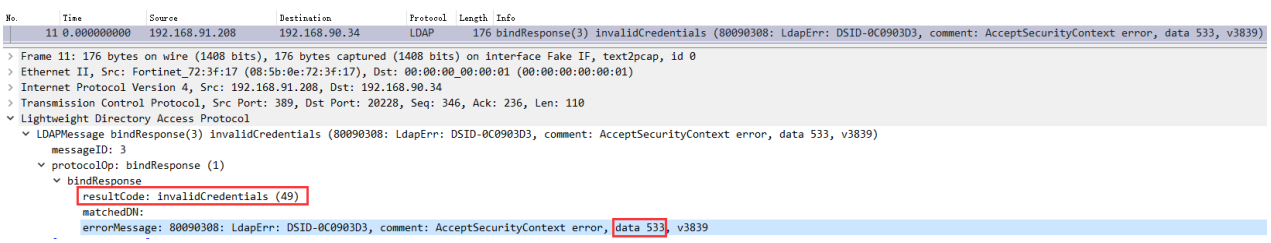
LDAP Error Message Data	Description
0x525	user not found
0x52e	invalid credentials
0x530	not permitted to logon at this time
0x531	not permitted to logon from this workstation
0x532	password expired
0x533	account disabled
0x701	account expired
0x773	user must reset password
0x775	account locked out

2.错误码抓包举例

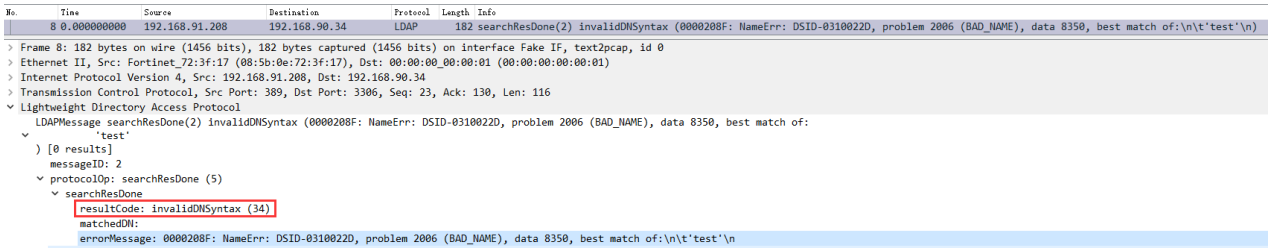
1. 密码错误



2. 账号禁用



3. DN 语法错误



LDAP 认证 debug

认证成功

1. 准备查询的用户名 user1 和 base DN。

```bash
# diagnose debug application fnbamd -1
# diagnose debug  enable

[1906] handle_req-Rcvd auth req 730526726 for user1 in LDAP208 opt=0000001b prot=0
[466] __compose_group_list_from_req-Group 'LDAP208', type 1
[616] fnbamd_pop3_start-user1
[989] __fnbamd_cfg_get_ldap_list_by_server-
[995] __fnbamd_cfg_get_ldap_list_by_server-Loaded LDAP server 'LDAP208'
[1150] fnbamd_cfg_get_ldap_list-Total ldap servers to try: 1

[1717] fnbamd_ldap_init-search filter is: sAMAccountName=user1
[1727] fnbamd_ldap_init-search base is: dc=fortibj,dc=com

[1149] __fnbamd_ldap_dns_cb-Resolved LDAP208:192.168.91.208 to 192.168.91.208, cur stack size:1
[924] __fnbamd_ldap_get_next_addr-
[1154] __fnbamd_ldap_dns_cb-Connection starts LDAP208:192.168.91.208, addr 192.168.91.208
[879] __fnbamd_ldap_start_conn-Still connecting 192.168.91.208.
[633] create_auth_session-Total 1 server(s) to try
[1107] __ldap_connect-tcps_connect(192.168.91.208) is established.
```

2. LDAP 管理员账号绑定。

```bash
[985] __ldap_rxtx-state 3(Admin Binding)
[363] __ldap_build_bind_req-Binding to 'CN=Administrator,CN=Users,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 69 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 1

[985] __ldap_rxtx-state 4(Admin Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:1, type:bind
```

3. 绑定成功。

```bash
response-ret=0
[1023] fnbamd_ldap_parse_response-ret=0
```

4. 查询 user1 的 DN，查询成功，获得 user1 的 DN 是 xiaoming,OU = TAC,DC = fortibj,DC = com。

```bash
[1052] __ldap_rxtx-Change state to 'DN search'
[985] __ldap_rxtx-state 11(DN search)
[750] fnbamd_ldap_build_dn_search_req-base:'dc=fortibj,dc=com' filter:sAMAccountName=user1
[1083] fnbamd_ldap_send-sending 73 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 2

[985] __ldap_rxtx-state 12(DN search resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 51
[1306] fnbamd_ldap_recv-Response len: 53, svr: 192.168.91.208
```

```
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-entry

response-ret=0
[1023] fnbamd_ldap_parse_response-ret=0
[1225] __fnbamd_ldap_dn_entry-Get DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
```

5. 用户 user1 绑定请求。

```bash
[1052] __ldap_rxtx-Change state to 'User Binding'
[985] __ldap_rxtx-state 5(User Binding)
[596] fnbamd_ldap_build_userbind_req-Trying DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[363] __ldap_build_bind_req-Binding to 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 93 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 3

[985] __ldap_rxtx-state 6(User Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:3, type:bind
```

6. 绑定成功。

```bash
response-ret=0
[1023] fnbamd_ldap_parse_response-ret=0
```

7. memberOf 属性查询。

```bash
[1052] __ldap_rxtx-Change state to 'Attr query'
[985] __ldap_rxtx-state 7(Attr query)
[649] fnbamd_ldap_build_attr_search_req-Adding attr 'memberOf'
[661] fnbamd_ldap_build_attr_search_req-base:'CN=xiaoming,OU=TAC,DC=fortibj,DC=com' filter:cn=*
[1083] fnbamd_ldap_send-sending 112 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 4

[985] __ldap_rxtx-state 8(Attr query resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 195
[1306] fnbamd_ldap_recv-Response len: 197, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:4, type:search-entry
```

8. 查询成功，获得 user1 的用户组 CN = grp1,CN = Users,DC = fortibj,DC = com。

```bash
[1023] fnbamd_ldap_parse_response-ret=0
[556] __get_member_of_groups-Get the memberOf groups.
[522] __retrieve_group_values-Get the memberOf groups.
[532] __retrieve_group_values- attr='memberOf', found 1 values
[542] __retrieve_group_values-val[0]='CN=grp1,CN=Users,DC=fortibj,DC=com'

[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:4, type:search-result
[1023] fnbamd_ldap_parse_response-ret=0
[1305] __fnbamd_ldap_attr_next-Entering CHKPRIMARYGRP state
```

9. Primary group 属性查询。

```bash
[1052] __ldap_rxtx-Change state to 'Primary group query'
[985] __ldap_rxtx-state 13(Primary group query)
[685] fnbamd_ldap_build_primary_grp_search_req-starting primary group check ...
[689] fnbamd_ldap_build_primary_grp_search_req-number of sub auths 5
[707] fnbamd_ldap_build_primary_grp_search_req-base:'dc=fortibj,dc=com' filter:(&(objectclass=group)
(objectSid=\01\05\00\00\00\00\00\05\15\00\00\00\b8\17\0b\a2\b5\db\a4\d8\c0\a2\54\19\01\02\00\00))
[1083] fnbamd_ldap_send-sending 121 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 5
```

```
[985] __ldap_rxtx-state 14(Primary group query resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 110
[1306] fnbamd_ldap_recv-Response len: 112, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-entry
```

10. 获得 Primary group 是 CN = Domain Users,CN = Users,DC = fortibj,DC = com。

```
[1023] fnbamd_ldap_parse_response-ret=0
[472] __get_one_group-group: CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

11. 查询完成。

```
[1052] __ldap_rxtx-Change state to 'Done'
[985] __ldap_rxtx-state 23(Done)
[1083] fnbamd_ldap_send-sending 7 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 6
[785] __ldap_done-svr 'LDAP208'
[755] __ldap_destroy-
[724] __ldap_stop-Conn with 192.168.91.208 destroyed.
```

12. 查询的结果。

```
[2678] fnbamd_ldap_result-Result for ldap svr 192.168.91.208(LDAP208) is SUCCESS
[401] ldap_copy_grp_list-copied CN=grp1,CN=Users,DC=fortibj,DC=com
[401] ldap_copy_grp_list-copied CN=Domain Users,CN=Users,DC=fortibj,DC=com

[2693] fnbamd_ldap_result-Skipping group matching
[216] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 730526726, len=2227
[789] destroy_auth_session-delete session 730526726
[755] __ldap_destroy-
[1764] fnbamd_ldap_auth_ctx_free-Freeing 'LDAP208' ctx
```

认证失败

1. 准备查询的用户名 user1 和 base DN。

```
# diagnose debug application fnbamd -1
# diagnose debug  enable

[1906] handle_req-Rcvd auth req 730526741 for user1 in LDAP208 opt=0000001b prot=0
[466] __compose_group_list_from_req-Group 'LDAP208', type 1
[616] fnbamd_pop3_start-user1
[989] __fnbamd_cfg_get_ldap_list_by_server-
[995] __fnbamd_cfg_get_ldap_list_by_server-Loaded LDAP server 'LDAP208'
[1150] fnbamd_cfg_get_ldap_list-Total ldap servers to try: 1

[1717] fnbamd_ldap_init-search filter is: sAMAccountName=user1
[1727] fnbamd_ldap_init-search base is: dc=fortibj,dc=com

[1149] __fnbamd_ldap_dns_cb-Resolved LDAP208:192.168.91.208 to 192.168.91.208, cur stack size:1
[924] __fnbamd_ldap_get_next_addr-
[1154] __fnbamd_ldap_dns_cb-Connection starts LDAP208:192.168.91.208, addr 192.168.91.208
[879] __fnbamd_ldap_start_conn-Still connecting 192.168.91.208.
[633] create_auth_session-Total 1 server(s) to try
[1107] __ldap_connect-tcps_connect(192.168.91.208) is established.
```

2. LDAP 管理员账号绑定。

```
[985] __ldap_rxtx-state 3(Admin Binding)
[363] __ldap_build_bind_req-Binding to 'CN=Administrator,CN=Users,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 69 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 1

[985] __ldap_rxtx-state 4(Admin Bind resp)
```

```
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:1, type:bind
```

3. 绑定成功。

```bash
response-ret=0
[1023] fnbamd_ldap_parse_response-ret=0
```

4. 查询 user1 的 DN。

```bash
[1052] __ldap_rxtx-Change state to 'DN search'
[985] __ldap_rxtx-state 11(DN search)
[750] fnbamd_ldap_build_dn_search_req-base:'dc=fortibj,dc=com' filter:sAMAccountName=user1
[1083] fnbamd_ldap_send-sending 73 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 2

[985] __ldap_rxtx-state 12(DN search resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 51
[1306] fnbamd_ldap_recv-Response len: 53, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-entry
```

5. 查询成功，获得 user1 的 DN 是 xiaoming,OU = TAC,DC = fortibj,DC = com。

```bash
response-ret=0
[1023] fnbamd_ldap_parse_response-ret=0
[1225] __fnbamd_ldap_dn_entry-Get DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
```

6. 用户 user1 绑定请求。

```bash
[1052] __ldap_rxtx-Change state to 'User Binding'
[985] __ldap_rxtx-state 5(User Binding)
[596] fnbamd_ldap_build_userbind_req-Trying DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[363] __ldap_build_bind_req-Binding to 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 88 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 3

[985] __ldap_rxtx-state 6(User Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 102
[1306] fnbamd_ldap_recv-Response len: 104, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:3, type:bind
```

7. 绑定失败，响应码 49，响应消息 52e，即密码不对。

```bash
[1009] fnbamd_ldap_parse_response-Error 49(80090308: LdapErr: DSID-0C0903D3, comment: AcceptSecurityContext error, data
52e, v3839)
[1023] fnbamd_ldap_parse_response-ret=49

[262] fnbamd_ldap_more_dn_left-idx:0, total:1
[1052] __ldap_rxtx-Change state to 'Done'
[985] __ldap_rxtx-state 23(Done)
[1083] fnbamd_ldap_send-sending 7 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 4
[785] __ldap_done-svr 'LDAP208'
[755] __ldap_destroy-
[724] __ldap_stop-Conn with 192.168.91.208 destroyed.
[216] fnbamd_comm_send_result-Sending result 1 (nid 0) for req 730526741, len=2148
[789] destroy_auth_session-delete session 730526741
[755] __ldap_destroy-
[1764] fnbamd_ldap_auth_ctx_free-Freeing 'LDAP208' ctx
```

其他 LDAP 配置参数

1.修改 LDAP 端口

```bash
config user ldap
    edit LDAP208
        set port xx
    next
end
```

2.指定原 IP 地址

```bash
config user ldap
    edit LDAP208
        set source-ip x.x.x.x
    next
end
```

创建 ldap 用户

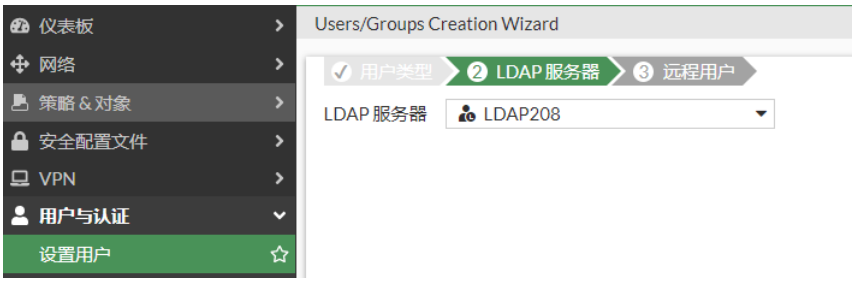
1. 选择“用户与认证”-->“设置用户”，点击新建



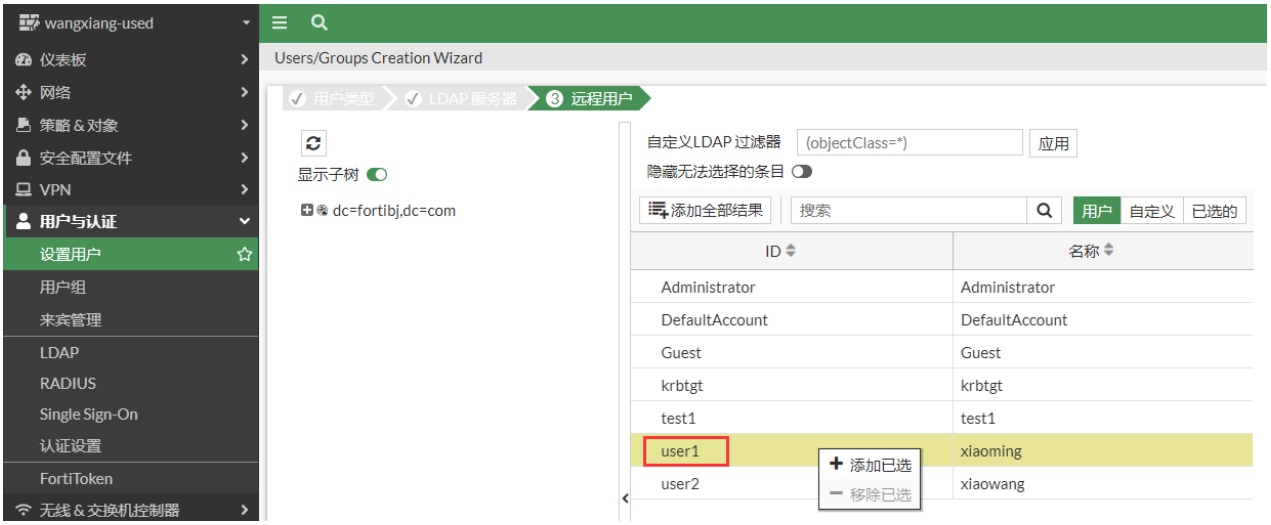
2. 选择 LDAP 用户，点击“下一步”



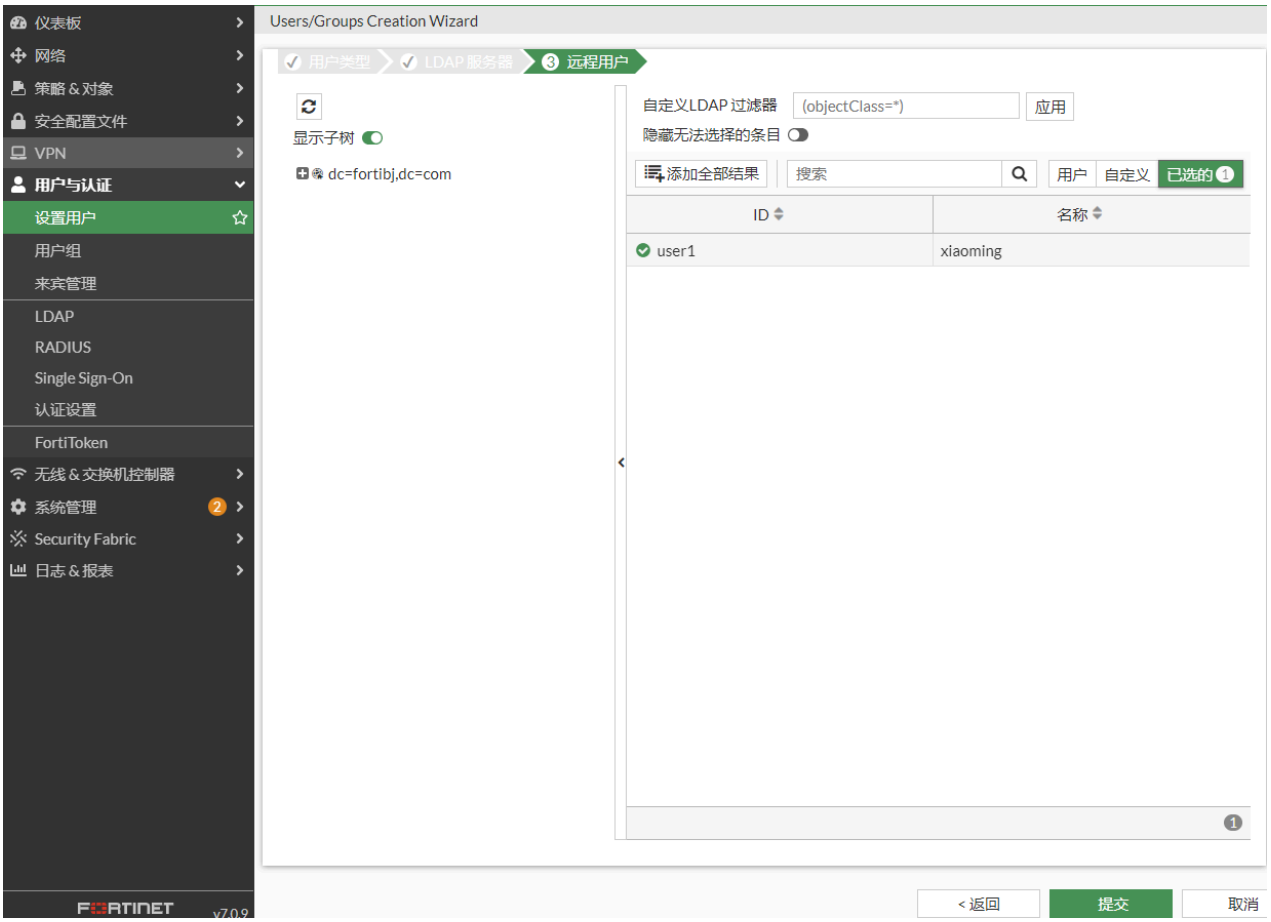
3. 选择配置好的 LDAP 服务器，点击“下一步”



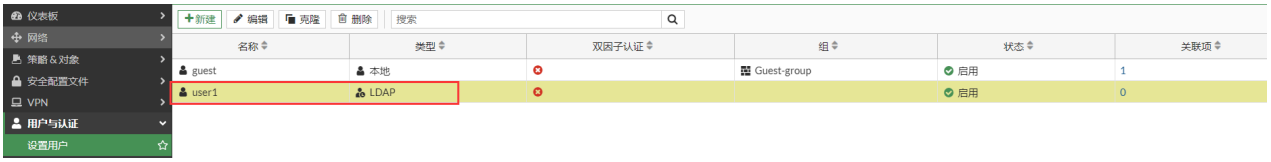
4. 这里添加 user1，右击 user1，点击“添加已选”



点击“已选的”，查看选中的用户，确认无误后，点击“提交”。

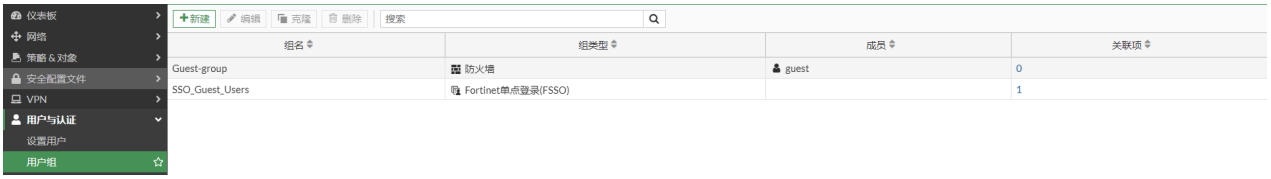


5. 完成 LDAP 用户创建



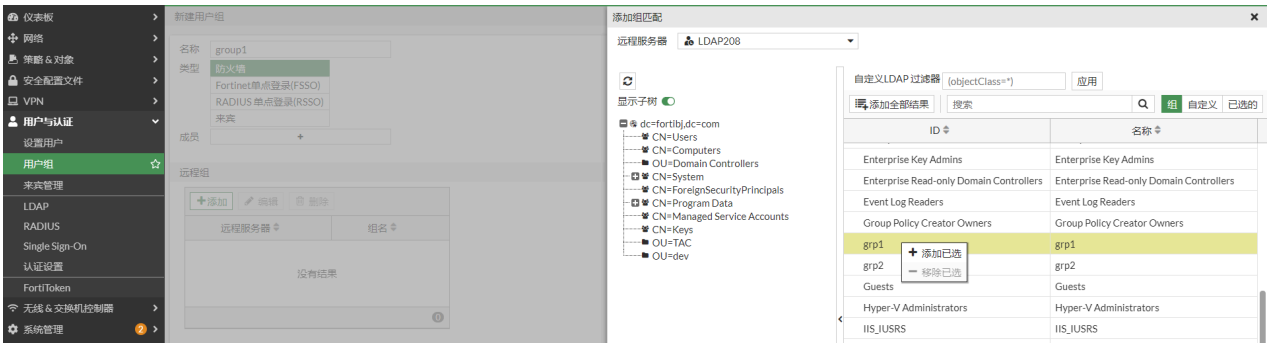
创建 ldap 用户组

1. 选择“用户与认证”-->“用户组”，点击“新建”

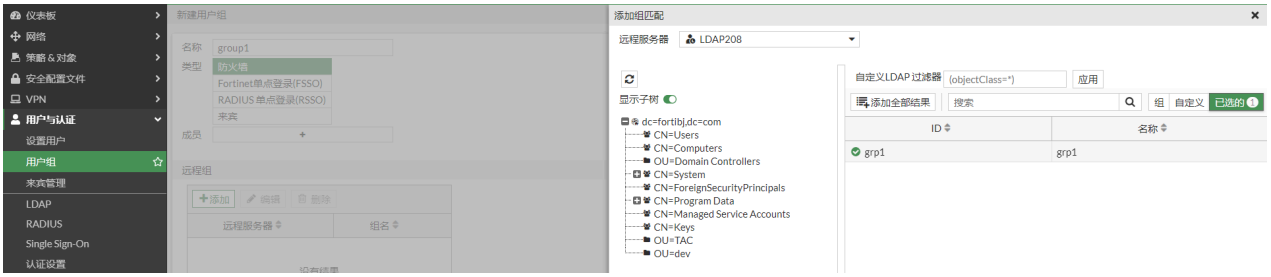


2. 配置 LDAP

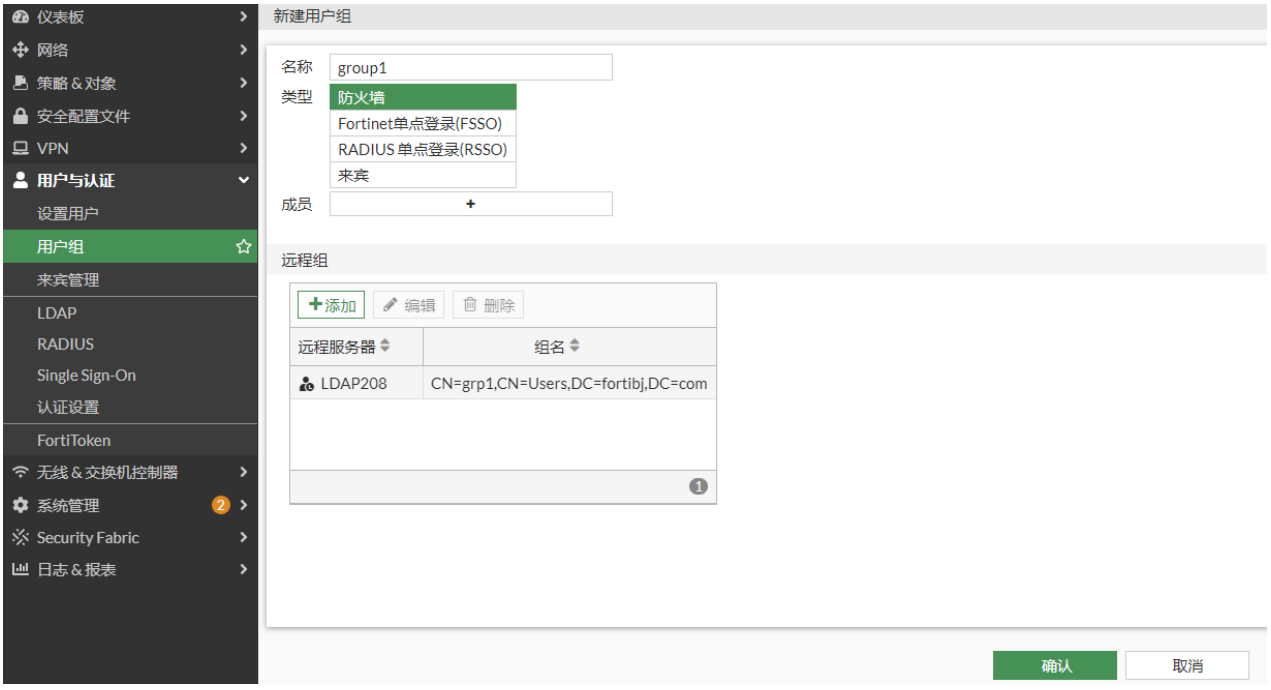
设置 FortiGate 用户组名称 group1，在远程组中，点击“添加”，选择配置好的远程服务器，会自动显示出目录树，找到需要添加的用户组，这里选择 grp1，右击“添加已选”，**那么 group1 匹配远端 LDAP 用户组 grp1 中的用户**。可以添加多个用户组，如再增加 grp2，那么 group1 匹配远端 LDAP 用户组 grp1 和 grp2 中的用户。



点击“已选的”，查看选中的用户组，确认无误后，点击“确认”。



3. 点击确认

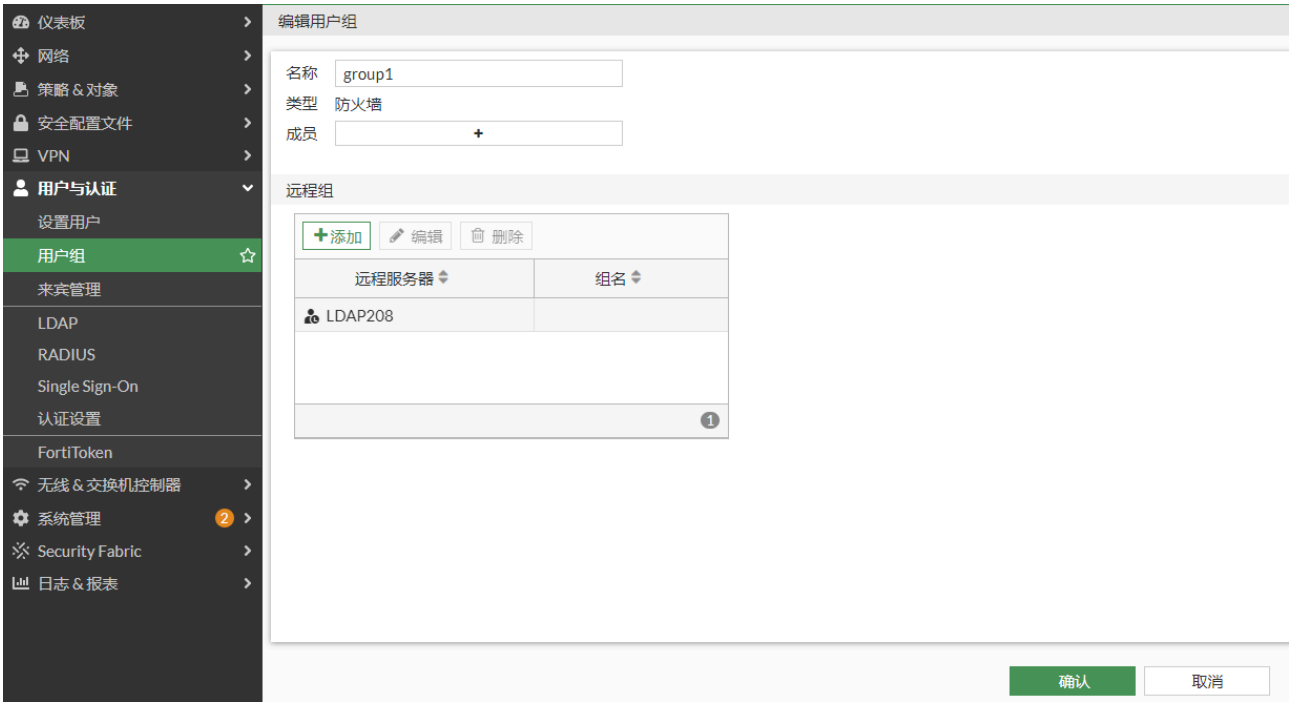


4. 配置完成



重要

如果在 group1 用不选择任何用户组，那么 group1 匹配 LDAP 服务器配置的 DN 目录下的所有用户。



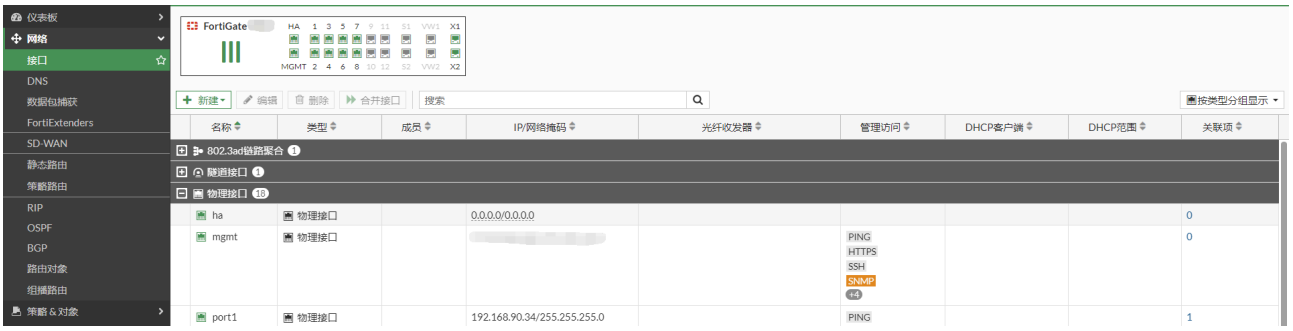
Radius

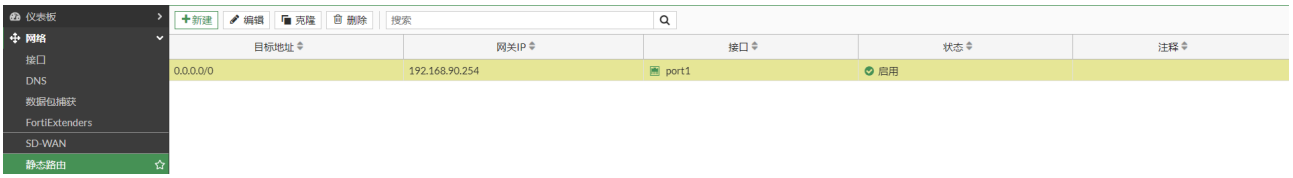
Radius 简介

RADIUS（Remote Authentication Dial In User Service，远程用户拨号认证）是一种广泛支持的客户机 - 服务器协议，提供集中的身份验证、授权和计费功能。RADIUS 服务器使用 UDP 报文与网络中的 RADIUS 客户端通信，在允许用户访问网络之前对用户进行认证，授权适当的用户访问资源，并对所使用的资源进行说明。

配置 Radius 认证

1.配置接口地址及路由



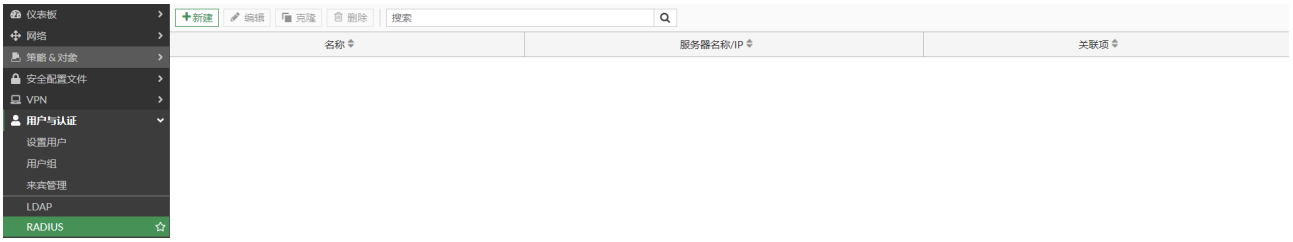


ping 测试 FortiGate 与 Radius Server 之间的连通性。测试环境使用 FortiAuthenticator 作为 Radius 服务器，地址是 192.168.91.209。

```
# execute ping 192.168.91.209
PING 192.168.91.209 (192.168.91.209): 56 data bytes
64 bytes from 192.168.91.209: icmp_seq=0 ttl=63 time=0.3 ms
64 bytes from 192.168.91.209: icmp_seq=1 ttl=63 time=0.2 ms
64 bytes from 192.168.91.209: icmp_seq=2 ttl=63 time=0.1 ms
64 bytes from 192.168.91.209: icmp_seq=3 ttl=63 time=0.1 ms
64 bytes from 192.168.91.209: icmp_seq=4 ttl=63 time=0.2 ms

--- 192.168.91.209 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms
```

2.配置 Radius 服务器

选择”用户与认证“-->” RADIUS“，点击”新建“。



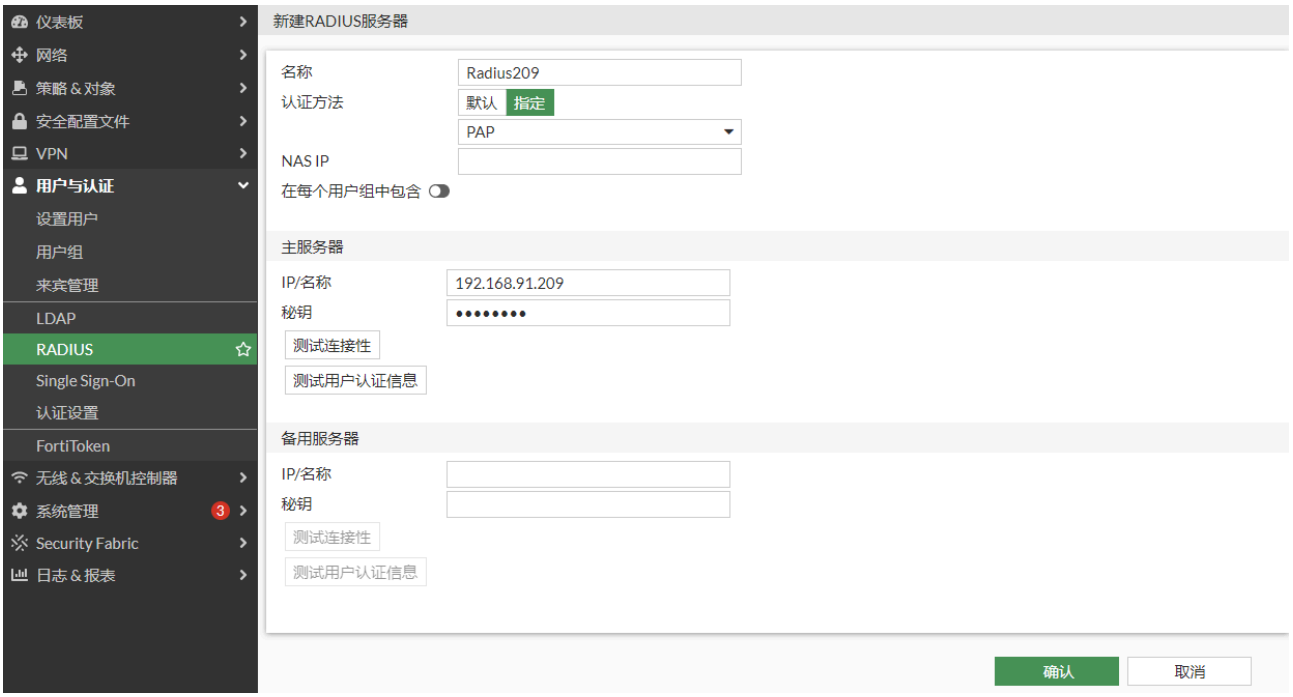
配置参数：

名称：输入一个名称，不冲突即可；

认证方法：FortiGate 支持 CHAP，MS - CHAP，MS - CHAP，PAP 认证方法。如果认证方法是默认，则 FortiGate 会使用 PAP、MSCHAP_v2 和 CHAP 三种方式向 Radius 服务器发送请求，Radius 服务器的日志中往往会报一些错误，因此 **请指定一种认证方法**，这里使用 PAP；

IP/名称：在主服务器输入 Radius 服务器的 IP 地址；

密钥：FortiGate 和 Radius 服务器之间的共享密钥，这里设置为 fortinet。



点击”确认“，配置完成。



对应的命令行如下：

```
config user radius
    edit "Radius209"
        set server "192.168.91.209"
        set secret ENC
fJ6Xhg0wVEZyp8qAJfiuNSJLqyNVrWsRwQ1zELqWc7GnyBso4IrSelJJR4DBKmXQ/ex9cFWq3nJfoWEbWjRy1nrwUMl5HJGPiHO+c8MXXy6IAUE38ec4Y/x5Vn
RPZpEgPPiVIuCXZfbpnjcR1t5RdO8lgTTRkhiNqBFvrZtyxsucpqRYSH7eyTN0ih5nJigvUn8oAg==
        set auth-type pap
    next
end
```

Radius 服务器测试

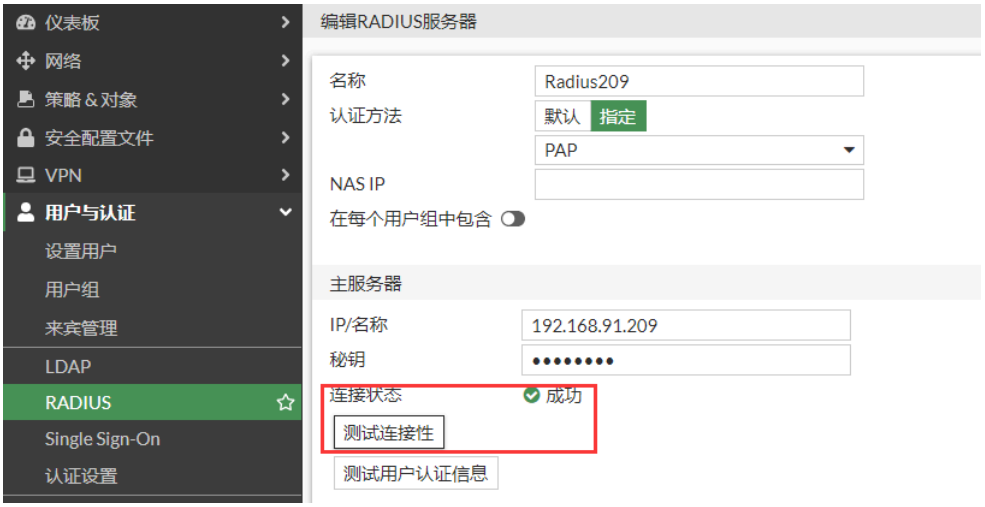
1.FortiGate 与 Radius 服务器之间连通性测试

GUI 编辑 Radius 服务器的配置界面，点击”测试连接性“。

CLI 的语法：

```
#diagnose test authserver radius-direct
<server_name or IP> <port number (0 default port)> <secret> <pap | chap | mschap | mschap2> <user> <password>
```

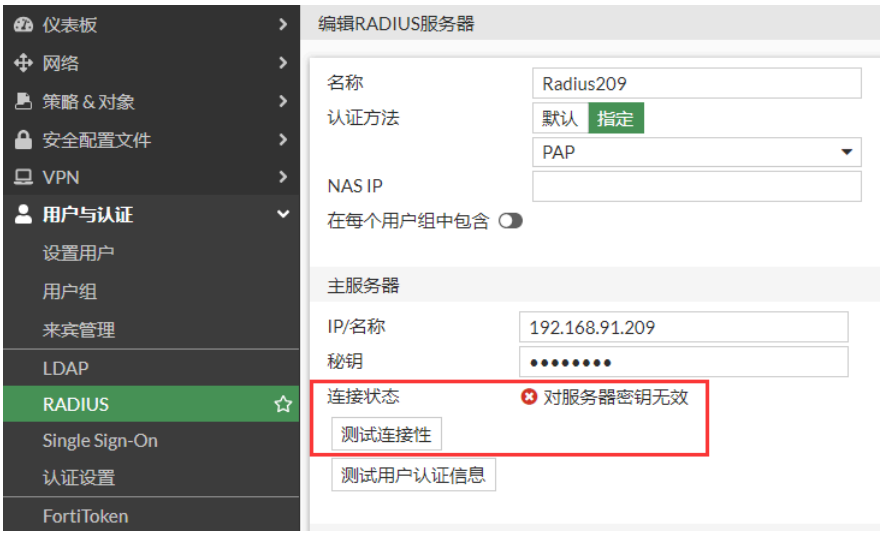
- 如果连接状态是”成功“，则 FortiGate 与 Radius 服务器之间通信正常。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is OK
Access-Accept
AVP: l=12 t=Vendor-Specific(26) v=Fortinet(12356)
  VSA: l=6 t=Fortinet-Group-Name(1)
  Value: 'grp1'
```

- 如果连接状态是“对服务器密钥无效”，则 FortiGate 与 Radius 服务器之间的密钥不一致。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is Secret invalid
```

- 如果连接状态是“不能连接 RADIUS 服务器”，则 Radius 服务器的地址或者端口不正确，或者 Radius 服务器没有授权 FortiGate 作为客户端。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is Server unreachable
```

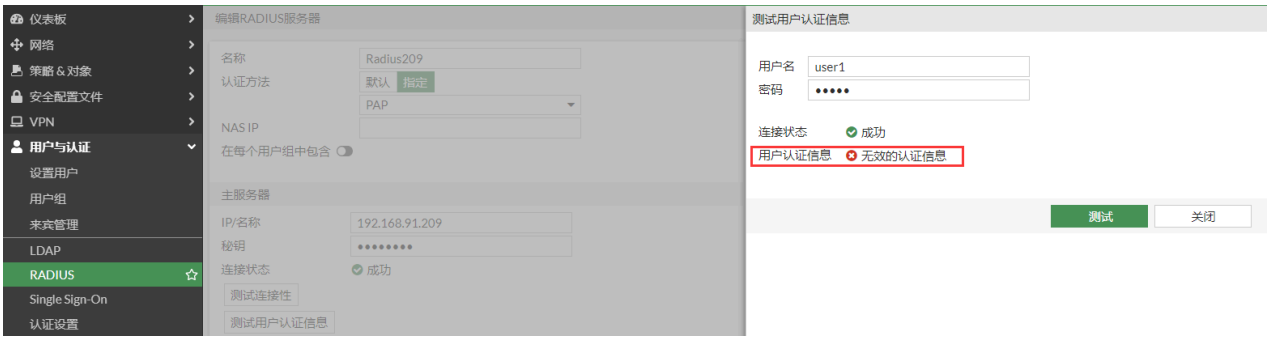
2.测试 Radius 服务器中的用户及密码

GUI 编辑 Radius 服务器的配置界面，点击”测试用户认证信息“。

CLI 的语法：

```
# diagnose test authserver radius
<server_name> <chap | pap | mschap | mschap2> <username> <password>
```

- 用户名或者密码不正确



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius Radius209 pap user1 test123
authenticate 'user1' against 'pap' failed, assigned_rad_session_id=730523922 session_timeout=0 secs idle_timeout=0 secs!
```

- 用户名和密码正确

可以看到 Radius 服务器返回的状态是成功，以及用户属于，如 user1 属于用户组 grp1。

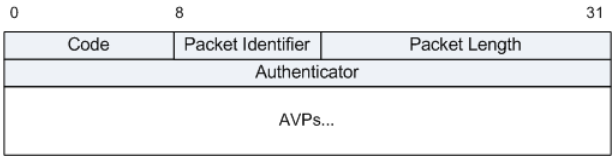


对应的 CLI 测试结果如下：

```
# diagnose test authserver radius Radius209 pap user1 12345678
authenticate 'user1' against 'pap' succeeded, server=primary assigned_rad_session_id=730523923 session_timeout=0 secs
idle_timeout=0 secs!
Group membership(s) - grp1
```

Radius 报文格式及交互过程

1.Radius 报文格式



Code：表示 Radius 报文的类型：

- Code = 1: Access-Request，认证请求报文；
- Code = 2: Access-Accept，认证成功报文；
- Code = 3: Access-Reject，认证拒绝报文；
- Code = 4: Accounting-Request，计费请求报文；
- Code = 5: Accounting-Response，计费响应报文；
- Code = 11: Access-Challenge，认证挑战报文；
- Code = 255: Reserved，保留；

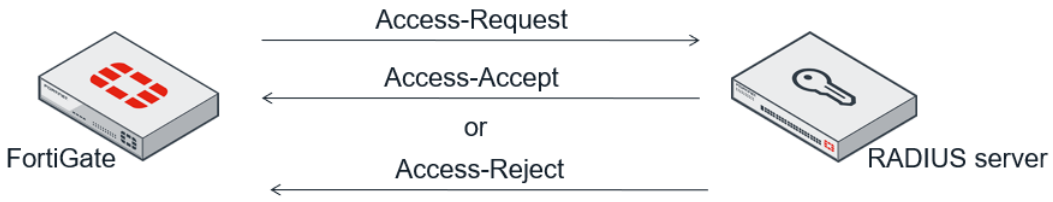
Packet Identifier：用于匹配请求和回应报文。

Packet Length：Radius 报文长度；

Authenticator：用于认证来自服务端的响应，也用于用户密码的加密处理；

AVPs：属性字段，用于 RADIUS 服务器和 RADIUS 客户端之间的信息交换，Fortinet RADIUS 属性的供应商 ID 是 12356。

2.Radius 报文交互过程



- FortiGate 向 RADIUS 服务器发送包含用户名和密码信息的认证请求报文。
- RADIUS 服务器对用户身份的合法性进行检验：

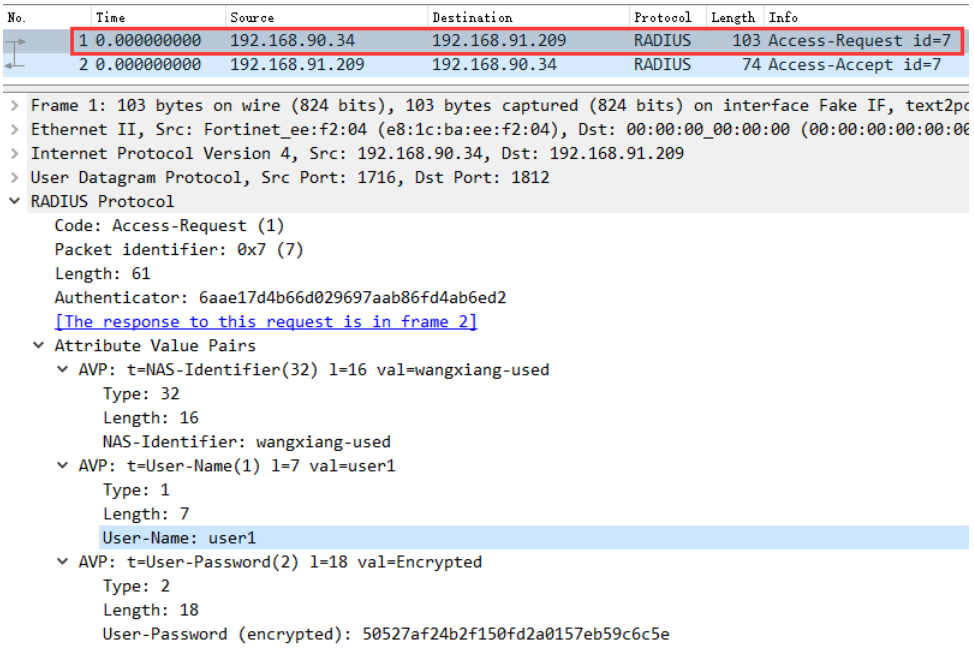
  如果用户身份合法，RADIUS 服务器向 FortiGate 返回认证接受报文。

  如果用户身份不合法，RADIUS 服务器向 RADIUS 客户端返回认证拒绝报文。

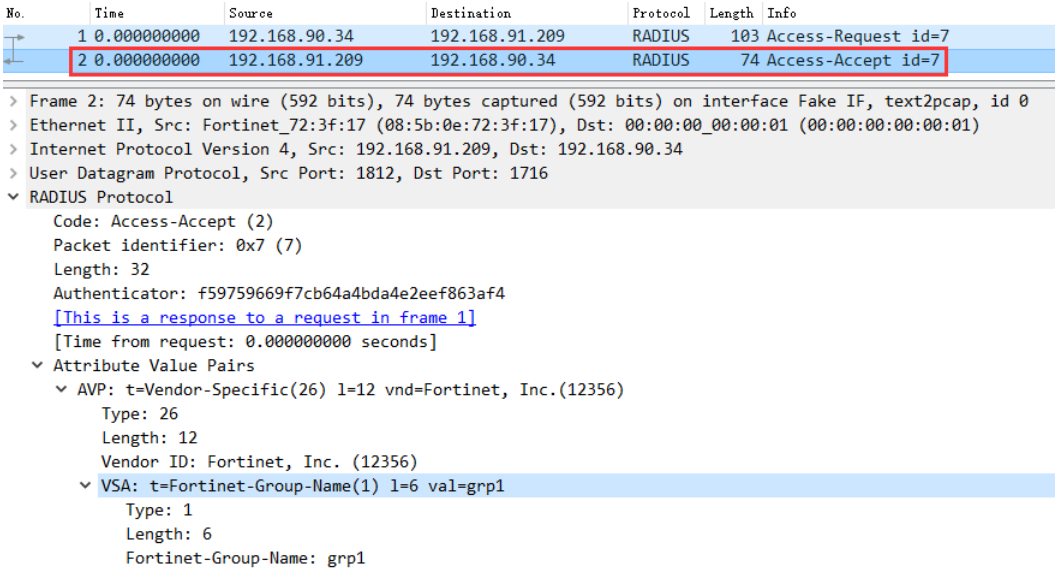
PAP 和 CHAP 认证

1.PAP 认证

在认证请求报文 Access-Request 报文中，PAP 将用户和加密后的密码传递给 Radius 服务器。



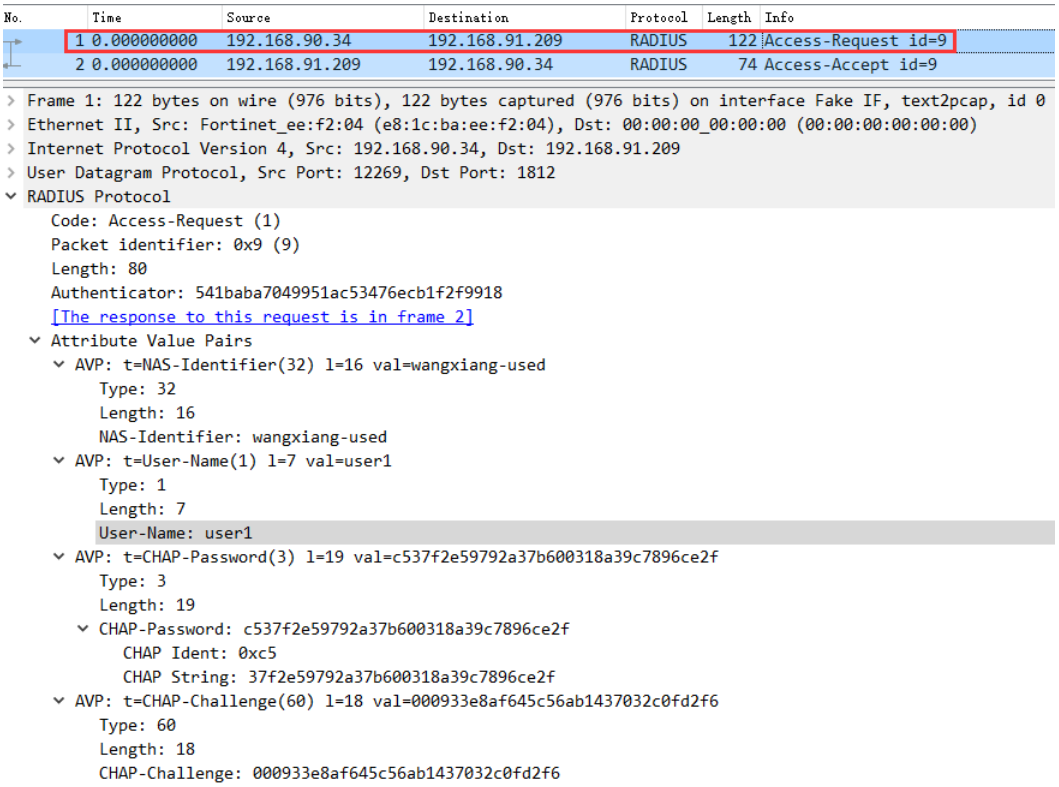
Radius 服务器将解密后的密码，与用户的密码进行比较，如果一致， 则认为密码正确。



2.CHAP 认证

> **重要**
>
> CHAP 认证传递的是一个 HASH 值，不是密码，因此这要求用户的密码保存在 Radius 服务器本地。如果 Radius 服务器用户的密码不在本地，如 FortiGate----Radius 服务器 ----LDAP 服务器，像这样的部署方式，用户的密码保存在 LDAP 服务器中，Radius 服务器无法获取 LDAP 服务器中用户的密码，因此认证将失败。

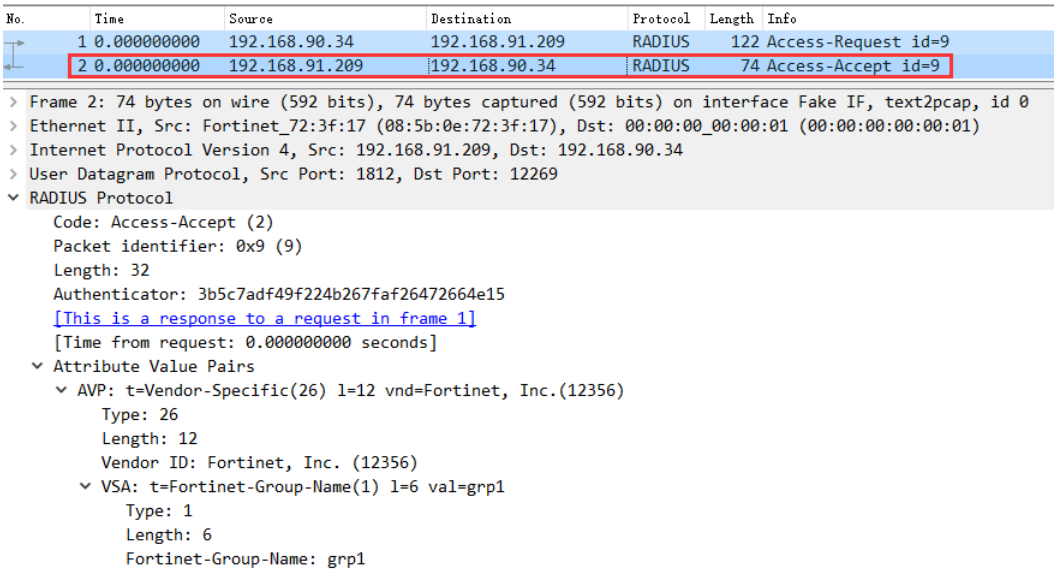
在认证请求报文 Access-Request 报文中，CHAP 认证并不会直接在认证过程传递用户密码， 而是传递 chapID + 密码 + challenge 的 MD5 值。



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.209	RADIUS	122	Access-Request id=9
2	0.000000000	192.168.91.209	192.168.90.34	RADIUS	74	Access-Accept id=9

```
> Frame 1: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.209
> User Datagram Protocol, Src Port: 12269, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x9 (9)
    Length: 80
    Authenticator: 541baba7049951ac53476ecb1f2f9918
    [The response to this request is in frame 2]
  v Attribute Value Pairs
    v AVP: t=NAS-Identifier(32) l=16 val=wangxiang-used
        Type: 32
        Length: 16
        NAS-Identifier: wangxiang-used
    v AVP: t=User-Name(1) l=7 val=user1
        Type: 1
        Length: 7
        User-Name: user1
    v AVP: t=CHAP-Password(3) l=19 val=c537f2e59792a37b600318a39c7896ce2f
        Type: 3
        Length: 19
      v CHAP-Password: c537f2e59792a37b600318a39c7896ce2f
          CHAP Ident: 0xc5
          CHAP String: 37f2e59792a37b600318a39c7896ce2f
    v AVP: t=CHAP-Challenge(60) l=18 val=000933e8af645c56ab1437032c0fd2f6
        Type: 60
        Length: 18
        CHAP-Challenge: 000933e8af645c56ab1437032c0fd2f6
```

Radius 服务器通过用户名找到对应的密码，计算 chapID + 密码 + challenge 的 MD5 值， 并判断是否与 chap-password 一致（从第 2 个字节开始比较），如果一致， 则认为密码正确。



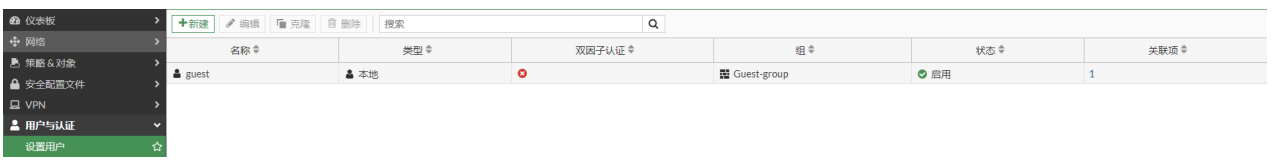
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.209	RADIUS	122	Access-Request id=9
2	0.000000000	192.168.91.209	192.168.90.34	RADIUS	74	Access-Accept id=9

```
> Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.209, Dst: 192.168.90.34
> User Datagram Protocol, Src Port: 1812, Dst Port: 12269
v RADIUS Protocol
    Code: Access-Accept (2)
    Packet identifier: 0x9 (9)
    Length: 32
    Authenticator: 3b5c7adf49f224b267faf26472664e15
    [This is a response to a request in frame 1]
    [Time from request: 0.000000000 seconds]
  v Attribute Value Pairs
    v AVP: t=Vendor-Specific(26) l=12 vnd=Fortinet, Inc.(12356)
        Type: 26
        Length: 12
        Vendor ID: Fortinet, Inc. (12356)
      v VSA: t=Fortinet-Group-Name(1) l=6 val=grp1
          Type: 1
          Length: 6
          Fortinet-Group-Name: grp1
```

Radius 认证 Debug

1.认证成功

```
[1906] handle_req-Rcvd auth req 730524052 for user1 in Radius209 opt=0000001d prot=0
[466] __compose_group_list_from_req-Group 'Radius209', type 1
[616] fnbamd_pop3_start-user1
[518] __fnbamd_cfg_get_radius_list_by_server-Loading RADIUS server 'Radius209'
[342] fnbamd_create_radius_socket-Opened radius socket 12
[342] fnbamd_create_radius_socket-Opened radius socket 13
[1394] fnbamd_radius_auth_send-Compose RADIUS request
[1351] fnbamd_rad_dns_cb-192.168.91.209→192.168.91.209

认证请求，radius服务器是192.168.91.209，用户名user1，认证方法是PAP
[1323] __fnbamd_rad_send-Sent radius req to server 'Radius209': fd=12,
```

```
IP=192.168.91.209(192.168.91.209:1812) code=1 id=6 len=101 user="user1" using PAP
  [319] radius_server_auth-Timer of rad 'Radius209' is added
  [633] create_auth_session-Total 1 server(s) to try
  [1360] fnbamd_auth_handle_radius_result-Timer of rad 'Radius209' is deleted

  认证响应，响应code 2，认证成功，返回的用户属性
  [1802] fnbamd_radius_auth_validate_pkt-RADIUS resp code 2
  [320] extract_success_vsas-FORTINET attr, type 1, val grp1
  [1385] fnbamd_auth_handle_radius_result⟶Result for radius svr 'Radius209' 192.168.91.209(1) is 0
  [266] find_matched_usr_grps-Skipped group matching
  [216] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 730524052, len=2154
  [789] destroy_auth_session-delete session 730524052
  ```
```

**2.认证失败**

```
  ```
  [1906] handle_req-Rcvd auth req 730524053 for user1 in Radius209 opt=0000001d prot=0
  [466] __compose_group_list_from_req-Group 'Radius209', type 1
  [616] fnbamd_pop3_start-user1
  [518] __fnbamd_cfg_get_radius_list_by_server-Loading RADIUS server 'Radius209'
  [342] fnbamd_create_radius_socket-Opened radius socket 12
  [342] fnbamd_create_radius_socket-Opened radius socket 13
  [1394] fnbamd_radius_auth_send-Compose RADIUS request
  [1351] fnbamd_rad_dns_cb-192.168.91.209→192.168.91.209

  认证请求，radius服务器是192.168.91.209，用户名user1，认证方法是PAP
  [1323] __fnbamd_rad_send-Sent radius req to server 'Radius209': fd=12,
IP=192.168.91.209(192.168.91.209:1812) code=1 id=7 len=101 user="user1" using PAP
  [319] radius_server_auth-Timer of rad 'Radius209' is added
  [633] create_auth_session-Total 1 server(s) to try
  [1360] fnbamd_auth_handle_radius_result-Timer of rad 'Radius209' is deleted

  认证响应，响应code 3，认证拒绝
  [1802] fnbamd_radius_auth_validate_pkt-RADIUS resp code 3
  [1385] fnbamd_auth_handle_radius_result⟶Result for radius svr 'Radius209' 192.168.91.209(1) is 1
  [216] fnbamd_comm_send_result-Sending result 1 (nid 0) for req 730524053, len=3172
  [789] destroy_auth_session-delete session 730524053
  ```
```

## 其他 Radius 配置参数

**1.修改 Radius 端口**

```
config user radius
    edit Radius209
        set radius-port xxx
    next
end
```

2.指定原 IP 地址

```
config user radius
    edit Radius209
        set source-ip x.x.x.x
    next
end
```

创建 radius 用户

1. 选择“用户与认证”-->“设置用户”，点击“新建”



2. 选择 "远程 RADIUS 用户"，点击“下一步”
```



3. 输入“用户名”和“RADIUS 服务器”，点击“下一步”



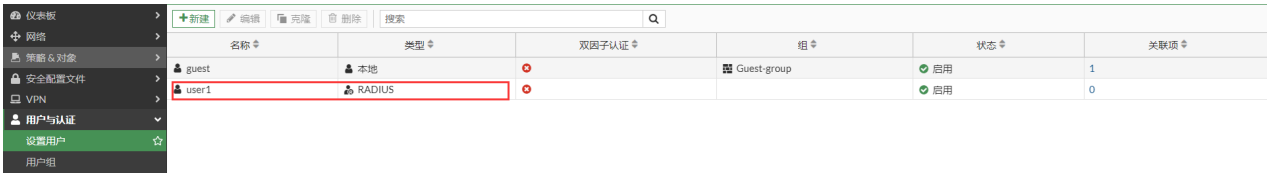
4. 如果 Radius 用户需要绑定 FortiGate 本地的 Fortitoken，则勾选“双因子认证”。这里不需要，直接点击“下一步”



5. 如果需要现在为 Radius 用户指定一个用户组，可以勾选“用户组”来指定，也可以等创建完用户后，在用户组菜单中添加。这里不勾选，直接点击“提交”



6. 创建完成



创建 radius 用户组

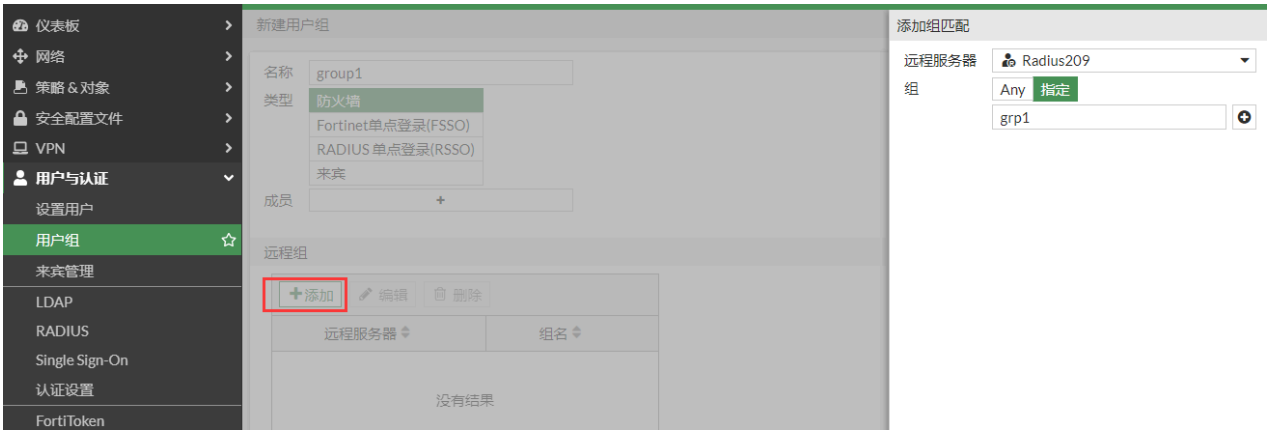
1. 选择“用户与认证”-->"用户组"，点击 "新建"



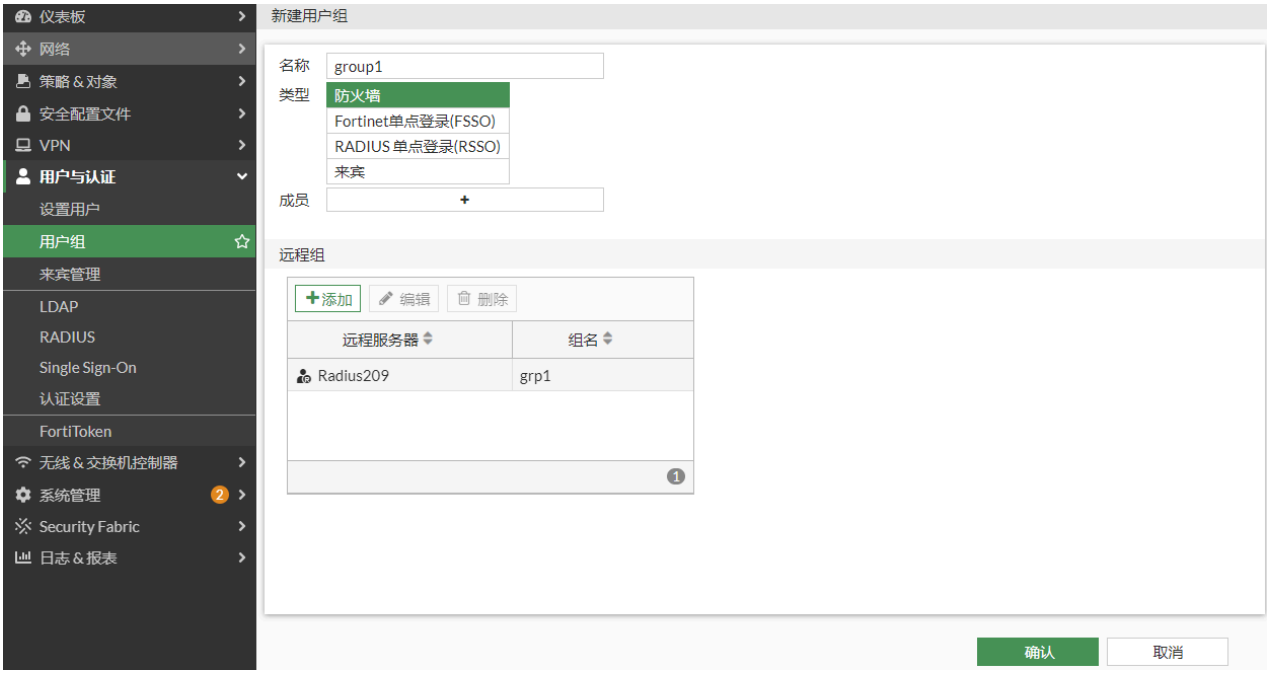
2. 配置参数

输入用户组名称 group1，在远程组下，点击“添加”，在”远程服务器“中选择已配置的 Radius 服务器。

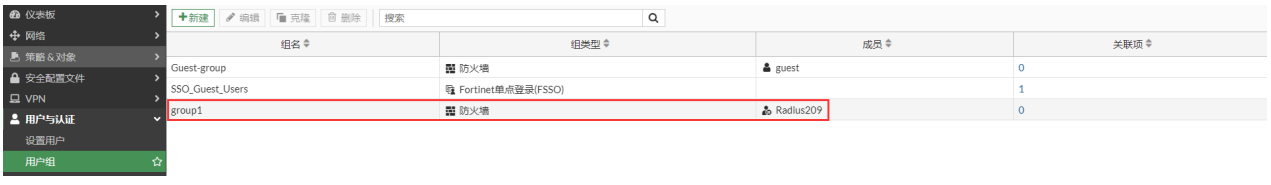
指定 Radius 的用户组，这里使用 grp1，表示 group1 匹配 Radius 用户组 grp1 下的用户，可以添加多个组；如果组选择“Any”，表示匹配 Radius 服务器下的所用用户。



选择完成后，点击“确认”。



3. 配置完成



LDAP 认证

- LDAP 认证原理
- LDAP 认证配置与排错
- LDAP 常见错误代码
- LDAPS CA 强制认证

LDAP 认证原理

LDAP 简介

LDAP（Light Directory Access Protocol），轻量级目录访问协议，是一个为查询、浏览和搜索而优化的专业分布式数据库，它成树状结构组织数据。目录服务是由目录数据库和一套访问协议组成的系统。

LDAP 组织结构

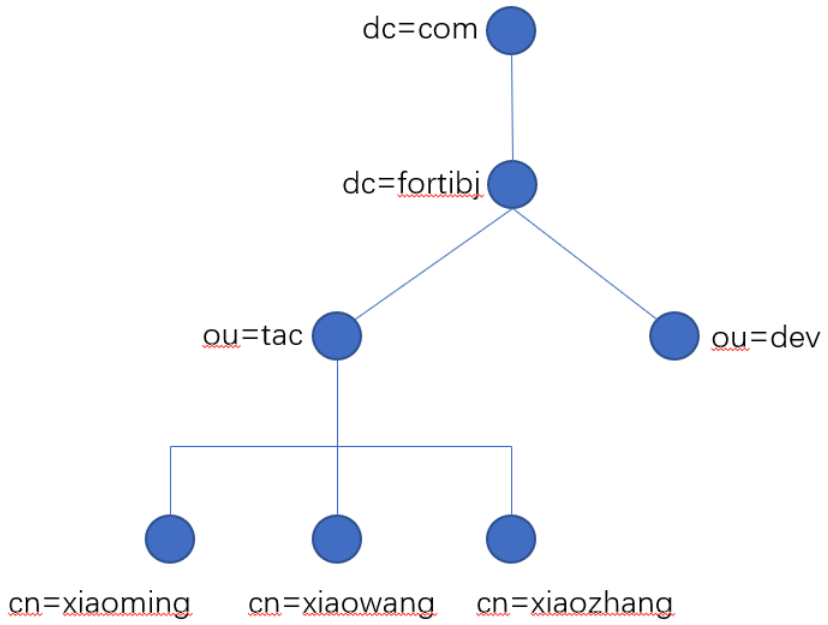
在 LDAP 中目录是按树形结构组织，而 DN（Distinguished Name）唯一可区别的名称，记录了一条记录的位置。DN 有三个主要属性，分别是 DC（Domain Component）、OU（Organizational Unit）与 CN（Common Name）。

DC：域控制器，类似于文件系统目录；

OU：组织单元，类似部门；

CN：区分身份的属性，类似名称。

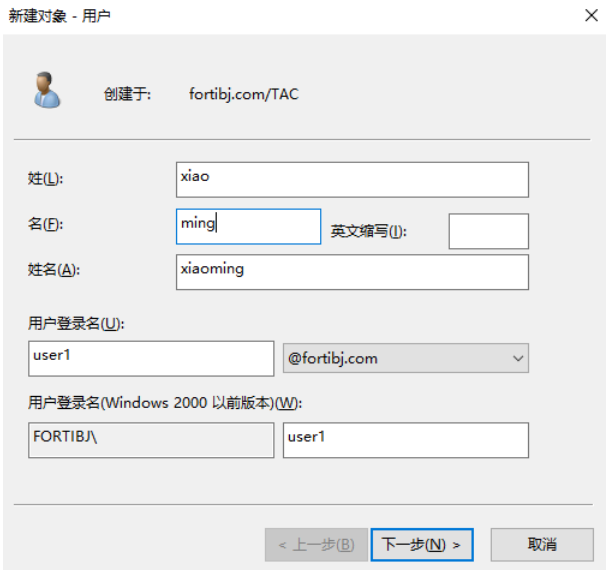
如下图所示，dn 是从树的叶子节点逐层向根节点推导出的一条唯一路径，表明了这个数据结构中的唯一一条记录。一个完整的 dn：cn = xiaoming,ou = tac,dc = fortibj,dc = com，查询得到一条完整的记录，然后再取出所需属性 (Attribute) 的值。



Windows AD 域中几个属性

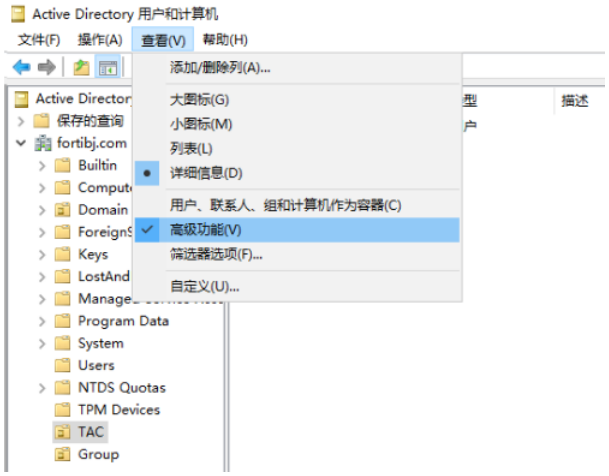
1.CN 和 sAMAccountName

Windows AD 域中创建用户时，输入信息如下：

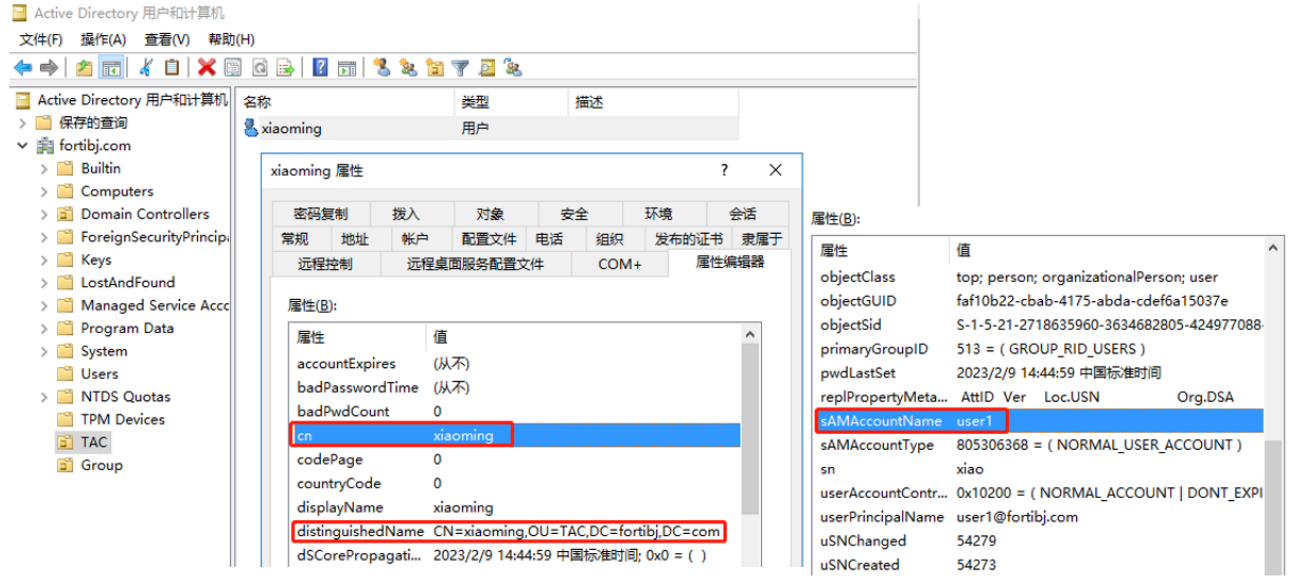


cn 表示姓名（full user name），即 xiaoming，sAMAccountName 表示用户登录名（login user name），即 user1。

在 windows Active Directory 用户和计算机中，点击”查看“，选择”高级功能“，然后可以查看更多的用户属性。



通过用户属性查询到用户 xiaoming 的 CN 是 xiaoming，sAMAccountName 是 user1，dn 是 CN = xiaoming,OU = TAC,DC = fortibj,DC = com



windows CLI 查询 DN

-根据 full name 查询

```
C:\Users\Administrator>dsquery user -name xiaoming
"CN=xiaoming,OU=TAC,DC=fortibj,DC=com"
```

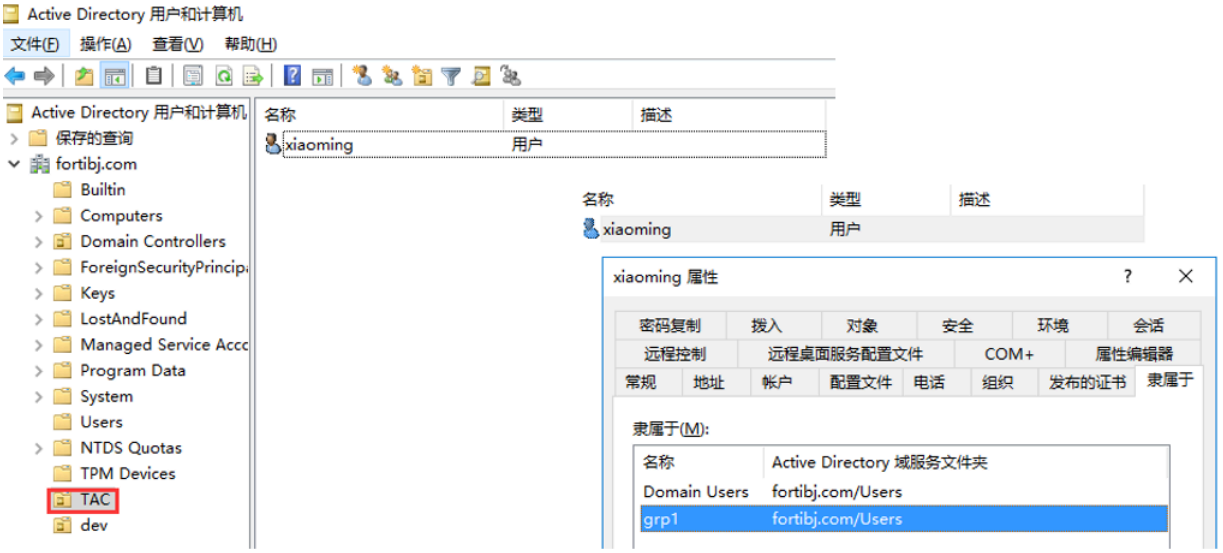
-根据 login name 查询

```
C:\Users\Administrator>dsquery user -samid user1
"CN=xiaoming,OU=TAC,DC=fortibj,DC=com"
```

2.OU 和 Group

OU 是组织单元，Group 是用户组，两者不一样。

从下图可以看出用户 xiaoming 在组织单元 OU TAC 下，属于用户组 Domain Users 和 grp1。



LDAP 交互过程

1.LDAP 交互过程



2.报文交互过程分析

FortiGate 执行如下命令验证用户名和密码，获取用户组属性。

```
# diagnose test authserver ldap LDAP208 user1 Pass@123456
authenticate 'user1' against 'LDAP208' succeeded!
Group membership(s) - CN=grp1,CN=Users,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

报文交互过程如下：

- 建立 TCP 连接

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.208	TCP	74	4996 → 389 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=15603975 TSecr=0 WS=16384
2	0.000000000	192.168.91.208	192.168.90.34	TCP	74	389 → 4996 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM TSval=246458301 TSecr=15603975
3	0.000000000	192.168.90.34	192.168.91.208	TCP	66	4996 → 389 [ACK] Seq=1 Ack=1 Win=180224 Len=0 TSval=15603975 TSecr=246458301

- FortiGate 以管理员 DN 和密码为参数向 LDAP 服务器发送管理员绑定请求报文（Bind Request）获得查询权限

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000000000	192.168.90.34	192.168.91.208	LDAP	135	bindRequest(1) "CN=Administrator,CN=Users,DC=fortibj,DC=com" simple

```
> Frame 4: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 1, Ack: 1, Len: 69
v Lightweight Directory Access Protocol
  v LDAPMessage bindRequest(1) "CN=Administrator,CN=Users,DC=fortibj,DC=com" simple
      messageID: 1
    v protocolOp: bindRequest (0)
      v bindRequest
          version: 3
          name: CN=Administrator,CN=Users,DC=fortibj,DC=com
        v authentication: simple (0)
            simple: Test@123456
```

- LDAP 服务器进行绑定请求报文的处理。如果绑定成功，则向 FortiGate 发送绑定成功的回应报文

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000000000	192.168.91.208	192.168.90.34	LDAP	88	bindResponse(1) success

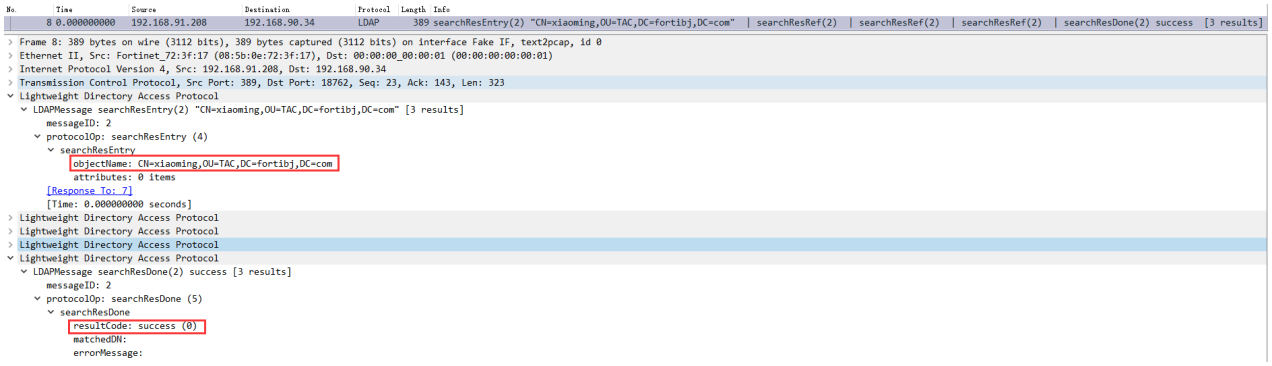
```
> Frame 5: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
> Transmission Control Protocol, Src Port: 389, Dst Port: 18762, Seq: 1, Ack: 70, Len: 22
v Lightweight Directory Access Protocol
  v LDAPMessage bindResponse(1) success
      messageID: 1
    v protocolOp: bindResponse (1)
      v bindResponse
          resultCode: success (0)
          matchedDN:
          errorMessage:
```

- FortiGate 以输入的用户名为参数，向 LDAP 服务器发送用户 DN 查询请求报文（Search Request）

No.	Time	Source	Destination	Protocol	Length	Info
7	0.000000000	192.168.90.34	192.168.91.208	LDAP	139	searchRequest(2) "dc=fortibj,dc=com" wholeSubtree

```
> Frame 7: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 70, Ack: 23, Len: 73
v Lightweight Directory Access Protocol
  v LDAPMessage searchRequest(2) "dc=fortibj,dc=com" wholeSubtree
      messageID: 2
    v protocolOp: searchRequest (3)
      v searchRequest
          baseObject: dc=fortibj,dc=com
          scope: wholeSubtree (2)
          derefAliases: neverDerefAliases (0)
          sizeLimit: 0
          timeLimit: 0
          typesOnly: True
        v Filter: (sAMAccountName=user1)
          v filter: equalityMatch (3)
            v equalityMatch
                attributeDesc: sAMAccountName
                assertionValue: user1
        v attributes: 1 item
            AttributeDescription: 1.1
```

- LDAP 服务器收到查询请求报文后，对用户 DN 进行查找。如果查询成功，则向 FortiGate 发送查询成功的回应报文，返回用户的 DN

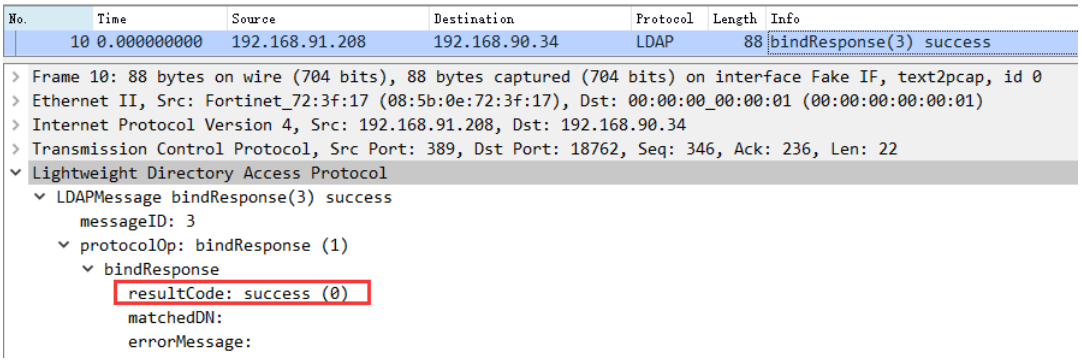


- FortiGate 以查询得到的用户 DN 和用户输入的密码为参数，向 LDAP 服务器发送用户 DN 绑定请求报文（Bind Request），检查用户密码是否正确

No.	Time	Source	Destination	Protocol	Length	Info
9	0.000000000	192.168.90.34	192.168.91.208	LDAP	159	bindRequest(3) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" simple

```
> Frame 9: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 143, Ack: 346, Len: 93
v Lightweight Directory Access Protocol
  v LDAPMessage bindRequest(3) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" simple
      messageID: 3
    v protocolOp: bindRequest (0)
      v bindRequest
          version: 3
          name: CN=xiaoming,OU=TAC,DC=fortibj,DC=com
        v authentication: simple (0)
            simple: Pass@123456
```

- LDAP 服务器进行绑定请求报文的处理，如果绑定成功，则向 FortiGate 发送绑定成功的回应报文；如果绑定失败，则向 FortiGate 发送绑定失败的回应报文

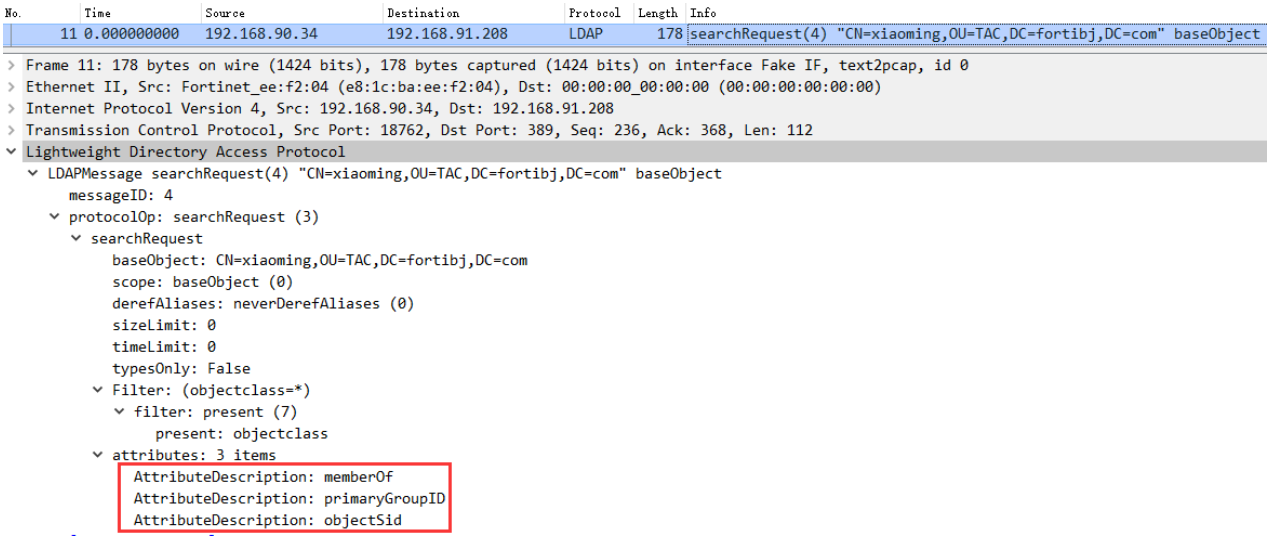


- 如果绑定成功，FortiGate 向 LDAP 服务器发送用户属性查询请求报文（Search Request），获取用户所属的用户组

  memberof：用户所属的组的 DN。

  primaryGroupID：指定用户的主组的相对标识符 (RID)。默认情况下是 Domain Users 组的 RID，用户是其主组的成员，Domain User 没有列在用户的 memberOf 属性中。因此需要单独查询主组的 DN。

  objectSid：指定安全主体对象的安全标识符 (SID)。SID 是用于标识安全主体对象的唯一值。



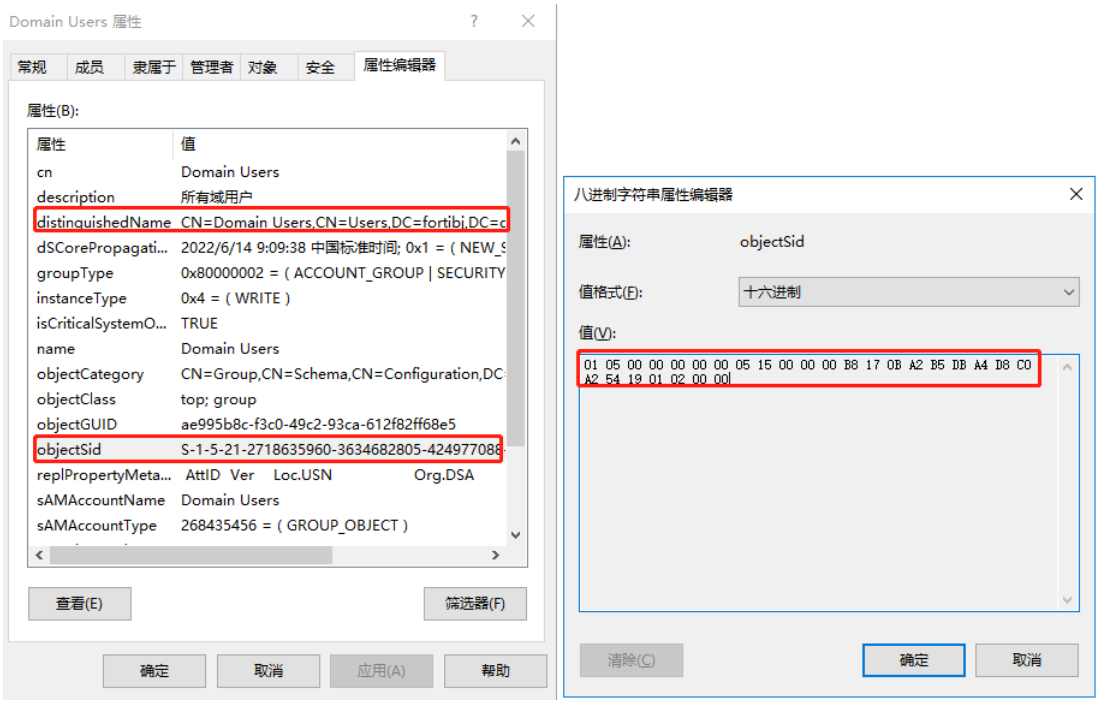
- LDAP 服务器对查询请求报文进行处理，返回用户所在的用户组

```
No.  Time          Source          Destination     Protocol  Length  Info
  12 0.000000000   192.168.91.208  192.168.90.34   LDAP         291  searchResEntry(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" | searchResDone(4) success [3 results]

> Frame 12: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
> Transmission Control Protocol, Src Port: 389, Dst Port: 18762, Seq: 368, Ack: 348, Len: 225
v Lightweight Directory Access Protocol
  v LDAPMessage searchResEntry(4) "CN=xiaoming,OU=TAC,DC=fortibj,DC=com" [3 results]
      messageID: 4
    v protocolOp: searchResEntry (4)
      v searchResEntry
          objectName: CN=xiaoming,OU=TAC,DC=fortibj,DC=com
        v attributes: 3 items
          v PartialAttributeList item memberOf
              type: memberOf
            v vals: 1 item
                AttributeValue: CN=grp1,CN=Users,DC=fortibj,DC=com
          v PartialAttributeList item primaryGroupID
              type: primaryGroupID
            v vals: 1 item
                AttributeValue: 513
          v PartialAttributeList item objectSid
              type: objectSid
            v vals: 1 item
              v SID: S-1-5-21-2718635960-3634682805-424977088-2104  (Domain SID-Domain RID)
                  Revision: 1
                  Num Auth: 5
                  Authority: 5
                  Subauthorities: 21-2718635960-3634682805-424977088-2104
                  RID: 2104  (Domain RID)
      [Response To: 11]
      [Time: 0.000000000 seconds]
v Lightweight Directory Access Protocol
  v LDAPMessage searchResDone(4) success [3 results]
      messageID: 4
    v protocolOp: searchResDone (5)
      v searchResDone
          resultCode: success (0)
          matchedDN:
          errorMessage:
```

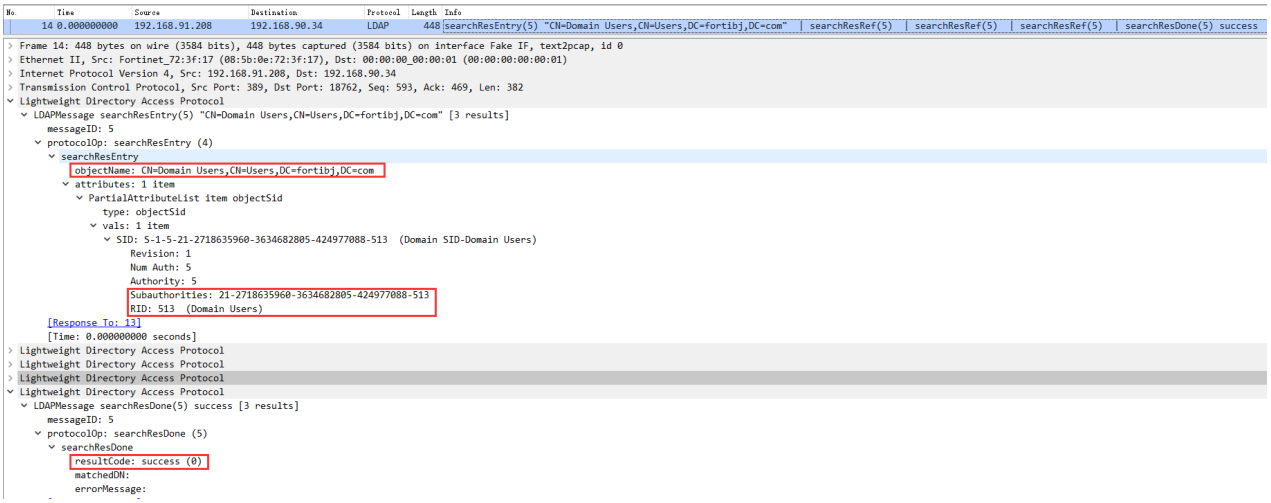
- FortiGate 向 LDAP 服务器发送用户属性查询请求报文（Search Request），获取用户所在的主组

```
No.  Time          Source          Destination     Protocol  Length  Info
  13 0.000000000   192.168.90.34   192.168.91.208  LDAP         187  searchRequest(5) "dc=fortibj,dc=com" wholeSubtree

> Frame 13: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.208
> Transmission Control Protocol, Src Port: 18762, Dst Port: 389, Seq: 348, Ack: 593, Len: 121
v Lightweight Directory Access Protocol
  v LDAPMessage searchRequest(5) "dc=fortibj,dc=com" wholeSubtree
      messageID: 5
    v protocolOp: searchRequest (3)
      v searchRequest
          baseObject: dc=fortibj,dc=com
          scope: wholeSubtree (2)
          derefAliases: neverDerefAliases (0)
          sizeLimit: 0
          timeLimit: 0
          typesOnly: False
        v Filter: (&(objectclass=group)(objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…))
          v filter: and (0)
            v and: (&(objectclass=group)(objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…))
              v and: 2 items
                v Filter: (objectclass=group)
                  v and item: equalityMatch (3)
                    v equalityMatch
                        attributeDesc: objectclass
                        assertionValue: group
                v Filter: (objectSid=01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…)
                  v and item: equalityMatch (3)
                    v equalityMatch
                        attributeDesc: objectSid
                        assertionValue: 01:05:00:00:00:00:00:05:15:00:00:00:b8:17:0b:a2:b5:db:a4:d8:c0:a2:54:19:…
        v attributes: 1 item
            AttributeDescription: objectSid

0000  00 00 00 00 00 00 e8 1c  ba ee f2 04 08 00 45 00   ........ ......E·
0010  00 ad 25 15 00 00 40 06  1d f3 c0 a8 5a 22 c0 a8   ··%···@· ····Z"··
0020  5b d0 49 4a 01 85 92 0d  7b 26 66 83 9a 91 80 18   [·IJ···· {&f·····
0030  00 0b e6 55 00 00 01 01  08 0a 00 8a d7 ac 0a d0   ···U···· ········
0040  16 f0 30 77 02 01 05 63  72 04 11 64 63 3d 66 6f   ··0w···c r··dc=fo
0050  72 74 69 62 6a 2c 64 63  3d 63 6f 6d 0a 01 02 0a   rtibj,dc =com····
0060  01 00 02 01 00 02 01 00  01 01 00 a0 41 a3 14 04   ........ ····A···
0070  0b 6f 62 6a 65 63 74 63  6c 61 73 73 04 05 67 72   ·objectc lass··gr
0080  6f 75 70 a3 29 04 09 6f  62 6a 65 63 74 53 69 64   oup·)··o bjectSid
0090  04 1c 01 05 00 00 00 00  00 05 15 00 00 00 b8 17   ·····... ........
00a0  0b a2 b5 db a4 d8 c0 a2  54 19 01 02 00 00 30 0b   ........ T·····0·
00b0  04 09 6f 62 6a 65 63 74  53 69 64                  ··object Sid
```

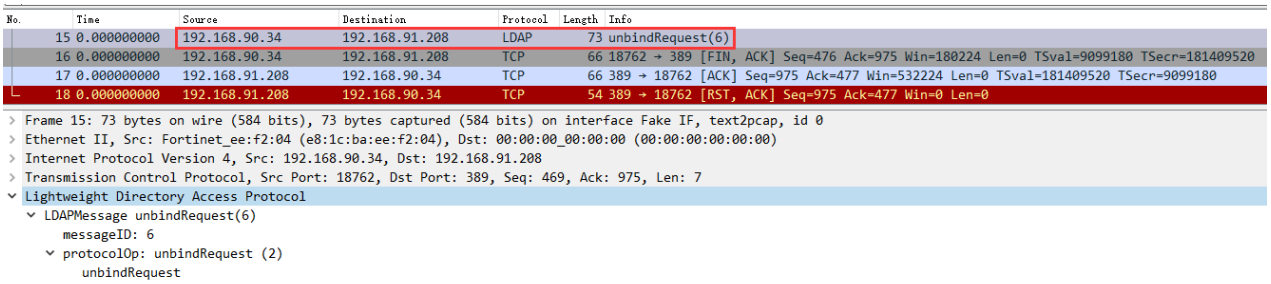
LDAP 服务器中 Domain User objectSid 的值与上图 assertionValue 的值一致。



- LDAP 服务器对查询请求报文进行处理，返回用户所在主组的 DN



- FortiGate 解除绑定



常见的标准的 LDAP 响应码

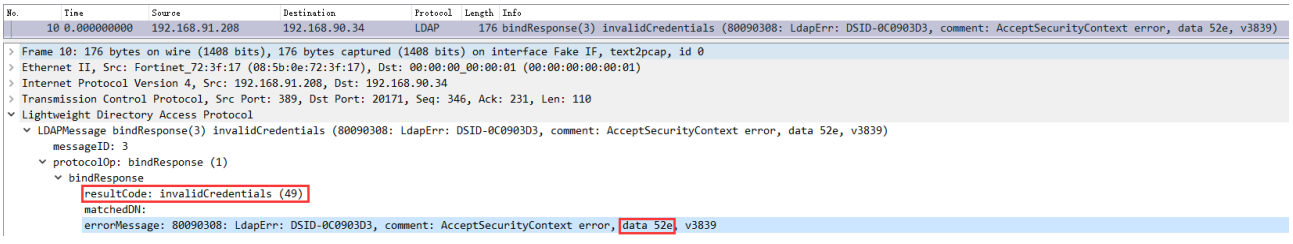
1.常见的标准的 LDAP 响应码

LDAP Result Code	Description
0	SUCCESS
2	PROTOCOL ERROR
7	AUTH METHOD NOT SUPPORTED
16	NO SUCH ATTRIBUTE
21	INVALID SYNTAX
32	NO SUCH OBJECT
34	INVALID DN SYNTAX
49	INVALID CREDENTIALS/ACCOUNT DISABLED
50	LDAP INSUFFICIENT ACCESS

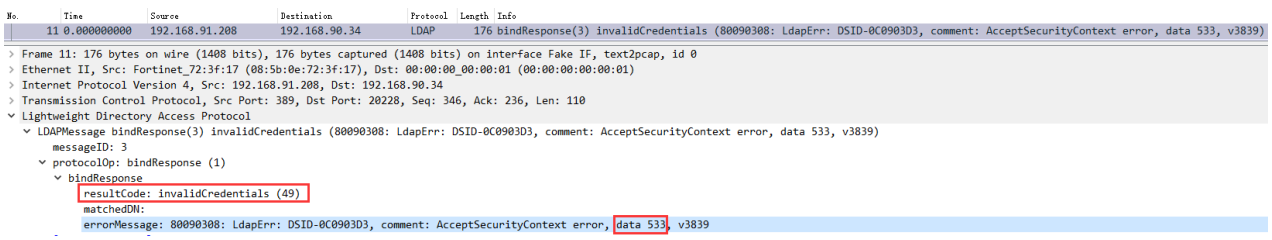
LDAP Error Message Data	Description
0x525	user not found
0x52e	invalid credentials
0x530	not permitted to logon at this time
0x531	not permitted to logon from this workstation
0x532	password expired
0x533	account disabled
0x701	account expired
0x773	user must reset password
0x775	account locked out

2.错误码抓包举例

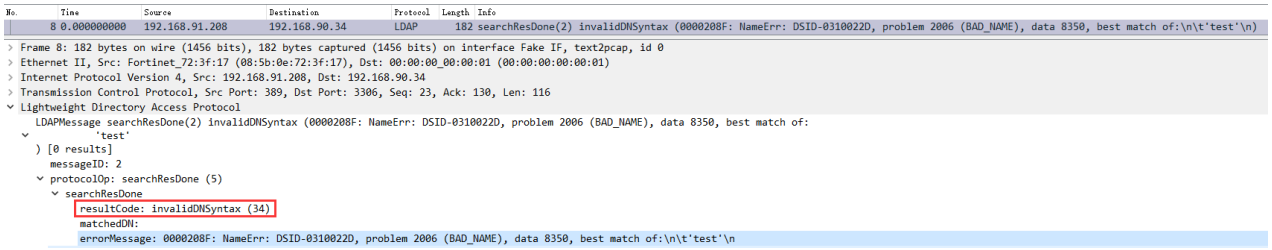
- 密码错误



- 账号禁用



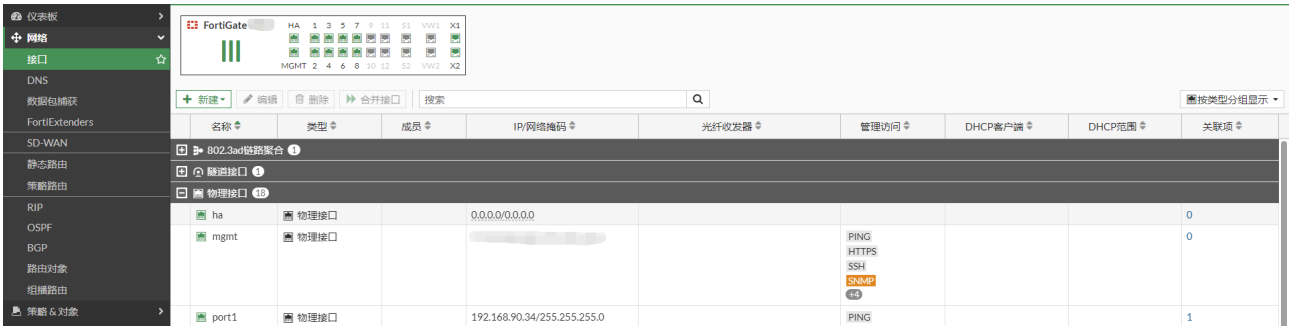
- DN 语法错误

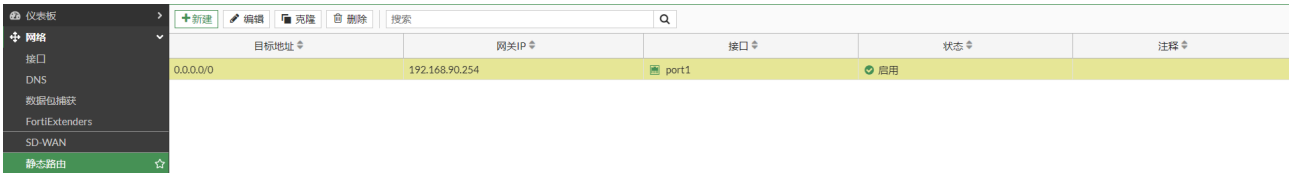

No. Time Source Destination Protocol Length Info
8 0.000000000 192.168.91.208 192.168.90.34 LDAP 182 searchResDone(2) invalidDNSyntax (0000208F: NameErr: DSID-0310022D, problem 2006 (BAD_NAME), data 8350, best match of:\n\t'test'\n)
Frame 8: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface Fake IF, text2pcap, id 0
Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.91.208, Dst: 192.168.90.34
Transmission Control Protocol, Src Port: 389, Dst Port: 3306, Seq: 23, Ack: 130, Len: 116
Lightweight Directory Access Protocol
LDAPMessage searchResDone(2) invalidDNSyntax (0000208F: NameErr: DSID-0310022D, problem 2006 (BAD_NAME), data 8350, best match of:
'test'
) [0 results]
messageID: 2
protocolOp: searchResDone (5)
searchResDone
resultCode: invalidDNSyntax (34)
matchedDN:
errorMessage: 0000208F: NameErr: DSID-0310022D, problem 2006 (BAD_NAME), data 8350, best match of:\n\t'test'\n

LDAP 认证配置与排错

配置 LDAP 认证

1.配置接口地址和路由





ping 测试 FortiGate 与 LDAP 服务器之间的连通性。测试环境使用 Windows AD 作为 LDAP 服务器，地址是 192.168.91.208。

```
# execute ping 192.168.91.208
PING 192.168.91.208 (192.168.91.208): 56 data bytes
64 bytes from 192.168.91.208: icmp_seq=0 ttl=127 time=0.3 ms
64 bytes from 192.168.91.208: icmp_seq=1 ttl=127 time=0.2 ms
64 bytes from 192.168.91.208: icmp_seq=2 ttl=127 time=0.2 ms
64 bytes from 192.168.91.208: icmp_seq=3 ttl=127 time=0.2 ms
64 bytes from 192.168.91.208: icmp_seq=4 ttl=127 time=0.2 ms

--- 192.168.91.208 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

2.配置 LDAP

选择”用户与认证“-->LDAP，点击”新建“。



配置参数：

名称：输入一个名称，不冲突即可；

服务器 IP/名称：LDAP 服务器的 IP 地址；

服务器端口：LDAP 通信端口，默认 TCP 389；

Common Name 标识符：FortiGate 用于识别连接用户的 LDAP 中对象的属性字段。Windows AD 域使用 sAMAccountName，OpenLDAP 使用 uid。这里使用 Windows AD 域；

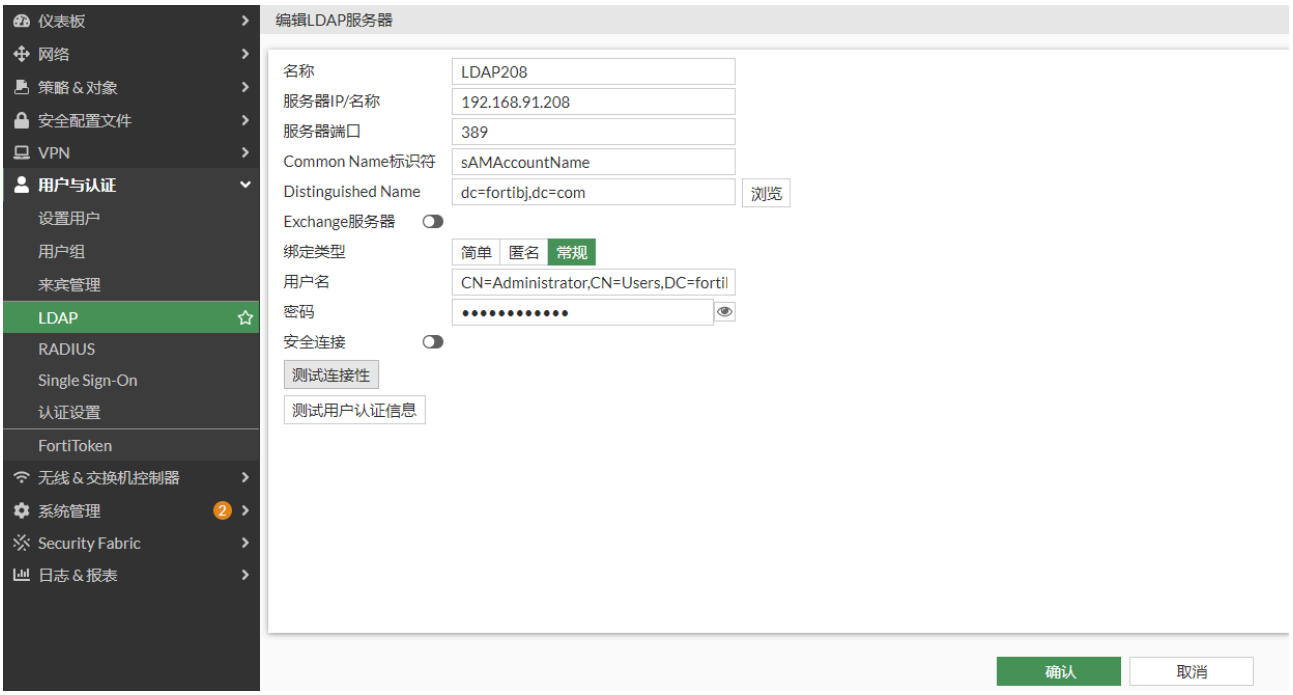
Distinguished Name：用于在 LDAP 服务器上查找用户帐户条目，反映了查找的 CN 标识符之上的 LDAP 的层次结构； 输入 dc = fortibj,dc = com 指定包含所有对象的域根； 输入 ou = tac,dc = fortibj,dc = com 查询指定组织单元下的用户；

绑定类型：选择常规，使用提供的用户名和密码进行绑定，并从 DN 开始搜索并递归遍历子树；

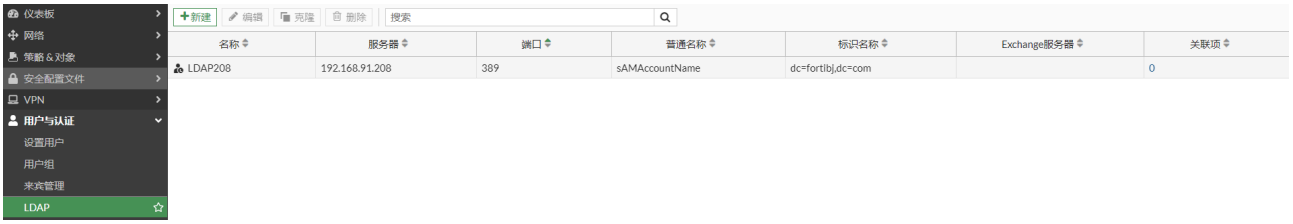
用户名：用户名支持的格式：格式 1：CN = Administrator,CN = Users,DC = fortibj,DC = com；格式 2：Administrator@fortibj.com。

```
C:\Users\Administrator>dsquery user -name administrator
"CN=Administrator,CN=Users,DC=fortibj,DC=com"
```

密码：用户的密码；



点击”确定“，配置完成。

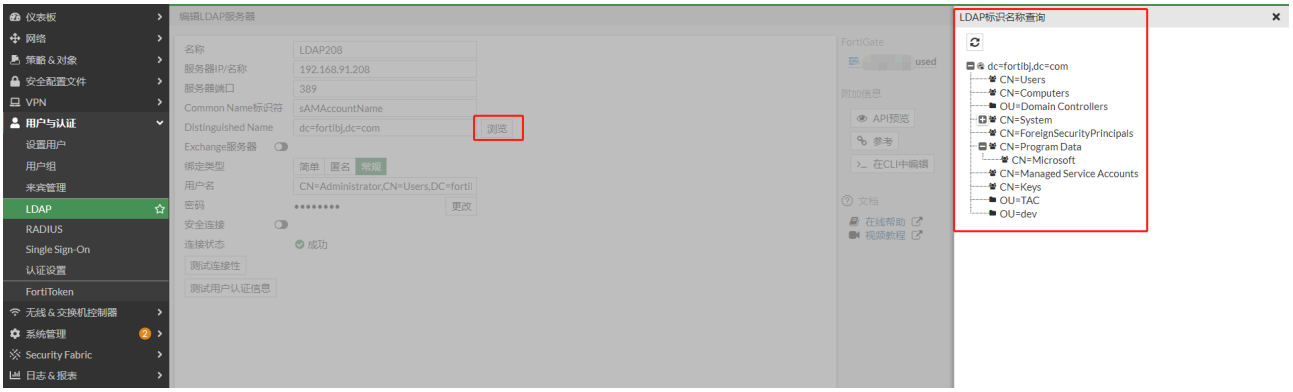


对应的命令行如下：

```
config user ldap
    edit "LDAP208"
        set server "192.168.91.208"
        set cnid "sAMAccountName"
        set dn "dc=fortibj,dc=com"
        set type regular
        set username "CN=Administrator,CN=Users,DC=fortibj,DC=com"
        set password xxxxxxxx
    next
end
```

3.查看 LDAP 目录结构

当 LDAP 配置正确时，可以点击”浏览“，查看 LDAP 目录结构。



LDAP 服务器测试

1.FortiGate 与 LDAP 服务器之间连通性测试

GUI 编辑 LDAP 服务器的配置界面，点击”测试连接性“。

CLI 语法测试连通性：

```
# diagnose test authserver ldap-direct
[IP] [port number]
```

CLI 语法测试用户和密码是否正确：

```
# diagnose test authserver ldap
<server_name> <username> <password>
```

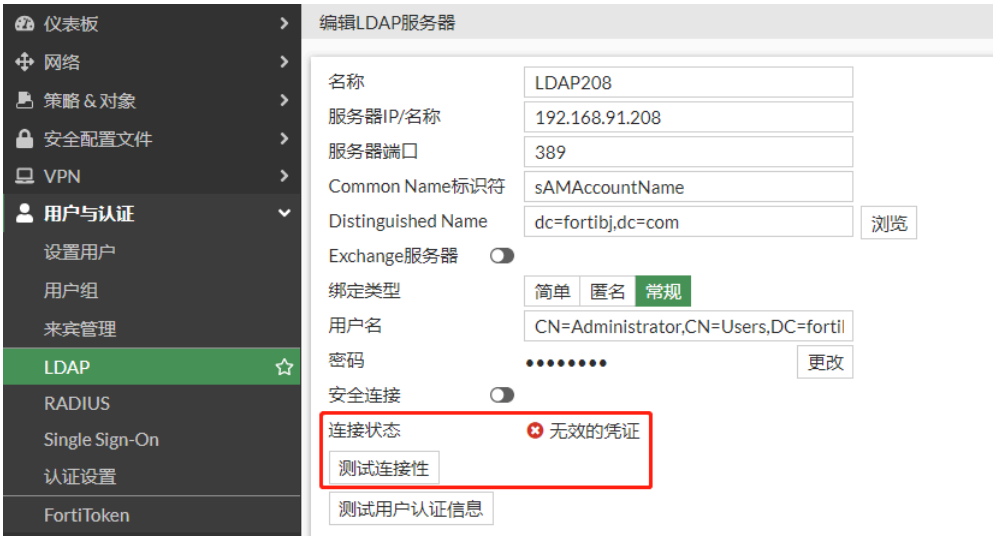
- 如果连接状态是”成功“，则 FortiGate 与 LDAP 服务器之间通信正常。



对应 CLI 的测试结果：

```
# diagnose test authserver ldap-direct 192.168.91.208 389
LDAP server '192.168.91.208' status is OK
```

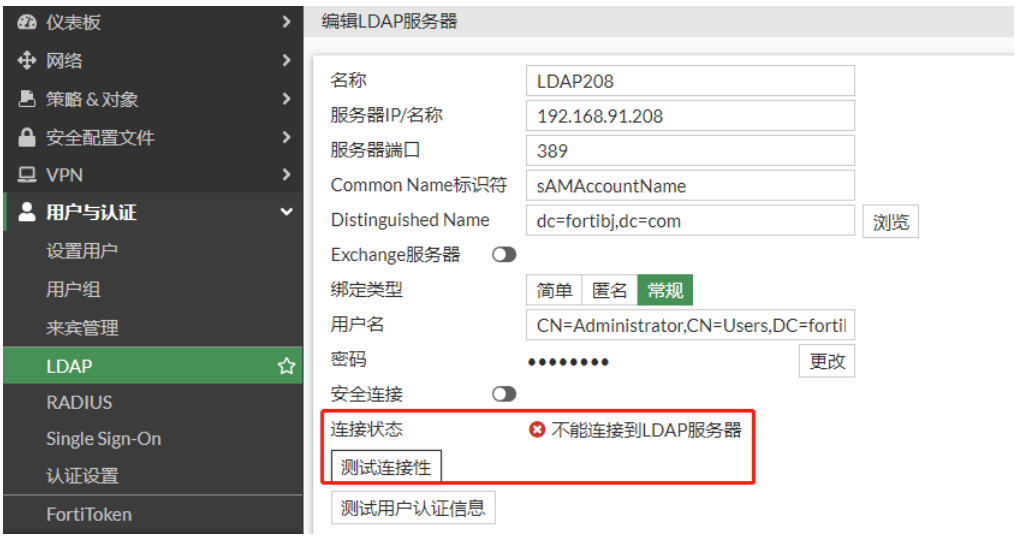
- 如果连接状态是”无效的凭证“，则说明 LDAP 服务器的账号或者密码不正确。



对应的命令行：

```
# diagnose test authserver ldap LDAP208 administrator Test@123456
authenticate 'administrator' against 'LDAP208' succeeded!
Group membership(s) - CN=Group Policy Creator Owners,CN=Users,DC=fortibj,DC=com
                      CN=Domain Admins,CN=Users,DC=fortibj,DC=com
                      CN=Enterprise Admins,CN=Users,DC=fortibj,DC=com
                      CN=Schema Admins,CN=Users,DC=fortibj,DC=com
                      CN=Remote Desktop Users,CN=Builtin,DC=fortibj,DC=com
                      CN=Administrators,CN=Builtin,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

- 如果连接状态是”不能连接到 LDAP 服务器“，则说明到 LDAP 服务器不可达。



对应 CLI 的测试结果：

```
# diagnose test authserver ldap-direct 192.168.91.208 389
LDAP server '192.168.91.208' status is Server unreachable
```

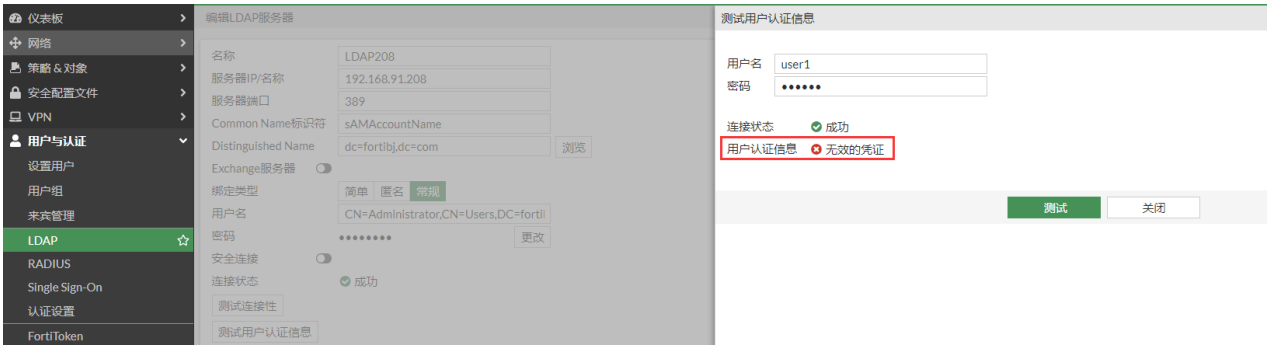
2.测试 LDAP 服务器中的用户及密码

GUI 编辑 Radius 服务器的配置界面，点击”测试用户认证信息“。

CLI 的语法：

```
# diagnose test authserver ldap
<server_name> <username> <password>
```

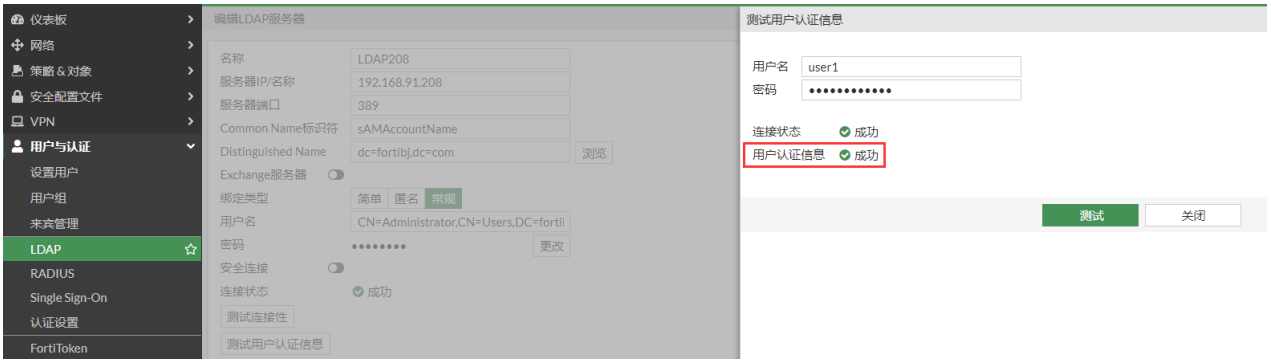
- 用户名或者密码不正确



对应的 CLI 测试结果如下：

```
# diagnose test authserver ldap LDAP208 user1 123456
authenticate 'user1' against 'LDAP208' failed!
```

- 用户名和密码正确



对应的 CLI 测试结果如下，CLI 显示了用户 user1 所在的用户组信息。

```
# diagnose test authserver ldap LDAP208 user1 Pass@123456
authenticate 'user1' against 'LDAP208' succeeded!
Group membership(s) - CN=grp1,CN=Users,DC=fortibj,DC=com
                      CN=Domain Users,CN=Users,DC=fortibj,DC=com
```

其他 LDAP 配置参数

1.修改 LDAP 端口

```
config user ldap
    edit LDAP208
        set port xx
    next
end
```

2.指定原 IP 地址

```
config user ldap
    edit LDAP208
        set source-ip x.x.x.x
    next
end
```

创建 ldap 用户

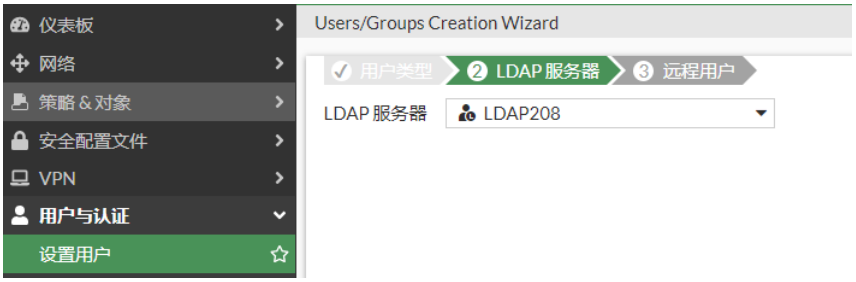
1. 选择“用户与认证”-->“设置用户”，点击新建



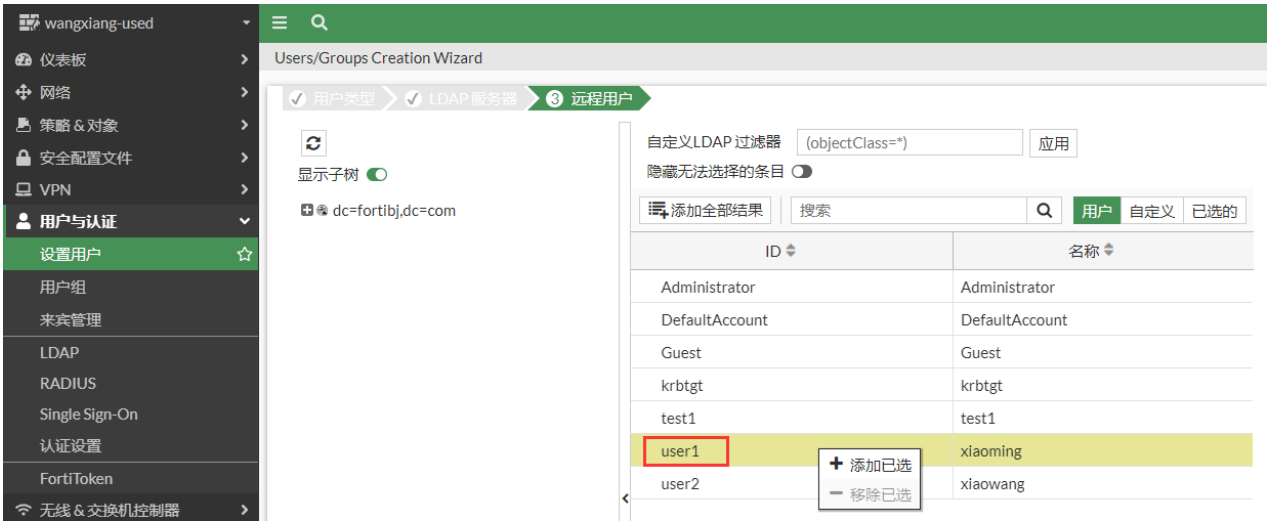
2. 选择 LDAP 用户，点击“下一步”



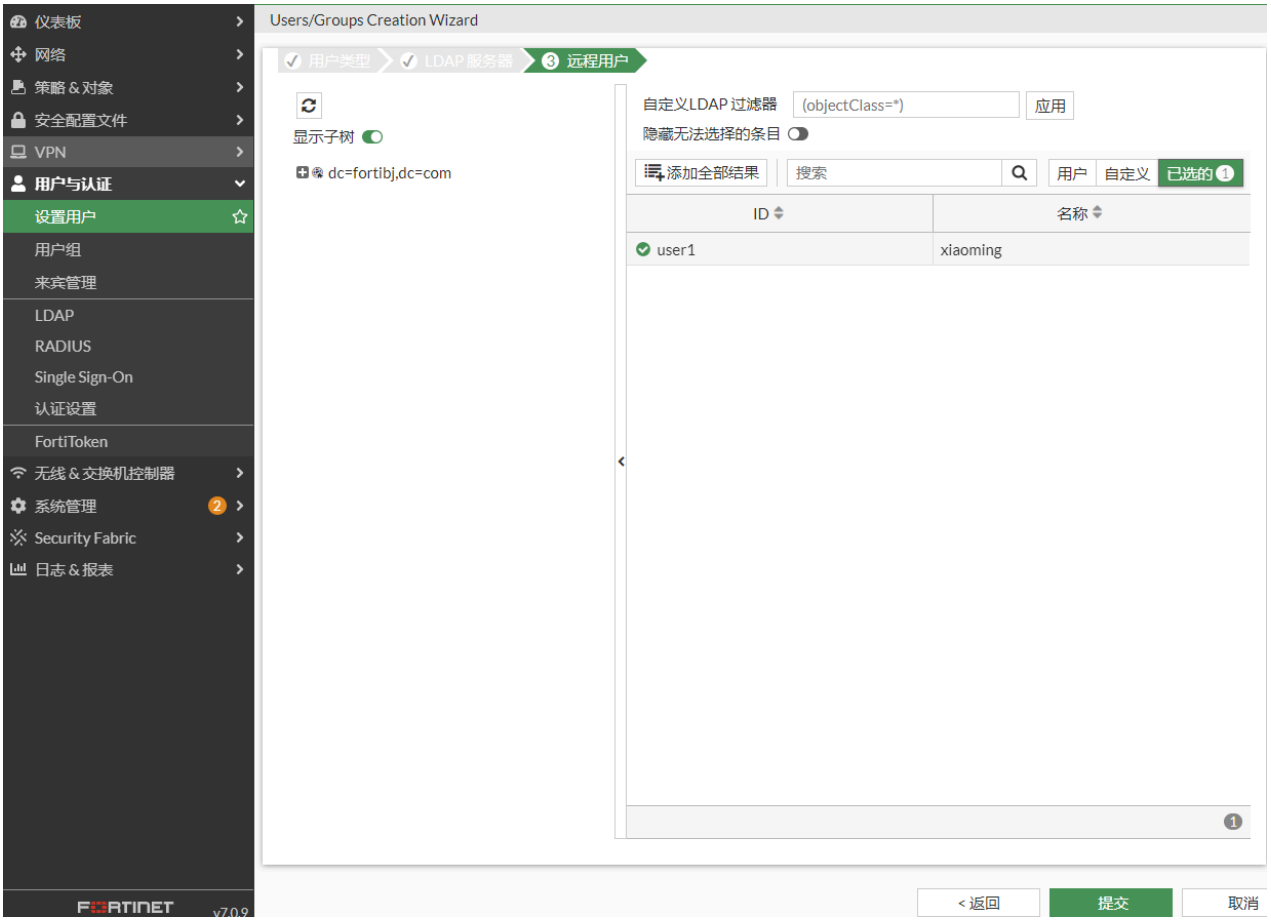
3. 选择配置好的 LDAP 服务器，点击“下一步”



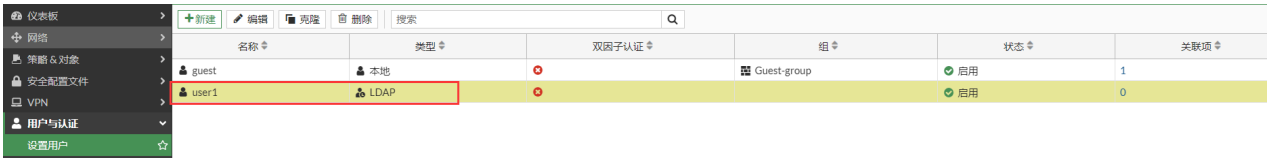
4. 这里添加 user1，右击 user1，点击“添加已选”



点击“已选的”，查看选中的用户，确认无误后，点击“提交”。

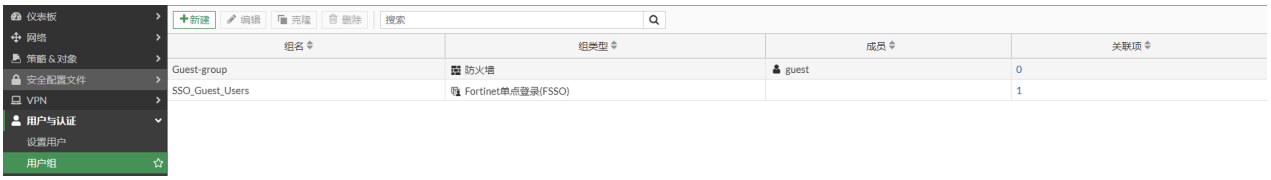


5. 完成 LDAP 用户创建



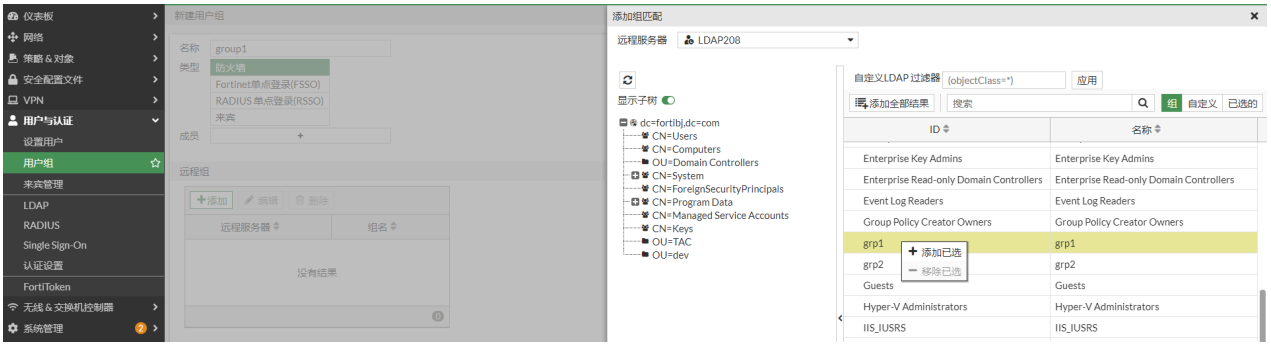
创建 ldap 用户组

1. 选择“用户与认证”-->“用户组”，点击“新建”

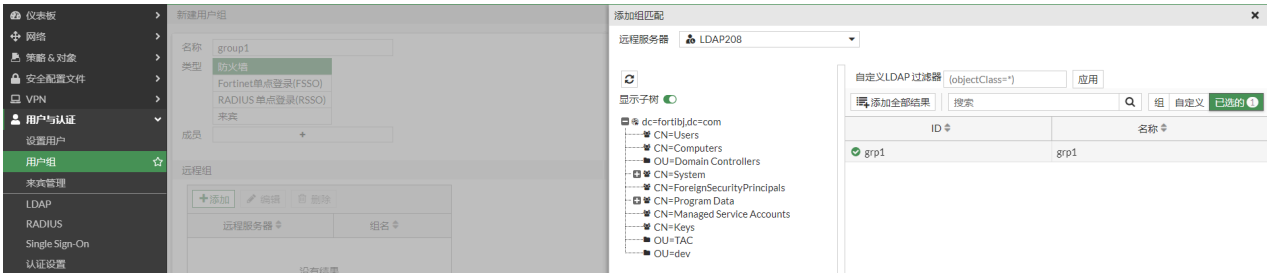


2. 配置 LDAP

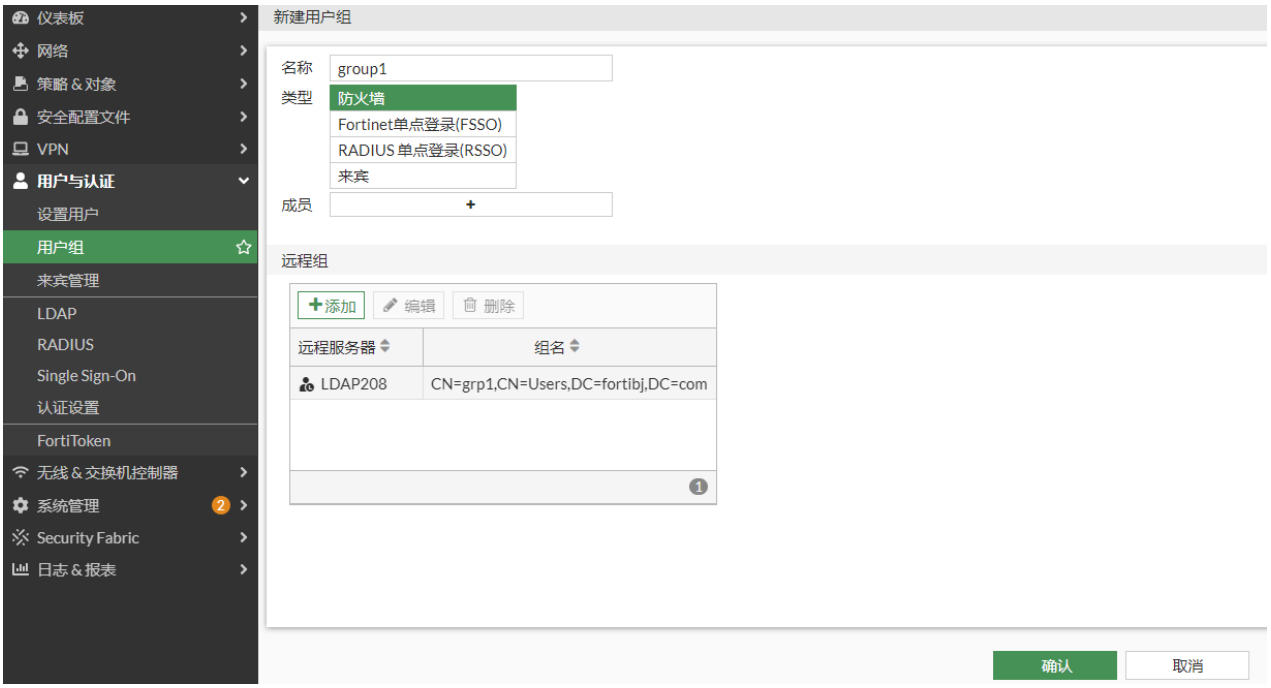
设置 FortiGate 用户组名称 group1，在远程组中，点击“添加”，选择配置好的远程服务器，会自动显示出目录树，找到需要添加的用户组，这里选择 grp1，右击“添加已选”，**那么 group1 匹配远端 LDAP 用户组 grp1 中的用户**。可以添加多个用户组，如再增加 grp2，那么 group1 匹配远端 LDAP 用户组 grp1 和 grp2 中的用户。



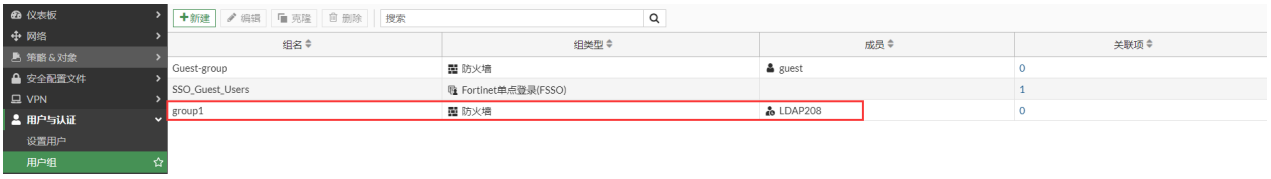
点击“已选的”，查看选中的用户组，确认无误后，点击“确认”。



3. 点击确认

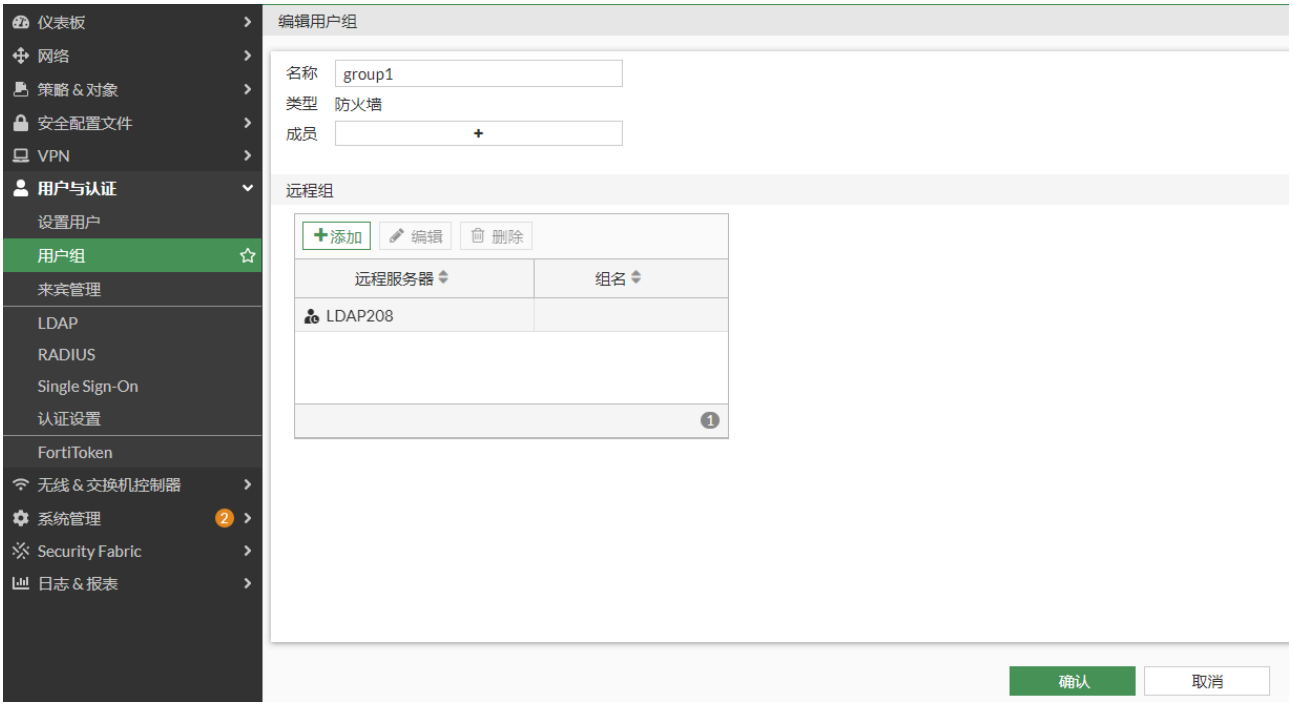


4. 配置完成



重要

如果在 group1 用不选择任何用户组，那么 group1 匹配 LDAP 服务器配置的 DN 目录下的所有用户。



LDAP 认证 debug

认证成功

1. 准备查询的用户名 user1 和 base DN。

```
# diagnose debug application fnbamd -1
# diagnose debug  enable

[1906] handle_req-Rcvd auth req 730526726 for user1 in LDAP208 opt=0000001b prot=0
[466] __compose_group_list_from_req-Group 'LDAP208', type 1
[616] fnbamd_pop3_start-user1
[989] __fnbamd_cfg_get_ldap_list_by_server-
[995] __fnbamd_cfg_get_ldap_list_by_server-Loaded LDAP server 'LDAP208'
[1150] fnbamd_cfg_get_ldap_list-Total ldap servers to try: 1

[1717] fnbamd_ldap_init-search filter is: sAMAccountName=user1
[1727] fnbamd_ldap_init-search base is: dc=fortibj,dc=com

[1149] __fnbamd_ldap_dns_cb-Resolved LDAP208:192.168.91.208 to 192.168.91.208, cur stack size:1
[924] __fnbamd_ldap_get_next_addr-
[1154] __fnbamd_ldap_dns_cb-Connection starts LDAP208:192.168.91.208, addr 192.168.91.208
[879] __fnbamd_ldap_start_conn-Still connecting 192.168.91.208.
[633] create_auth_session-Total 1 server(s) to try
[1107] __ldap_connect-tcps_connect(192.168.91.208) is established.
```

2. LDAP 管理员账号绑定。

```
[985] __ldap_rxtx-state 3(Admin Binding)
[363] __ldap_build_bind_req-Binding to 'CN=Administrator,CN=Users,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 69 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 1

[985] __ldap_rxtx-state 4(Admin Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:1, type:bind
response-ret=0    //绑定成功
[1023] fnbamd_ldap_parse_response-ret=0
```

3. 查询 user1 的 DN。

```
[1052] __ldap_rxtx-Change state to 'DN search'
[985] __ldap_rxtx-state 11(DN search)
[750] fnbamd_ldap_build_dn_search_req-base:'dc=fortibj,dc=com' filter:sAMAccountName=user1
[1083] fnbamd_ldap_send-sending 73 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 2

[985] __ldap_rxtx-state 12(DN search resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 51
[1306] fnbamd_ldap_recv-Response len: 53, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-entry
```

4. 查询成功，获得 user1 的 DN 是 xiaoming,OU = TAC,DC = fortibj,DC = com。

```
response-ret=0    //查询成功，获得user1的DN是 xiaoming,OU=TAC,DC=fortibj,DC=com
[1023] fnbamd_ldap_parse_response-ret=0
[1225] __fnbamd_ldap_dn_entry-Get DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'

[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 62
[1306] fnbamd_ldap_recv-Response len: 64, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-result
[1023] fnbamd_ldap_parse_response-ret=0
```

5. 用户 user1 绑定请求成功。

```
[1052] __ldap_rxtx-Change state to 'User Binding'
[985] __ldap_rxtx-state 5(User Binding)
[596] fnbamd_ldap_build_userbind_req-Trying DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[363] __ldap_build_bind_req-Binding to 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 93 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 3

[985] __ldap_rxtx-state 6(User Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:3, type:bind

response-ret=0 //绑定成功
[1023] fnbamd_ldap_parse_response-ret=0
```

6. memberOf 属性查询。

```
[1052] __ldap_rxtx-Change state to 'Attr query'
[985] __ldap_rxtx-state 7(Attr query)
[649] fnbamd_ldap_build_attr_search_req-Adding attr 'memberOf'
[661] fnbamd_ldap_build_attr_search_req-base:'CN=xiaoming,OU=TAC,DC=fortibj,DC=com' filter:cn=*
[1083] fnbamd_ldap_send-sending 112 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 4

[985] __ldap_rxtx-state 8(Attr query resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 195
[1306] fnbamd_ldap_recv-Response len: 197, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:4, type:search-entry
```

7. 查询成功，获得 user1 的用户组 CN = grp1,CN = Users,DC = fortibj,DC = com。

```
[1023] fnbamd_ldap_parse_response-ret=0
[556] __get_member_of_groups-Get the memberOf groups.
[522] __retrieve_group_values-Get the memberOf groups.
[532] __retrieve_group_values- attr='memberOf', found 1 values
[542] __retrieve_group_values-val[0]='CN=grp1,CN=Users,DC=fortibj,DC=com'

[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:4, type:search-result
[1023] fnbamd_ldap_parse_response-ret=0
[1305] __fnbamd_ldap_attr_next-Entering CHKPRIMARYGRP state
```

8. Primary group 属性查询。

```
[1052] __ldap_rxtx-Change state to 'Primary group query'
[985] __ldap_rxtx-state 13(Primary group query)
[685] fnbamd_ldap_build_primary_grp_search_req-starting primary group check ...
[689] fnbamd_ldap_build_primary_grp_search_req-number of sub auths 5
[707] fnbamd_ldap_build_primary_grp_search_req-base:'dc=fortibj,dc=com' filter:(&(objectclass=group)
(objectSid=\01\05\00\00\00\00\00\05\15\00\00\00\b8\17\0b\a2\b5\db\a4\d8\c0\a2\54\19\01\02\00\00))
[1083] fnbamd_ldap_send-sending 121 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 5

[985] __ldap_rxtx-state 14(Primary group query resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 110
[1306] fnbamd_ldap_recv-Response len: 112, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-entry
```

9. 获得 Primary group 是 CN = Domain Users,CN = Users,DC = fortibj,DC = com。

```
[1023] fnbamd_ldap_parse_response-ret=0
[472] __get_one_group-group: CN=Domain Users,CN=Users,DC=fortibj,DC=com

[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 62
[1306] fnbamd_ldap_recv-Response len: 64, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-reference
```

```
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:5, type:search-result
[1023] fnbamd_ldap_parse_response-ret=0
[1433] __fnbamd_ldap_primary_grp_next-Auth accepted
```

10. 查询完成。

```
[1052] __ldap_rxtx-Change state to 'Done'
[985] __ldap_rxtx-state 23(Done)
[1083] fnbamd_ldap_send-sending 7 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 6
[785] __ldap_done-svr 'LDAP208'
[755] __ldap_destroy-
[724] __ldap_stop-Conn with 192.168.91.208 destroyed.
```

11. 查询的结果。

```
[2678] fnbamd_ldap_result-Result for ldap svr 192.168.91.208(LDAP208) is SUCCESS
[401] ldap_copy_grp_list-copied CN=grp1,CN=Users,DC=fortibj,DC=com
[401] ldap_copy_grp_list-copied CN=Domain Users,CN=Users,DC=fortibj,DC=com

[2693] fnbamd_ldap_result-Skipping group matching
[216] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 730526726, len=2227
[789] destroy_auth_session-delete session 730526726
[755] __ldap_destroy-
[1764] fnbamd_ldap_auth_ctx_free-Freeing 'LDAP208' ctx
```

认证失败

1. 准备查询的用户名 user1 和 base DN。

```
# diagnose debug application fnbamd -1
# diagnose debug  enable

[1906] handle_req-Rcvd auth req 730526741 for user1 in LDAP208 opt=0000001b prot=0
[466] __compose_group_list_from_req-Group 'LDAP208', type 1
[616] fnbamd_pop3_start-user1
[989] __fnbamd_cfg_get_ldap_list_by_server-
[995] __fnbamd_cfg_get_ldap_list_by_server-Loaded LDAP server 'LDAP208'
[1150] fnbamd_cfg_get_ldap_list-Total ldap servers to try: 1

[1717] fnbamd_ldap_init-search filter is: sAMAccountName=user1
[1727] fnbamd_ldap_init-search base is: dc=fortibj,dc=com

[1149] __fnbamd_ldap_dns_cb-Resolved LDAP208:192.168.91.208 to 192.168.91.208, cur stack size:1
[924] __fnbamd_ldap_get_next_addr-
[1154] __fnbamd_ldap_dns_cb-Connection starts LDAP208:192.168.91.208, addr 192.168.91.208
[879] __fnbamd_ldap_start_conn-Still connecting 192.168.91.208.
[633] create_auth_session-Total 1 server(s) to try
[1107] __ldap_connect-tcps_connect(192.168.91.208) is established.
```

2. LDAP 管理员账号绑定成功。

```
[985] __ldap_rxtx-state 3(Admin Binding)
[363] __ldap_build_bind_req-Binding to 'CN=Administrator,CN=Users,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 69 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 1

[985] __ldap_rxtx-state 4(Admin Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
```

```
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:1, type:bind

response-ret=0    //绑定成功
[1023] fnbamd_ldap_parse_response-ret=0
```

3. 查询 user1 的 DN。

```
[1052] __ldap_rxtx-Change state to 'DN search'
[985] __ldap_rxtx-state 11(DN search)
[750] fnbamd_ldap_build_dn_search_req-base:'dc=fortibj,dc=com' filter:sAMAccountName=user1
[1083] fnbamd_ldap_send-sending 73 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 2

[985] __ldap_rxtx-state 12(DN search resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 51
[1306] fnbamd_ldap_recv-Response len: 53, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-entry
```

4. 查询成功，获得 user1 的 DN 是 xiaoming,OU = TAC,DC = fortibj,DC = com。

```
response-ret=0    //查询成功，获得user1的DN是 xiaoming,OU=TAC,DC=fortibj,DC=com
[1023] fnbamd_ldap_parse_response-ret=0
[1225] __fnbamd_ldap_dn_entry-Get DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'

[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 78
[1306] fnbamd_ldap_recv-Response len: 80, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 62
[1306] fnbamd_ldap_recv-Response len: 64, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-reference
[1023] fnbamd_ldap_parse_response-ret=0
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 14
[1306] fnbamd_ldap_recv-Response len: 16, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-result
[1023] fnbamd_ldap_parse_response-ret=0
```

5. 用户 user1 绑定请求。

```
[1052] __ldap_rxtx-Change state to 'User Binding'
[985] __ldap_rxtx-state 5(User Binding)
[596] fnbamd_ldap_build_userbind_req-Trying DN 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[363] __ldap_build_bind_req-Binding to 'CN=xiaoming,OU=TAC,DC=fortibj,DC=com'
[1083] fnbamd_ldap_send-sending 88 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 3

[985] __ldap_rxtx-state 6(User Bind resp)
[1127] __fnbamd_ldap_read-Read 8
[1233] fnbamd_ldap_recv-Leftover 2
[1127] __fnbamd_ldap_read-Read 102
[1306] fnbamd_ldap_recv-Response len: 104, svr: 192.168.91.208
[987] fnbamd_ldap_parse_response-Got one MESSAGE. ID:3, type:bind
```

6. 绑定失败，响应码 49，响应消息 52e，即密码不对。

```
[1009] fnbamd_ldap_parse_response-Error 49(80090308: LdapErr: DSID-0C0903D3, comment: AcceptSecurityContext error, data
52e, v3839)
[1023] fnbamd_ldap_parse_response-ret=49

[262] fnbamd_ldap_more_dn_left-idx:0, total:1
[1052] __ldap_rxtx-Change state to 'Done'
[985] __ldap_rxtx-state 23(Done)
[1083] fnbamd_ldap_send-sending 7 bytes to 192.168.91.208
[1096] fnbamd_ldap_send-Request is sent. ID 4
[785] __ldap_done-svr 'LDAP208'
[755] __ldap_destroy-
[724] __ldap_stop-Conn with 192.168.91.208 destroyed.
[216] fnbamd_comm_send_result-Sending result 1 (nid 0) for req 730526741, len=2148
[789] destroy_auth_session-delete session 730526741
[755] __ldap_destroy-
[1764] fnbamd_ldap_auth_ctx_free-Freeing 'LDAP208' ctx
```

LDAP 常见错误代码

LDAP 错误代码位置

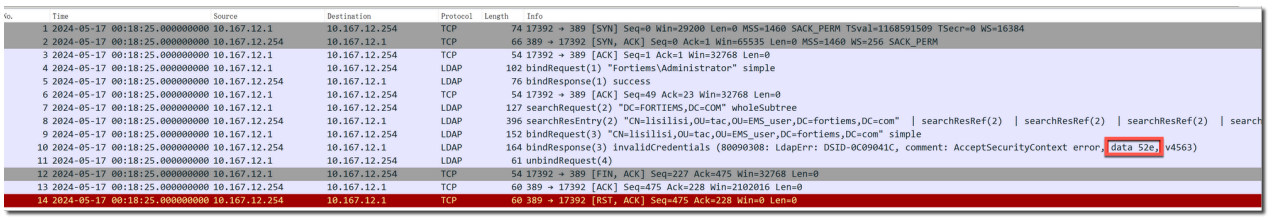
当通过 FortiGate 的 LDAP 认证失败时（例如显式代理认证、SSLVPN 认证、ZTNA 认证），可以通过在 FortiGate 上执行 `diagnose debug application fnbamd -1` 或通过抓取 LDAP 认证报文来排查错误原因。

`fnbamd` 的 Debug 信息或 LDAP 认证结果报文中会包含的 LDAP 服务器返回的错误代码，从错误代码代表的含义，可以快速定位到 LDAP 认证失败的原因。

- 如下所示的 `fnbamd` 的 Debug 信息，`[778] fnbamd_ldap_parse_response-Error 49(80090308: LdapErr: DSID-0C090446, comment: AcceptSecurityContext error, data 52e, v4563)` 中的 `data 52e` 即为错误代码”。

```bash
FortiGate # diagnose debug application fnbamd -1
FortiGate # diagnose debug enable

[584] fnbamd_ldap_build_dn_search_req-base:'DC=fortinet,dc=com' filter:sAMAccountName=xxxxx
[1100] __fnbamd_ldap_dn_entry-Get DN 'CN=XXX,CN=XXX,DC=XXX,DC=com'
[90] ldap_dn_list_add-added CN=XXX,CN=XXX,DC=XXX,DC=com
[52] ldap_dn_list_del_all-Del CN=XXX,CN=XXX,DC=XXX,DC=com
[2821] fnbamd_ldap_result-Result for ldap svr XXX.XXX.XXX.XXX is SUCCESS
[1552] fnbamd_ldap_init-search filter is: sAMAccountName=XXX
[1561] fnbamd_ldap_init-search base is: DC=XXX,dc=com
[584] fnbamd_ldap_build_dn_search_req-base:'DC=XXX,dc=com' filter:sAMAccountName=XXX
[1100] __fnbamd_ldap_dn_entry-Get DN 'CN=XXX,OU=XXX,OU=XXX,OU=XXX,OU=XXX,DC=XXX,DC=com'
[90] ldap_dn_list_add-added CN=XXX,OU=XXX,OU=XXX,OU=XXX,OU=XXX,DC=XXX,DC=com
[429] fnbamd_ldap_build_userbind_req-Trying DN ' CN=XXX,OU=XXX,OU=XXX,OU=XXX,OU=XXX,DC=XXX,DC=com '
[196] __ldap_build_bind_req-Binding to ' CN=XXX,OU=XXX,OU=XXX,OU=XXX,OU=XXX,DC=XXX,DC=com'
[852] fnbamd_ldap_send-sending 123 bytes to XXX.XXX.XXX.XXX
[864] fnbamd_ldap_send-Request is sent. ID 3
[815] __ldap_rxtx-state 6(User Bind resp)
[895] __fnbamd_ldap_read-Read 8
[895] __fnbamd_ldap_read-Read 102
[1075] fnbamd_ldap_recv-Response len: 104, svr: XXX.XXX.XXX.XXX
[756] fnbamd_ldap_parse_response-Got one MESSAGE. ID:3, type:bind
[778] fnbamd_ldap_parse_response-Error 49(80090308: LdapErr: DSID-0C090446, comment: AcceptSecurityContext error, data 52e, v4563)
//这里的“data 52e”，即为错误代码//
[791] fnbamd_ldap_parse_response-ret=49
[882] __ldap_rxtx-Change state to 'User Binding'
[815] __ldap_rxtx-state 5(User Binding)
[425] fnbamd_ldap_build_userbind_req-No more DN left
[737] __ldap_error-
[726] __ldap_stop-svr 'ad_server'
[52] ldap_dn_list_del_all-Del CN=XXX,OU=XXX,OU=XXX,OU=XXX,OU=XXX,DC=XXX,DC=com
[182] fnbamd_comm_send_result-Sending result 1 (error 0, nid 0) for req 1824141631
[653] destroy_auth_session-delete session 1824141631
```

- 使用 `diagnose sniffer packet any 'tcp and port 389' 6 0 l`（或 GUI 抓包）抓取 LDAP 报文，也可以看到类似的错误代码信息。



LDAP 错误代码含义

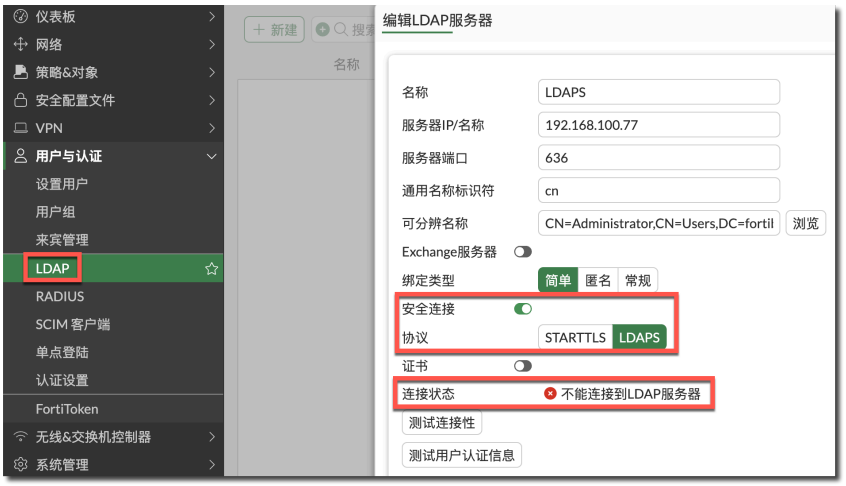
- `525 - User not found`：当提供的用户名无效时，将返回此错误，表明 LDAP 目录中不存在指定的用户。
- `52e - Invalid credentials`：表示提供了有效的用户名，但提供的密码或凭据不正确。此错误通常会阻止显示其他错误，因为没有有效凭据身份验证无法继续。
- `530 - Not permitted to logon at this time`：当在限制登录期间提供有效用户名和密码时，将返回此错误。可能存在基于时间的访问限制。

- `531 - Not permitted to logon from this workstation`：当提供有效的用户名和密码时，将返回该错误，但用户被限制使用尝试登录的工作站。基于工作站的访问限制生效。
- `532 - Password expired`：当提供了有效的用户名并且提供的密码正确但已过期时，会出现此错误。用户需要更改他们的密码。
- `533 - Account disabled`：当提供了有效的用户名和密码，但用户的帐户已被禁用时返回。由于帐户的禁用状态，身份验证被阻止。
- `701 - Account expired`：当提供了有效的用户名和密码，但用户的帐户已过期，无法成功进行身份验证时，将返回此错误。
- `773 - User must reset password`：如果提供了有效的用户名和密码，此错误表示用户需要在首次登录前或管理员重置密码后立即重置密码。
- `775 - Account locked out`：当提供了有效的用户名，但用户的帐户由于登录尝试失败次数过多而被锁定时，将返回此错误。请务必注意，即使提供的密码有效，也会返回此错误。

LDAPS CA 强制认证

功能介绍

自 FortiOS 7.4.4 版本开始，FortiGate 在 LDAPS/STARTTLS 配置中实施了证书颁发者强制执行。也就是说，服务器的证书颁发者证书（根证书）必须安装在 FortiGate 上，否则认证将会失败。




```bash
diagnose debug app fnbamd -1
diagnose debug enable

[433] start_remote_auth-Total 1 server(s) to try
[1881] handle_req-r=4
[1378] __ldap_tcps_connect-Start ldap conn timer.
[1666] __verify_cb-Cert error 20, unable to get local issuer certificate. Depth 0. Subject '/CN=ion-kvm88.labtest.local'
[1345] __ldap_tcps_connect-tcps_connect(10.192.11.88) failed: ssl_connect() failed: 167772294 (error:0A000086:SSL
routines::certificate verify failed).
[1642] __ldap_error-Ret 5, st = 0.
[1679] __ldap_error-
[1485] __ldap_tcps_close-closed.
[1567] __ldap_conn_stop-Stop ldap conn timer.
```

即使没有选择“服务器证书”（`ca-cert`），FortiGate 也要求验证 LDAP 服务器的 CA 证书。

```bash
config user ldap
    edit "LDAPS"
        set server "192.168.100.177"
        set server-identity-check disable
        set cnid "cn"
        set dn "CN=Administrator,CN=Users,DC=fortibj,DC=com"
        set type regular
        set username "adminfw"
        set password xxxxx
        set secure ldaps
        set ca-cert ""
        set port 636
    next
end
```

解决方法

为避免出现此错误，需要将签署服务器证书的根证书导入到 FortiGate 证书存储库中。可以通过以下步骤完成：

1. 导入根 CA 证书到 FortiGate（步骤请参考：系统管理 → 系统设置 → 证书管理 ↗章节）。

2. 重新测试 LDAPS/STARTTLS 认证连接。

3. 在 `fnbamd` 的 debug 信息中，服务器证书（深度 0）和根 CA（深度 1）都已正确安装。

```bash
diagnose debug app fnbamd -1
diagnose debug enable

[433] start_remote_auth-Total 1 server(s) to try
[1881] handle_req-r=4
[1378] __ldap_tcps_connect-Start ldap conn timer.
[1669] __verify_cb-Cert preverify ok. Depth 1. Subject '/DC=local/DC=labtest/CN=labtest-ION-KVM88-CA'
[1669] __verify_cb-Cert preverify ok. Depth 0. Subject '/CN=ion-kvm88.labtest.local'
[1378] __ldap_tcps_connect-Start ldap conn timer.
[1363] __ldap_tcps_connect-tcps_connect(10.192.11.88) is established. Current state: Connecting.
[1120] __ldap_auth_ctx_reset-
[984] __ldap_next_state-State: Connecting → Admin Binding
[1378] __ldap_tcps_connect-Start ldap conn timer.
[1221] __ldap_rxtx-fd 10, state 2(Admin Binding)
[1223] __ldap_rxtx-Stop ldap conn timer.
```

注意事项

如果在导入了根证书后仍然出现 `Unknown CA` 错误，请确保根证书被正确安装到远程 CA 存储区，而不是本地 CA 存储区。若证书安装在本地 CA 存储区，认证会失败。

Radius 认证

- Radius 认证原理
- Radius 认证配置与排错

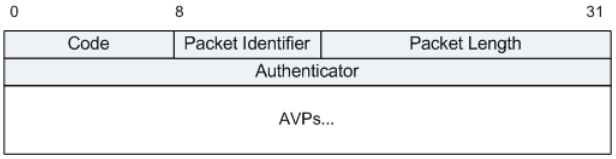
Radius 认证原理

Radius 简介

RADIUS（Remote Authentication Dial In User Service，远程用户拨号认证）是一种广泛支持的客户机 - 服务器协议，提供集中的身份验证、授权和计费功能。RADIUS 服务器使用 UDP 报文与网络中的 RADIUS 客户端通信，在允许用户访问网络之前对用户进行认证，授权适当的用户访问资源，并对所使用的资源进行说明。

Radius 报文格式及交互过程

1.Radius 报文格式



Code：表示 Radius 报文的类型：

- Code = 1: Access-Request，认证请求报文；
- Code = 2: Access-Accept，认证成功报文；
- Code = 3: Access-Reject，认证拒绝报文；
- Code = 4: Accounting-Request，计费请求报文；
- Code = 5: Accounting-Response，计费响应报文；
- Code = 11: Access-Challenge，认证挑战报文；
- Code = 255: Reserved，保留；

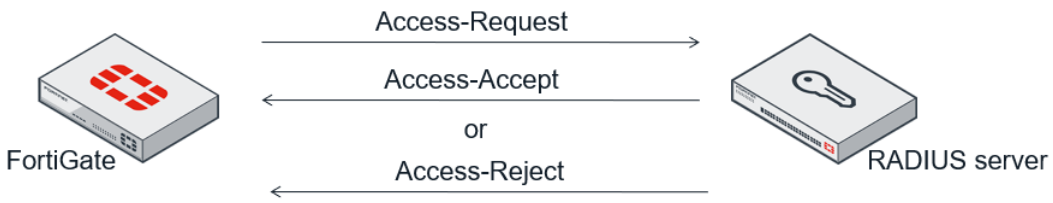
Packet Identifier：用于匹配请求和回应报文。

Packet Length：Radius 报文长度；

Authenticator：用于认证来自服务端的响应，也用于用户密码的加密处理；

AVPs：属性字段，用于 RADIUS 服务器和 RADIUS 客户端之间的信息交换，Fortinet RADIUS 属性的供应商 ID 是 12356。

2.Radius 报文交互过程



- FortiGate 向 RADIUS 服务器发送包含用户名和密码信息的认证请求报文。
- RADIUS 服务器对用户身份的合法性进行检验：

  如果用户身份合法，RADIUS 服务器向 FortiGate 返回认证接受报文。

  如果用户身份不合法，RADIUS 服务器向 RADIUS 客户端返回认证拒绝报文。

PAP 和 CHAP 认证

1.PAP 认证

在认证请求报文 Access-Request 报文中，PAP 将用户和加密后的密码传递给 Radius 服务器。

```
No.  Time          Source          Destination     Protocol  Length  Info
   1 0.000000000   192.168.90.34   192.168.91.209  RADIUS     103    Access-Request id=7
   2 0.000000000   192.168.91.209  192.168.90.34   RADIUS      74    Access-Accept id=7

> Frame 1: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface Fake IF, text2pc
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.209
> User Datagram Protocol, Src Port: 1716, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x7 (7)
    Length: 61
    Authenticator: 6aae17d4b66d029697aab86fd4ab6ed2
    [The response to this request is in frame 2]
  v Attribute Value Pairs
    v AVP: t=NAS-Identifier(32) l=16 val=wangxiang-used
        Type: 32
        Length: 16
        NAS-Identifier: wangxiang-used
    v AVP: t=User-Name(1) l=7 val=user1
        Type: 1
        Length: 7
        User-Name: user1
    v AVP: t=User-Password(2) l=18 val=Encrypted
        Type: 2
        Length: 18
        User-Password (encrypted): 50527af24b2f150fd2a0157eb59c6c5e
```

Radius 服务器将解密后的密码，与用户的密码进行比较，如果一致， 则认为密码正确。

```
No.  Time          Source          Destination     Protocol  Length  Info
   1 0.000000000   192.168.90.34   192.168.91.209  RADIUS     103    Access-Request id=7
   2 0.000000000   192.168.91.209  192.168.90.34   RADIUS      74    Access-Accept id=7

> Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.209, Dst: 192.168.90.34
> User Datagram Protocol, Src Port: 1812, Dst Port: 1716
v RADIUS Protocol
    Code: Access-Accept (2)
    Packet identifier: 0x7 (7)
    Length: 32
    Authenticator: f59759669f7cb64a4bda4e2eef863af4
    [This is a response to a request in frame 1]
    [Time from request: 0.000000000 seconds]
  v Attribute Value Pairs
    v AVP: t=Vendor-Specific(26) l=12 vnd=Fortinet, Inc.(12356)
        Type: 26
        Length: 12
        Vendor ID: Fortinet, Inc. (12356)
      v VSA: t=Fortinet-Group-Name(1) l=6 val=grp1
          Type: 1
          Length: 6
          Fortinet-Group-Name: grp1
```

2.CHAP 认证

> **重要**
>
> CHAP 认证传递的是一个 HASH 值，不是密码，因此这要求用户的密码保存在 Radius 服务器本地。如果 Radius 服务器用户的密码不在本地，如 FortiGate----Radius 服务器 ----LDAP 服务器，像这样的部署方式，用户的密码保存在 LDAP 服务器中，Radius 服务器无法获取 LDAP 服务器中用户的密码，因此认证将失败。

在认证请求报文 Access-Request 报文中，CHAP 认证并不会直接在认证过程传递用户密码， 而是传递 chapID + 密码 + challenge 的 MD5 值。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.209	RADIUS	122	Access-Request id=9
2	0.000000000	192.168.91.209	192.168.90.34	RADIUS	74	Access-Accept id=9

```
> Frame 1: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_ee:f2:04 (e8:1c:ba:ee:f2:04), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.90.34, Dst: 192.168.91.209
> User Datagram Protocol, Src Port: 12269, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x9 (9)
    Length: 80
    Authenticator: 541baba7049951ac53476ecb1f2f9918
    [The response to this request is in frame 2]
  v Attribute Value Pairs
    v AVP: t=NAS-Identifier(32) l=16 val=wangxiang-used
        Type: 32
        Length: 16
        NAS-Identifier: wangxiang-used
    v AVP: t=User-Name(1) l=7 val=user1
        Type: 1
        Length: 7
        User-Name: user1
    v AVP: t=CHAP-Password(3) l=19 val=c537f2e59792a37b600318a39c7896ce2f
        Type: 3
        Length: 19
      v CHAP-Password: c537f2e59792a37b600318a39c7896ce2f
          CHAP Ident: 0xc5
          CHAP String: 37f2e59792a37b600318a39c7896ce2f
    v AVP: t=CHAP-Challenge(60) l=18 val=000933e8af645c56ab1437032c0fd2f6
        Type: 60
        Length: 18
        CHAP-Challenge: 000933e8af645c56ab1437032c0fd2f6
```

Radius 服务器通过用户名找到对应的密码，计算 chapID + 密码 + challenge 的 MD5 值， 并判断是否与 chap-password 一致（从第 2 个字节开始比较），如果一致， 则认为密码正确。

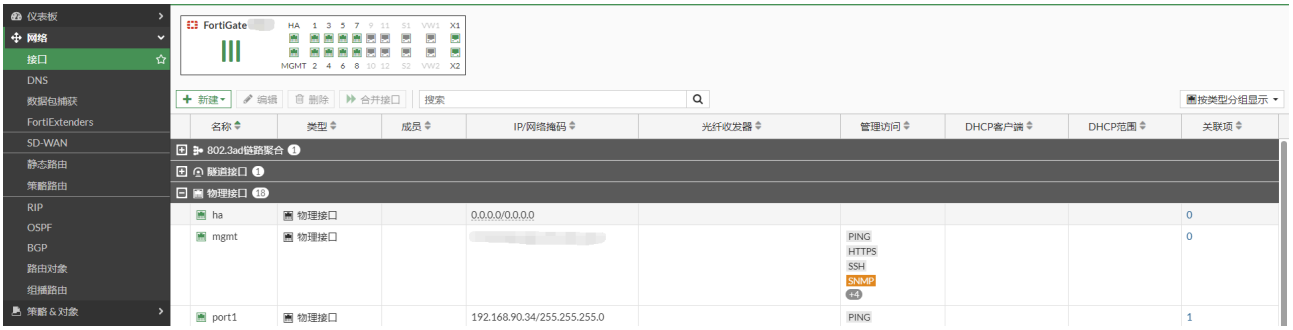
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.90.34	192.168.91.209	RADIUS	122	Access-Request id=9
2	0.000000000	192.168.91.209	192.168.90.34	RADIUS	74	Access-Accept id=9

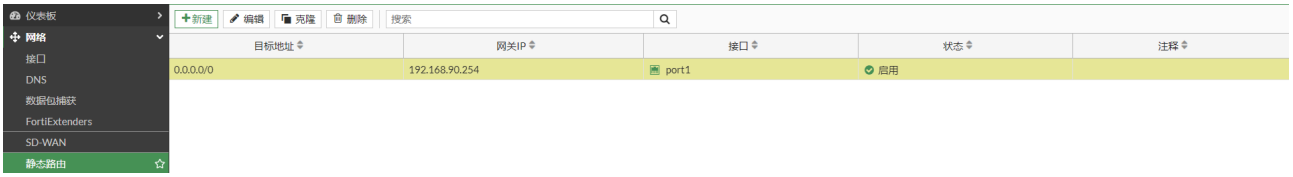
```
> Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface Fake IF, text2pcap, id 0
> Ethernet II, Src: Fortinet_72:3f:17 (08:5b:0e:72:3f:17), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> Internet Protocol Version 4, Src: 192.168.91.209, Dst: 192.168.90.34
> User Datagram Protocol, Src Port: 1812, Dst Port: 12269
v RADIUS Protocol
    Code: Access-Accept (2)
    Packet identifier: 0x9 (9)
    Length: 32
    Authenticator: 3b5c7adf49f224b267faf26472664e15
    [This is a response to a request in frame 1]
    [Time from request: 0.000000000 seconds]
  v Attribute Value Pairs
    v AVP: t=Vendor-Specific(26) l=12 vnd=Fortinet, Inc.(12356)
        Type: 26
        Length: 12
        Vendor ID: Fortinet, Inc. (12356)
      v VSA: t=Fortinet-Group-Name(1) l=6 val=grp1
          Type: 1
          Length: 6
          Fortinet-Group-Name: grp1
```

Radius 认证配置与排错

配置 Radius 认证

1.配置接口地址及路由





ping 测试 FortiGate 与 Radius Server 之间的连通性。测试环境使用 FortiAuthenticator 作为 Radius 服务器，地址是 192.168.91.209。

```
# execute ping 192.168.91.209
PING 192.168.91.209 (192.168.91.209): 56 data bytes
64 bytes from 192.168.91.209: icmp_seq=0 ttl=63 time=0.3 ms
64 bytes from 192.168.91.209: icmp_seq=1 ttl=63 time=0.2 ms
64 bytes from 192.168.91.209: icmp_seq=2 ttl=63 time=0.1 ms
64 bytes from 192.168.91.209: icmp_seq=3 ttl=63 time=0.1 ms
64 bytes from 192.168.91.209: icmp_seq=4 ttl=63 time=0.2 ms

--- 192.168.91.209 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms
```

2.配置 Radius 服务器

选择”用户与认证“-->” RADIUS“，点击”新建“。



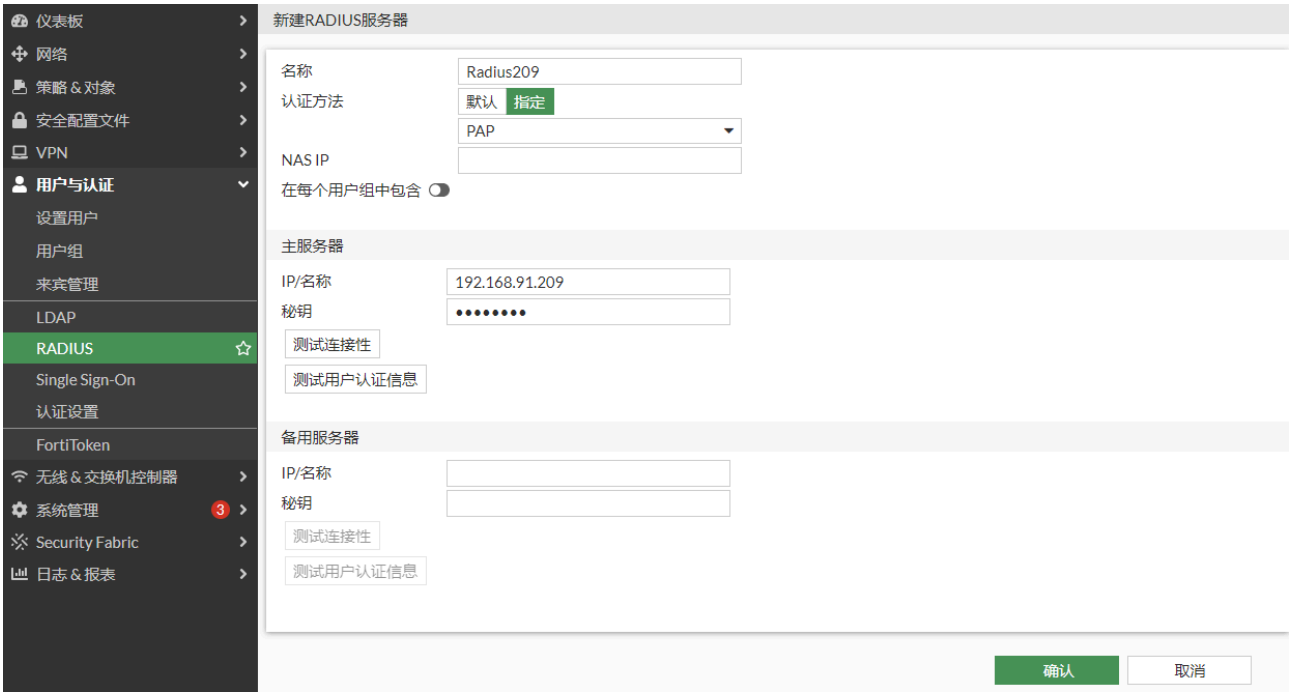
配置参数：

名称：输入一个名称，不冲突即可；

认证方法：FortiGate 支持 CHAP，MS - CHAP，MS - CHAP，PAP 认证方法。如果认证方法是默认，则 FortiGate 会使用 PAP、MSCHAP_v2 和 CHAP 三种方式向 Radius 服务器发送请求，Radius 服务器的日志中往往会报一些错误，因此 **请指定一种认证方法**，这里使用 PAP；

IP/名称：在主服务器输入 Radius 服务器的 IP 地址；

密钥：FortiGate 和 Radius 服务器之间的共享密钥，这里设置为 fortinet。



点击”确认“，配置完成。



对应的命令行如下：

```
config user radius
    edit "Radius209"
        set server "192.168.91.209"
        set secret ENC
fJ6Xhg0wVEZyp8qAJfiuNSJLqyNVrWsRwQ1zELqWc7GnyBso4IrSelJJR4DBKmXQ/ex9cFWq3nJfoWEbWjRy1nrwUMl5HJGPiHO+c8MXXy6IAUE38ec4Y/x5Vn
RPZpEgPPiVIuCXZfbpnjcR1t5RdO8lgTTRkhiNqBFvrZtyxsucpqRYSH7eyTN0ih5nJigvUn8oAg==
        set auth-type pap
    next
end
```

Radius 服务器测试

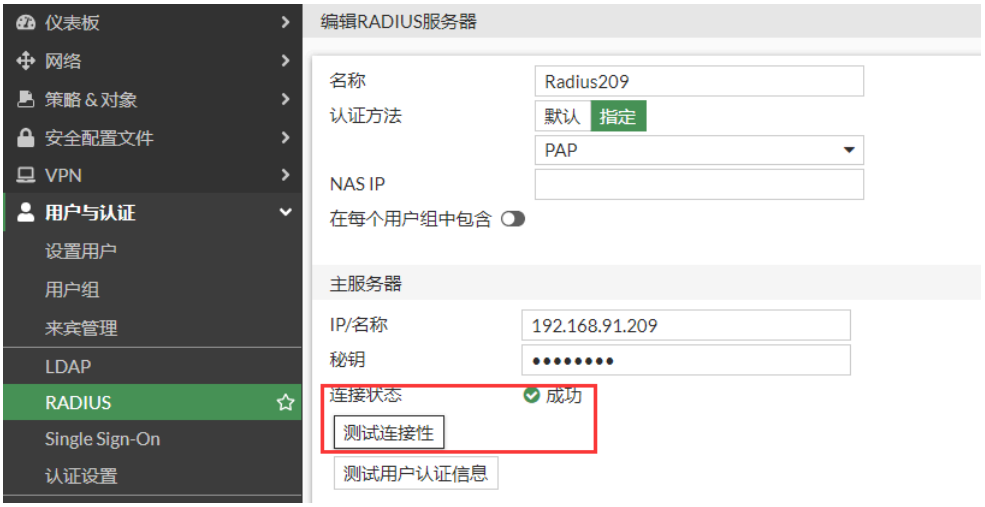
1.FortiGate 与 Radius 服务器之间连通性测试

GUI 编辑 Radius 服务器的配置界面，点击”测试连接性“。

CLI 的语法：

```
#diagnose test authserver radius-direct
<server_name or IP> <port number (0 default port)> <secret> <pap | chap | mschap | mschap2> <user> <password>
```

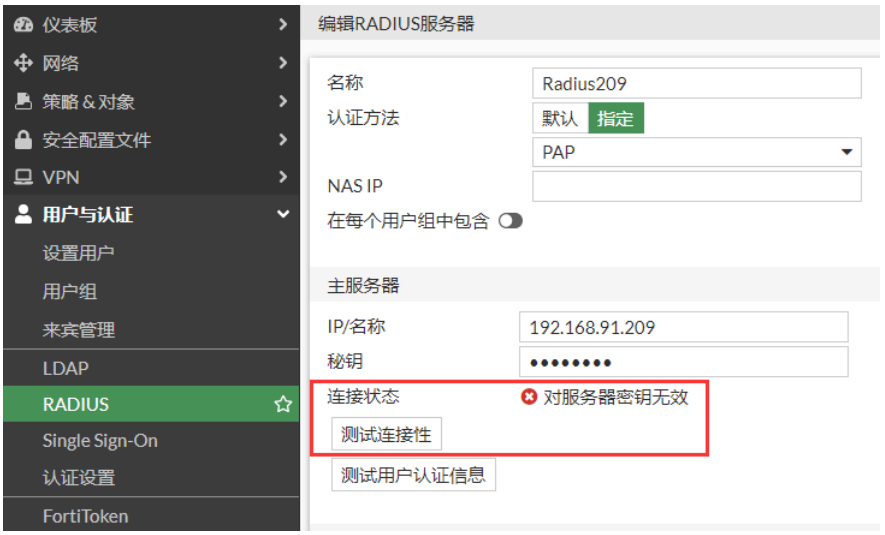
- 如果连接状态是”成功“，则 FortiGate 与 Radius 服务器之间通信正常。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is OK
Access-Accept
AVP: l=12 t=Vendor-Specific(26) v=Fortinet(12356)
  VSA: l=6 t=Fortinet-Group-Name(1)
  Value: 'grp1'
```

- 如果连接状态是“对服务器密钥无效”，则 FortiGate 与 Radius 服务器之间的密钥不一致。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is Secret invalid
```

- 如果连接状态是“不能连接 RADIUS 服务器”，则 Radius 服务器的地址或者端口不正确，或者 Radius 服务器没有授权 FortiGate 作为客户端。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius-direct 192.168.91.209 1812 fortinet pap user1 12345678
RADIUS server '192.168.91.209' status is Server unreachable
```

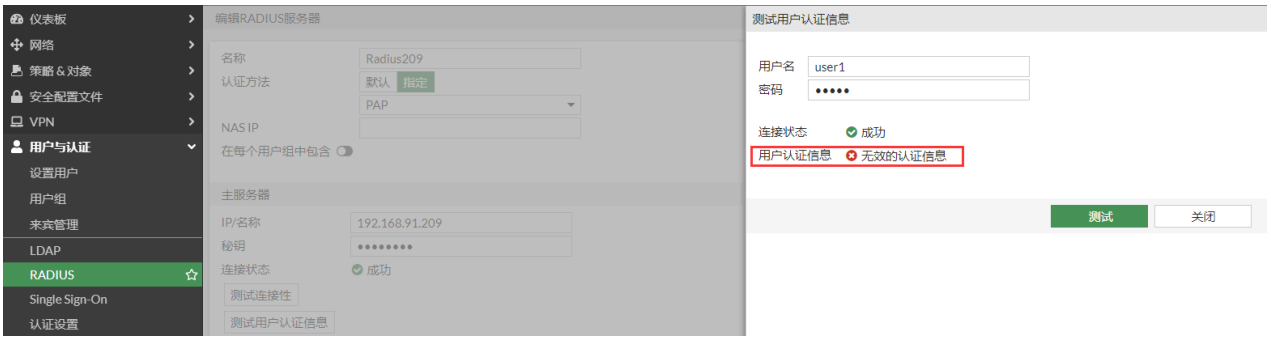
2.测试 Radius 服务器中的用户及密码

GUI 编辑 Radius 服务器的配置界面，点击”测试用户认证信息“。

CLI 的语法：

```
# diagnose test authserver radius
<server_name> <chap | pap | mschap | mschap2> <username> <password>
```

- 用户名或者密码不正确



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius Radius209 pap user1 test123
authenticate 'user1' against 'pap' failed, assigned_rad_session_id=730523922 session_timeout=0 secs idle_timeout=0 secs!
```

- 用户名和密码正确

可以看到 Radius 服务器返回的状态是成功，以及用户属于，如 user1 属于用户组 grp1。



对应的 CLI 测试结果如下：

```
# diagnose test authserver radius Radius209 pap user1 12345678
authenticate 'user1' against 'pap' succeeded, server=primary assigned_rad_session_id=730523923 session_timeout=0 secs
idle_timeout=0 secs!
Group membership(s) - grp1
```

其他 Radius 配置参数

1.修改 Radius 端口

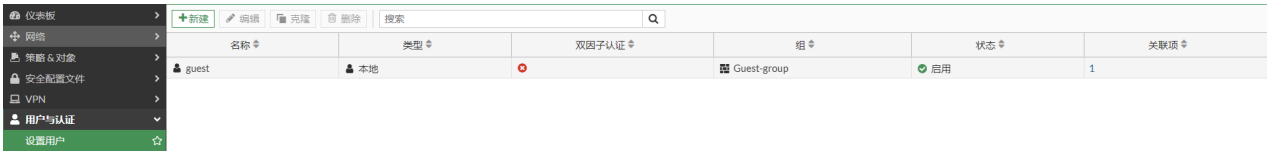
```
config user radius
    edit Radius209
        set radius-port xxx
    next
end
```

2.指定原 IP 地址

```
config user radius
    edit Radius209
        set source-ip x.x.x.x
    next
end
```

创建 radius 用户

1. 选择“用户与认证”-->“设置用户”，点击“新建”



2. 选择 "远程 RADIUS 用户"，点击“下一步”



3. 输入“用户名”和“RADIUS 服务器”，点击“下一步”



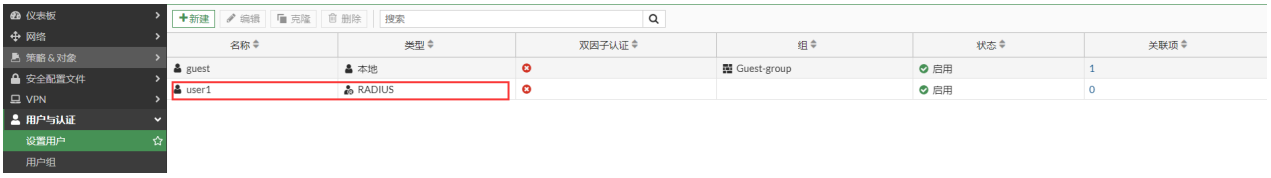
4. 如果 Radius 用户需要绑定 FortiGate 本地的 Fortitoken，则勾选“双因子认证”。这里不需要，直接点击“下一步”



5. 如果需要现在为 Radius 用户指定一个用户组，可以勾选“用户组”来指定，也可以等创建完用户后，在用户组菜单中添加。这里不勾选，直接点击“提交”



6. 创建完成



创建 radius 用户组

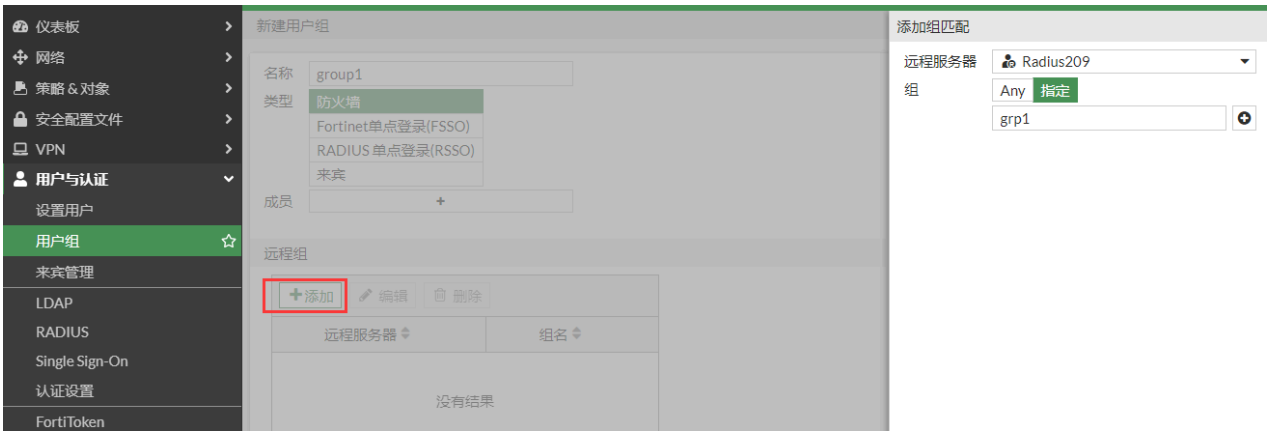
1. 选择“用户与认证”-->"用户组"，点击 "新建"



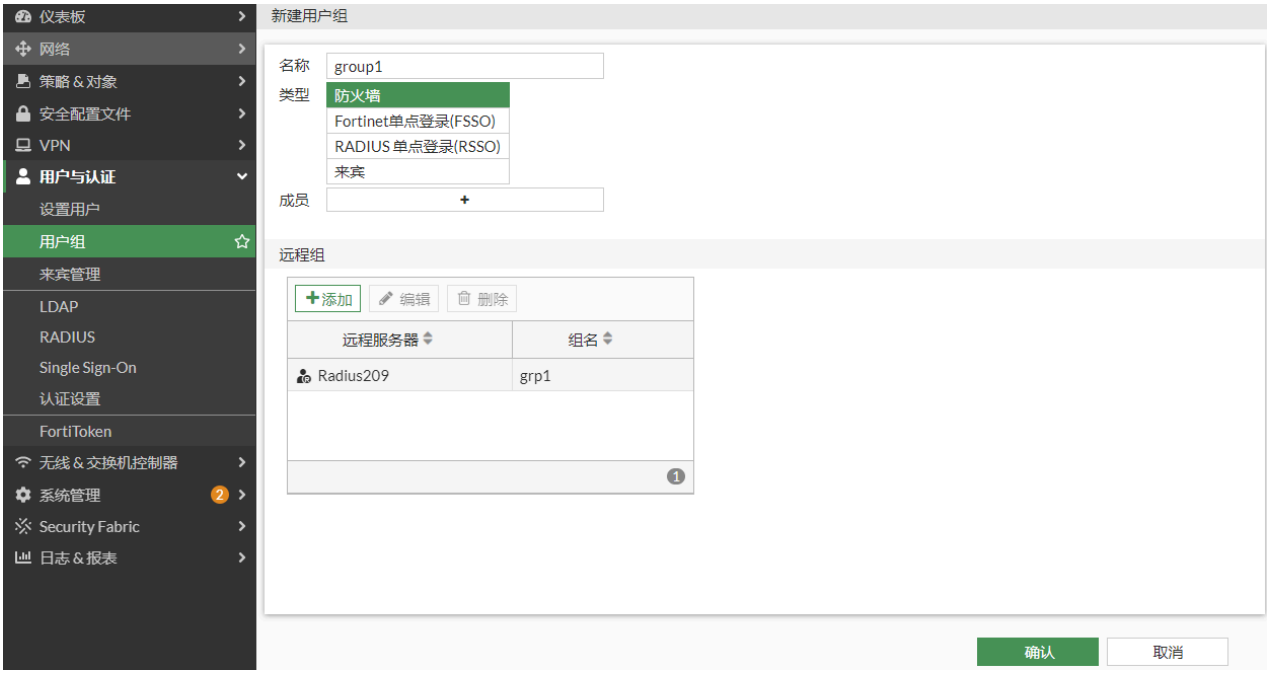
2. 配置参数

输入用户组名称 group1，在远程组下，点击“添加”，在”远程服务器“中选择已配置的 Radius 服务器。

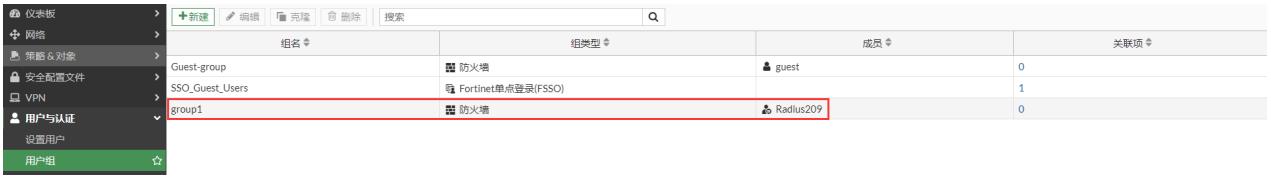
指定 Radius 的用户组，这里使用 grp1，表示 group1 匹配 Radius 用户组 grp1 下的用户，可以添加多个组；如果组选择“Any”，表示匹配 Radius 服务器下的所用用户。



选择完成后，点击“确认”。



3. 配置完成



Radius 认证 Debug

1.认证成功

```
[1906] handle_req-Rcvd auth req 730524052 for user1 in Radius209 opt=0000001d prot=0
[466] __compose_group_list_from_req-Group 'Radius209', type 1
[616] fnbamd_pop3_start-user1
[518] __fnbamd_cfg_get_radius_list_by_server-Loading RADIUS server 'Radius209'
[342] fnbamd_create_radius_socket-Opened radius socket 12
[342] fnbamd_create_radius_socket-Opened radius socket 13
[1394] fnbamd_radius_auth_send-Compose RADIUS request
[1351] fnbamd_rad_dns_cb-192.168.91.209→192.168.91.209

认证请求，radius服务器是192.168.91.209，用户名user1，认证方法是PAP
[1323] __fnbamd_rad_send-Sent radius req to server 'Radius209': fd=12, IP=192.168.91.209(192.168.91.209:1812) code=1 id=6
len=101 user="user1" using PAP
[319] radius_server_auth-Timer of rad 'Radius209' is added
[633] create_auth_session-Total 1 server(s) to try
[1360] fnbamd_auth_handle_radius_result-Timer of rad 'Radius209' is deleted

认证响应，响应code 2，认证成功，返回的用户属性
[1802] fnbamd_radius_auth_validate_pkt-RADIUS resp code 2
[320] extract_success_vsas-FORTINET attr, type 1, val grp1
[1385] fnbamd_auth_handle_radius_result-->Result for radius svr 'Radius209' 192.168.91.209(1) is 0
[266] find_matched_usr_grps-Skipped group matching
[216] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 730524052, len=2154
[789] destroy_auth_session-delete session 730524052
```

2.认证失败

```
[1906] handle_req-Rcvd auth req 730524053 for user1 in Radius209 opt=0000001d prot=0
[466] __compose_group_list_from_req-Group 'Radius209', type 1
[616] fnbamd_pop3_start-user1
[518] __fnbamd_cfg_get_radius_list_by_server-Loading RADIUS server 'Radius209'
[342] fnbamd_create_radius_socket-Opened radius socket 12
```

```
[342] fnbamd_create_radius_socket-Opened radius socket 13
[1394] fnbamd_radius_auth_send-Compose RADIUS request
[1351] fnbamd_rad_dns_cb-192.168.91.209→192.168.91.209

认证请求，radius服务器是192.168.91.209，用户名user1，认证方法是PAP
[1323] __fnbamd_rad_send-Sent radius req to server 'Radius209': fd=12, IP=192.168.91.209(192.168.91.209:1812) code=1 id=7
len=101 user="user1" using PAP
[319] radius_server_auth-Timer of rad 'Radius209' is added
[633] create_auth_session-Total 1 server(s) to try
[1360] fnbamd_auth_handle_radius_result-Timer of rad 'Radius209' is deleted

认证响应，响应code 3，认证拒绝
[1802] fnbamd_radius_auth_validate_pkt-RADIUS resp code 3
[1385] fnbamd_auth_handle_radius_result⟶Result for radius svr 'Radius209' 192.168.91.209(1) is 1
[216] fnbamd_comm_send_result-Sending result 1 (nid 0) for req 730524053, len=3172
[789] destroy_auth_session-delete session 730524053
```

FortiToken 认证

- FortiToken Mobile 在 FortiGate 上的激活
- FortiToken Mobile APP 软件下载与安装
- FortiToken Mobile 与用户进行绑定
- 免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法

FortiToken Mobile 在 FortiGate 上的激活

FortiToken Mobile 是 FortiToken 的手机软件安装版，是一种非常环保且方便的 FortiToken 实现方式，只需要在手机上安装一个 APP 即可实现 FortiToken 双因子的功能。默认每一台防火墙都有 2 个免费提供的 FortiToken Mobile 许可，自带且完全免费。

免费 token 激活

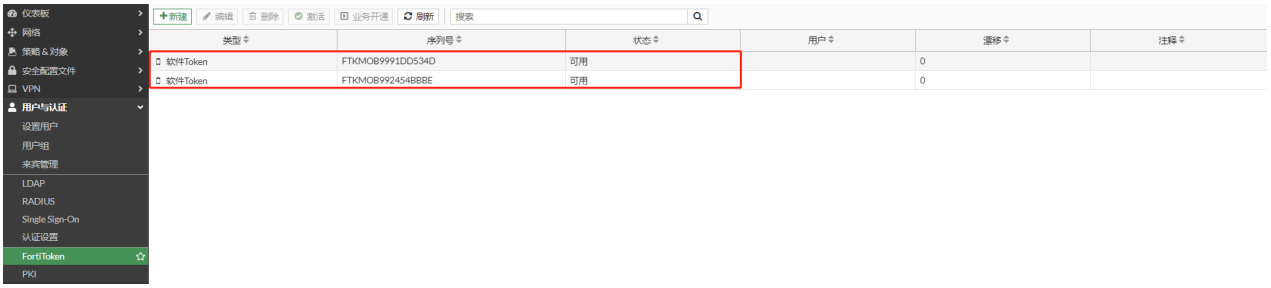
> **重要**
>
> 免费 token 常见问题请参考：免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法 ↗。

1. 保证 FortiGate 能够上网

2. 选择“用户与认证”-->” FortiToken“，点击“导入免费试用令牌”



即可生成两个免费的 token。



FortiToken 许可激活

1. 购买 FortiToken 后会收到一个 PDF 激活文件

FORTINET

FortiToken™ Mobile Redemption Certificate

25

Tokens

Activation Code	65CF-[redacted]-2C12
Certificate Valid Until	DEC 18, 2023
Serial Number	EFTM0[redacted]

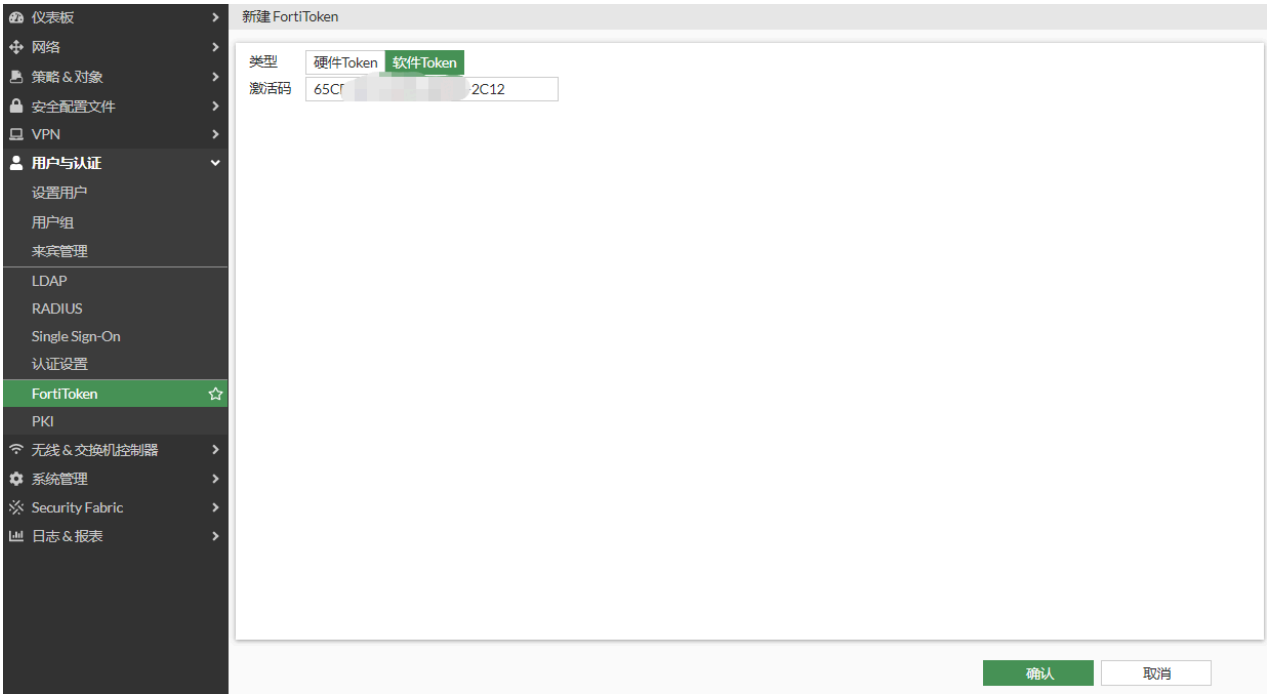
2. 保证 FortiGate 能够上网

3. 激活 FortiToken

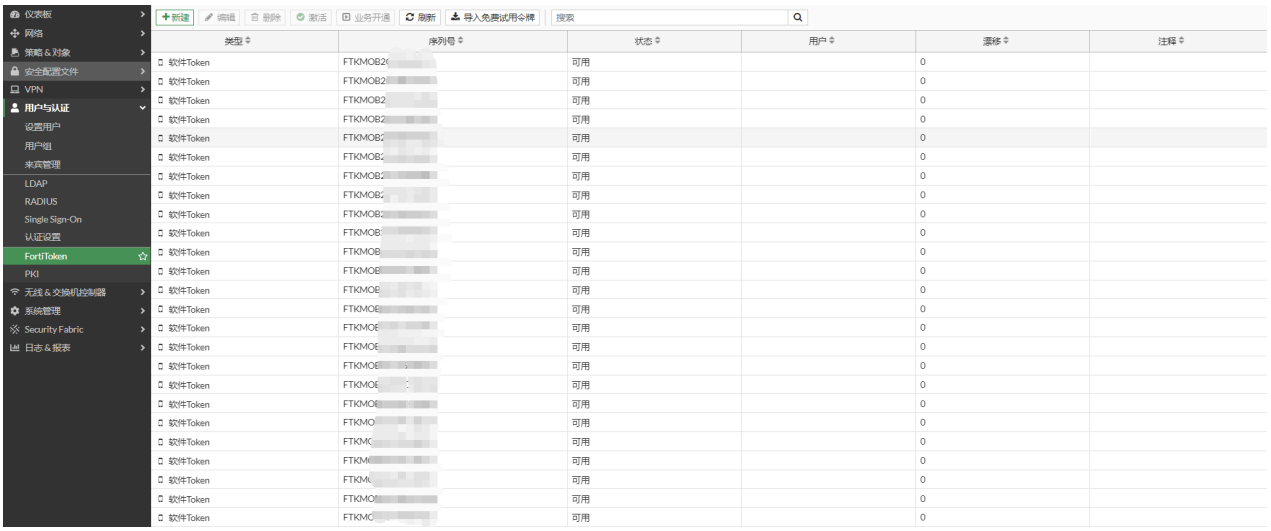
选择“用户与认证”-->” FortiToken“，点击“新建”



点击“软件 Token”，输入 PDF 文件的中的 Activation Code，点击确认。



FortiToken 导入成功。

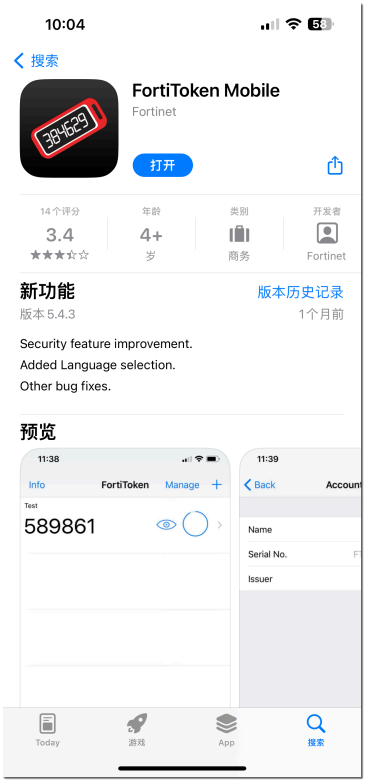


FortiToken Mobile APP 软件下载与安装

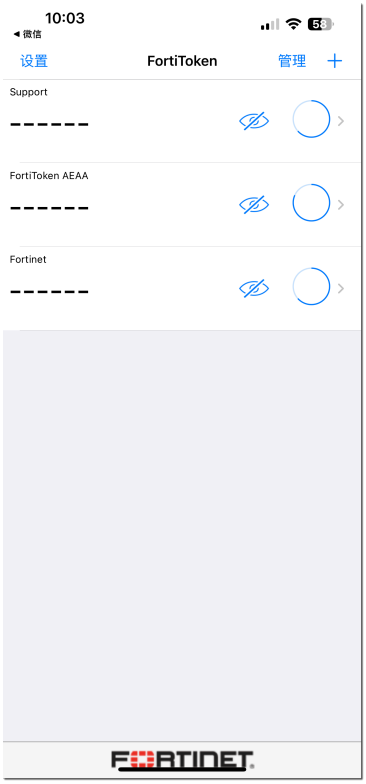
FortiToken Mobile APP 可以安装并运行在 iOS 系统、安卓系统手机上。

iPhone 安装

iPhone 直接在 Apple App Store 搜索“FortiToken Mobile”。



安装完后，显示如下（图中的包含了已添加过的两个 token）：



Android

安装包（v6.2.0-build0013）下载：FortiTokenAndroid.apk ↗。

也可在 Google Play 商店搜索“FortiToken Mobile”下载安装。

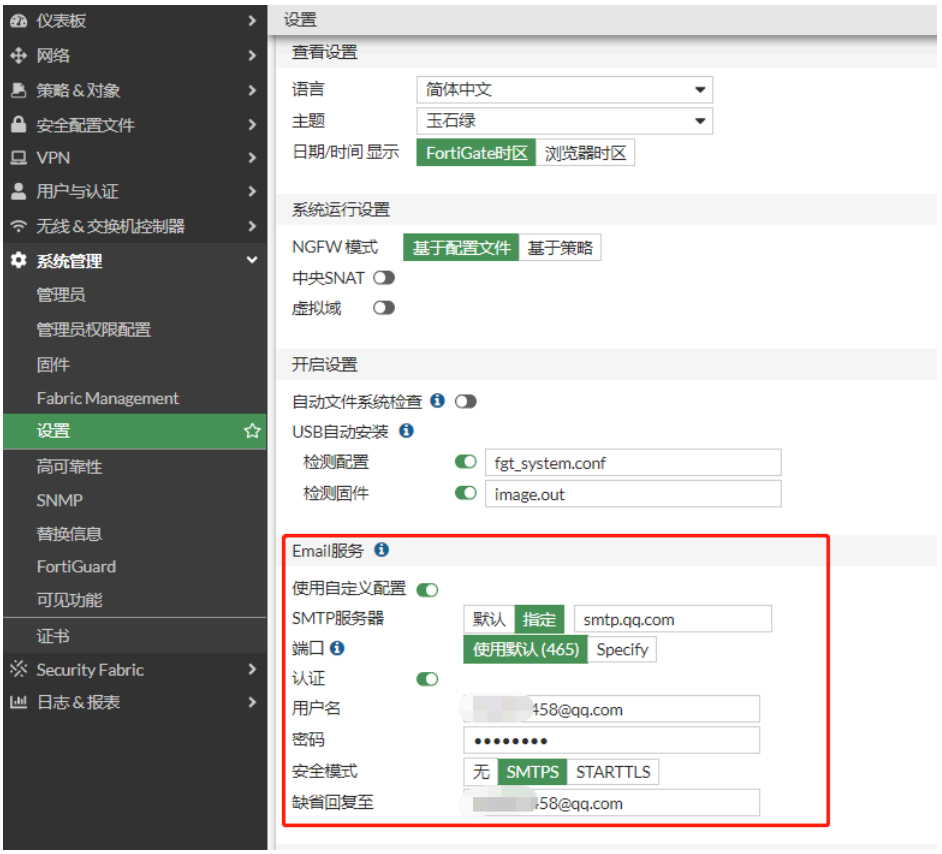


FortiToken Mobile 与用户进行绑定

FortiGate 使用邮件发送 FortiToken 的二维码图片，这样用户直接使用手机 APP 扫描二维码即可绑定 FortiToken Mobile。

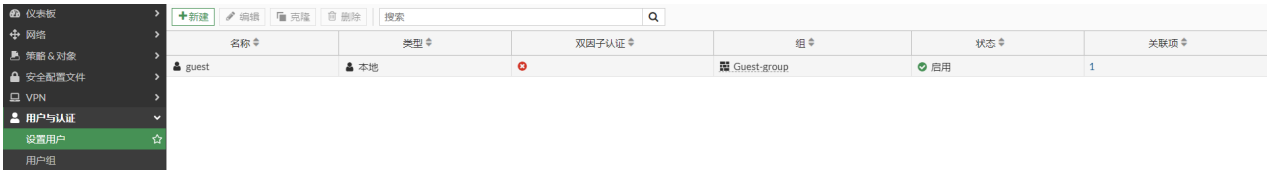
邮箱设置

这里使用 QQ 邮件发邮件。更多 Email 服务设置请参考系统管理 → 系统设置 → Email 服务 ↗章节。



绑定 FortiToken

1. 选择“用户与认证”-->“设置用户”，点击“新建”。



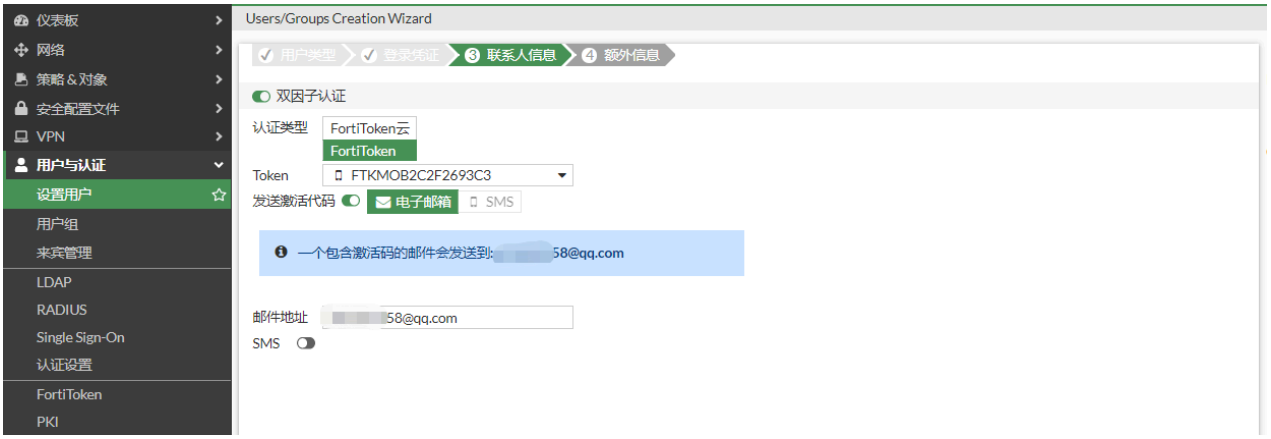
2. 选择“本地用户”，点击“下一步”。



3. 输入用户名和密码，点击“下一步”。



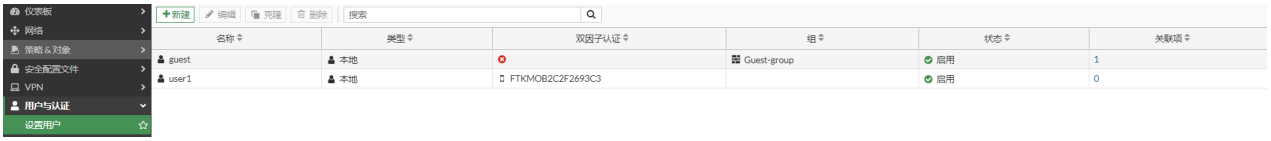
4. 启用双因子认证，选择一个 token 并填写用户邮箱，点击“下一步”。



5. 点击提交。

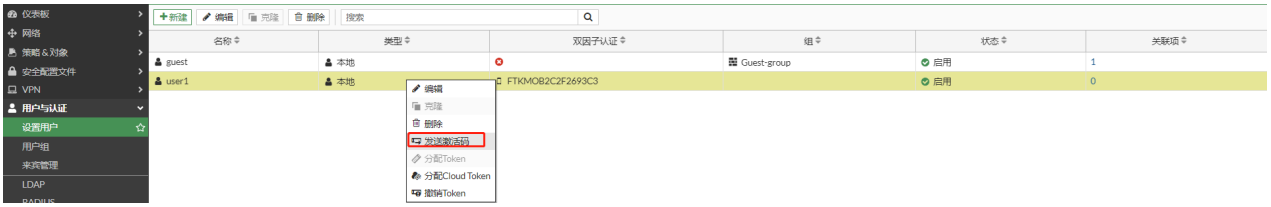


6. 完成用户创建。

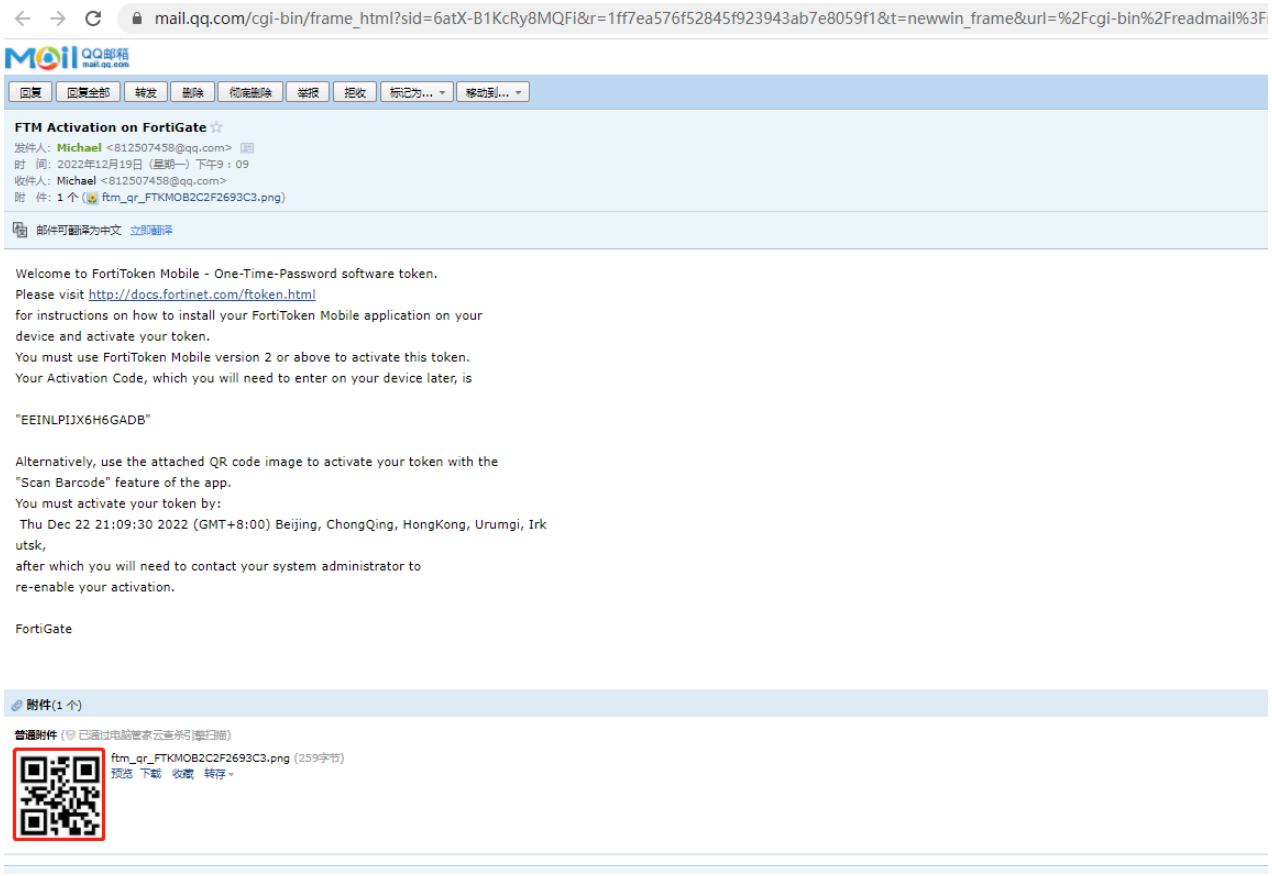


发送 token 到用户邮箱

1. 用户创建完成后会自动发送一封带有二维码的邮件给用户邮箱。如果没有收到，也可以右击该用户，选择“发送激活码”。

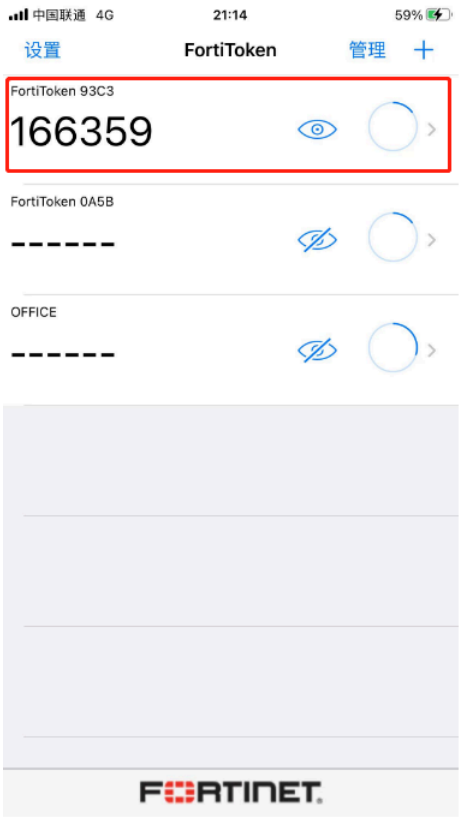


2. 邮件中的二维码如下：



FortiToken APP 扫描二维码

使用 FortiToken 扫描二维码即可拿到该 Token。



注意事项

1. 每个令牌只能在一台 FortiGate 设备上使用。
2. 每个令牌只能绑定一个用户。
3. 可以将令牌从用户上解除绑定，每次绑定用户均需要再次激活。

免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法

请参考链接：免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法 ↗。

FSSO 认证

- FSSO 认证
- FSSO Agent SSL 配置

FSSO 认证

重要

感谢东区 SE 提供的文档支持！

请参考链接：FortiGate FSSO 部署手册 ↗。

关于 FSSO 账户权限的更多信息，可参考：https://community.fortinet.com/t5/FortiGate/Technical-Tip-Restricting-a-Fortinet-Single-Sign-On-Agent/ta-p/198065 ↗

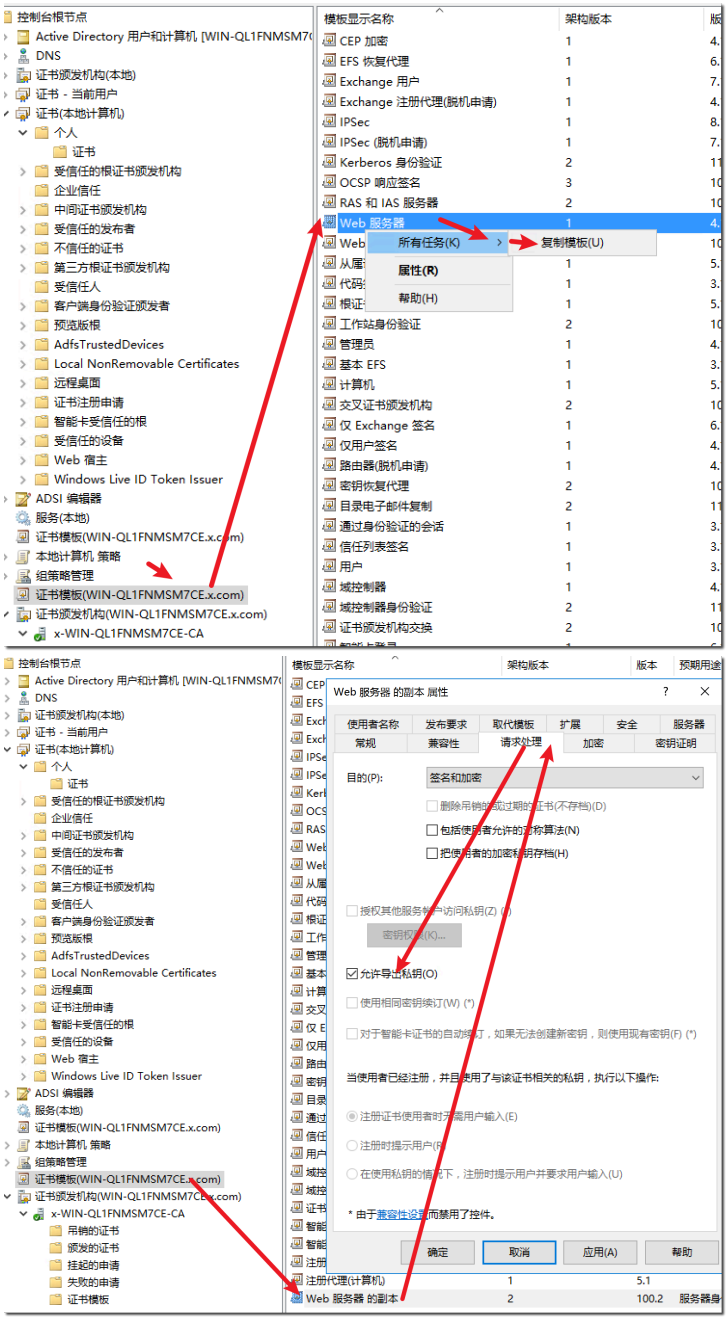
FSSO Agent SSL 配置

功能简介

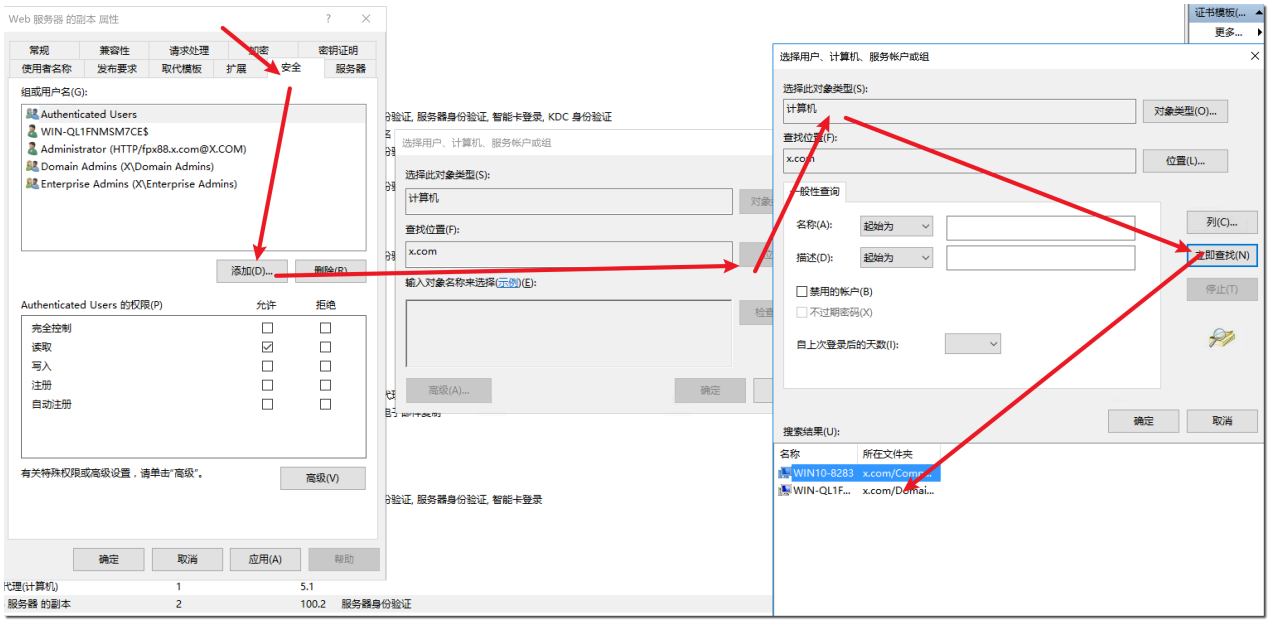
- FSSO Agent 与 FortiGate/FortiProxy 之间交互支持 SSL 加密。
- 其中 FSSO Agent 需要导入 `.crt` 和 `.key` 加上 phrase。
- FortiGate/FortiProxy 则需要导入根 CA 证书。
- openssl.exe 工具 ↗ 可以将常见的 Web 服务器证书 pfx 文件转换成这些文件。

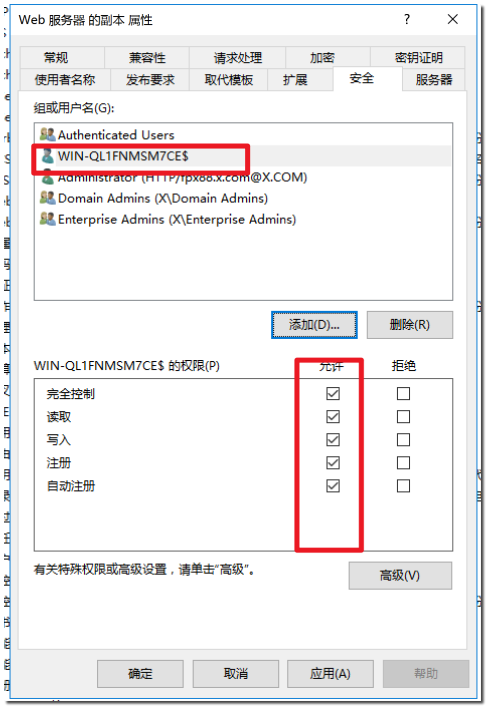
配置步骤

1. 克隆并发布 Web 证书模板（允许私钥导出）。

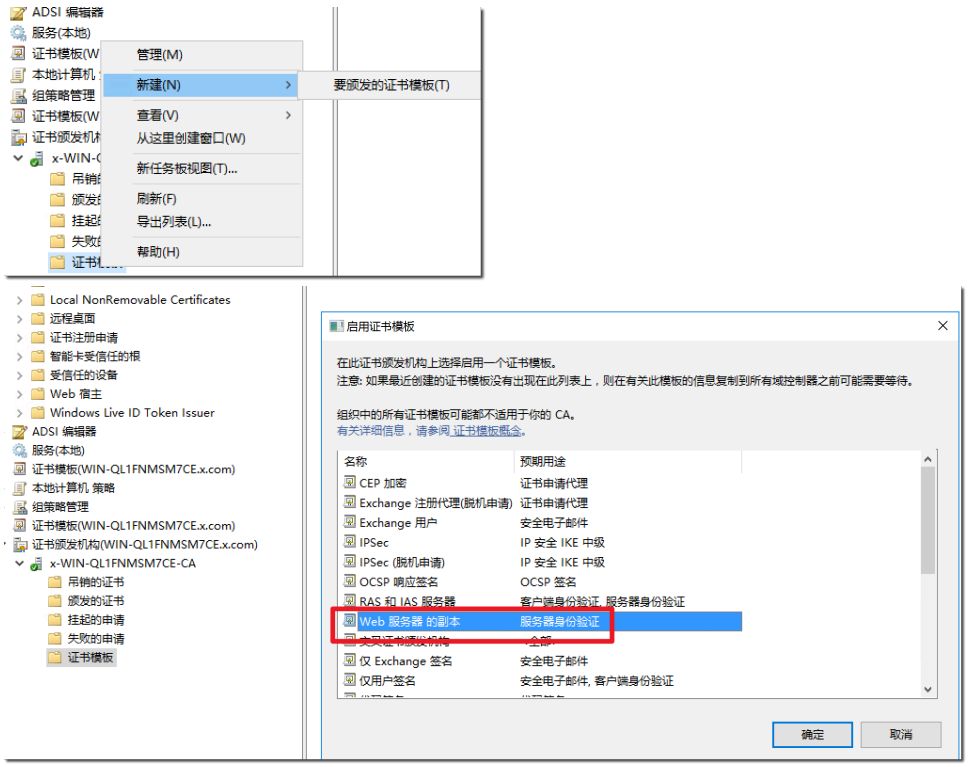


2. 对新模板授予 DC 完全权限。

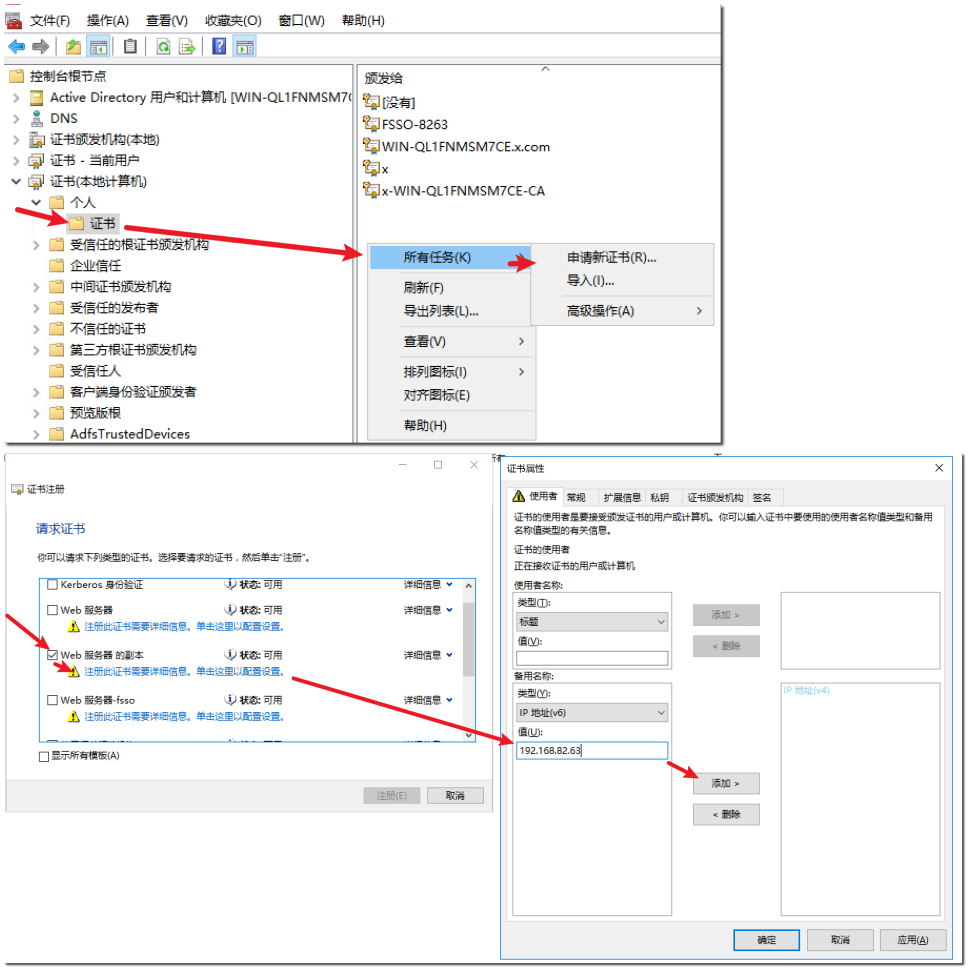

Web 服务器 的副本 属性
常规
兼容性
请求处理
加密
密钥证明
使用者名称
发布要求
取代模板
扩展
安全
服务器
组或用户名(G):
Authenticated Users
WIN-QL1FNMSM7CE$
Administrator (HTTP/fpx88.x.com@X.COM)
Domain Admins (X\Domain Admins)
Enterprise Admins (X\Enterprise Admins)
添加(D)...
删除(R)
Authenticated Users 的权限(P)
允许
拒绝
完全控制
读取
写入
注册
自动注册
有关特殊权限或高级设置，请单击"高级"。
高级(V)
确定
取消
应用(A)
帮助
选择用户、计算机、服务帐户或组
选择此对象类型(S):
计算机
对象类型(O)...
查找位置(F):
x.com
位置(L)...
一般性查询
名称(A):
起始为
描述(D):
禁用的帐户(B)
不过期密码(X)
自上次登录后的天数(I):
列(C)...
立即查找(N)
停止(T)
搜索结果(U):
名称
所在文件夹
WIN10-8283
WIN-QL1F...
证书模板(...
更多...

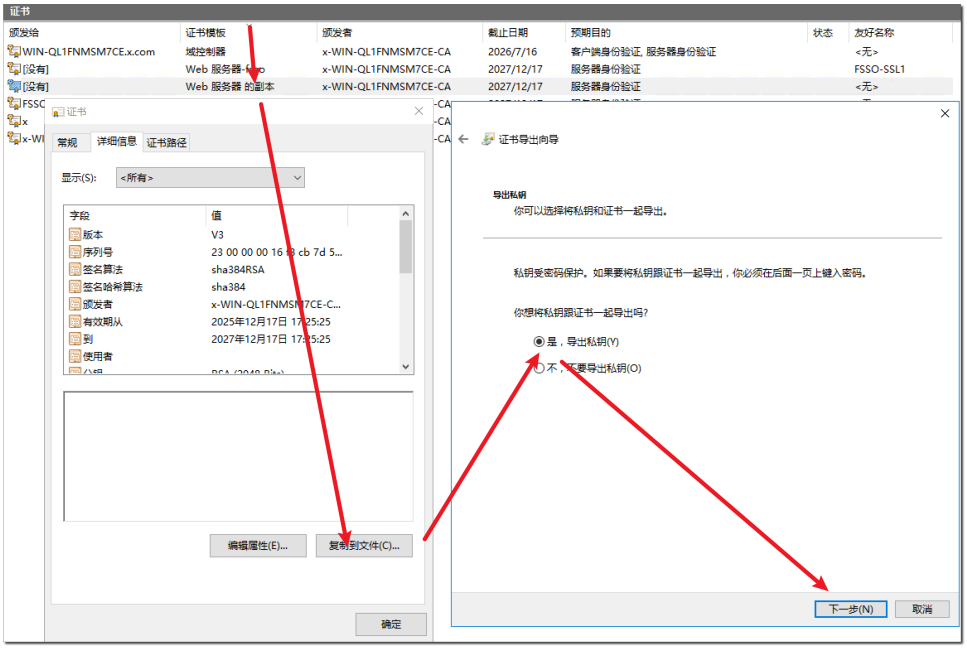

Web 服务器 的副本 属性
常规
兼容性
请求处理
加密
密钥证明
使用者名称
发布要求
取代模板
扩展
安全
服务器
组或用户名(G):
Authenticated Users
WIN-QL1FNMSM7CE$
Administrator (HTTP/fpx88.x.com@X.COM)
Domain Admins (X\Domain Admins)
Enterprise Admins (X\Enterprise Admins)
添加(D)...
删除(R)
WIN-QL1FNMSM7CE$ 的权限(P)
允许
拒绝
完全控制
读取
写入
注册
自动注册
有关特殊权限或高级设置，请单击"高级"。
高级(V)
确定
取消
应用(A)
帮助

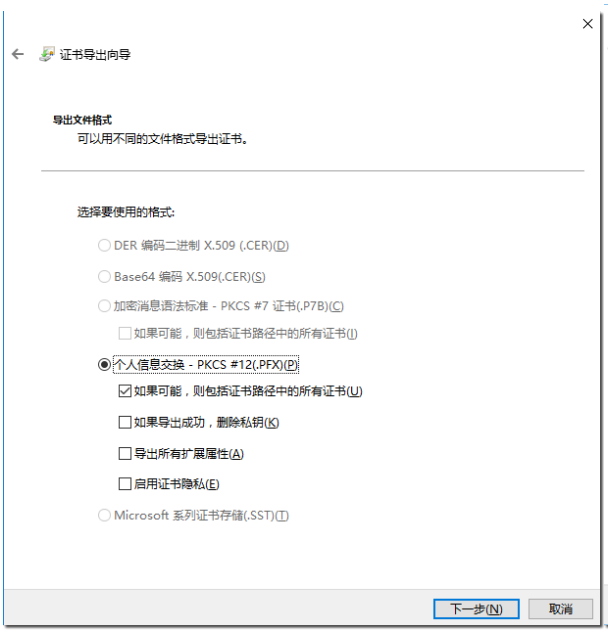
3. 颁发模板到 DC。

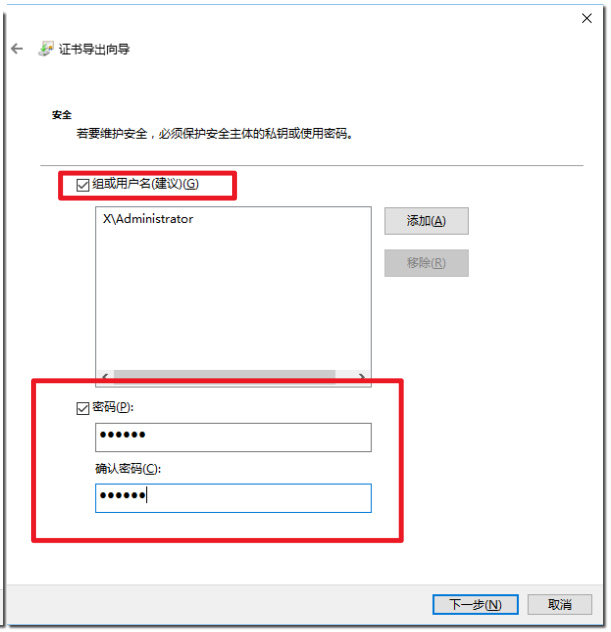


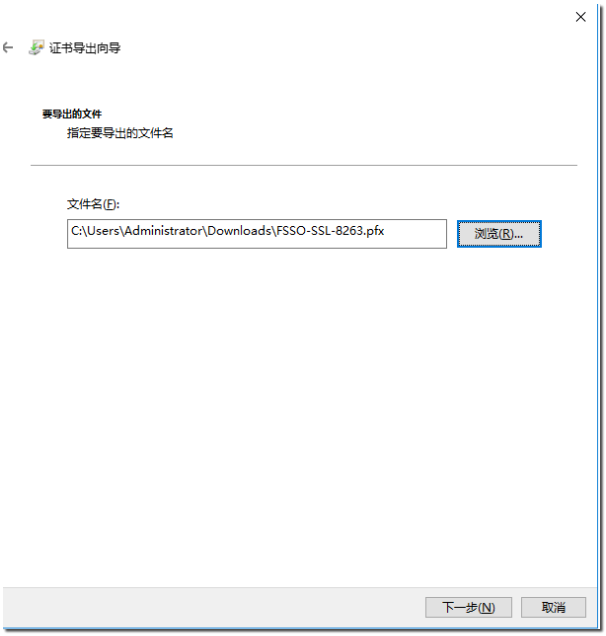
4. 申请并导出 pfx 证书文件。




证书
颁发给
证书模板
颁发者
截止日期
预期目的
状态
友好名称
WIN-QL1FNMSM7CE.x.com
域控制器
x-WIN-QL1FNMSM7CE-CA
2026/7/16
客户端身份验证, 服务器身份验证
<无>
[没有]
Web 服务器
x-WIN-QL1FNMSM7CE-CA
2027/12/17
服务器身份验证
FSSO-SSL1
[没有]
Web 服务器 的副本
x-WIN-QL1FNMSM7CE-CA
2027/12/17
服务器身份验证
<无>
证书
常规
详细信息
证书路径
显示(S):
<所有>
字段
值
版本
V3
序列号
签名算法
sha384RSA
签名哈希算法
sha384
颁发者
有效期从
2025年12月17日
到
2027年12月17日
使用者
编辑属性(E)...
复制到文件(C)...
确定
证书导出向导
导出私钥
你可以选择将私钥和证书一起导出。
私钥受密码保护。如果要将私钥跟证书一起导出，你必须在后面一页上键入密码。
你想将私钥跟证书一起导出吗?
是，导出私钥(Y)
不，不要导出私钥(O)
下一步(N)
取消


证书导出向导
导出文件格式
可以用不同的文件格式导出证书。
选择要使用的格式:
DER 编码二进制 X.509 (.CER)(D)
Base64 编码 X.509(.CER)(S)
加密消息语法标准 - PKCS #7 证书(.P7B)(C)
如果可能，则包括证书路径中的所有证书(I)
个人信息交换 - PKCS #12(.PFX)(P)
如果可能，则包括证书路径中的所有证书(U)
如果导出成功，删除私钥(K)
导出所有扩展属性(A)
启用证书隐私(E)
Microsoft 系列证书存储(.SST)(T)
下一步(N)
取消


证书导出向导
安全
若要维护安全，必须保护安全主体的私钥或使用密码。
组或用户名(建议)(G)
X\Administrator
添加(A)
移除(R)
密码(P):
确认密码(C):
下一步(N)
取消

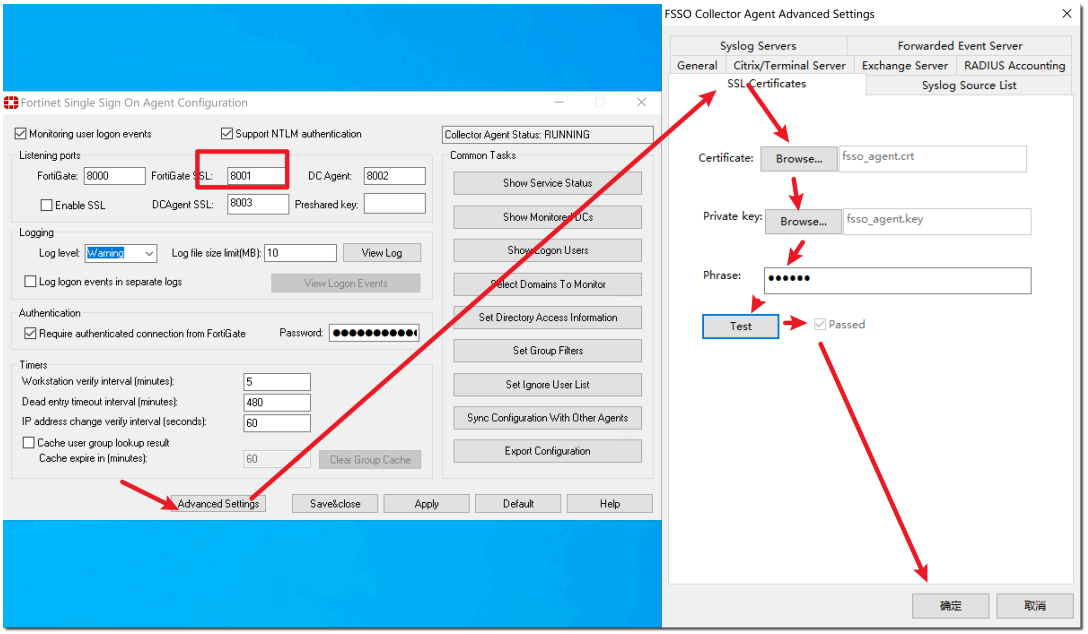

证书导出向导
要导出的文件
指定要导出的文件名
文件名(F):
C:\Users\Administrator\Downloads\FSSO-SSL-8263.pfx
浏览(R)...
下一步(N)
取消

5. pfx 文件转换：使用 **openssl.exe 工具** ↗ 从 `.pfx` 文件导出三个文件：

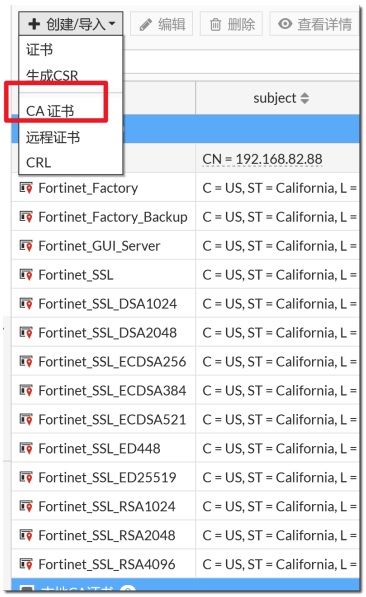
```bash
C:\Users\Administrator.X\Downloads\CA2>dir
 驱动器 C 中的卷没有标签。
 卷的序列号是 7053-3A8D

 C:\Users\Administrator.X\Downloads\CA2 的目录

2025/12/17  16:56    <DIR>          .
2025/12/17  16:56    <DIR>          ..
2025/12/17  16:55             5,329 FSSO-SSL-8263.pfx
               1 个文件          5,329 字节
               2 个目录 94,788,481,024 可用字节

C:\Users\Administrator.X\Downloads\CA2>C:\"Program Files"\OpenSSL-Win64\bin\openssl.exe  pkcs12 -in FSSO-SSL-8263.pfx -
out fsso_agent.crt
Enter Import Password:             \\MMC导出时的密码
Enter PEM pass phrase:             \\crt附带phrase
Verifying - Enter PEM pass phrase:  \\二次确认phrase
Bag Type: 1.3.6.1.4.1.311.17.4
Bag Value:
C:\Users\Administrator.X\Downloads\CA2>C:\"Program Files"\OpenSSL-Win64\bin\openssl.exe  pkcs12 -in FSSO-SSL-8263.pfx -
nocerts -out fsso_agent.key
Enter Import Password:             \\MMC导出时的密码
Enter PEM pass phrase:             \\crt附带phrase
Verifying - Enter PEM pass phrase:  \\二次确认phrase
Bag Type: 1.3.6.1.4.1.311.17.4
Bag Value:
C:\Users\Administrator.X\Downloads\CA2>C:\"Program Files"\OpenSSL-Win64\bin\openssl.exe  pkcs12 -in FSSO-SSL-8263.pfx  -
cacerts -nokeys -out root_ca.pem
Enter Import Password:
Bag Type: 1.3.6.1.4.1.311.17.4
Bag Value:
C:\Users\Administrator.X\Downloads\CA2>
C:\Users\Administrator.X\Downloads\CA2>
C:\Users\Administrator.X\Downloads\CA2>
C:\Users\Administrator.X\Downloads\CA2>dir
 驱动器 C 中的卷没有标签。
 卷的序列号是 7053-3A8D

 C:\Users\Administrator.X\Downloads\CA2 的目录

2025/12/17  16:58    <DIR>          .
2025/12/17  16:58    <DIR>          ..
2025/12/17  16:55             5,329 FSSO-SSL-8263.pfx
2025/12/17  16:58             7,843 fsso_agent.crt
2025/12/17  16:58             3,341 fsso_agent.key
2025/12/17  16:58             3,234 root_ca.pem
               4 个文件         19,747 字节
               2 个目录 94,788,460,544 可用字节
```

- `.crt` 文件和 `.key` 文件还有 phrase，用于导入 FSSO Agent。
- `.pem` 文件用于导入到 FortiGate/FortiProxy。

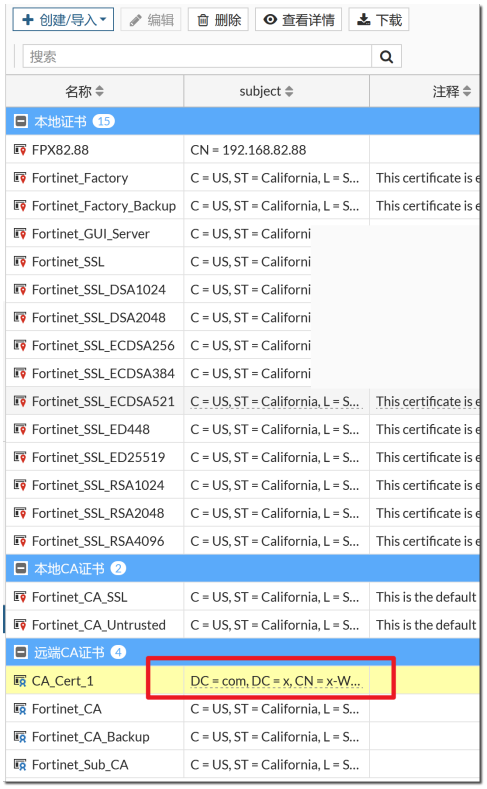
6. 导入 `.crt` 与 `.key` 文件到 FSSO Agent。



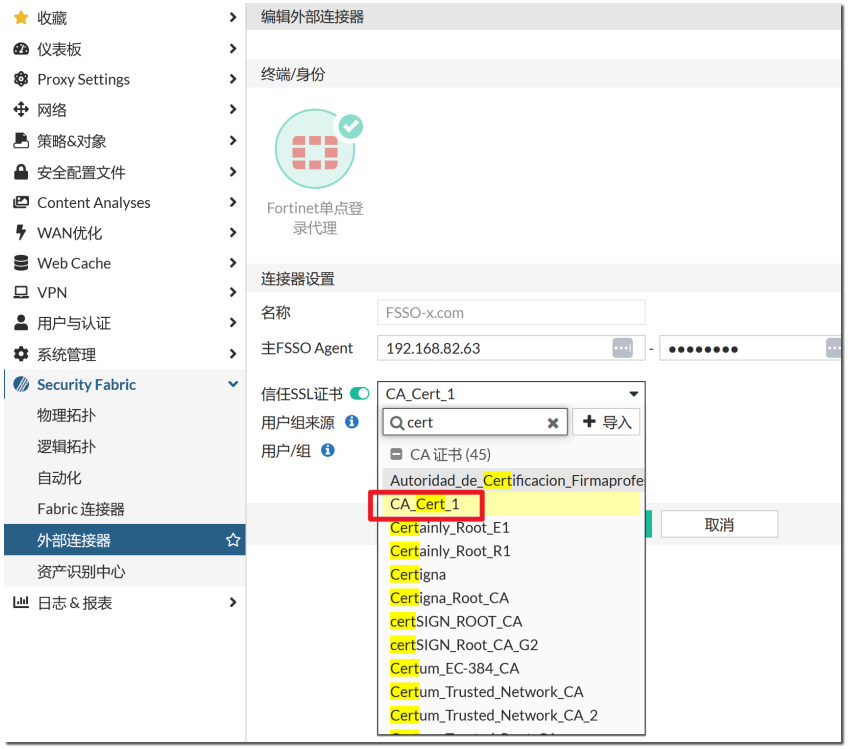
7. 导入 CA 到 FortiGate/FortiProxy。





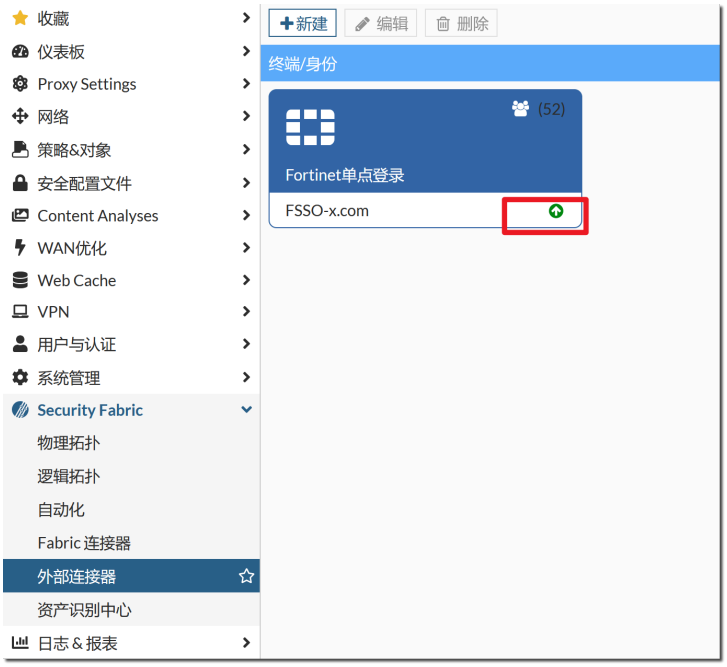


8. 在 FSSO 引用 CA 证书。



结果验证

1. 验证 FortiGate/FortiProxy 与 FSSO Agent 的 SSL 连接状态。



2. CLI 连接状态正常。

```bash
FortiGate # diagnose debug authd fsso server-status

Server Name              Connection Status     Version              Address
FSSO-x.com           connected             FSSO 5.0.0323        192.168.82.63

FPX-8288-7411 # show user fsso
config user fsso
    edit "FSSO-x.com"
        set server "192.168.82.63"
        set port 8001
        set password ENC
25fkg1YoX8sZs8euhJ+pSF3pb4jc1D1+GvQx1s9EC4af0udDi1/dgbieAGFZsc5ip/20H8uFAXGxbB07y8AV9jYXAinhMSM/+IsVXLFc2y/qmDtZBw6eSdpe
v4568bRfPhYsCB2IOi39dBSkzuvfrCuER/C9ueyPUpH92N1L0PHCs2MU+yyd8zSVqZlhrsfOeVhanQ==
        set port2 8001
        set port3 8001
        set port4 8001
        set port5 8001
        set ssl enable
        set ssl-trusted-cert "CA_Cert_1"
    next
end
```

3. Sniffer 抓包显示 SSL 端口 8001 交互正常。

```bash
747.634134 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: syn 3164597000
7747.634580 port1 in 192.168.82.63.8001 → 192.168.82.88.43096: syn 1037874556 ack 3164597001
7747.634625 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: ack 1037874557
7747.785340 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: psh 3164597001 ack 1037874557
7747.800330 port1 in 192.168.82.63.8001 → 192.168.82.88.43096: 1037874557 ack 3164597275
7747.800333 port1 in 192.168.82.63.8001 → 192.168.82.88.43096: psh 1037876017 ack 3164597275
7747.800392 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: ack 1037876017
7747.800441 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: ack 1037876726
7748.098050 port1 out 192.168.82.88.43096 → 192.168.82.63.8001: psh 3164597275 ack 1037876726
......
```

注意事项

- FSSO 相关 Debug 命令：`diagnose debug application authd -1`。
- 如果证书架构使用了 中级 CA (Intermediate CA)，请确保导入的 `.pem` 文件中包含中级 CA 证书和根 CA 证书的完整链，否则 FortiGate/FortiProxy 无法建立信任。
- 确保 FSSO Agent 的 SSL 端口（默认为 8001）可达。

SAML 认证

- 防火墙用户与 SSL VPN SAML 认证（Windows ADFS）
- 显式代理 SAML 认证（Windows ADFS）

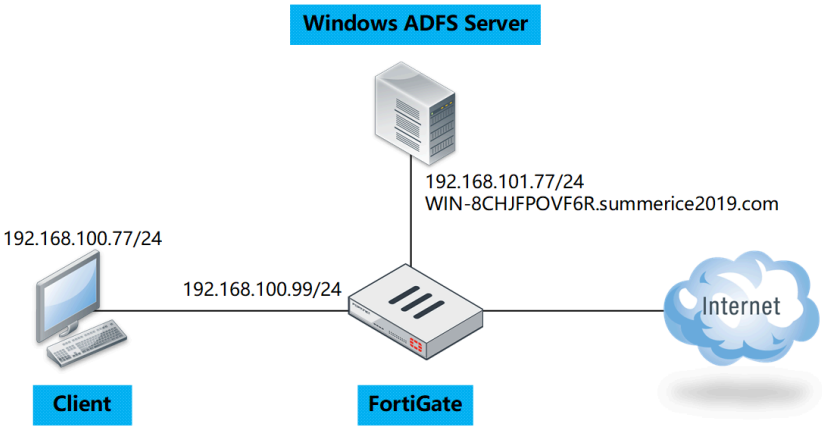
防火墙用户与 SSL VPN SAML 认证（Windows ADFS）

网络需求

ADFS（Active Directory Federation Service），Active Directory 联合身份验证服务是一项需要单独安装在 AD 服务器上的功能。其主要目的是为 AD 用户提供单一登录（SSO）访问权限。

ADFS 可以配合 FortiGate 实现 SAML 认证，本文详细介绍了 FortiGate 代理策略认证与 Windows Server 2019 ADFS 配合做 SAML 认证的配置，本例中 Fortigate 作为 SAML SP（Service Provider），ADFS 作为 SAML IDP（Identity Provider）。

网络拓扑



1. FortiGate 作为 SP，Windows Server 安装 ADFS 服务后作为 IDP。
2. FortiGate 和 Client 将 DNS 服务器地址配置为 Windows Server 的 IP。

配置步骤

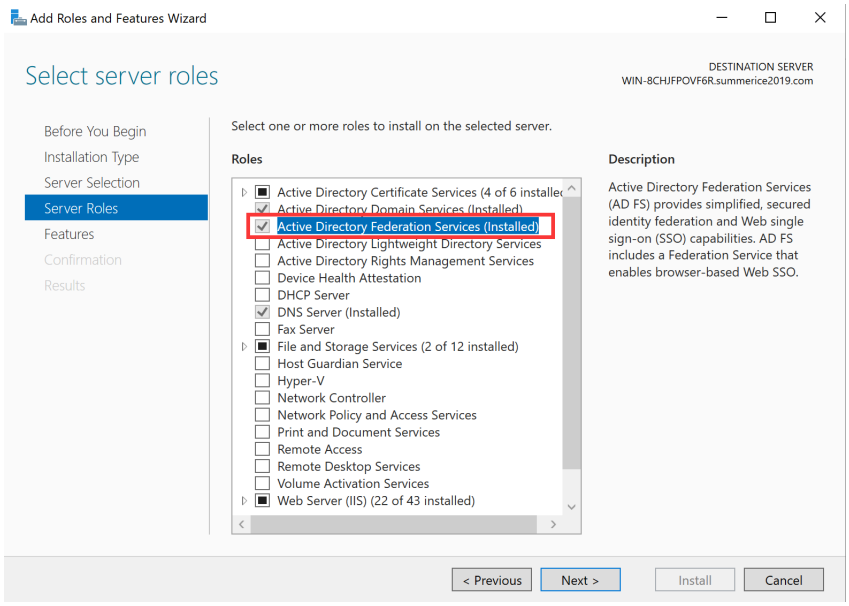
防火墙策略用户 SAML 认证

Windows Server 端配置

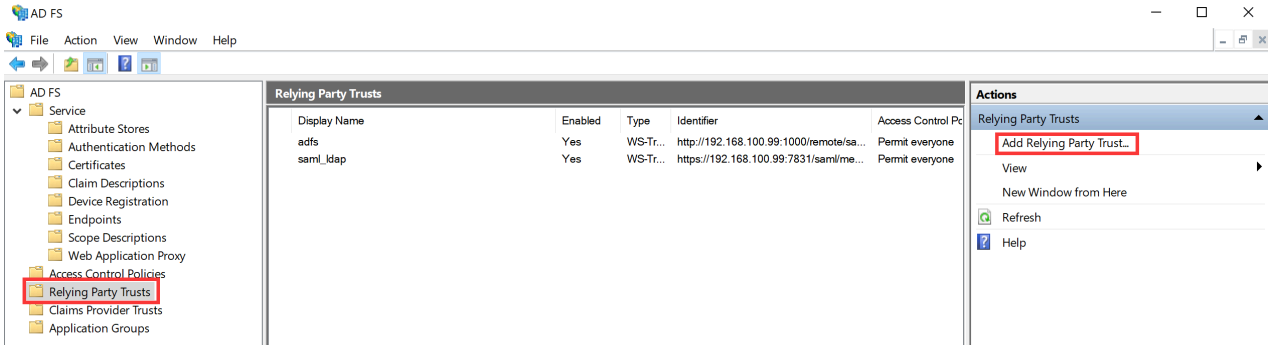
1. 在 Windows Server 中安装 AD 域控，加入或新建 AD 域。

2. 在服务器配置中，安装 Windows ADFS 服务。

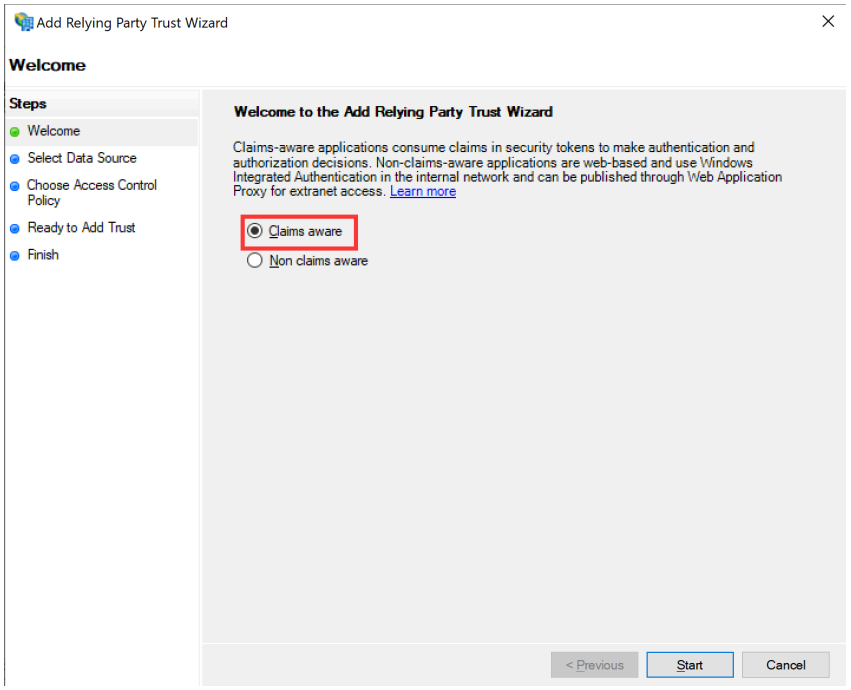
3. Client 为 AD 用户。



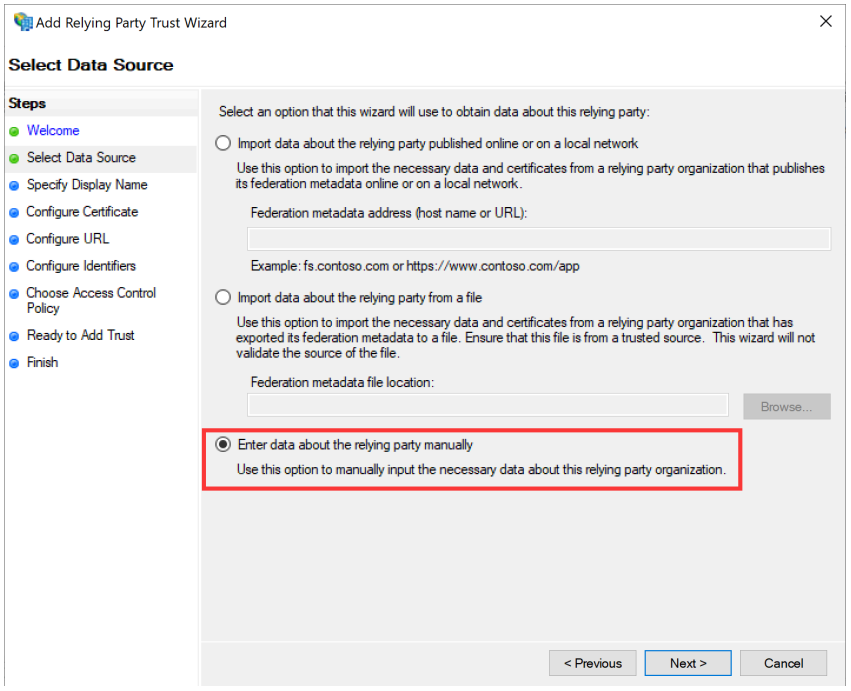
4. 安装完成后，通过 Server Manager -> Tools -> AD FS Management 打开 ADFS 管理器，并点击 Relying Party Trust，点击右侧的 Add Relying Party Trust 按钮添加信任方。



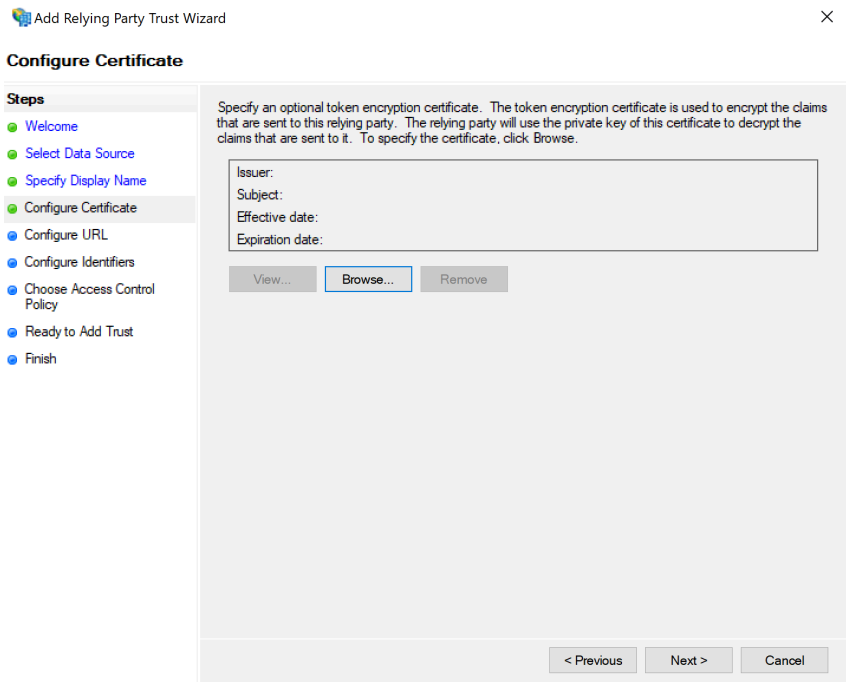
5. 弹出向导界面，这里选择感知方式，由于配合的是外部的 FortiGate，需要选择 Claims aware，点击 Start。



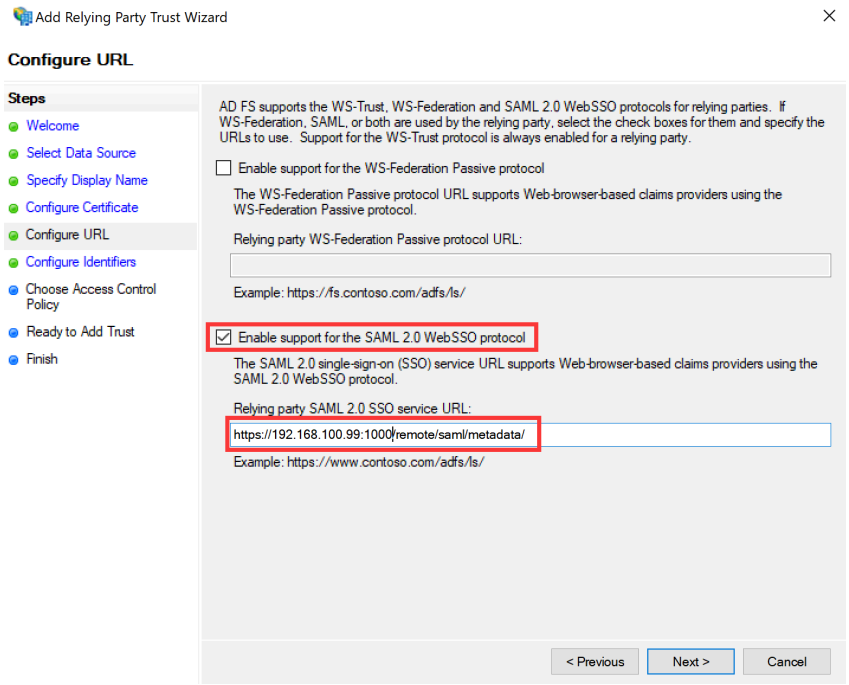
6. 数据源选择手动输入，点击 Next。



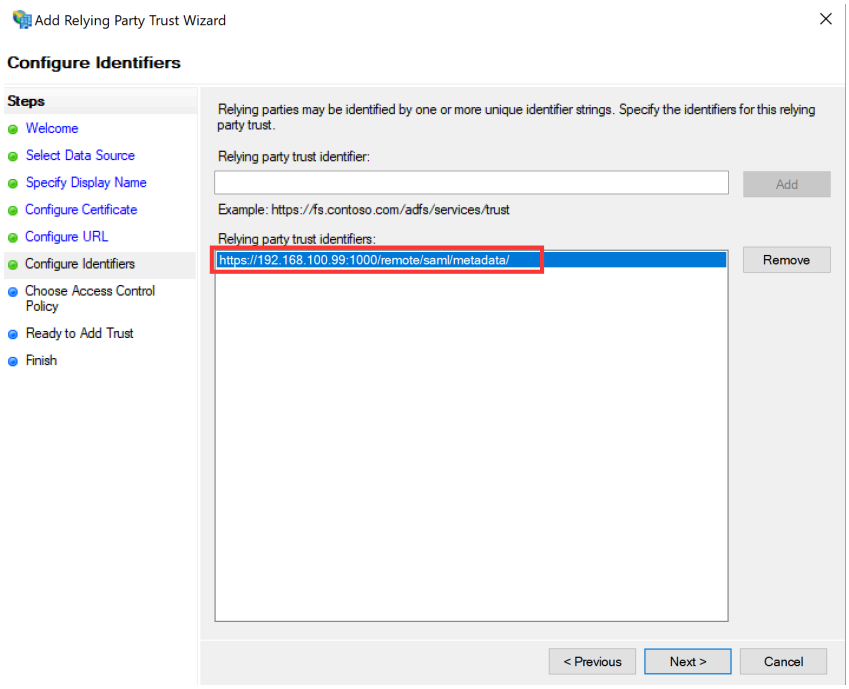
7. 直接点击 Next。



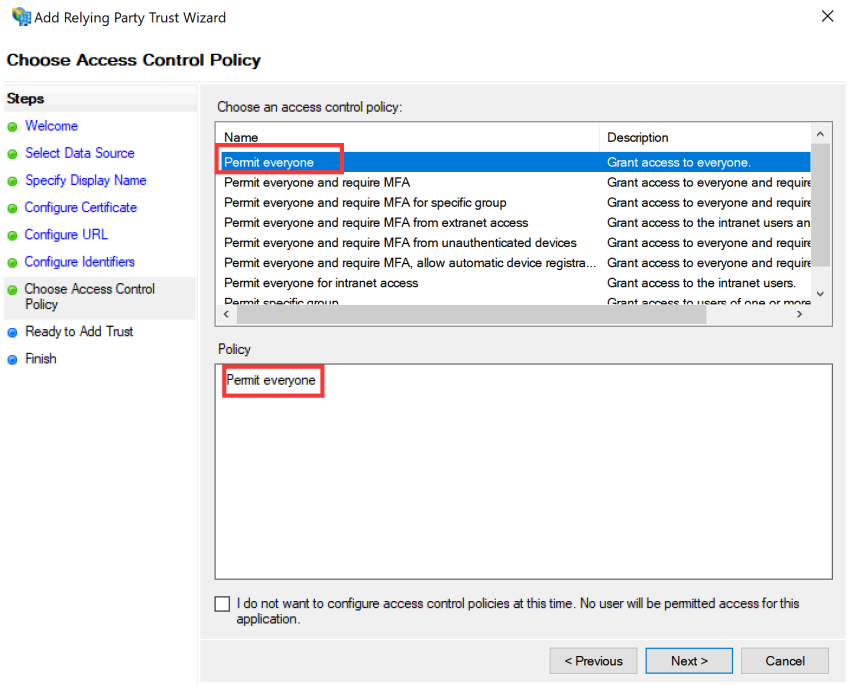
8. 选择 SAML 认证，并填写信赖方的 metadata 地址（FortiGate 内网口的认证地址和端口号）。



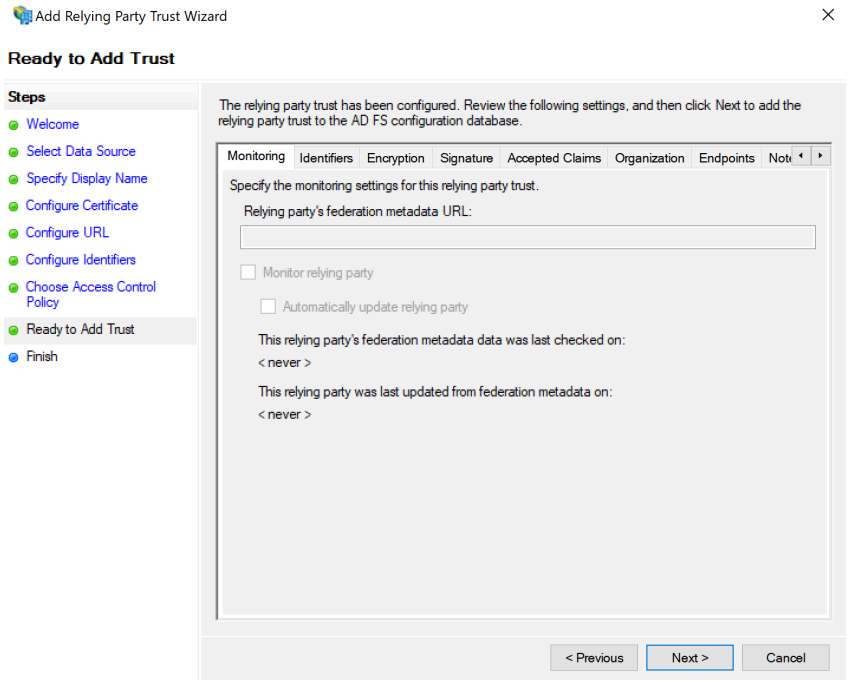
9. 配置信赖方 identifier URL，这里填写和上步相同的 URL，并点击 Add。



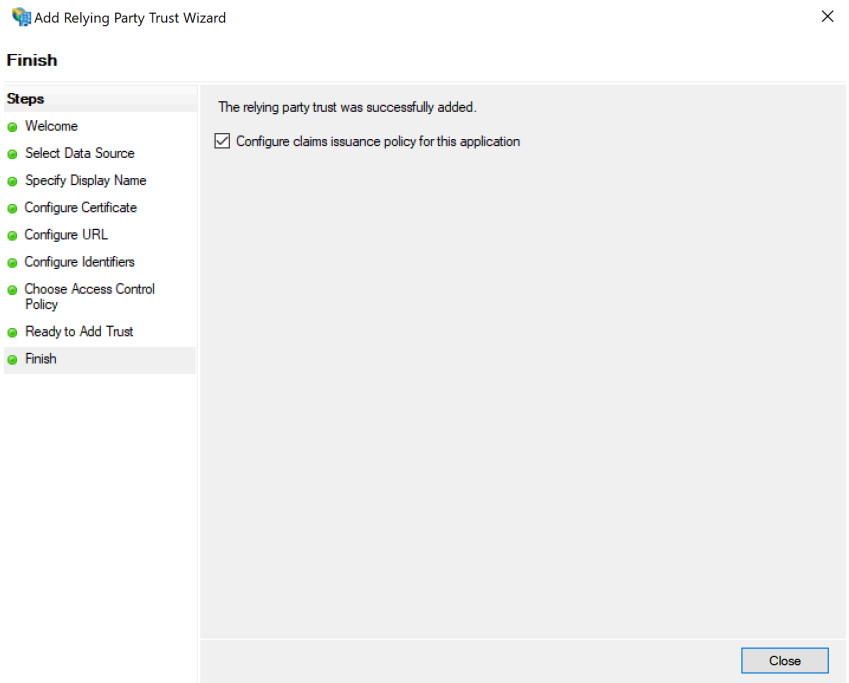
0. 访问控制策略选择允许所有人访问。



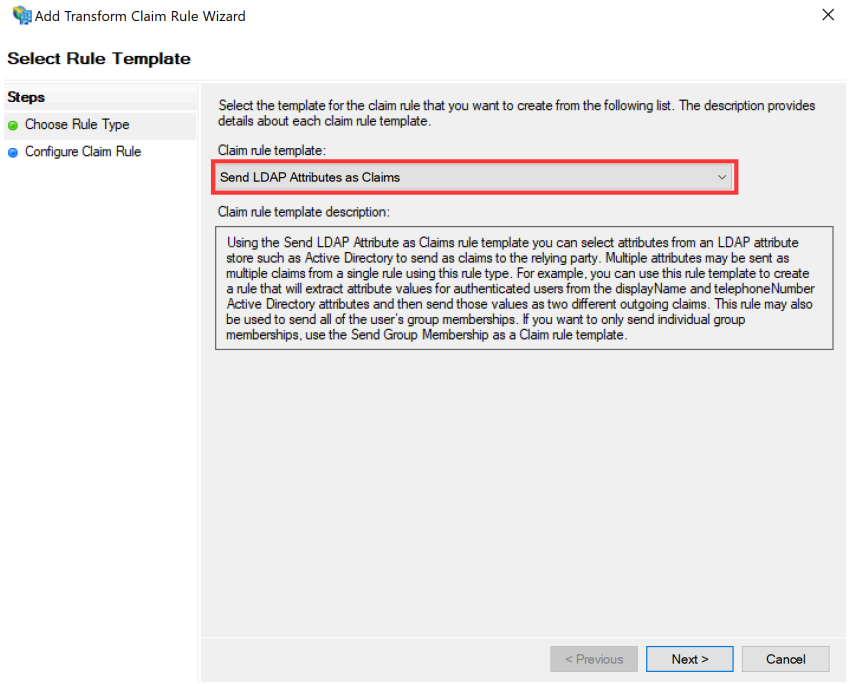
1. 检查配置无误，点击 Next。



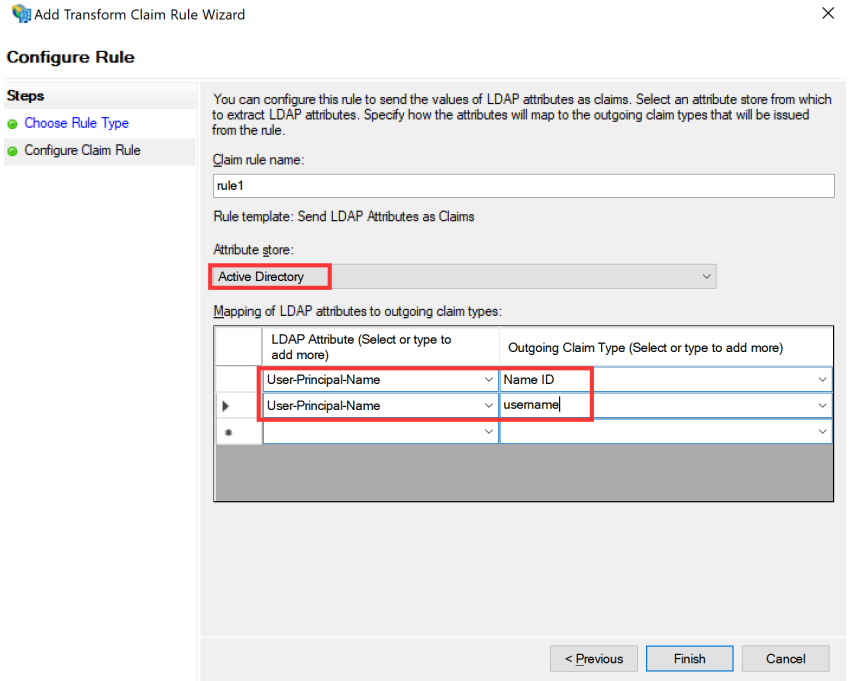
2. 点击 Configure claims issuance policy for this application，点击 Close。



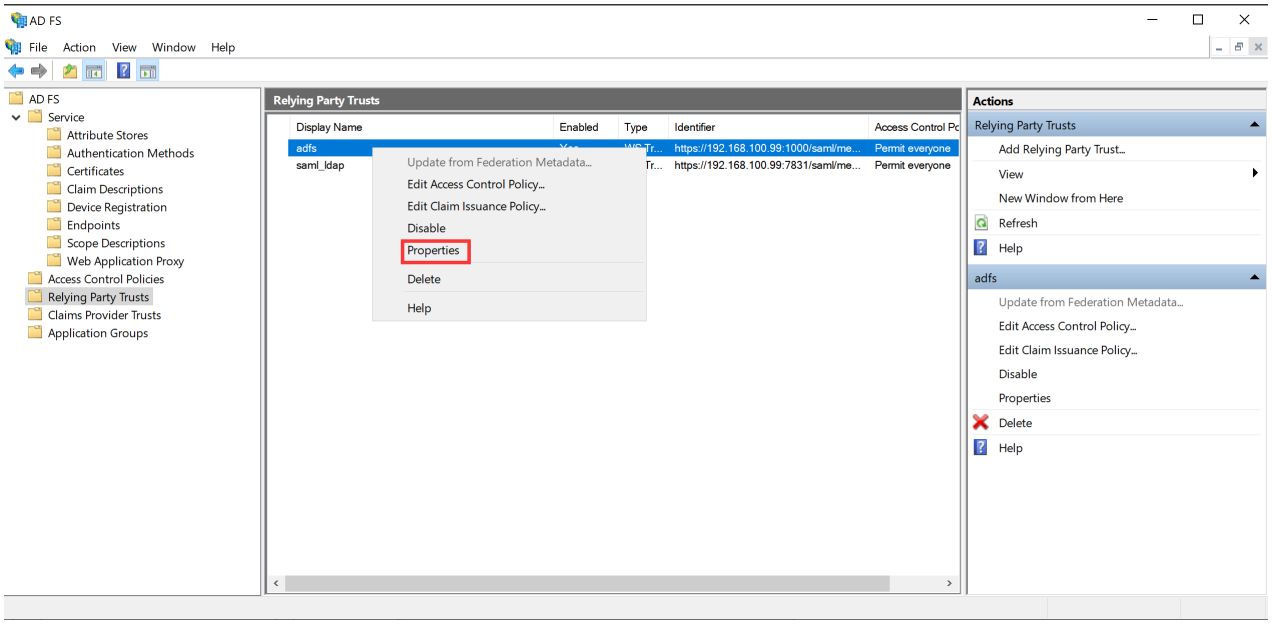
3. 在弹出的 Claim 规则中，选择模版为 Send LDAP Attributes as Claims。



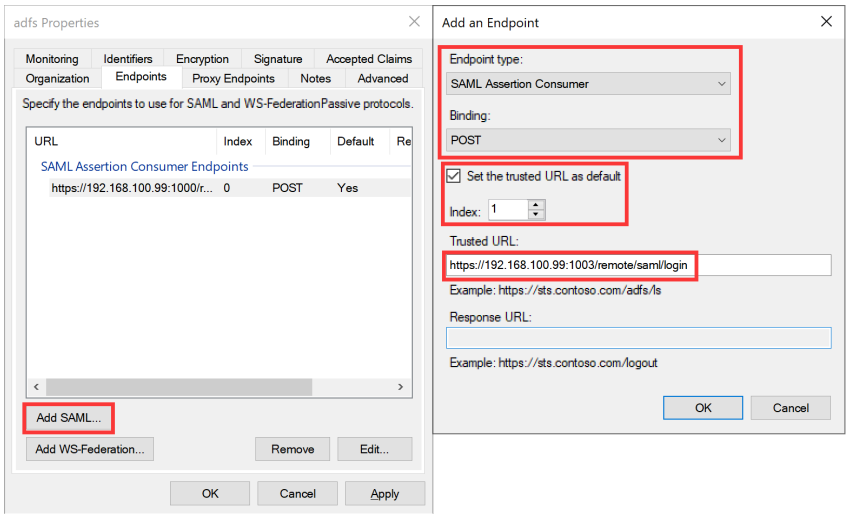
4. 配置 Claim 规则，Attribute store 选择 Active Dictionary，如下图配置映射关系，点击 Finish。



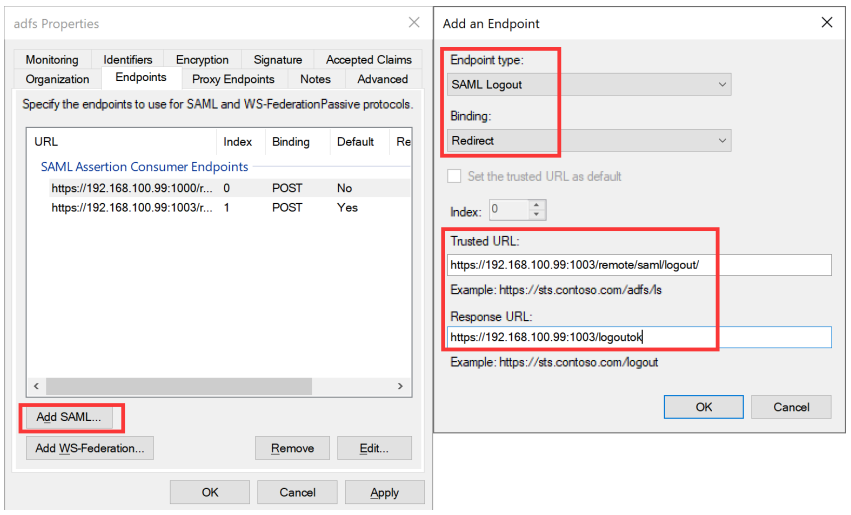
5. 右键单击已创建的 Relying Party Trust，点击 Properties。



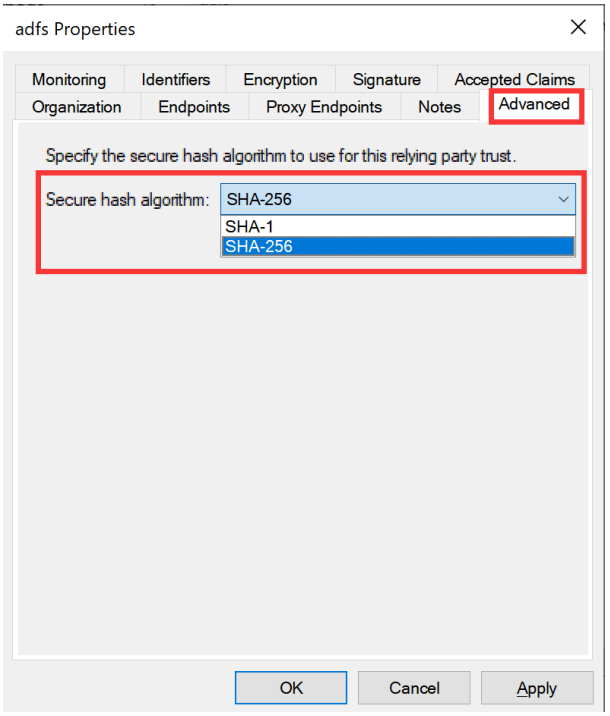
6. 选择 Endpoints 选项卡，点击 Add SAML 按钮，Endpoint type 选择 SAML Assertion Consumer，Binding 选择 POST，Index 修改为 1，Trusted URL 填写 FortiGate 内网口的 SAML 登陆 URL。



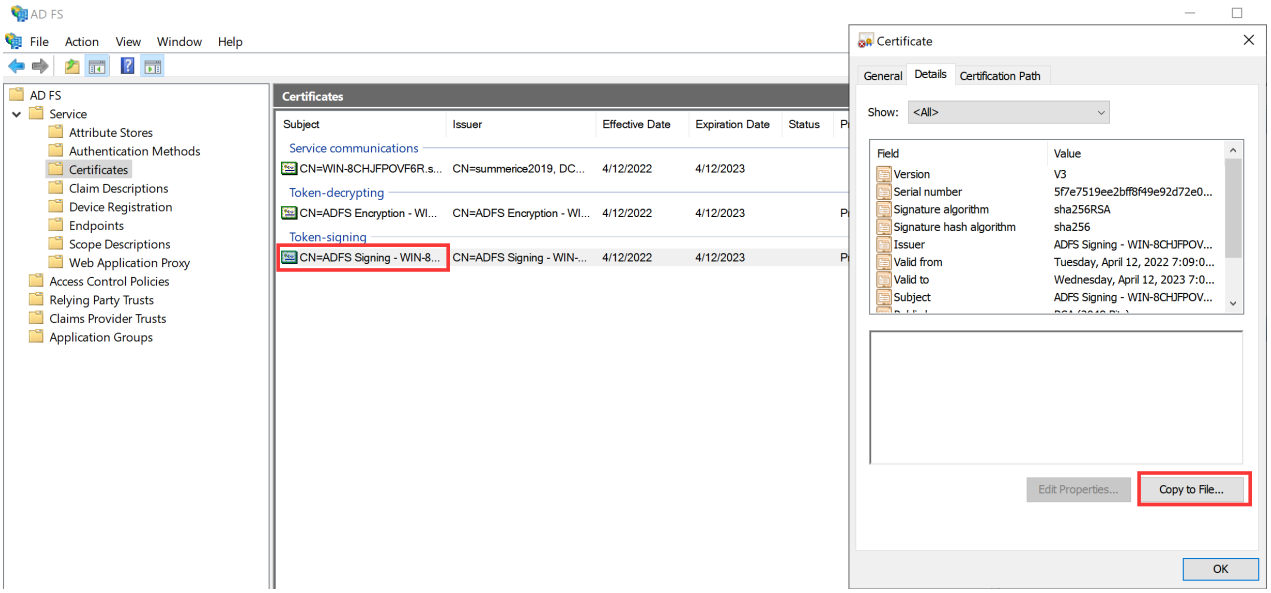
7. 再次新建一个 SAML，Endpoint type 选择 SAML logout，Binding 选择 Redirect，Trusted URL 填写 FortiGate 内网口的 SAML 登出 URL，Response URL 填写 FortiGate 的登出成功页面。



8. Advance 选项卡可以配置 hash 算法，默认为 SHA256。



9. 进入 ADFS 服务器管理器的 Service → Certificates 界面，选择 Token-signing 下的证书，导出该证书。



10. 将该证书导入 FortiGate 的 Remote Certificate 中。



1. 访问 URL：`https://<ADFS Domain name>/FederationMetadata/2007-06/FederationMetadata.xml`，例如实验这台 Windows Server 的 URL 为 `https://win-8chjfpovf6r.summerice2019.com/FederationMetadata/2007-06/FederationMetadata.xml`，下载的 XML 文件中可以看到 IDP 相关信息，并记录。
   - Metadata URL（`http://WIN-8CHJFPOVF6R.summerice2019.com/adfs/services/trust`）：



   - Login URL（`https://win-8chjfpovf6r.summerice2019.com/adfs/ls/`），Logout URL 为 `https://win-8chjfpovf6r.summerice2019.com/adfs/oauth2/logout`：



FortiGate 端配置

1. 在 FortiGate 上配置 SAML 用户，配置 SP 和 IDP 信息，注意这里填写的 SP 信息要和 ADFS 中配置的 Endpoints 信息一致，IDP 信息从 XML 文件中记录的得来，证书选择导入的 ADFS 的 Remote Certificate，算法选择 sha256。

```bash
config user saml
    edit "adfs_saml_user"
        set entity-id "https://192.168.100.99:1000/remote/saml/metadata"
        set single-sign-on-url "https://192.168.100.99:1003/remote/saml/login"
        set single-logout-url "https://192.168.100.99:1003/remote/saml/logout"
        set idp-entity-id "http://WIN-8CHJFPOVF6R.summerice2019.com/adfs/services/trust"
```

```
        set idp-single-sign-on-url "https://WIN-8CHJFPOVF6R.summerice2019.com/adfs/ls/"
        set idp-single-logout-url "https://win-8chjfpovf6r.summerice2019.com/adfs/oauth2/logout"
        set idp-cert "REMOTE_Cert_1"
        set user-name "username"
        set digest-method sha256
    next
end
```

2. 创建用户组，将 SAML 用户加入用户组。

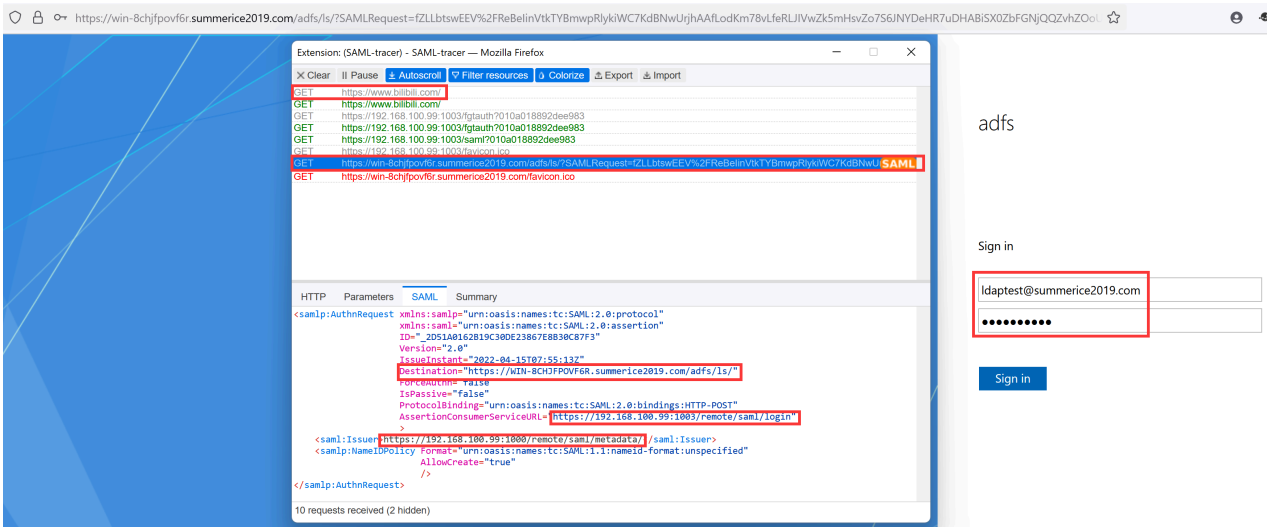
```bash
config user group
    edit "adfs_saml_group"
        set member "adfs_saml_user"
    next
end
```

3. 在 FortiGate 的防火墙策略中引用该用户组。

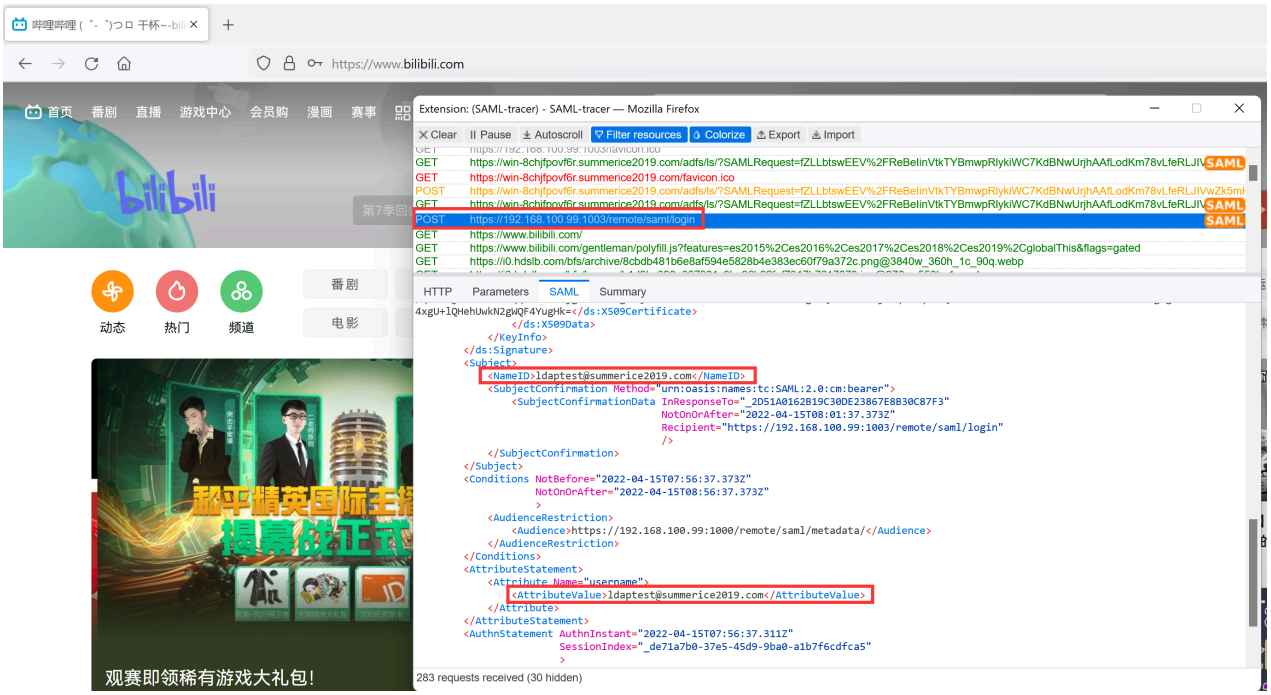
```bash
config firewall policy
    edit 1
        set srcintf "lan" "port20"
        set dstintf "port18" "port20"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        **set groups "adfs_saml_group"**
    next
end
```

验证配置

1. Client 访问 www.bilibili.com ↗ ，重定向到 IDP 弹出 ADFS 登录页面，使用 SAML tracer 可以看到 SAML SP 和 IDP 的具体 URL。



2. 输入帐号和密码，可以看到向 SP URL POST 了账户信息，登陆成功，正常弹出网页。




```
FortiGate-101F # diagnose firewall auth list

192.168.100.77, ldaptest@summerice2019.com
        src_mac: 0c:37:96:06:d7:5d
        type: fw, id: 0, duration: 146, idled: 0
        expire: 300, allow-idle: 300
        server: adfs_saml_user
        packets: in 72 out 172, bytes: in 3771 out 12163
        group_id: 2
        group_name: adfs_saml_group

----- 1 listed, 0 filtered ------
```

SSL VPN 用户认证

Windows Server 端配置

参考以上 IDP、SP URL 的端口号配置为 SSL VPN 端口号。

FortiGate 端配置

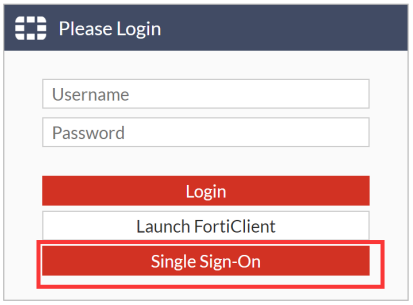
参考以上 IDP、SP URL 的端口号改为 SSL VPN 端口，并使用 SSL VPN 引用该用户组即可。

```bash
config user saml
    edit "adfs_saml_user"
        set entity-id "https://192.168.100.99:10443/remote/saml/metadata/"
        set single-sign-on-url "https://192.168.100.99:10443/remote/saml/login"
        set single-logout-url "https://192.168.100.99:10443/remote/saml/logout/"
        set idp-entity-id "http://WIN-8CHJFPOVF6R.summerice2019.com/adfs/services/trust"
        set idp-single-sign-on-url "https://WIN-8CHJFPOVF6R.summerice2019.com/adfs/ls/"
        set idp-single-logout-url "https://win-8chjfpovf6r.summerice2019.com/adfs/oauth2/logout"
        set idp-cert "REMOTE_Cert_1"
        set user-name "username"
        set digest-method sha256
    next
end
```

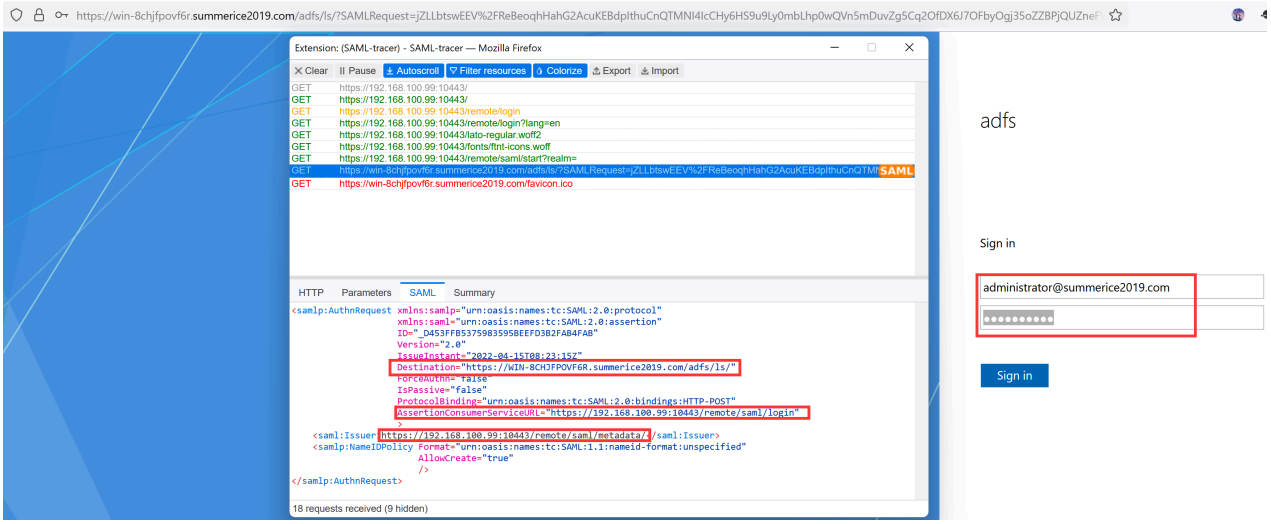
验证配置

1. 打开 SSL VPN Web 登陆界面，点击 Single Sign-On。

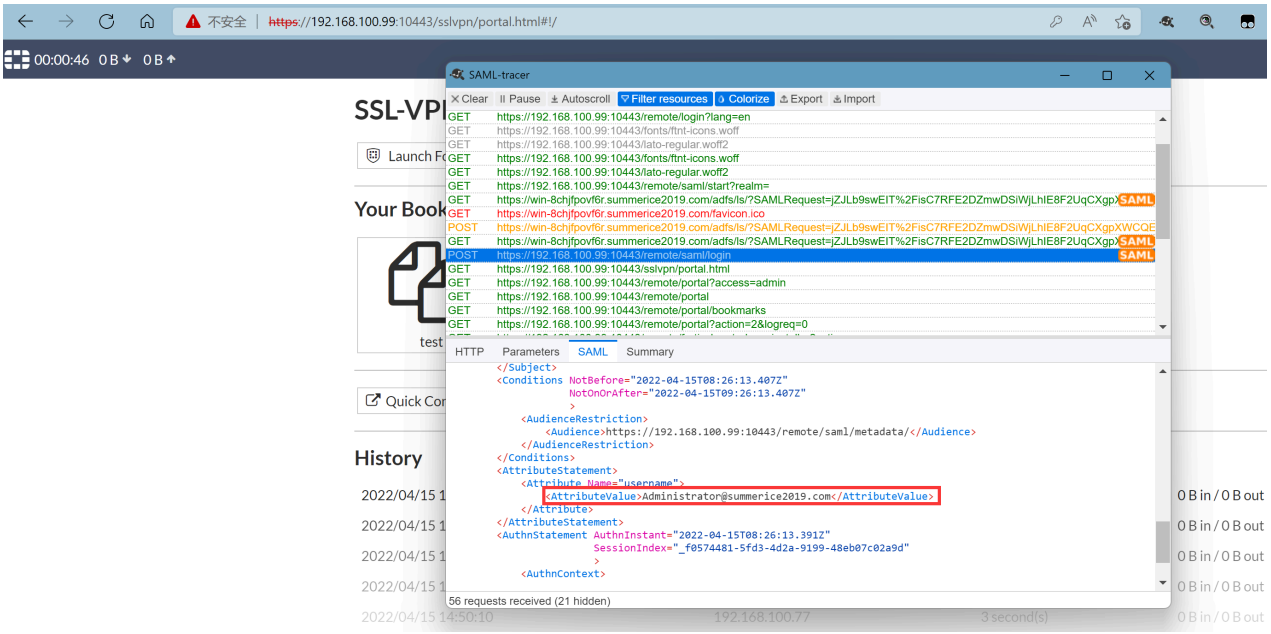




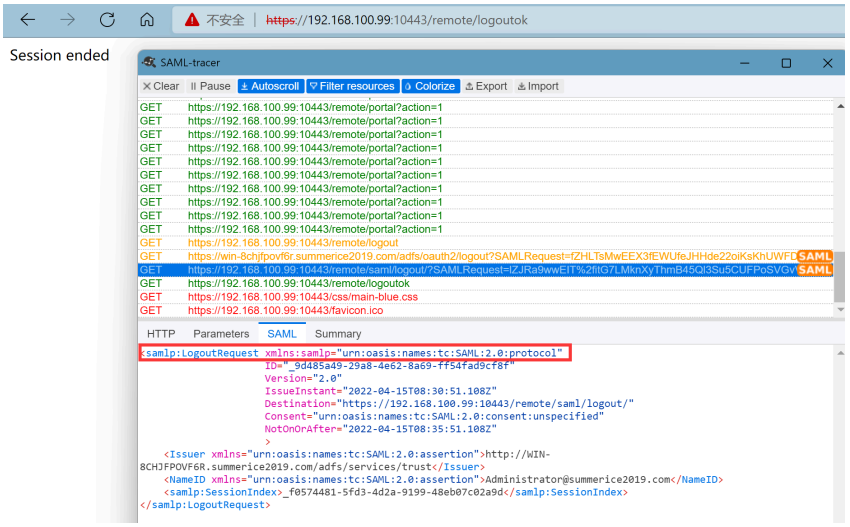
2. 重定向到 IDP 弹出 ADFS 登录页面，使用 SAML tracer 可以看到 SAML SP 和 IDP 的具体 URL。



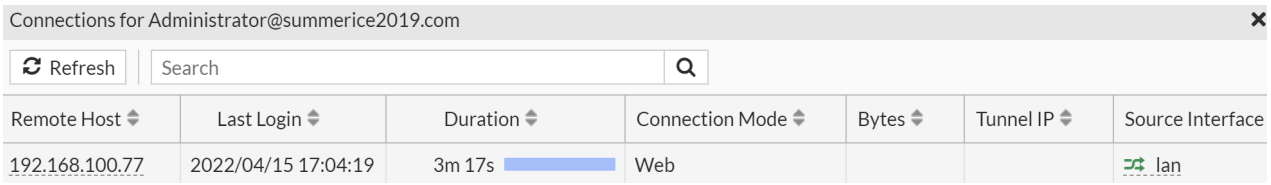
3. 输入帐号和密码，可以看到向 SP URL POST 了账户信息，登陆成功，正常弹出 SSL VPN 资源页面。



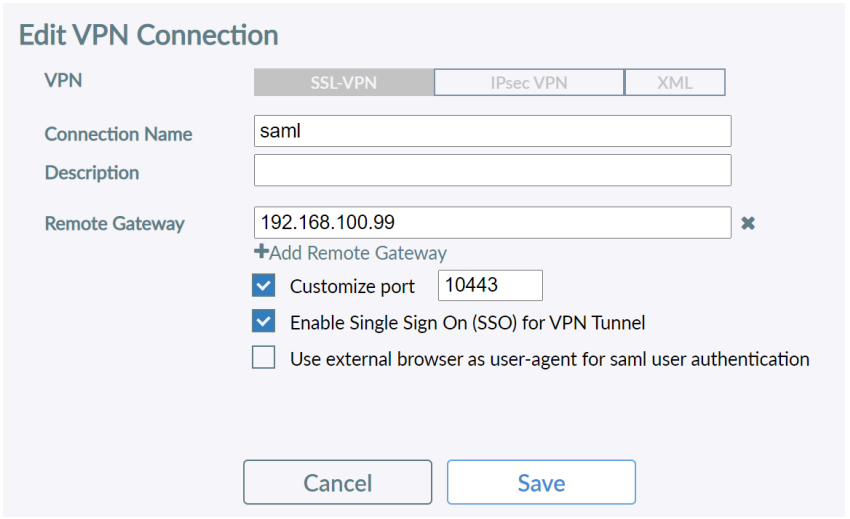
4. 登出 SSL VPN，可以看到 IDP 登出成功，且正常弹出 logoutok 页面。



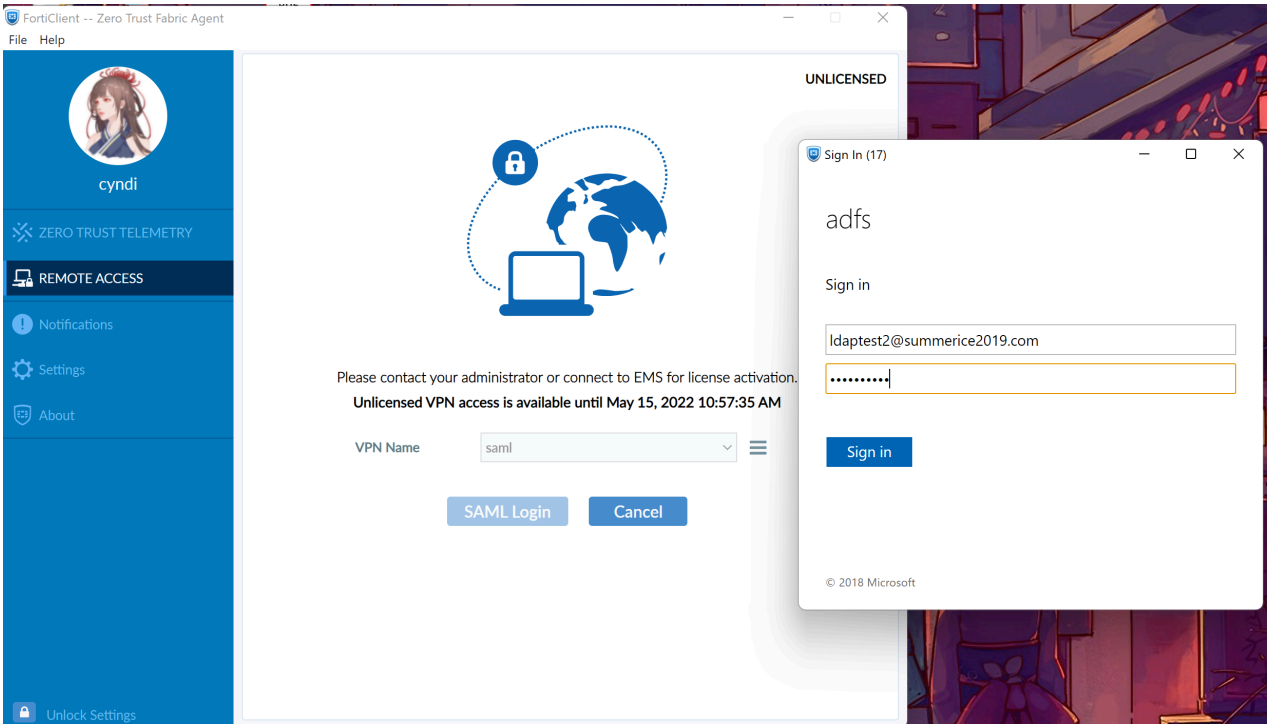
5. 在 FortiGate 查看登陆用户。



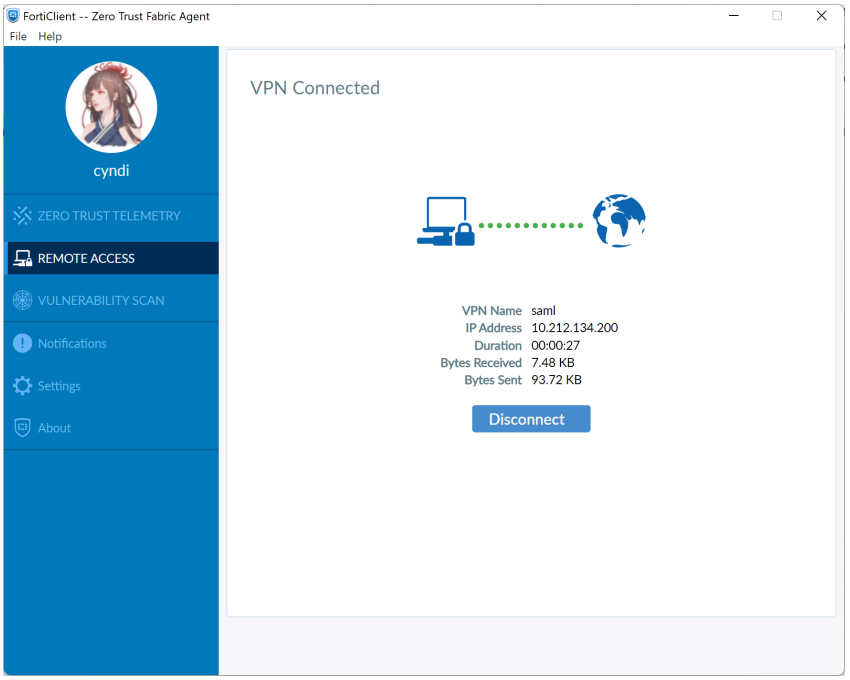
6. 使用 FortiClient 进行 SSL VPN SAML 认证。



7. 弹出 ADFS 认证页，输入帐号密码。



8. 登陆成功。



9. 在防火墙上查看登陆用户。

```
FortiGate-101F # diagnose firewall auth list

10.212.134.200, ldaptest2@summerice2019.com
        type: fw, id: 0, duration: 82, idled: 0
        expire: 28797, allow-idle: 28797
        flag(80): sslvpn
        server: adfs_saml_user
        packets: in 10898 out 14201, bytes: in 2628530 out 13760383
        group_id: 2
        group_name: adfs_saml_group
```

注意事项

1. 确保 SAML IDP 服务器的系统时间和客户端的时间一致，否则会认证失败，通过 diagnose debug application samld -1 可以看到该错误信息。

```bash
__samld_sp_login_resp [859]: Clock skew tolerance: 21

__samld_sp_login_resp [866]: Clock skew issue.
```

重要

7.0.4 GA 开始支持配置 Clock skew tolerance 超时时间：

```
config user saml
    edit <>
        set clock-tolerance <in seconds>      ( 0-300, 15 by default)
    next
end
```

2. 如果客户端访问 IDP 需要穿过防火墙，需要在认证策略前配置一条去往 IDP 服务器的策略，并开启 captive-portal-exempt。

```bash
config firewall policy
    edit 2
        set srcintf "lan"
        set dstintf "port18"
        set action accept
        set srcaddr "all"
        set dstaddr "192.168.101.77/32"
        set schedule "always"
        set service "ALL"
        **set captive-portal-exempt enable**
    next
```

3. 该错误表示 FortiGate 上的 SP、IDP URL 与 ADFS 中 Endpoints 中配置的 URL 不一致，需要检查配置。

发生错误

发生错误。有关详细信息，请与你的管理员联系。

错误详细信息

- Activity ID: 3be37eb3-f2db-4985-0400-0080040000e5
- Relying party: fortigate
- Error details: MSIS3200: 未在信赖方信任 “https://192.168.100.99:10443/remote/saml/metadata/” 上配置与请求所指定的 AssertionConsumerService URL “https://192.168.100.99:10443/remote/saml/login” 前缀匹配的任何 AssertionConsumerService。
- Node name: 0c0538e8-6ee4-4b61-9f20-b03002a914f4
- Error time: Tue, 12 Apr 2022 03:22:05 GMT
- Cookie: enabled
- User agent string: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75 Safari/537.36 Edg/100.0.1185.36

4. 该错误表示客户端未安装 ADFS 所属 AD 域的 CA 证书。

An error occurred

An error occurred. Contact your administrator for more information.

Error details

- Activity ID: ec30abac-dbf8-4c24-0f00-0080000000ea
- Relying party: adfs
- Error details: MSIS0037: No signature verification certificate found for issuer 'https://192.168.100.99:10443/remote/saml/metadata'.
- Node name: f24c844a-4b5f-4ead-8043-05964ba0f138
- Error time: Wed, 13 Apr 2022 03:14:40 GMT
- Cookie: enabled
- User agent string: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0

5. 如果 SAML 弹出的登陆界面超时时间过短，来不及输入密码就过期了，可以通过以下命令修改超时时间。

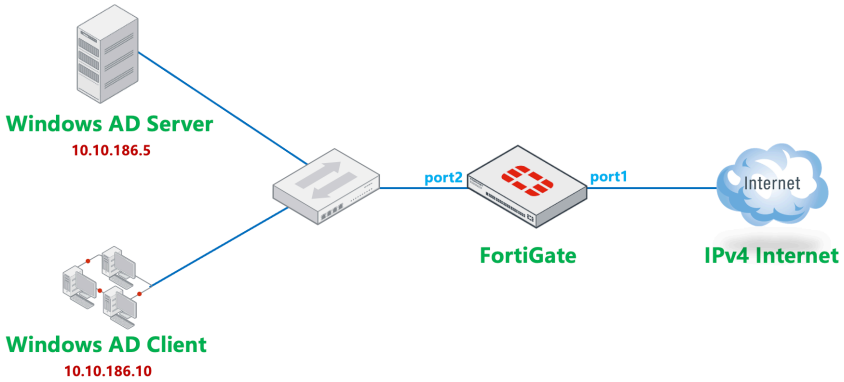
```bash
config system global
    set remoteauthtimeout 60    //Number of seconds that the FortiGate waits for responses from remote RADIUS, LDAP, or TACACS+ authentication servers. (1-300 sec, default = 5).
end
```

显式代理 SAML 认证（Windows ADFS）

网络需求

在防火墙上启用显示代理，并且配置基于 AD FS 的 SAML SSO 进行用户验证。

网络拓扑



1. FortiGate 的 LAN 口 port2 连接 Windows AD 服务器和 Windows AD 电脑。
2. FortiGate 的 WAN 口 port1 连接 Internet。
3. port2 上启用 explicit proxy 和 captive portal，验证通过 SAML SSO 完成。SAML SSO 由 Windows AD 服务器上的 AD FS 提供。
4. Windows AD 电脑上设置 FortiGate 的 port2 作为上网的代理服务器。
5. 完成配置后，Windows AD 服务器作为 SAML IdP（提供验证服务），FortiGate 作为 SAML SP（发起验证请求）。

配置步骤

Windows 服务器配置

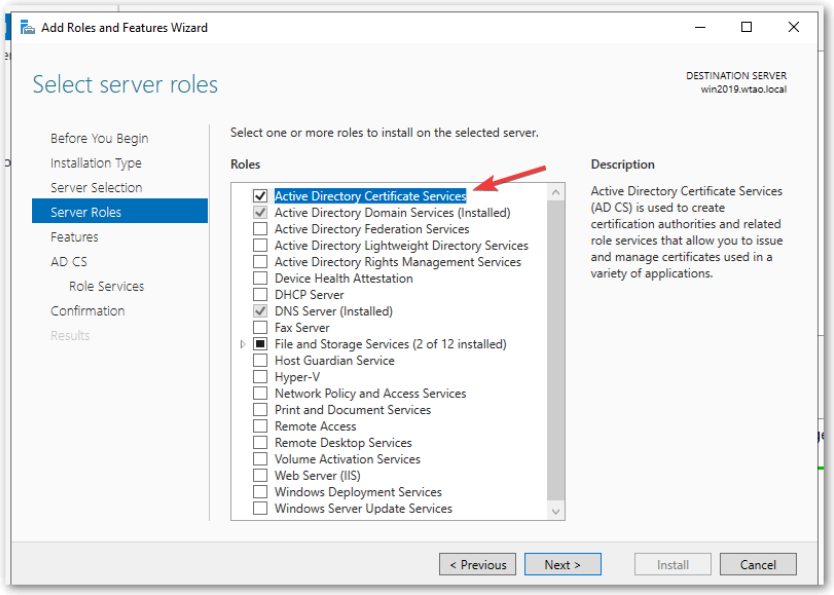
1. 在本示例中，Windows AD 服务器的 OS 是 Windows 2019, version 1809。FortiGate 是 VM 7.0.12，SAML SSO 验证服务由 Windows AD 服务器上的 AD FS (Active Directory Federation Service) 提供。
2. Windows AD 服务器和 FortiGate 的时间需要在同一时区，时间差默认不能超过 15 秒。建议都使用 NTP。可以在 FortiGate 上修改时间差的容忍值 ↗ 。

安装 LDAP 服务

1. LDAP 服务由 Active Direcotry Domain Service，或者 Active Directory Lightweight Directory Services 提供。
2. 在本示例中，Windows 服务器 10.10.186.5 已经安装了 Active Direcotry Domain Service。

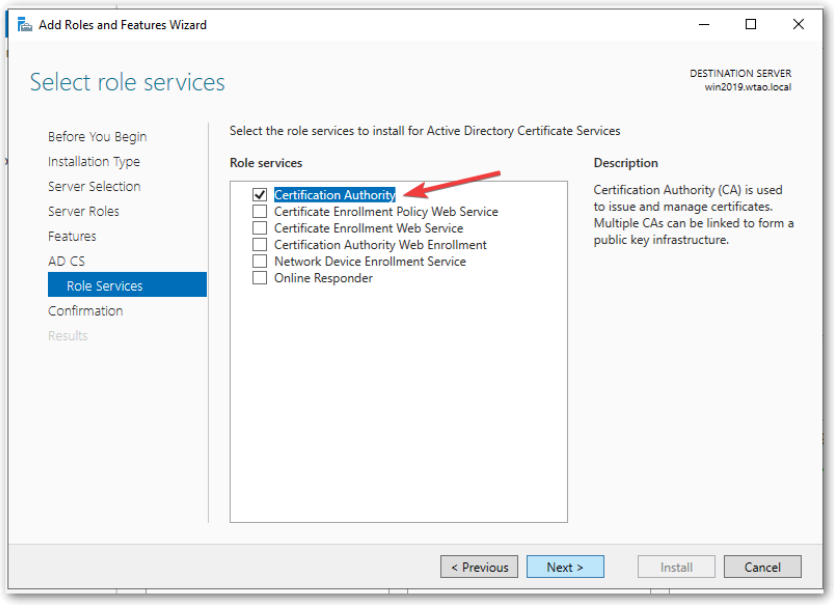
安装 AD Certificate Services 证书服务

1. AD FS 服务需要用到 SSL 证书，如果没有证书，可以在 Windows Server 上安装 AD DC，然后在 AD DC 上为 AD FS 服务新建证书。
2. 如果你已经有供 AD FS 服务使用 SSL 证书，或者已经安装了 AD DC，可以跳过这一步。
3. 你也可以使用 openssl 或者其他工具为 AD FS 服务生成证书，然后将证书导入 AD FS，将 CA 导入 Windows AD 服务器。
4. 在 Server Manager 中安装” Active Directory Certificate Services”。



5. Features: 直接 Next。

6. Role Services: Certification Authority。



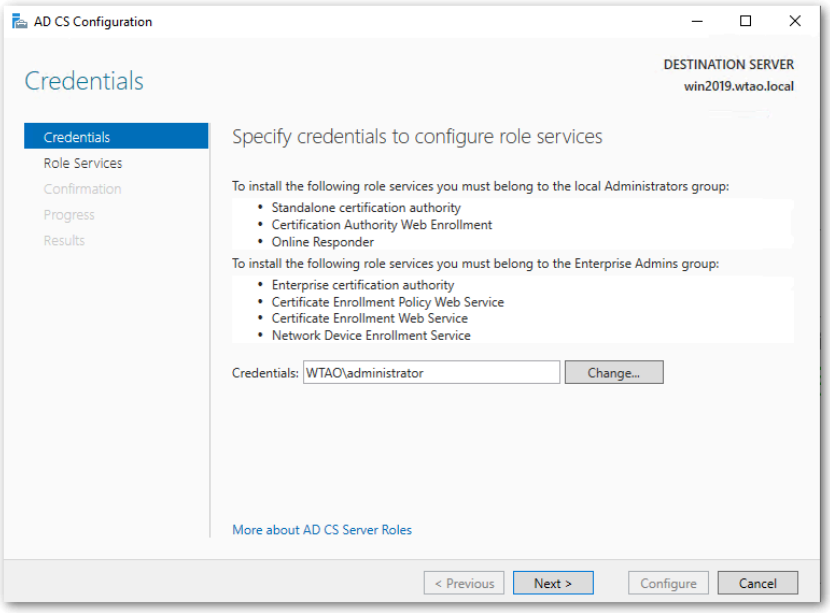
7. 使用默认配置，完成安装。

配置 AD 证书服务

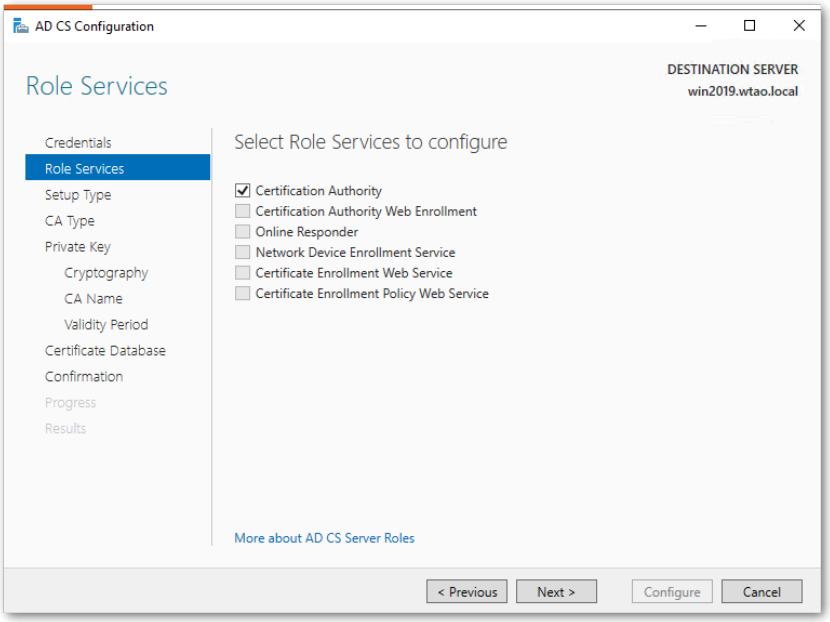
> **重要**
>
> 如果你已经有供 AD FS 服务使用 SSL 证书，可以跳过这一步。

1. 打开 AD CS configuration。

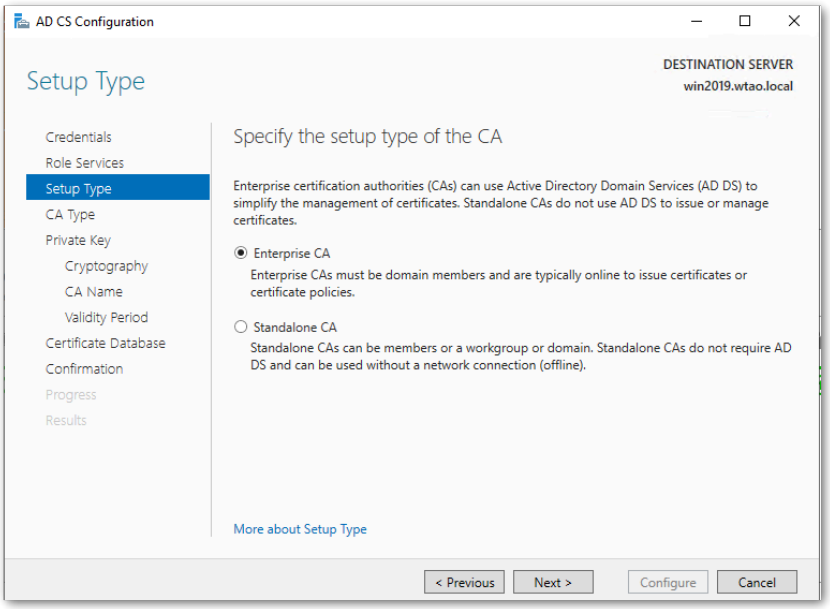
2. Credentials: 默认。



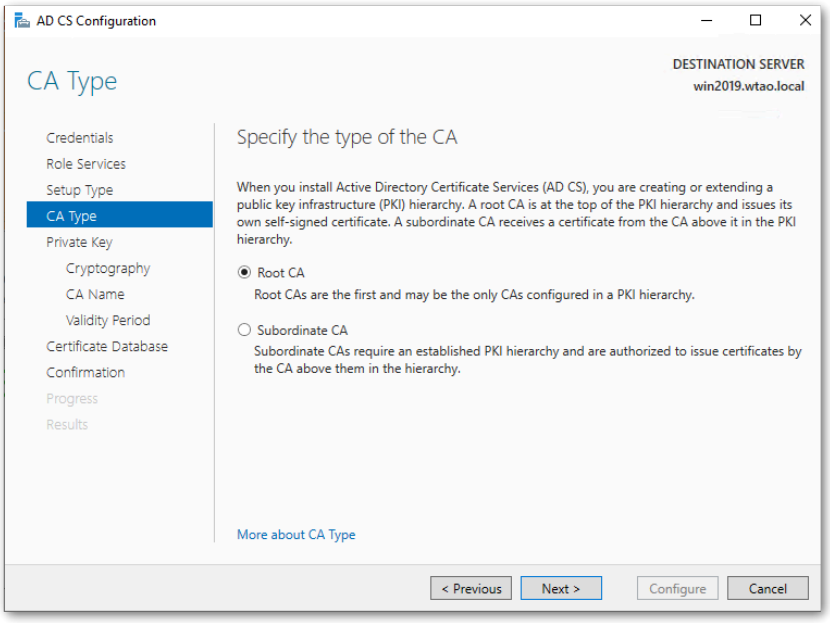
3. Role Serivces 选择 Certificateion Authority。



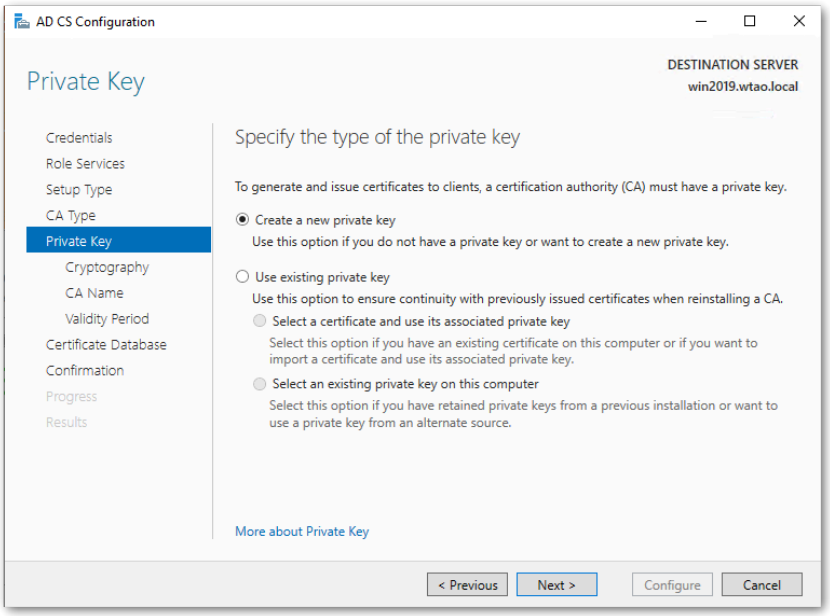
4. Setup Type 选择 Enternprise CA。

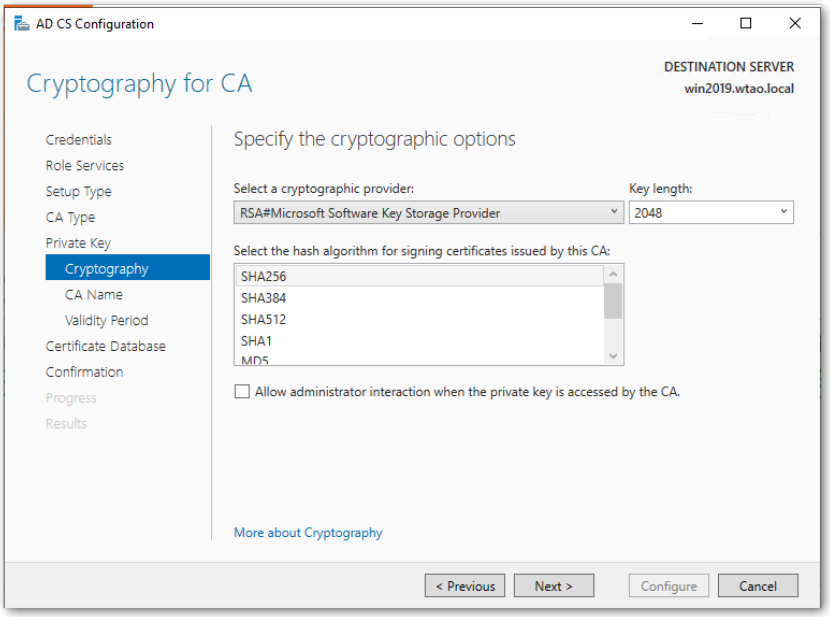


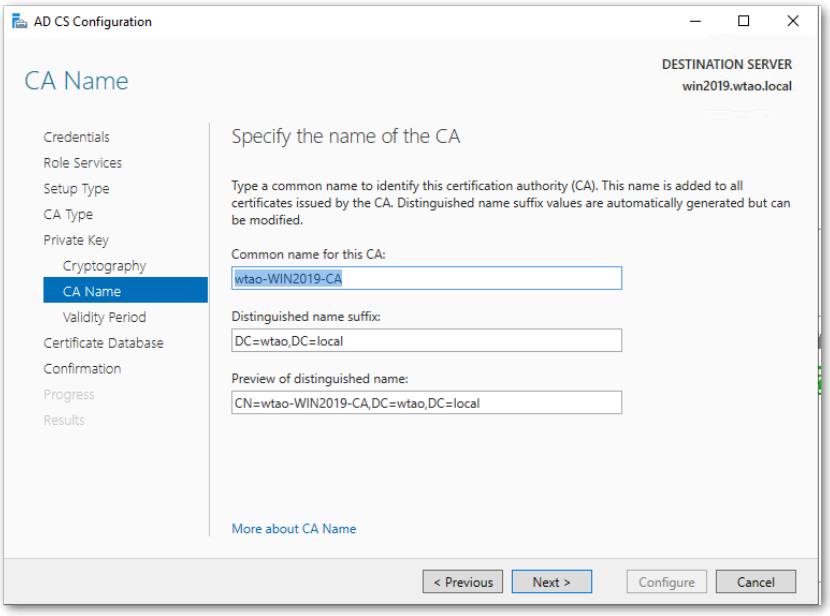
5. CA Type 选择 Root CA。

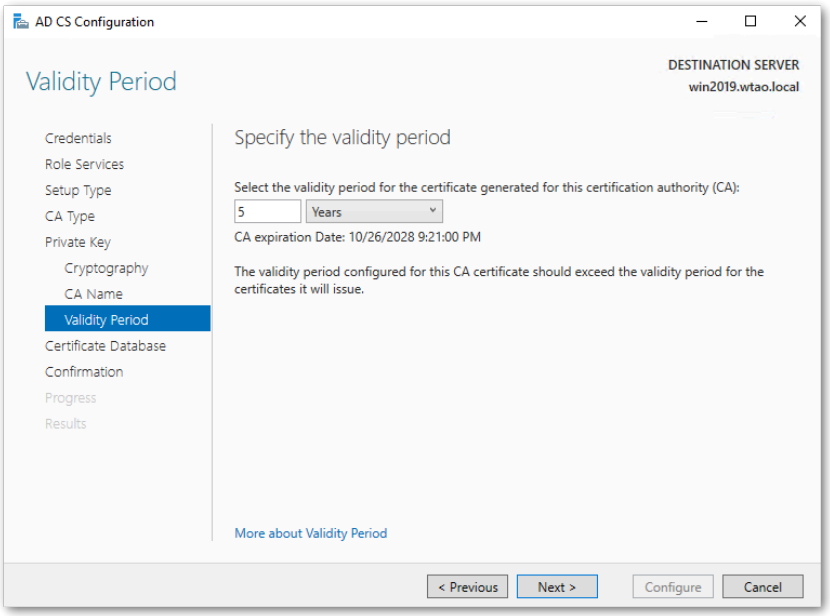


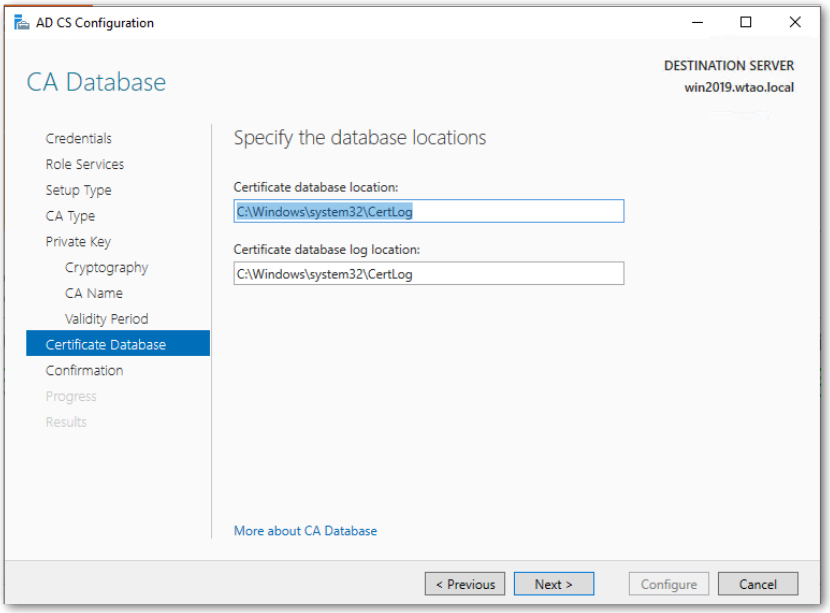
6. 使用默认配置，完成 AD CS 的配置。

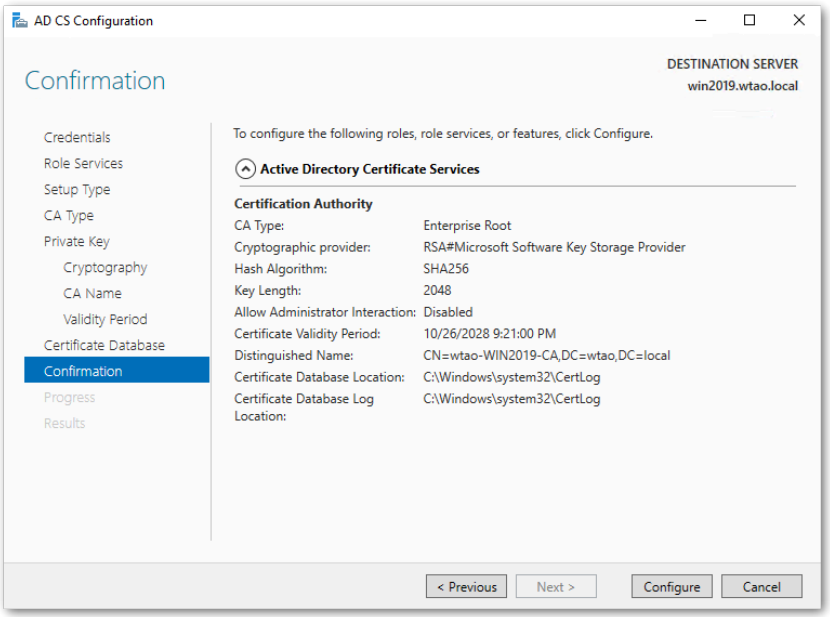

AD CS Configuration
Private Key
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
Specify the type of the private key
To generate and issue certificates to clients, a certification authority (CA) must have a private key.
Create a new private key
Use this option if you do not have a private key or want to create a new private key.
Use existing private key
Use this option to ensure continuity with previously issued certificates when reinstalling a CA.
Select a certificate and use its associated private key
Select this option if you have an existing certificate on this computer or if you want to import a certificate and use its associated private key.
Select an existing private key on this computer
Select this option if you have retained private keys from a previous installation or want to use a private key from an alternate source.
More about Private Key
< Previous
Next >
Configure
Cancel


AD CS Configuration
Cryptography for CA
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
Specify the cryptographic options
Select a cryptographic provider:
RSA#Microsoft Software Key Storage Provider
Key length:
2048
Select the hash algorithm for signing certificates issued by this CA:
SHA256
SHA384
SHA512
SHA1
MD5
Allow administrator interaction when the private key is accessed by the CA.
More about Cryptography
< Previous
Next >
Configure
Cancel


AD CS Configuration
CA Name
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
Specify the name of the CA
Type a common name to identify this certification authority (CA). This name is added to all certificates issued by the CA. Distinguished name suffix values are automatically generated but can be modified.
Common name for this CA:
wtao-WIN2019-CA
Distinguished name suffix:
DC=wtao,DC=local
Preview of distinguished name:
CN=wtao-WIN2019-CA,DC=wtao,DC=local
More about CA Name
< Previous
Next >
Configure
Cancel


AD CS Configuration
Validity Period
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
Specify the validity period
Select the validity period for the certificate generated for this certification authority (CA):
5
Years
CA expiration Date: 10/26/2028 9:21:00 PM
The validity period configured for this CA certificate should exceed the validity period for the certificates it will issue.
More about Validity Period
< Previous
Next >
Configure
Cancel


AD CS Configuration
CA Database
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
Specify the database locations
Certificate database location:
C:\Windows\system32\CertLog
Certificate database log location:
C:\Windows\system32\CertLog
More about CA Database
< Previous
Next >
Configure
Cancel

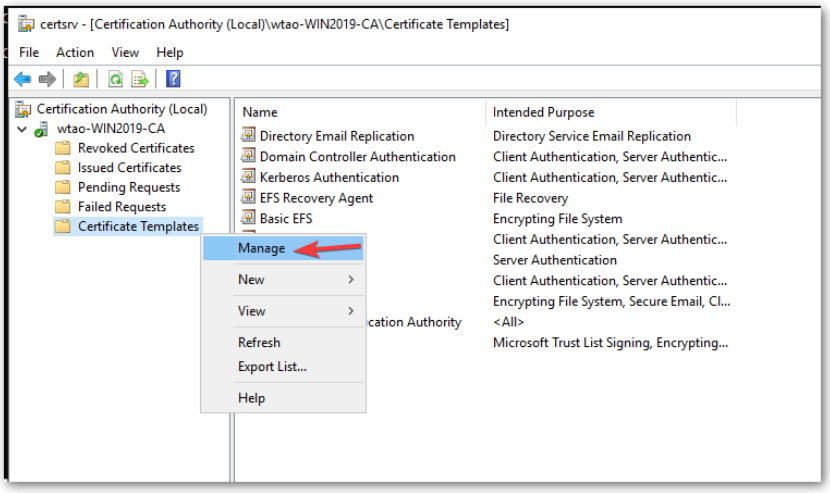

AD CS Configuration
Confirmation
DESTINATION SERVER
win2019.wtao.local
Credentials
Role Services
Setup Type
CA Type
Private Key
Cryptography
CA Name
Validity Period
Certificate Database
Confirmation
Progress
Results
To configure the following roles, role services, or features, click Configure.
Active Directory Certificate Services
Certification Authority
CA Type: Enterprise Root
Cryptographic provider: RSA#Microsoft Software Key Storage Provider
Hash Algorithm: SHA256
Key Length: 2048
Allow Administrator Interaction: Disabled
Certificate Validity Period: 10/26/2028 9:21:00 PM
Distinguished Name: CN=wtao-WIN2019-CA,DC=wtao,DC=local
Certificate Database Location: C:\Windows\system32\CertLog
Certificate Database Log Location: C:\Windows\system32\CertLog
< Previous
Next >
Configure
Cancel

在 AD 证书服务中创建证书模板

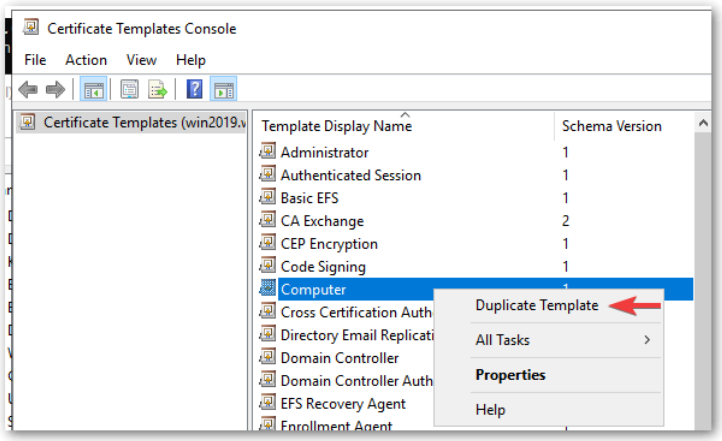
> **重要**
>
> 如果你已经有供 AD FS 服务使用 SSL 证书，可以跳过这一步。

1. 在 Windows CMD 中运行 certsrv.msc。

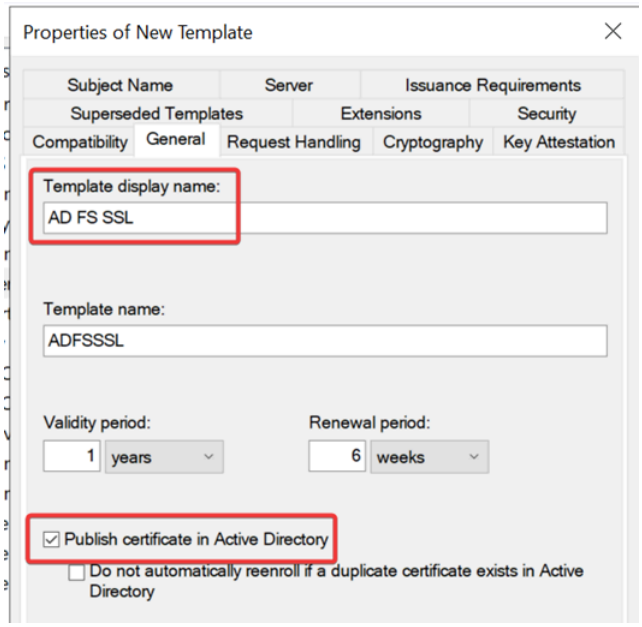
2. 右键点击 Certificate Templates，点击 Manage。



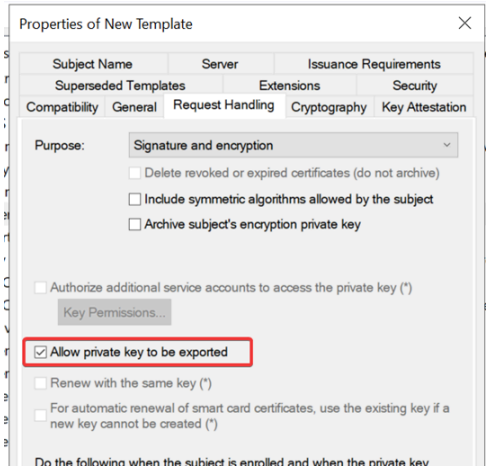
3. 右键点击 Computer，点击 Duplicate Template。



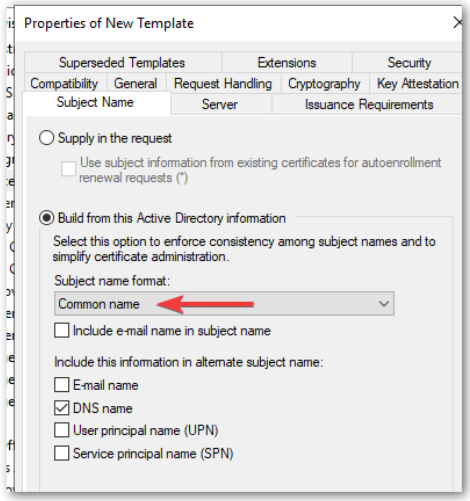
4. 在 General 标签页，给模板取名，并且选中” Publish certificate in Active Directory”。



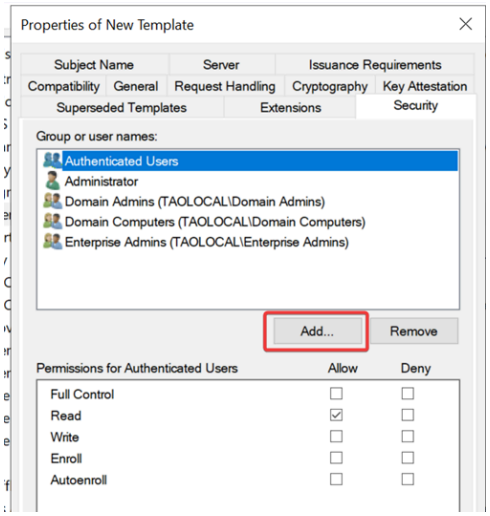
5. 在 Request Handling 标签页，选中” Allow private key to be exported”。



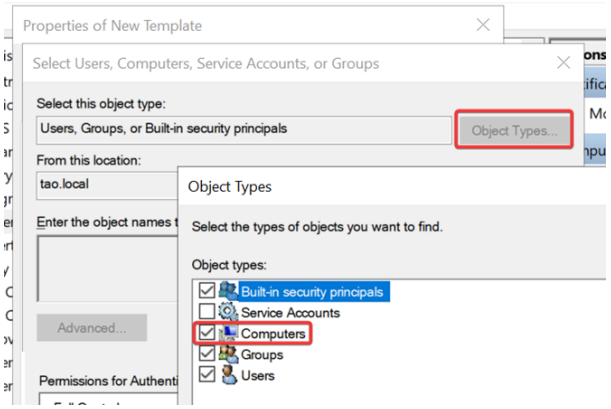
6. 在 Subject Name 标签页，设置 Subject name format: Common name。



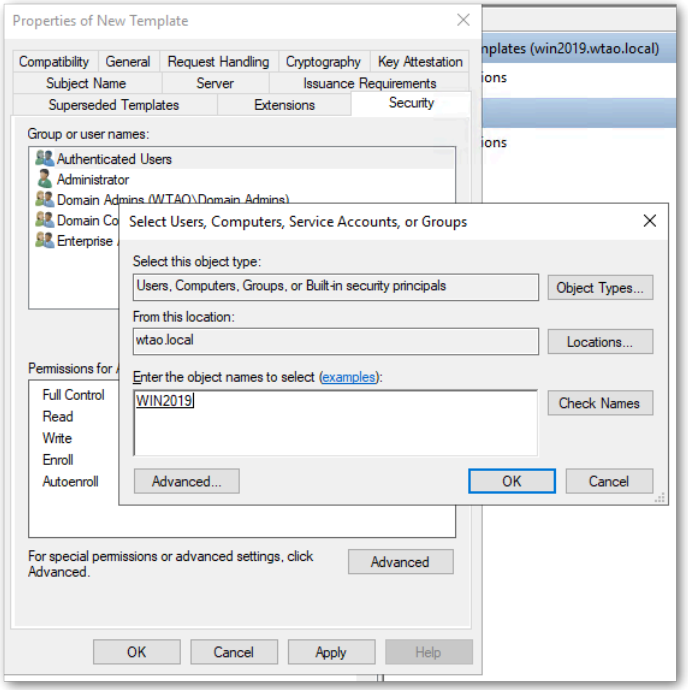
7. 在 Security 标签页，点击 Add…。



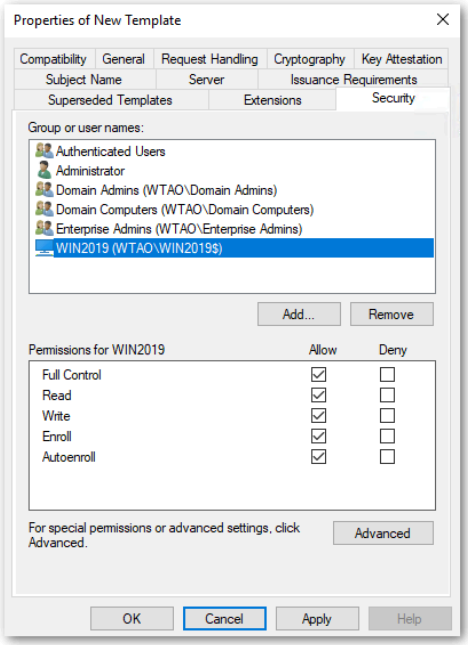
8. 修改 Select this object type: Computers。



9. Enter the ojbect names to select 中找到 AD 证书服务器。



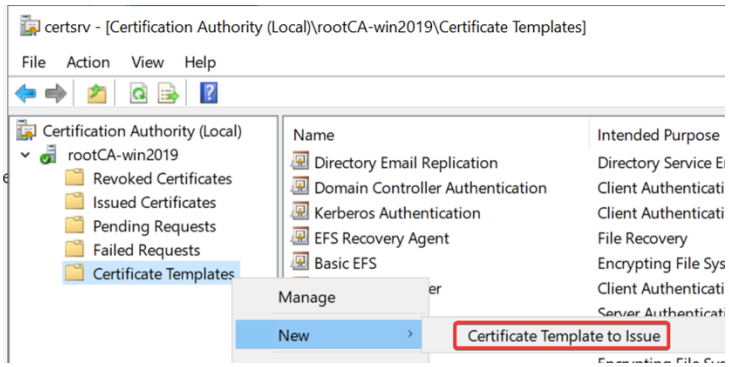
.0. 赋予 AD 证书服务器 Full Control。



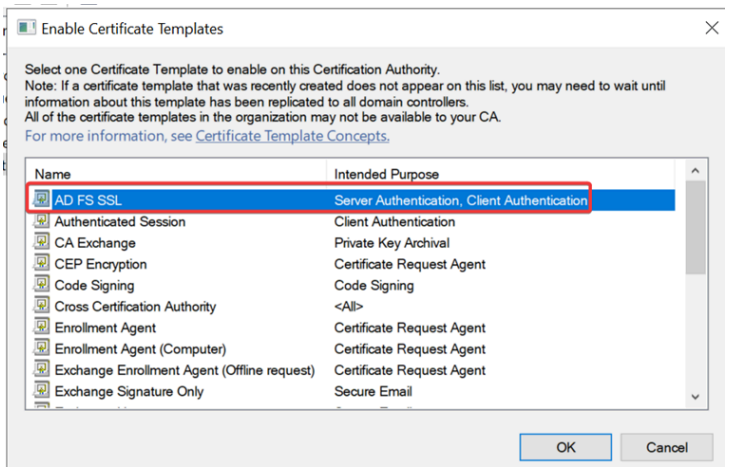
1. 点击 OK 完成证书模板配置。

2. 回到 certsrv.msc。

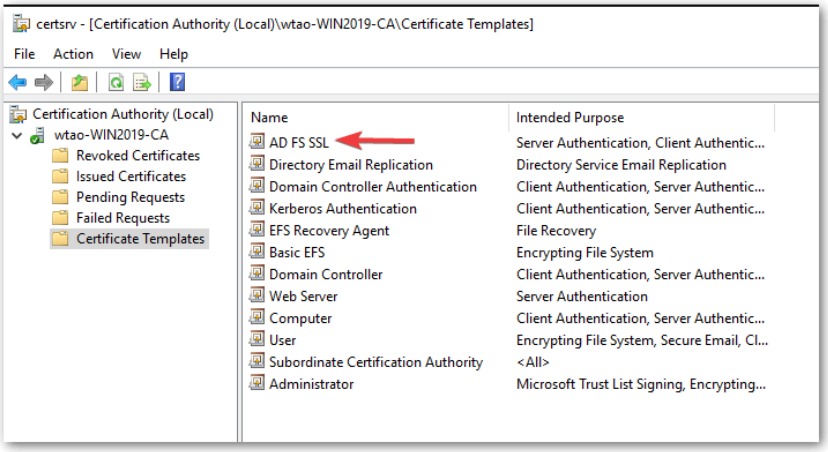
3. 右键点击 Certificate Templates，点击 New > Certificate Template to Issue。



4. 选中刚才创建的证书模板。点击 OK。



5. 在列表中可以看到发布的证书模板。

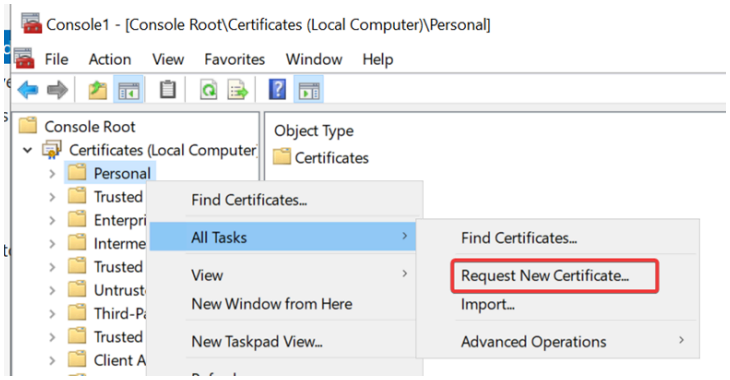


为 AD FS 创建证书

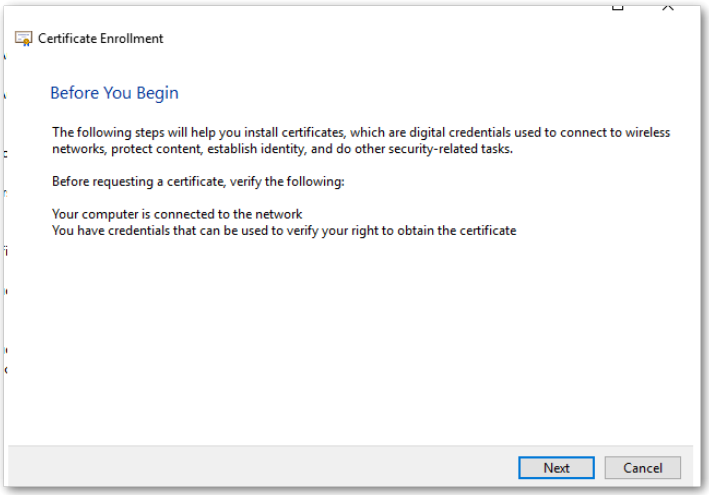
> **重要**
>
> 如果你已经有供 AD FS 服务使用 SSL 证书，可以跳过这一步。

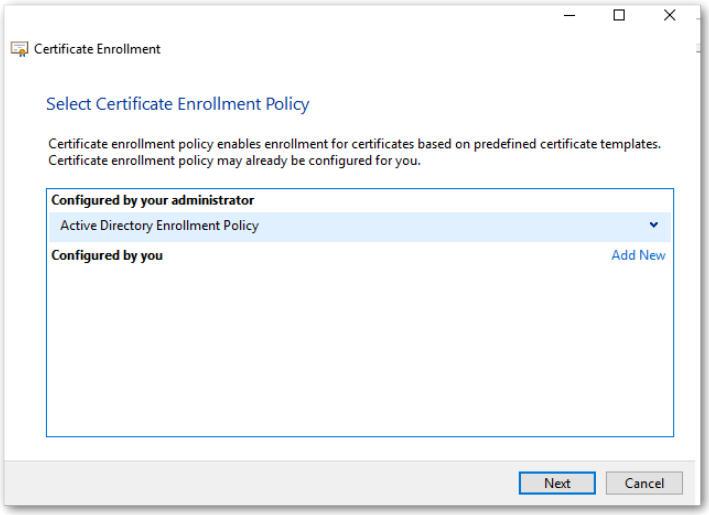
1. 在 Windows CMD 中运行 certlm.msc。

2. 右键点击 Personal， 点击 All Taks > Request New Certificate…

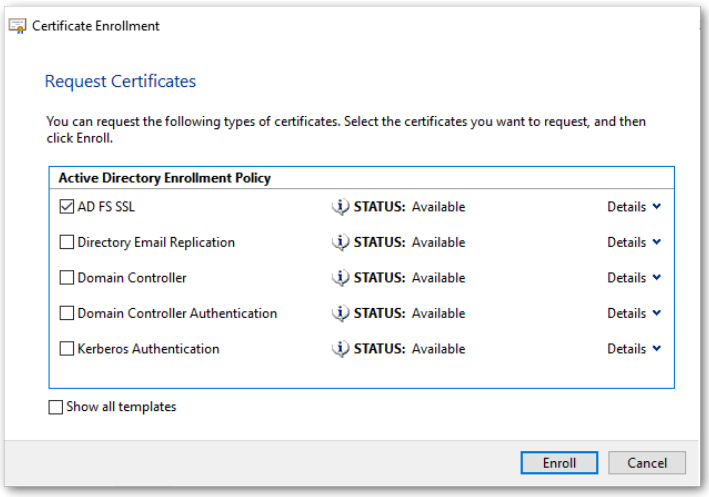


3. 点击 Next。

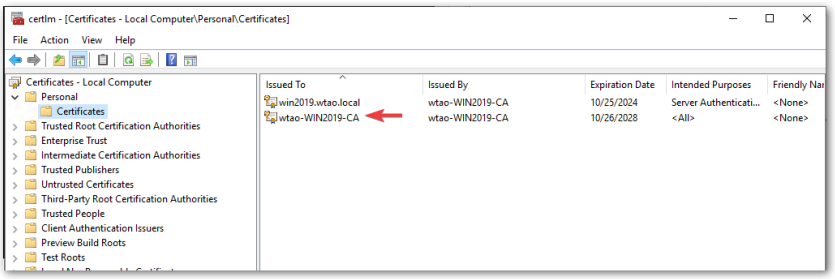




4. 选中刚才创建的证书模板，点击 Enroll。



5. 可以在 Personal > Certificate 中看到新创建的证书，这个证书将会提供给 AD FS 使用。

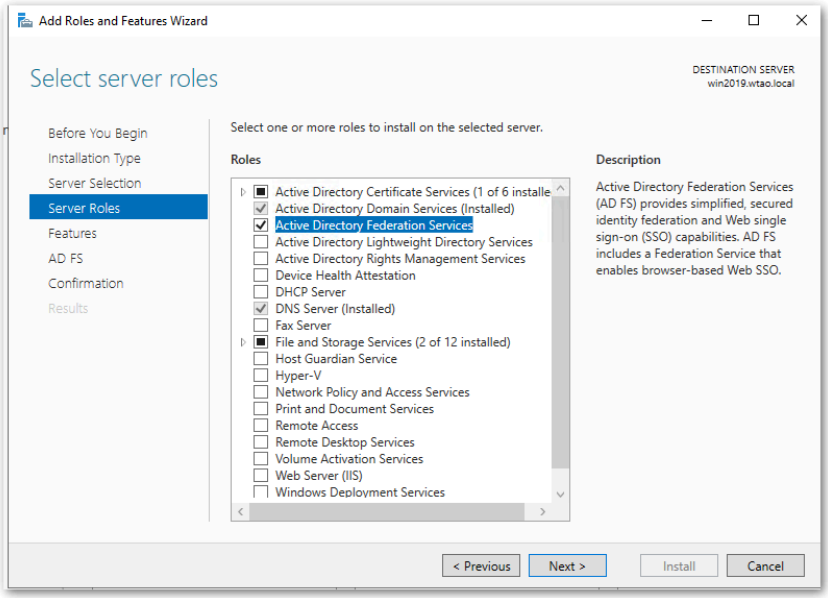


6. 这是生成的证书，供参考：AD FS SSL.cer ↗。

安装 AD FS

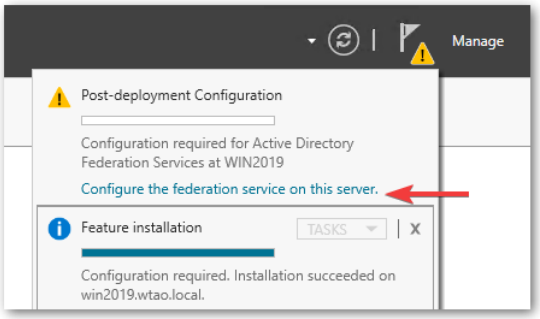
> **重要**
>
> 如果你已经安装了 AD FS，可以跳过这一步。

在 Server Manager 中安装 Active Direcotry Federation Services。选择默认配置，完成安装。

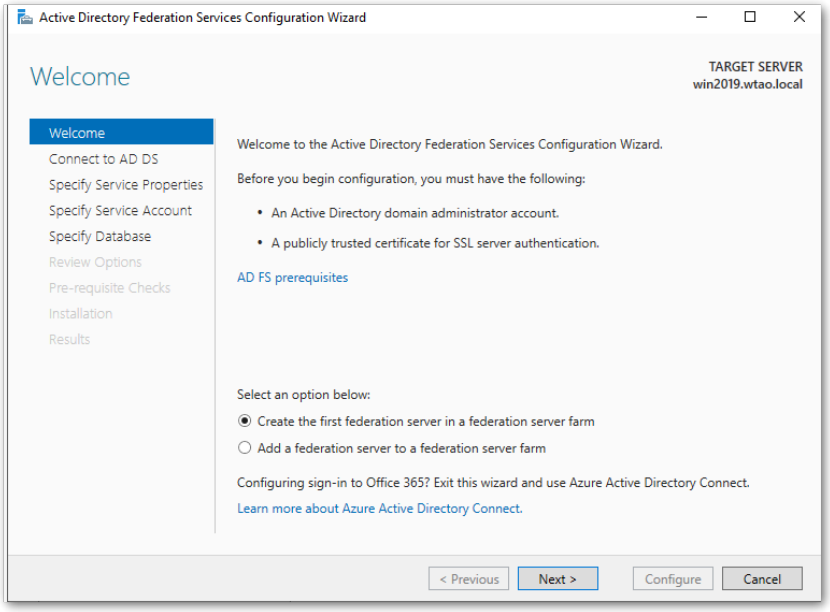


初始化 AD FS

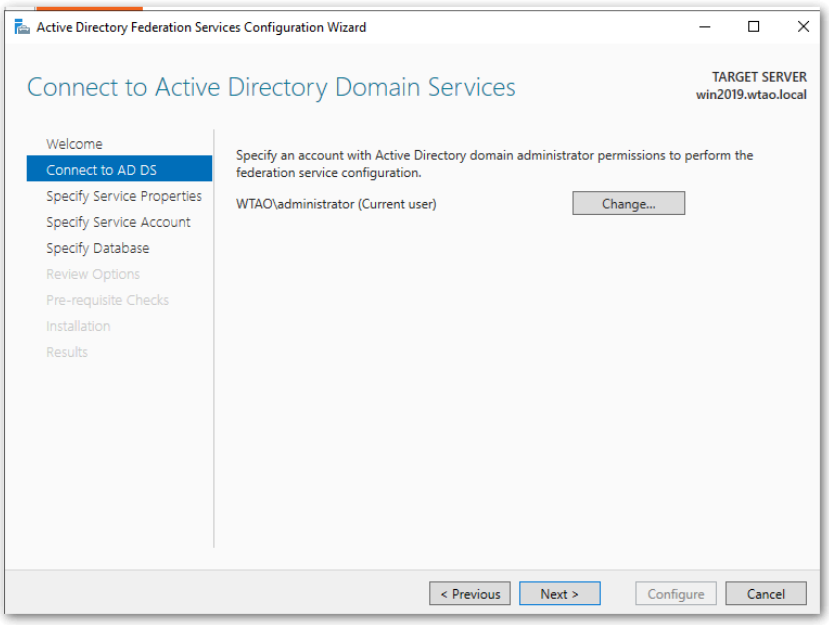
1. 在 Server Manger 点击 Configure the federation service on this server，初始化 AD FS。



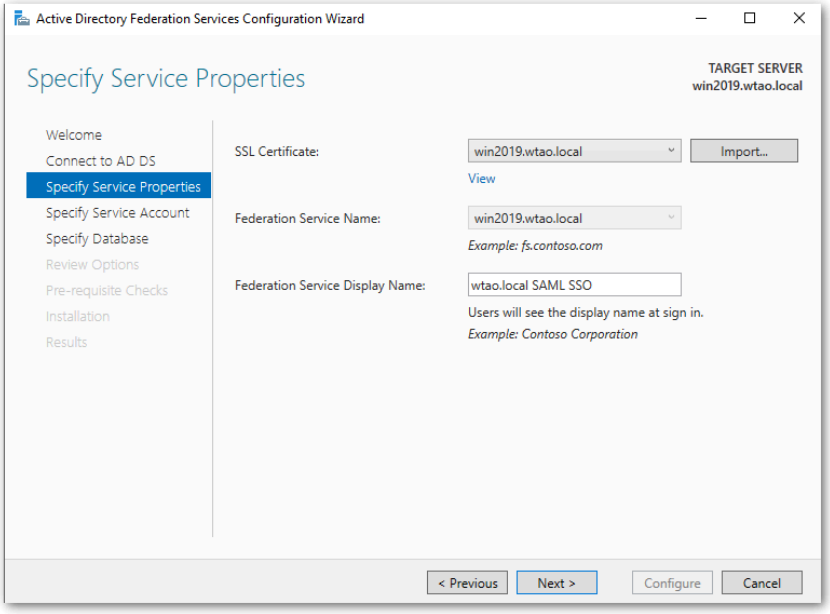
2. Welcome，保持默认。



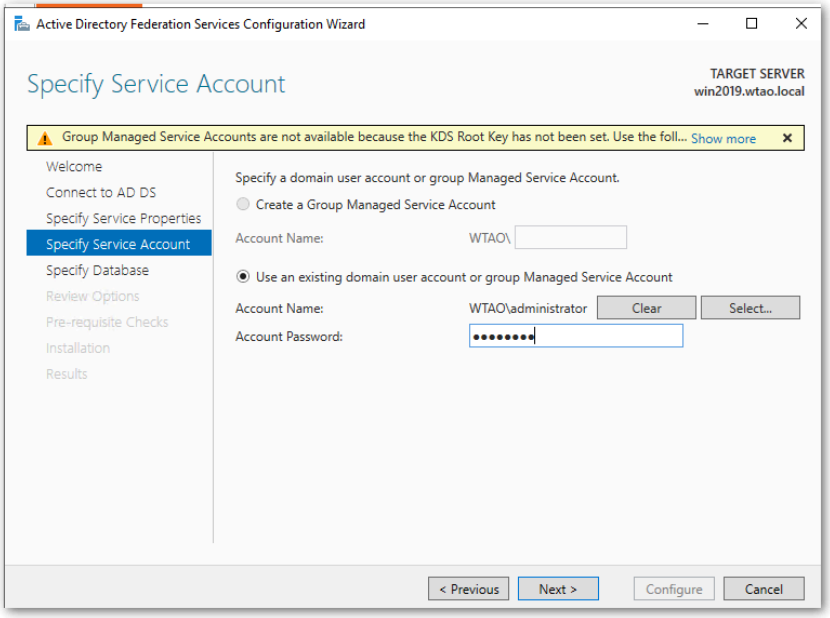
3. Connect to AD DS，保持默认。



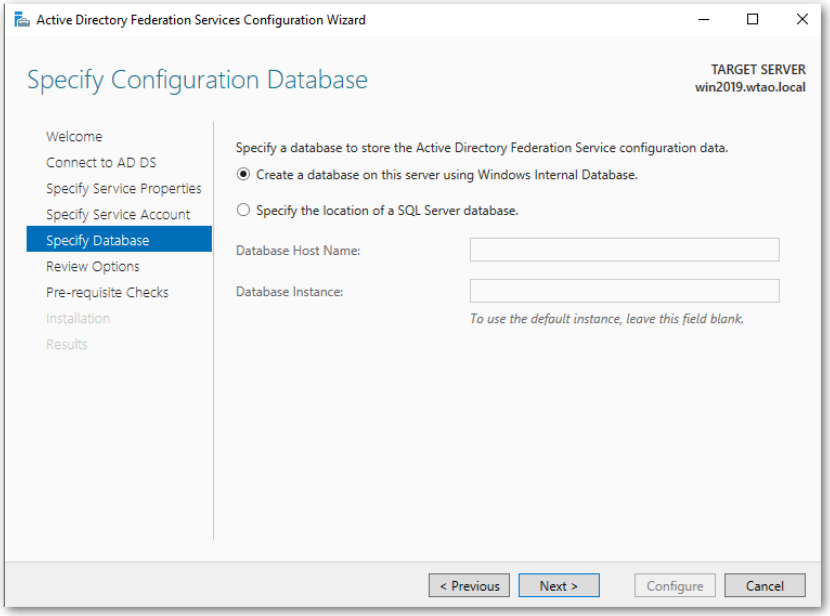
4. Specify Service Properties：SSL Certificate: 选择刚才生成的证书；Federation Service Display Name：可按需输入。这些字符将出现在 SSO 登陆页面。



5. Specify Service Account，可以使用普通的 AD user 账号，也可以是 AD admin 账号。



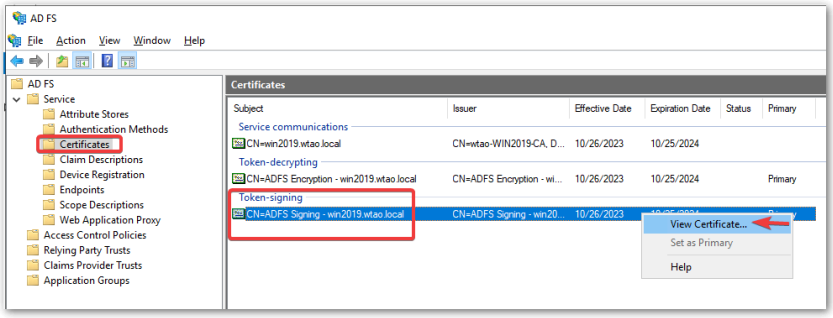
6. Specify Database，创建一个数据库即可。然后 Next，完成配置。



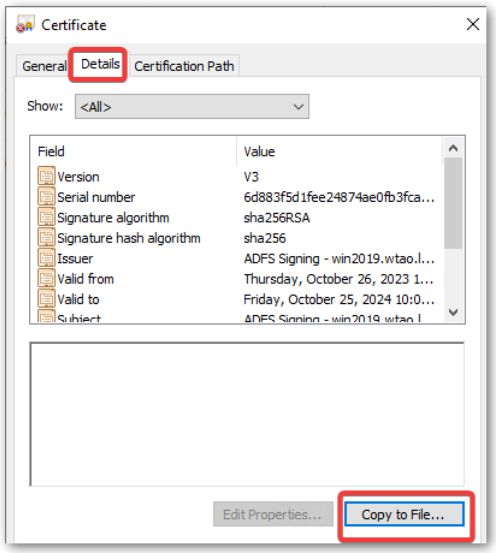
配置 FortiGate 作为 SAML SP

导入 AD FS 的证书

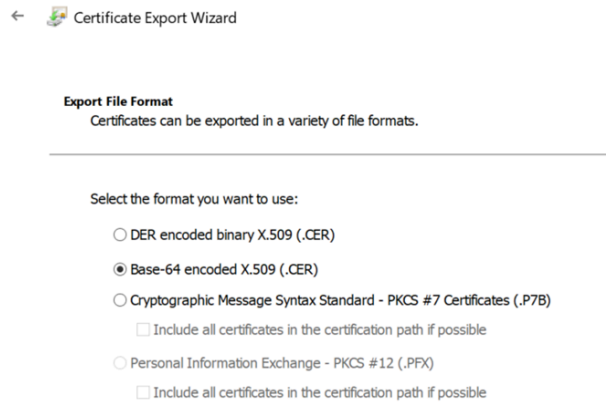
1. 进入 Windows AD FS Management，点击 Service > Certificates，右键点击 Token-signing 的证书，点击 View Certificate…



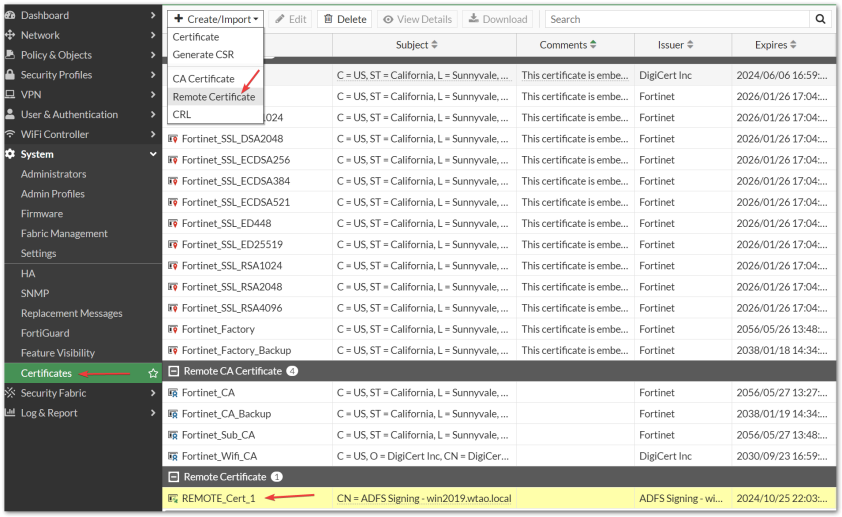
2. 在 Details 标签页，点击 Copy to File…



3. 以 Base-64 的格式保存。



4. 然后将证书以 Remote Certificate 的方式导入到防火墙。



配置 user saml，输入 SP 和 IdP URL

1. 现在我们可以在 user saml 中配置 SP 和 IdP 的 URL。

```bash
config user saml
    edit "saml_user"
        set cert "Fortinet_CA_SSL"
        set entity-id "https://fgt.wtao.local:7831/XX/YY/ZZ/saml/metadata/"
        set single-sign-on-url "https://fgt.wtao.local:7831/XX/YY/ZZ/saml/login/"
        set single-logout-url "https://fgt.wtao.local:7831/XX/YY/ZZ/saml/logout/"
        set idp-entity-id "http://win2019.wtao.local/adfs/services/trust"
        set idp-single-sign-on-url "https://win2019.wtao.local/adfs/ls/"
        set idp-single-logout-url "https://win2019.wtao.local/adfs/ls/"
        set idp-cert "REMOTE_Cert_1"
        set digest-method sha1
        set adfs-claim enable
        set user-claim-type name
        set group-claim-type group
    next
end
```

2. 配置解释：

- entity-id, single-sign-on-url, single-logout-url 中写的是 FortiGate 的 SP URL，这是由需要用到 SAML SSO 的服务预定义的。例如：
  - explicit proxy 需要 SAML SSO 时，这 3 项是：

```jsx
# 10.10.186.1是电脑访问的，位于防火墙上的提供explicit proxy的接口IP地址。
# 它也可以是FQDN - 用户电脑、防火墙(SP)和AD FS(IdP)需要能够解析到相同的IP地址。
# 7831是固定端口
set entity-id "https://10.10.186.1:7831/XX/YY/ZZ/saml/metadata/"
set single-sign-on-url "https://10.10.186.1:7831/XX/YY/ZZ/saml/login/"
```

```
set single-logout-url "https://10.10.186.1:7831//XX/YY/ZZ/saml/logout/"
```

- captive portal 需要 SAML SSO 时，这 3 项是

```jsx
# 10.10.186.1是防火墙上进行captive portal验证的接口IP地址，1003是固定端口。
set entity-id "https://10.10.186.1:1003/remote/saml/metadata/"
set single-sign-on-url "https://10.10.186.1:1003/remote/saml/login/"
set single-logout-url "https://10.10.186.1:1003/remote/saml/logout/"
```

- SSL VPN 需要 SAML SSO 时，这 3 项是

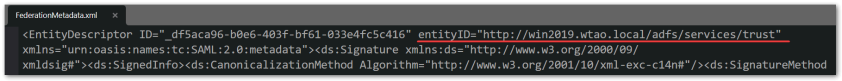
```jsx
# sslvpn.mycompany.com 是用户访问的SSL VPN地址，10443是SSL VPN服务端口
set entity-id "https://sslvpn.mycompany.com:10443/remote/saml/metadata/"
set single-sign-on-url "https://sslvpn.mycompany.com:10443/remote/saml/login/"
set single-logout-url "https://sslvpn.mycompany.com:10443/remote/saml/logout/"
```

- idp-entity-id, idp-single-sign-on-url, idp-single-logout-url 写的是 AD FS 提供 IdP 服务的 URL，这 3 个 URL 在 AD FS 安装后是固定的。所有使用 AD FS 的 SP 都是用相同的这 3 个 URL。

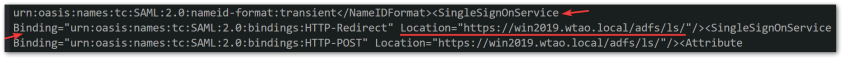
> **重要**
>
> 防火墙和用户电脑都必须能够解析 AD FS 的主机名。

```bash
set idp-entity-id "http://win2019.wtao.local/adfs/services/trust"
set idp-single-sign-on-url "https://win2019.wtao.local/adfs/ls/"
set idp-single-logout-url "https://win2019.wtao.local/adfs/ls/"
```

- 他们可以在 AD FS 的 metadata file 中找到（参考文档 FortiGate SSL VPN Authentication with AD FS ↗），AD FS 的 metadata file 可以在 https://win2019.wtao.local/FederationMetadata/2007-06/FederationMetadata.xml ↗ 打开。
  - *idp-entity-id*: 搜索 'entityID'。



  - *idp-single-sign-on-url*: 搜索 'SingleSignOnService'，找到 'Binding'属性旁边的'Location' 属性。



  - *idp-single-logout-url*: 搜索 'SingleLogoutService'，找到 'Binding'属性旁边的'Location' 属性。

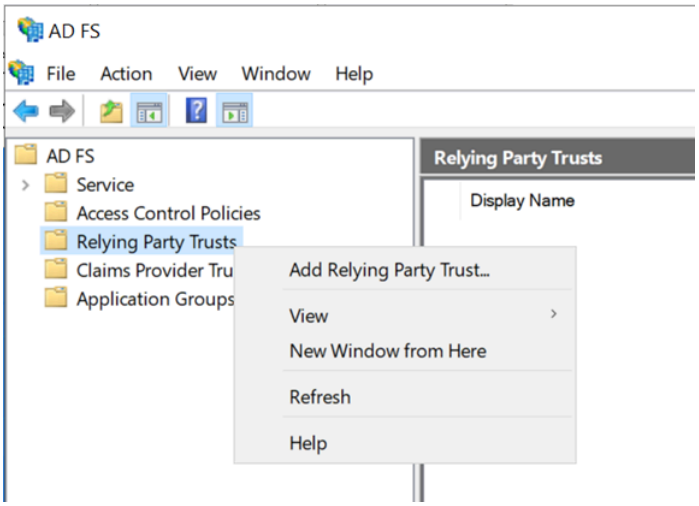


配置 AD FS 作为 SAML IdP

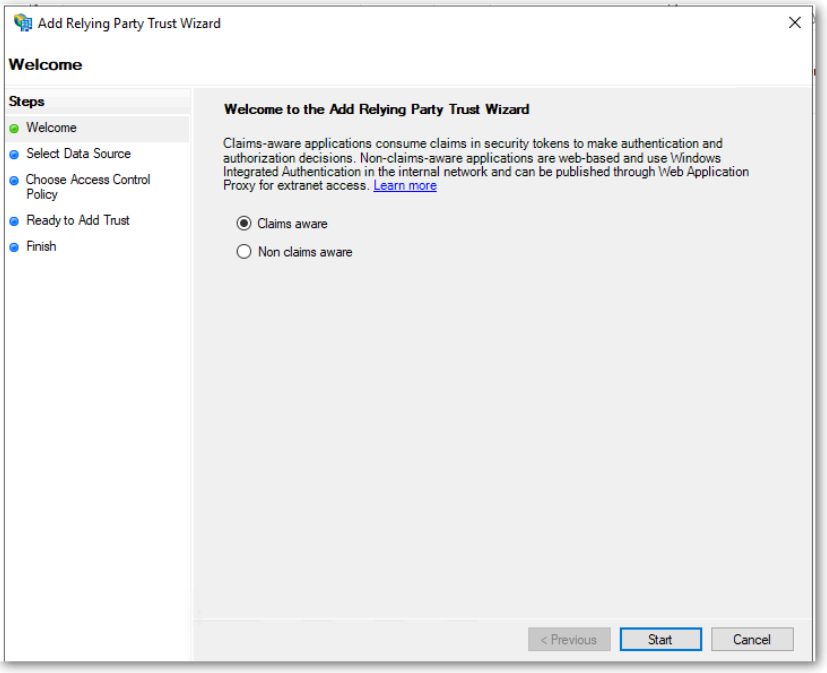
AD FS 上的 Relying Party Trust 定义了 AD FS 为哪个 SP 提供服务，当我们把防火墙配置成为 SAML SP 后，需要把那 3 个 SP URL 写入到 AD FS 的 Relying Party Trust。

添加 Relying Party Trust

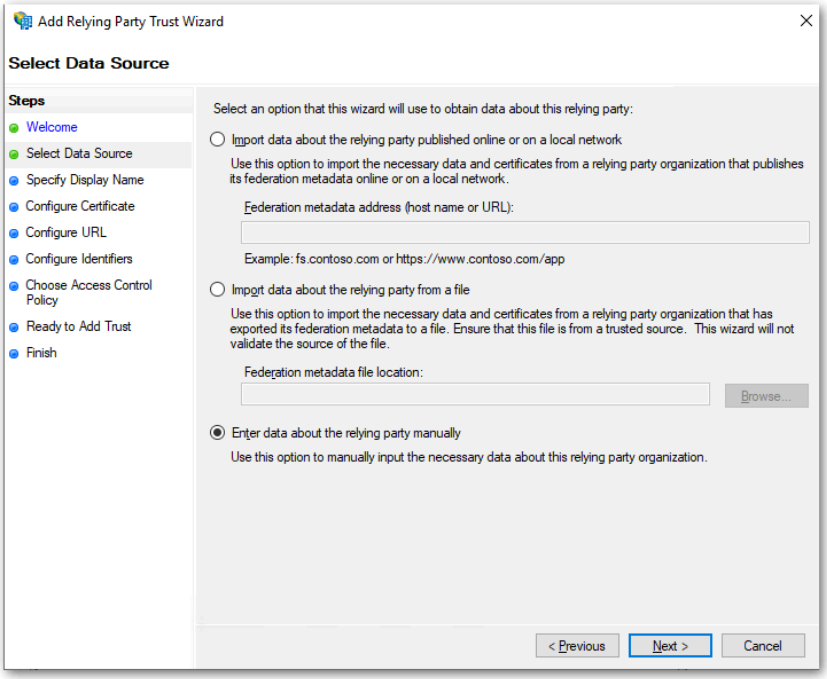
1. 进入 AD FS Management。

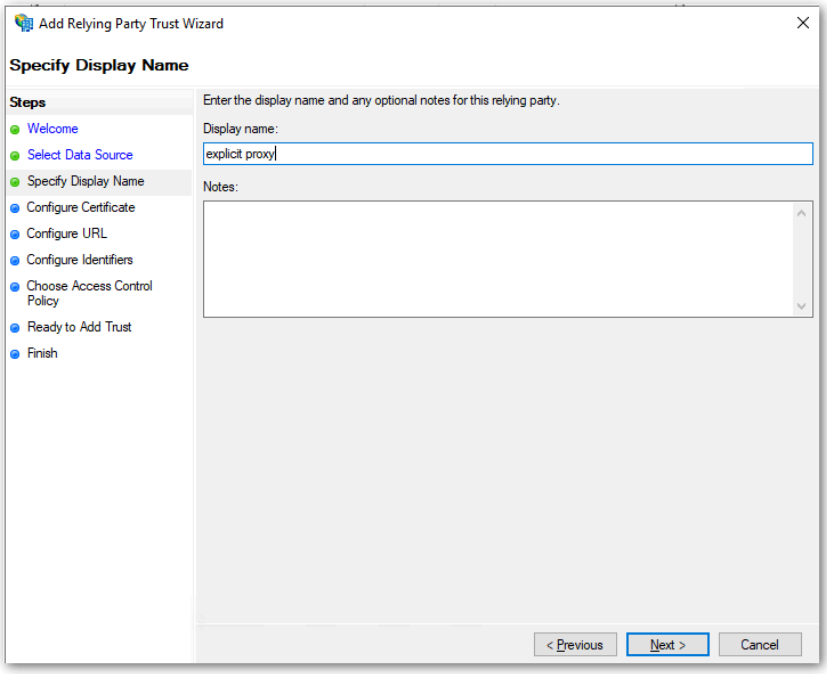


2. select 'Claims Aware' and click 'Start’ 。

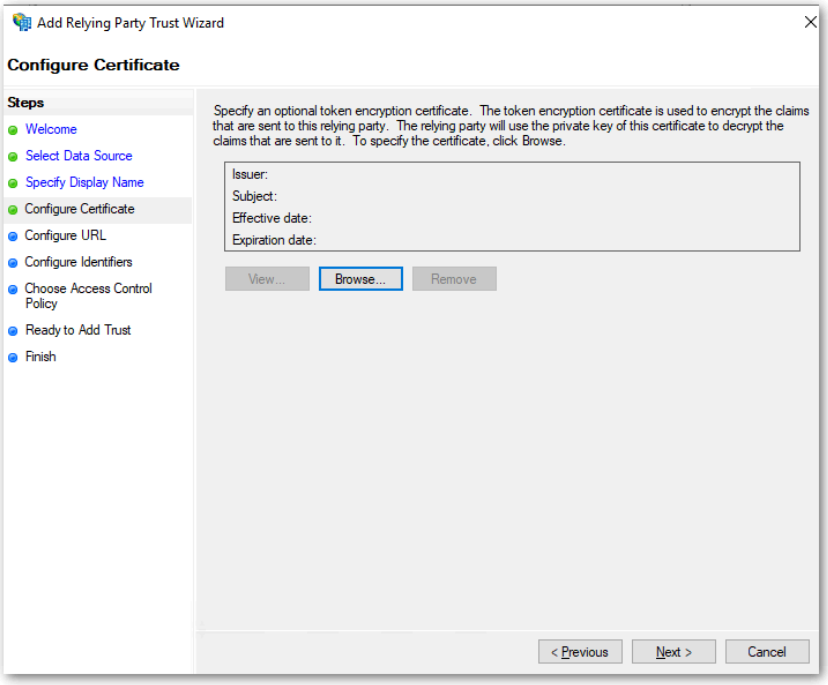


3. 选择“Enter data about the relying party manually”。

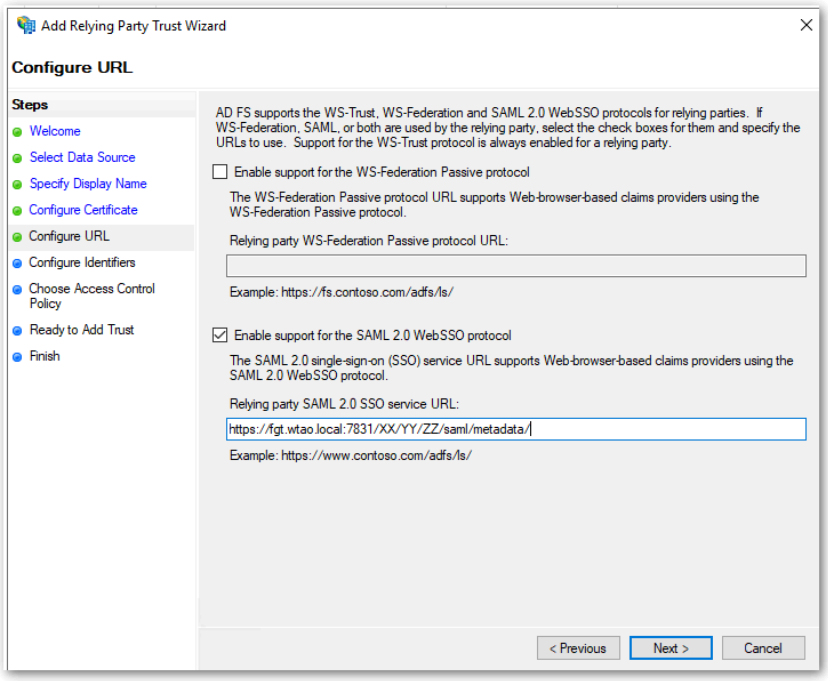

Add Relying Party Trust Wizard
Select Data Source
Steps
Welcome
Select Data Source
Specify Display Name
Configure Certificate
Configure URL
Configure Identifiers
Choose Access Control Policy
Ready to Add Trust
Finish
Select an option that this wizard will use to obtain data about this relying party:
Import data about the relying party published online or on a local network
Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.
Federation metadata address (host name or URL):
Example: fs.contoso.com or https://www.contoso.com/app
Import data about the relying party from a file
Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.
Federation metadata file location:
Browse...
Enter data about the relying party manually
Use this option to manually input the necessary data about this relying party organization.
< Previous
Next >
Cancel


Add Relying Party Trust Wizard
Specify Display Name
Steps
Welcome
Select Data Source
Specify Display Name
Configure Certificate
Configure URL
Configure Identifiers
Choose Access Control Policy
Ready to Add Trust
Finish
Enter the display name and any optional notes for this relying party.
Display name:
explicit proxy
Notes:
< Previous
Next >
Cancel

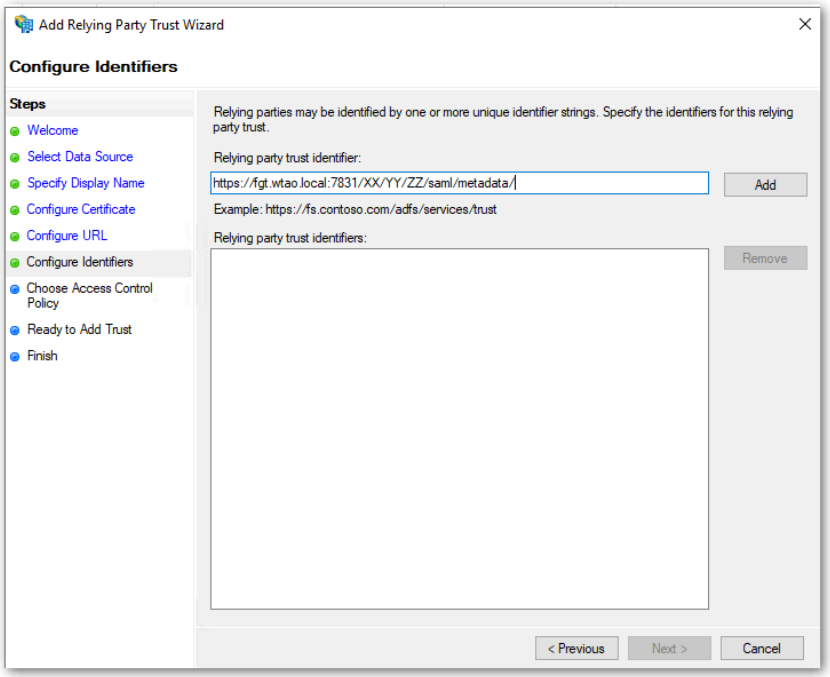
4. 保持默认，Next。



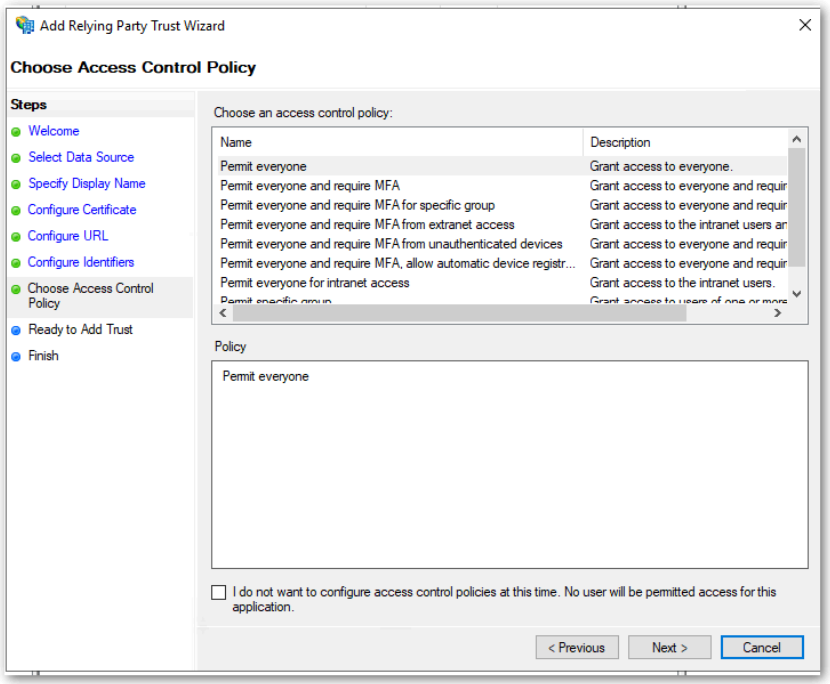
5. Configure URL, 选择” Enable support for the SAML 2.0 WebSSO protocol”，输入防火墙上 user saml 中的 entity-id, `https://fgt.wtao.local:7831/XX/YY/ZZ/saml/metadata/`。



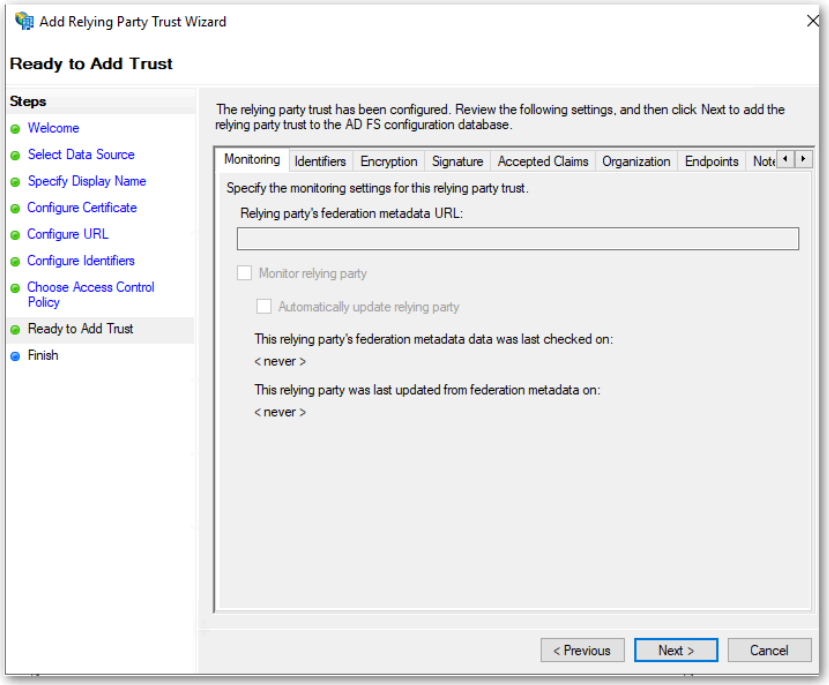
6. Configure Identifiers 中，还是输入防火墙上 user saml 中的 entity-id：`https://fgt.wtao.local:7831/XX/YY/ZZ/saml/metadata/`，点击 Add，再点击 Next。



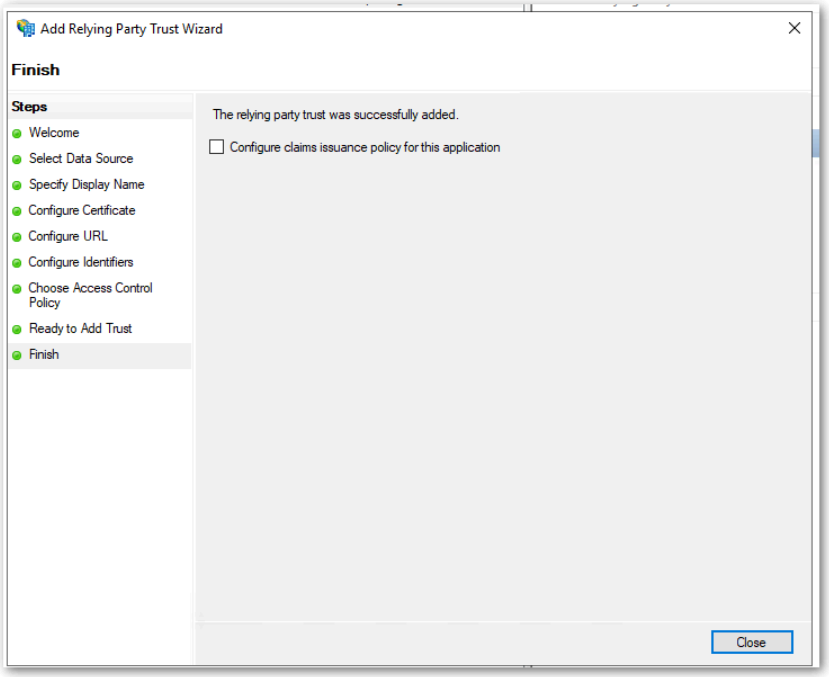
7. Choose Access Control Policy 选择 Permit everyone。



8. Ready to Add Trust，先不填写，Next。

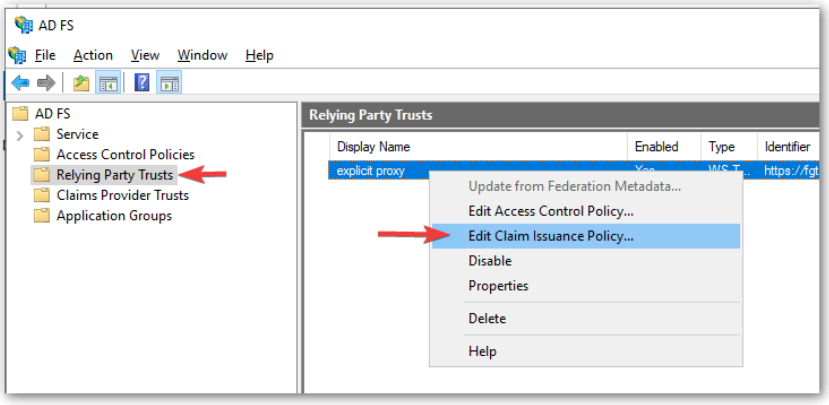


9. 取消“Configure claims issuance policy for this application”，点击 Close。

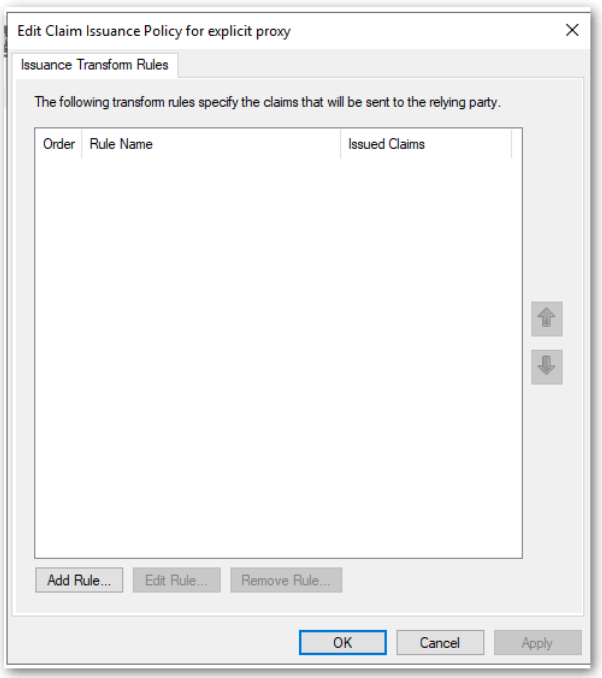


配置 Relying Party Trust > Claim Issuance Policy

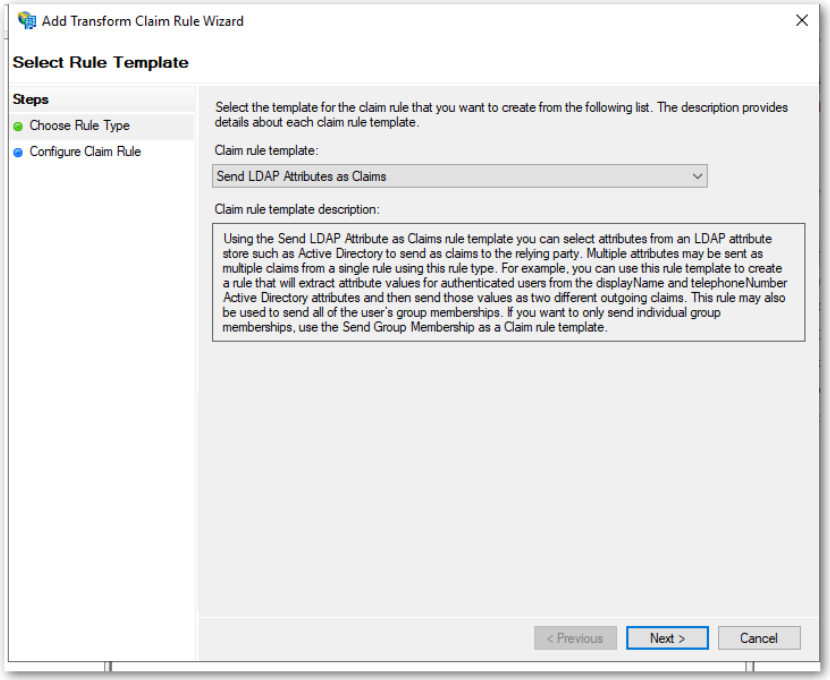
1. 在 Relying Party Trust，右键点击刚才创建的 relying party trust，点击“Edit Claim Issuance Policy”。



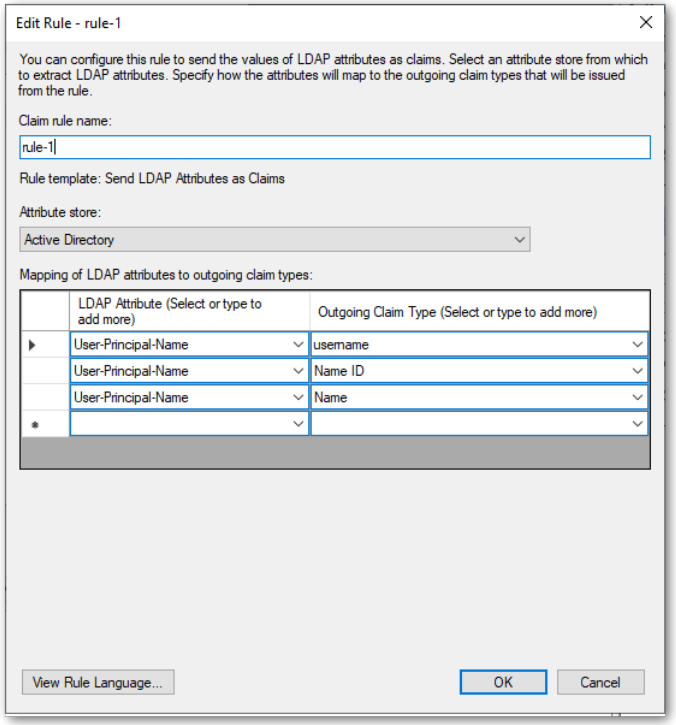
2. click on 'Add Rule'。



3. Claim rule template: Send LDAP Attributes as Claims。



4. Configure Claim Rule：



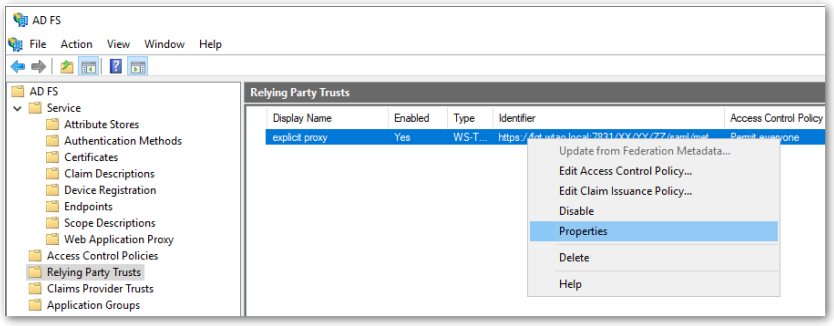
- Claim rule name: 按需取名。
- Attribute store: Active Directory。
- Mapping of LDAP attributes to outgoing claim types:
  - User-Principal-Name: username (手动输入)
  - User-Principal-Name: Name ID （有预选项）
  - User-Principal-Name: Name（有预选项）

5. 点击 Finish 保存。

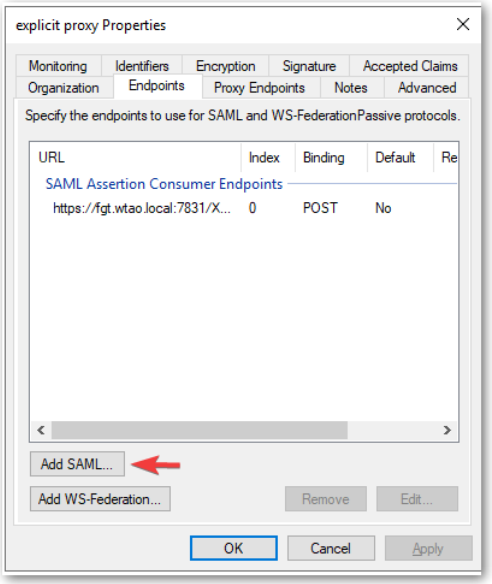
配置 Relying Party Trust > SP URL

1. 在 Relying Party Trust 中将防火墙的 SP URL 输入。

2. 在 Relying Party Trust，右键点击刚才创建的 relying party trust，点击” Properties”。

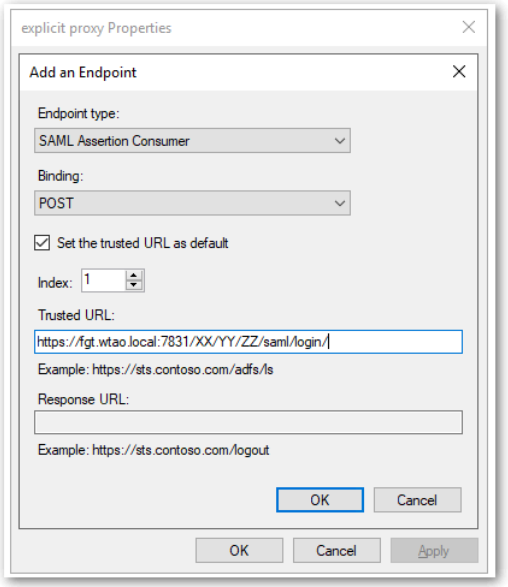


3. Endpoint 标签页, 点击 Add SAML…

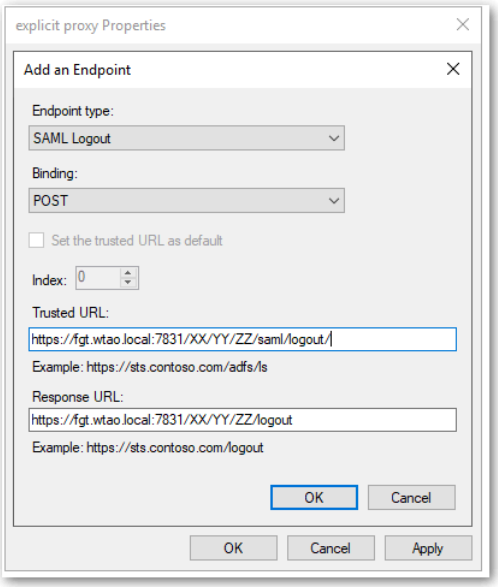


- Endpoint type: SAML Assertion Consumer
- Binding: POST
- Set the trusted URL as default: 选中
- Index: 1
- Trusted URL: 输入防火墙上 user saml 中的 single-sign-on-url, https://fgt.wtao.local:7831/XX/YY/ZZ/saml/login/ ↗

4. 点击 OK。

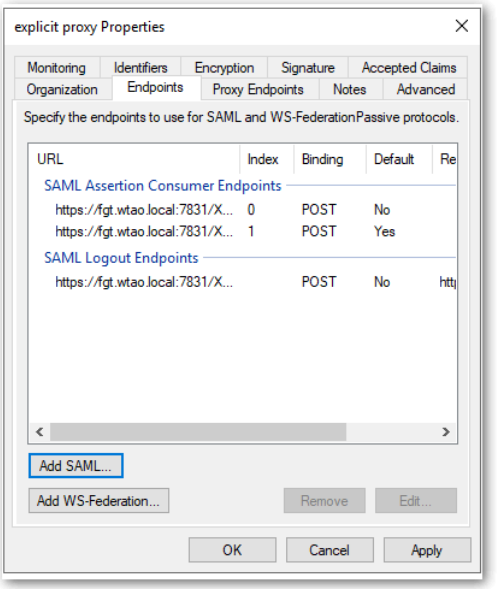


5. 再次点击 Add SAML…



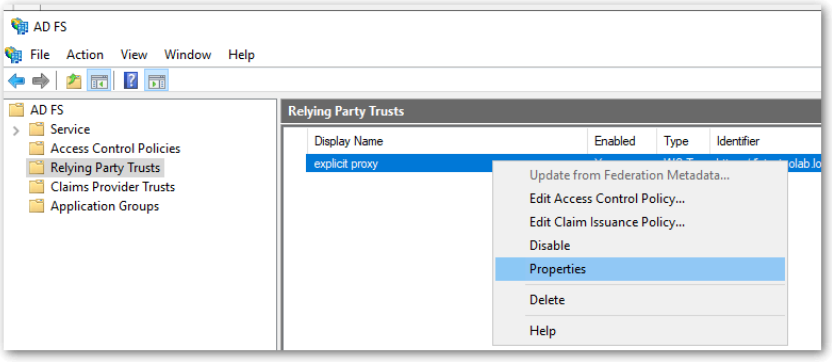
- Endpoint type: SAML Logout
- Binding: POST
- Trusted URL: 输入防火墙上 user saml 中的 single-logout-url, https://fgt.wtao.local:7831/XX/YY/ZZ/saml/logout/
- Response URL: 如果防火墙说明文档中没有提供，可以尝试填写 https://fgt.wtao.local:7831/XX/YY/ZZ/logout

6. 点击 OK，这是完成配置的 Endpoitns 标签页。点击 OK。

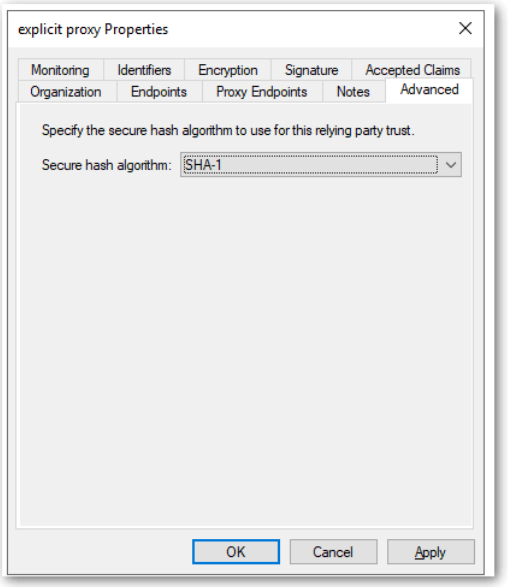


配置 Relying Party Trust >SHA 和防火墙证书

1. 在 Relying Party Trust，右键点击刚才创建的 relying party trust，点击” Properties”。

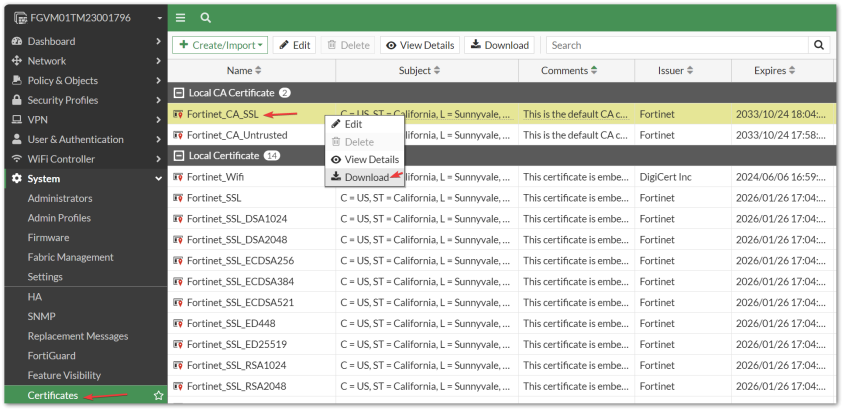


2. 在 Advanced 标签页，将 Secure hash algorithm 设置为 SHA-1，**和防火墙上的 user saml > “set digest-method sha1”保持一致。**

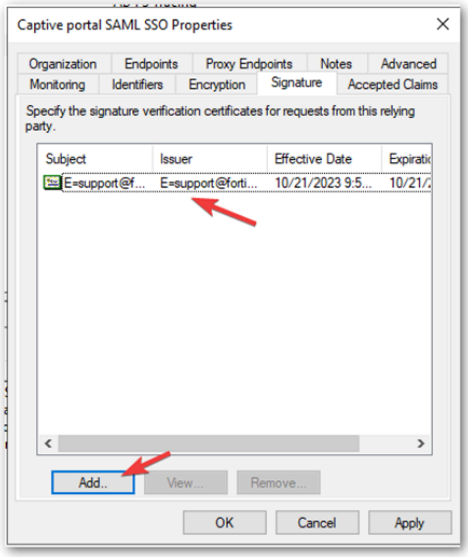


3. explicit proxy 使用 AD FS 作为 SAML SSO 时，**AD FS 会验证防火墙 SP URL 的证书**，因此需要将防火墙上的 Forinet_CA_SSL 导入 AD FS。

4. 在防火墙上下载 Forinet_CA_SSL。



5. 在 Signature 标签页，导入 Forinet_CA_SSL。



配置 FortiGate 启用 explicit proxy 和 captive portal

1. 启用 explicity proxy 和 captive portal。在内网口 port2 上启用 explicit-web-proxy 和 proxy-captive-portal。

```bash
# Enable the web proxy:
config system settings
  set gui-explicit-proxy enable
end

config web-proxy explicit
    set status enable
    set http-incoming-port 8080
    set unknown-http-version best-effort
end

# Enable the proxy captive portal:
config system interface
    edit "port2"
        set explicit-web-proxy enable
        set proxy-captive-portal enable
    next
end
```

2. 配置 LDAP 服务器和 LDAP 用户组。

```bash
# Configure the LDAP server
config user ldap
    edit "ldap-10.10.186.5"
        set server "10.10.186.5"
        set cnid "userPrincipalName"
        set dn "dc=wtao,dc=local"
        set type regular
        set username "cn=administrator,cn=users,dc=wtao,dc=local"
        set password password_of_the_user
        set group-search-base "cn=users,dc=wtao,dc=local"
    next
end

config user group
    edit "ldap-group-saml"
        set member "ldap-10.10.186.5"
    next
end
```

3. 配置 authentication scheme, rule 和 setting。

```bash
config authentication scheme
    edit "auth_scheme_saml"
        set method saml
        set saml-server "saml_user"
        set user-database "ldap-10.10.186.5"
    next
end

config authentication rule
    edit "auth_rule_saml"
        set srcaddr "all"
        set active-auth-method "auth_scheme_saml"
    next
end

config firewall address
    edit "fgt.wtao.local"
      set type fqdn
      set fqdn "fgt.wtao.local"
end
```

```
config authentication setting
    set active-auth-scheme "auth_scheme_saml"
    set captive-portal "fgt.wtao.local"
end
```

4. 配置 proxy-policy。proxy-policy #2，访问 Windows AD FS 服务器时，不需要用户名；proxy-policy #1，访问外网时，需要用户名，触发 SAML SSO 验证。

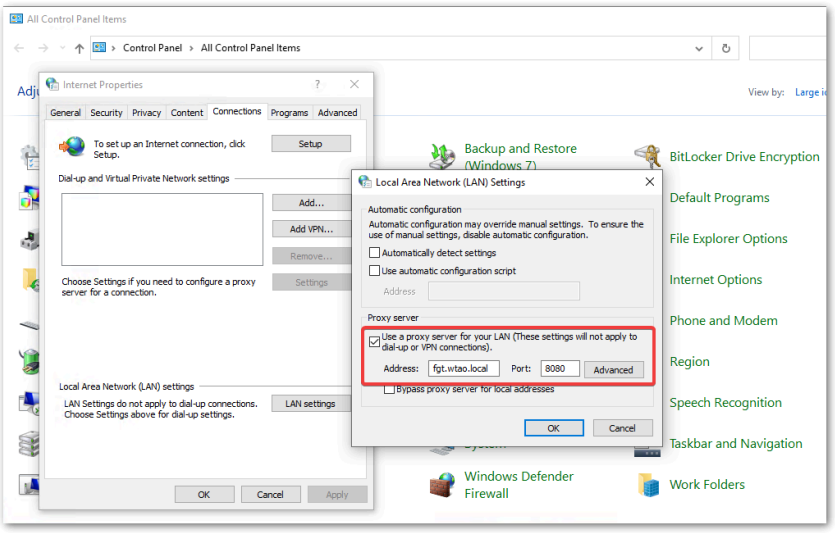
```bash
config firewall address
    edit "win2019.wtao.local_IP"
        set type iprange
        set start-ip 10.10.186.5
        set end-ip 10.10.186.5
    next
end
config firewall proxy-policy
    edit 2
        set name "bypass"
        set proxy explicit-web
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "win2019.wtao.local_IP"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set ssl-ssh-profile "certificate-inspection"
        set comment "no user identity is required for traffic to AD FS"
    next
    edit 1
        set name "proxy"
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "ldap-group-saml"
        set ssl-ssh-profile "certificate-inspection"
    next
end
```

5. 配置防火墙策略。firewall policy#1，访问外网时，需要用户名。

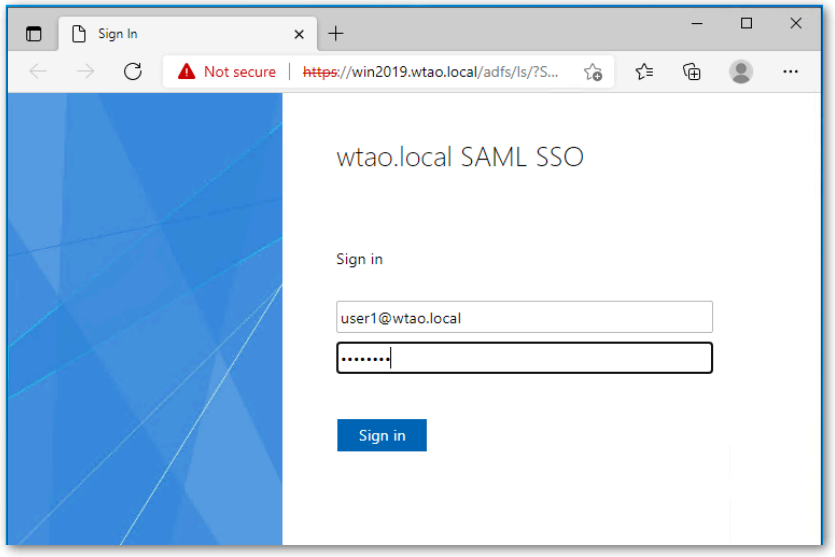
```bash
config firewall policy
    edit 1
        set name "LANtoWAN"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set groups "ldap-group-saml"
    next
end
```

电脑上设置使用防火墙作为上网代理

1. 在 Windows AD 电脑上设置使用 FortiGate 的 LAN port2 作为上网代理服务器。

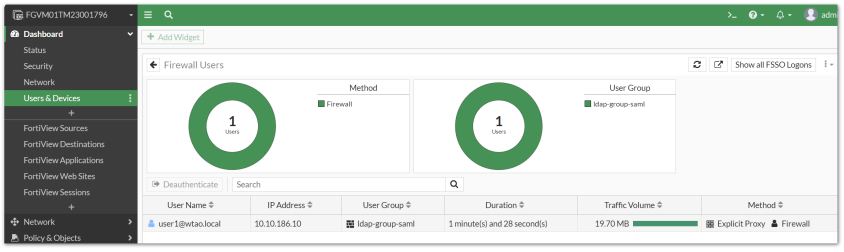


2. 当 Windows AD 电脑访问公网网站时，会被重定向到 AD FS SAML 登陆页面。登陆页面的标题文字是在” 1.7 初始化 AD FS”中设置的。用户名的格式是 UPN (User Principal Name)。



结果验证

1. GUI > Dashboard > Users & Devices > Firewall Users 显示登陆的用户。



2. diag wad user list 会显示用户详情。

```
FGVM01TM23001796 # diag wad user list

ID: 2, VDOM: root, IPv4: 10.10.186.10
  user name    : user1@wtao.local
  worker       : 0
  duration     : 70
  auth_type    : IP
  auth_method  : SAML
  pol_id       : 1
```

```
  g_id        : 2
  user_based  : 0
  expire      : no
  LAN:
    bytes_in=319243 bytes_out=17951352
  WAN:
    bytes_in=17925608 bytes_out=277637

FGVM01TM23001796 #
```

故障排查

下面的错误表明在步骤“配置 Relying Party Trust > Claim Issuance Policy”时，缺少 User-Principal-Name: Name 的映射。加上后，问题可以解决。

```bash
Authentication failed.
user.saml.saml_user.user-name='http://schemas.xmlsoap.org/ws/2005/05/identity/claims/**name**' was NOT FOUND in the SAML
Assertion Response.
Please check with the Administrator.
```

认证常见问题

- 认证与策略顺序的关系
- Web 认证弹出 2 个页面
- 升级版本后 Radius 连接失败

认证与策略顺序的关系

网络需求

1. 源地址为 192.168.100.77/32、192.168.100.78/32 的 PC 需要经过防火墙认证才能访问 Internet。
2. 其他源 IP 不需要防火墙认证就可以直接访问 Internet。

配置步骤

1. 在 FortiGate 上配置防火墙认证的用户与用户组。

```bash
config user local
    edit "user1"
        set type password
        set passwd xxxxxx
    next
    edit "user2"
        set type password
        set passwd xxxxxx
    next
end

config user group
    edit "Auth_Users"
        set member "user1" "user2"
    next
end
```

2. 创建第 1 条上网策略，源 IP 为 192.168.100.77/32、192.168.100.78/32，用户选择用户组 `Auth_Users`，目标为 all，开启 SNAT。



3. 在步骤 2 的策略下创建第 2 条上网策略，源 IP 为 192.168.100.0/24，不选择用户，目标为 all，开启 SNAT。

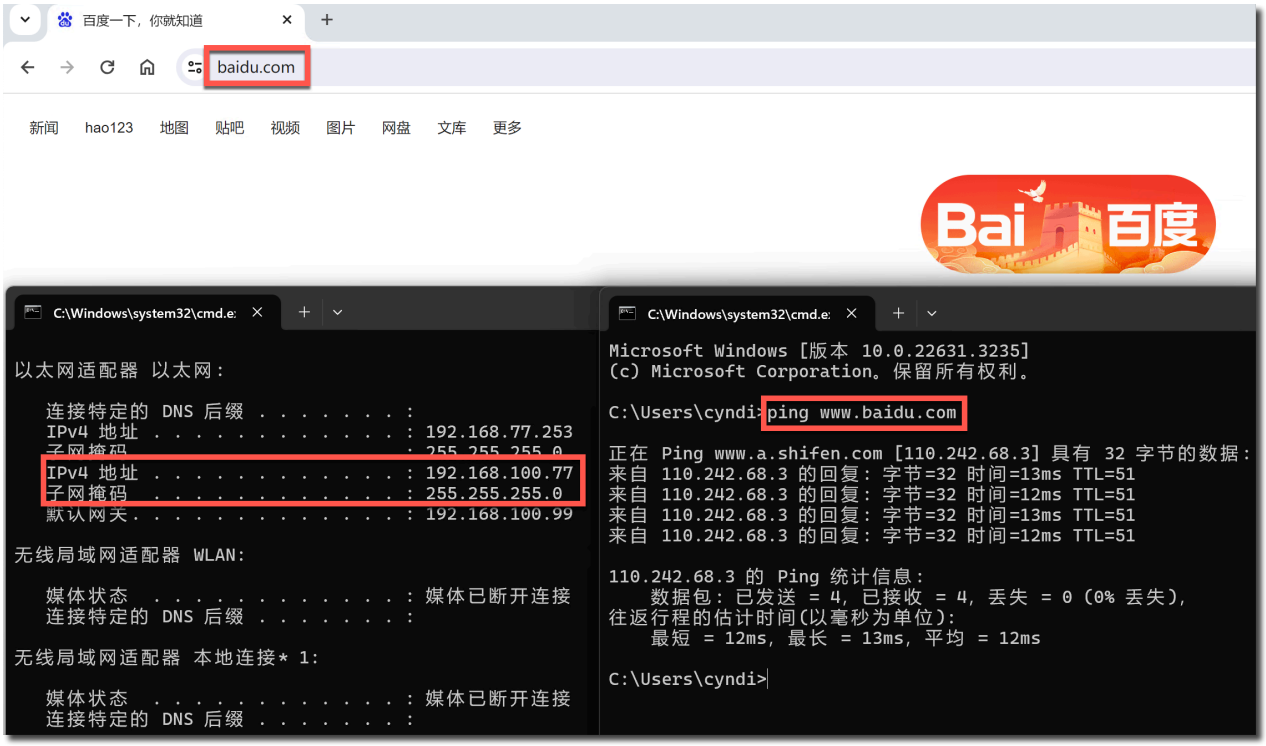





```bash
config firewall policy
    edit 17
        set srcintf "internal5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "192.168.100.77/32" "192.168.100.78/32"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "Auth_Users"
    next
    edit 1
        set srcintf "internal5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "192.168.100.0/24"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

结果验证

1. 192.168.100.77/32、192.168.100.78/32 与 192.168.100.0/24 的其他 PC 都不需要防火墙认证，可以直接访问 Internet，这显然与用户的需求是不一致的。



2. 对应会话显示匹配的策略为 ID 1（配置的第 2 条其他网段的上网策略），没有认证标记。

```bash
session info: proto=6 proto_state=01 duration=2 expire=3597 timeout=3600 flags=00000000 socktype=0 sockport=0 av_idx=0
use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu synced f00    //未包含auth标记//
statistic(bytes/packets/allow_err): org=92/2/1 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin→sink: org pre→post, reply pre→post dev=12→5/5→12 gwy=172.22.6.1/10.10.12.2
hook=post dir=org act=snat 192.168.100.77:18489→110.242.68.4:443(172.22.6.128:40598)
hook=pre dir=reply act=dnat 110.242.68.4:443→172.22.6.128:40598(192.168.100.77:18489)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=e0:23:ff:67:e3:9c
misc=0 policy_id=1 //匹配策略ID 17// pol_uuid_idx=636 auth_info=0 chk_client_info=0 vd=0
serial=013c33e2 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=3 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0×4000c00 ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=0/0, epid=64/71, ipid=71/64, vlan=0×0000/0×0000
vlifid=71/64, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=1/3
```

解决办法

1. 默认配置下，**未认证的流量**经过 FortiGate 时，允许匹配开启认证的策略（ID 17）后的防火墙策略，由于开启认证的策略（ID 17）后的策略（ID 1）为整个网段的上网策略，且未开启认证，所以需要认证的流量就可以匹配策略 ID 1 直接访问 Internet。

2. 默认情况，该特性配置如下，`auth-on-demand` 的模式为 `implicitly`。

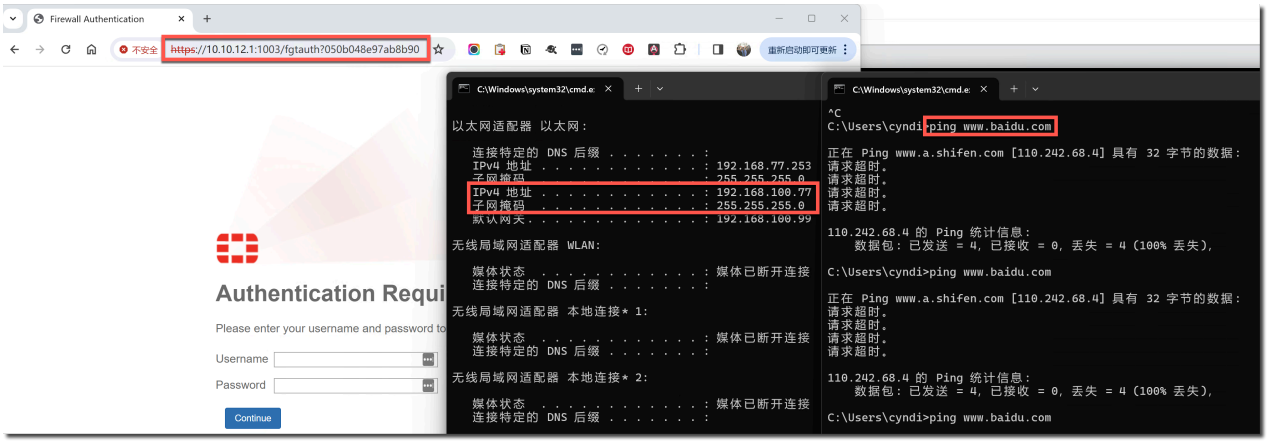
```bash
FortiGate # config user setting
FortiGate (setting) # set auth-on-demand ?
always        Always trigger firewall authentication on demand.
implicitly    Implicitly trigger firewall authentication on demand.
```

- `always`：按照策略配置的顺序来触发防火墙身份验证，如果 **未认证流量**已经匹配上认证策略，则 **未认证流量**不会匹配认证策略后的其他策略（需要强制认证，认证通过后，**已认证流量**可以匹配认证策略后的其他策略），如果 **未认证流量**没有匹配上开启认证的策略，则会按照顺序继续匹配认证策略后的其他策略。
- `implicitly`：默认配置。隐式触发防火墙身份验证，**未认证的流量**也可以匹配认证策略下的其他策略，如果认证策略后有其他可以匹配的策略，则没有经过认证也可以匹配其他策略放通。

3. 想要实现此用户的需求，需要将 `auth-on-demand` 的模式修改为 `always`，表示按照策略配置的顺序来触发防火墙身份验证：

```
config user setting
    set auth-on-demand always
end
```

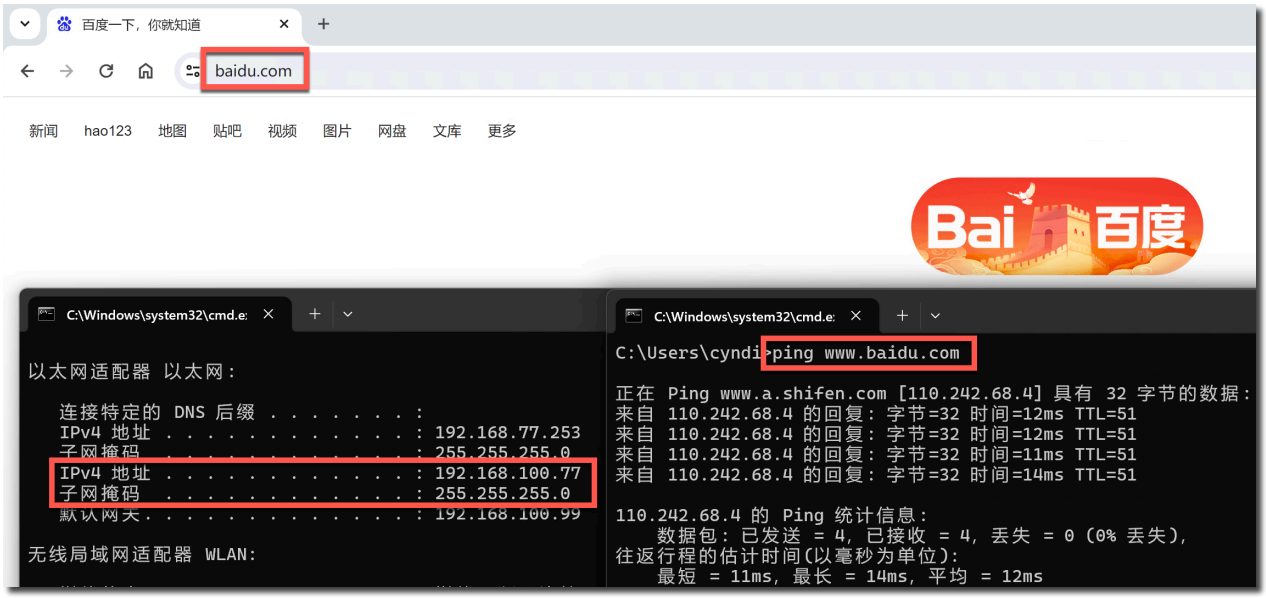
4. 再次使用 192.168.100.77/32 访问 Internet，弹出了认证页面。



5. 查看会话需要认证的流量匹配了策略 ID 17，也就是配置的第 1 条认证策略，会话中有 `auth` 标记。

```
session info: proto=6 proto_state=05 duration=0 expire=0 timeout=3600 flags=00000004 socktype=4 sockport=10100 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=redir log local may_dirty auth npu synced f00    //包含auth标记//
statistic(bytes/packets/allow_err): org=0/0/0 reply=6945/9/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=12->5/5->12 gwy=172.22.6.1/0.0.0.0
hook=pre dir=org act=noop 192.168.100.77:19011->110.242.68.4:443(0.0.0.0:0)
hook=post dir=reply act=noop 110.242.68.4:443->192.168.100.77:19011(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=e0:23:ff:67:e3:9c
misc=0 policy_id=17 //匹配策略ID 17// pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=013d220f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=3 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  redir-to-av auth    //提示重定向到认证模块//
```

6. 认证成功后，192.168.100.77/32 可以正常访问 Internet。



7. 会话状态的 auth 标记变为 authed，匹配的策略 ID 仍为认证策略 17。

```
session info: proto=6 proto_state=01 duration=3 expire=3596 timeout=3600 flags=00000000 socktype=0 sockport=0 av_idx=0 bash
use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
user=user1 state=log may_dirty npu authed synced f00 acct-ext    //包含authed标记//
statistic(bytes/packets/allow_err): org=92/2/1 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=12->5/5->12 gwy=172.22.6.1/10.10.12.2
hook=post dir=org act=snat 192.168.100.77:19231->110.242.68.4:443(172.22.6.128:19231)
hook=pre dir=reply act=dnat 110.242.68.4:443->172.22.6.128:19231(192.168.100.77:19231)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=e0:23:ff:67:e3:9c
misc=0 policy_id=17 //匹配策略ID 17// pol_uuid_idx=643 auth_info=3 chk_client_info=0 vd=0
serial=013d776b tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=3 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=64/71, ipid=71/64, vlan=0x0000/0x0000
vlifid=71/64, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/2
```

Web 认证弹出 2 个页面

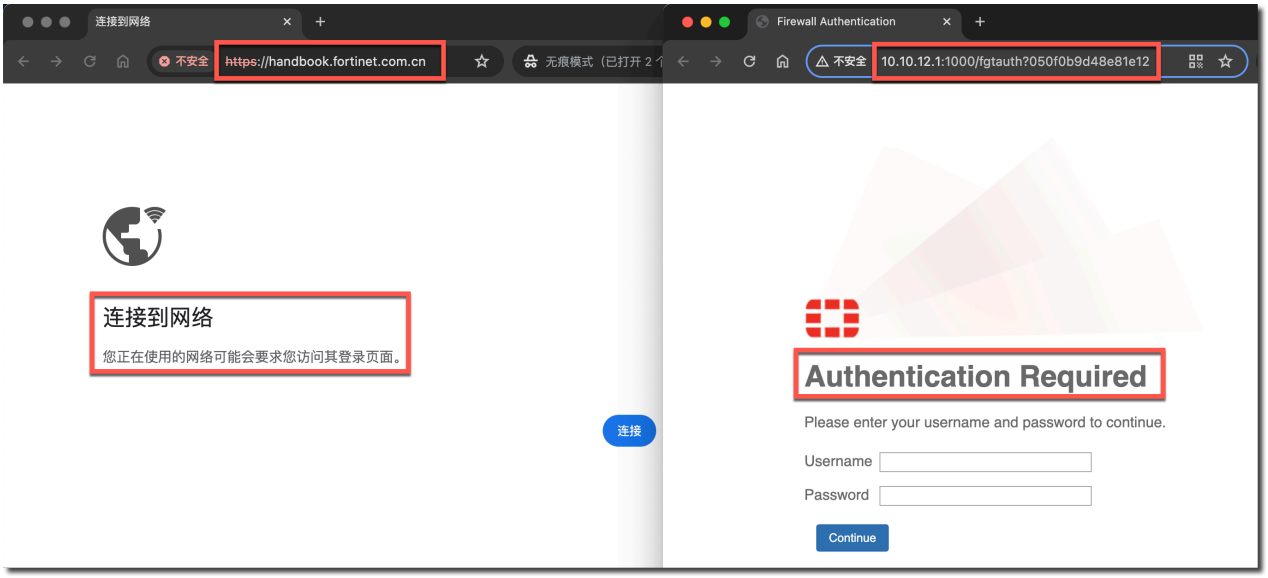
问题现象

1. 用户配置了防火墙 Web 认证，用户与用户组配置如下，认证相关配置（`config user setting`）均为默认配置，使用上网的防火墙策略引用用户组。

```bash
config user local
    edit "user1"
        set type password
        set passwd xxxxxx
    next
end

config user group
    edit "Auth_Users"
        set member "user1"
    next
end

config user setting
end

config firewall policy
    edit 1
        set name "LAN_to_Internet"
        set srcintf "internal5"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "Auth_Users"
    next
end
```

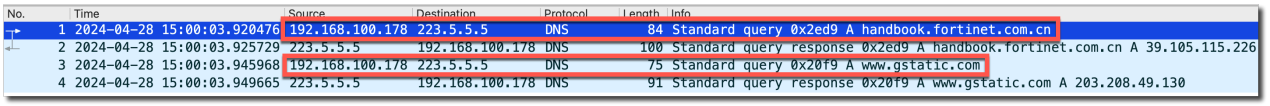
2. 内网 PC 使用浏览器（以 Chrome 为例）以 HTTPS 的 URL 格式访问 Internet 网站（例如 `https://handbook.fortinet.com.cn`）时，会弹出 2 个页面，1 个页面提示“连接到网络”，另一个为防火墙认证页面（理论上只应弹出认证页面）。

> **重要**
>
> 由于浏览器可能有历史记录的原因，即使输入的是 `handbook.fortinet.com.cn`，浏览器也可能会自动替换为 `https://handbook.fortinet.com.cn`。



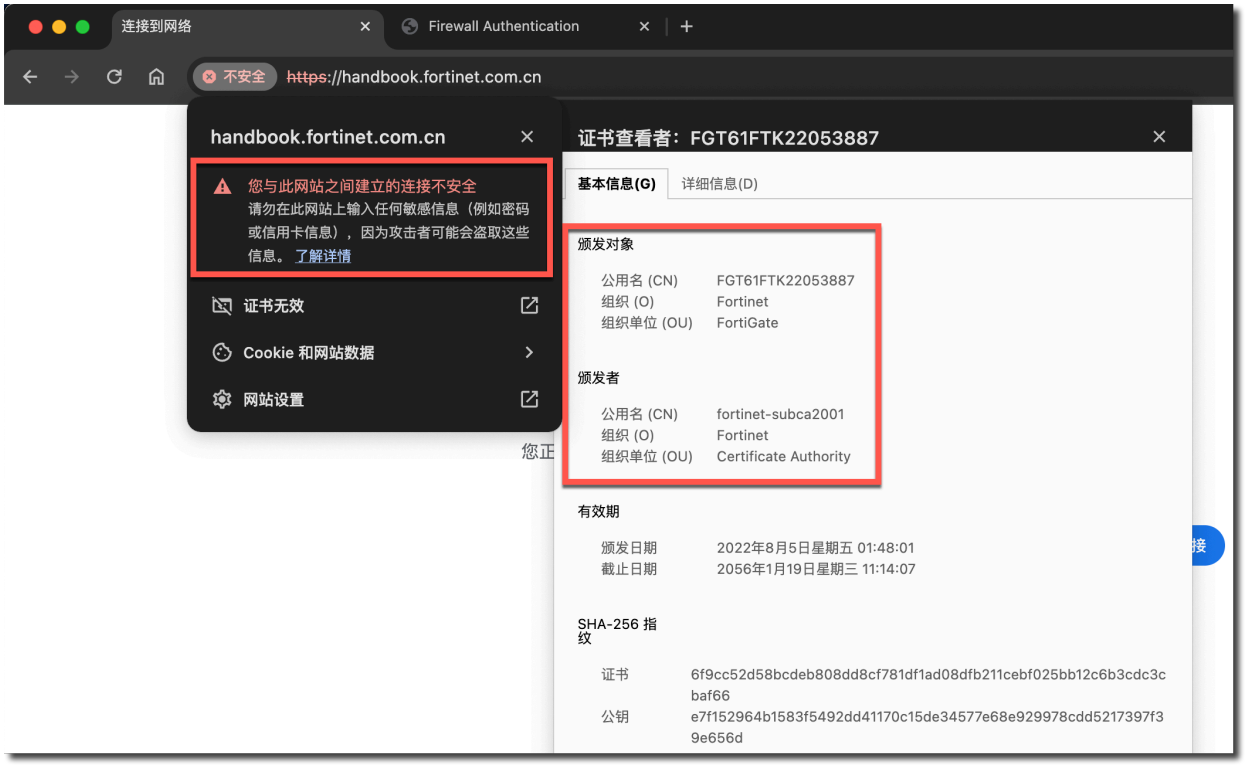
问题原因

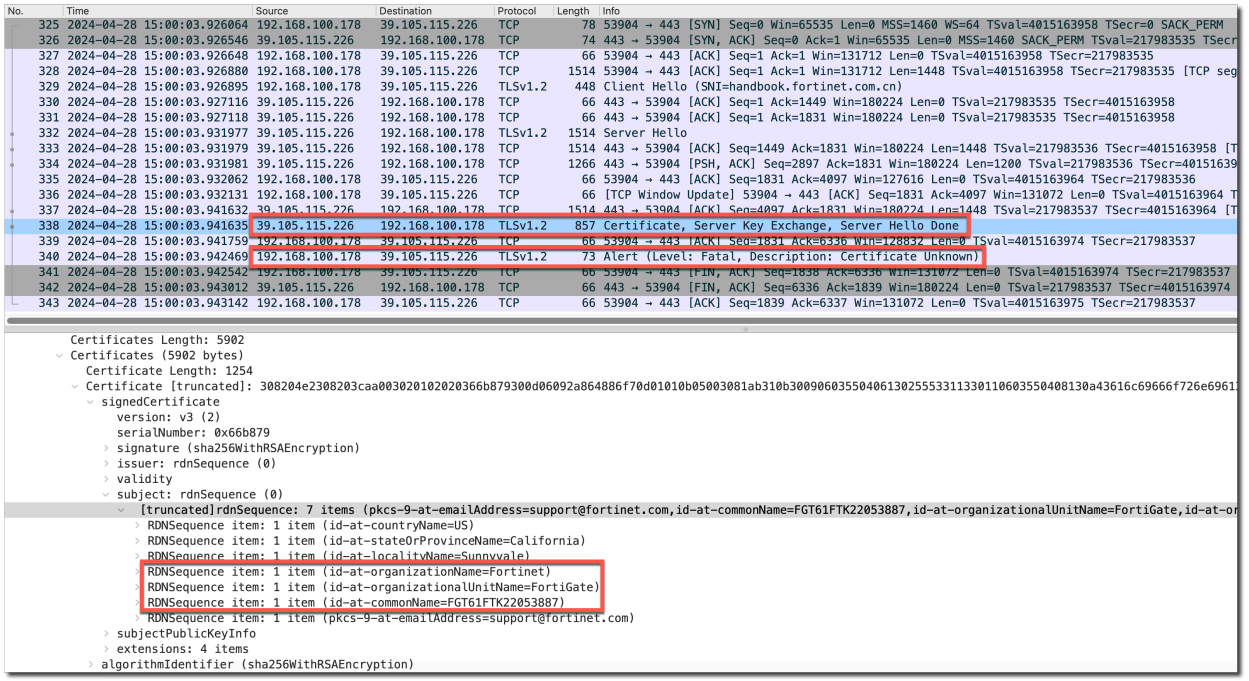
1. 通过抓包发现，PC 浏览器（Chrome）每次访问 `https://handbook.fortinet.com.cn`，不只会请求访问的域名 `handbook.fortinet.com.cn` 的 IP 地址，同时会请求 `www.gstatic.com` 的 IP 地址（抓包文件请参考：auth_dns.pcapng）。



No.	Time	Source	Destination	Protocol	Length	Info
1	2024-04-28 15:00:03.920476	192.168.100.178	223.5.5.5	DNS	84	Standard query 0x2ed9 A handbook.fortinet.com.cn
2	2024-04-28 15:00:03.925729	223.5.5.5	192.168.100.178	DNS	100	Standard query response 0x2ed9 A handbook.fortinet.com.cn A 39.105.115.226
3	2024-04-28 15:00:03.945968	192.168.100.178	223.5.5.5	DNS	75	Standard query 0x20f9 A www.gstatic.com
4	2024-04-28 15:00:03.949665	223.5.5.5	192.168.100.178	DNS	91	Standard query response 0x20f9 A www.gstatic.com A 203.208.49.130

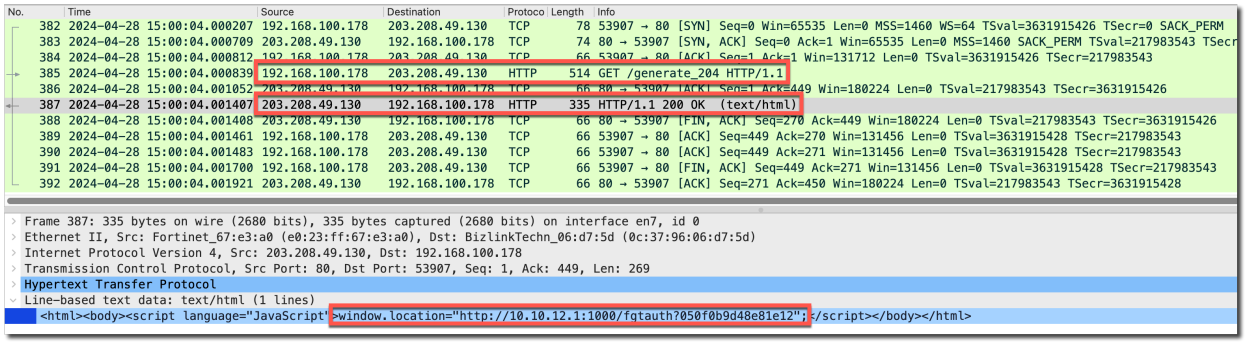
2. 此后 PC 浏览器会分发起 2 个 HTTP 请求：
   - 访问 `https://handbook.fortinet.com.cn`，由于 FortiGate 开启了防火墙认证，FortiGate 会拦截 PC 的对该网站的访问，并使用相同的 HTTPS 返回自签名的证书，默认情况下该证书不受 PC 信任，浏览器向服务器返回了“Certificate Unknown”的告警信息（抓包文件请参考：auth_to_handbook.pcapng））。



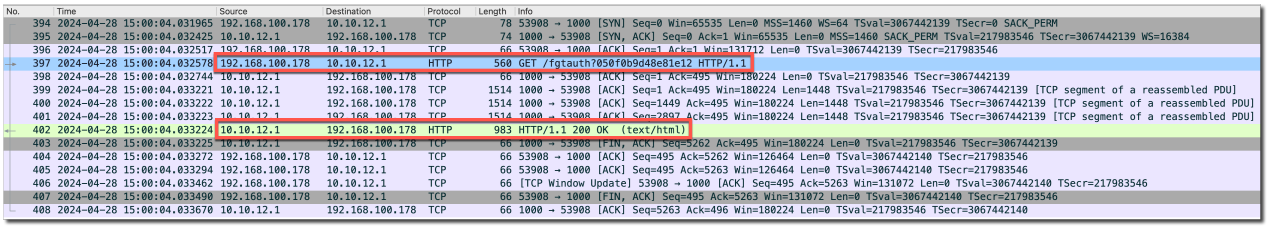


- 访问 `http://www.gstatic.com/generate_204`，由于 FortiGate 开启了防火墙认证，FortiGate 会拦截 PC 的对 `/generate_204` 网页的访问请求，并使用 HTTP 协议返回 HTTP 200 OK，其中携带需要重定向的认证页面（通过 JS 脚本的方式进行重定向，此例中重定向的 URL 为 `http://10.10.12.1:1000/fgtauth?050f0b9d48e81e12`，抓包文件请参考：auth_to_gstatic.pcapng ↗）。

> **重要**
>
> Chrome 浏览器在访问网页时会同时使用 `http://www.gstatic.com/generate_204` 来确认浏览器是否可以访问 Internet。



3. PC 浏览器使用返回的 HTTP 200 OK 中的重定向地址 `http://10.10.12.1:1000/fgtauth?050f0b9d48e81e12` 打开 FortiGate 的认证页面（抓包文件请参考：auth_to_authserver.pcapng ↗）。



4. 所以 PC 需要认证时，访问网页会打开 2 个页面。

解决方法

1. 在用户认证配置中，使用 FortiGate 的自签名 CA 证书 `Fortinet_CA_SSL`（默认没有指定证书，总是使用 HTTP Portal 进行认证），强制使用 HTTPS Portal 进行认证。

重要

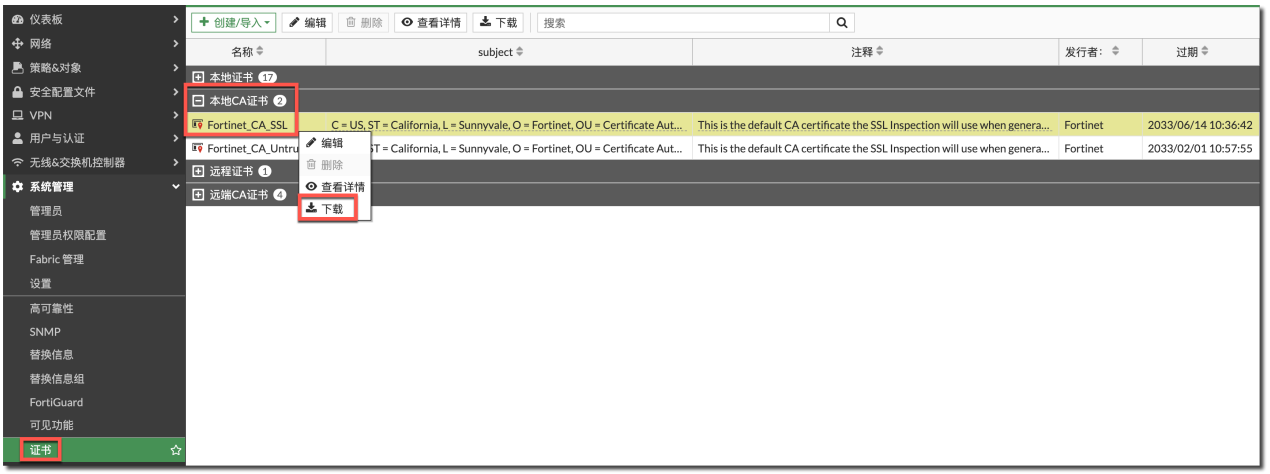
也可以使用其他证书颁发机构（如用户自建 CA）颁发的 Sub CA，需携带私钥导入 FortiGate（导入步骤请参考系统管理 → 系统设置 → 证书管理 ↗的“本地 CA 证书”部分）。

```bash
config user setting
    set auth-ca-cert "Fortinet_CA_SSL"
end
```

2. （可选）使用 IP 或域名作为认证的 Portal（使用域名时，DNS 需要能够解析该域名）。此步可以不配置，默认使用需要认证的流量的入接口 IP 地址作为 Portal 页面的 IP。

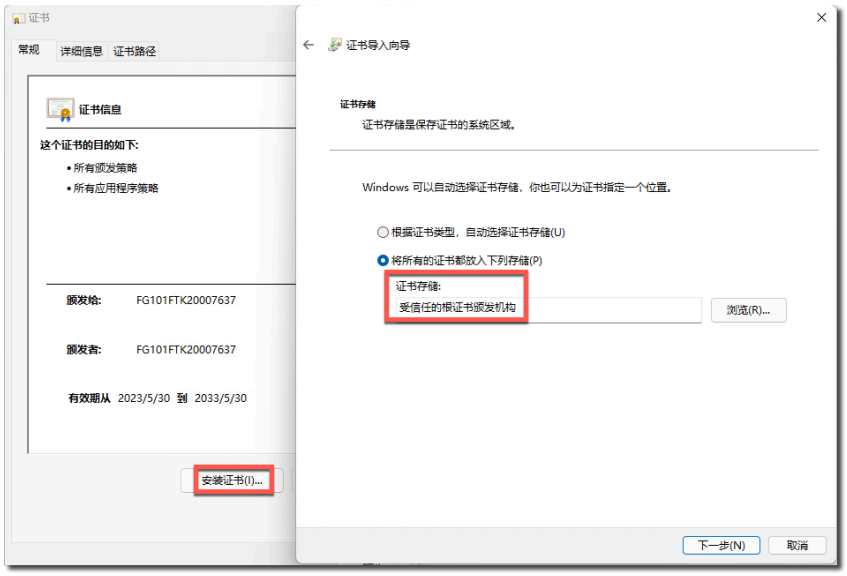
```bash
config firewall auth-portal
    set portal-addr "fgt.portal.com"
end
```

3. 从 FortiGate 管理页面的“系统管理 → 证书”页面中的“本地 CA 证书”中，右键点击并下载 `Fortinet_CA_SSL` 证书。

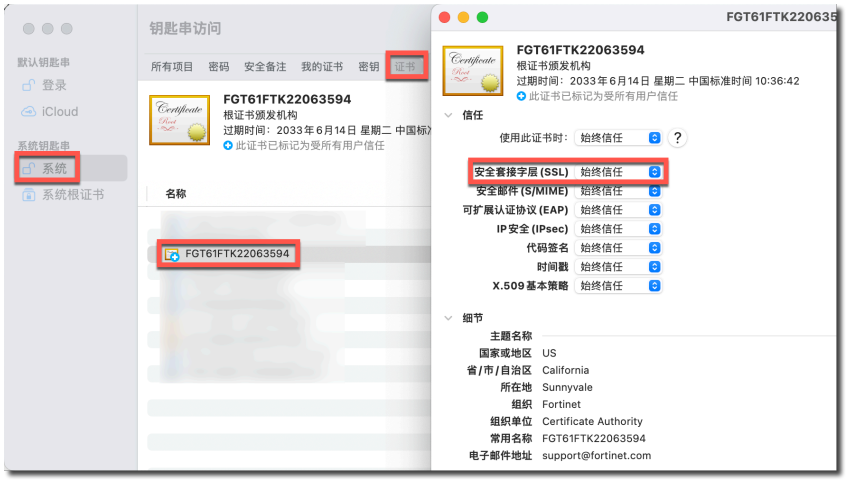


4. 在 PC 上信任该 CA 证书（具体证书信任步骤请参考系统管理 → 管理方式 → 消除 Web 证书告警 ↗的“导入和信任 CA 证书部分”）。

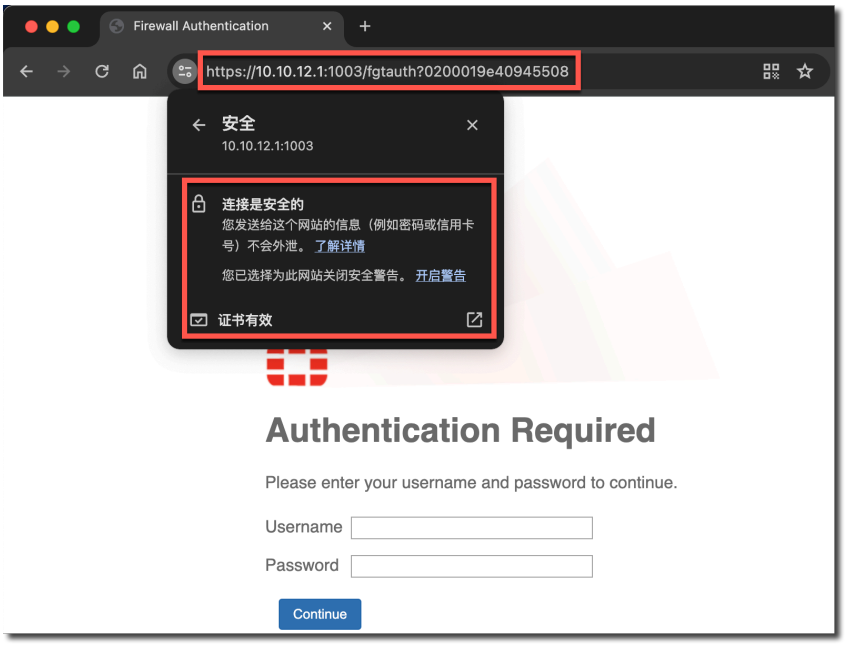
   - Windows：将 CA 证书导入 PC 的“受信任的根证书颁发机构中”。



   - MacOS：在“钥匙串访问”中信任该证书。



5. 再次使用 PC 访问 `https://handbook.fortinet.com.cn`，只会弹出 1 个 HTTPS 认证页面，且不会提示网站不安全。



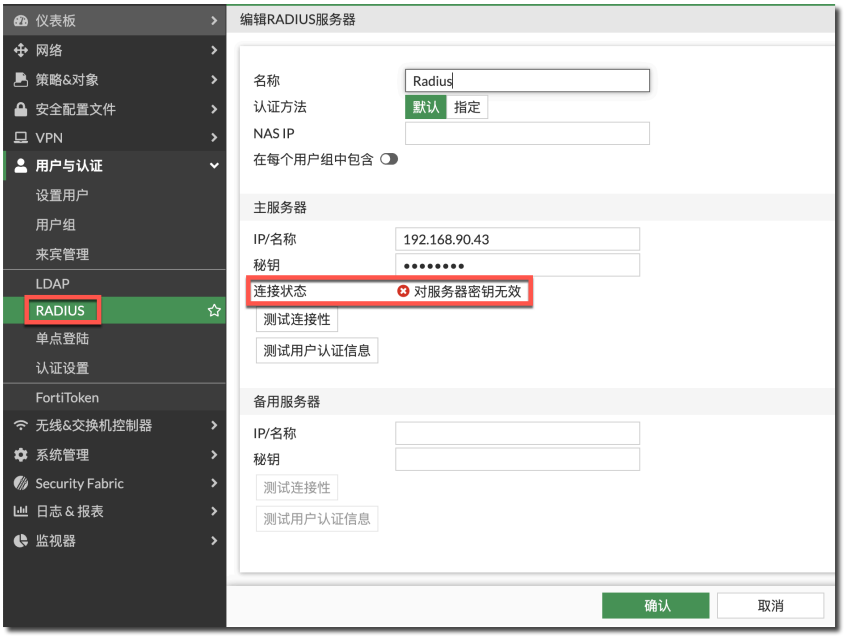
6. 查看该认证页面证书内容，可以看到是由 CA 证书 Fortinet_CA_SSL 签发的证书，CN 为需要认证流量的入接口地址（如果第 2 步中配置了 Portal 域名，则这里的 CN 为对应域名）。


证书查看者：10.10.12.1
基本信息(G)
详细信息(D)
颁发对象
公用名 (CN) 10.10.12.1
组织 (O) Fortinet Ltd.
组织单位 (OU) <未包含在证书中>
颁发者
公用名 (CN) FGT61FTK22063594
组织 (O) Fortinet
组织单位 (OU) Certificate Authority
有效期
颁发日期 2024年4月28日星期日 11:37:35
截止日期 2025年4月28日星期一 11:37:40
SHA-256 指纹
证书 8ba44be84e8da183e545ffc2e0ec4cf824f8c4125a002d0e4c28d0bc169f5a76
公钥 d44990e83630abc91493be5151a680c560fc5136de03bfcad3fb936a27347333

升级版本后 Radius 连接失败

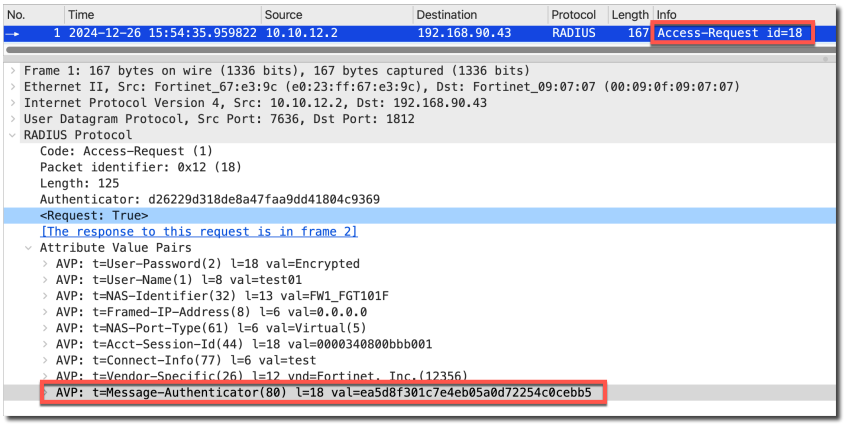
问题现象

在升级到 FortiOS 7.2.10、7.4.5 之后，FortiGate 已配置的 Radius 服务器无法正常连接，显示“对服务器密钥无效”，升级前是可以正常连接的。



问题原因

为了避免 Radius 协议漏洞 CVE-2024-3596 造成的影响，在 FortiOS 7.2.10 和 7.4.5 的版本中，FortiGate 强制验证 Radius Message-Authenticator 属性。即 FortiGate 作为 Radius Client 发送的 Radius Access Request 报文里面会携带 Radius Message-Authenticator 属性，FortiGate 要求 Radius Server 在返回给防火墙的 Radius 报文中也必须要携带 Radius Message-Authenticator 属性，如果 Radius Server 不携带该属性，则 FortiGate 拒绝其响应。



解决方法

1. 升级 Radius 服务器到已经修复漏洞 CVE-2024-3596 的版本，例如 Windows Server 系统在 KB5040268 补丁中增加了对 Radius Message-Authenticator 属性的支持。
2. 在之后的 7.2.11 或 7.4.6 的版本中，有选项可以关闭强制验证 Radius Message-Authenticator 属性（默认开启）。

```bash
config user radius
    edit xxx
        set require-message-authenticator disable
    next
end
```

3. 如果方法 1 和方法 2 都不可行，可以暂时降级到 7.2.9 或 7.4.5 版本。

云部署

AWS

- AWS 部署 FortiGate
- AWS 同 AZ 部署 FortiGate HA
- AWS 跨 AZ 部署 FortiGate HA
- AWS 扩充硬盘容量

阿里云

- 阿里云部署 FortiGate
- 阿里云同 AZ 部署 FortiGate HA
- 阿里云跨 AZ 部署 FortiGate HA

Azure

- Azure 部署 FortiGate
- 结合 Azure 负载均衡器部署 FortiGate HA

注意事项

- 阿里云更换 License 后无法增加 CPU
- SSH 使用账号/密码登录
- 云部署注意事项

AWS

- AWS 部署 FortiGate
- AWS 同 AZ 部署 FortiGate HA
- AWS 跨 AZ 部署 FortiGate HA
- AWS 扩充硬盘容量

AWS 部署 FortiGate

请参考 AWS 部署 FortiGate ↗。

AWS 同 AZ 部署 FortiGate HA

请参考 AWS 同 AZ 部署 FortiGate HA ↗。

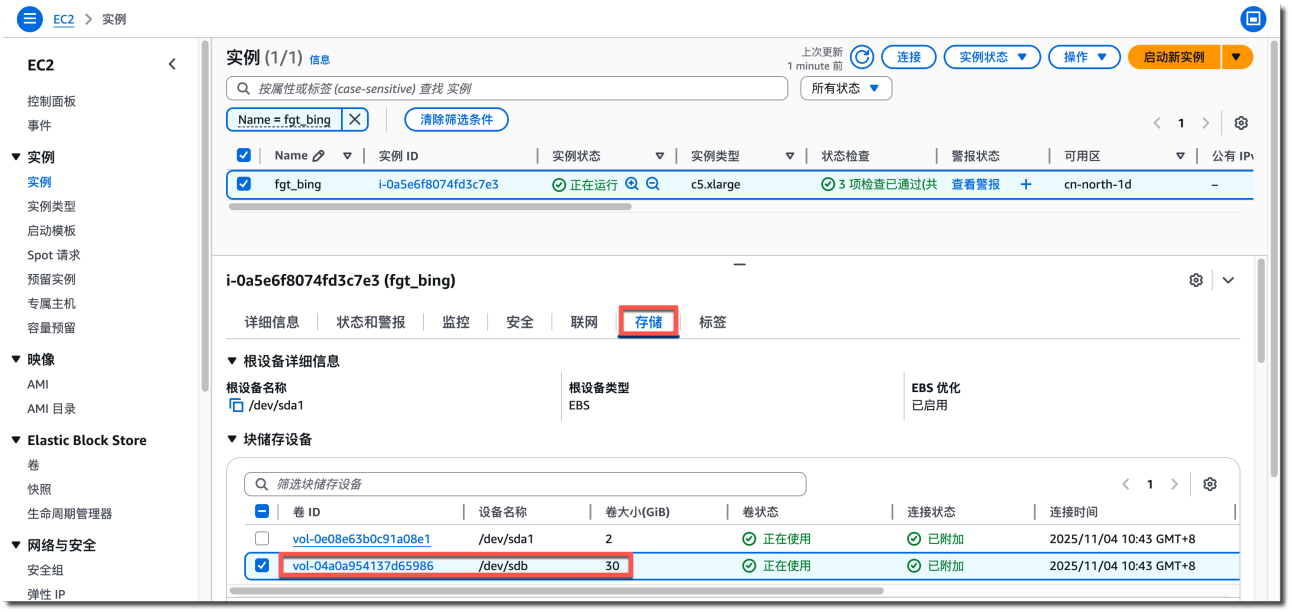
AWS 跨 AZ 部署 FortiGate HA

请参考 AWS 跨 AZ 部署 FortiGate HA ↗。

AWS 扩充硬盘容量

网络需求

某用户在 AWS 上部署的 FortiGate VM，原有硬盘容量分配了 30GB 用于日志记录，随着日志存储量的增加，现在需要将硬盘容量扩充到 200GB 用于日志记录。

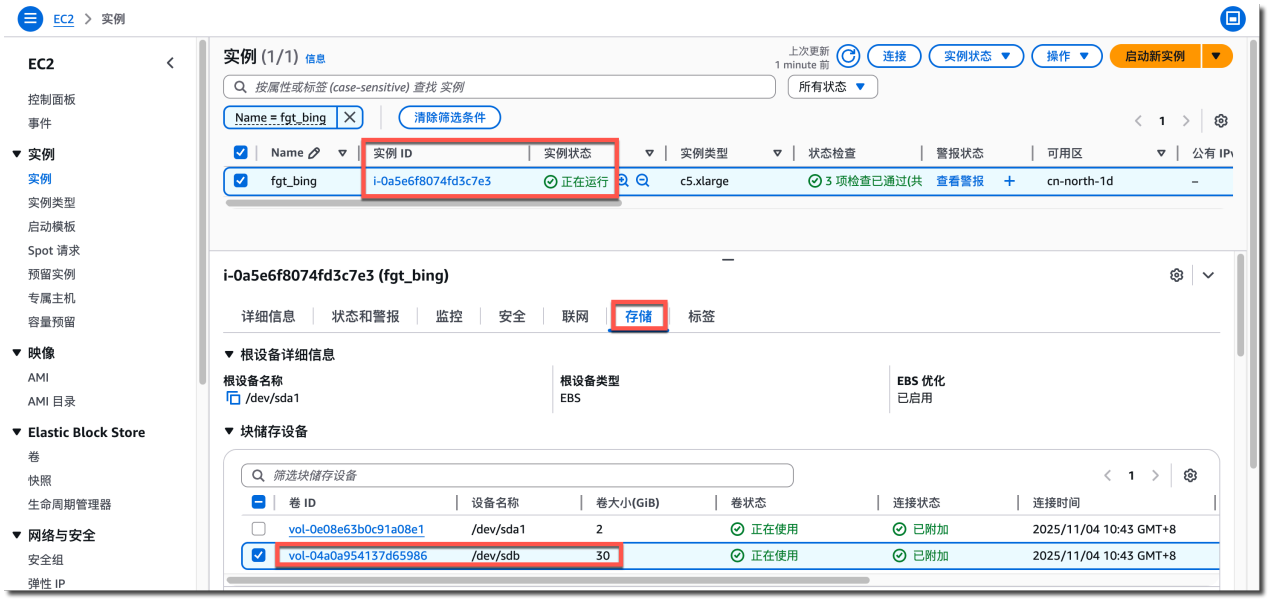



```bash
FortiGate_AWS # exe disk list
Disk Virtual-Disk    ref: 255  30.0GiB    type: SSD [NVME] dev: /dev/nvme1n1
  partition ref:   1  29.4GiB,  29.4GiB free  mounted: Y  label: LOGUSEDX8C26DB0A dev: /dev/nvme1n1p1 start: 2048
```

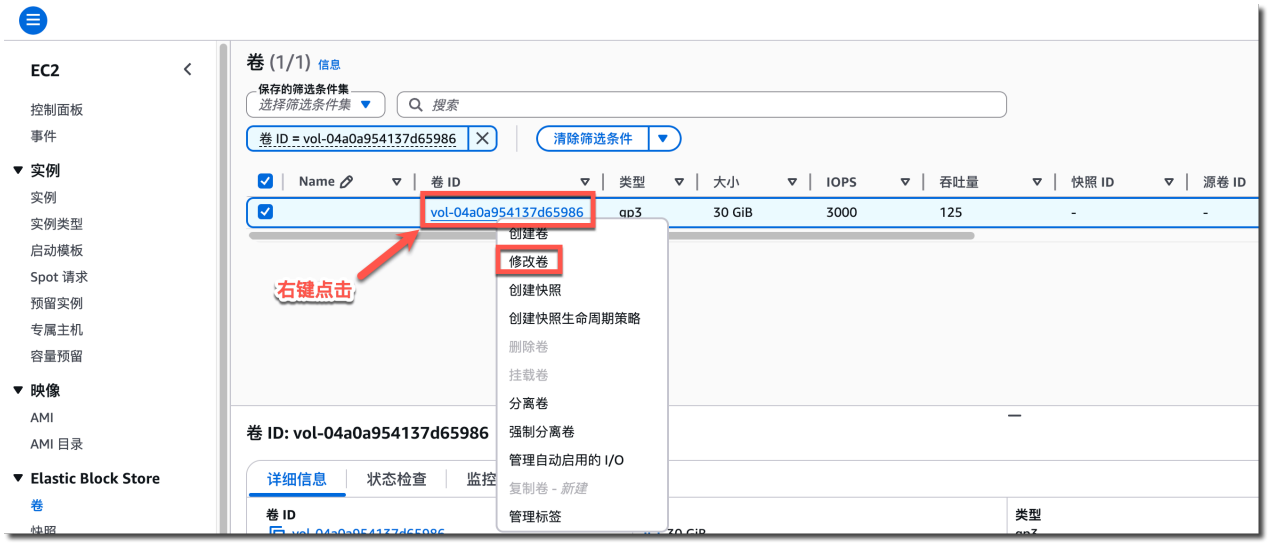
```bash
FortiGate_AWS # show system storage
config system storage
    edit "Virtual-Disk"
        set status enable
        set media-status enable
        set order 1
        set partition "LOGUSEDX8C26DB0A"
        set device "/dev/nvme1n1p1"
        set size 30108
        set usage log
    next
end
```

扩充步骤

1. 无需停止 AWS VM 实例，点击该实例下的存储选项，点击 30GB 的存储卷 ID。

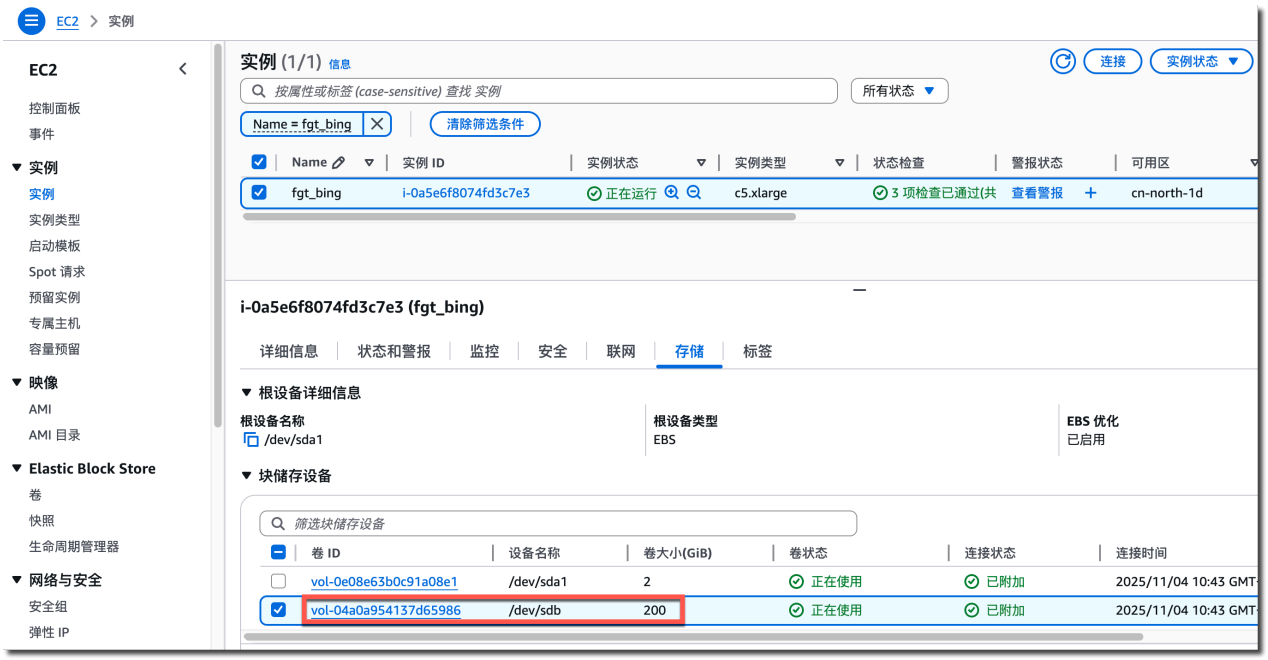


2. 在卷页面右键点击该卷 ID，并选择“修改卷”。

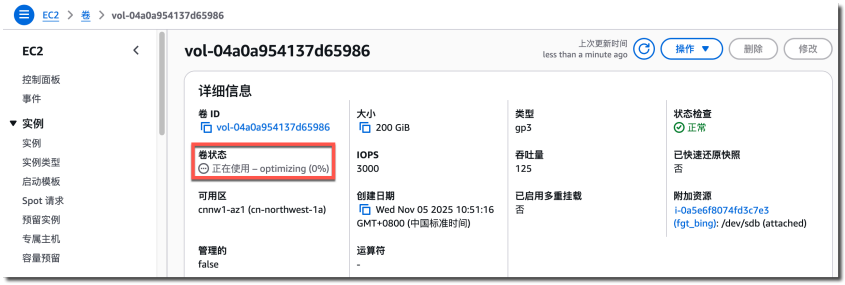


3. 将大小修改为 200GB，然后点击“修改”按钮，之后该实例的硬盘会被扩展到 200GB。





4. 硬盘扩展需要一定时间，期间卷状态状态为“optimizing”。



5. 等待硬盘扩展完成（卷状态变为“正在使用”）。



6. 在 FortiGate 的 CLI 查看硬盘状态，硬盘容量已经变为 200GB，但日志分区仍然为 30GB。

```bash
FortiGate_AWS # exe disk list
Disk Virtual-Disk    ref: 255 200.0GiB    type: SSD [NVME] dev: /dev/nvme1n1
  partition ref:   2  29.4GiB,  29.4GiB free  mounted: Y  label: LOGUSEDX8C26DB0A dev: /dev/nvme1n1p1 start: 2048

FortiGate_AWS # show system storage
config system storage
    edit "Virtual-Disk"
        set status enable
        set media-status enable
        set order 1
        set partition "LOGUSEDX8C26DB0A"
        set device "/dev/nvme1n1p1"
        set size 30108
        set usage log
    next
end
```

7. 需要在 FortiGate 上执行格式化整个硬盘的操作，参考“exe disk list”执行结果中的分区 ref 数字。输入 y 后确认，设备会重启并执行格式化操作。

> **重要**
> - 注意这里执行格式化的分区数字为整个硬盘的 ref 数字（Disk Virtual-Disk ref，此例中为 255），而不是日志分区的 ref 数字（partition ref，此例中为 1）
> - 您需要根据您的 FortiGate VM 中执行“exe disk list”结果中的硬盘 ref 数字来决定“execute disk format”命令后的数字。如您的硬盘 ref 数字（Disk Virtual-Disk ref）为 251，则执行“execute disk format 251”。

```bash
FortiGate_AWS # exe disk list
Disk Virtual-Disk    ref: 255 200.0GiB    type: SSD [NVME] dev: /dev/nvme1n1
  partition ref:   2  29.4GiB,  29.4GiB free  mounted: Y  label: LOGUSEDX8C26DB0A dev: /dev/nvme1n1p1 start: 2048

FortiGate_AWS # execute disk format 255
format requested for:  device=/dev/nvme1n1 255/(no storage);
Formatting this storage will erase all data on it, including
  logs, quarantine files;
  certificate revocation lists (CRLs);
This action requires the unit to reboot.
Do you want to continue? (y/n)y
```

8. 设备重启完成后，再次查看硬盘与日志分区状态，日志分区已经扩容到 200GB。

```bash
FortiGate_AWS # exe disk list
Disk Virtual-Disk    ref: 255 200.0GiB    type: SSD [NVME] dev: /dev/nvme1n1
  partition ref:   1 195.9GiB, 195.8GiB free  mounted: Y  label: LOGUSEDX8C26DB0A dev: /dev/nvme1n1p1 start: 2048

FortiGate_AWS # show system storage
config system storage
    edit "Virtual-Disk"
        set status enable
        set media-status enable
        set order 1
        set partition "LOGUSEDX8C26DB0A"
        set device "/dev/nvme1n1p1"
        set size 200561
        set usage log
    next
end
```

注意事项

1. FortiGate VM 只支持一块硬盘作为日志记录盘，如果是通过新增更大硬盘的方式扩充日志记录盘容量，新增的硬盘卷将无法用于日志记录，建议通过直接扩充现有硬盘卷的容量来扩充日志分区大小。
2. 格式化操作时必须使用“execute disk format XXX”命令，使用“exe formatlogdisk”是无效的。
3. 阿里云、Azure 上的 FortiGate VM 也可以参考此方法进行日志分区扩容。

阿里云

- 阿里云部署 FortiGate
- 阿里云同 AZ 部署 FortiGate HA
- 阿里云跨 AZ 部署 FortiGate HA

阿里云部署 FortiGate

请参考阿里云部署 FortiGate ↗。

阿里云同 AZ 部署 FortiGate HA

请参考阿里云同 AZ 部署 FortiGate HA ↗。

阿里云跨 AZ 部署 FortiGate HA

请参考阿里云跨 AZ 部署 FortiGate HA ↗。

Azure

- Azure 部署 FortiGate
- 结合 Azure 负载均衡器部署 FortiGate HA

Azure 部署 FortiGate

请参考 Azure 部署 FortiGate ↗

结合 Azure 负载均衡器部署 FortiGate HA

请参考结合 Azure 负载均衡器部署 FortiGate HA ↗。

注意事项

- 阿里云更换 License 后无法增加 CPU
- SSH 使用账号/密码登录
- 云部署注意事项

阿里云更换 License 后无法增加 CPU

问题现象

1. 某用户将运行在云上的 FortiGate HA 两台虚机的 License 由评估 License 更换为正式 License，如 FortiGate-VM04 替换为 FortiGate-VM04-S。

2. 更换后，备机 VM 的 CPU 变为 1 个。

```bash
FGT-VM64-KVM-1502 # get system status
......
Serial-Number: FGVMSLTM23007738
License Status: Valid
VM Resources: 1 CPU/1 allowed, 7735 MB RAM
......

FGT-VM64-KVM-1502 # diagnose debug vm-print-license
SerialNumber: FGVMSLTM23007738
CreateDate: Thu Dec 21 03:23:20 2023
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: SL (18)
CPU: 1 (subscription:0)
MEM: 2147483647
VDOM license:
  permanent: 1
  subscription: 0

FGT-VM64-KVM-1502 # diagnose hardware sysinfo vm full
UUID:     95790208a255482da2b95a817a2ffdd3
valid:    1
status:   1
code:     200
warn:     0
copy:     0
received: 4294939474
warning:  4294939474
recv:     202312260754
dup:
```

3. 无法通过 `exe cpu add` 添加 CPU。

```bash
FGT-VM64-KVM-1502 # exe cpu add 3
CPU number exceeds license limit.
Command fail. Return code -61
```

解决办法

1. 将 FortiGate HA 主备切换后，在新主机（原备机）执行 `execute update-now` 手动触发更新，之后可以正常添加 CPU Core。

```bash
FGT-VM64-KVM-1502 # exe update-now
......
FGT-VM64-KVM-1502 # exe cpu add 3
Active CPU number: 4
Total CPU number: 4

FGT-VM64-KVM-1502 # exe cpu show
Active CPU number: 4
Total CPU number: 4
```

2. 查看该 VM 相关参数均显示为新 License 的 CPU 信息。

```bash
FGT-VM64-KVM-1502 # get system status
......
Serial-Number: FGVMSLTM23007738
License Status: Valid
VM Resources: 4 CPU/4 allowed, 7735 MB RAM
......

FGT-VM64-KVM-1502 # diagnose debug vm-print-license
SerialNumber: FGVMSLTM23007738
CreateDate: Thu Dec 21 03:23:20 2023
License expires: Wed Feb 21 08:00:00 2024
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: SL (18)
CPU: 4 (subscription:4)
MEM: 2147483647
VDOM license:
  permanent: 1
  subscription: 0

FGT-VM64-KVM-1502 # diagnose hardware sysinfo vm full
UUID:     95790208a255482da2b95a817a2ffdd3
valid:    1
status:   1
code:     200
warn:     0
copy:     0
received: 4294939353
warning:  4294986323
recv:     202312260824
dup:
```

SSH 使用账号/密码登录

云上部署完成的 FortiGate，默认会使用实例创建过程中配置的密钥对来登录 SSH，无法通过账户/密码的方式登录 SSH。

如果要使用账户/密码的方式登录 SSH，需要在 FortiGate 中执行以下 CLI 命令：

```
config system global
    set admin-ssh-password enable
end
```

随后即可以通过账户/密码的方式登录云上部署的 FortiGate。

云部署注意事项

在云平台部署 Fortinet 相关产品时，请注意您需要承担所使用的资源产生的费用，包括但不限于：CPU、内存、存储卷、快照、数据传输及网络带宽。

当部署的 VM 正式运行后，系统可能会自动生成临时文件、系统日志或额外的存储卷与快照，这些都可能占用磁盘空间并产生额外费用。

为了帮助您更好地管理成本，建议采取以下措施：

- **定期检查**云环境中运行的收费服务与功能。
- **监控磁盘使用情况**，了解存储卷或快照的生成机制。
- **设置合理的用量限制、配额和预算告警**，以便及时掌握资源消耗情况。
- **配置磁盘空间告警**，并在收到告警后及时调整。

不同云服务提供商提供的管理工具可能有所差异，您可以参考相关云平台官方文档，配置适合的告警、预算及用量控制策略。

Security Fabric

威胁情报源

自动化

- 工作流
- 常用触发器

  系统事件日志触发器　UTM 日志触发器　证书过期触发器　计划时间触发器
- 常用动作

  内存/CPU 达到阈值时收集信息　Email 动作　Teams Webhook　系统动作　自动抓包
- 常用动作

威胁情报源

简介

威胁源列表是一个动态列表，其中可以包含域名、IPv4/IPv6 地址、地址范围及子网、MAC 地址等。FortiGate 可将其用作防火墙策略、代理策略、Local-in 策略、ZTNA 规则中的源地址或目标地址，DNS Filter 分类、Web Filter 分类等功能中。

威胁源列表有以下几种类型：

- **FortiGuard 分类**：可以在 WebFilter 中的远程分类中使用。
- **IP 地址**：IPv4/IPv6 地址、地址范围及子网。可将其用作防火墙策略、代理策略、Local-in 策略、ZTNA 规则中的源地址或目标地址。
- **域名**：可将其作为 DNS 过滤配置文件中 "远程类别" 组下的一个类别，用于阻止或监控匹配该类别的域名。
- **MAC 地址**：可以包含 MAC 地址、MAC 地址范围及 MAC OUI。可将其作为防火墙策略、代理策略及 ZTNA 规则的源。对于透明模式策略或虚拟线对（VWP）策略，MAC 地址威胁源可作为源地址或目标地址使用。
- **恶意软件哈希**：FortiGate 获取该列表后，当在 AV 启用 "使用外部恶意软件阻止列表" 功能后，可以选择将其应用于 AV 的文件哈希检查。该功能不需要开启“防病毒扫描”（av-scan）功能。

威胁源列表有两种更新方式：

- **外部订阅**
  - 该文件需要以文本文件格式（txt）存储于外部服务器（HTTP/HTTPS），FortiGate 会从外部服务器定期更新该列表。
- 如果 FortiGate 与外部服务器失去连接，威胁源列表将继续运行。但威胁源列表将不会更新（添加/删除条目），直到与外部服务器重新建立连接。
- **API 推送（FortiOS 7.2.1 开始支持）**
  - 通过 Push API 的方式更新威胁源列表。
  - 威胁源列表存放在 FortiGate 本地，而不是 HTTP/HTTPS 服务器上。
  - Push API 的方法分为三种：add（添加）、remove（删除）、snapshot（覆盖原有全部条目为推送的条目）。

资源格式要求

文件格式

- 外部资源需要使用纯文本格式，其中每个条目独占一行。文件内可通过 `#` 号添加注释，例如：`# 这是注释内容`。
- 外部资源更新周期可设置为 1 ~ 43200 min。
- 外部威胁源不进行重复条目验证。
- 若外部威胁源内的条目数量超过限制，将显示警告，超出阈值的额外条目将不会被加载。

FortiGuard 分类

- 可以使用通配符，例如：`*.domain.com`。
- 支持 IDN 和 UTF 编码的 URL。
- URL 可为 IPv4 或 IPv6 地址。IPv6 URL 必须采用 `[ ]` 格式，如 `[2100::2]`。
- 示例 txt 文本：

```bash
# 以下为示例内容
www.fortinet.com
*.fortinet.com
münchen.de
202.103.1.1
[2001:0db8::eade:27ff:fe04:9a01]
```

IP 地址

- IP 地址可为单个地址、子网地址或地址范围。例如：`192.168.1.1`、`192.168.10.0/24`、`192.168.100.1-192.168.100.254`。
- 地址可为 IPv4 或 IPv6 格式。
- 示例 txt 文本：

```bash
# 以下为示例内容
192.168.2.100
172.200.1.4/32
172.16.1.0/24
172.16.8.1-172.16.8.100
2001:db8::eade:27ff:fe04:9a00/120
2001:0db8::eade:27ff:fe04:aa01-2001:0db8::eade:27ff:fe04:ab01
```

域名

- 域名列表支持通配符，例如：`*.test.com`。
- 支持 IDN 格式。
- 示例 txt 文本：

```bash
# 以下为示例内容
*.test.com
www.fortinet.com
handbook.fortinet.com.cn
münchen.de
新华网.cn
```

- DNS Filter 引用示例：

```bash
config system external-resource
    edit "fqdn"
        set type domain
        set category 193
        set resource "https://handbook.fortinet.com.cn/fqdn.txt"
    next
end

config dnsfilter profile
    edit "default"
        set external-ip-blocklist "fqdn"
    next
end
```

MAC 地址

- MAC 地址可以是单个地址、MAC OUI 或地址范围。如：`01:01:01:01:01:01`、`8c:aa:b5`、`01:01:01:01:01:01-01:01:02:50:20:ff`。
- MAC 地址中的十六进制数字必须用`:`分隔。
- 示例 txt 文本：

```bash
# 以下为示例内容
01:01:01:01:01:01
8c:aa:b5
01:01:01:01:01:01-01:01:02:50:20:ff
```

- 防火墙策略引用示例：

```bash
config system external-resource
    edit "mac"
        set type mac-address
        set resource "https://handbook.fortinet.com.cn/mac.txt"
    next
end

config firewall policy
    edit 2
        set srcintf "lan"
        set dstintf "wan1"
        set action deny
        set srcaddr "mac"
```

```
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

恶意软件哈希

- 恶意软件哈希列表需遵循严格格式方为有效。每个哈希签名条目必须独占一行。有效签名格式如下：

```bash
# 加上hash算法的描述，以空格隔开
aa67243f746e5d76f68ec809355ec234  md5

# 加上hash算法的描述，以空格隔开
a57983cb39e25ab80d7d3dc05695dd0ee0e49766  sha1

# 加上hash算法的描述，以空格隔开
ae9bc0b4c5639d977d720e4271da06b50f7c60d1e2070e9c75cc59ab30e49379 sha256

# 无效条目
7688499dc71b932feb126347289c0b8a_md5_sample2
7614e98badca10b5e2d08f8664c519b7a906fbd5180ea5d04a82fce9796a4b87sha256_sample3
```

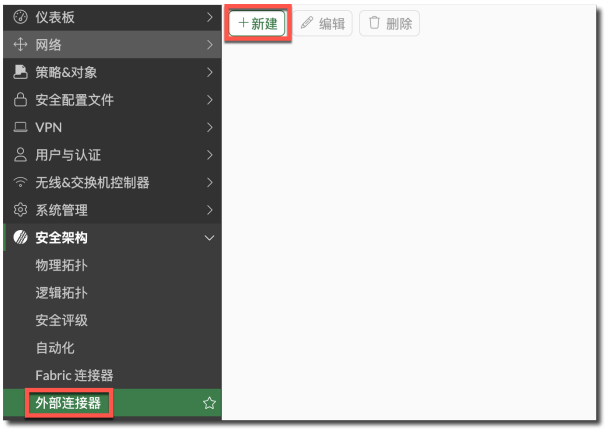
- AV 配置文件引用示例：

```bash
config system external-resource
    edit "av"
        set type malware
        set resource "https://handbook.fortinet.com.cn/av.txt"
    next
end

config antivirus profile
    edit "default"
        set external-blocklist "av"
    next
end
```

配置步骤

外部订阅

1. 这里以 IP 地址威胁源为例。进入 FortiGate 的“安全架构 → 外部连接器”页面，点击“新建”按钮。



2. 选择“外部源”中的“IP 地址”选项。



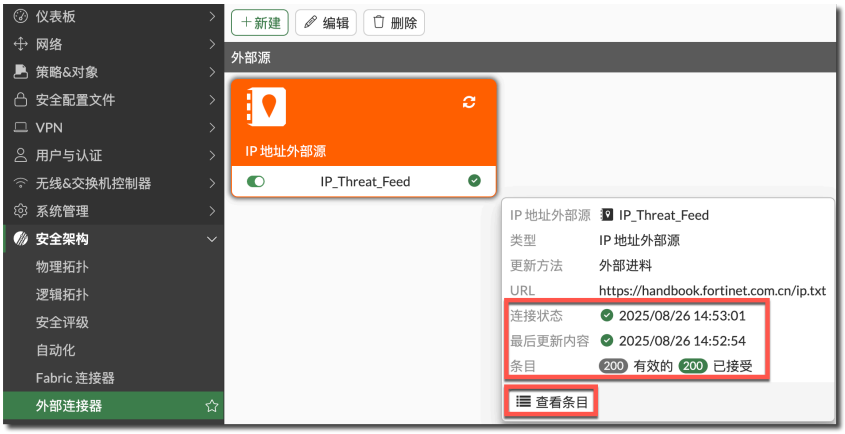
3. 配置 IP 威胁源列表的名称，更新方法选择“外部进料”，外部资源 URL 填写 HTTP/HTTPS 服务器上的 txt 文本文件链接，按需开启“HTTP 认证”与“客户端认证书认证”，刷新率使用默认的 5min。

> **重要**
>
> - 示例中提供的 URL 仅用于演示目的，并不代表一个维护良好的 IP 地址列表。建议您使用一个信誉良好的外部 IP 地址列表 URL，且该列表应定期更新。
> - 为了提高连接的安全性，建议使用 HTTPS 链接，并按需开启“HTTP 认证”与“客户端认证书认证”。

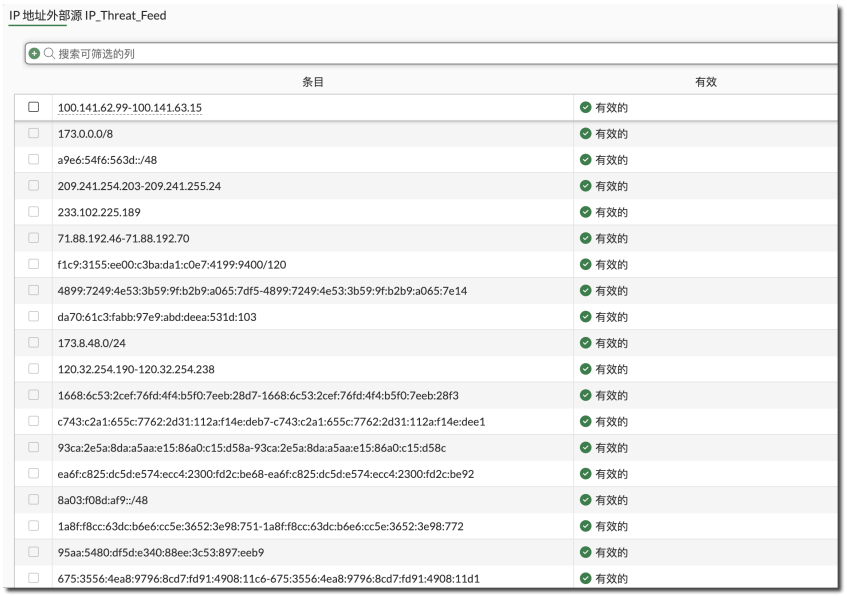



```bash
config system external-resource
    edit "IP_Threat_Feed"
        set type address
        set resource "https://handbook.fortinet.com.cn/ip.txt"
    next
end
```

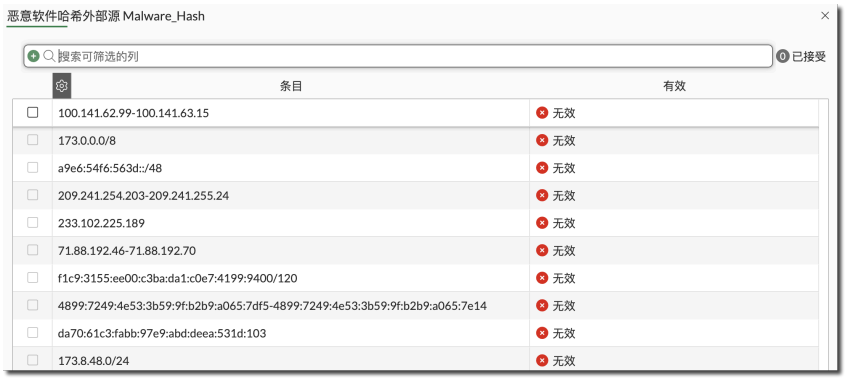
4. 鼠标悬浮至该 IP 威胁源列表上，可以看到该外部源的连接状态（绿色√表示与外部服务器正常连接，红色 × 表示与外部服务器通信中断）、最后更新时间和当前的条目数，点击右上角的刷新按钮可以立刻请求更新。



5. 点击查看条目按钮，可以看到 FortiGate 从外部服务器上获取到的 IP 地址列表，有效状态表示该 IP 格式是否符合 FortiGate 要求。



6. 如果是无法识别的 IP 格式，则会显示红色 ×，如下所示恶意软件哈希外部源获取到的格式为 IP 地址，就会显示无效标记。



7. 创建防火墙策略，在目标地址中引用该 IP 威胁源列表对象，动作选择“拒绝”。

重要

由于示例里的 IP 威胁源列表中同时包含 IPv4 和 IPv6 地址，针对防火墙策略配置：

- 源要同时引用 IPv4 地址对象和 IPv6 地址对象。
- 目的地址要同时引用 IPv4 威胁源和 IPv6 威胁源，FortiGate 获取到“IP 威胁源列表”后，会自动将其中的 IPv4 和 IPv6 地址分别放在两个地址对象中。
- 当然，也可以将此防火墙策略拆分成 2 条，一条 IPv4 的，一条 IPv6 的。




```bash
config firewall policy
    edit 2
        set name "Deny_Threat_IP"
        set srcintf "lan"
        set dstintf "wan1"
        set srcaddr "192.168.100.0/24"
        set dstaddr "IP_Threat_Feed"
        set srcaddr6 "all"
        set dstaddr6 "IP_Threat_Feed"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

8. 内网客户端穿过 FortiGate 访问 IP 威胁源列表中的 IP 地址，访问失败，FortiGate 产生了拦截的防火墙日志。

```bash
date=2025-08-26 time=15:28:29 eventtime=1756193309197961159 tz="+0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" srcip=192.168.100.177 srcport=53915 srcintf="lan" srcintfrole="lan"
dstip=17.57.145.25 dstport=5223 dstintf="wan1" dstintfrole="wan" dstthreatfeed="IP_Threat_Feed" srccountry="Reserved"
dstcountry="United States" sessionid=392389 proto=6 action="deny" policyid=2 policytype="policy" poluuid="f18e907a-824b-
51f0-8704-409b8dba5152" policyname="Deny_Threat_IP" service="tcp/5223" trandisp="noop" appcat="unscanned" duration=0
sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 crscore=30 craction=131072 crlevel="high" srcserver=0
```

9. 在 Local-in 策略中调用 IP 威胁源列表。

```
config firewall local-in-policy
    edit 1
        set intf "port1"
        set srcaddr "IP_Threat_Feed"
        set dstaddr "all"
        set action deny
        set service "ALL"
        set schedule "always"
    next
end
```
bash

.0. 在 DNS 过滤器中调用 IP 威胁源列表，可以阻止客户端以 DNS 的方式访问威胁源中的 IP。

```
config dnsfilter profile
    edit "dns_profile"
        set external-ip-blocklist "IP_Threat_Feed"
    next
end

date=2025-08-26 time=15:06:50 eventtime=1675724810452621179 tz="-0800" logid="1501054400" type="utm" subtype="dns"
eventtype="dns-response" level="warning" vd="root" policyid=0 sessionid=555999 srcip=192.168.100.178 srcport=35113
srccountry="Reserved" srcintf="lan" srcintfrole="undefined" dstip=17.57.145.25 dstport=53 dstcountry="Reserved"
dstintf="root" dstintfrole="undefined" proto=17 profile="dns_profile" xid=25532 qname="www.fortinet.com" qtype="A"
qtypeval=1 qclass="IN" ipaddr="2.2.2.2" msg="Domain was blocked because it is in the domain-filter list" action="redirect"
domainfilteridx=0 domainfilterlist="IP_Threat_Feed"
```
bash

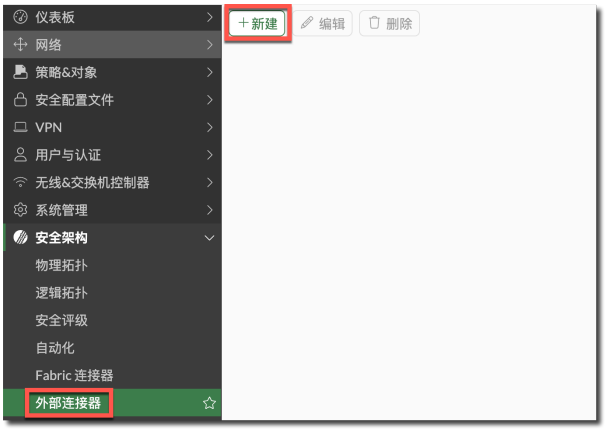
API 推送

重要

该功能在 FortiOS 7.2.1 及更高版本支持。

1. 这里以 FortiGuard 威胁源为例。创建 API 用户，赋予 super_admin 权限，并生成 API key。配置方法请参考：系统管理 → 管理员配置 → API 用户↗章节。

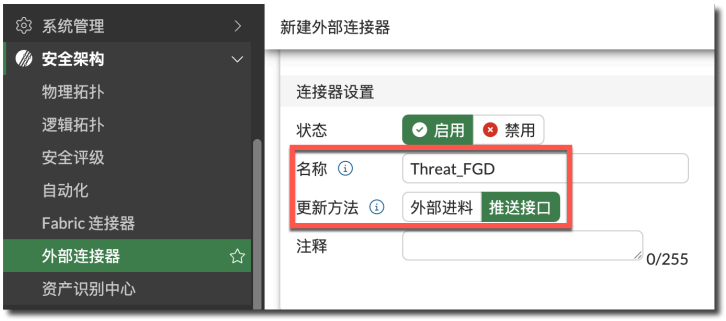
2. 进入 FortiGate 的“安全架构 → 外部连接器”页面，点击“新建”按钮。



3. 选择“外部源”中的“FortiGuard 分类”选项。



4. 配置 FortiGuard 分类威胁源的名称，更新方法选择“推送接口”，点击确认按钮。




```bash
config system external-resource
    edit "Threat_FGD"
        set update-method push
        set category 192
    next
end
```

5. 随后会弹出 API 示例窗口，在此窗口中，可以根据输入的 API 密钥、条目（可以多个，以，隔开）、推送命令（添加、删除、快照），来即时生成对应的 cURL 请求样本，生成后，点击右侧的复制按钮可以复制 cURL。注意是否需要修改 IP 为设备接受 API 推送的接口 IP，这里以添加 2 个域名为例。



6. 客户端执行该 cURL，结果显示执行成功（如果报 403 错误，可能是 API 用户权限不够）。

```bash
# curl -k -X POST -H 'Authorization: Bearer q6G9mhwppNjNns40xN5qp45pzmxHHG' --data '{ "commands": [{ "name": "Threat_FGD", "command": "add", "entries": [ "www.summerice.com", "www.abc.com" ] } ]}' "https://192.168.100.99/api/v2/monitor/system/external-resource/dynamic"

{"http_method":"POST","results":[{"name":"Threat_FGD","command":"add","status":"success"}],"vdom":"root","path":"system","name":"external-resource","action":"dynamic","status":"success","serial":"FG101FTK20007637","version":"v7.6.4","build":3596}%
```

7. GUI 查看该威胁源的条目数为 2。



8. 点击查看条目按钮，可以看到两个域名被成功添加。



9. 失败的 API 更新，可以返回错误原因，如下错误原因为更新方法填写错误。

```bash
# curl -k -X POST -H 'Authorization: Bearer q6G9mhwppNjNns40xN5qp45pzmxHHG' --data '{ "commands": [{ "name":
"Threat_FGD", "command": "update", "entries": [ "www.summerice.com", "www.abc.com" ] } ]}'
"https://192.168.100.99/api/v2/monitor/system/external-resource/dynamic"

{"http_method":"POST","results":[{"name":"Threat_FGD","command":"update","error":"Invalid
command.","status":"error"}],"vdom":"root","path":"system","name":"external-
resource","action":"dynamic","status":"success","serial":"FG101FTK20007637","version":"v7.6.4","build":3596}%
```

0. 在 WebFilter 配置文件中的远程分类中，将上述创建的 FortiGuard 威胁源的动作配置为“屏蔽”。

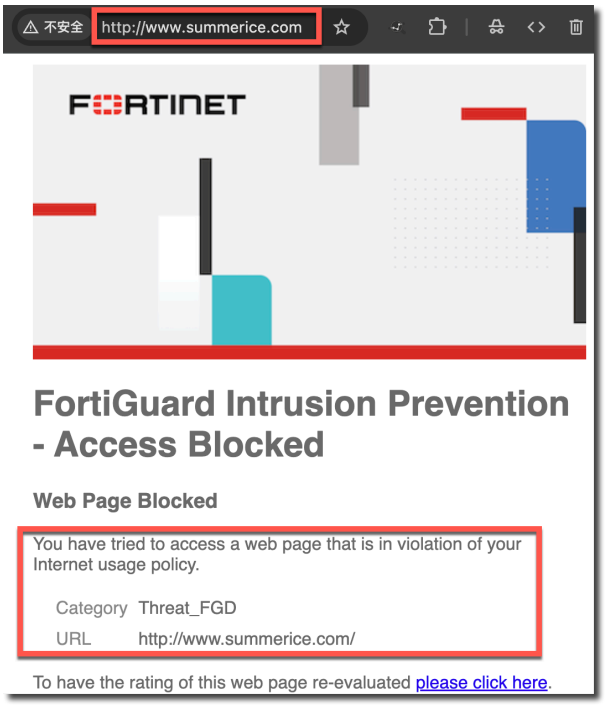



```bash
config webfilter profile
    edit "Treat_FGD"
        config ftgd-wf
            config filters
                edit 35
                    set category 192
                    set action block
                next
            end
        end
    next
end
```

1. 在上网的防火墙策略中调用该 WebFilter 配置文件。

```bash
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "Treat_FGD"
        set logtraffic all
        set nat enable
    next
end
```

2. 客户端浏览器通过 FortiGate 访问 FortiGuard 威胁源中的域名，可以正常拦截威胁源中的网址。



3. cURL 中也可以在 commands 字段中包含多个威胁源，执行不同的动作，如下所示，分别对多个威胁源连接器进行更新，以 , 隔开。

```bash
# curl -k -X POST -H 'Authorization: Bearer q6G9mhwppNjNns40xN5qp45pzmxHHG' --data '{ "commands": [{ "name": "Threat_FGD", "command": "add", "entries": [ "www.summerice.com", "www.abc.com" ]} ,
{"name":"ip","command":"delete","entries":["10.10.10.3/32","10.10.10.4/32"]},
{"name":"domain","command":"snapshot","entries":["www.baidu.com","www.sina.com.cn"]}]}'
"https://192.168.100.99/api/v2/monitor/system/external-resource/dynamic"
```

CLI 相关

- 查看威胁源已获取并保存在 FortiGate 上的列表内容。

```bash
FortiGate # diagnose sys external-resource list IP_Threat_Feed
Dump external resource address table 'IP_Threat_Feed':
167.3.182.140
225.60.62.231/24
31.158.227.180-31.158.228.166
52.231.72.77/24
152.39.145.129/8
......

FortiGate # diagnose sys external-resource list Threat_FGD
Dump external resource category 'Threat_FGD':
www.abc.com
www.summerice.com
```

- 查看外部威胁源列表的统计信息。

```bash
FortiGate # diagnose sys external-resource stats
name: IP_Threat_Feed; uuid_idx: 662; type: address table; update_method: feed; total lines: 200; valid lines: 200; error lines: 0; used: yes; buildable: 200; total in count file: 200; ranges: 151;
name: Threat_FGD; uuid_idx: 667; type: category; update_method: push; total lines: 2; valid lines: 2; error lines: 0; used: yes; buildable: 2; total in count file: 2;
```

- 通过 FortiGate 的 CLI 对 Push API 类型的威胁源进行更新（add、remove、snapshot）。

```bash
diagnose sys external-resource {push-add | push-remove | push-snapshot} <feed_name> <entry>

FortiGate # diagnose sys external-resource push-remove Threat_FGD www.abc.com
Command result: { "name": "Threat_FGD", "command": "remove", "status": "success" }

FortiGate # diagnose sys external-resource list Threat_FGD
Dump external resource category 'Threat_FGD':
www.summerice.com
```

- 通过 FortiGate 的 CLI 对威胁源进行更新（使用 json command 格式，可以对多个威胁源执行不同的动作）。

```bash
FortiGate # diagnose sys external-resource push-api-json-commands '{"commands": [{"name":"ip","command":"add","entries":
["10.10.10.1","10.10.10.2"]}, {"name":"Threat_FGD","command":"snapshot","entries":
["www.fortinet.com","www.sina.com.cn"]}]}'

Returned json:
[
  {
    "name":"ip",
    "command":"add",
    "status":"success"
  },
  {
    "name":"Threat_FGD",
    "command":"snapshot",
    "status":"success"
  }
]

FortiGate # diagnose sys external-resource list ip
Dump external resource address table 'ip':
10.10.10.2/32
10.10.10.1
10.10.10.2
10.10.10.1/32

FortiGate # diagnose sys external-resource list Threat_FGD
Dump external resource category 'Threat_FGD':
www.fortinet.com
www.sina.com.cn
```

配置规格

外部资源文件的最大容量和条目数量受设备型号限制。

- 外部资源个数的配置规格可以使用 `print tablesize` 查看 `system.external-resource` 部分，如下所示，此 FortiGate 的全局限制为 512 个，每个虚拟域 VDOM 的限制为 256 个。

```bash
FortiGate # print tablesize
...
system.external-resource: 0 256 512 0
...
```

- 每种外部威胁源文件中的最大条目数及每种型号的文件大小限制如下（入门型号、中端型号、高端！）型号的具体型号可以参考：系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗章节）：

外部威胁源类型	入门型号	中端型号	高端型号
FortiGuard 分类	150 000	300 000	2 000 000
IP 地址	300 000	1 000 000	5 000 000
域名	1 000 000	3 000 000	5 000 000
MAC 地址	1 000 000	1 000 000	1 000 000
文件大小限制（MB）	32	64	128

ext-resource-session-check

外部威胁源对象（如 IP、域名、Web 分类等）的内容发生变化后，相关会话的 dirty 状态变化会根据 `ext-resource-session-check` 配置决定。

> **重要**
>
> 会话 dirty 的机制请参考：策略与对象 → 会话 dirty 机制 ↗章节。

```bash
config system settings
    set fqdn-session-check { enable | disable }
end
```

- `disable`（默认）：现有会话不会被标记为 `dirty`，而是被标记为 `persistent`，原有匹配该会话的流量也不会被重新评估（直至该会话老化），仅检查新创建的流量。
- `enable`：现有相关会话被标记为 `dirty`，无论是匹配原有会话的流量还是新创建的流量，重新评估为新会话。

自动化

工作流

常用触发器

- 系统事件日志触发器
- UTM 日志触发器
- 证书过期触发器
- 计划时间触发器

常用动作

- 内存/CPU 达到阈值时收集信息
- Email 动作
- Teams Webhook
- 系统动作
- 自动抓包

常用动作

工作流

自动化简介

自动化（Automation）功能能够根据特定的触发条件自动执行相应的动作，可以让客户更快的发现设备发生的事件，以及更快地处理事件。它可以监控来自 Security Fabric 中任何成员的事件，然后设置对应的执行动作。

自动化设置可以在 Security Fabric 成员中同步，或只用于 Security Fabric 中的一个单独 FortiGate。当 FortiGate 加入了 Security Fabric，自动化只能在 Security Fabric 的 Root FortiGate 上创建。

当然，如果一台 FortiGate 未加入 Security Fabric，也可以使用自动化功能。

FortiGate 上的每个自动化配置包含 3 个部分：

- 触发器（Trigger）：用于触发自动化流程的条件，可以是某个系统事件日志、UTM 日志、计划时间等。系统会内置一些常用的 Trigger。
- 动作（Action）：触发器触发后执行的动作，可以是执行 CLI 脚本、发送邮件、重启设备等。系统会内置一些常用的 Action。
- 串联器（Stitch）：将触发器（Trigger）和动作（Action）关联起来，在 Stitch 中，一个 Stitch 可以包含 1 个 Trigger 和多个 Action，可以单独设置每个 Action 和 Trigger 的执行延迟时间。系统会内置一些常用的 Stitch。Stitch 有两种执行 Action 的模式：
  - 顺序：如果有多个动作（Action），触发器（Trigger）触发后，按照动作（Action）的顺序依次执行。
  - 并行：如果有多个动作（Action），触发器（Trigger）触发后，同时执行所有动作（Action）。

工作流（Workflow）就是每个自动化配置的统称。

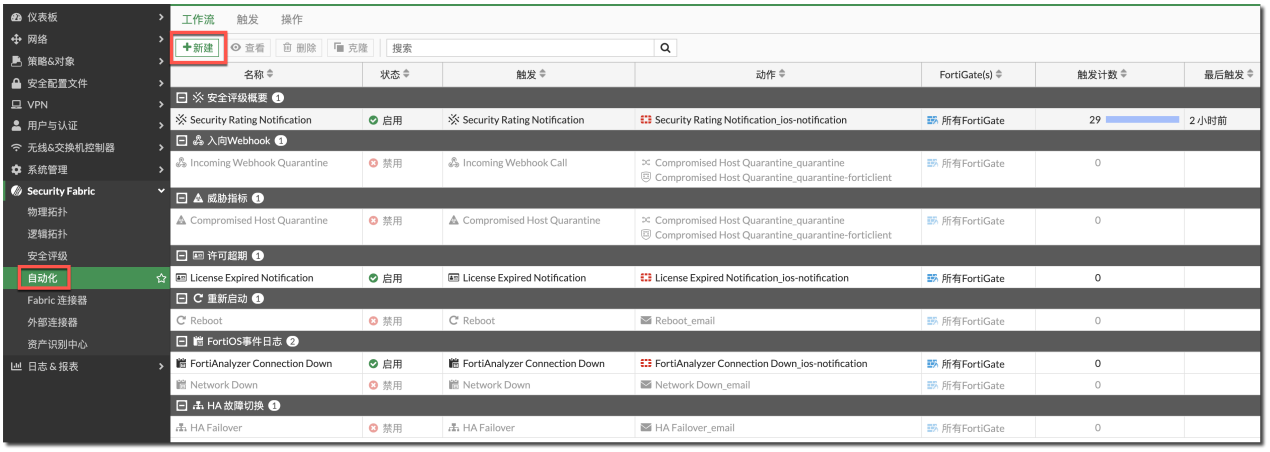
单动作配置示例

网络需求

当 IPSec 由于 DPD 超时中断，等待 5s 后，向管理员的邮箱发送通知邮件。

配置步骤

1. 首先配置告警邮件发送的配置（参考日志 → 邮件告警 ↗章节，也可使用默认配置）。
2. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击新建按钮，新建一个工作流（此页也有一些内置的工作流，可根据需求自行使用或禁用）。



3. 配置工作流的名称，点击“添加触发器”按钮，在右侧的“选择条目”中点击“新建”按钮。



- 操作执行 - 顺序：如果有多个动作（Action），触发器（Trigger）触发后，按照动作（Action）的顺序依次执行。
- 操作执行 - 并行：如果有多个动作（Action），触发器（Trigger）触发后，同时执行所有动作（Action）。

4. 在右侧弹出的窗口中，找到“其他条件”分类，点击“FortiOS 事件日志”选项。



5. 配置触发器（Trigger）的名称，事件选择“IPSec DPD failed”（可根据需求在字段过滤中配置日志中要求包含的字段），点击确认下发配置。




```bash
config system automation-trigger
    edit "DPD_Failure"
        set event-type event-log
        set logid 37136
    next
end
```

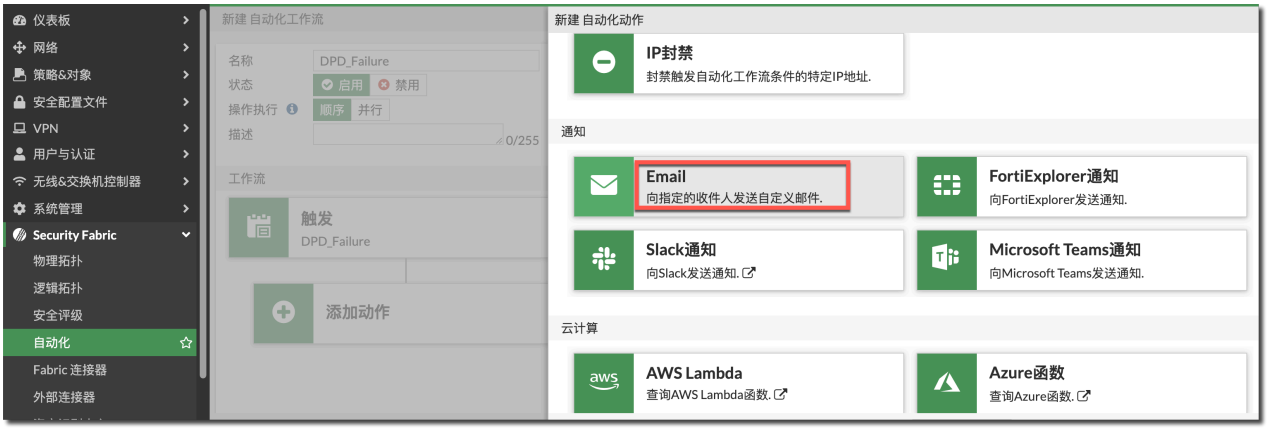
6. 在选择条目中选择上步创建的触发器（Trigger），点击应用按钮。



7. 点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



8. 在右侧弹出的窗口中，找到“通知”分类，点击“Email”选项。



9. 配置动作（Action）的名称，以及邮件通知的收件人，邮件内容为 DPD 失败的日志内容，点击确认下发配置。



- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 `DoNotReply@notification.fortinet.net`。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

```bash
config system automation-action
    edit "Email_DPD_Failure"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "DPD_Failure_Notice"
    next
end
```

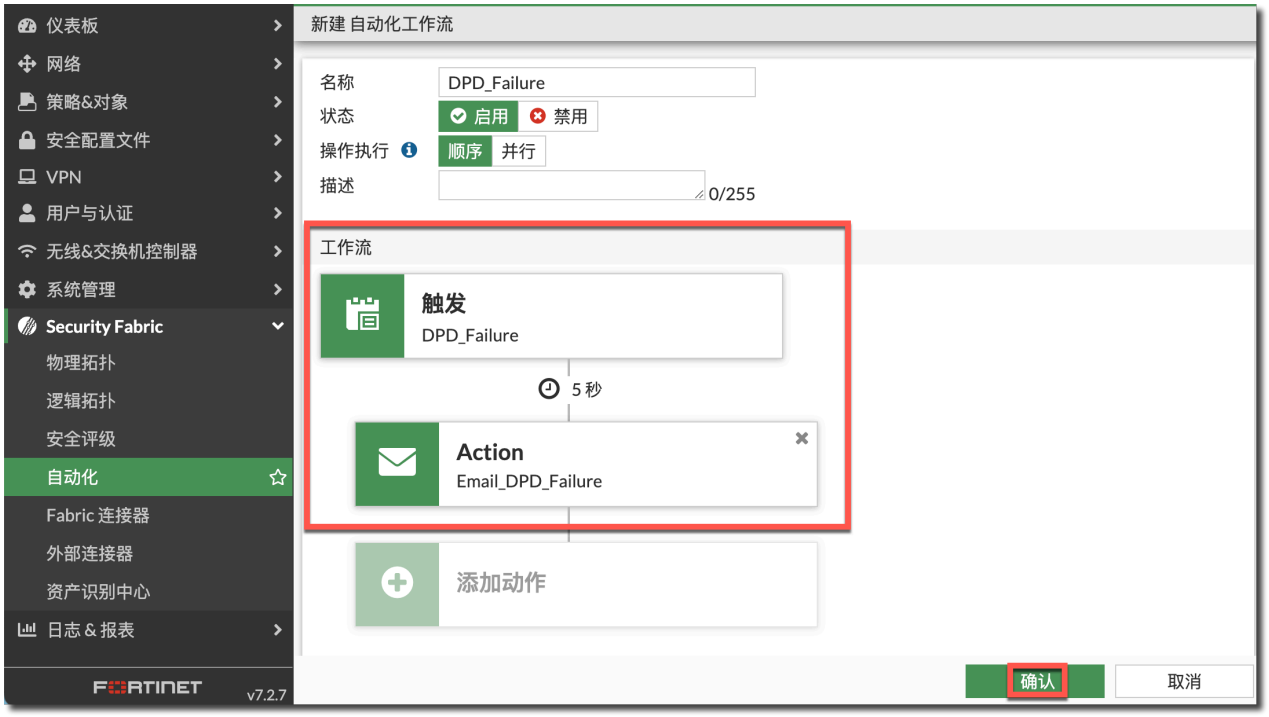
10. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



11. 可以点击“Add delay”添加触发器（Trigger）触发后执行动作（Action）间的延迟，这里以 5s 为例。



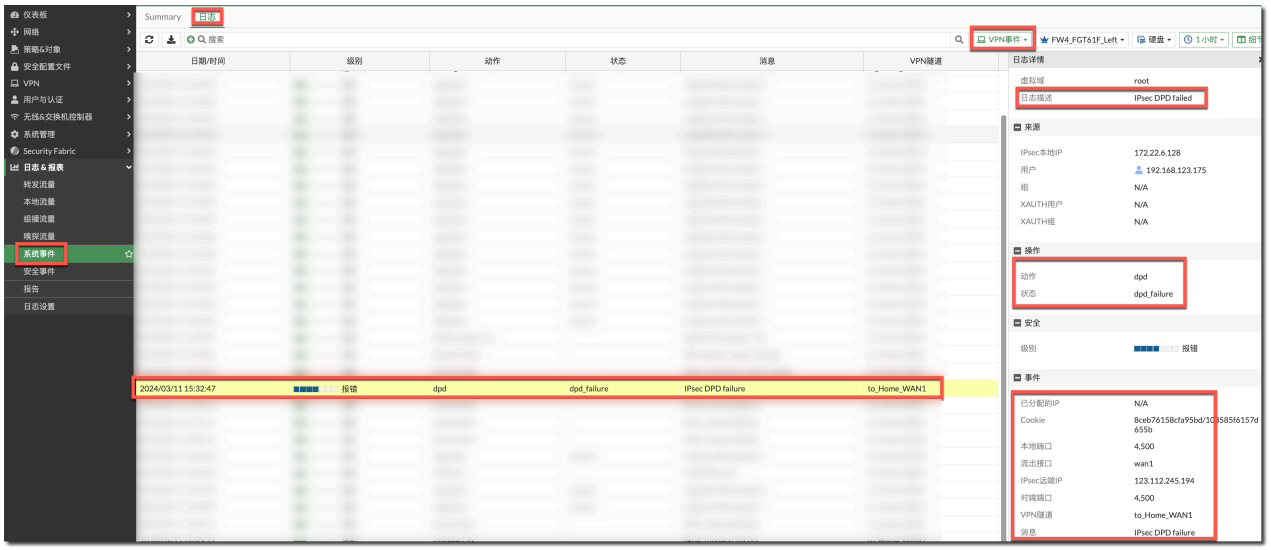
12. 检查无误后，点击确认下发配置。




```bash
config system automation-stitch
    edit "DPD_Failure"
        set trigger "DPD_Failure"
        config actions
            edit 1
                set action "Email_DPD_Failure"
                set delay 5
                set required enable
            next
        end
    next
end
```

配置验证

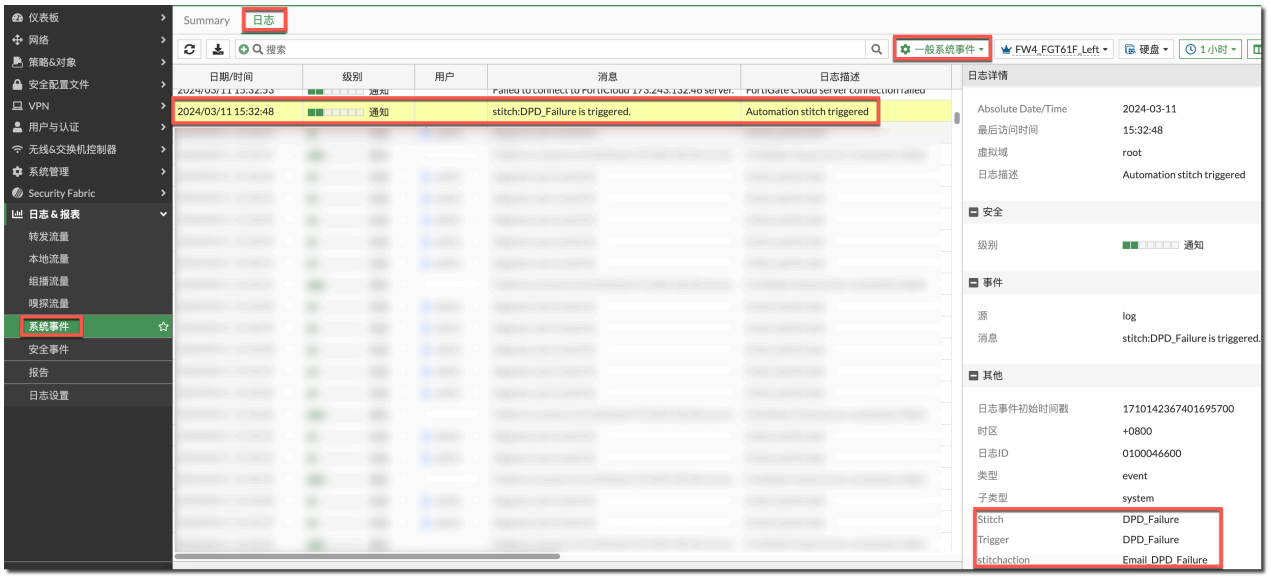
1. 触发 FortiGate 上的 DPD 超时日志。




```bash
date=2024-03-11 time=15:32:47 eventtime=1710142366681265100 tz="+0800" logid="0101037136" type="event" subtype="vpn"
level="error" vd="root" logdesc="IPsec DPD failed" msg="IPsec DPD failure" action="dpd" remip=123.112.245.194
locip=172.22.6.128 remport=4500 locport=4500 outintf="wan1" cookies="8ceb76158cfa95bd/108585f6157d655b"
user="192.168.123.175" group="N/A" useralt="N/A" xauthuser="N/A" xauthgroup="N/A" assignip=N/A vpntunnel="to_Home_WAN1"
```

```
status="dpd_failure" advpnsc=0
```

2. 5s 后，触发自动化程序，相关自动化日志如下。




```bash
date=2024-03-11 time=15:32:48 eventtime=1710142367401695780 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="DPD_Failure" trigger="DPD_Failure"
```

3. 收件人邮箱收到了告警邮件，邮件内容为 DPD Failed 的日志内容，发件人为默认的 DoNotReply@notification.fortinet.net ↗。



4. 对应的自动化触发 Debug 信息。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable
......
__action_email_hdl()-181: email action (Email_DPD_Failure) is called.
from:
to:bbai@fortinet.com;
subject:DPD_Failure_Notice
......
```

5. 发送邮件的 Debug 信息如下。

```bash
FortiGate # diagnose debug application alertmail -1
FortiGate # diagnose debug enable

Arrived msg(type 9, 415 bytes):bbai@fortinet.com;
DPD_Failure_Notice
date=2024-03-11 time=16:11:15 devid="FGT61FTK22063594" devname="FortiGate" eventtime=1710144674391662240 tz="+0800"
logid="0100032014" type="event" subtype="system" level="warning" vd="root" logdesc="Support license expiring"
ui="automation" action="test" msg="Automation Stitch Test: FortiCare license will expire in 1 day(s)" stitch="License
Expired Notification"

mail_info:
        from:notification.fortinet.net  user:DoNotReply@notification.fortinet.net
mail_info:
```

```
        reverse path:DoNotReply@notification.fortinet.net
        user name:DoNotReply
to[0]:bbai@fortinet.com
⟸_init_mail_info
create session
resolve notification.fortinet.net to 1 IP
⟹ send mail
connecting to 208.91.114.151 port 465
send mail 0×9988160 session 0×99863c0
session_io_event: creating ssl structure for session 0×99863c0
create_ssl: 0×7f9ff88000
sessionn 0×99863c0, SSL connected
session: 0×99863c0, rsp_state: greeting, code: 220
session: 0×99863c0, rsp_state: ehlo, code: 250
session: 0×99863c0, rsp_state: mail, code: 250
session: 0×99863c0, rsp_state: rcpt, code: 250
session: 0×99863c0, rsp_state: data, code: 354
⟸ send: date=2024-03-11 time=16:11:15 devid="FGT61FTK22063594" devname="FortiGate" eventtime=1710144674391662240
tz="+0800" logid="0100032014" type="event" subtype="system" level="warning" vd="root" logdesc="Support license expiring"
ui="automation" action="test" msg="Automation Stitch Test: FortiCare license will expire in 1 day(s)" stitch="License
Expired Notification"

session: 0×99863c0, rsp_state: data2, code: 250
session: 0×99863c0, rsp_state: quit, code: 221
session finined
_session_on_destroy
⟸ send mail success, m = 0×9988160 s = 0×99863c0
```

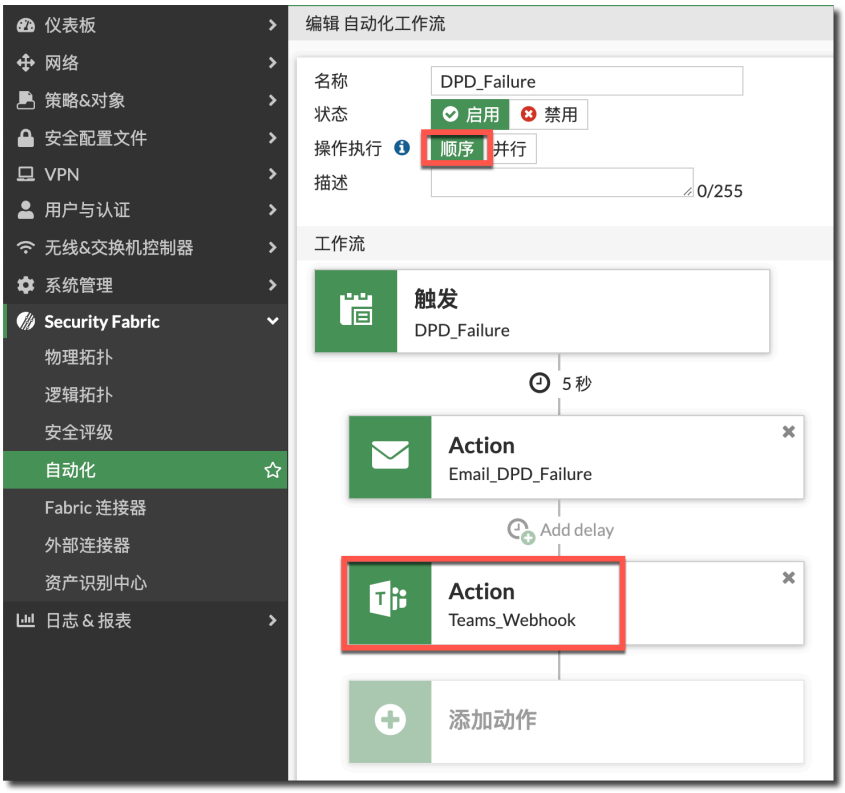
多动作顺序执行示例

网络需求

当 IPSec 由于 DPD 超时中断时，等待 5s 后向管理员的邮箱发送通知邮件，邮件发送成功后，再通过 Webhook 向 Teams 发送消息。

配置步骤

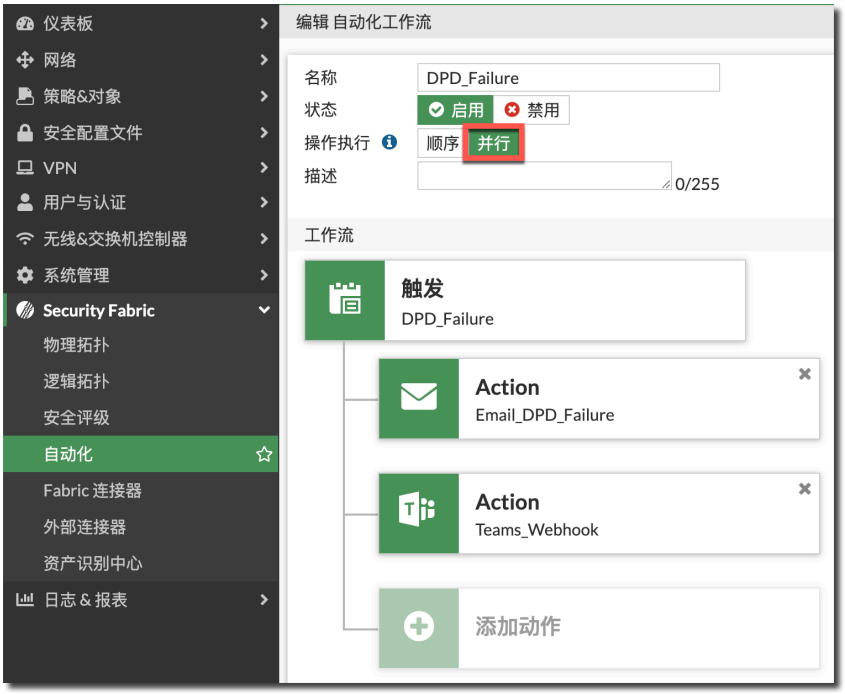
1. 在以上配置的基础上，使该自动化工作流以顺序模式执行多个动作。新建一个动作（Action），这里以 Teams Webhook 为例。



2. 配置下发后，查看自动化工作流对应的 CLI 配置，可以看到 action 部分多了 `set required enable` 字段，该字段表示必须前一个 Action 执行完毕后，才会执行本 Action，这也就实现了“顺序”的效果。

```bash
config system automation-stitch
    edit "DPD_Failure"
        set trigger "DPD_Failure"
        config actions
            edit 1
                set action "Email_DPD_Failure"
                set delay 5
                set required enable
            next
            edit 2
                set action "Teams_Webhook"
                set required enable
            next
        end
    next
end
```

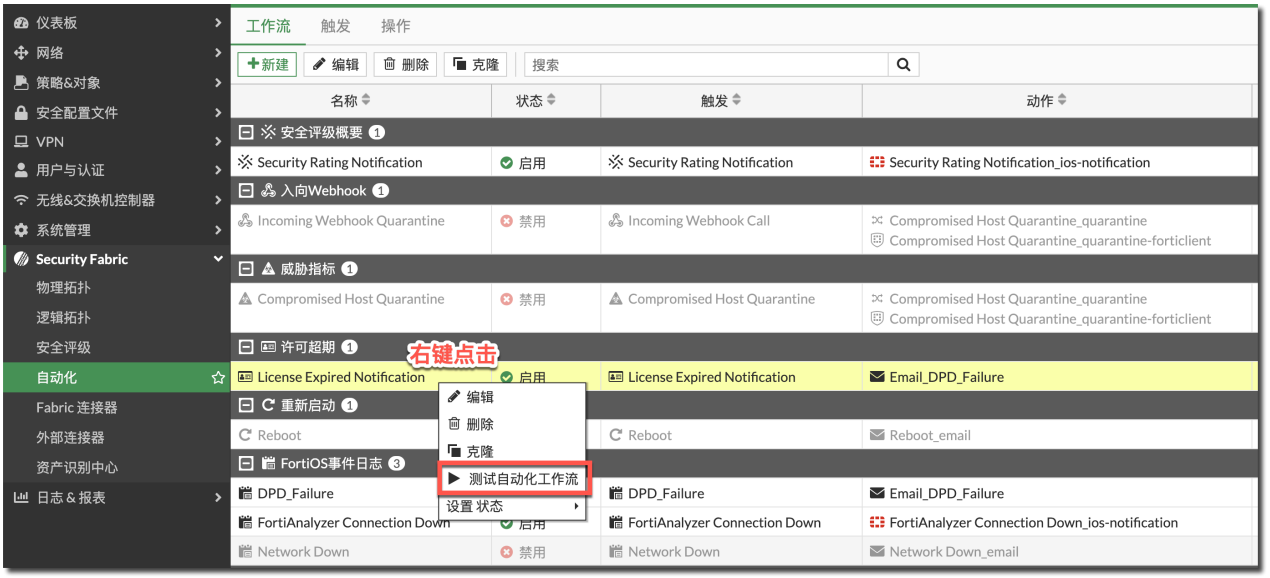
3. 将该自动化工作流改为并行模式，查看 CLI 配置，可以看到 `set required enable` 配置消失。只要触发器被触发，所有的动作会立即执行。




```bash
config system automation-stitch
    edit "DPD_Failure"
        set trigger "DPD_Failure"
        config actions
            edit 1
                set action "Email_DPD_Failure"
            next
            edit 2
                set action "Teams_Webhook"
            next
        end
    next
end
```

手动测试自动化

1. 部分自动化程序可以手动触发，作为配置完成后的测试手段。如下所示，在工作流页面中，右键点击某个自动化工作流，点击“测试自动化工作流程”按钮。



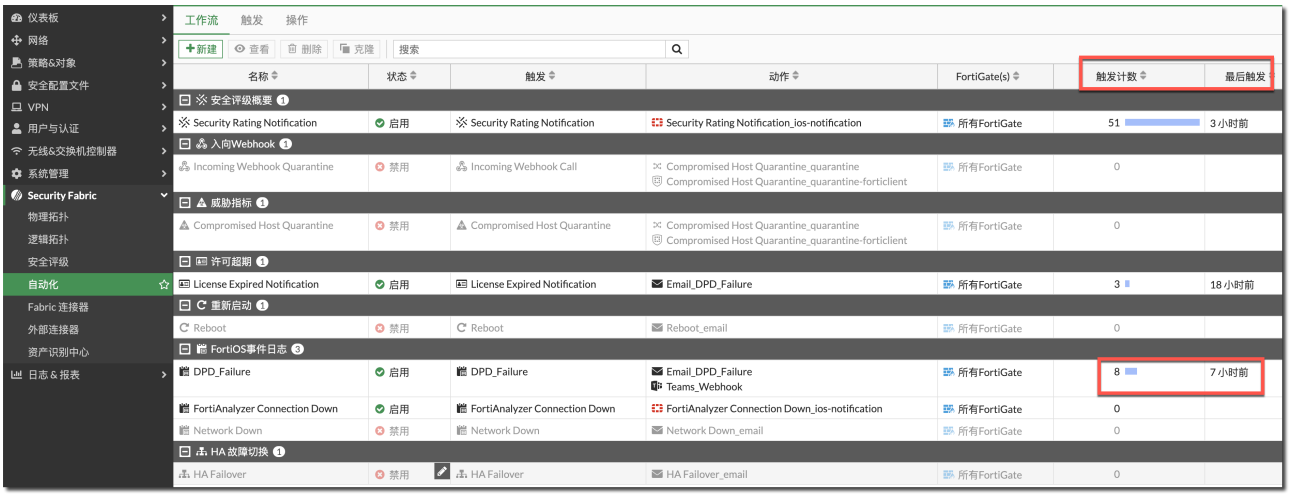
2. 此动作可以不需要配置的触发器真实触发，而直接触发执行动作。通过 CLI `diagnose automation test <Stitch_name>` 也可以执行该动作。

```bash
FortiGate # diagnose automation test "License Expired Notification"
automation test is done. stitch:License Expired Notification
```

查看自动化状态信息

GUI

在工作流页面可以直接查看每个工作流的触发次数和最后触发时间。



CLI

1. 通过 `diagnose test application autod 2` 可以看到所有的自动化工作流的配置与匹配计数。

```bash
FortiGate # diagnose test application autod 2
csf: disabled   root: no        sync connection: connecting
version:0 sync time:
total stitches activated: 4

stitch: DPD_Failure
        destinations: all
        trigger: DPD_Failure
                type:logid
                        logids:
                                0: 37136
        local hit: 6 relayed to: 0 relayed from: 0
```

```
        actions:
                Email_DPD_Failure type:email interval:0
                        delay:5 required:yes
                        subject: DPD_Failure_Notice
                        body: %%log%%
                        sender:
                        mailto:bbai@fortinet.com;
......
```

2. 查看自动化工作流的触发次数、最后触发时间、被自动化引用的日志 ID 等统计。

```
FortiGate # diagnose test application autod 3
alert mail log count: 0

stitch: DPD_Failure

        local hit: 7 relayed to: 0 relayed from: 0
        last trigger:Mon Mar 11 16:00:45 2024
        last relay:
        actions:
                Email_DPD_Failure:
                        done: 6 relayed to: 0 relayed from: 0
                        last trigger:Mon Mar 11 16:00:45 2024
                        last relay:

                          ......

logid to stitch mapping:
id:20101  local hit: 0 relayed hits: 0
        License Expired Notification

id:20102  local hit: 0 relayed hits: 0
        License Expired Notification

id:20103  local hit: 0 relayed hits: 0
        License Expired Notification

id:20104  local hit: 0 relayed hits: 0
        License Expired Notification

id:22902  local hit: 0 relayed hits: 0
        FortiAnalyzer Connection Down

id:32014  local hit: 2 relayed hits: 0
        License Expired Notification

id:32045  local hit: 0 relayed hits: 0
        License Expired Notification

id:32049  local hit: 0 relayed hits: 0
        License Expired Notification

id:37136  local hit: 7 relayed hits: 0
        DPD_Failure

id:52000  local hit: 47 relayed hits: 0
        Security Rating Notification

log category to stitch mapping:
```

3. 查看正在运行的自动化工作流状态，以及已运行的自动化工作流的执行统计。

```
FortiGate # diagnose test application autod 5

runned stitches: total:56 done:55 drop:0

running stitches: 1

1. stitch:DPD_Failure id:55 running time:Mon Mar 11 16:00:45 2024

        log relayed:0 sn:FGT61FTK22063594 logid:37136 log(531):date=2024-03-11 time=16:00:45 devid="FGT61FTK22063594"
```

```
devname="FortiGate" eventtime=1710144045841282220 tz="+0800" logid="0101037136" type="event" subtype="vpn" level="error"
vd="root" logdesc="IPsec DPD failed" msg="IPsec DPD failure" action="dpd" remip=123.112.245.97 locip=172.22.6.128
remport=4500 locport=4500 outintf="wan1" cookies="aa470ef86bf5705d/9d33378890990748" user="192.168.123.175" group="N/A"
useralt="N/A" xauthuser="N/A" xauthgroup="N/A" assignip=N/A vpntunnel="to_Home_WAN1" status="dpd_failure" advpnsc=0

        1. action:Email_DPD_Failure status:ready id:0 running time:Mon Mar 11 16:09:05 2024
 delay:500 required:1
```

常用触发器

- 系统事件日志触发器
- UTM 日志触发器
- 证书过期触发器
- 计划时间触发器

系统事件日志触发器

简介

FortiGate 可以配置系统事件日志作为自动化的触发器（Trigger），在发生特定系统事件日志 ID 时触发。配置时可以选择多个系统事件日志 ID，还可以自定义日志字段筛选器。

> **重要**
>
> 每个自动化工作流最多可以配置 16 个系统事件日志 ID 作为触发器（Trigger）。

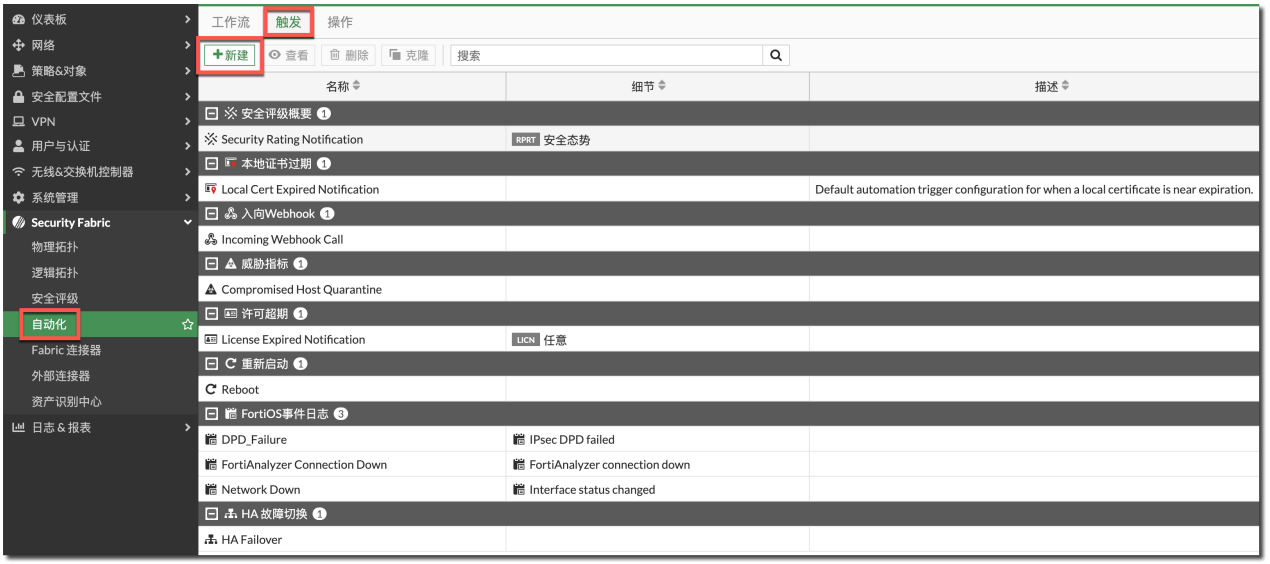
系统事件日志触发器可以从“Security Fabric → 自动化 → 触发”页面进行配置。也可以在“日志和报告 → 系统事件 → 日志”页面上右键点击某个日志进行快速配置。

网络需求

用户想使用管理员名称为 `fgt-user1`，且登录 IP 为 192.168.100.77 的管理员登录、登出日志作为触发器。

在 Security Fabric 中创建

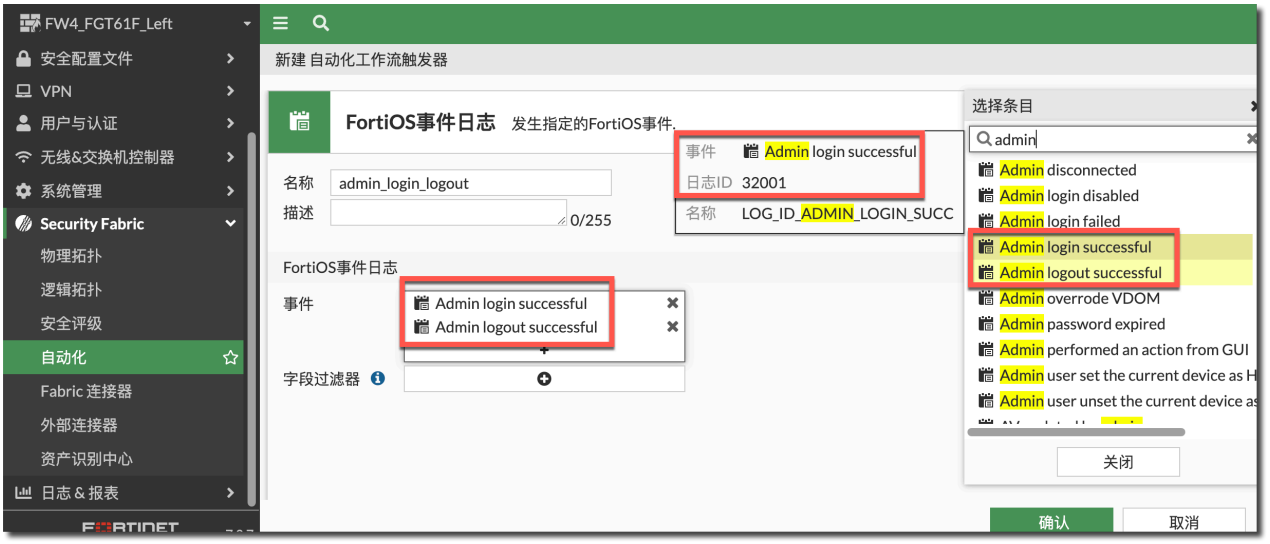
1. 进入 FortiGate 的“Security Fabric → 自动化 → 触发”页面，点击新建按钮。



2. 在“其他条件”分类中选择“FortiOS 事件日志”。



3. 在“事件”中选择“Admin login successful”（Log ID 32001）和“Admin logout successful”（Log ID 32003）两个日志类型。

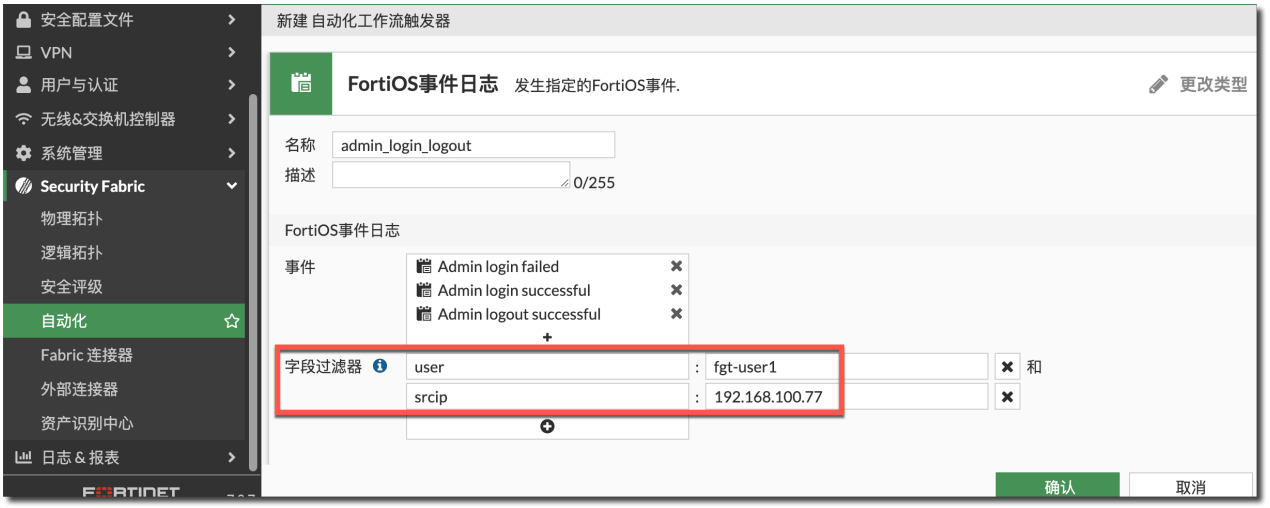


4. 在字段过滤器中，指定第一个日志字段 `user` 的值为 `fgt-user1`，指定第二个日志字段 `srcip` 的值为 `192.168.100.77`，点击确认下发配置。

重要

日志字段可以从相关日志文本中获取，根据需求进行配置。

```bash
date=2024-03-11 time=17:03:18 eventtime=1710147798720951460 tz="+0800" logid="0100032003" type="event" subtype="system"
level="information" vd="root" logdesc="Admin logout successful" sn="1710147790" user="fgt-user1" ui="https(192.168.100.77)"
method="https" srcip=192.168.100.77 dstip=10.10.12.1 action="logout" status="success" duration=8 state="Config-Changed"
reason="exit" msg="Administrator fgt-user logged out from https(192.168.100.77)"
```




```bash
config system automation-trigger
    edit "admin_login_logout"
        set event-type event-log
        set logid 32002 32001 32003
        config fields
            edit 1
                set name "user"
                set value "fgt-user1"
            next
            edit 2
                set name "srcip"
                set value "192.168.100.77"
            next
        end
    next
end
```

5. 参考 Security Fabric → 自动化 → 工作流 ↗章节继续配置对应的 Action 和 Stitch。

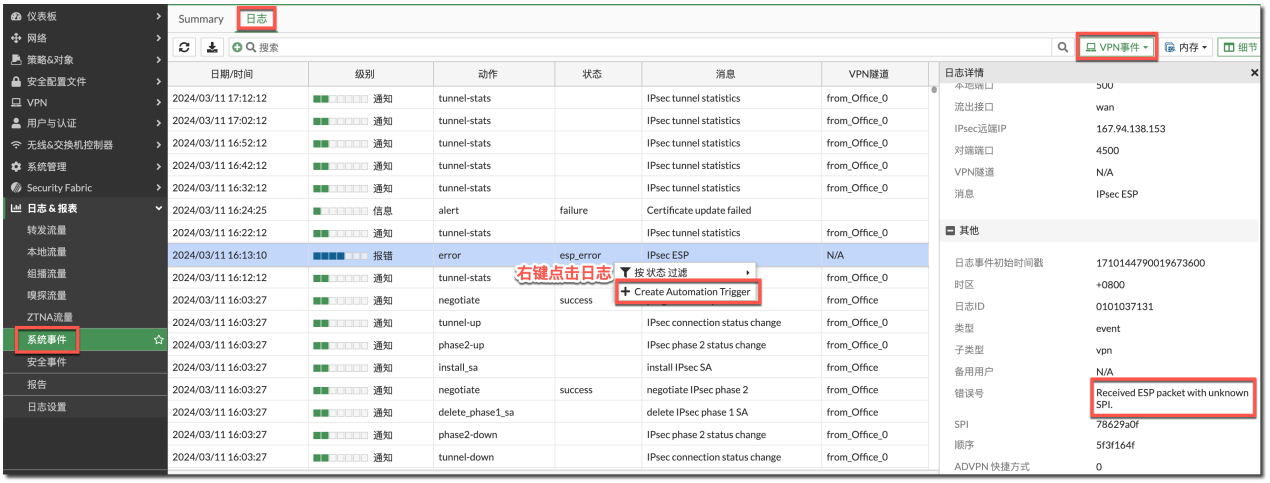
在日志中快捷创建

重要

该功能在 7.4.0 及更新版本支持。

需求：用户在浏览 VPN 事件日志时，发现有 `Received ESP packet with unknown SPI.` 的错误日志，直接通过正在浏览的该日志创建自动化的触发器。

1. 进入“日志 & 报表 → 系统事件 → 日志”页面，找到需要创建自动化触发器的日志，右键点击该日志，并选择“Create Automation Trigger”按钮。



2. 在弹出的新建“自动化工作流触发器”窗口中，配置触发器的名称，可自定义字段过滤器，点击确认即可下发该触发器。




```bash
config system automation-trigger
    edit "ESP_unknown_SPI"
        set event-type event-log
        set logid 37131
    next
end
```

3. 参考 Security Fabric → 自动化 → 工作流 ↗章节继续配置对应的 Action 和 Stitch。

UTM 日志触发器

> **重要**
>
> 该功能在 7.2.0 及以上版本支持。

简介

FortiGate 可以配置 UTM 日志作为自动化的触发器（Trigger），在发生特定 UTM 日志 ID 时触发。配置时可以选择多个 UTM 日志 ID，还可以自定义日志字段筛选器。UTM 日志触发器可以从“Security Fabric → 自动化 → 触发”页面进行配置。

> **重要**
>
> 每个自动化工作流最多可以配置 16 个 UTM 日志 ID 作为触发器（Trigger）。

UTM 日志触发器支持以下安全事件日志，且可以指定基于 VDOM 的安全事件日志：

- 异常日志（Anomaly）
- 入侵防御日志（IPS）
- SSH 日志
- 反病毒日志（AV）
- 流量阻断日志（Traffic violations，被防火墙策略阻断的流量日志）
- Web 过滤（Web Filter）

CLI 配置方法

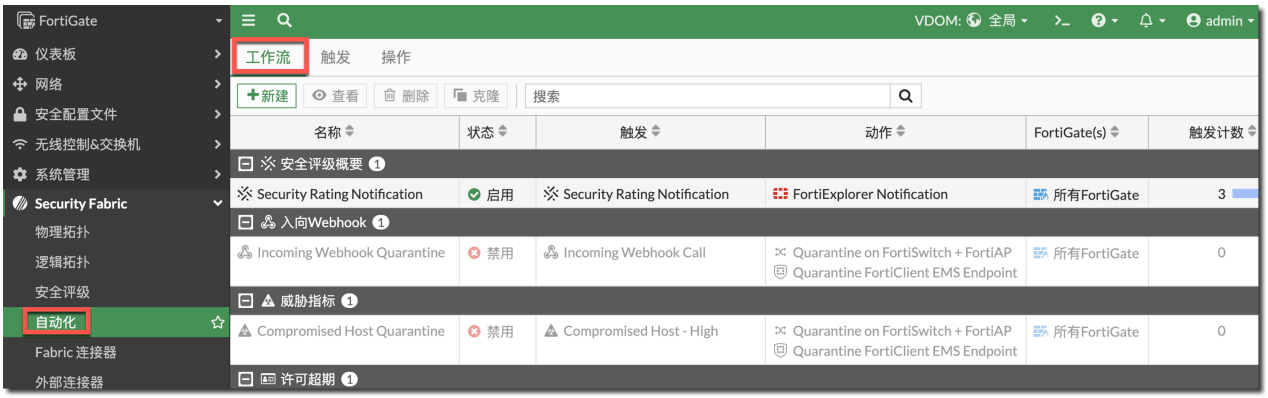
```bash
config system automation-trigger
    edit "Anomaly Logs"
        set trigger-type event-based
        set event-type anomaly-logs
        set vdom <name>
    next
    edit "IPS Logs"
        set trigger-type event-based
        set event-type ips-logs
        set vdom <name>
    next
    edit "SSH Logs"
        set trigger-type event-based
        set event-type ssh-logs
        set vdom <name>
    next
    edit "Traffic Violation"
        set trigger-type event-based
        set event-type traffic-violation
        set vdom <name>
    next
    edit "Virus Logs"
        set trigger-type event-based
        set event-type virus-logs
        set vdom <name>
    next
    edit "Webfilter Violation"
        set trigger-type event-based
        set event-type webfilter-violation
        set vdom <name>
    next
end
```

网络需求

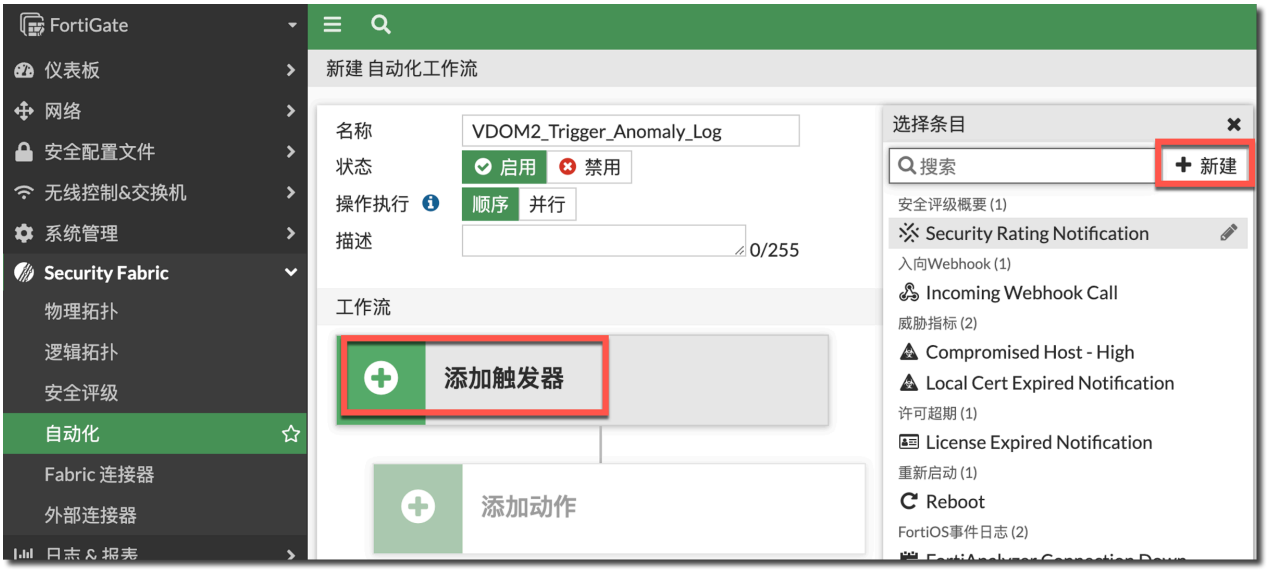
FortiGate 的 VDOM2 在遭受 DoS 攻击后，自动发送邮件到指定邮箱。

配置举例

1. 进入 FortiGate 全局 VDOM 的 Security Fabric → 自动化页面，新建自动化工作流。



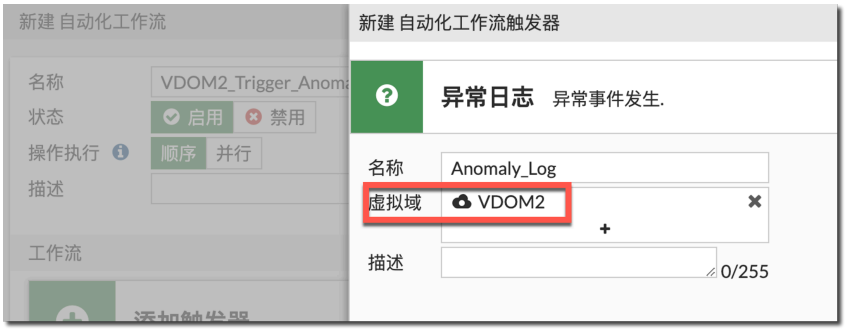
2. 点击添加触发器按钮，在弹出的条目中选择新建。



3. 在弹出的日志类型列表中，选择 Event Log 分类的异常日志。

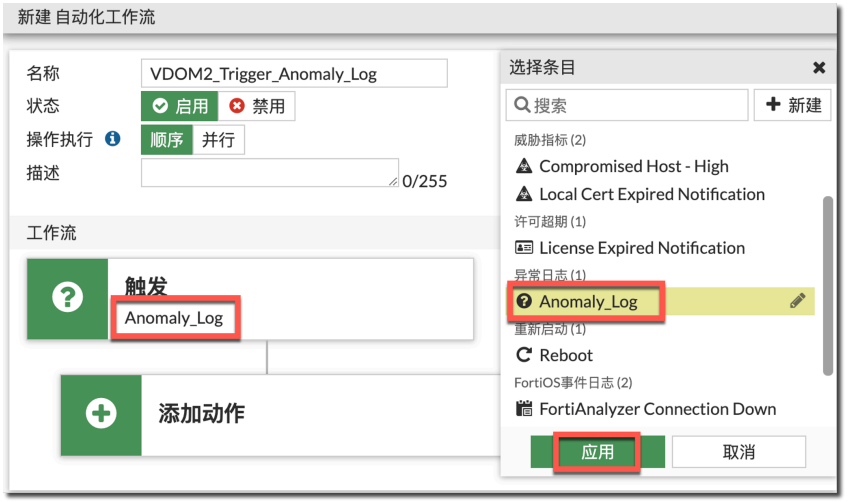


4. 配置触发器名称，并配置 VDOM 作为触发条件，这里以 VDOM2 为例，也可以选择多个 VDOM，下发配置。

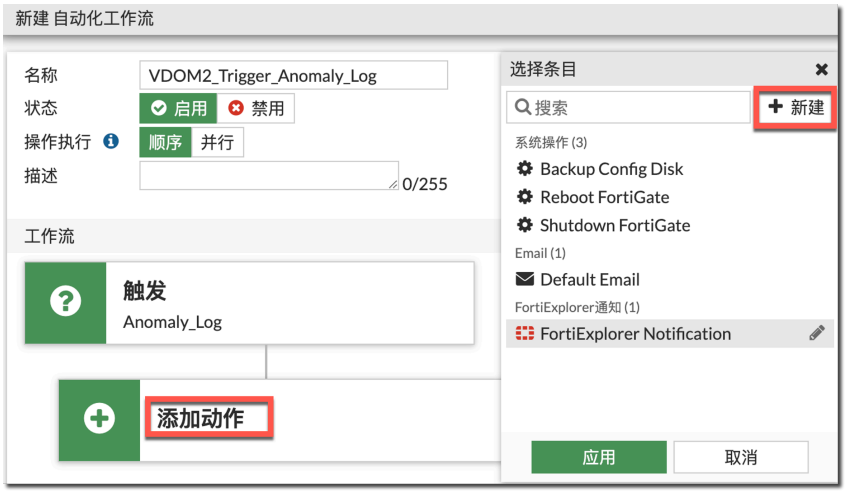



```bash
config system global
config system automation-trigger
    edit "Anomaly_Log"
        set event-type anomaly-logs
        set vdom "VDOM2"
    next
end
```

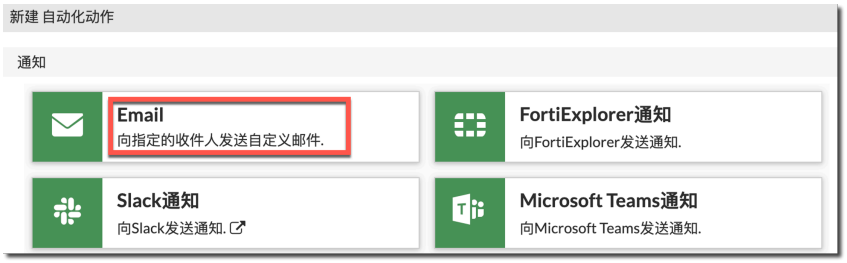
5. 选择上步创建的触发器作为触发条件，也可以创建并选择多个触发器。



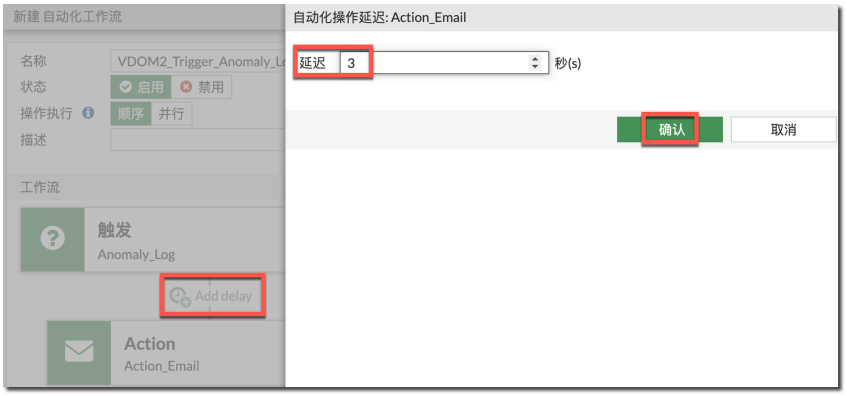
6. 点击添加动作按钮，并在弹出的条目窗口中点击新建按钮。



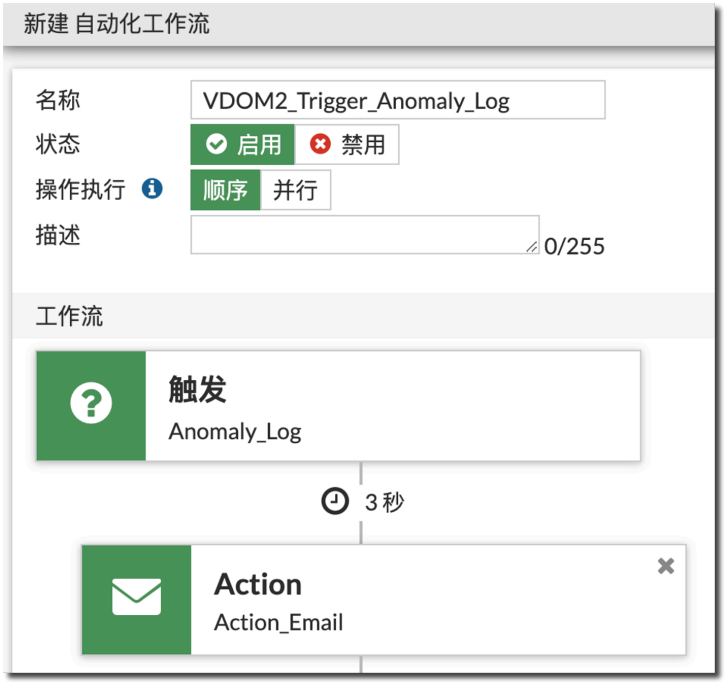
7. 在通知分类中选择 Email 类型的动作。



8. 添加触发器与动作之间的延迟为 3s，点击确认下发工作流配置。




```bash
config system global
config system automation-action
    edit "Action_Email"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-from "DoNotReply@notification.fortinet.net"
        set email-subject "CSF stitch alert"
        set replacement-message enable
    next
end
```

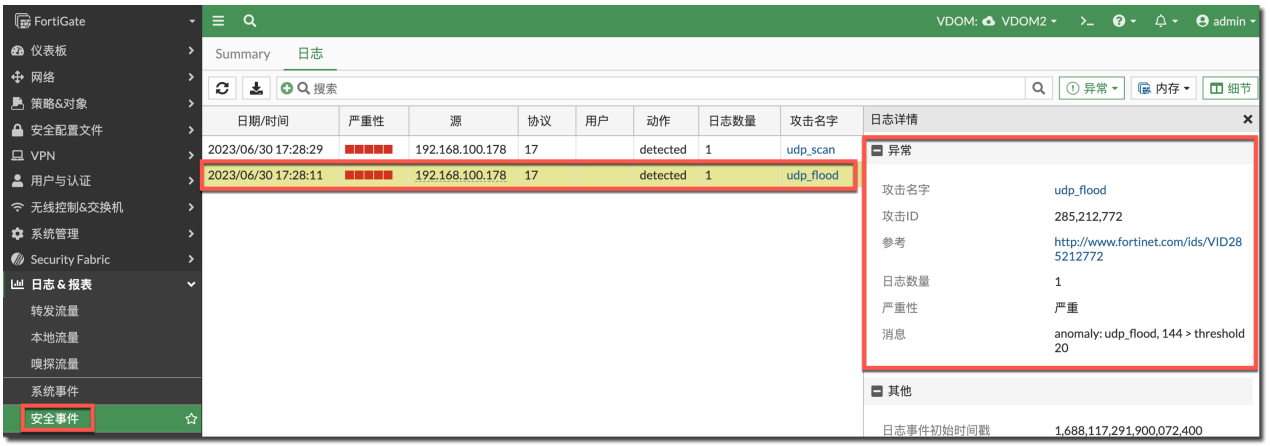



```bash
config system global
config system automation-stitch
    edit "VDOM2_Trigger_Anomaly_Log"
        set trigger "Anomaly_Log"
        config actions
            edit 1
                set action "Action_Email"
```

```
                set delay 3
                set required enable
            next
        end
    next
end
```

配置验证

1. 在 FortiGate 的 VDOM2 中配置 DoS 策略，并开启异常日志记录，从终端对 FortiGate 的 VDOM2 接口 port2 发起 DoS 攻击，在日志 & 报表 → 安全事件中的异常分类中，可以看到产生了 UDP Flood 日志。



2. 3s 后收件人邮箱收到了 CSF 自动化发送的邮件，邮件内容为异常日志的文本。



3. 查看 autod 进程的 debug 信息，可以看到 Email 动作被触发。

```bash
FortiGate (global) # diagnose debug application autod -1
FortiGate (global) # diagnose debug enable
......
__action_email_hdl()-181: email action (Action_Email) is called.
from:DoNotReply@notification.fortinet.net
to:bbai@fortinet.com;
subject:CSF stitch alert
......
```

4. 查看该自动化流程的的配置和触发统计。

```bash
FortiGate (global) # diagnose test application autod 2
csf: disabled   root: no        sync connection: connecting
version:0 sync time:
total stitches activated: 4
stitch: VDOM2_Trigger_Anomaly_Log
        destinations: all
        trigger: Anomaly_Log
                type:anomaly logs
                field ids:
```

```
                (id:6)vd=VDOM2
        local hit: 6 relayed to: 0 relayed from: 0
        actions:
                Action_Email type:email interval:0
                        delay:3 required:yes
                        subject: CSF stitch alert
                        body: %%log%%
                        sender: DoNotReply@notification.fortinet.net
                        mailto:bbai@fortinet.com;
```

证书过期触发器

简介

证书过期触发器（Certificate expiration trigger）可以在本地证书即将过期时触发自动化流程。可配置证书到期前（自定义天数 0~100，默认 14 天）触发。

> **重要**
>
> 只针对本地证书分类中的证书（如用户证书或带私钥的 Sub CA）生效。

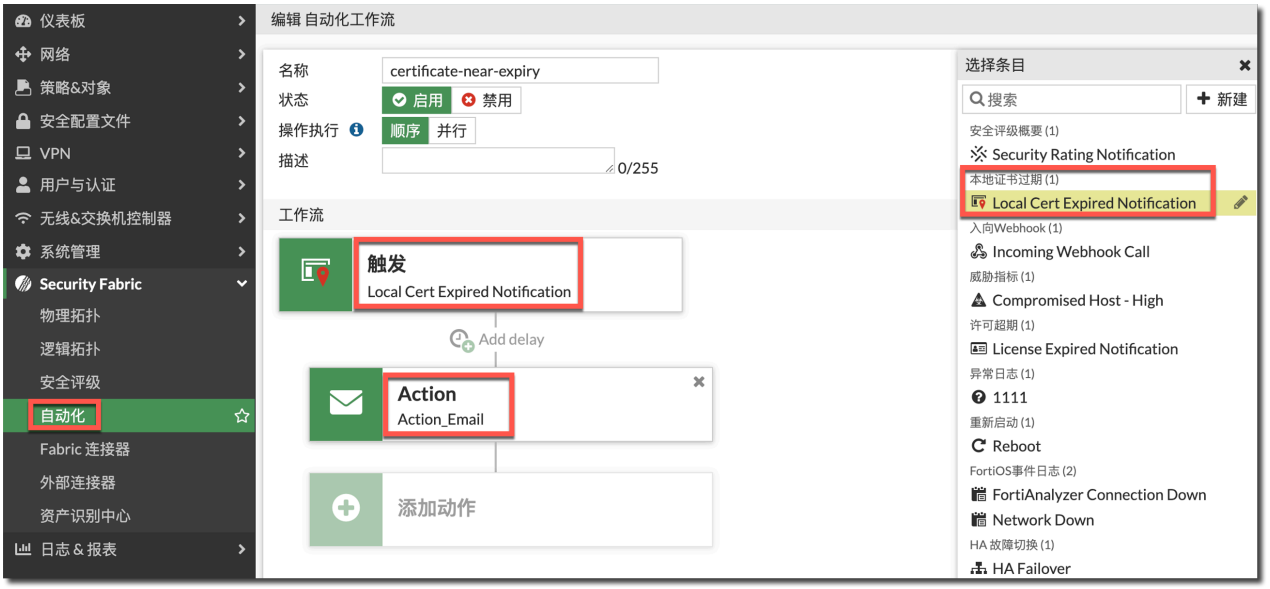
```bash
config vpn certificate setting
    set cert-expire-warning <integer>    //设置本地证书到期前触发告警的天数，范围0～100天，默认14天//
end
```

网络需求

当设备的本地证书的到期时间前 1 天，触发自动化流程，将本地证书即将过期的系统事件日志发送至指定邮箱。

配置步骤

1. 在 Security Fabric → 自动化中新建自动化工作流，触发条件选择预置的 Local Certain Expired Notification（也可以手动创建该触发器），动作选择发送邮件至指定邮箱。




```bash
config system automation-trigger
    edit "Local Cert Expired Notification"
        set event-type local-cert-near-expiry
    next
end

config system automation-action
    edit "Action_Email"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-from "DoNotReply@notification.fortinet.net"
        set email-subject "CSF stitch alert"
        set replacement-message enable
    next
end

config system automation-stitch
    edit "certificate-near-expiry"
```

```
        set trigger "Local Cert Expired Notification"
        config actions
            edit 1
                set action "Action_Email"
                set required enable
            next
        end
    next
end
```

2. 配置证书过期提醒为过期前 1 天。

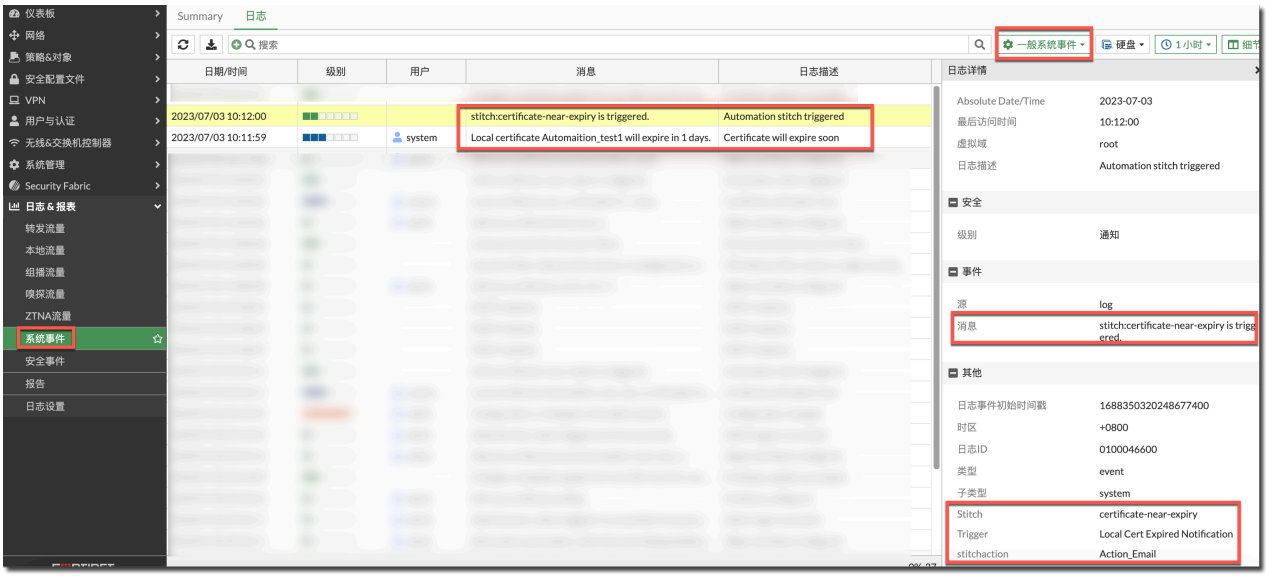
```bash
config vpn certificate setting
    set cert-expire-warning 1
end
```

3. 在 FortiGate 查看即将过期的用户证书、SubCA 证书、CA 证书（剩余有效期 2 天）。

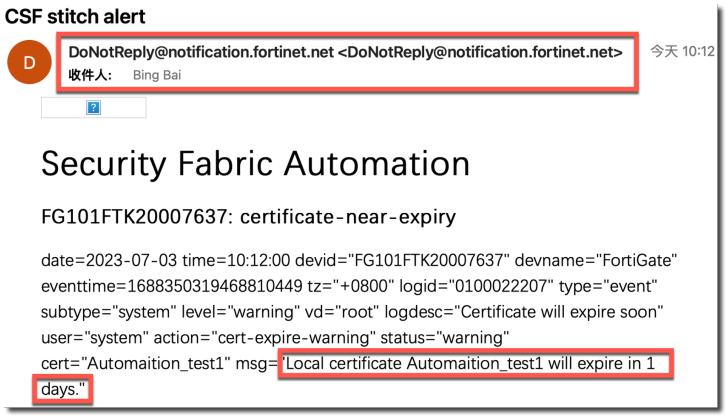


结果验证

1. 证书即将过期的触发日志如下，可以看到证书的过期倒数时间＜2 天后，就会产生证书即将过期的日志，并触发配置的自动化流程。



2. 收到的邮件内容如下，只有本地证书（用户证书、Sub CA）可以触发该自动化，CA 证书不会触发。





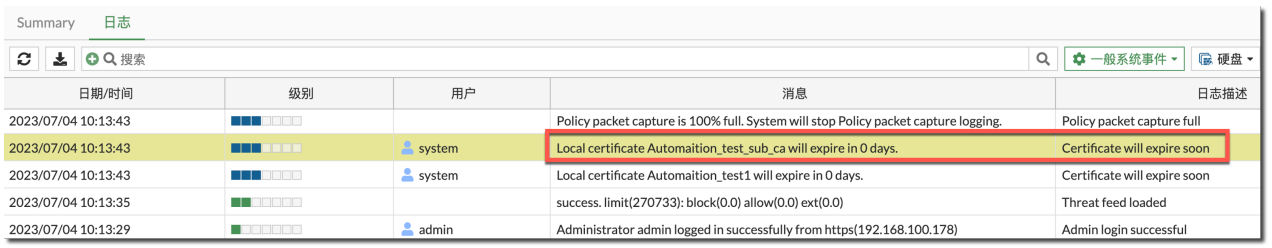
3. 查看 autod 进程的 debug 信息，可以看到 Email 动作被触发。

```bash
FortiGate (global) # diagnose debug application autod -1
__action_email_hdl()-181: email action (Action_Email) is called.
from:DoNotReply@notification.fortinet.net
to:bbai@fortinet.com;
subject:CSF stitch alert
```

4. 查看该自动化流程的的配置和触发统计（使用 2 可以看统计和配置）。

```bash
FortiGate (global) # diagnose test application autod 3
stitch: certificate-near-expiry

        local hit: 3 relayed to: 0 relayed from: 0
        last trigger:Mon Jul  3 10:12:00 2023
        last relay:
        actions:
                Action_Email:
                        done: 3 relayed to: 0 relayed from: 0
                        last trigger:Mon Jul  3 10:12:00 2023
                        last relay:
log category to stitch mapping:
id:22207 local hit: 3 relayed hits: 0
        certificate-near-expiry
```

5. 当证书的剩余时间小于 1 天时，产生的证书即将过期日志中，剩余时间为 0 天。




CSF stitch alert
D
DoNotReply@notification.fortinet.net <DoNotReply@notification.fortinet.net>
收件人: Bing Bai
为了保护你的隐私，未下载此邮件中的某些外部图像。
Security Fabric Automation
FG101FTK20007637: certificate-near-expiry
date=2023-07-04 time=10:13:44 devid="FG101FTK20007637" devname="FortiGate"
eventtime=1688436823524907520 tz="+0800" logid="0100022207" type="event"
subtype="system" level="warning" vd="root" logdesc="Certificate will expire soon"
user="system" action="cert-expire-warning" status="warning"
cert="Automaition_test1" msg="Local certificate Automaition_test1 will expire in 0
days."

计划时间触发器

计划时间触发器（Schedule trigger）可以按照计划时间来触发自动化流程，支持的计划类型如下：

- 每小时（hourly）：指定在每小时的特定分钟循环触发。
- 每天（daily）：指定在每天的特定小时和分钟循环触发。
- 每周（weekly）：指定在每周的特定某天的小时和分钟循环触发。
- 每月（monthly）：指定在每月的特定某天的小时和分钟循环触发。
- 单次（once）：指定在特定的年/月/日/小时/分钟触发一次。

```bash
config system automation-trigger
    edit <name>
        set trigger-type scheduled
        set trigger-frequency {hourly | daily | weekly | monthly | once}
        set trigger-hour <inetger>
        set trigger-minute <integer>
        set trigger-weekday {sunday | monday | tuesday | wednesday | thursday | friday | saturday}
        set trigger-day <integer>
        set trigger-datetime <YYYY-MM-DD HH:MM:SS>
    next
end
```

网络需求

在每天的凌晨 3 点备份 FortiGate 的配置文件到 FTP 服务器，备份的配置文件名称中要包含备份的日期和时间。

配置步骤

1. 进入 FortiGate 的“Security Fabric → 自动化 → 触发器”页面，点击“新建”按钮。



2. 在弹出的页面的“其他条件”分类中点击“计划任务”。



3. 配置触发器名称，计划任务频率选择“每天”，小时填写“3”，分钟保持默认“0”（24 小时表示法），点击确认下发配置。






```bash
config system automation-trigger
    edit "Daily_3am"
        set trigger-type scheduled
        set trigger-hour 3
    next
end
```

4. 进入 FortiGate 的“Security Fabric → 自动化 → 操作”页面，点击“新建”按钮。



5. 在弹出的页面的“通用”分类中点击“CLI 脚本”。



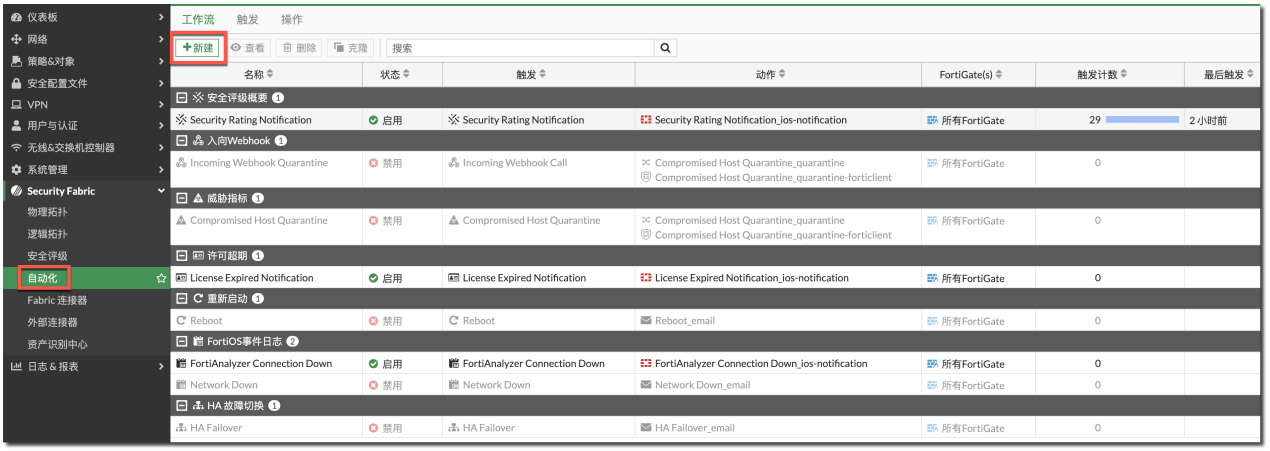
6. 配置动作名称，脚本中填写备份配置到 FTP 服务器的 CLI 命令，点击确认下发配置。



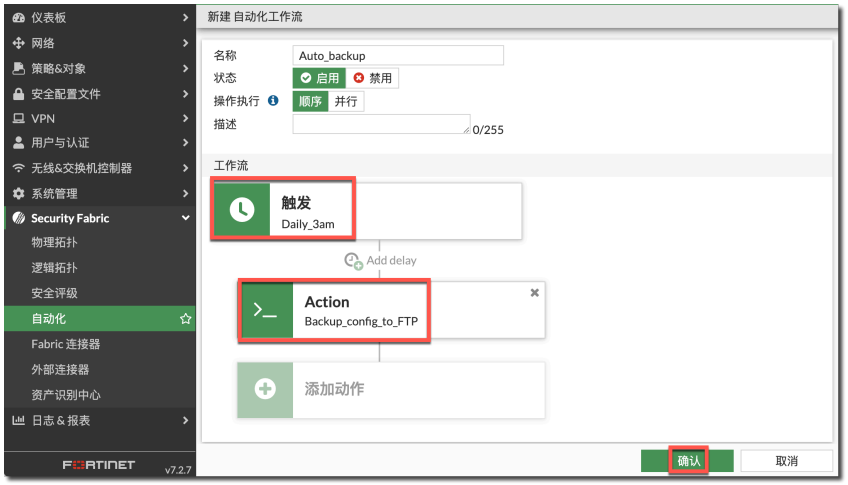

```bash
config system automation-action
    edit "Backup_config_to_FTP"
        set action-type cli-script
        set script "exe backup config ftp AutoBackup_%%date%%_%%time%%.conf 192.168.100.110 backup 111111"
        set accprofile "super_admin"
    next
end
```

- `AutoBackup_%%date%%_%%time%%.conf` 表示备份到 FTP 服务器的配置文件名称，其中 `%%date%%` 是一个变量，表示备份时的日期，`%%time%%` 为具体的时间（`小时:秒` 格式）。
- `192.168.100.110` 为 FTP 服务器的 IP，如果 FTP 服务器使用的是非 21 端口，则需要在 IP 后添加加 `:<port>`，如 `192.168.100.110:2121`。
- `backup`：FTP 用户名。
- `111111`：FTP 用户密码。

7. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击“新建”按钮。



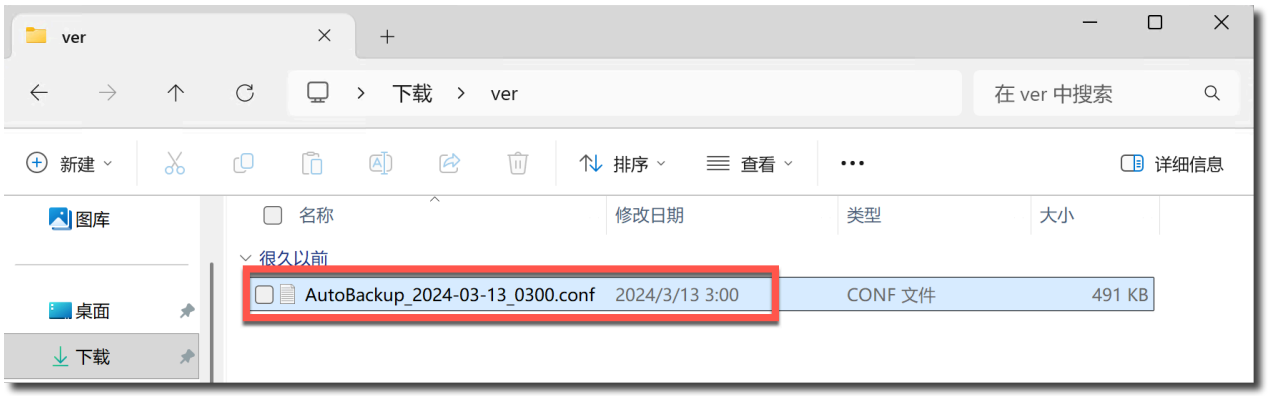
8. 配置工作流（Stitch 的）名称，在触发中选择步骤 3 创建的触发器，在 Action 中选择步骤 6 创建的动作，点击“确认”下发配置。




```bash
config system automation-stitch
    edit "Auto_backup"
        set trigger "Daily_3am"
        config actions
            edit 1
                set action "Backup_config_to_FTP"
                set required enable
            next
        end
    next
end
```

结果验证

1. 等待自动化流程被触发，检查 FortiGate 自动备份并上传到 FTP 服务器上的配置文件，可以看到文件名字与自动化 Action 中配置的 CLI 脚本的格式一致，包含备份日期。



2. 查看 FortiGate 的自动化触发日志，可以看到该自动化流程在凌晨 3 点被触发并完成执行对应动作。

```bash
date=2024-03-13 time=03:00:17 eventtime=1710316517425603720 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="Auto_backup" trigger="Daily_3am"
stitchaction="Backup_config_to_FTP" from="scheduled" msg="stitch:Auto_backup is triggered."

date=2024-03-13 time=03:00:22 eventtime=1710316522252196480 tz="+0800" logid="0100032549" type="event" subtype="system"
level="information" vd="root" logdesc="Autoscript stop automatically" msg="script autod.10 stopped automatically"
date=2024-03-13 time=15:55:22 eventtime=1710316522239782720 tz="+0800" logid="0100032142" type="event" subtype="system"
level="notice" vd="root" logdesc="System configuration backed up" ui="auto-script" action="backup" msg="User  backed up
the configuration from auto-script"
```

3. 查看自动化触发的 Debug 信息，可以看到整个触发和执行备份配置文件到 FTP 的全部过程。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable

__action_cli_script_open()-186: cli script action:Backup_config_to_FTP is called. svc ctx:0×a1cabf0
accprof:super_admin script:
exe backup config ftp AutoBackup_%%date%%_%%time%%.conf 192.168.100.110 backup 111111

pid:283-__handle_msg()-414: Subscriber:4 received package. pubid:2 pkgid:65017 pkg_index:0
pid:283-__handle_pkg_logs()-356: Subscriber:4 processing package size:1198 logs:1 pickup:1
auto_event_pub_handle_log()-331: publish 46600 to subscr-node
pid:283-__subscr_close_cur_pkg()-140: close package size:1198 logs:1
__cli_script_close()-121: cli script:
autod.10
 output:
########## script name: autod.10 ##########

========== #1, 2024-03-13 15:55:17 ==========
FW1_FGT101F $  exe backup config ftp AutoBackup_2024-03-13_0300.conf 192.168.100.110 backup 111111

Connect to ftp server 192.168.100.110 ...
Please wait...
Send config file to ftp server OK.

__action_cli_script_close()-213: cli script action is done. script:
exe backup config ftp AutoBackup_%%date%%_%%time%%.conf 192.168.100.110 backup 111111
output:
########## script name: autod.10 ##########

========== #1, 2024-03-13 15:55:17 ==========
FW1_FGT101F $  exe backup config ftp AutoBackup_2024-03-13_0300.conf 192.168.100.110 backup 111111

Connect to ftp server 192.168.100.110 ...
Please wait...
Send config file to ftp server OK.
```

4. 查看该自动化流程的状态统计信息。

```bash
FortiGate # diagnose test application autod 3
alert mail log count: 0

stitch: Auto_backup (scheduled)

        local hit: 1 relayed to: 0 relayed from: 0
        last trigger:Wed Mar 13 15:55:17 2024
        last relay:
        next scheduled trigger:Thu Mar 14 15:55:00 2024
        actions:
                Backup_config_to_FTP:
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Wed Mar 13 03:00:17 2024
                        last relay:

logid to stitch mapping:
id:0 (scheduled stitches) local hit: 1 relayed hits: 0
        Auto_backup

log category to stitch mapping:
```

5. 附备份配置文件的 CLI 命令格式。

```bash
FTP:
execute backup config ftp <backup_filename> <ftp_server>[<:ftp_port>] [<user_name>] [<password>] [<backup_password>]

TFTP:
execute backup config tftp <backup_filename> <tftp_servers> [<backup_password>]

SFTP:
execute backup config sftp <backup_filename> <sftp_server>[<:sftp_port>] <user> <password> [<backup_password>]
```

常用动作

- 内存/CPU 达到阈值时收集信息
- Email 动作
- Teams Webhook
- 系统动作
- 自动抓包

内存/CPU 达到阈值时收集信息

简介

当 FortiGate CPU 利用率达到阈值或设备内存占用过高进入保护模式时，客户有可能为了快速恢复业务而重启设备，重启完成后，可以收集到的故障信息较少，难以定位。

自动化功能可以在 FortiGate CPU 利用率达到阈值或内存进入保护模式时，触发相关的 CLI 命令（CLI script action）来收集故障时的设备状态信息，以方便技术人员进行问题定位。

> **重要**
>
> CLI 类型的动作（Action）可以收集的信息输出结果最大支持 192000 个字符，超过 192000 个字符的内容会被截断。

CLI script action 配置

```bash
config system automation-action
    edit <name>
        set action-type cli-script
        set output-size <integer>
        set timeout <integer>
    next
end
```

- `output-size <integer>`：最大输出大小。单位为 megabytes，范围 1 ~ 1024，默认为 10。
- `timeout <integer>`：执行 CLI 的超时时间，超过这个时间 CLI 未执行完成，则结束 action。单位为秒，范围 0 ~ 300，默认为 0，表示持续等待 CLI 执行完成。

内存保护模式

网络需求

当 FortiGate 进入内存保护模式后，触发 CLI 动作收集设备当前的运行信息，并将信息发送至指定邮箱。

配置步骤

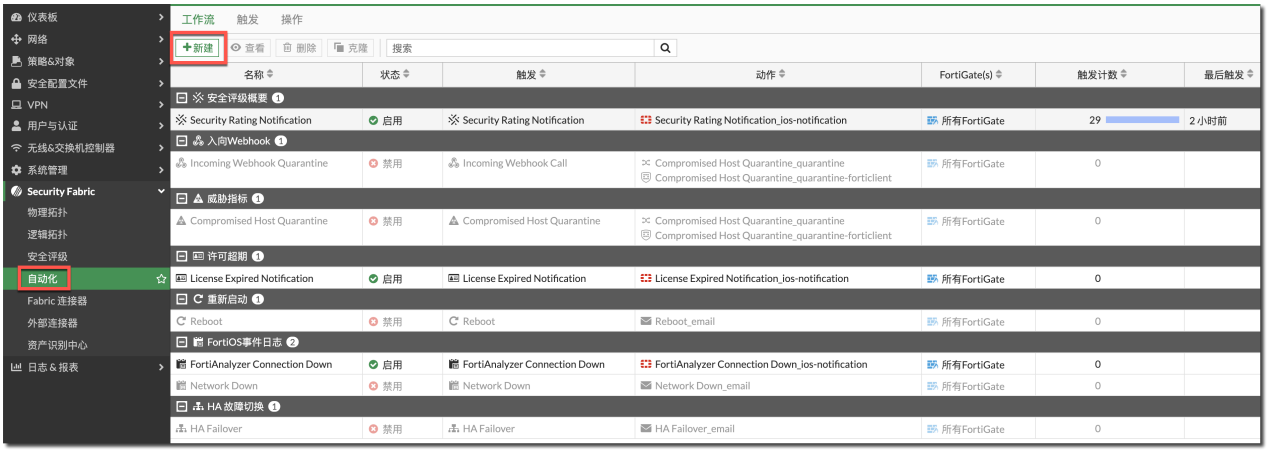
1. 内存保护模式阈值配置。

```bash
config system global
    set memory-use-threshold-extreme <percent>
    set memory-use-threshold-green <percent>
    set memory-use-threshold-red <percent>
end
```

   - `memory-use-threshold-red`：进入内存保护模式的阈值，默认为 88%。
   - `memory-use-threshold-extreme`：内存保护的 extreme 模式，新建会话会被丢弃，默认为 95%。
   - `memory-use-threshold-green`：退出内存保护模式的低阈值，默认为 82%。

2. 首先配置告警邮件发送的配置（参考日志 → 邮件告警↗章节，也可使用默认配置）。

3. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击新建按钮，新建一个工作流。



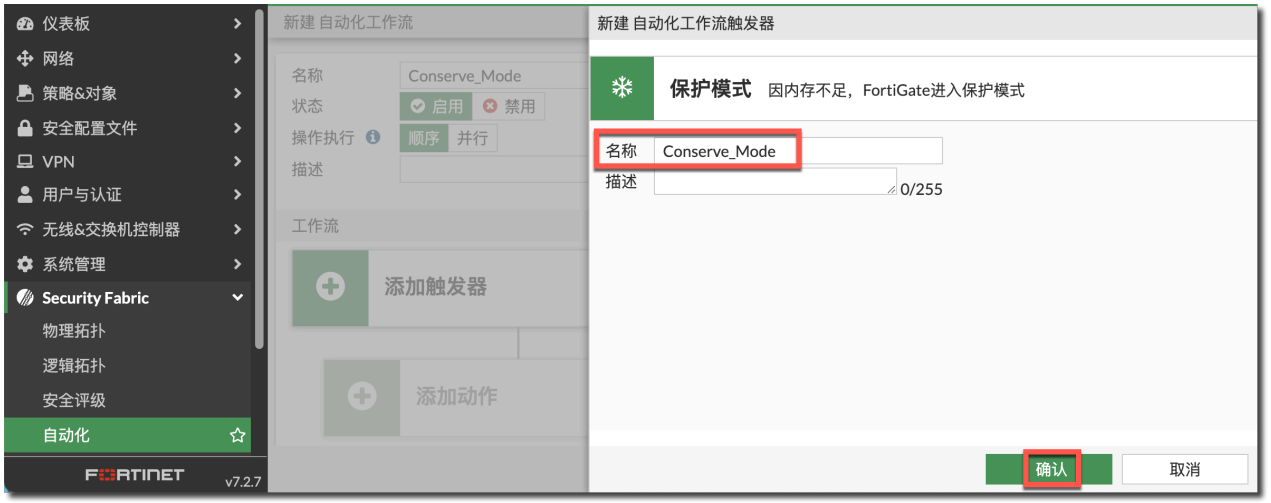
4. 配置工作流的名称，点击“添加触发器”按钮，在右侧的“选择条目”中点击“新建”按钮。



5. 在右侧弹出的窗口中，找到“系统”分类，点击“保护模式”选项。



6. 配置触发器（Trigger）的名称，点击确认下发配置。




```bash
config system automation-trigger
    edit "Conserve_Mode"
        set event-type low-memory
    next
end
```

7. 在选择条目中选择上步创建的触发器（Trigger），点击应用按钮。



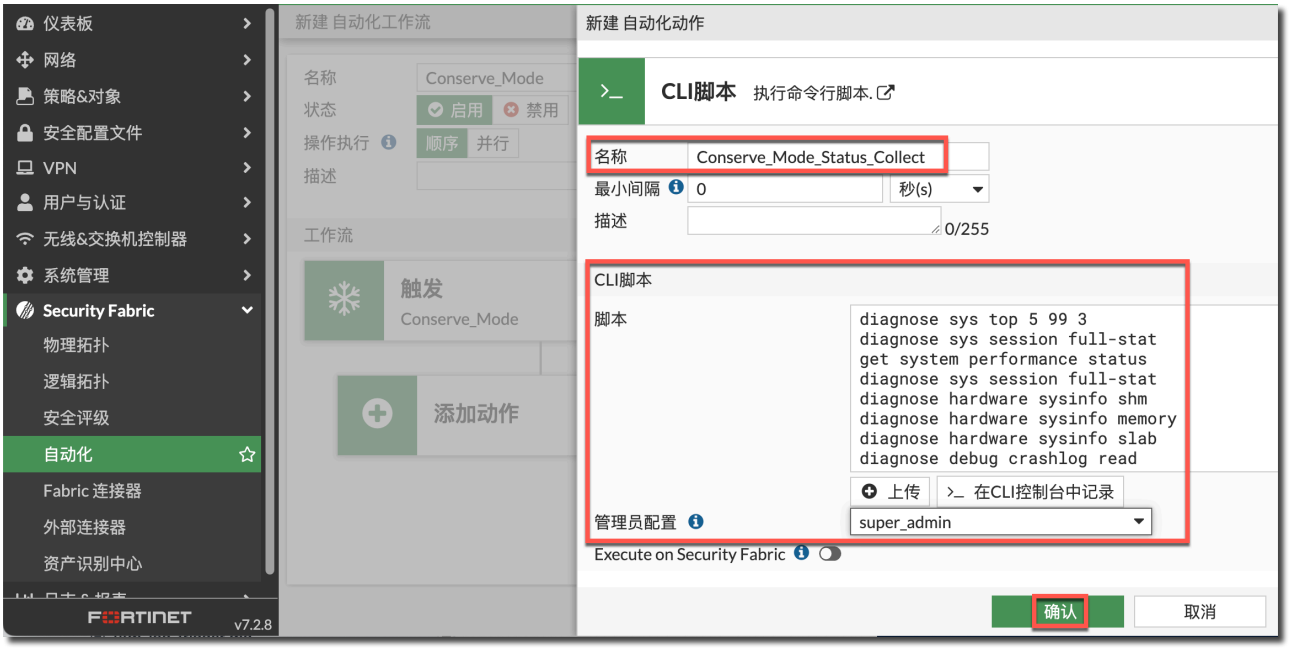
8. 点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



9. 在右侧弹出的窗口中，找到“通用”分类，点击“CLI 脚本”选项。

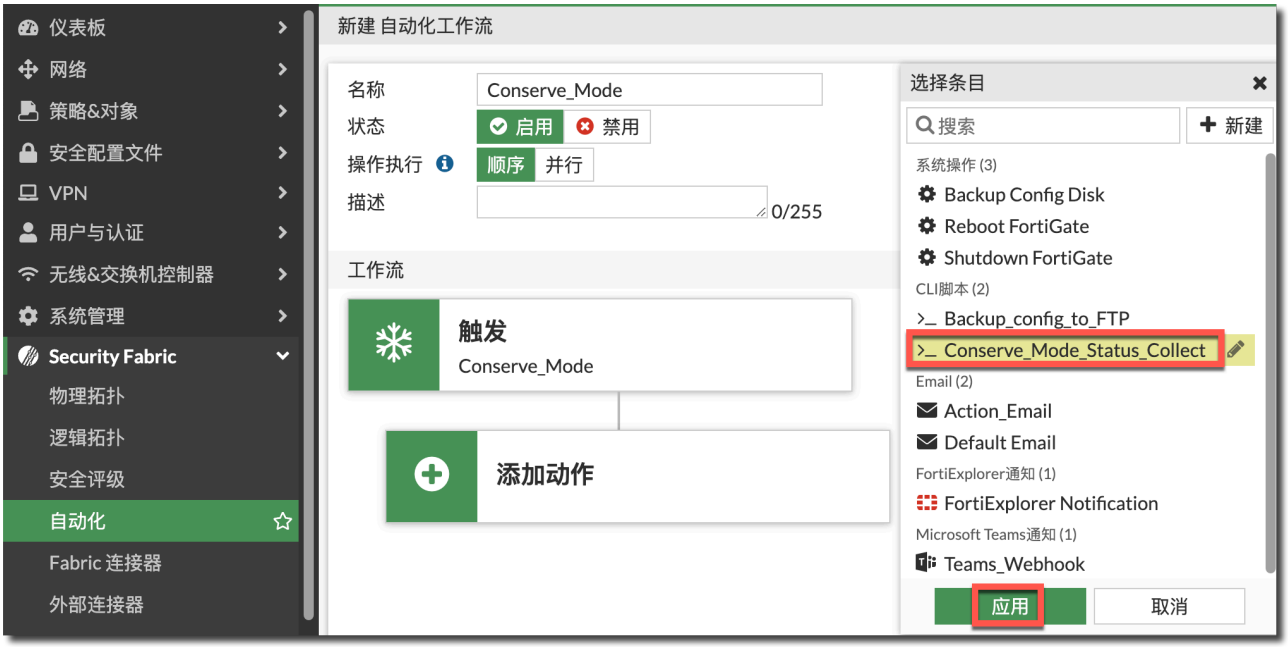


0. 配置动作（Action）的名称，配置 CLI 脚本内容为 FortiGate 进入内存保护模式后收集信息需要执行的 CLI 命令，点击确认下发配置。

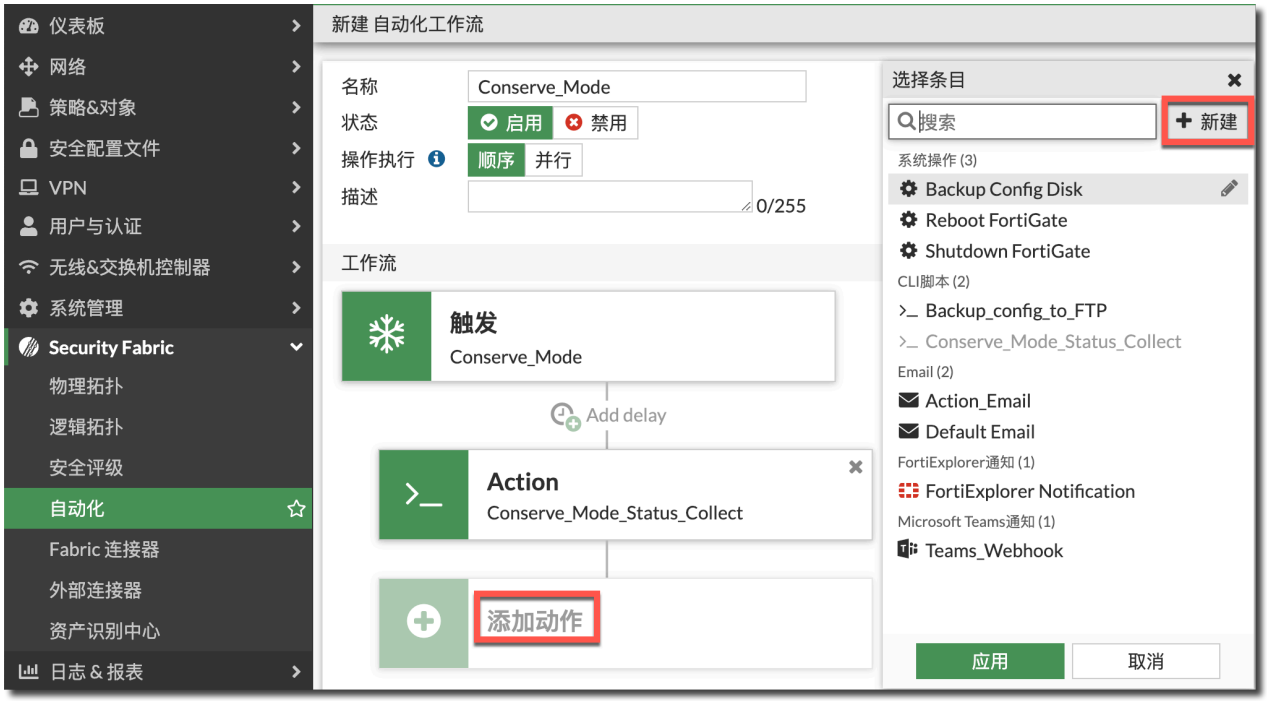



```bash
config system automation-action
    edit "Conserve_Mode_Status_Collect"
        set action-type cli-script
        set script "diagnose sys top 5 99 3
diagnose sys session full-stat
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo shm
diagnose hardware sysinfo memory
diagnose hardware sysinfo slab
diagnose debug crashlog read"
        set accprofile "super_admin"
    next
end
```

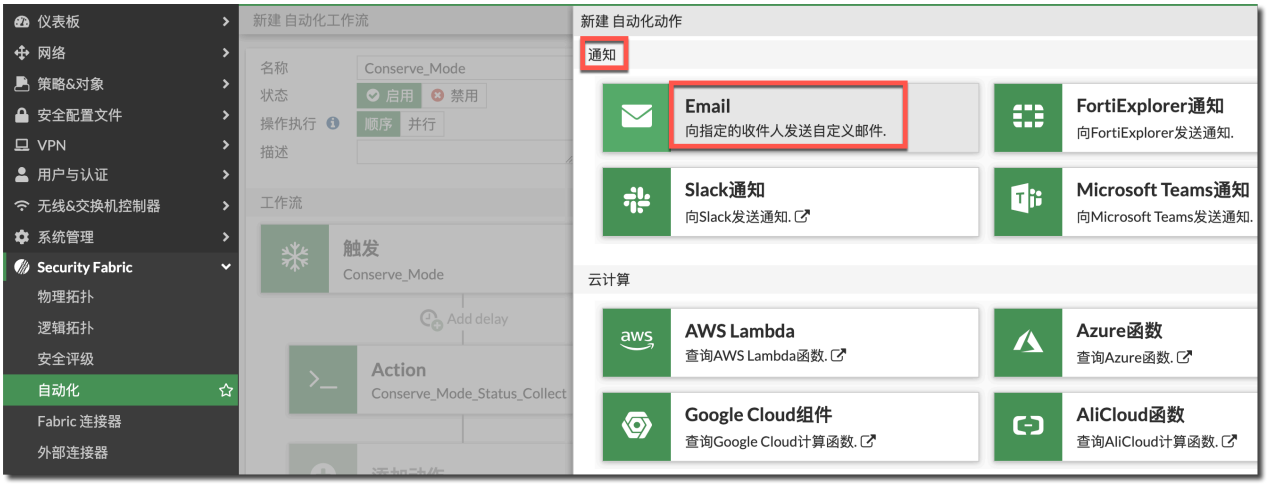
1. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



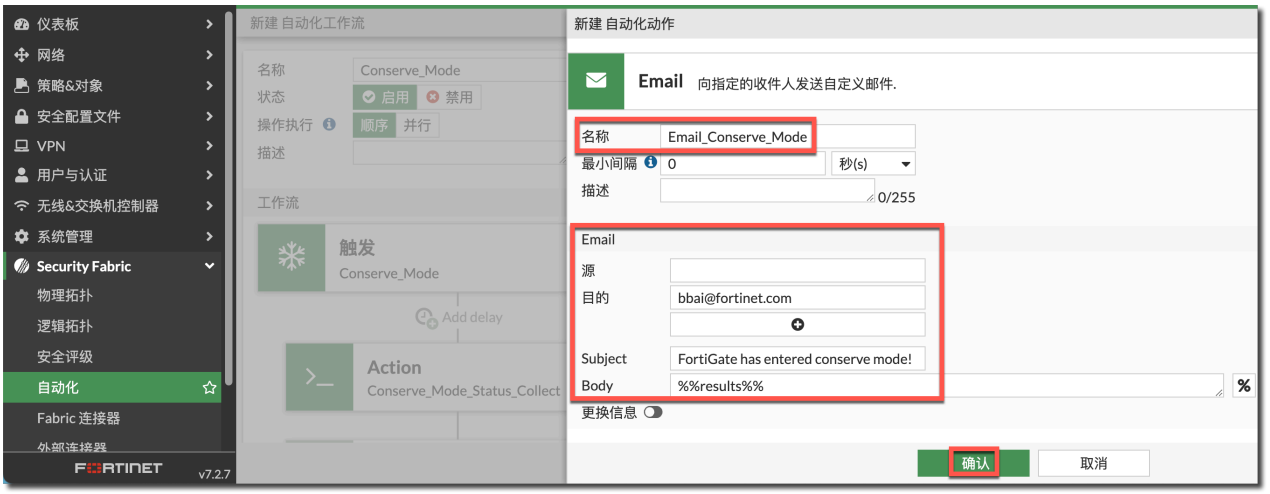
2. 再次点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



3. 在右侧弹出的窗口中，找到“通知”分类，点击“Email”选项。



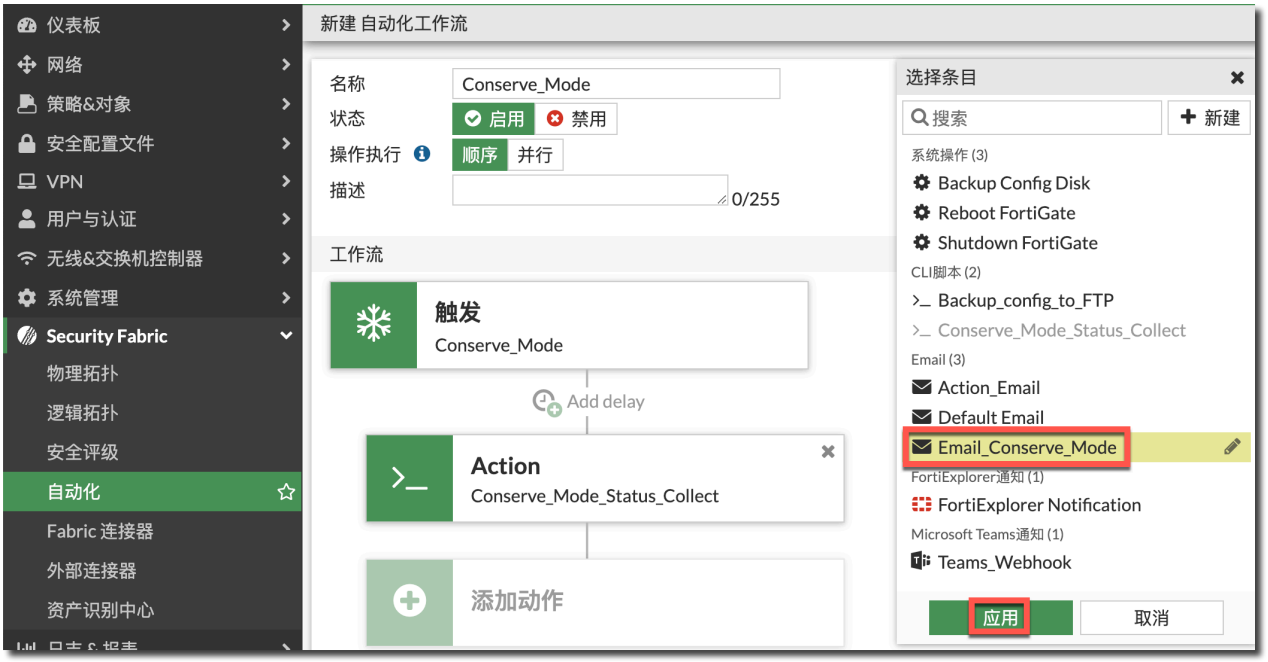
4. 配置动作（Action）的名称，以及邮件通知的收件人、邮件标题，邮件内容填写 `%%results%%`，这是一个变量，表示使用前序动作 CLI 命令执行的结果作为邮件正文，点击确认下发配置。



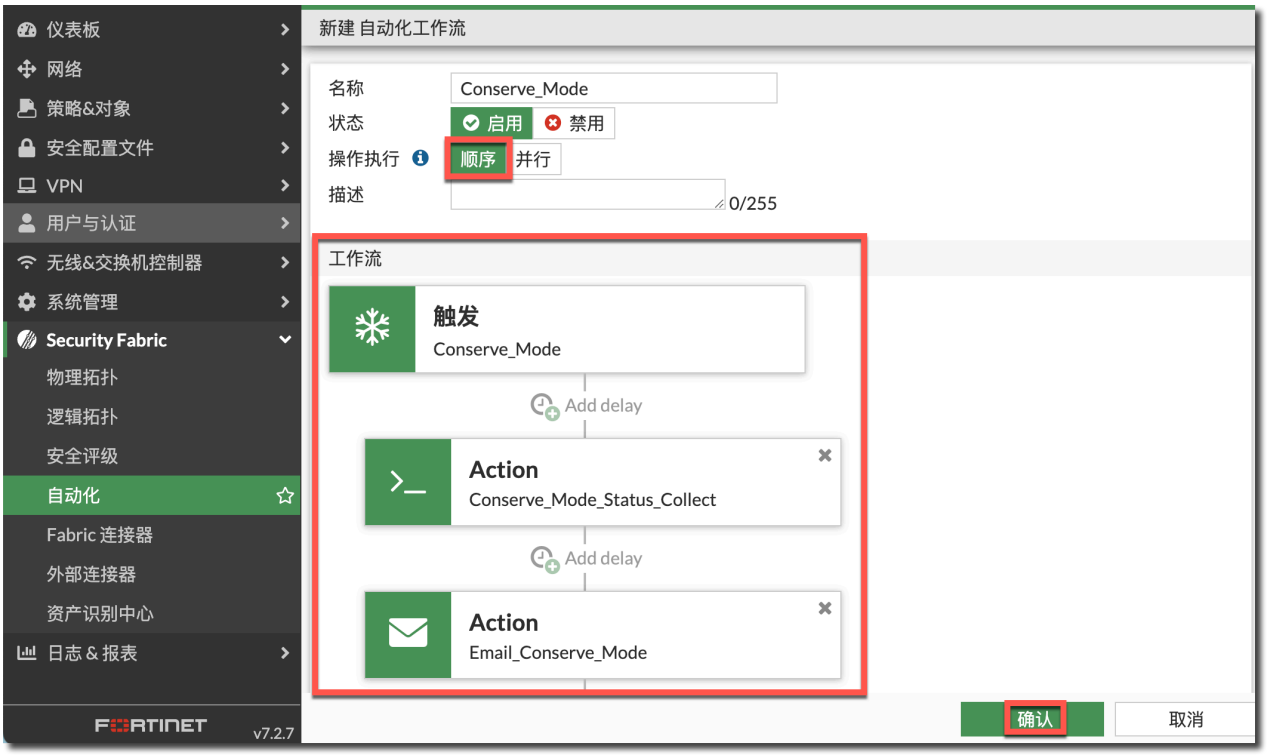
- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 `DoNotReply@notification.fortinet.net`。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

```bash
config system automation-action
    edit "Email_Conserve_Mode"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "FortiGate has entered conserve mode!"
        set message "%%results%%"
    next
end
```

5. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



6. 检查无误后，点击确认下发配置。

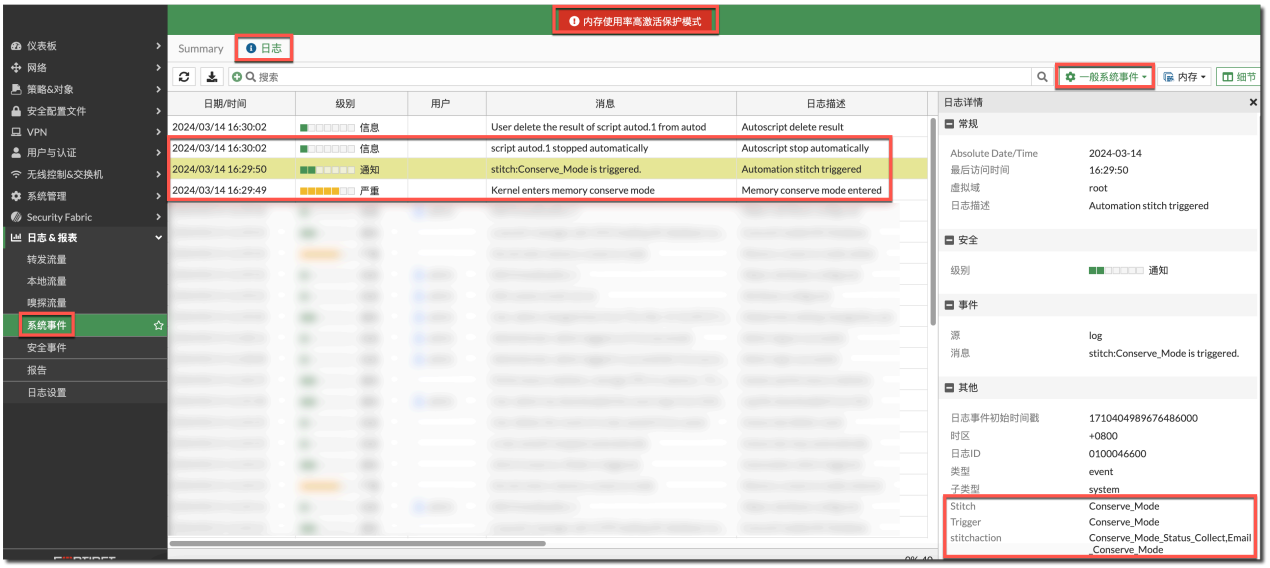



```bash
config system automation-stitch
    edit "Conserve_Mode"
        set trigger "Conserve_Mode"
        config actions
            edit 1
                set action "Conserve_Mode_Status_Collect"
                set required enable
            next
            edit 2
                set action "Email_Conserve_Mode"
                set required enable
            next
        end
    next
end
```

配置验证

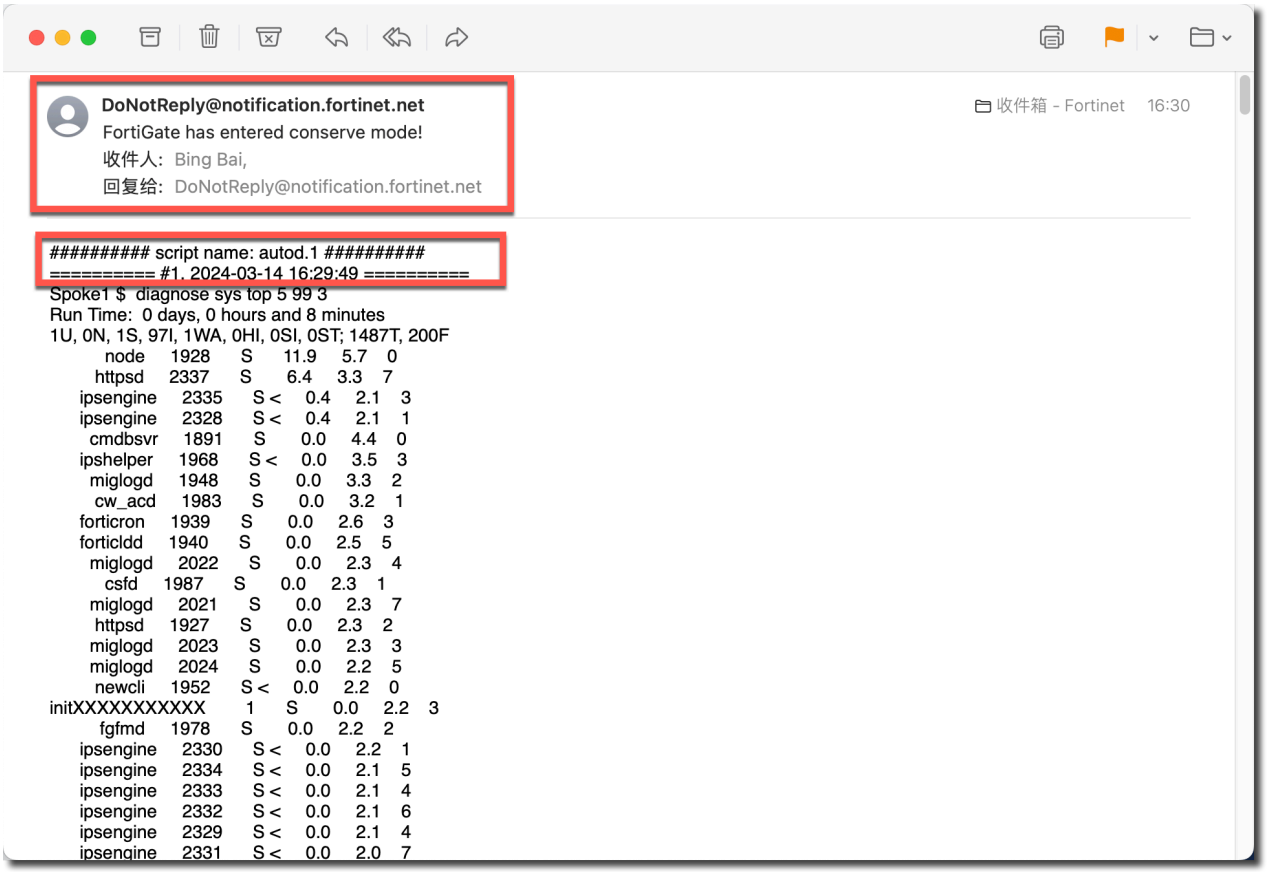
1. 触发 FortiGate 进入内存保护模式（Conserve Mode）。

2. 触发自动化程序，相关自动化日志如下。




```bash
date=2024-03-14 time=16:29:50 eventtime=1710404989676485934 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="Conserve_Mode" trigger="Conserve_Mode"
stitchaction="Conserve_Mode_Status_Collect,Email_Conserve_Mode" from="log" msg="stitch:Conserve_Mode is triggered."

date=2024-03-14 time=16:29:49 eventtime=1710404989471003340 tz="+0800" logid="0100022011" type="event" subtype="system"
level="critical" vd="root" logdesc="Memory conserve mode entered" service="kernel" conserve="on" total=1487 MB used=1118
MB red="1115 MB" green="1040 MB" msg="Kernel enters memory conserve mode"
```

3. 收件人邮箱收到了告警邮件，邮件内容为通过 CLI 动作收集到的设备状态信息，发件人为默认的 `DoNotReply@notification.fortinet.net`。



4. 对应的自动化触发 Debug 信息。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable

__action_cli_script_open()-186: cli script action:Conserve_Mode_Status_Collect is called. svc ctx:0×fc044a0
accprof:super_admin script:
diagnose sys top 5 99 3
diagnose sys session full-stat
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo shm
diagnose hardware sysinfo memory
diagnose debug crashlog read

__cli_script_close()-121: cli script:
autod.2
 output:
########## script name: autod.2 ##########

========== #1, 2024-03-14 16:35:13 ==========
Spoke1 $  diagnose sys top 5 99 3
Run Time:  0 days, 0 hours and 13 minutes
0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 1487T, 217F
          autod     1988      S       1.0     1.7    5
      ipshelper     1968      S <     0.5     3.4    7
      forticron     1939      S       0.5     2.6    7
      ipsengine     2441      S <     0.5     2.2    0
      ipsengine     2439      S <     0.5     2.1    4

......

__action_cli_script_close()-213: cli script action is done. script:
diagnose sys top 5 99 3
diagnose sys session full-stat
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo shm
```

```
diagnose hardware sysinfo memory
diagnose debug crashlog read
output:
########## script name: autod.2 ##########

========== #1, 2024-03-14 16:35:13 ==========
Spoke1 $  diagnose sys top 5 99 3
Run Time:  0 days, 0 hours and 13 minutes
0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 1487T, 217F
           autod     1988      S       1.0     1.7    5
       ipshelper     1968      S <     0.5     3.4    7
       forticron     1939      S       0.5     2.6    7
       ipsengine     2441      S <     0.5     2.2    0
       ipsengine     2439      S <     0.5     2.1    4

......

pid:1988-__handle_msg()-414: Subscriber:4 received package. pubid:2 pkgid:73 pkg_index:0
pid:1988-__handle_pkg_logs()-356: Subscriber:4 processing package size:516 logs:1 pickup:1
auto_event_pub_handle_log()-331: publish 32550 to subscr-node
pid:1988-__subscr_close_cur_pkg()-140: close package size:516 logs:1
__action_email_hdl()-181: email action (Email_Conserve_Mode) is called.
from:
to:bbai@fortinet.com;
subject:FortiGate has entered conserve mode!
```

5. 发送邮件的 Debug 信息如下。

```bash
FortiGate # diagnose debug application alertmail -1
FortiGate # diagnose debug enable

Arrived msg(type 9, 46668 bytes):bbai@fortinet.com;

FortiGate has entered conserve mode!
########## script name: autod.3 ##########
========== #1, 2024-03-14 16:35:13 ==========
Spoke1 $  diagnose sys top 5 99 3
Run Time:  0 days, 0 hours and 13 minutes
0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 1487T, 217F
           autod     1988      S       1.0     1.7    5
       ipshelper     1968      S <     0.5     3.4    7
       forticron     1939      S       0.5     2.6    7
       ipsengine     2441      S <     0.5     2.2    0
       ipsengine     2439      S <     0.5     2.1    4

......

    mail_info:
        from:notification.fortinet.net  user:DoNotReply@notification.fortinet.net
mail_info:
        reverse path:DoNotReply@notification.fortinet.net
        user name:DoNotReply
to[0]:bbai@fortinet.com
⟸_init_mail_info
create session
resolve notification.fortinet.net to 1 IP
⟹ send mail
connecting to 208.91.114.151 port 465
send mail 0×fb2cbb0 session 0×fb315e0
session_io_event: creating ssl structure for session 0×fb315e0
create_ssl_ctx
create_ssl: 0×7f99cedbd000
sessionn 0×fb315e0, SSL connected
session: 0×fb315e0, rsp_state: greeting, code: 220
session: 0×fb315e0, rsp_state: ehlo, code: 250
session: 0×fb315e0, rsp_state: mail, code: 250
session: 0×fb315e0, rsp_state: rcpt, code: 250
session: 0×fb315e0, rsp_state: data, code: 354
⟹ send: ########## script name: autod.3 ##########
========== #1, 2024-03-14 16:35:13 ==========
Spoke1 $  diagnose sys top 5 99 3
Run Time:  0 days, 0 hours and 13 minutes
```

```
0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 1487T, 217F
           autod     1988      S       1.0     1.7    5
       ipshelper     1968      S <     0.5     3.4    7
       forticron     1939      S       0.5     2.6    7
       ipsengine     2441      S <     0.5     2.2    0
       ipsengine     2439      S <     0.5     2.1    4

......

1session: 0×fb315e0, rsp_state: data2, code: 250
session: 0×fb315e0, rsp_state: quit, code: 221
session finined
_session_on_destroy
⟸ send mail success, m = 0×fb2cbb0 s = 0×fb315e0
```

平均 CPU 使用率高

网络需求

当 FortiGate 的 CPU 多个核心的平均利用率达到设定的阈值后，触发 CLI 动作收集设备当前的运行信息，并将信息发送至指定邮箱。

配置步骤

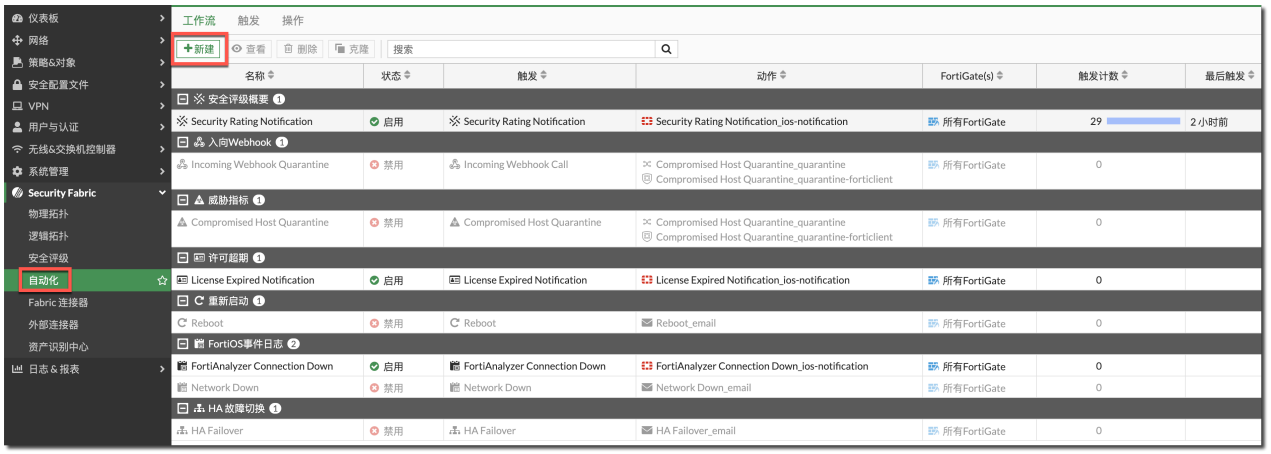
1. CPU 阈值配置。

```bash
config system global
    set cpu-use-threshold 85
end
```

   - `cpu-use-threshold`：CPU 所有核心的平均利用率阈值，默认为 90%。

2. 首先配置告警邮件发送的配置（参考日志 → 邮件告警↗章节，也可使用默认配置）。

3. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击新建按钮，新建一个工作流。



4. 配置工作流的名称，点击“添加触发器”按钮，在右侧的“选择条目”中点击“新建”按钮。



5. 在右侧弹出的窗口中，找到“系统”分类，点击“高 CPU 使用率”选项。



6. 配置触发器（Trigger）的名称，点击确认下发配置。




```bash
config system automation-trigger
    edit "CPU_High"
        set event-type high-cpu
    next
end
```

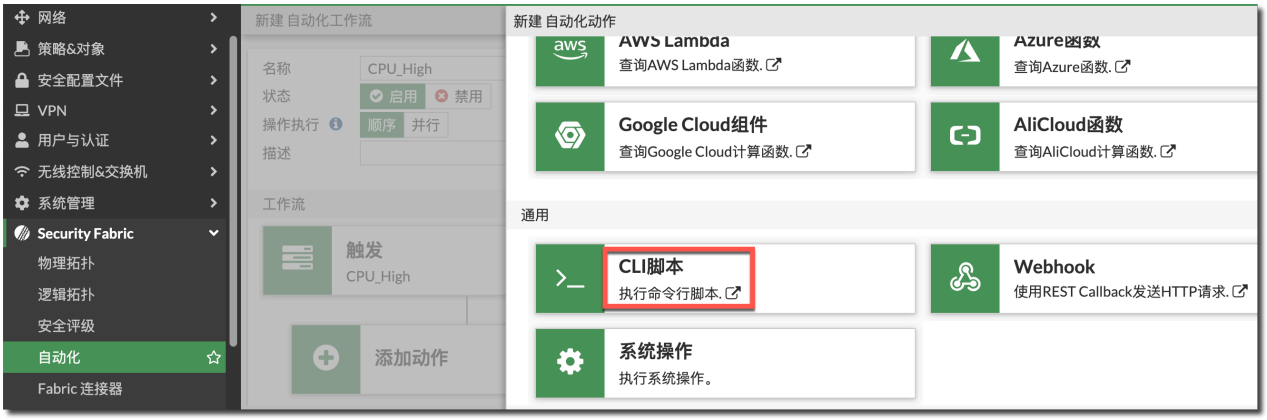
7. 在选择条目中选择上步创建的触发器（Trigger），点击应用按钮。



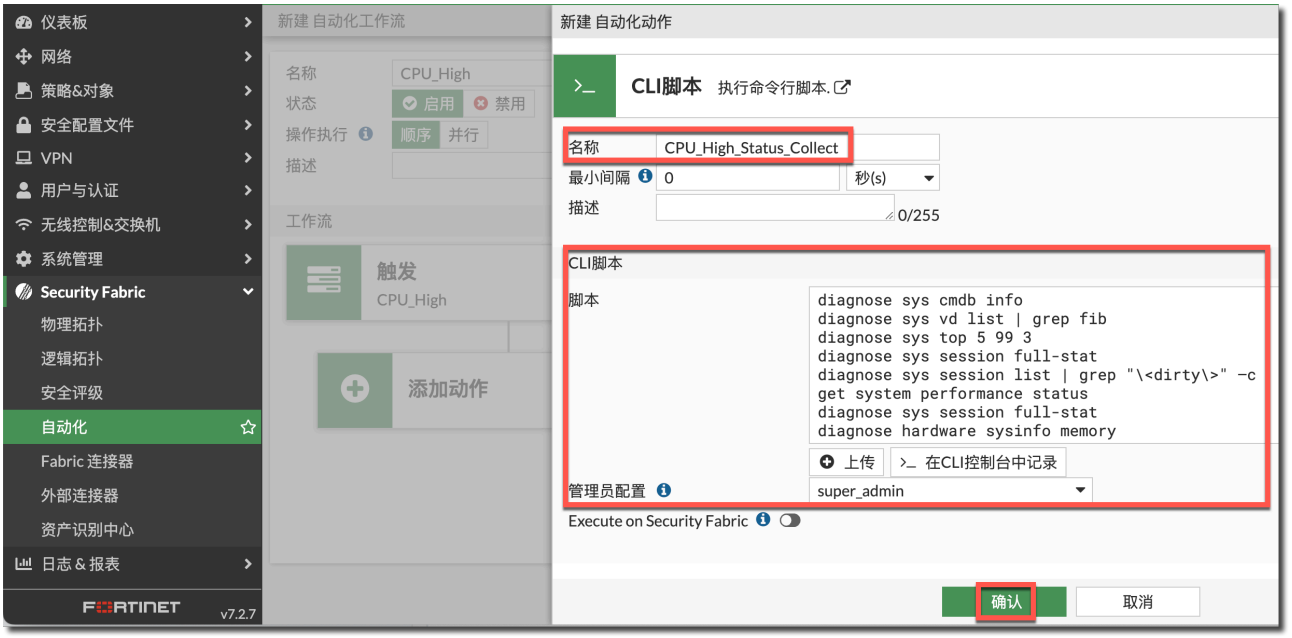
8. 点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



9. 在右侧弹出的窗口中，找到“通用”分类，点击“CLI 脚本”选项。



.0. 配置动作（Action）的名称，配置 CLI 脚本内容为 FortiGated 的 CPU 利用率达到阈值后收集信息需要执行的 CLI 命令，点击确认下发配置。

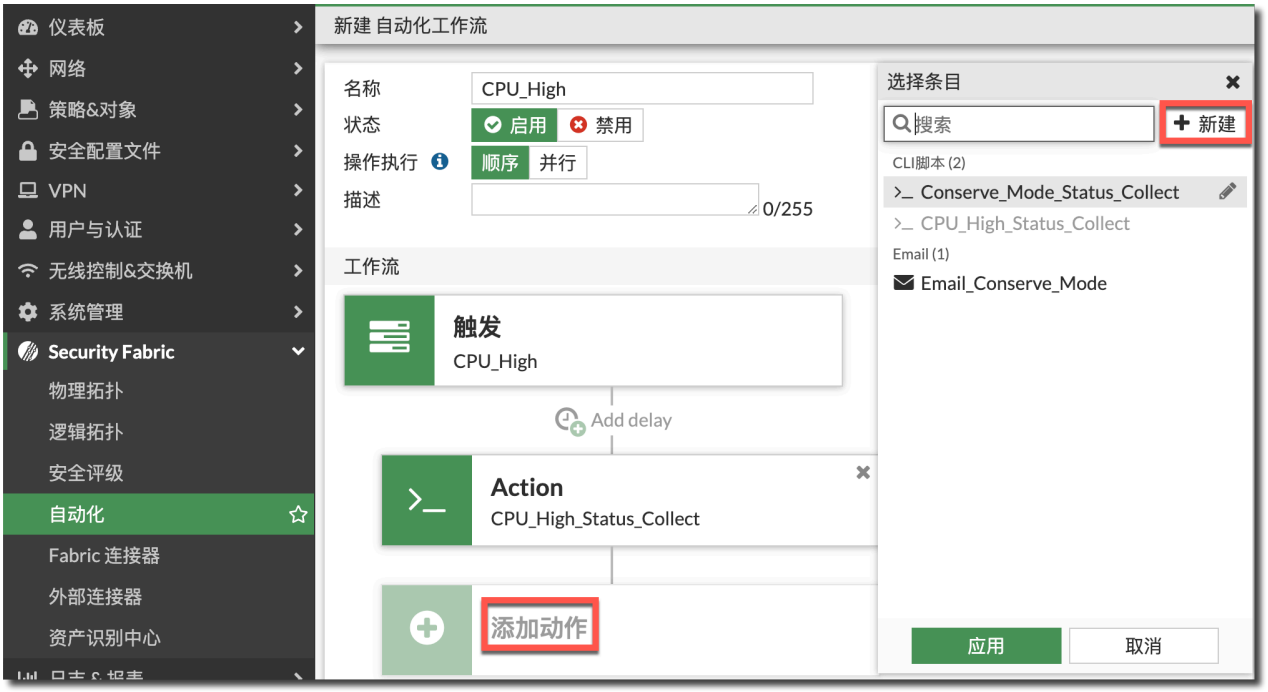



```bash
config system automation-action
    edit "CPU_High_Status_Collect"
        set action-type cli-script
        set script "diagnose sys cmdb info
diagnose sys vd list | grep fib
diagnose sys top 5 99 3
diagnose sys session full-stat
diagnose sys session list | grep \"\\<dirty\\>\" -c
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo memory"
        set accprofile "super_admin"
    next
end
```

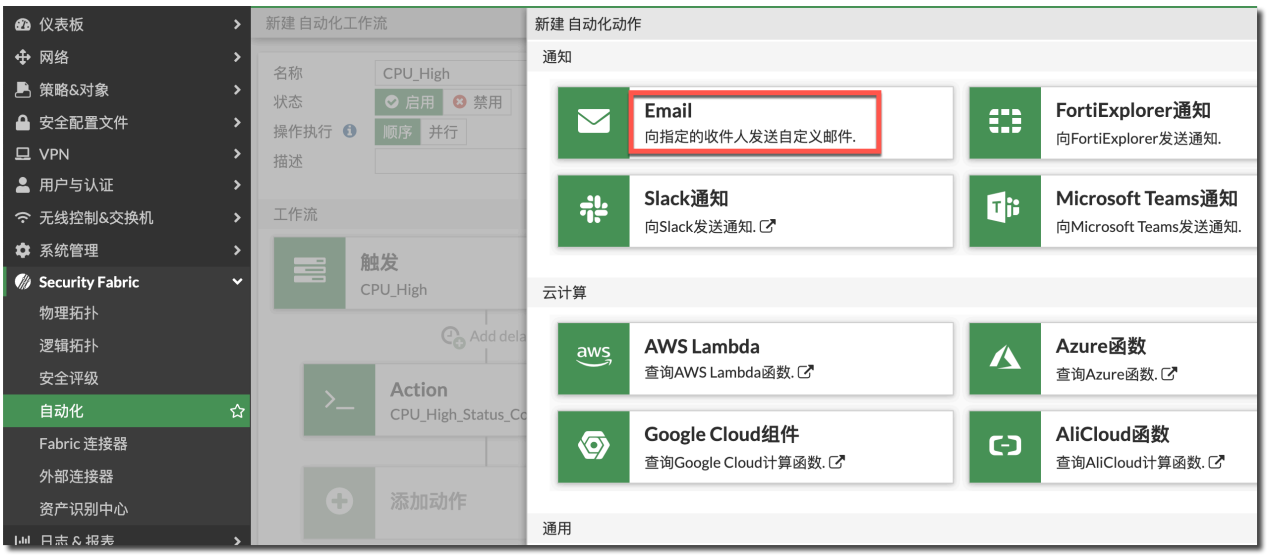
1. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



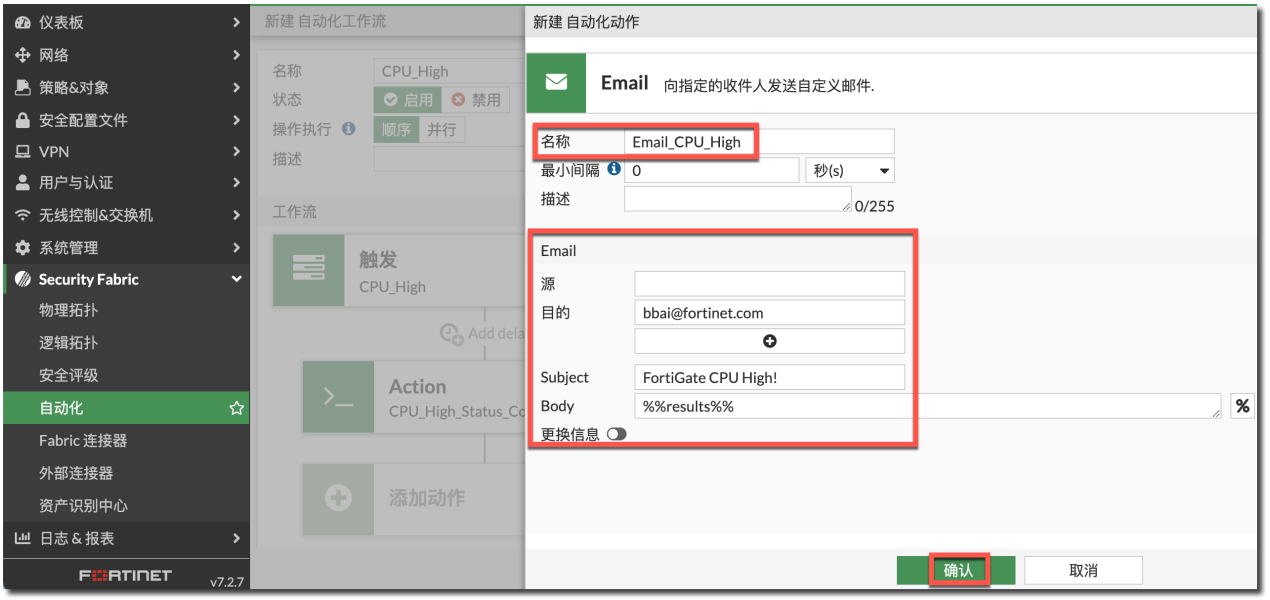
2. 再次点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



3. 在右侧弹出的窗口中，找到“通知”分类，点击“Email”选项。



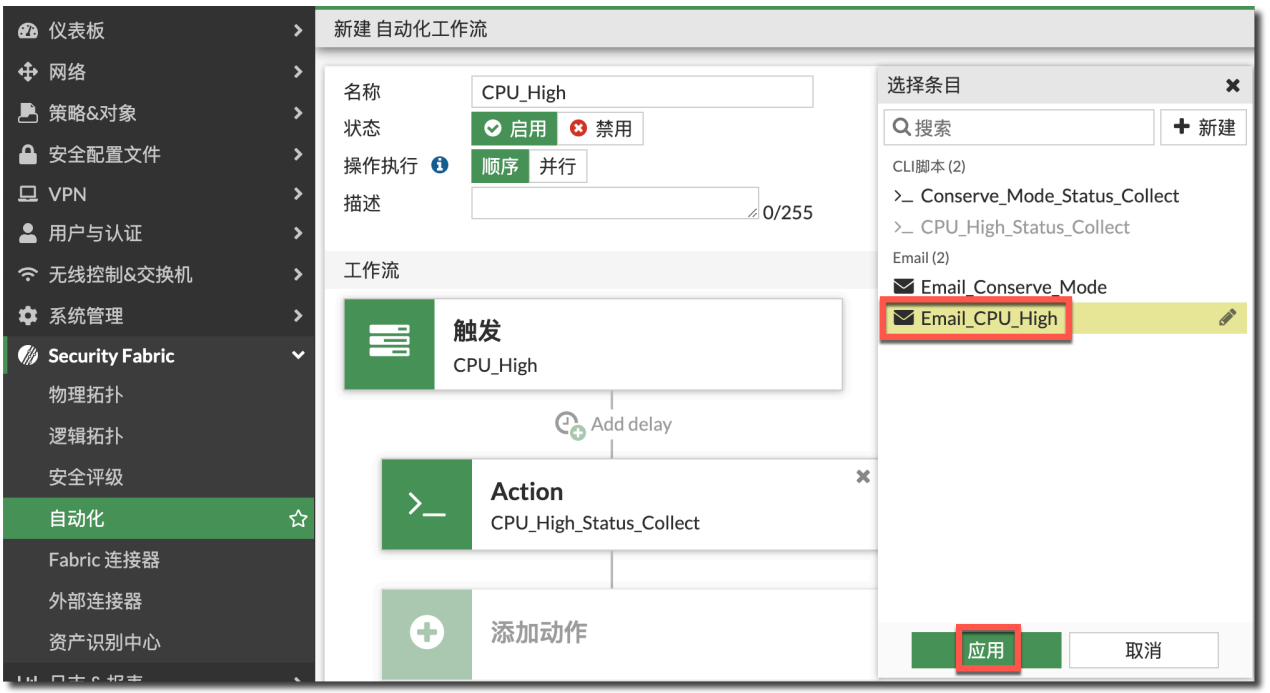
4. 配置动作（Action）的名称，以及邮件通知的收件人、邮件标题，邮件内容填写 `%%results%%`，这是一个变量，表示使用前序动作 CLI 命令执行的结果作为邮件正文，点击确认下发配置。



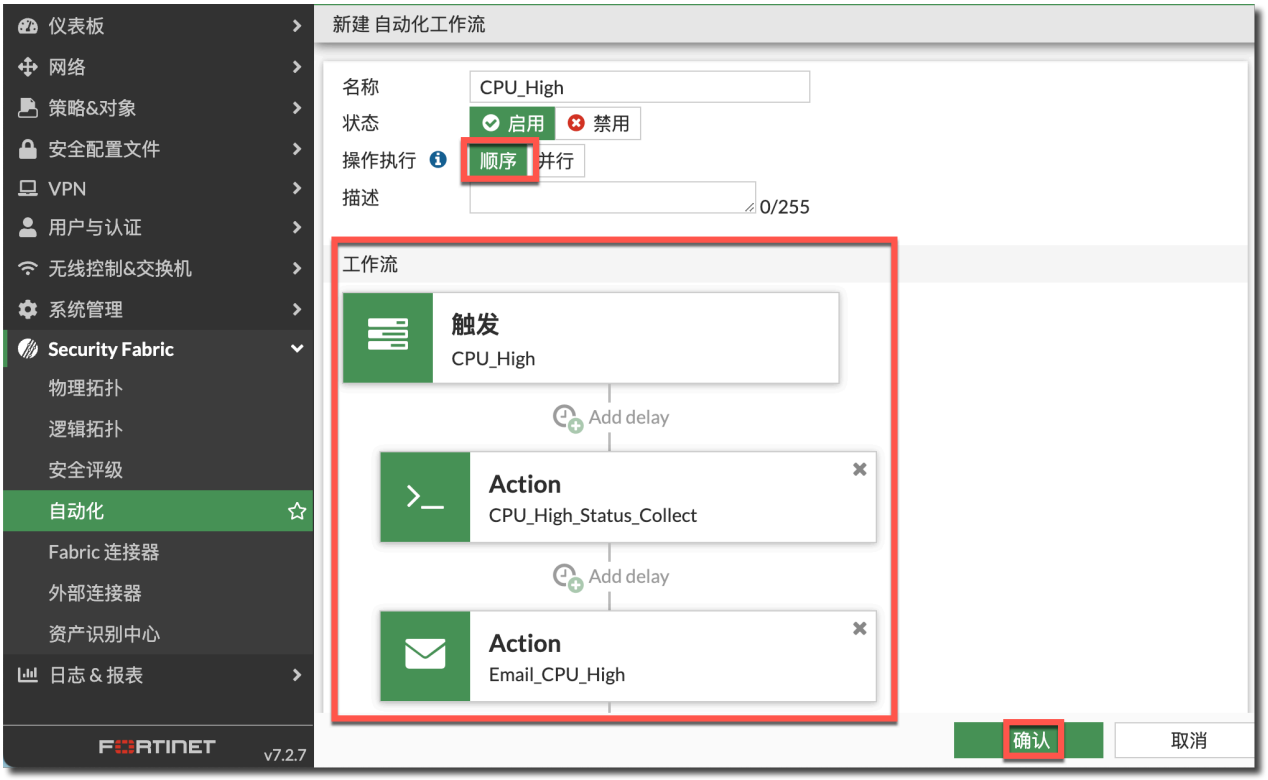
- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 DoNotReply@notification.fortinet.net。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

```bash
config system automation-action
    edit "Email_CPU_High"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "FortiGate CPU High!"
        set message "%%results%%"
    next
end
```

5. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



6. 检查无误后，点击确认下发配置。

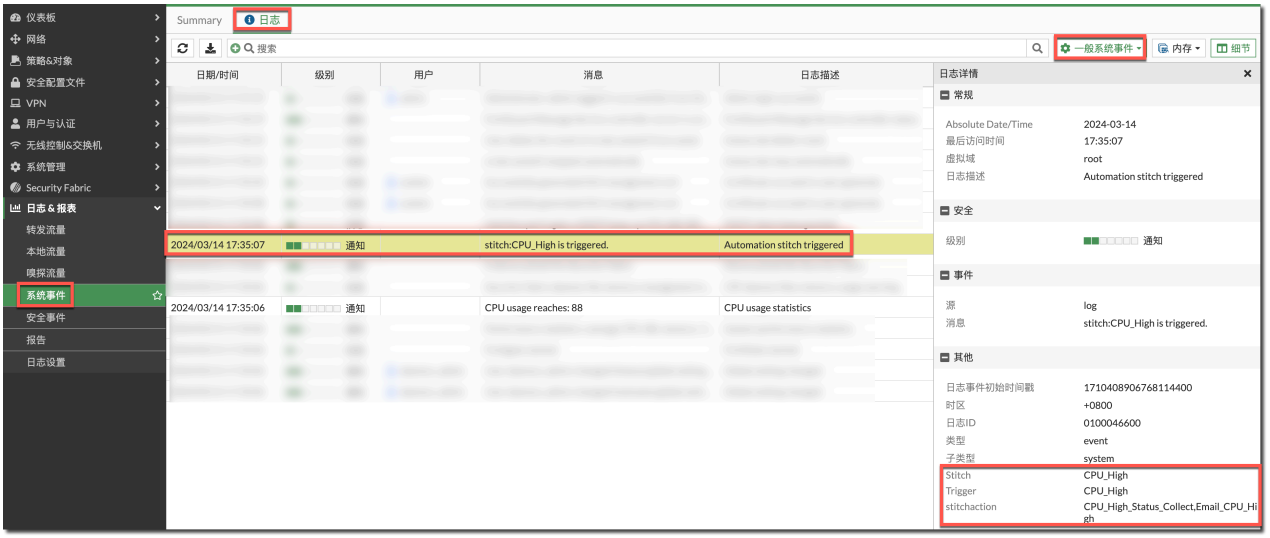



```bash
config system automation-stitch
    edit "CPU_High"
        set trigger "CPU_High"
        config actions
            edit 1
                set action "CPU_High_Status_Collect"
                set required enable
            next
            edit 2
                set action "Email_CPU_High"
                set required enable
            next
        end
    next
end
```

配置验证

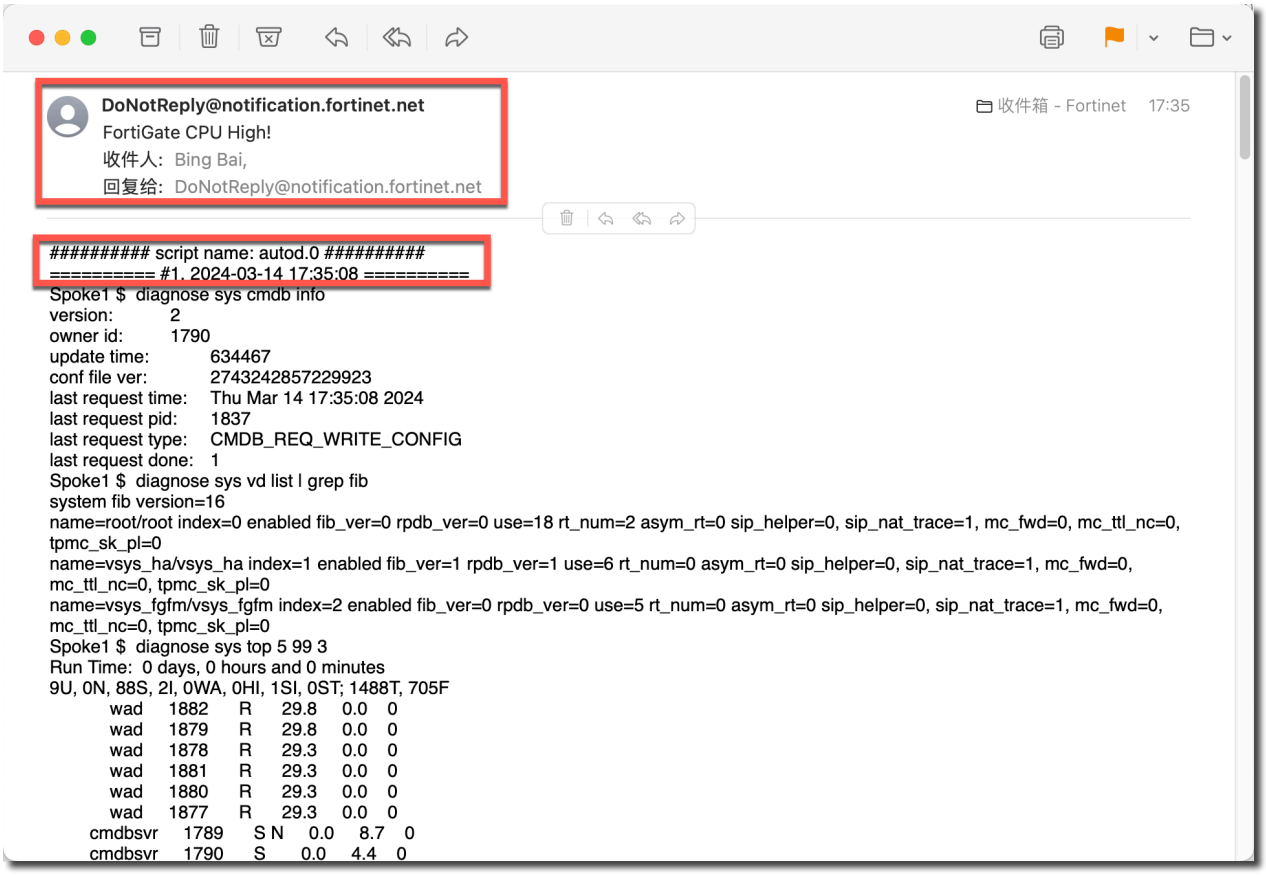
1. 触发 FortiGate 的 CPU 利用率达到设定的阈值（此例中的 FortiGate CPU 高阈值设置为 85%）。

2. 触发自动化程序，相关自动化日志如下。




```bash
date=2024-03-14 time=17:35:07 eventtime=1710408906768114390 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="CPU_High" trigger="CPU_High"
stitchaction="CPU_High_Status_Collect,Email_CPU_High" from="log" msg="stitch:CPU_High is triggered."

date=2024-03-14 time=17:35:06 eventtime=1710408897701912754 tz="+0800" logid="0100040705" type="event" subtype="system"
level="notice" vd="root" logdesc="CPU usage statistics" action="cpu-usage" cpu=88 msg="CPU usage reaches: 88"
```

3. 收件人邮箱收到了告警邮件，邮件内容为通过 CLI 动作收集到的设备状态信息，发件人为默认的 DoNotReply@notification.fortinet.net。



4. 查看自动化触发 Debug 信息（略，参考内存保护模式自动化的 Debug 信息）。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable
```

5. 查看发送邮件的 Debug 信息（略，参考内存保护模式自动化动作的邮件告警 Debug 信息）。

```
FortiGate # diagnose debug application alertmail -1
FortiGate # diagnose debug enable
```
bash

单核 CPU 使用率高

重要

该功能在 7.2.4 及以上版本支持。

网络需求

很多情况下，FortiGate 的某个 CPU 持续高也会影响到正常运行，这种情况下是无法触发上个章节中平均 CPU 高触发器的，此章节可以实现单个 CPU 核利用率超过配置的阈值时，触发 CLI 动作收集设备当前的运行信息，并将信息发送至指定邮箱。

配置步骤

1. 首先配置告警邮件发送的配置（参考日志 → 邮件告警↗章节，也可使用默认配置）。

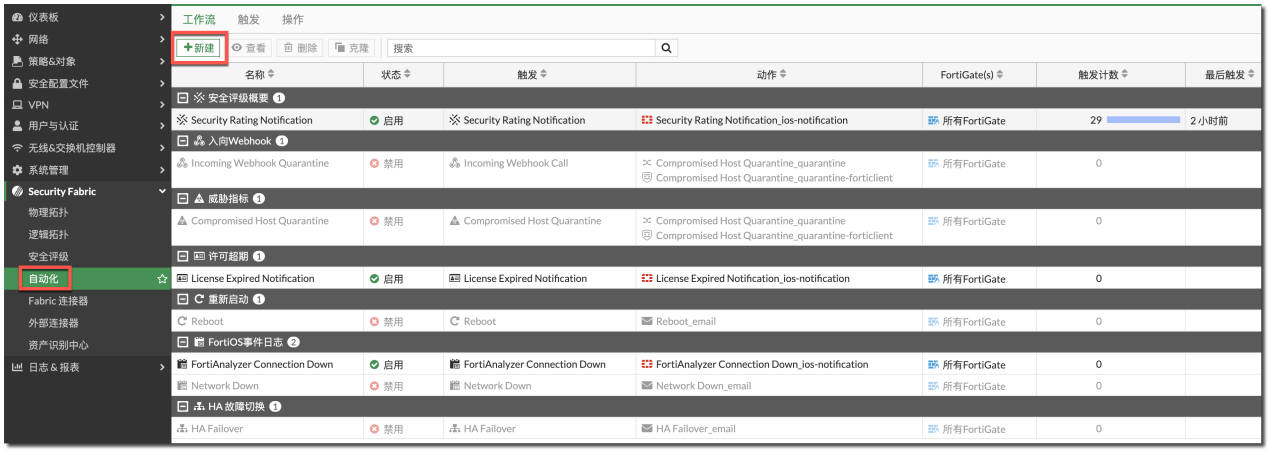
2. 在 FortiGate 的 CLI 中开启日志记录单核 CPU 高的功能，开启后，`cpu-use-threshold` 也会对单个 CPU 核心生效。

```
config system global
    set log-single-cpu-high enable
    set cpu-use-threshold 85
end
```
bash

3. 产生的日志示例如下，Log ID 为 40707。

```
date=2024-03-15 time=14:08:23 eventtime=1710482896991349328 tz="+0800" logid="0100040707" type="event" subtype="system"
level="notice" vd="root" logdesc="CPU single core usage statistics" action="cpu-single-core-usage" core=1 cpu=97
msg="CPU core 1 usage reaches: 97"
```
bash

4. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击新建按钮，新建一个工作流。



5. 配置工作流的名称，点击“添加触发器”按钮，在右侧的“选择条目”中点击“新建”按钮。



6. 在右侧弹出的窗口中，找到“其他条件”分类，点击“FortiOS 事件日志”选项。



7. 配置触发器（Trigger）的名称，FortiOS 事件日志中选择“CPU single core usage statistics”，点击确认下发配置。




```bash
config system automation-trigger
    edit "Single_CPU_High"
        set event-type event-log
        set logid 40707
    next
end
```

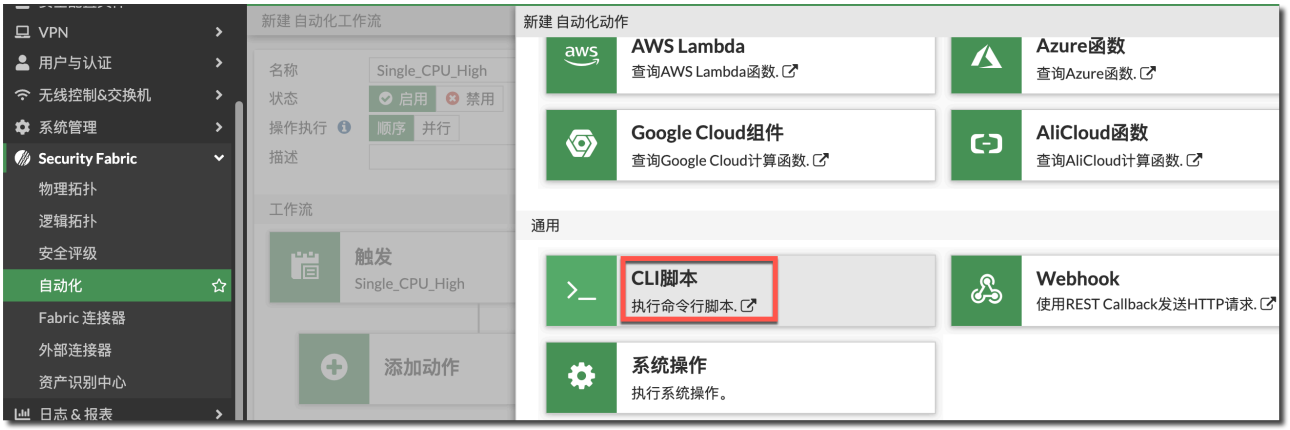
8. 在选择条目中选择上步创建的触发器（Trigger），点击应用按钮。



9. 点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



0. 在右侧弹出的窗口中，找到“通用”分类，点击“CLI 脚本”选项。



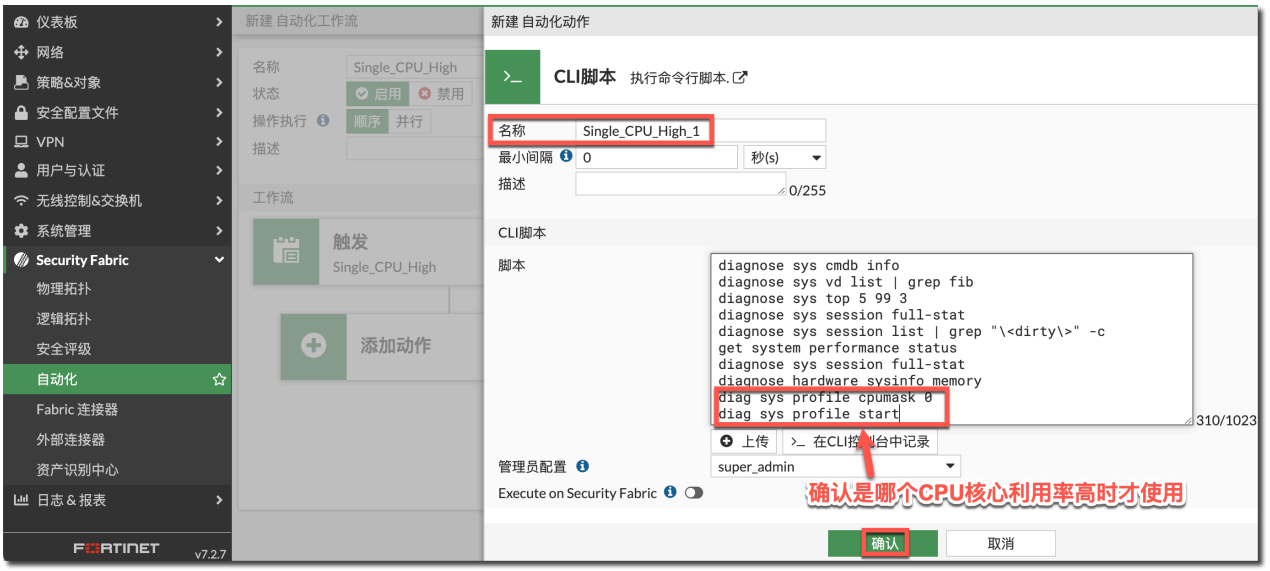
1. 配置动作（Action）的名称，配置 CLI 脚本内容为 FortiGated 的 CPU 利用率达到阈值后收集信息需要执行的 CLI 命令，点击确认下发配置。

重要

如果已经确定是哪个 CPU 核心持续高，但出现该单核高的时间不确定，可以在上个章节收集的 CLI 命令后加入如下命令来查看该 CPU 代码级别的 Debug 信息（由于前 2 条命令和后 2 条命令中间需要 1 分钟的收集间隔，需要 2 个 Action 来完成所有的信息收集）：

```
diag sys profile cpumask 0      //抓取特定的CPU核心，这里以CPU核心0为例//
diag sys profile start          //开启profile程序，等待1分钟，再执行后续命令//
diag sys profile stop
diag sys profile show order
```

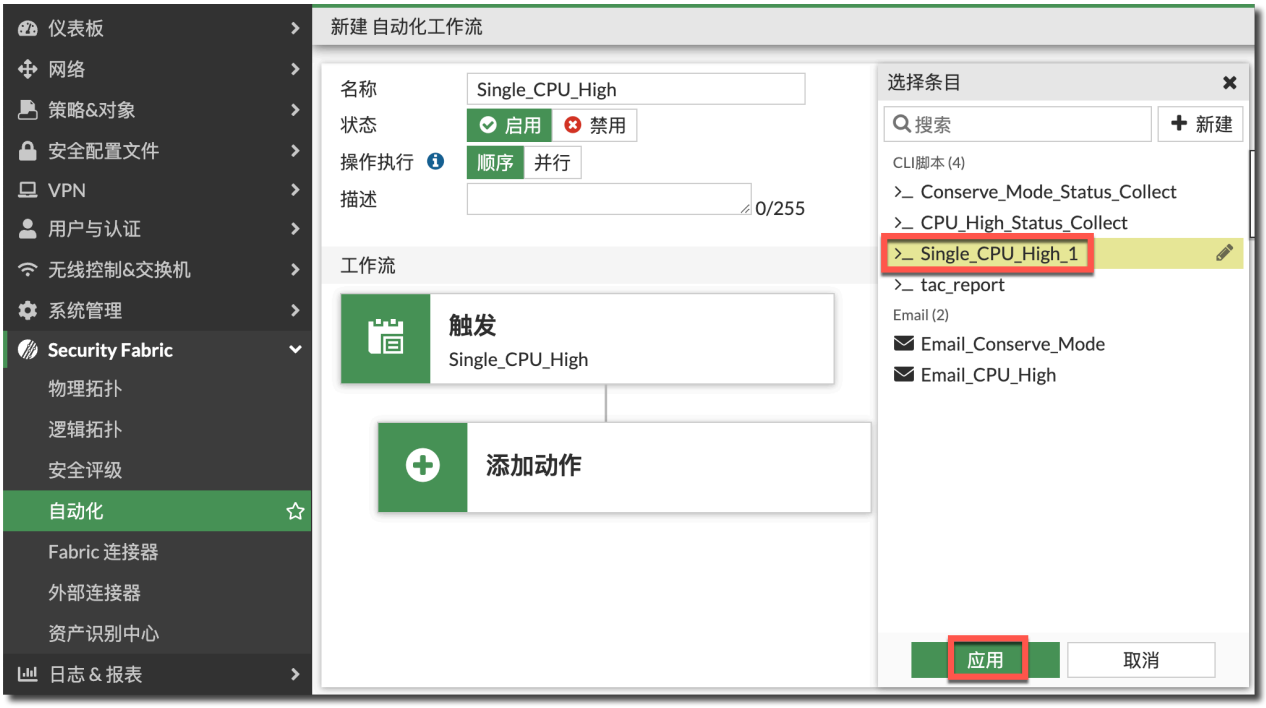
bash



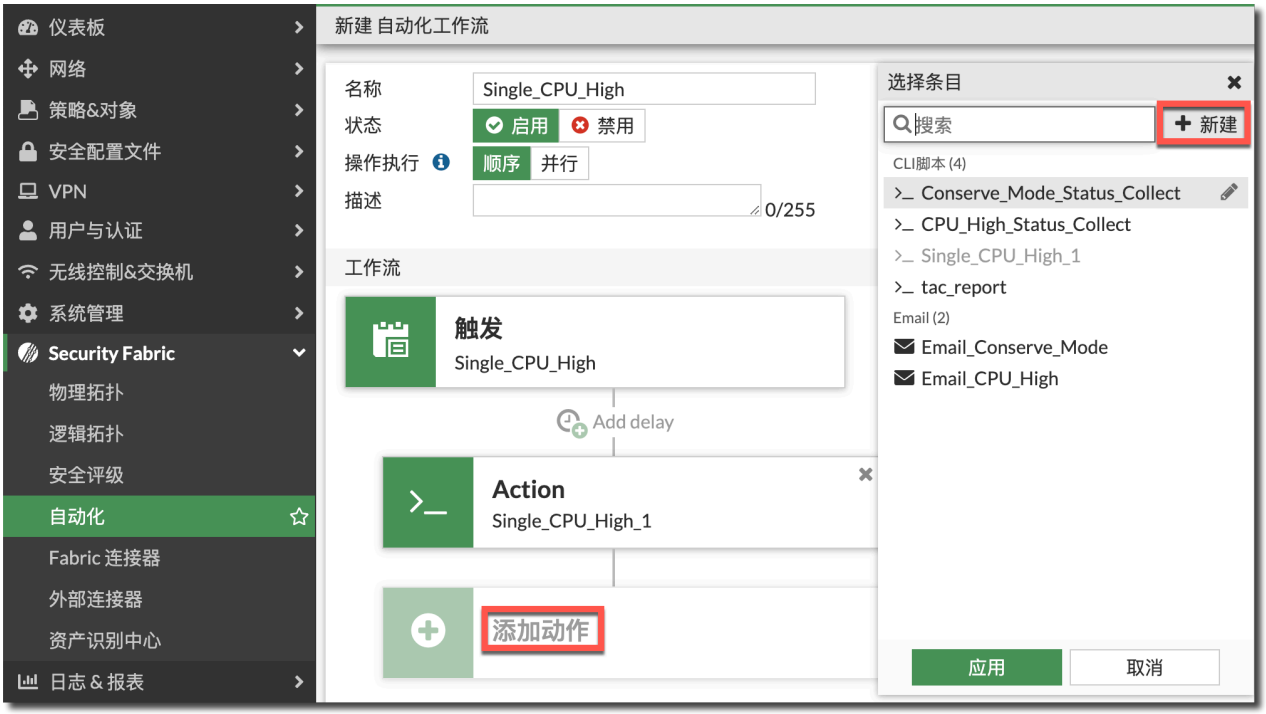

```
config system automation-action
    edit "CPU_High_Status_Collect"
        set action-type cli-script
        set script "diagnose sys cmdb info
diagnose sys vd list | grep fib
diagnose sys top 5 99 3
diagnose sys session full-stat
diagnose sys session list | grep "\<dirty\>" -c
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo memory"
————以下CLI在确认哪个CPU核心高的情况下可用（CLI脚本不包含此行）————
diag sys profile cpumask 0      //抓取特定的CPU核心，这里以CPU核心0为例//
diag sys profile start          //开启profile程序，等待1分钟，再执行后续命令//
————以上CLI在确认哪个CPU核心高的情况下可用（CLI脚本不包含此行）————
        set accprofile "super_admin"
    next
end
```

bash

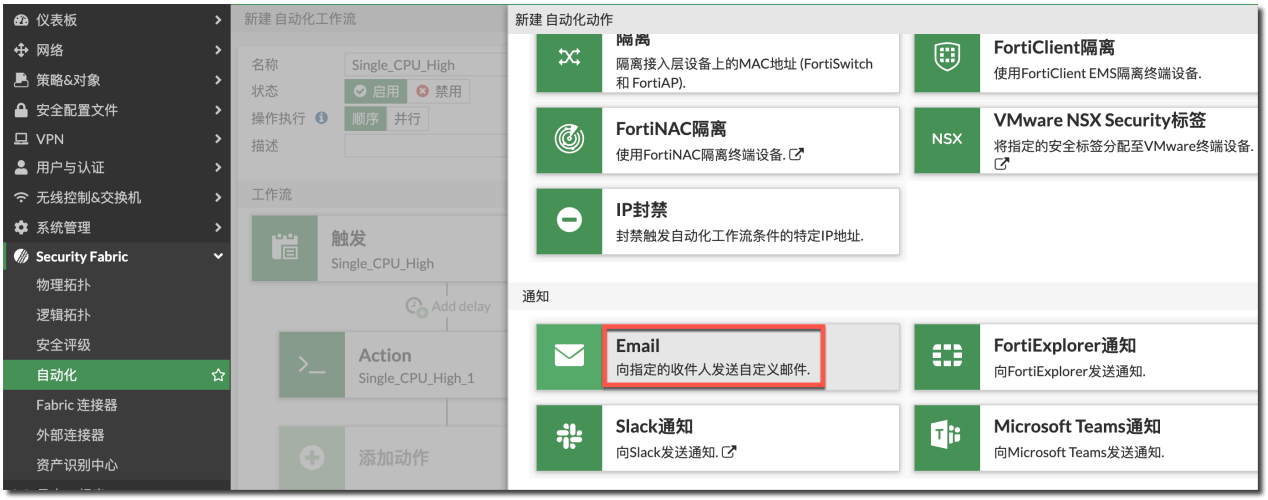
2. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



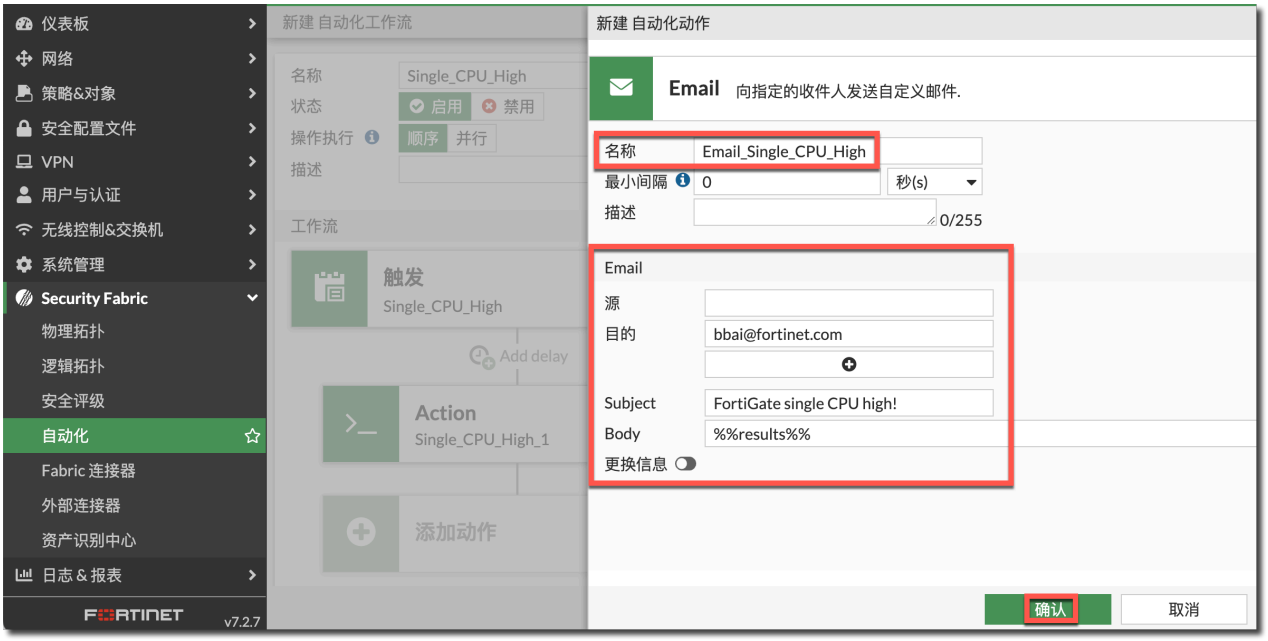
3. 再次点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



4. 在右侧弹出的窗口中，找到“通知”分类，点击“Email”选项。



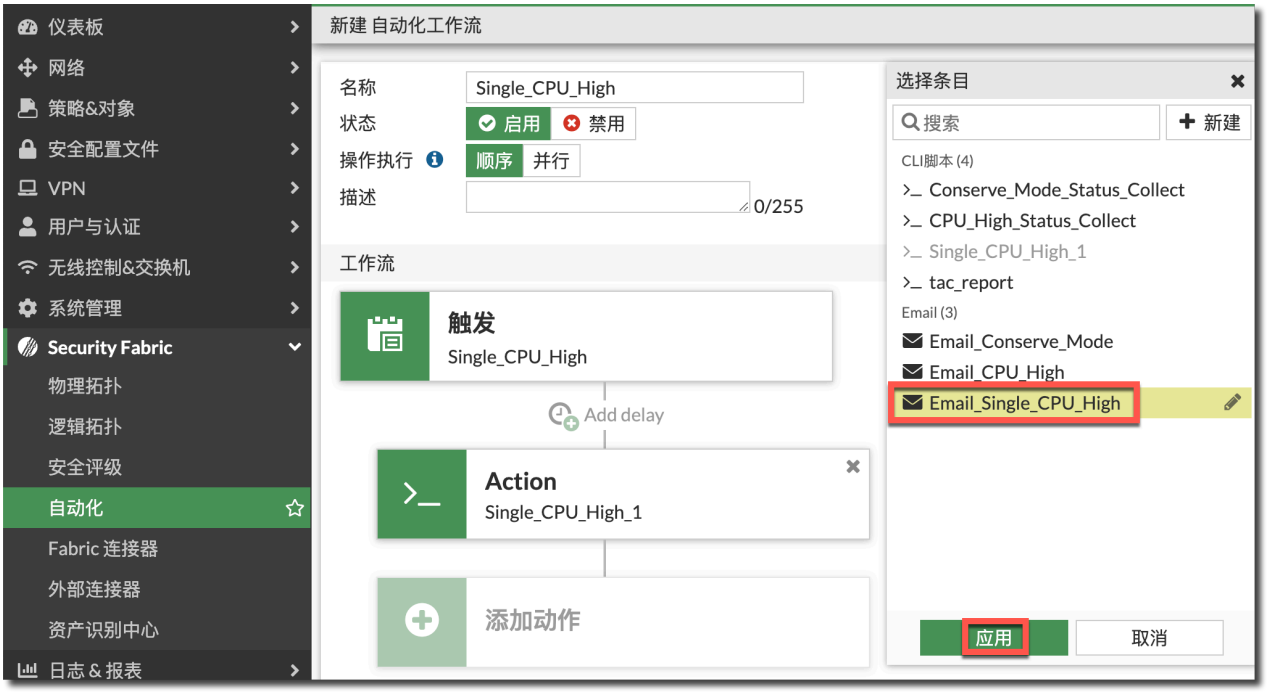
5. 配置动作（Action）的名称，以及邮件通知的收件人、邮件标题，邮件内容填写 `%%results%%`，这是一个变量，表示使用前序动作 CLI 命令执行的结果作为邮件正文，点击确认下发配置，发送第一个 CLI Action 执行的输出结果。



- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 DoNotReply@notification.fortinet.net ↗。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

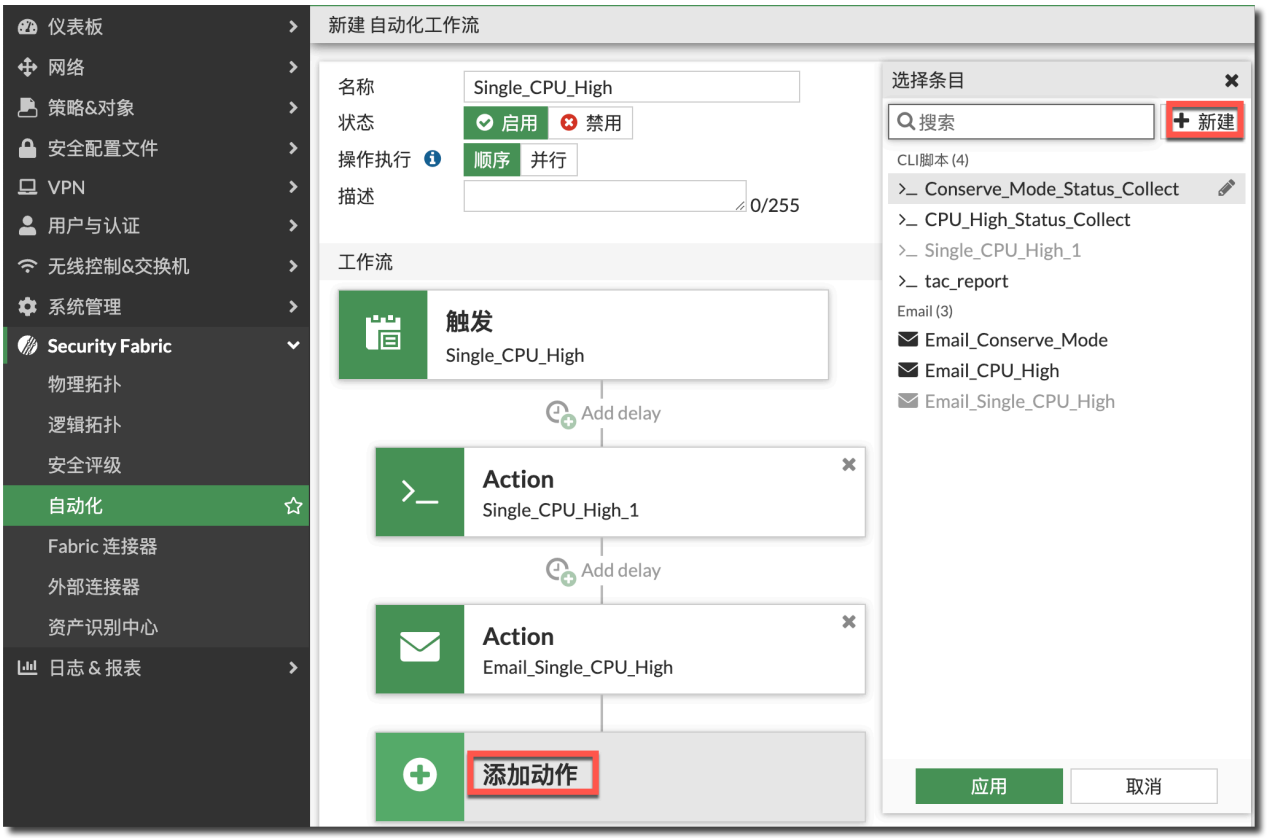
```bash
config system automation-action
    edit "Email_Single_CPU_High"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "FortiGate single CPU high!"
        set message "%%results%%"
    next
end
```

6. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



7. 再次点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。

> **重要**
>
> 参考步骤 11，在确认是哪个 CPU 核心高，但高的时间随机的情况下，才执行此步，否则跳过 17 ~ 25 步。



8. 在右侧弹出的窗口中，找到“通用”分类，点击“CLI 脚本”选项。

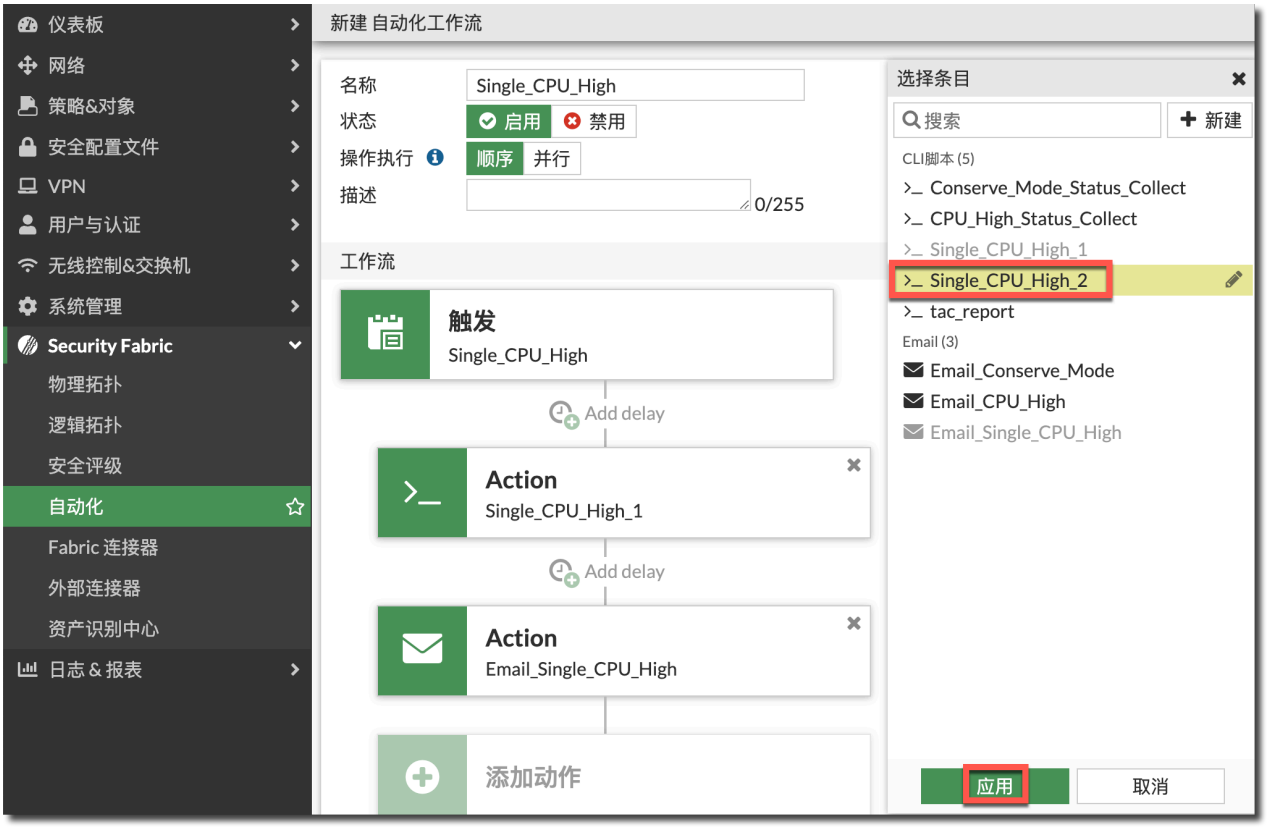


9. 配置动作（Action）的名称，配置 CLI 脚本内容为结束 CPU profile 信息收集并打印统计结果，点击确认下发配置。

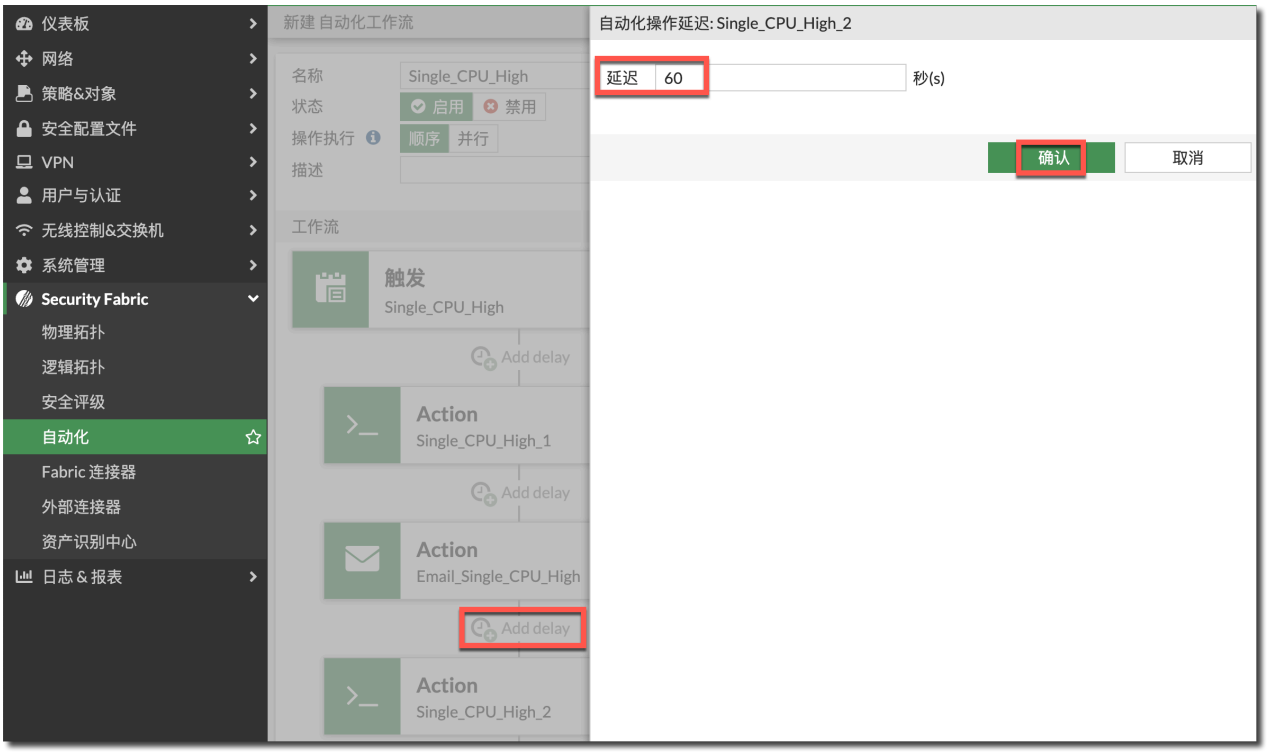



```bash
config system automation-action
    edit "Single_CPU_High_2"
        set action-type cli-script
        set script "diag sys profile stop
diag sys profile show order"
        set accprofile "super_admin"
    next
end
```

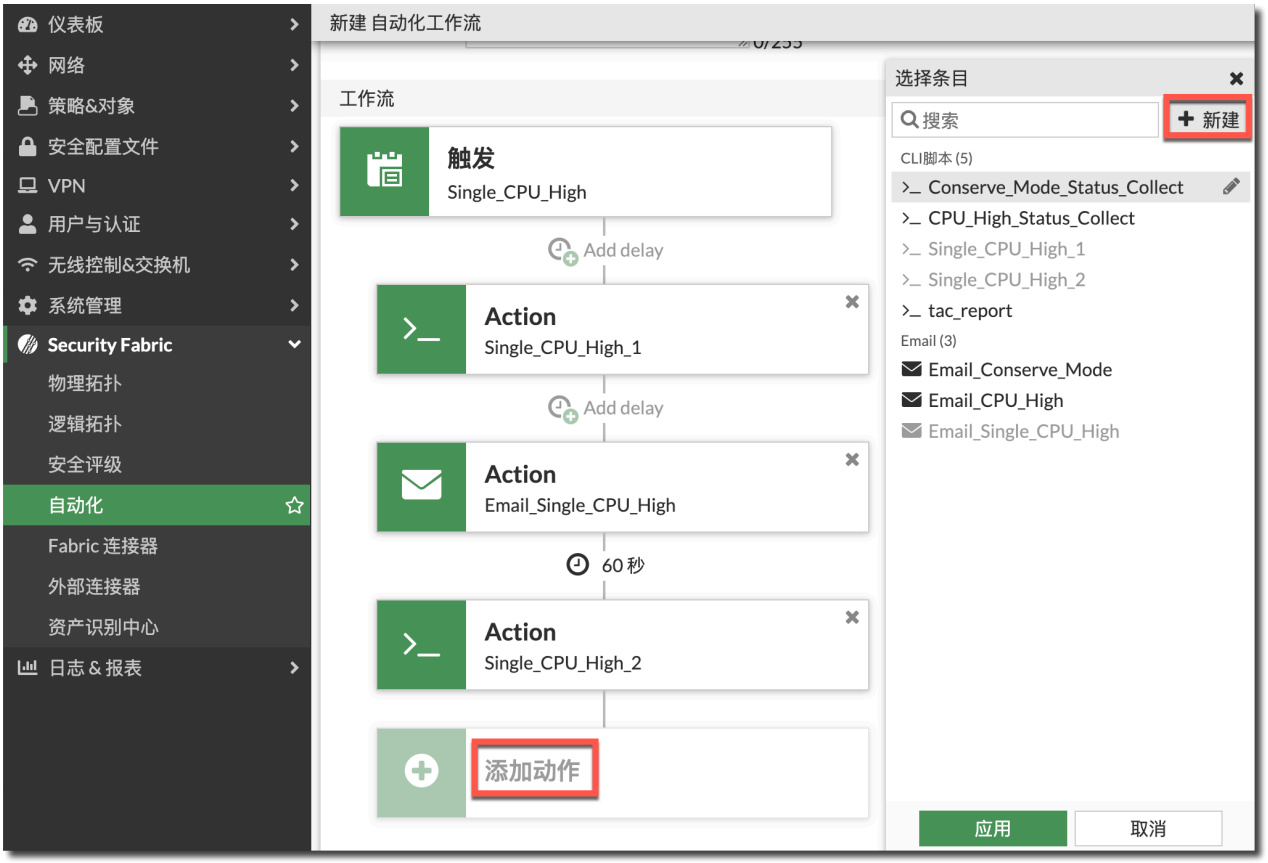
10. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



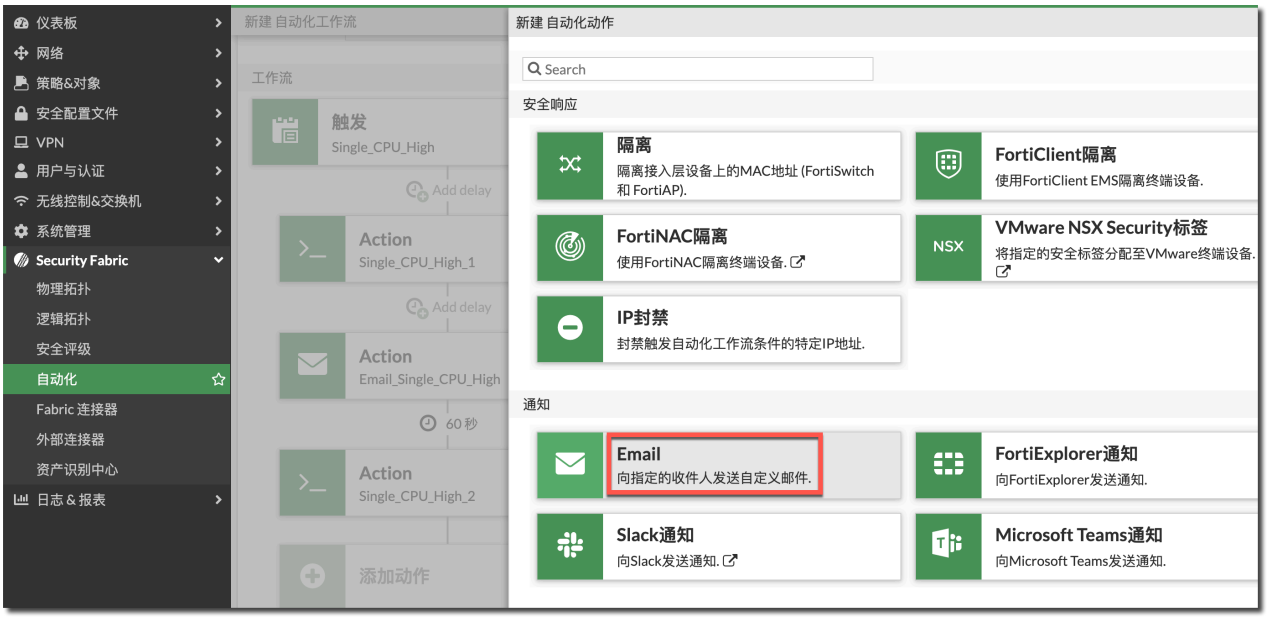
1. 在第 2 个 Action 和第 3 个 Action 之间点击“Add delay”，添加延迟为 60 秒，点击确认按钮下发配置。



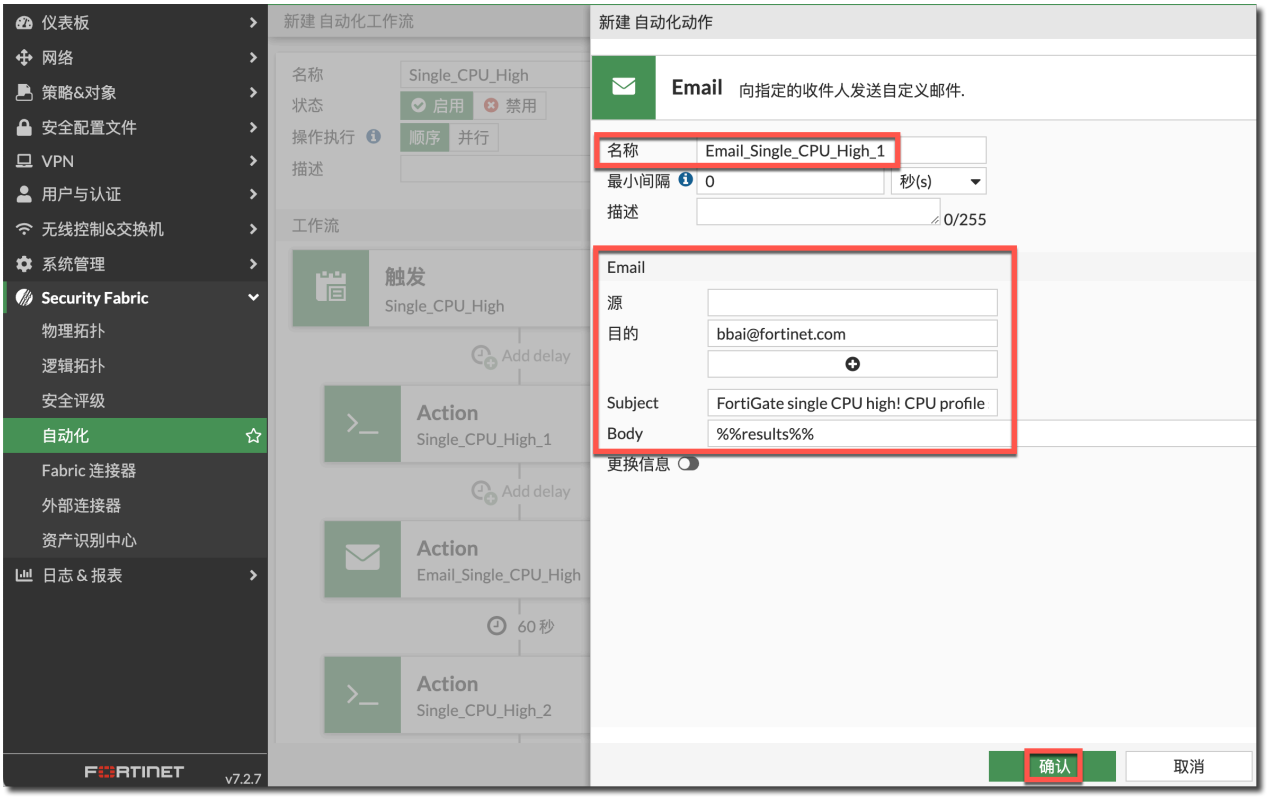
2. 再次点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



3. 在右侧弹出的窗口中，找到“通知”分类，点击“Email”选项。



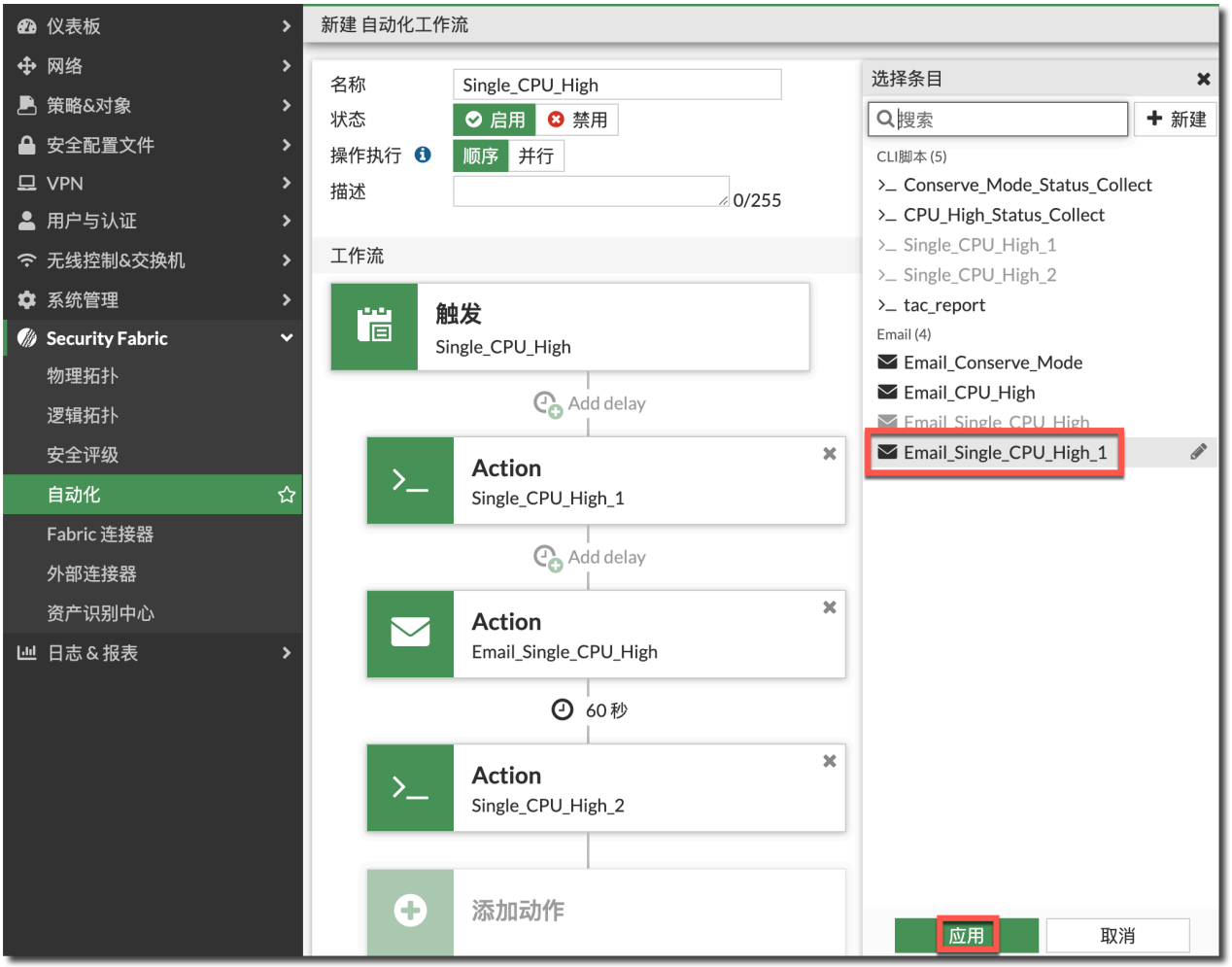
4. 配置动作（Action）的名称，以及邮件通知的收件人、邮件标题，邮件内容填写 `%%results%%`，这是一个变量，表示使用前序动作 CLI 命令执行的结果作为邮件正文，点击确认下发配置。



- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 DoNotReply@notification.fortinet.net ↗。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

```bash
config system automation-action
    edit "Email_Single_CPU_High"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "FortiGate single CPU high!"
        set message "%%results%%"
    next
end
```

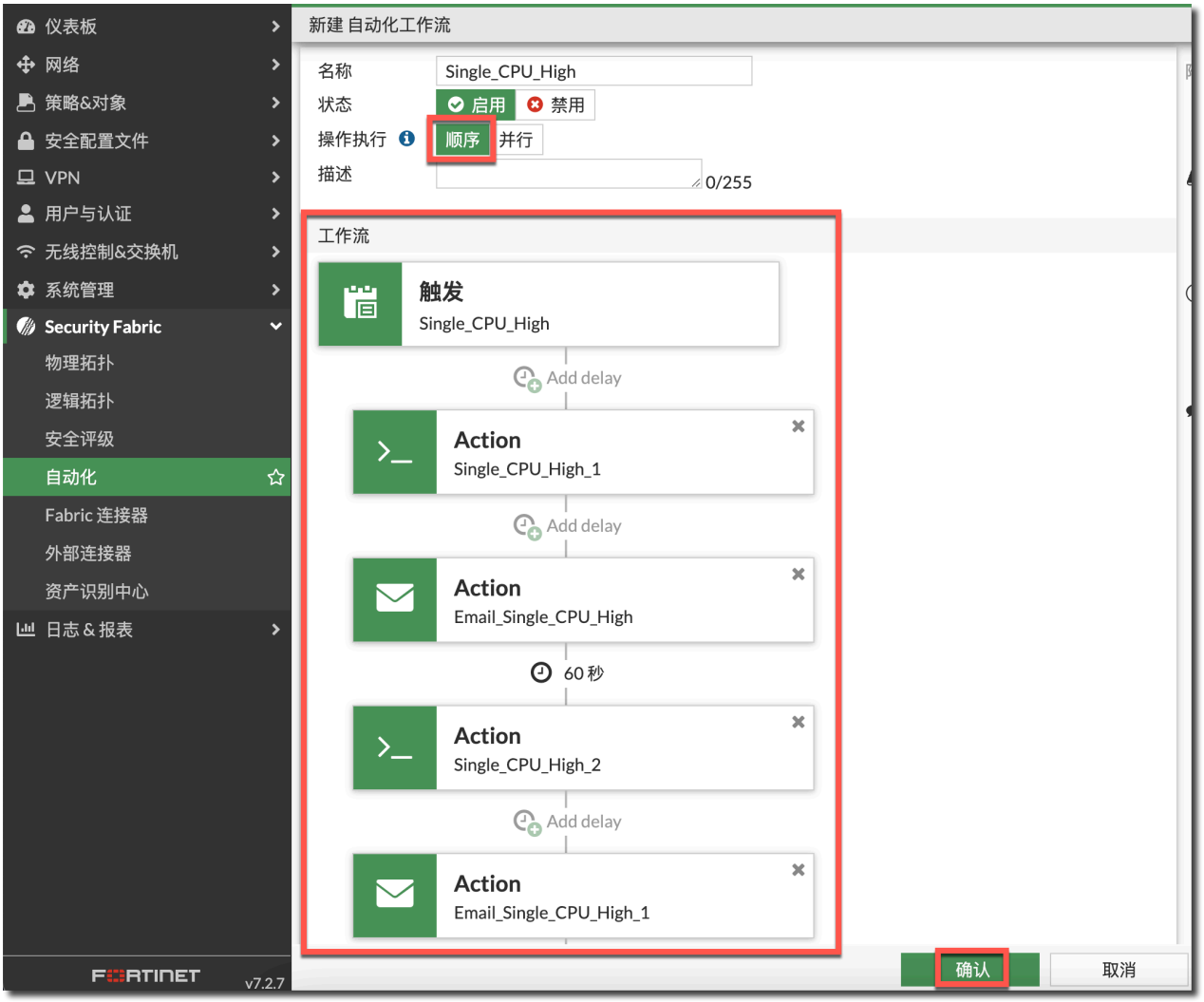
5. 在选择条目中选择上步创建的动作（Action），点击应用按钮。



6. 检查无误后，点击确认下发配置。

重要

参考步骤 11，在确认是哪个 CPU 核心高，但高的时间随机的情况下，才配置第 3 和第 4 个 Action。

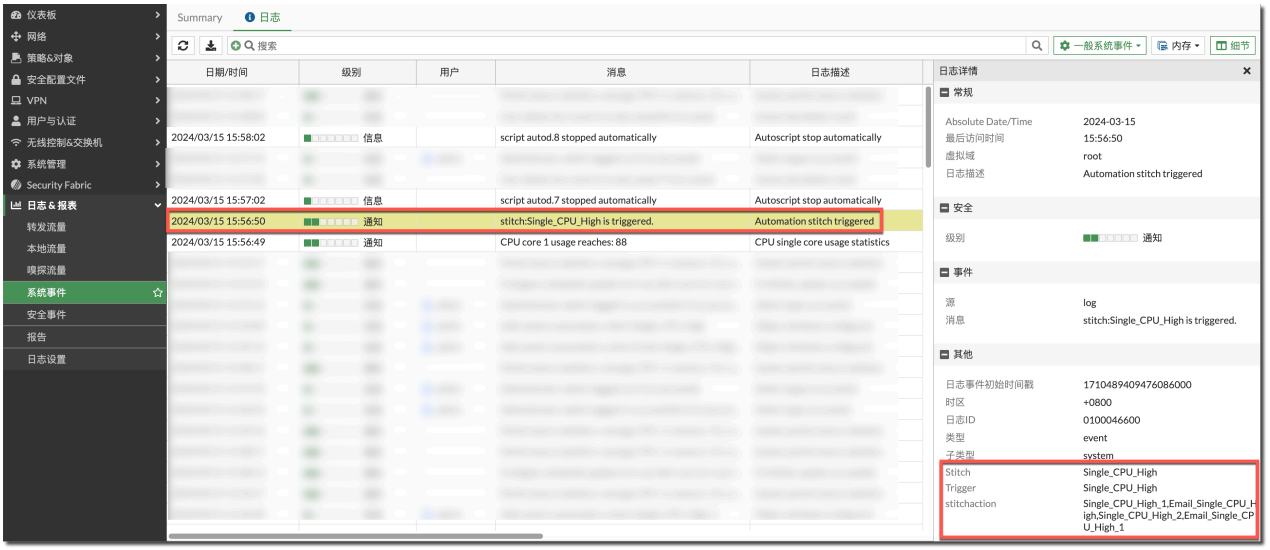



```bash
config system automation-stitch
    edit "Single_CPU_High"
        set trigger "Single_CPU_High"
        config actions
            edit 1
                set action "Single_CPU_High_1"
                set required enable
            next
            edit 2
                set action "Email_Single_CPU_High"
                set required enable
            next
            edit 3
                set action "Single_CPU_High_2"
                set delay 60
                set required enable
            next
            edit 4
                set action "Email_Single_CPU_High_1"
                set required enable
            next
        end
    next
end
```

配置验证

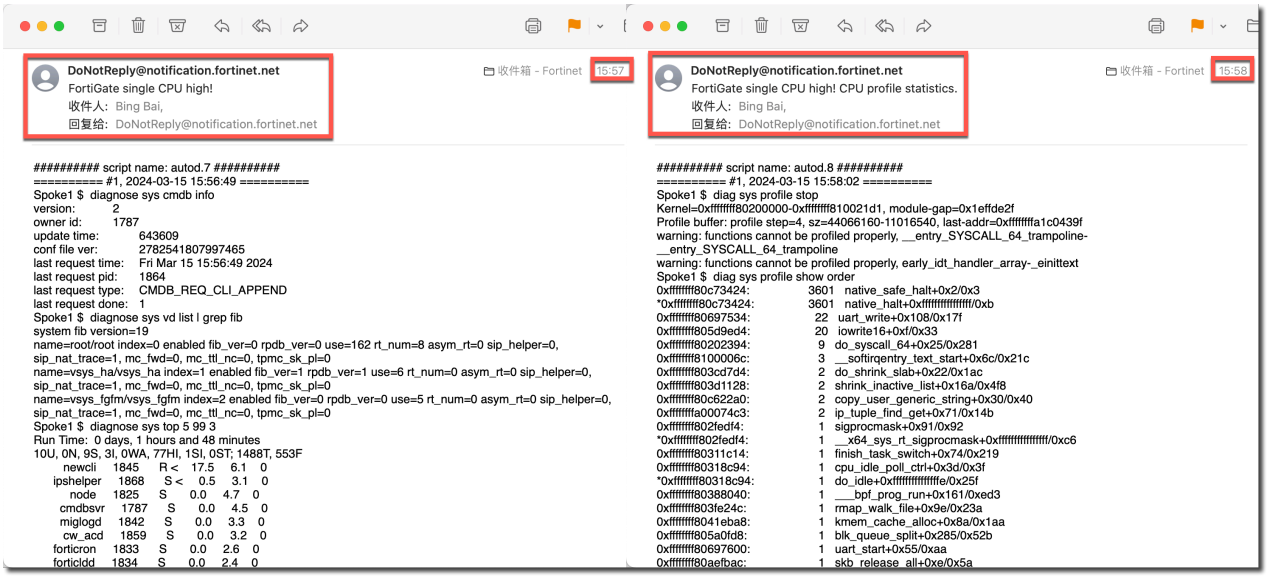
1. 触发 FortiGate 的 CPU 核心 0 利用率达到设定的阈值（此例中的 FortiGate CPU 高阈值设置为 85%）。

2. 触发自动化程序，相关自动化日志如下。




```bash
date=2024-03-15 time=15:56:50 eventtime=1710489409476085970 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="Single_CPU_High" trigger="Single_CPU_High"
stitchaction="Single_CPU_High_1,Email_Single_CPU_High,Single_CPU_High_2,Email_Single_CPU_High_1" from="log"
msg="stitch:Single_CPU_High is triggered."

date=2024-03-15 time=15:56:49 eventtime=1710489408805031532 tz="+0800" logid="0100040707" type="event" subtype="system"
level="notice" vd="root" logdesc="CPU single core usage statistics" action="cpu-single-core-usage" core=1 cpu=88
msg="CPU core 1 usage reaches: 88"
```

3. 收件人邮箱收到了 2 封告警邮件，第 1 封为除 CPU profile 统计信息外的收集信息，第 2 封为 CPU profile 的统计信息，发件人为默认的 DoNotReply@notification.fortinet.net ↗。



4. 查看自动化触发 Debug 信息（略，参考内存保护模式自动化的 Debug 信息）。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable
```

5. 查看发送邮件的 Debug 信息（略，参考内存保护模式自动化动作的邮件告警 Debug 信息）。

```bash
FortiGate # diagnose debug application alertmail -1
FortiGate # diagnose debug enable
```

注意事项

如果 FortiGate 的内存保护模式已经进入了 Extreme 状态，可能无法完成相关 CLI 命令的信息收集。

Email 动作

简介

本章节介绍 Email 动作的两个特性：

- 邮件内容可以指定为前序 CLI 动作执行的输出结果，或前序事件日志触发器中对应的日志内容。在此基础上，也可以指定日志中的特定字段，或是过滤符合某些条件的日志。
- 自定义邮件的格式。

自定义邮件内容

网络需求

用户想要在 FortiGate 的管理员登录或等出时，通过邮件通知登录和登出的系统事件日志信息，但只想得到系统事件日志中的源 IP、目的 IP、和登录/登出信息，过滤掉其他的日志内容。

配置步骤

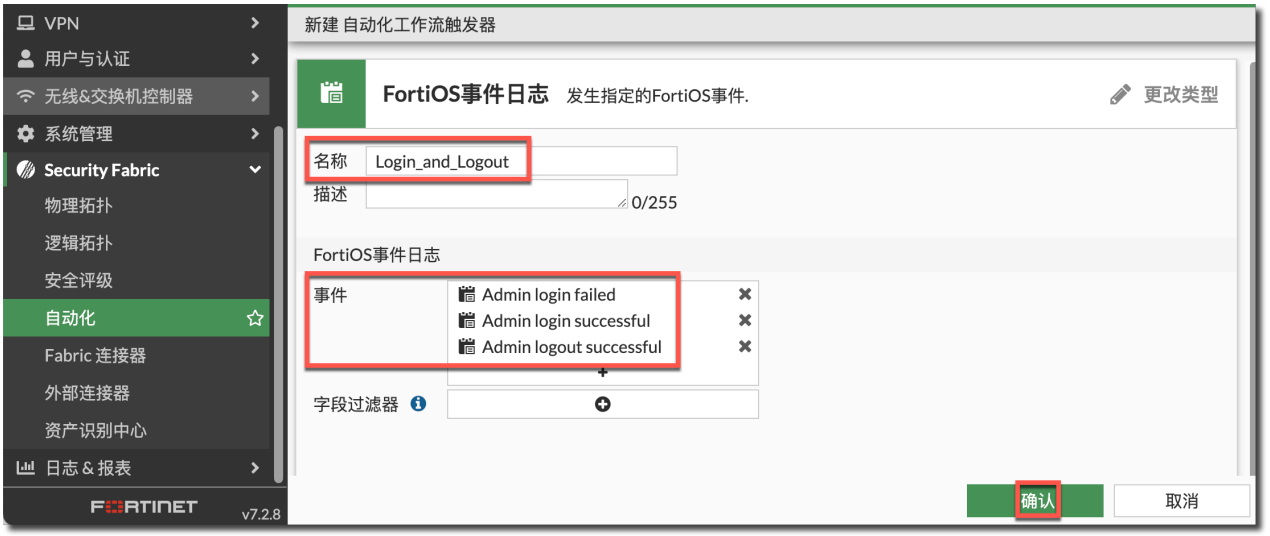
1. 进入 FortiGate 的“Security Fabric → 自动化 → 触发器”页面，点击“新建”按钮。

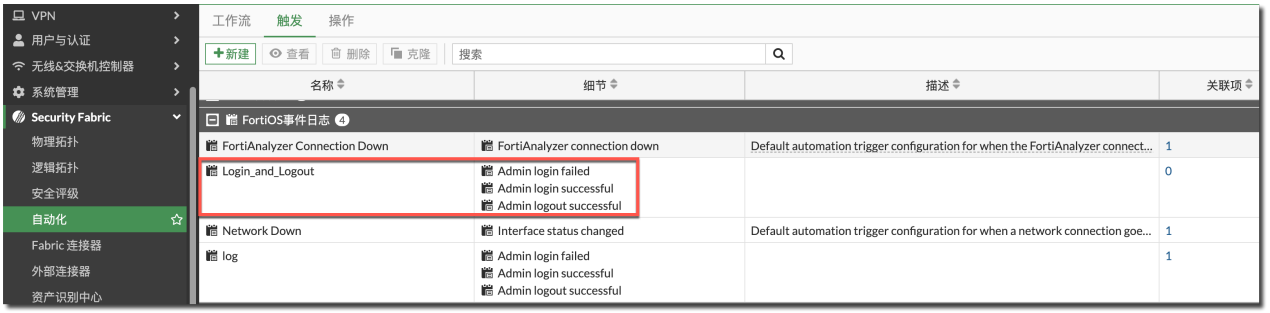


2. 在弹出的页面的“其他条件”分类中点击“FortiOS 事件日志”。



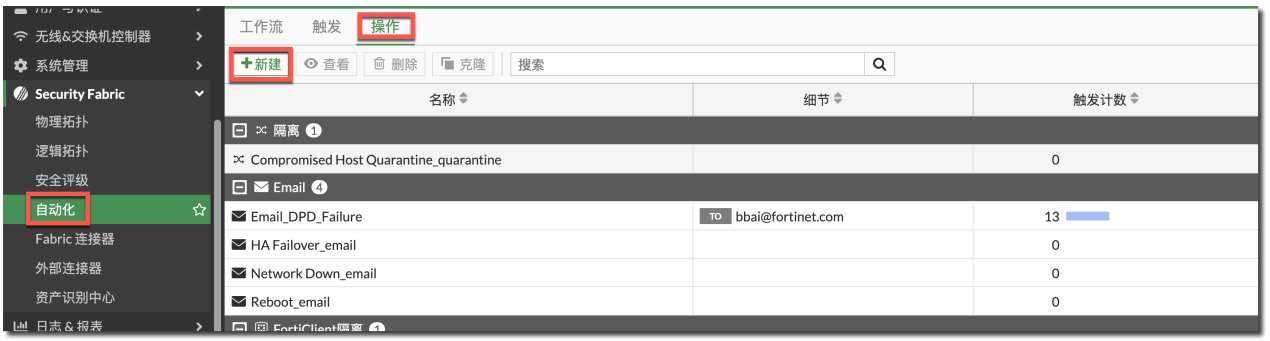
3. 配置触发器名称，事件选择管理员登录成功、失败、以及登出三种类型，点击确认下发配置。



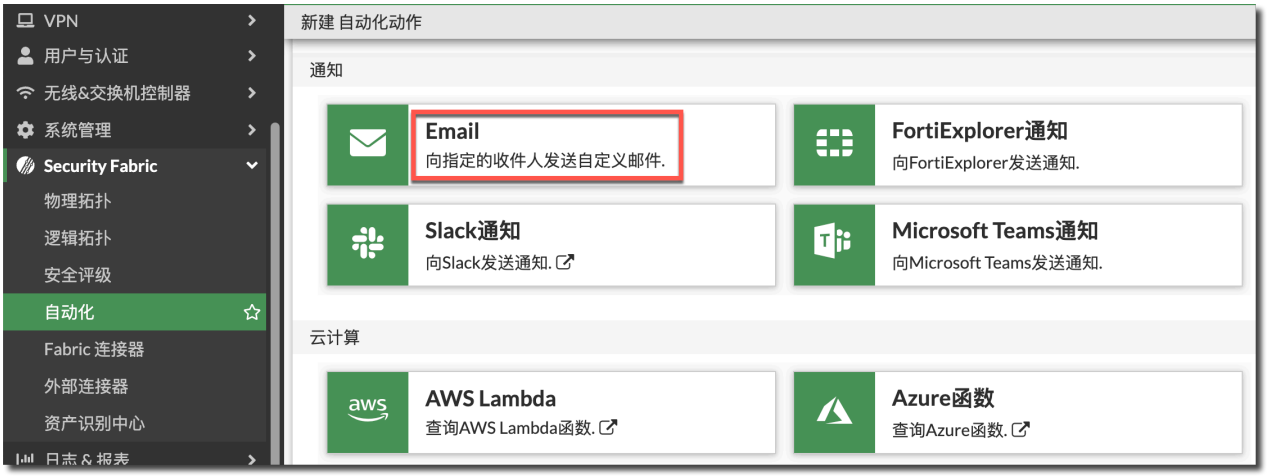



```bash
config system automation-trigger
    edit "Login_and_Logout"
        set event-type event-log
        set logid 32002 32001 32003
    next
end
```

4. 进入 FortiGate 的“Security Fabric → 自动化 → 操作”页面，点击“新建”按钮。



5. 在弹出的页面的“通知”分类中点击“Email”。



6. 配置动作名称，Email 的目的，以及邮件标题，在“Body”配置中，默认配置为 `%%log%%`，也就是前序日志触发器的完整日志内容，这里我们可以点击“Body”右侧的“%”按钮，会弹出一些其他的变量示例。

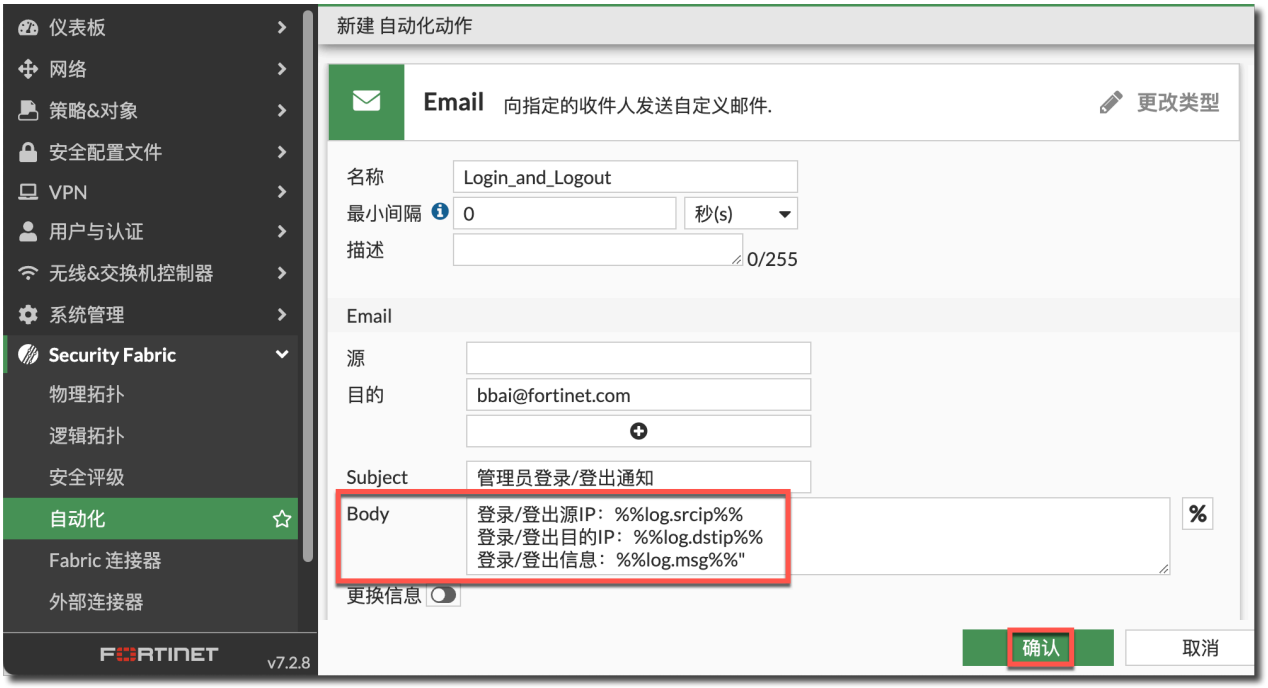



```bash
config system automation-action
    edit "Backup_config_to_FTP"
        set action-type cli-script
        set script "exe backup config ftp AutoBackup_%%date%%.conf 192.168.100.110 backup 111111"
        set accprofile "super_admin"
    next
end
```

- 最小间隔：如果 Trigger 频繁触发了，Action 执行的最小间隔。
- 源：指定邮件通知的发件人，默认为 DoNotReply@notification.fortinet.net ↗。
- 目的：配置邮箱的收件人，可以配置多个收件人。
- Body：邮件内容。
- 更换信息：可以更换为自定义内容和格式。

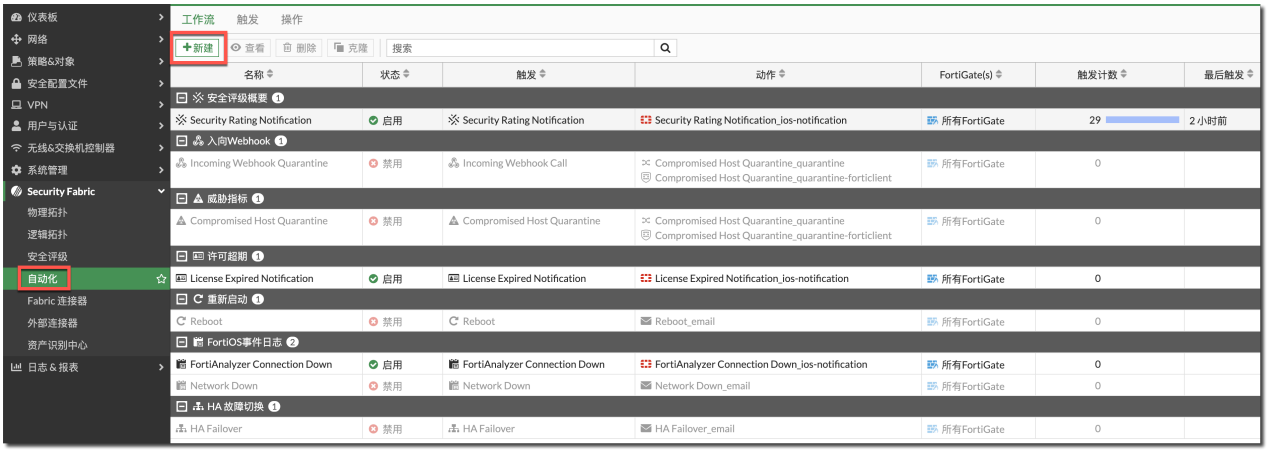
7. 按照上步的变量示例，来自定义发送邮件的内容，点击“确认”按钮下发配置。

> **重要**
>
> 内容包括三个日志字段的信息：登录/登出源 IP：`%%log.srcip%%`（表示从日志中的 `srcip` 字段取值）、登录/登出目的 IP：`%%log.dstip%%`（表示从日志中的 `dstip` 字段取值）、登录/登出信息：`%%log.msg%%"`（表示从日志中的 `msg` 字段取值）

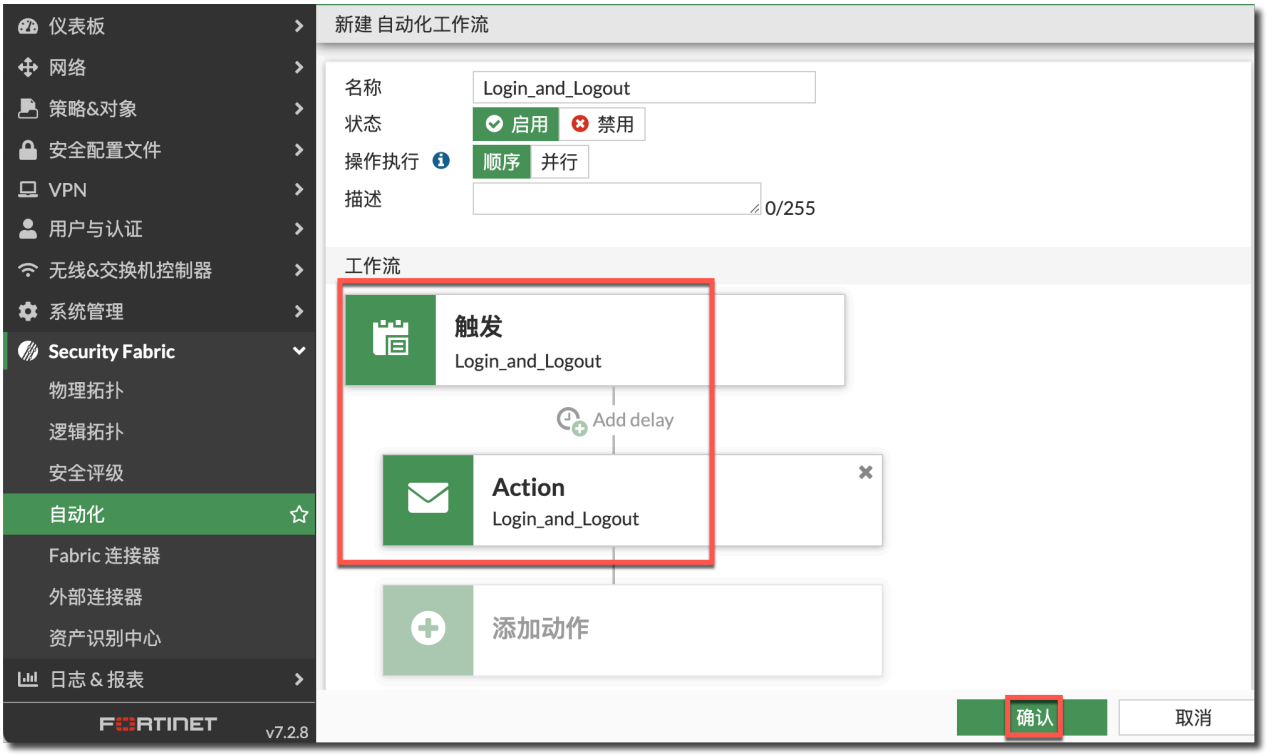



```bash
config system automation-action
    edit "Login_and_Logout"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "管理员登录/登出通知"
        set message "登录/登出源IP：%%log.srcip%%
登录/登出目的IP：%%log.dstip%%
登录/登出信息：%%log.msg%%\""
    next
end
```

8. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击“新建”按钮。



9. 配置工作流（Stitch 的）名称，在触发中选择步骤 3 创建的触发器，在 Action 中选择步骤 7 创建的动作，点击“确认”下发配置。




```bash
config system automation-stitch
    edit "Login_and_Logout"
        set trigger "Login_and_Logout"
        config actions
            edit 1
                set action "Login_and_Logout"
                set required enable
            next
        end
    next
end
```

配置验证

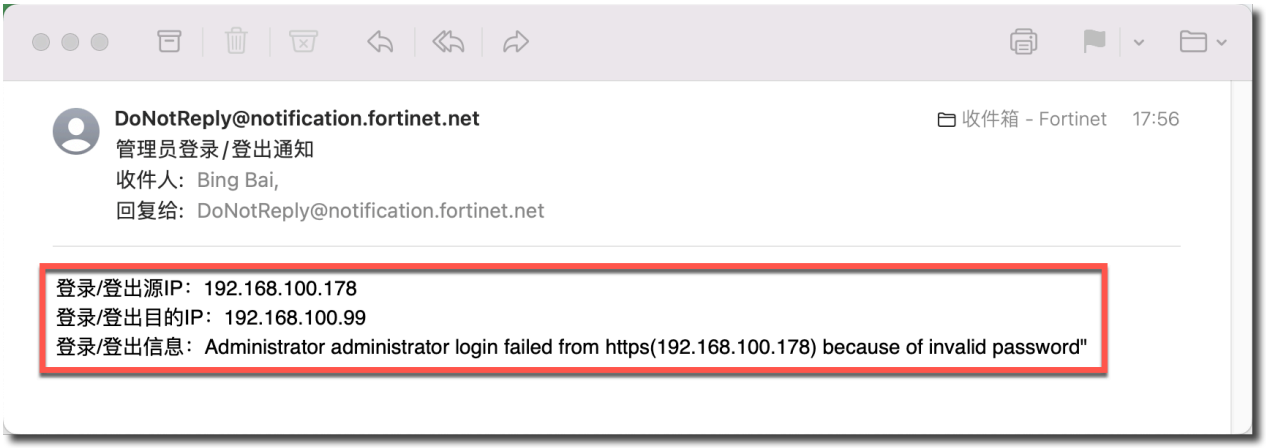
1. 触发 FortiGate 上的登录或登出日志，如下所示一次管理员登录失败的日志。

```bash
date=2024-03-20 time=17:56:17 eventtime=1710928577859214379 tz="+0800" logid="0100032002" type="event" subtype="system"
level="alert" vd="root" logdesc="Admin login failed" sn="0" user="administrator" ui="https(192.168.100.178)"
method="https" srcip=192.168.100.178 dstip=192.168.100.99 action="login" status="failed" reason="passwd_invalid"
msg="Administrator administrator login failed from https(192.168.100.178) because of invalid password"
```

2. 触发自动化程序，相关自动化日志如下。

```bash
date=2024-03-20 time=17:56:18 eventtime=1710928577859816539 tz="+0800" logid="0100046600" type="event" subtype="system"
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="Login_and_Logout" trigger="Login_and_Logout"
stitchaction="Login_and_Logout" from="log" msg="stitch:Login_and_Logout is triggered."
```

3. 收件人邮箱收到了告警邮件，邮件内容为自定义的日志字段内容，发件人为默认的 DoNotReply@notification.fortinet.net。



4. 对应的自动化触发 Debug 信息。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable
......
__action_email_hdl()-181: email action (Login_and_Logout) is called.
from:
to:bbai@fortinet.com;
subject:管理员登录/登出通知
......
```

5. 发送邮件的 Debug 信息如下。

```bash
FortiGate # diagnose debug application alertmail -1
FortiGate # diagnose debug enable

Arrived msg(type 9, 251 bytes):bbai@fortinet.com;

管理员登录/登出通知
登录/登出源IP：192.168.100.178
登录/登出目的IP：192.168.100.99
登录/登出信息：Administrator administrator login failed from https(192.168.100.178) because of invalid password"

mail_info:
        from:notification.fortinet.net  user:DoNotReply@notification.fortinet.net
mail_info:
        reverse path:DoNotReply@notification.fortinet.net
        user name:DoNotReply
to[0]:bbai@fortinet.com
⟸_init_mail_info
create session
resolve notification.fortinet.net to 1 IP
⟹ send mail
connecting to 208.91.114.151 port 465
send mail 0×a1b6d80 session 0×a1bc7e0
session_io_event: creating ssl structure for session 0×a1bc7e0
create_ssl: 0×7fa5d8a000
sessionn 0×a1bc7e0, SSL connected
session: 0×a1bc7e0, rsp_state: greeting, code: 220
session: 0×a1bc7e0, rsp_state: ehlo, code: 250
session: 0×a1bc7e0, rsp_state: mail, code: 250
session: 0×a1bc7e0, rsp_state: rcpt, code: 250
session: 0×a1bc7e0, rsp_state: data, code: 354
⟹ send: 登录/登出源IP：192.168.100.178
登录/登出目的IP：192.168.100.99
登录/登出信息：Administrator administrator login failed from https(192.168.100.178) because of invalid password"

session: 0×a1bc7e0, rsp_state: data2, code: 250
session: 0×a1bc7e0, rsp_state: quit, code: 221
session finined
_session_on_destroy
⟸ send mail success, m = 0×a1b6d80 s = 0×a1bc7e0
```

自定义邮件格式

网络需求

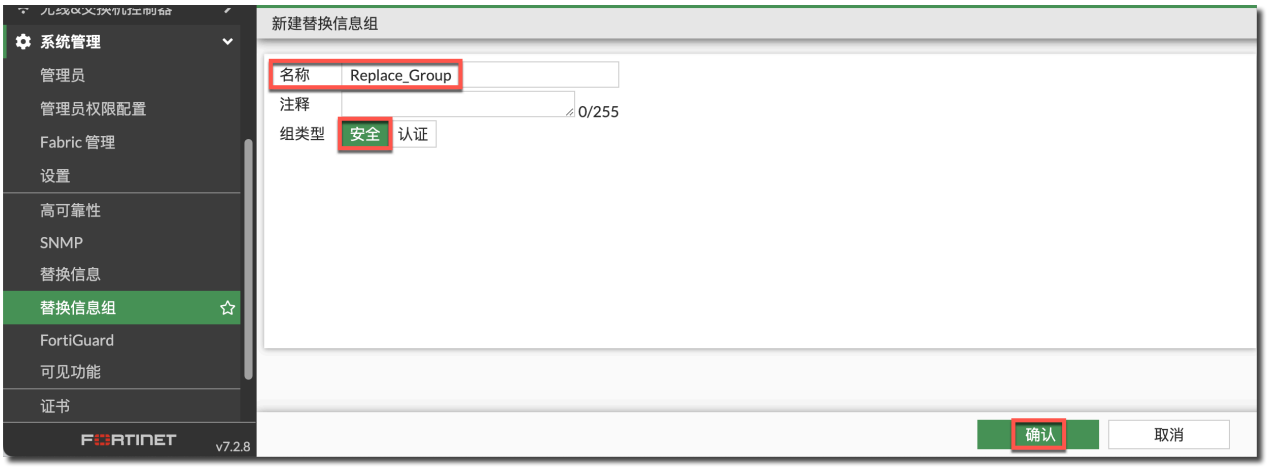
在上个章节的需求基础上，修改发送邮件的样式，添加一个告警图片。

配置步骤

1. 进入 FortiGate 的“系统管理 → 替换信息组”页面，点击“新建”按钮。



2. 配置替换信息组名称，“组类型”选择“安全”，点击“确认”按钮下发配置。



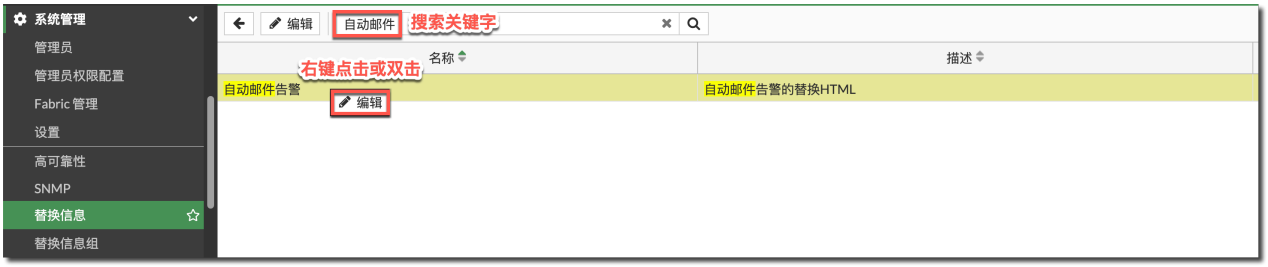
3. 右键点击新建的替换信息组，点击“编辑”按钮（或双击编辑）。



4. 搜索关键字“自动邮件”，找到“自动邮件告警”条目，右键或双击编辑该条目。

重要

替换信息组中也包含其他功能模块页面的替换信息，当用户需要替换多个功能模块的页面信息时，可以使用此功能。



5. 在 HTML 编辑页面，将想要添加的告警图片 URL 添加到 `message-container` 部分，如下图所示（`<img src="https://www.fortinet.com/content/dam/fortinet/images/icons/benefits/icon-wireless-fortinet-ips.svg" alt="Fortinet Logo"/>`），左侧的预览可以正常显示该图片，点击“备份配置”按钮保存配置。

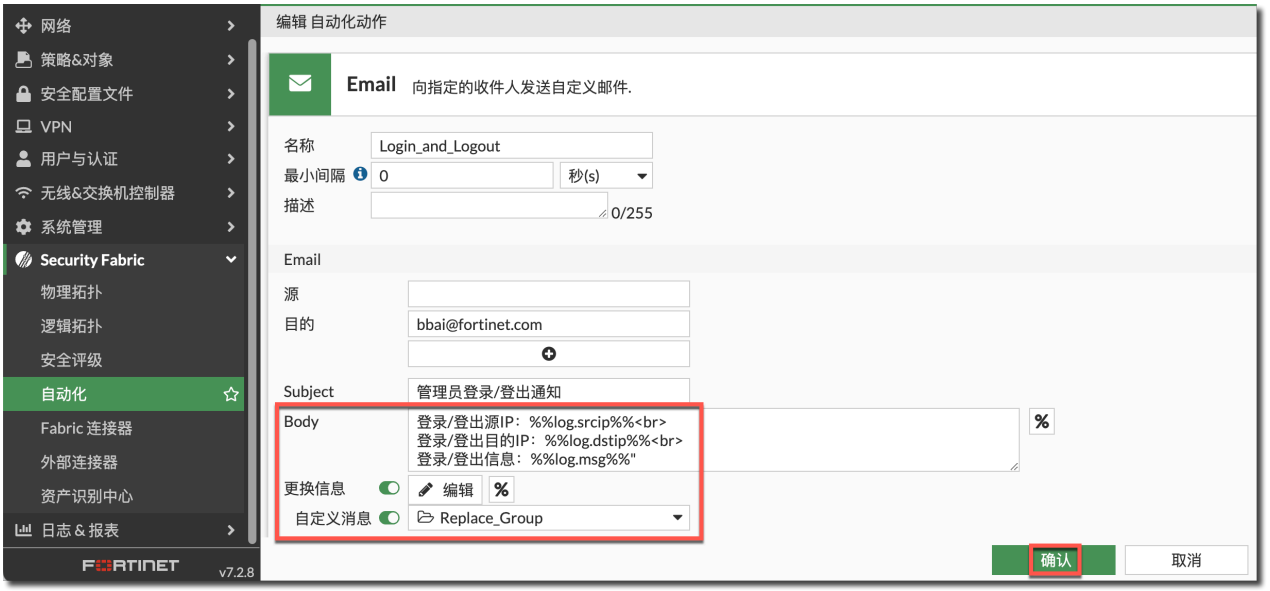



```bash
config system replacemsg-group
    edit "Replace_Group"
        set comment ''
        set group-type utm
        config automation
            edit "automation-email"
                set buffer "<!DOCTYPE html>
<html lang=\"en\">
 <head>
 <meta charset=\"UTF-8\">
 <meta http-equiv=\"X-UA-Compatible\" content=\"IE=8; IE=EDGE\">
 <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">
 <link href=\"https://fonts.googleapis.com/css?family=Roboto&display=swap\" rel=\"stylesheet\">
 <style>
 body {
 height: 100%;
 font-family: Roboto, Helvetica, Arial, sans-serif;
 margin: 0;
 display: flex;
 align-items: center;
 justify-content: center;
 }
 .message-container{
 margin: 0 auto;
 max-width: 580px;
 }
 .email-body {
 line-height: 1.5em;
 }
 </style>
 </head>
 <body>
 <div class=\"message-container\">
 <img src=\"https://www.fortinet.com/content/dam/fortinet/images/icons/benefits/icon-wireless-fortinet-ips.svg\"
```

```
alt=\"Fortinet Logo\"/>
 <h1>
 Security Fabric Automation
 </h1>
 <h3>
 %%AUTOMATION_FGT_SERIAL%%: %%AUTOMATION_STITCH_NAME%%
 </h3>
 <div class=\"email-body\">
 %%AUTOMATION_EMAIL_BODY%%
 </div>
 </div>
 </body>
</html>
"
                set header http
                set format html
            next
        end
    next
end
```

6. 进入 FortiGate 的“Security Fabric → 自动化 → 操作”页面，编辑上个章节创建的动作“Login_and_Logout”。开启“更换信息”选项，然后开启“自定义消息”选项，并选择上步创建的替换信息组。注意由于调用了替换信息组（HTML 格式），所以需要在“Body”中每行加入 HTML 换行符 `<br>`。点击“确认”按钮下发配置。

> **重要**
>
> 如果仅需要替换此邮件动作的格式，则不需要步骤 1~5 创建替换信息组，只在此页面开启“更换信息”选项并点击编辑按钮编辑 HTML 格式即可。

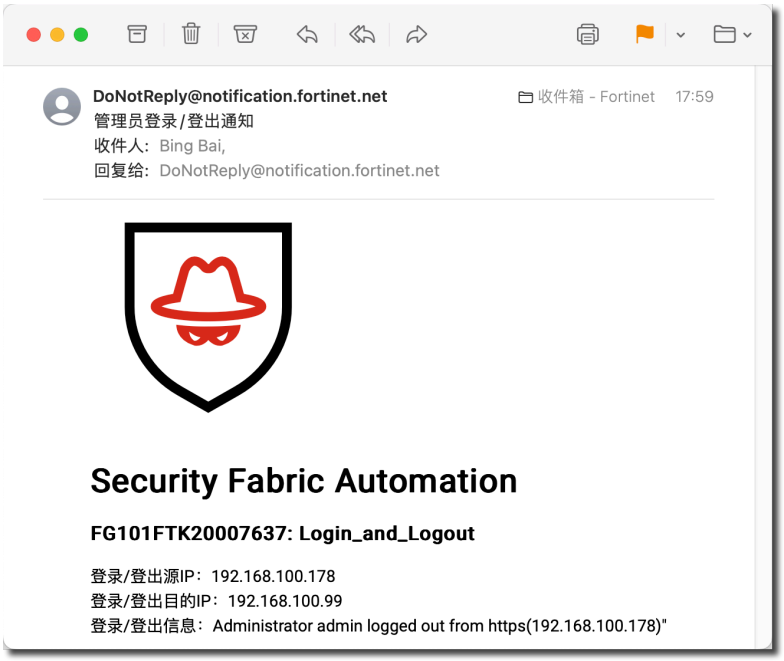



```
config system automation-action
    edit "Login_and_Logout"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "管理员登录/登出通知"
        set message "登录/登出源IP：%%log.srcip%%<br>
登录/登出目的IP：%%log.dstip%%<br>
登录/登出信息：%%log.msg%%\""
        set replacement-message enable
        set replacemsg-group "Replace_Group"
    next
end
```

bash

配置验证

触发 FortiGate 上的登录或登出日志，查看收件人收到的邮件格式按照替换信息组中的配置的样式显示。


DoNotReply@notification.fortinet.net
收件箱 - Fortinet 17:59
管理员登录/登出通知
收件人：Bing Bai,
回复给：DoNotReply@notification.fortinet.net
Security Fabric Automation
FG101FTK20007637: Login_and_Logout
登录/登出源IP：192.168.100.178
登录/登出目的IP：192.168.100.99
登录/登出信息：Administrator admin logged out from https(192.168.100.178)"

Teams Webhook

简介

Microsoft Teams 通知动作可以通过 Webhook 将自动化执行的结果信息发送到指定的 Teams 频道。

需要 FortiGate 可以访问 Internet。

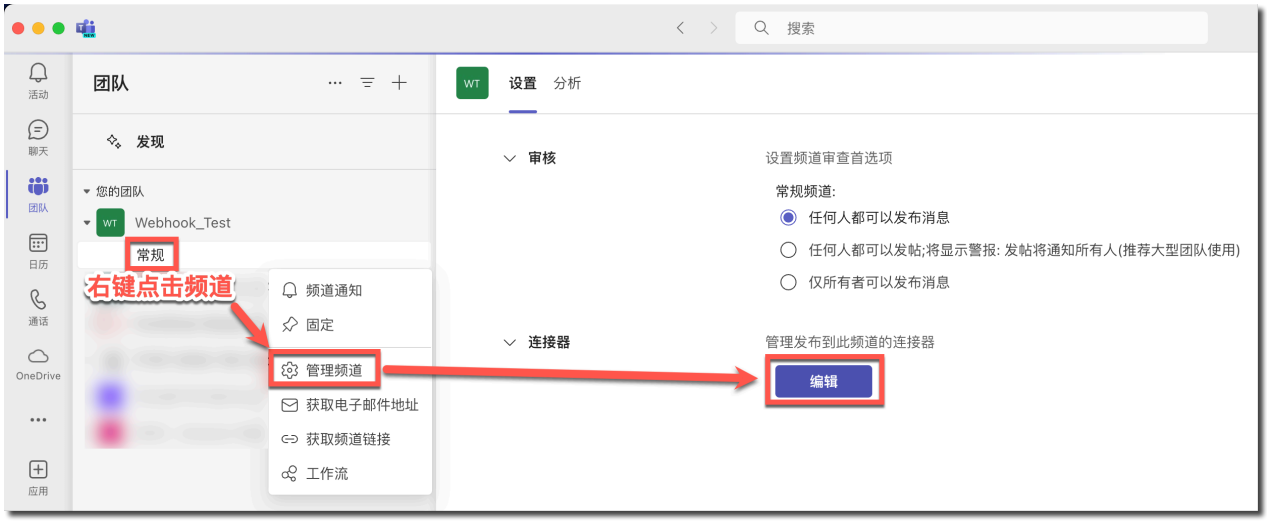
网络需求

当 FortiGate 的配置发生变更时，向 Teams 对应频道发送通知消息。

配置步骤

获取 Teams Webhook 链接

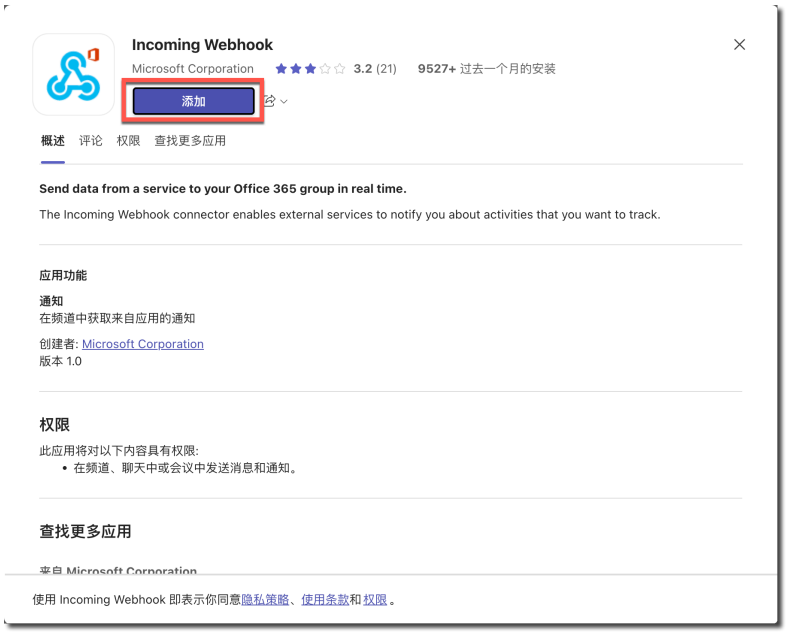
1. 打开 Teams，进入团队页面，右键点击需要接收消息的频道，选择“管理频道”，在“连接器”中点击“编辑”按钮。



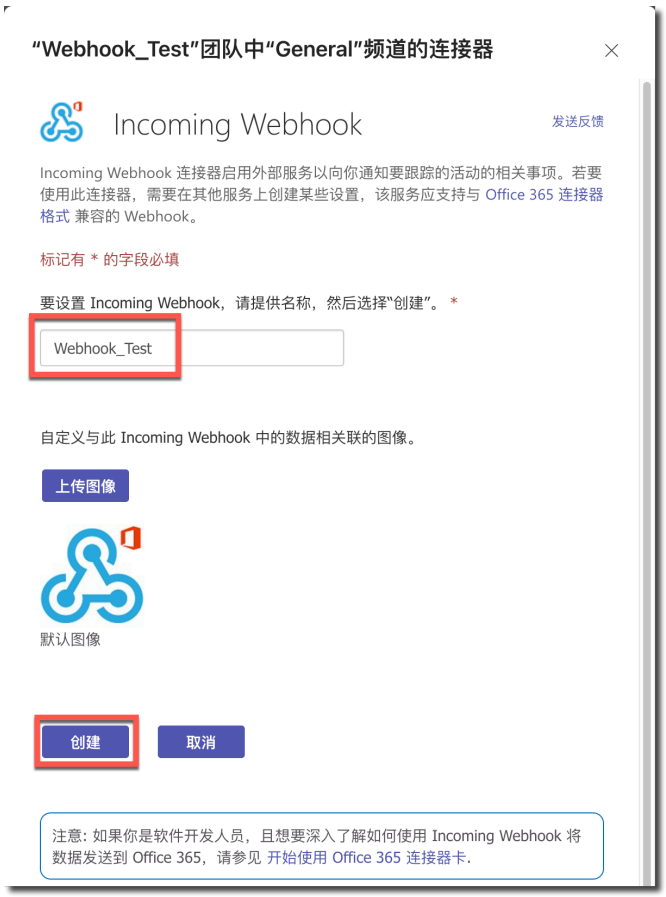
2. 在弹出的连接器列表中选择“Incoming Webhook”，点击该连接器右侧的“添加”按钮。



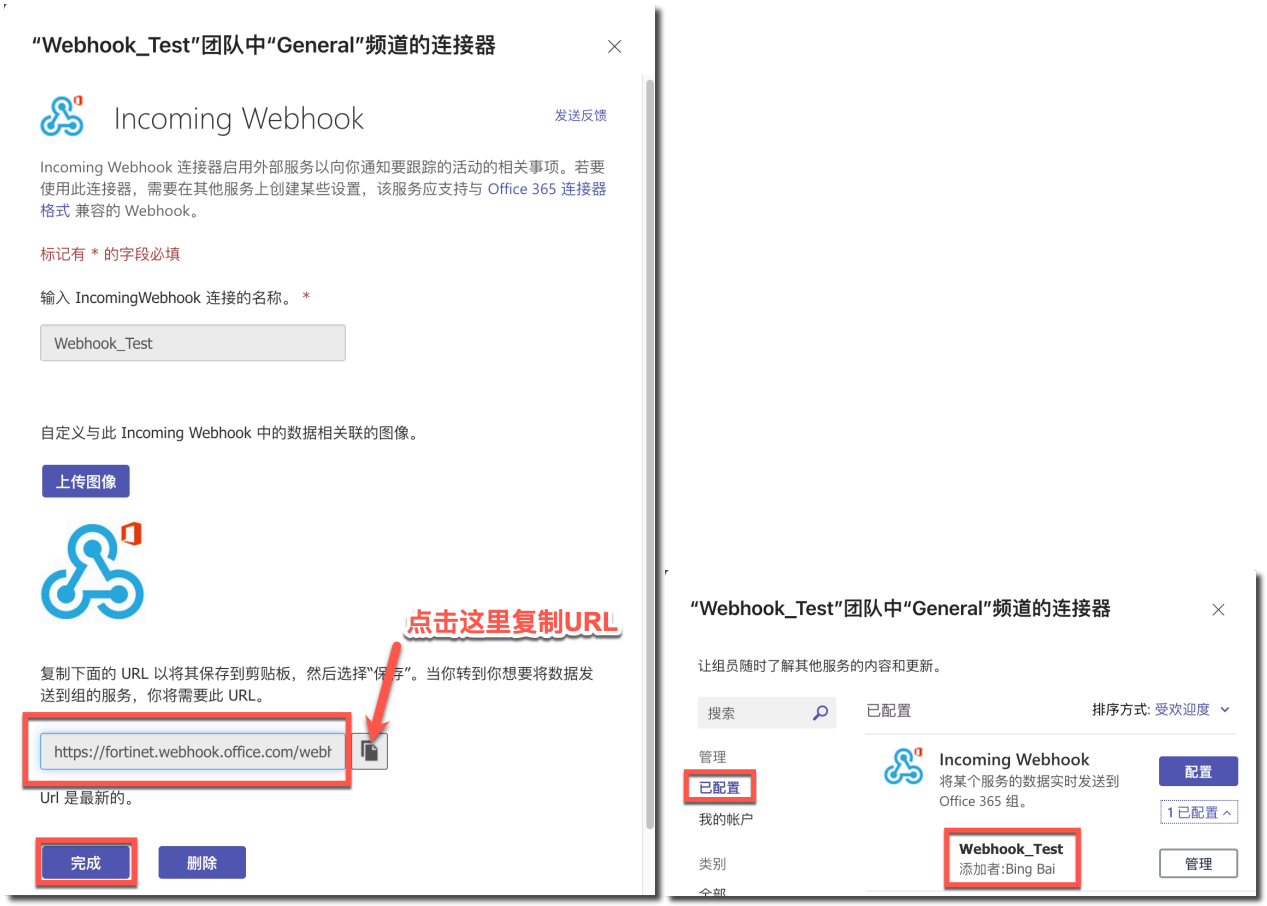
3. 在弹出的连接器详情页面中，继续点击“添加”按钮。



4. 随后会自动弹出连接器配置界面，配置连接器名称，可自行选择性上传发送消息时显示的图标，点击“创建”按钮。

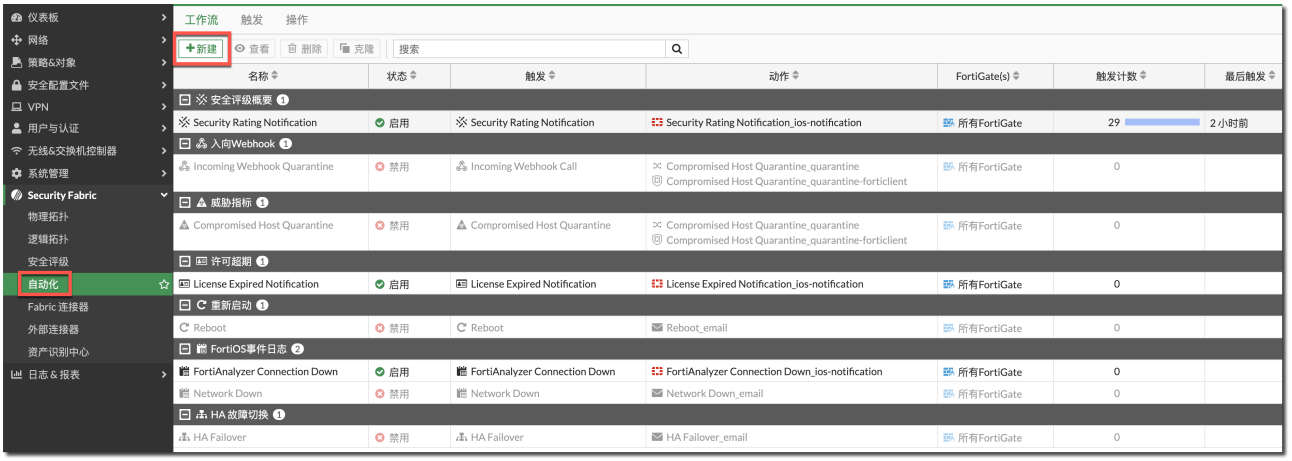


5. 复制生成的 Webhook URL 并妥善保存，点击“完成”按钮，完成“Incoming Webhook”连接器的配置。



FortiGate 自动化配置

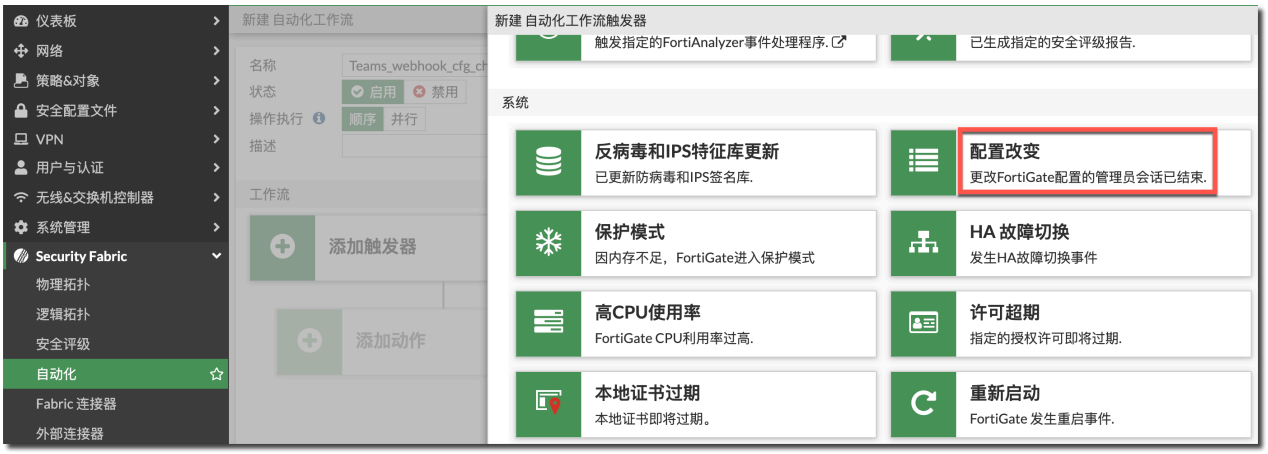
1. 进入 FortiGate 的“Security Fabric → 自动化 → 工作流”页面，点击“新建”按钮。



2. 配置工作流的名称，点击“添加触发器”按钮，在右侧的“选择条目”中点击“新建”按钮。



3. 在右侧弹出的窗口中，找到“系统”分类，点击“配置改变”选项。

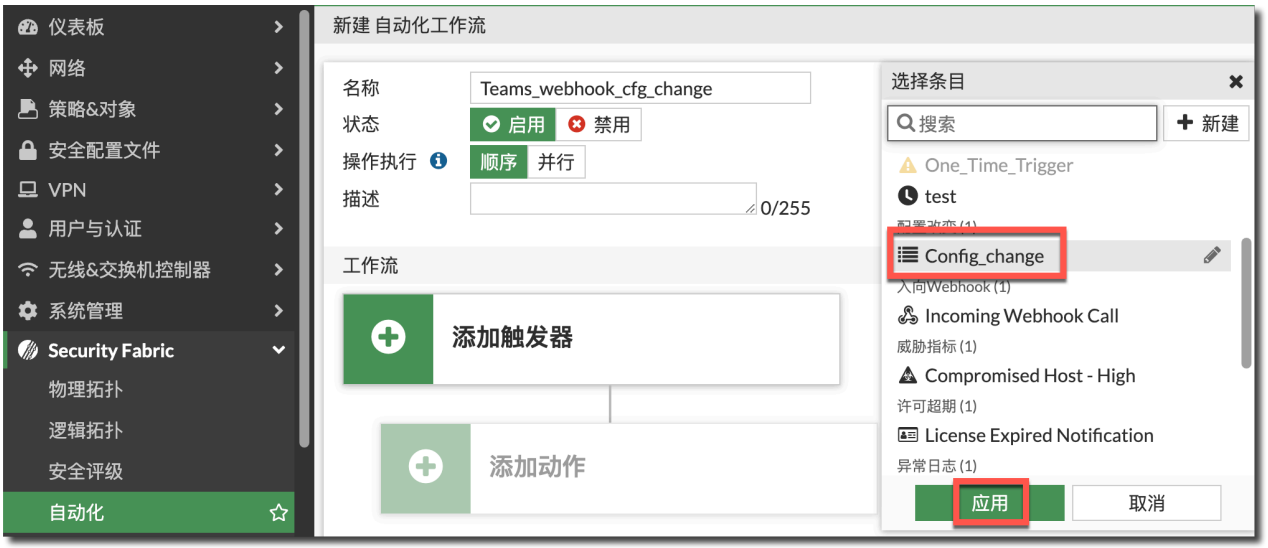


4. 配置触发器（Trigger）的名称，点击确认下发配置。

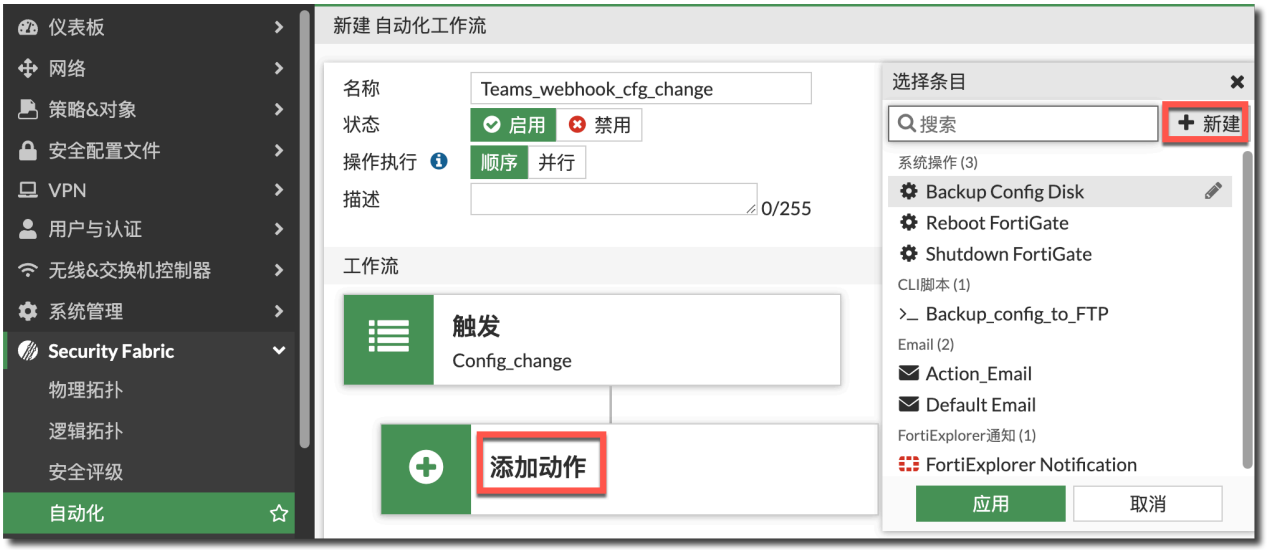



```bash
config system automation-trigger
    edit "Config_change"
        set event-type config-change
    next
end
```

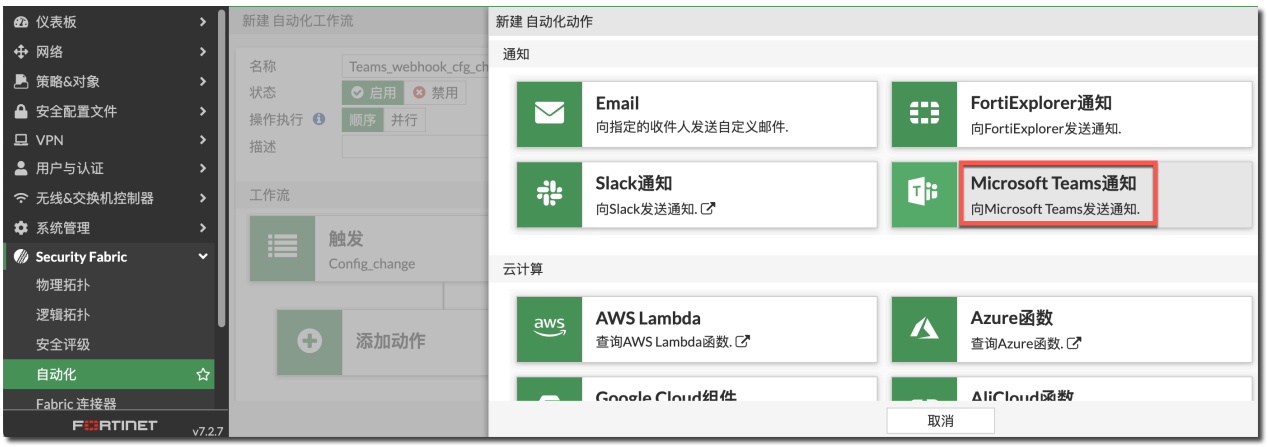
5. 在选择条目中选择上步创建的触发器（Trigger），点击应用按钮。



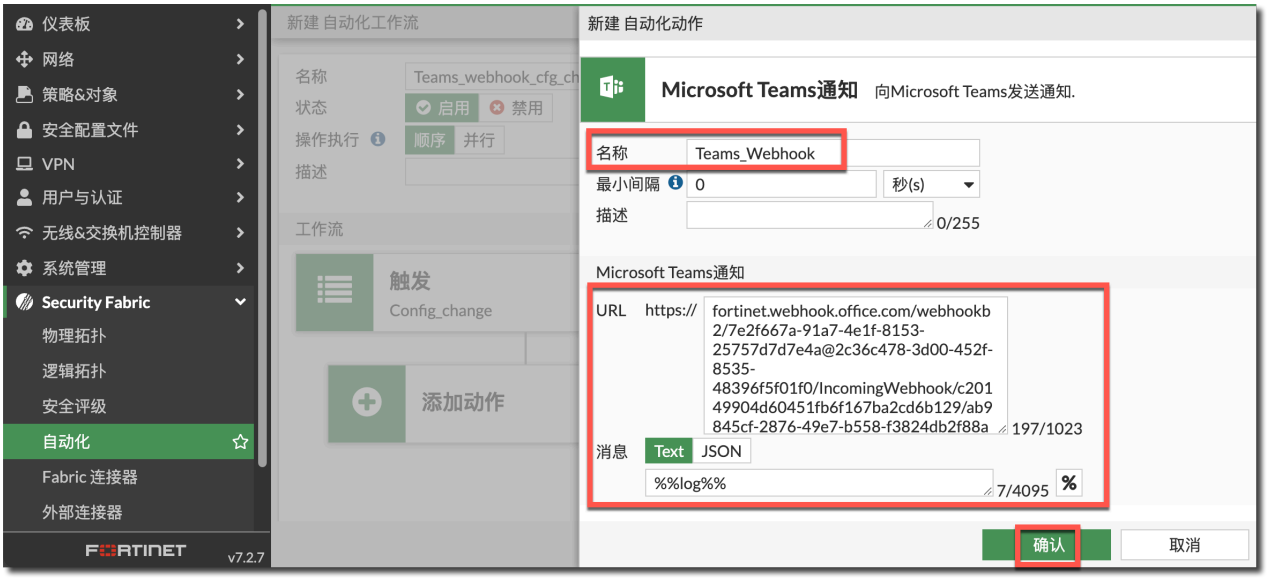
6. 点击“添加动作”按钮，在右侧弹出的“选择条目”中点击“新建”按钮。



7. 在右侧弹出的窗口中，找到“通知”分类，点击“Microsoft Teams 通知”选项。



8. 配置动作（Action）的名称，将从 Teams 连接器中复制出的 Webhook URL 粘贴进“URL”选项（不包含 `https` 字段），消息格式这里以“Text”为例，消息内容为配置变更的日志信息，点击确认下发配置。

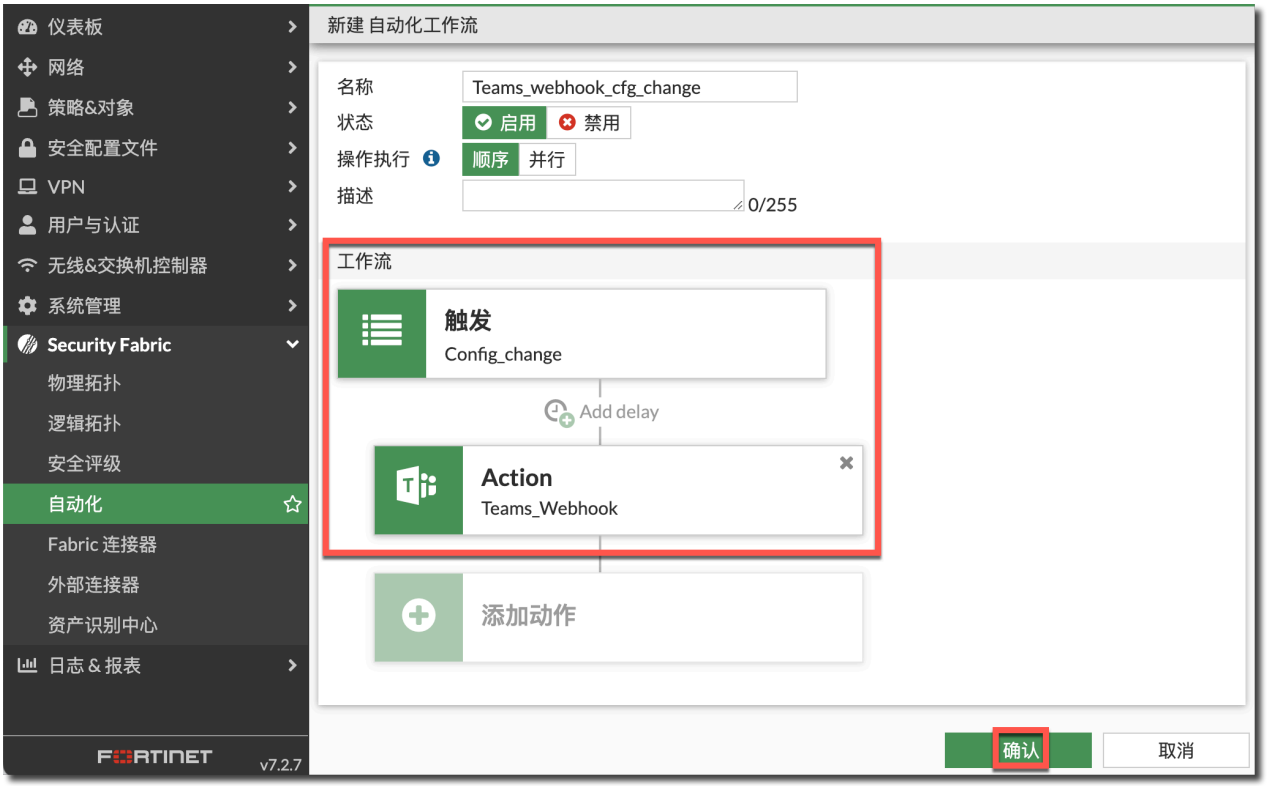


9. 在选择条目中选择上步创建的动作（Action），点击应用按钮。




```bash
config system automation-action
    edit "Teams_Webhook"
        set action-type microsoft-teams-notification
        set uri "fortinet.webhook.office.com/webhookb2/7e2f667a-91a7-4e1f-8153-25757d7d7e4a@2c36c478-3d00-452f-8535-48396f5f01f0/IncomingWebhook/c20149904d60451fb6f167ba2cd6b129/ab9845cf-2876-49e7-b558-f3824db2f88a"
    next
end
```

.0. 检查无误后，点击确认下发配置。

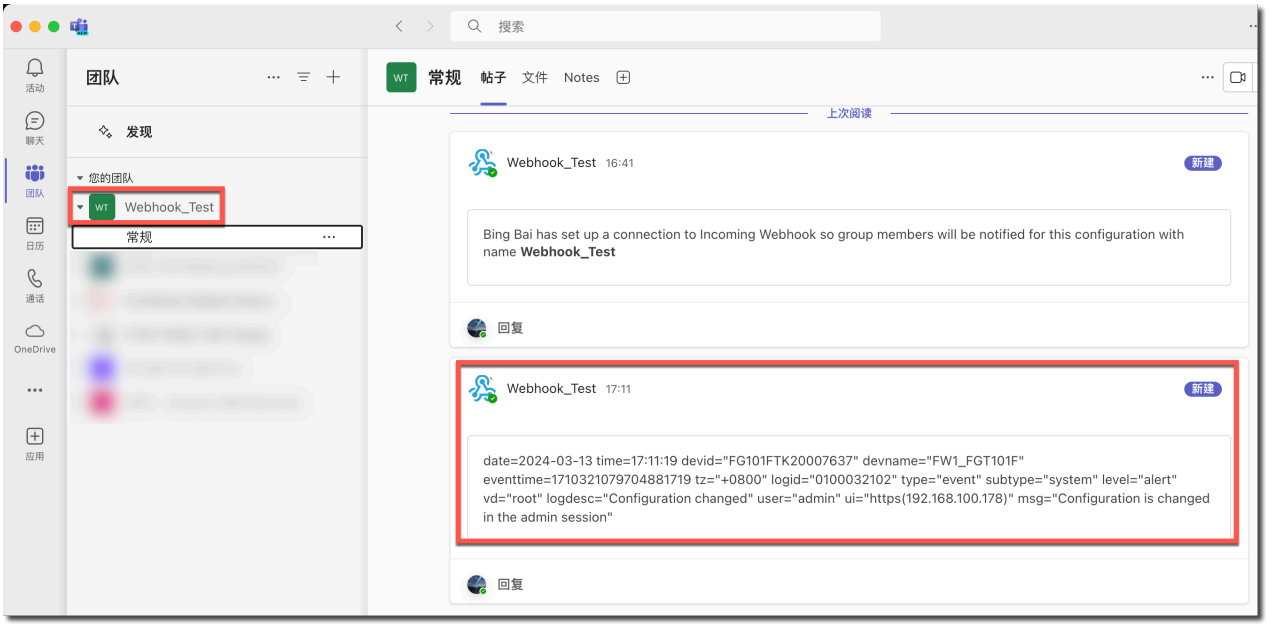



```bash
config system automation-stitch
    edit "Teams_webhook_cfg_change"
        set trigger "Config_change"
        config actions
            edit 1
                set action "Teams_Webhook"
                set required enable
            next
        end
    next
end
```

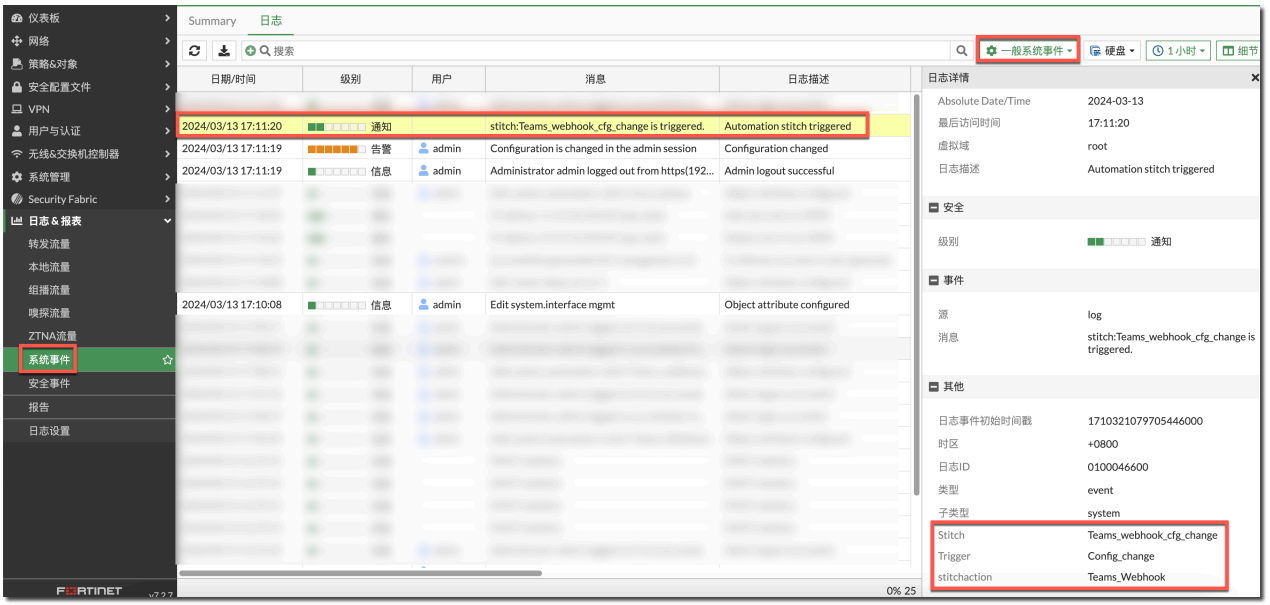
配置验证

1. 修改 FortiGate 的配置并退出登录管理员账户。

2. 在 Teams 的对应频道中可以看到 FortiGate 通过 Webhook 自动发送了配置变更的消息，消息内容为修改配置产生的系统事件日志。



3. 查看自动化触发的系统事件日志。




```bash
date=2024-03-13 time=17:11:20 eventtime=1710321079705445859 tz="+0800" logid="0100046600" type="event" subtype="system" level="notice" vd="root" logdesc="Automation stitch triggered" stitch="Teams_webhook_cfg_change" trigger="Config_change" stitchaction="Teams_Webhook" from="log" msg="stitch:Teams_webhook_cfg_change is triggered."
```

4. 查看自动化触发的 Debug 信息。

```bash
FortiGate # diagnose debug application autod -1
FortiGate # diagnose debug enable

auto_generate_generic_curl_request()-441: Generating generic automation CURL request for action (Teams_Webhook).
auto_generate_generic_curl_request()-493: Generic automation CURL request POST data for action (Teams_Webhook):
{ "text": "date=2024-03-13 time=17:11:19 devid=\"FG101FTK20007637\" devname=\"FW1_FGT101F\"
eventtime=1710321079704881719 tz=\"+0800\" logid=\"0100032102\" type=\"event\" subtype=\"system\" level=\"alert\"
vd=\"root\" logdesc=\"Configuration changed\" user=\"admin\" ui=\"https(192.168.100.178)\" msg=\"Configuration is
changed in the admin session\"" }

auto_generate_generic_curl_request()-548: Generic automation CURL request Host header: fortinet.webhook.office.com
auto_generate_generic_curl_request()-551: Adding 0 user defined headers
pid:283-__handle_msg()-414: Subscriber:4 received package. pubid:2 pkgid:65509 pkg_index:127
pid:283-__handle_pkg_logs()-356: Subscriber:4 processing package size:1215 logs:1 pickup:1
auto_event_pub_handle_log()-331: publish 46600 to subscr-node
```

```
pid:283-__subscr_close_cur_pkg()-140: close package size:1215 logs:1
auto_generic_curl_request_close()-632: Generic CURL request response body from
Teams_Webhook(https://fortinet.webhook.office.com/webhookb2/7e2f667a-91a7-4e1f-8153-25757d7d7e4a@2c36c478-3d00-452f-
8535-48396f5f01f0/IncomingWebhook/c20149904d60451fb6f167ba2cd6b129/ab9845cf-2876-49e7-b558-f3824db2f88a):
1
```

5. 查看该自动化工作流的状态统计信息。

```bash
FortiGate # diagnose test application autod 3
alert mail log count: 0

stitch: Teams_webhook_cfg_change

        local hit: 1 relayed to: 0 relayed from: 0
        last trigger:Wed Mar 13 17:11:19 2024
        last relay:
        actions:
                Teams_Webhook:
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Wed Mar 13 17:11:19 2024
                        last relay:

logid to stitch mapping:
id:32099  local hit: 0 relayed hits: 0
        Teams_webhook_cfg_change

id:32102  local hit: 1 relayed hits: 0
        Teams_webhook_cfg_change

log category to stitch mapping:
```

系统动作

简介

系统动作针对 FortiGate 包含备份配置（到 Flash）、重新启动和关机。

即使 FortiGate 处于内存保护模式，系统动作也能完成，自动化流程会自动绕过 CLI 命令确认提示（手动在 CLI 脚本执行这些操作时需要确认执行）。

CLI 配置

3 种系统动作的配置已经内置在 FortiGate 默认配置中。

```bash
config system automation-action
    edit "Backup Config Disk"
        set action-type system-actions
        set system-action backup-config
    next
    edit "Reboot FortiGate"
        set action-type system-actions
        set system-action reboot
    next
    edit "Shutdown FortiGate"
        set action-type system-actions
        set system-action shutdown
    next
end
```

网络需求

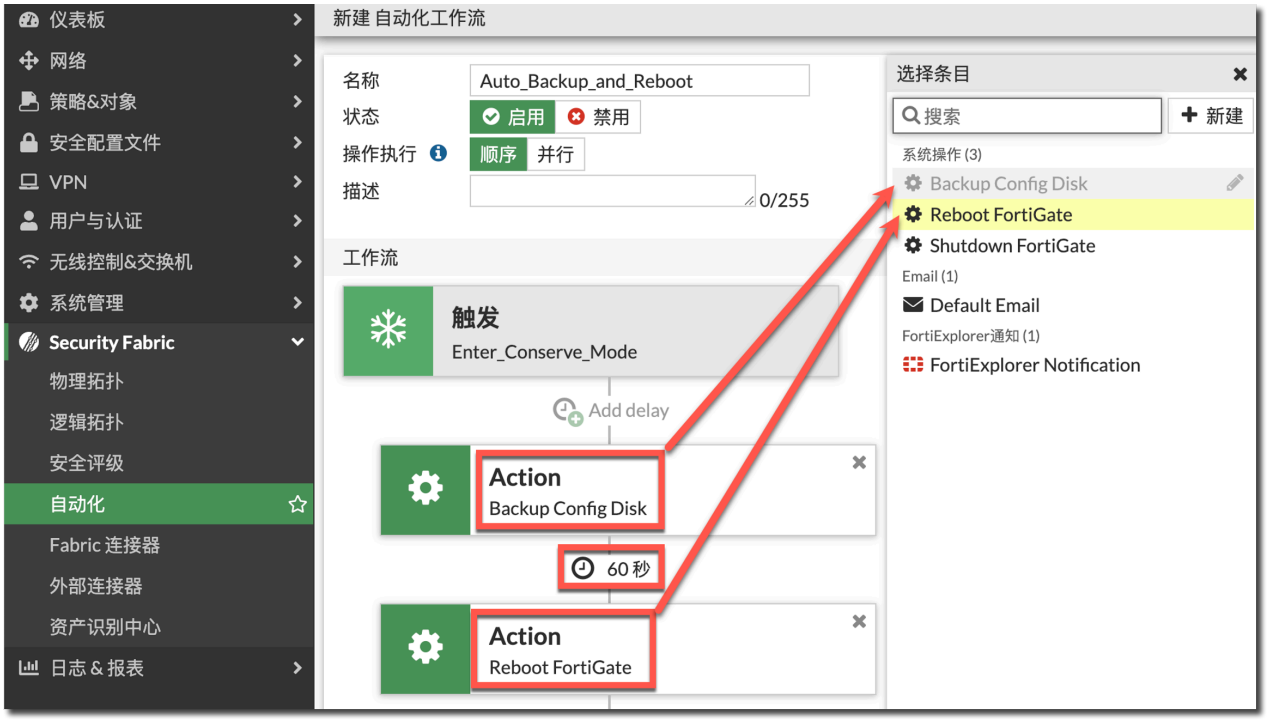
当 FortiGate 进入内存保护模式时，自动备份配置文件到 Flash，随后重启设备。

配置步骤

1. 配置设备进入保护模式后自动备份配置并重启。在 Security Fabric → 自动化中新建自动化工作流，触发条件新建并选择系统分类中的保护模式。



2. 添加第一个动作为预置的 Backup Config Disk，第二个动作为 Reboot FortiGate，中间添加 60s 的间隔，然后下发该工作流配置。



3. 相关 CLI 如下，配置了 Reboot 动作的最小间隔为 300s，防止频繁重启。

```bash
config system automation-trigger
    edit "Enter_Conserve_Mode"
        set event-type low-memory
    next
end

config system automation-action
    edit "Reboot FortiGate"
        set action-type system-actions
        set system-action reboot
        set minimum-interval 300
    next
    edit "Backup Config Disk"
        set action-type system-actions
        set system-action backup-config
    next
end

config system automation-stitch
    edit "Auto_Backup_and_Reboot"
        set trigger "Enter_Conserve_Mode"
        config actions
            edit 1
                set action "Backup Config Disk"
                set required enable
            next
            edit 2
                set action "Reboot FortiGate"
                set delay 60
                set required enable
            next
        end
    next
end
```

配置验证

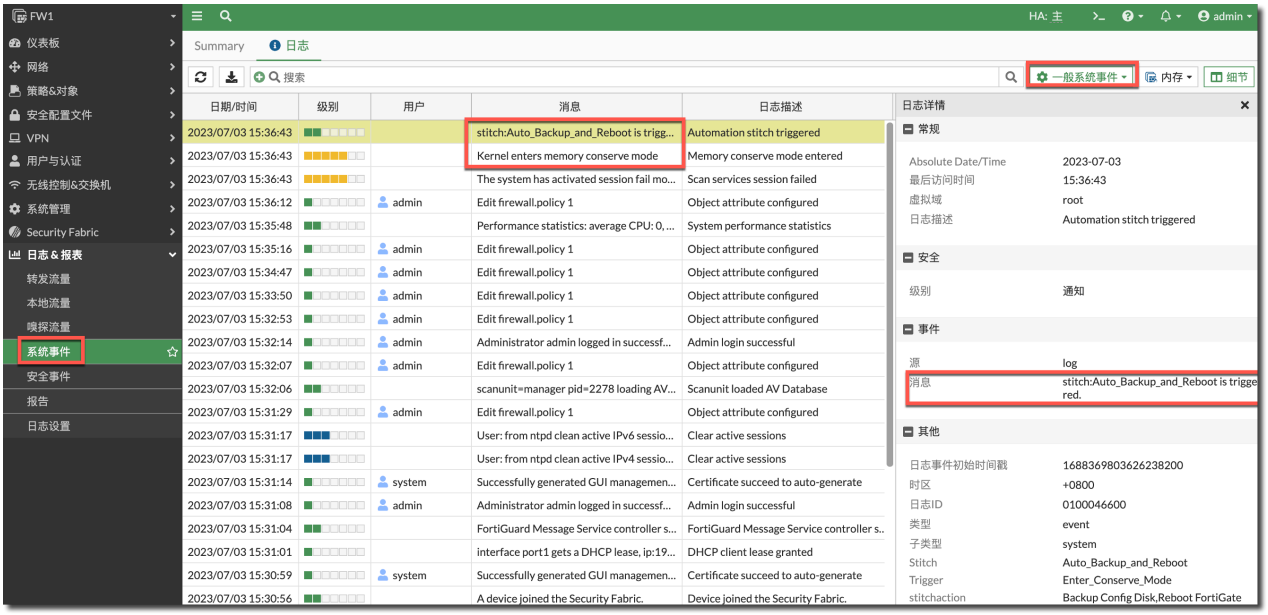
1. 触发 FortiGate 进入内存保护模式。

```bash
FortiGate # diagnose hardware sysinfo conserve
memory conserve mode:                          on
total RAM:                                   1465 MB
memory used:                                 1293 MB   88% of total RAM
memory freeable:                              150 MB   10% of total RAM
memory used + freeable threshold extreme:    1392 MB   95% of total RAM
memory used threshold red:                   1289 MB   88% of total RAM
memory used threshold green:                 1201 MB   82% of total RAM
```

2. 查看 autod 进程的 debug 信息，可以看到 FortiGate 进入保护模式后，触发了配置的 Trigger，自动执行配置备份到 Flash Disk，60s 后重启设备。

```bash
FortiGate # diagnose debug application autod -1
......
__dump_action_msg()-26: System will backup config, msg: Autod backup config by stitch: Auto_Backup_and_Reboot
......
__dump_action_msg()-26: System will reboot, msg: Autod reboot by stitch: Auto_Backup_and_Reboot
```

3. 查看自动化触发的系统日志。



4. 查看自动备份的配置文件。

```bash
FortiGate # execute revision list config
Last Firmware Version: V0.0.0-build000-REL0

ID  TIME                ADMIN           FIRMWARE VERSION          COMMENT
6   2023-07-03 15:36:44 daemon_admin    V7.2.5-build1517-REL0     Autod backup config by stitch:
Auto_Backup_and_Reboot
```

5. 查看该自动化流程的的配置和触发统计（使用 2 可以看统计和配置）。

```bash
FortiGate # diagnose test application autod 3
stitch: Auto_Backup_and_Reboot
        local hit: 3 relayed to: 0 relayed from: 0
        last trigger:Mon Jul  3 15:58:03 2023
        last relay:
        actions:
                Backup Config Disk:
                        done: 3 relayed to: 0 relayed from: 0
                        last trigger:Mon Jul  3 15:57:03 2023
                        last relay:
                Reboot FortiGate:
                        done: 3 relayed to: 0 relayed from: 0
                        last trigger:Mon Jul  3 15:57:03 2023
                        last relay:
```

6. 在 HA 环境下，自动化中备份配置的动作是不会同步的，只有触发了自动化的那台设备会备份配置到 Flash Disk。

```bash
FortiGate1 # execute revision list config
Last Firmware Version: V0.0.0-build000-REL0
ID  TIME                  ADMIN             FIRMWARE VERSION          COMMENT
6   2023-07-03 15:36:44   daemon_admin      V7.2.5-build1517-REL0     Autod backup config by stitch:
Auto_Backup_and_Reboot

FortiGate2 # execute revision list config
Last Firmware Version: V0.0.0-build000-REL0
```

自动抓包

网络需求

某客户遇到 HA 中的成员 FortiGate 由于 HA 心跳包丢失而成为双主状态，影响到正常业务。但出现该问题后一段时间会自动恢复，且出现的时间随机。

需要在出现此问题时在两台 FortiGate 的心跳接口抓取 HA 心跳报文，以进一步定位问题，但由于该问题出现的时间不固定，不好掌握抓包的时机。

通过自动化功能，可以实现在 FortiGate HA 心跳包丢失时，自动抓取心跳接口的报文，并发送至指定邮箱。

配置步骤

自动化触发器

配置自动化触发器为事件日志类型，事件日志类型为“Heartbeat packet lost”，logid 为 37910。

```bash
config system automation-trigger
    edit "HA_HB_Lost"
        set event-type event-log
        set logid 37910
    next
end
```

自动化动作

1. 配置第 1 个自动化动作，在 HA 心跳接口抓取 HA 心跳报文，该报文为二层报文，ether protocol 默认为 0x8890，抓取个数为 100 个。为了防止频繁触发抓包进程导致的内存占用问题，配置最小触发间隔为 120s。

```bash
config system automation-action
    edit "Sniffer_HA_HB_Start"
        set action-type cli-script
        set minimum-interval 120
        set script "diagnose sniffer packet port8 \'ether proto 0x8890\' 6 100 l"
        set accprofile "super_admin"
    next
end
```

2. 配置第 2 个自动化动作，将抓包结果发送到指定邮箱。

```bash
config system automation-action
    edit "Email_HA_HB_Lost_Sniffer_Result"
        set action-type email
        set email-to "bbai@fortinet.com"
        set email-subject "HA_HB_Lost_Sniffer_Result"
        set message "%%results%%"
    next
end
```

自动化工作流

配置自动化工作流，触发器引用已创建的“Heartbeat packet lost”类型的事件日志触发器，触发器触发后，动作为开始在 HA 心跳接口抓包，并发送抓包结果。

```bash
config system automation-stitch
    edit "HA_HB_Lost_Sniffer"
        set trigger "HA_HB_Lost"
        config actions
            edit 1
                set action "Sniffer_HA_HB_Start"
                set required enable
            next
            edit 2
                set action "Email_HA_HB_Lost_Sniffer_Result"
```

```
                set required enable
            next
        end
    next
end
```

- 触发器引用已创建的“Heartbeat packet lost”类型的事件日志触发器。
- 第 1 个动作引用开始抓取 HA 心跳报文的动作，抓取到 100 个包后会自动停止。
- 第 2 个动作引用发送邮件的动作，发送抓包结果。
- 第 3 个动作引用停止抓包的动作。

配置验证

1. 触发 HA 中两台 FortiGate 的心跳报文检测超时，并产生“Heartbeat packet lost”的 HA 系统事件日志。

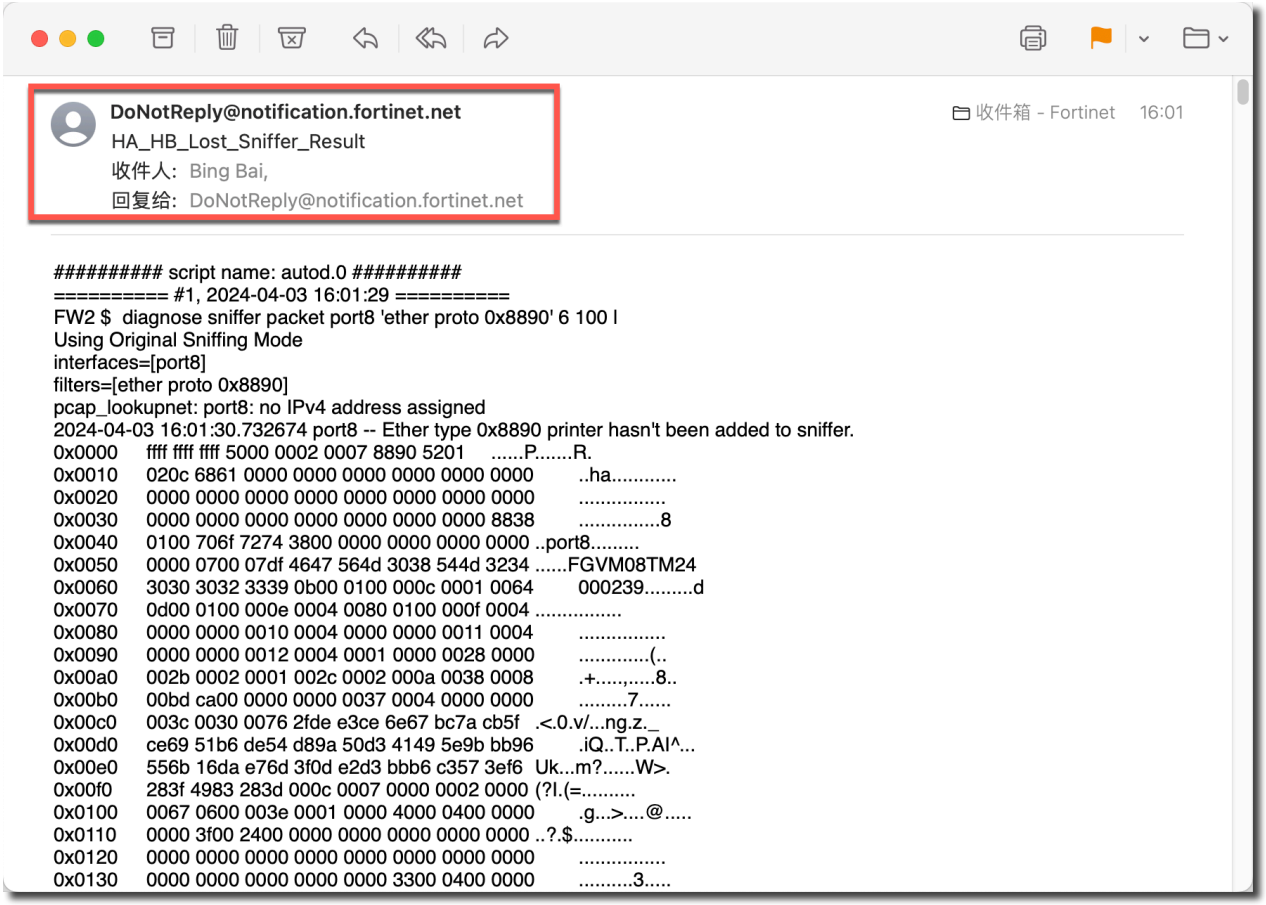
```
date=2024-04-02 time=17:02:22 eventtime=1712048543265466260 tz="+0800" logid="0108037910" type="event" subtype="ha"    bash
level="critical" vd="root" logdesc="Heartbeat packet lost" msg="Heartbeat packet lost" ha_role="primary"
devintfname="dmz"
```

2. 随后触发了自动化流程。

```
date=2024-04-02 time=17:02:22 eventtime=1712048543265705880 tz="+0800" logid="0100046600" type="event" subtype="system"bash
level="notice" vd="root" logdesc="Automation stitch triggered" stitch="HA_HB_Lost_Sniffer" trigger="HA_HB_Lost"
stitchaction="Sniffer_HA_HB_Start,Sniffer_HA_HB_Stop,Email_HA_HB_Lost_Sniffer_Result" from="log"
msg="stitch:HA_HB_Lost_Sniffer is triggered."
```

3. 配置的收件邮箱收到了包含抓包结果的邮件。

> **重要**
>
> 当 HA 心跳接口只有一对时，HA 发生心跳丢失会造成 HA 成员成为双主状态，此时 HA 的 2 个成员设备由于都触发了“Heartbeat packet lost”类型的事件日志，均会执行自动化流程，并发送抓包结果邮件到指定邮箱。



4. FortiGate 上的 autod 进程对应的自动化执行 Debug 信息。

```bash
FW1 # diagnose debug application autod -1
FW1 # diagnose debug enable

pid:2044-__handle_pkg_logs()-356: Subscriber:4 processing package size:459 logs:1 pickup:1
pid:2044-__subscr_close_cur_pkg()-140: close package size:459 logs:1
__action_cli_script_open()-186: cli script action:Sniffer_HA_HB_Start is called. svc ctx:0×1072dd90
accprof:super_admin script:
diagnose sniffer packet port8 'ether proto 0×8890' 6 100 l

......

__cli_script_close()-121: cli script:
autod.0
 output:
########## script name: autod.0 ##########

========== #1, 2024-04-03 16:01:29 ==========
FW1 $  diagnose sniffer packet port8 'ether proto 0×8890' 6 100 l
Using Original Sniffing Mode
interfaces=[port8]
filters=[ether proto 0×8890]
pcap_lookupnet: port8: no IPv4 address assigned
2024-04-03 16:01:30.743701 port8 -- Ether type 0×8890 printer hasn't been added to sniffer.
0×0000   ffff ffff ffff 5000 0001 0007 8890 5201        ......P.......R.

......

__action_email_hdl()-181: email action (Email_HA_HB_Lost_Sniffer_Result) is called.
from:
to:bbai@fortinet.com;
subject:HA_HB_Lost_Sniffer_Result
```

常用动作

工作流

常用触发器

- 系统事件日志触发器
- UTM 日志触发器
- 证书过期触发器
- 计划时间触发器

常用动作

- 内存/CPU 达到阈值时收集信息
- Email 动作
- Teams Webhook
- 系统动作
- 自动抓包

常用动作

日志

- 日志存储方式
- 日志类型
- 删除日志
- 导出日志
- 自定义日志过滤器
- CLI 审计日志
- 日志显示地区/主机名
- 硬盘与 Flash 闪存
- 邮件告警
- 日志常见问题

日志存储方式

防火墙上的日志存储主要有 4 种方式：**1. 硬盘；2. 内存；3.FortiAnalyzer；4. 第三方 syslog 服务器**

硬盘记录日志

1.格式化硬盘

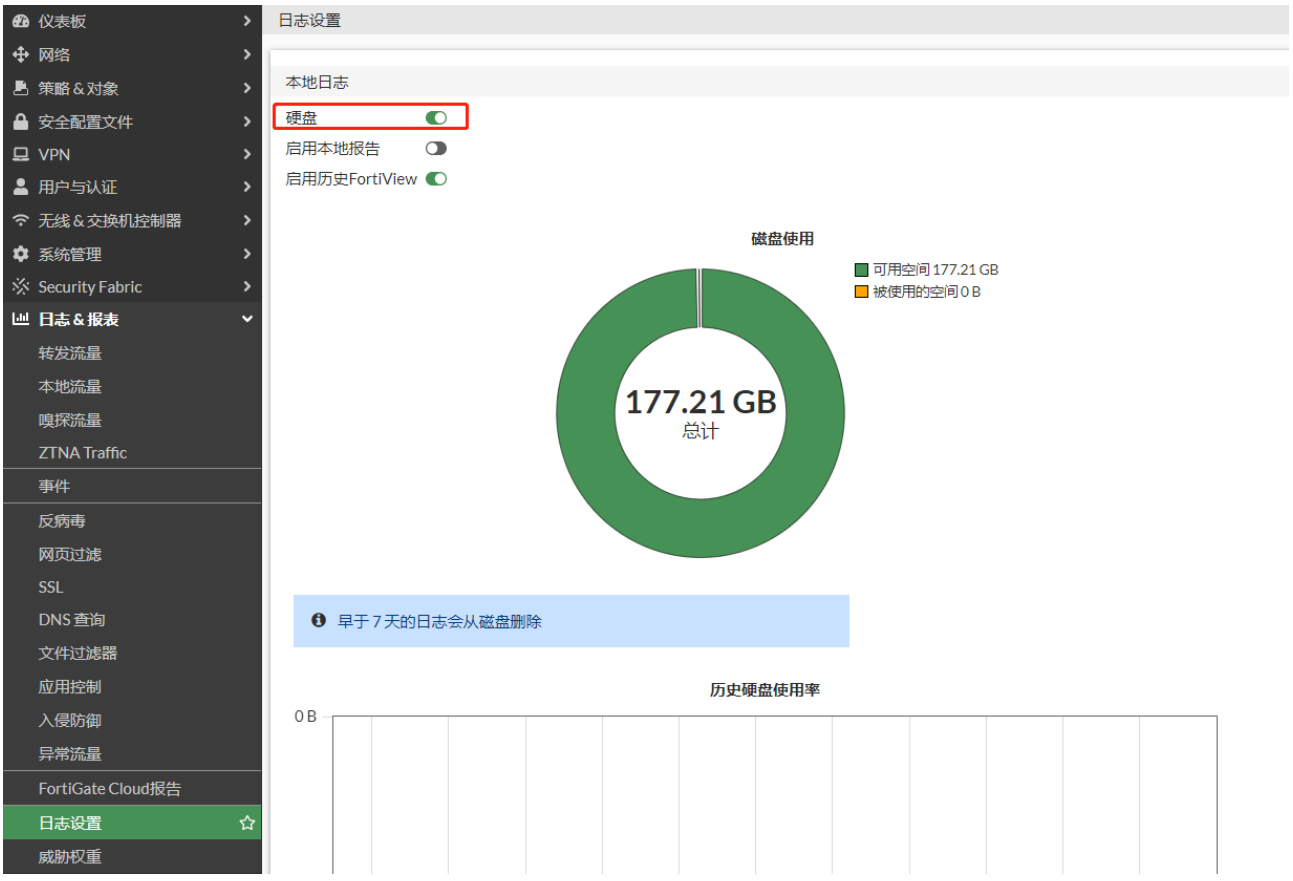
在设备部署前，请先格式化硬盘，以免后续使用硬盘记录日志时产生异常情况，格式化硬盘会重启设备，硬盘中的数据将清空。设备的版本，配置文件是保存在 flash 中的，不会被影响。

不是所有的 FortiGate 设备都有硬盘，如 FortiGate 500E，FortiGate600F 不带硬盘，FortiGate 501E，FortiGate601F 带硬盘，没有硬盘则没有 execute formatlogdisk 命令。

```
FGT # execute formatlogdisk
Log disk is /dev/sda1.
Formatting this storage will erase all data on it, including
  logs, quarantine files;
and require the unit to reboot.
Do you want to continue? (y/n)
```

2.开启硬盘日志

选择“日志设备”，在本地日志，在”本地日志“中打开”硬盘“选项即可。



2.不建议使用硬盘记录流量日志

由于流量日志非常大，所以记录于硬盘将带来设备性能损耗，减少硬盘寿命，以及从大量的流量日志中进行查询也非常消耗 CPU 资源。建议将流量日志发送到 FortiAnalyzer 或者 syslog 服务器。

过滤硬盘记录流量日志的配置如下：

```
config log disk filter
    config free-style
        edit 1
            set category traffic
```

```
            set filter "type traffic"
            set filter-type exclude
        next
    end
end
```

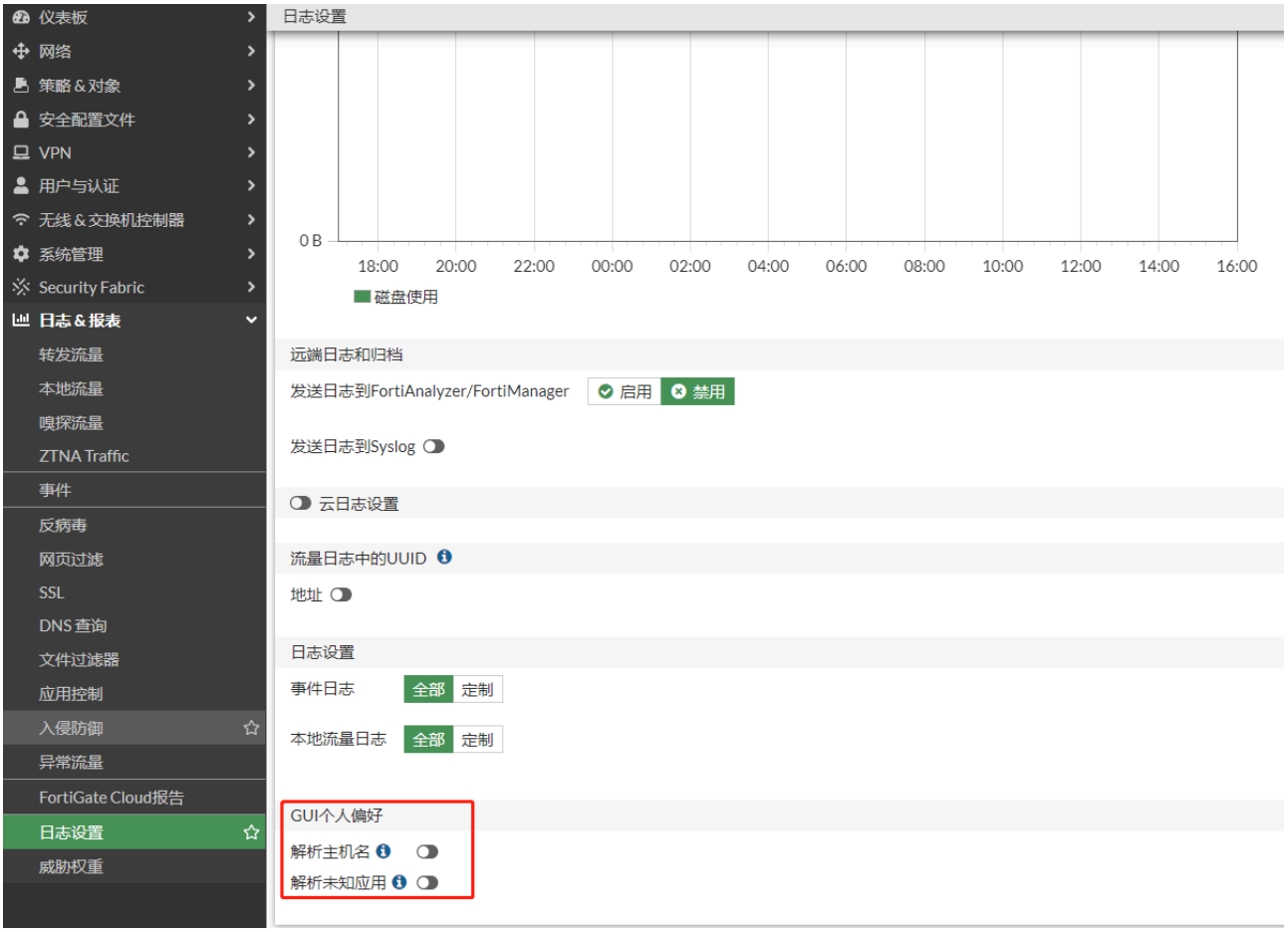
3.设置硬盘记录日志的参数

默认硬盘日志保存的时间为 7 天，可以根据需求进行修改。当硬盘满时，会覆盖最早的日志。

```
config log disk setting
    set maximum-log-age 7
    set diskfull overwrite
end
```

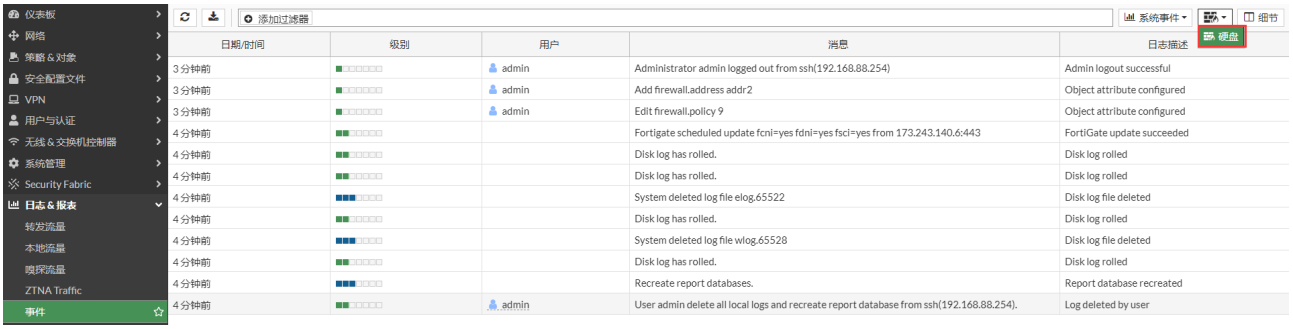
建议取消勾选“解析主机名”和“解析未知应用”两个选项，影响日志观察。

解析主机名：使用反向 DNS 解析；**解析未知应用**：使用 Internet 服务数据库解析

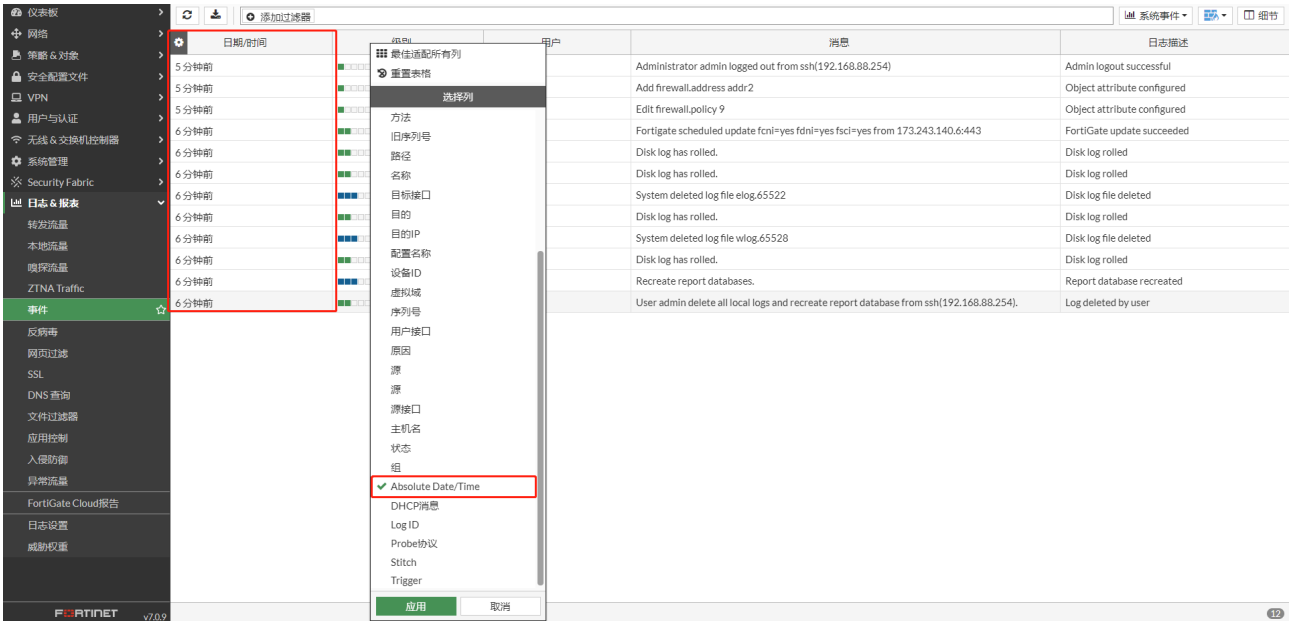


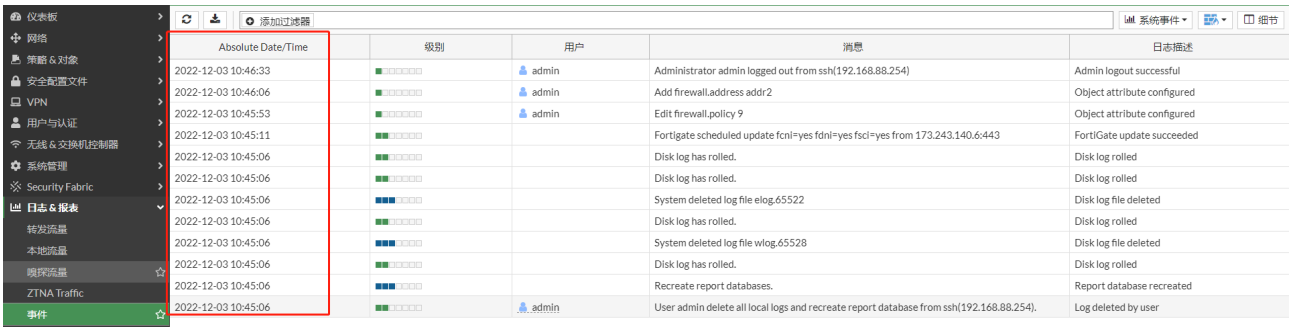
5.查看硬盘日志

在 GUI 界面右上角选择“硬盘”，就可以查看到硬盘的对应的日志，也可以通过“添加过滤器”来过滤日志。



上图日志的第一栏显示了日志产生的相对时间，但排错往往需要查看日志的绝对时间。通过右击日志的标题栏，选择“Absolute Date/Time”，去掉原有的“时间/日期”列，就可以看到日志的绝对时间了，而且列是可以拖动的。





内存记录日志

重要

在 FortiGate 中不建议使用内存记录日志，建议将日志发送到 FortiAnalyzer 或者 syslog 服务器。通常是为了排除任何特定问题或在本地监视日志而临时开启，稍后再禁用它。重启后内存日志会丢失。

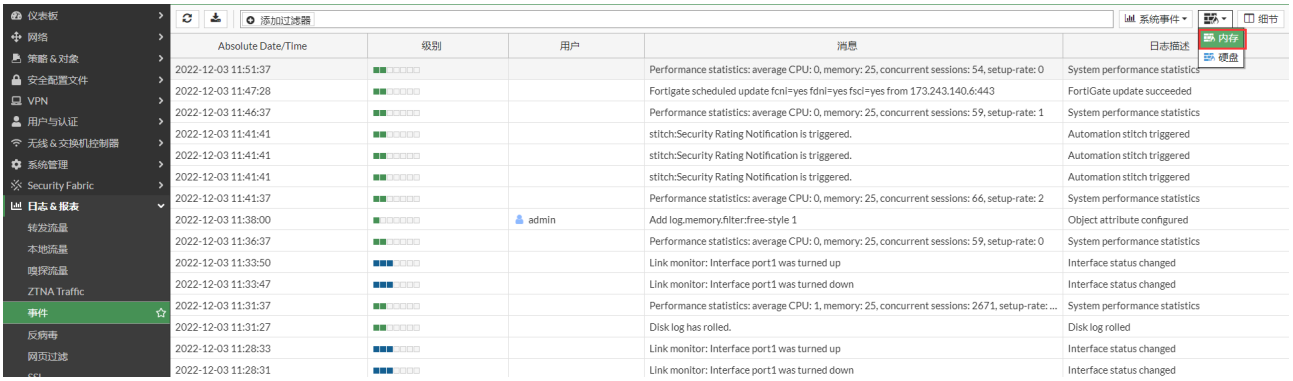
1.开启内存记录日志

通过在 config log memory setting 中开启内存记录日志。

```
config log memory setting
    set status enable
end
```

2.查看内存日志

在 GUI 界面右上角选择“内存”，查看内存日志。

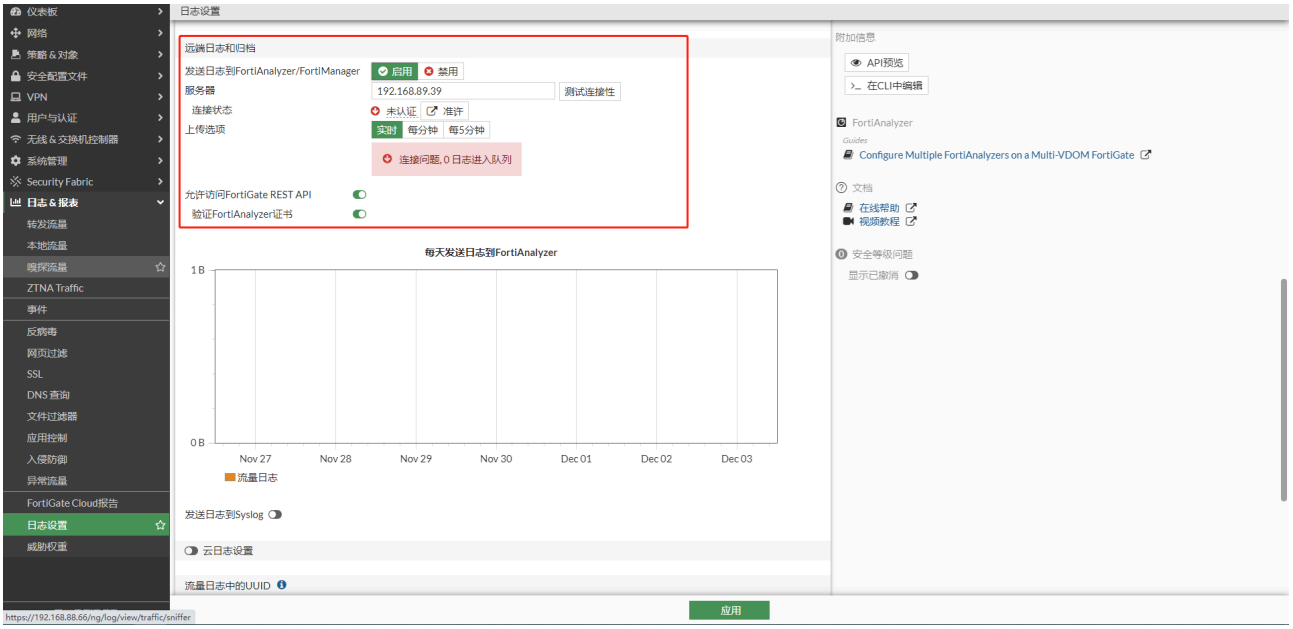


FortiAnalyzer 记录日志

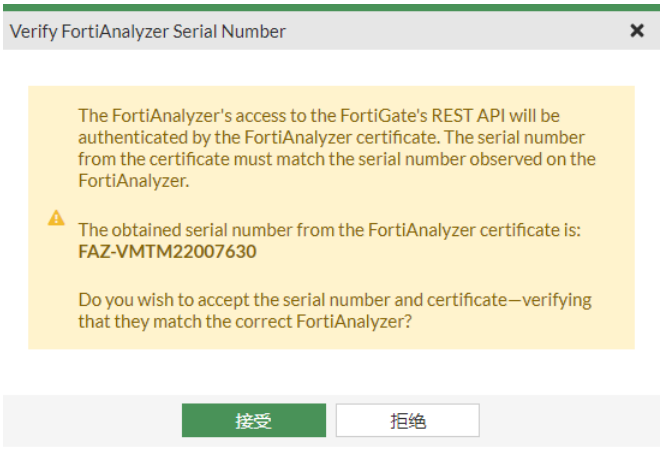
FortiAnalyzer 是专门用来做为 Fortinet 设备的网路安全日志记录、分析、统计和产生报表的产品。FortiGate 和 FortiAnalyzer 之间使用 OFTP 协议交互，端口是 TCP 514。FortiAnalyzer 的固件版本应该等于或高于运行在 FortiGate 上的版本，如 FortiAnalyzer 的版本如果是 7.0，FortiGate 的版本可以是 7.0 或者 6.4。

1.开启 FortiAnalyzer 记录日志

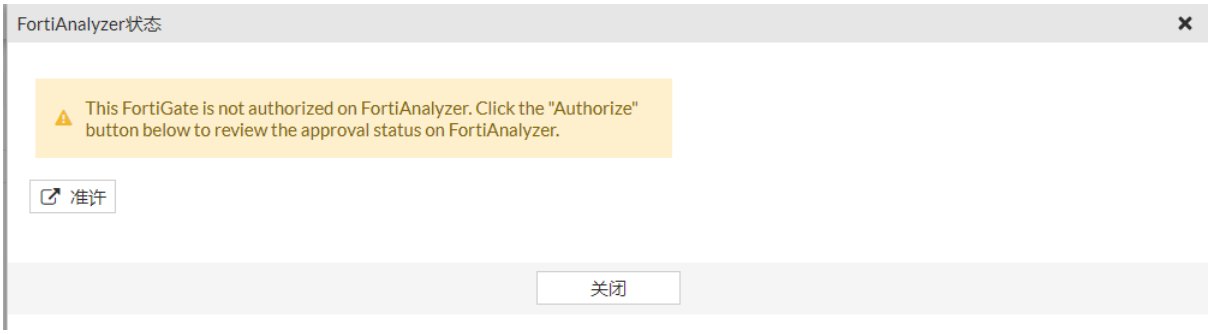
在“日志设置”中，启用”发送日志到 FortiAnalyzer/FortiManager“，“服务器”填写 FortiAnalyzer 的 IP 地址，”上传选项“选择“实时”，然后点击“应用”。



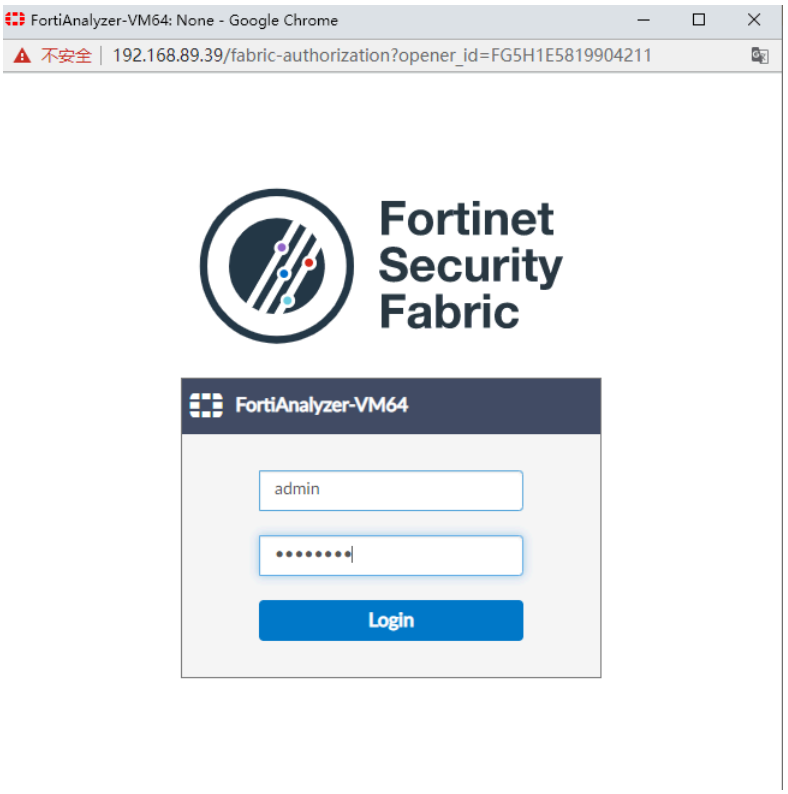
弹出” Verify FortiAnalyzer Serial Number“，点击“接受”。



点击“准许”。



弹出 FortiAnalyzer 登录窗口，输入 FortiAnalyzer 账号密码登录。



登录成功后，点击“Approve”准许 FortiGate 注册到 FortiAnalyzer，并点击“OK”。

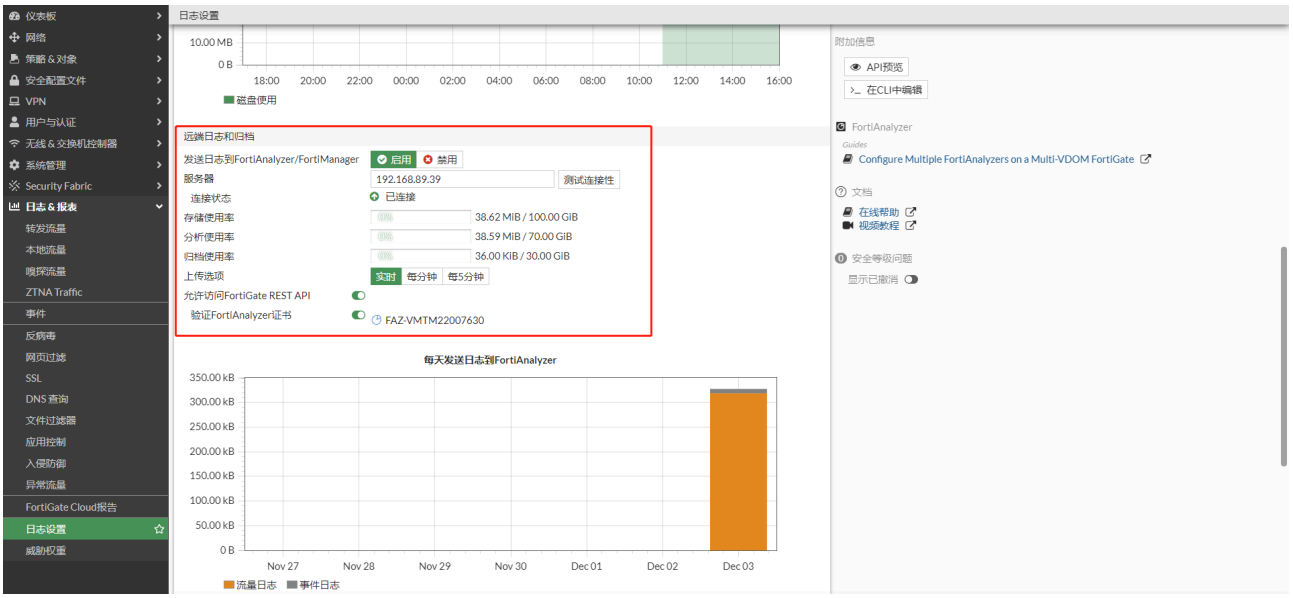


FortiGate 已成功被 FortiAnalyzer 授权成功。



2.查看连接状态

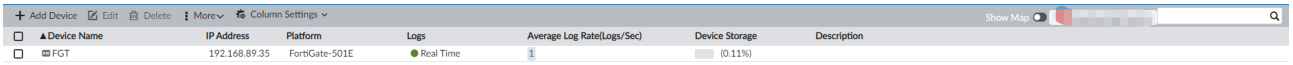
分别从 FortiGate 和 FortiAnalyzer 上查看两者的状态。FortiGate 连接状态如下，可以通过“测试连接性”进行测试。



FortiGate CLI 通过 execute log fortianalyzer test-connectivity 测试连接状态。

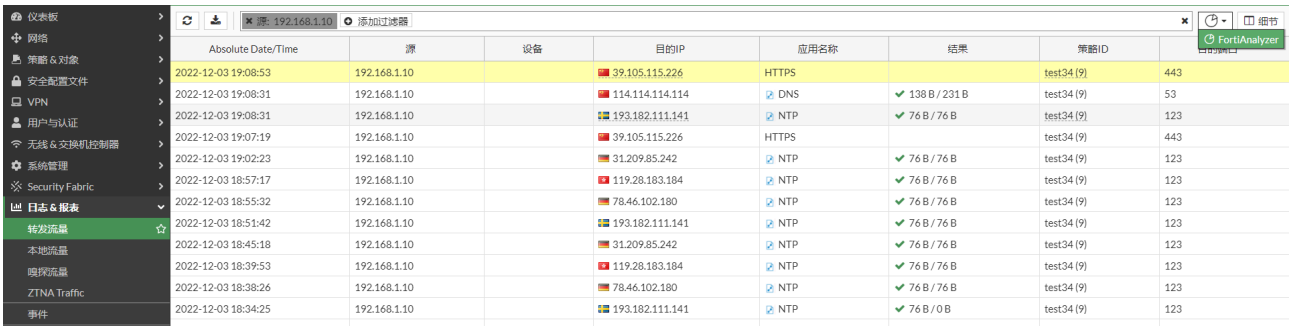
```
# execute log fortianalyzer test-connectivity
FortiAnalyzer Host Name: FAZVM64
FortiAnalyzer Adom Name: root
FortiGate Device ID: FG5H1E5819904211
Registration: registered
Connection: allow
Adom Disk Space (Used/Allocated): 41997274B/107374182400B
Analytics Usage (Used/Allocated): 41960410B/75161927680B
Analytics Usage (Data Policy Days Actual/Configured): 60/60 Days
Archive Usage (Used/Allocated): 36864B/32212254720B
Archive Usage (Data Policy Days Actual/Configured): 0/365 Days
Log: Tx & Rx (33 logs received since 16:19:54 12/03/22)
IPS Packet Log: Tx & Rx
Content Archive: Tx & Rx
Quarantine: Tx & Rx

Certificate of Fortianalyzer valid and serial number is:FAZ-VMTM22007630
```

FortiAnalyzer 查看该 FortiGate 连接如下：



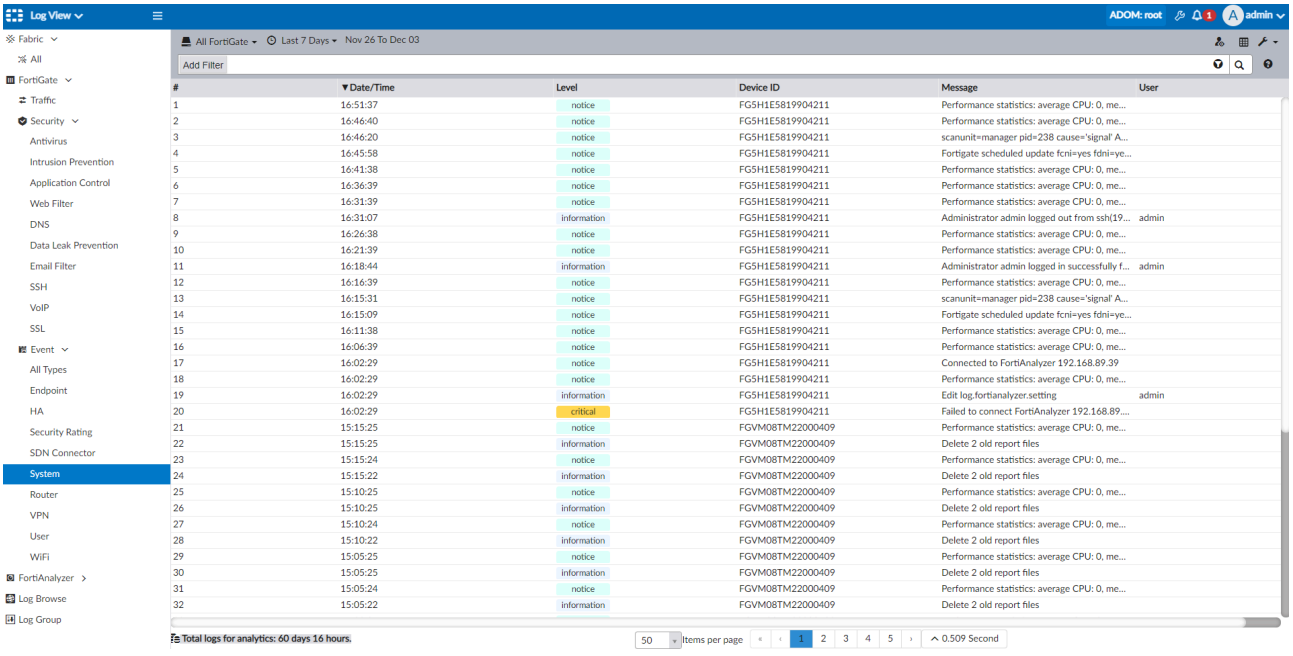
3.在 FortiGate 上查看 FortiAnalyzer 的日志

在 GUI 界面右上角选择“FortiAnalyzer”，查看 FortiAnalyzer 记录的此 FortiGate 的日志。



4.在 FortiAnalyzer 上查看日志

登录 FortiAnalyzer，在 Log View 界面可以查看到 FortiGate 产生的系统日志，流量日志，UTM 日志等。

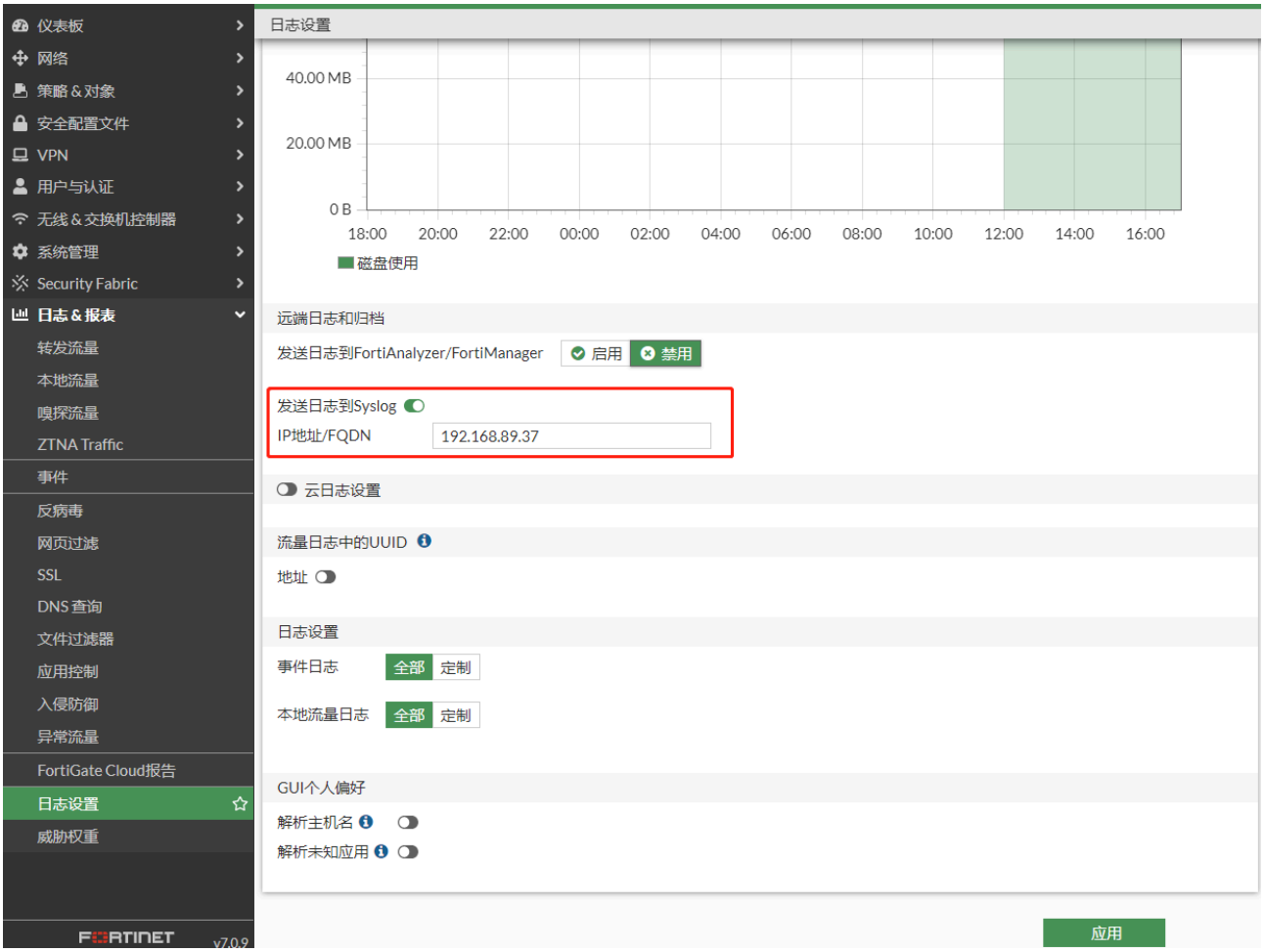


Syslog 记录日志

Syslog 也称为系统日志协议，是一种标准协议，用于将事件消息或系统日志发送到特定服务器，即 Syslog 服务器。 Syslog 主要用于从各种机器收集不同的设备日志，并将它们存储在一个中央位置以进行监控和查看。Syslog 协议使用 UDP 514 端口。

1.开启 syslog 记录日志

开启“发送日志到 Syslog"，IP 地址是 syslog 服务器的 IP 地址。



对应的命令行如下：

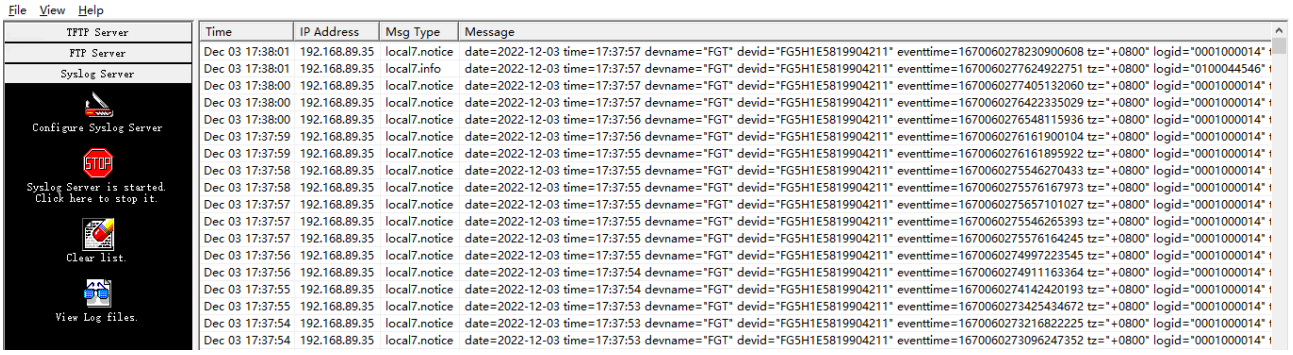
```
config log syslogd setting
    set status enable
    set server "192.168.89.37"
    set port 514           //syslog默认使用UDP 514端口传输数据，可以根据需求修改
end
```

如果有多个 syslog 服务器，可以通过命令行设置多个。

```
config log syslogd (敲? 号)
syslogd     Configure first syslog device.
syslogd2    Configure second syslog device.
syslogd3    Configure third syslog device.
syslogd4    Configure fourth syslog device.

如设置第2个日志服务器:
config log syslogd2 setting
    set status enable
    set server "192.168.89.38"
end
```

2.在 syslog 服务器上查看日志

如下使用简易的 syslog 测试软件 3CD 查看收到的日志。



日志类型

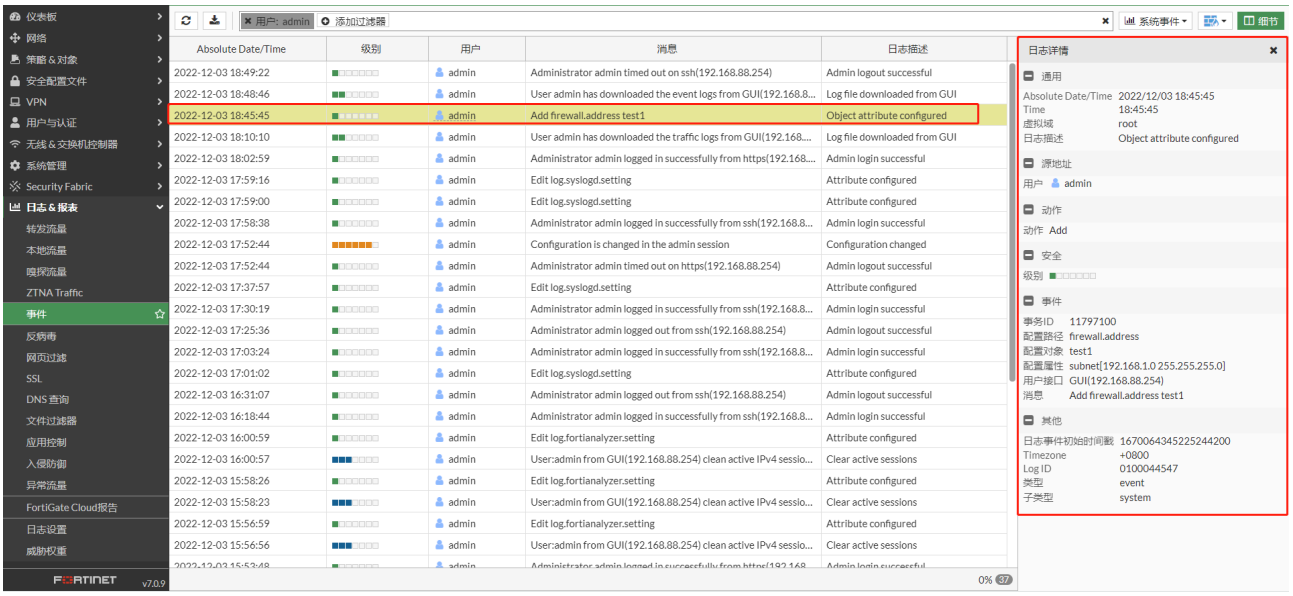
每个日志条目都包含一个 Type (类型) 字段，该字段表示其日志类型。主要有三种类型：Event（事件日志），Traffic（流量日志），UTM（安全日志）。

事件日志

事件日志用于记录系统和管理事件，如用户的操作日志，进程的活动日志等。根据事件日志 目的原因 的不同又分为很多 Subtype（子类），如：System，Router，VPN，SD-WAN，User，Endpoint control，High Availability 等，可以在 GUI 界面根据子菜单查看对应的事件。



如管理员 admin 的添加地址对象的操作日志



该日志对应的文本

```
date=2022-12-03 time=18:45:45 eventtime=1670064345225244185 tz="+0800" logid="0100044547" type="event" subtype="system"
level="information" vd="root" logdesc="Object attribute configured" user="admin" ui="GUI(192.168.88.254)" action="Add"
cfgtid=11797100 cfgpath="firewall.address" cfgobj="test1" cfgattr="subnet[192.168.1.0 255.255.255.0]" msg="Add
firewall.address test1"
```

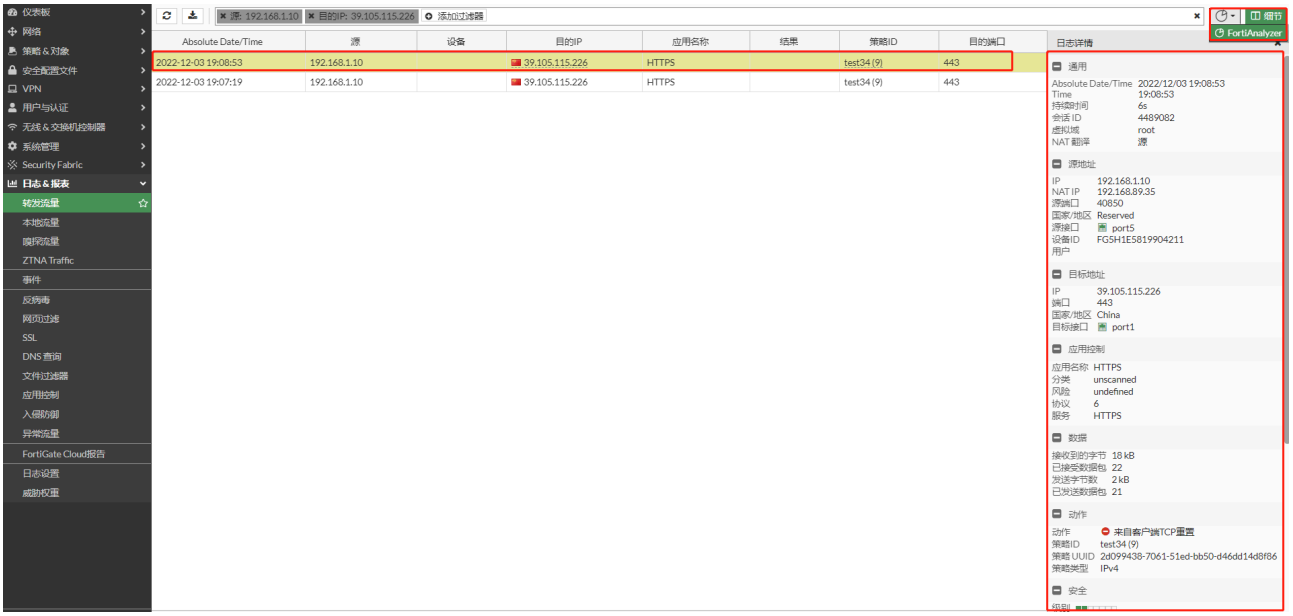
流量日志

流量日志用于记录流量信息，如终端访问 HTTPS 站点。根据流量日志 目的原因 的不同又分为很多 Subtype（子类），如：Forward，Local，Sniffer 等。建议将流量日志发送到 FortiAnalyzer 或者 syslog 服务器。

在策略中，”记录允许流量“选择”全部会话“，这样 FortiGate 才会记录流量日志。“全部会话”包含了“安全事件”。



流量日志可以在 GUI 界面通过对应的菜单查看，这里 FortiGate GUI 的”转发流量“通过 FortiAnalyzer 读取。



该日志对应的文本

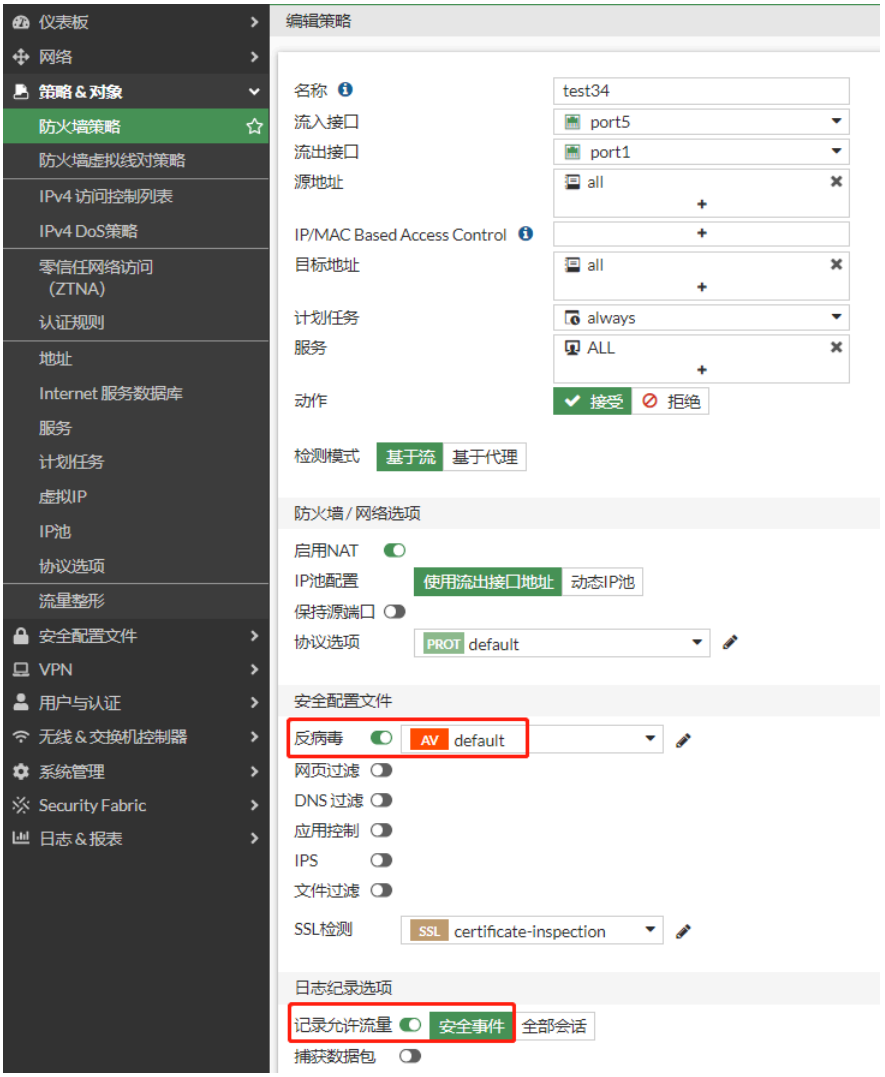
```
date=2022-12-03 time=19:08:53 id=7172877726888493056 itime="2022-12-03 19:08:58" euid=3 epid=1030 dsteuid=3 dstepid=101
logflag=1 logver=700090444 type="traffic" subtype="forward" level="notice" action="client-rst" policyid=9
sessionid=4489082 srcip=192.168.1.10 dstip=39.105.115.226 transip=192.168.89.35 srcport=40850 dstport=443 transport=40850
trandisp="snat" duration=6 proto=6 sentbyte=2101 rcvdbyte=18304 sentpkt=21 rcvdpkt=22 logid=0000000013 service="HTTPS"
app="HTTPS" appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
```

```
eventtime=1670065734672429913 poluuid="2d099438-7061-51ed-bb50-d46dd14d8f86" srccountry="Reserved" dstcountry="China"
srcintf="port5" dstintf="port1" policyname="test34" tz="+0800" devid="FG5H1E5819904211" vd="root" dtime="2022-12-03
19:08:53" itime_t=1670065738 devname="FGT"
```

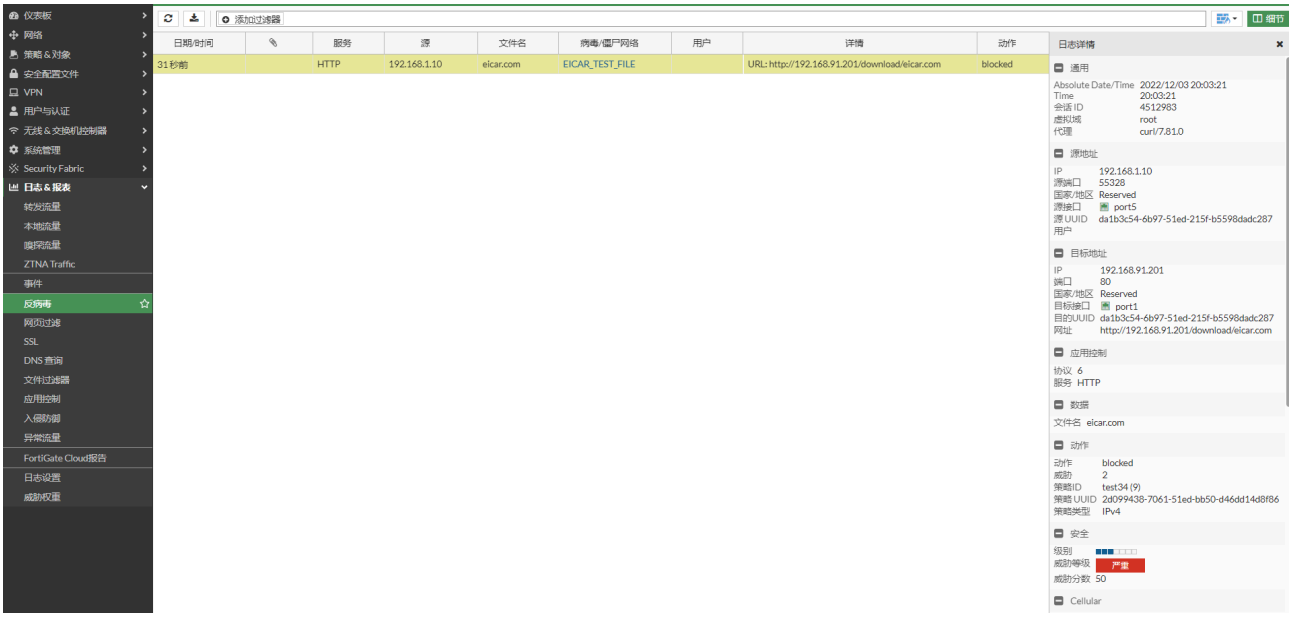
UTM 日志

UTM 日志用于记录 UTM 事件，如 IPS 拦截的流量。根据 UTM 日志 目的原因 的不同又分为很多 Subtype（子类），如：Virus，Web Filter，IPS，App-CTRL 等。

在策略中，根据需求开启所需的 UTM 功能，这里开启 AV 功能。UTM 日志默认是开启的，即“记录允许流量”中的“安全事件”。



反病毒日志可以在 GUI 界面通过对应的菜单查看。



该日志对应的文本

```
date=2022-12-03 time=20:03:21 eventtime=1670069001604241098 tz="+0800" logid="0211008192" type="utm" subtype="virus"
eventtype="infected" level="warning" vd="root" policyid=9 poluuid="2d099438-7061-51ed-bb50-d46dd14d8f86"
policytype="policy" msg="File is infected." action="blocked" service="HTTP" sessionid=4512983 srcip=192.168.1.10
dstip=192.168.91.201 srcport=55328 dstport=80 srccountry="Reserved" dstcountry="Reserved" srcintf="port5"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" srcuuid="da1b3c54-6b97-51ed-215f-b5598dadc287"
dstuuid="da1b3c54-6b97-51ed-215f-b5598dadc287" proto=6 direction="incoming" filename="eicar.com" quarskip="Quarantine-
disabled" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 url="http://192.168.91.201/download/eicar.com" profile="default" agent="curl/7.81.0"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f" analyticssubmit="false" crscore=50
craction=2 crlevel="critical"
```

删除日志

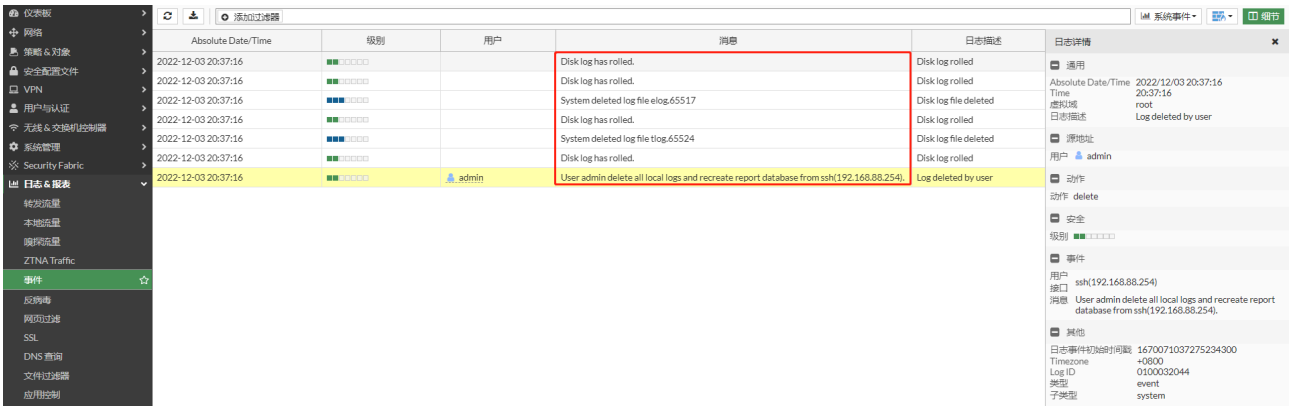
FortiGate 可以对记录在本地的日志（硬盘和内存）进行删除。

1.删除所有的本地日志

执行 execute log delete-all 删除所有日志

```
# execute log delete-all
This will delete all local logs and recreate report database.
Do you want to continue? (y/n)y
```

删除日志后，删除日志动作本身也会产生日志



2.指定删除某个类型的日志

选择日志的存储类型

```
# execute log filter device
Available devices:
 0: memory
 1: disk
 2: fortianalyzer
 3: fortianalyzer-cloud
 4: forticloud
```

选择日志的类型

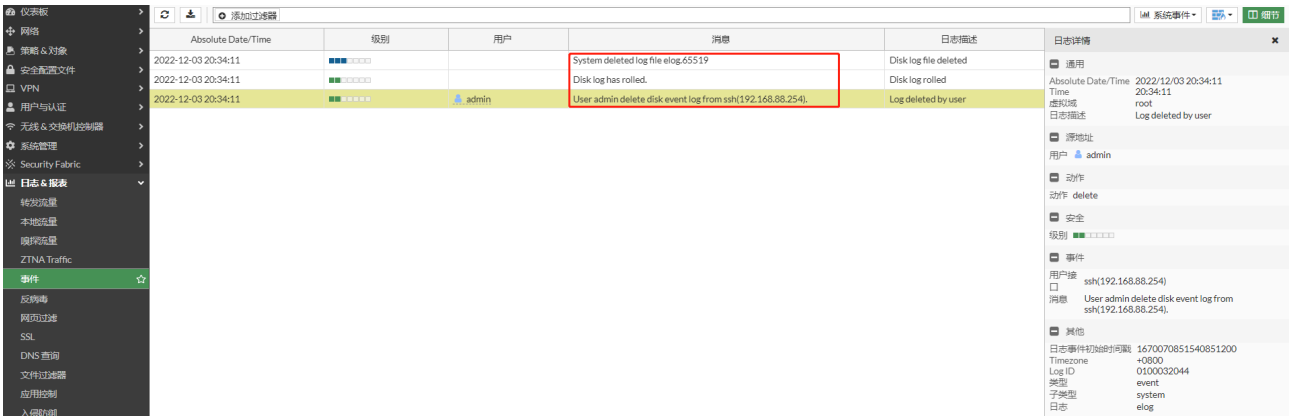
```
# execute log  filter category
Available categories:
 0: traffic
 1: event
 2: utm-virus
 3: utm-webfilter
 4: utm-ips
 5: utm-emailfilter
 7: utm-anomaly
 8: utm-voip
 9: utm-dlp
10: utm-app-ctrl
12: utm-waf
15: utm-dns
16: utm-ssh
17: utm-ssl
19: utm-file-filter
20: utm-icap
22: utm-sctp-filter
```

删除日志

这里删除硬盘的日志事件日志

```
# execute log filter device 1
# execute log filter category 1
# execute log delete
This will delete disk event log
Do you want to continue? (y/n)y
```

删除日志后，删除动作本身也会产生日志

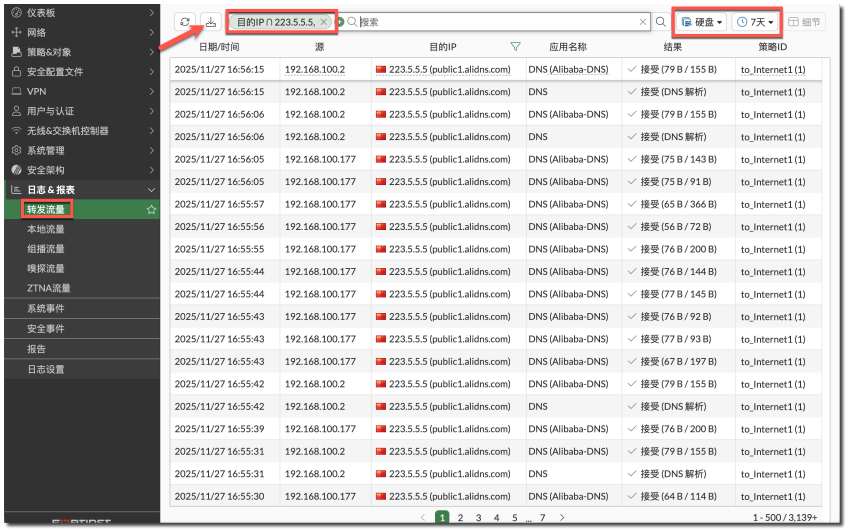


导出日志

通过 GUI 导出

> **提示**
>
> 通过 GUI 单次最多可以下载最近的 400,000 条日志条目。如日志条目数超过 400,000，需导出全部日志，请使用 FTP/TFTP/USB 方式导出。

1. 在“日志 & 报表”中进入想要下载的日志类型页面，这里以转发流量为例。
2. 过滤想要下载的特定日志（这里以目标 IP 223.5.5.5 为例），在右上角选择存储日志的位置（这里以硬盘为例），选择日志时间范围（这里以最近 7 天为例）。



3. 点击左上角的下载按钮，即可下载明文日志文件，后缀为 `.log`，如 `disk-traffic-forward-2025-11-27_17-00.log`。

通过 FTP/TFTP/USB 导出

1. 通过如下 CLI 命令可以以 LZ4 压缩格式导出当前硬盘/内存的全部/单类日志到 FTP/TFTP/USB。

```
execute backup [disk | memory] alllogs [ftp | tftp | usb] <ftp server>[:ftp port] <user> <passwd> uncompressed
execute backup [disk | memory] log [ftp | tftp | usb] <ftp server>[:ftp port] <user> <passwd> <log_type> uncompressed
```

   - `alllogs`：备份所有以下类型日志。
   - `log`：可在 `<log_type>` 字段定义要备份的一种日志类型，包含 `traffic`、`event`、`virus`、`webfilter`、`ips`、`emailfilter`、`anomaly`、`voip`、`dlp`、`app-ctrl`、`waf`、`dns`、`ssh`、`ssl`、`file-filter`、`icap`、`sctp-filter`、`forti-switch`、`virtual-patch`、`casb`、`debug`。
   - `uncompressed`：表示以未压缩方式备份日志。

   > **提示**
   >
   > 如果日志备份命令不携带 `uncompressed` 参数，或使用 `compressed`，则会以 LZ4 压缩格式备份日志，需要使用解压工具进行解压才能看到原始日志（详见 https://community.fortinet.com/t5/FortiAnalyzer/Technical-Tip-Transferring-historical-logs-from-a-FortiGate-hard/ta-p/193850 ↗）。

2. 在导出日志前，使用如下 CLI 可以查看预估导出特定分类日志的大小（包含压缩大小和未压缩大小），并根据日志文件大小规划 FTP/TFTP 服务器或 USB 的剩余空间。如下所示，查看系统日志（`1: event`）的日志文件大小（未压缩大小为 4 MB，压缩大小为 400 KB）。

```bash
execute log list <category>
Available categories:
 0: traffic
 1: event
 2: utm-virus
 3: utm-webfilter
 4: utm-ips
```

```
 5: utm-emailfilter
 7: utm-anomaly
 8: utm-voip
 9: utm-dlp
10: utm-app-ctrl
12: utm-waf
15: utm-dns
16: utm-ssh
17: utm-ssl
19: utm-file-filter
20: utm-icap
22: utm-sctp-filter
23: forti-switch
24: utm-virtual-patch
25: utm-casb
26: debug

FortiGate # execute log list 1
elog.65447         106538        18156   Thu Nov 20 00:00:00 2025
elog.65446         108001        18558   Fri Nov 21 00:00:00 2025
elog.65445         105451        17813   Sat Nov 22 00:00:00 2025
elog.65444         113756        20098   Sun Nov 23 00:00:00 2025
elog.65443         106007        17791   Mon Nov 24 00:00:00 2025
elog.65442         105157        17607   Tue Nov 25 00:00:00 2025
elog.65441         162282        32429   Wed Nov 26 00:00:00 2025
elog.65440         132276        25886   Thu Nov 27 00:00:00 2025
elog.65439        3286179       241574   Thu Nov 27 18:03:17 2025
9 elog file(s) found.
Uncompressed Total:  4 MB       Compressed Total:  400 KB
```

3. 这里以备份硬盘中的系统日志到 FTP 服务器为例，FortiGate 会将所有系统日志分段（如需）发送到 FTP 服务器（未压缩日志体积大于压缩文件，系统会给出提示信息，按 `y` 继续导出）。

> **提示**
>
> 并非所有型号包含硬盘，可以通过 `execute disk list` 命令查看。如设备不包含硬盘，可以选择 `memory` 或 `usb` 方式。

```
FortiGate # execute backup disk log ftp 192.168.90.253 user1 password event uncompressed
```
bash

```
WARNING: The size of uncompressed log files can be very large.
Please check uncompressed log size for each log category
before uploading to FTP server by command "execute log list [logcategory].
And uploading uncompressed files to FTP server could take hours.
Do you want to continue? (y/n)y

Upload uncompressed log to FTP server!

Connect to ftp server 192.168.90.253 ...
Please wait...
uploading 24255B/241001B(10%)
uploading 50519B/241001B(20%)
uploading 75236B/241001B(31%)
uploading 102429B/241001B(42%)
uploading 125595B/241001B(52%)
uploading 150578B/241001B(62%)
uploading 174505B/241001B(72%)
uploading 197624B/241001B(82%)
uploading 224653B/241001B(93%)
uploading 241001B/241001B(100%)
uploaded file size:241001B
Sent log file elog.65439 to ftp server as disk-event_FG101FTK20007637_root_20251127_175317_65439 OK.

Connect to ftp server 192.168.90.253 ...
Please wait...
uploading 4081B/25886B(15%)
uploading 6522B/25886B(25%)
uploading 12769B/25886B(49%)
uploading 15420B/25886B(59%)
uploading 20131B/25886B(77%)
```

```
uploading 23349B/25886B(90%)
uploading 25886B/25886B(100%)
uploaded file size:25886B
Sent log file elog.65440 to ftp server as disk-event_FG101FTK20007637_root_20251127_000000_65440 OK.
......
```

4. 导出的日志可以直接使用文本工具打开查看。

自定义日志过滤器

功能需求

1. 默认情况下，在 log filter 中配置日志过滤器时（FortiOS 7.0 及以上版本），只能指定某个分类的日志的开启和关闭；对于记录的日志级别，只能指定一个级别的日志的记录（例如日志级别设置为 warning，则会记录 warning、error、critical、alert、emergency 级别的所有日志）。

```bash
FortiGate # config log disk filter

FortiGate (filter) # set ?
severity             Log to disk every message above and including this severity level.
forward-traffic      Enable/disable forward traffic logging.
local-traffic        Enable/disable local in or out traffic logging.
multicast-traffic    Enable/disable multicast traffic logging.
sniffer-traffic      Enable/disable sniffer traffic logging.
ztna-traffic         Enable/disable ztna traffic logging.
anomaly              Enable/disable anomaly logging.
voip                 Enable/disable VoIP logging.
dlp-archive          Enable/disable DLP archive logging.

FortiGate (filter) # set severity ?
emergency       Emergency level.
alert           Alert level.
critical        Critical level.
error           Error level.
warning         Warning level.
notification    Notification level.
information     Information level.
debug           Debug level.
```

2. 有些情况下，用户想要自定义本地或 syslog 日志的类型或级别。

   - **举例 1：只需要记录或发送流量日志中包含源地址为 192.168.100.77 或 192.168.100.177，同时，协议为 TCP 的日志。**
   - **举例 2：只需要记录或发送事件日志中级别为 alert 或 information，同时用户为 test 的日志。**

3. 这种情况下通过简单的 log filter 是无法实现的。

解决方案

通过配置 free-style 类型的 log filter，允许用户为每个日志记录设备配置个性化的日志类型/级别组合。筛选器可以包括日志类别和特定日志字段，可以创建为包含（include）关系和排除（exclude）关系，该方式需要通过 CLI 配置，配置方法如下，可配置多个 free-style ID，多个 free-style ID 过滤的日志均会被记录/发送。

```bash
config log disk filter
    config free-style
        edit <id>
            set category <option>
            set filter <string>
            set filter-type {include | exclude}
        next
    end
end
```

选项	说明
`category <option>`	日志类型： `traffic` , `event` , `virus` , `webfilter` , `attack` , `spam` , `anomaly` , `voip` , `dlp` , `app-ctrl` , `waf` , `dns` , `ssh` , `ssl` , `file-filter` , `icap` , `ztna`
`filter <string>`	过滤关键字，可填写多个关键字，通过“or”或“and”字符指定多个关键字之间的关系，举例：set filter "((srcip 192.168.100.77) and (dstip 8.8.8.8)) or (srcip 192.168.100.177)"，表示过滤源目 IP 字段为 192.168.100.177 和 8.8.8.8，或源地址为 192.168.100.177 的日志。
`filter-type {include exclude}`	过滤器类型：include 表示包含，exclude 表示排除

配置举例

我们使用功能需求中的举例 1 和举例 2 来进行配置。

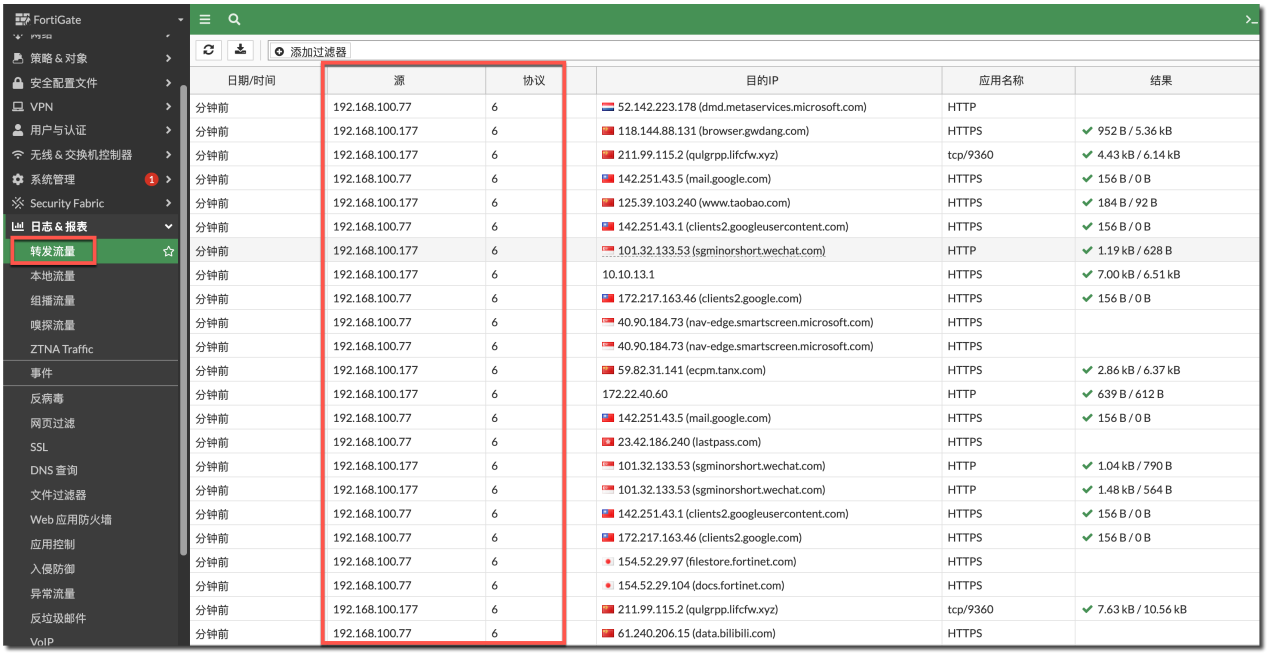
举例 1

只需要记录流量日志中包含源地址为 192.168.100.77 或 192.168.100.177，同时，协议为 TCP 的日志。

1. 在 FortiGate 的 CLI 下配置 free-style 类型的过滤器，需要注意的是，freestyle 过滤的日志类型和级别，需要在 filter 中开启（同理可以配置 syslog 和 FortiAnalyzer 日志的过滤器）。

```bash
config log disk filter
    set forward-traffic enable    //首先需要确认流量日志的总开关是开的//
    set set severity information    //freestyle日志记录的流量日志的级别为information，所以filter中的日志级别至少要设置为information//
    config free-style
        edit 1
            set category traffic
            set filter "((srcip 192.168.100.77) or (srcip 192.168.100.177)) and  proto 6"
            set filter-type include
        next
    end
end
```

2. 进入 FortiGate 的“日志 & 报表 → 转发流量”页面，查看相关日志，可以看到只有源地址为 192.168.100.77 或 192.168.100.177，同时，协议为 TCP 的流量日志被记录。



日期/时间	源	协议	目的IP	应用名称	结果
分钟前	192.168.100.77	6	52.142.223.178 (dmd.metaservices.microsoft.com)	HTTP	
分钟前	192.168.100.177	6	118.144.88.131 (browser.gwdang.com)	HTTPS	952 B / 5.36 kB
分钟前	192.168.100.177	6	211.99.115.2 (qulgrpp.lifcfw.xyz)	tcp/9360	4.43 kB / 6.14 kB
分钟前	192.168.100.77	6	142.251.43.5 (mail.google.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.177	6	125.39.103.240 (www.taobao.com)	HTTPS	184 B / 92 B
分钟前	192.168.100.77	6	142.251.43.1 (clients2.googleusercontent.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.177	6	101.32.133.53 (sgminorshort.wechat.com)	HTTP	1.19 kB / 628 B
分钟前	192.168.100.177	6	10.10.13.1	HTTPS	7.00 kB / 6.51 kB
分钟前	192.168.100.77	6	172.217.163.46 (clients2.google.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.77	6	40.90.184.73 (nav-edge.smartscreen.microsoft.com)	HTTPS	
分钟前	192.168.100.77	6	40.90.184.73 (nav-edge.smartscreen.microsoft.com)	HTTPS	
分钟前	192.168.100.177	6	59.82.31.141 (ecpm.tanx.com)	HTTPS	2.86 kB / 6.37 kB
分钟前	192.168.100.177	6	172.22.40.60	HTTP	639 B / 612 B
分钟前	192.168.100.77	6	142.251.43.5 (mail.google.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.77	6	23.42.186.240 (lastpass.com)	HTTPS	
分钟前	192.168.100.177	6	101.32.133.53 (sgminorshort.wechat.com)	HTTP	1.04 kB / 790 B
分钟前	192.168.100.177	6	101.32.133.53 (sgminorshort.wechat.com)	HTTP	1.48 kB / 564 B
分钟前	192.168.100.77	6	142.251.43.1 (clients2.googleusercontent.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.77	6	172.217.163.46 (clients2.google.com)	HTTPS	156 B / 0 B
分钟前	192.168.100.77	6	154.52.29.97 (filestore.fortinet.com)	HTTPS	
分钟前	192.168.100.77	6	154.52.29.104 (docs.fortinet.com)	HTTPS	
分钟前	192.168.100.177	6	211.99.115.2 (qulgrpp.lifcfw.xyz)	tcp/9360	7.63 kB / 10.56 kB
分钟前	192.168.100.77	6	61.240.206.15 (data.bilibili.com)	HTTPS	

3. 使用 CLI 设置该 freestyle 过滤器，以及过滤器的 dump。

```bash
FortiGate # execute log filter free-style  "((srcip 192.168.100.77) or (srcip 192.168.100.177)) and (proto 6)"
FortiGate # execute log display
...

FortiGate # execute log filter dump
category: traffic
device: disk
start-line: 11
view-lines: 10
max-checklines: 0
HA member:
log search mode: on-demand
pre-fetch-pages: 2
Filter: ((srcip 192.168.100.77) or (srcip 192.168.100.177)) and (proto 6)
Oftp search string:  (and  (or proto==6)  (or  (or srcip==192.168.100.177)  (or srcip==192.168.100.77)))
```

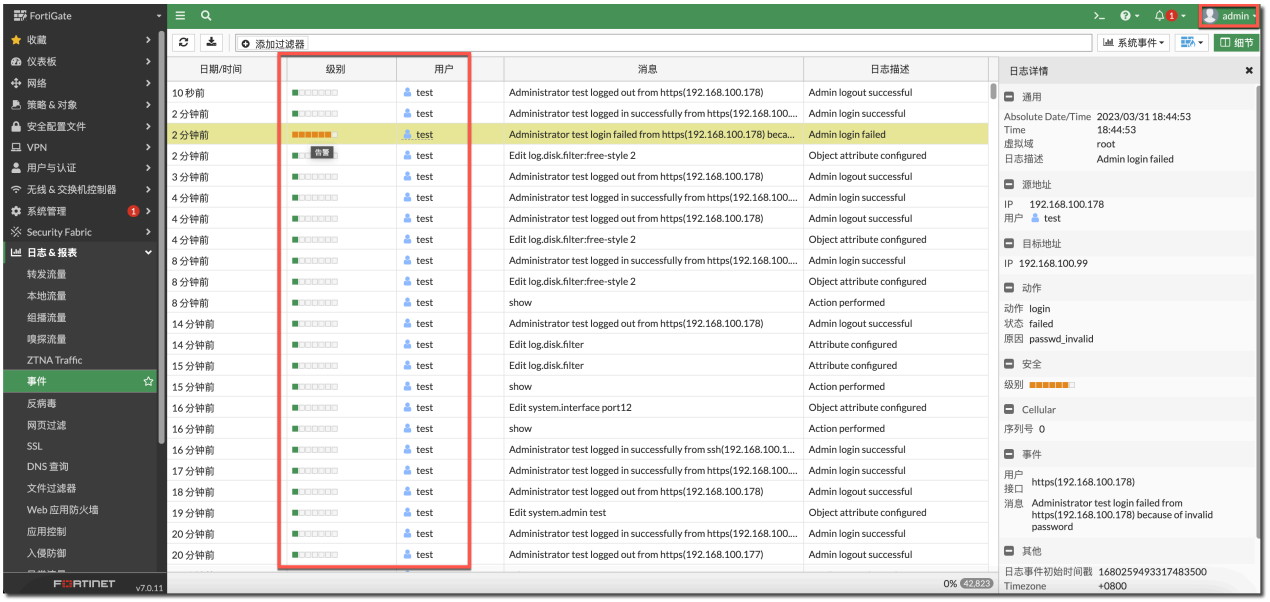
举例 2

只需要记录或发送事件日志中级别为 alert 或 information，同时用户为 test 的日志。

1. 在 FortiGate 的 CLI 下配置 free-style 类型的过滤器，需要注意的是，freestyle 过滤的日志类型和级别，需要在 filter 中开启（同理可以配置 syslog 和 FortiAnalyzer 日志的过滤器）。

```bash
config log disk filter
    set set severity information    //freestyle日志记录的事件日志的级别为warning或information，所以filter中的日志级别至少要设置为information//
    config free-style
        edit 1
            set category traffic
            set filter  "((level alert) or (level information)) and (user test)"
            set filter-type include
        next
    end
end
```

2. 使用 admin 账户登录 FortiGate，进入“日志 & 报表 → 事件 → 系统事件”页面，查看相关日志，可以看到事件日志中只有级别为 alert 或 information，同时用户为 test 的日志。



3. 使用 CLI 设置该 freestyle 过滤器，以及过滤器的 dump。

```bash
FortiGate # execute log filter free-style "((level alert) or (level information)) and (user test)"
FortiGate # execute log display
...

FortiGate # execute log filter dump
category: traffic
device: disk
```

```
start-line: 1
view-lines: 10
max-checklines: 0
HA member:
log search mode: on-demand
pre-fetch-pages: 2
Filter: ((level alert) or (level information)) and (user test)
Oftp search string:  (and  (or user=="test" not-exact)  (or  (or level=="information" not-exact)  (or level=="alert"
not-exact)))
```

举例 3

Syslog 发送的日志中排除本地流量日志、流量日志、应用控制日志和 Web 过滤日志，剩余其他类型日志正常发送。

在 FortiGate 的 CLI `config log syslogd filter` 下配置关闭 `local-traffic` 发送。配置 free-style 类型的过滤器（同理可以配置 Disk 和 FortiAnalyzer 日志的过滤器）。

> **重要**
>
> `set filter "(logid *)"` 可以表示该分类下的所有日志。

```bash
config log syslogd filter
    set local-traffic disable
    config free-style
        edit 3
            set category traffic
            set filter "(logid *)"
            set filter-type exclude
        next
        edit 4
            set category app-ctrl
            set filter "(logid *)"
            set filter-type exclude
        next
        edit 5
            set category webfilter
            set filter "(logid *)"
            set filter-type exclude
        next
    end
end
```

注意事项

1. 在 6.4 版本中 freestyle 的 log filter 配置方式与 7.0 有所不同，6.4 版本中自定义 filter 中多个条件的关系为或（or）。如果 6.4 版本配置了自定义的 log filter 后，再升级到 7.0 版本后，或（or）的关系配置会被转换为与（and）的关系，请在升级后自行手动更改（将 and 改为 or）。

```bash
6.4版本：
config log disk filter
    set severity information
    set filter "event-level(warning) logid(44547)"    //6.4版本中filter中的多个条件关系为或（or）//
    set filter-type include
end

7.0版本：
config log disk filter
    config free-style
        edit 1
            set category event
            set filter "((level warning error critical alert emergency) and (logid 44547))"    //升级到7.0后，配置被转换为与
（and）的关系//
        next
    end
end
```

2. 多个 free-style ID 之间为与（and）的关系，多个 free-style ID 过滤的日志均会被记录/发送。

3. Log ID 说明：

- `logid` 是一个 10 位数字的日志唯一标识。
- 相同类型、相同原因的日志，具有相同的 `logid`。
- 不同日志类别的 `logid` 前缀不同：
  - 流量（Traffic）日志：`00`。
  - 事件（Event）日志：`01`。

CLI 审计日志

功能介绍

默认配置下，FortiGate 的系统事件日志不会记录在 CLI 下的相关操作（除了 execute 命令）。

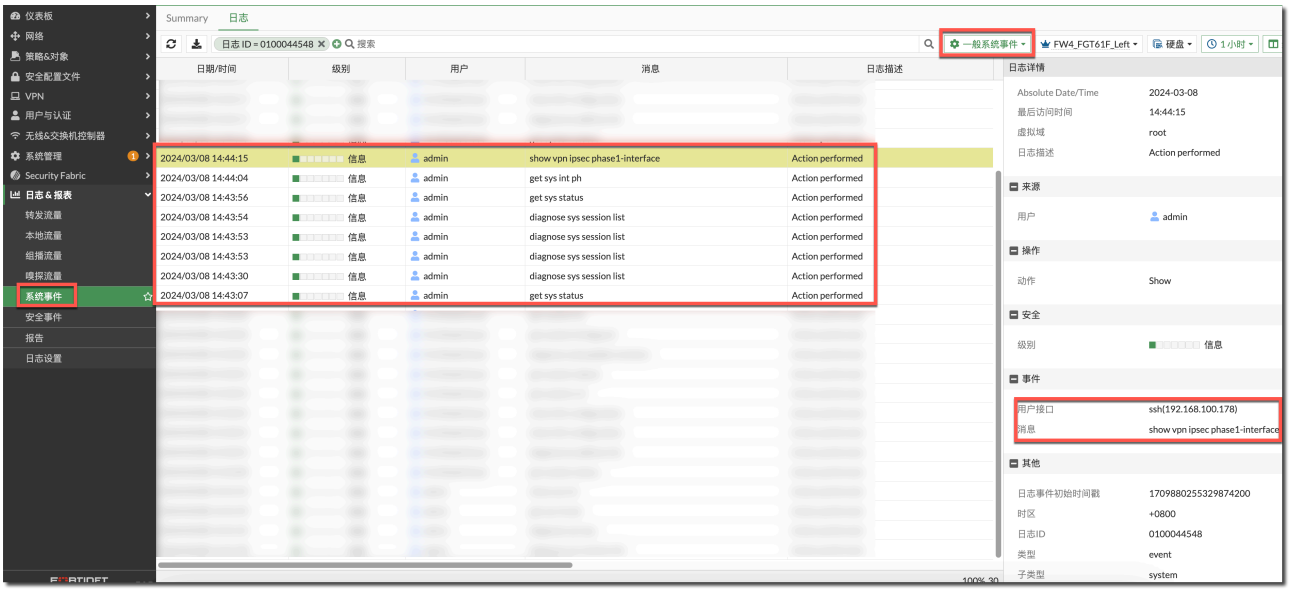
如果想要记录 CLI 下的相关操作日志，可以通过如下命令开启记录。

```bash
config system global
    set cli-audit-log enable
end
```

开启此功能后，FortiGate 会记录在 CLI 下所有以 `show`、`get`、`diagnose` 开头执行成功的命令。

日志示例

Log ID 为 `0100044548`，级别为 `Information`，日志类别为“一般系统事件”。




```bash
date=2024-03-08 time=14:44:15 eventtime=1709880255329874080 tz="+0800" logid="0100044548" type="event" subtype="system" level="information" vd="root" logdesc="Action performed" user="admin" ui="ssh(192.168.100.178)" action="Show" msg="show vpn ipsec phase1-interface"
```

日志显示地区/主机名

本文介绍了“network-visibility”功能，在流量日志中显示源/目的 IP 对应的国家/地区、对应国家/地区的图标，以及目标 IP 对应的主机名，可通过如下 CLI 开启或关闭。

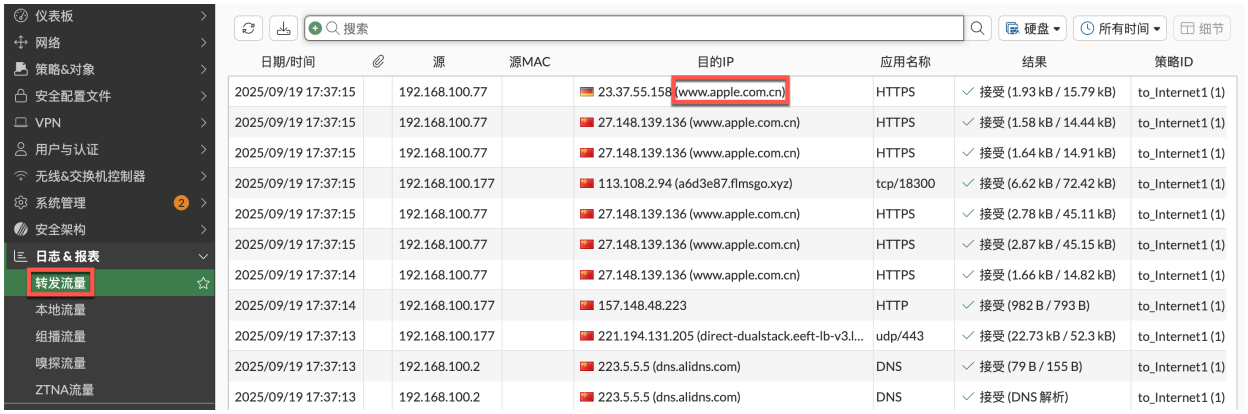
```
config system network-visibility
    set destination-visibility enable
    set source-location enable
    set destination-hostname-visibility enable
    set destination-location enable
end
```

bash

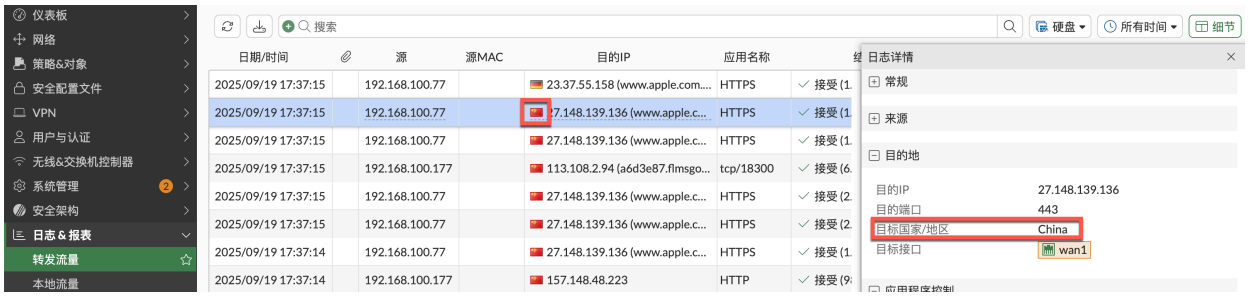
- `destination-visibility`：在转发流量日志中显示目标国家/地区，默认开启。
  - `destination-hostname-visibility`：需要先开启 `destination-visibility`。在流量日志中的目标 IP 中嵌入对应的主机名。主机名将与 IP 地址一同显示在转发流量日志的“目的 IP”字段中。默认开启。

    重要

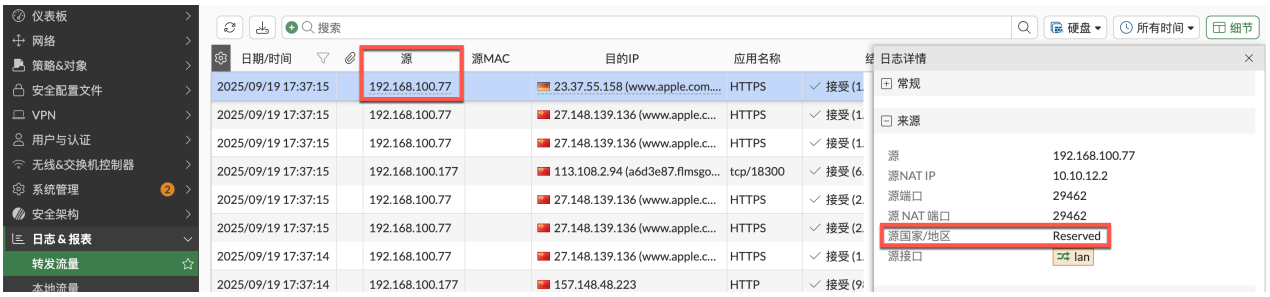
    该功能通过 FortiGate 进行 DNS 反查实现，所以需要 FortiGate 可以正常连接 Internet。



  - `destination-location`：需要先开启 `destination-visibility`。在转发流量日志中显示目标国家/地区字段及目标国家“标识”。默认开启。



- `source-location`：在转发流量日志中显示“源国家/地区”（对于内部 IP 源将显示“Reserved”标记）。默认开启。



硬盘与 Flash 闪存

本文档主要介绍 FortiGate 硬盘与 Flash 闪存相关调试方法。

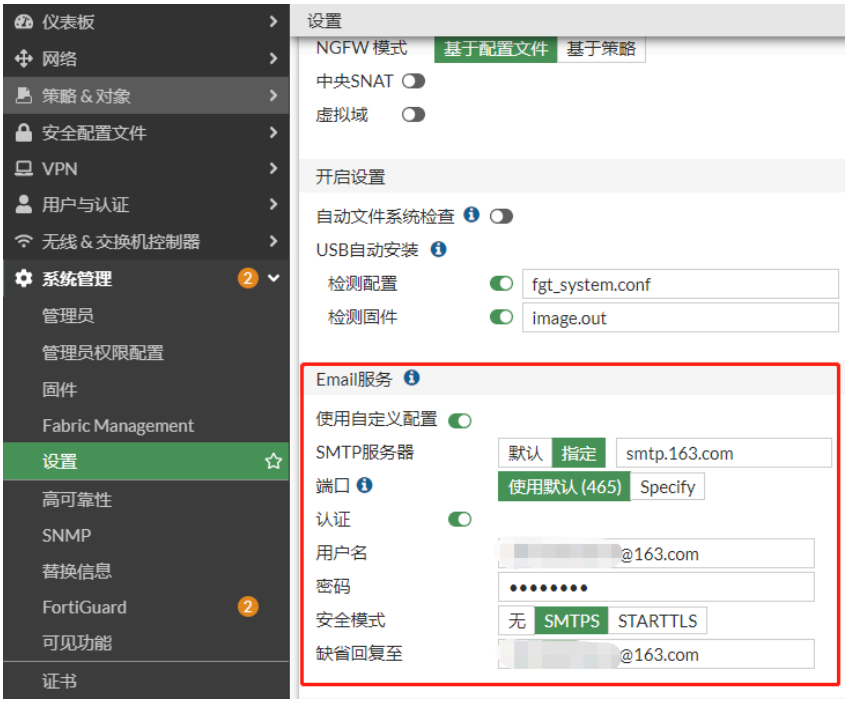
请参考：FortiGate 硬盘与 Flash 闪存 ↗。

邮件告警

邮件告警用于通知管理员有关 FortiGate 设备上的事件，从而快速处理问题。

邮件告警配置

1. 配置 Email 服务：这里使用 163 邮件发邮件。更多 Email 服务设置请参考系统管理 → 系统设置 → Email 服务 ↗章节。



2. 配置邮件告警：配置发件人和收件人。

```bash
config alertemail setting
    set username "186xxxxxxxx@163.com" ←——username和email服务中的用户名一致，即邮箱账号；如果使用ForitGate默认的邮件服务，则username配置为DoNotReply@notification.fortinet.net。
    set mailto1 "test1@qq.com"
end
```

3. 可以配置多个收件人。

```bash
config alertemail setting
    set mailto2 "test2@qq.com"
    set mailto3 "test3@qq.com"
end
```

4. 配置邮件告警的内容：FortiGate 可以基于日志类别发送邮件告警，也可以基于日志安全级别发送邮件告警。

   - 基于日志的类别：

```
config alertemail setting
    set username "186xxxxxxxx@163.com"
    set mailto1 "test1@qq.com"
    set filter-mode category ←——过滤模式为类别
    set IPS-logs enable ←——根据需求选择需要发送的日志类别，如这里发送IPS和AV日志
    set antivirus-logs enable
    set ? ←——查看支持的类别
end
```

   - 基于安全级别：

```bash
config alertemail setting
    set username "186xxxxxxxx@163.com"
    set mailto1 "test1@qq.com"
    set filter-mode threshold  ←——过滤模式为安全级别
    set severity critical ←——critical 及以上级别的日志发送日志告警
end

# 日志的级别从高到底为：
emergency       Emergency level.
alert           Alert level.
critical        Critical level.
error           Error level.
warning         Warning level.
notification    Notification level.
information     Information level.
debug           Debug level.
```

日志常见问题

- 日志存储方式
- 日志类型
- 删除日志
- 导出日志
- 自定义日志过滤器
- CLI 审计日志
- 日志显示地区/主机名
- 硬盘与 Flash 闪存
- 邮件告警
- 日志常见问题

常见问题

- 产品知识
- 防火墙部署
- 功能配置
- 流量无法被 NP 加速
- 阻断/认证页面无法点击继续
- 8008/8010/8015/8020 端口
- FortiGate Cloud 无法登录
- 获取接口对应的公网 IP

产品知识

1.FortiGate 系列全新下一代防火墙硬件架构是什么样子的?

答：FortiGate 采用多核 CPU + ASIC 芯片的一种处理架构，芯片分为两种、一种是 NP 芯片、一种是 CP 芯片，NP 芯片是处理网络层以及传输层的一些数据转发、可以达到线速转发。一旦防火墙建立会话之后，在后续的匹配同一个会话的数据包都会有 NP 芯片直接处理转发。还有一种是 CP 芯片，CP 芯片主要是用于处理应用层内容加速的，比如 ips、防病毒、应用过滤等需要应用层检测的都会由 CPU 送到 CP 芯片进行加速处理。正式因为有了 NP 芯片和 CP 芯片的处理架构，所以 FortiGate 具有非常高的性能。

2.哪些型号带交换口？是否可以转换成路由口？如何转换?

FortiGate60E、FortiGate100E、FortiGate100F 等低端型号设备带交换口，均可以转换为路由口， 转换方式如下（以 FortiGate60E 举例）:

> **重要**
>
> 执行操作前，确保 internal 或 lan 口没有被防火墙策略、dhcp、路由等引用。

```
config system virtual-switch
     edit internal
           config port
                delete internal1
                delete internal2
      end
end
```

3.FortiGate60E、FortiGate100E、FortiGate100F 设备面板的小孔是否可以用来恢复出厂

答：设备面板的小孔可以用来恢复出厂。防火墙系统正常启动后（可以 ping 通或登录管理页面）30S 内，长按 10S 左右，系统将自动重启，届时将恢复为出厂设置；

4.combo 光电复用口默认什么优先，如何切换?

答：combo 口没有光电优先之分，插什么口，什么口都可以用，不需要切换。

防火墙部署

1.防火墙支持几种模式

答：防火墙支持 nat（路由模式）、透明模式两种模式，不支持混合模式。

2.透明模式和 nat (路由模式) 有什么区别

答：透明模式不支持路由功能、不支持接口模式 ipsec vpn、不支持 sslvpn、l2tp、pptp、gre 功能，仅仅为二层转发。

3.如何实现混合模式场景需求

答：如果需要实现混合模式的场景需求，需要采用 VDOM 方式来实现，把防火划分成两个虚拟域，一个虚拟域做透明模式，另外一个虚拟域做路由模式。

4.什么是 VDOM（虚拟域）模式

答：虚拟域 (VDOM)，可以理解为虚拟防火墙，是一种将一个 FortiGate 设备分为两个或两个以上虚拟设备的技术，它能起到多个独立设备的作用。VDOM 可提供单独防火墙策略。在 NAT/路由模式下，它可完全分开配置，从而为各连通网络或组织提供路由或 VPN 服务，同时 VDOM 之间也可以互访。设备的系统资源可以手动分配给不同的 VDOM。一般常用于需要完成独立分割的多个网络，比如云环境；当存在多个区域，即有透明模式又有路由模式的混合环境，由于 FortiGate 只支持透明模式和 NAT/路由模式，所以这种环境必须用 VDOM 功能来解决。

5.防火墙是否支持旁路部署模式?

答：支持旁路部署模式，接口开启 sniffer 模式，做数据分析之用，使用场景较少，可以用于 ips 测试回访攻击包使用 。

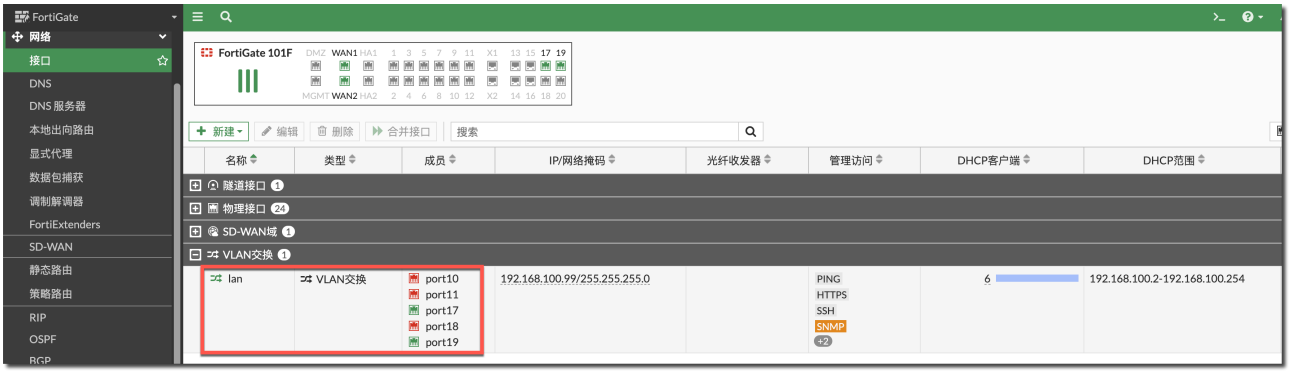
6.wan 口可以单做内网口使用吗?

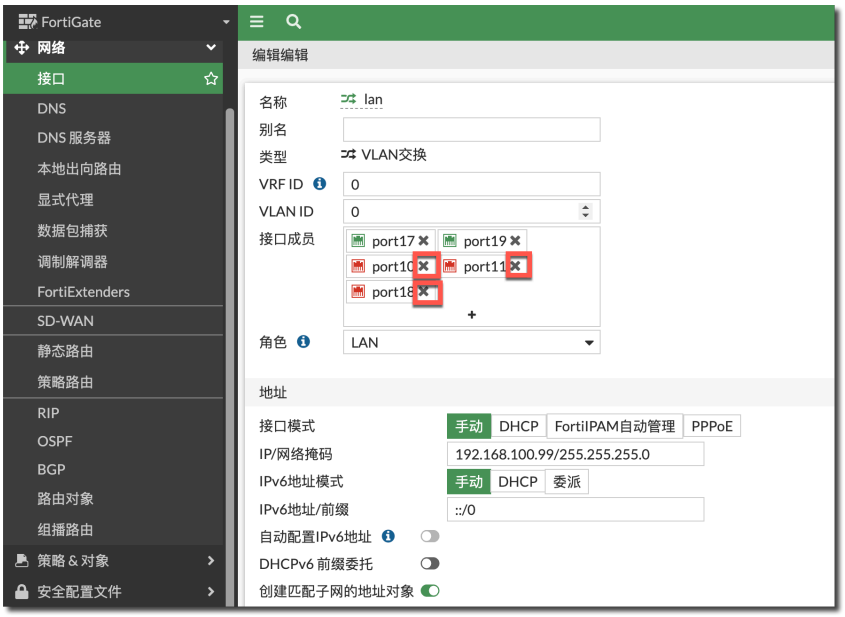
答：wan 口可以做内网口使用，接口的名称只是一个代号，不具有特殊特性。可以按照自己习惯以及现场环境灵活使用。

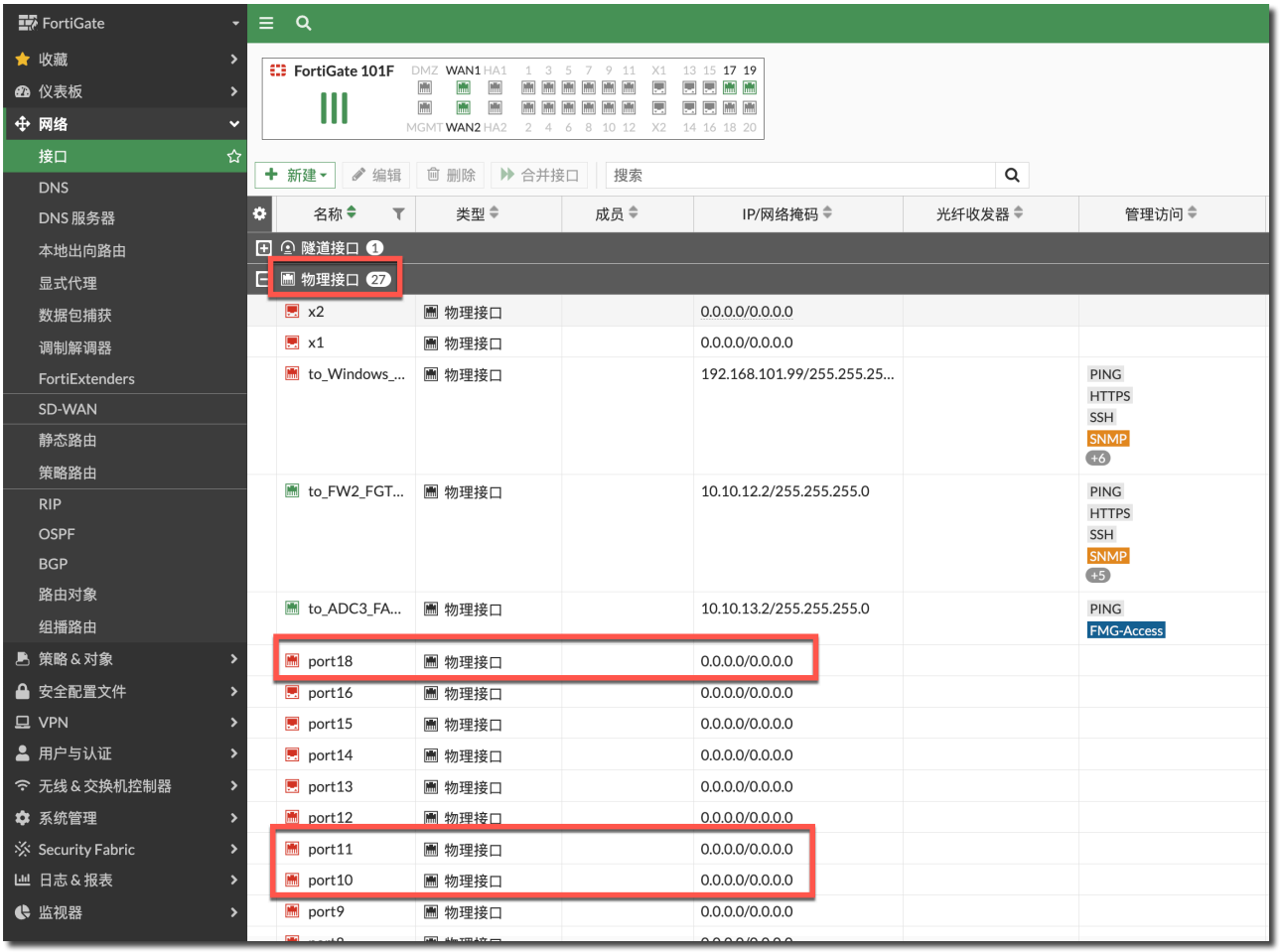
功能配置

1.为什么我的防火墙有些接口在“网络 → 接口”页面无法显示，命令行无法编辑，无法单独配置 IP 地址。

答：部分型号的 FortiGate（如 FortiGate 101F、60F、80F）内置有一个 Hardware Switch（VLAN Switch）接口，默认情况下在配置文件中该接口被命名为 lan 或 internal，这是一个交换接口，默认配置下，与该 Hardware Switch（VLAN Switch）内部相连的接口均会被加入该接口（二层接口，类似于交换机的 Access 接口）。如果想单独配置某个 Hardware Switch（VLAN Switch）的成员接口，需要将该接口从 Hardware Switch（VLAN Switch）接口中移除，移除后，成员接口会变为独立的三层接口，可以独立配置 IP 等信息。







2.如何实现源 NAT，防火墙策略中启用 NAT、ip 地址池、和中央 NAT 分别是什么意思?

答：源 NAT：即源地址转换，源 NAT 在防火墙策略中实现，通过开启 NAT、引用 ip 地址池、和引用中央 NAT 来实现源地址转换。

策略中启用 NAT：代表源地址转换为此策略出接口的 ip 地址。

ip 地址池：如果选择了 ip 地址池，代表源地址转换为 ip 地址池中的 ip 地址，或者也可以直接定义源地址对应关系，不受策略限制。

中央 NAT：中央 NAT 是一个单独的地址转换表，防火墙策略选择了中央 NAT 后，会根据策略的源地址去到中央 NAT 查找匹配相应的转换关系。

3.什么是虚拟 ip?

答：虚拟 ip 即 vip，虚拟 ip 是用来做目的地址转换使用，实现目的映射的对应关系的策略，需要引用到防火墙策略中才能生效，需要注意，虚拟 ip 的映射关系会影响到源地址转换，也就是，防火墙会优先匹配虚拟 ip 的映射关系，再匹配防火墙策略中选择的源地址转换方式。

4.如何实现目的 NAT?

答：目的 NAT 即目的转换，目的转换在防火墙策略中实现，通过事先新建一个 vip，然后在防火墙策略中的目的选择 vip 即可实现目的地址映射。

5.如何实现双向 NAT?

答：如果要实现双向地址转换，在防火墙策略的目的选择相应的 vip，然后在策略中启用 NAT 或者引用 ip 地址池，或者直接使用中央 NAT。

6.接口和区的有什么关系

答：区域可以包含多个接口，可以理解为接口组，接口一旦划入到区中，就不能单独做策略，在防火墙策略中只能找到相应的区，而无法选择所划入区的接口了，所以区通常较少使用。

7.内网用户需要通过公网映射地址访问内部服务器，需要什么特殊配置?

答：请参考策略与对象 → 虚拟 IP 映射 → 内网主机通过公网 IP 访问 VIP ↗ 章节。

8.HA 主备模式系统升级是否会中断业务，如何操作?

答：主备防火墙升级系统不会中断，只要按照正常的方式升级即可，通过管理主墙，上传版本文件后，主墙把版本文件同步给备墙，备墙先升级，备墙升级完后，主墙开始升级，升级过程中会切换到备墙工作，整个过程透明，无需其他额外操作，通单台升级操作一样。

9.sslvpn 支持几种模式? 是否支持 sslvpn 的 site-to-site 模式?

答： a. 支持代理模式：代理模式只支持 B/S 架构的 http、https 应用代理

b. 支持客户端模式：即隧道模式，无应用限制。

c. 不支持插件方式

d. 不支持 site-to-site 模式

10.移动终端 vpn 支持情况?

答：IPSEC vpn 支持 ios 和 android，ios 和 android 均采用系统自带的客户端

SSLVPN 支持支持 ios 和 android，FortiExplore VPN 支持 ios，FortiClient 支持 Android。

11.防火墙支持 stp 功能吗?

答：低端型号如 FortiGate 60E，FortiGate 100E，FortiGate 100F 的硬交换接口支持。

12.防火墙是否支持 802.1x ?

答：低端型号如 FortiGate 60E，FortiGate 100E，FortiGate 100F 的硬交换接口支持。

13.防火墙支持链路聚合 802.3ad 功能吗?

答：支持

14.防火墙聚合口成员可以从聚合口中分离吗？需要注意什么?

可以从聚合口中取出，但需要注意：成员接口从聚合口中取出后，务必重启一次防火墙，否则可能会出现网络通信问题，此为硬件限制，目前无解。
所以建议在部署前就规划好聚合口的成员，避免后期频繁修改聚合口成员配置。

15.转换或接口提示被占用，如何查找接口被哪些模块被调用

关联项就是接口被引用的次数，如点击 "7" 可以查看具体被哪些模块引用。

16.防火墙通过默认的 notification.fortinet.net:465 ↗ 邮件服务器，邮件方式发送 FortiToken（双因子认证）、告警或日志到特定邮箱时，为什么收件邮箱经常收不到，或被识别为了垃圾邮件?

答：出现该情况时，请在系统 → 设置 → Email Service 中配置“缺省回复至”，定义 username，域名保证为 notification.fortinet.net，就不会被其他 email server 认为是垃圾邮件。

在 FortiGate 版本 7.0.11，7.2.4 及后续的版本中，默认的 notification.fortinet.net:465 邮件服务器 将取消“缺省回复至”选项，将该选项强制为 DoNotReply@notification.fortinet.net

17.配置了 FQDN 形式的地址对象并使用静态/策略路由引用，但实际效果不稳定，客户端访问配置的 FQDN 时，仍然从其他线路出去了。

答：FQDN 地址对象中获取的 IP 地址有两种方式：1、设备使用自身配置的 DNS 解析到的结果；2、穿过防火墙的 DNS 流量被设备记录。(具体原理可以参考附件：https://handbook.fortinet.com.cn/files/fqdn.pdf)

在 FQDN 地址对象的 CLI 配置下：

1）如果配置的 cache-ttl>DNS 服务器返回的 TTL，认为该 FQDN 下面配置的 cache-ttl 时间优先。

2）如果 cache-ttl < DNS 服务器返回的 TTL，则 DNS 服务器返回的 TTL 优先。

3）如果没有没配置 cache-ttl（默认为 0），以 DNS 服务器返回的 TTL 为准。

当 TTL 变为 0 时，防火墙会删除该 FQDN 下的相关地址记录。

解决办法：在 CLI 下，编辑 FQDN 形式的地址对象，将 cache-ttl 修改为 86400。

```
config firewall address
  edit "www.baidu.com"
    set type fqdn
    set allow-routing enable
    set fqdn "www.baidu.com"
    set cache-ttl 86400
  next
end
```

16.Flow 模式和 Proxy 模式的区别。

答：

Flow 模式：

当防火墙策略的检测方式设置为 Flow 时，经过该策略的流量不会被 FortiGate 进行缓冲。与代理模式不同的是，通过策略的内容负载将被逐包检查，最后一个包由 FortiGate 保存，直到扫描返回结果。如果检测到异常流量，则向接收端发送一个重置包，终止连接，并阻止有效负载成功发送。

基于 Flow 检测能够实时识别和阻断安全威胁。所有适用的基于流的安全模块在一次通过中同时应用，使用 DFA（Direct Filter Approach）模式匹配来识别可能的攻击或威胁。模式匹配由 CP8 或 CP9 处理器卸载和加速。

基于流程的检查通常比基于代理的检查需要更少的处理资源，并且不改变报文，除非发现威胁并阻断报文。

Proxy 模式：

当防火墙策略的检测方式设置为 Proxy 时，通过该策略的流量会被 FortiGate 进行缓冲检测。这意味着 FortiGate 将保存文件、电子邮件消息或网页的信息包，直到整个有效负载被检查是否违反（病毒、垃圾邮件或恶意 web 链接）。在 FortiOS 完成检查之后，如果流量是干净的，负载要么被发送到目的地，如果流量有异常，则被丢弃并用替换消息替换。

代理模式提供了最彻底的流量检查，但是它的彻底的牺牲了性能，使得它的吞吐量比流模式策略慢。

17.在策略中调用“应用控制”的安全配置文件后，发现微信发图片很慢，或一些手机 App 无法正常访问。

答：微信、支付宝等一些手机 App 在使用时会用到 QUIC 协议。默认情况下，应用控制安全配置文件不允许 QUIC 业务通过，此时微信在发图片时，首先会考虑使用 QUIC 协议，当 QUIC 协议被防火墙阻断后，微信再使用 HTTPS 协议进行发送，导致出现发图片慢的情况。通过在安全配置文件 → 应用控制中，将 QUIC 的动作配置为允许，可以解决此问题。

18.License 无法验证，或特征库无法更新。

答：FortiGate 与 FortiGuard 服务器通信问题造成。FortiGate 默认使用 anycast 方式连接 FortiGuard，在国内使用时这种方式可能会出现网络通讯故障，建议 FortiGate 使用如下 CLI 命令关闭 anycast 方式连接 FortiGuard，使用单播方式连接 FortiGuard。

```bash
config system fortiguard
    set fortiguard-anycast disable
    set protocol udp
    set port 8888
end

exe update-now
```

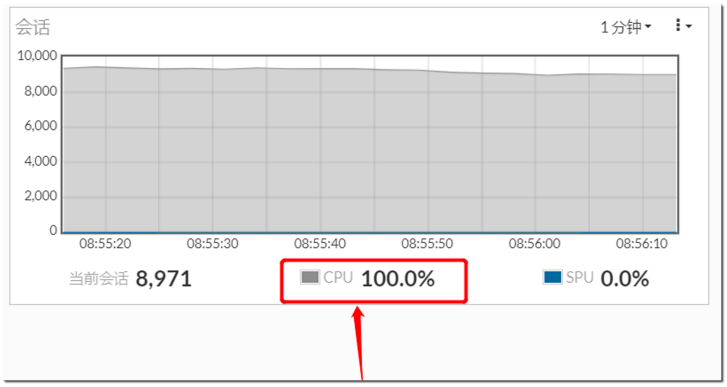
流量无法被 NP 加速

问题现象

1. 拓扑信息：

   LAN network----LAG (port9, port10)-**FortiGate**-LAG (port5/port6)----Internet

2. 在 FortiGate 200E/201E、2000E、2500E 设备上，出现大部分或所有会话均由 CPU 处理，没有被硬件加速。



3. 以 FortiGate 201E 为例，查看会话，可以看到大部分会话中的接口信息是从内网聚合口 PG_SNAT_LAN（成员为 port9 和 port10）与 port5/port6 之间的流量。

```bash
diagnose sys session list | grep dev=
dev=43→24/24→43
dev=43→23/23→43
dev=23→34/34→23
...

diagnose ip address list
...
IP=10.179.1.62→10.179.1.62/255.255.255.252 index=43 devname=PG_SNAT_LAN
IP=123.157.210.114→123.157.210.114/255.255.255.248 index=23 devname=port5
IP=39.174.163.216→39.174.163.216/255.255.255.128 index=24 devname=port6
...
```

4. 会话信息显示无法加速的原因为 `non-npu-intf`。

```bash
session info: proto=6 proto_state=01 duration=4505 expire=3587 timeout=3600 flags=00000000 socktype=0 sockport=0
av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=40MB
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty per_ip
statistic(bytes/packets/allow_err): org=442701/2567/1 reply=624811/3382/1 tuples=2
tx speed(Bps/kbps): 155/1 rx speed(Bps/kbps): 188/1
orgin→sink: org pre→post, reply pre→post dev=43→24/24→43 gwy=39.174.163.254/10.179.1.61
hook=post dir=org act=snat 10.179.16.15:61926→120.232.31.208:8080(39.174.163.216:61926)
hook=pre dir=reply act=dnat 120.232.31.208:8080→39.174.163.216:61926(10.179.16.15:61926)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=54:c6:ff:a0:18:c1
misc=0 policy_id=3 pol_uuid_idx=1050 auth_info=0 chk_client_info=0 vd=1
serial=424a822d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=80000000 ngfwid=n/a
npu_state=0×040000
//no_ofld_reason:  non-npu-intf//
```

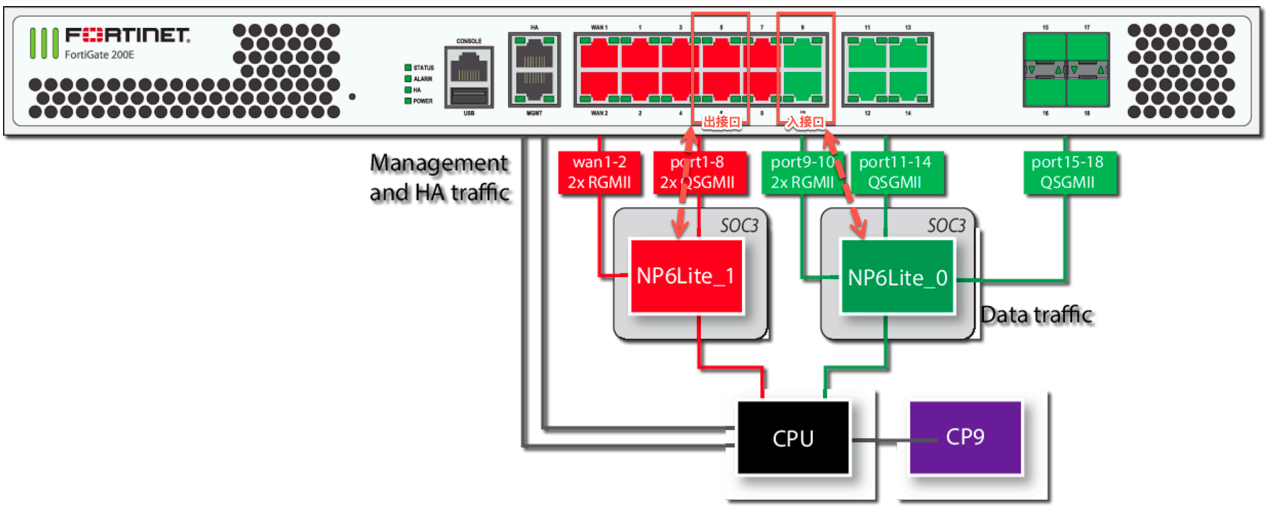
5. 关于 `non-npu-intf` 或其他无法被 NP 加速的原因，可以参考 Fortinet 官方文档 ↗。其中提到 `non-npu-intf` 的可能原因：

   - 会话的入接口或出接口不是 NP 加速接口。
   - 会话的入接口或出接口是软交换（Software Switch）接口的成员。（此例中不符合）
   - 禁用了 config system npu 下的 fastpath。（此例中不符合）

问题原因

1. 在此例中，流量从 port9/port10 进入，从 port5/port6 流出，不满足 201E 的硬件加速条件，因为 port9/port10 与 port5/port6 属于不同的 NP 芯片，且接口间的硬件结构没有内置交换模块，所以是无法被 NP 加速的。

2. 参考 FortiGate 201E 的硬件加速架构 ↗：



   - 只有当流量在连接到同一个 NP6Lite 的接口上转发时，才能被 NP 加速。
   - NP6Lite 连接的接口如下：
     - NP6Lite_0 连接到 6 个 1GE RJ-45 接口（port9 ~ 14）和 4 个 1GE SFP 接口（port15 ~ port18）。
     - NP6Lite_1 连接到 10 个 1GE RJ45 接口（wan1、wan2、port1 ~ port8）。

解决方法

- 选择流量的入接口为 port9 ~ port18，出接口为 port9 ~ port18。
- 或选择流量的入接口为 wan1/wan2、port1 ~ port8，出接口为 wan1/wan2、port1 ~ port8。
- 参考：点击链接查看 FortiGate 2000E ↗ 与 2500E ↗ 的硬件加速架构。

阻断/认证页面无法点击继续

问题描述

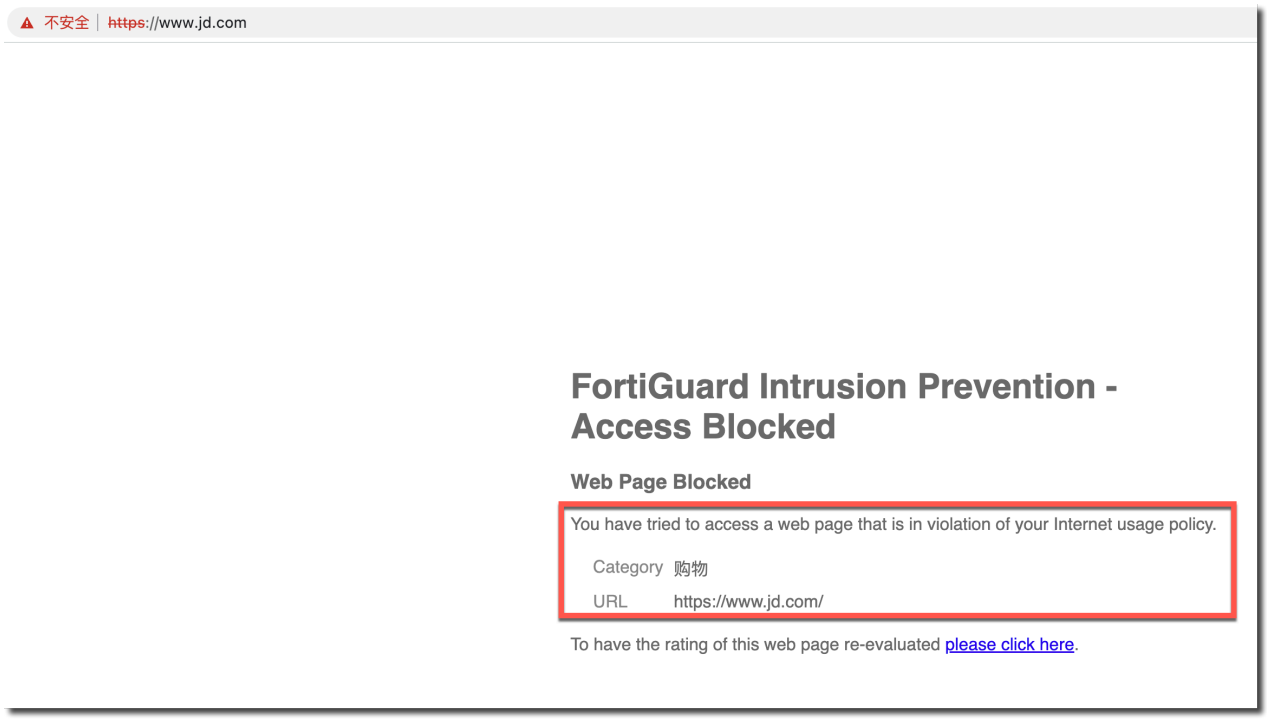
1. 当用户访问的网页被 WebFilter 功能阻断（或开启了防火墙用户认证）时，会在浏览器中弹出 WebFilter 阻断（或用户登录）页面，由于默认情况下 FortiGate 阻断页面使用的是 FortiGate 的自带证书（未被客户端信任的 CA 颁发的证书），会导致浏览器提示网页不安全（以 Chrome 为例）。





2. 该提示信息有时可以跳过，从而显示出 FortiGate 的阻断（或用户登录）页面。





3. 但有时会出现没有点击继续的按钮，导致无法继续显示 UTM 阻断页面（或用户登录页面），即使使用键入“thisisunsafe”的方法也无法继续。



问题原因

这是由于访问的网站开启了 HSTS 功能。

HSTS 介绍

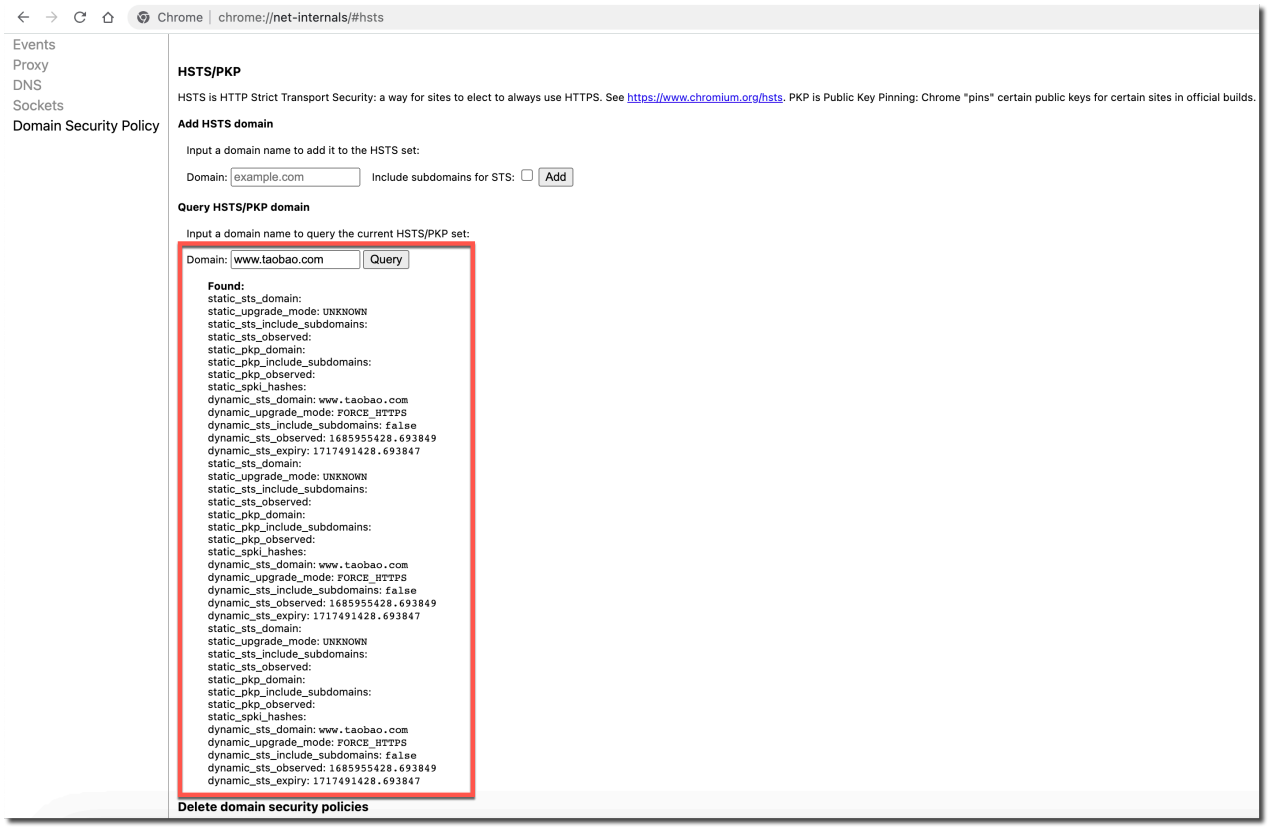
HTTP 严格传输安全（HTTPStrictTransportSecurity）是一套由互联网工程任务组发布的互联网安全策略机制。网站可以选择使用 HSTS 策略，来让浏览器强制使用 HTTPS 与网站进行通信，以减少会话劫持风险。

用户首次访问某网站是不受 HSTS 保护的。这是因为首次访问时，浏览器还未收到网站的 HSTS，所以仍有可能通过明文 HTTP 来访问。

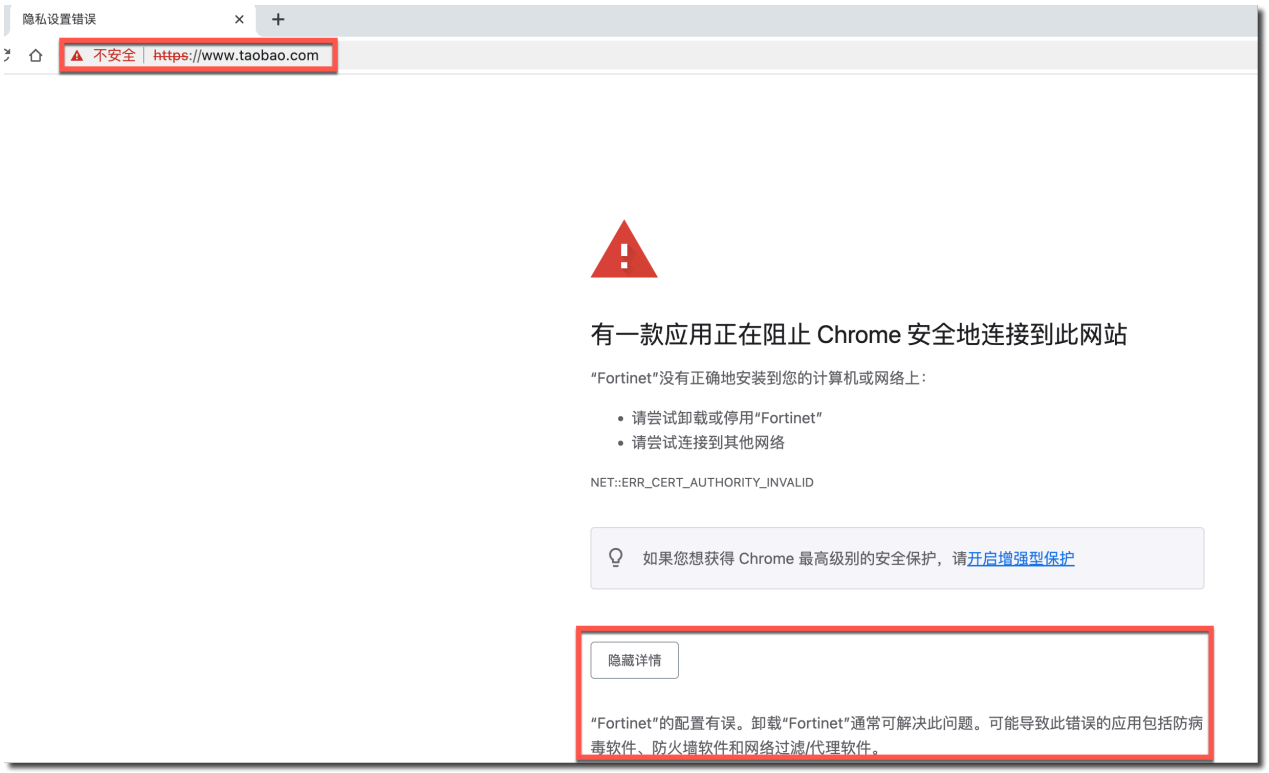
另外，如果中间人使用自己的自签名证书来替换网站证书，浏览器会给出警告（FortiGate 的 UTM 阻断页面或登录页面就是此种情况），但是许多用户会忽略警告。HSTS 为了解决了这一问题，一旦服务器发送了 HSTS 字段（HSTS 信息被储存在浏览器中），在 HSTS 老化时间到期前，将不再允许用户忽略警告。这就导致了浏览器除了第 1 次访问开启 HSTS 的网站可以点击继续按钮显示出阻断页面（或登录页面），后续再访问此网站时，没有继续按钮显示。

浏览器中的 HSTS 信息查询

1. 我们可以在 Chrome 里的 chrome://net-internals/#hsts（Edge 为 edge://net-internals/#hsts）页面下查询网站的 HSTS 信息是否被储存在了浏览器中，如下举例，www.taobao.com ↗ 网站开启了 HSTS，并在某次成功访问的时候被加入了 Chrome 的 HSTS 列表，目前还未过期。

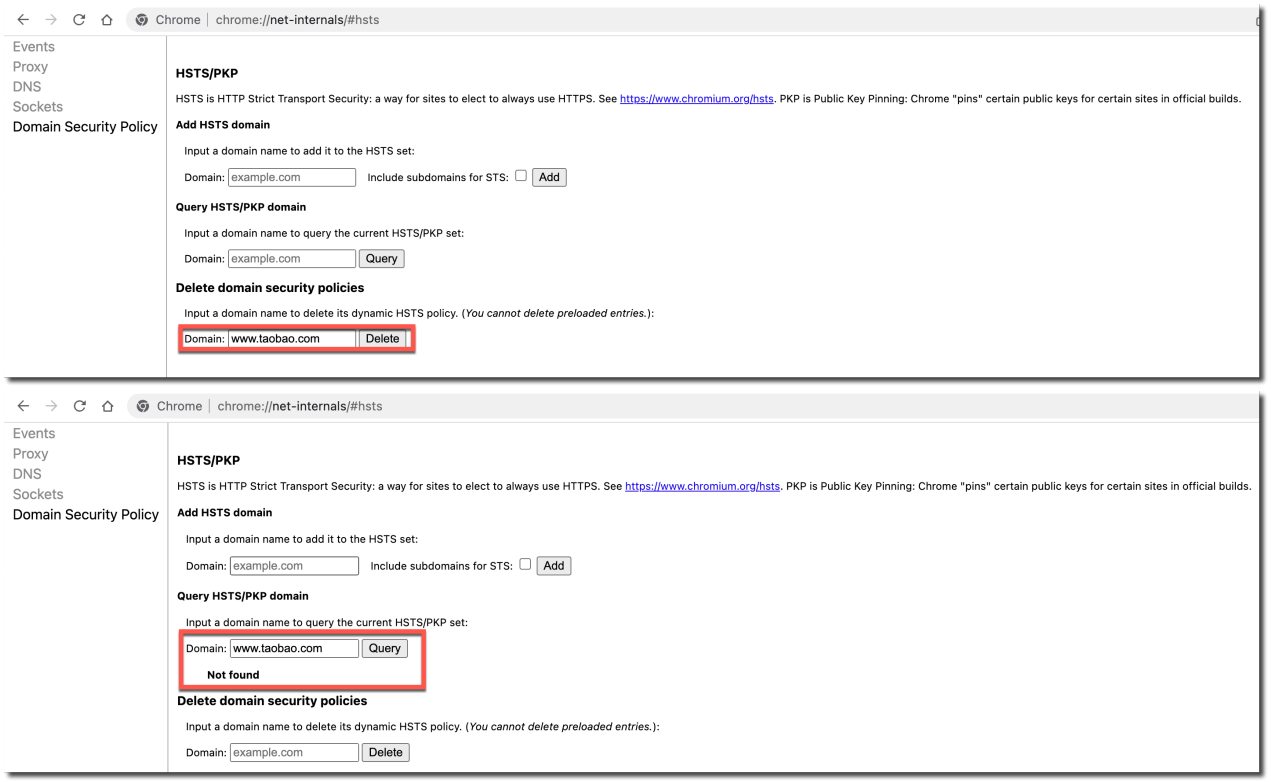


2. FortiGate 开启 WebFilter 并阻断该网站后，客户端访问该网站时会出现警告页面无法点击继续按钮的情况。

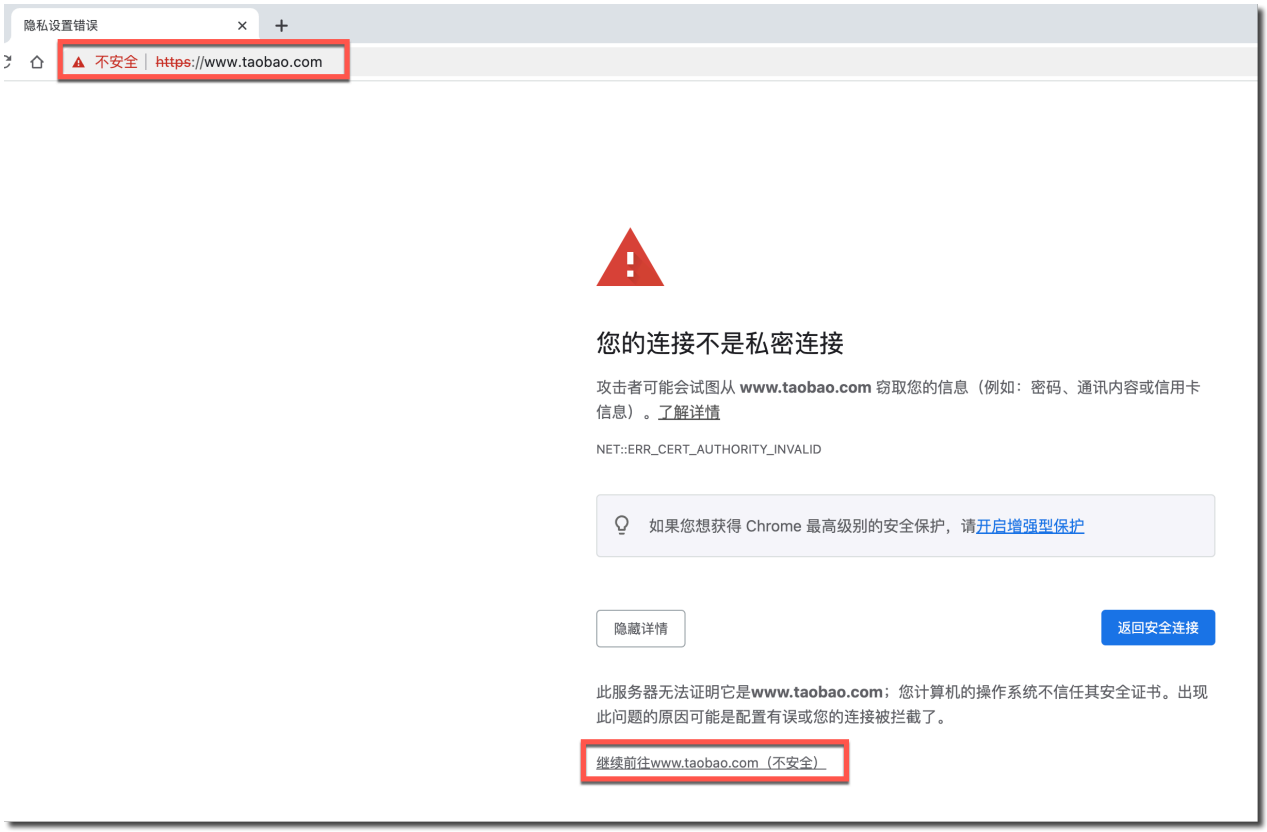


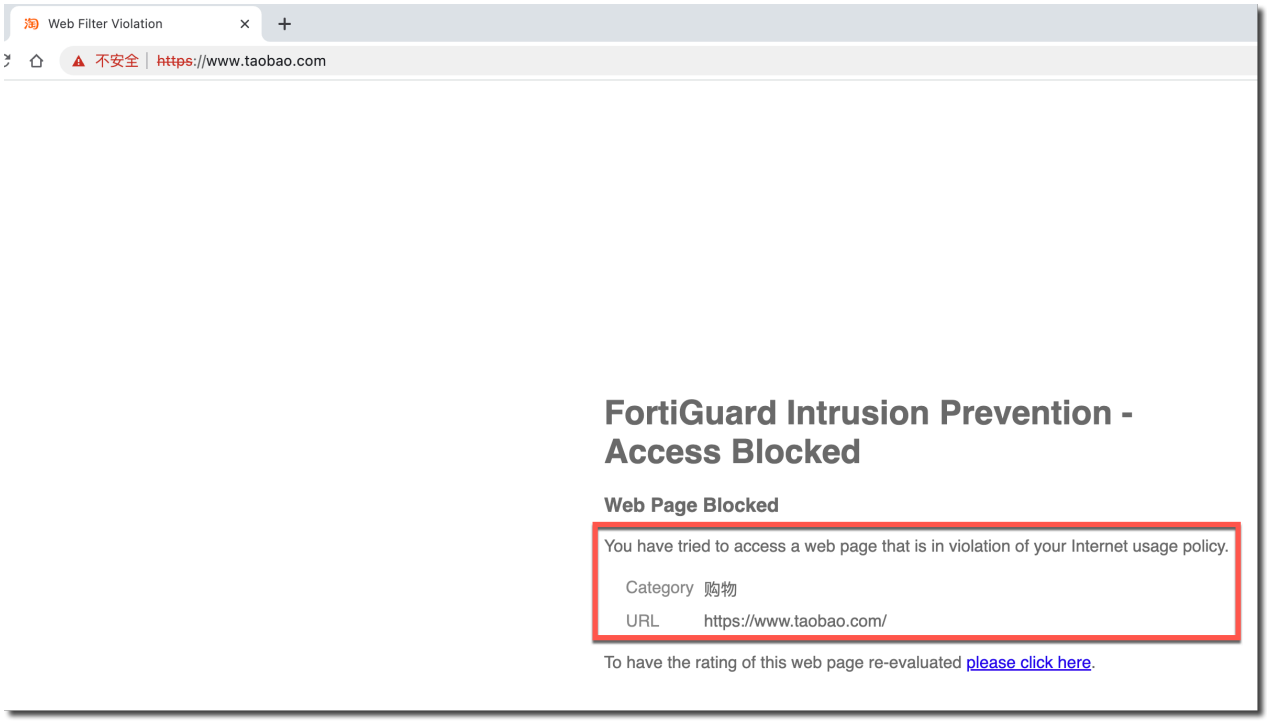
删除浏览器中的 HSTS 信息

1. 在 Chrome 里的 chrome://net-internals/#hsts（Edge 为 edge://net-internals/#hsts）页面下删除网站的 HSTS 信息，删除后，无法再查询到该网站的 HSTS 信息。

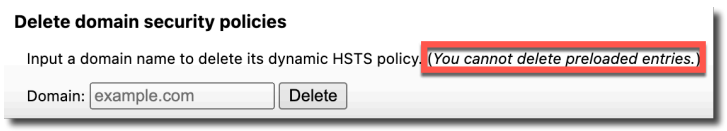


2. 客户端再次访问该网站时，警告页面可以点击继续按钮，并显示出 FortiGate 的阻断页面。





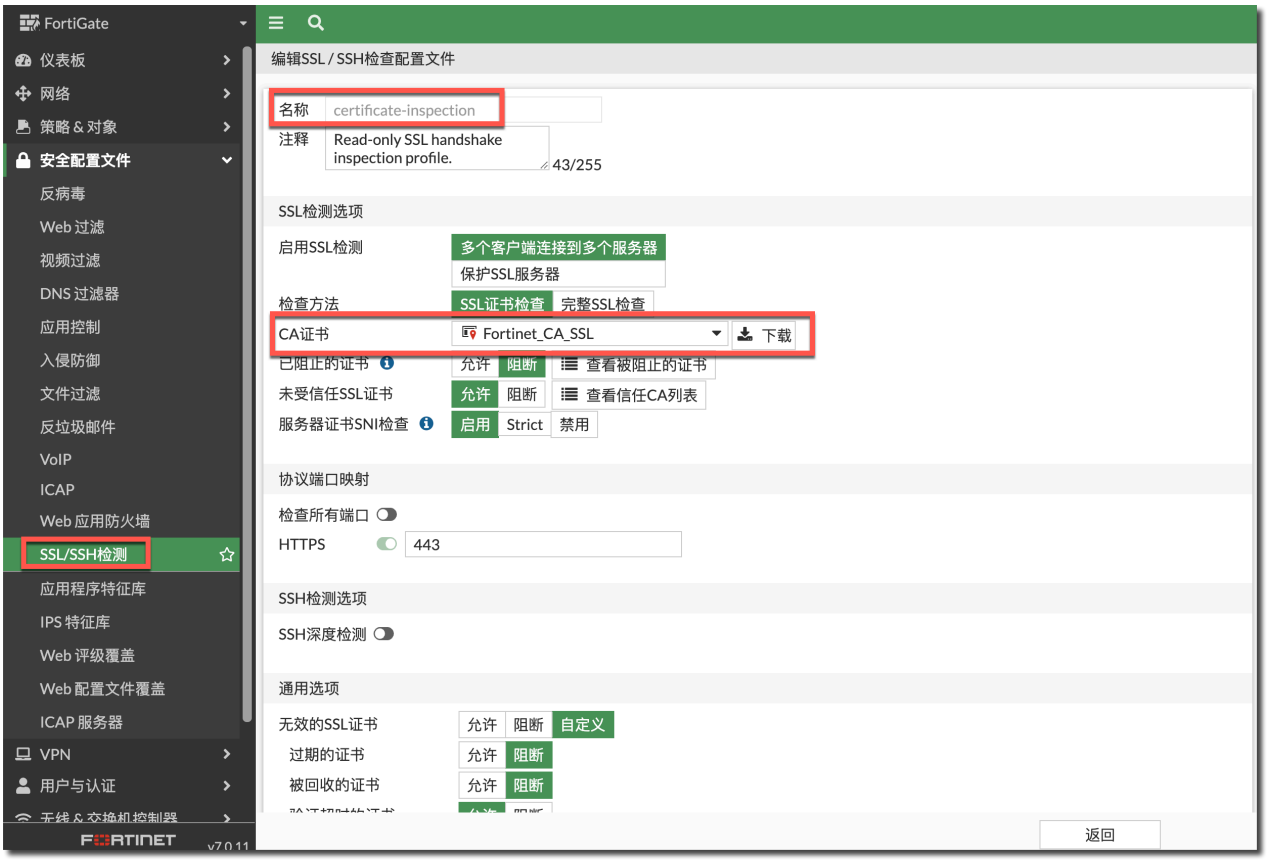
3. 请注意，有一些知名网站的 HSTS 信息被预置在了浏览器中，如 www.google.com ↗ 、www.gmail.com ↗ 、www.facebook.com ↗ 等，这些 HSTS 信息是无法被删除的。

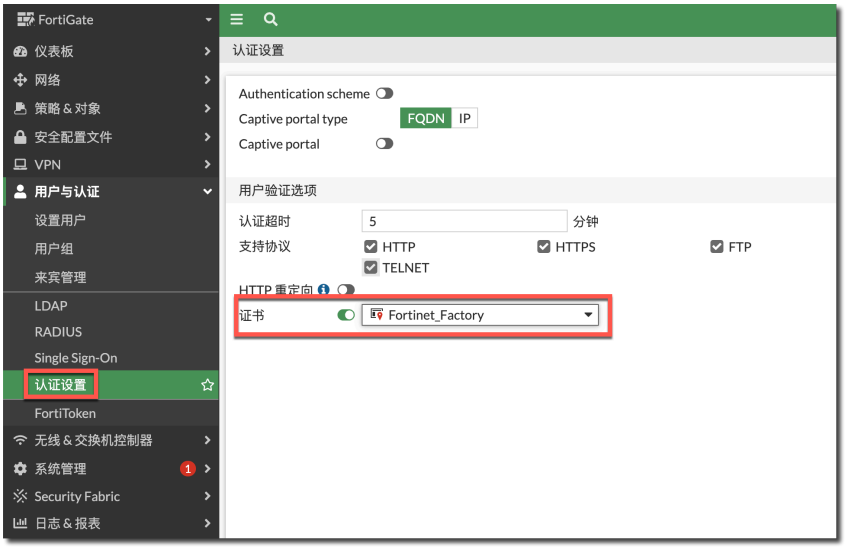


4. Firefox 需要通过删除对应网站的缓存信息来删除 HSTS 信息。

解决方法

1. 下载 FortiGate 用于 UTM 阻断页面（或用户认证页面，用户认证的证书需要到“系统管理 → 证书”页面下载）使用的 CA 证书，如下图所示，WebFilter 功能所属的安全策略使用的 SSL 检查配置文件为默认的“certificate-inspection”，编辑该 SSL 检查配置文件后，下载 CA 证书。





2. 将 FortiGate 使用的 CA 证书加入设备或者浏览器的信任列表。

重要

- Chrome/Edge 依赖系统的证书信任列表。
- 而 Firefox 有自己的信任列表，需要在浏览器设置中添加，或在浏览器的证书设置中开启“允许 Firefox 自动信任您安装的第三方证书”，开启后 Firefox 也会信任系统的证书信任列表。



Windows

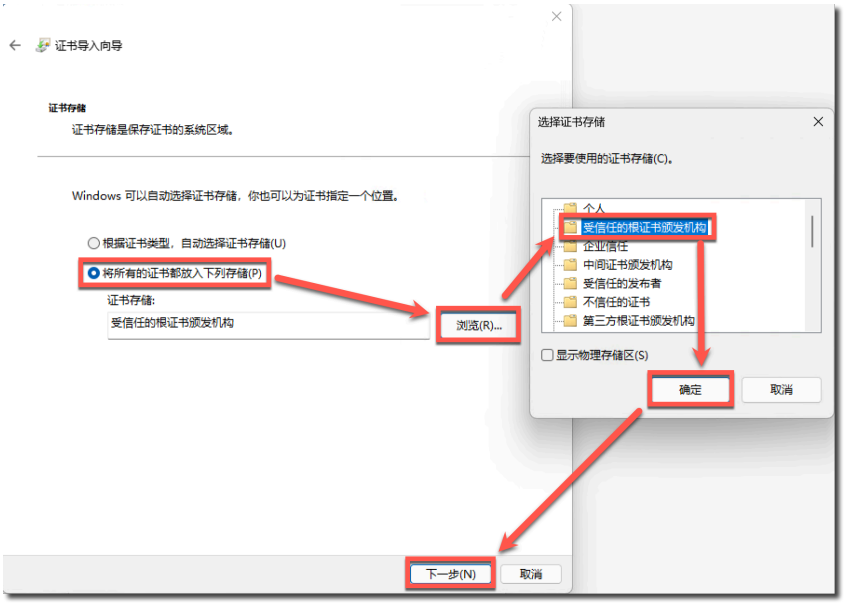
1. 双击下载的 CA 证书，点击“安装证书 (I)...”。



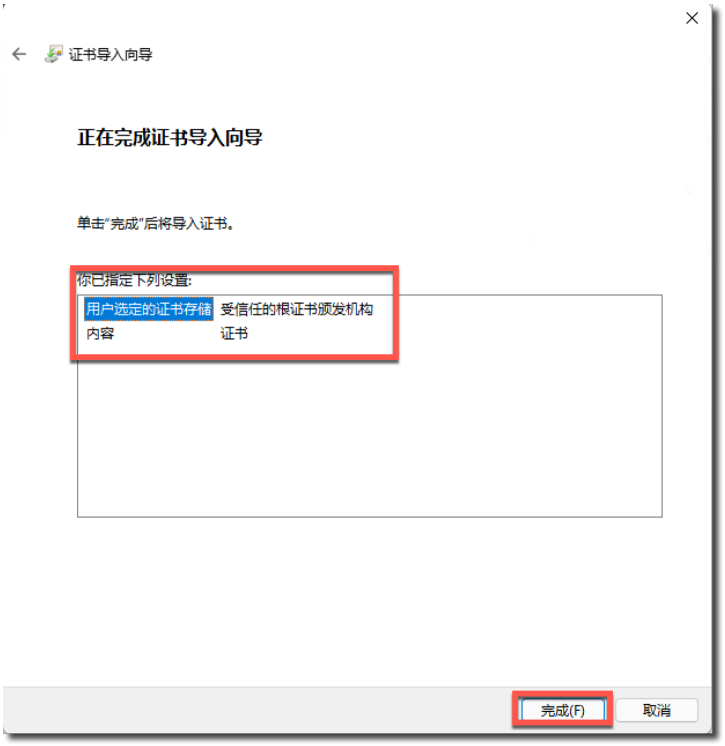
2. 存储位置选择“本地计算机 (L)”，然后点击下一步。



3. 选择“将所有的证书都放入下列存储 (P)”，点击“浏览 (R)...”按钮，选择“受信任的根证书颁发机构”。



4. 完成证书导入。



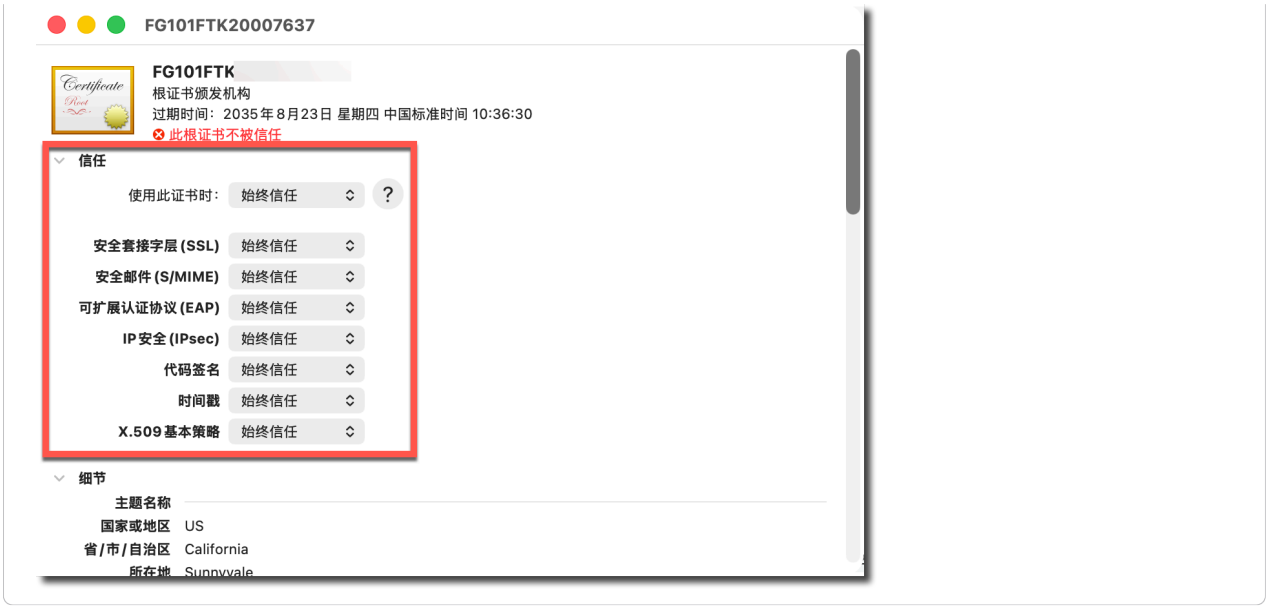
macOS

1. 双击下载的 CA 证书，会打开“钥匙串访问”应用。

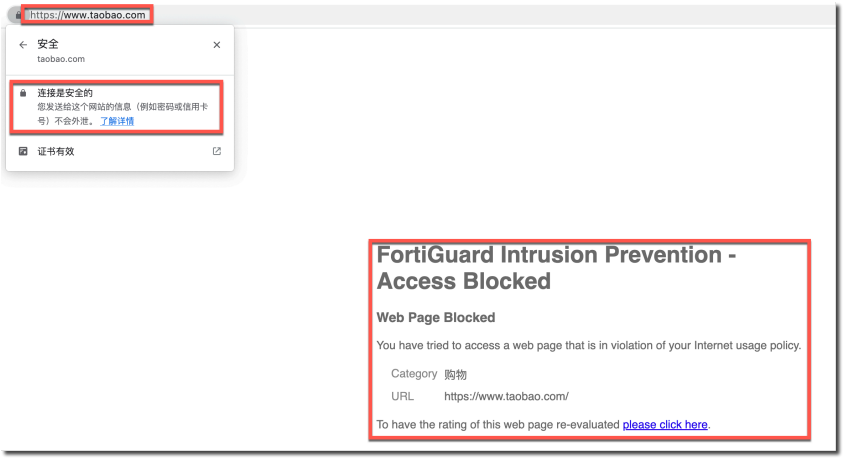
2. 进入“系统 → 证书”页面，可以看到刚刚导入的 CA 证书，然后双击该证书。



3. 在“使用此证书时”选项中选择“始终信任”，关闭窗口（会提示输入系统密码）。



3. 客户端再次访问该网站时，会直接显示出 FortiGate 的阻断页面（或登录页面），不会弹出证书告警页面。



4. 因为浏览器在访问网站时，检查 HSTS 的时候会优先查找自己本地证书的信任关系（Firefox 有自己的信任列表，需要在浏览器设置中添加），只要证书被设备或者浏览器本身信任，那么就不会被阻止，同样的事情，就算网站没有开启 HSTS，因为对 FortiGate 证书信任了后，也不会弹出对应的证书错误提示页面，从而直接访问到 UTM 阻断页面（或用户登录页面）。

8008/8010/8015/8020 端口

问题现象

- 当在防火墙策略中应用 UTM Profiles（例如 Web Filter、AntiVirus、Application Control）后，在进行端口扫描或 PCI 合规检测（如外部安全审计）时，可能会发现 FortiGate 意外开放了一些非业务端口（8008/8010/8015/8020）。
- 即使目标网络内并不存在对应目的 IP，只要探测报文（TCP SYN）能够匹配到带 UTM 的策略，FortiGate 可能会对 8008/8010/8015/8020 端口返回 SYN ACK，使扫描结果显示这些端口为开启状态。
- 即使这些端口并未在防火墙策略中显式放行，扫描工具仍可能检测到端口状态异常，从而导致合规检测失败。

> **相关信息**
>
> 该现象主要体现在端口探测时 FortiGate 的响应行为，导致合规扫描工具/安全基线检查看到“端口开放”。即便扫描结果显示端口为开启状态，流量并不会通过策略在这些端口“泄露/放通”，不会存在安全问题。

问题原因

当防火墙策略启用了 Web Filter/AV 等 UTM 配置文件后，FortiGate 可能会为以下功能开启临时监听内部端口，用于认证信息交互和替换页面内容传输，这些端口并非用于业务流量转发，而是 FortiGate 内部机制所需。

- Authentication Override（当 UTM 动作为“认证”时的认证跳转页面）
- Web Filter Replacement Message（阻断提示页面）

解决方法

如果出于安全合规要求，这些端口不能呈现开启状态，可通过以下配置让 FortiGate 关闭相关端口响应：

```bash
VDOM关闭状态下的命令:
config webfilter fortiguard
    set close-ports enable
end

VDOM开启状态下的命令:
config global
    config webfilter fortiguard
        set close-ports enable
    end
end
```

启用 `close-ports` 的影响

受影响的功能：

- Web Filter 中 FortiGuard 分类的“警告（warning）”和“认证（authenticate）”动作将无法生效。
- 手动“临时放行/覆盖（override）被阻止的分类”功能将失效。
- Web 过滤触发的 替换页面（阻断页面）将不显示 Fortinet 的 Logo。

不受影响的功能：

- Web Filter 中 FortiGuard 分类的“屏蔽（block）”动作仍然有效。
- 本地 URL Filter（白名单/黑名单） 仍然生效。

配置建议

- 对于对合规性（PCI/安全扫描）要求高的生产环境，建议开启。
- 如果依赖用户认证覆盖或警告页面功能，请谨慎评估。

FortiGate Cloud 无法登录

问题描述

1. 某用户环境使用 SD-WAN 部署，SD-WAN 成员中包含 IPSec 隧道（无法访问 Internet）和 ISP 线路（可以访问 Internet），本机默认路由指向 SD-WAN 接口（默认路由在 IPSec 接口和 ISP 线路上负载），FortiGuard 配置中指定 FortiGuard 流量的出接口为 wan1。

```bash
Spoke1 # get router info routing-table all

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.22.6.1, wan2, [1/0]
                  [10/0] via 172.22.6.1, wan1, [1/0]
```

```bash
config system fortiguard
    set interface-select-method specify
    set interface "wan1"
end
```

2. FortiGuard 更新正常，但登录 FortiGate Cloud 时提示错误，无法登录。





问题原因

FortiGate Cloud 登录、记录日志流量的源 IP、线路选择方式在 config log fortiguard setting 下配置，而不是在“config system fortiguard”下。默认配置如下，接口选择方式为自动，参考路由表决定出接口和源 IP，可能选择 IPSec 接口发送，导致 FortiGate Cloud 无法登录或发送日志到 FortiCloud。

```bash
config log fortiguard setting
    set status enable
    set source-ip 0.0.0.0
    set interface-select-method auto
end
```

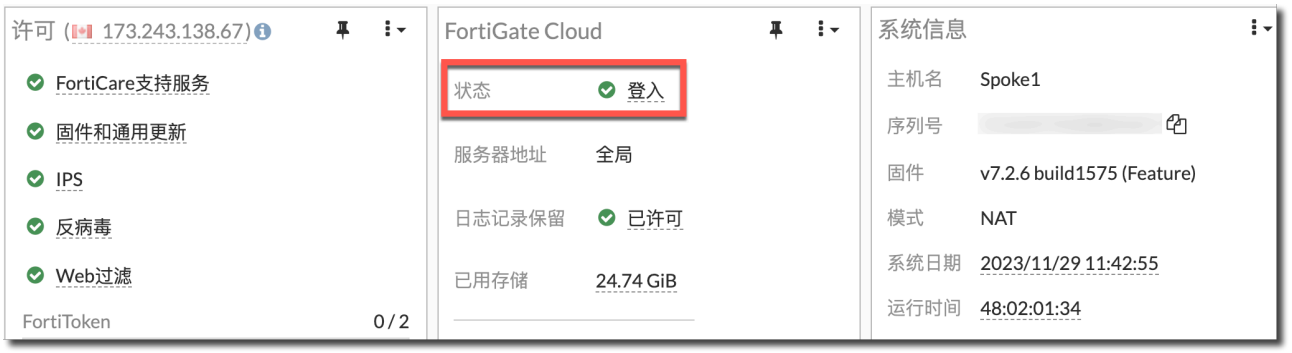
解决方法

在 config log fortiguard setting 下指定 FortiCloud 流量的接口为 ISP 线路接口。

```bash
config log fortiguard setting
    set interface-select-method specify
    set interface "wan1"
end
```

结果验证

FortiGate Cloud 可以正常登录，且可以正常发送日志到 FortiCloud。



获取接口对应的公网 IP

1. 当 FortiGate 的 WAN 接口（配置私网 IP）位于 Internet 设备（针对 FortiGate 的 WAN 接口做 SNAT）后时，可以通过如下 CLI 命令获取 WAN 接口对应的公网 IP。

   重要

   该 CLI 命令需要 FortiGate 可以正常解析并访问 `https://api.ipify.org`。

   ```
   diagnose sys waninfo ipify
   ```

2. 如下所示：

   ```bash
   FortiGate # diagnose sys waninfo ipify
   Try to get my public IP through https://api.ipify.org with src_ip=0.0.0.0 device=unspecified vfid=0(root) ...

   Public/WAN IP: 60.247.121.226
   Location:
           Latitude: 39.904202
           Longitude: 116.407402
           Accuracy radius: Unknown
           Time zone: Asia/Shanghai
   City: Beijing
   Subdivisions:
           0: Beijing
   Country: China
   Postal: Unknown
           Code: Unknown
   Continent: Asia
   Registered country: Unknown
   ISP: Unknown
   ```

3. 如果存在多个 WAN 接口连接，需要获取特定 WAN 接口对应的公网 IP，请使用如下 CLI 命令，指定要获取对应公网 IP 的接口，或使用接口私网 IP 作为源 IP 去获取对应的公网 IP。

   ```bash
   diagnose sys waninfo ipify <interface_name> | <source_IP_address>
   ```

4. 如下所示：

   ```bash
   FortiGate # diagnose sys waninfo ipify wan
   Try to get my public IP through https://api.ipify.org with src_ip=0.0.0.0 device=wan vfid=0(root) ...

   Public/WAN IP: 45.62.167.221
   Location:
           Latitude: 22.278320
           Longitude: 114.174690
           Accuracy radius: Unknown
           Time zone: Asia/Hong_Kong
   City: Hong Kong
   Subdivisions:
           0: Hong Kong
   Country/Zone: Hong Kong
   Postal: Unknown
           Code: Unknown
   Continent: Asia
   Registered country: Unknown
   ISP: Unknown
   ```

   ```bash
   FortiGate # diagnose sys waninfo ipify 192.168.123.175
   Try to get my public IP through https://api.ipify.org with src_ip=192.168.123.175 device=unspecified vfid=0(root) ...

   Public/WAN IP: 123.112.242.161
   Location:
           Latitude: 39.904202
           Longitude: 116.407402
           Accuracy radius: Unknown
           Time zone: Asia/Shanghai
   ```

```
City: Beijing
Subdivisions:
        0: Beijing
Country: China
Postal: Unknown
        Code: Unknown
Continent: Asia
Registered country: Unknown
ISP: Unknown
```

故障排查

常用排错命令

查找对象引用

Ping 工具

会话工具

- 会话管理
- 会话参数分析
- 会话统计命令
- 基于状态过滤会话

Sniffer 工具

- sniffer 工具介绍
- Sniffer 过滤条件的合理使用
- 使用 Wireshark 打开抓包文件
- Wireshark 显示抓包接口和方向
- 通过 GUI 界面抓包
- NP7 芯片抓包

Debug Flow 工具

- Debug Flow 介绍
- Debug Flow 排错举例
- 通过 GUI 进行 Debug Flow

Sniffer/Debug Flow/NP 加速的矛盾

数据包处理流程

- 简介
- Parallel Path Processing
- 无 NP 处理的数据包流程
- 被 NP 加速的流量
- 安全检测模式

  基于流的检测　基于代理的检测　显式 Web 代理

FortiGate 之间测速

进程监控与管理

WAD 进程 Debug

AV/IPS 内存高排查

Kernel CPU 高的定位方法

终端日志记录

信息收集

Debug 信息分析引擎

FortiGate Support Tool

常用排错命令

本文整理 FortiGate 常用 CLI 排错命令，适合在故障定位、信息收集、Ticket 开单、现场割接验证时快速查找。不同 FortiOS 版本、设备型号、VDOM 模式下可用命令可能存在差异，实际执行前可在命令后输入 `?` 确认本机支持的参数。

> **重要**
>
> - 开启实时 debug 前建议先设置过滤条件，避免控制台输出过多日志。
> - `diagnose debug disable` 只停止在控制台打印 debug，后台 debug 可能仍在运行；排查完成后应执行 `diagnose debug reset` 完全停止并清理 debug。
> - `factoryreset`、`reboot`、`format`、`restart`、`clear`、`delete`、`flush` 等命令可能中断业务或清除状态，生产环境执行前需要确认影响范围。

调试开关

本节用于统一控制 FortiGate CLI debug 的开启、停止和输出时长。排查完成后建议执行 `diagnose debug reset`，避免后台 debug 持续运行。

- `diagnose debug reset`：停止并清理此前在前台或后台运行的所有 debug。
- `diagnose debug enable`：开始在控制台打印 debug 输出。
- `diagnose debug disable`：停止控制台 debug 输出，但后台 debug 可能仍在运行，需要配合 `diagnose debug reset` 完全停止。
- `diagnose debug duration 0`：设置 debug 无限时长；默认 debug 时长为 30 分钟。

系统

此类命令主要用于查看 FortiGate 运行状态、进程、硬件、FortiGuard、会话表、网络诊断、抓包和各类本地 debug。

系统状态与进程

- `get system status`：查看系统基本信息。
- `execute time`：查看当前系统时间。
- `get system performance status`：查看 CPU、内存等系统性能状态。
- `execute tac report`：生成用于提交 Fortinet Support Ticket 的 TAC 报告。
- `diagnose sys top {s} {n} {i}`：每隔 `s` 秒显示前 `n` 个进程，循环 `i` 次；在输出界面中可用 `Shift + C` 按 CPU 排序，用 `Shift + M` 按内存排序。
- `diagnose debug crashlog read`：查看系统或应用崩溃记录。
- `diagnose sys process pidof <daemon>`：查看指定守护进程的 PID，例如 `diagnose sys process pidof httpsd`；守护进程名称可通过 `diagnose sys top` 查看。
- `diagnose sys kill 11 <pid>`：使用 signal 11 终止指定 PID。
- `diagnose sys session stat`：查看会话统计信息。
- `diagnose sys session exp-stat`：查看 expectation session 统计信息。
- `diagnose sys vd list`：查看 VDOM 信息及系统统计。
- `diagnose sys cmdb info`：查看最近一次由守护进程执行的配置变更信息。
- `execute factoryreset [keepvmlicense]`：立即恢复出厂设置并重启；VM 型号可通过 `keepvmlicense` 保留 VM License。
- `execute factoryreset-shutdown [keepvmlicense]`：立即恢复出厂设置并关机；VM 型号可通过 `keepvmlicense` 保留 VM License。
- `execute factoryreset2 [keepvmlicense]`：恢复大部分配置，但保留系统设置、系统接口、VDOM、静态路由和虚拟交换机；VM 型号可通过 `keepvmlicense` 保留 VM License。
- `diagnose debug config-error-log read`：查看配置文件中的错误。
- `diagnose snmp ip frags`：查看 IP 分片与重组信息。
- `diagnose sys process dump <PID>`、`diagnose sys process pstack <PID>`、`diagnose sys process trace <PID>`：查看指定进程的 dump、调用栈、trace 等进程相关信息。
- `diagnose sys mpstat {n}`：每隔 `n` 秒查看 CPU 使用率。
- `diagnose hardware sysinfo memory`：查看系统内存信息。
- `diagnose firewall packet distribution`：查看数据包分布统计。
- `execute reboot`：重启设备。

硬件

- `diagnose hardware sysinfo interrupts`：查看 CPU 中断统计。
- `diagnose hardware test suite all`：执行硬件诊断测试，也就是 HQIP 测试。
- `diagnose hardware deviceinfo disk`：查看磁盘信息。
- `diagnose sys flash list`：查看 Flash 分区。
- `execute disk list`：查看已挂载磁盘。
- `execute disk format <partition ref>`：格式化指定分区。

- `diagnose disktest device <device>`、`diagnose disktest block <block>`、`diagnose disktest size <mb>`、`diagnose disk test run`：配置并执行磁盘读写测试，用于检查磁盘是否异常；`<device>` 为待测试设备，`<block>` 为每次读写块大小，`<mb>` 为每轮测试大小上限。
- `execute formatlogdisk`：格式化日志磁盘。
- `diagnose hardware sysinfo cpu`：查看 CPU 信息。
- `diagnose sys modem detect`、`diagnose debug application modemd -1`、`diagnose debug enable`：检测 Modem，并开启 Modem 守护进程实时 debug。

FortiGuard

- `diagnose webfilter fortiguard statistics`：查看 Web Filter Rating 缓存与守护进程统计。
- `diagnose debug rating`：查看 Web Filter Rating 服务器信息。
- `diagnose debug application update -1`、`diagnose debug enable`：开启 update 守护进程 debug，用于排查 FortiGuard 更新问题。
- `execute update-now`：手动触发 FortiGuard 更新。
- `diagnose autoupdate status`、`diagnose autoupdate versions`：查看 License 与自动更新版本信息。

会话表

- `diagnose sys session filter <filter>`：设置会话表过滤条件。
- `diagnose sys session filter`：查看当前会话过滤条件。
- `diagnose sys session list`：按过滤条件查看会话表。
- `diagnose sys session clear`：清除符合当前过滤条件的会话。
- `diagnose firewall iprope list`：查看 FortiGate 内部防火墙表。

网络诊断

- `execute ping-options {options}`、`execute ping <x.x.x.x>`：按指定选项对 IPv4 地址执行 Ping 测试。
- `execute ssh-options {options}`、`execute ssh <x.x.x.x>`：按指定选项从 FortiGate 发起 SSH 连接。
- `execute traceroute-options {options}`、`execute traceroute <x.x.x.x>`：按指定选项执行 Traceroute。
- `get system arp`、`diagnose ip arp list`：查看 ARP 表项。
- `diagnose netlink brctl list`：查看 FortiGate 上所有 switch 名称。
- `diagnose netlink brctl name host <switch-name>`：查看指定 switch 的二层转发表。
- `get system interface`、`get sys interface physical`：查看接口概要信息，包括接口 IP 信息。
- `diagnose ip address list`：查看 IP 地址信息。
- `diagnose hardware deviceinfo nic <interface>`、`get hardware nic <interface>`：查看指定接口的详细硬件/链路信息。
- `get sys interface transceiver`：查看已连接光模块信息。

抓包

- `diagnose sniffer packet <interface> <'filter'> <verbose> <count> <a|l>`：使用内置 sniffer 在指定接口按过滤条件抓包，并指定详细级别、抓包数量、ASCII/链路层输出方式。

Debug Flow

- `diagnose debug reset`：停止并清理此前运行的 debug。
- `diagnose debug flow filter clear`：清除 IPv4 debug flow 过滤条件。
- `diagnose debug flow filter6 clear`：清除 IPv6 debug flow 过滤条件。
- `diagnose debug flow filter <filter>`：设置 IPv4 debug flow 过滤条件。
- `diagnose debug flow filter6 <filter>`：设置 IPv6 debug flow 过滤条件。
- `diagnose debug flow show function-name enable`：在 debug flow 输出中显示代码函数名。
- `diagnose debug flow show iprope enable`：在 debug flow 输出中显示流量经过的内部防火墙策略信息。
- `diagnose debug console timestamp enable`：在 debug 输出中显示时间戳。
- `diagnose debug flow trace start <n>`：输出 `n` 行 IPv4 debug flow。
- `diagnose debug flow trace start6 <n>`：输出 `n` 行 IPv6 debug flow。
- `diagnose debug enable`：开始在控制台打印 debug 输出。

UTM

- `diagnose debug urlfilter <filter>`、`diagnose debug application urlfilter -1`、`diagnose debug enable`：开启 Web Filter 实时 debug。
- `diagnose debug enable`、`diagnose test application urlfilter`：列出 Web Filter debug 输出选项。
- `diagnose test application urlfilter <option>`：查看指定 Web Filter debug 输出。
- `diagnose debug application dnsproxy -1`、`diagnose debug enable`：开启 DNS Proxy 实时 debug；DNS Proxy 负责 DNS Filter、DNS Translation、DNS Resolution 等功能。

- `diagnose debug enable`、`diagnose test application dnsproxy`：列出 DNS Proxy debug 输出选项。
- `diagnose test application dnsproxy <option>`：查看指定 DNS Proxy debug 输出。
- `diagnose ips filter set "host <x.x.x.x> and port <port>"`、`diagnose ips debug enable all`、`diagnose debug enable`：按主机和端口过滤，开启 Application Control/IPS 安全配置文件相关的 IPS Engine debug。
- `diagnose ips debug enable av`、`diagnose ips debug status show`、`diagnose sys scanunit debug all enable`、`diagnose sys scanunit debug level verbose`、`diagnose sys scanunit debug show`、`diagnose debug enable`：在防病毒配置文件使用 Flow 模式时，开启 AV 实时 debug。
- `diagnose wad debug enable category scan`、`diagnose wad stream-scan av-test "debug enable"`、`diagnose wad stream-scan av-test "debug all:debug"`、`diagnose sys scanunit debug all enable`、`diagnose sys scanunit debug level verbose`、`diagnose sys scanunit debug show`、`diagnose debug enable`：在防病毒配置文件使用 Proxy 模式时，开启 AV 实时 debug。

IPS Engine

- `diagnose test application ipsmonitor 1`：查看 IPS Engine 信息。
- `diagnose test application ipsmonitor 2`：设置 IPS Engine 启用/禁用状态。
- `diagnose test application ipsmonitor 99`：重启所有 IPS Engine 与 monitor。
- `diagnose test application ipsmonitor 97`：启动所有 IPS Engine。
- `diagnose test application ipsmonitor 98`：停止所有 IPS Engine。
- `diagnose ips session list`、`diagnose test application ipsmonitor 13`：查看每个 IPS Engine 内存空间中的 IPS Session。
- `diagnose ips filter set "host <x.x.x.x> and port <port>"`、`diagnose ips debug enable all`、`diagnose debug enable`：查看符合过滤条件流量的 IPS Engine debug。

WAD

- `diagnose test application wad 1000`：查看所有 WAD 进程。
- `diagnose test application wad 2`：查看 WAD 总内存使用量。
- `diagnose test application wad 99`：重启所有 WAD 进程。
- `diagnose wad debug display pid enable`、`diagnose wad filter <filter>`、`diagnose wad filter list`、`diagnose wad debug enable level <level>`、`diagnose wad debug enable category <category>`、`diagnose debug enable`：开启由 WAD 处理流量的实时 debug。
- `diagnose wad filter <filter>`：设置 WAD debug 过滤条件。
- `diagnose wad filter list`：查看已设置的 WAD debug 过滤条件。
- `diagnose wad filter clear`：清除 WAD debug 过滤条件。
- `diagnose wad debug enable level <level>`：设置 WAD debug 详细级别。
- `diagnose wad debug enable category <category>`：设置 WAD debug 流量类别。
- `diagnose wad debug display pid enable`：在 debug 中显示处理该会话请求的 WAD worker PID。
- `diagnose debug enable`：开始在控制台打印 debug 输出。

CPU Profiling

- `diagnose sys profile cpumask <cpu_id>`：设置需要进行 CPU profiling 的 CPU Core。
- `diagnose sys profile start`：开始 CPU profiling，通常等待 1 到 2 分钟后停止。
- `diagnose sys profile stop`：停止 CPU profiling。
- `diagnose sys profile module`：查看已应用的 Kernel Module。
- `diagnose sys profile show detail`、`diagnose sys profile show order`：查看对应 CPU Core 的 profiling 结果。

Tree

- `tree`：查看完整 CLI 命令树。
- `tree execute`：查看 `execute` 命令树。
- `tree diagnose`：查看 `diagnose` 命令树。

路由

此类命令主要用于查看 IPv4/IPv6 路由表、动态路由协议、组播路由、SD-WAN 规则与链路监控状态。

IPv4/IPv6 路由

- `get router info routing-table all`：查看路由表。
- `get router info routing-table database`、`get router info6 routing-table database`：查看 IPv4/IPv6 路由数据库信息。
- `diagnose ip route list`、`get router info kernel`、`diagnose ipv6 route list`、`get router info6 kernel`：查看 IPv4/IPv6 Kernel 路由表。
- `get router info protocols`、`get router info6 protocols`：查看 IPv4/IPv6 路由协议状态。

- `execute router restart`：重启路由守护进程。
- `get router info bfd neighbor`、`get router info6 bfd neighbor`：查看 IPv4/IPv6 BFD 邻居。
- `diagnose test application bfd 1`、`diagnose test application bfd 2`、`diagnose test application bfd 3`：查看 BFD 统计信息。
- `diagnose debug application bfdd <debug level>`、`diagnose debug enable`：开启 BFD 实时 debug。
- `get router info bgp summary`、`get router info6 bgp summary`：查看 IPv4/IPv6 BGP Summary。
- `get router info bgp neighbors`、`get router info6 bgp neighbors`、`get router info bgp neighbors <x.x.x.x> advertised-routes`、`get router info6 bgp neighbors <x:x::x:x/m> advertised-routes`、`get router info bgp neighbors <x.x.x.x> received-routes`、`get router info6 bgp neighbors <x:x::x:x/m> received-routes`、`get router info bgp neighbors <x.x.x.x> routes`、`get router info6 bgp neighbors <x:x::x:x/m> routes`：查看 BGP 邻居、宣告路由、接收路由和邻居路由；`<x.x.x.x>` 为 IPv4 Peer 地址，`<x:x::x:x/m>` 为 IPv6 Peer 地址。
- `diagnose ip router bgp all enable`、`diagnose ip router bgp level info`、`diagnose debug enable`：开启 BGP 实时 debug。
- `execute router clear bgp {all | as <ASN> | ip x.x.x.x | ipv6 y:y:y:y:y:y:y:y}`：按指定条件执行 BGP Hard Reset；`all` 表示所有 BGP Peer，`as <ASN>` 表示指定 AS 的 Peer，`ip x.x.x.x` 表示指定 IPv4 Peer，`ipv6 y:y:y:y:y:y:y:y` 表示指定 IPv6 Peer。
- `execute router clear bgp {all | ip x.x.x.x | ipv6 y:y:y:y:y:y:y:y} soft {in|out}`：按指定条件执行 BGP Soft Reset；`in` 表示仅重置接收路由，`out` 表示仅重置宣告路由，未指定 `in` / `out` 时两个方向都会执行 Soft Reset。
- `get router info ospf status`、`get router info6 ospf status`：查看 IPv4/IPv6 OSPF 状态。
- `get router info ospf interface`、`get router info6 ospf interface`：查看 IPv4/IPv6 OSPF 接口状态。
- `get router info ospf neighbor all`、`get router info6 ospf neighbor all`：查看 IPv4/IPv6 OSPF 邻居信息。
- `get router info ospf database brief`、`get router info6 ospf database brief`：查看 IPv4/IPv6 OSPF 数据库简要信息。
- `diagnose ip router ospf all enable`、`diagnose ip router ospf level info`、`diagnose debug enable`：开启 OSPF 实时 debug。

组播路由

- `get router info multicast igmp interface`：查看接口 IGMP 统计。
- `get router info multicast igmp groups`：查看通过 IGMP 订阅的组播组。
- `diagnose ip multicast get-igmp-limit`：查看 IGMP 状态数量上限。
- `diagnose ip router igmp decode enable`、`diagnose ip router igmp level info`、`diagnose debug console timestamp enable`、`diagnose debug enable`：开启 IGMP 守护进程实时 debug。
- `execute mrouter clear igmp-interface <interface>`：清除指定接口上的所有 IGMP 条目。
- `execute mrouter clear igmp-group <group-address>`：清除指定组或所有组的 IGMP 条目。
- `get router info multicast pim sparse-mode interface`：查看 PIM Sparse Mode 接口信息。
- `get router info multicast pim sparse-mode neighbor`：查看 PIM Sparse Mode 邻居信息。
- `get router info multicast pim sparse-mode rp-mapping`：查看 RP 到组播组的映射信息。
- `get router info multicast pim sparse-mode table`：查看 PIM Sparse Mode 路由表。
- `diagnose ip router pim-sm events enable`、`diagnose ip router pim-sm all enable`、`diagnose ip router pim-sm level info`、`diagnose debug enable`：开启 PIM Sparse Mode 实时 debug。

SD-WAN

- `diagnose sys sdwan health-check status`：查看 SD-WAN Health Check 统计。
- `diagnose sys sdwan service4`、`diagnose sys sdwan service6`：查看控制平面的 IPv4/IPv6 SD-WAN 规则。
- `diagnose sys sdwan member`：查看 SD-WAN 成员。
- `diagnose firewall proute list`：查看数据平面的 SD-WAN 规则与 Policy Route。
- `diagnose sys link-monitor status`、`diagnose sys link-monitor interface <interface>`：查看 Link Monitor 统计。
- `diagnose debug application link-monitor -1`、`diagnose debug enable`：开启 Link Monitor 实时 debug。
- `diagnose test application lnkmtd 1`、`diagnose test application lnkmtd 2`、`diagnose test application lnkmtd 3`：查看 Link Monitor 统计。

认证

认证类命令主要用于查看认证用户、测试认证服务器、排查 LDAP/RADIUS/FSSO/SAML 等认证问题。

- `diagnose firewall auth filter <filter>`：设置认证用户列表过滤条件。
- `diagnose firewall auth list`：列出过滤后的已认证 IPv4 用户。
- `diagnose wad user list`：列出当前由 Proxy/WAD 认证的用户。
- `diagnose debug application fnbamd -1`、`diagnose debug application authd -1`、`diagnose debug enable`：开启远程认证与本地认证实时 debug。
- `diagnose test authserver <auth_protocol> <server_name> <user> <password>`：直接从 CLI 测试认证；密码会以明文显示，记录日志时需要注意。
- `diagnose test authserver ldap <server_name> <user> <password>`：测试 LDAP 用户认证；密码会以明文显示。
- `diagnose test authserver radius <server_name> <auth_type> <user> <password>`：测试 RADIUS 用户认证；密码会以明文显示。
- `diagnose debug fsso-polling detail`、`diagnose debug fsso-polling summary`：查看 FortiGate 到 DC 的 FSSO Polling 信息。
- `diagnose debug fsso-polling user`、`diagnose debug authd fsso list`：查看 FortiGate 轮询 DC 获取到的 FSSO 登录用户。

- `diagnose debug application fssod -1`、`diagnose debug application smbcd -1`、`diagnose debug enable`：FortiGate 用作 FSSO Polling 设备时，开启相关实时 debug。
- `diagnose debug fsso-polling refresh-user <id>`、`execute fsso refresh`：刷新当前 FSSO 登录用户列表；该操作可能影响业务，需谨慎使用。
- `diagnose debug authd fsso server-status`：查看 FortiGate 与 Collector Agent 的连接状态。
- `diagnose debug application authd 8256`、`diagnose debug enable`：开启 FortiGate 与 Collector Agent 连接相关的实时 debug。
- `diagnose debug authd fsso refresh-logons`：让 Collector Agent 重新发送登录用户列表到 FortiGate。
- `diagnose debug application samld -1`、`diagnose debug enable`：开启 SAML 实时 debug。

VPN

此类命令主要用于查看、清除和调试 IPsec Phase 1/Phase 2、IKE 进程与加密计数器。

IPSec

- `diagnose vpn ike gateway list`：查看 IPsec Phase 1 信息。
- `diagnose vpn tunnel list`：查看 IPsec Phase 2 信息。
- `get vpn ipsec tunnel summary`、`get vpn ipsec tunnel details`：查看 IPsec 隧道概要与详细信息。
- `diagnose vpn tunnel flush`：清除所有 Phase 2 Tunnel SA。
- `diagnose vpn tunnel flush <name> [name]`：按名称清除一个或多个指定 Phase 2 Tunnel。
- `diagnose vpn ike gateway <clear | flush>`：清除/Flush IKE Gateway（Phase 1）；可先用 `diagnose vpn ike gateway filter` 设置过滤条件。
- `diagnose vpn ike gateway <clear | flush> name <name>`：按名称清除/Flush 指定 IKE Gateway（Phase 1）。
- `diagnose vpn ike gateway filter`：设置或查看 IKE Gateway 过滤条件。
- `diagnose vpn ipsec status`：查看 IPsec 加密计数器等信息。
- `diagnose vpn ike log filter <filter>`：设置 IKE 守护进程 debug 过滤条件。
- `diagnose debug application ike -1`、`diagnose debug enable`：按过滤条件开启 IKE 守护进程实时 debug。
- `diagnose vpn ike restart`：重启 IKE 进程。
- `diagnose vpn ike counts`、`diagnose vpn ike routes`、`diagnose vpn ike errors`、`diagnose vpn ike stats`、`diagnose vpn ike status`、`diagnose vpn ike crypto`：查看 IKE 计数、路由、错误、统计、状态、加密相关信息。

受管设备

受管设备类命令主要用于查看受 FortiGate 管理的 FortiSwitch 与 FortiAP 状态、连接、端口、无线客户端和相关 debug。

FortiSwitch

- `diagnose switch-controller switch-info mac-table`：查看受管 FortiSwitch MAC 地址表。
- `diagnose switch-controller switch-info port-stats`：查看受管 FortiSwitch 端口统计。
- `diagnose switch-controller switch-info trunk status`：查看受管 FortiSwitch Trunk 信息。
- `diagnose switch-controller switch-info mclag`：查看 FortiSwitch MCLAG 相关信息。
- `diagnose switch-controller switch-info poe`：查看 PoE 相关信息。
- `diagnose switch-controller switch-info lldp`：查看 LLDP 相关信息。
- `diagnose switch-controller switch-info port-properties`：查看受管 FortiSwitch 端口属性。
- `diagnose switch-controller switch-info acl-counters`：查看受管 FortiSwitch 端口 ACL 计数器。
- `diagnose switch-controller switch-info pdu-counters-list`：查看受管 FortiSwitch PDU 计数器。
- `diagnose switch-controller switch-info flapguard`：查看受管 FortiSwitch Flap Guard 信息。
- `diagnose switch-controller switch-info qos-stats`：查看受管 FortiSwitch QoS 统计。
- `diagnose switch-controller switch-info modules`：查看 FortiSwitch 模块相关信息。
- `diagnose switch-controller switch-info stp`：查看受管 FortiSwitch STP 实例状态。
- `diagnose switch-controller switch-info bpdu-guard-status`：查看受管 FortiSwitch STP BPDU Guard 状态。
- `diagnose switch-controller switch-info igmp-snooping`：查看受管 FortiSwitch IGMP Snooping 信息。
- `diagnose switch-controller switch-info loop-guard`：查看受管 FortiSwitch Loop Guard 状态。
- `diagnose switch-controller switch-info dhcp-snooping`：查看受管 FortiSwitch DHCP Snooping 接口列表。
- `diagnose switch-controller switch-info arp-inspection`：查看受管 FortiSwitch ARP Inspection 接口列表。
- `diagnose switch-controller switch-info option82-mapping`：查看受管 FortiSwitch DHCP Option 82 Mapping 信息。
- `diagnose switch-controller switch-info 802.1X`：查看受管 FortiSwitch 端口 802.1X 状态。
- `diagnose switch-controller switch-info 802.1X-dacl`：查看受管 FortiSwitch 端口 802.1X Dynamic ACL 状态。
- `diagnose switch-controller switch-info mac-limit-violations`：查看受管 FortiSwitch 违反 MAC Limit 的 MAC 信息。
- `diagnose switch-controller switch-info flow-tracking`：查看受管 FortiSwitch Flow 信息。
- `diagnose switch-controller switch-info mirror`：查看受管 FortiSwitch Mirror 信息。
- `diagnose switch-controller switch-info ip-source-guard`：查看受管 FortiSwitch 硬件中的 IP Source Guard 信息。

- `diagnose switch-controller switch-info rpvst`：查看与 Rapid PVST 网络互通时的 FortiSwitch STP 端口信息。
- `execute switch-controller get-conn-status <fortiswitch>`：查看 FortiSwitch 连接状态。
- `execute switch-controller get-physical-conn standard <fortilink-name>`：查看 FortiLink 物理连接拓扑图。
- `execute switch-controller diagnose-connection <fortiswitch>`：查看 FortiSwitch 连接诊断信息。

FortiAP

- `diagnose wireless-controller wlac -c wtp`、`diagnose wireless-controller wlac -d wtp`：查看 FortiAP 设备信息。
- `diagnose wireless-controller wlac -c sta`、`diagnose wireless-controller wlac -d sta`：查看连接到 FortiAP 的无线客户端信息。
- `diagnose wireless-controller wlac help`：查看 Wireless Controller 可用 debug 选项。
- `diagnose wireless-controller wlac sta_filter`、`diagnose wireless-controller wlac sta_filter clear`、`diagnose wireless-controller wlac sta_filter <aa:bb:cc:dd:ee:ff> 255`、`diagnose debug enable`：按无线客户端 MAC 地址开启实时 debug；`<aa:bb:cc:dd:ee:ff>` 为终端/Station MAC 地址。
- `diagnose wireless-controller wlac -c vap`：查看 VAP 信息，包括 MAC、BSSID、SSID、接口名以及正在广播该 VAP 的 AP IP 地址。
- `diagnose wireless-controller wlac wtp_filter`、`diagnose wireless-controller wlac wtp_filter clear`、`diagnose wireless-controller wlac wtp_filter <FAP-SN> 0-<x.x.x.x>:5246 255`、`diagnose debug application cw_acd 0x7ff`：当 FortiAP 无法连接 FortiGate 时，在 Wireless Controller 上查看 WTP/FortiAP debug；`<FAP-SN>` 为 FortiAP 序列号，`<x.x.x.x>` 为 FortiAP IP 地址。

其他服务

此类命令覆盖 HA、ZTNA、日志、流量整形和 SIP 相关排查。

HA

- `diagnose sys ha status`、`get system ha status`：查看 HA 状态与信息。
- `execute ha manage <index> <username>`：登录并管理指定 HA 成员。
- `diagnose sys ha checksum cluster`：查看所有集群成员的 checksum 信息。
- `diagnose sys ha checksum show <vdom>`：查看指定 VDOM 的详细 checksum 信息。
- `diagnose sys ha checksum recalculate`：重新计算 HA checksum。
- `diagnose sys ha recalculate-extfile-signature`：重新计算 HA 外部文件签名。
- `diagnose sys ha reset-uptime`：重置 HA uptime，通常用于测试故障切换。
- `diagnose debug application hatalk -1`、`diagnose debug application hasync -1`、`diagnose debug application harelay -1`、`diagnose debug enable`：开启 HA 守护进程实时 debug。
- `diagnose sys ha history read`：查看 HA 历史记录。
- `execute ha synchronize stop`、`execute ha synchronize start`：手动停止或启动 HA 同步。

ZTNA

- `diagnose endpoint fctems test-connectivity <EMS>`：验证 FortiGate 到 FortiClient EMS 的连通性。
- `execute fctems verify <EMS>`：验证 FortiClient EMS 证书。
- `diagnose test application fcnacd 2`：Dump EMS 连通性信息。
- `diagnose debug app fcnacd -1`、`diagnose debug enable`：运行 FortiClient NAC 守护进程实时 debug。
- `diagnose endpoint ec-shm list <ip> <mac> <EMS_serial_number> <EMS_tenant_id>`：查看 Endpoint 记录列表，可按 IP、MAC、EMS 序列号、EMS Tenant ID 过滤。
- `diagnose endpoint lls-comm send ztna find-uid <uid> <EMS_serial_number> <EMS_tenant_id>`：按 Client UID、EMS 序列号、EMS Tenant ID 查询 Endpoint。
- `diagnose endpoint lls-comm send ztna find-ip-vdom <ip> <vdom>`：按 Client IP 与 VDOM 查询 Endpoint。
- `diagnose wad dev query-by uid <uid> <EMS_serial_number> <EMS_tenant_id>`：通过 WAD diagnose 命令按 UID、EMS 序列号、EMS Tenant ID 查询。
- `diagnose wad dev query-by ipv4 <ip>`：通过 WAD diagnose 命令按 IPv4 地址查询。
- `diagnose firewall dynamic list`：列出 EMS Security Posture Tag，以及所有动态 IP/MAC 地址。
- `diagnose test application fcnacd 7`、`diagnose test application fcnacd 8`：检查 FortiClient NAC 守护进程中的 ZTNA 与路由缓存。
- `diagnose wad worker policy list`：显示 Application Gateway 规则相关统计。
- `diagnose wad debug enable category all`、`diagnose wad debug enable level verbose`、`diagnose debug enable`：运行 WAD 实时 debug。
- `diagnose debug reset`：ZTNA/WAD debug 完成后清理 debug。

日志

- `diagnose log test`：生成测试日志。
- `execute log filter <filter>`：设置日志过滤条件。
- `execute log filter dump`：查看日志过滤条件。

- `exec log display`：显示过滤后的日志。
- `execute log delete`：删除过滤后的日志。
- `diagnose debug application miglogd -1`、`diagnose debug enable`：开启日志进程 `miglogd` 实时 debug。
- `execute log fortianalyzer test-connectivity`：测试 FortiGate 与 FortiAnalyzer 的连通性。

Traffic Shaping（流量整形）

- `diagnose firewall shaper traffic-shaper list`：查看已配置的 Traffic Shaper。
- `diagnose firewall shaper traffic-shaper stats list`：查看 Traffic Shaper 统计。

SIP Session Helper

- `diagnose sys sip status`：查看 SIP 状态。
- `diagnose sys sip mapping list`：查看 SIP Mapping 列表。
- `diagnose sys sip dialog list`：查看 SIP Dialog 列表。
- `diagnose debug application sip -1`、`diagnose debug enable`：开启 SIP 实时 debug。

SIP ALG

- `diagnose sys sip-proxy calls list`：查看活动 SIP Proxy Call 列表。
- `diagnose sys sip-proxy stats`：查看 SIP Proxy 统计。
- `diagnose sys sip-proxy session list`：查看 SIP Proxy Session 列表。
- `diagnose debug application sip -1`、`diagnose debug enable`：开启 SIP 实时 debug。

查找对象引用

网络需求

在删除一些配置对象前，通常需要先解除该对象的引用关系，如果对象被其他配置被引用，那么通常无法删除它们或将它们分配给某些新的配置，例如：

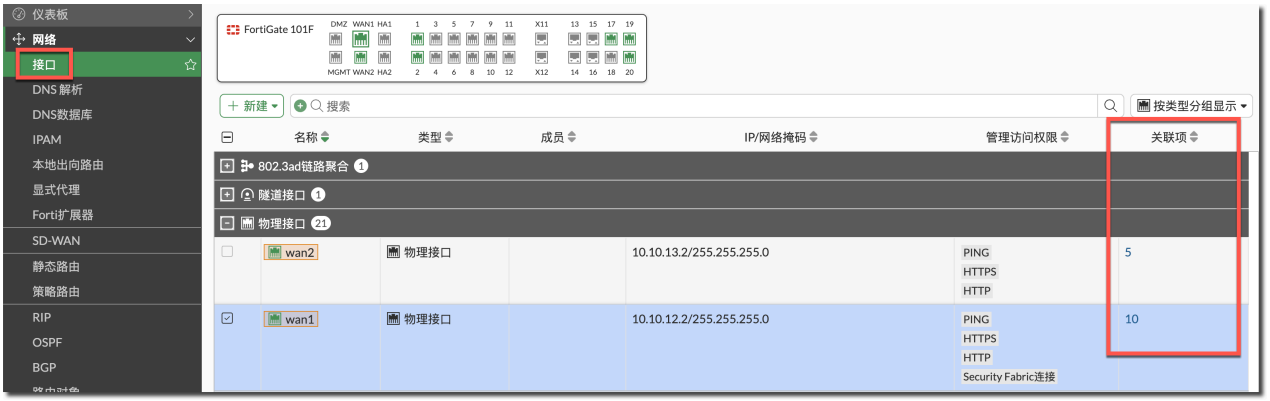
- 在 GUI 下将接口 internal1 添加到一个新的 Hardware/Software 交换接口时，新的交换接口成员列表中可能不显示该接口。从 CLI 执行此操作时也会因“Object is in use”错误而中断，这是因为该接口已被一些其他配置引用。例如配置了静态 IP 地址、配置了 DHCP 服务器、设置了静态路由等，这些都与该接口相关联。
- 尝试删除一个未使用的地址对象，但无法执行此操作。这是因为该地址对象可能被防火墙策略、地址组等配置引用。

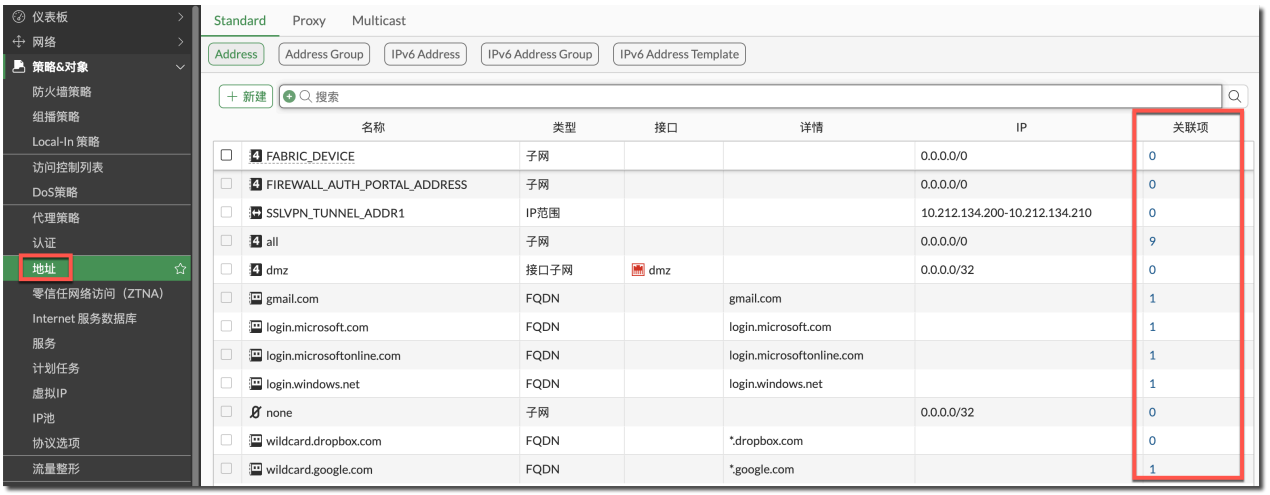
本文主要介绍如何快速找到并解除这些引用关系。

GUI

方法 1

1. 在某个功能配置页面的列表上方，可以看到“关联项”列。







2. 如果“关联项”列不可见，点击左侧的齿轮按钮（或右键单机标题列），然后选中“关联项”。



3. 点击“关联项”下的数字。



4. 弹出“使用率”页面，可以看到该对象当前被哪些配置引用，在该页面可以直接修改/删除引用该对象的配置，如下所示，点击查询一个 IP 池的关联项，有 2 条防火墙策略引用了这个 IP 池。



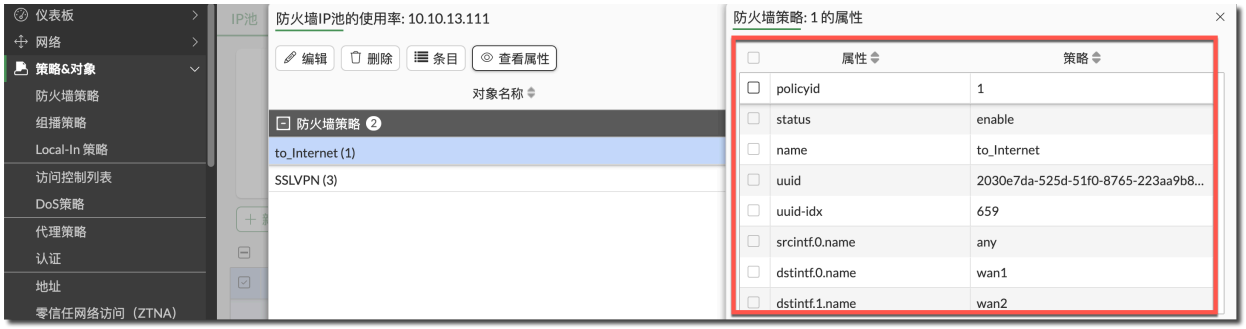
- 编辑：选中某个配置并点击编辑按钮，可以在当前页面编辑引用该对象（防火墙策略）的配置。



- 删除：直接删除引用该对象的配置。
- 条目：跳转到选中的配置所在页面。

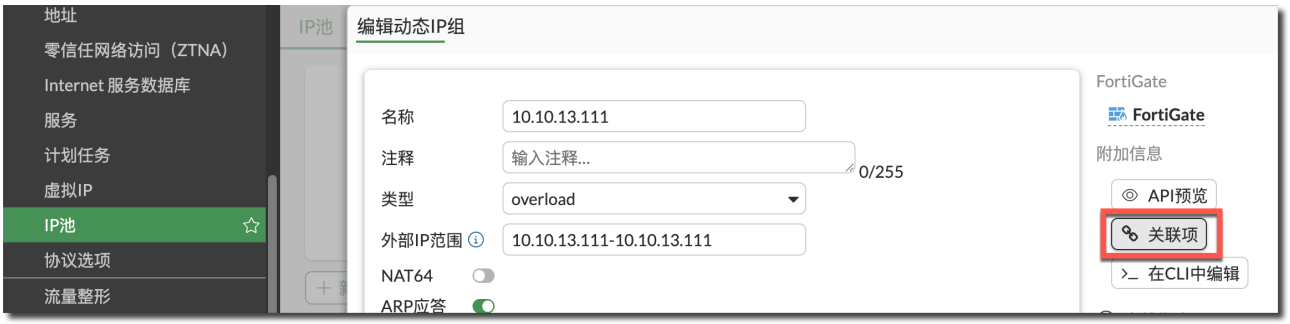


- 查看属性：查看对象被引用的配置的简要信息。



方法 2

编辑某个配置对象，页面右侧点击“关联项”按钮，也可以弹出“使用率”页面。



CLI

主要方法

FortiOS 的 CLI 采用树状语法定义配置对象的位置，例如防火墙地址对象的路径为 `Firewall/Address/<name>`，进入该配置的 CLI 语法为 `config firewall address <name>`。

通过如下命令可以查找对象的配置关联项。

```bash
diagnose sys cmdb refcnt show <path.object.mkey>
```

`<path.object.mkey>` 是对象存储位置、对象类别、对象数据库主键的组合。如下所示，该命令显示 `config system interface` 下名称为“wan2”的接口的所有配置引用，接口“wan2”已被 firewall sniffer、ACME 以及 SD-WAN member 配置引用。

```bash
FortiGate # diag sys cmdb refcnt show system.interface.name wan2

entry used by child table members:seq-num '2' of complex system.sdwan:members.interface
entry used by table firewall.sniffer:id '1'
entry used by child table interface:interface-name 'wan2' of complex system.acme:interface.interface-name
```

定位对象的 mkey

上述命令要求使用 `<path.object.mkey>` 格式指定需要查找关联项的对象。虽然可以很容易找到 path 和 object 的字段（例如，地址对象的 `path.object` 为 `firewall.address`），但要确定该对象使用的具体 mkey 字段需要用到如下方法（mkey 并不总是 `name`）。

方法 1 - 使用 tree 命令

1. 如下所示的 SD-WAN 成员配置，该 SD-WAN 区域的成员序号 1 为 wan1。

```bash
config system sdwan
    config members
        edit 1    ←——SD-WAN成员序号
            set interface "wan1"
            set zone "virtual-wan-link"
        next
    end
end
```

2. 使用 tree 命令查看，SD-WAN 功能的配置路径为 `system.sdwan`，对象为 `members`，而 mkey 为 `seq-num`（被 `*` 标记的为 mkey），而不是接口名称（SD-WAN 成员在配置中是通过其成员 ID 编号而非接口名称进行引用的）。

```bash
FortiGate # tree system sdwan members
-- [members] --*seq-num (0,512)    ←——“*”标记表示该对象的mkey
├─ interface (15)
├─ zone (35)
├─ gateway
......
```

3. 已经知道了 SD-WAN 成员的 mkey `seq-num` 在配置（`config system sdwan` → `config members`）中为“1”，使用 `diagnose sys cmdb refcnt show system.sdwan.members.seq-num 1` 命令就可以查找所有引用 SD-WAN 成员 #1 的配置，如下所示，该 SD-WAN 成员被一个 SD-WAN 健康检查引用（`health-check.members.seq-num`）：

```bash
FortiGate # diagnose sys cmdb refcnt show system.sdwan.members.seq-num 1
entry used in complex system.sdwan:health-check.members.seq-num child table health-check:name '223.5.5.5'  entry
members:seq-num '1' (From VDOM: 'root')

config system sdwan
    config health-check
        edit "223.5.5.5"
            set server "223.5.5.5"
            set members 1    ←——引用关系
        next
    end
end
```

重要

`<path.object.mkey>` 之间的分隔符可以使用 `.` 或 `:`。如下所有分隔方式均可正常使用：

```bash
diagnose sys cmdb refcnt show system.interface.name wan1
diagnose sys cmdb refcnt show system.interface:name wan1
diagnose sys cmdb refcnt show system:interface:name wan1
```

方法 2 - 使用 get 命令

`get` 命令可以仅列出对象的 mkey，如下所示，防火墙地址对象的 mkey 为 `name`。

```bash
get firewall address
== [ all ]
name: all    ←——mkey为“name”
== [ SSLVPN_TUNNEL_ADDR1 ]
name: SSLVPN_TUNNEL_ADDR1
== [ 10.10.1.4 ]
name: 10.10.1.4
......
```

举例

1. 查找防火墙地址对象“all”的关联配置。如下所示，地址对象“all”正在被多个防火墙策略中的源地址引用。

```bash
FortiGate # diagnose sys cmdb refcnt show firewall.address:name all

entry used by child table source-address:name 'all' of complex vpn.ssl.settings:source-address.name
entry used by child table dst:name 'all' of table service:id '1' of complex system.sdwan:service.dst.name
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '10'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '25'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '20'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '2'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '23'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '13'
entry used by child table srcaddr:name 'all' of table firewall.policy:policyid '12'
```

2. 查找对象所有可能关联的所有配置（并非实际被引用），将 mkey 留空即可：

```bash
FortiGate # diagnose sys cmdb refcnt show firewall.address:name

may used by table firewall.addrgrp.member.name
may used by table vpn.ssl.web.portal.widget.ip-pools.name
may used by table vpn.ssl.web.portal.widget.split-tunneling-routing-address.name
may used by table vpn.ssl.settings.tunnel-ip-pools.name
may used by table vpn.ipsec.phase2.src-name
may used by table vpn.ipsec.phase2.dst-name
may used by table vpn.ipsec.phase1-interface.ipv4-split-include
may used by table vpn.ipsec.phase2-interface.src-name
may used by table vpn.ipsec.phase2-interface.dst-name
may used by table firewall.policy.srcaddr.name
may used by table firewall.policy.dstaddr.name
may used by table firewall.interface-policy.srcaddr.name
may used by table firewall.interface-policy.dstaddr.name
may used by table firewall.sniff-interface-policy.srcaddr.name
may used by table firewall.sniff-interface-policy.dstaddr.name
```

3. 其他举例：

```bash
diagnose sys cmdb refcnt show system.interface.name <interface name>
diagnose sys cmdb refcnt show firewall.address:name <address name>
diagnose sys cmdb refcnt show firewall.profile: <profile name>
diagnose sys cmdb refcnt show firewall.service.group:name <servicegroup name>
```

Ping 工具

FortiGate CLI 的 `execute ping` 和 `execute ping6` 可配合 `ping-options` / `ping6-options` 调整报文参数，用于连通性、路由、SD-WAN 规则、MTU/分片、DNS 解析等场景的排查。

常用参数：

- `adaptive-ping`：快速发出 Ping Request 报文（收到上一个 Echo Reply 后立即发送下一个 Echo Request，也就是快速 Ping）。
- `repeat-count`：指定 Ping 测试次数，避免默认次数不足或持续输出过多。
- `source`：指定源 IP，可用于模拟从不同网段发起的 FortiGate 本地流量。
- `interface`：指定出接口，可用于验证指定链路或路由方向。
- `use-sdwan`：指定是否让 Ping 流量参与 SD-WAN 规则和策略路由匹配。

IPv4 Ping 选项

查看 IPv4 Ping 可用选项：

```bash
execute ping-options ?
```

常见选项如下：

```bash
execute ping-options adaptive-ping <enable|disable>
execute ping-options data-size <bytes>
execute ping-options df-bit {yes | no}
execute ping-options pattern <2-byte_hex>
execute ping-options interface <auto | interface_name>
execute ping-options interval <seconds>
execute ping-options repeat-count <repeats>
execute ping-options source {auto | <source-intf_ip>}
execute ping-options timeout <seconds>
execute ping-options tos <service_type>
execute ping-options ttl <hops>
execute ping-options validate-reply {yes | no}
execute ping-options view-settings
execute ping-options use-sdwan <yes | no>
execute ping-options reset
```

参数说明

- `adaptive-ping <enable|disable>`：开启后，收到上一个 Echo Reply 后立即发送下一个 Echo Request，也就是快速 Ping；默认为关闭状态，每秒 Ping 1 个包。
- `data-size <bytes>`：指定 ICMP 数据报文大小，单位为字节。
- `df-bit {yes | no}`：设置 DF bit。`yes` 表示禁止分片，`no` 表示允许分片，常用于 MTU/分片问题排查。
- `pattern <2-byte_hex>`：使用指定 2 字节十六进制内容填充 ICMP 末尾的数据缓冲区，可配合 `data-size` 测试不同报文大小对链路的影响。
- `interface <auto | interface_name>`：指定 Echo Request 的出接口。`auto` 表示 FortiGate 根据目的地址查路由自动选择出接口。
- `interval <seconds>`：指定每次 Ping 之间的间隔。
- `repeat-count <repeats>`：指定 Ping 的发送次数。
- `source {auto | <source IP>}`：指定 Ping 的源 IP。`auto` 表示 FortiGate 根据实际出接口自动选择源地址。指定不同网段的源 IP 可用于模拟不同源地址发起的连接。
- `timeout <seconds>`：指定等待 Echo Reply 的超时时间，单位为秒。
- `tos <service_type>`：设置 IP 头部 ToS 字段，表达期望的服务质量。
- `ttl <hops>`：指定 TTL，达到该跳数后报文会被丢弃或返回。
- `validate-reply {yes | no}`：设置是否校验返回报文中的数据内容。
- `view-settings`：查看当前 Ping 选项配置。
- `use-sdwan <yes | no>`：默认值为 `no`。设置为 `yes` 后，Ping 流量会检查 SD-WAN 规则和策略路由，通常配合 `source` 使用，以验证特定源地址是否命中特定 SD-WAN 规则。
- `reset`：将 Ping 选项恢复为默认值。

`tos` 可用值如下：

tos 值	ToS 字段	说明
default	0x00	默认值。
lowcost	0x02	尽量降低成本。
lowdelay	0x10	尽量降低延迟。
throughput	0x08	尽量提高吞吐。
reliability	0x04	尽量提高可靠性。

使用示例

指定源 IP、出接口、测试次数，并让测试流量参与 SD-WAN 规则匹配：

```bash
execute ping-options reset
execute ping-options source 10.10.10.1
execute ping-options interface port1
execute ping-options repeat-count 5
execute ping-options use-sdwan yes
execute ping-options view-settings
execute ping 8.8.8.8
```

使用 DF bit 和指定报文大小测试链路 MTU：

```bash
execute ping-options reset
execute ping-options df-bit yes
execute ping-options data-size 1472
execute ping 8.8.8.8
```

测试完成后建议恢复默认值，避免后续 Ping 测试沿用旧参数：

```bash
execute ping-options reset
```

使用 Ping 验证 DNS 解析

`execute ping` 也可以用于验证 FortiGate 是否能解析域名。目的地址填写域名后，如果输出中出现解析后的 IP 地址，说明域名解析成功。

```bash
execute ping google.com
PING google.com (142.250.179.78): 56 data bytes
```

IPv6 Ping 选项

当 FortiGate 已开启 IPv6 功能可见性，并且接口配置了 IPv6 地址后，可以使用 `execute ping6` 测试 IPv6 连通性。

```bash
execute ping6 2001:db8::1
```

查看 IPv6 Ping 可用选项：

```bash
execute ping6-options ?
```

常见选项如下：

```bash
execute ping6-options adaptive-ping <enable|disable>
execute ping6-options data-size <bytes>
execute ping6-options interface <auto | outgoing_interface>
execute ping6-options interval <seconds>
execute ping6-options pattern <2-byte_hex>
execute ping6-options repeat-count <repeats>
execute ping6-options reset
execute ping6-options source6 <auto | IPv6_source_interface>
execute ping6-options timeout <seconds>
```

```
execute ping6-options tos <IPv6_type-of-service_value>
execute ping6-options ttl <time-to-live>
execute ping6-options use-sdwan <yes | no>
execute ping6-options validate-reply <yes | no>
execute ping6-options view-settings
```

IPv6 选项的用途基本与 IPv4 对应选项一致，但有两点差异：

- IPv4 中的 `source` 在 IPv6 中对应为 `source6`。
- IPv6 Ping 选项中没有 `df-bit`。

Sniffer/Debug Flow/NP 加速的矛盾

问题现象

sniffer 或 debug flow 抓不到包或抓不全。这是由于数据被 NP 加速后，数据直接由 NP 转发，不经过 CPU 处理，因此 sniffer、debug flow 将抓不到相应的数据包。

只有被 CPU 处理的数据才能被 sniffer 工具完整抓获，通常是新建数据流才可以抓到包，比如：ICMP 的第一个/第二个包、TCP 三次握手和四次拆链的包、UDP 的第一个/第二个包。

网络拓扑

PC1(192.168.1.10)-----(port5：192.168.1.1)FGT(port1:192.168.89.35)-----Internet

原因分析

1. 设置 Debug Flow：

```bash
diagnose debug flow filter addr 114.114.114.114
diagnose debug flow filter proto 1
diagnose debug flow show function-name enable
diagnose debug console timestamp enable
diagnose debug flow trace start 10
diagnose debug enable
```

2. 使用 PC1（192.168.1.10）ping 114.114.114.114。

3. 第 1 个数据包：ping 请求 echo request。

```bash
2022-11-30 19:54:07 id=20085 trace_id=87 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2589→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2589, seq=1."
2022-11-30 19:54:07 id=20085 trace_id=87 func=init_ip_session_common line=6024 msg="allocate a new session-000ee70c,
tun_id=0.0.0.0"
2022-11-30 19:54:07 id=20085 trace_id=87 func=vf_ip_route_input_common line=2605 msg="find a route: flag=04000000 gw-
192.168.89.254 via port1"
2022-11-30 19:54:07 id=20085 trace_id=87 func=get_new_addr line=1221 msg="find SNAT: IP-192.168.89.35(from IPPOOL),
port-63005"
2022-11-30 19:54:07 id=20085 trace_id=87 func=fw_forward_handler line=881 msg="Allowed by Policy-9: SNAT"
2022-11-30 19:54:07 id=20085 trace_id=87 func=__ip_session_run_tuple line=3471 msg="SNAT 192.168.1.10-
>192.168.89.35:63005"
```

4. 第 2 个数据包：ping 响应 echo reply。

```bash
2022-11-30 19:54:07 id=20085 trace_id=88 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
114.114.114.114:63005→192.168.89.35:0) tun_id=0.0.0.0 from port1. type=0, code=0, id=63005, seq=1."
2022-11-30 19:54:07 id=20085 trace_id=88 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000ee70c, reply direction"
2022-11-30 19:54:07 id=20085 trace_id=88 func=__ip_session_run_tuple line=3484 msg="DNAT 192.168.89.35:0-
>192.168.1.10:2589"
2022-11-30 19:54:07 id=20085 trace_id=88 func=vf_ip_route_input_common line=2605 msg="find a route: flag=00000000 gw-
192.168.1.10 via port5"
2022-11-30 19:54:07 id=20085 trace_id=88 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
port1 to port5, skb.npu_flag=00000000 ses.state=00010204 ses.npu_state=0×04000000"
2022-11-30 19:54:07 id=20085 trace_id=88 func=fw_forward_dirty_handler line=410 msg="state=00010204, state2=00000001,
npu_state=04000000"
```

5. 第 3 个数据包：ping 请求 echo request，数据包发起方向推送到 NP 加速（`npu session installation succeeded`）：

```bash
2022-11-30 19:54:08 id=20085 trace_id=89 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2589→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2589, seq=2."
2022-11-30 19:54:08 id=20085 trace_id=89 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000ee70c, original direction"
2022-11-30 19:54:08 id=20085 trace_id=89 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
```

```
port5 to port1, skb.npu_flag=00000400 ses.state=00010204 ses.npu_state=0×04000000"
2022-11-30 19:54:08 id=20085 trace_id=89 func=ip_session_install_npu_session line=346 msg="npu session installation
succeeded"
2022-11-30 19:54:08 id=20085 trace_id=89 func=fw_forward_dirty_handler line=410 msg="state=00010204, state2=00000001,
npu_state=04000400"
```

6. 第 4 个数据包：ping 响应 echo reply，数据包的响应方向推送到 NP 加速（`npu session installation succeeded`）：

```
2022-11-30 19:54:08 id=20085 trace_id=89 func=__ip_session_run_tuple line=3471 msg="SNAT 192.168.1.10-
>192.168.89.35:63005"
2022-11-30 19:54:08 id=20085 trace_id=90 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
114.114.114.114:63005→192.168.89.35:0) tun_id=0.0.0.0 from port1. type=0, code=0, id=63005, seq=2."
2022-11-30 19:54:08 id=20085 trace_id=90 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000ee70c, reply direction"
2022-11-30 19:54:08 id=20085 trace_id=90 func=__ip_session_run_tuple line=3484 msg="DNAT 192.168.89.35:0-
>192.168.1.10:2589"
2022-11-30 19:54:08 id=20085 trace_id=90 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
port1 to port5, skb.npu_flag=00000400 ses.state=00010204 ses.npu_state=0×04000400"
2022-11-30 19:54:08 id=20085 trace_id=90 func=ip_session_install_npu_session line=346 msg="npu session installation
succeeded"
2022-11-30 19:54:08 id=20085 trace_id=90 func=fw_forward_dirty_handler line=410 msg="state=00010204, state2=00000001,
npu_state=04000c00"
```

7. 对应的 sniffer 抓包如下。

```
FGT # diagnose sniffer packet any 'host 114.114.114.114 and icmp' 4
interfaces=[any]
filters=[host 114.114.114.114 and icmp]
2.544370 port5 in 192.168.1.10 → 114.114.114.114: icmp: echo request
2.544394 port1 out 192.168.89.35 → 114.114.114.114: icmp: echo request

2.564639 port1 in 114.114.114.114 → 192.168.89.35: icmp: echo reply
2.564941 port5 out 114.114.114.114 → 192.168.1.10: icmp: echo reply

3.546345 port5 in 192.168.1.10 → 114.114.114.114: icmp: echo request
3.546356 port1 out 192.168.89.35 → 114.114.114.114: icmp: echo request

3.566460 port1 in 114.114.114.114 → 192.168.89.35: icmp: echo reply
3.566467 port5 out 114.114.114.114 → 192.168.1.10: icmp: echo reply
```

8. 后续的 ping 报文直接通过 NP 芯片转发，因此 sniffer 和 debug flow 都没有输出。此时查看 session list，`offload=8/8` 表示数据包双向卸载到 NP6（NP7 的双向加速状态为 `9/9`），即由 NP 芯片转发。

```
FGT # diagnose sys session list

session info: proto=1 proto_state=00 duration=9 expire=51 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin→sink: org pre→post, reply pre→post dev=13→9/9→13 gwy=192.168.89.254/192.168.1.10
hook=post dir=org act=snat 192.168.1.10:2589→114.114.114.114:8(192.168.89.35:63005)
hook=pre dir=reply act=dnat 114.114.114.114:63005→192.168.89.35:0(192.168.1.10:2589)
misc=0 policy_id=9 pol_uuid_idx=521 auth_info=0 chk_client_info=0 vd=0
serial=000ee70c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0×4000c00 ofld-O ofld-R
npu info: flag=0×81/0×81, offload=8/8, ips_offload=0/0, epid=148/156, ipid=156/148, vlan=0×0000/0×0000
vlifid=156/148, vtag_in=0×0000/0×0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/3
```

解决方法

NP6

方法 1

1. 对于需要抓包的流量新建一条临时策略关闭 NP 加速或在对应策略（找到流量对应的策略及策略 ID）里临时关掉 NP 加速，这样数据的处理会全部走 CPU 处理，可以抓取到完整的数据包。

```bash
config firewall policy
    edit 27                     //注意此处27是策略ID号，不是策略顺序号//
        set auto-asic-offload disable    //在策略中临时关闭NP加速,Sniffer完毕可再enable此命令//
    next
end
```

2. 对于已经建立起来的会话（数据已经走 NP 处理了），可以先过滤出来并清除掉，然后再抓包。

```bash
diagnose sys session filter dst 202.106.1.100
diagnose sys session filter proto 1
diagnose sys session list               //通过session list可以找到对应的策略ID，还可以看到NP加速的标志位//
diagnose sys session clear              //清空该会话//
```

3. 最后再开启抓包命令，这样就可以抓到完全的数据流了。

```bash
diagnose sniffer packet any "host 202.106.1.100 and icmp" 4
```

方法 2

不关闭 NP 情况下的组合抓包脚本，但是只能抓到一次性的加速前的数据包，加速后的包将无法显示出来。

```bash
diagnose sys session filter proto 1
diagnose sys session filter dst 114.114.114.114
diagnose sys session clear    //先将需要抓包的数据流对应的会话清除//

diagnose sniffer packet any "host 114.114.114.114 and icmp" 4 0 l    //再通过抓包命令抓取，可以抓取NP加速前的报文//
```

NP7

NP7 芯片支持在不关闭 NP 加速的情况下抓取 NP7 已加速的报文，请参考章故障排查 → Sniffer 工具 → NP7 芯片抓包 ↗章节。

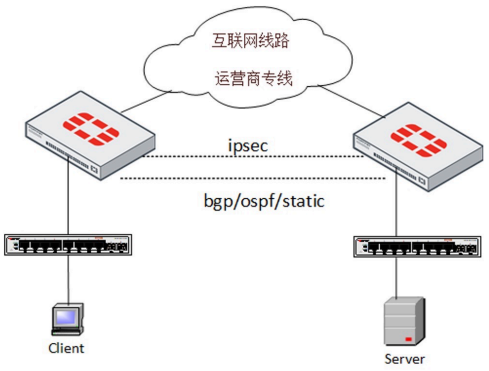
FortiGate 之间测速

功能介绍

从 FortiOS 7.0 起，FortiGate 可以作为 iPerf 服务器（之前只能做客户端）。

当遇到两台 FortiGate 之间的连接速度过低（如 VPN 隧道的带宽无法达到运营商带宽），可以在两台 FortiGate 之间进行 iPerf 测速，以便验证两台 FortiGate 之间的线路带宽。

网络拓扑



1. 客户反馈 VPN Tunnel/专线互联场景下，C/S 上传下载速度达不到预期值（运营商承诺的值）。
2. 某些情况下，当怀疑是营运商的问题时，可以开启防火墙的 iPerf Server 功能，直接在两台防火墙上跨运营商线路进行打流测试。

配置步骤

iPerf Server 配置

1. 全局下开启 speedtest-server，iPerf Server 运行在 TCP 5201，此端口无法修改，只支持 TCP：

```
config system global
   set speedtest-server enable
end
```

2. 在 FortiGate 的 WAN 接口下开启允许 seed-test：

```
config system interface
  edit "port1"
     append allowaccess speed-test
end
```

iPerf Client 配置

1. 配置 iPerf Client，server-intf 和 client-intf 均选择 WAN 接口，端口配置为 5201，协议配置为 0（TCP）：

```
Client # diagnose traffictest server-intf port1
server-intf:    port1
Client # diagnose traffictest client-intf port1
client-intf:    port1
Client # diagnose traffictest port 5201
port:   5201
Client # diagnose traffictest proto 0
proto:  TCP

Client # diagnose traffictest show
server-intf:    port1
client-intf:    port1
port:   5201
proto:  TCP
```

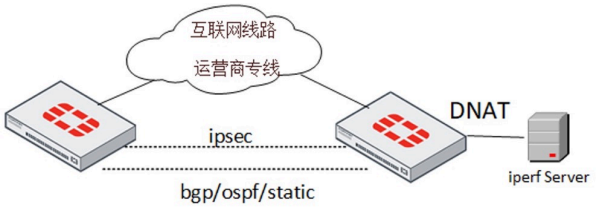
2. Client FortiGate 使用 traffictest 工具向 Server FortiGate 打流（相关参数参考 iPerf 命令）：

```
Client # diagnose traffictest run -c 192.168.91.97 -t 120 -b 10M
```

```
Connecting to host 192.168.91.97, port 5201
[ 11] local 192.168.91.98 port 20157 connected to 192.168.91.97 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[ 11]   0.00-1.00   sec  1.24 MBytes  10.4 Mbits/sec    0   72.1 KBytes
[ 11]   1.00-2.00   sec  1.17 MBytes  9.79 Mbits/sec    0   90.5 KBytes
[ 11]   2.00-3.00   sec  1.21 MBytes  10.1 Mbits/sec    0    123 KBytes
```

注意事项

1. FortiGate 作为 iPerf 服务器时有以下局限性：
   - Server 不能修改端口
   - Server 不支持 UDP 流量
   - Server 不支持多会话并发流量
2. 某些运营商会限制客户端每条会话的速率，此时在 FortiGate 之间 iPerf 测试就无法准确的测出两台 FortiGate 之间运营商的实际带宽。
3. 这种情况下需要改变测试拓扑，将 iPerf 服务器更换为内网的真实 iPerf Server，FortiGate 使用 DNAT 映射内网的 iPerf Server，可以接受多会话 iPerf 测速：




```
Client # diagnose traffictest run -c 192.168.91.92 -P 5
```

```
Connecting to host 192.168.91.92, port 5201
[ 11] local 192.168.91.98 port 13276 connected to 192.168.91.92 port 5201
[ 13] local 192.168.91.98 port 20172 connected to 192.168.91.92 port 5201
[ 15] local 192.168.91.98 port 24766 connected to 192.168.91.92 port 5201
[ 17] local 192.168.91.98 port 6267 connected to 192.168.91.92 port 5201
[ 19] local 192.168.91.98 port 17611 connected to 192.168.91.92 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[ 11]   0.00-1.00   sec  1.05 MBytes  8.80 Mbits/sec    0   1.43 KBytes
[ 13]   0.00-1.00   sec   843 KBytes  6.90 Mbits/sec    0   1.43 KBytes
[ 15]   0.00-1.00   sec  1.37 MBytes  11.5 Mbits/sec    0   1.43 KBytes
[ 17]   0.00-1.00   sec  5.58 MBytes  46.8 Mbits/sec    0   1.43 KBytes
[ 19]   0.00-1.00   sec  1.09 MBytes  9.15 Mbits/sec    0   1.43 KBytes
[SUM]   0.00-1.00   sec  9.92 MBytes  83.2 Mbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
[ 11]   1.00-2.00   sec  1.48 MBytes  12.4 Mbits/sec    0   1.43 KBytes
[ 13]   1.00-2.00   sec   901 KBytes  7.38 Mbits/sec    0   1.43 KBytes
[ 15]   1.00-2.00   sec  1.29 MBytes  10.8 Mbits/sec    0   1.43 KBytes
[ 17]   1.00-2.00   sec  1.70 MBytes  14.2 Mbits/sec    0   1.43 KBytes
[ 19]   1.00-2.00   sec  1.19 MBytes  9.98 Mbits/sec    0   2.85 KBytes
[SUM]   1.00-2.00   sec  6.53 MBytes  54.8 Mbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
[ 11]   2.00-3.00   sec  1.00 MBytes  8.40 Mbits/sec    0   2.85 KBytes
[ 13]   2.00-3.00   sec  1.45 MBytes  12.2 Mbits/sec    0   1.43 KBytes
[ 15]   2.00-3.00   sec  1.24 MBytes  10.4 Mbits/sec    0   2.85 KBytes
[ 17]   2.00-3.00   sec   649 KBytes  5.31 Mbits/sec    0   1.43 KBytes
[ 19]   2.00-3.00   sec  1.02 MBytes  8.60 Mbits/sec    0   1.43 KBytes
[SUM]   2.00-3.00   sec  5.35 MBytes  44.9 Mbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
```

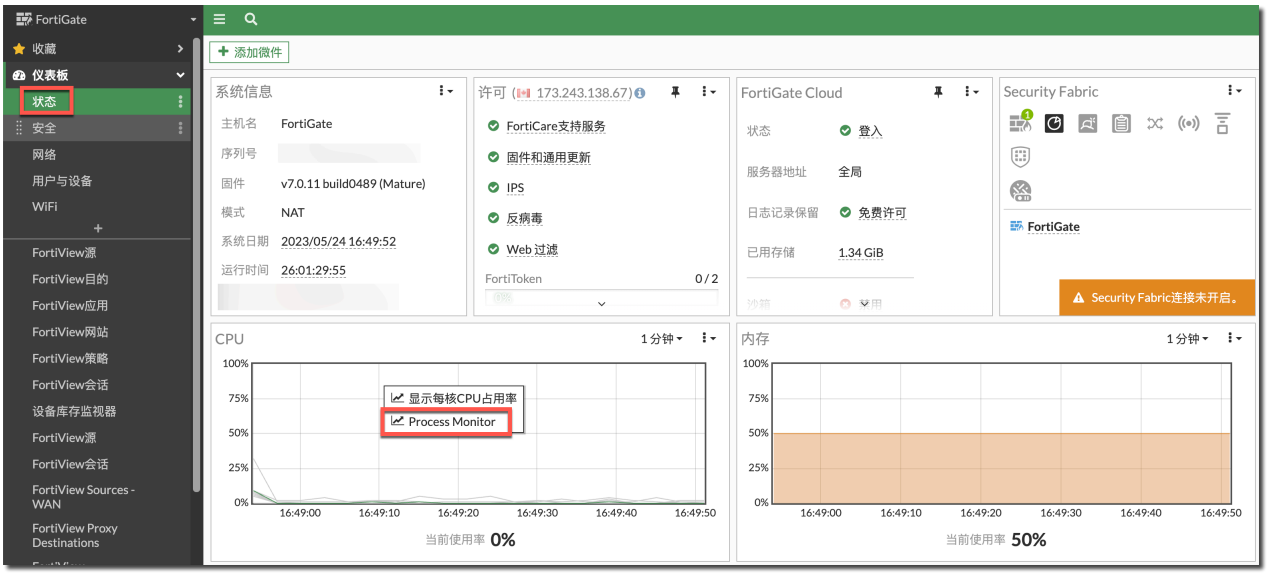
进程监控与管理

功能介绍

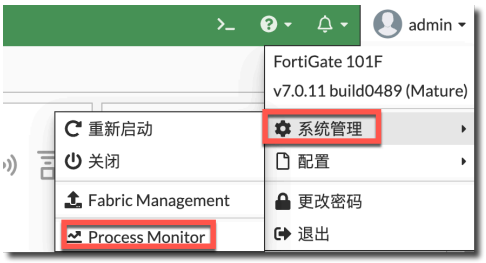
FortiOS 7.0 版本后，在 GUI 上可以监控和管理设备运行的进程。

打开方式

1. 在仪表盘 → 状态的 CPU 监控微件中，左键单击图表任意位置，然后选择 Process Monitor。



2. 在管理页面右上角点击 admin → 系统管理 → Process Monitor。

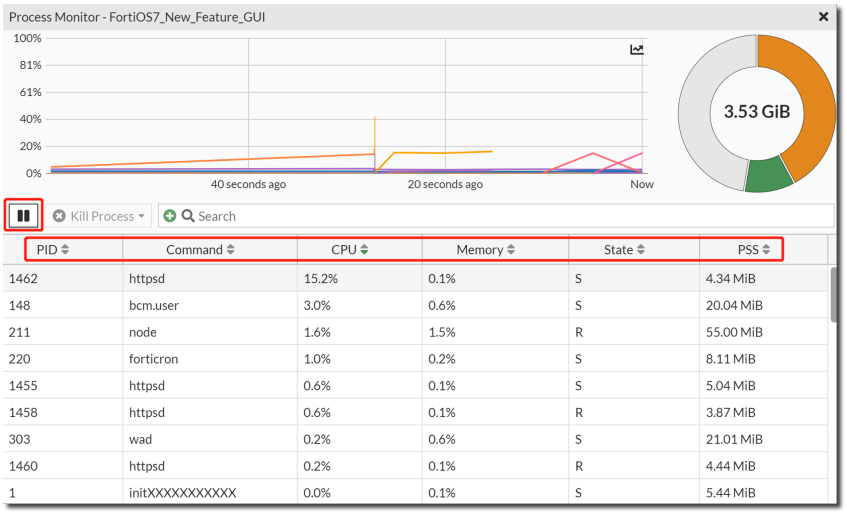


进程查看

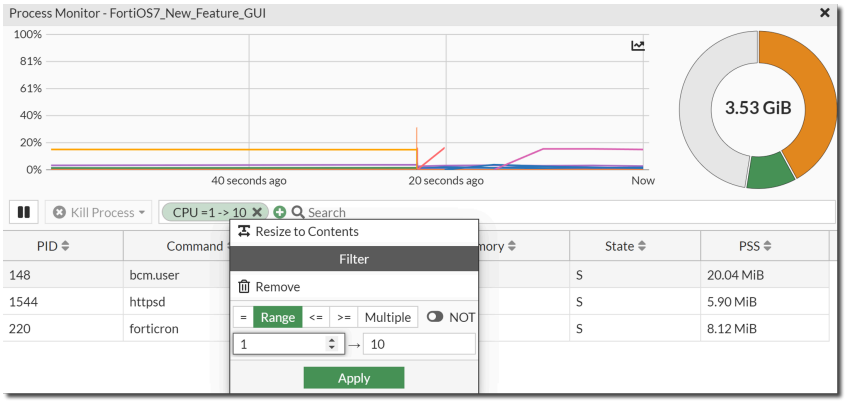
1. 可以查看进程 PID、进程名称、CPU 占用比例、内存占用比例、状态、内存占用，并根据某个进程属性排序。

2. 图表可显示最近 60s 的监控信息。

3. 左侧的暂停按钮可以暂停/继续监控。

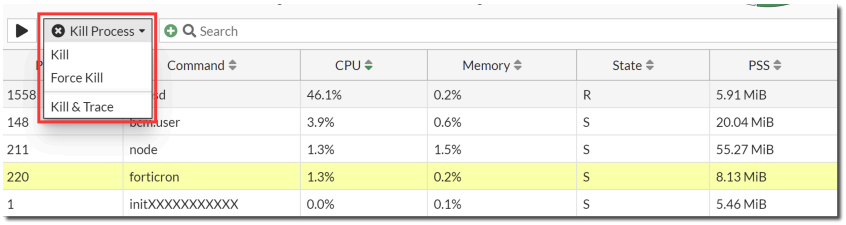


4. 根据过滤条件过滤进程，如 PID、命令、CPU 利用率范围、Memory 利用率范围、状态、PSS 范围。



进程管理

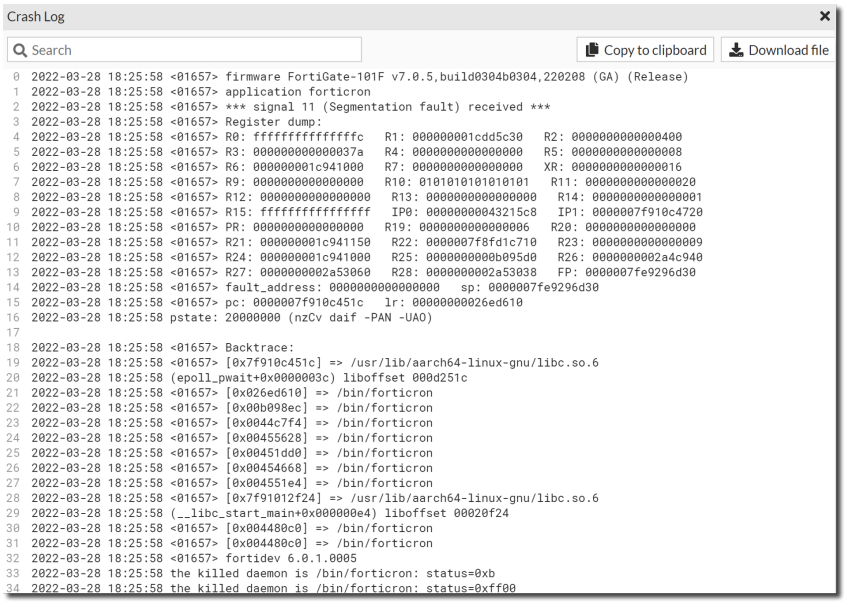
选中某个进程，右键进程或点击左上角的 Kill Process 按钮，可以选择三种 Kill 方式。



- Kill（相当于 CLI 下的 diagnose sys kill PID_Number）。
- Force Kill（相当于 CLI 下的 diagnose sys kill 9 PID_Number）。



- Kill & Trace（相当于 CLI 下的 diagnose sys kill 11 PID_Number），弹出 Crashlog 页面，可下载、复制、搜索。当某个进程出现异常时，可以使用此方式重启进程，并将产生的 Crash log 发送至 TAC 团队进行分析。


Crash Log
Search
Copy to clipboard
Download file
2022-03-28 18:25:58 <01657> firmware FortiGate-101F v7.0.5,build0304b0304,220208 (GA) (Release)
2022-03-28 18:25:58 <01657> application forticron
2022-03-28 18:25:58 <01657> *** signal 11 (Segmentation fault) received ***
2022-03-28 18:25:58 <01657> Register dump:
2022-03-28 18:25:58 <01657> R0: fffffffffffffffc R1: 000000001cdd5c30 R2: 0000000000000400
2022-03-28 18:25:58 <01657> R3: 000000000000037a R4: 0000000000000000 R5: 0000000000000008
2022-03-28 18:25:58 <01657> R6: 000000001c941000 R7: 0000000000000000 XR: 0000000000000016
2022-03-28 18:25:58 <01657> R9: 0000000000000000 R10: 0101010101010101 R11: 0000000000000020
2022-03-28 18:25:58 <01657> R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000001
2022-03-28 18:25:58 <01657> R15: ffffffffffffffff IP0: 00000000043215c8 IP1: 0000007f910c4720
2022-03-28 18:25:58 <01657> PR: 0000000000000000 R19: 0000000000000006 R20: 0000000000000000
2022-03-28 18:25:58 <01657> R21: 000000001c941150 R22: 0000007f8fd1c710 R23: 0000000000000009
2022-03-28 18:25:58 <01657> R24: 000000001c941000 R25: 0000000000b095d0 R26: 0000000002a4c940
2022-03-28 18:25:58 <01657> R27: 0000000002a53060 R28: 0000000002a53038 FP: 0000007fe9296d30
2022-03-28 18:25:58 <01657> fault_address: 0000000000000000 sp: 0000007fe9296d30
2022-03-28 18:25:58 <01657> pc: 0000007f910c451c lr: 00000000026ed610
2022-03-28 18:25:58 pstate: 20000000 (nzCv daif -PAN -UAO)
2022-03-28 18:25:58 <01657> Backtrace:
2022-03-28 18:25:58 <01657> [0x7f910c451c] => /usr/lib/aarch64-linux-gnu/libc.so.6
2022-03-28 18:25:58 (epoll_pwait+0x0000003c) liboffset 000d251c
2022-03-28 18:25:58 <01657> [0x026ed610] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x00b098ec] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x0044c7f4] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x00455628] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x00451dd0] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x00454668] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x004551e4] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x7f91012f24] => /usr/lib/aarch64-linux-gnu/libc.so.6
2022-03-28 18:25:58 (__libc_start_main+0x000000e4) liboffset 00020f24
2022-03-28 18:25:58 <01657> [0x004480c0] => /bin/forticron
2022-03-28 18:25:58 <01657> [0x004480c0] => /bin/forticron
2022-03-28 18:25:58 <01657> fortidev 6.0.1.0005
2022-03-28 18:25:58 the killed daemon is /bin/forticron: status=0xb
2022-03-28 18:25:58 the killed daemon is /bin/forticron: status=0xff00

WAD 进程 Debug

在生产环境中进行 WAD 相关调试时，应结合客户端侧的精确代理访问控制，以及 FortiGate/FortiProxy 上对 WAD Debug 的严格过滤策略，既能满足问题定位需求，又能最大限度降低对设备性能和业务的影响。

- FortiGate/FortiProxy 很多功能依赖 wad 进程来实现，很多时候需要使用 diag wad debug 来诊断故障。
- 但是 WAD Debug 的输出量很大，很容易造成设备 CPU 100%，甚至造成业务故障。
- 所以对 WAD Debug 做最细颗粒度的过滤是非常有必要的。

在进行 WAD 进程的 Debug 时，应注意如下事项：

- 控制客户端只有单个页面通过代理进行访问，而不是整机。方便 WAD Debug。
- 在 FortiGate/FortiProxy 精确限制 WAD Debug 的源 IP 和分类。

WAD Debug 举例

FortiGate 配置

配置代理功能，这里以 HTTP 显式代理端口 8080 为例。配置代理策略放通访问 Internet 的流量。

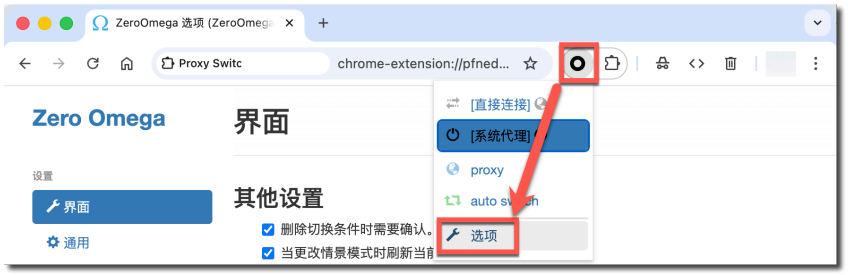
```bash
config web-proxy explicit
    set status enable
    set http-incoming-port 8080
    set https-incoming-port 8080
end

config firewall proxy-policy
    edit 1
        set name "web_proxy_internet"
        set proxy explicit-web
        set dstintf "wan1"
        set srcaddr "192.168.100.0/24"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
    next
end
```

客户端配置

1. 为客户端（192.168.100.99）浏览器（这里以 Chrome 为例）安装 ZeroOmega ↗ 插件（Edge 可以通过微软商店 ↗ 安装）。
2. 插件安装成功后，在扩展程序设置（`chrome://extensions/?id=pfnededegaaopdmhkdmcofjmoldfiped`，Edge 为 `edge://extensions/?id=pfnededegaaopdmhkdmcofjmoldfiped`）中，建议将其固定到工具栏，并开启“在无痕模式下启用”。



3. 点击 ZeroOmega 插件图表，点击“选项”按钮。



4. 点击“新建情景模式”按钮，填写“情景模式名称”，选择类型为“代理服务器”，然后点击“创建”按钮。



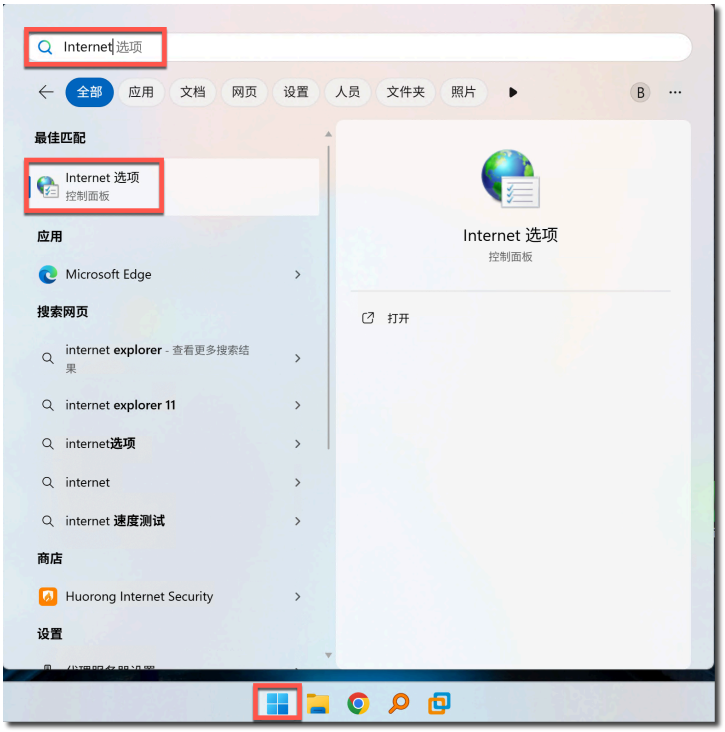
5. 填写代理服务器地址、端口、认证信息（如需），然后点击“应用选项”。



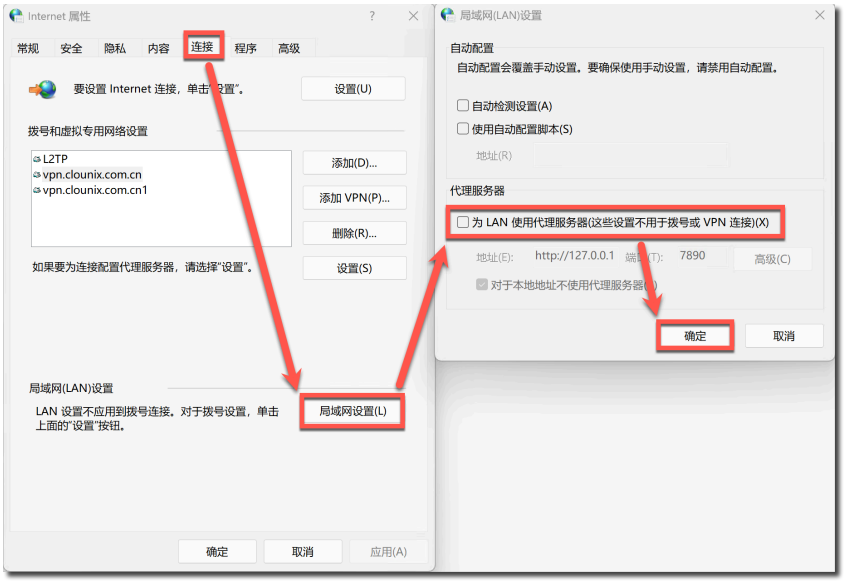
6. 关闭客户端系统的代理。

Windows

1. 在开始菜单中输入“Internet 选项”，点击搜索结果中的“Internet 选项”按钮。

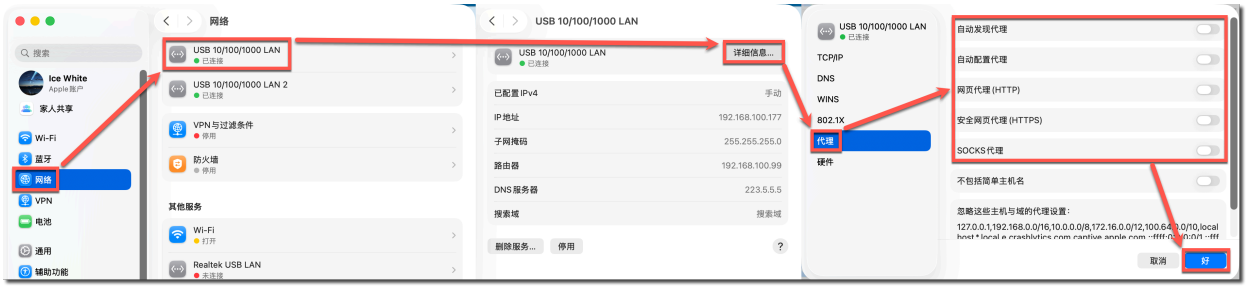


2. 在“连接”选项卡中点击“局域网设置 (L)”按钮，关闭代理服务器配置，然后点击确定按钮。



macOS

打开“系统设置 → 网络”，点击上网使用的网卡，点击“详细信息”，点击“代理”选项，关闭所有的代理配置。

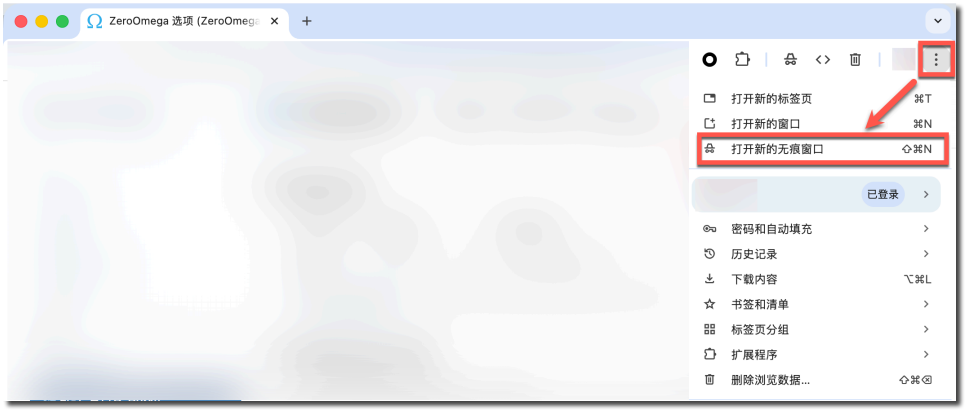


WAD Debug 过程

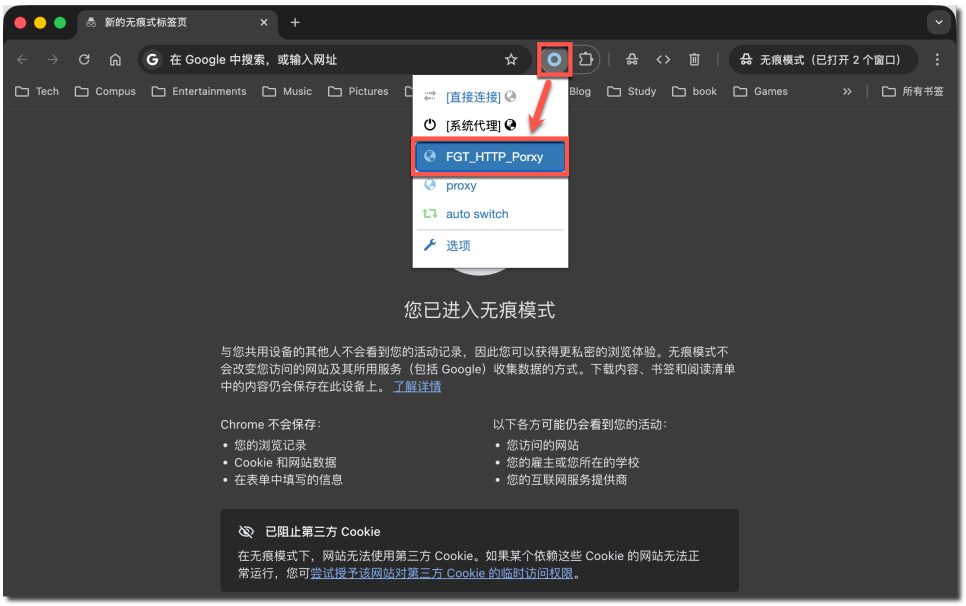
1. 使用 SSH 登录 FortiGate/FortiProxy 并执行 Debug（建议开启终端软件的日志记录），注意限制 Debug 的过滤条件。

```bash
diagnose debug reset
diag debug console timestamp enable
diagnose wad filter src 192.168.100.177      ←——必须
diagnose wad debug enable level verbose
diagnose wad debug enable category policy    ←——按需，打印策略匹配相关Debug
diagnose wad debug enable category dns       ←——按需，打印DNS proxy相关Debug
diagnose wad debug enable category http      ←——按需，打印HTTP代理相关Debug
diagnose debug enable
```

2. 在 Chrome 中打开新的无痕窗口，等待浏览器自身更新/推送/主页广告等会话结束，约 5 分钟。



3. 在 Chrome 浏览器的无痕窗口中点击“ZeroOmega”插件按钮，选择上文创建的代理情景模式。



4. 在 Chrome 浏览器的无痕窗口访问：`https://community.fortinet.com.cn`。

5. 在 FortiGate 上查看 WAD 的 Debug 信息，可以看到 FortiGate 只会记录特定过滤条件的 Debug 日志。

```bash
[V]2025-11-28 21:22:22.595455 [p:22145][s:11400441]   wad_nontp_cache_make              :2169  Matched webproxy object web-proxy
[I]2025-11-28 21:22:22.595497 [p:22145][s:11400441]   wad_http_srv_selector_static_make :1205  make static server selector.
[I]2025-11-28 21:22:22.595506 [p:22145][s:11400441]   wad_http_full_ses_make            :18278 make ok session=0x7f7198f66450 server=(nil).
[V]2025-11-28 21:22:22.595612 [p:22145][s:11400441]   wad_http_mstrm_read_buff          :1742  hs=0x7f7198fc5d78 mstrm=0x7f7198fc5da8 is_clt=1 len=250
[V]2025-11-28 21:22:22.595633 [p:22145][s:11400441][r:585] wad_http_req_alloc                :1751  req=0x7f71a00e2cc0(ses_ctx:x|Phx|M|Hh|C|A7|O) dst(orig=192.168.81.89:8080 dst=192.168.81.89:8080 srv=) hs=0x7f7198fc5d78 clt_port=0x7f719b9df8d8 svr_port=(nil)
[V]2025-11-28 21:22:22.595641 [p:22145][s:11400441][r:585] wad_http_req_line_reader_parse    :624   request line method_len=7 url_len=26 ver_len=8 len=45
[V]2025-11-28 21:22:22.595648 [p:22145][s:11400441][r:585] __wad_http_clt_read_req_line      :1981  http client request line clt=0x7f7198fc5da8
[V]2025-11-28 21:22:22.595651 [p:22145][s:11400441][r:585] wad_http_hdr_reader_parse         :1232  header line len=34 name_len=4 val_len=26 is_piggy=0 invalid=0
[V]2025-11-28 21:22:22.595657 [p:22145][s:11400441][r:585] wad_http_msg_read_hdr             :936   header name=Host: id=41 invalid=0.
[V]2025-11-28 21:22:22.595660 [p:22145][s:11400441][r:585] wad_http_hdr_reader_parse         :1232  header line len=30 name_len=16 val_len=10 is_piggy=0 invalid=0
[V]2025-11-28 21:22:22.595663 [p:22145][s:11400441][r:585] wad_http_msg_read_hdr             :936   header name=Proxy-Connection: id=62 invalid=0.
[V]2025-11-28 21:22:22.595665 [p:22145][s:11400441][r:585] wad_http_hdr_reader_parse         :1232  header line len=139 name_len=10 val_len=125 is_piggy=0 invalid=0
[V]2025-11-28 21:22:22.595669 [p:22145][s:11400441][r:585] wad_http_msg_read_hdr             :936   header name=User-Agent: id=85 invalid=0.
[V]2025-11-28 21:22:22.595671 [p:22145][s:11400441][r:585] wad_http_hdr_reader_parse         :1237  empty line.
[I]2025-11-28 21:22:22.595675 [p:22145][s:11400441][r:585] wad_dump_http_request             :2897  hreq=0x7f71a00e2cc0
Received request from client: 192.168.100.178:12918

CONNECT community.fortinet.com:443 HTTP/1.1
Host: community.fortinet.com:443
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36 Edg/142.0.0.0

[I]2025-11-28 21:22:22.595687 [p:22145][s:11400441][r:585] wad_http_str_canonicalize         :2200  enc=0 path=/ len=1 changes=0
[I]2025-11-28 21:22:22.595700 [p:22145][s:11400441][r:585] wad_http_conn_req_classify        :6393  no security profile HTTPS/HTTP, tport=443
[V]2025-11-28 21:22:22.595708 [p:22145][s:11400441][r:585] _wad_url_match_find               :151   Empty url-matcher!
[V]2025-11-28 21:22:22.595735 [p:22145][s:11400441][r:585] wad_http_req_check_dns            :81    hn=0x7f719ce07ba8
```

```
sn=(nil)
[I]2025-11-28 21:22:22.595740 [p:22145][s:11400441][r:585] wad_http_dns_resolve              :8874  [0×7f71a00e2cc0] DNS
request name=community.fortinet.com len=22 type/pref/pref-strict=2/0/0
[I]2025-11-28 21:22:22.595756 [p:22145][s:11400441][r:585] wad_dns_req_msg_send_local_req    :245   send unreq to
dnsproxy.
msg_len=40, type=wad_local_client_req, vfid=0, vrf=0, ifindex=3, policy_id=0 src_addr=192.168.100.178[I]2025-11-28
21:22:22.595760 [p:22145][s:11400441][r:585] wad_dns_req_msg_send_local_req    :245   send unreq to dnsproxy.
msg_len=40, type=wad_local_client_req, vfid=0, vrf=0, ifindex=3, policy_id=0 src_addr=192.168.100.178[V]2025-11-28
21:22:22.595764 [p:22145][s:11400441][r:585] wad_http_msg_strm_pause           :1119  strm paused, flag=0×2 is_clt=1
[V]2025-11-28 21:22:22.595766 [p:22145][s:11400441][r:585] wad_http_clt_read_sync            :2076  hs=0×7f7198fc5d78
pause=(1/0×2) ret=1 execute=wad_http_clt_read_req_line
[I]2025-11-28 21:22:22.730740 [p:22145]              wad_dnsproxy_conn_proc_common_hdr :626   msg_len=186 msg_type=11
session_id=0
[I]2025-11-28 21:22:22.730762 [p:22145]              wad_dnsproxy_conn_proc_hdr        :603   msg_len=186 msg_type=11
session_id=0
[I]2025-11-28 21:22:22.730772 [p:22145][s:11400441][r:585] wad_http_dns_request_done         :14108 [0×7f71a00e2cc0] DNS
resolved: 52.84.20.38
[V]2025-11-28 21:22:22.730789 [p:22145][s:11400441][r:585] wad_http_msg_strm_resume          :1142  strm resumed,
execute=wad_http_clt_read_req_line is_clt=1
[V]2025-11-28 21:22:22.730808 [p:22145][s:11400441][r:585] wad_http_req_get_dst_intf         :13830 vd=0 dst=52.84.20.38
ifidx=3
[V]2025-11-28 21:22:22.730811 [p:22145][s:11400441][r:585] wad_http_req_proc_dst             :14026 HTTP
req=0×7f71a00e2cc0 check destination/quarantine ret=0
[V]2025-11-28 21:22:22.730817 [p:22145][s:11400441][r:585] wad_http_req_check_policy         :13700 start match policy
vd=0(ses_ctx:x|Phx|M|Hh|C|A7|O) (192.168.100.178:12918@3→52.84.20.38:443@3) absUrl=1
[I]2025-11-28 21:22:22.730829 [p:22145][s:11400441][r:585] wad_fast_match_is_enable          :4031  fast matching is
enabled
[V]2025-11-28 21:22:22.730834 [p:22145][s:11400441][r:585] wad_fast_match_get_addr           :3739  Get key
src:192.168.100.178
[V]2025-11-28 21:22:22.730837 [p:22145][s:11400441][r:585] wad_fast_match_get_dst_intf       :3769  Get key dst intf:2
[V]2025-11-28 21:22:22.730840 [p:22145][s:11400441][r:585] wad_fast_match_pol_array          :3806  Try to maching
pol:1, 0/1(pos/sz)
[I]2025-11-28 21:22:22.730844 [p:22145][s:11400441][r:585] wad_fw_policy_async_match         :7632  pol_ctx:xhcf|Ad|7?
⊨d
[I]2025-11-28 21:22:22.730849 [p:22145][s:11400441][r:585] wad_http_req_policy_set           :11730 match policy-
id=1(pol_ctx:xhcf|Ad|7?⊨d) vd=0(ses_ctx:x|Phx|Mde|Hh|C|A7|O) (192.168.100.178:12918@3 → 52.84.20.38:443@3)
```

6. Debug 结束后关闭所有 Debug 进程。

```
diagnose debug disable
diagnose debug rest
```

AV/IPS 内存高排查

背景介绍

在使用 Antivirus/IPS 的场景中，内存保护模式的触发可能属于高负载条件下的正常保护机制，并不一定表示系统异常或缺陷。

本文以常用的 flow mode 下的 ipsengine + antivirus 为示例，用一台 4GB 内存的 FortiGate 演示：穿过 FortiGate 对一个 2MB 文件发起大量 HTTP: 80 下载请求，目标是制造多个下载同时处于 pending 与 buffered 状态。后续主要聚焦 antivirus 与 ipsengine 的表现与关联，并给出采集、判断与改进方向。

问题现象

系统挂起（System hangs）

- 当发起下载且文件总大小大于 512KB 时，系统会分配 1MB Cached RAM（用于后续缓冲与处理）。
- 示例：若同时启动 1024 个 4MB 文件传输：
  - 初始预分配约 1MB * 1024 = 1GB RAM。
  - 在传输接近结束且多数会话同时处于 pending 状态时，可能累积到 4MB * 1024 = 4GB 的缓冲需求，从而占满 4GB 设备的可用内存。
- ipsengine 没有缓存限制。
- 此类极端场景可能导致系统 hang：转发与管理均不可用。

系统死锁（System deadlock）

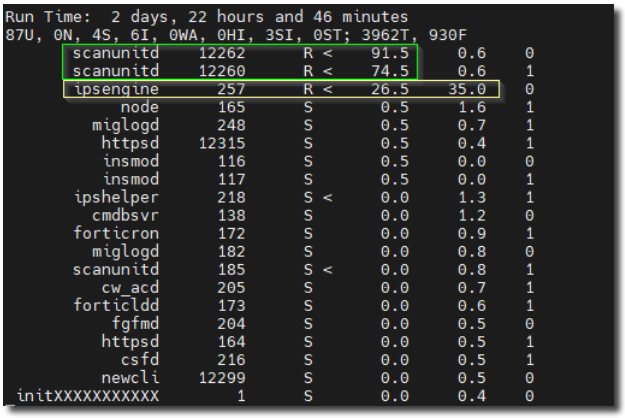
- 在高内存压力下，系统可能耗尽可用 cached memory。
- ipsengine 进入“S”（sleeping）状态，等待 cache 资源可用以继续缓存流量。
- 该状态可能导致业务中断。

排查准则

- **必须在问题发生当下采集数据**：在所有内存保护模式场景中，应在问题发生期间采集调试数据。内存指针、进程内存分配、潜在内存泄漏迹象等关键证据通常只会在异常发生时体现。如果在系统恢复正常或接近正常后再采集，很多现象不会出现在数据中，影响判断。
- **需要观察一段时间内的变化趋势**：不少场景需要通过脚本或周期性采集，在一段时间内观察内存使用的演化过程，判断是否存在持续增长、周期性峰值、或与流量/会话变化一致的波动关系。

初步排查

diagnose sys top 2



- `2` 表示每 2 秒刷新一次。
- 在 top 输出界面中：
  - 按“m”按内存排序。
  - 按“p”按 CPU 排序。
- 记录进程 PID，用于后续追踪。
- 建议连续采集一段时间，观察 CPU 与内存是否波动、是否同步变化，并尝试总结规律。

diagnose sys top-mem 10

```
FG200E-01 # diagnose sys top-mem 10
ipsengine (257): 1103690kB
node (165): 54631kB
ipshelper (218): 39153kB
cmdbsvr (138): 26249kB
forticron (172): 21140kB
scanunitd (185): 18354kB
miglogd (248): 17523kB
miglogd (182): 16424kB
forticldd (173): 14093kB
cw_acd (205): 11594kB
Top-10 memory used: 1322851kB
```

- 列出内存消耗最高的 10 个进程。
- 在较大平台上可适当提高显示进程数。
- 重点关注：
  - 哪个进程消耗最高。
  - 该进程与其他进程合计占用多少。
  - 这几个进程内存之和占系统总内存比例。

diagnose hardware sysinfo conserve

```
FG200E-01 # diag hardware sysinfo conserve
memory conserve mode:                    off   Not in conserve mode now
total RAM:                              3962 MB
memory used:                            2635 MB   66% of total RAM
memory freeable:                         169 MB    4% of total RAM
memory used + freeable threshold extreme: 3764 MB   95% of total RAM
memory used threshold red:              3486 MB   88% of total RAM
memory used threshold green:            3248 MB   82% of total RAM
```

```
FG200E-01 # diag hardware sysinfo conserve
memory conserve mode:                     on   Currenctly in conserve mode
total RAM:                              3962 MB
memory used:                            3605 MB   91% of total RAM
memory freeable:                         142 MB    3% of total RAM
memory used + freeable threshold extreme: 3764 MB   95% of total RAM
memory used threshold red:              3486 MB   88% of total RAM
memory used threshold green:            3248 MB   82% of total RAM
```

- 当前 FortiGate 是否进入了内存保护模式。
- 内存占用率是否超过 red threshold。
- 如果进入了内存保护模式，则比较其值是否高于 red threshold。

diagnose hardware sysinfo memory

```
FG200E-01 # diag hardware sysinfo memory
MemTotal:        4057316 kB 4Gb
MemFree:           30020 kB
Buffers:              88 kB
Cached:          3325712 kB 3.3Gb
SwapCached:            0 kB
Active:           572028 kB
Inactive:        2911196 kB 3.3Gb
Active(anon):     571952 kB
Inactive(anon):  2909128 kB
Active(file):         76 kB
Inactive(file):    2068 kB
Unevictable:     174340 kB
Mlocked:               0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:        331976 kB
Mapped:          2131120 kB
Shmem:           3149108 kB 3.1Gb
Slab:             124228 kB
SReclaimable:      15968 kB
SUnreclaim:       108260 kB
KernelStack:        2592 kB
PageTables:        31756 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     2028656 kB
Committed_AS:   10418248 kB
VmallocTotal:   34359738367 kB
VmallocUsed:      167400 kB
VmallocChunk:   34359329212 kB
DirectMap4k:        6144 kB
DirectMap2M:     4188160 kB
```

- 可能看到 Cached 占用非常高（例如 3.3GB/4GB）。
- 其他可能较高的 bucket 包括 Inactive、Inactive (anon)、Shmem。
- 该输出与 Linux 的/proc/meminfo（cat/proc/meminfo）一致，可参考 Linux 文档 ↗进一步理解各 counter 含义。

get system performance status

```
FG200E-01 # get system performance status
CPU states: 0% user 46% system 0% nice 54% idle 0% iowait 0% irq 0% softirq
CPU0 states: 0% user 64% system 0% nice 36% idle 0% iowait 0% irq 0% softirq
CPU1 states: 0% user 29% system 0% nice 71% idle 0% iowait 0% irq 0% softirq
Memory: 4057316k total, 3903048k used (96.2%), 12748k free (0.3%), 141520k freeable (3.5%)
Average network usage: 78 / 0 kbps in 1 minute, 55641 / 49743 kbps in 10 minutes, 37128 / 33170 kbps in 30 minutes
Maximal network usage: 118 / 3 kbps in 1 minute, 724017 / 720815 kbps in 10 minutes, 724017 / 720815 kbps in 30 minutes
Average sessions: 2239 sessions in 1 minute, 855 sessions in 10 minutes, 723 sessions in 30 minutes
Maximal sessions: 2245 sessions in 1 minute, 8328 sessions in 10 minutes, 8328 sessions in 30 minutes
Average session setup rate: 0 sessions per second in last 1 minute, 18 sessions per second in last 10 minutes, 12 sessions per second in last 30 minutes
Maximal session setup rate: 2 sessions per second in last 1 minute, 590 sessions per second in last 10 minutes, 590 sessions per second in last 30 minutes
Average NPU sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Maximal NPU sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Virus caught: 0 total in 1 minute
IPS attacks blocked: 0 total in 1 minute
Uptime: 0 days,  0 hours,  13 minutes
```

- 该输出包含实时与历史统计，例如：
  - 过去 30 分钟平均网络使用。
  - 最近 10/30 分钟会话数变化。
- 若设备在几分钟前经历过流量冲击，部分非实时的变化在其他命令中不易直接观察，但在此统计中可能更直观。

总结

- 此例中，ipsengine 是主要的内存占用者（示例中 PID 257）。
- scanunitd（一个或多个 worker）是主要 CPU 消耗者（示例中 PID 12262、12260）。
- 设备处于 内存保护模式且超过 red threshold。
- 内存主要被 Cached、Inactive、Inactive (anon)、Shmem 占用。
- 基于以上现象，下一步应重点排查：ipsengine 为何占用大量内存，以及其与 antivirus（scanunitd 高 CPU）之间的关联关系。

ipsengine/antivirus 计数器

本章节目标是确定“pending 状态文件传输缓存用于 AV 检测导致 ipsengine 内存上升”。

diagnose sys process dump 254

- 查看进程 PID 254（前文中查看的 ipsengine 进程 PID，示例因设备重启导致 PID 变化，请忽略 PID 差异）的状态：

```
FG200E-01 # diag sys process dump 254
Status:
Name:   ipsengine
State:  R (running)
Tgid:   254
Pid:    254
PPid:   167
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 4096
Groups:
VmPeak:  3680796 kB
VmSize:  2490724 kB
VmLck:         0 kB
VmPin:         0 kB
VmHWM:   2497780 kB
VmRSS:    952024 kB
VmData:   236508 kB
VmStk:       208 kB
VmExe:     41204 kB
VmLib:     33452 kB
VmPTE:      4136 kB
VmSwap:        0 kB
Threads:        1
SigQ:   0/15627
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000005206
SigIgn: 0000000000001200
SigCgt: 000000018400e4e8
CapInh: 0000000000000000
CapPrm: ffffffffffffffff
CapEff: ffffffffffffffff
CapBnd: ffffffffffffffff
Cpus_allowed:   3
Cpus_allowed_list:      0-1
voluntary_ctxt_switches:        4293630
nonvoluntary_ctxt_switches:     34541
```

3.6Gb

  - 进程状态：running（表示进程正在工作）。
  - VmSize：示例约 3.6GB，表示该进程正在分配/映射大量内存。

- 继续向下滚动输出结果：

```
Maps:
00400000-0043a000 r--p 00000000 00:01 2575                              /bin/init
0043a000-02c3d000 r-xp 0003a000 00:01 2575                              /bin/init
02c3d000-041bb000 r--p 0283d000 00:01 2575                              /bin/init
041bc000-043d8000 r--p 03dbb000 00:01 2575                              /bin/init
043d8000-048fa000 rw-p 03fd7000 00:01 2575                              /bin/init
048fa000-08131000 rw-p 00000000 00:00 0
09e93000-09f16000 rw-p 00000000 00:00 0                                 [heap]
09f16000-0a22f000 rw-p 00000000 00:00 0                                 [heap]
7f50bf20c000-7f50bf45c000 rw-s 00000000 00:0d 20263                     /dev/shm/ips001/f_Wzwrlq (deleted)
7f50bf45c000-7f50bf6ac000 rw-s 00000000 00:0d 21294                     /dev/shm/ips001/f_D4DXCn (deleted)
7f50bf6ac000-7f50bf8b4000 rw-s 00000000 00:0d 26437                     /dev/shm/ips001/f_G5fFap (deleted)
7f50bf8b4000-7f50bfb04000 rw-s 00000000 00:0d 20343                     /dev/shm/ips001/f_K2YOmr (deleted)
7f50bfb04000-7f50bfc0c000 rw-s 00000000 00:0d 28305                     /dev/shm/ips001/f_wZ1X4q (deleted)
7f50bfc0c000-7f50bfd14000 rw-s 00000000 00:0d 28304                     /dev/shm/ips001/f_KJr5Fo (deleted)
7f50bfd14000-7f50bfe1c000 rw-s 00000000 00:0d 28300                     /dev/shm/ips001/f_jcGFQq (deleted)
7f50bfe1c000-7f50bff24000 rw-s 00000000 00:0d 28299                     /dev/shm/ips001/f_Hcz4Hn (deleted)
7f50bff24000-7f50c002c000 rw-s 00000000 00:0d 28298                     /dev/shm/ips001/f_qCrm1q (deleted)
7f50c002c000-7f50c0134000 rw-s 00000000 00:0d 28297                     /dev/shm/ips001/f_syF2Vo (deleted)
7f50c0134000-7f50c023c000 rw-s 00000000 00:0d 28296                     /dev/shm/ips001/f_Wt82Xo (deleted)
7f50c023c000-7f50c0344000 rw-s 00000000 00:0d 28295                     /dev/shm/ips001/f_Z9Boko (deleted)
7f50c0344000-7f50c044c000 rw-s 00000000 00:0d 28294                     /dev/shm/ips001/f_sKZbLq (deleted)
7f50c044c000-7f50c069c000 rw-s 00000000 00:0d 21120                     /dev/shm/ips001/f_ipCU3o (deleted)
7f50c069c000-7f50c07a4000 rw-s 00000000 00:0d 28292                     /dev/shm/ips001/f_slfnQn (deleted)
7f50c07a4000-7f50c08ac000 rw-s 00000000 00:0d 28285                     /dev/shm/ips001/f_eMc29p (deleted)
7f50c08ac000-7f50c0afc000 rw-s 00000000 00:0d 20590                     /dev/shm/ips001/f_DdCliq (deleted)
7f50c0afc000-7f50c0d4c000 rw-s 00000000 00:0d 21908                     /dev/shm/ips001/f_I6GImr (deleted)
7f50c0d4c000-7f50c0f9c000 rw-s 00000000 00:0d 21795                     /dev/shm/ips001/f_wPEEYm (deleted)
7f50c0f9c000-7f50c10a4000 rw-s 00000000 00:0d 28994                     /dev/shm/ips001/f_RUGscp (deleted)
7f50c10a4000-7f50c11ac000 rw-s 00000000 00:0d 28989                     /dev/shm/ips001/f_5jO5Rq (deleted)
7f50c11ac000-7f50c12b4000 rw-s 00000000 00:0d 28988                     /dev/shm/ips001/f_Ozqlon (deleted)
7f50c12b4000-7f50c14bc000 rw-s 00000000 00:0d 26475                     /dev/shm/ips001/f_iM0Shp (deleted)
```

- 在 shm（shared memory）目录下出现大量标记为“deleted”的文件映射。
- ipsengine 创建了这些文件并删除它们。
- 这些文件位于 shm（共享内存空间），通常表示 ipsengine 将数据放入共享内存供其他处理组件使用，或处于等待相邻处理完成的状态。
- 对 scanunitd 做类似的 process dump，通常可看到相同的 shm 文件处理模式。

```
Maps:
00400000-0043a000 r--p 00000000 00:01 2575                              /bin/init
0043a000-02c3d000 r-xp 0003a000 00:01 2575                              /bin/init
02c3d000-041bb000 r--p 0283d000 00:01 2575                              /bin/init
041bc000-043d8000 r--p 03dbb000 00:01 2575                              /bin/init
043d8000-048fa000 rw-p 03fd7000 00:01 2575                              /bin/init
048fa000-08131000 rw-p 00000000 00:00 0
09e93000-09f16000 rw-p 00000000 00:00 0                                 [heap]
09f16000-0a972000 rw-p 00000000 00:00 0                                 [heap]
0a972000-0a9d8000 rw-p 00000000 00:00 0                                 [heap]
7f5198445000-7f5199800000 r--s 00000000 00:0c 54441                     /tmp/.virext_flat_sig
7f5199800000-7f519a000000 rw-p 00000000 00:00 0
7f519a860000-7f519a98d000 rw-p 00000000 00:00 0
7f519af6e000-7f519b006000 rw-s 00000000 00:04 0                         /SYSV64564856 (deleted)
7f519b006000-7f519b567000 rw-p 00000000 00:00 0
7f519bb46000-7f519bd91000 rw-s 00000000 00:0d 144259                    /dev/shm/ips001/f_UgZm6n (deleted)
7f519bd91000-7f519bd99000 rw-s 00000000 00:0d 144295                    /dev/shm/ips001/f_PukFir (deleted)
7f519bd99000-7f519bda9000 r--s 00000000 00:0e 954                       /dev/cmdb/307
7f519bda9000-7f519c0e6000 r--s 00000000 00:0c 54440                     /tmp/.mmdb_flat_sig
7f519c0e6000-7f519c4e3000 r--s 00000000 00:0c 54439                     /tmp/.avai_flat_sig
7f519c4e3000-7f519cb49000 r--s 00000000 00:0c 54438                     /tmp/.vir_flat_sig
7f519cb49000-7f519cbba000 r--p 00000000 08:01 63491                     /data/lib/libav.so
7f519cbba000-7f519cef9000 r-xp 00071000 08:01 63491                     /data/lib/libav.so
7f519cef9000-7f519d170000 r--p 003b0000 08:01 63491                     /data/lib/libav.so
7f519d170000-7f519d19b000 r--p 00626000 08:01 63491                     /data/lib/libav.so
7f519d19b000-7f519d1e4000 rw-p 00651000 08:01 63491                     /data/lib/libav.so
```

- ipsengine/scanunitd 可能会派生多个 worker，PID 可能变化，建议在问题发生时尽快采集。

ipsengine/scanunitd debug

- 启用 Debug：

```bash
diagnose ips debug enable av
diagnose ips debug enable content
diagnose ips debug status show
diagnose sys scanunit debug all enable
diagnose sys scanunit debug level verbose
diagnose sys scanunit debug show
diagnose debug enable
```

- 运行数秒后结束 Debug（调试输出可能非常多，一般不需要运行超过 10 秒）：

```bash
diagnose ips debug disable all
diagnose ips debug status show
diagnose sys scanunit debug all disable
diagnose sys scanunit debug reset
diagnose sys scanunit debug show
```

- 收集 Debug 输出结果：

```
[254@117168]ips_flowav_append: query: 58095, flowav_append: 1448, 452210, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1151287, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 270476, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115851]ips_flowav_append: query: 57438, flowav_append: 1448, 1789181, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1545897, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117168]ips_flowav_append: query: 58095, flowav_append: 1448, 453658, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 642999, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1152735, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 1448, 771157, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115851]ips_flowav_append: query: 57438, flowav_append: 1448, 1790629, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 984033, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 271924, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1547345, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 644447, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1154183, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 1448, 772605, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 985481, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115851]ips_flowav_append: query: 57438, flowav_append: 1448, 1792077, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1548793, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 645895, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 273372, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1155631, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 1448, 774053, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 986929, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115851]ips_flowav_append: query: 57438, flowav_append: 1448, 1793525, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 647343, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1550241, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 274820, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1157079, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 1448, 775501, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 988377, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115851]ips_flowav_append: query: 57438, flowav_append: 1448, 1794973, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 648791, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 276268, bypassed: 0, blk_oversize: 0, oversize_log: 0
```

  - 可以看到“ips_flowav_append”任务持续处理多个 query/task。
  - 多个下载会话不断接收 1448 字节的 chunk，但未完成。
  - 未出现如下“scanunit 任务完整完成”相对应的输出，判断以上 scanunit 任务处于“pending”状态。

```
su 638 req vfid 0 id 56928 ep 0 new request from ipsengine pid 254, size 2401880, fwd-pol 1, oversize 0, url-exempt 0x0, ff-done 0, partial-data 0, dir srv→clt, http-block 0
su 638 job 971 req vfid 0 id 56928 ep 0 received; ack 971, data type: 2
su 638 job 971 request info:
su 638 job 971   client N/A server N/A
su 638 job 971   object_name 'vmwp.exe'
su 638 job 971 heuristic scan enabled
su 638 job 971 enable databases 0f (core avai mmdb extended)
su 638 job 971 scan file 'vmwp.exe' bytes 2401880
su 638 job 971 scan return status 0
su 638 job 971 not wanted for analytics: post-transfer scan submission is disabled at protocol level (m 2 r 0)
su 638 job 971 send result
su 638 job 971 close
su 638 open
su 638 req vfid 0 id 56204 ep 0 new request from ipsengine pid 254, size 2401880, fwd-pol 1, oversize 0, url-exempt 0x0, ff-done 0, partial-data 0, dir srv→clt, http-block 0
su 638 job 972 req vfid 0 id 56204 ep 0 received; ack 972, data type: 2
su 638 job 972 request info:
su 638 job 972   client N/A server N/A
su 638 job 972   object_name 'vmwp.exe'
su 638 job 972 heuristic scan enabled
su 638 job 972 enable databases 0f (core avai mmdb extended)
su 638 job 972 scan file 'vmwp.exe' bytes 2401880
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1168663, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 367, 786004, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 657846, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1561825, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 999961, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 287852, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116099]ips_flowav_append: query: 57583, flowav_append: 1448, 1097344, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117051]ips_flowav_append: query: 58035, flowav_append: 1448, 1170111, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117005]ips_flowav_append: query: 58013, flowav_append: 1448, 659294, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@115781]ips_flowav_append: query: 57404, flowav_append: 1448, 1563273, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117157]ips_flowav_append: query: 58090, flowav_append: 1448, 787452, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@116746]ips_flowav_append: query: 57886, flowav_append: 1448, 1001409, bypassed: 0, blk_oversize: 0, oversize_log: 0
[254@117528]ips_flowav_append: query: 58274, flowav_append: 1448, 289300, bypassed: 0, blk_oversize: 0, oversize_log: 0
```

diagnose test application ipsmonitor 24

- 查看 ipsmonitor 统计中的 antivirus 统计信息：

```
FG200E-01 # diag test application ipsmonitor 24

pid: 254 from 20231222-11:06:21 to 20231222-11:15:58
av_failopen: enabled
    FlowAV mmap                 : 0
    FlowAV file open            : 0
    FlowAV timeout              : 6
    FlowAV req success          : 212
    FlowAV req fail             : 0
    FlowAV req retry success    : 0
    FlowAV req retry fail       : 0
    FlowAV bypassed scan        : 0
    FlowAV buffer scan          : 0
    FlowAV file scan            : 0
    FlowAV interface file open  : 4152
    FlowAV interface file close : 2002
    FlowAV ignored files        : 0
    FlowAV legacy scan          : 212
    FlowAV default scan         : 0
    FlowAV buffer allocation fail : 0
    FlowAV buffer reallocation  : 1290
    FlowAV buffer reallocation fail: 0
FlowAV queue count: 0 retry_count: 0

ips.scanunit_service.counts.connection: now 0  max 0  total 1
ips.scanunit_service.counts.job: now 0  max 15  total 212
ips.scanunit_service.counts.buffer: now 0  max 15  total 212
ips.scanunit_service.counts.queued_jobs: now 0  max 0  total 0
ips.scanunit_service.times.client_to_scanunit : max 1139188 us total 52426075 us avg 254495.52 us (206 values)
ips.scanunit_service.times.scanunit : max 15463513 us total 54683504 us avg 265453.91 us (206 values)
ips.scanunit_service.times.scanunit_to_client : max 1286723 us total 3088473 us avg 14992.59 us (206 values)
```

- `file open` 在 ipsengine 需要缓存一个传输会话并将数据交给 antivirus 做进一步检查时递增。
- `file close` 在对应会话完成/关闭后递增。
- 若传输正常完成，`file open` 与 `file close` 通常应保持相对一致的增长趋势。
- 4152（file open） - 2002（file close） = 2150，存在约 2150 个 pending 状态的传输任务正在被缓存，从而消耗 Cached memory。

- 继续向下滚动 ipsmonitor 输出，可见 `total allocated memory`（Bytes）。

```
shm memory usage
==================
pool buffer size: 4096 available: 0
pool buffer size: 8192 available: 0
pool buffer size: 16384 available: 0
pool buffer size: 32768 available: 0
pool buffer size: 65536 available: 0
pool buffer size: 98304 available: 0
pool buffer size: 131072 available: 0
pool buffer size: 163840 available: 0
pool buffer size: 196608 available: 0
pool buffer size: 229376 available: 0
pool buffer size: 262144 available: 0
total memory used by buffer pools: 0 (kB)
total allocated memory 2761613312 max 2847744000 pool_size 0 pool_max 31673384
allocs: 113129, frees: 110979
pool allocs: 24920, frees: 24996
```

- 2761613312 Bytes 约为 2.7GB。
- 该数值可能与其他命令观察略有差异，因为会话仍在动态处理，但数量级仍处于较高水平。

排查结论

- 当前现象可理解为系统在压力条件下的正常表现：由用户流量与配置组合触发资源占用上升。内存保护模式的触发可能属于高负载条件下的正常保护机制，并不一定表示系统异常或缺陷。
- ipsengine 内存飙升主要由大量 pending 文件传输引起。这些传输需要被缓存，才能由 Antivirus 完成检测。
- 在 flow mode 下，如果不对文件内容进行缓存，将无法完成 Antivirus 检测流程。原因是 Antivirus 首先需要计算 MD5 校验，而 MD5 只能在获取完整文件内容后计算。
- 当 Antivirus 完成 allow/block 判定且会话关闭后，相关缓存内存会随之释放。
- 优化方向：
  - 选择更高型号平台，以提供更高的资源上限来承载更多并发 pending 文件传输与 Antivirus 检测。
  - 评估是否必须对特定业务流量启用 Antivirus。
  - 提升带宽：目标是降低 pending 状态传输任务的数量与持续时间，减少重叠并发带来的缓存累积。
  - 配置优化：参考：开局与日常维护 → 2GB 内存设备配置优化↗章节。

Kernel CPU 高的定位方法

在某些场景下，客户设备会遇到 CPU 持续高占用的情况，通过 FortiGate 自带的 CPU Profiling 程序，可以收集到 CPU 高时的代码级别的 Debug 信息。

通常来说，CPU 可以在 Kernel/System Space 和 User Space 内工作，Kernel Space 用于运行操作系统的可执行文件，以下方法仅适用于 Kernel Space 下 CPU 使用率高的情况，不适用于某个用户进程（如 IPS engine/WAD）CPU 占用高的情况。如下所示，获取的 CPU 状态信息中，System 或 softirg 占用较高，即为 Kernel Space CPU 使用率高（而不是使用 diagnose sys top 中看到的用户进程占用 CPU 高）。

```bash
FortiGate # get system performance status

CPU states: 0% user 99% system 0% nice 1% idle 0% iowait 0% irq 0% softirq
CPU0 states: 0% user 99% system 0% nice 1% idle 0% iowait 0% irq 0% softirq
CPU1 states: 0% user 98% system 0% nice 2% idle 0% iowait 0% irq 0% softirq

FortiGate # get system performance status
CPU36 states: 0% user 0% system 0% nice 0% idle 0% iowait 0% irq 100% softirq
CPU37 states: 0% user 0% system 0% nice 0% idle 0% iowait 0% irq 100% softirq
CPU38 states: 0% user 0% system 0% nice 0% idle 0% iowait 0% irq 100% softirq
CPU39 states: 0% user 0% system 0% nice 0% idle 0% iowait 0% irq 100% softirq
```

定位方法

1. 在 Kernel Space 下 CPU 使用率持续高的情况下，使用如下命令收集信息。

```bash
diag sys profile cpumask 15    //抓取特定的CPU核心，这里以15为例
diag sys profile start    //开启profile程序，等待1分钟左右，再执行后续命令
diag sys profile stop
diag sys profile show order
diag sys profile show detail
diag sys profile sysmap
```

2. 以上操作执行完成后，使用如下命令输出 CPU profile 的结果。

```bash
FortiGate # diag sys profile show order

0×ffffffffa007990f: 2322 fqdn_ip_check+0×4f/0×a0
0×ffffffffa0078537: 547 oal_ioctl+0×a57/0×1070
0×ffffffff803ac8ec: 100 __write_lock_failed+0×c/0×20
0×ffffffffa00323bb: 57 __iprope_check_one_policy+0×32b/0×21c0
0×ffffffffa002bd4b: 38 timer_function+0×ffb/0×2e70
0×ffffffffa009c287: 37 __iprope_tree_check+0×1f7/0×3e0
```

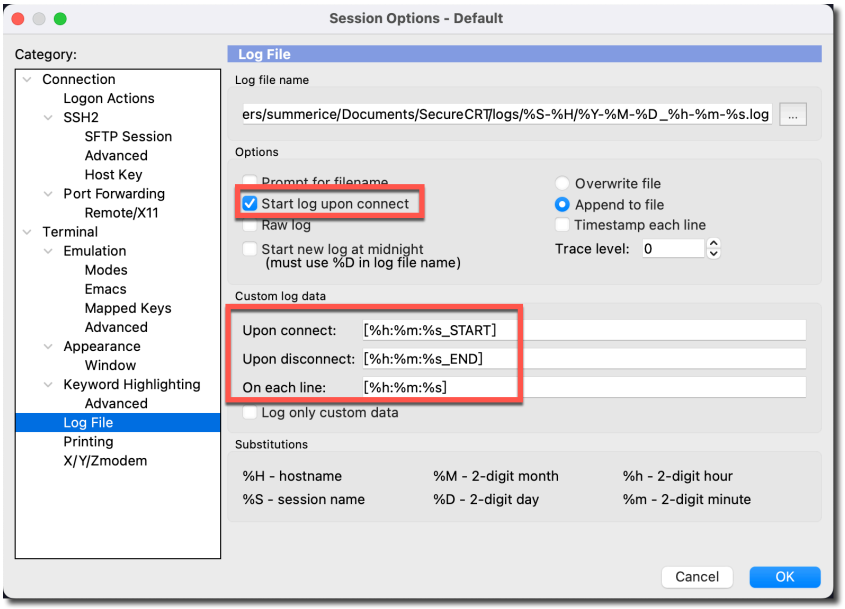
3. 以上结果表明 CPU 在收集期间正在执行的函数，这些函数有两种类型，一种为标准的 Linux 函数，其意义何以在 Linux 标准中找到，另一种为 Fortinet 私有的函数。常见代表的意义如下：

函数名称	含义	排查方法
poll_idle	CPU 没有在做任何操作	忽略
inet_hash_connect inet_check_established	用于打开 Socket 连接	通常在显式代理中出现，可能是 FortiGate 无法在出接口上打开更多的端口导致，解决方法是在显示代理的出接口上设置更多的端口或 IP（config web-proxy explicit）
inet_frag_find ip_defrag	分片	执行分片相关的调试命令（如 sniffer、diagnose snmp ip frags），检查用户网络的 MTU 设置
is_csf_valid	ISDB	与 ISDB 进程或 SD-WAN 有关，检查调用大量 ISDB 条目的相关配置
read_ses_rate	SNMP	检查是否有大批量查询 FortiGate SNMP OID 的流量
__local_bh_enable_ip		
rb_next fqdn_check	FQDN	检查是否存在大量引用 FQDN 的策略
strncasecmp	FTP	与 FTP Session Helper 会话有关
queue_work_on process_one_work	Internal loop inside linux kernel	
__write_lock_failed	通用 Linux 函数，表示进程正在等待进程锁释放	
sal_no_sleep	FIB 转发表项变化	检查是否有大规模的路由变化（BGP/OSPF/IPsec tunnel up/down）
__posix_lock_file	配置变更/更新特征库	本地配置变更，或 FortiGate 正在更新特征库
fqdn_ip_check oal_ioctl __write_lock_failed	被拒绝的流量	DDoS 攻击、或大量数据包被 FortiGate 阻断
ip_session_mark_one_dirty ip6_session_mark_one_dirty	会话正在 dirty	使用命令“diag sys cmdb info”检查配置变化情况，收集 fcnacd 进程的 Debug 信息，也可能有 FortiClient 终端注册情况导致的 Session Dirty
release_pages pagevec_lru_move_fn	与 WAD 向 IPS 发送数据有关	与 UTM 深度包检测有关
do_gettimeoffset_tsc	IPS 检测相关	
unix_find_other.constprop	配置变更	与移动安全策略顺序有关，使用“fnsysctl cat/proc/net/unix”收集信息
add_ip_mcast_session	与组播流量转发有关	
STAT_IVS_COPY_CNT	该计数器用于交换模块的 Flood 复制	与 NP 丢包无关

终端日志记录

对常用的终端工具，建议开启自动生成 SSH 会话日志的功能，对于任何通过 ssh/telnet 进行的 CLI 操作都能保留记录，方便日后追溯，文本化记录也有利于我们协同问题排查。

下图为 SecureCRT 默认会话的日志文件自动化生成配置示例（建议开启时间戳自动记录功能，以便后期方便追溯）。



信息收集

基本信息

版本/序列号

```bash
get system status
```

HA 状态

主备状态/优先级

```
diagnose sys ha status
```

同步状态

```
get system ha status
```

硬件相关

电源状态

```
diagnose hardware deviceinfo psu
```

风扇状态、CPU 温度信息

```
exe sensor list
```

Flash/硬盘状态

```
diagnose hardware deviceinfo disk
diagnose sys flash list
exe disk list
```

Flash Disk 状态检查

```
diagnose hardware smartctl /dev/sda -a
diagnose hardware smartctl /dev/sdb -a
diagnose hardware smartctl /dev/sdc -a
```

接口相关

网络接口状态

```
get system interface
```

接口丢包统计

```
diagnose hardware deviceinfo nic port1（接口名称）
```

光口收发光情况

```
get sys interface transceiver
```

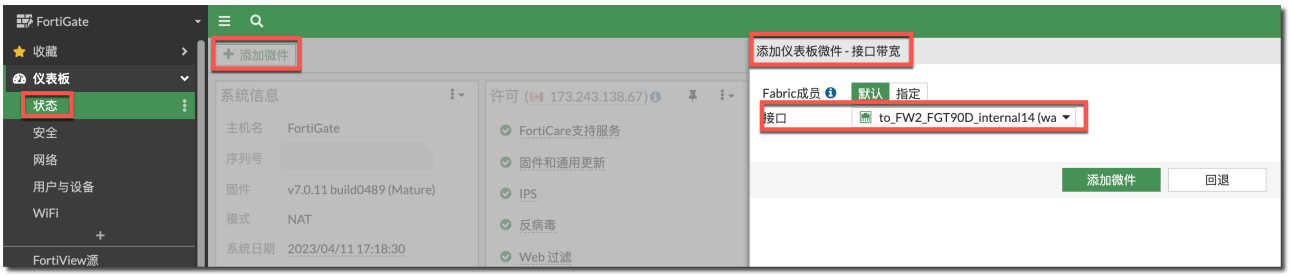
聚合口状态

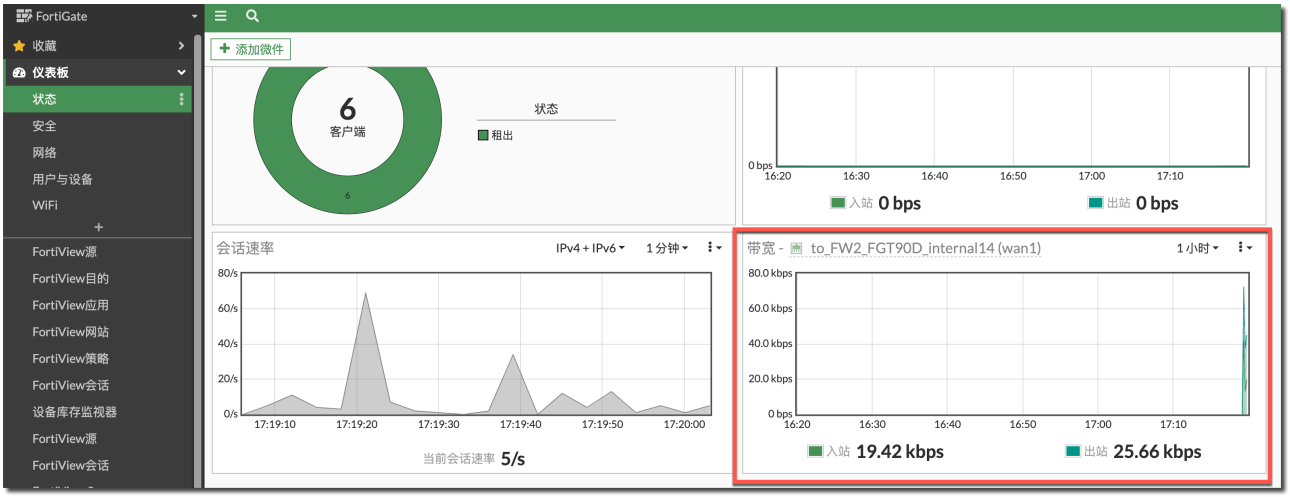
```
diagnose netlink aggregate list
```

VLAN 状态

```
diagnose sys vlan list
```

接口带宽利用率





ARP 表

```
get system arp
```

性能相关

CPU/内存/吞吐量

```
get system performance status
```

用户进程 TOP

```
diagnose sys top 5 20
（键入shift+P以CPU占用排序，键入shift+M以内存占用排序）
```

追踪某个 CPU 的处理模块

> **重要**
>
> 适用于某个 CPU 高时的信息收集。

```
diagnose sys profile cpumask x      //x为CPU编号
diagnose sys profile start
diagnose sys profile stop           //运行完start后，等一会儿再stop
diagnose sys profile show order     //一定要先运行stop再show orfer
```

内存占用

```bash
diagnose hard sysinfo memory
MemTotal: 1911036 kB            ← Total Memory of the hardware.
MemFree: 911804 kB              ← Total free memory of the hardware.
Buffers: 47244 kB               ← Buffer size for temporary memory.
Cached: 393348 kB               ← Cached = Active + Inactive.
SwapCached: 0 kB                ← Cache for memory pages.
Active: 573964 kB               ← Memory used in a particular process.
Inactive: 76000 kB              ← Pages which were not accessed recently.
Active(anon): 463264 kB         ← Anonymous memory.
Inactive(anon): 24252 kB        ← Anonymous memory.
Active(file): 110700 kB         ← Buffer/cache memory which is in active use. This memory is in recent use and is not
used by any other process.
Inactive(file): 51748 kB        ← Buffer/cache memory which is free. This memory is not in use and can be used by any
other process.
Unevictable: 0 kB               ← Pages can't be swapped out for multiple reasons.
Mlocked: 0 kB                   ← Pages locked to memory, Mlocked pages are also Unevictable.
SwapTotal: 0 kB                 ← Total Swap space in memory.
SwapFree: 0 kB                  ← Free Swap space in memory.
Dirty: 20 kB                    ← Memory in wait to get written back to the disk.
Writeback: 0 kB                 ← Memory which is currently written on the disk.
AnonPages: 209384 kB            ← Non file back pages mapped into user space.
Mapped: 78796 kB                ← Mapped files like, libraries.
Shmem: 278144 kB                ← Shared memory.
Slab: 89648 kB                  ← Kernel cache.
SReclaimable: 8292 kB           ← Part of slab/memory which can be claimed.
SUnreclaim: 81356 kB            ← Part of slab/memory which can not be claimed.
KernelStack: 1360 kB            ← Non-reclaimable kernel memory.
PageTables: 16720 kB            ← Amount of memory dedicated to the lowest level of the page.
NFS_Unstable: 0 kB              ← Memory not promised to the storage.
Bounce: 0 kB                    ← Memory used for bounce buffers.
WritebackTmp: 0 kB              ← Memory used for temporary purpose for writeback buffers.
CommitLimit: 955516 kB          ← Amount of memory available to be allocated.
Committed_AS: 7521252 kB        ← Sum of all the memory available to be allocated.
VmallocTotal: 663552 kB         ← Total size of Vmalloc memory area.
VmallocUsed: 152496 kB          ← Total size of Vmalloc memory area in used.
VmallocChunk: 349128 kB         ← Total size of free Vmalloc memory area.
```

内存 Slab 占用

```bash
diagnose hardware sys slab
......
active_objs        ← Number of objects that are currently in use.
num_objs           ← Total number of allocated objects that are in use or not in use.
objsize            ← Size of objects in this slab (in bytes).
objperslab         ← Number of objects stored in each slab.
pagesperslab       ← Number of pages allocated for each slab.

FortiGate # diagnose hardware sys slab
slabinfo - version: 2.1
# name            <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> : tunables <limit> <batchcount>
<sharedfactor> : slabdata <active_slabs> <num_slabs> <sharedavail>
packet_de_duplication6      0      0    192   20    1 : tunables  252  126    8 : slabdata      0      0      0
packet_de_duplication       0      0    128   30    1 : tunables  252  126    8 : slabdata      0      0      0
ip6_nat_record          0      0    128   30    1 : tunables  252  126    8 : slabdata      0      0      0
tcp6_session            0      0   1536    5    2 : tunables   60   30    8 : slabdata      0      0      0
ip6_session             0      3   1344    3    1 : tunables   60   30    8 : slabdata      0      1      0
ip_nat_record           0      0     64   59    1 : tunables  252  126    8 : slabdata      0      0      0
sctp_session            0      0   1536    5    2 : tunables   60   30    8 : slabdata      0      0      0
tcp_session            48    110   1536    5    2 : tunables   60   30    8 : slabdata     22     22      0
ip_session             70    105   1344    3    1 : tunables   60   30    8 : slabdata     35     35      0
fib6_nodes             20     59     64   59    1 : tunables  252  126    8 : slabdata      1      1      0
ip6_dst_cache          95    190    384   10    1 : tunables  124   62    8 : slabdata     19     19      0
```

```
ndisc_cache            5     36    320   12    1 : tunables  124   62    8 : slabdata      3      3      0
ip6_mrt_cache          0      0    128   30    1 : tunables  252  126    8 : slabdata      0      0      0
RAWv6                 13     16   1024    4    1 : tunables  124   62    8 : slabdata      4      4      0
UDPLITEv6              0      0   1024    4    1 : tunables  124   62    8 : slabdata      0      0      0
UDPv6                 17     36   1024    4    1 : tunables  124   62    8 : slabdata      9      9      0
tw_sock_TCPv6          0      0    320   12    1 : tunables  124   62    8 : slabdata      0      0      0
request_sock_TCPv6     0      0    192   20    1 : tunables  252  126    8 : slabdata      0      0      0
TCPv6                 70     72   1856    2    1 : tunables   60   30    8 : slabdata     36     36      0
uhci_urb_priv          0      0     56   67    1 : tunables  252  126    8 : slabdata      0      0      0
scsi_sense_cache       1     30    128   30    1 : tunables  252  126    8 : slabdata      1      1      0
scsi_cmd_cache         1     15    256   15    1 : tunables  252  126    8 : slabdata      1      1      0
sd_ext_cdb             2    112     32  112    1 : tunables  252  126    8 : slabdata      1      1      0
cfq_io_context       133    210    128   30    1 : tunables  252  126    8 : slabdata      7      7      0
cfq_queue            133    204    232   17    1 : tunables  252  126    8 : slabdata     12     12      0
bsg_cmd                0      0    312   12    1 : tunables  124   62    8 : slabdata      0      0      0
mqueue_inode_cache     1      9    832    9    2 : tunables  124   62    8 : slabdata      1      1      0
fat_inode_cache        0      0    640    6    1 : tunables  124   62    8 : slabdata      0      0      0
fat_cache              0      0     32  112    1 : tunables  252  126    8 : slabdata      0      0      0
journal_handle         8    144     24  144    1 : tunables  252  126    8 : slabdata      1      1      0
journal_head          19    136    112   34    1 : tunables  252  126    8 : slabdata      4      4      0
revoke_table           6    202     16  202    1 : tunables  252  126    8 : slabdata      1      1      0
revoke_record          0      0     32  112    1 : tunables  252  126    8 : slabdata      0      0      0
ext2_inode_cache       0      0    720    5    1 : tunables  124   62    8 : slabdata      0      0      0
ext2_xattr             0      0     88   44    1 : tunables  252  126    8 : slabdata      0      0      0
ext3_inode_cache    9188   9395    712    5    1 : tunables  124   62    8 : slabdata   1879   1879      0
kioctx                 0      0    384   10    1 : tunables  124   62    8 : slabdata      0      0      0
kiocb                  0      0    256   15    1 : tunables  252  126    8 : slabdata      0      0      0
dio                    0      0    640    6    1 : tunables  124   62    8 : slabdata      0      0      0
fasync_cache         413    693     48   77    1 : tunables  252  126    8 : slabdata      9      9      0
posix_timers_cache     0      0    144   27    1 : tunables  252  126    8 : slabdata      0      0      0
uid_cache              4     60    128   30    1 : tunables  252  126    8 : slabdata      2      2      0
UNIX                 729    765    832    9    2 : tunables  124   62    8 : slabdata     85     85      0
ip_mrt_cache           0      0  20608    1    8 : tunables    8    4    0 : slabdata      0      0      0
UDP-Lite               0      0    832    9    2 : tunables  124   62    8 : slabdata      0      0      0
tcp_bind_bucket       98    336     32  112    1 : tunables  252  126    8 : slabdata      3      3      0
inet_peer_cache       57    120    192   20    1 : tunables  252  126    8 : slabdata      6      6      0
ip_fib_trie           51    134     56   67    1 : tunables  252  126    8 : slabdata      2      2      0
ip_fib_alias          58    154     48   77    1 : tunables  252  126    8 : slabdata      2      2      0
ip_dst_cache          85    285    256   15    1 : tunables  252  126    8 : slabdata     19     19      0
vf                     4      4   6784    1    2 : tunables    8    4    0 : slabdata      4      4      0
arp_cache              9     24    320   12    1 : tunables  124   62    8 : slabdata      2      2      0
PING                   0      0    832    9    2 : tunables  124   62    8 : slabdata      0      0      0
RAW                   21     36    832    9    2 : tunables  124   62    8 : slabdata      4      4      0
UDP                  100    108    832    9    2 : tunables  124   62    8 : slabdata     12     12      0
tw_sock_TCP            6     24    320   12    1 : tunables  124   62    8 : slabdata      2      2      0
request_sock_TCP       0      0    128   30    1 : tunables  252  126    8 : slabdata      0      0      0
TCP                   96    108   1664    4    2 : tunables   60   30    8 : slabdata     27     27      0
eventpoll_pwq       1139   1378     72   53    1 : tunables  252  126    8 : slabdata     26     26      0
eventpoll_epi       1117   1290    128   30    1 : tunables  252  126    8 : slabdata     43     43      0
sgpool-128             2      2   4096    1    1 : tunables   60   30    8 : slabdata      2      2      0
sgpool-64              2      2   2048    2    1 : tunables   60   30    8 : slabdata      1      1      0
sgpool-32              2      4   1024    4    1 : tunables  124   62    8 : slabdata      1      1      0
sgpool-16              2      8    512    8    1 : tunables  124   62    8 : slabdata      1      1      0
sgpool-8               2     15    256   15    1 : tunables  252  126    8 : slabdata      1      1      0
scsi_data_buffer       0      0     24  144    1 : tunables  252  126    8 : slabdata      0      0      0
blkdev_queue          27     28   1664    4    2 : tunables   60   30    8 : slabdata      7      7      0
blkdev_requests       12     22    344   11    1 : tunables  124   62    8 : slabdata      2      2      0
blkdev_ioc           101    236     64   59    1 : tunables  252  126    8 : slabdata      4      4      0
bio-0                  2     20    192   20    1 : tunables  252  126    8 : slabdata      1      1      0
biovec-256             2      2   4096    1    1 : tunables   60   30    8 : slabdata      2      2      0
biovec-128             0      0   2048    2    1 : tunables   60   30    8 : slabdata      0      0      0
biovec-64              0      0   1024    4    1 : tunables  124   62    8 : slabdata      0      0      0
biovec-16              0      0    256   15    1 : tunables  252  126    8 : slabdata      0      0      0
sock_inode_cache    1280   1309    576    7    1 : tunables  124   62    8 : slabdata    187    187      0
skbuff_head_cache   2390   3213    832    9    2 : tunables  124   62    8 : slabdata    357    357      0
file_lock_cache      336    336    184   21    1 : tunables  252  126    8 : slabdata     16     16      0
shmem_inode_cache   2794   3094    592   13    2 : tunables  124   62    8 : slabdata    238    238      0
task_delay_info      274    340    112   34    1 : tunables  252  126    8 : slabdata     10     10      0
taskstats              1     12    328   12    1 : tunables  124   62    8 : slabdata      1      1      0
proc_inode_cache    2548   2556    592    6    1 : tunables  124   62    8 : slabdata    426    426    232
sigqueue              26     72    160   24    1 : tunables  252  126    8 : slabdata      3      3      0
bdev_cache            17     20    768    5    1 : tunables  124   62    8 : slabdata      4      4      0
```

```
sysfs_dir_cache    10098  10098    144   27    1 : tunables  252  126    8 : slabdata    374    374      0
mnt_cache             48     60    256   15    1 : tunables  252  126    8 : slabdata      4      4      0
filp               11892  13635    256   15    1 : tunables  252  126    8 : slabdata    909    909      0
inode_cache         3559   3577    528    7    1 : tunables  124   62    8 : slabdata    511    511      0
dentry             55582  56260    192   20    1 : tunables  252  126    8 : slabdata   2813   2813    236
names_cache           43     43   4096    1    1 : tunables   60   30    8 : slabdata     43     43      0
iint_cache           111    128    120   32    1 : tunables  252  126    8 : slabdata      4      4      0
buffer_head        24590  24605    104   37    1 : tunables  252  126    8 : slabdata    665    665      0
nsproxy                2     77     48   77    1 : tunables  252  126    8 : slabdata      1      1      0
vm_area_struct     42696  43470    168   23    1 : tunables  252  126    8 : slabdata   1890   1890    252
mm_struct            149    180    768    5    1 : tunables  124   62    8 : slabdata     36     36      0
fs_cache             156    354     64   59    1 : tunables  252  126    8 : slabdata      6      6      0
files_cache          156    176    704   11    2 : tunables  124   62    8 : slabdata     16     16      0
signal_cache         223    224   1024    4    1 : tunables  124   62    8 : slabdata     56     56      0
sighand_cache        223    225   2112    3    2 : tunables   60   30    8 : slabdata     75     75      0
task_struct          273    273   2368    3    2 : tunables   60   30    8 : slabdata     91     91      0
cred_jar             283    570    128   30    1 : tunables  252  126    8 : slabdata     19     19      0
anon_vma_chain     31924  34188     48   77    1 : tunables  252  126    8 : slabdata    444    444    252
anon_vma           15085  16579     64   59    1 : tunables  252  126    8 : slabdata    281    281      0
pid                  269    330    128   30    1 : tunables  252  126    8 : slabdata     11     11      0
radix_tree_node     4302   4788    560    7    1 : tunables  124   62    8 : slabdata    684    684      0
idr_layer_cache      215    219   2112    3    2 : tunables   60   30    8 : slabdata     73     73      0
size-4194304(DMA)      0      0 4194304    1 1024 : tunables    1    1    0 : slabdata      0      0      0
size-4194304           0      0 4194304    1 1024 : tunables    1    1    0 : slabdata      0      0      0
size-2097152(DMA)      1      1 2097152    1  512 : tunables    1    1    0 : slabdata      1      1      0
size-2097152           0      0 2097152    1  512 : tunables    1    1    0 : slabdata      0      0      0
size-1048576(DMA)      4      4 1048576    1  256 : tunables    1    1    0 : slabdata      4      4      0
size-1048576          69     69 1048576    1  256 : tunables    1    1    0 : slabdata     69     69      0
size-524288(DMA)       0      0 524288     1  128 : tunables    1    1    0 : slabdata      0      0      0
size-524288            0      0 524288     1  128 : tunables    1    1    0 : slabdata      0      0      0
size-262144(DMA)       0      0 262144     1   64 : tunables    1    1    0 : slabdata      0      0      0
size-262144            4      4 262144     1   64 : tunables    1    1    0 : slabdata      4      4      0
size-131072(DMA)       0      0 131072     1   32 : tunables    8    4    0 : slabdata      0      0      0
size-131072            3      3 131072     1   32 : tunables    8    4    0 : slabdata      3      3      0
size-65536(DMA)        0      0  65536     1   16 : tunables    8    4    0 : slabdata      0      0      0
size-65536             6      6  65536     1   16 : tunables    8    4    0 : slabdata      6      6      0
size-32768(DMA)        0      0  32768     1    8 : tunables    8    4    0 : slabdata      0      0      0
size-32768            38     38  32768     1    8 : tunables    8    4    0 : slabdata     38     38      0
size-16384(DMA)        0      0  16384     1    4 : tunables    8    4    0 : slabdata      0      0      0
size-16384            24     24  16384     1    4 : tunables    8    4    0 : slabdata     24     24      0
size-8192(DMA)         0      0   8192     1    2 : tunables    8    4    0 : slabdata      0      0      0
size-8192            109    109   8192     1    2 : tunables    8    4    0 : slabdata    109    109      0
size-4096(DMA)         0      0   4096     1    1 : tunables   60   30    8 : slabdata      0      0      0
size-4096           2498   2500   4096     1    1 : tunables   60   30    8 : slabdata   2498   2500      0
size-2048(DMA)         0      0   2048     2    1 : tunables   60   30    8 : slabdata      0      0      0
size-2048            796    798   2048     2    1 : tunables   60   30    8 : slabdata    399    399      0
size-1024(DMA)         1      4   1024     4    1 : tunables  124   62    8 : slabdata      1      1      0
size-1024           1267   1328   1024     4    1 : tunables  124   62    8 : slabdata    332    332      0
size-512(DMA)          0      0    512     8    1 : tunables  124   62    8 : slabdata      0      0      0
size-512            1399   1416    512     8    1 : tunables  124   62    8 : slabdata    177    177      0
size-256(DMA)          0      0    256    15    1 : tunables  252  126    8 : slabdata      0      0      0
size-256             256    285    256    15    1 : tunables  252  126    8 : slabdata     19     19      0
size-192(DMA)          0      0    192    20    1 : tunables  252  126    8 : slabdata      0      0      0
size-192           11992  12340    192    20    1 : tunables  252  126    8 : slabdata    617    617    188
size-128(DMA)          0      0    128    30    1 : tunables  252  126    8 : slabdata      0      0      0
size-64(DMA)           0      0     64    59    1 : tunables  252  126    8 : slabdata      0      0      0
size-64           897962 905591     64    59    1 : tunables  252  126    8 : slabdata  15349  15349      0
size-32(DMA)           0      0     64    59    1 : tunables  252  126    8 : slabdata      0      0      0
size-128            3926   7350    128    30    1 : tunables  252  126    8 : slabdata    245    245      0
size-32             7544   8024     64    59    1 : tunables  252  126    8 : slabdata    136    136      0
kmem_cache           143    160    192    20    1 : tunables  252  126    8 : slabdata      8      8      0
```

会话相关

会话统计

```
diagnose sys session full-stat
```

路由相关

路由表查看

当前路由表

```
get router info routing-table all
```

路由 database

```
get router info routing-table database
```

OSPF

OSPF 状态

```
get router info ospf status
```

OSPF 邻居

```
get router info ospf neighbor
```

BGP

BGP 状态

```
get router info bgp summary
```

BGP 邻居

```
get router info bgp neighbors
```

BGP 路由

```
get router info bgp network
```

BGP 邻居收到/通告的路由

```bash
get router info bgp neighbors x.x.x.x received-routes
get router info bgp neighbors x.x.x.x advertised-routes
```

Debug 信息分析引擎

FTNT Plotter 是一个面向排障场景的在线 Debug 信息分析工具，可将 Debug 信息文本快速转换为可视化结果，帮助定位异常时间段与关键事件。

适用场景：

1. 在大量 Debug 信息中快速识别异常峰值、重复报错和时间分布。
2. 对比不同阶段日志变化，辅助判断问题是否持续或已恢复。
3. 在远程协作排障时，快速共享统一的分析视图。

使用提示：

1. 建议先对 Debug 信息做脱敏处理，再上传或粘贴到在线工具中。
2. Debug 信息量较大时（最大支持单次上传 100MB 文本），可先按时间窗口拆分分析，以提升定位效率。
3. 如内嵌页面无法正常显示，请在新窗口打开 FTNT Plotter ↗。



Zzzz

This app has gone to sleep due to inactivity. Would you like to wake it back up?

Yes, get this app back up!

If you believe this is a bug, please contact us or visit the Streamlit forums.

FortiGate Support Tool

功能描述

FortiGate Support Tool 是一个浏览器插件，它能够在 FortiGate 的图形用户界面上执行后台调试，以收集各种运行信息或错误信息。当您遇到 FortiGate GUI 相关的问题时，如页面无法正常显示，页面打开速度慢等，可以尝试使用该插件收集相关信息，并发送至 Fortinet TAC 团队进行分析。

该插件需要 FortiOS 为 6.2.2 GA 或更高版本才能正常工作，目前支持 Chrome/Edge 浏览器。

获取插件

- 通过 Chrome 商店 ↗下载。
- 离线安装包 ↗。

重要

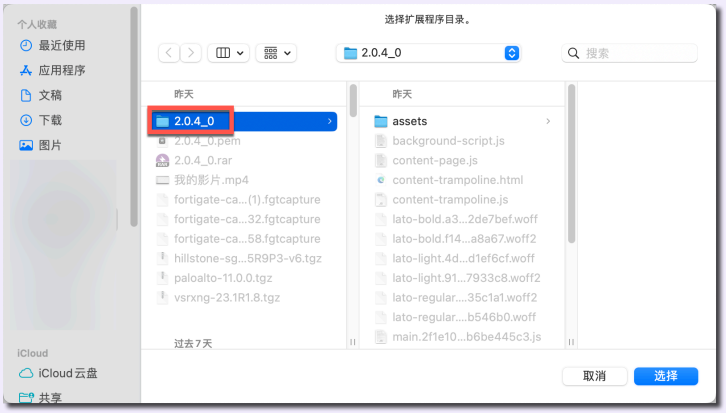
插件离线安装包的安装方法：

1. 解压离线安装包。

2. 进入 Chrome 的插件管理，开启“开发者模式”，点击“加载已解压的扩展程序”。



3. 选择已解压的离线安装包文件夹。

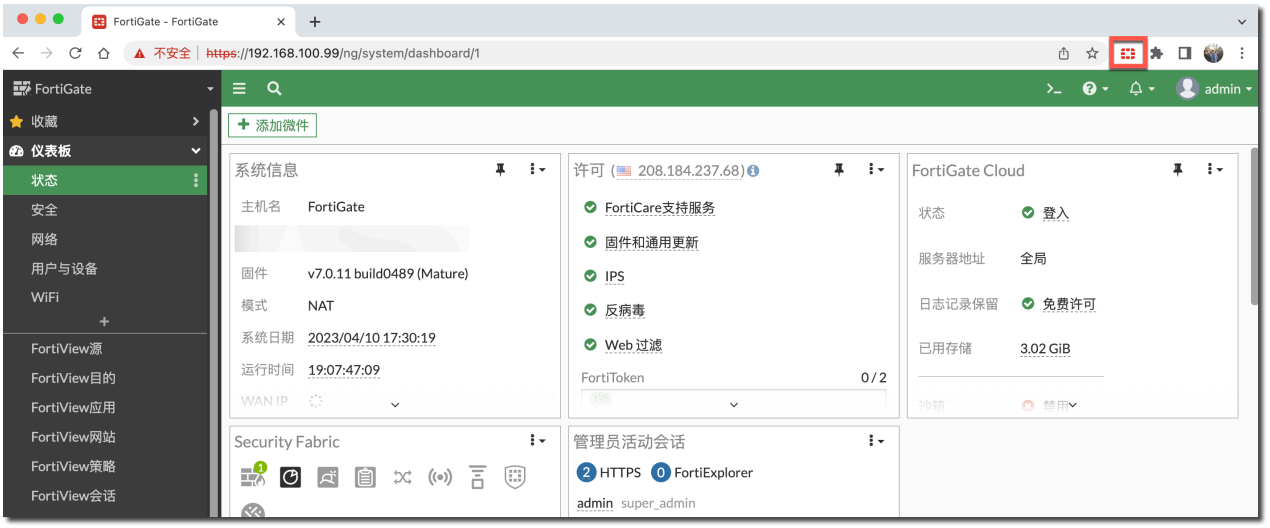


4. 安装成功。

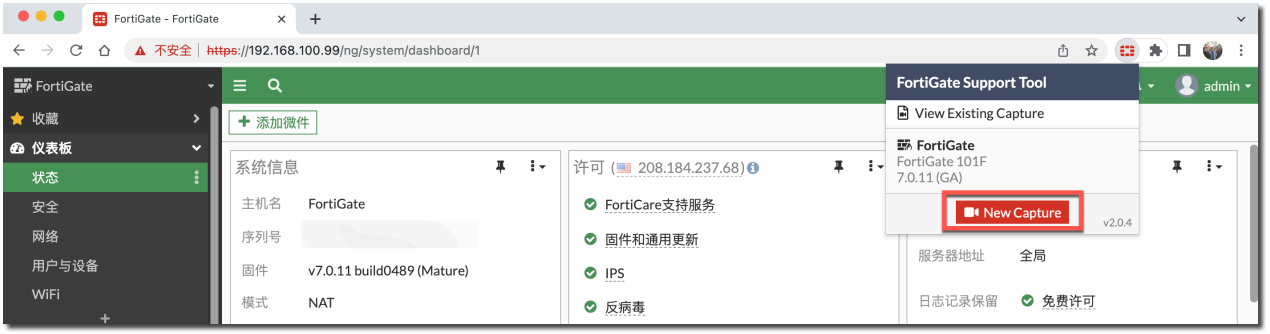


使用方法

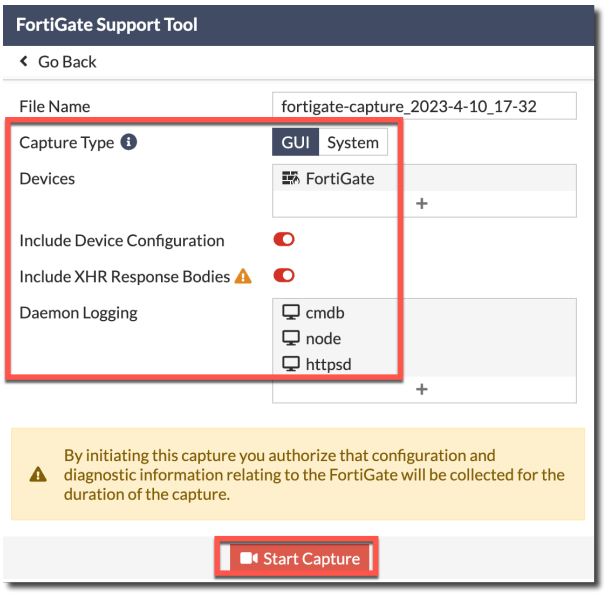
1. 在非 FortiGate 管理页面浏览期间，扩展程序的图标颜色显示为灰色。 当进入 FortiGate 的 GUI 时（登录成功），可以看到该扩展程序图标变红。



2. 点击该插件图标，并点击“New Capture”按钮，开始捕获和收集信息。



3. 可选择收集 GUI 或 System 种类。

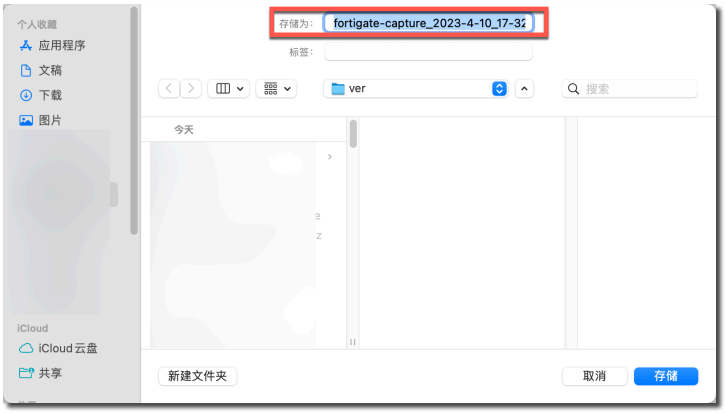


- GUI：全面的捕获，包括视频操作记录、浏览器日志和用户选择的后端守护程序日志。最长时间限制为 10 分钟。
- System：轻量级捕获用户选择的后端守护程序日志，无需记录视频或浏览器活动。最长时间限制为 120 分钟。

4. 这里以 GUI 为例，选择“Deamon Logging”（收集的守护进程日志，一般默认即可），点击“Start Capture”。

5. 开启插件的捕获后，在 GUI 操作重现您的问题现象，重现完成后，停止捕获（点击页面右上角的停止图标），将自动从浏览器下载收集信息文件。





6. 最后，创建 Ticket 并将该文件提供给 FortiGate TAC 团队以供进一步分析。

7. 附操作过程演示视频：



会话工具

- 会话管理
- 会话参数分析
- 会话统计命令
- 基于状态过滤会话

会话管理

查看会话总数

使用 `get system session status` 命令可以查看当前 VDOM 的 IPv4 会话总数。

```bash
FortiGate # get system session status
The total number of IPv4 sessions for the current VDOM: 223
```

查看简要会话列表

使用 `get system session list` 命令可以以表格形式列出当前会话，每行显示一条会话，包含协议类型、超时倒计时、源/目的 IP 端口及 NAT 后地址等基本信息。

```bash
FortiGate # get system session list
PROTO EXPIRE SOURCE              SOURCE-NAT          DESTINATION         DESTINATION-NAT
udp   41     192.168.10.10:50979 192.168.2.18:50979  192.168.2.1:53      -
udp   31     192.168.10.10:51139 192.168.2.18:51139  192.168.2.1:53      -
tcp   3578   192.168.10.10:57186 192.168.2.18:57186  172.172.255.218:443 -
```

各列说明：

- `PROTO`：协议类型。
- `EXPIRE`：会话剩余超时时间（秒）。
- `SOURCE`：源 IP:端口。
- `SOURCE-NAT`：SNAT 转换后的源 IP:端口，未做 SNAT 时与 SOURCE 相同。
- `DESTINATION`：目的 IP:端口。
- `DESTINATION-NAT`：DNAT 转换后的目的 IP:端口，未做 DNAT 时显示为 `-`。

可以使用 `grep` 命令快速过滤特定 IP 地址的会话。

```bash
FortiGate # get system session list | grep 192.168.10.10
tcp   3560   192.168.10.10:58169 192.168.2.18:58169 72.153.5.135:443   -
udp   125    192.168.10.10:51588 192.168.2.18:51588 192.168.2.1:53     -
```

> **提示**
>
> `get system session list` 仅显示摘要信息，如需查看会话的详细字段（如匹配策略、NAT 转换详情、会话状态标志等），请使用 `diagnose sys session list` 命令，详细参数说明请参考会话参数分析章节。

CLI 方式

使用 `diagnose sys session` 命令可以过滤、查看和删除会话。

过滤条件

1. 查看和删除某特定会话时，都需要先设置 filter。

```bash
diagnose sys session filter ?    //键入?查看会话支持的过滤条件，如源IP，目的IP，目的端口等//
vd                Index of virtual domain. -1 matches all.
vd-name           Name of virtual domain. -1 or "any" matches all.
sintf             Source interface.
dintf             Destination interface.
src               Source IP address.
nsrc              NAT'd source ip address
dst               Destination IP address.
proto             Protocol number.
sport             Source port.
nport             NAT'd source port
dport             Destination port.
policy            Policy ID.
expire            expire
duration          duration
```

```
proto-state       Protocol state.
session-state1    Session state1.
session-state2    Session state2.
ext-src           Add a source address to the extended match list.
ext-dst           Add a destination address to the extended match list.
ext-src-negate    Add a source address to the negated extended match list.
ext-dst-negate    Add a destination address to the negated extended match list.
clear             Clear session filter.
negate            Inverse filter.
```

2. 查看当前的过滤条件。

```bash
diagnose sys session filter    //直接键入回车查看当前的过滤条件//
session filter:
        vd: any
        sintf: any
        dintf: any
        proto: any
        proto-state: any
        source ip: any
        NAT'd source ip: any
        dest ip: any
        source port: any
        NAT'd source port: any
        dest port: any
        policy id: any
        expire: any
        duration: any
        state1: any
        state2: any
```

查看会话

1. 查看某特定会话时，需要先设置 filter。如果不设置过滤条件，则 diagnose sys session list 将列出 FortiGate 当前所有的会话。

```bash
diagnose sys session filter proto 1
diagnose sys session filter src 192.168.1.10
```

2. 查看设置的过滤条件。

```bash
diagnose sys session filter
session filter:
        vd: any
        sintf: any
        dintf: any
        proto: 1-1
        proto-state: any
        source ip: 192.168.1.10-192.168.1.10
        NAT'd source ip: any
        dest ip: any
        source port: any
        NAT'd source port: any
        dest port: any
        policy id: any
        expire: any
        duration: any
        state1: any
```

3. 查看过滤后的会话。

```bash
diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=192.168.2.10/192.168.1.10
```

```
hook=pre dir=org act=noop 192.168.1.10:13898->192.168.2.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.2.10:13898->192.168.1.10:0(0.0.0.0:0)
misc=0 policy_id=8 pol_uuid_idx=520 auth_info=0 chk_client_info=0 vd=0
serial=00091df7 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=158/156, ipid=156/158, vlan=0x0000/0x0000
vlifid=156/158, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=5/4
total session 1    //过滤的会话，总共有1条//
```

删除会话

1. 删除某特定会话时，需要先设置 filter。如果不设置过滤条件，则 diagnose sys session clear 将清空 FortiGate 当前所有的会话，可能会引起断网。

```bash
diagnose sys session filter proto 1
diagnose sys session filter src 192.168.1.10
```

2. 查看设置的过滤条件。

```bash
diagnose sys session filter
session filter:
        vd: any
        sintf: any
        dintf: any
        proto: 1-1
        proto-state: any
        source ip: 192.168.1.10-192.168.1.10
        NAT'd source ip: any
        dest ip: any
        source port: any
        NAT'd source port: any
        dest port: any
        policy id: any
        expire: any
        duration: any
        state1: any
```

3. 查看过滤后的会话是否是需要删除的会话。

```bash
FortiGate # diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=192.168.2.10/192.168.1.10
hook=pre dir=org act=noop 192.168.1.10:13898->192.168.2.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.2.10:13898->192.168.1.10:0(0.0.0.0:0)
misc=0 policy_id=8 pol_uuid_idx=520 auth_info=0 chk_client_info=0 vd=0
serial=00091df7 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=158/156, ipid=156/158, vlan=0x0000/0x0000
vlifid=156/158, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=5/4
total session 1
```

4. 确认无误后，再执行删除会话的命令。

```bash
FortiGate # diagnose sys session clear
```

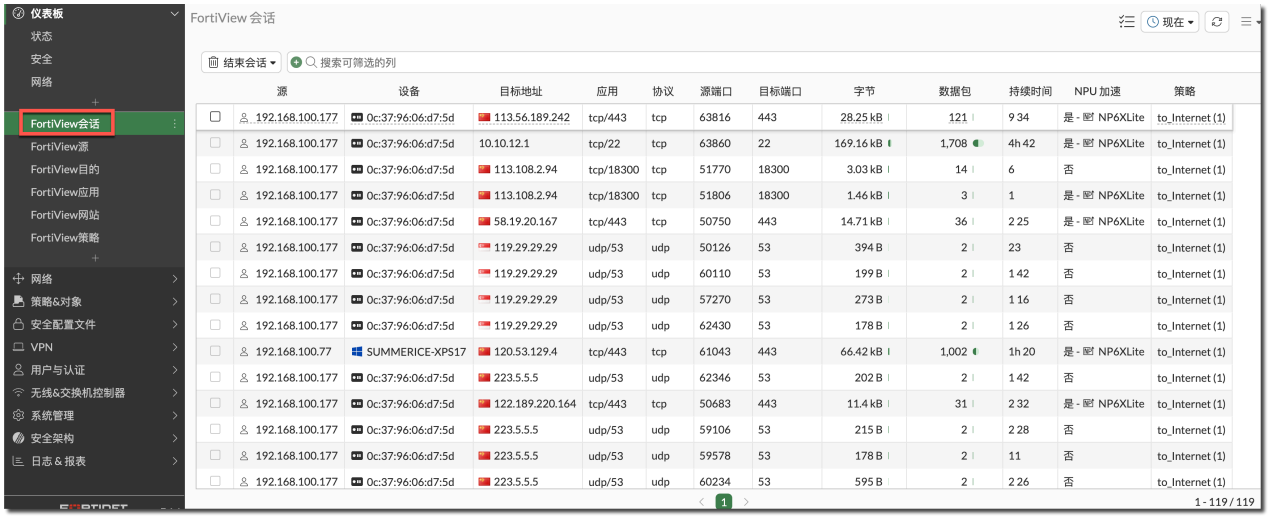
GUI 方式

重要

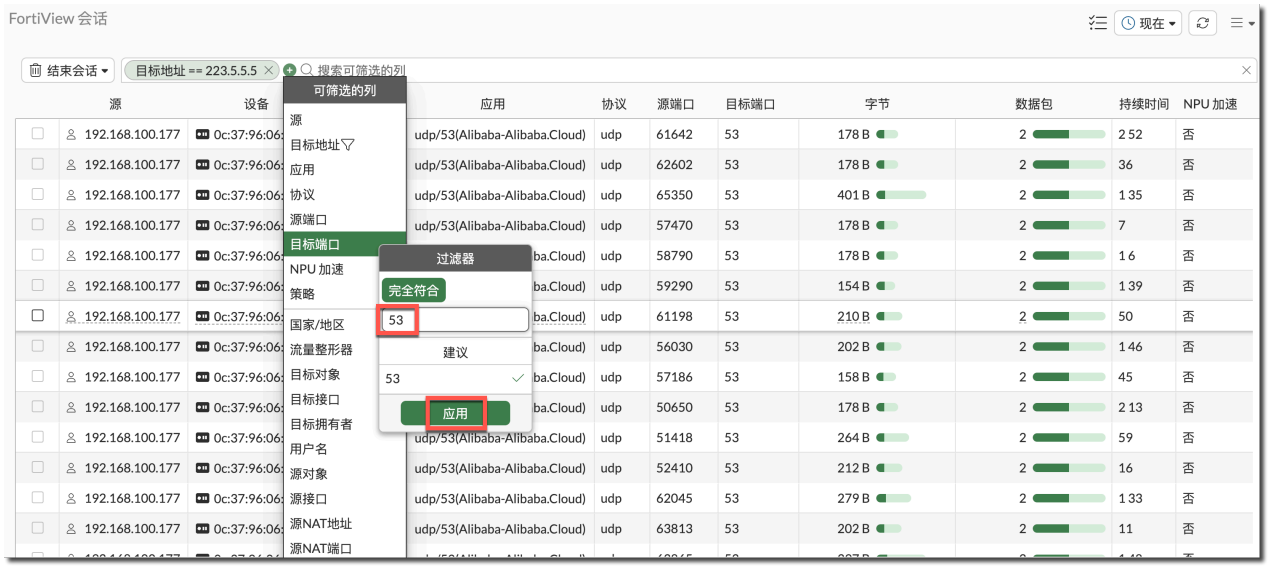
该功能在 FortiOS 7.4.0 及以上版本支持。

过滤条件

1. 登录设备的管理页面，进入仪表盘 → FortiView 会话页面，可以看到当前设备所有活动的会话信息。

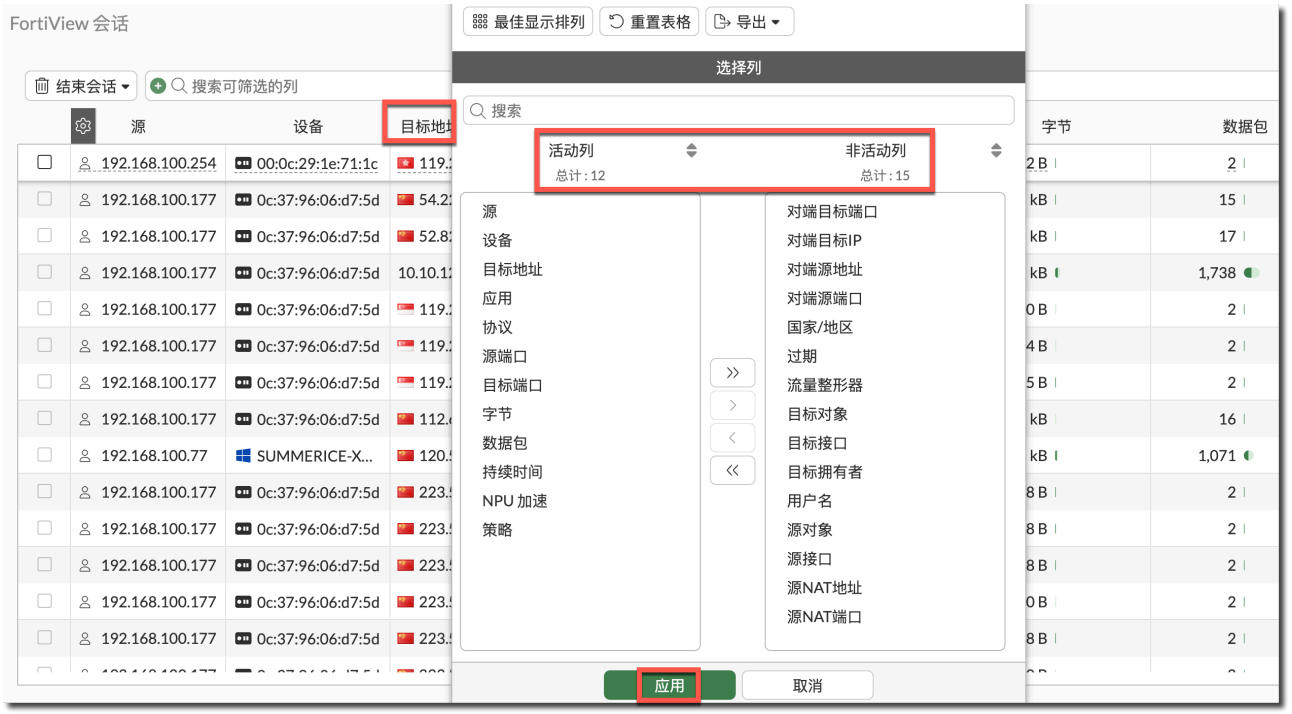


2. 在页面上方的搜索框中点击“+”按钮，可以根据不同的会话参数配置过滤条件，可以设置多个过滤条件，点击应用按钮，可以过滤出对应的会话列表。



查看会话

部分列会被隐藏，右键点击任意列标题，可以配置显示/隐藏的会话参数列。



删除会话

删除过滤结果中的会话

1. 删除过滤结果中的会话，配置筛选条件后查询会话信息（如下所示过滤所有的 ICMP 协议会话），然后点击页面左上角的“结束会话 → 仅筛选”按钮（不会删除本地会话）。



2. 弹出确认提示，点击 OK 按钮。

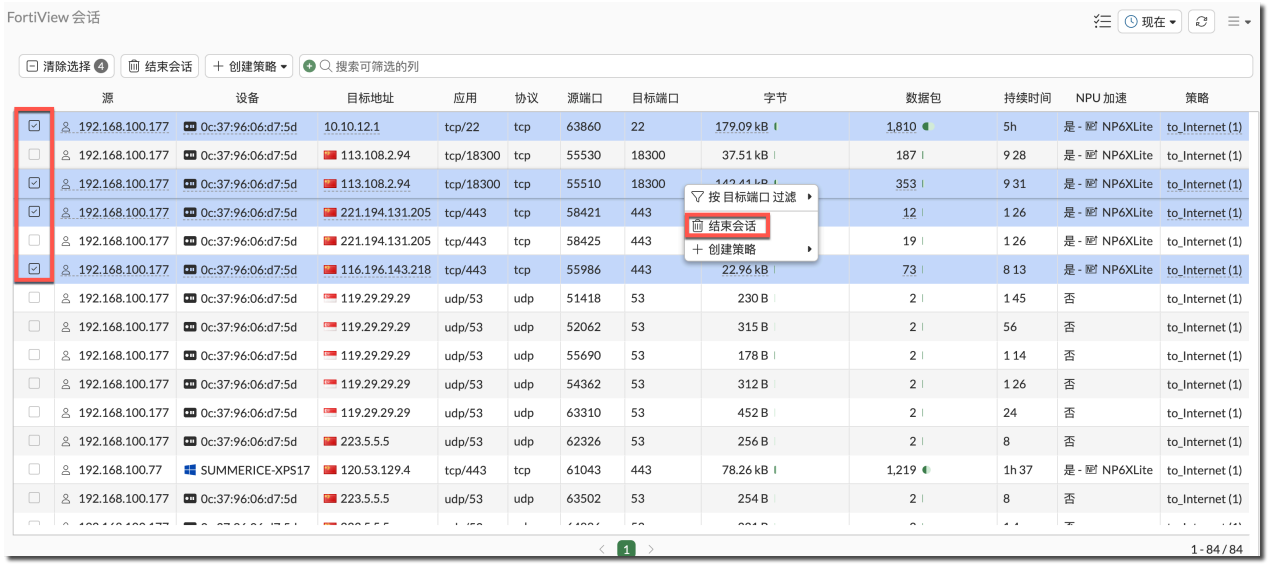


3. 会话删除成功，查看系统日志，显示管理员删除会话的记录，日志中会显示删除会话时所用的过滤器（`filter:vd:0;sintf:not 0;dintf:not 0;proto:1-1;policy id:not 4294967295-4294967295;state1:not (100 mask:102);)`，包含 IPv4 和 IPv6 会话。

```
date=2025-08-29 time=16:34:19 eventtime=1756456458243468840 tz="+0800" logid="0100036883" type="event" subtype="system"bash
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv6 sessions ( filter:vd:0;sintf:not 0;dintf:not 0;proto:1-
1;policy id:not 4294967295-4294967295;state1:not (100 mask:102);)"

date=2025-08-29 time=16:34:19 eventtime=1756456458231928019 tz="+0800" logid="0100036883" type="event" subtype="system"
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;sintf:not 0;dintf:not 0;proto:1-
1;policy id:not 4294967295-4294967295;state1:not (100 mask:102);)"
```

删除手工选中的会话

1. 会话左侧手工选中 4 条会话（或使用 Ctrl/Shift + 鼠标点击选中），右键点击会话，并选择“结束会话”，只会删除手工选中的这 4 个会话（不会删除本地会话）。



2. 弹出确认提示，点击 OK 按钮。



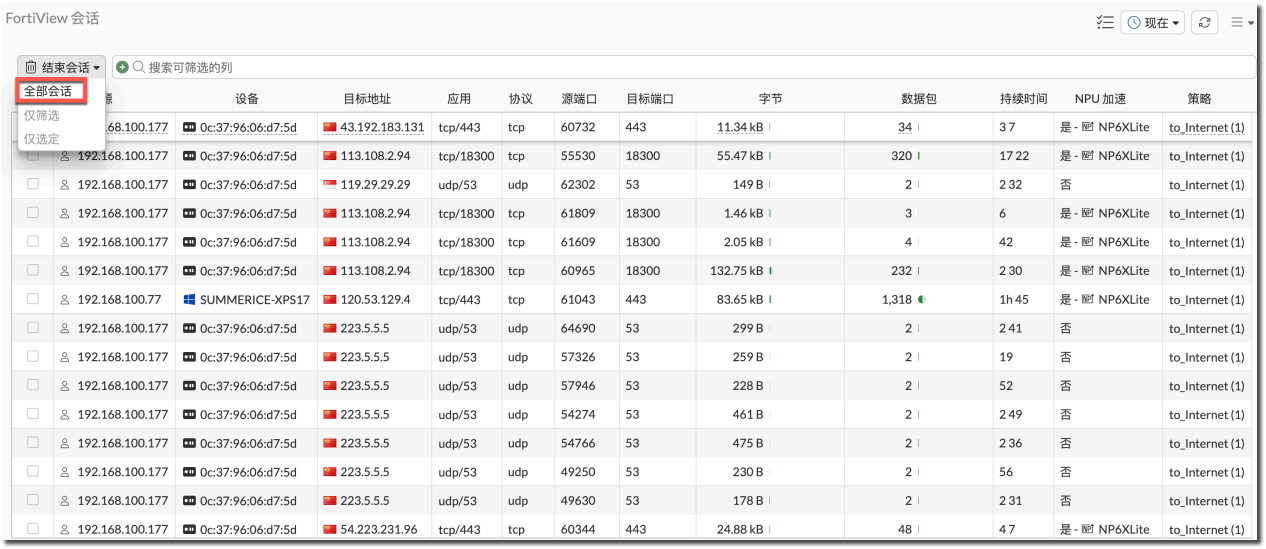
3. 会话删除成功，查看系统日志，显示管理员删除会话的记录，日志中会显示这 4 个会话对应的过滤器。

```
date=2025-08-29 time=16:44:38 eventtime=1756457078331716159 tz="+0800" logid="0100036883" type="event" subtype="system"bash
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;proto:6-6;source ip:192.168.100.177-
192.168.100.177;dest ip:116.196.143.218-116.196.143.218;source port:55986-55986;dest port:443-443;)"

date=2025-08-29 time=16:44:38 eventtime=1756457078281143959 tz="+0800" logid="0100036883" type="event" subtype="system"
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;proto:6-6;source ip:192.168.100.177-
192.168.100.177;dest ip:221.194.131.205-221.194.131.205;source port:58421-58421;dest port:443-443;)"

date=2025-08-29 time=16:44:38 eventtime=1756457078230556839 tz="+0800" logid="0100036883" type="event" subtype="system"
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;proto:6-6;source ip:192.168.100.177-
192.168.100.177;dest ip:113.108.2.94-113.108.2.94;source port:55510-55510;dest port:18300-18300;)"

date=2025-08-29 time=16:44:38 eventtime=1756457078179695119 tz="+0800" logid="0100036883" type="event" subtype="system"
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;proto:6-6;source ip:192.168.100.177-
192.168.100.177;dest ip:10.10.12.1-10.10.12.1;source port:63860-63860;dest port:22-22;)"
```

删除全部会话

> **重要**
>
> 通过 GUI 删除全部会话，全部会话也包含本地会话，例如 BGP 路由会话和管理会话等，请谨慎操作。

1. 删除整机转发流量的全部 IPv4/IPv6 会话，点击页面左上角的“结束会话 → 全部会话”按钮。



2. 弹出确认提示，注意全部会话也包含本地会话，例如 BGP 路由会话和管理会话等，请谨慎操作，点击 OK 按钮。



3. 会话删除成功，查看系统日志，显示管理员删除会话的记录，日志中显示全部会话的过滤器为 `filter:vd:0`。

```
date=2025-08-29 time=16:51:29 eventtime=1756457488487037179 tz="+0800" logid="0100036883" type="event" subtype="system"bash
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv6 sessions ( filter:vd:0;)"

date=2025-08-29 time=16:51:29 eventtime=1756457488475491179 tz="+0800" logid="0100036883" type="event" subtype="system"
level="warning" vd="root" logdesc="Clear active sessions" user="admin" ui="GUI(192.168.100.177)" method="https"
msg="User:admin from GUI(192.168.100.177) clean active IPv4 sessions ( filter:vd:0;)"
```

会话参数分析

```bash
FortiGate # diagnose sys session list
session info: proto=6 proto_state=01 duration=2 expire=3597 timeout=3600 flags=00000000 socktype=0 sockport=0 av_idx=0
use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=192.168.2.10/192.168.1.10
hook=pre dir=org act=noop 192.168.1.10:44610->192.168.2.10:22(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.2.10:22->192.168.1.10:44610(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=8 pol_uuid_idx=520 auth_info=0 chk_client_info=0 vd=0
serial=000c48ed tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=158/156, ipid=156/158, vlan=0x0000/0x0000
vlifid=156/158, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=5/5
```

- `proto`：协议号，ICMP:1，TCP:6，UDP:17。更多请参考：https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml ↗ 。
- `proto_state`：会话状态，跟协议相关。
  - ICMP：ICMP 没有状态，一直显示 proto_state = 00。
  - UDP：UDP 是无状态的协议，但 FortiGate 仍会跟踪两种状态。
    - proto_state = 00：表示没有收到 reply 方向的 UDP 报文。
    - proto_state = 01：表示已收到 reply 方向的 UDP 报文。
  - TCP：TCP 是有状态的协议，proto_state = xy。
    - x 与服务器的状态相关，当没有开启 UTM 或者 Proxy 时，x 一直为 0。
    - y 与客户端的状态相关联，当 FortiGate 收到 syn 时，y 的值为 2；收到 syn/ack 时，y 的值为 3；收到 ack 时，即完成三次握手，y 的值为 1。

State	Value
NONE	0
ESTABLISHED	1
SYN_SENT	2
SYN & SYN/ACK	3
FIN_WAIT	4
TIME_WAIT	5
CLOSE	6
CLOSE_WAIT	7
LAST_ACK	8
LISTEN	9

    TCP 超时时间可在 `config system global` 和 `config firewall service custom` 中调整。
  - SCTP（proto 132）：

State	Value	Expire Timer（默认）
SCTP_S_NONE	0	60 s
SCTP_S_ESTABLISHED	1	3600 s
SCTP_S_CLOSED	2	10 s
SCTP_S_COOKIE_WAIT	3	5 s
SCTP_S_COOKIE_ECHOED	4	10 s
SCTP_S_SHUTDOWN_SENT	5	30 s
SCTP_S_SHUTDOWN_RECD	6	30 s
SCTP_S_SHUTDOWN_ACK_SENT	7	3 s
SCTP_S_MAX	8	n/a

- `duration`：会话存在的时间。
- `expire`：从会话经过最后一个包开始，从 timeout 时间开始倒计时。
- `timeout`：会话在当前状态下的超时时间（TCP 会话各状态超时时间，非特殊情况不要修改，请参考策略与对象 → 配置 Session TTL ↗ 章节）。
- `origin-shaper`：会话发起方流量匹配的 traffic shaper 策略。
- `reply-shaper`：会话反向流量匹配的 traffic shaper 策略。
- `per_ip_shaper`：会话匹配的 per-ip 策略。
- `tunnel`：隧道的名称。
- `state`：会话状态标志。

会话状态	状态说明
may-dirty	会话参数允许被修改。
dirty	会话参数已被修改（需要先有 may-dirty 标志）。
npu	会话通过硬件加速芯片（ASIC）转发。
npd	会话被拒绝进行硬件加速。
npr	会话具备硬件加速资格（可通过 npu info: offload = x/y 查看详情）。
rem	当内存不足时，该会话允许被强制删除。
eph	会话处于临时（ephemeral）状态。
oe	会话属于 IPsec 隧道流量（发起方）。
re	会话属于 IPsec 隧道流量（响应方）。
local	会话与 FortiGate 本地 IP 协议栈关联（本机流量）。
br	会话处于桥接模式（VDOM 为透明模式）。
redir	会话被重定向至 FortiGate 内部代理处理。
wccp	会话被 WCCP 进程拦截处理。
nlb	会话来自负载均衡 VIP。
log	会话正在记录日志。
os	会话在发起方向上做了流量整形。
rs	会话在响应方向上做了流量整形。
ndr	会话正在被 IPS 特征检测。
nds	会话正在被 IPS 异常检测。
auth	会话需要通过认证。
block	会话被重新评估后判定为阻断（策略发生变更）。
ext	（已废弃）会话由 session helper 处理。
app_ntf	会话匹配了包含 `set block-notification enable` 的策略。
f00	在策略中启用流量日志后，会话将具有此标志。
pol_sniff	在策略中启用抓包功能后，会话将具有此标志。
log-start	会话建立时同样记录日志（一条会话产生两条日志：开始和结束）。
rst_tcp	防火墙策略中启用了 `timeout-send-rst` 时可见。
synced	会话已同步至 HA 对端（HA 主设备上可见，需开启 session-pickup）。
syn_ses	会话已从主设备同步至本设备（HA 备设备上可见，需开启 session-pickup）。
need_sync	会话将在存活 30 秒后同步到 HA 对端。
complex	会话由 session helper 处理。
app_valid	应用控制已识别出该会话的应用（会话中会出现 `app=` 字段）。
intree	该会话为子会话（由 SCTP、RPC、SIP、FTP 等 session helper 创建）。子会话的 serial 与父会话相同；父会话删除时，关联子会话同时删除。

hlife	流量匹配了配置有 `set schedule-timeout enable` 的策略，到达计划结束时间后会话将被关闭。

- `dev`：FortiGate 接口索引 index，可以通过 `diagnose netlink interface list` 查看。`dev=13→14/14→13` 表示数据流发起方从 port5 到 port6，数据流的响应方从 port6 到 port5。

```bash
FortiGate# diagnose netlink interface list | grep 13
if=port5 family=00 type=1 index=13 mtu=1500 link=0 master=0
FortiGate# diagnose netlink interface list | grep 14
if=port6 family=00 type=1 index=14 mtu=1500 link=0 master=0
```

- `hook/dir`：会话方向与 NAT 动作。
  - `dir=org`：发起方向，源 IP → 目的 IP，括号内为 NAT 后的地址（`0.0.0.0:0` 表示无 NAT）。
  - `dir=reply`：响应方向，源 IP → 目的 IP，与发起方相反。
  - 做 SNAT 时，NAT 地址覆盖源 IP: 端口；做 DNAT 时，NAT 地址覆盖目的 IP: 端口。格式为 `<源IP>:<源端口>⟶<目的IP>:<目的端口>(<NAT后IP>:<NAT后端口>)`，同时做 SNAT 和 DNAT 的会话示例如下：

```bash
hook=pre  dir=org   act=dnat 120.5.195.158:48186→20.5.194.12:5364(10.25.165.2:5364)
hook=post dir=org   act=snat 120.5.195.158:48186→10.25.165.2:5364(10.3.33.11:48186)
hook=pre  dir=reply act=dnat 10.25.165.2:5364→10.3.33.11:48186(120.5.195.158:48186)
hook=post dir=reply act=snat 10.25.165.2:5364→120.5.195.158:48186(20.5.194.12:5364)
```

- `policy_id`：匹配的策略 ID。
- `vd`：VDOM 的索引 index，root vdom 的索引是 0，通过 `diagnose sys vd list` 查看。

```bash
FortiGate # diagnose sys vd list
name=root/root index=0 enabled fib_ver=136 rpdb_ver=0 use=172 rt_num=35 asym_rt=0 sip_helper=0, sip_nat_trace=1,
mc_fwd=0, mc_ttl_nc=0, tpmc_sk_pl=0
```

  配置了多 VRF 时，vd 字段格式为 `vdom_index:vrf_id`，例如 `vd=0:20` 表示 VDOM index 为 0，VRF ID 为 20。

- `serial`：会话 ID。子会话（expectation session）与其父会话共享相同的 serial。
- `app`：应用 ID。可通过以下命令查看应用详情。

```bash
FortiGate # get application name status | grep "17678" -B 1 -A 11
app-name: "FortiClient"
id: 17678
category: "Update"
cat-id: 17
popularity: 4.low
risk: 1.medium
```

- `url_cat`：URL 分类 ID。常见分类对照如下。

ID	分类	ID	分类
0	Unrated	17	Advertising
1	Drug Abuse	18	Brokerage and Trading
3	Hacking	19	Freeware and Software Downloads
4	Illegal or Unethical	20	Games
5	Discrimination	23	Web-based Email
6	Explicit Violence	24	File Sharing and Storage
25	Streaming Media and Download	28	Entertainment
29	Arts and Culture	30	Education
31	Finance and Banking	33	Health and Wellness
34	Job Search	35	Medicine
36	News and Media	37	Social Networking
38	Political Organizations	39	Reference
40	Global Religion	41	Search Engines and Portals
42	Shopping	43	General Organizations
44	Society and Lifestyles	46	Sports
47	Travel	48	Personal Vehicles
49	Business	50	Information and Computer Security
51	Government and Legal Organizations	52	Information Technology
53	Armed Forces	54	Dynamic Content
55	Meaningless Content	56	Web Hosting
58	Folklore	59	Proxy Avoidance
62	Plagiarism	68	Web Chat
69	Instant Messaging	70	Newsgroups and Message Boards
72	Peer-to-peer File Sharing	75	Internet Radio and TV
76	Internet Telephony	77	Child Education
78	Real Estate	79	Restaurant and Dining
80	Personal Websites and Blogs	81	Secure Websites
82	Content Servers	83	Child Abuse
84	Web-based Applications	85	Domain Parking
87	Personal Privacy	89	Auction
92	Charitable Organizations	93	Remote Access
94	Web Analytics	95	Online Meeting
140	custom1	141	custom2

- `offload`：NP 硬件加速卸载状态（发起方向/响应方向）。
  - 0：会话未卸载到 NP。
  - 8：会话已卸载到 NP6。
  - 9：会话已卸载到 NP7。
  - `offload=8/8`：表示发起方向和响应方向流量均已卸载到 NP6。
  - `offload=9/9`：表示发起方向和响应方向流量均已卸载到 NP7。

会话统计命令

通过 `diagnose sys session full-stat` 命令，可以查看 FortiGate 整机会话的统计信息。

如下所示 `diagnose sys session full-stat` 命令的输出结果。

```bash
FortiGate # diagnose sys session full-stat

session table:           table_size=524288 max_depth=1 used=252
misc info:       session_count=127 setup_rate=0 exp_count=0 reflect_count=0 clash=0
        memory_tension_drop=0 ephemeral=0/239104 removeable=0 extreme_low_mem=0
        npu_session_count=55
        nturbo_session_count=21
delete=0, flush=16, dev_down=64/174
session walkers: active=0, vf-397, dev-64, saddr-0, npu-0, wildcard-0
TCP sessions:
         58 in ESTABLISHED state
         2 in SYN_SENT state
         5 in CLOSE state
firewall error stat:
error1=00000000
error2=00000000
error3=00000000
error4=00000000
tt=00000000
cont=00103420
ips_recv=002aff57
policy_deny=000e0704
av_recv=01a29d3a
fqdn_count=0000000a
fqdn6_count=00000000
```

- `ephemeral`：当前处于 ephemeral（临时）状态的会话数量。ephemeral 会话指处于建立阶段的 TCP 会话，或仅接收单向数据包且未双向通信的 UDP 会话。
  - 由于 FortiGate 需要分配内存来管理会话，此类半连接会话是常见的 DoS 攻击形式。
  - 为避免此类攻击，FortiGate 将处于这些状态的会话归类为 ephemeral 会话，并根据设备型号设定同时存在的会话数量上限，以防止内存耗尽。其核心目的是提供一种 DoS 防护。
- `memory_tension_drop`：因系统内存不足而删除的会话数量，将删除最旧的会话，TCP 本地会话不会被删除。
- `removeable`：当可用内存不足以支持创建新会话时，统计现有防火墙会话中可供删除的候选会话数量。
- `npu_session_count`：当前被 NP 芯片加速转发的会话数量。
- `nturbo_session_count`：当前被 nturbo 加速的会话数量。
- `flush`：统计因用户操作或某些功能删除的会话数量，例如：
  - 管理员使用 GUI 删除特定会话。
  - 被 IPS 规则阻断的会话。
  - 当 SIP ALG 会话因通话结束而被删除时，若该通话通过 UDP 建立，SIP ALG 将删除对应的会话，以避免该会话被意外重复用于其他 SIP 呼叫。
- `dev_down`：统计当 IPSec 接口被管理性关闭时，删除的相关 IPSec 接口会话。
- `cont`：统计被发送到 IPS 并由 IPS 检测后继续放通的报文数量。计数结果以十六进制形式显示。
- `ips_recv`：统计发往 IPS 处理的报文数量。若设备支持 nTurbo 且该功能已启用，则通过 nTurbo 传递至 IPS 的数据包不计入此统计。计数结果以十六进制形式显示。
- `av_recv`：统计发往 AV 处理的数据包数量。统计结果以十六进制形式显示。
- `fqdn_count`：统计内核中当前存在，且与 `config firewall address`、`config firewall address6` 和 `config firewall service` 中配置对应的 FQDN 条目数量。计数结果以十六进制形式显示。
- `error1`：统计接收到的数据包加密状态与其所属会话不匹配。例如：
  - 当会话要求解密数据包时，该数据包仍为明文（例如在欺骗攻击）。
  - 数据包已正确解密，但会话未关联任何 IPsec 隧道。
- `error2`：统计在数据包出站时，所选接口与关联会话中的出接口不匹配的次数。
- `error3`：ALG 的 expectation 会话用于在防火墙上建立针孔（如 FTP 协议），并始终关联至生成该 expectation 的父会话。系统为每个父会话设定了可同时关联的 expectation 上限；一旦达到该上限，error3 计数器递增一次。

基于状态过滤会话

概述

`diagnose sys session filter` 命令提供了 `session-state1` 和 `session-state2` 两个参数，可以根据会话的状态标志对会话表进行精确过滤。

命令格式如下：

```bash
diagnose sys session filter <session-state1|session-state2> <state_bits> <state_bits_mask>
```

- `state_bits`：指定要过滤的状态位，多个状态可以通过相加（按位或）的方式组合使用。
- `state_bits_mask`：
  - 若设置为 `ffffffff`：只匹配 **仅包含**指定状态的会话（精确匹配）。
  - 若设置为与 `state_bits` 相同的值：匹配 **包含**指定状态的会话（含其他状态也可命中）。

session-state1 状态位

在命令行输入以下命令（不加参数），可查看 `session-state1` 支持的所有状态位：

```bash
FortiGate # diagnose sys session filter session-state1
```

```bash
Arguments: state_bits state_bits_mask

state1 bits:
|00000001: new              |00000002: redir            |00000004: log              |00000008: block
|00000010: oe               |00000020: re               |00000040: wccp             |00000080: dirty
|00000100: local            |00000200: may_dirty        |00000400: per_ip           |00000800: auth
|00001000: nb               |00002000: ndr              |00004000: nds              |00008000: br
|00010000: npu              |00020000: npd              |00040000: src-vis          |00080000: ssc
|00100000: nlb              |00200000: dst-vis          |00400000: 3way             |00800000: pol_sniff
|01000000: authed           |02000000: need_sync        |04000000: synced           |08000000: os
|10000000: rs               |20000000: ha_replicate     |40000000: ndri
```

提示

CLI 帮助中未显示的额外状态位：`80000000: EXPECT (f31)`

常用 `session-state1` 状态位说明：

状态位	状态名	说明
00000004	log	会话正在记录日志
00000080	dirty	会话参数已被修改
00000100	local	会话与 FortiGate 本地 IP 协议栈关联
00000200	may_dirty	会话参数允许被修改
00010000	npu	会话通过 NP 硬件加速芯片转发
00020000	npd	会话被拒绝进行硬件加速
00100000	nlb	会话来自负载均衡 VIP
00800000	pol_sniff	策略中启用抓包功能后可见
02000000	need_sync	会话将在存活 30 秒后同步到 HA 对端
04000000	synced	会话已同步至 HA 对端

session-state2 状态位

在命令行输入以下命令（不加参数），可查看 `session-state2` 支持的所有状态位：

```bash
FortiGate # diagnose sys session filter session-state2
```

```bash
Arguments: state_bits state_bits_mask

state2 bits:
|00000008: pcp_outbound      |00000010: pcp_inbound       |00000020: dym_src_port      |00000040: inherit_sockport
|00000100: netflow-origin    |00000200: netflow-reply     |00000400: syn_ses           |00000800: fec
|00001000: nosyn_ses         |00002000: csf_syncd_log     |00004000: app_valid         |00008000: url_cat_valid
|00010000: route_preserve    |00040000: exp_notify        |00080000: pkt_dup           |00100000: force_dup
|00200000: de_dup            |00400000: dynamic_shaping   |00800000: tcp_3way_rtt      |01000000: access_proxy
|02000000: svc_dup           |04000000: rpdb_dup
```

提示

CLI 帮助中未显示的额外状态位：

状态位	状态名	适用场景
00000001	LOG_FAILED_ATTEMPT	通用
00000002	LOG_FAILED_DNS	通用
00000004	LOG_FAILED_IP_CONN	通用
00020000	CLUSTER_SYNC	通用
10000000	LO_FWD	仅 SLBC 机箱设备
20000000	LI_REDIR	仅 SLBC 机箱设备
40000000	LO_MFPM_FWD	SLBC IPsec VPN 负载均衡
80000000	LI_MFPM_REDIR	SLBC IPsec VPN 负载均衡

过滤示例

示例一：精确匹配指定状态的会话

过滤仅包含 `may_dirty`、`nlb`、`nosyn_ses` 三个状态（不含其他状态）的会话：

- `may_dirty` 的状态位（session-state1）：`00000200`
- `nlb` 的状态位（session-state1）：`00100000`
- `nosyn_ses` 的状态位（session-state2）：`00001000`

```bash
diagnose sys session filter clear
diagnose sys session filter session-state1 00100200 ffffffff
diagnose sys session filter session-state2 00001000 ffffffff
diagnose sys session list
```

命令输出示例：

```
session info: slot=0 ori_slot=0 proto=6 proto_state=02 duration=0 expire=9 timeout=3600 flags=00000000 sockflag=00000000
sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=rsh vlan_cos=8/8
state=may_dirty nlb nosyn_ses
```

```
statistic(bytes/packets/allow_err): org=60/1/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=103->105/105->103 gwy=169.254.0.66/0.0.0.0
hook=pre  dir=org  act=noop 10.101.10.5:11971->172.168.17.2:514(0.0.0.0:0)
hook=post dir=reply act=noop 172.168.17.2:514->10.101.10.5:11971(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=4294967295 auth_info=0 chk_client_info=0 vd=1
serial=0117979c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 dd_type=0 dd_mode=0
npu_state=0x040000 no_ofld_reason:  non-npu-intf
total session 60
```

输出结果的 `state` 字段中可见 `may_dirty nlb nosyn_ses`，与过滤条件一致。

示例二：匹配包含某状态的所有会话

过滤所有包含 `log` 状态的会话（允许同时存在其他状态）：

- `log` 的状态位（session-state1）：`00000004`
- `state_bits_mask` 设置为与 `state_bits` 相同的值（`00000004`），表示模糊匹配

```bash
diagnose sys session filter clear
diagnose sys session filter session-state1 00000004 00000004
diagnose sys session list
```

命令输出示例：

```bash
session info: slot=0 ori_slot=0 proto=17 proto_state=00 duration=1179952 expire=139 timeout=0 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=8/8
state=log local may_dirty
statistic(bytes/packets/allow_err): org=1415664/19662/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 1/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->in, reply out->post dev=4->0/0->4 gwy=0.0.0.0/0.0.0.0
hook=pre  dir=org  act=noop 10.5.21.24:1950->10.5.31.255:8014(0.0.0.0:0)
hook=post dir=reply act=noop 10.5.31.255:8014->10.5.21.24:1950(0.0.0.0:0)
misc=0 policy_id=4294967295 auth_info=0 chk_client_info=0 vd=2
serial=00000872 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 dd_type=0 dd_mode=0
npu_state=00000000 no_ofld_reason:  local
total session 20
```

输出结果的 `state` 字段中包含 `log`，同时还存在 `local`、`may_dirty` 等其他状态标志。

Sniffer 工具

- sniffer 工具介绍
- Sniffer 过滤条件的合理使用
- 使用 Wireshark 打开抓包文件
- Wireshark 显示抓包接口和方向
- 通过 GUI 界面抓包
- NP7 芯片抓包

sniffer 工具介绍

FortiGate 具备完善的 sniffer 功能，可以基于指定接口或所有接口附带过滤条件进行抓包。

Sniffer 命令

```bash
diagnose sniffer packet <interface> <filter> <verbose> <count> <tsformat> <frame size>
```

- `<interface>`：可以指定一个具体的 interface 名字，比如“port1”或所有接口“any”。
- `<filter>`：非常强大的过滤功能，可以过滤出想要过滤的流，过滤条件需要引号括起来 `'<filter>'` 或者 `"<filter>"`，与 tcpdump 语法基本相同。

  ```bash
  [[src|dst] host {<host1_ipv4>}]
  [and|or] [[src|dst] host <host2_ipv4>}]
  [and|or] [[arp|ip|gre|esp|udp|tcp] port <port1_int>]
  [and|or] [[arp|ip|gre|esp|udp|tcp] port <port2_int>]
  ```

  - 过滤单个 IP（该 IP 可以是源，也可以是目的）：`'host 1.1.1.1'`
  - 过滤源 IP 和目的 IP：`'host 1.1.1.1 and host 2.2.2.2'`
  - 过滤目的 IP 和协议端口：`'dst 2.2.2.2 and tcp port 80'`
  - 过滤源 IP 和协议：`'src 1.1.1.1 and icmp'`
- `<verbose>`：不同级别的不同内容展示，按需定义 sniffer 级别。
  - `1`：print header of packets（只输出报文头部内容）
  - `2`：print header and data from ip of packets（输出从 ip 头部开始，包含数据内容）
  - `3`：print header and data from ethernet of packets（输出从 ethernet 头部开始，包含数据内容）
  - `4`：print header of packets with interface name（只输出报文头部和进出接口，使用最多）
  - `5`：print header and data from ip of packets with interface name（输出从 IP 头部开始，包含数据内容和进出接口）
  - `6`：print header and data from ethernet of packets (if available) with intf name（从 ethernet 头部开始，包含数据内容和进出接口，使用较多）
- `<count>`：抓包个数，到达指定个数后停止 sniffer，如 100 表示抓 100 个包，填写为 0 表示不限制抓包的数量。
- `<tsformat>`：时间戳格式。
  - `a`：UTC 绝对时间，格式为 `yyyy-mm-dd hh:mm:ss.ms`
  - `l`：本地时间，`yyyy-mm-dd hh:mm:ss.ms`
  - 不填写：开始抓包后的相对时间，`ss.ms`
- `<frame size>`：设置截断前帧的大小，默认是接口 MTU。

Sniffer 示例

- 在 port1 上抓取源或目的 IP 为 172.16.1.100 的包，只输出报文头部内容。

  ```bash
  diagnose sniffer packet port1 "host 172.16.1.100" 1
  ```
- 在所有接口上抓取源或目的 IP 为 172.16.1.100 的 ICMP 报文，输出报文头部和进出接口。

  ```bash
  diagnose sniffer packet any "host 172.16.1.100 and icmp" 4
  ```
- 在所有接口上抓取源或目的 IP 为 172.16.1.100 的 TCP 报文，输出报文头部和进出接口。

  ```bash
  diagnose sniffer packet any "host 172.16.1.100 and tcp" 4
  ```
- 在所有接口上抓取源或目的 IP 为 172.16.1.100 的 UDP 报文，输出报文头部和进出接口。

  ```bash
  diagnose sniffer packet any "host 172.16.1.100 and udp" 4
  ```
- 在所有接口上抓取源或目的 IP 为 172.16.1.100，端口号为 443 的 TCP 报文，输出报文头部和进出接口。

  ```bash
  diagnose sniffer packet any "host 172.16.1.100 and tcp port 443" 4
  ```

- 在所有接口上抓取源或目的 IP 为 172.16.1.100 且 TCP 端口≠22 的报文， 输出报文头部和进出接口。

```
diagnose sniffer packet any "host 172.16.1.100 and !tcp port 22" 4
```
bash

- 在所有接口上抓取源或目的 IP 为 172.16.1.100 且 TCP/UDP 端口为 10000 的报文， 输出报文头部和进出接口。

```
diagnose sniffer packet any "host 172.16.1.100 and port 10000" 4
```
bash

- 在所有接口上抓取源或目的 IP 为 114.114.119.119 且协议为 ICMP 的报文，携带 0 表示不限制抓包个数，携带 a 表示携带抓包的时间戳，时间戳的格式为抓包开始后的相对时间。

```
diagnose sniffer packet any "host 114.114.119.119 and icmp" 4 0 a
```
bash

- 在所有接口上抓取目的 IP 为 114.114.119.119 且协议为 ICMP 的报文，携带 0 表示不限制抓包个数，携带 l 表示携带抓包的时间戳，时间戳格式为绝对时间，与设备上的时间一致。

```
diagnose sniffer packet any "dst 114.114.119.119 and icmp" 4 0 l
```
bash

- 在 port1 接口上抓取源 IP 为 10.1.1.1，源或目的端口号为 80，并携带 VLAN Tag 10 的报文。

```
diagnose sniffer packet port1 'vlan 10 and src 10.1.1.1 and port 80' 4
```
bash

- 在 port1 上抓取 VLAN tag 为 3001 且 IP host 为 192.168.1.1 的 VLAN trunk 报文，或 VLAN tag 为 3002 且 IP host 为 192.168.2.1 的 VLAN trunk 报文。

```
diagnose sniffer packet port1 "vlan and ((ether[14:2] == 3001 and host 192.168.1.1) or (ether[14:2] == 3002 and host 192.168.2.1))" 4 0 l
```
bash

- 在所有接口上抓取（源或目的 IP 为 172.16.1.100）或（源或目的 IP 为 172.16.1.101 且端口号为 1701）的报文。

```
diagnose sniffer packet any "host 172.16.1.100 or (host 172.16.1.101 and port 1701)" 4
```
bash

- 在 internal 接口上抓取 SYN 置位的 TCP 报文。

```
diagnose sniffer packet internal "tcp[13]&2≠0 and port 23" 4
```
bash

- 在 internal 接口上抓取 SYN - ACK 置位的 TCP 报文。

```
diagnose sniffer packet internal "tcp[13] = 18" 4
```
bash

- 在 internal 接口上抓取 RST 置位的 TCP 报文。

```
diagnose sniffer packet internal "tcp[13]&4≠0" 4
```
bash

- 在 wan1 接口上抓取 PPPoE 协商报文（其中 0x8863 发现阶段，0x8864 为会话阶段）。

```
diagnose sniffer packet wan1 "ether proto 0×8863 or ether proto 0×8864" 4
```
bash

- 在 internal 接口上抓取源或目 IP 属于网段为 4.2.2.0/24 的所有报文。

```
diagnose sniffer packet internal "net 4.2.2.0/24" 4
```
bash

- 在 internal 接口上抓取源 MAC 地址为 00:09:0f:89 的报文。

```
diagnose sniffer packet internal "(ether[6:4]=0×00090f89) and (ether[10:2]=0×10ea)" 4
```
bash

- 在所有接口上抓取 IP TTL 为 1 的报文。

```
diagnose sniffer packet any "ip[8:1] = 0×01" 4
```
bash

- 在 internal 接口上抓取特定 Payload 的报文。

```
Payload:
0×0040   8f48 8a0c 7468 6973 2069 7320 7069 6e67    .H..this.is.ping

Filter:
diagnose sniffer packet internal "ether[0×44:4]=0×74686973" 4
```
bash

- 在所有接口上抓取大于 1500 Bytes（除去二层头部长度）的报文。

```
diagnose sniffer packet any "ip[2:2] > 1500" 4
```
bash

- 抓取三层组播报文。

```
diagnose sniffer packet port1 'ip multicast' 6 0 l
```
bash

- 抓取二层组播报文。

```
diagnose sniffer packet port1 'ether multicast' 6 0 l
```
bash

- 抓取三层广播报文。

```
diagnose sniffer packet port1 'ip broadcast' 6 0 l
```
bash

- 抓取二层广播报文。

```
diagnose sniffer packet port1 'ether broadcast' 6 0 l
```
bash

- 抓取二层和三层组播。

```
diagnose sniffer packet port1 'multicast' 6 0 l
```
bash

Sniffer 过程分析

ICMP

1. ICMP 的请求报文是 echo request，响应报文是 echo reply，in 表示 FortiGate 收到一个报文，out 表示 FortiGate 发出一个报文。在 FortiGate 上抓取 ICMP 报文。

```
FortiGate # diagnose sniffer packet any 'host 192.168.2.10 and icmp' 4
interfaces=[any]
filters=[host 192.168.2.10 and icmp]
```
bash

2. FortiGate 从 port5 收到 192.168.1.10 发送给 192.168.2.10 ping 的请求报文 echo request。

```
7.074724 port5 in 192.168.1.10 → 192.168.2.10: icmp: echo request
```
bash

3. FortiGate 将 192.168.1.10 发送给 192.168.2.10 ping 的请求报文 echo request 从 port6 转发出去了。

```
7.074748 port6 out 192.168.1.10 → 192.168.2.10: icmp: echo request
```
bash

4. FortiGate 从 port6 收到 192.168.2.10 发送给 192.168.1.10 ping 的响应报文 echo reply。

```
7.075033 port6 in 192.168.2.10 → 192.168.1.10: icmp: echo reply
```
bash

5. FortiGate 将 192.168.2.10 发送给 192.168.1.10 ping 的响应报文 echo reply 从 port5 转发出去了。

```
7.075042 port5 out 192.168.2.10 → 192.168.1.10: icmp: echo reply
```
bash

6. 从上面的抓包来看，FortiGate 成功转发了 192.168.1.10 和 192.168.2.10 之间 ping 的请求和响应报文。

TCP

1. TCP 报文交互之前都需要完成 3 次握手，in 表示 FortiGate 收到一个报文，out 表示 FortiGate 发出一个报文。在 FortiGate 上抓取 TCP 报文。

```
FortiGate # diagnose sniffer packet any 'host 192.168.2.10 and port 22' 4
interfaces=[any]
filters=[host 192.168.2.10 and port 22]
```
bash

2. FortiGate 从 port5 收到 192.168.1.10 发送给 192.168.2.10 的 syn 报文。

```
12.009383 port5 in 192.168.1.10.46028 → 192.168.2.10.22: syn 486804292
```
bash

3. FortiGate 将 192.168.1.10 发送给 192.168.2.10 的 syn 报文从 port6 转发出去了。

```bash
12.009752 port6 out 192.168.1.10.46028 → 192.168.2.10.22: syn 486804292
```

4. FortiGate 从 port6 收到 192.168.2.10 发送给 192.168.1.10 的 syn ack 报文。

```bash
12.009957 port6 in 192.168.2.10.22 → 192.168.1.10.46028: syn 1279904517 ack 486804293
```

5. FortiGate 将 192.168.2.10 发送给 192.168.1.10 的 syn ack 报文从 port5 转发出去了。

```bash
12.010120 port5 out 192.168.2.10.22 → 192.168.1.10.46028: syn 1279904517 ack 486804293
```

6. FortiGate 从 port5 收到 192.168.1.10 发送给 192.168.2.10 的 ack 报文。

```bash
12.010233 port5 in 192.168.1.10.46028 → 192.168.2.10.22: ack 1279904518
```

7. FortiGate 将 192.168.1.10 发送给 192.168.2.10 的 ack 报文从 port6 转发出去了。

```bash
12.010245 port6 out 192.168.1.10.46028 → 192.168.2.10.22: ack 1279904518
```

8. 从上面的抓包来看，FortiGate 成功转发了 192.168.1.10 和 192.168.2.10 之间 tcp 三次握手的报文。

Sniffer 过滤条件的合理使用

过滤条件使用总结

Sniffer 应该抓取一条流的五元组不变的那个参数，且过滤的条件越精确抓取的报文越准确。

- 在没有 NAT 场景下，源目 IP 及目的端口都不会变化，则过滤条件可以设置为 `host <src IP> and host <dst IP> and <dst port>`。
- 在源 NAT 场景下，源 IP 会做 NAT 转换，因此目的 IP 和目的端口是不会变化的，则过滤条件可以设置为 `host <dst IP> and <dst port>`。
- 在目的 NAT 场景下，目的 IP 会做 NAT 转换，因此源 IP 和目的端口是不会变化的，则过滤条件可以设置为 `host <src ip> and <dst port>`。
- 在目的端口 NAT 场景下，目的 IP 和端口都会做 NAT 转换，因此源 IP 是不会变化，则过滤条件可以设置为 `host <src IP>`。
- 在源 NAT 和目的 NAT 都存在的场景下，源 IP 和目的 IP 都会发生变化，则只能通过 `or` 参数分别抓取 NAT 前和 NAT 后的报文，过滤条件可以设置为 `(<源NAT前的源IP> and <目的NAT前的目的IP> and <目的NAT前目的端口>) or (<源NAT后的源IP> and <目的NAT后的目的IP> and <目的NAT后的目的端口>)`。

Sniffer 过滤条件使用举例

> **重要**
>
> Ping、ESP、GRE 是没有端口的，直接使用 icmp、esp、gre 协议过滤即可。

普通三层转发，没有 NAT 场景

1. 普通三层转发，没有 NAT 的情况下，抓 `host <src IP> and host <dst IP> and <dst port>` 可以抓到完整的数据流。

2. 拓扑：PC1 (172.16.1.100)-----(port9：172.16.1.1) FGT (port10:202.106.1.1)-----PC2 (202.106.1.100)。

3. 抓包结果如下。

```bash
FGT # diagnose sniffer packet  any "host 172.16.1.100 and host 202.106.1.100 and icmp" 4
interfaces=[any]
filters=[host 172.16.1.100 and icmp]
3.583522 port9 in 172.16.1.100 → 202.106.1.100: icmp: echo request
3.583541 port10 out 172.16.1.100 → 202.106.1.100: icmp: echo request
3.583808 port10 in 202.106.1.100 → 172.16.1.100: icmp: echo reply
3.583818 port9 out 202.106.1.100 → 172.16.1.100: icmp: echo reply
```

普通三层转发，有源 NAT 场景

1. 普通三层转发，有源 NAT 的情况下，抓 `host <dst IP> and <dst port>` 可以抓到完整的数据流。

2. 拓扑：PC1 (172.16.1.100)-----(port9：172.16.1.1) FGT (port10:202.106.1.1)-----PC2 (202.106.1.100)。

3. 抓包结果如下。

```bash
FGT # diagnose sniffer packet  any “host 172.16.1.100 and icmp” 4
//抓源IP and icmp协议，是抓不全的//
interfaces=[any]
filters=[host 172.16.1.100 and icmp]
2.821403 port9 in 172.16.1.100 → 202.106.1.100: icmp: echo request
2.821707 port9 out 202.106.1.100 → 172.16.1.100: icmp: echo reply    //只能在port9看到数据包in/out，port10的看不到//

FGT # diagnose sniffer packet  any "host 202.106.1.100 and icmp" 4
//需要抓目的IP and icmp协议//
interfaces=[any]filters=[host 202.106.1.100 and icmp]
4.624647 port9 in 172.16.1.100 → 202.106.1.100: icmp: echo request
4.625032 port10 out 202.106.1.1 → 202.106.1.100: icmp: echo request
4.625235 port10 in 202.106.1.100 → 202.106.1.1: icmp: echo reply
4.625243 port9 out 202.106.1.100 → 172.16.1.100: icmp: echo reply
```

普通三层转发，有目的 NAT（一对一 NAT）场景

1. 普通三层转发，有目的 NAT 的情况下，抓 `host <src ip> and <dst port>` 可以抓到完整的数据流。

2. 拓扑：PC1 (172.16.1.100)-----(port9：172.16.1.1) FGT (port10:202.106.1.1)-----PC2 (202.106.1.100)。

3. 访问 VIP（DNAT），流量从外网侧发起，抓包结果如下。

```bash
FGT # diagnose sniffer packet any “host 202.106.1.10” 4
//抓目的IP and icmp协议，是抓不全的//
interfaces=[any]
filters=[host 202.106.1.10]
3.809005 port10 in 202.106.1.100 → 202.106.1.10: icmp: echo request
3.809294 port10 out 202.106.1.10 → 202.106.1.100: icmp: echo reply    //只能在port10看到数据包in/out，port9的看不到//

FGT # diagnose sniffer packet any “host 202.106.1.100” 4
//需要抓源IP and icmp协议//
interfaces=[any]
filters=[host 202.106.1.100]
3.523666 port10 in 202.106.1.100 → 202.106.1.10: icmp: echo request
3.523687 port9 out 202.106.1.100 → 172.16.1.100: icmp: echo request
3.523937 port9 in 172.16.1.100 → 202.106.1.100: icmp: echo reply
3.523943 port10 out 202.106.1.10 → 202.106.1.100: icmp: echo reply
```

普通三层转发，有双向 NAT (源 NAT + 目的 NAT）场景

1. 普通三层转发，有双 NAT 的情况下，抓 `(<源NAT前的源IP> and <目的NAT前的目的IP> and <目的NAT前目的端口>) or (<源NAT后的源IP> and <目的NAT后的目的IP> and <目的NAT后的目的端口>)` 可以抓到完整的数据流。

2. 拓扑：PC1 (172.16.1.100)-----(port9：172.16.1.1) FGT (port10:202.106.1.1)-----PC2 (202.106.1.100)。

3. 访问 VIP，流量从外网侧发起，同时在内网接口侧开启了 SNAT，抓包结果如下。

```bash
FGT # diagnose sniffer packet any "host 202.106.1.100 and icmp" 4
//只抓源IP是抓不全的//
interfaces=[any]
filters=[host 202.106.1.100 and icmp]
0.416187 port10 in 202.106.1.100 → 202.106.1.10: icmp: echo request
0.416576 port10 out 202.106.1.10 → 202.106.1.100: icmp: echo reply

FGT # diagnose sniffer packet any "host 172.16.1.100 and icmp" 4
//只目的IP也是抓不全的//
interfaces=[any]
filters=[host 172.16.1.100 and icmp]
1.417838 port9 out 172.16.1.1 → 172.16.1.100: icmp: echo request
1.418326 port9 in 172.16.1.100 → 172.16.1.1: icmp: echo reply

FGT # diagnose sniffer packet any "host 202.106.1.100 or host 172.16.1.1 and icmp" 4
//使用or组合，变化前源目的ip和变化后源目的IP可抓全//
interfaces=[any]
filters=[host 202.106.1.100 or host 172.16.1.1 or icmp]
0.419407 port10 in 202.106.1.100 → 202.106.1.10: icmp: echo request
0.419436 port9 out 172.16.1.1 → 172.16.1.100: icmp: echo request
0.419781 port9 in 172.16.1.100 → 172.16.1.1: icmp: echo reply
0.419794 port10 out 202.106.1.10 → 202.106.1.100: icmp: echo reply
```

抓取 PPPoE 报文

1. PPPoE Discovery 阶段（协商）的的 ethernet 类型为 `0x8863`，PPPoE Session 阶段（数据）的 ethernet 类型为 `0x8864`。

```bash
diagnose sniffer packet <PPPoe Interface> 'ether proto 0×8863 or ether proto 0×8864' 6 0 l
```

2. 也可以使用 `pppoed` 和 `pppoes` 参数来作为过滤条件。

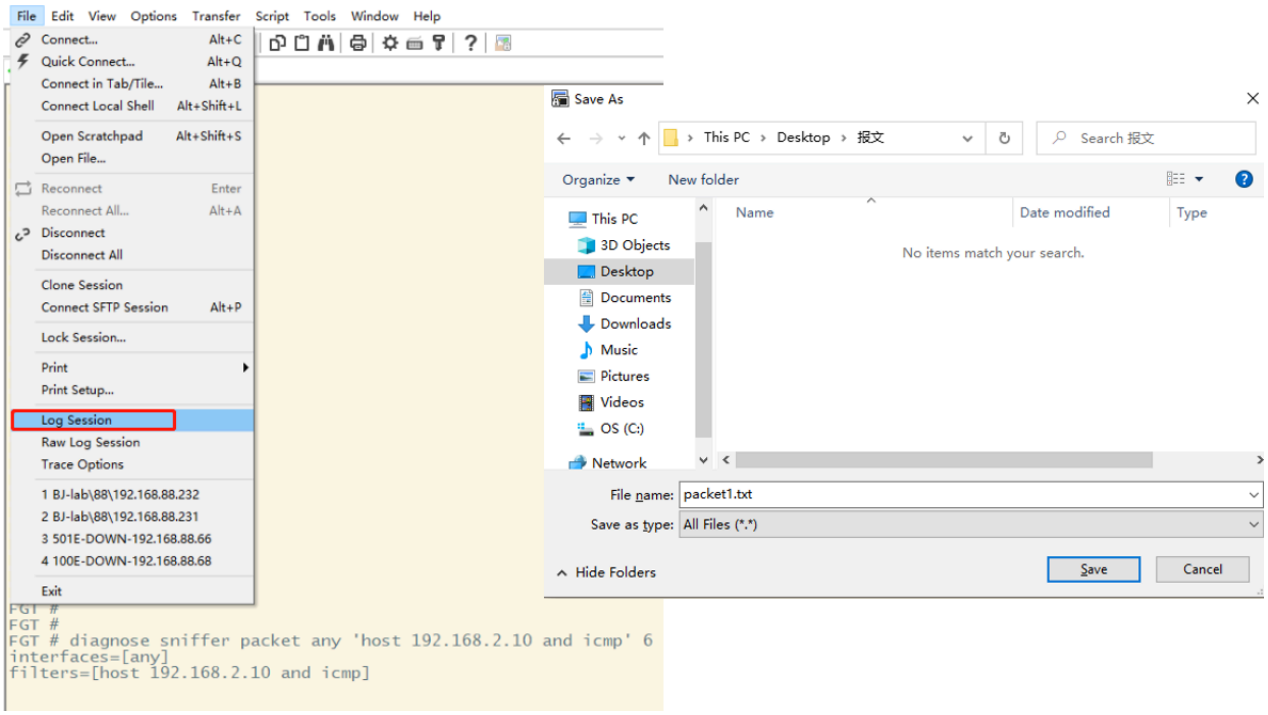
```bash
diagnose sniffer packet any 'pppoed or pppoes' 6 0 l
```

使用 Wireshark 打开抓包文件

通过抓包转换工具 ConvertPcap（点击此处下载 ConvertPcap ↗），我们可以将使用 Sniffer 格式 6 抓到的报文转换为 Wireshark 可以打开的文件格式。

> **重要**
>
> 除了使用 ConvertPcap 工具，还可以使用 sniftran 工具，该工具可以在 Wireshark 中显示报文的出接口和入接口。请参考“故障排查 → Sniffer 工具 → Wireshark 显示抓包接口和方向 ↗”章节。

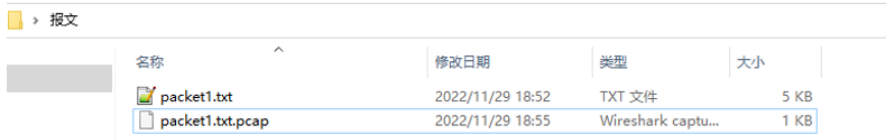
1. 通过 `diagnose sniffer packet any "xxxx" 6 0 l` 抓取报文内容，保存为 txt 文本格式。

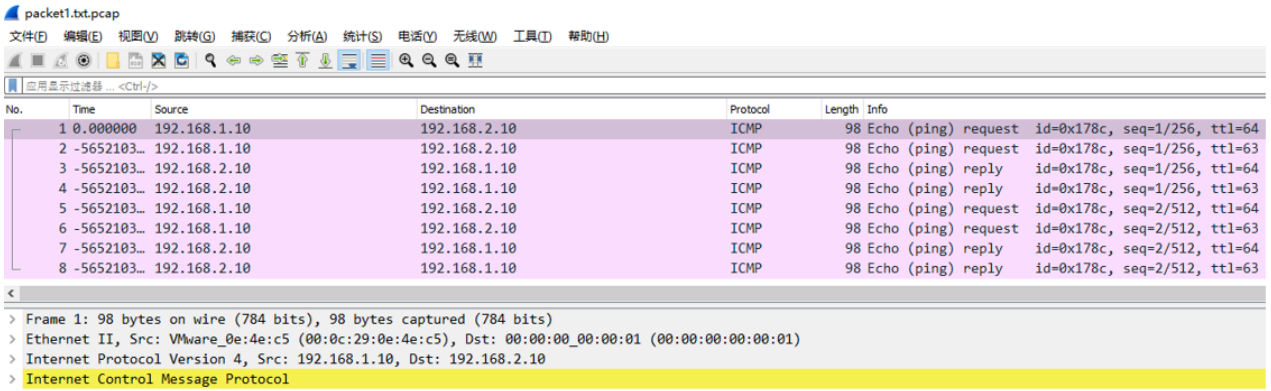


2. 安装“ConvertPcap.exe”，然后右击 txt 文本格式的报文，选择 Convert to PACP 完成数据格式的转换。

3. 安装完成后，右键点击要转换格式的抓包 txt 文件，点击 Convert to PCAP，即可在当前目录自动生成转换后的 Wireshark 可打开的抓包文件。


报文
名称
修改日期
类型
大小
packet1.txt
TXT 文件
5 KB
打开(O)
编辑(E)
使用 Skype 共享
Convert to PCAP


报文
名称
修改日期
类型
大小
packet1.txt
2022/11/29 18:52
TXT 文件
5 KB
packet1.txt.pcap
2022/11/29 18:55
Wireshark captu...
1 KB

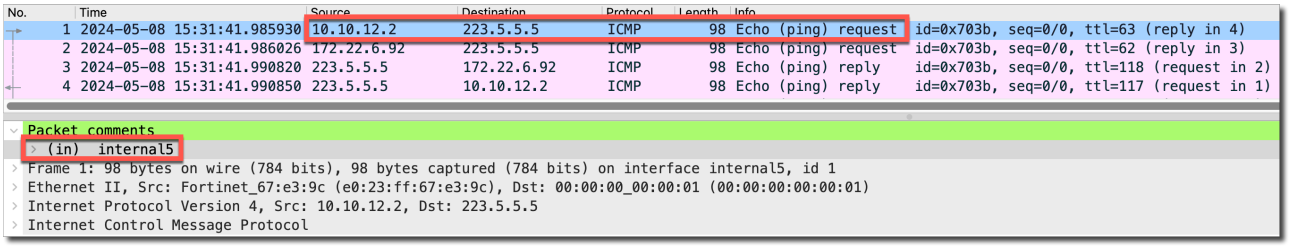

packet1.txt.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) request id=0x178c, seq=1/256, ttl=64
2 -5652103… 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) request id=0x178c, seq=1/256, ttl=63
3 -5652103… 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) reply id=0x178c, seq=1/256, ttl=64
4 -5652103… 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) reply id=0x178c, seq=1/256, ttl=63
5 -5652103… 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) request id=0x178c, seq=2/512, ttl=64
6 -5652103… 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) request id=0x178c, seq=2/512, ttl=63
7 -5652103… 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) reply id=0x178c, seq=2/512, ttl=64
8 -5652103… 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) reply id=0x178c, seq=2/512, ttl=63
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: VMware_0e:4e:c5 (00:0c:29:0e:4e:c5), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.2.10
Internet Control Message Protocol

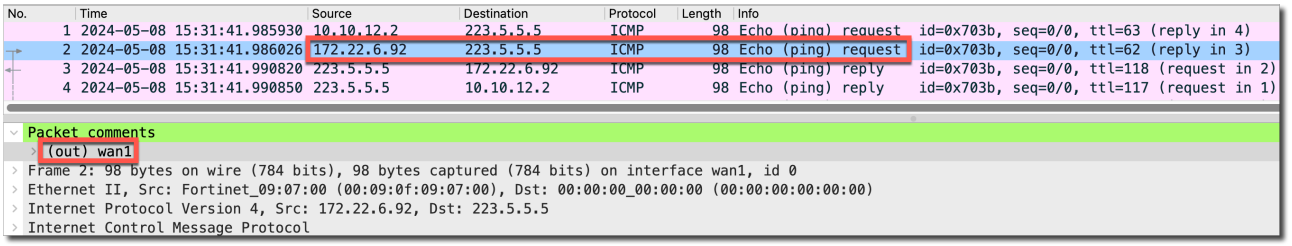
Wireshark 显示抓包接口和方向

工具介绍

在“故障排查 → Sniffer 工具 → 使用 Wireshark 打开抓包文件”章节中，使用抓包转换工具 ConvertPcap 可以将 Sniffer 格式 6 抓到的文本转换为 Wireshark 可以打开的文件格式，但转换后的 Wireshark 格式中没有报文的接口与方向（in/out）信息。

如果需要同时将 Sniffer 格式 6 抓到的文本中的接口与方向（in/out）信息同步到转换后的 Wireshark 文件中，可以使用 SnifTran 工具，转换后的抓包文件如下所示，可以显示每个报文的接口与方向（in/out）信息。





SnifTran 工具下载

本地下载

sniftran-master.zip

GitHub 下载

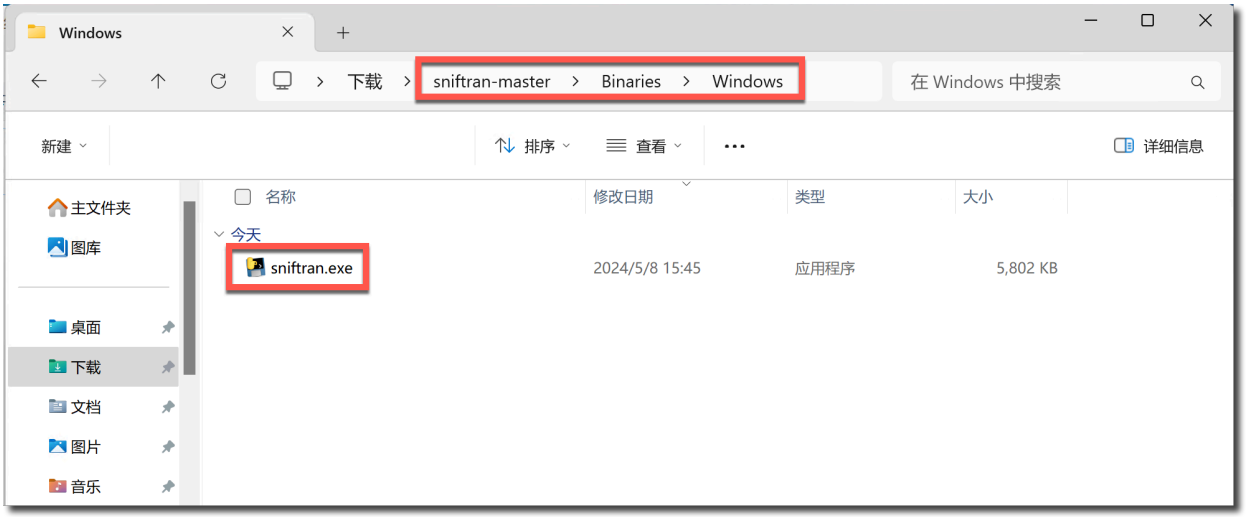
https://github.com/ondrejholecek/sniftran

使用方法

提前准备好使用 Sniffer 格式 6 抓到的文本文件。

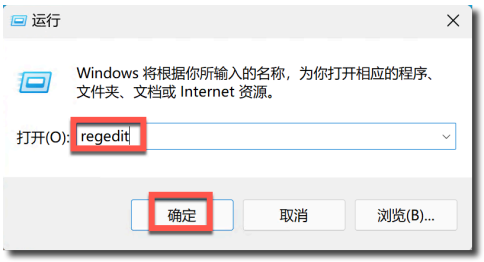
Windows

1. 下载 `sniftran-master.zip` 文件并解压，在 `sniftran-master/Binaries/Windows` 文件夹中，将 `sniftran.exe` 文件复制到一个特定目录下。



2. 这里将 `sniftran.exe` 文件复制到了 `D:\sniftran\` 目录下。

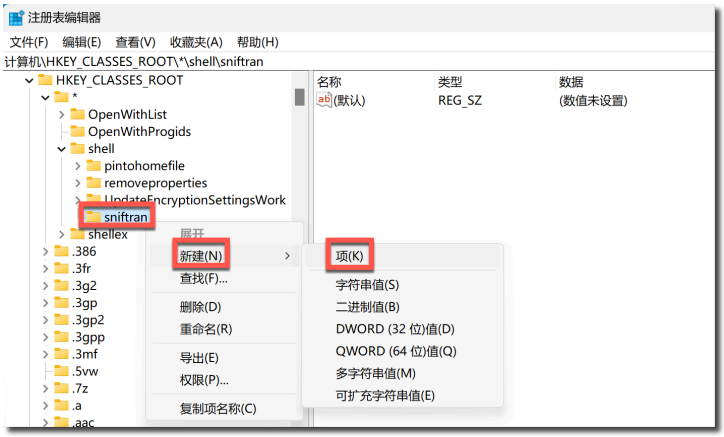
3. 使用 win + R 键打开运行窗口，输入 regedit，然后点击“确定”按钮，打开注册表。



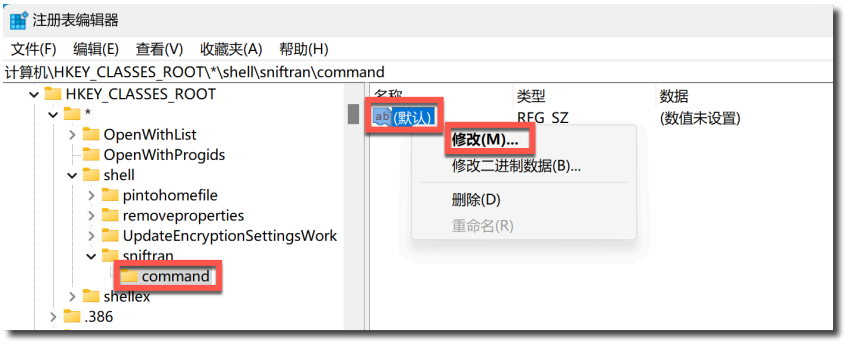
4. 找到 `计算机\HKEY_CLASSES_ROOT\*\shell` 路径，右键 `shell` 文件夹，点击“新建 (N)” → “项 (K)”。



5. 为新的“项”命名 `sniftran` （或其他名称）。右键新创建的 `sniftran` 文件夹，点击“新建 (N)” → “项 (K)”。

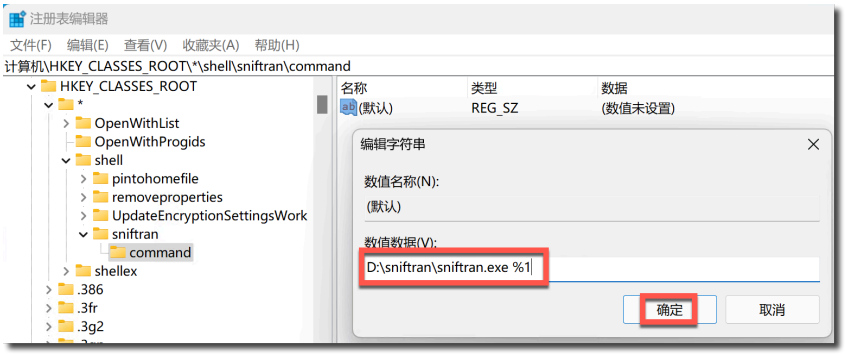


6. 为新的“项”命名 `command` ，选中 `command` 文件夹，在右侧窗口中右键点击“默认”条目，点击“修改 (M)...”按钮。

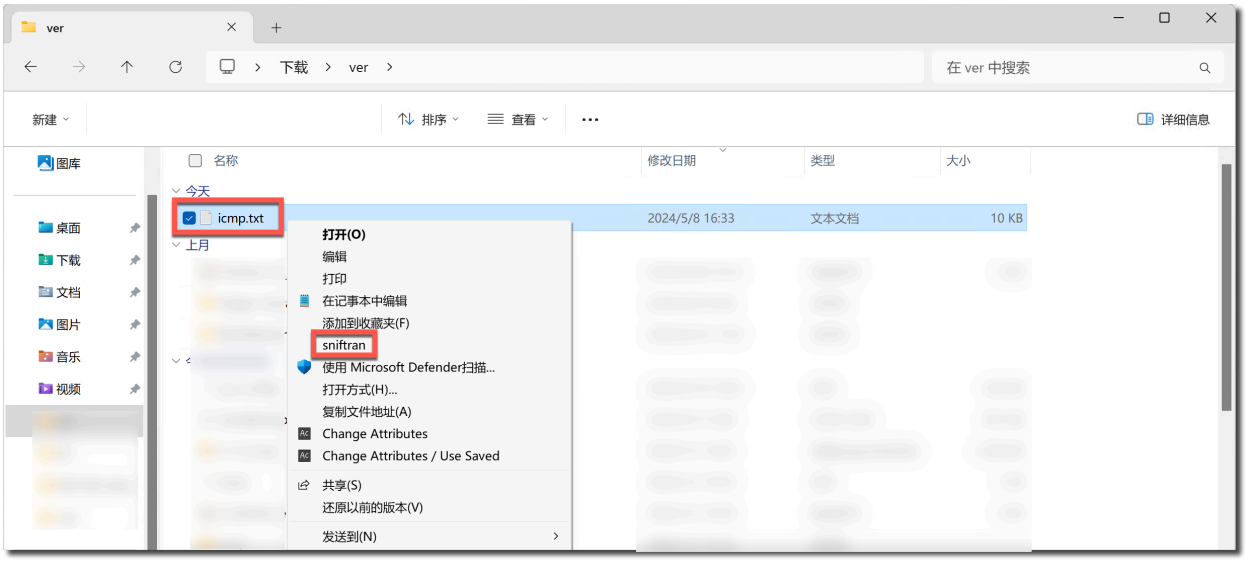


7. 在“编辑字符串”窗口中的“数值数据 (V)”中填入 `D:\sniftran\sniftran.exe %1`，然后点击“确定”按钮。

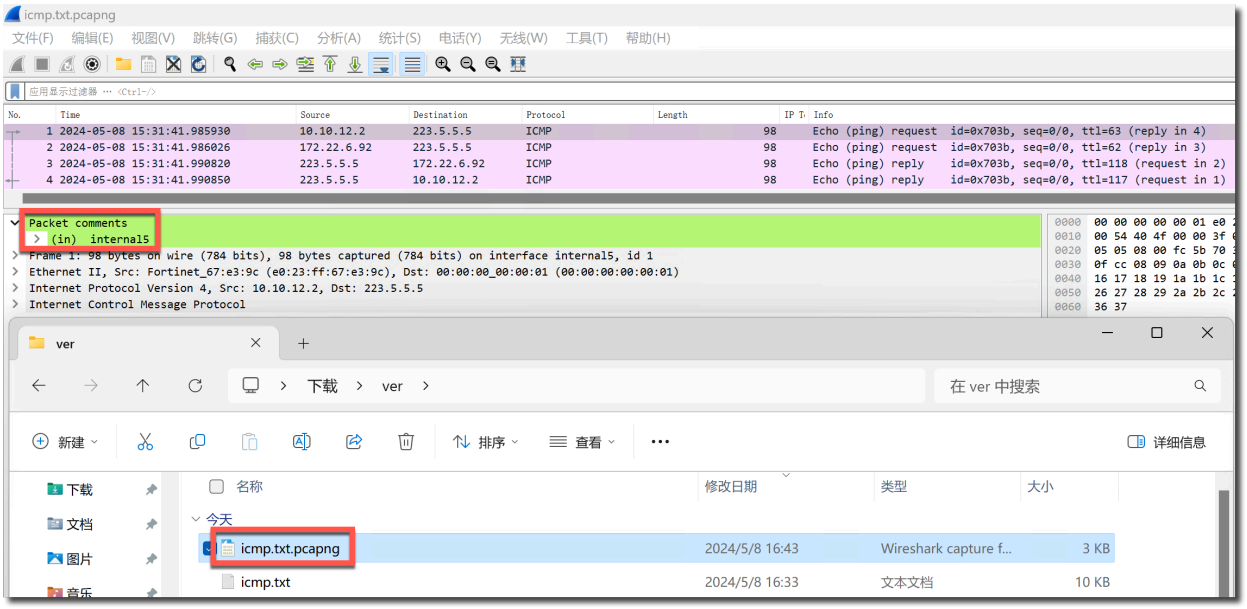
> **重要**
>
> `D:\sniftran\sniftran.exe` 是 sniftran.exe 文件所在的路径，请根据您自己文件的路径修改。



8. 右键点击需要转换格式的抓包文本文件，点击 `sniftran` 按钮（Windows 11 需要先点击“显示更多选项”）。

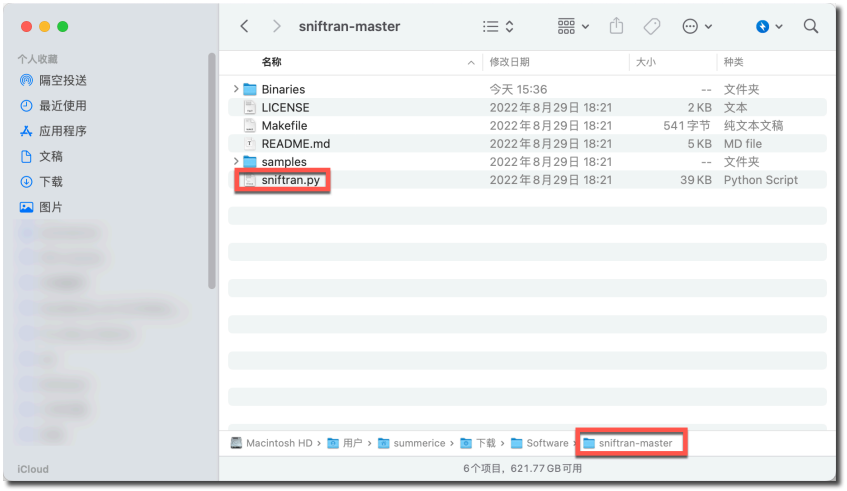


9. 随后会在当前目录下生成转换后的抓包文件，使用 Wireshark 打开后，可以显示每个报文的接口与方向（in/out）信息。



macOS

1. 由于 macOS 预装了 Python，可以直接使用 SnifTran 工具中的 `sniftran.py` 脚本执行转换。下载 sniftran-master.zip 文件并解压，在 `sniftran-master` 文件夹中，将 `sniftran.py` 文件复制到一个特定目录下。



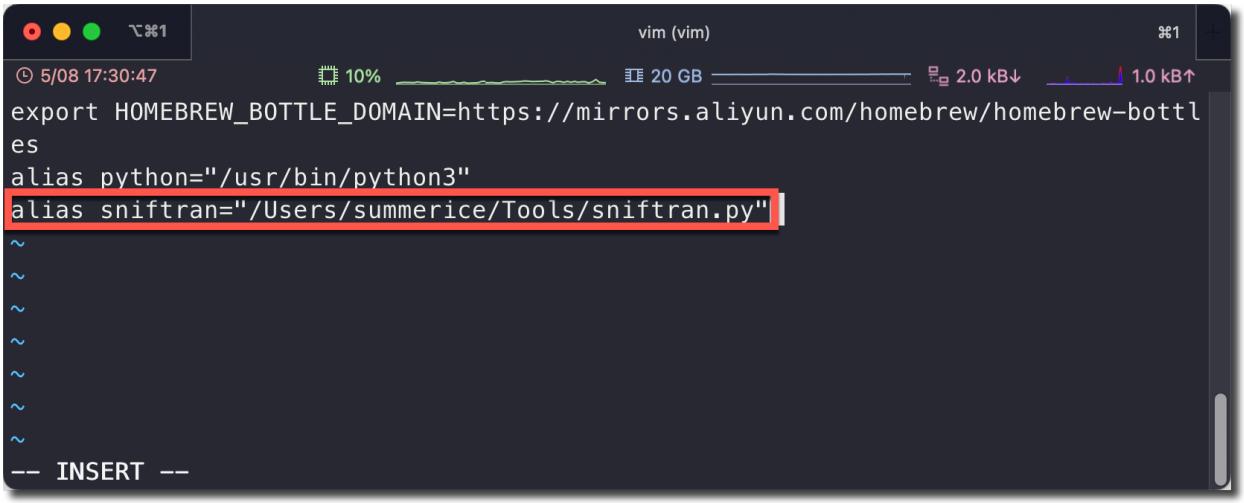
2. 这里将 `sniftran.py` 文件复制到了 `/Users/summerice/Tools` 目录下。

3. 将 `sniftran.py` 文件加入系统变量，在终端中编辑 `~/.bash_profile` 文件。

> **重要**
>
> 如果系统终端使用的是 zsh，请编辑 `~/.zshrc` 文件。

```bash
vim ~/.bash_profile
```

4. 添加一行 `alias sniftran="/Users/summerice/Tools/sniftran.py"`，其中 `sniftran` 为系统变量名称，可以自行定义，`"/Users/summerice/Tools/sniftran.py"` 为 `sniftran.py` 文件所在路径，随后保存该文件。

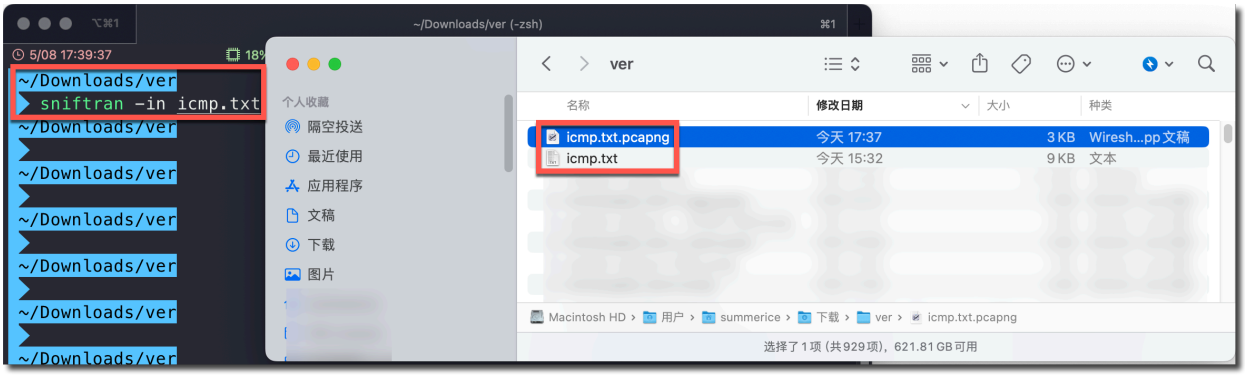


5. 使用 `source` 命令应用系统变量的修改。

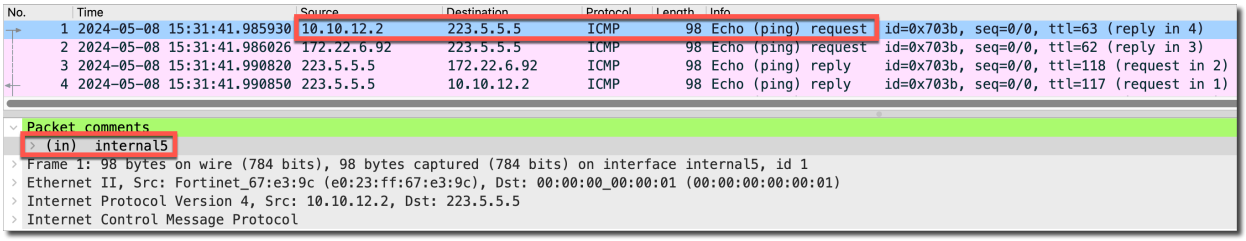
```bash
source ~/.bash_profile
```

> **重要**
>
> 如果系统终端使用的是 zsh，请使用 `source ~/.zshrc` 文件。

6. 在终端中进入需要转换格式的抓包文本文件所在目录，使用命令 `sniftran -in <file_name>` 将文本转换为 Wireshark 可以打开的抓包文件。



7. 随后会在当前目录下生成转换后的抓包文件，使用 Wireshark 打开后，可以显示每个报文的接口与方向（in/out）信息。



通过 GUI 界面抓包

功能介绍

除了通过在 CLI 下捕获数据包，FortiGate 也支持在 GUI 界面捕获数据包，捕获的数据包下载后可以直接用 Wireshark 打开（PCAP 格式），方便不熟悉 Sniffer 工具的用户使用。

使用方法

重要

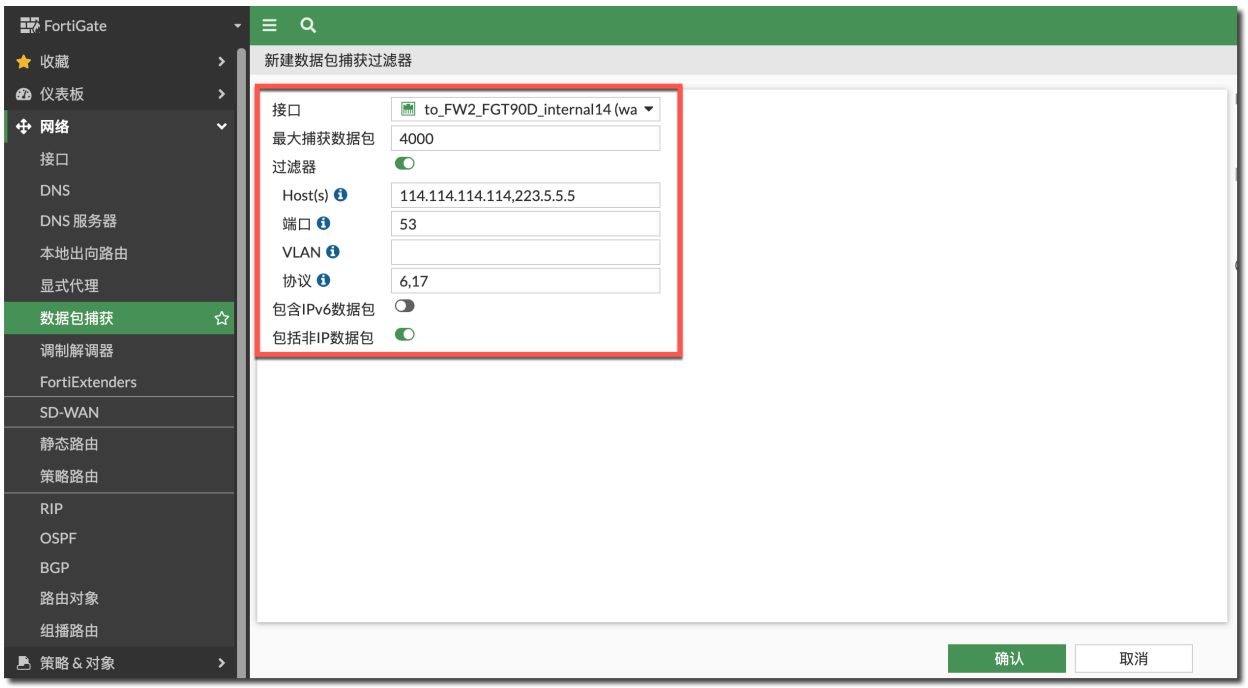
抓包之前，必须在抓取流量对应的防火墙策略上禁用 NPU 硬件加速才能查看所有数据包，请在 CLI 中将对应防火墙策略中的“auto-asic-offload”设置为“disable”。详情参考：故障排查 → Sniffer、Debug Flow 与 NP 加速的矛盾 ↗。

FortiOS 7.0

1. 进入“网络 → 数据包捕获”页面，点击“新建”按钮。



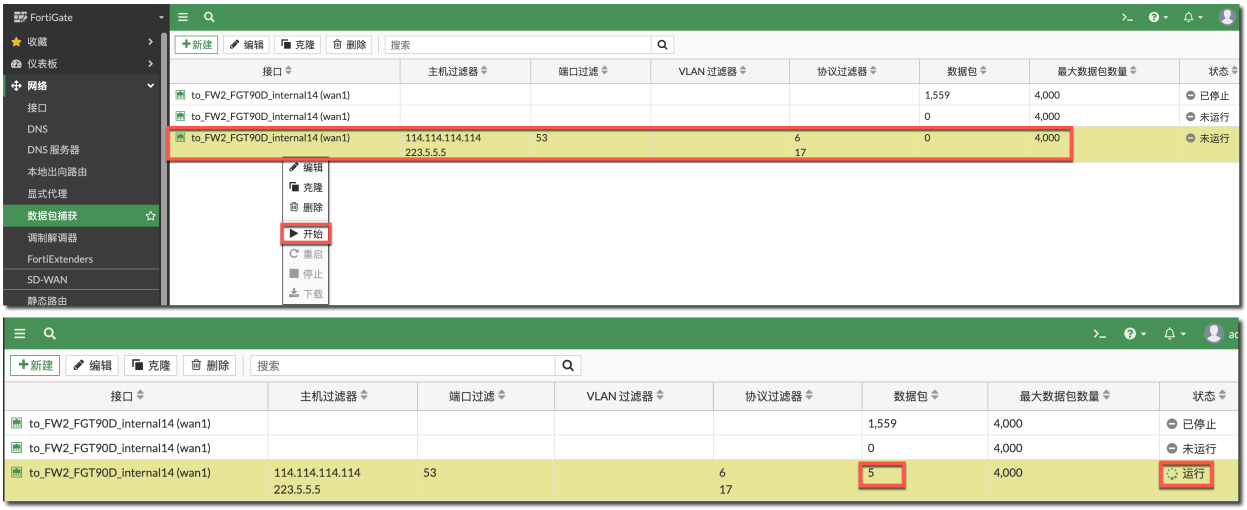
2. 设置捕获数据包的条件，并点击确认下发配置。



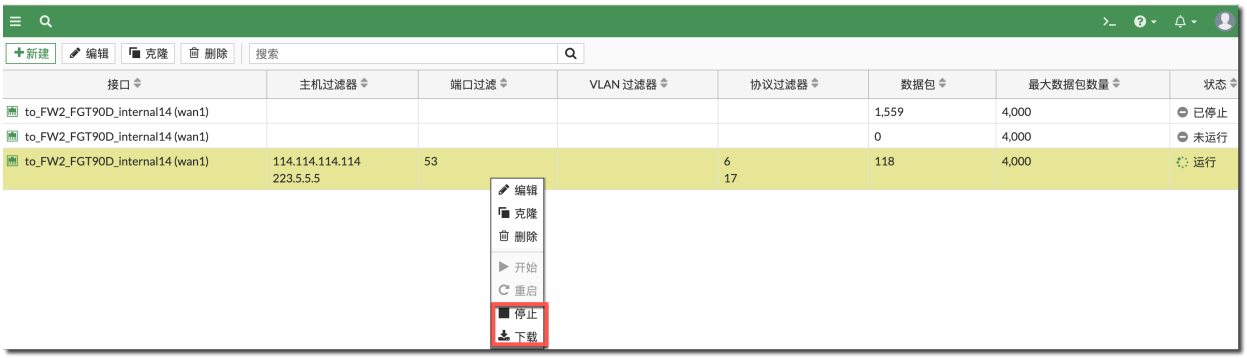
- 接口：指定抓取数据包的接口，无法指定 any，如果接口开启了 SNAT，抓取到的为 SNAT 后的数据包。
- 最大捕获数据包：每个数据包捕获可以抓取的最大数据包个数，达到该数字后，抓包自动停止（不同型号之间的最大捕获数量可能不同）。
- Host：抓取数据包过滤的源或目的 IP 地址/网段，可以填写多个 IP 地址/网段（网段的格式为 192.168.1.0/24），中间用英文逗号（,）隔开，它们之间是或的关系。
- 端口：过滤抓取数据包的端口号，可填写多个，中间用英文逗号（,）隔开，它们之间是或的关系。

- VLAN：如果要过滤抓取的数据包的 VLAN ID，可以配置此选项，可填写多个，中间用英文逗号（,）隔开，它们之间是或的关系。
- 协议：过滤抓取数据包的协议号，可填写多个，中间用英文逗号（,）隔开，它们之间是或的关系。
- 包含 IPv6 数据包：默认只抓取 IPv4 数据包，需要抓取 IPv6 时，开启此选项。
- 包括非 IP 数据包：默认只抓取包含 IP 头的数据包，如果想抓取不含 IP 头的数据包（如 ARP 报文），开启此选项。

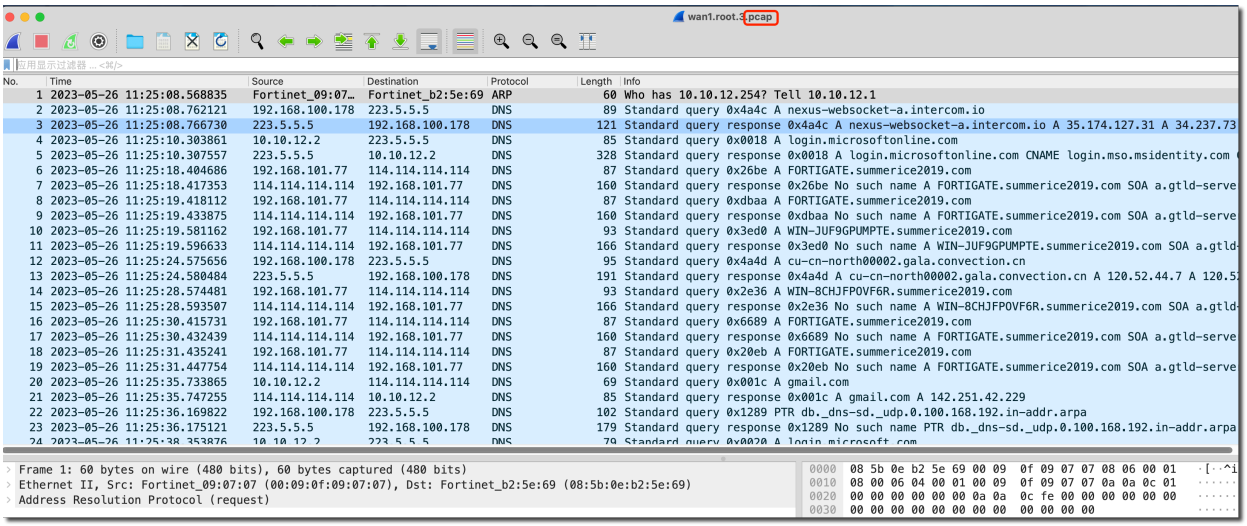
3. 右键点击创建的抓包进程，点击“开始”按钮，开启抓包，页面会显示抓包运行状态与已捕获的数据包个数。



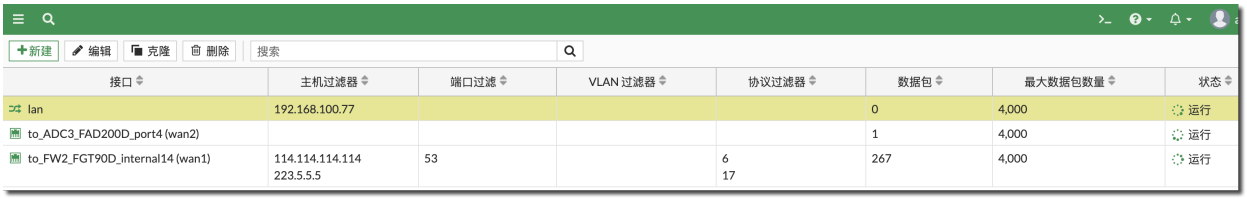
4. 抓包完成后，右键点击该抓包进程，可以选择停止抓包，也可以不停止抓包，直接下载当前已捕获的数据包。



5. 下载的数据包为 PCAP 格式，可以直接用 Wireshark 打开并分析。

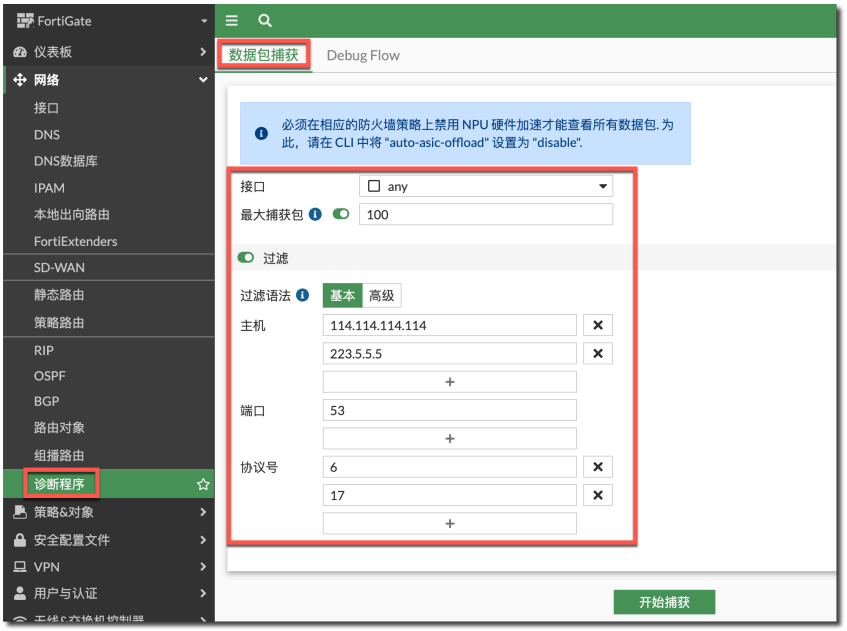


6. 同时可以创建多个抓包进程在后台抓取报文，离开此页面或登出设备不会中断抓包进程。



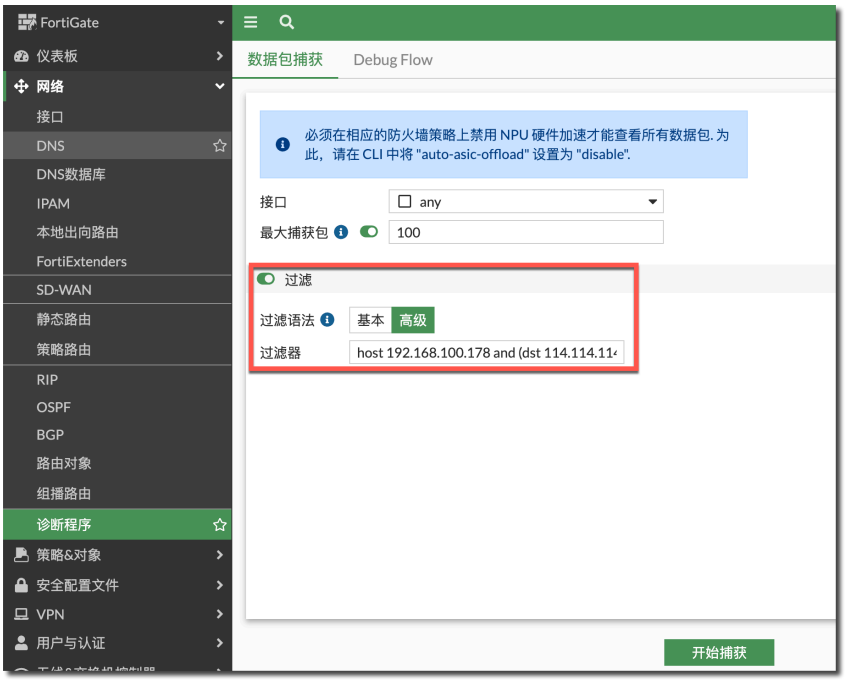
FortiOS 7.2

1. 进入“网络 → 诊断程序 → 数据包捕获”页面，配置捕获数据包的过滤条件，过滤语法选择基本时，可以过滤如下条件。

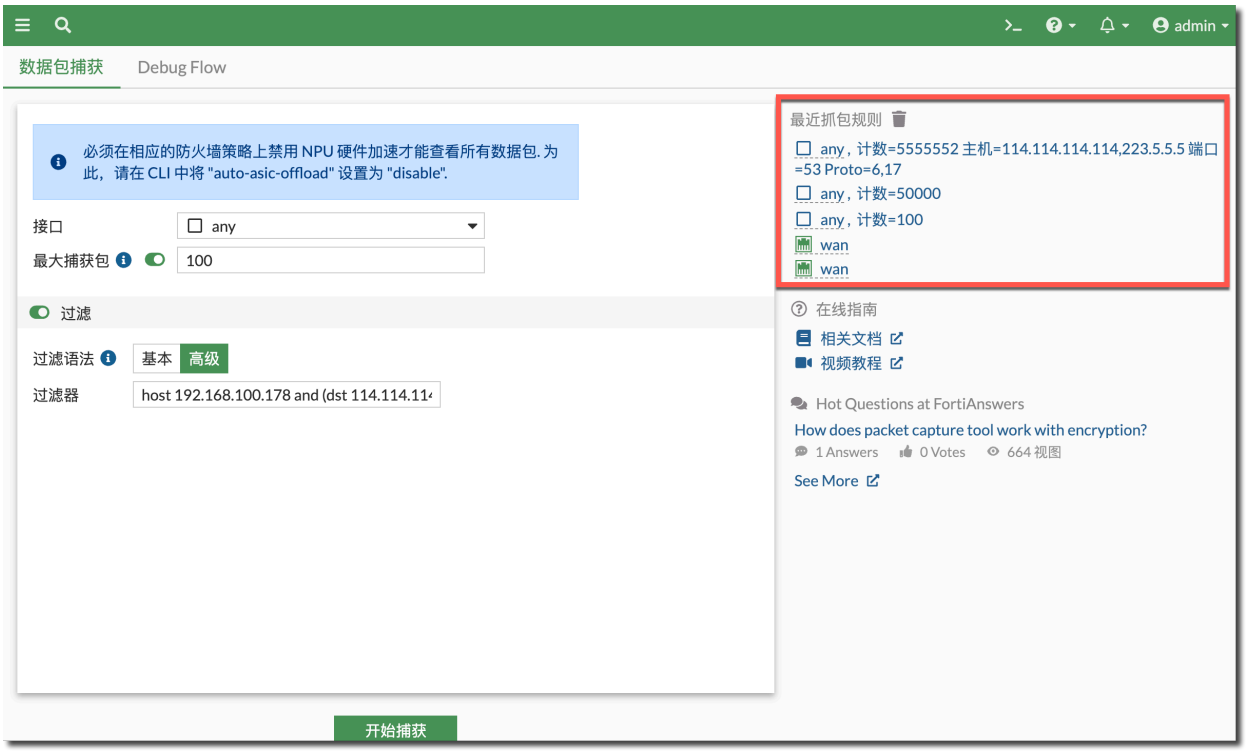


- 接口：指定抓取数据包的接口，可以指定 any，如果接口开启了 SNAT，抓取到的为 SNAT 后的数据包。
- 最大捕获数据包：每个数据包捕获可以抓取的最大数据包个数，达到该数字后，抓包自动停止，如果不指定最大捕获数量，默认在抓取 50000 个数据包后停止。
- Host：抓取数据包过滤的源或目的 IP 地址/网段，可以填写多个 IP 地址/网段（网段的格式为 192.168.1.0/24），它们之间是或的关系。
- 端口：过滤抓取数据包的端口号，可填写多个，它们之间是或的关系。
- 协议号：过滤抓取数据包的协议号，可填写多个，它们之间是或的关系。

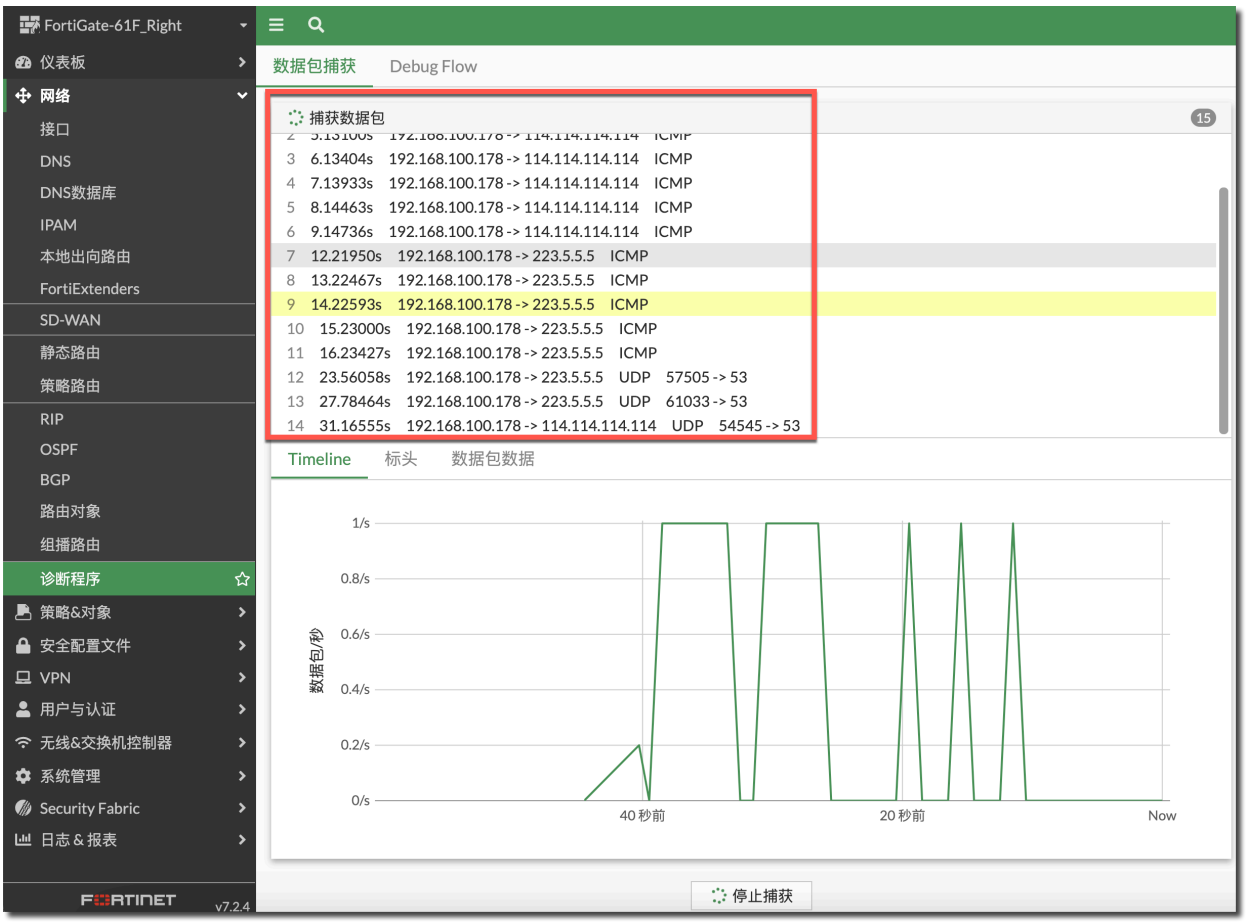
2. 过滤语法选择高级时，可以使用 CLI 下 Sniffer 命令中的过滤条件语法作为过滤条件（参考故障排查 → Sniffer 工具 → Sniffer 工具介绍 ↗），例如“*host 192.168.100.178 and (dst 114.114.114.114 or dst 223.5.5.5)*”。



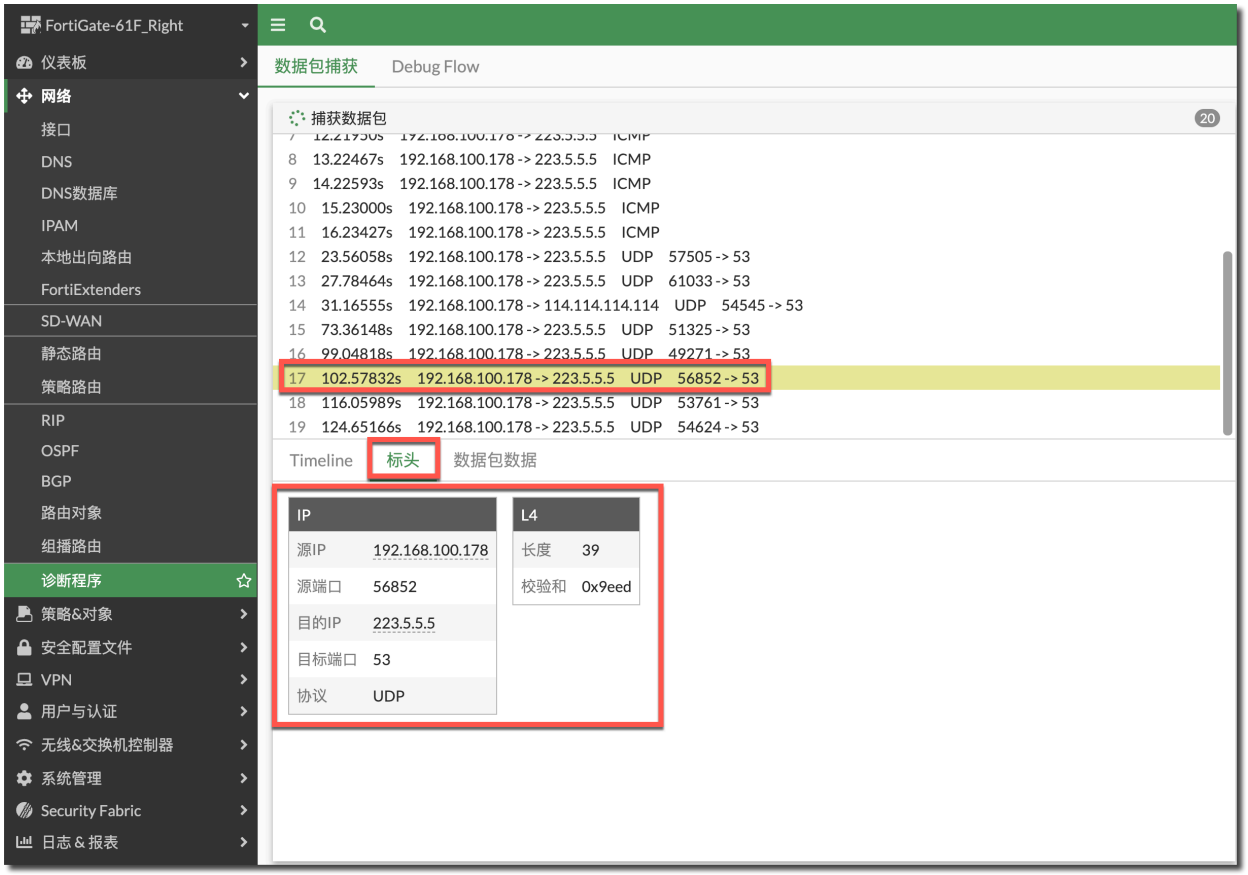
3. 页面右侧可以选择最近使用过的抓包规则。



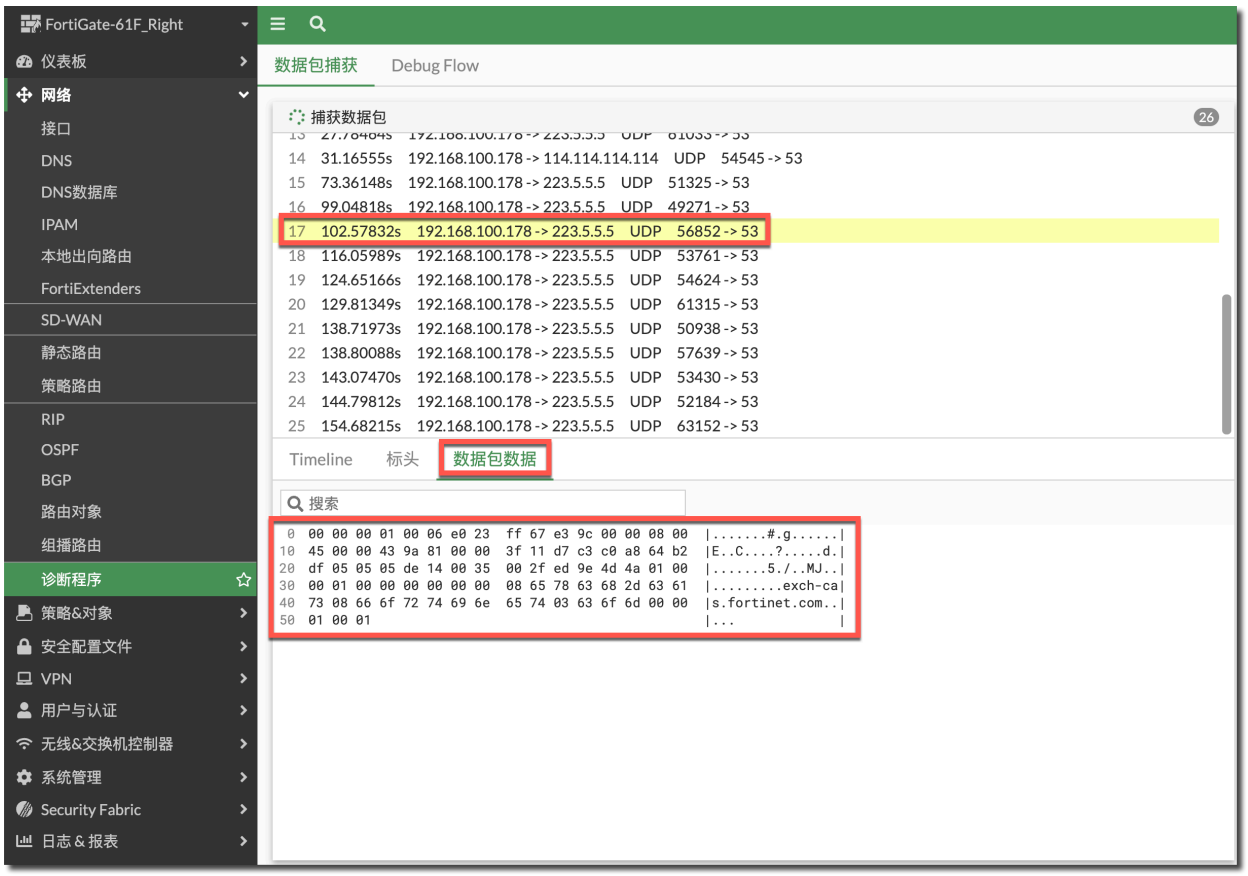
4. 点击“开始捕获”按钮开启抓包，可以看到数据包捕获的基本信息，以及对应的捕获速率。



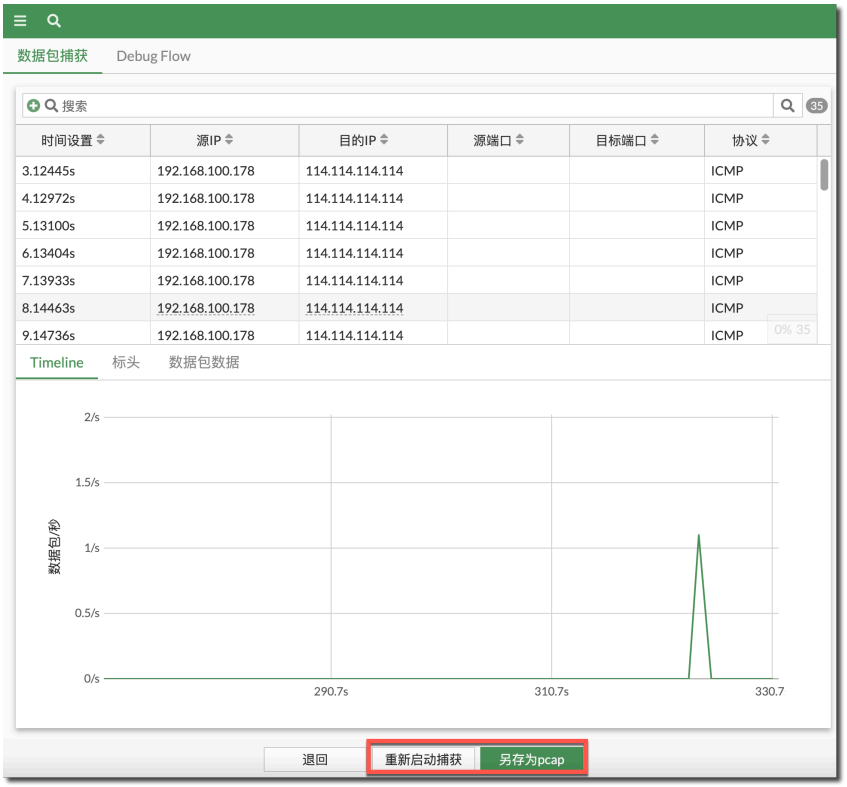
5. 选择某个数据包，点击“标头”，可以看到数据包的 Header 信息。



6. 选择某个数据包，点击“数据包数据”，可以看到数据包的具体 Data 内容。



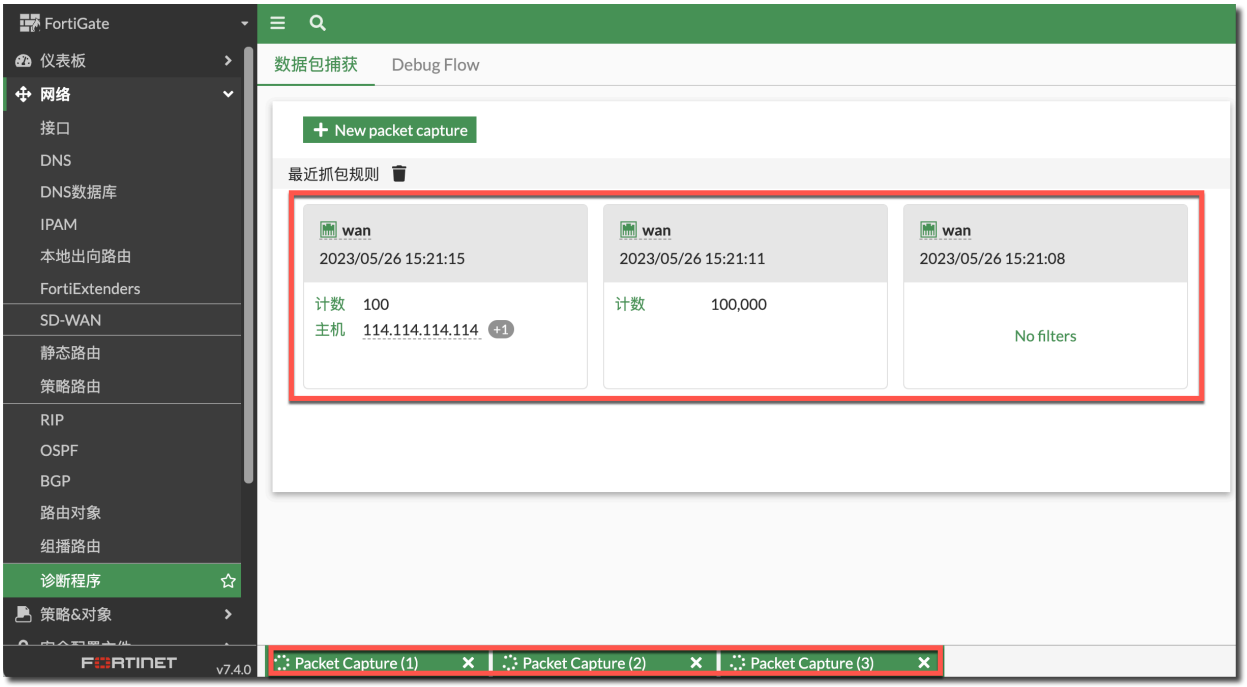
7. 抓包自动停止或手动点击“停止捕获”，可选择重新启动捕获，或将已捕获的数据包另存为 PCAP 格式以供使用 Wireshark 进一步分析，退出后，已捕获的数据包不会存储在 FortiGate 上。



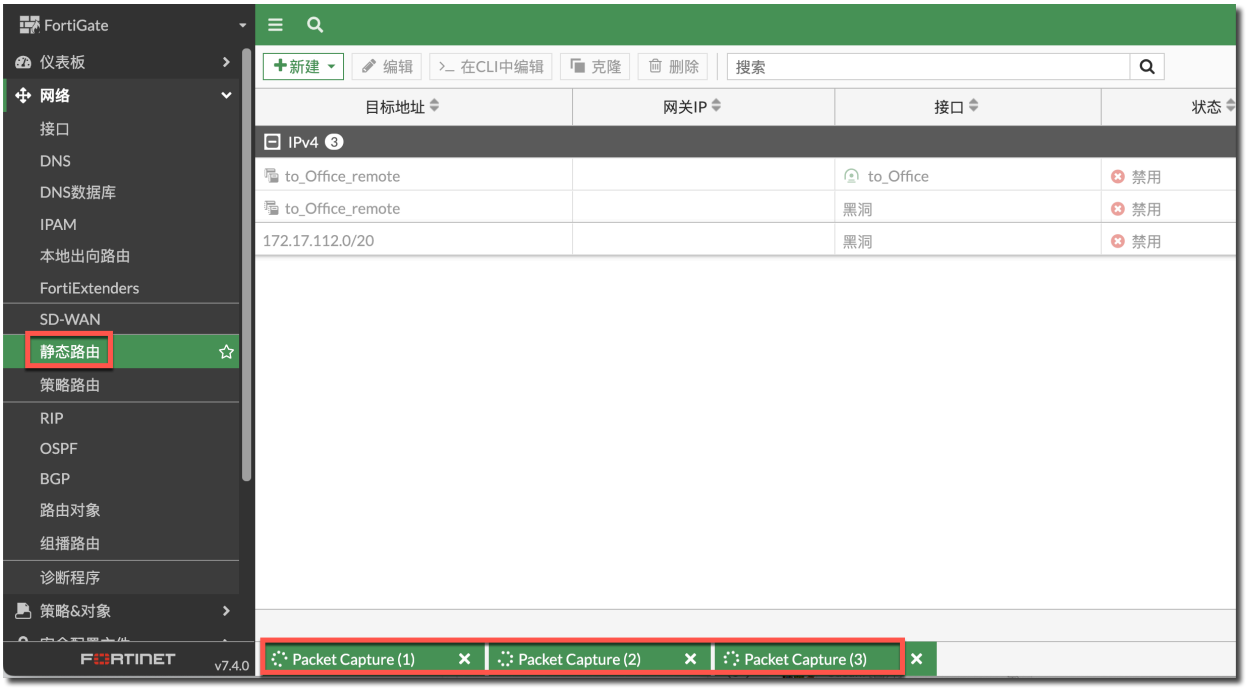
FortiOS 7.4

1. FortiOS 7.4 GUI 的抓包方法与 FortiOS 7.2 一致，相比较 FortiOS 7.2，FortiOS 7.4 可以同时进行多个抓包进程，且可以将多个抓包进程最小化到页面下方。

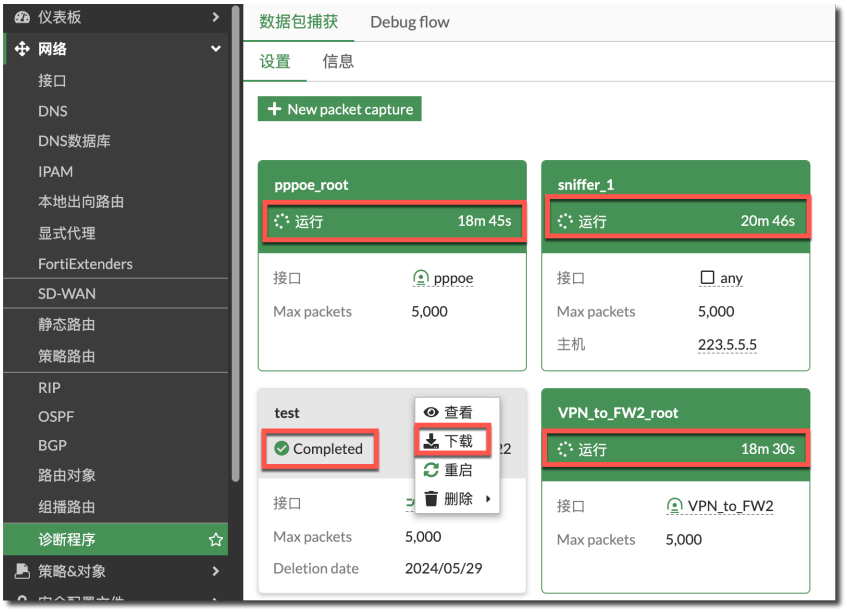




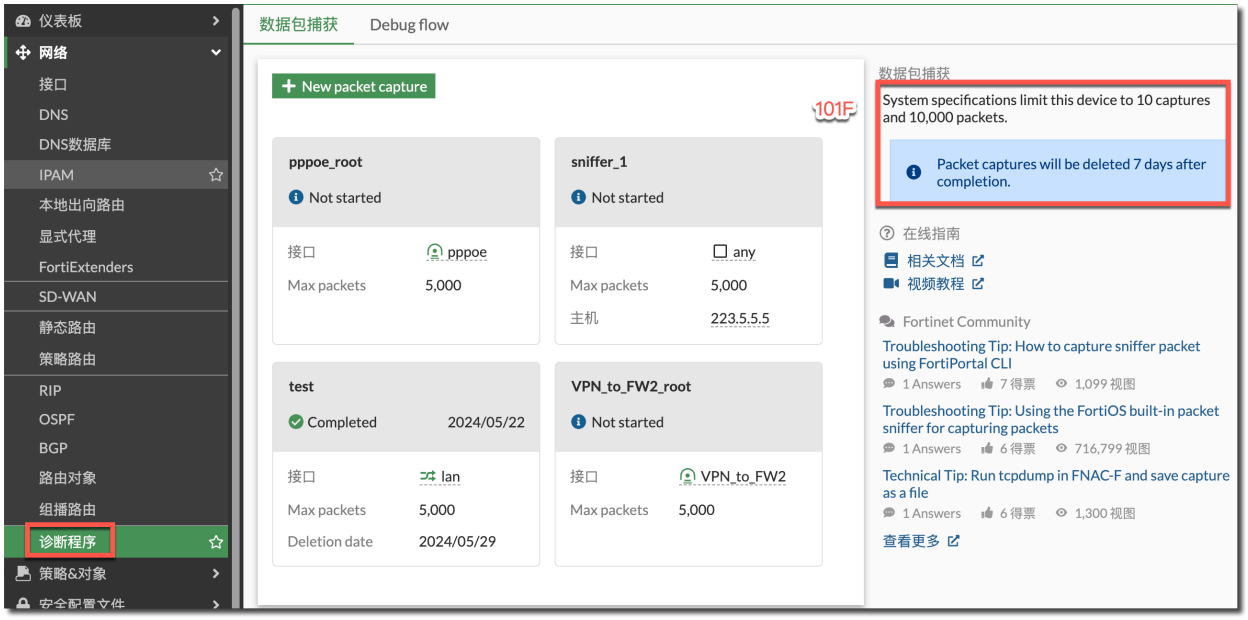
2. 离开“诊断程序”页面后，抓包进程仍然可以继续运行。

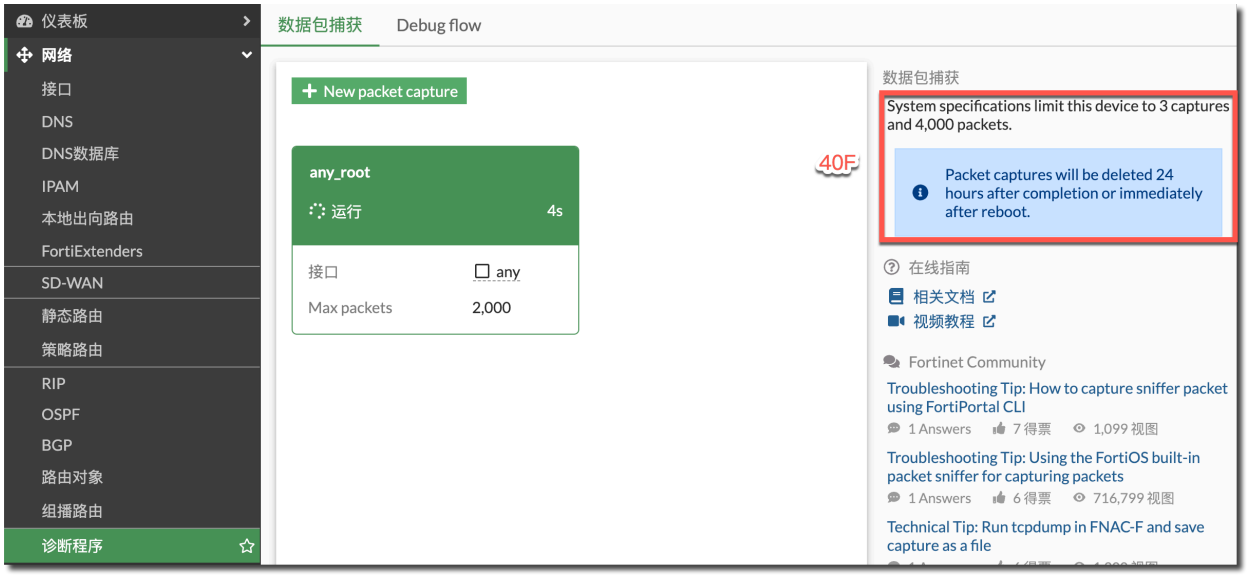


3. 退出登录或刷新页面后，页面下方的抓包窗口会消失，但回到“网络 → 诊断程序”页面后，抓包进程仍在进行中，已保存的抓包文件不会消失。



4. 重启设备后，带硬盘的设备（如 101F）可以保留已结束的抓包文件，不带硬盘的设备（如 60F）会删除已结束的抓包文件。不同型号 FortiGate 具体抓包规格和抓包文件最大保存天数可以参考“网络 → 诊断程序”页面右侧的提示信息。


仪表板
网络
接口
DNS
DNS数据库
IPAM
本地出向路由
显式代理
FortiExtenders
SD-WAN
静态路由
策略路由
RIP
OSPF
BGP
路由对象
组播路由
诊断程序
策略&对象
数据包捕获
Debug flow
New packet capture
101F
pppoe_root
Not started
接口 pppoe
Max packets 5,000
sniffer_1
Not started
接口 any
Max packets 5,000
主机 223.5.5.5
test
Completed 2024/05/22
接口 lan
Max packets 5,000
Deletion date 2024/05/29
VPN_to_FW2_root
Not started
接口 VPN_to_FW2
Max packets 5,000
数据包捕获
System specifications limit this device to 10 captures and 10,000 packets.
Packet captures will be deleted 7 days after completion.
在线指南
相关文档
视频教程
Fortinet Community
Troubleshooting Tip: How to capture sniffer packet using FortiPortal CLI
1 Answers 7 得票 1,099 视图
Troubleshooting Tip: Using the FortiOS built-in packet sniffer for capturing packets
1 Answers 6 得票 716,799 视图
Technical Tip: Run tcpdump in FNAC-F and save capture as a file
1 Answers 6 得票 1,300 视图
查看更多


仪表板
网络
接口
DNS
DNS数据库
IPAM
本地出向路由
FortiExtenders
SD-WAN
静态路由
策略路由
RIP
OSPF
BGP
路由对象
组播路由
诊断程序
数据包捕获
Debug flow
New packet capture
40F
any_root
运行 4s
接口 any
Max packets 2,000
数据包捕获
System specifications limit this device to 3 captures and 4,000 packets.
Packet captures will be deleted 24 hours after completion or immediately after reboot.
在线指南
相关文档
视频教程
Fortinet Community
Troubleshooting Tip: How to capture sniffer packet using FortiPortal CLI
1 Answers 7 得票 1,099 视图
Troubleshooting Tip: Using the FortiOS built-in packet sniffer for capturing packets
1 Answers 6 得票 716,799 视图
Technical Tip: Run tcpdump in FNAC-F and save capture as a file

NP7 芯片抓包

功能简介

当使用 NP7 平台芯片的设备（如 400F、600F、1000F、1801F 等）时，可以在数据流已经被 NP7 加速的情况下直接抓包，而不像 NP6 那样必须关闭流量匹配策略的 NP 加速后在 CPU 抓包。另外，也可以在 NP7 芯片上将过滤出的流量镜像到指定设备接口上。

```
diagnose npu sniffer {start | stop | filter}
```
bash

过滤条件

必须指定过滤条件后才可以开启抓包，使用 `diagnose npu sniffer filter` 可以定义 NP7 抓包的过滤条件，可以过滤的选项如下：

- `selector`：用于保存过滤条件，最多可以创建 4 个 selector（0 ~ 3，默认为 0）。每个 selector 可以保存一套过滤条件。使用 `diagnose npu sniffer filter selector <0 ~ 3>` 可以切换到对应的 selector 进行设置，用于当一个 selector 无法表示出全部的过滤条件时使用。多个 selector 下的过滤条件为或（or）的关系，同时生效。
- `intf <interface-name>` ：过滤抓包接口，可以抓取除 IPsec VPN 隧道接口以外的任何接口上的流量。
- `dir {0 | 1 | 2}`：过滤数据包在接口上的方向。0 表示入向数据包，1 表示出向数据包，2 表示双向数据包（默认为 2）。
- `ethtype <type>`： 如果抓取二层数据包，过滤数据包二层 ethertype 值。
- `protocol <number>`：过滤抓包的 IP 协议号，范围为 0 ~ 255。只能抓取到 NP7 处理器可以卸载的协议报文。
- `srcip <ipv4-ip-address>/<ipv4-mask>` ：过滤抓包的 IPv4 源地址/掩码。
- `dstip <ipv4-ip-address>/<ipv4-mask>`：过滤抓包的 IPv4 目的地址/掩码。
- `ip <ipv4-ip-address>/ 与<ipv4-mask>`：过滤抓包的 IPv4 源或目的地址/掩码。
- `srcip6 <ipv6-ip-address>/<ipv6-mask>`：过滤抓包的 IPv6 源地址/掩码。
- `dstip6 <ipv6-ip-address>/<ipv6-mask>`：过滤抓包的 IPv6 目的地址/掩码。
- `ip <ipv6-ip-address>/ 与<ipv6-mask>`：过滤抓包的 IPv6 源或目的地址/掩码。
- `sport <port-number>`：过滤抓包的 4 层源端口。
- `dport <port-number>`：过滤抓包的 4 层目的端口。
- `port <port-number>`：过滤抓包的 4 层源或目的端口。
- `outgoing_intf <interface>`：如果需要将抓取的报文镜像到设备的指定接口，在这里设置镜像的目标接口。
- `outgoing_vlan <vlan-id>`：如果镜像的目标接口需要封装 VLAN tag 传输，在这里设置镜像的目标接口的 VLAN ID。
- `clear`： 清空所有 selector 的过滤条件。

开启/关闭 NP7 抓包

配置好过滤条件后，使用 `diagnose npu sniffer {start | stop}` 在 NP7 上开始或停止抓包，然后使用 `diagnose sniffer packet npudbg` 显示抓取的数据包。

> **重要**
>
> 可根据需求抓取 4 还是 6 格式的报文，并指定抓包数量与时间显示方式，如下所示，这里的参数与普通 sniffer 是一样的。

```
diagnose npu sniffer {start | stop}
diagnose sniffer packet npudbg '' 4 0 l
```
bash

NP7 抓包示例

单一 selector

```
diagnose npu sniffer filter
diagnose npu sniffer filter intf port23
diagnose npu sniffer filter dir 1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer start
diagnose sniffer packet npudbg '' 4 0 l
```
bash

- 首先清除所有的抓包过滤器。
- 过滤接口为 port23。
- 过滤流量方向为出方向。
- 过滤协议为 TCP。

- 开启抓包。
- 在 CLI 下显示抓到的数据包。

多个 selector

```bash
diagnose npu sniffer filter selector 0
diagnose npu sniffer filter intf port1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer filter dir 2
diagnose npu sniffer filter selector 1
diagnose npu sniffer filter intf port2
diagnose npu sniffer filter protocol 17
diagnose npu sniffer filter dir 1
diagnose npu sniffer start
diagnose sniffer packet npudbg '' 4 0 l
```

- 第 1 个 selector 0：抓取 port1 上的双向 TCP 数据包。
- 第 2 个 selector 1：抓取 port2 上的出向 UDP 数据包。
- 在 CLI 下显示经过这两个 selector 过滤后抓到的数据包。

Debug Flow 工具

- Debug Flow 介绍
- Debug Flow 排错举例
- 通过 GUI 进行 Debug Flow

Debug Flow 介绍

Debug Flow 通常用于定位调试穿过或访问 FortiGate 数据流的处理过程，对于定位问题有很大的帮助。

命令解析

```bash
diagnose debug flow filter addr x.x.x.x          //过滤某个IP//
diagnose debug flow show function-name enable    //显示功能模块名称//
diagnose debug console timestamp enable          //显示时间戳//
diagnose debug flow trace start 10               //开启debug flow trace并显示10条debug信息//
diagnose debug enable                            //开启debug命令//
diagnose debug flow trace stop                   //关闭debug flow trace//
diagnose debug flow filter clear                 //清除过滤条件//
diagnose debug disable                           //关闭debug命令//
diagnose debug reset                             //重置所有的debug命令//
```

过滤条件

```bash
diagnose debug flow filter ?    //键入?查看Debug Flow支持的过滤条件，如IP，端口，协议等//
clear      Clear filter.
vd         Index of virtual domain.
vd-name    Name of virtual domain.
proto      Protocol number.
addr       IP address.
saddr      Source IP address.
daddr      Destination IP address.
port       port
sport      Source port.
dport      Destination port.
negate     Inverse filter.

diagnose debug flow filter    //直接键入回车查看当前的过滤条件//
        vf: any
        proto: any
        Host addr: any
        Host saddr: any
        Host daddr: any
        port: any
        sport: any
        dport: any

diagnose debug flow filter clear    //清除filter过滤条件//
```

Debug Flow 配置

1. 拓扑：PC1 (192.168.1.10)-----(port5：192.168.1.1) FGT (port1:192.168.89.35)-----Internet。

2. 策略配置。

```bash
config firewall policy
    edit 9
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

3. 设置 Debug Flow，并使用 PC1 ping 114.114.114.114。

```bash
diagnose debug flow filter addr 114.114.114.114
diagnose debug flow filter proto 1
diagnose debug flow show function-name enable
diagnose debug console timestamp enable
diagnose debug flow trace start 10
diagnose debug enable
```

Debug Flow 输出

1. 第 1 个数据包：Ping 的请求报文 echo reqeust。

```bash
2022-11-30 17:22:54 id=20085 trace_id=26 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2280→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2280, seq=1."
//received a packet表示FortiGate从port5收到一个192.168.1.10到114.114.114.114的echo request报文//

2022-11-30 17:22:54 id=20085 trace_id=26 func=init_ip_session_common line=6024 msg="allocate a new session-000e1480,
tun_id=0.0.0.0"
//创建会话，会话id是000e1480//

2022-11-30 17:22:54 id=20085 trace_id=26 func=vf_ip_route_input_common line=2605 msg="find a route: flag=04000000 gw-
192.168.89.254 via port1"
//查找路由，该报文需要从port1转发出去//

2022-11-30 17:22:54 id=20085 trace_id=26 func=get_new_addr line=1221 msg="find SNAT: IP-192.168.89.35(from IPPOOL),
port-62696"
//需要执行源NAT//

2022-11-30 17:22:54 id=20085 trace_id=26 func=fw_forward_handler line=881 msg="Allowed by Policy-9: SNAT"
//报文匹配id是9的策略//

2022-11-30 17:22:54 id=20085 trace_id=26 func=__ip_session_run_tuple line=3471 msg="SNAT 192.168.1.10-
>192.168.89.35:62696"
//将数据包的源地址192.168.1.10转换为192.168.89.35并发出//
```

2. 第 2 个数据包：ping 的响应报文 echo reply。

```bash
2022-11-30 17:22:54 id=20085 trace_id=27 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
114.114.114.114:62696→192.168.89.35:0) tun_id=0.0.0.0 from port1. type=0, code=0, id=62696, seq=1."
//received a packet表示FortiGate从port1收到一个114.114.114.114到的192.168.89.35 echo reply报文//

2022-11-30 17:22:54 id=20085 trace_id=27 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000e1480, reply direction"
//匹配已创建的会话id-000e1480//

2022-11-30 17:22:54 id=20085 trace_id=27 func=__ip_session_run_tuple line=3484 msg="DNAT 192.168.89.35:0-
>192.168.1.10:2280"
//反向报文需要进行目的NAT转换，将192.168.89.35转换为PC1的真实IP 192.168.1.10//

2022-11-30 17:22:54 id=20085 trace_id=27 func=vf_ip_route_input_common line=2605 msg="find a route: flag=00000000 gw-
192.168.1.10 via port5"
//查找路由，该报文需要从port5转发出去//

2022-11-30 17:22:54 id=20085 trace_id=27 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
port1 to port5, skb.npu_flag=00000000 ses.state=00010200 ses.npu_state=0×04000000"
2022-11-30 17:22:54 id=20085 trace_id=27 func=fw_forward_dirty_handler line=410 msg="state=00010200, state2=00000000,
npu_state=04000000"
```

3. 第 3 个数据包：ping 的请求报文 echo request 直接匹配会话转发。

```
2022-11-30 17:22:55 id=20085 trace_id=28 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2280→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2280, seq=2."
2022-11-30 17:22:55 id=20085 trace_id=28 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000e1480, original direction"
2022-11-30 17:22:55 id=20085 trace_id=28 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
port5 to port1, skb.npu_flag=00000400 ses.state=00010200 ses.npu_state=0×04000000"
2022-11-30 17:22:55 id=20085 trace_id=28 func=ip_session_install_npu_session line=346 msg="npu session installation
```

```
succeeded"
2022-11-30 17:22:55 id=20085 trace_id=28 func=fw_forward_dirty_handler line=410 msg="state=00010200, state2=00000000,
npu_state=04000400"
2022-11-30 17:22:55 id=20085 trace_id=28 func=__ip_session_run_tuple line=3471 msg="SNAT 192.168.1.10-
>192.168.89.35:62696"
```

4. 第 4 个数据包：ping 的影响报文 echo reply 直接匹配会话转发。

```bash
2022-11-30 17:22:55 id=20085 trace_id=29 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
114.114.114.114:62696→192.168.89.35:0) tun_id=0.0.0.0 from port1. type=0, code=0, id=62696, seq=2."
2022-11-30 17:22:55 id=20085 trace_id=29 func=resolve_ip_tuple_fast line=5931 msg="Find an existing session, id-
000e1480, reply direction"
2022-11-30 17:22:55 id=20085 trace_id=29 func=__ip_session_run_tuple line=3484 msg="DNAT 192.168.89.35:0-
>192.168.1.10:2280"
2022-11-30 17:22:55 id=20085 trace_id=29 func=npu_handle_session44 line=1183 msg="Trying to offloading session from
port1 to port5, skb.npu_flag=00000400 ses.state=00010200 ses.npu_state=0×04000400"
2022-11-30 17:22:55 id=20085 trace_id=29 func=ip_session_install_npu_session line=346 msg="npu session installation
succeeded"
2022-11-30 17:22:55 id=20085 trace_id=29 func=fw_forward_dirty_handler line=410 msg="state=00010200, state2=00000000,
npu_state=04000c00"
```

Debug Flow 排错举例

Denied by forward policy check (policy 0)

从 Debug Flow 输出发现 192.168.1.10 ping 114.114.114.114 匹配了策略 0（隐式丢包策略），即该数据包没有匹配的策略。

```bash
2022-11-30 18:16:43 id=20085 trace_id=30 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2328→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2328, seq=1."
2022-11-30 18:16:43 id=20085 trace_id=30 func=init_ip_session_common line=6024 msg="allocate a new session-000e4d3d,
tun_id=0.0.0.0"
2022-11-30 18:16:43 id=20085 trace_id=30 func=vf_ip_route_input_common line=2605 msg="find a route: flag=04000000 gw-
192.168.89.254 via port1"
2022-11-30 18:16:43 id=20085 trace_id=30 func=fw_forward_handler line=719 msg="Denied by forward policy check (policy 0)"
```

Denied by forward policy check (policy XX)

1. 从 Debug Flow 输出发现 192.168.1.10 ping 114.114.114.114 被 policy 9 丢弃。

```bash
2022-11-30 18:22:05 id=20085 trace_id=37 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.1.10:2335→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2335, seq=1."
2022-11-30 18:22:05 id=20085 trace_id=37 func=init_ip_session_common line=6024 msg="allocate a new session-000e5327,
tun_id=0.0.0.0"
2022-11-30 18:22:05 id=20085 trace_id=37 func=vf_ip_route_input_common line=2605 msg="find a route: flag=04000000 gw-
192.168.89.254 via port1"
2022-11-30 18:22:05 id=20085 trace_id=37 func=fw_forward_handler line=719 msg="Denied by forward policy check (policy
9)"
```

2. policy id 为 9 的策略如下。

```bash
config firewall policy
    edit 9
        set srcintf "port5"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic disable
        set action deny
    next
end
```

reverse path check fail, drop

1. 拓扑：PC1 (192.168.5.10)-----(192.168.5.1) Router (192.168.1.2)-----(port5:192.168.1.1) FortiGate (port1:192.168.89.35)----Internet

2. FortiGate 路由配置。

```bash
config router static
    edit 1
        set dst 0.0.0.0/0
        set gateway 192.168.89.254
        set device "port1"
    next
end
```

3. 从 Debug Flow 输出发现 192.168.5.10 ping 114.114.114.114 被反向路径检查丢弃，数据包是从 port5 接口收到的，FortiGate 会检查 port5 接口是否有到源地址 192.168.5.10 的路由，如果没有，则将丢弃该报文。

```bash
2022-11-30 19:02:32 id=20085 trace_id=38 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=1,
192.168.5.10:2471→114.114.114.114:2048) tun_id=0.0.0.0 from port5. type=8, code=0, id=2471, seq=1."
2022-11-30 19:02:32 id=20085 trace_id=38 func=init_ip_session_common line=6024 msg="allocate a new session-000e7d86,
tun_id=0.0.0.0"
2022-11-30 19:02:32 id=20085 trace_id=38 func=ip_route_input_slow line=2267 msg="reverse path check fail, drop"
```

```
2022-11-30 19:02:32 id=20085 trace_id=38 func=ip_session_handle_no_dst line=6110 msg="trace"
```

4. 解决方法：增加到 192.168.5.0/24 的路由。

```bash
config router static
    edit 2
        set dst 192.168.5.0/24
        set gateway 192.168.1.2
        set device "port5"
    next
end
```

iprope_in_check() check failed on policy 0, drop

原因 1

1. 访问一个防火墙上未开放的端口（包括自身管理端口/VIP 端口等），如下 port5 接口只允许 ping。

```bash
config system interface
    edit "port5"
        set ip 192.168.1.1 255.255.255.0
        set allowaccess ping
    next
end
```

2. 从 Debug Flow 发现 192.168.1.10 访问 FortiGate port5 接口 192.168.1.1 的 22 端口被拒绝，因为 port5 没有允许 SSH 访问。

```bash
2022-11-30 19:27:23 id=20085 trace_id=52 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=6,
192.168.1.10:53814→192.168.1.1:22) tun_id=0.0.0.0 from port5. flag [S], seq 2134956194, ack 0, win 29200"
2022-11-30 19:27:23 id=20085 trace_id=52 func=init_ip_session_common line=6024 msg="allocate a new session-000eb1ae,
tun_id=0.0.0.0"
2022-11-30 19:27:23 id=20085 trace_id=52 func=vf_ip_route_input_common line=2605 msg="find a route: flag=84000000 gw-
192.168.1.1 via root"
2022-11-30 19:27:23 id=20085 trace_id=52 func=fw_local_in_handler line=500 msg="iprope_in_check() check failed on policy
0, drop"
```

原因 2

1. 访问 FGT 本机流量，接口下已经开启了此服务，但是配置了管理员可信任主机，可信任主机不包括发起访问的源 IP，如下 port5 接口允许 ping，ssh，同时配置管理主机为 192.168.88.0/24 网段。

```bash
config system interface
    edit "port5"
        set ip 192.168.1.1 255.255.255.0
        set allowaccess ping ssh
    next
end
config system admin
    edit "admin"
        set trusthost1 192.168.88.0 255.255.255.0
    next
end
```

2. 从 Debug Flow 发现 192.168.1.10 访问 FortiGate port5 接口 192.168.1.1 的 22 端口被拒绝，因为 FortiGate 设置了信任主机，只允许 192.168.88.0/24 段的地址访问。

```bash
2022-11-30 19:35:13 id=20085 trace_id=58 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=6,
192.168.1.10:53818→192.168.1.1:22) tun_id=0.0.0.0 from port5. flag [S], seq 419563927, ack 0, win 29200"
2022-11-30 19:35:13 id=20085 trace_id=58 func=init_ip_session_common line=6024 msg="allocate a new session-000ec1a6,
tun_id=0.0.0.0"
2022-11-30 19:35:13 id=20085 trace_id=58 func=vf_ip_route_input_common line=2605 msg="find a route: flag=84000000 gw-
192.168.1.1 via root"
2022-11-30 19:35:13 id=20085 trace_id=52 func=fw_local_in_handler line=500 msg="iprope_in_check() check failed on policy
0, drop"
```

原因 3

1. 访问 FGT 本机流量，接口下已经开启了此服务，如下所示。

```bash
config system interface
    edit "port1"
        set ip 192.168.89.35 255.255.255.0
        set allowaccess ping ssh
    next
end
```

2. 但访问流量是从另外的一个接口发起的，而另外一个接口到本接口没有配置相应的防火墙策略。

3. 从 Debug Flow 发现 192.168.1.10 访问 FortiGate port1 接口 192.168.89.32 的 22 端口被拒绝，因为数据包是从 port5 接口来的，因此被拒绝。

```bash
2022-11-30 19:42:32 id=20085 trace_id=72 func=print_pkt_detail line=5845 msg="vd-root:0 received a packet(proto=6,
192.168.1.10:54384→192.168.89.35:22) tun_id=0.0.0.0 from port5. flag [S], seq 2800378420, ack 0, win 29200"
2022-11-30 19:42:32 id=20085 trace_id=72 func=init_ip_session_common line=6024 msg="allocate a new session-000ed260,
tun_id=0.0.0.0"
2022-11-30 19:42:32 id=20085 trace_id=72 func=vf_ip_route_input_common line=2605 msg="find a route: flag=84000000 gw-
192.168.89.35 via root"
2022-11-30 19:42:32 id=20085 trace_id=72 func=fw_local_in_handler line=500 msg="iprope_in_check() check failed on policy
0, drop"
```

4. 解决方法：增加 port5 到 port1 接口的策略。

```bash
config firewall policy
    edit 0
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "192.168.89.35/32"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

通过 GUI 进行 Debug Flow

功能介绍

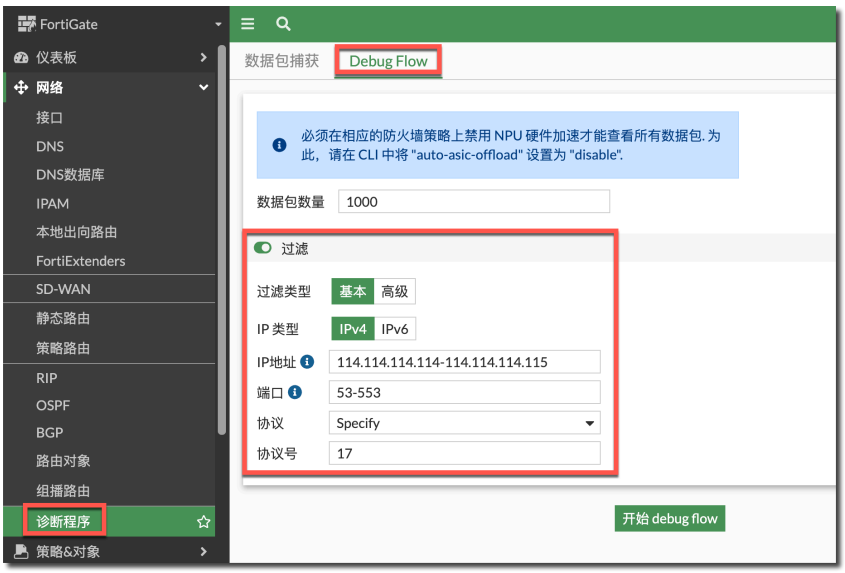
除了通过在 CLI 下进行 Debug Flow，FortiGate 也支持在 GUI 界面进行 Debug Flow，方便不熟悉 Debug Flow 工具的用户使用。

> **重要**
>
> 本功能在 FortiOS 7.2 和 7.4 版本支持。

使用方法

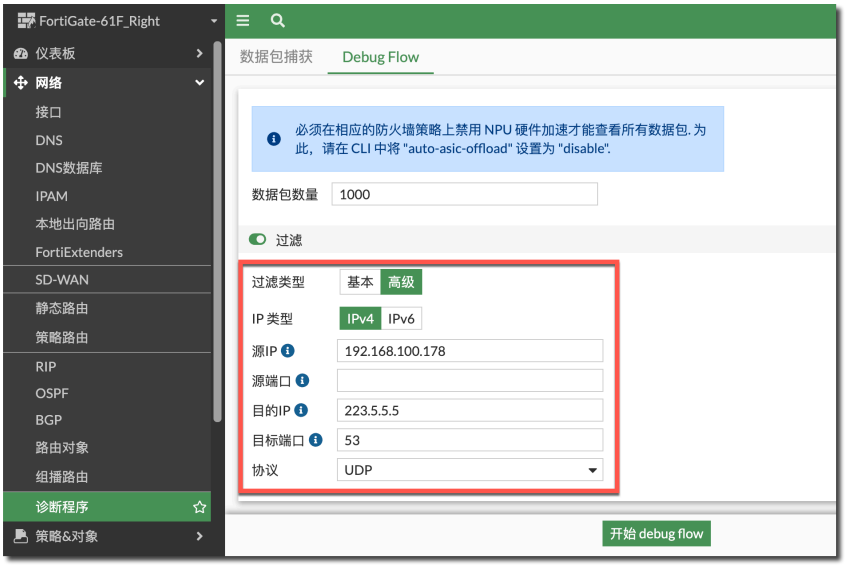
> **重要**
>
> 进行 Debug Flow 之前，必须在抓取流量对应的防火墙策略上禁用 NPU 硬件加速才能查看所有数据包的处理流程，请在 CLI 中将对应防火墙策略中的“auto-asic-offload”设置为“disable”。
>
> 详情参考：故障排查 → Sniffer、Debug Flow 与 NP 加速的矛盾 ↗。

1. 进入“网络 → 诊断程序 → Debug Flow”页面，配置 Debug Flow 的过滤条件，过滤语法选择基本时，可以过滤如下条件。

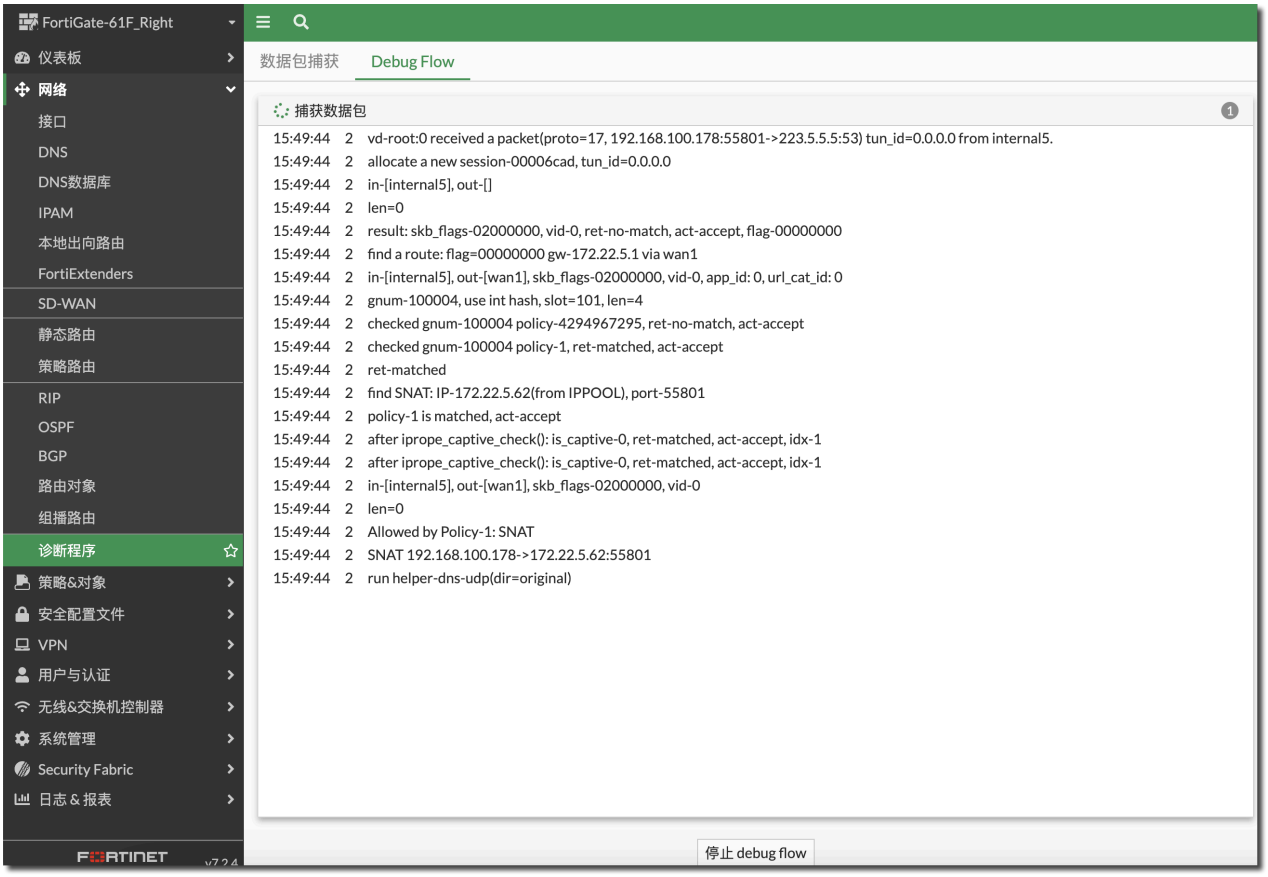


- 数据包数量：Debug Flow 可以捕获的最大数据包数量。
- IP 类型：过滤 IP 版本类型。
- IP 地址：Debug Flow 过滤的 IP 地址，可以是源 IP，也可以是目的 IP。格式可以写单个地址，也可以写一个 IP 范围，中间用“-”标识。
- 端口：过滤 Debug Flow 的端口号，格式可以写单个端口，也可以写一个端口范围，中间用“-”标识。。
- 协议：过滤 Debug Flow 的协议，可选择 Any、Specify（手工配置过滤的协议号）、TCP、UDP、SCTP、ICMP。

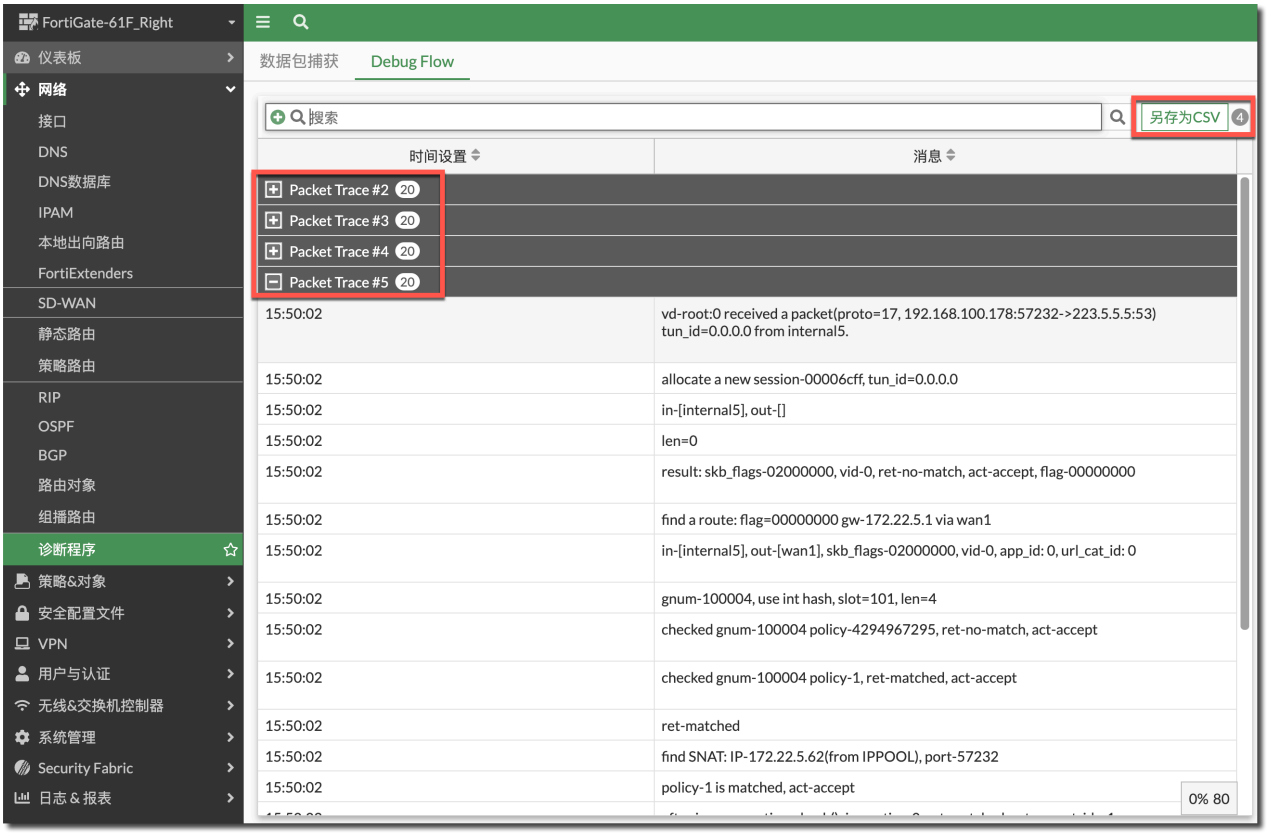
2. 过滤语法选择高级时，可以指定如下过滤条件。



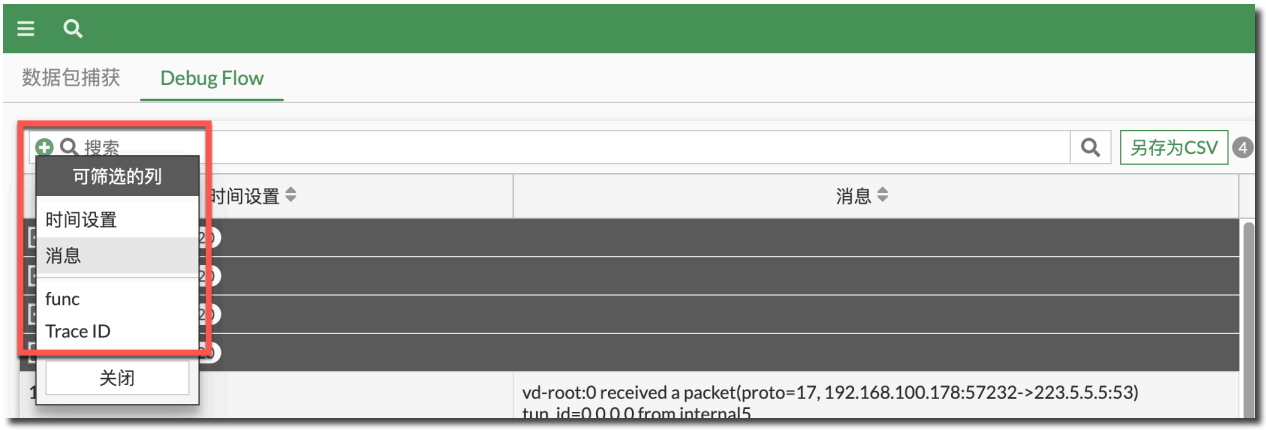
3. 点击“开始 Debug Flow”按钮开启数据流分析。

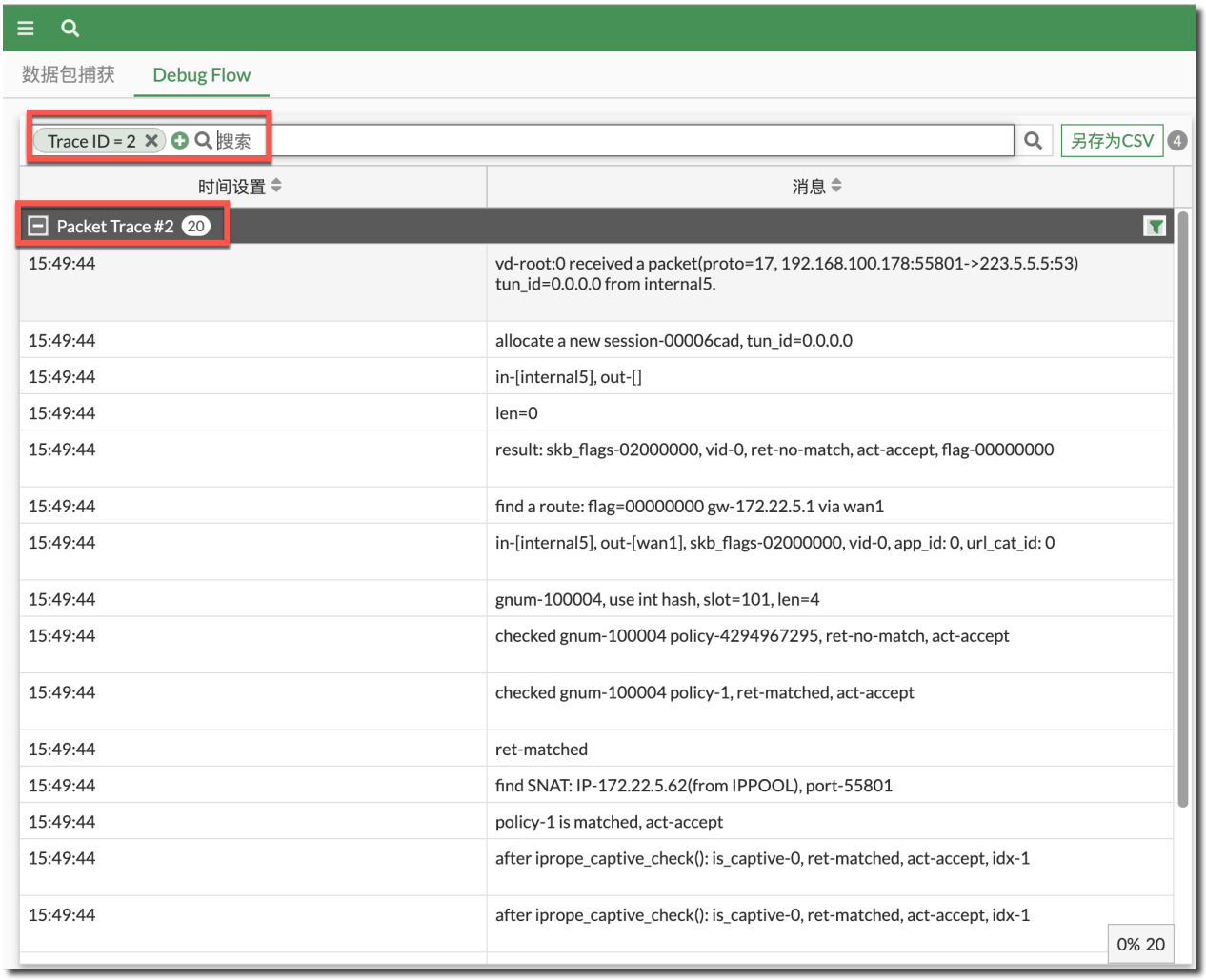


4. 点击“停止 Debug Flow”按钮，FortiGate 会将每个数据流合并打哦一个 Trace 中，可以在该结果中进行关键字搜索，也可点击“另存为 CSV”下载 Debug Flow 的文本信息。



5. 搜索时可以使用如下条件。


数据包捕获
Debug Flow
搜索
另存为CSV
可筛选的列
时间设置
消息
func
Trace ID
关闭
时间设置
消息
vd-root:0 received a packet(proto=17, 192.168.100.178:57232->223.5.5.5:53)
tun_id=0.0.0.0 from internal5.


数据包捕获
Debug Flow
Trace ID = 2
搜索
另存为CSV
时间设置
消息
Packet Trace #2 20
15:49:44
vd-root:0 received a packet(proto=17, 192.168.100.178:55801->223.5.5.5:53)
tun_id=0.0.0.0 from internal5.
15:49:44
allocate a new session-00006cad, tun_id=0.0.0.0
15:49:44
in-[internal5], out-[]
15:49:44
len=0
15:49:44
result: skb_flags-02000000, vid-0, ret-no-match, act-accept, flag-00000000
15:49:44
find a route: flag=00000000 gw-172.22.5.1 via wan1
15:49:44
in-[internal5], out-[wan1], skb_flags-02000000, vid-0, app_id: 0, url_cat_id: 0
15:49:44
gnum-100004, use int hash, slot=101, len=4
15:49:44
checked gnum-100004 policy-4294967295, ret-no-match, act-accept
15:49:44
checked gnum-100004 policy-1, ret-matched, act-accept
15:49:44
ret-matched
15:49:44
find SNAT: IP-172.22.5.62(from IPPOOL), port-55801
15:49:44
policy-1 is matched, act-accept
15:49:44
after iprope_captive_check(): is_captive-0, ret-matched, act-accept, idx-1
15:49:44
after iprope_captive_check(): is_captive-0, ret-matched, act-accept, idx-1
0% 20

数据包处理流程

简介

Parallel Path Processing

无 NP 处理的数据包流程

被 NP 加速的流量

安全检测模式

- 基于流的检测
- 基于代理的检测
- 显式 Web 代理

简介

FortiGate 防火墙从网络层到应用层对网络流量进行全面的安全检查，检查流程由 FortiGate 的安全策略控制，而具体的处理路径取决于设备的硬件结构以及启用的安全检测模式：

- **硬件配置**：是否配备 NP6/NP7 等网络处理器，以及 CP8/CP9/CP10 等内容处理器。
- **检测模式**：基于流检查或基于代理检查。检查模式基于安全策略，可针对每个安全策略单独配置检查模式，灵活性大幅提升。

本章节主要介绍数据包在运行 FortiOS 6.4 及更高版本的 FortiGate 中“入站 → 处理 → 出站”的完整流程。

FortiGate 的三类安全检查

FortiGate 执行的主要安全检测类型：

- **基于内核的状态检查（Stateful Inspection）**：是一种由防火墙内核实现的安全机制，它不仅检查单个数据包的内容，还会跟踪整个连接的状态信息，这是 FortiGate 基于会话状态工作的基本原理。
  - 当一个连接（如 TCP 或 UDP 通信）开始时，防火墙会在内核中建立一条会话状态表项，记录此会话的关键信息。
  - 当后续的报文经过防火墙时，内核会首先查找会话状态表。
    - 如果报文属于一个已建立的合法会话，则直接根据会话状态快速放行（或执行相应动作）。
    - 如果是一个新的会话请求，或半连接报文（如未匹配任何会话的 TCP ACK 报文），则需要按照安全策略、包头字段、协议行为等进一步检查，决定是否允许建立新连接。
- **流基于检查（Flow-based Inspection）**：通过对数据流内容取样（snapshot），并进行模式匹配（pattern matching），实时识别潜在的安全威胁。
- **基于代理检查（Proxy-based Inspection）**：通过重构经过 FortiGate 的数据内容，对完整会话内容进行深度安全分析。

安全功能与检查类型关系表

安全功能	内核（状态检查）	基于流检查	基于代理检查
防火墙（Firewall）	是		
IPsec VPN	是		
流量整形（Traffic Shaping）	是		
用户认证（User Authentication）	是		
管理流量（Management Traffic）	是		
SSL VPN	是		
IPS（入侵防御系统）		是	
僵尸网络检查（Botnet Checking）		是	
防病毒（AntiVirus）		是	是
应用控制（Application Control）		是	
Web 过滤（Web Filtering）		是	是
DLP（数据泄露防护）		是	是
邮件过滤（反垃圾邮件）		是	是
VoIP 检查（VoIP Inspection）			是
ICAP			是

检查方法特性对比

特性	状态检查（Stateful）	基于流检查（Flow）	基于代理检查（Proxy）
每会话检查单位	每个数据包	特定数据包，单遍并行	完整内容，顺序执行
内存、CPU 需求	低	中	高
威胁防护级别	良好（Good）	更好（Better）	最佳（Best）
认证支持	是	-	-
IPsec/SSL VPN 支持	是	-	-
防病毒防护	-	是	是
Web 过滤	-	是	是
DLP 支持	-	是	是
应用控制支持	-	是	-
IPS 支持	-	是	-
流量延迟	轻微（Minor）	无（No）	较小（Small）
重构完整内容	-	否（No）	是（Yes）

Parallel Path Processing

Parallel Path Processing（PPP，并行路径处理）通过防火墙策略配置从一组并行处理选项中，为数据包选择最优处理路径。大部分 FortiOS 功能通过防火墙策略触发，而策略中启用的功能直接决定数据包的处理逻辑。

PPP 的作用

- **差异化防护高风险流量**：对 HTTP、邮件等可能包含安全威胁的流量，可通过策略施加 UTM/NGFW 处理，并借助 CP8/CP9/CP10 处理器卸载加速。支持从基础 IPS 攻击防护（检测网络层攻击）到高级威胁管理（ATM）、应用控制、防病毒、DLP（数据泄露防护）的全维度防护。
- **优化无风险流量性能**：对低安全威胁的流量（如内部办公系统通信），可创建专用策略绕过 UTM/NGFW 检查。配备 NP 处理器的 FortiGate 可将此类流量卸载到 NP，释放 CPU 资源用于高风险流量处理，实现“安全不降级、性能不损耗”。

加速技术

- **NTurbo**：部分 FortiGate 型号支持 NTurbo，可将基于流 UTM/NGFW 的会话卸载到 NP 处理器。
- **IPSA**：基于流的会话通过 IPSA 技术，将模式匹配任务卸载到 CP 内容处理器，进一步降低 CPU 负载。

数据包的处理流程

穿越 FortiGate 的数据包会经历以下流程（并非所有数据包都需经过全部流程），具体取决于数据包类型和 FortiGate 软硬件配置。

入站流程（Ingress）

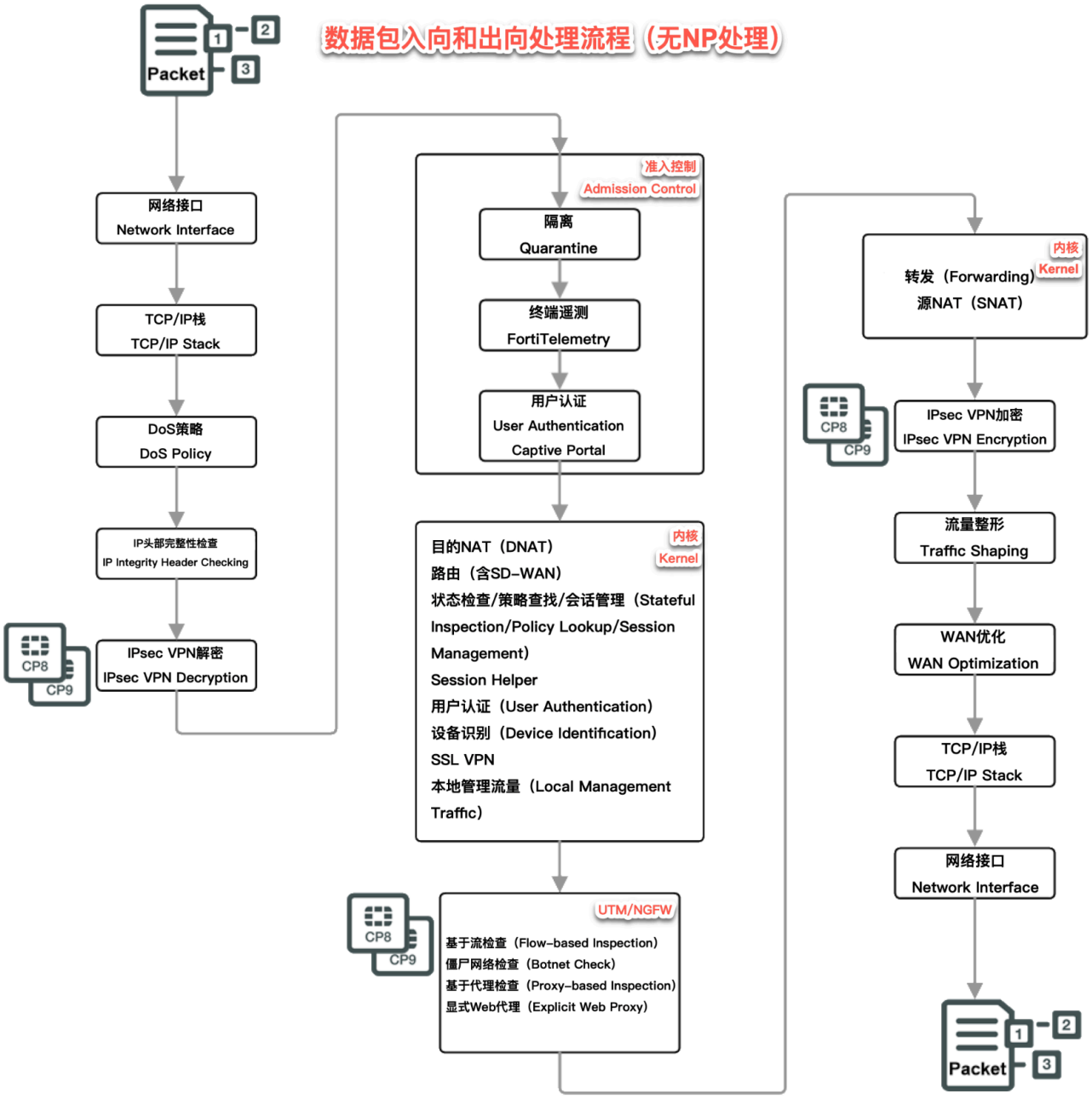
- 入站数据包流（Ingress Packet Flow）
  - 网络接口（Network Interface）
  - TCP/IP 栈（TCP/IP Stack）
  - DoS 策略（DoS Policy）
  - IP 完整性头部检查（IP Integrity Header Checking）
  - IPsec VPN 解密（IPsec VPN Decryption）
- 准入控制（Admission Control）
  - 隔离（Quarantine）
  - 终端遥测（FortiTelemetry）
  - 用户认证（User Authentication）
- 内核相关（Kernel）
  - 目的 NAT（Destination NAT，DNAT）
  - 路由（含 SD-WAN）
  - 状态检查/策略查找/会话管理（Stateful Inspection/Policy Lookup/Session Management）
  - Session Helper
  - 用户认证（User Authentication）
  - 设备识别（Device Identification）
  - SSL VPN
  - 本地管理流量（Local Management Traffic）
- UTM/NGFW 相关
  - 基于流检查（Flow-based Inspection）
    - NTurbo
    - IPSA
  - 僵尸网络检查（Botnet Check）
  - 基于代理检查（Proxy-based Inspection）
  - 显式 Web 代理（Explicit Web Proxy）
- 内核相关（Kernel）
  - 转发（Forwarding）
  - 源 NAT（Source NAT，SNAT）

出站流程（Egress）

- IPsec VPN 加密（IPsec VPN Encryption）
- 流量整形（Traffic Shaping）
- WAN 优化（WAN Optimization）
- TCP/IP 栈（TCP/IP Stack）
- 网络接口（Network Interface）

无 NP 处理的数据包流程

本节描述无 NP 处理的 FortiGate 中，数据包从入站到出站的完整处理步骤。



入站流程（Ingress）

所有被 FortiGate 接收的数据包均需通过网络接口，并由 TCP/IP 协议栈进行处理。若已配置 DoS 策略，数据包还必须通过这些策略以及自动 IP 头部完整性检查。

DoS 策略检查

在报文进入 FortIGate 的早期阶段，DoS 模块会检测流量是否异常，以判断是否属于拒绝服务攻击（DoS attack）。

- 检查范围：仅跟踪可能用于 DoS 攻击的数据包（如 TCP SYN 包），验证其是否符合预设参数。
- 处理逻辑：疑似攻击数据包直接阻断，合法数据包进入下一环节。

IP 完整性头部检查

此步骤主要验证 TCP、UDP、ICMP、SCTP、GRE 等协议头部长度是否合法。

- 支持协议：TCP、UDP、ICMP、SCTP、GRE。
- 处理逻辑：头部长度正确则进入下一步，错误则直接丢弃。

IPsec VPN 解密

若入站数据包匹配已配置的 IPsec 隧道：

- IPsec 引擎尝试用正确密钥解密。
- 解密成功：解密后数据包进入下一环节。
- 解密失败/非 IPsec 流量：直接进入下一环节（不处理）。

> **重要**
>
> IPsec 解密可由 CP 处理器卸载加速。

准入控制

执行三层准入校验，确保流量来源合法：

1. 隔离检查：阻断源/目的地址在“隔离（quarantine）列表”中的数据包。
2. FortiTelemetry（可选）：要求设备安装 FortiClient 才能通过。
3. 强制门户认证（可选）：对入站流量触发 Captive Portal 认证。

内核处理

数据包通过入站检查后，由 FortiOS 内核执行核心转发与控制逻辑。

DNAT

内核查询 NAT 表，判断是否需要修改入站流量的目的 IP 地址。

- 典型场景：将互联网流量导向内网服务器（如 Web 服务器），隐藏内网真实地址。
- 关键顺序：DNAT 必须在“路由”前执行，确保 FortiGate 能正确路由修改后的地址。
- 路由验证：同步检查目的地址是否存在有效路由。

路由（含 SD-WAN）

通过路由表确定数据包的出接口，并区分“本地流量”（目的地为 FortiGate 自身）和“转发流量”（目的地为其他网络）：

- 策略匹配依据：源接口（入站时已知）+ 目的接口（路由确定）。
- SD-WAN 特殊逻辑：
  1. SD-WAN 首先利用应用识别模块（Application Control）对新会话的首包进行 L4 的 ISDB 匹配。
  2. 匹配成功：按 SD-WAN 规则路由整个会话。
  3. 匹配失败：应用“隐式 SD-WAN 规则”，并通过七层应用控制继续识别。
  4. 识别成功：在 ISDB 缓存中添加四层匹配记录，优化后续匹配效率。

状态检查/策略查找/会话管理

针对会话的第一个数据包执行策略匹配与会话管理，是防火墙的核心逻辑。

- 会话边界识别：通过 TCP SYN/FIN 标志判断会话开始/结束。
- 合法性校验：检查源/目的 IP、端口、协议，以及载荷/序列号的完整性。
- 会话表优化：第一个数据包匹配策略后，将会话信息存入“会话表”，后续数据包直接查询会话表（比每次都查询策略高效）。
- 会话生命周期：处理完最后一个数据包后删除会话，或触发“空闲超时”后删除。

> **重要**
>
> 更多状态检查细节可查阅维基百科“状态防火墙”词条：https://en.wikipedia.org/wiki/Stateful_firewall ↗

Session Helper

部分协议（如 SIP、FTP）需分析数据包载荷才能正常转发，FortiOS 通过 Session Helper 解析这些协议的载荷信息，并动态调整防火墙规则。

- 支持的 Session Helper：PPTP、H323、RAS、TNS、TFTP、RTSP、FTP、MMS、PMAP、SIP、DNS - UDP、RSH、DCERPC、MGCP 等。
- 典型场景：SIP 协议通过 TCP 控制包建立通话，会话助手提取载荷中的语音端口信息，允许语音数据包通过。

用户认证

当防火墙策略匹配成功且该策略包含用户认证要求时，系统会启动认证流程（基于 IP 地址）。

设备识别

对流量进行设备特征分析，以识别来源设备类型（例如 PC、手机、IoT 终端等）。

SSL VPN

SSL VPN 流量通过目的端口识别，解密后路由到 SSL VPN 接口，再通过策略转发（SSL 加解密由 CP 加速）。

本地管理流量

- 终止接口：多 VDOM 模式终止于管理接口，透明模式终止于管理 IP。
- 流量范围：管理员访问（GUI/CLI）、FortiManager 集中管理、FortiGuard 通信等。
- 访问控制：由 local-in 策略定义（参考策略与对象 → 本地流量策略 → Local-in 策略 ↗章节），不参与后续状态检查。

UTM/NGFW 处理

若匹配的安全策略包含“安全配置文件”，数据包需进入 UTM/NGFW 处理，处理方式取决于策略的检测模式。

基于流检查（单遍架构）

通过单遍 DFA（直接过滤法）模式匹配实时阻断威胁，无代理参与。

- 处理流程：先通过解码器识别协议，再执行单遍模式匹配（IPS、应用控制等并行），最后检查僵尸网络地址。
- 支持功能：IPS、应用控制、Web 过滤、DLP、僵尸网络检查、防病毒。
- 加速方式：模式匹配由 CP 卸载。

基于代理检查（混合模式）

结合基于流与基于代理检查，深度处理高风险流量。

- 第一步：IPS 引擎执行单遍基于流检查（IPS、应用控制、僵尸网络检查）。
- 第二步：代理重构内容，按顺序执行检查（VoIP 检查 → DLP → 反垃圾邮件 → Web 过滤 → 防病毒 → ICAP）。
- 支持功能：除基于流功能外，额外支持 VoIP 深度检查、ICAP 等。

显式 Web 代理检查

逻辑与基于代理检查类似，仅针对 HTTP/HTTPS 流量，通过指定端口（默认 8080）监听。

CP（内容处理器）

大部分 FortiGate 配备 SPU（安全处理单元）内容处理器（CP），加速资源密集型安全任务（由主 CPU 分配任务）。不同型号 CP 能力不同，新型号以 CP9/CP10 为主。CP 的核心能力如下：

- 基于流检查加速：优化 IPS/应用控制模式吞吐量。
- VPN 优化：高性能 IPsec/SSL/TLS 处理，支持 DES/3DES/AES 等加密算法。
- 哈希与 HMAC：支持 MD5/SHA 系列哈希，符合 FIPS 标准。
- 密钥加速：IKE/RSA 密钥交换加速，含真随机数生成器。
- 其他：DLP 指纹支持、TTTD 内容分块。

出站流程（Egress）

数据包完成内核与 UTM/NGFW 处理后，通过以下步骤离开 FortiGate。

源 NAT（SNAT）

查询 NAT 表，修改出站流量的源 IP（如内网访问互联网时隐藏真实地址）。

IPsec VPN 加密

若需进入 IPsec 隧道，执行加密封装（由 CP 加速）。

流量整形

按配置调整流量速率，保障关键业务带宽。

WAN 优化

执行压缩、缓存等优化，提升 WAN 链路效率。

TCP/IP 栈处理

完成协议封装。

出接口

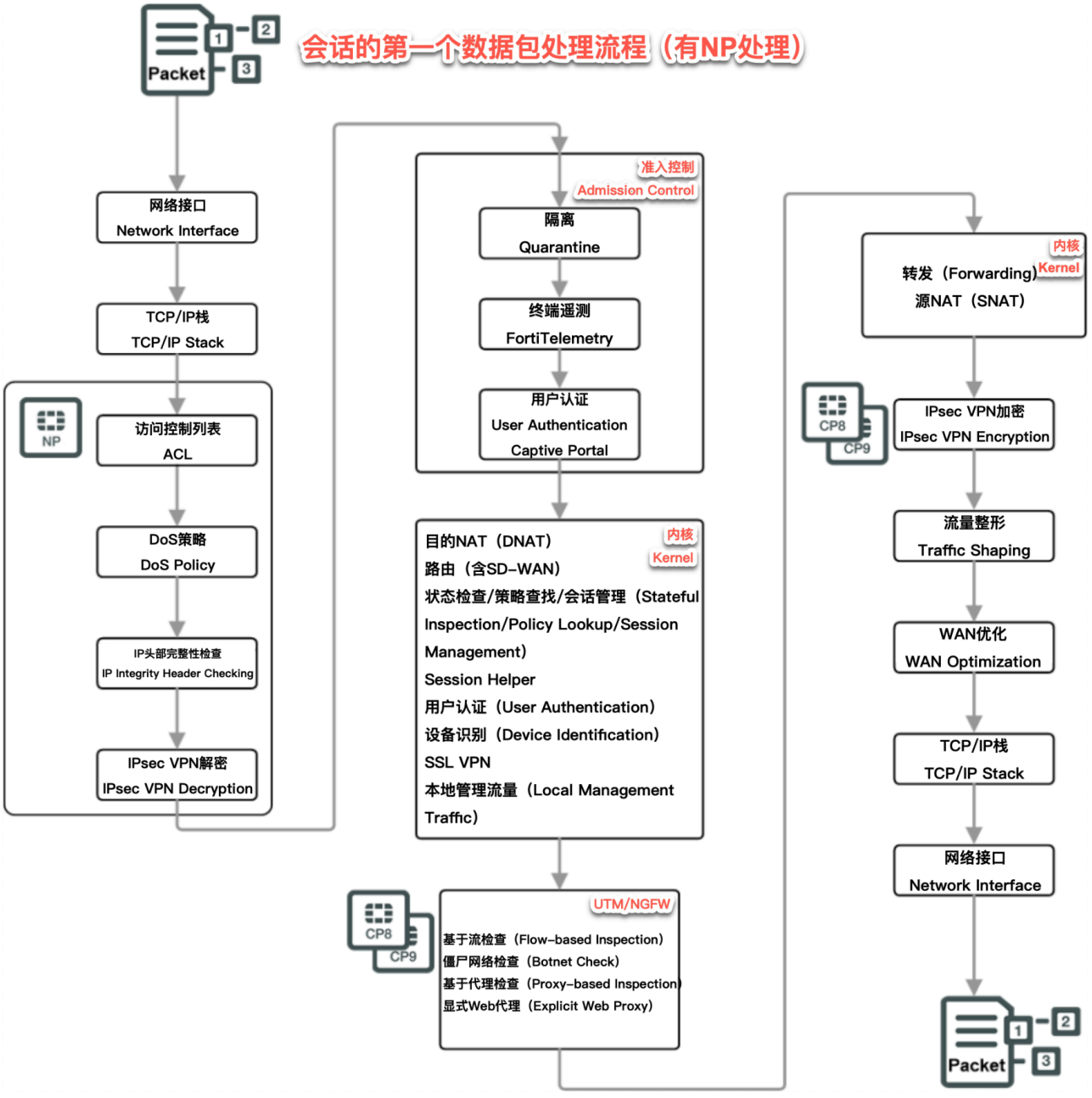
数据包通过出接口离开。

被 NP 加速的流量

NP 会话

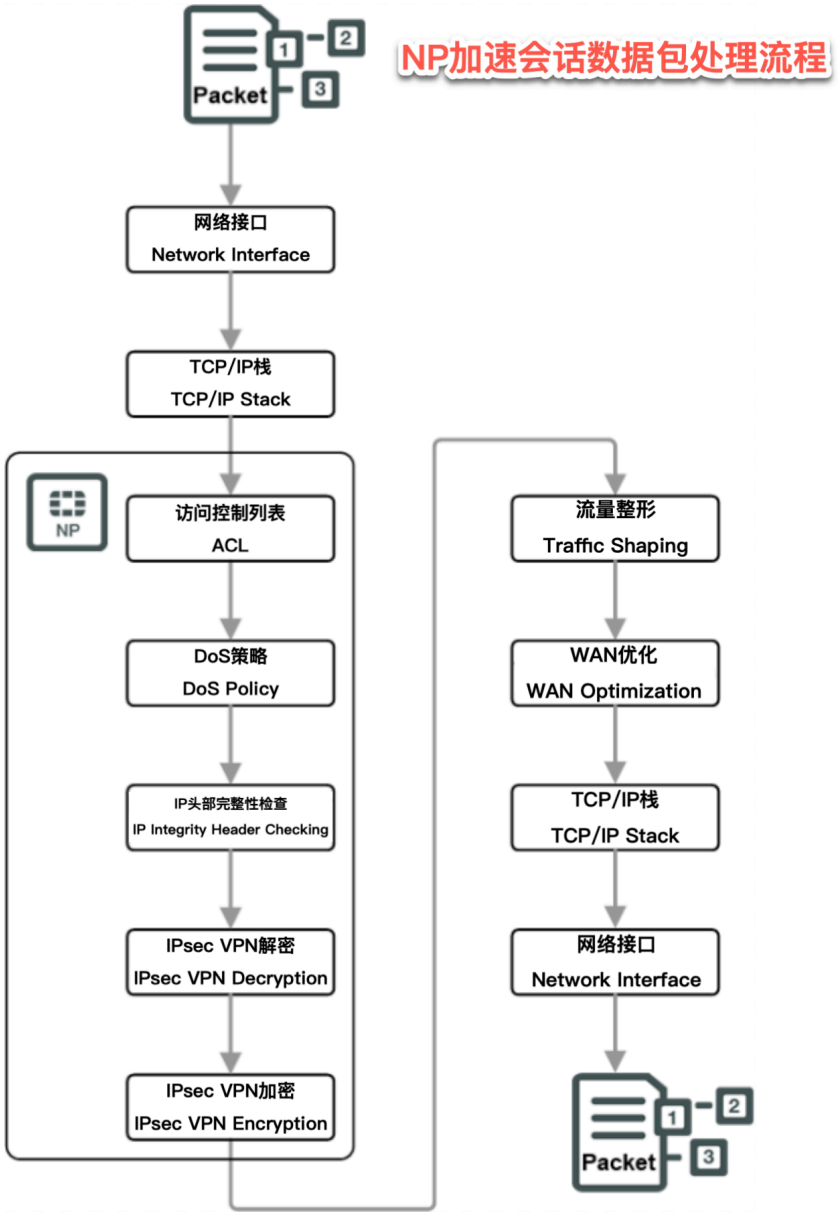
第一个数据包流程

配备 NP 处理器的 FortiGate，新会话的第一个数据包会判断是否可卸载（无基于代理 UTM/NGFW 时大多可卸载），其处理流程与“无 NP 加速设备”基本一致。后续数据包则进入简化的卸载流程。



已加速会话数据包流程

新会话第一个数据包处理完成后，后续数据包跳过路由、UTM/NGFW、内核，直接由 NP 转发，安全检查也由 NP 加速。



- 入站：网络接口 → TCP/IP 栈 → NP6 执行 DoS 检查/IP 完整性检查/ACL 检查。
- 处理：（若需）IPsec VPN 加密 → 流量整形 → WAN 优化。
- 出站：TCP/IP 栈 → 出口接口。

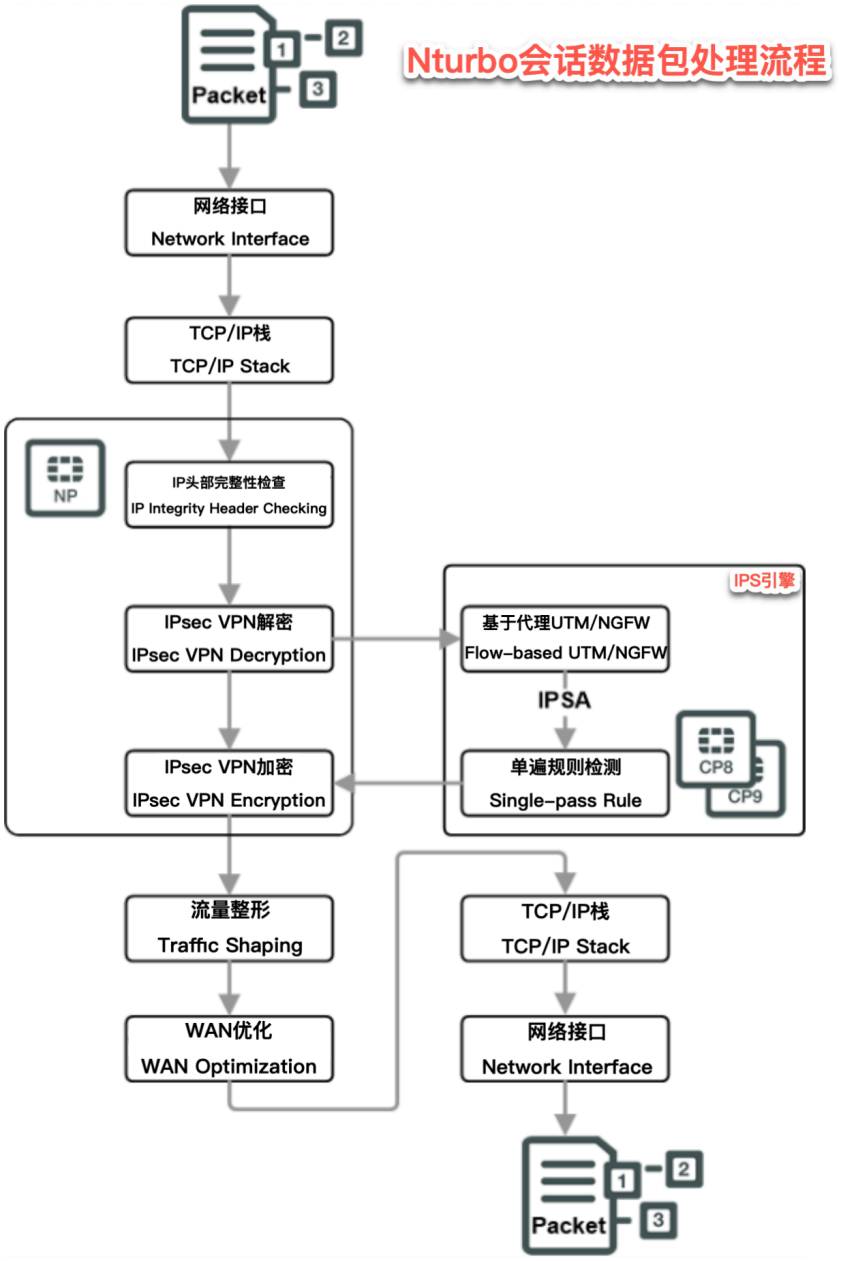
访问控制列表（ACL）

ACL 是 NP 的功能之一，在物理接口层（CPU 分析前）丢弃不合规 IPv4/IPv6 数据包，仅适用于 NP 接口。

- 优势：检查早、由 NP 执行，不占用 CPU 资源。
- 作用：在数据包到达 CPU 前阻断威胁，提升设备安全性。

NTurbo 的数据包流程

支持 NTurbo 的 FortiGate，可将基于流检测的 UTM/NGFW 会话卸载到 NP，进一步提升处理效率。在已卸载的基于流的 UTM/NGFW 会话中，首个数据包处理完成后，后续数据包将跳过路由和内核处理器。基于流的 UTM/NGFW 操作仍由 CPU 处理，同时将 IPSA 卸载模式匹配任务转发至 CP 处理器。



- 入站：网络接口 → TCP/IP 栈 → NP6 执行 IP 完整性检查。
- 处理：IPS 引擎通过 IPSA 卸载模式匹配 →（若需）IPsec VPN 加密 → 流量整形 → WAN 优化。
- 出站：TCP/IP 栈 → 出口接口。
- 威胁处理：检测到威胁则阻断会话，否则由 NP 转发。
- NTurbo 兼容性限制：启用以下功能时，基于流 UTM/NGFW 会话无法通过 NTurbo 卸载。
- DoS 策略。
- Session Helper。
- 大部分隧道类型（如 GRE）。

安全检测模式

- 基于流的检测
- 基于代理的检测
- 显式 Web 代理

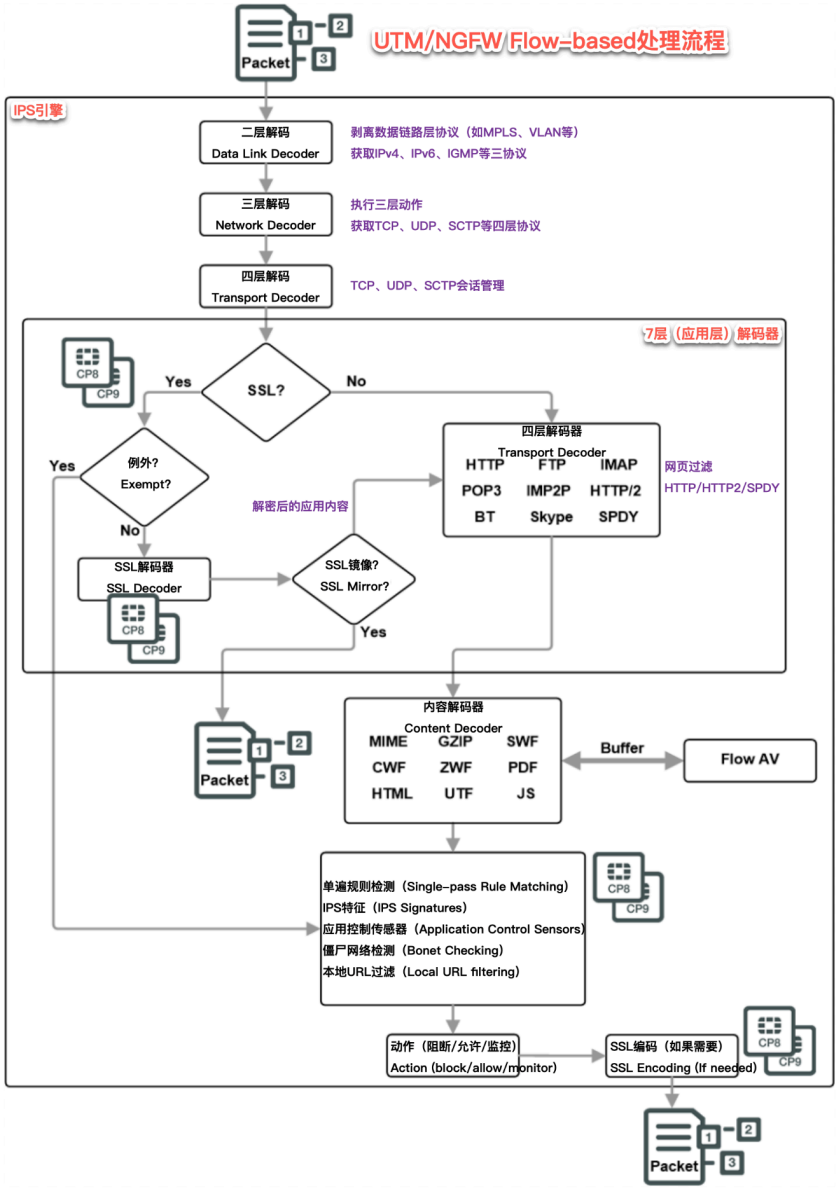
基于流的检测

基于流的 UTM/NGFW 检测采用单次扫描架构，通过直接过滤方法（DFA）模式匹配实时识别潜在攻击或威胁，并即时拦截安全威胁。

若防火墙策略配置为基于流检测，则根据接受会话的防火墙策略中选定的选项，基于流检测可应用入侵防御系统（IPS）、应用控制、网页过滤、数据防泄漏（DLP）、僵尸网络检测及防病毒功能。基于流检测完全由 IPS 引擎完成，不涉及代理操作。

```bash
config firewall policy
    edit 1
        set inspection-mode flow    //默认配置//
    next
end
```

处理流程



协议解码与 SSL 处理

在应用基于流的检测之前，IPS 引擎会通过一系列解码器，根据数据包协议及策略设置确定应启用的安全模块。此外，若配置了 SSL 深度检测功能，IPS 引擎还会对 SSL 数据包进行解密。SSL 解密操作由 CP 处理器卸载并加速处理。

如果您配置了 SSL 镜像功能，IPS 引擎会复制解密后的应用程序内容，将其封装在 TCP 数据包中（包含 IP 和以太网头），并发送至配置的镜像接口，二层目标地址为广播地址（FF:FF:FF:FF:FF:FF），源地址为全零地址。

单遍安全检查

所有适用安全模块并行执行，模式匹配由 CP 加速。

- 并行检查：IPS、应用控制、本地 URL 过滤、僵尸网络检查、DLP。
- 防病毒：协议解码时缓存文件，同步执行病毒扫描。

总结

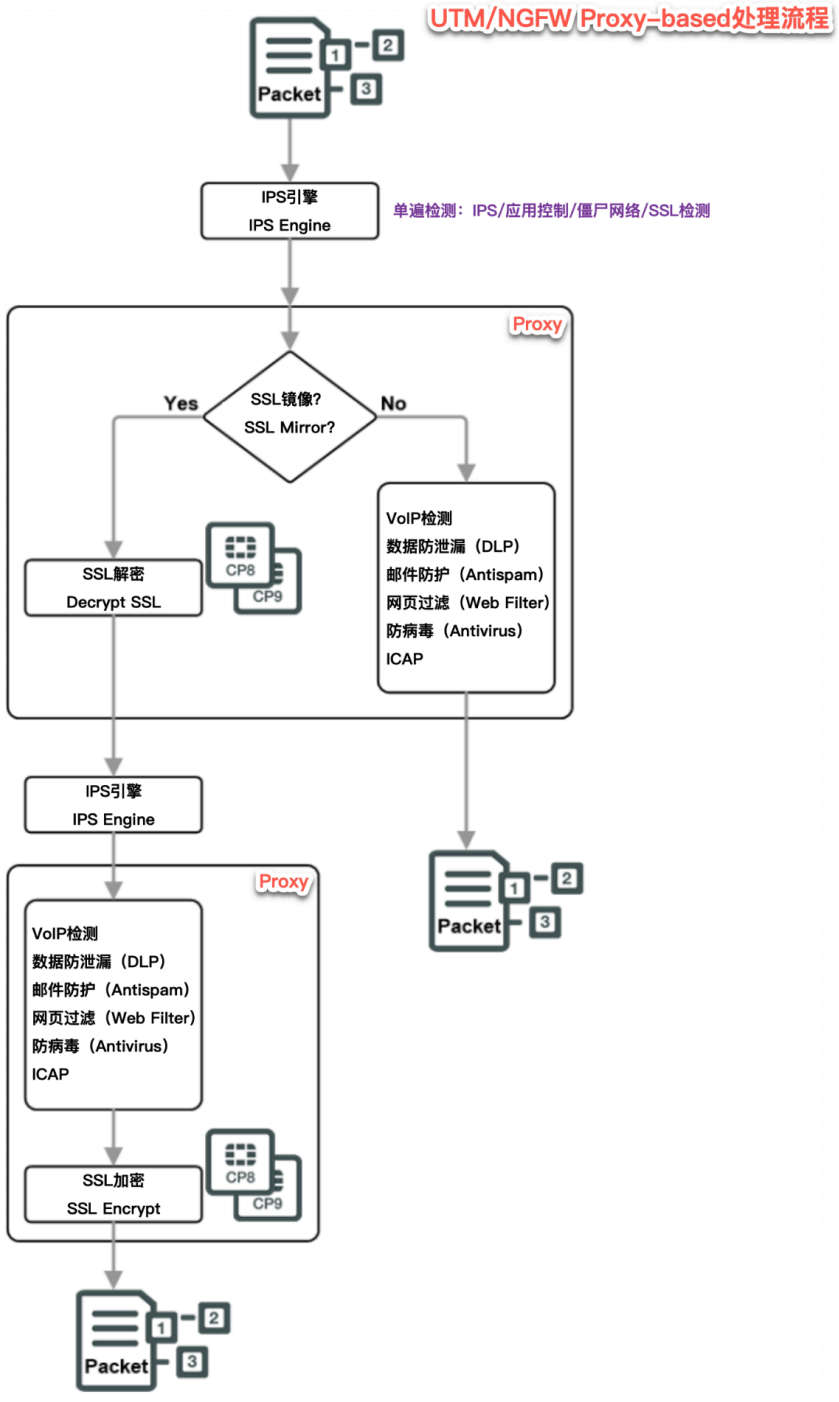
- 优点：资源占用低、不修改数据包（除阻断外）。
- 缺点：支持功能少于基于代理（如无客户端舒适、部分替换消息）。

基于代理的检测

若防火墙策略配置为基于代理的检测，则会同时进行基于流和基于代理的混合检测。数据包首先经过 IPS 引擎处理，该引擎采用与基于流的 UTM/NGFW 数据包流相同的步骤：通过基于流的检测执行单次通过式入侵防御、僵尸网络检测以及应用控制。

```bash
config firewall policy
    edit 1
        set inspection-mode proxy
    next
end
```

处理流程



基于流检查

流量首次进入后，与基于流的检查一致，执行单遍 IPS、应用控制、僵尸网络检查，SSL 解密由 CP 加速。

代理的深度处理

代理重构内容，按以下固定顺序检查：

- VoIP 检查：识别威胁并打开“针孔”允许合法语音流量。
- DLP：防止敏感数据泄露。
- 反垃圾邮件：过滤垃圾邮件。
- Web 过滤：拦截不合规网页。
- 防病毒：扫描病毒。
- ICAP：转发 HTTP/HTTPS 流量到 ICAP 服务器，作为“中间人”传递请求/响应。

SSL 加密与转发

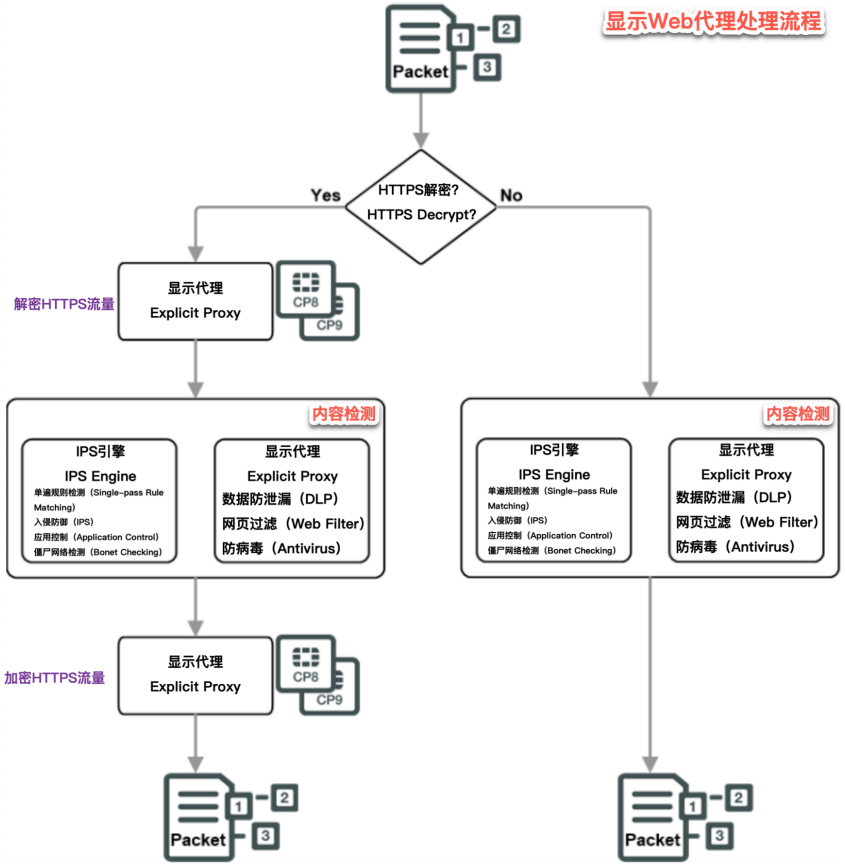
- 若为 SSL 流量：解密检查后重新加密（CP8/CP9 加速）；
- 转发逻辑：无威胁则转发，有威胁则阻断并替换为“替换消息”（如果配置了替换消息）。

显式 Web 代理

启用显式 Web 代理后，FortiGate 通过指定端口（默认 8080）监听 HTTP/HTTPS 会话，采用“并行检查”模式处理流量。

- 全局/VDOM 级别启用显式 Web 代理。
- 配置接口监听代理端口（默认 8080）。
- 创建显式 Web 代理策略定义匹配范围。

处理流程



HTTP 流量处理

- 并行检查：HTTP 流量同时发送到 IPS 引擎（IPS/僵尸网络/应用控制）和代理（DLP/Web 过滤/防病毒）。
- 转发逻辑：无威胁则转发，检测到威胁则阻断。

HTTPS 流量处理

- 代理解密 HTTPS 流量（CP 加速）。
- 并行检查（与 HTTP 一致）。
- 无威胁则重新加密（CP）并转发，有威胁则阻断。

源地址处理

数据包离开出口接口前，代理修改源地址：

- 普通模式：改为出口接口 IP。
- 透明模式：改为透明模式管理 IP。
- 可选配置：保留原始客户端 IP。

开局与日常维护

高频关注

运维注意事项

2GB 内存设备配置优化

常用维护命令

实施前准备

- 收集客户信息
- 设备软件检查
- 设备硬件检查

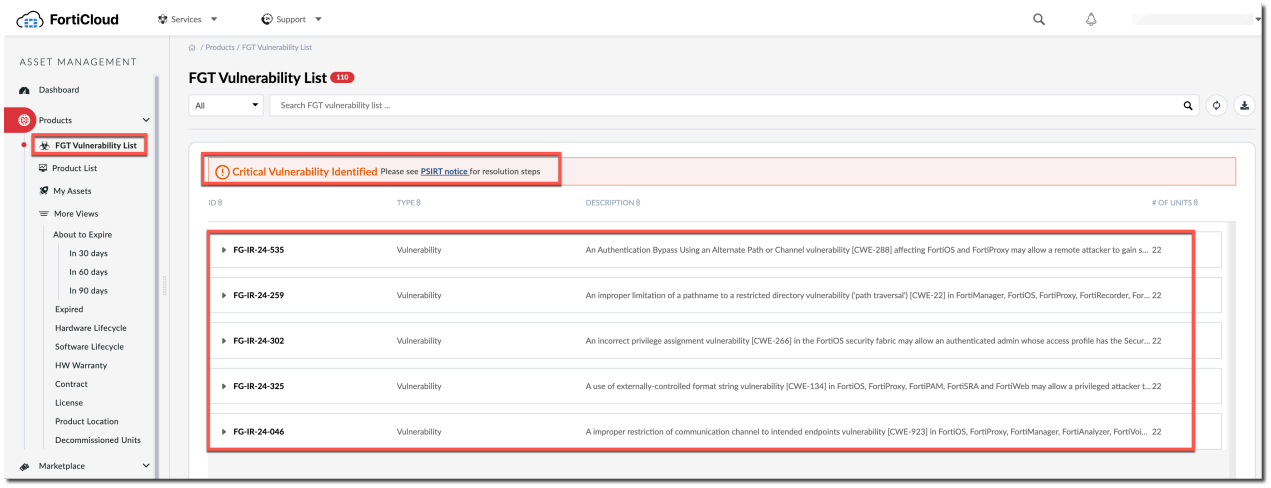
实施后设备运行检查

- 基础检查
- 运行情况
- 日志查看
- 运维监控

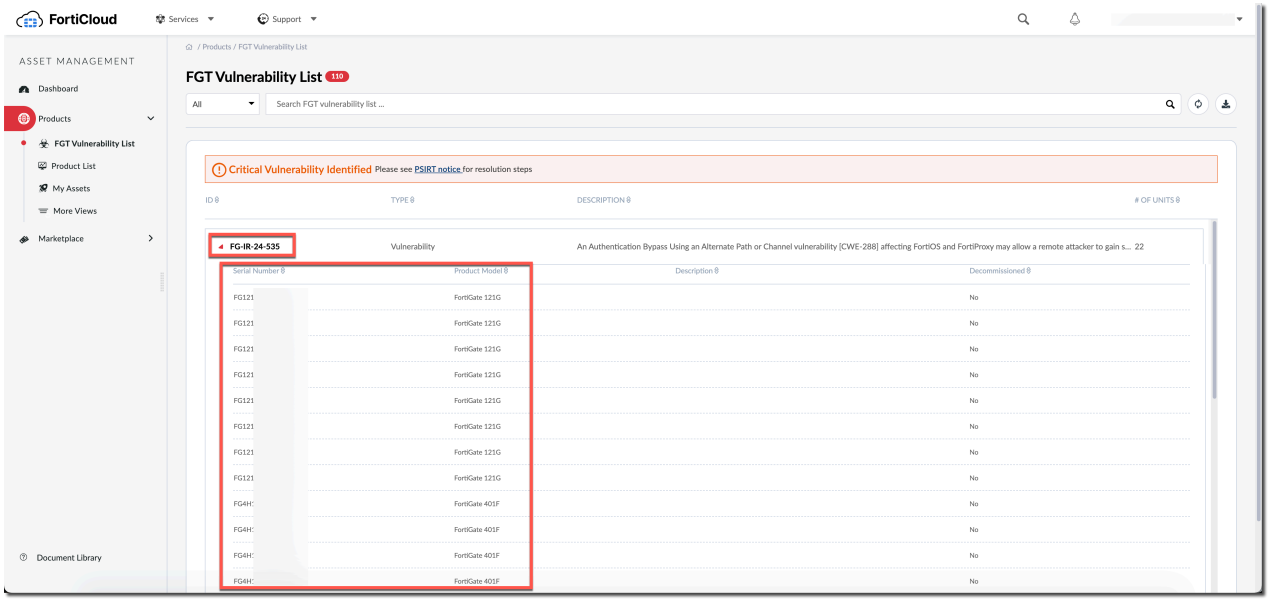
高频关注

版本漏洞查询

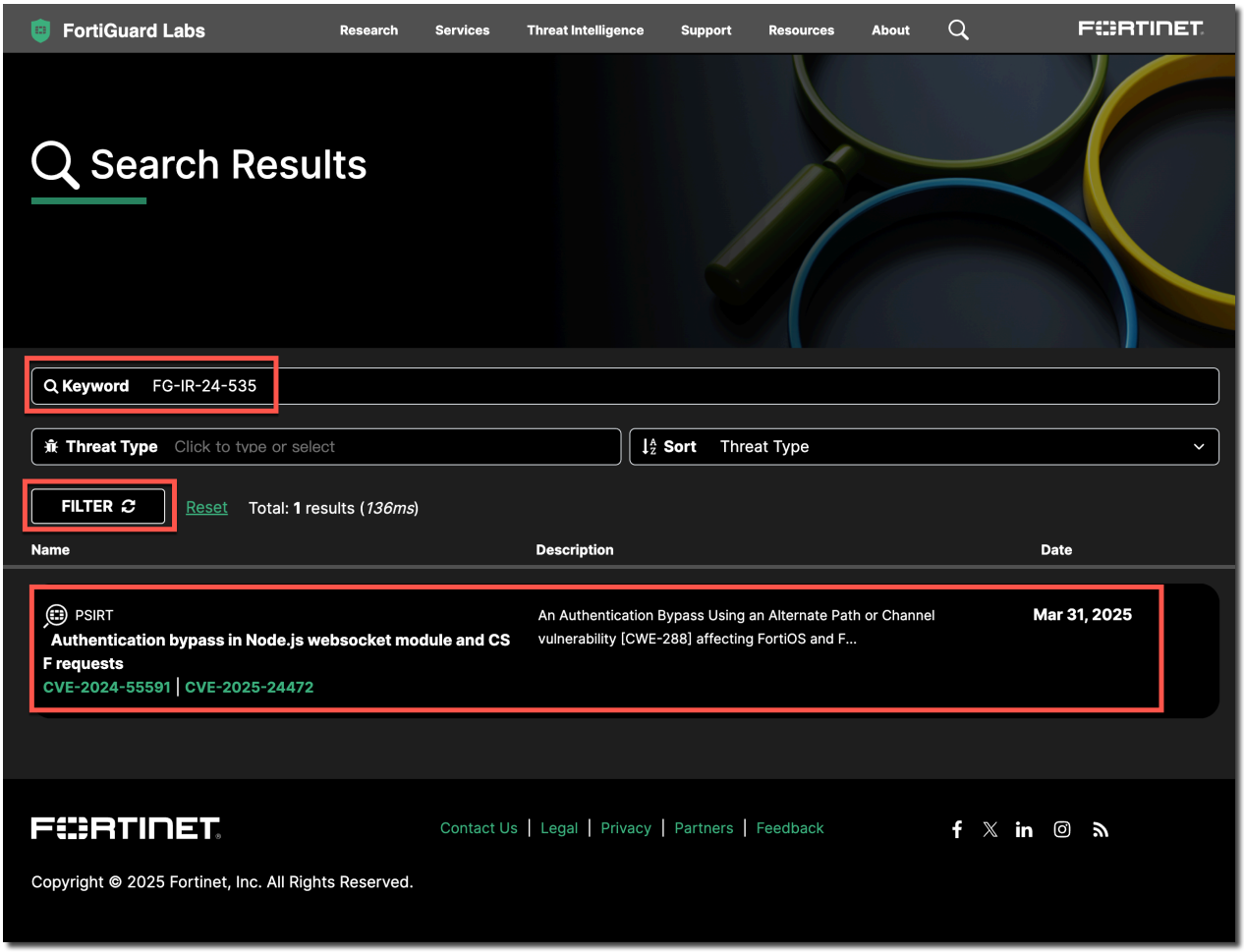
1. 使用设备注册的 FortiCloud 账户登录 https://support.fortinet.com/asset/#/views/products ↗ ，如果当前账户中检测到至少一个已存在漏洞的设备时，导航菜单中的“Product”中将会出现“FGT Vulnerability List”页面，该页面会显示所有相关漏洞信息。如下所示，FortiCloud 账户中存在 FortiOS 版本为 7.0.15 的设备，该版本存在的漏洞会被列出。



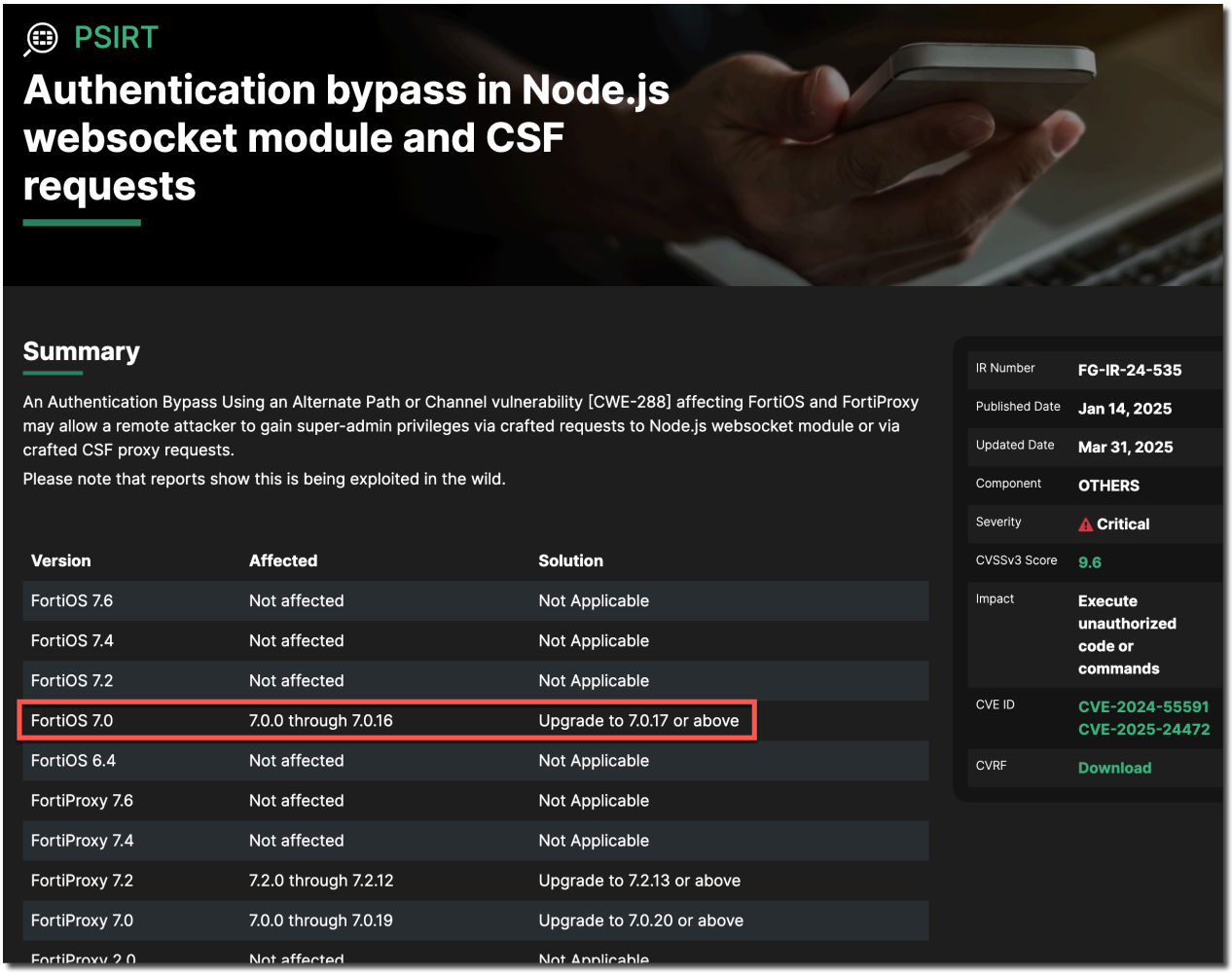
2. 点开某个漏洞 ID 的折叠项，漏洞列表包括已确定存在关键漏洞的设备列表以及漏洞相关信息。



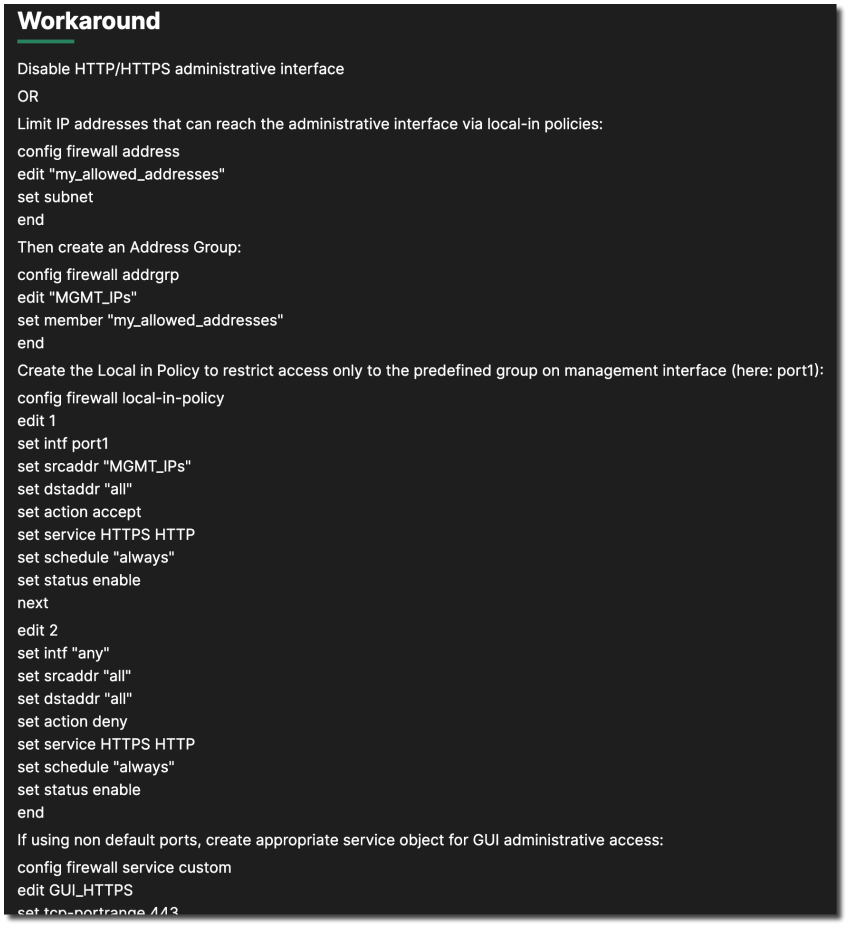
3. 如果您在以上页面中发现存在漏洞的设备版本，请访问 https://fortiguard.fortinet.com/search ↗ ，搜索漏洞 ID（如 `FG-IR-24-535` ）。



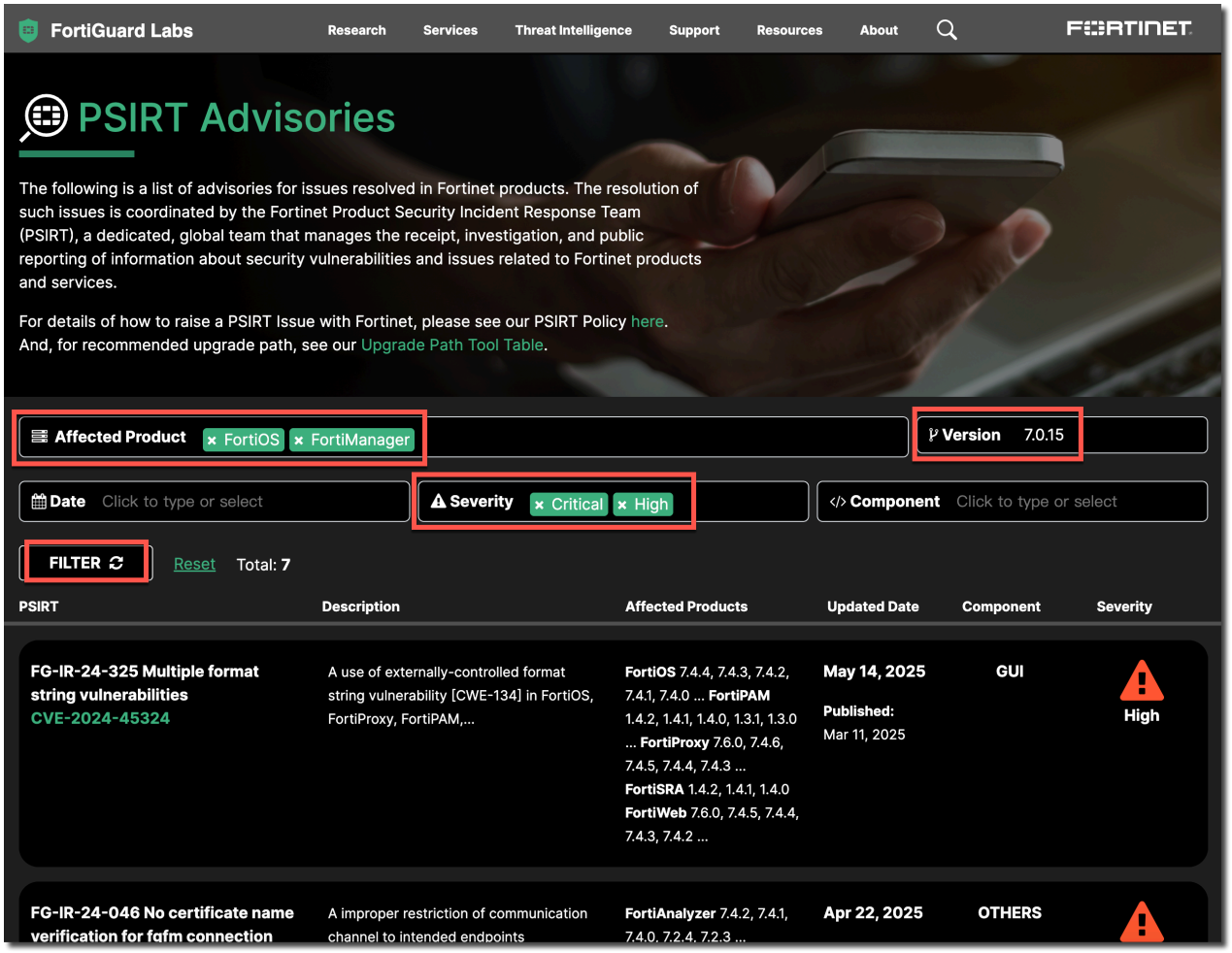
4. 点击进入该漏洞详情，可以查询该漏洞的升级解决方案、Workaround 等信息。如下所示，FortiOS 需要升级至 7.0.17 版本以解决该漏洞。



5. 也可以根据页面内的 Workaround（如果有）暂时规避该漏洞。



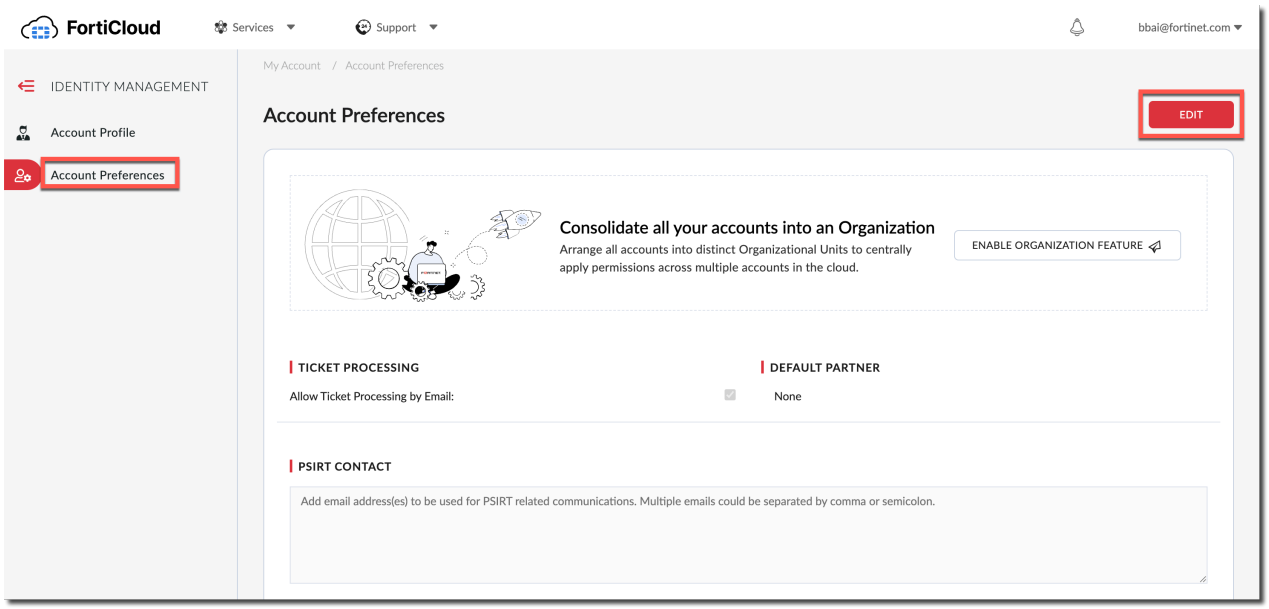
6. 访问此链接：https://www.fortiguard.com/psirt ↗ ，可以通过过滤特定产品/版本/严重程度等信息，查询相关的漏洞信息。



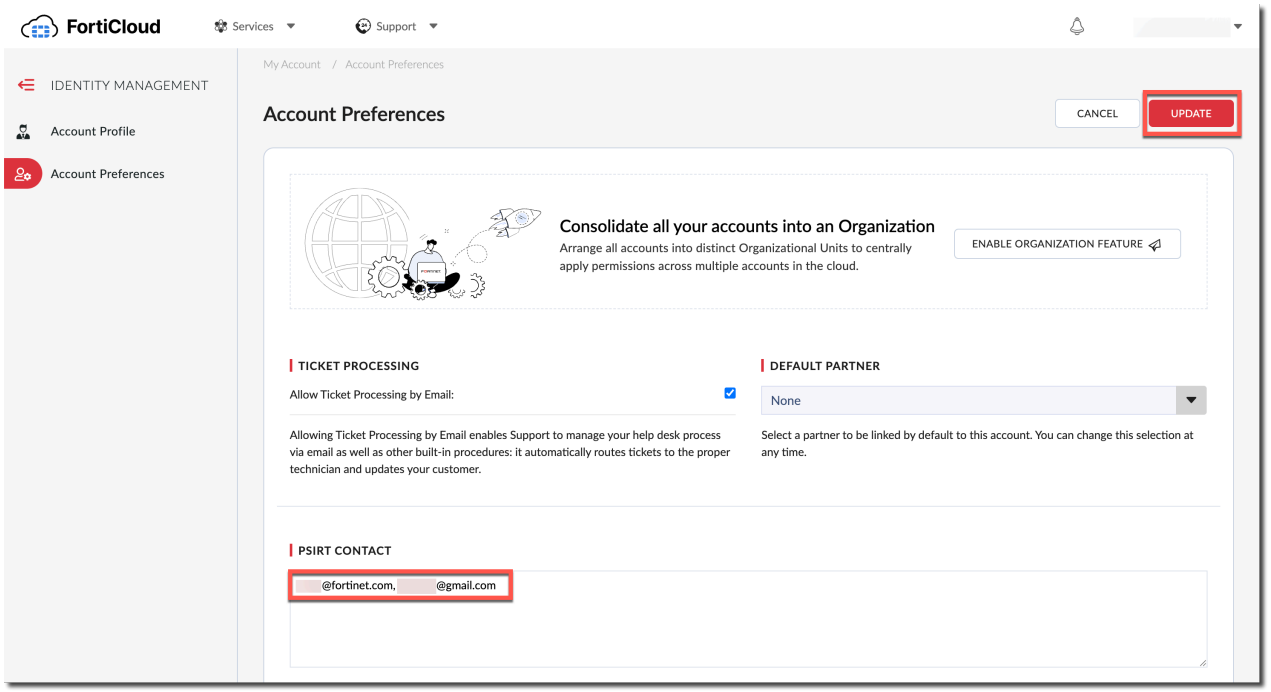
订阅漏洞更新

Email 订阅

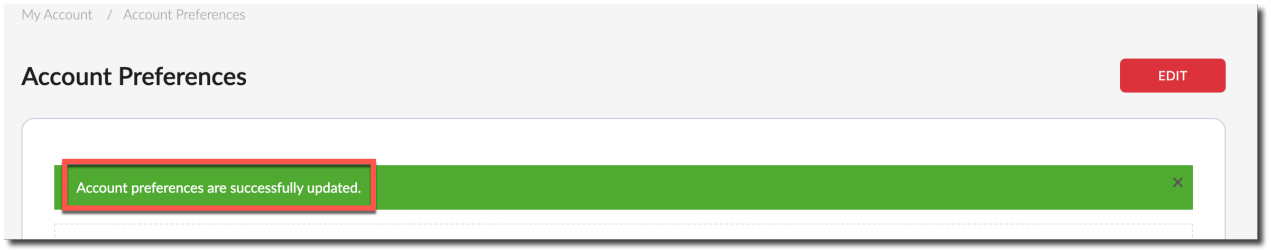
1. 使用设备注册的 FortiCloud 账户登录 https://support.fortinet.com/cred/#/sec/my-account/account-preferences ↗ （Account Preferences 页面），点页面击右上角的“EDIT”按钮。



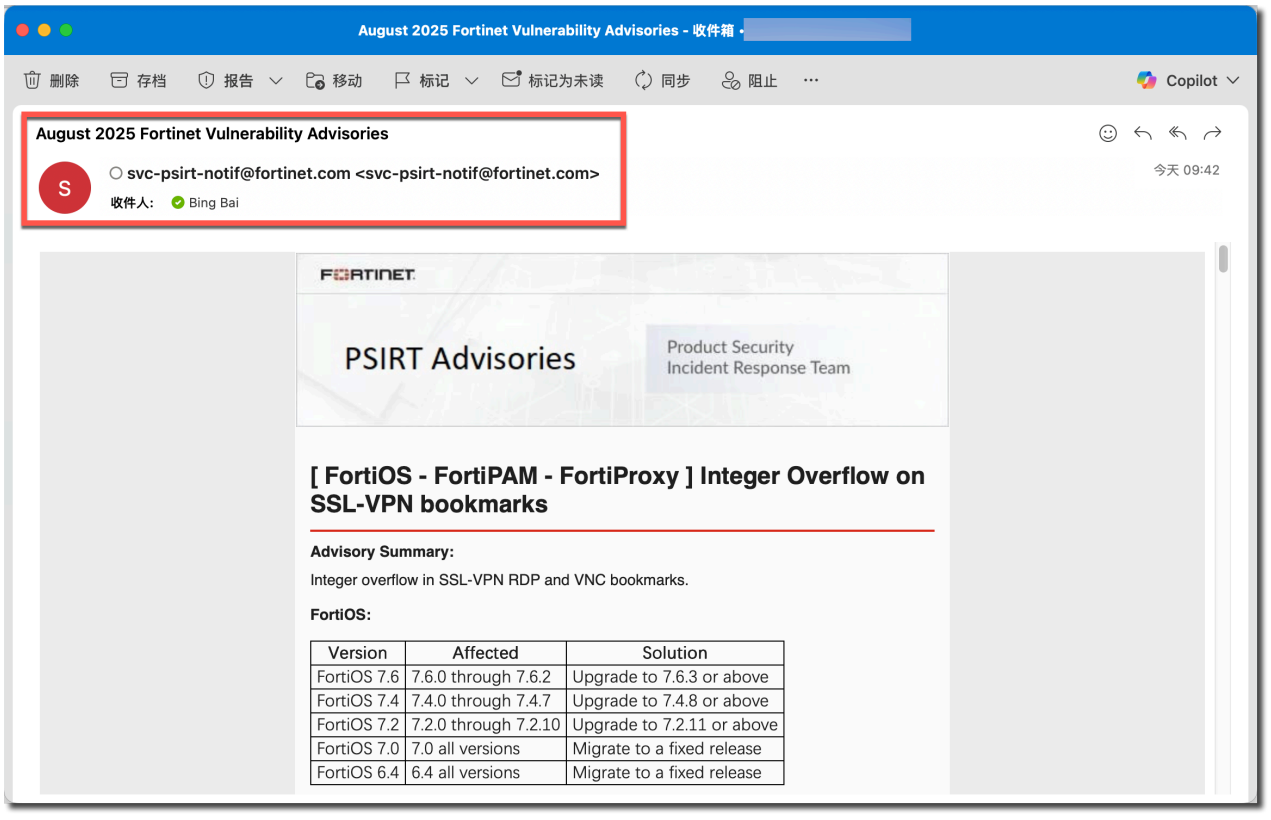
2. 在“PSIRT CONTACT”处填写需要接收漏洞公告的邮箱，可以填写多个邮箱，以英文`,`或`;`分隔。填写完成后，点击页面右上角的“UPDATE”按钮。



3. 页面提示账户更新成功。



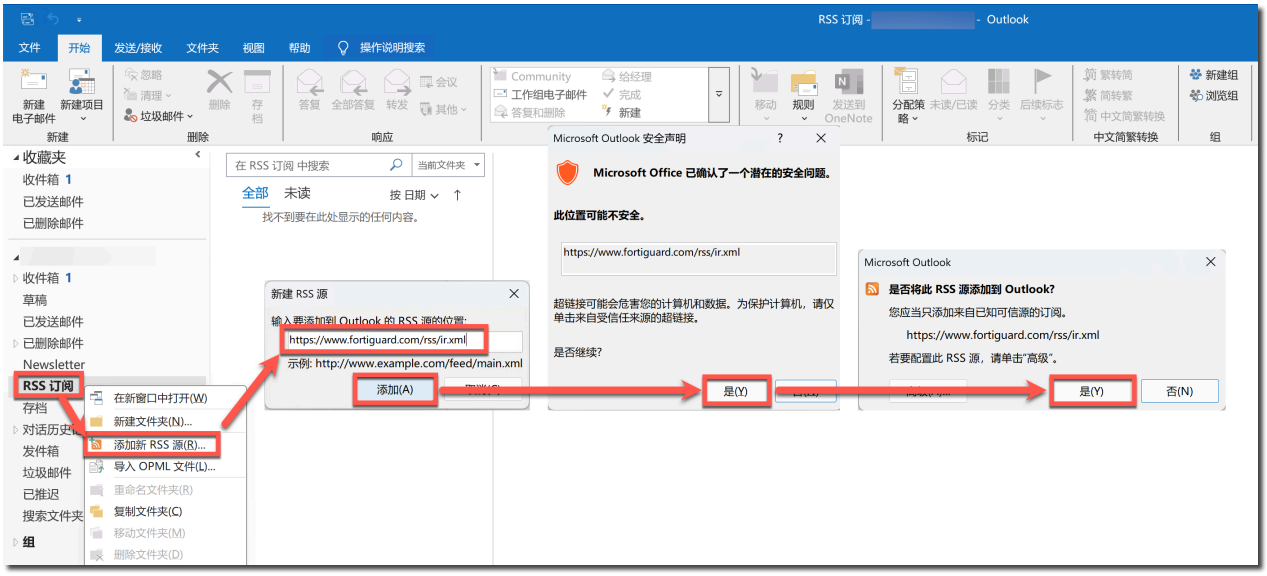
4. 在新的 PSIRT 更新发布后，相关邮箱可以收到 PSIRT 更新邮件，发件人为 `svc-psirt-notif@fortinet.com`。



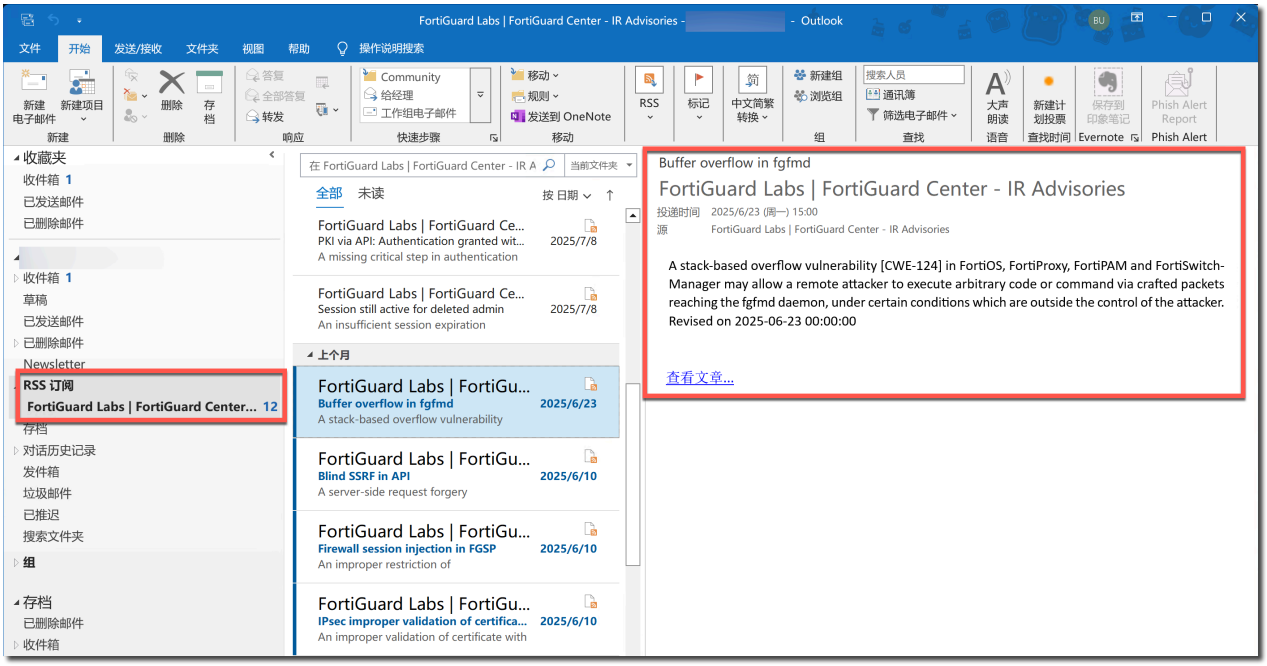
RSS 订阅

1. RSS 源（`https://www.fortiguard.com/rss/ir.xml`），包含所有漏洞的链接，可在所选 RSS 阅读器中进行通知。

2. 以 Outlook 为例，右键点击“开始 → RSS 订阅”，选择“添加新 RSS 源”，输入 FortiGuard PSIRT RSS 源（`https://www.fortiguard.com/rss/ir.xml`），点击“添加”按钮，随后在弹出的“Microsoft Outlook 安全声明”窗口中点击“是”，随后再次在弹出的窗口中点击“是”。

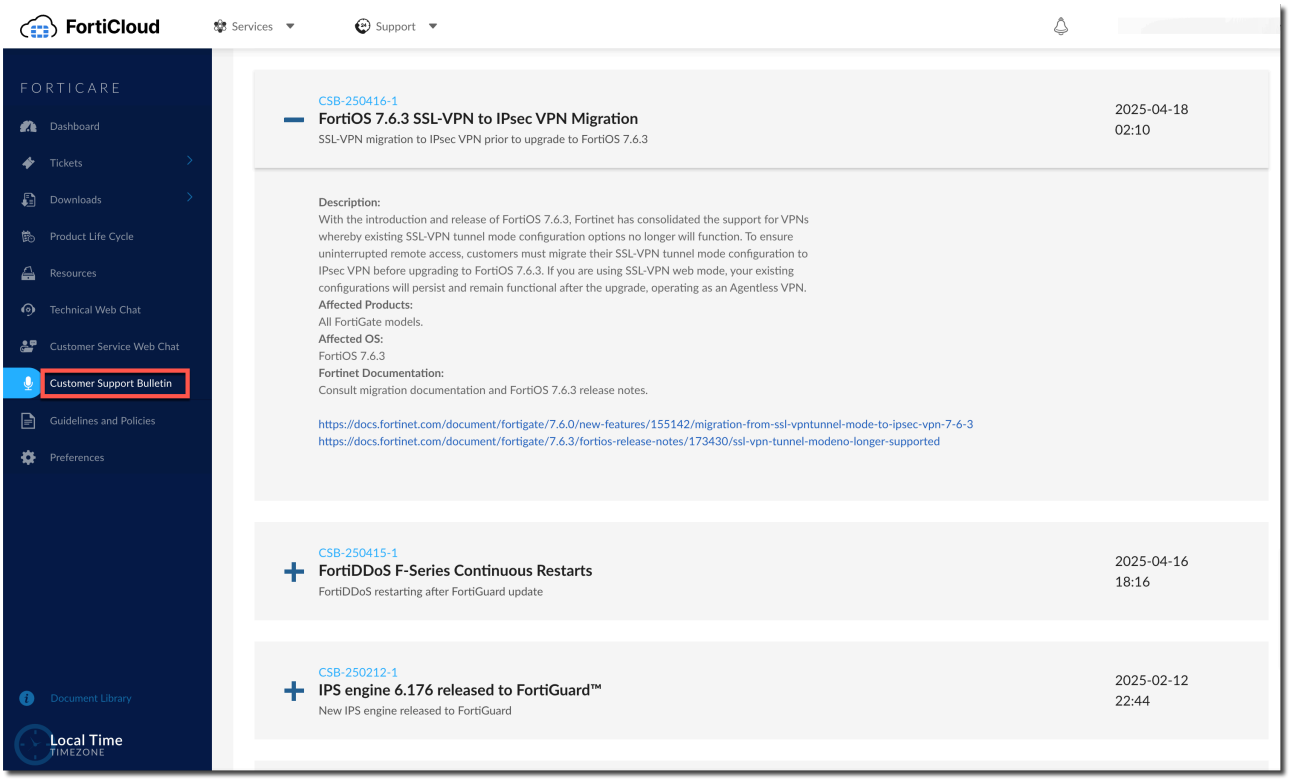


3. RSS 订阅添加成功，Outlook 自动下载了历史所有的 PSIRT 公告，并会在新的 PSIRT 公告发布后自动更新。



重要事件公告

使用设备注册的 FortiCloud 账户登录 https://support.fortinet.com/support/#/bulletin ↗，可以查看 Fortinet 各个产品发布的新功能、错误修复、版本漏洞发布、版本注意事项等信息。


FortiCloud
Services
Support
FORTICARE
Dashboard
Tickets
Downloads
Product Life Cycle
Resources
Technical Web Chat
Customer Service Web Chat
Customer Support Bulletin
Guidelines and Policies
Preferences
Document Library
Local Time
TIMEZONE
CSB-250416-1
FortiOS 7.6.3 SSL-VPN to IPsec VPN Migration
SSL-VPN migration to IPsec VPN prior to upgrade to FortiOS 7.6.3
2025-04-18
02:10
Description:
With the introduction and release of FortiOS 7.6.3, Fortinet has consolidated the support for VPNs whereby existing SSL-VPN tunnel mode configuration options no longer will function. To ensure uninterrupted remote access, customers must migrate their SSL-VPN tunnel mode configuration to IPsec VPN before upgrading to FortiOS 7.6.3. If you are using SSL-VPN web mode, your existing configurations will persist and remain functional after the upgrade, operating as an Agentless VPN.
Affected Products:
All FortiGate models.
Affected OS:
FortiOS 7.6.3
Fortinet Documentation:
Consult migration documentation and FortiOS 7.6.3 release notes.
https://docs.fortinet.com/document/fortigate/7.6.0/new-features/155142/migration-from-ssl-vpntunnel-mode-to-ipsec-vpn-7-6-3
https://docs.fortinet.com/document/fortigate/7.6.3/fortios-release-notes/173430/ssl-vpn-tunnel-modeno-longer-supported
CSB-250415-1
FortiDDoS F-Series Continuous Restarts
FortiDDoS restarting after FortiGuard update
2025-04-16
18:16
CSB-250212-1
IPS engine 6.176 released to FortiGuard™
New IPS engine released to FortiGuard
2025-02-12
22:44

运维注意事项

请参考运维注意事项↗。

2GB 内存设备配置优化

简介

本文介绍如何减少 2GB 内存入门级 FortiGate 设备在进入内存保护模式前的内存使用率。当前≤2GB 设备的型号如下：

30D/30E/30G/40C/40F/50E/50G/60D/60E/61E/60F/61F/70D/80C/80D/80E/81E/90D/91/92D

这类设备的一个突出问题是：基础内存占用就很高（约 50% ~ 70%），预留的内存空间有限。一旦遭遇突发事件，如网络流量突然增加、大量并发会话、攻击或短时高峰流量，内存极易飙升至阈值，从而触发“内存保护”模式。进入该模式后，设备会作为保护措施阻止新的连接请求，虽然能避免系统崩溃，但也可能影响业务连续性。这种情况在低端型号中尤为常见，因此必须特别关注内存的合理使用和优化。

查看内存大小

要检查 FortiGate 的内存大小与使用情况，可在设备 CLI 下运行 `diagnose hardware sysinfo memory`（或 `get hardware memory`、`fnsysctl cat/proc/meminfo`）命令，如下所示，FortiGate-60F 的总内存大小为 2GB：

```bash
FortiGate-60F # diagnose hardware sysinfo memory
MemTotal:        1963616 kB
MemFree:          493552 kB
Buffers:           39388 kB
Cached:           537824 kB
SwapCached:            0 kB
Active:           675400 kB
Inactive:         126488 kB
Active(anon):     485728 kB
Inactive(anon):    42588 kB
Active(file):     189672 kB
Inactive(file):    83900 kB
Unevictable:      190276 kB
Mlocked:           20400 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                76 kB
Writeback:             0 kB
AnonPages:        414976 kB
Mapped:           135568 kB
Shmem:            121484 kB
Slab:             209332 kB
SReclaimable:      16468 kB
SUnreclaim:       192864 kB
KernelStack:        3968 kB
PageTables:        31116 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:      981808 kB
Committed_AS:   13450496 kB
VmallocTotal:   260046784 kB
VmallocUsed:      126860 kB
VmallocChunk:   259812288 kB
```

优化建议

使用推荐版本或型号

- 建议在网络规划阶段评估业务需求，并根据性能要求选择合适规格的 FortiGate 型号，以避免后续扩容风险。设备选型请联系 [se_china@fortinet.com](mailto: se_china_fortinet.com)。
- 强烈建议在 FortiGate 上使用推荐的 FortiOS 版本。推荐版本通常包含对内存相关问题的修复和优化，可提升系统稳定性与可靠性。FortiGate 各型号版本推荐请参考：系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载↗章节。

FortiGuard 更新

1. 优化 FortiGuard 更新相关配置。

```bash
config system fortiguard
    set webfilter-cache-ttl 600              ←——减小WebFilter缓存时间，默认值3600
    set antispam-cache-ttl 600               ←——减小反垃圾邮件缓存时间，默认值1800
    set outbreak-prevention-cache-ttl 300    ←——减小病毒爆发防御缓存时间，默认值300
    set update-ffdb disable                  ←——只在设备未进入“内存保护模式”时更新ISDB
end
```

2. 修改 FortiGuard 更新间隔。

```bash
config system autoupdate schedule
    set frequency daily                      ←——将FortiGuard自动更新间隔增加到每天
    set time 03:00                           ←——FortiGuard自动更新时间为凌晨3:00
end
```

会话时间

1. 减少 TCP 会话相关状态的超时时间，更快释放内存。请根据 FortiGate 的实际流量情况调整以下值：

```bash
config system global
    set reset-sessionless-tcp disable
    set tcp-halfclose-timer 30  ←——减小TCP halfclose超时时间，默认值120
    set tcp-halfopen-timer 5    ←——减小TCP halfopen超时时间，默认值10
    set tcp-timewait-timer 0    ←——减小TCP timewait超时时间，默认值1
    set udp-idle-timer 60       ←——减小UDP超时时间，默认值180
end
```

2. 减少 TCP 会话和 DNS 会话的老化时间。

```bash
config system session-ttl
    set default 300
    config port
        edit 0
            set protocol 17
            set timeout 10
            set end-port 53
            set start-port 53
        next
    end
end
```

系统配置

1. 关闭安全评级相关功能。

```bash
config system global
    set internet-service-database on-demand        ←——仅下载已使用的ISDB条目
    set security-rating-result-submission disable  ←——关闭安全评级结果提交
    set security-rating-run-on-schedule disable    ←——关闭安全评级执行计划
end
```

2. 减少部分系统进程的数量。如果 FortiGate 上某些进程同时运行多个实例，应适当减少其数量（谨慎操作）。

```bash
config system global
    set miglogd-children 1
    set sslvpn-max-worker-count 1
    set wad-worker-count 1
    set scanunit-count 2
end
```

3. 减少 DNS 缓存条目的老化时间。

```bash
config system dns
    set dns-cache-limit 600  ←——DNS缓存条目最大数，默认值5000
    set dns-cache-ttl 600    ←——DNS缓存条目老化时间，默认值1800
end
```

日志相关

1. 关闭报告功能。对于 2GB 或 4GB 内存型号，不建议启用报告功能，会消耗较多比例的内存。

```bash
config report setting
    set pdf-report disable
    set fortiview disable
end

diagnose report-runner clean  ←——删除之前已生成的报告文件
```

2. 关闭内存日志，关闭硬盘日志记录流量日志，推荐将日志发送到 FortiAnalyzer 或 syslog 服务器。

```bash
config log memory setting
    set status disable
end

config log disk filter
    set forward-traffic disable
end
```

UTM 相关

1. 启用 IPS 的 `fail-open`，设备进入“内存保护模式”后 bypass 需要检测的流量。

```bash
config ips global
    set fail-open enable
    set cp-accel-mode none
end
```

2. 关闭 AV 启发式扫描。

```bash
config antivirus settings
    set machine-learning-detection disable
end
```

3. 在 FortiOS 7.4.2 及以后版本中，可以设置系统内存中允许用于 AV 扫描的最大百分比（10 - 50，默认值为 0）。设置为 0 表示禁用该功能。禁用后，AV 扫描对内存的使用将不受限制。

```bash
config ips global
    set av-mem-limit <0, 10-50>
end
```

4. 缺省的 AV 扫描文件大小为 10 MB，可能会导致内存消耗太多进入“内存保护模式”。根据病毒要求传播快的特点，一般病毒文件都不会太大，所以可以调整扫描文件的大小：
    - 配置自定义的 profile-protocol-options，设置各个应用的扫描阈值为 1 MB，大于 1 MB 的文件会被 AV bypass：

```bash
config firewall profile-protocol-options
    edit "av-profile-options"
        set oversize-log enable
        config http
            set ports 80
            set options clientcomfort servercomfort chunkedbypass
            unset post-lang
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
            set stream-based-uncompressed-limit 2
        end
        config ftp
            set ports 21
            set options clientcomfort splice
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
```

```
            set stream-based-uncompressed-limit 2
        end
        config imap
            set ports 143
            set options fragmail
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
        config mapi
            set ports 135
            set options fragmail
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
        config pop3
            set ports 110
            set options fragmail
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
        config smtp
            set ports 25
            set options fragmail splice
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
        config nntp
            set ports 119
            set options splice
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
        config ssh
            unset options
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
            set stream-based-uncompressed-limit 2
        end
        config dns
            set ports 53
        end
        config cifs
            set ports 445
            unset options
            set oversize-limit 1
            set uncompressed-oversize-limit 2
            set uncompressed-nest-limit 2
        end
    next
end
```

- 需要在防火墙策略内调用相应的 profile-protocol-options 才能生效。

```bash
config firewall policy
    edit <policy_id>
          set profile-protocol-options av-profile-options
    next
end
```

自动化重启 IPS

结合正常期间的内存占用，调整内存利用率高的阈值，当内存高时，自动触发 IPS engine 重启并释放内存。重启 IPS engine 不会影响业务流量。

1. 降低进入“内存保护模式”的 green 和 red 阈值，内存值到达 `memory-use-threshold-red` 设置的 75% 时，会进入 “内存保护模式”，当 FortiGate 进入 “内存保护模式”后，如果内存在 red 和 extreme 之间时，新会话不会被丢弃，继续转发，当内存到达 extreme 时开始丢弃新会话。

```bash
config system global
    set memory-use-threshold-extreme 95
    set memory-use-threshold-green 70  <———默认值82
    set memory-use-threshold-red 75    <———默认值88
end
```

2. 配置自动化触发器，触发条件为设备进入了“内存保护模式”。

```bash
config system automation-trigger
    edit "Enters_Conserve_Mode"
        set event-type low-memory
    next
end
```

3. 配置自动化动作，动作内容为重启 IPS Engine。

```bash
config system automation-action
    edit "RestartIPS"
        set action-type cli-script
        set script "config global    <———如果未开启多VDOM，则不需要此行
diagnose test application ipsmonitor 99"
        set accprofile "super_admin"
    next
end
```

4. 配置自动化工作流，绑定触发条件和动作。

```bash
config system automation-stitch
    edit "Restart_IPS_during_Conserve_Mode"
        set trigger "Enters_Conserve_Mode"
        config actions
            edit 1
                set action "RestartIPS"
            next
        end
    next
end
```

5. 补充：也可以设置一个周期性重启 IPS 的自动化（可以与上述脚本同时使用），如下所示，每天凌晨 4 点重启 IPS engine。

```bash
config system automation-trigger
    edit "4AM"
        set trigger-type scheduled
        set trigger-hour 4
    next
end

config system automation-action
    edit "RestartIPS"
        set action-type cli-script
        set script "config global    <———如果未开启多VDOM，则不需要此行
diagnose test application ipsmonitor 99"
        set accprofile "super_admin"
    next
end

config system automation-stitch
    edit "RestartIPS"
        set trigger "4AM"
        config actions
            edit 1
                set action "RestartIPS"
            next
        end
    next
end
```

内存保护模式收集信息

当设备进入内存保护模式时，请协助收集下列命令的输出结果，并提交给 TAC 团队（需要 super_admin 权限）。

```bash
fnsysctl date
get system status
get hardware status
diagnose autoupdate versions
get sys perf status
diag sys session stat
get sys perf firewall statistics
diag hardware sysinfo memory
diag hardware sysinfo slab
diagnose hardware sysinfo shm
diag sys vd list | grep fib
diagnose sys mpstat 1 5
diag sys top-all 2 50
diag sys top-mem 20
diagnose sys top-fd 20
diag snmp ip frags
diagnose ips session status
diagnose ips memory status
diagnose ips raw status
diagnose ips session performance
diagnose ips session list by-mem
fnsysctl df -k
fnsysctl ls -l /tmp
fnsysctl du -i /tmp
fnsysctl du -a /tmp
fnsysctl du -a / -d 1
fnsysctl du -i /dev/shm
fnsysctl du -a /dev/shm
fnsysctl du -i /node-scripts
fnsysctl du -a /node-scripts
```

注意事项

1. 上述优化配置的具体参数应根据您的实际网络状态进行调整。
2. 从 FortiOS 7.4.4 版本开始，从 ≤ 2GB 内存的设备上移除了代理相关的功能。
3. 从 FortiOS 7.6.3 版本开始，从 ≤ 2GB 内存的设备上移除了部分 Security Fabric 相关功能。

常用维护命令

请参考：常用维护命令手册 ↗。

实施前准备

- 收集客户信息
- 设备软件检查
- 设备硬件检查

收集客户信息

在项目实施前，请先确认以下关键信息是否有遗漏，并提前做好上点规划，避免上线后出现问题。同时为后续可能出现故障问题能快速排查定位打下铺垫。

1. 网络拓扑信息（了解网络拓扑信息有助于网络方案的规划）

2. 环境信息（了解部署位置、部署模式、最大吞吐、最大用户数有助于对设备性能的评估）

3. 客户需求，对 FortiGate 部署的功能要求（了解客户要求部署上 FortiGate 后需要开启哪些功能和访问控制策略以明确配置思路）

4. FortiGate 设备的 IP 地址规划（了解 FortiGate 设备部署时所需要的基本网络配置，如 IP 地址、子网掩码、网关、路由、DNS 以确定具体配置内容）

5. FortiGate 接入链路信息（了解现网的链路及接口状态，提前确认设备接口是否能够满足）

6. 收集好割接前后的业务访问信息，如 ping 包延时、网页打开速度等信息，割接后对网络的性能做对比 。

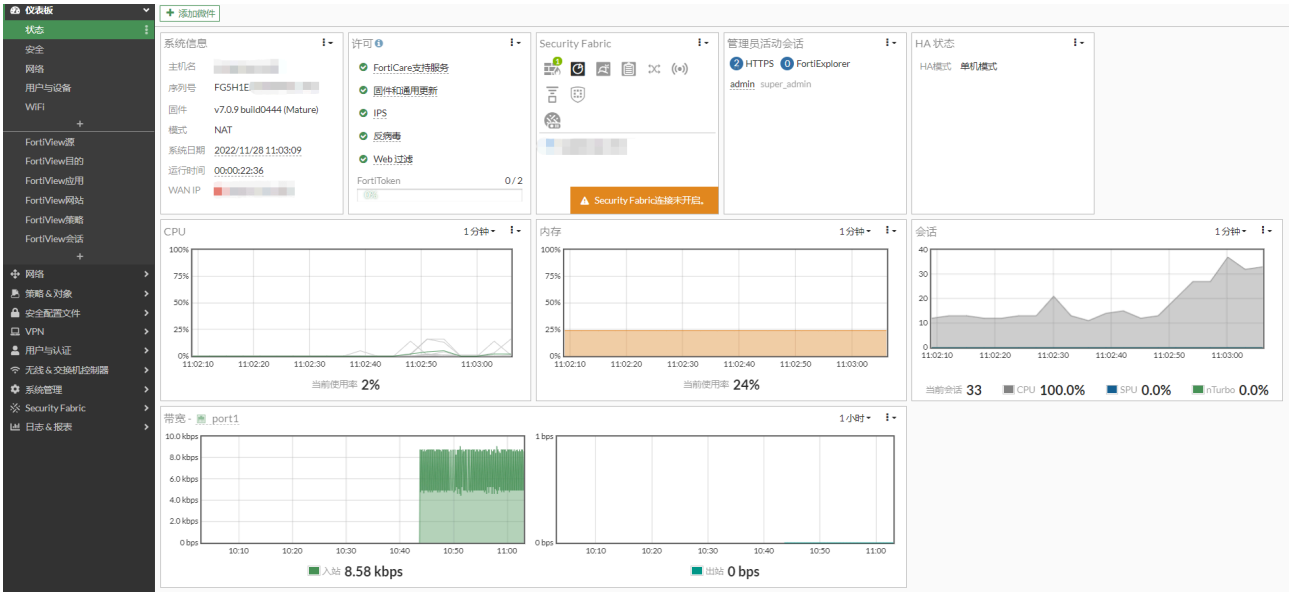
记录项目	ping包延时	
	平时	高峰
割接前	网站Ping延时 ping www.sina.com 延时：XXms ping www.baidu.com 延时：XXms	网站Ping延时 ping www.sina.com 延时：XXms ping www.baidu.com 延时：XXms
割接后	网站Ping延时 ping www.sina.com 延时：XXms ping www.baidu.com 延时：XXms 与割接前进行对比，确认延时浮动是否会超过10ms以上	网站Ping延时 ping www.sina.com 延时：XXms ping www.baidu.com 延时：XXms 与割接前进行对比，确认延时浮动是否会超过10ms以上

记录项目	网页打开速度	
	平时	高峰
割接前	记录打开网页的体验 打开 www.163.com 延时：XXms 打开 www.sina.com 延时：XXms	记录打开网页的体验 打开 www.163.com 延时：XXms 打开 www.sina.com 延时：XXms
割接后	记录打开网页的体验 打开 www.163.com 延时：XXms 打开 www.sina.com 延时：XXms 与割接前进行对比，确认网页打开时间是否会比割接前慢5秒以上	记录打开网页的体验 打开 www.163.com 延时：XXms 打开 www.sina.com 延时：XXms 与割接前进行对比，确认网页打开时间是否会比割接前慢5秒以上

设备软件检查

首页信息

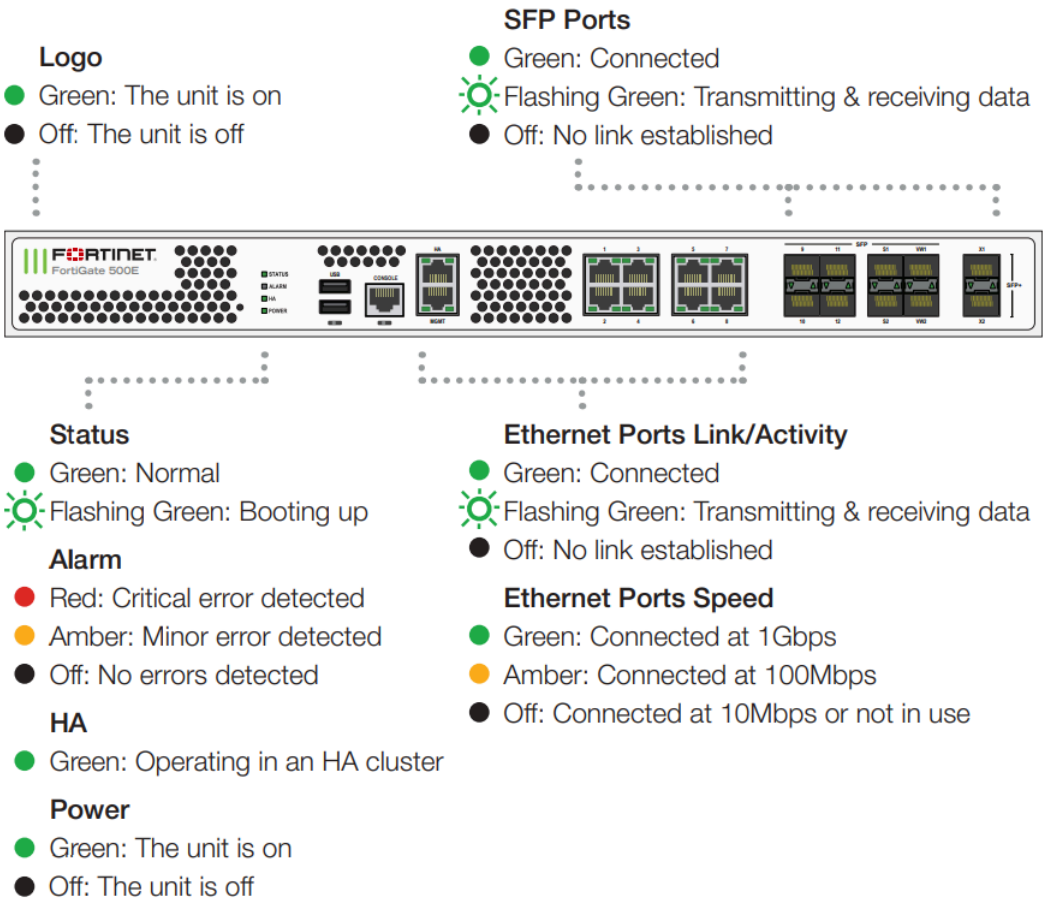
进入系统首页可以产看 设备的型号、硬件序列号、软件序列号、系统版本、运行时间、系统时间、HA 模式、运行模式、接口状态、系统资源：cpu、内存、磁盘使用情况，告警信息等。



设备硬件检查

环境/指示灯维护

1. 清洁状况：设备外壳、设备内部、机架内各防尘网、地板等都应干净整洁无明显尘土附着。注意机柜防尘网的清洁状况，及时清洗更换，以免影响机柜门及风扇框的通风散热。
2. 散热状况：设备正常工作时，要求保持风扇正常运转，擅自关闭风扇会引起设备温度升高，并可能损坏设备。不要在设备子架上通风口处放置杂物，还应定期清理风扇的防网。
3. 设备指示灯及状态，请参考设备硬件指南：https://docs.fortinet.com/product/fortigate/hardware ↗ ，如 FortiGate 500E



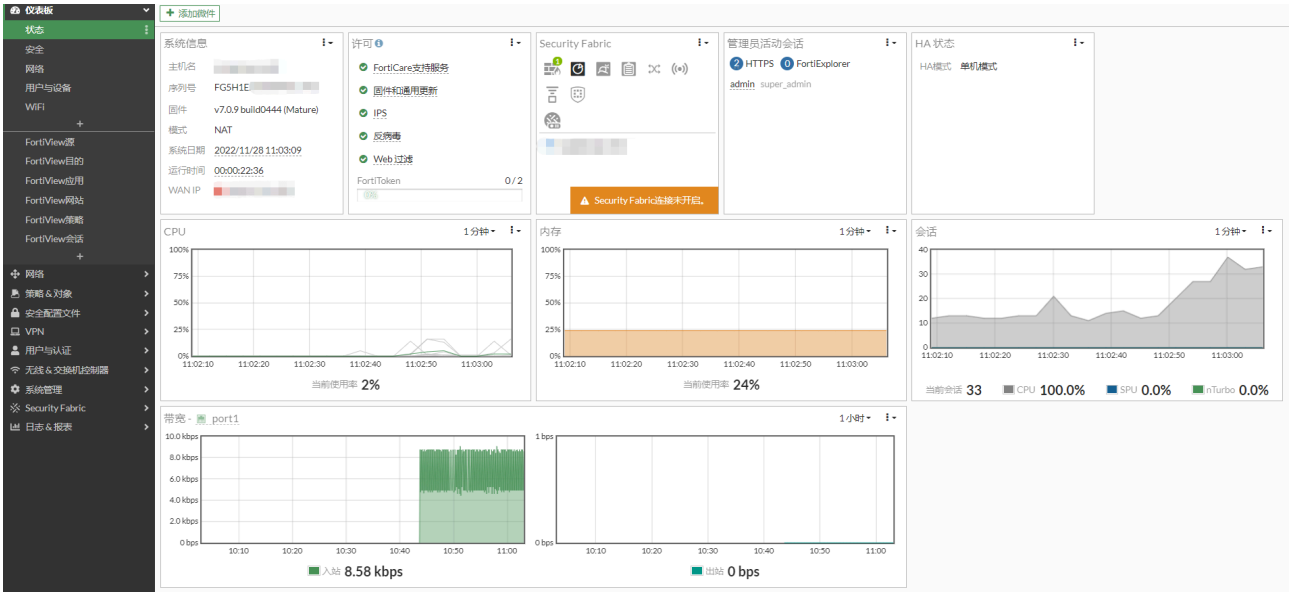
实施后设备运行检查

- 基础检查
- 运行情况
- 日志查看
- 运维监控

基础检查

在 "仪表板"-->"状态" 界面，通过添加 "微件" 的方式，如接口流量，如下图，查看 cpu、内存、接口流量和会话数。

CPU 和内存利用率通常应低于 80%，如果 CPU 或者内存长时间利用率高，应分析原因。



运行情况

使用命令行的方式查看更详细的信息

1. 查看系统基本信息

```
# get system status
Version: FortiGate-501E v7.0.9,build0444,221121 (GA.M)
Firmware Signature: certified
Virus-DB: 90.08232(2022-11-27 22:26)
Extended DB: 90.08232(2022-11-27 22:26)
Extreme DB: 1.00000(2018-04-09 18:07)
AV AI/ML Model: 2.08576(2022-11-27 21:45)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 22.00446(2022-11-24 01:49)
APP-DB: 22.00444(2022-11-22 02:49)
INDUSTRIAL-DB: 6.00741(2015-12-01 02:30)
IPS Malicious URL Database: 4.00543(2022-11-27 12:14)
Serial-Number: FG5H1Exxxxxx
BIOS version: 05000005
System Part-Number: P21592-08
Log hard disk: Available
Hostname: FGTxxxxxx
Private Encryption: Disable
Operation Mode: NAT
Current virtual domain: root
Max number of virtual domains: 10
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: disable
Current HA mode: standalone
Branch point: 0444
Release Version Information: GA
FortiOS x86-64: Yes
System time: Mon Nov 28 14:49:17 2022
Last reboot reason: warm reboot
```

2. 查看硬件状态

```
# get hardware status
Model name: FortiGate-501E
ASIC version: CP9
ASIC SRAM: 64M
CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Number of CPUs: 8
RAM: 16047 MB
Compact Flash: 15331 MB /dev/sdc
Hard disk: 228936 MB /dev/sda
USB Flash: not available
Network Card chipset: Intel(R) Gigabit Ethernet Linux Driver (rev.0003)
Network Card chipset: FortiASIC NP6 Adapter (rev.)
```

3. 查看 CPU 和内存利用率

```
# get system performance status
CPU states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU0 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU1 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU2 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU3 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU4 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU5 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU6 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU7 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
Memory: 16432720k total, 4086464k used (24.9%), 11961200k free (72.8%), 385056k freeable (2.3%)
Average network usage: 191 / 2 kbps in 1 minute, 263 / 11 kbps in 10 minutes, 268 / 4 kbps in 30 minutes
Maximal network usage: 283 / 42 kbps in 1 minute, 553 / 1422 kbps in 10 minutes, 553 / 1422 kbps in 30 minutes
Average sessions: 11 sessions in 1 minute, 10 sessions in 10 minutes, 8 sessions in 30 minutes
Maximal sessions: 16 sessions in 1 minute, 19 sessions in 10 minutes, 19 sessions in 30 minutes
```

```
Average session setup rate: 0 sessions per second in last 1 minute, 0 sessions per second in last 10 minutes, 0 sessions
per second in last 30 minutes
Maximal session setup rate: 3 sessions per second in last 1 minute, 3 sessions per second in last 10 minutes, 3 sessions
per second in last 30 minutes
Average NPU sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Maximal NPU sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Average nTurbo sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Maximal nTurbo sessions: 0 sessions in last 1 minute, 0 sessions in last 10 minutes, 0 sessions in last 30 minutes
Virus caught: 0 total in 1 minute
IPS attacks blocked: 0 total in 1 minute
Uptime: 0 days,  4 hours,  10 minutes
```

4. 查看会话状态

```
# diagnose sys session full-stat
session table:           table_size=4194304 max_depth=1 used=66
misc info:       session_count=35 setup_rate=0 exp_count=0 clash=0
        memory_tension_drop=0 ephemeral=0/1113088 removeable=0 extreme_low_mem=0
        npu_session_count=0
        nturbo_session_count=0
delete=6, flush=5, dev_down=98/137 ses_walkers=0
TCP sessions:
         11 in ESTABLISHED state
         1 in TIME_WAIT state
firewall error stat:
error1=00000000
error2=00000000
error3=00000000
error4=00000000
tt=00000000
cont=00000000
ips_recv=00000000
policy_deny=00006bae
av_recv=00000000
fqdn_count=00000001
fqdn6_count=00000000
```

5. 查看 HA 状态

```
# get system ha status
HA Health Status: OK
Model: FortiGate-501E
Mode: HA A-P
Group: 0
Debug: 0
Cluster Uptime: 0 days 0:21:11
Cluster state change time: 2022-11-28 15:13:06
Primary selected using:
    <2022/11/28 15:13:06> FG5H1ExxxxxxB is selected as the primary because its uptime is larger than peer member
FG5H1E5819904211.
    <2022/11/28 15:13:06> FG5H1ExxxxxxA is selected as the primary because its override priority is larger than peer
member FG5H1E5819904036.
    <2022/11/28 15:10:15> FG5H1ExxxxxxA is selected as the primary because it's the only member in the cluster.
ses_pickup: disable
override: disable
Configuration Status:
    FG5H1ExxxxxxA(updated 4 seconds ago): out-of-sync
    FG5H1ExxxxxxB(updated 0 seconds ago): in-sync
System Usage stats:
    FG5H1ExxxxxxA(updated 4 seconds ago):
        sessions=11, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=24%
    FG5H1ExxxxxxB(updated 0 seconds ago):
        sessions=17, average-cpu-user/nice/system/idle=0%/0%/1%/98%, memory=21%
HBDEV stats:
    FG5H1ExxxxxxA(updated 4 seconds ago):
        port6: physical/1000full, up, rx-bytes/packets/dropped/errors=47426063/99344/0/0, tx=461507/1005/0/0
    FG5H1ExxxxxxB(updated 0 seconds ago):
        port6: physical/1000full, up, rx-bytes/packets/dropped/errors=1169224/2536/0/0, tx=3000908/6578/0/0
Secondary   : FGT1, FG5H1ExxxxxxA, HA cluster index = 0
```

```
Primary     : FGT2, FG5H1ExxxxxxB, HA cluster index = 1
number of vcluster: 1
vcluster 1: standby 169.254.0.2
Secondary: FG5H1ExxxxxxA, HA operating index = 1
Primary: FG5H1ExxxxxxA, HA operating index = 0
```

6. 查看 crashlog 是否异常

```
# diagnose debug crashlog read
1: 2022-11-23 17:32:57 the killed daemon is /bin/sflowd: status=0×0
2: 2022-11-23 17:34:55 the killed daemon is /bin/dhcpd: status=0×0
3: 2022-11-23 01:38:40 the killed daemon is /bin/getty: status=0×0
4: 2022-11-23 01:41:32 the killed daemon is /bin/fgfmd: status=0×0
5: 2022-11-28 10:41:47 the killed daemon is /bin/sflowd: status=0×0
Crash log interval is 3600 seconds
Max crash log line number: 16384
```

7. 查看进程状态

```
# diagnose sys top 3 100 3    可判断每个进程的cpu和内存的利用率，敲一次c 按cpu高低排序；敲一次m，按mem高低排序，敲q退出
第一个3是间隔3s，100是显示的进程数，最后一个3是执行3次
Run Time:  0 days, 4 hours and 16 minutes
0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 16047T, 11679F
       ipshelper      228      S <     0.0     1.6    0
       ipsengine      357      S <     0.0     0.5    1
       ipsengine      360      S <     0.0     0.5    4
       ipsengine      363      S <     0.0     0.5    6
       ipsengine      358      S <     0.0     0.5    2
       ipsengine      361      S <     0.0     0.5    7
       ipsengine      362      S <     0.0     0.5    5
       ipsengine      359      S <     0.0     0.5    3
            node      215      S       0.0     0.3    1
         cmdbsvr      180      S       0.0     0.3    2
         miglogd      234      S       0.0     0.2    0
             wad      284      S       0.0     0.2    7
             wad      283      S       0.0     0.2    4
             wad      285      S       0.0     0.2    2
             wad      286      S       0.0     0.2    0
             wad      287      S       0.0     0.2    3
             wad      289      S       0.0     0.2    2
             wad      290      S       0.0     0.2    0
             wad      288      S       0.0     0.2    4
       scanunitd      238      S <     0.0     0.2    2
         miglogd      298      S       0.0     0.2    4
         miglogd      299      S       0.0     0.2    1
         miglogd      301      S       0.0     0.2    1
         miglogd      300      S       0.0     0.2    6
       forticron      222      S       0.0     0.1    2
       extenderd      266      S       0.0     0.1    0
          cw_acd      257      S       0.0     0.1    0
         updated      239      S       0.0     0.1    5
           fgfmd      256      S       0.0     0.1    0
          httpsd      214      S       0.0     0.1    4
         reportd      235      S       0.0     0.1    5
           newcli    1330      S       0.0     0.1    6
       scanunitd     1318      S <     0.0     0.1    6
       scanunitd     1319      S <     0.0     0.1    1
       scanunitd     1320      S <     0.0     0.1    6
       scanunitd     1321      S <     0.0     0.1    6
       scanunitd     1322      S <     0.0     0.1    2
       scanunitd     1323      S <     0.0     0.1    0
       scanunitd     1324      S <     0.0     0.1    2
       scanunitd     1325      S <     0.0     0.1    4
       forticldd      223      S       0.0     0.1    2
 initXXXXXXXXXXX        1      S       0.0     0.1    2
        dnsproxy      268      S       0.0     0.1    2
             wad      281      S       0.0     0.0    6
          cu_acd      261      S       0.0     0.0    0
          httpsd     1300      S       0.0     0.0    2
          fnbamd      220      S       0.0     0.0    2
           autod      267      S       0.0     0.0    2
```

```
           authd      224      S       0.0     0.0    0
          newcli     1357      R       0.0     0.0    1
           authd      312      S       0.0     0.0    5
           authd      311      S       0.0     0.0    2
           authd      313      S       0.0     0.0    1
           authd      314      S       0.0     0.0    2
       eap_proxy      255      S       0.0     0.0    1
      wf_monitor      245      S       0.0     0.0    2
         foauthd      225      S       0.0     0.0    0
             wad      237      S       0.0     0.0    6
            sshd      247      S       0.0     0.0    2
           voipd      236      S       0.0     0.0    2
           snmpd      240      S       0.0     0.0    6
          fcnacd      230      S       0.0     0.0    0
             nsm      198      S       0.0     0.0    0
         syslogd      219      S       0.0     0.0    2
          flcfgd      262      S       0.0     0.0    2
           radvd      251      S       0.0     0.0    4
           dhcpd      241      S       0.0     0.0    6
       urlfilter      377      S <     0.0     0.0    2
         wpad_ac      259      S       0.0     0.0    4
     fortilinkd      260      S       0.0     0.0    1
 zebos_launcher      193      S       0.0     0.0    2
          quard      250      S       0.0     0.0    0
  cw_acd_helper      258      S       0.0     0.0    7
           ntpd      246      S <     0.0     0.0    0
       dnsproxy      253      S       0.0     0.0    6
         lldptx      243      S       0.0     0.0    3
   ipmc_sensord      212      S       0.0     0.0    2
           sshd     1329      S       0.0     0.0    0
         lldprx      242      S       0.0     0.0    2
         flpold      263      S       0.0     0.0    6
        uploadd      211      S       0.0     0.0    6
     ipsmonitor      217      S       0.0     0.0    5
            acd      244      S       0.0     0.0    2
          getty      216      S <     0.0     0.0    4
      alertmail      252      S       0.0     0.0    6
            fsd      265      S       0.0     0.0    0
          fsvrd      249      S       0.0     0.0    2
            fas      229      S       0.0     0.0    6
            wad      279      S       0.0     0.0    4
        telnetd      248      S       0.0     0.0    3
merged_daemons      218      S       0.0     0.0    6
      clearpass      226      S       0.0     0.0    1
      fsso_ldap      232      S       0.0     0.0    1
       kmiglogd      213      S       0.0     0.0    2
          xcvrd      233      S       0.0     0.0    4
        httpclid      227      S       0.0     0.0    2
      fclicense      221      S       0.0     0.0    6
            imi      210      S       0.0     0.0    6
           bgpd      203      S <     0.0     0.0    5
           pimd      205      S       0.0     0.0    0
```

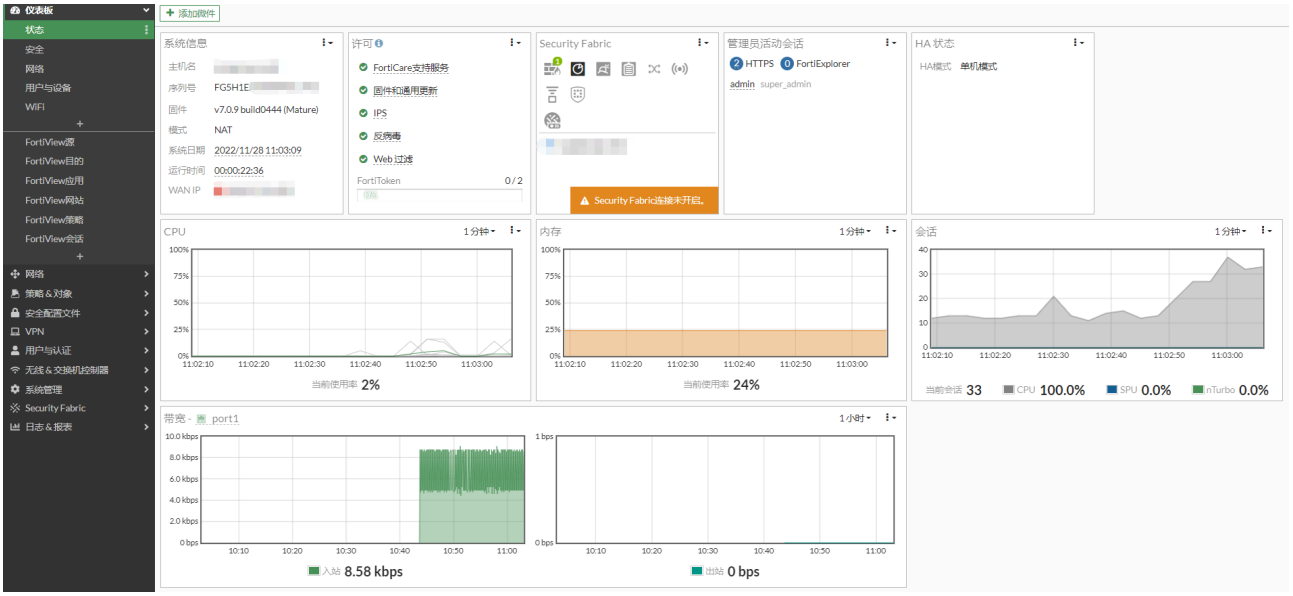
日志查看

查看 FortiGate 各日志模块是否有异常日志，如系统事件，HA 事件，入侵防御，反病毒等。



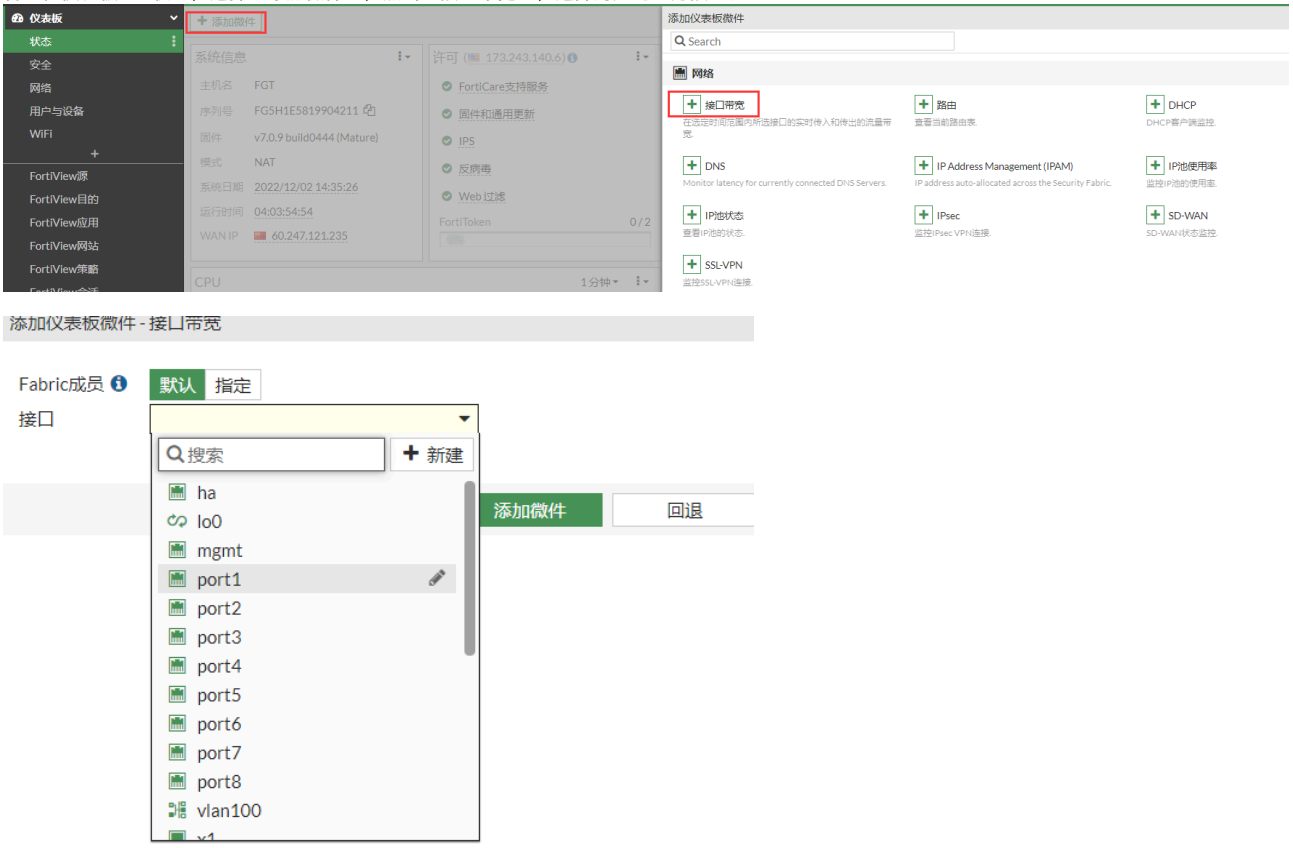
运维监控

1.在 "仪表板"-->"状态" 界面，通过添加 "微件" 的方式，如接口流量，如下图，查看 cpu、内存、接口流量和会话数。



2.如何查看设备接口的流量

答：在仪表板 --> 状态，选择“添加微件”，点击“接口带宽”，选择对应的业务接口。



3.如何通过 SNMP 管理并监控 FortiGate 的系统运行及性能情况。

答：a. 开启设备的 SNMP，配置步骤请参考系统管理 → SNMP → SNMP 的设置 ↗。

b. 在 SNMP 配置界面下方可直接下载 FortiGate SNMP MIB 文件，并将 MIB 文件导入进 MIB Browser、PRTG Network Monitor、Cacti 等 MIB 管理监控工具。

c. 通过 MIB 工具可读取出设备的相关 CPU 使用率、MEM 使用率、硬盘使用率、新建会话数、并发会话数等关键参数， 以下为整理出来的常用 MIB OID 参数值。

```
FortiGate设备名称          sysName          OID: 1.3.6.1.2.1.1.5
FortiGate设备序列号        fnSysSerial      OID: 1.3.6.1.4.1.12356.100.1.1.1
FortiGate设备软件版本      fgSysVersion     OID: 1.3.6.1.4.1.12356.101.4.1.1
FortiGate设备运行时间      sysUpTime        OID: 1.3.6.1.2.1.1.3
CPU 使用率                 fgSysCpuUsage    OID: 1.3.6.1.4.1.12356.101.4.1.3
MEM使用率                  fgSysMemUsage    OID: 1.3.6.1.4.1.12356.101.4.1.4
硬盘使用率                 fgSysDiskUsage   OID: 1.3.6.1.4.1.12356.101.4.1.6
IPv4并发会话数             fgSysSesCount    OID: 1.3.6.1.4.1.12356.101.4.1.8
IPv4新建会话数（1分钟）    fgSysSesRate1    OID: 1.3.6.1.4.1.12356.101.4.1.11
IPv6并发会话数             fgSysSes6Count   OID: 1.3.6.1.4.1.12356.101.4.1.15
IPv6新建会话数（1分钟）    fgSysSesRate1    OID: 1.3.6.1.4.1.12356.101.4.1.16
```

附件

- FortiGate 通用多运营商路由导入脚本
- 命令行脚本批量转换工具说明
- 更新日志
- 更新日志

FortiGate 通用多运营商路由导入脚本

运营商地址列表

电信、网通的路由表是不断在更新的，可以通过 https://ispip.clang.cn/查看运营商的地址列表

ISP IP列表 HTML格式 CIDR

文件	描述	数据	更新时间	大小
all_cn_cidr.html	中国大陆所有IP地址	5441条	2022-11-23 00:33:36	85.57 KB
chinatelecom_cidr.html	中国电信	1724条	2022-11-23 00:33:36	28.48 KB
unicom_cnc_cidr.html	中国联通/网通	655条	2022-11-23 00:33:36	12.77 KB
cmcc_cidr.html	中国移动	46条	2022-11-23 00:33:36	3.68 KB
crtc_cidr.html	中国铁通	17条	2022-11-23 00:33:36	3.23 KB
cernet_cidr.html	中国教育网	81条	2022-11-23 00:33:36	4.17 KB
gwbn_cidr.html	长城宽带/鹏博士	172条	2022-11-23 00:33:36	5.64 KB
othernet_cidr.html	中国其他ISP	3290条	2022-11-23 00:33:36	53.71 KB
hk_cidr.html	中国香港	2250条	2022-11-23 00:33:36	37.86 KB
mo_cidr.html	中国澳门	29条	2022-11-23 00:33:36	3.44 KB
tw_cidr.html	中国台湾	657条	2022-11-23 00:33:36	13.08 KB

ISP IP列表 TXT格式 CIDR

文件	描述	数据	更新时间	大小
all_cn_cidr.txt	中国大陆所有IP地址	5441条	2022-11-23 00:33:36	82.58 KB
chinatelecom_cidr.txt	中国电信	1724条	2022-11-23 00:33:36	25.49 KB
unicom_cnc_cidr.txt	中国联通/网通	655条	2022-11-23 00:33:36	9.77 KB
cmcc_cidr.txt	中国移动	46条	2022-11-23 00:33:36	710 B

如点击“unicom_cnc_cidr.html”，就可以看到联通的地址列表

ispip.clang.cn/unicom_cnc.html

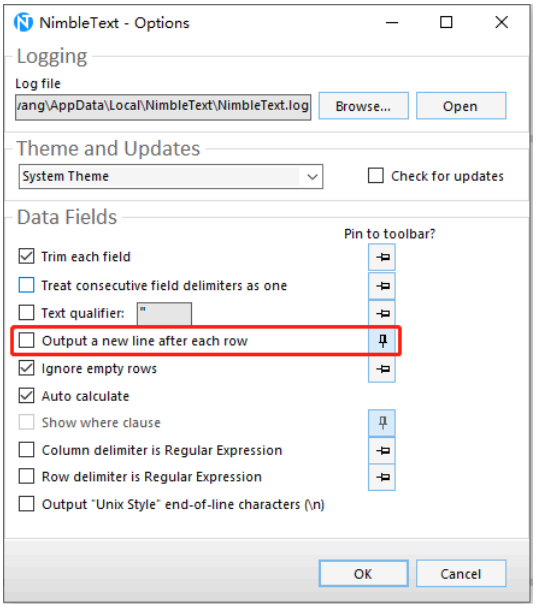
```
#################################################################
# ISP 中国联通(网通) 所有IP地址段表
# 数据来自于: https://ispip.clang.cn/
# 数据生成时间:2022-11-23 00:33:36
# 共计数据 890 条
# 脚本自动生成,如有错漏或任何建议,请联系admin[at]clang.cn
#################################################################
1.24.0.0/13
1.56.0.0/13
1.188.0.0/14
14.204.0.0/15
27.0.128.0/22
27.8.0.0/13
27.36.0.0/14
27.40.0.0/13
27.50.128.0/17
27.54.192.0/18
27.98.224.0/19
27.112.0.0/18
27.115.0.0/17
27.192.0.0/11
36.32.0.0/14
36.248.0.0/14
39.64.0.0/11
42.1.128.0/17
42.4.0.0/14
42.48.0.0/15
```

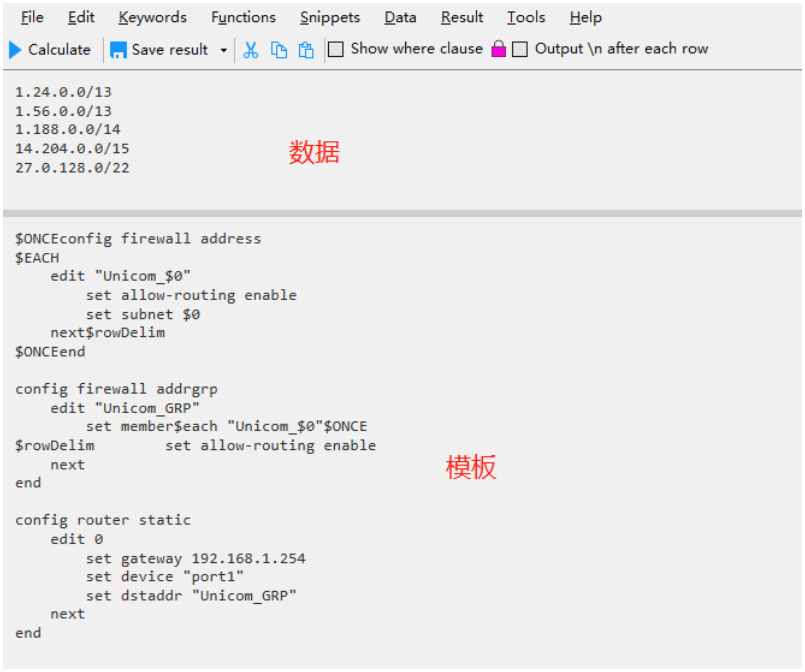
生成运营商路由脚本

根据运营商地址列表，在 FortiGate 中生成对应的运营商路由。可以 python，NimbleText 等工具生成，这里使用 NimbleText，其官网链接：http://nimbletext.com/

1. 通过 NimbleText 生成地址对象条目，这里截取联通地址列表的一部分做示范，其他的同理。

   在 NimbleText 选择“Tools”，点击“options...”，取消勾选“Output a new line after each row”





Calculate ☑ auto

```
config firewall address
    edit "Unicom_1.24.0.0/13"
        set allow-routing enable
        set subnet 1.24.0.0/13
    next
    edit "Unicom_1.56.0.0/13"
        set allow-routing enable
        set subnet 1.56.0.0/13
    next
    edit "Unicom_1.188.0.0/14"
        set allow-routing enable
        set subnet 1.188.0.0/14
    next
    edit "Unicom_14.204.0.0/15"
        set allow-routing enable
        set subnet 14.204.0.0/15
    next
    edit "Unicom_27.0.128.0/22"
        set allow-routing enable
        set subnet 27.0.128.0/22
    next
end

config firewall addrgrp
    edit "Unicom_GRP"
        set member "Unicom_1.24.0.0/13" "Unicom_1.56.0.0/13" "Unicom_1.188.0.0/14" "Unicom_14.204.0.0/15" "Unicom_27.0.128.0/22"
        set allow-routing enable
    next
end

config router static
    edit 0
        set gateway 192.168.1.254
        set device "port1"
        set dstaddr "Unicom_GRP"
    next
end
```

结果

模板中的脚本，其中 config router static 下的 gateway 和 device 需要根据现场的环境修改

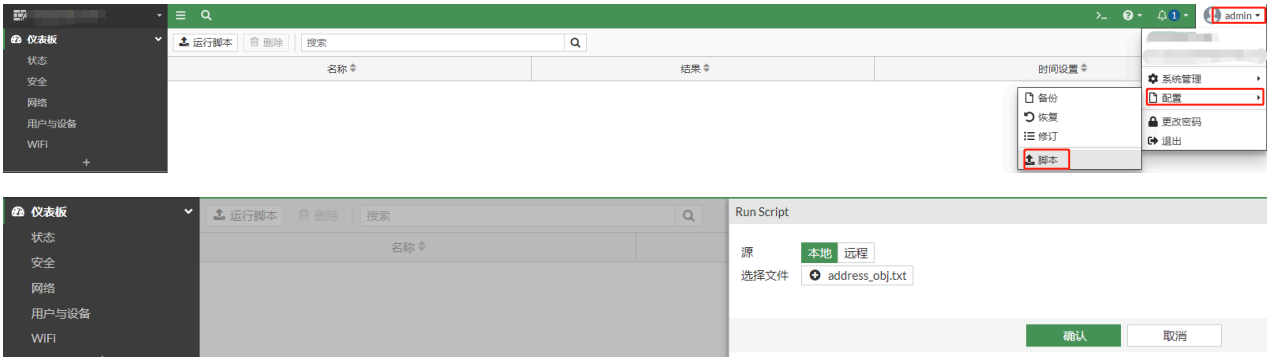
```
$ONCEconfig firewall address
$EACH
    edit "Unicom_$0"
        set allow-routing enable
        set subnet $0
    next$rowDelim
$ONCEend

config firewall addrgrp
    edit "Unicom_GRP"
        set member$each "Unicom_$0"$ONCE
$rowDelim       set allow-routing enable
    next
end

config router static
    edit 0
        set gateway 192.168.1.254       #改为现场环境使用的网关ip
        set device "port1"              #改为现场环境使用的接口
        set dstaddr "Unicom_GRP"
    next
end
```

2. 将生成的结果保存在 notepad++ 中，然后点击“编码”-->“使用 UTF-8 编码“并保存

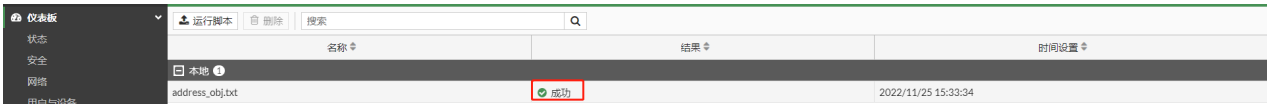


```
config firewall address
    edit "Unicom_1.24.0.0/13"
        set allow-routing enable
        set subnet 1.24.0.0/13
    next
    edit "Unicom_1.56.0.0/13"
        set allow-routing enable
        set subnet 1.56.0.0/13
    next
    edit "Unicom_1.188.0.0/14"
        set allow-routing enable
        set subnet 1.188.0.0/14
    next
    edit "Unicom_27.8.0.0/13"
        set allow-routing enable
        set subnet 27.8.0.0/13
    next
end

config firewall addrgrp
    edit "Unicom_GRP"
        set member "Unicom_1.24.0.0/13" "Unicom_1.56.0.0/13" "Unicom_1.188.0.0/14" "Unicom_27.8.0.0/13"
        set allow-routing enable
    next
end

config router static
    edit 0
        set gateway 192.168.1.254
        set device "port1"
        set dstaddr "Unicom_GRP"
    next
end
```

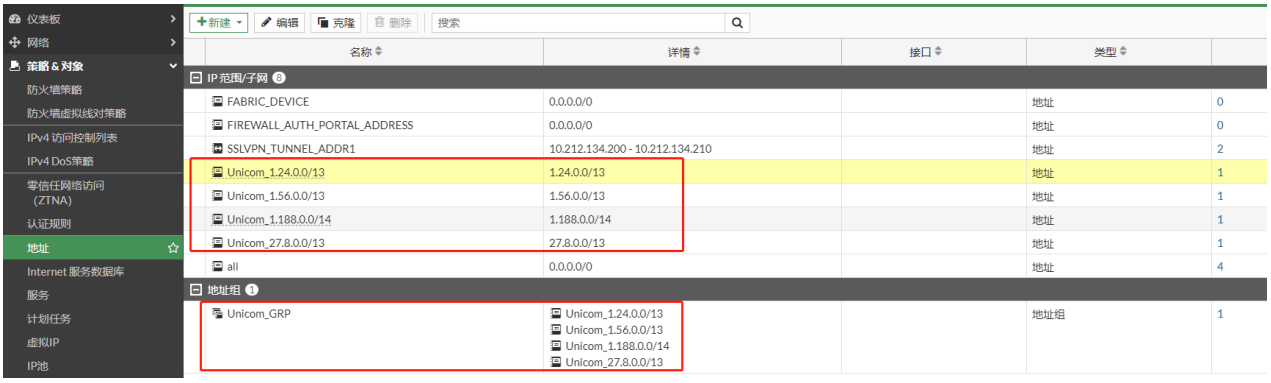
3. 点击 FortiGate 右上角账号旁边的下拉箭头，选择“配置”-->“脚本”，点击“运行脚本”

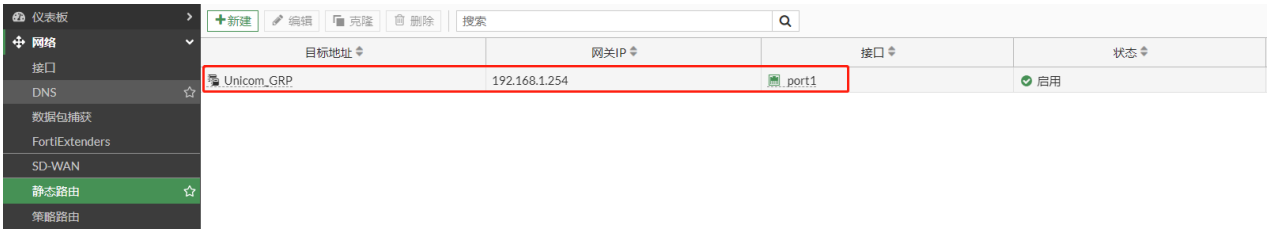


4. 脚本上传成功后会显示“成功”的字样



5. 在 FortiGate 地址对象界面查看刚刚上传的地址对象，地址对象组，路由




仪表板
网络
接口
DNS
数据包捕获
FortiExtenders
SD-WAN
静态路由
策略路由
新建
编辑
克隆
删除
搜索
目标地址
网关IP
接口
状态
Unicom_GRP
192.168.1.254
port1
启用

命令行脚本批量转换工具说明

说明

通常使用 NimbleText 工具生成相应的脚本，然后使用 Notepad++ 保存脚本。

NimbleText：用于批量生成命令行脚本。官网链接：http://nimbletext.com/

Notepad++：文本编辑器，保存配置脚本，保存成.txt 格式，编码为“UTF-8 编码”。官网链接：https://notepad-plus-plus.org/

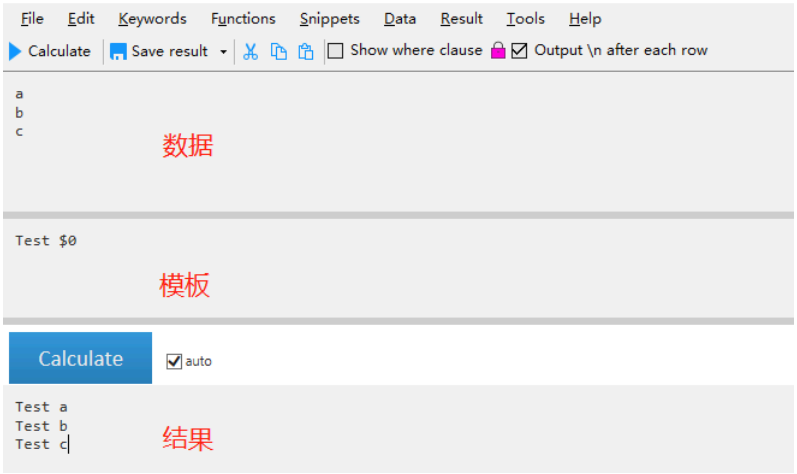
NimbleText 的脚本编辑举例

NimbleText 帮助文档：

https://nimbletext.com/Help/Keywords

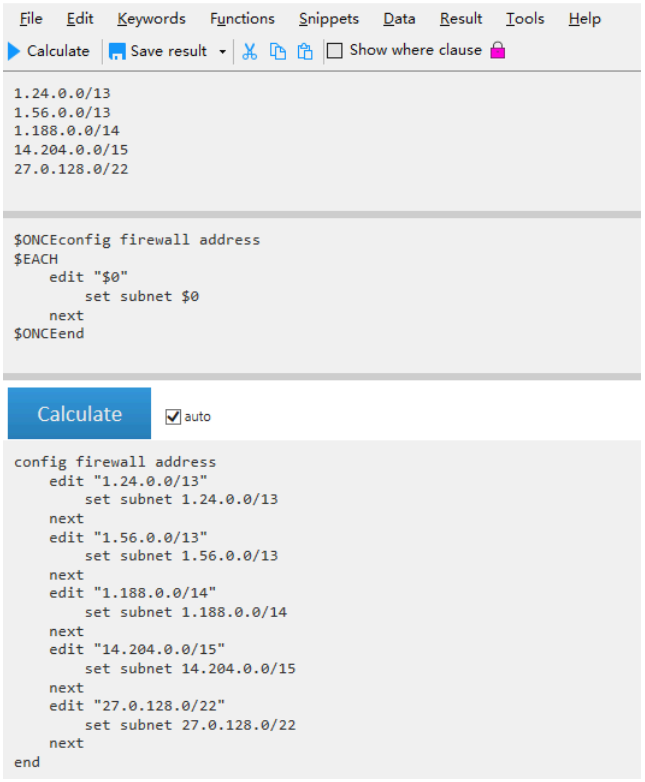
https://nimbletext.com/help

NimbleText 的界面分成三个部分：数据，模板，结果，NimbleText 就是将数据和模板结合生成对应的结果。



1. 批量生成地址对象

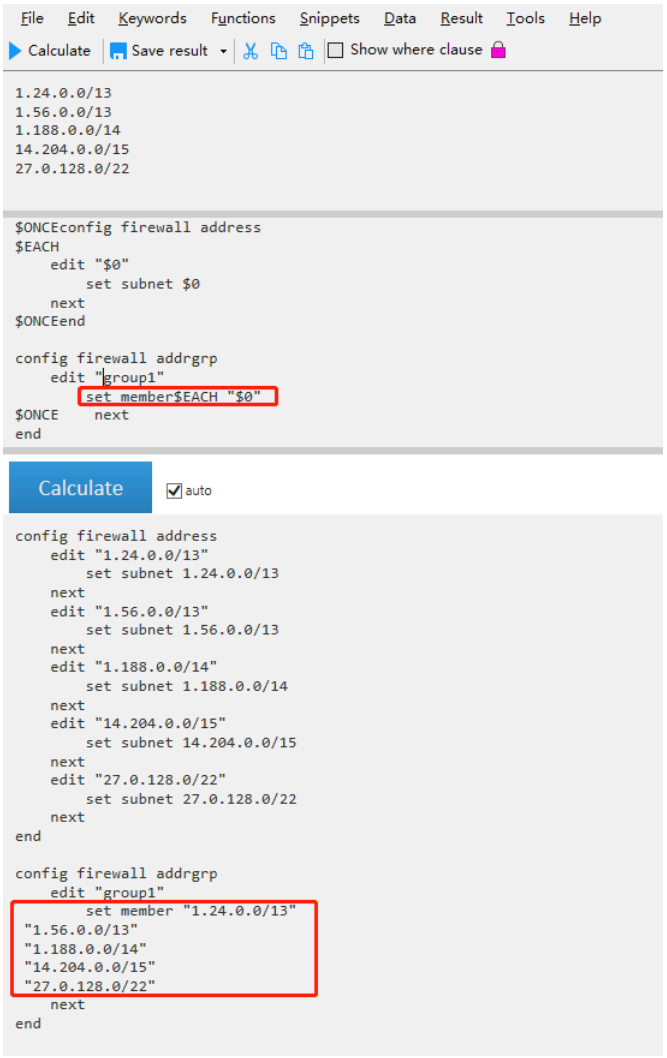
```
地址列表
1.24.0.0/13
1.56.0.0/13
1.188.0.0/14
14.204.0.0/15
27.0.128.0/22

FortiGate地址对象配置示例
config firewall address
    edit "1.1.1.0/24"
        set subnet 1.1.1.0/24
    next
end

根据地址对象命令分析，config firewall address和end不需要重复执行，对于edit和set subnet需要根据地址列表的变化而重复。
$ONCEconfig firewall address
$EACH
    edit "$0"
        set subnet $0
    next
$ONCEend

$ONCE，$EACH，$0都是NimbleText的关键字
$ONCE:任何在'$ONCE'关键字之后的文本将只输出一次，不管有多少行，到'$EACH'关键字或者到文件末尾为止。
$EACH:在'$EACH'关键字之后的任何文本将为每行输出一次（默认会换行），不管有多少行，到'$ONCE'关键字或者到文件末尾为止。
可以随意交错任意多的“$ONCE”和“$EACH”。
$0，$1:$0是第1列的数据，$1是第2列的数据，依次类推。
```



2. 批量生成地址对象，并将其加入地址对象组

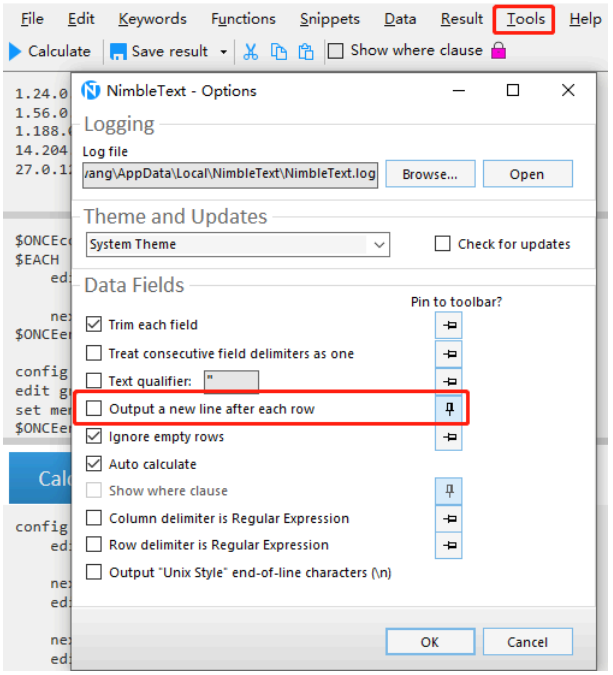
```
地址列表
1.24.0.0/13
1.56.0.0/13
1.188.0.0/14
14.204.0.0/15
27.0.128.0/22

FortiGate地址对象,地址对象组配置示例
config firewall address
    edit "1.1.1.0/24"
        set subnet 1.1.1.0/24
    next
    edit "1.1.2.0/24"
        set subnet 1.1.2.0/24
    next
end
config firewall addrgrp
    edit "group1"
        set member "1.1.1.0/24" "1.1.2.0/24"
    next
end
```

按照示例 1 的方式，NimbleText 输出如下：

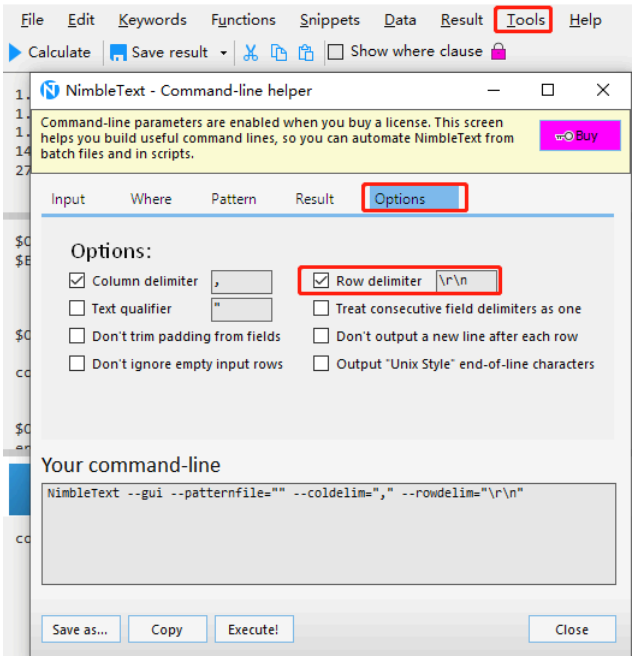


在地址对象配置中 edit "$0" 和 set subnet $0 循环第 1 列数据时，需要换行；在地址对象组 set member$EACH $0 循环第 1 列的数据时，不需要换行。因此需要解决 $EACH 关键自动换行的问题。

修改 $EACH 关键字的输出不自动换行，选择” Tools“-->“Options...”，取消勾选“Output a new line after each row”



$rowDelim：行分隔符，这里利用这个关键字来换行。选择“Tools”-->“Command-line helper”，点击“Options”可以看出行分隔符是 \r\n。



NimbleText 修改后模板脚本如下：

```
$ONCEconfig firewall address
$EACH
    edit "$0"
        set subnet $0
    next$rowDelim
$ONCEend

config firewall addrgrp
    edit "group1"
        set member$EACH "$0"
$ONCE$rowDelim    next
end
```

File Edit Keywords Functions Snippets Data Result Tools Help

Calculate | Save result | Show where clause

```
1.24.0.0/13
1.56.0.0/13
1.188.0.0/14
14.204.0.0/15
27.0.128.0/22
```

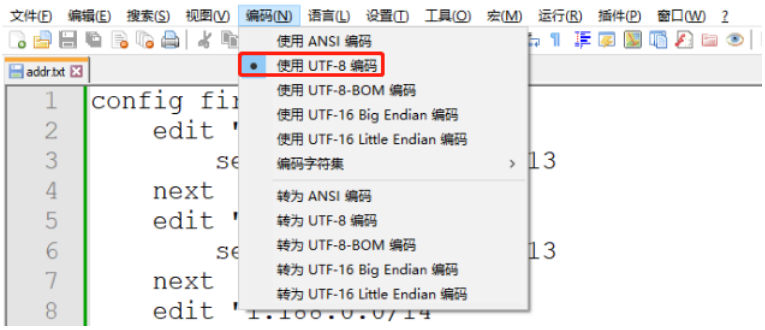
```
$ONCEconfig firewall address
$EACH
    edit "$0"
        set subnet $0
    next$rowDelim
$ONCEend

config firewall addrgrp
    edit "group1"
        set member$EACH "$0"
$ONCE$rowDelim    next
end
```

Calculate auto

```
config firewall address
    edit "1.24.0.0/13"
        set subnet 1.24.0.0/13
    next
    edit "1.56.0.0/13"
        set subnet 1.56.0.0/13
    next
    edit "1.188.0.0/14"
        set subnet 1.188.0.0/14
    next
    edit "14.204.0.0/15"
        set subnet 14.204.0.0/15
    next
    edit "27.0.128.0/22"
        set subnet 27.0.128.0/22
    next
end

config firewall addrgrp
    edit "group1"
        set member "1.24.0.0/13" "1.56.0.0/13" "1.188.0.0/14" "14.204.0.0/15" "27.0.128.0/22"
    next
end
```

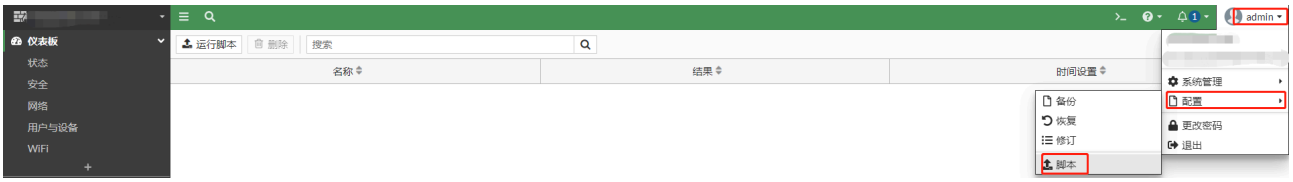
Notepad++ 保存脚本并导入

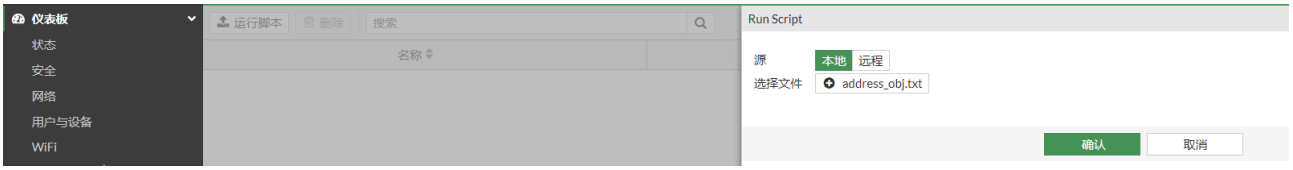
将 NimbleText 的输出保存在 notepadd++ 中，选择“编码”-->“使用 UTF-8 编码”，并保存。

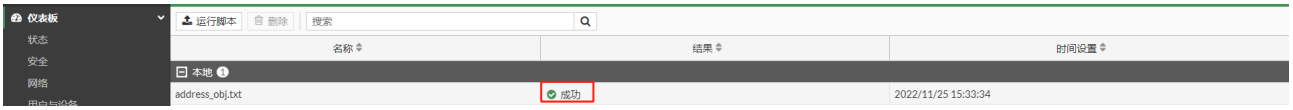


```
config firewall address
    edit "1.24.0.0/13"
        set subnet 1.24.0.0/13
    next
    edit "1.56.0.0/13"
        set subnet 1.56.0.0/13
    next
    edit "1.188.0.0/14"
        set subnet 1.188.0.0/14
    next
    edit "14.204.0.0/15"
        set subnet 14.204.0.0/15
    next
    edit "27.0.128.0/22"
        set subnet 27.0.128.0/22
    next
end

config firewall addrgrp
    edit "group1"
        set member "1.24.0.0/13" "1.56.0.0/13" "1.188.0.0/14" "14.204.0.0/15" "27.0.128.0/22"
    next
end
```

点击右上角下拉箭头，选择”配置“-->”脚本“，点击”运行脚本“，上传 address_obj.txt，脚本上传成功后会显示“成功”的字样。

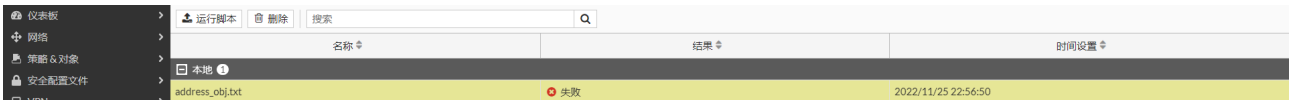






上传脚本失败排查

脚本上传显示“失败”的字样，即该脚本有问题。



在上传时，通过 ssh 或者 web console 登录 FortiGate 命令行，可以通过 debug 命令查看具体是哪一行有问题。

```
# diagnose debug  cli 8
# diagnose debug  enable
在diagnose debug cli 8的输出中，只有值为0才是成功的。

cmd=/tmp/monitor_upload_nRf481
0: config firewall address
0:     edit "1.24.0.0/13"
0:         set subnet 1.24.0.0/13
0:     next
0:     edit "1.56.0.0/13"
0:         set subnet 1.56.0.0/13
0:     next
0:     edit "1.188.0.0/14"
0:         set subnet 1.188.0.0/14
0:     next
```

```
0:     edit "14.204.0.0/15"
0:         set subnet 14.204.0.0/15
0:     next
0:     edit "27.0.128.0/22"
-160:         set subnet 27.0.128.0     #从这里可以看出这个地址没有掩码，修改后就能够正常上传了
1:     next
0: end
0:
0: config firewall addrgrp
0:     edit "group1"
-3:         set member "1.24.0.0/13" "1.56.0.0/13" "1.188.0.0/14" "14.204.0.0/15" "27.0.128.0/22"
1:     next
0: end
0:
```

更新日志

提示

更新内容标注 `author` 的表示该章节为该作者编写，标注 `from` 的表示该章节由其提出建议后由一本通作者编写。

2026

40th

Apr. 30, 2026

内容更新

- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 过期 License 强制升级”和“系统管理 → 固件与配置管理 → 固件版本管理 → 产品生命周期”中，同步补充了 CSB-260330-1 生命周期公告说明，用于标注 7.4/7.6 版本 EOES/EOS 的顺延信息。
- 在“路由 → 动态路由 → BGP”中增加了附加 AS-Path 章节。
- 在“VPN → IPSec VPN”中增加了重命名 IPSec 隧道章节。
- 在“系统管理 → FortiGuard 管理 → 服务指南”中增加了 FortiGate 强制注册章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理”中新增 FortiGate G 系列修改安全级别章节。
- 在“故障排查”中增加了 Ping 选项章节。
- 在“故障排查”中更新了常用排错命令章节。
- 在“用户与认证”中增加了本地用户密码策略章节。
- 更新 FortiClient 版本。
- 文本化部分 PDF 文档。
- 更新 PDF 版本到 40th。

39th

Mar. 26, 2026

内容更新

- 在“VPN → IPSec VPN → IPSec VPN 排错”中增加了 IPSec 使用附加 IP 时隧道无法建立章节。
- 更新“故障排查 → 会话工具 → 会话管理”章节，新增 `get system session status` 查看会话总数及 `get system session list` 简要列表的使用说明。
- 更新“故障排查 → 会话工具 → 会话参数分析”章节，补充 SCTP 协议状态表、NAT hook 信息说明、更多 state 标志的中文说明及 URL 分类 ID 对照表。
- 在“故障排查 → 会话工具”中增加了基于状态过滤会话章节。
- 在“VPN → IPSec VPN → IPSec VPN 排错”中新增 IPSec 相关诊断命令章节。
- 在“路由 → 组播”中新增将被拒绝的组播会话加入会话表章节。
- 在“VPN → IPSec VPN → IPSec VPN 排错”中新增多个拨号隧道使用相同 IP 池章节。
- 在“系统管理 → 管理员配置 → API 用户 → API 常见问题”中新增 API 认证失败章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理”中新增过期 License 升级限制章节。
- 在“网络管理 → 路由模式”中新增 ARP 表章节。
- 更新 PDF 版本到 39th。

界面更新

- 开启网站 blog 模式，支持分类与 tag 页面。

后台更新

- sidebar.ts 修改为自动填充模式。

38th

Feb. 13, 2026

内容更新

- 在“日志”中增加了自定义日志过滤器章节，并补充了 Log ID 说明部分。

- 在“路由 → 组播”中增加了组播转发 ↗章节。
- 在“Security Fabric → 自动化 → 常用触发器 → 计划时间触发器 ↗”中，增加了备份文件名中包含时间信息的配置。
- 在“系统管理 → FortiGuard 管理”中增加了 License 到期的影响 ↗章节。
- 在“系统管理 → 管理员配置”中增加了管理员登录锁定 ↗章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗”中，更新了推荐版本。
- 在“用户与认证 → 认证相关参数设置 ↗”中，增加了防火墙用户认证锁定章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理”中增加了过期 License 强制升级 ↗章节。
- 在“日志 → 日志常见问题”中增加了 FAZ 拒绝 FortiGate 连接 ↗章节。
- 在“VPN → IPSec VPN → IPSec VPN 排错”中增加了 IPsec 绑定 Loopback 时无法建立 ↗章节。
- 更新 PDF 版本到 38th。

37th Jan. 30, 2026

内容更新

- 在“系统管理 → 管理方式”中增加了 GUI/CLI 连接问题 ↗章节。
- 在“路由 → 路由常见问题”中增加了 TTL=1 的组播无法转发 ↗章节。
- 在“系统管理 → 管理方式”中增加了 CLI 输出模式 ↗章节。
- 在“网络管理”中增加了 EMAC VLAN ↗ 章节。
- 在“常见问题”中增加了 8008/8010/8015/8020 端口 ↗章节。
- 在“用户与认证 → 认证相关参数设置 ↗”中，增加了限制认证并发源 IP 数章节。
- 在“故障排查”中增加了 AV/IPS 内存高排查 ↗章节。
- 在“UTM → SSL/SSH 检查”中增加了检测模式 ↗、SSL/SSH 检测配置文件 ↗、certificate-inspection ↗、deep-inspection ↗、保护 SSL 服务器 ↗共 5 个章节。
- 在“HA 双机热备”中增加了 HA 虚拟 MAC ↗ 章节。
- 在“开局与日常维护 → 2GB 内存设备配置优化 ↗”中，增加了允许用于 AV 扫描的最大内存百分比设置。
- 在“常见问题 → 阻断/认证页面无法点击继续 ↗”中，增加了 Windows/macOS 系统下信任证书的具体步骤。
- 在“策略与对象 → 同一接口进出的三层转发 ↗”中，增加了 FortiOS 7.4.10/7.6.5/8.0.0 及之后版本的变化。
- 在“系统管理 → FortiGuard 管理 → 查看合同/更新状态 ↗”中，增加了 2 条查看更新相关状态的命令。
- 在“系统管理 → FortiGuard 管理”中增加了 FortiGuard 更新服务器位置 ↗章节。
- 在“VPN → IPSec VPN → 拨号 VPN”中增加了 IPSec 多用户组访问控制 ↗章节。
- 更新 PDF 版本到 37th。

后台更新

- 修复了 Umami 统计脚本在访问每个网页时都发送 2 次 API 请求的问题。
- 优化 images 文件夹资源利用，避免 images 文件夹重复上传到服务器。

2025

36th Dec. 29, 2025

内容更新

- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.4.5 版本，Android 系统的 FortiClient 更新到 7.4.4 版本（FortiClient VPN only 版本仍然保留 7.4.3）。
- 在“路由 → 动态路由 → BGP”中增加了 additional-path ↗ 章节。
- 在“策略与对象 → 本地流量策略”中增加了 Preferred Source ↗ 章节。
- 在“路由 → 动态路由 → BGP → BGP 常见问题”中增加了 RR 反射路由无法修改属性 ↗章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 软件签名增强 ↗”中，增加了系统文件校验异常时需要收集的信息。
- 在“网络管理 → 路由模式”中增加了 npu-neighbor-update ↗ 章节。
- 在“用户与认证 → LDAP 认证”中新增了 LDAPS CA 强制认证 ↗章节。
- 在“路由 → 动态路由 → BGP → BGP 常见问题”中增加了 BGP 选路忽略 AS-PATH ↗ 章节。

- 在“HA 双机热备 → HA 选取主设备的方法 ↗”中，增加了查看当前 HA 成员 uptime 间隔的方法。
- 在“系统管理 → SNMP → SNMP 的 HA 独立管理 ↗”中，在“监控 BGP/OSPF 的注意事项”部分增加了 FortiOS 7.2.11/7.4.8 及以后版本中的 SNMP 查询格式。
- 在“开局与日常维护 → 2GB 内存设备配置优化 ↗”中，增加了设备进入内存保护模式后需要收集的信息。
- 在“系统管理 → 管理方式 → 消除 Web 告警或提示”中增加了文件系统检查 ↗章节。
- 在“路由 → 动态路由 → BGP”中增加了路由反射器 ↗章节。
- 在“系统管理 → FortiGuard 管理”中增加了 FortiGuard 连接问题 ↗章节。
- 在“策略与对象 → ISDB”中增加了查询 ISDB/地理位置信息 ↗章节。
- 在“用户与认证 → FSSO 认证”中增加了 FSSO Agent SSL 配置 ↗章节（author 林宗勋）。
- 在“故障排查”中增加了 WAD 进程 Debug ↗ 章节（from 林宗勋）。
- 在“SAA-FortiLink → FortiLink 部署方案和举例”中增加了交换机零配置跨二层部署 ↗章节（author 夏苗青）。
- 更新 PDF 版本到 36th。

后台更新

- 隐藏所有页面 URL 中的 `.html` 后缀，访问带 `.html` 的链接会自动跳转到不带 `.html` 的链接；并修复因此产生的面包屑（vp-breadcrumb）路径显示问题。

35th Nov. 28, 2025

内容更新

- 在“VPN → IPSec VPN → IPSec VPN 排错”中增加了 Hub 反向流量异步 ↗章节。
- 在“策略与对象 → 策略常见问题”中增加了 VIP 匹配顺序 ↗章节（from 王爽）。
- 在“VPN → IPSec VPN”中增加了自动协商与 keepalive ↗ 章节（from 王爽）。
- 在“UTM → 病毒防护 → 基于文件 Hash 的 AV 豁免 ↗”中，增加了 FortiOS 7.4.X 不需要使用代理模式防火墙策略的说明（from 林宗勋）。
- 在“路由 → 动态路由 → BGP → BGP 常见问题 → BGP 与静态路由的优先顺序 ↗”中，更正了优先顺序的错误描述（from 于笙）。
- 在“系统管理 → FortiGuard 管理”中增加了禁用云通信 ↗章节。
- 在手册首页增加了快速开始 ↗章节。
- 在“网络管理”中增加了代理 ARP ↗ 章节。
- 在“系统管理 → SNMP”中增加了基于 VDOM 的 SNMP ↗ 章节。
- 在“日志”中增加了导出日志 ↗章节。
- 在“系统管理 → FortiGuard 管理”中增加了查看合同/更新状态 ↗章节。
- 在“HA 双机热备 → HA-Cluster 的网管 → HA 集群带外管理 → 独立管理接口 ↗”中，增加了在多 VDOM 环境下如何进入 vsys_hamgmt VDOM 的部分。
- 更新 PDF 版本到 35th。

界面更新

- 为每个页面增加阅读量显示（位于.page-info）。
- 修复了网站有更新后，浏览器依然使用旧缓存的问题（移除了 pwa 组件）。

后台更新

- 更新 umami 到 3.0 版本，对应更新相关 API 参数。
- 更换搜索模块为 Algolia（DocSearch 类型），在 Github 的 workflow 中增加了 MeiliSearch 自动化索引爬虫。
- 在搜索模块添加 AI 搜索（使用 gemini-2.5-flash-lite 模型）。
- 增加自动生成网站 PDF 的功能，并实现自动添加目录书签与封面。

34th Oct. 31, 2025

内容更新

- 在“策略与对象”中增加了 iprope 列表 ↗章节。
- 在“故障排查 → Sniffer 工具 → sniffer 工具介绍 ↗”中，增加了二层/三层组播/广播抓取条件。
- 在“云部署 → AWS → AWS 扩充硬盘容量 ↗”中，更正了“扩充硬盘动作实际不需要停机”的错误描述（from 林经纬）。

界面更新

- 使用 vuepress-theme-hope 项目重构网站，界面更加现代化，编译速度缩短到 1 分钟左右，访问性能大幅提升。
- 修复面包屑（vp-breadcrumb）路径显示为英文的问题。
- 修复了目录路径中目录乱序的问题。
- 为每个页面增加显示创建/编辑/预估阅读时间。
- 增加夜间模式对应的表格样式。
- 为夜间模式增加了新的主页动画。

后台更新

- 修复了访问统计 js 获取访问数据的 API 时执行多次的问题。

33rd

Sep. 30, 2025

内容更新

- 在“系统管理 → 固件与配置管理 → 固件版本管理”中增加了产品生命周期 ↗章节。
- 在“系统管理 → FortiGuard 管理 → 注册指南”中增加了 FortiCare 服务介绍 ↗章节（author 于笙）。
- 在“故障排查”中增加了数据包处理流程章节，包含 Parallel Path Processing ↗、无 NP 处理的数据包流程 ↗、被 NP 加速的流量 ↗、安全检测模式 ↗等 7 个章节。
- 在“部署前注意事项 → TLS 探测配置 ↗”中，增加了 TLS 探测失败默认动作章节。
- 在“策略与对象 → 策略常见问题”中增加了地址池 ARP 刷新 ↗章节（from 王祥）。

界面更新

- 页面右侧增加导航栏，删除原有的右上角导航按钮，删除每个页面页首的目录信息。
- 优化导航栏显示逻辑：①有 H2 时显示导航页面，无 H2 时隐藏导航页面；②导航栏会跟随页面位置实时显示当前标题位置。
- 手机浏览器适配优化：手机访问时隐藏页面导航、搜索、网站标题、顶部工具栏（除目录按钮）。
- 网站字体优化：优先使用客户端系统默认显示字体。

32nd

Aug. 31, 2025

内容更新

- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗”中，更新了推荐版本。
- 在“系统管理 → 管理方式 → 语言/时区切换”中加入了语言/时区切换 ↗章节。
- 在“系统管理 → 管理员配置 → 管理员设置 ↗”中，增加了管理员登录超时时间设置。
- 在“故障诊断 → 会话工具 → 会话管理”中增加了会话统计命令 ↗章节。
- 在“日志”中增加了日志显示地区/主机名 ↗章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件自动升级 ↗”中，更正了强制自动升级的条件，由原来的 EOS 到期时间修改为 EOES，并增加了针对 EOES 与 EOS 间的区别说明；去除了部分错误的例外情况。

界面更新

- 为图片增加了圆角样式。

31st

Jul. 31, 2025

内容更新

- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.2.11 版本，Android 系统的 FortiClient 更新到 7.4.3 版本。

后台更新

- 网站大部分 URL 替换为英文，以改善复制网页 URL 时如果包含中文，URL 过长的问题。
- 替换所有页面内的一本通跳转 URL 到对应英文格式。
- 替换网站构建程序，由 Gitbook 替换为 Honkit，提升了 30% 的编译速度。
- 替换网站的访问统计为自建的 umami 服务器，更换网站的 API 为自建服务器的 API URL。
- 网站页脚的统计信息对应更换为自建 umami 服务器获取（通过 umami-stats.js 读取数据，并对应修改 gitbook-plugin-tbfed-pagefooter 插件的 index.js）。

- 替换每页页眉处的“访问统计”链接到自建的 umami 服务器。
- 去除了页脚的二维码生成 js。

30th

Jun. 30, 2025

内容更新

- 在“开局与日常维护”中增加了 2GB 内存设备配置优化 ↗章节。
- 在“Security Fabric”中增加了威胁情报源 ↗章节。
- 在“系统管理 → 固件与配置管理 → 配置管理 → 备份配置文件到 Flash ↗”中，增加了 `revision-image-auto-backup` 和 `revision-backup-on-logout` 的开启建议。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，增加了免费版 FortiClient 不支持的 VPN 功能列表。
- 在“系统管理 → FortiGuard 管理 → 在线更新特征库 ↗”中，增加了特征库版本显示为 0.000 的原因。
- 在“策略与对象”中增加了会话 dirty 机制 ↗章节。
- 在“故障诊断 → 会话工具 → 会话管理 ↗”中，增加了 GUI 管理方式。

29th

May. 31, 2025

内容更新

- 在“故障排查”中增加了查找对象引用 ↗章节。
- 在“系统管理 → FortiGuard 管理 → 注册指南”中增加了 API 文档获取 ↗章节。
- 在“系统管理 → 管理员配置”中增加了 API 用户 ↗章节。
- 在“UTM 安全应用功能 → Web 过滤功能”中增加了 URL 过滤表达式 ↗章节。
- 重新编写了“VPN 技术 → FortiClient VPN → mac FortiClient 安装及使用 ↗”章节，添加了 macOS 权限要求部分。
- 更新了“VPN 技术 → FortiClient VPN → FortiClient 连接问题 ↗”章节，添加了若干百分比连接失败的原因和解决办法。
- 在“VPN 技术 → SSL VPN → SSL VPN 常见问题”中增加了 SSL VPN 连接后断开 ↗章节。

界面更新

- 加深了``（反引号）行内代码（inline code）的灰色（由 #f7f7f7 改为 #f0f0f0），并添加了圆角样式，此修改不会影响 code 样式。

28th

Apr. 30, 2025

内容更新

- 在“路由 → 动态路由 → BGP → BGP 常见问题”中增加了多个本地 AS 的配置和通告 ↗章节。
- 在“HA 双机热备”中增加了 HA 共享单一 License ↗ 章节（from 董书城、王爽）。
- 在“开局与日常维护”中增加了高频关注 ↗章节。
- 在“路由 → 动态路由 → BGP → BGP GR ↗”中，增加了关于 `graceful-end-on-timer` 命令的说明。
- 在“路由 → 动态路由 → BGP”中增加了 BGP 计时器 ↗章节。
- 在“HA 双机热备 → HA 常见问题”中增加了 HA 无法更新特征库/License ↗ 章节。

界面更新

- 移除了每页目录中的一级菜单（H1），包含每页首部的目录和右上角的悬浮目录按钮。

后台更新

- 网站性能提升：支持 HTTP2，使用 HTTP2 访问网站的延迟降低了大约 300ms。

27th

Mar. 31, 2025

内容更新

- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件自动升级 ↗”中，增加了查看自动升级状态章节。
- 在“网络管理 → DNS → DNS 配置 ↗”中，增加了对 FQDN 地址对象学习 IP 地址的 2 种方式说明。
- 在“部署前注意事项”中增加了云部署注意事项 ↗章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.2.10 版本。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件自动升级 ↗”中，增加了强制自动升级章节。

界面更新

- 增加了访问不存在的（或链接错误）资源时返回的错误页面 error.html。
- 浏览器首次访问网站时，页面会提示右上角的目录按钮。
- 右上角的目录按钮改为了圆角样式。

26th

Feb. 28, 2025

内容更新

- 在“VPN 技术 → FortiClient VPN → FortiClient 下载”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.2.9 版本。
- 在“VPN 技术 → 系统原生客户端接入 → Windows 系统 → L2TP VPN”中，增加了 PC 修改注册表的相关注意事项。
- 在“VPN 技术 → SSL VPN → 开启 SSL VPN”中，增加了 SSL VPN 在 7.6.3 版本下的支持情况。
- 在“VPN 技术 → IPSec VPN → 拨号 VPN”中增加了基于 TCP 的 FortiClient 拨号章节。
- 在“系统管理 → 固件版本管理”中增加了通过 FortiGuard 升级章节。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了 ESP 分片丢包章节。
- 在“策略与对象”中增加了导出策略为 CSV 或 JSON 章节，以及导出的 CSV 中文乱码的解决方法。
- 在“路由 → 动态路由 → BGP → BGP GR”和“路由 → 动态路由 → OSPF → OSPF GR”中，增加了 GR 与 BFD 不要同时使用的建议。
- 更新 PDF 版本到 26th。

25th

Jan. 31, 2025

内容更新

- 在“VPN 技术 → SSL VPN → 开启 SSL VPN”中，增加了 SSL VPN 在不同版本/型号组合下的支持情况，并删除了 GUI 开启方法。
- 在所有的 ADVPN 相关页面增加“ADVPN 目前不支持 Maximize Bandwidth (SLA) 模式”的提示信息。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 软件签名增强”中，增加了在 CLI 下查看安全级别时新的命名方式。
- 在“路由 → 动态路由 → 动态路由协议排错 → BGP Debug”中，增加了过滤邻居的命令。
- 在“故障排查 → Sniffer 工具 → sniffer 工具介绍”中，增加了 PPPoE 协商报文的抓取条件。
- 在“策略与对象”中增加了同一接口进出的三层转发章节。
- 在“网络管理 → SD-WAN → SD-WAN 规则 → Maximize Bandwidth → 基于轮询的 Maximize Bandwidth → 基于轮询的 Maximize Bandwidth (SLA)”中，删除了需要配置预估带宽的配置，并加以说明。
- 在所有包含 `snat-route-change` 配置的地方加入接口 up，路由存在，但健康检查 dead 的说明。

2024

24th

Dec. 31, 2024

内容更新

- 在“网络管理 → DNS → DNS 配置”中，增加了在 PPPoE 接口下关闭“DNS 覆盖”功能会导致 PPPoE 重启拨号的说明。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载”中，更新了推荐版本。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.2.8 版本。
- 在“部署前注意事项 → PPPoE 接口配置”中，增加了推荐使用 PPPoE 虚拟接口的理由以及配置步骤。
- 在“Security Fabric → 自动化 → 常用动作”中增加了自动抓包章节。
- 在“Security Fabric → 自动化 → 常用动作”中增加了 Teams Webhook、内存/CPU 达到阈值时收集信息、系统动作章节。
- 在“系统管理 → 系统设置”中增加了登录提示信息章节。
- 在“HA 双机热备 → HA 配置命令集”中，增加了 `route-ttl` 的说明及配置建议。

23rd

Nov. 30, 2024

内容更新

- 在“系统管理 → SNMP → FortiGate 的 SNMP 建议监控指标”中，删除了建议阈值与告警级别列。
- 在“网络管理 → DNS”中增加了穿越接口的 DNS 服务章节。

- 在“网络管理 → DNS”中增加了 DNS 配置 ↗章节。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了 Hub-Spoke 静态路由问题 ↗章节。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 接口模式 ↗”中，增加了使用 `local-gw` 自定义协商源 IP 的方法与注意事项。

界面更新

- 为目录展开/折叠添加了过渡动画效果（expandable-chapters.js）。
- 为 banner 增加动态粒子特效。
- 美化了右上角的搜索框样式（gitbook-plugin-theme-fexa）。
- 修复了部分表格错位问题。

22nd

Oct. 31, 2024

内容更新

- 在“部署前注意事项”中添加了公网接口访问限制 ↗章节。
- 在“网络管理 → SD-WAN”中增加了健康检查配置实践 ↗章节。
- 在“网络管理 → DNS”中增加了可选 DNS 服务器 ↗章节。
- 在“路由 → 动态路由 → BGP”中增加了 BGP 路由刷新 ↗章节。
- 在“策略与对象 → DoS 策略 → 什么是 DoS ↗”中，修正了防 DoS 攻击方法的介绍。
- 在“产品介绍 → 产品简介 ↗”中，增加了 FortiGate 功能规格、性能规格的链接。
- 在“故障排查 → Sniffer 工具”中增加了 Wireshark 显示抓包接口和方向 ↗章节。
- 将部署前注意事项相关章节整合为一个目录页面。

21st

Sep. 30, 2024

内容更新

- 在“部署前注意事项”中添加了 TLS 探测配置 ↗章节。
- 在“VPN 技术 → SSL_VPN”中增加了解锁 SSL VPN 用户 ↗章节。
- 在“部署前注意事项”中增加了开启 COMLog ↗ 章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.2.7 版本。
- 在“系统管理 → 固件与配置管理 → 配置管理”中增加了配置回退 ↗章节。
- 在“产品介绍 → 产品简介 ↗”中，补充了产品型号说明。
- 在“VPN 技术 → IPSec VPN”中增加了 IPSec DPD 工作模式 ↗章节。
- 在“系统管理 → 固件版本管理 → 固件下载 ↗”中，添加了新型号设备的推荐版本。

界面更新

- 美化页内目录的 CSS 样式。

20th

Aug. 31, 2024

内容更新

- 在“VPN 技术 → FortiClient VPN”中增加了系统兼容性列表 ↗章节（from 于笙）。
- 在“HA 双机热备 → HA-Cluster 不间断升级 ↗”中，增加了 HA 下降级的注意事项。
- 在“系统管理 → FortiGuard 管理”中增加了特征库误报/缺失上报 ↗章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理”中增加了固件自动升级 ↗章节。
- 在“系统管理 → FortiGuard 管理 → 注册指南”中增加了请求技术支持 ↗章节。
- 在“用户与认证 → 认证常见问题”中增加了升级版本后 Radius 连接失败 ↗章节。
- 在“网络管理 → 交换接口”中增加了交换接口的 MAC 地址表 ↗章节。

界面更新

- 在页面底部增加网站的访问统计信息。

后台更新

- 修改所有的文件下载链接至 handbook.fortinet.com.cn 域名下，以符合备案审查。

19th Jul. 31, 2024

内容更新

- 在“系统管理 → 固件版本管理 → 固件下载 ↗”中，添加了版本管理更详细的图片说明。
- 在“系统管理 → 固件版本管理”中增加了软件签名增强 ↗章节。
- 在“VPN 技术 → SSL VPN”中增加了开启 SSL VPN ↗ 章节。
- 在“系统管理 → 固件版本管理 → Flash 刷机 ↗”章节中，更换 Windows 下的 TFTP 服务器软件为 3CDeamon，并增加了 macOS 下的 TFTP 软件 Transfer。
- 在“系统管理 → FortiManager 管理 → FortiManger 版本推荐 ↗”、“系统管理 → 固件与配置管理 → 固件版本管理 → 升级路径 ↗”和“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗”中，增加了 FortiGate 版本与 FortiManager、FortiAnalyzer 产品版本的适应关系查询网站。
- 在“云部署 → 常见问题”中增加了 SSH 使用账号/密码登录 ↗章节。
- 在“系统管理 → 管理方式 → 消除 Web 告警或提示 → 关闭 Web 提示信息 ↗”中，增加了关闭 FortiConverter 提示信息的方法。
- 在“策略与对象 → 策略常见问题”中增加了 NAT 端口耗尽 ↗章节。

18th Jun. 30, 2024

内容更新

- 在“故障排查 → Sniffer 工具 → 通过 GUI 界面抓包 ↗”中，增加了 7.4 版本 GUI 抓包的特性说明。
- 在“系统管理 → 管理方式 → 消除 Web 告警或提示 → 消除 Web 证书告警 ↗”中，增加了通过 CSR 的方式配置 SAN 的格式。
- 在“常见问题 → 功能配置 ↗”中，增加了 License 无法验证或特征库无法更新的问题。
- 在“UTM 安全应用功能 → Web 过滤功能 → URL 过滤 ↗”中，增加了带目录的 URL 识别注意事项、URL 中携带 `#` 符号章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.0.13 版本。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗”和“VPN 技术 → FortiClient VPN → FortiClient 下载 ↗”中，替换下载页面为新版 support 网站页面。
- 在“部署前注意事项”中增加了 120G/121G HA 心跳口问题 ↗章节。

后台更新

- 增加访问统计脚本，可以点击每页页首的“访问统计”按钮查看网站的访问统计，删除了原有的百度统计 JS。

17th May. 17, 2024

内容更新

- 在“日志 → 自定义日志过滤器 ↗”中，增加了举例 3（基于日志 category 的多个 free-style ID）。
- 在“VPN 技术 → IPSec VPN”中增加了 IPSec 重协商机制 ↗章节。
- 在“路由 → 路由常见问题”中增加了黑洞路由不生效 ↗和策略路由不生效 ↗2 个章节。
- 在“系统管理 → FortiManager 管理”中增加了 FortiManger 版本推荐 ↗章节。
- 在“用户与认证 → 认证常见问题”中增加了 Web 认证弹出 2 个页面 ↗章节。
- 在“UTM 安全应用功能 → Web 过滤功能 → Web 过滤常见问题”中增加了 Chrome/Edge 访问网页无法被拦截 ↗章节。
- 在“策略与对象 → 虚拟 IP 映射 → 内网主机通过公网 IP 访问 VIP ↗”中，增加了一个新的场景，以及通过策略路由控制上网场景下的配置注意事项。
- 在“常见问题”中增加了获取接口对应的公网 IP ↗ 章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 连接问题 ↗”中，增加了 48% 断开连接的一个可能原因（from 王爽）。
- 在“用户与认证 → LDAP 认证”中增加了 LDAP 常见错误代码 ↗章节（from 于笙）。
- 更新 PDF 版本到 17th。

16th Apr. 16, 2024

内容更新

- 在“系统管理 → 固件与配置管理 → 固件版本管理 → Flash 刷机 ↗”中，使操作步骤匹配新版本 BIOS。
- 在“系统管理 → 固件与配置管理 → 配置管理”中增加了通过 SCP 备份配置文件 ↗章节。
- 在“系统管理 → 管理员配置”中增加了限制管理员 SSH/Telnet 跳板 ↗章节。

- 在“UTM 安全应用功能 → 病毒防护”中增加了基于文件 Hash 的 AV 豁免 ↗章节。
- 在“Security Fabric → 自动化 → 常用动作”中增加了 Email 动作 ↗章节。
- 在“故障排查 → Sniffer 工具 → Sniffer 过滤条件的合理使用 ↗”中，增加了抓取 PPPoE 报文的命令。
- 在“VPN 技术 → IPSec VPN → 拨号 VPN → 野蛮模式/network-id 适用场景 ↗”中，增加了 ADVPN 下的 network-id 或野蛮模式 ID 章节。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了 Hub-Spoke 双线路隧道抖动 ↗章节。
- 在“系统管理 → 系统设置 → 页面替换 ↗”中，增加了“替换信息组”章节。
- 在“系统管理 → 管理方式 → 消除 Web 告警或提示”中增加了关闭 Web 提示信息 ↗章节。
- 在“网络管理 → NetFlow → NetFlow 的配置 ↗”中，增加了 NetFlow 在多 VDOM、多 VDOM 环境 + HA 环境下的注意事项，并增加了 7.2.8 版本的 NetFlow 配置 CLI 变化说明。
- 在“网络管理 → SD-WAN → SD-WAN 配置举例 → ADVPN 场景 ↗”中，将 IPSec 协商模式修改为 IKEv2，并增加了 network-id 配置，以避免 Hub 双线路的 ADVPN 模式下出现 Shortcut 建立问题。
- 在“故障排查 → Sniffer 工具”中增加了 NP7 芯片抓包 ↗章节。
- 对“故障排查 → Session List 工具、Sniffer 工具、Debug Flow 工具”下的所有章节进行重新排版。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了查看预共享密钥明文 ↗章节。
- 更新 PDF 版本到 16th。

界面更新

- 为页面 banner 增加背景图片。

15th

Mar. 15, 2024

内容更新

- 在“VPN 技术 → SSL VPN”中增加了 SSL VPN 分配 IP 的方式 ↗章节。
- 在“路由 → 动态路由 → BGP → BGP 常见问题”中增加了关闭 Leftmost AS 检查 ↗的方法。
- 在“路由 → 动态路由 → BGP → BGP 基础配置 ↗”中，增加了 `network-import-check` 的作用说明。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了 IPSec 无法修改隧道类型 ↗的说明。
- 在“策略与对象 → 策略常见问题”中增加了 VIP 流量无法匹配防火墙策略 ↗章节。
- 在“策略与对象 → 策略常见问题”中增加了策略页面加载慢 ↗章节（from 刘琪琪）。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错 → 配置 IPSec 名称的注意事项 ↗”中，增加了 IPSec 隧道命名格式的注意事项。
- 在“网络管理 → SD-WAN → SD-WAN 配置举例”中增加了将已关联接口移至 SD-WAN 区域 ↗章节。
- 在“VPN 技术 → SSL VPN”中增加了 SSL VPN 源地址连接控制 ↗章节。
- 在“用户与认证 → 认证常见问题”中增加了认证与策略顺序的关系 ↗章节。
- 在“系统管理 → SNMP → FortiGate 的 SNMP 建议监控指标 ↗”中，增加了 FortiGate Licence 到期时间的相关 OID。
- 在“日志”中增加了 CLI 审计日志 ↗章节。
- 在“Security Fabric → 自动化”中新增了工作流 ↗章节。
- 在“Security Fabric → 自动化 → 常用触发器”中新增了系统事件日志触发器 ↗、UTM 日志触发器 ↗、证书过期触发器 ↗、计划时间触发器 ↗章节。
- 在“系统管理 → 管理方式”中增加了消除 Web 证书告警 ↗章节。
- 更新 PDF 版本到 15th。

界面更新

- 美化 code（代码块）的 CSS 样式。
- 美化 blockquote（引用）的 CSS 样式。
- 美化 pageTop（页面内顶部）类型的导航样式。
- 美化了 table（表格）的 CSS 样式。

14th

Feb. 8, 2024

内容更新

- 在“策略与对象 → 虚拟 IP 映射 → 内网主机通过公网 IP 访问 VIP ↗”中，增加了开启异步路由 + 策略路由下的特殊情况说明。
- 在“网络管理 → SD-WAN”中增加了 SD-WAN 排错 ↗章节。

- 在“故障排查 → 信息收集 ↗”中，增加了内存占用状态以及内存 Slab 状态的命令（包含输出结果的解释）。
- 在“路由 → 动态路由 → BGP → BGP 常见问题”中增加了使用 Loopback 接口建立 BGP 邻居 ↗和 BGP 与静态路由的优先顺序 ↗的说明。
- 在“故障排查 → 信息收集 ↗”中，增加了 Flash Disk 硬件检查的命令。
- 在“系统管理 → SNMP → SNMP 的设置 ↗”中，增加了团体名称配置的注意事项。
- 在“网络管理 → SD-WAN”中增加了 SD-WAN 常用 SNMP 监控项 ↗章节。
- 优化和更正了“网络管理 → SD-WAN → SD-WAN 介绍 → SD-WAN 健康检查 ↗”中的部分内容。
- 在“IPv6”中增加了 DS-Lite ↗ 章节。
- 在“路由 → 动态路由”中增加了路由对象 ↗部分，包含 Route Map、Access List、Prefix List、AS Path List、Community List 共 5 个章节。
- 在“策略与对象”中增加了转换 DNS 解析结果 ↗章节。
- 在“VPN 技术 → FortiClient VPN”中增加了 FortiClient 连接问题 ↗章节。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接 → 与友商对接 IPSec 的注意事项 → PKI 证书认证 ↗”中，增加了与友商建立 IPSec 时使用证书认证的注意事项（from 李威峰）。
- 在“常见问题”中增加了流量无法被 NP 加速 ↗章节（from 王涛）。
- 在“策略与对象”中增加了本地流量策略 ↗部分，包含 Local-in 策略和 Local-out 流量控制共 2 个章节。
- 在“UTM 安全应用功能 → 入侵防御”中增加了虚拟补丁 ↗章节。
- 在“UTM → DNS 过滤”中增加了透明 DNS 转发器 ↗章节。
- 更新 PDF 版本到 14th。

13th Jan. 19, 2024

内容更新

- 在“VPN 技术 → 系统原生客户端接入 → Windows 系统 → IKEv1 ↗”和“VPN 技术 → 其他 VPN 技术 → L2TP VPN ↗”中，增加了 L2TP 用户为 LDAP 用户时应使用 PAP 认证方法的注意事项。
- 在“路由 → 动态路由 → BGP”中增加了 BGP ↗ 相关章节，包含 BGP 基础配置、BGP GR、BGP 路由过滤、BGP 下一跳递归、BGP 多路径、BGP 常见问题共 6 个章节。
- 在“系统管理 → SNMP → SNMP 有 HA 独立管理的设置 → SNMP 的 HA 独立管理 ↗”中，增加了 BGP/OPSF 相关 OID 的注意事项。
- 在“SAA-FortiLink → FortiLink 堆叠组网说明和举例 → FortiLink MCLAG - 与第三方设备进行跨堆叠交换机聚合 ↗”中，增加了与 Cisco 交换机对接时需要增加的配置。
- 在“VPN 技术 → IPSec VPN → 拨号 VPN → 野蛮模式/network-id 适用场景 ↗”中，增加了 IKEv1 主模式、野蛮模式、IKEv2 拨号匹配条件的顺序，以及 IKEv2 network-id 相关内容。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错 → IPSec 协商失败举例 → IPSEC 协商失败举例 ↗”中，增加了 IKEv2 重协商失败可能的问题原因。
- 在“云部署 → 常见问题”中增加了更换 VM License 导致 CPU 无法添加 ↗的解决办法（from 王爽）。
- 在“路由 → 动态路由 → OSPF”中增加了 OSPF GR ↗ 章节。
- 在“故障排查”中增加了常用排错命令 ↗章节。
- 更新 PDF 版本到 13th。

界面更新

- 替换所有使用字符串表示的拓扑图为图片格式（共 17 个拓扑）。
- 替换所有拓扑图中较旧的设备元素（共 44 个拓扑）。
- 修改页面表格 CSS 样式宽度根据客户端屏幕宽度自适应。

后台更新

- 修复了部分用户主页左侧目录无法完整加载，导致页面加载时间过长的问题（将目录插件引用的 js 脚本改为本地获取）。

2023

12th Dec. 27, 2023

内容更新

- 重新编写了“策略与对象 → 虚拟 IP 映射 → 内网主机通过公网 IP 访问 VIP ↗”，增加了几种常见场景与配置方法。

- 在“策略与对象 → 虚拟 IP 映射 → 地址映射”、“策略与对象 → 虚拟 IP 映射 → 端口映射”和“策略与对象 → 多线路上网配置 → 双线路不同运营商上网配置”中，增加了 VIP、IPPool 的配置注意事项。
- 在“常见问题”中增加了流量到达 FortiGate 后无法转发章节。
- 在“网络管理 → SD-WAN → SD-WAN 配置举例”中增加了互联网出口场景、互联网出口和 VPN 混合场景、ADVPN 场景共 3 个章节。
- 在“HA 双机热备 → HA-Cluster 不间断升级”中，增加了 7.4.1 间断升级的配置变化。
- 在“故障排查”中增加了 IPSec 解密章节，包含 ESP 报文解密和 IKE 报文解密。
- 在“VPN 技术 → 系统原生客户端接入 → Android 系统 → Android12 或以上系统”中，修改了 IPSec 一阶段和二阶段算法配置，以适配 Android 14 系统。
- 在“策略与对象 → 虚拟 IP 映射”中增加了 VIP 映射的主机通过 SNAT 访问 Internet 场景章节。
- 在“UTM 安全应用功能 → Web 过滤功能 → URL 过滤”相关章节以及“UTM 安全应用功能 → Web 过滤功能 → Web 过滤简介”中，增加了 URL Filter 与 WebFilter 的执行顺序说明。
- 在“云部署”中增加了 Azure 相关章节，包括 Azure 部署 FortiGate 和结合 Azure 负载均衡部署 FortiGate HA（author 王祥）。
- 更新 PDF 版本到 12th。

界面更新

- 现在所有的代码块都可以根据客户端页面宽度自动换行了，更易于阅读。
- 美化每个页面表格的 CSS 样式。

11th

Nov. 30, 2023

内容更新

- 在“云部署 → AWS”中增加了 AWS FortiGate 虚机扩充硬盘容量章节。
- 在“用户与认证 → SAML 认证”中增加了防火墙用户与 SSL VPN SAML 认证（Windows ADFS）（author 白冰）和显式代理 SAML 认证（Windows ADFS）（author 王涛）两篇文章。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载”中，增加了目前大部分型号的推荐版本列表，并对 Mature/Feature 版本增加了更详细的说明。
- 修复了若干代码块的显示问题，修正了若干章节的内容错误。
- 在“常见问题”中增加了无法登录 FortiGate Cloud 的解决方法。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 原理”中增加了 IPSec DPD 的工作模式章节。
- 在“路由 → 动态路由 → HA 配置命令集”相关章节中，增加了 HA + 动态路由部署模式下配置 `route-ttl`、`graceful-restart` 的建议。
- 重新编写了“网络管理 → 透明模式 → 传统透明模式 → VLAN 与透明模式 Forward-Domain”章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载”中，将 macOS、Windows 系统的 FortiClient 离线下载包更新到 7.0.10 版本。
- 更新 PDF 版本到 11th。

10th

Oct. 25, 2023

内容更新

- 在“系统管理 → 管理员配置”中增加了通过 FortiCloud 账户登录 FortiGate 章节。
- 在“用户与认证 → FSSO 认证”中，增加了对于 FSSO 管理员账户权限说明的 KB 链接。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接 → 与思科 ASA 防火墙建立 IPSec VPN (IKE v1)”和“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接 → 与思科 ASA 防火墙建立 IPSec VPN (IKE v2)”中，增加了二阶段 SA rekey 条件的注意事项。
- 在“系统管理 → SNMP → FortiGate 的 SNMP 建议监控指标”中，增加了关于 VPN Tunnel 接口状态的 OID 与说明。
- 在“路由”中增加了 BFD 相关章节，包含 BFD 原理、BFD 基本配置、BFD 与静态路由、BFD 与 OSPF、BFD 与 BGP、BFD 多跳模式共 6 篇文档。
- 在“HA 双机热备 → HA-Cluster 的网管 → HA 集群带外管理 → 单独指定用于更新的 VDOM”中，增加了 VDOM DNS 服务器的配置说明。
- 在“VPN 技术 → SSL VPN”中增加了 DTLS 章节。
- 更新 PDF 版本到 10th。

9th

Sep. 19, 2023

内容更新

- 在“用户与认证 → FortiToken 认证”中增加了免费 FortiToken Mobile 在 FortiGate 绑定时遇到的问题及处理方法 ↗章节。
- 在“系统管理 → 管理员配置”中增加了远程认证管理员 ↗章节。
- 在“网络管理 → 交换接口 → VLAN/Hardware Switch ↗”中，补充了 VLAN Switch 与 Hardware Switch 的区别。
- 在“系统管理 → 固件与配置管理 → 配置管理”中新增了备份配置文件到 Flash ↗ 章节。
- 在“系统管理 → 管理员配置”中增加了远程认证与本地认证的优先级 ↗章节。
- 在“HA 双机热备 → HA-Cluster 不间断升级 ↗”中，增加了关闭不间断升级功能的注意事项。
- 在“系统管理 → 管理员配置 → 密码恢复 ↗”中，增加了 7.2.0 之后不再支持密码恢复功能的提示信息。
- 在“HA 双机热备 → HA-Cluster 不间断升级 ↗”中，将升级步骤更新到最新版本（7.0.12、7.2.5）。
- 在“IPv6 → NAT64”中增加了 NAT64 + DNS64 上网 ↗章节。
- 在“HA 双机热备 → HA-Cluster 的网管 → 独立管理 VDOM ↗”中，增加了不要在被 FortiManager 管理的 FortiGate HA 上开启独立管理 VDOM 的说明。
- 在“HA 双机热备 → HA-Cluster 不间断升级 ↗”中，增加了 HA 切换测试的注意事项。
- 在“HA-direct → 独立管理接口 ↗”相关章节中，增加了 HA-direct 开启后管理报文发送方式的说明。
- 在“策略与对象 → 配置 Session TTL ↗”中，完善了对不同协议的 Session TTL 说明。
- 优化部分目录结构。
- 替换部分拓扑图中的设备图标到新版设备图标。
- 更新 PDF 版本到 9th。

界面更新

- 修复在 Firefox 浏览器下高分辨率的图片无法正常缩放的问题。
- 所有页面中的图片可以点击放大到悬浮窗中。
- 在每个页面增加了该页面 URL 的二维码，通过扫描该二维码，可以直接访问并分享该页面。
- 目录上方加入一本通版本切换功能，可以快速切换至 6.0 一本通，删除原右上角页眉处的 6.0 一本通链接（已移除）。

后台更新

- 删除大量冗余图片文件。

8th

Aug. 31, 2023

内容更新

- 在“网络管理 → 透明模式 → 传统透明模式 → Bypass 部署 ↗”中，增加了可做 Bypass 的产品型号和参考链接。
- 在“UTM 安全应用功能 → 应用控制 → 应用控制配置 ↗”的注意事项中，增加了关于企业微信无法发送图片的解决方法。
- 在“用户与认证 → 认证相关参数 → 认证相关参数设置 ↗”中，增加了防火墙认证关闭源 MAC 检查的方法。
- 在“系统管理 → 固件与配置管理 → 固件版本管理 → 固件下载 ↗”中，使升级方法匹配较新版本的 Web 页面。
- 在“VPN 技术 → SSL VPN → SSL VPN WEB 代理模式配置 ↗”中，增加了 Web 代理模式自定义代理 IP 的方法。
- 在“故障排查”中增加了 Kernel CPU 高的定位方法 ↗章节。
- 在“网络管理”中增加了 NetFlow ↗ 章节，包含 NetFlow 的配置 ↗、NetFlow 模板 ↗、在 VPN 接口上配置 NetFlow ↗。
- 在“IPv6 → NAT46 映射 ↗”和“IPv6 → NAT64 → Embedded 模式的 NAT64 ↗”中，将 NAT46 和 NAT64 章节中的图片和步骤替换为中文。
- 在“VPN 技术 → IPSec VPN → IPSec VPN 排错”中增加了配置 IPSec 名称的注意事项 ↗章节，以说明 ADVPN 的 Shortcut 无法建立问题。
- 更新 PDF 版本到 8th version。

7th

Jul. 31, 2023

内容更新

- 在“用户与认证”中增加了 FSSO 认证 ↗章节。
- 在“故障排查”中增加了 FortiGate 之间测速 ↗章节。
- 在“系统管理 → 管理方式”中增加了语言/时区切换 ↗章节。

- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接”中增加了与友商对接 IPSec 的注意事项↗。
- 在“系统管理 → 管理员配置”中新增了管理员密码策略↗章节。
- 在“系统管理 → SNMP → SNMP 的设置↗”中，增加了 iReasoning MIB Browser 使用 `get bulk` 方式获取 OID 信息时显示不全的问题解决方法。
- 在“系统管理 → 固件与配置管理”中增加了备份不包含密码的配置文件↗章节。
- 在“系统管理 → FortiGuard 管理 → 注册指南 → FortiGuard 服务介绍↗”的注意事项中，增加了 FortiGuard 服务过期后会发生什么的说明。
- 在“系统管理 → FortiGuard 管理”中增加了离线导入硬件设备 License↗章节。
- 在“VPN 技术 → FortiClient VPN → FortiClient 下载↗”中，将各个系统的 FortiClient 离线下载包更新到 7.0.9 版本。

6th

Jun. 27, 2023

内容更新

- 在“VPN 技术 → FortiClient VPN → FortiClient 下载↗”中，增加了各个系统的 FortiClient 离线下载包。
- 在“VPN 技术 → FortiClient VPN → Windows FortiClient 安装及使用↗”中，增加了 CLI 安装方式（可实现静默安装等功能）。
- 在“SAA-FortiLink”中增加了 FortiLink 排障↗章节，包含“快速查找异常流量”。
- 重新编写了“用户与认证 → 认证超时时间 → 认证相关参数设置↗”章节，增加了防火墙用户认证超时时间、SSL VPN 超时时间的介绍。
- 在“常见问题”中增加了阻断/认证页面无法点击继续↗的处理方法。
- 在“网络管理 → 交换接口”中增加了 VLAN/Hardware Switch↗接口章节，并在软交换配置↗中增加了软交换接口和硬件交换接口的区别对比。
- 在“网络管理 → 交换接口 → 软交换配置↗”中，增加了关于软交换接口内部成员之间流量默认策略的说明。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 修改 IPSec 协商端口↗”中，增加了修改 IKE 端口后 VPN 业务层面的变化说明。
- 在“用户与认证 → 认证超时时间 → 认证相关参数设置↗”中，增加了关于 SSL VPN 超时时间的说明。

5th

May. 26, 2023

内容更新

- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接”中增加了与 Check Point 防火墙建立 IPSec VPN↗章节。
- 在“策略 & 对象”中增加了 ISDB↗章节。
- 在“系统管理 → 固件与配置管理 → 固件版本管理”中新增了固件下载↗、升级路径↗、Flash 刷机↗、FTP/TFTP/USB 升级↗、固件切换↗章节。
- 在“故障排查”中增加了进程监控与管理↗章节。
- 在“故障排查 → Sniffer 工具”中增加了通过 GUI 界面抓包↗章节。
- 在“故障排查 → Debug Flow 工具”中增加了通过 GUI 进行 Debug Flow↗章节。
- 在“系统管理 → 系统设置”中增加了页面替换↗章节。
- 在“HA 双机热备”中更换 HA 不间断升级的流程图，更换 FGCP 的拓扑图。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → GRE over IPSec”中增加了 GRE over IPSec↗配置方法说明信息。
- 在“VPN 技术 → SSL VPN → SSL VPN FortiClient 隧道模式↗”中，增加了对“基于策略目标启用”使用限制的说明（from 肖芹涛）。
- 在“网络管理 → 透明模式 → 传统透明模式↗”和“网络管理 → 透明模式 → 虚拟接口对（VWP）↗”中，修正了关于 FortiGate 透明模式和虚拟接口对模式下对广播、组播默认动作的说明（from 肖芹涛）。
- 在“VDOM → VDOM 基本配置”中增加了无法删除空 VDOM 时的处理方法↗（from 李威峰）。
- 在“用户与认证 → FortiToken 认证 → FortiToken Mobile APP 软件下载与安装↗”下，增加了 FortiToken Android 的离线下载包。
- 在“SAA-FortiLink → FortiLink 部署方案说明和举例”中增加了 FortiLink 交换机零配置跨三层部署↗章节（author 夏苗青）。
- 更新 PDF 版本到 5th version。

4th

Apr. 13, 2023

内容更新

- 在“系统管理 → SNMP”中增加了 SNMP 与管理员可信主机↗章节。
- 在“IPv6”中增加了 DHCPv6↗章节，包含有状态、无状态 + 有状态和 DHCPv6-PD。

- 在“日志”中增加了自定义日志过滤器 ↗章节。
- 在“VPN 技术 → IPSec VPN → 点到点 VPN → 与友商进行 IPSec VPN 对接”中增加了与华为防火墙建立 IPSec VPN ↗、与 H3C 防火墙建立 IPSec VPN ↗、与 Juniper 防火墙建立 IPSec VPN ↗、与山石防火墙建立 IPSec VPN ↗、与 Palo Alto 防火墙建立 IPSec VPN ↗ 章节。
- 在“故障排查”中增加了 FortiGate Support Tool ↗ 的使用章节。
- 在“故障排查”中增加了信息收集 ↗章节。
- 重新编写了“系统管理 → 管理方式 → Console 管理 ↗”章节。
- 重新编写了“系统管理 → 管理方式 → SSH/Telnet 管理 ↗”章节。
- 在“系统管理 → 固件与配置管理 → 配置备份及恢复 ↗”中，增加了关于导出配置文件时权限问题的说明。
- 在 ippool 相关配置中增加配置绑定接口的建议，在 SSL VPN 中增加优先使用隧道模式的建议。
- 增加 FAC 一本通的链接（author 夏苗青）。
- 增加了云部署 ↗章节，包含 AWS 部署 FortiGate ↗ 和阿里云部署 FortiGate ↗（author 王祥）。
- 在“日志”中增加了硬盘与 Flash 闪存 ↗章节，介绍硬盘/Flash 相关操作（author 王振华）。
- 在“策略与对象 → 配置 Session TTL ↗”中，增加了全局自定义 Session TTL 的用法（from Nick）。
- 在“VPN 技术 → IPSec VPN → 拨号 VPN → HUB and SPOKE 模式 → Hub-and-spoke 模式 ↗”中，优化了关于 HUB 使用主模式的说明（from 林春龙）。
- 在“常见问题 → 功能配置”中增加交换接口成员改为三层接口的说明（from 肖芹涛）。
- 更新 PDF 版本到 4th version。

界面更新

- 每页首部增加 PDF 下载按钮。
- PDF 下载按钮移至首页。
- 修复部分注意事项（引用格式）字体为灰色的问题。

后台更新

- 网站发布更改为 HTTPS 协议。
- 为网站增加 sitemap，用于 SEO（目前只有 Google 的 sitemap）。
- 为网站增加状态统计插件，可以进入“附件 → 一本通服务器状态”页面查看。

3rd Mar. 17, 2023

内容更新

- 在“VPN 技术”章节中增加了系统原生客户端接入 ↗章节，包含 iOS、macOS、Android、Windows 系统（IKEv1/IKEv2）。
- 增加了 SAA-FortiLink ↗ 章节。
- 在“开局与日常维护”中增加了常用维护命令 ↗章节。
- 在“系统管理 → 设备管理 → FortiGuard 管理”中增加了 PRMA 高级服务注册步骤 ↗、产品服务续费 ↗、RMA 服务迁移 ↗章节。
- 在“HA 双机热备 → HA 选取主备设备的方法 → HA 选取主设备的方法 ↗”中，更换了更直观的逻辑图，并明确了 HA uptime 的说明。
- 在“故障排查”中增加了终端日志记录 ↗章节（from 王爽）。
- 修正了关于 PPPoE 接口的建议说明（from Nick）。
- 更新了 SD-WAN 转发逻辑的说明。
- 将注册指南相关章节整合为一个目录，优化了系统管理章节的目录结构，在页面导航栏增加了注册指南链接与 6.0 一本通的链接。
- 更新 PDF 版本到 3rd version。

界面更新

- 修复了部分表格影响章节编号的问题（将表格更改为 HTML 格式）。
- 删除了 theme-fexa 主题右侧的导航窗格（通过修改 CSS 与 HTML 文件），增加 anchor-navigation-ex-navbar 插件（修改插件显示高度为 100px），取代 theme-fexa 原有的导航窗格，同时修改 theme-fexa 主题的列宽为 Gitbook 默认 CSS 列宽，这样手机端的浏览体验更好了。
- 所有 PDF 文件全部页面内嵌显示，不需要跳转到外部查看。

后台更新

- 增加了百度统计的 js 脚本（可以多方面统计每日一本通网站的访问情况）。
- 增加了每页查看次数的统计脚本，通过关联 MemFire Cloud 数据库表实现（每页最上方显示）。

2nd

Feb. 16, 2023

内容更新

- 在“VPN → FortiClient VPN”中增加了 iPhone FortiClient 安装及使用 ↗和 Android FortiClient 安装及使用 ↗章节。
- 在“VPN → IPSec VPN → 点对点 VPN → 和友商进行 IPSec VPN 对接”中增加了与阿里云建立 IPSec VPN ↗、与 OpenSwan 建立 IPSec VPN ↗、与 StrongSwan 建立 IPSec VPN ↗ 章节。
- 在“系统管理 → 系统设置”中增加了 Email 服务 ↗和证书管理 ↗章节。
- 增加了用户与认证 ↗章节（包含本地、LDAP、Radius 认证）。
- 更新 SD-WAN 转发流程。
- 增加了部署前注意事项 ↗章节。
- 在“策略与对象 → 虚拟 IP 映射”中增加了 NAT64 ↗ 和 NAT46 映射 ↗章节。
- 在“HA 双机热备 → HA-Cluster 的网管”中增加了单独指定用于更新的 VDOM ↗ 章节。
- 优化目录结构。

界面更新

- 在每个页面的右上角添加了页内导航。
- 更新页面主题。

1st

Jan. 9, 2023

内容更新

- 基于 FortiOS 7.0 版本重新编写。
- 增加部分 7.0 新增功能，删除部分老旧内容。

界面更新

- 基于 Markdown 语言重新排版编写，并通过 GitBook 发布。
- 增加了全局搜索功能。
- 现在可以单独复制每个页面的 URL 了。
- 可以在每个页面的左上角调整页面字体大小、页面风格。
- 为每个子页面添加了目录树。