智能 IP 摄像头 C SDK (Amazon S3)

(IPC-H264-HLS-C-SDK)

部署手册

李挚

版本：v1.0.0

最后更新时间: 2021 年 08 月

亚马逊云科技

目　录

背景

本部署指南包含两个主要部分，第一部分详细描述如何使用，编译 IPC-H264-HLS-C-SDK，并提供示例伪代码，使得 IPC 可以使用内置 SoC 计算能力，完成视频分片、打包，并上传到 S3 存储。

第二部分详细描述如何搭建与 IPC-H264-HLS-C-SDK 配合使用的云端视频回放架构。

解决方案描述

本解决方案通过直接调用 Amazon S3 API，利用 Signature V4 签名算法，实现 IPC 鉴权，并完成视频分片对象上传。得益于无需在云端或边缘侧部署收流资源，因此可以将视频云存成本减少到最低。

以 3 秒视频分片，1080P 分辨率 5Mbps 码率，视频保存 7 天为例，每月视频云存+S3 API 调用成本按照 2021 年 5 月价格计算约为 55.5 元/月。其中 API 调用成本约为 3.5 元/月（计算公式见下），7*24 存储成本约为 52 元/月（计算公式见下）。

```
1200 请求/小时×24 小时/天×30 天/月×0.00000405 元/请求 = 3.5 元/月

0.5MB/秒×3600 秒/小时×24 小时/天×7 天×0.1755GB/月÷30 天/月÷1024MB/GB=52 元/月
```

推送到云端的视频分片以 Transport Stream（简称 TS）格式存储。

值得注意的是，IP 摄像头厂商可以通过调节帧率（FPS），码率（bitrate），编码算法等，实现对音、视频码流进行调整，从而使得实际成本低于以上计算成本，从而实现成本优化。

在回看时，通过 API Gateway 调用 Lambda 根据指定时间段防伪，检索视频分片并且动态添加到 m3u8 播放列表中，即可实现 HLS 回看功能。支持指定时间段回看、近实时（延迟 6-20 秒取决于网络情况）回看等场景。

使用场景

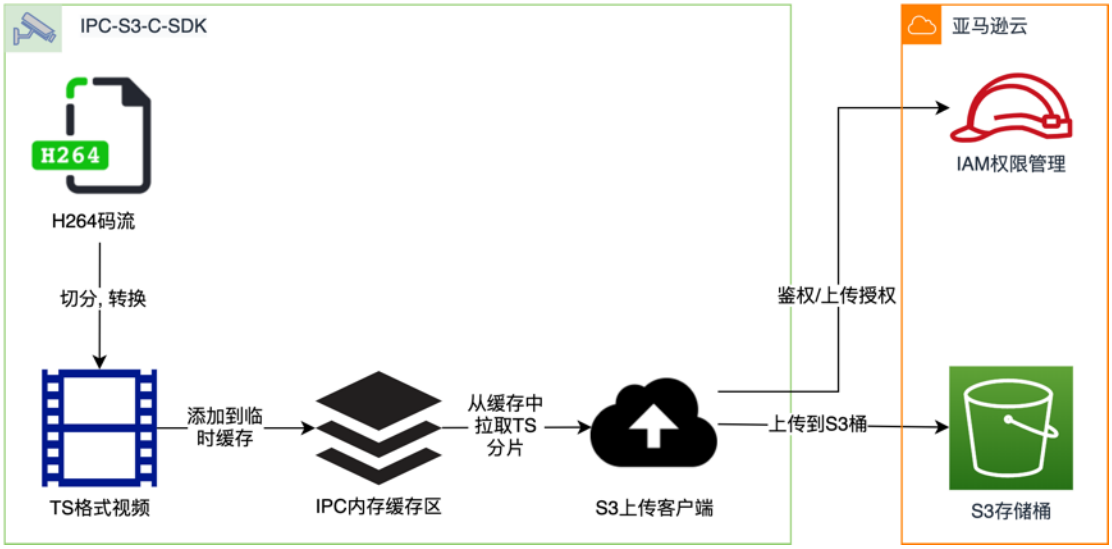
本解决方案适用于安防监控视频云存，智能门铃，居家监控等场景。

系统架构

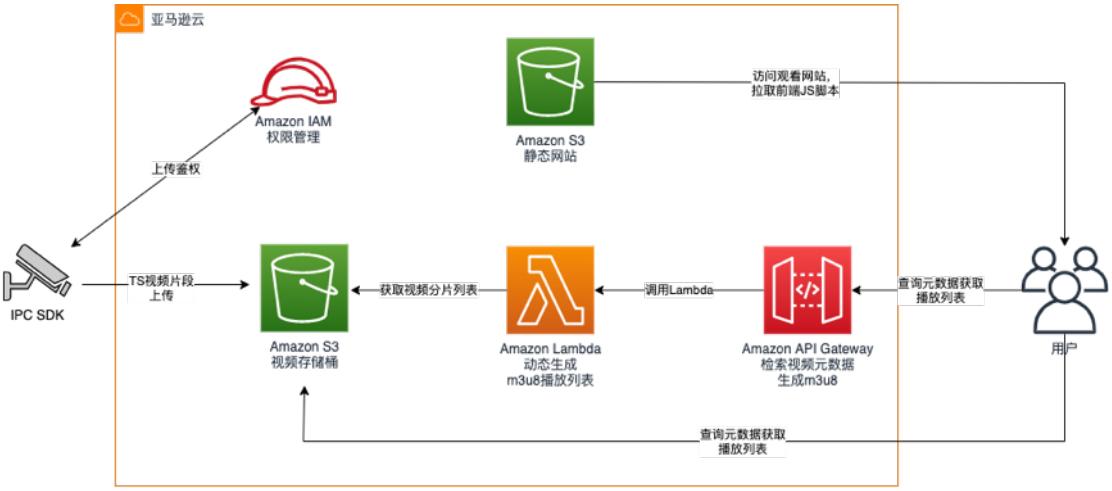
架构图

本解决方案包含设备端 SDK 和回放参考系统架构。

其中设备端 SDK 运行于 IPC 内部，用于完成视频切片和上传功能。



回放参考系统架构如下图所示：



组件

IPC-H264-HLS-C-SDK

- 支持编译到设备固件中
- 利用 SoC 计算能力和内存，完成视频切片、封包、缓存
- 将缓存的视频上传到 S3 存储桶

Amazon IAM

- 管理 IPC-H264-HLS-C-SDK 上传权限，实现鉴权、授权功能

S3 视频存储桶

- 云原生托管服务，天然高可用，且具备强大的扩展潜力
- S3 视频存储桶用于存储视频文件
- 视频存储格式为：<前缀>/<年>/<月>/<日>/<时>/<分>/<秒>.ts
- 前缀在设备端指定，格式可以任意定义，建议在与 IoT 设备管理一同使用时，以设备证书 ID 为起始
- 与 IoT 设备管理一同使用时，支持通过前缀控制 IPC 上传路径和权限，从而提升系统安全性

S3 静态网站桶

- 云原生托管服务，天然高可用，且具备强大的扩展潜力
- 对公网可见，与 CloudFront 配合，搭建静态，基于 Video.js 的 HLS 视频会看网站
- 独立于视频存储桶，从而确保视频内容更加安全
- 利用 S3 强大的性能和 CloudFront 的内容分发平台支持海量用户同时访问

API Gateway 生成 M3U8 播放列表

- 云原生托管服务，天然高可用，且具备强大的扩展潜力
- 无需设备侧生成或上传 M3U8，设备侧仅上传视频分片，从而节省成本
- 回放时，根据回放条件，动态检索视频分片，并动态生成 M3U8 播放列表，实现会看成本与回看时长正相关的成本模型
- 无服务器架构，无用户访问时无基础运行成本，降低生产环境上线门槛与业务系统前期运行成本

Lambda 提供 M3U8 生成逻辑

- 云原生托管服务，天然高可用，且具备强大的扩展潜力
- 实现自动检索 S3 查找视频分片
- 实现动态生成播放列表内容

部署说明

部署前提

- IPC 需要获取 AK/SK/Token 信息用于访问亚马逊云服务。建议通过 IoT 设备管理添加设备，配置证书，并通过 IoT Credential Provider 生成 AK/SK/Token。

- IPC 系统时间与标准时间只差不能大于 5 分钟。建议在固件内提供 NTP 服务实现时间同步。
- 若在中国区部署，需准备 ICP 备案过的域名，用于申请 Amazon Certificate Manager 证书。
- 如用戶直接部署在海外区域，则无需 ICP 备案过的域名，API Gateway 输出的网址可以直接访问，如用戶无需自定义域名，可以跳过对应步骤。

已测试 IPC SoC

- Hisi 3518EV300
- Ingenic T31

IPC-H264-HLS-C-SDK 交叉编译步骤

同步代码到本地

```
git clone https://github.com/aws-samples/ipc-h264-hls-c-sdk.git
```

同步第三方依赖库

进入刚刚同步完成的代码目录

```
cd ipc-h264-hls-c-sdk/
```

初始化并同步第三方依赖库

```
git submodule init

git submodule update
```

配置 OpenSSL 依赖库

```
cd 3rd/openssl/
```

```
./Configure no-asm no-async no-egd --prefix=$PWD –-cross-
compile-prefix=<YOUR_CROSS_COMPILER> <YOUR_PLATFORM>
```

注意：

需要替换以上命令中<YOUR_CROSS_COMPILER>为交叉编译工具链前缀，例如：mips-linux-gnu-

需要替换以上命令中<YOUR_TARGET_PLATFORM>为目标平台。例如：linux-x86_64。

编译 OpenSSL 依赖库

```
make
```

检查目录中已经输出 libcrypto.a 和 libssl.a 文件。

配置 Curl 依赖库

```
cd ../curl
autoreconf -fi
export LD_LIBRARY_PATH=../openssl
export CROSS_COMPILE=
export LDFLAGS="-L$PWD/../openssl/"
./configure ac_cv_func_RAND_egd=no --disable-shared --enable-static --with-
ssl=$PWD/../openssl
```

编译 Curl 依赖库

```
make
```

检查确保 libcurl.a 文件已经生成到 lib/.libs/ 目录中。

编译 IPC-H264-HLS-C-SDK

回到 IPC-H264-HLS-C-SDK 目录

```
cd ../..
```

编辑 Makefile 文件，添加 CROSS_COMPILE 前缀

```
CROSS_COMPILE=<YOUR_CROSS_COMPILER>

CC=$(CROSS_COMPILE)gcc

LD=$(CROSS_COMPILE)ld

AR=$(CROSS_COMPILE)ar
```

编译 IPC-H264-HLS-C-SDK

```
make
```

检查目录中已经输出 s3_hls.a 文件。在编译固件时需要引用 SDK 中的 .h 头文件，并链接 s3_hls.a 文件。

引用 IPC-H264-HLS-C-SDK 伪代码

初始化 SDK

```
S3_Put_Initialize(BUFFER_SIZE, region_name, bucket_name, prefix,

custom_endpoint_name);


#define BUFFER_SIZE   4*1024*1024 // 4MB

if(S3_HLS_OK != S3_HLS_SDK_Initialize(BUFFER_SIZE, region_name, bucket_name,

prefix, custom_endpoint_name) ) {

    return FAILED;

}
```

配置访问 S3 存储桶使用的身份 信息

```
if(S3_HLS_OK != S3_HLS_SDK_Set_Credential(access_key, secret_key, token)) {
    S3_HLS_SDK_Finalize();
    return FAILED;
}
```

用户可以选择为上传到 S3 存储桶中的视频片段增加标签：

```
S3_HLS_SDK_Set_Tag("key1=value1");
```

启动上传线程：

```
S3_HLS_SDK_Start_Upload();
```

在主线程中调用视频帧处理代码

```
// Main thread for processing frames:
while(!exit) {
    // Get video stream from IPC
    int nr_pack = stream->packCount;

        S3_HLS_FRAME_PACK s3_frame_pack;
        s3_frame_pack.item_count = nr_pack;

        for(int i=0; i < nr_pack; i++) {
                IMPEncoderPack *pack = &stream->pack[i];
                // use timestamp generated by encoder
                s3_frame_pack.items[i].timestamp = pack->timestamp;
```

```
                if(pack->length){
                        uint32_t remSize = stream->streamSize - pack->offset;
                        if(remSize < pack->length){ // IPC buffer acrossed internal
ring buffer boundary
                                s3_frame_pack.items[i].first_part_start = (void
*)(stream->virAddr + pack->offset);
                                s3_frame_pack.items[i].first_part_length = remSize;


                                s3_frame_pack.items[i].second_part_start = (void
*)stream->virAddr;
                                s3_frame_pack.items[i].second_part_length =
pack->length - remSize;
                        }else { // IPC buffer does not across internal ring buffer
boundary
                                s3_frame_pack.items[i].first_part_start = (void
*)(stream->virAddr + pack->offset);
                                s3_frame_pack.items[i].first_part_length =
pack->length;


                                s3_frame_pack.items[i].second_part_start = NULL;
                                s3_frame_pack.items[i].second_part_length = 0;
                        }
        }


        S3_HLS_SDK_Put_Video_Frame(&s3_frame_pack);
}
```

当程序退出时，清理已分配的资源

```
S3_HLS_SDK_Finalize();
```

验证运行结果

在 IPC 固件中运行应用，检查目标 S3 桶中是否有对应分片上传。

创建 Certificate Manager 证书 （仅限中国区域部署）

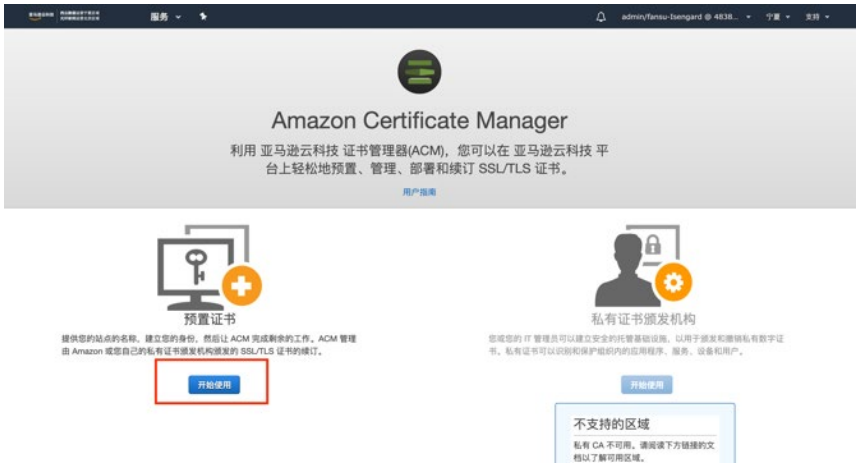
为了能确保该解决方案部署后能以 HTTPS 方式进行调用，必须使用亚马逊云科技 Certificate Manager 证书或者其他第三方的证书。具体使用方式，请参考:

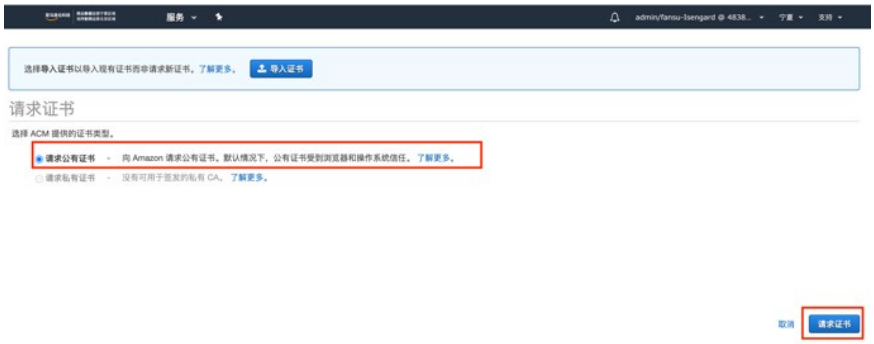
如何上传 SSL 证书并将其导入 Amazon Identity and Access Management (IAM)

本部署指南以使用 Amazon Certificate Manager 为例进行说明。关于亚马逊云科 技 Certificate Manager 更多信息，请参考 :

https://aws.amazon.com/cn/certificate-manager/

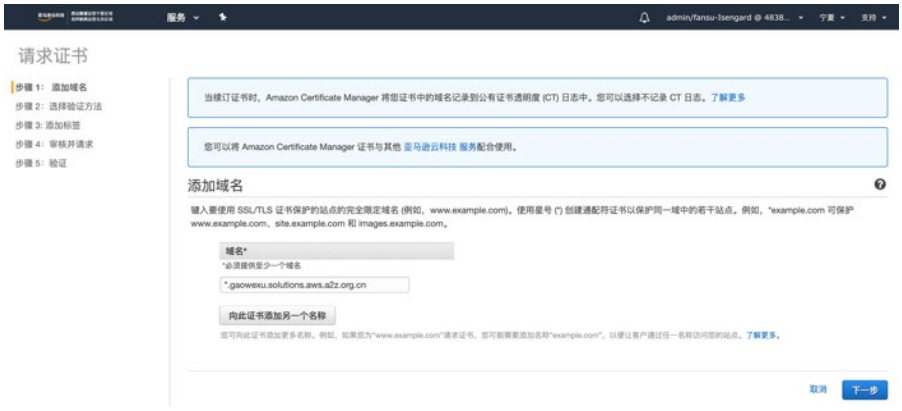
首先，登录亚马逊云科技 Certificate Manager 控制台。依次点击 【预置证书-开始使用】，【请求证书】(请求共有证书)，如下图所示。





在【添加域名】中的【域名】输入框，输入 IP 摄像头服务所需要使用的 域名(需经过 ICP 备案)，如:example.org.cn(用戶输入需输入 自己的 ICP 备案过的域名，如用戶部署在海外区域且无需自定义域名，则可以忽略该小节，直接跳转到云端回放参考架构部署方法)。

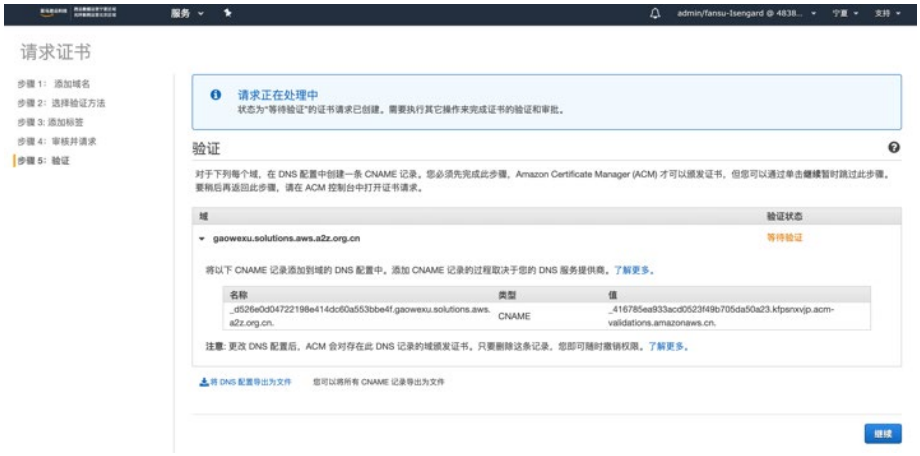
这里我们在 ICP 备案过的域名前加“*”好表示通配符，可以保护同一域中的若干站点，请求证书时域名填写如:*.example.org.cn;如下图所示:



勾选【DNS 验证】，点击【下一步】。

点击【审核】按钮;

点击【确认并请求】，如下图所示:



请记录下 【名称】【类型】及【值】三个字段的值，这三个值需要在后面的 Route53 中添加记录使用。点击【继续】按钮。

Route53 域名认证

在 Route 53 中添加 CNAME 记录，以认证该域名为您所有并可以使用。首先，进入 Route 53 的管理页面，点击你的域名。

如果没有创建托管区域，请参考: https://aws.amazon.com/cn/route53/

点击【创建记录】，在窗口中【记录名称】，【记录类型】和【值】三个输入区域输 入在 Certificate Manager 控制台验证步骤中记录的【名称】【类型】及【值】三个字 段值，然后点击【创建记录】，如下图所示。

创建记录完成后回到 Certificate Manager 控制台界面等待大概 5 分钟。点击刷新按 钮。等待 Certificate Manager 证书的状态变为 【已颁发】。

到此为止，Certificate Manager 证书已经申请完毕。该证书会在后续部署完之后 API Gateway 中自定义域名时使用。

云端回放参考架构部署方法

在由西云数据运营的亚马逊云科技(宁夏)区域或由光环新网运营的亚马逊云科技 (北京)区域中部署该解决方案，您可以直接基于 CloudFormation 模板来进行快速部 署和管理 。

打开亚马逊云科技管理控制台(如果还没登录会先跳转到登录页面，登录后进入模板 启动页面)。默认情况下，中国区此模板在北京区域启动，海外区会在 us-east-1 区域 启动。您同时可以使用控制台右上方的区域选择链接，以在其他区域部署该方案。然后单击下面的按钮以启动亚马逊云科技 CloudFormation 模板。

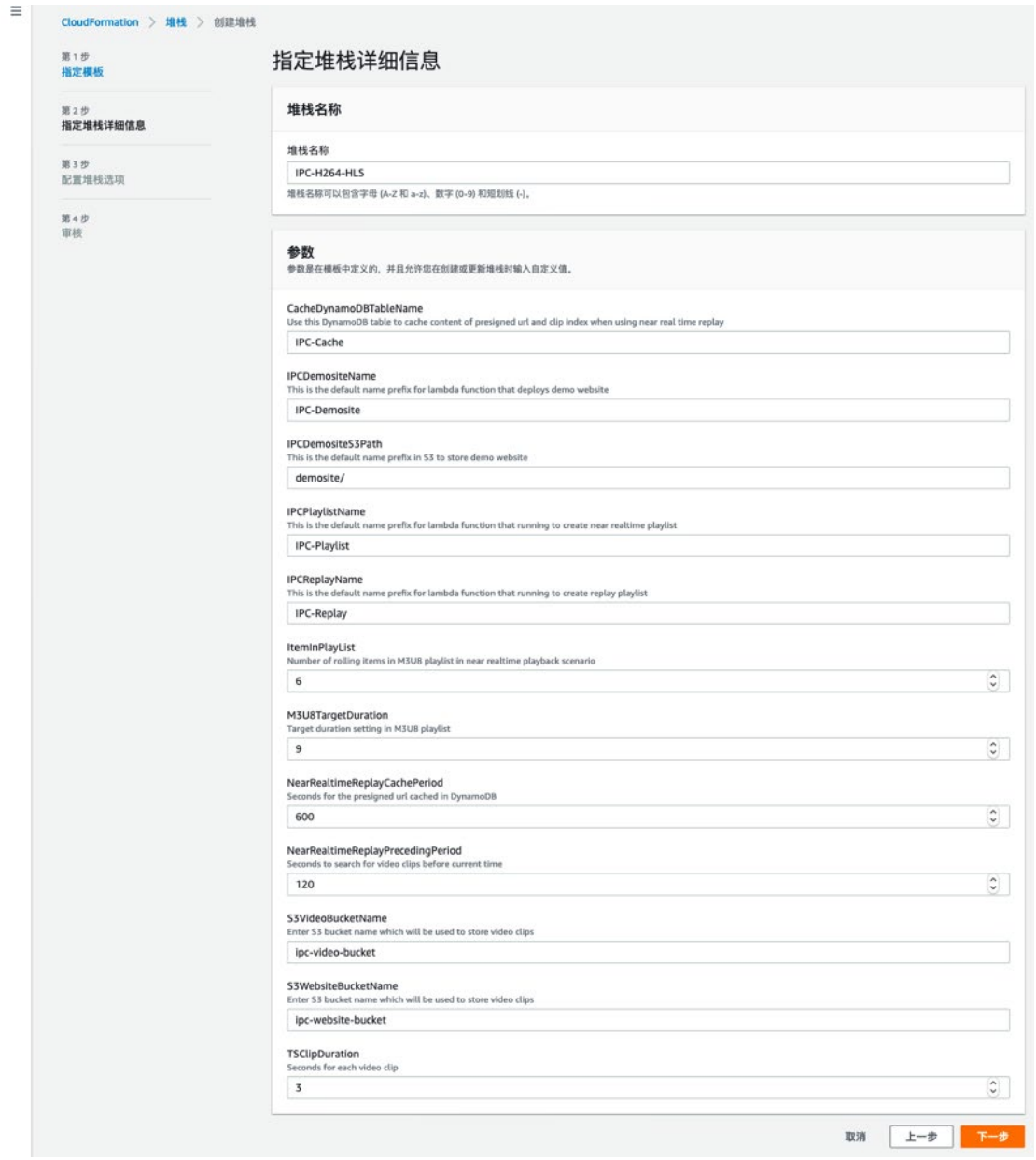
启动 CloudFormation 堆栈

中国区部署链接:

同时该解决方案还支持在海外区部署，部署链接为:

点击 【下一步】。

指定堆栈详细信息



部署过程中一共有 12 个参数需要用戶进行输入，如上图所示。

12 个配置参数的参数名 称及其默认值如下表所示:

参数	默认值
CacheDynamoDBTable	IPC-Cache
IPCDemositeName	IPC-Demosite
IPCDemositeS3Path	demo/
IPCPlaylistName	IPC-Playlist
IPCReplayName	IPC-Replay
IteminPlaylist	6
M3U8TargetDuration	9
NearRealtimeReplayCachePeriod	600
NearRealtimeReplayPrecedingPeriod	120
S3VideoBucketName	ipc-video-bucket
S3WebsiteBucketName	ipc-website-bucket
TSClipDuration	3

12 个配置参数的具体含义与作用如下:

- CacheDynamoDBTable

  近实时回看时用于缓存用户 Session 数据的 DynamoDB 表名称

- IPCDemositeName

  用于部署 Demo 网站的 Lambda 函数和对应角色名称前缀

- IPCDemositeS3Path

  Demo 网站存储在 S3 桶中的路径。默认部署在前缀为 demo 的目录下

- IPCPlaylistName

Demo 网站使用的生成近实时回看 M3U8 播放列表的 Lambda 函数名称前缀

- IPCReplayName

  Demo 网站使用的生成选时回看 M3U8 播放列表的 Lambda 函数名称前缀

- IteminPlaylist

  近实时回看模式下，M3U8 播放列表滚动更新过程中，最多视频分片数量

- M3U8TargetDuration

  近实时回看模式下，M3U8 播放列表滚动更新过程中，缓存的视频时长，单位：秒

- NearRealtimeReplayCachePeriod

  近实时回看模式下，DynamoDB 中缓存的 Session 信息时长，单位：秒

- NearRealtimeReplayPrecedingPeriod

  近实时回看模式下，生成会看列表时，从当前时间向前检索的时间范围。单位：秒。调整此值可以一定程度缓解 IPC 内置时钟与标准时间不一致时，特别是 IPC 始终时间较慢时，无法检索到会看分片的情况

  注意：由于 SSL 传输要求，当 IPC 时钟与标准时间差异较大时，会导致视频分片上传失败。建议 IPC 内部系统包含 NTP 时间同步机制。

- S3VideoBucketName

  用于存储 IPC 上传视频的 S3 存储桶名称前缀。该存储桶亦用于存储实例视频文件

注意：为了确保视频数据安全，在删除堆栈时，该存储桶不会自动被删除。如果确认该存储桶中的内容不在需要，可以手动清空该存储桶，然后删除堆栈。如果该存储桶中数据仍然需要保存，在删除堆栈时，请选择保留该存储桶。

- S3WebsiteBucketName

  用于托管 Web 页面，并提供视频回看入口

- TSClipDuration

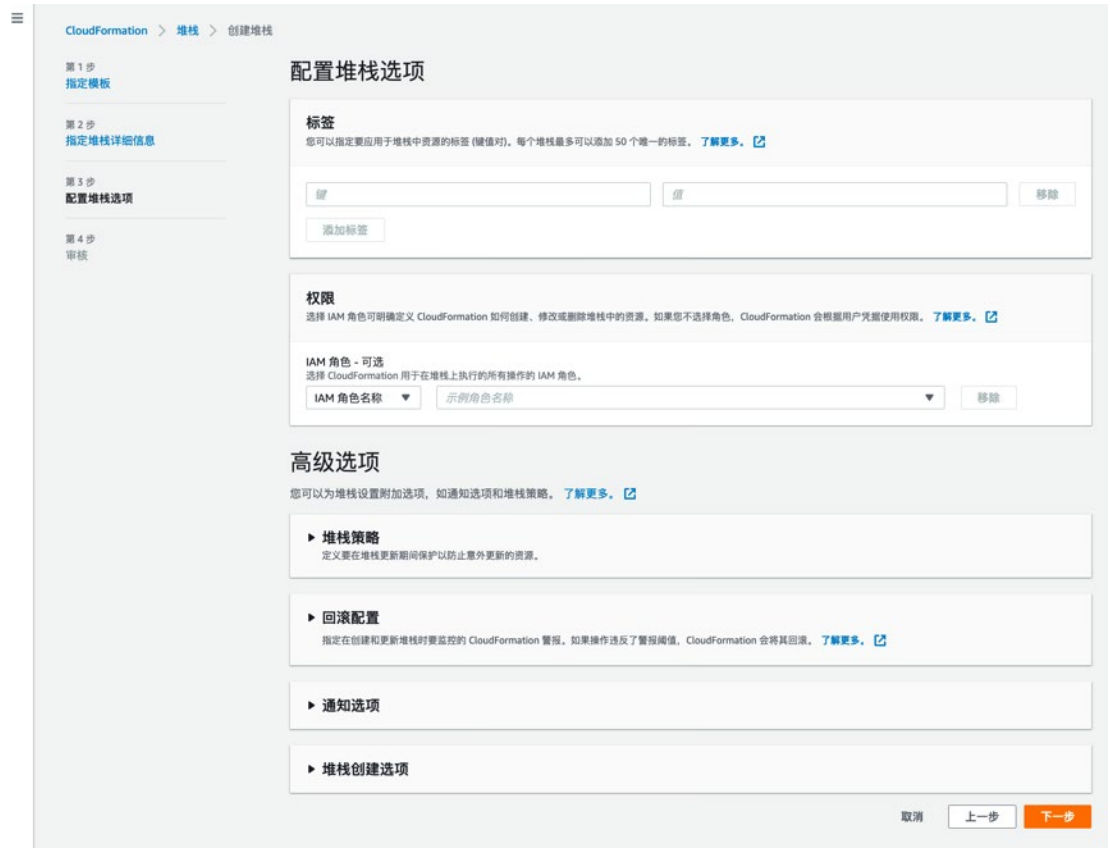
  每个视频分片的时间长度。需要结合 IPC 设备侧 SDK 配置进行调整。

  单位：秒。

  注意：示例视频使用 3 秒分片，调整此参数会影响示例视频回看

堆栈配置选项

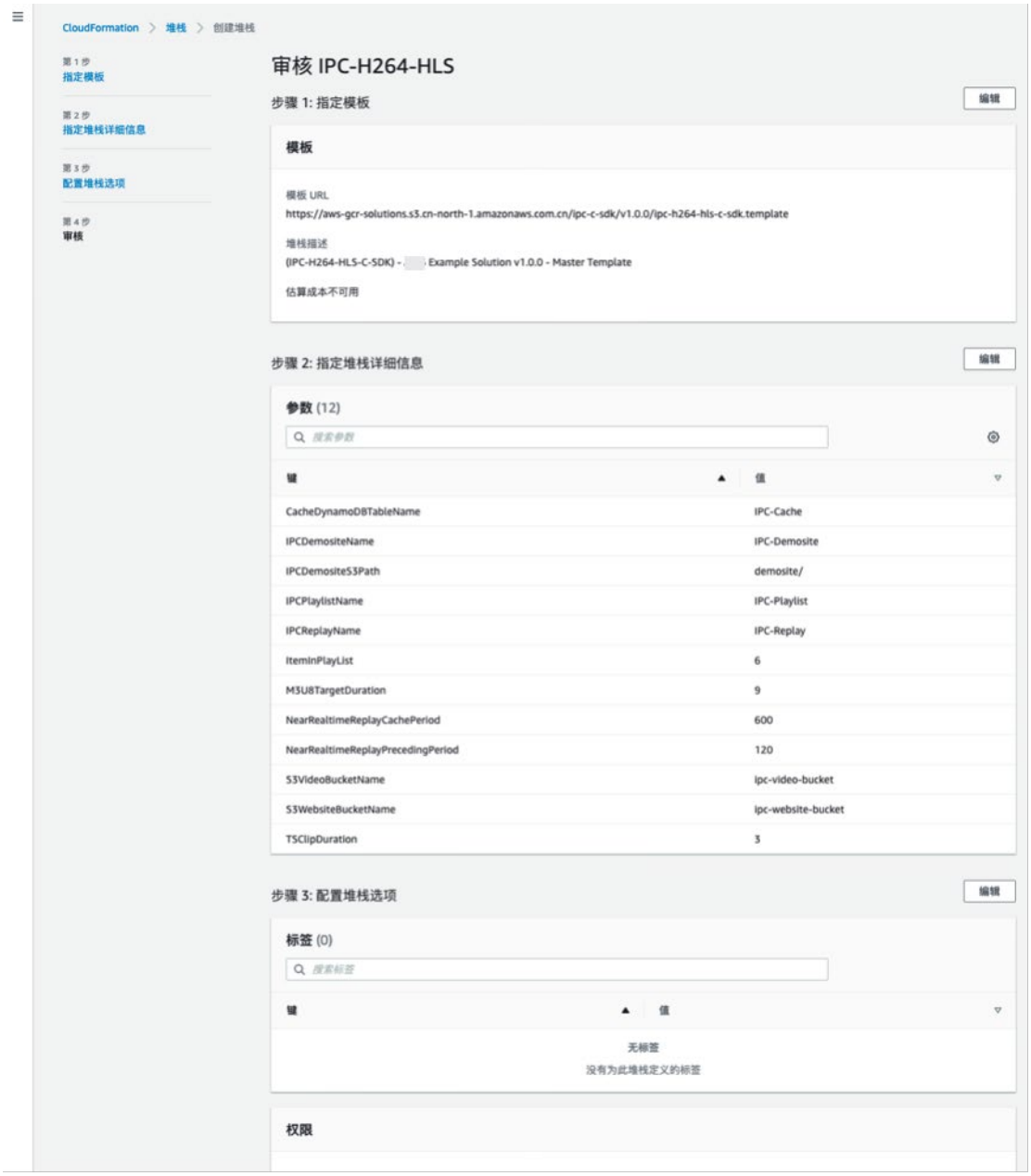
保持默认值不变即可：

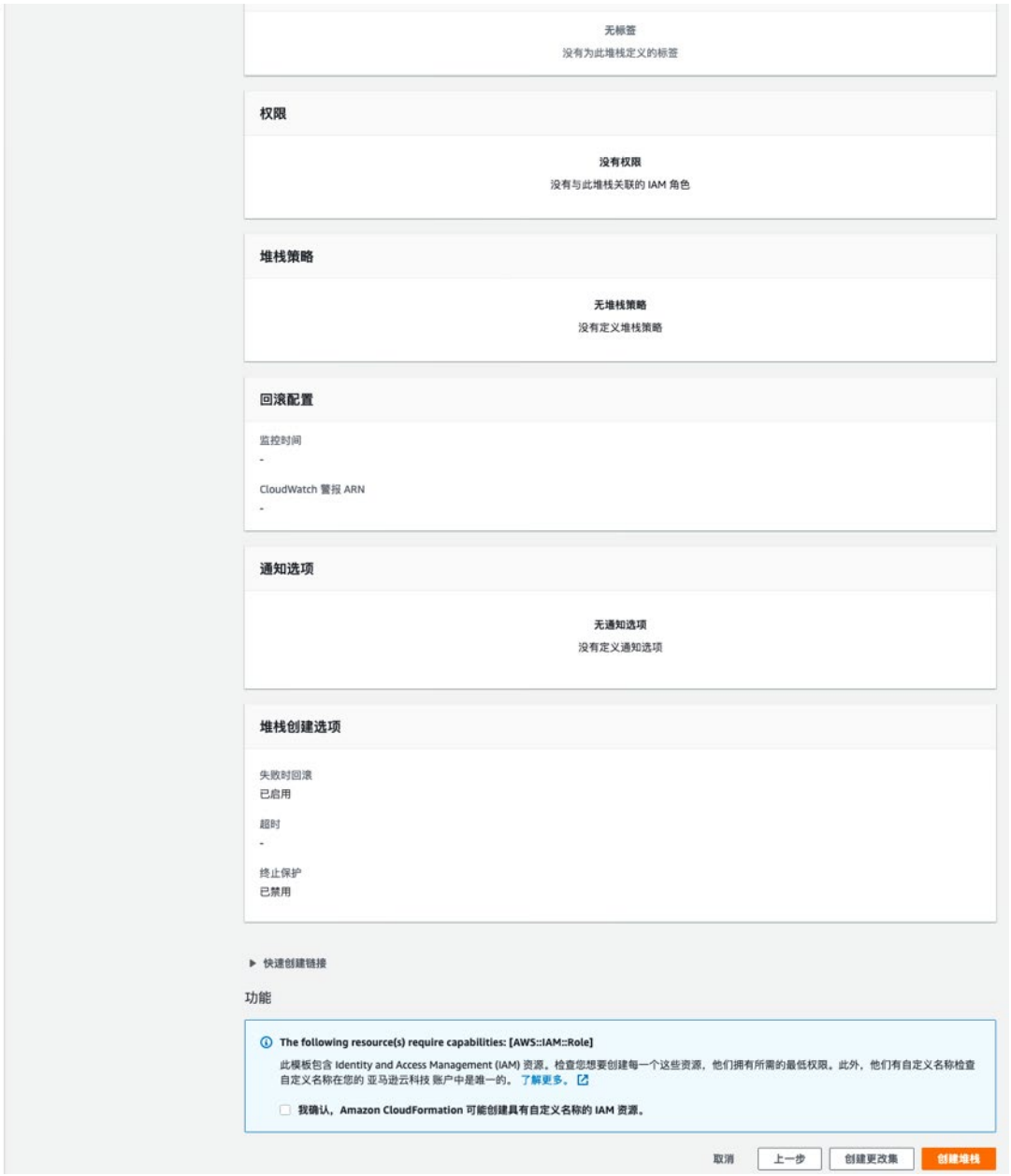


点击【下一步】。

审核堆栈配置

保持默认值。

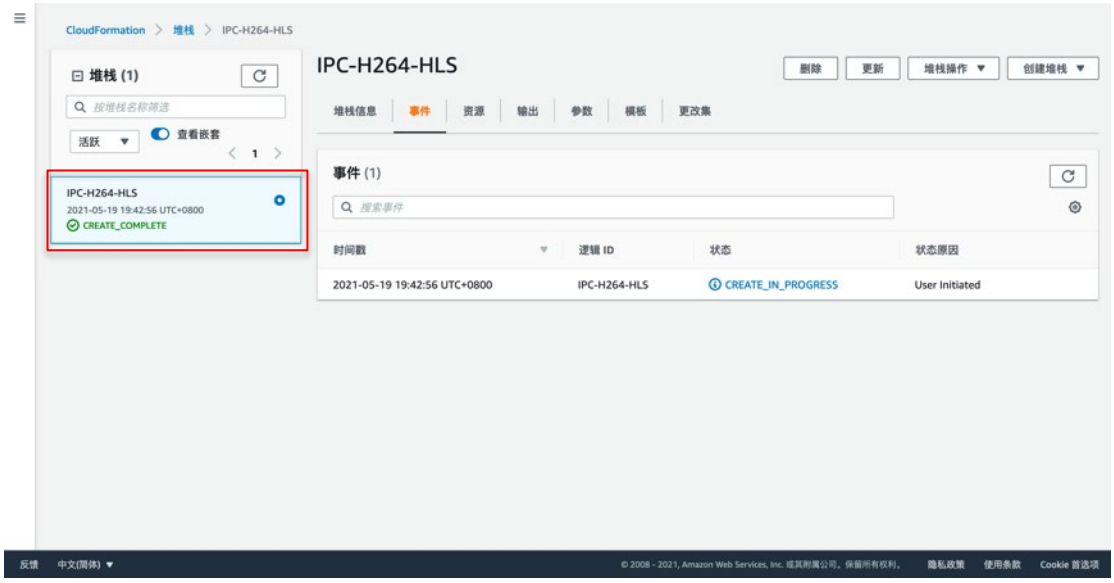




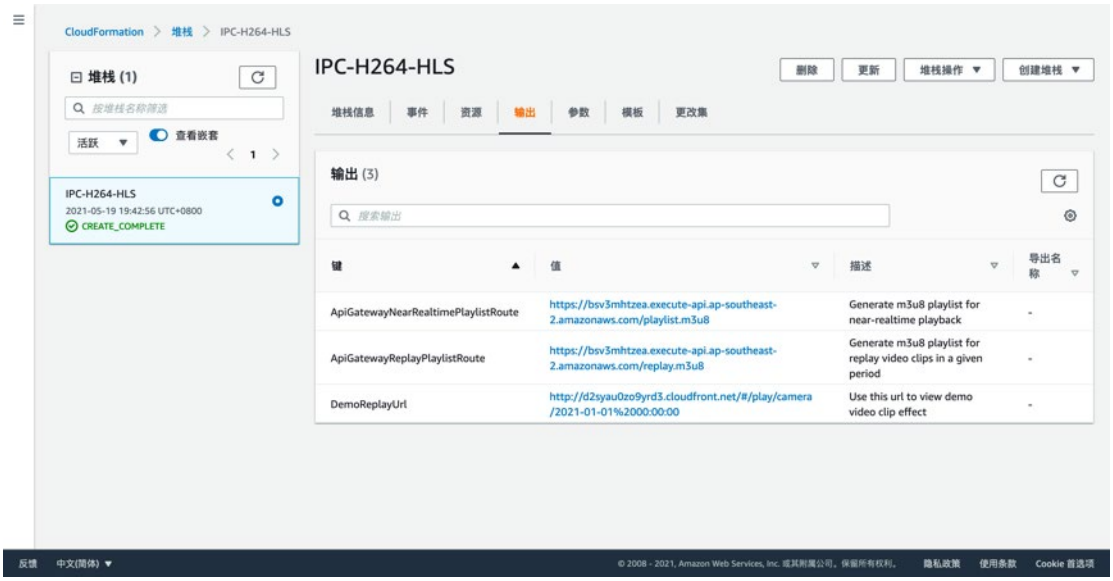
请勾选 “我确认，Amazon CloudFormation 可能创建具有自定义名称的 IAM 资源。”的单选框。然后，点击 【创建堆栈】。

界面跳转到堆栈创建事件列表。

等待大概 10 分钟，界面左侧显示堆栈创建完成。



点击堆栈的【输出】标签页。



获得 3 个 URL 链接。

- ApiGatewayNearRealtimePlaylistRoute

  用于近实时回看使用的 M3U8 播放列表。该路径接受一个参数 camera_id。该参数等同于 IPC 设备上传到 S3 时使用的前缀。生成播放列表时，Lambda 函数会在该前缀下检索视频分片

  该 URL 格式如下：https://bsv3mhtzea.execute-api.ap-southeast-2.amazonaws.com/playlist.m3u8

- ApiGatewayReplayPlaylistRoute

  用于选时回看使用的 M3U8 播放列表。该路径接受 3 个参数 camera_id，start, end。camera_id 参数等同于 IPC 设备上传到 S3 时使用的前缀。生成播放列表时，Lambda 函数会在该前缀下检索视频分片。start 和 end 用于指定回放时间段范围格式均为 yyyy-MM-dd HH-

mm-ss。需要注意的是 IPC 上传时使用的是 UTC 时间，因此这里查询传入也需要使用 UTC 时间进行检索。

该 URL 格式如下：https://bsv3mhtzea.execute-api.ap-southeast-2.amazonaws.com/replay.m3u8

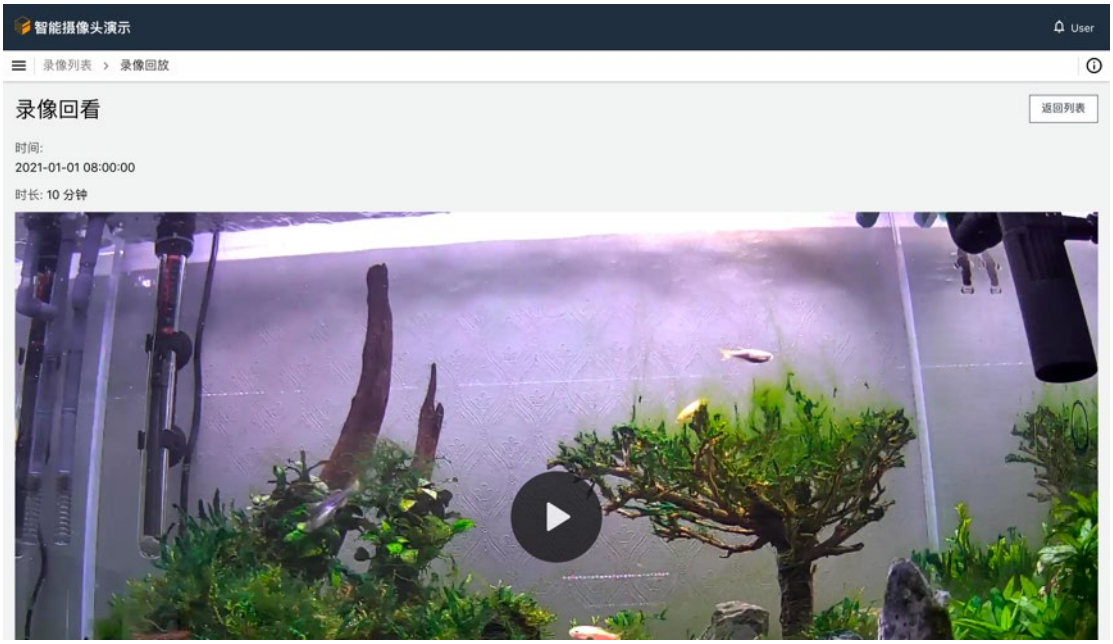
- DemoReplayUrl

  本参考架构包含了示例视频。在全球区域部署后可以使用该 URL 观看示例视频。中国区域部署后，需要完成对应配置，并使用自定义域名进行观看。

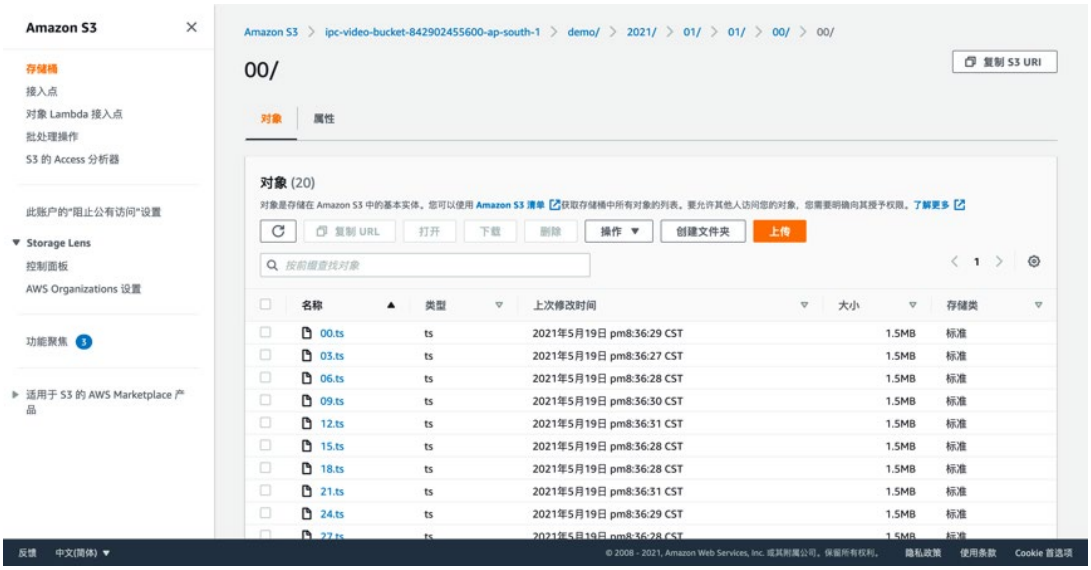
  该 URL 格式如下：
  http://d2syau0zo9yrd3.cloudfront.net/#/play/camera/2021-01-01%2000:00:00

需要注意的是，如果在海外区域部署，该 URL 可以直接使用浏览器访问，但是在中国区部署，需要 ICP 备案，并结合 API Gateway 自定义域名使用。

使用第三个链接可以进入示例回看页面。效果如下：



该回看内容使用本方案 SDK 预先录制并上传到视频存储桶中的 demo 前缀下。用户可以在该前缀下找到对应视频分片。



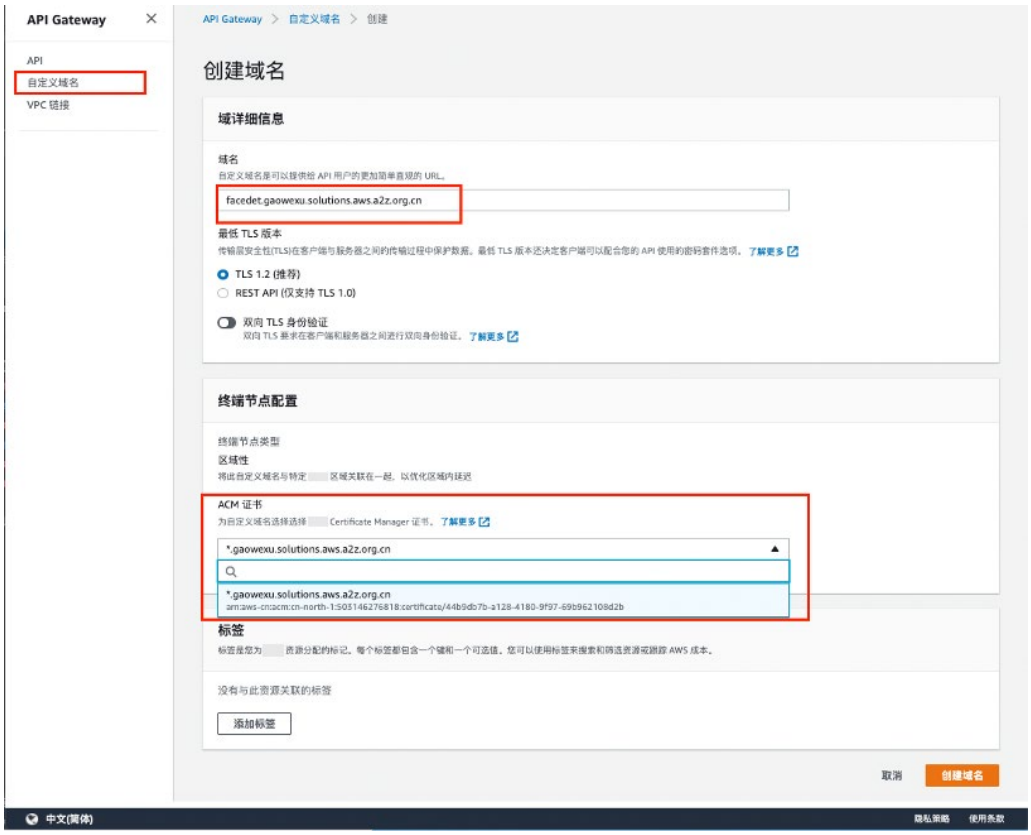
API Gateway 自定义域名 （仅限中国区域部署）

进入 API Gateway 控制台界面，点击 【自定义域名】-【创建】按钮，创建过程中需要 输入自定义的域名名称以及相关的 ACM 证书，这里以前缀形式定义

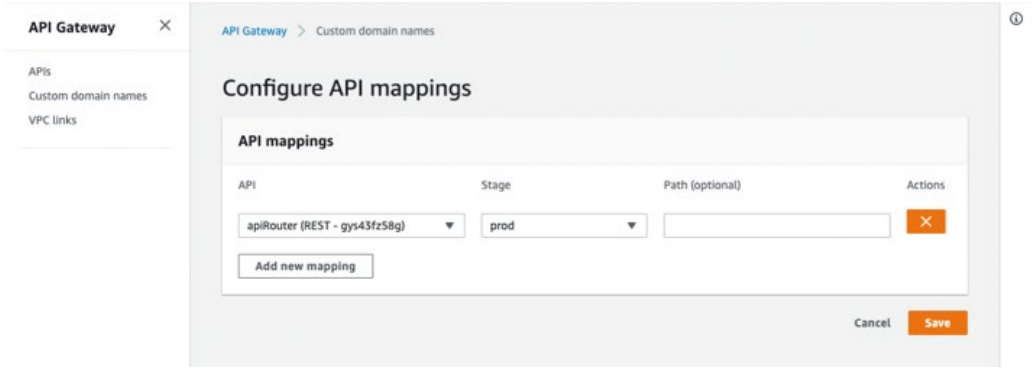
域名，即在 ICP 备 案的域名前添加了 facedet 前缀，如果部署的是人形检测，人脸比较，可以自行更改， 如 bodydet, facecomp 等等。如下图所示:

ACM 证书选择前述在 Certificate Manager 控制台中已颁发的证书，点击【创建域名】；

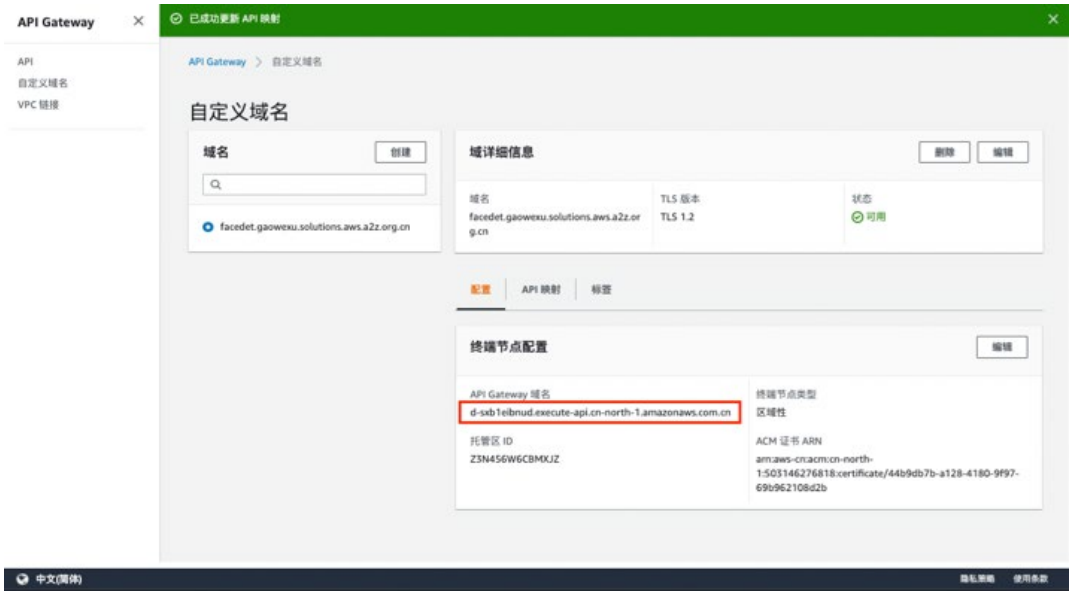
创建域名完成之后，点击控制台上的【API 映射】，如下图所示:



点击【配置 API 映射】-【添加新映射】，选择对应的 API，阶段，路径保持不变，无 需填任何值，点击【保存】，如下图所示:



最后需要将自定义的域名更新到 Route 53 中，即将【终端节点配置】中的 API Gateway 域名(如: d-sxb1eibnud.execute-api.cn-north-1.amazonaws.com.cn)记 录下来，接下来需要在 Route 53 中的创建一条 CNAME 记录来关联自定义的域名和 API Gateway 域名。



切换至 Route53 控制台，进入域名所在的托管区域，点击【创建记录】，输入记录名 称为刚刚自定义域名的前缀部分，如 facedet; 记录类型为 CNAME 类型，值填入 API Gateway 控制台【自定义域名】-【终端节点配置】中的 API Gateway 域名(如:d- sxb1eibnud.execute-api.cn-north-1.amazonaws.com.cn)

点击【创建记录】完成自定义域名解析。

示例调用

近实时回看调用 API 的伪代码示例如下：

```
<video id="roomVideo" ...>
  <source src="https://xxxxxxxx.execute-api.ap-southeast-2.amazonaws.com.cn/playlist.m3u8?camera_id=demo"  type="application/x-mpegURL">
</video>
```

选时回看调用 API 的伪代码示例如下：

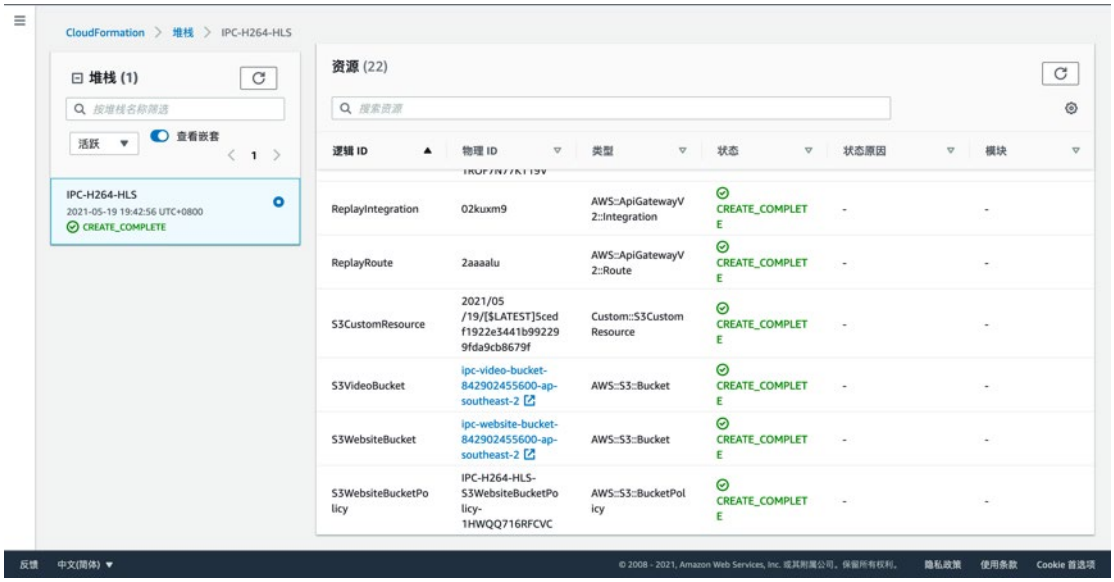
```
<video id="roomVideo" ...>
  <source src="https://xxxxxxxx.execute-api.ap-southeast-
2.amazonaws.com.cn/replay.m3u8?camera_id=demo&start=2021-05-01
11:00:00&end=2021-05-01 11:10:00"  type="application/x-mpegURL">
</video>
```

删除堆栈

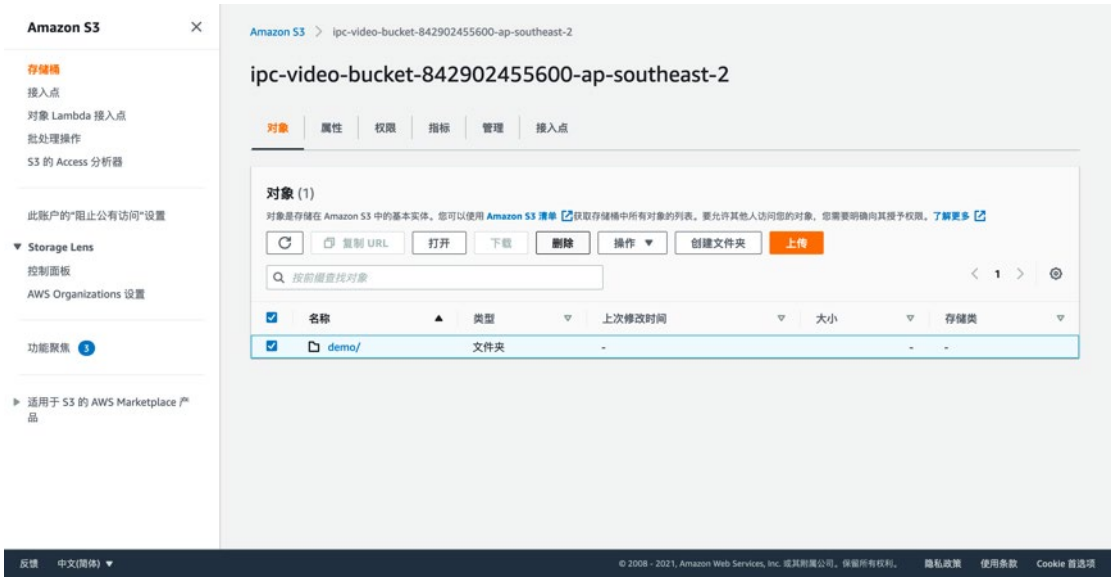
注意：由于视频存储桶中可能存储有重要的视频分片信息，因此在删除堆栈时，不会自动清除该存储桶的内容。用户需要在删除堆栈前手工清空存储桶，或在删除堆栈时选择保留存储桶。

删除视频存储桶并删除堆栈

首先在堆栈配置中查看资源选项卡



单击”S3VideoBucket”右侧的链接，打开 S3 存储桶。选中存储桶中文件，并单击“删除”。

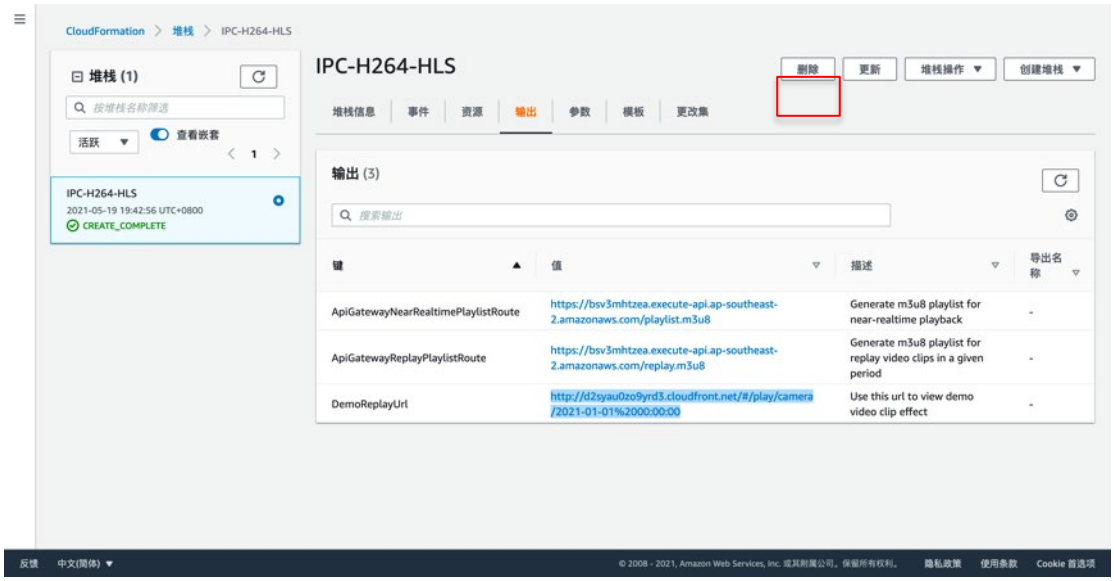


在打开的页面底部按照提示输入“永久删除”或提示中指定的内容，单击“删除对象”。

重复以上步骤，直至清空存储桶。

用户也可以在 S3 桶管理界面直接清空 S3 桶，但务必仔细检查需要清空的 S3 桶名称，以免误操作。

在清空 S3 桶后，回到 CloudFormation 堆栈管理界面。



单击删除按钮。

单击删除堆栈从而删除整个堆栈。

等待约 10 分钟，检查 CloudFormation 堆栈页面，堆栈消失说明删除成功。

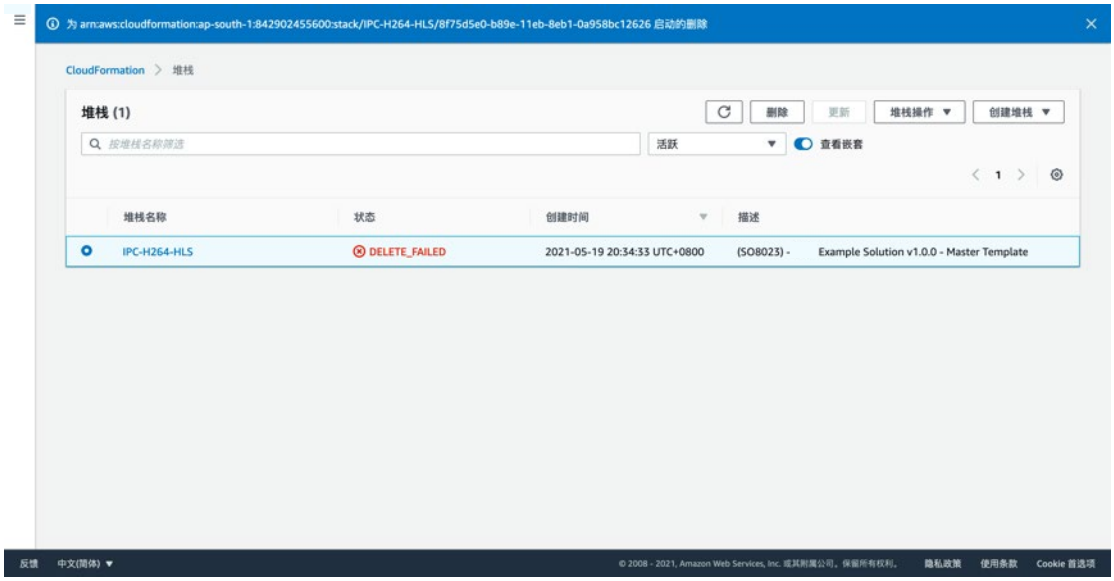
删除堆栈但保留视频存储桶

单机堆栈页面上的“删除”按钮。

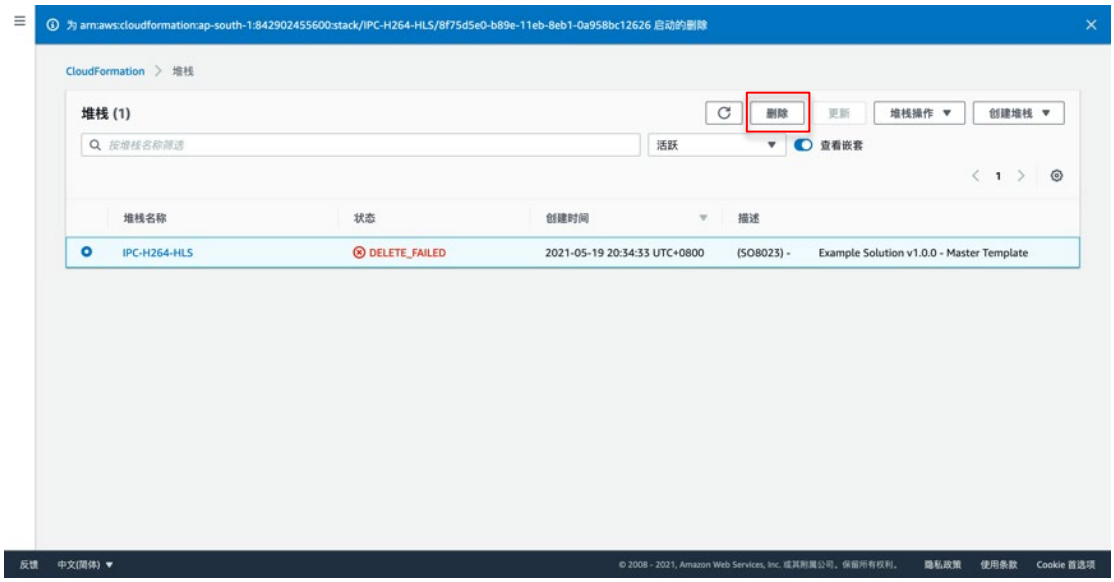
单击删除堆栈。

等待约 10 分钟时间。

由于 S3 视频存储桶中仍然存在视频数据，因此堆栈删除失败。效果如下图所示：



再次单击页面上的删除按钮。



这一次，页面会显示保留存储桶的提示。在要保留的资源中选中 S3VideoBucket，则可以删除堆栈，并保留视频存储桶。在选中该资源后，单击“删除堆栈“。

等待 1 分钟左右，堆栈删除成功。进入 S3 服务，该桶和桶中视频文件得以保留。

注意：在同一个区域内部重新部署时，需要首先完成该视频桶内容的迁移并删除视频桶。否则再次部署可能会失败。