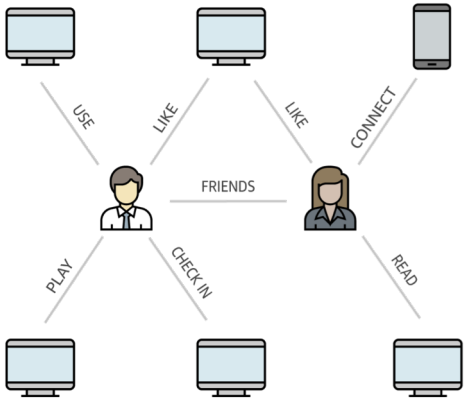
云计算上的图数据库 – Amazon Neptune

费良宏，Technical Evangelist @ AWS
lianghon@amazon.com

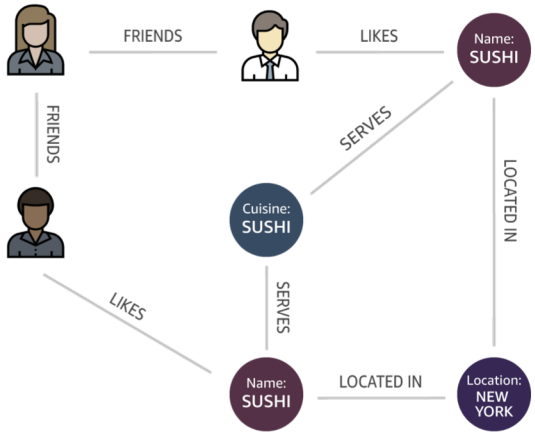
内容提要

- 图数据库的特点及应用场景
- Amazon Neptune 概述
- Amazon Neptune 技术浅析
- 用户案例

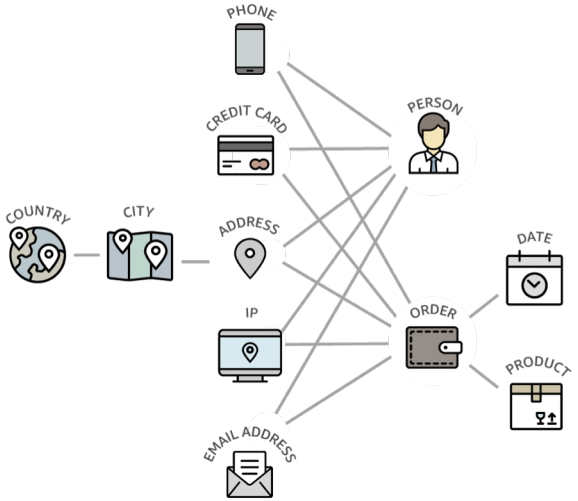
高度连接的数据



社交媒体



餐厅的推荐系统



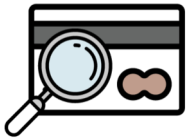
零售的欺诈检测

高度连接的数据的使用场景

社交网络

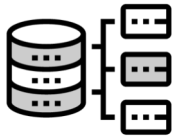
推荐系统

知识图谱



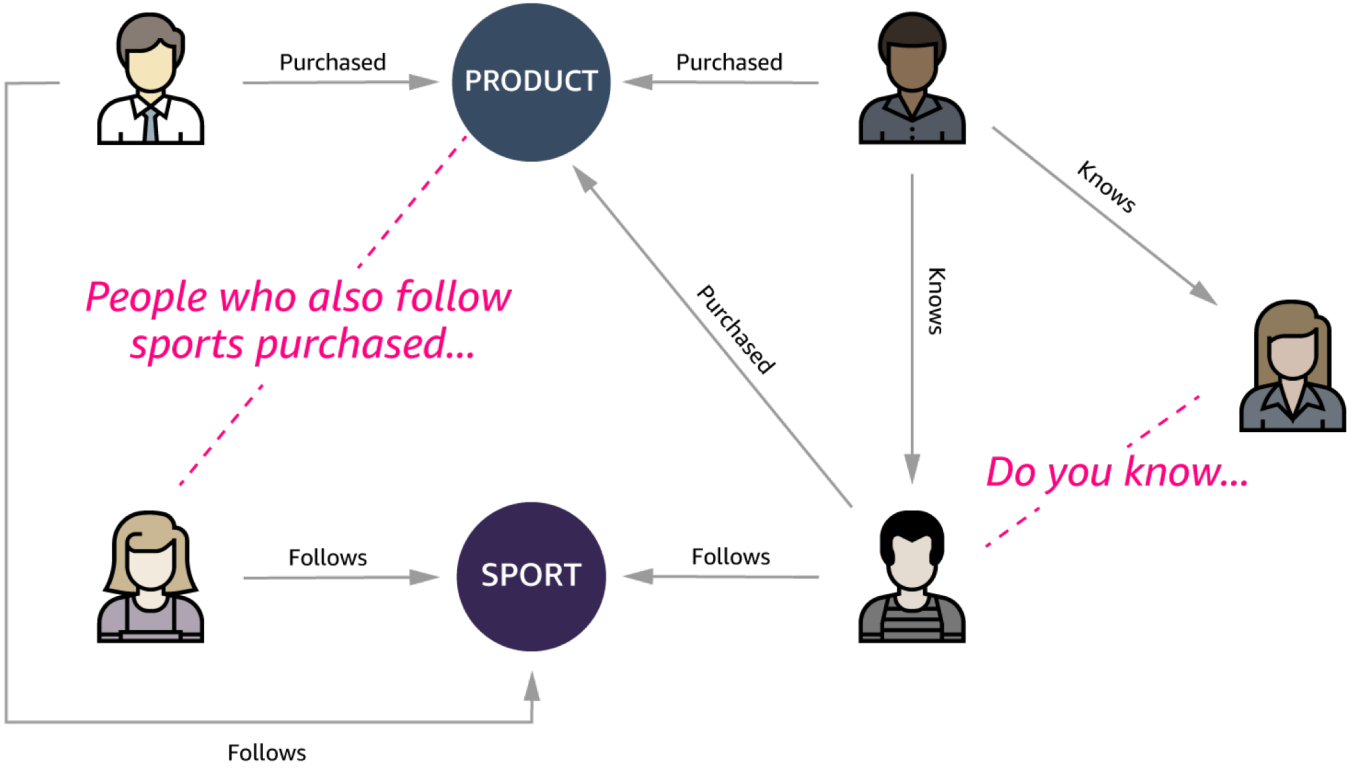
欺诈检测

生命科学



网络 & IT 运维

基于关系的推荐系统

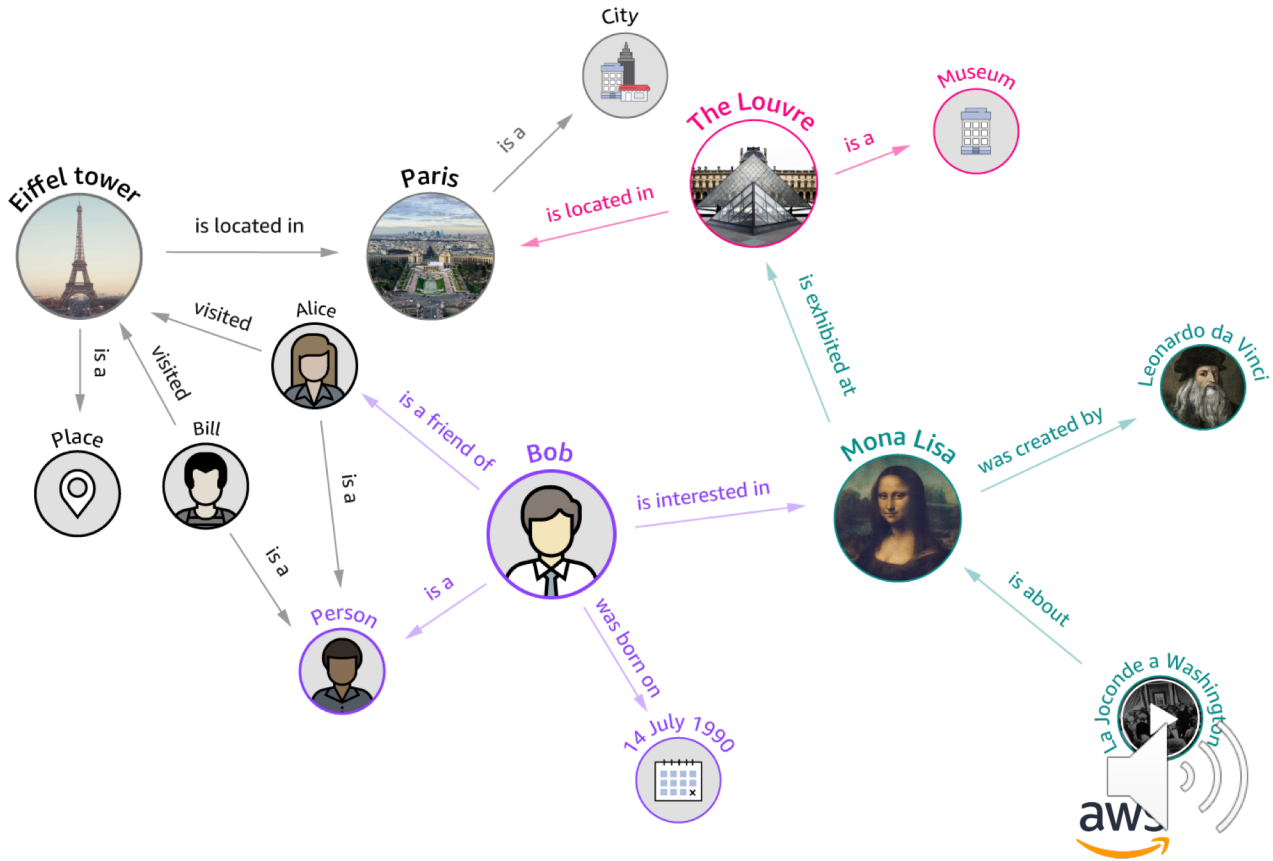

Purchased
PRODUCT
Purchased
Knows
Knows
Purchased
People who also follow sports purchased...
Do you know...
Follows
SPORT
Follows
Follows

知识图谱应用

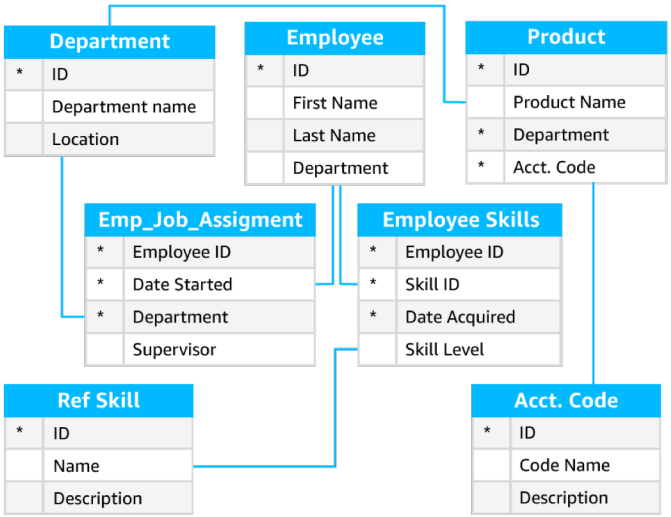
谁绘制了蒙娜丽莎？

爱丽丝在巴黎时应该参观哪些博物馆?

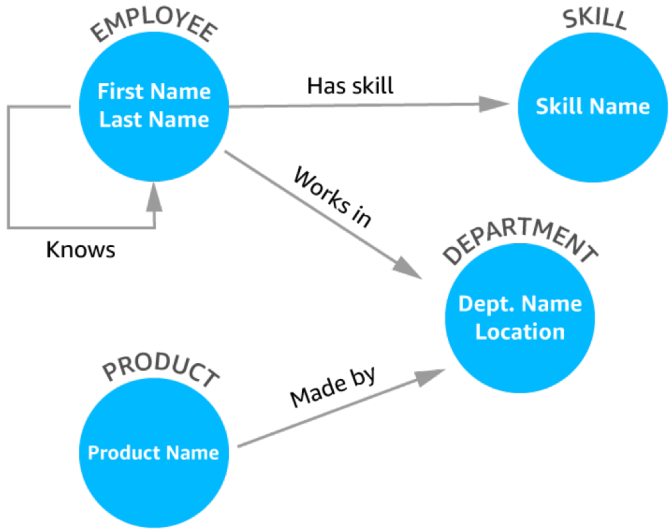
在卢浮宫有哪些艺术家的油画?



高度连接数据的处理方法



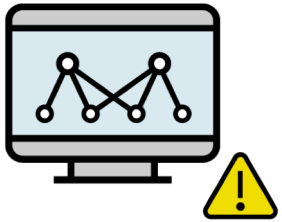
为特定业务流程而构建



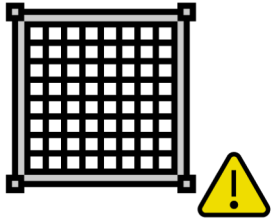
为回答关于关系的问题而构建

关系数据库的挑战- 构建高度连接的数据应用

不自然的对图数据的查询



低效的图处理



刚性模式对于数据的变化是不灵活的

aws

现有的图数据库的挑战

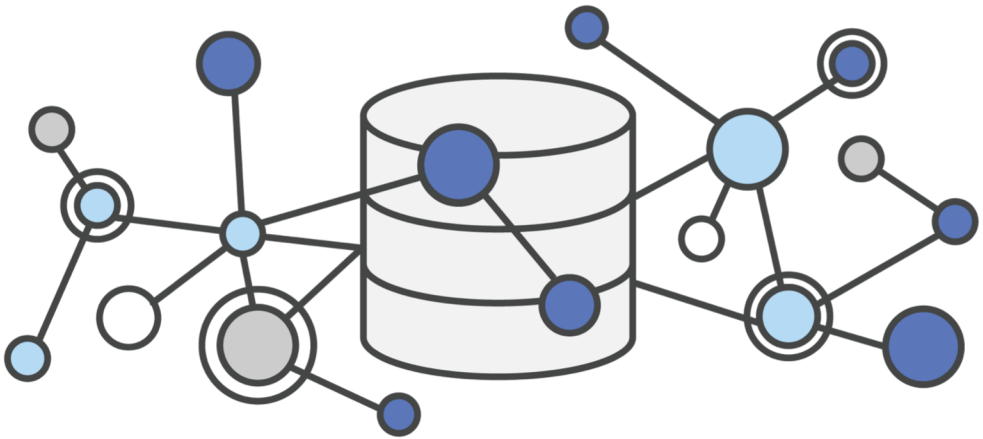
难于扩展

难以维护高可用性

太昂贵

对开放标准的支持有限

图数据库 – 为高效存储和检索高度连接的数据而优化



AWS 上的数据库服务

Relational Databases

Amazon RDS

- Aurora
- Commercial
- Community

Amazon Redshift

Data Warehouse

Non-Relational Databases

NEW!

Amazon DynamoDB

Key Value

Document

Amazon ElastiCache

In-Memory Data Store

Amazon Neptune

Graph

AWS Database Migration Service

图（Graph）模型和框架

属性图
(PROPERTY GRAPH)

开源的 Apache TinkerPop™

Gremlin Traversal Language

资源描述框架RDF
(RESOURCE DESCRIPTION FRAMEWORK)

W3C 标准

SPARQL Query Language

aws

属性图

属性图是具有各自属性(即键/值对)的顶点和边的集合

- **顶点（Vertex）** 顶点代表实体/域
- **边（Edge）** 表示顶点之间的方向关系
  - 每条边都有一个表示关系类型的标签
- 每个顶点和边都有唯一的标识符
- 顶点和边可以有**属性**
- **属性**表示关于顶点和边的非关系信息

属性图 & APACHE TINKERPOP

- **Apache TinkerPop**

  针对属性图的开源图形计算框架

- **Gremlin**

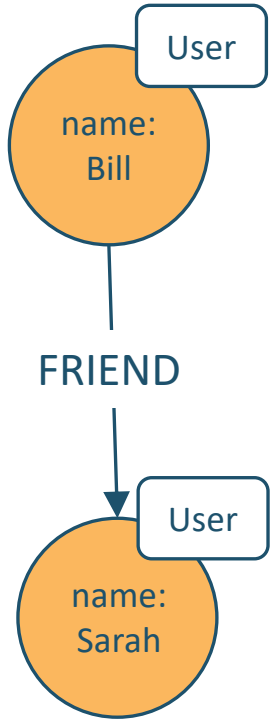
  图遍历语言用于图的分析

Amazon Neptune 与 Tinkerpop Gremlin 3.3.0(最新版本于2017年8月发布)完全兼容，并为Gremlin查询语言提供了优化的查询执行引擎

aws

建立一个 TinkerPop 图

```
//Connect to Neptune and receive a remote graph, g.

user1 = g.addVertex (id, 1,  label, "User", "name", "Bill");
user2 = g.addVertex (id, 2,  label, "User", "name", "Sarah");

...

user1.addEdge("FRIEND", user2, id, 21);
```



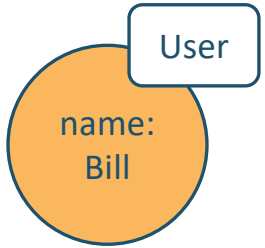
Gremlin (Apache TinkerPop 3.3)

资源描述框架（RDF）图

- **RDF图** 被描述为三元组的集合:主语、谓语和宾语

主语：
谓语：
宾语：

```
<http://www.socialnetwork.com/person#1>
        rdf:type contacts:User;
        contact:name: "Bill" .
```



- **国际化资源标识符(IRIs)** 唯一地标识主题

IRI

```
<http://www.socialnetwork.com/person#1>
```

- 宾语可以是 IRI 或者文字
  - RDF中的文字类似于属性，RDF支持XML数据类型
  - 当对象是IRI时，它在图中形成一条“边”

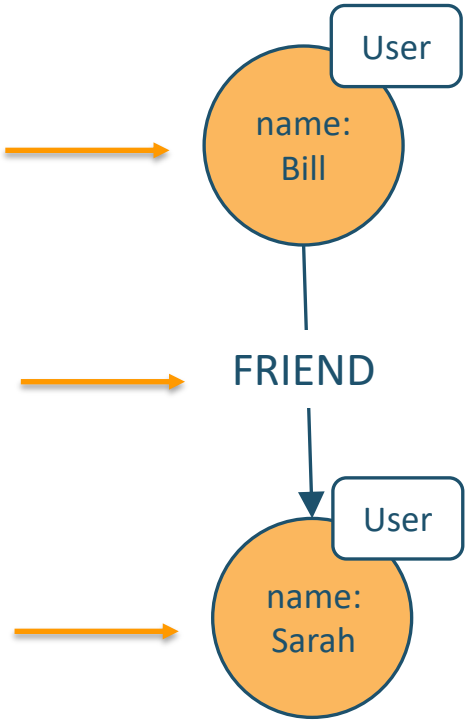
主语：
谓语：
宾语 (IRI)

```
<http://www.socialnetwork.com/person#1>
        contacts:friend
        <http://www.socialnetwork.com/person#2> .
```

aws

RDF 三元组示例

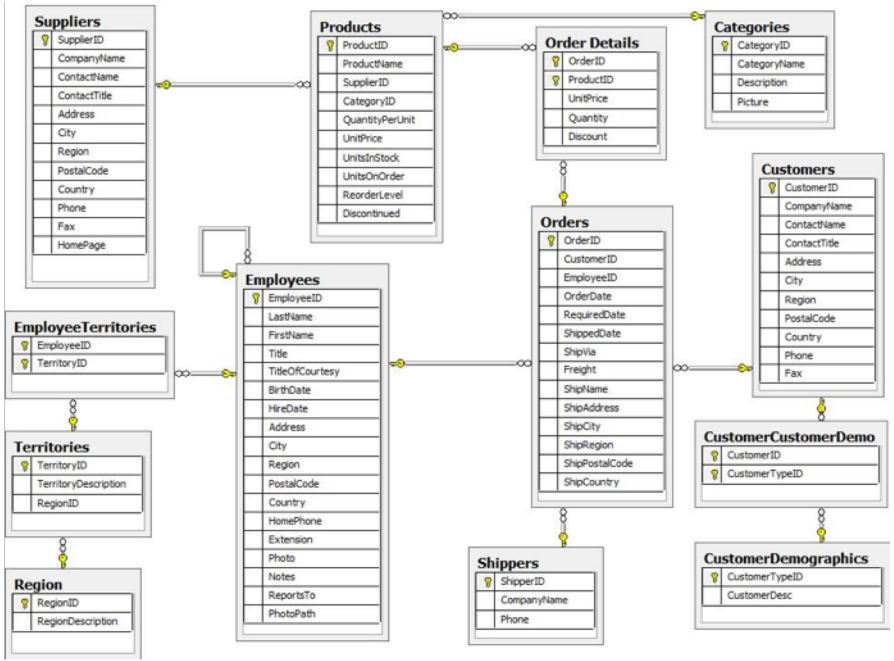
```
@prefix contacts: <http://www.socialnetwork.com/people#>.
<http://www.socialnetwork.com/person#1>
     rdf:type contacts:User;
     contact:name: ”Bill” .

<http://www.socialnetwork.com/person#1>
     contacts:friend <http://www.socialnetwork.com/person#2> .


<http://www.socialnetwork.com/person#2>
     rdf:type contacts:User;
     contact:name: ”Sarah” .
```



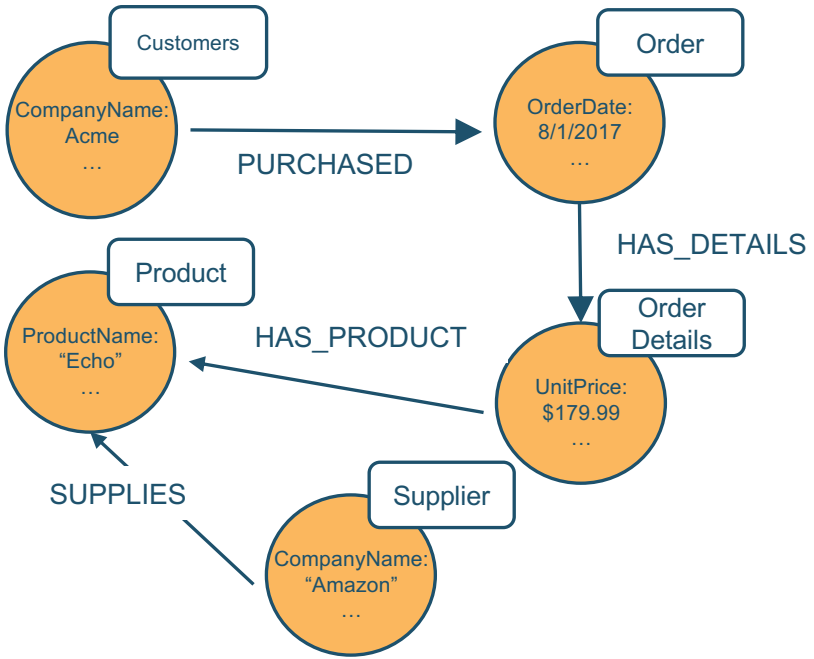
RDF
(Turtle Serialization)

图 vs. 关系 数据库模型

关系模型



图模型子集



* 来源：http://www.playnexacro.com/index.html#show:article

aws

SQL 关系数据库查询

查找购买过Echo产品的公司的名称

```
SELECT distinct c.CompanyName
FROM customers AS c
    JOIN orders AS o ON                  /* Join the customer from the order */
        (c.CustomerID = o.CustomerID)
    JOIN order_details AS od     /* Join the order details from the order
*/
        ON (o.OrderID = od.OrderID)
    JOIN products as p         /* Join the products from the order details
*/
        ON (od.ProductID = p.ProductID)
    WHERE p.ProductName = 'Echo';       /* Find the product named ‘Echo’ */
```

aws

SPARQL 声明式图查询

查找购买过Echo产品的公司的名称

```
PREFIX sales_db: <http://sales.widget.com/>
SELECT distinct ?comp_name WHERE {
    ?customer   <sales_db:HAS_ORDER> ?order ;       #customer graph pattern
            <sales_db:CompanyName> ?comp_name .       #orders graph pattern
    ?order  <sales_db:HAS_DETAILS> ?order_d .  #order details graph pattern
    ?order_d    <sales_db:HAS_PRODUCT> ?product .       #products graph
pattern
    ?product    <sales_db:ProductName> “Echo” .
}
```

* 来源：http://www.playnexacro.com/index.html#show:article

Gremlin 命令式图遍历

查找购买过Echo产品的公司的名称

```
/* All products named ”Echo” */
g.V().hasLabel(‘Product’).has('name','Echo')
    .in(’HAS_PRODUCT') /* Traverse to order details */
    .in(‘HAS_DETAILS’) /* Traverse to order */
    .in(’HAS_ORDER’) /* Traverse to Customer */
    .values(’CompanyName’).dedup()  /* Unique Company Name */
```

aws

Python 样例代码（Gremlin）

- 安装 gremlinpython 程序包

pip3 install gremlinpython --user

- 遍历图

```
from gremlin_python import statics
from gremlin_python.structure.graph import Graph
from gremlin_python.process.graph_traversal import __
from gremlin_python.process.strategies import *
from gremlin_python.driver.driver_remote_connection import DriverRemoteConnection

graph = Graph()
g = graph.traversal().withRemote(DriverRemoteConnection('ws://your-neptune-endpoint:8182/gremlin','g'))
print(g.V().limit(2).toList())
```

AMAZON NEPTUNE
全托管的图数据库

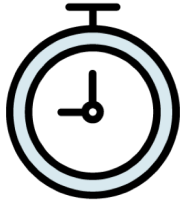
开放

支持 Apache TinkerPop & W3C RDF 图模型

快速

查询十亿级关系数据仅有毫秒级延迟

可靠



数据拥有跨3个可用区的6个副本，具有完全备份和恢复

易用

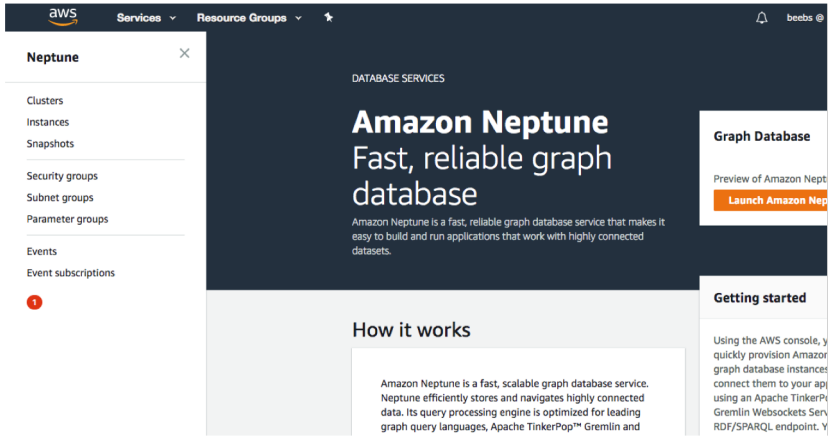
使用Gremlin和SPARQL轻松构建强大的查询

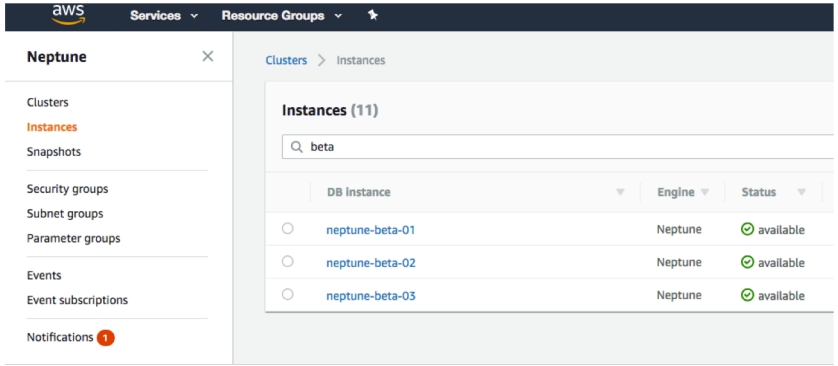
aws

全托管的服务

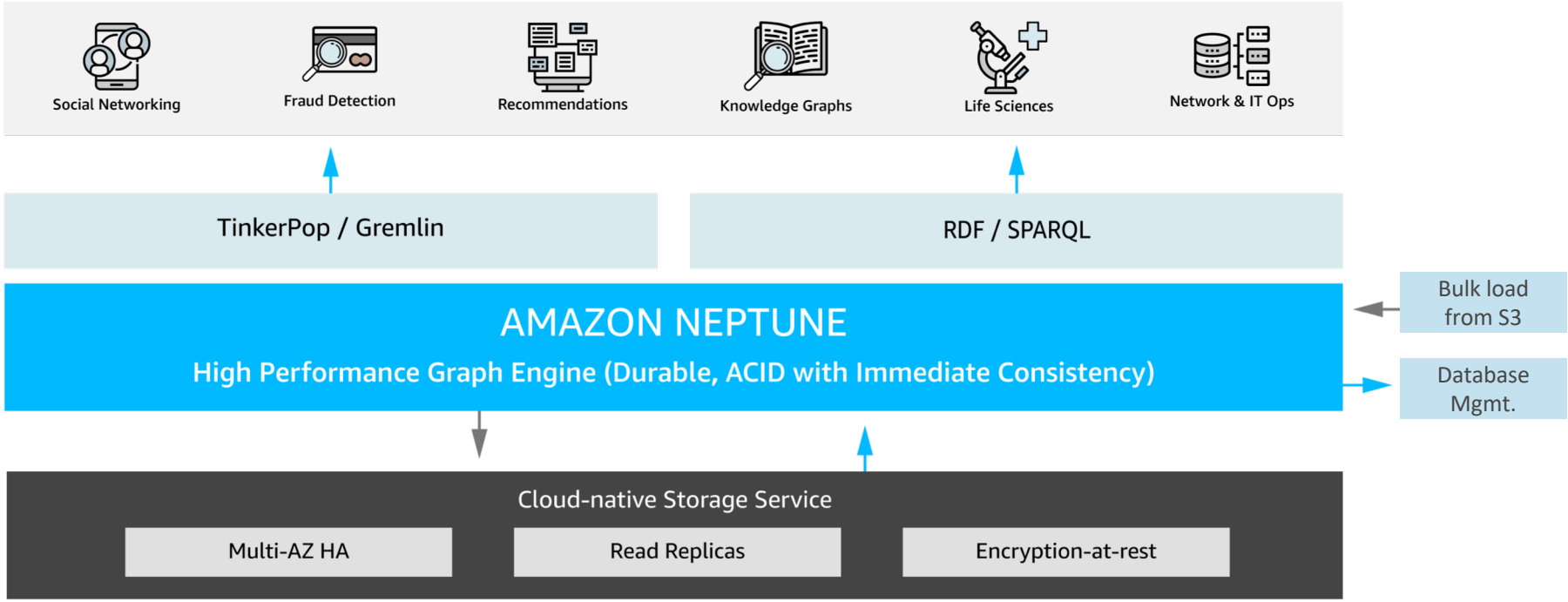
优点

- 通过控制台容易配置
- Multi-AZ高可用性、ACID
- 支持最多15个只读副本
- 支持数据加密存储
- 支持传输加密(TLS)
- 备份和恢复，时间点恢复

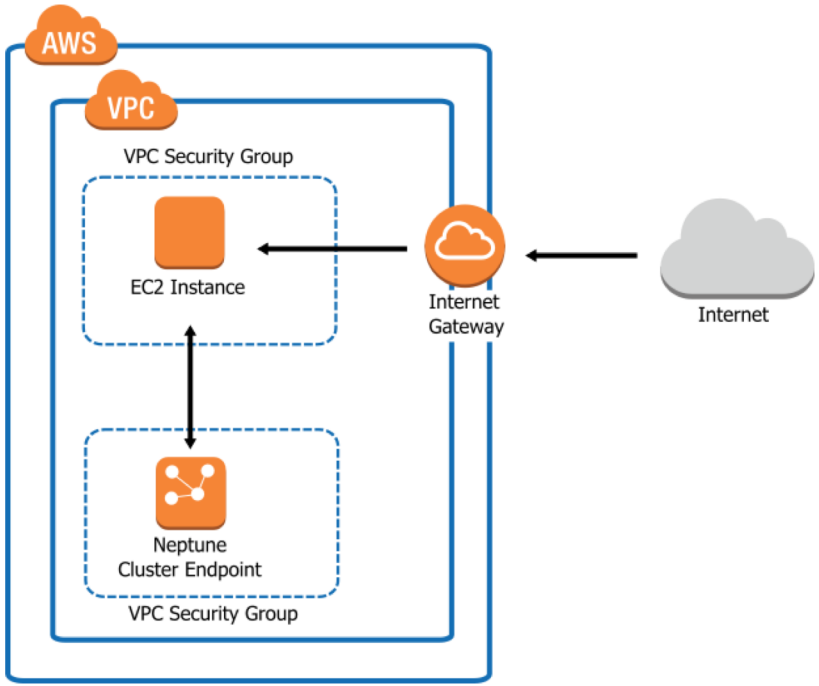




AMAZON NEPTUNE 架构



AMAZON NEPTUNE: VPC 部署



- 安全部署在VPC
- 通过在两个不同可用区(AZs)的两个子网中部署来增加可用性
- 集群组总是跨越三个AZ以提供持久性的存储
- 有关VPC安装细节，请参阅Amazon Neptune文档

Amazon Neptune 存储引擎概述

在3个可用区中数据被复制6份

数据持续备份到Amazon S3
(耐久性为11个9)

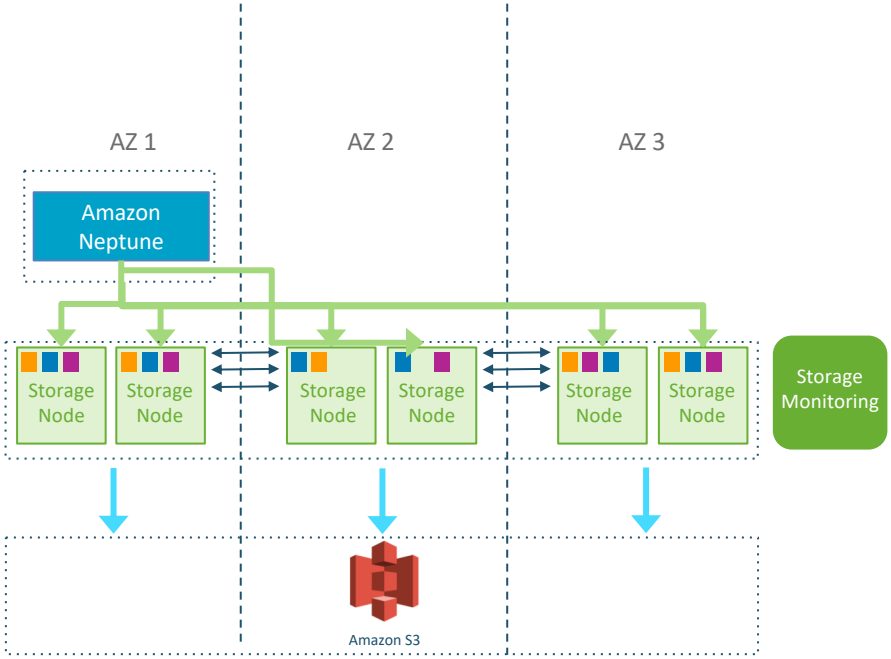
持续监控节点和磁盘的修复

10 GB段作为修复单位或平衡的热点

法定读/写系统的数量; 容忍延迟

仲裁成员资格的变化不会延迟写入

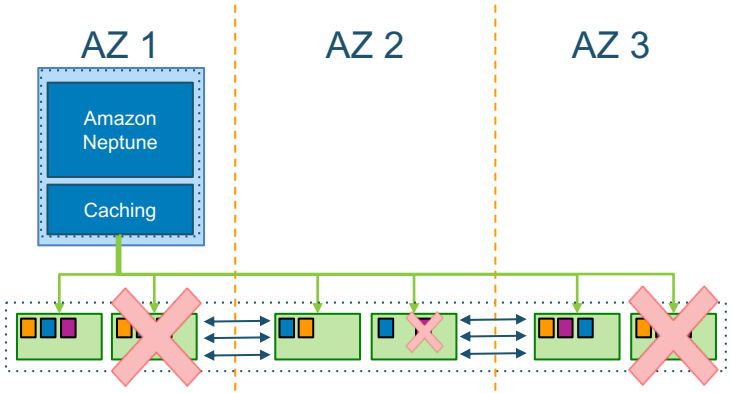
存储容量自动增长到64 TB



Amazon Neptune 高可用性和容错能力

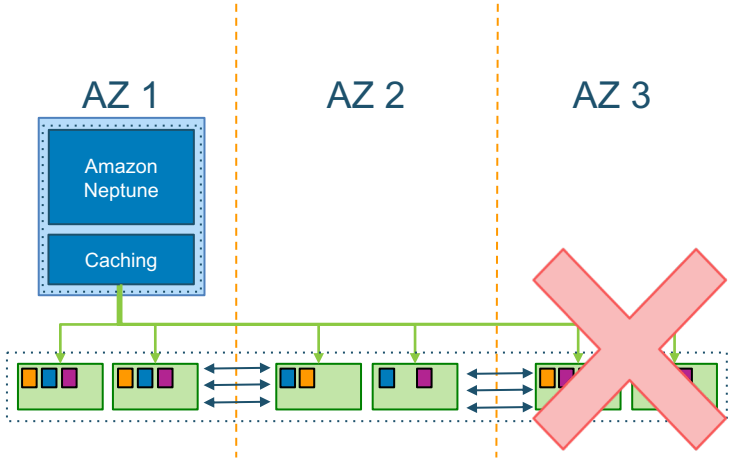
为何失效?

段故障(磁盘)
节点故障(机器)
可用区故障(网络或数据中心)



优化方法:

6个副本中4个写成功是必要的
6个副本中4个读成功是必要的
对等复制以修复故障

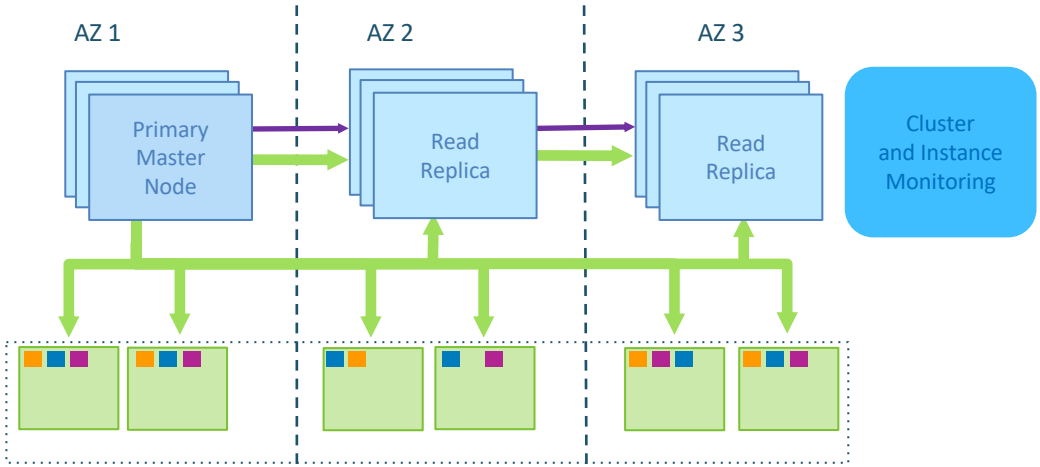


aws

Amazon Neptune 读复制

可用性

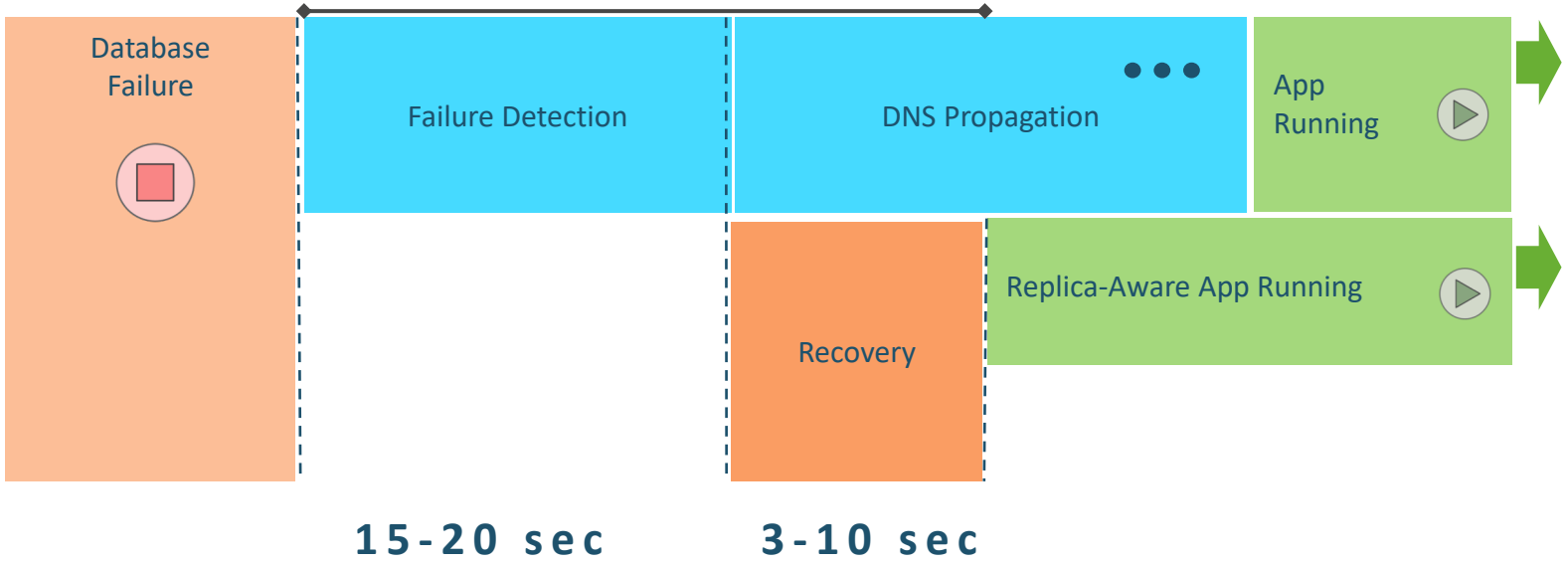
- 失败的数据库节点将被自动检测和替换
- 失败的数据库进程将被自动检测和回收
- 如果需要，副本将自动提升为主副本(故障转移)
- 客户指定的故障转移



性能

- 客户应用程序可以跨读副本扩展读流量
- 在读副本之间进行读操作的平衡

Amazon Neptune故障转移时间通常小于30秒



AMAZON NEPTUNE 的数据加载

Gremlin 加载数据格式
- 必须采用UTF-8格式编码
- CSV 格式遵循 RFC 4180 CSV 规范

RDF 加载数据格式
- 必须采用UTF-8格式编码
- 规范 (https://www.w3.org/TR/n-triples/) 中的 N -Triples (ntriples)
- 规范 (https://www.w3.org/TR/n-quads/) 中的 N-Quads (nquads)
- 规范 (https://www.w3.org/TR/rdf-syntax-grammar/) 中的 RDF/XML (rdfxml)
- 规范 (https://www.w3.org/TR/turtle/) 中的 N-Quads (turtle)

示例：加载数据

```
curl -X POST \
    -H 'Content-Type: application/json' \
    http://your-neptune-endpoint:8182/loader -d '
    {
      "source" : "s3://bucket-name/object-key-name",
      "format" : "format",
      "iamRoleArn" : "arn:aws:iam::account-id:role/role-name",
      "region" : "region",
      "failOnError" : "FALSE"
    }'
```

浏览全球税收政策的网络

THOMSON REUTERS

*“我们的客户越来越需要浏览复杂的全球税收政策和法规网络。我们需要一种方法来模拟我们最大客户的复杂企业结构，并提供端到端税收解决方案。我们的平台使用微服务体系结构，并开始利用**Amazon Neptune**作为基于图的系统，快速在数据中创建链接。”*

- Tim Vanderham, CTO , Thomson Reuters Tax & Accounting

aws

用户案例


THOMSON REUTERS


SIEMENS


FINRA


AstraZeneca


LIFEOMIC


amazon alexa


intuit


Pearson


SAMSUNG


Blackfynn

AMAZON NEPTUNE

全托管的图数据库

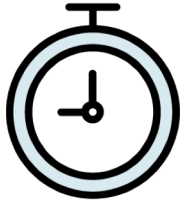
开放

支持 Apache TinkerPop & W3C RDF 图模型

快速

查询十亿级关系数据仅有毫秒级延迟

可靠



数据拥有跨3个可用区的6个副本，具有完全备份和恢复

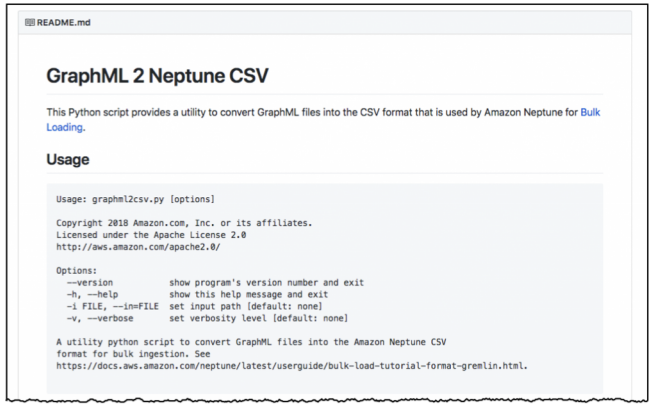
易用

使用Gremlin和SPARQL轻松构建强大的查询

aws

AMAZON NEPTUNE 的资源

- Amazon Neptune 服务介绍
- Amazon Neptune 文档
- Amazon Neptune Tools Repo
- Amazon Neptune Samples Repo

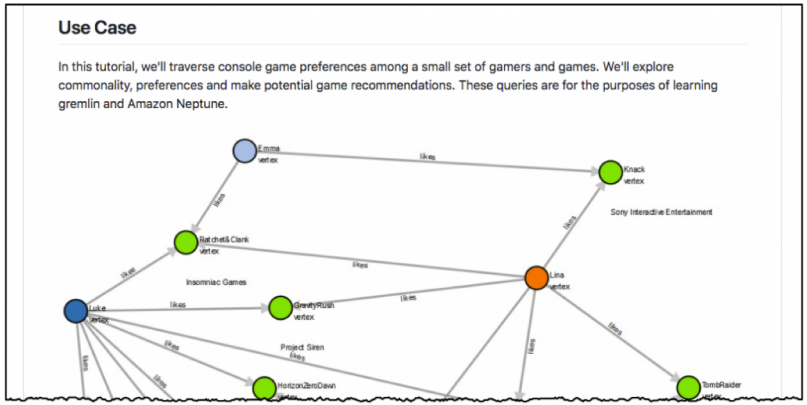

README.md

GraphML 2 Neptune CSV

This Python script provides a utility to convert GraphML files into the CSV format that is used by Amazon Neptune for Bulk Loading.

Usage

```
Usage: graphml2csv.py [options]

Copyright 2018 Amazon.com, Inc. or its affiliates.
Licensed under the Apache License 2.0
http://aws.amazon.com/apache2.0/

Options:
  --version           show program's version number and exit
  -h, --help          show this help message and exit
  -i FILE, --in=FILE  set input path [default: none]
  -v, --verbose       set verbosity level [default: none]

A utility python script to convert GraphML files into the Amazon Neptune CSV
format for bulk ingestion. See
https://docs.aws.amazon.com/neptune/latest/userguide/bulk-load-tutorial-format-gremlin.html.
```



感谢!