WEBINAR

How Getir built a comprehensive fraud detection system using Amazon Neptune and Amazon DynamoDB

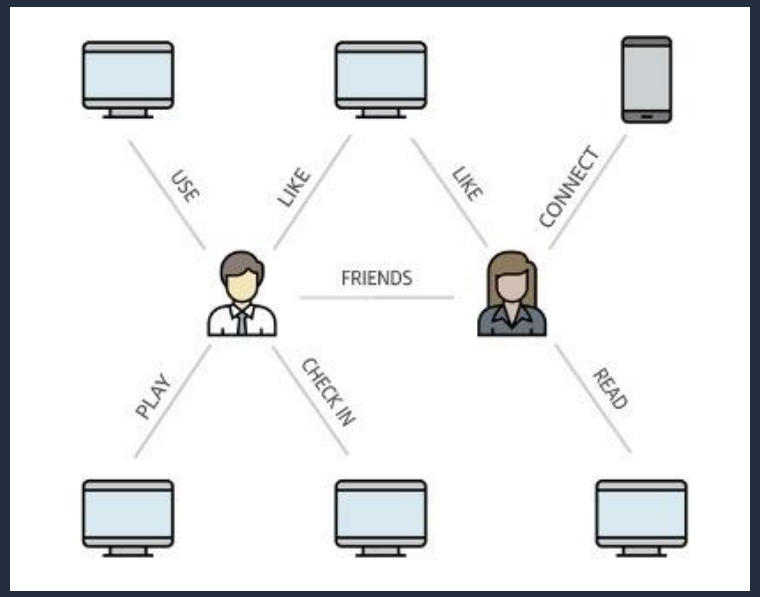
Berkay Berkman

Senior Data Engineer I
Getir

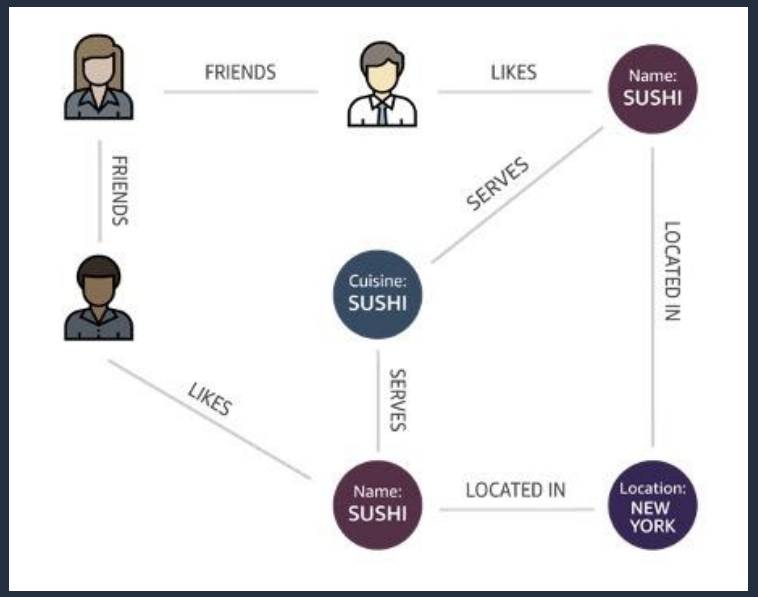
Esra Kayabali

Senior Solutions Architect
AWS

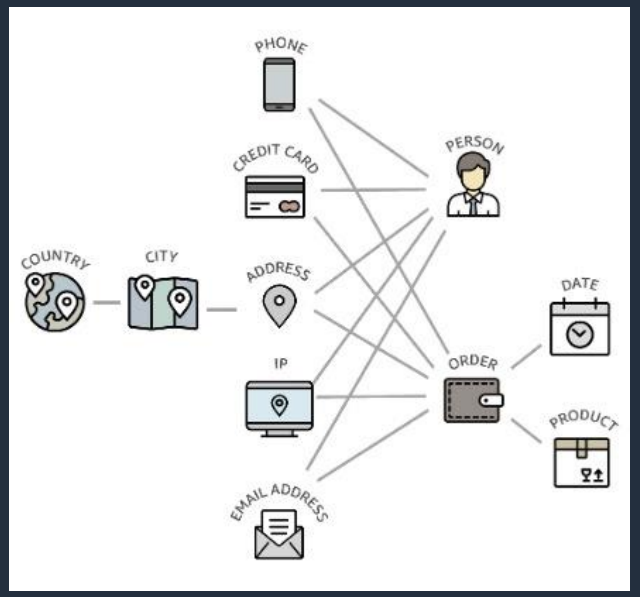
Highly connected data



Social Networks



Restaurant Recommendations



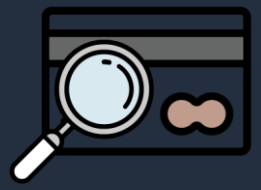
Retail Fraud Detection

Use cases for highly connected data

Social Networking

Recommendations

Knowledge Graphs



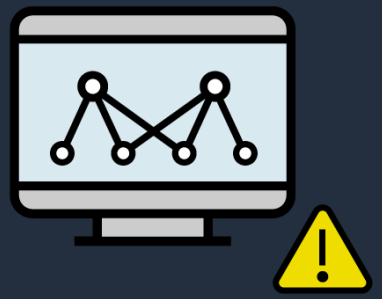
Fraud Detection

Life Sciences

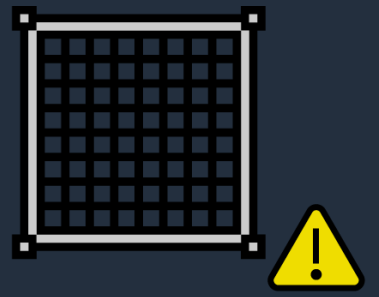
Network & IT Operations

Relational database challenges building apps with highly connected data

Unnatural for querying graph



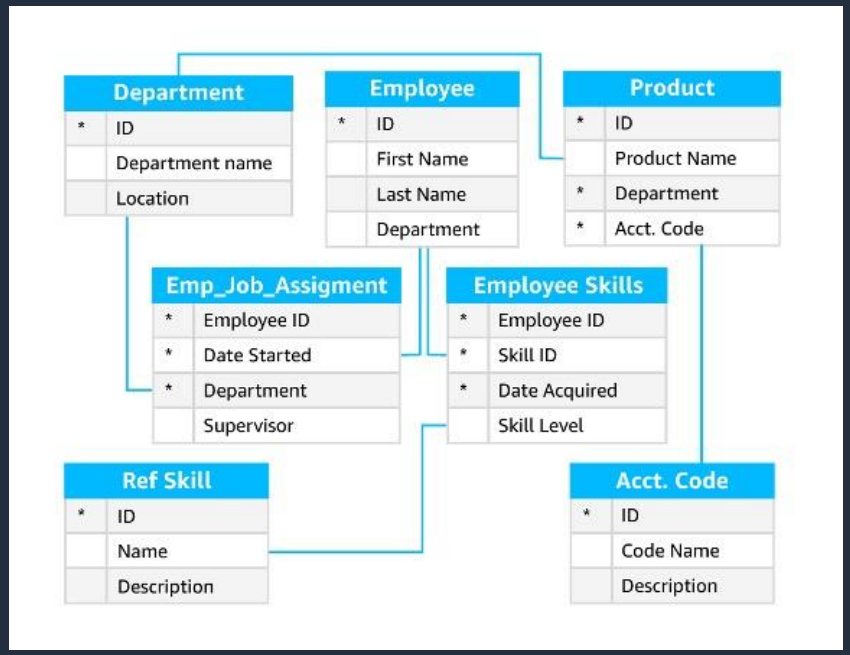
Inefficient graph processing



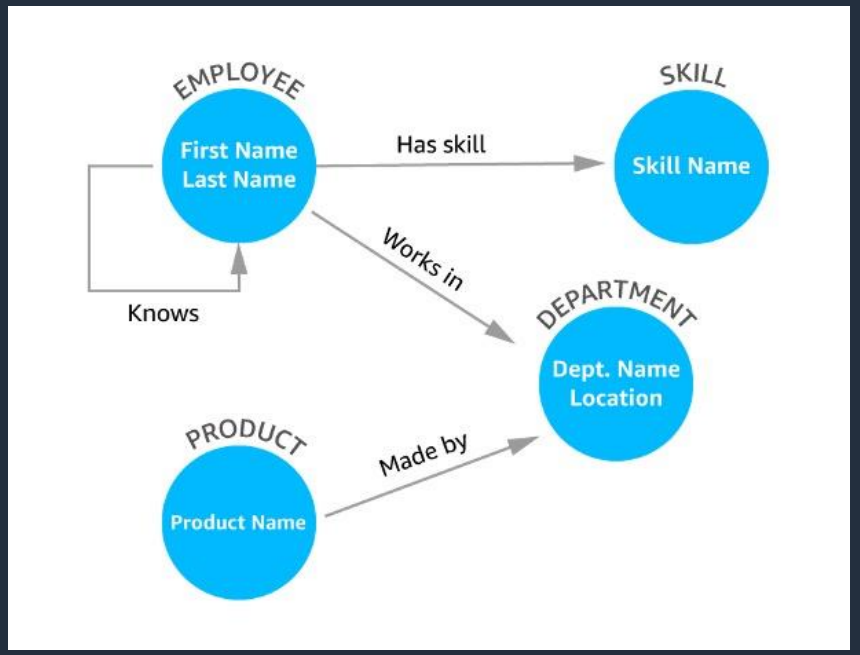
Rigid schema inflexible for changing data

Different approaches for highly connected data

Purpose-built for a business process



Purpose-built to answer questions about relationships



What is graph?

- Graph databases are purpose-built to store and navigate relationships
- Nodes represent real-world objects
- Edges store relationships between objects

Amazon Neptune

Amazon Neptune

Fast, reliable graph database built for the cloud

Supports Property Graph & W3C RDF graph models

Query billions of relationships with millisecond latency

6 replicas of your data across 3 AZs with full backup and restore

Build powerful queries easily with Gremlin and SPARQL

Useful when

Relationships matter as much as the data
Results depend on the *strength*, *weight,* or *quality* of relationships

---

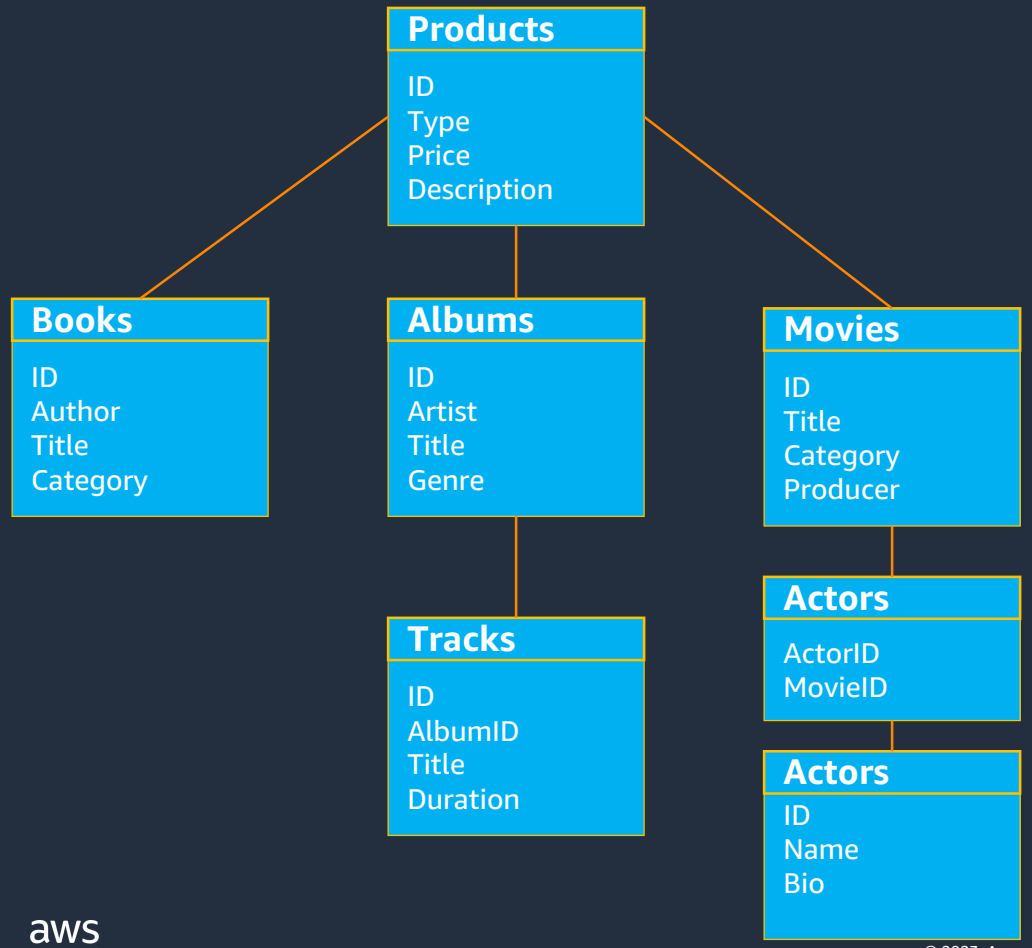
Amazon Neptune is ideal for

Social networking • Recommendations • Fraud detection • Life sciences

aws

Data Modeling: SQL vs. NoSQL

SQL



NoSQL

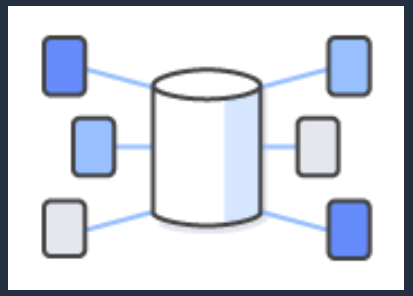
NoSQL design optimizes for compute instead of storage.

Amazon DynamoDB

Amazon DynamoDB

Fully managed

Consistently fast at any scale



Highly available and durable

Secure

Integrates with AWS Lambda, Amazon Redshift, and more

Cost-effective

Customer story

About Getir

- Pioneers of super fast delivery.
- Our founding idea: Groceries in 10 minutes, delivered to your door through our app.

- Founded in 2015, Getir means **"bring**" in Turkish
- Ultrafast delivery services, **anything at your door in 10'**

- All verticals – groceries, restaurants, water, car rental, jobs – offered through a **single Getir app**
- Available in **5 countries, in 3 continents**

- Super fast expansion, making it the **second-ever decacorn from Turkey** in only 6 years

Fraud Detection: Fighting Financial Crime

- Fraud Detection comes with several challenges.
- Correctly identifying changing fraud patterns in real time is very difficult
- Detecting and preventing fraud adds friction to the customer experience
- Increases in manual reviews of suspicious activity drives up staffing costs

Why Getir need a real-time fraud detection solution?

- Business teams were manually checking data daily. Meanwhile, the cost of fraud was very high.
- These manual controls became impossible as the use of the app increased.
- The request from the Audit Team is to identify these users based on rules using analytical methods.

Why Getir chose Amazon Neptune?

- Graph databases use nodes to store data entities, and edges to store relationships between entities. And this storage gives

- An edge always has a start node, end node, type, and direction, and an edge can describe parent-child relationships, actions, ownership, and the like.

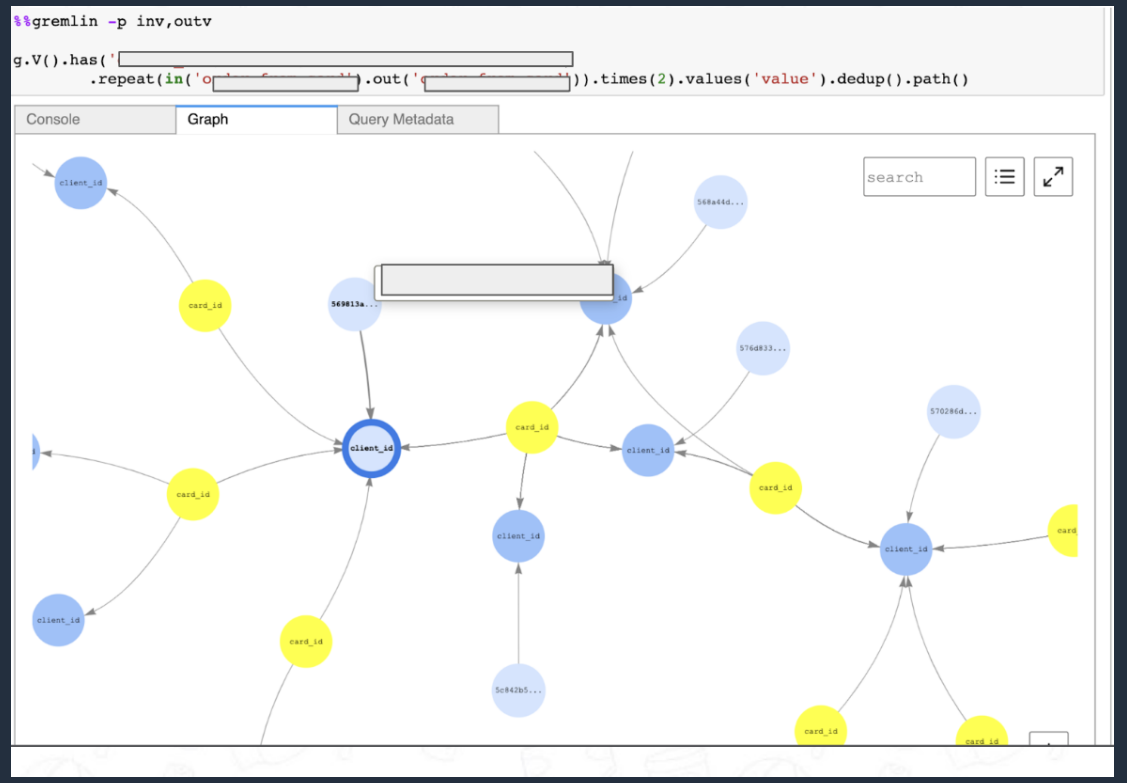
- There is no limit to the number and kind of relationships a node can have.

- Amazon Neptune provides us with a much more effective environment for detecting fraud circles compared to Relational Databases. With SQL, we get our result in minutes, but using Amazon Neptune with Gremlin Graph Query Language, we get our answer in milliseconds.

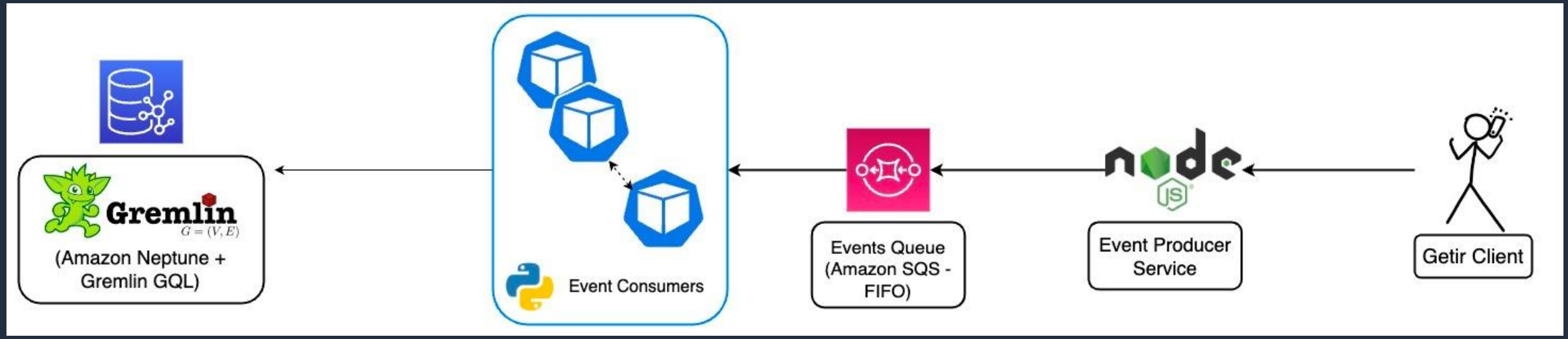
Why Getir chose Amazon Neptune?

- Amazon Neptune is a fully managed database service built for the cloud that makes it easier to build and run graph applications.

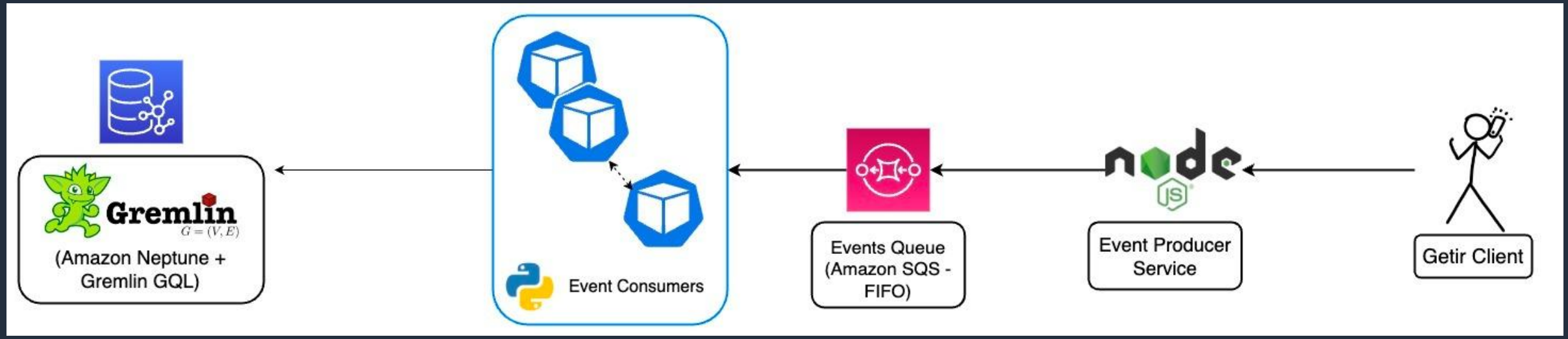
- Neptune provides built-in security, continuous backups, serverless compute, and integrations with other AWS services.



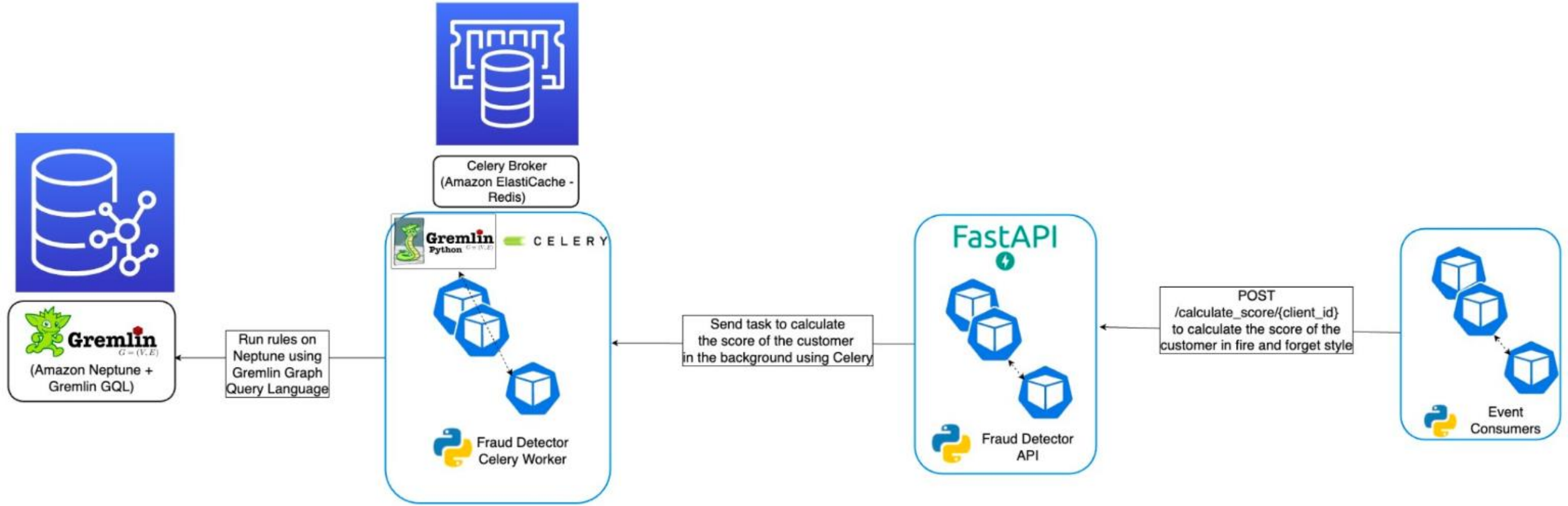
Getir’s fraud detection solution



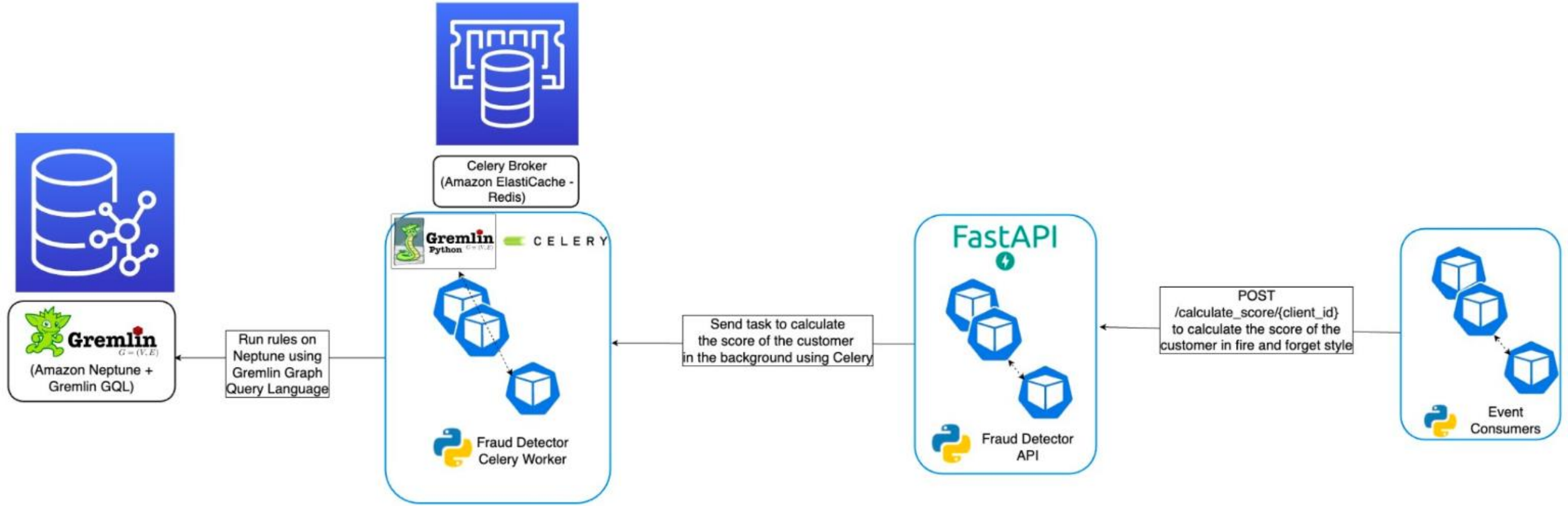
Getir’s fraud detection solution



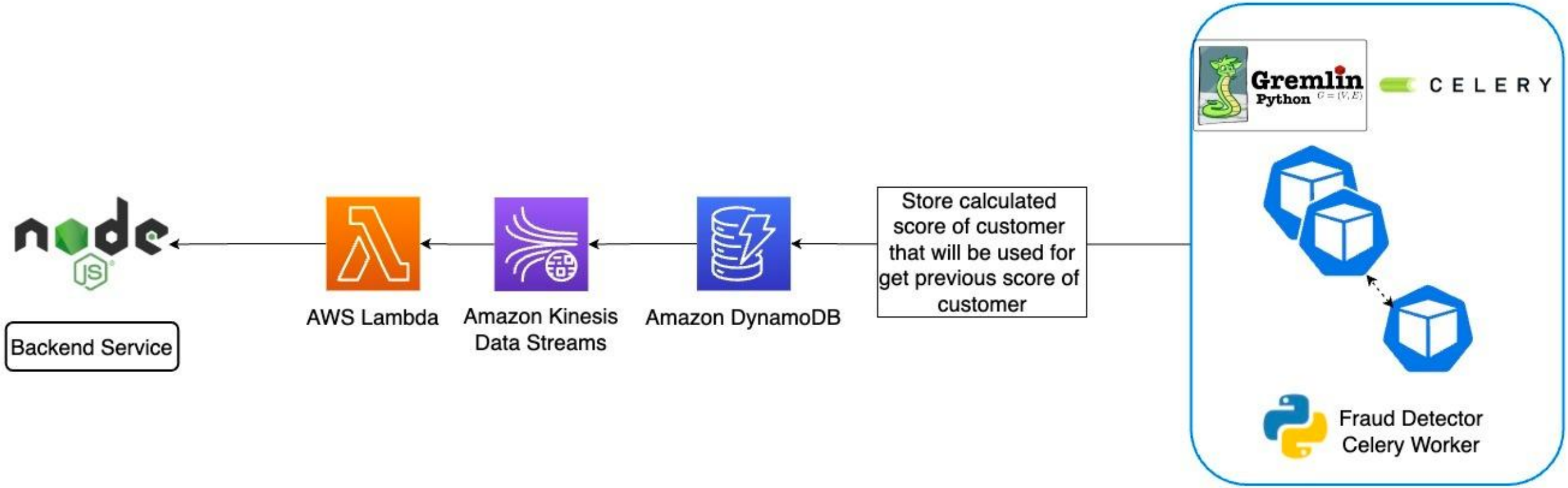
Getir's fraud detection solution



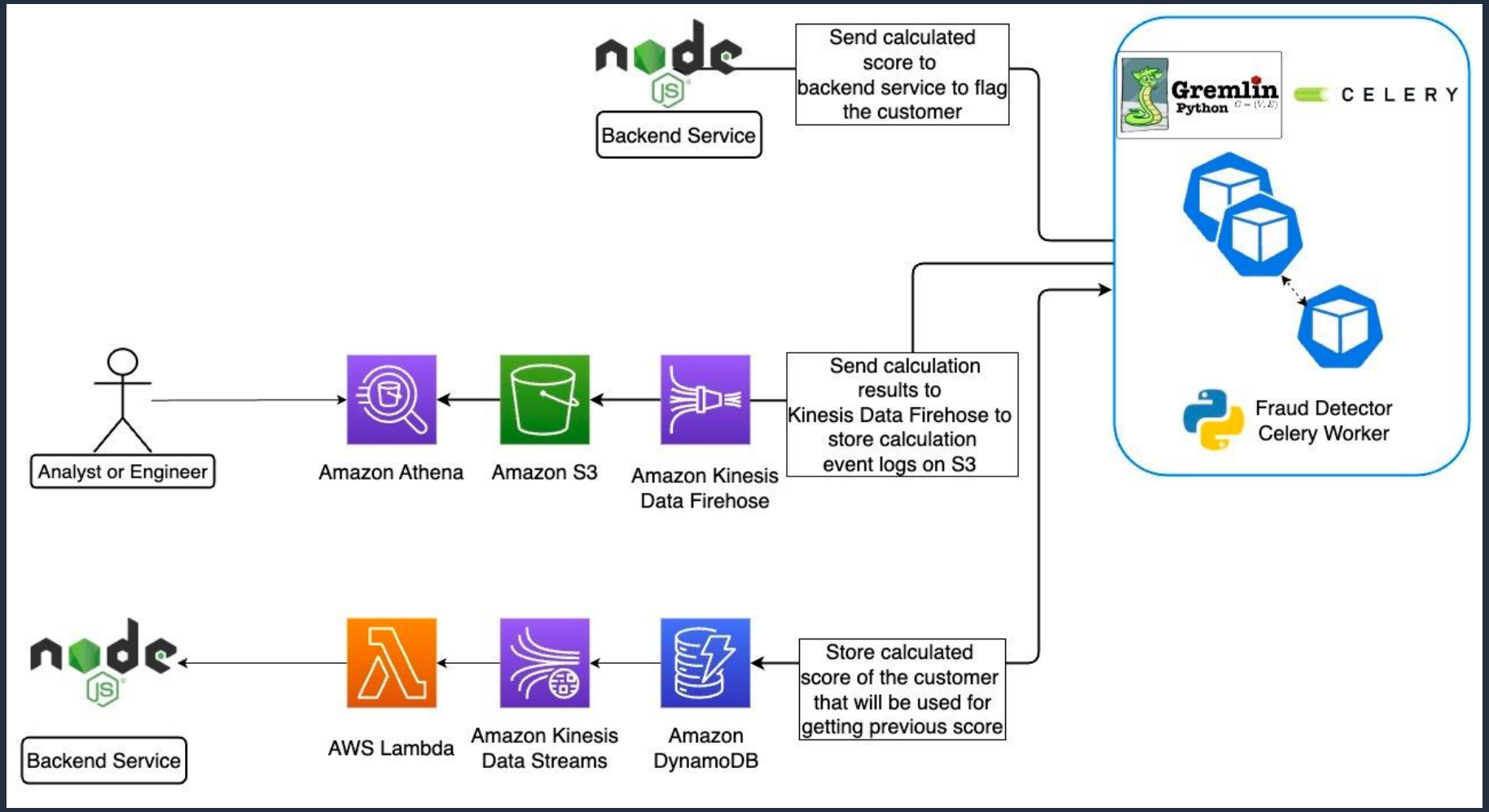
Getir's fraud detection solution



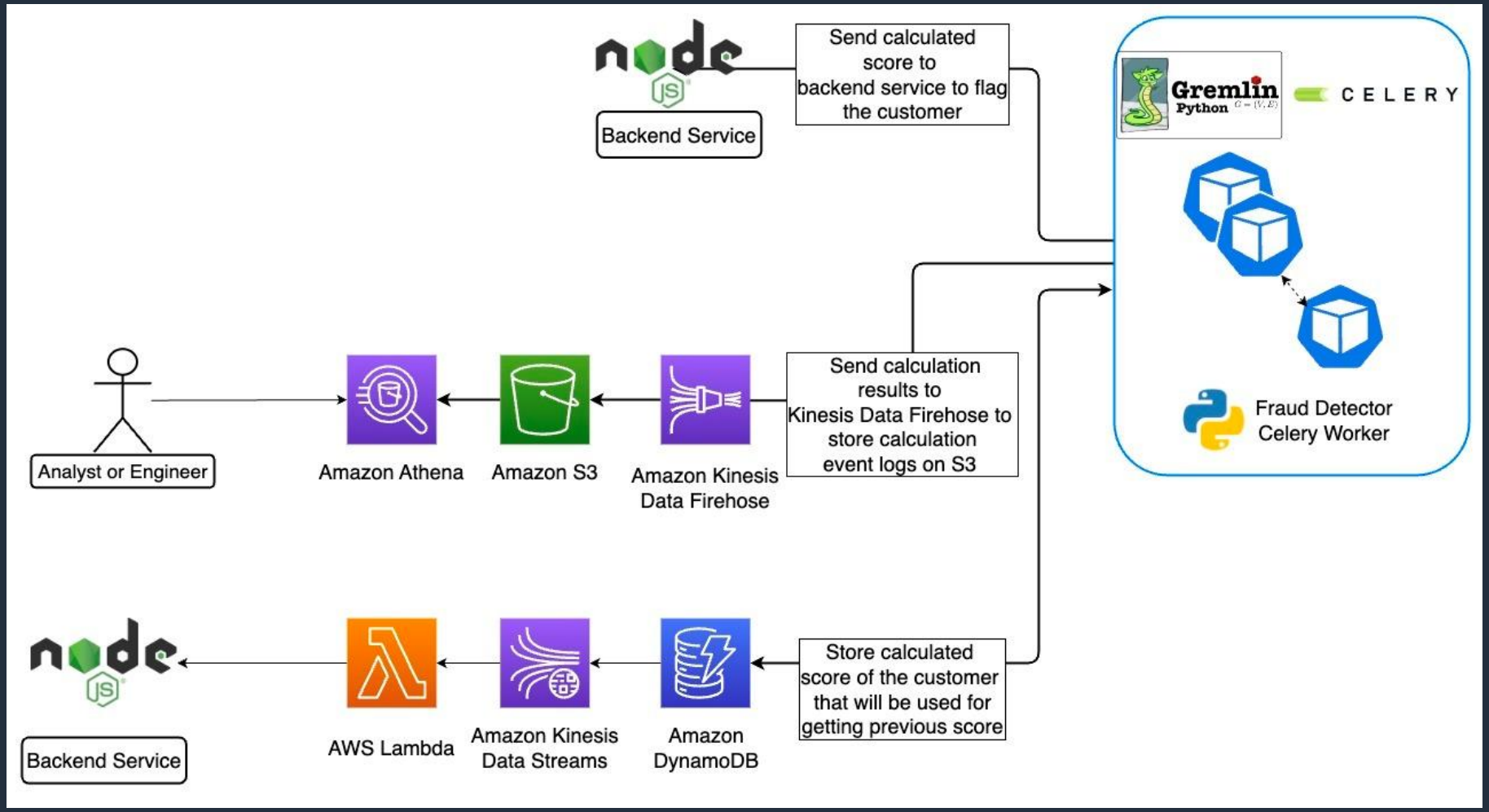
Getir’s fraud detection solution



Getir's fraud detection solution

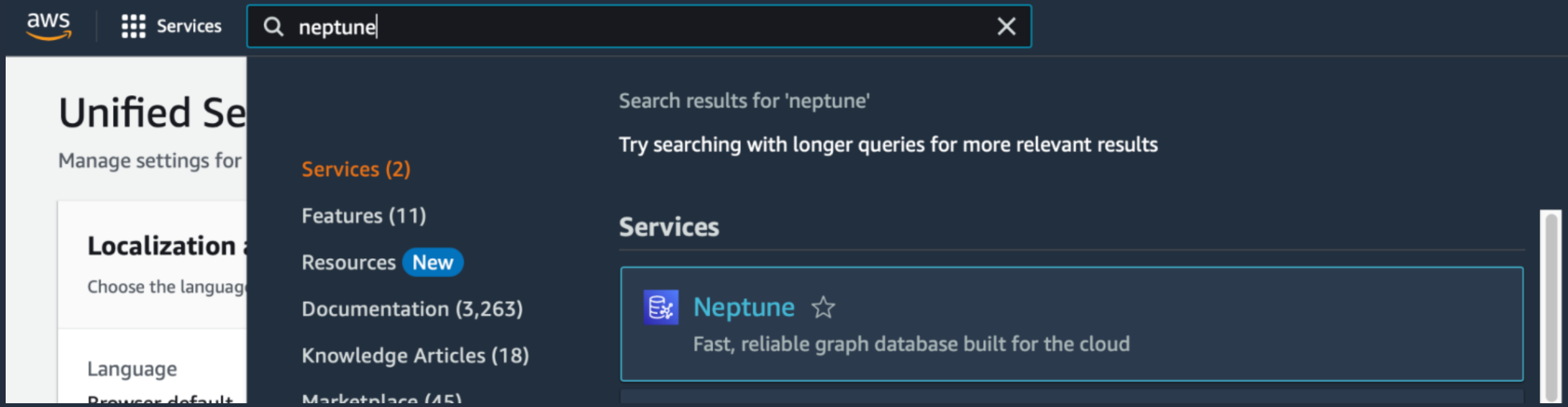


Getir's fraud detection solution

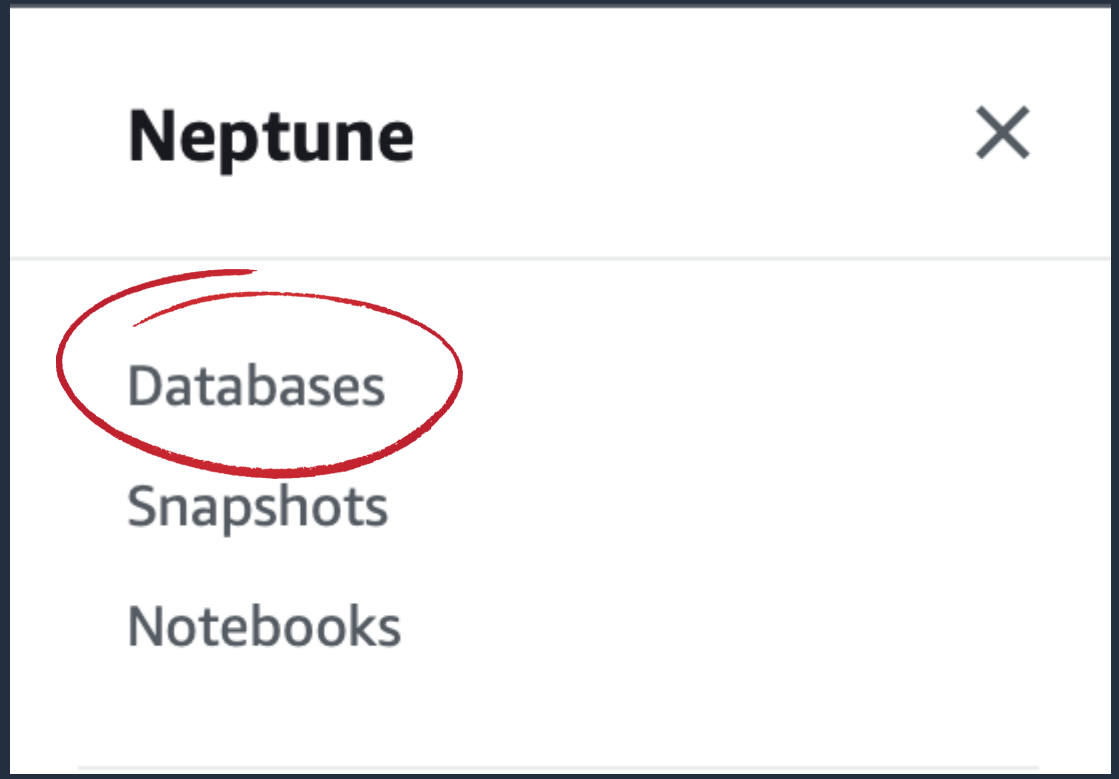


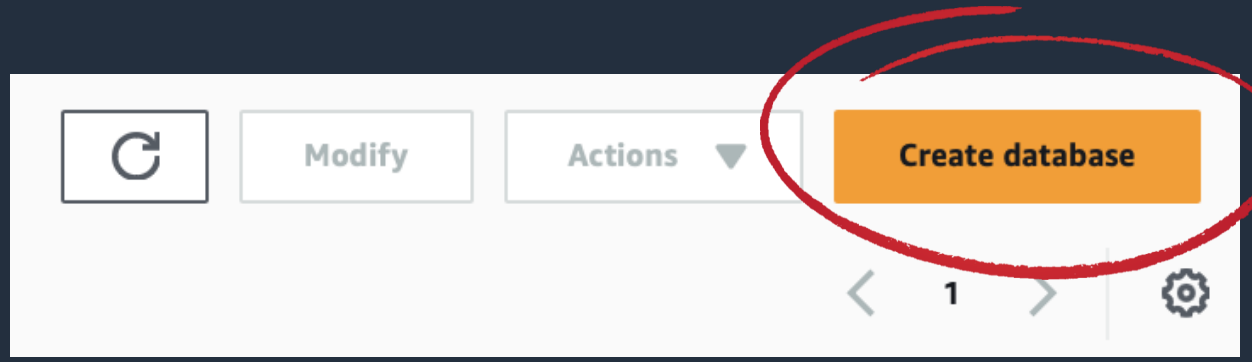
Demo

Demo

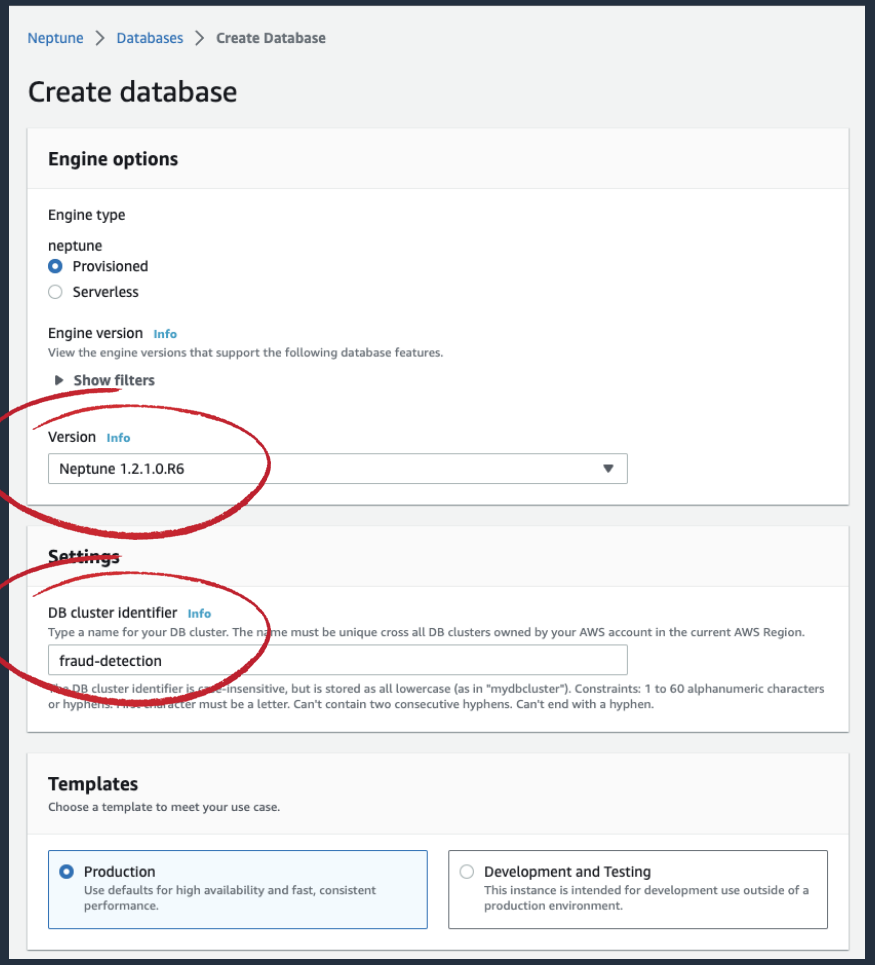


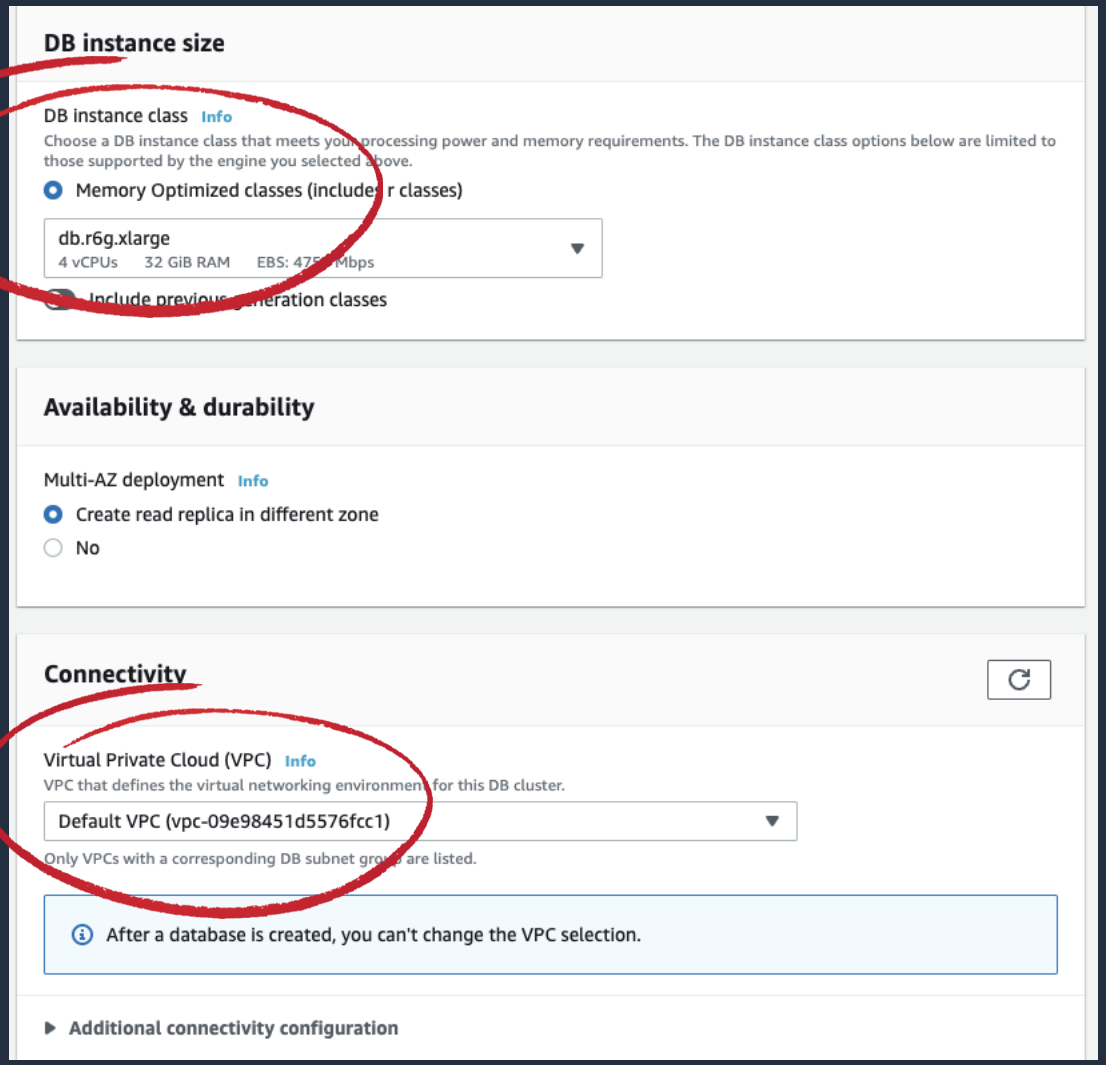
Demo



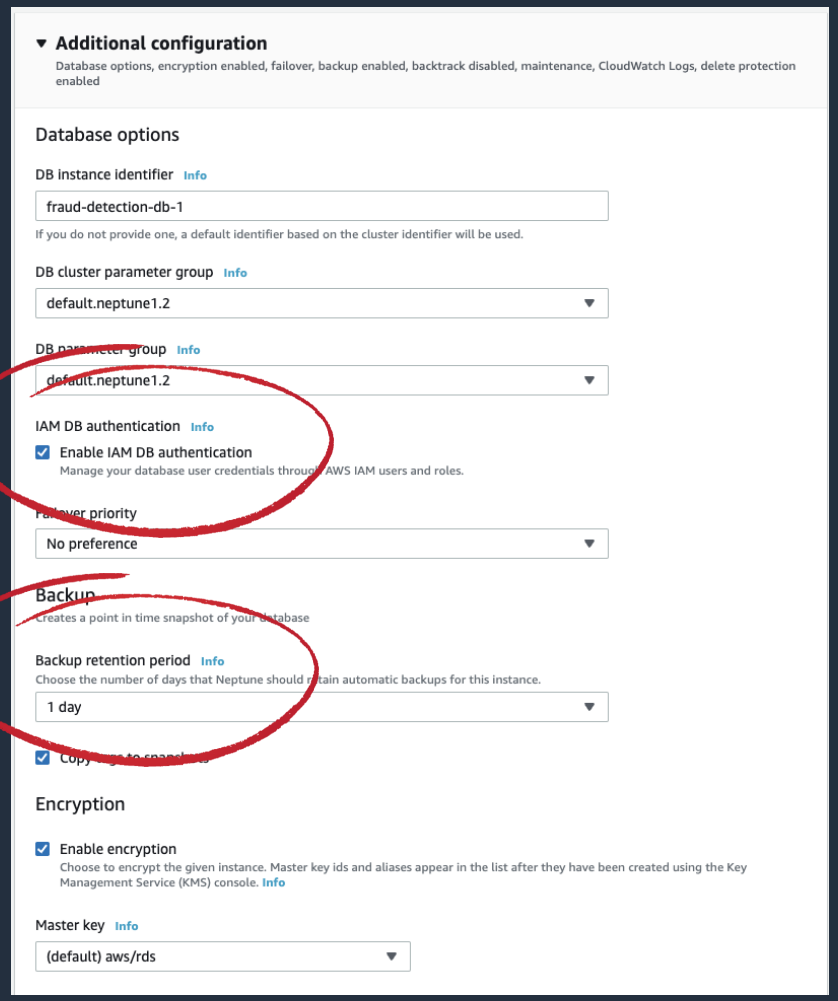


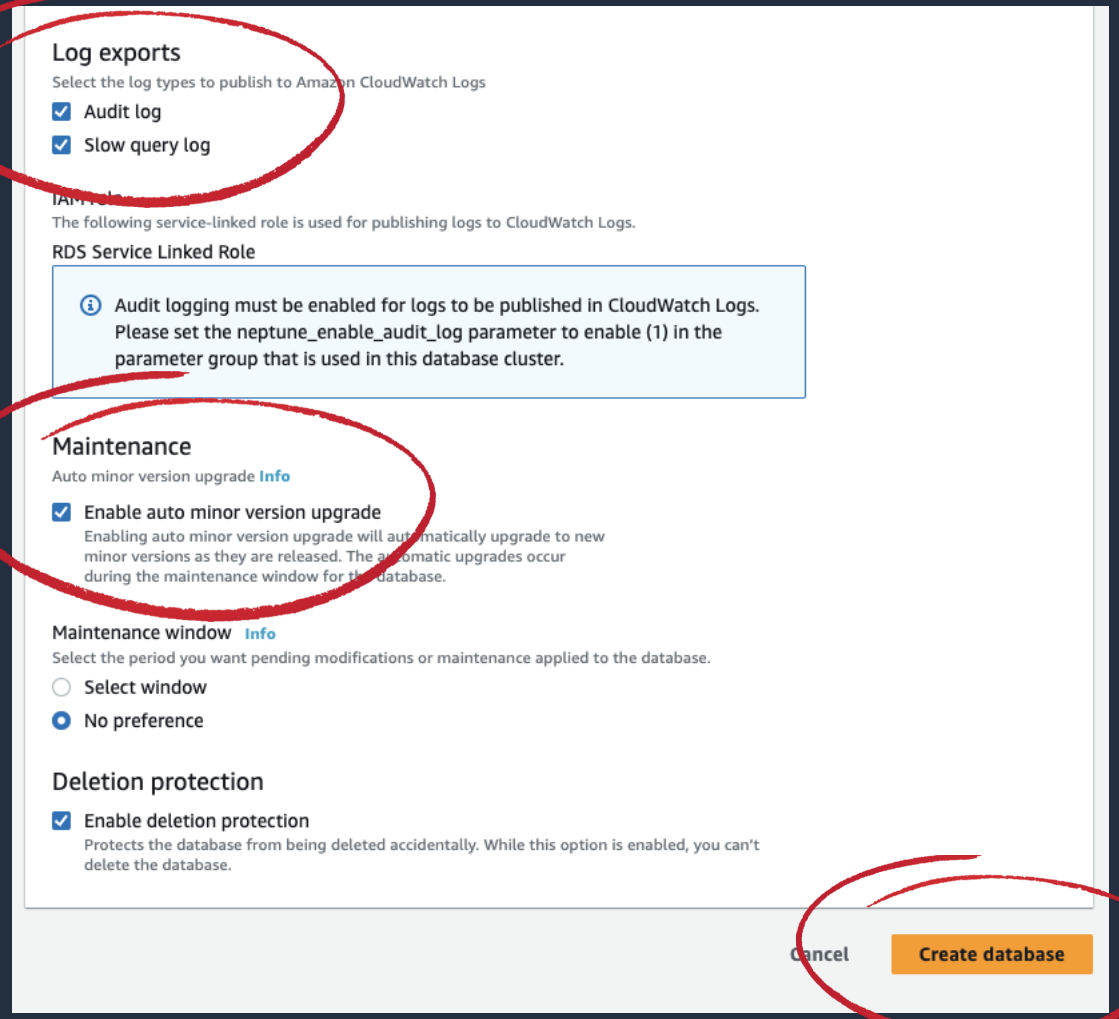
Demo



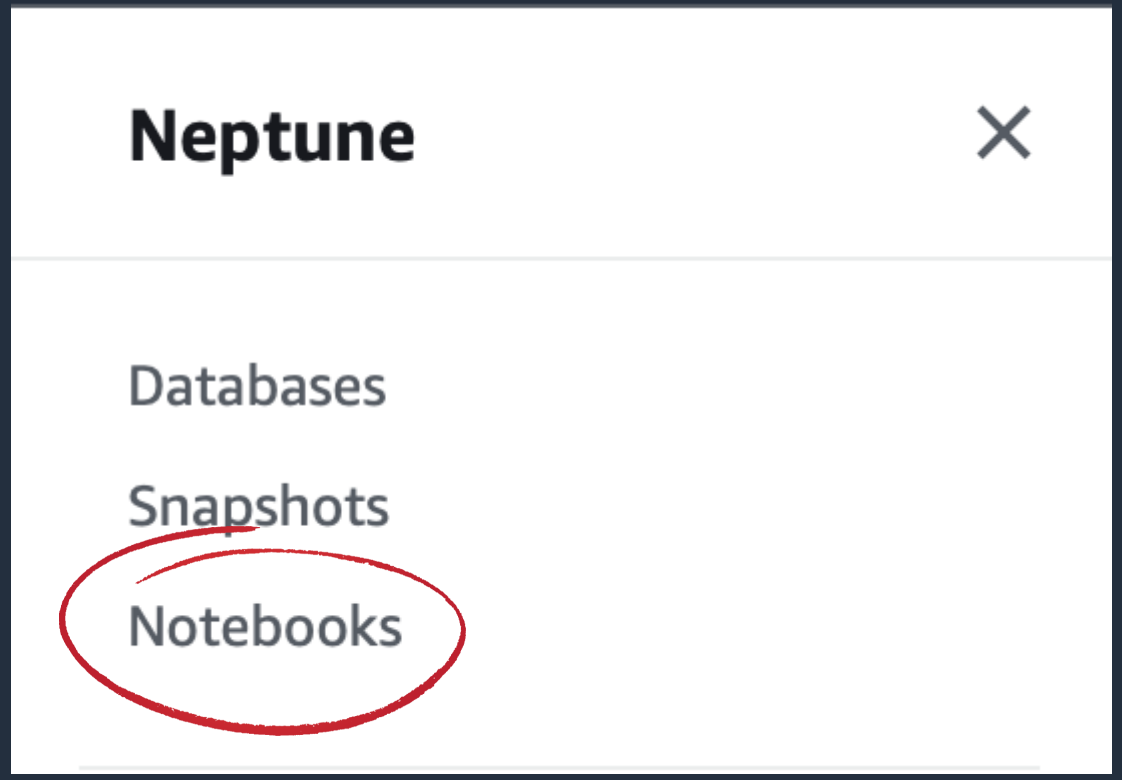


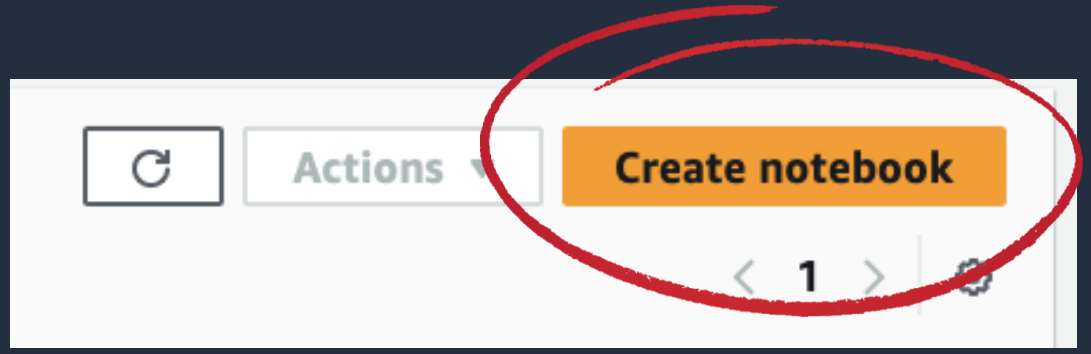
Demo



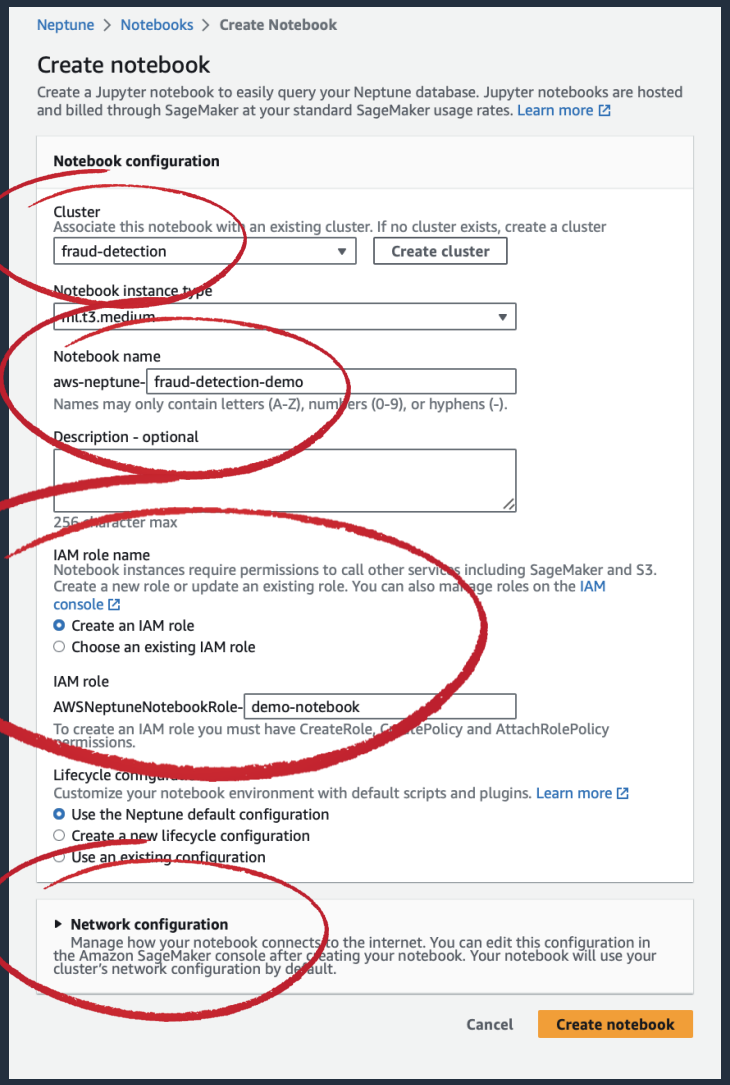


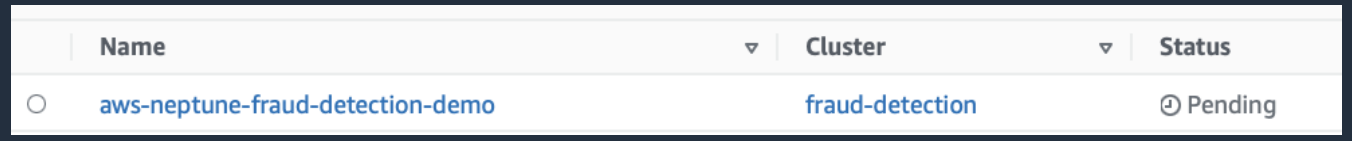
Demo

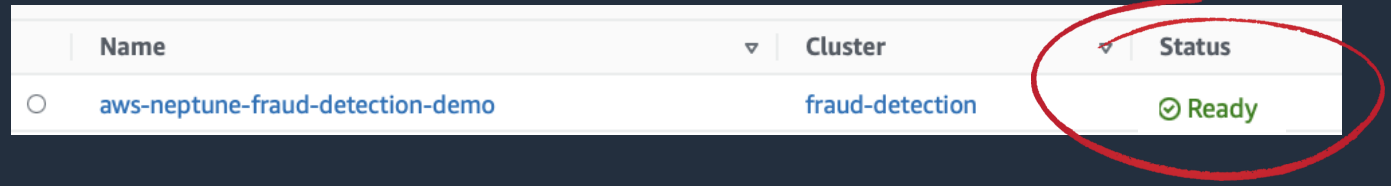


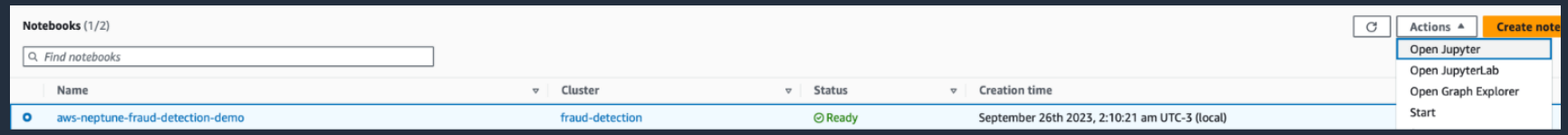


Demo

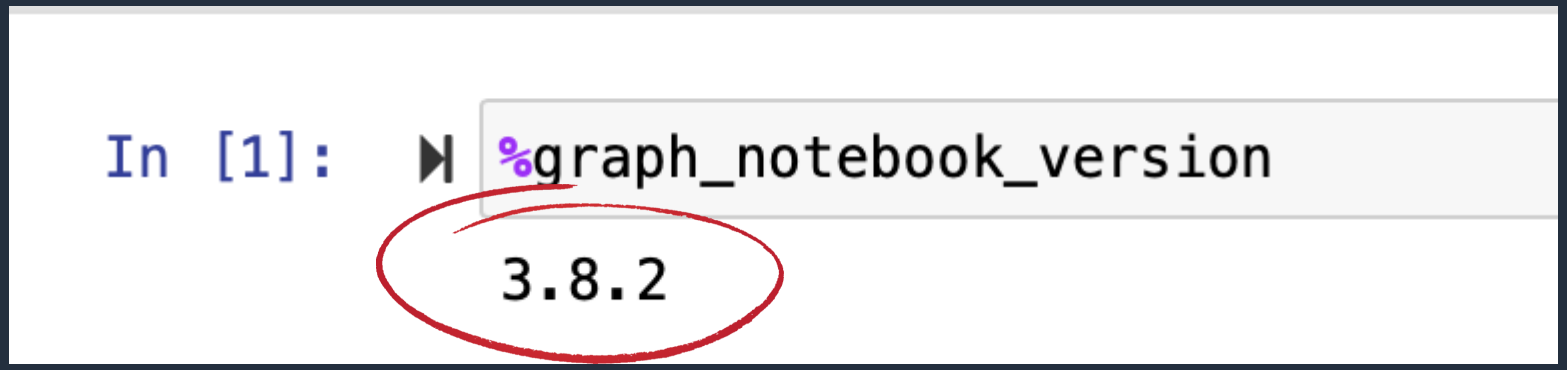




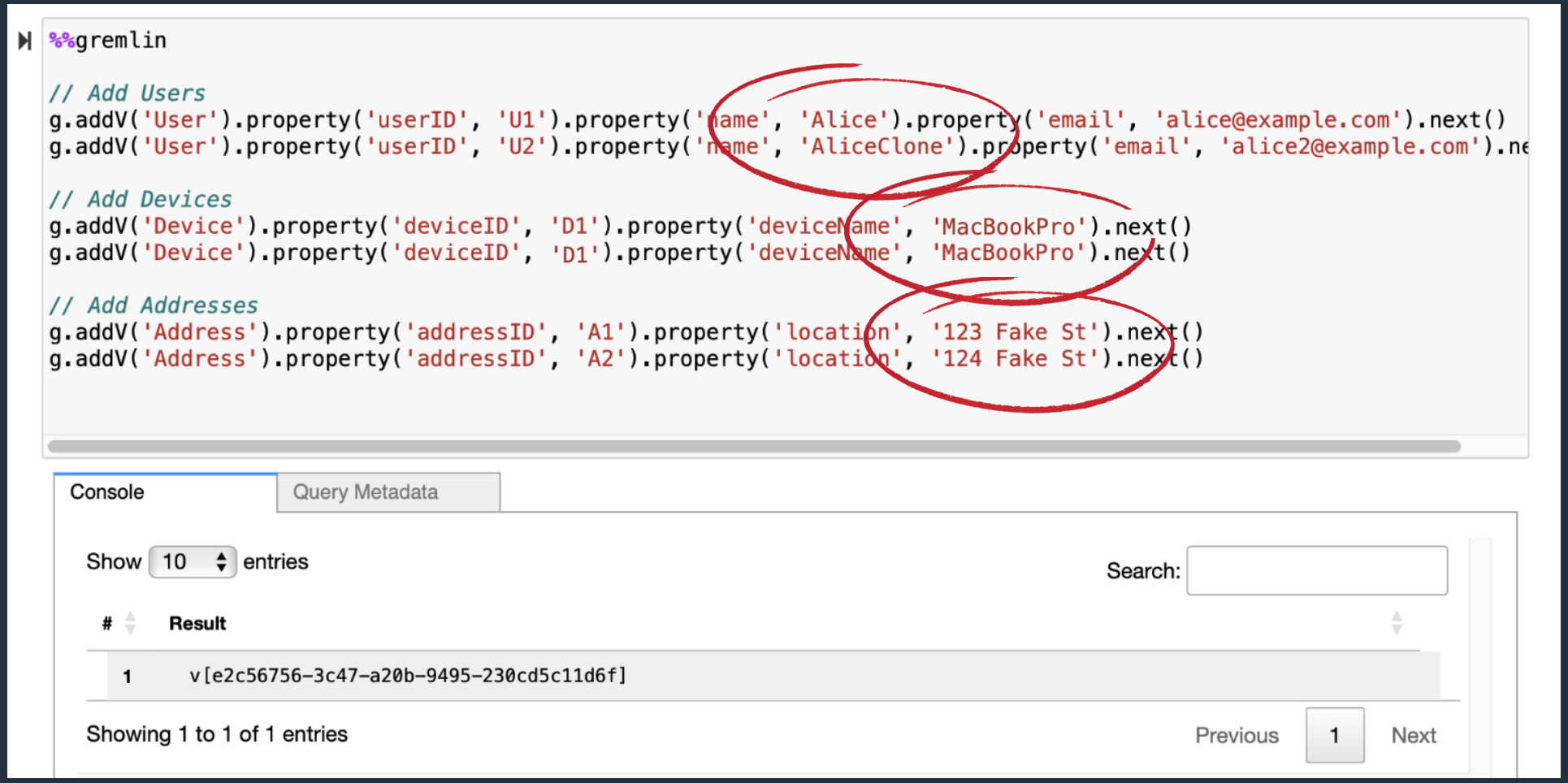




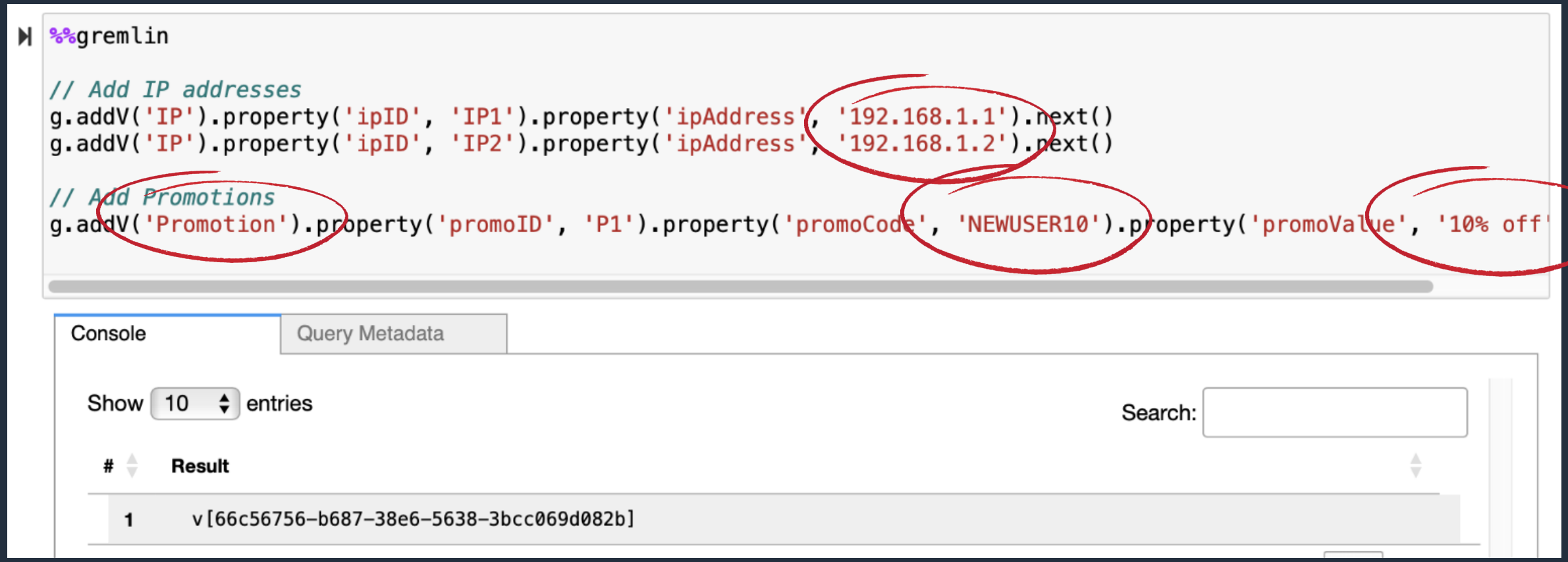
Demo



Demo



Demo



Demo

```
%%gremlin


// Link Users to Devices, Addresses, IPs, and Promotions
g.V().has('User', 'userID', 'U1').addE('USED_DEVICE').to(g.V().has('Device', 'deviceID', 'D1')).next()
g.V().has('User', 'userID', 'U2').addE('USED_DEVICE').to(g.V().has('Device', 'deviceID', 'D1')).next()

g.V().has('User', 'userID', 'U1').addE('HAS_ADDRESS').to(g.V().has('Address', 'addressID', 'A1')).next()
g.V().has('User', 'userID', 'U2').addE('HAS_ADDRESS').to(g.V().has('Address', 'addressID', 'A2')).next()

g.V().has('User', 'userID', 'U1').addE('USED_IP').to(g.V().has('IP', 'ipID', 'IP1')).next()
g.V().has('User', 'userID', 'U2').addE('USED_IP').to(g.V().has('IP', 'ipID', 'IP1')).next()

g.V().has('User', 'userID', 'U1').addE('CLAIMED_PROMOTION').to(g.V().has('Promotion', 'promoID', 'P1')).next()
g.V().has('User', 'userID', 'U2').addE('CLAIMED_PROMOTION').to(g.V().has('Promotion', 'promoID', 'P1')).next()
```

Console | Query Metadata

Show 10 entries Search:

#	Result
1	e[14c56757-b7c4-6260-f08f-58a4bd6e2752][1ec56756-3c29-b17d-fe20-ba8a0504e655-CLAIMED_PROMOTION->66c56756-b687-38e6-

Demo

```
%%gremlin


// Link Users to Devices, Addresses, IPs, and Promotions
g.V().has('User', 'userID', 'U1').addE('USED_DEVICE').to(g.V().has('Device', 'deviceID', 'D1')).next()
g.V().has('User', 'userID', 'U2').addE('USED_DEVICE').to(g.V().has('Device', 'deviceID', 'D1')).next()

g.V().has('User', 'userID', 'U1').addE('HAS_ADDRESS').to(g.V().has('Address', 'addressID', 'A1')).next()
g.V().has('User', 'userID', 'U2').addE('HAS_ADDRESS').to(g.V().has('Address', 'addressID', 'A2')).next()

g.V().has('User', 'userID', 'U1').addE('USED_IP').to(g.V().has('IP', 'ipID', 'IP1')).next()
g.V().has('User', 'userID', 'U2').addE('USED_IP').to(g.V().has('IP', 'ipID', 'IP1')).next()

g.V().has('User', 'userID', 'U1').addE('CLAIMED_PROMOTION').to(g.V().has('Promotion', 'promoID', 'P1')).next()
g.V().has('User', 'userID', 'U2').addE('CLAIMED_PROMOTION').to(g.V().has('Promotion', 'promoID', 'P1')).next()
```

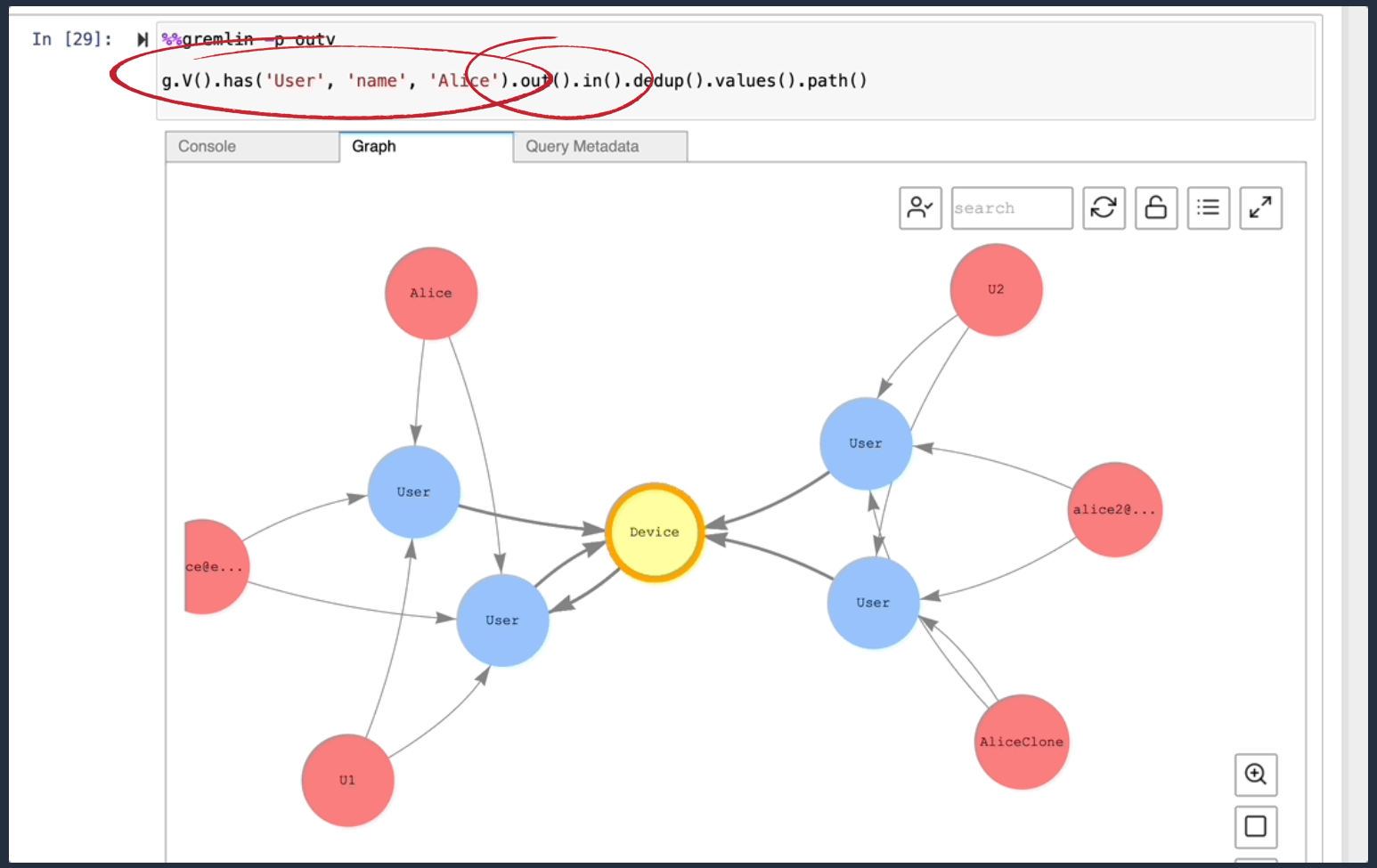
Console | Query Metadata

Show 10 entries

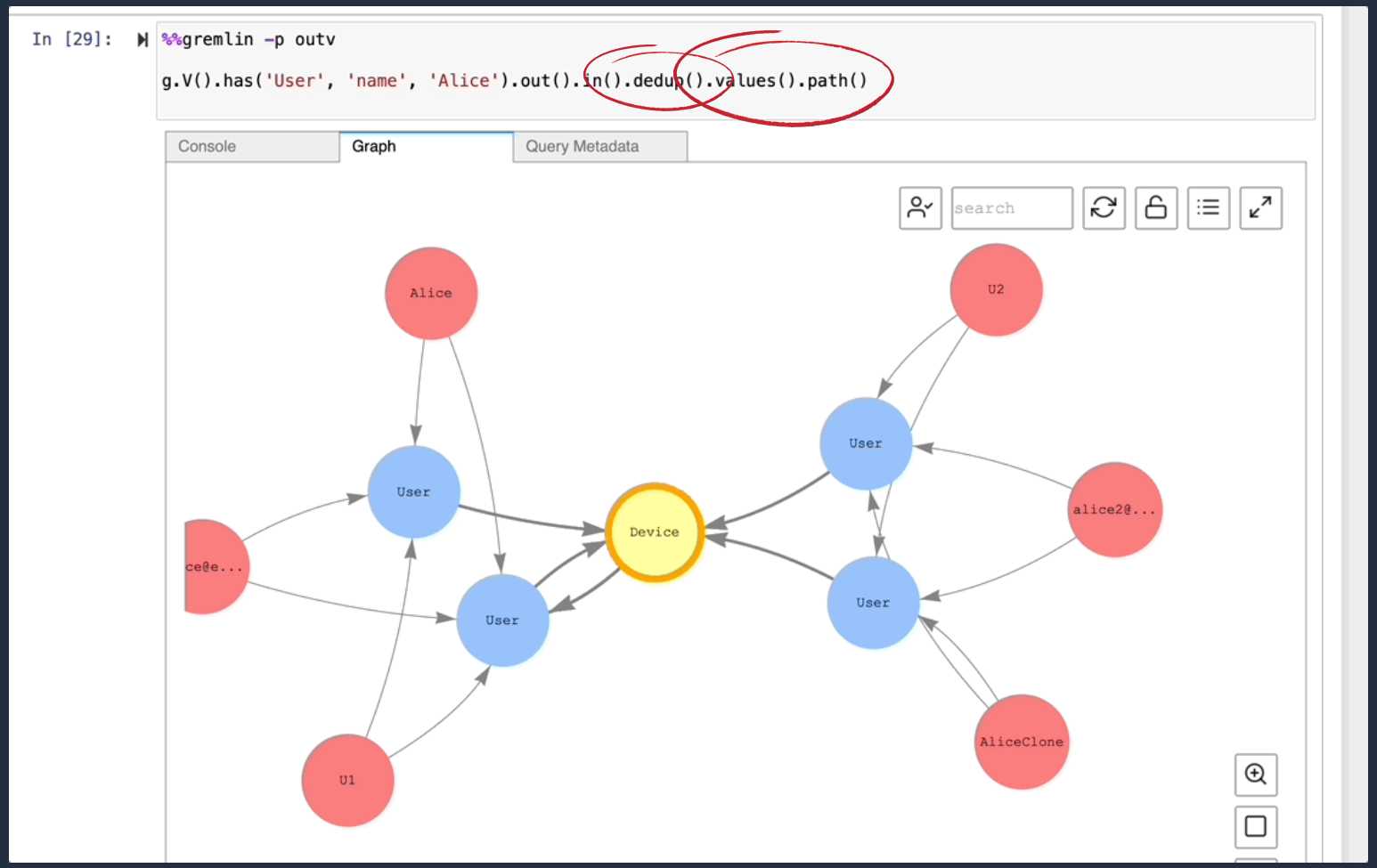
Search:

#	Result
1	e[14c56757-b7c4-6260-f08f-58a4bd6e2752][1ec56756-3c29-b17d-fe20-ba8a0504e655-CLAIMED_PROMOTION->66c56756-b687-38e6-

Demo



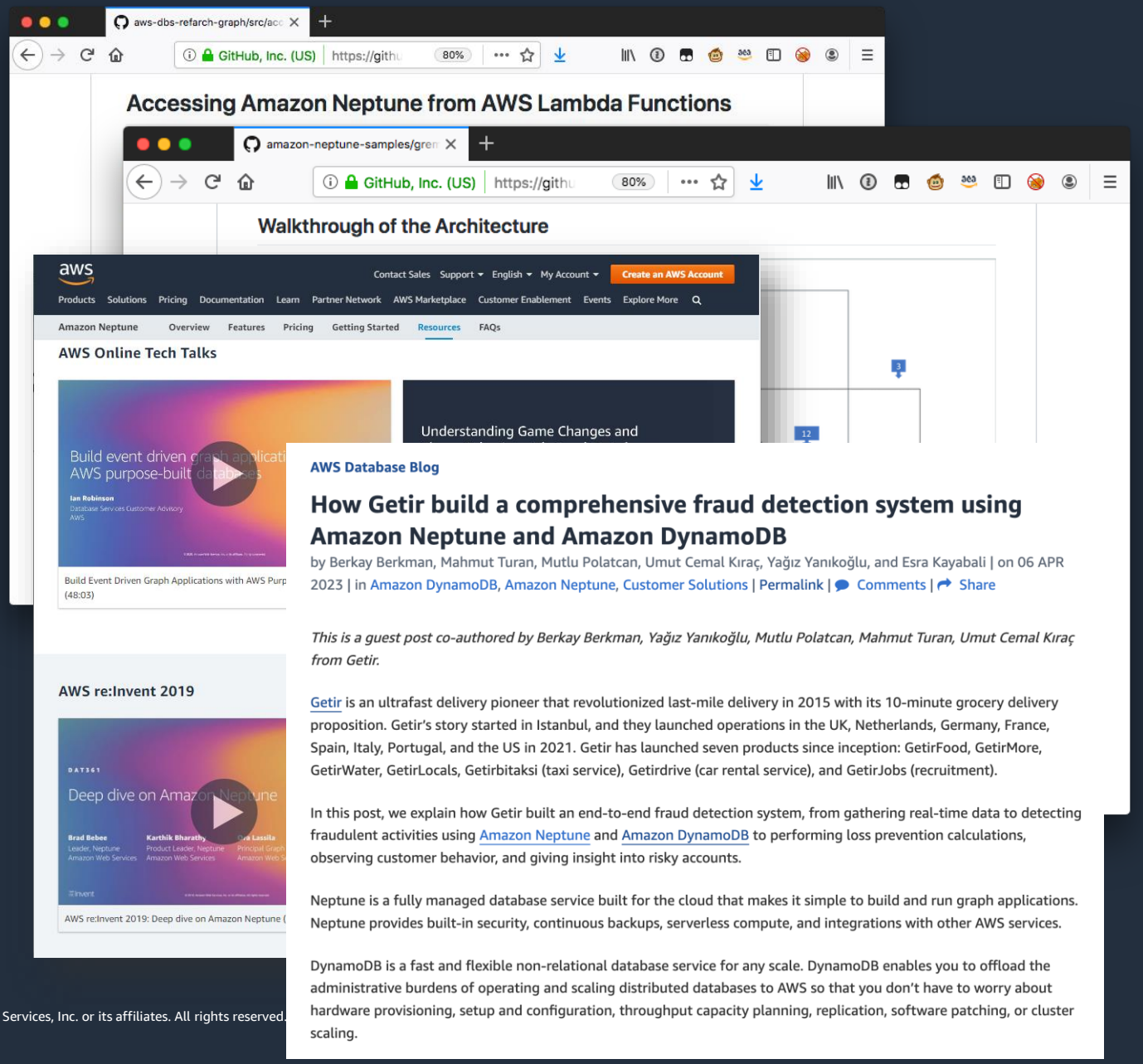
Demo



What is next?

Additional resources

- Neptune notebooks/graph notebook
- Neptune reference architectures
- Use cases, videos, blogs, code . . .
- Customer success stories
- How Getir build a comprehensive fraud detection system using Amazon Neptune and Amazon DynamoDB



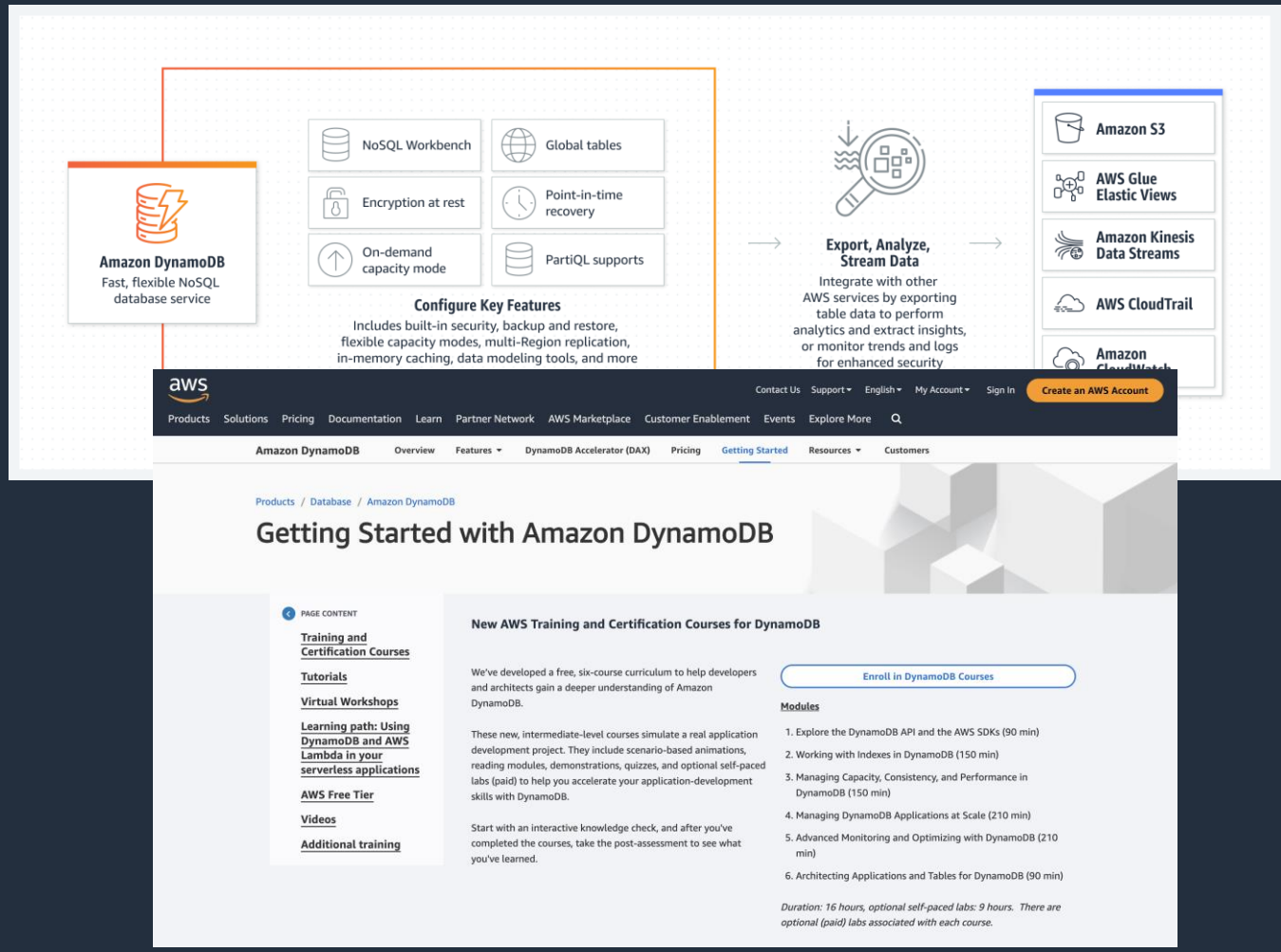
Additional resources

Amazon DynamoDB

Getting Started

Amazon DynamoDB resources

Customer success stories



Thank you!

Berkay Berkman
berkay.berkman@getir.com

Esra Kayabali
kayabali@amazon.com