AWS re:Invent

DAT347-R

Neptune best practices: How to optimize your graph queries

Michael Schmidt

Sr. Software Development Engineer

Amazon Web Services

AWS re:Invent

Agenda

Amazon Neptune introduction, architecture, and monitoring

Query optimization and performance tuning

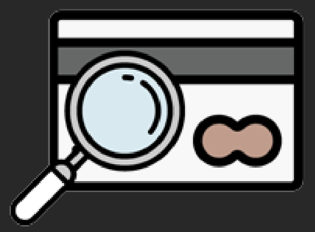
Neptune introduction, architecture, and monitoring

Graph use cases

Social networking

Recommendations

Knowledge graphs



Fraud detection

Life Sciences

Network and IT operations

Connected data queries

Navigate (variably) connected structure

Filter or compute a result based on *strength*, *weight*, or *quality* of relationships

Neptune: Fully managed graph database

Fast

Query billions of relationships with millisecond latency

Reliable

Six replicas of your data across three AZs with full backup and restore

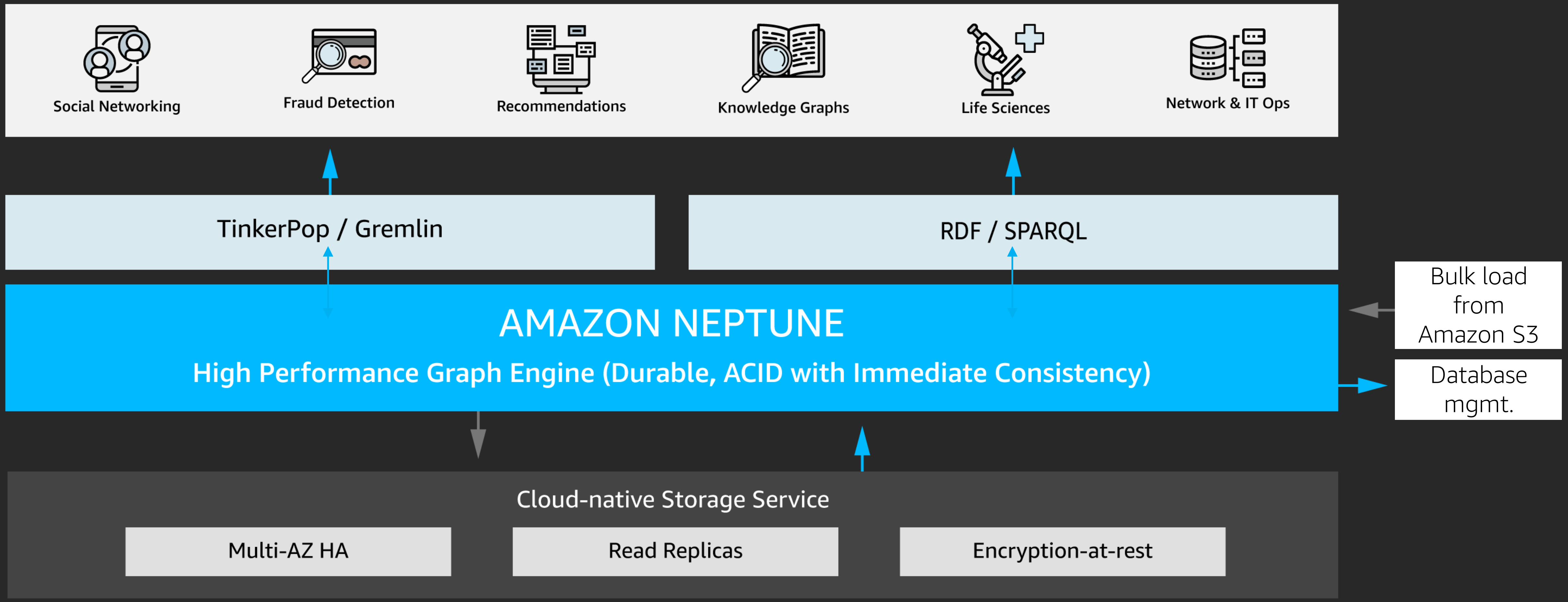
Easy

Build powerful queries easily with Gremlin and SPARQL

Open

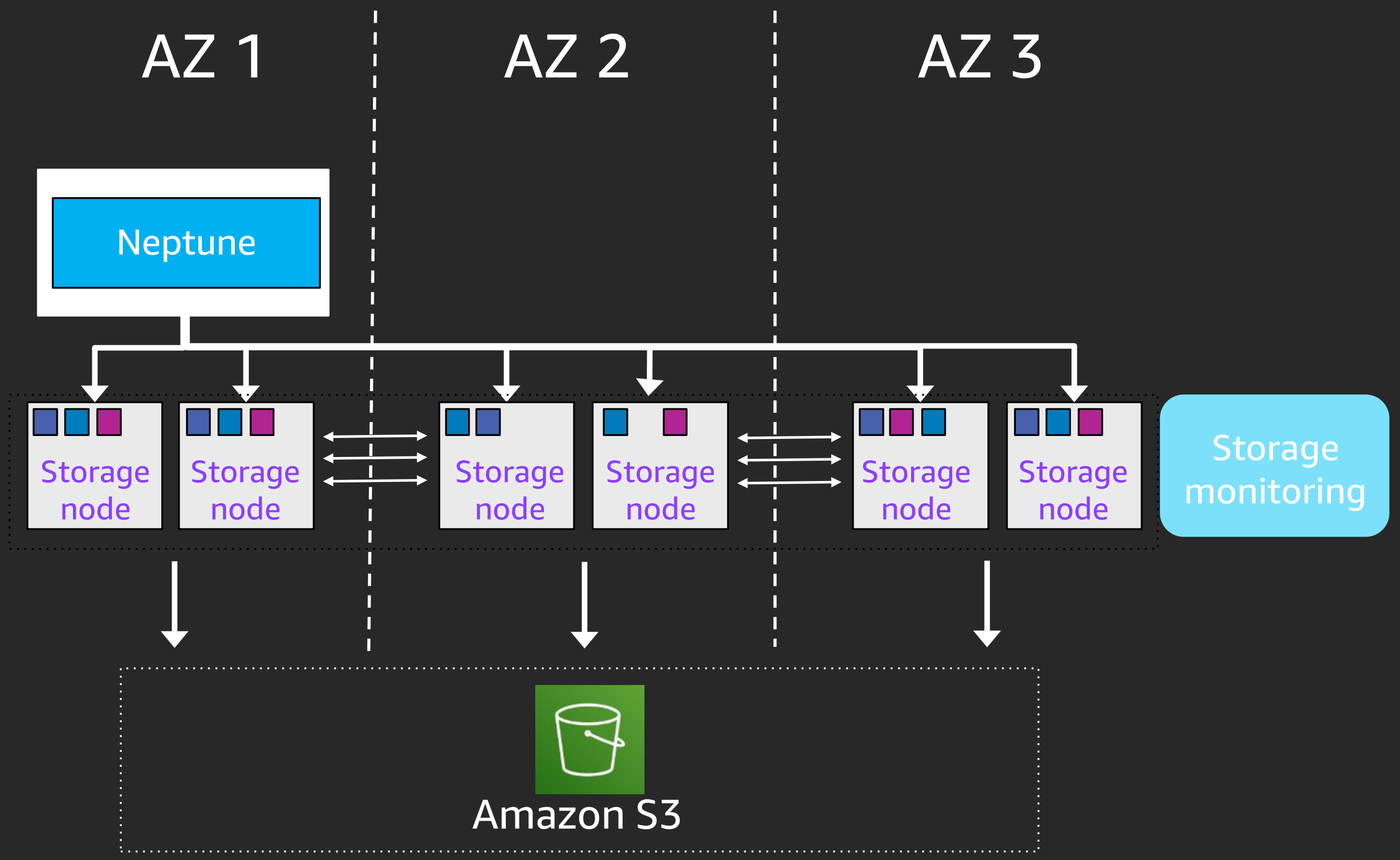
Supports Apache TinkerPop and W3C RDF graph models

Neptune architecture



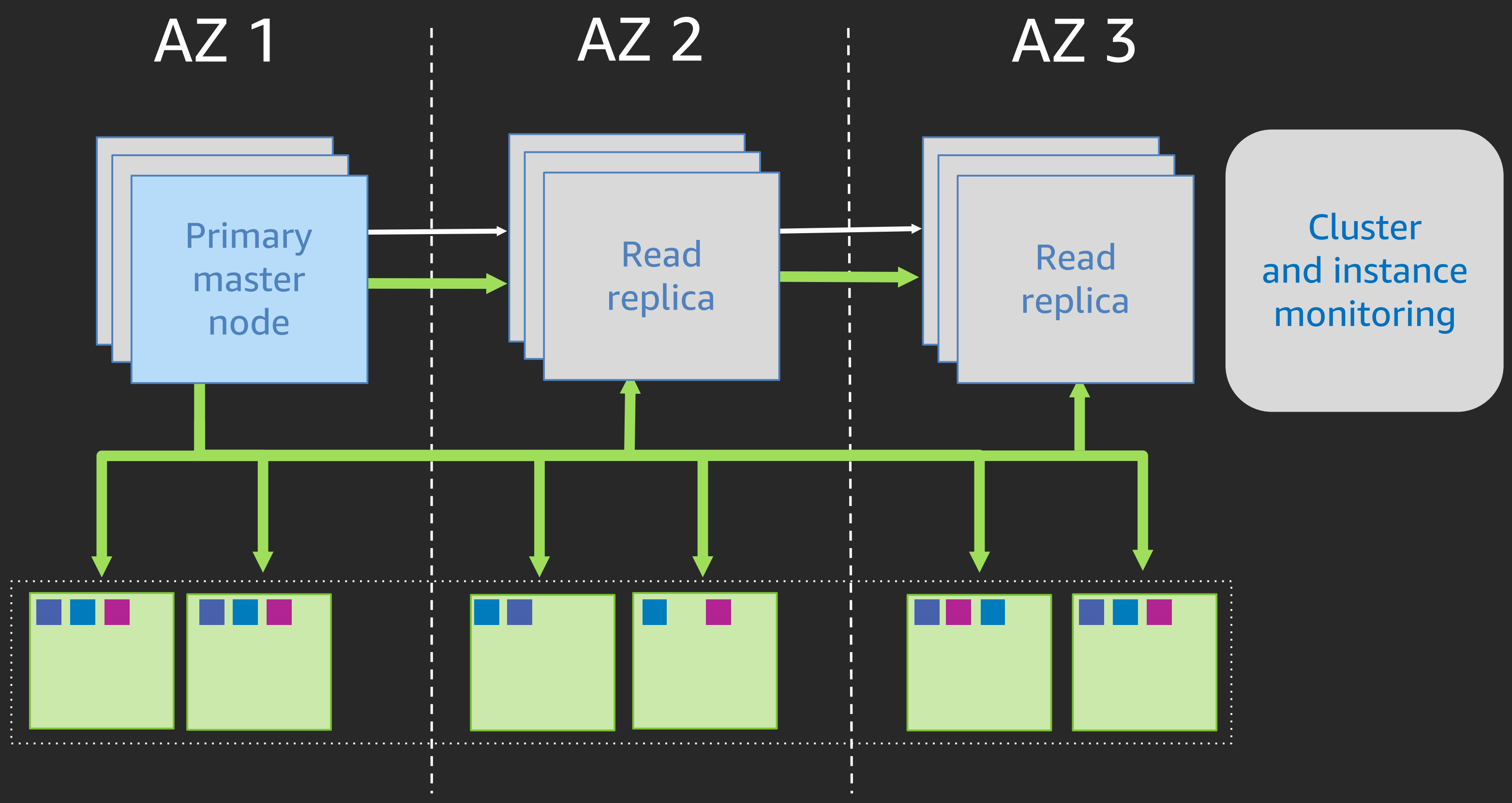
Cloud-native storage

- Data is replicated six times across three AZs
- Continuous backup to Amazon S3
  - Built for eleven nines of durability
- Continuous monitoring of nodes and disks
- 10 GB segments as unit of repair of hotspot rebalance
- Quorum system for read/write; latency tolerant
- Quorum membership changes do not stall writes
- Storage volume automatically grows up to 64 TB



Read replicas

- Availability
- Failing database nodes are automatically detected and replaced
- Failing database processes are automatically detected and recycled
- Replicas are automatically promoted to primary if needed (failover)
  - Customer specifiable failover order
- Performance
- Customer applications can scale out read traffic across read replicas
- Read balancing across read replicas
  - Use reader endpoint



Monitoring

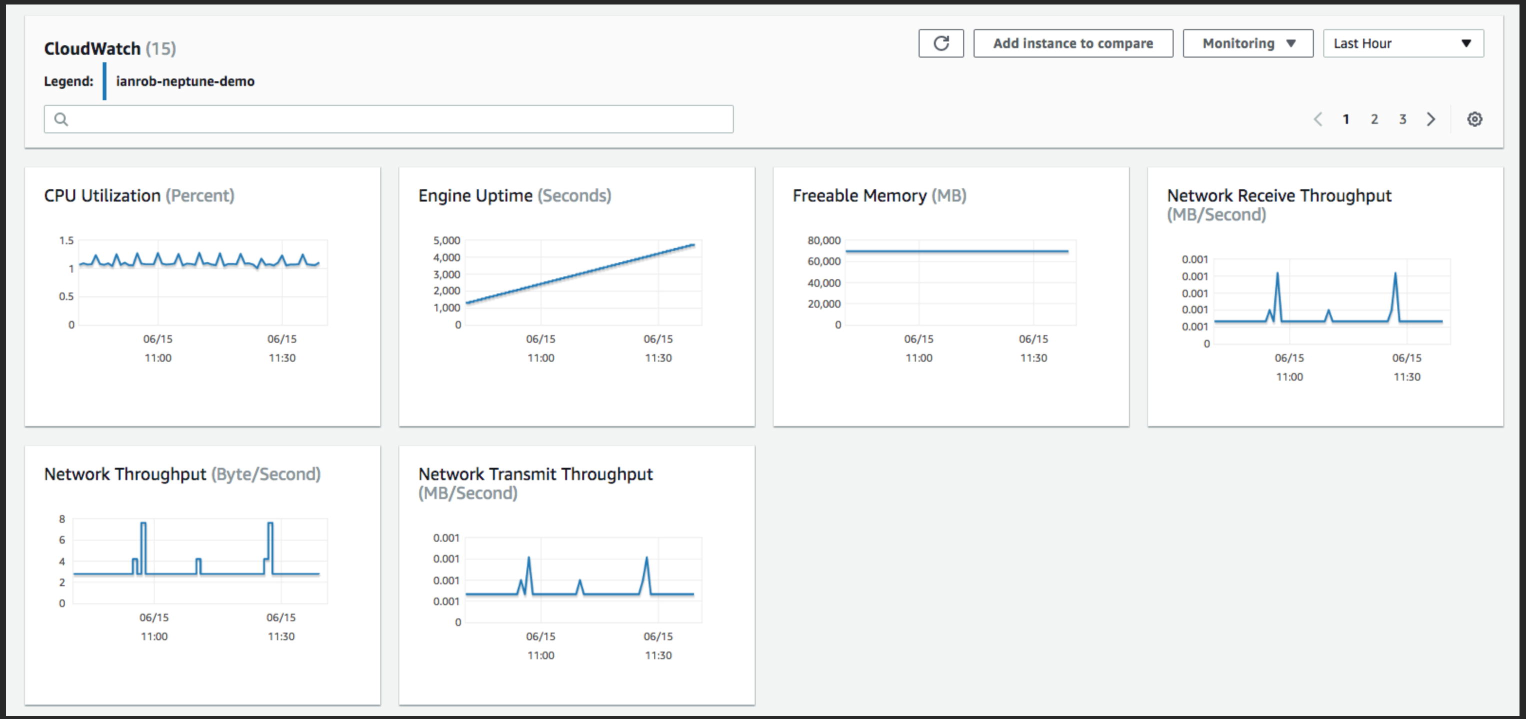
AWS CloudTrail

Log all Neptune API calls to S3 buckets

Event notifications

Create Amazon SNS subscription via CLI or SDK

Sources: db-instance | db_cluster | db-parameter-group | db-security-group | db-snapshot | db-cluster-snapshot



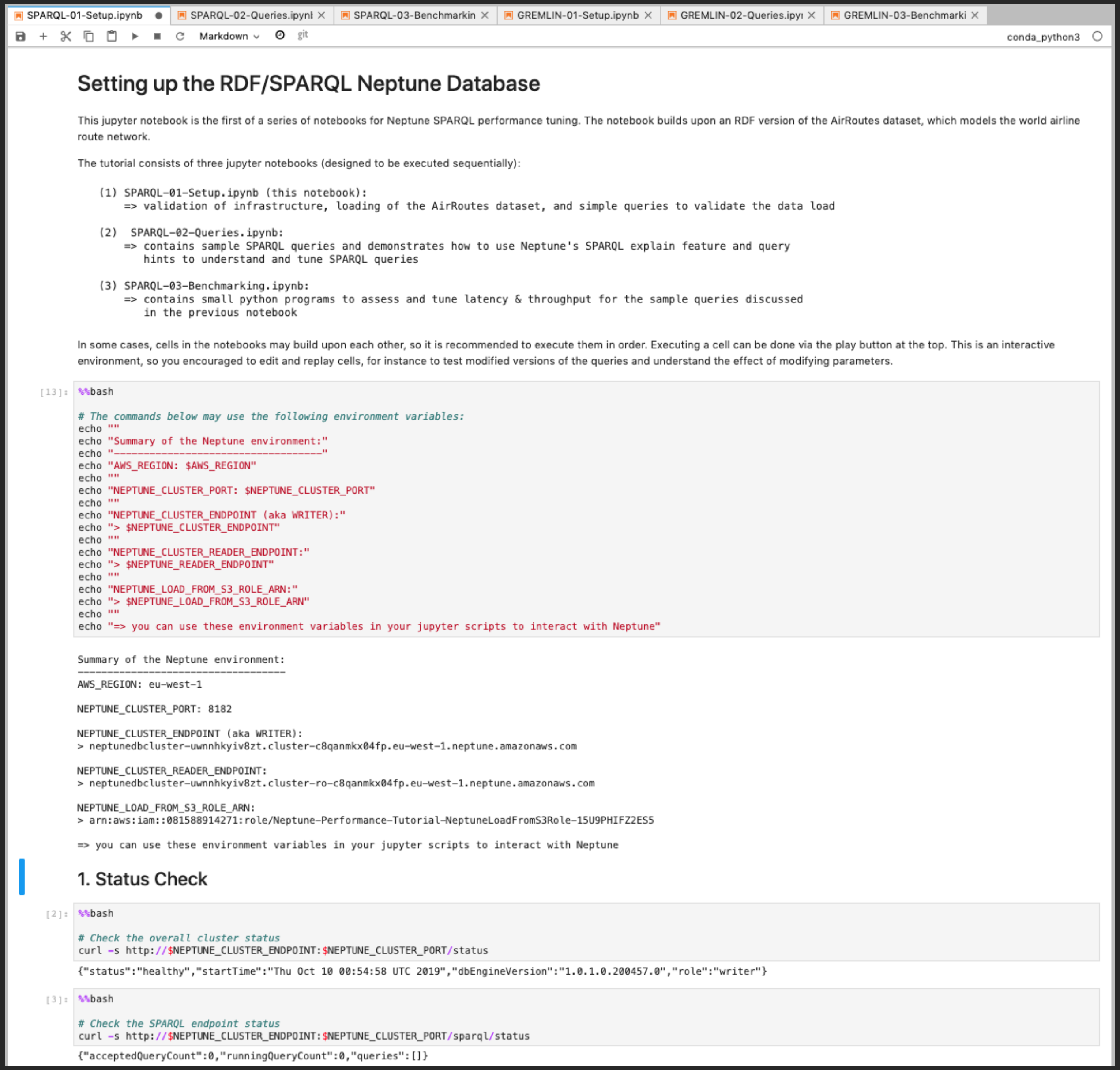
Amazon CloudWatch

BackupRetentionPeriodStorageUsed	GremlinRequestsPerSec	NumTxCommitted	TotalRequestsPerSec
CPUUtilization	GremlinWebSocketOpenConnections	NumTxOpened	TotalServerErrorsPerSec
ClusterReplicaLag	LoaderRequestsPerSec	NumTxRolledBack	VolumeBytesUsed
ClusterReplicaLagMaximum	MainRequestQueuePendingRequests	SnapshotStorageUsed	VolumeReadIOPs
ClusterReplicaLagMinimum	NetworkReceiveThroughput	SparqlRequestsPerSec	VolumeWriteIOPs
EngineUptime	NetworkThroughput	TotalBackupStorageBilled	
FreeableMemory	NetworkTransmitThroughput	TotalClientErrorsPerSec	

Hands on!

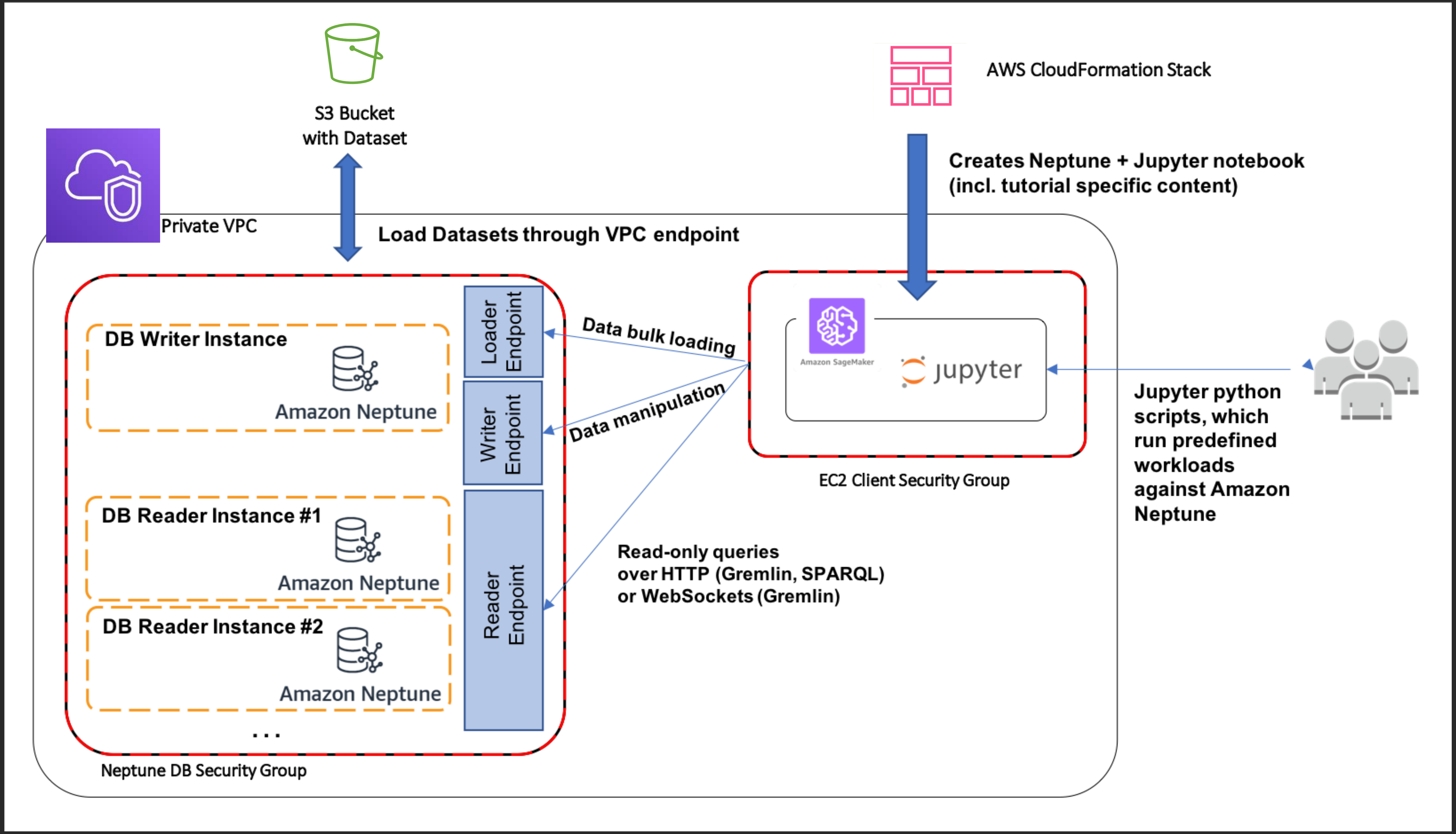
Run the experiments in your Jupyter notebook to learn about:

- Basic Neptune APIs
- Loading data into Neptune



https://dashboard.eventengine.run/login

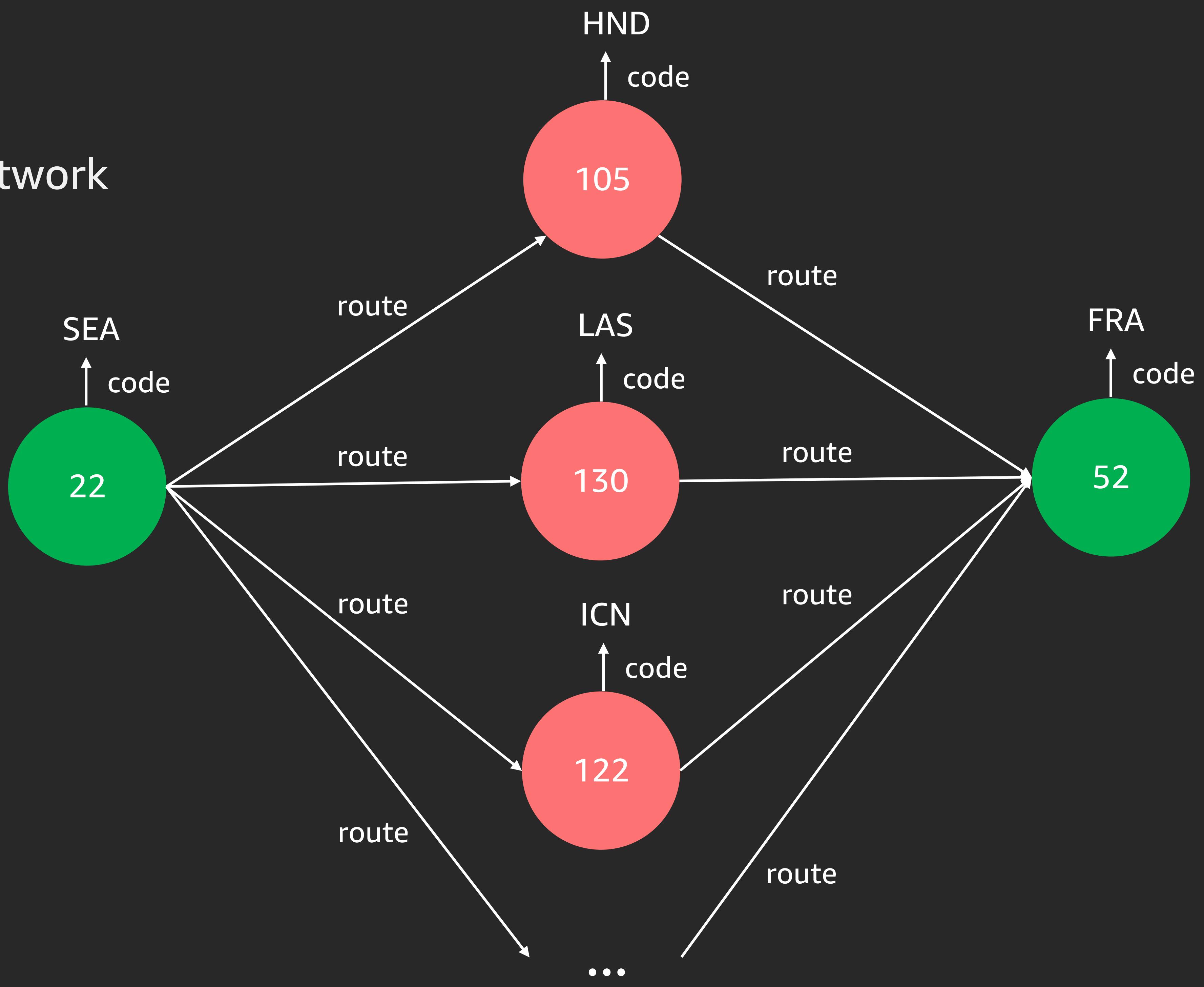
Architecture for hands-on experiments



Query optimization and performance tuning

Air routes dataset

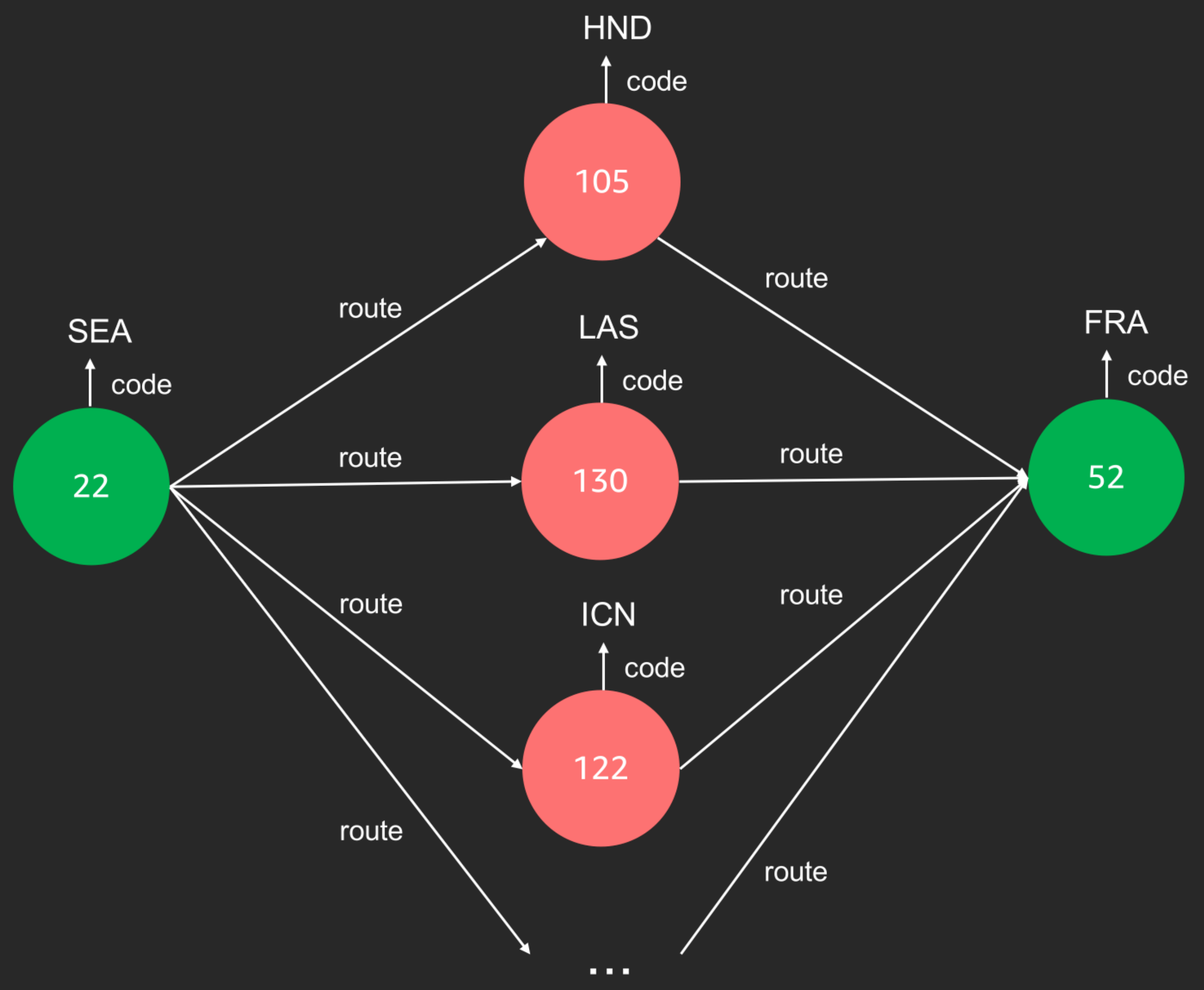
- Models the world's airline route network
- Queries operating over the airport connectivity graph
- Sample queries
  - Given
    - Source and target airport
  - Find
    - All one-stop connections



https://github.com/krlawrence/graph/tree/master/sample-data

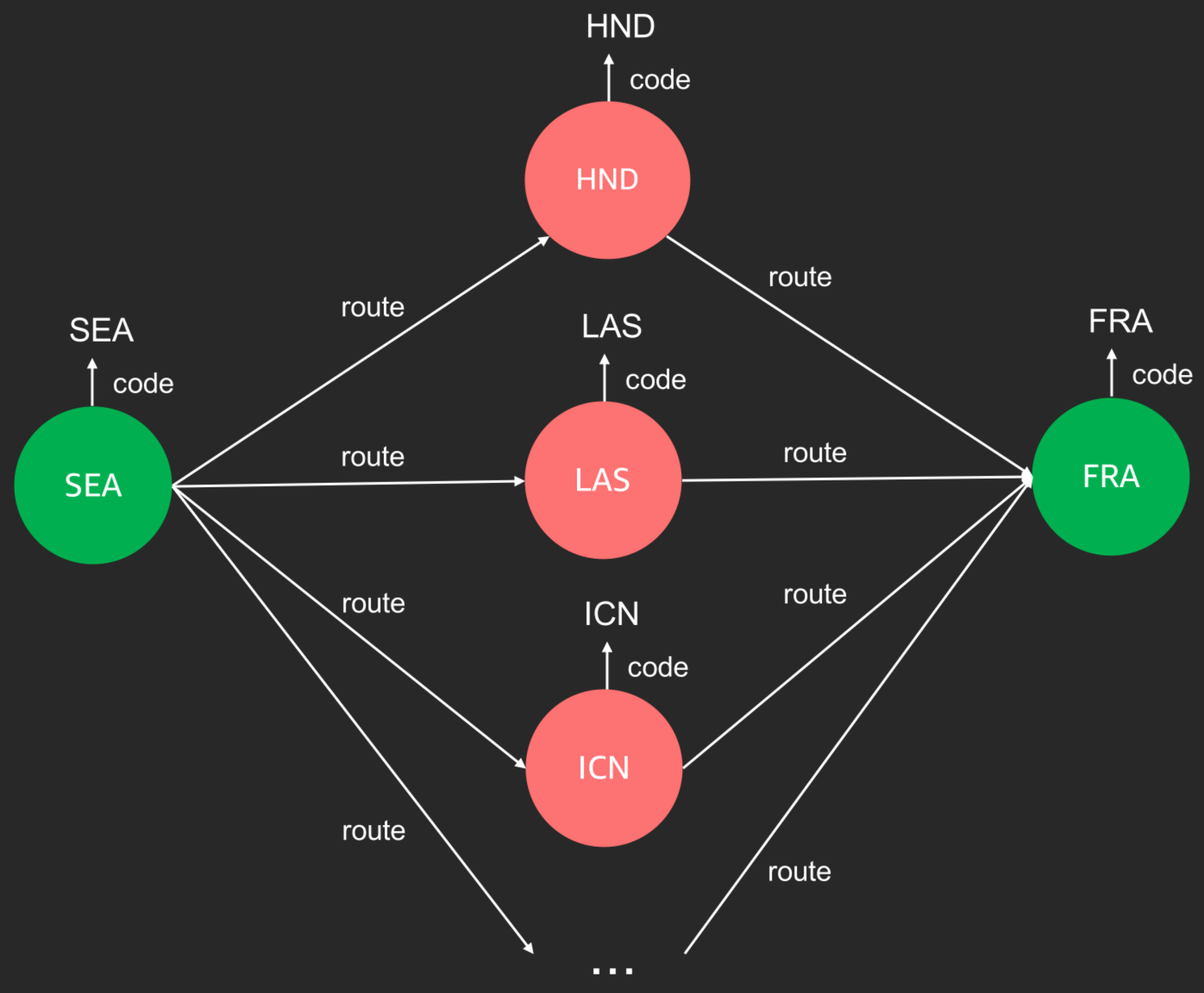
Sample query: Gremlin

```
# Gremlin

g.V()                // start out with all vertices
  .has('code','SEA') // select vertices having code = 'SEA'
  .out('route')      // follow 'route' edge
  .as('via')         // save node in variable 'via'
  .out('route')      // follow 'route edge again
  .has('code','FRA') // assert we ended up in FRA
  .select('via')     // jump back to the via airport
  .values('code')    // select airport code
```



Sample query: SPARQL

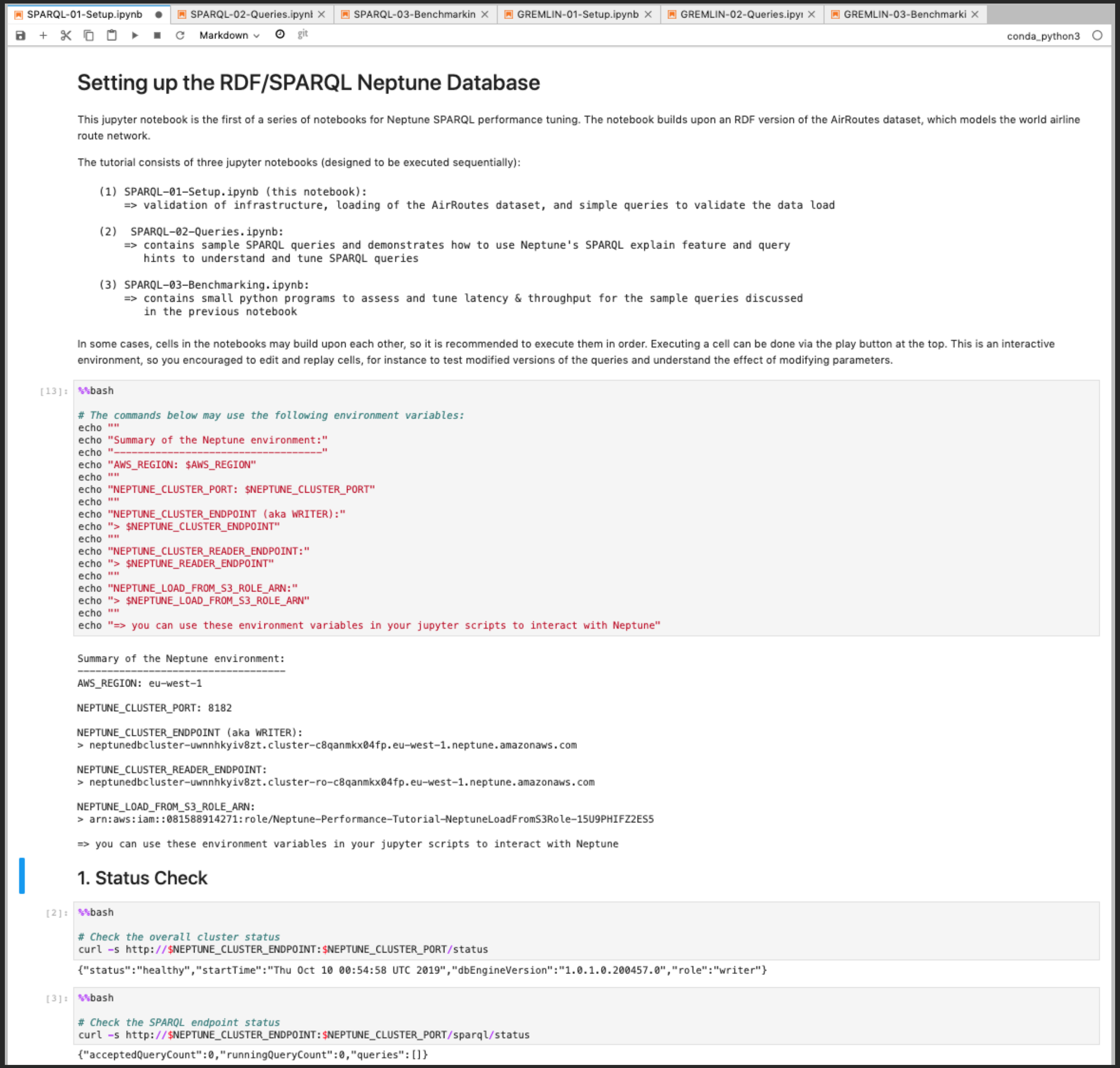
```
# SPARQL

PREFIX airport: <http://kelvinlawrence.net/air-routes/resource/airport/>

PREFIX edge: <http://kelvinlawrence.net/air-routes/objectProperty/>

PREFIX prop: <http://kelvinlawrence.net/air-routes/datatypeProperty/>

SELECT ?via ?viaCode WHERE {

    airport:SEA edge:route ?via .

    ?via prop:code ?viaCode .

    ?via edge:route airport:FRA .

}
```



Hands on!

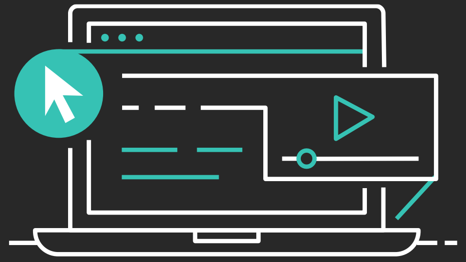
Run the experiments in your Jupyter notebook to learn about:

- Running, understanding, and tuning queries
- Performance monitoring via CloudWatch
- Measuring latency and throughput at client side
- Scaling throughput



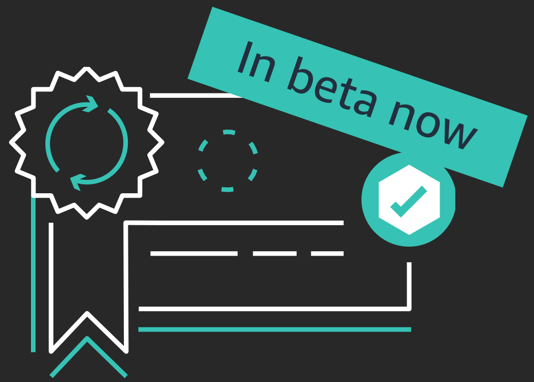
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

aws training and certification

Thank you!

AWS re:Invent

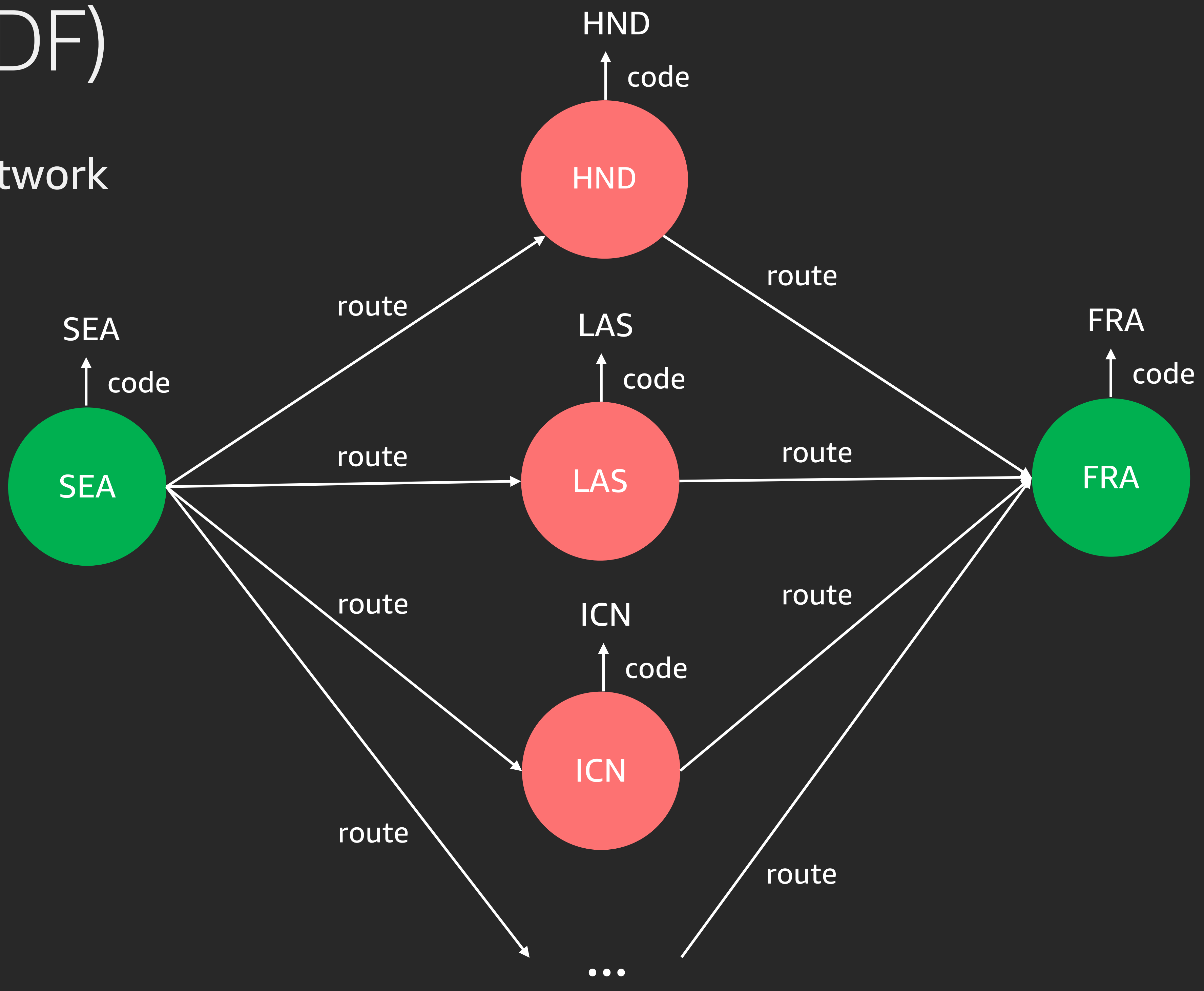
Neptune Reference Customers

THOMSON REUTERS
FINRA
SIEMENS
AstraZeneca
LIFEOMIC
amazon alexa
intuit
FIREEYE
SAMSUNG
Blackfynn
Uber ATG
NETFLIX
NBCUniversal
HUUUGE
asurion
PaySense
zeta
Pearson
MARINUS ANALYTICS
noonum

aws

Air routes dataset (RDF)

- Models the world's airline route network
- Queries operating over the airport connectivity graph
- Sample queries
  - Given
    - Source and target airport
  - Find
    - All one-stop connections



https://github.com/krlawrence/graph/tree/master/sample-data

Please complete the session survey in the mobile app.