Amazon Neptune Updates

Neptune ML と OpenCypher のサポートを中心に

五十嵐 建平
Sr. Manager, Enterprise Solutions Architecture
Neptune AoD

自己紹介

五十嵐 建平 (いがらし けんぺい)
Sr. Manager, CPG and Retail, Enterprise Solutions Architecture
Neptune Area of Depth (AoD)

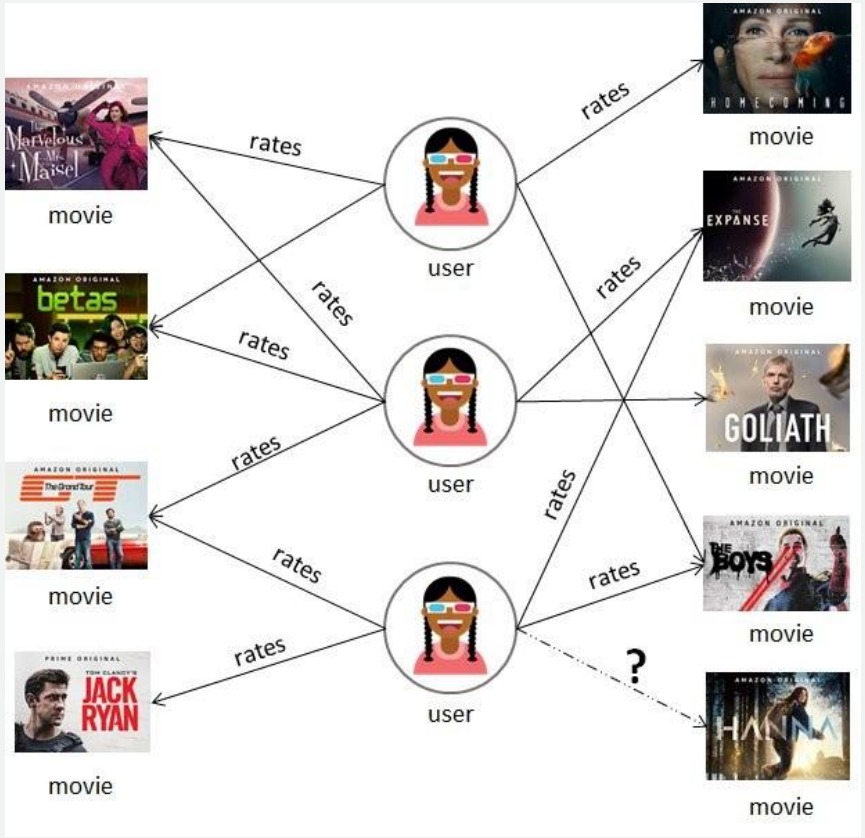
- 2017年～ AWS にてソリューションアーキテクト
  - 日本における Amazon Neptune の担当
- 2001-2017年までオラクル社でコンサルタント
  - Oracle RAC, Oracle Coherence などをデリバリ

アジェンダ

- Amazon Neptune ML
- OpenCypher サポート
- その他最新アップデート

Amazon Neptune ML

グラフ構造と機械学習



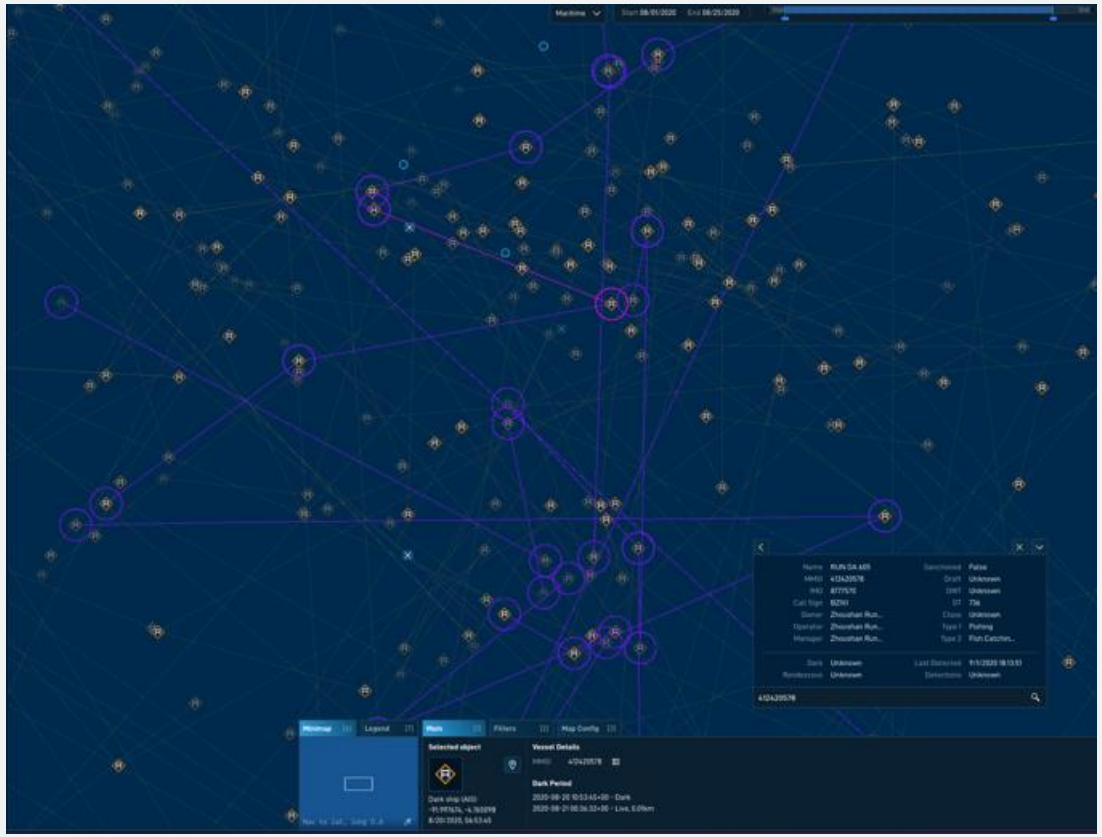
- データとデータの「つながり」はMLの予測精度を高めるために活用でき、他の方法に比べてユニークな洞察が得られる
- グラフ構造は柔軟性が高いため不正検知などで求められるデータやパターンの変化に対応がしやすい
- グラフ構造そのものを特徴量とした機械学習のアルゴリズムが複数実用化されている

Graph ML を実現するために従来求められてきたこと

- グラフDB、深層学習フレームワーク、モデル生成などの複数領域に渡る専門家
- グラフDBからデータをエキスポートして、モデルを作成し、推論を行い、結果をグラフDBに反映するという手間
- グラフ構造に関わる特徴量エンジニアリング
- Deep Graph Library に対する継続的な開発
- 一連の開発に関わるインフラの作成/管理/運用
- モデルの評価/選択やデプロイ
- 変化したデータに対処するための一連のプロセスの自動化

DGL + Neptune で実装された HawkEye360

- Amazon Neptune
  船舶間の関係を把握し船舶同士が
  どのように関連しているか分析
- DGL
  他の疑わしい船舶との関連性から
  各船舶のリスクスコアを作成し
  その船舶がどれだけ疑わしいことを
  するかを予測



HawkEye 360 による Deep Graph Libraryと Amazon Neptune を用いた船舶の運航管理リスク予測

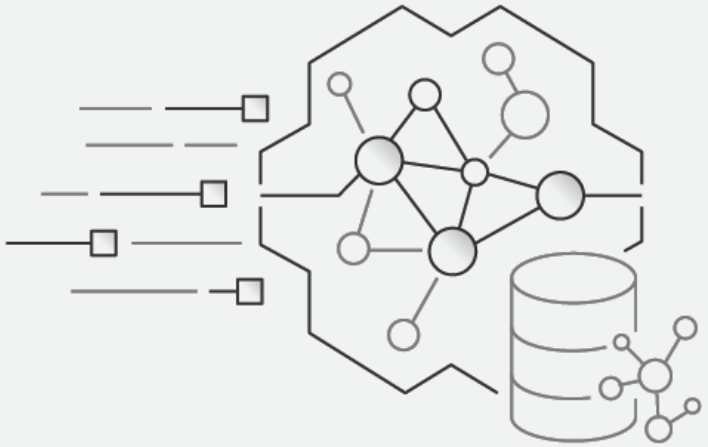
他のサービスと同様のチャレンジを GraphML にも

専門家が不要

スケーラビリティ

迅速に実現

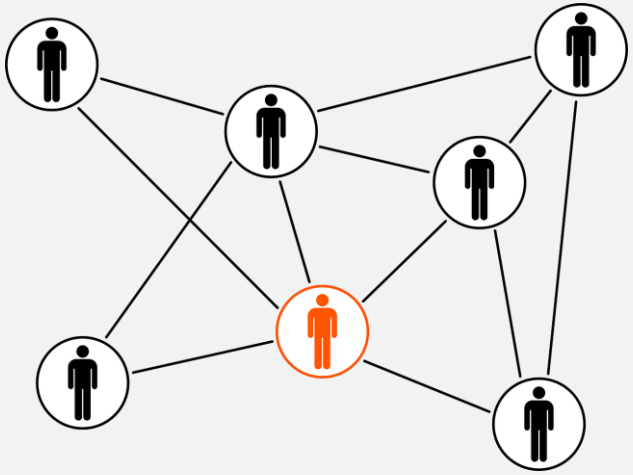
Amazon Neptune ML

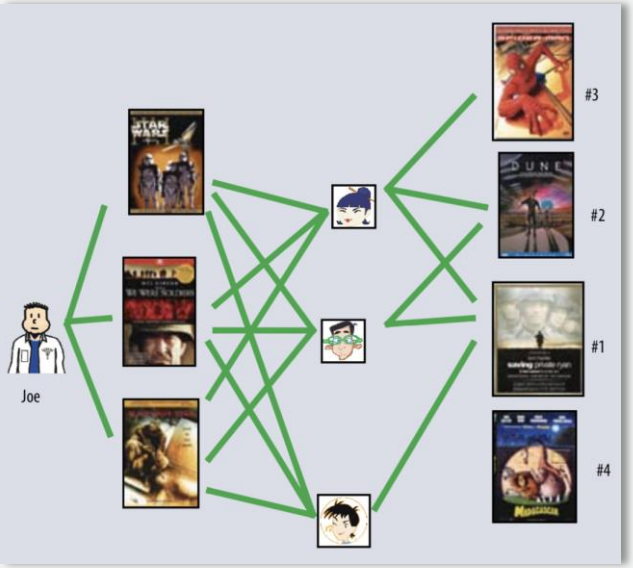


- グラフデータ上での予測をML専門家なしで実現
  - 自動的にベストなMLモデルを数週間ではなく数時間で選択/学習可能に
- サービス化されたグラフ上の深層学習
  - GNNは、スタンフォード大学の研究に基づき、グラフの関係性を利用することを目的として構築された最先端のMLであり、従来に比べて最大50%の精度向上を実現
- 巨大なデータセットにも対応するスケールする推論ランタイム
  - 大規模なナレッジグラフや不正検知、レコメンデーションで必要とされるような億単位のノード/リレーションにも対応

何が予測できるのか？

- ノードの予測
  - そのノードが持つカテゴリや数値のプロパティ
- エッジの予測
  - そのエッジが持つカテゴリや数値のプロパティ
- リンクの予測
  - ノード間のコネクションの存在を予測





Neptune ML の機能

これらが全て自動的に処理される

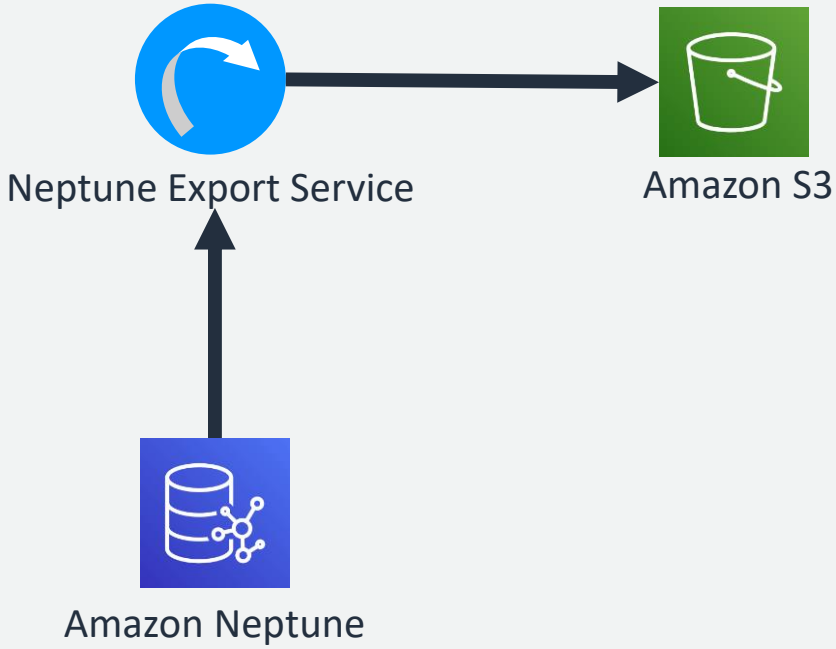
Neptune ML の動作

手順は4つ

データのエクスポートと設定

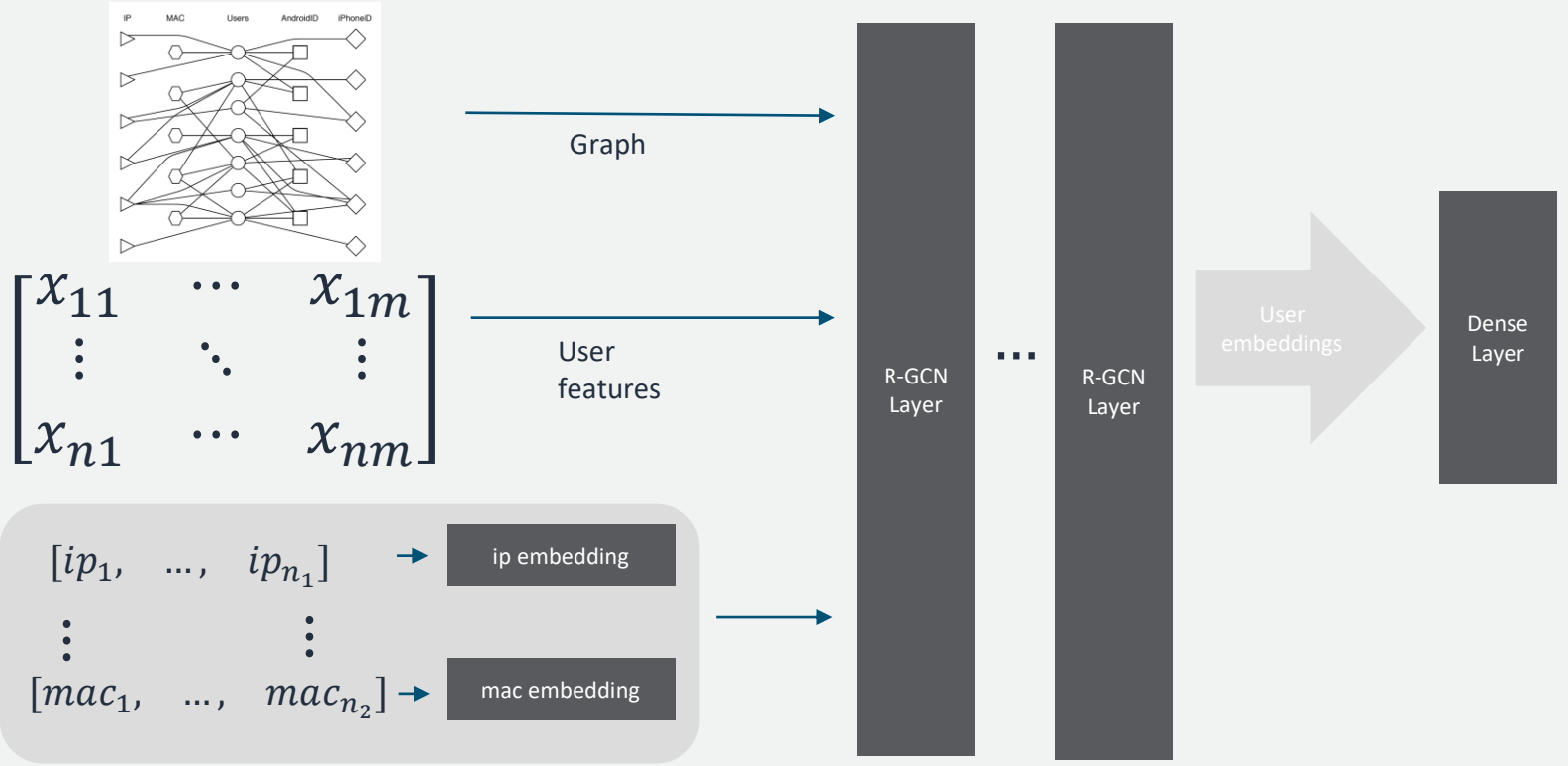
Neptune MLによってデータのエクスポートと設定の生成プロセスは単一のRESTコールに自動化される

```
{
"command": "export-pg",
"params": { "endpoint": "{hostname}",
            "profile": "neptune_ml",
            "useIamAuth": "{iam_role}",
            "cloneCluster": False
            },
"outputS3Path": "s3://{s3_bucket_uri}/neptune-export",
"additionalParams": {
        "neptune_ml": {
          "targets": [
            {
              "node": "movie",
              "property": "genre"
            }
          ]
          }
      },
"jobSize": "medium”
}
```



GNN MLモデルの学習

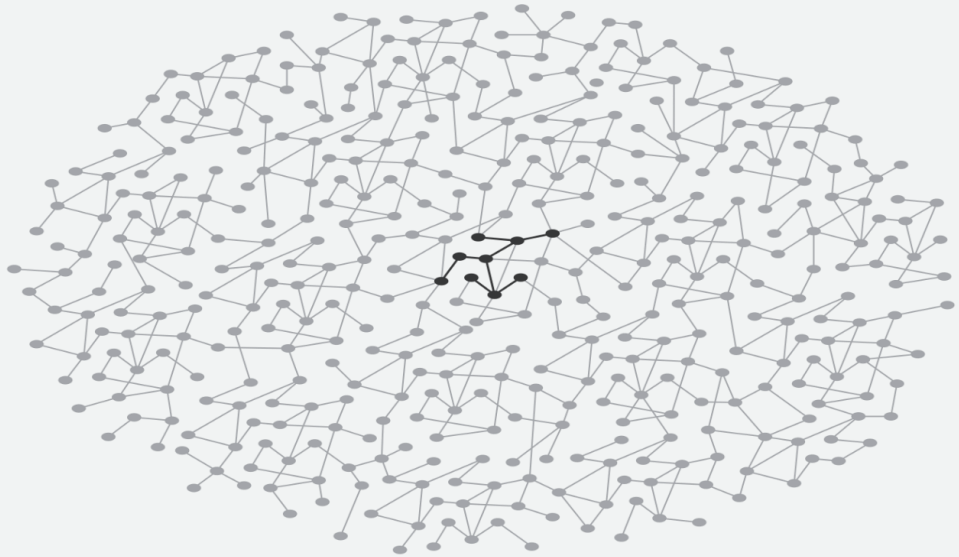
Neptune MLは、GNNモデルの学習、テスト、検証を自動化し、パラメータの調整やカスタマイズを可能にしながら、最適なモデルを自動的に選択します



推論エンドポイントの作成

Neptune MLはモデルに合わせたプロダクションレベルの機械学習エンドポイントのデプロイメントを自動化します

```
{
        "id" : ”{endpoint id}",
        "mlModelTrainingJobId": ”{training job id}"
}
```



トラバーサル(探索)中に推論

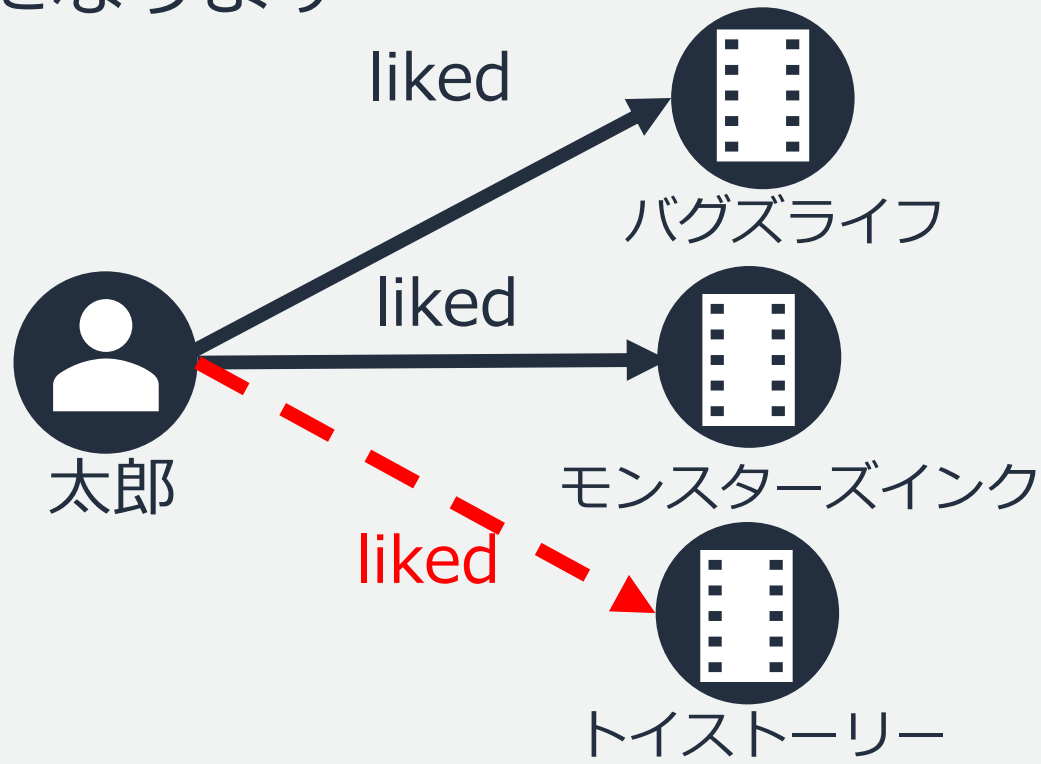
Neptune ML によって、探索クエリの途中で値を推論したり、リンクを推論して探索することができるようになります

推論しない通常の探索

```
g.
V().has('name', 'Bob').
out('liked').
hasLabel('movie').values('title')
⇒ A Bugs Life
⇒ Monsters Inc.
```

推論による探索

```
g.with("Neptune#ml.endpoint","ENDPOINT").
V().has('name', 'Bob').
out('liked').with("Neptune#ml.prediction").
hasLabel('movie').values('title')
⇒ Toy Story
```



グラフデータ更新に伴う推論モデルの更新

ワークフローの再実行

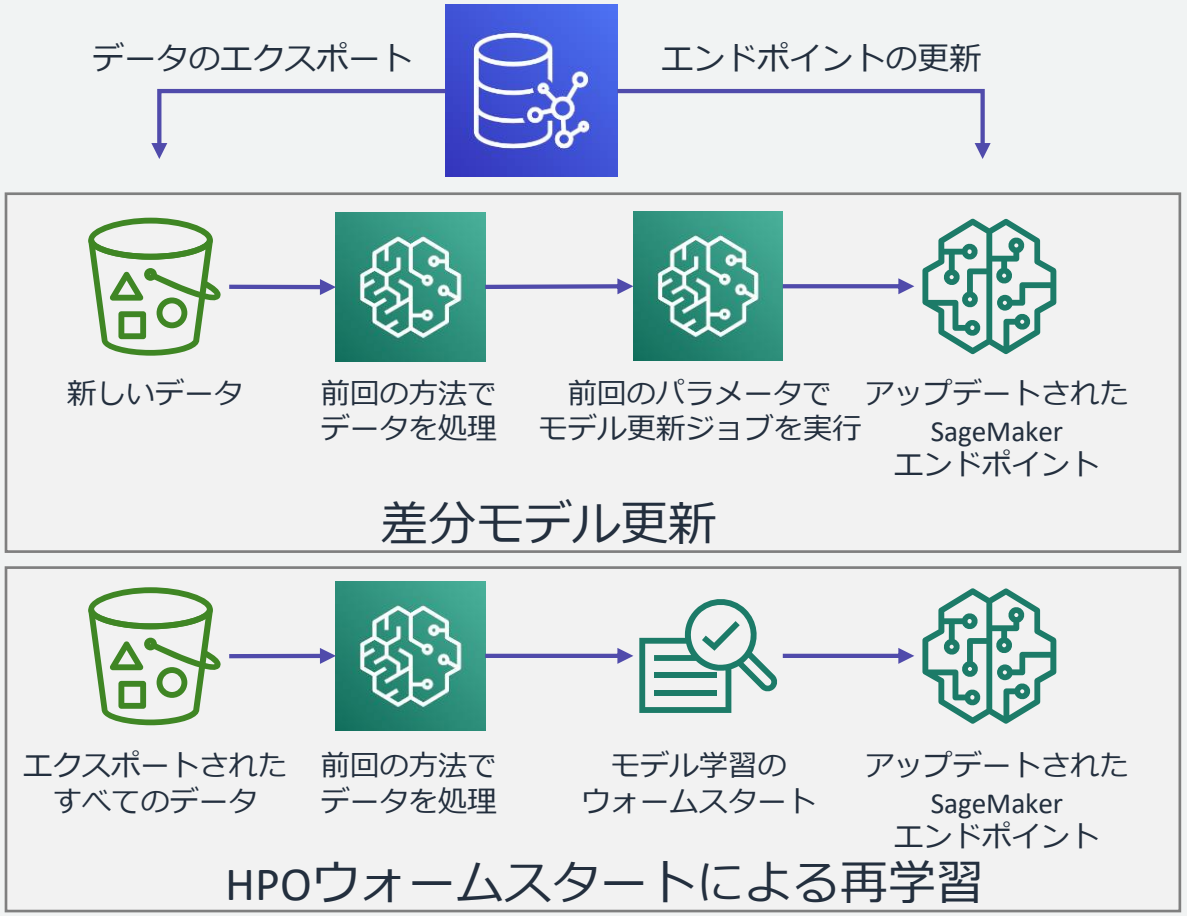
データのエクスポートし、データ処理を行い、トレーニングし新たな推論モデルを得る

差分モデル更新

差分データを用いて既存モデルを更新する

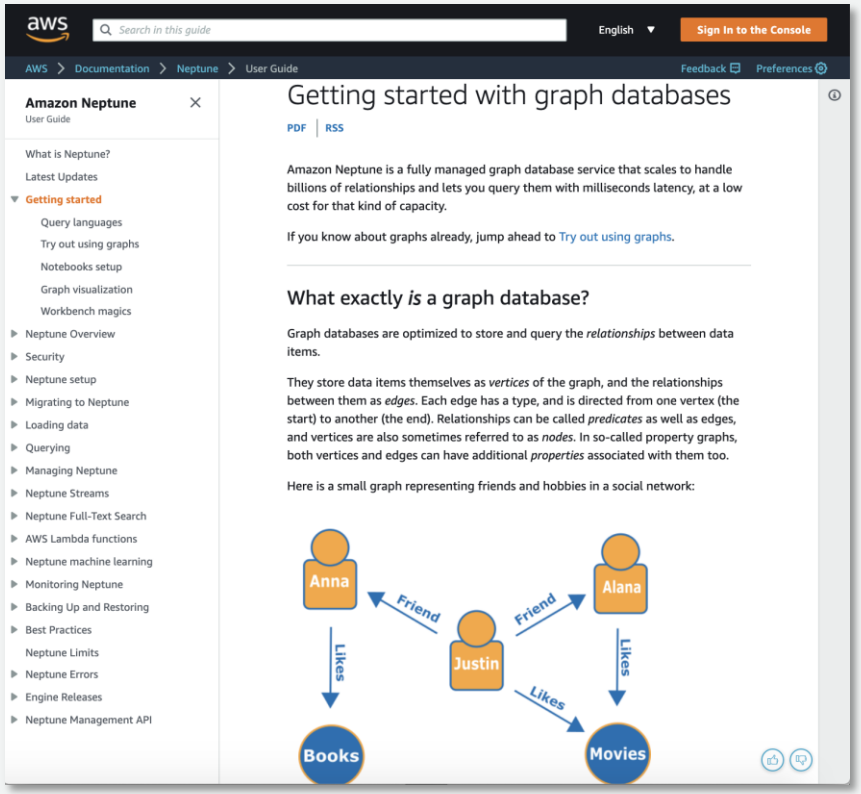
HPOウォームスタートによる再学習

既存モデルを開始地点としてハイパーパラメータチューニングジョブを実行して新たなデータのためのモデルとして更新する



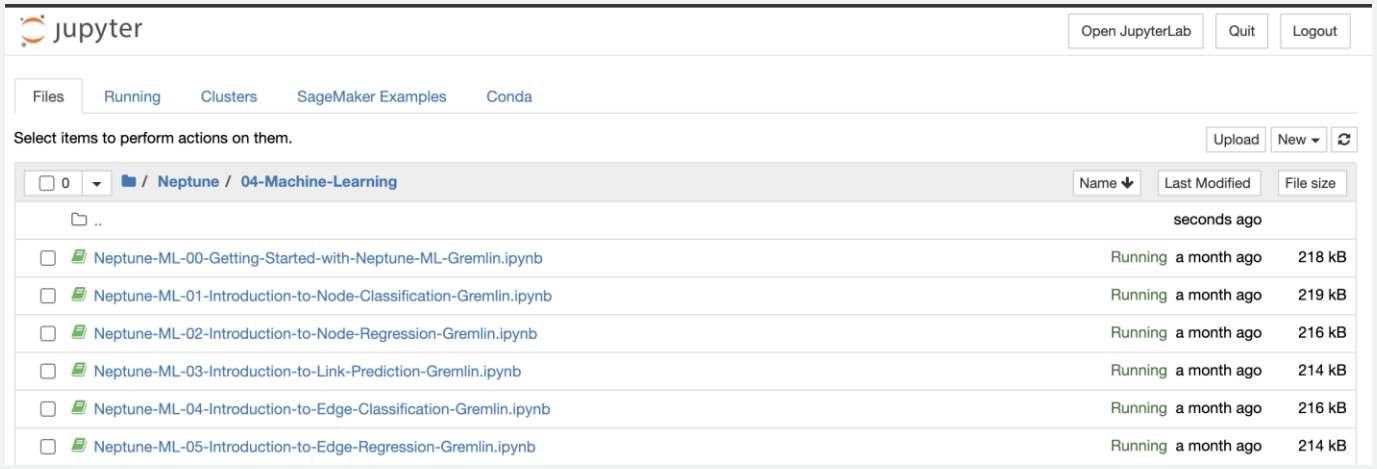
Neptune ML セットアップ

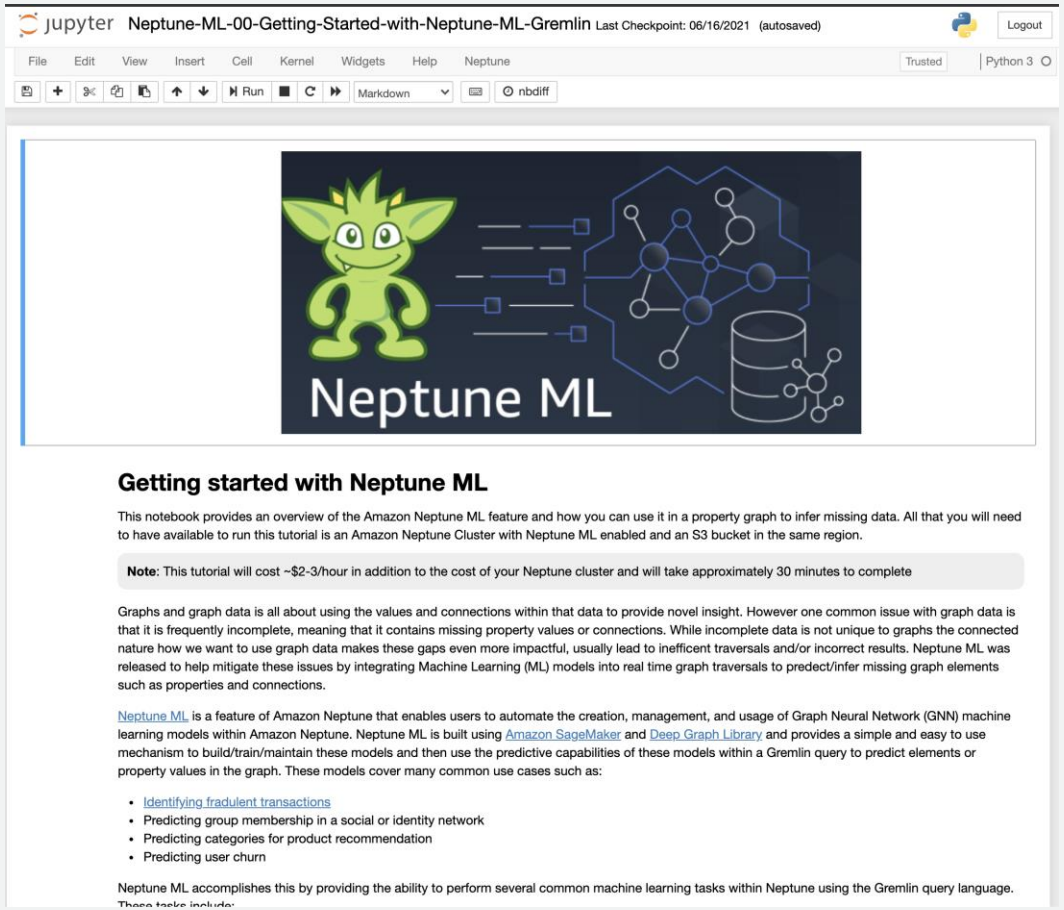
ドキュメントの Getting Started と Neptune ML の2つのセクションでセットアップできます



CFn テンプレートを使わない手動セットアップ手順はこちら

Neptune Notebooks に詳細なチュートリアルがあります





Neptune Notebooks: Introduction to Link Prediction の例

Predicting the top 10 users most likely to rate a movie

To accomplish this we would want to start at the movie vertex and predict the rated edge back to the user. Since we want to return the top 10 recommended users we need to use the `.with("Neptune#ml.limit",10)` configuration option. Combining these together we get the query below which finds the top 10 users most likely to rate `Apollo 13` .

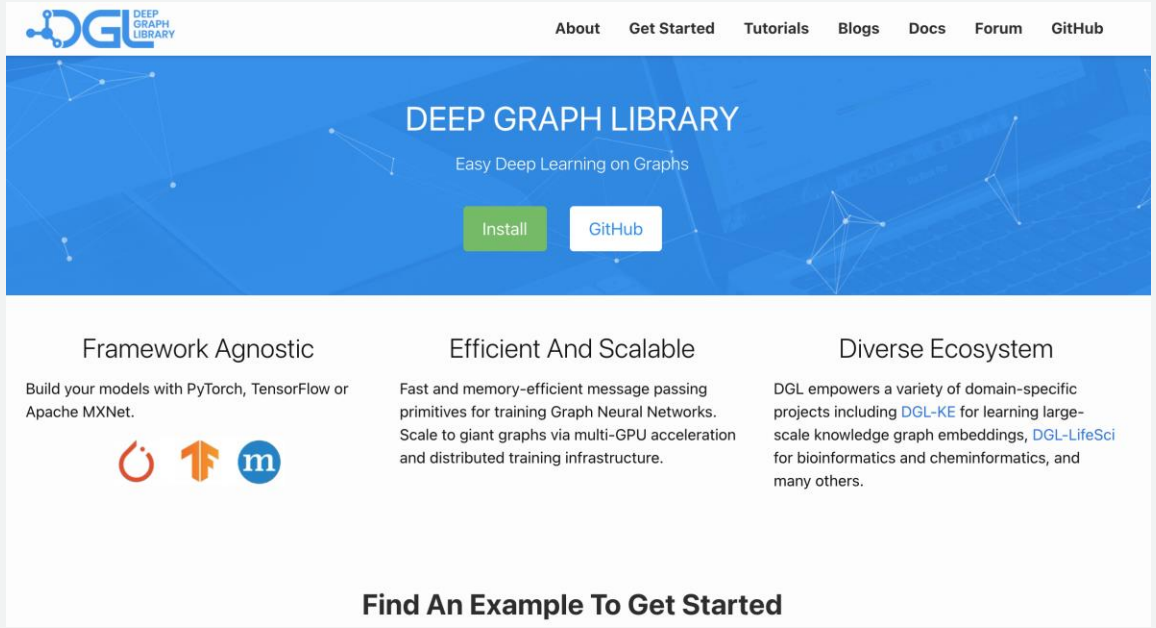
```
%%gremlin
g.with("Neptune#ml.endpoint","${endpoint}").
   with("Neptune#ml.limit",10).
    V().has('title', 'Apollo 13 (1995)').
     in('rated').with("Neptune#ml.prediction").hasLabel('user').id()
```

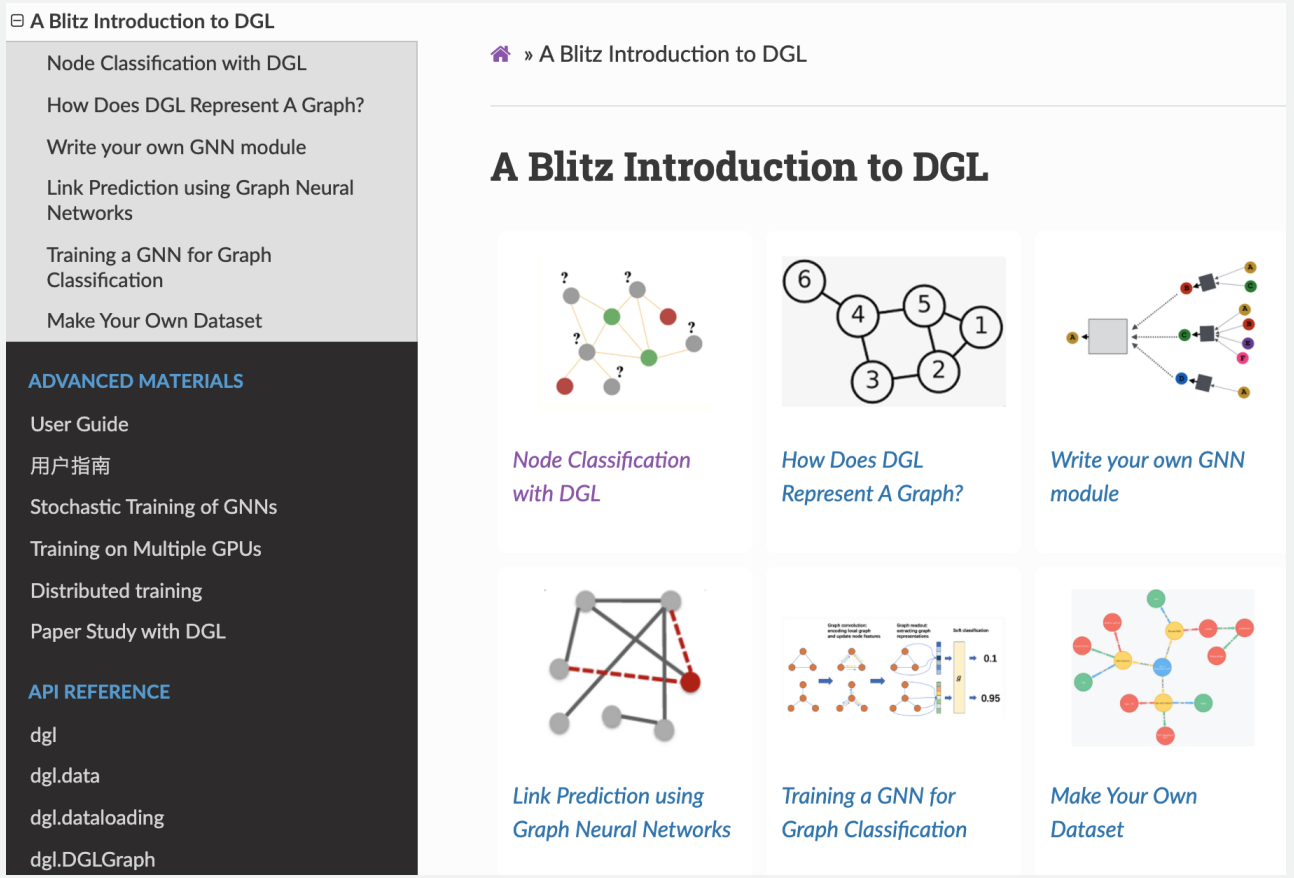
With that we have sucessfully been able to show how you can use link prediction to predict edges starting at either end.

From the examples we have shown here you can begin to see how the ability to infer unknown connections within a graph starts to enable many interesting and unique use cases within Amazon Neptune.

DGL ドキュメント

GET STARTED と User Guide でより詳細を学ぶことができます





OpenCypher サポート

クエリ種別: Global (OLAP) と Local (OLTP)

Graph Global “グラフ全体” (OLAP)
→ Graph Neural Network、クラスタリング、PageRank など

グラフ全体を分析するためのプラットフォーム
→ Spark GraphX、Deep Graph Library など

DGL+Neptune で実装された HawkEye360

クエリ種別: Global (OLAP) と Local (OLTP)

Graph Global “グラフ全体” (OLAP)
= Graph Neural Network、クラスタリング、PageRank など

グラフ全体を分析するためのプラットフォーム
= Spark GraphX、Deep Graph Library など

DGL+Neptune で実装された HawkEye360

Graph Local “グラフの一部” (OLTP)
= 補完、ルールベース探索、マッチング など

グラフの一部を探索するためのプラットフォーム
= Neptune など

起点ノードを取得 → 起点ノードの周辺を探索 → 境界パターンによりマッチ

2つのグラフモデルと3つのクエリ言語

プロパティグラフ

Gremlin トラバーサル言語

openCypher クエリ言語

(ラボモード)

Resource Description Framework (RDF)

SPARQL クエリ言語

OpenCypher

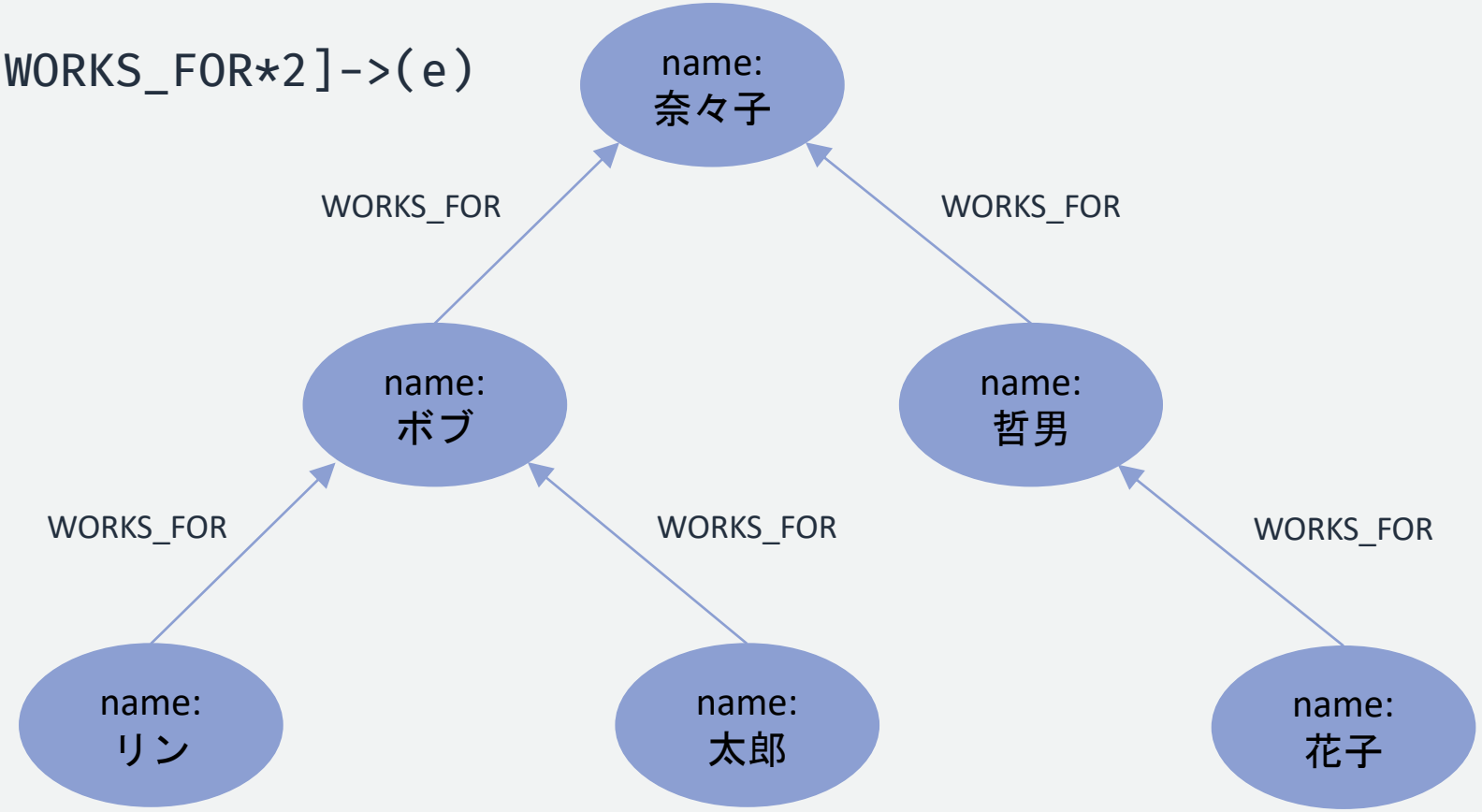
- パターンマッチングのためにデザインされたクエリ言語
  - Gremlin は探索のためにデザインされている
- SQL と似た宣言的構文
  - Gremlin は手続き的構文
- 複雑なクエリを作成する際には節を繋いでいく
- ASCIIアート的に目的となるパターンを表現する

```
MATCH (a:airport {code:'SEA'}0-[:route]->(dest:airport)
RETURN dest
```

OpenCypher の例

奈々子にレポートしている人数は？

```
MATCH ({name: '奈々子')-[:WORKS_FOR*2]->(e)
RETURN count(e)
```

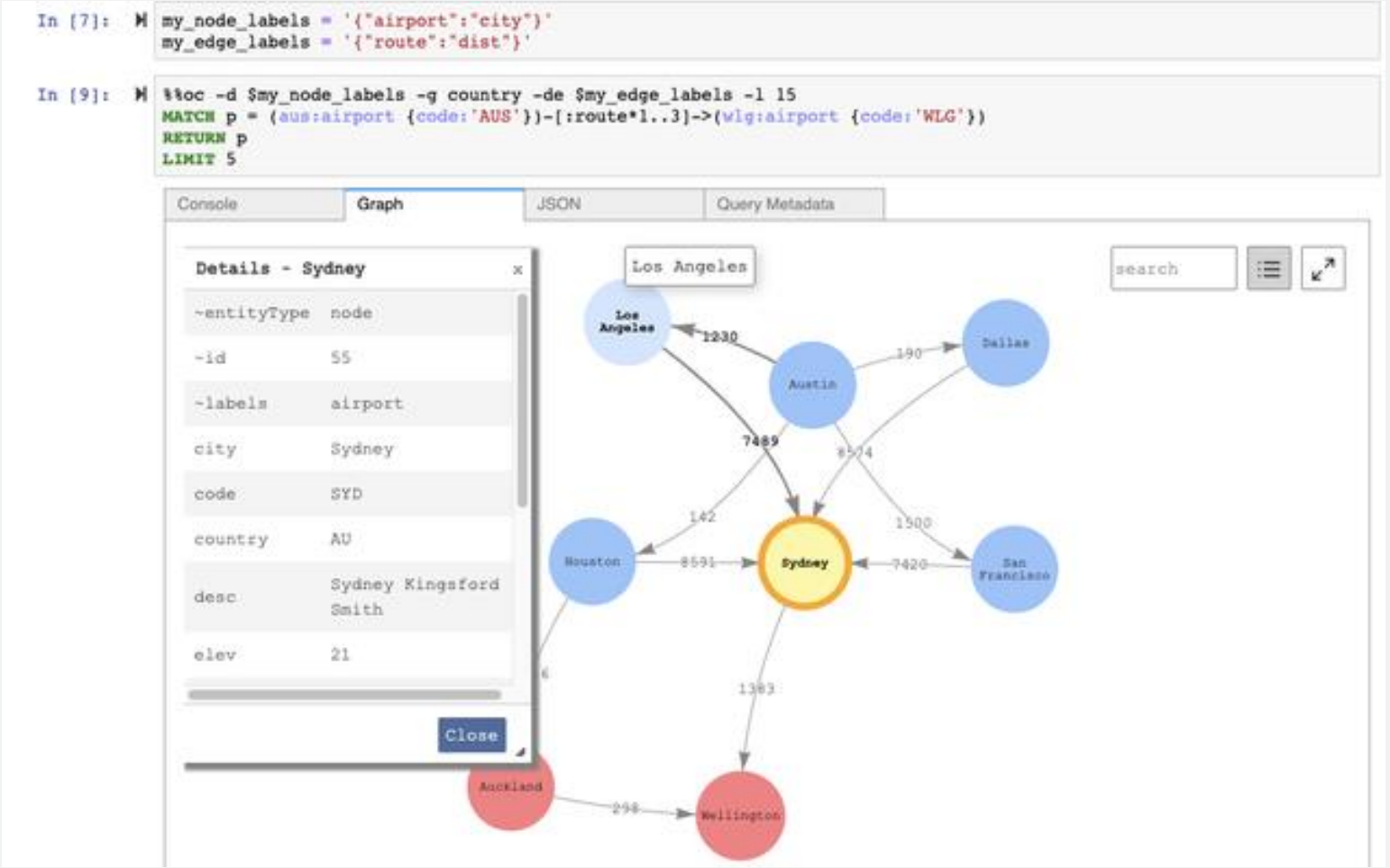


OpenCypher 関連のサポートするインタフェース

- HTTPSエンドポイント
- Bolt プロトコル
  - Java: neo4j-java-driver
  - Python: neo4j パッケージ
  - .NET: Neo4j.Driver
  - その他 Bolt をサポートするツール郡
- Neptune Notebook
- Statusエンドポイント

Neptune Notebook における OpenCypher サポート

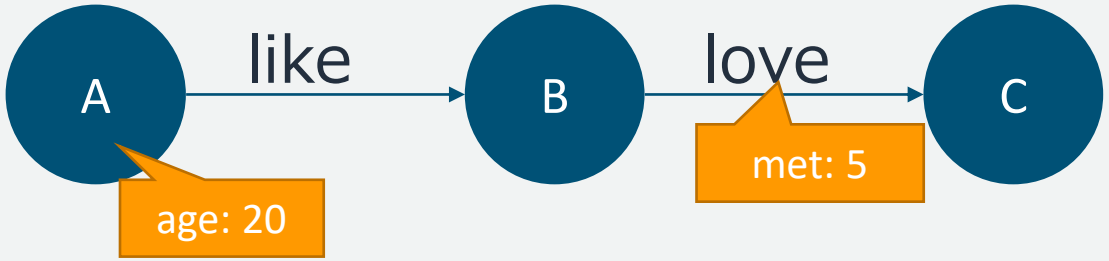
%%oc
にてセル内に記述



Neptune 内部でのグラフデータモデルの実装

プロパティグラフ/RDF どちらも同じグラフデータモデルを使用

https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/feature-overview-data-model.html



A	like	B	A	age	20
S	P	O	S	P	O
B	love	C	love	met	5
S	P	O	S	P	O

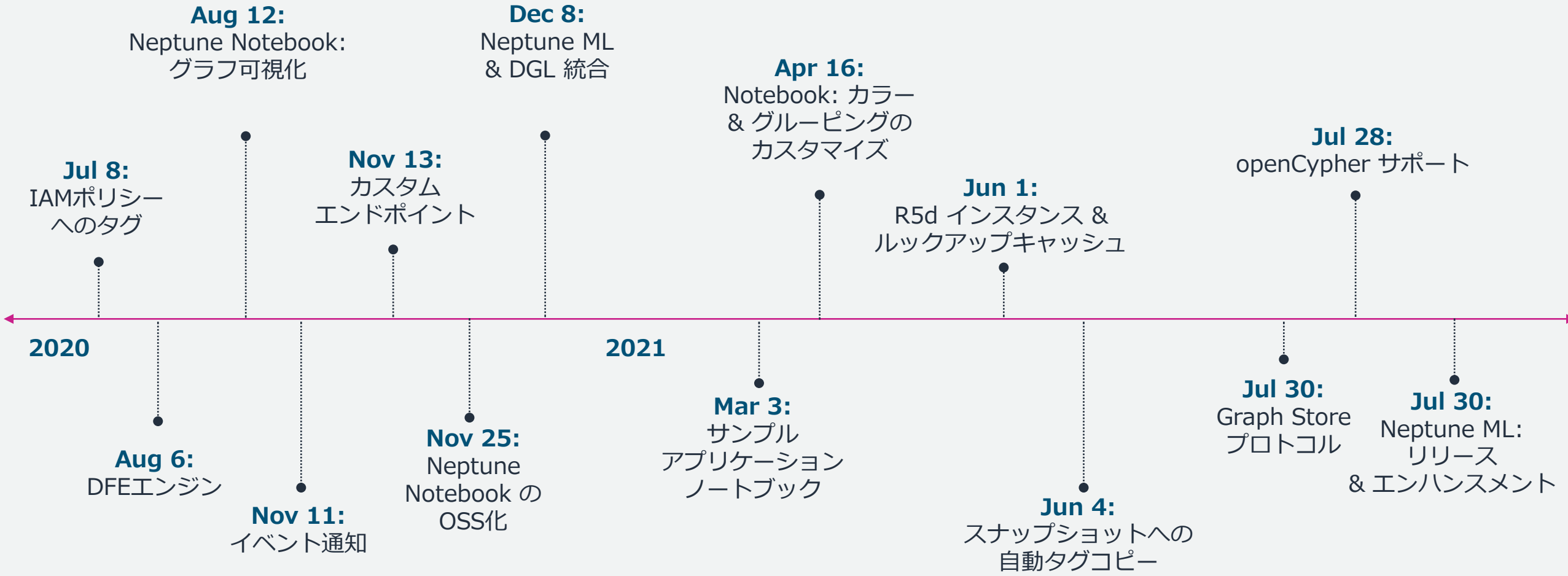
つまり同一データに対して
Gremlin / OpenCypher 両方で
アクセス可能

OpenCypher は現在ラボモード

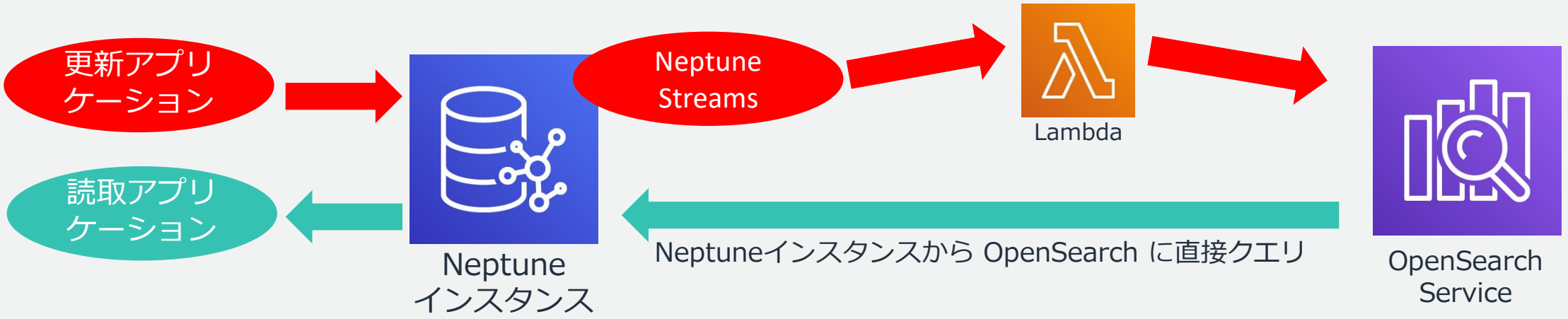
- ラボモードのため本番使用は不可
- 現在の制限事項はこちら
- 不具合等フィードバックがあれば以下へメールでご連絡を
  - neptune-opencypher-feedback@amazon.com (英語)

その他アップデート

2020-2021年のアップデート



Amazon OpenSearch Service 連携



- 索引の代わりに OpenSearch を使用して、**起点ノードのIDを OpenSearch から取得**
- Neptune に組み込まれた拡張構文でフェデレーションクエリとして動作する
- Neptune Streams を用いた更新アーキテクチャについても提供されている(必須ではない)

全文検索による起点ノードの取得: Gremlin の例

```
Ex1)
g.withSideEffect("Neptune#fts.endpoint", "your-es-endpoint-URL")
.withSideEffect(“Neptune#fts.queryType”, “query_string”)
.withSideEffect(“Neptune#fts.minScore”, 1.5)
.V().has(”company", "Neptune#fts *Amazon*");

Ex2)
g.withSideEffect("Neptune#fts.endpoint", "your-es-endpoint-URL")
.withSideEffect(“Neptune#fts.queryType”, “fuzzy”)
.V().has(”company", "Neptune#fts Amazon");
```

https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/full-text-search.html

全文検索による起点ノードの取得: SPARQL の例

- 経済産業省様作成の
  gBizInfo APIマニュアルより抜粋
  https://info.gbiz.go.jp/api/document/API.pdf
- gBizInfo 事例の詳細はこちら
  https://aws.amazon.com/jp/solutions/case-studies/meti/

(ウ) サンプル SPARQL クエリ

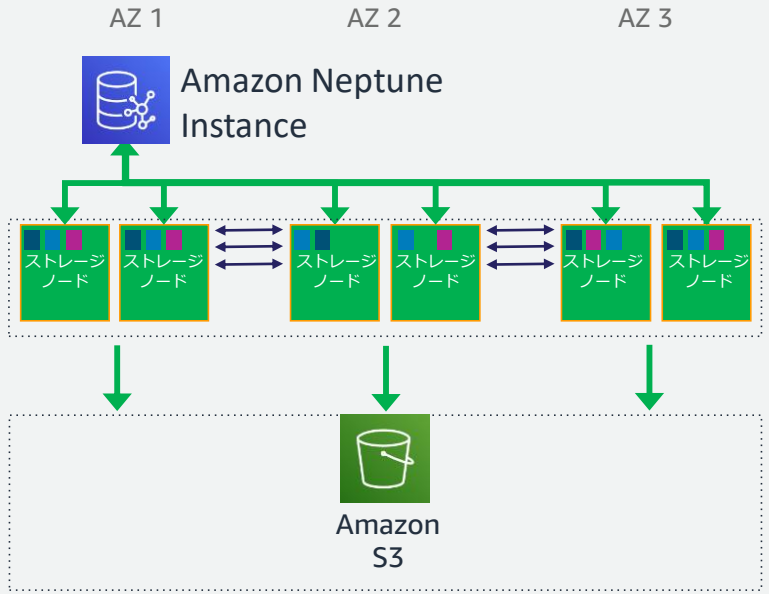
サンプル SPARQL クエリを以下に示す。

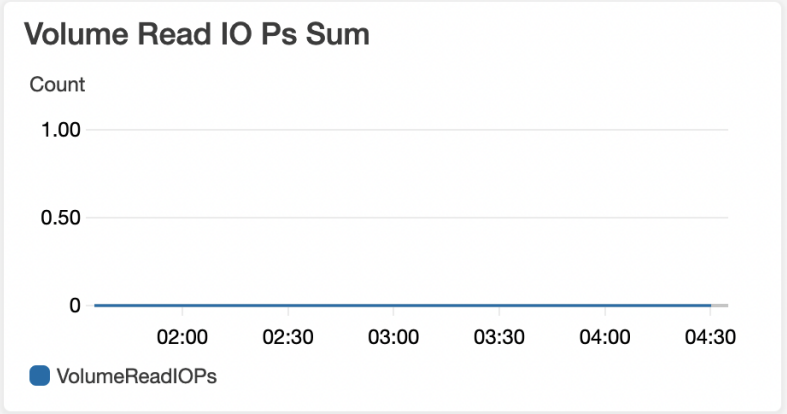
① 法人を指定して検索する

```
PREFIX  hj: <http://hojin-info.go.jp/ns/domain/biz/1#>
PREFIX  ic: <http://imi.go.jp/ns/core/rdf#>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
 SELECT distinct ?corporateID ?corporateName ?location
 {
    GRAPH <http://hojin-info.go.jp/graph/hojin> {
        SERVICE neptune-fts:search {
             neptune-fts:config      neptune-fts:endpoint      'https://vpc-pgi-es-domain-1-
gxjkidgb7s75w7nqzc7rlneig4.ap-northeast-1.es.amazonaws.com' .
            neptune-fts:config neptune-fts:queryType 'simple_query_string' .
            neptune-fts:config neptune-fts:field ic:表記 .
            neptune-fts:config neptune-fts:query '日立製作所' .
            neptune-fts:config neptune-fts:maxResults 100 .
            neptune-fts:config neptune-fts:return ?key .
        }
        ?key ic:名称 ?keyCorporateName.
        ?keyCorporateName ic:表記 ?corporateName.
        ?key ic:ID/ic:識別値 ?corporateID.
        ?key ic:住所/ic:表記 ?location.
    }
}
```

隣接探索時のストレージ I/O について

- クエリ処理はすべて**インスタンスのメモリ内**で実施される
- 必要なノード、エッジ、索引情報がメモリ内になければクラウドネイティブストレージからロードされる





モンスターノード問題

P(N) = エッジ数Nのノード
単純に辿るとエッジ数が乗算されていく
Nが大きい = モンスターノード/スーパーノード

P(10) → P(10) → P(10) = 1000
P(10) → P(10000) → P(1000) = 100000000
P(10000) → P(10000) → P(10000) = 1000000000000

→ 対策として探索の戦略を変更するという方法がある

幅優先探索と深さ優先探索

Neptune の探索アルゴリズムのデフォルトは幅優先(BFS)なので
条件に合うノードが深い場所にしかないことがわかっている場合効率が悪い

例: 条件[葉]: 木(2) → 幹(100) → 枝(500) → 葉(100) → 葉脈…

```
g.withSideEffect('Neptune#repeatMode','DFS')
.V().hasLabel('root')
.emit().repeat(out()).until(hasLabel('leaf')).path()
```

Best Practice:
repeatMode を変更することで深さ優先探索(DFS)に変更することができる

深い場所に必ずしもない場合 DFS では計算量爆発する可能性が高い
→ **CHUNKED_DFS**: 各深度で1000までしか探索しないハイブリッド探索も有効

参照: Gremlin repeatMode query hint

大量結果によるメモリ侵食を緩和: ルックアップキャッシュ

- 得られた結果セットが大きく大量のプロパティも合わせて返却しなければならない場合メモリが侵食されてパフォーマンスに影響がある
- リードレプリカでも解決可能ではあるが『ルックアップキャッシュ』を用いるとより低コストに解決できる可能性がある
(R5dインスタンスを使用すると自動的に有効になる)
- 探索時/集計時の処理では使われないので注意

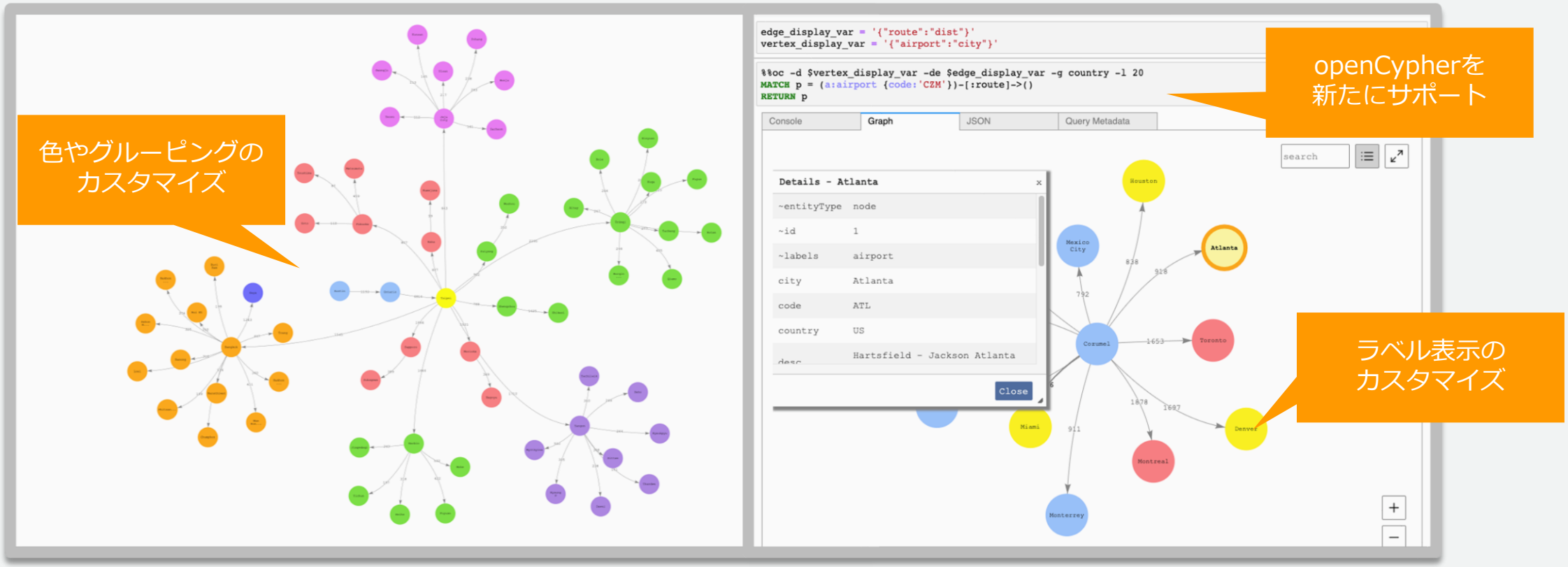
クエリ結果

ID: ## X 10000

```
company: AWS
ipaddr: x.x.x.x
cookie: xyz
counter: xxx
country: yyy
...
```

OSS: Graph Notebook

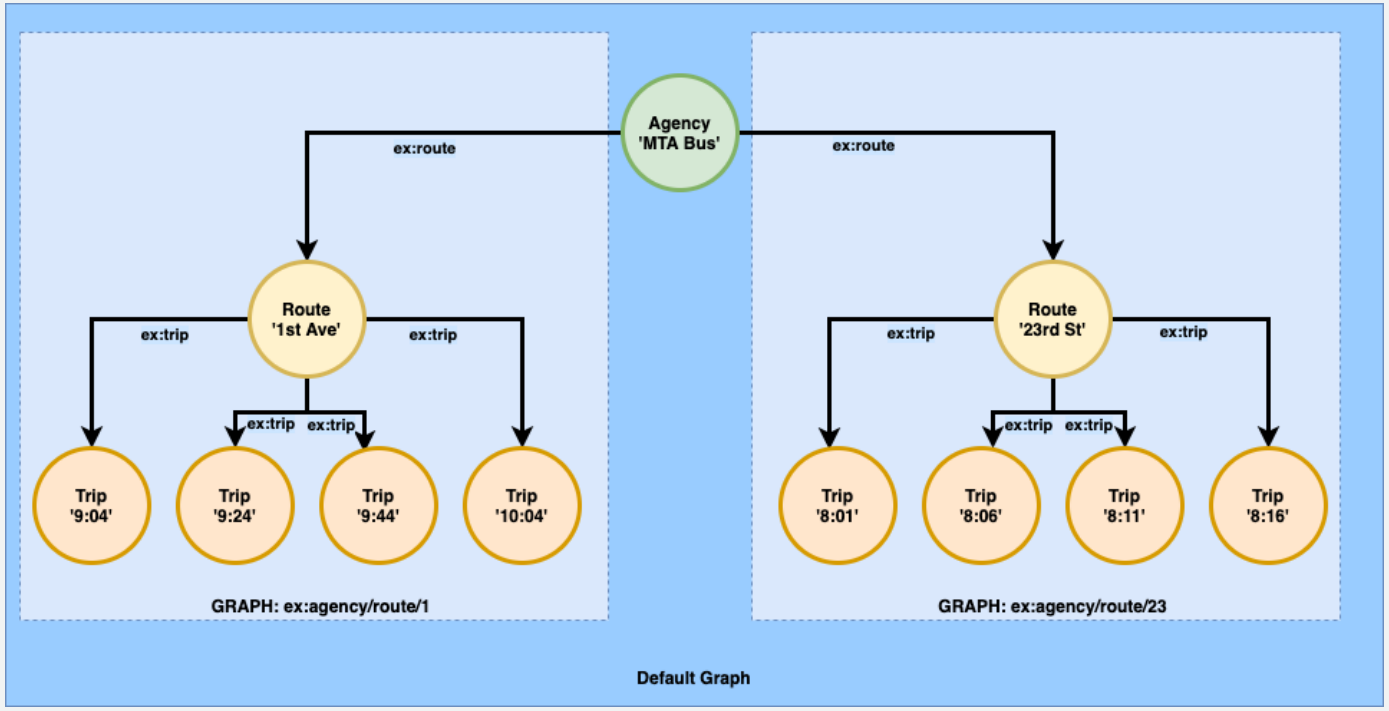
グラフ可視化をサポートする Python ベースの Jupyter ノートブック



https://github.com/aws/graph-notebook

SPARQL 1.1 Graph Storeプロトコルをサポート

- SPARQL で定義された GSP をサポート
- 名前付きグラフの更新に用いる
- Apache Jena など GSP に対応したツールで活用可能に

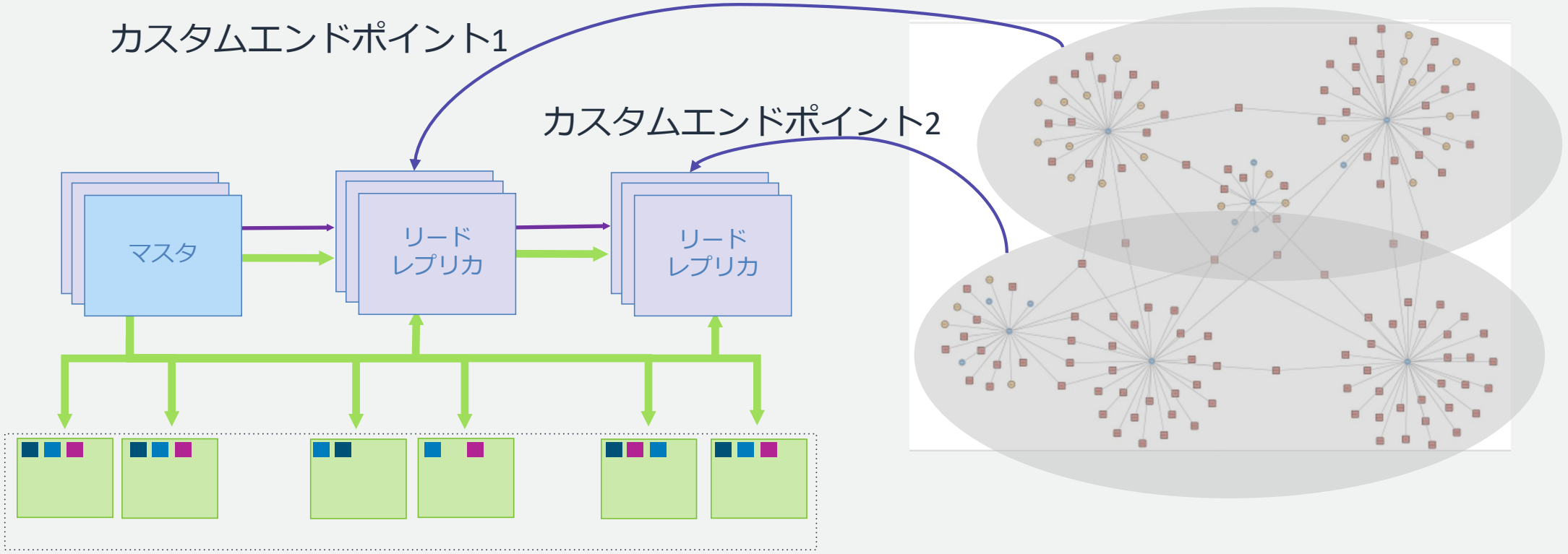


DFE エンジン

- 統計情報をベースに実行計画を組み立てる新エンジン
- CPU / メモリ / IOをより効率的に使いパフォーマンスの高いクエリ実行を狙う
  - DFE Statistics: 新たなデータが10%以上になったら or 10日ごとに更新
- ラボモードのため本番使用は推奨されない
- Left-depth や bushy など、既存のエンジンでは実装していないプランタイプなどを駆使する

リードレプリカ + カスタムエンドポイント

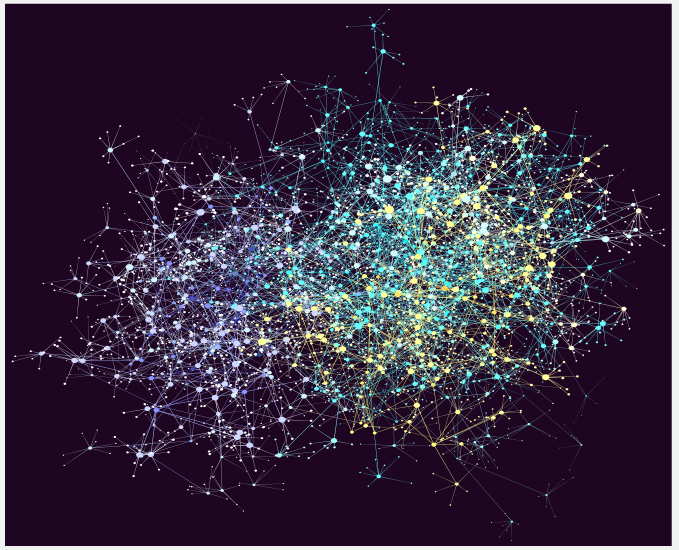
起点ノードのグラフ空間ごとにリードレプリカ集合を使い分ける



まとめ

このセッションでご紹介したこと

- Amazon Neptune ML
- OpenCypher サポート
- その他最新アップデート



Amazon Neptune ライブデモ

Learning Gremlin

ライブデモセッション

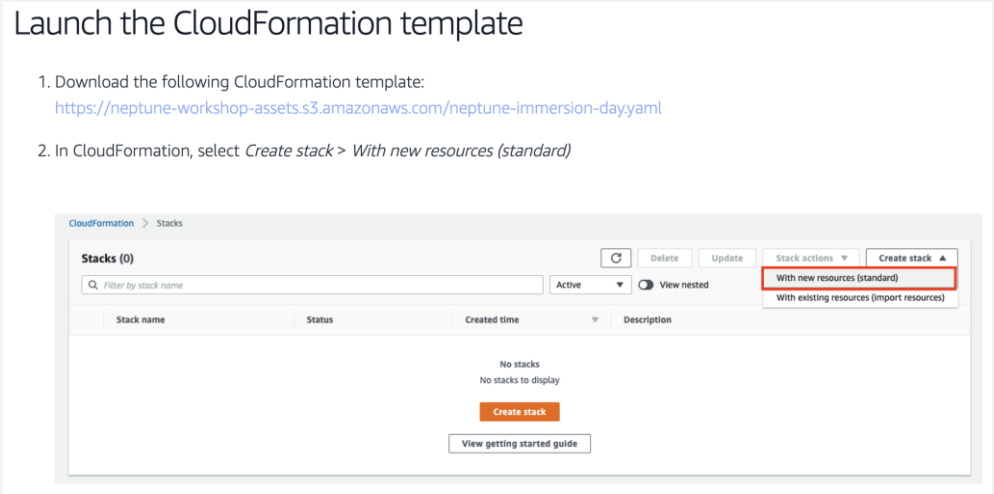
Neptune Deep Dive Workshop を
解説しながら実行していきます

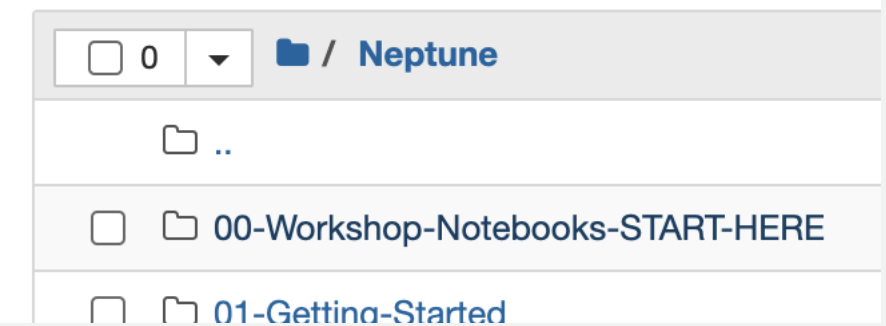
Gremlin と SPARQL の基礎を
ステップを踏んで学びましょう

BUILD YOUR FIRST GRAPH APPLICATION WITH AMAZON NEPTUNE

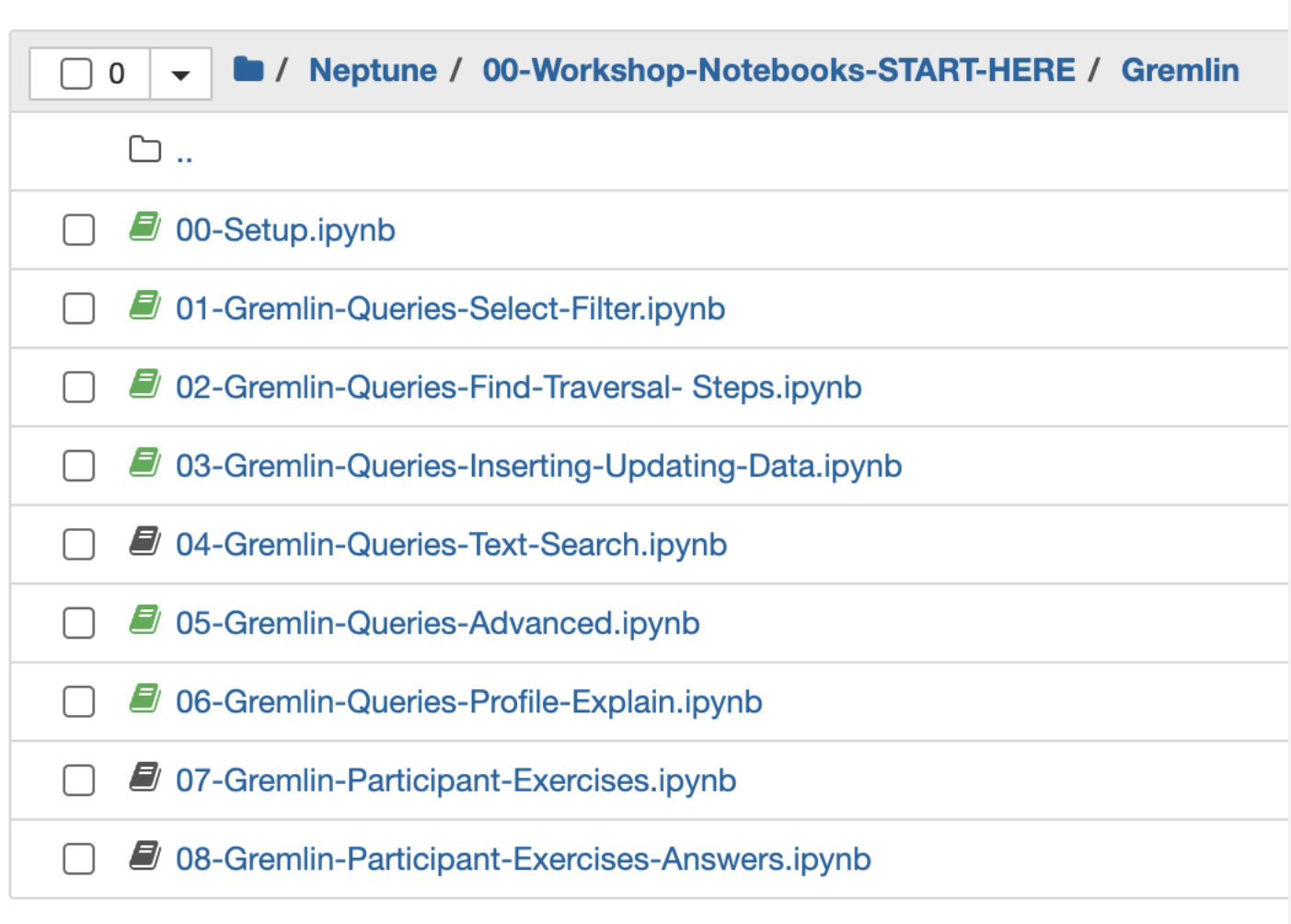
ワークショップを自身でも実行してみましょう！

- ワークショップの Appendix にある手順で Neptune インスタンスやノートブックが作成され、初期データロードも実行されます
- あとはノートブックから Workshop-Notebooks を開けばOK





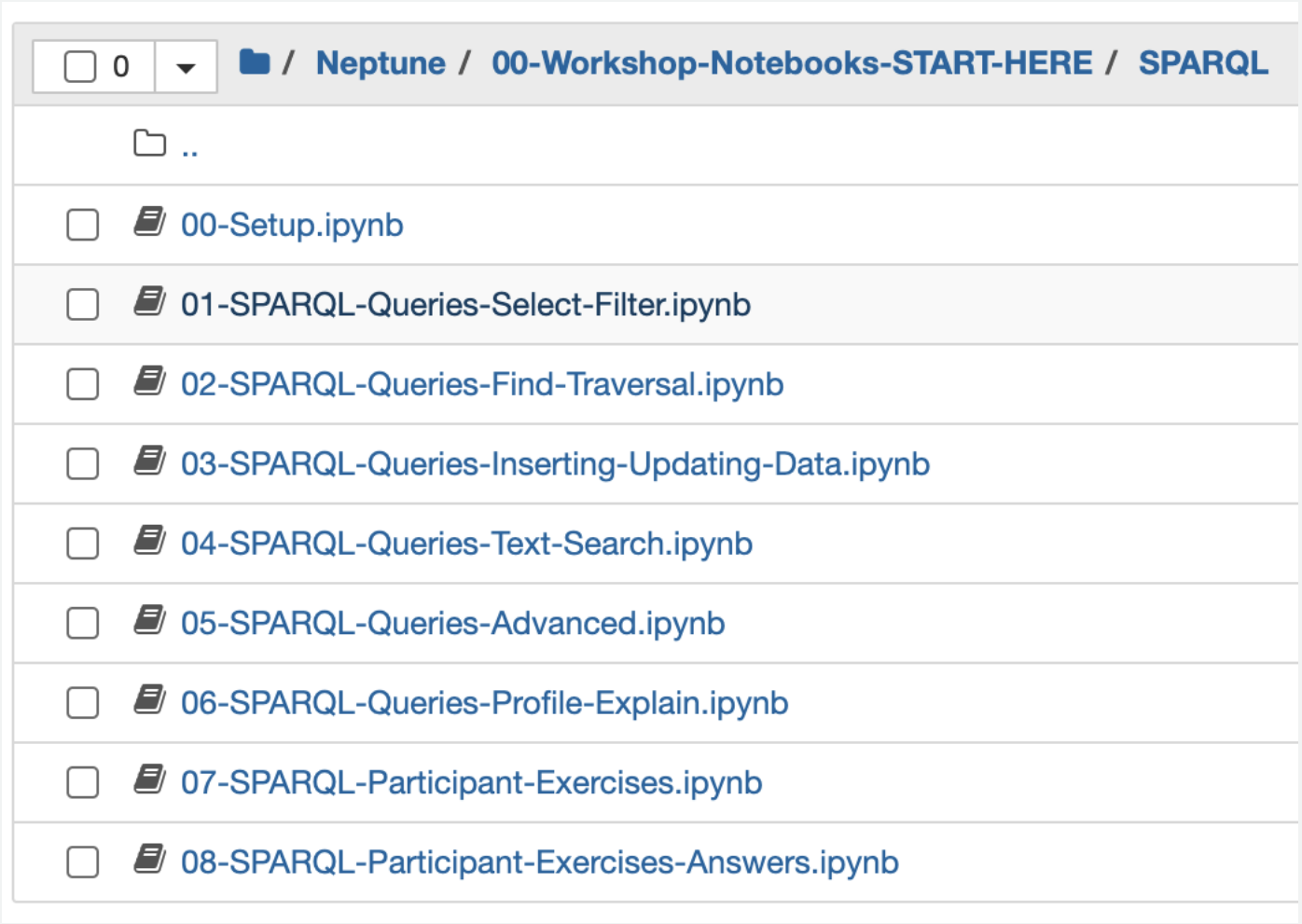
Gremlin Workshop のメニュー



Amazon Neptune ライブデモ

Learning SPARQL

SPARQL Workshop のメニュー



Thank you!