



Amazon Neptuneではじめる グラフデータベース入門

AWS Black Belt Online Seminar

Tatsuya Kimura

Senior Solutions Architect

2023/07

AWS Black Belt Online Seminarとは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWSの技術担当者が、AWSの各サービスやソリューションについてテーマごとに動画を公開します
- 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます
- 以下のURLより、過去のセミナー含めた資料などをダウンロードすることができます
- <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
- <https://www.youtube.com/playlist?list=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>

内容についての注意点

- 本資料では 2023 年 7 月時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<https://aws.amazon.com/>)にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます

自己紹介

名前：木村 達也（きむら たつや）

所属：アマゾン ウェブ サービス ジャパン合同会社

データベーススペシャリスト ソリューションアーキテクト

好きなAWSサービス：Amazon Aurora、Amazon Neptune



本セミナーの対象者

- グラフデータベースに興味を持っている方
- Amazon Neptuneに興味を持っている方

本セミナーの目的

- グラフデータベースの概要や使い方をご理解頂く
- Amazon Neptuneの特徴をご理解頂く

アジェンダ

1. グラフデータベースとは
2. Amazon Neptuneとは
3. グラフデータベースの使い方
4. グラフデータベースの学習方法

1. グラフデータベースとは

グラフデータベースとは何か?

- グラフデータベースは関係の保存やナビゲートするための設計されたデータベース
- ノード(Node, Vertex)は実世界のオブジェクト(Object)を表す
- エッジ(Edge)はオブジェクト間の関係を格納する
- プロパティ(属性)とラベル(種別)はノードとエッジの両方に追加することができる



グラフデータベースの代表的なユースケース



ソーシャル
ネットワーク



レコメンテーション



ナレッジ
グラフ



不正検知



ライフ
サイエンス

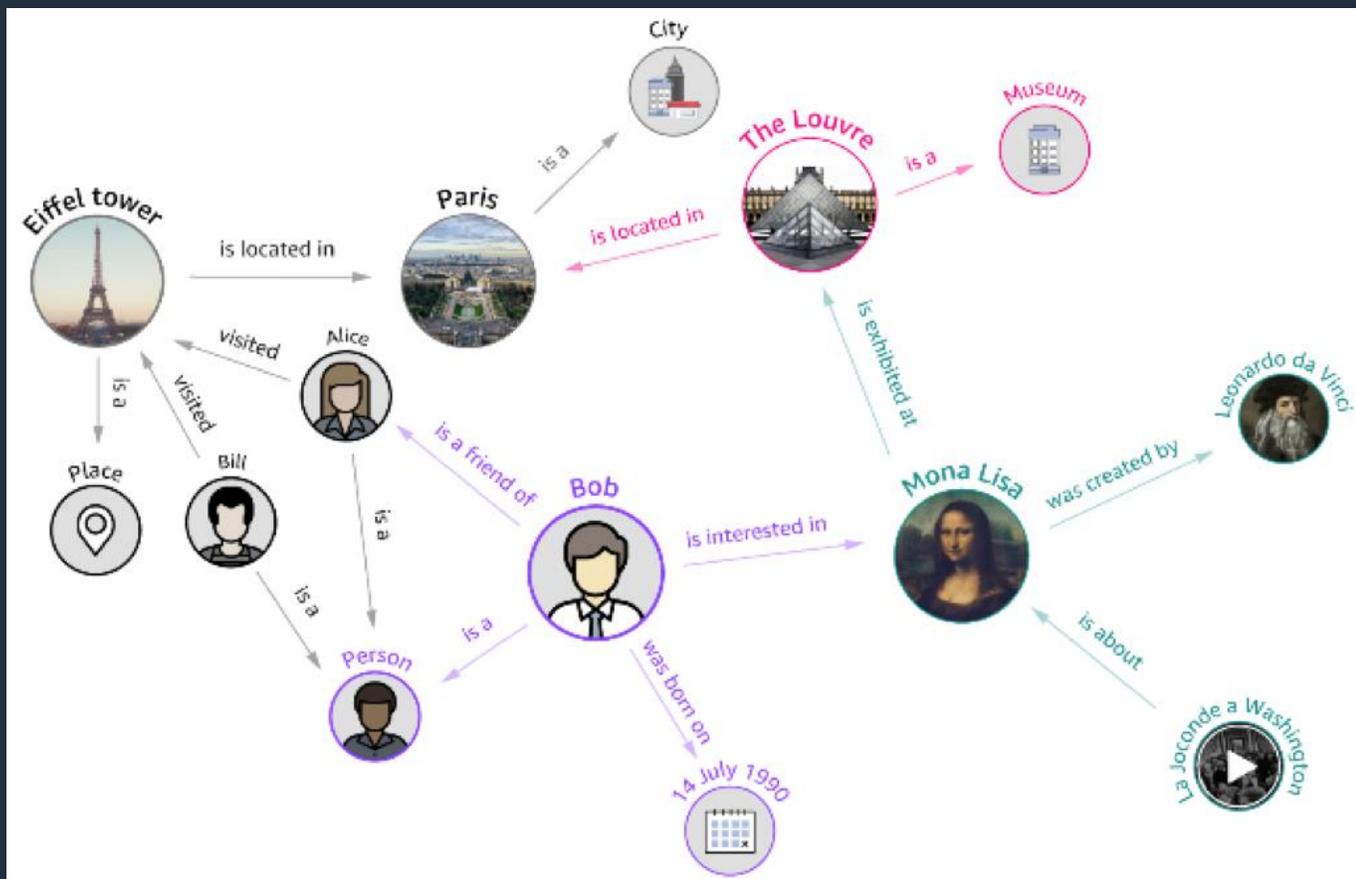


ネットワーク
IT運用

- 高度に接続されたデータドメイン
- 接続された構造を辿る
- 関係性の強さや重みなどに基づいて計算する

ユースケース例①：ナレッジグラフ

誰が、何を、いつ、どこで、を理解



利点

1. 異なるデータソースをリンク

異なる、不均質なデータソースをリンクして、隠れた接続を発見する

2. 検索結果の改善

検索の関連性を改善してデータに簡単にアクセスできるようにすることで、生産性を向上させる

3. ML/AIを強化する

関連するコンテンツでコンテキストを提供することで、機械学習モデルの効率と効果を向上させる

ユースケース例②：アイデンティティグラフ

Unified 360° View Of The Customer

利点

1. デバイスおよびチャネル間での永続ID

関連するすべてのデバイスとIDをリンクする永続的な識別子。一元的なプロフィールの作成、ターゲティング、パーソナライズを可能にする

2. 閲覧者の構築とセグメンテーション

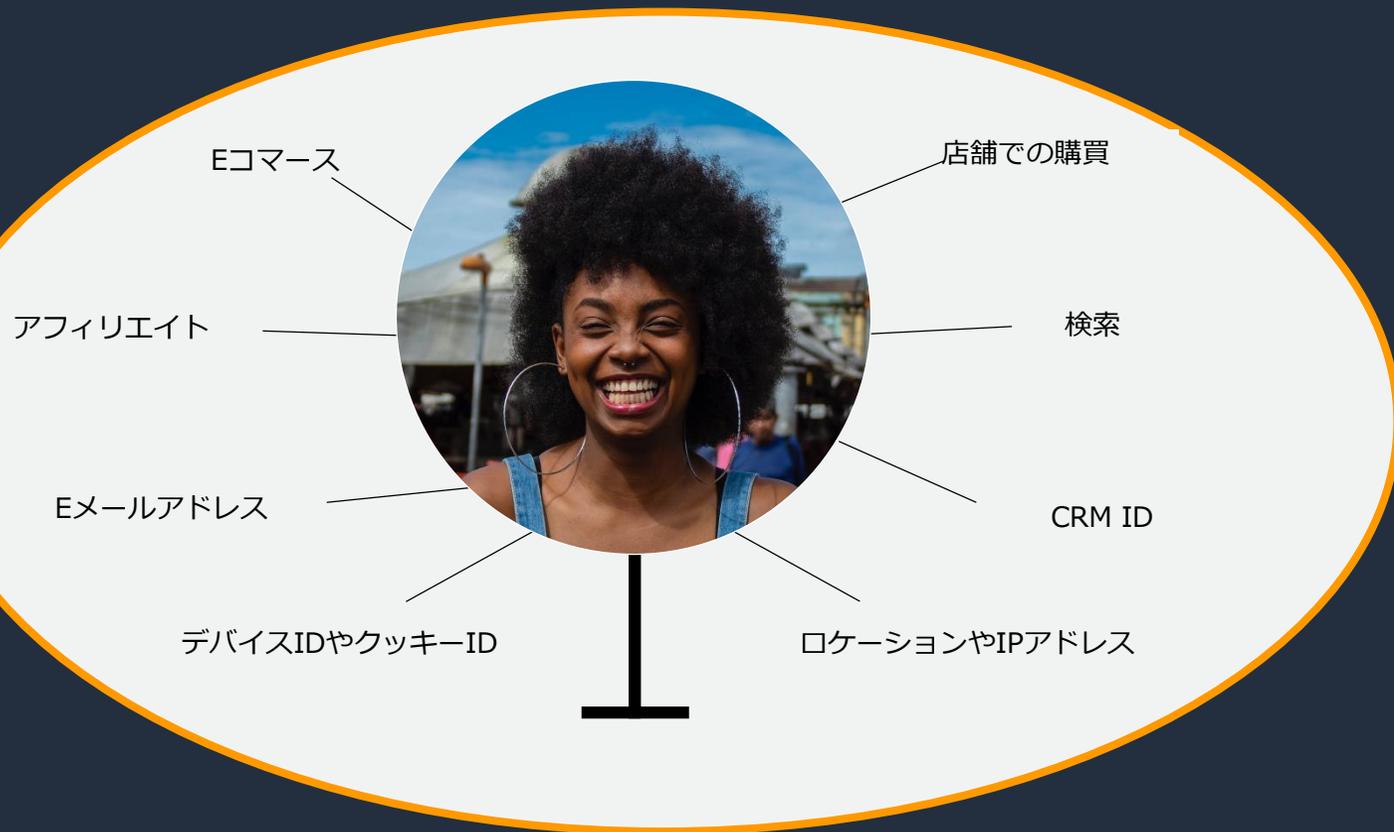
同様の興味、好み、購入の基づいて閲覧者を作成する

3. クロスデバイスの頻度キャッピング

閲覧者向けにデバイスとブラウザ全体の頻度のキャッピングを改善して、カスタマーエクスペリエンスを向上させる

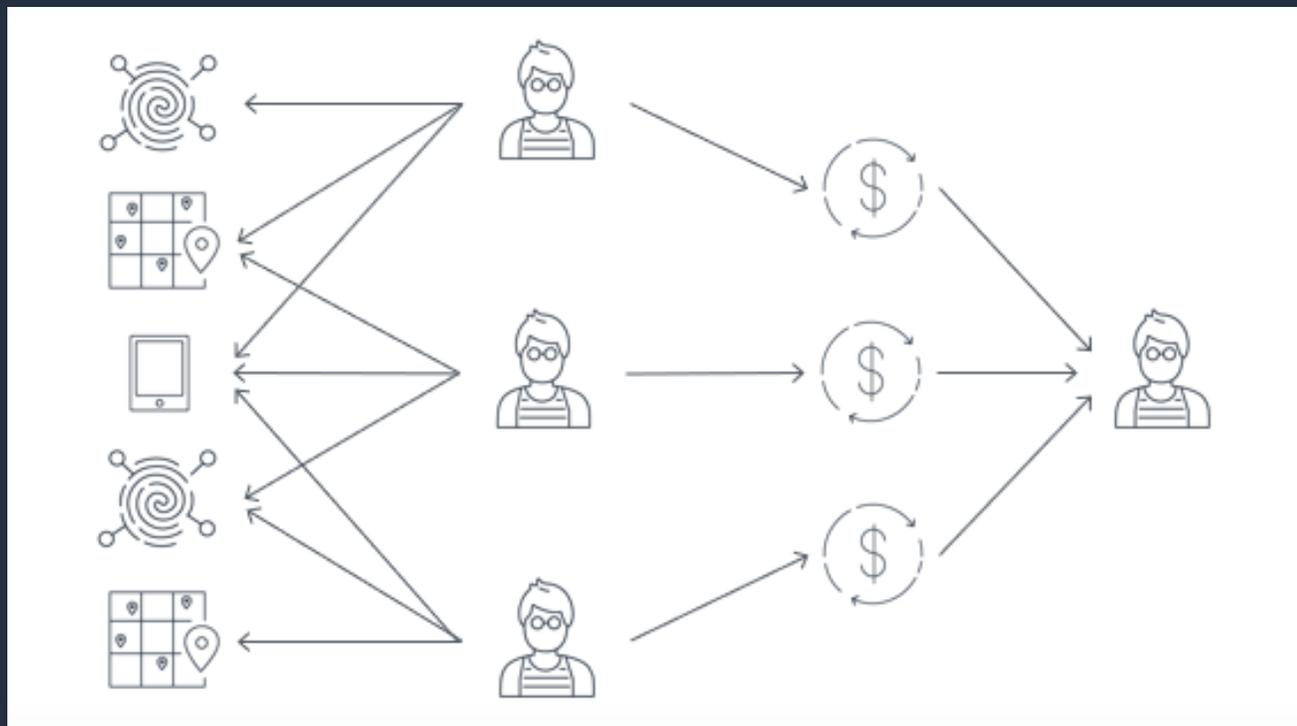
4. カスタマージャーニーの分析

エンドツーエンドの顧客行動を分析し、購入へのパスと属性を追跡する



ユースケース例③：不正検知グラフ

不正検出



利点

1. リアルタイムでの不正防止

トランザクションの不正パターンを特定し、発生後ではなく発生時に防止する

2. 独自のインサイト

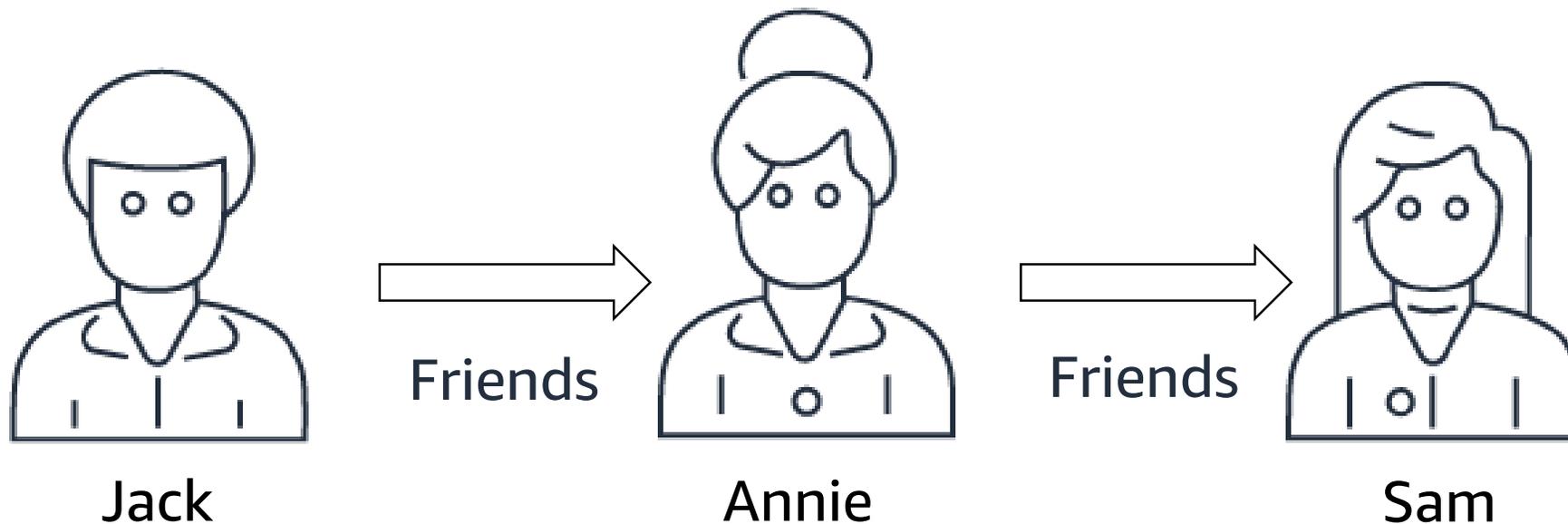
ユースケースは金融関連だけに留まらず、ゲームや種々の異常検知などに及ぶ

3. 変化への対応

接続されたデータの柔軟な性質を使用して、最小限の労力に変化する不正の性質に適応する

データモデルについて考えてみよう①

3人の関係性をどのようなデータモデルで実現するのがよいか？



Jackさんの友達の友達は誰？

データモデルについて考えてみよう②

リレーショナルモデルで実現する場合

Person	
Person ID	Name
P1	Jack
P2	Annie
P3	Sam

X

Friends				
Friend ID	Person ID1	Person Name1	Person ID2	Person Name2
E1	P1	Jack	P2	Annie
E2	P2	Annie	P3	Sam

=

Friends of a Friend						
Friend ID	Person ID1	Person Name1	Person ID2	Person Name2	Person ID3	Person Name3
E4	P1	Jack	P2	Annie	P3	Sam

データモデルについて考えてみよう③

グラフモデルで実現する場合



グラフモデル：①PROPERTY GRAPH

- firstNameがBobを探す

Gremlin

```
g.V().has('firstName', 'Bob').  
=>['Bob']
```

openCypher

```
MATCH (u:User)  
WHERE u.firstName="Bob"  
RETURN u.firstName  
=>['Bob']
```

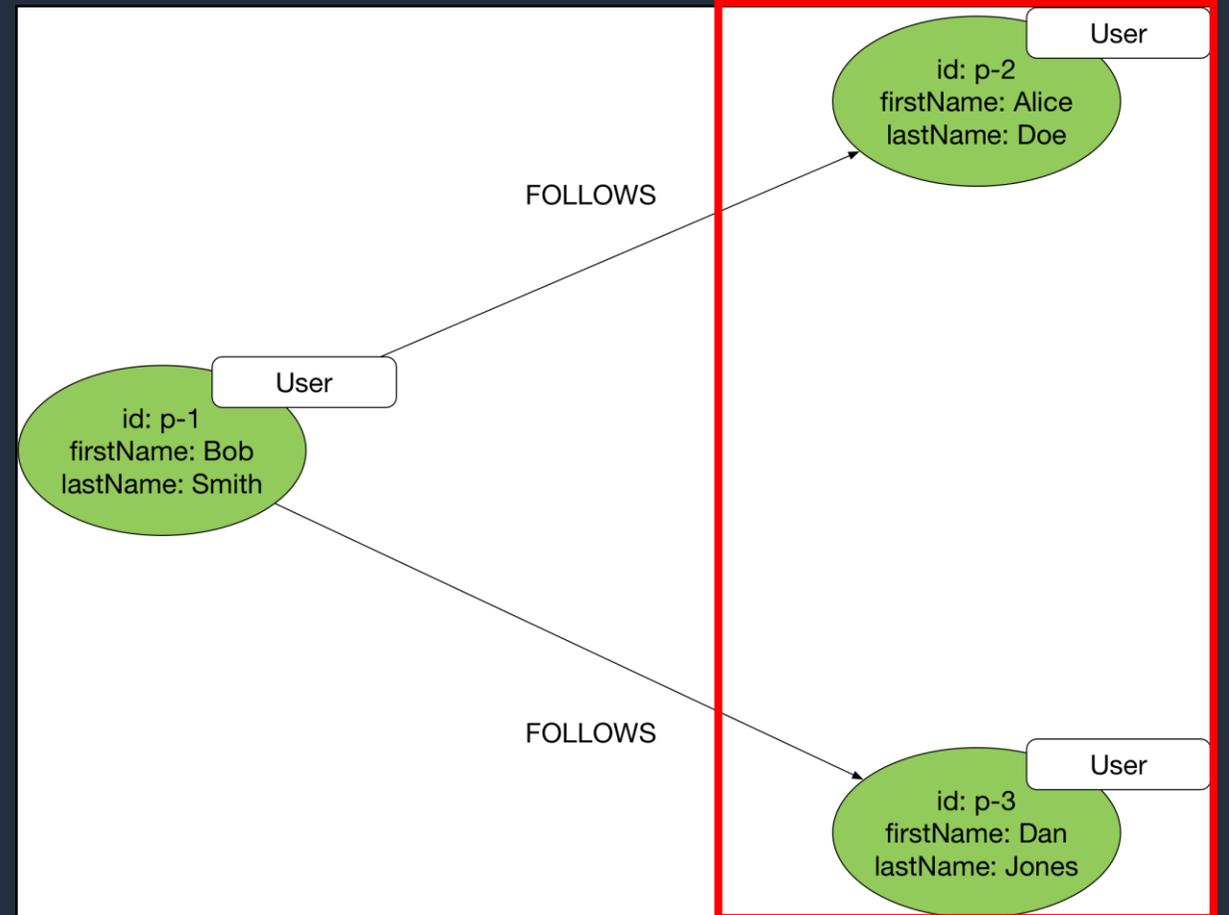
- firstName:BobがFOLLOWSしている人を探す

Gremlin

```
g.V().has('firstName', 'Bob').out('FOLLOWS')  
=>['Alice', 'Dan']
```

openCypher

```
MATCH (u:User)-[:FOLLOWS]->(f:User)  
WHERE u.firstName="Bob"  
RETURN f.firstName  
=>['Alice', 'Dan']
```

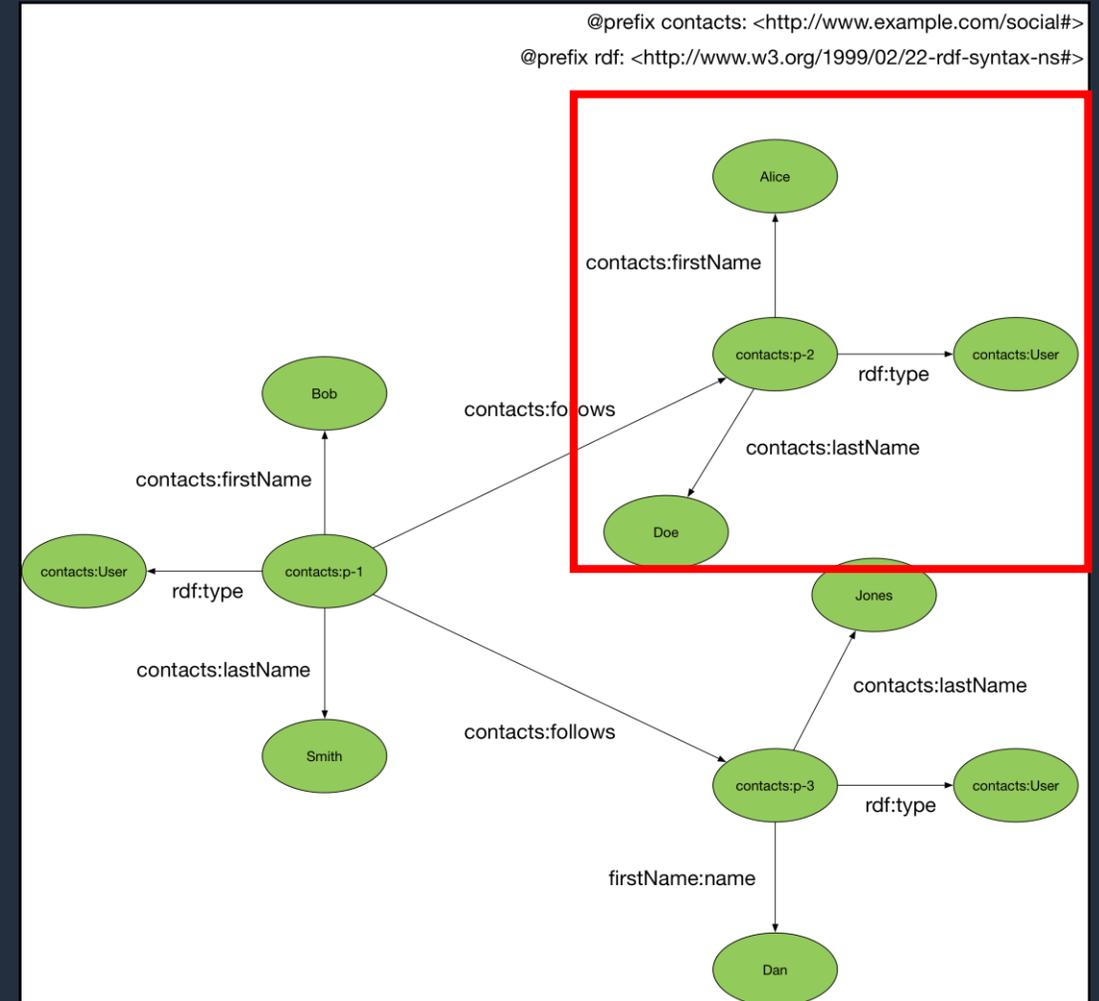


グラフモデル : ②Resource Description Framework(RDF)

@prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

@prefix contacts: <<http://www.example.com/social#>>

contacts:p-2	contacts:firstName	"Alice"
contacts:p-2	rdf:type	contacts:User.
主語(IRI)	述語(IRI)	目的語(IRI またはリテラル)



2. Amazon Neptuneの概要

Amazon Neptune



スケーラブル



セキュア・信頼性



サーバーレス



インテグレーション

スケーラブル



- ノードやエッジのもつ**プロパティに制限がない**
- **ミリ秒のレイテンシ**で数十億の関係を照会
- 最大15個までのリードレプリカ
- **128TiBまでスケール可能**なストレージ
- ストレージとコンピューートを分離することでコストを最適化

セキュア・信頼性



- 保存および転送時の**データ暗号化**
- 3つのAZにまたがって6つのレプリカを保存する**高可用性ストレージ**
- 10秒以内の**自動フェールオーバー**
- **自己修復、自動スケール**のストレージ
- ポイントインタイム・リカバリ(PITR) を備えた自動で継続的なバックアップ

サーバーレス

- **瞬時に数十万のクエリ**までスケール
- ピークによるキャパシティプランニングと比較してデータベースコストを最大90%節約
- **キャパシティ管理が不要**



Amazon
Neptune
Serverless
(db.serverless)

インテグレーション



- Amazon CloudWatch や CloudTrail からの
モニタリング
- AWS Lambda との連携による
イベントドリブンのファンクションの実装
- Amazon OpenSearch による**全文検索**

サポートするグラフモデルとクエリ言語

PROPERTY GRAPH

Apache TinkerPop™
Gremlin Traversal Language



openCypher Query Language

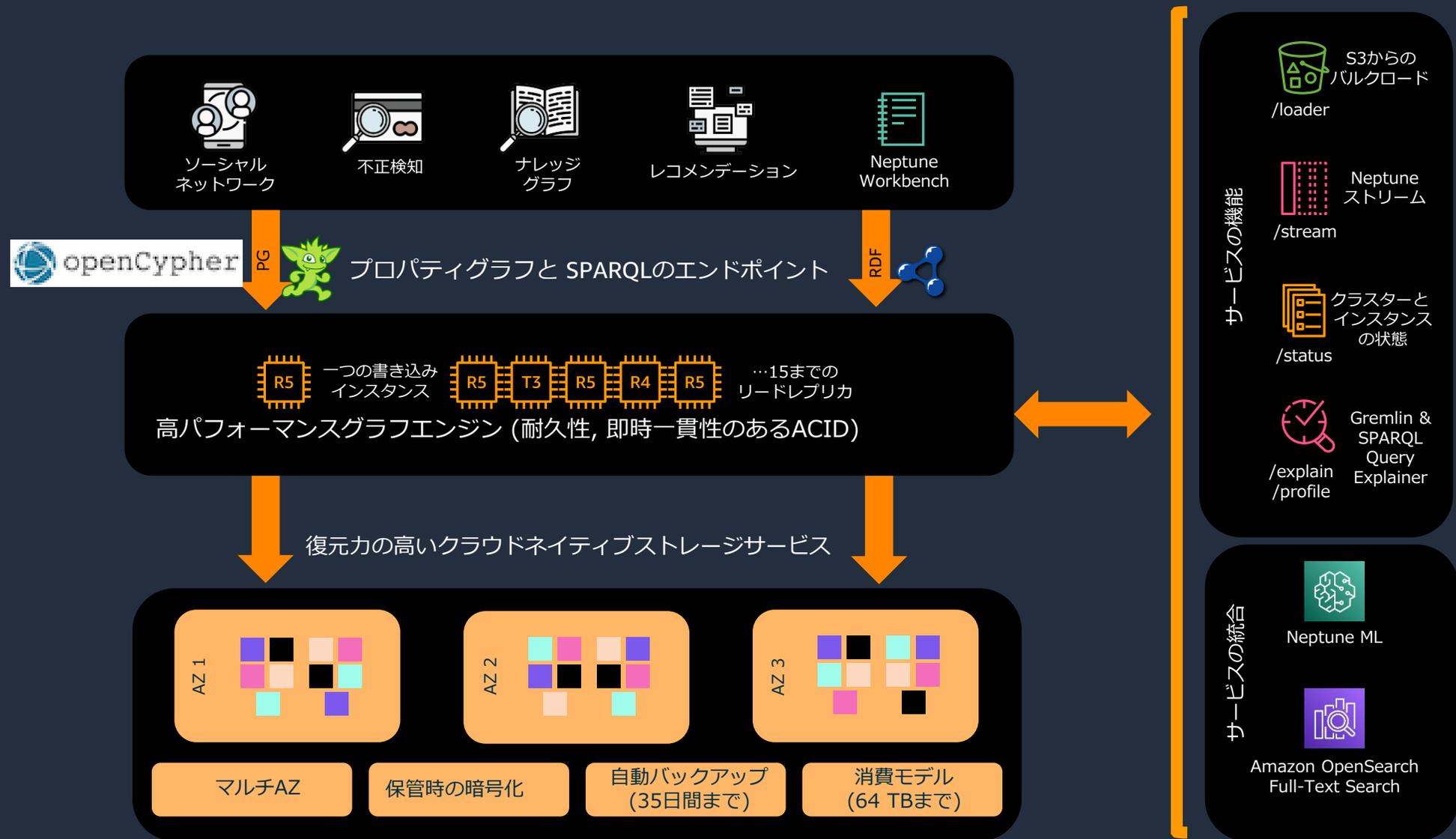


RESOURCE DESCRIPTION FRAMEWORK (RDF)

W3C Standard
SPARQL Query Language

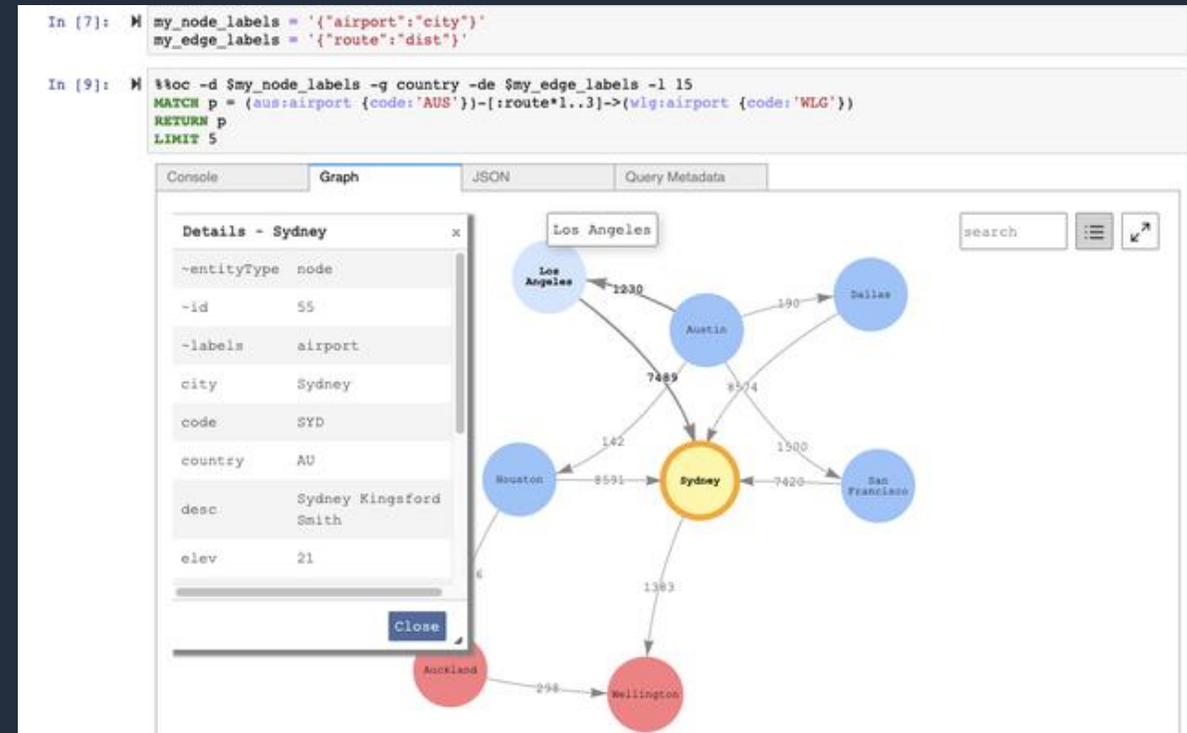


ハイレベルアーキテクチャ



Neptune Workbench (Jupyter Notebook)

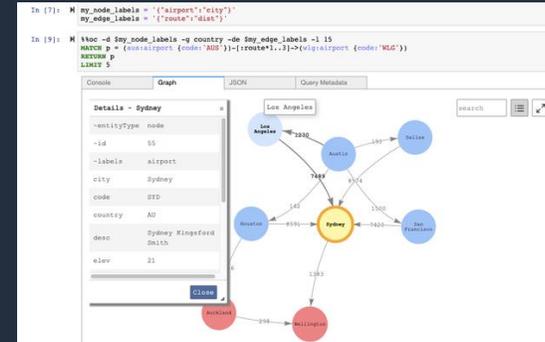
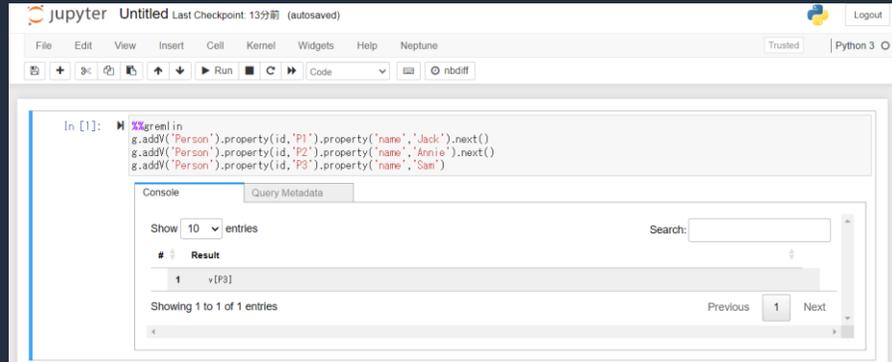
- Neptuneを使用するのに便利な Jupyter Notebook
 - Neptune インスタンスを指定するだけで簡単に使用開始
- OSSの [graph-notebook](#) がデプロイされる
- クエリの実行による試行錯誤が直感的に行える
 - “%%gremlin”、“%%oc”や“%%sparql”を記述したセルでクエリを実行
- 可視化も実装されている



3. グラフデータベースの使い方

前提

- Neptune WorkbenchからNeptuneに対してクエリを実行する



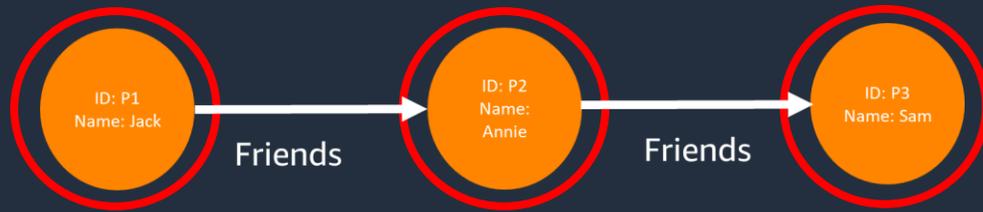
- クエリ言語はプロパティグラフ用のGremlinとopenCypherを使用する
- 以下のデータモデルを使用し、データの登録と探索を行う



データの登録① : Gremlin

ノードの登録

イメージ (赤枠が登録データ)



実行クエリ

```
%%gremlin
g.addV().property(id,'P1').property('name','Jack').next()
g.addV().property(id,'P2').property('name','Annie').next()
g.addV().property(id,'P3').property('name','Sam')
```

Neptune Workbench 実行結果

```
%%gremlin
g.addV().property(id,'P1').property('name','Jack').next()
g.addV().property(id,'P2').property('name','Annie').next()
g.addV().property(id,'P3').property('name','Sam')
```

#	Result
1	v[P3]

エッジの登録

イメージ (赤枠が登録データ)



実行クエリ

```
%%gremlin
g.V('P1').addE('Friends').to(__.V('P2')).next()
g.V('P2').addE('Friends').to(__.V('P3'))
```

Neptune Workbench 実行結果

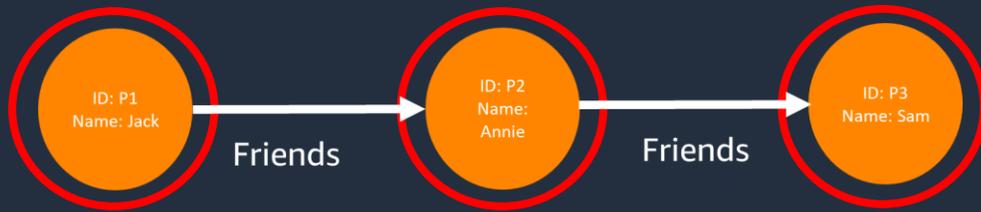
```
%%gremlin
g.V('P1').addE('Friends').to(__.V('P2')).next()
g.V('P2').addE('Friends').to(__.V('P3'))
```

#	Result
1	e[20c47e33-b450-b23a-3e59-f535cc6c2632][P2-Friends->P3]

データの登録② : openCypher

ノードの登録

イメージ (赤枠が登録データ)



実行クエリ

```
%%oc
CREATE (:User{id:"P1",Name:"Jack"}),
(:User{id:"P2",Name:"Annie"}), (:User{id:"P3",Name:"Sam"})
```

Neptune Workbench 実行結果

```
%%oc
CREATE (:User[id:"P1",Name:"Jack"]), (:User[id:"P2",Name:"Annie"]), (:User[id:"P3",Name:"Sam"])

JSON Query Metadata

{
  "results": []
}
```

エッジの登録

イメージ (赤枠が登録データ)



実行クエリ

```
%%oc
MATCH (a:User{Name:"Jack"}),(b:User{Name:"Annie"}),(c:User{Name:"Sam"})
CREATE (a)-[:Friends]->(b)-[:Friends]->(c)
```

Neptune Workbench 実行結果

```
%%oc
MATCH (a:User{Name:"Jack"}),(b:User{Name:"Annie"}),(c:User{Name:"Sam"})
CREATE (a)-[:Friends]->(b)-[:Friends]->(c)

JSON Query Metadata

{
  "results": []
}
```

データの探索① : Gremlin

Jackさんの友達を探索

イメージ (赤枠が探索したデータ)



実行クエリ

```
%%gremlin  
g.V('P1').out('Friends').valueMap()
```

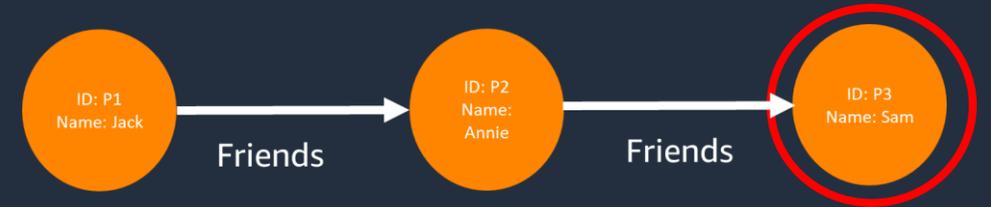
Neptune Workbench 実行結果 (赤枠が探索したデータ)

```
%%gremlin  
g.V('P1').out('Friends').valueMap()
```

#	Result
1	[{"name": ["Annie"]}]

Jackさんの友達の友達を探索

イメージ (赤枠が探索したデータ)



実行クエリ

```
%%gremlin  
g.V('P1').out('Friends').out('Friends').valueMap()
```

Neptune Workbench 実行結果 (赤枠が探索したデータ)

```
%%gremlin  
g.V('P1').out('Friends').out('Friends').valueMap()
```

#	Result
1	[{"name": ["Sam"]}]

データの探索② : openCypher

Jackさんの友達を探索

イメージ (赤枠が探索したデータ)



実行クエリ

```
%%oc
MATCH (a:User)-[:Friends]->(b)
WHERE a.id="P1"
RETURN b.Name
```

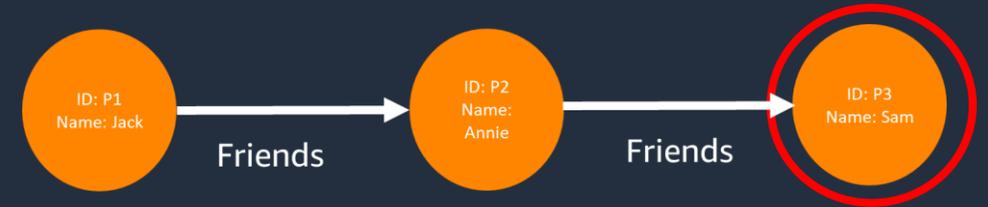
Neptune Workbench 実行結果 (赤枠が探索したデータ)

```
%%oc
MATCH (a:User)-[:Friends]->(b)
WHERE a.id="P1"
RETURN b.Name
```

#	b.Name
1	Annie

Jackさんの友達の友達を探索

イメージ (赤枠が探索したデータ)



実行クエリ

```
%%oc
MATCH (a:User)-[:Friends*2]->(c)
WHERE a.id="P1"
RETURN c.Name
```

Neptune Workbench 実行結果 (赤枠が探索したデータ)

```
%%oc
MATCH (a:User)-[:Friends*2]->(c)
WHERE a.id="P1"
RETURN c.Name
```

#	c.Name
1	Sam

4. グラフデータベースの学習方法

学習パスの例① : Gremlin

- Neptune Workbenchのサンプルノートブックを実行してみる
- AWS SamplesのTutorialに沿ってクエリを実行してみる
<https://github.com/aws-samples/amazon-neptune-samples/tree/master/gremlin>
- TinkerPopのTutorialで学習する
<https://tinkerpop.apache.org/docs/current/tutorials/getting-started/>
- Gremlinに関連するAWSブログを参照する
<https://aws.amazon.com/jp/blogs/news/queries-with-neptune-preparation/>
<https://aws.amazon.com/jp/blogs/news/queries-with-neptune-gremlin/>
- Neptuneにおける実装での相違点に留意しながら実行する
https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/access-graph-gremlin-differences.html
- クエリを少しずつ実行および調整しながら学ぶことが効果的

学習パスの例②：openCypher

- Neptune Workbenchのサンプルノートブックを実行してみる
- openCypher構文のチートシートを参照する

<https://github.com/aws-samples/amazon-neptune-samples/blob/master/opencypher/Cheatsheet.md>

- Cypherクエリ言語リファレンスを参照する

<https://opencypher.org/resources/>

- クエリを少しずつ実行および調整しながら学ぶことが効果的

学習パスの例③：RDF/SPARQL

- Neptune Workbenchのサンプルノートブックを実行してみる
- 公開されたグローバルなナレッジベースを用いてSPARQLを学習
<http://ja.dbpedia.org/sparql>
- クエリを少しずつ実行および調整しながら学ぶことが効果的

Neptune Getting Started Resources

- Website: <https://aws.amazon.com/neptune/>
- Technical Documentation: <https://docs.aws.amazon.com/neptune/>
- Customer Stories: <https://aws.amazon.com/neptune/developer-resources/>
- Re:Post: <https://repost.aws/search/questions?globalSearch=neptune>
- AWS Databases & Analytics on LinkedIn: [linkedin.com/showcase/aws-databases](https://www.linkedin.com/showcase/aws-databases)



Browse Amazon
Neptune
resources

本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料のAWSサポート窓口へお問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください
#awsblackbelt

その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!