



[AWS Black Belt Online Seminar]

Amazon Redshift

サービスカットシリーズ

Partner Solutions Architect
大林 加奈子
2020/3/18

AWS 公式 Webinar
<https://amzn.to/JPWebinar>



過去資料
<https://amzn.to/JPArchive>



自己紹介

大林 加奈子 (Kanakano Obayashi)

アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト



好きな AWS のサービス:
Analytics 関連のサービス

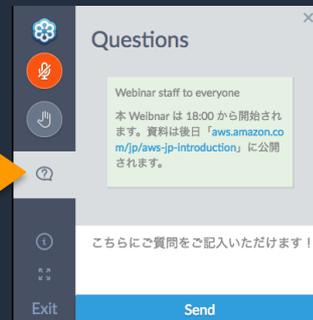
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾンウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
お答えできませんのでご了承下さい

- ① 吹き出しをクリック
- ② 質問を入力
- ③ Send をクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では 2020 年 3 月 18 日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 (<http://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。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

本日のアジェンダ

- Amazon Redshift 概要
- Amazon Redshift アーキテクチャ
- Amazon Redshift の特長
- まとめ

Amazon Redshift 概要

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>



Amazon Redshift とは？

高速、スケーラブルで費用対効果の高い
データウェアハウスおよび
データレイク分析マネージドサービス



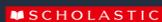
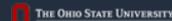
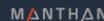
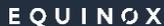
Amazon Redshift

数万社のお客様が Amazon Redshift を利用しています



Amazon Redshift

多くのお客様に活用されている
クラウドデータウェアハウス



Amazon Redshift のユースケース

主に大容量データを高速に集計・分析する必要があるワークロードに活用



経営ダッシュボード



定型レポートニング



アドホック分析



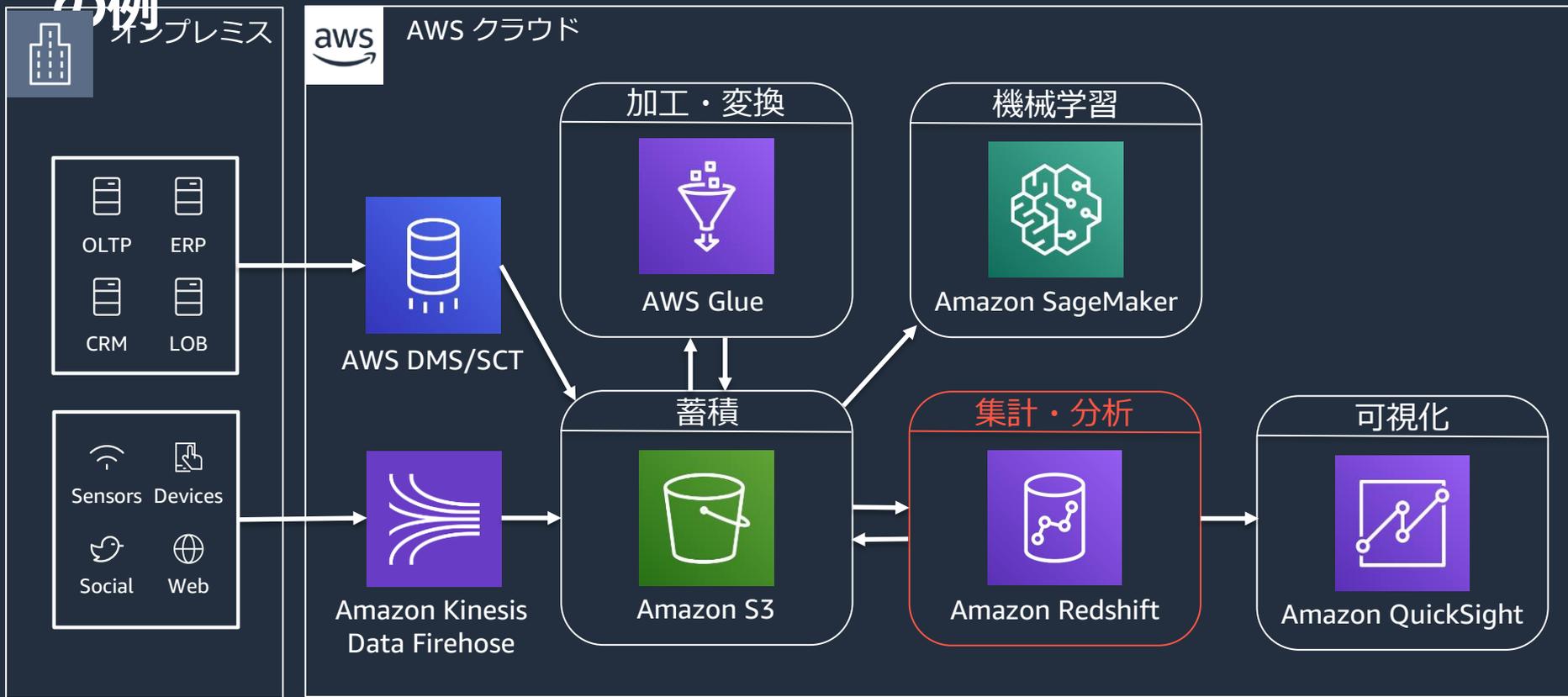
ETL/バッチ



機械学習の前処理

Amazon Redshift を中心としたデータ分析パイプライン

の例



Amazon Redshift アーキテクチャ

AWS 公式 Webinar

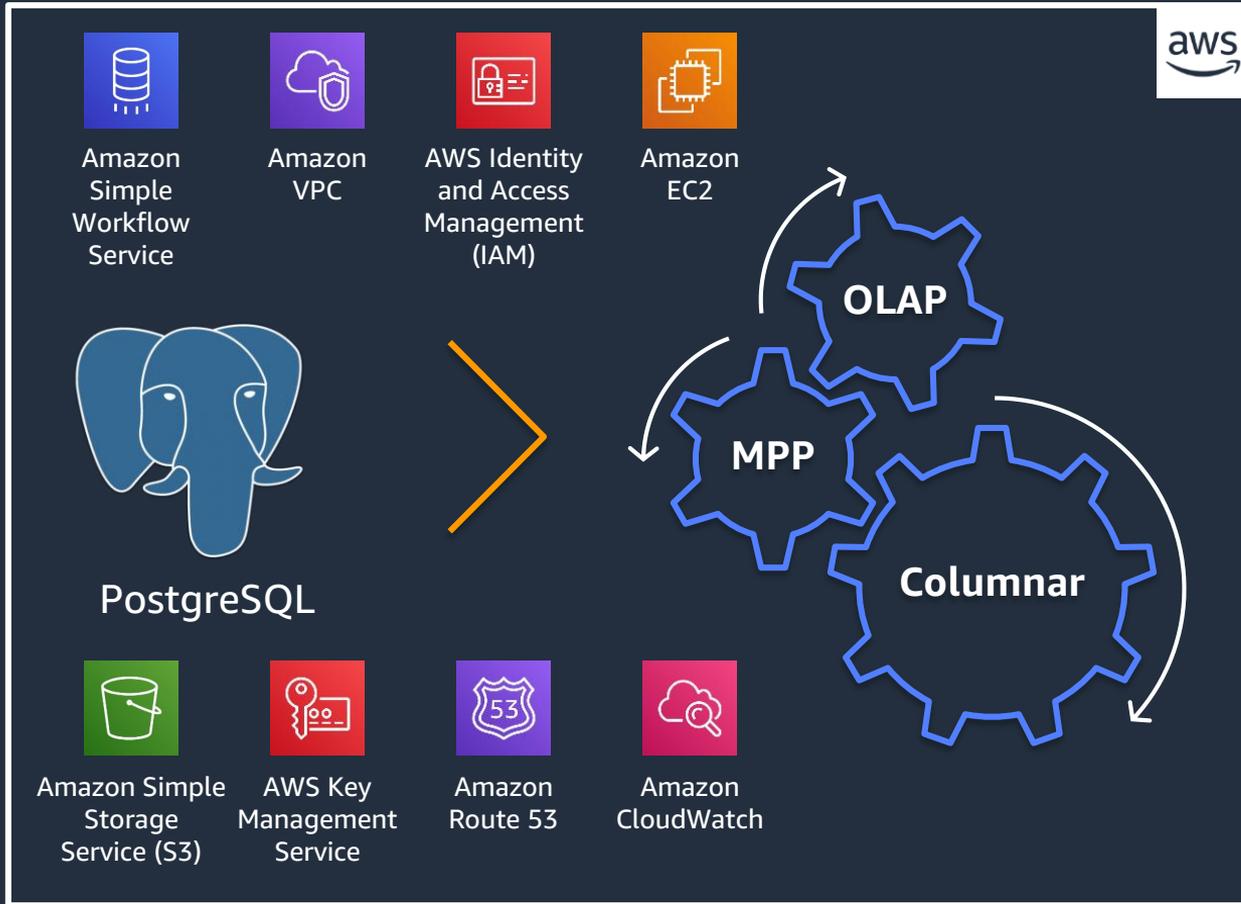
<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>





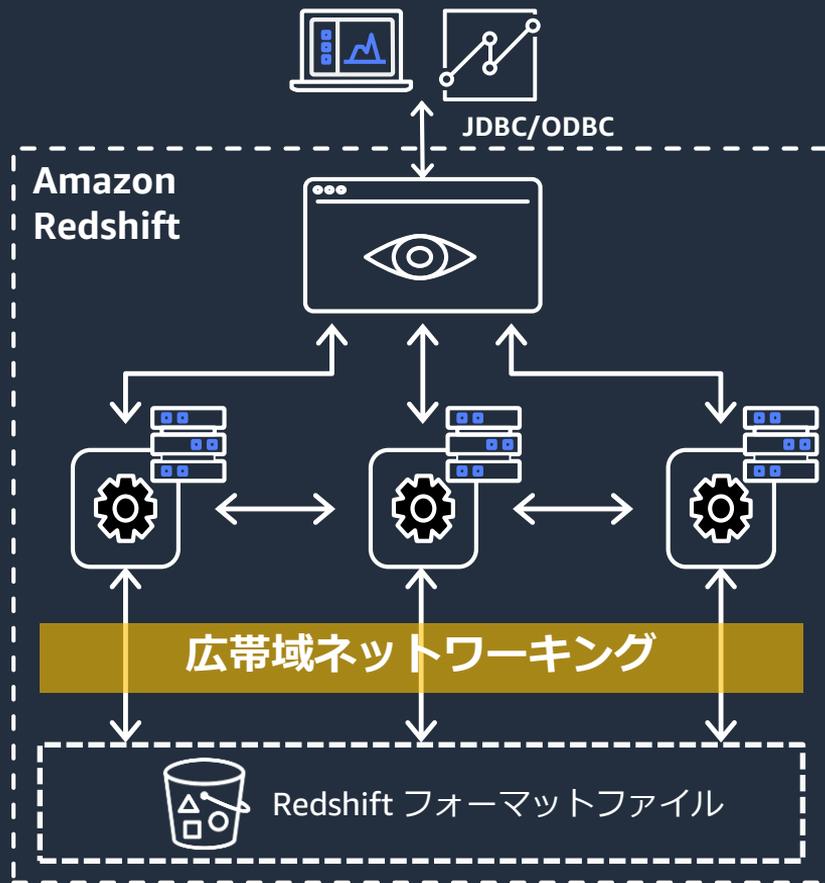
Amazon Redshift のアーキテクチャ (RA3インスタンス)

リーダーノード

- クエリのエンドポイント
- SQL 処理コードの生成と展開

マネージドストレージ

- Redshift 管理 S3 バケット

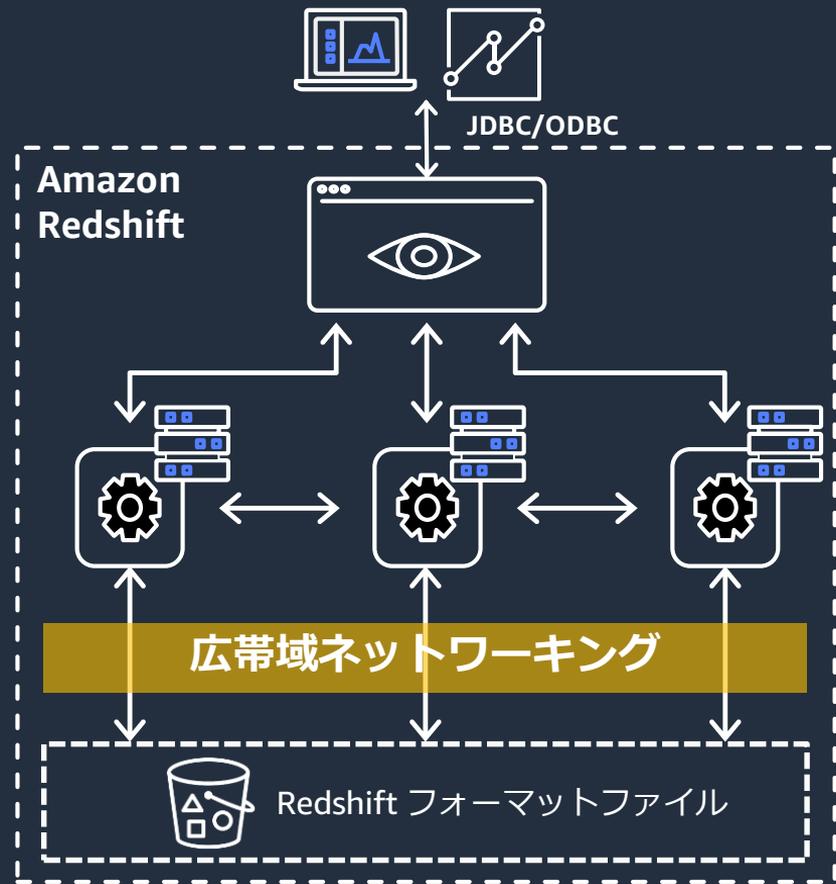


コンピューターノード

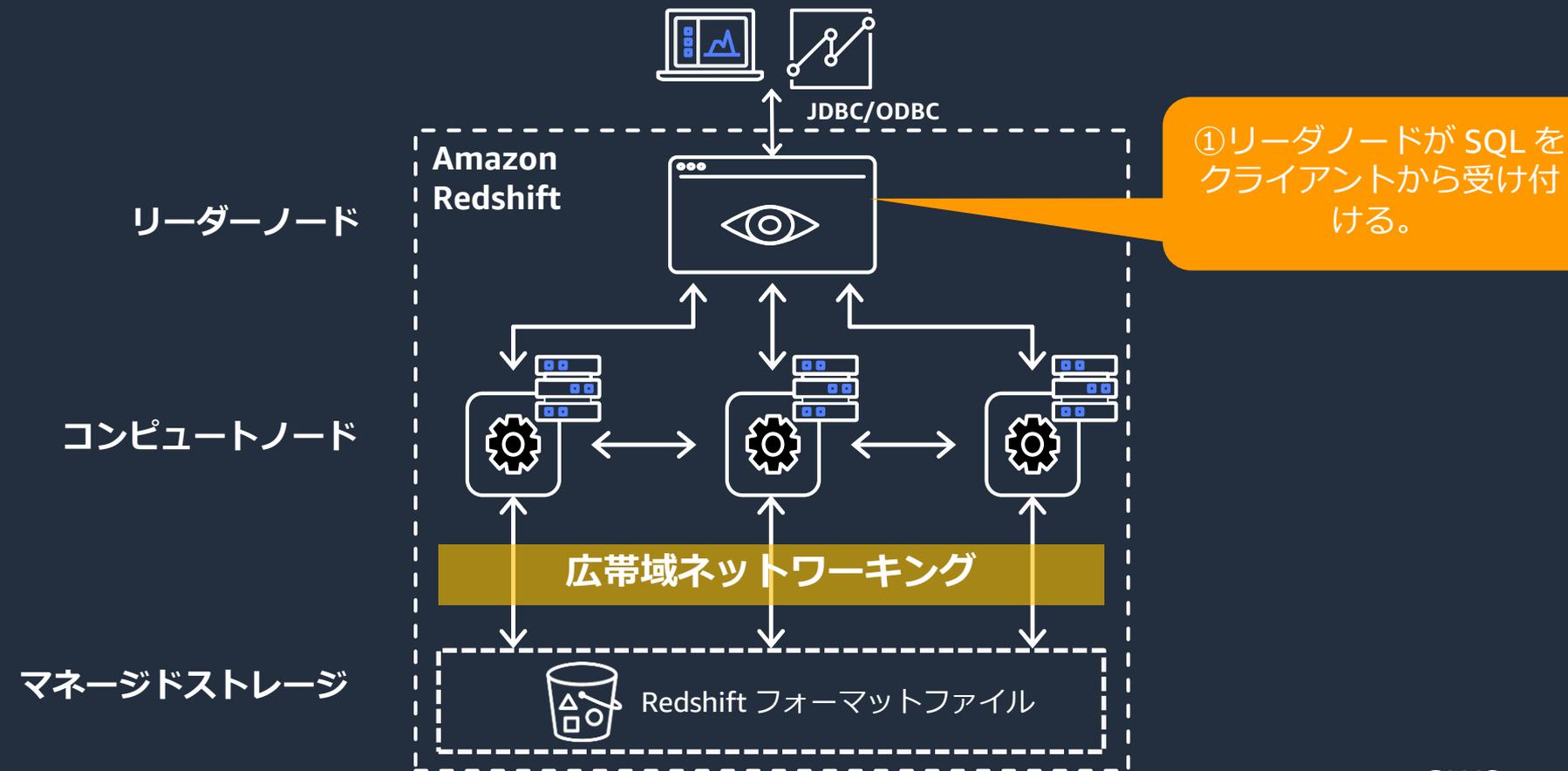
- 高速ローカル SSD キャッシュ
+ 大容量 RAM 搭載
+ 広帯域ネットワーキング
- クエリの並列実行

Amazon Redshift のアーキテクチャ (RA3インスタンス)

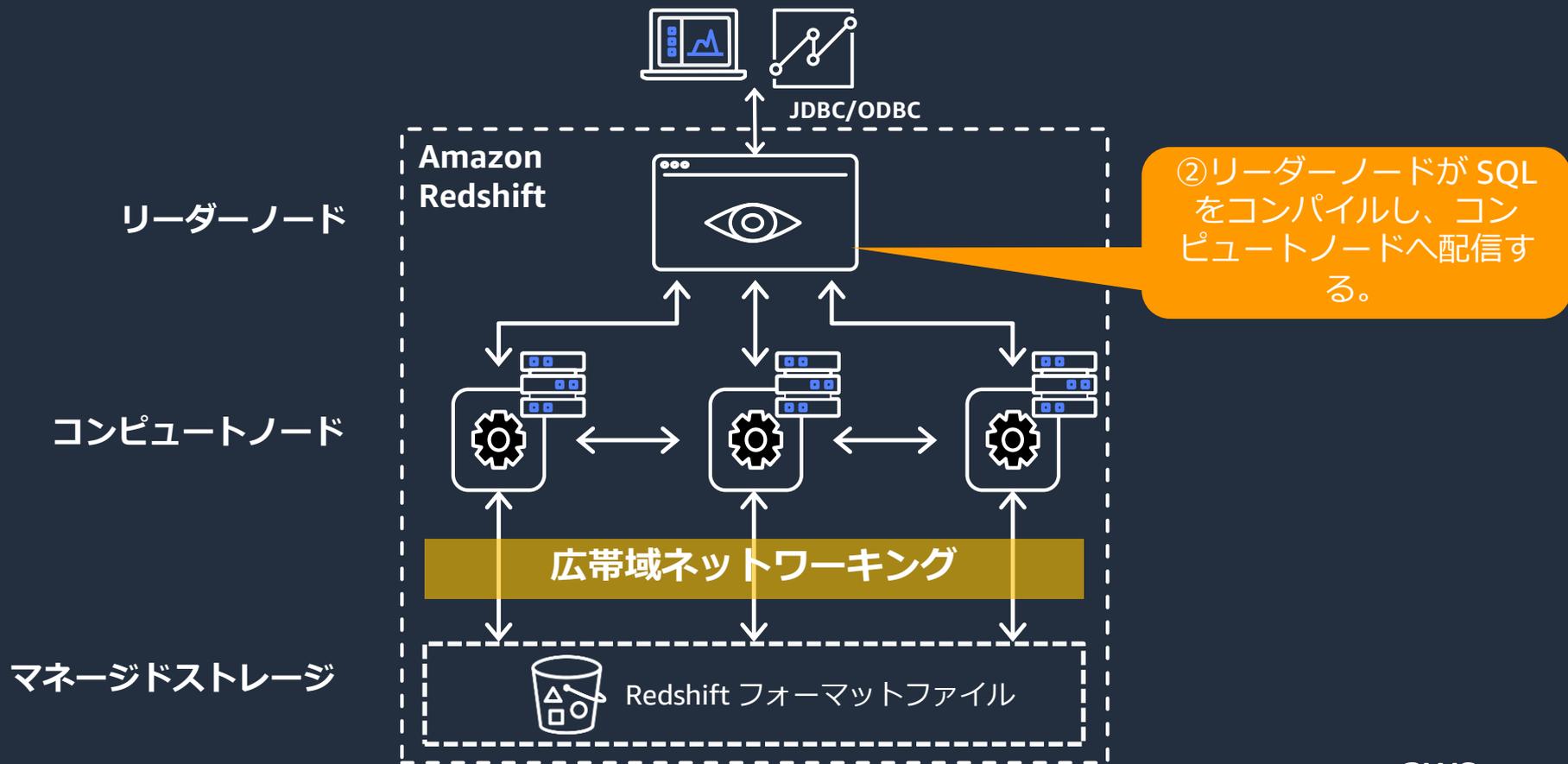
- コンピュートとストレージを分離しスケーリングと支払いを独立
- データは、永続ストレージとしての S3 とキャッシュとしてのローカル SSD に格納される
- アクセス頻度の高いブロックはキャッシュにとどまり、あまりアクセスされないブロックは自動的にキャッシュアウト



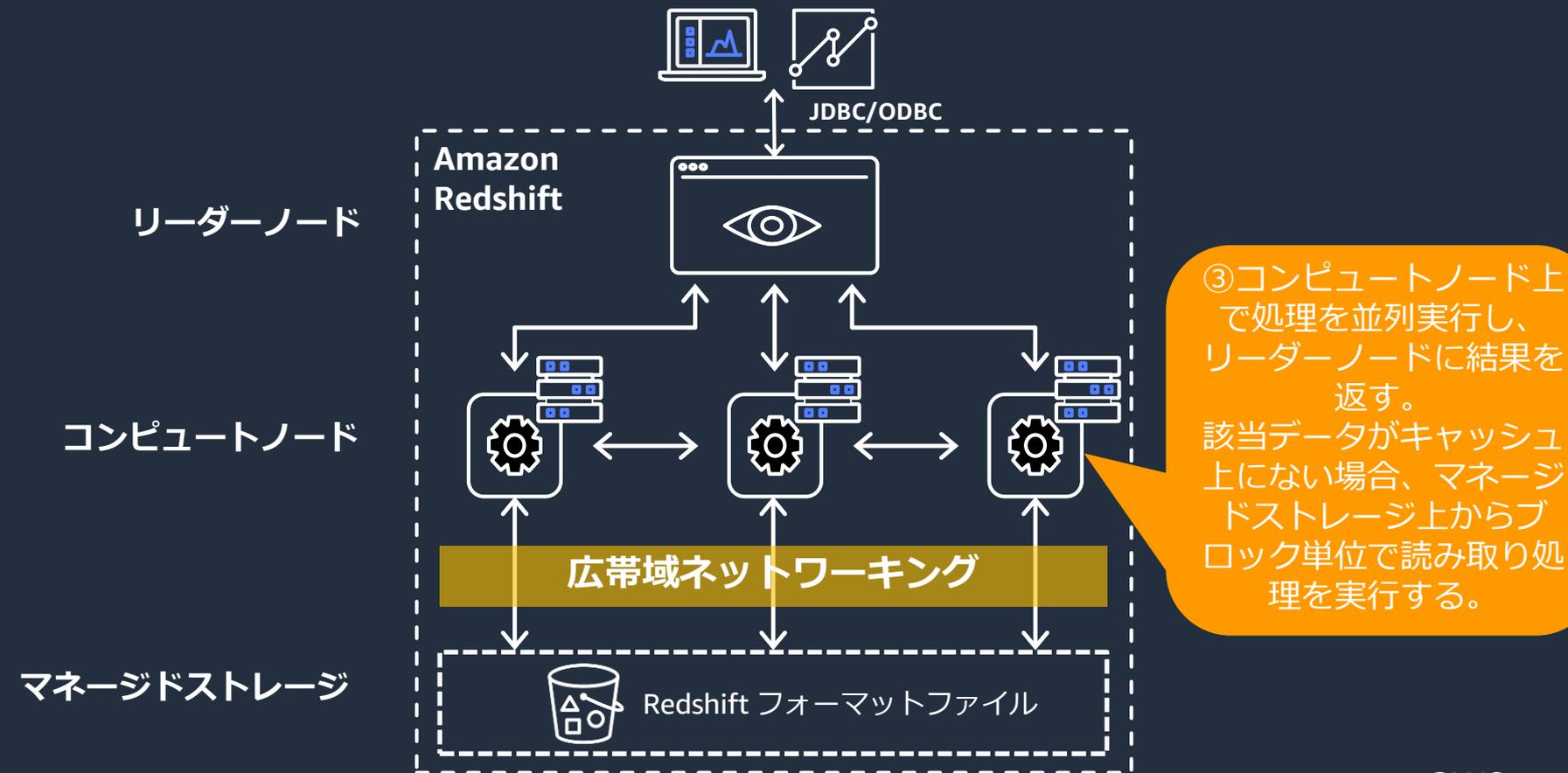
クエリの実行 (1/4)



クエリの実行 (2/4)



クエリの実行 (3/4)



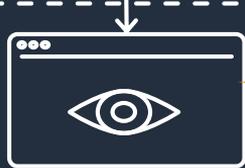
クエリの実行 (4/4)



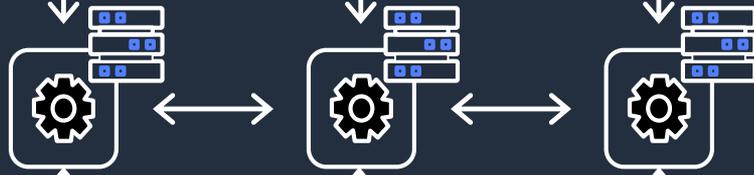
④リーダースタックが実行結果をクライアントに返す。

リーダーノード

Amazon Redshift



コンピューターノード



広帯域ネットワーク

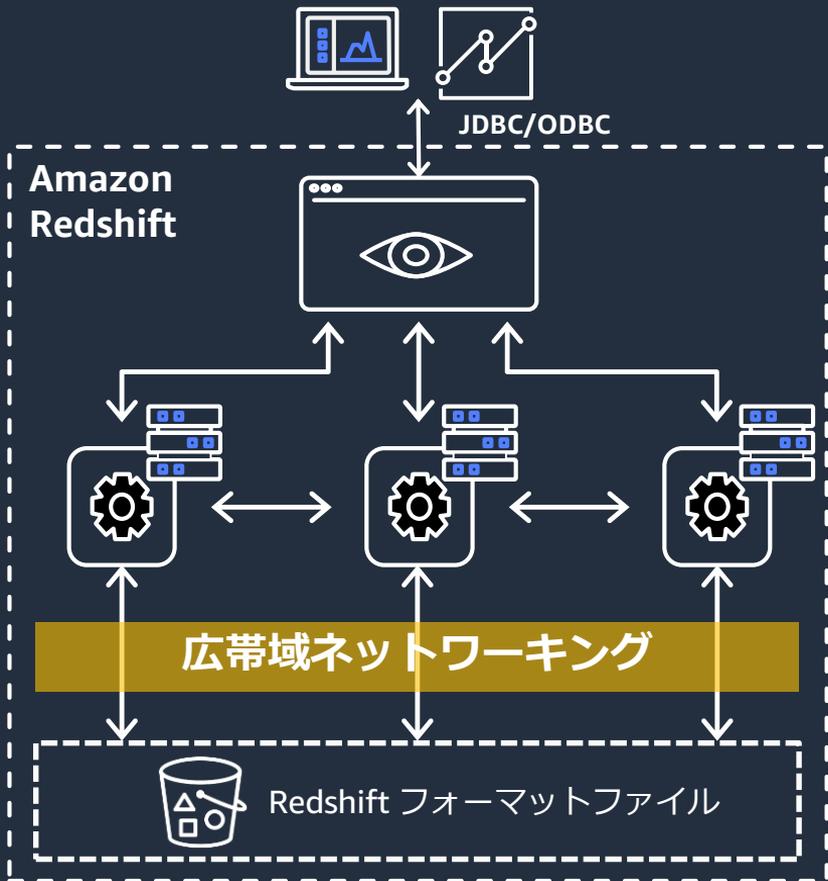
マネージドストレージ



Redshift フォーマットファイル

データロードとアンロード

- データはユーザー管理の S3 を経由してロード & アンロード



Amazon Redshift の特長

AWS 公式 Webinar

<https://amzn.to/JPWebinar>

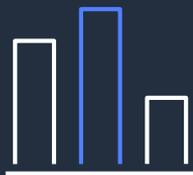


過去資料

<https://amzn.to/JPArchive>



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性

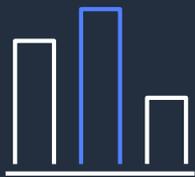


高いコスト効果



セキュリティ &
コンプライアンス

Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性



高いコスト効果



セキュリティ &
コンプライアンス

ハイパフォーマンスを実現するための4つのポイント

1.列指向ストレージ

2.データ圧縮

3.ソートキー

4.データ分散

ハイパフォーマンスを実現するための4つのポイント

1.列指向ストレージ

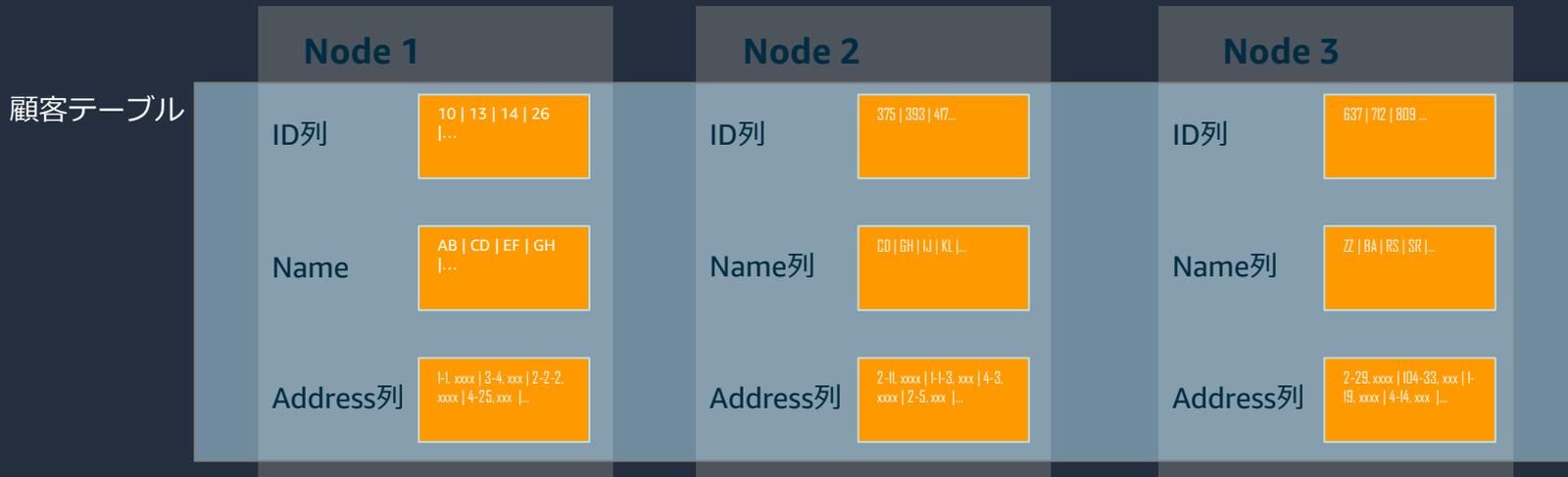
2.データ圧縮

3.ソートキー

4.データ分散

列指向ストレージとは

- Redshift は列指向ストレージを採用
 - データは列ごとに格納 / 必要な列のみの読み取りが可能に
 - 大容量データへのアクセスが必要となる分析クエリのボトルネックとなる ディスク IO を削減



行指向ストレージの検索例

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
,loc CHAR(3) --location  
,dt  DATE     --date  
);
```

aid	loc	dt
1	SFO	2019-10-20
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

aid	loc	dt

```
SELECT min(dt) FROM deep_dive;
```

行指向ストレージ

- 全ての行にアクセスする必要がある
- 不必要なディスク I/O が発生

列指向ストレージの検索例

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
,loc CHAR(3) --location  
,dt  DATE     --date  
);
```

aid	loc	dt
1	SFO	2019-10-20
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

aid	loc	dt

```
SELECT min(dt) FROM deep_dive;
```

列指向ストレージ

- 必要な列のブロックのみのスキャン
- 不必要なディスク I/O の削減

ハイパフォーマンスを実現するための4つのポイント

1.列指向ストレージ

2.データ圧縮

3.ソートキー

4.データ分散

データ圧縮とは

- 列の圧縮を行うことで、一度のディスクアクセスで読み込めるデータ量が多くなり、速度の向上が見込める
- 圧縮のエンコード（アルゴリズム）が 13 個用意されており、CREATE TABLE で各列に選択することが可能
- ANALYZE COMPRESSION コマンドを利用することにより、既存のテーブルの各列に最適な圧縮を確認することが可能
- エンコードタイプ変更したいときには、テーブル再作成 & INSERT-SELECT で対応

圧縮が未指定の場合、圧縮エンコードが自動割り当て

列の前提条件	圧縮エンコードの種類
ソートキーとして定義されている	RAW 圧縮
BOOLEAN、REAL、または DOUBLE PRECISION データ型	RAW 圧縮
SMALLINT、INTEGER、BIGINT、DECIMAL、DATE、TIMESTAMP、または TIMESTAMPTZ データ型	AZ64 圧縮
CHAR または VARCHAR データ型	LZO 圧縮

データ圧縮の設定例 (1/2)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
,loc CHAR(3) --location  
,dt  DATE     --date  
);
```

aid	loc	dt
1	SFO	2019-10-20
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

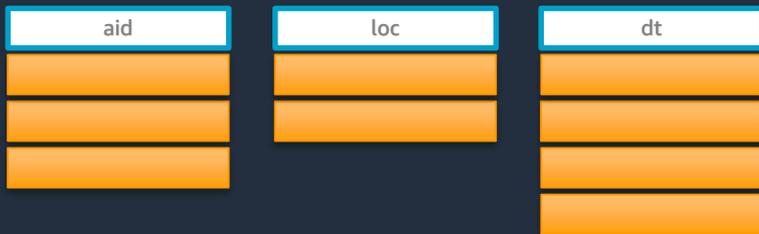
aid	loc	dt

各列に、ビルトインされた 13 のエンコーディングタイプのうちの 1 つを指定可能

データ圧縮の設定例 (2/2)

```
CREATE TABLE deep_dive (  
  aid INT      → AZ64 に自動設定  
  ,loc CHAR(3) → LZO に自動設定  
  ,dt  DATE    → AZ64 に自動設定  
);
```

aid	loc	dt
1	SFO	2019-10-20
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14



列ごとにデータを格納している
ため類似したデータが集まり、
高い圧縮率

ストレージ使用率の圧縮と I/O
削減に寄与

ハイパフォーマンスを実現するための4つのポイント

1.列指向ストレージ

2.データ圧縮

3.ソートキー

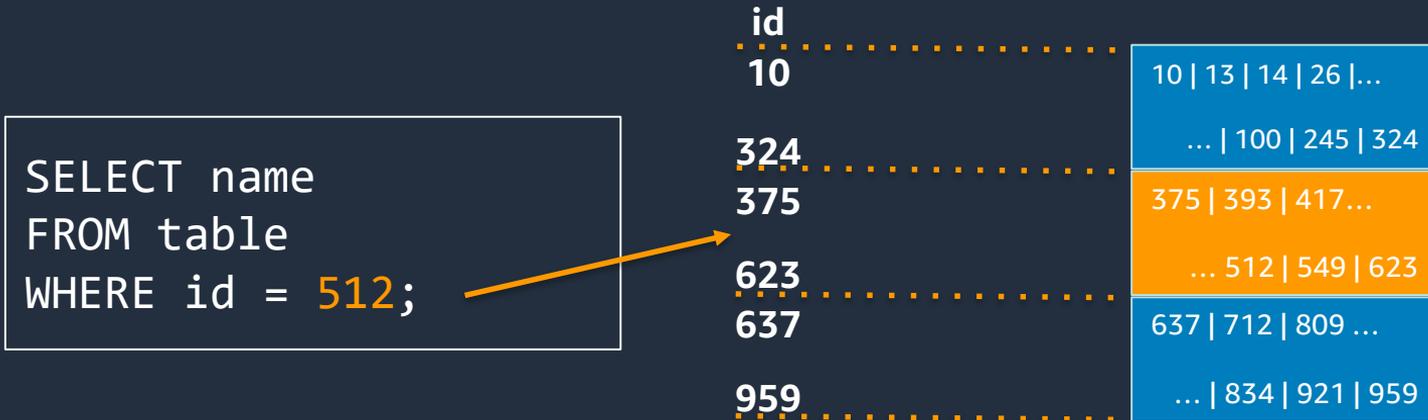
4.データ分散

1MB ブロックサイズ

- 分析ワークロード向けに、データ格納ブロックサイズを大きく確保（1MB/block）することにより、ディスク I/O を効率化している
 - RDBMSの一般的なデータブロックサイズは、8KB~32KB/block
- 各列のブロックは、13 の圧縮エンコーディングタイプのうちの1つを指定して個別にエンコードされる

ゾーンマップ (ブロックフィルタリング) とは

- データが格納されている各ブロック (1MB) に関するメタデータをリーダーノードのインメモリ上に格納
 - 各ブロック上に存在する最小値、最大値を保持
 - クエリの内容に応じて、処理に不必要なブロックは読み飛ばすよう効率的なアクセスを実現



ソートキーとは

- ゾーンマップ機能を活用してディスク I/O を削減し、クエリパフォーマンスを向上させるため、データソートは重要なポイント
- データをどの列順にソートするかを、テーブルごとにソートキーとして指定
 - 頻繁に使われる絞り込み（WHERE 句）条件キーが筆頭候補
 - 一般的には日付などの時系列や ID が多い
 - 実際のクエリパターンと処理優先度などから決定

データソートの実行例 (1/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
,loc CHAR(3) --location  
,dt  DATE     --date  
);
```

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

データソートの実行例 (2/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt  DATE    --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

データソートの実行例 (3/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01

データソートの実行例 (4/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01
4	JFK	2019-05-14

データソートの実行例 (5/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01
4	JFK	2019-05-14
2	JFK	2019-10-20

データソートの実行例 (6/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01
4	JFK	2019-05-14
2	JFK	2019-10-20
1	SFO	2019-11-11

データソートの実行例 (7/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01
4	JFK	2019-05-14
2	JFK	2019-10-20
1	SFO	2019-11-11

データソートの実行例 (8/8)

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) SORT KEY (dt);
```

1つ以上の列をソートキーに指定し、予めデータをソートしてディスク上に保持しておく

deep_dive		
aid	loc	dt
1	SFO	2019-11-11
2	JFK	2019-10-20
3	SFO	2019-04-01
4	JFK	2019-05-14

deep_dive (sorted)		
aid	loc	dt
3	SFO	2019-04-01
4	JFK	2019-05-14
2	JFK	2019-10-20
1	SFO	2019-11-11



ゾーンマップによる範囲限定スキヤンの実行例 (1/4)

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2019';
```

未ソートテーブル



MIN: 01-JUNE-2019

MAX: 20-JUNE-2019



MIN: 08-JUNE-2019

MAX: 30-JUNE-2019



MIN: 12-JUNE-2019

MAX: 20-JUNE-2019



MIN: 02-JUNE-2019

MAX: 25-JUNE-2019

ゾーンマップによる範囲限定スキヤンの実行例 (2/4)

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2019';
```

未ソートテーブル



MIN: 01-JUNE-2019

MAX: 20-JUNE-2019



MIN: 08-JUNE-2019

MAX: 30-JUNE-2019



MIN: 12-JUNE-2019

MAX: 20-JUNE-2019



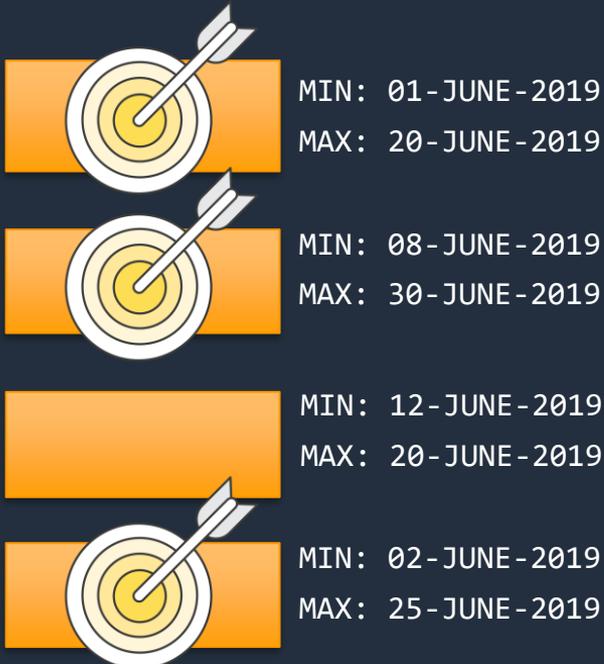
MIN: 02-JUNE-2019

MAX: 25-JUNE-2019

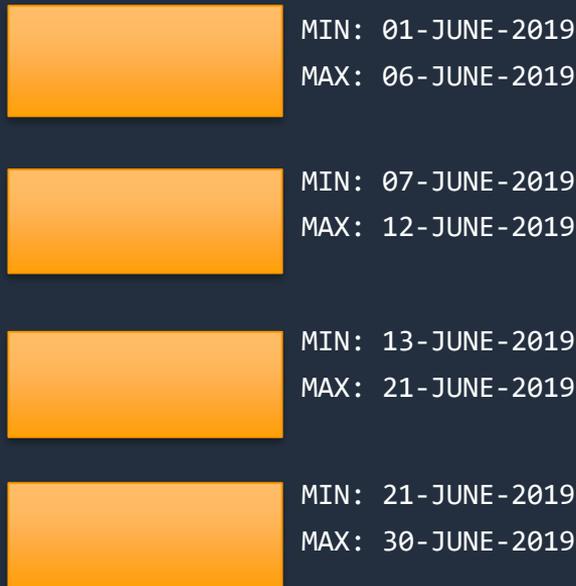
ゾーンマップによる範囲限定スキヤンの実行例 (3/4)

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2019';
```

未ソートテーブル



dt列でソートされたテーブル

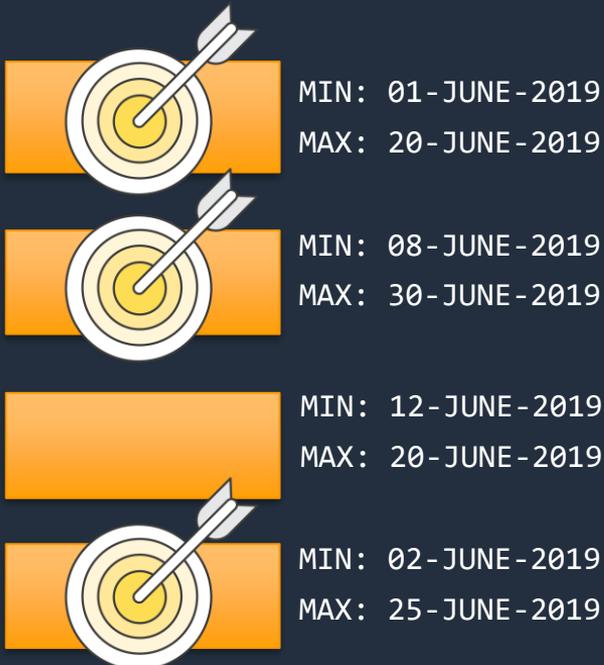


ゾーンマップによる範囲限定スキヤンの実行例 (4/4)

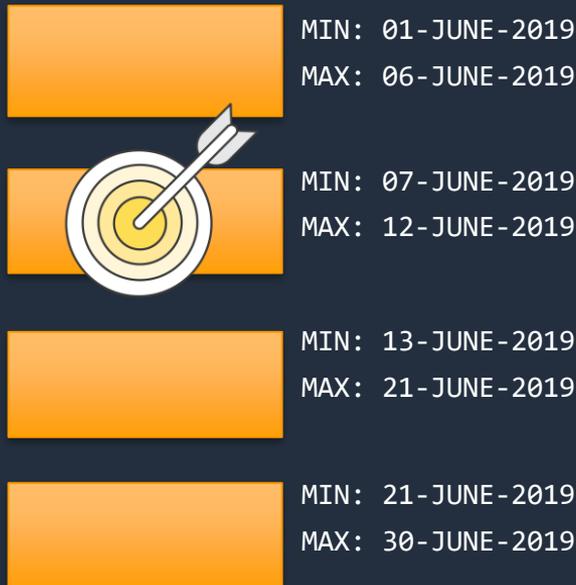
データがソートされている場合、ピンポイントで必要なブロックに到達できる

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2019';
```

未ソートテーブル



dt列でソートされたテーブル



ハイパフォーマンスを実現するための4つのポイント

1.列指向ストレージ

2.データ圧縮

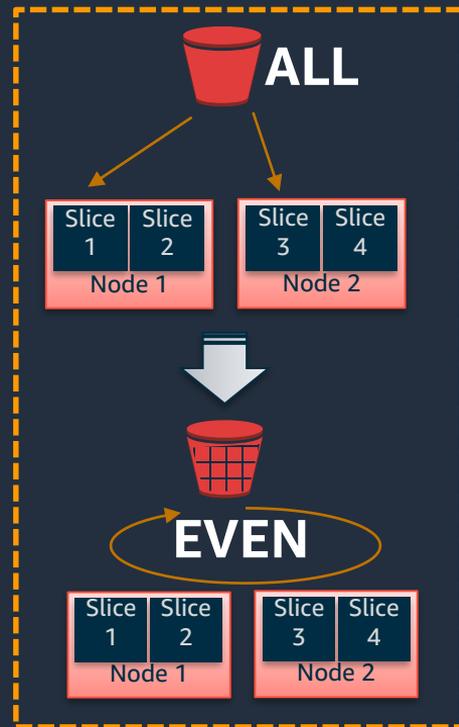
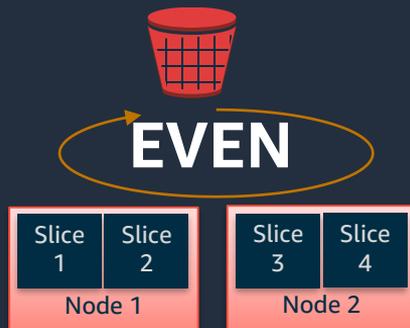
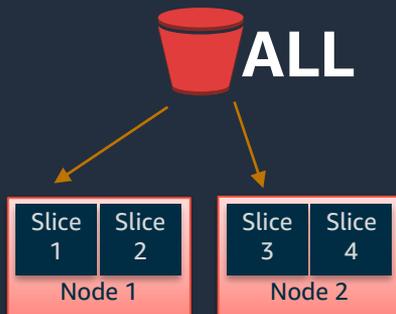
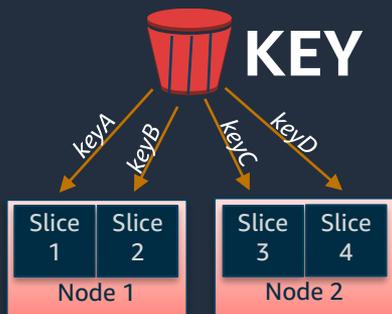
3.ソートキー

4.データ分散

データ量やクエリの内容に応じて分散形式を選択

```
CREATE TABLE deep_dive (  
  aid INT      --audience_id  
  ,loc CHAR(3) --location  
  ,dt DATE     --date  
) DISTSTYLE (KEY|ALL|EVEN|AUTO);
```

AUTO



同じキーは同じスライスへ 全てのデータを全ノードへ ラウンドロビンで均等分散

テーブルサイズに応じて
ALL→EVENに自動変換
(デフォルト)

参考: <https://www.slideshare.net/AmazonWebServicesJapan/aws-black-belt-online-seminar-2017-amazon-redshift/7>

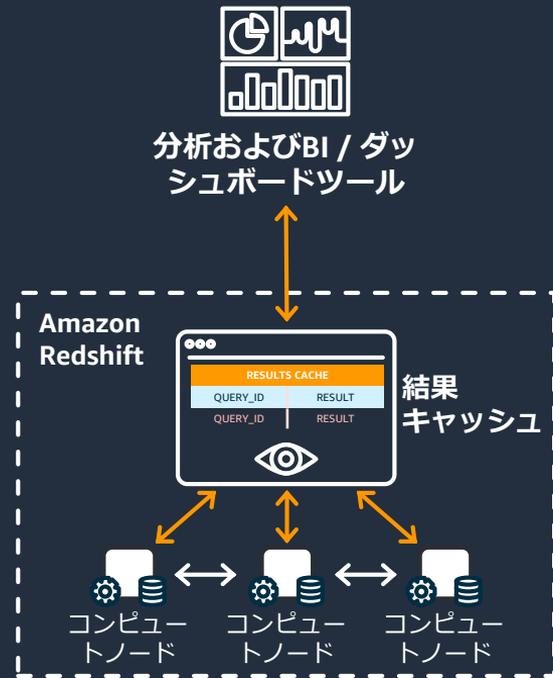
ハイパフォーマンスを支えるその他の機能

結果のキャッシュにより、繰り返し実行されるクエリを1秒未満で実行

1. クエリはリーダーノードにて受付される。
2. リーダーノード内のキャッシュにクエリ結果が含まれている場合、コンピューターノード上での処理を伴わずに返される。
3. クエリ結果がキャッシュに存在しない場合、コンピューターノード上でクエリが実行されて、その結果がキャッシングされる。

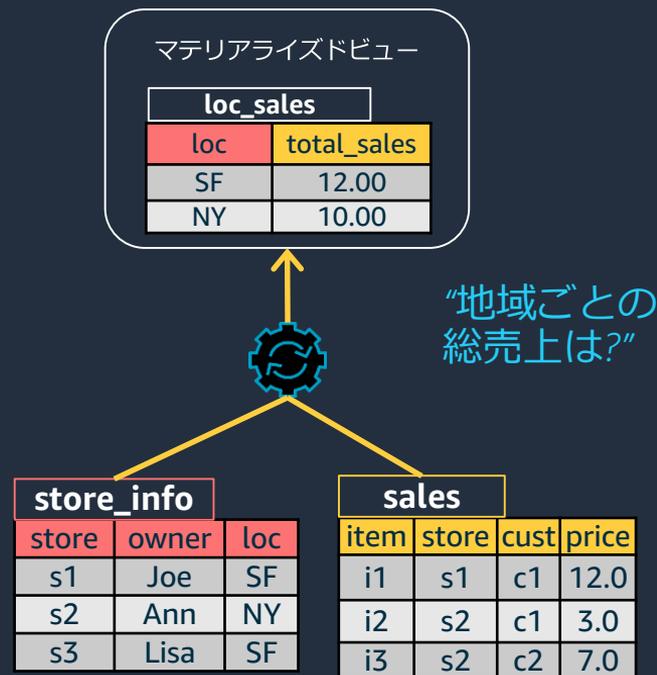
※機能はデフォルトで有効

➡ 結果のキャッシュ機能によって Amazon Redshift クラスターの処理に余裕が生じて、クエリ全般でパフォーマンスが向上する

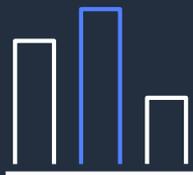


マテリアライズドビューを設定することでさらなる高速化

- 頻繁に実行するクエリパターンを高速化
 - 結合、フィルタ、集計、射影
- ETL/BI パイプラインの簡素化
 - 差分リフレッシュ
 - ユーザーによるメンテナンス
- Redshift へのより簡単で迅速な移行



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性



高いコスト効果



セキュリティ &
コンプライアンス

構築・運用の手間を削減

- 1画面の設定のみで起動
- ノード数やタイプは後から変更可能

クラスタの
ノード構成

クラスタを作成

クラスタ設定 ニーズに最適な構成を計算する

ノードの種類
CPU、RAM、ストレージ容量、およびドライブタイプの要件を満たすノードの種類を選択します。

推奨

RA3
スケーラブルなマネージド型ストレージによる高パフォーマンス

ra3.16xlarge	\$15.347/ノード/時
マネージド型ストレージ: 最大	\$0.026/GB/月
64 TB/ノード	

ノード構成: ra3.16xlarge, 48 vCPU (3世代)

▶ レガシー高密度ストレージノードタイプを表示

ノード
必要なノードの数を入力します。
2 (範囲: 2~128)

マスターユーザー名
DB インスタンスのマスターユーザーのログイン ID を入力します。
awsuser
名前は 1-128 文字の実数字にする必要があります。予約語にすることはできません。

マスターユーザーのパスワード

 パスワードを表示

- 値は 8-64 文字である必要があります。● 値には、少なくとも 1 つの大文字が含まれている必要があります。
- 値には、少なくとも 1 つの小文字が含まれている必要があります。
- 値には、少なくとも 1 つの数字が含まれている必要があります。
- マスターパスワードに含めることができるのは、ASCII 文字 (ASCII コード 33-126) のみです。ただし、(一重引用符)、(二重引用符)、(ハッシュ記号) は除きます。

▶ クラスタのアクセス許可 (オプション)

追加設定 デフォルトを使用
これらの設定はオプションであり、クラスタの使用を開始するためにデフォルト設定が定義されています。これらの設定を今すぐ変更するには、[Use default] をオフにしてください。

ネットワーク デフォルト VPC (vpc-) とデフォルトサブネットを使用	セキュリティ default (sg-69f71b12) クラスタセキュリティグループを使用
バックアップ 自動スナップショットは 8 時間ごとに作成され、2 日間保たれます。	設定 データベース暗号化なしの default.redshift-1.0 パラメータグループを使用
メンテナンス 現在のメンテナンストラックを使用	

キャンセル クラスタを作成

クラスタの
詳細設定

構築・運用の手間を削減

- 自動バックアップやモニタリング機能を内蔵
 - GUI (マネジメントコンソール)
 - API経由で操作も可能
- パッチ適用も自動実行
 - メンテナンスウィンドウでパッチの適用時間帯を指定可能
- スケジューリング機能
 - クラスターサイズの変更
 - クラスターの一時的停止と再開



The screenshot shows the 'Pause cluster' configuration form. At the top, there are three radio buttons: 'Pause now', 'Pause later', and 'Pause and resume on schedule' (which is selected). Below this, there are two input fields: 'スケジュール名' (Schedule name) with the value 'weekly-schedule' and 'Description - optional' (Description for the schedule). A note states: 'The identifier must be 1-63 characters. Valid characters are a-z (lowercase only), 0-9, and - (hyphen)'. There are two date pickers: '開始:' (Start) set to '2020/03/18' and '終了:' (End) set to '2021/03/18'. Below these are two tabs: 'エディタ' (Editor) and 'CRON 構文' (CRON syntax). At the bottom, there are two sections for scheduling: 'Pause every' and 'Resume every'. Both are set to 'オン' (On). The 'Pause every' section has a dropdown set to '週' (Week) and a row of buttons for days: '日' (Sun), '月' (Mon), '火' (Tue), 'W' (Wed), '木' (Thu), '金' (Fri), and '土' (Sat), with '土' selected. The time is set to '07:00' (時刻 (UTC)). The 'Resume every' section has a dropdown set to '週' (Week) and a row of buttons: '日' (Sun), '月' (Mon), '火' (Tue), 'W' (Wed), '木' (Thu), '金' (Fri), and '土' (Sat), with '月' selected. The time is also set to '07:00' (時刻 (UTC)).

機械学習ベースの自動最適化によるクエリ性能の向上

- テーブルメンテナンスの自動化
 - テーブルの分散スタイルの自動最適化
 - 統計情報の自動更新
 - データの再編成の自動実行
- 自動ワークロード管理とクエリ優先度で効率的に処理
- ショートクエリアクセラレーション (SQA) による高速化
- 動的に設定変更可能なリコメンデーションの提案



Automatic Analyze



Automatic Table Distribution Style



Distribution/Sort key advisors



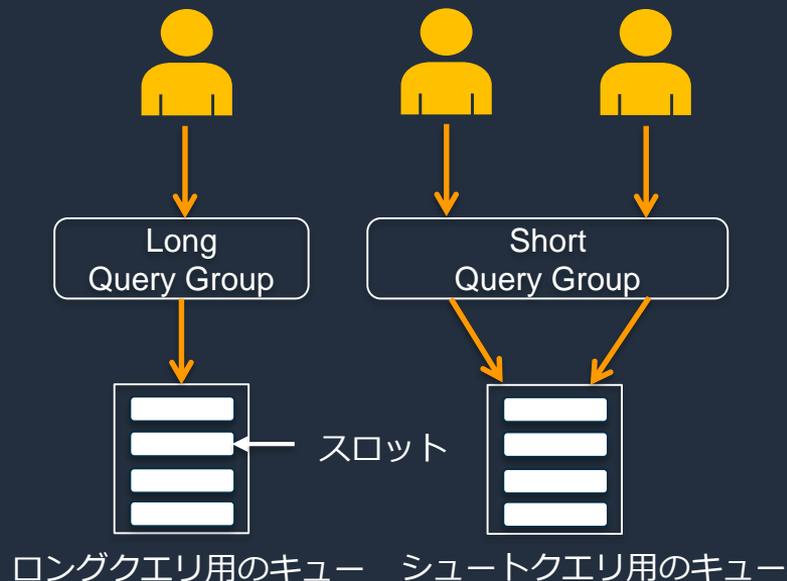
Automatic Vacuum Delete



Automatic Table Sort

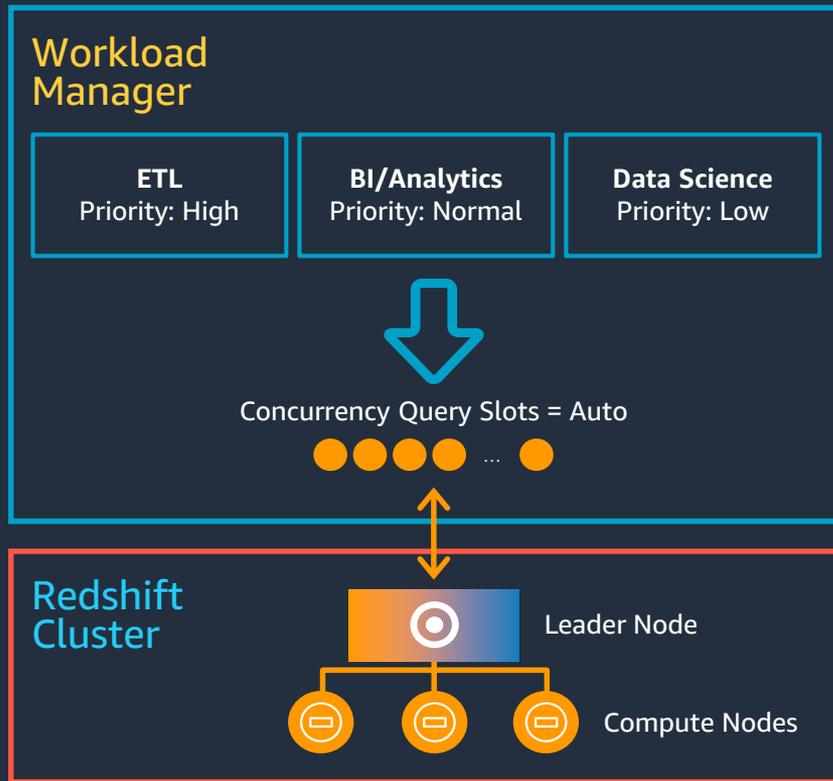
ワークロード管理：キューとスロット

- ユーザーはワークロードの種類などにもとづいてキューを複数個作成することが可能
- 各キューにはクラスターが使用できるメモリの一部を割り当てる
- クエリ割り当てルールにより、各クエリが各キューに割り当てられる
- スロットは、クエリを処理するために使用されるメモリと CPU の単位となり、各キューごとに設定可能
- スロットを増やすと並列度は上がるが、スロットあたりの割当てメモリが減る



自動ワークロード管理とクエリ優先度で効率的に処理

- ワークロード管理の設定を自動化（デフォルト）することで、クエリへのメモリ割り当てを最適化するために **並列スロット数を自動的に決定**
- クラスターのリソースを効率よく配分することをサポート
- スループットとパフォーマンスの最大化
- 機械学習を活用して、リソース要求をもとにしたクエリの分類を実施
- ビジネスの優先順位に基づいてワークロード毎のパフォーマンスを調整可能

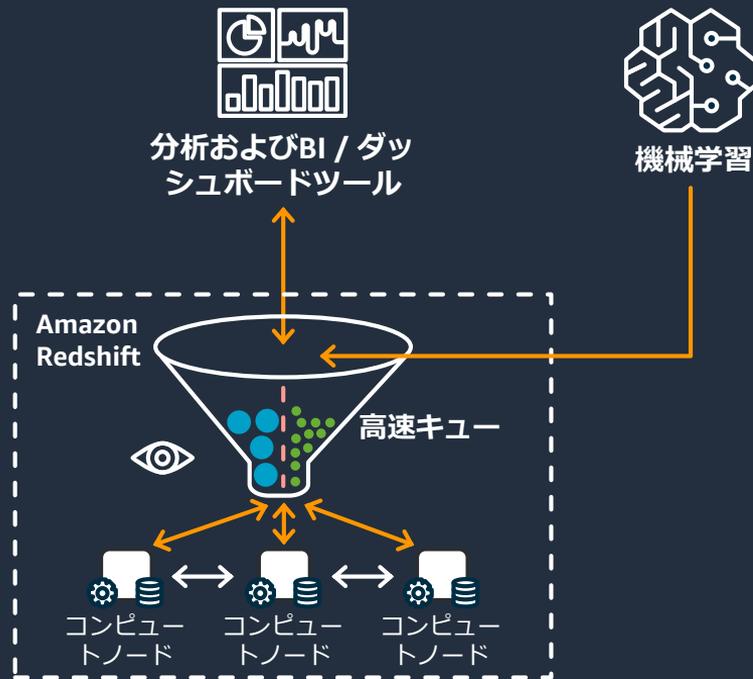


ショートクエリアクセラレーション (SQA) による高速化

実行時間の短いクエリのスループットを高速化

1. 機械学習によってクエリの実行時間を予測する。
2. ショートクエリと判断されたクエリは専用の高速キューにルーティングされる。
3. リソースはショートクエリのために動的に確保される。

※機能はデフォルトで有効
(ユーザー側での設定は不要)



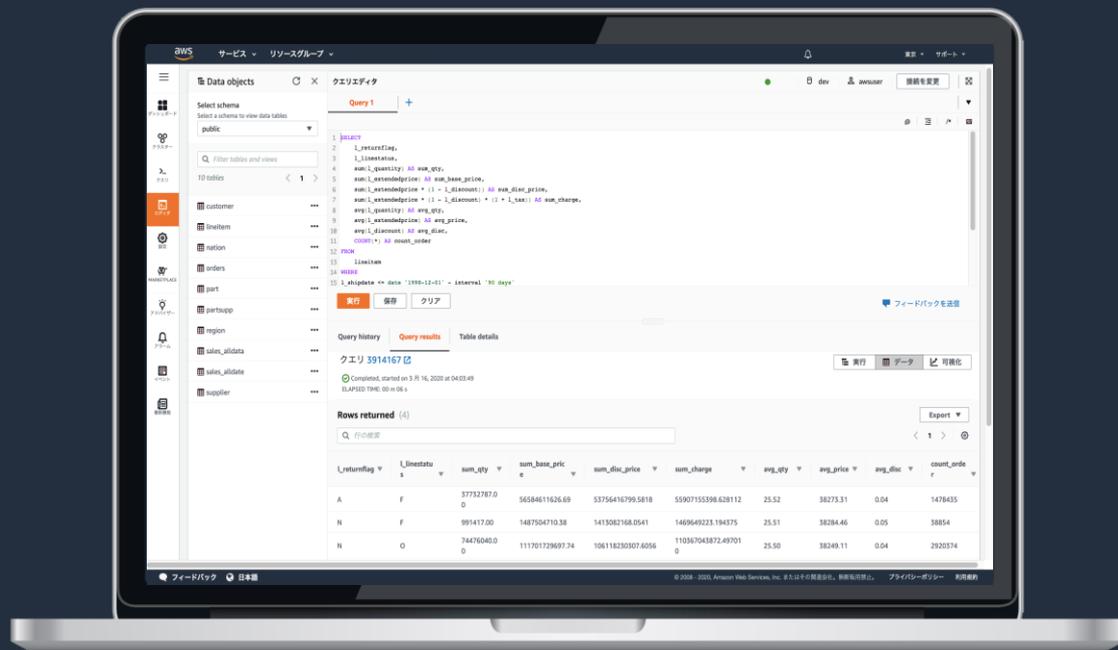
動的に設定変更可能なリコメンデーションの提案

- Redshift が自動でクラスターパフォーマンスや使用状況のメトリックを分析し、パフォーマンスの最適化や運用コストの削減のためのリコメンデーションを提供
- 例：lineitem テーブルにおいて、L_ORDERKEY 列を分散キーに指定した方がよい場合

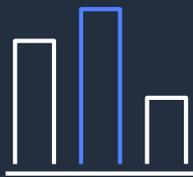
```
ALTER TABLE lineitem  
    ALTER DISTSTYLE KEY DISTKEY L_ORDERKEY ;
```

クエリエディタ機能

- マネジメントコンソールから直接 Redshift クラスターのデータにクエリを実行可能（専用クライアント不要）
- Saved queries 機能を使用すると、実行頻度の高いクエリを保存し、将来必要ときに簡単に使用可能



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性

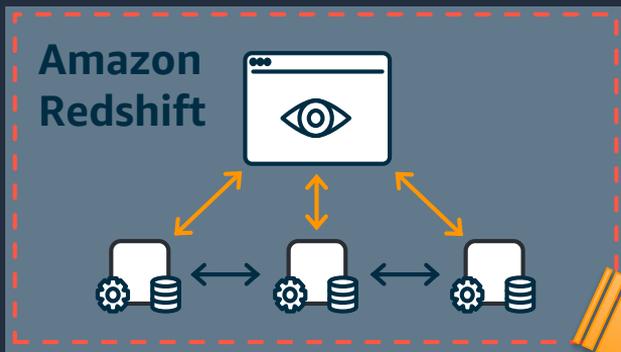


高いコスト効果

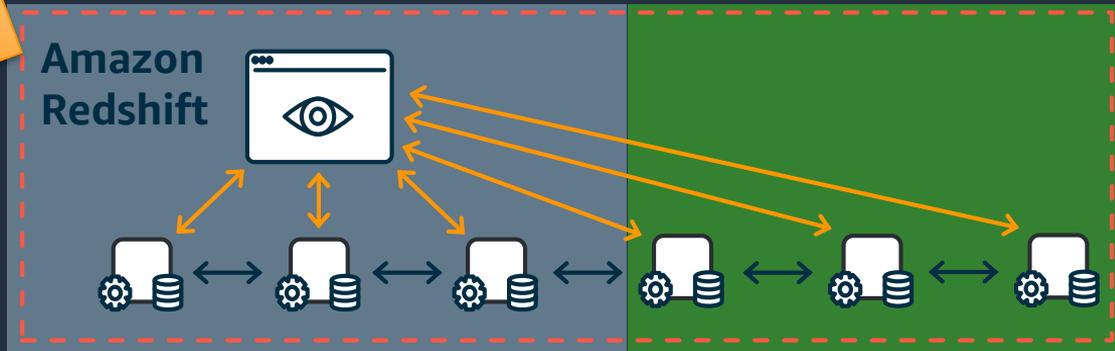


セキュリティ &
コンプライアンス

コンピューットノードの追加で拡張可能

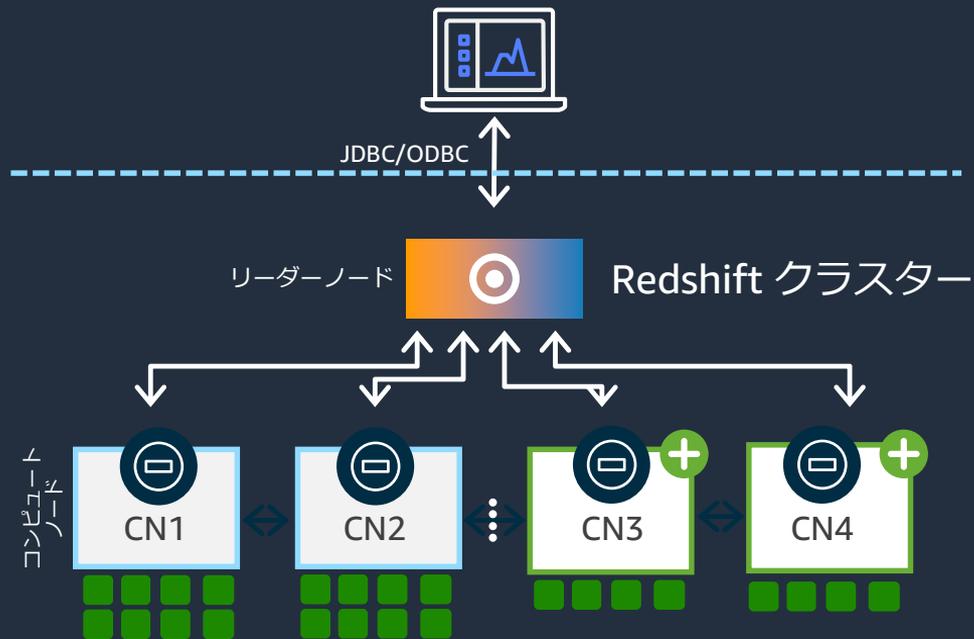


- コンピューットノードの追加でパフォーマンスがリニアに向上
- マネージメントコンソールから数クリックで拡張・縮小が可能



数分でクラスターの伸縮を実現する Elastic Resize

- 既存の Redshift クラスター上にノードを追加
- 繁忙期にクエリを高速に処理
- データ転送にかかる時間を最小化
- コンピュートとストレージをオンデマンドでスケール
- スケジュール設定可能



参考 : <https://www.slideshare.net/AmazonWebServicesJapan/20190122-aws-black-belt-online-seminar-amazon-redshift-update/31>

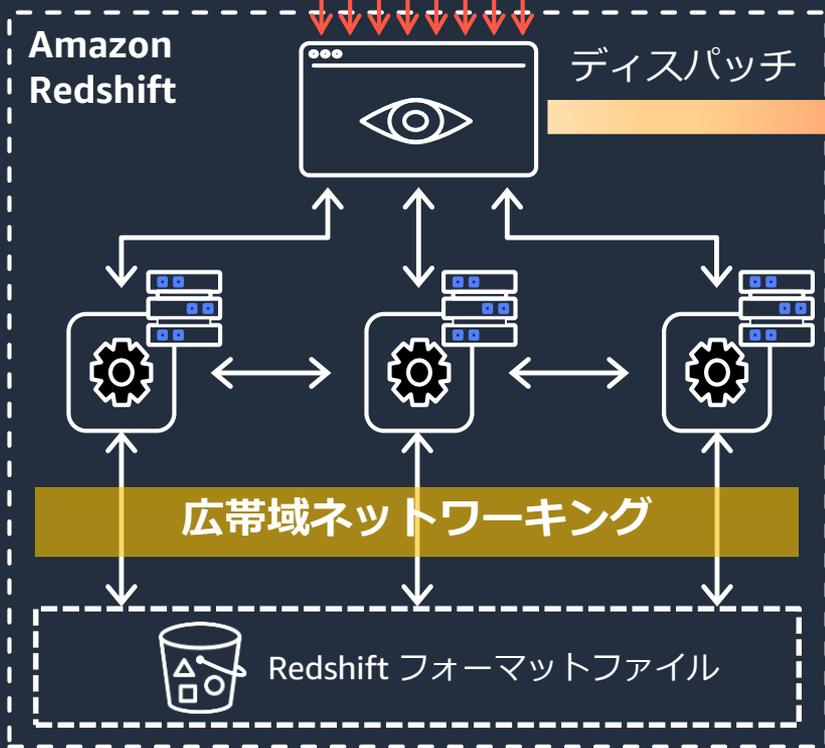
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



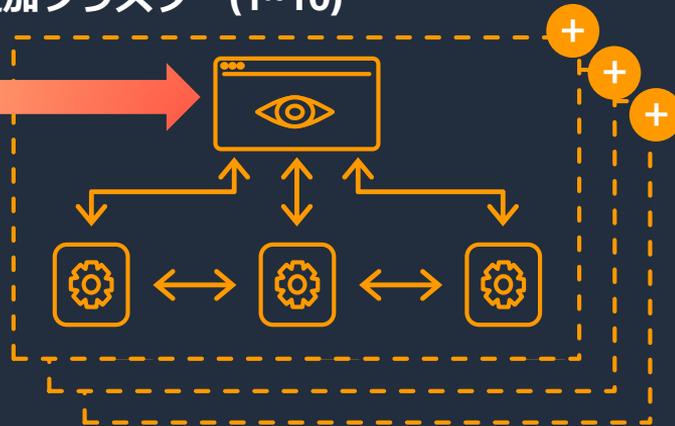
ピーク時にコンピューートを自動拡張する **Concurrency**

Scaling

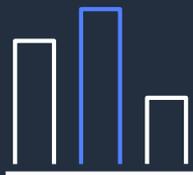
メインクラスター



追加クラスター(1~10)



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性

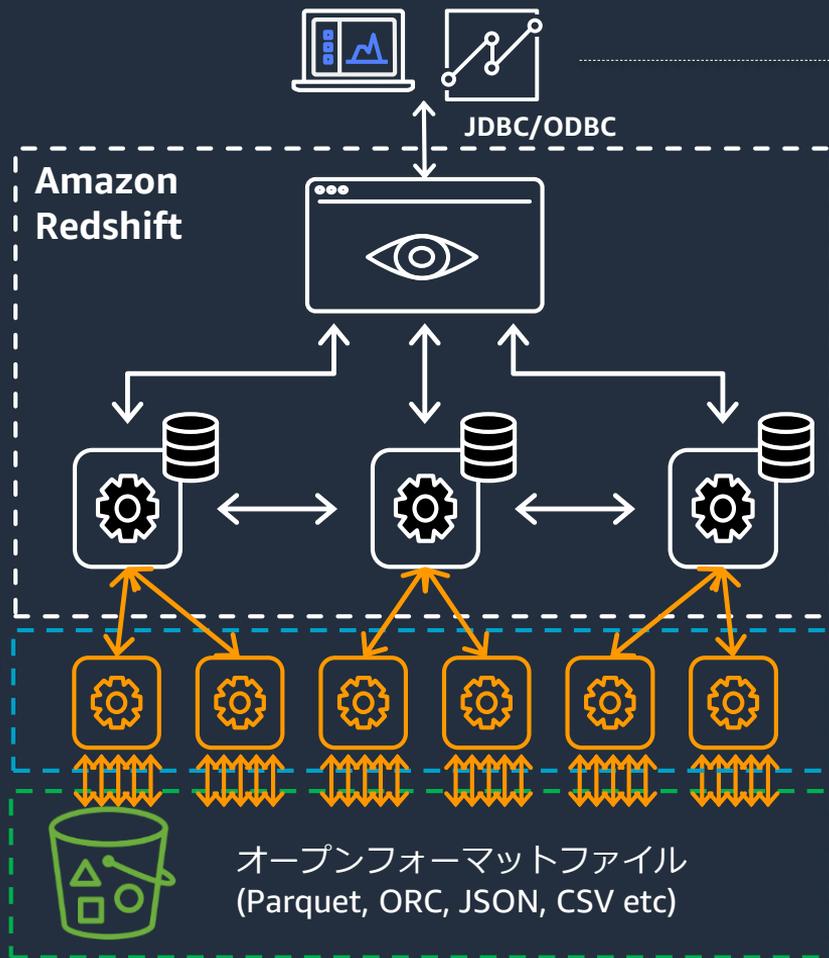


高いコスト効果



セキュリティ &
コンプライアンス

Redshift Spectrum でアーキテクチャをデータレイクに拡張



アプリケーションは、データウェアハウスとデータレイクの双方のデータに透過的にアクセスできる

Amazon Redshift Spectrum
S3 ファイルへのクエリ実行エンジン

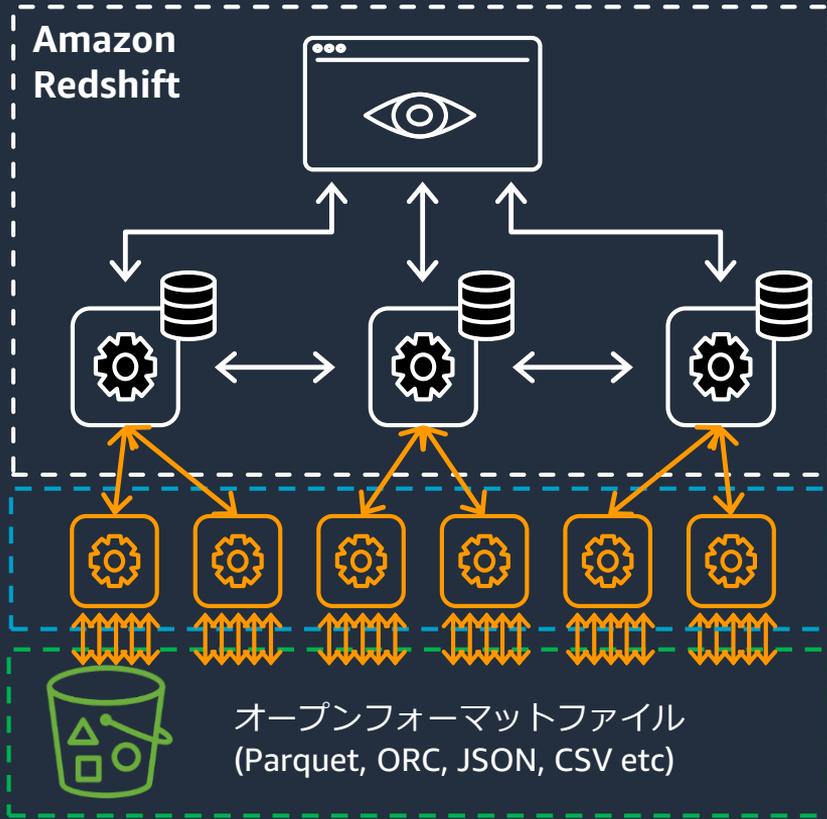
Amazon S3
ユーザーバケット

オープンフォーマットファイル
(Parquet, ORC, JSON, CSV etc)



Redshift Spectrum のアーキテクチャ

- 事前のデータロード不要で、S3上のデータに対して直接 SQL を実行
- Redshift と S3 それぞれに存在するデータを結合可能
- オープンファイルフォーマット対応 (Parquet, ORC, JSON, Grok, Avro, およびCSV等)



参考 : <https://www.slideshare.net/AmazonWebServicesJapan/20200218-aws-black-belt-online-seminar-next-generation-redshift/27>

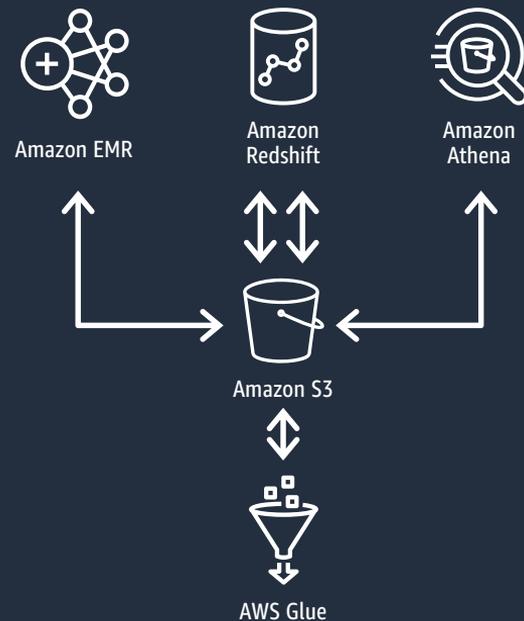
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

RA3 と Redshift Spectrum との違い・使い分け

	RA3	Redshift Spectrum
ストレージ	Redshift 管理の S3 領域を使用	ユーザー管理の S3 領域を使用
データフォーマット	Redshift 独自フォーマット	オープンフォーマット
データへのアクセス	Redshift からのアクセスのみ	他の AWS サービスからもアクセス可
データの更新	DML UPDATE に対応	DML UPDATE には非対応
コスト	Redshift インスタンス料金 + マネージドストレージ料金	Redshift インスタンス料金 + S3ストレージ格納料金 + Spectrum スキャン容量料金
特徴	RA3 のマネージドストレージは Redshift ストレージの拡張であり、その背後で S3 の技術が使われているが、利用時に S3 を意識する必要はない	データレイクとしての S3 にアクセスする機能であり、データレイクを介した他サービスとの連携のために今後も重要な機能であり続ける
使い分け	これまで Redshift 内のデータ増加を抑制する目的で過去データを削除したり、S3 にデータをオフロードして Spectrum を活用していた場合は、RA3 ストレージを活用	すべてのデータがデータレイクにあり、Redshift だけではなく他のサービスからも同じデータにアクセスする要件がある場合は Spectrum を活用

データレイクエクスポート - Parquet でデータを共有

- Redshift テーブルデータを Amazon S3 上へ Apache Parquet 形式でエクスポートすることが可能
- Apache Parquet は分析ワークロード向けのオープンな列指向ファイルフォーマット
- Redshift Spectrum だけでなく、Amazon Athena や Amazon EMR など AWS の他の分析サービスでもすぐに分析に活用可能

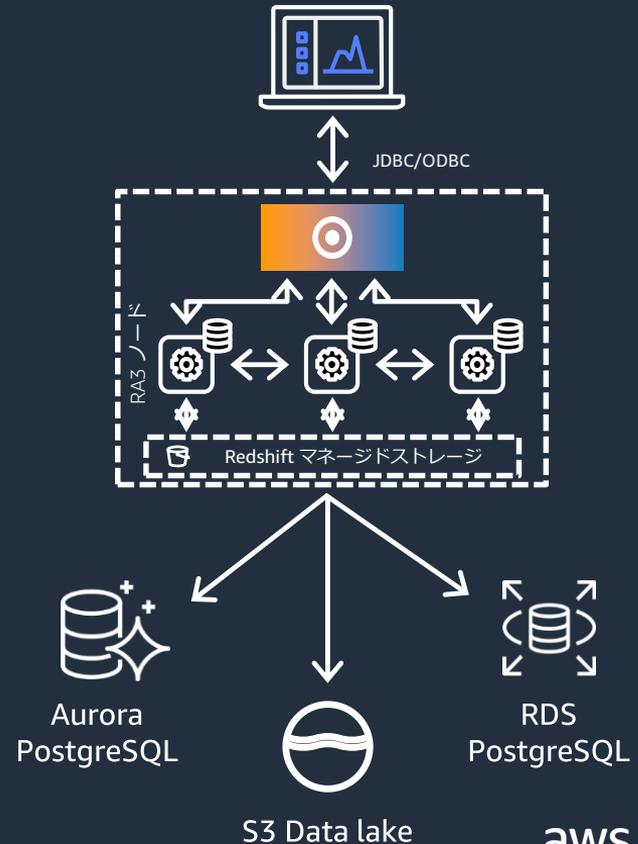


参考 : <https://www.slideshare.net/AmazonWebServicesJapan/20200218-aws-black-belt-online-seminar-next-generation-redshift/63>

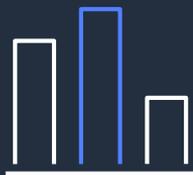
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

フェデレーテッドクエリ (プレビュー中)

- RDS/Aurora PostgreSQL に対して直接クエリ可能に
- データ移動なしにライブデータを分析
- データウェアハウス、データレイク、オペレーショナルデータベースのデータを統合して分析
- 高い性能でセキュアにデータアクセス



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性



高いコスト効果



セキュリティ &
コンプライアンス

Amazon Redshift の料金

*費用は 2020 年 3 月時点での東京リージョンのもので
コンピュータ料金は、1 時間未満の時間は 1 秒単位で請求されます

Redshift インスタンス起動時間 + ストレージ使用量 (RA3のみ)

- コンピュートノード数 x 1 時間あたりの価格 (リーダーノードは課金対象外)
- リザーブドインスタンス (1 年、3 年) にも対応

	vCPU	メモリ	ストレージ	I/O	スライス	ノード数	コンピュート料金 (*)	ストレージ料金 (*)
RA3 with Redshift Managed Storage(RMS)								
ra3.4xlarge	Stay tuned !							
ra3.16xlarge	48	384 GB	64TB RMS	8.0 GB/s	16	2~128	\$15.347 /h	\$0.026 /GB-Month
DC2 - Dense Compute								
dc2.large	2	15 GB	0.16TB SSD	0.6 GB/s	2	1~32	\$0.314 /h	
dc2.8xlarge	32	244 GB	2.56TB SSD	7.5 GB/s	16	2~128	\$6.095 /h	
DS2 - Dense Storage								
ds2.xlarge	4	31 GB	2TB HDD	0.4 GB/s	2	1~32	\$1.190 /h	
ds2.8xlarge	36	244 GB	16TB HDD	3.3 GB/s	16	2~128	\$9.520 /h	

参考 : <https://www.slideshare.net/AmazonWebServicesJapan/20200218-aws-black-belt-online-seminar-next-generation-redshift/16>

その他の関連料金

* 費用および課金体系は 2020 年 3 月時点での東京リージョンのものです

Concurrency Scaling の料金

- Redshift Concurrency Scaling クラスターでのクエリ実行時間で課金
 - 各追加クラスターでクエリが実行された期間 (秒)
 - 1 日あたり 1 時間分の**無料クレジット**が付与 (最大 30 時間)

Redshift Spectrum の料金

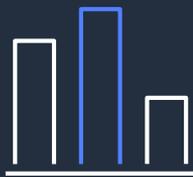
- Amazon S3 データレイクへのクエリ容量で課金
 - S3 上の「圧縮済み」データ 1 TB スキャンあたり \$5 (*)

参考: <https://aws.amazon.com/jp/redshift/pricing/>

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性



高いコスト効果

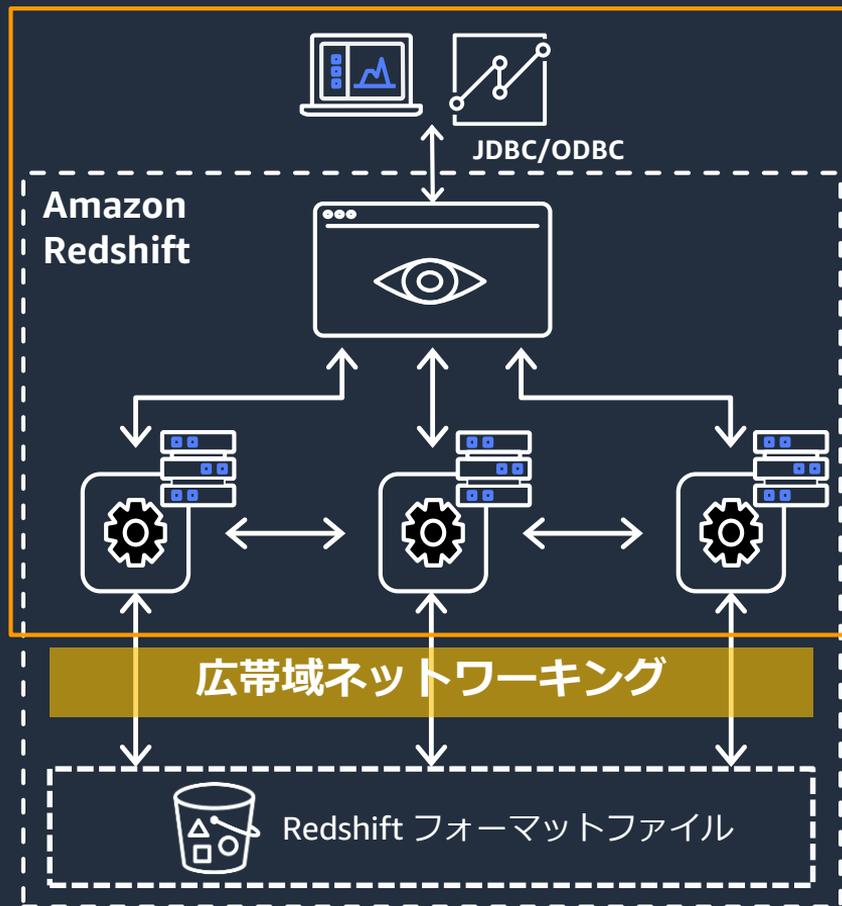


セキュリティ &
コンプライアンス

ビルトインされたセキュリティ機能

Customer VPC

- エンドツーエンドのデータ暗号化
- データ保管時の暗号化可能
- ネットワーク分離のための Amazon VPC
- 拡張された VPC のルーティング
- 監査ロギングとAWS CloudTrail 統合
- 多くの認定および準拠
 - SOC 1/2/3、PCI-DSS、FedRAMP、HIPAA



アクセスコントロール

- テーブル、データベース、スキーマ、関数、プロシージャ、言語、または列に対してアクセスコントロール設定可能
- テーブル、ビューに対して、列ごとにアクセスコントロール設定可能
 - GRANT/REVOKE コマンドで設定可能
 - テーブルの場合：列レベルで SELECT と UPDATE の権限付与可能
 - ビューの場合：列レベルで SELECT の権限付与可能

```
grant select(cust_name, cust_phone) on cust_profile to user1;
```

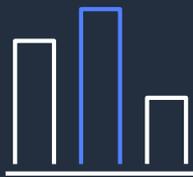
```
grant select(cust_name, cust_phone), update(cust_contact_preference) on  
cust_profile to group sales_group;
```

参考：https://docs.aws.amazon.com/ja_jp/redshift/latest/dg/r_GRANT.html

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



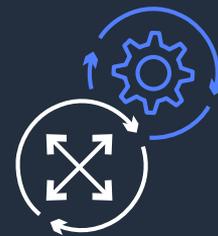
Amazon Redshift の特長



ハイパフォーマンス



フルマネージド



拡張性 & 柔軟性



データレイク &
AWS サービスとの親和性



高いコスト効果



セキュリティ &
コンプライアンス

まとめ

AWS 公式 Webinar
<https://amzn.to/JPWebinar>



過去資料
<https://amzn.to/JPArchive>



まとめ

- Redshift は、高速、スケーラブルで費用対効果の高いデータウェアハウスおよびデータレイク分析マネージドサービス
- 多くのメンテナンスタスクやデータ配置が自動化されており、煩雑になりがちなデータウェアハウスの運用を簡単に
 - 自動テーブルメンテナンス
 - 自動ワークロード管理 etc
- アーキテクチャの特長を踏まえ、ユーザー側で設定をカスタマイズすることによって、更に快適な環境にしていくことも可能

Q&A

お答えできなかったご質問については

AWS Japan Blog 「<https://aws.amazon.com/jp/blogs/news/>」にて
後日掲載します。

AWS の日本語資料の場所「AWS 資料」で検索



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼

コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)

[AWS 初心者向け »](#)

[業種・ソリューション別資料 »](#)

[サービス別資料 »](#)

<https://amzn.to/JPArchive>



AWS Well-Architected 個別技術相談会

毎週"W-A個別技術相談会"を実施中

- AWSのソリューションアーキテクト(SA)に
対策などを相談することも可能

- 申込みはイベント告知サイトから

(<https://aws.amazon.com/jp/about-aws/events/>)

AWS イベント

で[検索]



ご視聴ありがとうございました

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>

