



# Amazon Redshift 運用管理

## AWS Black Belt Online Seminar

平井 健治

Solutions Architect  
2023/03

# 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=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBIqY>

# 内容についての注意点

- 本資料では 2023 年 3 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：平井 健治

所属：アマゾン ウェブ サービス ジャパン  
ソリューションアーキテクト

流通小売業のお客様の技術支援を担当

経歴：データベース、データ活用に関するコンサルタント

好きなAWSサービス：Amazon S3、AWS Lambda、  
Amazon Redshift



# 本セミナーの対象者

Amazon Redshift をご利用予定の方、ご利用されている方

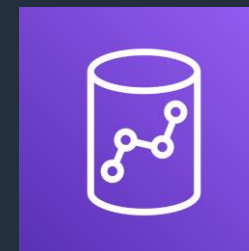
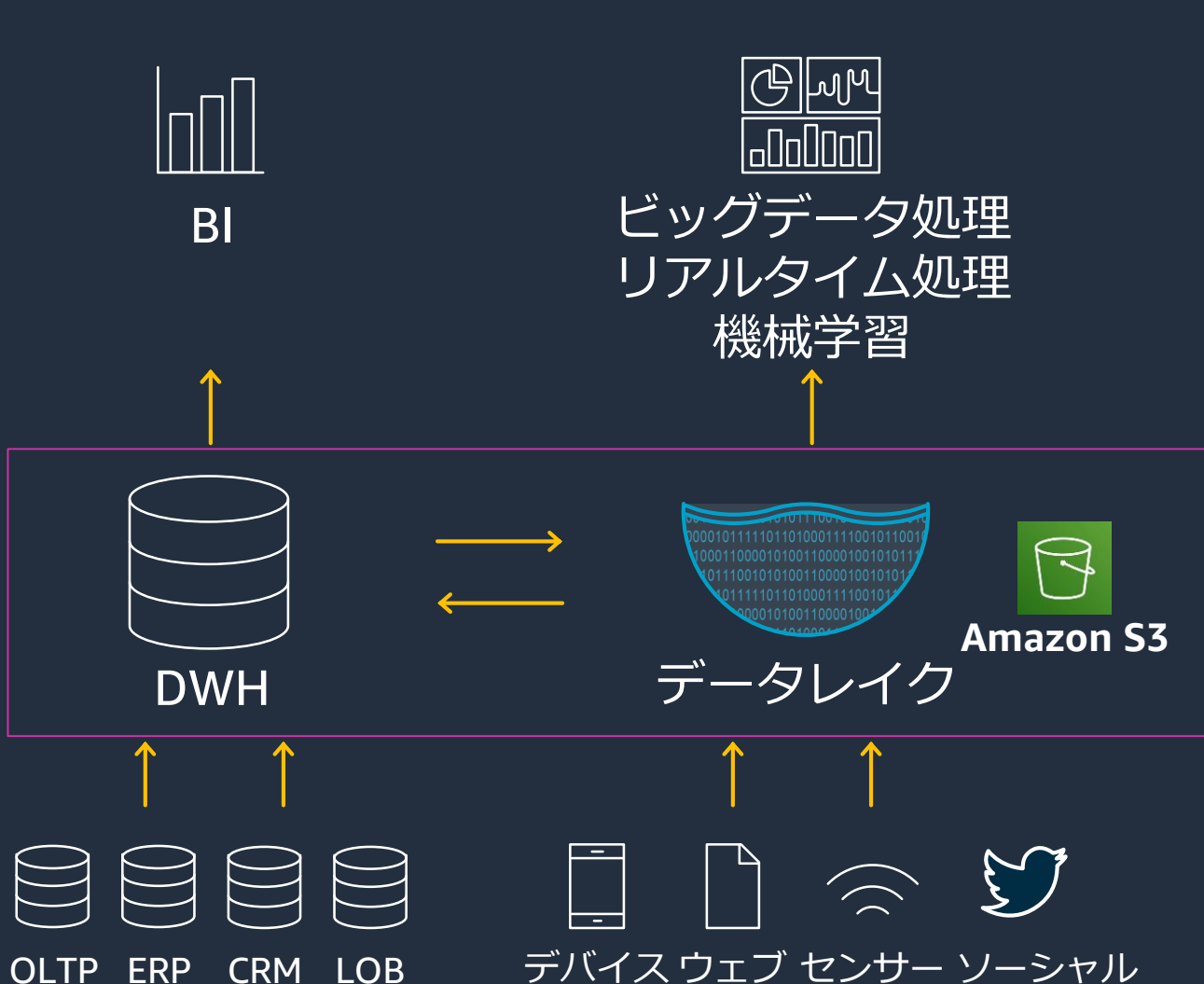
Amazon Redshift について深く学びたい方

# アジェンダ

1. Amazon Redshift とは
2. Amazon Redshift を使ってみる
3. 自動化された運用タスク
4. モニタリングと性能改善
5. 利用者の増加に対処する
6. データ連携はどのように行うか

# Amazon Redshift とは

# Amazon Redshift とは



## Amazon Redshift

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 は 分析用クエリ実行に特化した RDBMS

## 分析ワークロードに特化

- 列指向ストレージ
- 並列分散処理
- PostgreSQL 互換の SQL

## 柔軟なクラスタ構成

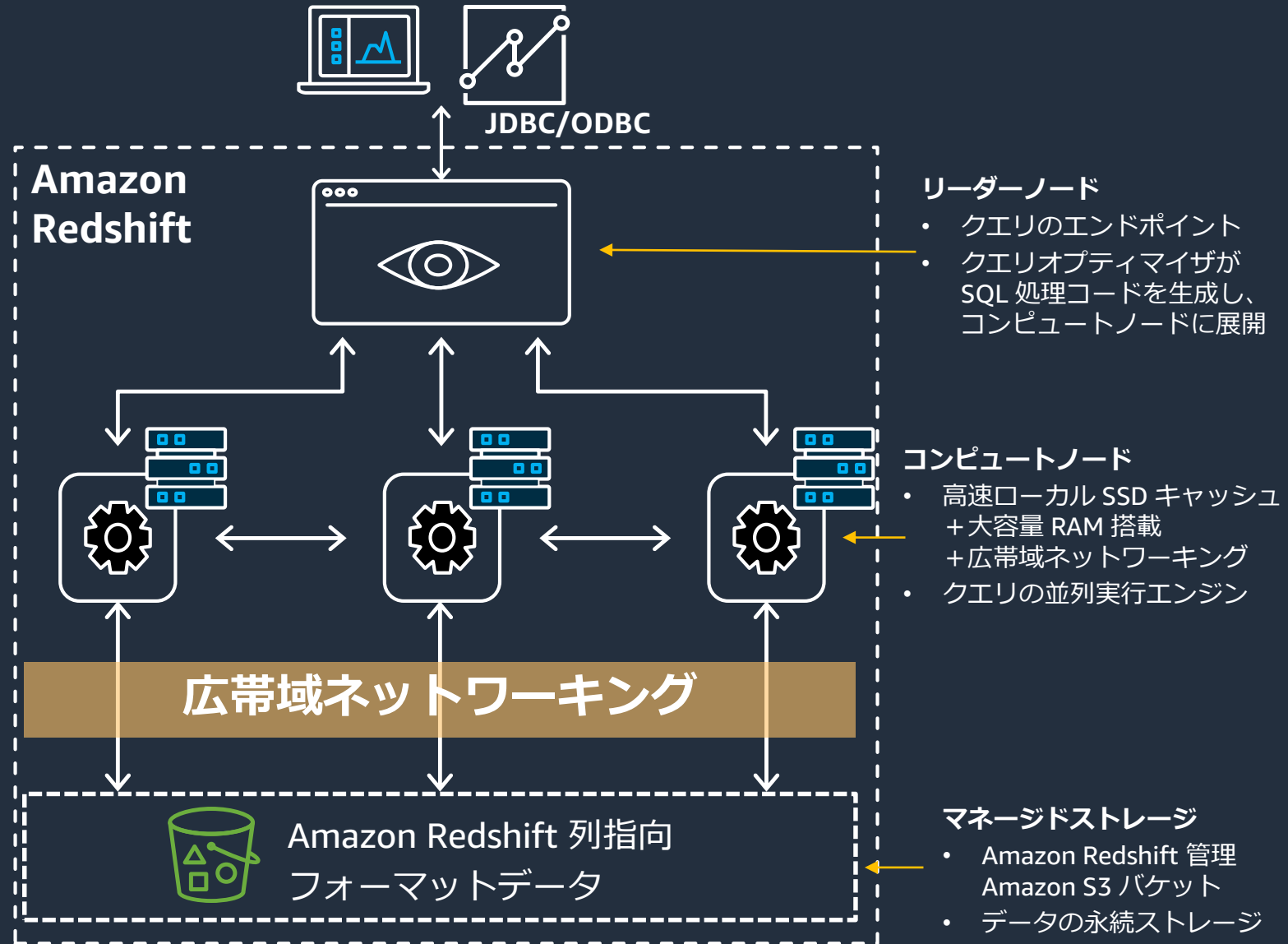
- コンピュートノードの追加で処理能力をスケール
- マネージドストレージにより、データ量増大をノード追加なく対処可能

## 自動的にワークロードを最適化

- テーブルメンテナンスの自動化
- 分散、ソートキー、圧縮、マテリアライズドビューの自動設定
- クエリへのリソース割当を自動で最適化

## 追加の手動チューニングも可能

- 圧縮、分散、ソート、マテリアライズドビュー
- クエリの優先度設定
- チューニングのヒントは自動でアドバイス



# 従来からの Provisioned 型に加え Serverless 型も利用可能

## シンプルで使いやすい

- DWH クラスタを管理することなくデータ分析の実行やスケーリングが可能に

## インテリジェントに自動でスケール

- 一貫した高速なパフォーマンスを提供するために、DWHの処理能力を自動的にプロビジョニングしスケーリングする

## 高度な機能・性能はそのまま

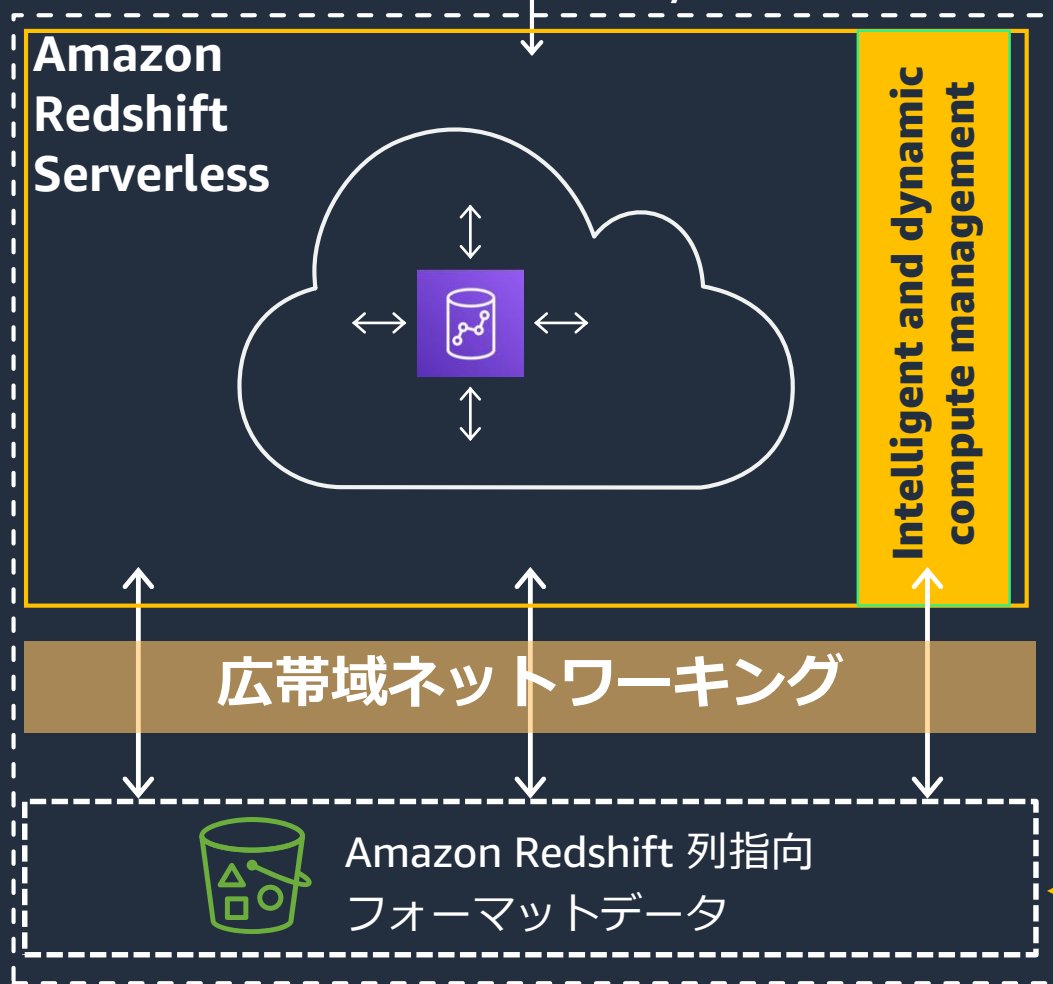
- Amazon Redshift の豊富な SQL の機能やデータレイクとのシームレスな統合、業界をリードする価格パフォーマンスをそのまま利用できる

## 使った分だけの課金

- コンピュート料金はワークロードの継続時間に  
応じて秒単位(\*)でのお支払い、アイドル時間の  
料金はかからない  
(\*) 60秒の最低料金あり



JDBC/ODBC



アプリケーションを Amazon Redshift Serverless endpoint へ接続すれば、すぐに実行可能

Amazon Redshift SQL のすべての機能が利用可能

### マネージドストレージ

- Amazon Redshift 管理 Amazon S3 バケット
- データの永続ストレージ

# 参考) 各種クエリサービスの使い分け



Amazon  
Athena



Amazon  
Redshift  
Serverless



Amazon  
Redshift  
Provisioned

用途

簡易な処理

簡易～複雑な処理  
(例: 複数の結合やサブクエリ)

簡易～複雑な処理  
(例: 複数の結合やサブクエリ)

規模

低頻度または予測不能  
ワークロード  
(小～中規模)

低頻度または予測不能  
ワークロード  
(小～大規模)  
最大 RPU(\*1) 512

予測可能  
ワークロード  
(中～超大規模)  
最大 ra3.16xlarge 128 ノード

基盤管理

自動

自動

プロビジョニング  
詳細なチューニング可

費用

クエリスキャン量  
(別途ストレージサービス使用料)

ワークロード実行時間  
RMS(\*2) ストレージ使用量

クラスタ稼働時間  
(リザーブドインスタンス適応可能)  
RMS ストレージ使用量  
クエリスキャン量  
(Spectrum 利用時)

Serverless と Provisioned の比較についての詳細は以下ドキュメントを参照

<https://docs.aws.amazon.com/redshift/latest/mgmt/serverless-console-comparison.html>

© 2023, Amazon Web Services, Inc. or its affiliates.

(\*1) RPU : Redshift Processing Unit

(\*2) RMS : Redshift Managed Storage

# Amazon Redshift を 使ってみる

# 数クリックで手軽にスタート

## Amazon Redshift Provisioned

- 1画面の入力作業ですぐに起動
- ノード数やタイプは後から変更可能

Amazon Redshift > クラスター > クラスターを作成

### クラスターを作成 情報

#### クラスター設定

クラスター識別子  
これは、クラスターを識別する一意のキーです。

識別子は 1~63 文字にする必要があります。有効な文字は a~z (小文字のみ) および - (ハイフン) です。

このクラスターを何に使用する予定ですか？

**本番稼働**  
高速で一貫したパフォーマンスを最良の料金で提供するよう  
に設定します。

**無料トライアル**  
Amazon Redshift について学習できるように設定します。  
組織が Amazon Redshift クラスターを作成したことがな  
い場合、この設定は一定期間無料になります。

クラスターのサイズを選択する

を選択します。  
 選択のヘルプ

ノードの種類 情報  
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。

ノードの数  
必要なノードの数を入力します。

範囲 (2~32)

## Amazon Redshift Serverless

- デフォルトセッティングでサーバーレスエンドポイントの有効化してすぐに分析可能
- ノード数やタイプなど指定不要

Amazon Redshift サーバーレス > Amazon Redshift サーバーレスの使用を開始する

### Amazon Redshift サーバーレスの使用を開始する

Amazon Redshift サーバーレスの使用を開始するには、サーバーレスデータウェアハウスを設定し、データベースを作成します。  
このアカウントの Redshift サーバーレスの使用に対する 300.00 USD のクレジットが付与されます。

#### 設定 情報

**デフォルト設定を使用**  
使用を開始しやすくするために、デフォルト設定が定義されて  
います。後でいつでも変更できます。

**設定をカスタマイズ**  
特定のニーズに合わせて設定をカスタマイズします。

#### ▼ 仕組み

 **デフォルト設定の使用**

 **設定のカスタマイズ**

#### 名前空間 情報

名前空間は、データベースオブジェクトとユーザーのコレクションです。データプロパティには、データベース名とパスワード、許可、および暗号化とセキュリティが含まれます。

# ところで、何を選べば良い？

## 選定フロー

まずは運用がシンプルな  
Amazon Redshift Serverless を検討

- 定常的なワークロードでコスト最適化したい
- ワークロード管理やクラスタ管理機能を使用して、手動で詳細なチューニングを行いたい

Amazon Redshift Provisioned の  
RA3 を検討

- データがそれほど多くない（例：未圧縮で1TB未満）
- とにかく性能が必要
- RA3 だけ利用可能な機能は使わなくて良い

Amazon Redshift Provisioned の  
DC2 を検討

### DC2 を選択する場合の注意点

- データ量の増加と共にノード追加が必要
- RA3 でだけ利用可能な機能がある
  - AQUA（クエリ高速化機能）
  - Data Sharing
  - クロスAZクラスターリカバリー

# Amazon Redshift Provisioned のノードタイプ

- Amazon Redshift Provisioned の現行世代は、2つのノードファミリーそれぞれに2~3のサイズがあり、合計 5 タイプから選択可能
- ファミリーの違いは主にストレージの種類による
  - RA3 : マネージドストレージ & SSD キャッシュ
  - DC2 : SSD

\*1 2023 年 3 月時点での東京リージョンのものです

	vCPU	メモリ	ストレージ	I/O	デフォルト スライス数	ノード数	コンピュート 料金 (*1)	ストレージ 料金 (*1)
<b>RA3 with Redshift Managed Storage(RMS)</b>								
ra3.xlplus	4	32 GiB	32TB RMS	0.65 GB/s	2	2~16 (*2)	\$1.278 /h	\$0.0261 /GB-Month
ra3.4xlarge	12	96 GiB	128TB RMS	2.0 GB/s	4	2~32 (*2)	\$3.836 /h	
ra3.16xlarge	48	384 GiB	128TB RMS	8.0 GB/s	16	2~128	\$15.347 /h	
<b>DC2 - Dense Compute</b>								
dc2.large	2	15 GiB	0.16TB SSD	0.6 GB/s	2	1~32	\$0.314 /h	
dc2.8xlarge	32	244 GiB	2.56TB SSD	7.5 GB/s	16	2~128	\$6.095 /h	

ノード一覧 : <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html>

\*2 クラスタ作成後、Elastic Resize により ra3.xlplusは 32ノード、ra3.4xlarge は 64ノードまで拡張可能です

# 小さいサイズのノードタイプを選択する際の注意点

dc2.large や ra3.xlplus を選択する場合：

- リーダーノードのノードタイプは、コンピュータノードと同じ
- リーダーノードで実行される処理（同時実行ワークロードの制御、クエリコンパイルなど）の負荷が高くなると、リーダーノードがボトルネックとなる可能性がある → その際は上のサイズへのスケールアップを検討する
- 特に dc2.large は搭載メモリ量が少ないため、同時接続数が多い場合にもメモリ枯渇の可能性が出てくることに注意する



# 管理コンソールからサイジング

## 実機検証でのサイジングに先立って初期サイジングの参考にできる

Amazon Redshift > クラスター > クラスターを作成

### クラスターを作成 情報

#### クラスター設定

クラスター識別子  
これは、クラスターを識別する一意のキーです。

redshift-cluster-1

識別子は 1~63 文字にする必要があります。有効な文字は a~z (小文字のみ) および - (ハイフン) です。

このクラスターを何に使用する予定ですか?

本番稼働  
高速で一貫したパフォーマンスを最高の料金で提供するよう  
に設定します。

無料トライアル  
Amazon Redshift について学習できるように設定します。  
組織が Amazon Redshift クラスターを作成したことがな  
い場合、この設定は一定期間無料になります。

クラスターのサイズを選択する

を選択します。

選択のヘルプ

この見積もりは圧縮データまたは未加工データですか? [詳細はこちら](#)

圧縮データに対する見積もり  
Amazon Redshift にロードした後の圧縮データの予測値である場合に選択します。

データウェアハウスに必要な推定ストレージ領域はどれくらいですか?  
Amazon Redshift にロードされるデータは、平均して、オープンデータ形式よりも 3 倍小さ  
くなります。

サイズ

メインで溜まる  
データが時系列の  
履歴ならこちら

格納するデータと、  
そのうちクエリに利  
用するデータの期間

「選択のヘルプ」  
をクリック

一度にクエリするデータの量はどれくらいですか?

私のデータは時間ベースです  
データが時間順にデータウェアハウスに追加される場合に選択します。たとえば、販売データが毎月  
追加されます。

データが時間ベースではない  
データに時間ディメンションがない場合に選択します。たとえば、地理的リージョン別にインベント  
リ内のパーツを一覧表示します。

データウェアハウスには何か月分のデータが含まれていますか?  
保存する予定のデータの月数を予測します。

1 か月 3 か月 12 か月 36 か月 無制限

ワークロードでクエリを実行するデータは何か月分ですか?  
実行するたびに一般的なワークロードがアクセスする月数を予測します。

1 週間 2 週間 1 か月 3 か月 12 か月 無制限

#### 計算された設定の概要

[見積もりを変更する](#) 設定概要を再計算します。

ra3.4xlarge | 8 個のノード

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

コンピューティング  
12 vCPU (gen 3)/ノード x 8 = 96 個のvCPU

オンデマンドコンピューティング料金 の見積もり	予想リザーブド (1 年)	予想リザーブド (3 年)
\$228,460.80/年	\$153,063.985/年	\$90,240.027/年
\$3.26/ノード/時	\$1.092/ノード/時	\$0.644/ノード/時
	33% 割引	61% 割引

マネージド型ストレージ容量  
最大 128 TB x 8 個のノード = 1024 TB  
\$35,389.44/年 (120TB)  
\$0.024/GB/月

[料金の詳細](#)

### 時系列データではない場合

データが時間ベースではない  
データに時間ディメンションがない場合に選択します。たとえば、地理的リージョン別にインベント  
リ内のパーツを一覧表示します。

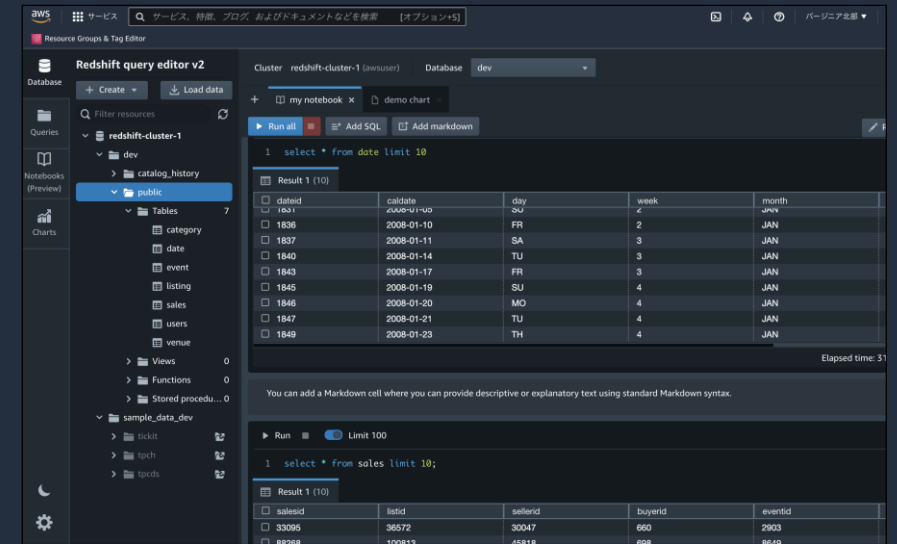
クエリが頻繁にアクセスするデータの推定割合 (%) はどれくらいですか?  
この見積もりは、最適なパフォーマンスのためにプロビジョニングするコンピューティング  
ノードの数を決定するのに役立ちます。たとえば、データボリューム全体の 20% に頻繁にア  
クセスする場合は、データボリューム全体の 80% をクエリする場合よりも少ない数の定常状  
態のコンピューティングノードを使用することをお勧めします。

パーセント



# Amazon Redshiftにクエリしてみよう

- Amazon Redshiftへの主なアクセス手段
  - 管理コンソール内の「クエリエディタ」  
「クエリエディタ v2」
  - Amazon Redshift RSQL
  - Amazon Redshift ODBC/JDBCを利用したアプリケーション
  - Amazon Redshift Data API
  - PostgreSQLへのアクセスをサポートしたアプリケーション(※)



クエリエディタ v2

- データベースへのアクセスにはデータベースユーザーを使う
  - SQL文「CREATE USER」で作成
  - SAML2.0準拠のシングルサインオン(SSO)にも対応
    - 利用可能なプロバイダ : ADFS, Okta, PingFederate, Azure AD

※ PostgreSQLのみをサポートしているアプリケーションの場合、必ずしもAmazon Redshiftへの接続が保証されているわけではないことにご注意ください。

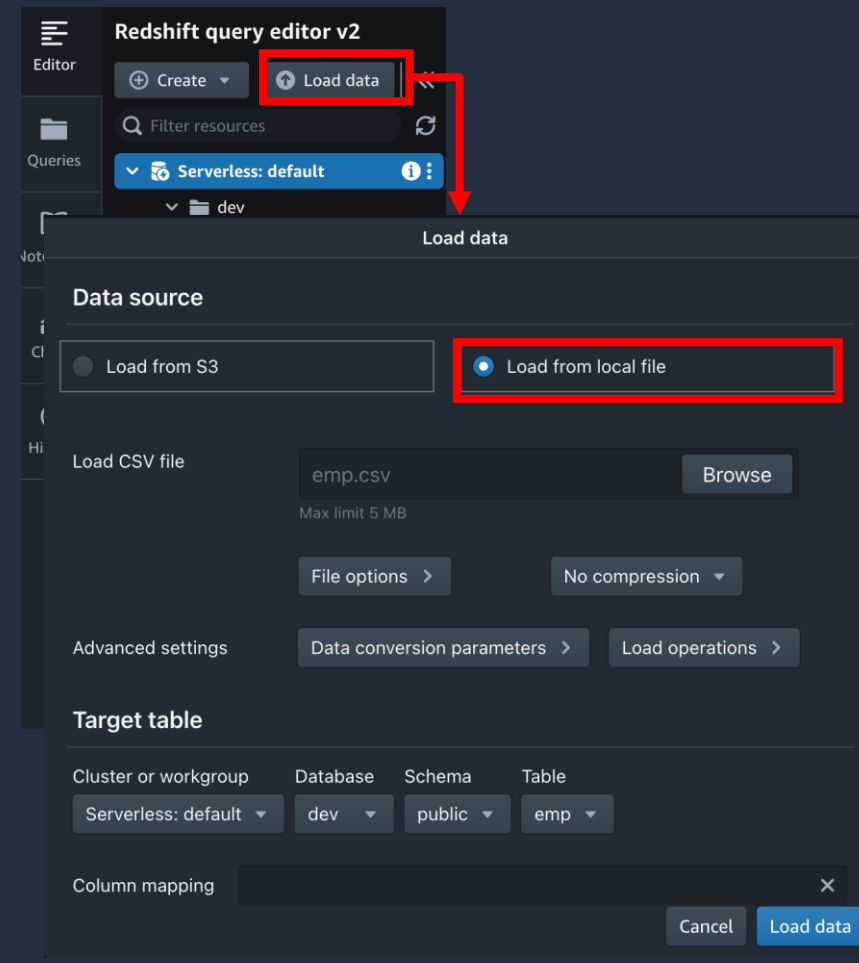
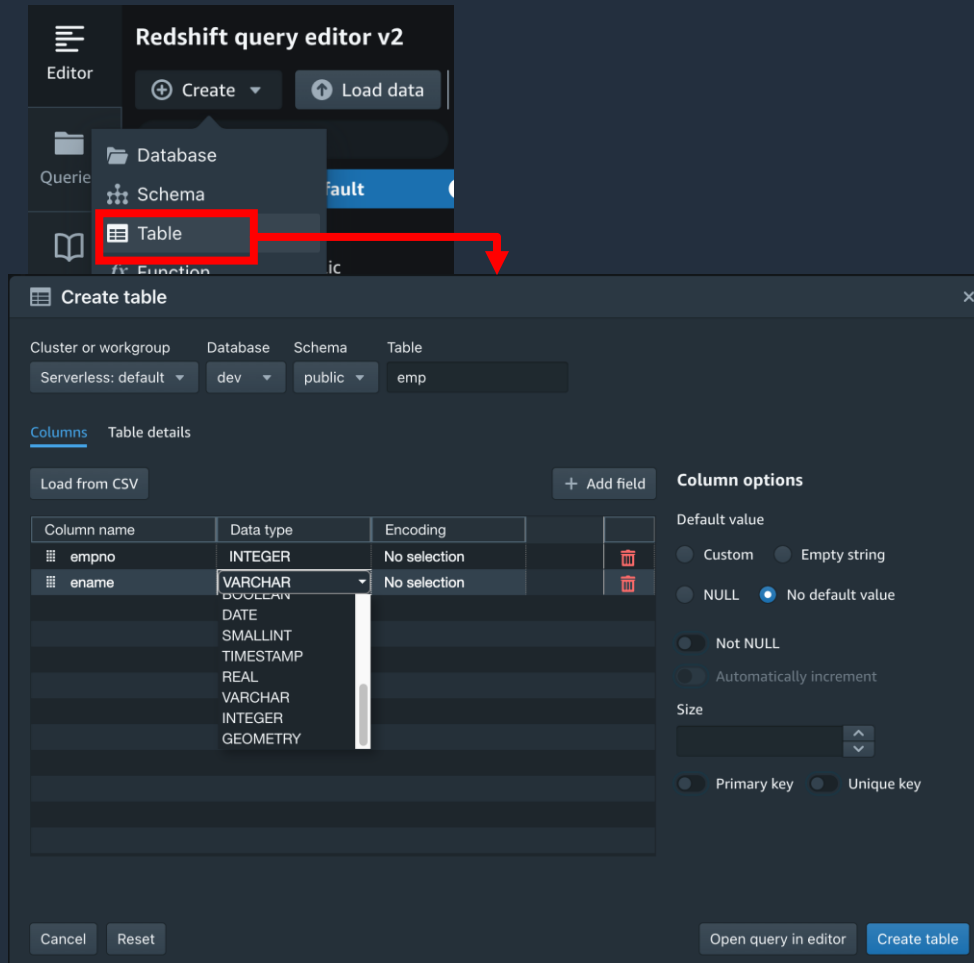
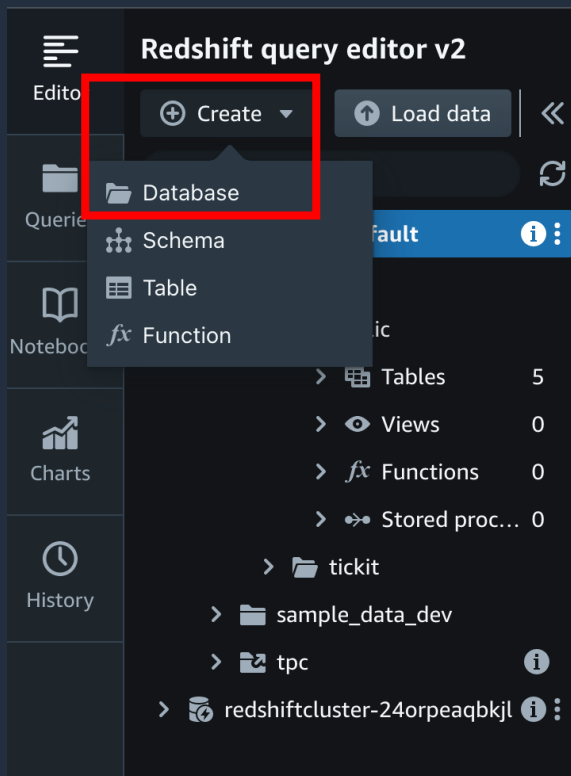
# すべてのデータを分析に活用

## DWH にデータを簡単ロード

**1** データをロードする  
データベースを簡単に作成

**2** テーブルもGUIで簡単に作成

**3** ローカルのデータを直接ブラウズしてロードすることが可能 (※)



# 自動化された運用タスク

# 一般的な DWH の運用タスク

- バックアップ
- リストア
- 監視 (システム/ワークロード)
- アクセス監査
- バージョンアップグレード
- キャパシティ管理

など

対応には相応のコストと工数が必要

# 一般的な DWH の運用タスク

- バックアップ ➡ S3 への自動(\*)/手動スナップショット
- リストア ➡ スナップショット/復旧ポイントからの復元
- 監視 (システム/ワークロード) ➡ コンソールや CLI 経由で状況を確認
- アクセス監査 ➡ 監査ログを S3 上に自動取得
- バージョンアップグレード ➡ 定期的な自動アップグレード
- キャパシティ管理 ➡ 厳密な管理は必ずしも必要ではない

(マネージドストレージで自動スケール)  
(サーバーレスでコンピューティングが自動スケール)

など

多くのタスクが自動化・簡易化される

(\*) サーバーレスは「復旧ポイント」が作成される (詳細は後述)

# テーブルの物理設計やメンテナンスタスクが 機械学習ベースで自動的に最適化

物理データ配置やその最適化を  
自動化

---

データやワークロードの増加に  
応じてピーク性能を最適化

---

機械学習を活用し、変化する  
ワークロードに適応

---

パフォーマンスチューニングの  
自動化

---



Automatic  
vacuum  
delete



Automatic  
distribution keys



Automatic  
sort keys



Auto  
workload  
manager



Automatic  
table sort



Automatic column  
compression  
encoding



Auto Analyze



Auto refresh & rewrite  
Materialized Views

# Amazon Redshift の高速化の工夫 (1)

## ゾーンマップとソートキー

### ゾーンマップ

- Amazon Redshiftはカラムごとに「ブロック」単位でディスクにデータを格納。  
1ブロック = 1 MB
- ブロック内の最小値と最大値をメモリに保存
- **不要なブロックを読み飛ばす**ことが可能

10	10   13   14   26   ...
324	...   100   245   324
375	375   393   417 ...
623	... 512   549   623
637	637   712   809 ...
959	...   834   921   959

### ソートキー

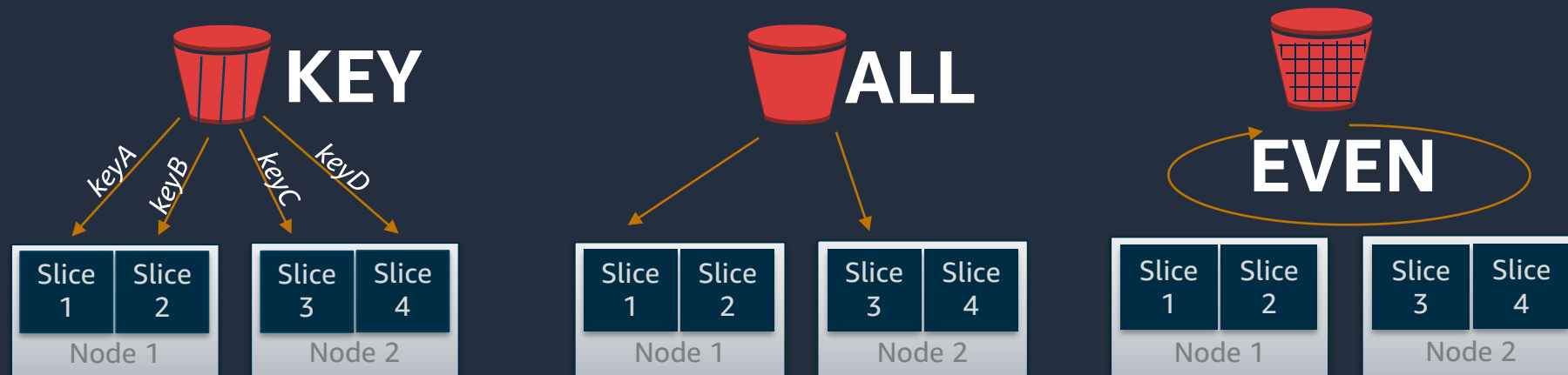
- カラムのデータをあらかじめソートしておけば、ゾーンマップによる読み込みブロックの絞り込みを効果的に行うことが可能
- クエリで頻繁に絞り込みを行う列をソートキーとして指定することで、データロードやメンテナンス時にデータをソートしておくことができる



# Amazon Redshiftの高速化の工夫（2）

## 分散スタイル

- 通常は、全ノード・全スライス（並列処理の単位）にデータを均等に分散させることで、並列処理を効率的に行い、クエリを高速に実行することが可能
- ただし、テーブルをJOINする場合は、ノード間でネットワークI/Oが多発して速度が落ちる可能性がある
- JOINするレコード同士を同じノード内に置くことで、ネットワークI/Oを削減することが可能



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

# 自動テーブル最適化

- 自動テーブル最適化機能では、Amazon Redshiftがクエリとテーブルの利用状況を継続的に監視し、ソートキーや分散キーがパフォーマンス向上に寄与すると判断すると、それらを自動設定する
- ソートキーや分散スタイルを指定せずに新規作成されたテーブルでは、自動テーブル最適化はデフォルトで有効
- 自動テーブル最適化が適用されていないテーブルでも、ALTER TABLE文であとから有効にすることが可能

```
ALTER TABLE table_name ALTER SORTKEY AUTO;  
ALTER TABLE table_name ALTER DISTSTYLE AUTO;
```



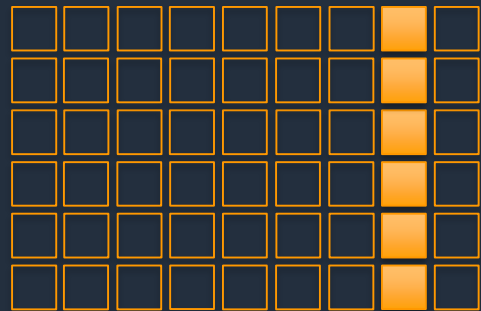
Sort Keys



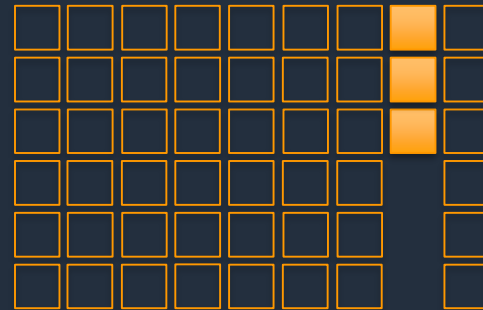
Distribution Style

# 圧縮エンコードの自動管理

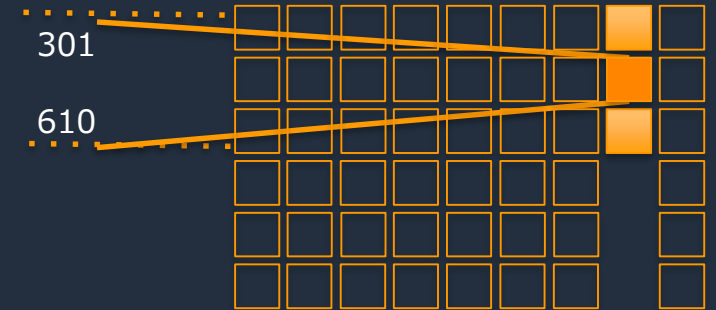
- 列の圧縮を行うことで、一度のディスクアクセスで読み込めるデータ量が多くなり、速度の向上が見込める
- 圧縮のエンコード (アルゴリズム) が複数用意されており、データ型やソーキー/非ソーキーに応じて自動選択される
- CREATE TABLE / ALTER TABLE で各列に明示選択/変更することも可能



列指向  
カラムナーフォーマット



圧縮

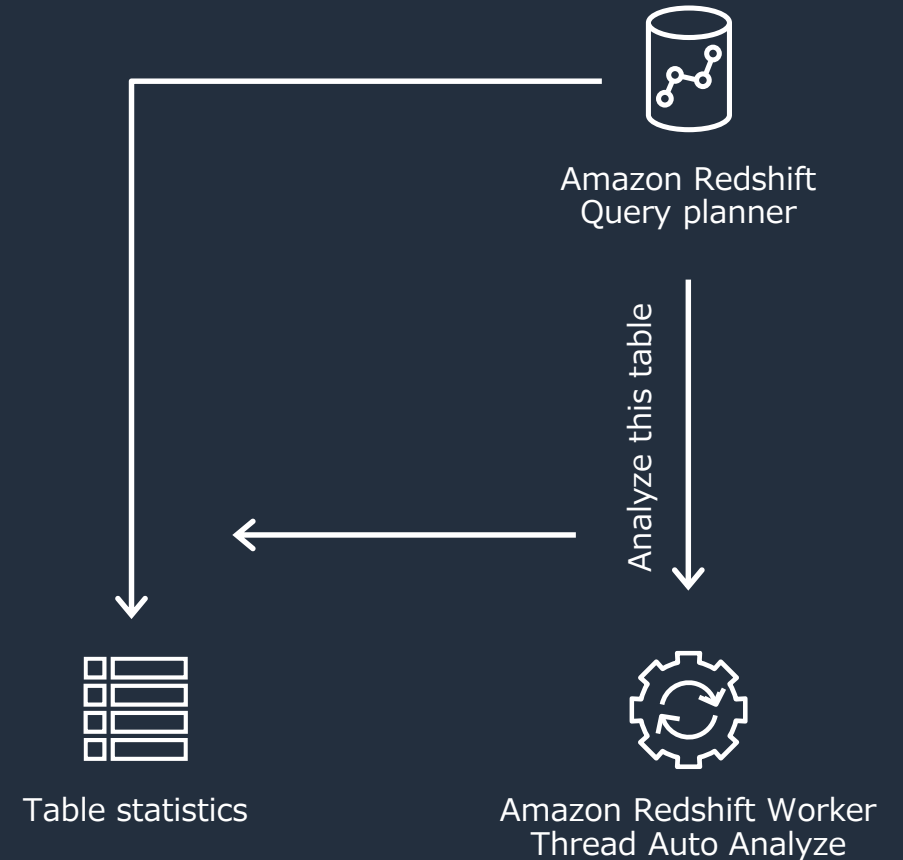


ゾーンマップ

# Analyze の自動化

Analyze は、最適なクエリ実行計画作成のために必要な、  
テーブルの統計情報を更新する処理

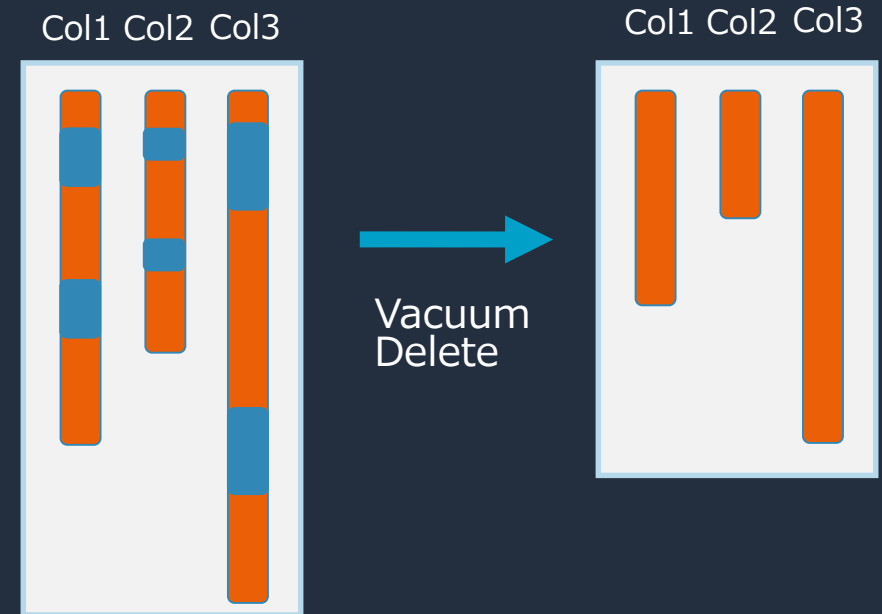
テーブルデータの変化と  
クエリの負荷状況に基づいて、  
適切なタイミングで**自動実行**される



# Vacuum Delete の自動化

Vacuum Delete は Delete や Update 後の削除領域の解放を行う処理

テーブル内の削除済み行数とクエリの負荷状況に基づいて、適切なタイミングで**自動実行**される



# テーブルソートの自動化

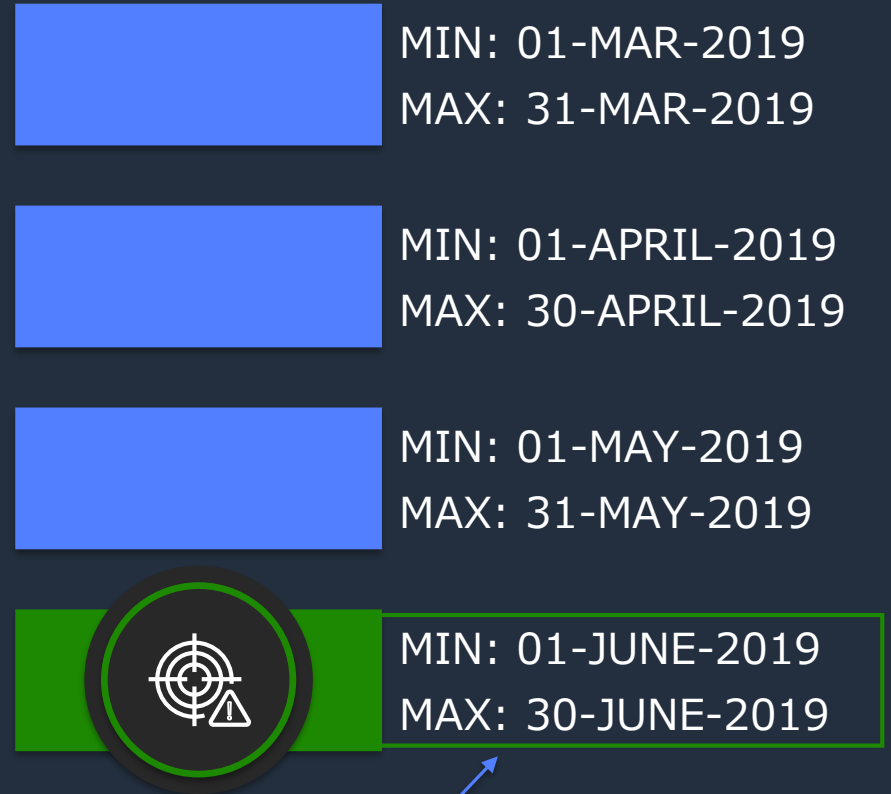
テーブルへのデータの追加・更新がされ続けると、テーブルの未ソート領域が次第に増えていくため、メンテナンスが必要となる

テーブル全体のソートは負荷が高いため、機械学習によりソートが必要な部分を判断し、ピンポイントでソートする。このため低負荷でソートを実行可能

適切なタイミングで**自動実行**される

パフォーマンスやアクセスに影響なし

## Sorted by date

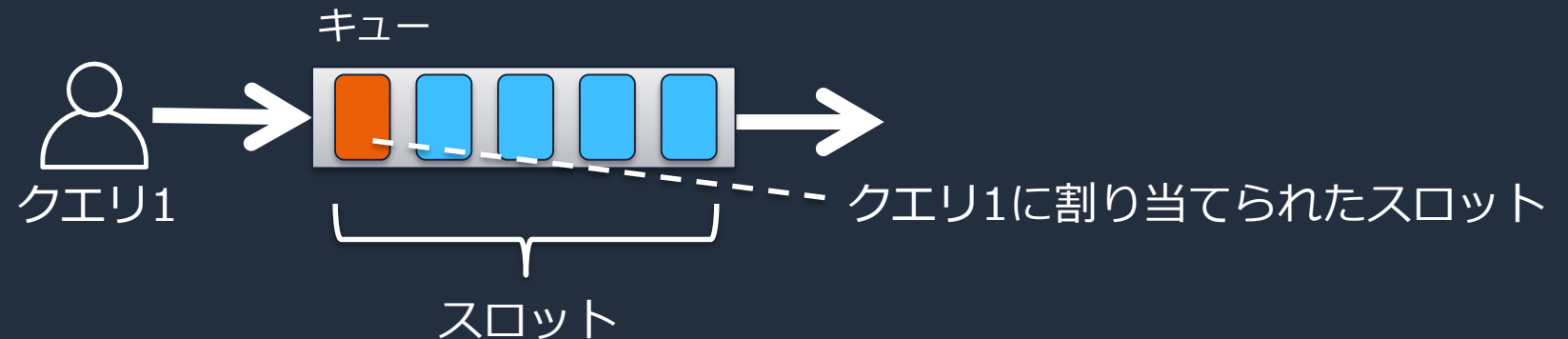


Most accessed partition

# Amazon Redshiftの高速化の工夫（3）

## ワークロード管理機能（WLM: workload management）

- Amazon Redshift がクエリに対してどのようにリソース(メモリ/CPU/IO)を割り当てるかを決定づけるしくみ
- クエリが実行されると、必ず待ち行列(**キュー**)にアサインされる。キューはデフォルトで用意されているが、別途作成して増やすことも可能で、クエリの種類(ユーザーグループ、クエリグループ)に応じてアサイン先を振り分けることが可能
- クエリには、キューで利用可能なリソースセット(**スロット**)が割り当てられる。スロットが全て割り当てられたら、後続のクエリは実行待ちとなる

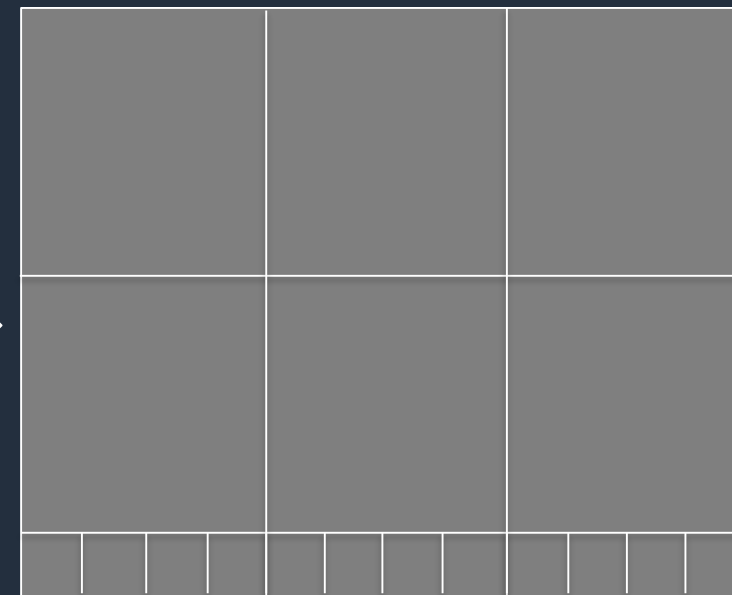


# ワークロード管理の自動化 – Auto WLM

- WLMでの複雑なリソース割当を、Amazon Redshift が自動的に行う仕組み
- クラスター全体で柔軟にリソースを使用
- クエリの内容に応じて、割り当てるメモリを適切に調整
- クエリ単体のパフォーマンスとクエリの同時実行数のバランスを自動的に最適化



クラスターの総メモリ：40GB





# 監視・パッチ適用の自動化

左 : Provisioned、右 : Serverless

- 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

- パッチ適用も自動的
- Provisioned : メンテナンスウィンドウでパッチ適用の時間帯を指定可能
- Serverless : メンテナンスウィンドウはなく、更新はシームレスに処理される



## Provisioned

### メンテナンスの詳細を編集する

▼ 設定オプション

Maintenance window schedule

- メンテナンスウィンドウを変更する  
クラスターに保留中の通知またはメンテナンスを適用する毎週の時間範囲の開始時刻を選択します。
- メンテナンスウィンドウを遅らせる  
クラスターのメンテナンス開始を遅らせる必要がある場合は、メンテナンスを最大 45 日間延期できます。遅延を指定しても、ハードウェアを更新する必要がない限り、クラスターのメンテナンスは行われません。

Schedule maintenance window

日 時刻

水曜日 19:00 UTC+09:00

メンテナンストラック

メンテナンストラックは、メンテナンスウィンドウ中に適用される各クラスターバージョンを制御します。

- メンテナンストラックへの変更は、今後のメンテナンスウィンドウで Amazon Redshift クラスターに適用されます。[クラスターステータス]の[保留中の修正]に[メンテナンストラック]と表示されるまでは、リクエストをキャンセルし、元のトラックに戻ることができます。詳細はこちら

最新

最新の承認済みクラスターバージョンを使用します。

- トレーニング  
現在のバージョンの前のクラスターバージョンを使用します。
- プレビュー  
新しい機能のベータリリースでクラスターバージョンを使用します。

# クラスターの一時停止と再開でコスト削減

- Amazon Redshift を使用しない間（週末等）はクラスターを一時停止（Pause）し、必要時にクラスターを再開（Resume）することでコスト発生を抑制可能
- スケジュール実行で自動的に一時停止・再開させることも可能

Amazon Redshift > クラスター > redshiftcluster-24orpeaqbjkl

redshiftcluster-24orpeaqbjkl

アクション ▲ 編集 パートナー統合を追加 クエリデータ ▼

クラスタを管理

サイズ変更

再起動

一時停止

削除

メンテナンスを延期

パブリックアクセス可能な設定を変更

バックアップと災害対策

テーブルの復元

スナップショットを作成

クロスリージョンスナップショットを設定

再配置

許可

IAM ロールの管理

管理者ユーザーパスワードの変更

一般的な情報

クラスター識別子	状態	ノードの
redshiftcluster-24orpeaqbjkl	Available	ra3.4xlarge
クラスター名前空間	作成日	ノードの
bb16a89c-faa3-485f-80e7-67ea19d68c07	March 18, 2022, 19:58 (UTC+09:00)	2
クラスター設定	使用するストレージ	
Production	0.02% (256 TB 中0.05 を使用)	
	Multi-AZ	
	いいえ	

クラスターのパフォーマンス | クエリのモニタリング | データベース | データ共有 | スケジューラ

Amazon Redshift > クラスター > スケジュールの再開を一時停止する

redshiftcluster-24orpeaqbjklを一時停止

クラスタを一時停止

今すぐ一時停止

後で一時停止

スケジュールどおりに一時停止して再開する

クラスタを一時停止すると、クエリに使用できなくなり、モニタリング、メンテナンス、請求に影響します。 [詳細はこちら](#)

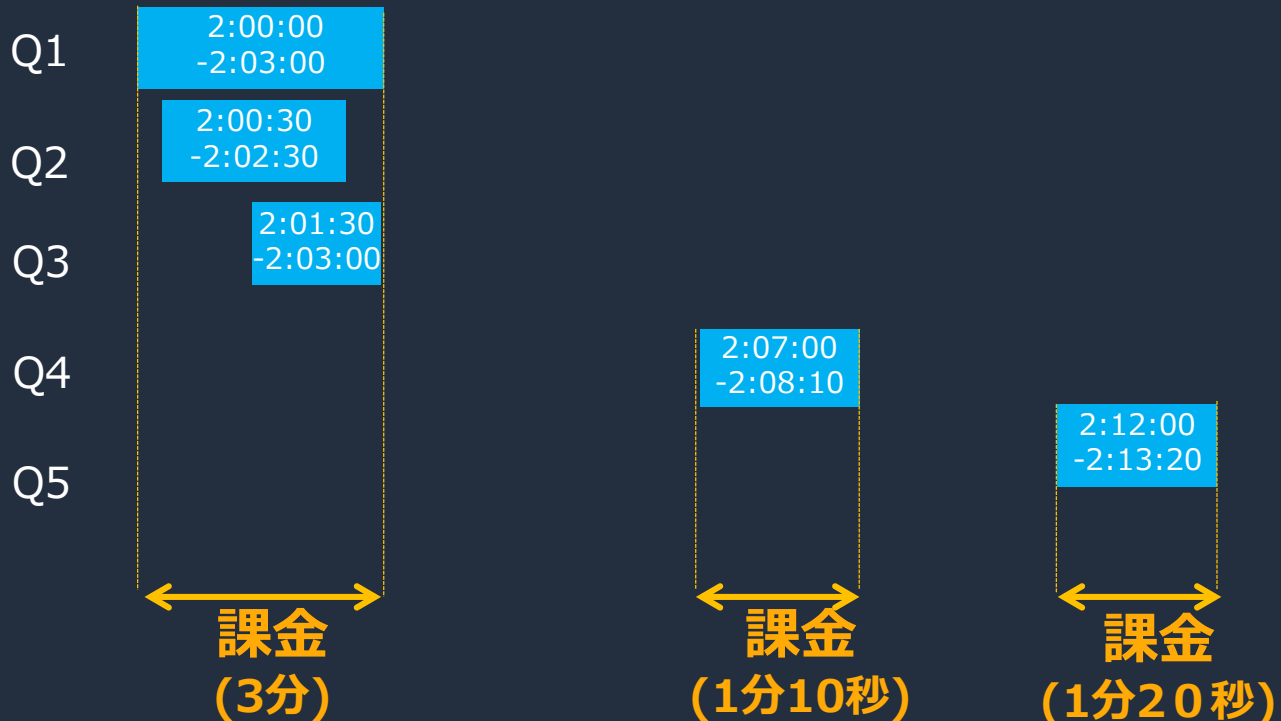
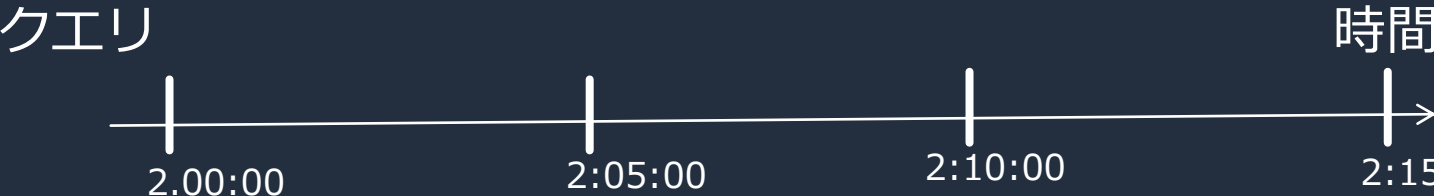
このオペレーションをキャンセルすることはできません

このクラスタを一時停止しますか?

キャンセル

# 使った分だけのお支払い

クラスタ管理が不要な Serverless はワークロード実行期間のコンピューティング容量利用のみ課金（秒単位）



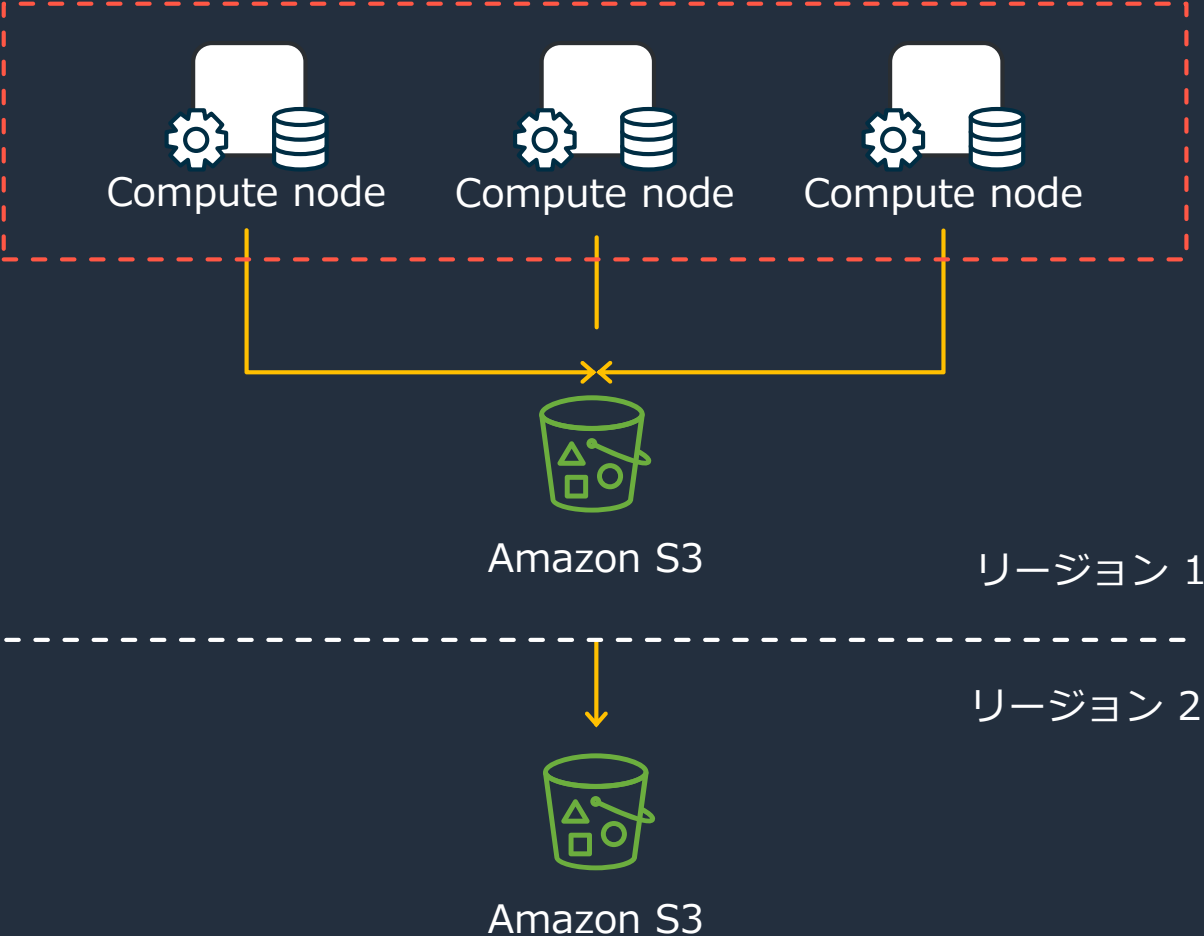
課金対象期間	クエリ実行時間
@2:03	3分 (for Q1, Q2, Q3)
@2:09	1分10秒 (for Q4)
@2:14	1分20秒 (for Q5)
課金対象合計	5分30秒

## アイドル期間は課金対象外

- 同時実行スケーリングやデータレイクへのクエリも包含（別途追加料金なし）
  - 60秒の最低料金あり
- <https://aws.amazon.com/jp/redshift/pricing/>

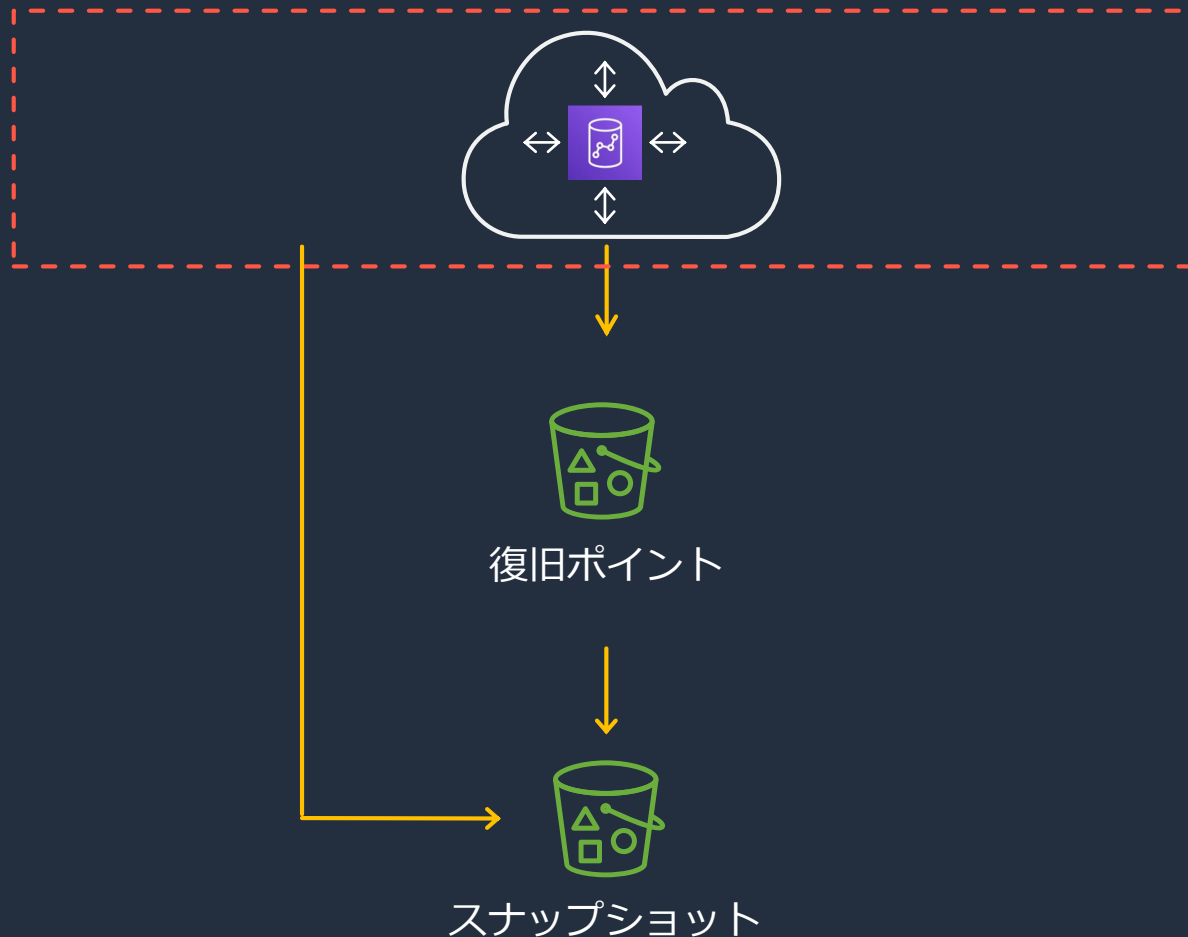


# 継続的な自動データバックアップ



- Amazon S3 に継続的な増分バックアップスナップショットを自動取得
- バックアップ&リストア中も SQL を実行可能
- リージョン間スナップショットコピーも可能
- スナップショットは Serverless にもリストア可能
- ストリーミングリストアにも対応

# 継続的な自動データバックアップ

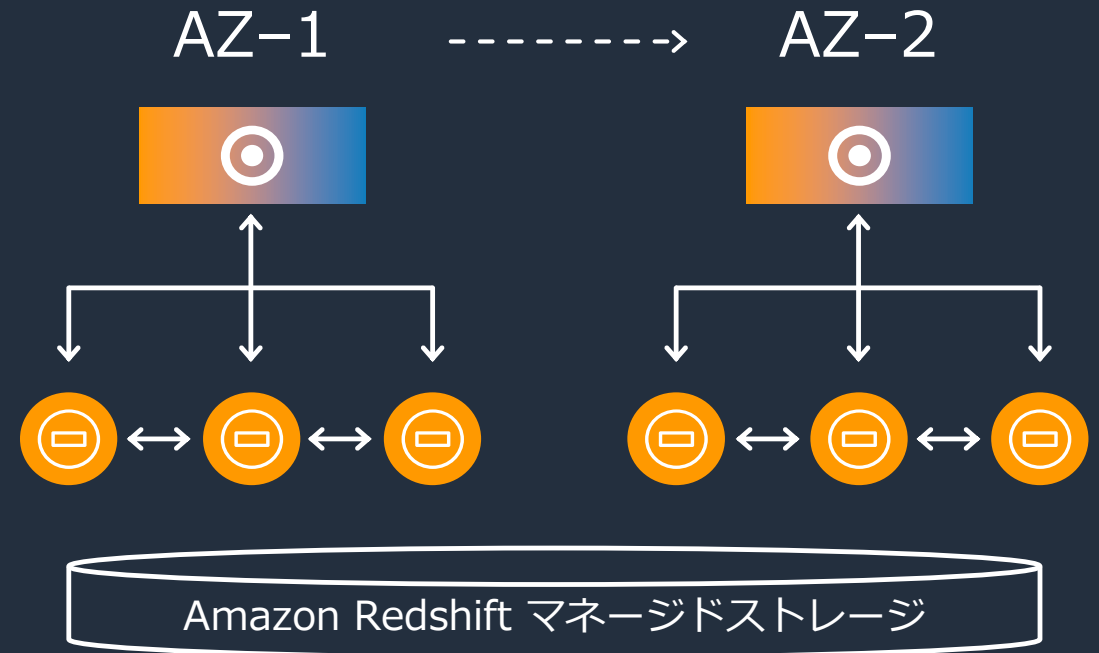


- 復旧ポイントが 30 分ごとに自動作成され、24 時間保持される
- 復旧ポイントからは、直接復旧することも、スナップショットを作成することもできる
- 手動でスナップショットも作成できる
- スナップショットは Provisioned にもリストア可能
- 復旧は 2 段階で行われ、数分でクエリが使用できるようになる

# Availability Zone(AZ) 障害時はマネージドストレージを使って自動復旧

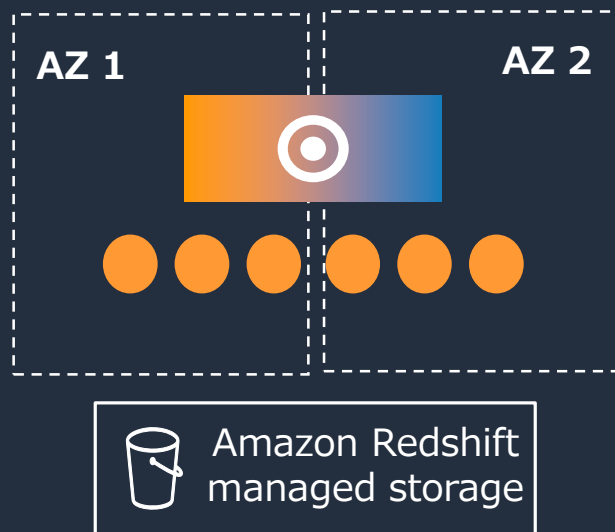
## クロス AZ クラスターリカバリー

- ✓ データロスなしにリカバリー (RPO = 0)
- ✓ スナップショットからのリストアが不要に
- ✓ オンデマンドフェールオーバー
- ✓ クラスターは別の AZ にオンデマンドで作成されるため、スタンバイレプリカクラスター不要
- ✓ RA3 インスタンスでのサポート



# Multi-AZ デプロイメント

ビジネスクリティカルアプリケーション向けの高可用性と耐障害性



- 1つのエンドポイントを持つ単一のデータウェアハウスとしてアクセスでき、ワークロード処理を複数のAZに自動的に分散
- AZ 障害の場合
  - ✓ 手動操作不要で自動フェイルオーバー
  - ✓ データ損失なし
  - ✓ 単一のエンドポイントのため、障害時も接続先切り替え不要
- サポートは RA3 クラスターのみ
- 対応リージョン
  - US East (N. Virginia), US East (Ohio), US West (Oregon), **Asia Pacific (Tokyo)**, Europe (Ireland) and Europe (Stockholm)

# モニタリングと性能改善

実際にどのようにチューニングするのかの具体的な手法は、  
パフォーマンスチューニング編を参照ください



# Amazon Redshiftの主なモニタリング項目

- クラスターのパフォーマンスメトリクス
  - Amazon CloudWatchによる監視メトリクスを提供
  - 管理コンソールからも確認可能
- クラスターのイベント
  - クラスターで発生したエラーや、クラスターのメンテナンス予告、パッチ適用予告などがイベントとして登録される
  - Amazon SNS を利用して通知することも可能
- システムテーブルとビュー
  - 個別クエリの実行統計や、テーブルの構成情報などのメタデータを確認可能

# アドバイザー機能

- Amazon Redshift がワークロードを自動分析し、パフォーマンス向上やコスト削減につながる推奨事項を表示する機能
- アドバイスが出ていないかを監視し、出ていれば優先して実行を検討

## 推奨事項の例：

- COPY コマンド実行時のファイル圧縮・分割
- 未使用クラスターの停止やダウンサイジング
- 設定すべきソートキーや分散キー（自動テーブル最適化を使っていない場合にはこちらを参照すると便利）

The screenshot shows the Amazon Redshift console interface. On the left is a navigation sidebar with options like 'Redshift サーパーレス 新規', 'プロビジョニングされたクラスターダッシュボード', 'クラスター', 'クエリエディタ', 'クエリエディタ v2', 'クエリとロード', 'データ共有', '設定', 'AWS Partner Integration', 'Informatica Data Loader', 'アドバイザー', 'AWS Marketplace', 'アラーム', 'イベント', and '最新機能'. The main content area is titled 'Amazon Redshift > 推奨事項' and shows a recommendation for cluster 'redshiftcluster-24orpeaqbjl (1)'. The recommendation is 'Improve Query Performance and Optimize Storage with Compression Encoding', dated February 12, 2023. It includes a '確認した内容' (Verified content) section with an analysis of the cluster's workload and database schema, and a 'できること' (What you can do) section with SQL statements to apply the recommended compression encodings. A '確認した内容' section explains that compression is a column-level operation that reduces the size of data when it's stored, conserving storage space and improving query performance by reducing disk I/O. A 'できること' section provides SQL statements to apply the recommended compression encodings. A '確認した内容' section explains that compression is a column-level operation that reduces the size of data when it's stored, conserving storage space and improving query performance by reducing disk I/O. A 'できること' section provides SQL statements to apply the recommended compression encodings. A '確認した内容' section explains that compression is a column-level operation that reduces the size of data when it's stored, conserving storage space and improving query performance by reducing disk I/O. A 'できること' section provides SQL statements to apply the recommended compression encodings.

# クエリーやデータベース、リソースのモニタリング

- マネジメントコンソールの活用
  - マネジメントコンソール → Redshift → クラスター → [クラスター名]
  - 以下の情報を得ることができる

## クラスターのパフォーマンス

CPU、ディスクI/O、ネットワークI/Oなど

## クエリーの履歴

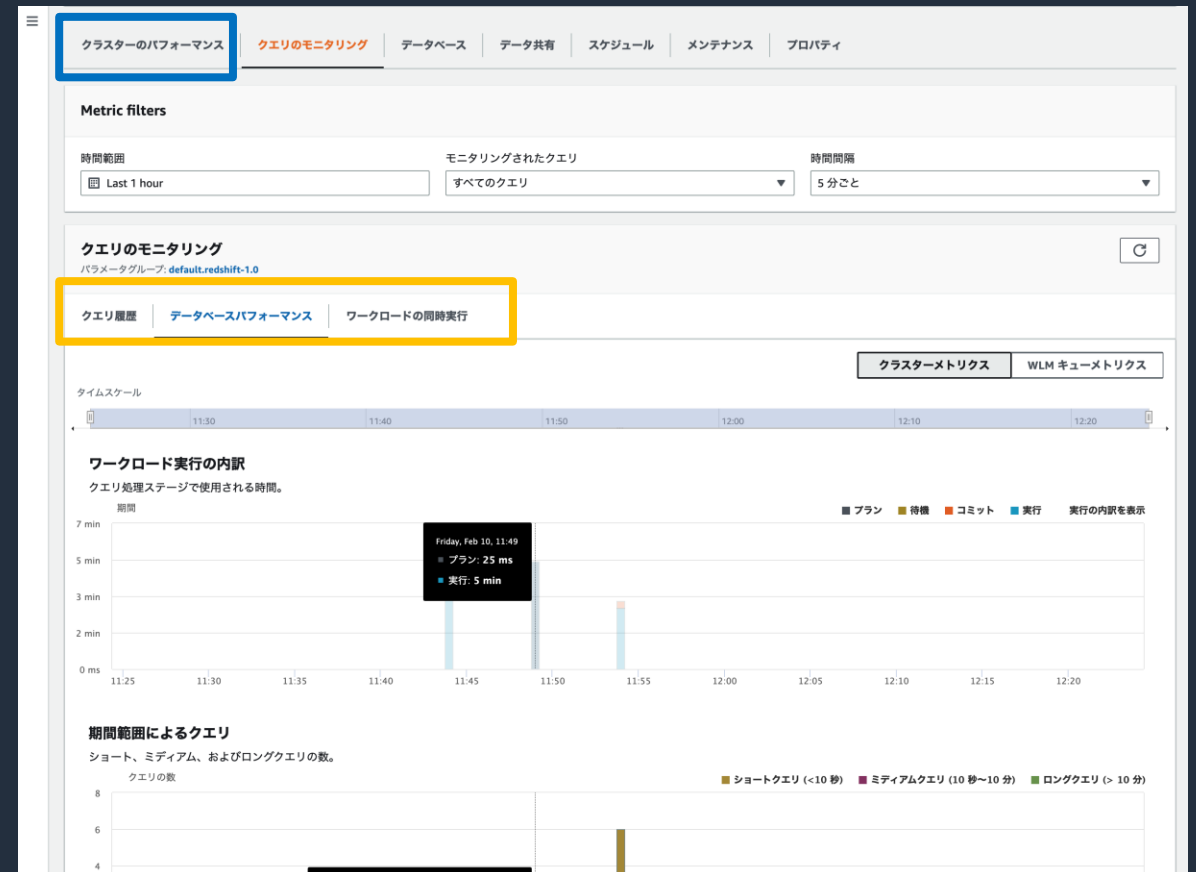
クエリーやロードの実行時間やステータスなど

## データベースパフォーマンス

スループット、レイテンシーなど

## ワークロードの同時実行

クエリーの同時実行状況 (キュー待ちや実行時間)



# クエリーやデータベース、リソースのモニタリング

- マネジメントコンソールの活用
  - マネジメントコンソール → Redshift (Redshift サーバーレス)
  - 以下の情報を得ることができる

## クエリーの履歴

クエリーやロードの実行時間やステータスなど

## データベースパフォーマンス

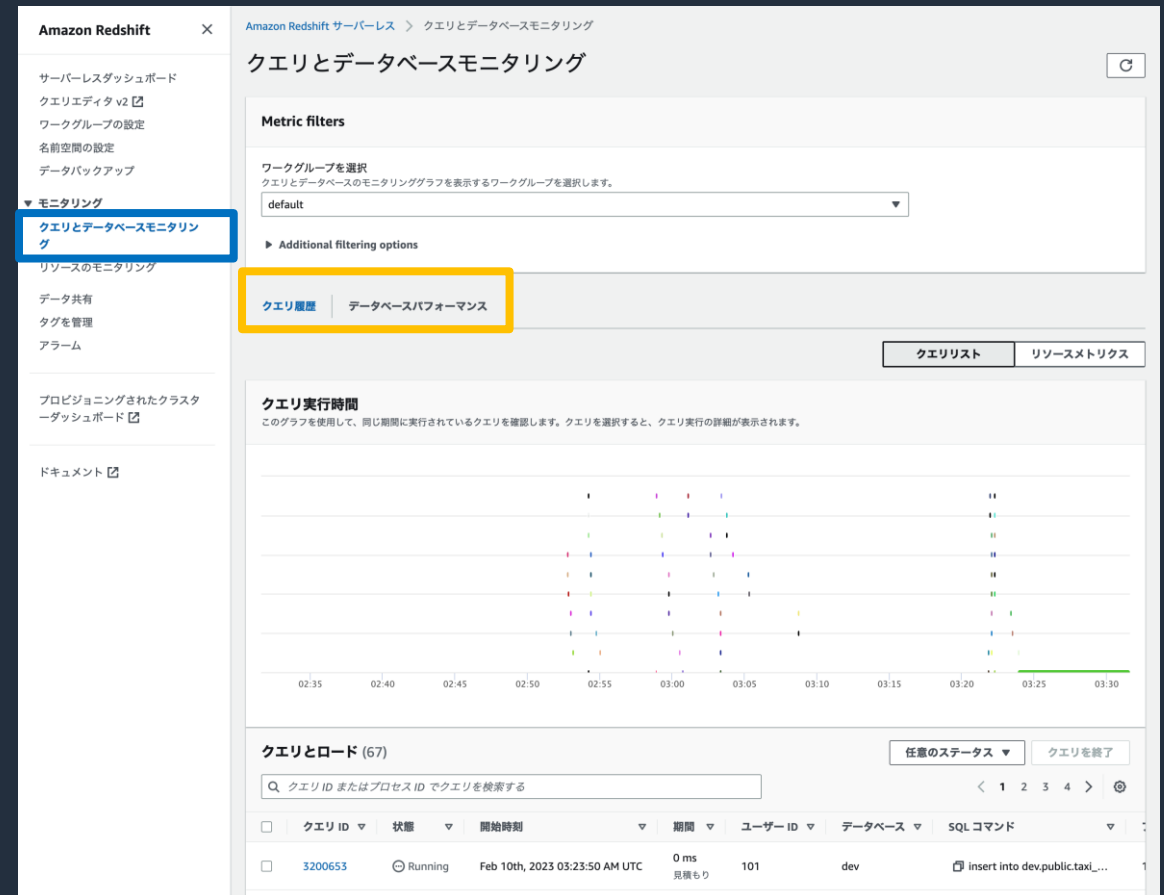
スループット、レイテンシーなど

## リソースのモニタリング

RPU(\*)使用容量、コンピューティング使用状況

CPU使用率などクラスタの物理的なパフォーマンスに関するメトリクスは存在しない

(\*) Redshift Proceccing Unit



# システムテーブルとビューによる性能分析

- Provisioned

- 発生元や用途によって異なる4種類のシステムテーブル/ビューを保持
- いずれも DB 領域外に保持されるためスナップショットには含まれない (スナップショットからの復元では残存していない)

	STL	STV	SVL	SVV
タイプ	テーブル	仮想テーブル	ビュー	ビュー
生成方法	ディスク上のログから生成 (7日間保持)	オンメモリーデータから生成	STLへの参照	STVへの参照
主たる用途	過去の実行記録の参照	現在進行中の処理の参照	STL/STVデータを組み合わせて別軸で分析	

- Serverless

- SVV\_\* の中で主にテーブルのメタデータに関するもののみが参照可能
- クエリの実行統計やデータロード履歴などは、SYS\_\* ビュー で確認できる

# システムテーブルとビューによる性能分析

- 実行されたクエリには一意のクエリ ID が割り当てられる
- システムテーブル・ビューから情報を取得する際は、多くの場合クエリ ID をキーとする
- Tips
  - (Provisioned) 同一セッションで情報を取得する場合は以下が便利
    1. クエリを実行
    2. `select pg_last_query_id() as q into temp table hoge;`
    3. `select * from stl_explain where query=(select q from hoge);`
    4. `select * from svl_query_summary where query=(select q from hoge);`
    5. :

# Admin Views / Scripts

- 管理、モニタリングに有益な View およびスクリプト群
  - AWSにより開発・提供される
  - 無料で利用可能  
<https://github.com/awslabs/amazon-redshift-utils>
- クラスターの状態確認、トラブルシューティングに利用可能
  - 例：ノードスライス間でのデータ分散状況の確認

# クエリが性能要件を満たせない場合は

## クエリのチューニング

- クエリの実行計画・実行統計を見て非効率な動作を確認した場合
- テーブル設計の変更や SQL の修正など

→ 圧縮、ソートキー、分散スタイルといった Amazon Redshift 特有のテーブル物理設計の最適化について、現在は大部分自動化されている

## リソースの増強（サイズ変更）

- Provisioned : クラスタのスケールアウト（またはスケールアップ）
- Serverless : ベースキャパシティ（Base RPU capacity）を増やす
- 夜間バッチ処理など、性能が必要な時間帯が決まっている場合は、その時間だけリソースを増やす



# マテリアライズドビューによるクエリ高速化

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
- ベーステーブルとは異なる分散キーやソートキーを指定できる
- 外部表の参照も可能

ETL/BI パイプラインの簡素化

- データは自動でリフレッシュされる

クエリリライト

- アプリケーションからはテーブルに対してクエリするだけで必要に応じて実行計画を Rewrite し、マテリアライズドビューを参照

Auto MV : 自動マテリアライズドビュー

- Amazon Redshift が継続的にワークロードをモニターし、機械学習によりパフォーマンス面で最適なマテリアライズドビューを自動的に生成・管理

Materialized View

loc_sales	
loc	total_sales
SF	12.00
NY	10.00

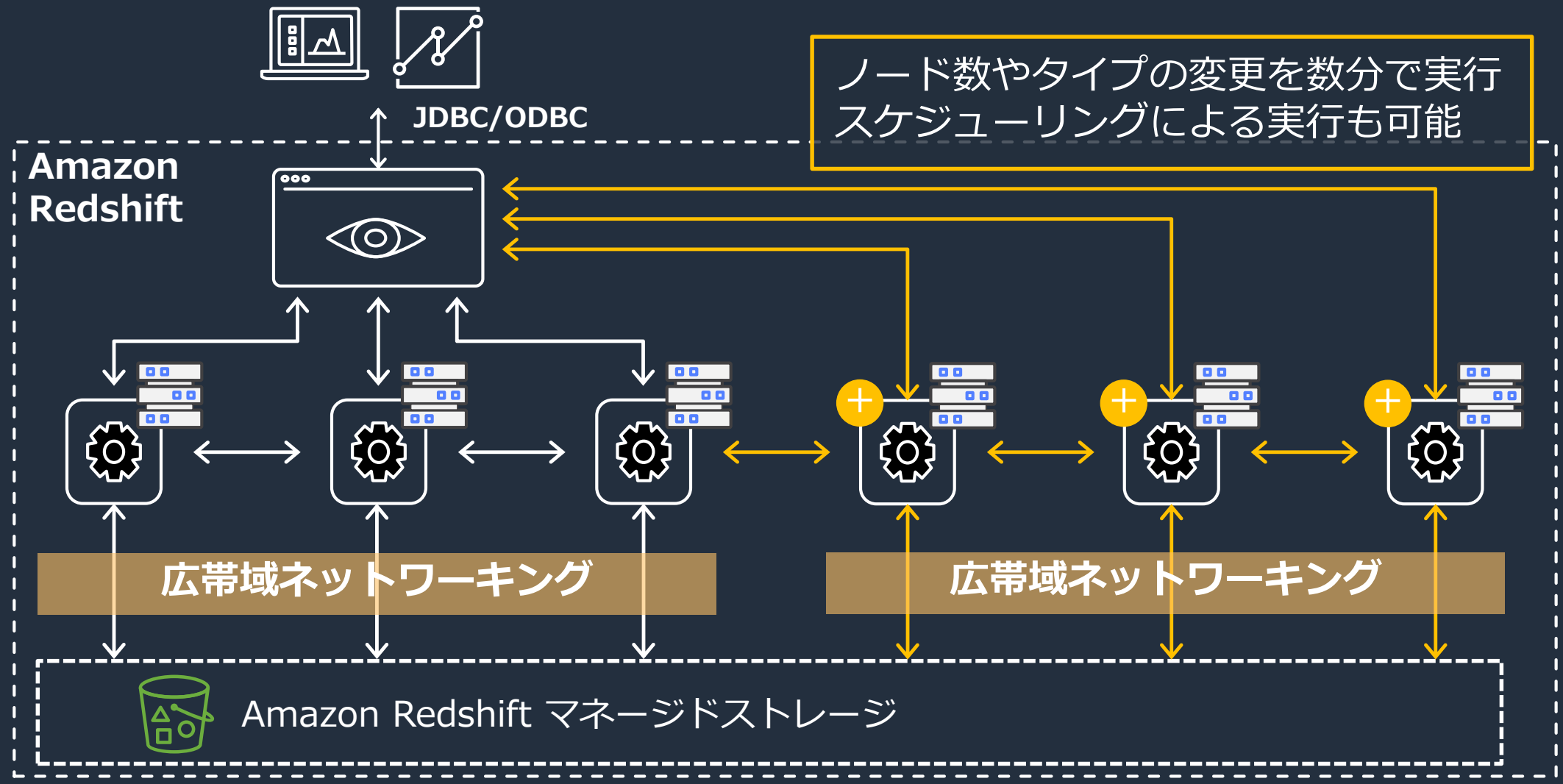


“地域ごとの  
総売上は?”

store_info		
store	owner	loc
s1	Joe	SF
s2	Ann	NY
s3	Lisa	SF

sales			
item	store	cust	price
i1	s1	c1	12.0
i2	s2	c1	3.0
i3	s2	c2	7.0

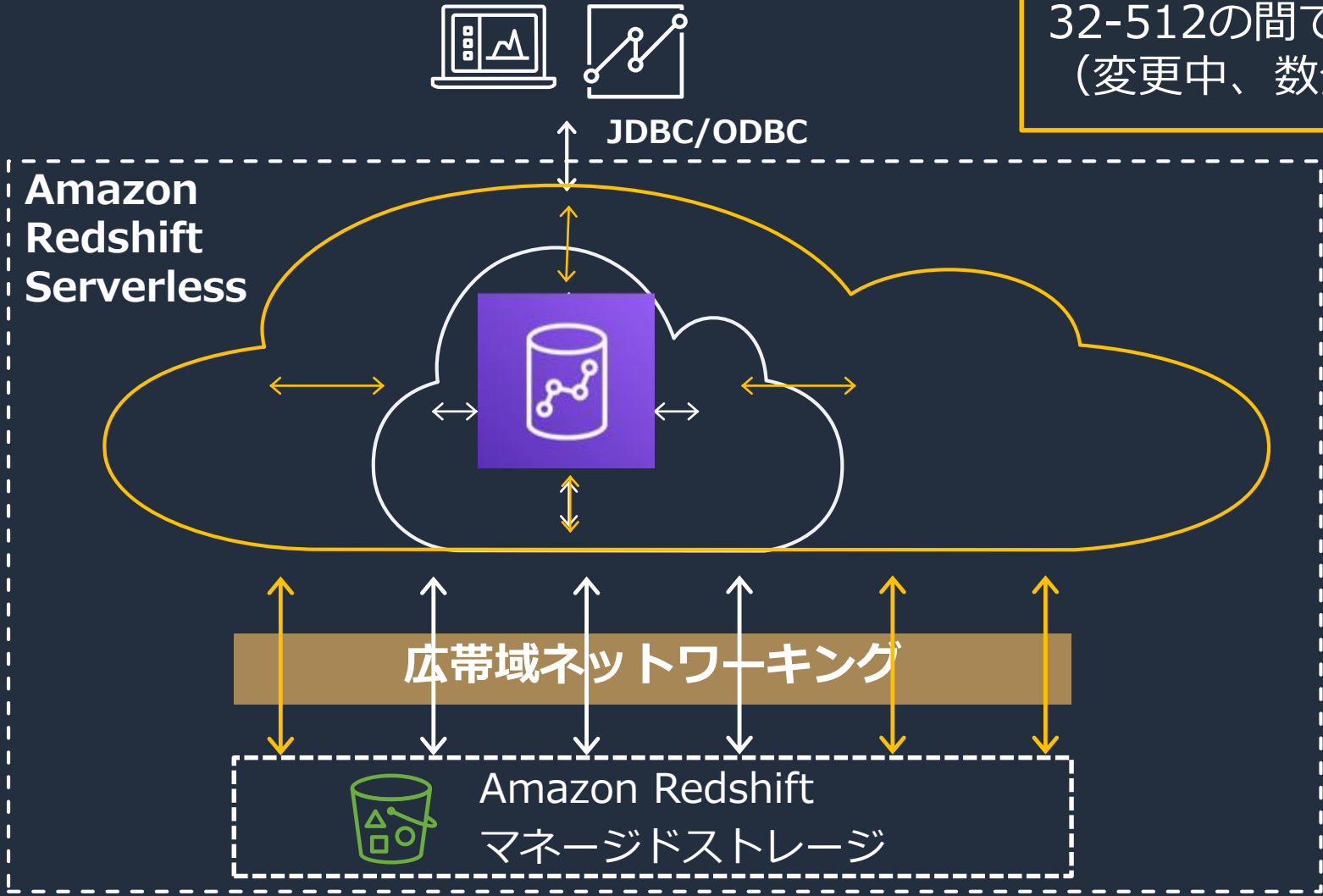
# 必要な性能に合わせてノード構成を変更する Elastic Resize



# 必要な性能に合わせてノード構成を変更する

## Base RPUの変更

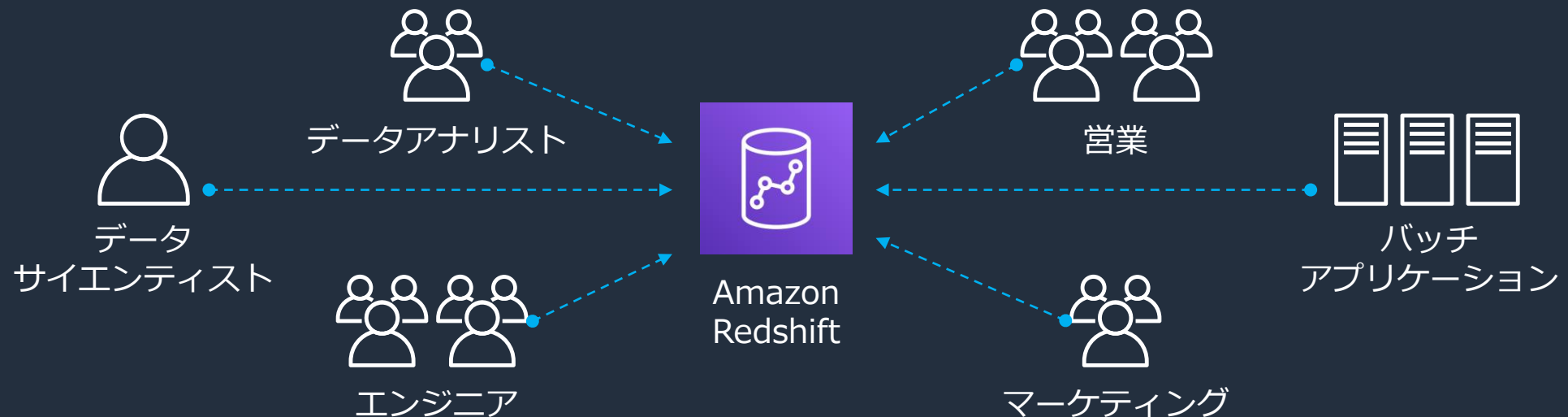
32-512の間で、8単位に変更できる  
(変更中、数分間は利用できない)



# 利用者の増加に対処する

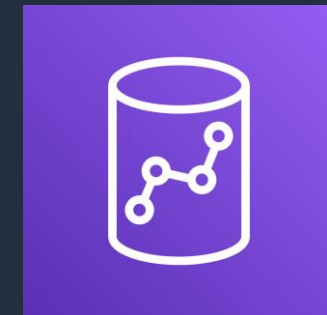
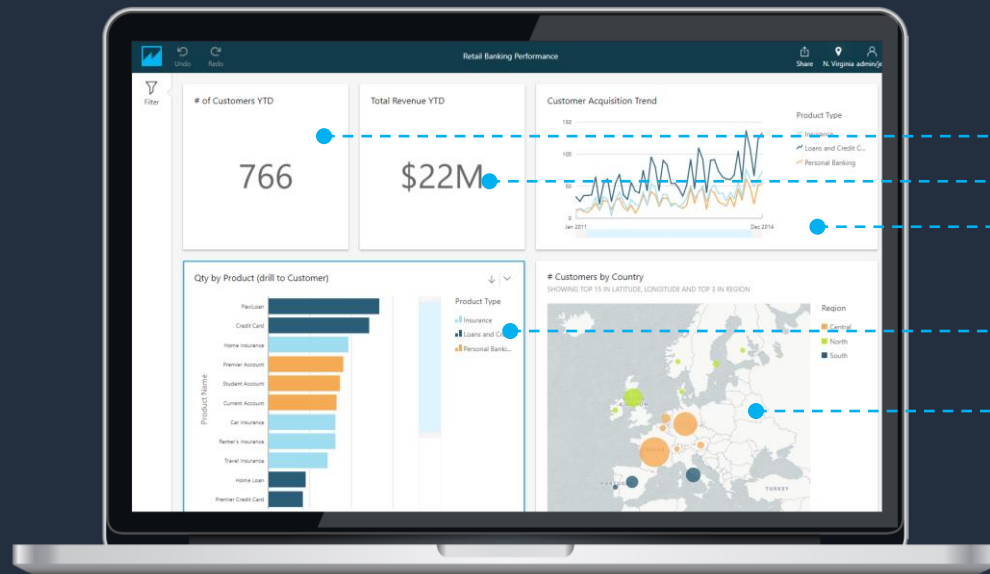
# DWH ワークロードの特徴

- DWHには多様なユーザーが様々なツールを使って接続する可能性がある
- 評判のよい DWH は成長していく



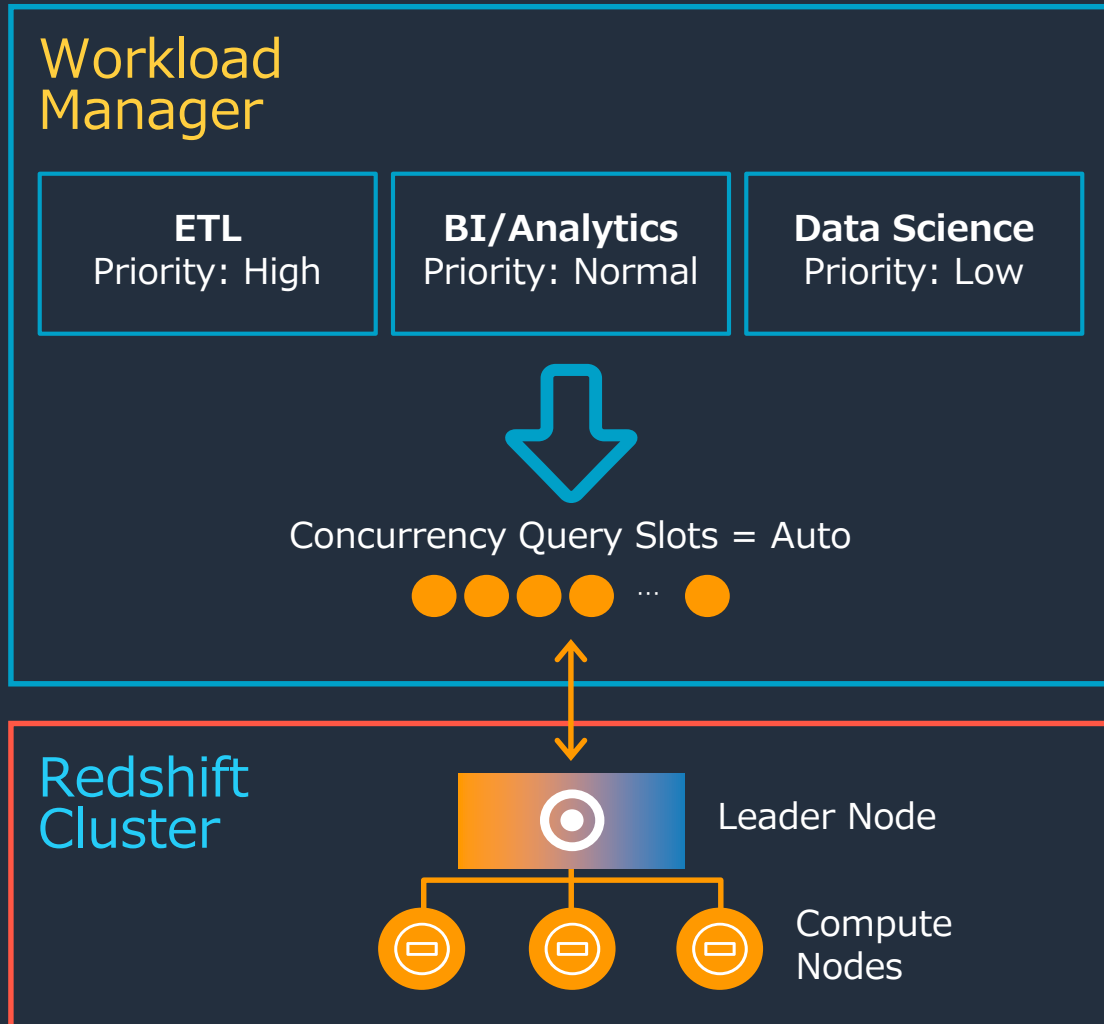
# DWH ワークロードの特徴

- BI ダッシュボードからはバックグラウンドで複数のクエリが流れる
- DWH は接続ユーザー数以上のリクエストを受けられる可能性がある



Amazon  
Redshift

# 重要なクエリを優先的に実行



Auto WLMでは、ワークロード毎にキューを分け、相対的な優先度を設定することが可能

Amazon Redshift が、優先度の高いキューで実行された処理に自動的に多く CPU, IOリソースを割り当てる

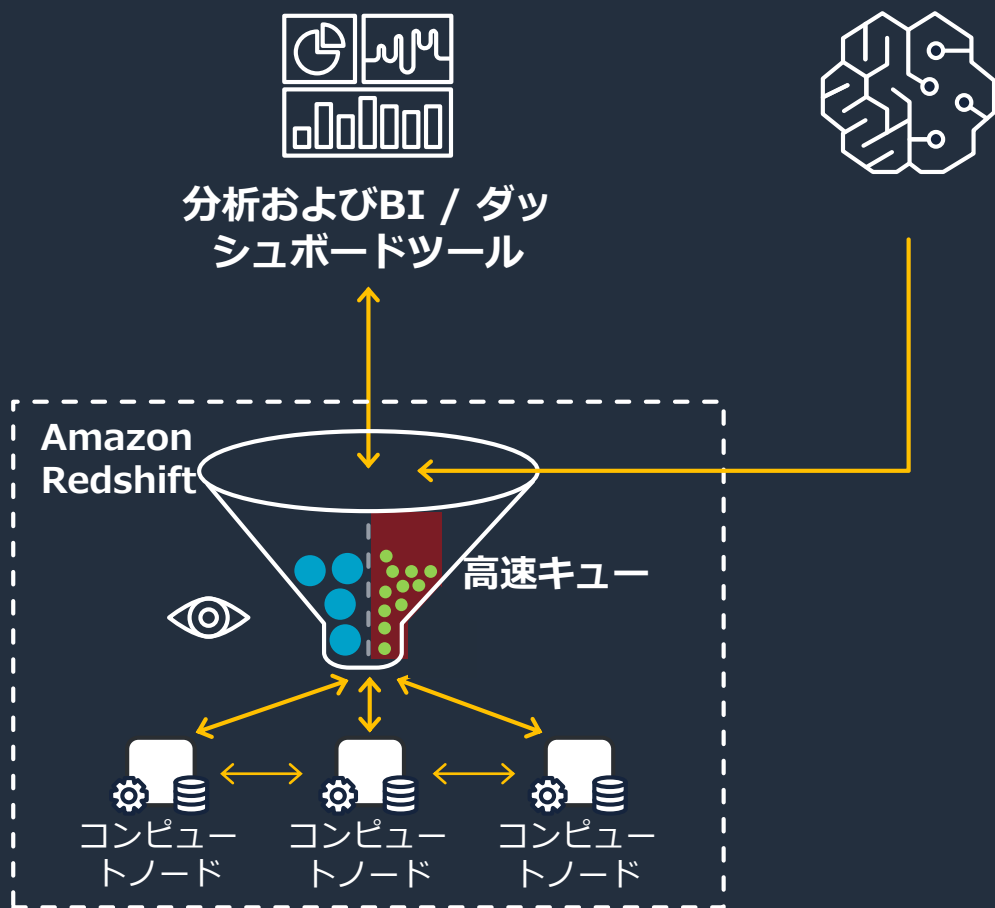
優先度はコマンドなどで動的に変更が可能

## Serverless

コンピューティングリソースが動的に自動管理されるので設定不要

# ショートクエリアクセラレーション (SQA)

実行時間の短いクエリは、通常のキューとは別の領域で処理



## SQAの機能

- 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)



# ピークタイムのワークロード

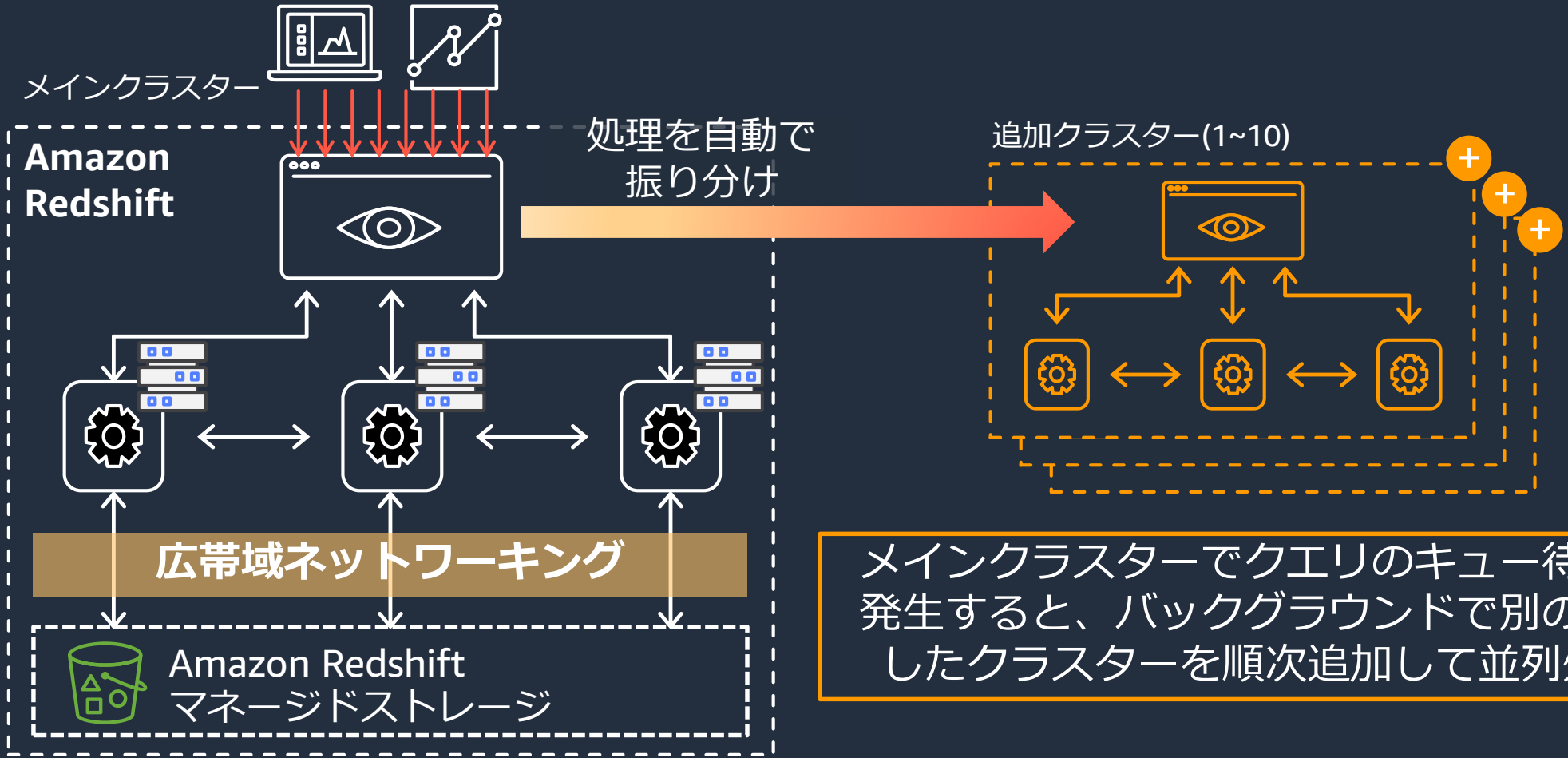
- アクセスの集中する時間帯では、システム全体のパフォーマンス低下を引き起こす可能性がある
- ピークタイムに合わせた DWH のリソースサイジングが必要になる



Serverless

コンピューティングリソースが動的に自動管理されるのでピークタイムのサイジングは不要  
(ただし、コスト管理を目的に、日次/週次/月次で RPU 時間を制限することはできる)

# ピーク時にコンピューートを自動拡張する Concurrency Scaling



メインクラスターでクエリのキュー待ちが発生すると、バックグラウンドで別の独立したクラスターを順次追加して並列処理

Serverless

コンピューティングリソースが動的に自動管理されるので設定不要



# Concurrency Scaling を有効化するには

クラスターに設定されているパラメータグループの WLM のキュー定義を編集し、「同時実行スケールモード」設定値を「自動」に設定

The screenshot shows the Amazon Redshift console interface. On the left sidebar, the '設定' (Settings) menu is expanded, and 'ワークロード管理' (Workload Management) is selected. The main content area displays the 'ワークロード管理' (Workload Management) page for a specific parameter group. The 'パラメータグループ (1/3)' (Parameter Group) section shows a list of groups, with 'concurrency-scaling-enable' selected. The 'ワークロードキュー' (Workload Queue) section is highlighted, and the 'ワークロードキューを編集' (Edit Workload Queue) button is visible. The 'WLM モードを切り替え' (Switch WLM Mode) button is also present.

The screenshot shows the 'ワークロードキュー concurrency-scaling-enable を変更します。' (Change workload queue concurrency-scaling-enable) page. The '同時実行スケールモード' (Concurrency Scaling Mode) dropdown menu is set to '自動' (Automatic). The '保存' (Save) button is highlighted. The page also shows the 'Default queue' section and the 'クエリモニタリングルール (0)' (Query Monitoring Rules) section.

# 追加クラスターの最大数を指定

- パラメータ  
“max\_concurrency\_scaling\_clusters”  
にて、追加クラスターの最大数を  
0-10 の範囲で指定 (デフォルト : 1)

パラメータ concurrency-scaling-enable を変更します。

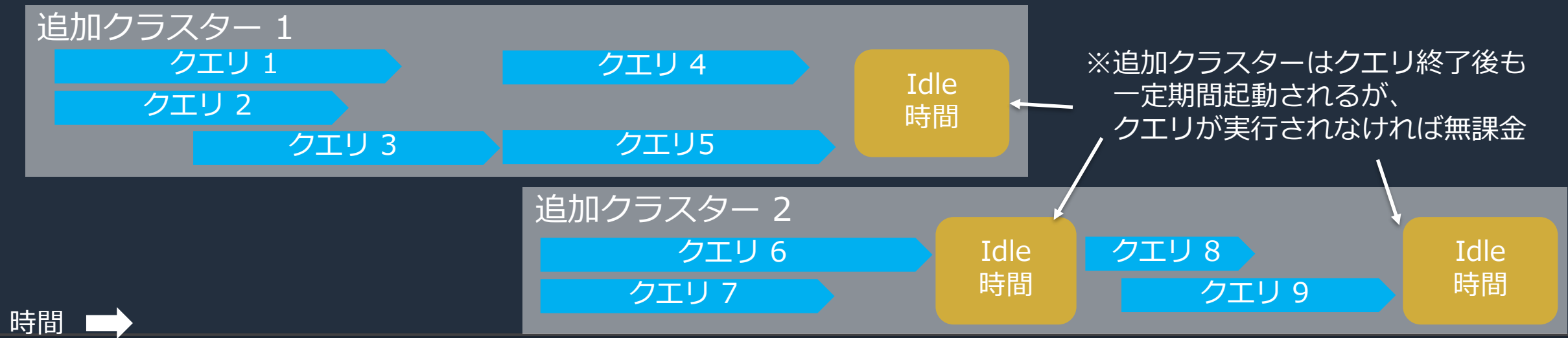
名前	値
auto_analyze boolean: true,false	true
auto_mv boolean: true,false	true
datestyle string	ISO, MDY
enable_case_sensitive_identifier boolean: true,false	false
enable_user_activity_logging boolean: true,false	false
extra_float_digits integer: -15-2	0
<b>max_concurrency_scaling_clusters</b> integer: 0-10	1
max_cursor_result_set_size integer: 0-14400000	
query_group string	default
require_ssl boolean: true,false	false
search_path string	\$user, public
statement_timeout integer: 0,100-2147483647	0
use_fips_ssl boolean: true,false	false

キャンセル

保存

# Concurrency Scaling の料金

- 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）
- 日/週/月毎の単位で実行時間上限を設定してコストをコントロール可能



## 追加クラスターへの課金

1クラスター分

2クラスター分

1

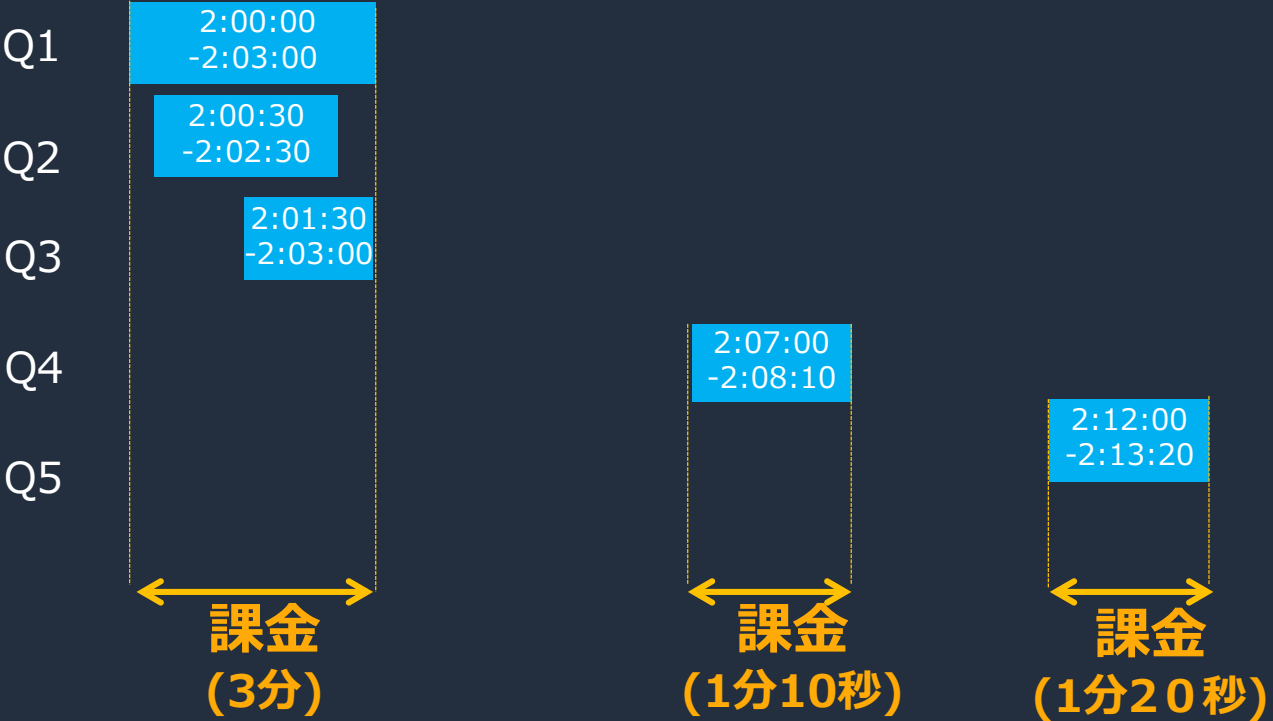
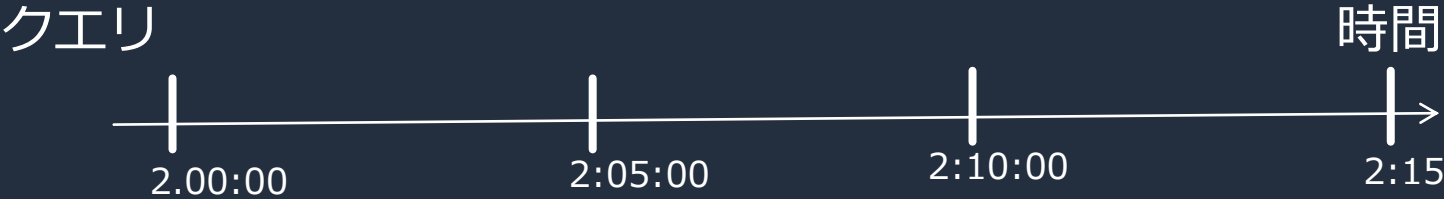
課金なし

1クラスター分

課金なし

# Serverless のコンピューティング料金 (再掲)

クラスタ管理が不要な Serverless はワークロード実行期間のコンピュートキャパシティ利用のみ課金 (秒単位)



課金対象期間	クエリ実行時間
@2:03	3 分 (for Q1, Q2, Q3)
@2:09	1 分 10 秒 (for Q4)
@2:14	1 分 20 秒 (for Q5)
課金対象合計	5 分 30 秒

## アイドル期間は課金対象外

- 同時実行スケーリングやデータレイクへのクエリも包含 (別途追加料金なし)
  - 60 秒の最低料金あり
- <https://aws.amazon.com/jp/redshift/pricing/>



# データ連携はどのように 行うか

# データロードと変換の基本

## ロード

- Amazon S3にファイルを置いて、SQLのCOPYコマンドでロードする
- データが圧縮されている場合は、均等のサイズ（1 MB – 1 GB）に分割して並列にロードする
  - Provisioned : ファイルの数がクラスターのスライスの数の倍数
  - Serverless : 数十から百程度（スライス数を確認できない）
- データが圧縮されていない場合は、ファイルを分割する必要はない（COPYコマンドが内部で分割してロードする）



ra3.16xlarge は 16 スライス/ノード



COPY



3ノードの場合、48 (=3\*16) の倍数で分割

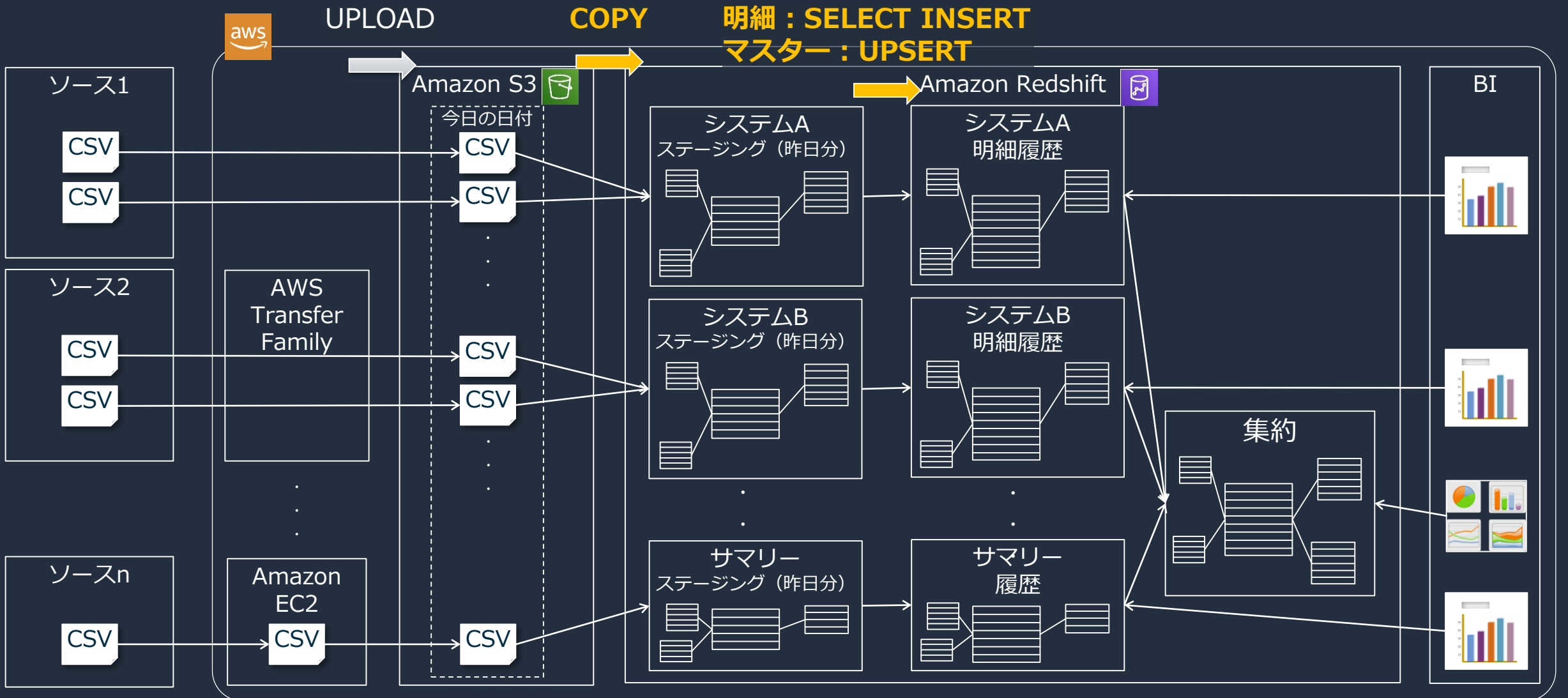
→各スライス間でデータを均等に分割して並列にロード

## 変換 (Transform)

- 外部ETLツール、AWS Glue、Hadoop 等で実行
- または、一旦 Amazon Redshiftにデータをロードし、SQLで変換



# データロード & 変換の例



# Amazon Redshift が提供する多様なデータ連携の選択肢

例) 全体像や各選択肢の概要については、概要編を参照ください

- データレイクや外部データベースのライブデータをクエリする
  - Amazon Redshift Spectrum (Amazon S3)
  - Amazon Redshift Federated Query (Amazon RDS for PostgreSQL/MySQL、Amazon Aurora)
- ETL パイプラインなしにデータを取り込む
  - Amazon Aurora との Zero-ETL 統合 (Preview)
  - Amazon S3 からの自動 COPY (Preview)
- 機械学習や分析アプリケーションから高速にアクセスする
  - Apache Spark インテグレーション
- 他の Amazon Redshift クラスターにデータを共有する
  - Amazon Redshift Data Sharing

# まとめ

# まとめ

- Amazon Redshift は、分析用クエリ実行に特化した RDBMS
- Amazon Redshiftの基本的な運用タスクは、大部分自動化されている
- ワークロードに応じて、処理能力を自動管理が可能（サーバーレス）
- 特定の用途に応じたクラスター管理も可能（プロビジョンド）
  - 単体処理の性能が足りない → 必要な時間帯だけスケールアウト
  - 一時的な利用ユーザーの増加 → クラスター追加を自動対応
- データ連携はデータロード、またはデータを動かさずに各サービスと連携

# 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!