



このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

[AWS Black Belt Online Seminar]

Amazon Redshift Update

サービスカットシリーズ

Junpei Ozono, Solutions Architect
2019/1/22

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>



自己紹介

大園 純平

Redshift Specialist Solutions Architect

役割：

- Redshift のサービス担当
- DWH や Data Lake の案件支援担当

好きなサービス：

- Amazon Redshift
- Amazon QuickSight



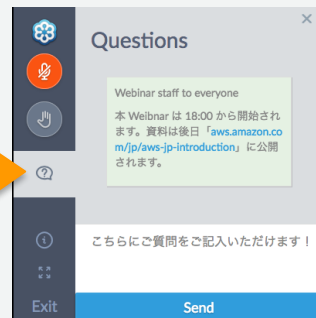
AWS Black Belt Online Seminar とは

「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2019年1月22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が東京リージョンを使用する場合、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.

本セミナーのゴール

- Redshift の概要とアーキテクチャを理解する
- Redshift の最新アップデートのキャッチアップ



本日のアジェンダ

- Redshift の概要とアーキテクチャ
- Redshift の2018年の主なアップデート
- Redshift の今後のアップデート (re:Invent2018での発表)

本日のアジェンダ

- Redshift の概要とアーキテクチャ
- Redshift の2018年の主なアップデート
- Redshift の今後のアップデート (re:Invent2018での発表)

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

Amazon Redshift の特長



高速

データウェアハウス・分析向けに特化したRDB

さまざまな種類の分析ワークロードについて、より迅速な洞察を取得可能



スケーラブル

ペタバイト級までスケールアウト

分析要求やデータ容量が増加しても、動的にスケールアップ/アウトが可能



データレイク
への拡張

データレイク上のデータへの直接アクセス

Amazon S3 上のオープンフォーマットデータをそのまま分析可能



高いコスト
効果

初期費用なし、小規模から始めて、利用に応じた支払いが可能

約\$1,000/TB/年、最小\$0.314/時*から

*費用は2019年1月時点での東京リージョンのもので

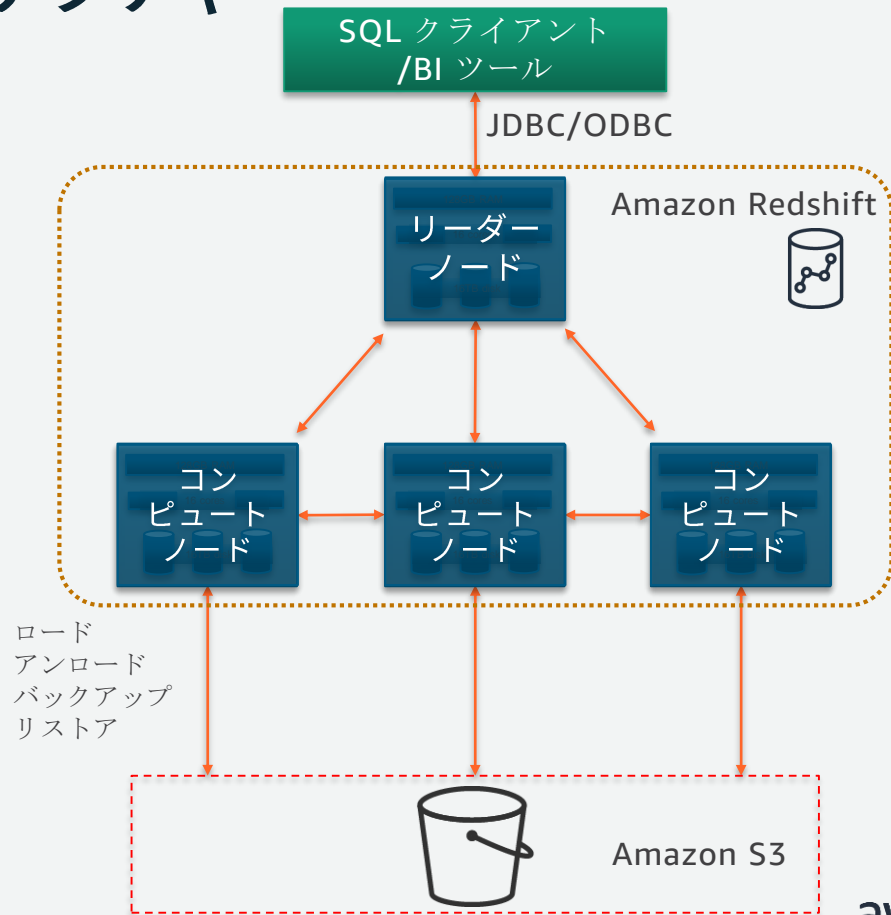
Amazon Redshift アーキテクチャ

リーダーノード

- SQL エンドポイント
- メタデータの格納
- 並列SQL処理コードの生成

コンピューターノード

- ローカル列指向ストレージ
- クエリの並列実行
- ロード/アンロード
- バックアップ/リストア



Amazon Redshift アーキテクチャ

リーダーノード

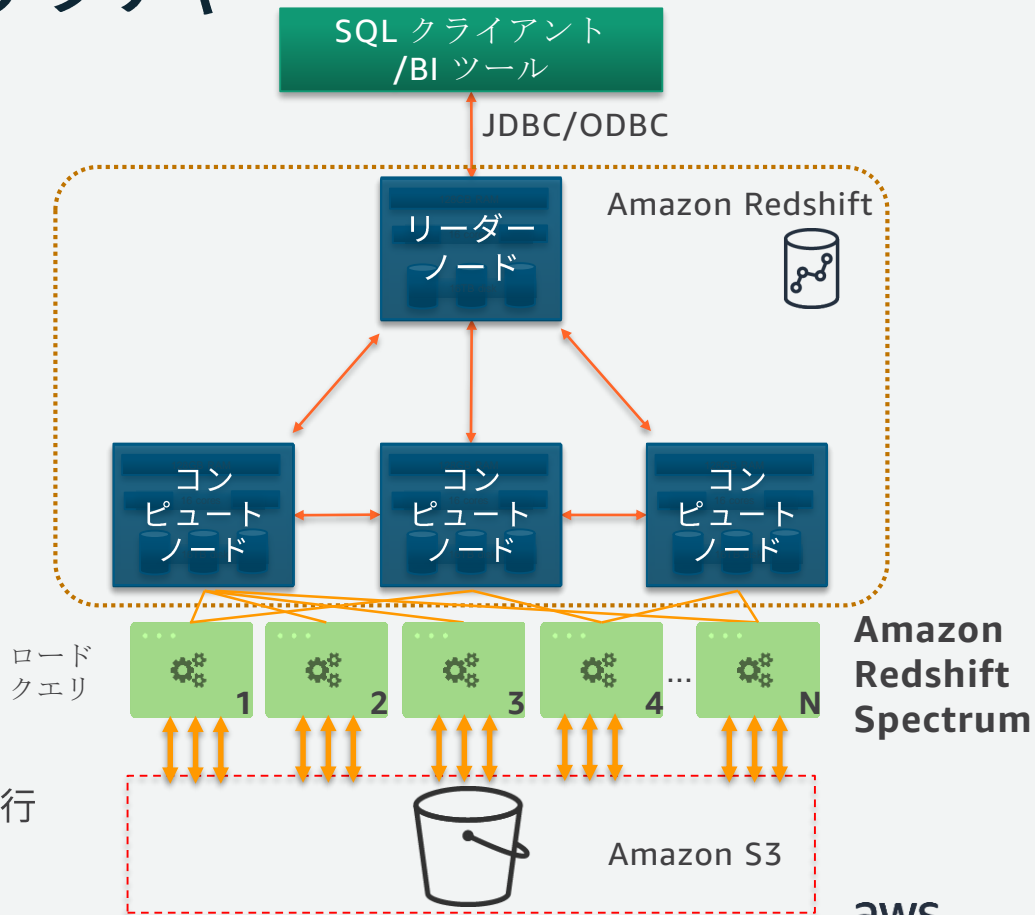
- SQL エンドポイント
- メタデータの格納
- 並列SQL処理コードの生成

コンピューターノード

- ローカル列指向ストレージ
- クエリの並列実行
- ロード/アンロード
- バックアップ/リストア

Amazon Redshift Spectrum

- Amazon S3 に対する直接クエリ実行



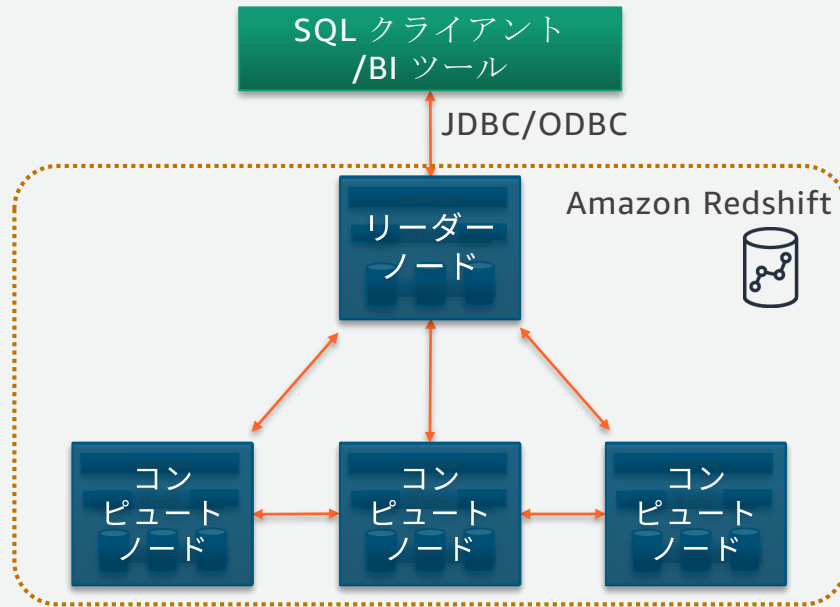
高速処理のための並列構成

シェアードキャッシング

- ディスクをノード間で共有しない構成
- Redshift のデータは各コンピュータノード内のローカルディスクに分散されて格納される

MPP : Massive Parallel Processing

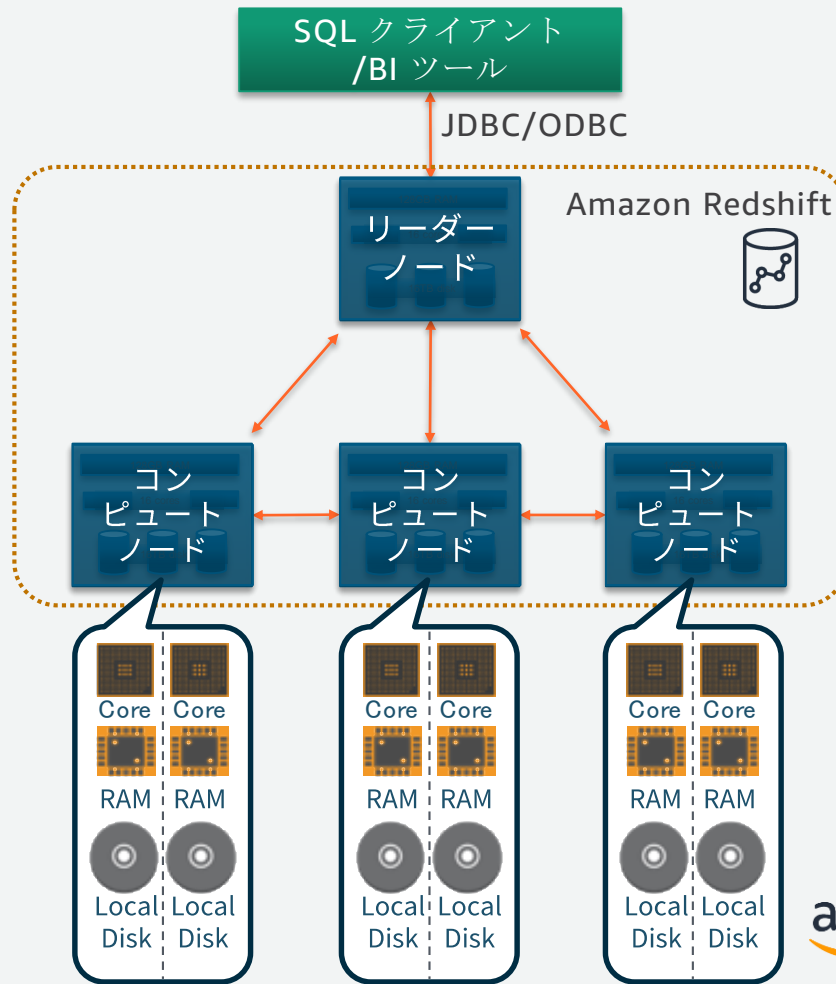
- 1つのタスクを複数のノードで分散して実行する仕組み
- Redshift ではリーダーノードがタスクをコンピュータノードに分散して実行
- 各コンピュータノードの処理結果はリーダーノードを経由してクライアントに返却



高速処理のための並列構成

スライス

- メモリとディスクをノード内で分割した論理的な処理単位
- 各コンピューターノードはインスタンスタイプに応じて2, 16のスライスを持つ
- 各スライスに割り当てられたリソースが、配下に分散されたデータに対して並列で処理を実行



Amazon Redshift インスタンスタイプ

- SSDベースのDC2とHDDベースのDS2から選択
 - データは圧縮格納されるため、ストレージ総量より多くのデータが格納可能
- クラスタータイプとノード数は後から変更可能

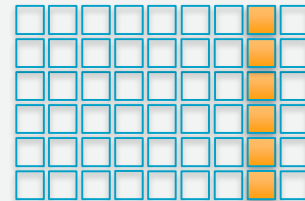
DC2 - Dense Compute						
	vCPU	メモリ(GB)	ストレージ	スライス	ノード数	価格(※)
dc2.large	2	15	0.16TB NVMe SSD	2	1~32	\$0.314 /1時間
dc2.8xlarge	32	244	2.56TB NVMe SSD	16	2~128	\$6.095 /1時間
DS2 - Dense Storage						
ds2.xlarge	4	31	2TB HDD	2	1~32	\$1.190 /1時間
ds2.8xlarge	36	244	16TB HDD	16	2~128	\$9.520 /1時間

※価格は東京リージョンにおいて2019年1月時点のものです

高速処理のための I/O 削減

列指向ストレージ

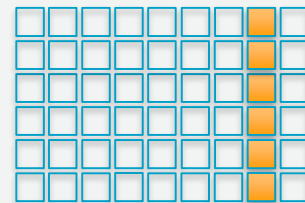
- データはコンピュータノードのローカルストレージ上のブロックに列ごとに格納。必要な列のみの読み取りが可能に
- ブロックサイズは1MB



高速処理のための I/O 削減

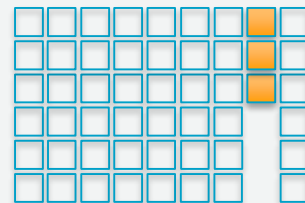
列指向ストレージ

- データはコンピュータノードのローカルストレージ上のブロックに列ごとに格納。必要な列のみの読み取りが可能に
- ブロックサイズは1MB



データ圧縮

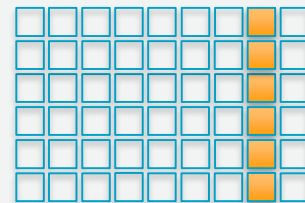
- データは列ごとに高度に圧縮される
- ブロック内に多くのデータが格納され、I/O 効率が向上



高速処理のための I/O 削減

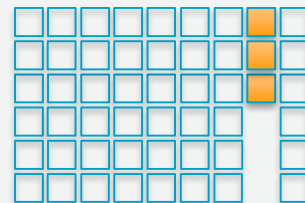
列指向ストレージ

- データはコンピュータノードのローカルストレージ上のブロックに列ごとに格納。必要な列のみの読み取りが可能に
- ブロックサイズは1MB



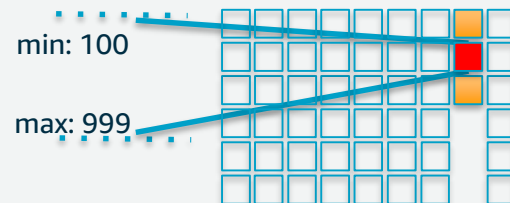
データ圧縮

- データは列ごとに高度に圧縮される
- ブロック内に多くのデータが格納され、I/O 効率が向上



ゾーンマップ

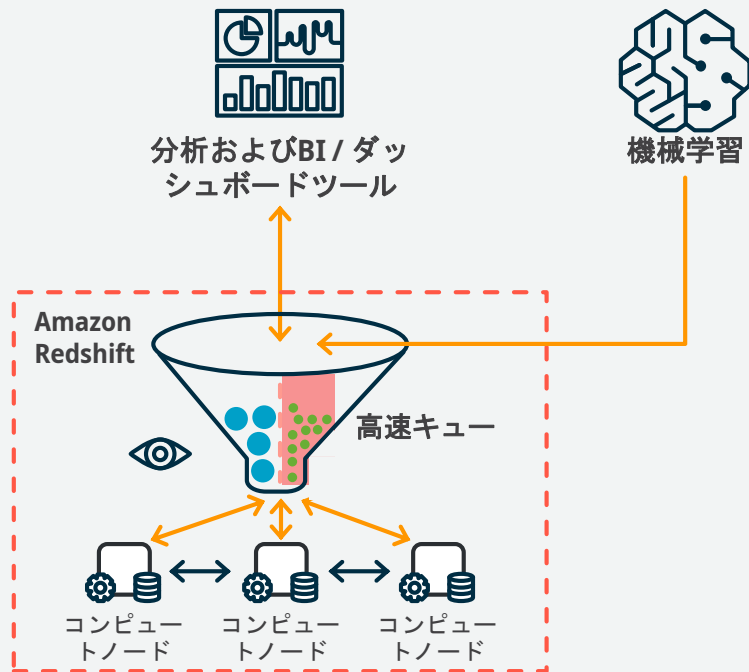
- データが格納されている各ブロック上に存在する最小値、最大値をリーダーノードのメモリ上に保持
- クエリの内容に応じて、処理に不必要なブロックは読み飛ばすよう効率的なアクセスを実現



高速処理のための機能①

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のスループットを高速化



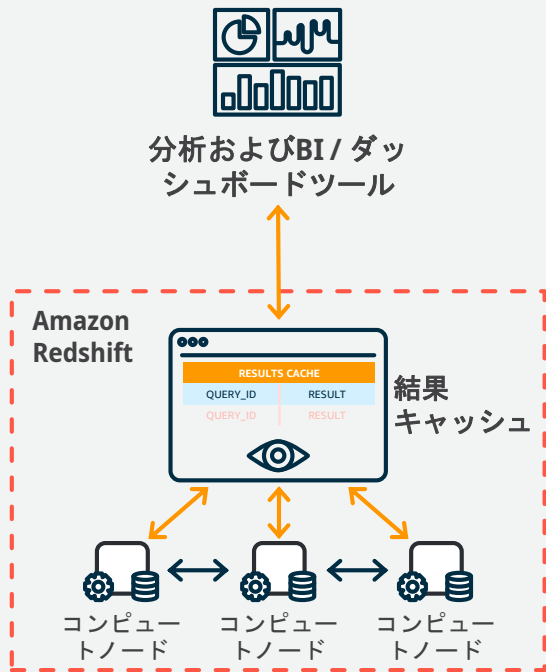
SQAの機能

- 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

高速処理のための機能②

結果セットのキャッシュ(リザルトキャッシュ)

繰り返し実行されるクエリを1秒未満で実行

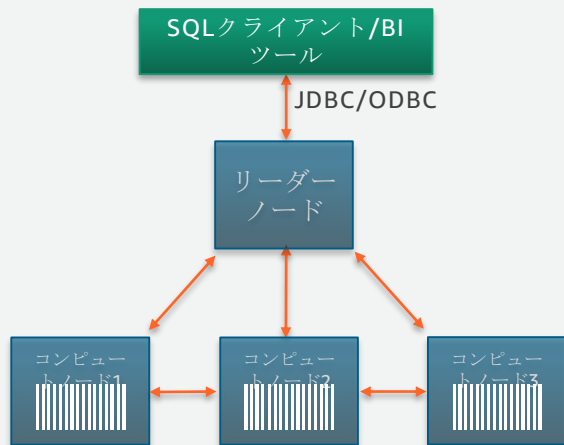


結果セットのキャッシュ機能

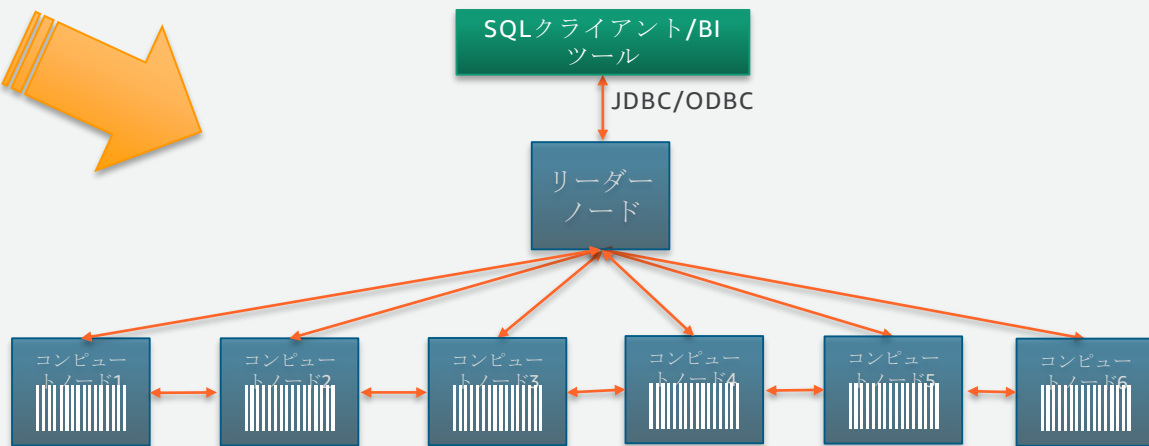
- 1 クエリはリーダーノードにて受付
- 2 リーダーノード内のキャッシュにクエリ結果が含まれている場合、コンピュータノード上での処理を伴わずに返される
- 3 クエリ結果がキャッシュに存在しない場合、コンピュータノード上でクエリが実行されて、その結果がキャッシングされる

スケールアウトによるパフォーマンス向上

クラスターリサイズ



- コンピューター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



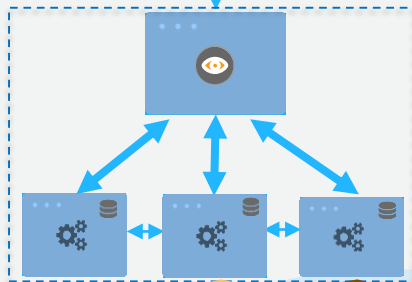
データウェアハウスを S3 データレイクへ拡張



```
クエリ  
SELECT COUNT(*)  
FROM S3.EXT_TABLE  
GROUP BY ...
```

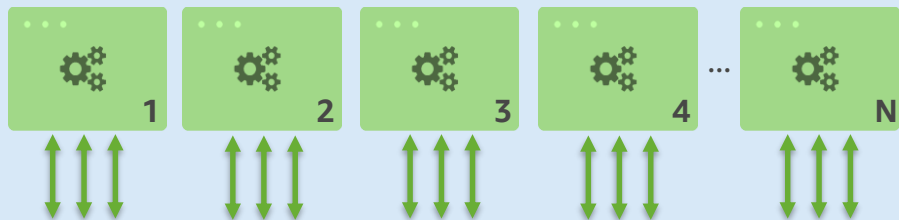
JDBC/ODBC

Amazon Redshift
データウェアハウス



Redshift Spectrum

サーバーレスコンピューティングのスケールアウト



Amazon S3

データレイク
エクサバイトスケールのオブジェクトストレージ

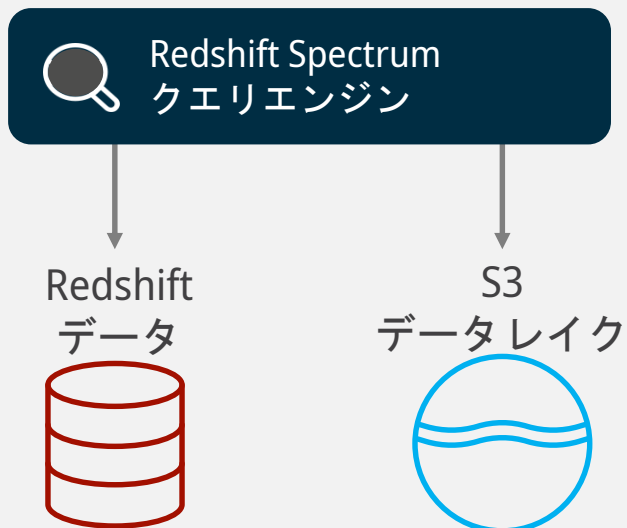


AWS Glue

データカタログ



Redshift Spectrum 特長



データロード処理が不要に

コンピュートとストレージを分けてスケーリング

RedshiftとS3それぞれに存在するデータを結合

安定したクエリパフォーマンスと高い並行処理

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)

スキャンしたデータ量のみの課金制

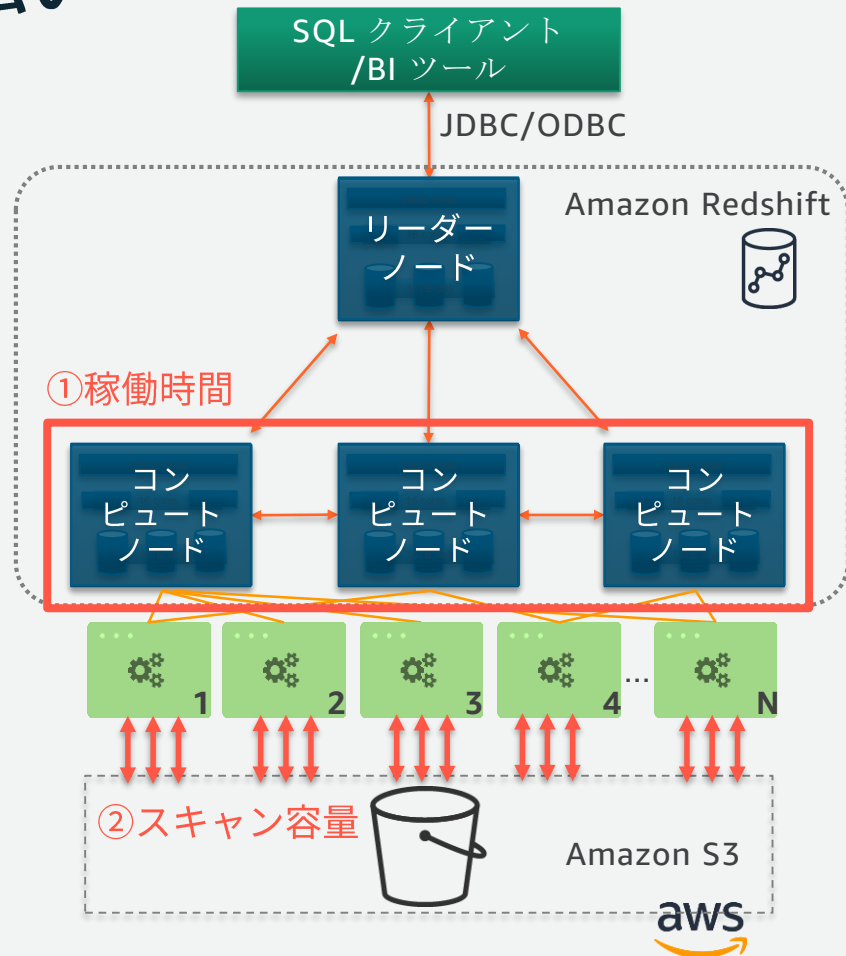
初期費用不要・利用分だけの支払い

Redshift のインスタンス料金

- 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
- 使っただけの課金
- リザーブドインスタンス（3年, 1年）にも対応

Amazon S3 データレイクへのクエリ料金

- S3 上のデータ 1TB スキャンあたり\$5



本日のアジェンダ

- Redshift の概要とアーキテクチャ
- Redshift の2018年の主なアップデート
- Redshift の今後のアップデート (re:Invent2018での発表)

Redshift の2018年の主なアップデート

継続的なイノベーション

Automatically enabled short query acceleration

Support for lateral column alias reference

New Quick Starts

New CloudWatch metrics

Customized Recommendations with Advisor

Current and trailing tracks for release update

Federated authentication with single sign-on

Improved performance for commits

COPY from Parquet and ORC file formats

Support for Parquet and ORC in Kinesis Data Firehose

Improved workload management console experience

Query Editor

Support for late-binding views

SQL Scalar user-defined functions

Integration with AWS Glue

Support for Nested Data with Spectrum

Spectrum support for DATE data type

Improved performance for UNION ALL queries


Free upgrade from DC1 to DC2 RIs

Query monitoring rules (QMR)

Support for Zstandard high compression encoding

Query processing improvements

220+

 features and enhancements released*

Support for Python UDF logging module

Enhanced VPC routing

Additional Spectrum regions

Support for Scalar JSON and Ion data types

Late materialization for faster query processing

Support for DATE data type with Spectrum

Short Query Acceleration

Utilization reports

Machine learning integration to accelerate dashboards and interactive analysis

Improved resource management for memory-intensive queries

Faster string manipulation

Automatically hopping queries without restarts

Support for uppercase column names

Result Caching for Repeat Queries

Support for LISTAGG DISTINCT

Support for ORC and Grok file formats

Integration with QuickSight

DMS support with Redshift

3.5x Improved Throughput

Improved performance for repeat queries

***Since re:Invent 2017**

Redshift の2018年の主なアップデート

継続的なイノベーション

Automatically enabled short query acceleration

Support for lateral column alias reference

New Quick Starts

New CloudWatch metrics

Customized Recommendations with Advisor

Current and trailing tracks for release update

Federated authentication with single sign-on

Improved performance for commits

COPY from Parquet and ORC file formats

Support for Parquet and ORC in Kinesis Data Firehose

Improved workload management console experience

Query Editor

Support for stateful views

Support for scalar user-defined functions

Support for Amazon S3 Glue

Support for Nested Data with Spectrum

Spectrum support for DATE data type

Improved performance for UNION ALL queries

Free upgrade from DC1 to DC2 RIs

Speed

Query monitoring rules (QMR)

Support for Zstandard high compression encoding

Query performance improvements

Query Editor

Simplicity

Support for Python UDF logging module

Enhanced VPC routing

Additional Spectrum regions

Support for Scalar JSON and Ion data types

Late materialization for faster query processing

Support for DATE data type with time zone

Short Query Acceleration

Scale

Machine learning integration to accelerate dashboards and interactive analysis

Improved resource management for memory-intensive queries

Faster string manipulation

Automatically hopping queries without restarts

Support for uppercase column names

Result Caching for Repeat Queries

Support for LISTAGG DISTINCT

Support for ORC and Grok file formats

Integration with QuickSight

DMS support with Redshift

3.5x Improved Throughput

Improved performance for repeat queries

***Since re:Invent 2017**

Compiled code cache

Support for lateral column alias reference

Query planning

Cluster resize operations

Short query acceleration

Improvements for the COPY operation when ingesting data from Parquet and ORC formats

Commit processing enhancements

Query rewrites that pushdown selective joins into a subquery

COPY operation when ingesting data from Parquet and ORC formats

Query processing improvements

Resource management for memory-intensive queries

Faster string manipulation

Result caching

Queries operating over CHAR and VARCHAR columns

2x the number of tables in a cluster

Queries that refer to stable functions with constant expressions

Performance improvement for queries that refer to stable functions over constant expressions

Late materialization

Queries with intermediate subquery results that can be distributed

Hash join memory utilization optimizations and cache line prefetching

Expressions on the partition columns of external tables

Joins involving large numbers of NULL values in a join key column

Single-row inserts

Complex EXCEPT subqueries

DC2 nodes

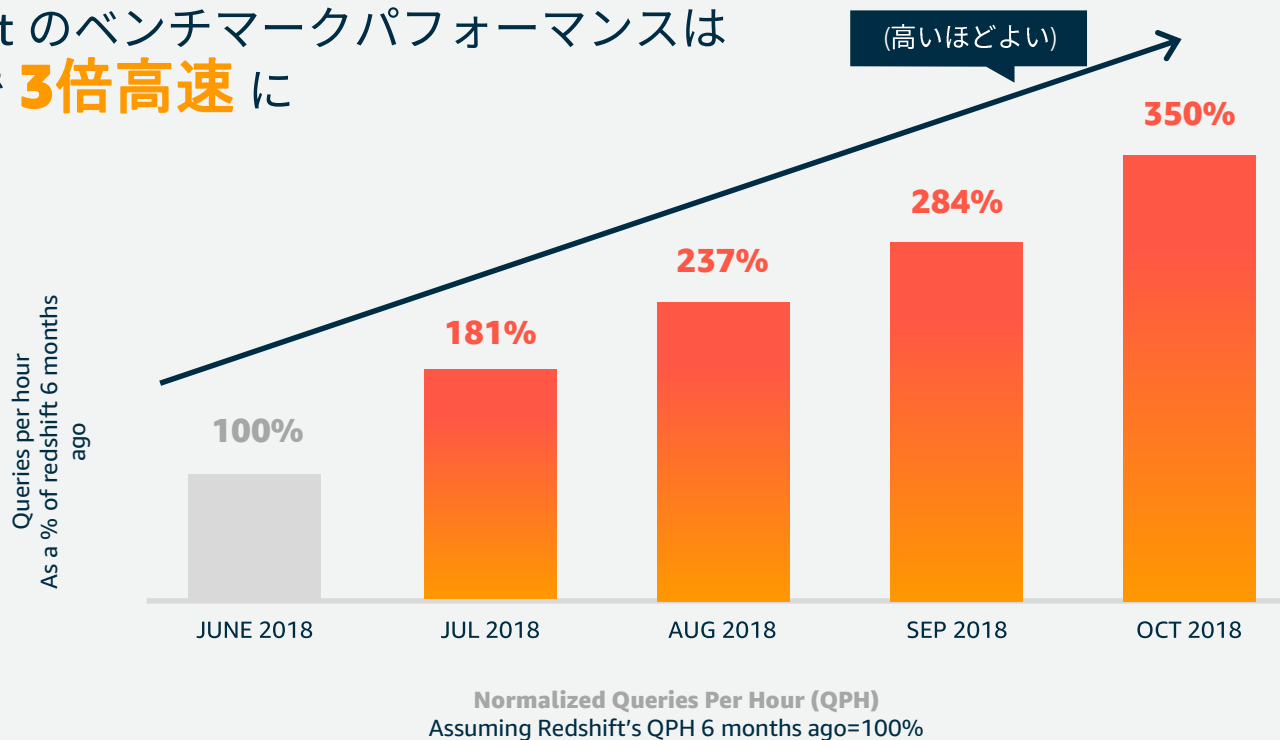
Improvements to speed

***Since re:Invent 2017**

クエリパフォーマンスの継続的向上

New!

Redshift のベンチマークパフォーマンスは
6ヶ月で **3倍高速** に



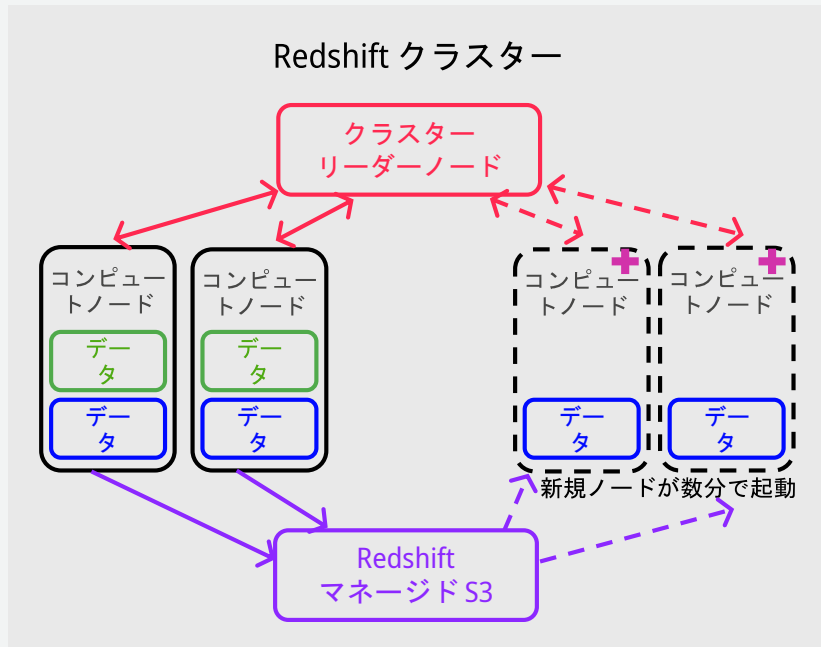
クエリパフォーマンスの継続的向上

- ハッシュジョインの高速化
 - ハッシュアルゴリズムの改善
 - メモリ使用率の大幅な改善
 - キャッシュラインプリフェッチによる結合パフォーマンスの改善
- TPC-HやTPC-DSなど、結合を多用するワークロードでは、さまざまなクエリでパフォーマンスが28%~100%向上
- ハッシュジョインや集約処理でメモリフットプリント全般の削減幅が64倍
スループット全般で大幅な改善

Elastic Resize

クラスターをオンデマンドで迅速に伸縮し、パフォーマンスを向上

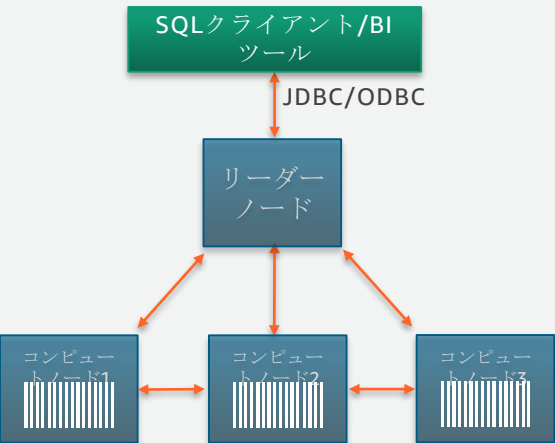
New!



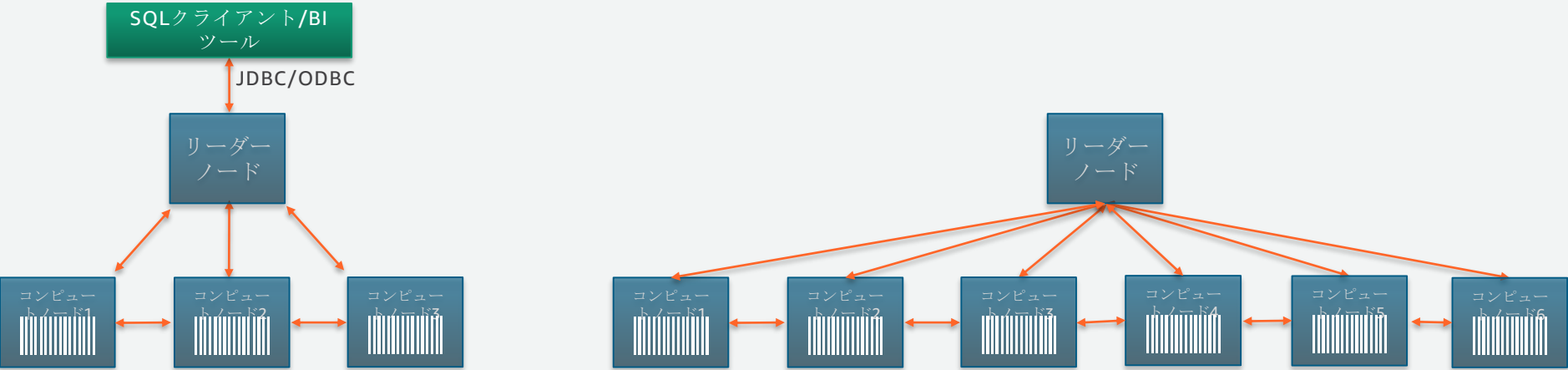
クラスターに追加するノードを数分で追加/削除する

既に Redshift が動作する全リージョンで利用可能

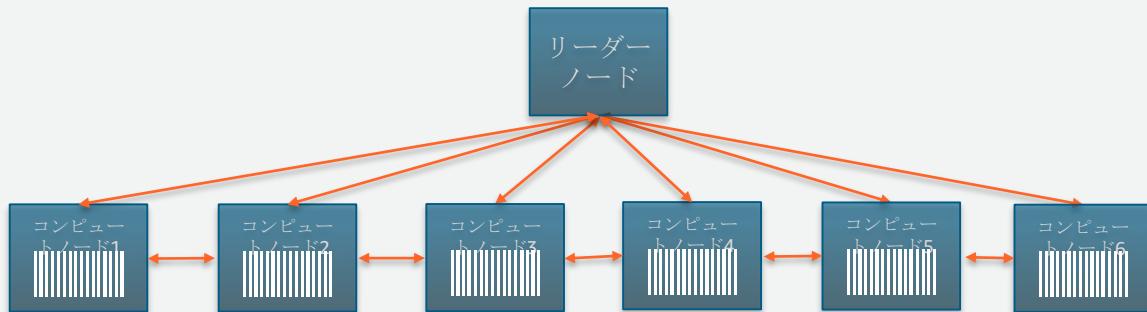
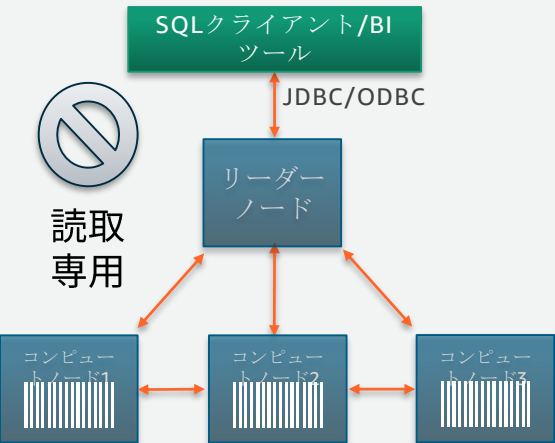
Classic Resize (従来のリサイズ)



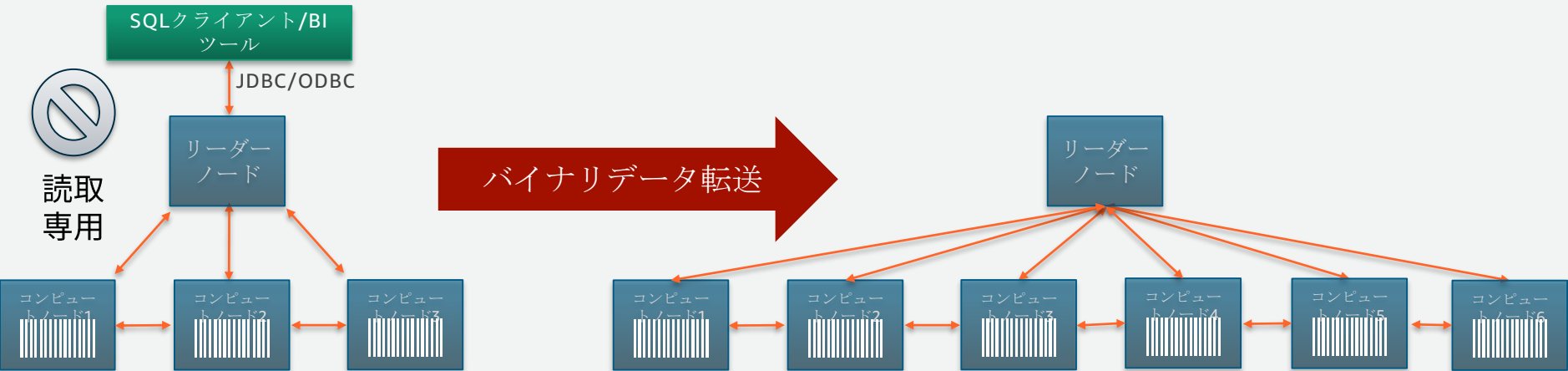
Classic Resize (従来のリサイズ)



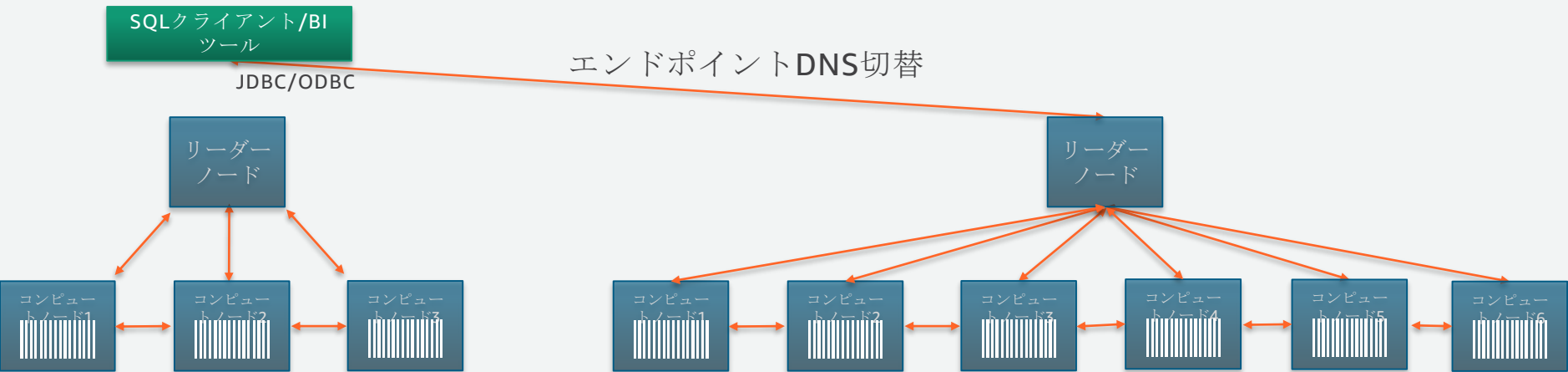
Classic Resize (従来のリサイズ)



Classic Resize (従来のリサイズ)



Classic Resize (従来のリサイズ)



- データはクラスター間を移動して再分散される（それなりに時間が必要）
- リサイズ中はクラスターは読み取り専用モードになる

Elastic Resize のメリットとユースケース



Elastic Resizeにより、数分でノードを増減できる

ユースケース：

- ✓ ノードを一時的に増やし、クエリ処理をより高速化
- ✓ ノードを一時的に増やし、ETLをより高速に完了
- ✓ ノードを一時的に減らし、オフピーク時間のコストを抑制

コンソールまたはCLIより実行可能

Resize Cluster

To add or subtract nodes within minutes, choose Elastic resize. To change node type or if Elastic resize isn't available for your configuration, choose Classic resize.

Type of resize Elastic resize Classic resize

Node type ⓘ

Current number of nodes 2

New number of nodes* ⓘ

Warning: Your cluster will be unavailable for a few minutes and some connections may be terminated. [Learn more](#).

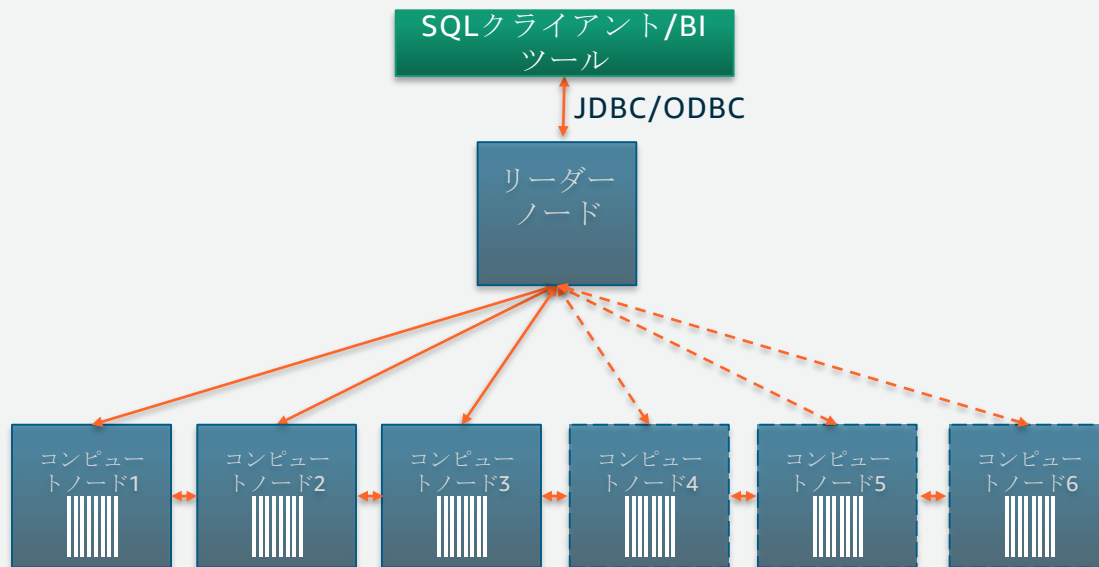
resize-cluster

resize-cluster

```
--cluster-identifier <value>
[--cluster-type <value>]
[--node-type <value>]
--number-of-nodes <value>
[--classic | --no-classic]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

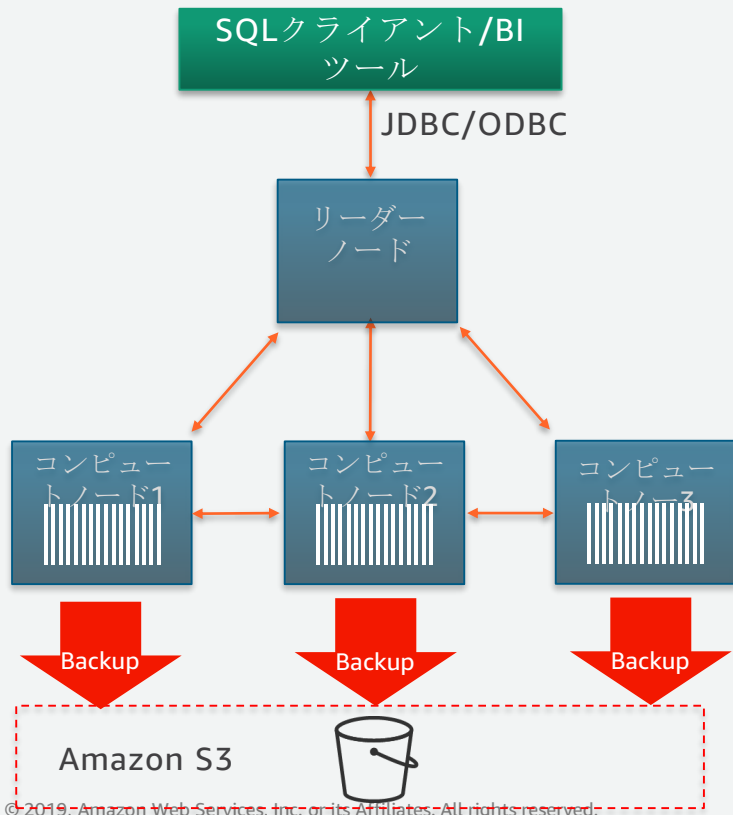
Elastic Resize Deep Dive

3ノードから6ノードへリサイズする例



Elastic Resize の内部処理 : 初期フェーズ

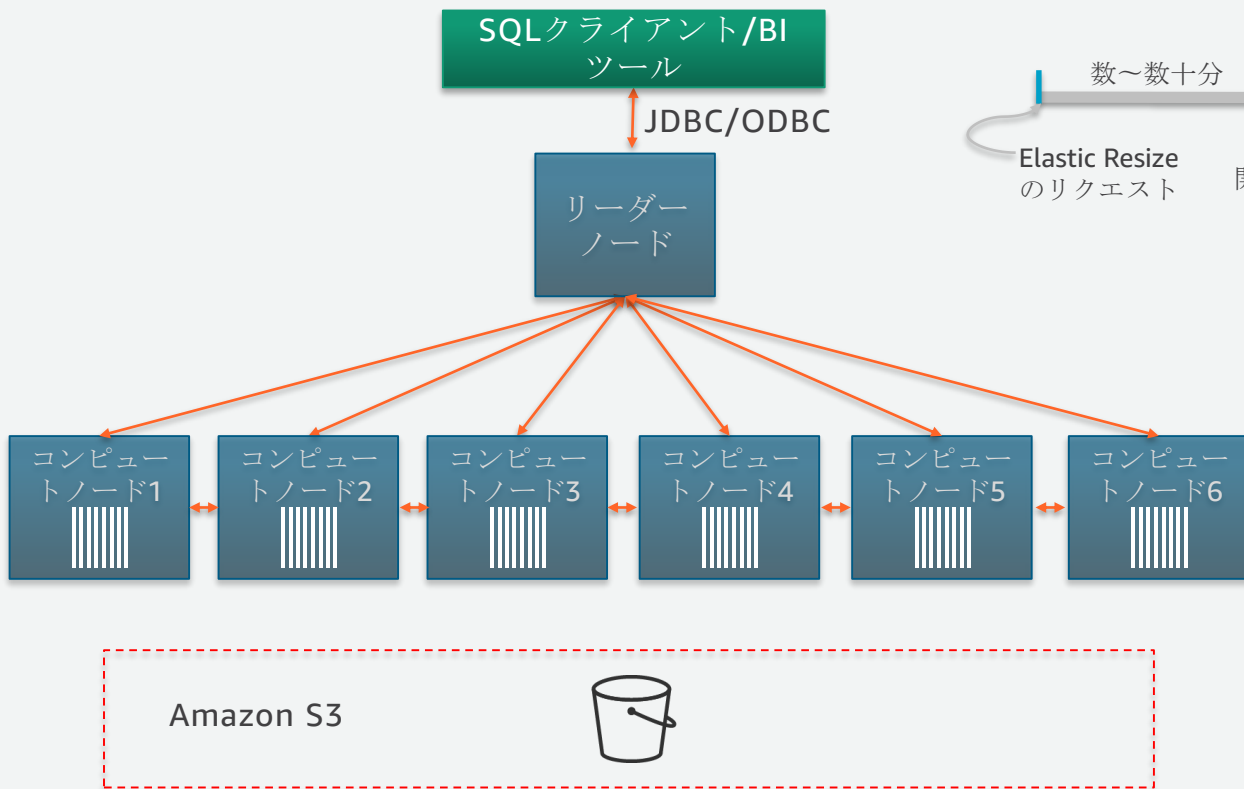
S3へ最新スナップショットを取得



- Elastic Resizeが開始されると、Amazon S3へ自動的にスナップショットが取得される
- その間のクラスターに対する読み書きは可能

Elastic Resize の内部処理 : ノード再構成フェーズ

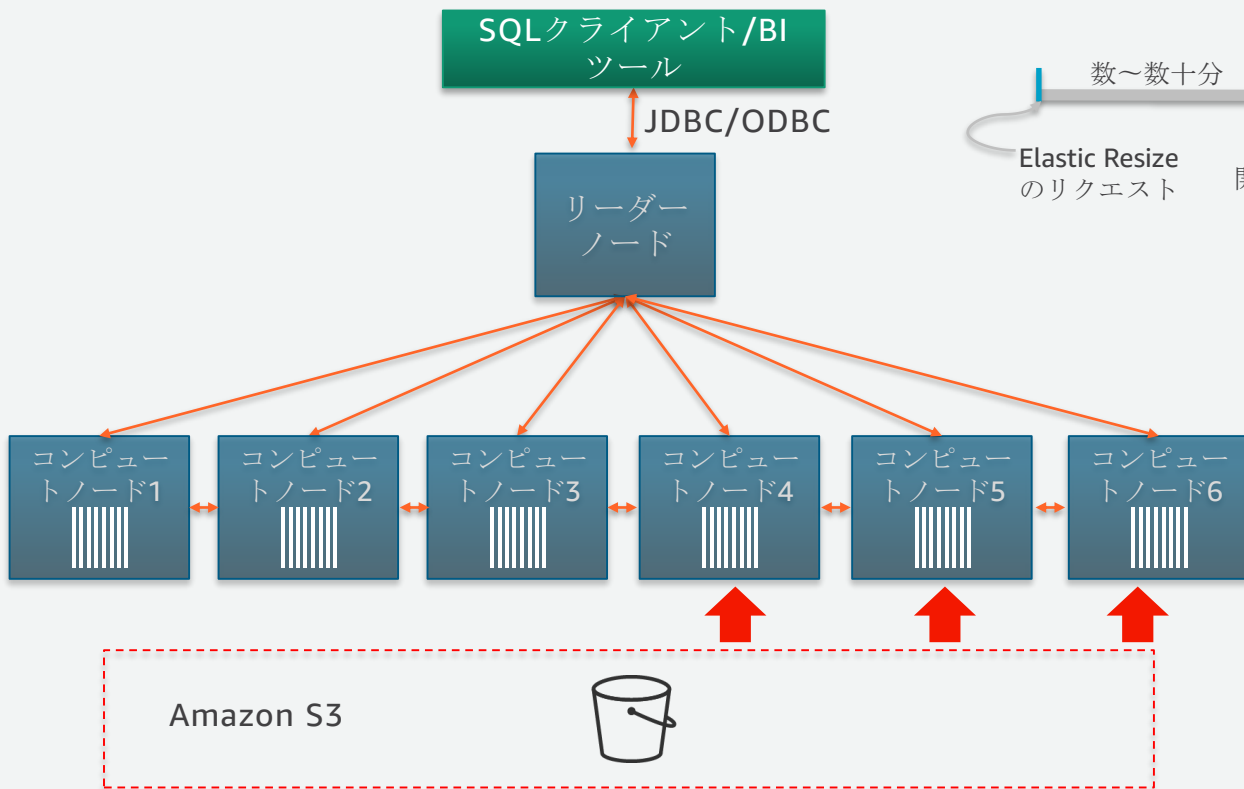
数分のうちに完了



- リサイズ中は、既存クラスターに新しいノードが追加、または削除される
- スライスはシャッフル
- すべてのクエリセッションはリーダーノードでキューイング
- いくつかのクエリは中断される可能性がある

Elastic Resize の内部処理：データ転送フェーズ

ワークロードに応じて優先度付けしながら処理



- このフェーズでクラスターは使用可能になり、データはS3からバックグラウンドで転送
- クエリに必要なホットブロックから先に戻す

Elastic Resize によるノード増減範囲

インスタンスタイプ	許容される増減範囲	例：4ノードクラスターの場合に指定可能なノード数
dc2.large ds2.xlarge	元のサイズから1/2倍または2倍のサイズのみ	2, 4, 8
dc2.8xlarge ds2.8xlarge	元のサイズの1/2倍～2倍の間の任意のサイズ	2, 3, 4, 5, 6, 7, 8

Elastic Resize と Classic Resize の比較

	Elastic resize	Classic resize
ワークロードスパイク時に即時にスケールアップ/ダウンさせたい	✓	
徐々にストレージを増やしたいまたは減らしたい	✓	
クラスターのインスタンスタイプを変更したい (SSD <-> HDD)		✓
Elastic Resize が対応していないノード増減範囲の場合		✓
リサイズ中に使用が制限される時間	数分間 (ただしクエリのコネクションは維持される)	数十分～数時間 (読み取り専用モード)

Automatic vacuum delete

Cluster resize

CloudWatch metrics
for Query
Throughput by WLM
Queues

Current and trailing
tracks for release
updates

Lateral column alias reference

Stream real-time data in
Parquet or ORC formats
using Kinesis Data
Firehose

CloudWatch metrics
for Query Duration
by WLM Queues

Free upgrade from for DC1 RIs to DC2

Enhancements to
VACUUM DELETE

CloudWatch metrics
for Workload Execution
Breakdown

Efficiency of backup performance

Redshift Advisor for best practice recommendations

Improvements to simplicity

Query Editor

CloudWatch metrics for Query
Throughput, Query Duration

Short query acceleration is self-optimizing

Manage
components of a
multi-part query in
the AWS console

Cluster resize operations

Query Monitoring Rules (QMR)
now support 3x more rules

CloudWatch
Query Runtime Breakdown metric

DISTSTYLE AUTO distribution style

***Since re:Invent 2017**

Vacuum と Analyze がバックグラウンドで自動実行

New!

Vacuum Delete

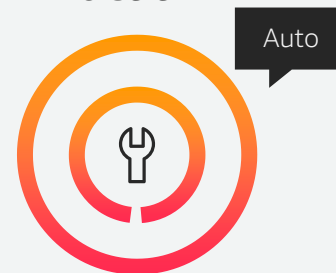
- Vacuum は DELETE や UPDATE 後の削除領域の開放とデータソートを行う処理で、このうちの前者が自動実行されるようになった（ソートは別途手動で実行する必要あり）
- クエリ負荷とテーブル内の削除済み行数に基づいてスケジュール実行される
- Vacuum 処理自体も、テーブル全体に実行されるのではなく列ごとに実行されるようになっており、処理に必要なリソースの量が劇的に減少している
- 自動実行をユーザーにて停止することは不可

Analyze

- 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
- クエリ負荷に基づいてスケジュール実行される
- パラメータ `auto_analyze` を変更することで、自動実行をOFFにすることも可能



Vacuum



Analyze

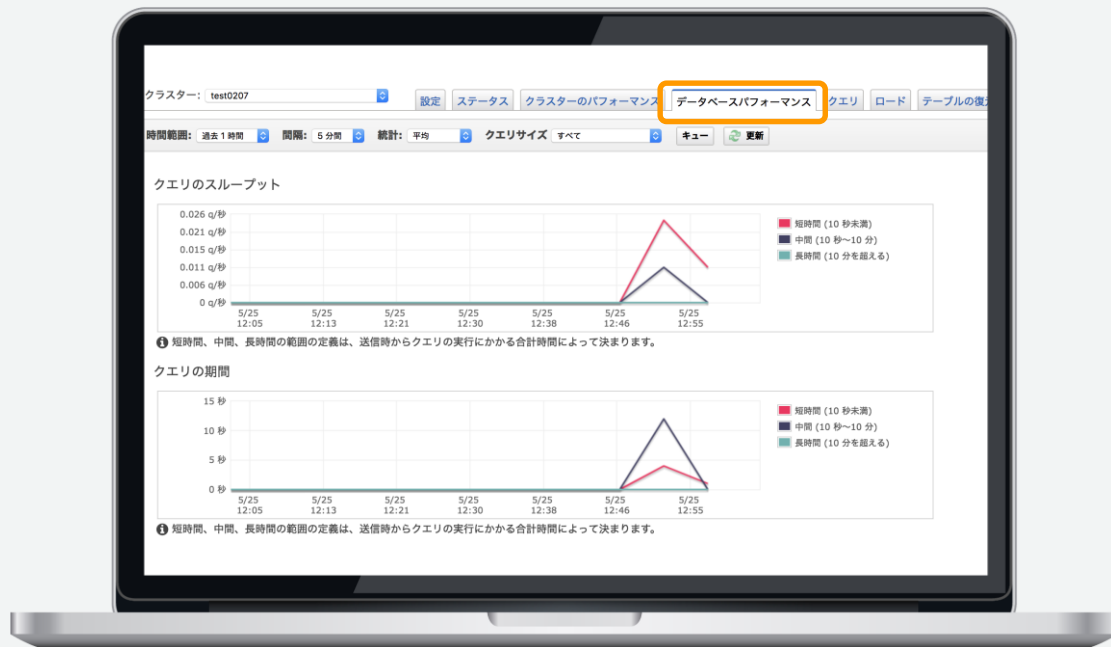
マネジメントコンソールから見る Redshift のアップデート



マネジメントコンソールから見る Redshift のアップデート 「クラスター」



クラスターモニタリングに関するアップデート



データベースパフォーマンス タブの追加

マネージメントコンソールに
データベースパフォーマンス
タブが追加

ワークロードの実行内訳、ク
エリのスループットとレイテ
ンシーの状況をグラフィカル
に確認可能

Amazon CloudWatch と連携
し、通知アラームが出される
ように設定も可能

クラスター全体としてのパ
フォーマンス状況をチェック
する際に便利

クラスターメンテナンスに関するアップデート(1/2)



メンテナンスウィンドウの延期

最大14日間のメンテナンスウィンドウの延期が可能に

ただし、必須のハードウェア交換がクラスターでスケジュールされている場合は、メンテナンス期間の延期は無効に

次のメンテナンスウィンドウまでの間にこのような保留中のイベントがスケジュールされている場合は事前通知が送信される (Amazon SNS 経由でサブスクライブ)

クラスターメンテナンスに関するアップデート(2/2)



メンテナンストラックの設定

デフォルトは”Current”トラックで、最新バージョンがメンテナンスウィンドウで適用される従来通りの挙動

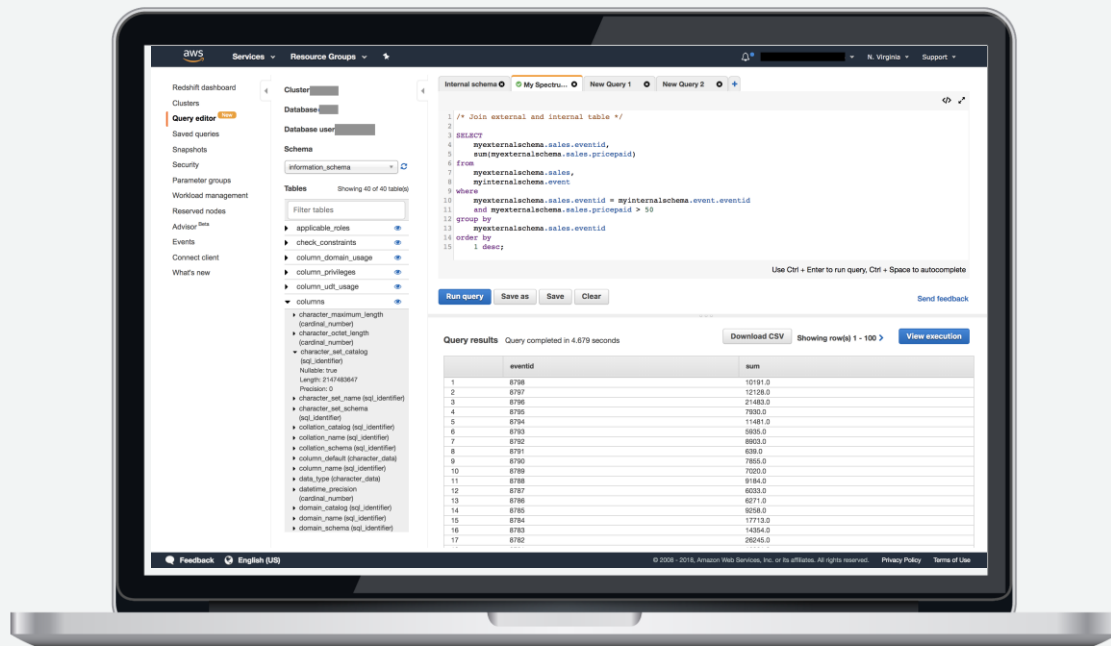
”Trailing”トラックに設定することで、アップデートに対し一定の猶予期間を設定可能に

クラスターは、最新の一つ前のバージョンで起動し、次期バージョンが出るまでは更新されない

マネジメントコンソールから見る Redshift のアップデート 「Query editor & Saved queries」



Redshift クエリエディタ & 保存クエリ



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に (専用クライアント不要)

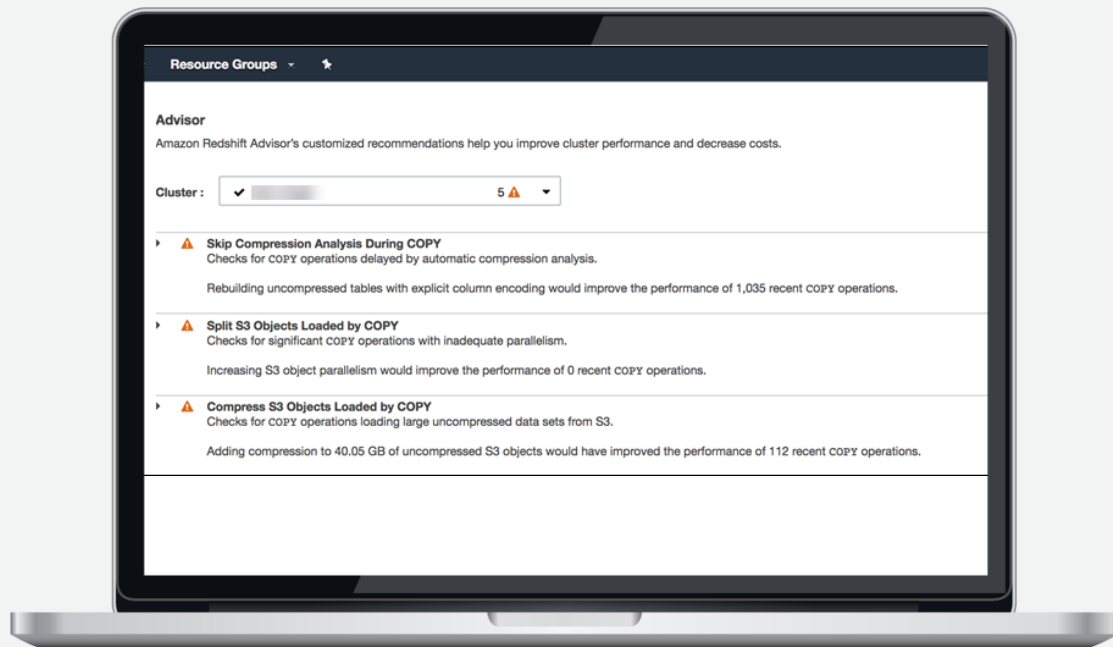
Saved queries 機能を使用すると、実行頻度の高いクエリを保存し、将来必要なときに簡単に使用可能

対応インスタンスタイプ：
dc1.8xlarge, dc2.large,
dc2.8xlarge, or ds2.8xlarge
(注意：ds2.xlargeは非対応)

マネジメントコンソールから見る Redshift のアップデート 「Advisor」



Redshift アドバイザ



アドバイザ機能

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

1. テーブルデータが適切に圧縮されているか
2. COPYコマンドにてロードされるS3オブジェクトが適切に圧縮されているか
3. 複数のアクティブデータベースをクラスター隔離できるか
4. WLMのメモリ割り当ては適切か
5. COPY実行中に不要な圧縮分析がされていないか
6. COPYコマンドにてロードされるS3オブジェクトが適切に分割されているか
7. テーブル統計情報が正確か
8. SQA は有効活用されているか
9. Interleaved Sort Key に単一の列を与えていないか

DATE data type

Support for Parquet, ORC, Avro, CSV, and other open file formats

Renaming external table columns

Push the LENGTH() string function to Spectrum

Retrieving metadata for late-binding views
Support for Enhanced VPC Routing

Query external tables during a resize operation

New Spectrum regions

Improvements to scale

Integrate seamlessly with your data lake

Specify the root of an S3 bucket as the source for an existing table

Arrays of arrays and arrays of maps

Spectrum support for JSON and ION

Table property to specify the file compression type for external tables

IN-list predicate processing in Spectrum scans

Spectrum support for nested data

ALTER TABLE ADD/DROP COLUMN for external tables is now supported via standard JDBC calls

Spectrum queries with aggregations on partition columns

Map datatypes in Spectrum to contain arrays

***Since re:Invent 2017**

「拡張された VPC のルーティング」と Spectrum

- 「拡張された VPC のルーティング」とは、VPC 内の Redshift クラスターと S3 間の通信を、VPC 内に閉じた接続とする（閉域網に閉じる）機能
- Redshift クラスター作成時もしくは作成後の設定変更により、有効化可能
- これまでは、Redshift クラスターで「拡張された VPC のルーティング」を有効化すると、Spectrum が利用できない制限があった
- 現在は上記制限はなくなっているため、「**拡張された VPC のルーティング**」を有効化しているお客様でも**Spectrum**が活用できる

https://docs.aws.amazon.com/ja_jp/redshift/latest/mgmt/spectrum-enhanced-vpc.html

Parquet / ORC ファイルフォーマットから COPY 実行可能に

ORC や Parquet ファイルのRedshiftへのデータロードをサポート

```
COPY table FROM 's3 prefix' FORMAT AS ORC | PARQUET ;
```



Spectrum ネスト化されたデータのサポート

- ネスト化された半構造化データを、Redshift Spectrumの外部表として指定することが可能に
- オープンファイルフォーマットをサポート: Parquet, ORC, JSON, Ion

```
{ Id: 1,  
  Name:  {Given:"John", Family:"Smith"},  
  Phones: ["123-457789"],  
  Orders: [ {Shipdate: "Jul 12,2018 11:59:59",  
            Price: 100.50}  
            {Shipdate: "Jul 13,2018 09:10:00",  
            Price: 99.12} ]  
}  
{ Id: 2,  
  Name:  {Given:"Jenny", Family:"Doe"},  
  Phones: ["858-8675309", "415-9876543"],  
  Orders: [ ]  
}  
{ Id: 3,  
  Name: {Given:"Andy", Family:"Jones"},  
  Phones: [ ]  
  Orders: [ {Shipdate: "Jul 12,2018 08:02:15",  
            Price: 13.50} ]  
}
```

Spectrum ネスト化されたデータのサポート

- ネスト化された半構造化データを、Redshift Spectrumの外部表として指定することが可能に
- オープンファイルフォーマットをサポート: Parquet, ORC, JSON, Ion
- サポートするComplex Data Type : struct, array, map
- 既存のSQLを拡張し、ネスト構造をドット表記で表現
- CTASを用いて、ネスト化されたデータのETL (Redshift Localテーブルへロード)が容易に

```
create external table
datalake.nested_customers_orc(
  id int,
  name struct<given:varchar(20),
              family:varchar(20)>,
  phones array<varchar(20)>,
  orders array<struct
              <shipdate:timestamp,
              price:double precision>>
)
STORED AS ORC
LOCATION 's3://mybucket/nested_orc/';
```

```
SELECT c.id, c.name.given, c.name.family,
o.shipdate, o.price
FROM datalake.nested_customers_orc c
LEFT JOIN c.orders o
ON true;
```

クエリの例

```
SELECT
  c.id, c.name.given, c.name.family, o.shipdate, o.price
FROM
  datalake.nested_customers_orc c
LEFT JOIN
  c.orders o
ON true;
```

① Array型の列であるordersをFROM句のテーブル名のエイリアス"c"で修飾し、1つの別テーブルのように表現

② Orders列に紐づく配列要素は、さらにドットで区切って表現

id	given	family	shipdate	price
1	John	Smith	2018-07-12 08:02:15	100.5
1	John	Smith	2018-07-13 09:10:00	99.12
2	Jenny	Doe		
3	Andy	Jones	2018-07-12 08:02:15	13.5

(4 rows)

```
create external table
datalake.nested_customers_orc(
  id int,
  name struct<given:varchar(20),
             family:varchar(20)>,
  phones array<varchar(20)>,
  orders array<struct
             <shipdate:timestamp,
             price:double precision>>
)
:
```

本日のアジェンダ

- Redshift の概要とアーキテクチャ
- Redshift の2018年の主なアップデート
- Redshift の今後のアップデート (re:Invent2018での発表)

Redshift のこれまでの課題

OLTP系クエリと異なり、分析系のSQLは1クエリあたりに必要なリソースが大きくなるため、必然的に同時実行可能なクエリ数は下がる傾向にある
そのため、同時実行性の高い要件がある場合、これまでは...

- 同時実行数の多い定型分析などは RDS/Aurora にマートとして切り出し
- もしくは BI ツール側に定期的にキャッシュ
- Redshift クラスタを複数立てる (Spectrum の活用)

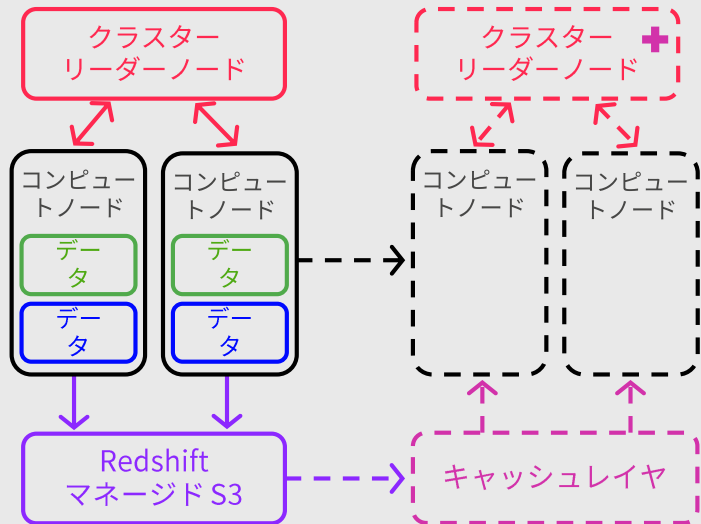
という選択肢を取ることで、要件をカバーしてきた

Concurrency Scaling (プレビュー)

従来よりはるかに高い同時実行性で一貫して高速なパフォーマンス

New!

Redshift クラスター 追加 Redshift クラスター



オンデマンドで追加のクラスターを自動的に起動

従来よりはるかに多くのクエリを同時に処理

エンドポイントはメインクラスターのまま

すべてのクラスターでデータの整合性は保たれる

Concurrency Scaling (プレビュー)

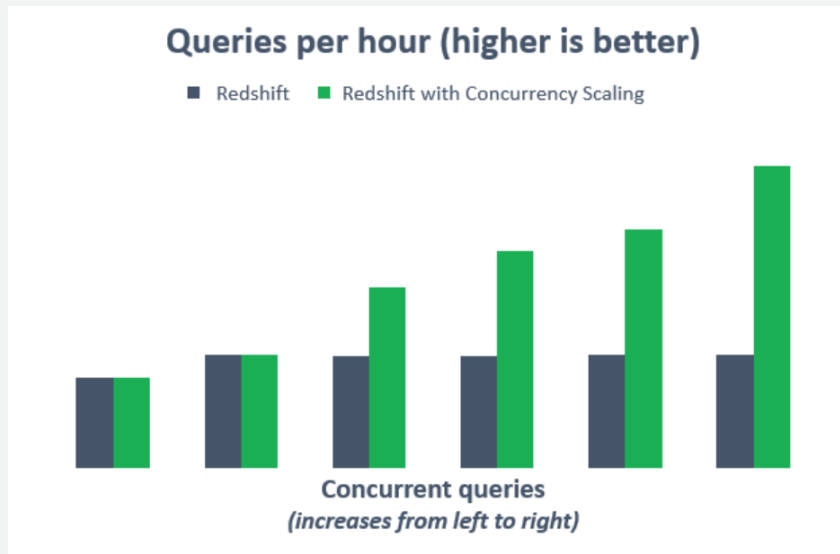
従来よりはるかに高い同時実行性で一貫して高速なパフォーマンス

New!

メインクラスターが使用されている24時間毎に、Concurrency Scalingのために1時間分のクレジットを提供

追加クラスターで稼働したクエリに対して1秒単位の課金

そのため多くのお客様は、Concurrency Scaling を無料をご利用可能



Amazon Redshift インテリジェントなメンテナンス

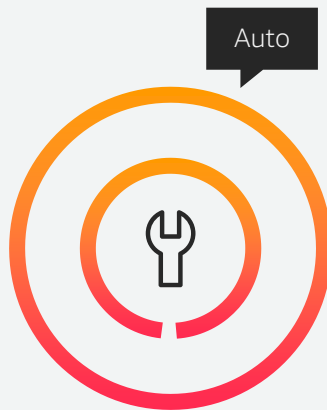


Vacuum や Analyze のようなメンテナンスプロセスはバックグラウンドで自動実行されるようになる

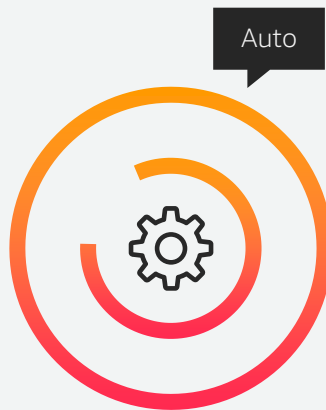
ゼロメンテナンスを目指して

Redshift は最適化されたスループットを提供するため、WLM concurrency setting の自動調整機能を提供予定

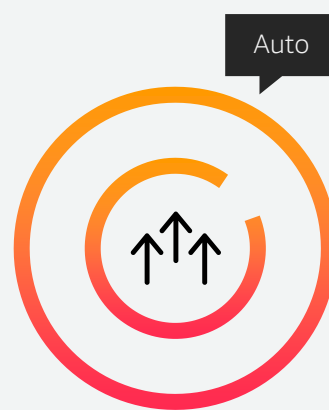
Coming Soon!



Analyze



WLM
Concurrency
Setting



Vacuum



Amazon Redshift インテリジェントな管理



パフォーマンス向上と
ディスクスペース最適化
のためテーブルのデータ
分散を自動化します

もう distkey で悩む
必要はありません!

Coming Soon!

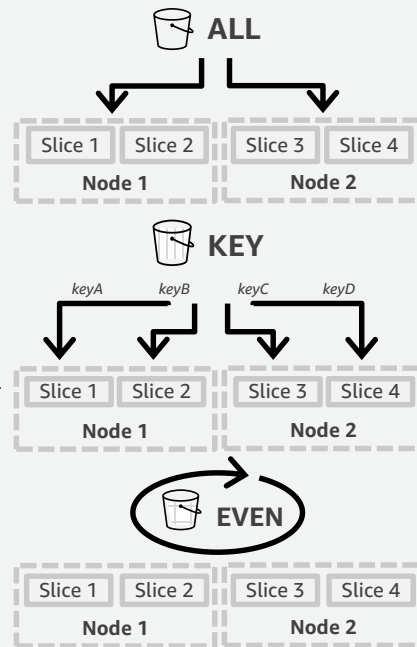


**Amazon
Redshift**

recommended
distribution key

Advise

継続的なワークロード分析に
基づき、インテリジェントな
チューニングリコメンデー
ションを提供します



ストアードプロシージャ in Redshift

Coming Soon!



既存のストアードプロシージャを使ってRedshift上で動かすことができるようになります

Redshift への移行が
より容易に！

Redshift は PL/pgSQL フォーマットのストアードプロシージャをサポートする予定です。これにより、既存のストアードプロシージャをそのまま持ち込むことができますようになります

ストアードプロシージャのサポートは、ETL、データバリデーション、カスタムビジネスロジックを効率的に実行するためのコードを実行可能にします



Amazon Redshift Spectrum

データウェアハウスを S3 データレイク上のエクサバイトのデータへ拡張

データロード処理が不要に

コンピューとストレージを分けてスケーリング

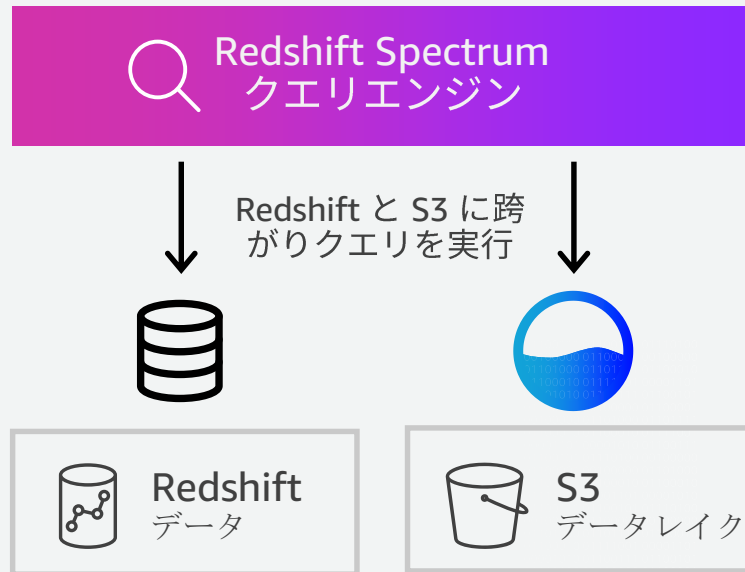
S3 上のデータに直接クエリを実行

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)

→ Unload to Parquet

→ Spectrum Request Accelerator

Coming
Soon!



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
- マネージドサービスとして、既存の機能拡張、新しい機能のリリースを定期的に行い、常に進化を続けている
 - 以前に存在した仕様や制限は過去のものになっている可能性がある
 - 以前に存在しなかった追加機能で、これまでできなかったことができるようになっていく可能性がある

参考資料

Amazon Redshift ホームページ

- <https://aws.amazon.com/jp/redshift/>

Amazon Redshift ドキュメント

- <https://aws.amazon.com/jp/documentation/redshift/>

Amazon Redshift フォーラム (Q&Aや新機能の告知) ※要AWSアカウント

- <https://forums.aws.amazon.com/forum.jspa?forumID=155>

Amazon Redshift Release Notes (新機能、修正等)

- <https://aws.amazon.com/releasenotes/Amazon-Redshift>

Amazon Redshiftのパフォーマンスチューニングテクニック Top 10

- <http://aws.typepad.com/sajp/2015/12/top-10-performance-tuning-techniques-for-amazon-redshift.html>

Amazon Redshift Spectrum 10 のベストプラクティス

- <https://aws.amazon.com/jp/blogs/news/10-best-practices-for-amazon-redshift-spectrum/>

AWS Bigdata Blog

- <https://aws.amazon.com/jp/blogs/big-data/>

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

資料公開と併せて、後日掲載します。

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>

