



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 Aurora MySQL

Archived

Solutions Architect 星野 豊

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



過去資料
<https://amzn.to/JPArchive>



自己紹介

星野 豊 (ほしの ゆたか)

- Aurora/RDS (MySQL) Specialist SA



経歴

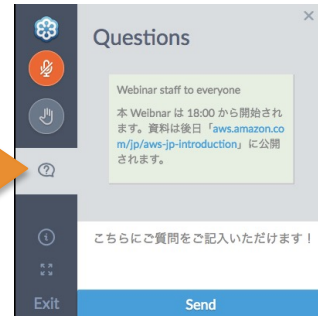
- 全てオンプレ環境のインフラエンジニア
- 全てAWS環境のインフラエンジニア



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

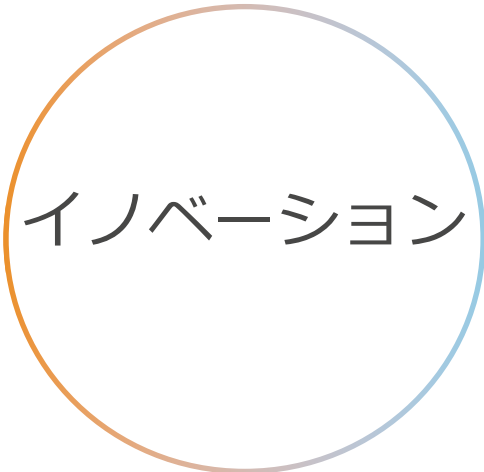


Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2019年4月24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.

Amazon Relational Database Service (Amazon RDS)



Amazon RDS

様々なオープンソースDBとコマーシャルDBの選択肢

クラウドネイティブ



オープンソース



コマーシャル



RDS プラットフォーム

自動フェイルオーバー
バックアップ・リカバリ
クロスリージョン・レプリケーション

セキュリティ
業界標準の認証・認可
自動的なパッチ適用

拡張モニタリング
定期的なメンテナンス
ボタン一つでスケール

Amazon Aurora...

オープンソースデータベースのコスト効率でエンタープライズデータベースを利用可能

データベースを**マネージド** サービスで



Amazon Aurora

コマーシャルデータベースの**性能** と **可用性**

オープンソースデータベースの**シンプルさ**と**コスト効果**

MySQL, PostgreSQLと**互換性**がある

利用した分だけお支払い

Amazon Aurora innovations

Re-imagining databases for the cloud

- 1 スケールアウト, 分散, マルチテナントデザイン
- 2 AWSサービスを活用したサービスオリエンテッドアーキテクチャ
- 3 自動化されたタスク - 完全マネージド・サービス

Leveraging AWS services



**Lambda
function**

AWS Lambdaイベントを
stored procedures/triggers
から実行



**AWS Identity
and Access
Management**

AWS Identity and Access
Management (IAM) rolesを利用
してデータベースユーザの認証



**Amazon
S3**

Amazon Simple Storage
Service (Amazon S3)デー
タをロード, S3を利用した
バックアップデータの保存、
リストア



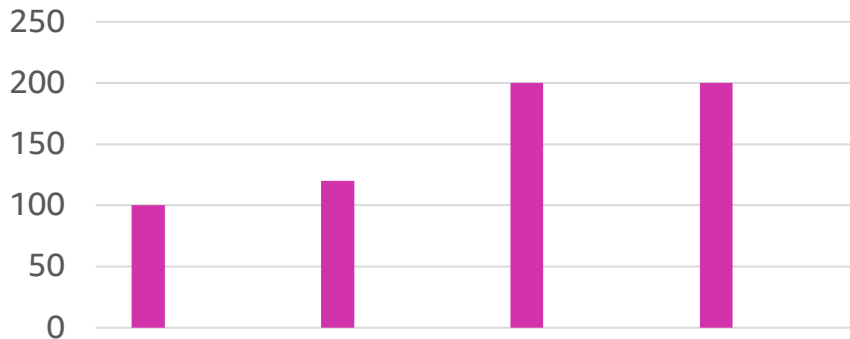
**Amazon
CloudWatch**

システムメトリックスやログを
CloudWatchへアップロード

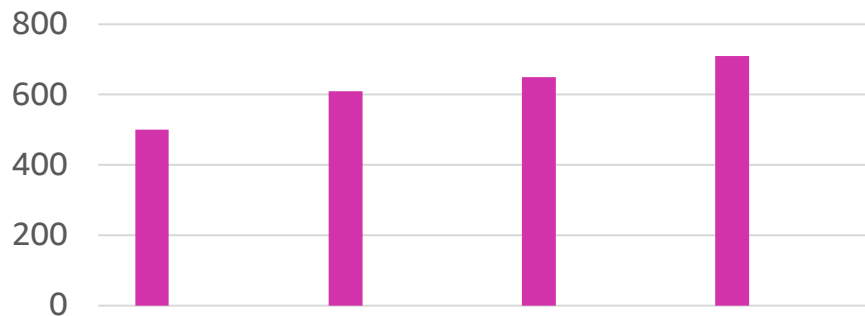
Performance improvement over time

Aurora MySQL – 2015-2018

Max write throughput – up 100% (K)



Max read throughput – up 42% (K)



多くのパフォーマンス最適化に加えて、HWプラットフォームもアップグレード

Launched with R3.8xl
32 cores, 256GB memory

Now support R4.16xl
64 cores, 512GB memory

R5.24xl coming soon
96 cores, 768GB memory

Auroraのストレージ

- SSDを利用したシームレスにスケールするストレージ

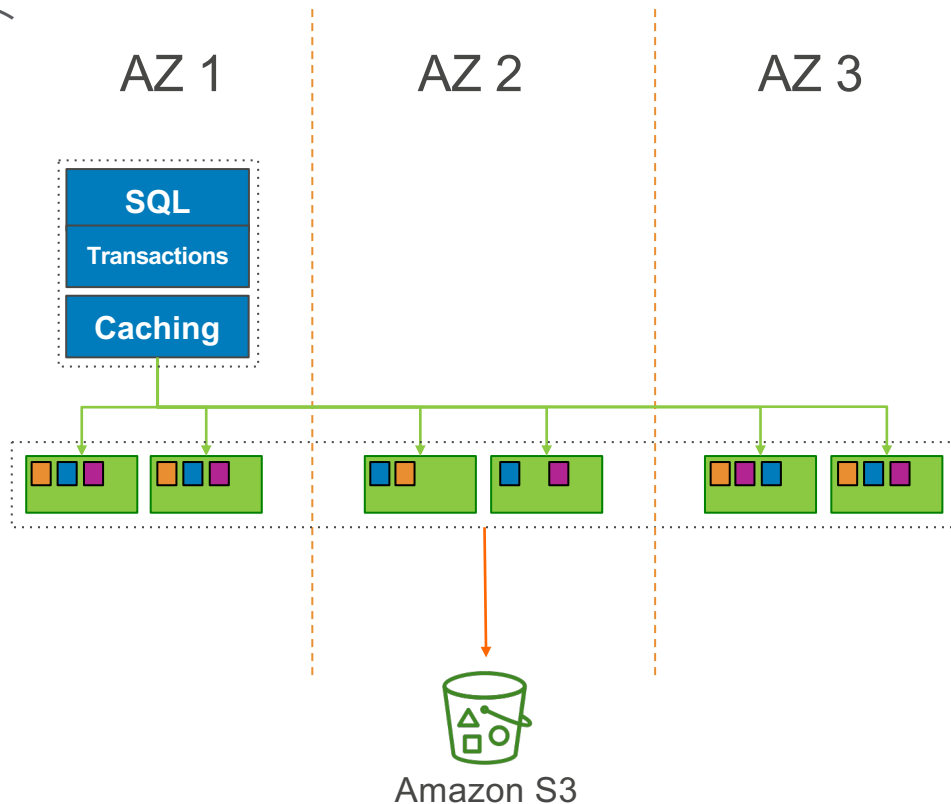
- 10GBから64TBまでシームレスに自動でスケールアップ
- 実際に使った分だけ課金
- Peer-to-peer gossipレプリケーション

- 標準で高可用性を実現

- 3AZに6つのデータのコピーを作成
- 2つのディスクが利用不能でも読み書き可能
 - 万が一1つのAZが利用不能になっても3本で読み書き可能な状態で稼働
- 3つのディスクが利用不能でも読み込みは可能

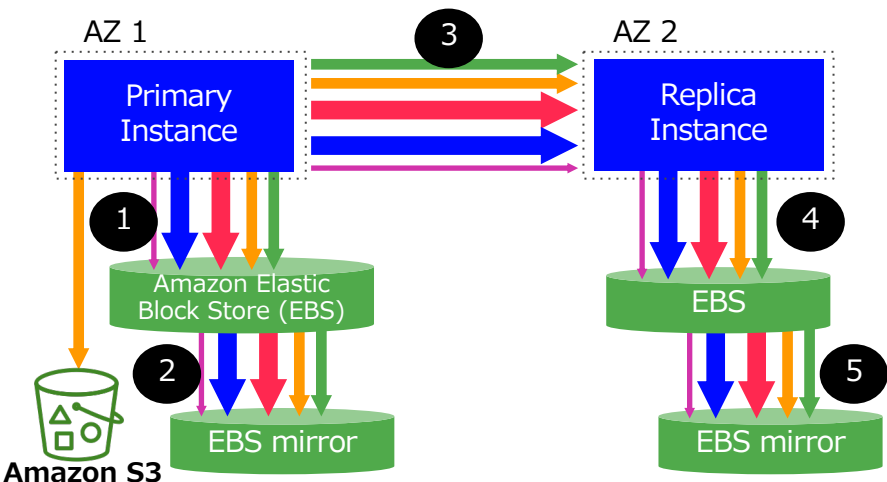
- Log structured Storage

- redo logを複数の小さなセグメントに分割
- Log pageによってData pageを作成



Aurora I/O profile

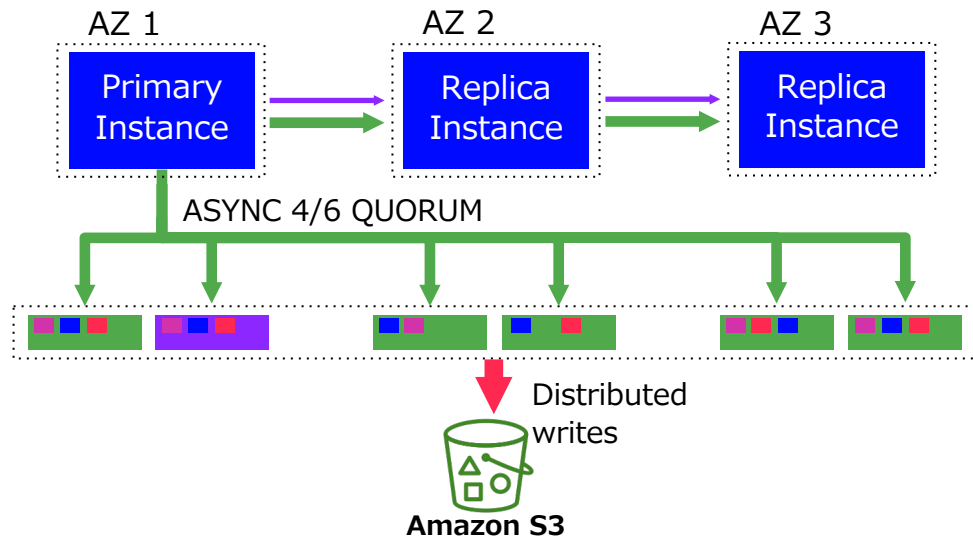
MySQL with Replica



MySQL I/O profile for 30 min Sysbench run

780K transactions
 7,388K I/Os per million txns (excludes mirroring, standby)
 Average 7.4 I/Os per transaction

Amazon Aurora



Aurora IO profile for 30 min Sysbench run

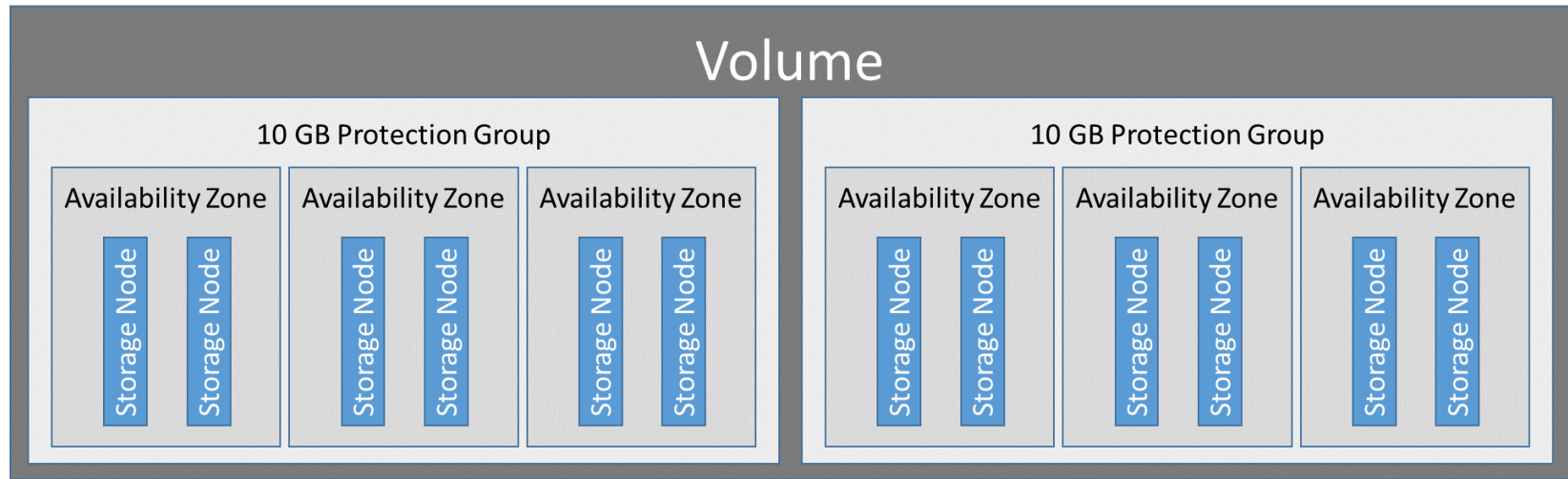
27,378K transactions – **35X MORE**
 0.95 I/Os per transaction (6X amplification) – **7.7X LESS**

→ Log
 → Binlog
 → Data
 → Double-write
 → From files



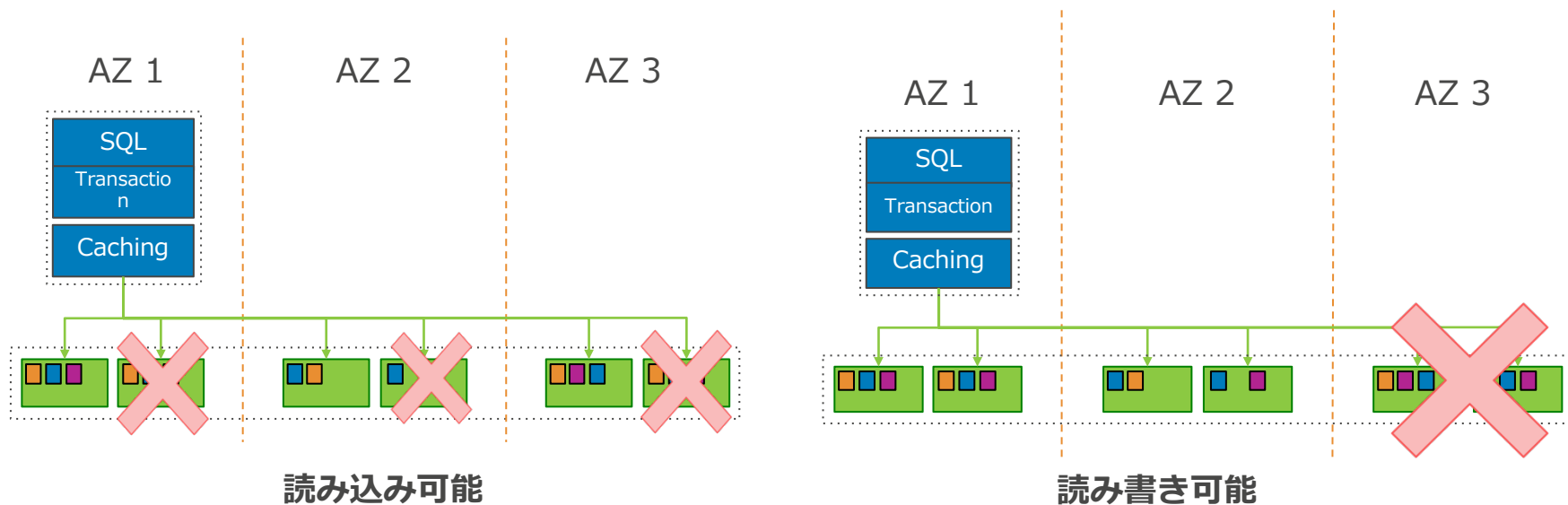
ストレージノードクラスタ

- Protection Group毎に6つのストレージノードを使用
- 各ログレコードはLog Sequence Number(LSN)を持っており不足・重複しているレコードを判別可能
 - 不足している場合はストレージノード間でgossip protocolを利用し補完を行う



ディスク障害検知と修復

- 2つのコピーに障害が起こっても、読み書きに影響は無い
- 3つのコピーに障害が発生しても読み込みは可能
- 自動検知、修復



クォーラムモデル

- レプリケーション管理のためのクォーラムのルール

- $V_r + V_w > V$ (読み込みクォーラム(V_r)、書き込みクォーラム(V_w)が、少なくとも1つ共通のコピーを保持)
- $V_w > V/2$ (書き込みクォーラムは、過半数のコピーを保持)

V (コピーの数)	V_w (書き込みクォーラム)	V_r (読み込みクォーラム)
1	1	1
2	2	1
3	2	2
4	3	2
5	3	3
6	4	3
7	4	4

クォーラムモデル

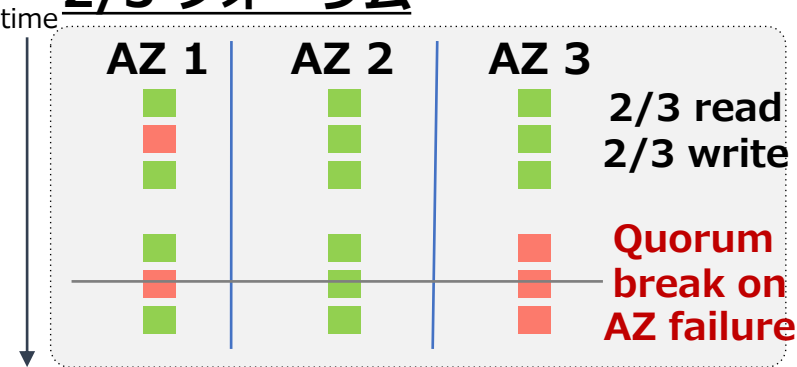
- レプリケーション管理のためのクォーラムのルール

- $V_r + V_w > V$ (読み込みクォーラム(V_r)、書き込みクォーラム(V_w)が、少なくとも1つ共通のコピーを保持)
- $V_w > V/2$ (書き込みクォーラムは、過半数のコピーを保持)

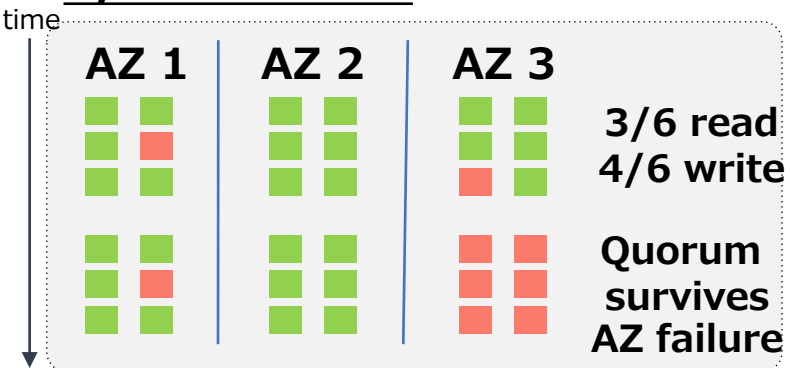
V (コピーの数)	V_w (書き込みクォーラム)	V_r (読み込みクォーラム)
1	1	1
2	2	1
3	2	2
4	3	2
5	3	3
6	4	3
7	4	4

なぜ6つのコピーが必要なのか？

2/3 クォーラム



4/6 クォーラム



- AWS は大規模環境
- AZ障害は影響範囲が広域
- AZ + 1の障害(二重障害)を許容し、修復する必要がある
- 3つのAZの場合、6つのコピーが必要

メンバーシップチェンジ



Epoch 1: 全てのノードがHealthy



Epoch 2: ノードFは疑わしい状態と認識されると、第2のクォーラムグループがノードGを用いて作成され、両方のクォーラムがアクティブ



Epoch 3: ノードFはunhealthyであることが確認されます。ノードGを持つ新しいクォーラムグループがアクティブになる

- ほとんどのシステムは、メンバーシップの変更にコンセンサスを使用しますがジッタとストールを引き起こす
- Auroraはクォーラムセットとエポックを使用
- 停止なしのAZ + 1のフォールトトレランス、積極的に障害を検出可能

クォーラムの効率化



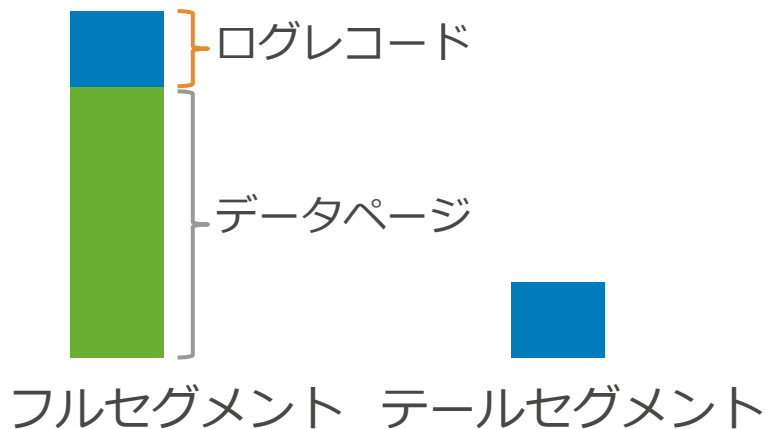
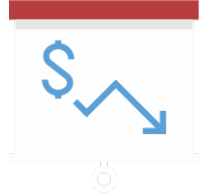
各データブロックに対して、クォーラムグループ内の少なくとも4つのノードが最新のデータを持つ



これら4つのノードのいずれかから読み取ると、最新のデータが返される

- ほとんどのクォーラムベースのシステムでは読み込みにコストがかかる
- Auroraはどのノードが最新でレイテンシが少ないかの情報を持っている
- リードクォーラムは、修理やクラッシュリカバリに必要

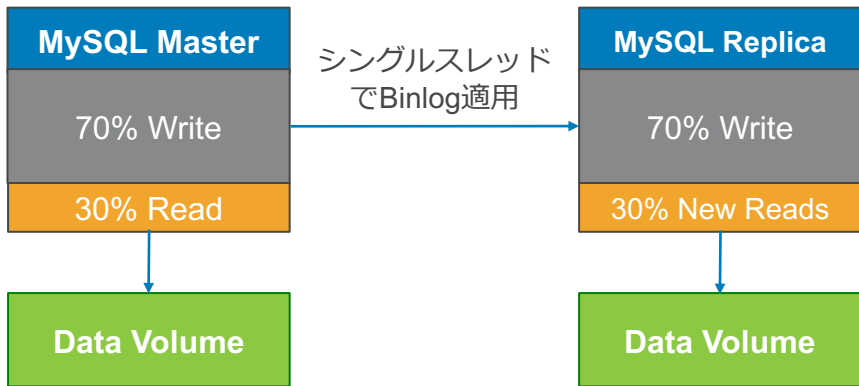
コストを抑えるための工夫



- 6つのコピーは全て同一ではない
 - 3つのフルセグメント：データページとログレコード
 - 3つのテールセグメント：ログレコードのみ
- 6つのコピーが必要だが、物理ストレージとして必要な容量は、6倍ではなく、3倍より少し多い程度

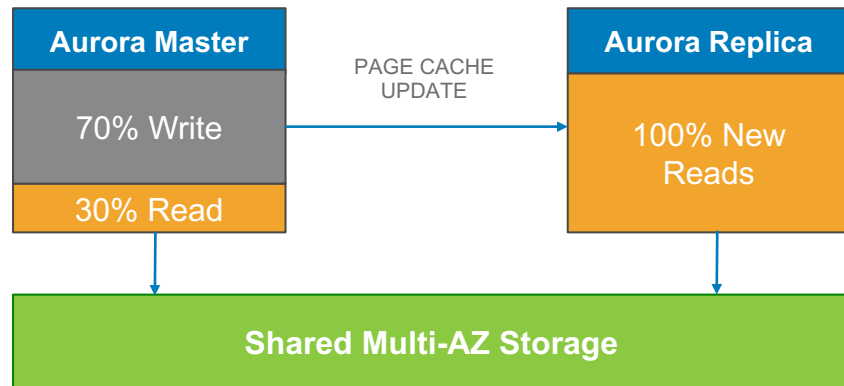
※ ストレージ料金は、実際に使った容量のみ(3倍～6倍になるわけではない)

レプリケーション



MySQL read scaling

- レプリケーションにはbinlog / relay logが必要
- レプリケーションはマスターへ負荷がかかる
- レプリケーション遅延が増加していくケースがある
- フェイルオーバーでデータロスの可能性がある

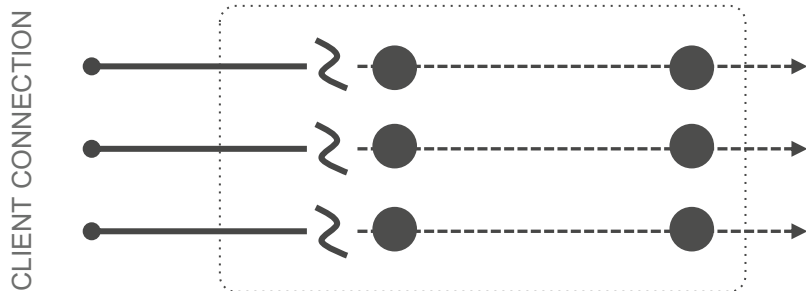


Amazon Aurora read scaling

- Binlogによるレプリケーションではない
- マスターへの負荷を最小限に15台までリードレプリカを作成可能
- 10-20ms程度のレプリケーション遅延
- フェイルオーバーでデータロスが無い

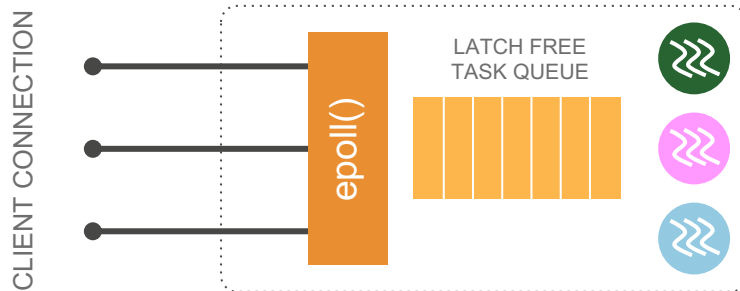
アダプティブスレッドプール

MYSQL THREAD MODEL



- Standard MySQL – コネクション毎に1
- コネクション数に応じてスケールしない
- MySQL EE – スレッドプール毎にコネクションをアサイン
- しきい値を慎重に設定する必要がある

AURORA THREAD MODEL

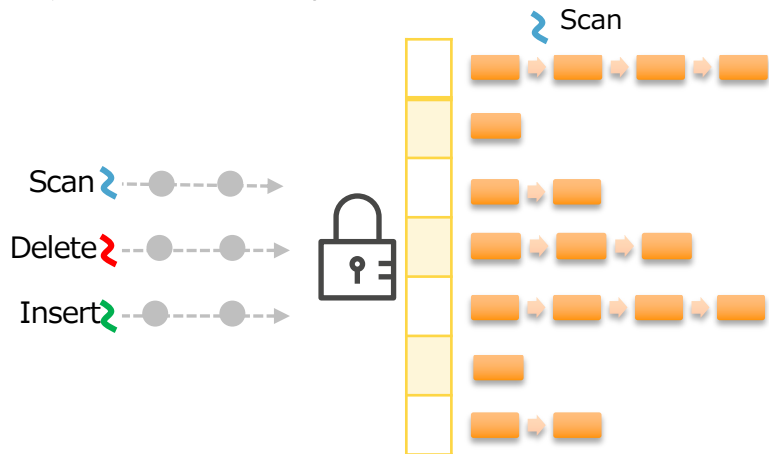


- アクティブなスレッドに複数のコネクションを収容
- Kernel空間の `epoll()` がラッチフリーキューにコネクションをアサイン
- スレッドプールの数は動的に調整される
- r3.8xlインスタンスのAmazon Auroraで5,000同時コネクションを扱える

Aurora lock management

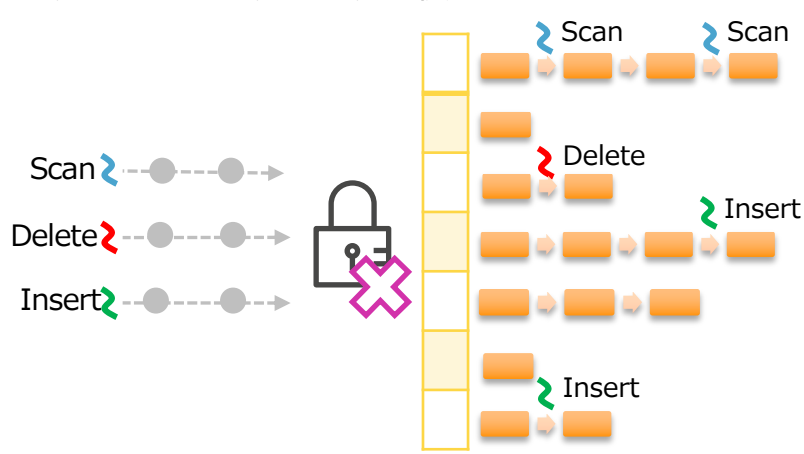
MySQL lock manager

MySQLと同じロックセマンティクス
ロックチェーンへの同時アクセス



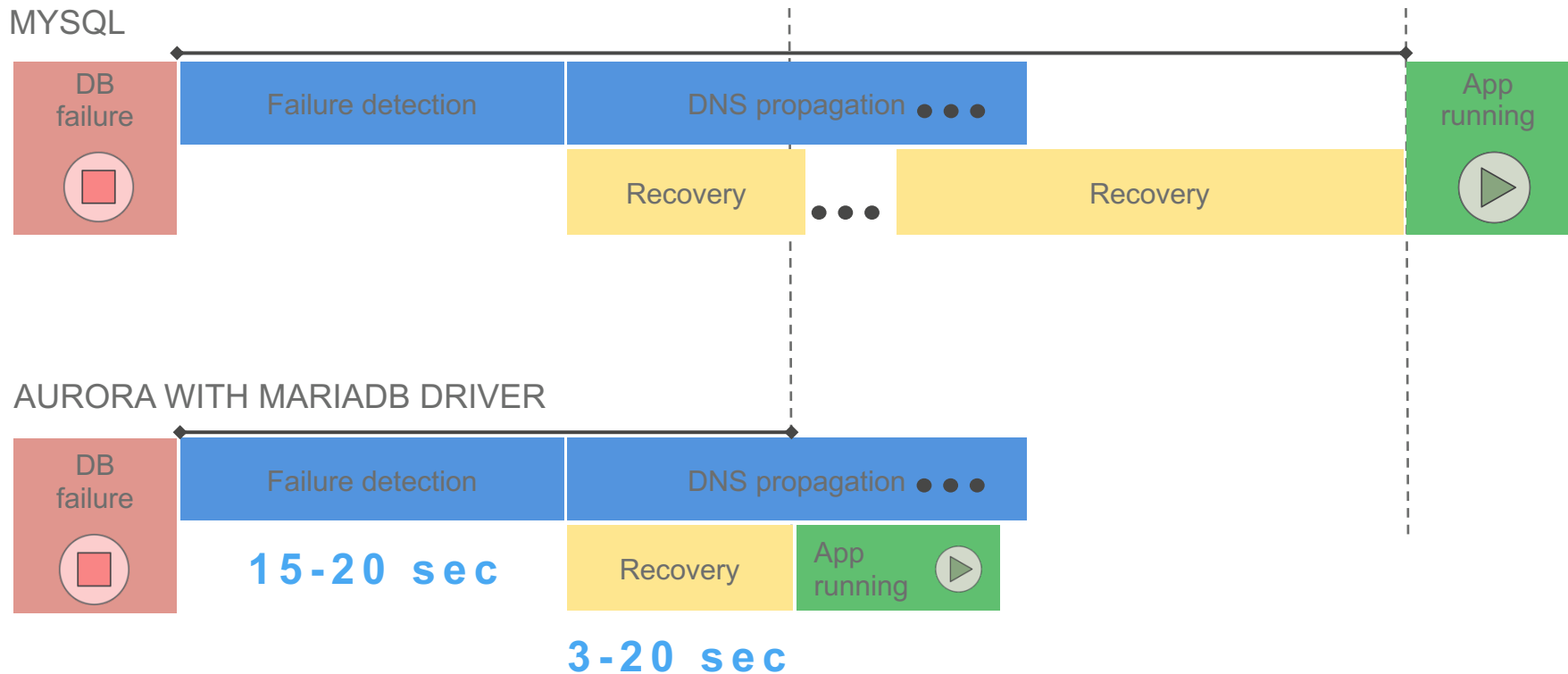
Aurora lock manager

個々のロックチェーン内の複数のスキャナ
ロックフリーデッドロック検出



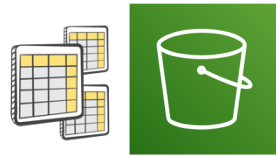
多くの同時セッションをサポートする、高い更新スループット

高速でより予測可能なフェイルオーバー時間



Streaming snapshotとPITR

- Amazon Auroraでは各セグメント毎にAmazon S3へ継続的に増分バックアップを取得している
 - Backup retention periodでバックアップを残す期間を指定可能
- Amazon Auroraが使用しているディスクの仕組みによりパフォーマンスへ影響を与えない
- PITRで5分前からBackup Retention Periodまでの任意の位置に秒単位で復元可能

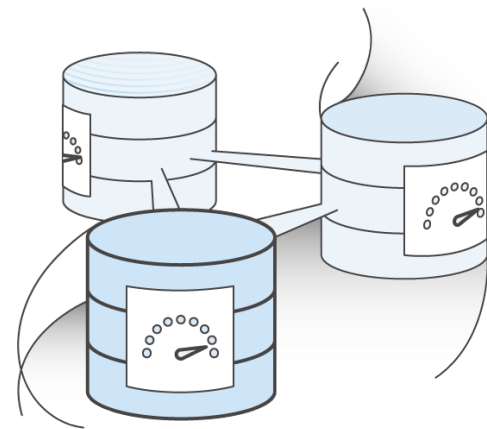


SQLによるフェイルオーバのテスト

SQLによりノード・ディスク・ネットワーク障害をシミュレーション可能

- データベースノードのクラッシュをシミュレート:
`ALTER SYSTEM CRASH [{INSTANCE | DISPATCHER | NODE}]`
- レプリケーション障害をシミュレート:
`ALTER SYSTEM SIMULATE percentage_of_failure PERCENT
 READ REPLICA FAILURE [TO ALL | TO "replica name"]
 FOR INTERVAL quantity [YEAR | QUARTER | MONTH | WEEK | DAY | HOUR |
 MINUTE | SECOND];`
- 他にも
 - ディスク障害をシミュレート
 - ディスクコンジェスションをシミュレート

新機能



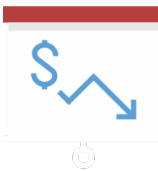
Global Transaction ID support

- Aurora MySQL 2.04以降で対応
 - レプリケーションソース・ターゲットとして利用可能
 - 設定方法などの詳細はドキュメントを参照
- 新規パラメータ
 - gtid_mode (Management Consoleではgtid-mode)
 - enforce_gtid_consistency

```
CALL mysql.rds_set_external_master_with_auto_position (  
  host_name  
  , host_port  
  , replication_user_name  
  , replication_user_password  
  , ssl_encryption  
);
```

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/mysql-replication-gtid.html>

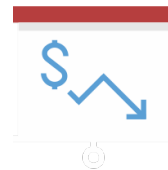
Aurora backup billing



- Amazon CloudWatchで、継続バックアップとスナップショットで使用している容量を確認可能に(Aurora DB クラスターごとに個別に計算される)
 - **BackupRetentionPeriodStorageUsed**: 継続的バックアップの保存に使用されているバックアップストレージの量。クラスターボリュームのサイズと、保存期間中に行った変更の量によって変化する
 - クラスターサイズが 100 GiB で、保存期間が 2 日間である場合、BackupRetentionPeriodStorageUsed の最大値は 200 (100 GiB + 100 GiB) となります
 - **SnapshotStorageUsed**: 手動スナップショットを保存するために使用されているバックアップストレージの量。保存期間内に作成された手動スナップショットや自動スナップショットは、バックアップストレージ量にカウントされない
 - 保存期間外のスナップショットが 1 つあり、このスナップショットを作成した時点のクラスターのボリュームサイズが 100 GiB であった場合、SnapshotStorageUsed の値は 100 です

https://docs.aws.amazon.com/ja_jp/documentdb/latest/developerguide/backup-restore.understanding-storage.html

Aurora backup billing



– TotalBackupStorageBilled:

BackupRetentionPeriodStorageUsed と SnapshotStorageUsed の合計から、1日のクラスターボリュームのサイズと等しいバックアップストレージの量を引いた値

- データベースサイズが 100 GiB、保存期間が 1 日で、保存期間外のスナップショットが 1 つある場合、TotalBackupStorageBilled は 100 (100 GiB + 100 GiB - 100 GiB) となります

https://docs.aws.amazon.com/ja_jp/documentdb/latest/developerguide/backup-restore.understanding-storage.html



起動バージョン・バージョンアップ対象の指定方法改善

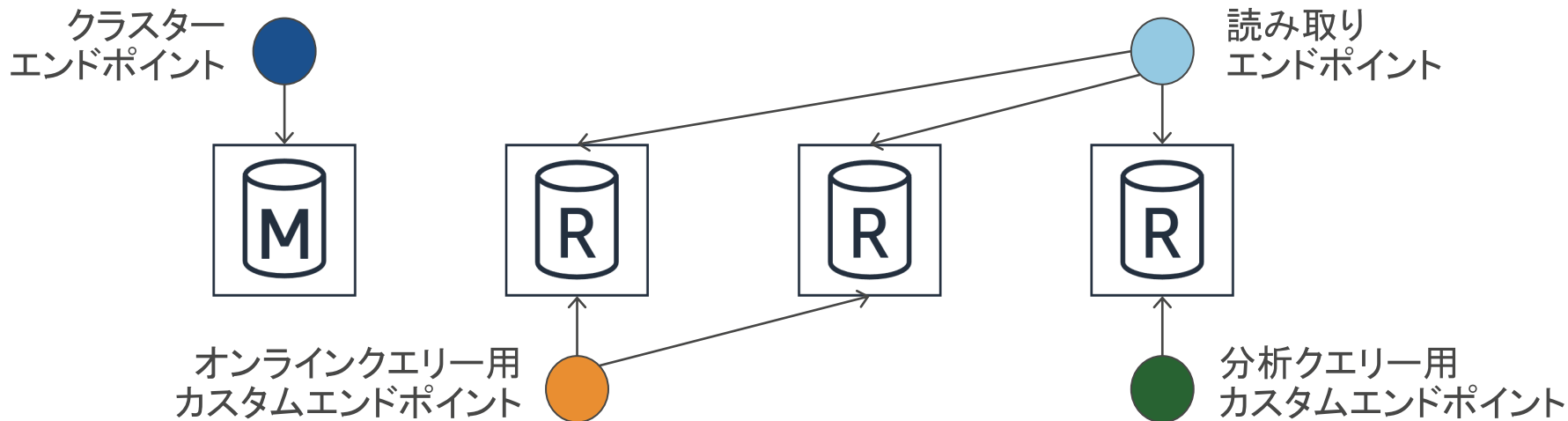
- Aurora MySQL 1.19, 2.03.2以上にバージョンアップを行う場合、ターゲットバージョンを指定可能に
 - Aurora MySQL クラスター起動時にも指定可能
 - バージョンアップグレードが Pending Maintenance Action ではなく Modify DB Instance で Engine Version 指定に
 - 例: `--engine-version 5.7.mysql_aurora.2.03.2` or `5.6.mysql_aurora.1.19.0`

DB engine version [Info](#)

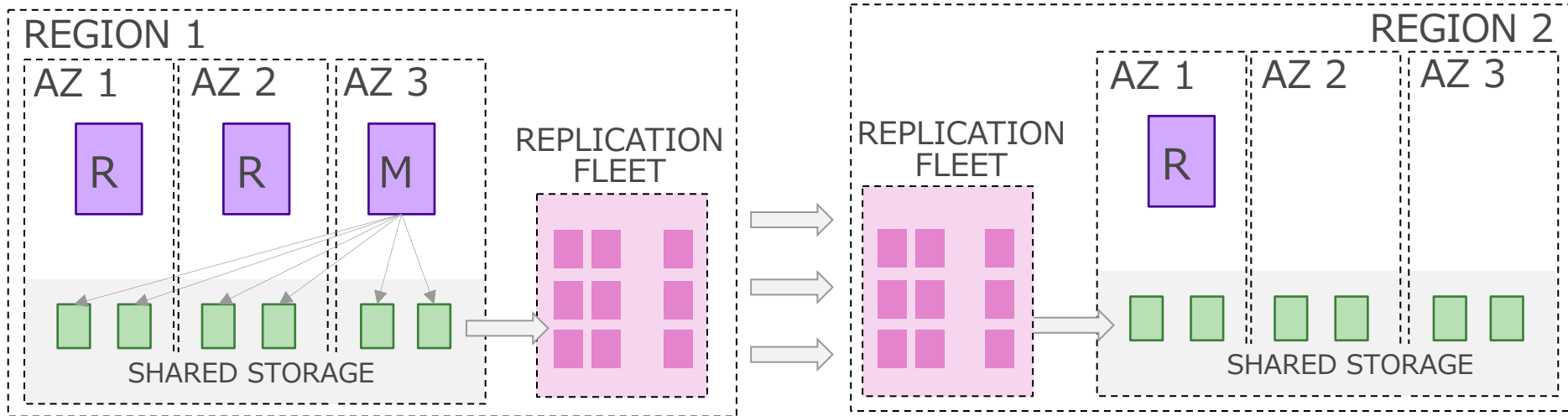
Aurora (MySQL)-5.7.12	▼
Aurora (MySQL 5.7) 2.03.3	
Aurora (MySQL 5.7) 2.03.2	
Aurora (MySQL)-5.7.12	

Custom Endpoints

- Auroraクラスター内の、どのインスタンスを含めるかをユーザが指定可能なエンドポイントが作成可能
- オンラインクエリ用のリードレプリカと分析クエリ用のリードレプリカを分離することが可能に
 - Readerだけを追加/Writerも含めて追加可能が選択可能
 - 明示的に指定したノードだけを追加や新規起動ノードを自動で追加するなどオプションがある



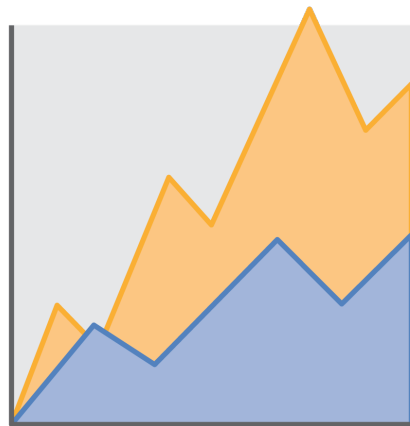
Global Database



- 通常1秒未満の低レイテンシなレプリケーションと、通常1分以内の高速なフェールオーバーが可能
 - ソースDBクラスタにパフォーマンス影響を与えない
- Binlogを利用せず、Auroraストレージレベルでクラスタ間のレプリケーションを行う
 - レプリケーション先には書き込みはできない
- 2019/4時点の対応リージョン
 - バージニア北部 / オハイオ / オレゴン / アイルランド

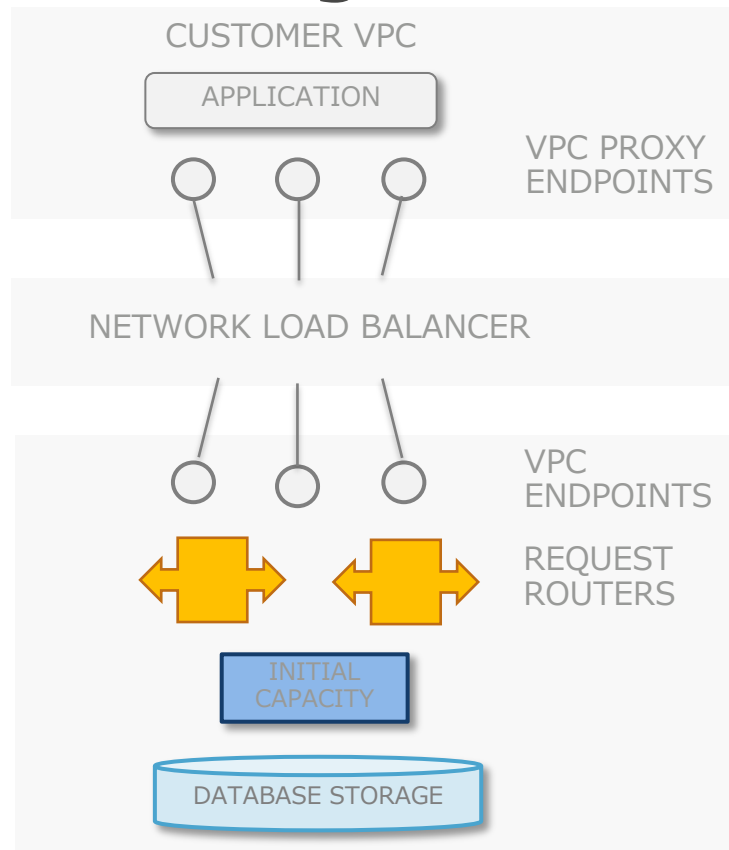
Aurora Serverless

- 不可の少ないアプリケーション（例：アクセスの少ないブログサイト）
- 可変負荷のアプリケーション - 予測が困難な活動のピーク（ニュースサイトなど）
- 夜間または週末に必要とされない開発データベースまたはテストデータベース
- マルチテナントSaaSアプリケーションの統合基盤



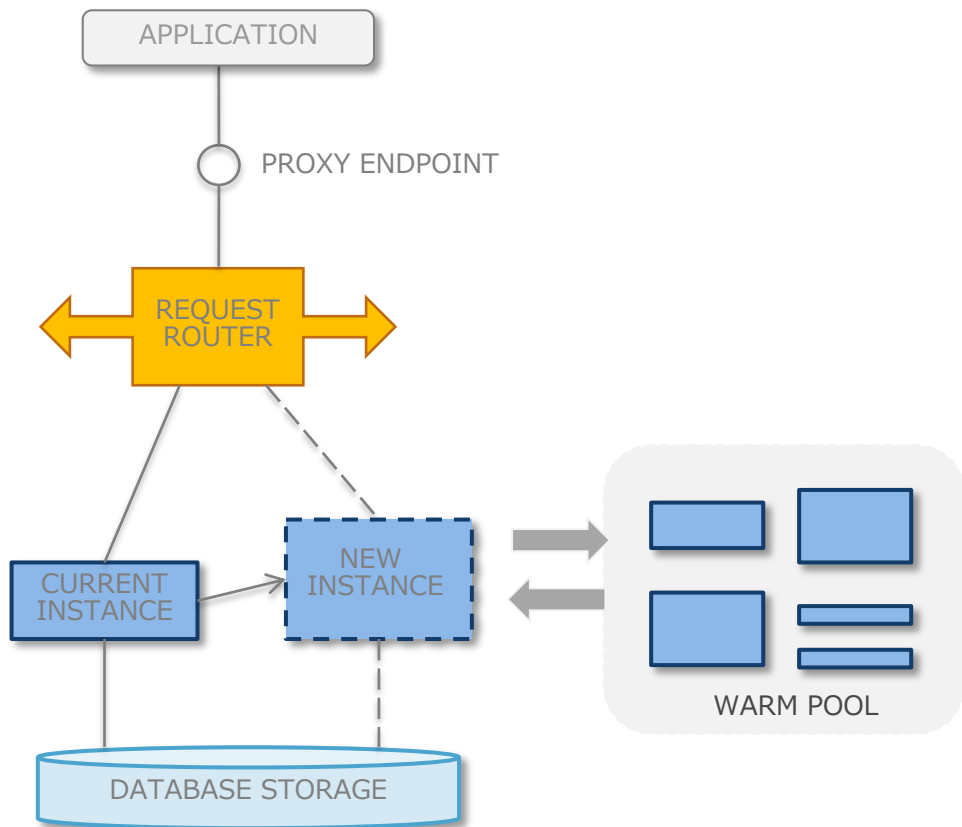
Serverless Endpoint Provisioning

- Aurora Serverless:
 - Auroraストレージボリュームを作成
 - アプリケーション接続のためのVPC内のプロキシエンドポイントの作成
 - プロキシの背後にネットワークロードバランサを構成
 - リクエストルータを初期化してデータベーストラフィックをルーティングする
 - 初期容量を確保



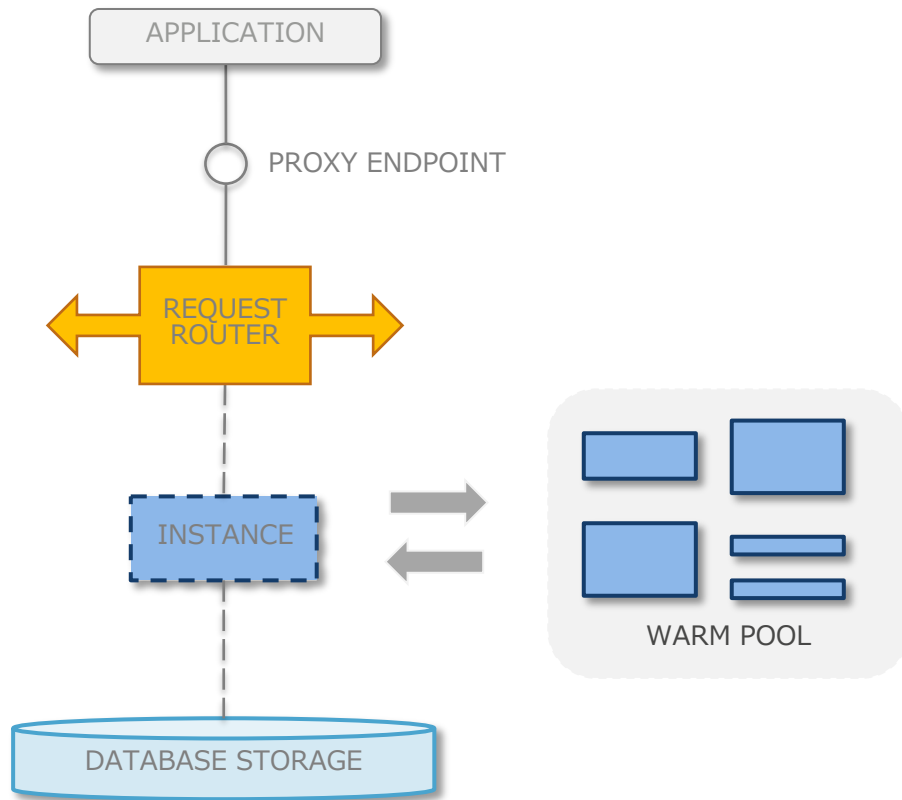
スケールアップ・ダウン

- Auroraはデータベース（CPU、メモリ、接続）のアプリケーション負荷を監視
- スケーリングのしきい値に達すると、インスタンスが自動的にスケールアップまたはスケールダウン
- スケーリング操作はアプリケーションに透過的。接続とセッションの状態は新しいインスタンスに転送される
- スケーリングの最小および最大値を設定可能

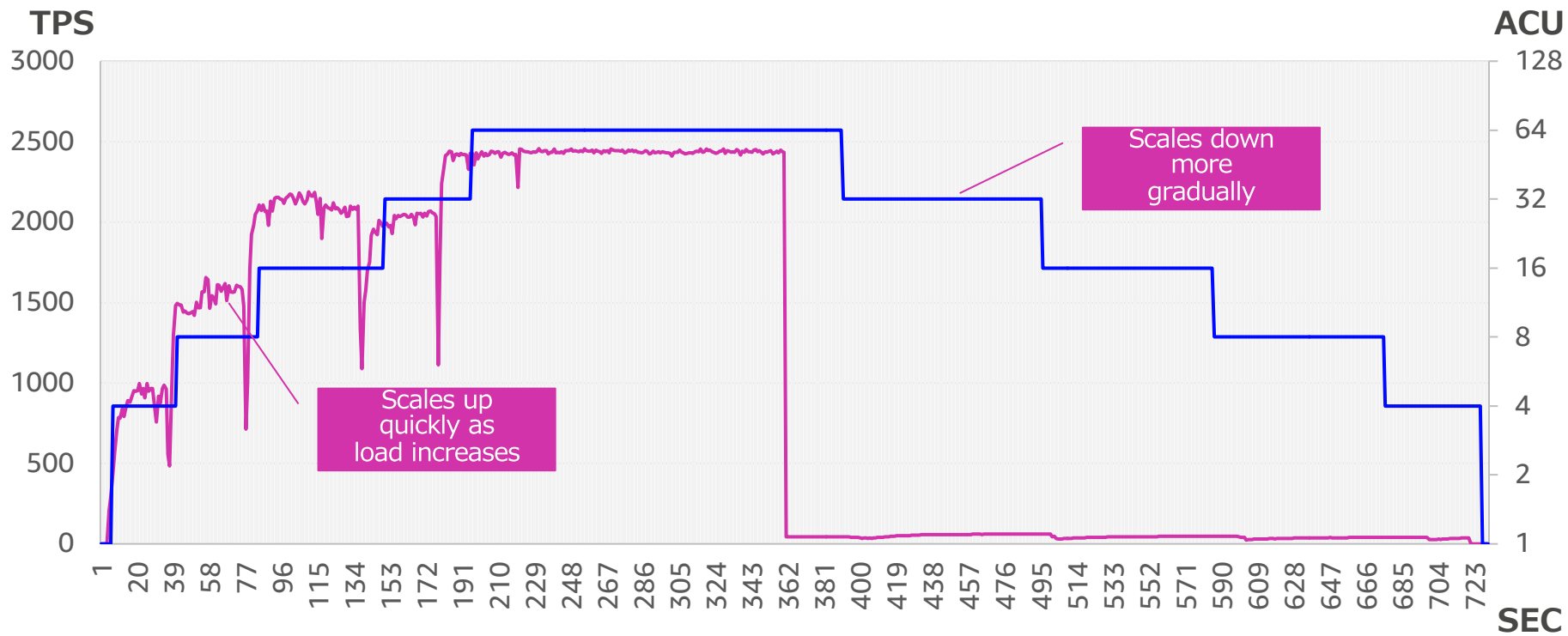


0からスケール

- 必要に応じて、ユーザー定義の非アクティブ期間が経過するとインスタンスが削除される
- データベースが停止している間は、ストレージの支払いのみ
- データベースへのリクエストによりプロビジョニング
- バッファキャッシュをコピーする実行中のサーバーがないため、0へのスケールはより高価なオペレーションです

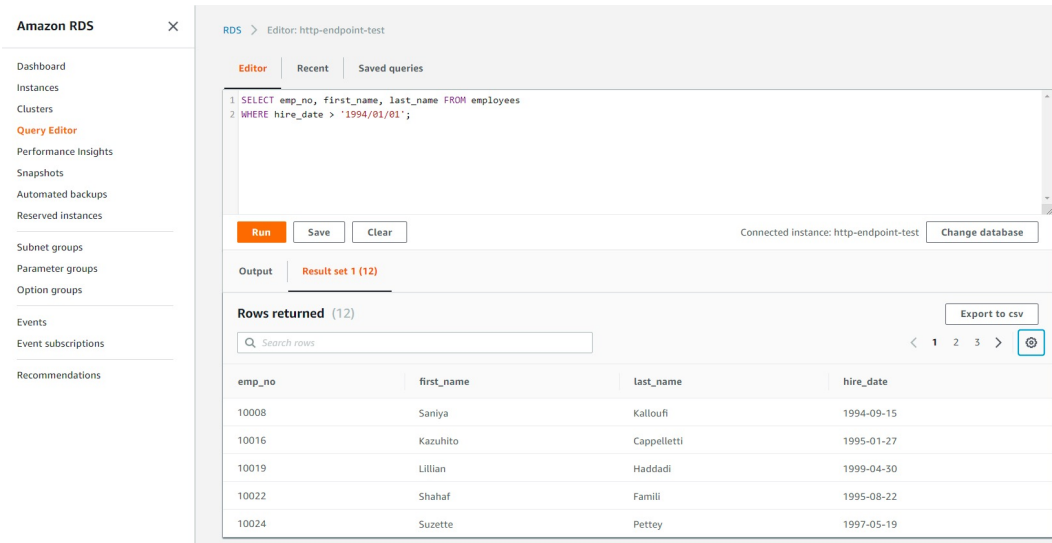


負荷に応じたスケールアップ・ダウン



Amazon Aurora Serverless Queries Directly from the AWS Management Console (Beta)

- Amazon Aurora Serverlessクラスタに直接マネジメントコンソールからクエリを実行可能
 - Query EditorへのアクセスはIAMで制御下
- よく利用するクエリを保存可能
 - 履歴機能もあり
- 2019/4現在バージニアリージョンで利用可能

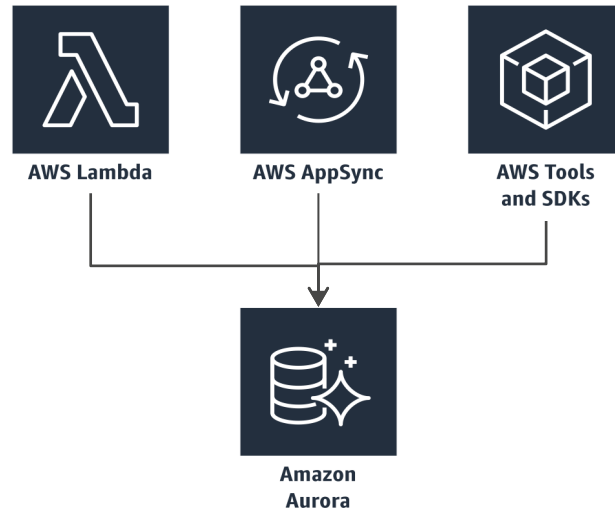


The screenshot displays the Amazon RDS Query Editor interface. On the left is a navigation sidebar with options like Dashboard, Instances, Clusters, Query Editor (highlighted), Performance insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations. The main area shows the 'Editor' tab with a SQL query: `1 SELECT emp_no, first_name, last_name FROM employees; 2 WHERE hire_date > '1994/01/01';`. Below the query are 'Run', 'Save', and 'Clear' buttons. The 'Output' section shows 'Result set 1 (12)' with an 'Export to csv' button. A table of results is displayed below, showing columns for emp_no, first_name, last_name, and hire_date.

emp_no	first_name	last_name	hire_date
10008	Saniya	Kalloufi	1994-09-15
10016	Kazuhito	Cappelletti	1995-01-27
10019	Lillian	Haddadi	1999-04-30
10022	Shahaf	Famili	1995-08-22
10024	Suzette	Petty	1997-05-19

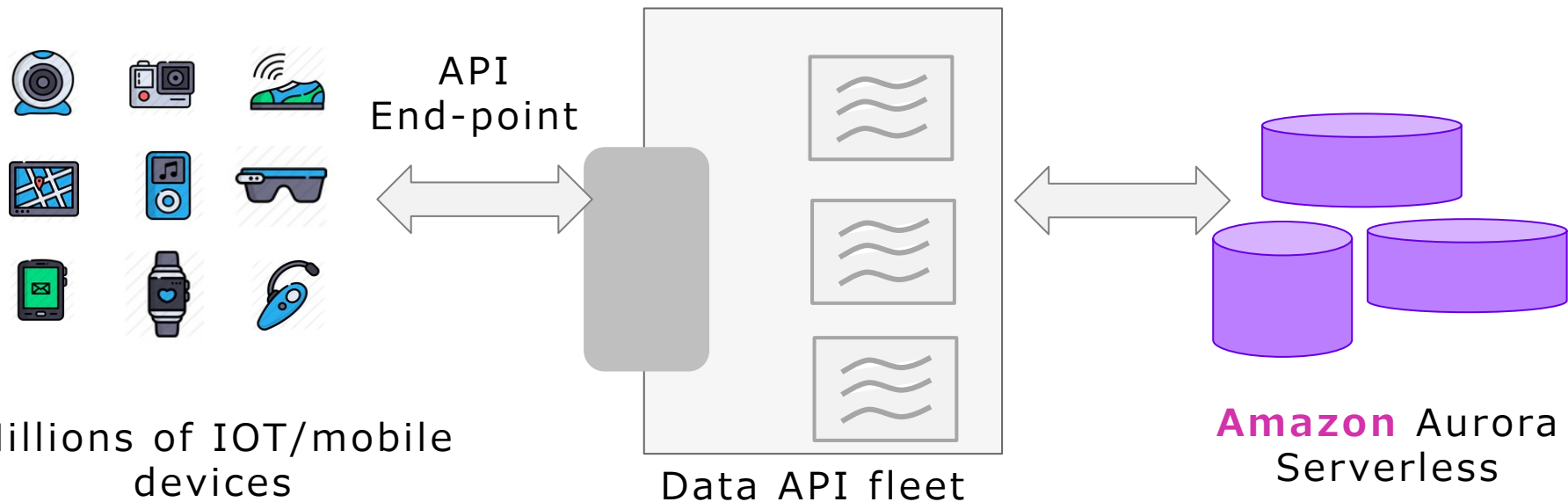
Amazon Aurora Serverless Database with the New Data API (Beta)

- Aurora Serverless へのアクセス方法として、MySQLネイティブプロトコルに加え、HTTPSエンドポイントおよび AWS SDK からのアクセスを提供
- AWS Lambda や AWS AppSync から、VPC にアクセスすることなく Aurora を利用可能
- クエリー結果はJSON形式で戻される
 - 1,000行および1MBが上限
- 同期APIであり、1分以内に完了しない場合は中断される
- 2019/4時点では、バージニア北部でベータ版として提供



```
aws rds-data execute-sql --db-cluster-or-instance-arn arn:aws:rds:us-east-1:123456789012:cluster:mydbcluster --schema "" \  
--database "mydatabase" --aws-secret-store-arn "arn:aws:secretsmanager:us-east-1:123456789012:secret:mysecret" \  
--sql-statements "select * from mytable" --region us-east-1 --no-verify-ssl \  
--endpoint-url https://rds-data.us-east-1.amazonaws.com --profile myprofile
```


Amazon RDS Data API for serverless applications



シンプルなwebインターフェース経由でアクセス

- どこからでもアクセスできるパブリックエンドポイント
- クライアントに特別な設定は不要
- 永続接続は不要

サーバーレスアプリケーション(Lambda)
軽量なアプリケーション(IOT)

Parallel Query

Orders of magnitude
faster queries

Parallelism increases
with data size

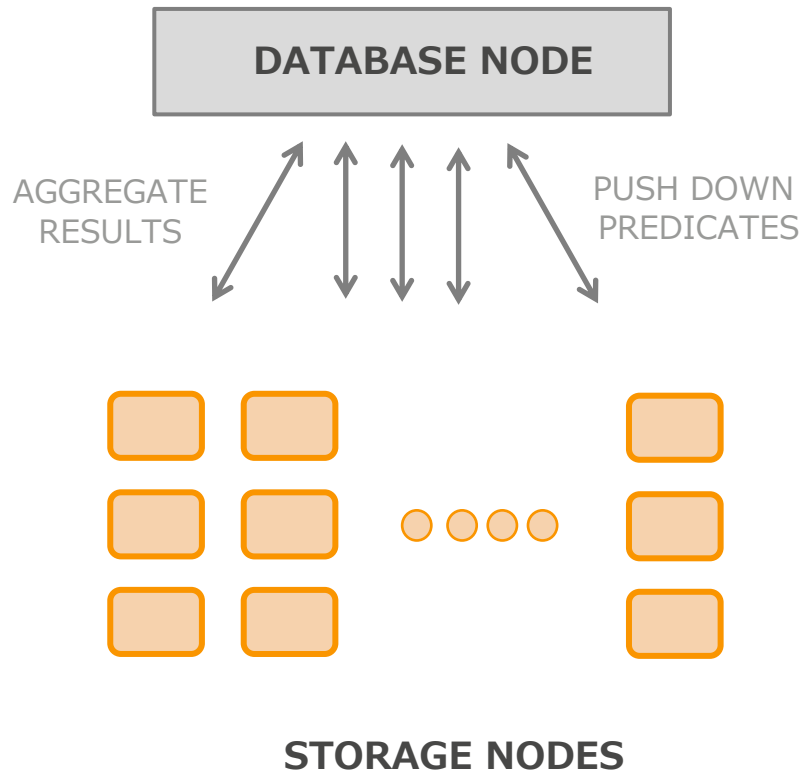
Reduced contention
with OLTP workload

OLTPワークロードの分析クエリ

- リアルタイムデータの分析
- アドホッククエリに対するETL
パイプラインを作らなくて良い
- 複数の分析クエリを同時実行

クエリをストレージノードの数千のCPUに プッシュダウン

- Auroraストレージには数千のCPUがある
 - ストレージ・フリートを使用して問合せ処理をプッシュダウンおよび並列化する機能を提供
 - v1: single-table predicates (selections, projections, 200+ SQL functions, case statements, filters) and hash joins
 - v2: group by, order by, aggregation
 - データに近い箇所で処理を実行すると、ネットワークのトラフィックとレイテンシを軽減出来る



データベースクラスタのstop/startサポート



- Amazon Auroraで、データベースクラスタのstop/startをサポート
 - 常にデータベースクラスタを実行する必要がない場合に簡単に手頃な価格で開発やテスト用にデータベースを使用可能に
 - データベースクラスタ停止中も、指定された自動バックアップ保持期間内であれば、ポイントインタイムの復元を実行可能
- データベースクラスタを停止すると、WriterインスタンスとReaderが停止
 - データベースクラスタの停止中に、指定された保持期間内のクラスタストレージ、手動のスナップショット、およびバックアップストレージに対しては課金されますが、データベースインスタンス時間に対しては課金されない
- データベースクラスタは 1 回で最大 7 日まで停止可能。7 日後に自動的に開始

Deletion Protection



- 削除保護フラグを設定可能
- 削除保護が有効な場合、インスタンスの削除リクエストはブロックされるため、インスタンスの削除リクエスト前に削除保護の無効化が必要
- Auroraの場合、クラスタに存在している最後のインスタンス削除時(クラスタ削除時)に有効

Performance Insights



Amazon RDS
Performance
Insights

- SQL/User/Host/Wait event毎に実行されたクエリのパフォーマンスを閲覧可能
 - データベースのパフォーマンスをクエリレベルでドリルダウンして調査える
 - API/SDK対応
 - CloudWatchにDBLoad/DBLoadCPU/DBLoadNonCPUをパブリッシュ可能
- Aurora MySQL 1.17.3以上から対応
 - T2インスタンスファミリー・5.7互換のAurora MySQLは非対応
- 7日間のデータ保存は無料/2年間保持できるプランはインスタンスファミリー・リージョンによって価格が異なる
 - <https://aws.amazon.com/jp/rds/performance-insights/pricing/>

Performance Insights



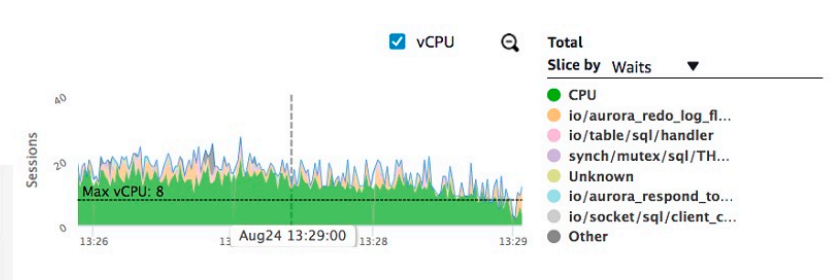
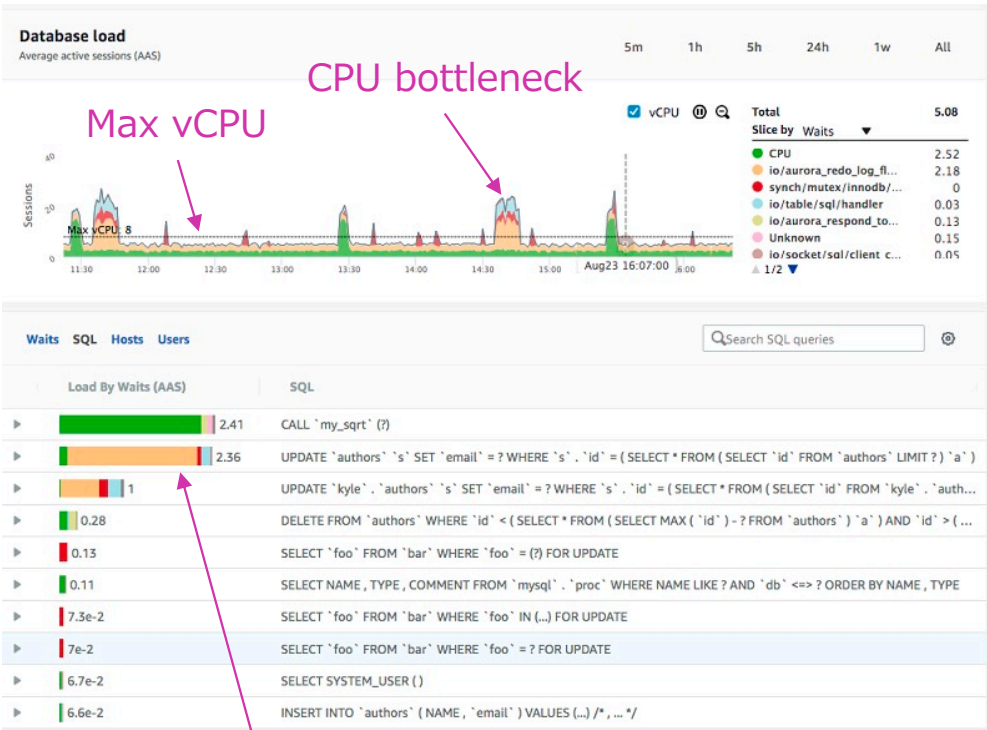
Amazon RDS
Performance
Insights

- 注意点

Aurora MySQL performance insightsは以下のパラメータを自動で有効にします

- performance_schema=1
 - performance-schema-consumer-events-waits-current=ON
 - performance-schema-instrument='wait/%=ON'
 - performance-schema-consumer-global-instrumentation=ON
 - performance-schema-consumer-thread-instrumentation=ON
- 一般的にパフォーマンススキーマ(P_S)有効化時のように、メモリ利用量の増加とパフォーマンスへの影響があります
 - AuroraではP_S有効化時の影響を軽減させる実装を入れていますが影響が0ではないため、ご注意ください
 - performance_schema パラメータの値を変更すると、DB インスタンスの再起動が必要になります

Performance Insights



Waits SQL Hosts Users

Search SQL queries

Load By Waits (AAS)	SQL
13.03	CALL `my_sqrt` (?)
2.16	UPDATE `authors` `s` SET `email` = ? WHERE `s`.`id` = (SE...
0.27	DELETE FROM `authors` WHERE `id` < (SELECT * FROM (SELE...

Legend (Total: 5.08):
 - CPU: 2.52
 - io/aurora_redo_log_fl...: 2.18
 - io/table/sql/handler: 0
 - synch/mutex/innodb/...: 0.03
 - synch/mutex/innodb/...: 0.13
 - io/aurora_respond_to...: 0.15
 - io/socket/sql/client_c...: 0.05
 - Other: 0

Waits SQL Hosts Users

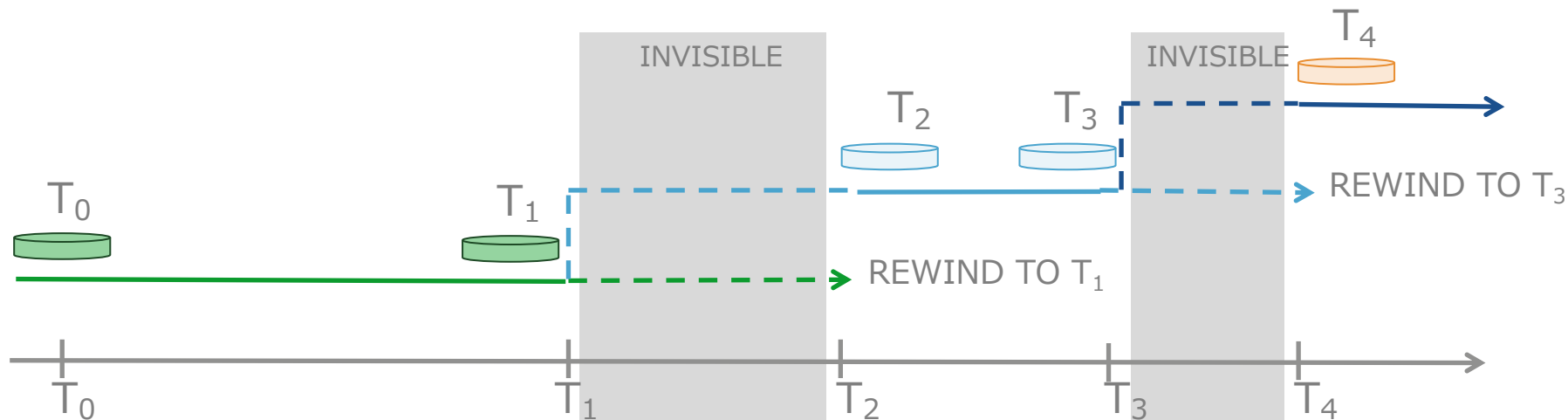
Search SQL queries

Load By Waits (AAS)	SQL
12.04	UPDATE `kyle`.`authors` `s` SET `email` = ? WHERE `s`.`id`...
3.88	UPDATE `authors` `s` SET `email` = ? WHERE `s`.`id` = (SE...
1.32	CALL `my_sqrt` (?)
0.25	DELETE FROM `authors` WHERE `id` < (SELECT * FROM (SELE...

SQL w/ high CPU



Backtrack

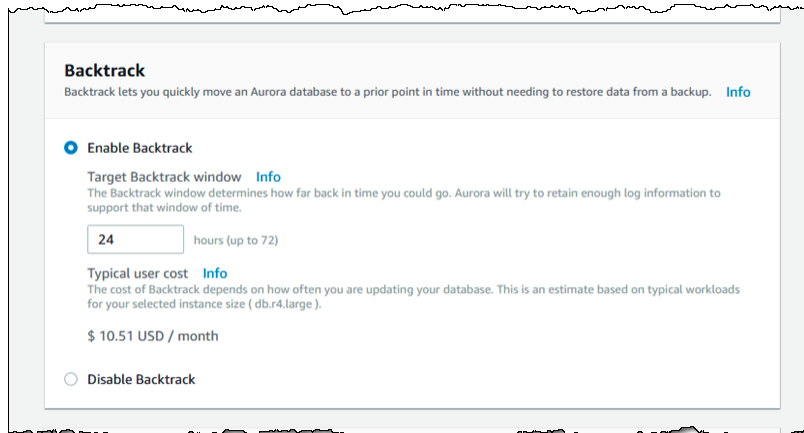


データベースの状態を容量によらず瞬時に特定の時点へ巻き戻す

オペミスなどをしてしまった場合に、作業実行前の状態にすぐに巻き戻すことでサービスへの影響を最小限に抑えることが可能

Backtrack

- Target backtrack window
 - DBクラスタをBacktrackできる時間
- Actual backtrack window
 - DBクラスタをBacktrackできる実際の時間
 - target Backtrack windowよりワークロードにより短くなる可能性がある
- Auroraは常に一貫性のある時間にBacktrackする
 - Backtrackの時間を指定すると、Auroraは自動的に最も近い一貫性のある時間を選択するため、完了したBacktrackが指定した時刻と正確に一致しない可能性があります
- 100万changeレコードにつき、\$0.014(東京リージョン)



Backtrack

- CloudWatchメトリクス

- Backtrack Change Records Creation Rate (Count)
 - 5分間隔で取得され、この間隔の間に生成されたchangeレコードの量。コストを試算する際に利用可能
- [Billed] Backtrack Change Records Stored (Count)
 - 実際にDBクラスタが格納しているchangeレコードの量
- Backtrack Window Actual (Minutes)
 - Target backtrack windowとActual backtrack windowの間に違いがあるかどうかを表示
 - 例えば、Target backtrack windowが2時間(120分)で、このメトリクスでActual backtrack windowが100分とを示している場合、実際のbacktrack windowはターゲットよりも小さくなる
- Backtrack Window Alert (Count)
 - 特定の期間、Actual backtrack windowがTarget backtrack windowよりも少なくなったかを表す

OOM Avoidance (1.19以降)

- システムメモリを監視し、データベースが様々なコンポーネントで使用するメモリを追跡して、システムのメモリが不足した際にOut of Memoryにならないようにベストエフォートでパラメータグループで指定したアクションを実行してOOMを可能な限り回避
 - T2インスタンスはデフォルトで有効
 - `aurora_oom_response`パラメータで設定
- `print`: 大量のメモリを使用するクエリの実行出力
- `tune`: 内部テーブルキャッシュをチューニングして、メモリをシステムに開放
- `decline`: インスタンスのメモリが不足すると新しいクエリを拒否
- `kill_query`: インスタンスのメモリが閾値より下がるまでメモリ消費の大きい順にクエリを強制終了する。DDLは強制終了されない
- `print, tune`: `print`と`tune`のアクションを実行
- `tune, decline, kill_query`: `tune`、`decline`、`kill_query`のアクションを実行

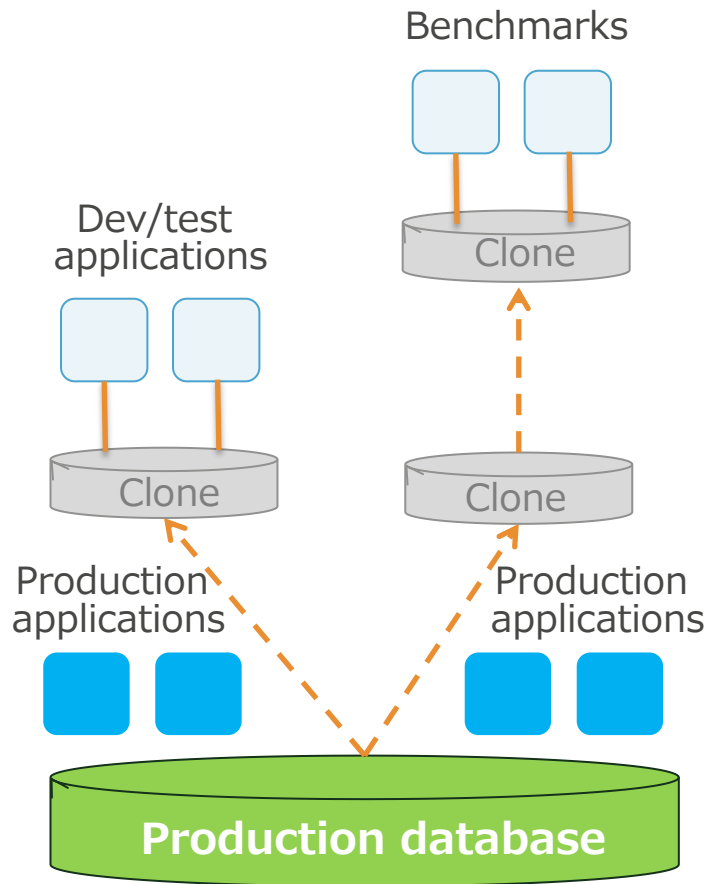
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.AuroraMySQLOOM

Database cloning

- ストレージコストを増やすことなくデータベースのコピーを作成
 - データをコピーするわけではないため、クローンの作成はほぼ即座に完了
 - データのコピーはオリジナルボリュームとコピー先のボリュームのデータが異なる場合の書き込み時のみ発生

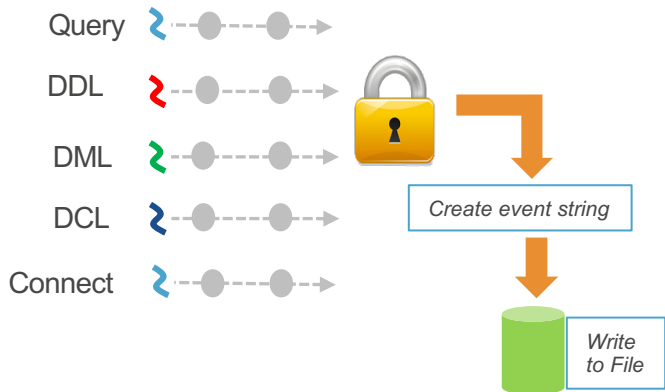
ユースケース

- プロダクションデータを使用したテスト
- データベースの再構成
- プロダクションシステムに影響を及ぼさずに分析目的で特定の時点でのスナップショットを保存

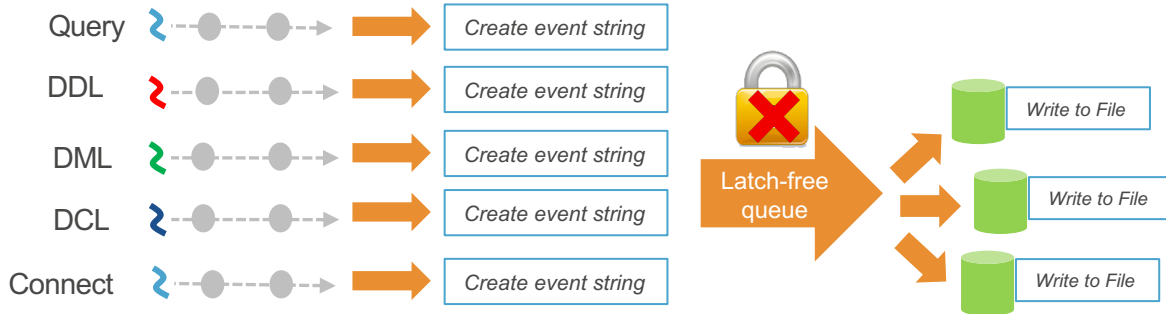


Advanced Auditing

MariaDB server_audit plugin



Aurora native audit support



- We can sustain over 500K events/sec

	MySQL 5.7	Aurora	
Audit Off	95K	615K	6.47x
Audit On	33K	525K	15.9x

Sysbench Select-only Workload on 8xlarge Instance

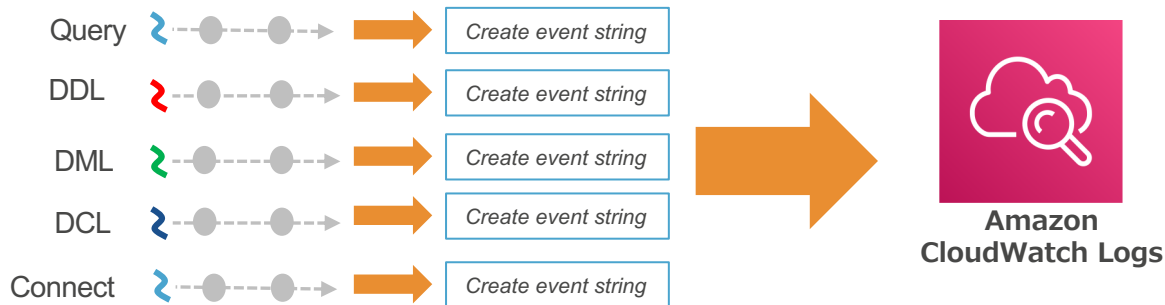
Advanced Auditing

- 注意点

- Aurora advanced auditingのタイムスタンプはUnix timeフォーマットで記録される
- イベントは各インスタンス毎に4つのログファイルに記録され、ログはシーケンシャルには並んでいない。必要に応じてファイルの結合やタイムスタンプやquery_idを使用してソートを行う
 - Unixをお使いの場合の例 `cat audit.log.* | sort -t"," -k1,1 -k6,6`
- ファイル数はDBインスタンスサイズに応じて変化しますファイルのローテーションは100MB毎に行われ、閾値の変更は不可



Amazon Aurora AuditログをAmazon CloudWatch Logsで管理



Amazon AuroraのAudit logをCloudWatch Logsに直接転送可能

Filter Patternを設定しMetricsを作成することが可能なため、CloudWatchを利用したアラートの発報も可能

事前にAmazon Auroraに対して、IAM Roleの設定が必要

The screenshot shows the 'Create Alarm' console in the Amazon CloudWatch interface. The '1. Select Metric' step is completed, and the '2. Define Alarm' step is active. The 'Metric Name' is set to 'FailedConnections'. The 'Update Graph' section shows a line graph with a single data point at approximately 19:00. The 'Time Range' is set to 'Relative' with a 'From' time of '12:06 hours ago' and a 'To' time of '0 hours ago'. The 'Limits' are set to 'Auto' for both 'Min' and 'Max'. The 'Create Alarm' button is highlighted in blue.

The screenshot shows the 'Define Logs Metric Filter' console. The 'Filter for Log Group' is set to '/aws/rds/cluster/pabertestlog2-1/audit'. The 'Filter Pattern' is set to '*FAILED_CONNECT*'. The 'Select Log Data to Test' section shows a list of log entries with a 'Test Pattern' button. The 'Results' section is empty, and the 'Assign Metric' button is highlighted in blue.

Load Data From S3



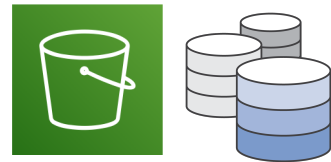
- S3バケットに保存されたデータを直接Auroraにインポート可能
 - テキスト形式(**LOAD DATA FROM S3**)・XML形式(**LOAD XML FROM S3**)
 - LOAD DATA INFILEとほぼ同様のオプションをサポート (圧縮形式のデータは現在未サポート)
 - Manifestによる一括ロードにも対応 (Version 1.11以降)

```
<row>
  <column1>value1</column1>
  <column2>value2</column2>
</row>
```

```
<row>
  <field name="column1">value1</field>
  <field name="column2">value2</field>
</row>
```

```
<row column1="value1" column2="value2" />
<row column1="value1" column2="value2" />
```

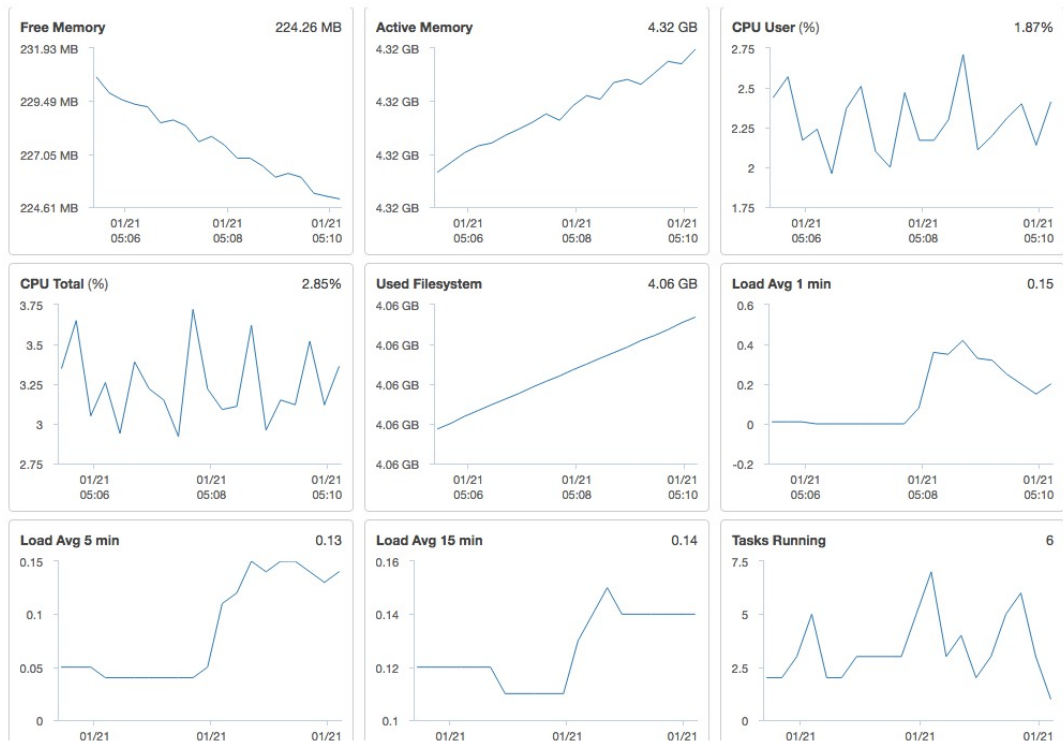
Export Data into S3



- S3バケットにデータを直接Auroraエクスポート可能
 - LOAD DATA FROM S3で利用できるManifestファイルを生成可能
 - 1ファイルは最大6GBずつ分割される
 - 25GBを超えるようなデータをexportする場合は、複数のSQLに分割してexportする領域をずらして実行する事を推薦

```
SELECT * FROM employees INTO OUTFILE S3 's3://bucket_name/prefix'  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '¥n'  
MANIFEST ON  
OVERWRITE ON;
```

拡張モニタリング



50+ system/OS metrics | sorted process list view | 1–60 sec granularity
alarms on specific metrics | egress to Amazon CloudWatch Logs | integration with third-party tools

重要なシステム/OSメトリクスに対応

CPU Utilization

User
System
Wait
IRQ
Idle

Processes

Num processes
Num interruptible
Num non-interruptible
Num zombie

Memory

MemTotal
MemFree
Buffers
Cached
SwapCached
Active
Inactive
SwapTotal
SwapFree
Dirty
Writeback
Mapped
Slab

Device IO

TPS
Blk_read
Blk_wrtn
read_kb
read_IOs
read_size
write_kb
write_IOs
write_size
avg_rw_size
avg_queue_len

Network

Rx per declared *ethn*
Tx per declared *ethn*

Process List

Process ID
Process name
VSS
Res
Mem %
consumed
CPU % used
CPU time
Parent ID

File System

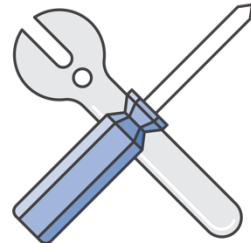
Free
capacity
Used
% Used

拡張モニタリング



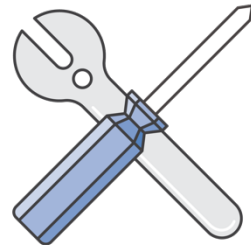
- CloudWatch Logsにメトリクスを送信出来る
- CloudWtch logs->Lambda->Elasticsearch Service連携も容易
 - Kibanaを使って可視化も可能 (Kibanaはインストール済)
 - アプリケーションやクエリの種類に応じたメトリクスも取得すれば、アプリケーション・DBサーバメトリクス・クエリのパフォーマンスを一箇所で閲覧可能

Lab mode



- 今後提供予定の機能を試すことが出来る
 - DBパラメータグループ `aurora_lab_mode` 変数で設定可能
 - 開発中の機能なので本番適用ではなく検証目的でお使い下さい
 - GAクオリティですが、全てのワークロードで性能が発揮出来るか検証を行っている段階です
 - フィードバックをお待ちしています！
- 現在ご提供中
 - **Lock compression**
 - ロックマネージャーが利用するメモリを最大66%削減
 - OOMを起こさず、更に多くの行ロックを同時に取得することが可能に
 - **Fast DDL**
 - nullableカラムをテーブルの最後に追加する場合にデータ件数によらず高速に変更が行なえます
 - <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/AuroraMySQL.Managing.html#AuroraMySQL.Managing.FastDDL>

Lab mode



• 現在ご提供中

– Hash Join













- equijoinクエリを高速化するためにAuroraのオプティマイザは、hash joinの使用を自動的に決定
- クエリプランで強制的に使用することも可能
- 内部テストでは、R3.8xlargeインスタンスでコールド・バッファー・プールを利用したケースで最大8.2xの性能向上が見られた

– Batched Scan

- インメモリークエリで最大1.8xの性能向上が見られた

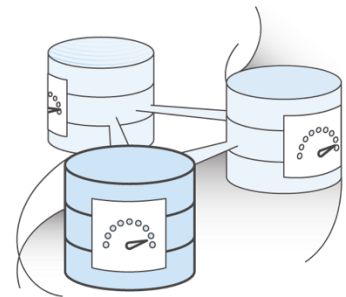
<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/AuroraMySQL.Updates.LabModeFeatures.html>

Amazon Auroraマイグレーションオプション

Source database	From where	Recommended option
 	RDS	コンソールベースの自動スナップショットインテグレーションとbinlogを用いたレプリケーション
   	EC2, オンプレミス	S3を用いたバイナリスナップショットインテグレーションとbinlogを用いたレプリケーション
     	EC2, オンプレミス, RDS	SCTを用いたスキーマ変換とDMSを用いたデータ変換

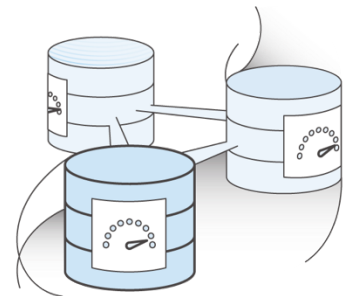
まとめ

Amazon Aurora



- クラウド時代にAmazonが再設計したRDBMS
 - MySQL5.6/5.7と互換があり既存の資産を活かしやすい
- 高いクエリ実行並列度・データサイズが大きい環境で性能を発揮
 - Amazon Auroraはコネクション数やテーブル数が多い環境で優位性を発揮
- 高可用性・高速なフェイルオーバー・PITRを実現するための多くのチャレンジ
 - Log Structured Storage
 - SOA

参考資料



- 更新情報について

- <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/AuroraMySQL.Updates.html>

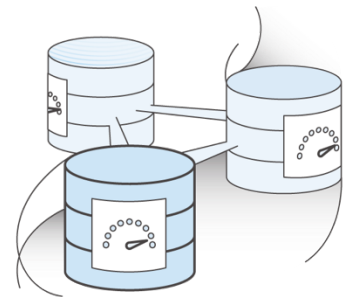
- Amazon Aurora MySQL 2.0 Database Engine Updates
 - Amazon Aurora MySQL 1.1 Database Engine Updates
 - MySQL Bugs Fixed by Amazon Aurora MySQL Database Engine Updates

- Lab. Modeについて

- <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/AuroraMySQL.Updates.LabModeFeatures.html>

参考資料

- Amazon Aurora MySQL詳細
 - <https://aws-ref.s3.amazonaws.com/aurora/Amazon+Aurora.pdf>
- Amazon Aurora SIGMOD論文(英語)
 - <http://www.allthingsdistributed.com/files/p1041-verbitski.pdf>
- Amazon Auroraストレージエンジン
 - <http://aws.typepad.com/sajp/2017/02/introducing-the-aurora-storage-engine.html>
- Amazon Auroraストレージのquorumに関する実装について
 - <https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>
 - <https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-reads-and-mutating-state/>
 - <https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-reducing-costs-using-quorum-sets/>
 - <https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-membership/>
- AWS Database Blog (Amazon Aurora)
 - <https://aws.amazon.com/blogs/database/category/aurora/>



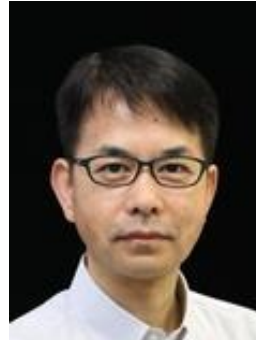
Q&A

- 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。

5月のBlack Belt Online Seminar 配信予定

<https://amzn.to/JPWebinar>

- 5/14 (火) 12:00-13:00 Amazon API Gateway
- 5/15 (水) 18:00-19:00 Let's dive deep into AWS Lambda Part3
- 5/21 (火) 12:00-13:00 Amazon Simple Email Service (SES)
- 5/22 (水) 18:00-19:00 AWS Step Functions
- 5/28 (火) 12:00-13:00 Let's dive deep into AWS Lambda Part4
- 5/29 (水) 18:00-19:00 AWSトレーニング



AWS の日本語資料の場所「AWS 資料」で検索

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 »](#)

[サービス別資料 »](#)

<https://amzn.to/JPArchive>

[申込受付中！] AWS Innovate オンラインカンファレンス



INNOVATE
ONLINE CONFERENCE

2019年4月8日(月) ~ 5月7日(火) 開催

今すぐ無料参加申込み >>

<https://amzn.to/AWSInnovateJP>

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>

