



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 Compatible Edition ユースケース毎のスケーリング手法

サービスカットシリーズ

Solutions Architect 齋藤 航
2020/09/29

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



過去資料
<https://amzn.to/JPArchive>



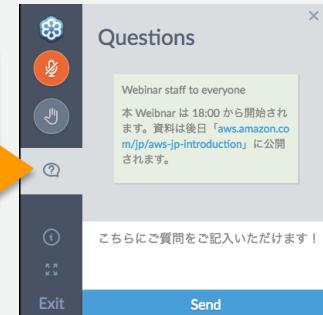
AWS Black Belt Online Seminar とは

「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、Amazon 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9月29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います。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.

自己紹介

齋藤 航

所属

- ・ アマゾン ウェブ サービス ジャパン 株式会社
Specialist Solution Architect, Databases



経歴

- ・ 大手ポータルサイトで
オンプレミス環境の DBA を経験

好きな AWS サービス

- ・ Amazon RDS
- ・ Amazon S3

本資料の前提知識とゴール

前提知識

- Amazon Aurora MySQL の基礎知識
 - Amazon Aurora の網羅的な紹介は今回のセミナーでは行いません。
2019/04 開催のセミナーをご確認ください
<https://aws.amazon.com/jp/blogs/news/webinar-bb-amazon-aurora-mysql-2019/>

ゴール

- Amazon Aurora MySQL をスケールさせるための、アーキテクチャパターンのご紹介
- スケーリングの選択肢である Aurora レプリカ, Aurora Serverless, グローバルデータベースなどの機能を紹介し、それらがフィットするユースケースをご理解いただく



アジェンダ

- Amazon Aurora 概要
- Latest updates
- Amazon Aurora MySQL をスケールさせる
- まとめ

アジェンダ

- Amazon Aurora 概要
- Latest updates
- Amazon Aurora MySQL をスケールさせる
- まとめ

Amazon Aurora

クラウド向けに再設計された MySQL, PostgreSQL と互換性のある RDBMS
コマーシャルデータベースの性能と可用性を 1/10 のコストで実現

優れた性能と拡張性



標準的なMySQLと比べて5倍、
標準的なPostgreSQLと比べて
3倍のスループットを実現;
リードレプリカを最大 15 個
追加してスケールアウト可能

高可用性と耐久性



耐障害性、自己修復機能を
兼ね備えたストレージ;
3 つのAZにわたり、6 個の
コピーを保持; Amazon S3
への継続的なバックアップ

高い安全性



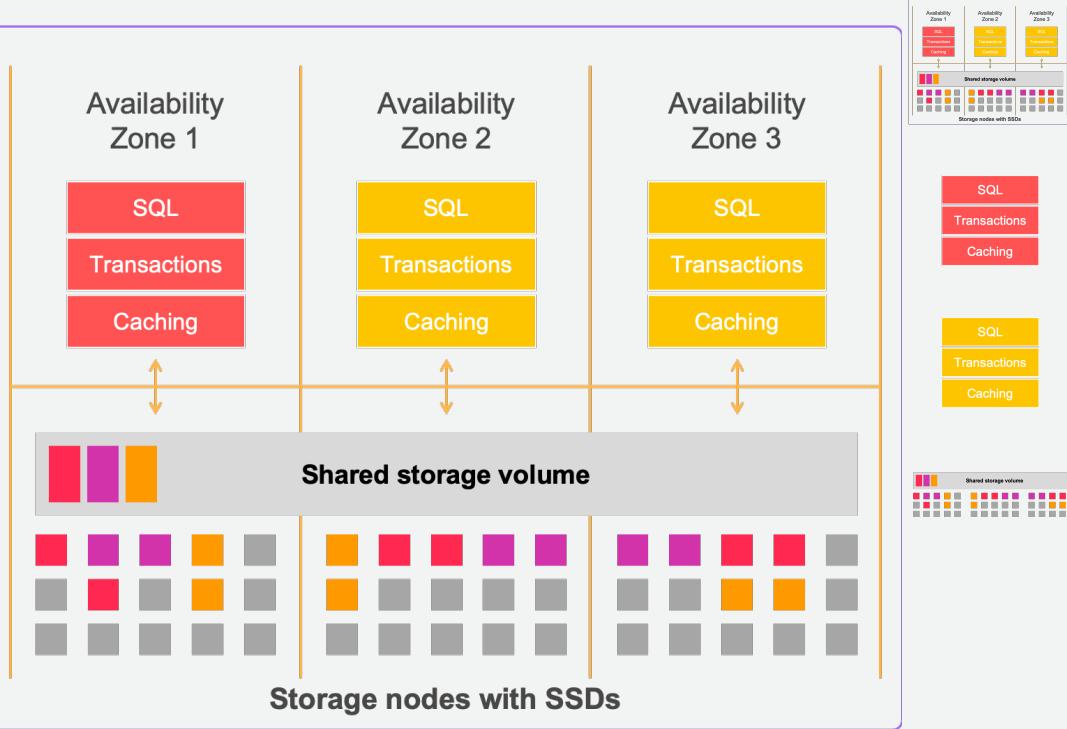
ネットワーク分離、
保管時/通信の暗号化

フルマネージド



ハードウェアのプロビジョニ
ング、ソフトウェアのパッチ
適用、セットアップ、構成、
バックアップといった
管理タスクからの解放

Aurora を構成するコンポーネント

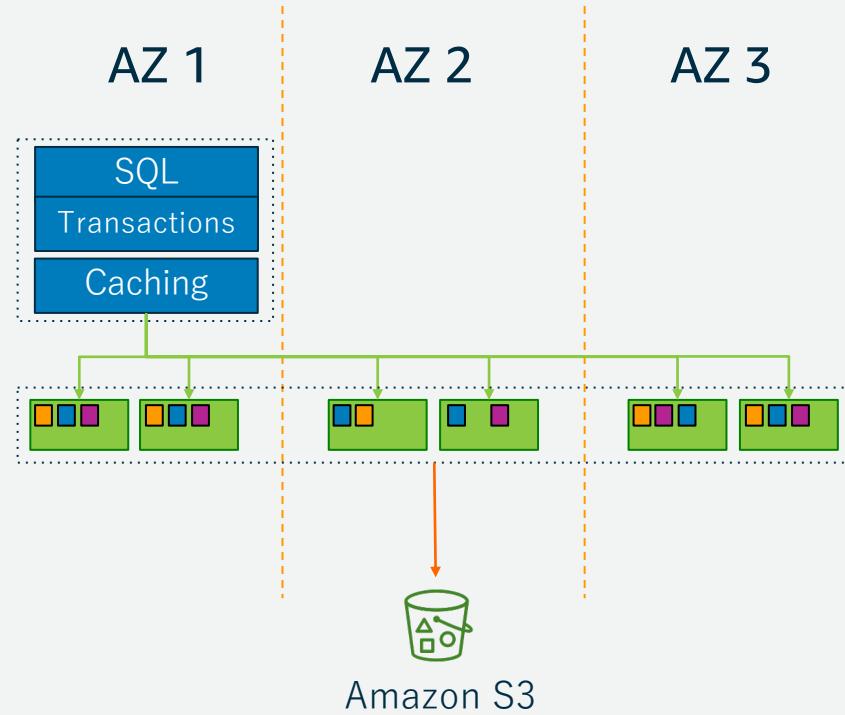


- **Amazon Aurora DB クラスター**
 - Amazon Aurora の管理単位
 - プライマリインスタンス、レプリカ、クラスター ボリュームの総称
- **プライマリインスタンス (Writer)**
 - 読み込み、書き込みを行うインスタンス
- **レプリカ (Reader)**
 - 読み込みをスケールアウトさせるレプリカ (15台まで作成可能)
- **クラスター ボリューム (ストレージ)**
 - 3つの AZ 間でレプリケートされる仮想ボリューム
 - プライマリインスタンスもレプリカも同じクラスター ボリュームを利用
- **エンドポイント**
 - Aurora の接続先を示す URL

詳細 : http://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/Aurora.Overview.html

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**
 - 最大 64TiB までシームレスに自動でスケールアップ
 - Aurora MySQL 1.23.x, 2.09.x からは最大 128 TiB に
 - 実際に使った分だけ課金
- **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
 - クオーラムシステムの採用
 - 自動修復、ホットスポット管理
 - 繙続的に S3 へ増分バックアップ



詳細 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html

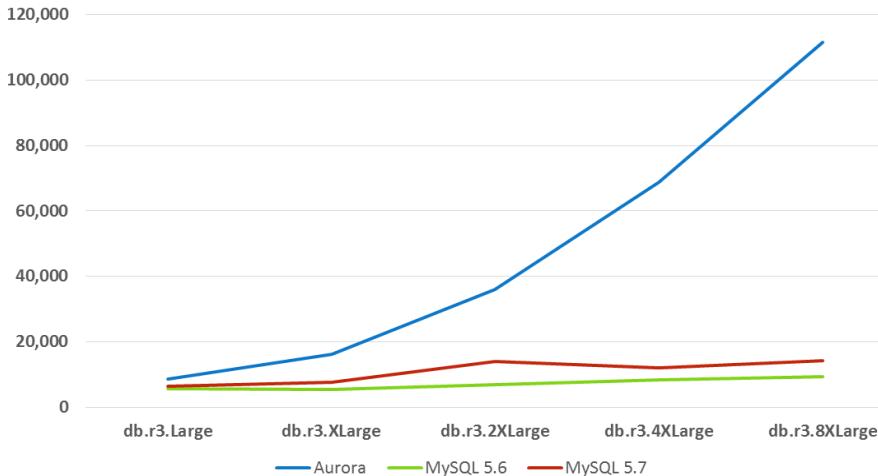
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



高いパフォーマンス：インスタンスサイズによるスケール

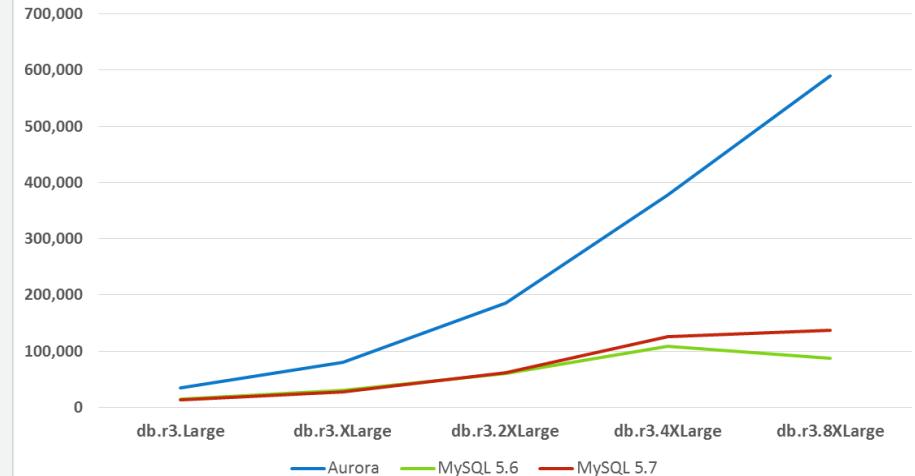
WRITE PERFORMANCE

Average writes per second



READ PERFORMANCE

Average reads per second



Aurora MySQL

MySQL 5.6

MySQL 5.7

Aurora は Read/Write パフォーマンス共に、
インスタンスサイズに比例してスループットがスケールする

アジェンダ

- Amazon Aurora 概要
- Latest updates
- Amazon Aurora MySQL のスケーリング
- まとめ

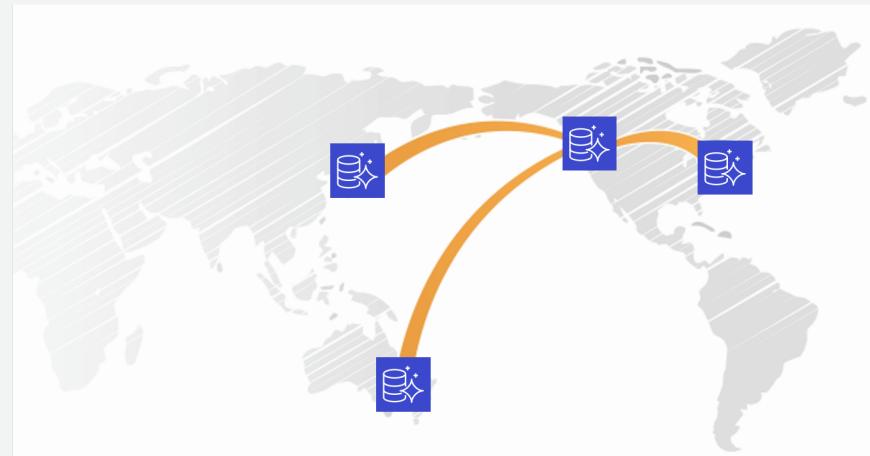
2019年4月以降の Aurora MySQL 関連のアップデート

1. [Amazon RDS 拡張モニタリングに新しいストレージメトリクスおよびホストメトリクスが追加](#) - 2019/04/12
2. [Amazon Aurora および Amazon RDS で MySQL 5.7 データベースからの移行を高速化](#) - 2019/04/18
3. [Amazon Aurora Serverless がスナップショットの共有およびクロスリージョンコピーをサポート開始](#) - 2019/04/18
4. [Amazon Aurora Serverless が 1 ユニットのキャパシティーと新しいスケーリングオプションのサポートを開始](#) - 2019/04/29
5. [Amazon RDS の推奨事項で Amazon Aurora 向けのベストプラクティスガイダンスを提供開始](#) - 2019/05/22
6. [Amazon Aurora Serverless MySQL 5.6 が Data API のサポートを開始](#) - 2019/05/30
7. [Amazon Aurora が Aurora ストレージに対するコスト割り当てをサポート](#) - 2019/10/23
8. [Aurora Global Database が Amazon Aurora MySQL 5.7 でサポートされます](#) - 2019/11/26
9. [Amazon Aurora がデータベースから機械学習を直接サポート](#) - 2019/11/26
10. [Aurora Global Database が複数のセカンダリリージョンをサポート](#) - 2019/11/26
11. [Aurora で Global Database へのインプレース変換が可能に](#) - 2019/11/26
12. [Amazon Aurora MySQL 5.7 がゼロダウンタイムのパッチ修正に対応](#) - 2019/11/26
13. [Amazon Aurora がリードレプリカで READ COMMITTED の分離レベルをサポート](#) - 2020/01/16
14. [Amazon Relational Database Service \(RDS\) スナップショットの S3 へのエクスポートを発表](#) - 2020/01/23
15. [Amazon RDS Data API が、AWS PrivateLink のサポートを開始](#) - 2020/02/06
16. [Amazon RDS Performance Insights が MySQL 互換の Amazon Aurora で SQL レベルのメトリクスのサポートを開始](#) - 2020/05/05
17. [Amazon Aurora Multi-Master が 8 の AWS リージョンで利用可能に](#) - 2020/05/07
18. [データベースアクティビティストリームが Aurora \(MySQL 互換\) で利用可能に](#) - 2020/06/03
19. [Amazon Aurora スナップショットが AWS Backup 経由で管理可能に](#) - 2020/06/10
20. [Amazon Aurora Global Database がリードレプリカ書き込み転送のサポートを開始](#) - 2020/06/19
21. [Amazon Aurora Serverless \(MySQL 5.7 互換\) の発表](#) - 2020/06/24
22. [Amazon RDS Proxy が一般提供開始](#) - 2020/06/30
23. [Aurora Parallel Query のリージョンの拡張と MySQL 5.7 互換の発表](#) - 2020/09/17
24. [Amazon Aurora Increases Maximum Storage Size to 128TB](#) - 2020/09/24



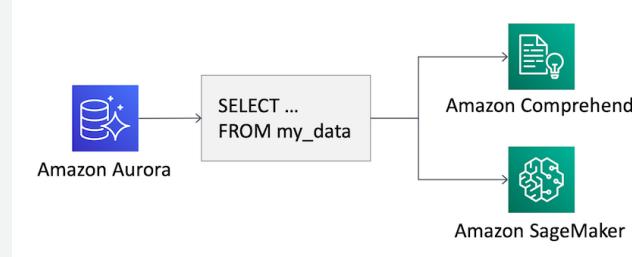
Aurora Global Database に関する複数のアップデート

- Global Database へのインプレース変換
- Aurora MySQL 5.7 対応
- 複数のセカンダリリージョンに対応
(最大5つのリージョン)
- Performance Insights 対応
- 書き込み転送機能のサポート



Amazon Aurora ML integration

- Amazon Aurora 内から、Amazon SageMaker と Amazon Comprehend を呼び出せるように
 - Amazon Aurora 内に保存されたデータに対して、データを移動するパイプラインを作成することなく、Amazon SageMaker や Amazon Comprehend にデータを投入して、結果を取得可能
 - Stored Function として実行するため、通常の SQL を利用可能
- ユースケース
 - 課金ログから不正トランザクション検出
 - Blog などについてのコメントに対してセンチメント分析
 - ユーザ情報などからポテンシャルカスタマーの抽出



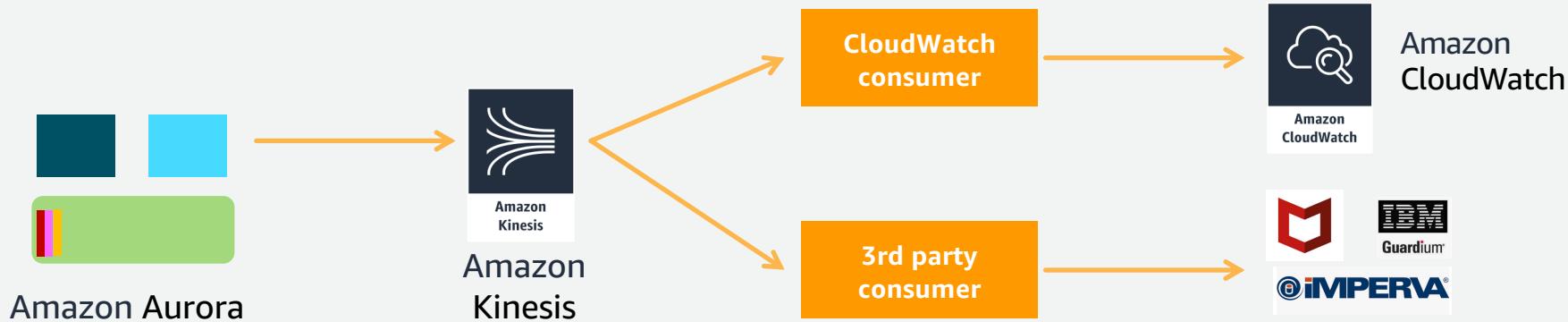
詳細：<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-ml.html>

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



Database Activity Stream (DAS)

データベースを保護し、コンプライアンス/規制要件を満たすためのモニタリング



- ・ 暗号化済み監査ログを CloudWatch Logs に送信して、DB クラスターのアクティビティを継続的に監視
- ・ アーカイブ用にS3にエクスポート; Amazon Athena を使用したログ分析; Amazon QuickSight でログを可視
- ・ PostgreSQL の場合、以下のパートナー製品が本機能を利用した監査に対応
 - ・ SecureSphere Database Audit and Protection (Imperva)
 - ・ Data Center Security Suite (McAfee)
 - ・ Infosphere Guardium (IBM)

DAS 以外に DB 標準の監査機能も利用可能

詳細：https://docs.aws.amazon.com/ja_ip/AmazonRDS/latest/AuroraUserGuide/DBActivityStreams.html

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



Amazon RDS Proxy

Amazon RDS 向けの高可用性フルマネージド型データベースプロキシ
アプリケーションのスケーラビリティや、データベース障害に対する回復力と
安全性の向上を実現



データベース接続をプールおよび共有する事でアプリケーションのスケーリングを改善



アプリケーションの可用性を高め、データベースのフェイルオーバー時間を短縮

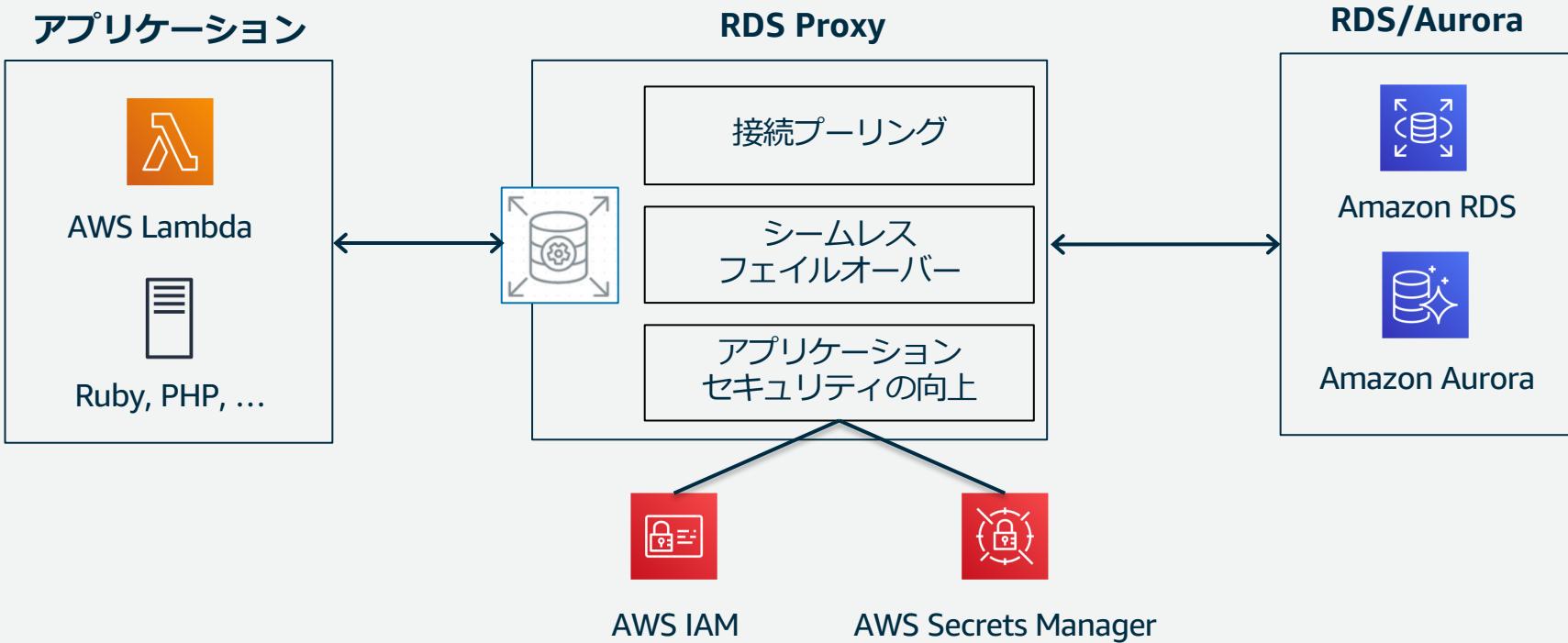


データベースアクセス制御で、アプリケーションデータのセキュリティを管理



フルマネージドデータベースプロキシ、データベースとの完全な互換性

Amazon RDS Proxy 全体像



詳細：<https://aws.amazon.com/jp/rds/proxy/>

最大ストレージサイズが 128TiB に拡張

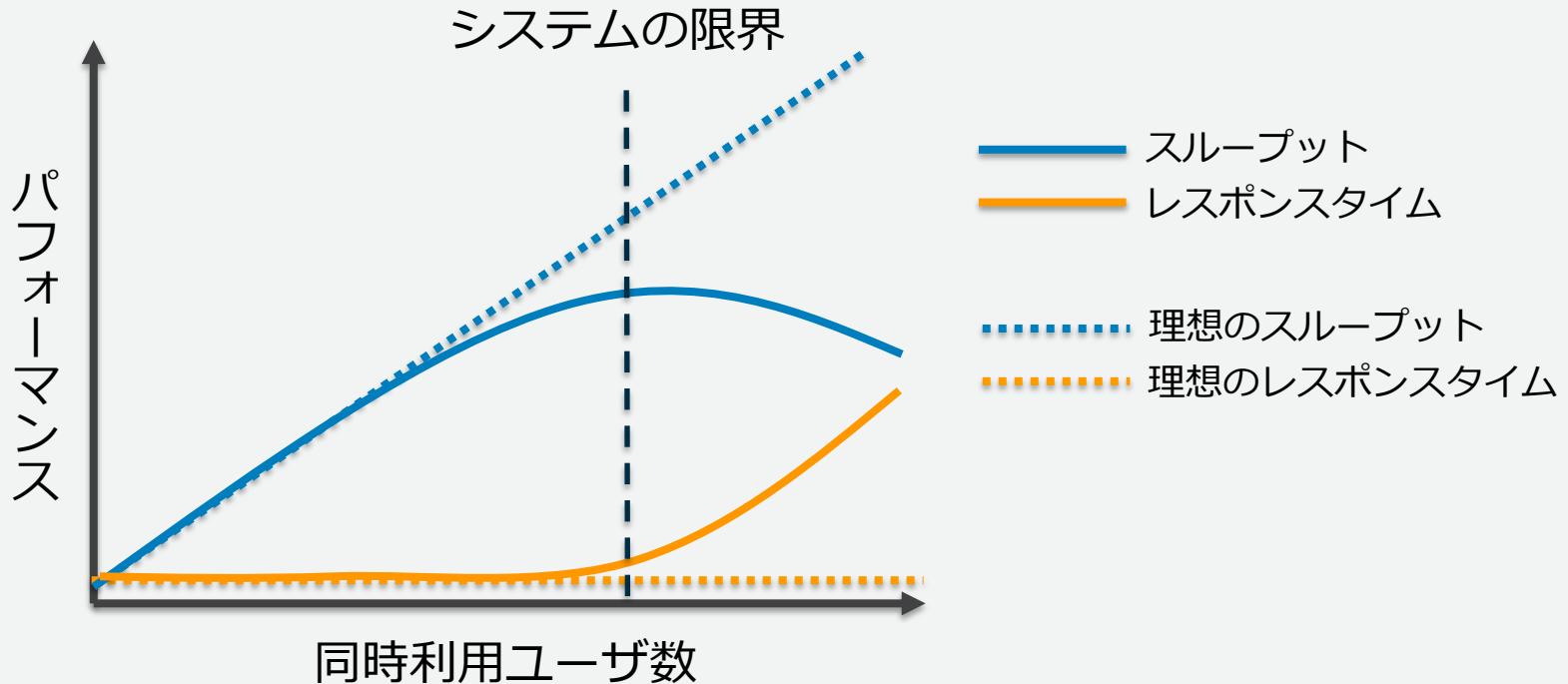
- 最大ストレージサイズが 128 TiB に拡張
 - Aurora MySQL 1.23.x および 2.09.x より対応
<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Updates.1230.html>
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Updates.2090.html
- ストレージサイズの上限に近づくと AWS マネジメントコンソールの RDS Event と MySQL ログ にアラートが表示される
- 注意点
 - 1テーブルの上限は最大 64TiB
 - db.t2/db.t3 インスタンスでは未サポート

アジェンダ

- Amazon Aurora 概要
- Latest updates
- Amazon Aurora MySQL をスケールさせる
- まとめ

“スケールする”システムとは？

- 利用負荷 や 規模の増大 に対応できるシステム・仕組み



なぜ、データベースのスケーリングを考えるのか

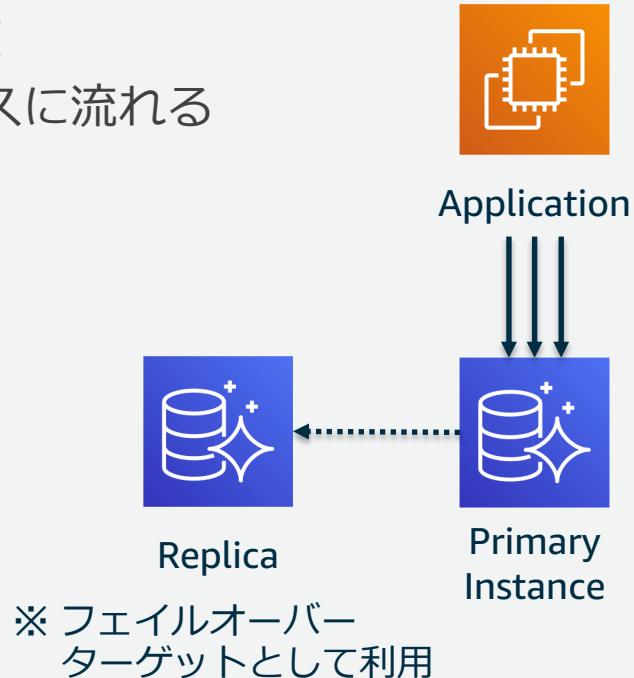
- サービスの成長に伴い、ユーザが増えていく
 - データ量が増え、キャッシュメモリに乗らなくなる
 - 新機能の追加による、当初想定していない負荷がかかる
- 多くのアプリケーションは状態を持たず、スケールさせやすい一方、データベースはボトルネックになりやすい
- (特にデータベースでは) 負荷に耐えられないと、**ビジネスの機会損失に直結する**

なぜ、データベースのスケーリングを考えるのか

- クラウドでは非常に簡単にコンピューティングの調達やスケールアップが可能。
相対的に、アーキテクチャを考えることの重要性は増している
- システム設計の時点で、「どれくらいの規模になりうるか」、「どう対応するか」を想定することで、後々の対応が楽になる
 - 一般に、利用規模とあらゆる運用コストは比例する
- 開発者でデータ量・利用規模のコントロールが難しく、
いつ、スケーリングが必要になるか分からぬ場合もある
 - SaaS アプリケーション等

サービス初期はシンプルな構成から始める

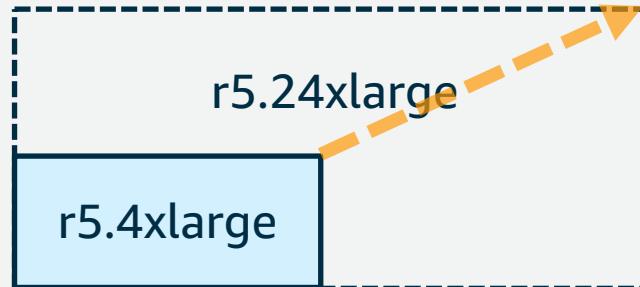
- ・ シングルプライマリ構成
 - ・ 全てのテーブルが1インスタンスに存在
 - ・ 全てのクエリがプライマリインスタンスに流れる
- ・ 初期から過度に作り込み過ぎない。
シンプルな設計から検討する



二つの戦略：スケールアップとスケールアウト

スケールアップ戦略

- インスタンスの性能を上げることで、**単体での** 性能改善を図る
- シンプルで、アーキテクチャに変更がない
 - 管理対象を増やしたくないときや、一時的な負荷増の緩和策に有効
- インスタンスサイズの限界がある

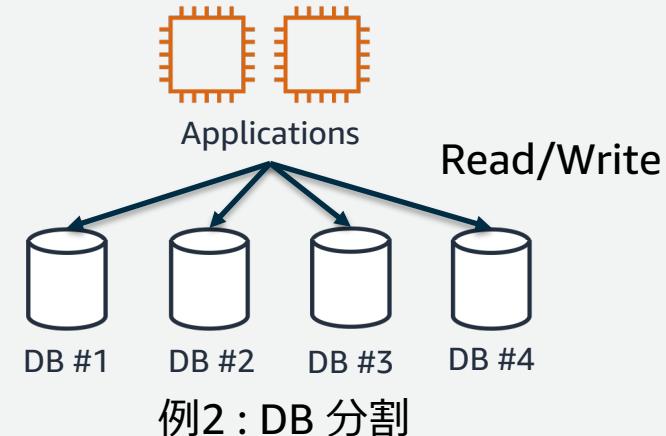
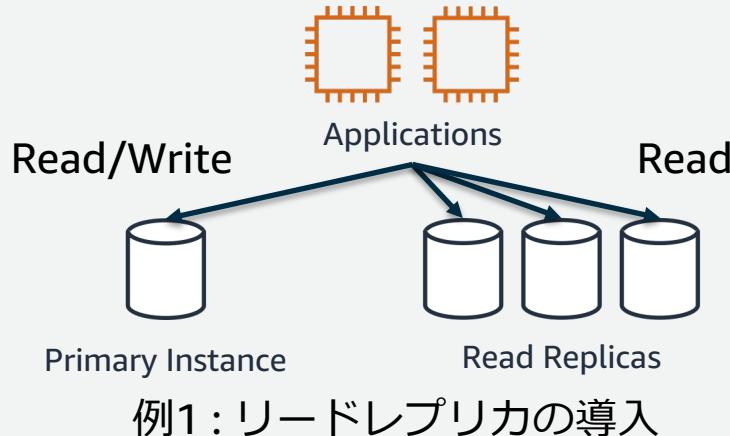


Aurora では、インスタンスサイズの変更により vCPU, メモリ, NW パフォーマンス, ローカルストレージの帯域幅に影響がある

二つの戦略 : スケールアップとスケールアウト

スケールアウト戦略

- ・ インスタンス数を増やして、**全体での** 性能改善を図る
- ・ インスタンスサイズの限界から逃れられるほか、
障害範囲 (Blast radius) を限定できる
- ・ 複雑性は増す・アプリケーションの改修が必要



スケールアップ・スケールアウトの前に考えること

解決したい課題（ボトルネック）は何かを特定する

- ロック待ち, 実行計画, メモリ不足, CPU, NW帯域, レイテンシ etc.
- スケールアップ（アウト）でボトルネックを解決できるか？



データベースレイヤでのチューニングの検討

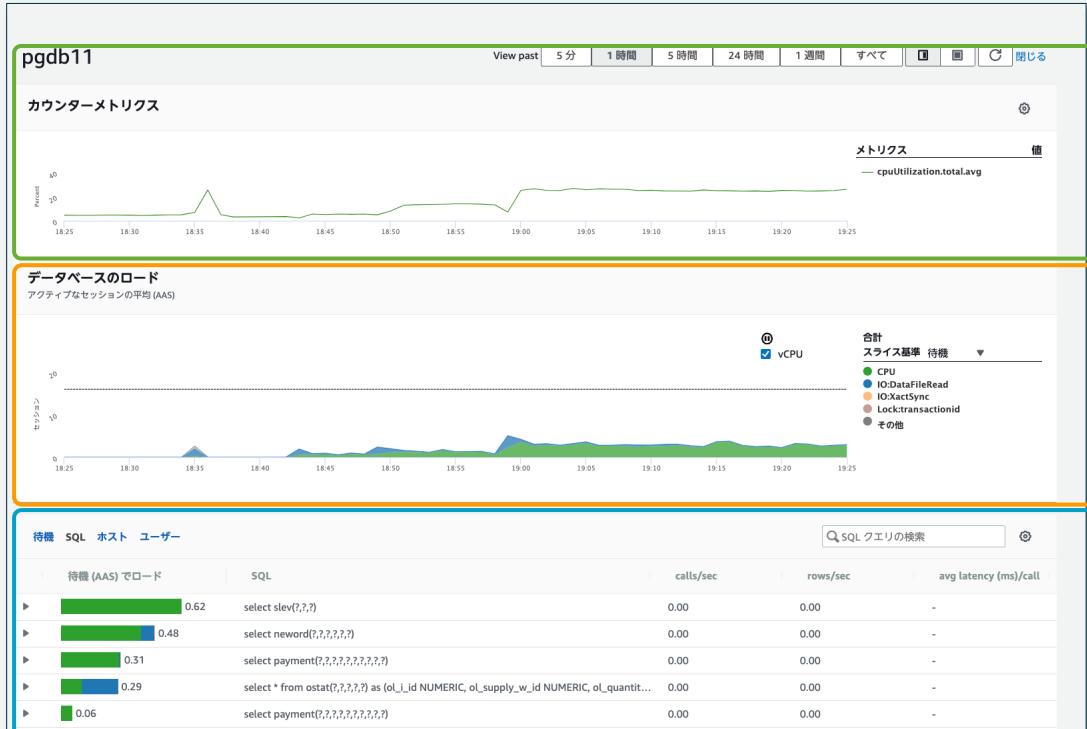
- インデックスを貼る・ヒント句の利用などによる実行計画の制御
- クエリの書き換え
- パラメータチューニング



スケールアップ・スケールアウト

[参考] Performance Insights によるボトルネックの特定

- 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



カウンターメトリクス

- OSのリソース情報(CPU、Memoryなど)
- DBの統計情報(セッション数、トランザクション数など)

データベースのロード

- 平均アクティブセッション数(AAS)
- CPU時間と待機イベント内訳

ボトルネックの分析軸

- ボトルネックの原因の待機
- ボトルネックとなっているSQL
- 性能影響の高いホスト、ユーザー

詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/USER_PerfInsights.html

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



Aurora MySQL のスケーリング

Case #1

読み込みパフォーマンスのスケーリング

Case #2

書き込みパフォーマンスのスケーリング

Case #3

インフラストラクチャの地理的スケーリング

Case #4

コスト効率の良いスケーリングをしたい

Aurora MySQL のスケーリング

Case #1

読み込みパフォーマンスのスケーリング

Case #2

書き込みパフォーマンスのスケーリング

Case #3

インフラストラクチャの地理的スケーリング

Case #4

コスト効率の良いスケーリングをしたい

#1 読み込みパフォーマンスをスケールしたい

解決したい課題

- 参照系クエリが多く、それによる DB への負荷を減らしたい
 - 多くの Web アプリケーションにおいては、更新系クエリよりも参照系クエリの方が多い

解決のパターン

- a. リードレプリカへの参照系クエリのオフロード
- b. キャッシュレイヤの導入

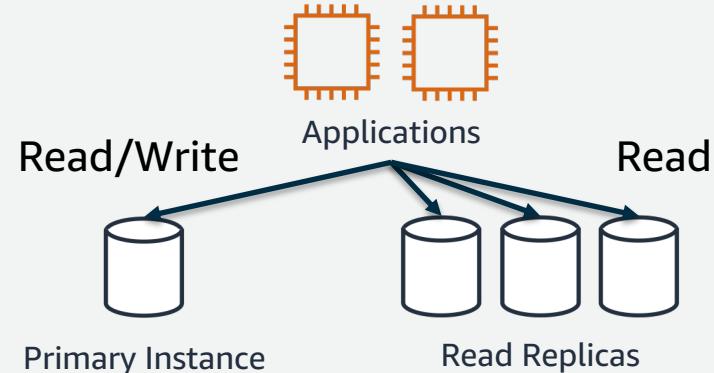
#1-a. リードレプリカの利用

フィットするケース

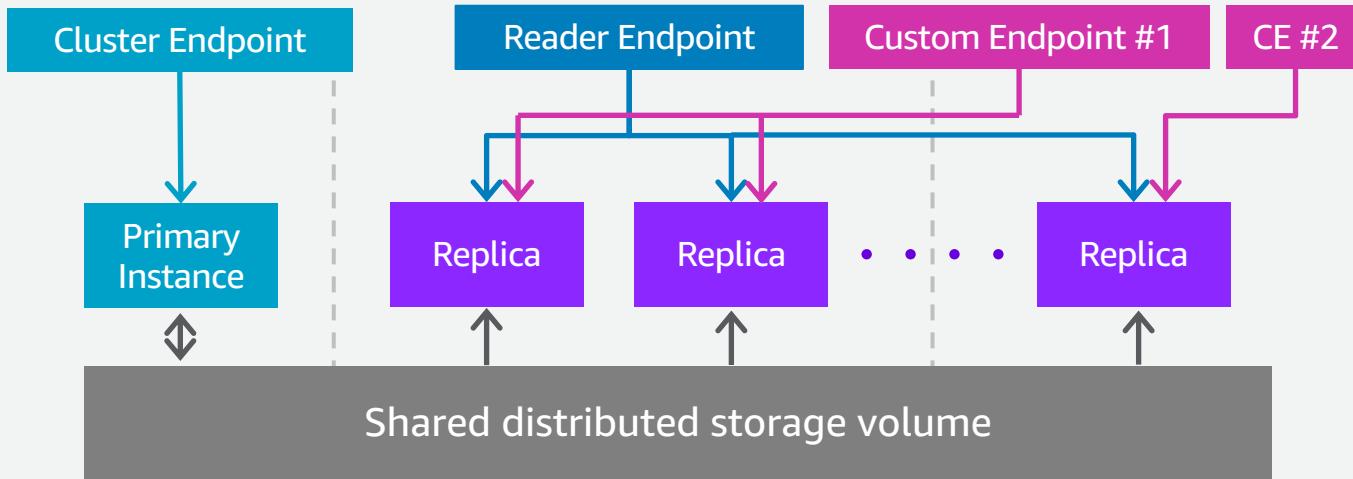
- 参照系クエリの負荷分散をしたい

キーポイント

- レプリカラグの考慮
 - 書き込んだ直後のデータを読む必要があるか
 - ただし、Aurora のレプリカラグは**非常に小さい**
- 参照系クエリの適切な振り分けは **アプリケーションの責務**



Aurora レプリカ



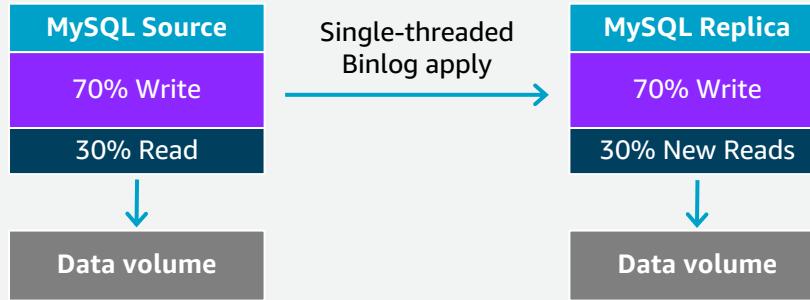
- 3つのアベイラビリティーゾーンで最大15個の昇格可能なレプリカ (Reader)
- フェイルオーバーの優先順位を設定可能
- Reader は Auto Scaling による自動増減が可能
- 低遅延な REDO ログベースの物理レプリケーション
- カスタムエンドポイントによる Reader の使い分け

詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Replication.html
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

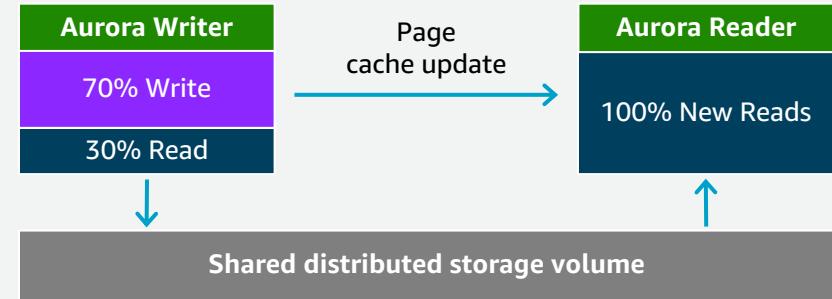


通常の MySQL のレプリカと Aurora レプリカの違い

MySQL read scaling



Amazon Aurora MySQL read scaling



Binlog を用いた 論理レプリケーション

レプリカにソースと 同等の書き込み が発生

独立した ストレージ

ワークロードによっては大きなレプリカラグ

Binlog を用いない 物理レプリケーション

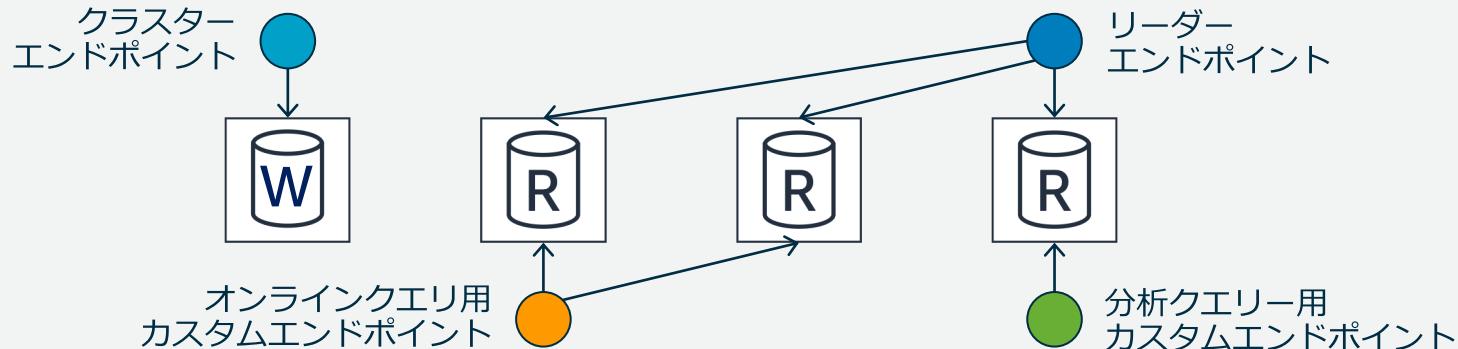
レプリカに書き込みは 発生しない

共有 ストレージ

通常 20~40 ミリ秒のレプリカラグ

Amazon Aurora エンドポイント

- 用途に応じてエンドポイントを使い分ける
 - クラスターエンドポイント：常にプライマリインスタンスを指す
 - リーダーエンドポイント：ラウンドロビンでどれか 1 つのレプリカに接続
レプリカが 1 インスタンスも居ない場合、プライマリにフェイルバック
 - インスタンスエンドポイント：各インスタンスごとの接続先
 - カスタムエンドポイント：ユーザーが定義可能なエンドポイント
- DNS ベースのため、クライアント側のキャッシュを避けることを推奨



詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.Endpoints.html

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

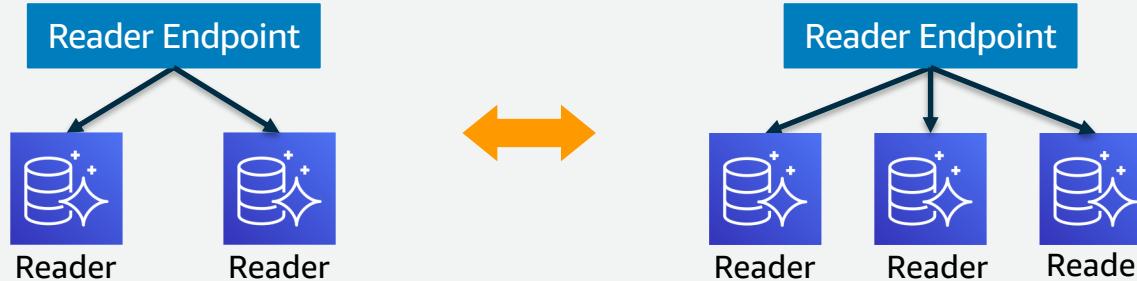


Aurora レプリカでの Amazon Aurora Auto Scaling

- 平均 CPU 使用率・平均接続数に応じて、Aurora Reader を自動増減
 - ワークロードへの追従・余分なコストを支払うリスクを軽減
 - リーダーエンドポイント・カスタムエンドポイントはレプリカの自動的な追加・削除に対応
 - Coldown Period や Min/Max Capacity を設定可能

注意点

- 追加されるのはプライマリインスタンスと同じ DB インスタンスクラス
- 監視間隔、起動までのタイムラグを考慮すると、急激なスパイクへの対応は困難
(予測できるイベントであれば、事前に Reader を追加することで対応)



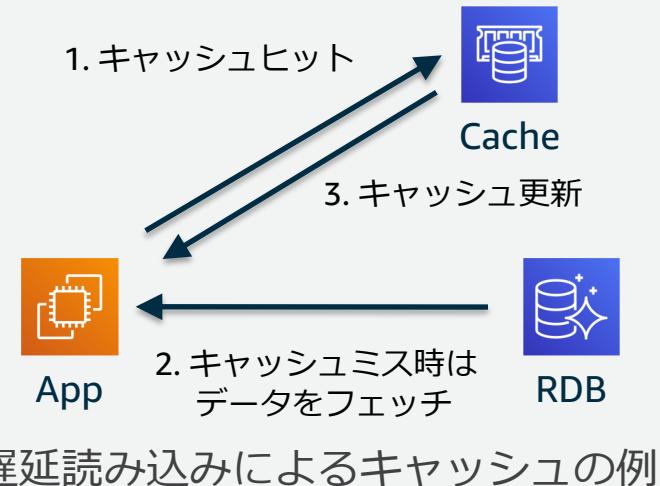
#1-b. [参考] キャッシュレイヤの導入

フィットするケース

- 応答速度を早めたい
- データベースへのアクセスそのものを減らしたい
- セッション管理、リアルタイム集計のオフロード

キーポイント

- キャッシュすべきデータかどうかの見極め
 - 頻繁に利用され、生存期間の長いデータ
- キャッシュ戦略の選択
 - 遅延読み込み or ライトスルー



Aurora MySQL のスケーリング

Case #1

読み込みパフォーマンスのスケーリング

Case #2

書き込みパフォーマンスのスケーリング

Case #3

インフラストラクチャの地理的スケーリング

Case #4

コスト効率の良いスケーリングをしたい

#2 書き込みパフォーマンスをスケールしたい

解決したい課題

- 更新系クエリが非常に多く、インスタンスの性能限界に達している

解決のパターン

- データベース分割による負荷分散
- 用途に合わせた適切なデータベースの利用

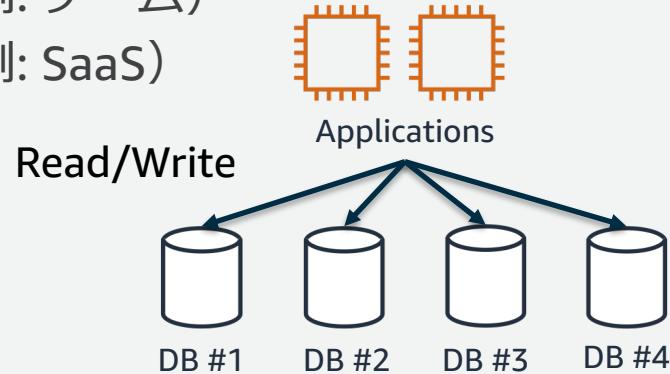
#2-a. データベース分割による負荷分散

フィットするケース

- 参照系・更新系の両方を負荷分散をしたい
 - 更新処理がスパイクするアプリケーション（例: ゲーム）
 - レコードを分離しやすいアプリケーション（例: SaaS）

キーポイント

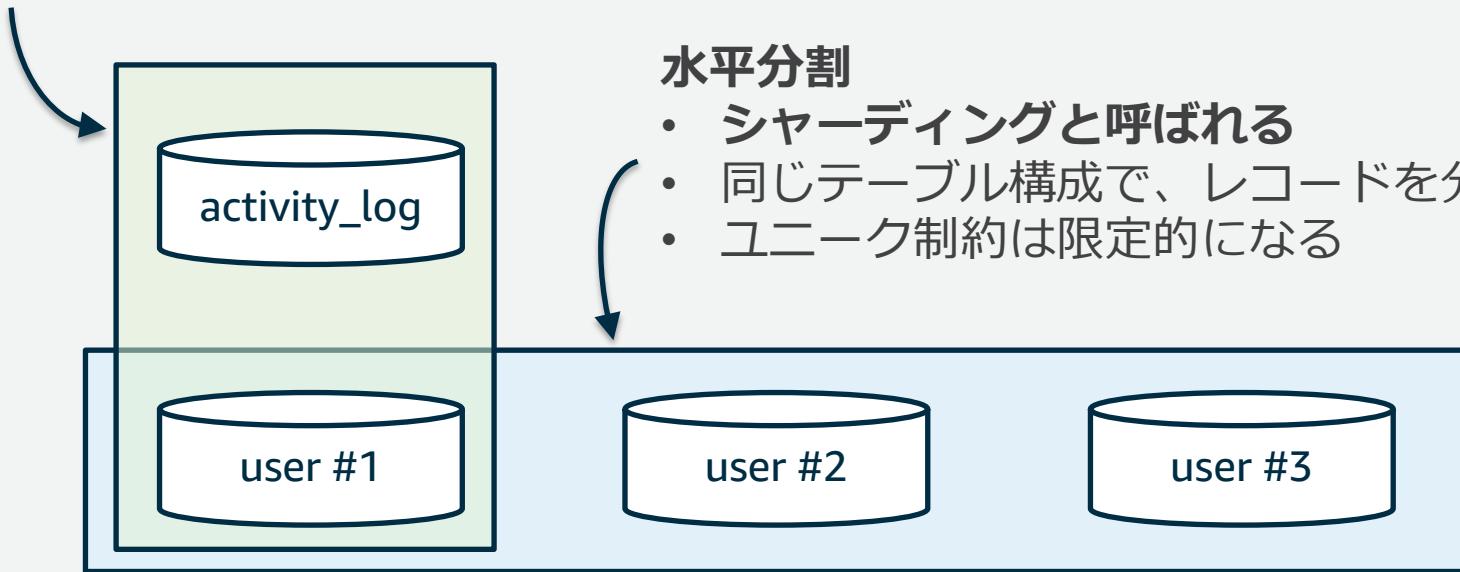
- データ分散のアルゴリズムの考慮
 - どういったルールでデータを振り分けるか
 - シャーディングでは、リシャードによるデータ移動が発生する
- 分割した DB/シャード をまたがるトランザクションは難しい



垂直分割と水平分割

垂直分割

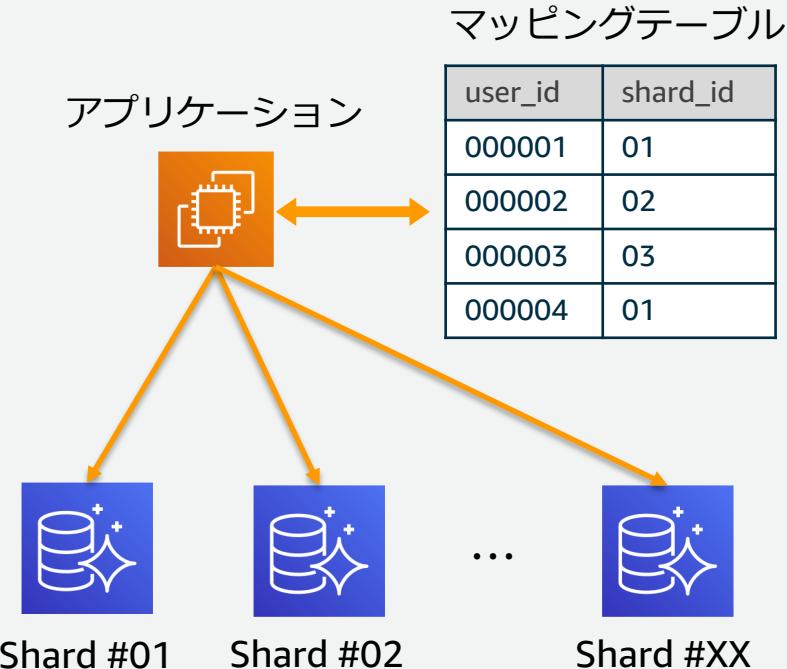
- ・ テーブル単位で分割
- ・ ログテーブルなど、**要件が全く異なるテーブル**に適用しやすい
- ・ ワークロードに偏りがある場合、スケールしない



水平分割

- ・ シャーディングと呼ばれる
- ・ 同じテーブル構成で、レコードを分割
- ・ ユニーク制約は限定的になる

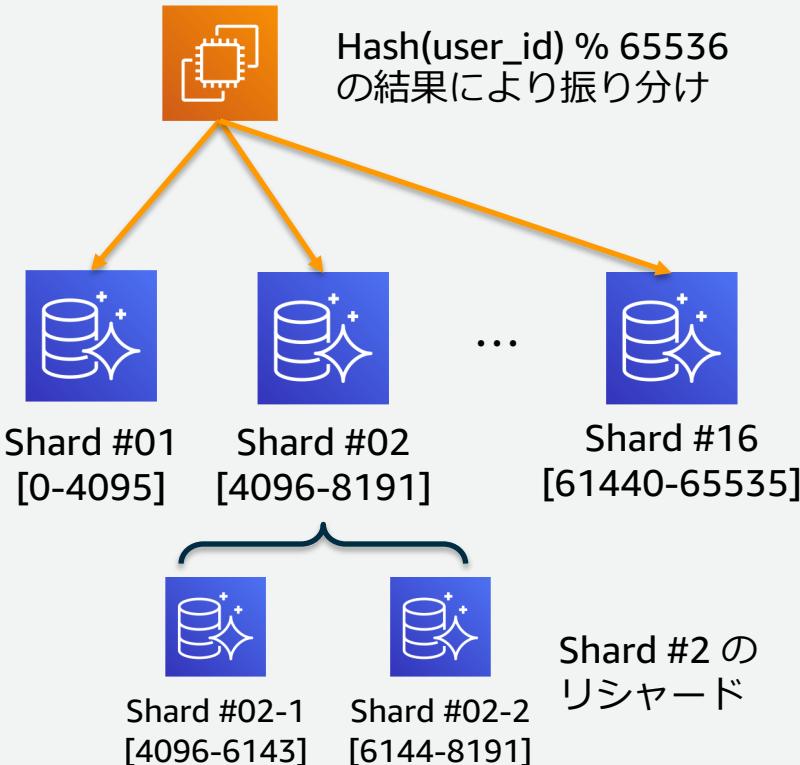
シャーディングロジックの例：マッピングテーブル



- レコードとシャードを対応づけるマッピングテーブルを作成
- ユーザごとの偏りに対処しやすい
 - 新規ユーザはシャードの容量が小さいものに振り分けるなど、柔軟な運用
- マッピングテーブルが肥大化するにつれボトルネックになる。単一障害点にならないよう注意する必要がある

シャーディングロジックの例：ハッシュ+レンジ分割

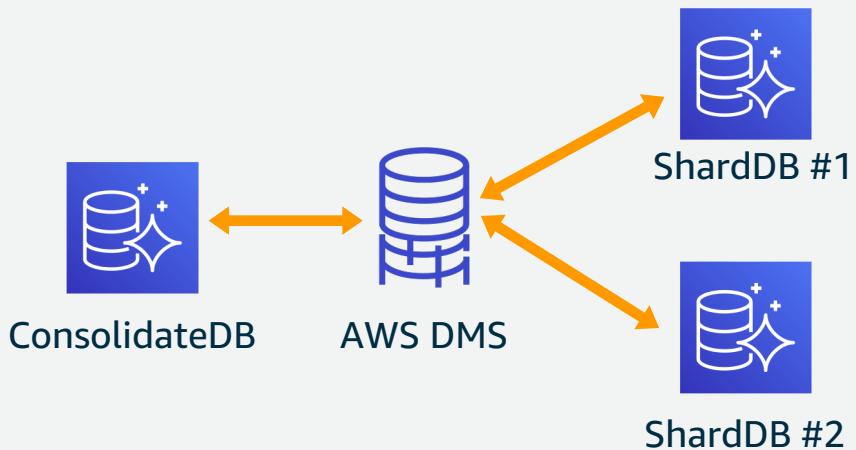
アプリケーション



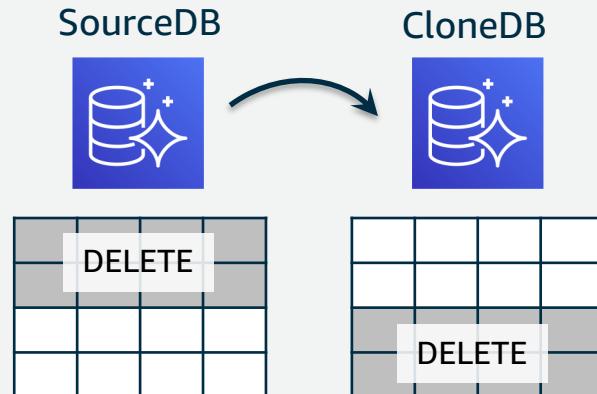
- 特定のカラムをもとに、
ハッシュ関数と modulo により
取りうる範囲の値を絞って分割
 - ハッシュ関数は crc32 などを利用
- $Hash(value) \% N$ の N は、
シャード数よりも十分に大きく設定
 - 再分割時のデータ移動を防ぐ
- 2のべき乗に分割しておき、
リシャードはインスタンスを
2倍に増やす方法がポピュラー

ダウンタイムの少ないデータベース分割

AWS DMS (Database Migration Service)によるデータベース分割・集約



Aurora の Cloning 機能を用いて
クローン後、不要なデータの削除



参考: <https://aws.amazon.com/jp/blogs/news/sharding-with-amazon-relational-database-service/>

Cloning 詳細 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Clone.html

#2-b. [参考] 用途に合わせた適切なデータベースの利用

フィットするケース

- ・ ワークロード・データ構造が RDBMS に向いていない場合は、データベースサービスの使い分けも検討
 - ・ 巨大なデータ・KVS なら DynamoDB
 - ・ 分析・集計なら Redshift
 - ・ データ構造がグラフなら Neptune

キーポイント

- ・ 変更の大きさと、メリットとして得られる運用性のトレードオフ
- ・ 異種 DB 間のデータ移行に DMS が利用できる

[参考] 目的に応じたデータベースの選択



Relational

Analytics

Key-value

Document

In-memory

Graph

Time-series

Ledger

特徴

参照整合性、
ACIDトランザクション、
Schema-On-Write

ペタバイト級
までスケール
アウト、並列
分散処理に
による高い
スループット

高スループット、
低レイテンシー
の読み取り、
書き込み、
無限のスケール

ドキュメントを
保存し、任意の
属性にクエリー
でアクセス、
Schema-On-
Read

マイクロ秒の
レイテンシー
でキーによる
クエリ

すばやく簡単に
データ間の
関係を作成し
ナビゲート

データを
時間順に収集、
格納、処理

完全で不变で
検証可能な
アプリケーションデータに
対するすべての
変更履歴

ユースケース

リフト&シフト、
ERP、CRM、
金融

DWH、データ
マート、BI基盤、
経営ダッシュ
ボード

リアルタイム
入札、
ショッピング
カート、
ソーシャル、
製品カタログ、
顧客の好み

コンテンツ管理、
パーソナライゼーション、
モバイル

リーダー
ボード、
リアルタイム
分析、
キャッシング

不正検出、
ソーシャル
ネットワーキング、
レコメンド
エンジン

IoTアプ
リケーション、
イベント
トラッキング

SoR(System of
Record)、
サプライチェー
ン、ヘルスケア、
届出、財務

[参考] AWS のフルマネージド型データベースサービス

							
Relational	Analytics	Key-value	Document	In-memory	Graph	Time-series	Ledger
							
Amazon RDS	Amazon Redshift	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon QLDB

マネージドサービス



[参考] Aurora Multi-Master はスケールアウト手段か？

- Writer が2インスタンス = 書き込み性能が2倍 ではない。
Multi-Master は ”**継続的な可用性**” を目的としたソリューション
 - 片方の Writer のインスタンス障害時にアプリケーションがもう片方の Writer へ接続することで、書き込みの可用性を継続する
- Single-Master (通常の Aurora) はほとんどのワークロードに適しており、スケーラビリティ・パフォーマンス要件がある場合も Single-Master を検討
- 詳しくは、ユーザーガイドをご確認ください
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-multi-master.html

Aurora MySQL のスケーリング

Case #1

読み込みパフォーマンスのスケーリング

Case #2

書き込みパフォーマンスのスケーリング

Case #3

インフラストラクチャの地理的スケーリング

Case #4

コスト最適化のためのスケーリング

#3 インフラストラクチャの地理的スケーリング

解決したい課題

- 異なるリージョンからのリクエストのレイテンシを小さくしたい

解決のパターン

- a. Aurora グローバルデータベース
- b. クロスリージョンリードレプリカ

#3-a. Aurora グローバルデータベース

フィットするケース

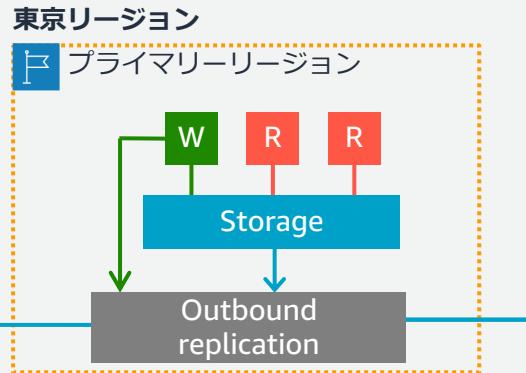
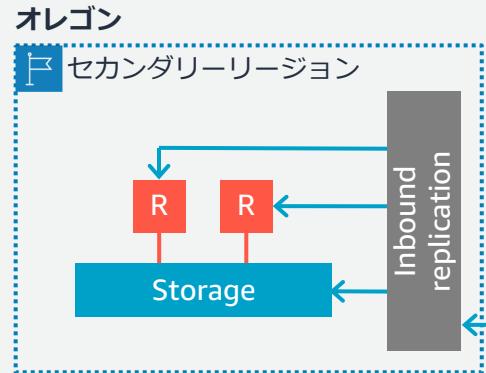
- データを他のリージョンにも持ちたい
 - データをアプリケーションに近づけ、レイテンシを減らす
 - 災害対策（DR）

キーポイント

- binlog ではなく Aurora 独自のインフラストラクチャを利用したレプリケーションであり、クロスリージョンリードレプリカに比べレプリカラグははるかに小さい

Aurora グローバルデータベースとは

高速な災害対策と拡張されたデータローカリティー



高いスループット: 最大200K writes/sec

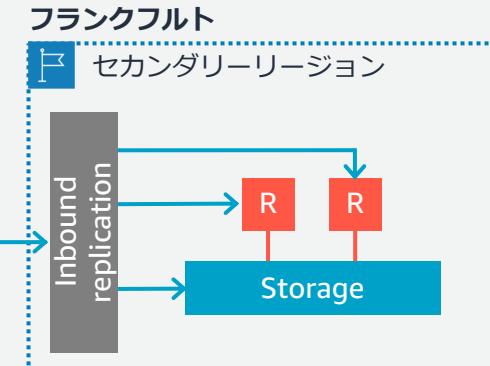
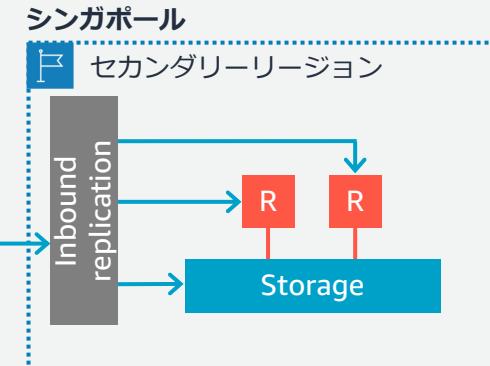
低いレプリカラグ: 高負荷状態でもリージョン間でレプリカラグは1秒未満

高速なフェイルオーバー: リージョン障害後、通常1分未満

- 最大5つの複数のセカンダリーリージョンの利用が可能
- インプレースでのグローバルデータベースへの変換をサポート

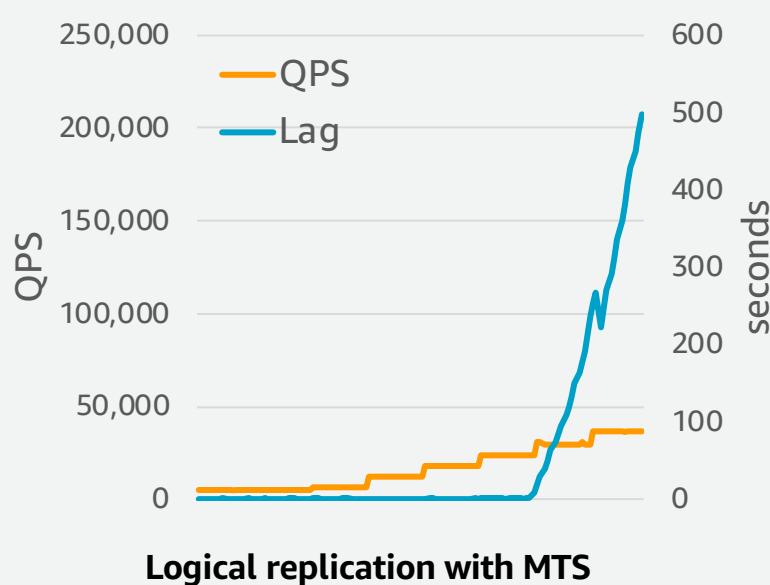
詳細 : https://docs.aws.amazon.com/ja_ip/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

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

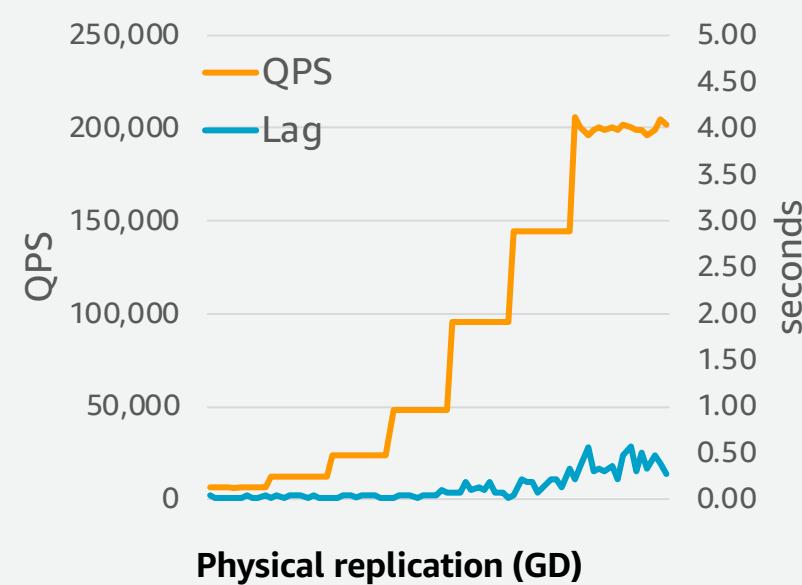


グローバルデータベースのパフォーマンス

Logical (binlog) vs. Physical (GD) MySQL replication



Logical replication with MTS



Physical replication (GD)

SysBench OLTP (write-only) stepped every 600 seconds on R4.16xlarge

詳細：https://docs.aws.amazon.com/ja_ip/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

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

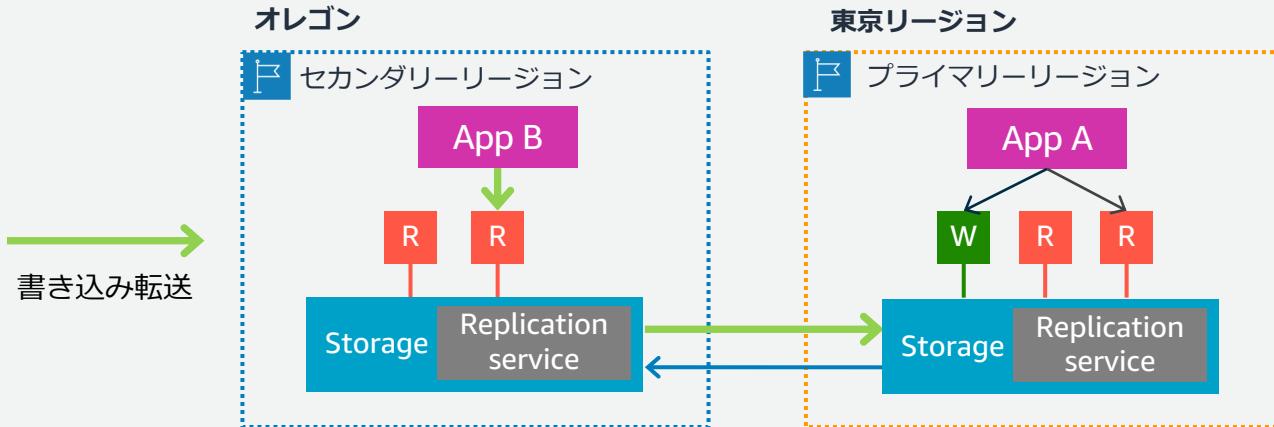


Aurora グローバルデータベース: 書き込み転送

セカンダリリージョンの Aurora クラスタにも書き込みが可能

- 書き込みは透過的にプライマリリージョンに転送され、適用された変更はすべてのセカンダリリージョンへプリケーションされる
- セッション変数 `aurora_replica_read_consistency` のセットにより、セカンダリリージョンのクラスタの読み取り整合性レベルを制御
- 詳細はユーザガイドを参照

https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding.html



Aurora MySQL のスケーリング

Case #1

読み込みパフォーマンスのスケーリング

Case #2

書き込みパフォーマンスのスケーリング

Case #3

インフラストラクチャの地理的スケーリング

Case #4

コスト効率の良いスケーリングをしたい

#4 コスト効率の良いスケーリングをしたい

解決したい課題

- たまにしか使用しないデータベースのコストを抑えたい
 - 社内ツール、開発・テスト用データベース
- 予測が困難で、ピーク時と非ピーク時の負荷の差が激しく、コスト効率の良いスケーリングが難しい
 - 例えば、雨天時にのみリクエストが急増するなど

解決のパターン

- a. Aurora Serverless

#4-a. Aurora Serverless

フィットするケース

- 開発・テスト用インスタンス
- 不定期利用のインスタンス
- 予測不能なワークロード



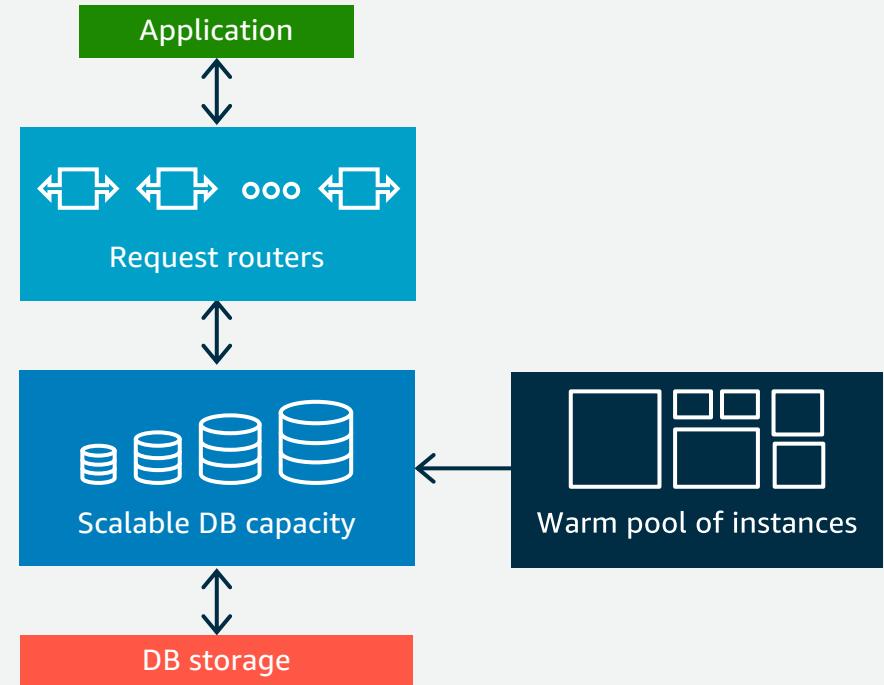
キーポイント

- Aurora Serverless によるスケーリングが、実際に要求されるキャパシティを満たすものかどうかは、要検証であること
- スケーリングポイントの理解

Aurora Serverless とは

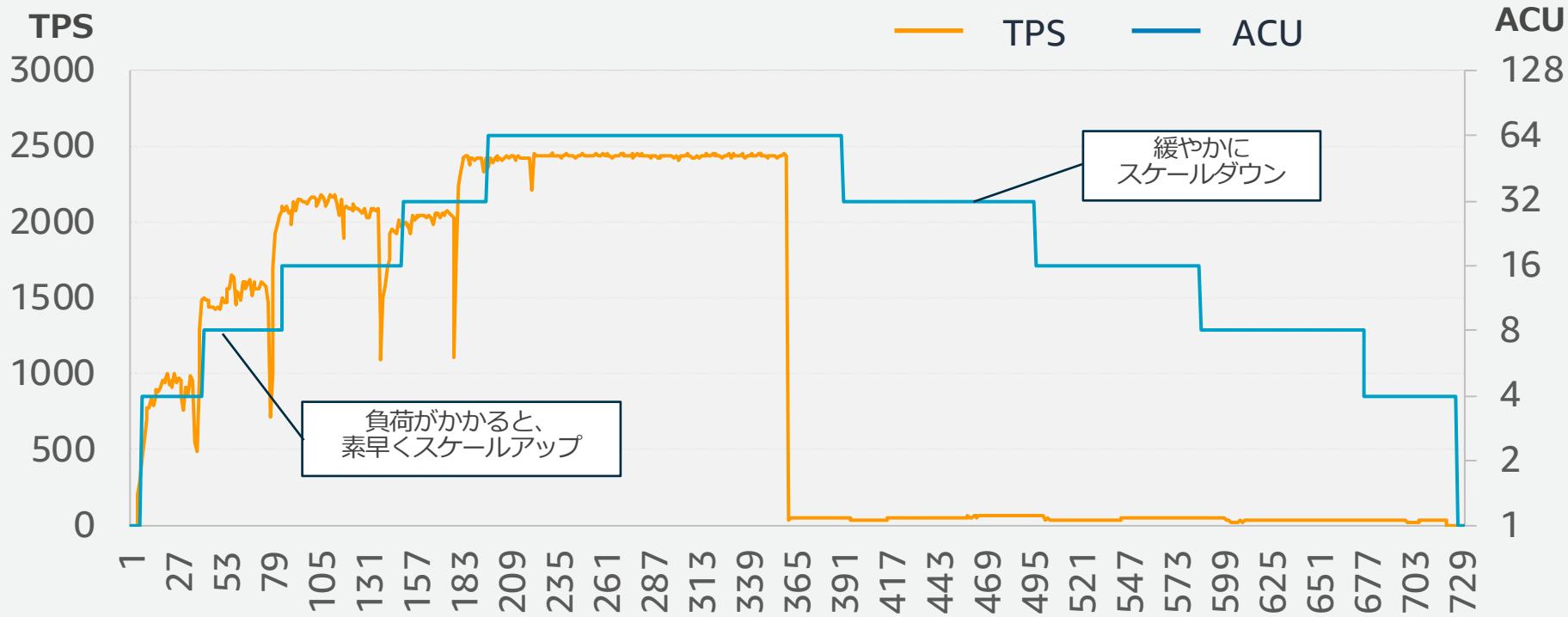
コンピューティングキャパシティーが自動管理される Amazon Aurora

- オンデマンドで起動
- 利用されていない場合は自動停止（ストレージ料金のみ課金）
- データベース（CPU、メモリ、接続）の負荷を監視し、**利用状況に応じて自動的にスケーリング**
- スケーリング操作はアプリケーションに透過的で、クライアント接続の中断なし
- 秒単位の課金
 - ただし、1分が最低利用料金



詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html

負荷に応じたスケールアップ・ダウン



詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html

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



スケーリングポイント

Aurora Serverless はシームレスにスケーリングをするために、**スケーリングポイント**を探す。しかし、以下の条件ではスケーリングポイントが見つけられず、スケールアップ・ダウンに失敗する場合がある

- ・ 長期実行のクエリまたはトランザクションが実行中である
- ・ 一時テーブル・テーブルロックが利用中である

スケーリングポイントを見つけられなかった場合のふるまいを設定できる

- ・ **キャパシティー変更のロールバック:** キャパシティーを変更しない
- ・ **キャパシティーの強制変更:** タイムアウト時、キャパシティーを強制的に変更する。
スケーリングポイントの検出を妨げる接続は、**切断される可能性がある**

▼ スケーリングの追加設定

- タイムアウトに達すると、容量を指定された値に強制的にスケーリングします **情報**

できるだけ早く容量スケーリングを強制できるようにします。タイムアウトになったときに容量変更のキャンセルを無効にします

コストの比較

メモリを基準に、オンデマンド/リザーブドインスタンスを常時起動した場合の料金との比較

タイプ	メモリ	支払い方法	Cost [USD/hour]	備考
Provisioned: db.t3.small	2 GB	オンデマンド	0.063	Aurora Serverless の1日あたりの起動時間が 約15時間以下 : オンデマンドインスタンスよりも安価 約11時間以下 : リザーブドインスタンスよりも安価
		リザーブド	0.046	
		-	0.10	
Serverless: 1ACU	16 GB	オンデマンド	0.35	Aurora Serverless の1日あたりの起動時間が 約10.5時間以下 : オンデマンドインスタンスよりも安価 約5.85時間以下 : リザーブドインスタンスよりも安価
		リザーブド	0.195	
		-	0.80	

※ 東京リージョン、リザーブドインスタンスはスタンダード1年間、全額前払いで計算

- ストレージ料金、I/O リクエストは通常の Aurora、Aurora Serverless どちらも同じ料金設定で、**インスタンス料金のみ異なる**
- Aurora Serverless におけるインスタンス料金は ACU (Aurora Capacity Unit) が相当する。1ACU あたり 約2GB のメモリと対応する CPU/Network が割り当てられる。料金は 0.10 USD/ACU hour となる。

詳細：<https://aws.amazon.com/jp/rds/aurora/pricing/>

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



Aurora Serverless の注意事項

- Provisioned Aurora と異なり、障害時は別 AZ にインスタンスを再作成
 - ただし、フェイルオーバー時間は未定義
- メンテナンス適用時も、スケーリングポイントを探す

Serverless クラスターの制限

- パブリック IP アドレスの割り当てはできない
- レプリカ、クローン、バックトラック、Parallel Query などの機能は利用できない etc

詳細およびその他の注意事項は、ユーザーガイドをご参照ください

https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html

アジェンダ

- Amazon Aurora 概要
- Latest updates
- Amazon Aurora MySQL をスケールさせる
- まとめ

まとめ

- Amazon Aurora はクラウド向けに Amazon が再設計した RDBMS
- お客様の課題を解決するため、Amazon Aurora は進化を続けています
- 課題に合わせた適切なソリューションを選択しましょう
 - レプリカ
 - グローバルデータベース
 - Serverless
 - Multi-Master
- ...

参考資料

- Amazon Aurora MySQL 詳細
 - <https://aws-ref.s3.amazonaws.com/aurora/Amazon+Aurora.pdf>
- Amazon Aurora ユーザーガイド
 - https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/CHAP_AuroraOverview.html
- Amazon Aurora MySQL データベース管理者ハンドブック
 - https://d1.awsstatic.com/whitepapers/ja_JP/RDS/amazon-aurora-connection-management-handbook.pdf

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

The screenshot shows the AWS Japan Language Resources homepage. At the top, there's a navigation bar with the AWS logo, search bar, and links for '日本担当チームへお問い合わせ' (Contact Support), 'サポート' (Support), '日本語' (Japanese), 'アカウント' (Account), and 'コンソールにサインイン' (Sign In). Below the navigation is a main menu with links for '製品' (Products), 'ソリューション' (Solutions), '料金' (Pricing), 'ドキュメント' (Documentation), '学習' (Learning), 'パートナー' (Partners), 'AWS Marketplace' (AWS Marketplace), 'その他' (Other), and a search icon. The main content area features a large title 'AWS クラウドサービス活用資料集トップ' (Top of the AWS Cloud Service Utilization Document Collection) and a descriptive paragraph about the service. At the bottom, there are four buttons: 'AWS Webinar お申込' (Apply for AWS Webinar), 'AWS 初心者向け' (For AWS beginners), '業種・ソリューション別資料' (Industry and Solution-specific documents), and 'サービス別資料' (Service-specific documents).

<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>

