

AWS Black Belt Online Seminar

Amazon Aurora の コスト最適化

西原 陽介

Technical Account Manager

2025/06



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
- 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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBIqY>



ご感想は X (Twitter) へ！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

- 本資料では資料作成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。AWS のサービスは常にアップデートを続けているため、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
- 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
- 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)



自己紹介

西原 陽介

アマゾンウェブサービスジャパン
テクニカルアカウントマネージャ

金融業界のお客様を中心にご支援しています。

好きな AWS サービス
Amazon Aurora



本セミナーの対象者

- データベースのクラウド移行を検討されている方
- Amazon Aurora の利用を検討中、または今後検討をご予定の方
- Amazon Aurora をご利用中でコスト削減を検討中の方



アジェンダ

Amazon Aurora コスト最適化アプローチ (基本編)

- Amazon Aurora のコスト構造
- インスタンスのダウンサイジング
- インスタンスタイプの変更
- Reserved Instance の購入
- 使用しないクラスタ (インスタンス) の停止や削除

Amazon Aurora コスト最適化アプローチ (応用編)

- Aurora Serverless V2
- Aurora I/O 最適化
- ストレージ費用削減
- Amazon ElastiCache との併用



Amazon Aurora コスト最適化アプローチ (基本編)



Amazon Aurora のコスト構造

Aurora Global Database

Data Transfer

Backup

Extended Support

Backtrack (Aurora MySQL)

Snapshot

Data API

Export to S3

ユースケースに
依存するコスト

今回、説明する箇所

インスタンス
(Provisioned/Serverless)

I/O
(I/O 最適化/スタンダード)

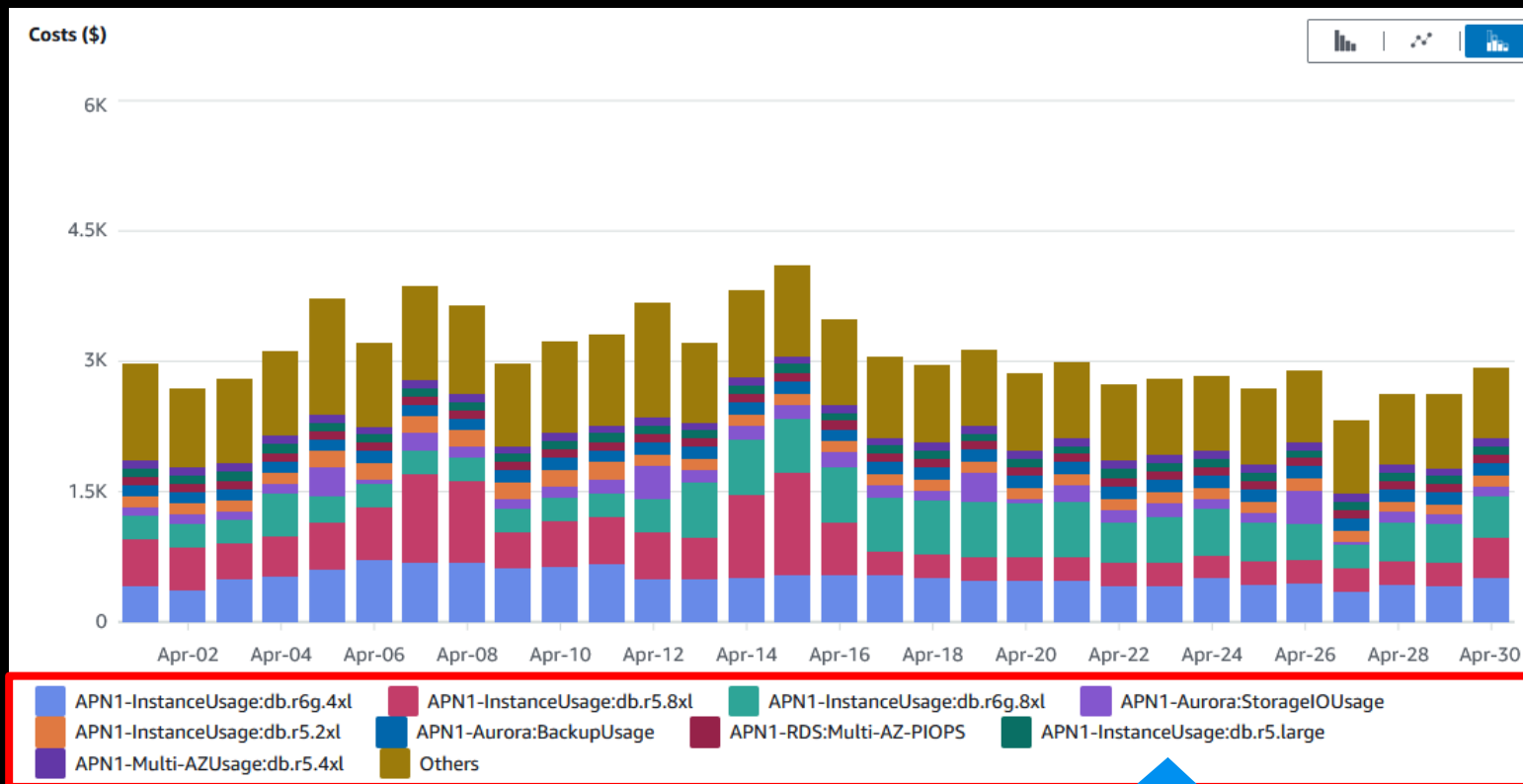
ストレージ

基本的なコスト



参考: Cost Explorer での Aurora の費用確認

- Cost Explorer の「使用タイプ」を表示する事で、RDS/Aurora の利用料の詳細が確認可能



Aurora の費用の詳細を
確認する事が可能

▼ グループ化の条件

ディメンション クリア

使用タイプ ▼

▼ フィルター 情報

適用フィルター (1) すべてをクリア | ⚙

サービス クリア

サービスを含む (1) ▼

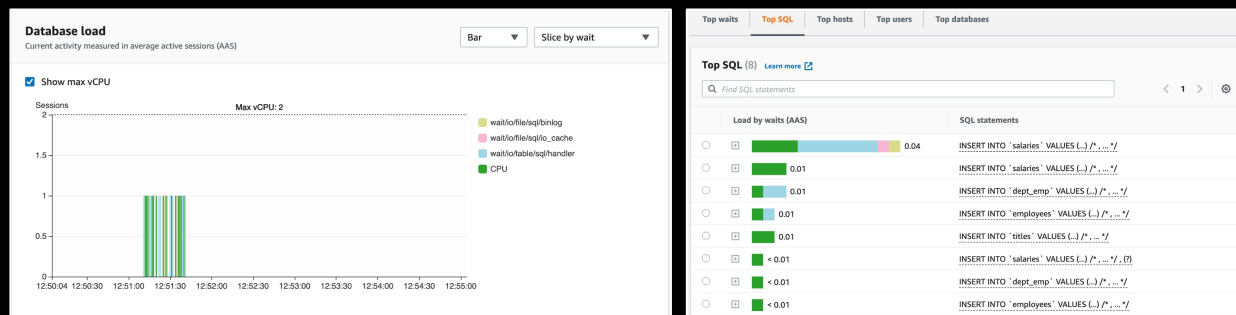
Relational Database Service (RDS) X



インスタンスのダウンサイジング

適切なサイズに変更する事でコスト削減を図る

- Performance Insights や CloudWatch でリソース状況に余裕のあるインスタンスを確認



CPU 使用率や I/O wait

SQL 統計

- Performance Insights は 7 日分のパフォーマンスデータを保持
- より長期間のパフォーマンスデータの履歴を利用する為には追加料金が必要

<https://aws.amazon.com/jp/rds/performance-insights/pricing/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

- インスタンスタイプを小さいサイズに変更する




db.r6g.4xlarge
\$ 2.506/h



db.r6g.2xlarge
\$ USD 1.253/h

* 2025 年 5 月時点の Aurora Standard 東京リージョンの価格

- 性能影響がない事を事前に確認した上で変更する
- インスタンスを変更する際は対象の DB エンジンでサポートされているかを確認する
- SQL チューニングの余地が無いか確認**  事も推奨

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

インスタンスタイプの変更

世代やプロセッサファミリーを見直す事によりコストパフォーマンス向上を図る

- 最新世代へ移行



db.r5.4xlarge
\$ 2.80/h



db.r7i.4xlarge
\$ 2.80/h

金額は同一だが、
スペックは改善



db.r7i.2xlarge
\$ 1.40/h

インスタンスサイズ
縮小を検討

- CPU の変更



db.r6i.4xlarge
\$ 2.80/h



db.r6g.4xlarge
\$ 2.506/h
(△10%)

Graviton CPU が Intel
CPU よりもコア当たり
の単価は低い

* 2025 年 5 月時点の Aurora Standard 東京リージョンの価格

- インスタンスを変更する際は対象の DB エンジンでサポートされているか事前に確認する



参考: Compute Optimizer – Aurora インスタンスの推奨事項

- Compute Optimizer コンソール画面で有効化する事で利用可能に
- Aurora インスタンスの推奨事項を提示
- Performance Insights を有効とすると、過剰なプロビジョニングとなっているインスタンスと推奨するインスタンスタイプを提示してくれる

AWS Compute Optimizer > ダッシュボード > Recommendations for RDS DB instances

インスタンス | ストレージ

Recommendations for RDS DB instances (0) 情報

コストとパフォーマンスを改善するための現在の RDS インスタンスに関する推奨事項。

CPU アーキテクチャの詳細設定: 現在, Graviton (aws-arm64)

1 つ以上のプロパティでフィルタリング

タグキー | タグ値

データベース識別子 | エンジン | アイドル 情報 | インスタンスの検出結果 情報 | インスタンスの検出結果の理由 情報 | 推定月間節約額 (割引後) 情報 | 推定月間節約額 (オンデマンド) 情報

インスタンス検出結果

プロビジョニング不足

最適化済み

過剰なプロビジョニング

「過剰なプロビジョニング」の検出は、Performance Insights の有効化が前提

アイドルの検出

推定節約額

https://docs.aws.amazon.com/ja_ip/compute-optimizer/latest/ug/what-is-compute-optimizer.html



Reserved Instance (RI) の購入

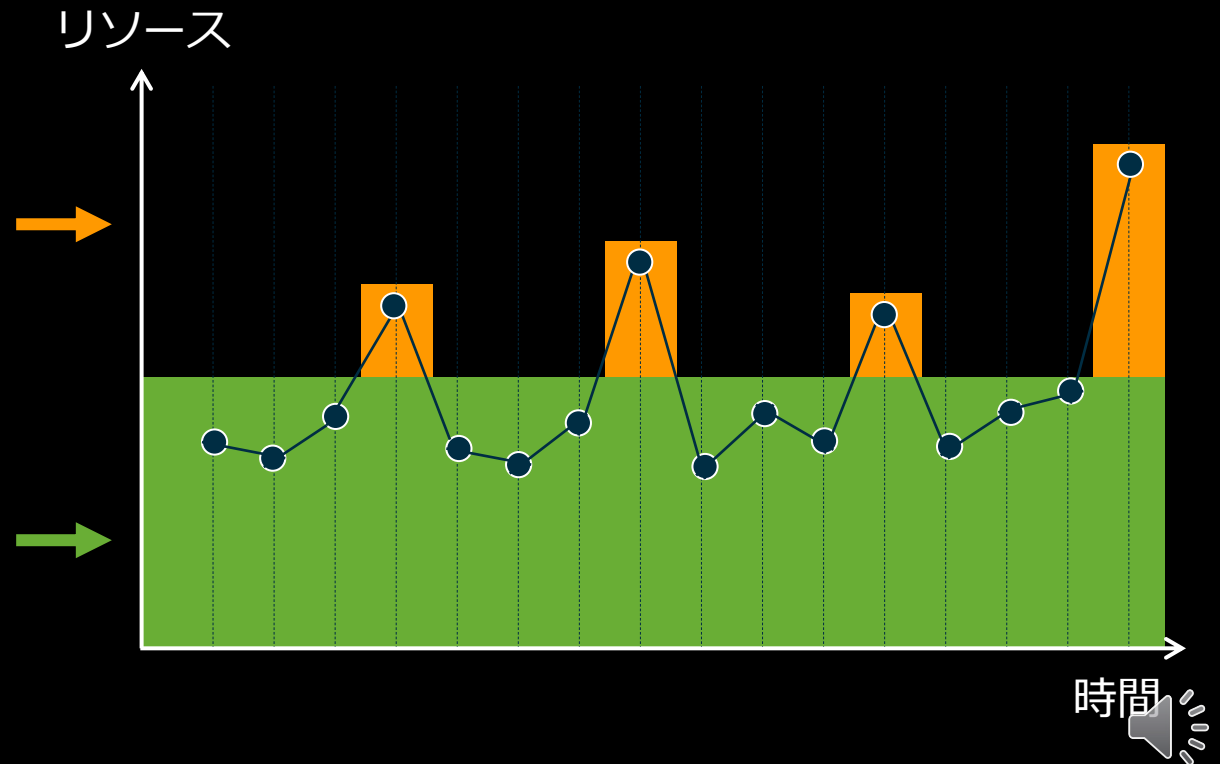
1 年 or 3 年の契約コミットメントにより、オンデマンド料金と比較して最大 69% のコストを削減可能

オンデマンド
インスタンス

初期費用なし、
コミットなし、
従量課金

リザーブド
インスタンス

長期(1年or3年)の
利用コミットに
よる割引の適用
(最大69%割引)



https://docs.aws.amazon.com/ja_ip/AmazonRDS/latest/UserGuide/USER_WorkingWithReservedDBInstances.html

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



参考: AWS Cost Explorer による RI の適用箇所の確認

これまでの利用履歴データを利用してリザーブドインスタンス推奨事項を確認することが可能

AWS コスト管理

- ホーム
- Cost Explorer
- レポート
- Budgets
- コスト異常検出
- サイズの適正化に関する推奨事項
- ▼ Savings Plans
 - 概要
 - インベントリ
 - 推奨事項 **新規**
 - Savings Plans の購入
 - 使用状況レポート
 - カバレッジレポート
 - カート 0
- ▼ 予約
 - 概要
 - 推奨事項**
 - 使用状況レポート
 - カバレッジレポート

推奨事項

推奨事項パラメータ 情報

サービス: Relational Database Service (RDS)

期間: 1-年 (選択), 3-年

基準とする直近の期間: 7日, 30日 (選択), 60日

支払いオプション 情報: 全額前払い (選択), 一部前払い, 前払いなし

推奨事項 (1) 情報 Amazon RDS コンソール

購入推奨事項の総数: 1

推定月別削減額: \$81.49

オンデマンド料金に対する推定削減額: 45%

推奨されるアクション

インスタンスタイプでフィルタリング

アクション	インスタンスタイプ	リージョン	データ...	ライセ...	現在の世代	前払いコスト
1 db.r5.large を購入する	db.r5.large	米国東部 (オハイオ)	PostgreSQL	No license...	はい	\$1,180.00

https://docs.aws.amazon.com/ja_ip/awsaccountbilling/latest/aboutv2/ce-what-is.html



利用しないクラスタ (インスタンス) の停止や削除

クラスタ

- 開発環境やテスト環境を週末夜間等の利用しない時間帯に停止することでコストを抑える
- 停止中はストレージ、バックアップのみ課金
- 使用頻度が少ない場合、バックアップを取得してクラスタ削除
- Aurora Global Database のヘッドレス構成を検討

<注意点>

- クラスタを停止しても、停止後 7 日間で自動で起動する
- Aurora Global Database の一部であるクラスタの停止は不可
- クロスリージョンリードレプリカを持つクラスタの停止は不可

インスタンス

- 過剰なリードレプリカ等、ワークロードに対して不要と判断できるインスタンスを削除

https://docs.aws.amazon.com/ia_jp/AmazonRDS/latest/AuroraUserGuide/aurora-cluster-stop-start.html
https://docs.aws.amazon.com/ia_jp/AmazonRDS/latest/AuroraUserGuide/USER_DeleteCluster.html

クラスタ停止時の設定イメージ

DB クラスタを一時的に停止

この database-1 DB クラスタは最大 7 日間停止しようとしています。DB クラスタはいつでも手動で再起動できます。 [詳細はこちら](#)

了承する

☒ 私は、DB インスタンスを停止しても、請求が一時停止されるのはインスタンス時間だけであることを了承します。DB インスタンスは May 13, 2025, 09:56 (UTC+09:00) に自動的に再起動します。

⚠ DB クラスタが停止している間、プロビジョンド IOPS を含め、プロビジョンドストレージに対して請求されます。また、バックアップストレージの料金も請求されます。これには、保持期間内の手動スナップショットと自動バックアップが含まれます。詳細については、「[請求に関するよくある質問](#)」を参照してください。

ℹ Lambda 関数を使用すると、メンテナンス期間の開始時と終了時に DB クラスタを自動的に起動および停止できます。 [詳細はこちら](#)

キャンセル 一時的に停止

インスタンス削除の設定

アクション ▲ データベースの作成

再起動

削除

フェイルオーバー

Amazon Aurora コスト最適化 (基本編) まとめ

事前確認

Cost Explorer
Aurora 費用確認

Performance Insights
CloudWatch 確認

インスタンス最適化

インスタンス
ダウンサイジング

インスタンスタイプ
変更

RI 購入／クラスタ停止

Reserved Instance
購入

利用しないクラスタ
停止・削除



Amazon Aurora コスト最適化アプローチ (応用編)



Aurora Serverless v2



オンデマンドで自動的にスケール

アプリケーションのニーズに応じて**自動的に容量を拡張**

秒単位のシンプルな**従量課金**

柔軟に拡張し、要求の厳しいアプリケーションをサポート

データベースの容量管理の心配からの解放



Aurora Serverless v2 の特徴

- インスタンスのリソース容量を **ACU (Aurora Capacity Unit)** で管理
- 各 ACU は約 **2GiB(ギビバイト)** のメモリと対応する **CPU**、ネットワークが組み合わされた容量となる
- 最小 ACU と最大 ACU を指定し、その範囲で**自動的にスケールアップ／スケールダウン**を実施する
- ACU は **0 ～ 256** の範囲で指定可能
- Aurora MySQL 3.02.0 以降、Aurora PostgreSQL 13.6、14.3、15.2、16.1、17.4 以降で利用可能
- プロビジョンドインスタンスとの混在に対応

インスタンスの設定
以下の DB インスタンスの設定オプションは、上記で選択したエンジンでサポートされているものに制限されています。

DB インスタンスクラス | [情報](#)

▼ **フィルター**の非表示

☐ 以前の世代のクラスを含める

☒ **Serverless v2**

☐ メモリ最適化クラス (r クラスを含む)

☐ パースト可能クラス (t クラスを含む)

☐ 最適化された読み取りクラス - 新規

Serverless v2
最も要求の厳しいワークロードでも瞬時にスケーリング。

容量の範囲 | [情報](#)
データベース容量は Aurora 容量ユニット (ACU) で測定されます。1 ACU は 2 GiB のメモリと、対応するコンピューティングとネットワーキングを提供します。

最小キャパシティ (ACU) (0 GiB) 最大キャパシティ (ACU) (128 GiB)

0～256 (0.5 の増分) 1～256 (0.5 の増分)

非アクティブ後に一時停止 | [情報](#)
最小キャパシティ設定が 0 ACU のクラスターは、非アクティブの間は一時停止できます。DB インスタンスがアイドル状態になれる時間 (一時停止するまでの時間) を指定します。詳細については、[Aurora Serverless DB インスタンスを一時停止](#) を参照してください

5 分から 24 時間までの時間を入力してください

Aurora Serverless V2 のインスタンス設定



Aurora Serverless v2 のユースケース

- **使用頻度の低いアプリケーション**
 - 1 日または 1 週間に数回、数分間のみ使用されるアプリケーション等
- **開発、テスト用のデータベース**
 - 夜間や週末にデータベースを使用しないケース
- **可変ワークロード**
 - 予算編成や人事異動等の特定のイベント時のみ必要となるケース
- **予測不可能なワークロード**
 - 1 日を通して常時稼働しているデータベースでアクティビティのピークを予測するのが困難なケース
- **企業のデータベースフリート管理**
 - 各データベースキャパシティの継続的な管理からの解放
- **Software-as-a-Service アプリケーション**
 - 顧客毎にデータベースを用意するケース等で複雑なリソース管理からの解放
- **複数のサーバに分割され、スケールアウトされたデータベース**
 - Reader のスケールを自動調節してパフォーマンスとコストを自動的に最適化



Aurora Serverless v2 のコスト試算

ピーク時のみスケールするようなワークロードに対して、Serverless V2 を利用することでコスト最適化が見込める

種類		スペック	インスタンスタイプ 別料金 (*2)	料金 (30日換算)
Provisioned	Amazon Aurora	4vCPU/32GiB memory	db.r5.xlarge \$ 0.70/時間	\$ 504.00
		4vCPU/32GiB memory	db.r6g.xlarge \$ 0.627/時間	\$ 451.44
Serverless	Aurora Serverless V2	2 ACU (4GB) ~ 16 ACU (32GiB) *1	下記ピーク時は 16 ACU (通常時は 2 ACU) 毎日 11:00~13:00 (2h) \$ 0.15/ACU 時間	\$ 342.00
		0 ACU ~ 16 ACU (32GiB)	下記ピーク時は 16 ACU (通常時は 0 ACU) 毎日 11:00~13:00 (2h) \$ 0.15/ACU 時間	\$ 144.00

*1 - Performance Insights 利用時における推奨の ACU 最小容量は 2 ACU

*2 - 2025 年現在の Aurora Standard 東京リージョンの価格

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



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



混在環境によるコスト最適化

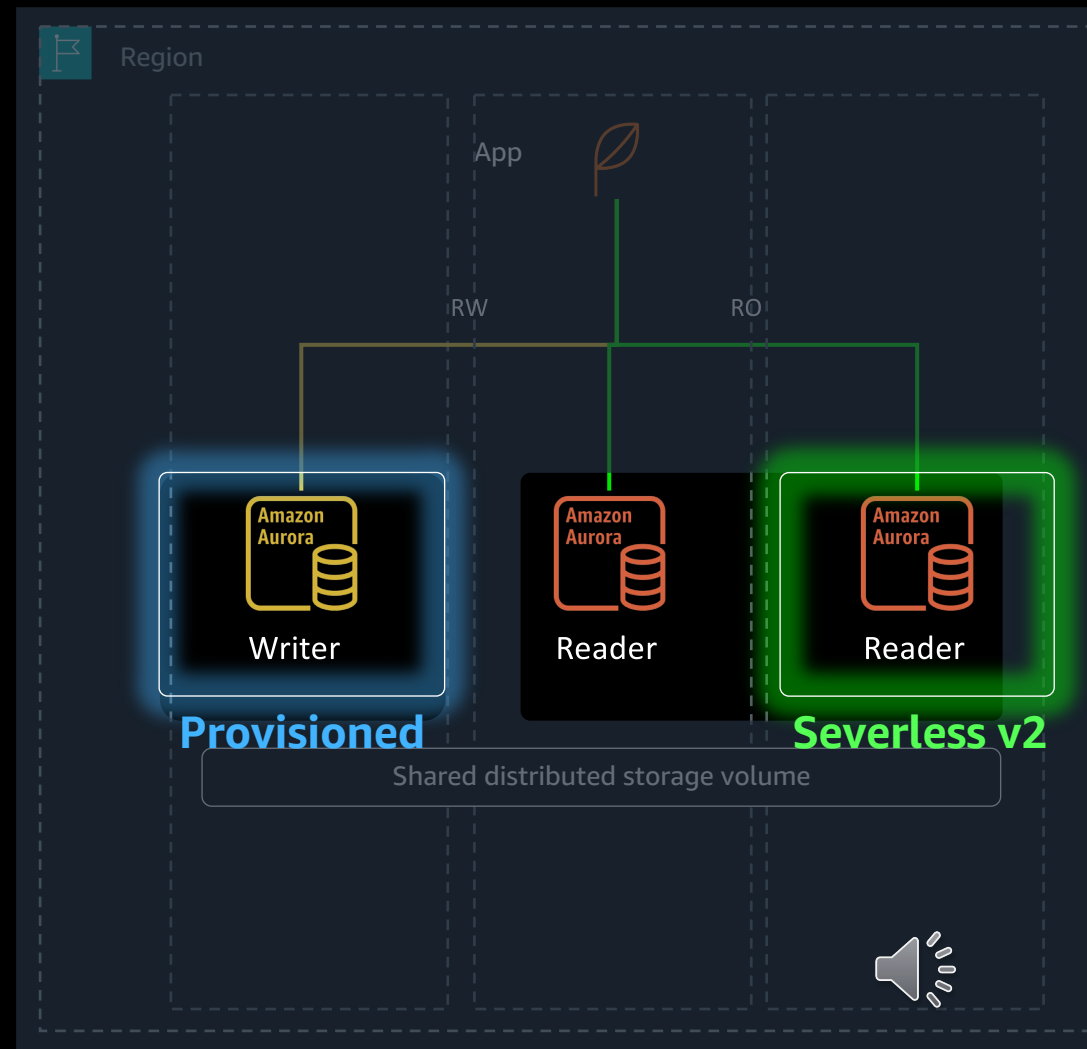
Provisioned と Serverless v2 の混在環境

- 既存のクラスター内のインスタンスを Aurora Serverless v2 に変換可能

例)

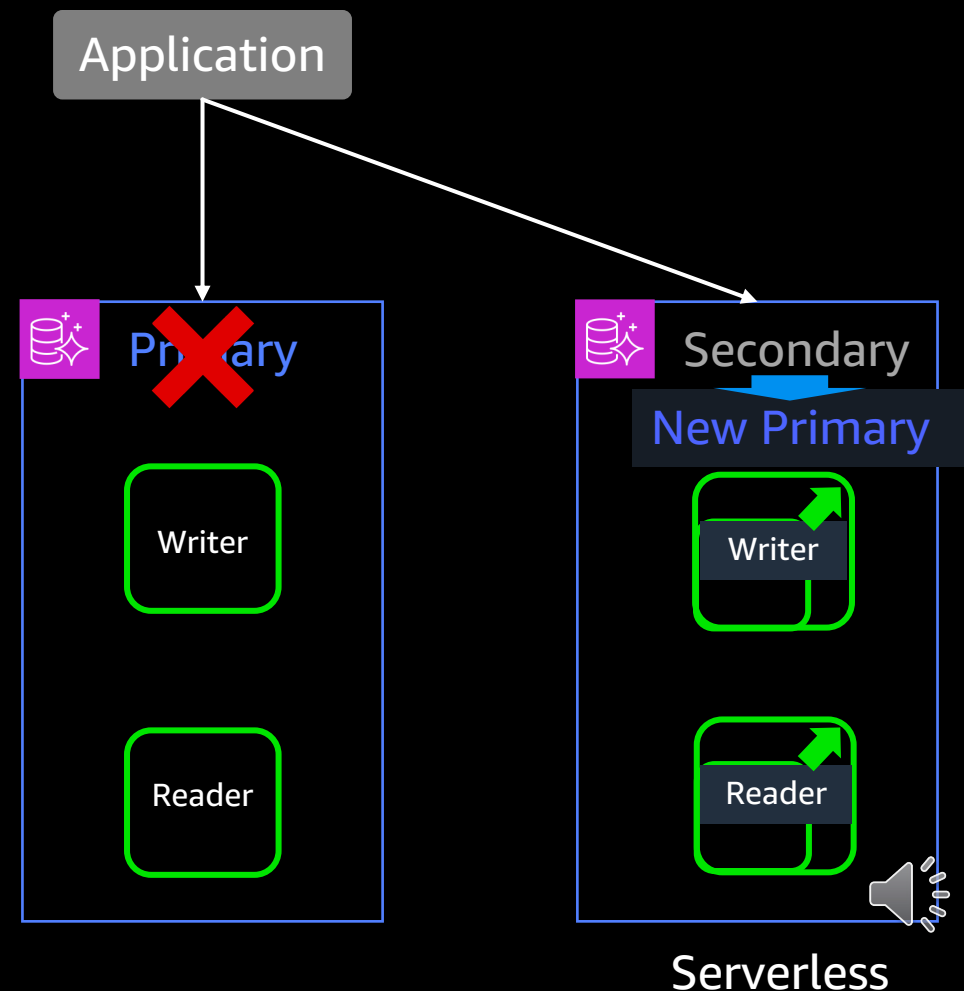
- Writer のキャパシティは Provisioned(Reserved Instance含む)で運用
- Reader のキャパシティは Serverless v2 で運用し、Reader がアイドル時は最小キャパシティ分のみの支払いにとどめる

* 混在環境でも Reader(Serverless v2) を Tier0, 1 とした場合、Reader は Writer と同等のリソースとなるよう自動調整されます

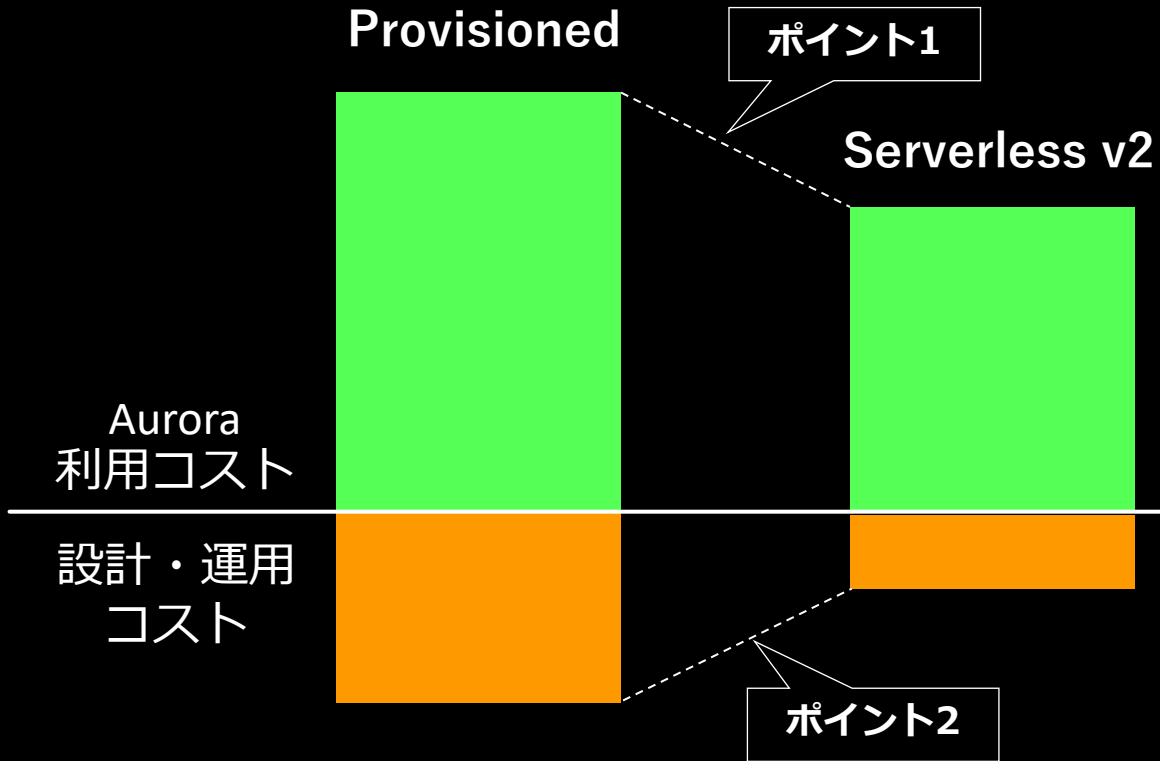


Serverless Global Database 構成によるコスト最適化

- Aurora Serverless v2 によるセカンダリーリージョンの構成
- **セカンダリーリージョンがアイドル時は最小キャパシティ分のみの支払いにとどめる**
(従来のヘッドレス構成よりオペレーションが自動化され、サービス再開が高速)
- フェイルオーバー後のライターとリーダーのスケールアップは Serverless v2 の特徴を利用
- エンドユーザーの地域に近い読み取り処理のためにセカンダリーリージョンは独立してスケールさせる



コスト観点で Serverless v2 導入を検討するポイント



ポイント1

- ワークロードによっては Provisioned の方がコストメリットがある場合もある
- ワークロードの特性を見極めた上で、Serverless v2 利用によるコストメリットの有無を確認する
- 机上でのワークロード特性の見極めが困難な場合は、PoCを実施した確認が有効

ポイント2

- Serverless v2 により、データベースインスタンスのキャパシティ管理コストを抑制することができるため、どの程度の設計・運用コストが削減できるか？という観点で検討する
- 設計・運用コストだけでなく、人的リソースにおける負荷軽減という観点でも検討する価値あり



* 上記図は実際のコストの比率を図示したものではありません。イメージとして捉えてください。

Aurora I/O 最適化の概要

クラスターストレージ設定 情報

アプリケーションの料金予測可能性と料金パフォーマンスのニーズに最適な Aurora DB クラスターのストレージ設定を選択します。

設定オプション

データベースインスタンス、ストレージ、I/O の料金は、設定によって異なります。詳細はこちら [🔗](#)

☒ Aurora I/O 最適化

- すべてのアプリケーションの予測可能な料金設定。I/O の多い (I/O コスト > データベース全体のコストの 25%) アプリケーションの料金パフォーマンスが向上。
- 読み取り/書き込み I/O 操作には追加料金はかかりません。DB インスタンスとストレージの料金には I/O 使用量が含まれています。

☐ Aurora スタンダード

- I/O 使用量が中程度 (I/O コスト < 25% of total database costs) の多くのアプリケーション向けの費用対効果の高い料金設定
- リクエストごとの支払い I/O 料金が適用されます。DB インスタンスとストレージの料金には I/O 使用量は含まれていません。

- お客様はコンピューティングとストレージのみで、**読み取り/書き込み IO にコストがかからない**
- Aurora の多様なワークロードにおいて**コストの予測が容易に**
- I/O コストが Aurora データベースの総コストの **25%** を超えている場合、最大 40% のコスト削減が期待できる
- リザーブドインスタンス (RI) を使用すると、コストをさらに節約可能
- Aurora クラスタは **30 日に 1 回、ストレージオプション (スタンダード⇔I/O 最適化) を変更**する事が可能
- Aurora Serverless V2, Provisioned, リザーブドインスタンス等での利用が可能



Aurora I/O 最適化のコスト

東京リージョンにおける I/O 最適化とスタンダードのコスト比較 (2025 年 5 月現在)

ストレージ設定	コストカテゴリ	費用
Aurora スタンダード	インスタンス	db.r5.xlarge : \$ 0.70/時間
	ストレージ	\$ 0.12/毎月の GBあたり
	I/O	\$ 0.24/100万リクエスト
Aurora I/O 最適化	インスタンス	db.r5.xlarge : \$ 0.91/時間
	ストレージ	\$ 0.27/毎月の GBあたり
	I/O	利用料に含まれる

Aurora スタンダードの +30%

Aurora スタンダードの +125%



Aurora I/O 最適化のコスト試算

シミュレーション 例1) ～IO 料金が 50% を占めている場合～

- Aurora スタンドでは、クラスターのコンピューティング料金は \$ 1,000、ストレージ料金は \$ 500、IO 料金は \$ 1,500 と仮定する(合計: **\$ 3,000**)
- Aurora I/O 最適化でも、データベーストラフィックに変化がないと仮定する
 - コンピューティング料金は 約 1.3 倍になるため、お客様は $1.3 \times \$ 1,000 = \$ 1,300$ の支払い
 - ストレージ料金は 約 2.25 倍になるため、お客様は $2.25 \times \$ 500 = \$ 1,125$ の支払い
 - IO 料金はゼロ (合計: **\$ 2,425**)

Aurora Standard	Aurora I/O-Optimized
\$ 3,000	\$ 2,425

※ この例では、Aurora I/O 最適化にすることで、コストが低減することがわかる

シミュレーション 例2) ～IO 料金が 20% を占めている場合～

- Aurora スタンドでは、クラスターは \$ 500 のコンピューティング料金、\$ 100 のストレージ料金、IO 料金は \$ 150 と仮定する(合計: **\$ 750**)
- Aurora I/O 最適化でも、データベーストラフィックに変化がないと仮定する
 - コンピューティング料金は 約 1.3 倍になるため、お客様は $1.3 \times \$ 500 = \$ 650$ の支払い
 - ストレージ料金は 約 2.25 倍になるため、お客様は $2.25 \times \$ 100 = \$ 225$ の支払い
 - IO 料金はゼロ (合計: **\$ 875**)

Aurora Standard	Aurora I/O-Optimized
\$ 750	\$ 875

※ この例では、Aurora I/O 最適化 にすることで、コスト予測の観点におけるメリットはあるが、コストが上昇することがわかる

ストレージ費用削減

- クラスターボリュームのサイズは 1 時間ごとに評価され、ストレージコストが決定される
 - 東京リージョンの Aurora Standard のストレージ料金は \$ 0.12/毎月の GB あたり
- 課金対象となるのはそのボリュームの使用した領域分のみ
- **不要になったテーブルやデータベースを削除することでストレージ料金を削減**することができる
 - DROP TABLE、DROP DATABASE、TRUNCATE TABLE、ALTER TABLE、DROP PARTITION 等でリソース削除が可能
 - **DELETE の場合、レコード削除後にテーブル再編成を実行**することによりストレージ容量削減が可能
 - Aurora MySQL : OPTIMIZE TABLEの実行
 - Aurora PostgreSQL : pg_repack エクステンションの実行



Amazon ElastiCache との併用

- Amazon ElastiCache と併用する事で、Aurora に対する読み込みパフォーマンスの向上とコスト削減につなげる事が出来る
- Aurora のコンソール画面からも、データベースの設定を継承した ElastiCache クラスターの設定が可能

ElastiCache クラスターの作成: database-1 情報

ソース DB から設定を継承する互換性のある ElastiCache クラスターを作成します。RDS だけで実行するよりも、RDS で ElastiCache を使用すると、コストを節約し、読み取りパフォーマンスを向上させることができます。*例: RDS for MySQL で ElastiCache を使用すると (MySQL だけの RDS と比較して)、コストを最大 55% 節約し、読み取りパフォーマンスを最大 80 倍高速化できます。

ElastiCache クラスター設定 情報

ソース DB 識別子
database-1

クラスタータイプ

☒ Redis

☐ Memcached

☐ Valkey

デプロイオプション

☒ サーバーレスキャッシュ - 新規

これを使用すると、アプリケーショントラフィックの需要に合わせて自動的にスケーリングするキャッシュをすばやく作成でき、サーバーを管理する必要はありません。

☐ 独自のキャッシュの設計

これを使用して、ノードタイプ、サイズ、数値を選択してキャッシュを作成できます。

キャッシュ設定

キャッシュの名前と説明 (オプション) を入力します。

名前

名前

名前は最大 40 文字を使用でき、スペースを含めることはできません。

説明 - オプション

説明

説明には 255 文字まで入力でき、< and > 文字を含めることはできません。



まとめ

基本編

- Amazon Aurora のコスト構造
- インスタンスのダウンサイジング
- インスタンスタイプの変更
- Reserved Instance の購入
- 使用しないクラスター (インスタンス) の停止や削除

応用編

- Aurora Serverless V2
- Aurora I/O 最適化
- ストレージ費用削減
- Amazon ElastiCache との併用

① Amazon Aurora の利用料を確認頂き、費用傾向を確認しましょう

② コストコンポーネントに応じたモニタリングと対策を継続的に実施しましょう

Thank you!

