



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>



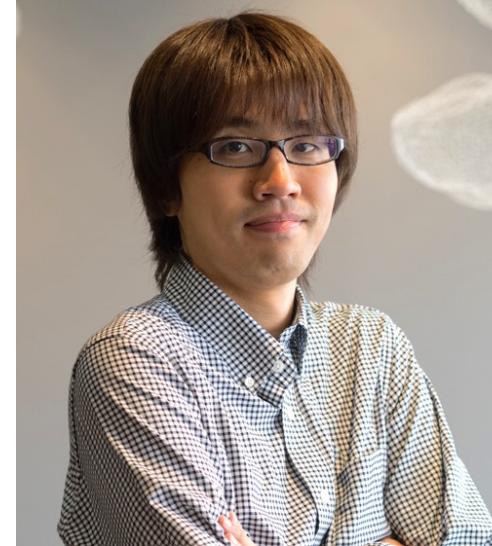
過去資料
<https://amzn.to/JPArchive>



自己紹介

星野 豊 (ほしの ゆたか)

- Aurora/RDS (MySQL) Specialist SA



経歴

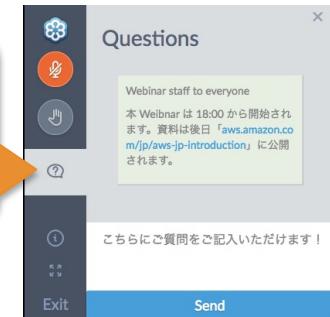
- 全てオンプレ環境のインフラエンジニア
- 全てAWS環境のインフラエンジニア



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、Amazon Web 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います。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)

選択肢

価値

イノベーション

Amazon RDS

様々オープンソース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ィ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



**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



**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



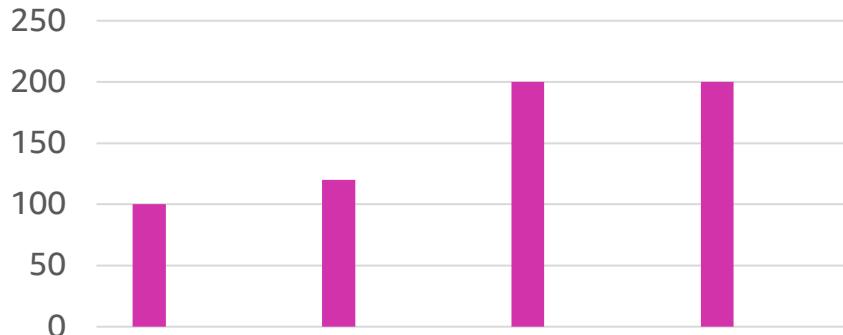
**Amazon
CloudWatch**

システムメトリックスやログを
CloudWatchへアップロード

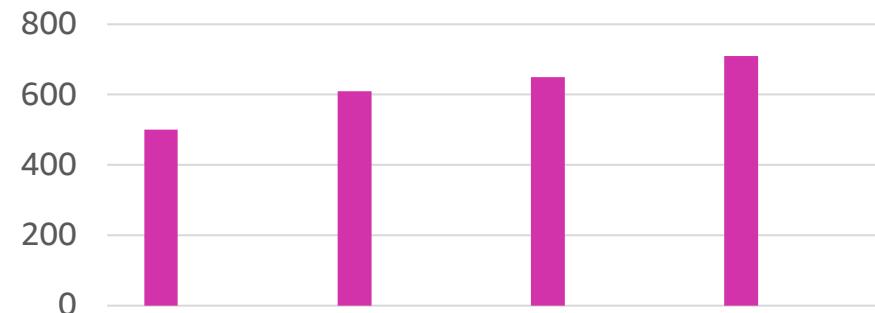
Performance improvement over time

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

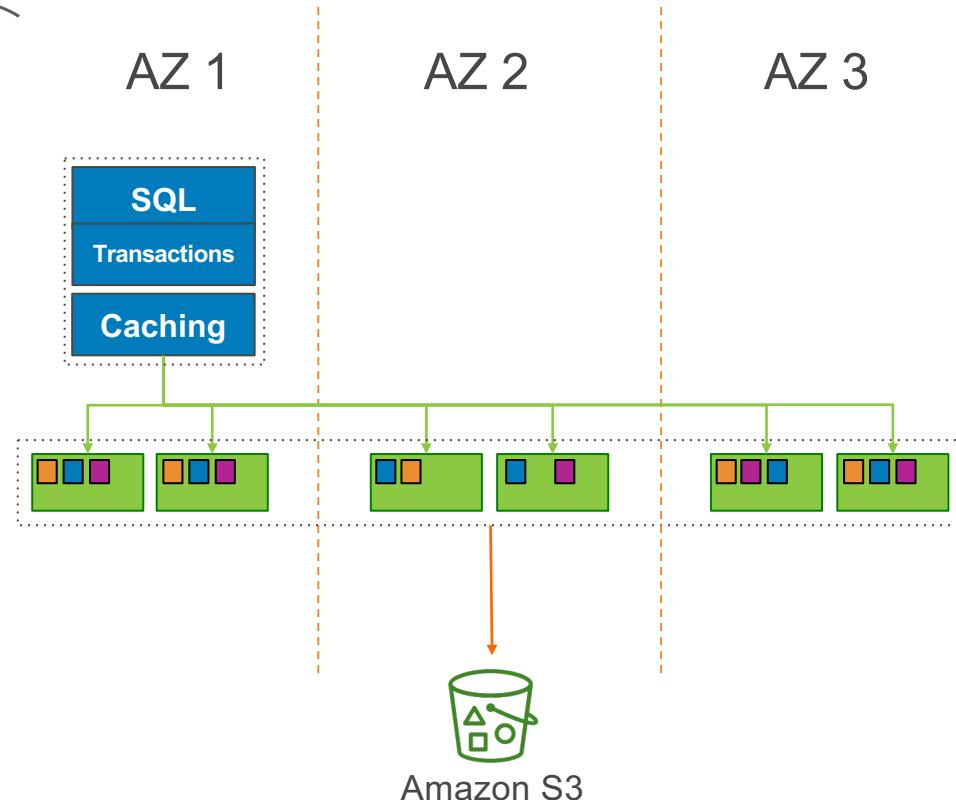
Launched with R3.8xl
32 cores, 256GB memory

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

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

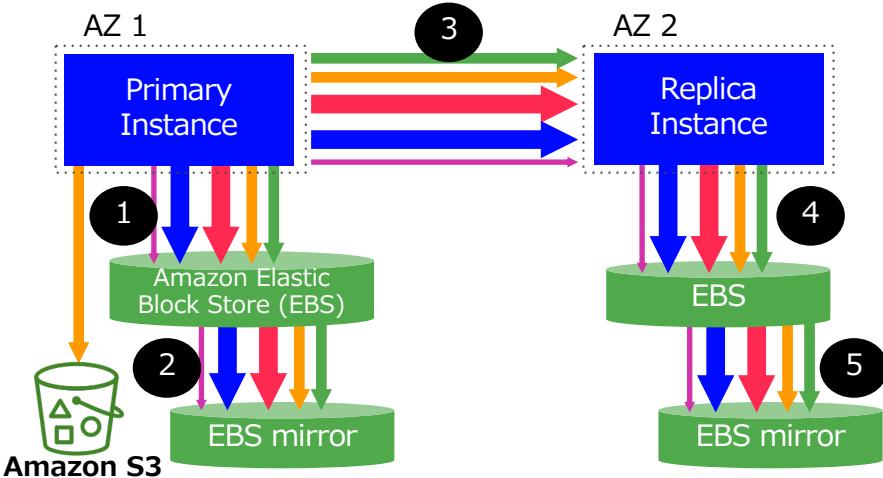
Auroraのストレージ

- 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しまっても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



Aurora I/O profile

MySQL with Replica



MySQL I/O profile for 30 min Sysbench run

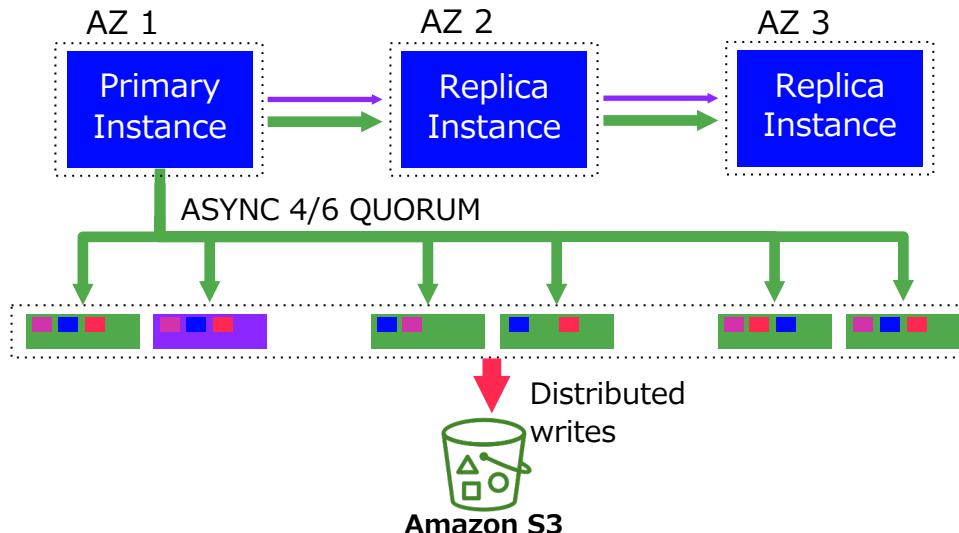
780K transactions

7,388K I/Os per million txns (excludes mirroring, standby)

Average 7.4 I/Os per transaction

→ Log → Binlog → Data

Amazon Aurora



Aurora IO profile for 30 min Sysbench run

27,378K transactions – **35X MORE**

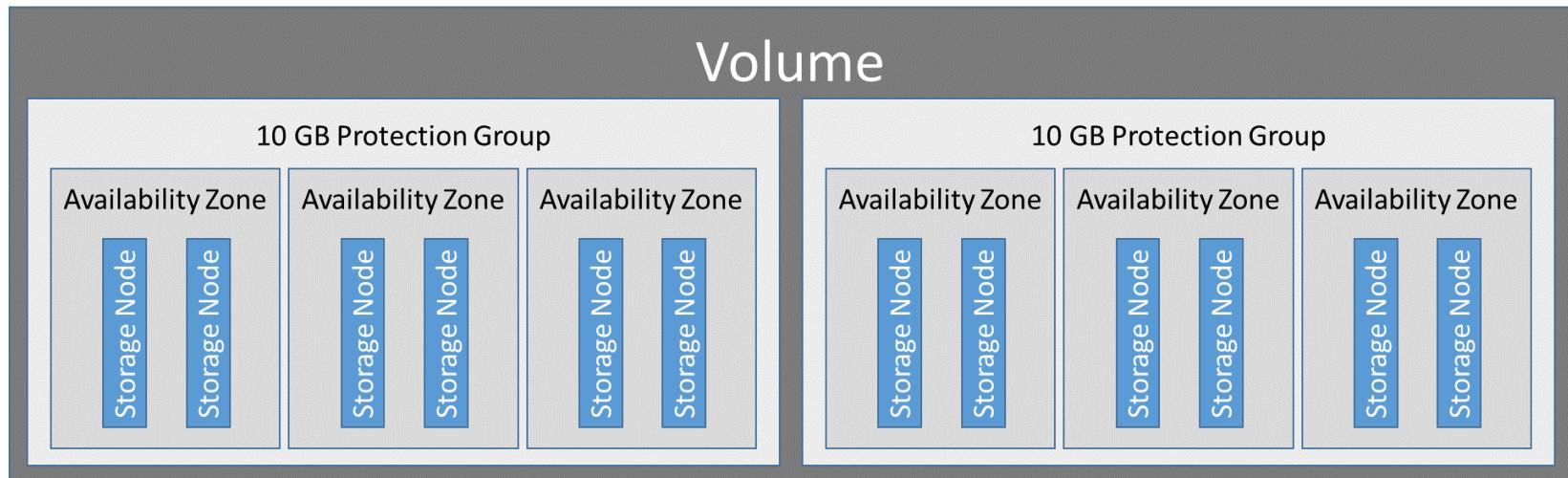
0.95 I/Os per transaction (6X amplification) – **7.7X LESS**

→ Double-write → From files



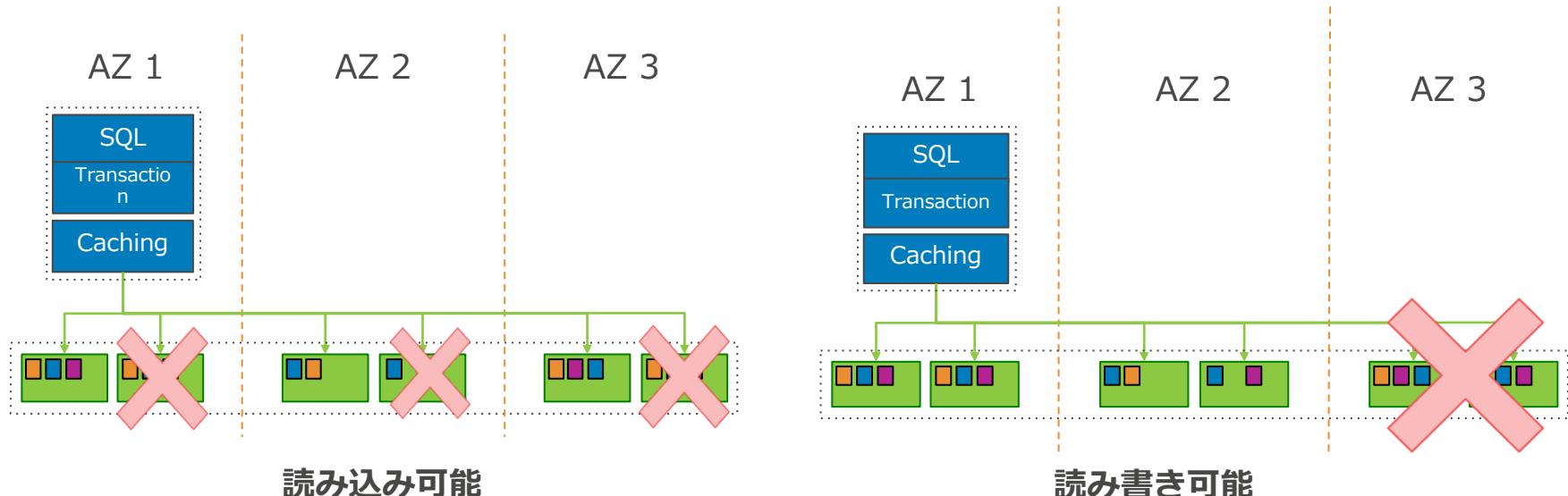
ストレージノードクラスタ

- 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



クオーラムモデル

- レプリケーション管理のためのクオーラムのルール

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

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



クオーラムモデル

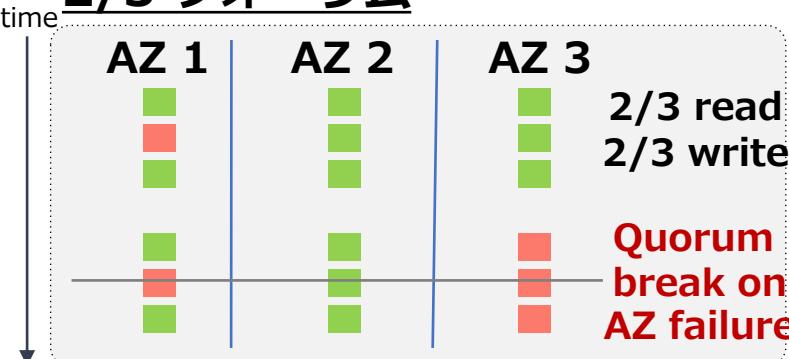
- レプリケーション管理のためのクオーラムのルール

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

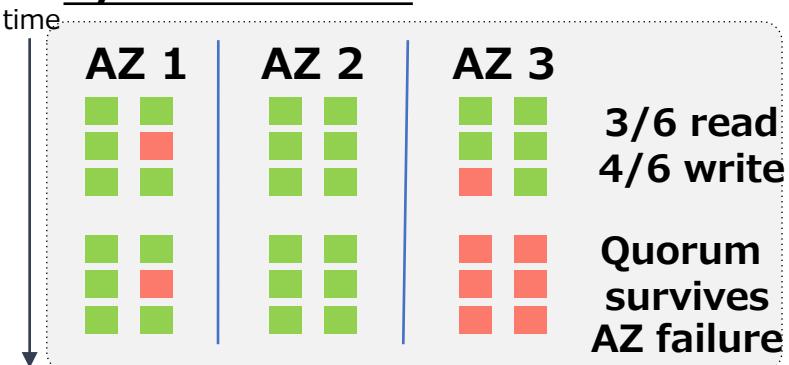
V (コピーの数)	V_w (書き込みクオーラム)	V_r (読み込みクオーラム)
1	1	1
2	2	1
3	2	2
4	3	2
5	3	3
6	4	3
7	4	4

なぜ6つのコピーが必要なのか？

2/3 クオーラム



4/6 クオーラム



- 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のクオーラムグループがノードGを用いて作成され、両方のクオーラムがアクティブ

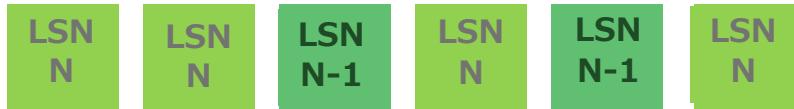


Epoch 3: ノードFはunhealthyであることが確認されます。ノードGを持つ新しいクオーラムグループがアクティブになる

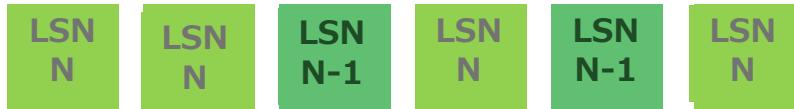
- ほとんどのシステムは、メンバーシップの変更にコンセンサスを使用しますがジッタとストールを引き起こす

- Auroraはクオ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

クオーラムの効率化



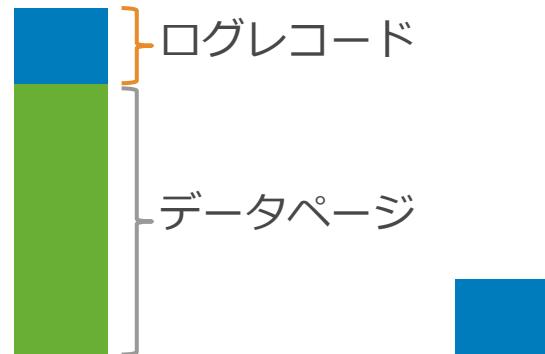
各データブロックに対して、クオーラムグループ内の少なくとも4つのノードが最新のデータを持つ



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

- ほとんどのクオ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
- リードクオーラムは、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

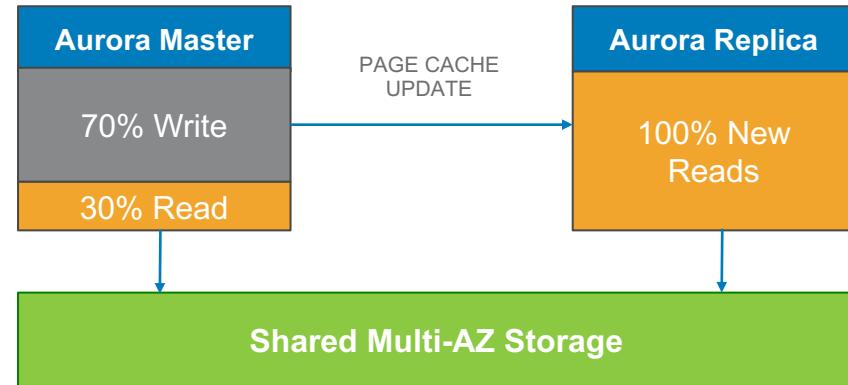
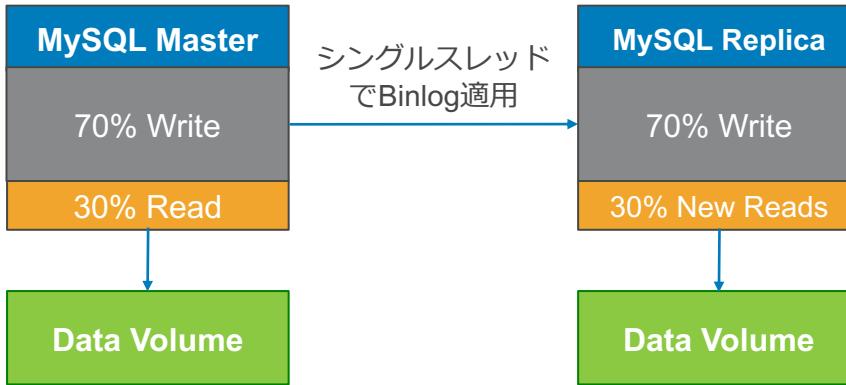


- 6つのコピーは全て同一ではない
 - 3つのフルセグメント：データページとログレコード
 - 3つのテールセグメント：ログレコードのみ
- 6つのコピーが必要だが、物理ストレージとして必要な容量は、6倍ではなく、3倍より少し多い程度

※ 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

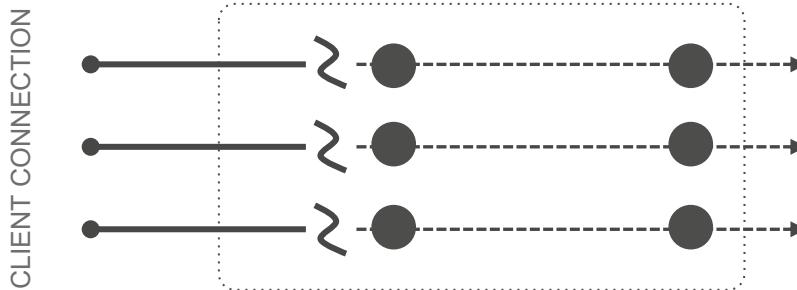
- レプリケーションにはbinlog / relay logが必要
- レプリケーションはマスターへ負荷がかかる
- レプリケーション遅延が増加していくケースがある
- フェイルオーバでデータロスの可能性がある

Amazon Aurora read scaling

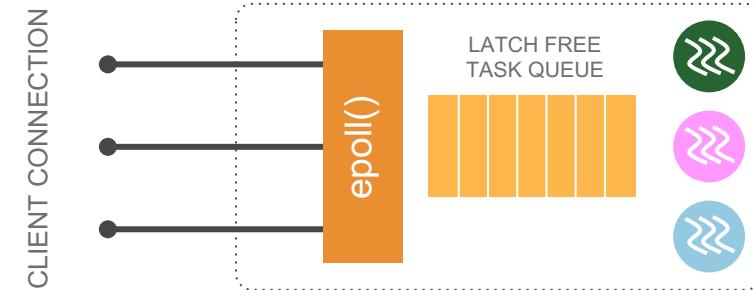
- Binlogによるレプリケーションではない
- マスターへの負荷を最小限に15台までリードレプリカを作成可能
- 10-20ms程度のレプリケーション遅延
- フェイルオーバでデータロスが無い

アダプティブスレッドプール

MYSQL THREAD MODEL



AURORA THREAD MODEL

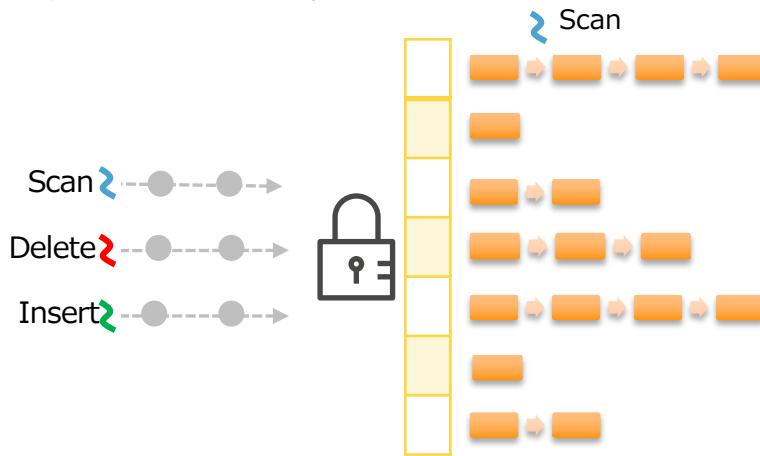


- Standard MySQL – コネクション毎に1
- コネクション数に応じてスケールしない
- MySQL EE – スレッドプール毎にコネクションをアサイン
- しきい値を慎重に設定する必要がある
- アクティブなスレッドに複数のコネクションを収容
- Kernel空間の `epoll()` がラッチフリーキューにコネクションをアサイン
- スレッドプールの数は動的に調整される
- r3.8xlargeインスタンスのAmazon Auroraで5,000同時コネクションを扱える

Aurora lock management

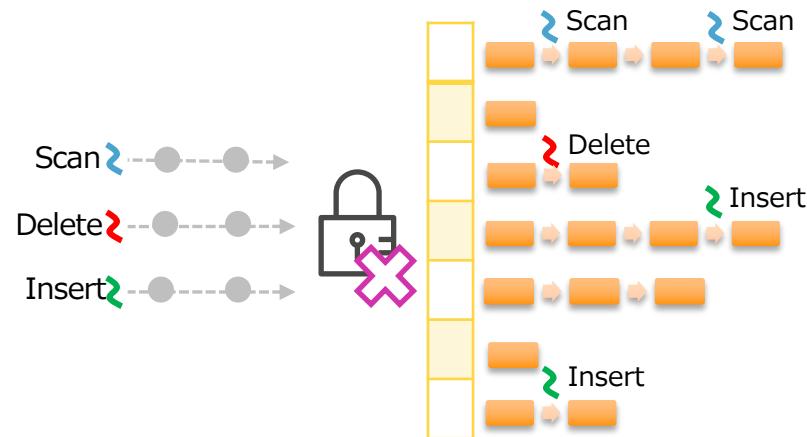
MySQL lock manager

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



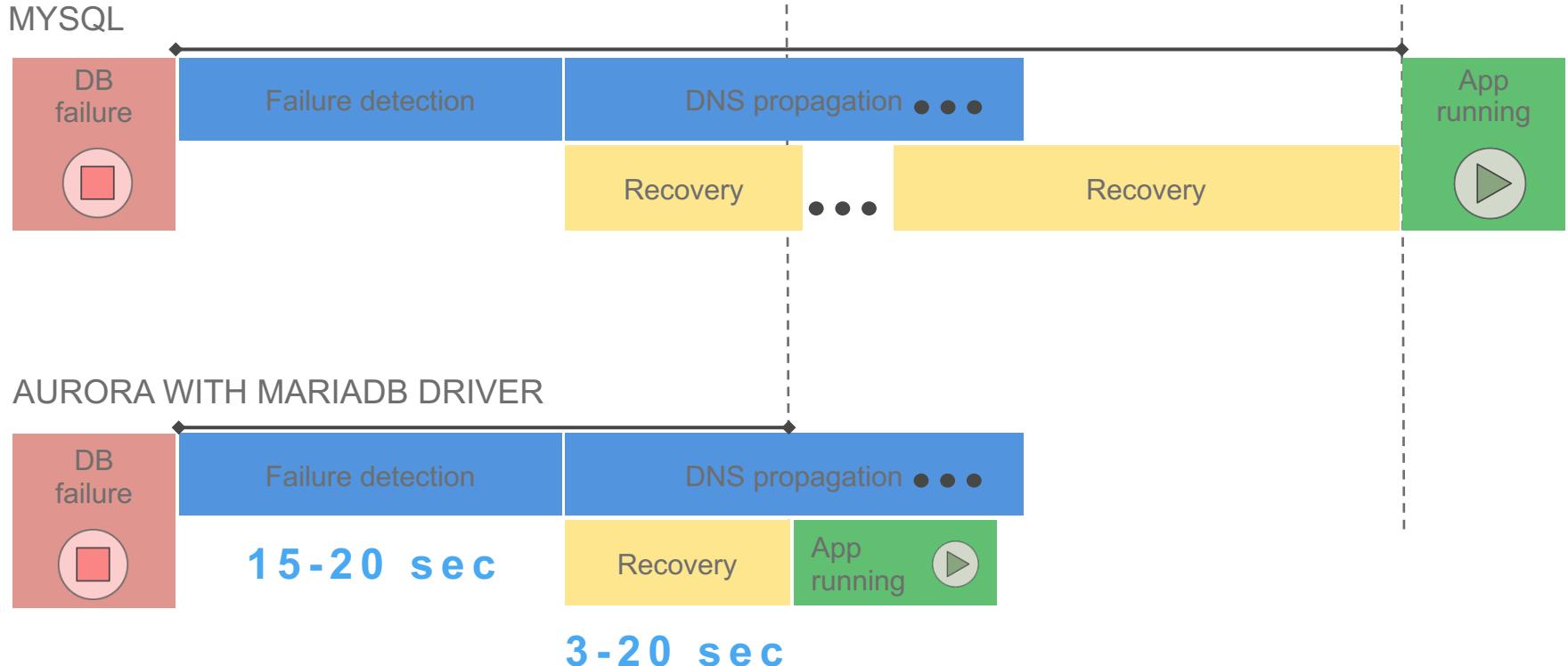
Aurora lock manager

個々のロックチェーン内の複数のスキーナ
ロックフリーデッドロック検出



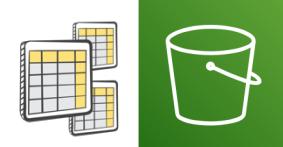
多くの同時セッションをサポートする、高い更新スループット

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



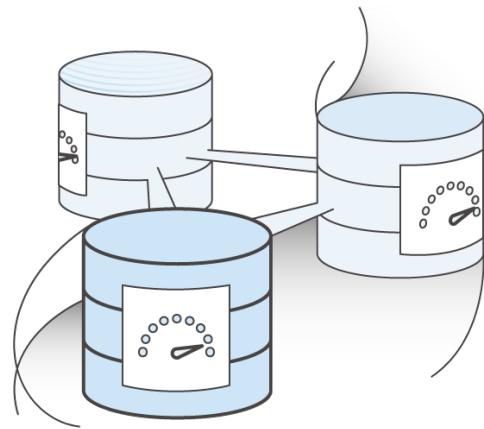
SQLによるフェイルオーバのテスト

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



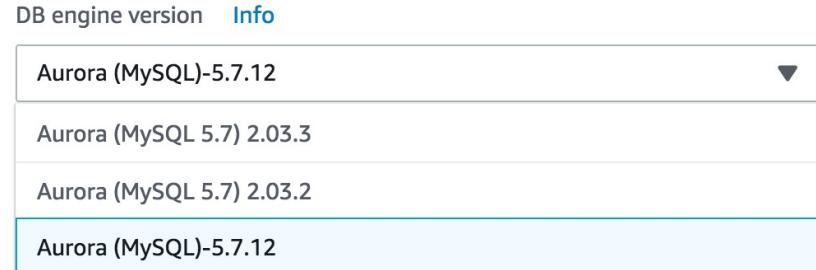
– **TotalBackupStorageBilled:**

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 $100 \times (100 \text{ GiB} + 100 \text{ GiB} - 100 \text{ GiB})$ となります

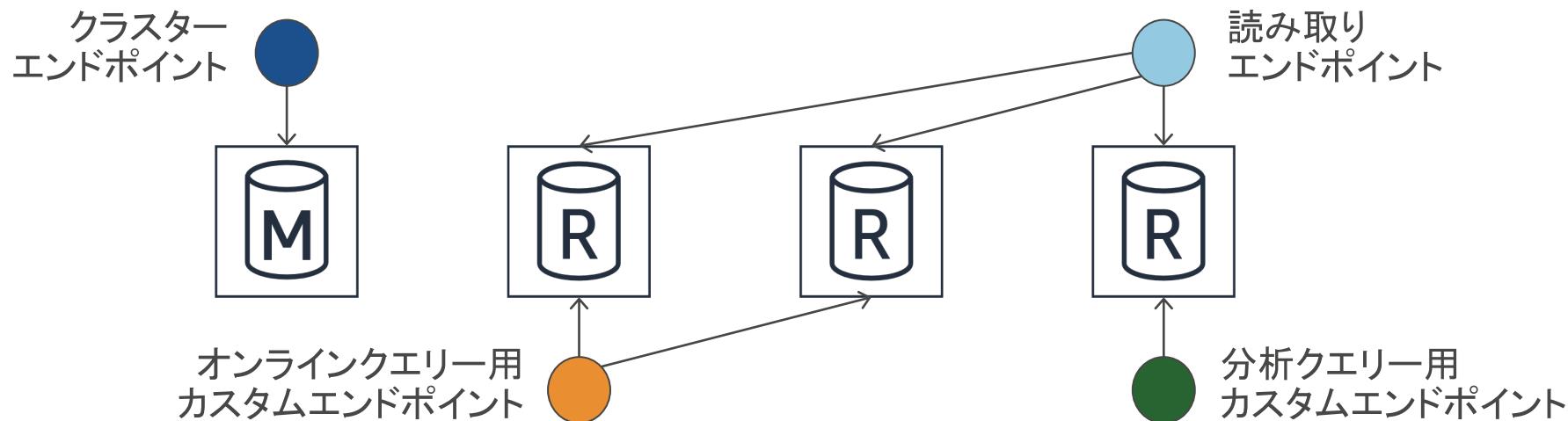
起動バージョン・バージョンアップ対象の指定方法改善

- 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<ModifyDBInstanceでEngineVersion指定に
 - 例: --engine-version **5.7.mysql_aurora.2.03.2 or 5.6.mysql_aurora.1.19.0**

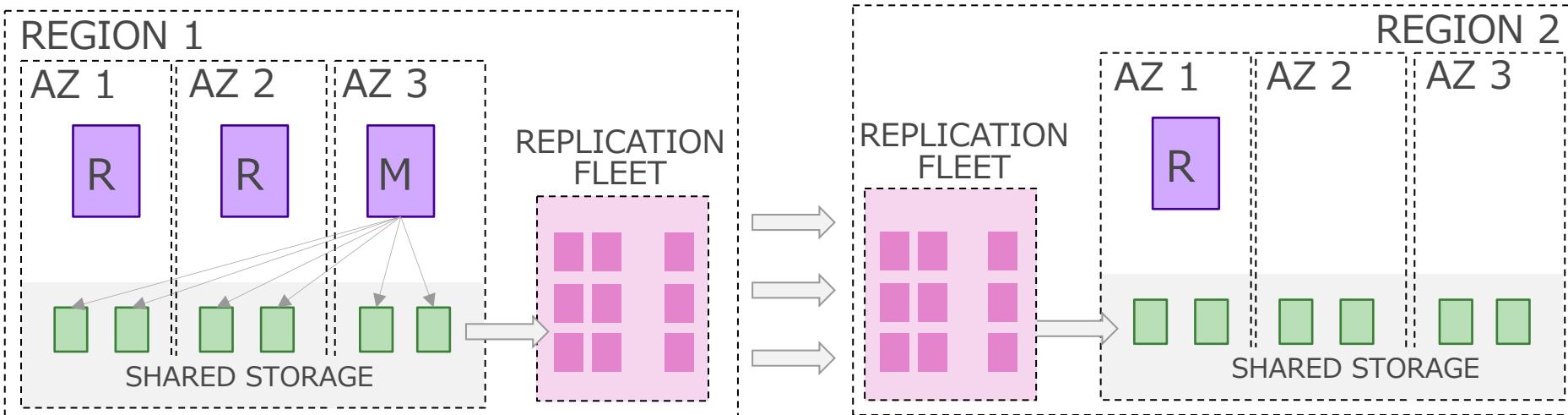


Custom Endpoints

- 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



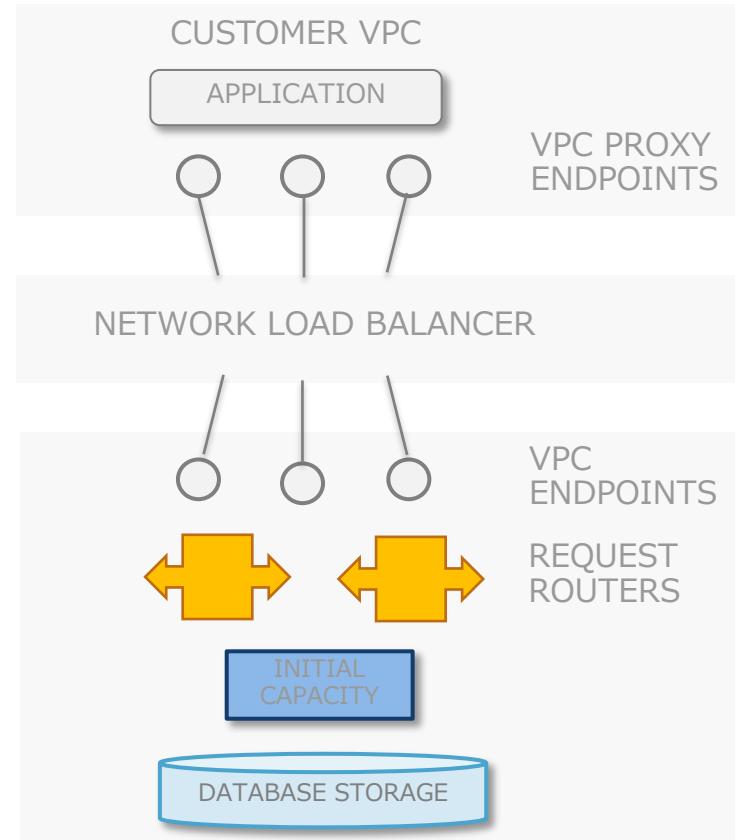
Aurora Serverless

- 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



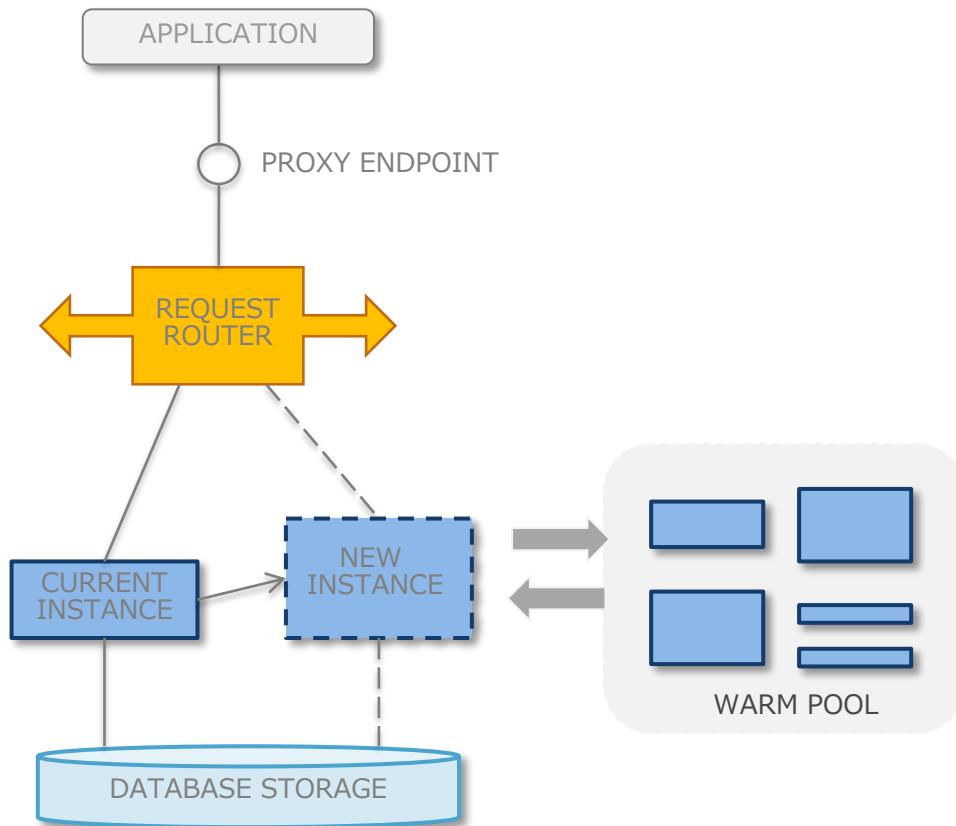
Serverless Endpoint Provisioning

- 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



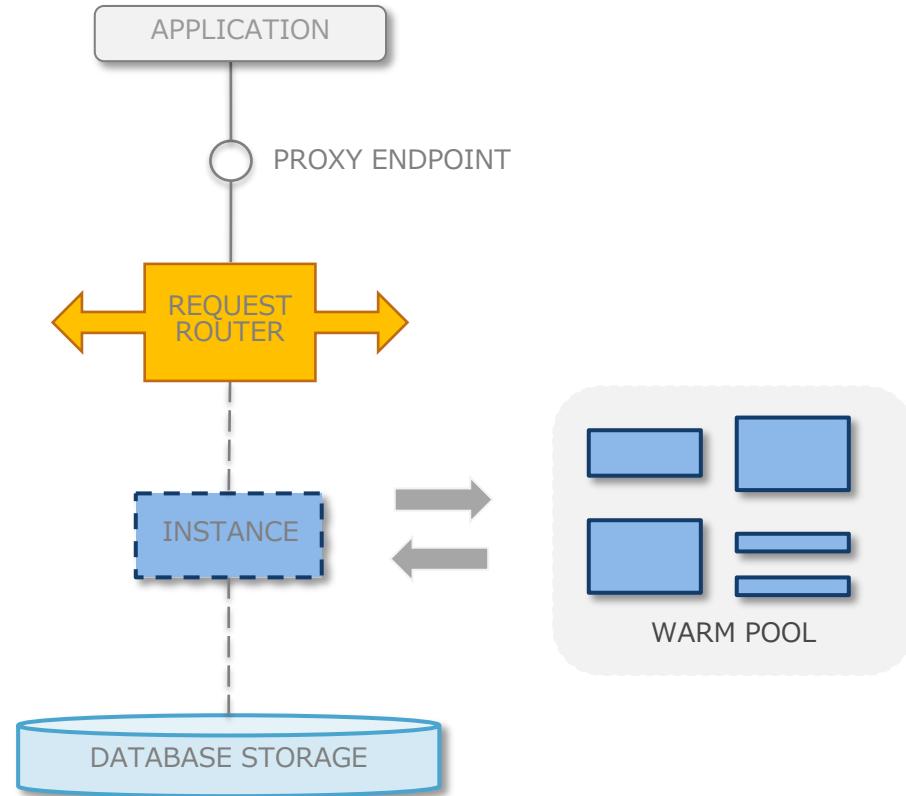
スケールアップ・ダウソ

- 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

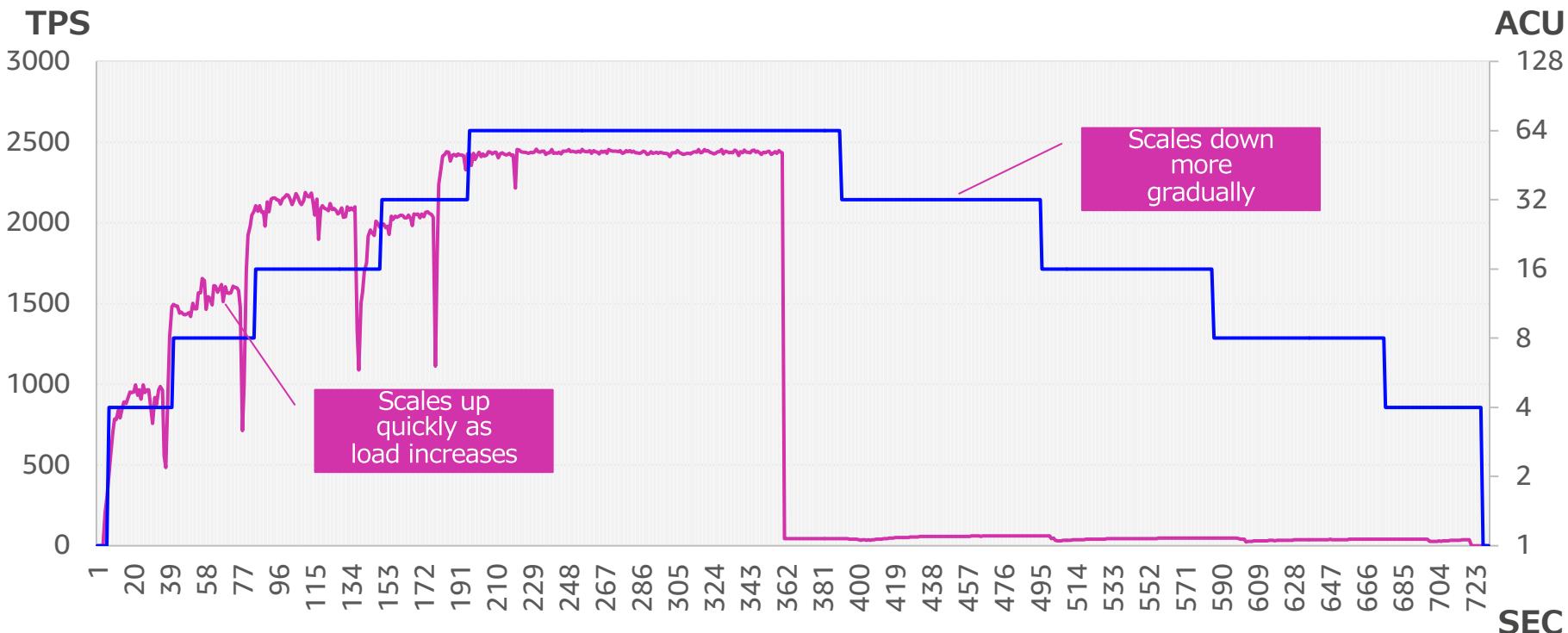


0からスケール

- 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へのスケーリング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・ダウソ



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

- Amazon Aurora Serverlessクラスタに直接マネー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 shows the AWS RDS Query Editor interface. On the left is a sidebar with navigation links: Dashboard, Instances, Clusters, **Query Editor** (which is highlighted in orange), Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations. The main area has tabs for Editor, Recent, and Saved queries. The Editor tab is selected, showing the following SQL code:

```
1: SELECT emp_no, first_name, last_name FROM employees
2: WHERE hire_date > '1994/01/01';
```

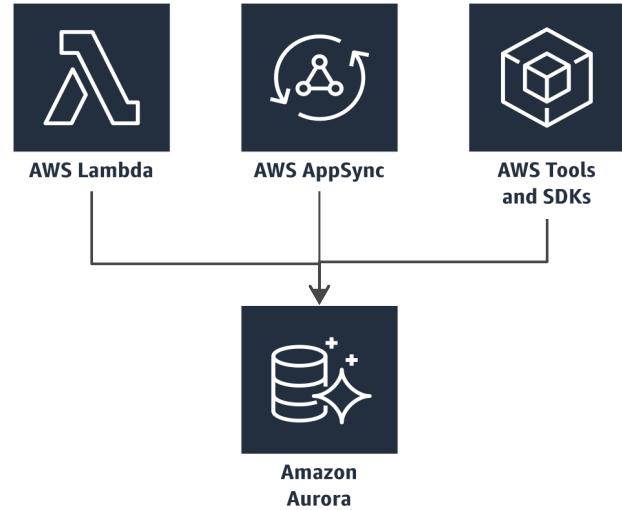
Below the code are buttons for Run, Save, and Clear. The Run button is highlighted in orange. To the right, it says "Connected instance: http-endpoint-test" and "Change database". Under the Run button, there are tabs for Output and Result set 1 (12). The Output tab is selected, displaying a table titled "Rows returned (12)". The table has columns: emp_no, first_name, last_name, and hire_date. The data is as follows:

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	Petkey	1997-05-19

At the bottom right of the table are buttons for Export to csv and a refresh icon. There are also navigation arrows and a search bar labeled "Search rows".

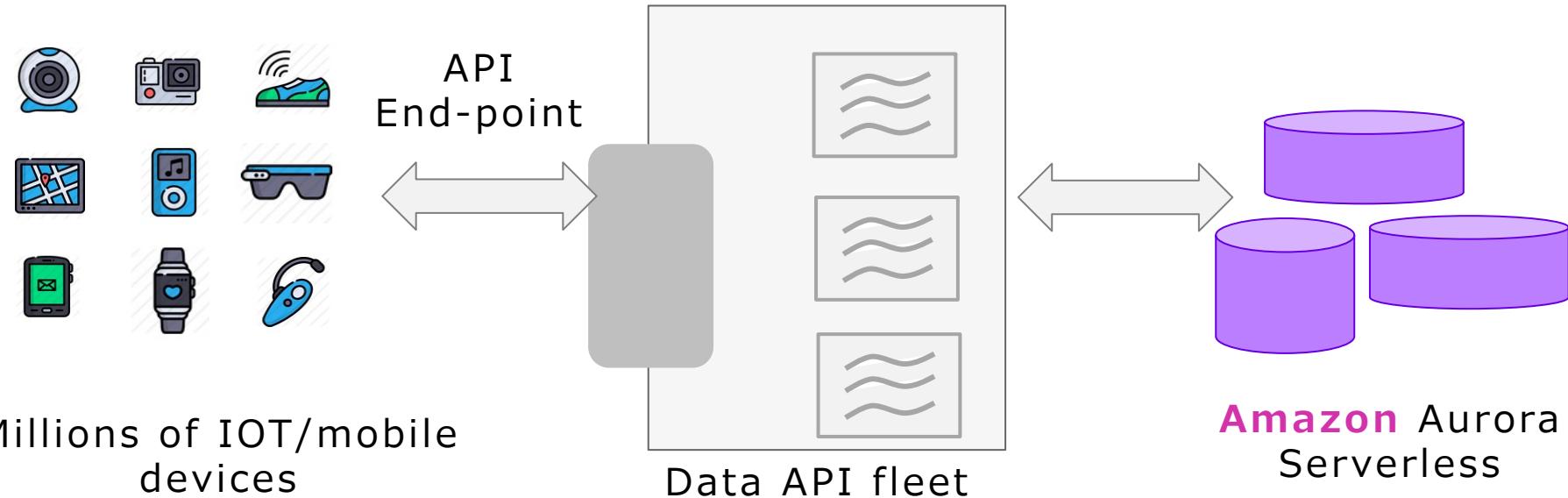
Amazon Aurora Serverless Database with the New Data API (Beta)

- 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

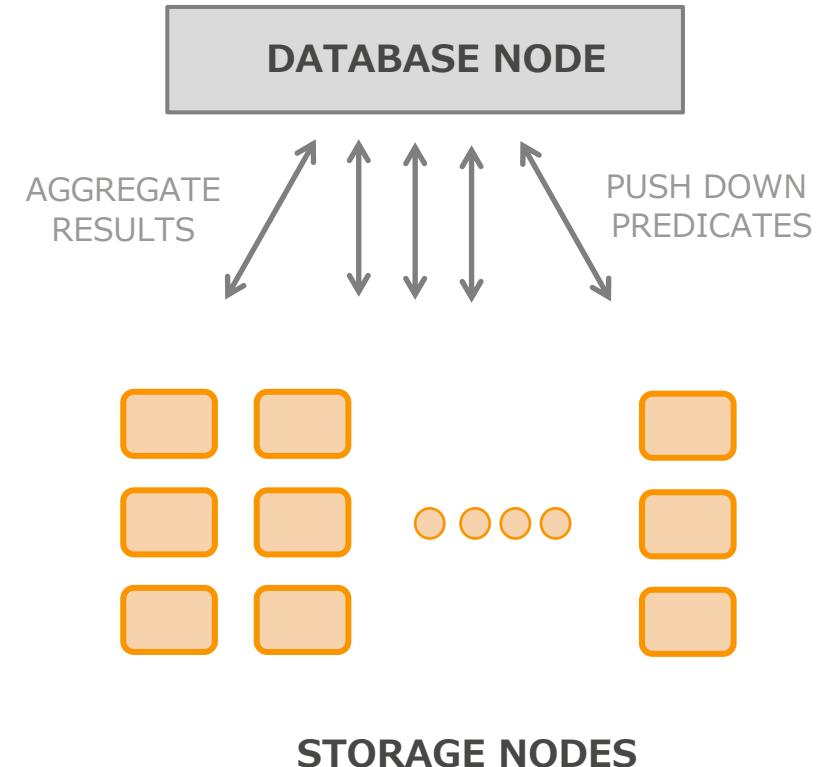
Reduced contention
with OLTP workload

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

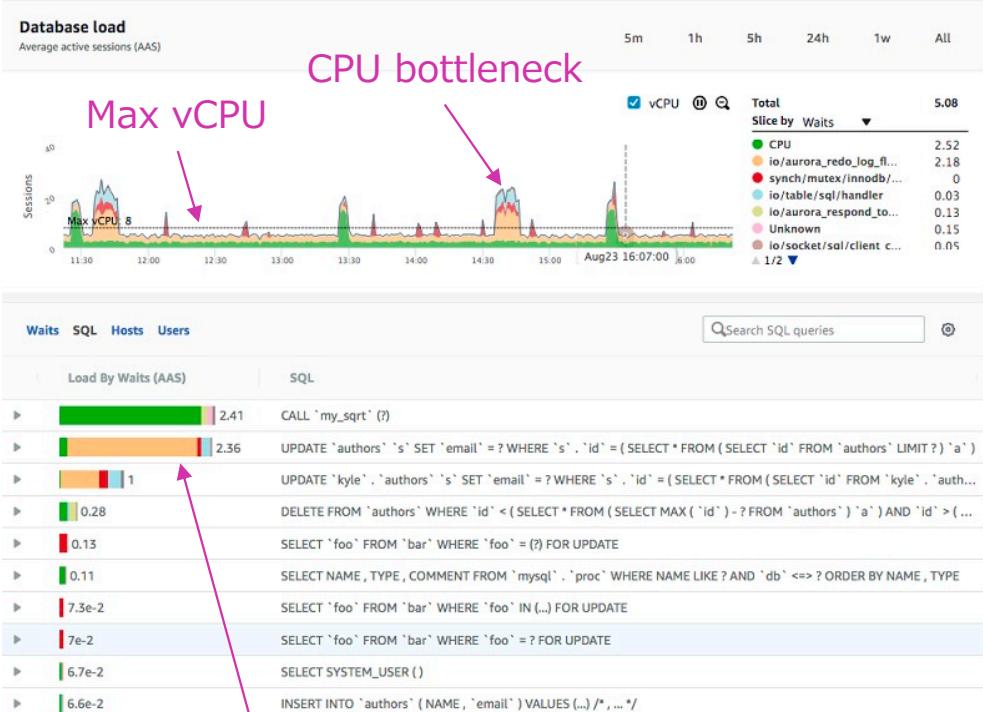
- 注意点

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

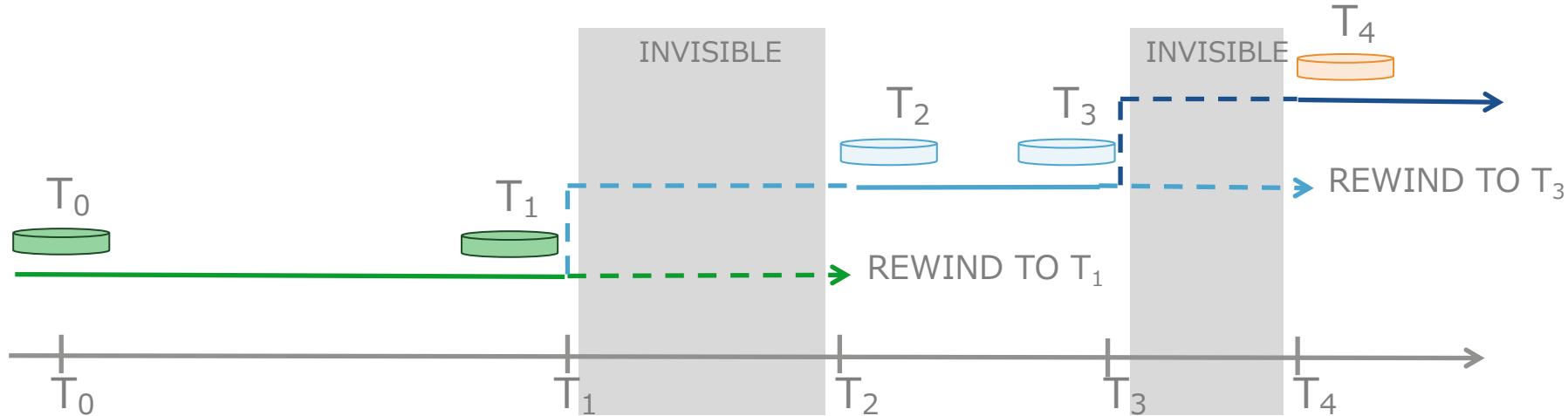
- `performance_schema=1`
- `performance-schema-consumer-events-waits-current=ON`
- `performance-schema-instrument='wait/%=ON'`
- `performance-schema-consumer-global-instrumentation=ON`
- `performance-schema-consumer-thread-instrumentation=ON`

- 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、ご注意下さい
- `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



Backtrack

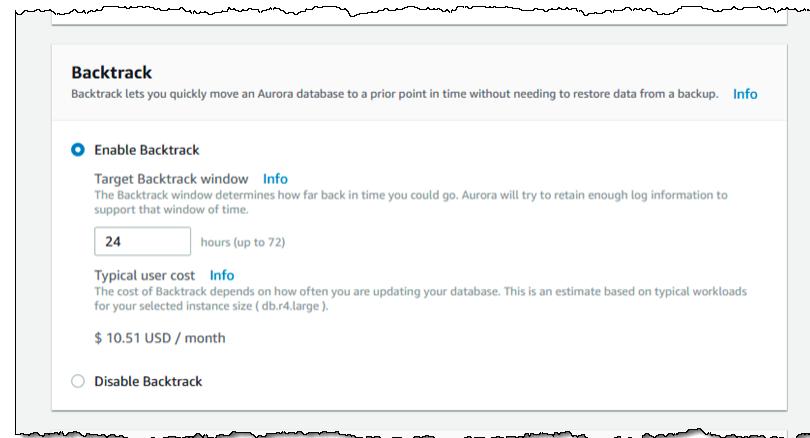


データベースの状態を容量によらず瞬時に特定の時点へ巻き戻す

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

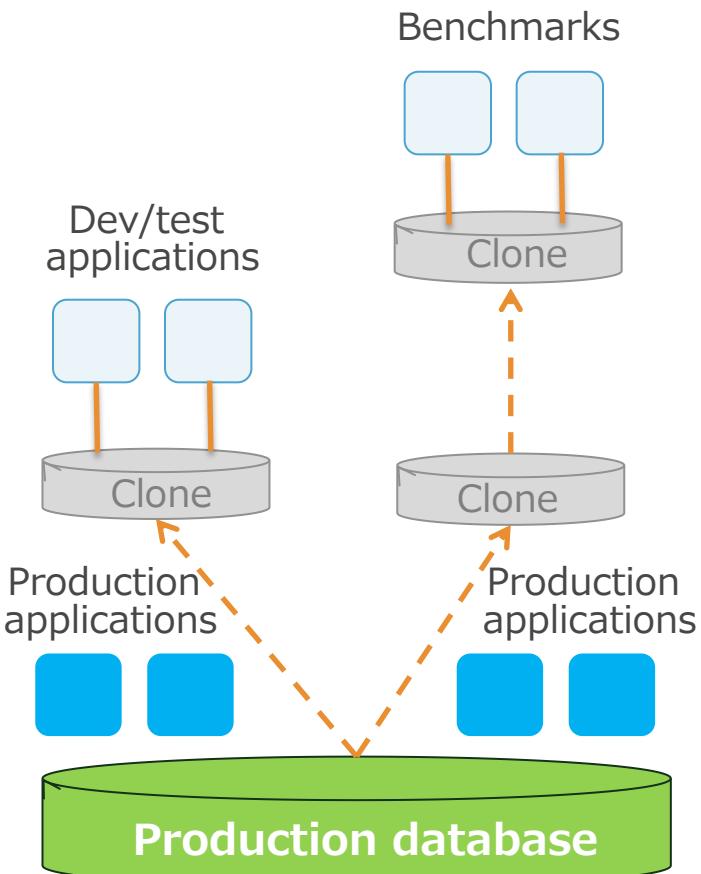


Database cloning

- 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

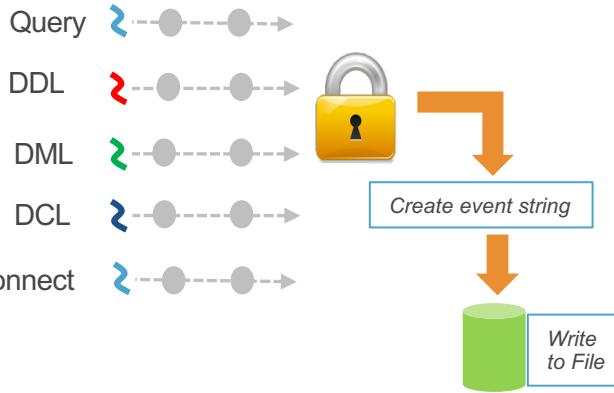
ユースケース

- 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

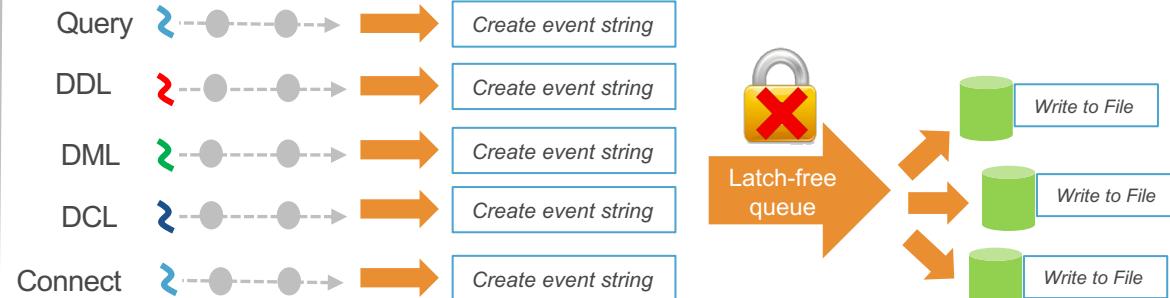


Advanced Auditing

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

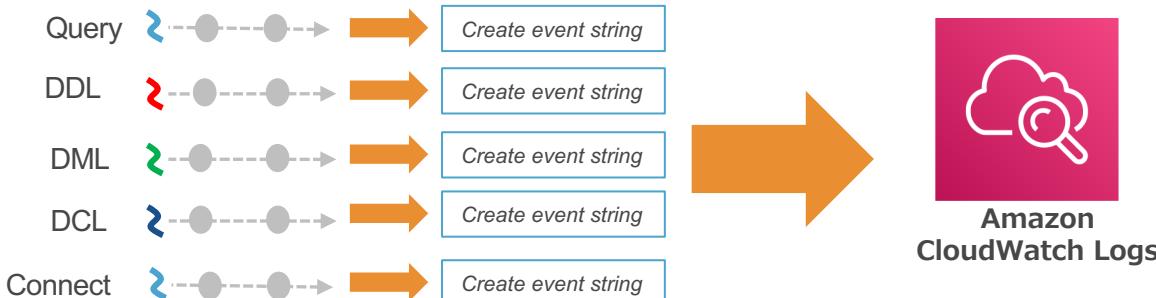
Sysbench Select-only Workload on 8xlarge Instance

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

Define Logs Metric Filter

Filter for Log Group: /aws/rds/cluster/pabertestlog2-1/audit

You can use metric filters to monitor events in a log group as they are sent to CloudWatch Logs. You can monitor and count specific terms or extract values from log events and associate the results with a metric. Learn more about pattern syntax.

Filter Pattern: "FAILED_CONNECT"

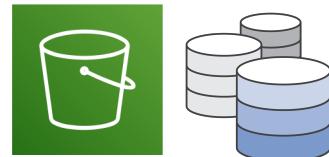
Select Log Data to Test: audit.log.2017-08-08-21-06.00

Results:

Please paste logs lines above and click Test Pattern.

Cancel Assign Metric

Load Data From S3



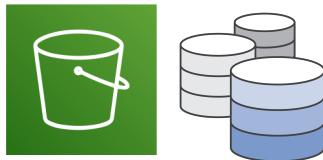
- S3バケットに保存されたデータを直接Auroraにインポート可能
 - テキスト形式(**LOAD DATA FROM S3**)・XML形式(**LOAD XML FROM S3**)
 - LOAD DATA INFILEとほぼ同様のオプションをサポート (圧縮形式のデータは現在未サポート)
 - Manifestによる一括ロードにも対応 (Version 1.11以降)

```
<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

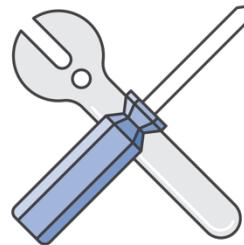


拡張モニタリング



- CloudWatch Logsにメトリクスを送信出来る
- CloudWatch 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

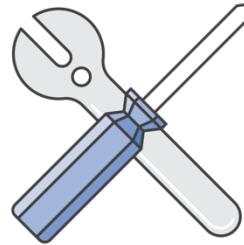
Lab mode



- 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>

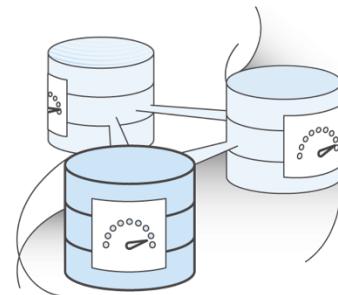
Amazon Auroraマイグレーションオプション

Source database	From where	Recommended option	
 MySQL	 PostgreSQL	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
 MySQL	 PostgreSQL	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
 PERCONA	 MariaDB		
 ORACLE	 Microsoft SQL Server	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
 mongoDB	 cassandra		
 IBM	 DB2		

まとめ

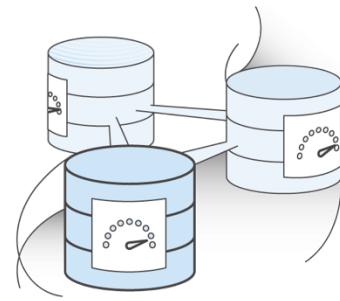


Amazon Aurora



- 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・PITRを実現するための多くのチャレンジ
 - Log Structured Storage
 - SOA

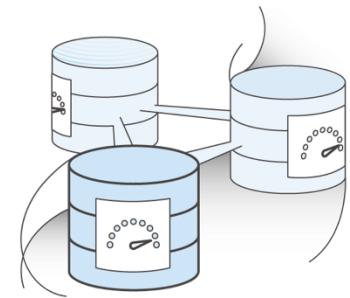
参考資料



- 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/>



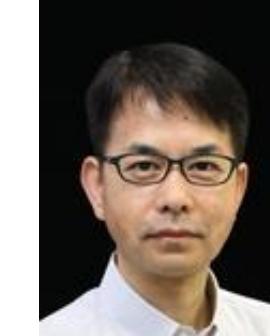
Q&A

- 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

The screenshot shows the homepage of the AWS Cloud Services Utilization Document Collection. The title 'AWS クラウドサービス活用資料集トップ' is displayed prominently. Below the title is a detailed description of AWS's capabilities and the purpose of the site. At the bottom, there are three call-to-action buttons: 'AWS Webinar お申込 »', 'AWS 初心者向け »', and 'サービス別資料 »'. The background features a network-like graphic with blue lines and dots.

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



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>





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 with PostgreSQL Compatibility

Archived

Solutions Architect 江川 大地
2019/8/28

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



過去資料
<https://amzn.to/JPArchive>



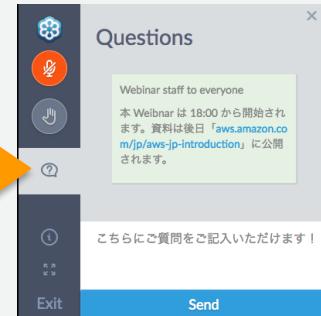
AWS Black Belt Online Seminar とは

「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。

質問を行うことができます！

- 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8月28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.

本日のアジェンダ

- Amazon Aurora とは
- 可用性と耐久性
- 性能と拡張性
- 運用管理
- 新機能
- まとめ

Amazon Relational Database Service(Amazon RDS) オープンソースDBとコマーシャルDBの多様な選択肢

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



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

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



RDSを使ったマネージドサービス
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

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ィ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

Aurora を構成するコンポーネント

Amazon Aurora DB クラスター

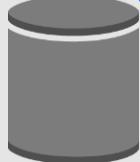
Availability Zone a

プライマリ
インスタンス
(Writer)



Writes
(書き込み)

Reads
(読み込み)



データコピー

Availability Zone b

Aurora
レプリカ
(Reader)



データコピー

Availability Zone c

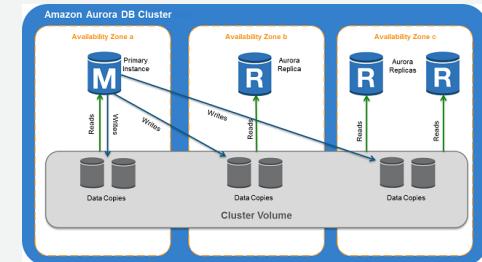


データコピー

クラスター ボリューム (Amazon Aurora ストレージ)

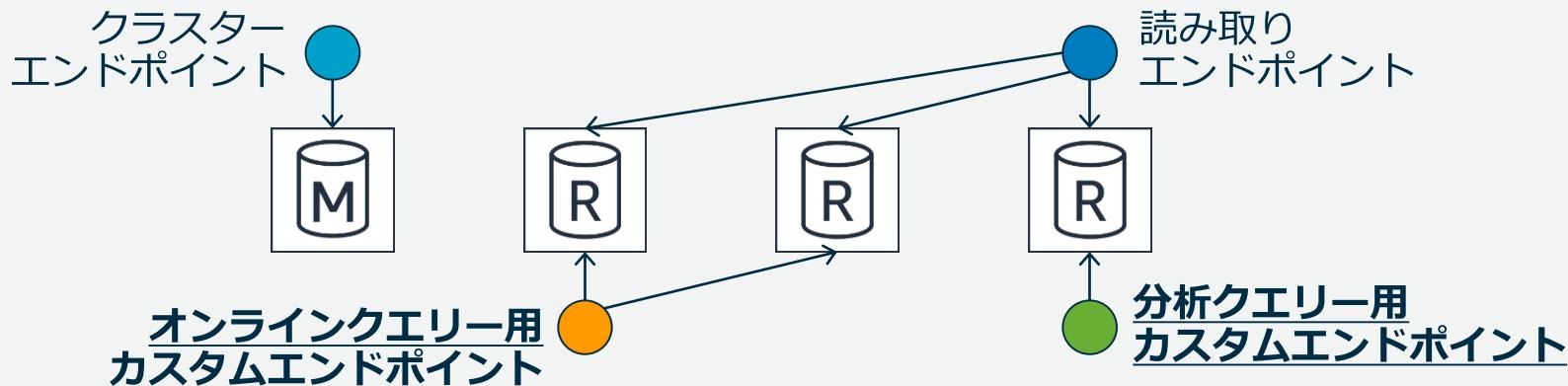
Aurora を構成するコンポーネント

- **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 볼륨의 総称
- **プライマリ DB インスタンス(Writer)** 
 - 読み込み、書き込みを行うマスターインスタンス
- **Aurora 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(Aurora 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
- **Aurora エンドポイント**
 - Aurora の接続先を示す URL



Amazon Aurora エンドポイント

- Auroraクラスター内のインスタンスへの接続先
- 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
 - 読み取りエンドポイント：どれか1つのレプリカに接続
 - 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



Aurora PostgreSQL のバージョン

- Aurora PostgreSQL では以下の形式でバージョンを表す

`<major version>.<minor version>.<patch version>`

(例: 2.3.3, 1.5.2 など)

- **major version:** PostgreSQL との対応関係あり
 - 1.y.z: 9.6系との互換性
 - 2.y.z: 10系との互換性
- **minor version:** PostgreSQL との対応関係あり
 - 例えば、2.3.z は 10.7との互換性があることを示す
- **patch version:** Aurora 独自のパッチバージョン

- 以下の SQL でバージョン確認が可能

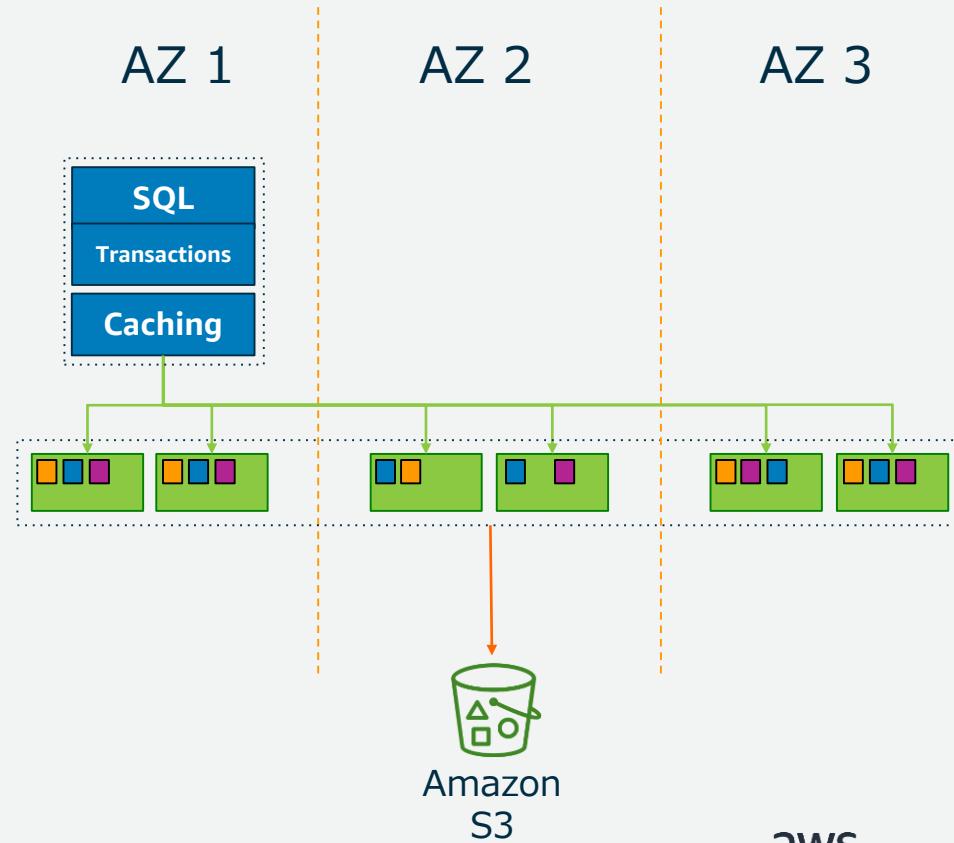
`=>SELECT AURORA_VERSION();`

可用性と耐久性

Aurora ストレージ

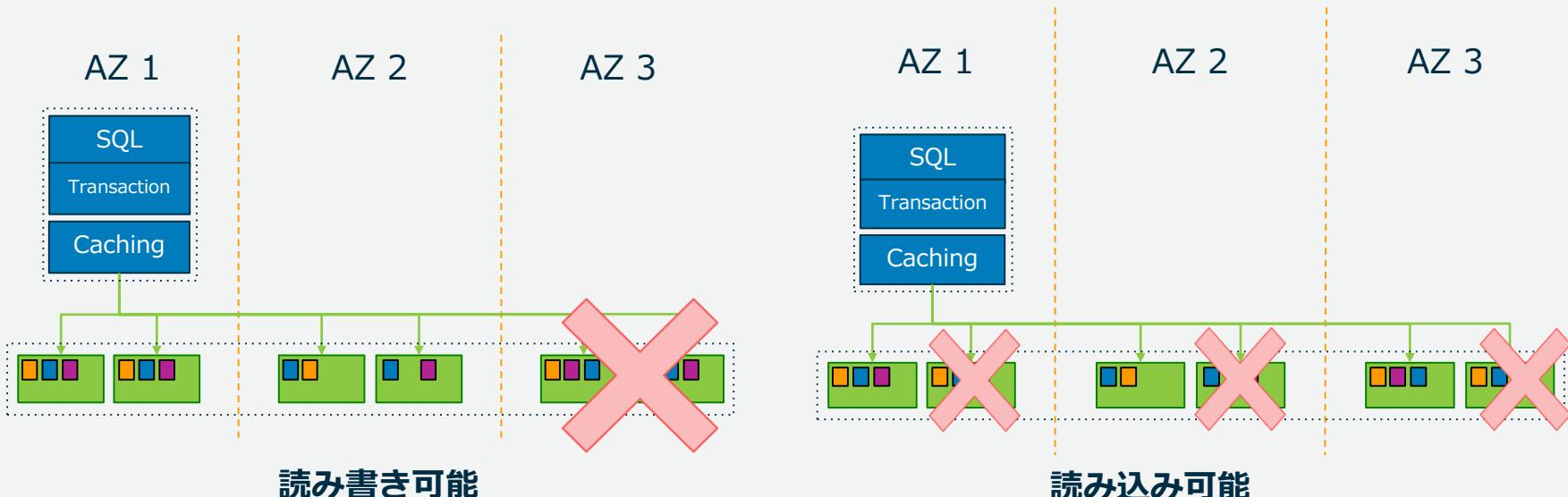
- 標準で高可用性を実現

- 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
- 継続的に S3 へ増分バックアップ



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可能
- 自動検知、修復(ノードの修復にはPeer to peer “gossip protocol”を利用)



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ゴシッププロトコルを利用し補完

Volume

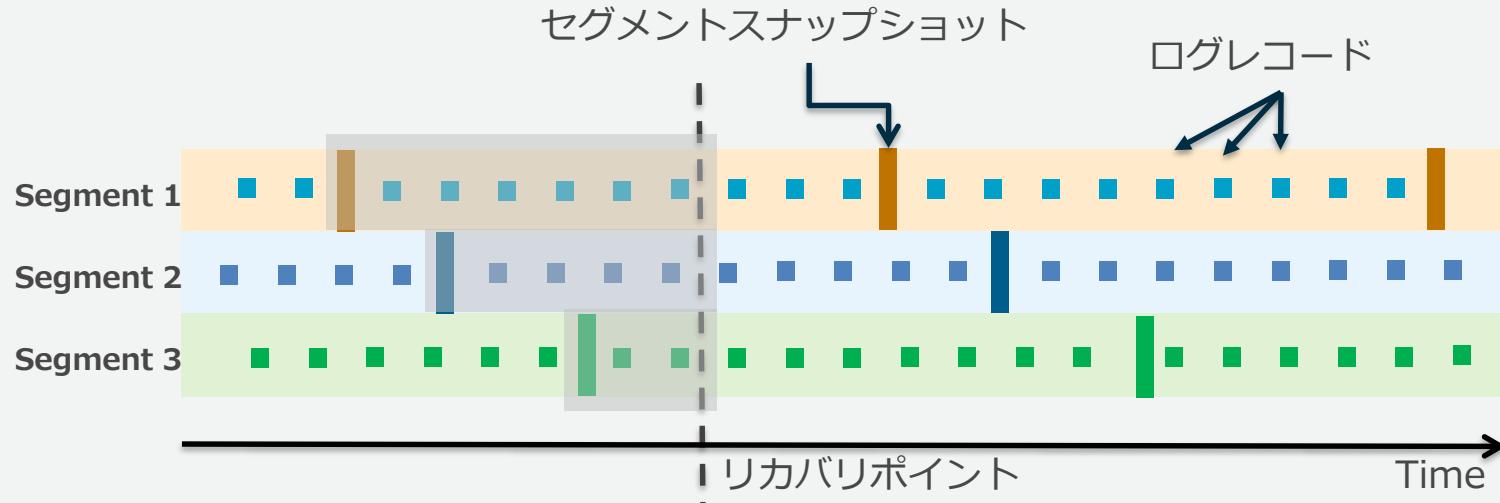
10 GB Protection Group



10 GB Protection Group



Amazon Aurora Continuous Backup



- 各セグメントごとに Amazon S3 へ継続的なスナップショットを並列に取得
- 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
- リストア時、並列非同期に適切なセグメントのスナップショットとログを取得し、ストレージへ適用

高速なデータ修復

既存のデータベース

最後のチェックポイントからログを適用していく

PostgreSQL ではシングルプロセスなため適用完了までの時間が増加

T₀ でクラッシュが発生すると最後のチェックポイントからのログを適用する必要がある

Checkpointed Data

WAL

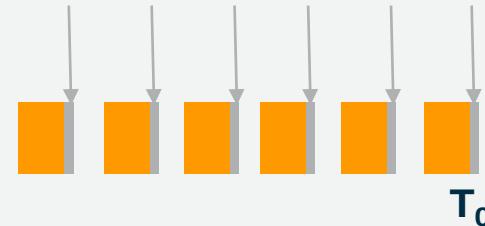
T₀

Amazon Aurora

リカバリ中かどうかに関わらず、ストレージノードでは継続的にログレコードを再生

並列、分散、非同期で行われる

T₀ でクラッシュが発生するとredo を並列で分散して非同期でログの適用を行う

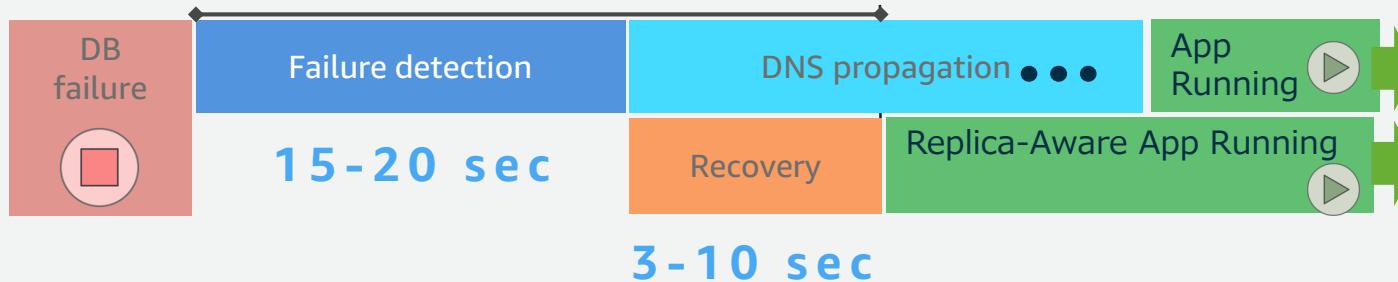


高速でより予測可能なフェイルオーバー時間

Amazon RDS for PostgreSQL is good:
failover times of ~60 seconds



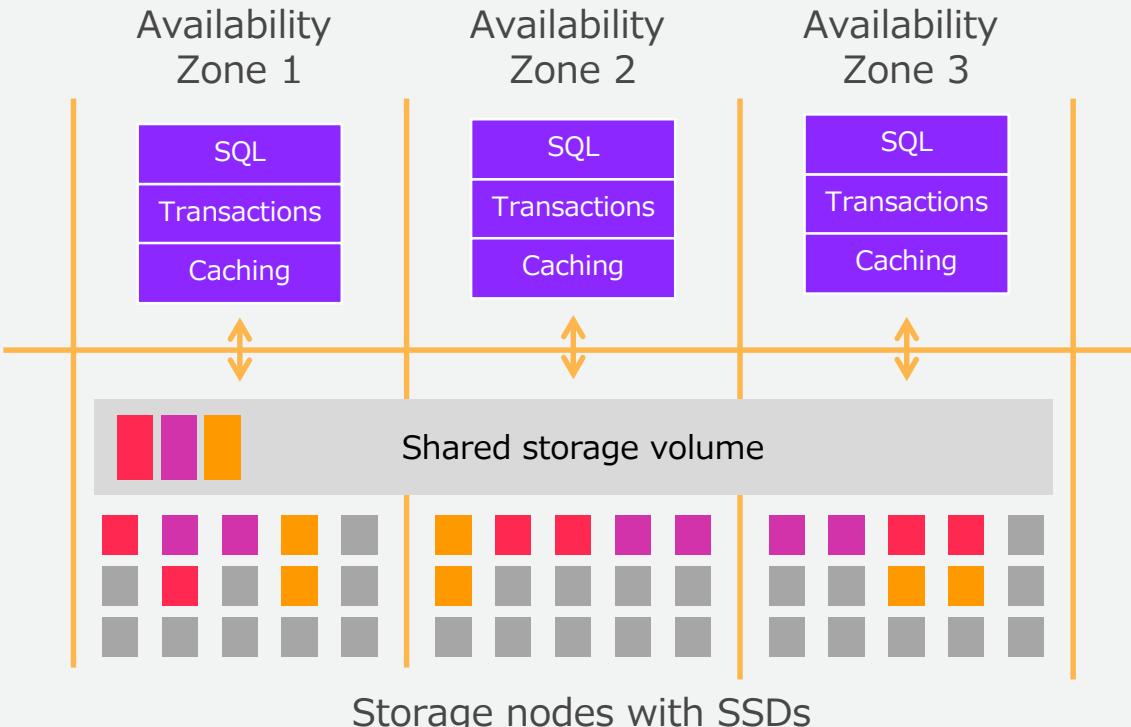
Amazon Aurora is better:
failover times < 30 seconds



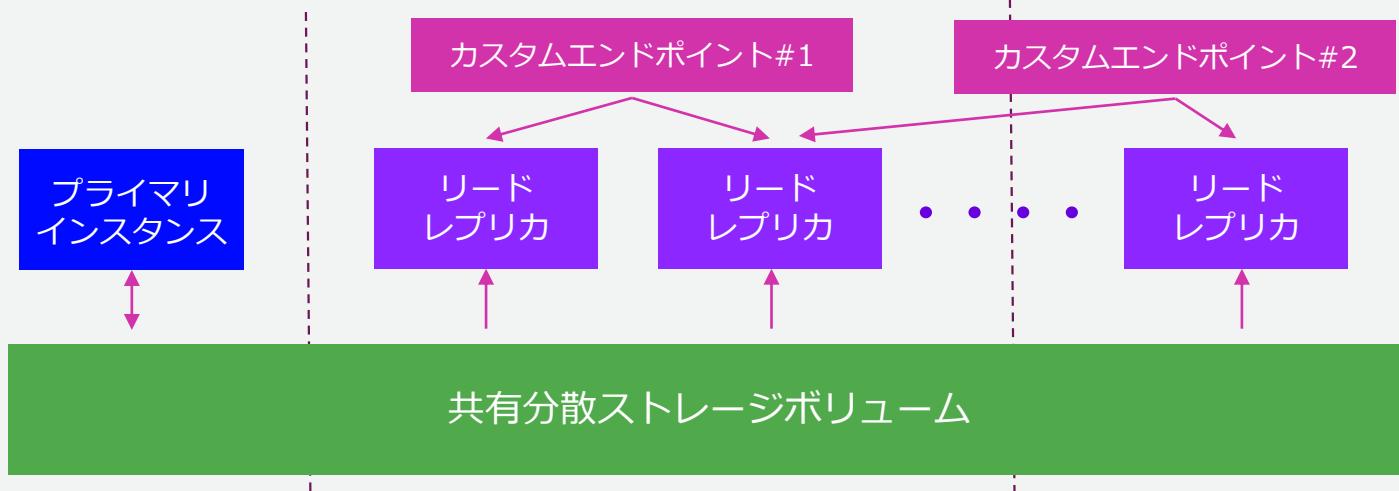
性能と拡張性

Aurora scale-out, distributed architecture

- Amazon Aurora向けに作られた log-structured 分散ストレージ システム
- データは10GBずつ“protection groups”に保存され、64TBまで自動的にスケールアップ
 - 10GBから64TBまで自動でスケールアップ
 - 実際に使った分だけ課金
- ストレージボリュームは3AZに数百～数千インスタンスのストレージノードを配置



Aurora リードレプリカ



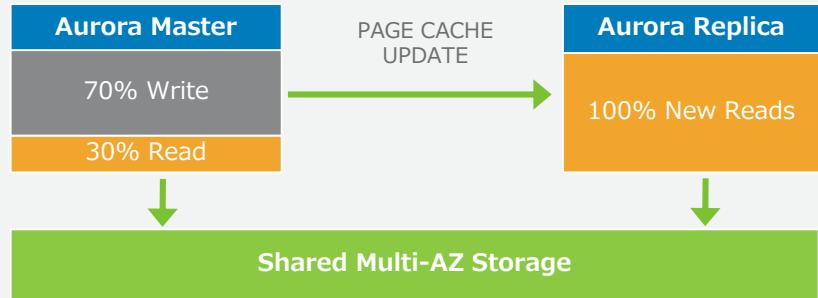
- 複数のアベイラビリティーゾーンに最大15の昇格可能なリードレプリカ
- レプリカの遅延が低い（通常10ms未満）
- フェイルオーバーの順序を設定可能
- カスタムリーダーエンドポイントによるレプリカの使い分け

IO traffic in Aurora Replicas

PostgreSQL READ SCALING



Amazon Aurora READ SCALING



Physical: WAL をレプリカに送信

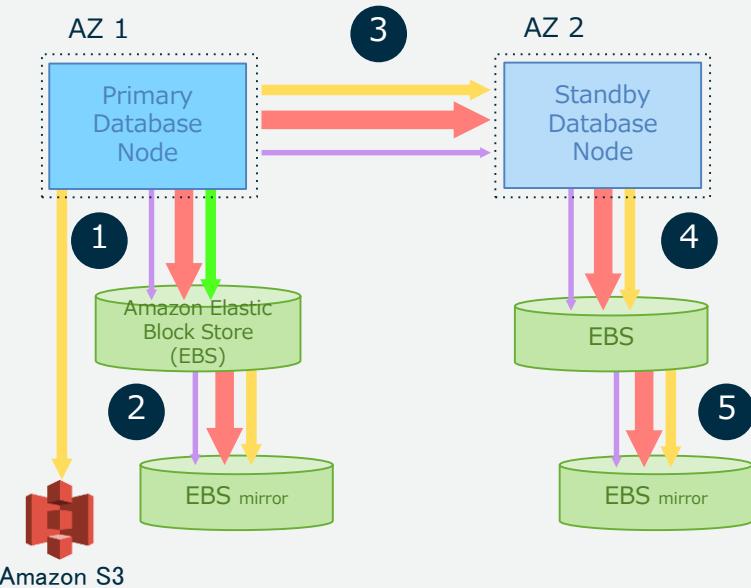
- 書き込みは各インスタンス同様に実施
- ストレージはそれぞれ独立
(Shared Nothing)

Physical: マスタからレプリカにWALを送信

- ストレージは共有。レプリカ側でストレージへの書き込みは起こらない
- ページキャッシュが適用

IO traffic in Amazon RDS for PostgreSQL

RDS FOR POSTGRESQL WITH MULTI-AZ



IO FLOW

EBSに書き込み – EBSがミラーへ複製し、両方終了後ack
スタンバイインスタンス側のEBSに書き込み

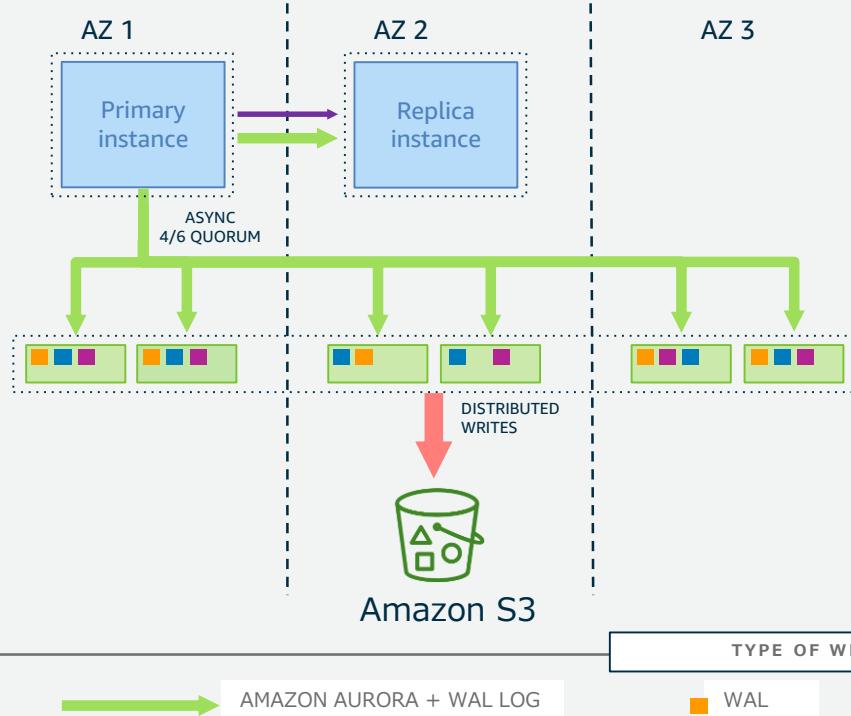
OBSERVATIONS

ステップ1, 3, 5はシーケンシャルかつ同期
それによりレイテンシーもパフォーマンスのゆらぎも増加
各ユーザー操作には様々な書き込みタイプがある
書き込み破損を避けるためにデータブロックを2回書く必要性



IO traffic in Aurora (データベース)

AMAZON AURORA



IO FLOW

REDOログレコードをまとめる – 完全にLSN順に並ぶ
適切なセグメントに分割する – 部分ごとに並ぶ
ストレージノードへまとめて書き込む

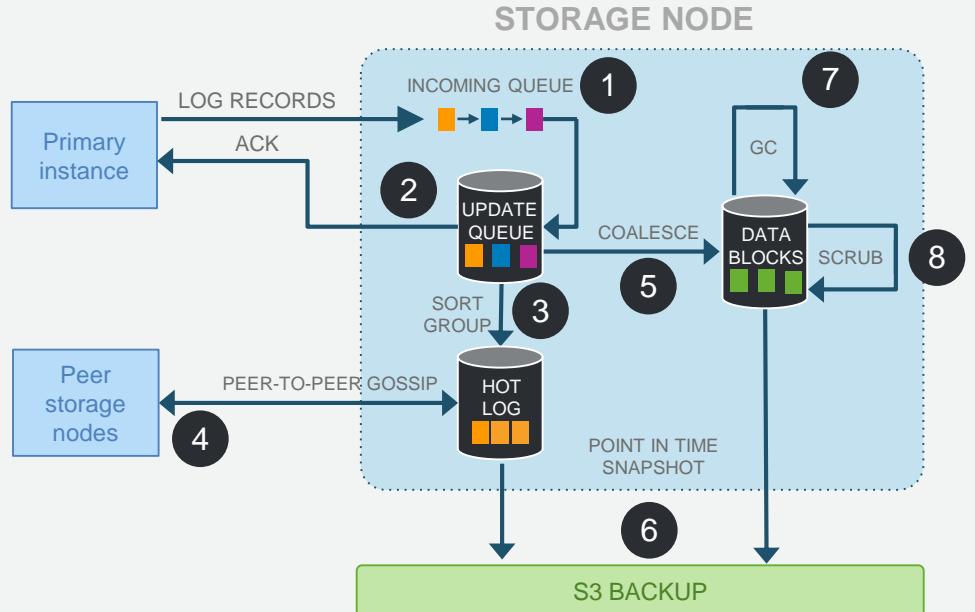
OBSERVATIONS

REDOログレコードのみ書き込む; 全てのステップは非同期
データブロックは書かない(チェックポイント, キャッシュ置換時)
6倍のログ書き込みだが, **1/9の**ネットワークトラフィック
ネットワークとストレージのレイテンシー異常時の耐性

PERFORMANCE

write-only もしくは、read/write が混在するワークロードにて、PostgreSQL のコミュニティエディションに比べて、2倍以上の性能を発揮

IO traffic in Aurora (ストレージノード)



IO FLOW

- ① レコードを受信しインメモリのキューに追加
- ② レコードをSSDに永続化してACK
- ③ レコードを整理してギャップを把握
- ④ ピアと通信して穴埋め
- ⑤ ログレコードを新しいバージョンのデータブロックに合体
- ⑥ 定期的にログと新しいバージョンのブロックをS3に転送
- ⑦ 定期的に古いバージョンのガベージコレクションを実施
- ⑧ 定期的にブロックのCRCを検証

OBSERVATIONS

全てのステップは非同期
ステップ1と2だけがフォアグラウンドのレイテンシーに影響
インプットキューはPostgreSQLに比べて**極めて小さい**
レイテンシーにセンシティブな操作に向く
ディスク領域をバッファーに使ってスパイクに対処

運用管理

Amazon Aurora クラスター全体のライフサイクル

利用可能

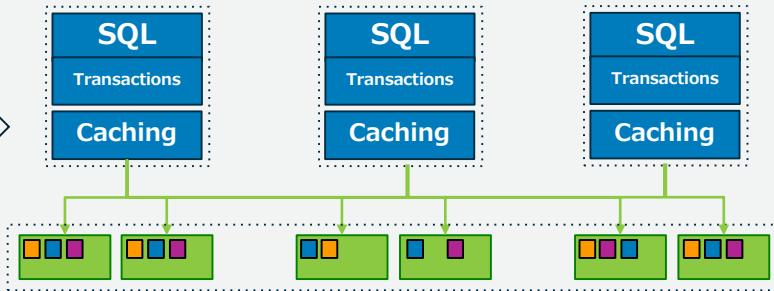
AZ 1

AZ 2

AZ 3

削除

起動



停止

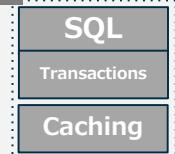
停止*

停止

停止*

開始

停止*



削除

Terminated

※ 削除直前のスナップショットを取得可能

停止に関するTips

- 全インスタンス(プライマリ、レプリカ)が停止
- 停止可能な期間は最大 7 日間
- 停止に伴い、ログは削除されるため、必要に応じてエクスポート
- 停止中もストレージ料金は発生

削除に関するTips

- 削除時に最終スナップショットが取得可能
- 削除したクラスターの復元にはスナップショットが必要

* 実態としては削除

クラスター内のインスタンスライフサイクル

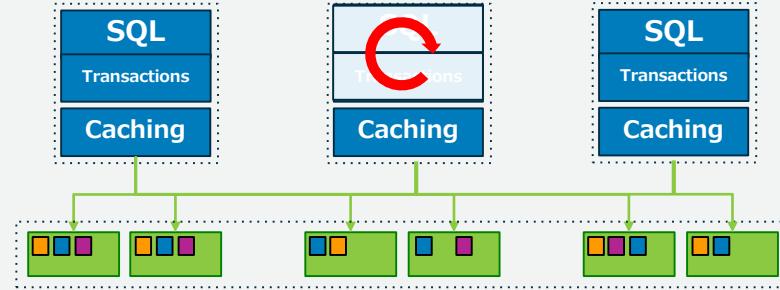
利用可能

AZ 1

AZ 2

AZ 3

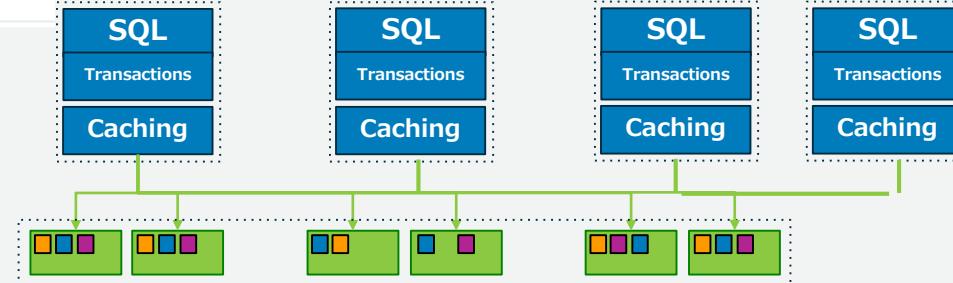
再起動中



レプリカ追加

レプリカ削除

利用可能



再起動に関するTips

- データベースプロセスのリストートが発生してもキャッシュが残った状態を維持可能
- プライマリインスタンスを再起動すると、Aurora レプリカもすべて自動的に再起動

レプリカ追加に関するTips

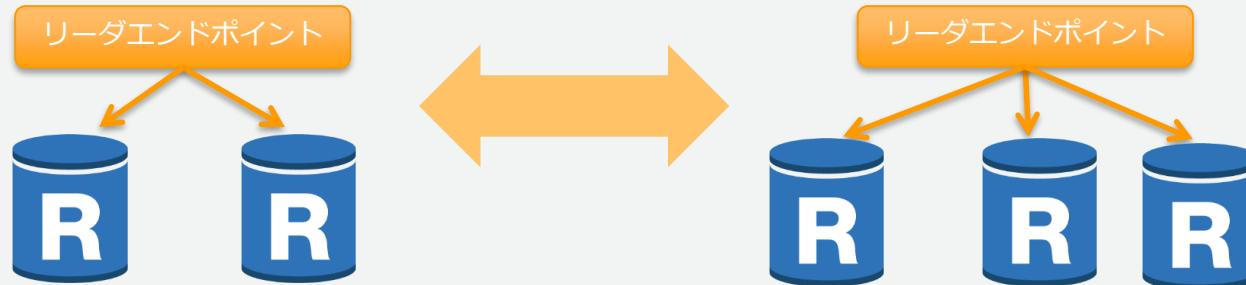
- クラスターのダウンタイムなく追加可能
- DNSの伝播時間についての考慮
- 追加したインスタンスのキャッシュについての考慮
- 最大15台まで

レプリカ削除に関するTips

- クラスターのダウンタイムなく削除可能
- DNSの伝播時間についての考慮

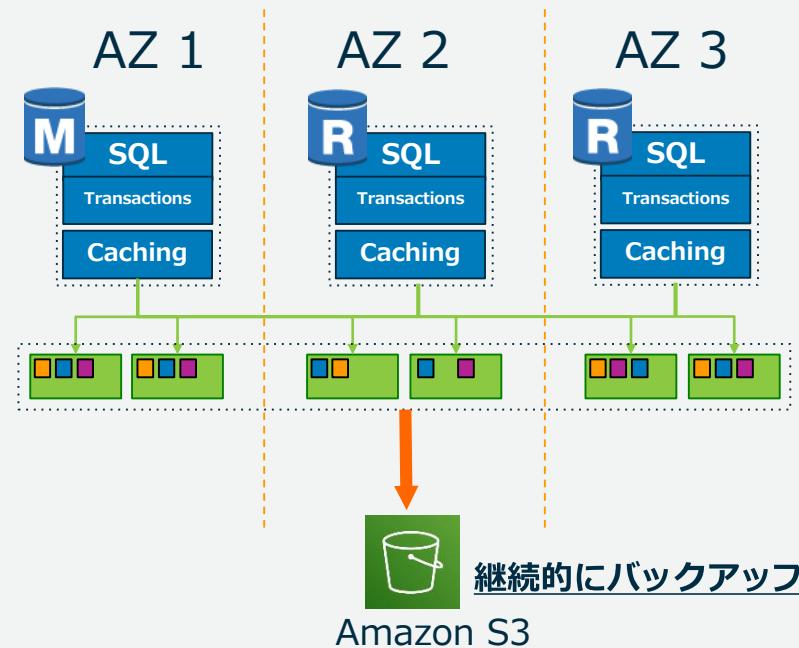
Auto Scaling によるレプリカの自動増減

- Aurora レプリカをメトリクスに応じて動的に増減
 - 読み取りクエリの分散や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を利用して自動的なレプリカの追加削除に対応可能
 - Cooldown PeriodやMinimum / Maximum Capacityを設定可能
 - 注意点
 - 追加されるのはPrimary Instanceと同じDBインスタンスクラス
 - 監視間隔、起動時間を考慮すると急激なスパイクへの対応は困難
※ 予測可能なイベントでは事前にレプリカを追加
 - 増えたレプリカのキャッシュ管理



Aurora のバックアップ機能

- 自動バックアップが常に有効
 - Amazon S3 へ継続かつ自動的に増分バックアップ(バックアップウィンドウも指定不要)
 - Aurora ストレージの仕組みによりパフォーマンスへ影響は出ない
 - バックアップの保持期間(1~35日)のみ指定
- データの復元
 - 取得したスナップショットから復元可能
 - ポイントインタイムリカバリとして、5分前からバックアップの保持期間までの任意の位置に秒単位で復元可能
 - 最新の復元可能時刻はマネジメントコンソールで確認可能(Latest Restorable Time 値または Earliest Restorable Time 値)



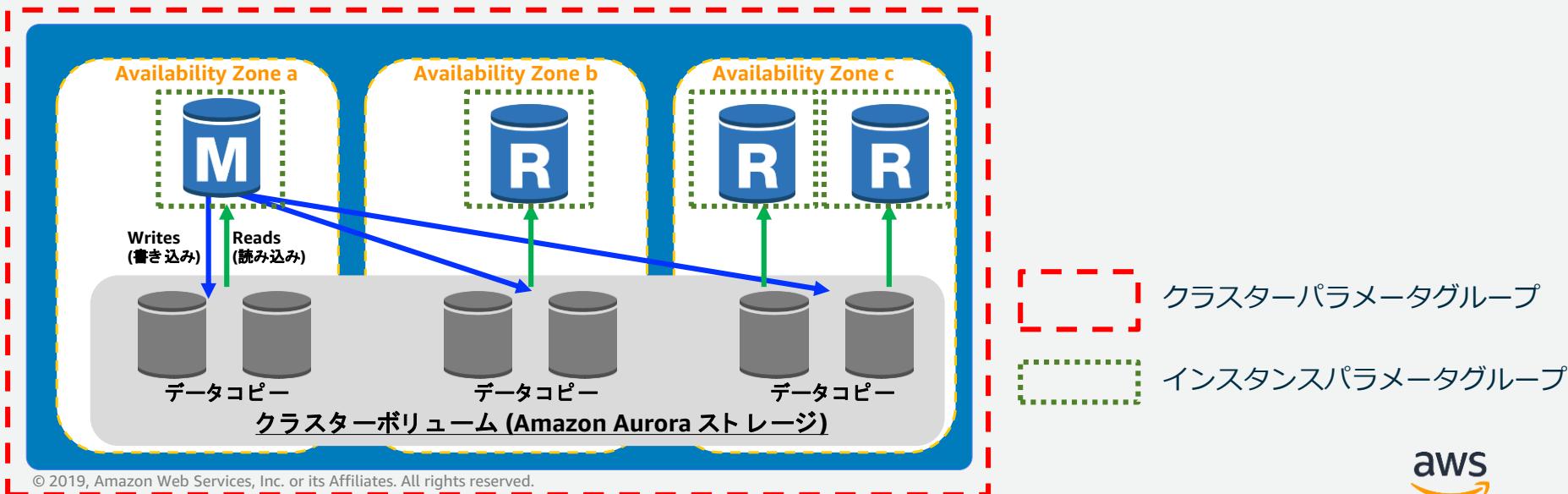
パラメーターグループによる設定

- パラメータ設定(主にpostgresql.conf の内容)作業を省力化
 - 複数インスタンスに同じパラメーターを簡単にアタッチ可能
 - デフォルトのパラメータはチューニング済み
- 考慮事項
 - パラメータ変更が適用されるタイミング
 - 静的パラメータ : 適用にインスタンスの再起動が必要
 - 動的パラメータ : 即座に適用
 - DB クラスター パラメータとDB パラメータ(次のページで解説)
 - パラメータ変更に伴うテスト



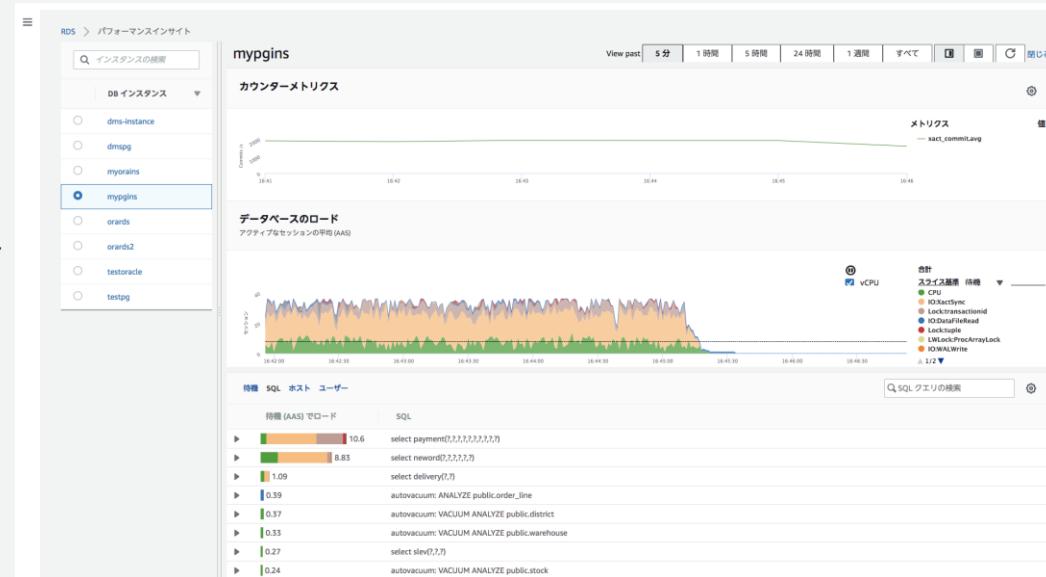
DB クラスター パラメータと DB パラメータ

- クラスター、インスタンスそれぞれにパラメータ グループを設定
 - DB クラスター パラメータ グループ: クラスター内の全インスタンスに適用
 - DB パラメータ グループ: 単一のインスタンスに適用
- ※ 重複するパラメータがある場合、DB パラメータ グループの設定値が優先される



Performance Insights によるワークロード監視

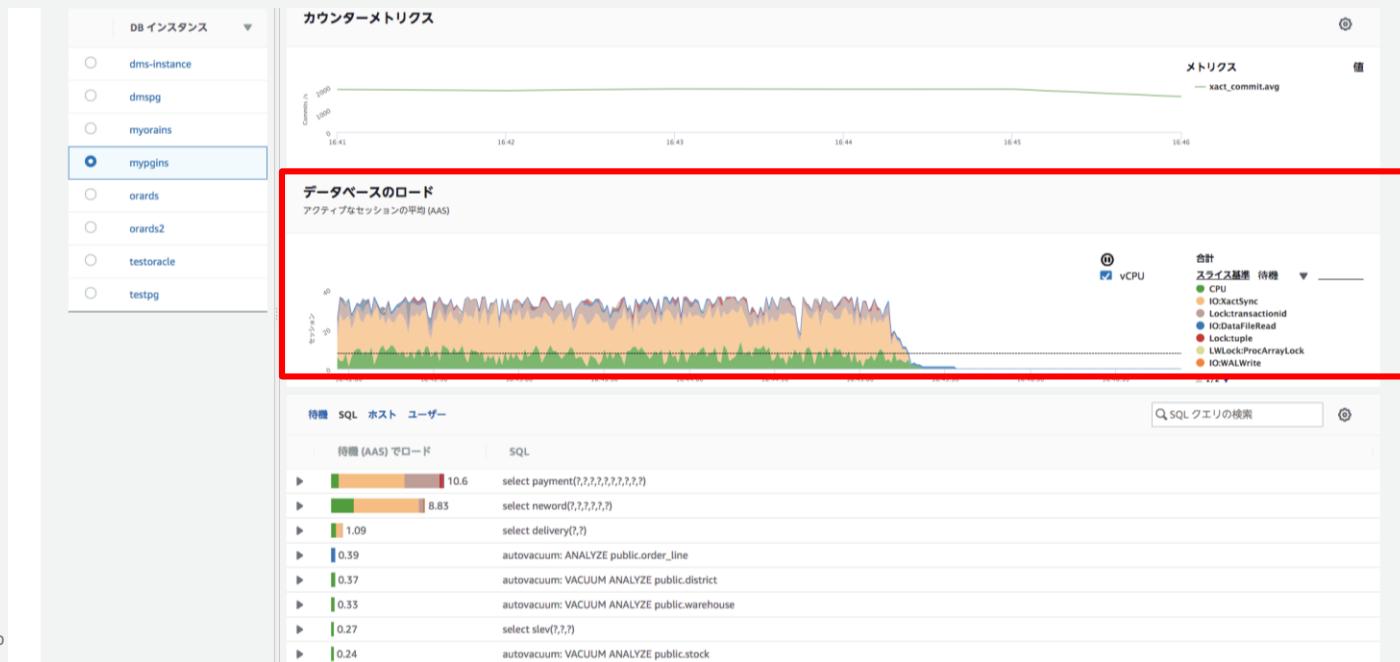
- データベースへの負荷をリアルタイムに表示
- データベースのキャパシティ、統計情報を表示
- 主要な機能
 - “データベースロード” チャート
 - “カウンターメトリクス” チャート
 - “Top N ディメンション” テーブル
 - ディメンション: 待機、SQL、ホスト、ユーザー



データベースロードチャート

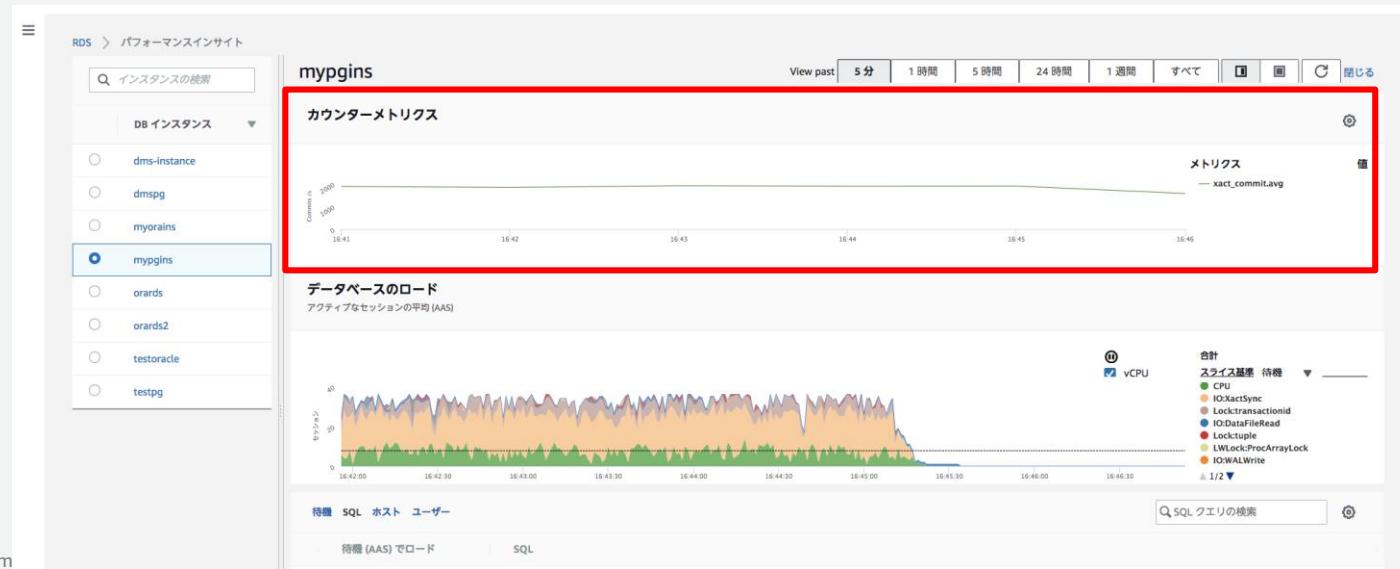
- データベースロードとは

- 全てのデータベースエンジンは“Active” / “Idle”属性の接続がある
- 1秒おきに“Active”な接続の詳細な情報をサンプリング
 - SQL文、状態(CPU処理中、待機中)、接続元ホスト、接続ユーザー



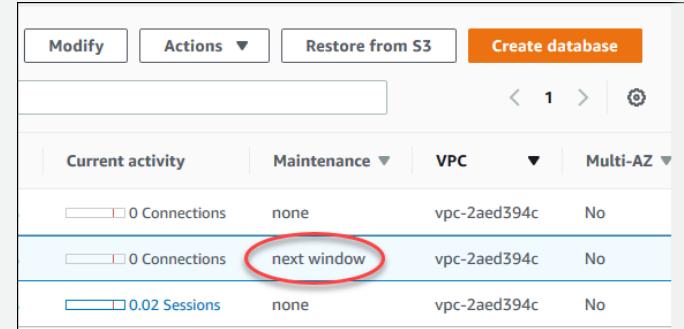
カウンターメトリクスチャート

- Performance Insights Dashboard にOS, データベースのパフォーマンスマetriクスを追加可能
 - 統計情報などの表示が可能



Amazon AuroraDB クラスターのメンテナンス

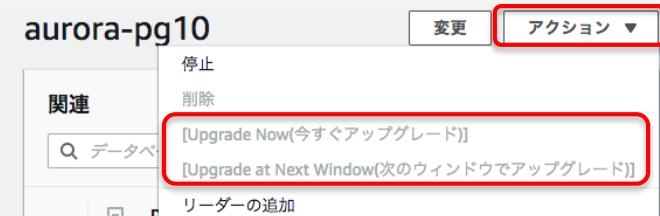
- サービス側でメンテナンスを定期的に実行
 - メンテナンスウィンドウで指定した曜日・時間帯に自動実施
 - 安全性・堅牢性に関わるソフトウェアパッチを自動適用
 - リブートやダウンタイムを伴うケースあり
 - 通常、数ヶ月に一度の頻度で発生（毎週必ずではない）
 - 指定した時間帯の数分間で実施（メンテナンス内容に依存）
- 考慮事項
 - メンテナンス有無の定期的な確認
 - マネジメントコンソール、CLI, API
 - イベント通知の設定
 - 適切なメンテナンスウィンドウ設定
 - トラフィックが少ない曜日・時間帯



Current activity	Maintenance	VPC	Multi-AZ
0 Connections	none	vpc-2aed394c	No
0 Connections	next window	vpc-2aed394c	No
0.02 Sessions	none	vpc-2aed394c	No

マネジメントコンソールでのメンテナンス確認と適用

- メンテナンス(Maintenance)列の表示を確認
 - 必須(required): メンテナンスアクションがリソースに適用(期限あり)
 - 利用可能(available) – メンテナンスアクションが利用可能
(自動的にはリソースに適用されないので、必要に応じて手動で適用)
 - 次のウィンドウ(next window) – 次回のメンテナンスウィンドウ中に適用
 - 進行中(in progress) – メンテナンスアクション適用中
- 更新の適用
 - 対象のクラスターを選択し、以下のいずれかの操作
 - [アクション]-[Upgrade Now]
 - [アクション]-
[Upgrade at Next Window]



AWS CLI でのメンテナンス確認と適用

- AWS CLI によりメンテナンスがスケジュールされた場合の通知、適用を自動化可能
 - メンテナンスの確認: describe-pending-maintenance-actions
 - 詳細: <https://docs.aws.amazon.com/cli/latest/reference/rds/describe-pending-maintenance-actions.html>
 - メンテナンスの適用 : apply-pending-maintenance-action
 - 詳細: <https://docs.aws.amazon.com/cli/latest/reference/rds/apply-pending-maintenance-action.html>
- メンテナンス適用の例

```
$ aws rds apply-pending-maintenance-action --resource-identifier <arn> --apply-action system-update --opt-in-type immediate
```

Aurora の更新情報の確認

以下のリファレンスをAurora の更新情報やサービス全体に関わるメンテナ
ンス情報の確認として利用可能

- Amazon Aurora PostgreSQL のデータベースエンジンのバージョン:
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.Updates.20180305.html
- Amazon Aurora Forum:
<https://forums.aws.amazon.com/forum.jspa?forumID=227>

新機能

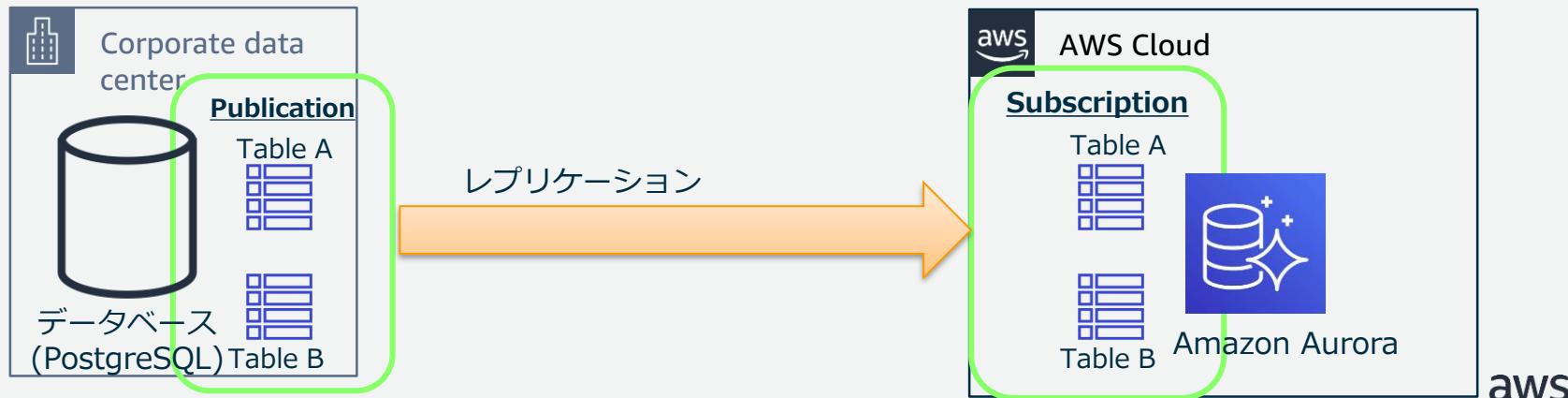
Aurora with PostgreSQL Compatibility : 新しいメジャーバージョン 10系をリリース

- PostgreSQL 10.y に対応(Aurora PostgreSQL としては v2.y)
 - 主な新機能：
 - ネイティブ・パーティショニング(宣言的パーティショニング)
 - パラレルクエリ強化(パラレルクエリに対するScan方式、Join方式の追加)
 - postgres_fdw強化(リモートサーバでの集約に対応)

→その他 : <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.Updates.20180305.html>

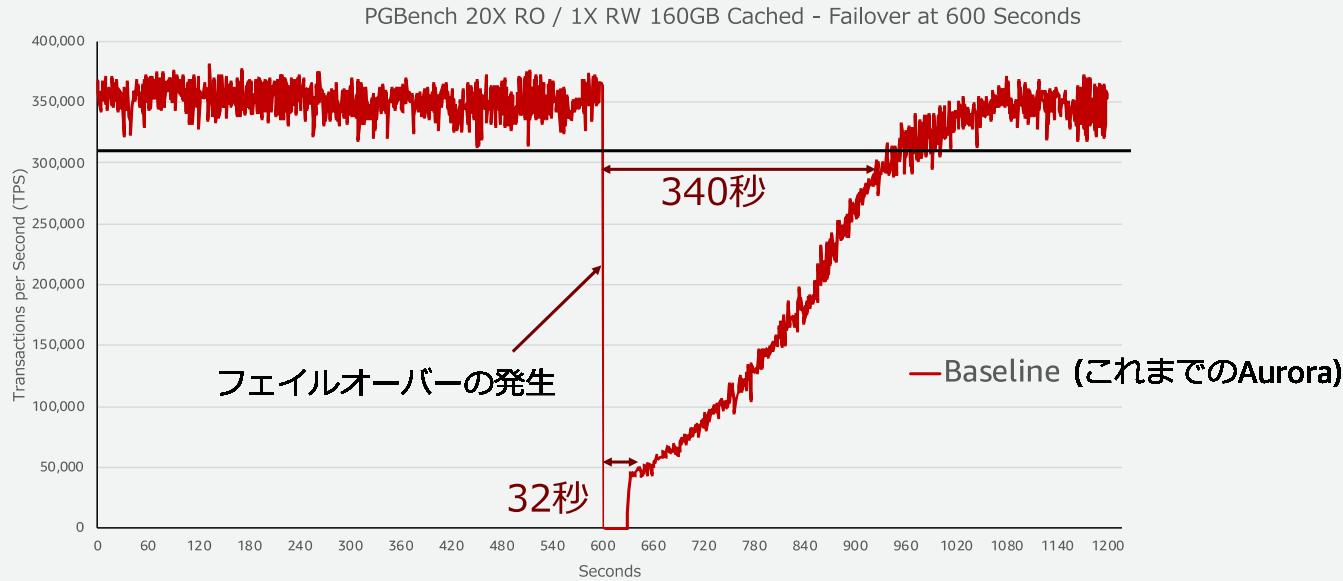
PostgreSQL 論理レプリケーションのサポート

- ロジカルレプリケーションが利用可能
 - Aurora PostgreSQL と外部のPostgreSQL でのレプリケーションが可能
 - 別リージョンの RDS/Aurora PostgreSQL ヘレプリケーションする災害対策
 - 移行元の PostgreSQL から同期することで移行時のダウンタイムを短縮
 - 基本的な手順/注意点(制限)は PostgreSQL マニュアルを参照
 - 例)DDLなどのレプリケーションは出来ない
 - 利用可能なバージョン
 - Aurora: Aurora with PostgreSQL Compatibility 2.2.0(PostgreSQL 10.6互換)以降
 - RDS: RDS for PostgreSQL 10.4以降



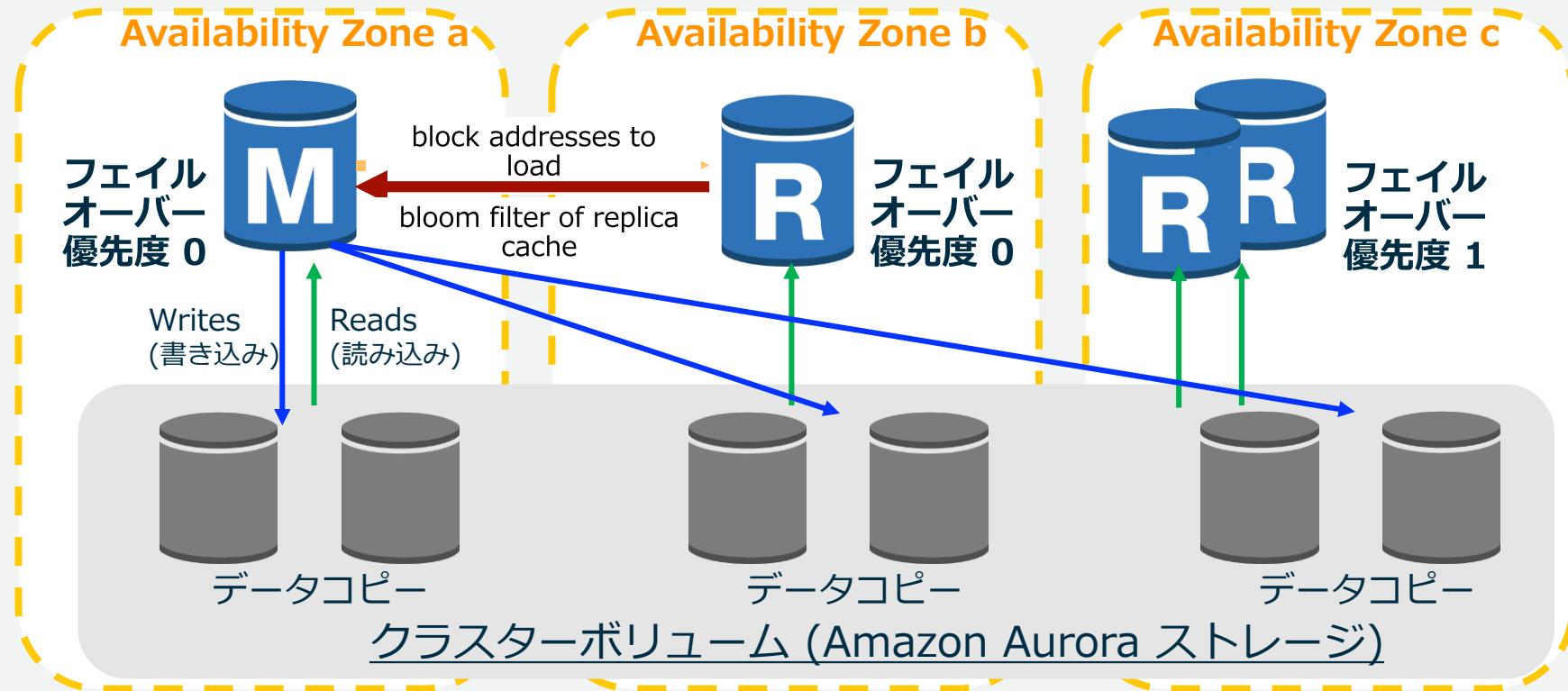
Cluster Cache Management による高速リカバリ

フェイルオーバー後のコールドキャッシュを回避



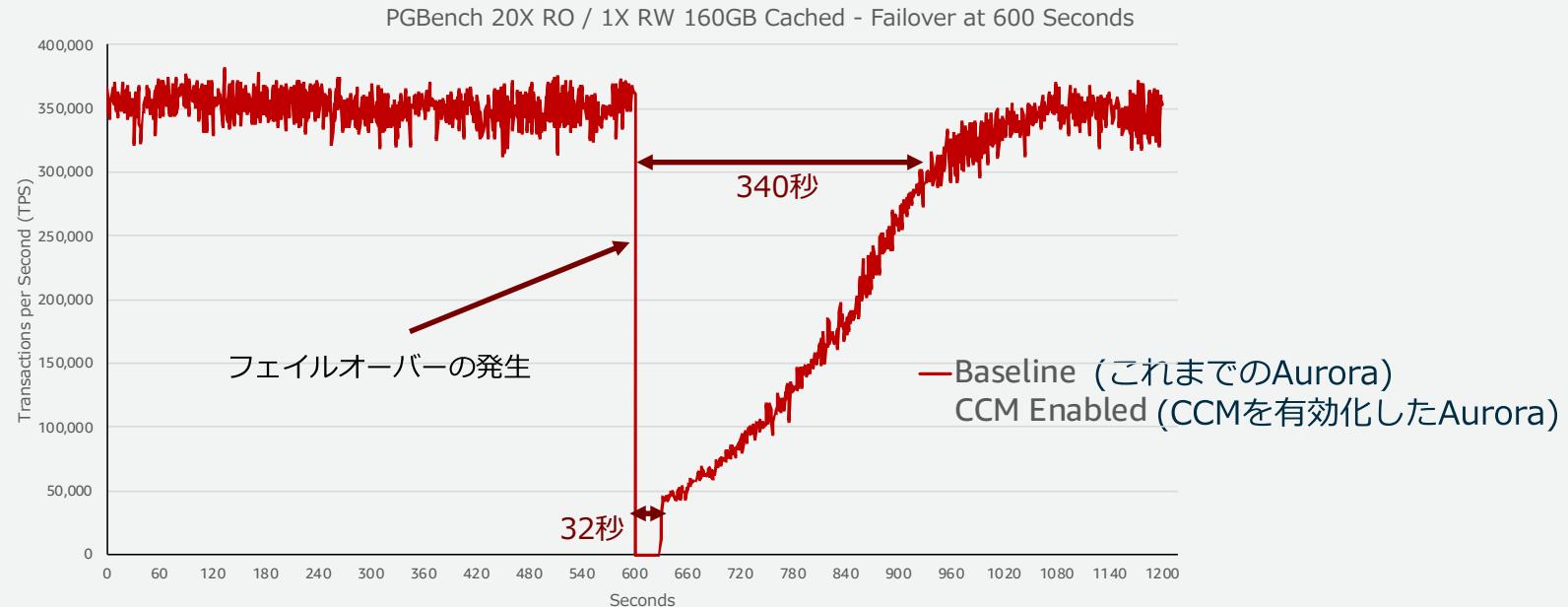
通常、データベースをバックアップしてキャッシュをウォームアップするにはしばらく時間がかかります。このフェールオーバーの例では、CCMがないと、DBの起動に32秒かかりましたが、パフォーマンスの90%を回復するには340秒でした

Cluster Cache Management(CCM) の動作



Cluster Cache Managementによる高速リカバリ

Writerとほぼ同期されたウォームキャッシュを持つレプリカへフェイルオーバー



CCMが有効になっていると、データベースはウォームアップされたキャッシュにフェイルオーバ。フェイルオーバーから32秒後には、90%のパフォーマンスを回復

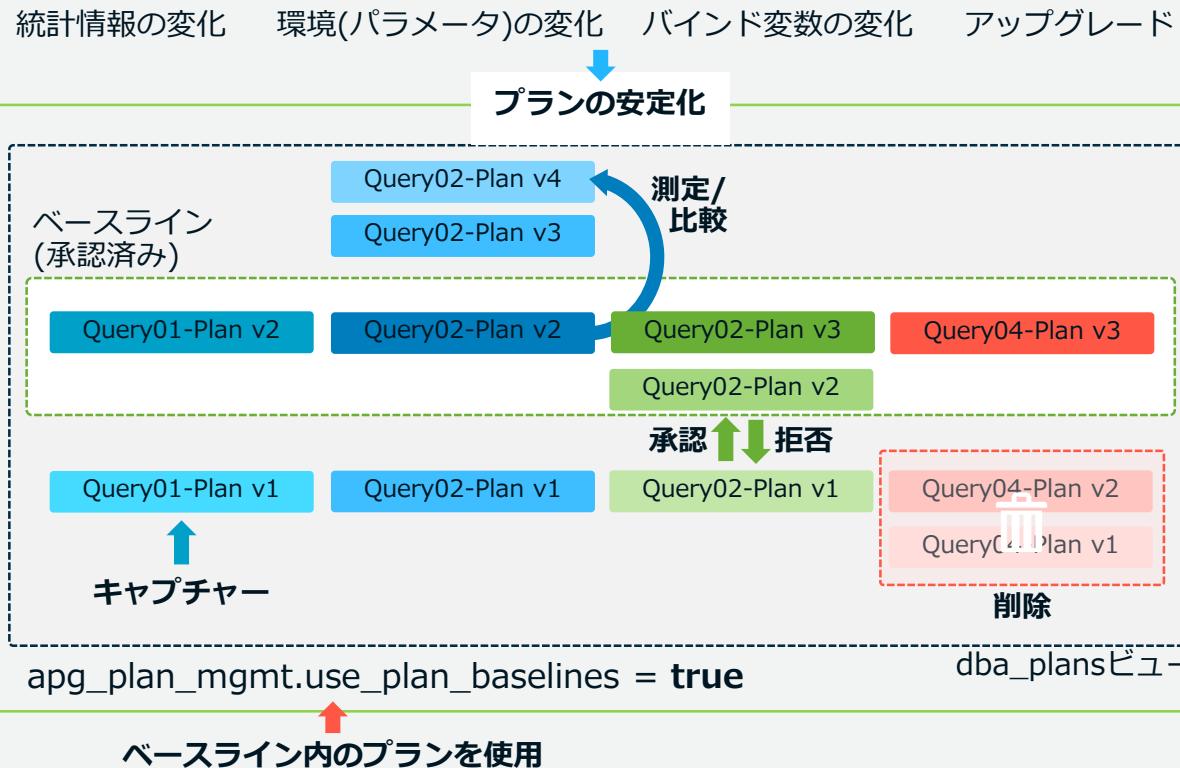
Query Plan Managementの概要

機能概要

- ✓ 手動/自動でプランのキャプチャー
- ✓ プランの測定/比較
- ✓ プランの承認/拒否
- ✓ ベースライン内のプランを使用
- ✓ pg_hint_planを使ったプランの修正
- ✓ プランの削除
- ✓ プランのエクスポート/インポート

サポートバージョン/制限

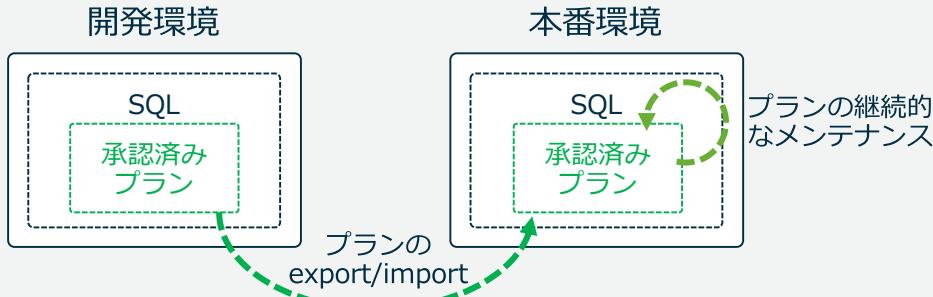
- ✓ Aurora PostgreSQL 2.1.0以上
(PostgreSQL 10.5互換)
- ✓ PL/pgSQLは未サポート



Query Plan Managementのユースケース

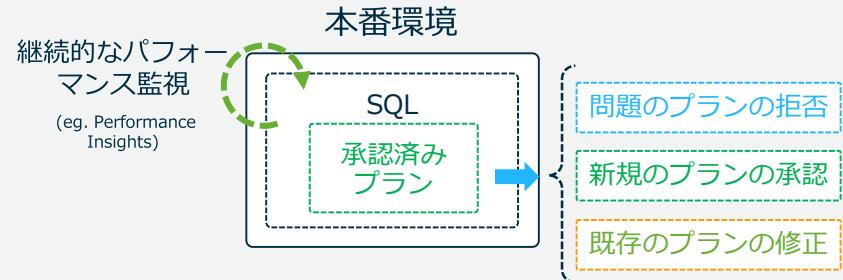
パフォーマンス低下を防止する事前予防

- 開発環境でパフォーマンスに影響を与えるSQL文を特定し、**手動/自動のキャプチャー**でプランを取得
- 開発環境から承認済みプランを**エクスポート**し、本番環境に**インポート**
- 本番環境では承認された**ベースラインのプランを強制**
- 新しくキャプチャーされたプランの効率性を**分析**し、必要な場合は承認する



パフォーマンス低下を検出し修復する事後対応

- アプリケーションのプランを**ベースラインで固定**しつつ、新しいプランのキャプチャーを継続
- 実行中のアプリケーションの**パフォーマンス低下を監視** (ex. Performance Insights)
- 既存のベースラインのプランを**拒否**し、適切な別の承認済みプランを使用させる
- pg_hint_plan拡張**でプランを**修正**することも可能

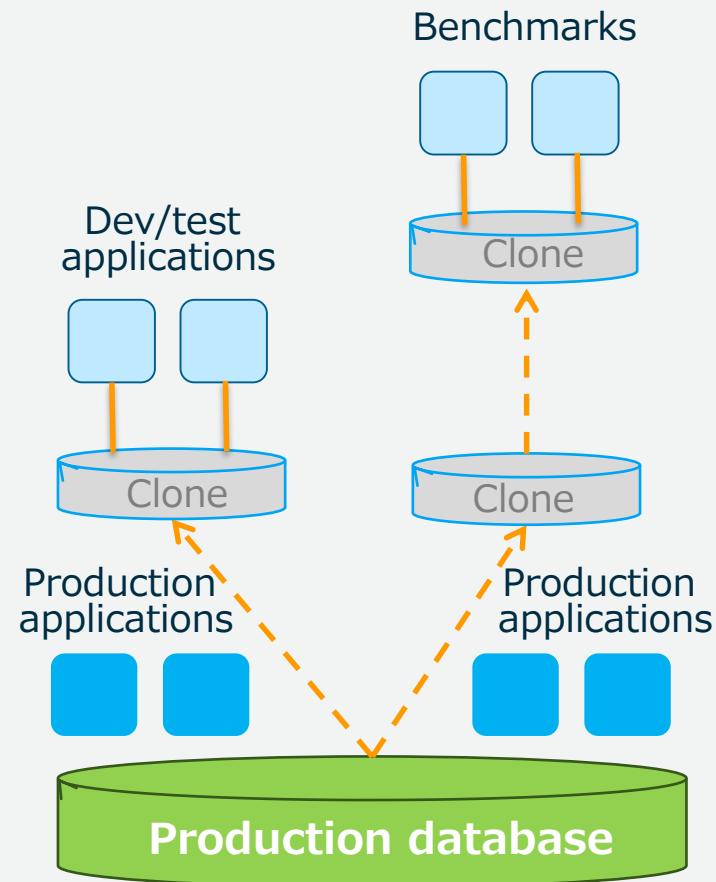


Database cloning

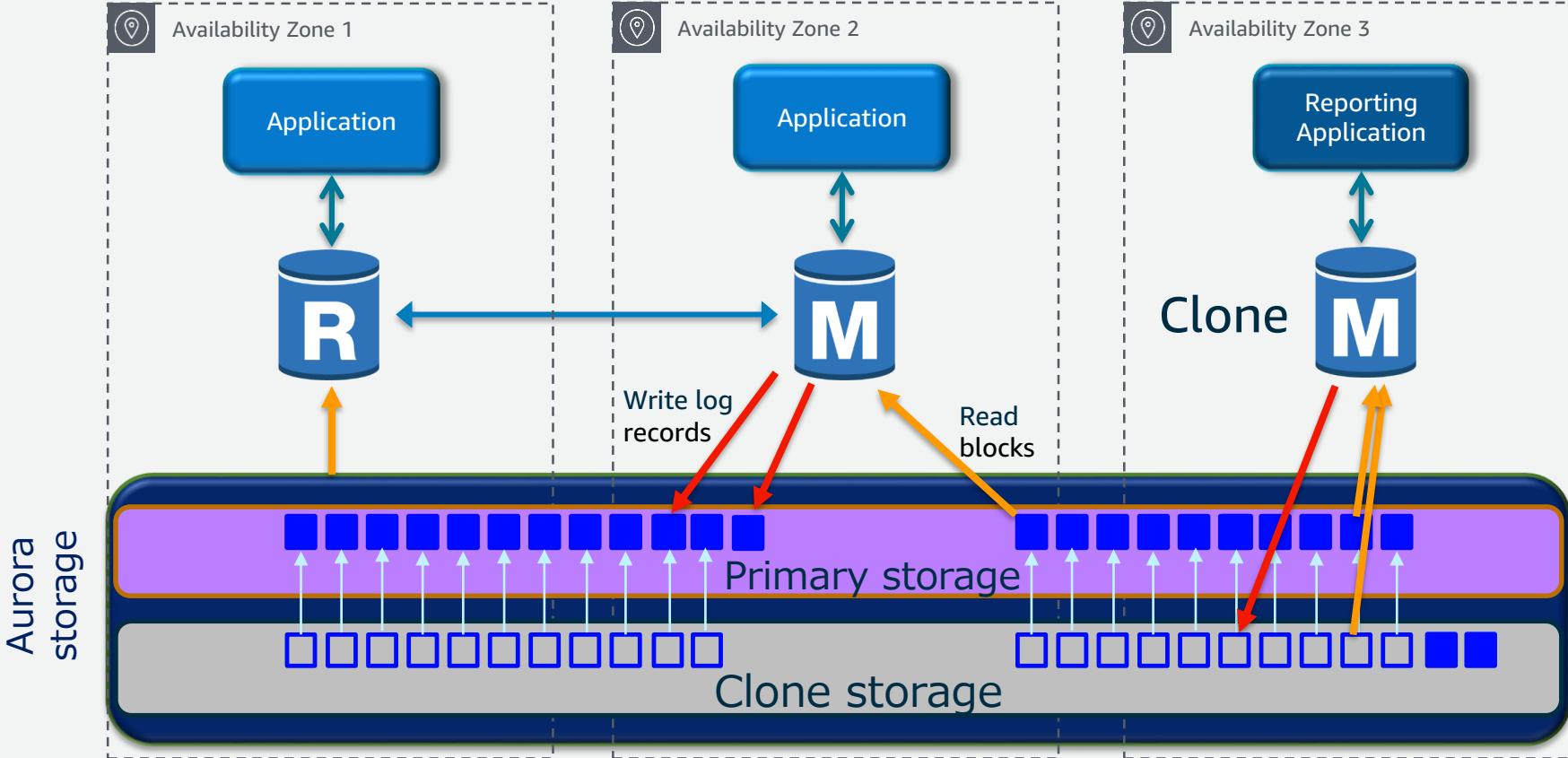
- **ストレージコストを増やすことなくデータベースのコピーを作成**
 - データをコピーするわけではないため、クローンの作成はほぼ即座に完了
 - データのコピーはオリジナルボリュームとコピー先のボリュームのデータが異なる場合の書き込み時のみ発生
 - アカウント間のクローン共有も可能

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



Database cloning のしくみ



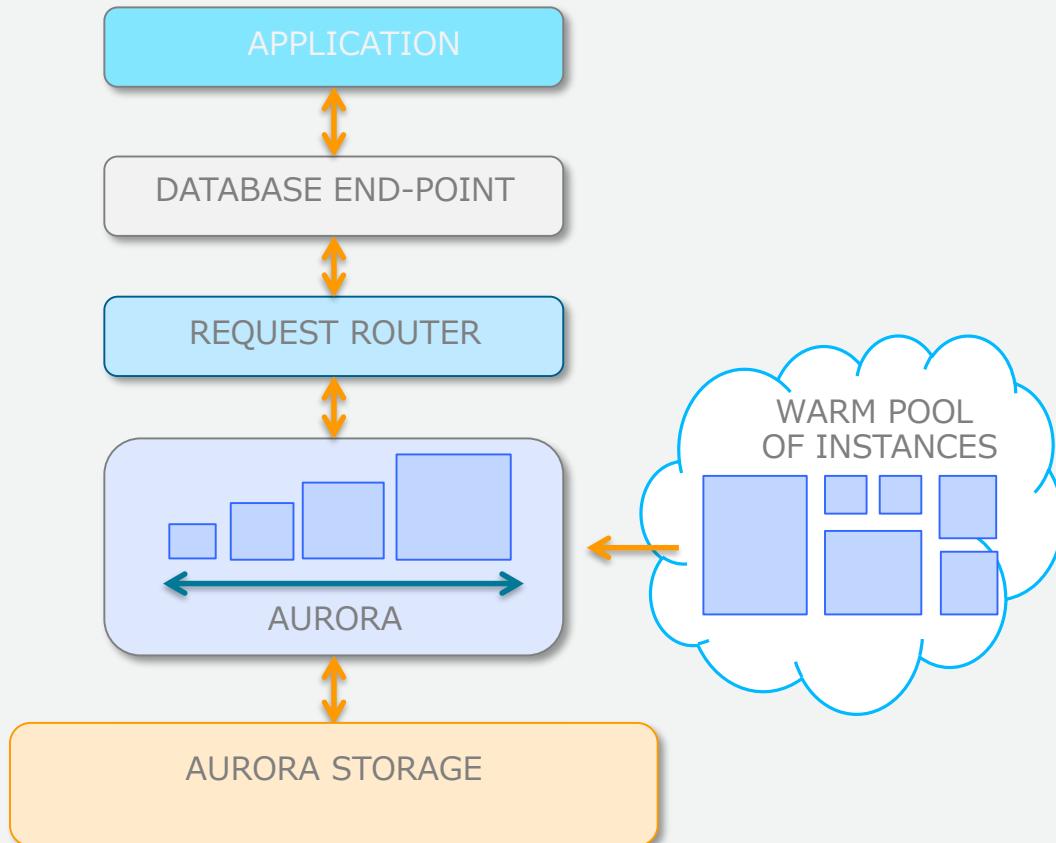
Aurora Serverless

特徴

- 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
 - 利用状況に応じたにスケーリング
 - 利用されていない場合は自動停止
- スケール時のクライアント接続の中断なし
- 秒課金(ただし、1分が最低利用料金)

ユースケース

- 不定期利用のアプリケーション
- 開発・テスト用途
- リクエスト予測が難しい場合
※定常にリクエストが予測できる場合は通常の Aurora(*)の利用がおすすめ



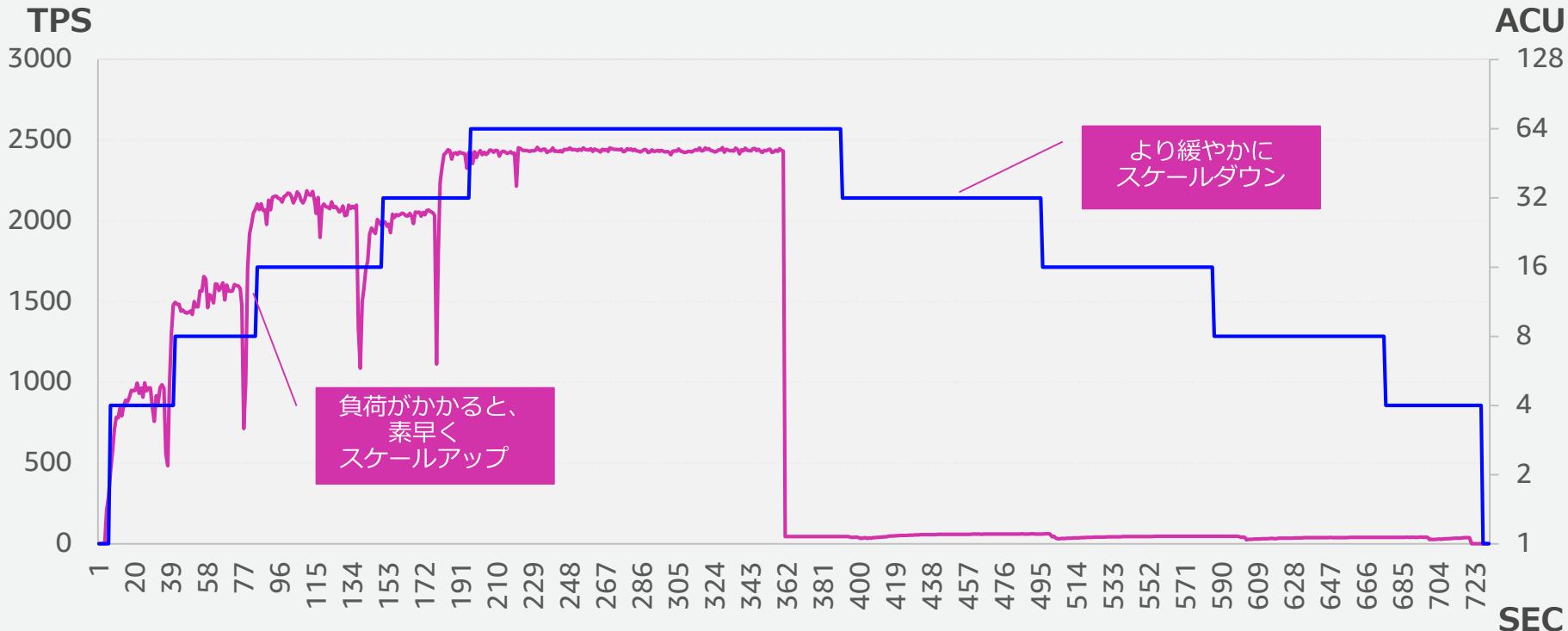
(*)通常のAurora(サーバーレスではないAurora)をprovisioned DB cluster と呼びます

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

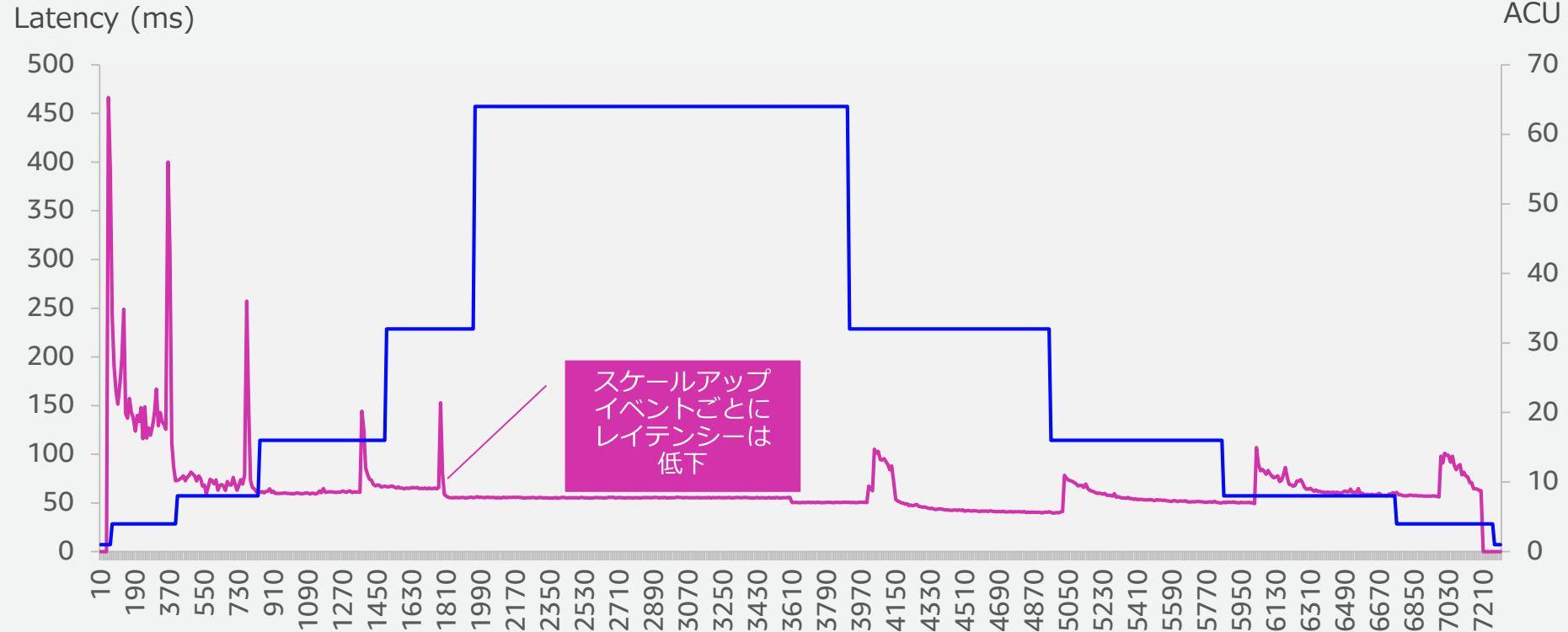
詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html



Aurora Serverless: 負荷に応じたスケールアップ・ダウン



Aurora Serverless: レイテンシーを即座に安定化



Amazon Aurora / RDS for PostgreSQL: IAM認証のサポート

- 通常のパスワードではなく、接続前にIAMに権限に基づいて認証トークン作成を依頼し、その認証トークンをパスワードの代わりに使用
- 認証トークンの有効期限は15分
- データベースクラスターの 1 秒あたりの最大接続数は、インスタンスタイプとワークロードに応じて制限

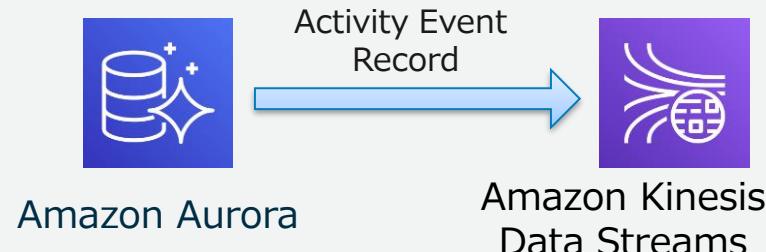
```
GRANT rds_iam TO foo;  
$ export PGPASSWORD="$(aws rds generate-db-auth-token ¥  
  > --hostname $HOST --port 5432 --region $REGION --username foo)"  
$ psql "host=$HOST port=5432 dbname=$DBNAME user=foo ¥  
  > sslmode=verify-full sslrootcert=certificateFile"
```

厳密なパスワード管理

- 機能概要
 - ロール/ユーザに対するパスワードを誰が変更可能かを制御する機能
 - rds_password ロールを持つメンバのみをパスワード変更とする
 - パスワード要件(有効期限など)の設定を可能に
- 利用の仕方
 - rds.restrict_password_commands = 1 に設定(パラメータグループ)
 - rds_password ロール
- 注意点
 - RDS/Aurora PostgreSQL 10.6以降で利用可能
 - rds_superuser は rds_password ロールを常に持つ

Database Activity Stream による リアルタイム監視

- データベースアクティビティをニアリアルタイムで Amazon Kinesis Data Stream へプッシュする
 - SQL コマンド、接続情報などがプッシュされる
 - DB アクティビティ情報は暗号化されている
 - 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)



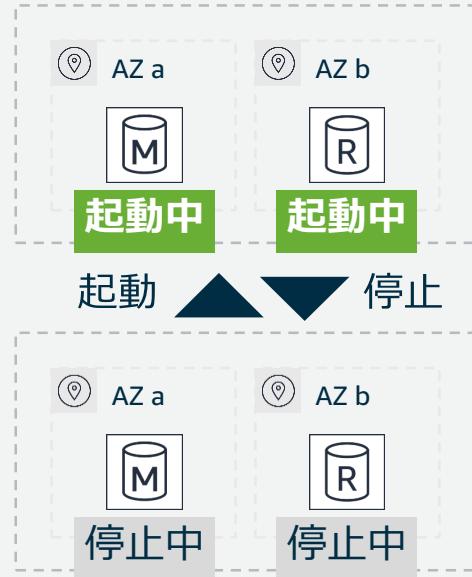
Deletion Protection(削除保護)

- 削除保護フラグを設定することで、オペミスによるDB クラスターの削除を防止することが可能
 - クラスターを削除(クラスター内に存在している最後のインスタンスを削除)しようとしても出来ないように設定
- 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



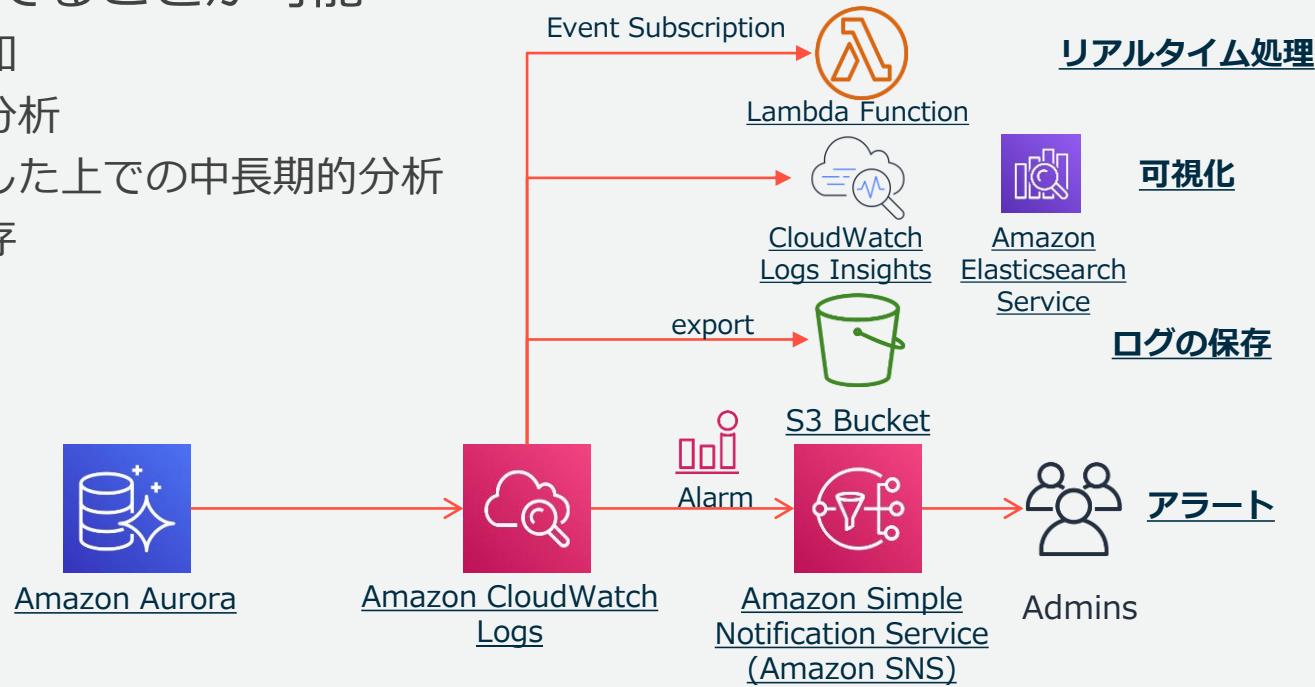
Amazon Aurora: クラスターの停止および起動をサポート

- Auroraクラスターを停止/起動可能に
- クラスターを停止すると、プライマリインスタンスとすべてのレプリカインスタンスが停止
- 停止するとインスタンス料金は課金されなくなるが、ストレージ料金は課金される
- 1回に最大7日間まで停止可能。7日後に自動起動



CloudWatch Logs との連携

- 出力されたログデータを Amazon CloudWatch Logs へ発行し、以下のタスクに役立てることが可能
 - アラームの通知
 - リアルタイム分析
 - ログを可視化した上での中長期的分析
 - ログの長期保存



CloudWatch Logs : 特定文字列検出のためのフィルター

- ログデータから特定の文字列のフィルタリングが可能
- 正規表現の利用ができない点に注意
- 一致したパターンに応じた処理が可能
 - 特定文字列の出力頻度により
アラーム通知(メトリクスフィルタ)
 - Lambda関数を実行/
Elasticsearch Service連携
(サブスクリプションフィルタ)
- ユースケース
 - メッセージレベルに応じたアラーム
 - 事前に想定されるパターンに応じて
自動処理

ログメトリクスフィルターの定義

ロググループのフィルター: /aws/rds/cluster/aurora-pg-logs/postgresql

メトリクスフィルタを使用し、ロググループ内のイベントが CloudWatch Logs に送信されるときに、それらのイベントを自動的にモニタリングできます。特定の用語のモニタリングやカウントを行ったり、ログイベントから値を抽出したりでき、その結果をメトリクスに関連付けることができます。[パターン構文の詳細はこちら](#)。

フィルターパターン

"password authentication failed"

例の表示

テストするログデータの選択

- カスタムログデータ -

クリア

パターンのテスト

```
1 UTC::@:[6668]:LOG: skipping missing configuration file "/rdsdbdata/db/postgresql.auto.conf"
1 UTC::@:[6563]:LOG: skipping missing configuration file "/rdsdbdata/db/postgresql.auto.conf"
1 UTC:10.0.0.146(51758):daichie@postgres:[20973]:FATAL: password authentication failed for user "daichie"
1 UTC:@:[6562]:LOG: received SIGHUP, reloading configuration files
1 UTC::@:[6562]:LOG: skipping missing configuration file "/rdsdbdata/db/postgresql.auto.conf"
1 UTC:@:[6562]:LOG: parameter "unix_socket_permissions" cannot be changed without restarting the server
1 UTC:@:[6562]:LOG: parameter "shared_preload_libraries" cannot be changed without restarting the server
```

結果

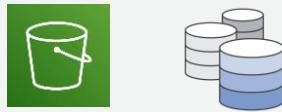
サンプルログの50 個のイベントから 2 の一致が見つかりました。

行の内容

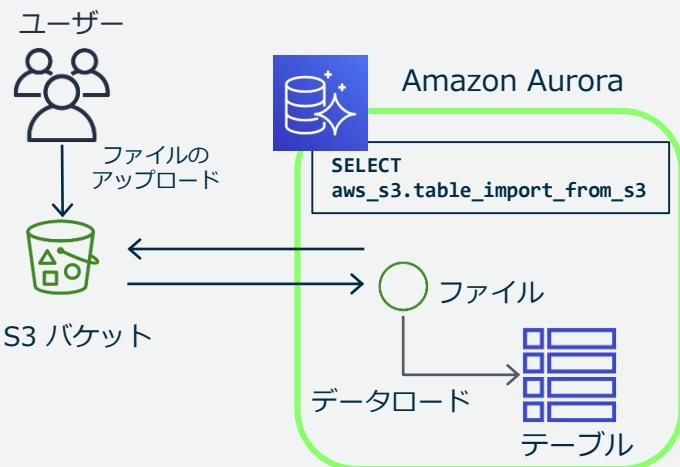
```
2019-08-16 04:49:40 UTC:10.0.0.146(51758):daichie@postgres:[20973]:FATAL: password authentication failed for user "daichie"
2019-08-16 04:58:31 UTC:10.0.0.146(51758):daichie@postgres:[20973]:FATAL: password authentication failed for user "daichie"
```



Load Data From S3



- S3 バケットに保存されたデータを直接 Aurora PostgreSQL にインポート可能
 - aws_s3 という拡張モジュールを使用
 - PostgreSQL の COPY 文でサポートされるファイル形式(csvなど)が利用可能
 - RDS/Aurora に付与した IAM ロールを通じて S3 へのアクセスを許可
 - 大量データのインポート、移行などのユースケースに対応



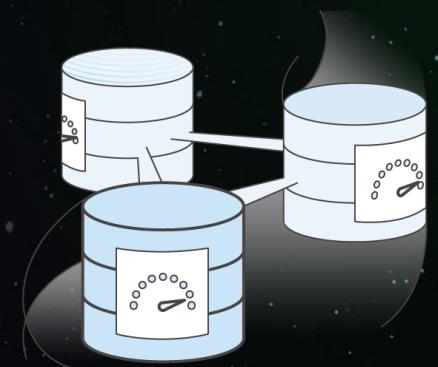
```
=> CREATE EXTENSION aws_s3 CASCADE;  
NOTICE: installing required extension "aws_commons"  
CREATE EXTENSION  
=> SELECT aws_s3.table_import_from_s3(  
    'aws_s3', '' ,  
    '(format csv)' ,  
    '(mys3bucket,path/myfile.csv,ap-northeast-1)' );  
table_import_from_s3  
-----  
30 rows imported into relation "aws_s3" from file path/myfile.csv of 912 bytes
```

New Instance Type

- R5, T3 インスタンスをサポート
 - R5: メモリ負荷の高いデータベースワークロードに最適なインスタンスタイプ
 - T3: テスト、開発などの小規模ワークロードをコスト効率よく実行
ベースラインパフォーマンス、バースト機能について事前に確
認

DBインスタンスタイプ	vCPU	メモリ(GiB)	ネットワーク
db.r5.24xlarge	96	768	25 Gbps
db.r5.12xlarge	48	384	10 Gbps
db.r5.4xlarge	16	128	最大 10 Gbps
db.r5.2xlarge	8	64	最大 10 Gbps
db.r5.xlarge	4	32	最大 10 Gbps
db.r5.large	2	16	最大 10 Gbps
db.t3.medium	2	4	最大 5 Gbps

さいごに



まとめ

- Amazon Aurora はクラウド時代に Amazon が再設計したRDBMS
 - スケーラブル、高い堅牢性、可用性をもつ設計
 - PostgreSQL 9.6, 10 との強い互換性
- 高可用性・実環境での性能向上を実現するための多くのチャレンジ/改善を継続して実行中
 - 高いスループット
 - 継続して新機能をリリース

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 page. 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", "サポート", "日本語", "アカウント", and "コンソールにサインイン". Below the navigation is a menu bar with links for "製品", "ソリューション", "料金", "ドキュメント", "学習", "パートナー", "AWS Marketplace", "その他", 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" and a descriptive paragraph about the service. At the bottom, there are four buttons: "AWS Webinar お申込", "AWS 初心者向け", "業種・ソリューション別資料", and "サービス別資料".

日本担当チームへお問い合わせ サポート 日本語 アカウント コンソールにサインイン

aws

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 Q

AWS クラウドサービス活用資料集トップ

Amazon Web Services (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 » 業種・ソリューション別資料 » サービス別資料 »

<https://amzn.to/JPArchive>

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>





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>



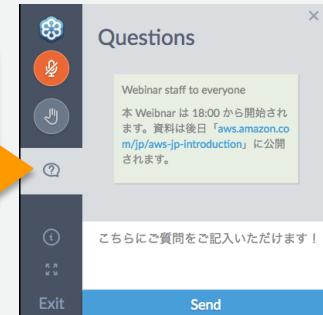
AWS Black Belt Online Seminar とは

「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



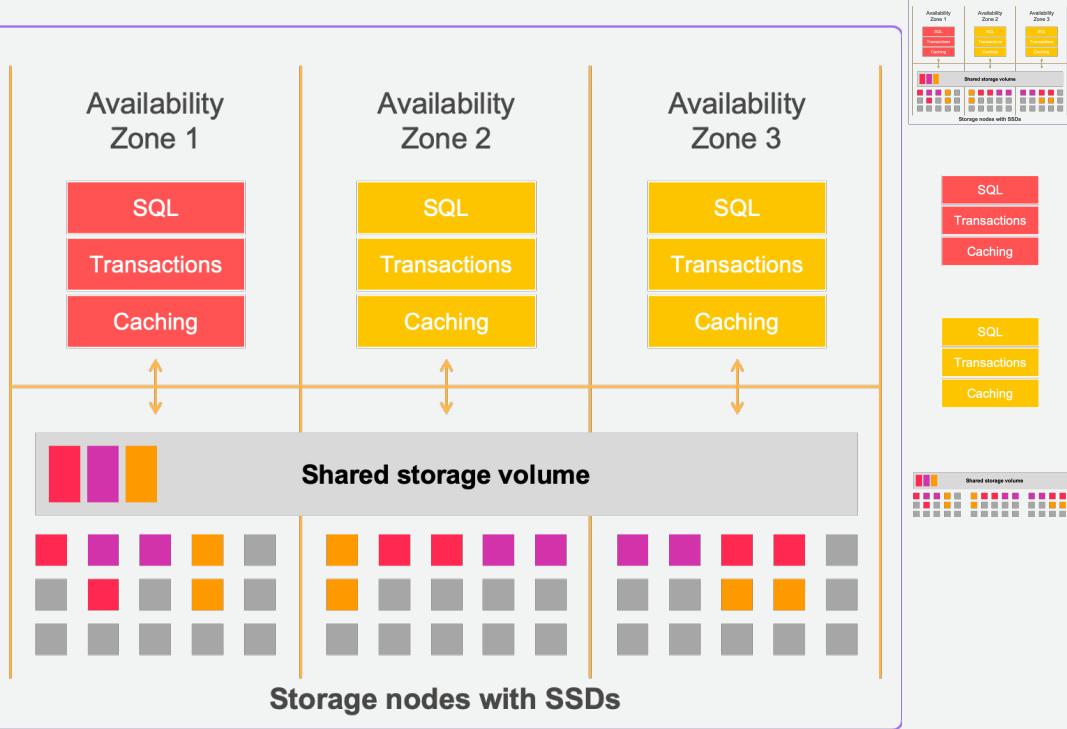
ネットワーク分離、
保管時/通信の暗号化

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

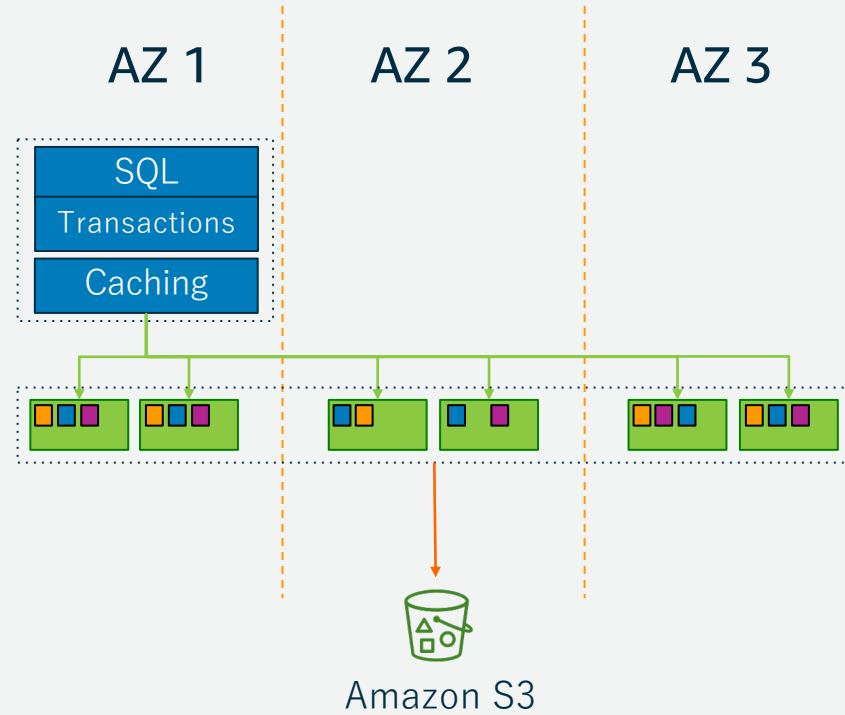


- **Amazon Aurora DB クラスター**
 - Amazon Aurora の管理単位
 - プライマリインスタンス、レプリカ、クラスター ボリュームの総称
- **プライマリインスタンス (Writer)**
 - 読み込み、書き込みを行うインスタンス
- **レプリカ (Reader)**
 - 読み込みをスケールアウトさせるレプリカ (15台まで作成可能)
- **クラスター ボリューム (ストレージ)**
 - 3つの AZ 間でレプリケートされる仮想ボリューム
 - プライマリインスタンスもレプリカも同じクラスター ボリュームを利用
- **エンドポイント**
 - Aurora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

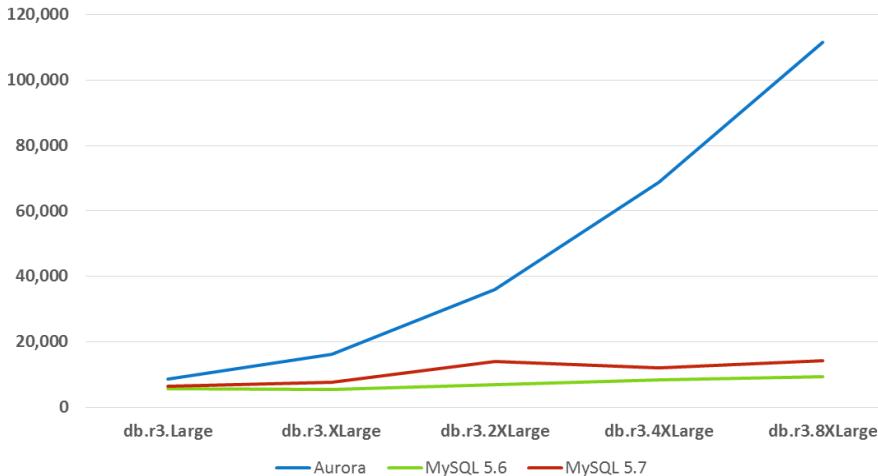
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



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

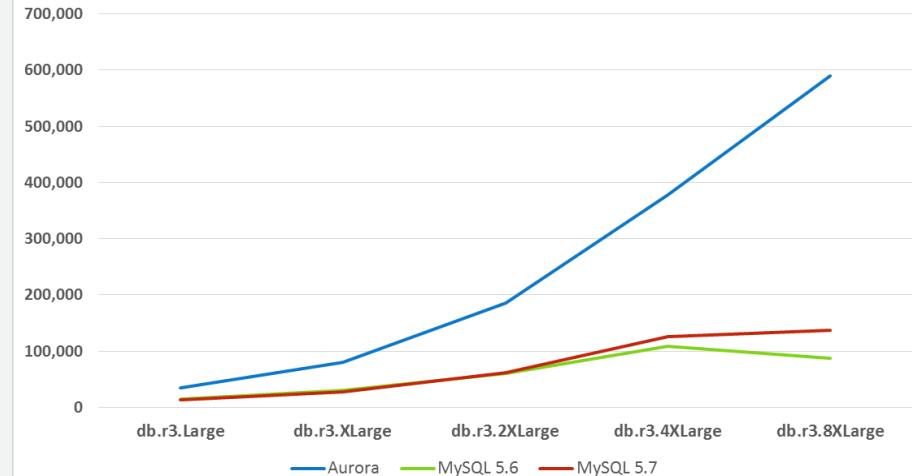
WRITE PERFORMANCE

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



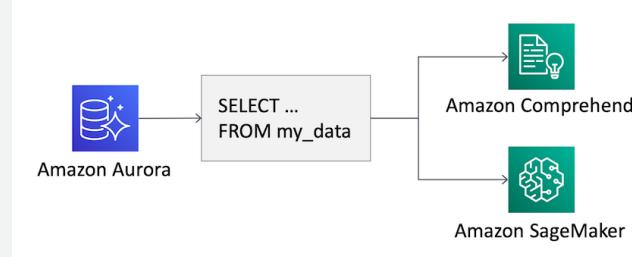
Aurora Global Database に関する複数のアップデート

- 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



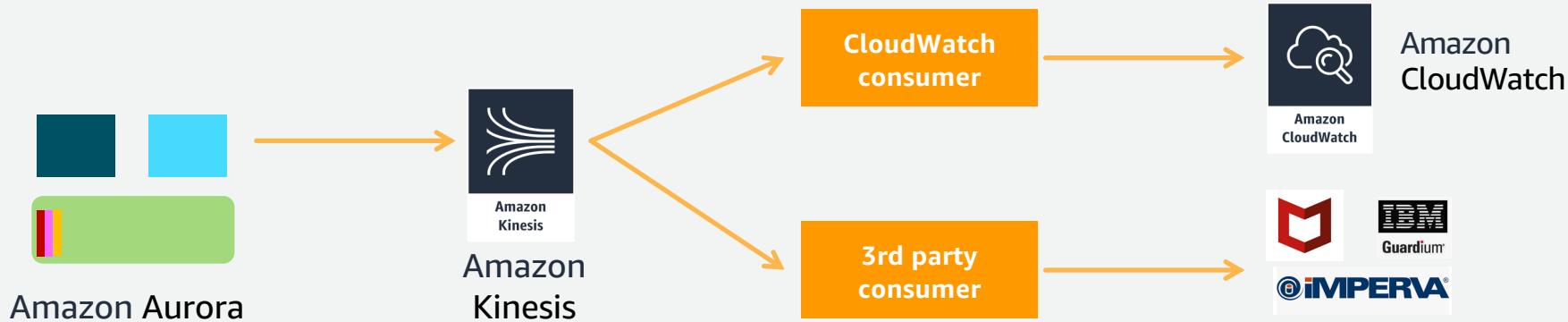
詳細：<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-ml.html>

© 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

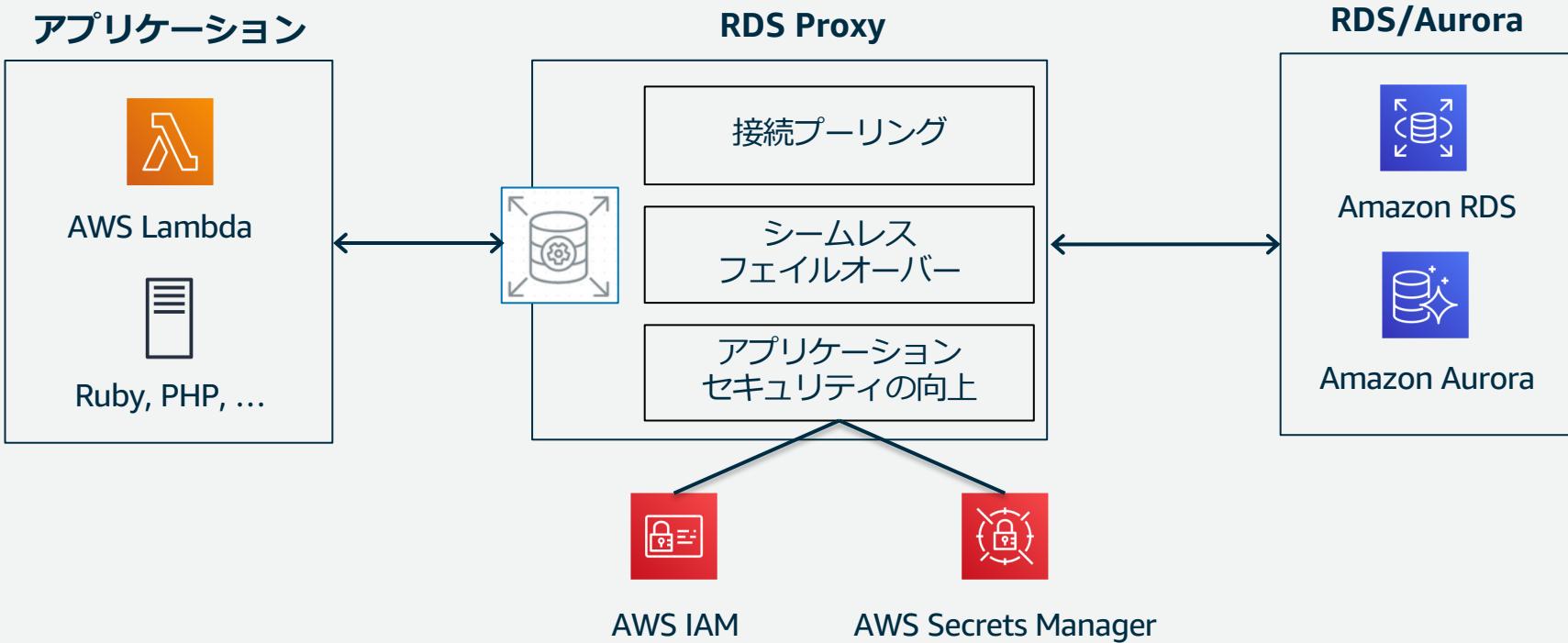


データベースアクセス制御で、アプリケーションデータのセキュリティを管理



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

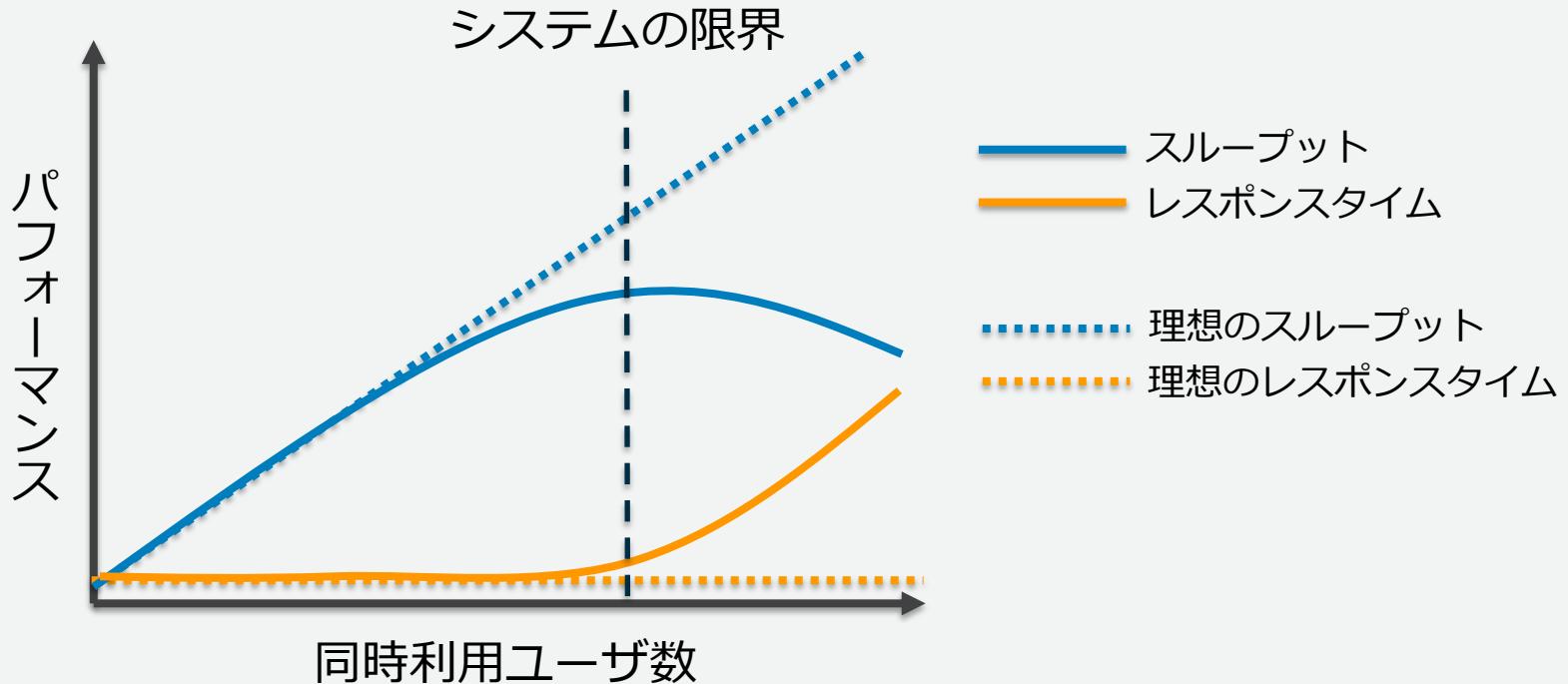
- 最大ストレージサイズが 128 TiB に拡張
 - Aurora MySQL 1.23.x および 2.09.x より対応
<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Updates.1230.html>
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Updates.2090.html
- ストレージサイズの上限に近づくと AWS マネジメントコンソールの RDS Event 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

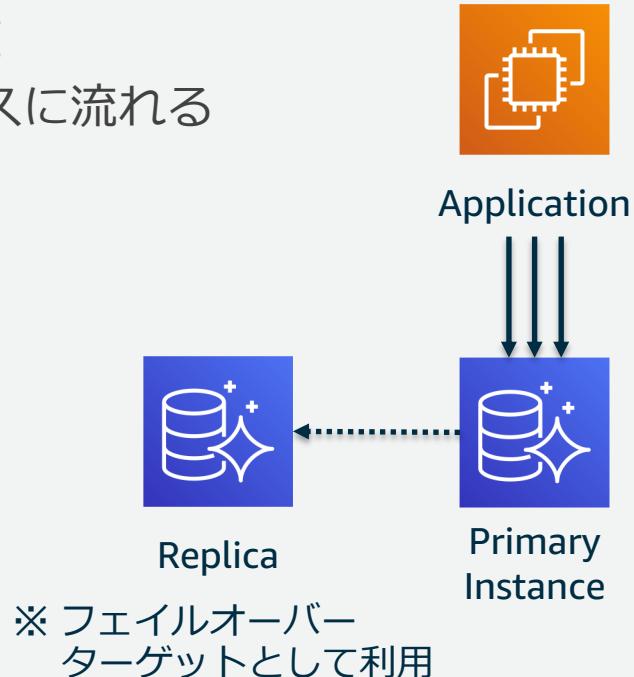
- サービスの成長に伴い、ユーザが増えていく
 - データ量が増え、キャッシュメモリに乗らなくなる
 - 新機能の追加による、当初想定していない負荷がかかる
- 多くのアプリケーションは状態を持たず、スケールさせやすい一方、データベースはボトルネックになりやすい
- (特にデータベースでは) 負荷に耐えられないと、**ビジネスの機会損失に直結する**

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

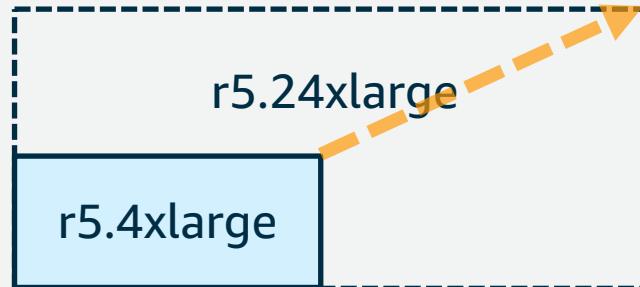
- ・ シングルプライマリ構成
 - ・ 全てのテーブルが1インスタンスに存在
 - ・ 全てのクエリがプライマリインスタンスに流れる
- ・ 初期から過度に作り込み過ぎない。
シンプルな設計から検討する



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

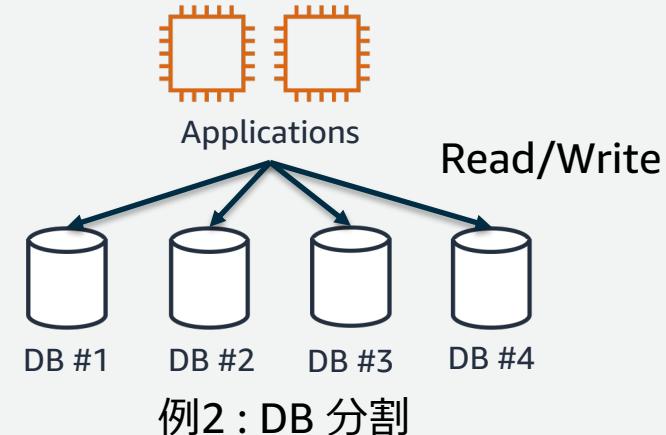
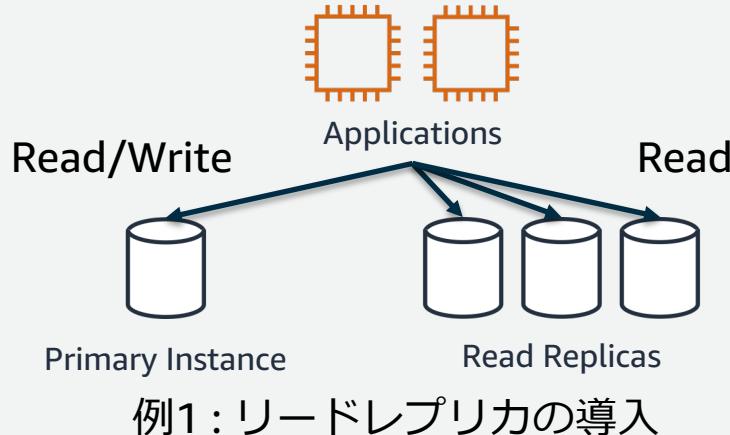


Aurora では、インスタンスサイズの変更により vCPU, メモリ, NW パフォーマンス, ローカルストレージの帯域幅に影響がある

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

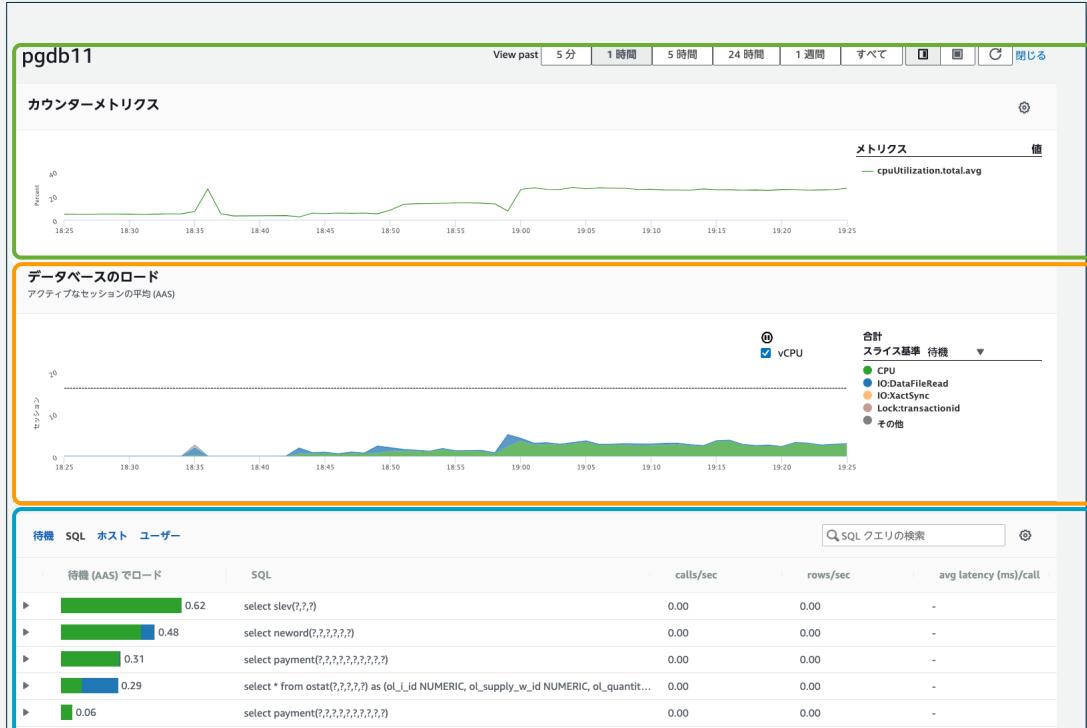
- インデックスを貼る・ヒント句の利用などによる実行計画の制御
- クエリの書き換え
- パラメータチューニング



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

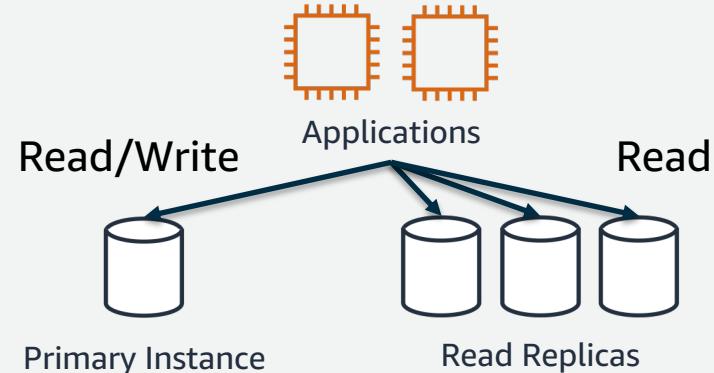
#1-a. リードレプリカの利用

フィットするケース

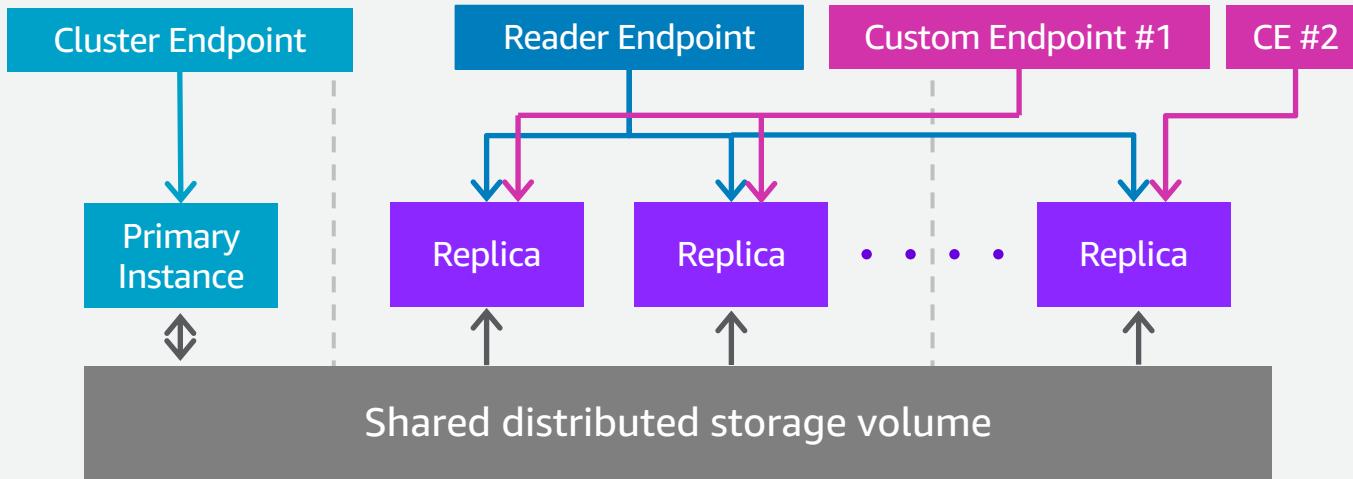
- 参照系クエリの負荷分散をしたい

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

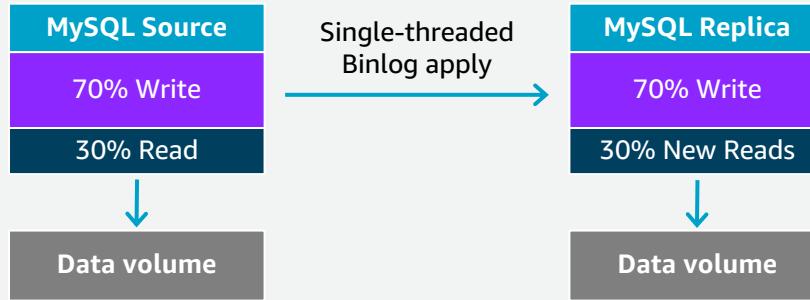
詳細：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Replication.html

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

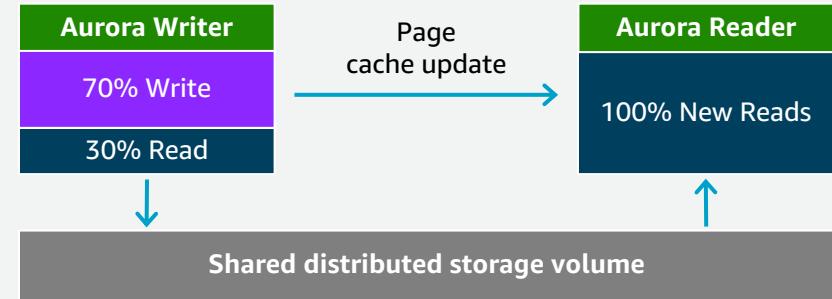


通常の MySQL のレプリカと Aurora レプリカの違い

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

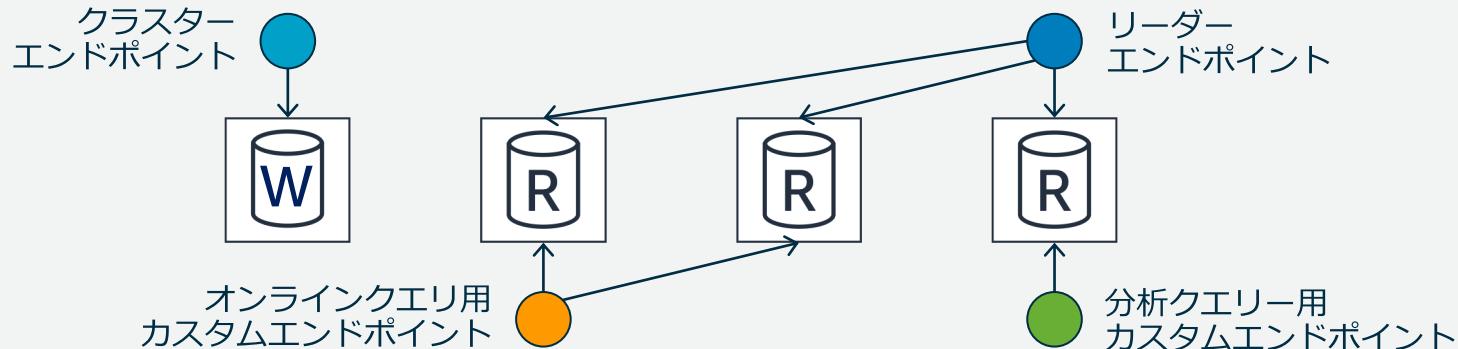
レプリカに書き込みは 発生しない

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.

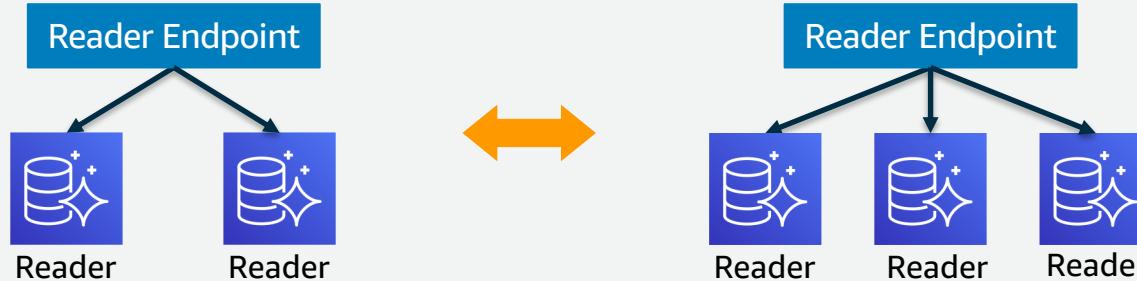


Aurora レプリカでの Amazon Aurora Auto Scaling

- 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)



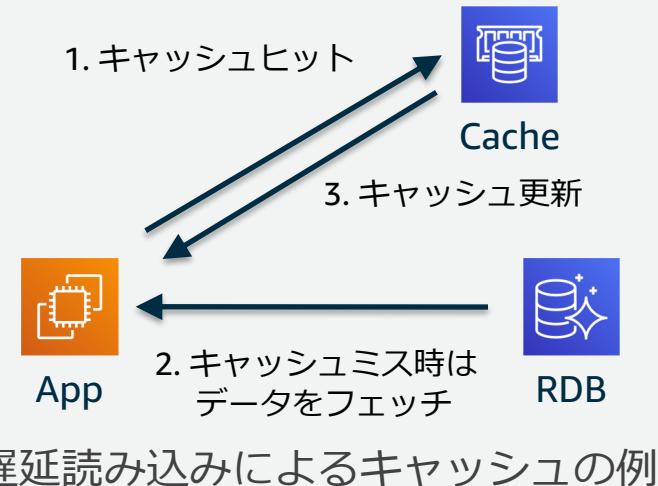
#1-b. [参考] キャッシュレイヤの導入

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

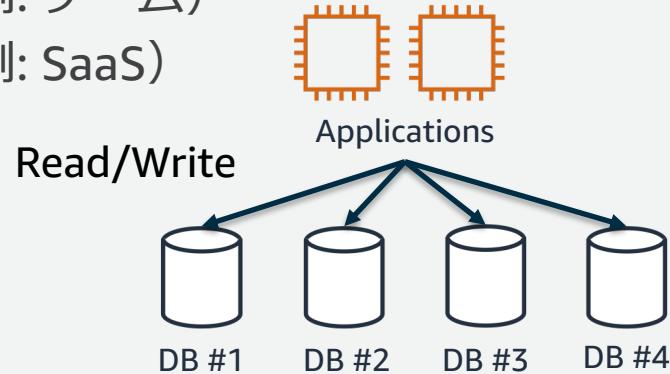
#2-a. データベース分割による負荷分散

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

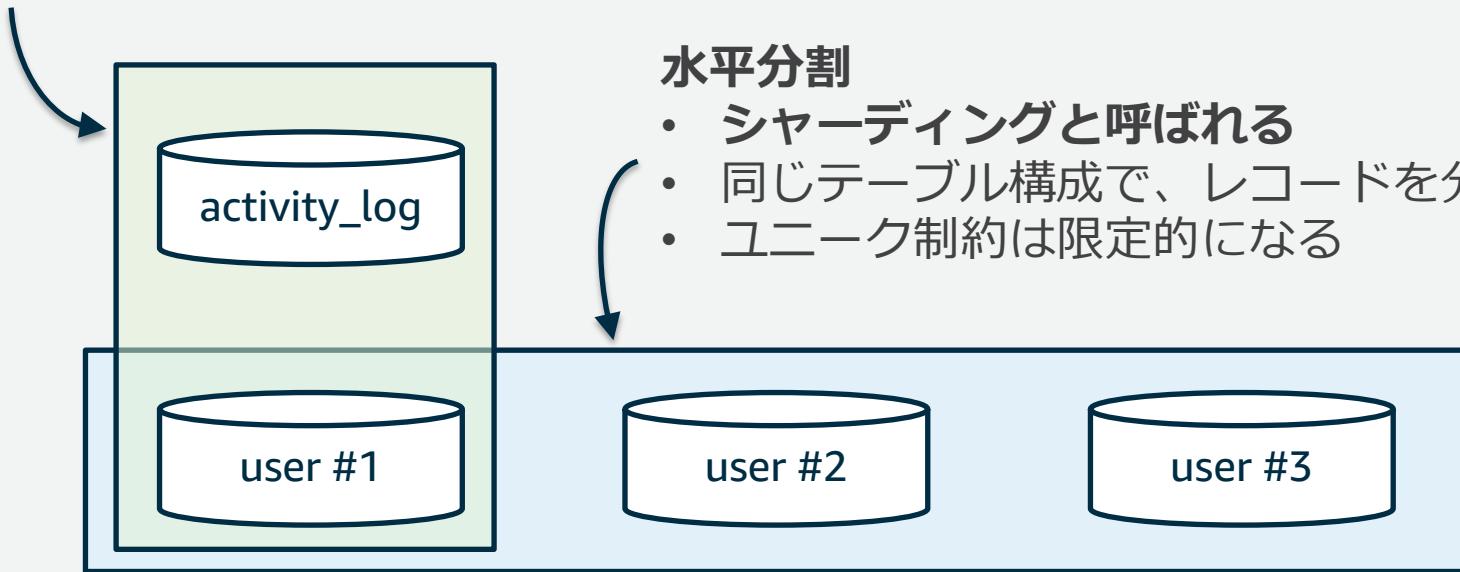
- データ分散のアルゴリズムの考慮
 - どういったルールでデータを振り分けるか
 - シャーディングでは、リシャードによるデータ移動が発生する
- 分割した DB/シャード をまたがるトランザクションは難しい



垂直分割と水平分割

垂直分割

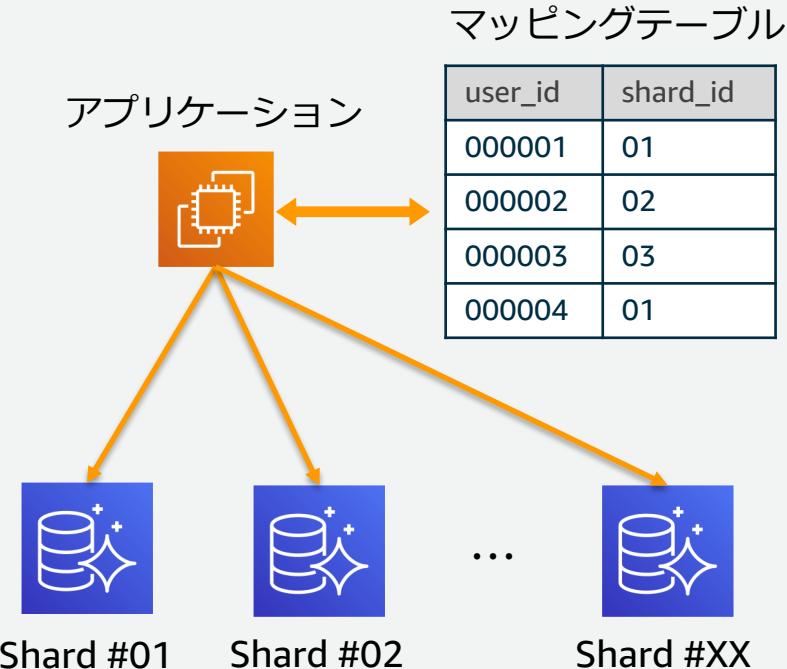
- ・ テーブル単位で分割
- ・ ログテーブルなど、**要件が全く異なるテーブル**に適用しやすい
- ・ ワークロードに偏りがある場合、スケールしない



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

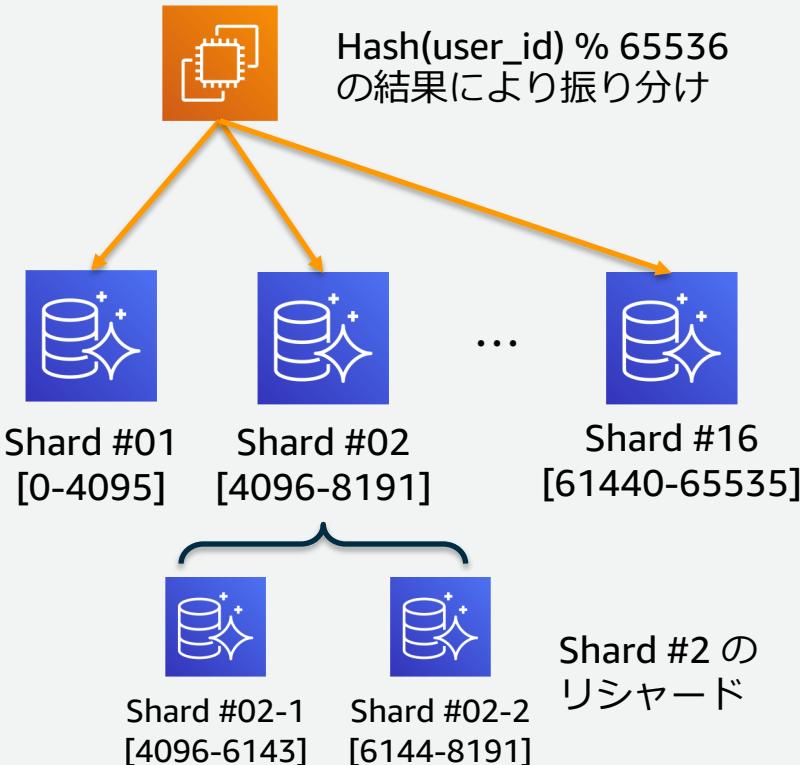
シャーディングロジックの例：マッピングテーブル



- 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

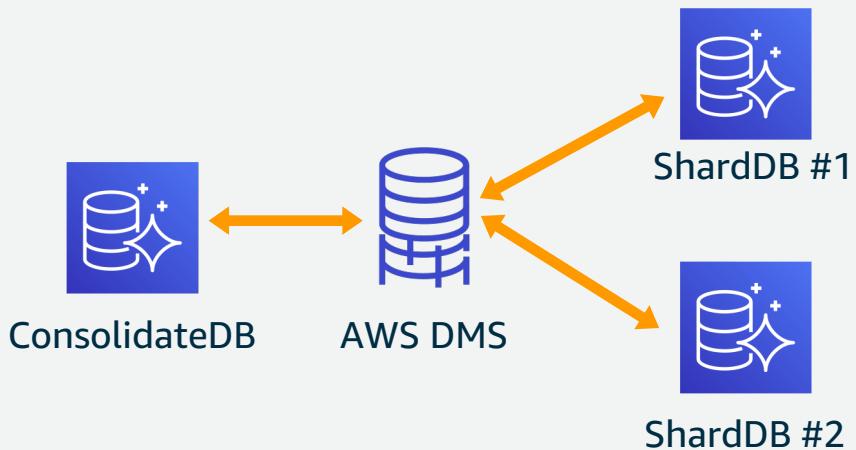
アプリケーション



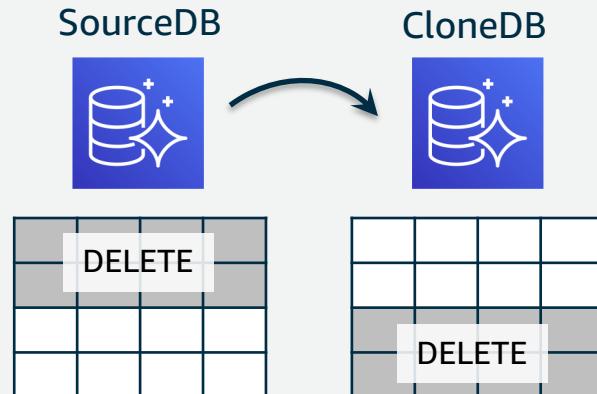
- 特定のカラムをもとに、
ハッシュ関数と modulo により
取りうる範囲の値を絞って分割
 - ハッシュ関数は crc32 などを利用
- $Hash(value) \% N$ の N は、
シャード数よりも十分に大きく設定
 - 再分割時のデータ移動を防ぐ
- 2のべき乗に分割しておき、
リシャードはインスタンスを
2倍に増やす方法がポピュラー

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

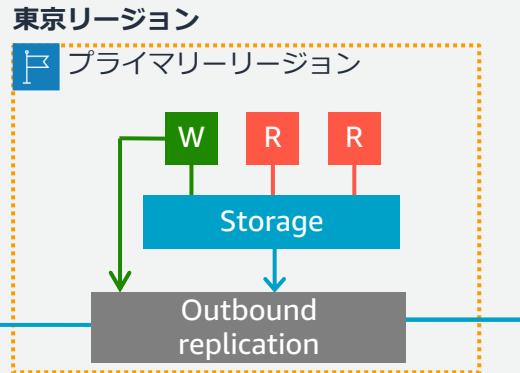
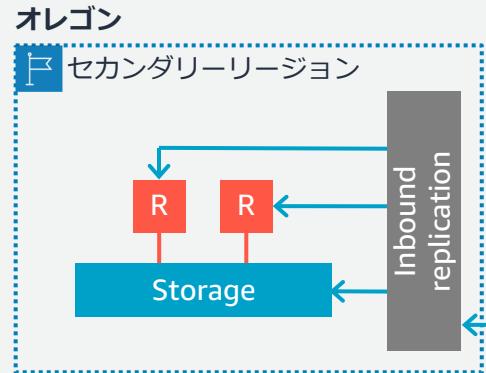
- データを他のリージョンにも持ちたい
 - データをアプリケーションに近づけ、レイテンシを減らす
 - 災害対策（DR）

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

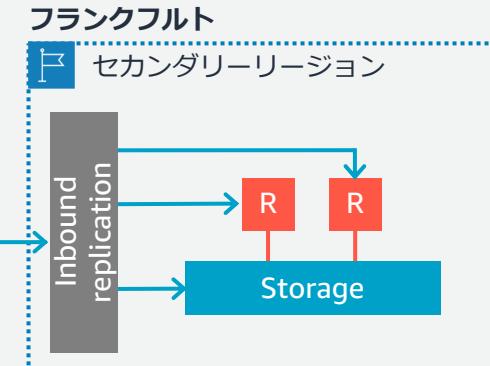
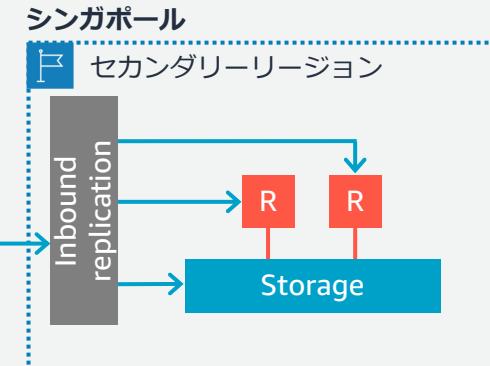
低いレプリカラグ: 高負荷状態でもリージョン間でレプリカラグは1秒未満

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

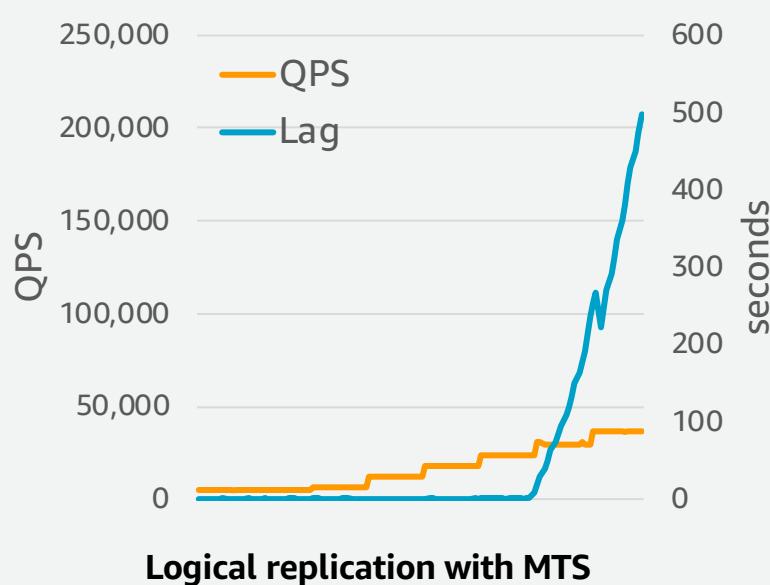
詳細 : https://docs.aws.amazon.com/ja_ip/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

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

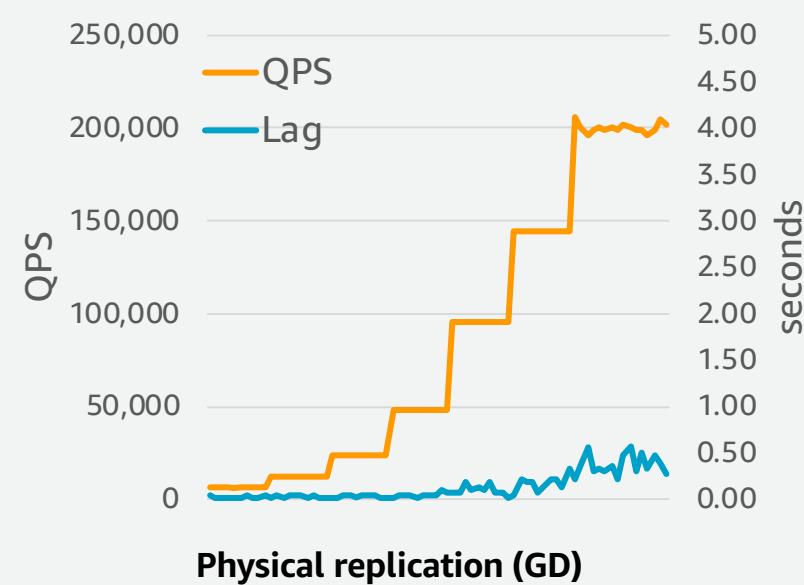


グローバルデータベースのパフォーマンス

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.

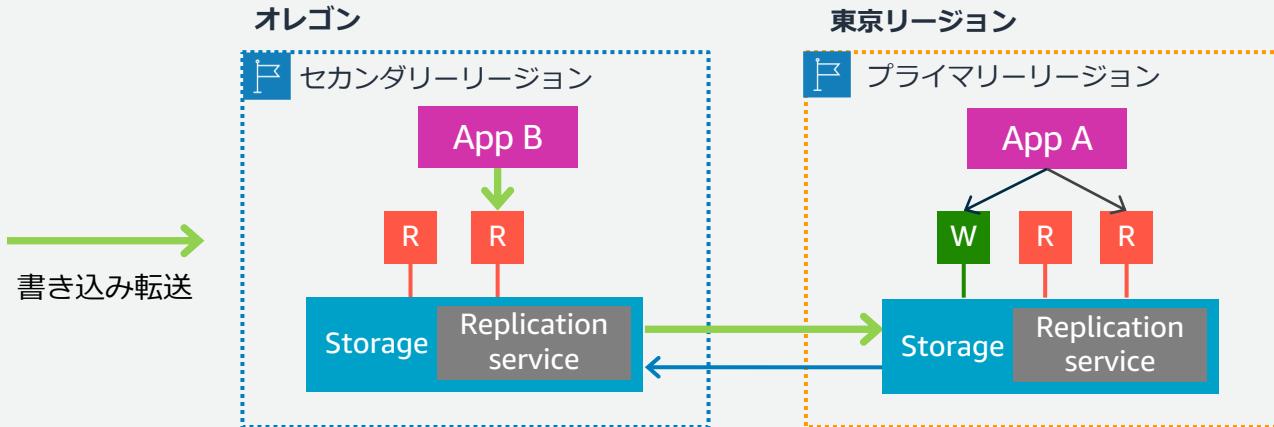


Aurora グローバルデータベース: 書き込み転送

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



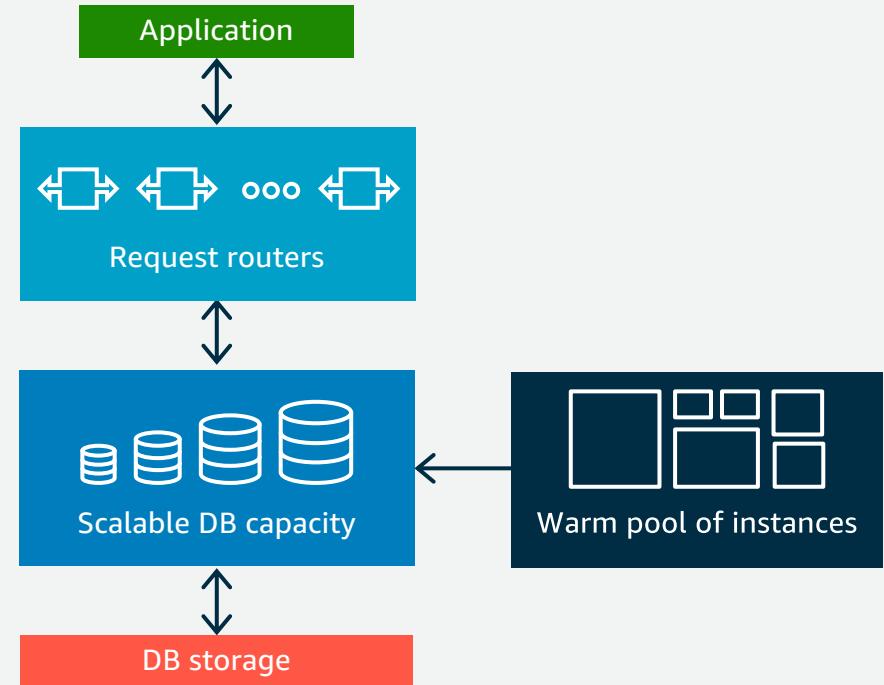
キーポイント

- 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

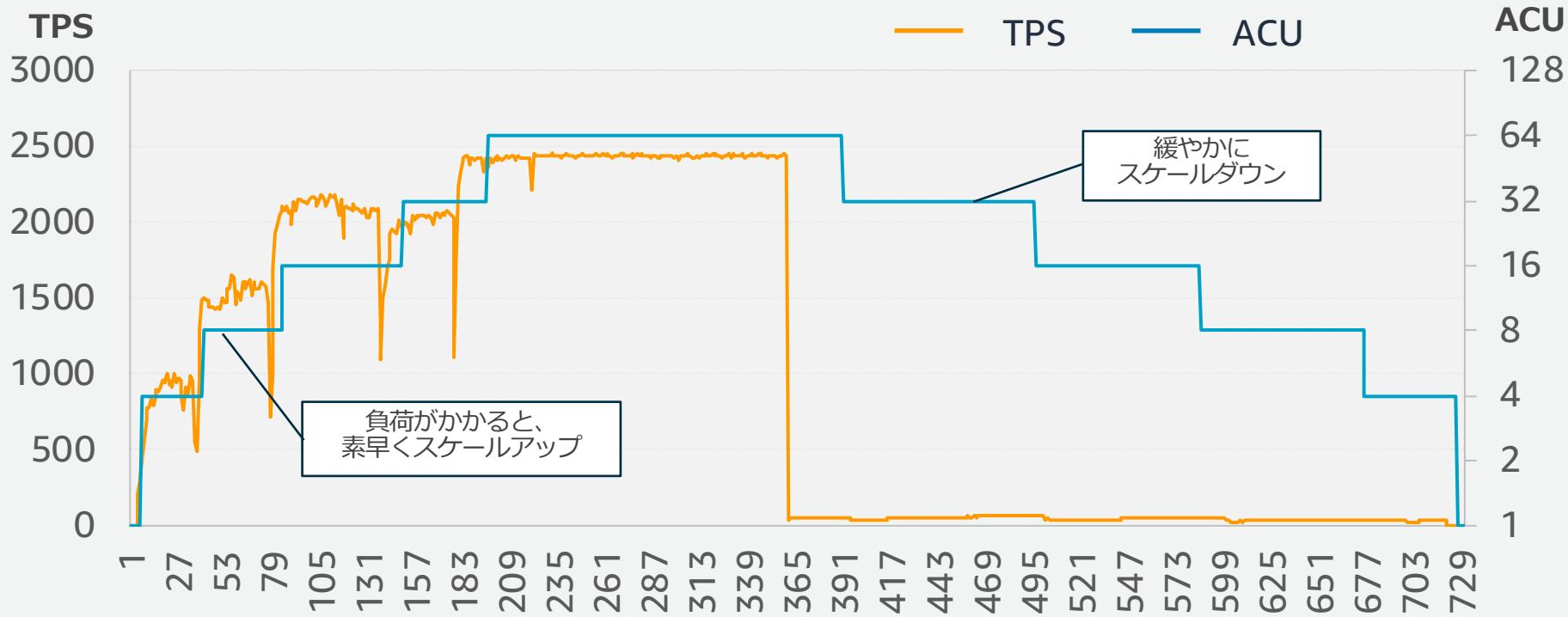
コンピューティングキャパシティーが自動管理される Amazon Aurora

- 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 page. 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", "サポート", "日本語", "アカウント", and "コンソールにサインイン". Below the navigation is a menu bar with links for "製品", "ソリューション", "料金", "ドキュメント", "学習", "パートナー", "AWS Marketplace", "その他", 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" and a descriptive paragraph about the resources available. At the bottom, there are four buttons: "AWS Webinar お申込", "AWS 初心者向け", "業種・ソリューション別資料", and "サービス別資料".

日本担当チームへお問い合わせ サポート 日本語 アカウント コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 Q

AWS クラウドサービス活用資料集トップ

Amazon Web Services (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 » 業種・ソリューション別資料 » サービス別資料 »

<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>



AWS Black Belt Online Seminar

Amazon Aurora Global Database

石渡 嘉之

Technical Account Manager
2025/09



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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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

石渡 嘉之

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ー

エンタープライズサポートをご利用されている
お客様のクラウド活用のご支援をしています。

好きな AWS サービス
Amazon Aurora, AWS Support



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おり、データベースの地理的冗長性に関心のある方
- ・Amazon Aurora 及び Amazon Aurora Global Database の利用を検討中、または今後検討をご予定の方
- ・Amazon Aurora Global Database の概要、他のレプリケーションソリューションとの違いを理解したい方

アジェンダ

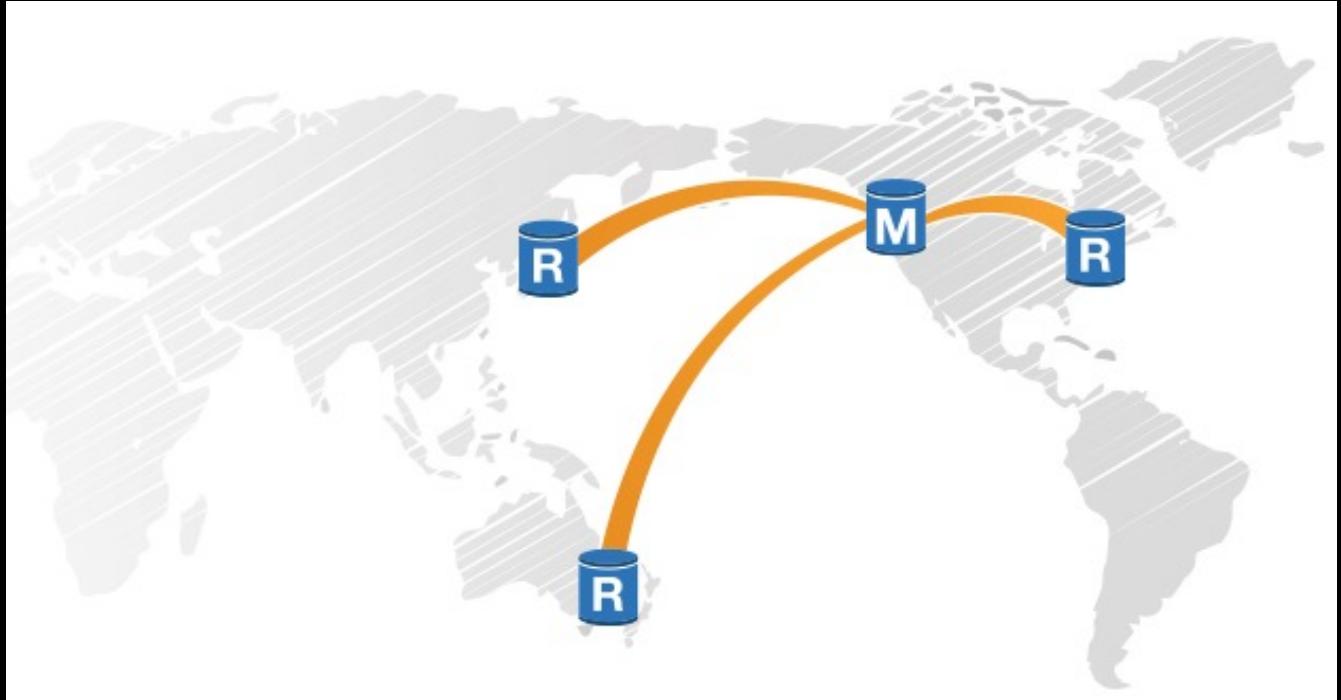
1. Amazon Aurora Global Database 概要
 - Amazon Aurora Global Database アーキテクチャと特徴
2. クロスリージョンクラスター切り替え
 - スイッチオーバー
 - フェイルオーバー
3. Amazon Aurora Global Database その他の主な機能と制限事項
 - 書き込み転送
 - RPO 管理
 - ヘッドレスクラスター
 - 制限事項

Amazon Aurora Global Database 概要

Amazon Aurora Global Database とは

複数の AWS リージョンにまたがる複数の Aurora DB クラスターを構成。

- ディザスタリカバリ
大規模障害におけるデータベース
の地理的冗長性を実現
- データの近接配置
異なるリージョンのユーザーの近
くにデータを配置し、高速な読み
取りが可能

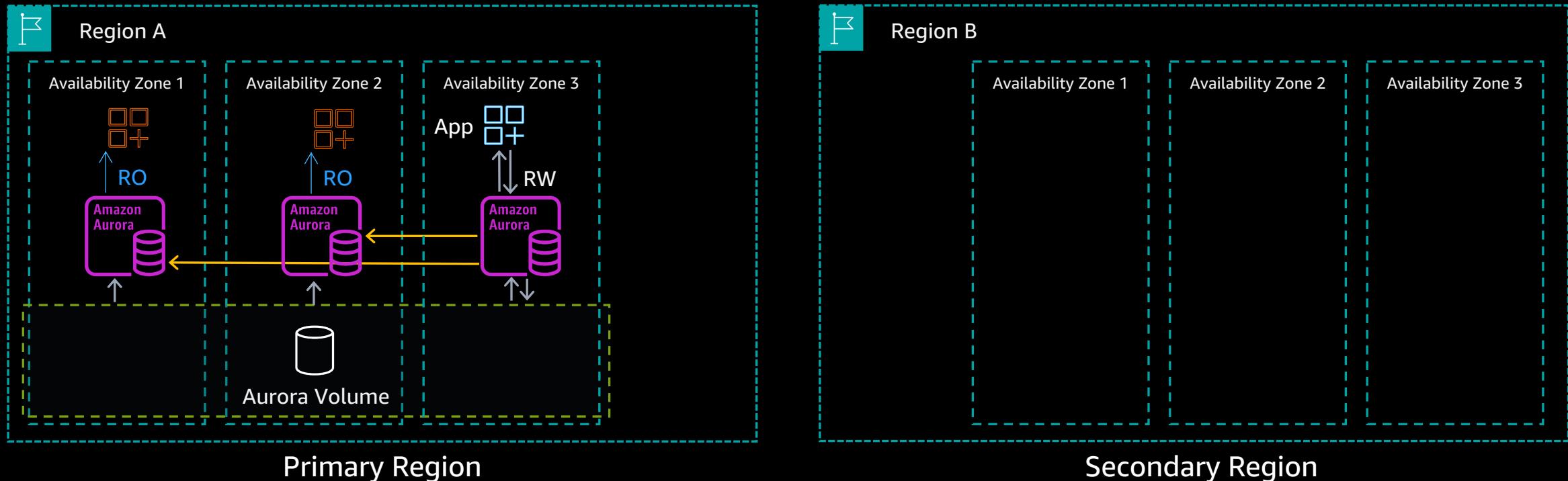


※ Aurora Global Database 概要についてはこちらでも紹介しています
Amazon Aurora 概要編 – 可用性 – 後半 : 資料 ([PDF](#)) | 動画 ([YouTube](#))

Aurora Global Database アーキテクチャ

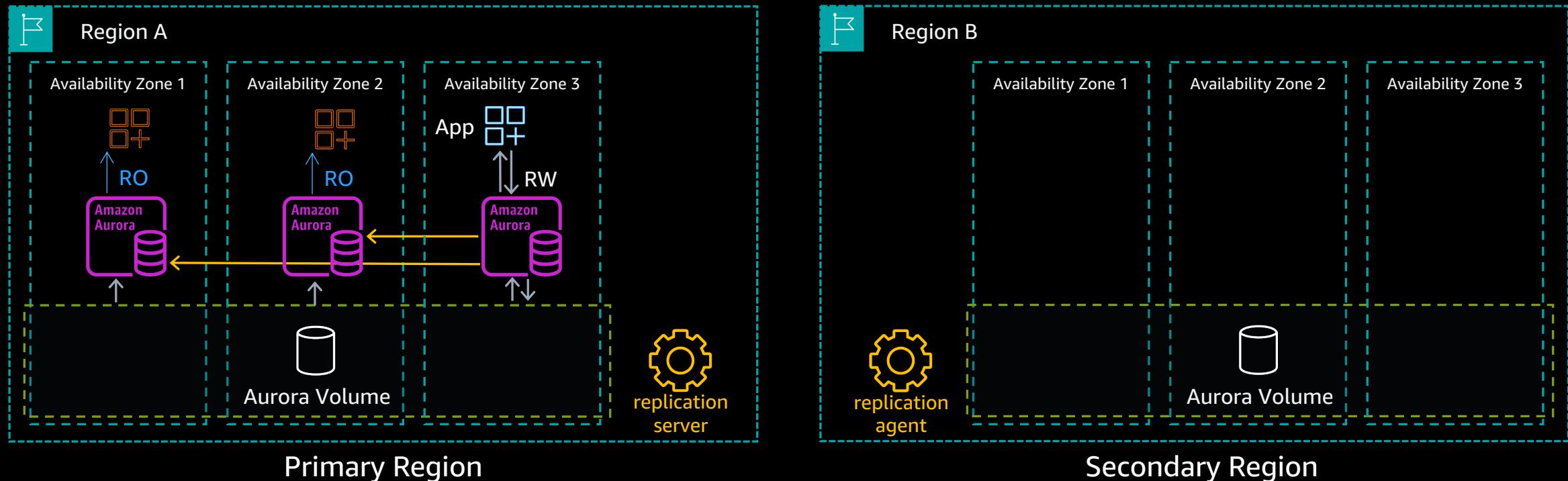
スタンドアロン Aurora DB クラスター

※ Aurora Global Database を利用していないケース



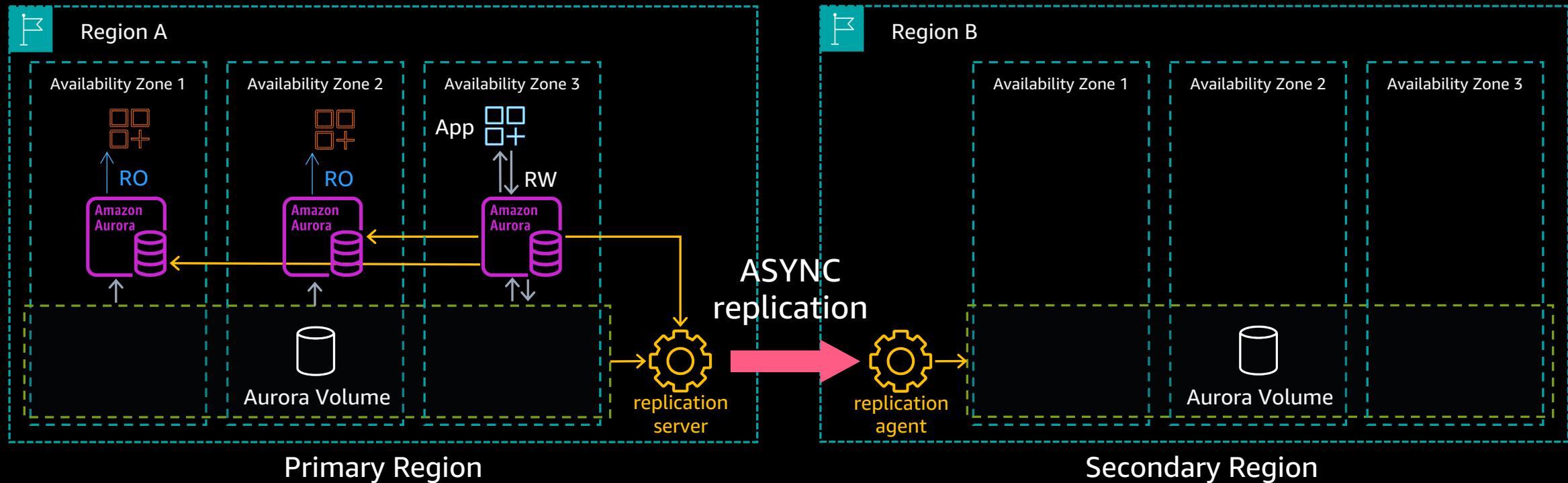
Aurora Global Database アーキテクチャ

Aurora Global Database



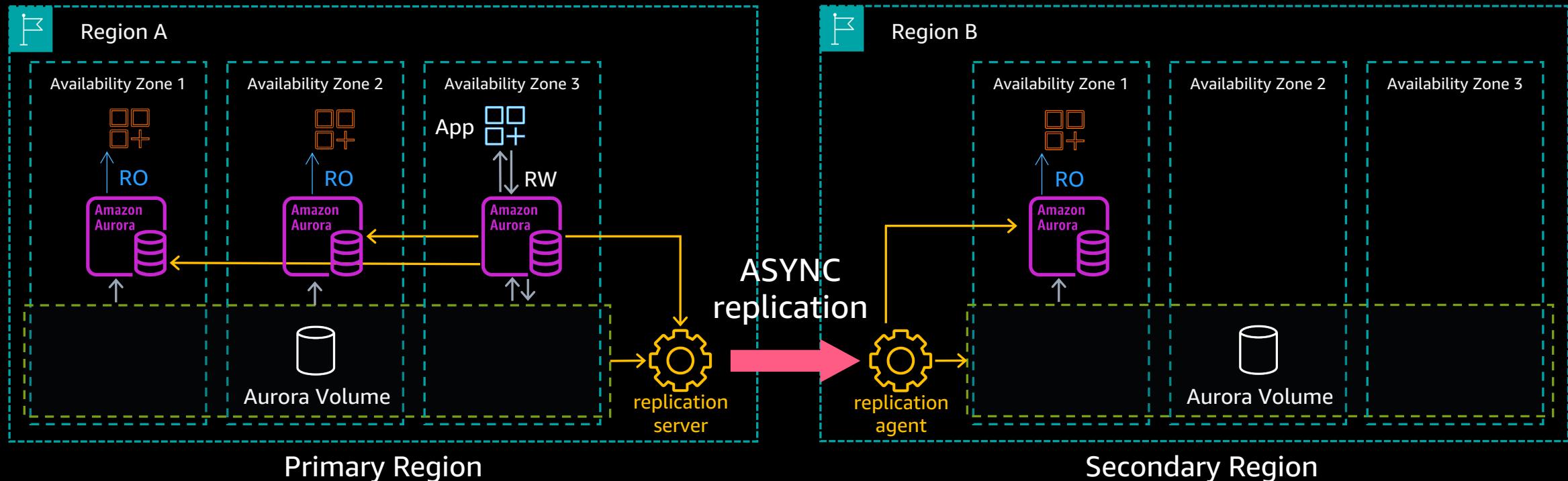
Aurora Global Database アーキテクチャ

Aurora Global Database



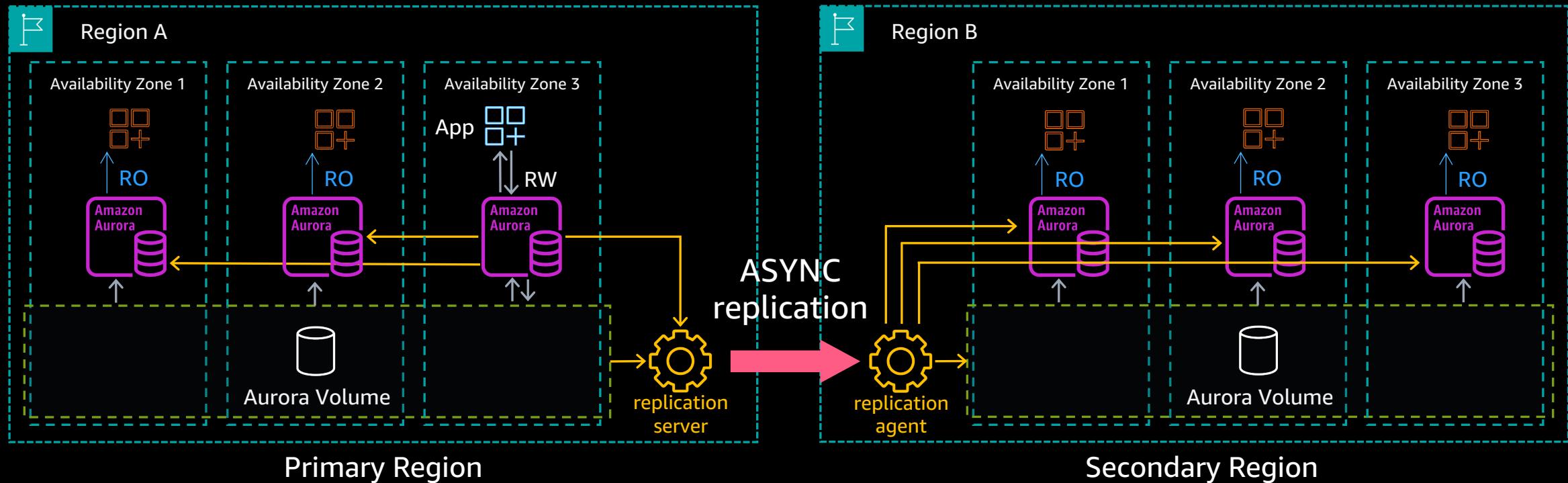
Aurora Global Database アーキテクチャ

Aurora Global Database

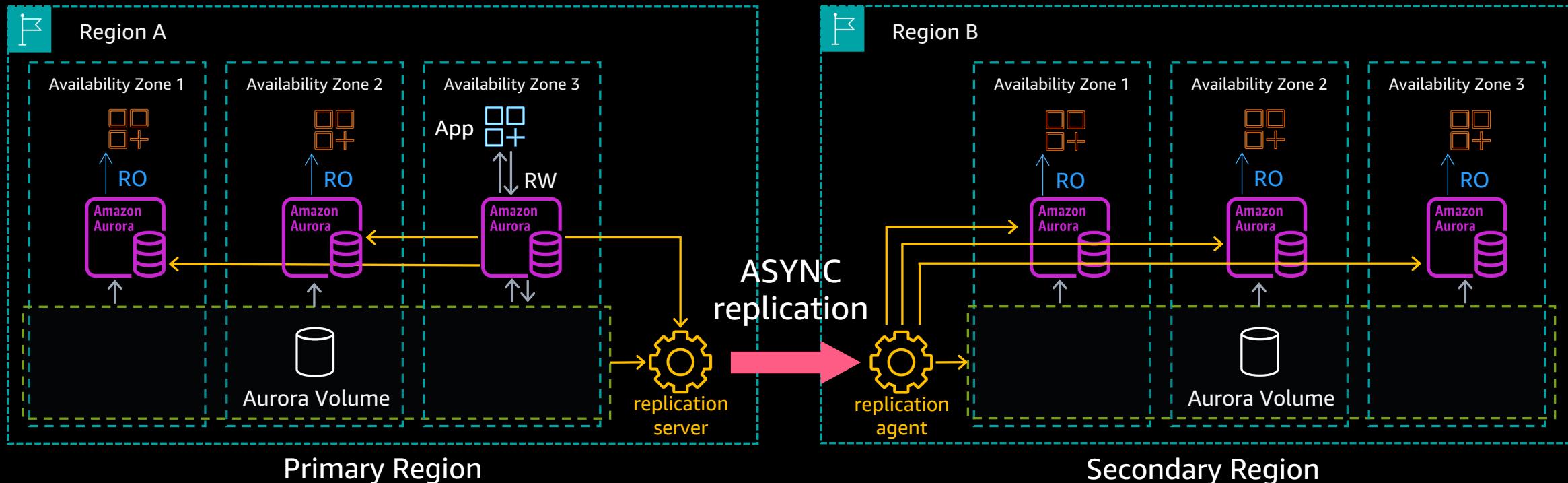


Aurora Global Database アーキテクチャ

Aurora Global Database

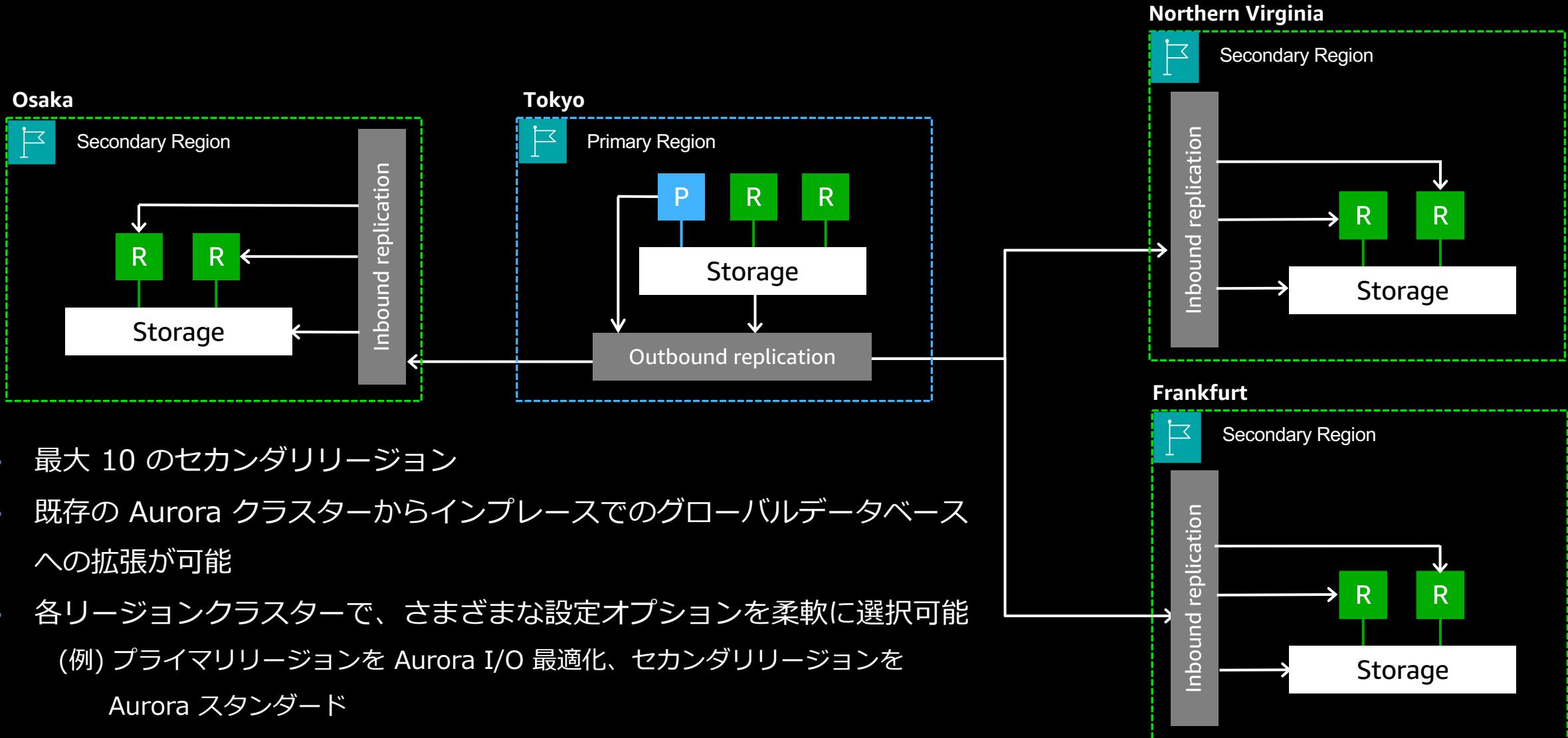


Aurora Global Database 特徴



- インスタンスへの影響が少ないレプリケーション
- 通常 1 秒未満の低いリージョン間レプリケーションラグ
- 計画的なスイッチャーバー: データ損失無しの切り替え
- 計画外のフェイルオーバー: 通常 1 分未満でリカバリ、少ないデータ損失

Aurora Global Database 柔軟な構成



Aurora Global Database 構成要件

- Aurora MySQL 及び Aurora PostgreSQL で利用可能
 - サポートされる AWS リージョンとエンジンバージョンはドキュメントを参照
 - [Supported Regions and DB engines for Aurora global databases](#)
- 構成可能なリーダーインスタンス (Aurora レプリカ) 数
 - セカンダリリージョン: 16
 - プライマリリージョン: 15 - 構成しているセカンダリリージョン数
- 一つのグローバルデータベースの各クラスターは異なるリージョンで構成 (同一のリージョンで構成は不可)
- メモリ最適化インスタンス (db.r5 以上のインスタンス推奨)
- 各 DB クラスター識別子は全リージョン間で一意 (異なるリージョンでも同一名称は不可)
- Aurora Serverless v2 を利用する場合、プライマリリージョンでは推奨される最小 ACU 8

詳細はドキュメント参照 : [Configuration requirements of an Amazon Aurora global database](#)



論理レプリケーションソリューションとの比較

Aurora Global Database の優位性

- ストレージレイヤーでのレプリケーション
 - インスタンスのパフォーマンスに影響しない
 - クエリが増加しても安定したレプリケーション
 - 低 RPO
- マネージドなクロスリージョン切り替え
 - 運用負荷低減
 - 低 RTO
- 多くのレプリカリージョンを構成可能

考慮事項

- レプリカに対して書き込み不可
- データベースバージョンの制約
- Global Database 固有の制約

クロスリージョンクラスター切り替え

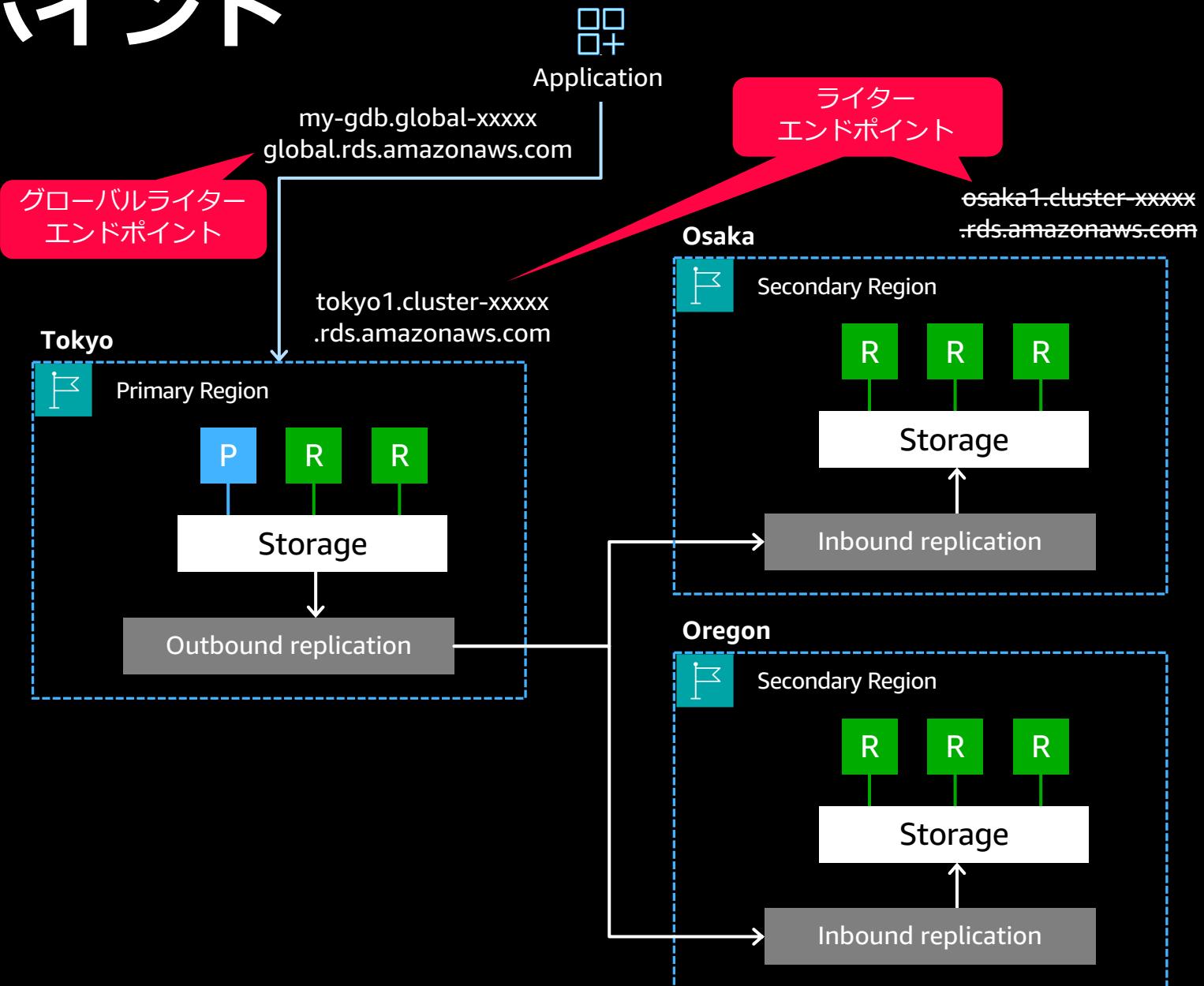
クロスリージョンクラスター切り替え

Aurora Global Database におけるクロスリージョンクラスターの切り替えは大きく 2 つのパターン。

- スイッチオーバー
 - 別リージョンにプライマリを計画的に移動
 - Aurora クラスターと関連するサービスが正常な状態で動作
 - データ損失無し (RPO 0)
- フェイルオーバー
 - プライマリとのレプリケーションを切り離し、セカンダリをプライマリに昇格
 - リージョン障害といった予期しない障害からの復旧
 - **データ損失が発生**
 - マネージドフェイルオーバーとマニュアルフェイルオーバー

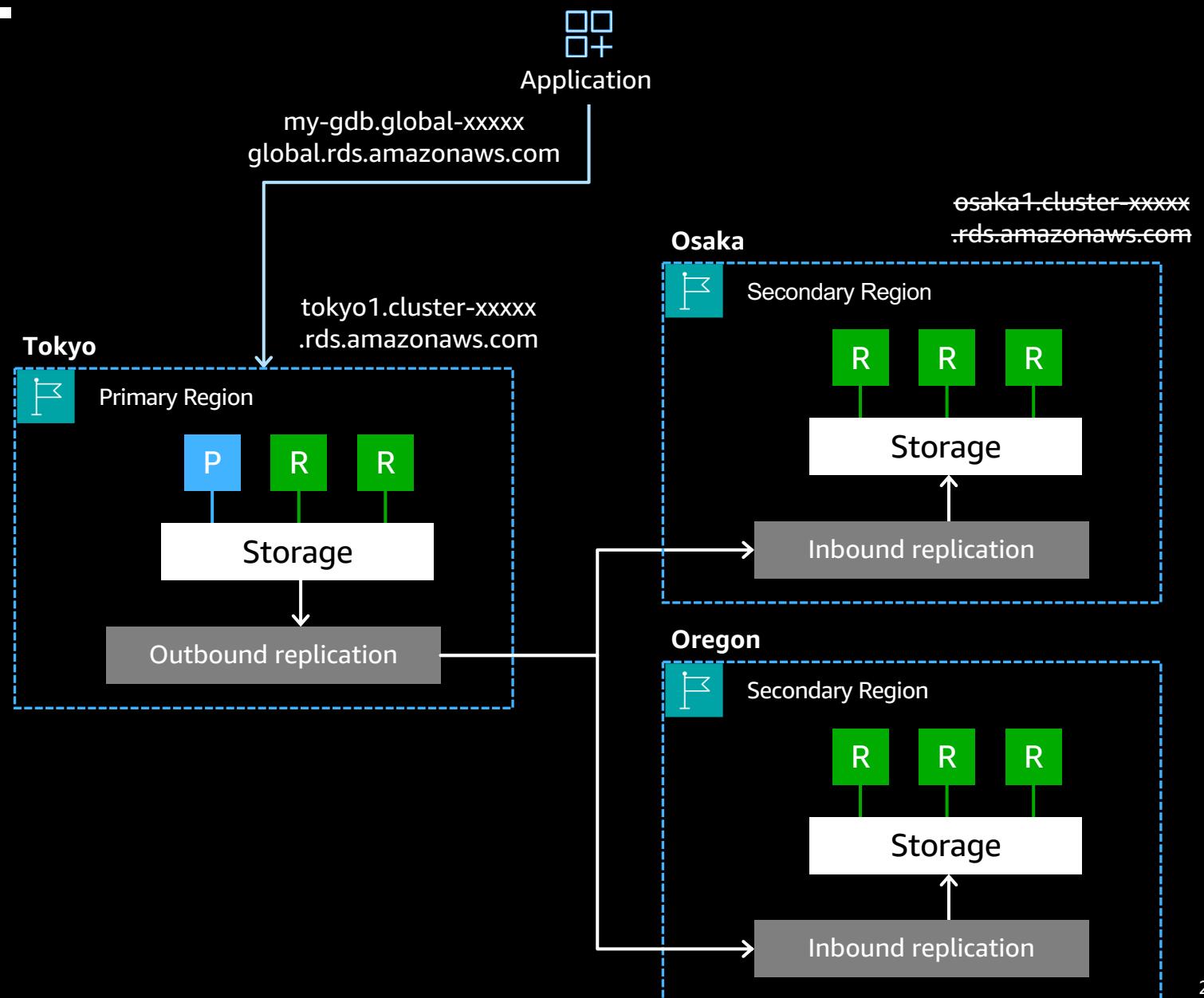
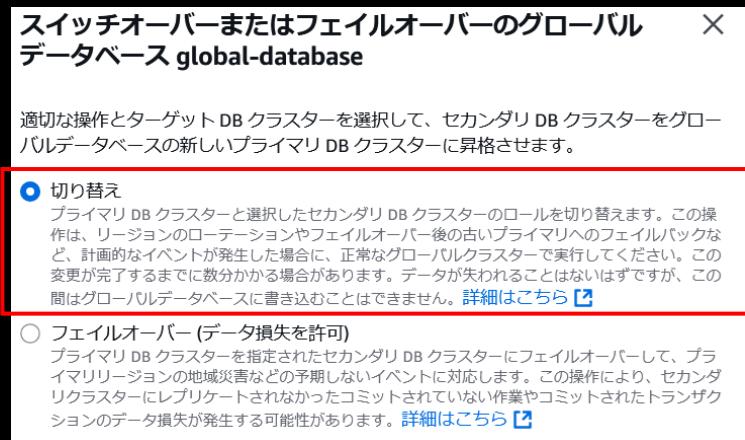
ライターエンドポイント

- ライターエンドポイントは各リージョンのクラスター毎に構成されている
- セカンダリリージョンでは inactive
- グローバルライターエンドポイントは、スイッチオーバー/フェイルオーバー時に自動で DNS エントリが更新される



スイッチャバー

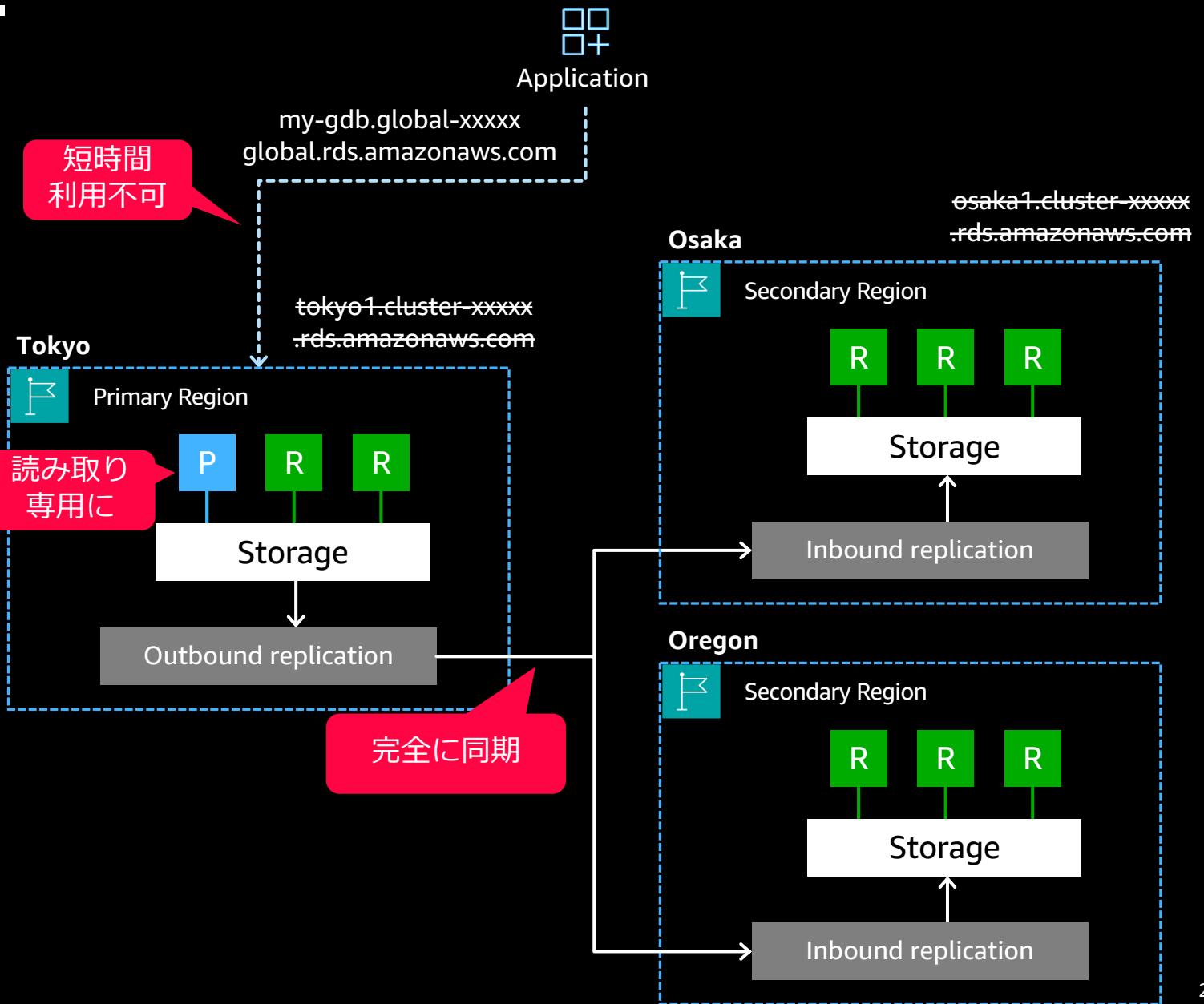
- スイッチャバーを実行
 - RDS コンソール



- AWS CLI
switchover-global-cluster

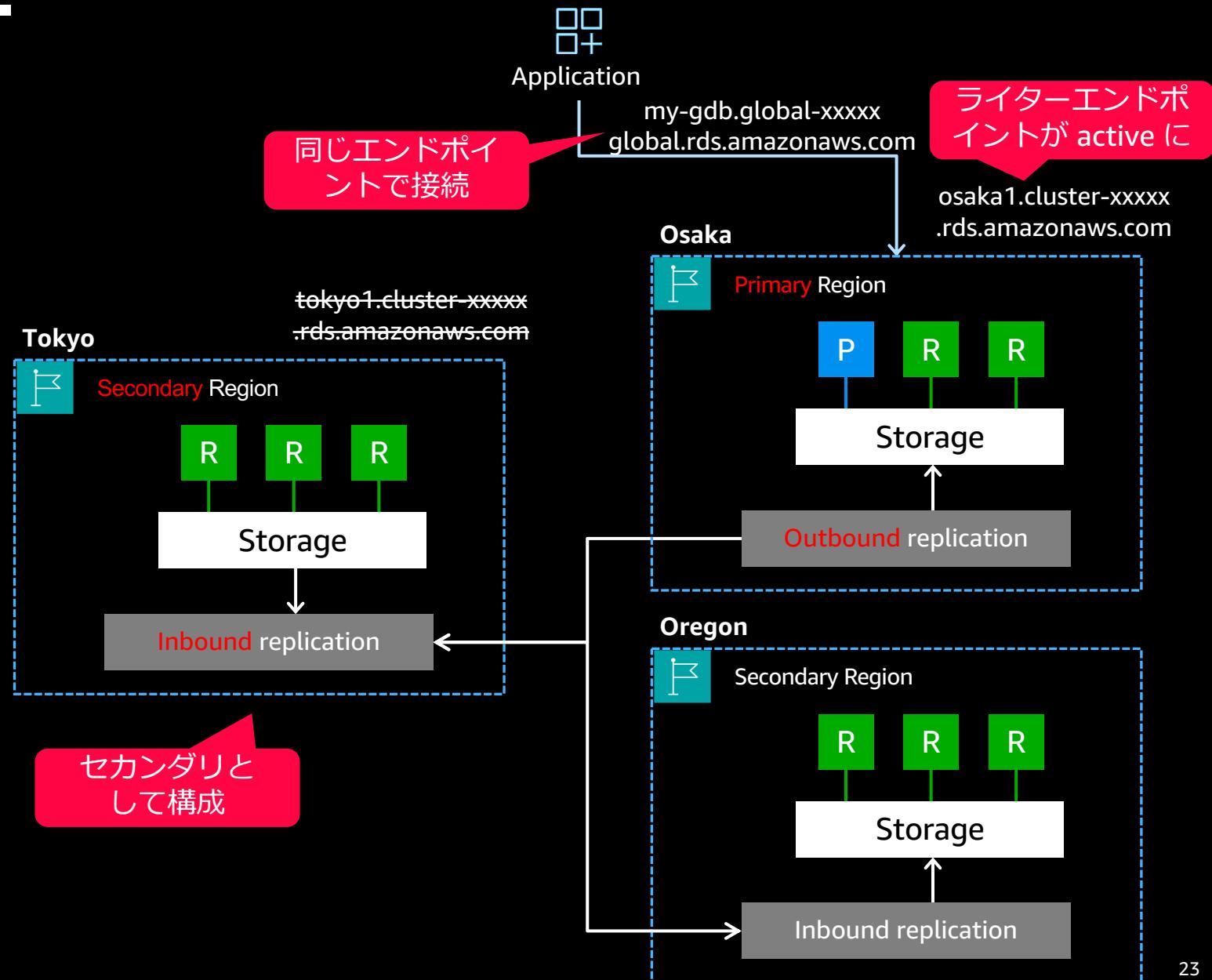
スイッチオーバー

- プライマリリージョンのライターインスタンスが読み取り専用に変更される
- 全てのデータがセカンダリクラスターに同期されていることを検証
- 短時間のデータベースが使用できない時間が発生
※ 特定バージョン以上では、通常 30 秒未満



スイッチオーバー

- セカンダリリージョンのクラスターがプライマリクラスターに昇格
- ライターエンドポイントが active になり、データベースが利用可能になる
- 以前のプライマリクラスターはセカンダリとして構成される

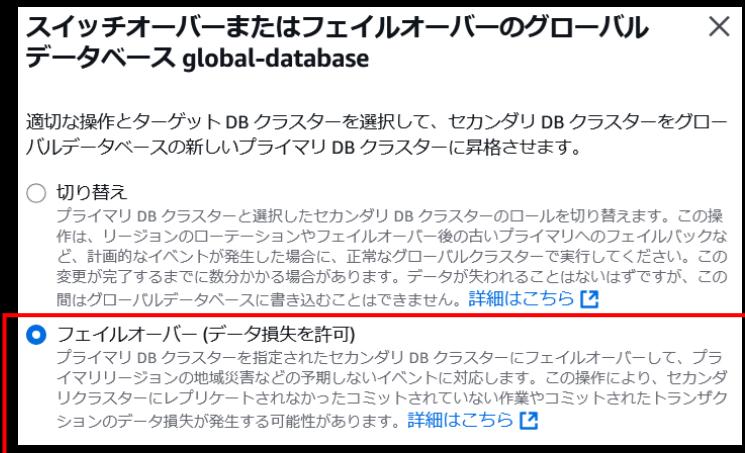


フェイルオーバー①

マネージドフェイルオーバー

フェイルオーバープロセスが自動で行われ、Global Database 構成が自動で復旧される。

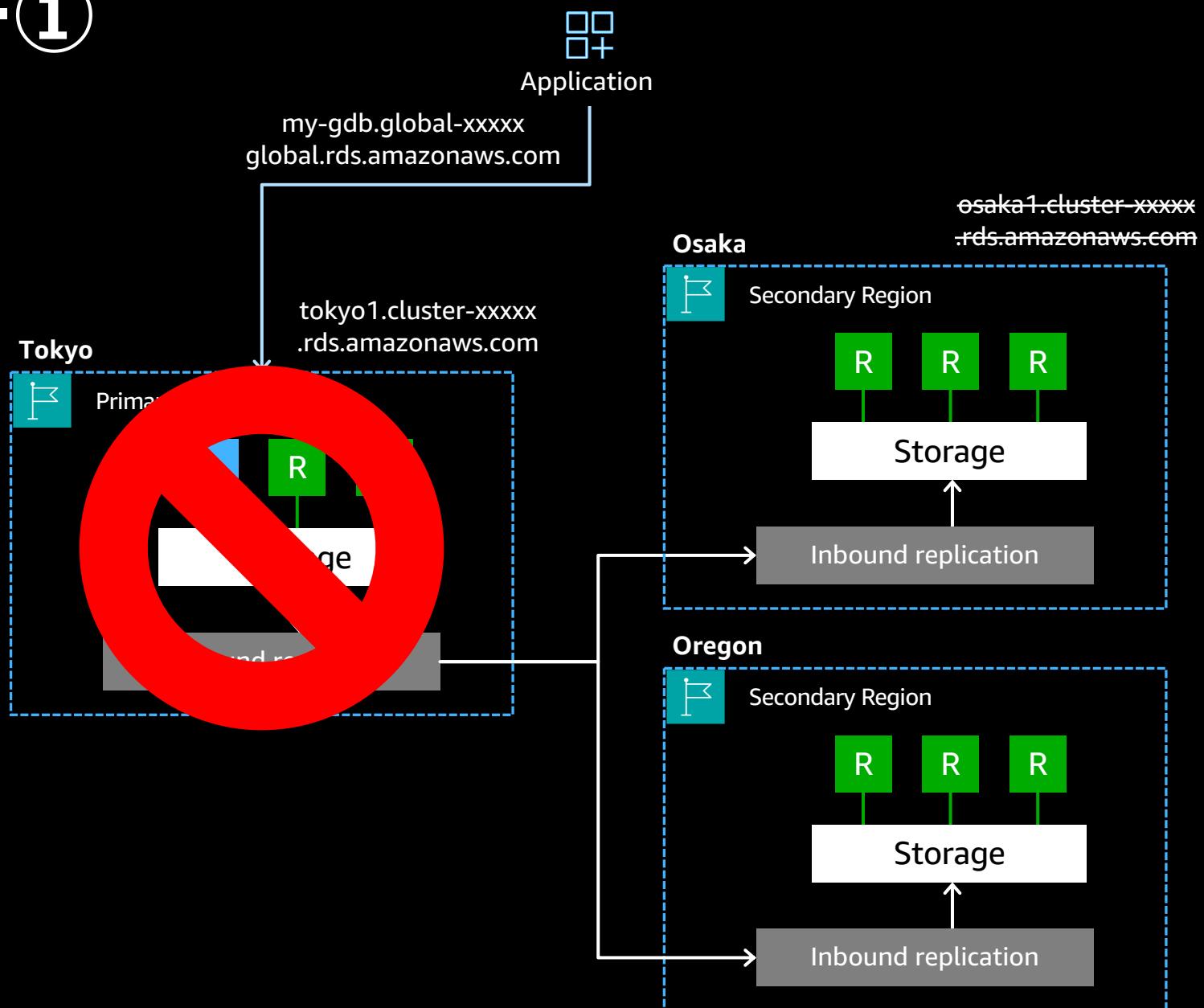
- フェイルオーバーを実行
 - RDS コンソール



- AWS CLI
`failover-global-cluster
--allow-data-loss`



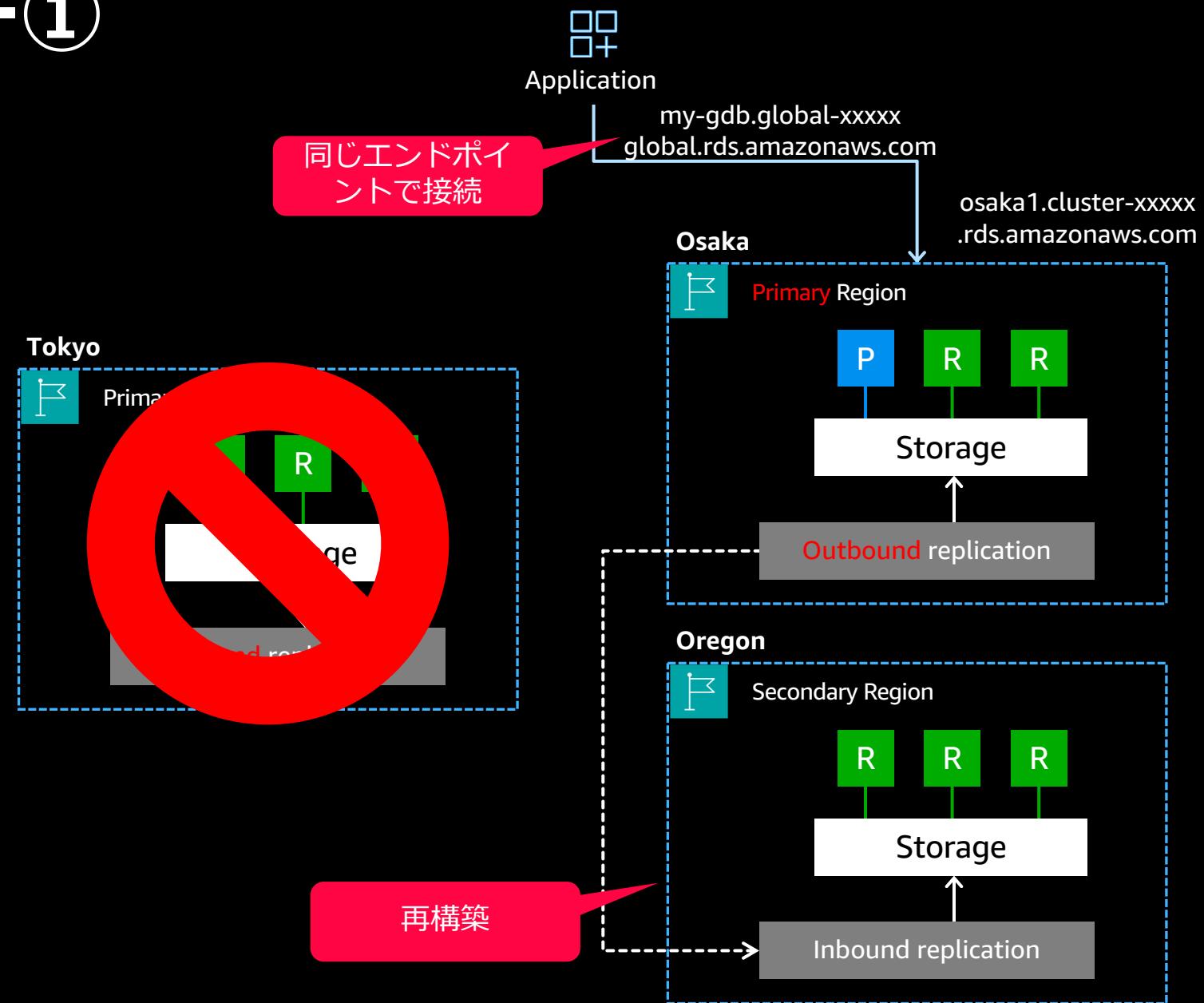
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



フェイルオーバー①

マネージドフェイルオーバー

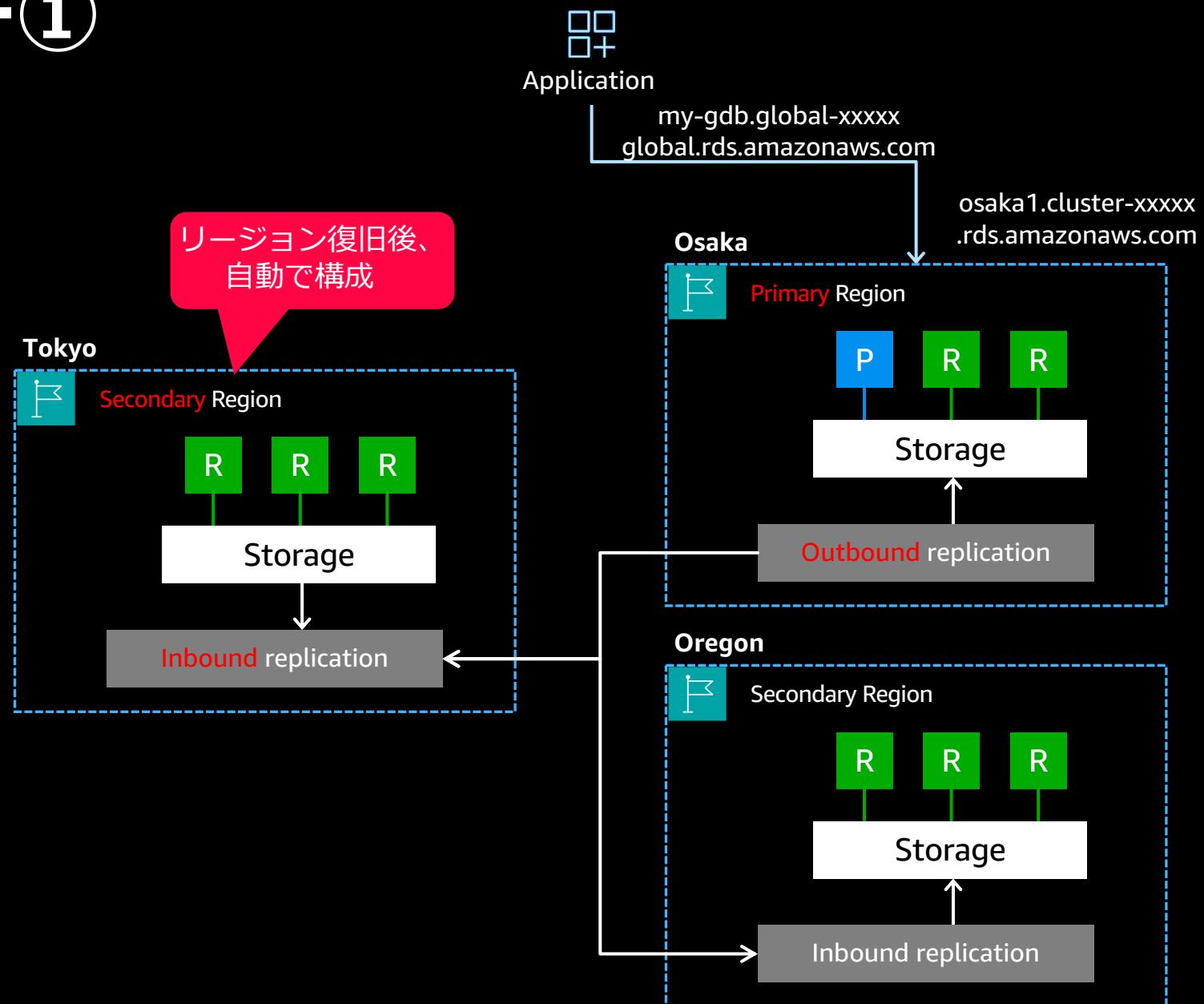
- セカンダリリージョンのクラスターがプライマリクラスターに昇格
- ライターエンドポイントが **active** になり、データベースが利用可能になる
- 他のセカンダリリージョンクラスターは再構築され、リーダーエンドポイントとして利用可能になる
※ 再構築完了までは、リージョン間の冗長性は失われている状態になる



フェイルオーバー①

マネージドフェイルオーバー

- 元のプライマリリージョンが利用可能になると、セカンダリリージョンとして自動的に構成される
- 元のプライマリクラスターの障害発生時点のストレージボリュームのスナップショットが作成される場合があり、欠落したデータの復旧に活用可能

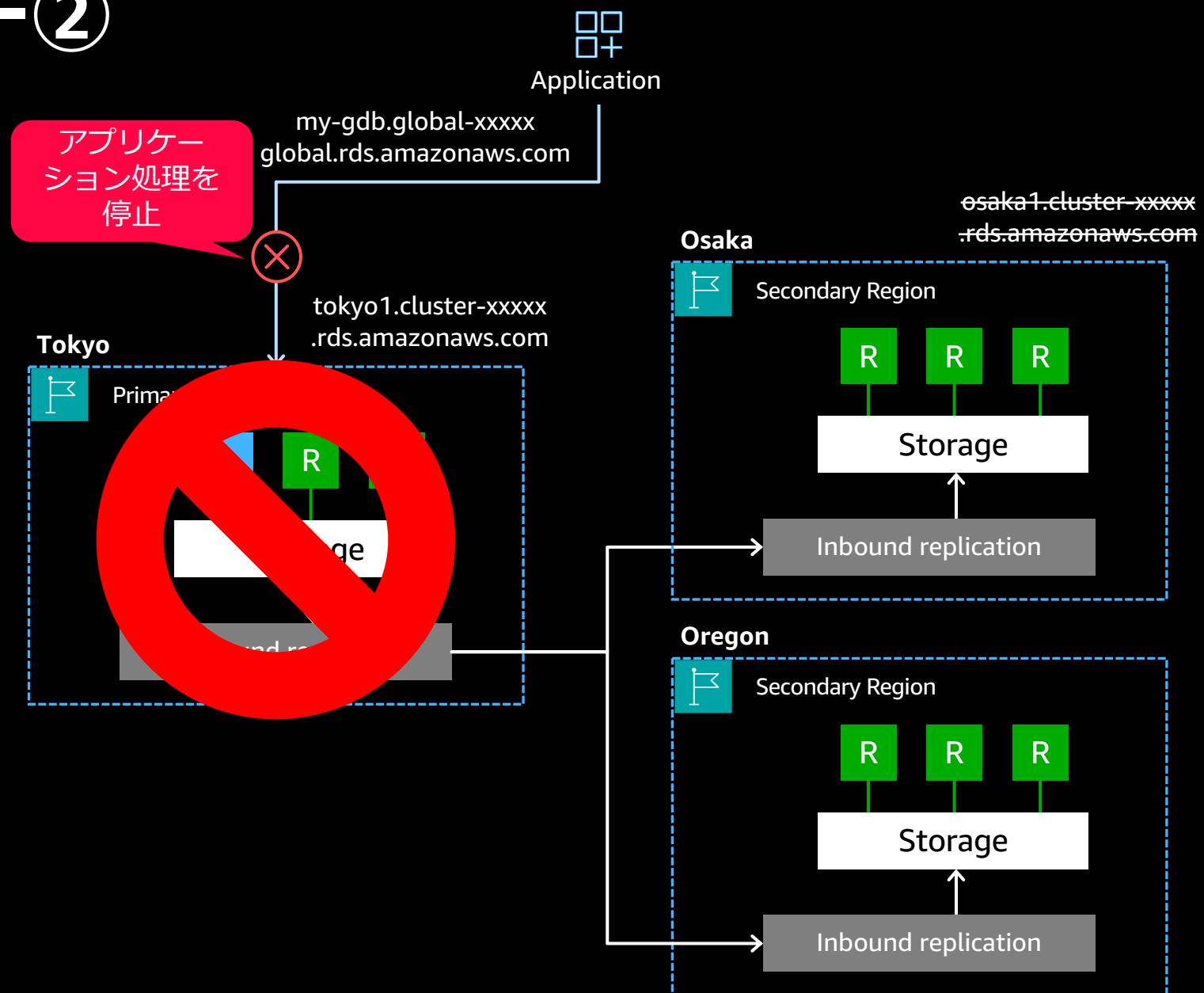


フェイルオーバー②

マニュアルフェイルオーバー

セカンダリクラスターを手動で元の Global Database から切り離し、新たな Global Database として再構成する。

- スプリットブレイン発生を避けるために、まずアプリケーションの書き込み処理は停止する



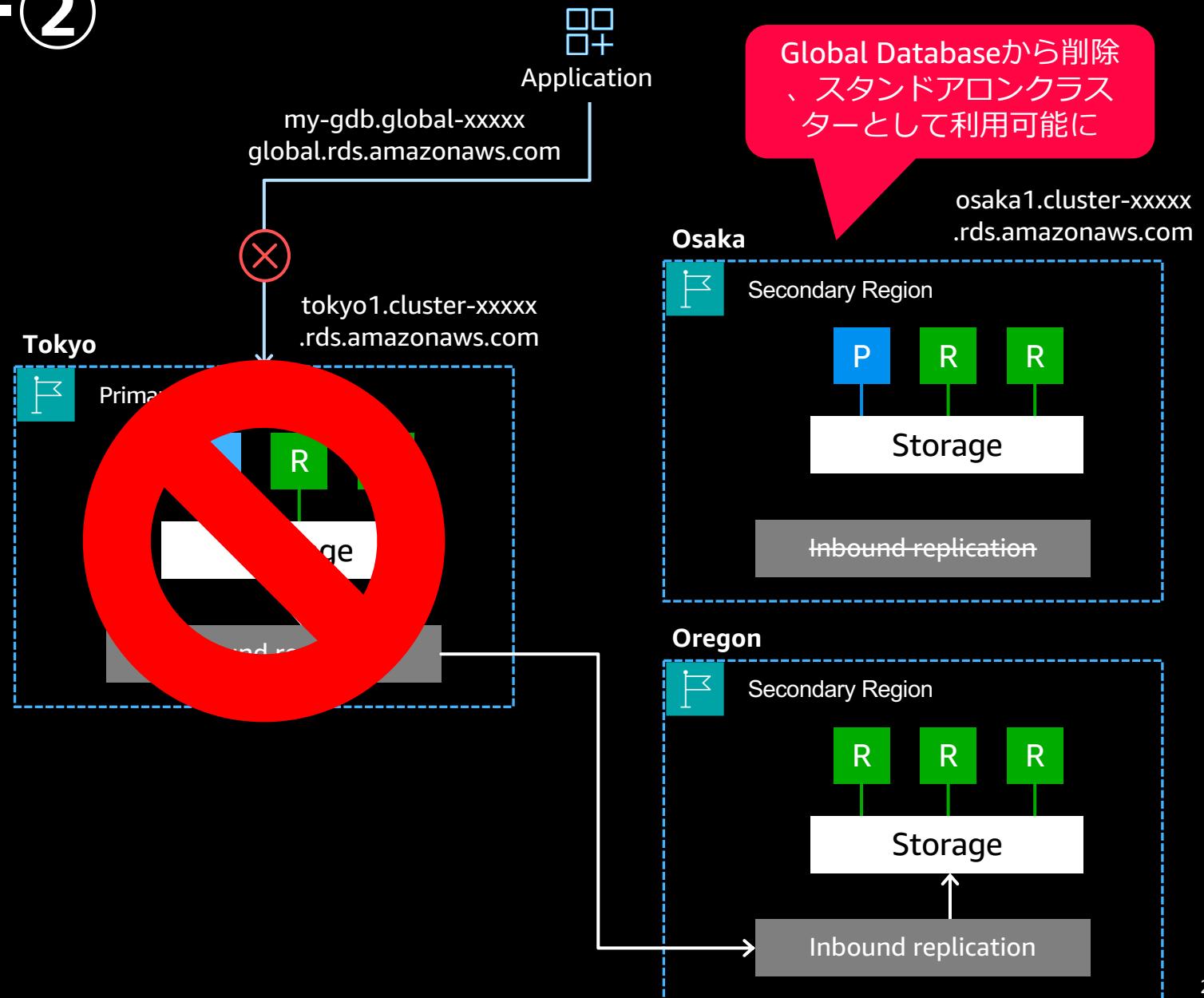
フェイルオーバー②

マニュアルフェイルオーバー

セカンダリクラスターを手動で元の Global Database から切り離し、新たな Global Database として再構成する。

- フェイルオーバー先セカンダリクラスターを Global Database から削除し、スタンドアロンクラスターにする

※ Global Database 構成とはなっておらずリージョン間の冗長性は無し

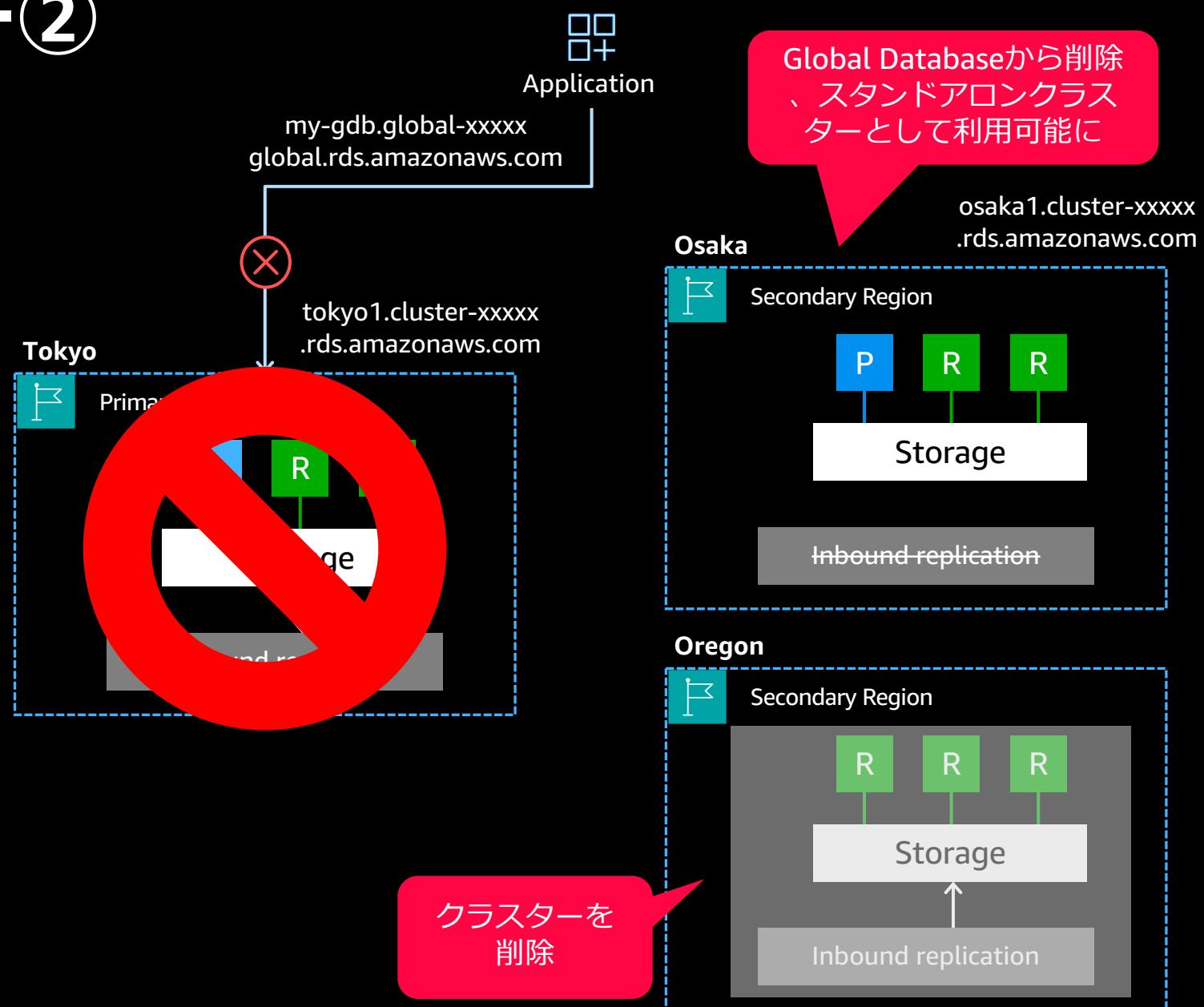


フェイルオーバー②

マニュアルフェイルオーバー

セカンダリクラスターを手動で元の Global Database から切り離し、新たな Global Database として再構成する。

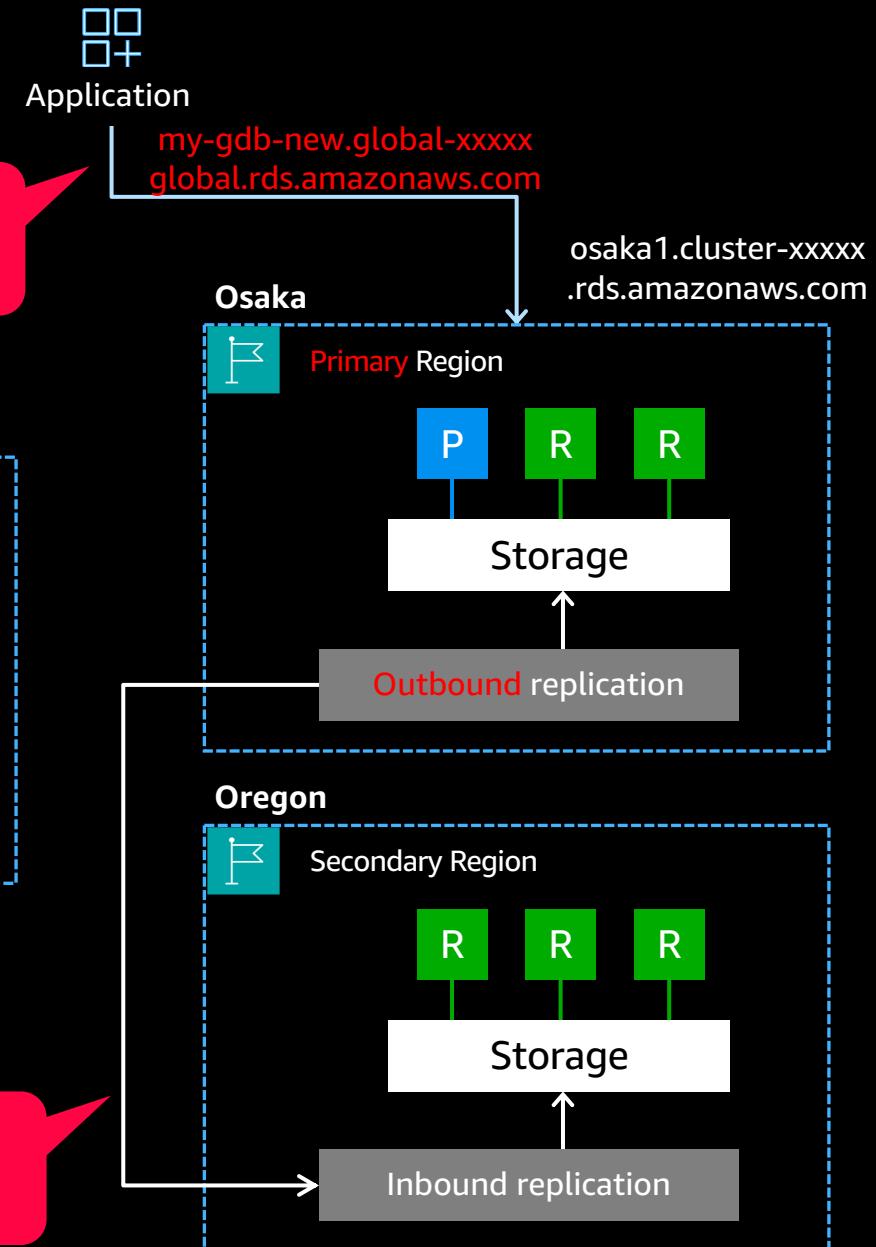
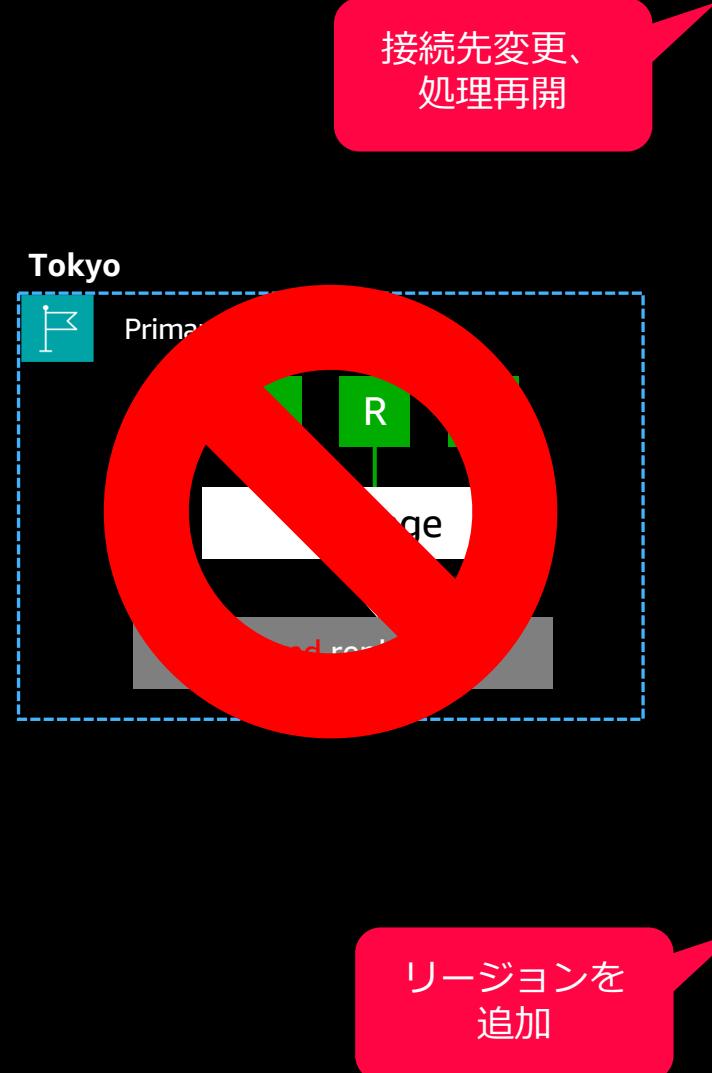
- フェイルオーバー先セカンダリクラスターを Global Database から削除し、スタンドアロンクラスターにする
※ Global Database 構成とはなっておらずリージョン間の冗長性は無し
- 他のセカンダリクラスターを削除する



フェイルオーバー②

マニュアルフェイルオーバー

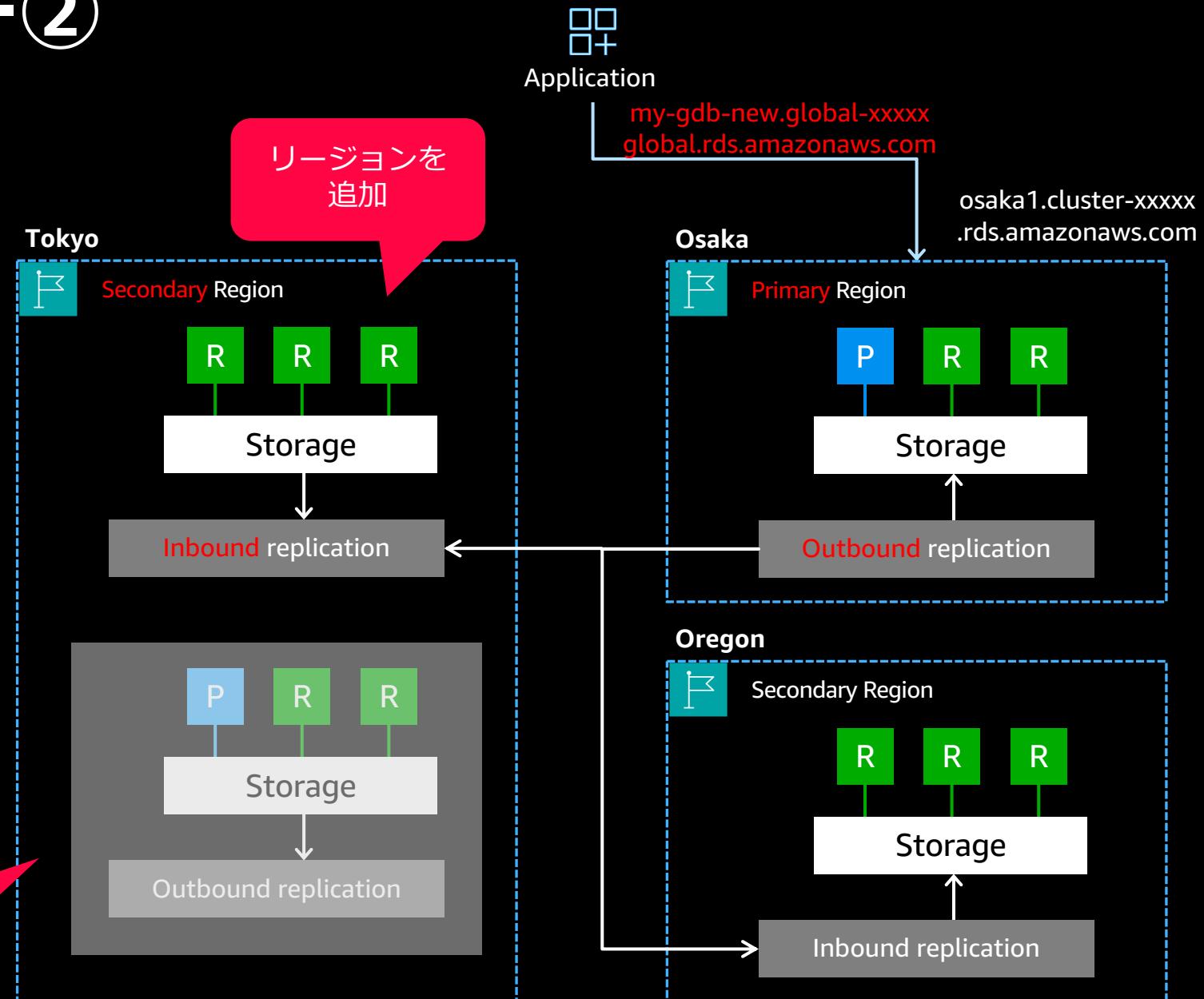
- ・ フェイルオーバー先ストアンドアロンクラスターに、正常なセカンダリリージョンを追加する
- ・ アプリケーション書き込みは、新しいグローバルライターエンドポイントか、プライマリリージョンのライターエンドポイントに接続するように接続情報を変更する



フェイルオーバー②

マニュアルフェイルオーバー

- 旧プライマリリージョンが復旧後、新 Global Database にリージョンを追加する
- 不要になった旧クラスターは削除する



クロスリージョンクラスター切り替えの注意事項

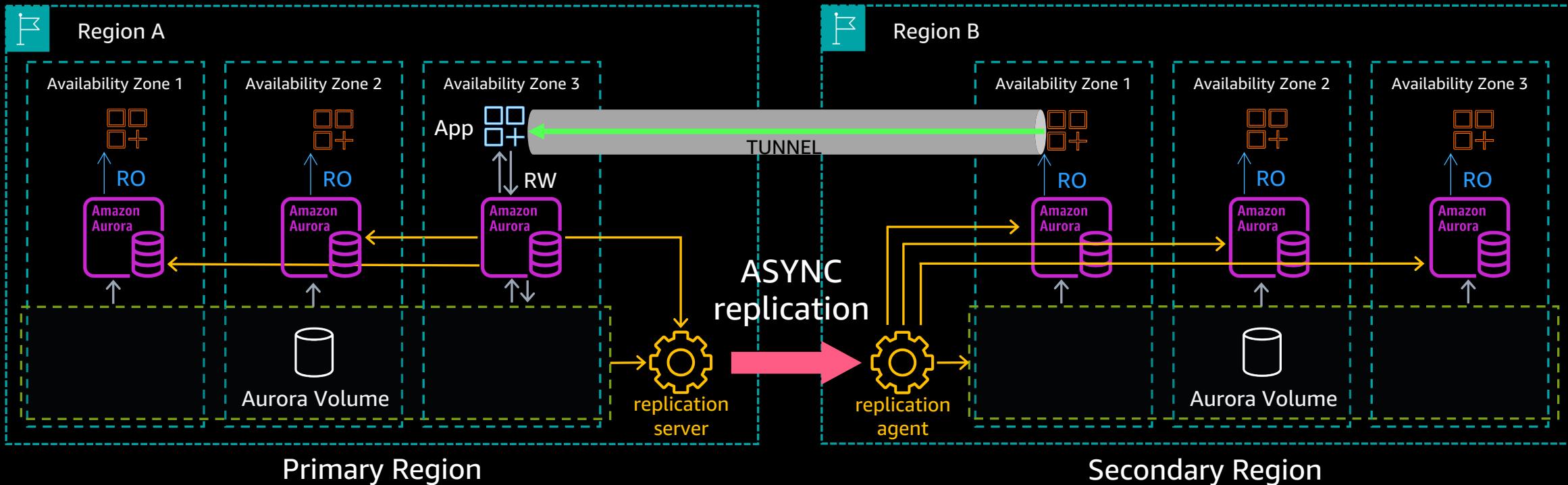
- 切り替え前にアプリケーションから書き込みが送信されないようにする
 - 切り替えプロセスの開始により Aurora はプライマリでの書き込み停止を試みるが、特にマネージドフェイルオーバーではベストエフォートの試行であり一時的に元のプライマリで書き込みが受け入れられスプリットブレインとなる可能性がある
- Global Databaseを構成するクラスターで全てのクラスターのバージョンをパッチレベルまで同一にできるだけ揃える
 - スイッチオーバー / マネージドフェイルオーバーを実行するにあたり、パッチレベルまで一致していることが必要なバージョンがある
 - > [Patch level compatibility for managed cross-Region switchovers and failovers](#)
 - マニュアルフェイルオーバーは、パッチレベルが同一でなくても実行可能
 - セカンダリクラスターを後から追加するといったケースで、より新しいパッチバージョンでクラスターが作成される場合がある

クロスリージョンクラスター切り替えの注意事項

- グローバルレライターエンドポイントを使用している場合、新しいプライマリに速やかに接続できるようにするために、クライアントの DNS キャッシュの存続時間 (TTL) を低い値にする
- 切り替えにおいて旧プライマリの設定オプションは継承されないため、構成の不一致による問題の回避を事前に精査しておく
 - グローバルデータベース内のすべてのクラスター間で、Aurora DB クラスター パラメータの設定を合わせる
 - セカンダリクラスターをプライマリクラスターに昇格した後に予期しない動作の変更が生じることを回避するため
 - Aurora Global Database 内の各 Aurora クラスターで、Aurora DB クラスター パラメータ グループを個別に設定することは可能
 - モニタリングツールの影響を精査しておく
 - レプリケーション ラグなどの一部の CloudWatch メトリクスは、セカンダリリージョンのみ使用可能であり、メトリクスの表示方法やアラームの設定方法が変わり、定義済みのダッシュボードを変更する必要が生じる場合がある
 - Amazon S3 や AWS Lambda など他の AWS サービスと連携している場合、セカンダリリージョンからのアクセスできるようになっていることを確認しておく

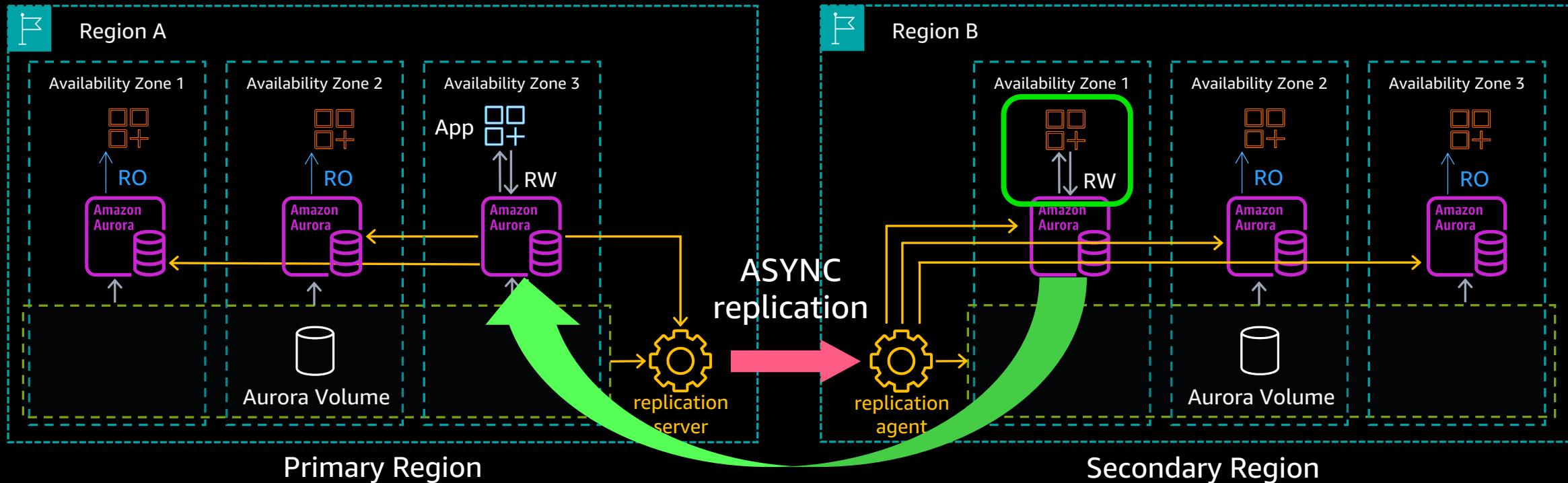
Amazon Aurora Global Database その他の主な機能や制限事項

Aurora Global Database 書き込み転送



- セカンダリリージョンでの Aurora レプリカは読み取り専用
- セカンダリリージョンのアプリケーションからの書き込みには、VPC 間を接続しライターエンドポイントへの接続が必要

Aurora Global Database 書き込み転送



- 書き込み転送機能により、セカンダリリージョンのリーダーインスタンスへの書き込みは、プライマリリージョンのライターインスタンスに転送される
- 書き込み処理の独自のメカニズムの実装や、クロスリージョンネットワーキングの設定が不要に

Aurora Global Database 書き込み転送

- セカンダリクラスターの読み取り整合性

- EVENTUAL

- > 同一セッションでコミットされた書き込みであっても、転送後にレプリケーションされる前は後続のクエリでは書き込み前のデータが取得される

- SESSION

- > 同一セッションでコミットされた書き込みの結果が、そのセッションの後続のクエリで取得される
 - > 転送された書き込みがレプリケーションされるまで後続のクエリは待機する

- GLOBAL

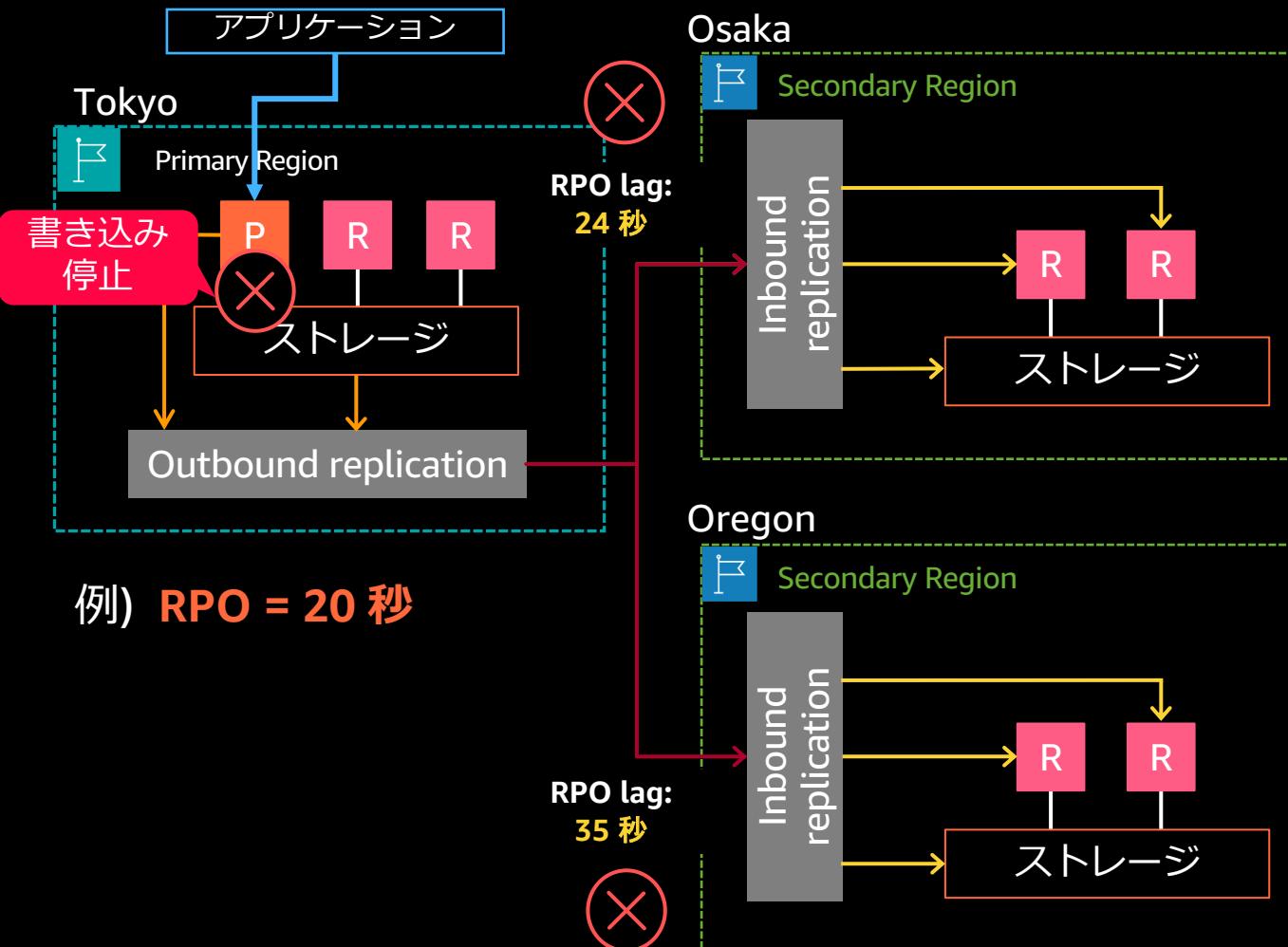
- > 他のリージョン含めた全セッションでコミットされた結果がを取得
 - > 各クエリは、クエリ開始時点のプライマリクラスターでコミットされた全てのデータがレプリケーションされるまで待機

- 書き込み転送の利用にあたっては、各データベースエンジン毎の使用可能な SQL ステートメントに注意



Aurora Global Database RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



機能:

- PostgreSQL ベースの Aurora Global Database
- 最大 RPO (目標復旧ポイント) を定義可能

動作:

- 設定した RPO 上限を全セカンダリリージョンが超過した場合、いずれかが追いつくまでプライマリクラスターの書き込みを一時停止

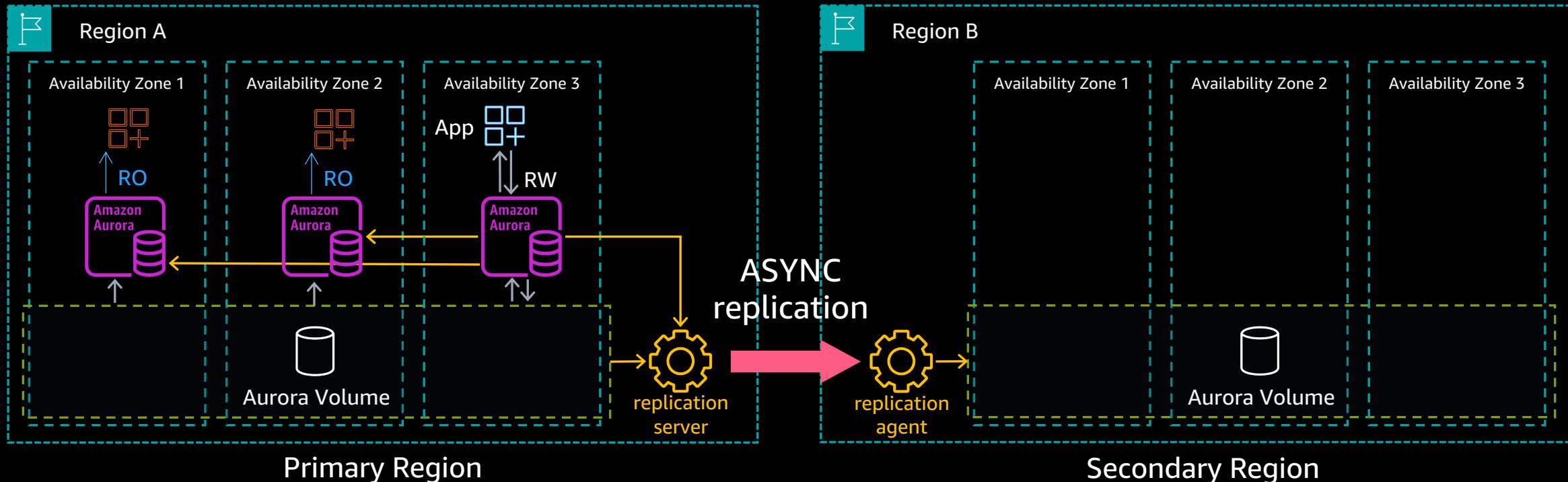
設定方法:

- rds.global_db_rpo パラメータ
- 20 秒 ~ 68 年の範囲で設定

監視メトリクス:

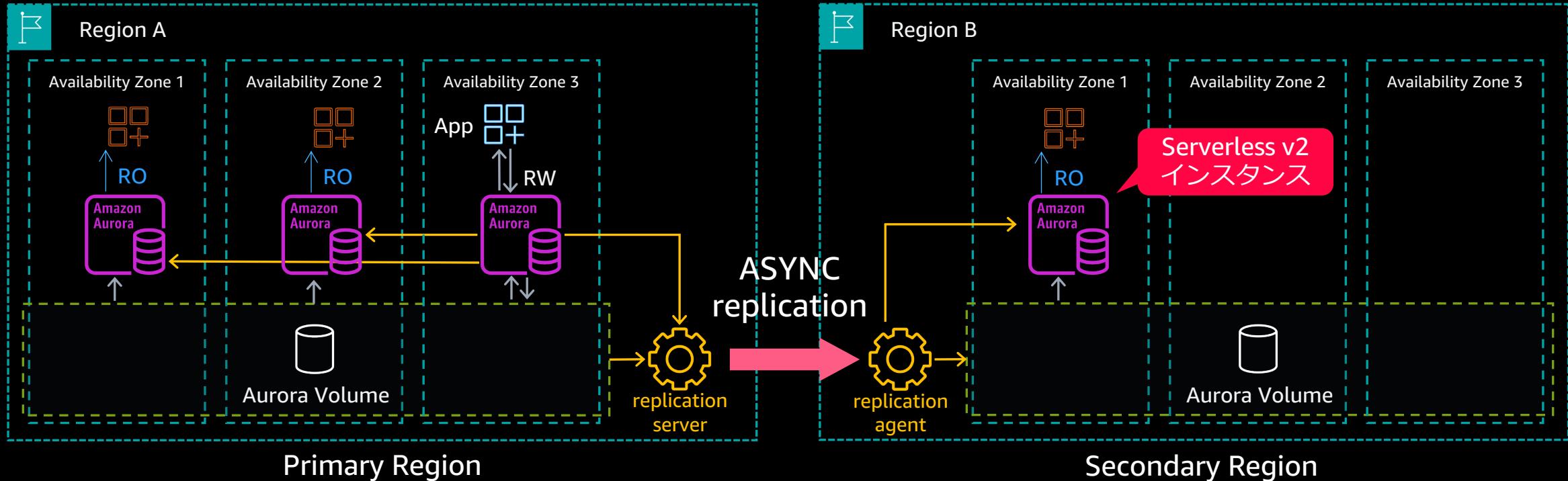
- AuroraGlobalDBRPOLag:
セカンダリクラスターの RPO ラグ時間 (秒)

セカンダリリージョンヘッドレスクラスター



- セカンダリリージョンで DB アクセスが不要な場合に、インスタンス費用を削減する構成
- セカンダリリージョン構成後に、リーダーインスタンスを削除することでも構成可能
- スイッチオーバー/フェイルオーバー前にインスタンスの追加が必要であり、RTO が長くなる
- インスタンス追加は RDS コンソールからは行えず、CLI/API でのオペレーションが必要

セカンダリーリージョン Serverless 構成



- 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
※ 最小容量を 0 ACU にしても、インスタンスは一時停止しない
- 従来のヘッドレス構成よりオペレーション自動化やサービス再開が容易に行える

Aurora Global Database 制限事項

- ・ 機能改善は特定バージョン以上のみ利用可能である場合がある
(例)
 - 計画外のイベント発生時にセカンダリリージョンのリードレプリカがサービスの継続性を維持できるようにする可用性の改善
 - > Aurora PostgreSQL 16.6, 15.10, 14.15, 13.18, 12.22 以上、Aurora MySQL 3.09 以上
 - クロスリージョンスイッチオーバーの時間を通常 30 秒未満に短縮する改善
 - > Aurora PostgreSQL 16.8, 15.12, 14.17, 13.20 以上、Aurora MySQL 3.09 以上
- ・ Blue/Green デプロイメントはサポートされない
 - 一時的にスタンダードアロンクラスターとすることで Blue/Green デプロイメントを利用することは可能 ※ Global Database 再構築になる
- ・ セカンダリ DB クラスターでは Aurora Auto Scaling 利用不可
- ・ クラスターを個別に停止や起動は不可

Aurora Global Database 制限事項

- プライマリクラスターのライターインスタンスが再起動やフェイルオーバーすると、セカンダリリージョンのリードレプリカが再起動する場合がある
- Secrets Manager 統合はサポートされない(セカンダリリージョン追加前に無効化する必要がある)
- Aurora バックトラックは利用不可
- Aurora PostgreSQL 固有の制限
 - RPO 管理機能を有効にしている場合、メジャーバージョンアップグレードを実行できない
 - セカンダリ DB クラスターでクラスターキャッシュ管理はサポートされない
 - プライマリ DB クラスターが Amazon RDS PostgreSQL インスタンスのレプリカをベースとしている場合、セカンダリクラスターの作成不可

詳細はドキュメント参照：[Limitations of Amazon Aurora Global Database](#)



まとめ

1. Amazon Aurora Global Database 概要
2. クロスリージョンクラスター切り替え
3. Amazon Aurora Global Database その他の主な機能と制限事項

Amazon Aurora Global Database

- リージョンを跨るデータベースの地理的冗長性やデータの近接配置を容易に構成可能
- ストレージレイヤーでのレプリケーションによる、インスタンスに影響を与えずに高スループット低遅延のリージョン間同期
- 計画したデータ損失 0 のスイッチオーバー、予期せぬ障害での高速なフェイルオーバー

クロスリージョンでの可用性やデータの地理的近接配置をお考えの際には、
まずは Aurora Global Database をご検討ください

※ Aurora Global Database 固有の制限に注意



Thank you!



AWS Black Belt Online Seminar

Amazon Aurora : 可用性

Yukki (高橋 敏行)

Partner Solutions Architect
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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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)

> Who am I ?

Yukki (ゆつきー) 別名: 高橋敏行

パートナー技術統括本部
テクニカルイネーブルメント部
パートナーソリューションアーキテクト

主にパートナー様の育成・技術支援を担当
前職はデータベースエンジニア

好きな AWS のサービス : **Amazon RDS and Amazon Aurora**



@YukkiTakahashi



アジェンダ

1. Amazon Aurora

- サービス概要

2. データベースの可用性

- 障害の種類
- RTO/RPO

3. Amazon Auroraで実現する高可用性

- クオーラムモデル
- マルチ AZ 構成
- Aurora Global Database

4. まとめ

AWS が提供する クラウドデータベース



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

Amazon RDS と Amazon Aurora

リレーショナルデータベース



Amazon RDS

- ✓ 6つの一般的なデータベースエンジンから選べるマネージド型リレーショナルデータベースサービス
- ✓ データベースエンジンのオリジナルに近い機能で提供

MySQL

PostgreSQL

MariaDB

SQL Server

Oracle Database

IBM Db2



Amazon Aurora

- ✓ オープンソースデータベースをクラウド向けに再設計したマネージド型リレーショナルデータベースサービス
- ✓ より高いパフォーマンスと可用性を提供

MySQL

PostgreSQL

DSQL

Amazon RDS と Amazon Aurora

リレーショナルデータベース



Amazon RDS

- ✓ 6つの一般的なデータベースエンジンから選べるマネージド型リレーショナルデータベースサービス
- ✓ データベースエンジンのオリジナルに近い機能で提供

MySQL

PostgreSQL

MariaDB

SQL Server

Oracle Database

IBM Db2



Amazon Aurora

- ✓ オープンソースデータベースをクラウド向けに再設計したマネージド型リレーショナルデータベースサービス
- ✓ より高いパフォーマンスと可用性を提供

MySQL

PostgreSQL

DSQL

Amazon Aurora

Enterprise database at open source price

AWS が提供するマネージドサービス



MySQL と PostgreSQL との互換性

オープンソースデータベースのシンプルさと費用対効果

商用データベースのスループットと可用性

シンプルな従量課金制

※ Aurora DSQL は別の機会でご説明します

データベース管理のフルマネージ化

オンプレミス

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

セルフマネージ

仮想サーバー

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

Amazon EC2

データベースサービス

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

Amazon Aurora 等

AWS マネージ



データベース管理のフルマネージ化

オンプレミス

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

セレクマネージ

仮想サーバー

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

データベースサービス

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

Amazon Aurora 等

Amazon EC2



Amazon Aurora

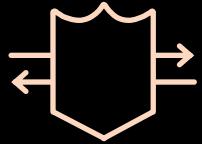
商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



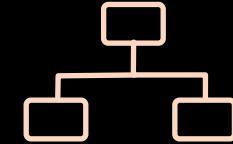
パフォーマンスと
拡張性



可用性と耐久性



高いセキュリティ



フル・マネージド

Amazon Aurora

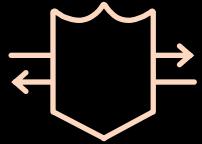
商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



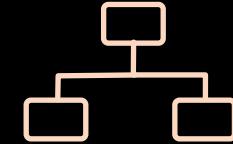
パフォーマンスと
拡張性



可用性と耐久性



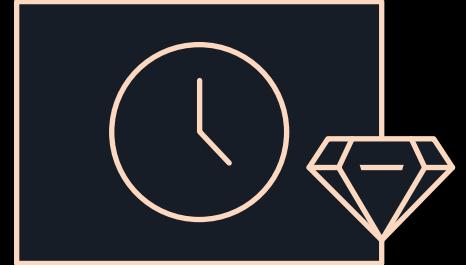
高いセキュリティ



フル・マネージド

Amazon Aurora

商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



可用性と耐久性

- マルチ AZ 構成を用いて 99.99% の可用性
- データは 3 AZ をまたがって保存 (課金は 1 コピー分のみ)
- 自動的な継続的な増分バックアップの取得
- ポイントインタイムリカバリ (PITR) 可能
- 通常 10 秒以下のフェイルオーバー
- フォルト・トレラントで自己修復ができ、自動拡張可能なストレージ
- ディザスタリカバリのための Global Database 機能

データベースの可用性



"Everything fails, all the time."

全てのものはいつでも壊れうる

Werner Vogels

VP and CTO, Amazon.com



ed the Turing test

すべては壊れるものとして考える

“

障害は発生するものであり、最終的にはすべてが時間の経過とともにフェイルオーバーします。

つまり、ルーターからハードディスクまで、TCP パケットを破壊するオペレーティングシステムからメモリユニットまで、そして一時的なエラーから永続的な障害まで、どれもが対象となるのです。

これは、最高品質のハードウェアを使用しているか、最低料金のコンポーネントを使用しているかにかかわらず、当たり前のことです。

- Werner Vogels, CTO, Amazon.com -

すべては壊れるものとして考える

“

障害は発生するものであり、最終的にはすべてが時間の経過とともにフェイルオーバーします。

つまり、ルーターからハードディスクまで、TCP パケットを破壊するオペレーティングシステムからメモリユニットまで、そして一時的なエラーから永続的な障害まで、どれもが対象となるのです。

これは、最高品質のハードウェアを使用しているか、最低料金のコンポーネントを使用しているかにかかわらず、当たり前のことです。

- Werner Vogels, CTO, Amazon.com -

データベース障害の種類

ディスク障害:

データ破損:

データベースサーバー障害:

データセンター障害:

大規模障害:

データベース障害の種類

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

データベース障害の種類

ディスク障害: ストレージデバイスの物理的または論理的な故障

- **原因:** 物理的損傷、媒体劣化、コントローラー故障、ファームウェアバグ、過熱、電源サージ

データ破損: データの整合性が損なわれる問題

- **原因:** 電源障害、ソフトウェアバグ、メモリエラー、不適切なシャットダウン

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

- **原因:** ハードウェア故障、メモリ不足、OS 障害、DB エンジンクラッシュ

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

- **原因:** 電源障害、冷却システム故障、ネットワーク切断、局所的自然災害

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

- **原因:** 大規模自然災害、広域インフラ障害、複数 AZ に影響する重大問題

データベースの可用性：RTO / RPO

RTO: Recovery Time Objective

- リカバリ時間目標
- 障害時に、どれだけの時間でリカバリを完了させるか

RPO: Recovery Point Objective

- リカバリポイント目標
- 障害時に、どの時点のデータまでリカバリ可能か

データベースの可用性：RTO / RPO

RTO (Recovery Time Objective) - 目標復旧時間

- ・ **定義:** システム障害やサービス停止から復旧までに許容される最大時間
- ・ **意味:** ダウンタイムの許容範囲を示す指標
- ・ **重要性:** ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ **例:** RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ **定義:** 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ **意味:** 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ **重要性:** データの重要性と更新頻度に基づいて設定される指標
- ・ **例:** RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容

データベースの可用性：RTO / RPO

RTO (Recovery Time Objective) - 目標復旧時間

- ・ 定義: システム障害やサービス停止から復旧までに許容される最大時間
- ・ 意味: ダウンタイムの許容範囲を示す指標
- ・ 重要性: ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ 例: **RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要**

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ 定義: 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ 意味: 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ 重要性: データの重要性と更新頻度に基づいて設定される指標
- ・ 例: **RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容**

データベースの可用性：RTO / RPO

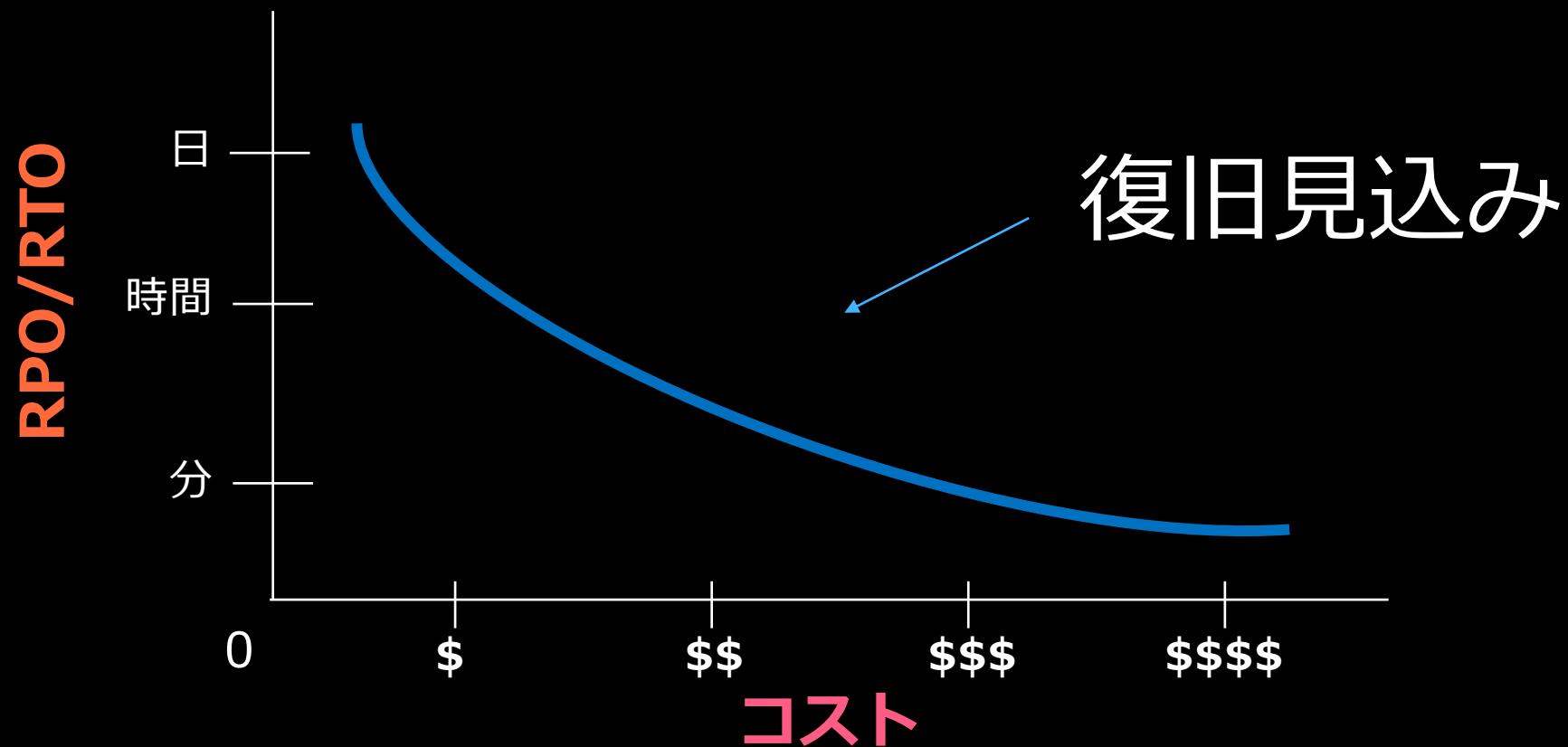
RTO (Recovery Time Objective) - 目標復旧時間

- ・ 定義: システム障害やサービス停止から復旧までに許容される最大時間
- ・ 意味: ダウンタイムの許容範囲を示す指標
- ・ 重要性: ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ 例: **RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要**

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ 定義: 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ 意味: 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ 重要性: データの重要性と更新頻度に基づいて設定される指標
- ・ 例: **RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容**

RPO/RTO の目標とコスト



必要な可用性を定義し、システム要件に応じて構成を選択する事が重要

Amazon Aurora で実現する 高可用性



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

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

ディスク障害:ストレージデバイスの物理的または論理的な故障

Amazon Aurora

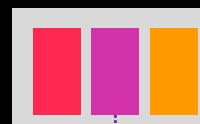
- ・ディスク障害が発生した場合、**Aurora ストレージ**の冗長化により継続稼働可能
- ・2つの破損までは稼働可能
- ・3つの破損が発生した場合は読み込みのみ可能

Aurora ストレージの高可用性

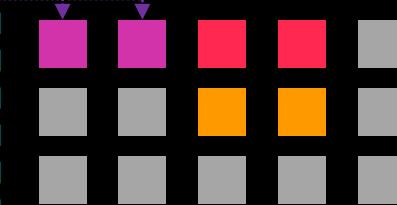
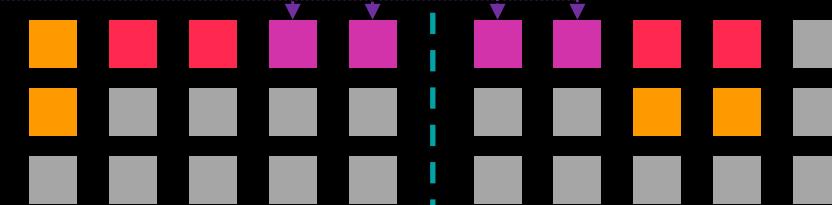
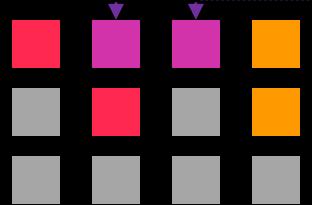
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

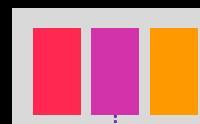
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html

Aurora ストレージの高可用性

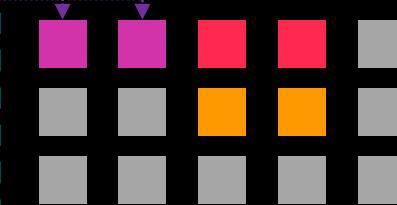
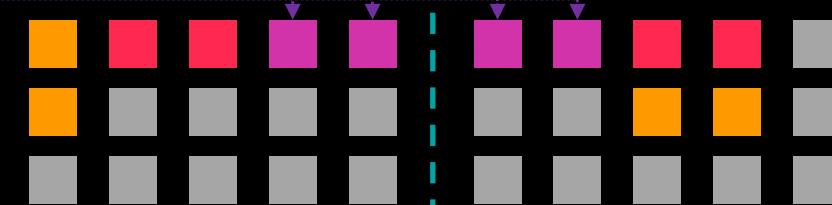
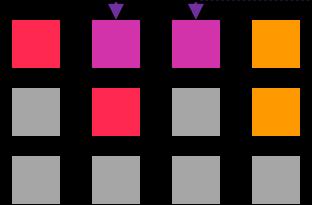
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

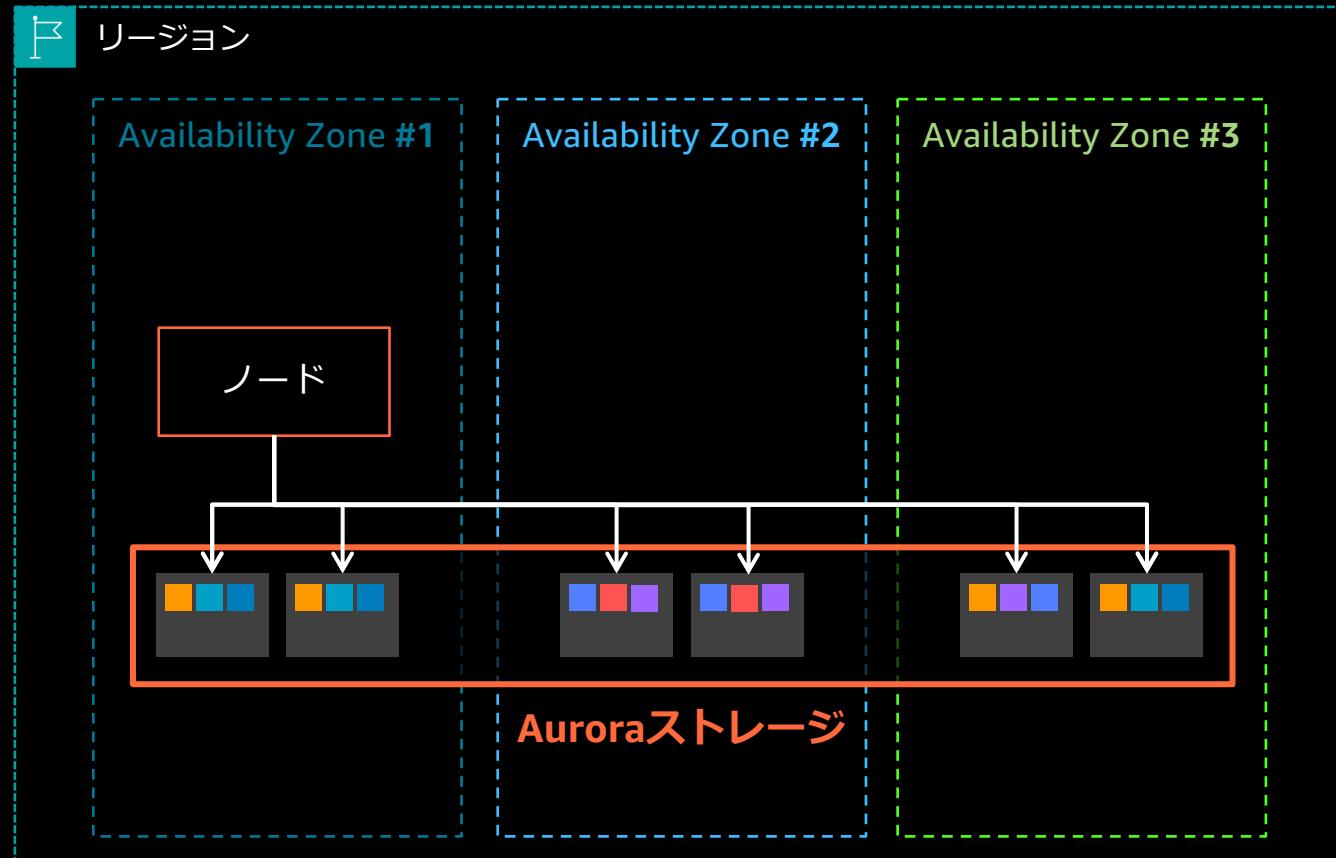
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html

ストレージ冗長化の違い (Amazon RDS / Amazon Aurora)

Amazon RDS



Amazon Aurora



Aurora ストレージの高可用性

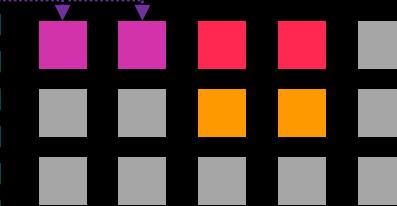
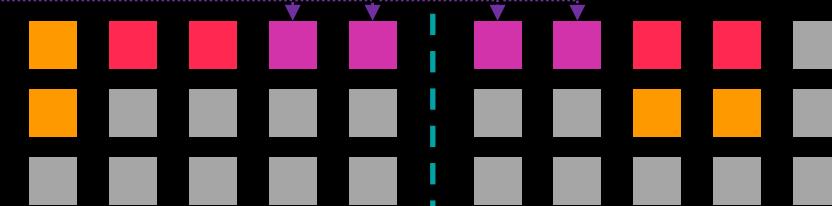
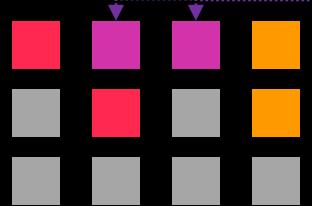
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

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

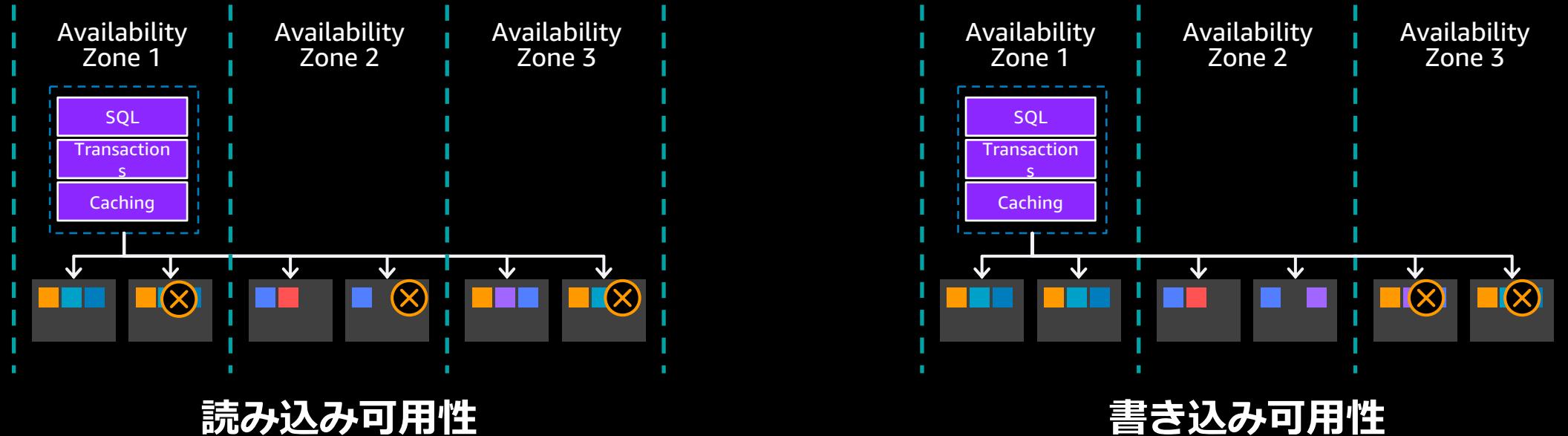
Aurora ストレージ：自動修復・耐障害性



データは 6 つのノードすべてに非同期・並列で書き込み
読み込みには 3/6 クオーラム、書き込みには 4/6 ノードのクオーラムが必要
各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

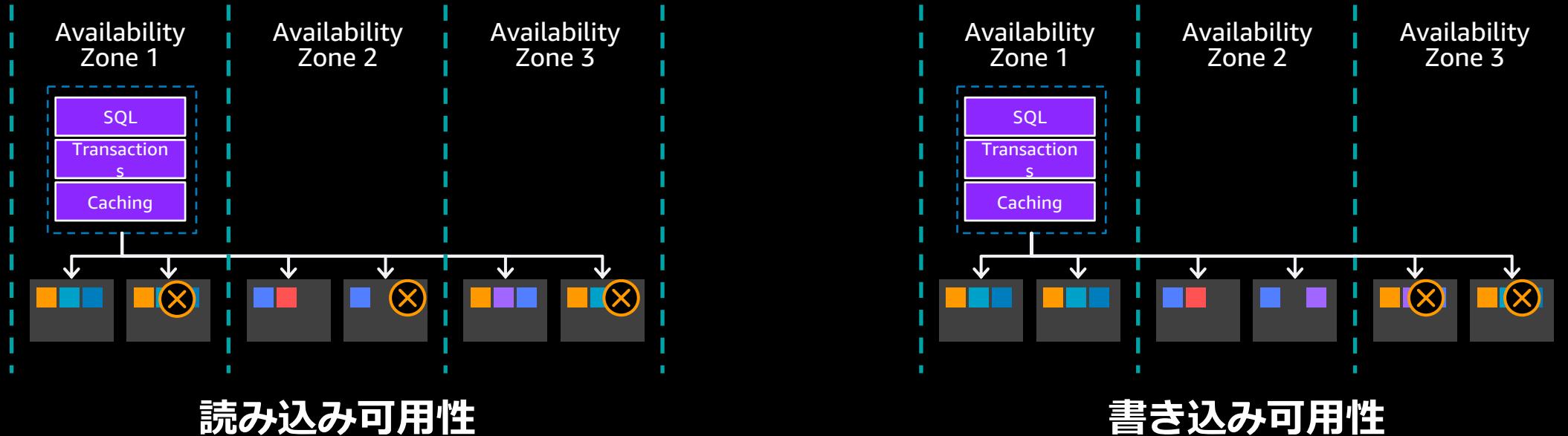
Aurora ストレージ：自動修復・耐障害性



データは 6 つのノードすべてに非同期・並列で書き込み
読み込みには 3/6 クオーラム、書き込みには 4/6 ノードのクオーラムが必要
各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

Aurora ストレージ：自動修復・耐障害性



データは 6 つのノードすべてに非同期・並列で書き込み

読み込みには **3/6 クオーラム**、書き込みには **4/6 ノードのクオーラム**が必要

各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

参考：クオーラムとは

- ・ 定義：集団における意思決定や**合意形成に必要な最小人数または最小比率**
- ・ 応用：分散コンピューティングシステムにおける合意形成アルゴリズム

分散システムでのクオーラムとは

- ・ 複数ノードで構成されるシステムにおいて、**有効な操作に必要な最小ノード数**
- ・ データの一貫性と可用性を確保するための合意形成メカニズム
- ・ 通常は全ノード数の過半数 ($N/2+1$) が基本的なクオーラム値として設定される

参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

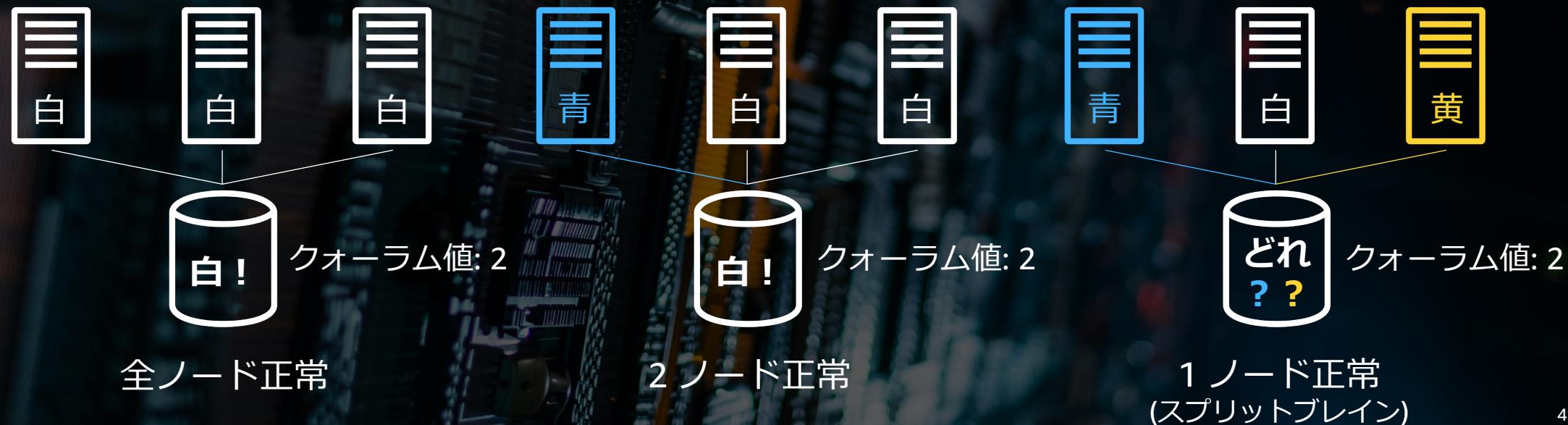
- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



クオーラム値: 3

全ノード正常



クオーラム値: 3

3ノード正常



クオーラム値: 3

1ノード正常
(スプリットブレイン)

クオーラムモデル

レプリケーション管理のためのクオーラムのルール

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

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

クオーラムモデル

レプリケーション管理のためのクオーラムのルール

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

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

Amazon Aurora の
クオーラムモデル

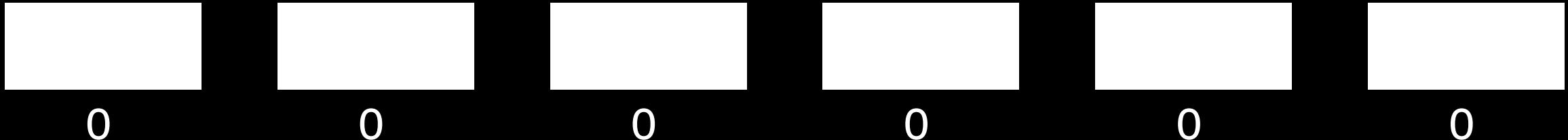
Aurora クオーラムモデル例

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$

コピー

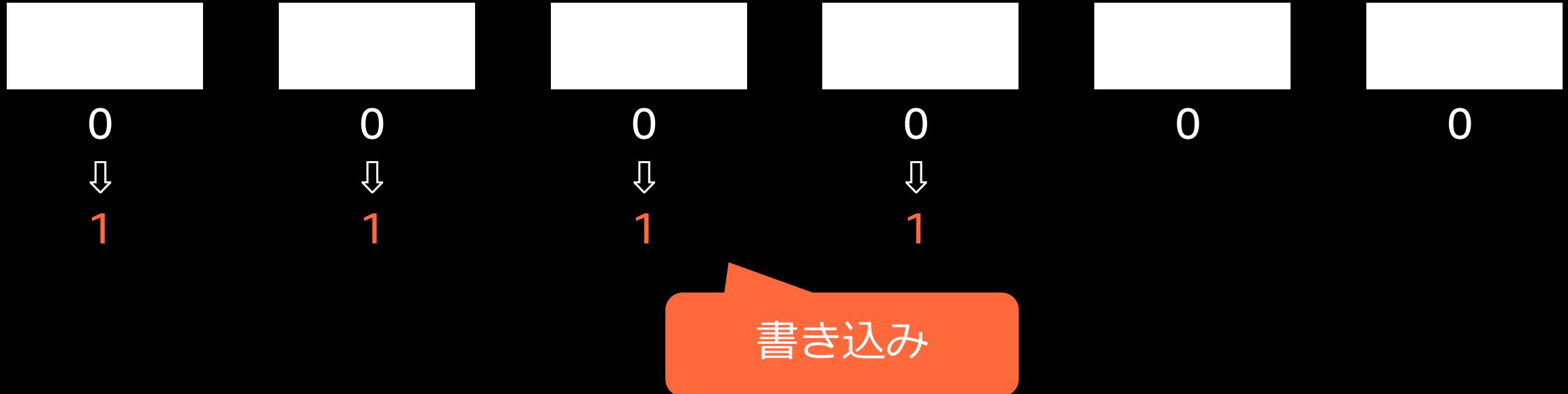
書き込み

読み込み



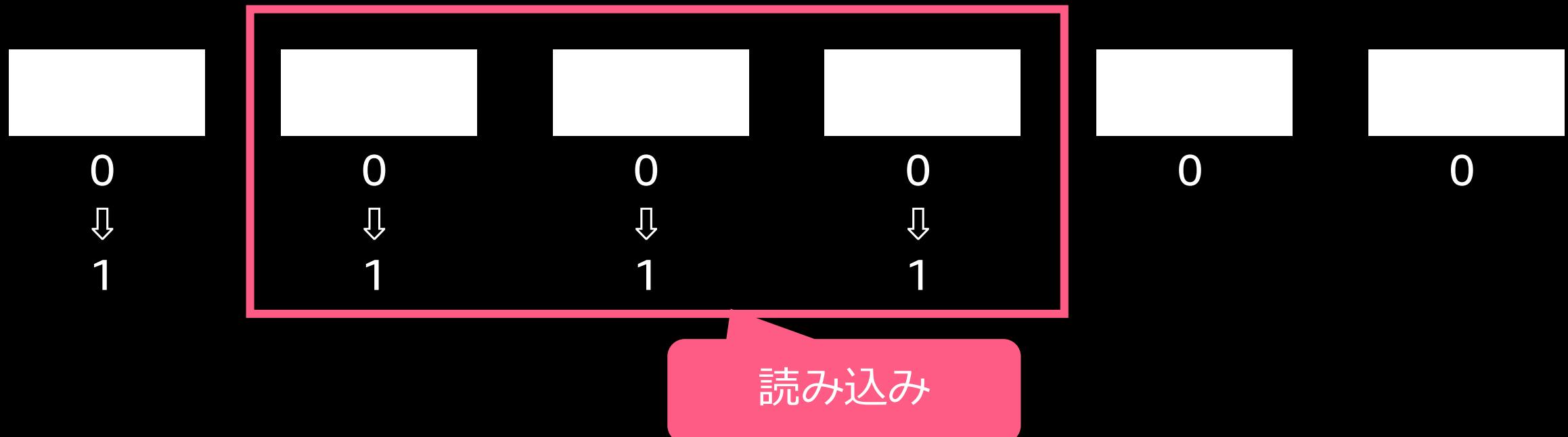
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



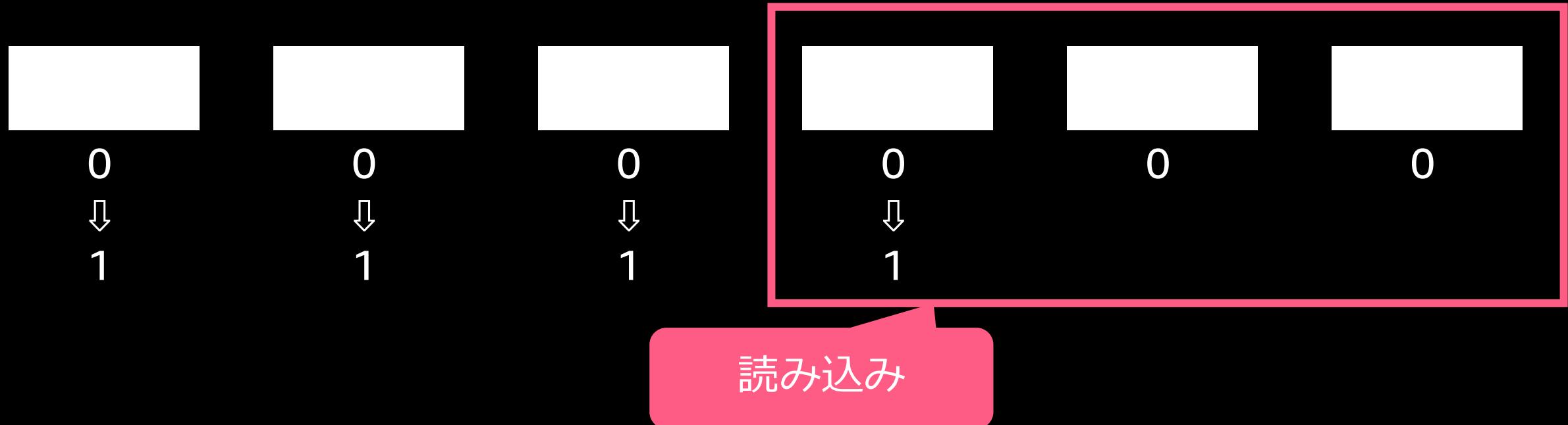
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



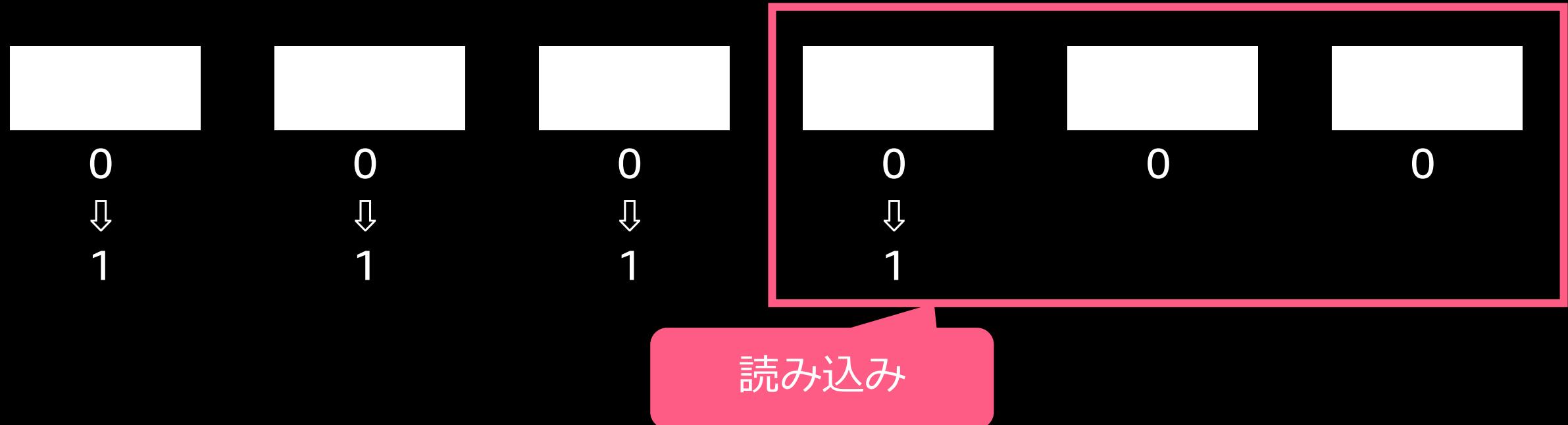
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$

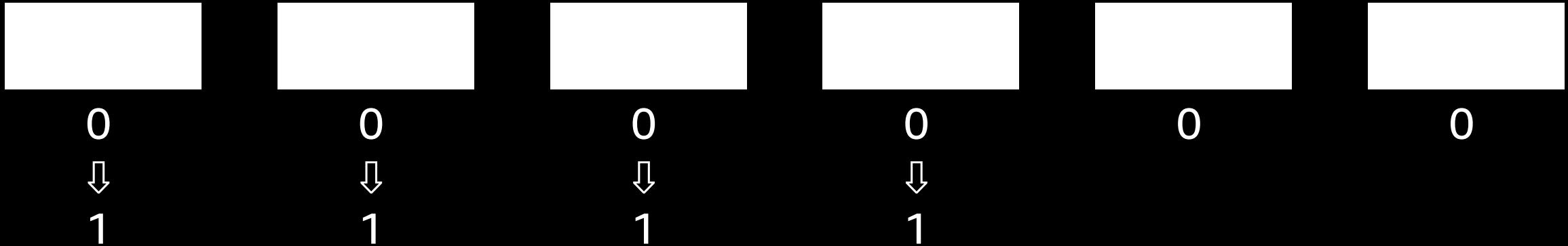


必ず最新のデータを読み取り可能

- $V_r + V_w > V$
- $V_w > V/2$

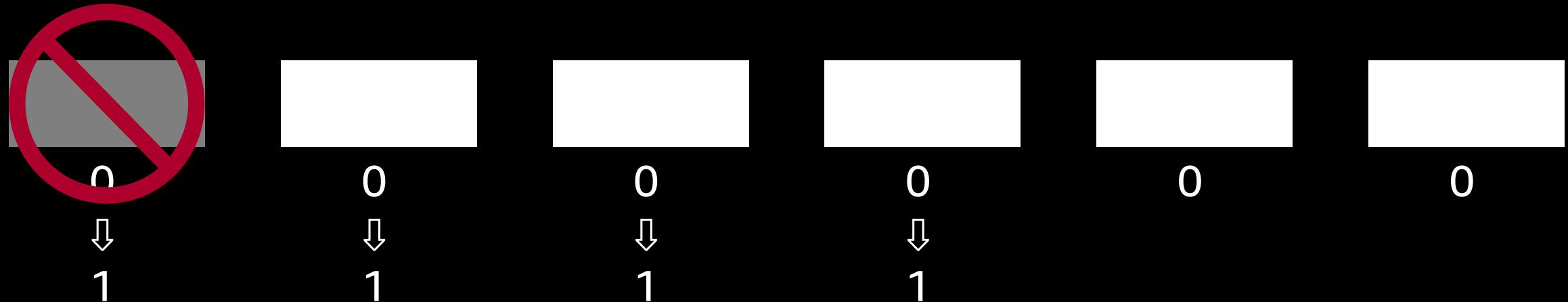
耐障害性 - クオーラムモデル

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



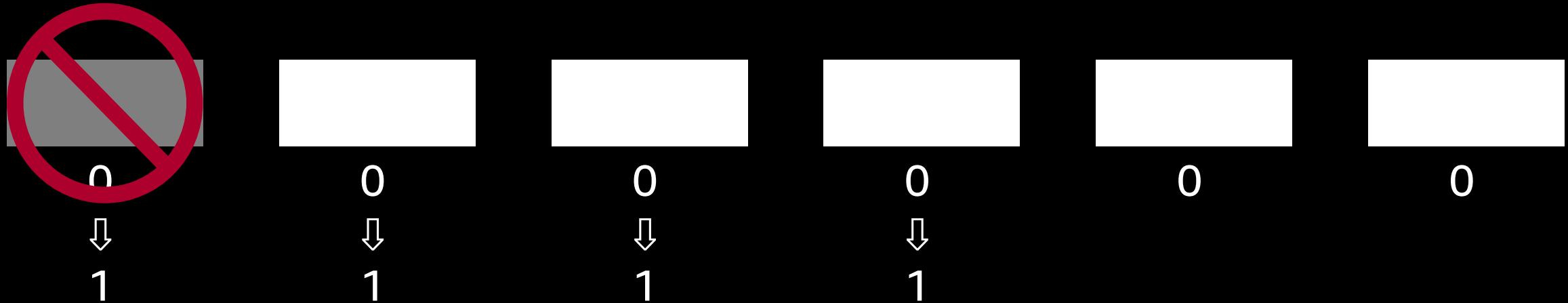
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



- $V_r + V_w > V$
- $V_w > V/2$

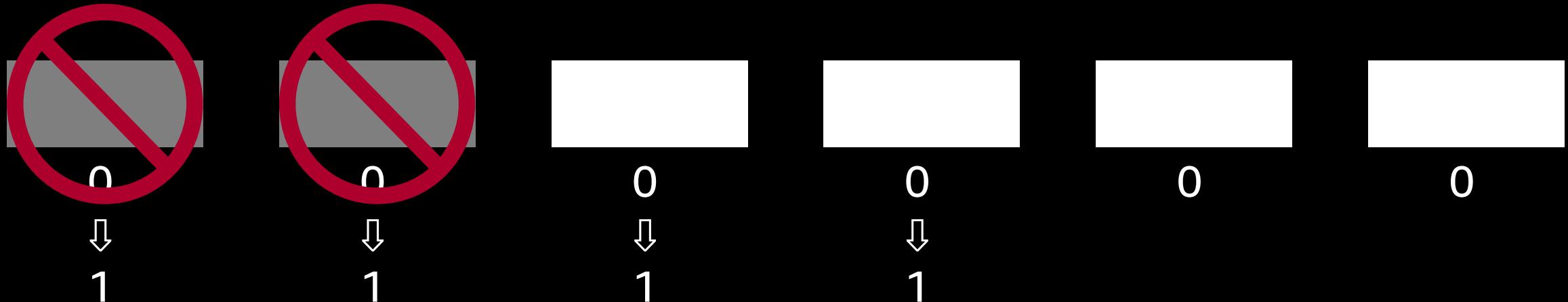
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



障害が起きても処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

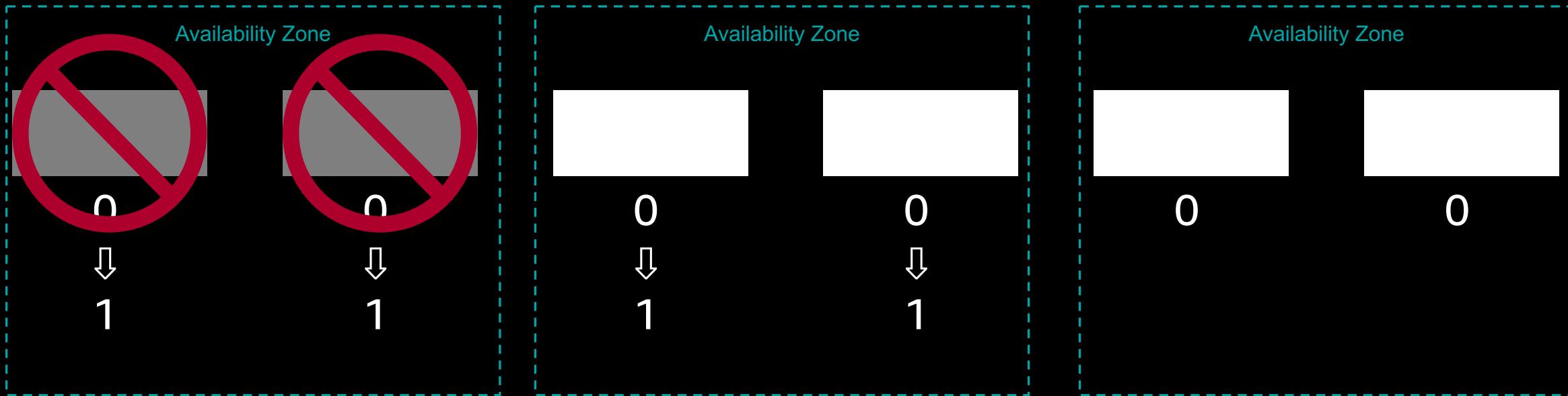
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



2重障害が起きても処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

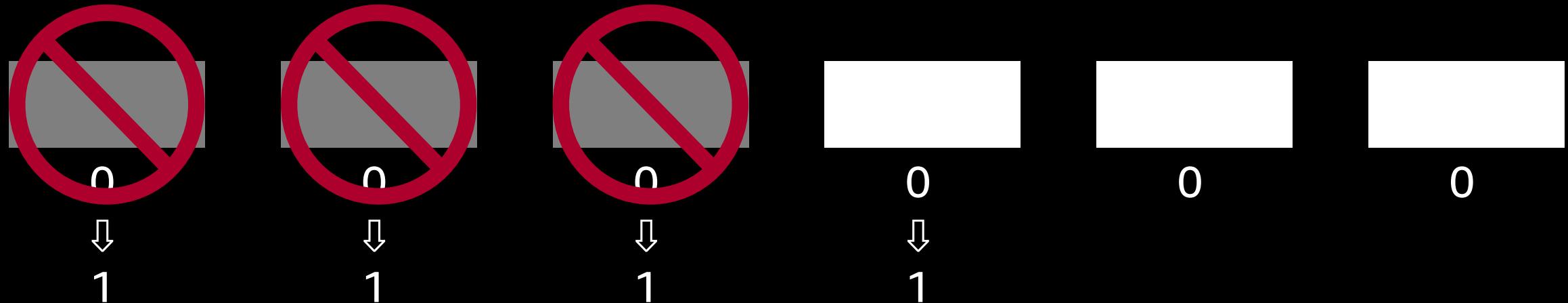
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



AZ 障害時も処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



3重障害が起きても読み込み可能

$$V_r=3$$

- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



AZ + 1 障害時も読み込み可能

- $V_r + V_w > V$
- $V_w > V/2$

Aurora では AZ 障害 + 1 の障害 を許容

1 AZ あたり 2つ、3 AZ に 6 つのコピー

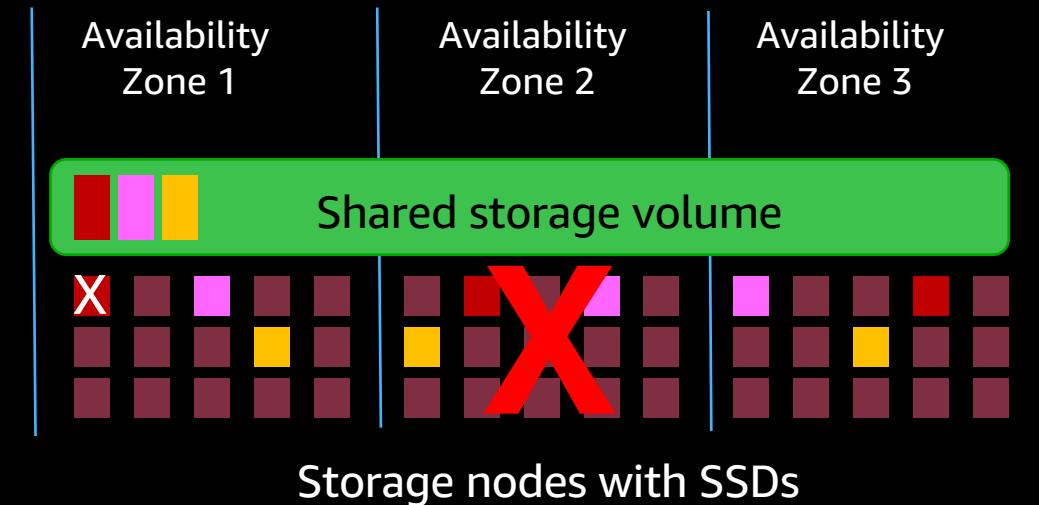
- 書き込みは 4/6 クオーラム

AZ 障害時

- 4/6 のコピーは維持
- 書き込みも可能

AZ 障害 + 1 の障害 (二重障害) 時

- 3/6 のコピーは維持
- データの損失は無
- 故障箇所は 3 つのコピーから再構築
- 再構築により書き込み可能に

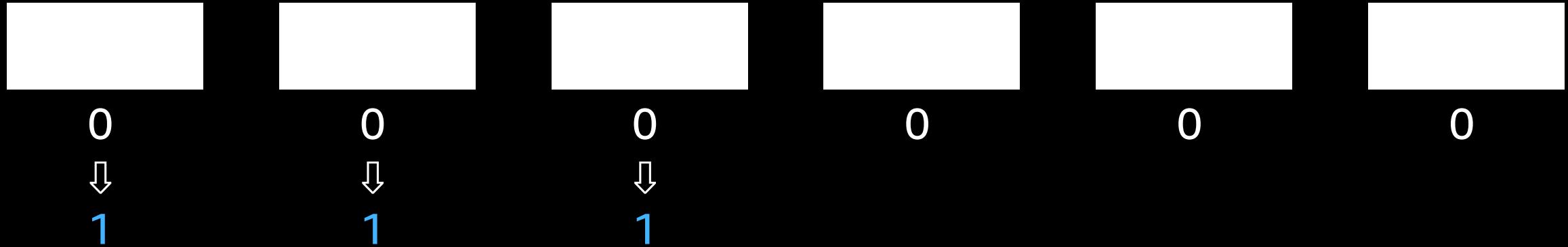


Question:

4/6 クオーラムも必要？

3/6 ($V_w=3$) でも良いのでは？？

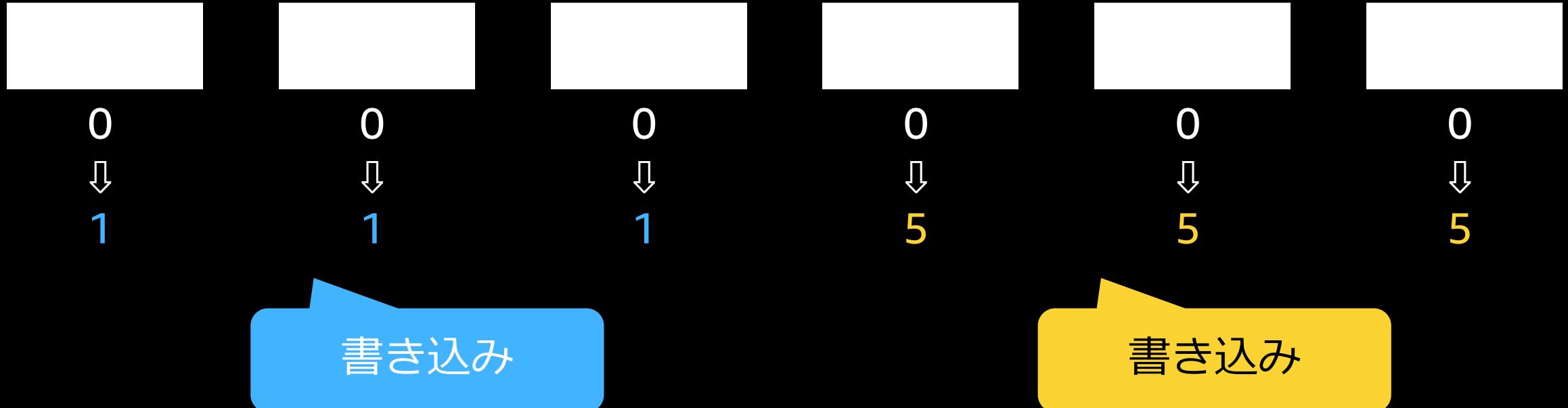
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



書き込み

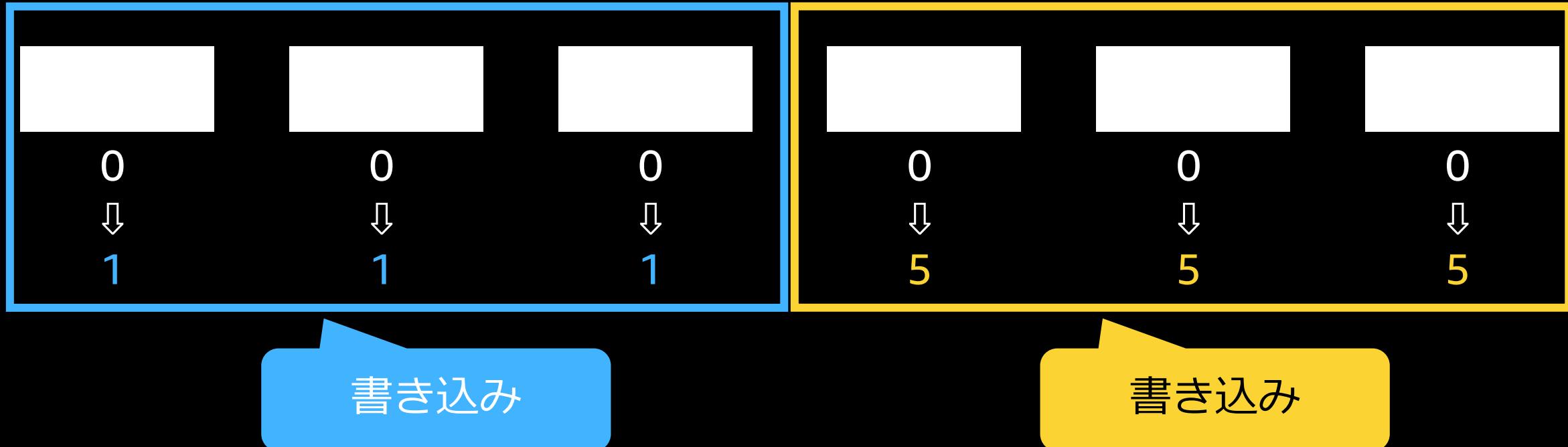
- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



- $V_r + V_w = V$
- $V_w = V/2$

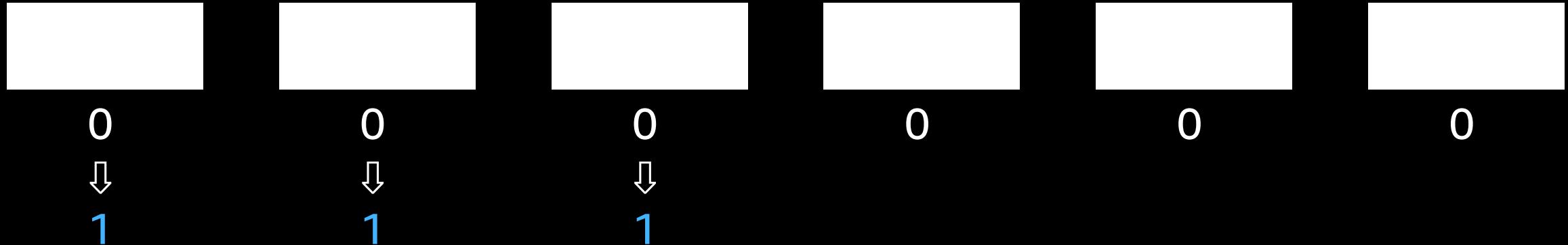
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



トランザクションが並列で書き込まれる可能性がある

- $V_r + V_w = V$
- $V_w = V/2$

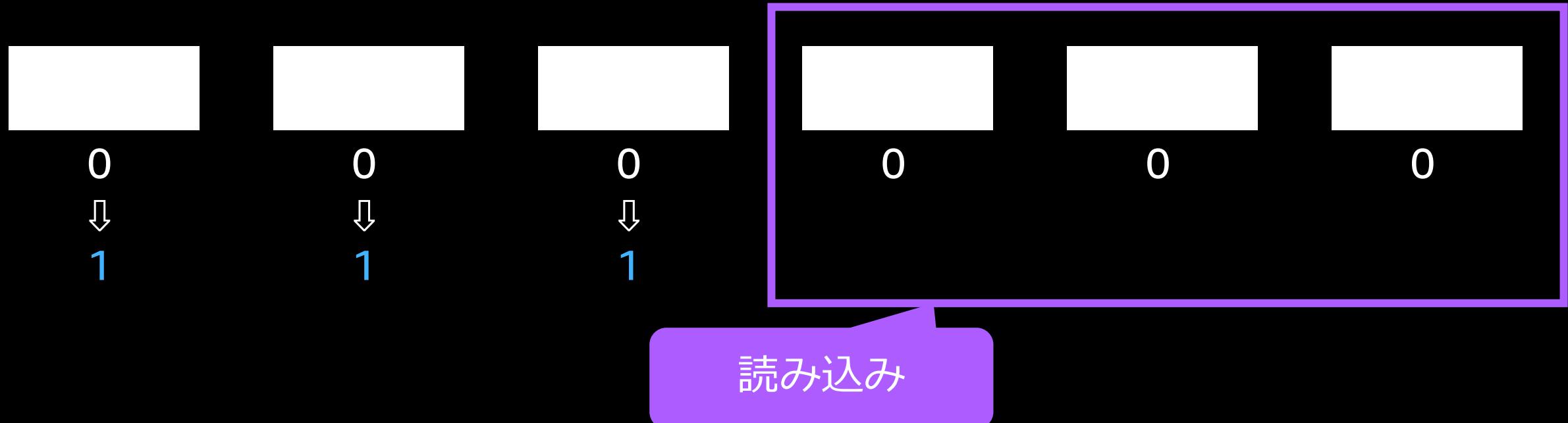
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



書き込み

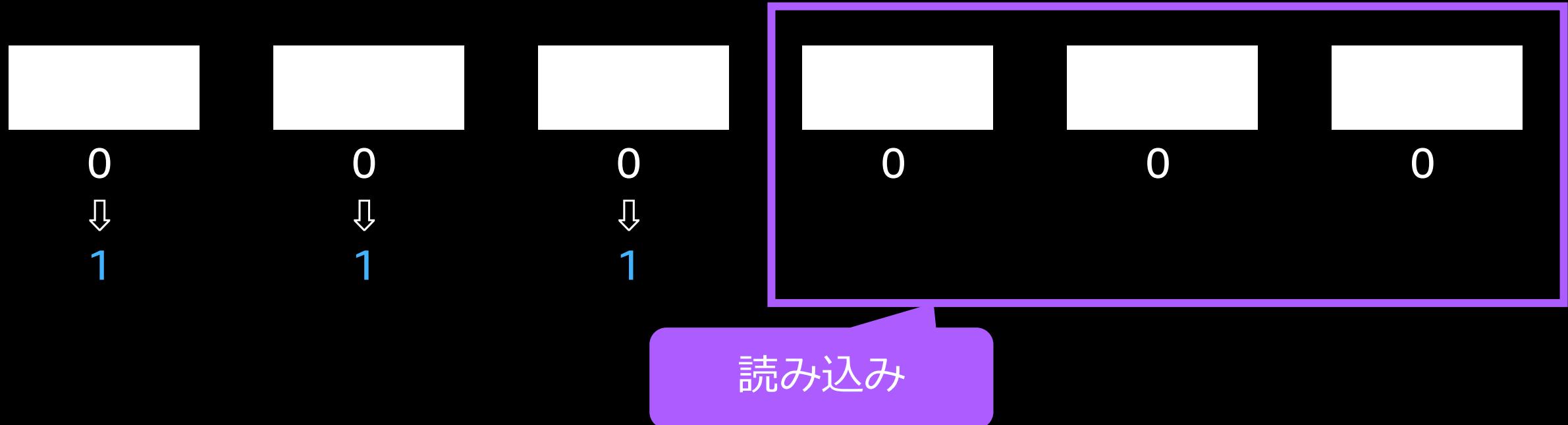
- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



必ず最新のデータを読み取れない可能性がある

- $V_r + V_w = V$
- $V_w = V/2$

再掲：クオーラムモデル

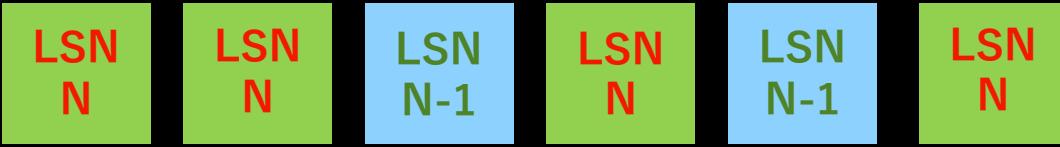
レプリケーション管理のためのクオーラムのルール

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

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

**Amazon Aurora の
クオーラムモデル**

参考：クオーラムにおける読み込みを回避



各データブロックに対して、
クオ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

ほとんどのクオーラムベースの
システムでは**読み込みにコスト**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

読み込みクオーラムは
修理やクラッシュリカバリに必要

LSN : Log Sequence Number
データベースの変更を追跡するための一意の識別子

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

データ破損: データの整合性が損なわれる問題

Amazon Aurora

- ユーザーエラーなどによりデータが論理的に破損した場合
バックアップからのリストア・リカバリにより対応可能
- ただしシステム停止が必要になる

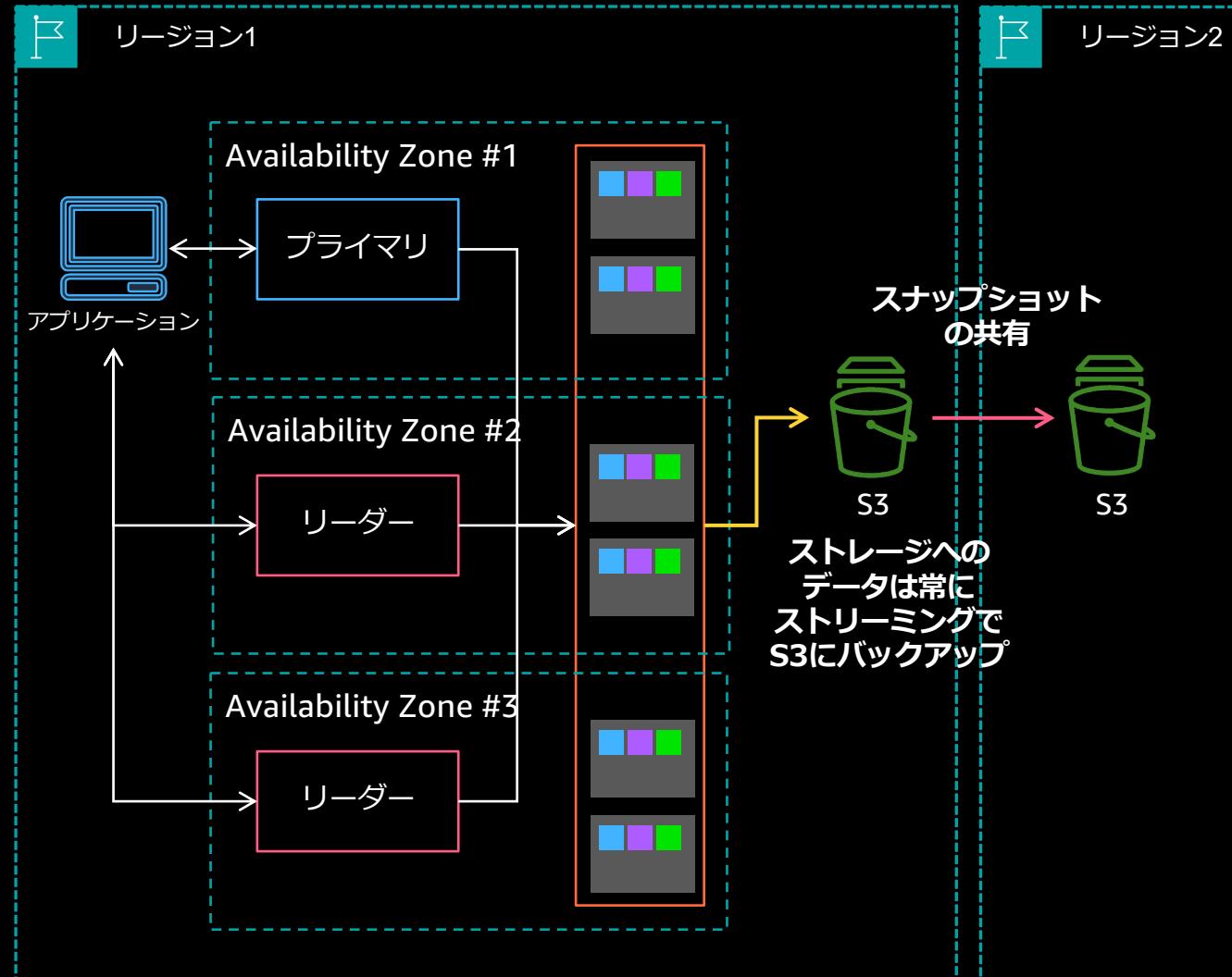
Amazon Aurora の可用性

データ破損: データの整合性が損なわれる問題

Amazon Aurora

- ユーザーエラーなどによりデータが論理的に破損した場合
バックアップからのリストア・リカバリにより対応可能
- ただしシステム停止が必要になる

Aurora のバックアップ & リカバリー



Aurora のバックアップ

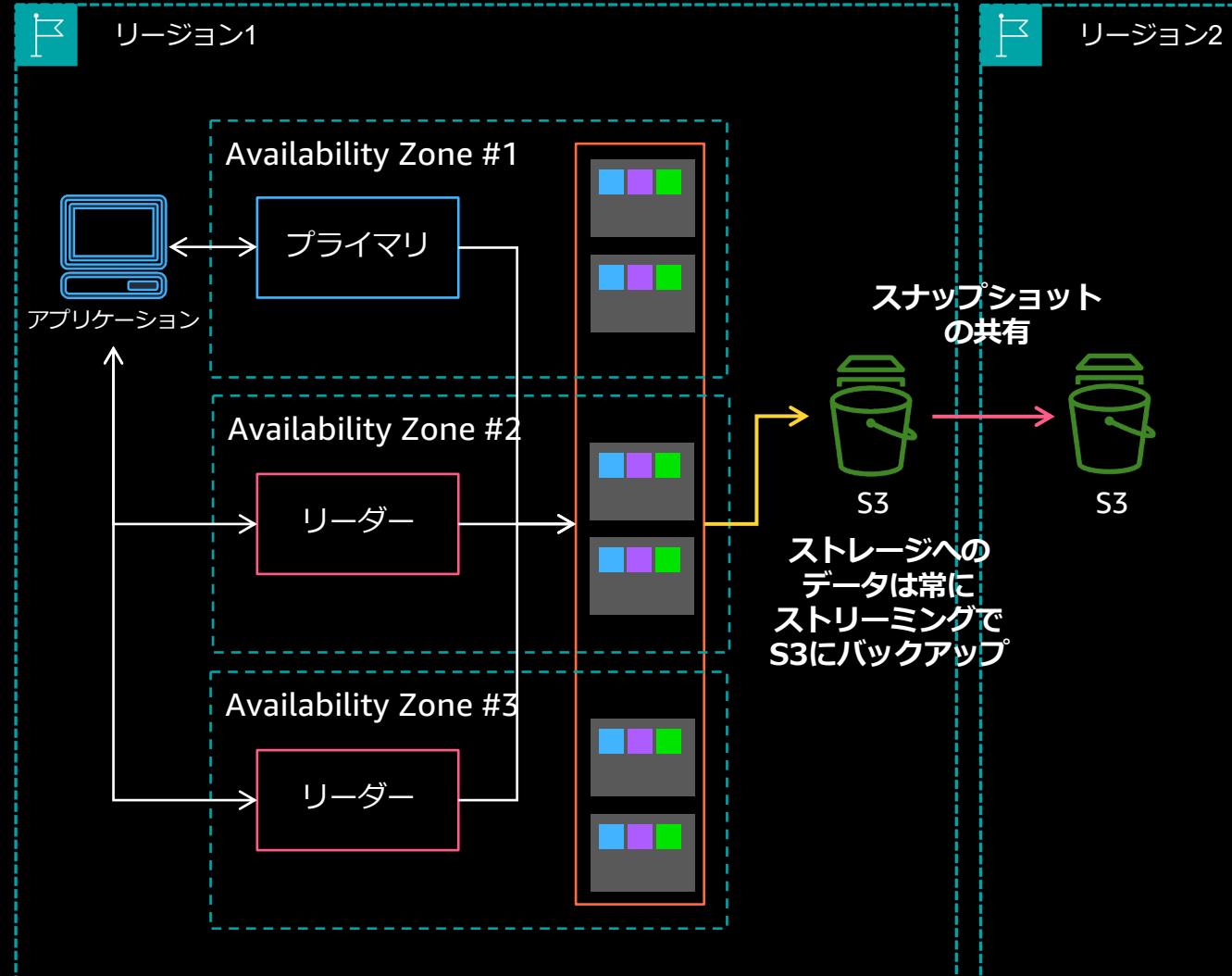
1. 自動バックアップ(必須)
2. 手動スナップショット(任意)

仕組み

- 繼続的にバックアップを S3 に保存
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の任意の時点にリカバリすることが可能 (ポイントインタイムリカバリ : PITR)
- バックアップはストリーミングで実行されバックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを別リージョンにコピーしたり別アカウントと共有ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

Aurora のバックアップ & リカバリー



Aurora のバックアップ

1. 自動バックアップ(必須)
2. 手動スナップショット(任意)

仕組み

- 繼続的に**バックアップを S3 に保存**
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の**任意の時点**に**リカバリ**することが可能 (ポイントインタイムリカバリ : PITR)
- バックアップは**ストリーミングで実行**され、バックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを**別リージョン**にコピーしたり別アカウントと共有ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

Aurora のバックアップコスト

バックアップストレージ

- ・ **無料枠:** DBクラスター サイズの100%までのバックアップストレージは無料
- ・ **自動バックアップ:** 保持期間内のバックアップは無料
- ・ **スナップショット:** 保持期間内のスナップショットは無料
- ・ **追加バックアップ:** 無料枠を超えるバックアップおよび保持期間外のスナップショットは、
GB-月あたり約\$0.021～\$0.023（リージョンにより異なる）
- ・ **削除されたクラスター:** 削除されたクラスターのスナップショットも課金対象

データ転送

- ・ **同一 AZ 内:** Aurora と EC2間のデータ転送は無料
- ・ **AZ 間レプリケーション:** DBクラスター レプリケーションのためのAZ間データ転送は無料
- ・ **リージョン間転送:** GB当たり\$0.01～\$0.02（送信元/送信先リージョンにより異なる）

復元操作

- ・ バックアップからの復元自体に追加料金はなし
- ・ 復元されたDBクラスターは新規インスタンスとして課金開始
- ・ クロスリージョン復元時はデータ転送料金が発生

As of May 2025

詳細はドキュメントを参照 : <https://aws.amazon.com/jp/rds/aurora/pricing/>



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

Aurora のバックアップコスト

バックアップストレージ

- ・ **無料枠:** DBクラスター サイズの100%までのバックアップストレージは無料
- ・ **自動バックアップ:** 保持期間内のバックアップは無料
- ・ **スナップショット:** 保持期間内のスナップショットは無料
- ・ **追加バックアップ:** 無料枠を超えるバックアップおよび保持期間外のスナップショットは、
GB-月あたり約\$0.021～\$0.023（リージョンにより異なる）
- ・ **削除されたクラスター:** 削除されたクラスターのスナップショットも課金対象

データ転送

- ・ **同一 AZ 内:** Aurora と EC2間のデータ転送は無料
- ・ **AZ 間レプリケーション:** DBクラスター レプリケーションのためのAZ間データ転送は無料
- ・ **リージョン間転送:** GB当たり\$0.01～\$0.02（送信元/送信先リージョンにより異なる）

復元操作

- ・ バックアップからの復元自体に追加料金はなし
- ・ 復元されたDBクラスターは新規インスタンスとして課金開始
- ・ クロスリージョン復元時はデータ転送料金が発生

As of May 2025

詳細はドキュメントを参照 : <https://aws.amazon.com/jp/rds/aurora/pricing/>

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

Amazon Aurora

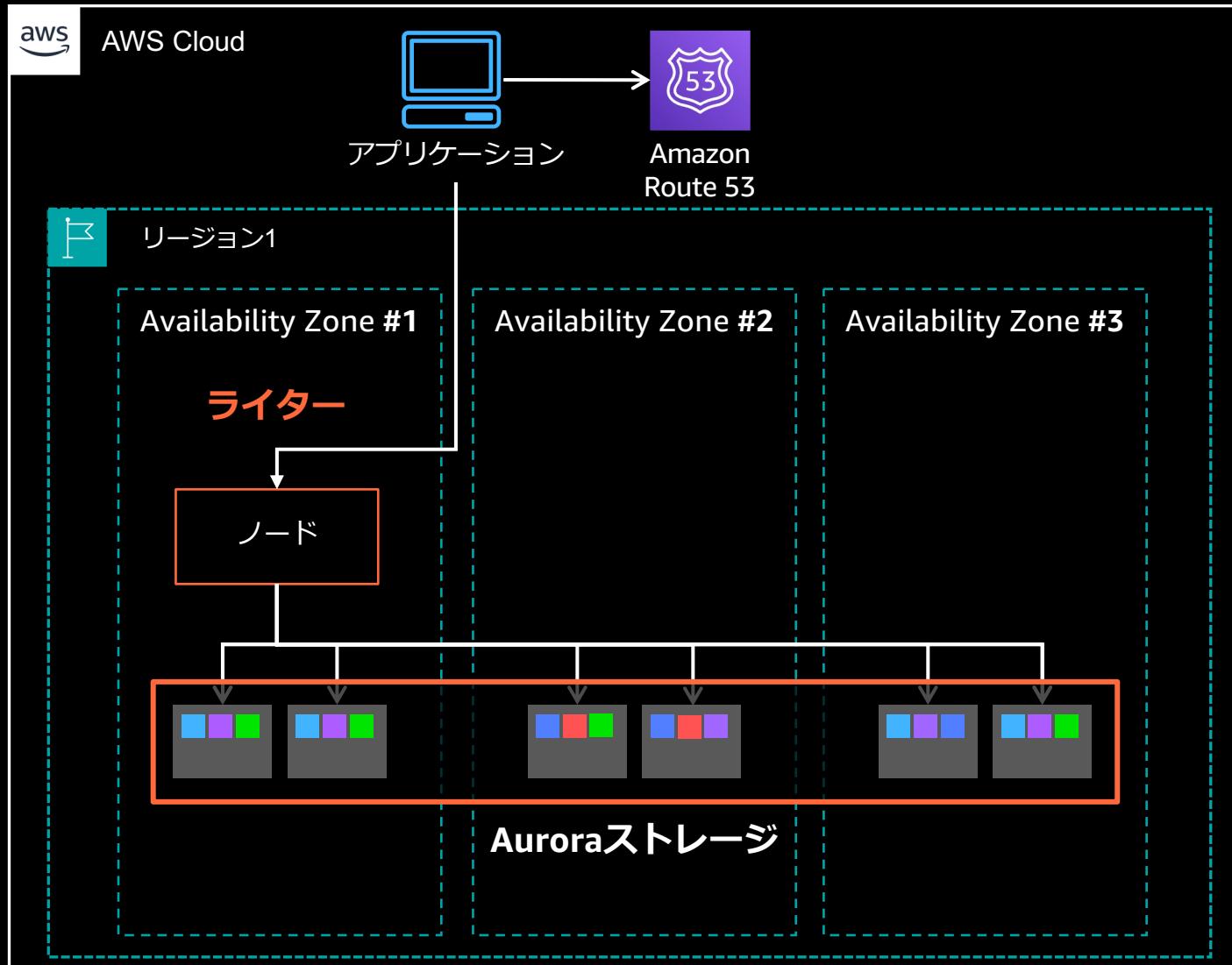
- シングル AZ 構成の場合

インスタンス再作成が必要な為、通常 10 分未満の停止時間が発生

- マルチ AZ 構成を使用した高可用性構成の場合

フェイルオーバーにより、通常 1 分未満でデータベース処理再開可能

シングル AZ 構成



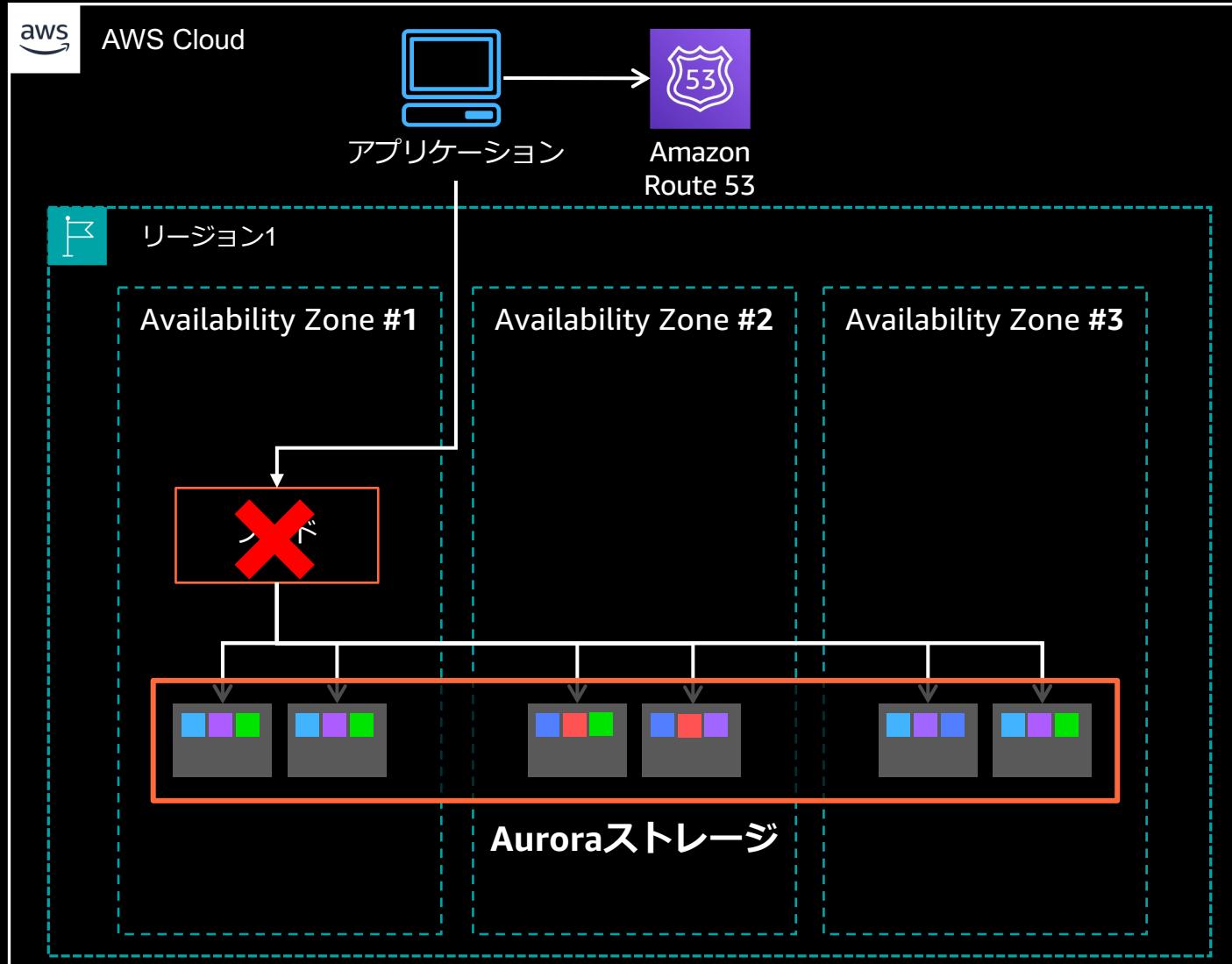
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：別 AZ に手動インスタンス作成が必要

シングル AZ 構成



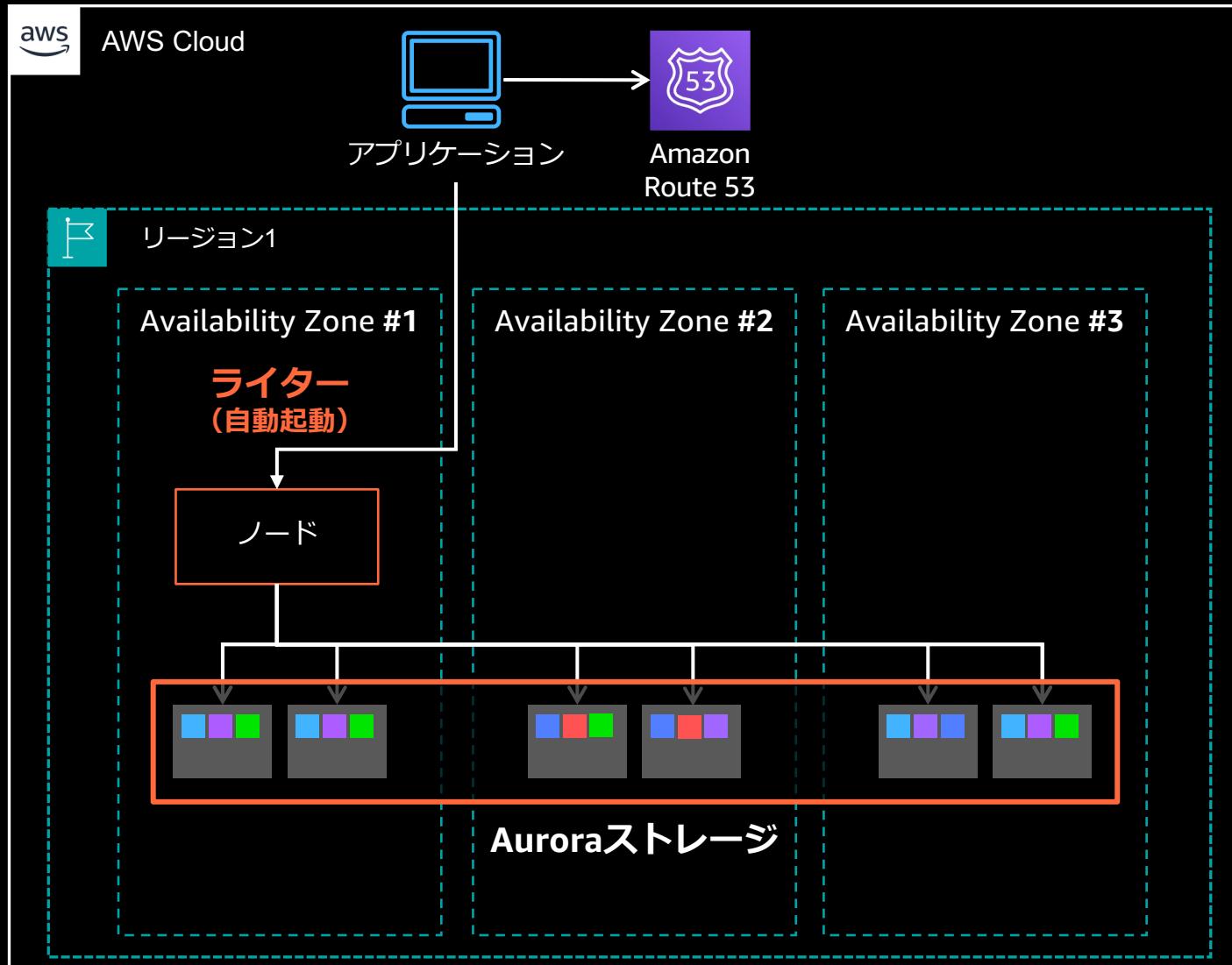
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- **インスタンス障害時 :** 同一 AZ に自動再作成 (10分未満)
- **AZ 全体障害時 :** 別 AZ に手動インスタンス作成が必要

シングル AZ 構成



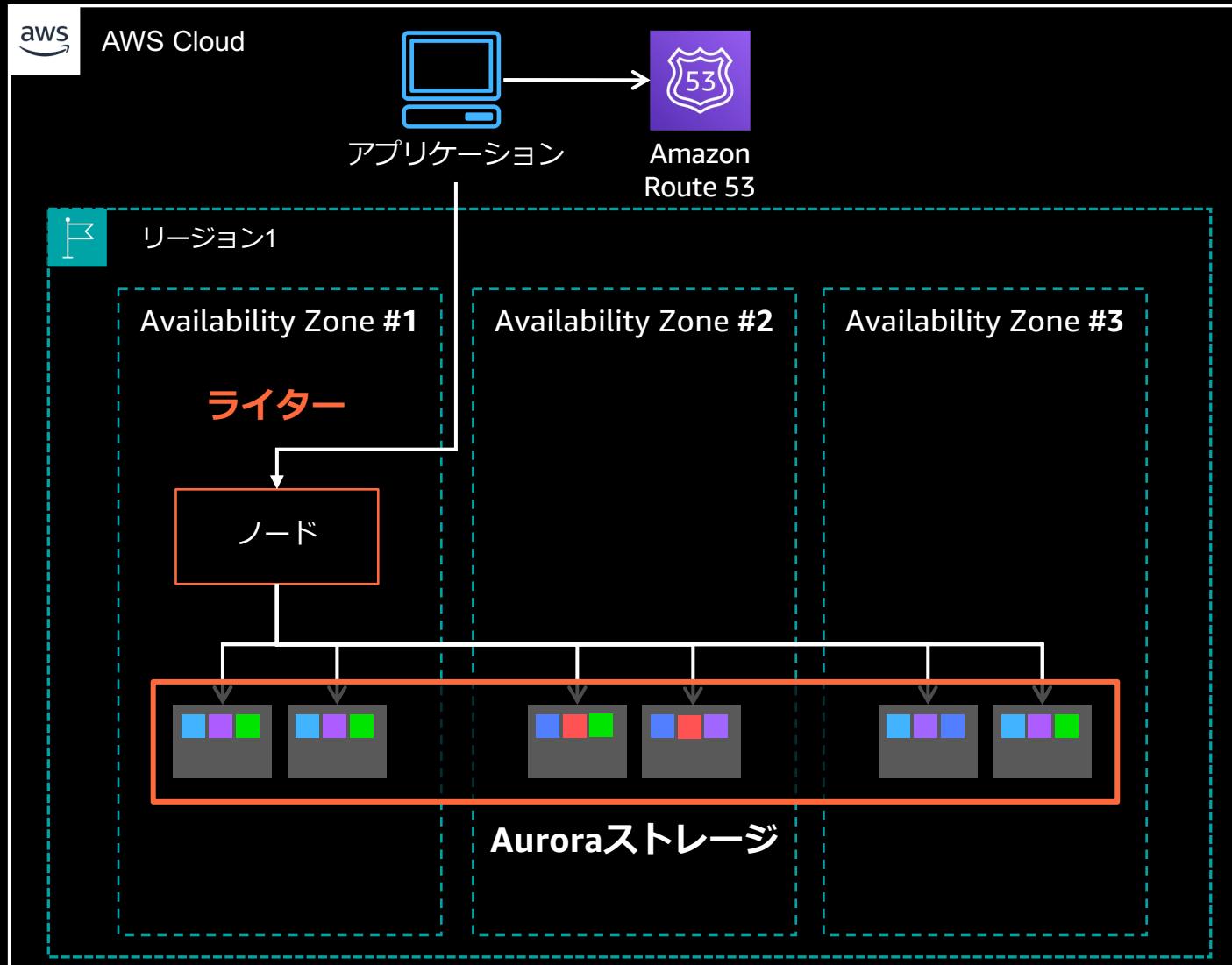
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：
同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：
別 AZ に手動インスタンス作成が必要

シングル AZ 構成



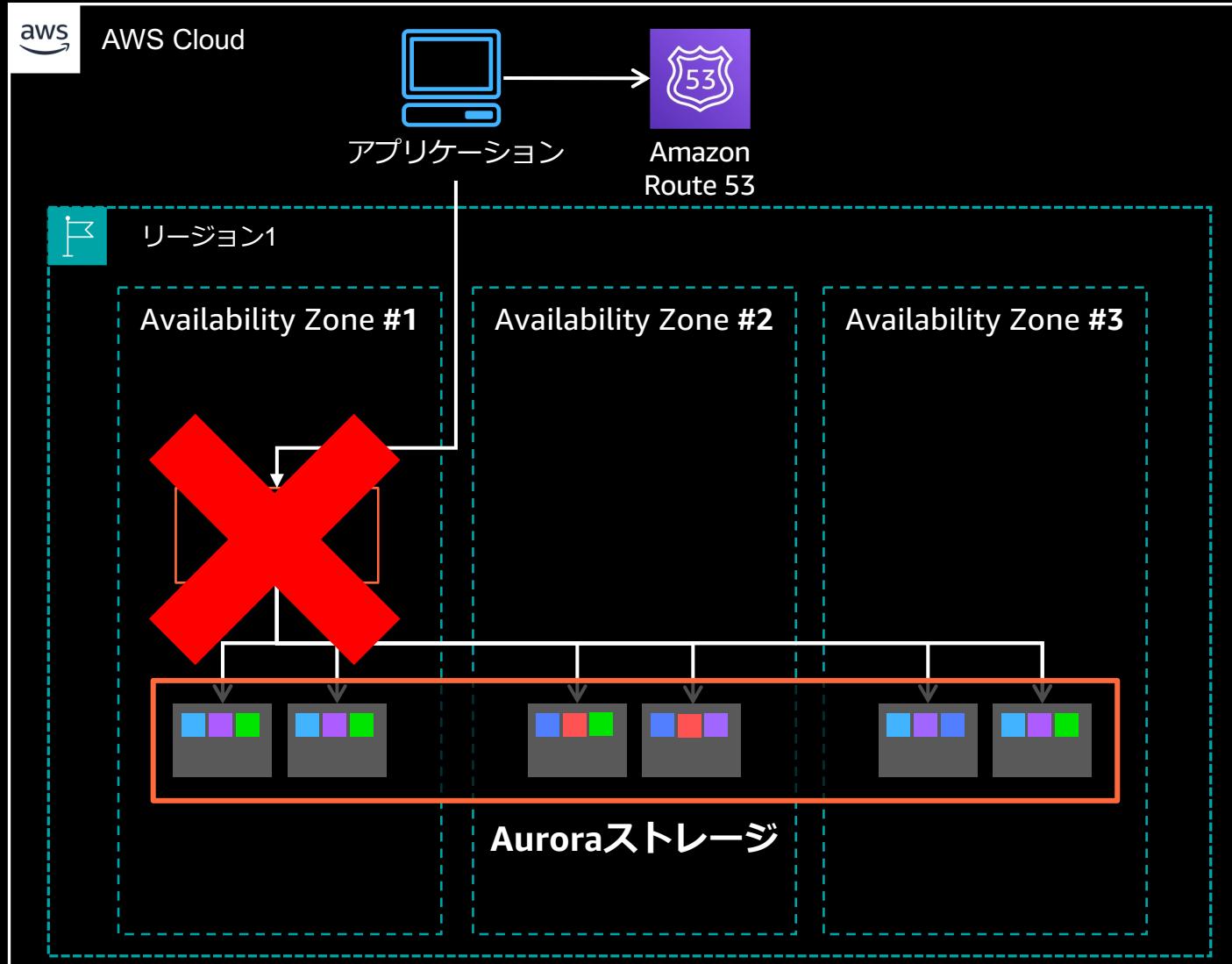
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：別 AZ に手動インスタンス作成が必要

シングル AZ 構成



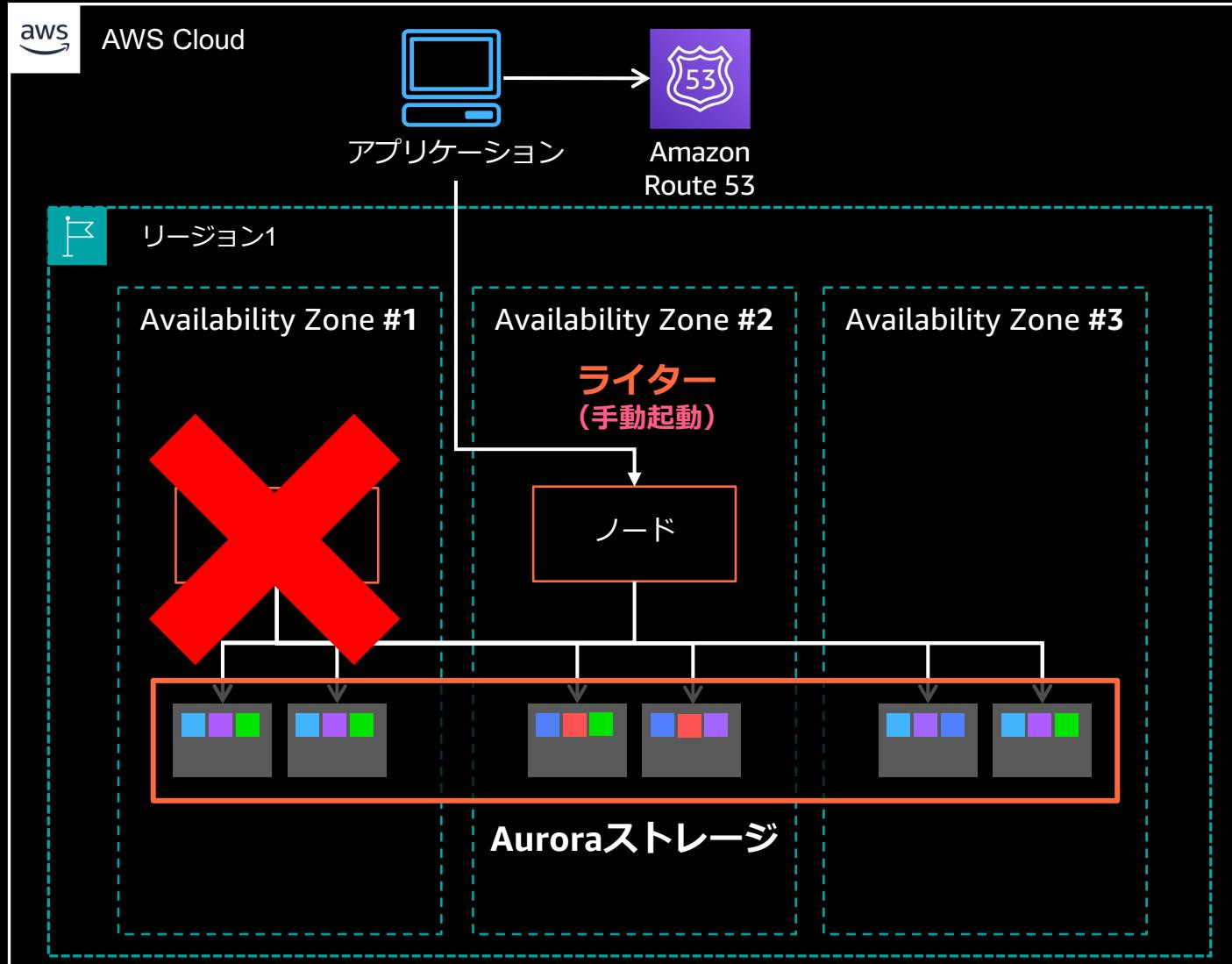
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- **AZ 全体障害時：**別 AZ に手動インスタンス作成が必要

シングル AZ 構成



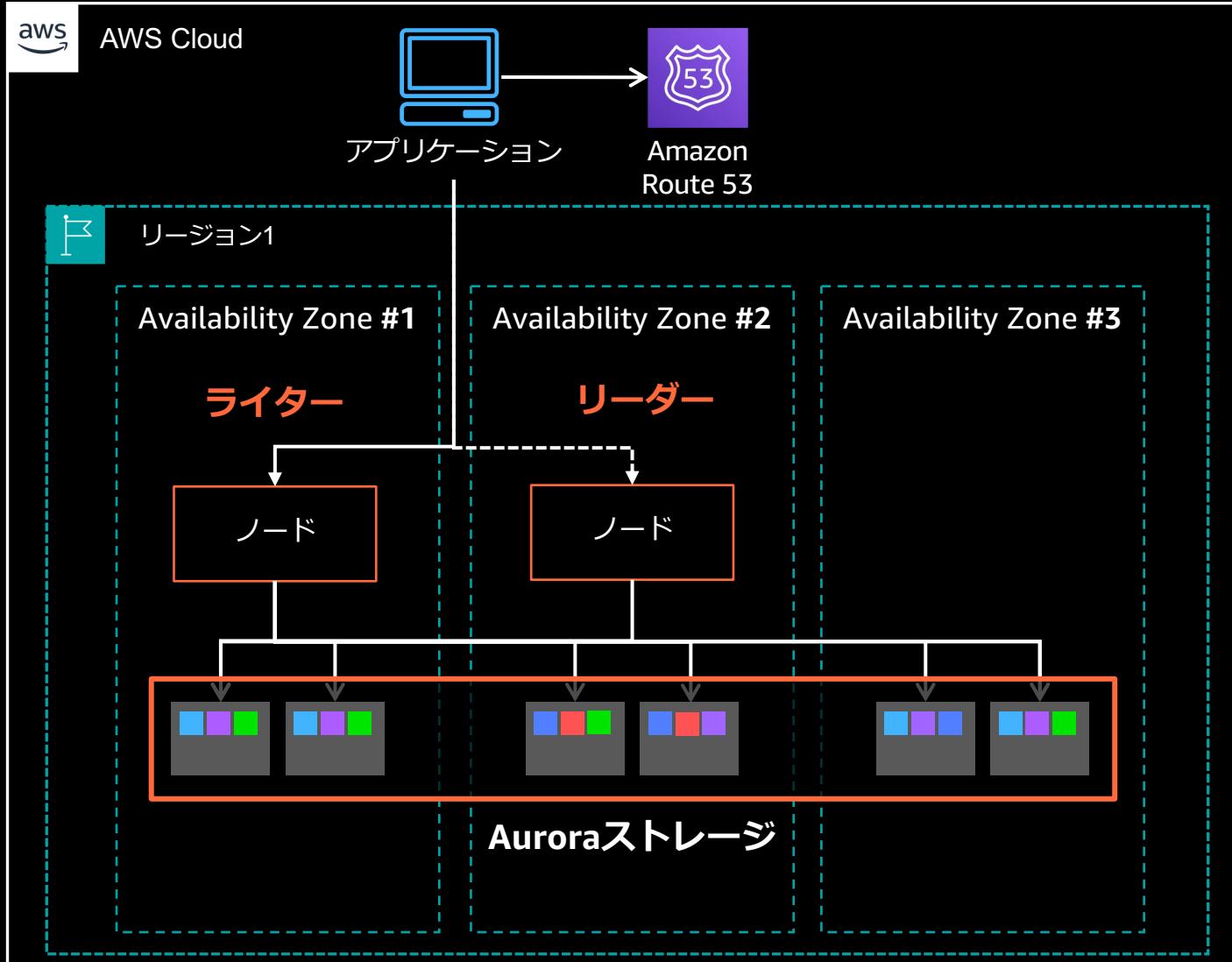
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：
同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：
別 AZ に手動インスタンス作成が必要

マルチ AZ 構成



基本構成

- 複数 AZ にインスタンス分散（最大15台）
- 1 ライター + リーダー (1台以上) 構成
- 3 AZにデータが配置

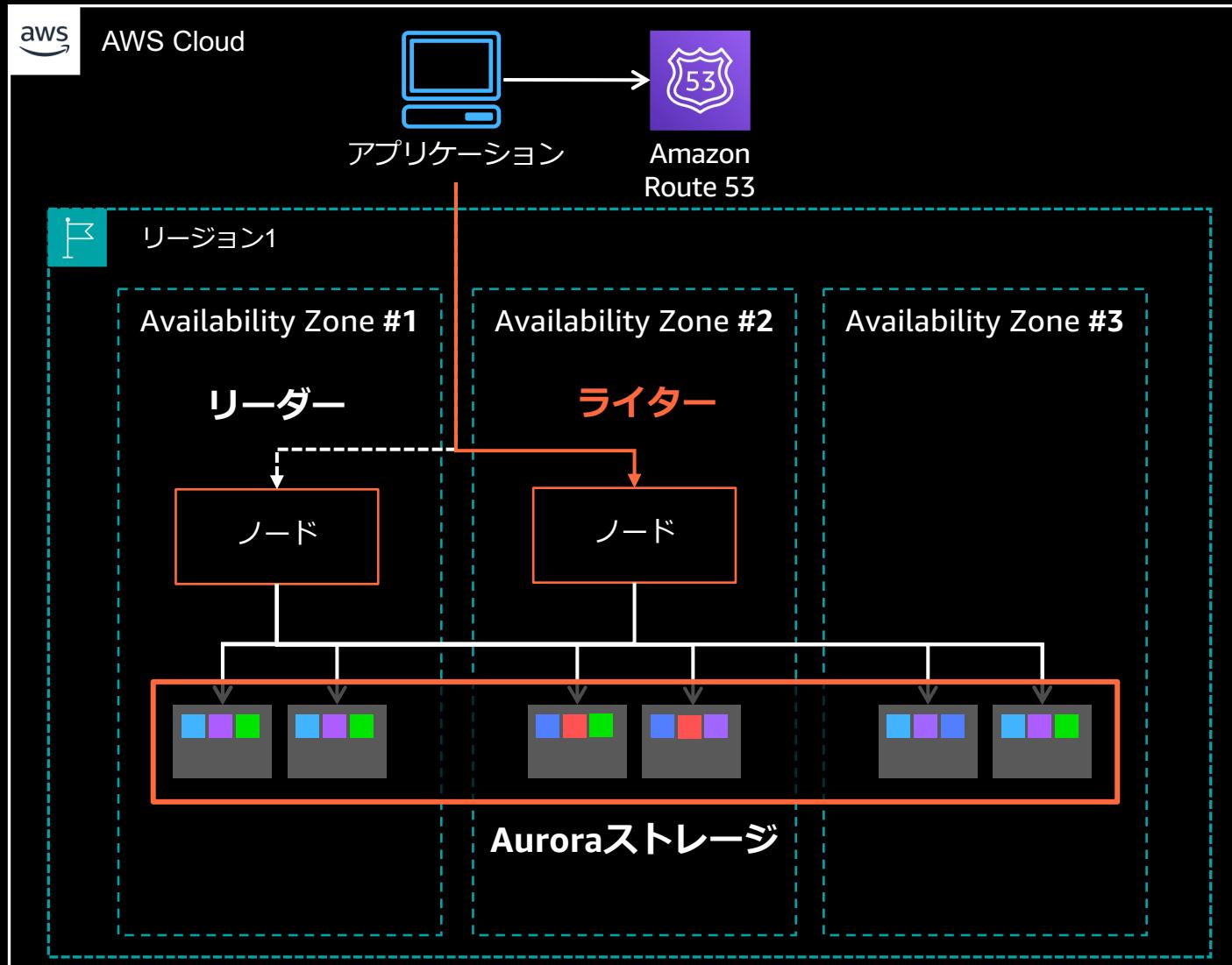
データ同期と可用性

- 非同期レプリケーション
(ミリ秒単位のラグ)
- 共有ストレージにより
フェイルオーバー時のデータ損失なし

運用管理

- リーダーエンドポイントによる
読み取り負荷分散 (ラウンドロビン)
- 外部オブザーバーによる監視と
クオーラム合意
- 自動/手動フェイルオーバー
(Route 53による DNS 切替)

マルチ AZ 構成



基本構成

- 複数 AZ にインスタンス分散（最大15台）
- 1 ライター + リーダー (1台以上) 構成
- 3 AZ にデータが配置

データ同期と可用性

- 非同期レプリケーション
(ミリ秒単位のラグ)
- 共有ストレージにより
フェイルオーバー時のデータ損失なし

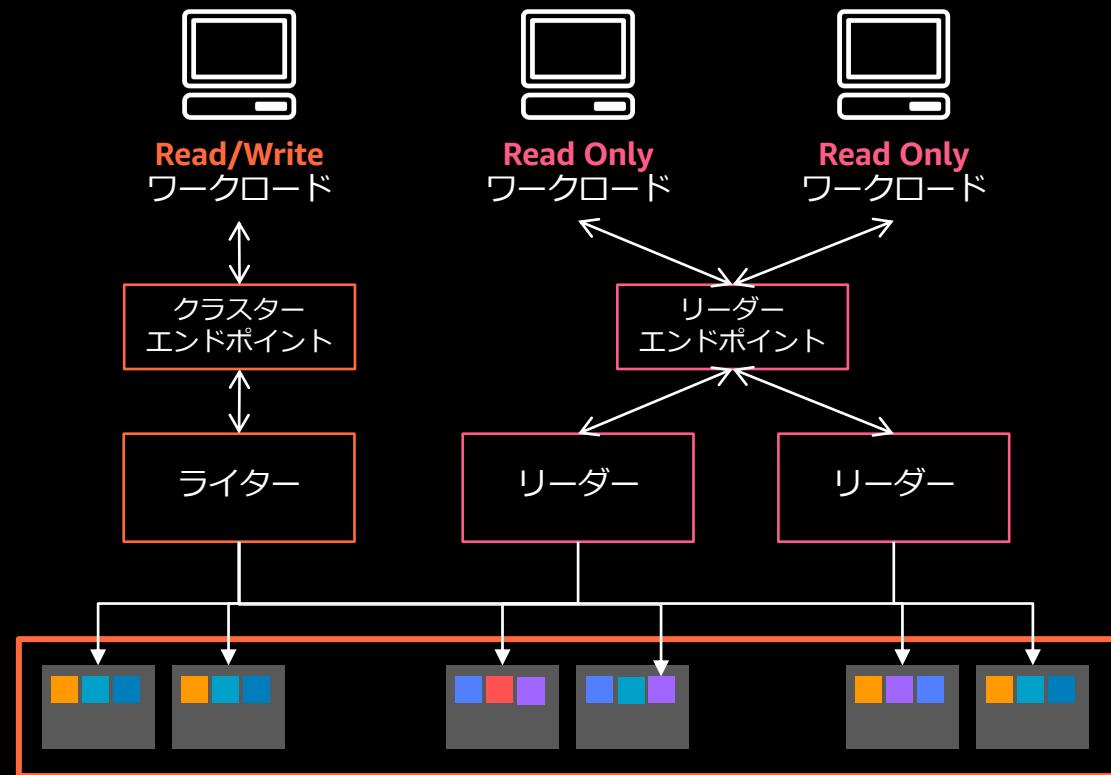
運用管理

- リーダーエンドポイントによる
読み取り負荷分散 (ラウンドロビン)
- 外部オブザーバーによる監視と
クオーラム合意
- 自動/手動フェイルオーバー
(Route 53による DNS 切替)

リーダーインスタンスとエンドポイント



リージョン1



読み取りスケーリング

- 読み取り専用リーダー (Aurora レプリカ) を利用
- 最大 15 台のリーダーで読み取り負荷を分散
- レプリカラグは通常 100ms 未満

効率と監視

- CloudWatch/RDS コンソールでレプリカラグ監視
- 共有ストレージ方式で無駄な書き込み処理なし

接続タイプ

- クラスター エンドポイント : ライター接続用
- リーダー エンドポイント : 読み取り負荷分散用
- インスタンス エンドポイント : 特定インスタンス接続用

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

別リージョンにスナップショットのコピーがある場合

- スナップショットから別リージョンで復元
- 復元には数時間要し、最新データの損失リスクあり (データ量に依存)

Aurora グローバルデータベースを事前構成した場合

- セカンダリリージョンへの手動フェイルオーバーで 30 秒に復旧可能
- ストレージベースのレプリケーションにより、通常1秒未満のレプリカラグを実現

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

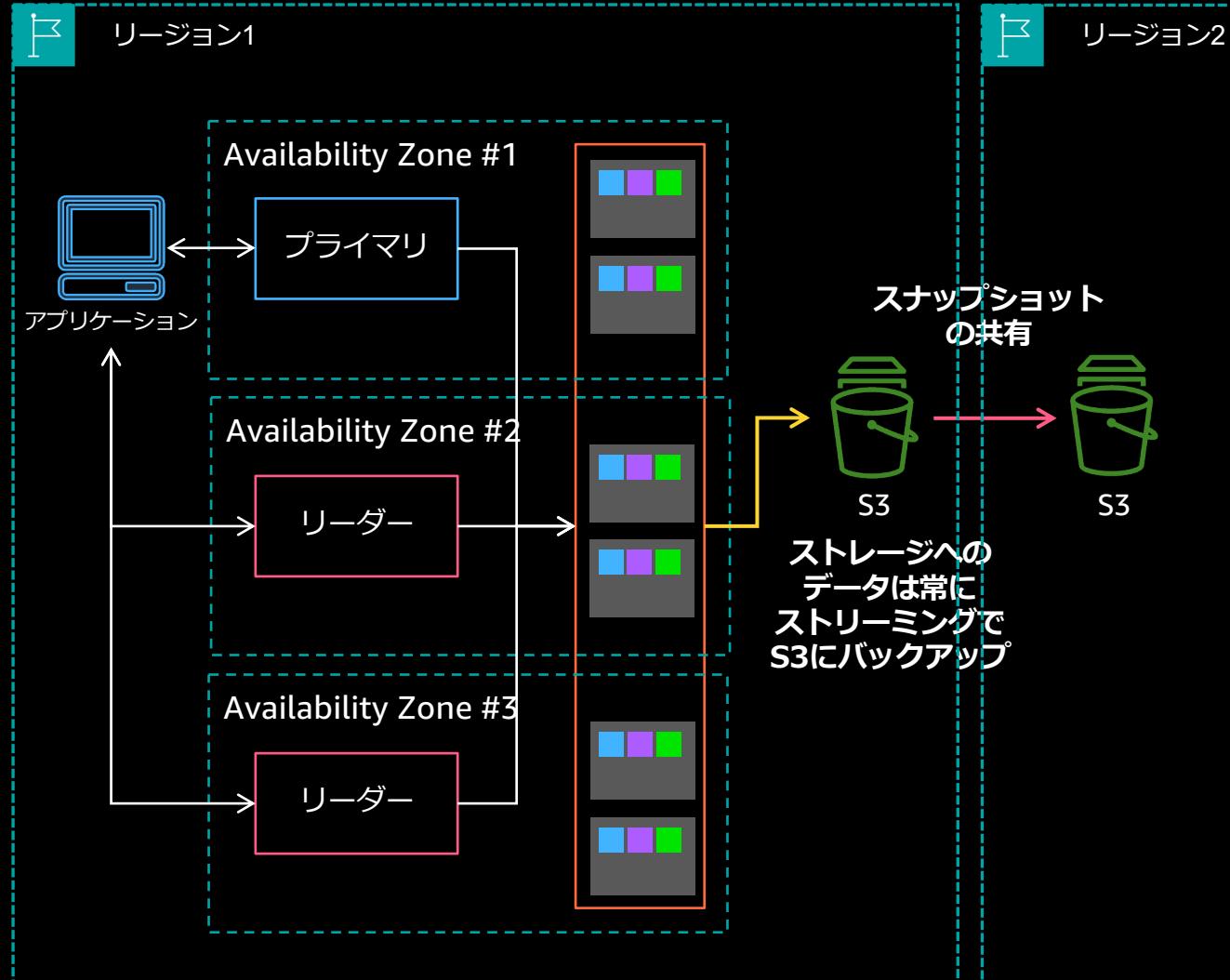
別リージョンにスナップショットのコピーがある場合

- スナップショットから別リージョンで復元
- 復元には数時間要し (データ量に依存)、最新データの損失リスクあり (バックアップ頻度に依存)

Aurora グローバルデータベースを事前構成した場合

- セカンダリリージョンへの手動フェイルオーバーで 30 秒に復旧可能
- ストレージベースのレプリケーションにより、通常 1 秒未満のレプリカラグを実現

再掲：Aurora のバックアップ & リカバリー



Aurora のバックアップ

- 自動バックアップ(必須)
- 手動スナップショット(任意)

仕組み

- 継続的にバックアップを S3 に保存
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の任意の時点にリカバリすることが可能 (ポイントインタイムリカバリ : PITR)
- バックアップはストリーミングで実行されバックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを別リージョンにコピーしたり別アカウントと共有ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

別リージョンにスナップショットのコピーがある場合

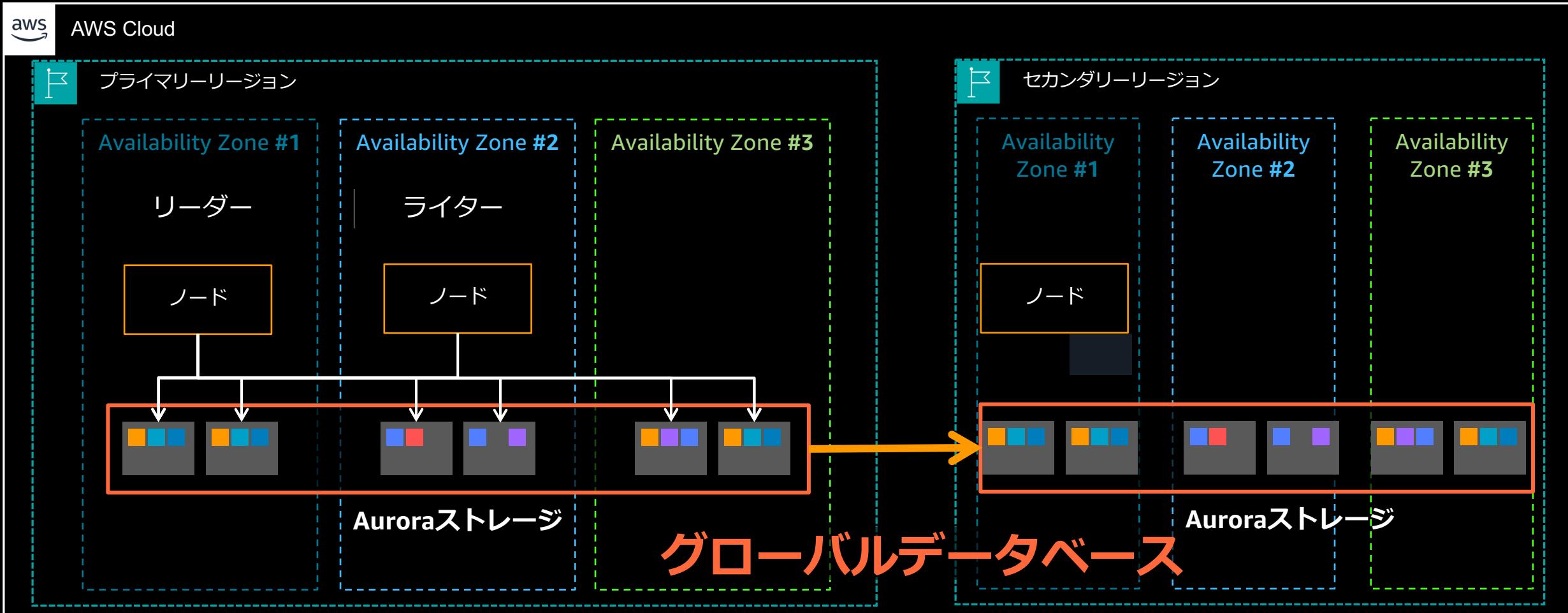
- スナップショットから別リージョンで復元
- 復元には数時間要し、最新データの損失リスクあり (データ量に依存)

Aurora グローバルデータベースを事前構成した場合

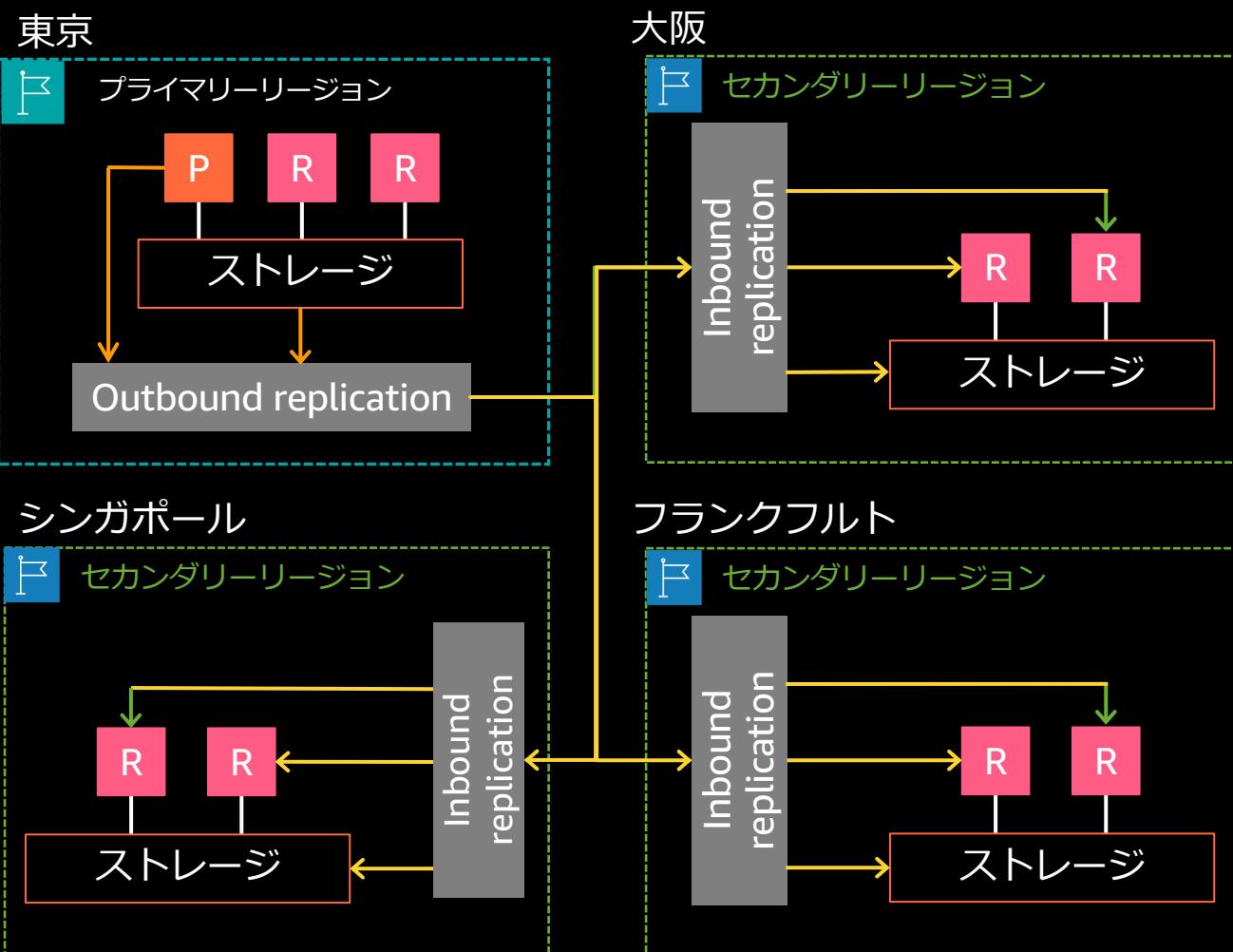
- セカンダリリージョンへの手動フェイルオーバーで通常 30 秒未満で復旧可能
- ストレージベースのレプリケーションにより、通常 1 秒未満のレプリカラグを実現

Aurora グローバルデータベース

複数の AWS リージョンにまたがる複数の Aurora DB クラスター : リージョンをまたいだ耐障害性向上



Amazon Aurora グローバルデータベース



高スループット:

低レプリケーションラグ:

- 高負荷状況下でも、リージョン間の
レプリケーションラグは通常1秒未満

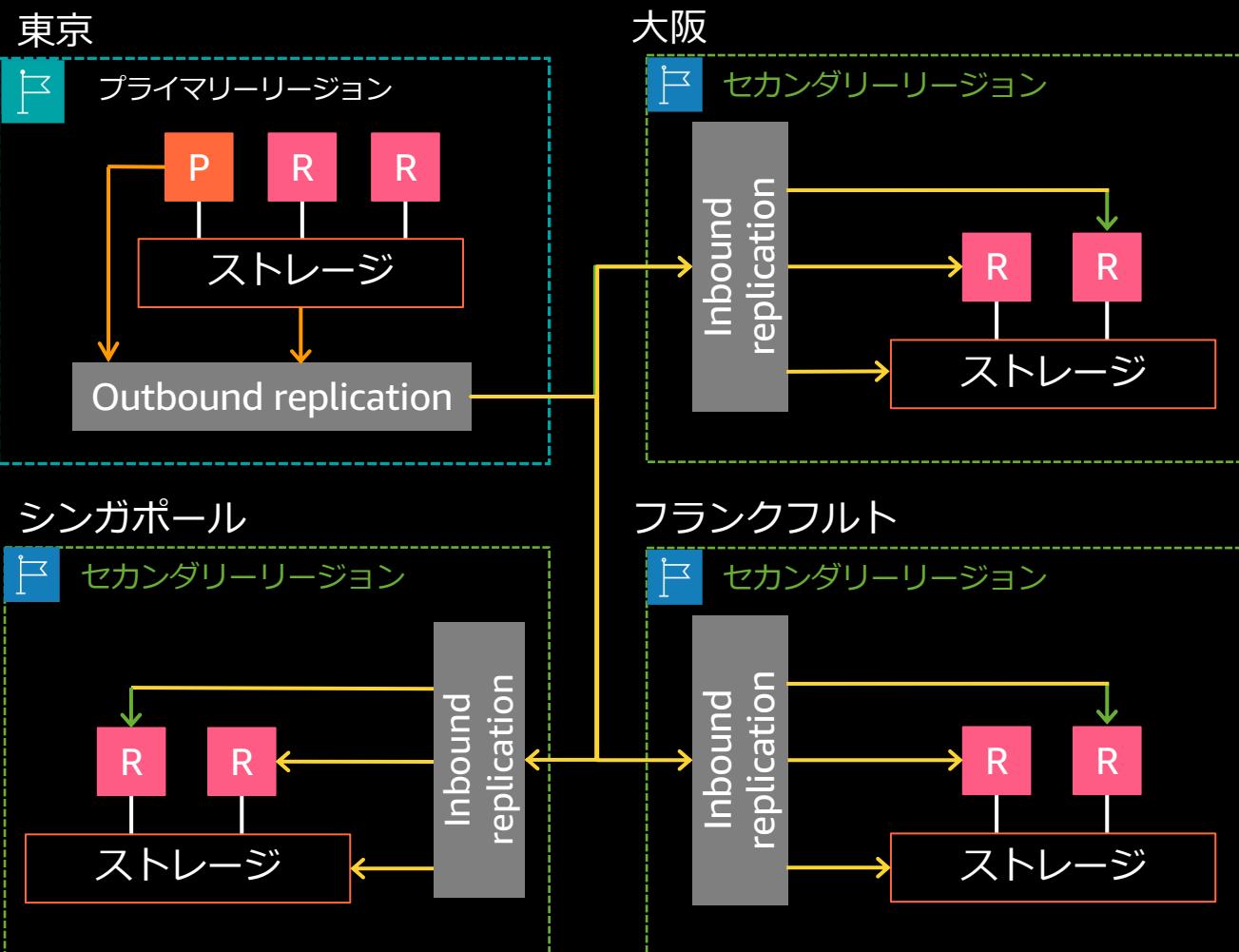
迅速な災害復旧:

- 計画的なスイッチオーバー:
復旧時間が通常 30 秒未満、データ損失なし
- 計画外のフェールオーバー:
リージョン障害後、通常 1 分未満でリカバリ

柔軟な構成:

- 最大 10 個のセカンダリリージョンをサポート
- 既存の Aurora クラスターからインプレースでの
グローバルデータベースへの拡張が可能

Amazon Aurora グローバルデータベース



高スループット:

低レプリケーションラグ:

- 高負荷状況下でも、リージョン間の
レプリケーションラグは通常1秒未満

迅速な災害復旧:

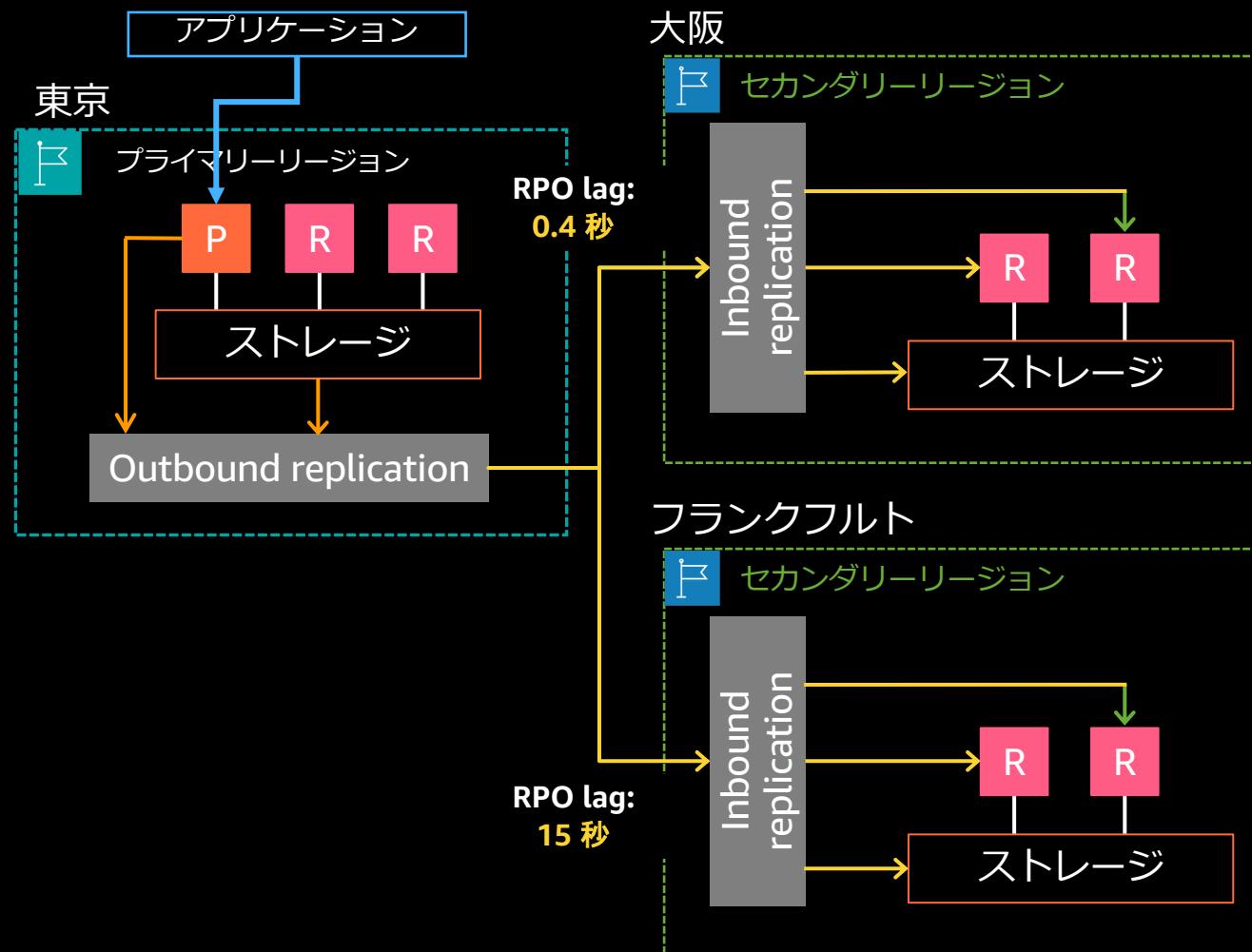
- 計画的なスイッチオーバー:
復旧時間が通常 30 秒未満、データ損失なし
- 計画外のフェールオーバー:
リージョン障害後、通常 1 分未満でリカバリ

柔軟な構成:

- 最大 10 個のセカンダリリージョンをサポート
- 既存の Aurora クラスターからインプレースでの
グローバルデータベースへの拡張が可能

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



機能:

- PostgreSQL ベースの Aurora Global Database
- 最大 RPO (目標復旧ポイント) を定義可能

動作:

- 設定した RPO 上限を全セカンダリリージョンが超過した場合、いずれかが追いつくまで PRIMARY クラスターの書き込みを一時停止

設定方法:

- rds.global_db_rpo パラメータ
- 20 秒 ~ 68 年の範囲で設定

監視メトリクス:

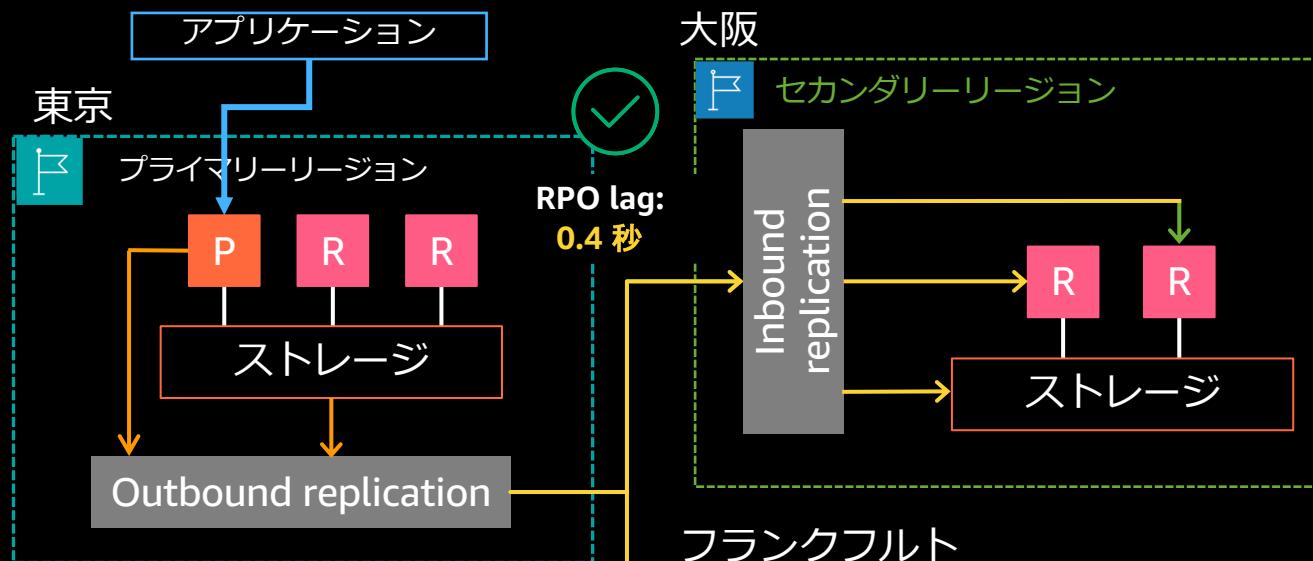
- AuroraGlobalDBRPOLag: セカンダリクラスターの RPO ラグ時間 (秒)

詳細はドキュメント参照 :

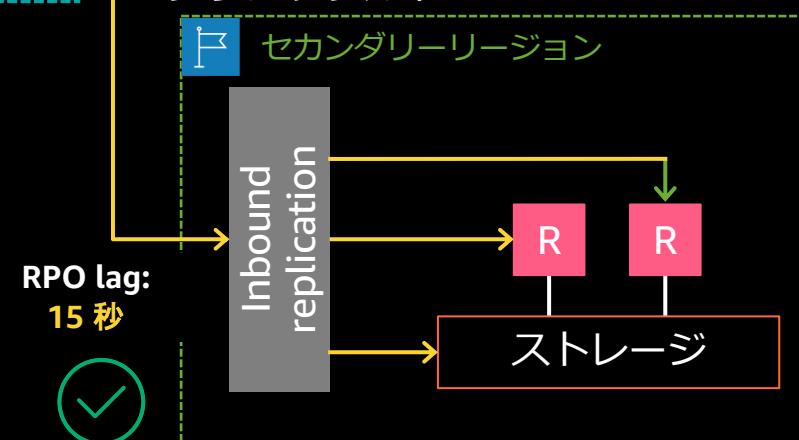
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒



例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 0.4 秒
- ・ フランクフルト - ラグ 15 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 未満

結果:

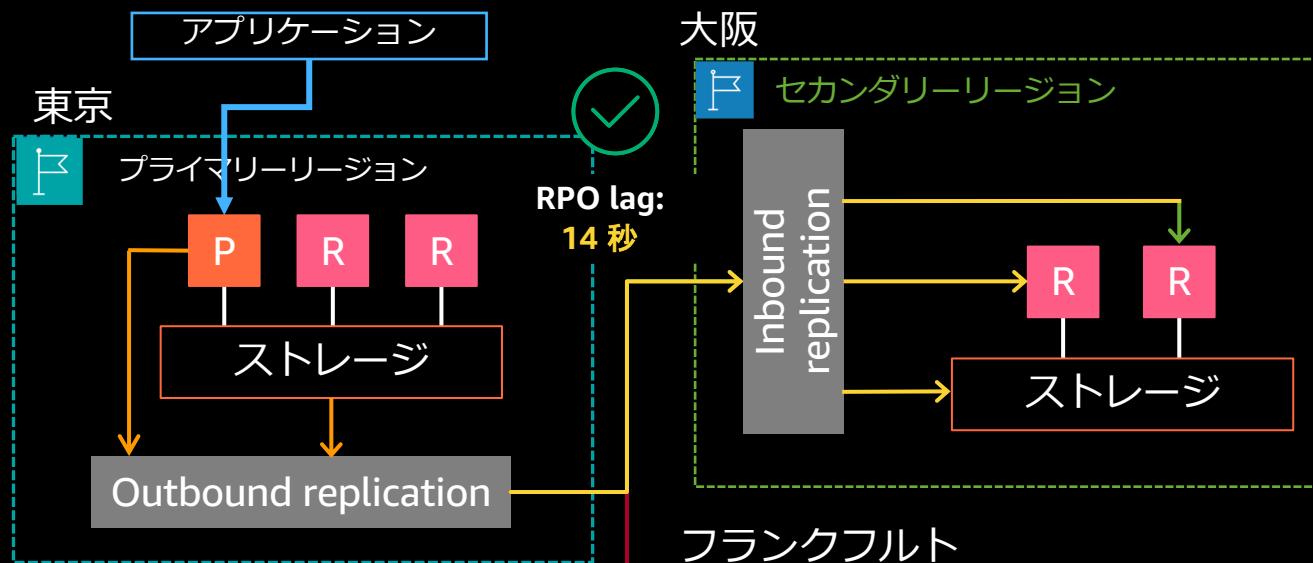
- ・ 全てのセカンダリーが基準値内

プライマリインスタンス:

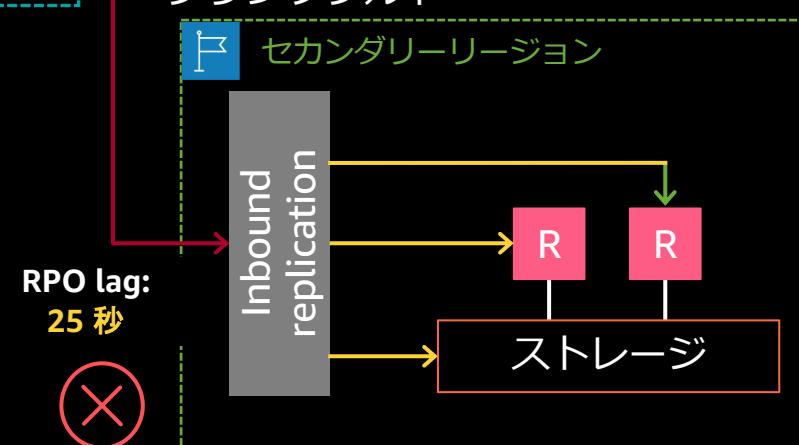
- ・ トランザクション継続

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒



例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 14 秒
- ・ フランクフルト - ラグ 25 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 以上

結果:

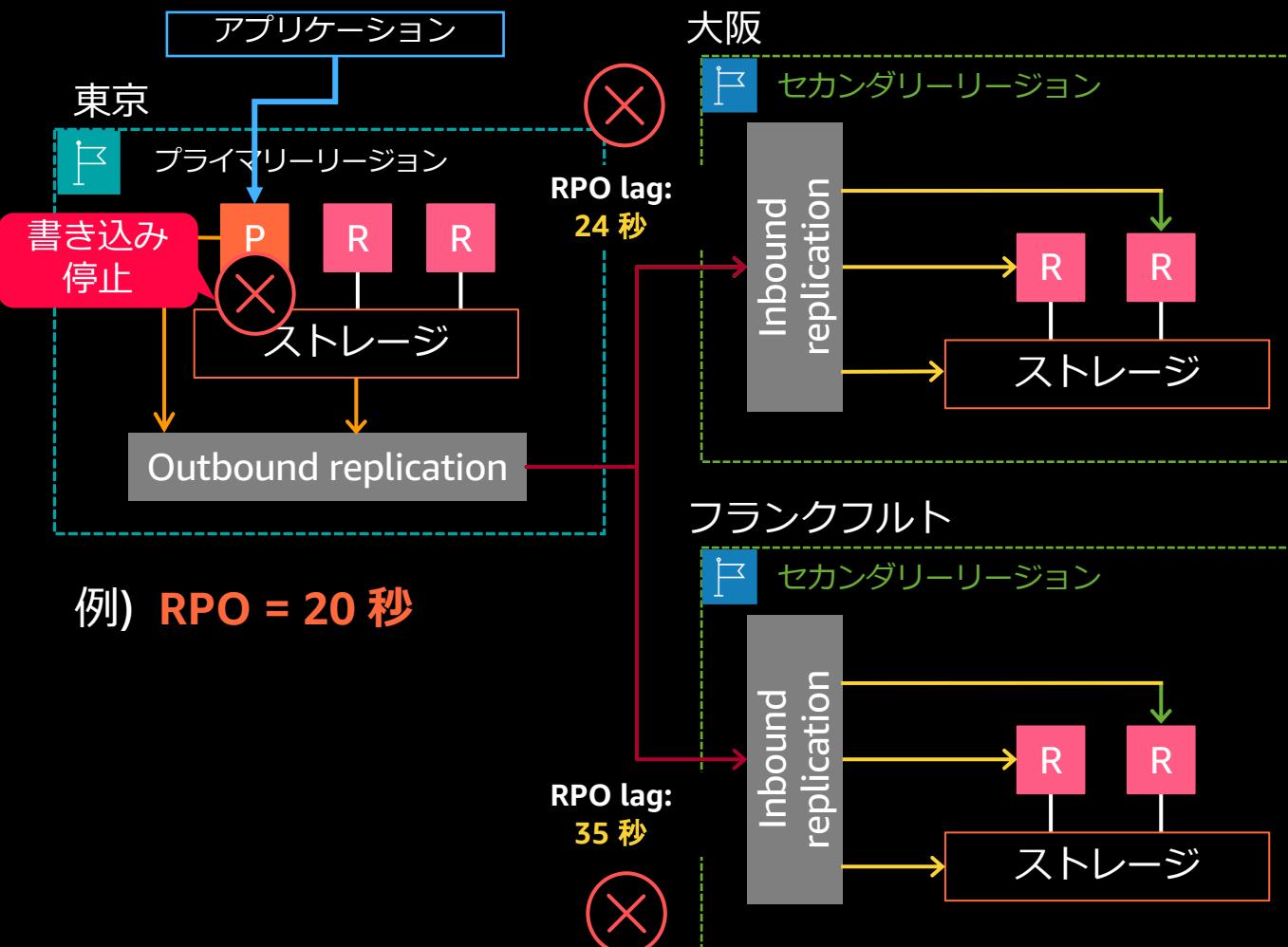
- ・ 1 つのセカンダリー (大阪) が基準値内

プライマリインスタンス:

- ・ トランザクション継続

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒

例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 24 秒
- ・ フランクフルト - ラグ 35 秒

上限確認:

- ・ 大阪 - 20 秒以上
- ・ フランクフルト - 20 秒以上

結果:

- ・ 全てのセカンダリーが基準値超過

プライマリインスタンス:

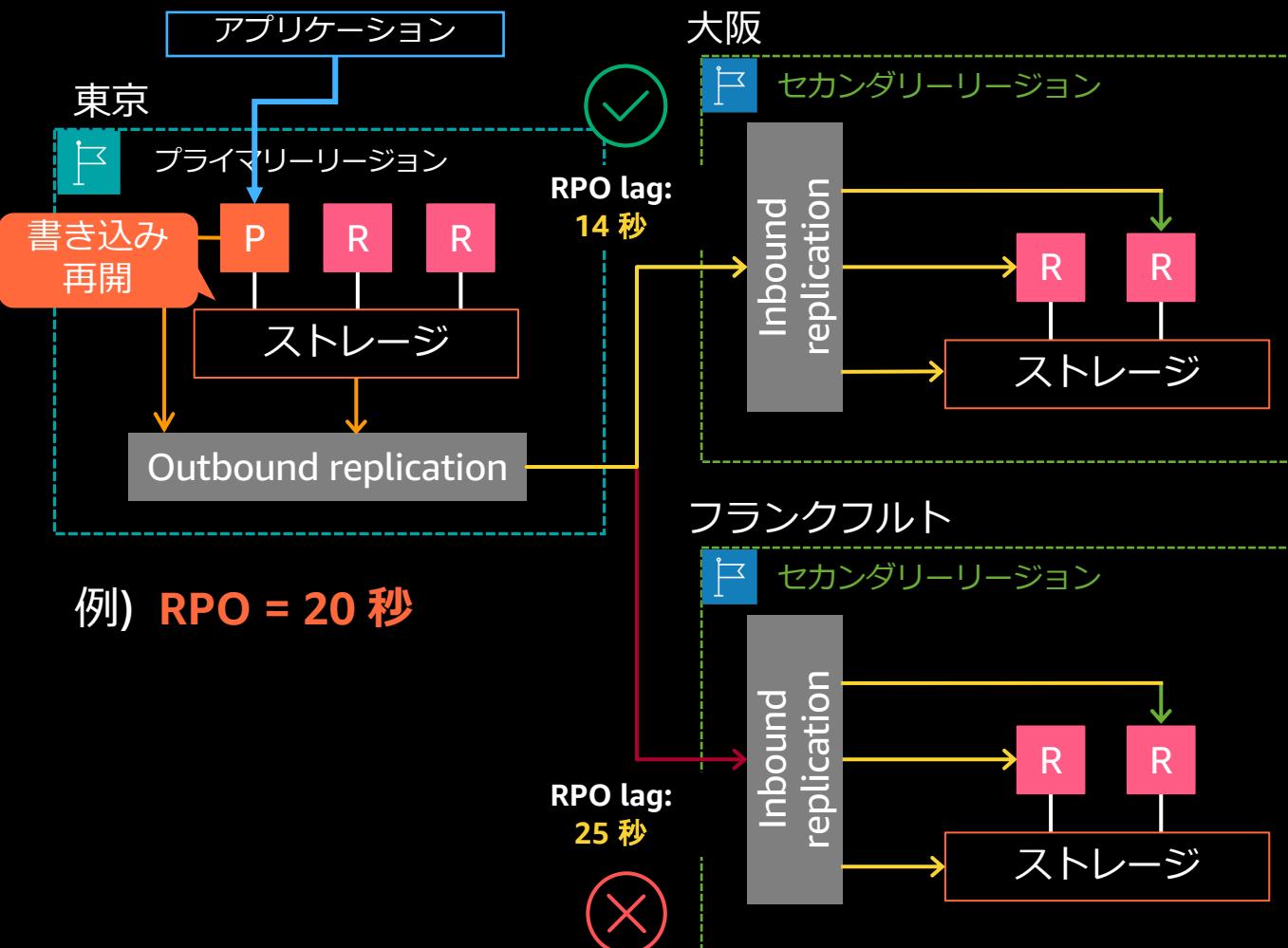
- ・ トランザクション停止

詳細はドキュメント参照 :

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

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 14 秒
- ・ フランクフルト - ラグ 25 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 以上

結果:

- ・ 1 つのセカンダリー (大阪) が基準値内

プライマリインスタンス:

- ・ トランザクション再開

グローバルデータベース制限事項

- ・ 特定の AWS リージョンと特定の Aurora MySQL および Aurora PostgreSQL バージョンでのみ利用可能
- ・ メモリ最適化インスタンス (db.r5 以上のインスタンス推奨)
- ・ Aurora Serverless v1、Aurora バックトラック機能は利用不可
- ・ クロスリージョンスイッチオーバー/フェールオーバーは、プライマリとセカンダリが同一のメジャー/マイナーバージョンの場合のみ実行可能
- ・ セカンダリ DB クラスターでは Aurora Auto Scaling 利用不可
- ・ グローバルデータベース内の Aurora DB クラスターを個別に停止/開始不可
- ・ Secrets Manager は未サポート (事前に Secrets Manager 統合を無効化する必要)
- ・ Aurora PostgreSQL 16.6, 15.10, 14.15, 13.18, 12.22 より前と、Aurora MySQL 3.09 より前で、計画外イベント中のセカンダリリージョンの読み取り可用性に未対応
- ・ PostgreSQL 固有の制限:
 - ・ セカンダリクラスターでクラスターキャッシュ管理がサポートされない
 - ・ RPO 機能が有効な場合、メジャーバージョンアップグレード不可

詳細はドキュメントを参照: https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html#aurora-global-database.limitations



グローバルデータベースのコスト

インスタンス:

- プライマリおよびセカンダリリージョンの各インスタンスに対して、標準の Aurora インスタンス料金

ストレージ:

- 各リージョンのストレージ使用量に対して標準の Aurora ストレージ料金

レプリケーション:

- プライマリリージョンから各セカンダリリージョンへのレプリケートされた書き込み I/O 操作に対して課金
- プライマリリージョンの書き込み I/O 操作と同数のレプリケート I/O 操作が各セカンダリリージョンで実行

データ転送:

- リージョン間のデータ転送に対して、標準の AWS クロスリージョンデータ転送料金

バックアップ:

- 各リージョンのバックアップストレージに対して標準料金

詳細はドキュメントを参照: https://aws.amazon.com/jp/rds/aurora/pricing/#Aurora_Global_Database_costs

グローバルデータベースのコスト

インスタンス:

- プライマリおよびセカンダリリージョンの各インスタンスに対して、標準の Aurora インスタンス料金

ストレージ:

- 各リージョンのストレージ使用量に対して標準の Aurora ストレージ料金

レプリケーション:

- プライマリリージョンから各セカンダリリージョンへのレプリケートされた書き込み I/O 操作に対して課金
- プライマリリージョンの書き込み I/O 操作と同数のレプリケート I/O 操作が各セカンダリリージョンで実行

データ転送:

- リージョン間のデータ転送に対して、標準の AWS クロスリージョンデータ転送料金

バックアップ:

- 各リージョンのバックアップストレージに対して標準料金

詳細はドキュメントを参照: https://aws.amazon.com/jp/rds/aurora/pricing/#Aurora_Global_Database_costs

まとめ

再掲：データベースの可用性：RTO / RPO

RPO: Recovery Point Objective

- リカバリポイント目標
- 障害時に、どの時点のデータまでリカバリ可能か

RTO: Recovery Time Objective

- リカバリ時間目標
- 障害時に、どれだけの時間でリカバリを完了させるか

まとめ

データベースの可用性設計

- RTO/RPO 要件：許容できる復旧時間と復旧ポイント
- 想定する障害の種類と対応策
- コストと運用負荷のバランス

Aurora は様々な障害に対する多層防御を提供

- ディスク障害 : Aurora ストレージで 3 AZ 6 箇所に複製
- データ破損 : 自己修復機能とバックアップ・リカバリ (ポイントインタイムリカバリ)
- DB サーバー/データセンター障害 : 自動フェイルオーバー (マルチ AZ 構成)
- 大規模障害 (リージョン障害) : Global Database による地理的冗長性

まとめ

データベースの可用性設計

- RTO/RPO 要件：許容できる復旧時間と復旧ポイント
- 想定する障害の種類と対応策
- コストと運用負荷のバランス

Aurora は様々な障害に対する多層防御を提供

- ディスク障害 : Aurora ストレージで 3 AZ 6 箇所に複製
- データ破損 : 自己修復機能とバックアップ・リカバリ (ポイントインタイムリカバリ)
- DB サーバー/データセンター障害 : 自動フェイルオーバー (マルチ AZ 構成)
- 大規模障害 (リージョン障害) : Global Database による地理的冗長性

「障害を前提とした設計 (Design for Failure)」の考え方に基づき
ビジネス要件に合わせたアーキテクチャを選択することが重要

Thank you!

Yukki

Partner Solutions Architect

@YukkiTakahashi

AWS Black Belt Online Seminar

Amazon Aurora : 可用性

Yukki (高橋 敏行)

Partner Solutions Architect

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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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)

> Who am I ?

Yukki (ゆつきー) 別名: 高橋敏行

パートナー技術統括本部
テクニカルイネーブルメント部
パートナーソリューションアーキテクト

主にパートナー様の育成・技術支援を担当
前職はデータベースエンジニア

好きな AWS のサービス : **Amazon RDS and Amazon Aurora**



@YukkiTakahashi



アジェンダ

1. Amazon Aurora

- サービス概要

2. データベースの可用性

- 障害の種類
- RTO/RPO

3. Amazon Auroraで実現する高可用性

- クオーラムモデル
- マルチ AZ 構成
- Aurora Global Database

4. まとめ

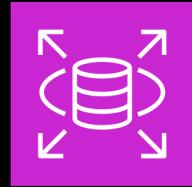
AWS が提供する クラウドデータベース



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

Amazon RDS と Amazon Aurora

リレーショナルデータベース



Amazon RDS

- ✓ 6つの一般的なデータベースエンジンから選べるマネージド型リレーショナルデータベースサービス
- ✓ データベースエンジンのオリジナルに近い機能で提供

MySQL

PostgreSQL

MariaDB

SQL Server

Oracle Database

IBM Db2



Amazon Aurora

- ✓ オープンソースデータベースをクラウド向けに再設計したマネージド型リレーショナルデータベースサービス
- ✓ より高いパフォーマンスと可用性を提供

MySQL

PostgreSQL

DSQL

Amazon RDS と Amazon Aurora

リレーショナルデータベース



Amazon RDS

- ✓ 6つの一般的なデータベースエンジンから選べるマネージド型リレーショナルデータベースサービス
- ✓ データベースエンジンのオリジナルに近い機能で提供

MySQL

PostgreSQL

MariaDB

SQL Server

Oracle Database

IBM Db2



Amazon Aurora

- ✓ オープンソースデータベースをクラウド向けに再設計したマネージド型リレーショナルデータベースサービス
- ✓ より高いパフォーマンスと可用性を提供

MySQL

PostgreSQL

DSQL

Amazon Aurora

Enterprise database at open source price

AWS が提供するマネージドサービス



MySQL と PostgreSQL との互換性

オープンソースデータベースのシンプルさと費用対効果

商用データベースのスループットと可用性

シンプルな従量課金制

※ Aurora DSQL は別の機会でご説明します

データベース管理のフルマネージ化

オンプレミス

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

セルフマネージ

仮想サーバー

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

Amazon EC2

データベースサービス

- アプリケーション最適化
- 拡張性
- 高可用性
- DB バックアップ
- DB パッチ適用
- DB インストール/構築
- OS パッチ適用
- OS インストール
- サーバーメンテナンス
- ハードウェア資産管理
- 電源/ネットワーク/空調

Amazon Aurora 等

AWS マネージ



データベース管理のフルマネージ化

オンプレミス

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

セレクマネージ

仮想サーバー

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

データベースサービス

アプリケーション最適化

拡張性

高可用性

DB バックアップ

DB パッチ適用

DB インストール/構築

OS パッチ適用

OS インストール

サーバーメンテナンス

ハードウェア資産管理

電源/ネットワーク/空調

Amazon Aurora 等



Amazon Aurora

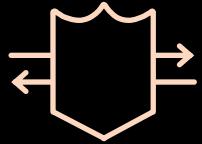
商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



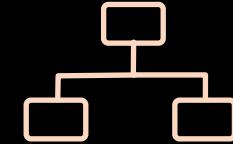
パフォーマンスと
拡張性



可用性と耐久性



高いセキュリティ



フル・マネージド

Amazon Aurora

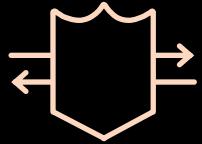
商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



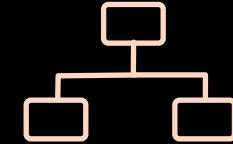
パフォーマンスと
拡張性



可用性と耐久性



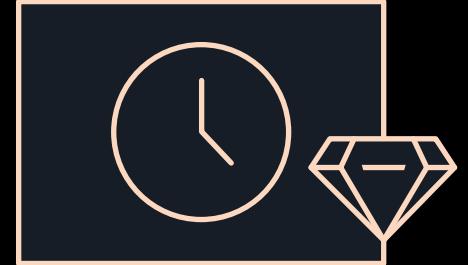
高いセキュリティ



フル・マネージド

Amazon Aurora

商用データベースの 1/10 のコストで MySQL と PostgreSQL と互換性があり、高いパフォーマンスと可用性を実現するために設計されたデータベース



可用性と耐久性

- マルチ AZ 構成を用いて 99.99% の可用性
- データは 3 AZ をまたがって保存 (課金は 1 コピー分のみ)
- 自動的な継続的な増分バックアップの取得
- ポイントインタイムリカバリ (PITR) 可能
- 通常 10 秒以下のフェイルオーバー
- フォルト・トレラントで自己修復ができ、自動拡張可能なストレージ
- ディザスタリカバリのための Global Database 機能

データベースの可用性



"Everything fails, all the time."

全てのものはいつでも壊れうる

Werner Vogels

VP and CTO, Amazon.com



ed the Turing test

すべては壊れるものとして考える

“

障害は発生するものであり、最終的にはすべてが時間の経過とともにフェイルオーバーします。

つまり、ルーターからハードディスクまで、TCP パケットを破壊するオペレーティングシステムからメモリユニットまで、そして一時的なエラーから永続的な障害まで、どれもが対象となるのです。

これは、最高品質のハードウェアを使用しているか、最低料金のコンポーネントを使用しているかにかかわらず、当たり前のことです。

- Werner Vogels, CTO, Amazon.com -

すべては壊れるものとして考える

“

障害は発生するものであり、最終的にはすべてが時間の経過とともにフェイルオーバーします。

つまり、ルーターからハードディスクまで、TCP パケットを破壊するオペレーティングシステムからメモリユニットまで、そして一時的なエラーから永続的な障害まで、どれもが対象となるのです。

これは、最高品質のハードウェアを使用しているか、最低料金のコンポーネントを使用しているかにかかわらず、当たり前のことです。

- Werner Vogels, CTO, Amazon.com -

データベース障害の種類

ディスク障害:

データ破損:

データベースサーバー障害:

データセンター障害:

大規模障害:

データベース障害の種類

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

データベース障害の種類

ディスク障害: ストレージデバイスの物理的または論理的な故障

- **原因:** 物理的損傷、媒体劣化、コントローラー故障、ファームウェアバグ、過熱、電源サージ

データ破損: データの整合性が損なわれる問題

- **原因:** 電源障害、ソフトウェアバグ、メモリエラー、不適切なシャットダウン

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

- **原因:** ハードウェア故障、メモリ不足、OS 障害、DB エンジンクラッシュ

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

- **原因:** 電源障害、冷却システム故障、ネットワーク切断、局所的自然災害

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

- **原因:** 大規模自然災害、広域インフラ障害、複数 AZ に影響する重大問題

データベースの可用性：RTO / RPO

RTO: Recovery Time Objective

- リカバリ時間目標
- 障害時に、どれだけの時間でリカバリを完了させるか

RPO: Recovery Point Objective

- リカバリポイント目標
- 障害時に、どの時点のデータまでリカバリ可能か

データベースの可用性：RTO / RPO

RTO (Recovery Time Objective) - 目標復旧時間

- ・ **定義:** システム障害やサービス停止から復旧までに許容される最大時間
- ・ **意味:** ダウンタイムの許容範囲を示す指標
- ・ **重要性:** ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ **例:** RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ **定義:** 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ **意味:** 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ **重要性:** データの重要性と更新頻度に基づいて設定される指標
- ・ **例:** RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容

データベースの可用性：RTO / RPO

RTO (Recovery Time Objective) - 目標復旧時間

- ・ 定義: システム障害やサービス停止から復旧までに許容される最大時間
- ・ 意味: ダウンタイムの許容範囲を示す指標
- ・ 重要性: ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ 例: **RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要**

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ 定義: 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ 意味: 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ 重要性: データの重要性と更新頻度に基づいて設定される指標
- ・ 例: **RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容**

データベースの可用性：RTO / RPO

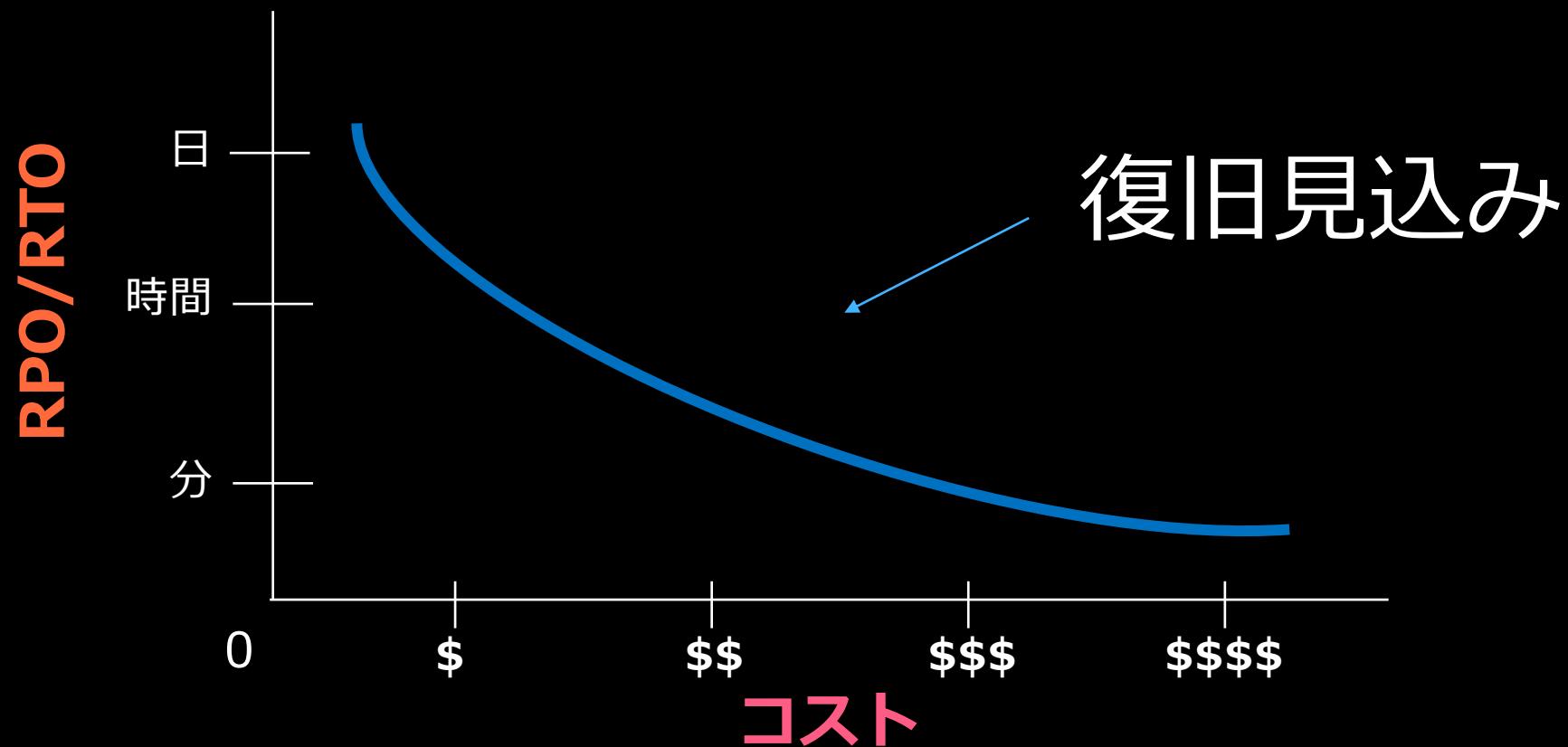
RTO (Recovery Time Objective) - 目標復旧時間

- ・ 定義: システム障害やサービス停止から復旧までに許容される最大時間
- ・ 意味: ダウンタイムの許容範囲を示す指標
- ・ 重要性: ビジネス継続性計画の基本となる指標、短いほど高度な復旧が必要
- ・ 例: **RTO = 4 時間の場合、システム障害発生から 4 時間以内に復旧する必要**

RPO (Recovery Point Objective) - 目標復旧ポイント

- ・ 定義: 障害発生時に許容されるデータ損失の最大量（時間で測定）
- ・ 意味: 最後のバックアップ時点から障害発生までの間に失われるデータ量
- ・ 重要性: データの重要性と更新頻度に基づいて設定される指標
- ・ 例: **RPO = 1 時間の場合、最大 1 時間分のデータ損失が許容**

RPO/RTO の目標とコスト



必要な可用性を定義し、システム要件に応じて構成を選択する事が重要

Amazon Aurora で実現する 高可用性



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

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

ディスク障害:ストレージデバイスの物理的または論理的な故障

Amazon Aurora

- ・ディスク障害が発生した場合、**Aurora ストレージ**の冗長化により継続稼働可能
- ・2つの破損までは稼働可能
- ・3つの破損が発生した場合は読み込みのみ可能

Aurora ストレージの高可用性

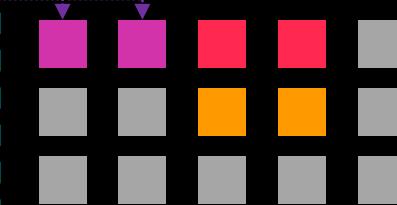
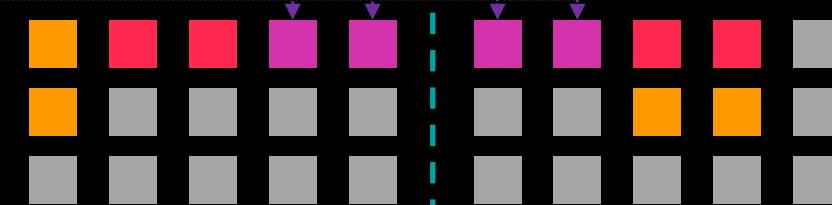
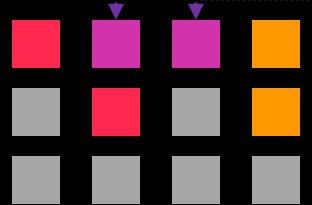
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

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

Aurora ストレージの高可用性

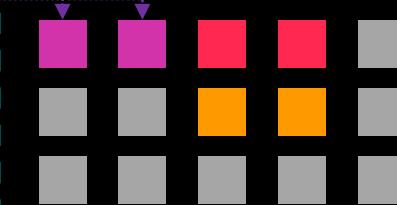
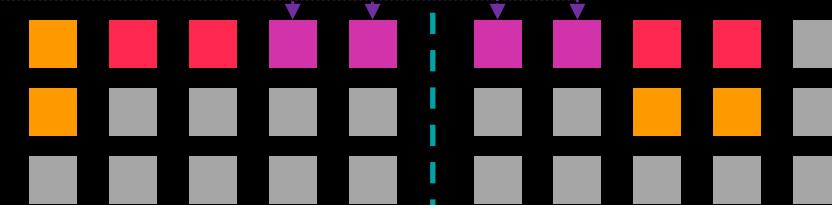
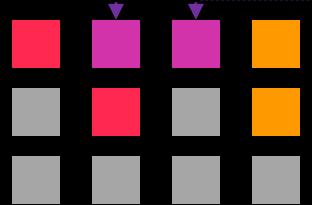
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

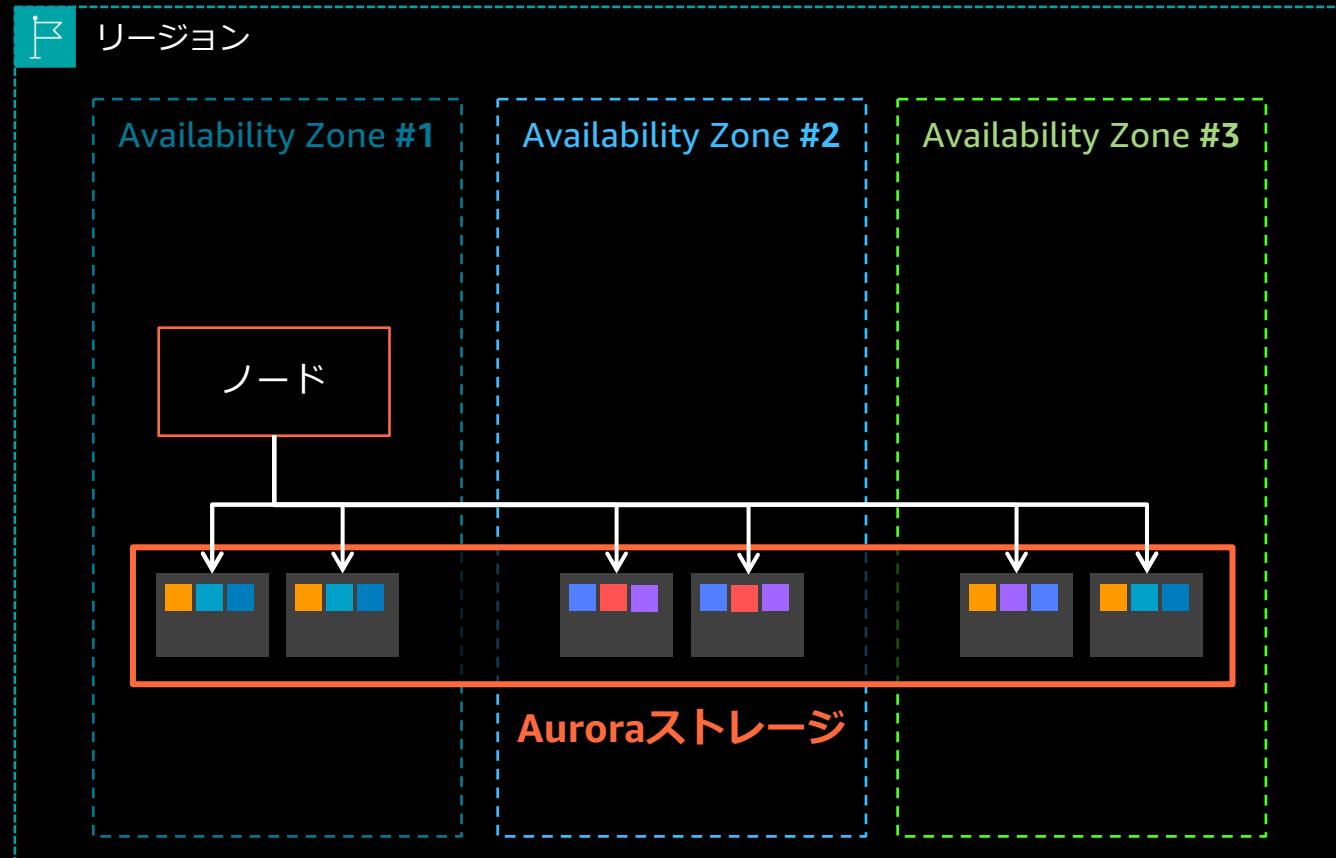
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html

ストレージ冗長化の違い (Amazon RDS / Amazon Aurora)

Amazon RDS



Amazon Aurora



Aurora ストレージの高可用性

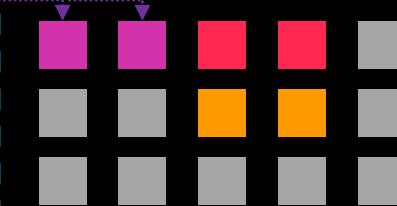
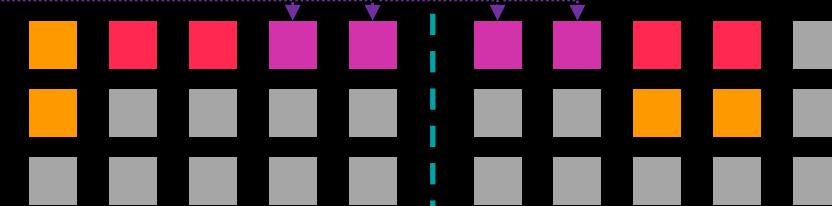
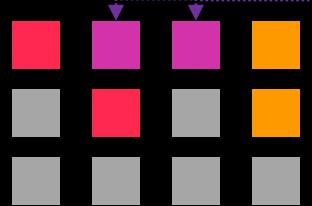
Availability
Zone 1

Availability
Zone 2

Availability
Zone 3



Shared distributed storage volume



データベース用に設計された
専用の **log-structured 分散ストレージシステム**

3 つの異なるアベイラビリティゾーンに分散された
数百のストレージノードにストライピングされた
ストレージボリューム

データは 10GB のプロテクショングループ の
単位で管理され、自動的に最大 128TB まで拡張

AZ+1 の障害から保護するためにデータを
各アベイラビリティーゾーンに 2 つのコピー、
リージョン内で計 6 つのコピー
(お客様の課金は 1 コピーのみ)

データは継続的に S3 へバックアップ

詳細 :

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

Aurora ストレージ：自動修復・耐障害性



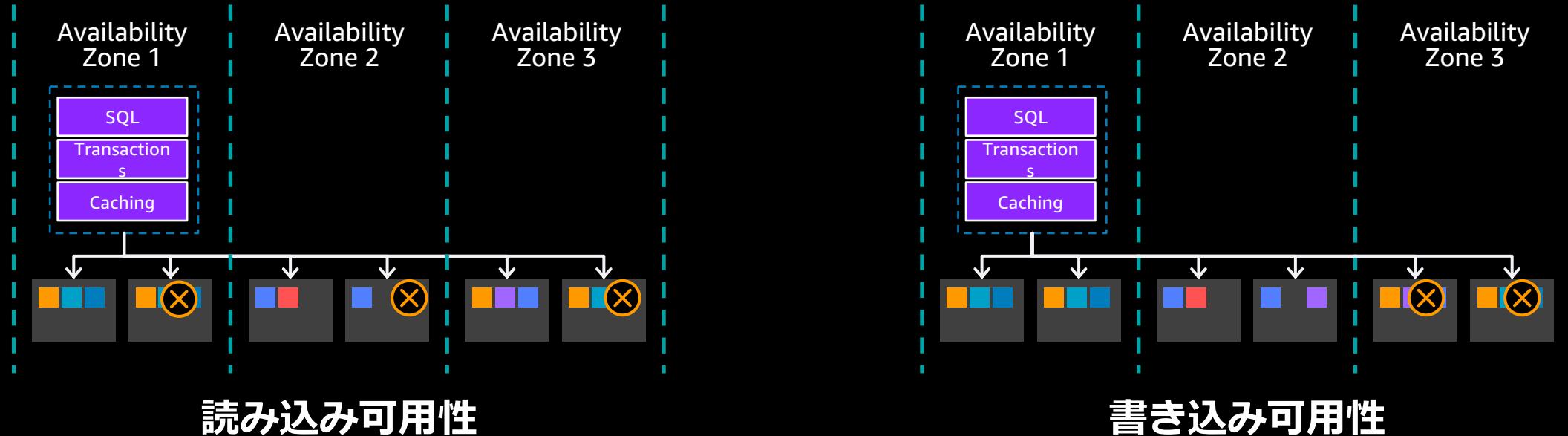
データは 6 つのノードすべてに非同期・並列で書き込み

読み込みには 3/6 クオーラム、書き込みには 4/6 ノードのクオーラムが必要

各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

Aurora ストレージ：自動修復・耐障害性



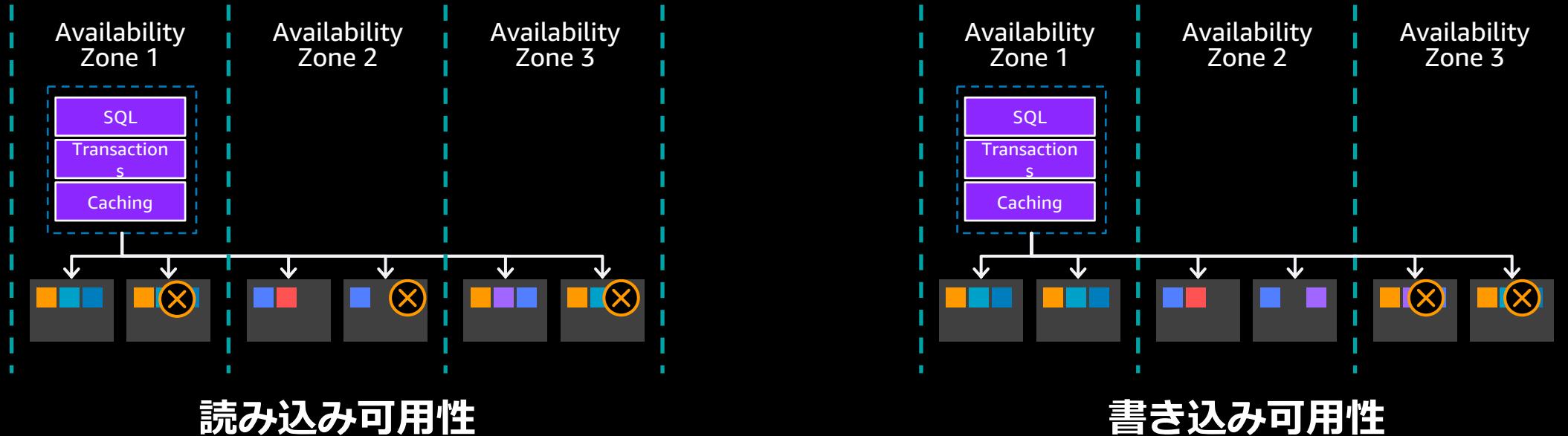
データは 6 つのノードすべてに非同期・並列で書き込み

読み込みには 3/6 クオーラム、書き込みには 4/6 ノードのクオーラムが必要

各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

Aurora ストレージ：自動修復・耐障害性



データは 6 つのノードすべてに非同期・並列で書き込み

読み込みには **3/6 クオーラム**、書き込みには **4/6 ノードのクオーラム**が必要

各ノード間のデータの欠損、破損は P2P のゴシッププロトコルで確認、修復される

詳細：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-under-the-hood-quorum-and-correlated-failure/>

参考：クオーラムとは

- ・ 定義：集団における意思決定や**合意形成に必要な最小人数または最小比率**
- ・ 応用：分散コンピューティングシステムにおける合意形成アルゴリズム

分散システムでのクオーラムとは

- ・ 複数ノードで構成されるシステムにおいて、**有効な操作に必要な最小ノード数**
- ・ データの一貫性と可用性を確保するための合意形成メカニズム
- ・ 通常は全ノード数の過半数 ($N/2+1$) が基本的なクオーラム値として設定される

参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

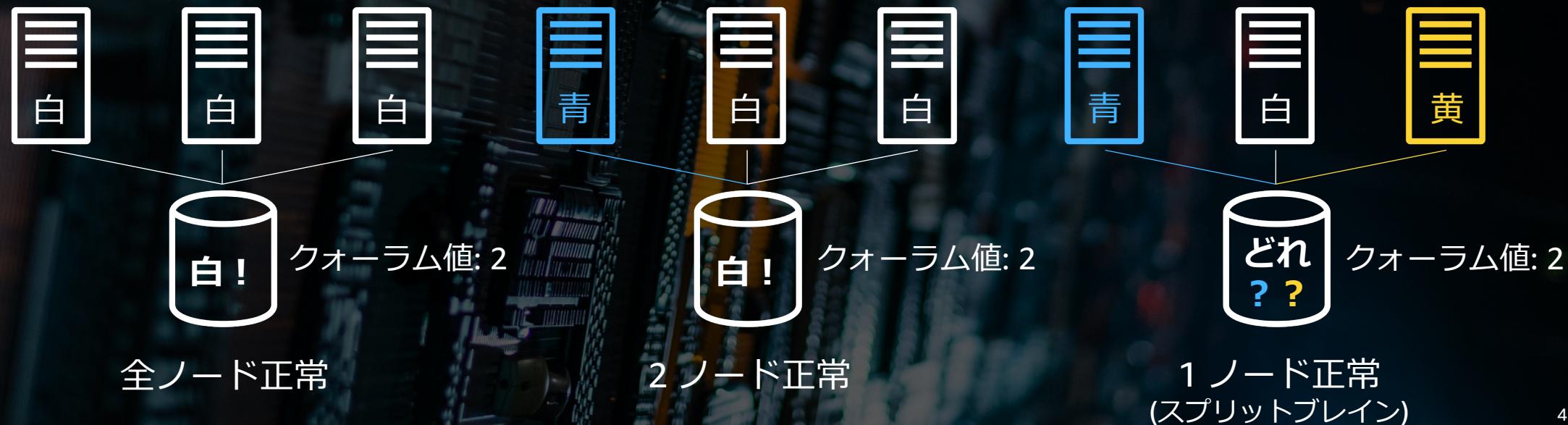
- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



参考：クオーラムとは

クオーラムの目的

- ・ システム全体の耐障害性を高める
- ・ 一部のノードが故障しても、システム全体は機能し続ける
- ・ データの整合性を維持しながら、高い可用性を実現する



クオーラム値: 3

全ノード正常



クオーラム値: 3

3ノード正常



クオーラム値: 3

1ノード正常
(スプリットブレイン)

クオーラムモデル

レプリケーション管理のためのクオーラムのルール

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

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

クオーラムモデル

レプリケーション管理のためのクオーラムのルール

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

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

Amazon Aurora の
クオーラムモデル

Aurora クオーラムモデル例



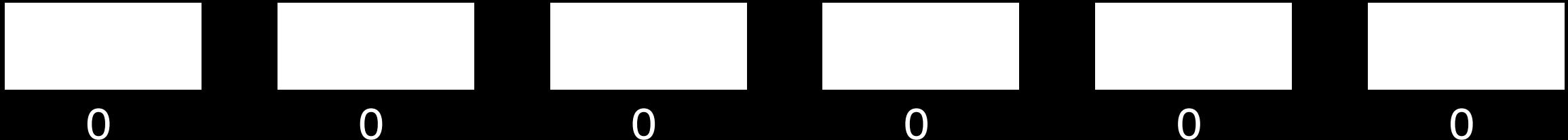
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$

コピー

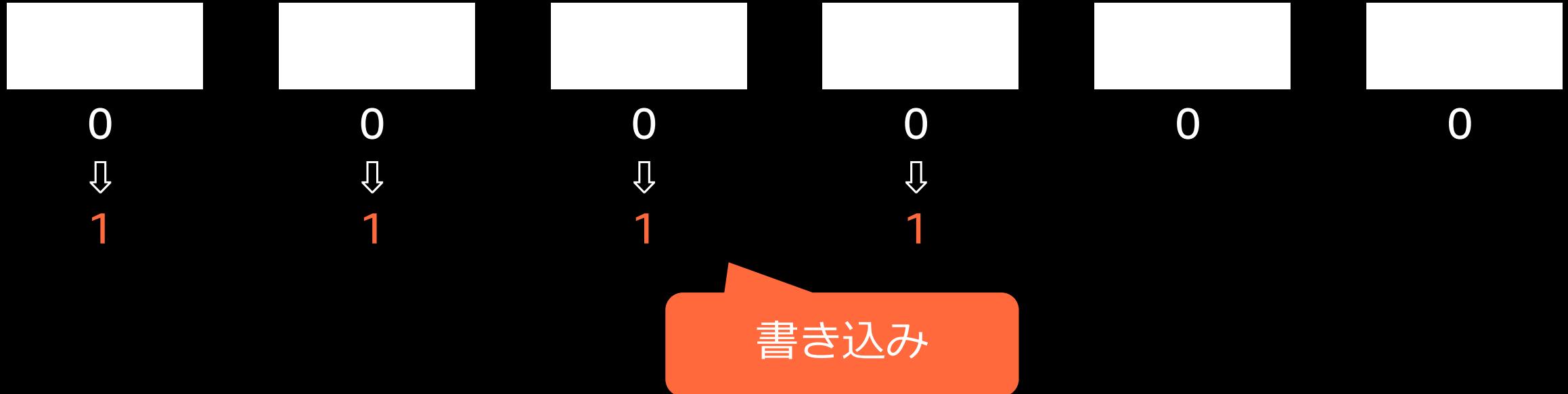
書き込み

読み込み



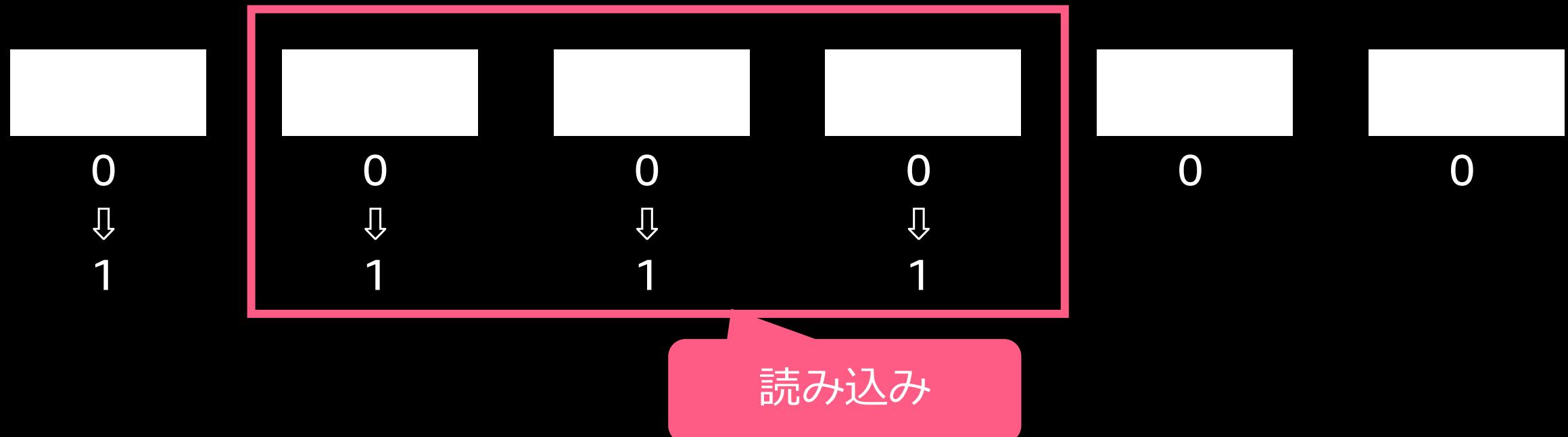
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



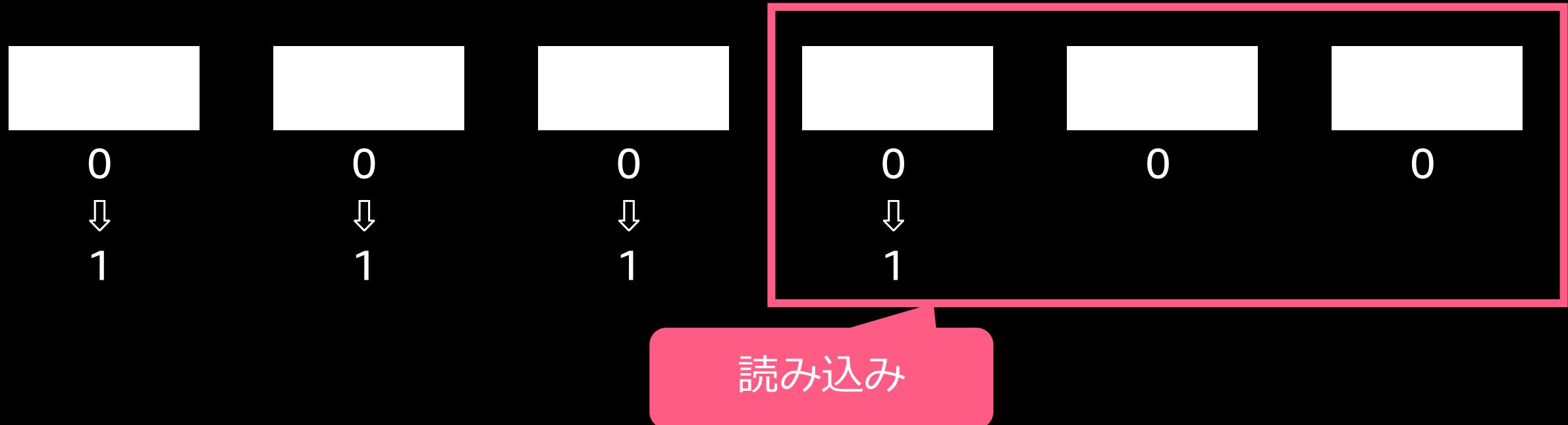
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



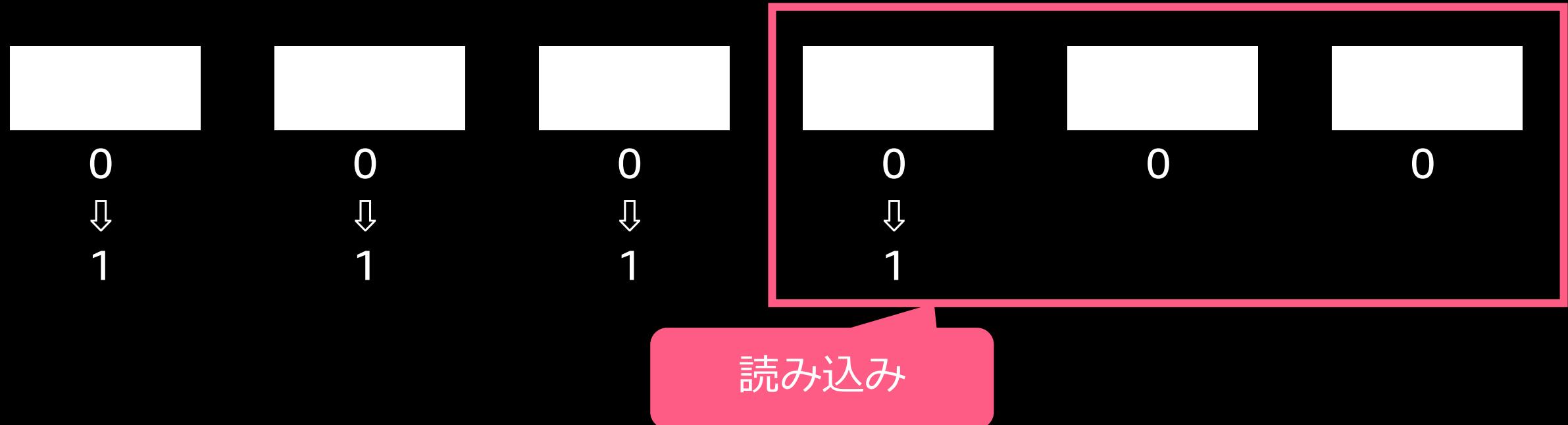
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$

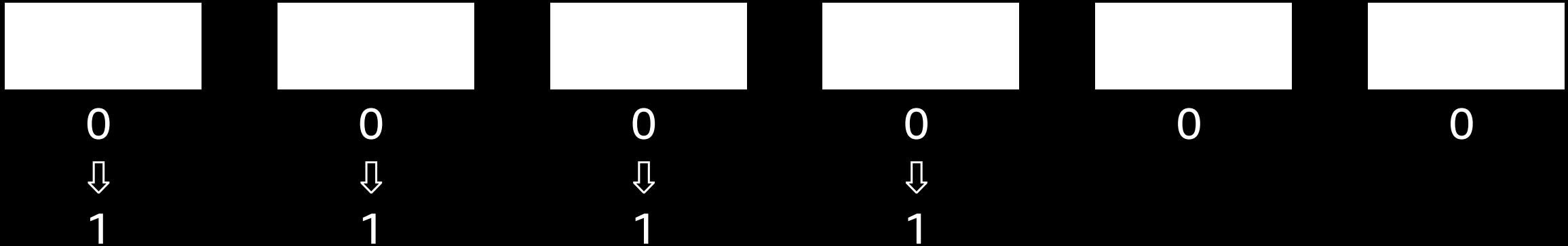


必ず最新のデータを読み取り可能

- $V_r + V_w > V$
- $V_w > V/2$

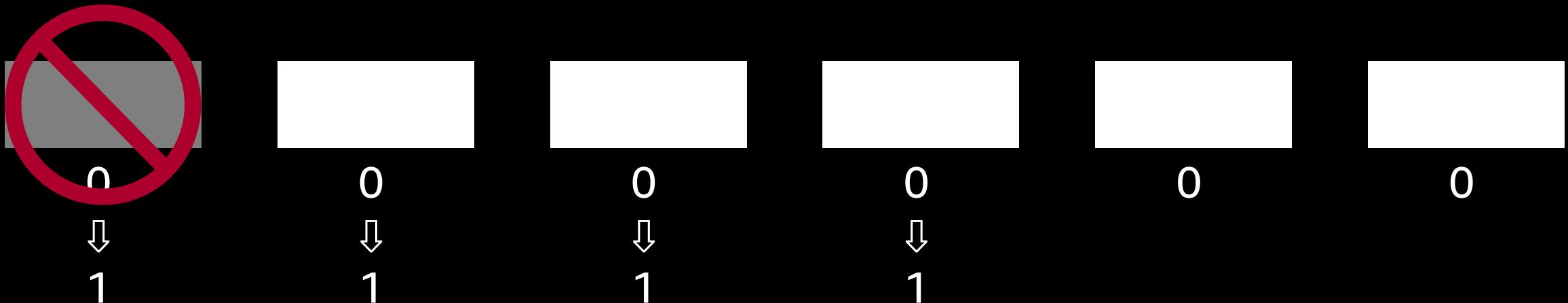
耐障害性 - クオーラムモデル

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



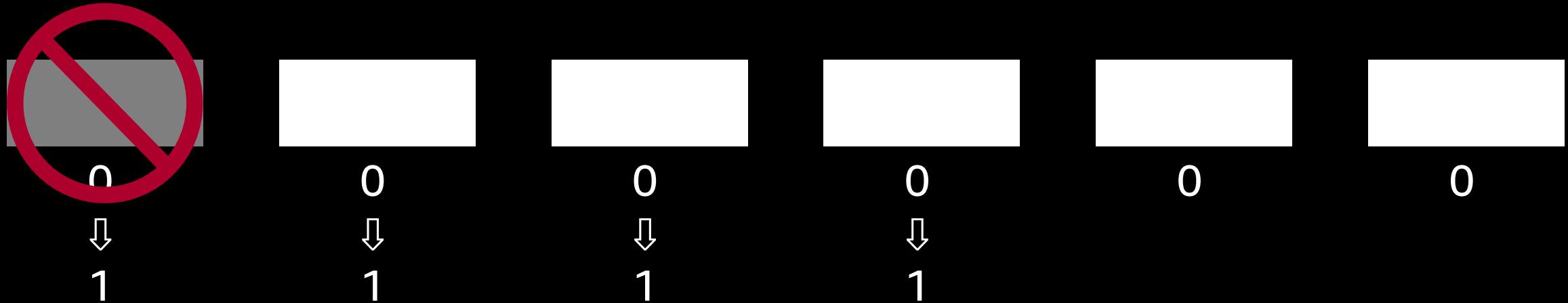
- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



- $V_r + V_w > V$
- $V_w > V/2$

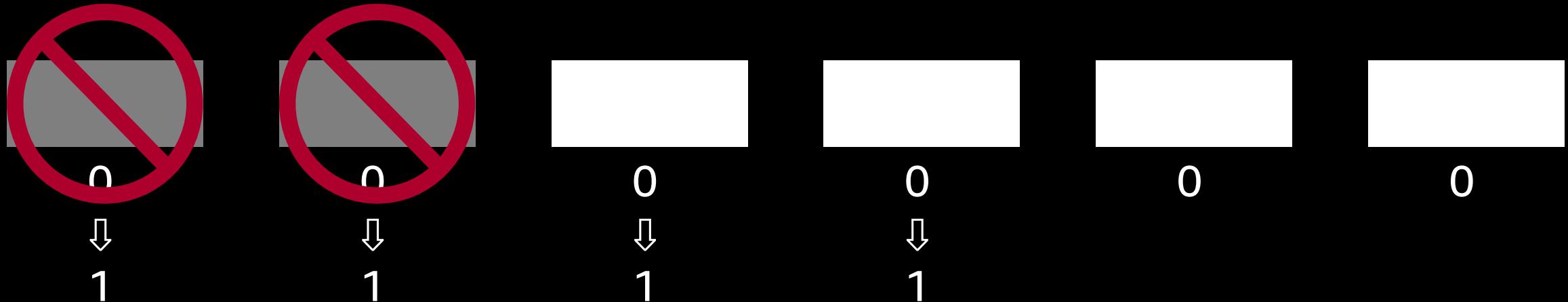
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



障害が起きても処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

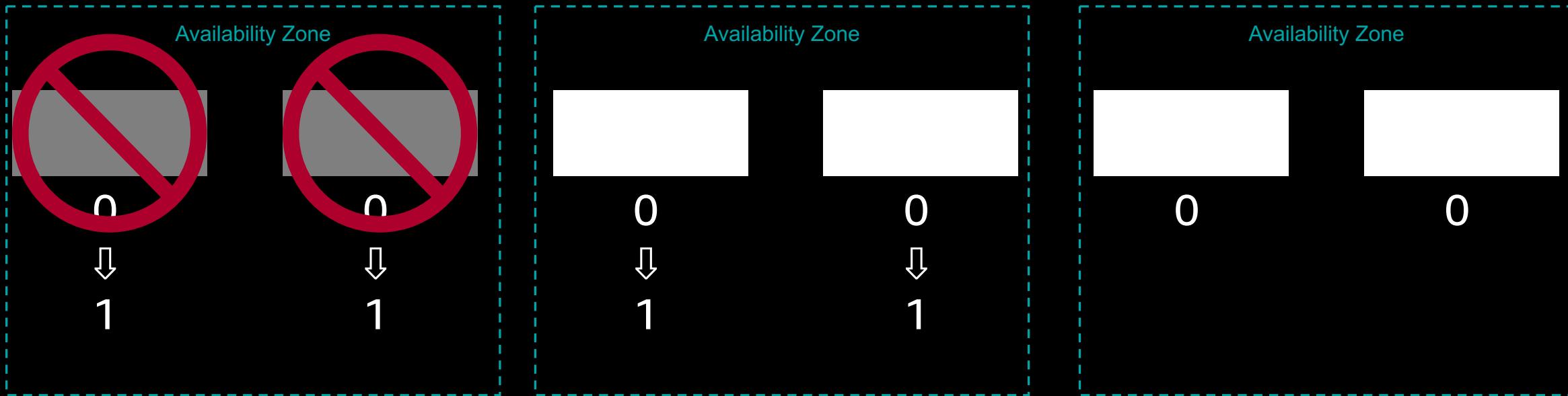
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



2重障害が起きても処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

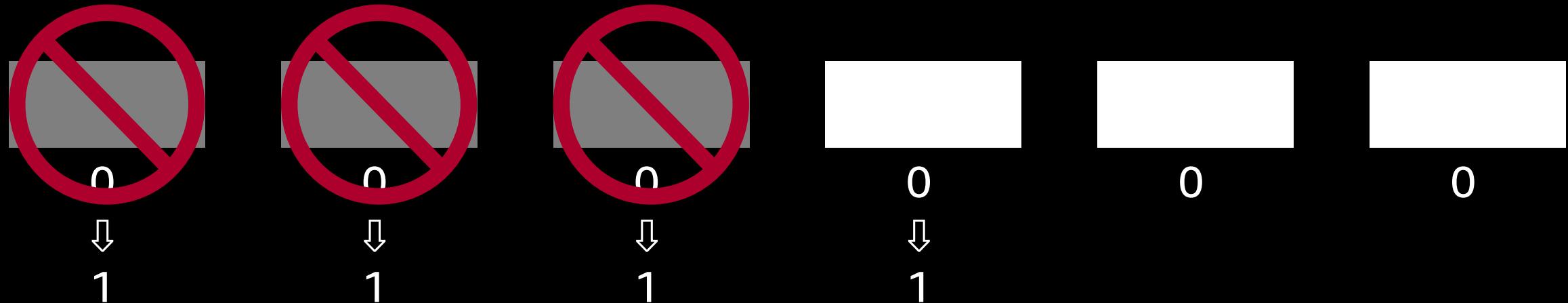
例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



AZ 障害時も処理継続可能

- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



3重障害が起きても読み込み可能

$V_r=3$

- $V_r + V_w > V$
- $V_w > V/2$

例) 4/6 クオーラム : $V=6$, $V_w=4$, $V_r=3$



AZ + 1 障害時も読み込み可能

- $V_r + V_w > V$
- $V_w > V/2$

Aurora では AZ 障害 + 1 の障害 を許容

1 AZ あたり 2つ、3 AZ に 6 つのコピー

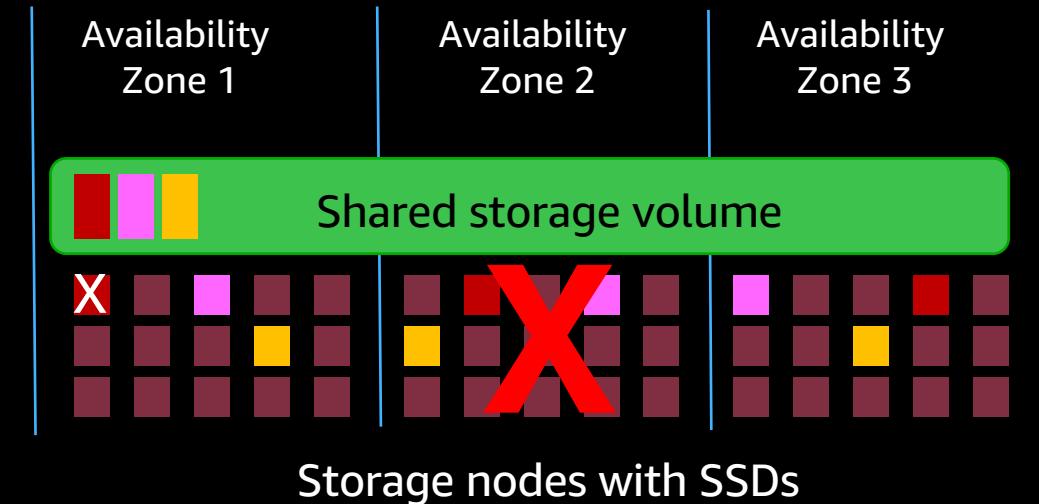
- 書き込みは 4/6 クオーラム

AZ 障害時

- 4/6 のコピーは維持
- 書き込みも可能

AZ 障害 + 1 の障害 (二重障害) 時

- 3/6 のコピーは維持
- データの損失は無
- 故障箇所は 3 つのコピーから再構築
- 再構築により書き込み可能に

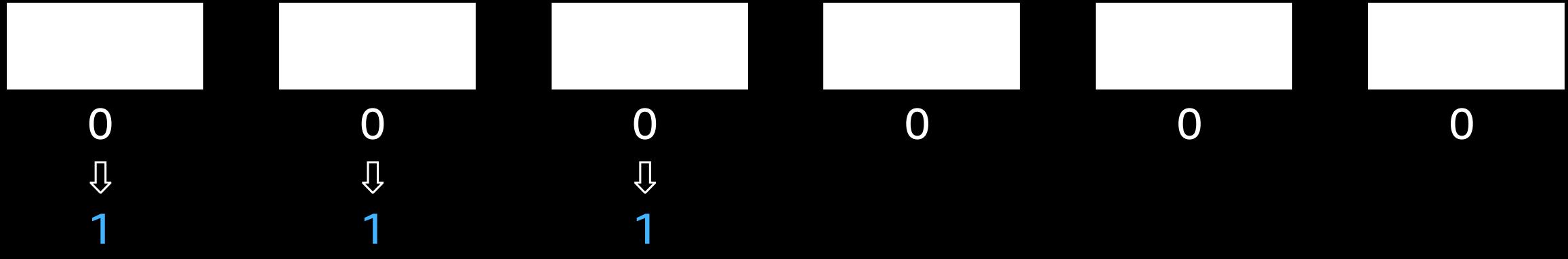


Question:

4/6 クオーラムも必要？

3/6 ($V_w=3$) でも良いのでは？？

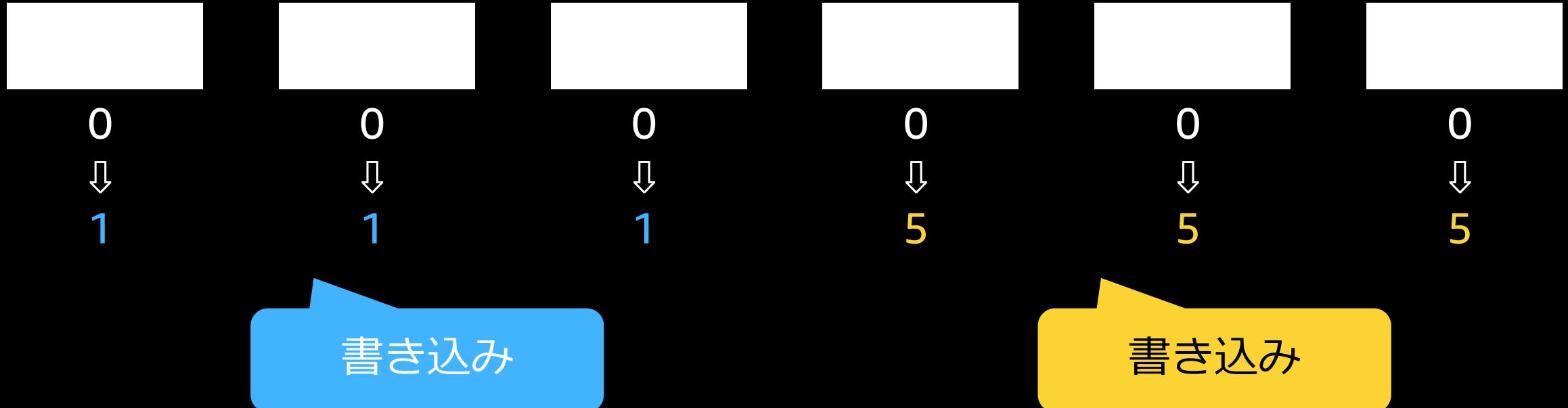
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



書き込み

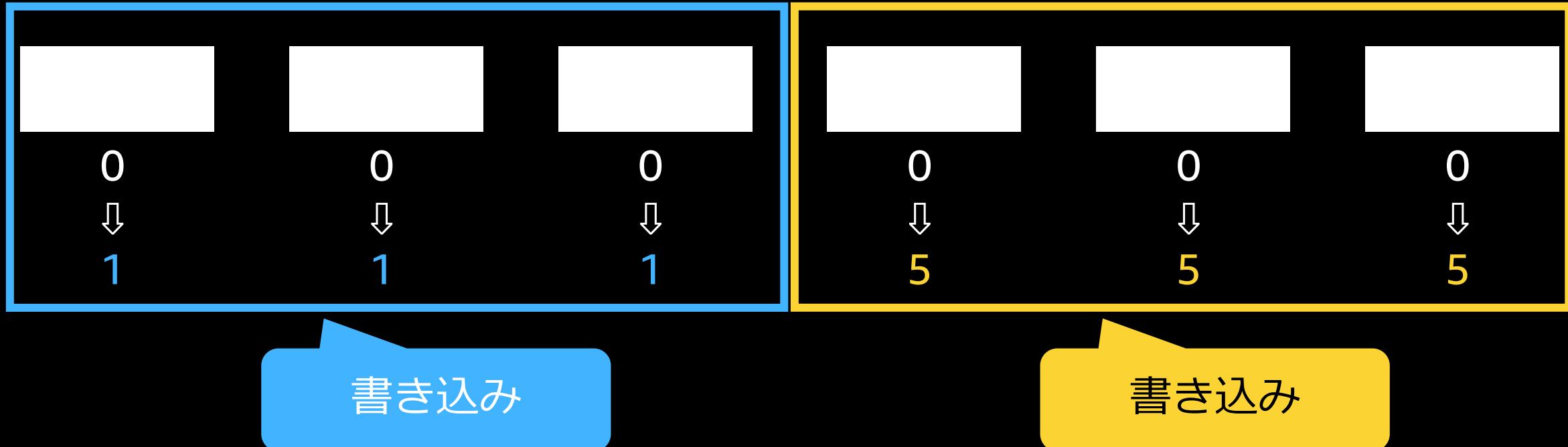
- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



- $V_r + V_w = V$
- $V_w = V/2$

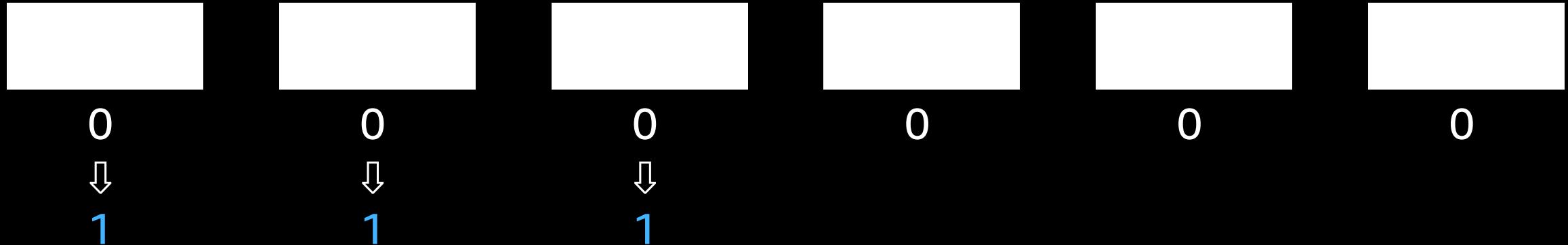
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



トランザクションが並列で書き込まれる可能性がある

- $V_r + V_w = V$
- $V_w = V/2$

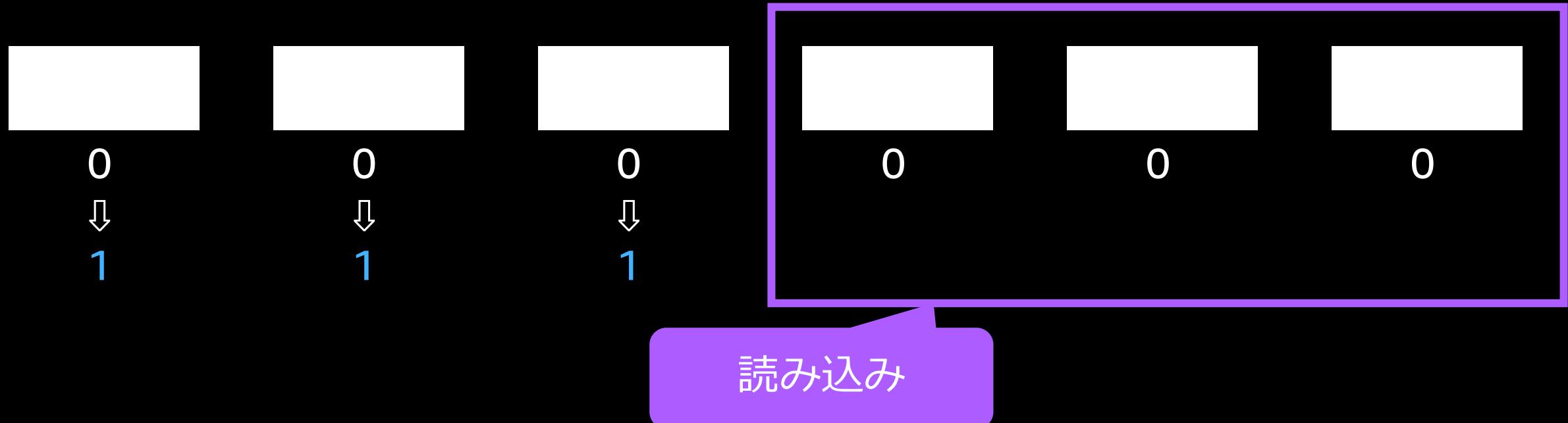
もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



書き込み

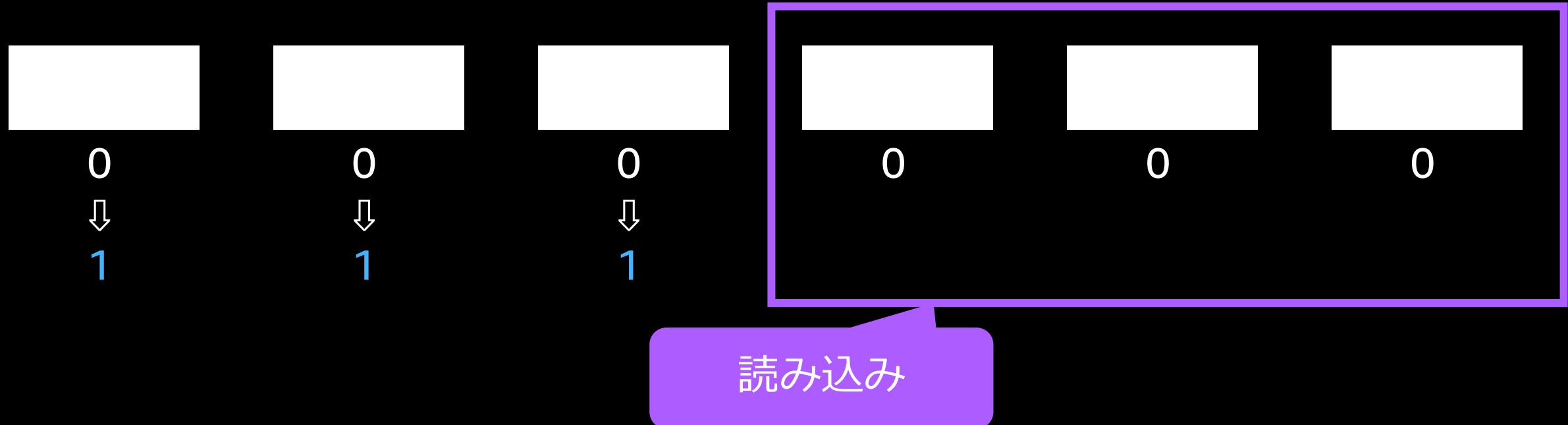
- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



- $V_r + V_w = V$
- $V_w = V/2$

もし $V_w=3$ だと : $V=6$, $V_w=3$, $V_r=3$



必ず最新のデータを読み取れない可能性がある

- $V_r + V_w = V$
- $V_w = V/2$

再掲：クオーラムモデル

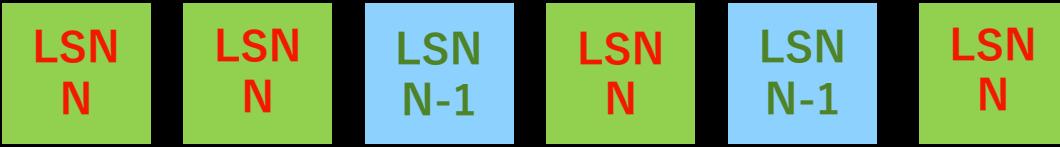
レプリケーション管理のためのクオーラムのルール

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

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

**Amazon Aurora の
クオーラムモデル**

参考：クオーラムにおける読み込みを回避



各データブロックに対して、
クオ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

ほとんどのクオーラムベースの
システムでは**読み込みにコスト**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

読み込みクオーラムは
修理やクラッシュリカバリに必要

LSN : Log Sequence Number
データベースの変更を追跡するための一意の識別子

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

データ破損: データの整合性が損なわれる問題

Amazon Aurora

- ユーザーエラーなどによりデータが論理的に破損した場合
バックアップからのリストア・リカバリにより対応可能
- ただしシステム停止が必要になる

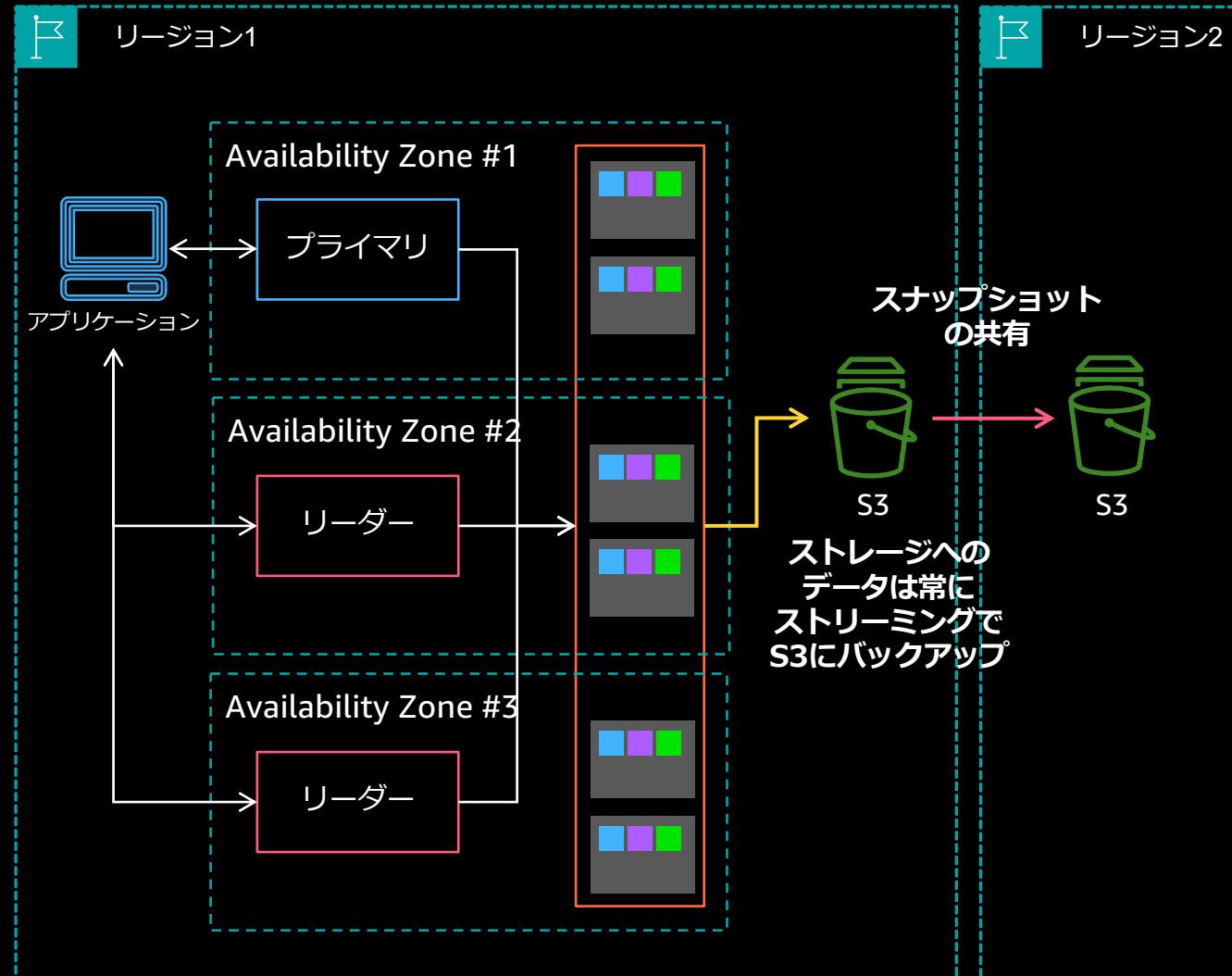
Amazon Aurora の可用性

データ破損: データの整合性が損なわれる問題

Amazon Aurora

- ユーザーエラーなどによりデータが論理的に破損した場合
バックアップからのリストア・リカバリにより対応可能
- ただしシステム停止が必要になる

Aurora のバックアップ & リカバリー



Aurora のバックアップ

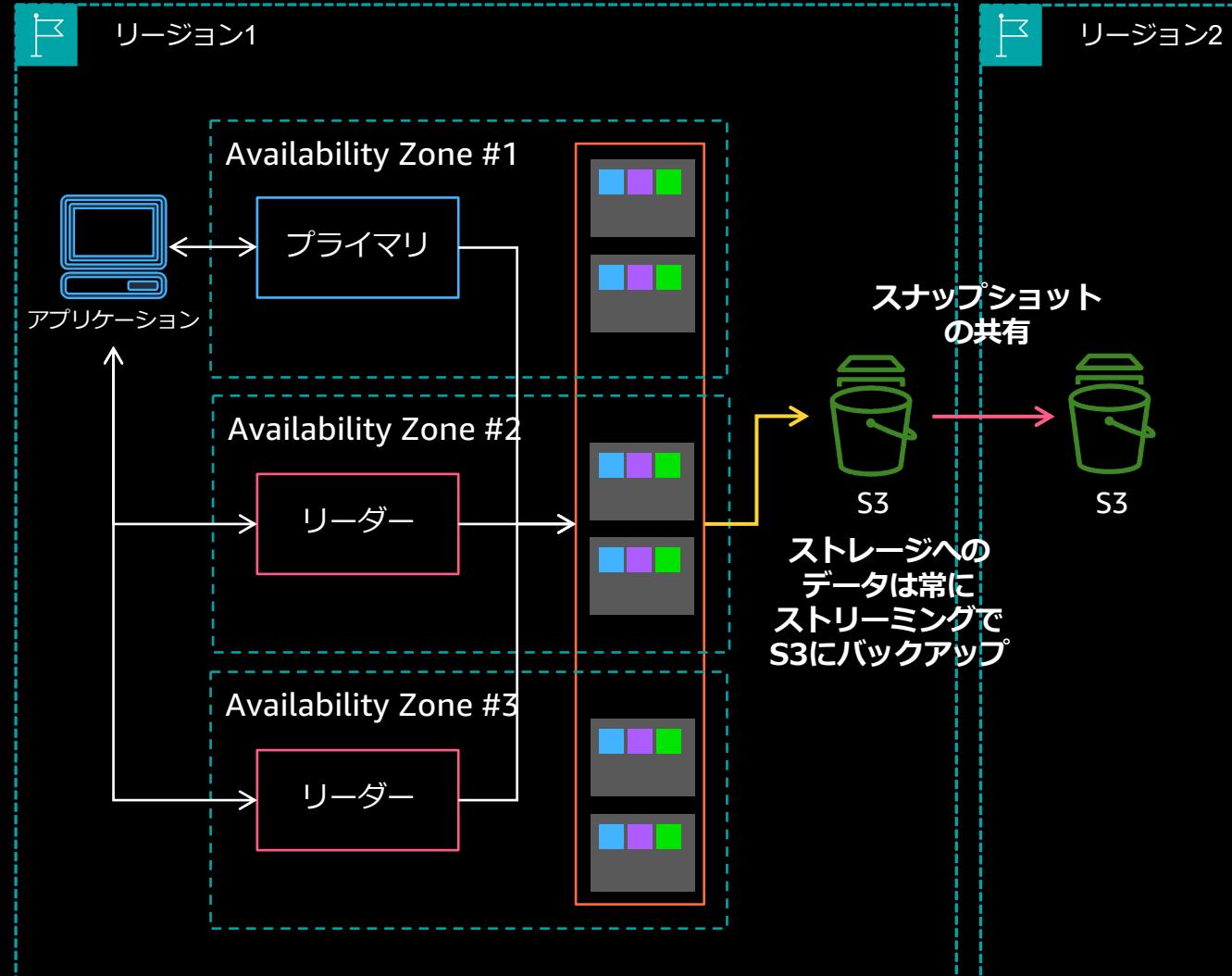
1. 自動バックアップ(必須)
2. 手動スナップショット(任意)

仕組み

- 繼続的にバックアップを S3 に保存
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の任意の時点にリカバリすることが可能 (ポイントインタイムリカバリ : PITR)
- バックアップはストリーミングで実行されバックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを別リージョンにコピーしたり別アカウントと共有ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

Aurora のバックアップ & リカバリー



Aurora のバックアップ

1. 自動バックアップ(必須)
2. 手動スナップショット(任意)

仕組み

- 繼続的に**バックアップを S3 に保存**
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の**任意の時点**に**リカバリ**することが可能 (ポイントインタイムリカバリ : PITR)
- バックアップは**ストリーミングで実行**され、バックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを**別リージョン**に**コピー**したり**別アカウント**と**共有**ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

バックアップの制限事項

自動バックアップ

- 保持期間は 1 日から最大 35 日間まで
- バックアップウィンドウ中に他のオペレーションを実行すると競合する可能性あり
- 削除された DB クラスターの自動バックアップは保持期間終了後に完全に削除される
- 最新の復元ポイントは現在時刻から最短 5 分前まで

スナップショット

- 手動スナップショットは最大 20 の AWS アカウントまで共有可能
- 暗号化されたスナップショットは、デフォルトの AWS KMS キーで暗号化されている場合は共有不可
- 暗号化されたスナップショットを共有する場合、共有先アカウントには KMS キーへのアクセス権が必要

ポイントインタイムリカバリ (PITR)

- 復元時は常に新しい DB クラスターが作成され、元のクラスターは変更されない
- 復元された DB クラスターはデフォルトのパラメータグループに関連付けられる
- カスタムパラメータグループを適用するには、復元時に明示的に指定する必要がある

詳細はドキュメントを参照 : https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-storage-backup.html

Aurora のバックアップコスト

バックアップストレージ

- ・ **無料枠:** DBクラスター サイズの100%までのバックアップストレージは無料
- ・ **自動バックアップ:** 保持期間内のバックアップは無料
- ・ **スナップショット:** 保持期間内のスナップショットは無料
- ・ **追加バックアップ:** 無料枠を超えるバックアップおよび保持期間外のスナップショットは、
GB-月あたり約\$0.021～\$0.023（リージョンにより異なる）
- ・ **削除されたクラスター:** 削除されたクラスターのスナップショットも課金対象

データ転送

- ・ **同一 AZ 内:** Aurora と EC2間のデータ転送は無料
- ・ **AZ 間レプリケーション:** DBクラスター レプリケーションのためのAZ間データ転送は無料
- ・ **リージョン間転送:** GB当たり\$0.01～\$0.02（送信元/送信先リージョンにより異なる）

復元操作

- ・ バックアップからの復元自体に追加料金はなし
- ・ 復元されたDBクラスターは新規インスタンスとして課金開始
- ・ クロスリージョン復元時はデータ転送料金が発生

As of May 2025

詳細はドキュメントを参照：<https://aws.amazon.com/jp/rds/aurora/pricing/>



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

Aurora のバックアップコスト

バックアップストレージ

- ・ **無料枠:** DBクラスター サイズの100%までのバックアップストレージは無料
- ・ **自動バックアップ:** 保持期間内のバックアップは無料
- ・ **スナップショット:** 保持期間内のスナップショットは無料
- ・ **追加バックアップ:** 無料枠を超えるバックアップおよび保持期間外のスナップショットは、
GB-月あたり約\$0.021～\$0.023（リージョンにより異なる）
- ・ **削除されたクラスター:** 削除されたクラスターのスナップショットも課金対象

データ転送

- ・ **同一 AZ 内:** Aurora と EC2間のデータ転送は無料
- ・ **AZ 間レプリケーション:** DBクラスター レプリケーションのためのAZ間データ転送は無料
- ・ **リージョン間転送:** GB当たり\$0.01～\$0.02（送信元/送信先リージョンにより異なる）

復元操作

- ・ バックアップからの復元自体に追加料金はなし
- ・ 復元されたDBクラスターは新規インスタンスとして課金開始
- ・ クロスリージョン復元時はデータ転送料金が発生

As of May 2025

詳細はドキュメントを参照 : <https://aws.amazon.com/jp/rds/aurora/pricing/>

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

Amazon Aurora

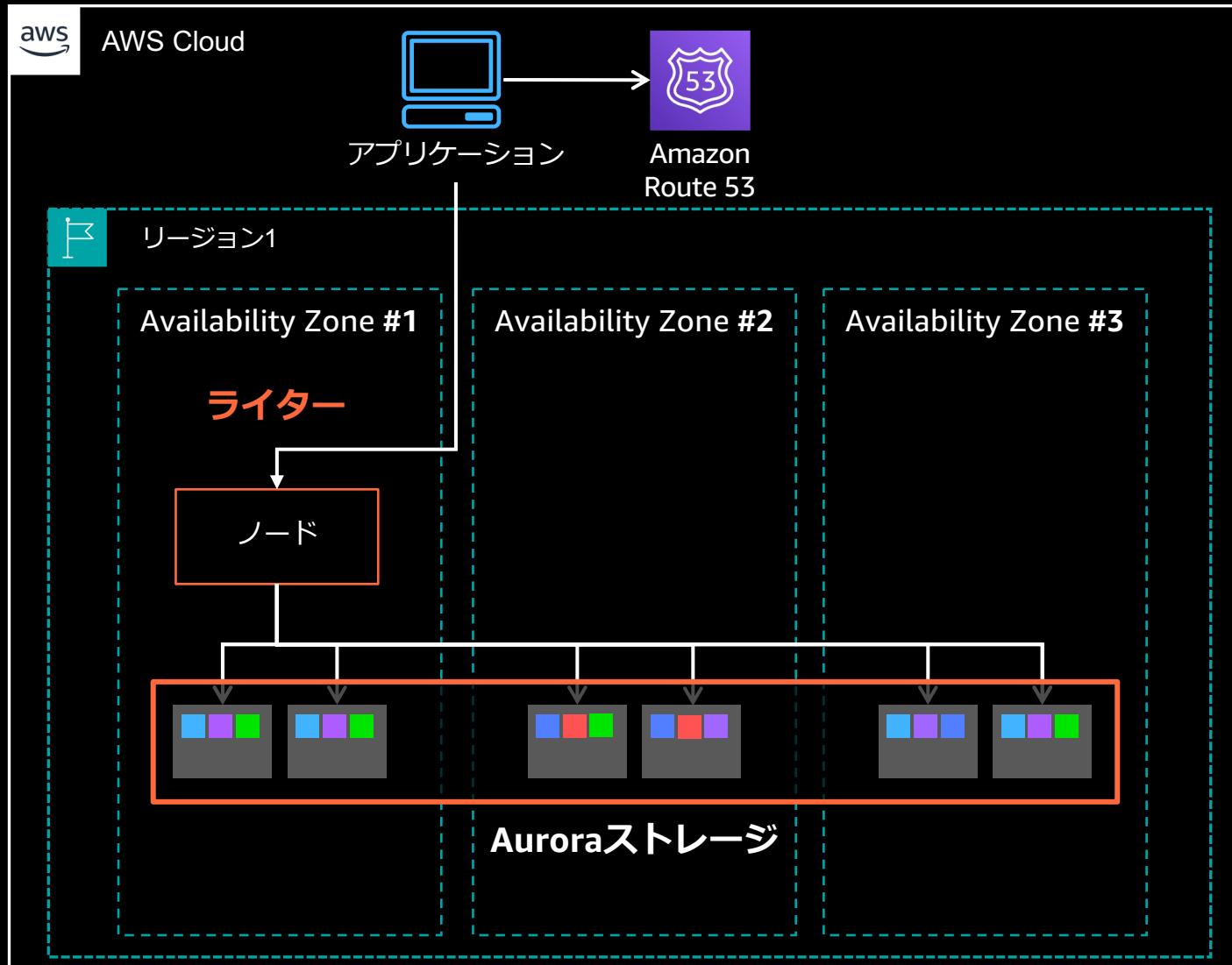
- シングル AZ 構成の場合

インスタンス再作成が必要な為、通常 10 分未満の停止時間が発生

- マルチ AZ 構成を使用した高可用性構成の場合

フェイルオーバーにより、通常 1 分未満でデータベース処理再開可能

シングル AZ 構成



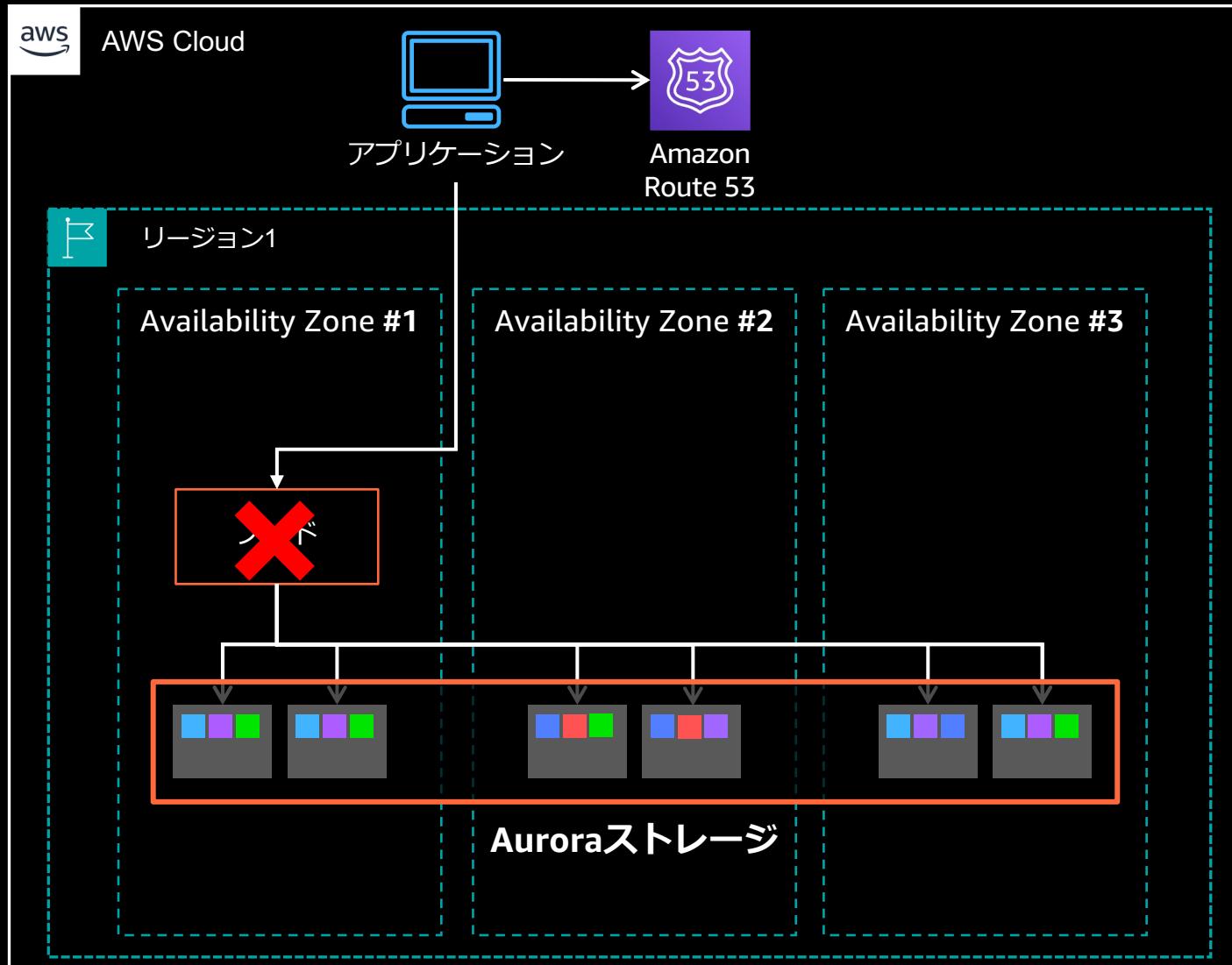
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：別 AZ に手動インスタンス作成が必要

シングル AZ 構成



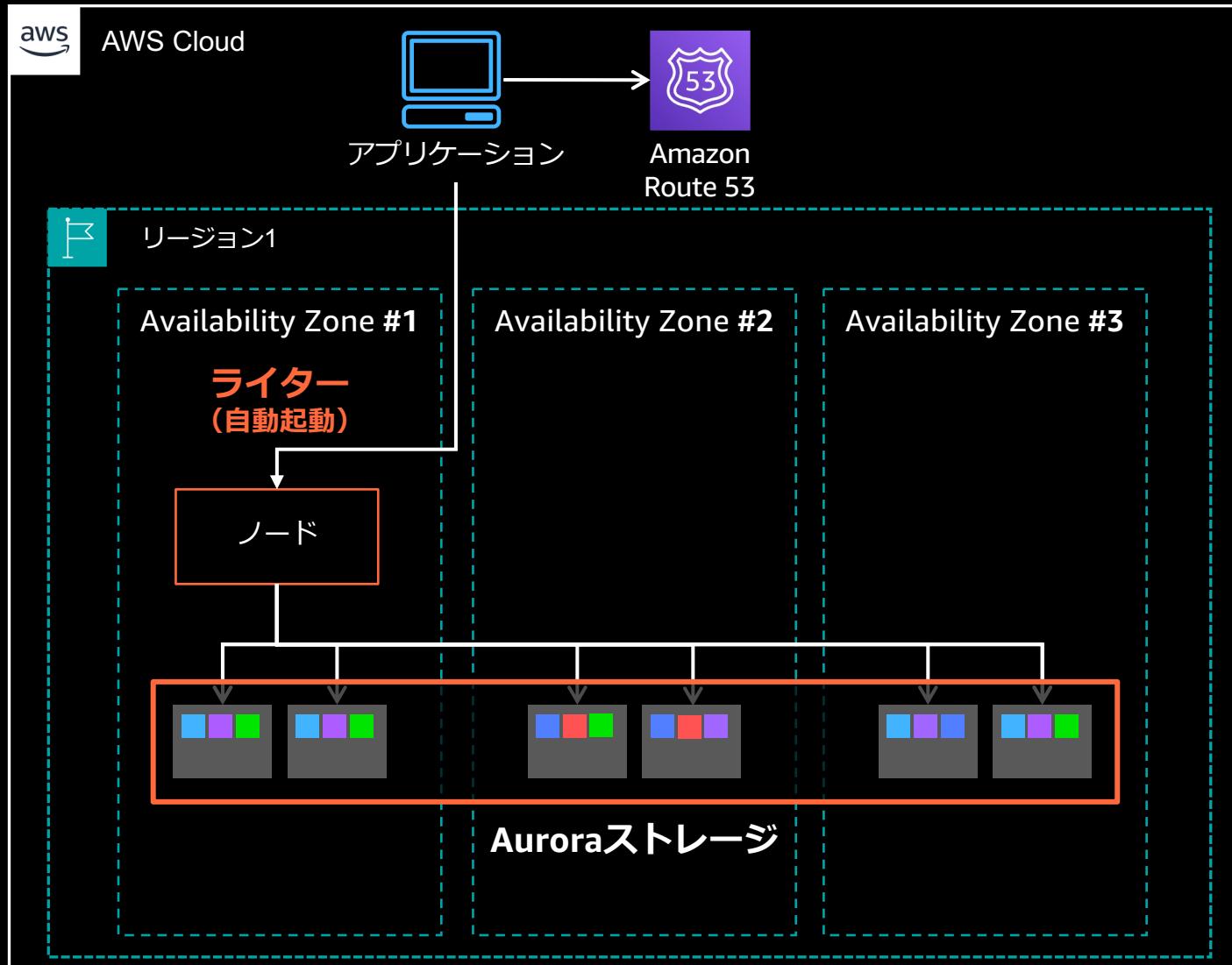
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- **インスタンス障害時 :** 同一 AZ に自動再作成 (10分未満)
- **AZ 全体障害時 :** 別 AZ に手動インスタンス作成が必要

シングル AZ 構成



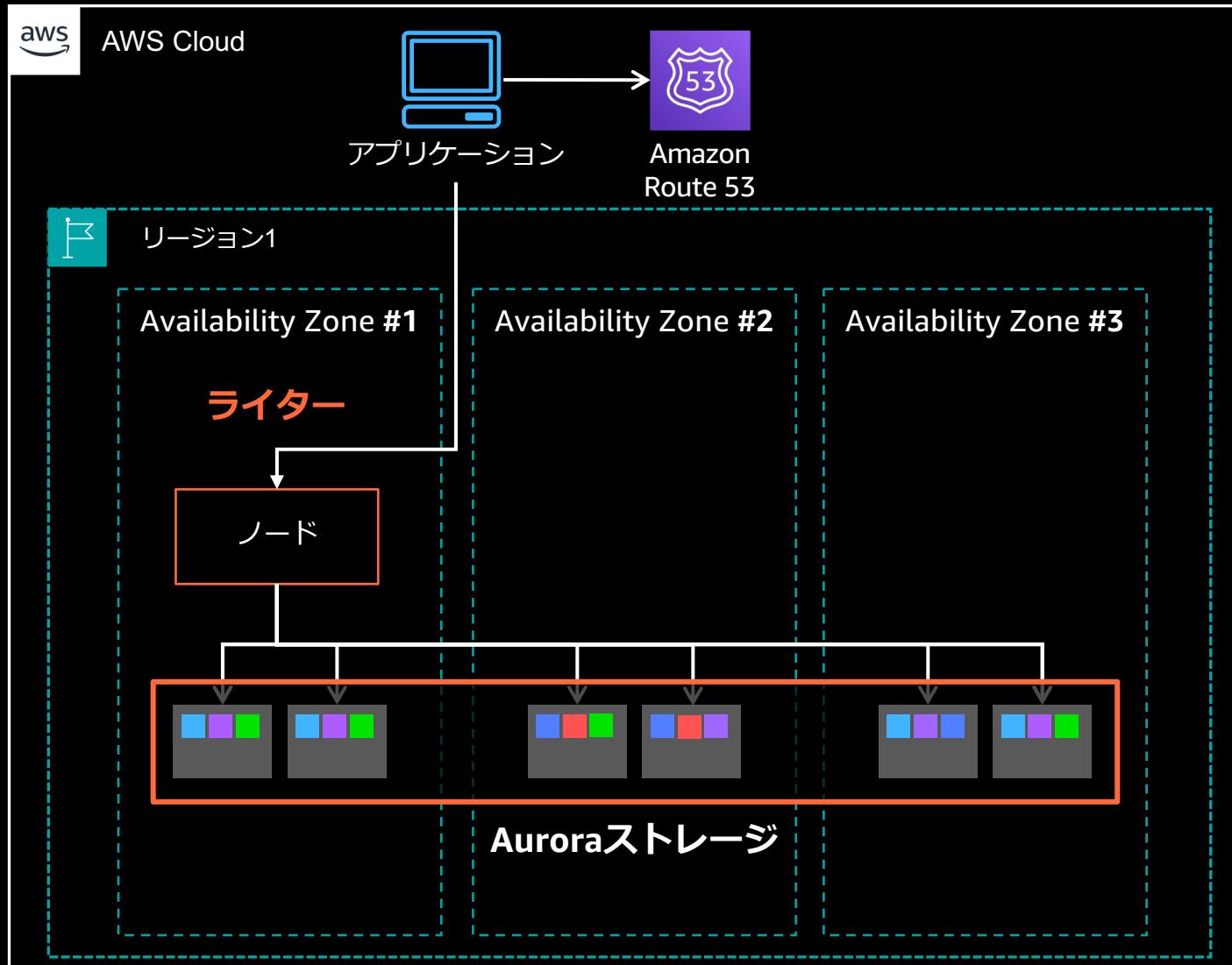
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：
同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：
別 AZ に手動インスタンス作成が必要

シングル AZ 構成



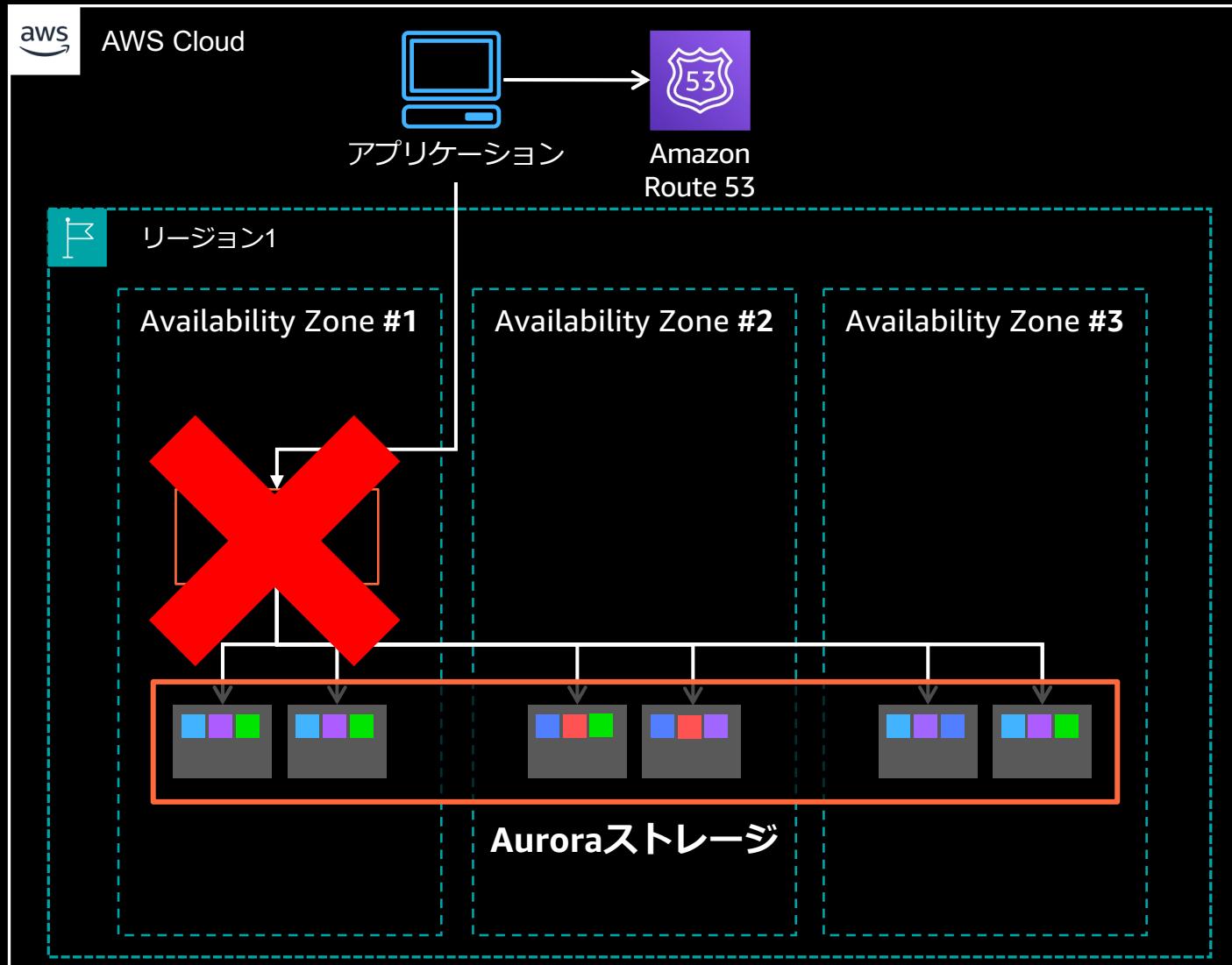
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：別 AZ に手動インスタンス作成が必要

シングル AZ 構成



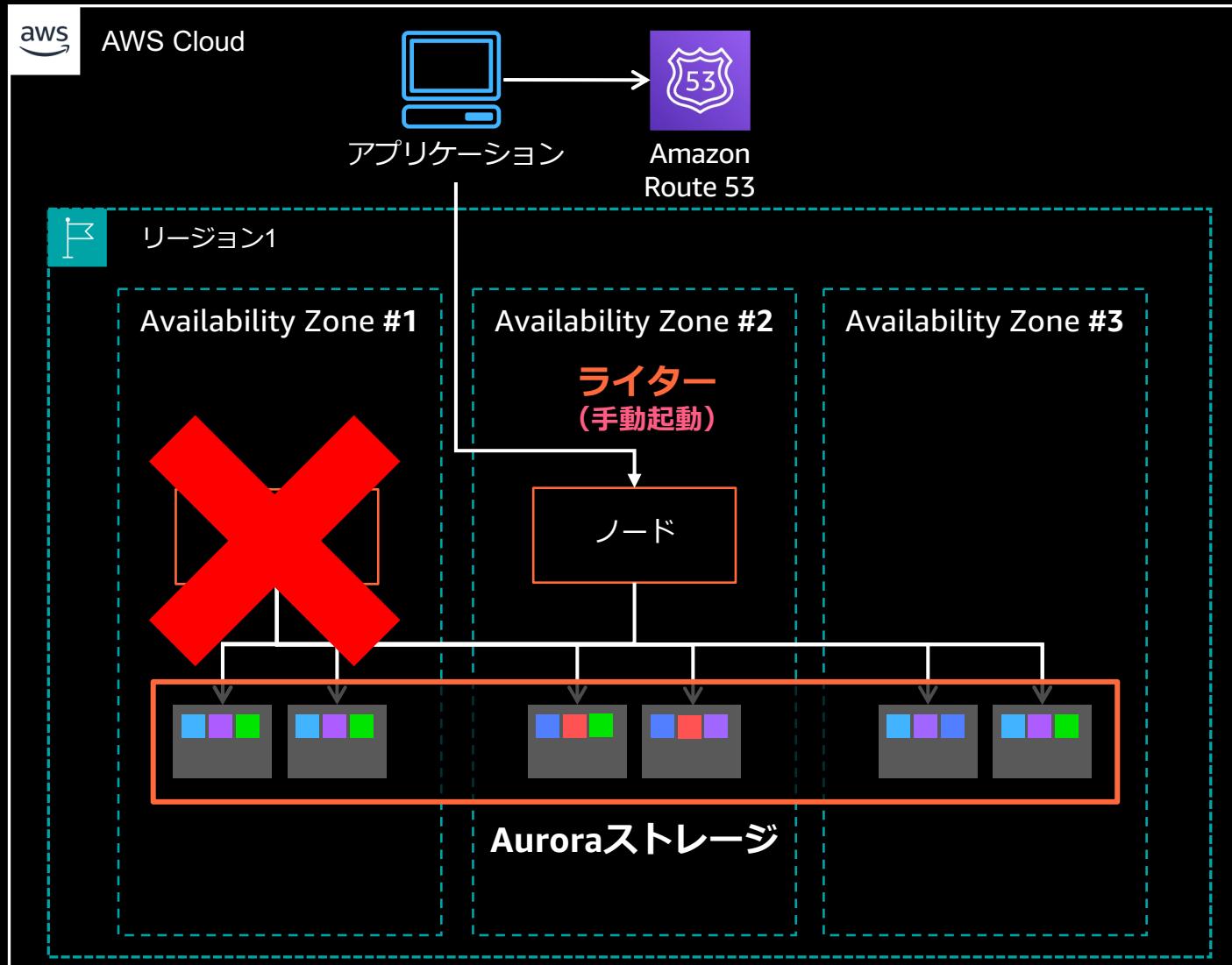
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：同一 AZ に自動再作成（10分未満）
- **AZ 全体障害時：**別 AZ に手動インスタンス作成が必要

シングル AZ 構成



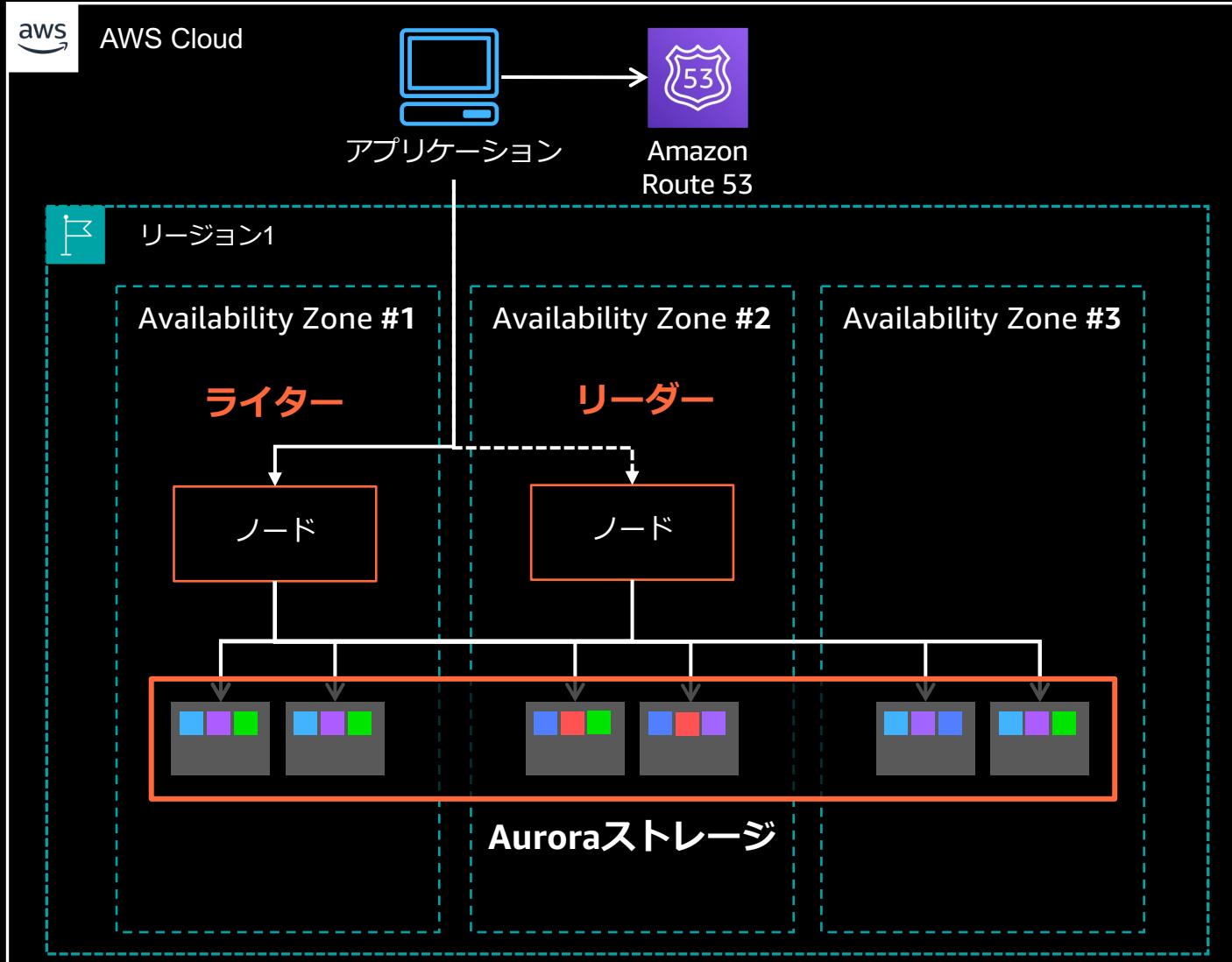
基本構成

- 単一 AZ に DB インスタンス配置
- ストレージは 3 つの AZ にわたり 6 コピー分散

障害対応

- インスタンス障害時：
同一 AZ に自動再作成（10分未満）
- AZ 全体障害時：
別 AZ に手動インスタンス作成が必要

マルチ AZ 構成



基本構成

- 複数 AZ にインスタンス分散（最大15台）
- 1 ライター + リーダー (1台以上) 構成
- 3 AZにデータが配置

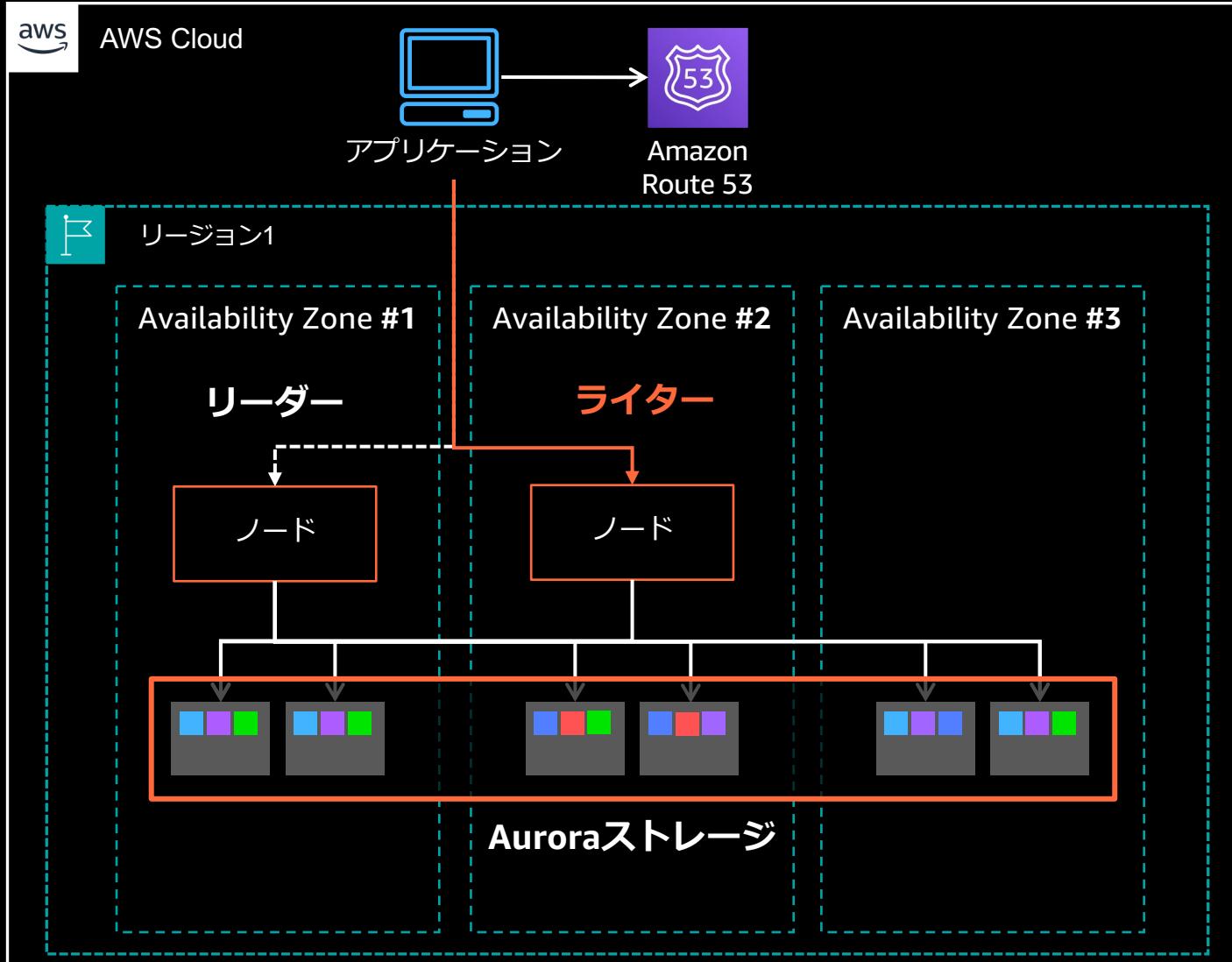
データ同期と可用性

- 非同期レプリケーション
(ミリ秒単位のラグ)
- 共有ストレージにより
フェイルオーバー時のデータ損失なし

運用管理

- リーダーエンドポイントによる
読み取り負荷分散 (ラウンドロビン)
- 外部オブザーバーによる監視と
クオーラム合意
- 自動/手動フェイルオーバー
(Route 53による DNS 切替)

マルチ AZ 構成



基本構成

- 複数 AZ にインスタンス分散（最大15台）
- 1 ライター + リーダー (1台以上) 構成
- 3 AZにデータが配置

データ同期と可用性

- 非同期レプリケーション
(ミリ秒単位のラグ)
- 共有ストレージにより
フェイルオーバー時のデータ損失なし

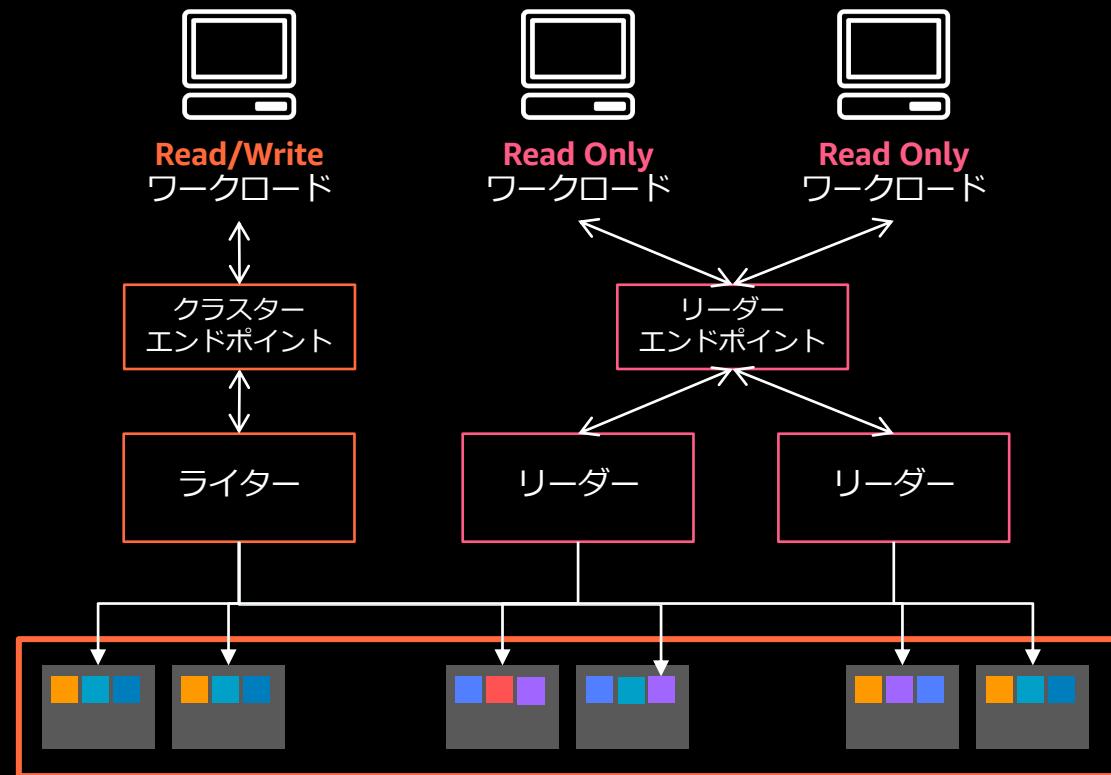
運用管理

- リーダーエンドポイントによる
読み取り負荷分散 (ラウンドロビン)
- 外部オブザーバーによる監視と
クオーラム合意
- 自動/手動フェイルオーバー
(Route 53による DNS 切替)

リーダーインスタンスとエンドポイント



リージョン1



読み取りスケーリング

- 読み取り専用リーダー (Aurora レプリカ) を利用
- 最大 15 台のリーダーで読み取り負荷を分散
- レプリカラグは通常 100ms 未満

効率と監視

- CloudWatch/RDS コンソールでレプリカラグ監視
- 共有ストレージ方式で無駄な書き込み処理なし

接続タイプ

- クラスター エンドポイント : ライター接続用
- リーダー エンドポイント : 読み取り負荷分散用
- インスタンス エンドポイント : 特定インスタンス接続用

Amazon Aurora の可用性

ディスク障害: ストレージデバイスの物理的または論理的な故障

データ破損: データの整合性が損なわれる問題

データベースサーバー障害: データベースを実行するコンピュートインスタンスの問題

データセンター障害: 単一のデータセンター全体に影響する障害 (AZ 障害)

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

別リージョンにスナップショットのコピーがある場合

- スナップショットから別リージョンで復元
- 復元には数時間要し、最新データの損失リスクあり (データ量に依存)

Aurora グローバルデータベースを事前構成した場合

- セカンダリリージョンへの手動フェイルオーバーで 30 秒に復旧可能
- ストレージベースのレプリケーションにより、通常1秒未満のレプリカラグを実現

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

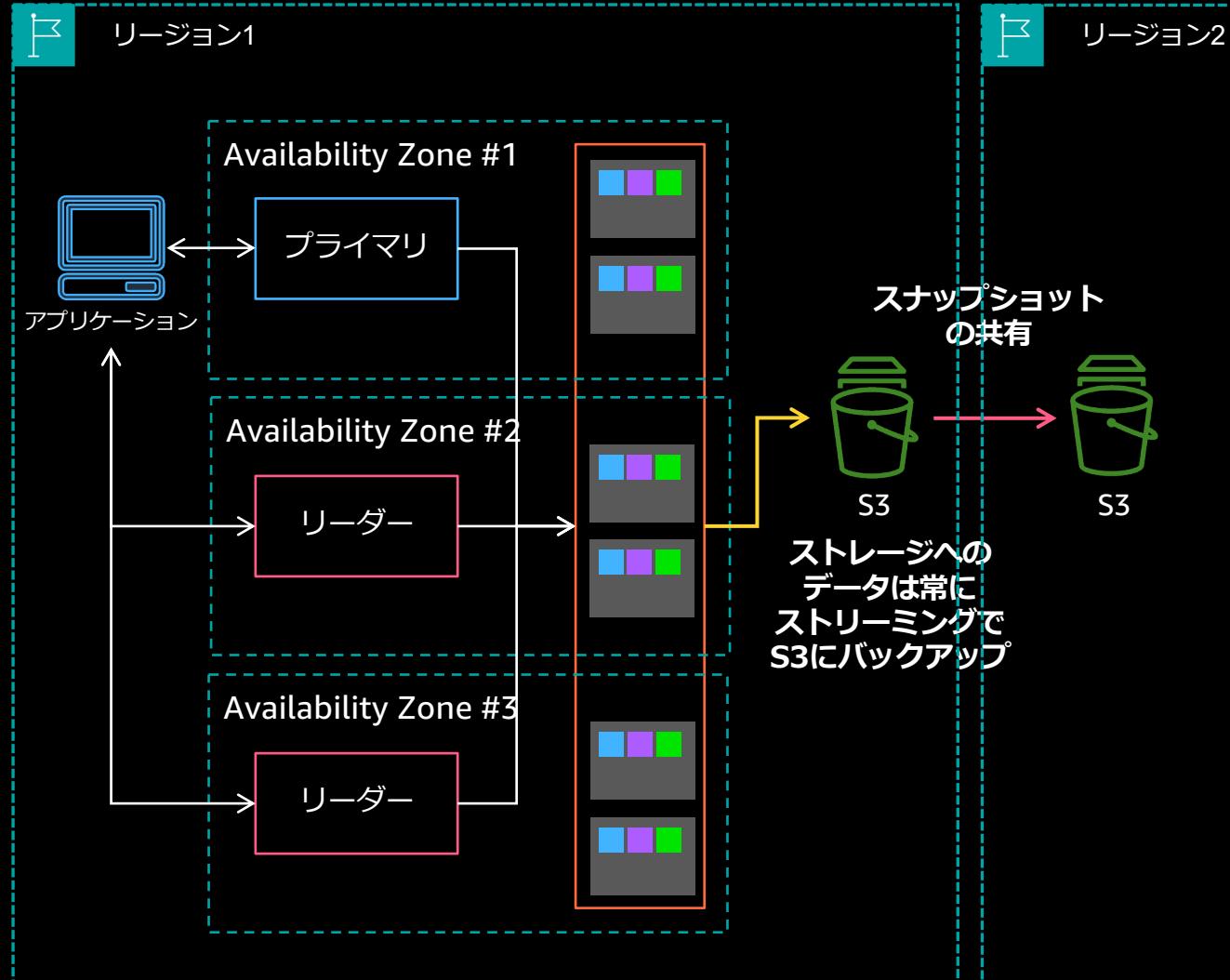
別リージョンにスナップショットのコピーがある場合

- スナップショットから別リージョンで復元
- 復元には数時間要し (データ量に依存)、最新データの損失リスクあり (バックアップ頻度に依存)

Aurora グローバルデータベースを事前構成した場合

- セカンダリリージョンへの手動フェイルオーバーで 30 秒に復旧可能
- ストレージベースのレプリケーションにより、通常 1 秒未満のレプリカラグを実現

再掲：Aurora のバックアップ & リカバリー



Aurora のバックアップ

- 自動バックアップ(必須)
- 手動スナップショット(任意)

仕組み

- 継続的にバックアップを S3 に保存
- Auroraのバックアップは差分で取得されるためバックアップ取得期間の任意の時点にリカバリすることが可能 (ポイントインタイムリカバリ : PITR)
- バックアップはストリーミングで実行されバックアップ処理がデータベースのパフォーマンスに影響を与える事はない
- スナップショットを別リージョンにコピーしたり別アカウントと共有ができる

*リストア・リカバリ時間がお客様の復旧時間要件を満たせるかを検証で確認することが重要

Amazon Aurora の可用性

大規模障害: 複数のデータセンターに影響する広範囲な障害 (AWS リージョン障害)

Amazon Aurora

別リージョンにスナップショットのコピーがある場合

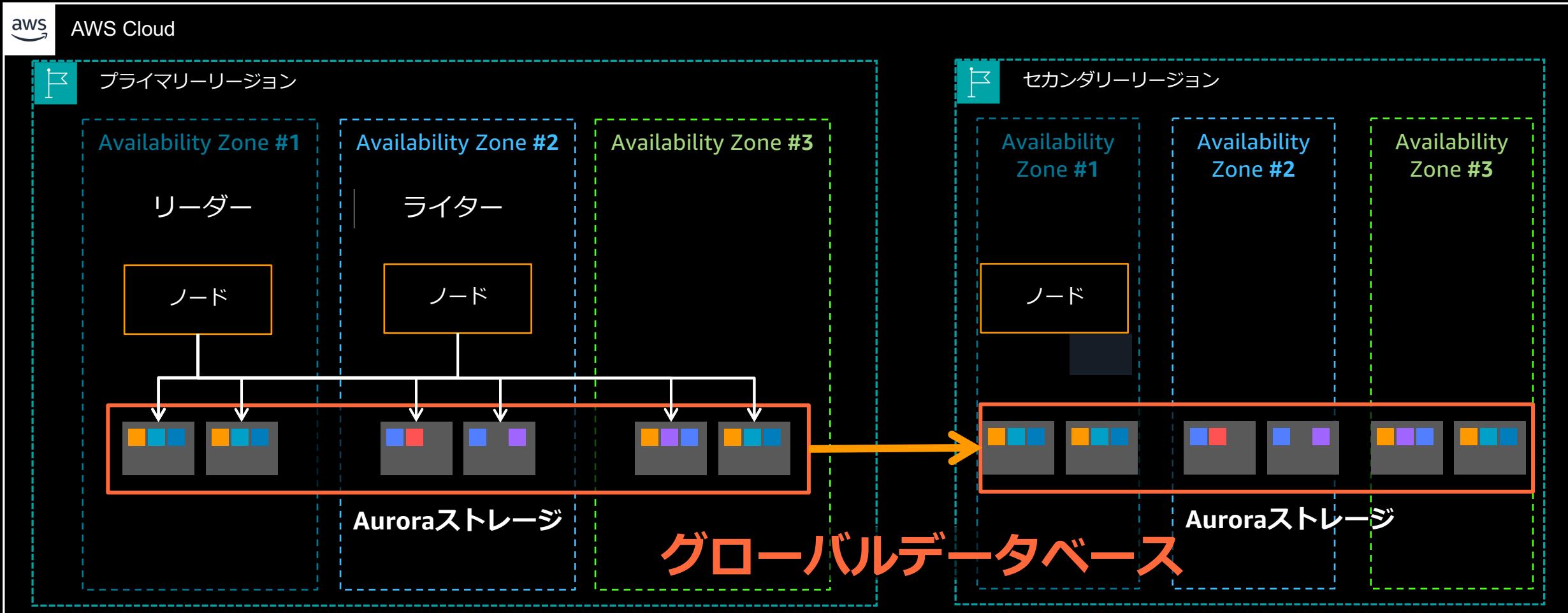
- スナップショットから別リージョンで復元
- 復元には数時間要し、最新データの損失リスクあり (データ量に依存)

Aurora グローバルデータベースを事前構成した場合

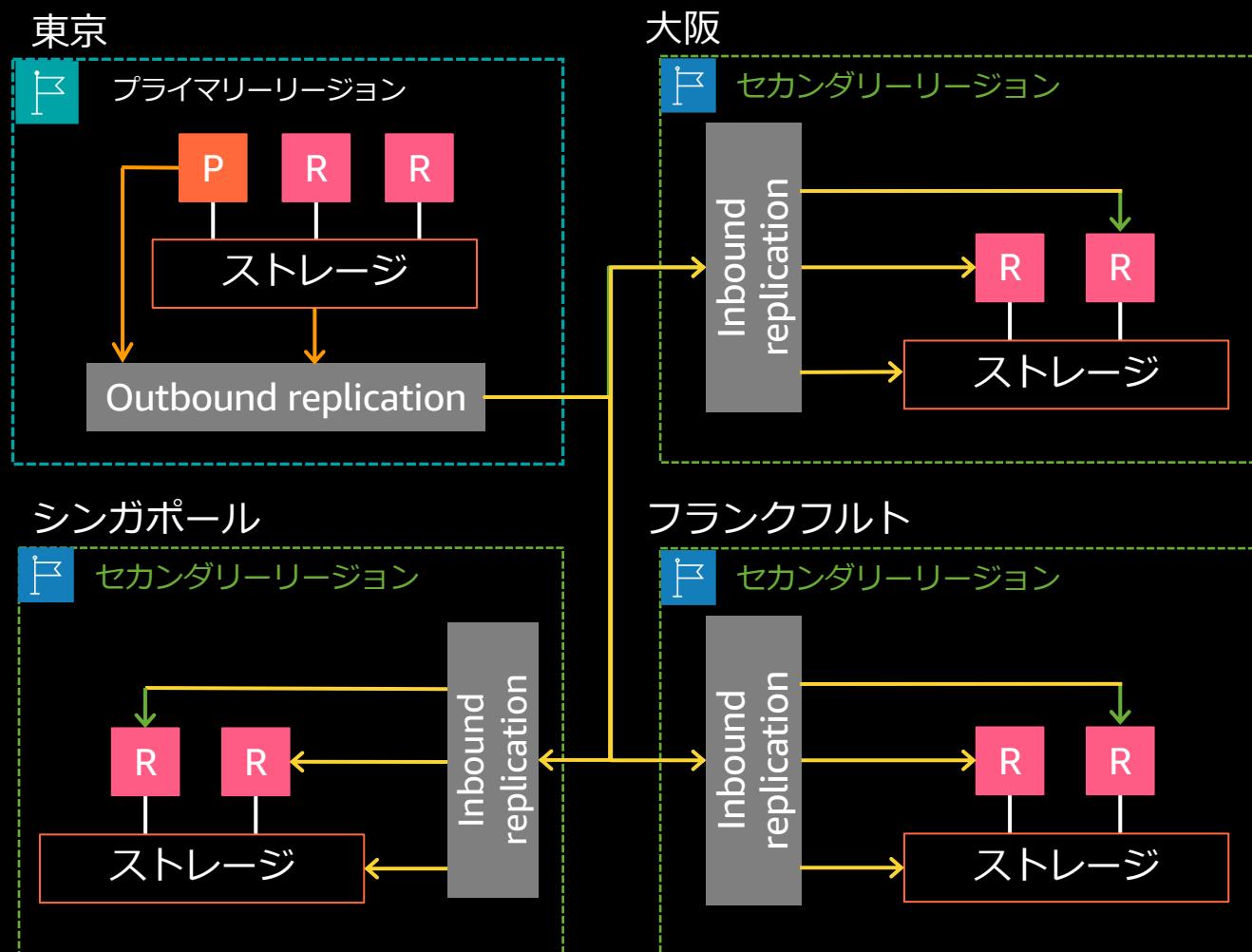
- セカンダリリージョンへの手動フェイルオーバーで通常 30 秒未満で復旧可能
- ストレージベースのレプリケーションにより、通常 1 秒未満のレプリカラグを実現

Aurora グローバルデータベース

複数の AWS リージョンにまたがる複数の Aurora DB クラスター : リージョンをまたいだ耐障害性向上



Amazon Aurora グローバルデータベース



高スループット:

低レプリケーションラグ:

- 高負荷状況下でも、リージョン間の
レプリケーションラグは通常1秒未満

迅速な災害復旧:

- 計画的なスイッチオーバー:
復旧時間が通常 30 秒未満、データ損失なし
- 計画外のフェールオーバー:
リージョン障害後、通常 1 分未満でリカバリ

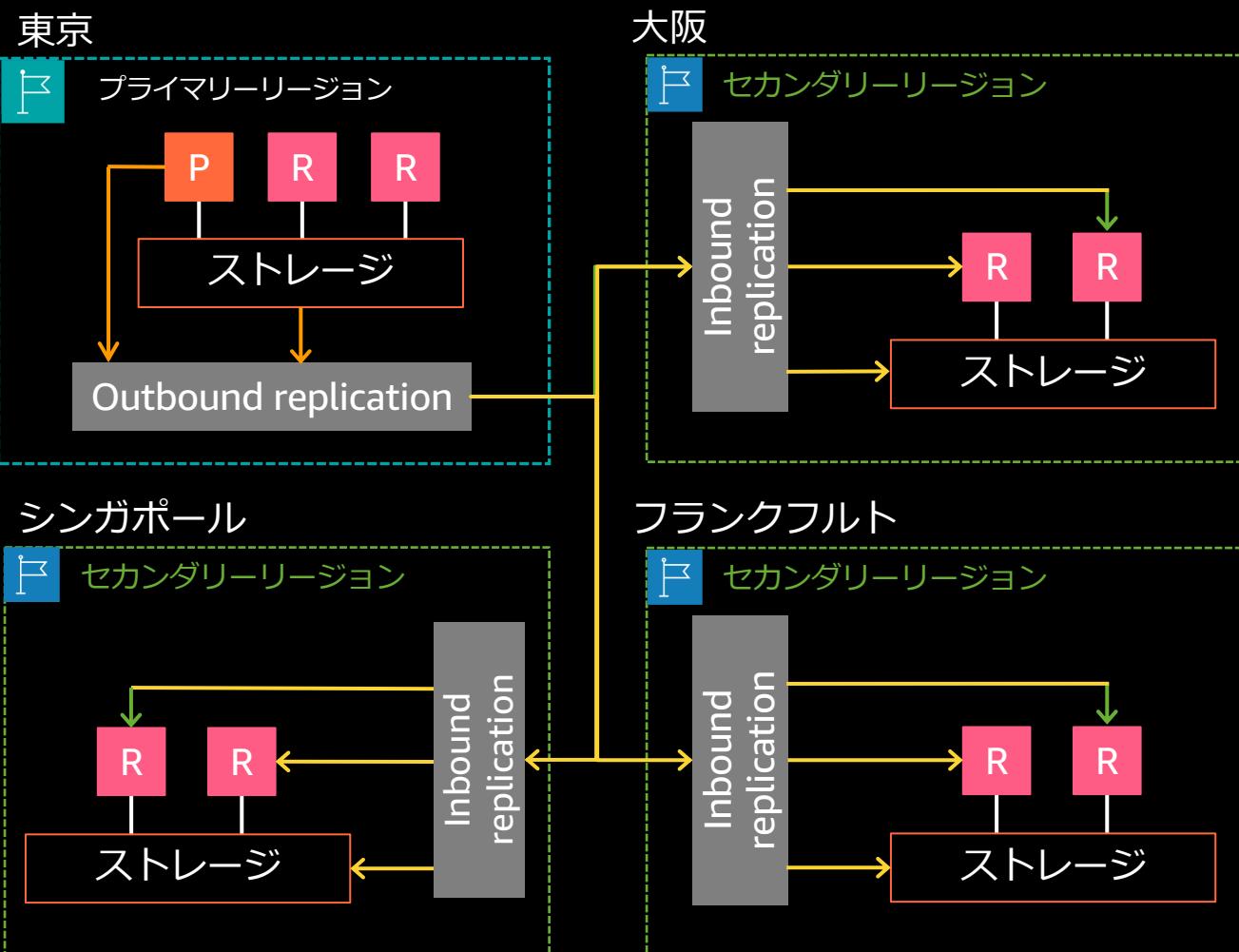
柔軟な構成:

- 最大 10 個のセカンダリリージョンをサポート
- 既存の Aurora クラスターからインプレースでの
グローバルデータベースへの拡張が可能

詳細はドキュメント参照 :

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

Amazon Aurora グローバルデータベース



高スループット:

低レプリケーションラグ:

- 高負荷状況下でも、リージョン間の
レプリケーションラグは通常1秒未満

迅速な災害復旧:

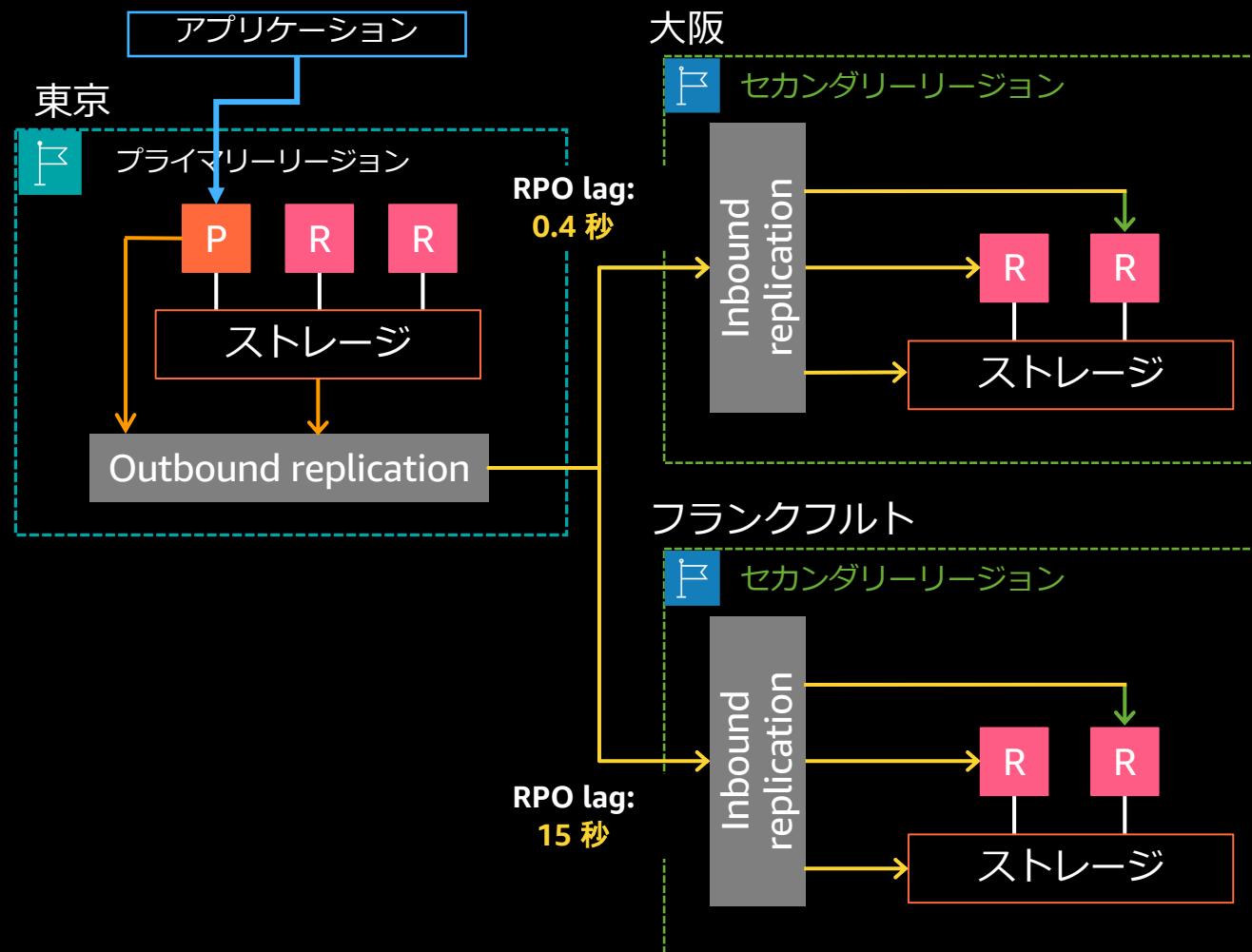
- 計画的なスイッチャーバー:
復旧時間が通常 30 秒未満、データ損失なし
- 計画外のフェールオーバー:
リージョン障害後、通常 1 分未満でリカバリ

柔軟な構成:

- 最大 10 個のセカンダリリージョンをサポート
- 既存の Aurora クラスターからインプレースでの
グローバルデータベースへの拡張が可能

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



機能:

- PostgreSQL ベースの Aurora Global Database
- 最大 RPO (目標復旧ポイント) を定義可能

動作:

- 設定した RPO 上限を全セカンダリリージョンが超過した場合、いずれかが追いつくまで Primary クラスターの書き込みを一時停止

設定方法:

- rds.global_db_rpo パラメータ
- 20 秒 ~ 68 年の範囲で設定

監視メトリクス:

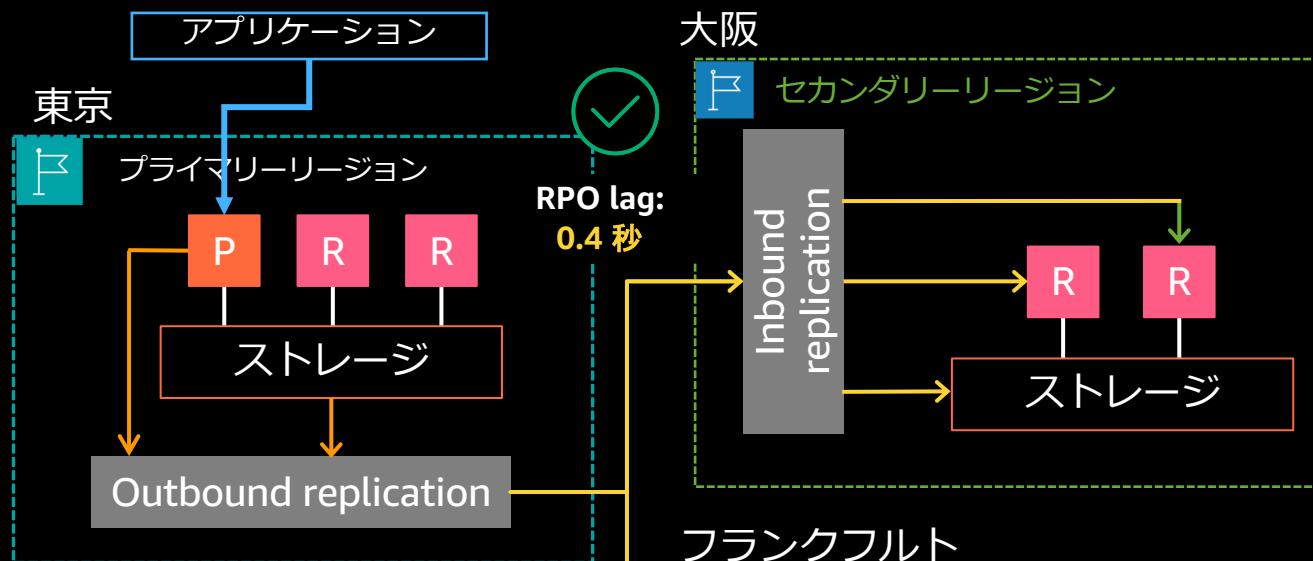
- AuroraGlobalDBRPOLag: セカンダリクラスターの RPO ラグ時間 (秒)

詳細はドキュメント参照 :

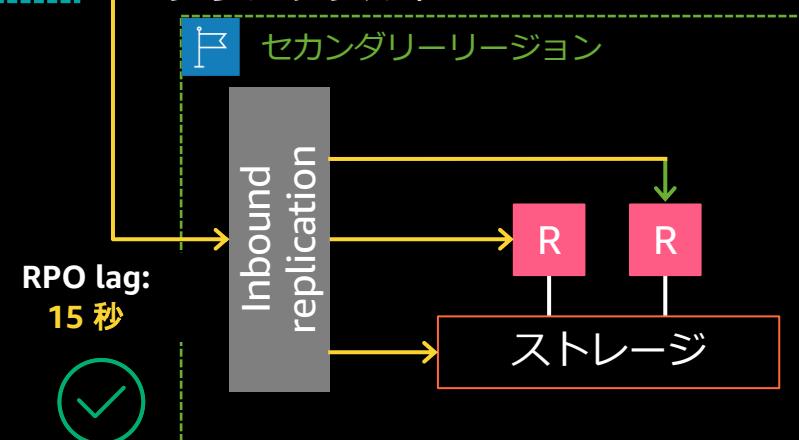
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒



例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 0.4 秒
- ・ フランクフルト - ラグ 15 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 未満

結果:

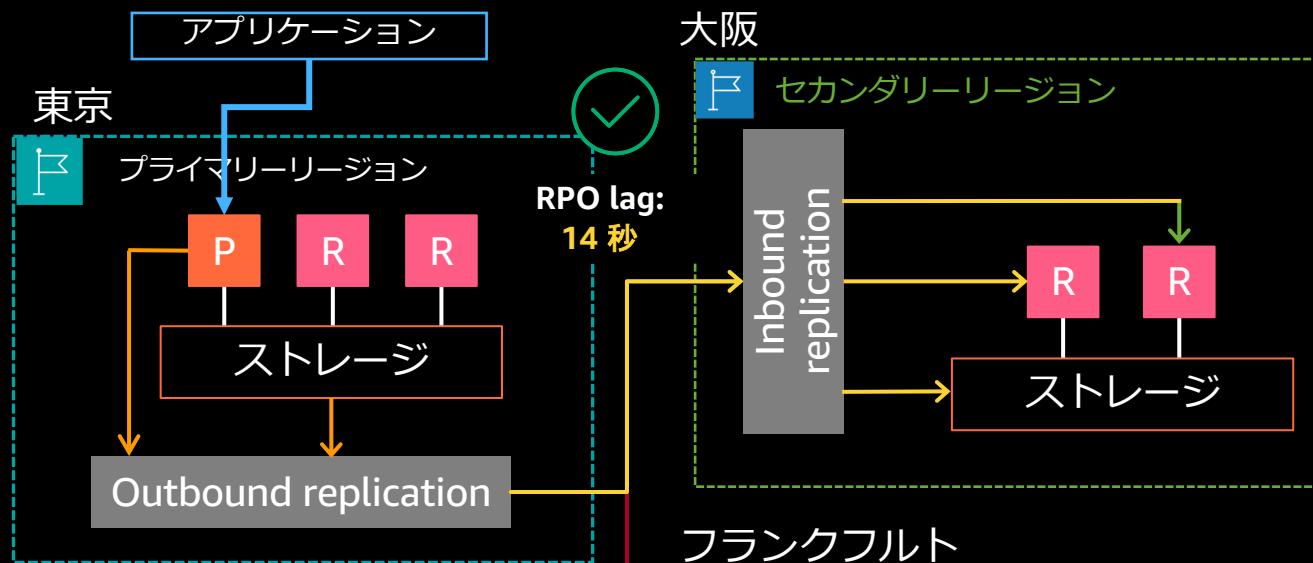
- ・ 全てのセカンダリーが基準値内

プライマリインスタンス:

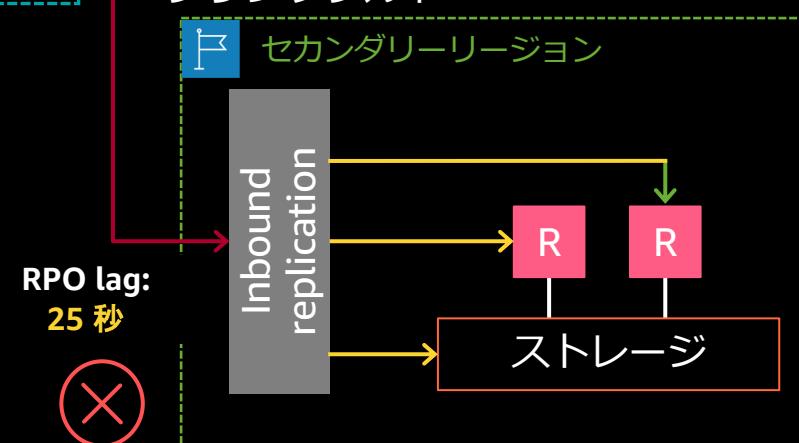
- ・ トランザクション継続

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒



例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 14 秒
- ・ フランクフルト - ラグ 25 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 以上

結果:

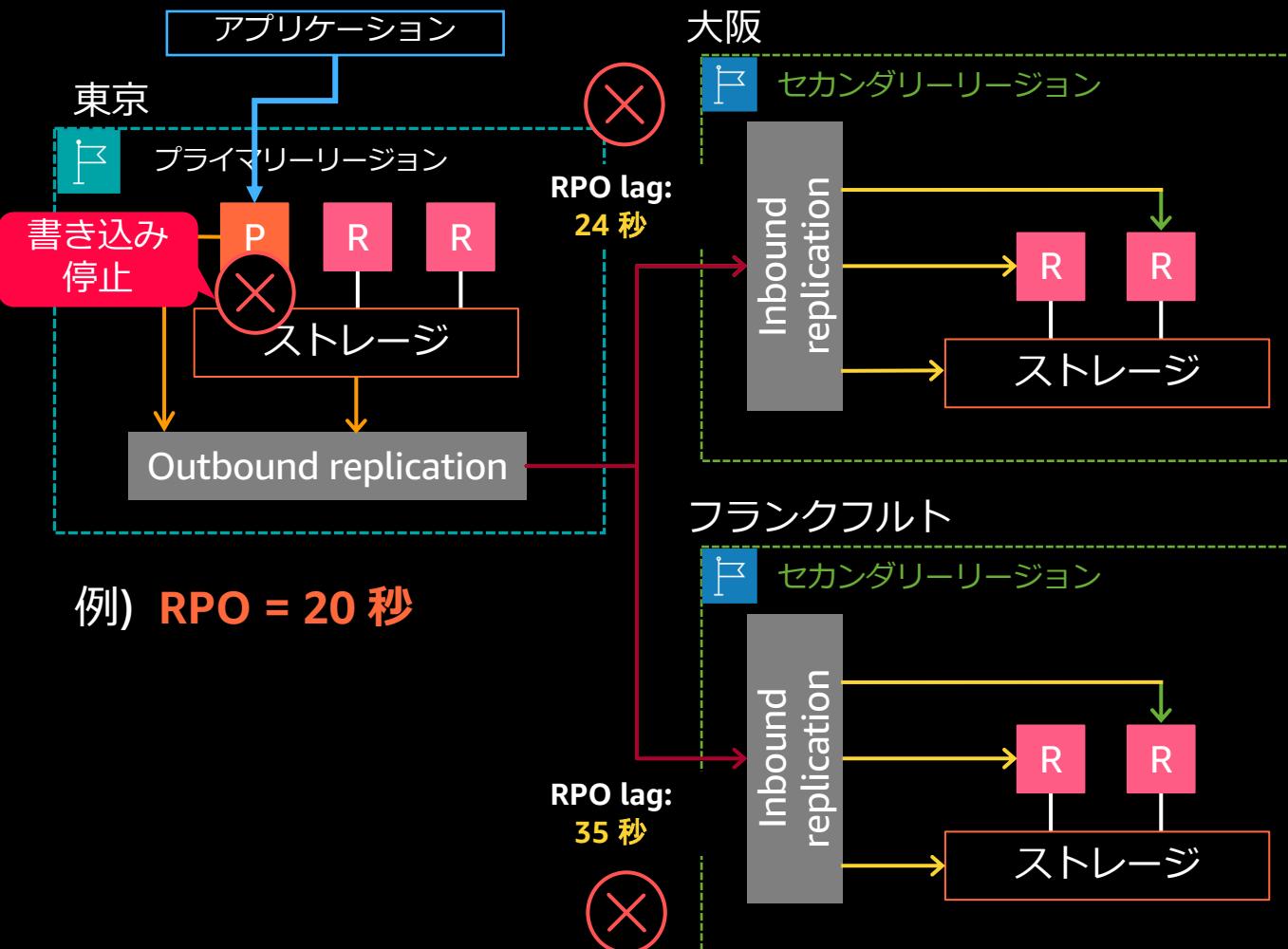
- ・ 1 つのセカンダリー (大阪) が基準値内

プライマリインスタンス:

- ・ トランザクション継続

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒

例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 24 秒
- ・ フランクフルト - ラグ 35 秒

上限確認:

- ・ 大阪 - 20 秒以上
- ・ フランクフルト - 20 秒以上

結果:

- ・ 全てのセカンダリーが基準値超過

プライマリインスタンス:

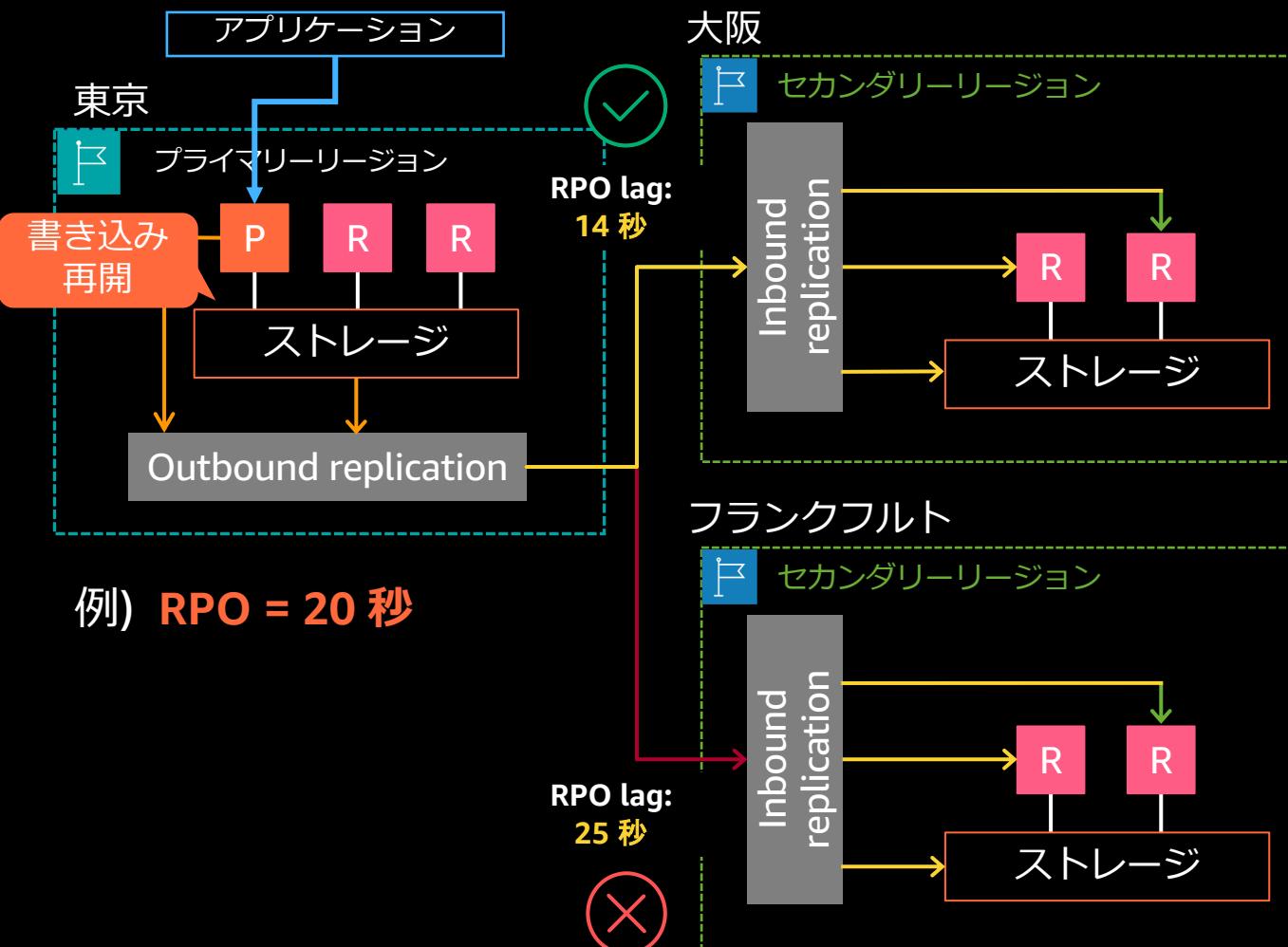
- ・ トランザクション停止

詳細はドキュメント参照 :

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

グローバルデータベース RPO 管理

RPO (停止時に失われる可能性があるデータの最大量) 要件が厳格なアプリケーション向け機能



例) RPO = 20 秒

例) RPO = 20 秒

PRO ラグ:

- ・ 大阪 - ラグ 14 秒
- ・ フランクフルト - ラグ 25 秒

上限確認:

- ・ 大阪 - 20 秒 未満
- ・ フランクフルト - 20 秒 以上

結果:

- ・ 1 つのセカンダリー (大阪) が基準値内

プライマリインスタンス:

- ・ トランザクション再開

詳細はドキュメント参照 :

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

グローバルデータベース制限事項

- ・ 特定の AWS リージョンと特定の Aurora MySQL および Aurora PostgreSQL バージョンでのみ利用可能
- ・ メモリ最適化インスタンス (db.r5 以上のインスタンス推奨)
- ・ Aurora Serverless v1、Aurora バックトラック機能は利用不可
- ・ クロスリージョンスイッチオーバー/フェールオーバーは、プライマリとセカンダリが同一のメジャー/マイナーバージョンの場合のみ実行可能
- ・ セカンダリ DB クラスターでは Aurora Auto Scaling 利用不可
- ・ グローバルデータベース内の Aurora DB クラスターを個別に停止/開始不可
- ・ Secrets Manager は未サポート (事前に Secrets Manager 統合を無効化する必要)
- ・ Aurora PostgreSQL 16.6, 15.10, 14.15, 13.18, 12.22 より前と、Aurora MySQL 3.09 より前で、計画外イベント中のセカンダリリージョンの読み取り可用性に未対応
- ・ PostgreSQL 固有の制限:
 - ・ セカンダリクラスターでクラスターキャッシュ管理がサポートされない
 - ・ RPO 機能が有効な場合、メジャーバージョンアップグレード不可

詳細はドキュメントを参照: https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html#aurora-global-database.limitations

グローバルデータベースのコスト

インスタンス:

- プライマリおよびセカンダリリージョンの各インスタンスに対して、標準の Aurora インスタンス料金

ストレージ:

- 各リージョンのストレージ使用量に対して標準の Aurora ストレージ料金

レプリケーション:

- プライマリリージョンから各セカンダリリージョンへのレプリケートされた書き込み I/O 操作に対して課金
- プライマリリージョンの書き込み I/O 操作と同数のレプリケート I/O 操作が各セカンダリリージョンで実行

データ転送:

- リージョン間のデータ転送に対して、標準の AWS クロスリージョンデータ転送料金

バックアップ:

- 各リージョンのバックアップストレージに対して標準料金

詳細はドキュメントを参照: https://aws.amazon.com/jp/rds/aurora/pricing/#Aurora_Global_Database_costs

グローバルデータベースのコスト

インスタンス:

- プライマリおよびセカンダリリージョンの各インスタンスに対して、標準の Aurora インスタンス料金

ストレージ:

- 各リージョンのストレージ使用量に対して標準の Aurora ストレージ料金

レプリケーション:

- プライマリリージョンから各セカンダリリージョンへのレプリケートされた書き込み I/O 操作に対して課金
- プライマリリージョンの書き込み I/O 操作と同数のレプリケート I/O 操作が各セカンダリリージョンで実行

データ転送:

- リージョン間のデータ転送に対して、標準の AWS クロスリージョンデータ転送料金

バックアップ:

- 各リージョンのバックアップストレージに対して標準料金

詳細はドキュメントを参照: https://aws.amazon.com/jp/rds/aurora/pricing/#Aurora_Global_Database_costs

まとめ

再掲：データベースの可用性：RTO / RPO

RPO: Recovery Point Objective

- リカバリポイント目標
- 障害時に、どの時点のデータまでリカバリ可能か

RTO: Recovery Time Objective

- リカバリ時間目標
- 障害時に、どれだけの時間でリカバリを完了させるか

まとめ

データベースの可用性設計

- RTO/RPO 要件：許容できる復旧時間と復旧ポイント
- 想定する障害の種類と対応策
- コストと運用負荷のバランス

Aurora は様々な障害に対する多層防御を提供

- ディスク障害 : Aurora ストレージで 3 AZ 6 箇所に複製
- データ破損 : 自己修復機能とバックアップ・リカバリ (ポイントインタイムリカバリ)
- DB サーバー/データセンター障害 : 自動フェイルオーバー (マルチ AZ 構成)
- 大規模障害 (リージョン障害) : Global Database による地理的冗長性

まとめ

データベースの可用性設計

- RTO/RPO 要件：許容できる復旧時間と復旧ポイント
- 想定する障害の種類と対応策
- コストと運用負荷のバランス

Aurora は様々な障害に対する多層防御を提供

- ディスク障害 : Aurora ストレージで 3 AZ 6 箇所に複製
- データ破損 : 自己修復機能とバックアップ・リカバリ (ポイントインタイムリカバリ)
- DB サーバー/データセンター障害 : 自動フェイルオーバー (マルチ AZ 構成)
- 大規模障害 (リージョン障害) : Global Database による地理的冗長性

「障害を前提とした設計 (Design for Failure)」の考え方に基づき
ビジネス要件に合わせたアーキテクチャを選択することが重要

Thank you!

Yukki

Partner Solutions Architect

@YukkiTakahashi

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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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



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



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징
- 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

基本的なコスト

インスタンス
(Provisioned/Serverless)

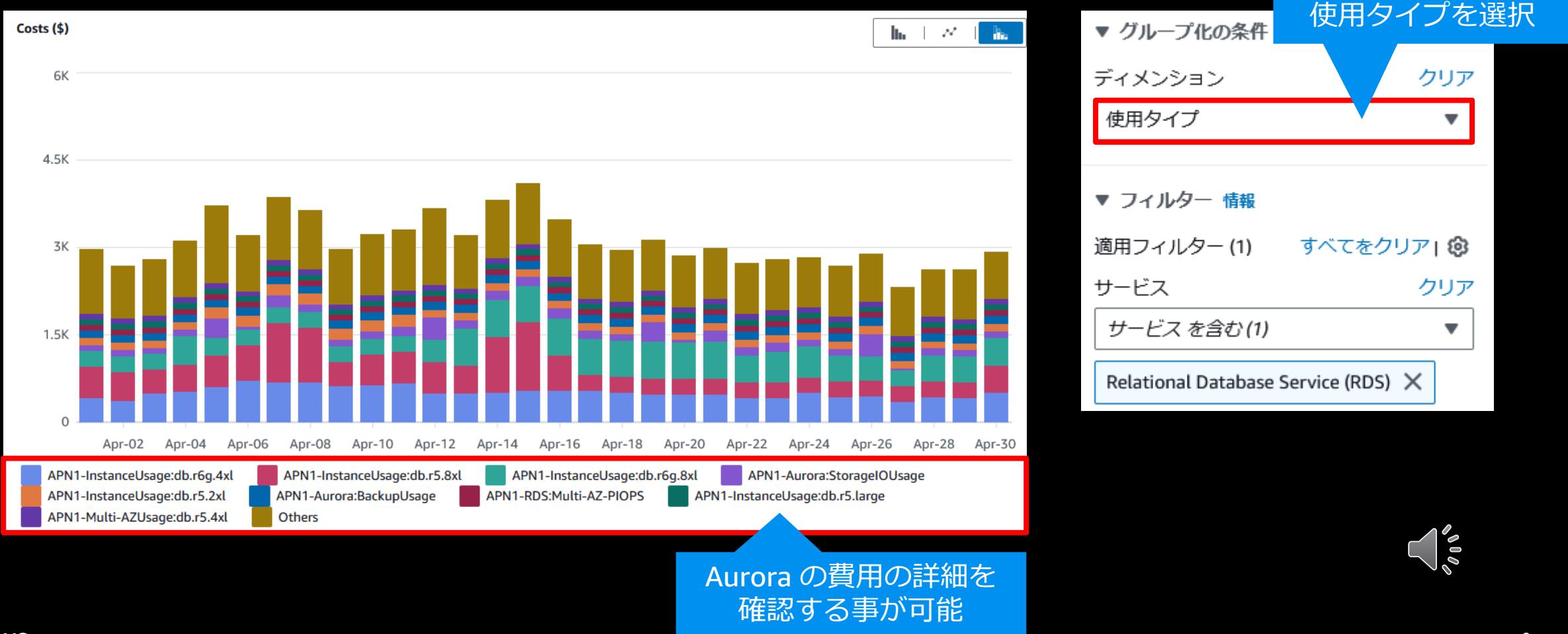
I/O
(I/O 最適化/スタンダード)

ストレージ



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



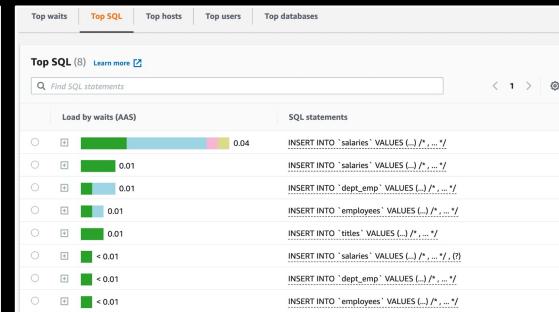
インスタンスのダウンサイジング

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



* 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** 頂スピーカー 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



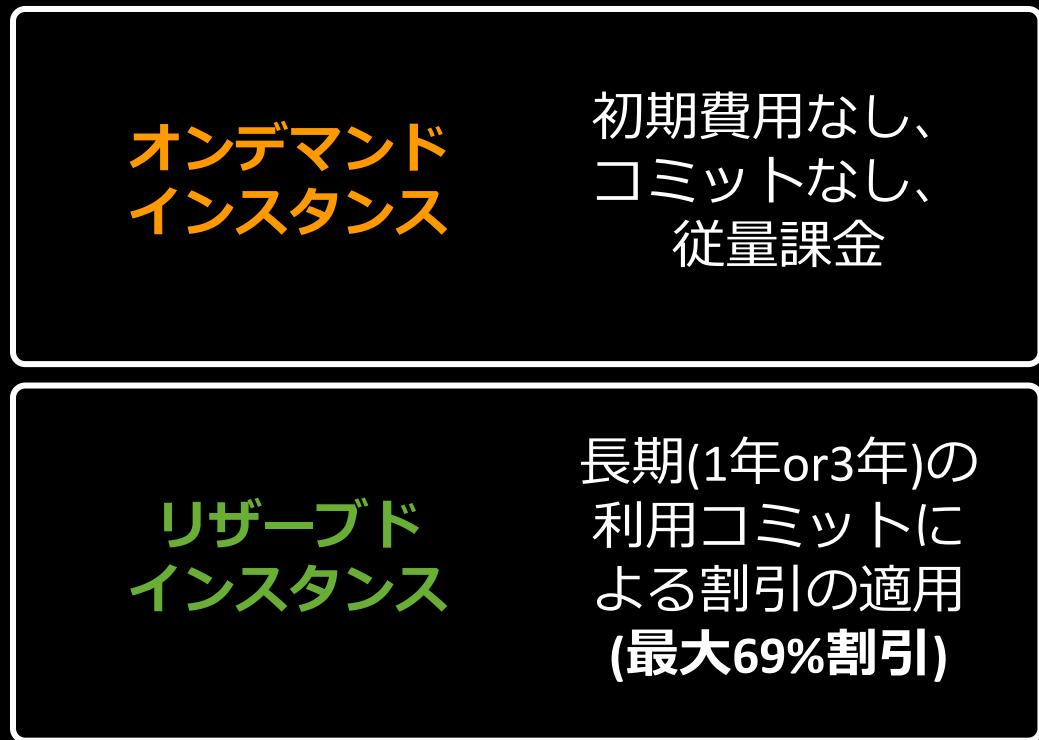
https://docs.aws.amazon.com/ja_jp/compute-optimizer/latest/ug/what-is-compute-optimizer.html



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

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



https://docs.aws.amazon.com/ja_jp/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

The screenshot shows the AWS Cost Explorer interface. On the left, under the 'AWS Cost Management' menu, the 'Recommendations' section is highlighted with a red box. A blue callout bubble with the text 'RDS を選択' (Select RDS) points to the 'Services' dropdown in the main content area. The main content area displays the 'Recommendations' page for Relational Database Service (RDS). It shows one recommendation for the db.r5.large instance type, which is estimated to save \$81.49 per month. The recommendation table includes columns for Action, Instance Type, Region, Data, License, Current Generation, and Previous Cost.

Action	Instance Type	Region	Data	License	Current Generation	Previous Cost
1 db.r5.large 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_jp/awsaccountbilling/latest/aboutv2/ce-what-is.html



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

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事でコストを抑える
- 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>

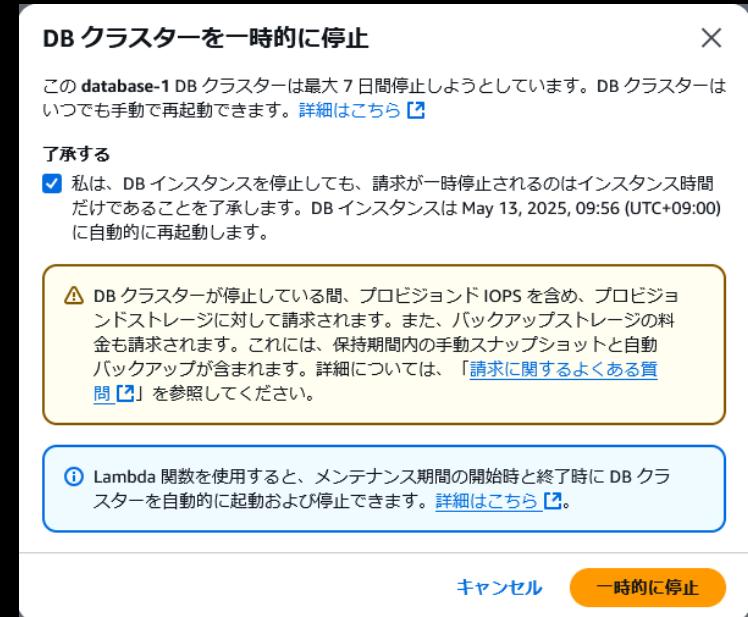
- クラスタを停止しても、停止後7日間で自動で起動する
- Aurora Global Database の一部であるクラスタの停止は不可
- クロスリージョンリードレプリカを持つクラスタの停止は不可

インスタンス

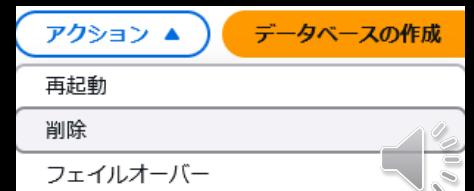
- 過剰なリードレプリカ等、ワークロードに対して不要と判断できるインスタンスを削除

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

クラスタ停止時の設定イメージ



インスタンス削除の設定



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(ギガバイト)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



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 を利用する事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/>



混在環境によるコスト最適化

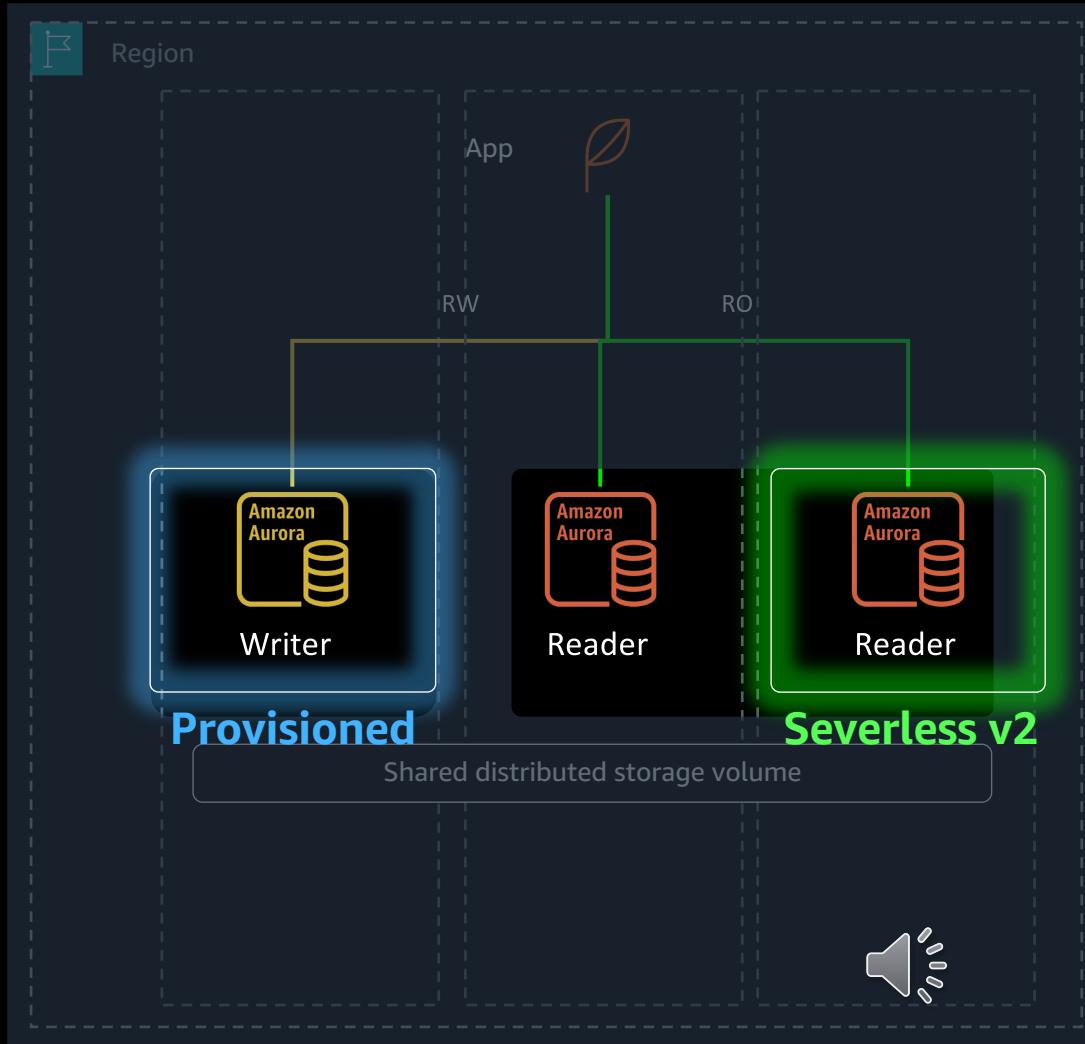
Provisioned と Serverless v2 の混在環境

- 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)

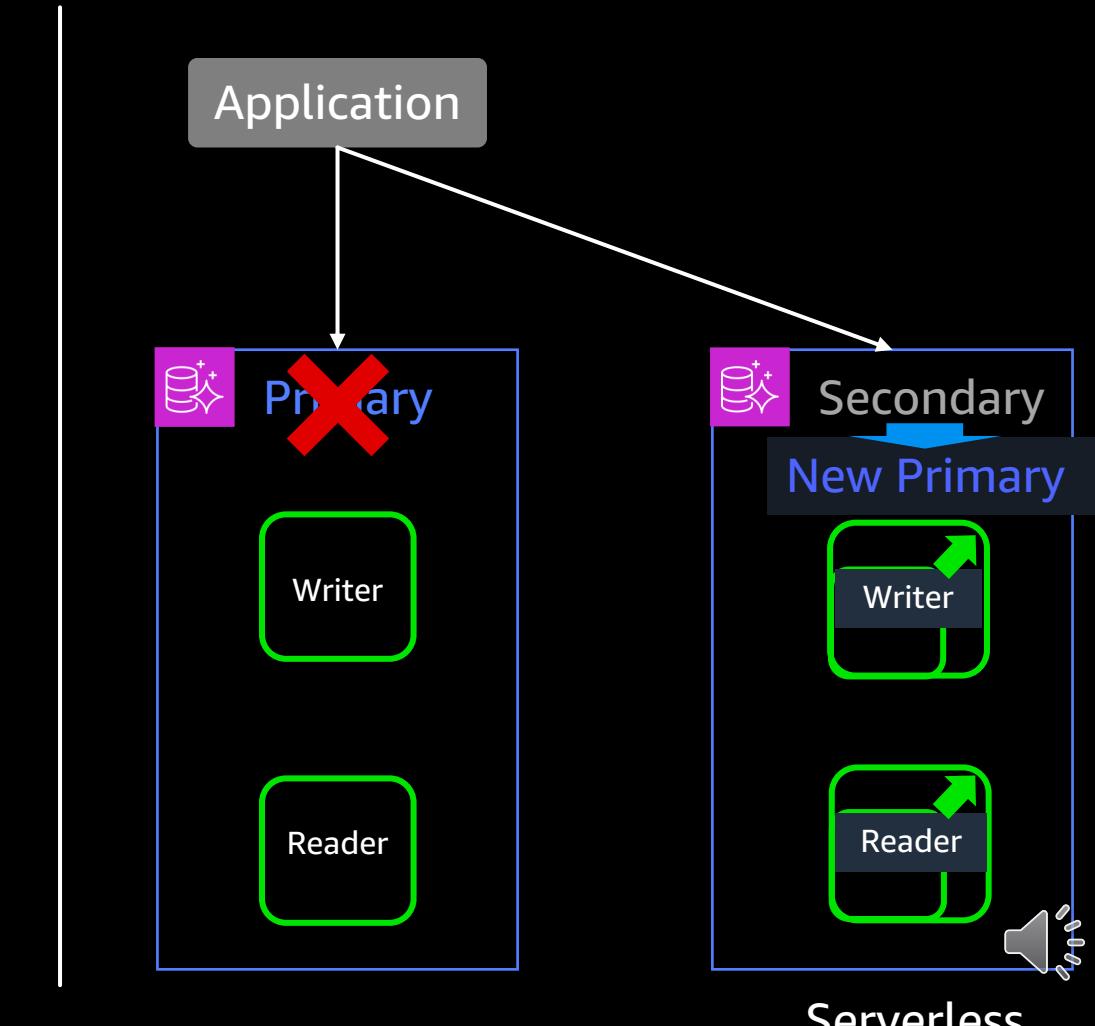
- Writer のキャパシティは Provisioned(Reserved Instance含む)で運用
- Reader のキャパシティは Serverless v2 で運用し、Reader がアイドル時は最小キャパシティ分のみの支払いにとどめる

* 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

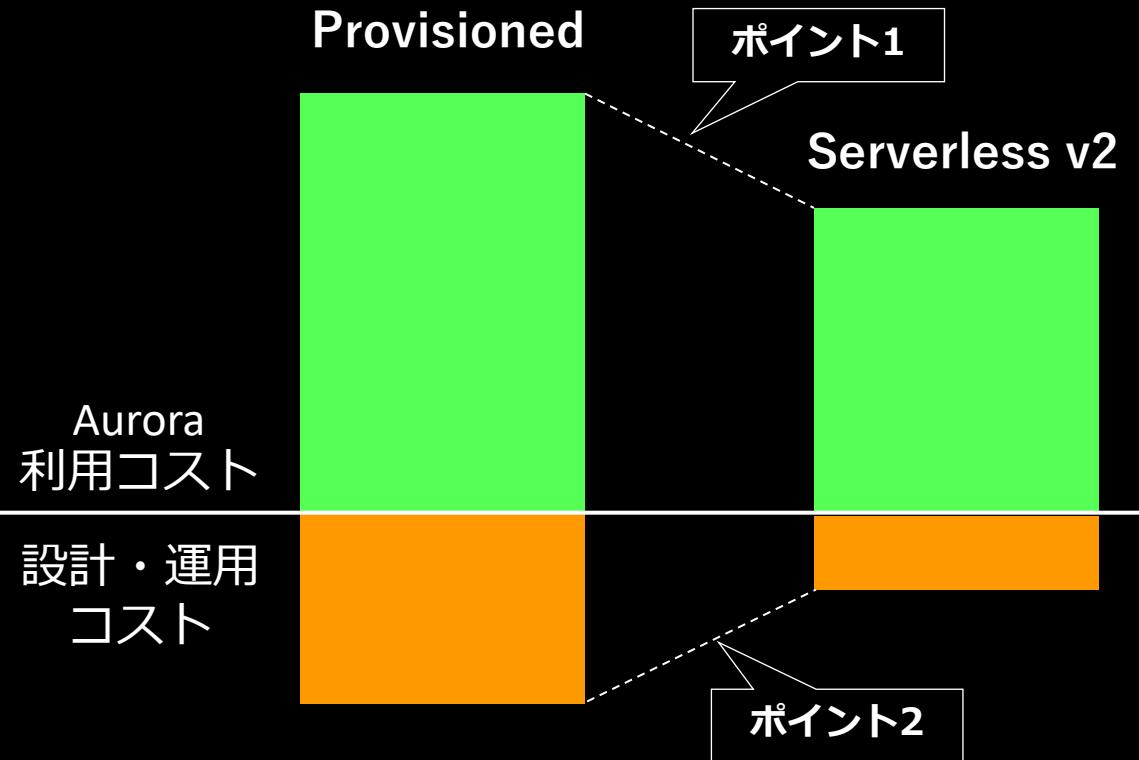


Serverless Global Database 構成によるコスト最適化

- 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 クラスターは **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 スタンダードでは、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
 - 1: 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
 - 2: 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
 - 3: 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 スタンダードでは、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
 - 1: 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
 - 2: 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
 - 3: 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ング - 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ング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。
 独自のキャッシングの設計
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ングを作成できます。

キャッシング設定
キャッシングの名前と説明 (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 (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!



AWS Black Belt Online Seminar

Amazon Aurora 移行支援プログラム&サービス

長久保 武

Solutions Architect

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=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>



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2025年5月時点のサービス内容および価格についてご説明しています。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

長久保 武

アマゾンウェブサービスジャパン
データベーススペシャリストソリューション
アーキテクト

主に製造・通信業界のお客様に対する
データベース関連の技術支援に従事しています

好きな AWS サービス
Amazon Aurora, RDS



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



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 への移行におけるAWSによる支援サービスやツール、また移行方式を把握したい方



アジェンダ

1. Aurora の概要と移行における課題
2. 移行支援プログラム & 移行サービス
 - 異種DBエンジンからの移行
 - 同種DBエンジンからの移行



アジェンダ

1. Aurora の概要と移行における課題
2. 移行支援プログラム & 移行サービス
 - 異種DBエンジンからの移行
 - 同種DBエンジンからの移行



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

優れた性能と拡張性



標準的なMySQLやPostgreSQLと比べて高いパフォーマンスを実現; 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



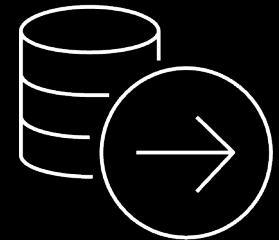
ネットワーク分離、保管時/通信の暗号化

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

Amazon Auroraへの移行で データベース管理の運用負荷を軽減



DB管理タスクをAWSへオフロード

- ・ サーバプロビジョニング
 - ・ パッチ適用
 - ・ バックアップ
 - ・ フェイルオーバー
 - ・ 拡張性や可用性担保の仕組み化
- etc.

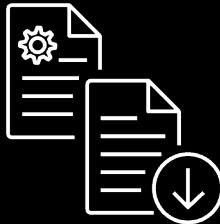
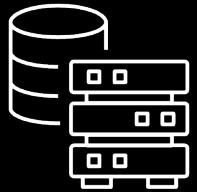


チームの生産性を高める

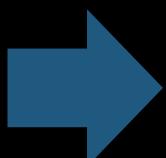
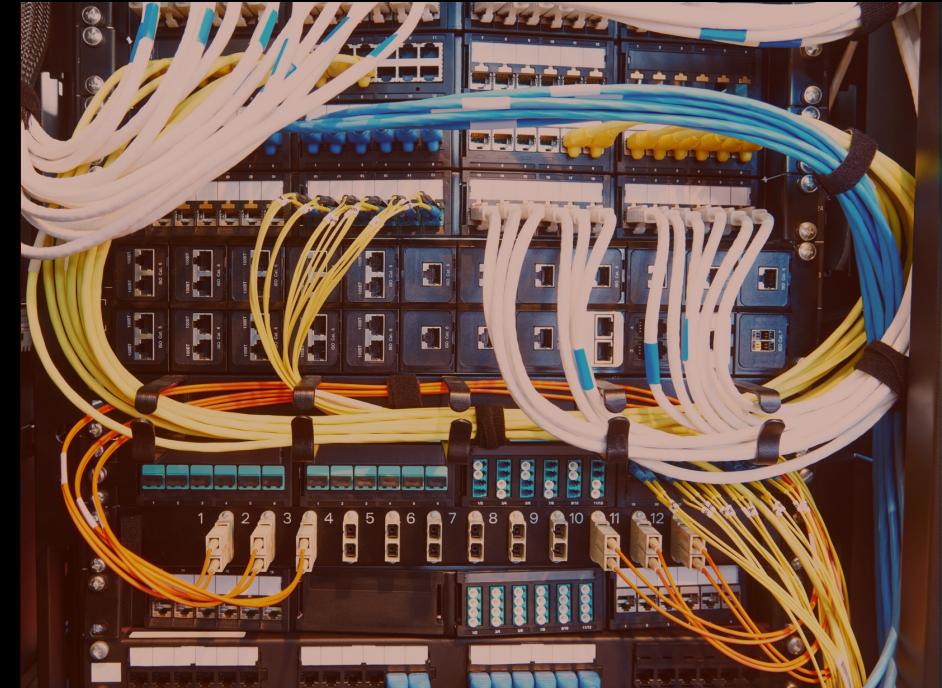
時間のかかるデータベース管理タスクや、複雑なライセンス管理から解放され、アプリケーションやビジネスに集中できる



データベースの移行作業には多くの工数がかかる



- ✓ 経験や知見が不足している中のデータベース移行計画の作成
- ✓ トラブルによる手戻りの発生
- ✓ 移行情報の収集
- ✓ 利用するツール選定
- etc.



移行の進め方や課題になりうるポイントの理解し
移行サービス・ツール・プログラムの活用で移行作業を効率化

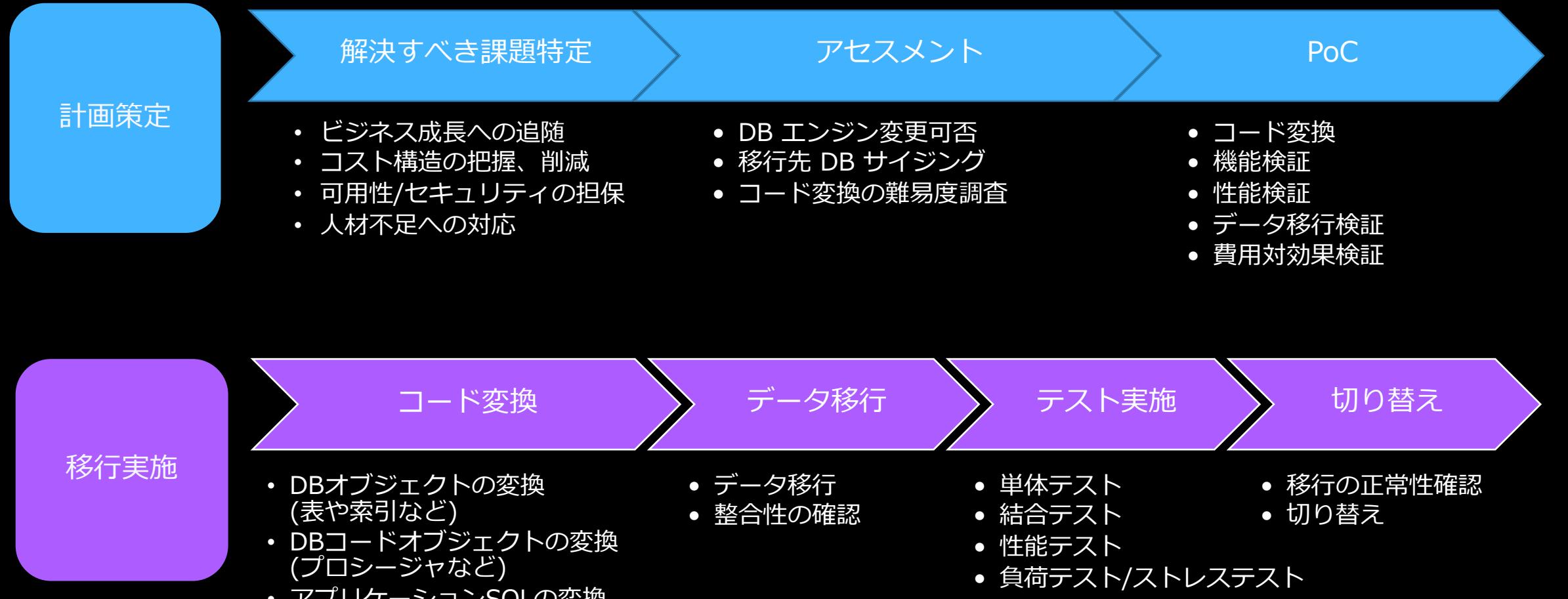


アジェンダ

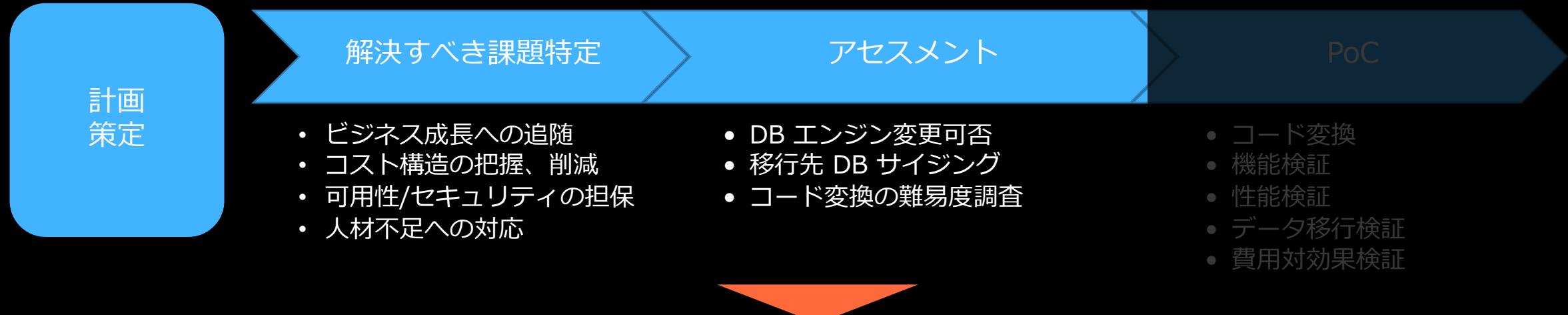
1. Aurora の概要と移行における課題
2. 移行支援プログラム＆移行サービス
 - 異種DBエンジンからの移行
 - 同種DBエンジンからの移行



データベース移行の流れ（異種DBからの移行）



移行計画の策定に対する支援プログラム



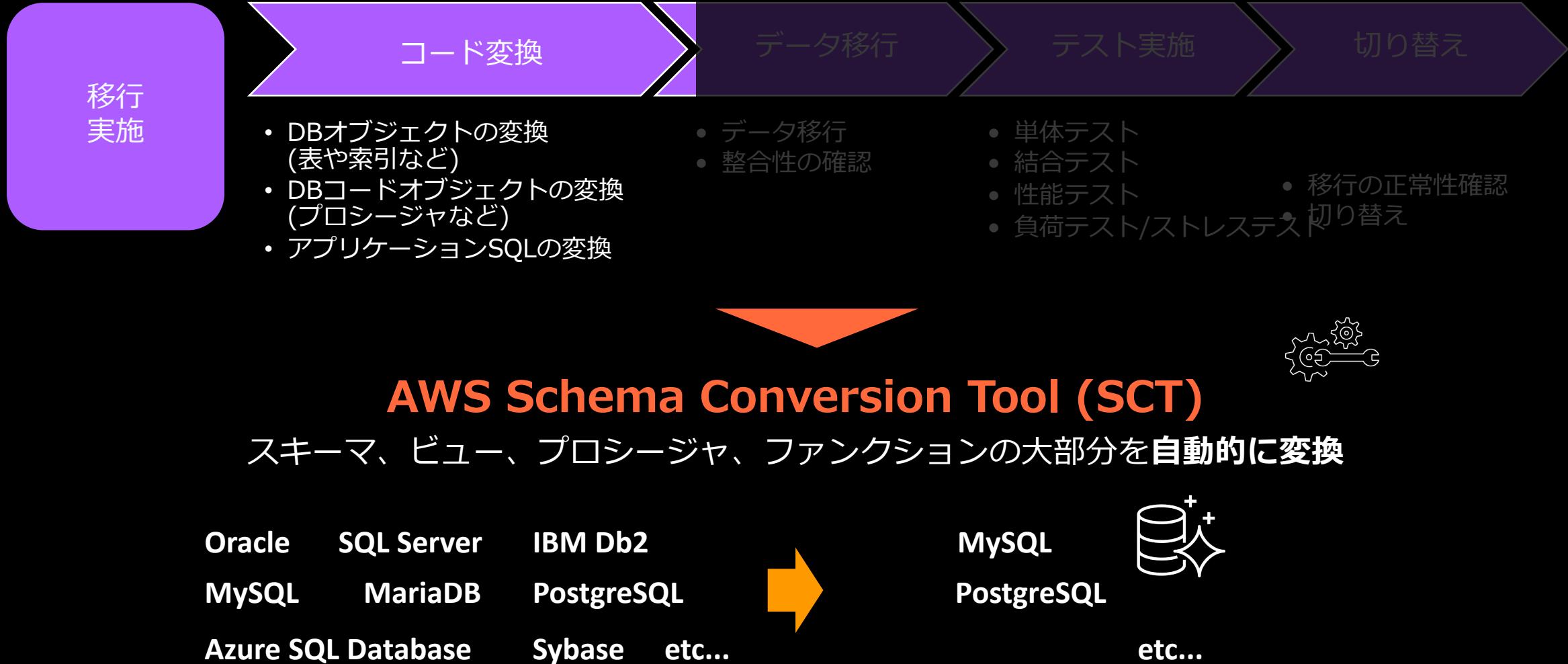
Database Freedom Workshop

Oracle、SQL Server、Db2など商用データベースからの移行を対象とした一連のワークショップ（無償）

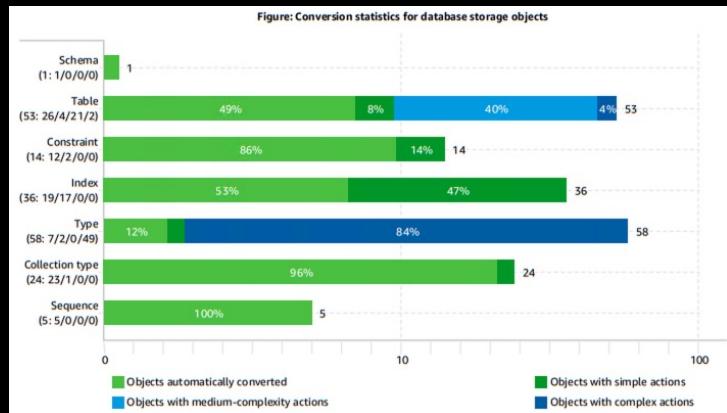
- ・移行先データベースの選定、移行先データベースに関するDive Deepセッションの提供
- ・ワーカードレポートに基づいたサイ징、コード変換の難易度の調査
- ・データ移行方式について実現性の検討
- ・移行リスク、PoCすべき点の抽出の支援 など



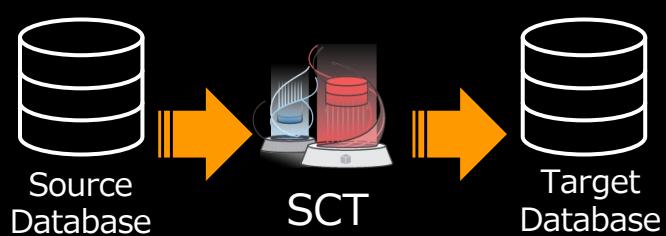
Auroraへのコード変換



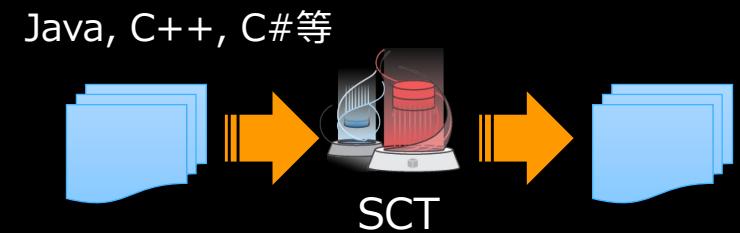
AWS SCTの使用用途



アセスメント(評価)レポート
の作成



スキーマの自動変換



アプリケーションソースコード内
のSQL移行



Database Migration Playbook

互換性・移行ベストプラクティスの詳細集

- 移行対象のトピックごとにどのようにデータベース、DWHをAWSのデータベースサービスに移行するかのベストプラクティスが記載されたガイドブック
- AWS SCTでの変換有無も記載
- データベース固有の機能と様々なデータベースオブジェクトを網羅
 - Oracle to Amazon Aurora PostgreSQL
 - Oracle to Amazon Aurora MySQL
 - Oracle to Amazon Redshift
 - SQL Server to Aurora MySQL
 - SQL Server to Aurora PostgreSQL



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

Oracle Database 19c
To Amazon Aurora with PostgreSQL Compatibility (12.4)

Migration Playbook

Configuration			
Oracle	Aurora PostgreSQL	Key Differences	Compatibility
Alerting	Alerting	<ul style="list-style-type: none">Use Event Notifications Subscription with Amazon Simple Notification Service (SNS)	
Cache and Pools	Cache and Pools	<ul style="list-style-type: none">Different cache names, similar usage	
Database Parameters	Database Parameters	<ul style="list-style-type: none">Use Cluster and Database/Cluster Parameters	
Session Parameters	Session Parameters	<ul style="list-style-type: none">SET options are significantly different in PostgreSQL	

Special Features			
Oracle	Aurora PostgreSQL	Key Differences	Compatibility
Character Set	Character Set	<ul style="list-style-type: none">UTF16 character and NCHAR/NVARCHAR data types are not supported	
Database Links	Database Links	<ul style="list-style-type: none">Different paradigm and syntax	
DBMS_SCHEDULER	DBMS_SCHEDULER	<ul style="list-style-type: none">Use Scheduler AWS Lambda	
External Tables	External Tables	<ul style="list-style-type: none">PostgreSQL doesn't support EXTERNAL TABLEs	
Inline Views	Inline Views		
JSON	JSON	<ul style="list-style-type: none">Different paradigm and syntax	

Oracle Merge vs PostgreSQL Merge

Feature Compatibility	SCT/DMS Automation Level	SCT Action Code Index	Key Differences
		MERGE	MERGE is not supported by PostgreSQL, work-around available

Oracle Usage

The MERGE statement provides a means to specify single SQL statements that conditionally perform INSERT, UPDATE, or DELETE operations on a target table—a task that would otherwise require multiple logical statements.

The MERGE statement selects record(s) from the source table and then, by specifying a logical structure, automatically performs multiple DML operations on the target table. Its main advantage is to help avoid the use of multiple inserts, updates or deletes. It is important to note that MERGE is a deterministic statement. That is, once a row has been processed by the MERGE statement, it cannot be processed again using the same MERGE Statement. MERGE is also sometimes known as UPSERT.

Examples

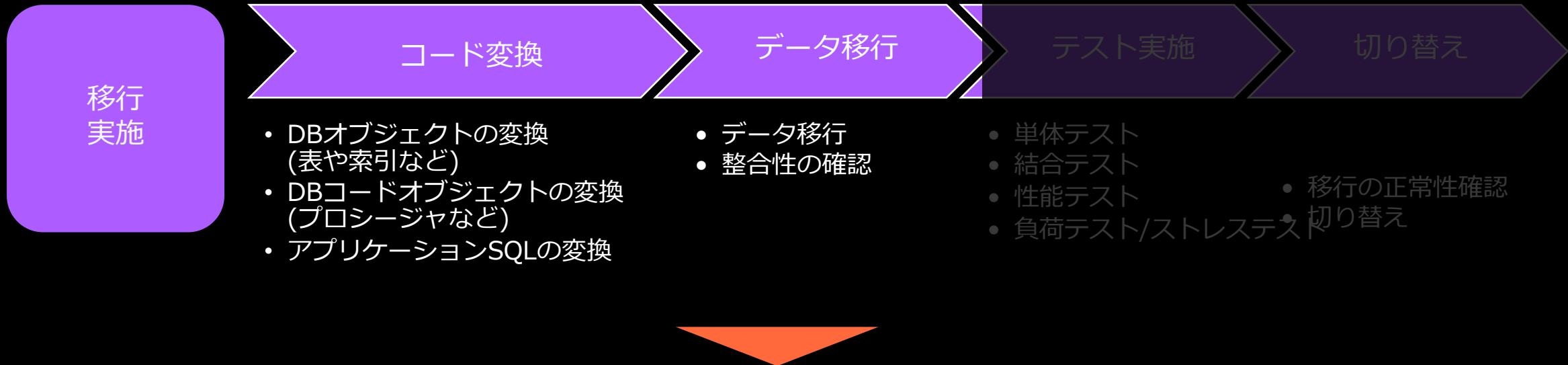
Use MERGE to insert or update employees who are entitled to a bonus (by year):

```
CREATE TABLE EMP_BONUS (EMPLOYEE_ID NUMERIC, BONUS_YEAR VARCHAR2(4),  
SALARY NUMERIC, BONUS NUMERIC, PRIMARY KEY (EMPLOYEE_ID, BONUS_YEAR));  
  
MERGE INTO EMP_BONUS B1  
USING (SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, DEPARTMENT FROM EMPLOYEES) E2 ON (B1.EMPLOYEE_ID = E2.EMPLOYEE_ID) WHEN MATCHED THEN  
UPDATE SET E1.BONUS = E2.SALARY * 0.5  
DELETE WHERE (E1.SALARY >= 10000)  
WHEN NOT MATCHED THEN  
INSERT (E1.EMPLOYEE_ID, E1.BONUS_YEAR, E1.SALARY, E1.BONUS)  
VALUES (E2.EMPLOYEE_ID, EXTRACT(YEAR FROM SYSDATE), E2.SALARY,  
E2.SALARY * 0.5)  
WHERE (E2.SALARY < 10000);  
  
SELECT * FROM EMP_BONUS;  
  
EMPLOYEE_ID BONUS-----  
103 2017      9000    4500  
104 2017      6000    3000  
105 2017      4800    2400
```

<https://aws.amazon.com/dms/resources/>

16

Auroraへのデータ移行

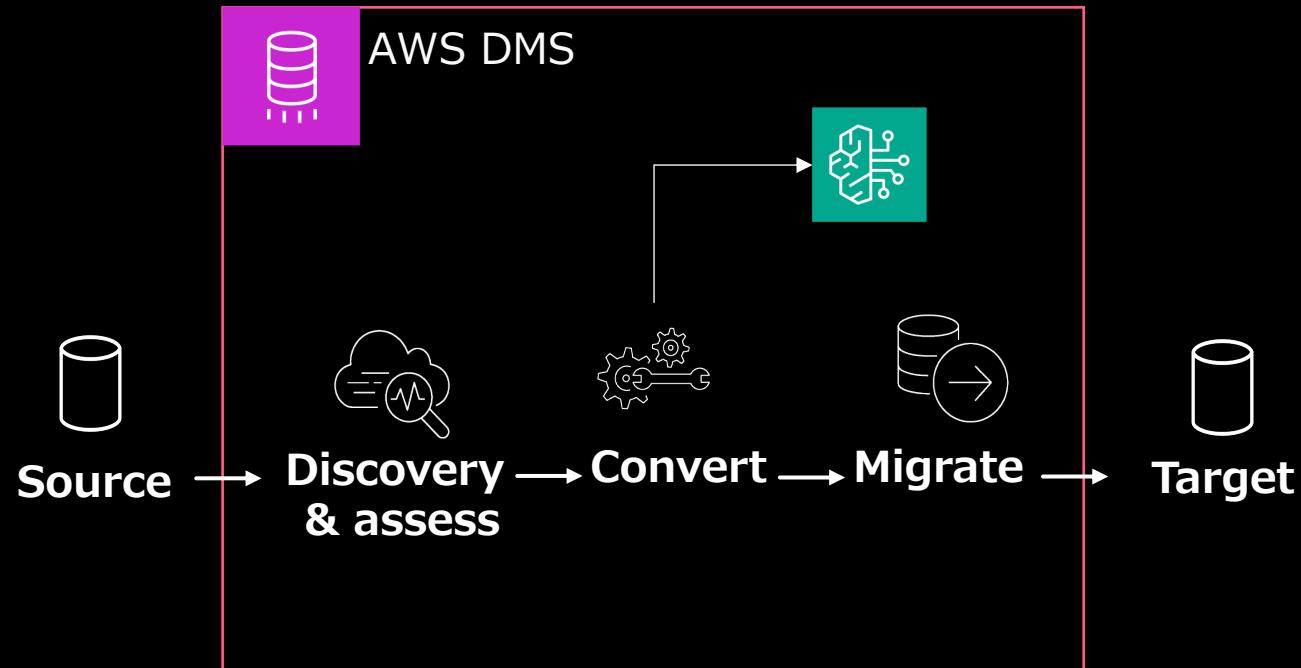


AWS Database Migration Service (DMS)

最小限のダウンタイムで異種DB間のデータ移行を実現



AWS DMS の用途



フルロード

- 特定時点のデータ断面を取得、ターゲットに対してロード

Change Data Capture (CDC)

- 特定時点以降にソースで発生した更新差分を、トランザクションログを元にDMLを作成し、ターゲットに適用

データ検証

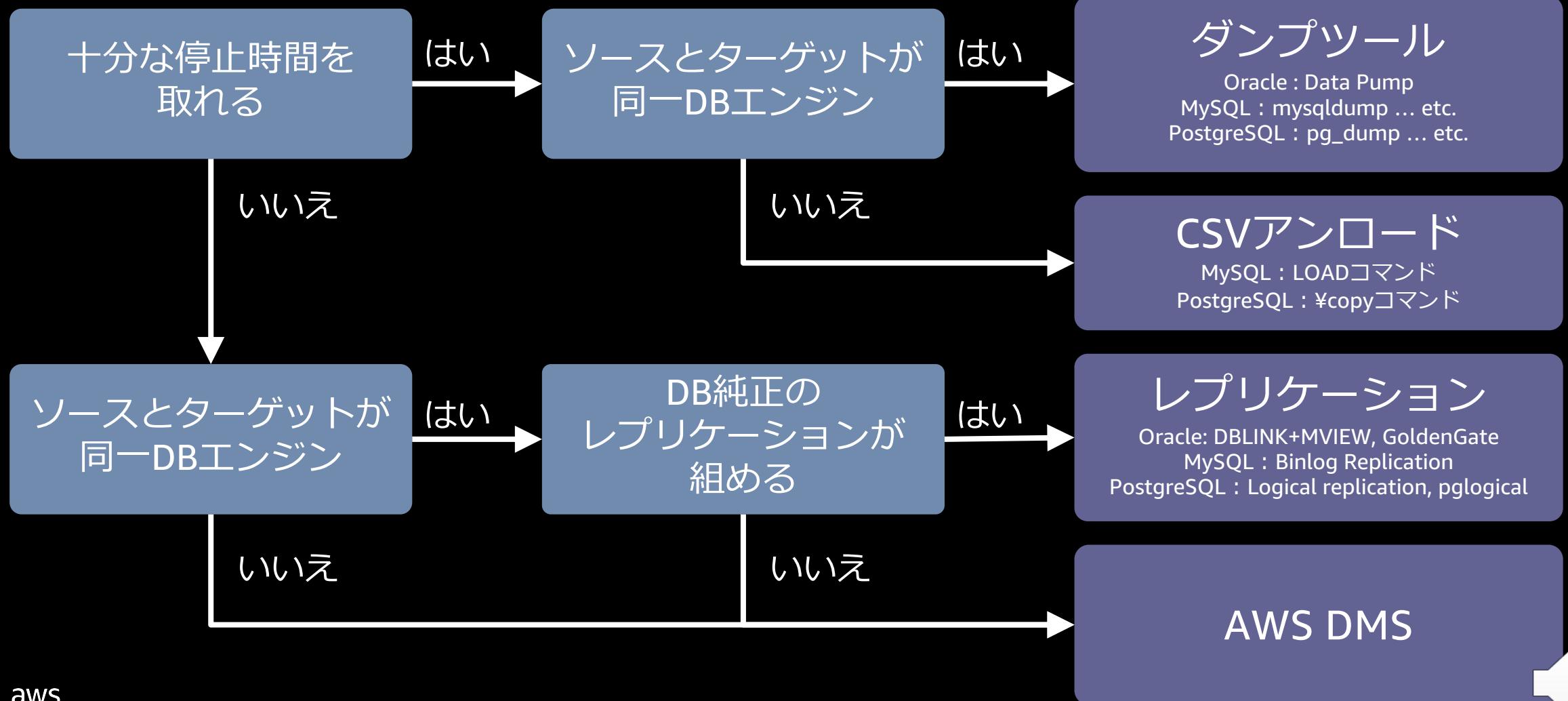
- データ移行完了後にソースとターゲットのデータを突合

スキーマ変換 **New!**

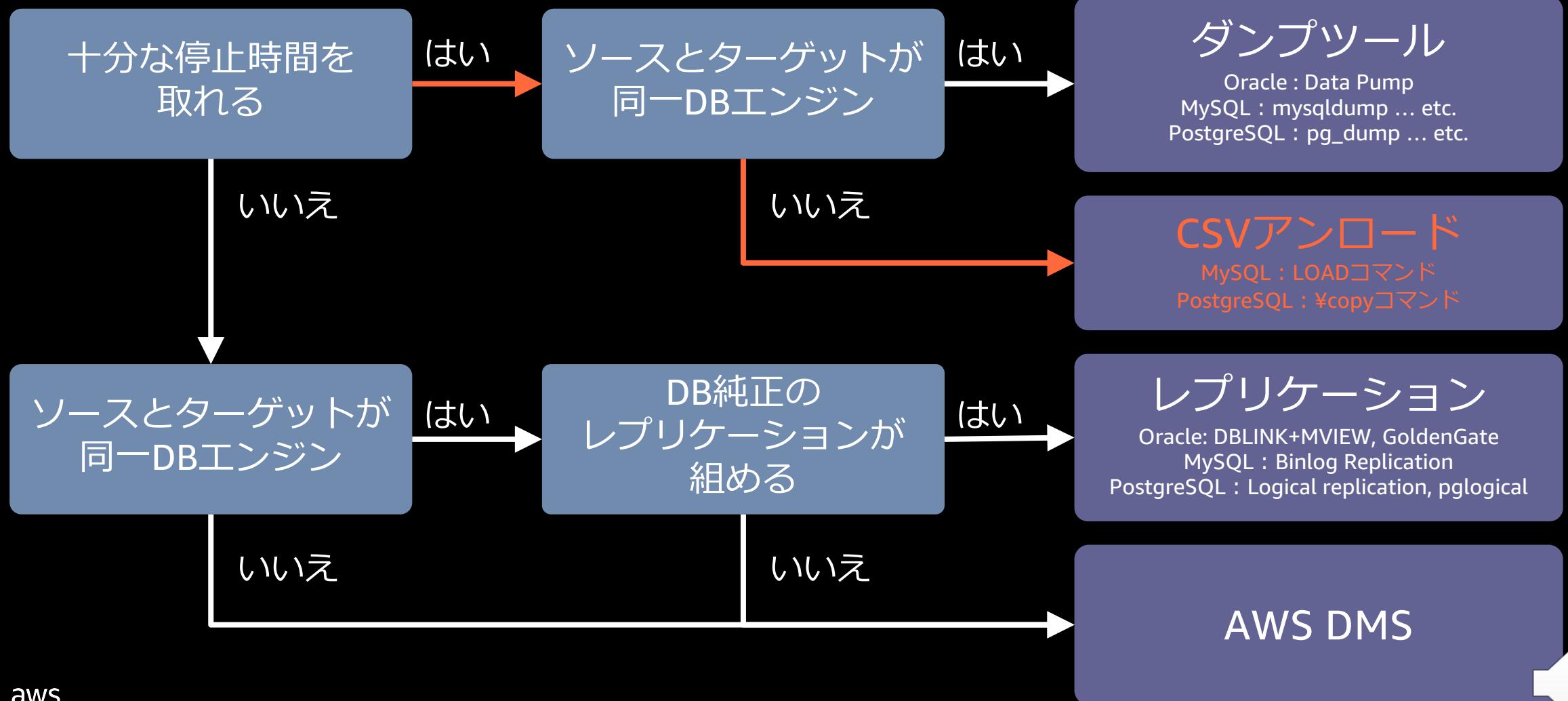
- 従来SCTと同様に、データベース内のオブジェクトの自動変換を実施
- その際に生成AIを活用



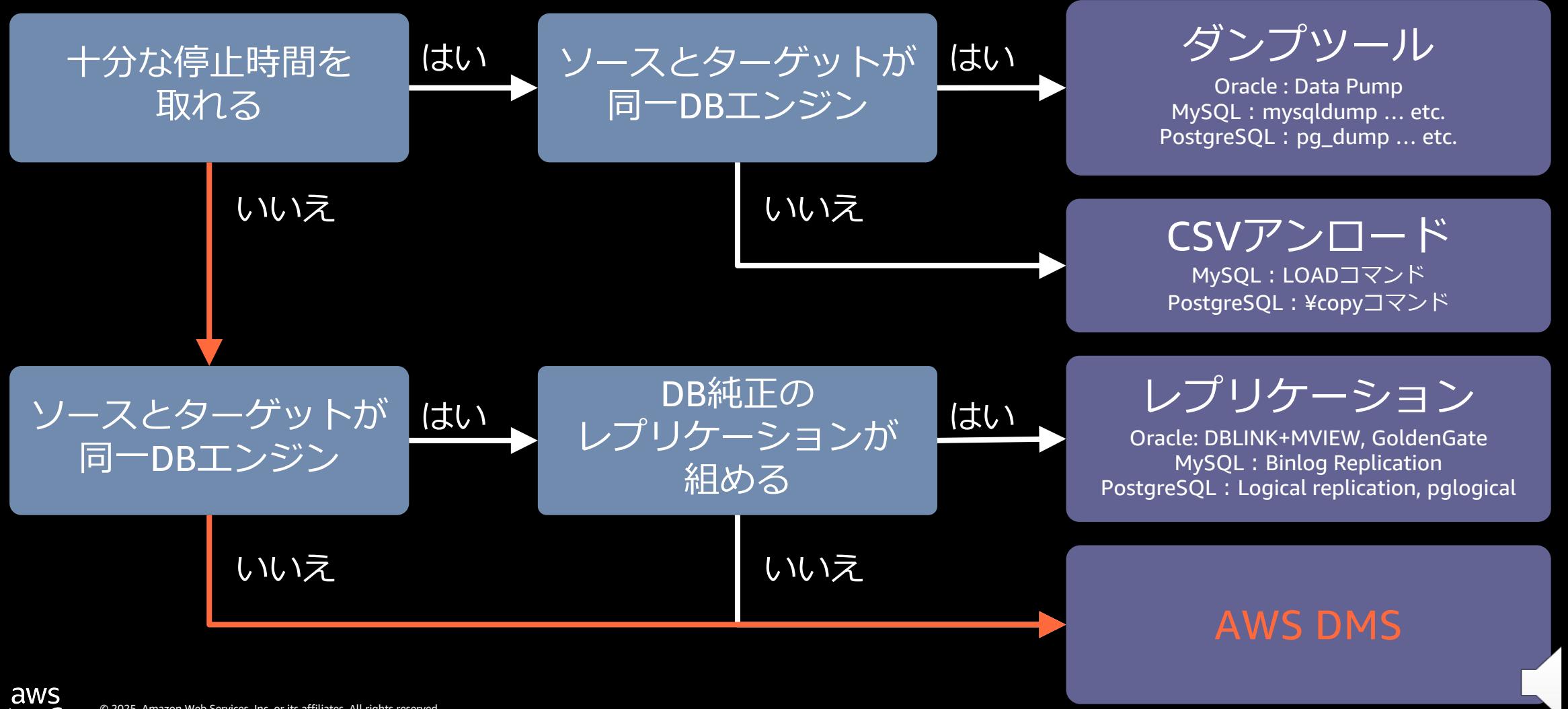
データ移行の選択肢と選定ポイント



データ移行の選択肢と選定ポイント



データ移行の選択肢と選定ポイント



アジェンダ

1. Aurora の概要と移行における課題
2. 移行支援プログラム & 移行サービス
 - 異種DBエンジンからの移行
 - 同種DBエンジンからの移行



同種DBエンジンから Aurora へのデータ移行方式

ソースの場所	データベース	推奨オプション
<ul style="list-style-type: none">・ オンプレミス・ EC2	<ul style="list-style-type: none">・ MySQL, MariaDB・ PostgreSQL	<ul style="list-style-type: none">✓ ダンプツールの利用✓ 必要に応じてレプリケーションを併用✓ DMS Homogeneous data migrationsを利用
・ RDS	<ul style="list-style-type: none">・ MySQL, MariaDB・ PostgreSQL	<ul style="list-style-type: none">✓ DBスナップショットから直接移行✓ リードレプリカをAurora側に作成



オンプレ / EC2 上の MySQL からの移行

ダンプツールを利用した移行

- mysqldump / MySQL Shell / Percona XtraBackup

https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Migrating.html

レプリケーションを併用することでダウンタイムを削減

- Binlog Replication

https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Replication.MySQL.html

DMS Homogeneous data migrationsを利用した移行

<https://docs.aws.amazon.com/dms/latest/userguide/dm-migrating-data-mysql.html>

- ネイティブのダンプツール、レプリケーション機能を利用し、フルロード、CDCを実行
- データ検証機能がないなど、一部制限事項あり



オンプレ / EC2 上の PostgreSQL からの移行

ダンプツールを利用した移行

- pg_dump / pg_dumpall / pg_restore / psql

<https://repost.aws/ja/knowledge-center/aurora-postgresql-migrate-from-rds>

レプリケーションを併用することでダウンタイムを削減

- ネイティブLogical replication / pglogical
 - Aurora は物理レプリケーション（ストリーミングレプリケーション）は未サポート

DMS Homogeneous data migrations を利用した移行

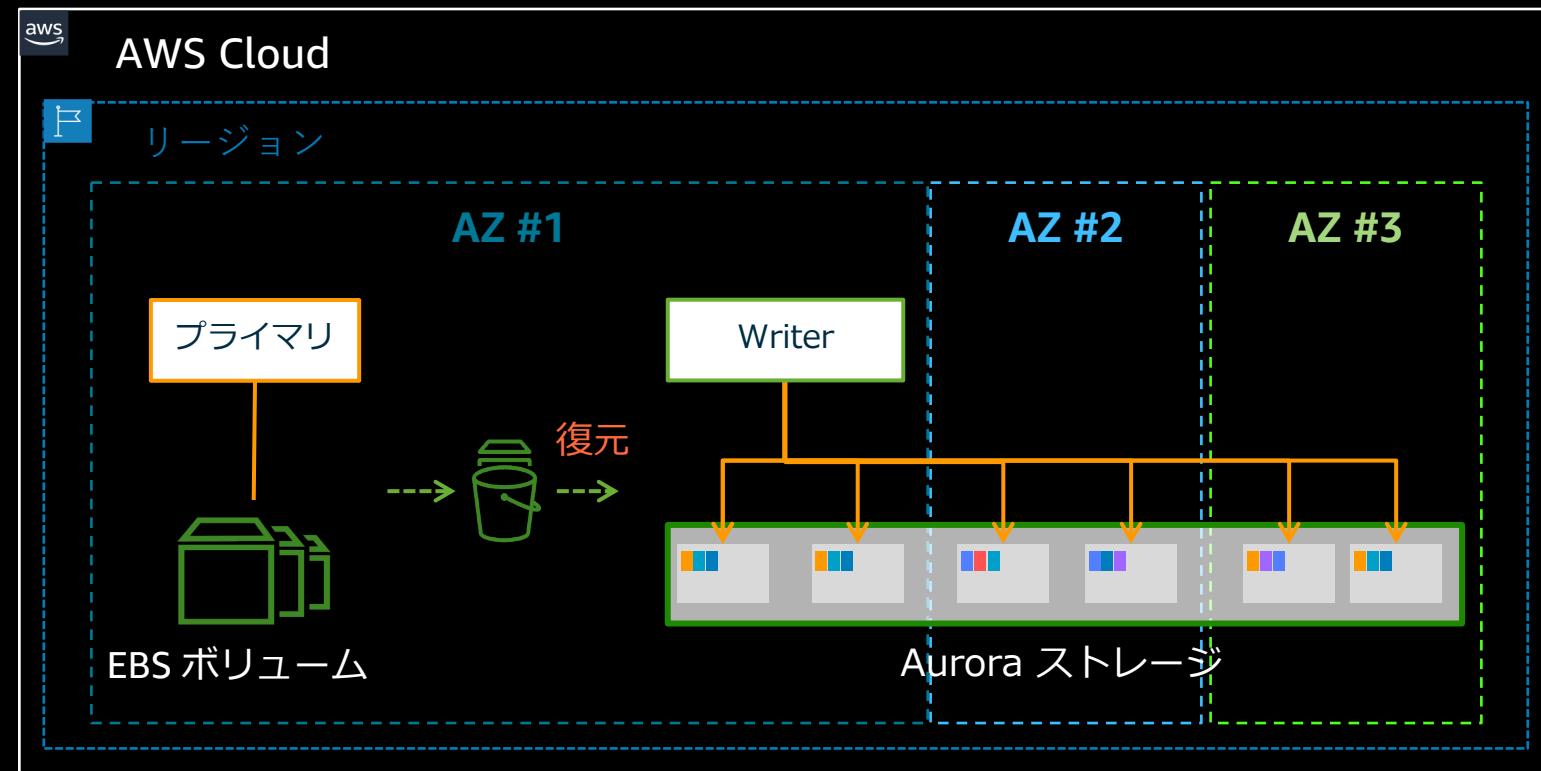
<https://docs.aws.amazon.com/dms/latest/userguide/dm-migrating-data-postgresql.html>

- ネイティブのダンプツール、レプリケーション機能を利用し、フルロード、CDCを実行
- データ検証機能がないなど、一部制限事項あり

RDS for MySQL, PostgreSQLからの移行

RDS スナップショットから Aurora クラスタを作成

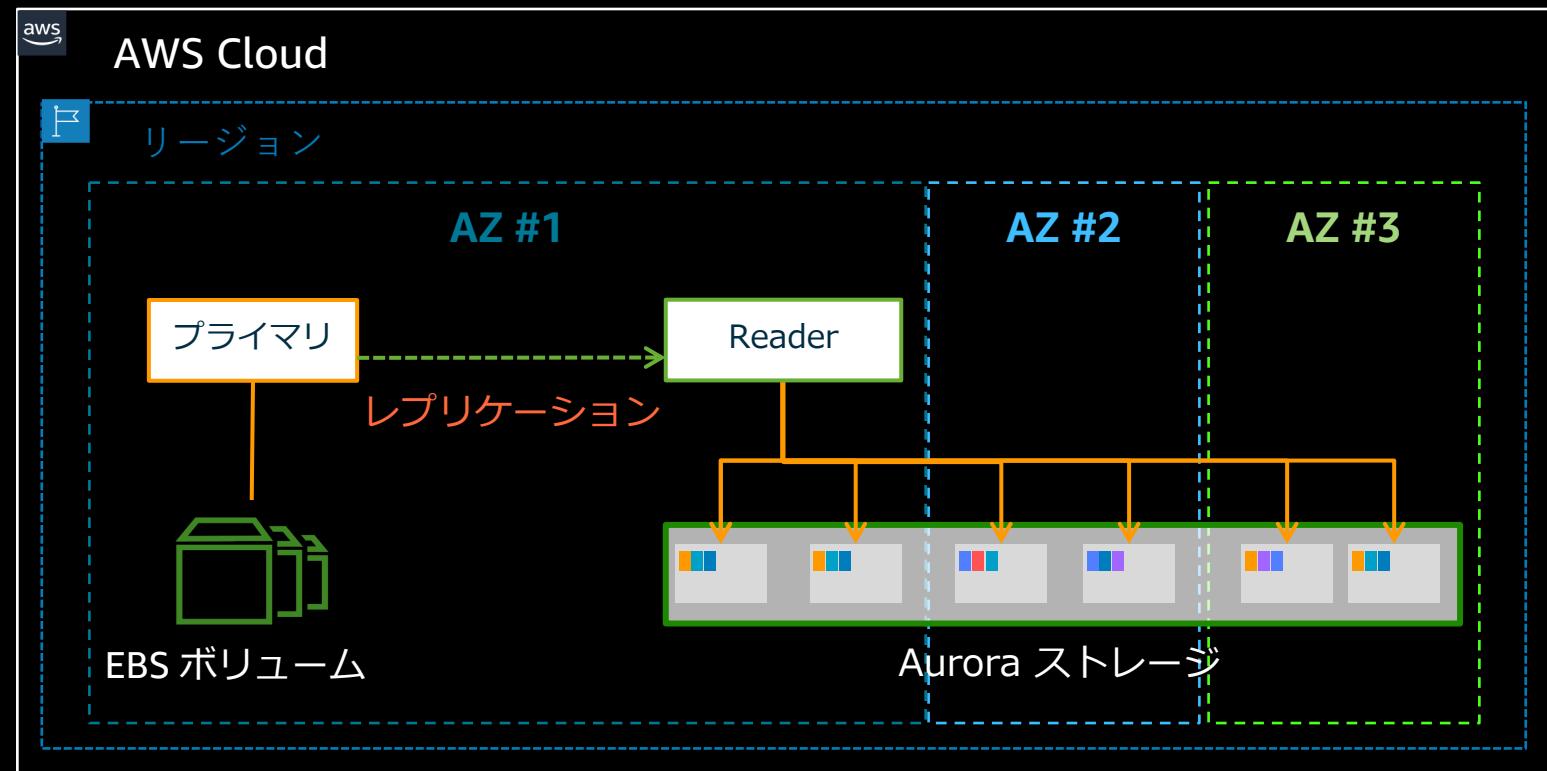
- ・ シンプルに構成可能
- ・ スナップショットから復元するために、一定のダウントIMEを要する



RDS for MySQL, PostgreSQLからの移行

RDS のリードレプリカを Aurora クラスタとして作成

- ・ レプリケーションしながら、最終的に切り離すことでダウンタイムを極小化



まとめ

- ✓ AWSでは、既存データベースを Aurora へ移行する際の支援策として、様々なプログラムやサービスを用意しています
- ✓ データベースのクラウド移行を検討されている方は、AWSの担当営業までお問い合わせください



Thank you!

AWS Black Belt Online Seminar

Amazon Aurora の 性能とスケーラビリティ

鈴木 大樹 (Daiki Suzuki)

Solutions Architect

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=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>



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



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



内容についての注意点

- ・ 本資料では2025年6月時点のサービス内容および価格についてご説明しています。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

鈴木 大樹 (Daiki Suzuki)

アマゾンウェブサービスジャパン
ソリューションアーキテクト

Web 業界の BtoC 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



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



アジェンダ

1. Amazon Aurora の性能
2. Amazon Aurora のスケーラビリティ
3. Amazon Aurora Serverless V2 の概要
4. まとめ



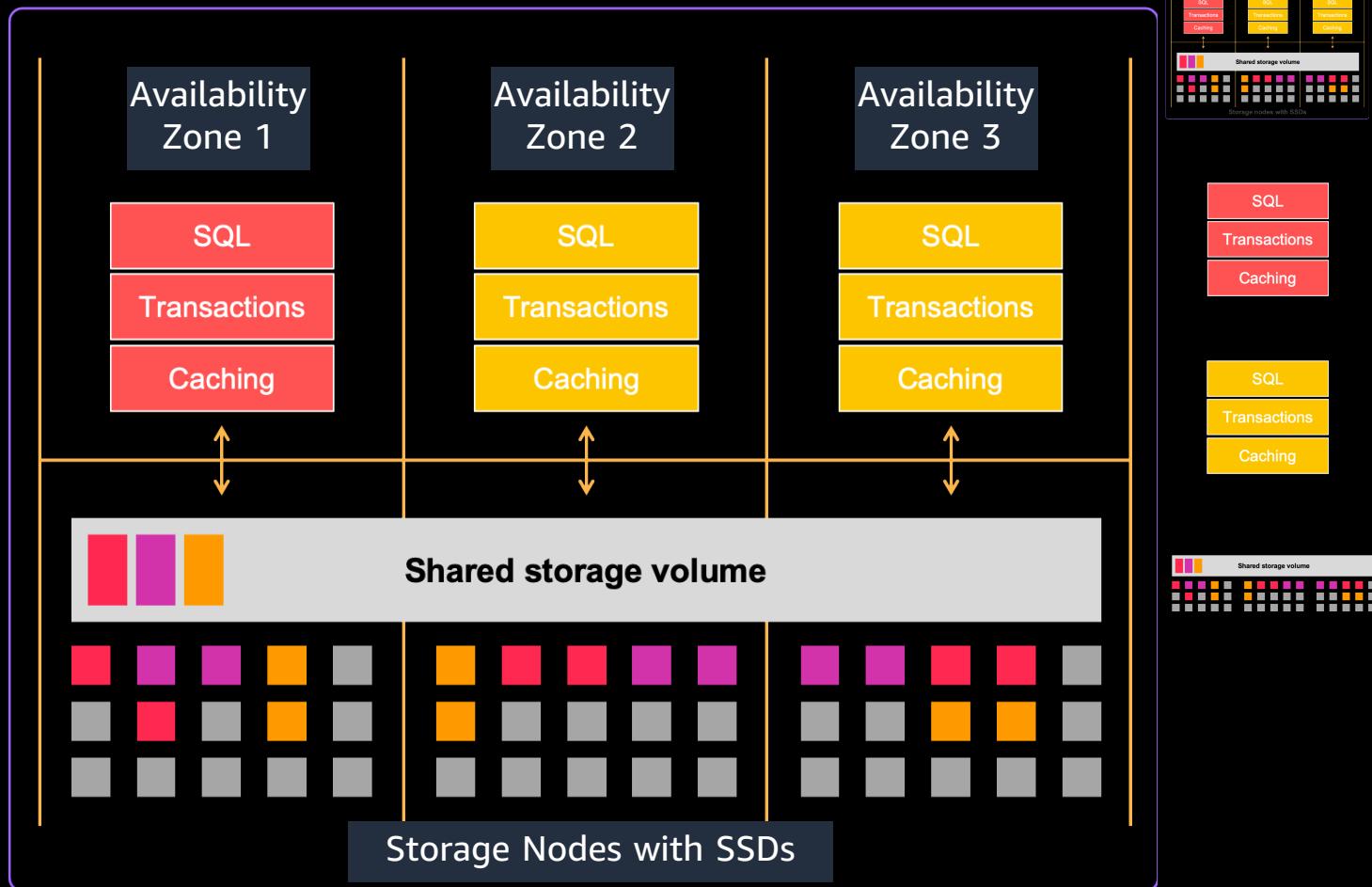
Amazon Aurora の性能



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



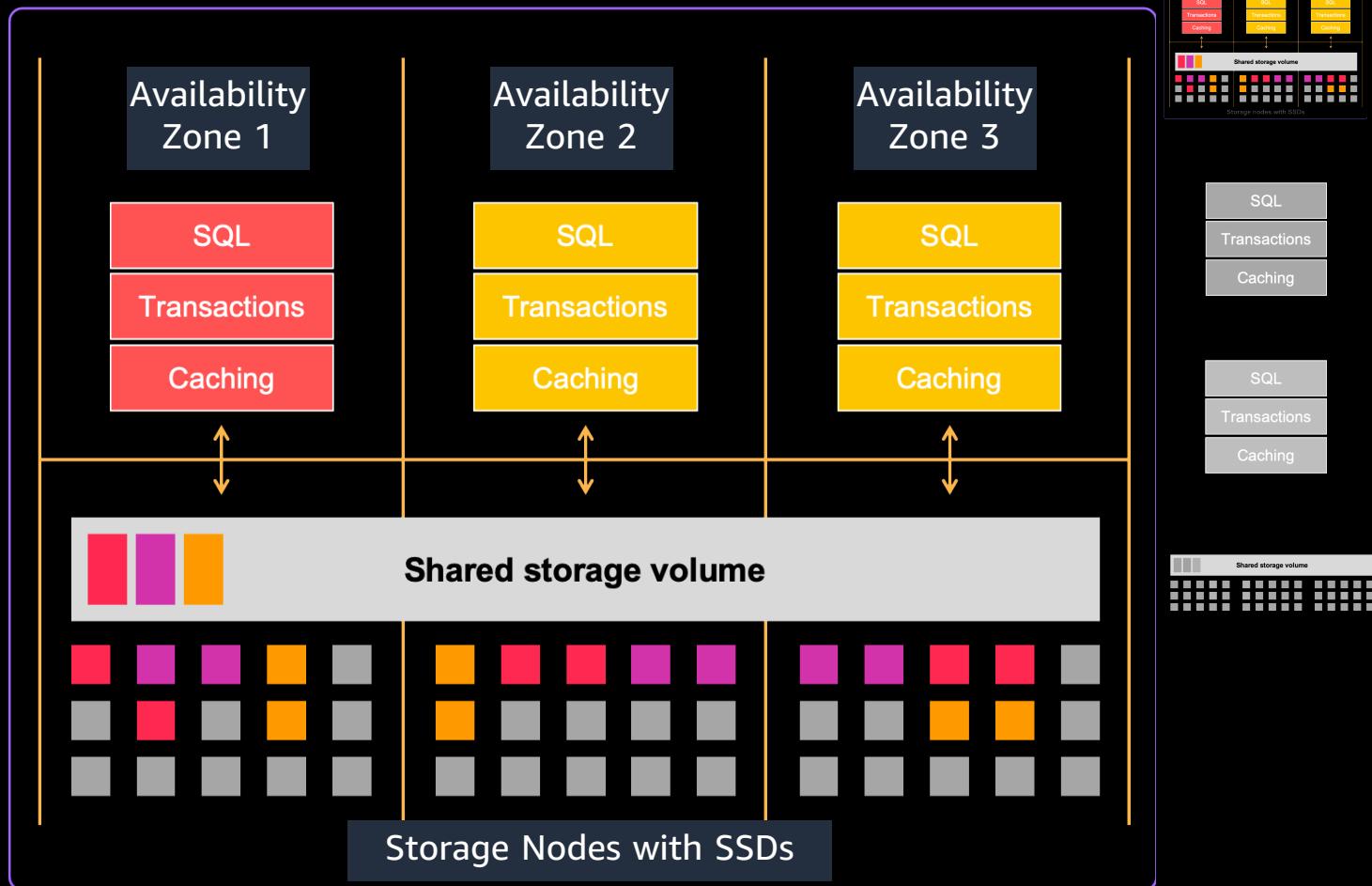
Auroraを構成するコンポーネント



- **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 DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(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(Aurora 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
 - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
 - Aurora の接続先を示す URL



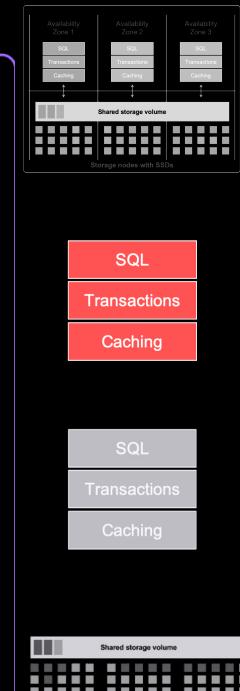
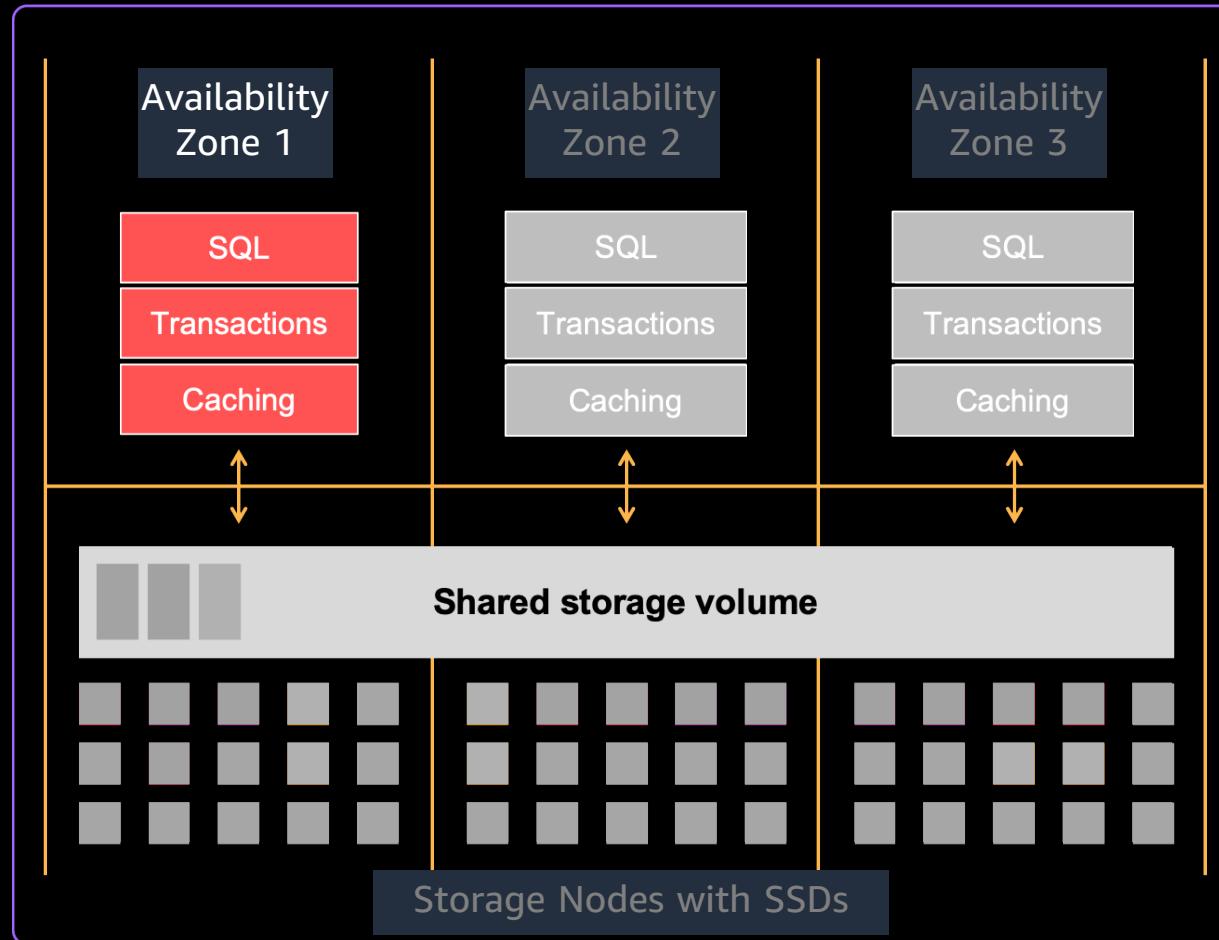
Auroraを構成するコンポーネント



- **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 DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(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(Aurora 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
 - Writer も Reader も同じクラスター ボリュームを利用
- **Aurora エンドポイント**
 - Aurora の接続先を示す URL

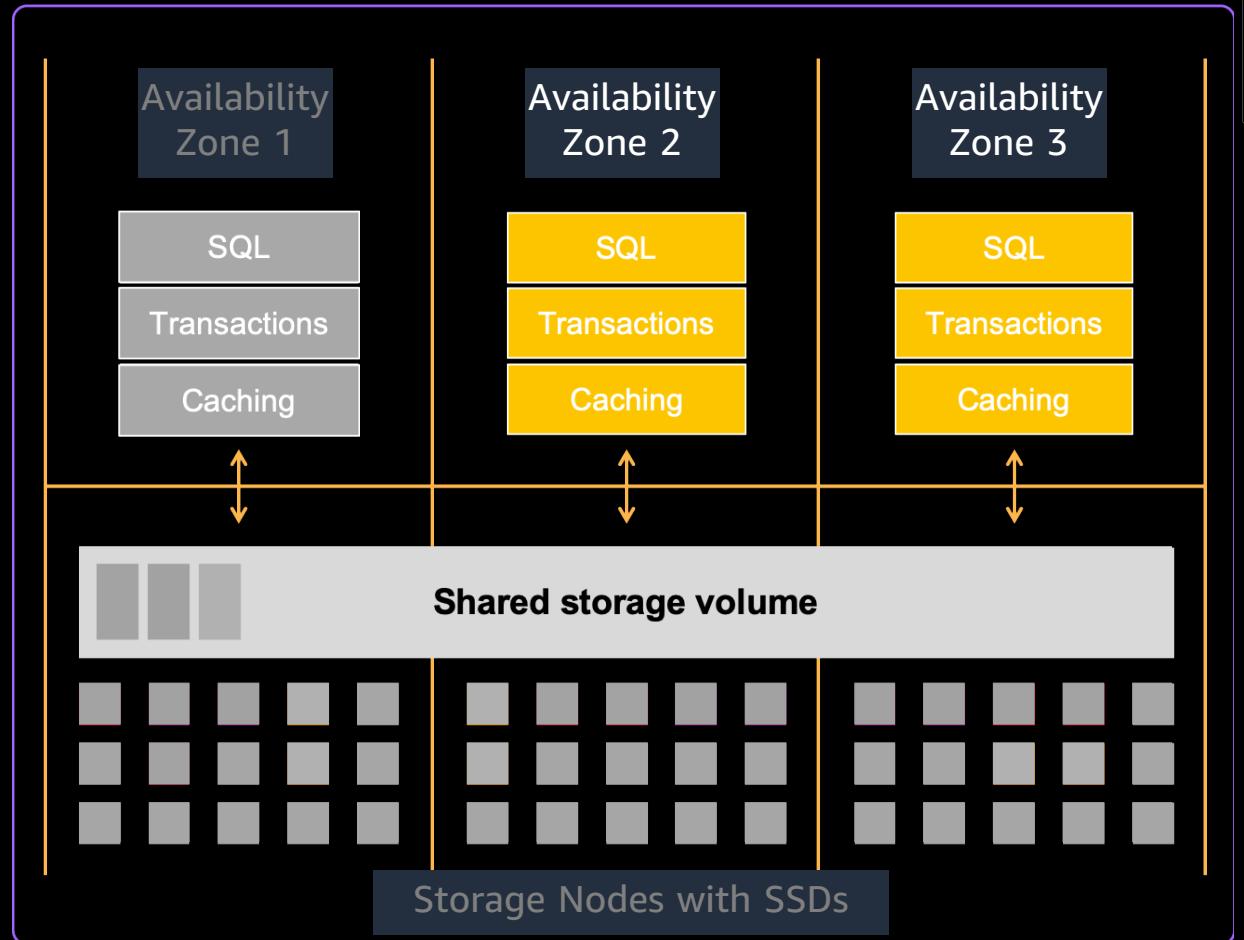


Auroraを構成するコンポーネント



- **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 DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(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(Aurora 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
 - Writer も Reader も同じクラスター ボリュームを利用
- **Aurora エンドポイント**
 - Aurora の接続先を示す URL

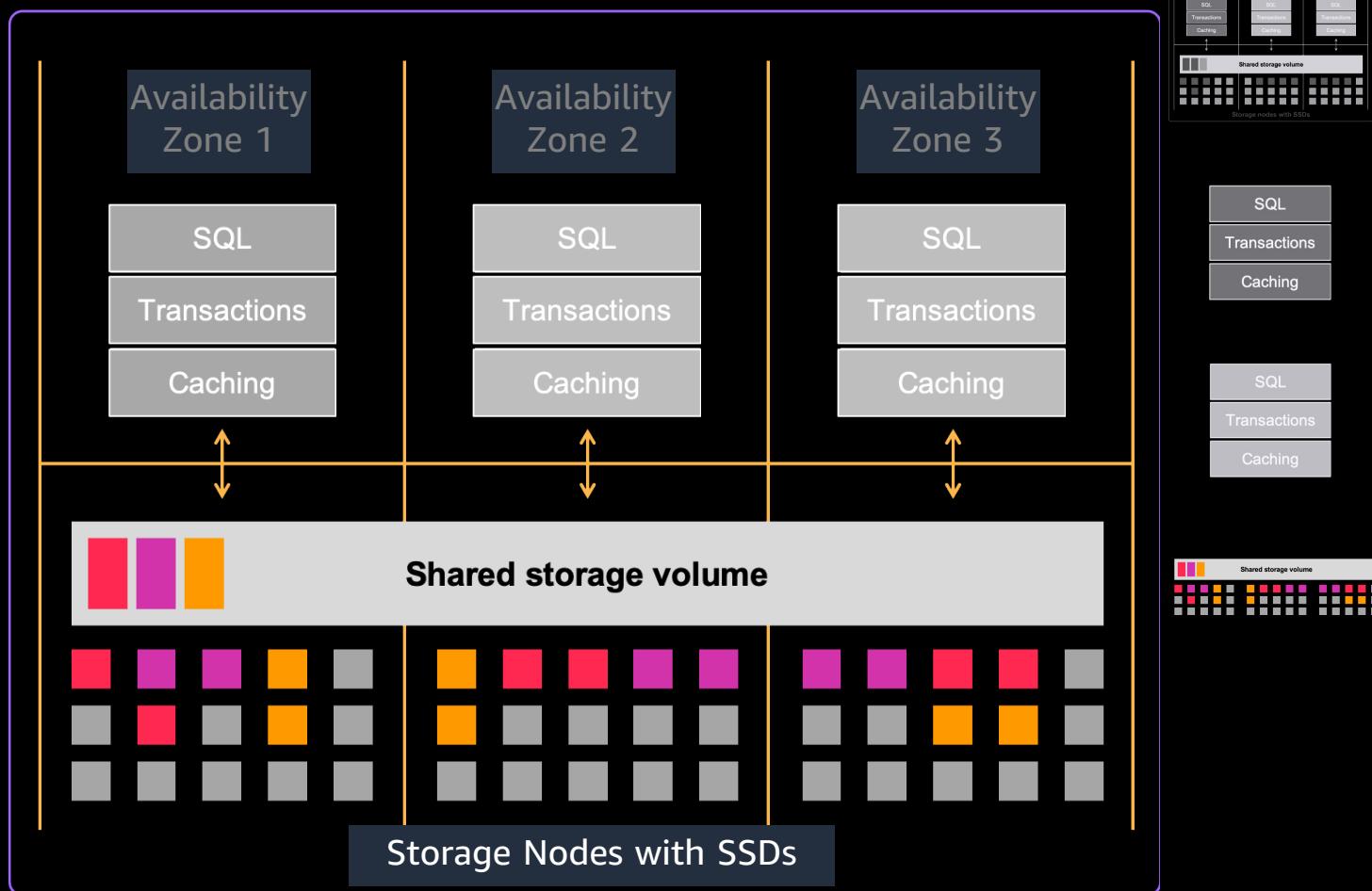
Auroraを構成するコンポーネント



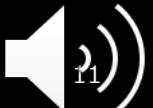
- **Amazon Aurora DB クラスター**
 - Amazon Aurora の管理単位
 - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(Reader)**
 - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム(Aurora ストレージ)**
 - 3つの AZ 間でレプリケートされる仮想ボリューム
 - Writer も Reader も同じクラスターボリュームを利用
- **Aurora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



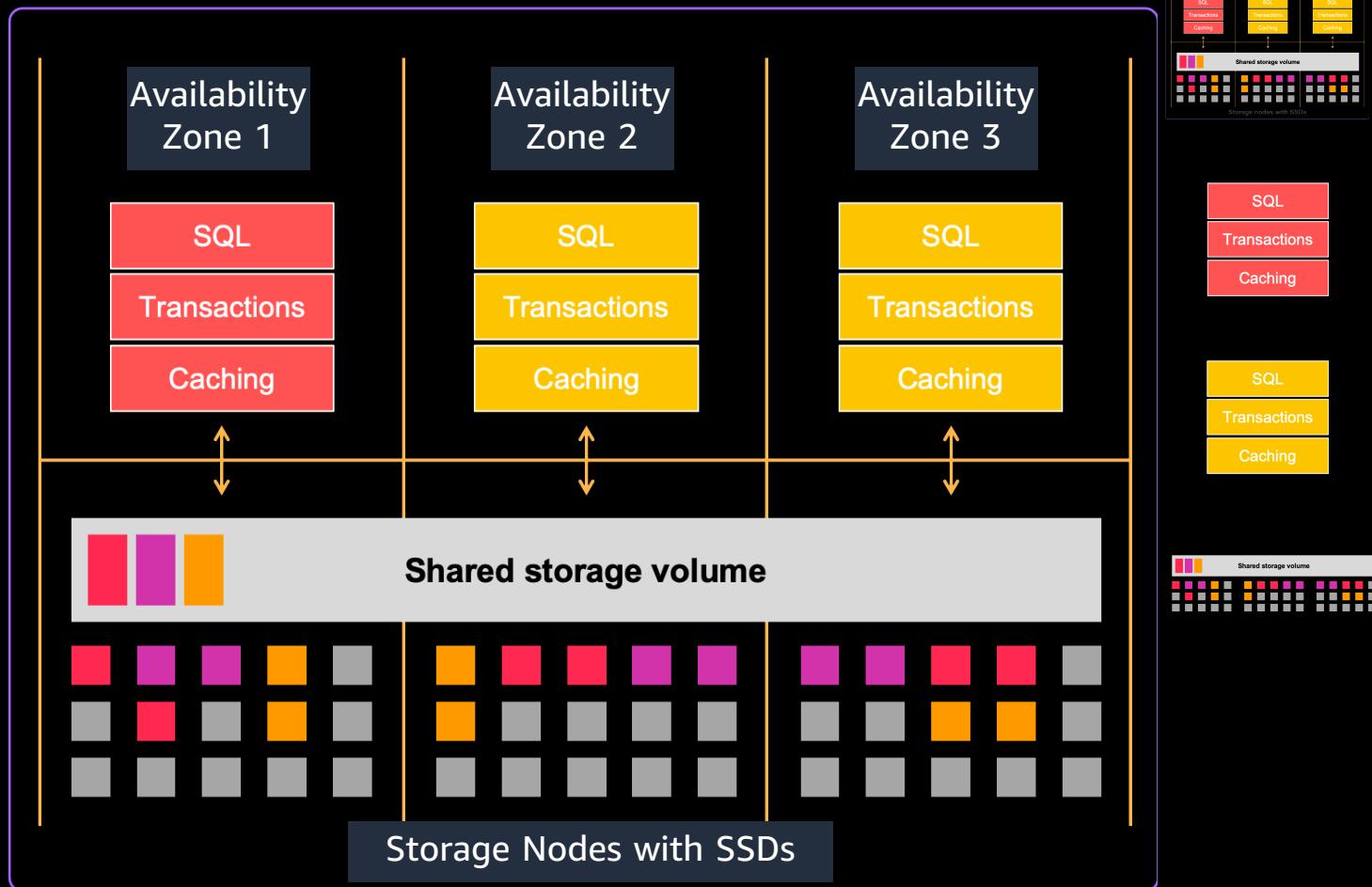
Auroraを構成するコンポーネント



- **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 DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(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(Aurora 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
 - Writer も Reader も同じクラスター ボリュームを利用
- **Aurora エンドポイント**
 - Aurora の接続先を示す URL



Auroraを構成するコンポーネント

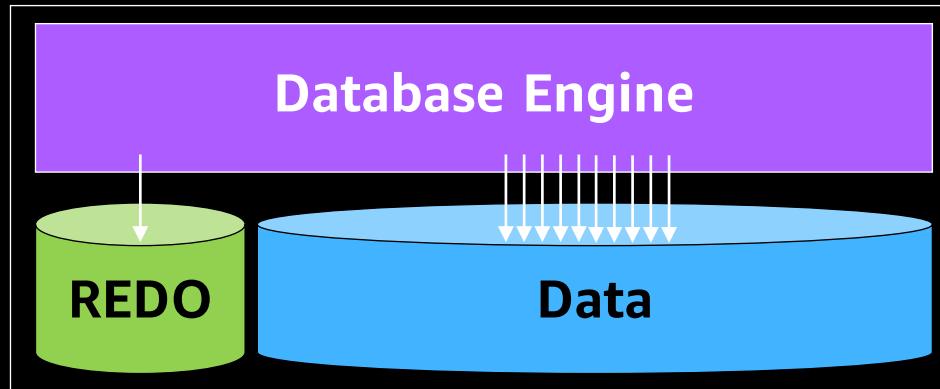


- **Amazon Aurora DB クラスター**
 - Amazon Aurora の管理単位
 - プライマリインスタンス、レプリカ、クラスター ボリュームの総称
- **プライマリ DB インスタンス(Writer)**
 - 読み込み、書き込みを行なうインスタンス
- **Aurora リードレプリカ(Reader)**
 - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスター ボリューム(Aurora ストレージ)**
 - 3つの AZ 間でレプリケートされる仮想ボリューム
 - Writer も Reader も同じクラスター ボリュームを利用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

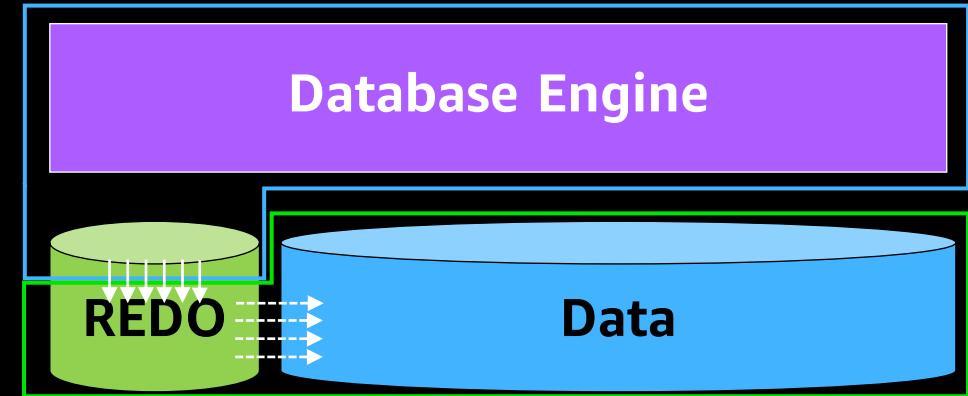


Aurora のストレージ: チェックポイント不要

Traditional Database

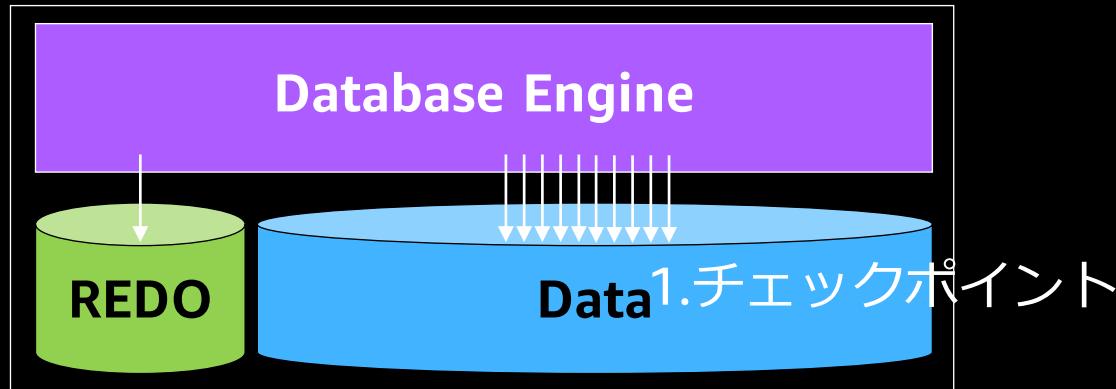


Amazon Aurora

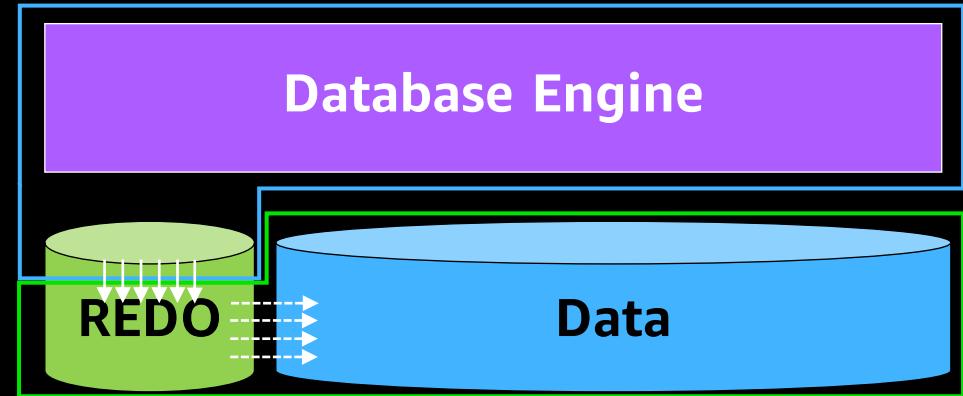


Aurora のストレージ: チェックポイント不要

Traditional Database

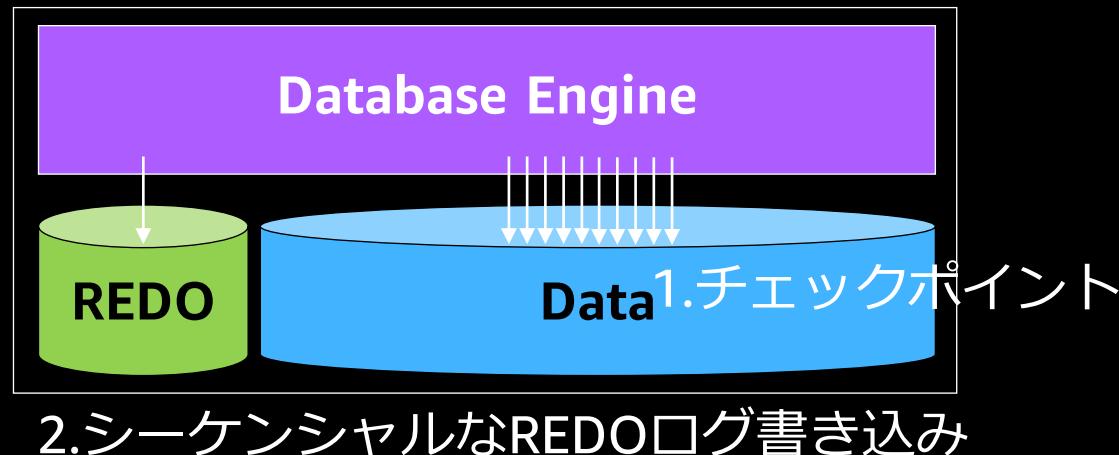


Amazon Aurora

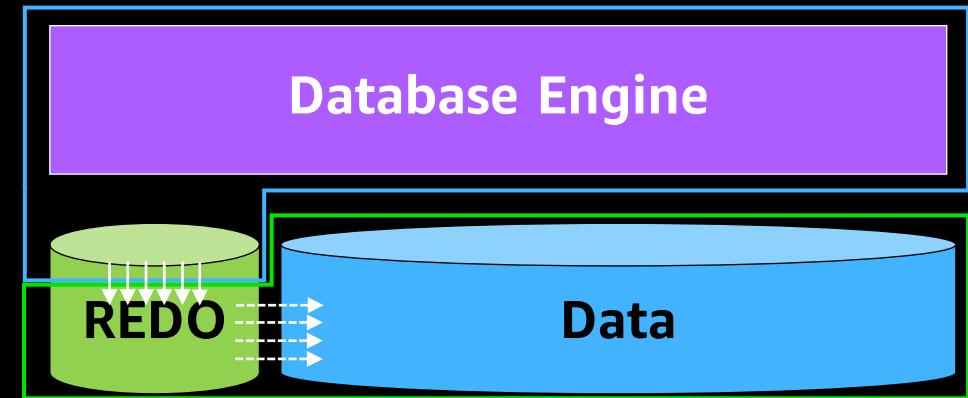


Aurora のストレージ: チェックポイント不要

Traditional Database

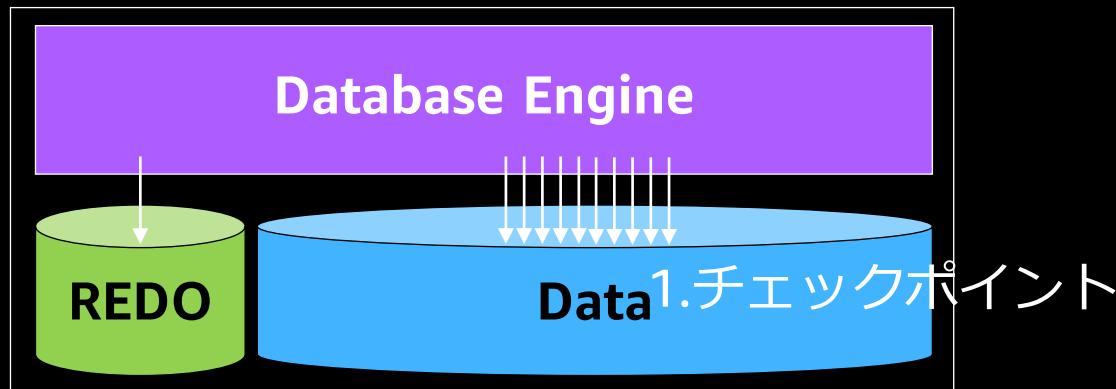


Amazon Aurora

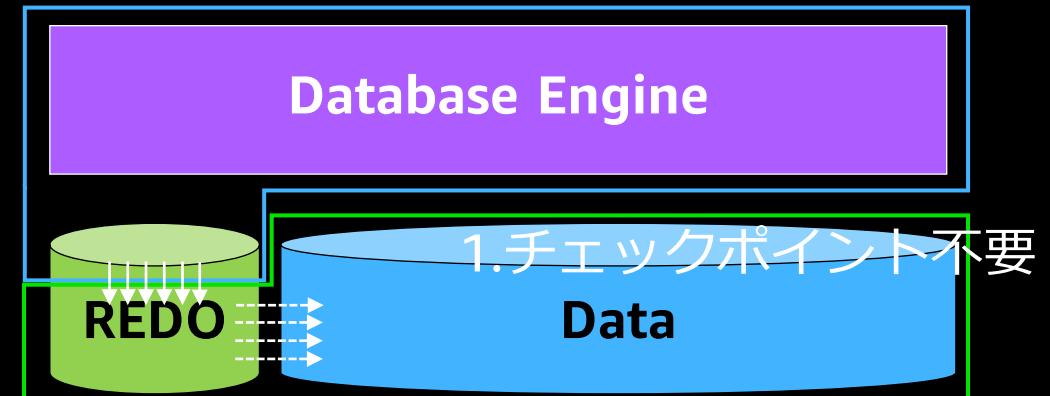


Aurora のストレージ: チェックポイント不要

Traditional Database



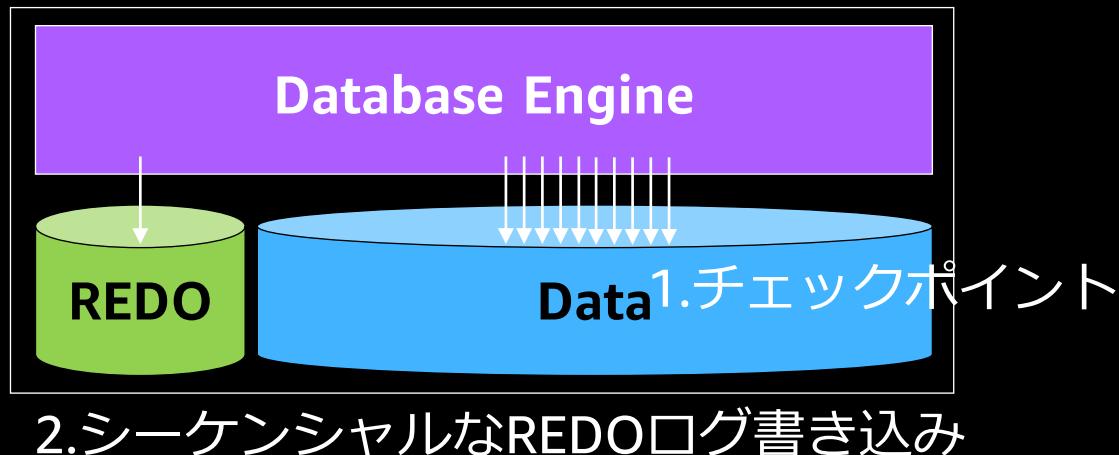
Amazon Aurora



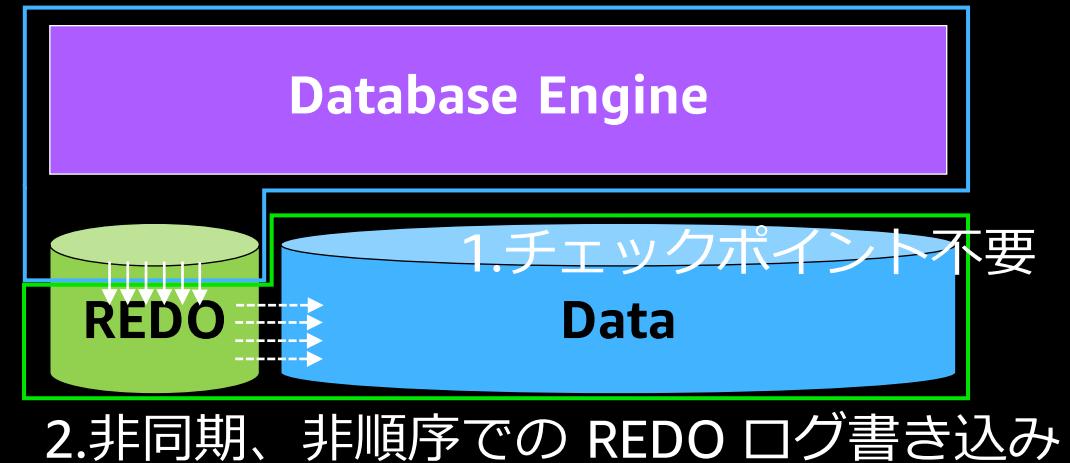
1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)

Aurora のストレージ: チェックポイント不要

Traditional Database



Amazon Aurora

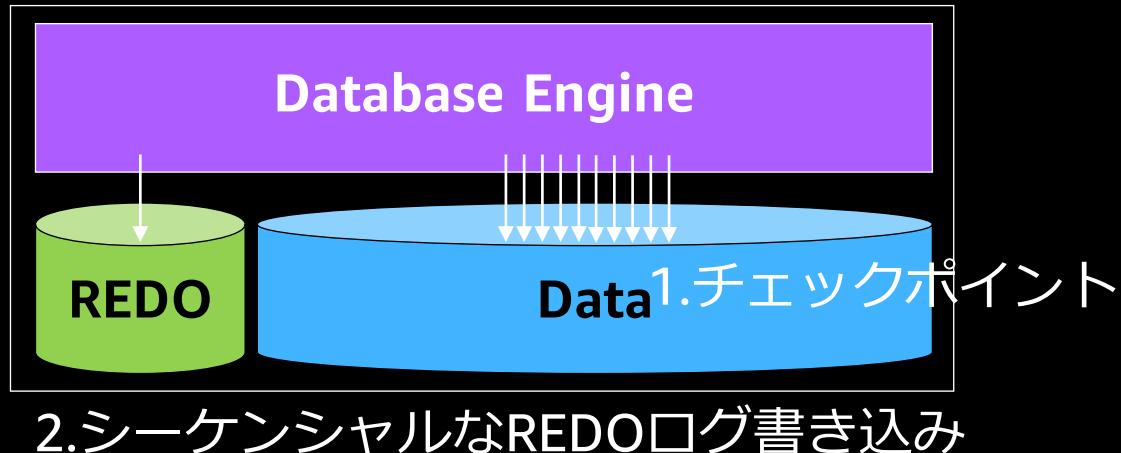


1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)
2. 非同期、非順序でのログ書き込み (ログ書き込みのスループットの向上)

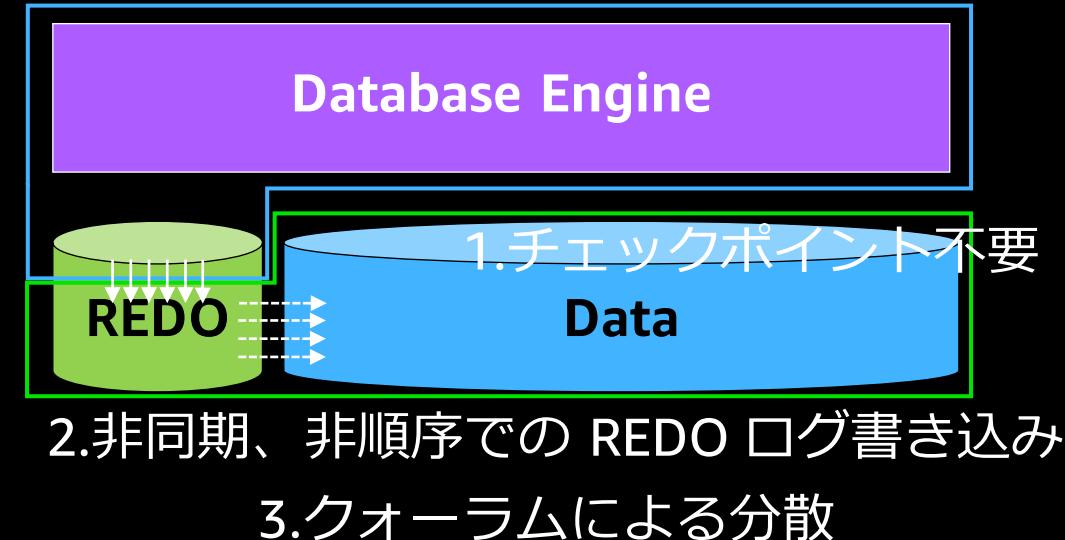


Aurora のストレージ: チェックポイント不要

Traditional Database



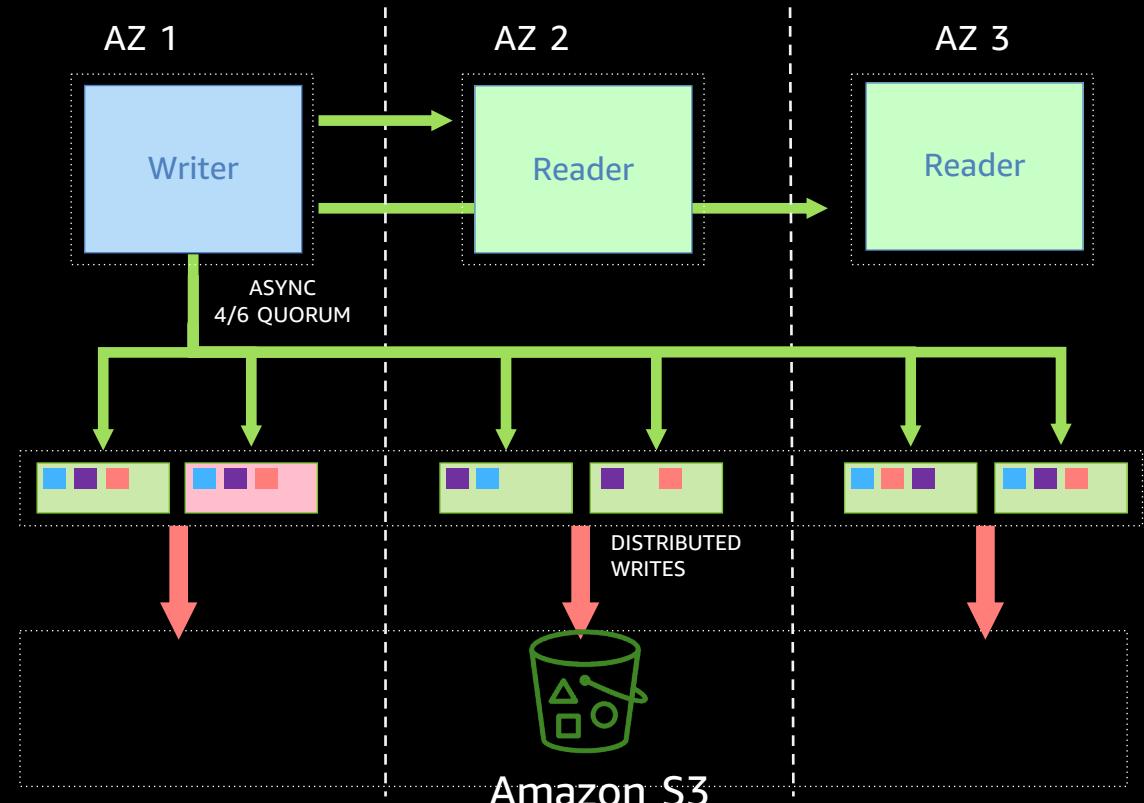
Amazon Aurora



1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)
2. 非同期、非順序でのログ書き込み (ログ書き込みのスループットの向上)
3. Heavy weight な分散合意アルゴリズムが不要(クオーラムによる Durability Tracking)

IO traffic in Aurora (PostgreSQL)

AMAZON AURORA

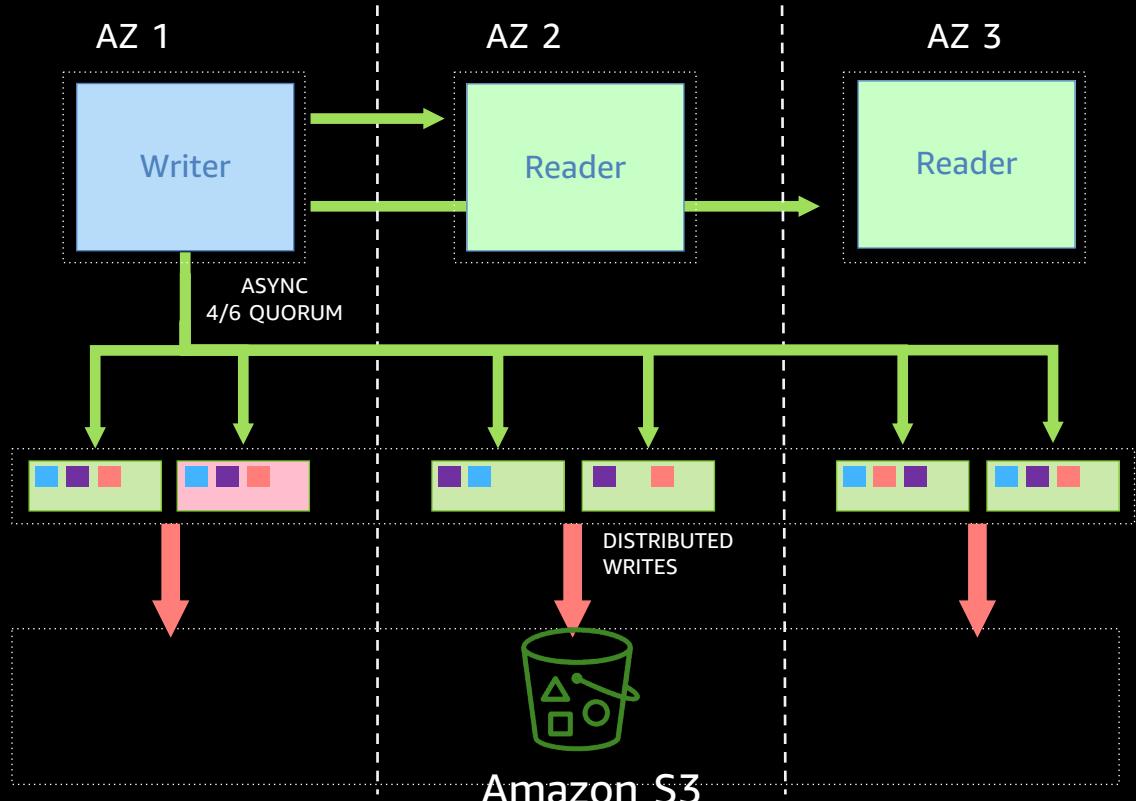


TYPE OF WRITE



IO traffic in Aurora (PostgreSQL)

AMAZON AURORA



IO FLOW

REDO ログレコードをまとめる - 完全に LSN 順に並ぶ
適切なセグメントに分割する - 部分ごとに並ぶ
ストレージノードへまとめて書き込む

→ LOG RECORD

TYPE OF WRITE

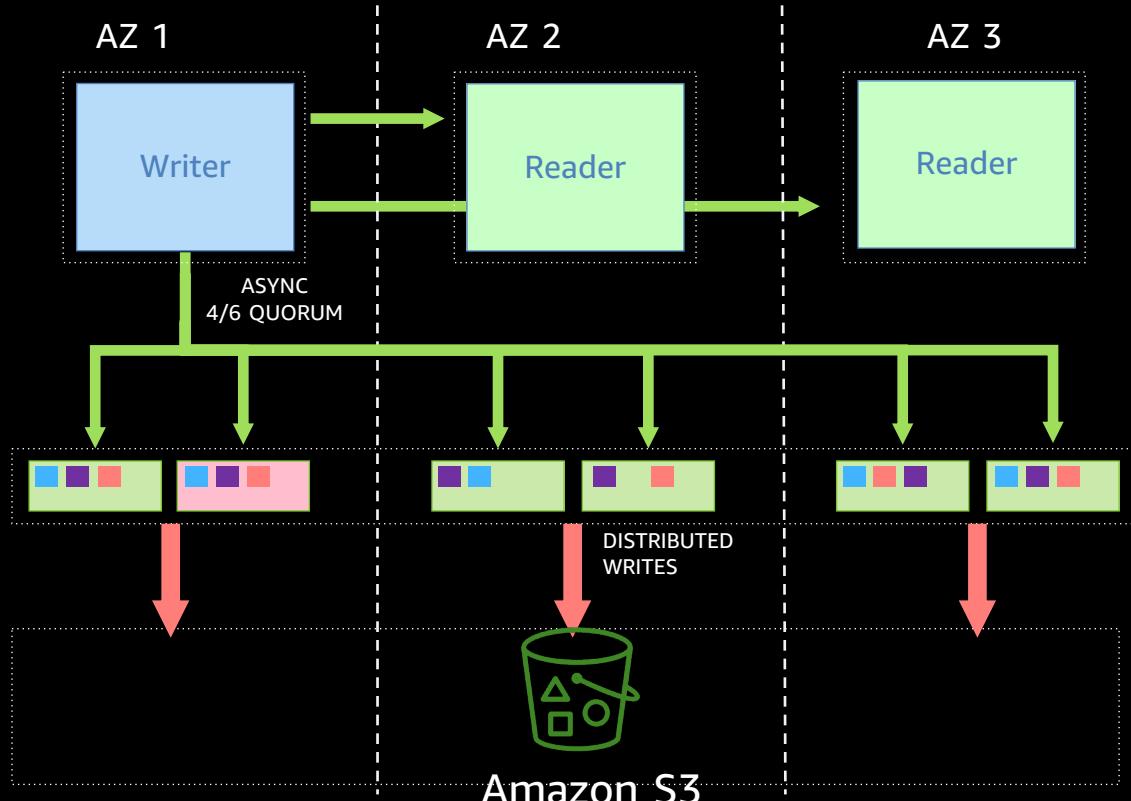
■■■ COMMIT LOG & FILE

→ DATA



IO traffic in Aurora (PostgreSQL)

AMAZON AURORA



IO FLOW

REDO ログレコードをまとめる - 完全に LSN 順に並ぶ
適切なセグメントに分割する - 部分ごとに並ぶ
ストレージノードへまとめて書き込む

OBSERVATIONS

REDO ログレコードのみ書き込む; 全てのステップは非同期
データブロックは書かない(チェックポイント, キャッシュ置換時)
6 倍のログ書き込みだが、**1/9**のネットワークトラフィック
ネットワークとストレージのレイテンシー異常時の耐性

→ LOG RECORD

TYPE OF WRITE

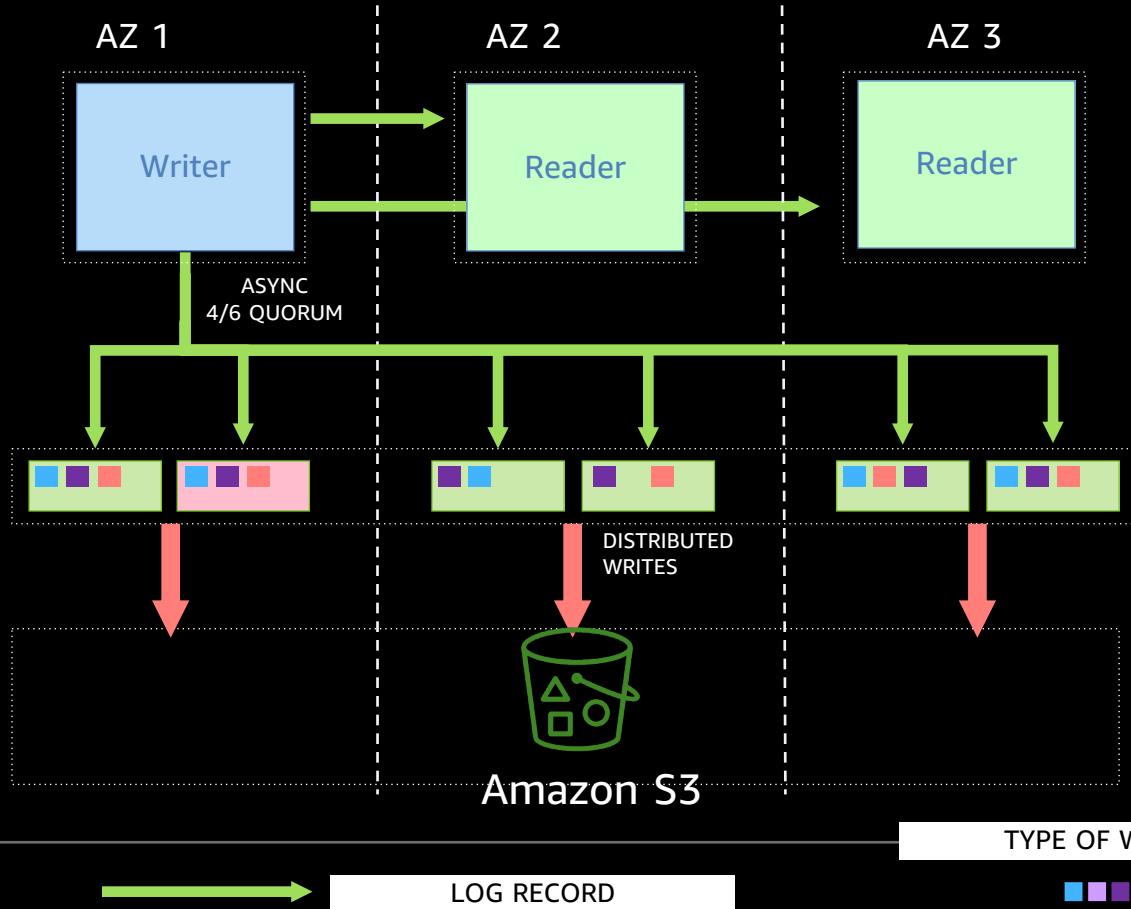
■■■ COMMIT LOG & FILE

→ DATA



IO traffic in Aurora (PostgreSQL)

AMAZON AURORA



IO FLOW

REDO ログレコードをまとめる - 完全に LSN 順に並ぶ
適切なセグメントに分割する - 部分ごとに並ぶ
ストレージノードへまとめて書き込む

OBSERVATIONS

REDO ログレコードのみ書き込む; 全てのステップは非同期
データブロックは書かない(チェックポイント, キャッシュ置換時)
6 倍のログ書き込みだが、1/9 のネットワークトラフィック
ネットワークとストレージのレイテンシー異常時の耐性

PERFORMANCE

write-only もしくは、read/write が混在するワークロードにて、PostgreSQL のコミュニティエディションに比べて、
2 倍以上の性能を発揮



一般的なクオーラムモデルの問題

クオーラムモデル:

- 複数のデータコピーのうち一定数に書き込みまたは読み込みできた場合にその処理を完了とし、それ以外のデータコピーは非同期で整合性をとる

読み込み/書き込みコスト

- ネットワーク負荷
(大きなデータページが流れることにより、ネットワークへの負荷が増大)
- 複数のストレージに対する読み込み
(3/6 読み込みクオーラムであれば、3つのコピーを読み込む必要あり)

コスト

- 6つのコピーからなるクオーラムであれば、物理ストレージ容量が6倍必要

一般的なクオーラムモデルの問題

読み込み/書き込みコスト

- ・ ネットワーク負荷
(大きなデータページが流れることにより、ネットワークへの負荷が増大)
→ Amazon Aurora では、ログレコードのみがストレージノードに送信されるので、
ネットワーク負荷を削減
(データページ全体が流れるわけではない)
- ・ 複数のストレージに対する読み込み
(3/6 読み込みクオーラムであれば、3つのコピーを読み込む必要あり)
→ クオーラムにおける読み込みを回避

コスト

- ・ 6つのコピーからなるクオーラムであれば、物理ストレージ容量が6倍必要
→ 完全なデータページを構成すべきコピーは3つとして物理容量を削減
(お客様へのコストという意味では実際に使った容量のみ(3倍や6倍になるわけではない))

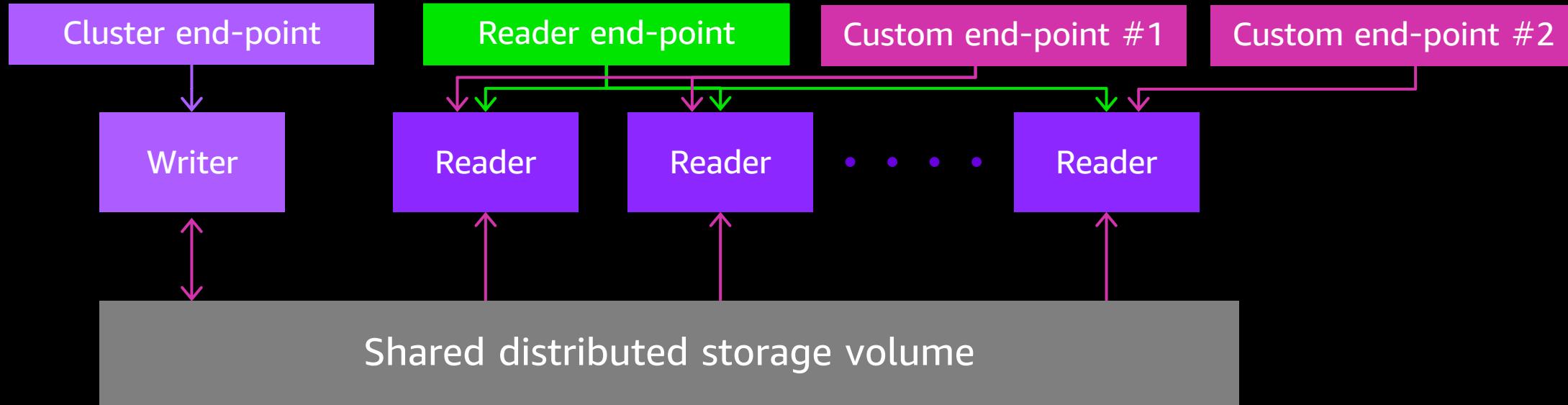
Amazon Aurora の スケーラビリティ



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



読み込みをスケールアウトさせる Reader



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 Reader を作成可能

Reader は Auto Scaling による自動増減が可能

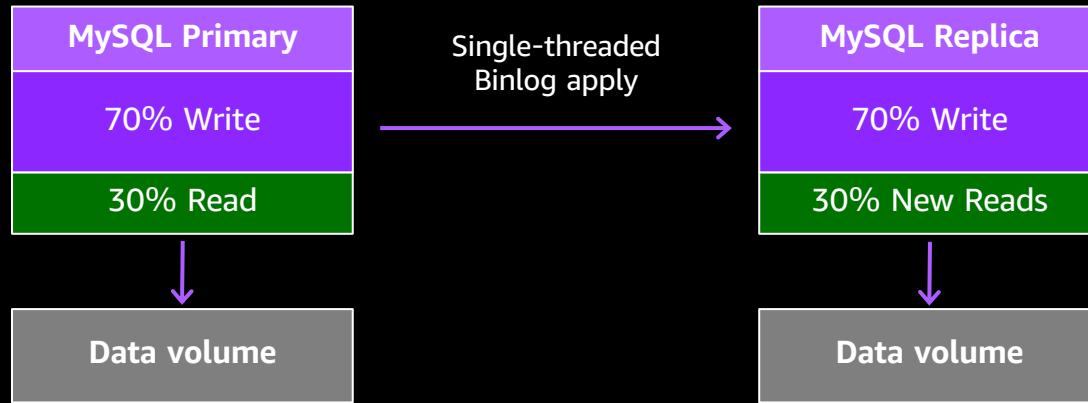
低遅延な REDO ログベース(物理)レプリケーション(通常20~40ミリ秒のレプリカラグ)

フェイルオーバーの順序を構成可能なカスタムリーダーエンドポイント

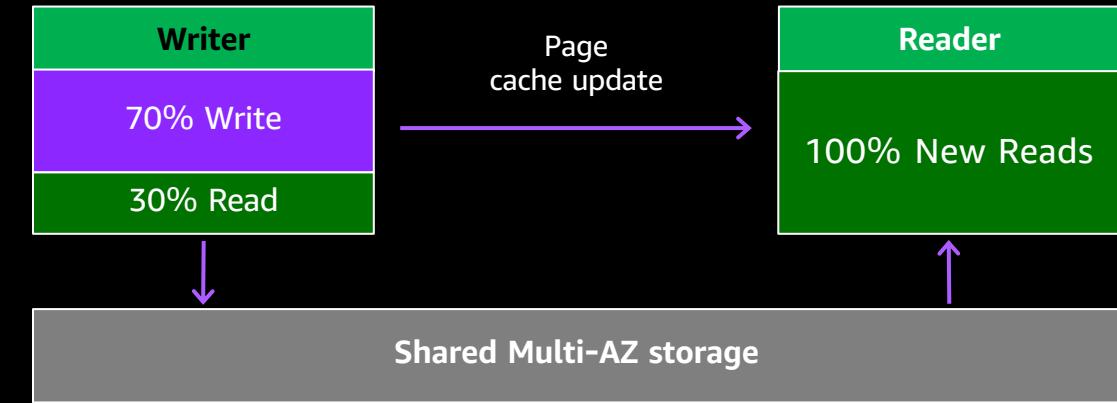
Writer と Reader はカスタムエンドポイントの一部になることが可能

通常のレプリカと Reader の違い

MySQL/PostgreSQL read scaling



Amazon Aurora read scaling



差分変更を用いた論理レプリケーション

プライマリと同等の書き込みワークロード

独立したストレージ

差分変更を用いた物理レプリケーション

Reader に書き込みは発生しない

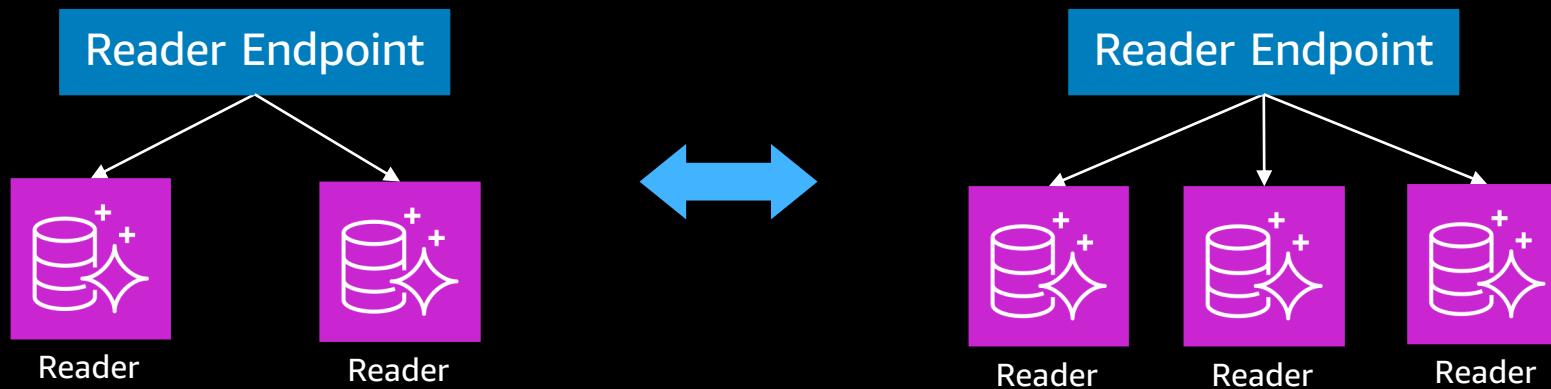
共有ストレージ

Reader での Auto Scaling

- 平均 CPU 使用率・平均接続数に応じて 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
 - Reader エンドポイント・カスタムエンドポイントは Reader の自動的な追加・削除に対応
 - Cooldown Period や Min/Max Capacity を設定可能

注意点

- 追加されるのは Primary 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)



自動管理された Aurora ストレージ



最大 128TB のストレージ – 10GB 単位で自動拡張

- 最大 128TB の自動ストレージスケーリング - パフォーマンスへの影響なし
 - 自動拡張 / 自動縮小
 - Amazon S3 への継続的な増分バックアップ
 - ユーザーのスナップショットを即座に作成 - パフォーマンスへの影響なし
 - 自動再ストライピング、ミラー修復、ホットスポット管理、暗号化

再起動だけで変更可能な様々なインスタンスタイプ

T3、T4g（開発用）

- 2vCPU, 2GB RAMから2vCPU, 8GB RAM
- バースト可能、t4g は Graviton2 を搭載

R5、R6i、R6g、R7g、R7i、R8g（メモリ最適化）

- 2vCPU, 16GB RAM から 192vCPU, 1,536GB RAM
- R6g、R7g は Graviton2、R8g は Graviton4 を搭載

X2g（メモリ最適化）

- 2vCPU, 32GB RAMから64vCPU, 1,024GB RAM
- X2gは Graviton2 を搭載

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.DBInstanceClass.html#Concepts.DBInstanceClass.Types>



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



Amazon Aurora Serverless v2 概要



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



31)

Amazon 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 (ギガバイト) 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

以下 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)

バースト可能クラス (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) <input type="text" value="0"/> 0~256 (0.5 の倍数)	最大キャパシティ (ACU) <input type="text" value="64"/> 1~256 (0.5 の倍数) (128 GiB)
---	---

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

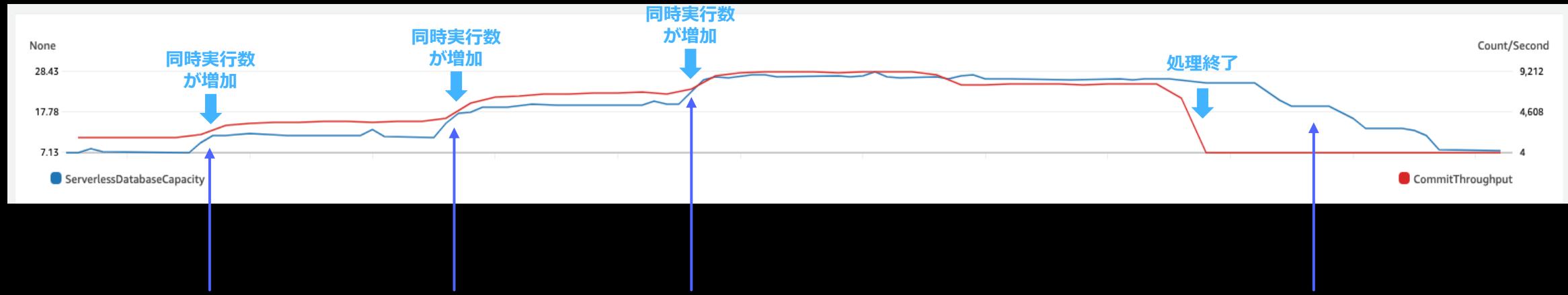
00:05:00
5 分から 24 時間までの時間を入力してください

[Aurora Serverless V2 のインスタンス設定](#)



Aurora Serverless v2 のシームレスなスケーリング

Aurora Serverless v2のスケーリング例 (定期的に同時実行数を上げながらOLTP処理を実施)



同時実行数が増加して、必要なリソースが
増加した時点で、Aurora Serverless v2 の
ACU が増加(青線)
また、スケール時にトランザクション
(赤線の CommitThroughput)を阻害しない

処理が終了して、リソースが不要
になると徐々に ACU が減少(青線)

まとめ



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



Amazon Aurora の性能とスケーラビリティ



Amazon Aurora

- **Amazon Aurora はクラウド時代に Amazon が再設計した RDBMS**
 - MySQL・PostgreSQL と互換があり既存の資産を活かしやすい
- **性能向上を実現するための多くのチャレンジ/改善を継続して実行中**
 - 高いスループットや安定性
 - チェックポイント不要のアーキテクチャー
- **負荷に応じたスケーラビリティ**
 - Reader の Auto Scaling
 - 自動管理されたストレージ
 - Aurora Serverless v2 によるシームレスなスケーリング



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



Thank you!

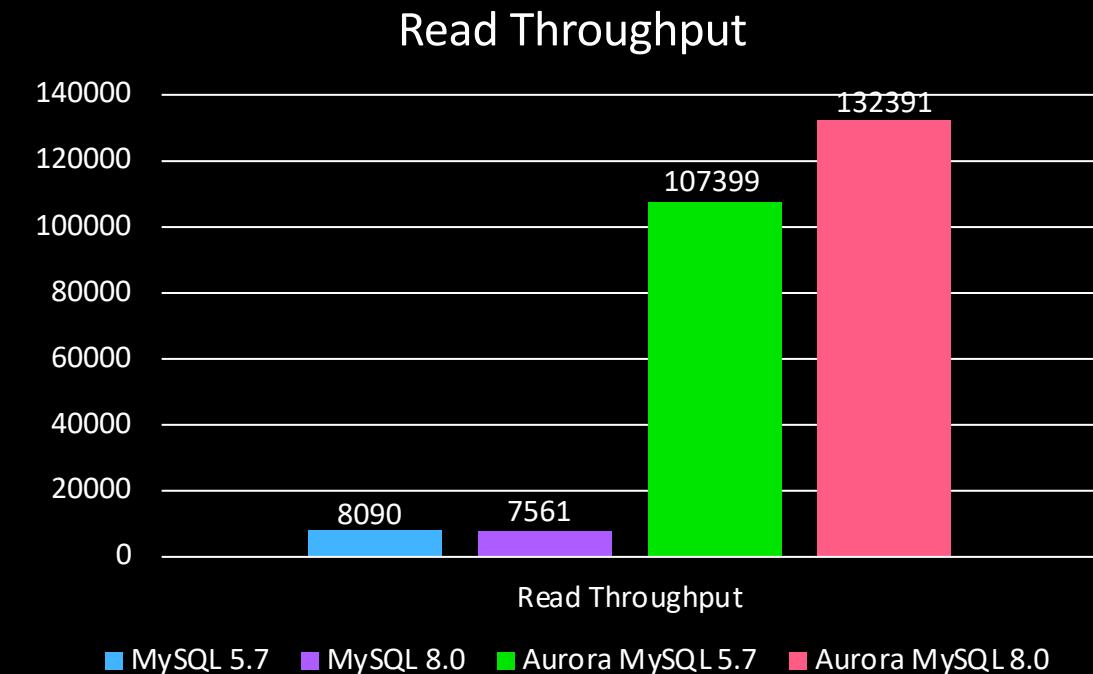
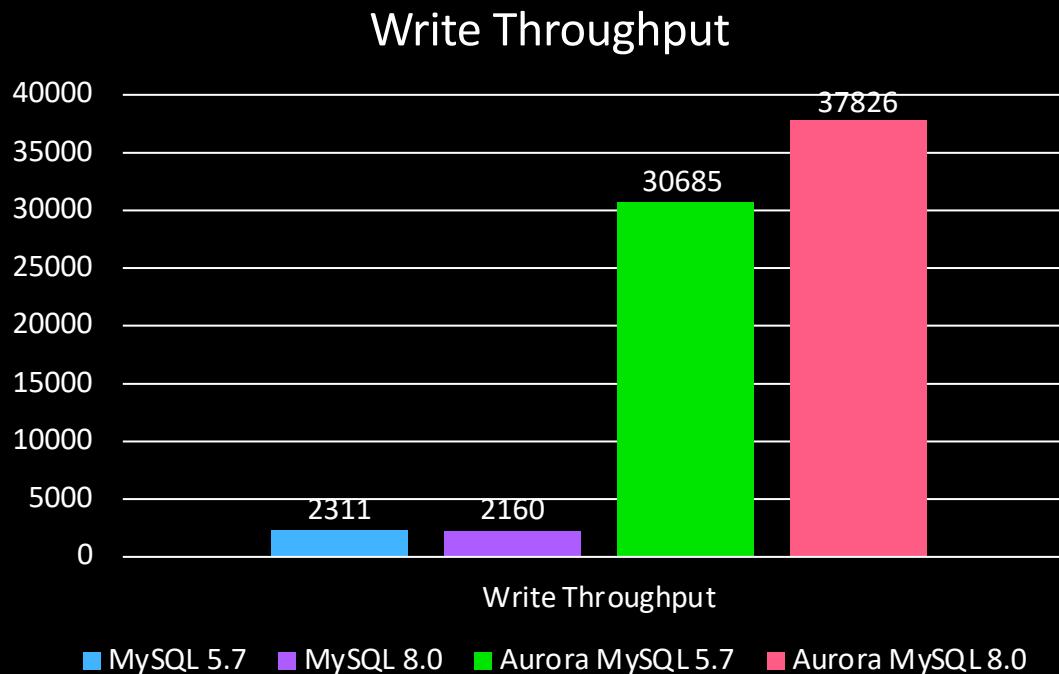
Daiki Suzuki
Solutions Architect

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



Appendix

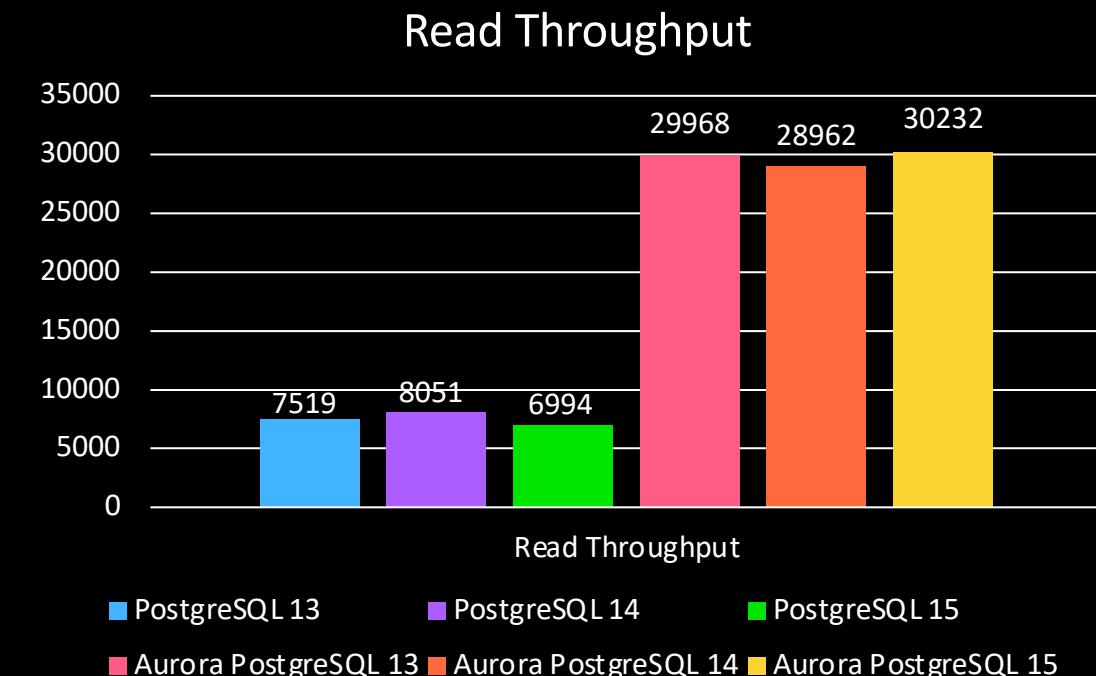
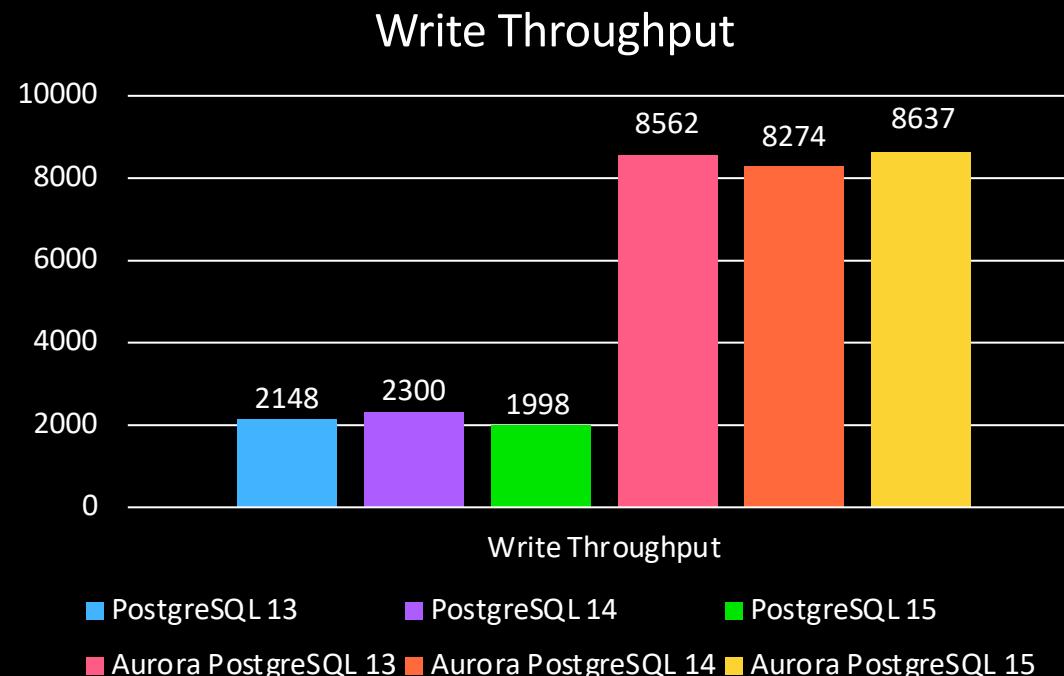
MySQL の書き込みと読み込みのスループット



Sysbench: 250テーブル、256スレッド、200,000行/テーブル (R4.16XL)

https://d1.awsstatic.com/product-marketing/Aurora/RDS_Aurora_Performance_Assessment_Benchmarking_v1-2.pdf

PostgreSQL の書き込みと読み込みのスループット



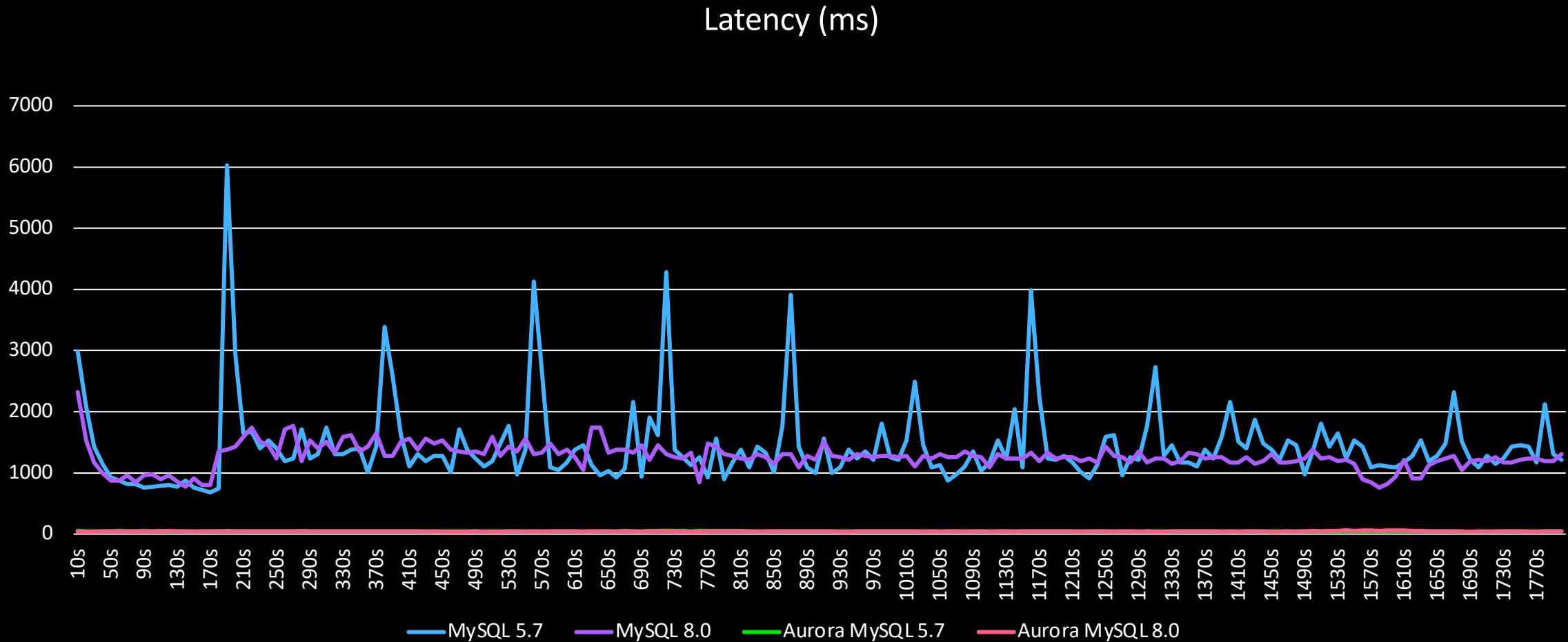
Sysbench: 250テーブル、256スレッド、200,000行/テーブル (R4.16XL)

https://d1.awsstatic.com/product-marketing/Aurora/RDS_Aurora_PostgreSQL_Performance_Assessment_Benchmarking_V1-0.pdf

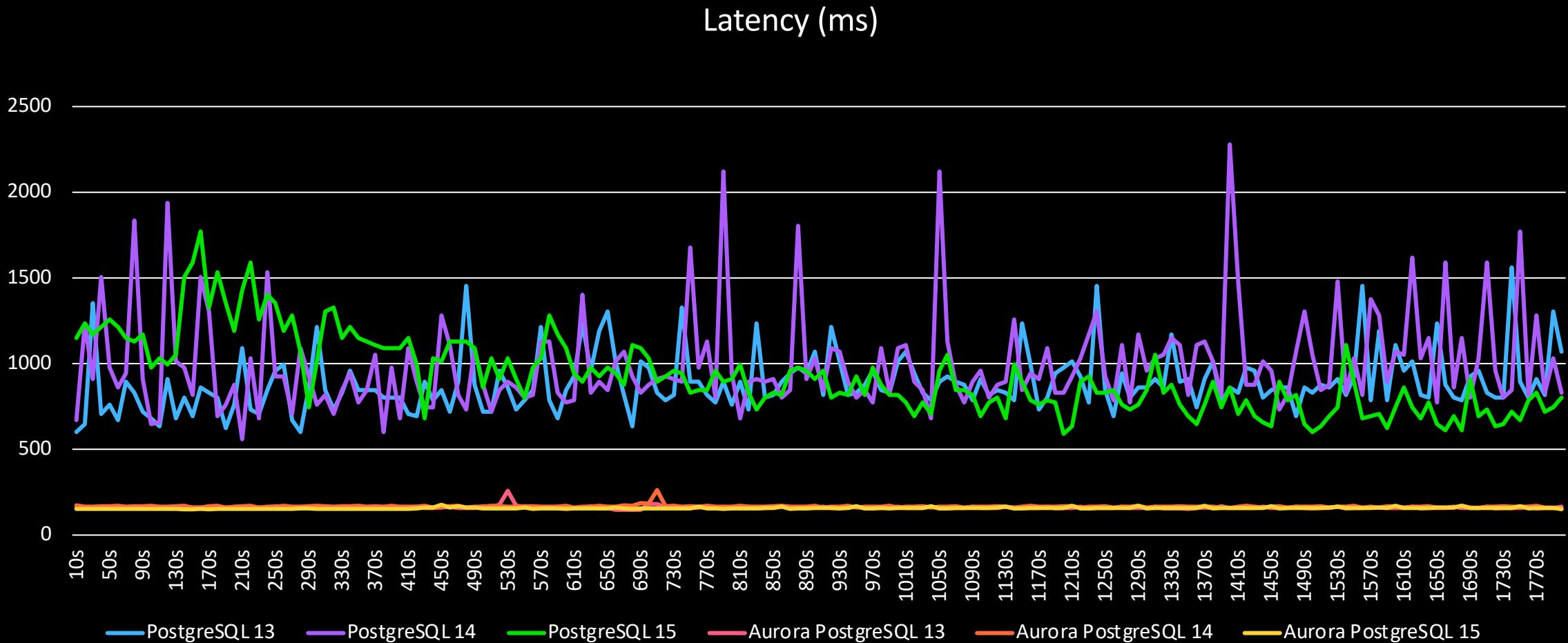


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

MySQL の安定性



PostgreSQL の安定性



AWS Black Belt Online Seminar

Amazon Aurora 概要紹介

藤田 洋平

Database Specialist Solutions Architect

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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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2025年5月時点のサービス内容および価格についてご説明しています。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)



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 の概要、他のデータベースとの違いを抑えたい方



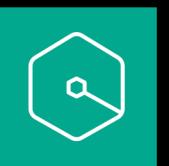
自己紹介

藤田 洋平

アマゾンウェブサービスジャパン
データベーススペシャリストソリューション
アーキテクト

公共部門のDatabase製品を担当しています。
お父さん業も奮闘中

好きな AWS サービス
Amazon Aurora, Amazon Q



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



アジェンダ

1. Purpose-built database
2. Amazon Aurora の概要
3. Amazon Aurora の詳細機能セッションのご案内



Purpose-built database



“データベースを選択する”



パフォーマンス



コスト



スケーラビリティ



データ



オペレーション

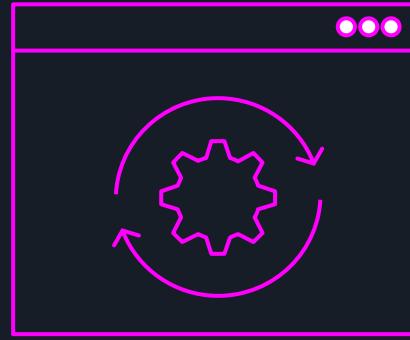


“理想”のデータベース

「万能なデータベース」の限界



パフォーマンスとコスト



一貫性と可用性



スケーラビリティと複雑性

トレードオフがあることを理解し
『過不足』なく要件整理することが重要



“One size fits nothing at all”



Purpose-built database



リレーショナル

参照整合性、
ACID Transaction、
schema-on-wirte



キーバリュー

高スループット、
低レイテンシの
read / write、
制限のない拡張性



ドキュメント

ドキュメントを
格納し、クエリ利
用により任意の
属性に対して
高速にアクセス



インメモリ

キーを利用して
マイクロ秒の
レイテンシでクエリ



グラフ

データ間の
関係性を高速、
容易に作成、探索



時系列

時間経過に従って
計測される
データを収集、
蓄積、処理



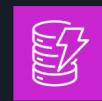
ワイドカラム

スケーラブルで、
可用性が高い、
マネージドな
Apache Cassandra
互換サービス

AWS サービス



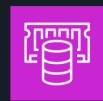
Amazon RDS
Amazon Aurora



Amazon
DynamoDB



Amazon
DocumentDB



Amazon
ElastiCache
Amazon
MemoryDB
for Redis



Amazon
Neptune



Amazon
Timestream



Amazon
Keyspaces

ユースケース

従来の
アプリケーション、
ERP、CRM、
会計

トラフィックの
多いウェブアプリ、
e コマース
システム、
リアルタイム入札

コンテンツ管理、
カタログ、
ユーザー
プロファイル

キャッシュ、
セッション管理、
ゲームの
リーダーボード、
地理空間
アプリケーション

不正検出、
ソーシャル
ネットワーク、
レコメンデーション
エンジン

IoT
アプリケーション、
イベント
トラッキング

低レイテンシな
アプリケーションの
構築、
Cassandra ベースの
アプリケーションを
クラウドに移行



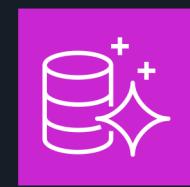
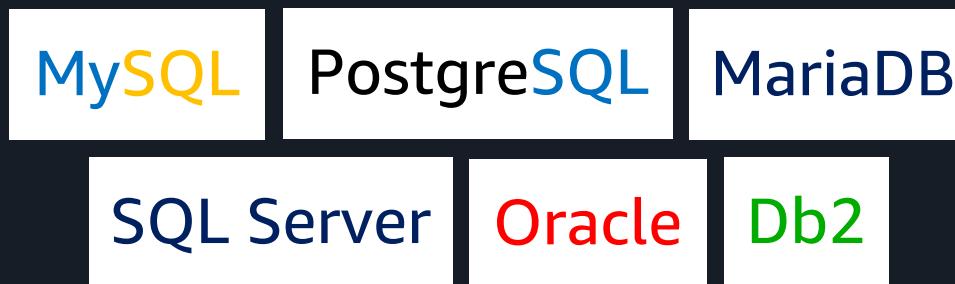
Amazon RDS と Amazon Aurora

リレーショナルモデル



Amazon RDS

6つの一般的なデータベースエンジンから選べる
マネージド型リレーショナルデータベースサービス
データベースエンジン オリジナルに近い機能で提供



Amazon Aurora

2つのオープンソースデータベースをクラウド向けに
再設計したマネージド型リレーショナルデータベースサービス
及びサーバーレスでスケーラブルな分散SQLデータベース



Amazon Aurora の概要



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

優れた性能と拡張性

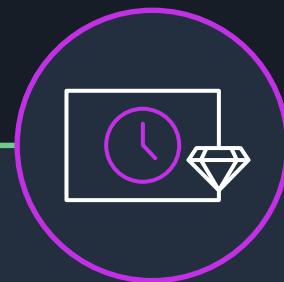
高可用性と耐久性

高い安全性

フルマネージド



標準的なMySQLやPostgreSQLと比べて高いパフォーマンスを実現; 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



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



ネットワーク分離、保管時/通信の暗号化



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

Amazon Auroraのイノベーション

- 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ィドアーキテクチャー
- 3 自動化されたタスク - 完全マネージド・サービス



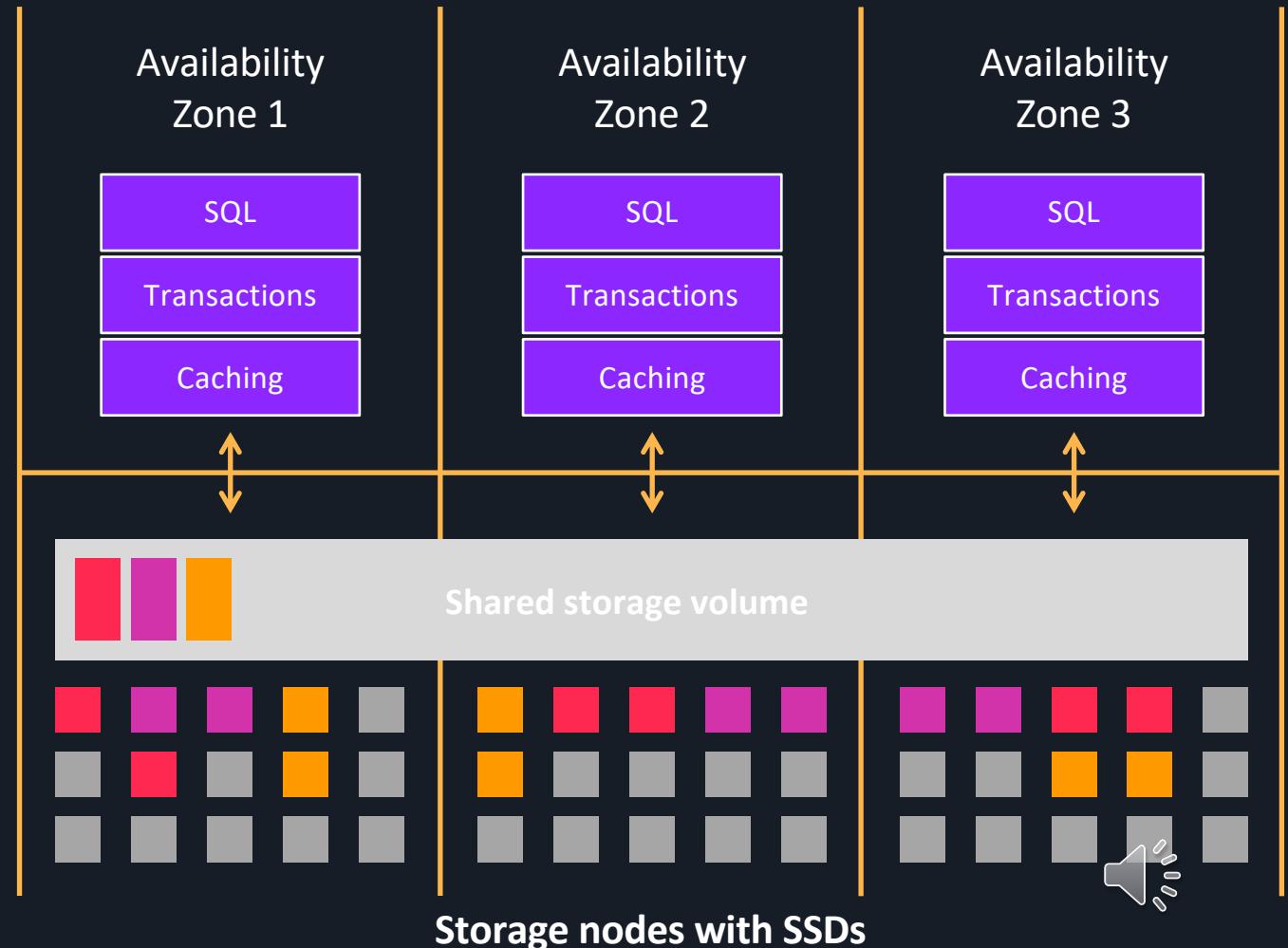
スケールアウト、分散、マルチテナントアーキテクチャー

データベース用に設計された専用のlog structured分散ストレージシステム(チェックポイントによるデータページの書き込みは不要)

3つの異なるアベイラビリティゾーンに分散された数百のストレージノードにストライピングされたストレージボリューム

AZ+1の障害から保護するためにデータを各アベイラビリティゾーンに2つのコピー、リージョン内で6つのコピー

データは10GBのプロテクショングループ(データの6つのコピー)で書き込まれ、自動的に最大128TBまで拡張



AWSサービスを活用したサービスオリエンティッド アーキテクチャー



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



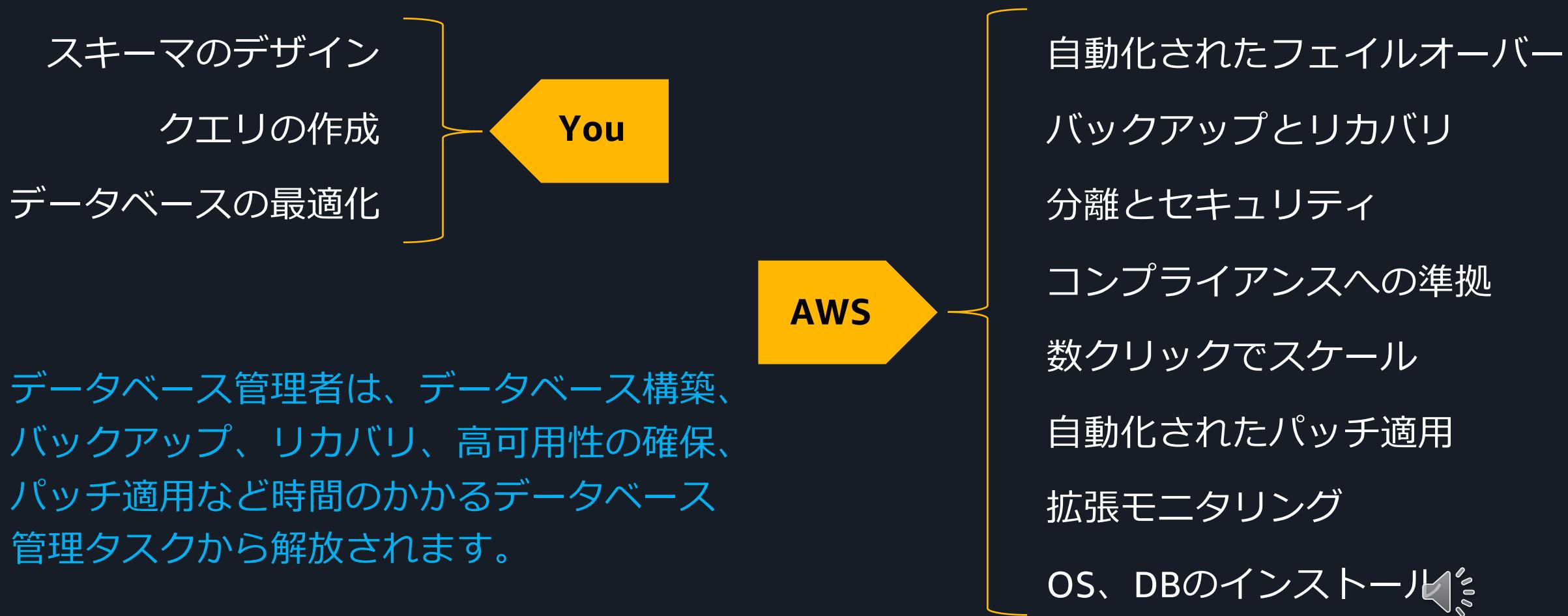
Amazon Simple Storage Service (Amazon S3)データをロード、S3を利用したバックアップデータの保存、リストア



システムメトリックスやログをCloudWatchへアップロード



自動化されたタスク – 完全マネージド・サービス



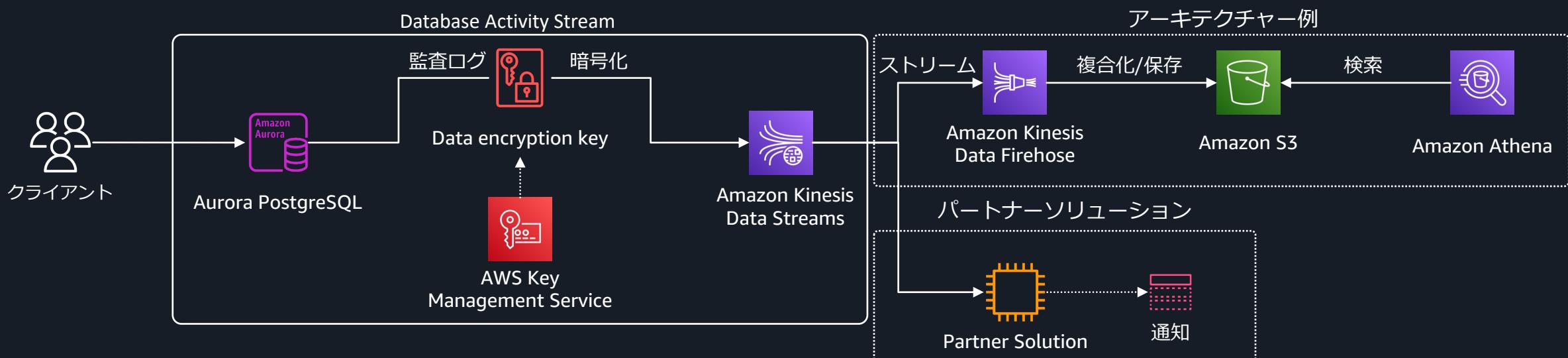
セキュリティとコンプライアンス

- データの暗号化
 - AES-256 (ハードウェア支援)
 - ディスクとAmazon S3に置かれている全ブロックを暗号化
 - AWS KMSを利用したキー管理
- SSLを利用したデータ通信の保護
- 標準でAmazon VPCを使ったネットワークの分離
- ノードへ直接アクセスは不可能
- AWS IAMによるリソースレベルでの権限制御
- 業界標準のセキュリティとデータ保護の認証をサポート



Database Activity Stream

- データベースのアクティビティをAmazon Kinesis Data Streams^(*)2)に格納 (サンプルの構成などの[ブログ](#))
- 格納したデータベースアクティビティをDBAから保護することが可能
- コンプライアンスや規制要件に利用可能
- 精度を優先する同期モードとパフォーマンスを優先する非同期モードの2つのモードがある
- 3rd パーティーの監査製品との連携 (IBM Security Guardium, Imperva SecureSphere Database Audit and Protection)



(*)1) <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/DBActivityStreams.html>

(*)2) <https://aws.amazon.com/jp/kinesis/data-streams/>

統合された高度な監視ツール Cloud Watch Database Insights

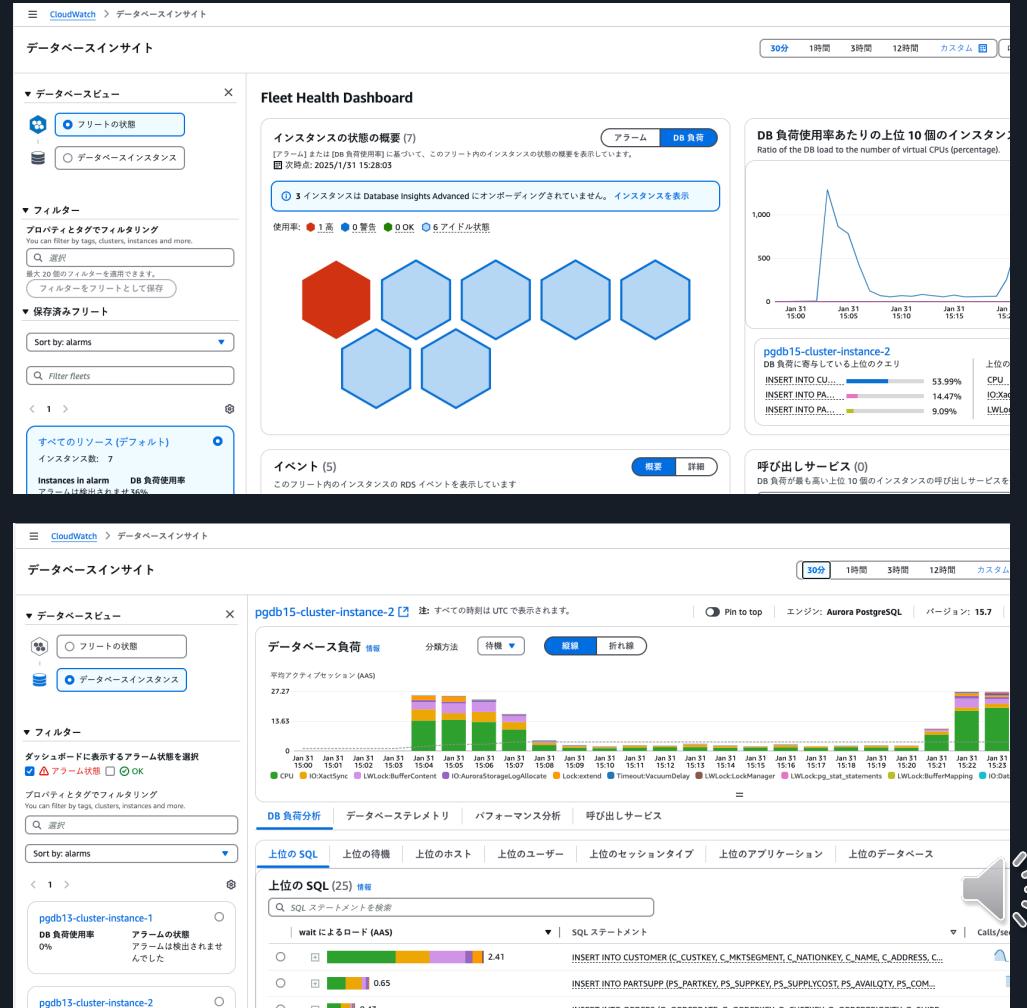
Before Dec' 2024

現在



統合された高度な監視ツール Cloud Watch Database Insights

- CloudWatchの統合ビューにすべてのデータベーステレメトリを統合
- 数百のデータベースを構成可能なfleet-wideビュー
- 詳細なSQL単位のメトリクス
- 呼び出し元のアプリケーションサービスとそのデータベース間の依存関係をマッピングした情報を含むアプリケーションコンテキスト
- 事前に作成されたヘルスダッシュボード、事前定義されたしきい値、推奨事項があるので、簡単に始められます
- フルマネージドエクスペリエンス - エージェントレスでメンテナンスも不要

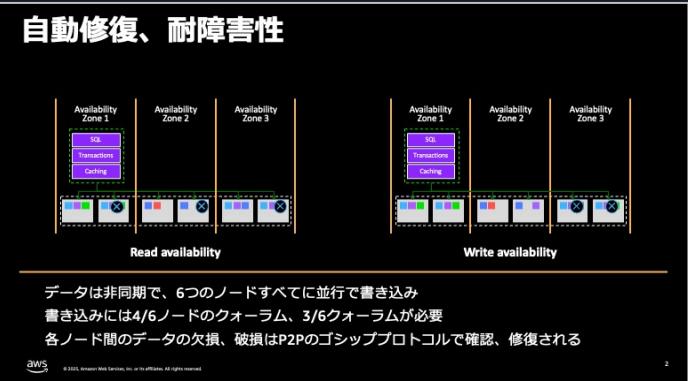


Amazon Aurora の詳細機能 セッションのご案内

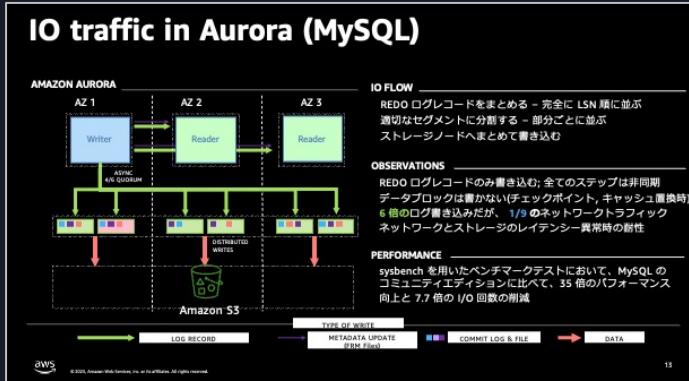


Amazon Aurora - Black Belt

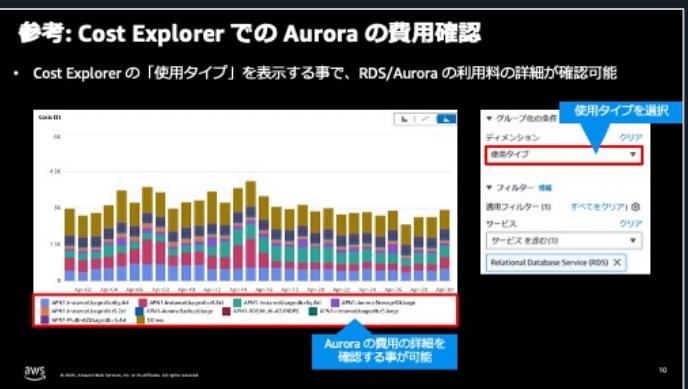
[可用性]



[高性能・スケーラビリティ]



[優れたコスト効率]



[移行支援プログラム・サービス]



詳細編

[性能チューニング]

[メンテナンス・運用]

[Global Database]

[Limitless Database]

etc...



Thank you!



AWS Black Belt Online Seminar

Amazon Aurora チューニングアプローチ

河合 智彦

Technical Account Manager
2025/09



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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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2025年9月時点のサービス内容および価格についてご説明しています。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

河合 智彦

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ー

DNB/ISVのお客様を中心にご支援しています。

Amazon Aurora をより身近に感じ、気軽にご利用
ただきたいです。

好きな AWS サービス
Amazon Aurora, AWS Cost Explorer



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 をご利用中でチューニングを検討中の方

アジェンダ

1. チューニングの流れ、確認方法

- ・性能データ収集、管理に必要とされるもの
- ・Amazon RDS/Aurora で利用可能なモニタリングツール

2. CloudWatch Database Insights の紹介

- ・Performance Insights からの移行
- ・Database Insights スタンダード / アドバンスト モード

3. Cloud Watch メトリクス、Cloud Watch Logs、拡張モニタリングの紹介

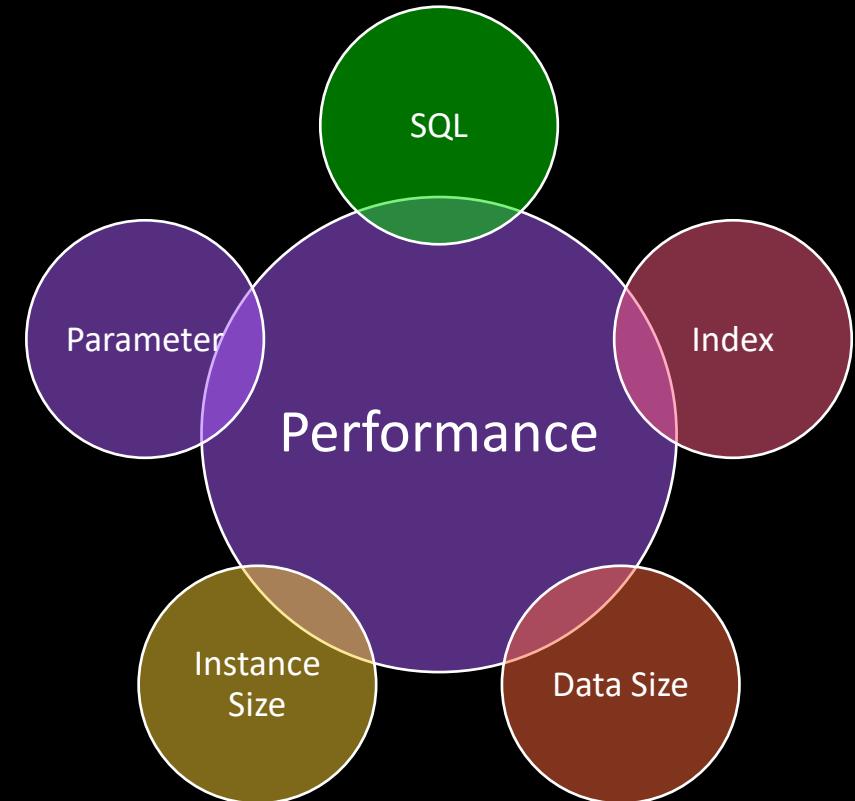
4. 待機イベントの見方

- ・データベースロード
- ・待機イベントについて
- ・データベースロードを待機イベントで分析

チューニングの流れ、確認方法

サービス最適化

- ・パフォーマンスチューニング工数の考慮
 - ・目標スケジュール、KPI、優先順位の設定
- ・個別対応・全体的な対応?
 - ・インスタンス全体 (パラメータ: メモリ、I/O等)
 - ・特定処理の最適化 (SQL最適化: SQL、インデックス等)
 - ・短期的・長期的な対応が必要か?



管理工数を抑え、継続的にデータベースを監視・管理して変化を検知
≒ 安定したサービスを提供し、常に最適なサービスを利用者に提供

一般的な性能管理のサイクルの課題

コレクション (データ収集)

アナリシス (分析)

チューニング (最適化)

- 細かい粒度のパフォーマンスデータを長期間にわたり、収集、管理していくのは運用コストが高い
 - OSのメトリクスだけではなくDB特有のメトリクスを対象データベースに負荷なく細かい粒度で取得することが困難
 - データ量が多くなる中で安定したパフォーマンス分析インフラを維持管理するコストが高い

- パフォーマンス問題のような複雑で広範な事象を特定の統計情報から分析する的是困難
- どの程度リソースが不足、余剰なのかを見積もることが困難もしくは不可能

- ボトルネックの分析が曖昧、不十分なため、SQLやDBの効果的なチューニングが困難もしくは非効率
 - どの程度チューニングされたら成功とみなすか判断できない
 - リアルタイムでDB全体のパフォーマンス測定できないため、チューニング効果の確認に時間がかかる

性能データ収集、管理に必要とされるもの

コレクション (データ収集)

アナリシス (分析)

チューニング (最適化)

- 突発的なパフォーマンス障害に対応するためには細かい粒度のデータが必要
- 過去に発生した問題を分析するためにはある程度長期間パフォーマンスデータの保存、削除するといった管理が必要
- パフォーマンス指標として一般的なOSのリソース情報やDB内部の動作を推測するためのDB統計情報に加えてDBのワークロードを端的に示す指標が必要

- シンプルなパフォーマンス指標を元に、該当時間帯でパフォーマンス問題が発生しているか否かを端的に理解できる必要がある
- パフォーマンス問題が発生している場合、問題の原因をドリルダウンして究明できる必要がある
- チューニングした後の姿を予測し、チューニングのゴールを明確にしておく必要がある

- チューニング対象におけるボトルネックを排除し、事前に設定したチューニングのゴールに向けて最適化を実施
 - 推測ではないエビデンスに基づいたチューニング
 - データ収集 → 分析の流れがシステム化されていれば、チューニング後即座に効果の測定が可能

性能データ収集、管理に必要とされるもの

コレクション

(データ収集)

アナリシス

(分析)

チューニング

(最適化)

パフォーマンスデータを平時から収集、管理し、有事には即座にそのデータを活用、分析できるインフラが必要

さらに、突発的なパフォーマンス障害を想定してリアルタイムに近いデータ収集、分析を可能にすることが望ましい

- チューニング対象におけるボトルネックを排除し、事前に設定したチューニングのゴールに向けて最適化を実施
 - 推測ではないエビデンスに基づいたチューニング
 - データ収集 → 分析の流れがシステム化されていれば、チューニング後即座に効果の測定が可能

Amazon RDS/Aurora で利用可能なモニタリングツール

ツール名	できること	設定方法
<u>CloudWatch</u>	Hypervisor等の各種メトリクスをデフォルト60秒間隔で取得・確認可能にする	<ul style="list-style-type: none">デフォルト有効 (無効化不可)
<u>Amazon RDS 拡張モニタリング</u>	OSレイヤーのCPU利用率やメモリなどの各種メトリクスを1、5、10、15、30、60秒間隔で取得・確認可能にする	<ul style="list-style-type: none">インスタンス作成時に有効/無効を指定 (デフォルト有効)インスタンス作成後に変更可能 (インスタンス再起動不要)
<u>CloudWatch Database Insights</u>	データベース内の性能情報を蓄積・視覚化し、15カ月過去に遡ってワークロード分析を可能にする※1	<ul style="list-style-type: none">インスタンス作成時にスタンダードかアドバンストを選択インスタンス作成後に変更可能

参照:

AWS ブログ : [Amazon RDS と Amazon Aurora のパフォーマンスとイベントの可視性を高める](#)

AWS ドキュメント : [Amazon Aurora のメトリクスのモニタリングの概要](#)

re:Post : [Amazon RDS for MySQL DB インスタンスのパフォーマンスをモニタリングする方法を教えてください。](#)

※1 性能情報の保持期間はアドバンストモードの場合であり、スタンダードモードの場合はワークロード分析においても利用できない機能があります

トップダウンからの分析



リソース監視

- Throughput
- Latency
- Utilization – CPU,IOPS,Memory
- Database availability
- Authentication failure
- Slow running queries

データベースの状況確認

- DB Load Analysis – SQL,waits,hosts,users,databases
- Database Telemetry – Metrics,Logs,Slow SQL Queries, Events, OS Processes
- Performance Analysis
- Calling Services

短いデータ収集間隔での詳細確認

- Cloud Watch Dashboard with Enhanced Monitoring
- OS process list

CloudWatch
によるリソース監視

Database Insights
による分析

拡張モニタリング
による詳細分析

CloudWatch Database Insights の紹介

Performance Insights からの移行

2025年9月時点ではデータベースのパフォーマンスチューニングとモニタリングを行う Amazon RDS Performance Insights が利用可能です。

2026 年 6 月 30 日に、RDS コンソールの Performance Insights ダッシュボード [1] とフレキシブルな保持期間とその価格設定 [2] が廃止となります。引き続き Performance Insights のメトリクスを表示するには、CloudWatch Database Insights を利用します。

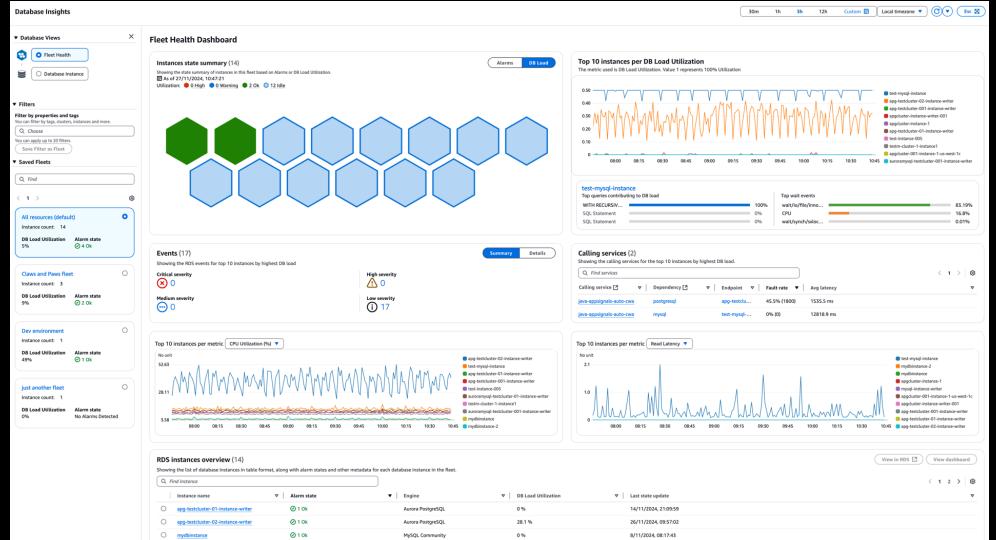
- Performance Insights (PI)ダッシュボードの廃止 ※PIのAPIは継続利用可能
- スタンダードモードとアドバンストモードがあり新規作成時に指定
- 料金設定に伴う柔軟な保持期間の廃止
 - Database Insights スタンダードモードは7日で固定 ※無料利用枠あり
 - Database Insights アドバンストモードは15か月で固定
- RDS for Oracle および RDS for SQL Server の実行計画キャプチャ機能は、Database Insights のアドバンストモードに移行
- RDS PostgreSQL、Aurora PostgreSQL、Aurora MySQL のオンデマンド分析機能は、Database Insights のアドバンストモードに移行

[1] https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/USER_PerfInsights.UsingDashboard.html

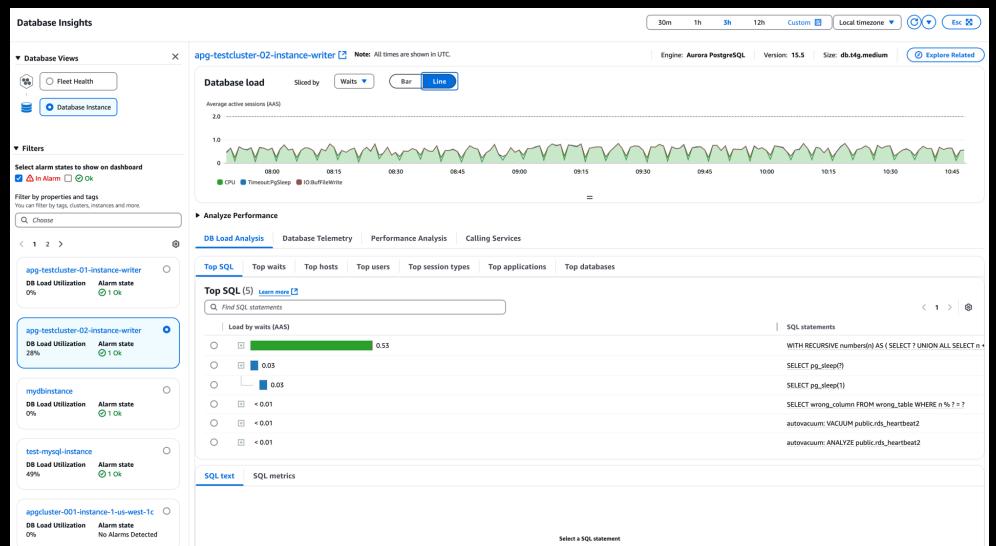
[2] <https://aws.amazon.com/rds/performance-insights/pricing/>

Database Insights について

- CloudWatchにおける統一されたビューでのすべてのデータベースステレメトリの統合
- 数百のデータベースにわたる設定可能なフリート全体のビュー
- 詳細なSQLクエリメトリクス
- 呼び出し元アプリケーションサービスと依存するデータベース間の依存関係マッピングによるアプリケーションコンテキスト
- 事前構築された健全性ダッシュボード、事前定義されたしきい値、推奨事項により簡単に着手可能
- フルマネージドのエクスペリエンス - エージェントやメンテナンスの手間なし



Fleet view



Instance view



Database Insights スタンダード / アドバンスト モード

機能	スタンダードモード (7 Days)	アドバンストモード (15 month)
DB Load の上位の寄与要因をディメンション別で分析する	サポート	サポート
最大7日間の保持期間を持つデータベースメトリクスにクエリ、グラフ、アラーム設定をする	サポート	サポート
きめ細かなアクセスコントロールポリシーを定義し、SQL テキストなどの機密性の高いディメンションへのアクセス権を制限する	サポート	サポート
実行中のプロセスごとに詳細なメトリクスを使用して、データベースで発生しているオペレーティングシステムのプロセスを分析する この機能を使用するには、[Amazon RDS 拡張モニタリング] が有効になっている必要があります。	サポート外	サポート
フリート全体のモニタリングビューを作成して保存し、何百ものデータベースの健全性を評価する	サポート外	サポート
15ヶ月の保持期間とガイド付き UX の環境で SQL ロックを分析する	サポート外	Aurora PostgreSQL でのみサポート
15ヶ月の保持期間とガイド付き UX の環境で SQL 実行プランを分析する	サポート外	Aurora PostgreSQL、RDS for Oracle、および RDS for SQL Server でのみ対応
遅い SQL クエリを分析する	サポート外	サポート
メトリクス、ログ、イベント、アプリケーションを含め、すべてのデータベーステレメトリの統合ダッシュボードを表示する	サポート外	サポート
CloudWatch で Amazon RDS イベントを表示する	サポート外	サポート
オンデマンド分析を使用し、選択した期間のデータベースのパフォーマンスを分析する	サポート外	Aurora PostgreSQL、Aurora MySQL、RDS for PostgreSQL、RDS for MySQL、および RDS for MariaDB でのみ対応

参考：[Database Insights のモード](#)

データベースビュー

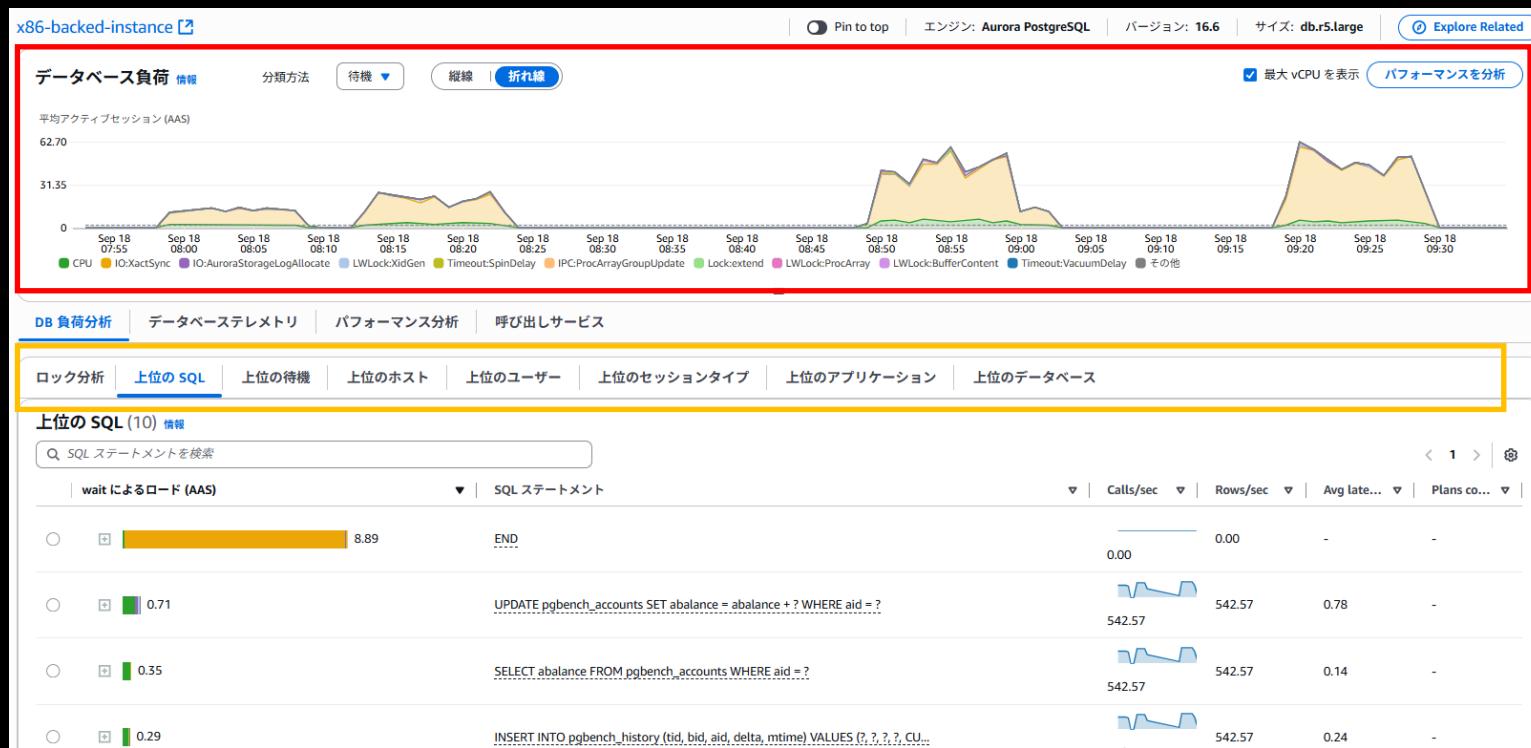
フリートの状態 (Fleet view)



- ヒートマップ (高負荷 + アラームの発生しているインスタンス)
- 高負荷な上位10インスタンス
- DB負荷に影響を与える項目の確認
 - 上位のSQLクエリ
 - 上位の待機イベント
 - 上位のアプリケーション/サービス
- 失敗した直近のデータベースイベント
- 上位10インスタンスの主要なメトリクスのウィジェット表示
- インスタンスリスト

データベースビュー

データベースインスタンス (Instance view)



「データベース負荷」

- データベース負荷指標の確認

「DB負荷分析」

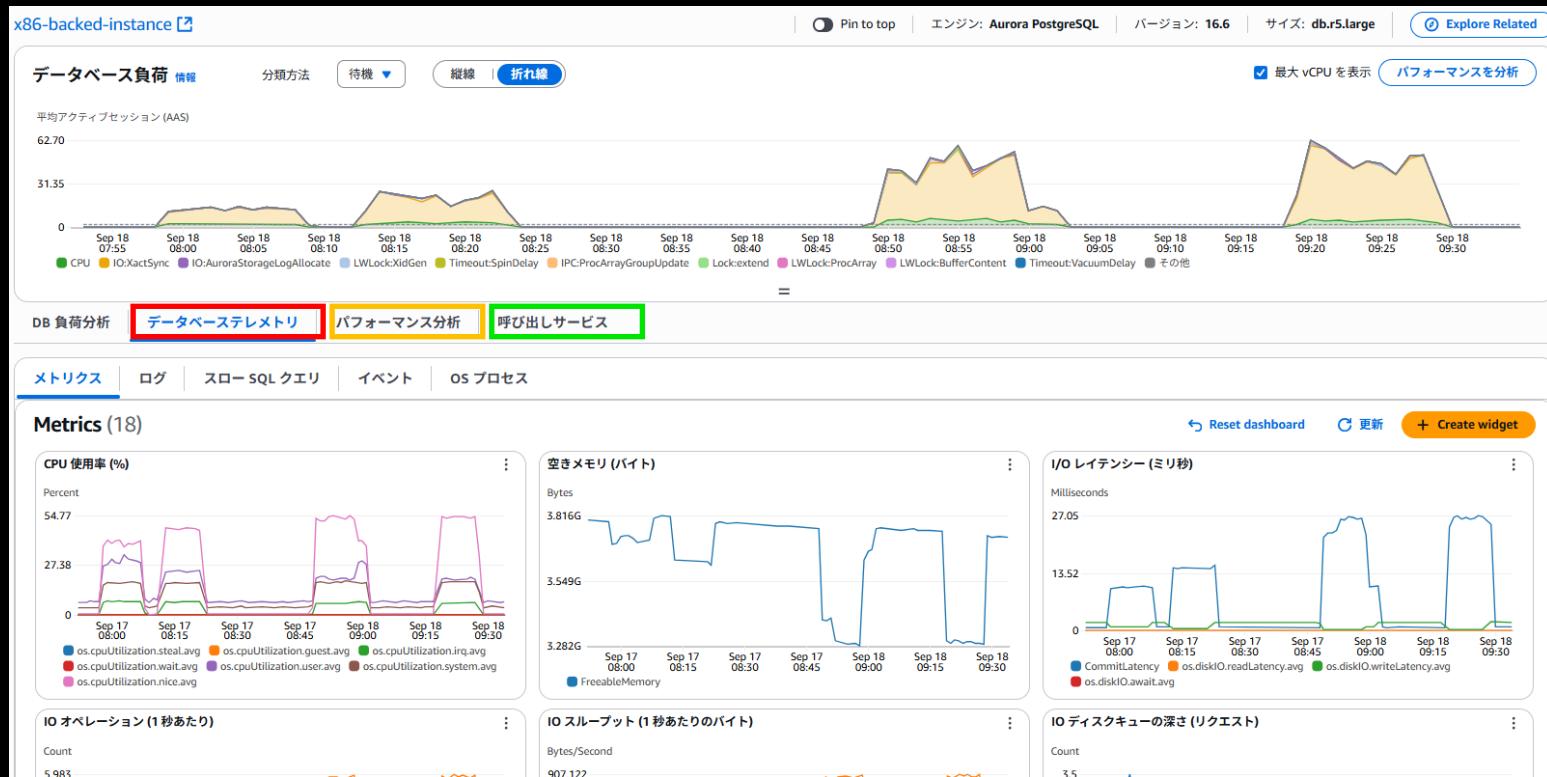
- データベース負荷に影響を与えるTop Nの確認

- 上位の SQL
- 上位の待機イベント
- 上位のホスト
- 上位のユーザ
- 上位のデータベース
- ロック分析※1
- 上位のセッションタイプ※2
- 上位のアプリケーション※2

※1 ロック分析がサポートされているDBエンジンのみ
※2 Aurora PostgreSQL のみサポート

データベースビュー

データベースインスタンス (Instance view)



「データベースステレメトリ」

- メトリクス、ログ、スローSQLクエリ、イベント、OSプロセスを単一のビューに統合

「パフォーマンス分析」

- 高度な根本原因分析

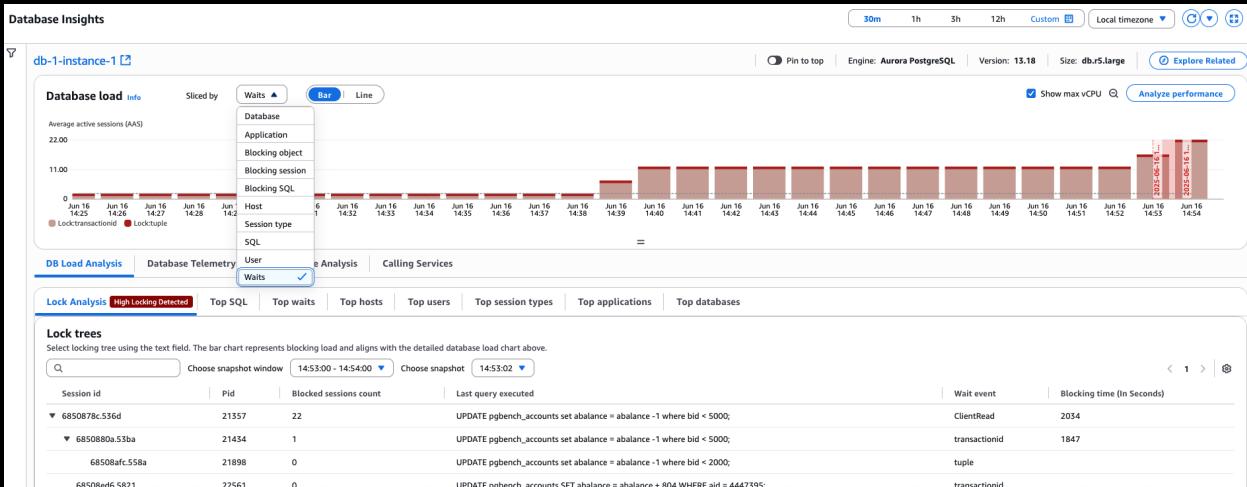
「呼び出しサービス」

- 呼び出し元サービスとオペレーションの確認

Database Insights の機能

アドバンストモードで利用可能

ロック分析



Aurora PostgreSQL のみ対応
2025/9 時点

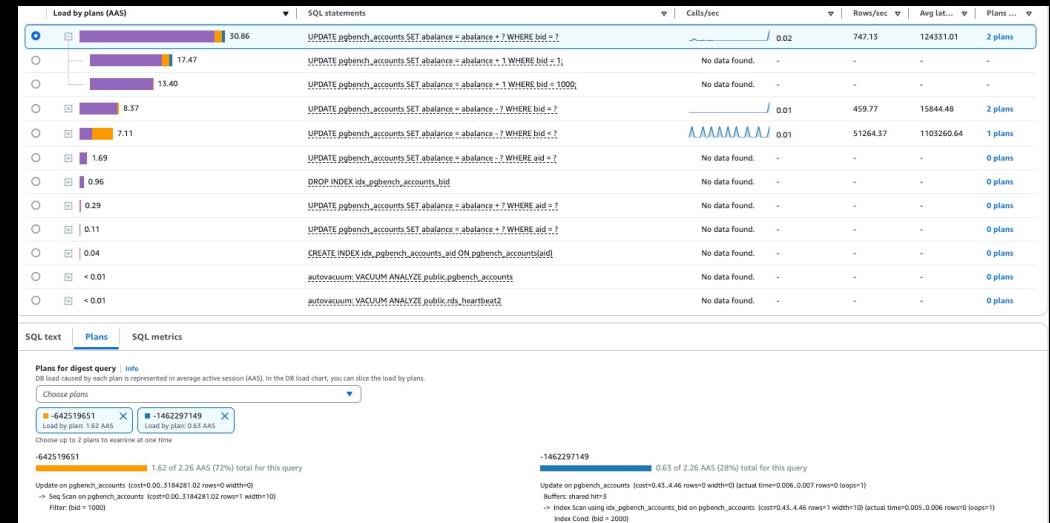
参照:

[Amazon Aurora PostgreSQL のロックツリーの分析](#)



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

実行プランの分析

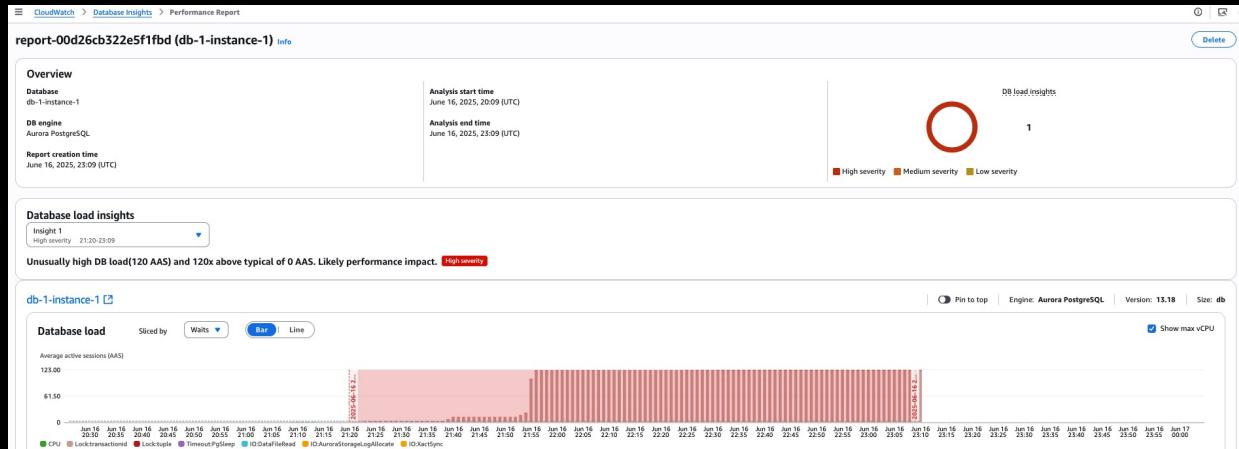


Aurora : PostgreSQL,
RDS : Microsoft SQL Server Oracleのみ対応
2025/9 時点
参照:
[実行プランの分析](#)

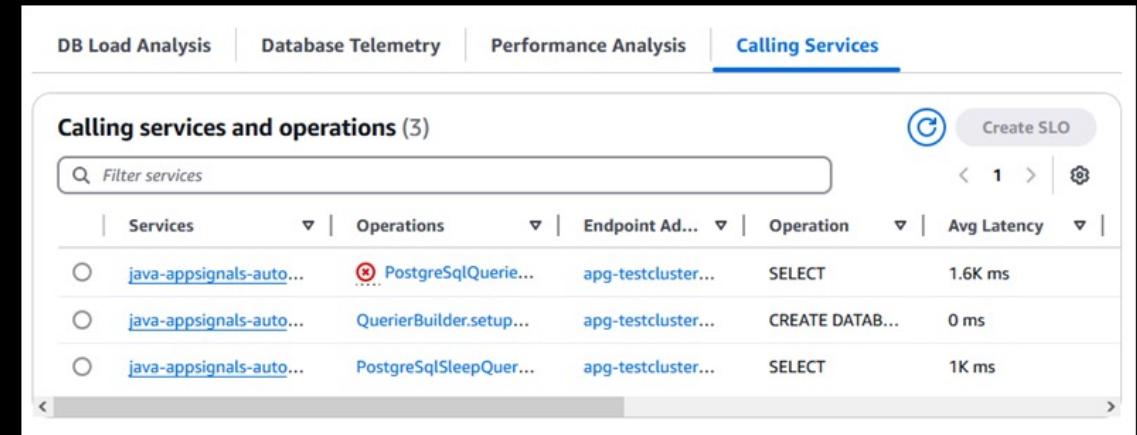
Database Insights の機能

アドバンストモードで利用可能

オンデマンド パフォーマンス分析



Application performance monitoring (APM)



参考:
[データベースのパフォーマンスをオンデマンド分析する](#)

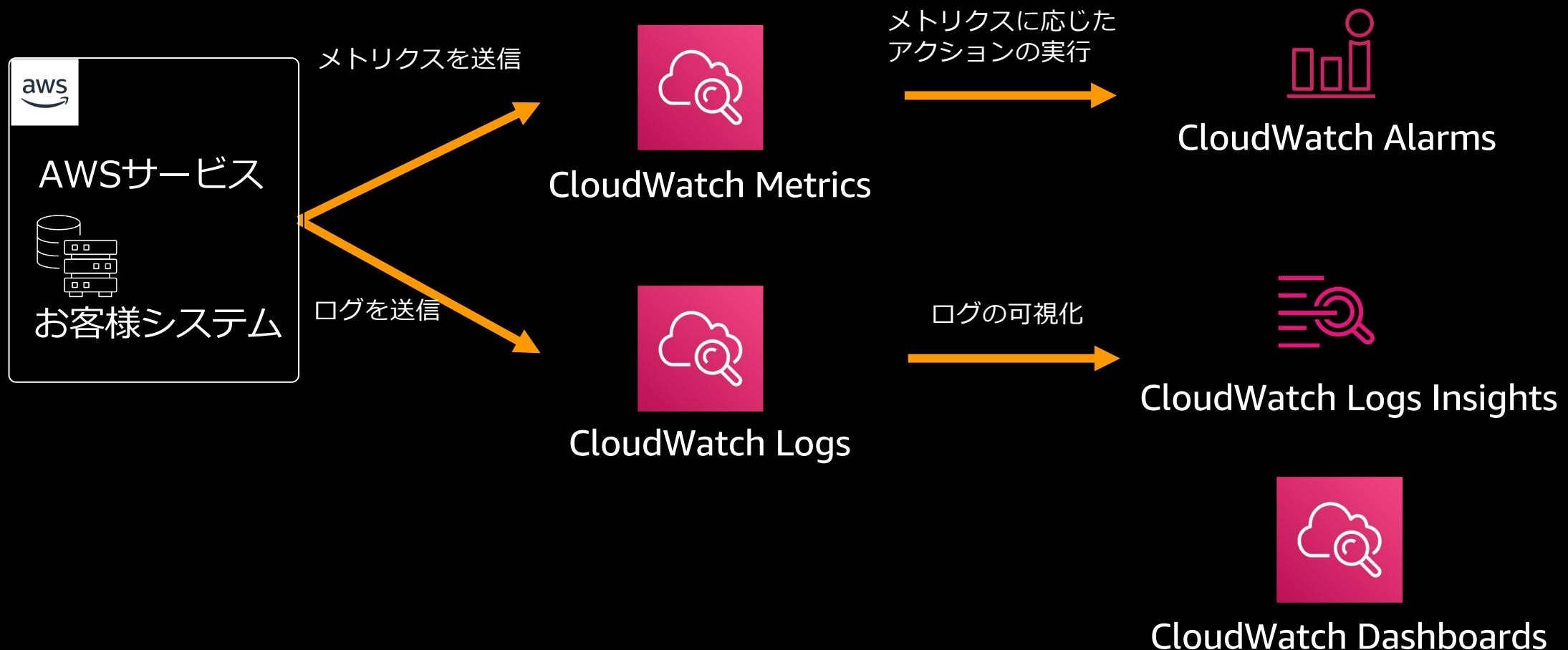


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

参考:
[アプリケーションパフォーマンスマニタリング \(APM\)](#)

Cloud Watch メトリクス Cloud Watch Logs 拡張モニタリングの紹介

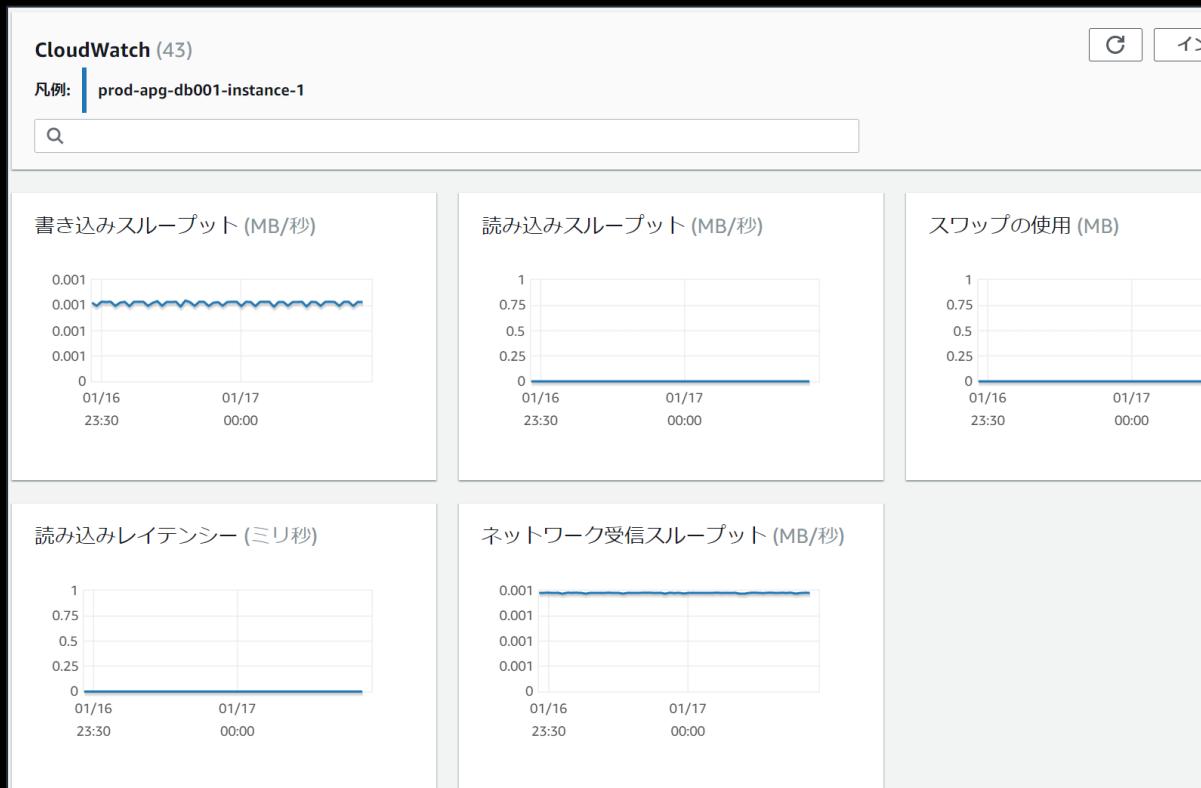
モニタリング基盤としてのAmazon CloudWatch



メトリクスとアラーム

CloudWatch Metrics

標準的なメトリクスを60秒間隔で収集



CloudWatch Alarm メトリクスの状態による通知

メトリクス

グラフ
このアラームは青線が5分内の1データポイントで上回る赤線を超える場合に、トリガーされます。

名前空間 AWS/RDS
メトリクス名 DBLoad
DBInstanceIdentifier ams-v3-02-1-instance-1
統計 平均値
期間 5分

条件

しきい値の種類

静的
値をしきい値として使用

異常検出
バンドをしきい値として使用

DBLoad が次の時...
アラーム条件を定義します。

より大きい
> しきい値

以上
>= しきい値

以下
<= しきい値

より低い
< しきい値

... よりも
しきい値を定義します。
10000
数字である必要があります



CloudWatch Logs

Amazon RDS、Amazon Auroraでは「ログのエクスポート」を設定することで、CloudWatch Logs へ連携が可能

- 監査ログ/エラーログ/一般ログ/スロークエリログなど

The screenshot shows the AWS CloudWatch Logs interface for an Aurora MySQL cluster named 'auroralab-mysql-node-1'. The log group path is /aws/rds/cluster/auroralab-mysql-cluster/slowquery. The page displays a list of log events, each containing timestamp, source, and query details. A search bar at the top allows filtering by event text. Buttons for actions like 'アクション' (Actions) and 'テーリングを開始' (Start Tailoring) are visible.

Time	User@Host	Query
2025-09-11T07:51:42.581+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:42.581370Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2526 # Query_time: 43.580308 Lock_time: 38.766561 Rows_sent: 0 Rows_examined: 31...
2025-09-11T07:51:42.581370Z	administrator[administrator] @ [172.30.0.63]	# User@Host: administrator[administrator] @ [172.30.0.63] Id: 2526 # Query_time: 43.580308 Lock_time: 38.766561 Rows_sent: 0 Rows_examined: 3196841
2025-09-11T07:51:42.581370Z	administrator[administrator] @ [172.30.0.63]	SET timestamp=1757544659;
2025-09-11T07:51:42.581370Z	administrator[administrator] @ [172.30.0.63]	UPDATE mylab.weather SET max_temp = 42 where id='USC00046699';
2025-09-11T07:51:46.938+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:46.938002Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 43.018324 Lock_time: 0.000000 Rows_sent: 0 Rows_examined: 31...
2025-09-11T07:51:50.041+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:50.041904Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 3.101420 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 319...
2025-09-11T07:51:51.727+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:51.727946Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 43.108047 Lock_time: 0.000000 Rows_sent: 0 Rows_examined: 31...
2025-09-11T07:51:54.109+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:54.109427Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 4.066375 Lock_time: 0.000002 Rows_sent: 0 Rows_examined: 319...
2025-09-11T07:51:55.705+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:55.705039Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 3.957913 Lock_time: 0.000002 Rows_sent: 1 Rows_examined: 319...
2025-09-11T07:51:57.938+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:57.938294Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2518 # Query_time: 47.792960 Lock_time: 41.583322 Rows_sent: 0 Rows_examined: 3...
2025-09-11T07:51:59.943+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:51:59.943199Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 4.237675 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 319...
2025-09-11T07:52:03.375+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:52:03.375394Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2518 # Query_time: 5.419015 Lock_time: 0.000002 Rows_sent: 0 Rows_examined: 319...
2025-09-11T07:52:04.256+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:52:04.256774Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 4.310388 Lock_time: 0.000003 Rows_sent: 0 Rows_examined: 319...
2025-09-11T07:52:05.175+09:00	administrator[administrator] @ [172.30.0.63]	# Time: 2025-09-10T22:52:05.175040Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2523 # Query_time: 50.618817 Lock_time: 43.383179 Rows_sent: 0 Rows_examined: 3...

ログのエクスポート
Amazon CloudWatch Logs に発行するログタイプを選択します

- 監査ログ
- エラーログ
- 全般ログ
- iam-db-auth-error ログ
- instance ログ
- スロークエリログ

Aurora MySQL の場合の選択肢

- 任意のクエリを選択して展開すると、そのクエリに関する詳細情報を確認可能

CloudWatch Logs Insights

CloudWatch Logs の可読性を向上させるために CloudWatch Logs Insights を使用

The screenshot shows the AWS CloudWatch Logs Insights interface. At the top, there's a navigation bar with 'CloudWatch' and 'ログのインサイト'. Below it, a sub-header says 'OpenSearch による分析 - 新規'. The main area is titled 'ログのインサイト' with a '情報' tab selected. It shows a query editor with the following SQL-like code:

```
1 filter @LogStream = 'auroralab-mysql-node-1'
2 | parse @message "# Time: * User@Host: * Id: * Query_time: * Lock_time: * Rows_sent: * Rows_examined: * timestamp=;" as Time, User, Id, Query_time, Lock_time, Rows_sent, Rows_examined, Timestamp, Query
3 | display Time, Query_time, Query, Rows_sent, Rows_examined
4 | sort Query_time asc
```

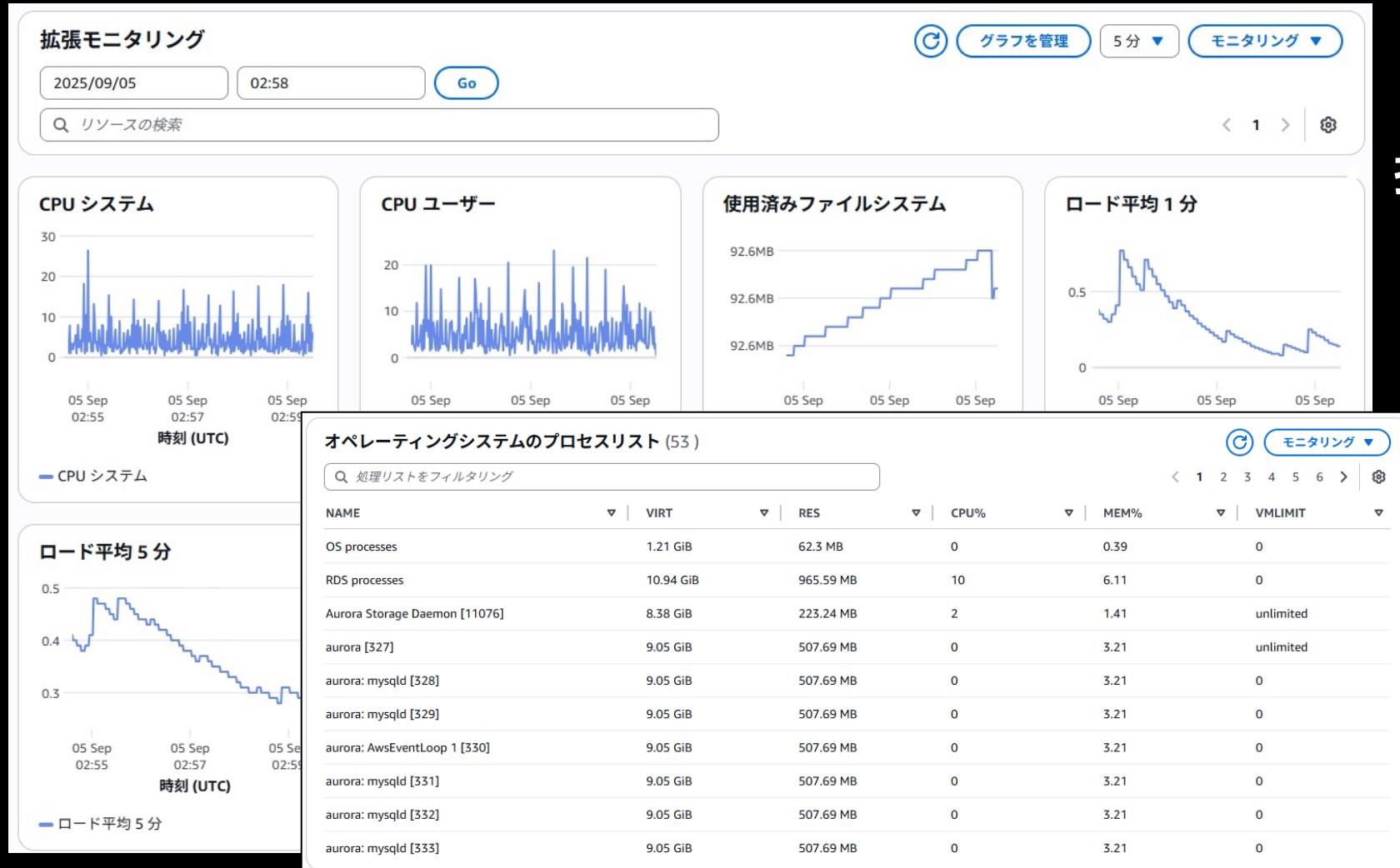
Below the editor are buttons for 'クエリの実行' (Run Query), 'キャンセル' (Cancel), '保存' (Save), and '履歴' (History). A note says 'Logs Insights QL クエリは、最大 60 分まで実行できます。' (Logs Insights QL queries can run up to 60 minutes). A message indicates 'Completed. Query executed for 1 log group.'.

The bottom section displays the results under 'Logs (647)'. It includes a histogram for 'Rows_examined' over time from 05:00 to 07:45. The results table has columns: '#', 'Time', 'query_time', 'Query', 'Rows_sent', and 'Rows_examined'. The first three rows of the table are:

#	Time	query_time	Query	Rows_sent	Rows_examined
1	2025-09-10T22:46:52.051677Z	2.725971	SELECT sql_no_cache max_temp,min_temp,station_name FROM weather WHERE max_...	1120	3197957 SET
2	2025-09-10T22:46:54.818286Z	2.736682	SELECT sql_no_cache count(id) FROM weather WHERE station_name = 'SANDBERG'...	1	3196837 SET
3	2025-09-10T22:46:37.584131Z	2.772714	SELECT sql_no_cache count(id) FROM weather WHERE station_name = 'SANDBERG'...	1	3196837 SET

- クエリを実行してスロークエリログを解析
例えば「Time」「Query_time」「Query」「Rows_sent」「Rows_examined」などの個々のフィールドを取得

Amazon RDS 拡張モニタリング



拡張モニタリング

- CloudWatch の標準メトリクスより詳細なOSのメトリクスを取得
- CPU使用率の内訳 (User/System/IO)
- ファイルシステムの使用率
- 最小で1秒ごとに各種メトリクスを取得
- プロセスごとのリソース使用量の取得など

Amazon RDS 拡張モニタリング

Db2、MariaDB、MySQL、Oracle、および PostgreSQL の主な OS メトリクス

グループ	メトリクス	コンソール名	説明
cpuUtilization	Nice	CPU Nice	最も低い優先順位で実行されているプログラムが使用中の CPU の使用率。
cpuUtilization	System	CPU システム	カーネルが使用中の CPU の使用率。
cpuUtilization	Total	CPU 合計	使用中の CPU の合計使用率。この値は nice 値を含みます。
cpuUtilization	User	CPU ユーザー	ユーザープログラムが使用中の CPU の使用率。
cpuUtilization	Wait	CPU 待機	I/O アクセスを待機中の CPU の未使用率。
swap	In	スワップイン	ディスクからスワップされたメモリの量 (キロバイト単位)。
swap	Out	スワップアウト	ディスクにスワップされたメモリの量 (キロバイト単位)。
processList	cpuUsedPc	CPU %	プロセスが使用中の CPU の使用率。
processList	memoryUsedPc	MEM%	プロセスが使用中のメモリの使用率。

参考

- 1) [Db2、MariaDB、MySQL、Oracle、および PostgreSQL の OS メトリクス](#)
- 2) [Microsoft SQL Server の OS メトリクス](#)

待機イベントの見方

データベースロード

データベースのパフォーマンスを評価するためのKPI

データベースロード → アクティブなセッション数(Active Sessions)

アクティブなセッションとは

- データベースのセッションは“アクティブ” / “アイドル”的なステータスがある
- さらに、アクティブなセッションは“CPU使用中”、“他の処理を待機中”的2つのステータス

データベースロードとは

- 1秒おきに“アクティブ”なセッションの詳細な情報をサンプリング
 - ステータス(CPU使用中、待機中(待機イベント^(*1))、SQL文、接続元ホスト、接続ユーザー)
- 単位時間^(*2)の“Average Active Sessions”とも言える

*1 各データベースの待機イベントの情報はAppendixを参照(参考となるリファレンスを掲載)

*2 データベースロードを選択する期間により平均を取得するための間隔は変動(最小で1秒間の平均値となる)

待機イベントについて(1)

待機イベントはセッションが待っているリソースを示します。

⇒ 例えば、待機イベント `io/socket/sql/client_connection` はスレッドが新しい接続を処理中であることを示します。

データベース全体の負荷状況の指標として利用できる**データベースロード(AAS)**をドリルダウンして待機イベントを分析することでパフォーマンスに影響を与える原因を知ることができます。

この待機イベントはデータベースエンジンごとに異なります。

- MariaDB および MySQL のすべての待機イベントの詳細については、MySQL ドキュメントの「[イベント待機サマリーテーブル](#)」を参照してください
- すべての PostgreSQL 待機イベントについては、PostgreSQL ドキュメントの「[PostgreSQL の待機イベント](#)」を参照してください
- すべての Oracle 待機イベントについては、Oracle ドキュメントの「[Descriptions of Wait Events](#)」を参照してください
- SQL Server のすべての待機イベントについては、SQL Server ドキュメントの「[待機の種類](#)」を参照してください

待機イベントについて (2)

Auroraの場合、MySQL と PostgreSQL に互換性があるため多くの待機イベントの意味を共有しています。しかし、一部 Aurora 独自の待機イベントもありますので、代表的な待機イベントは以下で確認できます。

- Aurora MySQL の代表的な待機イベントのリストについては、
[「Aurora MySQL の待機イベント」](#) を参照してください
- Aurora PostgreSQL の代表的な待機イベントのリストについては、
[「Amazon Aurora PostgreSQL のイベント」](#) を参照してください

Aurora MySQL の待機イベント

待機イベント名	説明
cpu	スレッドが CPU でアクティブな場合、または CPU を待っている際に発生します。
io/aurora_redo_log_flush	セッションは Aurora ストレージにデータを保持しています。通常、この待機イベントは Aurora MySQL の書き込み I/O オペレーション用です。詳細については、「 io/aurora_redo_log_flush 」を参照してください。
io/table/sql/handler	エンジンは、テーブルへのアクセスを待っています。このイベントは、データがバッファプールにキャッシュされているか、ディスク上でアクセスされているかにかかわらず、発生します。詳細については、「 io/table/sql/handler 」を参照してください。
synch/cond/innodb/row_lock_wait	複数のデータ操作言語 (DML) ステートメントが同じデータベース行に同時にアクセスしようとしています。詳細については、「 synch/cond/innodb/row_lock_wait 」を参照してください。
synch/mutex/innodb/buf_pool_mutex	スレッドがメモリ内のページにアクセスするために InnoDB バッファプールのロックを取得したときに発生します。
synch/mutex/innodb/fil_system_mutex	セッションがテーブルスペースのメモリキャッシュへのアクセスを待っているときに発生します。
synch/mutex/innodb/trx_sys_mutex	大量のトランザクションで高いデータベースアクティビティがある場合に発生します。
synch/sxlock/innodb/hash_table_locks	セッションは、バッファプール内のページを見つけることができませんでした。エンジンは、ファイルを読み取るか、バッファプールの最も長い時間使われていない (LRU) リストを変更する必要があります。バッファキャッシュのサイズを増やし、関連するクエリのアクセスパスを改善することを検討してください。

参照:

[待機イベントを使用した Aurora MySQL のチューニング](#)

Aurora PostgreSQL の待機イベント

待機イベント名	説明
CPU	スレッドが CPU でアクティブになっているか、CPU の待機中に発生します。
Client:ClientRead	Aurora PostgreSQL がクライアントからのデータ受信を待っているときに発生します。
Client:ClientWrite	Aurora PostgreSQL がクライアントへのデータ書き込みを待っているときに発生します。
IO:DataFileRead	バックエンドプロセスが必要なページをストレージから読み込む際に、ページが共有メモリで使用できないために接続が待機したときに発生します。
IO:XactSync	データベースが、Aurora ストレージサブシステムが通常のトランザクションのコミットを承認するか、準備されたトランザクションのコミットまたはロールバックの承認を待っているときに発生します。
Lock:advisory	PostgreSQL アプリケーションがロックを使用して、複数のセッションにわたるアクティビティを調整するときに発生します。
Lock:extend	バックエンドプロセスがリレーション拡張のためにロックするのを待機中に、他のプロセスが同じ目的でそのリレーションをロックしているときに発生します。
Lock:Relation	他のトランザクションによって現在ロックされているテーブルまたはビューに対するロックを取得するためにクエリが待っているときに発生します。
Lock:transactionid	トランザクションが行レベルロックを待っているときに発生します。
Lock:tuple	バックエンドプロセスがタプルのロック取得を待機中の場合に発生します。
LWLock:buffer_content	セッションがデータページのメモリ内への読み取りまたは書き込みのために待機中、そのデータページが他のセッションで書き込むためにロックされているときに発生します。

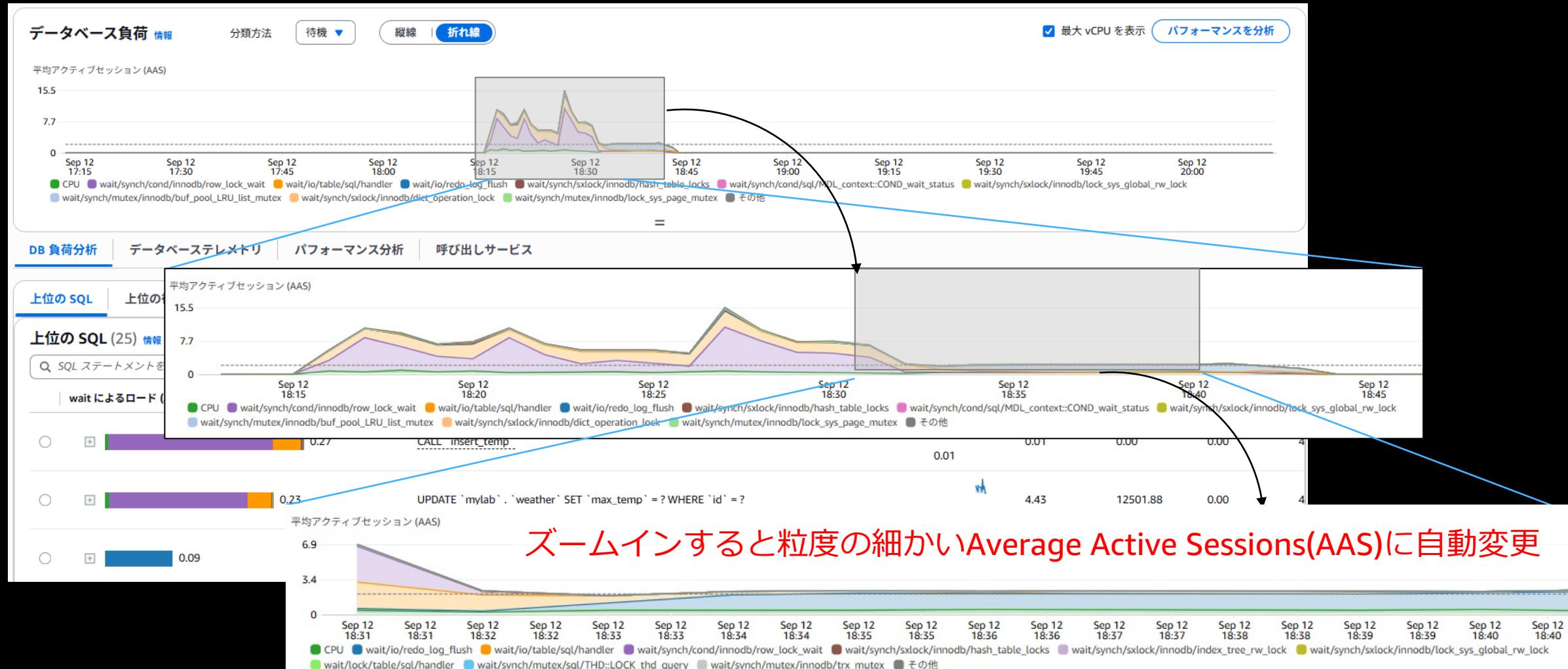
参照:

[Aurora PostgreSQL の待機イベント](#)

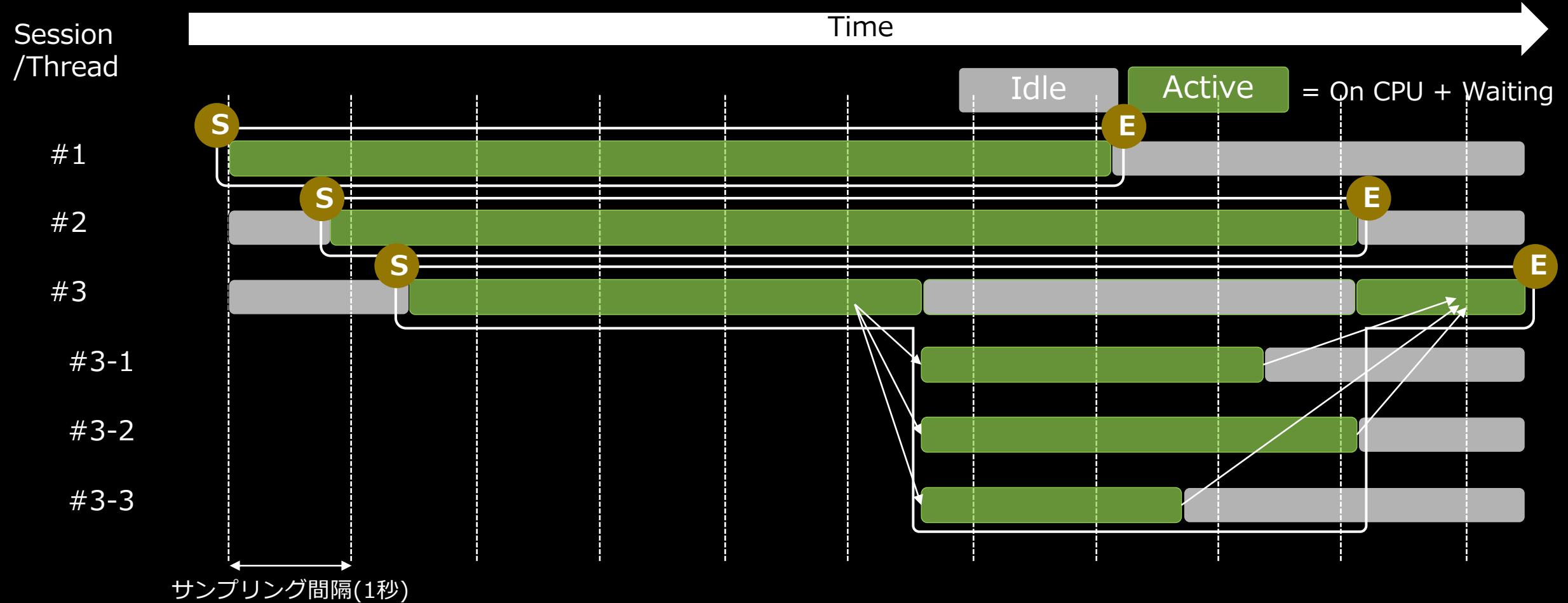
[Aurora PostgreSQL の待機イベントでのチューニング](#)

Database Insights ダッシュボードでの待機イベント確認

対象時間帯を選択して頂く事で、選択した時間帯に発生していた待機イベント詳細を確認可能

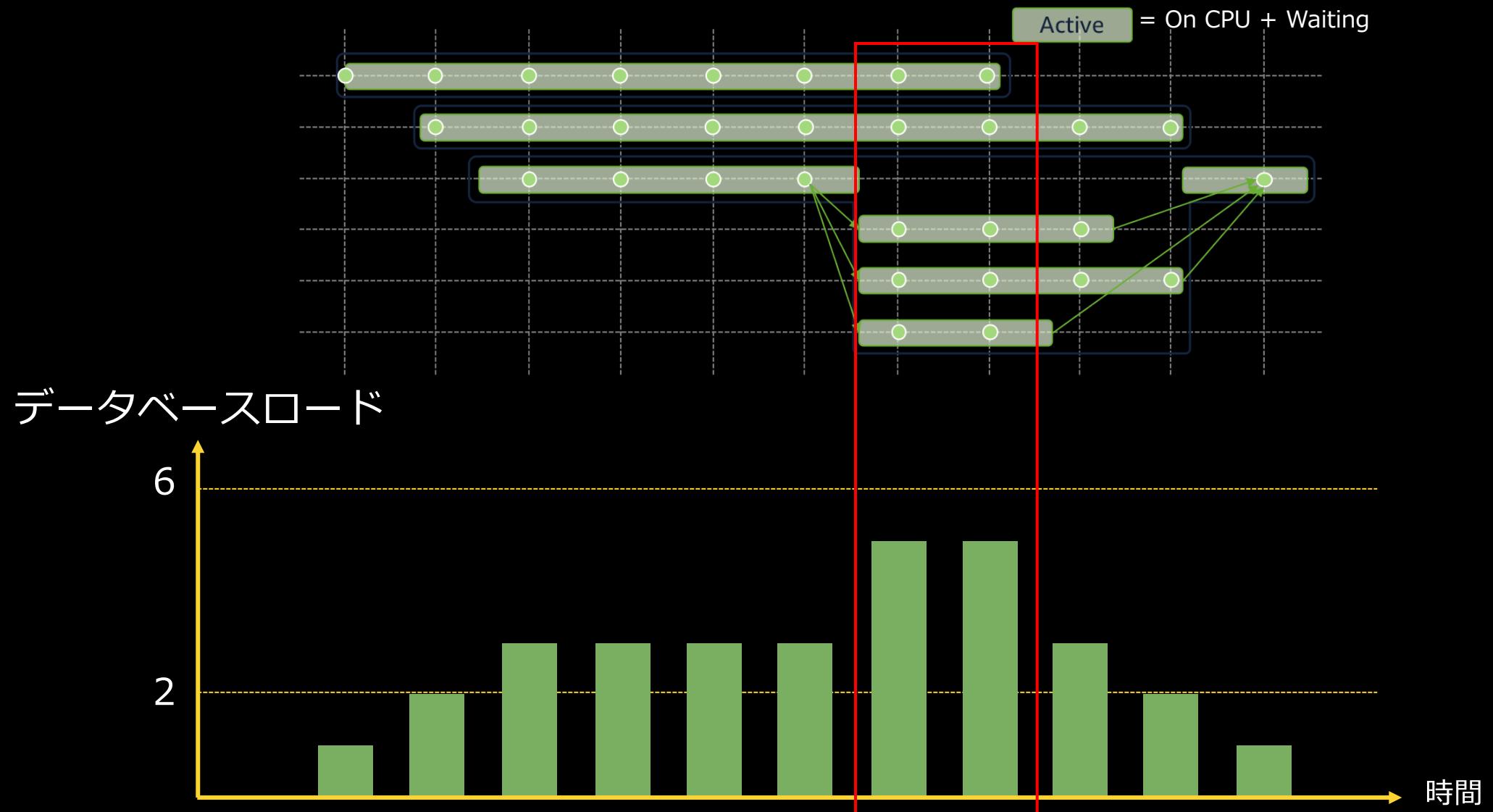


データベースロード

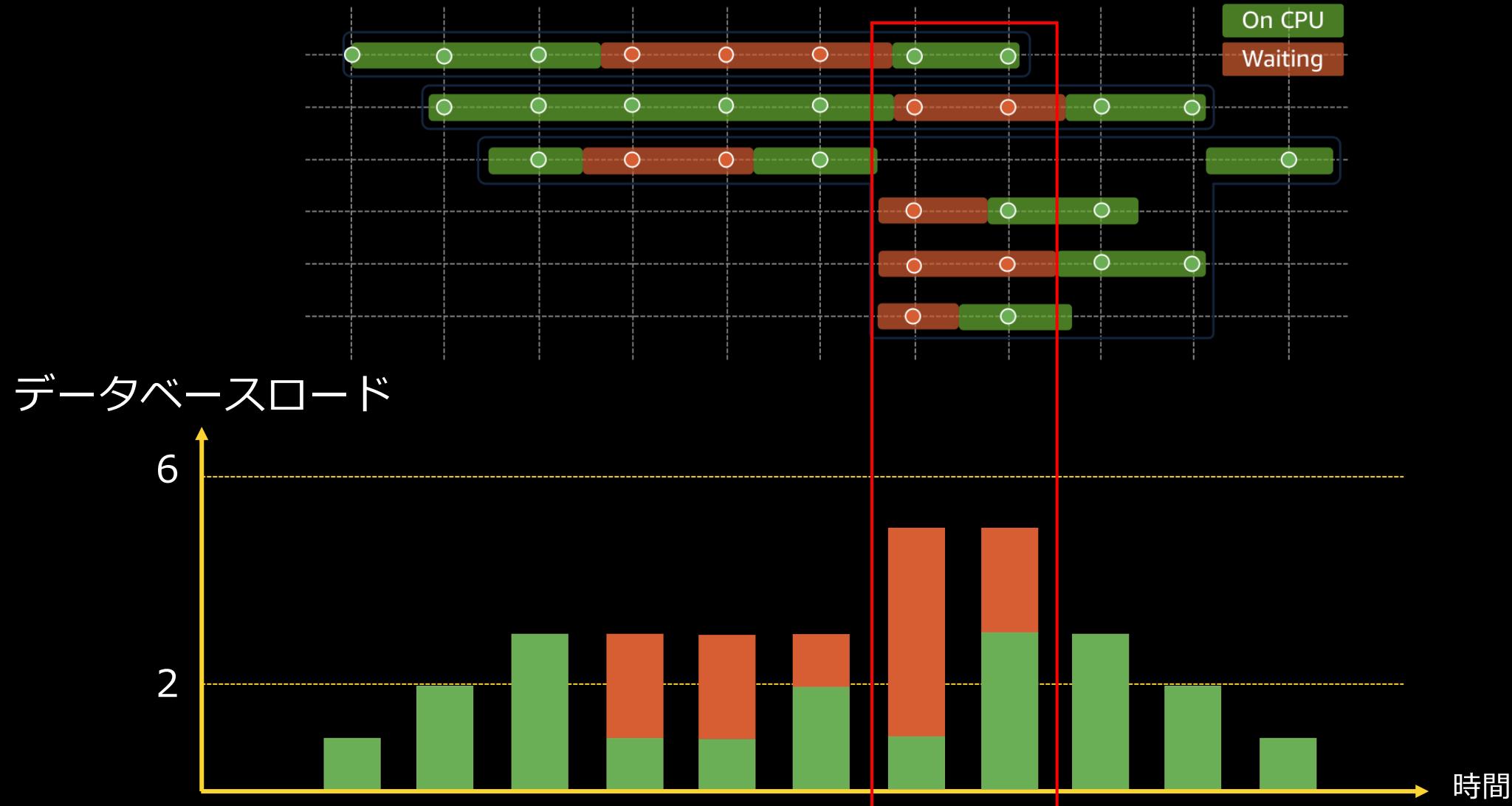


Database Insights は1秒のサンプリング間隔で**セッションの状態**を取得

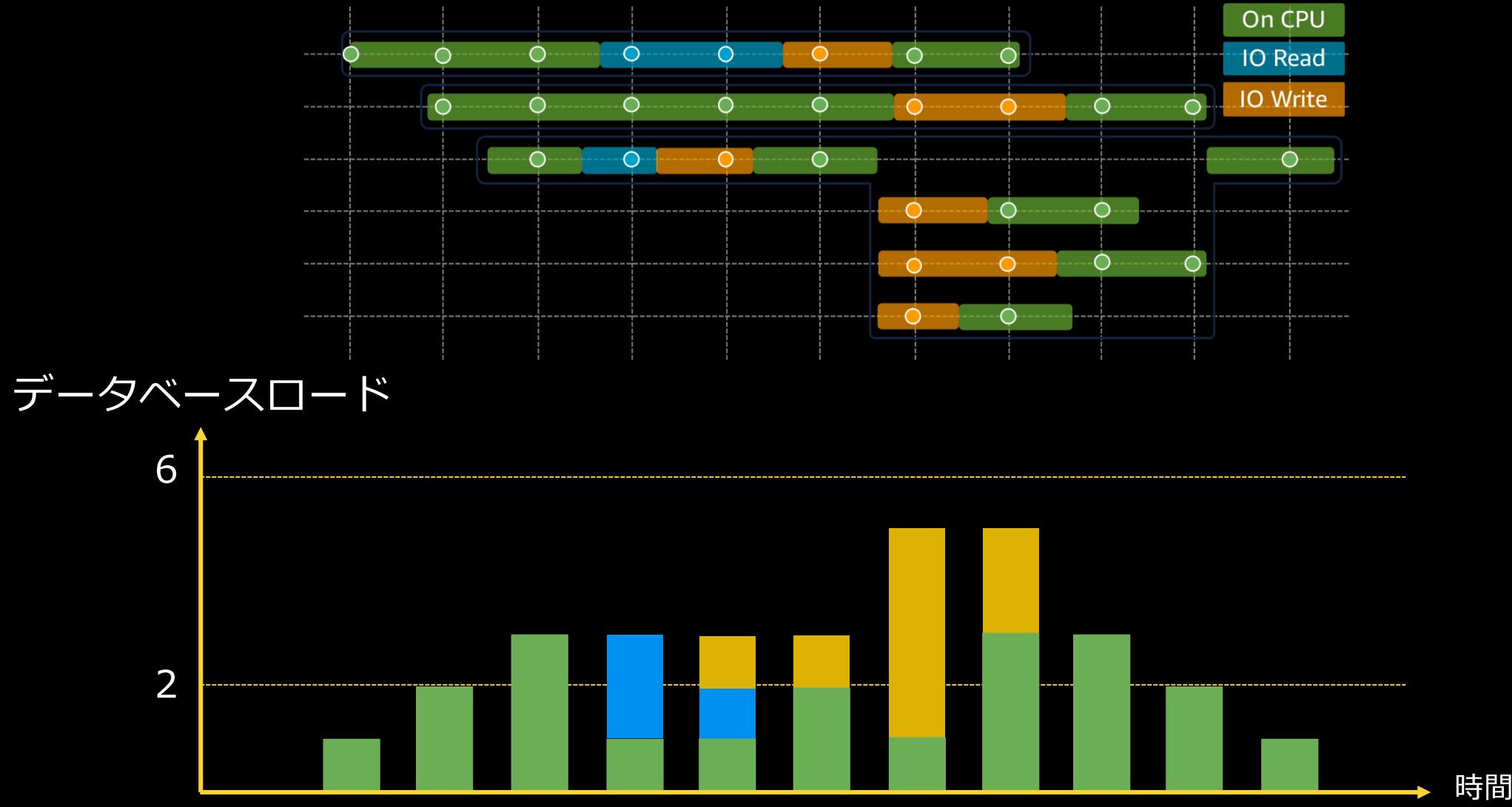
データベースロード = Average Active Sessions



アクティブセッションの状態を分析



さらに”待機イベント”で分析



データベースロードで問題の発見

データベースロード(AAS) ≈ 0

データベースロード(AAS) >= # of vCPUs

データベースロード(AAS) >> # of vCPUs

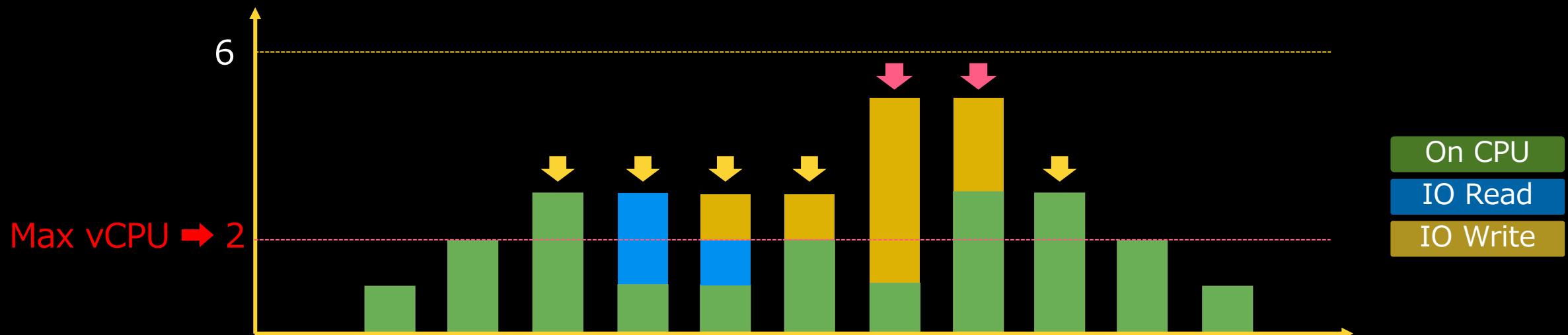
基本的にデータベースが使用されていない

パフォーマンス問題の可能性がある

パフォーマンス問題

※AAS = Average Active Sessions

→ パフォーマンス問題の有無は、データベースロードがインスタンスの最大vCPUを超えていているかどうかが基準



まとめ

- ・ チューニングのための検知・分析として活用できるツール
 - CloudWatch
 - 拡張モニタリング
 - CloudWatch Database Insights
- ・ **CloudWatch Database Insights** では
データベース全体の健全性を管理する包括的な機能を提供
 - メトリクス、ログ、スローSQLクエリ、イベント、OSプロセスを単一のビューに統合
 - アドバンストモードではオンデマンドパフォーマンス分析や実行プランの分析※1
など高度な分析も可能
- ・ データベースロード(AAS)を元に待機イベントを分析し問題を把握する

※1 サポートされているDBエンジンで利用可能 参考：[Database Insights のモード](#)

Thank you!

