Amazon Aurora Deep Dive (Part-1)

Amazon Aurora アーキテクチャについて

矢木 覚 (やぎ さとる)
アマゾン ウェブ サービス ジャパン 合同会社
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

内容についての注意点

- 本資料では 2022年9月1日時点のサービス内容および価格について説明しています。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ご利用される場合、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.

アジェンダ

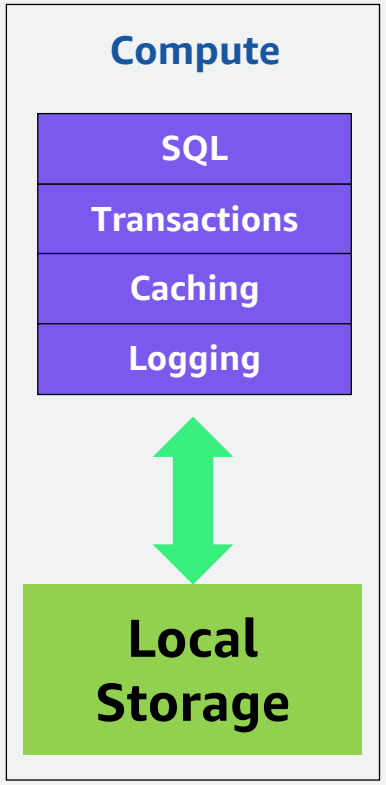
- Amazon Auroraの概要
- Auroraのアーキテクチャ
- ミッションクリティカルシステムに対応するAuroraの機能
- Amazon Aurora Serverless v2

Amazon Auroraの概要

従来のデータベースアーキテクチャー

全ての構成要素が1つにまとまっている
モノリシックな構成

ローカルストレージでパフォーマンスと可用性、
耐久性を両立させる構成

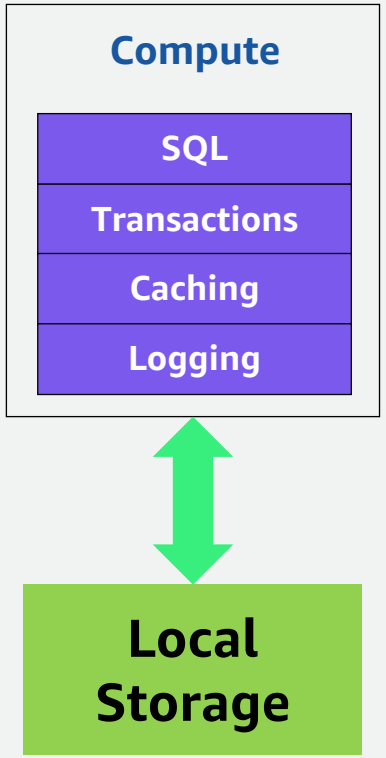


従来のデータベースアーキテクチャー

コンピュートからストレージを分離した構成

その後、コンピュートとストレージのレイヤーを分離し、
それぞれでスケール、カスタマイズ、管理を可能にした。

しかし、個々の構成要素は変わっていない。

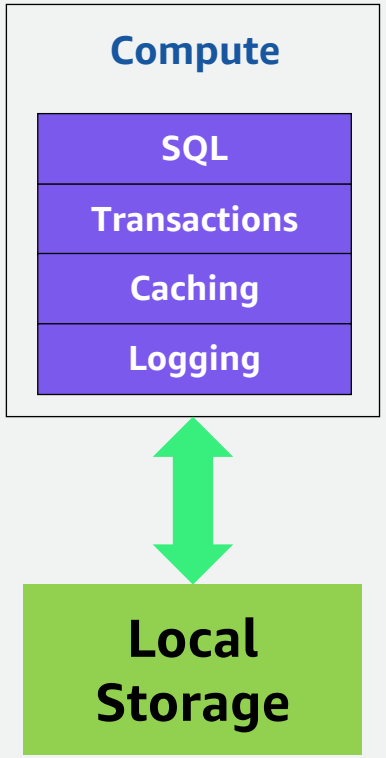


クラウドでのデータベース

コンピュートとストレージは異なるライフサイクルを持ち、全く異なる要件を満たす必要がある

- インスタンス障害や再配置の可能性
- インスタンス停止の可能性
- インスタンスのスケールアップ/スケールダウン
- インスタンスのスケールアウト/スケールイン

コンピューティングとストレージは
スケーラビリティ、可用性、耐久性を考慮して
最適に分離される必要がある



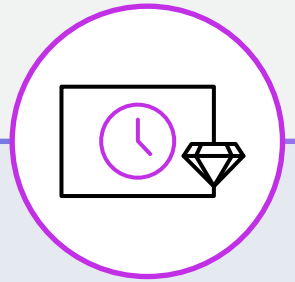
Amazon Aurora

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低コストで実現

優れた性能と拡張性

標準的なMySQL,PostgreSQL
に比べパフォーマンス効率が
良い
15個のリードレプリカ

高可用性と耐久性



フォールトトレラント
自己回復ストレージ
3つのAZで6つのデータコピー
グローバルデータベースとクロ
スリージョンレプリケーション

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による管理：
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、設定、
バックアップは不要

Amazon Aurora

サポートしているメジャーバージョンとエンジン

MySQL

- Aurora MySQL 1 (MySQL 5.6互換)
- Aurora MySQL 2 (MySQL 5.7互換)
- Aurora MySQL 3 (MySQL 8.0互換)

PostgreSQL

- Aurora PostgreSQL 10
- Aurora PostgreSQL 11
- Aurora PostgreSQL 12
- Aurora PostgreSQL 13
- Aurora PostgreSQL 14

※ Auroraのサポートポリシーについ てはApendixをご参照ください
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.VersionPolicy.html#Aurora.VersionPolicy.Engines

Amazon Aurora Innovations

RE-IMAGINING DATABASE FOR CLOUD

1. スケールアウト, 分散, マルチテナントデザイン
2. AWSサービスを活用したサービスオリエンテッドアーキテクチャー
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, マルチテナントデザイン

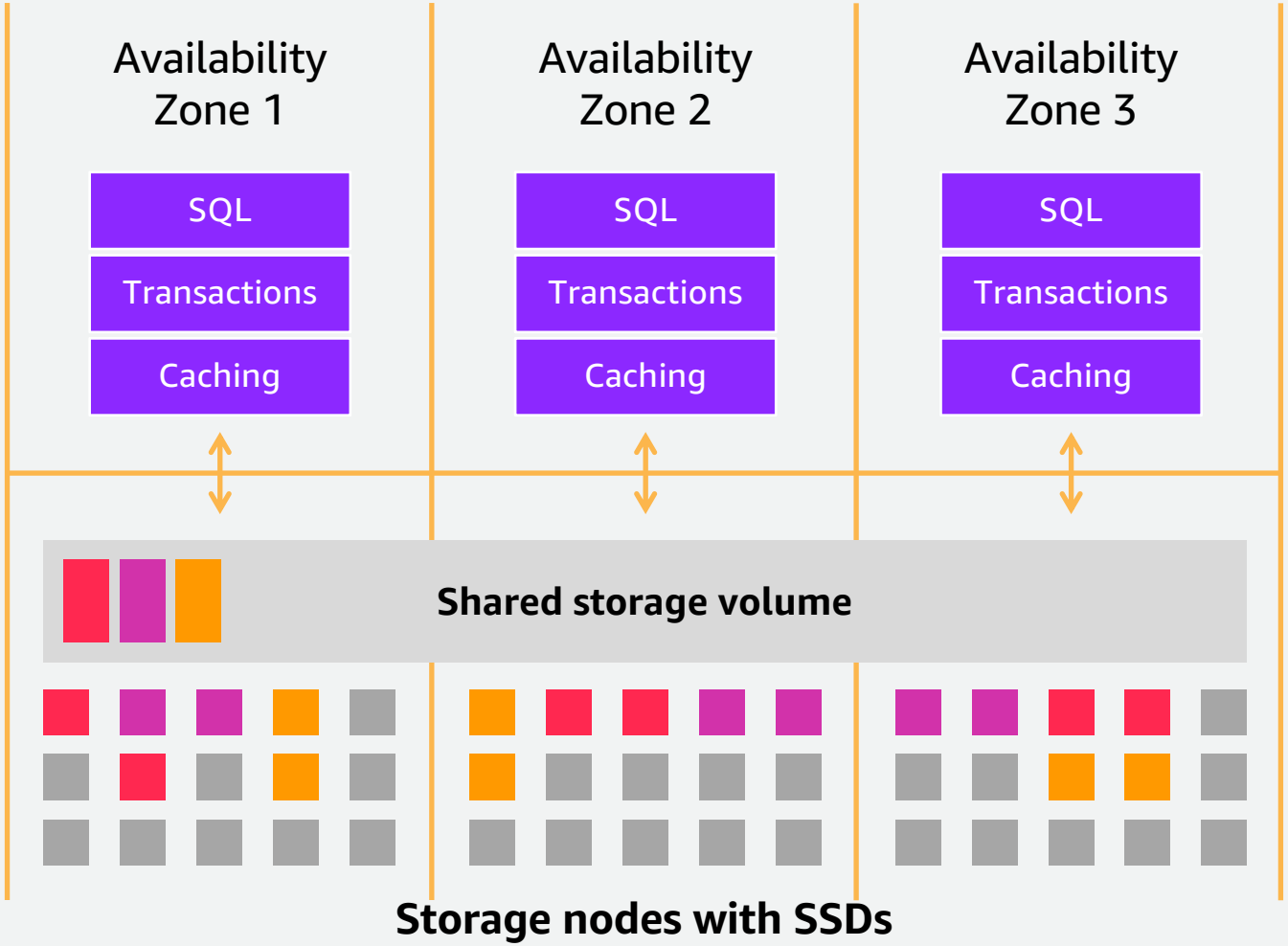
AURORA SCALE-OUT, DISTRIBUTED ARCHITECTURE

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ーゾーンに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ッドアーキテクチャー

Lambda function

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

AWS Identity and Access Management

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

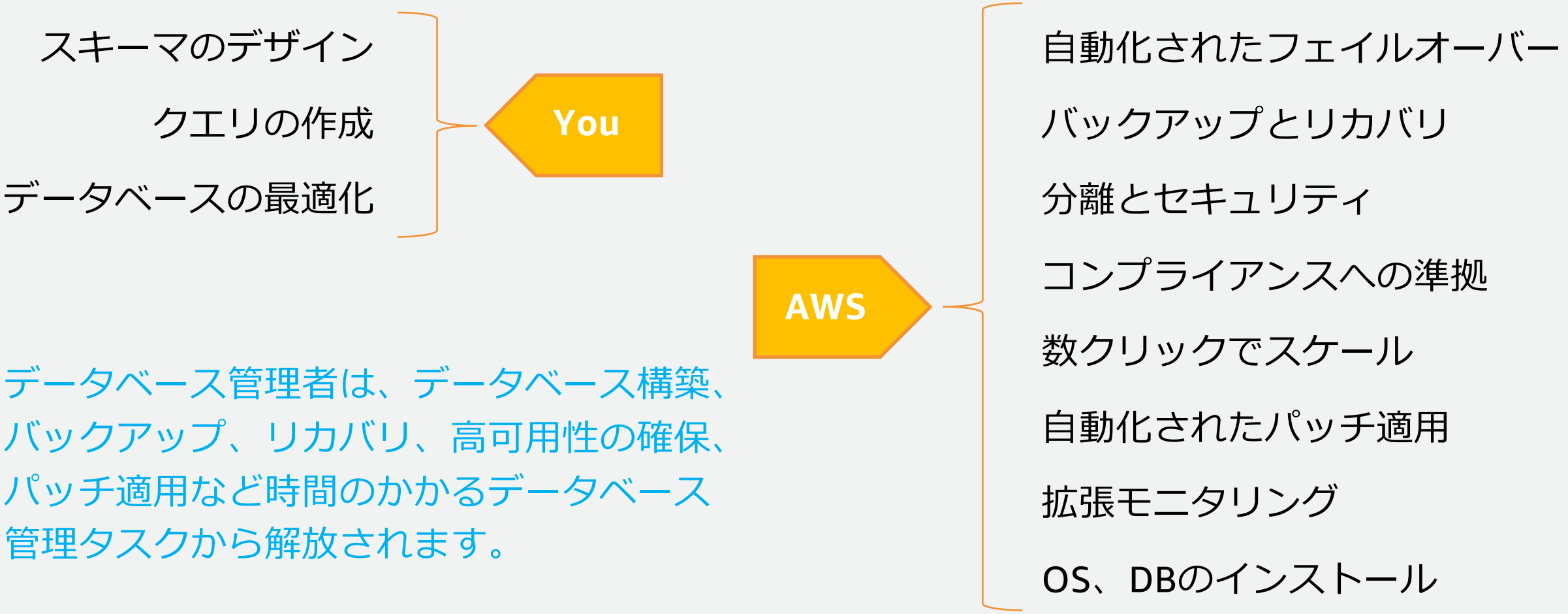
Amazon S3

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

Amazon CloudWatch

システムメトリックスやログをCloudWatchへアップロード

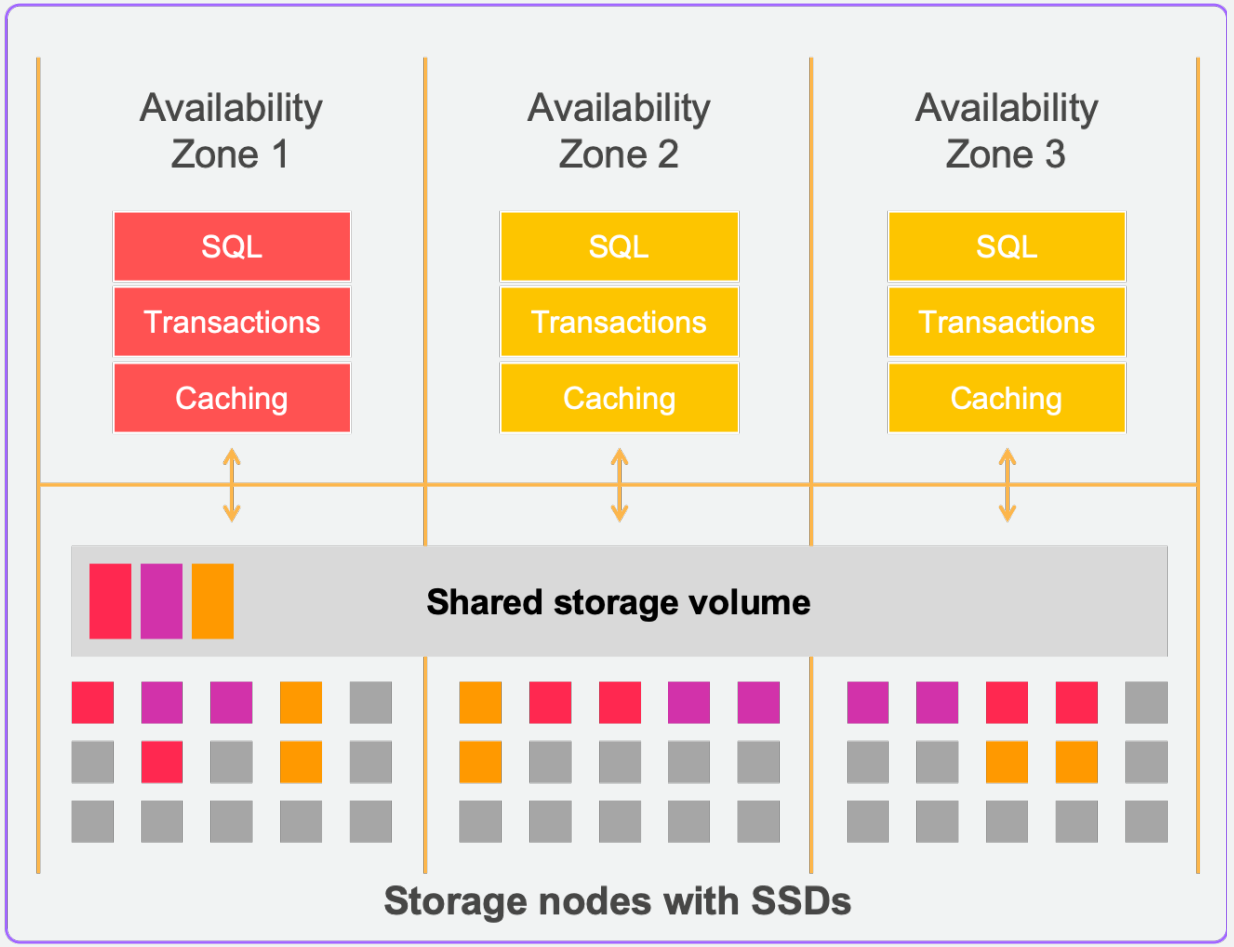
自動化されたタスク – 完全マネージド・サービス

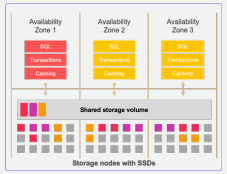


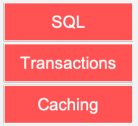
データベース管理者は、データベース構築、バックアップ、リカバリ、高可用性の確保、パッチ適用など時間のかかるデータベース管理タスクから解放されます。

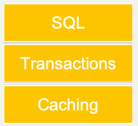
Auroraのアーキテクチャ

Auroraを構成するコンポーネント



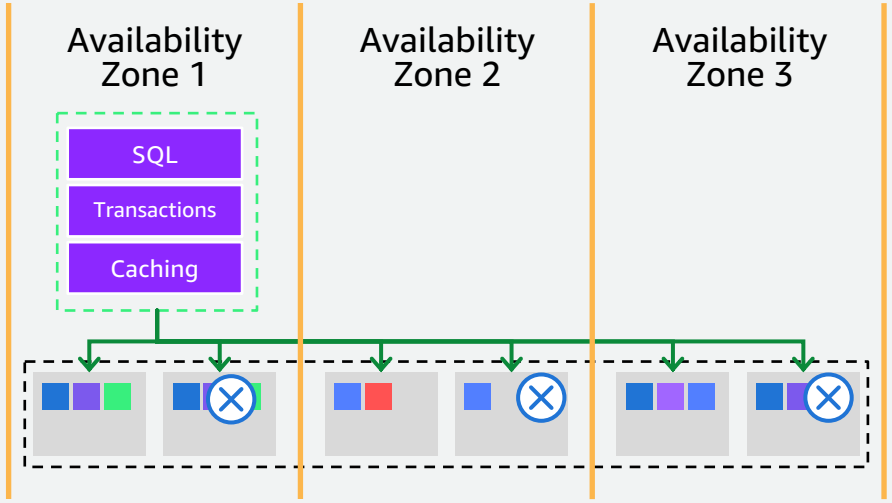




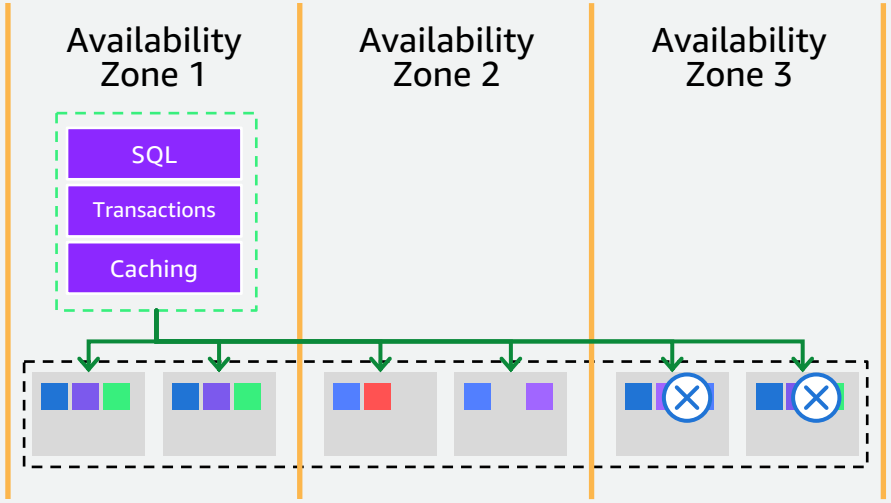


- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うマスターインスタンス
- **Aurora レプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - プライマリインスタンスもレプリカも同じクラスターボリュー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

Auroraストレージ - 自動修復、耐障害性



Read availability



Write availability

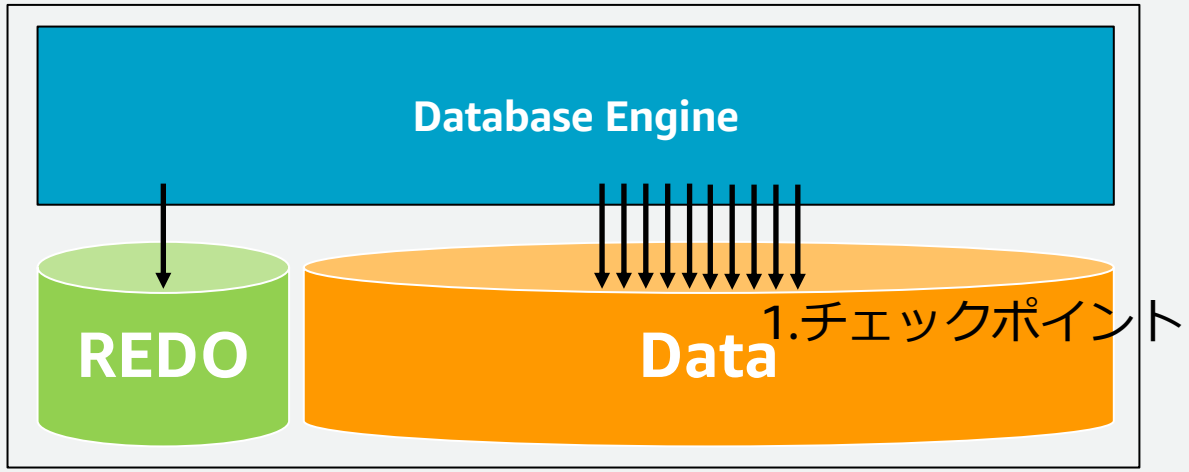
データは非同期で、6つのノードすべてに並行で書き込み

書き込みには4/6ノードのクォーラム、読み込みには3/6クォーラムが用いられる

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

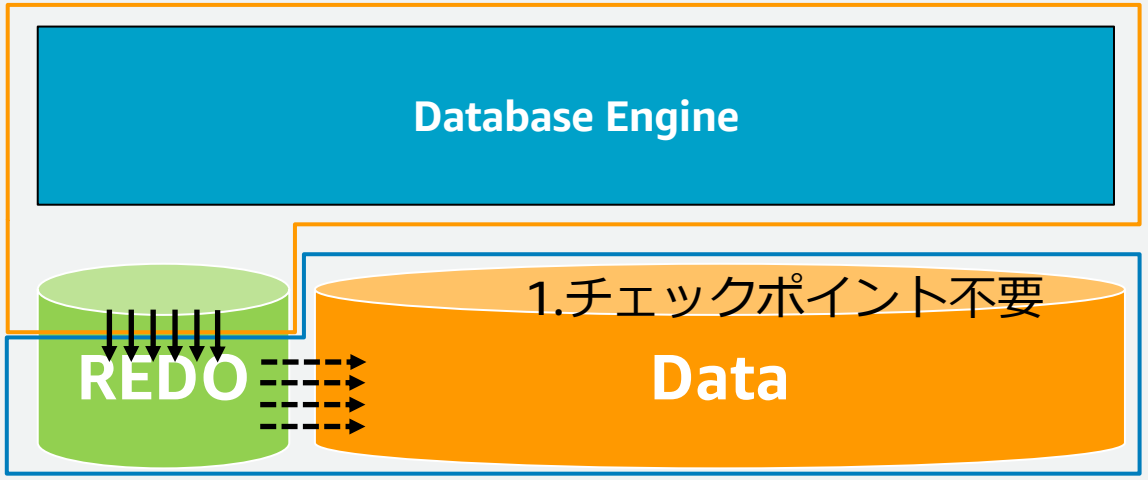
Auroraストレージ - チェックポイント不要

Traditional Database



2.シーケンシャルなREDOログ書き込み

Amazon Aurora



2.非同期、非順序でのREDOログ書き込み
3.クォーラムによる分散

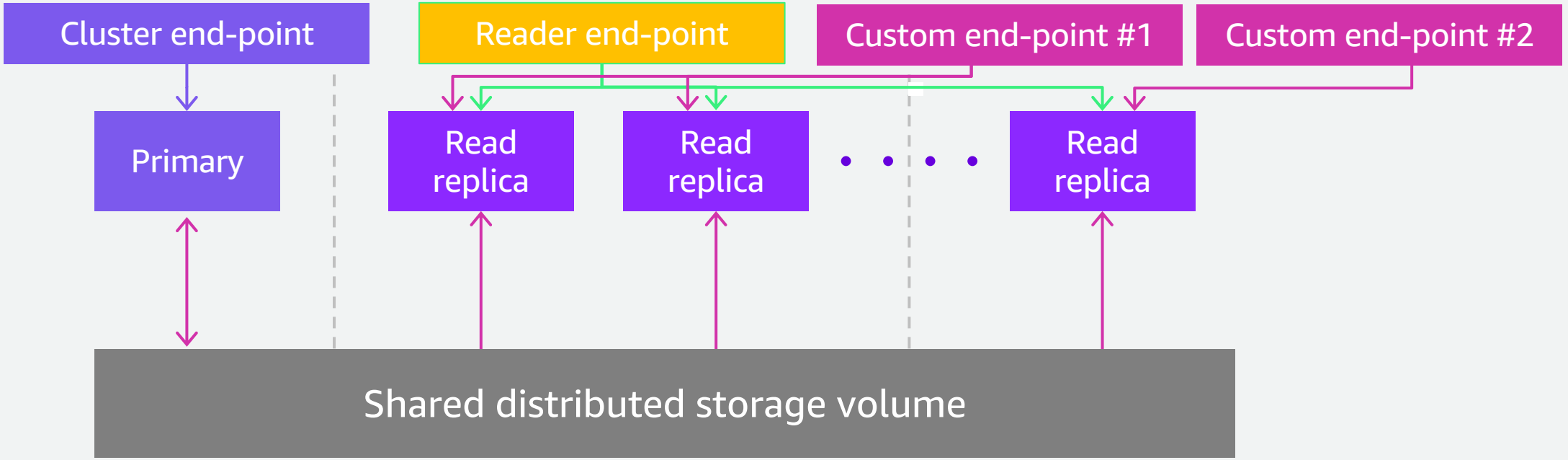
- チェックポイントが存在しない(I/O削減によりパフォーマンスの安定性が高い)
- 非同期、非順序でのログ書き込み(ログ書き込みのスループットの向上)

Auroraストレージ - 自動化された管理

最大128TBのストレージ – 10GB単位で自動拡張

- 最大128 TB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
- 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

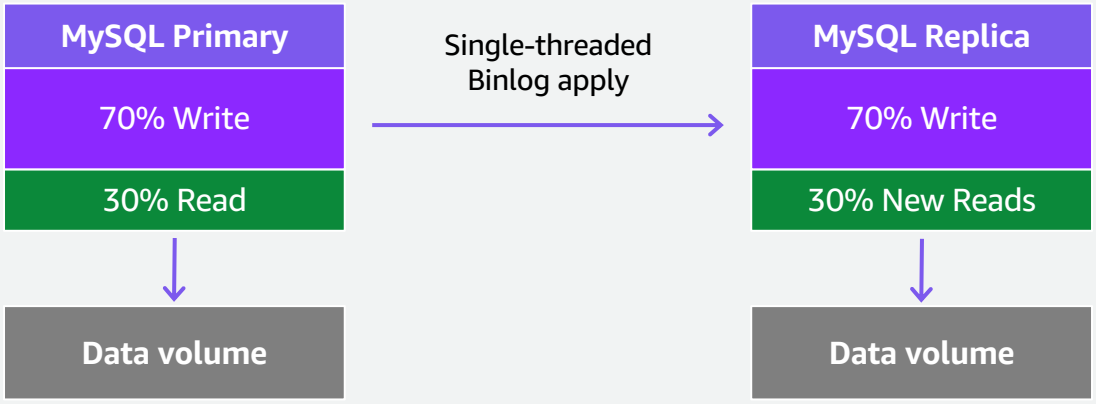
Auroraレプリカ、エンドポイント



3つのアベイラビリティーゾーンで最大15個の昇格可能なリードレプリカ
リードレプリカは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
ライターとリーダー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Auroraレプリカの違い

MySQL/PostgreSQL read scaling

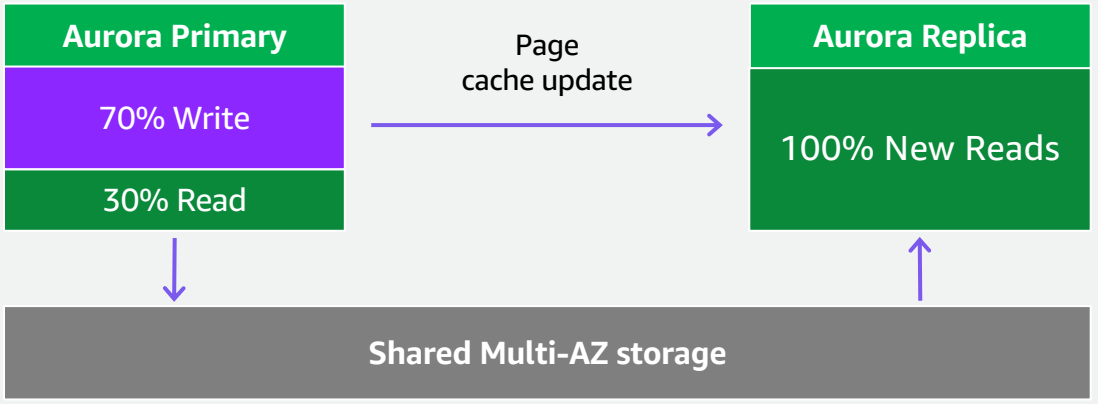


差分変更を用いた論理レプリケーション

マスターと同等の書き込みワークロード

独立したストレージ

Amazon Aurora read scaling

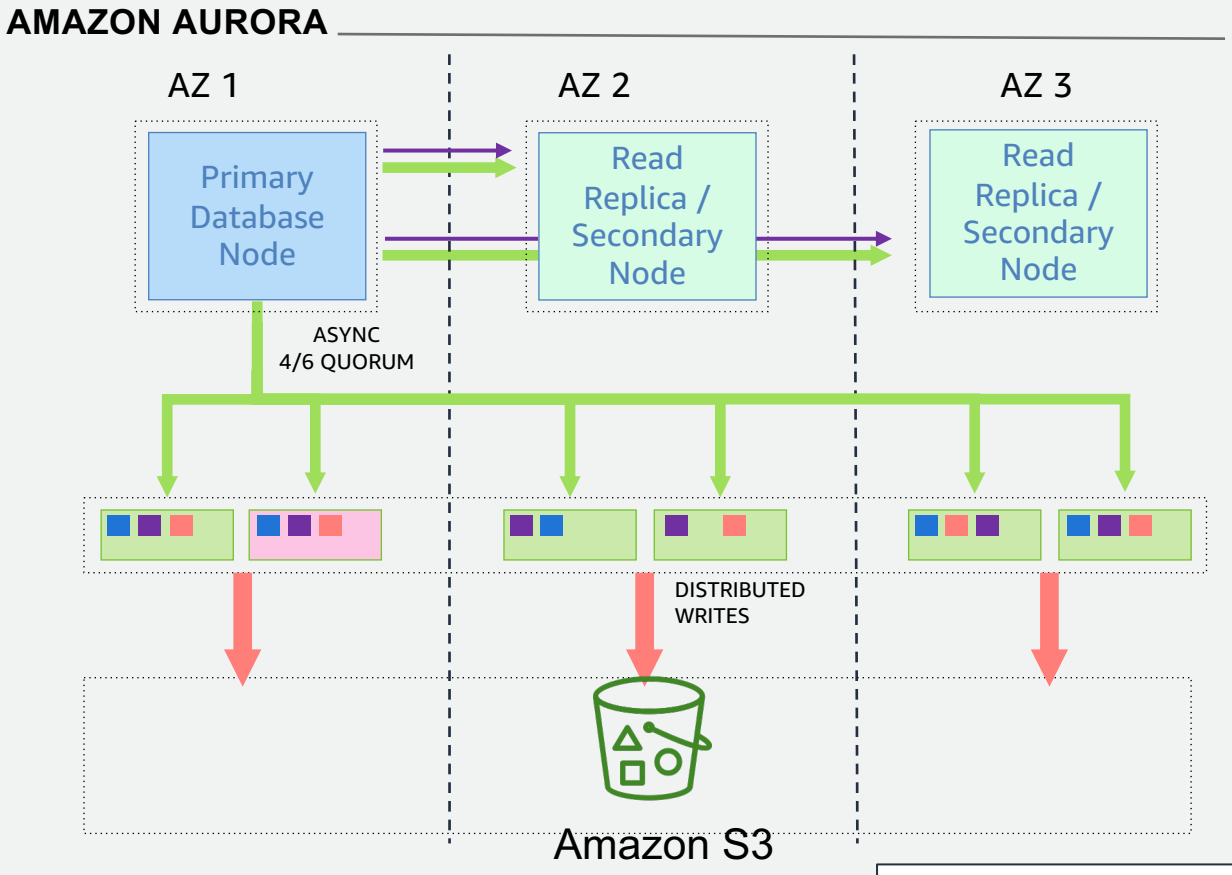


差分変更を用いた物理レプリケーション

レプリカに書き込みは発生しない

共有ストレージ

IO traffic in Aurora (MySQL)



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

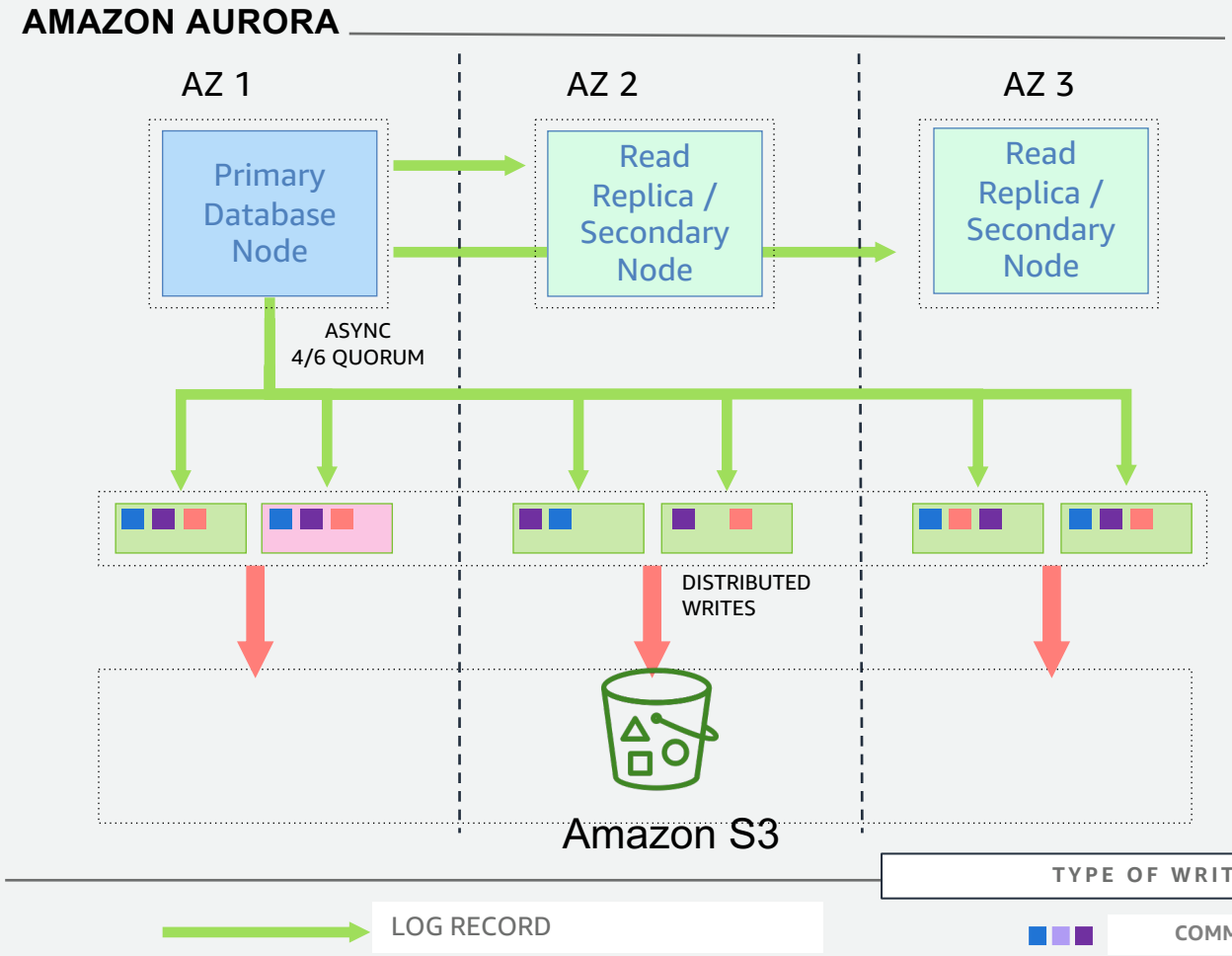
OBSERVATIONS

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

sysbenchを用いたベンチマークテストにおいて、MySQL のコミュニティエディションに比べて、35倍のパフォーマンス向上と7.7倍のI/O回数の削減

IO traffic in Aurora (PostgreSQL)



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

Aurora DBインスタンス

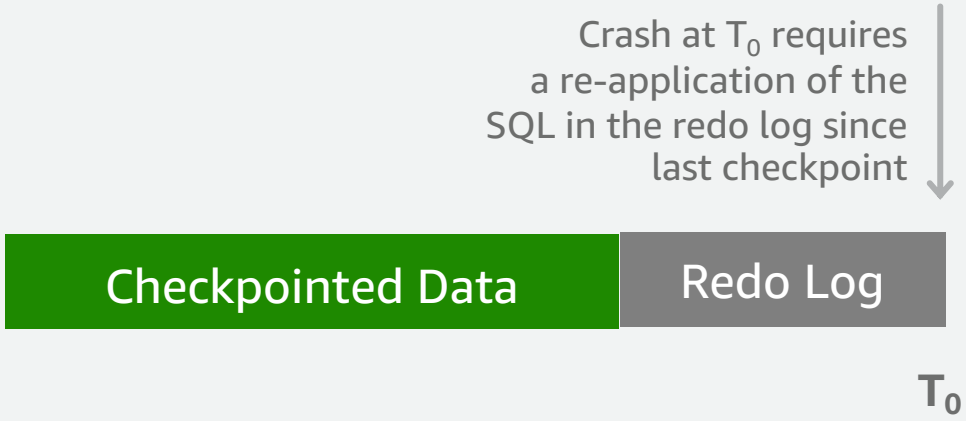
高速なクラッシュリカバリー

Traditional database

最後のチェックポイント以降のログを再生する必要がある

通常、チェックポイント間の5分

MySQLの場合、処理はシングルスレッドで多数のディスクアクセスが必要

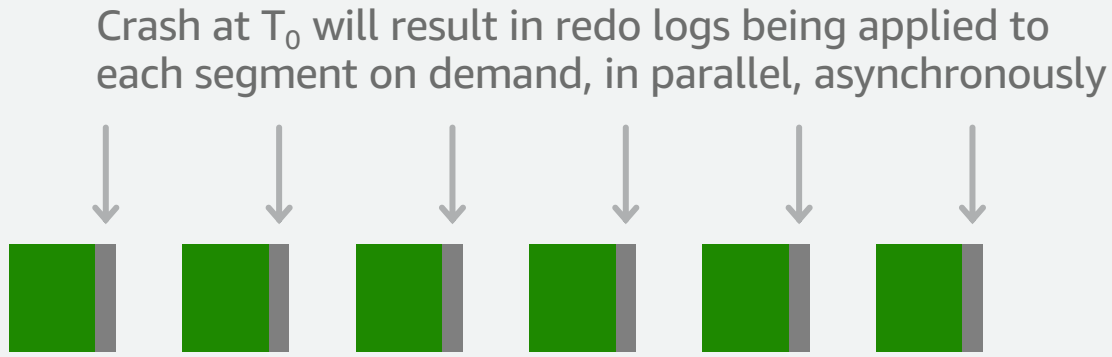


Amazon Aurora

Aurora ストレージは、ディスク読み取りの一部としてオンデマンドでREDOレコードを再生

並列、分散、非同期

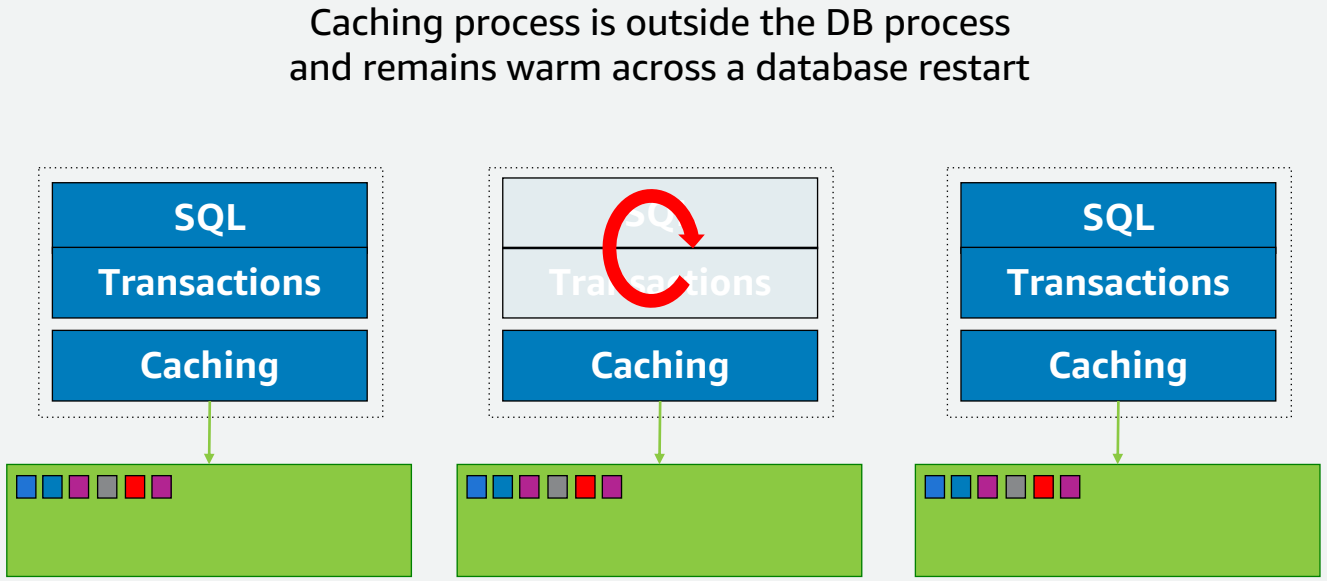
スタートアップのリプレイなし



Aurora DBインスタンス

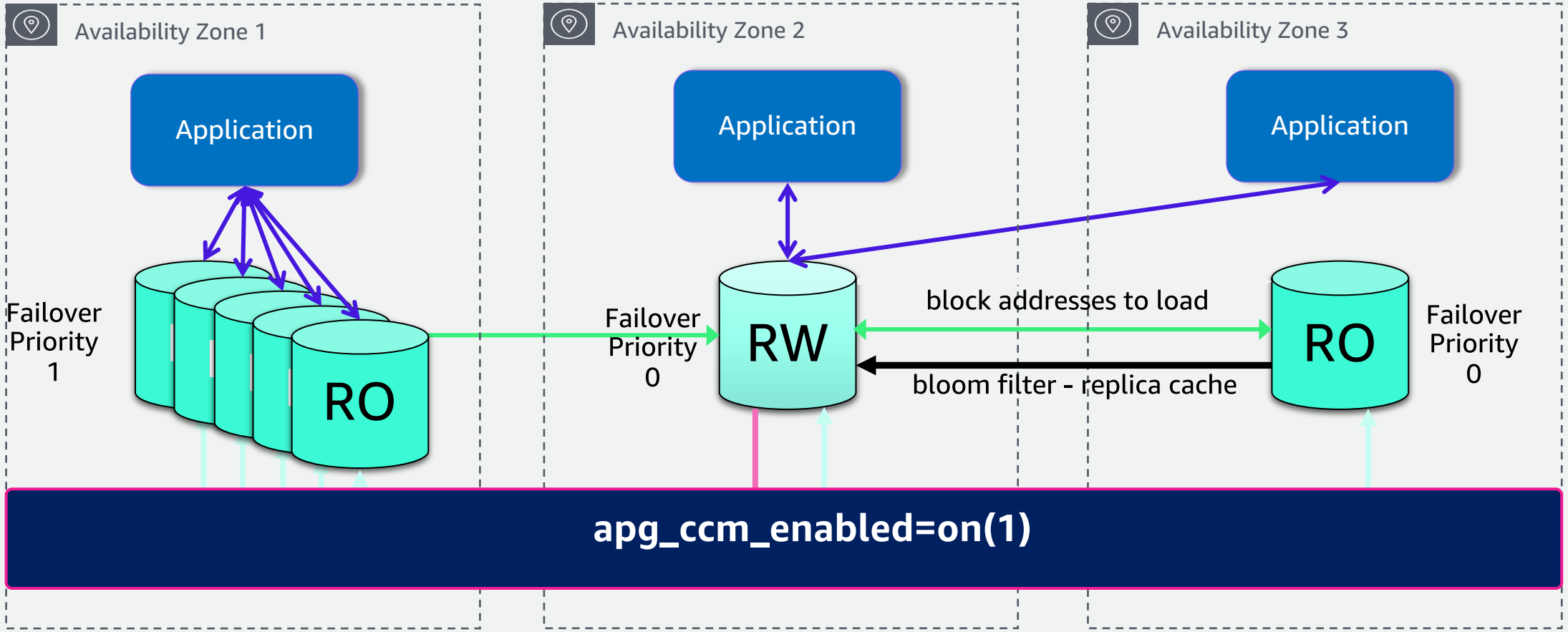
存続可能なキャッシュ

- データベースのプロセスからキャッシュを分離
- データベースの再起動が発生してもキャッシュはWarm状態を維持
- 再起動後でもオンキャッシュの処理を素早く再開可能
- 高速クラッシュリカバリー + 存続可能なキャッシュ = DB障害から素早く簡単なリカバリー



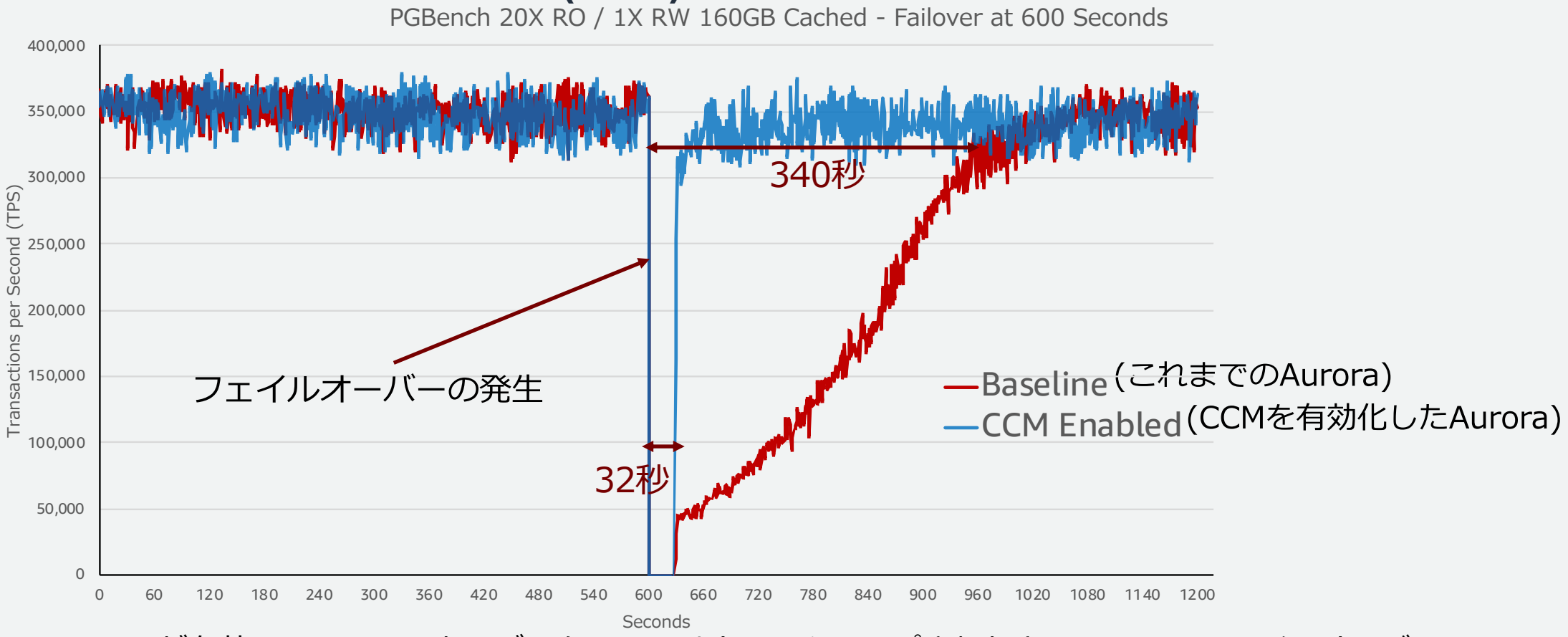
Aurora DBクラスター

CLUSTER CACHE MANAGEMENT (CCM)



Aurora DBクラスター

CLUSTER CACHE MANAGEMENT (CCM)



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

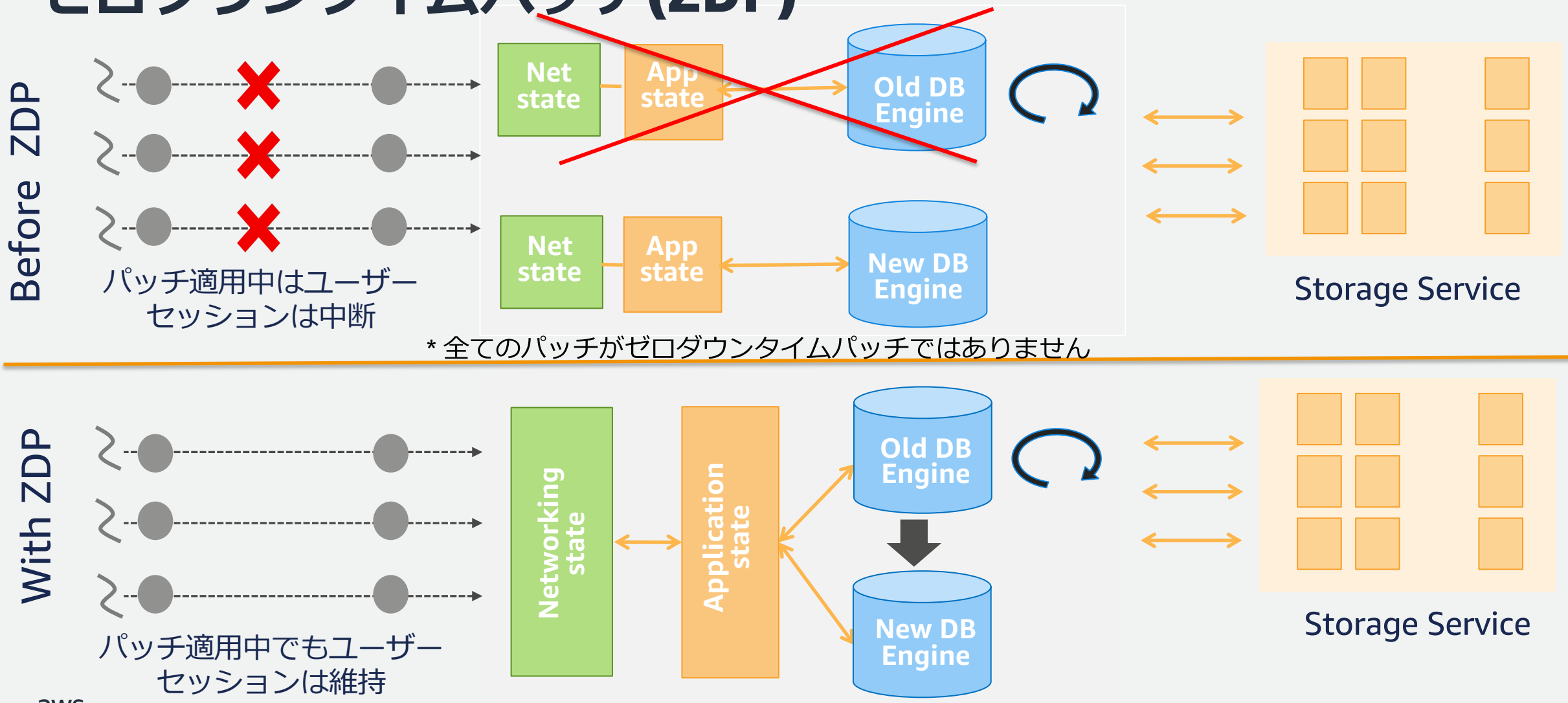
ミッションクリティカル
システムに対応するAuroraの機能

ミッションクリティカルシステムに対応するAuroraの機能

エンタープライズシステムの要求に応えるAURORAの解決策

1. データベースソフトウェアへのパッチ適用
   → ゼロダウンタイムパッチ
2. 大量データのコピーや再構成
   → 高速なデータベースクローン作成
3. DBAのミスによるデータベースのリストア
   → Auroraバックトラック
4. 災害
   → Auroraグローバルデータベース(replication)
5. SQL実行プランの変動によるアプリケーションへの影響
   → Query Plan Management

ゼロダウンタイムパッチ(ZDP)



高速なデータベースのクローン作成

重複するストレージのコストなしでデータベースのコピーを作成(CoW)

クローンの作成は瞬時に行われ、データはコピーはしない

クローンボリュームにデータを書き込む際に、オリジナルデータのコピーが行われる

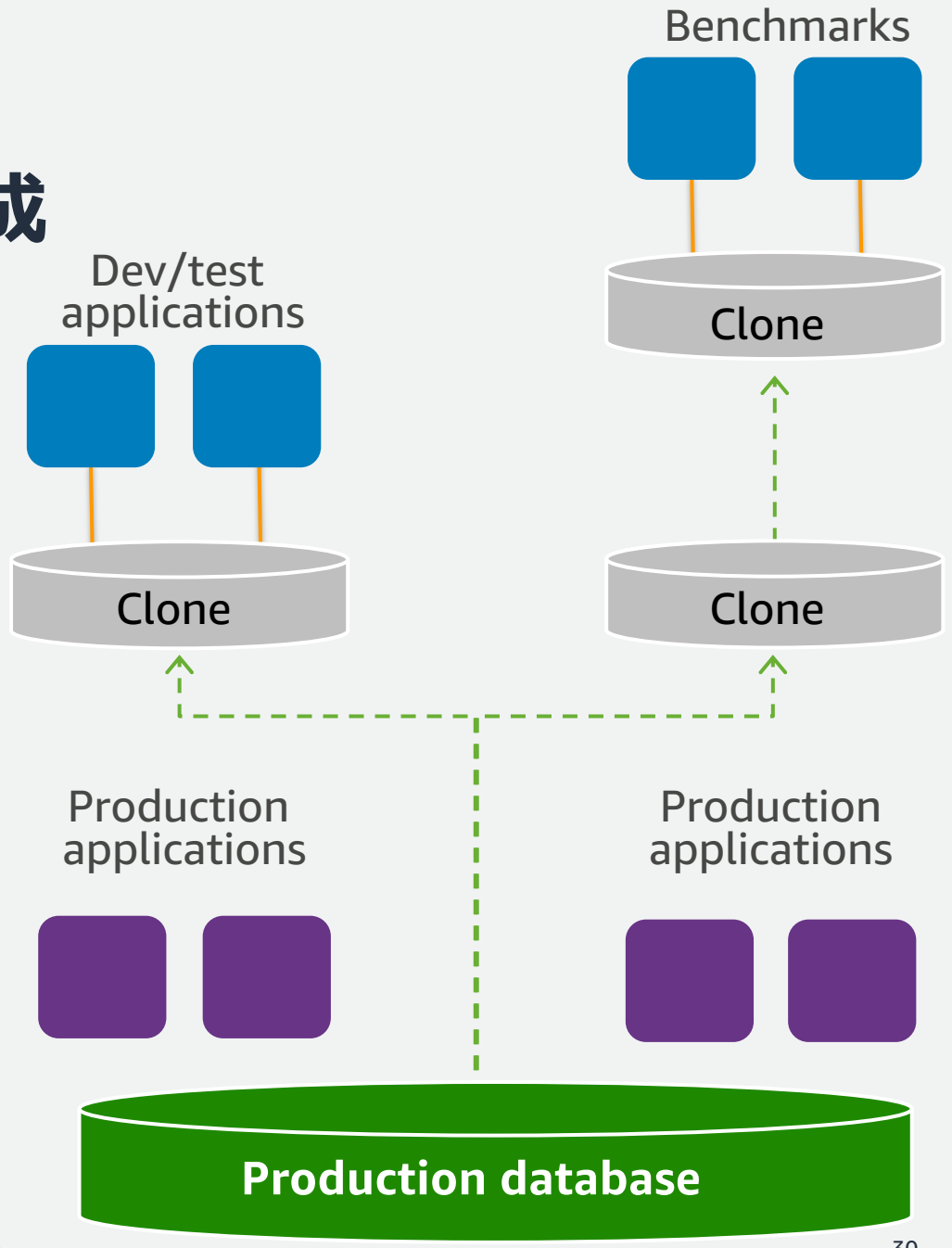
最大15個のクローンを作成可能

ユースケース

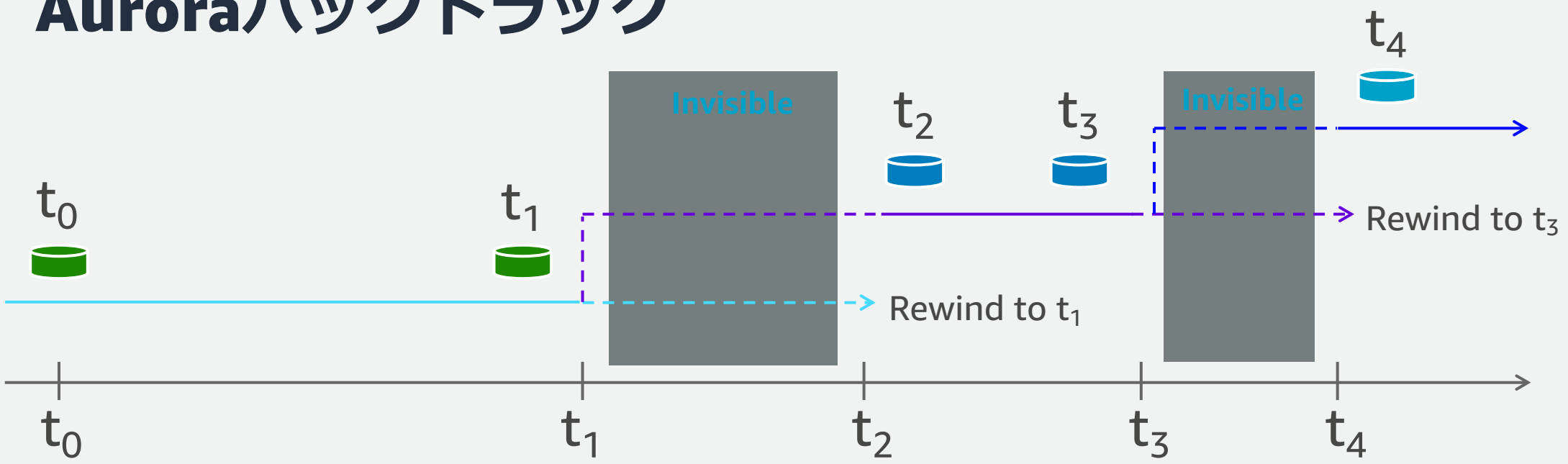
テスト環境用に実稼働DBのクローンを作成する

データベースを再編成する

本番機に影響を与えることなく、分析のためにポイントインタイムスナップショットを保存する



Auroraバックトラック



バックトラックは、バックアップからリカバリーを実施せずにデータベースを特定の時点に戻します

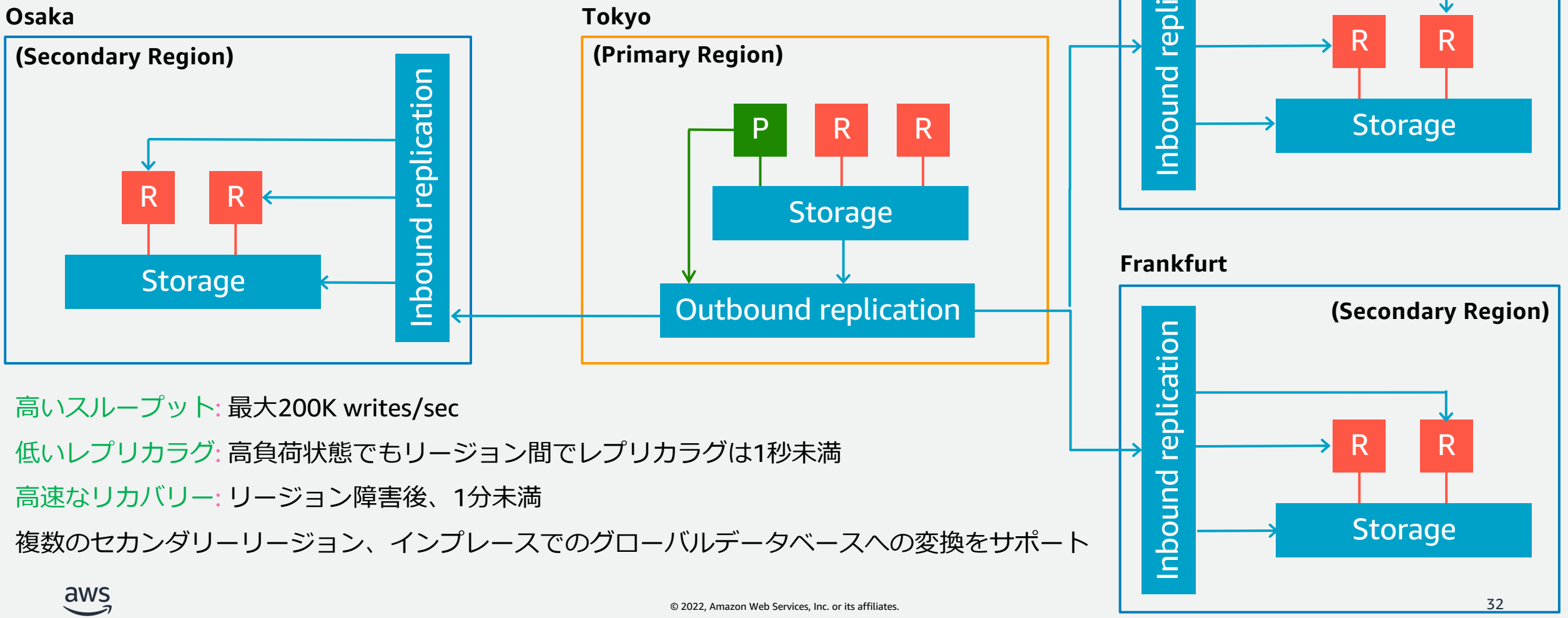
意図しないDMLまたはDDL操作から復旧

バックトラックは破壊的ではないので、複数回バックトラックを実行して、適切なポイントを見つけることが可能

QAにも役立つ (テスト実行の間にDBを巻き戻す)

Auroraグローバルデータベース

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

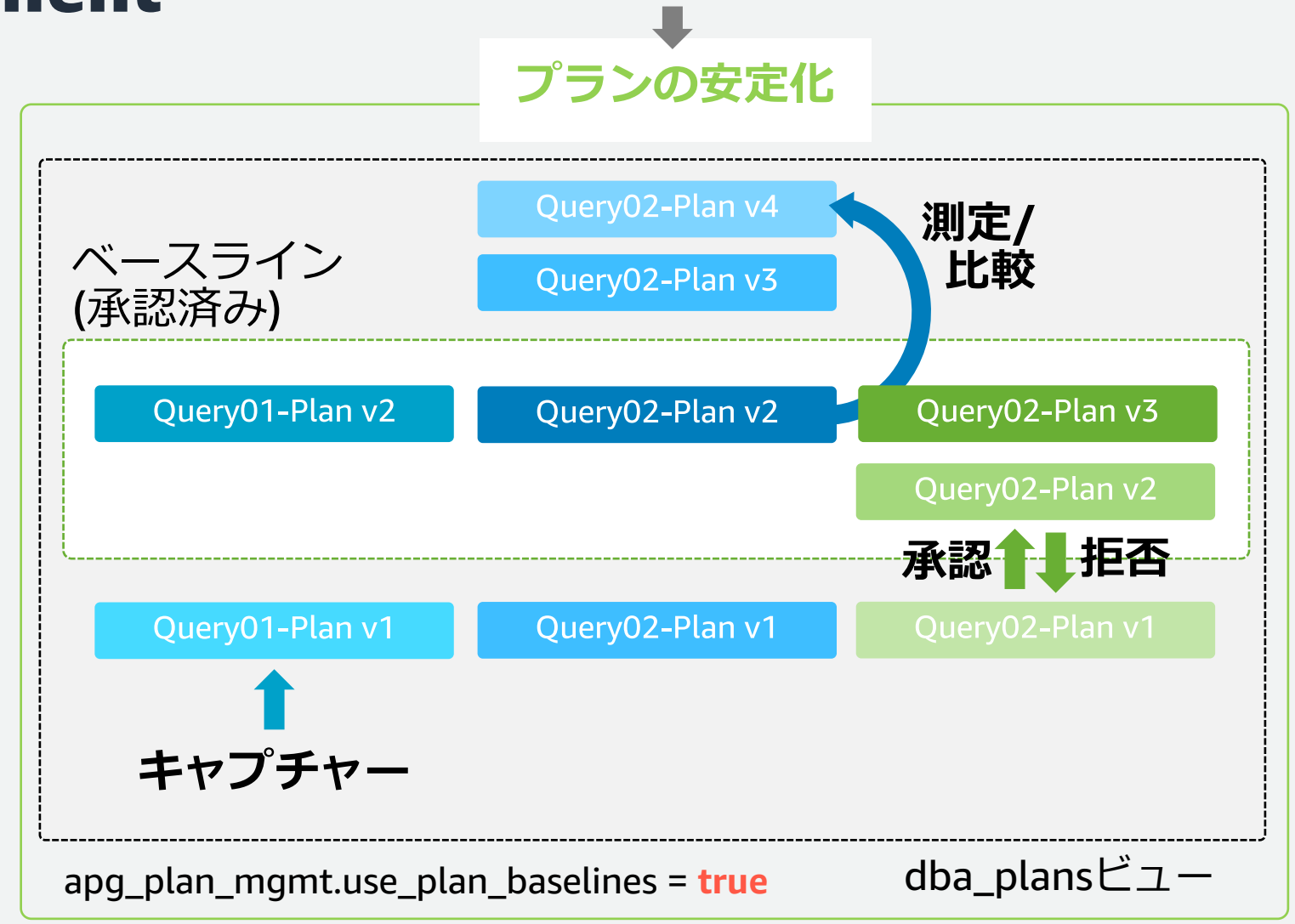
高速なリカバリー: リージョン障害後、1分未満

複数のセカンダリーリージョン、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

Query Plan Management

統計情報の変化　　環境(パラメータ)の変化

バインド変数の変化　　アップグレード



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)

Amazon Aurora Serverless v2

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**

v2は負荷に応じて拡張し、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

Amazon Aurora Serverless v2

サポートしているメジャーバージョンとエンジン

MySQL 8.0

- Aurora MySQL 3.02.0 以降
  - Window関数
  - アトミック オンラインDDL
  - SKIP LOCKED / NOWAIT オプション
  - JSON 関連機能の向上

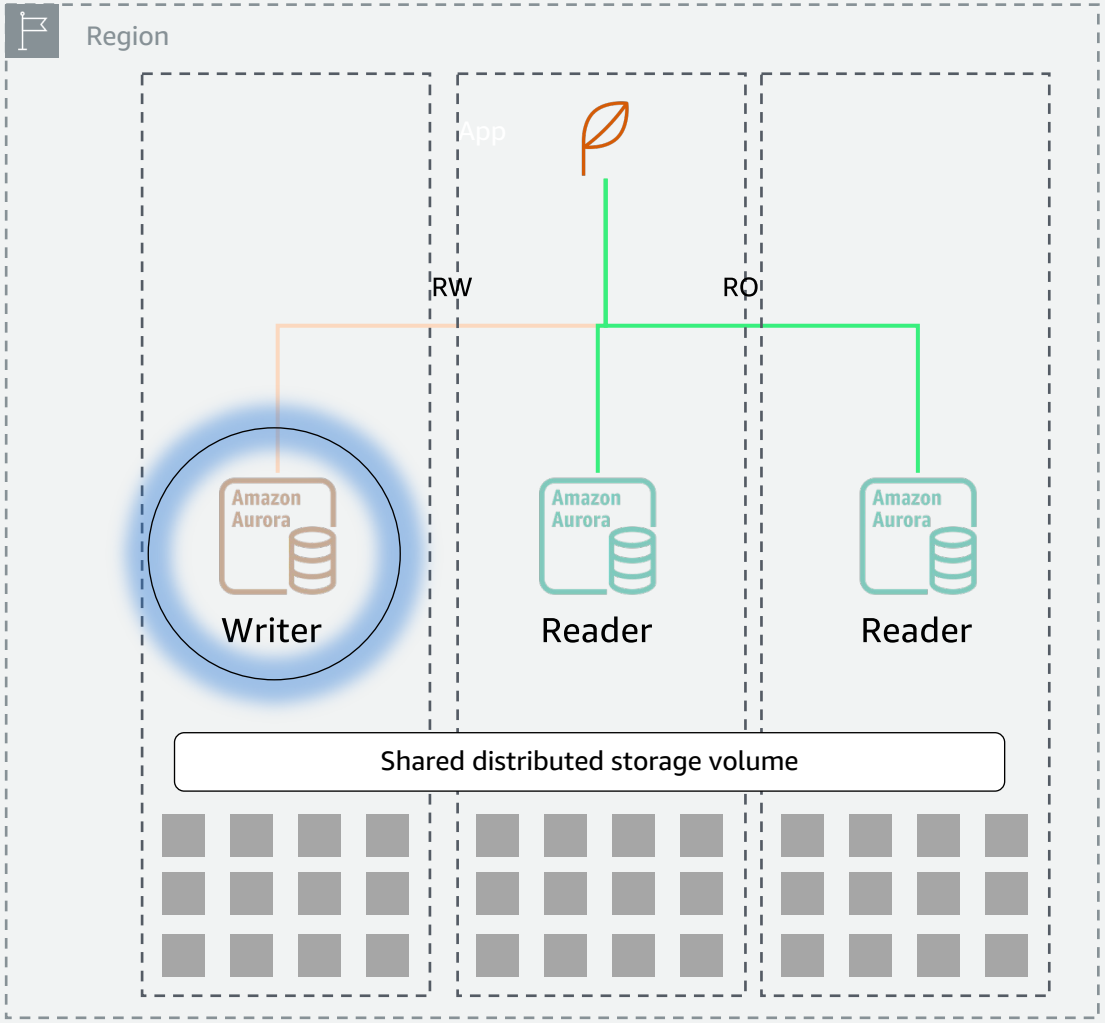
PostgreSQL 13・14

- Aurora PostgreSQL 13.6 以降, 14.3以降
  - Bツリーインデックス内の重複排除
  - パーティションテーブルのパフォーマンス向上
  - Incremental sortによるソートの高速化
  - Indexのバキュームの並列化

Aurora Serverless v2のスケーリングの特徴

インスタンスのスケール

- CPU/Memory/Networkの使用量などからシームレスにスケールしセッションの状態、バッファープールの内容も維持される
- v2ではv1と異なりScaling Pointを取ることなしに必要な時に即座にスケール可能
- v2ではv1と異なり0.5 ACUの粒度で最大128 ACUまでスケールアップ可能
(v1は1 > 2 > 4 > 8のように2倍(2のべき乗)のキャパシティで増加し最大256 ACU)
- スケールアップと異なりスケールダウンは、ある程度を時間をかける(ただしv1と比べて最大15倍高速(<1分))

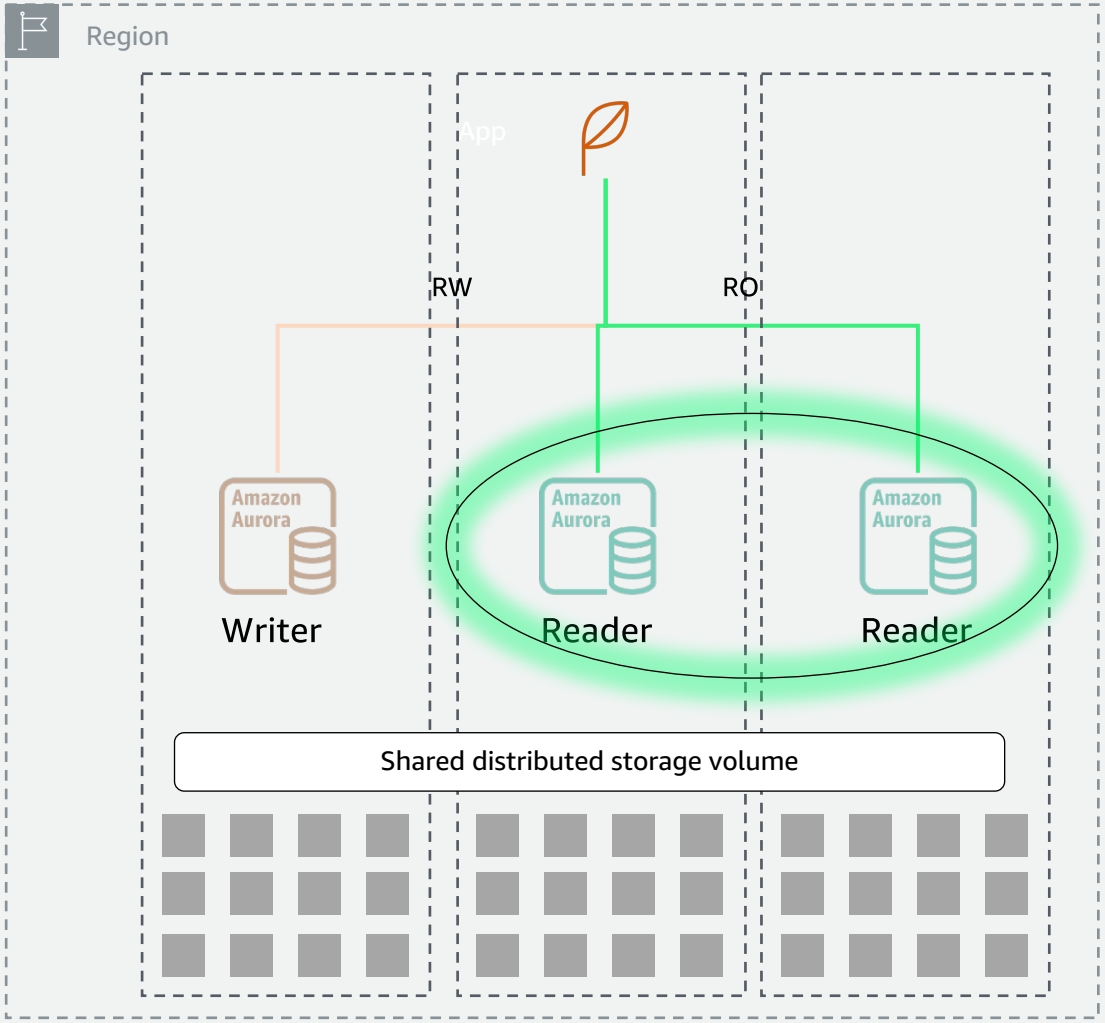


Aurora Serverless v2のスケーリングの特徴

READERによるスケール

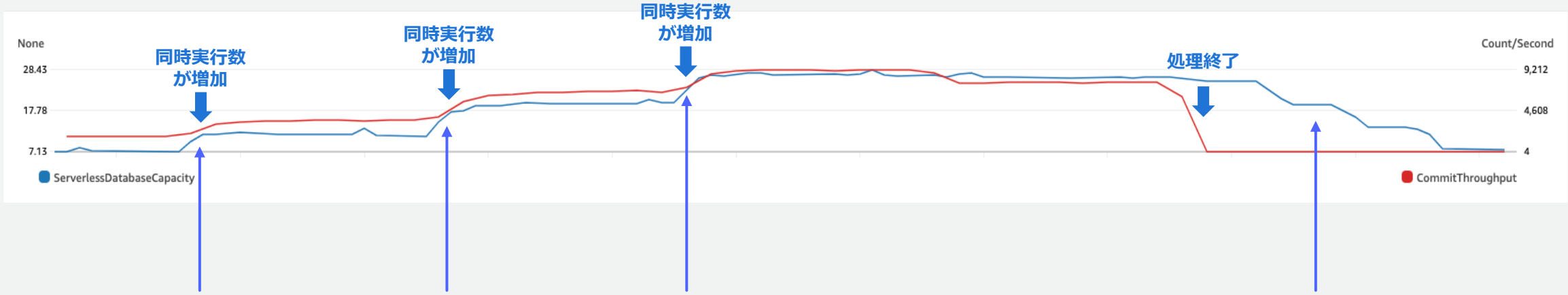
- ReaderはProvisionedと同様に
  最大15インスタンスまで設定可能
- 各Readerは最大128 ACUまで
  自動でスケールアップ可能

  Provisioned Clusterで提供されているReaderの
  Auto Scaling機能(負荷状況に
  応じた Readerの自動追加)はAurora Serverless
  v2ではサポートされていません



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)

Aurora Serverless v2のMixed Configuration

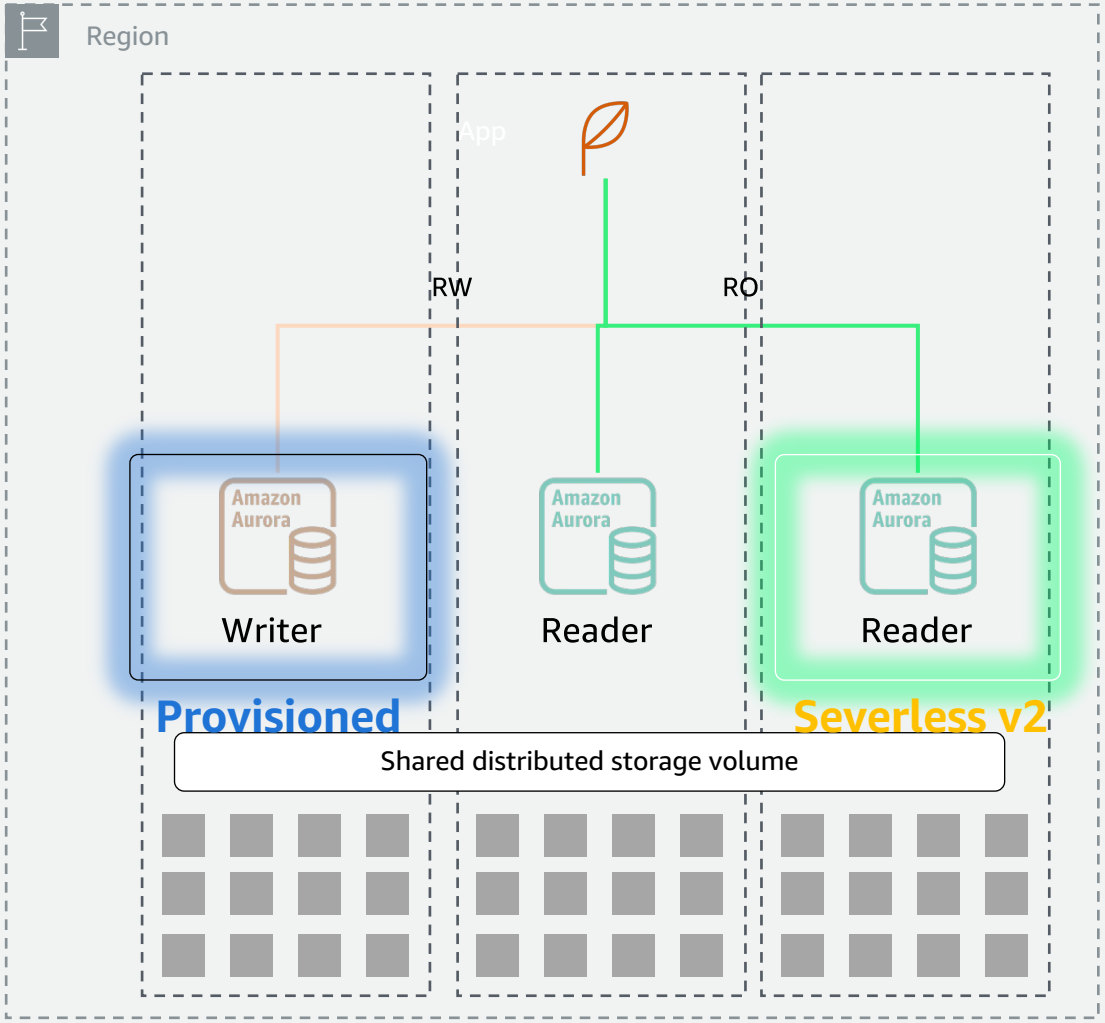
PROVISIONEDとSERVERLESS V2の混在環境

- 既存のProvisionedクラスター内のインスタンスは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

  * この場合、ReaderはWriterと一緒にスケールできません



Aurora Serverless v1とv2の簡単な比較

	Aurora Serverless v1	Aurora Serverless v2
スケールアップレイテンシー	5～50秒	数十万件/秒のトランザクションを処理できるスケーラビリティ
スケールダウンレイテンシー	最大15分	最大15倍高速(< 1分)
開始時の最小キャパシティー	1 ACU (Aurora MySQL) 2 ACU (Aurora PostgreSQL)	0.5 ACU
キャパシティ増加の粒度	スケールアップ毎にキャパシティは2倍	最小で0.5 ACUの細かい粒度
リードレプリカ	-	最大15インスタンス
Multi-AZとSLA	-	リードレプリカを別AZに配置可能
Global Database	-	DR用にGlobal Databaseを配置可能
料金(東京)	$ 0.1 ACU-Hour	$ 0.2 ACU-Hour

Aurora Serverless v2のユースケース

- Aurora Serverless v1のユースケース(可変、予測不可能なワークロード)に加えて
- 企業のデータベースフリート管理

  システム要件が変動する中、性能、可用性を確保し、予算内に収まるように、各データベースのキャパシティを継続的に監視・調整することは困難な作業です。Aurora Serverless v2では、実際の需要に基づいてデータベースのキャパシティが自動的に調整され、数百、数千のデータベースを手動で管理する必要がなくなります。
  また、Global Database、Multi-AZなどの機能により、高可用性と迅速なリカバリが保証されます。

- Software-as-a-Serviceアプリケーション

  SaaS(Software-as-a-Service)ベンダーは利用率とコスト効率を向上させるために、異なる顧客をサポートする数百から数千のデータベースを1つのAuroraで運用しています。しかし、共有するリソースの競合を回避するために各データベースを個別に管理する必要があります。Aurora Serverless v2を顧客ごとに用意することで、実ワークロードに応じてデータベースのキャパシティとコストを自動調整し複雑なリソース管理から解放します。

- 複数のサーバーに分割され、スケールアウトされたデータベース

  ワークロードのパフォーマンス要件が高いお客様は、より高いスループットを実現するために複数のRead Replicaに処理を分散します。しかし、Read Replicaの数やサイズの調整が適切でない場合、無駄なコストの発生や適切なパフォーマンスが得られないことが起こります。Aurora Serverless v2では、Read Replicaのスケールを自動調整しパフォーマンスとコストを最適化することが可能です。

Thank you!