

AWS RE:INVENT

re:Cap





ONLINE WEBINAR

AWS re:Invent Recap - ソリューション編

セッションは 14:00 から開始となります。
今しばらくお待ちください。

推奨ブラウザ

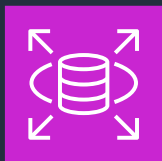
- Google Chrome (most recent 2 versions)
- Mozilla Firefox (most recent 2 versions)

お知らせとお願い

- 本日の資料はセミナー終了後にご案内するアンケートにダウンロードリンクがございます。
- 質問はセッション中、休み時間中、随時受け付けております。
GoToWebinar のチャット機能ではなく質問機能でご質問ください。

自己紹介

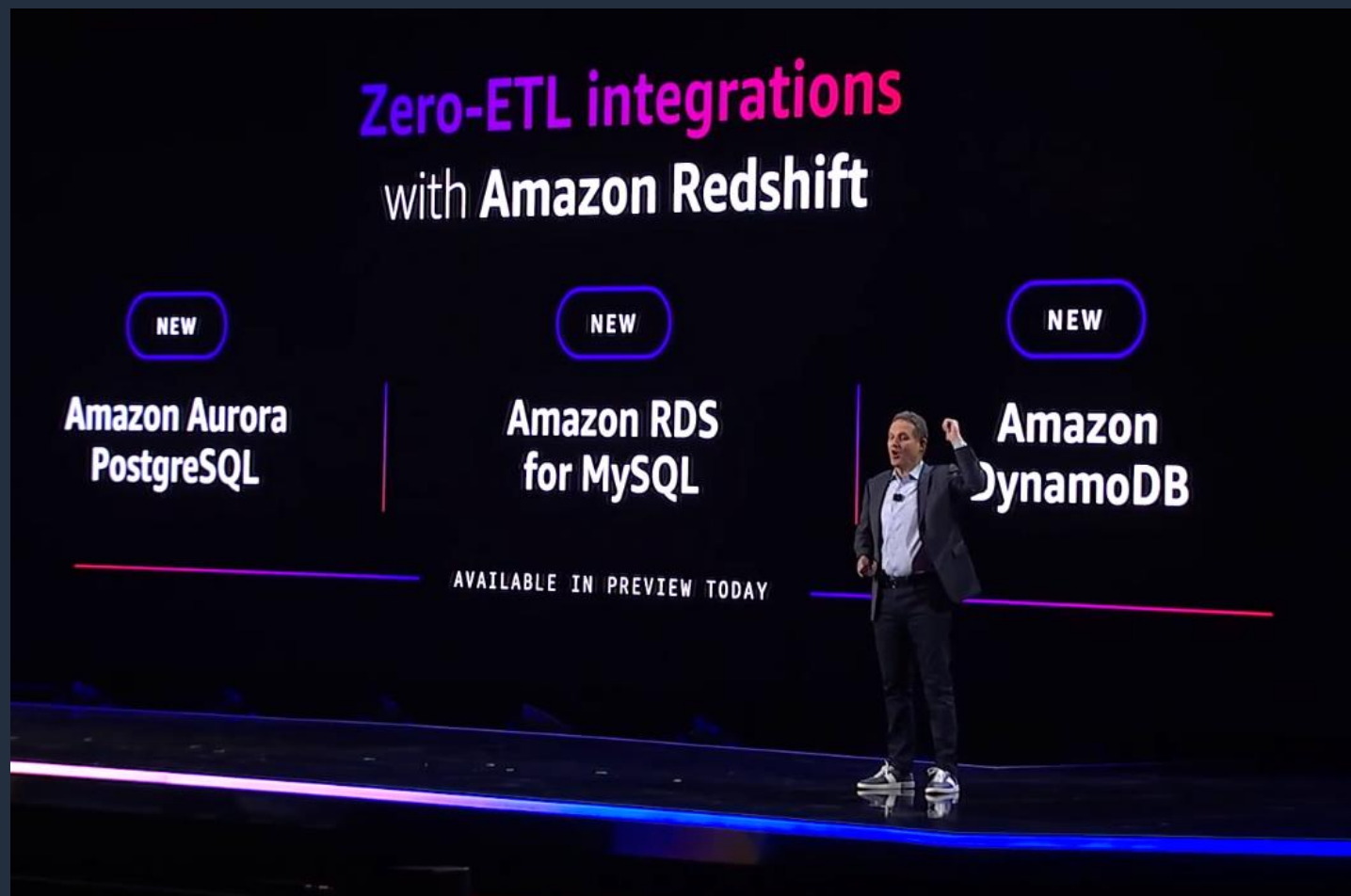
- 鈴木 大樹 (Suzuki Daiki)
 - ソリューションアーキテクト
- クラウド上でのデータ活用のお手伝いを
様々なお客様に行いたいと考えています。
- 好きな AWS サービス
 - Amazon Aurora, Amazon RDS



Amazon RDS/Amazon Aurora 新機能まとめ Part2

- Amazon RDS for MySQL が Amazon Redshift への zero-ETL integration をサポート (Public Preview)
- Amazon Aurora PostgreSQL が Amazon Redshift への zero-ETL integration をサポート (Public Preview)
- Amazon RDS for Oracle が Oracle Multitenant をサポート
- Amazon RDS Custom for SQL Server が changing the server-level collation をサポート
- RDS Custom for SQL Server が SQL Server Developer Edition をサポート
- Amazon RDS for PostgreSQL で pgactive 拡張機能サポート
- Amazon Aurora PostgreSQL Optimized Reads をサポート
- Amazon Aurora Global Database for PostgreSQL が Write forwarding をサポート
- Amazon Relational Database Service が Dedicated Log Volume をサポート
- Amazon Aurora と Amazon RDS が MySQL と PostgreSQL データベースの延長サポートを提供

• Amazon RDS for MySQL が Amazon Redshift への zero-ETL integration をサポート (Public Preview)



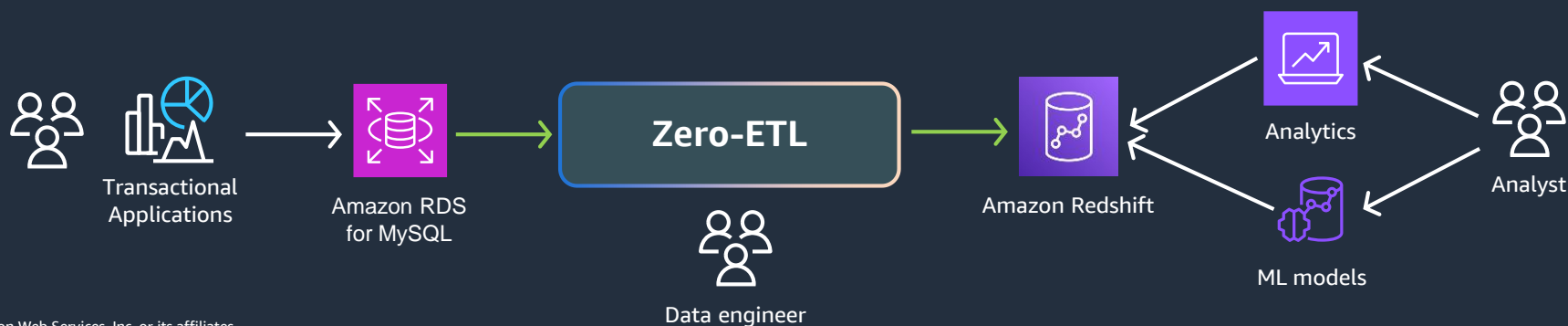
What's New

1. “Amazon Relational Database Service (Amazon RDS) for MySQLのzero-ETL integration with Amazon Redshiftにより、Amazon RDS for MySQL からトランザクション データにアクセスして、Amazon Redshiftのペタバイト規模のデータに対して分析と機械学習 (ML) を実行できます”
2. Zero ETL integration を使用すると、抽出、変換、ロード (ETL) 操作を実行するために複雑なデータ パイプラインを構築および維持する必要がありません”

[AWS announces Amazon RDS for MySQL zero-ETL integration with Amazon Redshift \(Public Preview\)](#)

Amazon RDS for MySQL Zero ETL Integration

- Amazon Redshift からデータベースのデータにアクセスして分析できる
- 組織内のデータサイロを解消でき、RDS MySQLインスタンスからのデータの分析が容易になる
- Integration したデータを Amazon AuroraやAmazon DynamoDBなどの他の Zero ETL Integration からのデータと同じ Amazon Redshift クラスタに Integration できる



導入メリット

- 複数のデータソースから総合的なインサイトを引き出す
- ETL 操作を実行する複雑なデータパイプラインを構築して管理する必要がなくなる
- Amazon Redshift の分析機能と ML 機能を活用して、トランザクションデータやその他のデータからインサイトを引き出し、重要で時間的制約のあるイベントに効果的に対応

※ MySQLをOLTP処理に利用されているお客様においては、MySQLが苦手とする大規模なデータ集計やレポート処理に活用頂く事で、必要なレポートを短時間で集計し、これまで以上に迅速な意思決定をする事が可能になる

Amazon RDS for MySQL Zero ETL Integration

RDS for MySQL から Redshift へ簡単な設定でシームレスにオブジェクトとデータ同期が可能

```
mysql> desc t_note;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int           | NO   | PRI | NULL             | auto_increment |
| note           | varchar(255)  | NO   |     |                  |                |
| dml_user       | varchar(255)  | NO   |     | user()           | DEFAULT_GENERATED |
| dml_current_user | varchar(255)  | NO   |     | current_user()   | DEFAULT_GENERATED |
| updated_time   | datetime      | YES  | MUL | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into t_note(note) values("Amazon RDS zero-ETL integrations with Amazon Redshift");
Query OK, 1 row affected (0.01 sec)

mysql> insert into t_note(note) values("Amazon RDS ゼロ ETL インテグレーション");
Query OK, 1 row affected (0.00 sec)
```

The screenshot shows the Amazon RDS console interface. At the top, there are controls for 'Run', 'Limit 100', 'Explain', 'Isolated session', and the instance details 'Serverless: rd...' and 'my_database'. The SQL query being executed is:

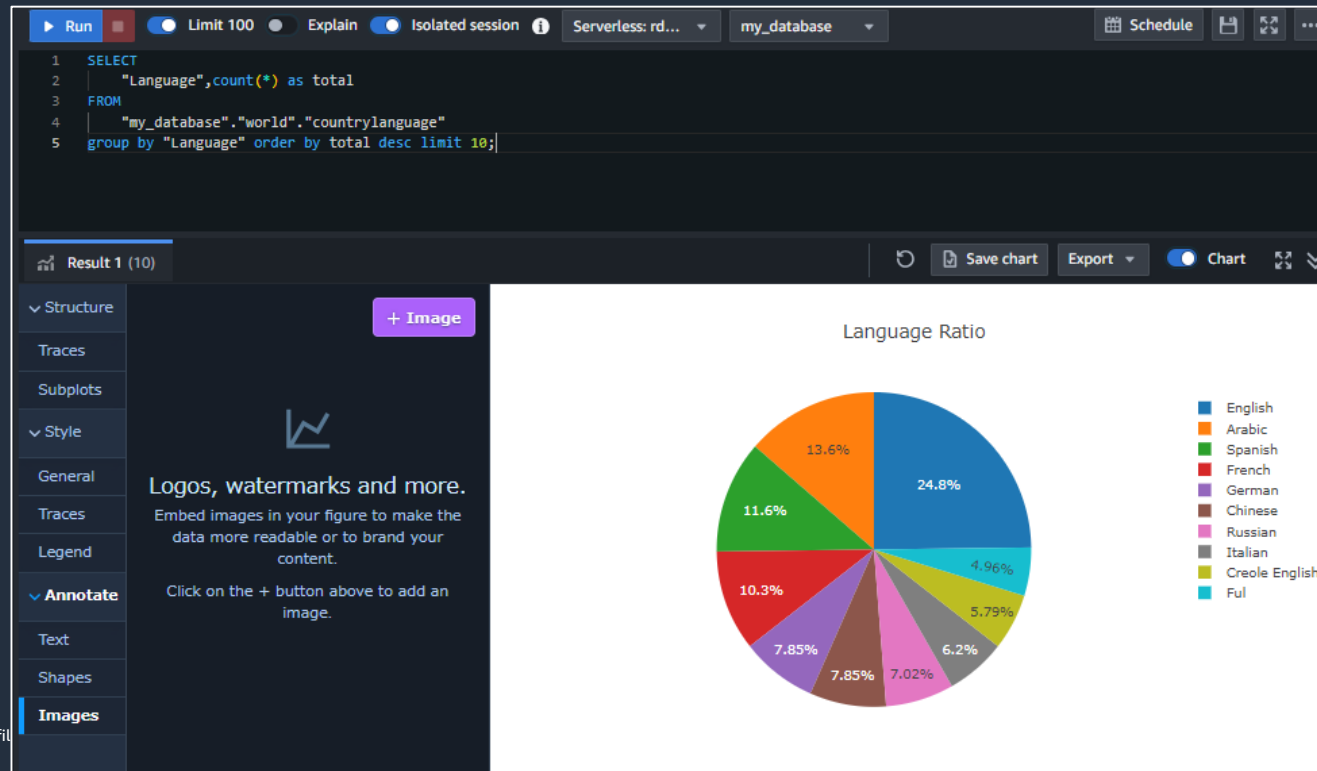
```
1 SELECT
2   *, current_time
3 FROM
4   "my_database"."ZETL"."t_note"
5 order by id;
```

Below the query, the results are displayed in a table with 7 columns: id, note, dml_user, dml_current_user, updated_time, and timetz. There are two rows of data.

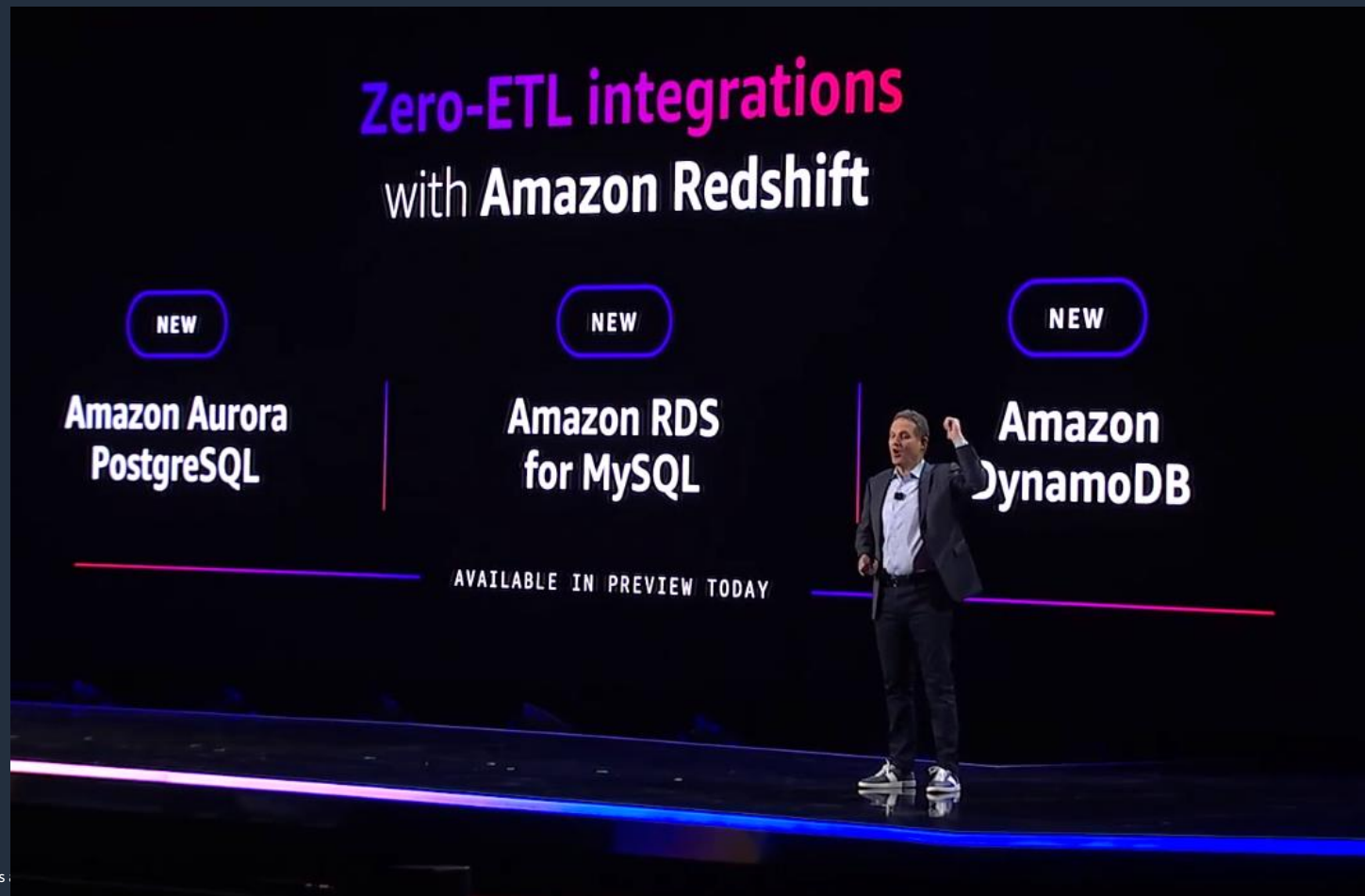
id	note	dml_user	dml_current_user	updated_time	timetz
1	Amazon RDS zero-ETL integrations with ...	zeroetl@172.3...	zeroetl@%	2023-12-01 04:50:10	04:51:57.294609+00
2	Amazon RDS ゼロ ETL インテグレーション	zeroetl@172.3...	zeroetl@%	2023-12-01 04:50:30	04:51:57.294609+00

お客様の意思決定をサポート

RedshiftにはMySQLには実装されていない、Materialized View等が実装されているので、短時間で大量なデータを解析出来る
可視化と情報共有をサポートするチャート等の機能や、ビジネスインサイトを深堀する為の機械学習との連携が強化されているため、これまで以上に迅速かつ効率良く意思決定をサポート

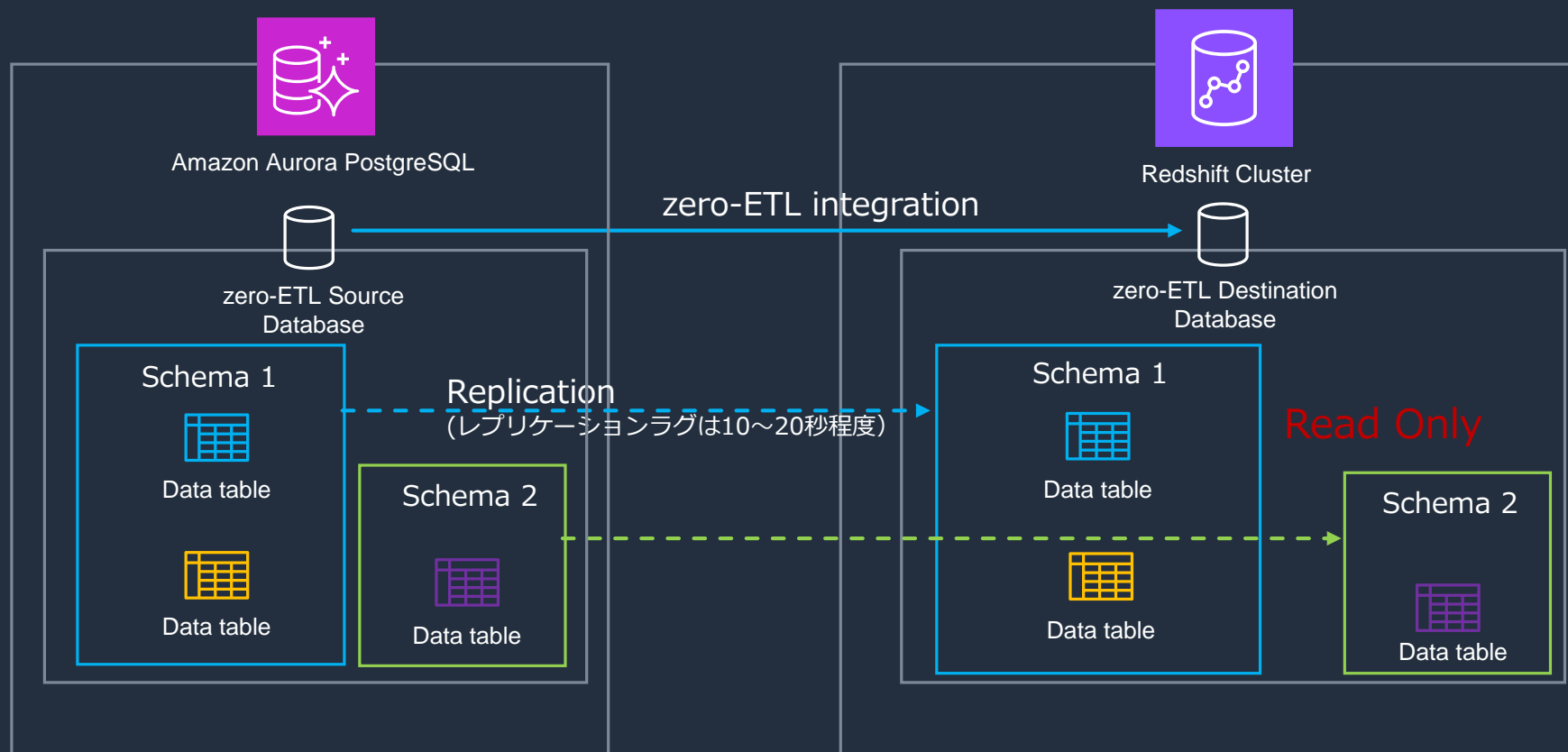


Amazon Aurora PostgreSQL が Amazon Redshift への zero-ETL integration をサポート (Public Preview)



zero-ETL による Aurora PostgreSQL から Redshift へのデータ連携

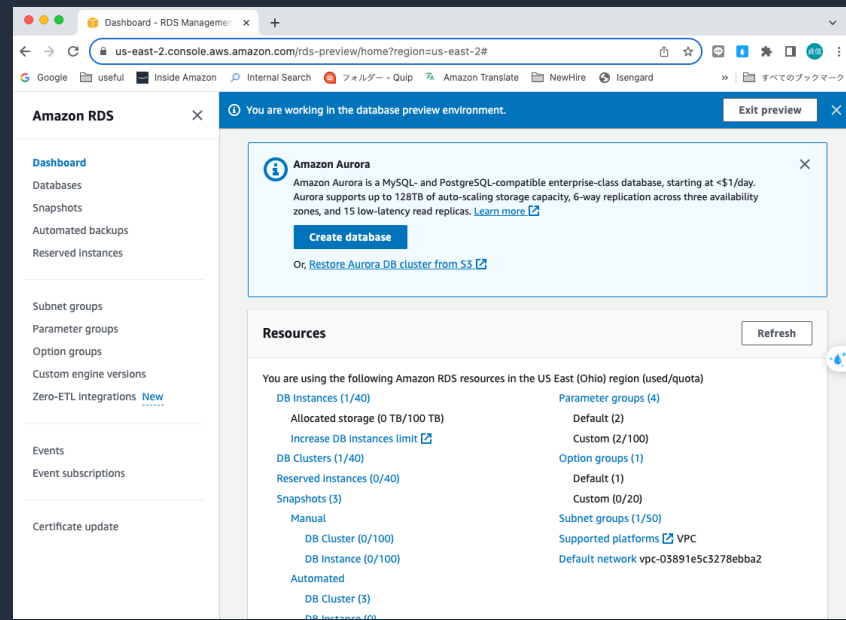
Aurora PostgreSQL の DB は Redshift 上のデータベース単位で連携される



- zero-ETL で連携したテーブルは **Read Only** となる
- 連携したテーブルから別 DB にマテリアライズドビューの作成も可能

Amazon Aurora PostgreSQL zero-ETL integration 作成手順

1. カスタム DB Cluster parameter group の作成
2. ソースDBクラスターの作成時に
カスタムDB Cluster Parameter Groupを指定する
3. ターゲットAmazon Redshift の作成時に
専用のAmazon Redshift feature in preview 環境
から作成する
4. 作成したRedshiftに承認されたプリンシパル、
Integration Source を設定する



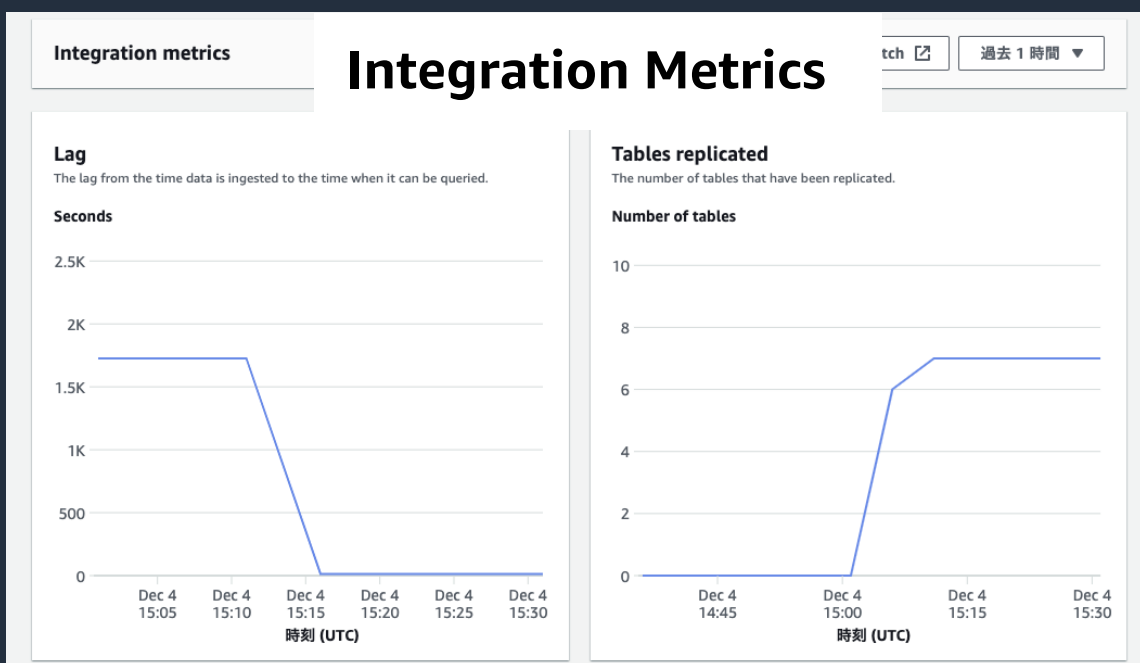
※zero-ETL integrationを RDS のコンソールから作成時、
Primary key の無いテーブルは連携されないため注意が必要

Viewing and monitoring

Aurora の zero-ETL と Amazon Redshift の integration は、AWS マネジメントコンソール、AWS CLI、RDS API を使用して表示可能

Lag: データが取り込まれてから照会できるようになるまでのタイムラグ

Tables replicated: レプリケートされたテーブルの数



The Table Statistics dashboard shows a list of 8 tables. The 'Status' column indicates whether each table is 'Synced' (green checkmark) or 'Failed' (red X). The 'Last updated' column shows the timestamp for each table.

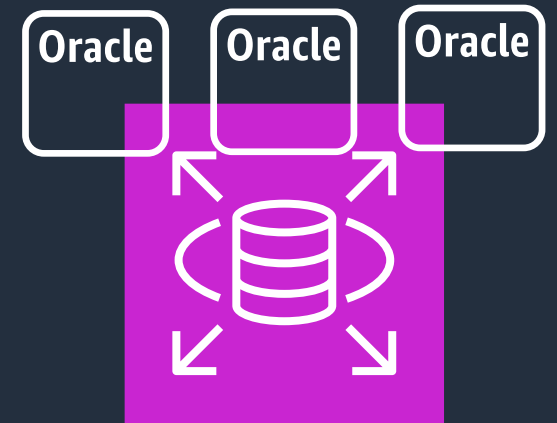
Schema name	Table name	Status	Database	Last updated
myschema		Failed	zeroetl	December 04, 2023, 06:13 (UTC+09:00)
public	pgbench_branches	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)
public	books3_table	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)
public	pgbench_accounts	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)
myschema	kenko_record1	Synced	zeroetl	December 04, 2023, 06:14 (UTC+09:00)
public	books_table	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)
public	pgbench_history	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)
public	pgbench_tellers	Synced	zeroetl	December 04, 2023, 06:05 (UTC+09:00)

Aurora PostgreSQL - preview 時点の制限事項（抜粋）

- Aurora PostgreSQL との zero-ETL integrationのサポート対象外
 - Aurora Serverless v2
 - ソース DB クラスタ上のサブトランザクション
 - ソース DB クラスタ内のスキーマまたはデータベースの名前変更
 - Oversize-Attribute Storage Technique (TOAST) を必要とする大きなフィールド値

Amazon RDS for Oracle が Oracle Multitenant をサポート

- RDS for Oracleで複数の Pluggable Database(PDB)構成をサポートする Oracle マルチテナント構成が可能になった
- Container Database(CDB)上に複数のPluggable Database(PDB)が構成される
- 本構成を利用することにより 1RDS インスタンスの中で、複数の DB を構成することが可能
- 3 PDB までは、Oracle License Options の考慮なしで利用可能



Amazon RDS for Oracle

Amazon RDS for Oracle の Multitenant 作成方法

- RDS Oracleインスタンス作成時に、“マルチテナントアーキテクチャ”と“マルチテナント設定”を選択
- 1つ目のテナントデータベース (Pluggable Database – PDB)の設定を入力

データベース管理タイプ 情報

Amazon RDS

RDS は、自動パッチ適用を含め、データベースを完全に管理します。環境をカスタマイズする必要がない場合は、このオプションを選択します。

Amazon RDS Custom

RDS はデータベースを管理し、OS への特権アクセスを提供します。データベース、OS、インフラストラクチャをカスタマイズする場合は、このオプションを使用します。

アーキテクチャ設定 情報

マルチテナントアーキテクチャ

コンテナデータベース (CDB) には、1 つ以上のテナントデータベース (PDB) が含まれます。非 CDB はマルチテナントアーキテクチャを使用しません。非 CDB を CDB に変換することはできませんが、CDB を非 CDB に変換することはできません。

アーキテクチャ設定

シングルテナント設定

CDB には初期 PDB が 1 つ含まれています。後で PDB を追加することはできません。シングルテナント設定をマルチテナント設定に変換すると、PDB を追加できます。

マルチテナント設定

CDB には初期 PDB が 1 つ含まれています。後で PDB を追加することができます。この設定をシングルテナント設定に変換することはできません。

テナントデータベース設定 情報

テナントデータベース名

これは最初のプラグイン可能データベース (PDB) の名前です。

pdb1

この名前は CDB 名とは異なる必要があります。

テナントデータベースマスターユーザー名

これは PDB のログイン ID です。

admin

ユーザー名には 30 文字以内の英数字を使用できます。1 文字目は英文字である必要があります。

パスワードの自動生成

Amazon RDS がパスワードを生成するか、お客様がご自身でパスワードを指定することができます。

テナントデータベースのマスターパスワード

制約事項: 表示可能な ASCII 文字で 8 文字以上で入力してください次の文字を含めることはできません: / (スラッシュ)、' (単一引用符)、* (二重引用符)、および @ (アットマーク)。

パスワードを確認

テナントデータベース文字セット

This value can be different from the CDB character set.

AL32UTF8

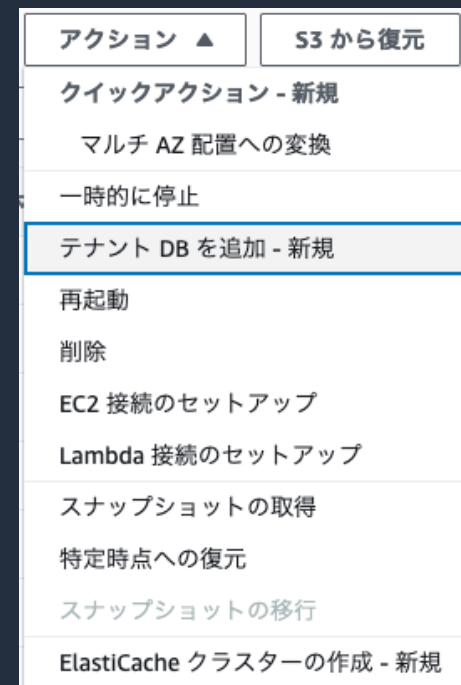


Amazon RDS for Oracle の Multitenant 作成方法

- 1PDB構成マルチテナント・データベースの作成が完了

<input type="radio"/>	<input type="checkbox"/>	mt-test	✔ 利用可能	インスタンス	Oracle Enterprise Edition (CDB)
<input type="radio"/>	<input type="checkbox"/>	PDB1	✔ 利用可能	テナントデータベース	-

- 「アクション」から「テナントDBを追加・新規」を選択し、2つ目以降のPDBを追加

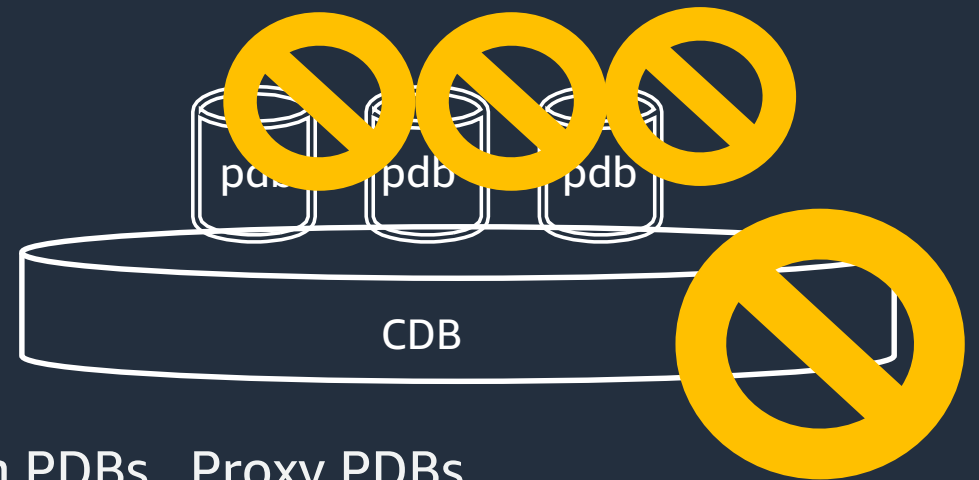


- 複数PDB構成のデータベースが構築完了

<input checked="" type="radio"/>	<input type="checkbox"/>	mt-test	✔ 利用可能	インスタンス	Oracle Enterprise Edition (CDB)
<input type="radio"/>	<input type="checkbox"/>	PDB1	✔ 利用可能	テナントデータベース	-
<input type="radio"/>	<input type="checkbox"/>	PDB2	✔ 利用可能	テナントデータベース	-

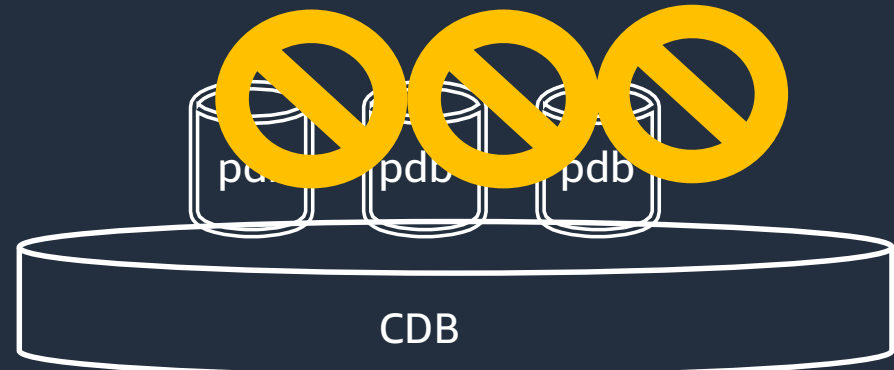
Amazon RDS for Oracle の Oracle Multitenantの制限事項

- データベースのユーザーアカウントは各PDBごとに管理される
- 以下の機能はサポートされない
 - Oracle Data Guard
 - Oracle Label Security
 - Oracle Enterprise Manager(OEM)
 - OEM Agent
 - Database Activity Streams
 - Unplugging and plugging in PDBs, Application PDBs, Proxy PDBs



Amazon RDS for Oracle の Oracle Multitenantの制限事項

- 以下の機能はCDBレベルでのみサポートされ、PDBレベルではサポートされない
 - Backup and recovery
 - Database upgrades
 - Maintenance actions
 - Performance Insights
 - Options groups(オプションはCDB上に構成されている全てのPDBに導入される)
 - Parameter groups(全てのパラメータはCDBインスタンスに関連つけられて構成される)



より詳細な制限事項はこちらのドキュメントを参照ください

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Oracle.Concepts.CDBs.html>

RDS for Oracle の Oracle Multitenant における注意事項

- Amazon RDS for Oracle では、最大で30PDB構成までが構成可能
- 一方で、4つ以上のPDB構成については、お客様とOracle社との間のライセンス契約に注意が必要
- オンプレミスの構成では、Oracle Enterprise Edition と Oracle Multitenant Options を購入することで、4つ以上のPDB構成がサポートされている
- Oracle 社のライセンスガイドには、「Non-Oracle Cloud Environments: Use is limited to three pluggable databases per container database.」という記載があり、本記載の制約を受けるか否かはお客様とOracle 社の個別の契約に基づく

Amazon RDS Custom for SQL Server サーバーレベル照合順序の変更をサポート

- サーバーレベル照合順序の変更がサポートされ、日本語の照合順序も利用可能になった
- Japanese_* で 300 超の照合順序が利用可能

▼ 追加設定

データベースオプション, バックアップ をオンにしました, バックトラック をオフにしました, メンテナンス, 削除保護 をオンにしました.

データベースの選択肢

タイムゾーン

Tokyo Standard Time ▼

日本語 [情報](#)

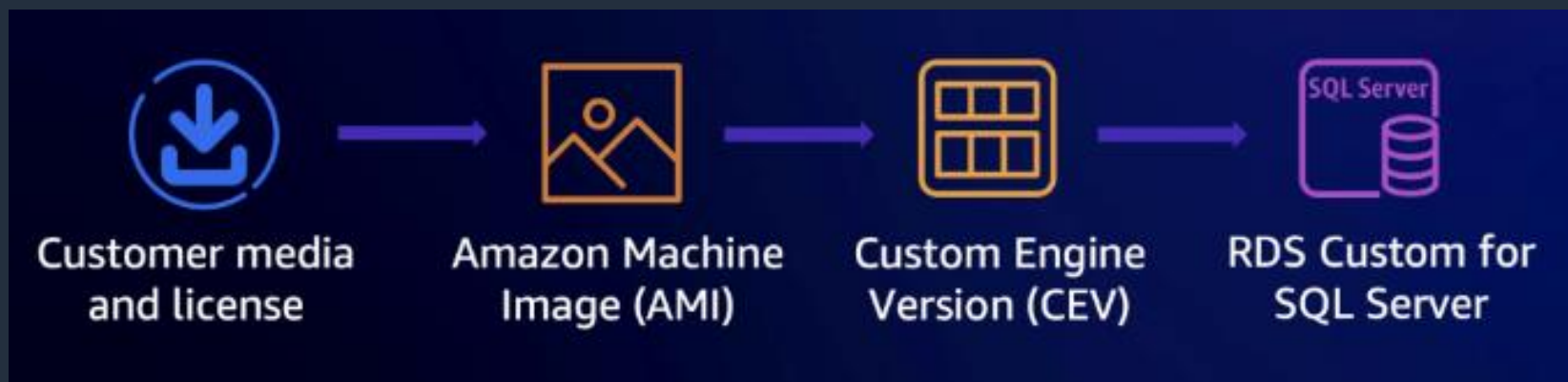
Japanese_XJIS_140_CI_AI_KS_VSS

Amazon RDS Custom for SQL Server サーバーレベル照合順序の変更をサポート

- RDS Custom インスタンス作成時に設定
 - 作成後にサーバーレベルの照合順序を変更する事は不可
 - CEV (Custom Engine Version) 利用有無にかかわらず、RDS Custom インスタンス作成時に設定
- 利用できるサーバーレベル照合順序リスト
 - <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/custom-reqs-limits-MS.html#custom-reqs-limits-MS.collation>

Amazon RDS Custom for SQL Server Developer エディションをサポート

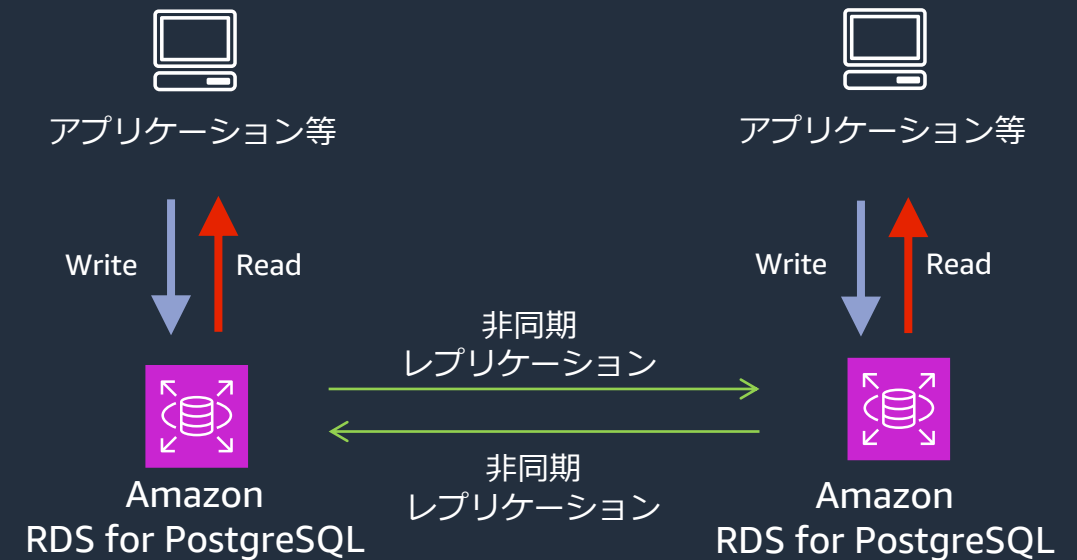
- お客様が持ち込みのインストールメディアを使うことで、RDS Custom で Developer エディションを利用でき、開発やテスト用途での費用を抑えられる



Developer エディションの利用の際は持ち込みのインストールメディアを使用して EC2 上に SQL Server を構築し、AMI を取得して CEV を作成

Amazon RDS for PostgreSQL で pgactive 拡張機能サポート

- pgactive 拡張機能が、RDS for PostgreSQL でサポート
- 複数のPostgreSQL データベース間で、active-active のレプリケーションを実現
- 全てのノードで参照、更新が可能
 - 書き込みの負荷分散はアンチパターン
- OSS のBDR から派生した機能
 - BDR (Bi-Directional Replication) :2ndQuadrant 社 (現EDB社) を中心に開発された、オープンソースのマルチマスタ・レプリケーションシステム
 - https://wiki.postgresql.org/wiki/BDR_Project
- RDS for PostgreSQL 15.4-R2 以降で使用可能
- Aurora PostgreSQL は現時点で非サポート



pgactive のユースケース

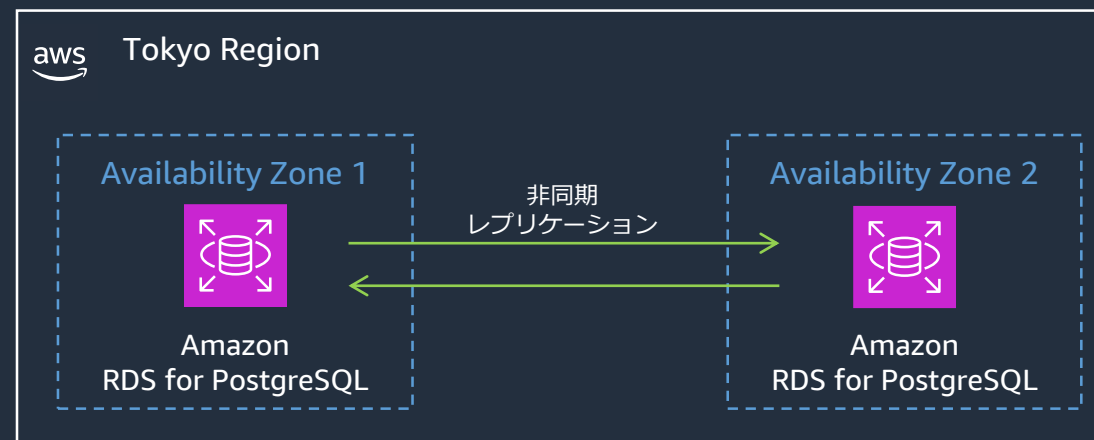
• 同一リージョン内でのマルチマスター構成

• 高可用性構成

- 書き込み失敗時にもう片方に書き込みをリトライ
- フェイルオーバー時間の短縮

※非同期レプリケーションの為、障害時のデータ損失などは考慮が必要

※書き込みの負荷分散はアンチパターン



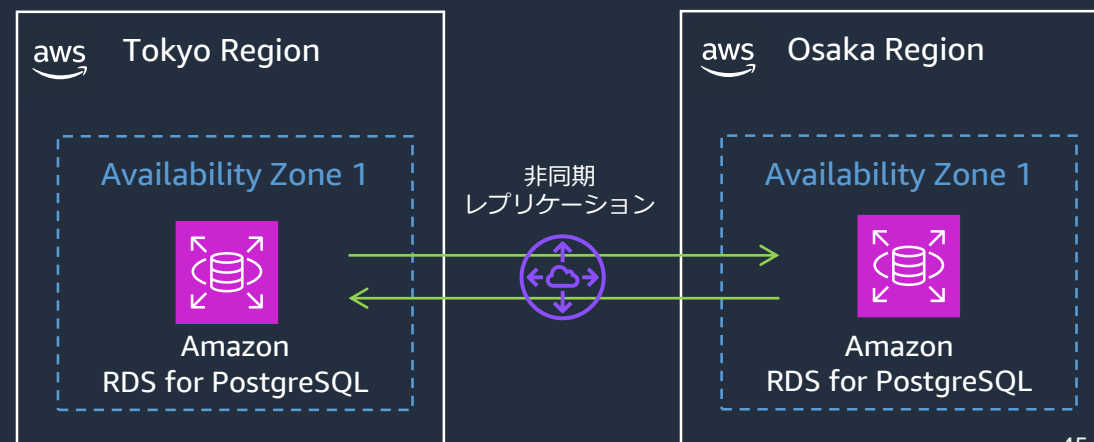
• リージョン間でのマルチマスター構成

• グローバルなアプリケーション

- 異なるリージョンで少量の書き込みが発生するようなケース
- 同一テーブルや行への更新が発生しないようなケース

• DR 環境

- セカンダリで書き込みが不要であれば、クロスリージョンリードレプリカをお勧め



pgactiveの注意点

データ連携はデータベース単位

テーブル単位の指定不可

初回構築時は全オブジェクトを連携

初回以降の DDL は連携されない

コミットはインスタンス単位

・最後に COMMIT された値が反映

・競合した場合、pgactive.pgactive_conflict_history で情報を確認可能

Database

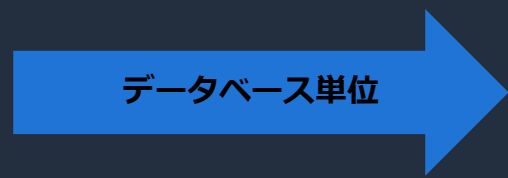
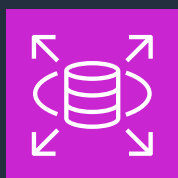
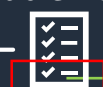


Table List



テーブル単位の指定不可



USERS テーブル

ID	NAME
1	SMITH
2	ALLEN

Node 1

BEGIN TRANSATIONS;

UPDATE USERS
SET NAME = 'MARTIN'
WHERE ID = 1;

最後のCOMMIT が
反映

COMMIT;

Node 2

BEGIN TRANSATIONS;

UPDATE USERS
SET NAME = 'BLAKE'
WHERE ID = 1;

ロック待ちなし

COMMIT;

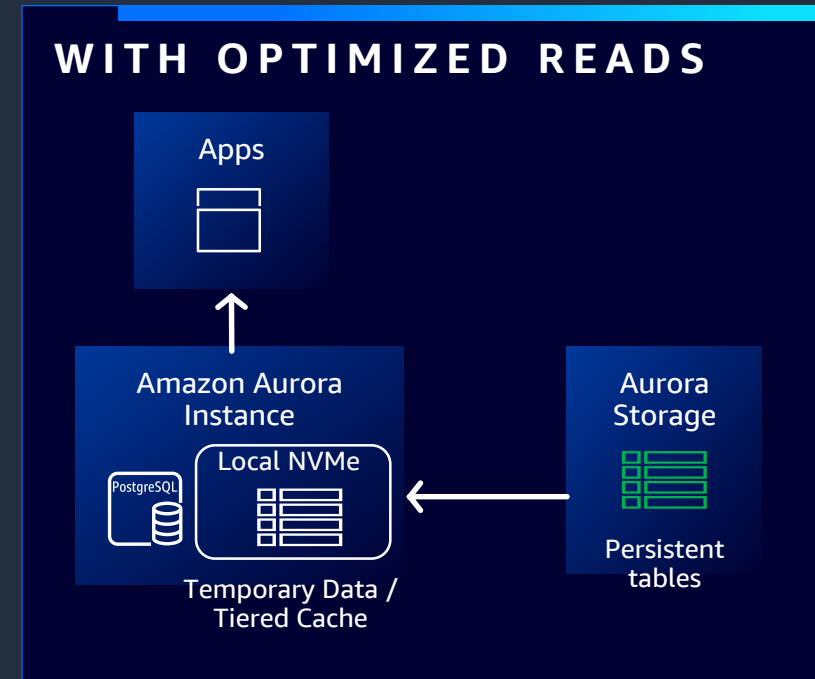
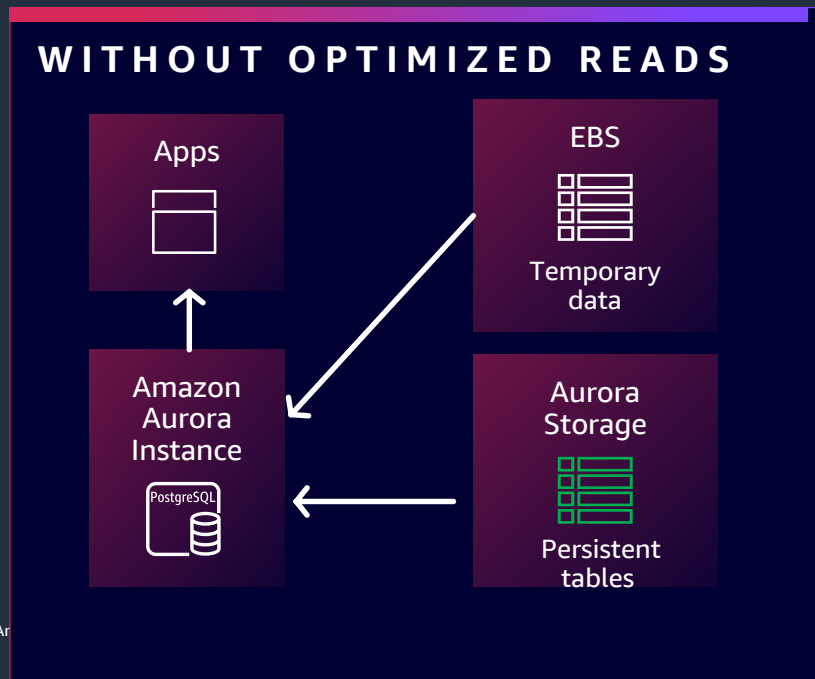
ID	NAME
1	SMITH
2	MARTIN

制限事項

- すべてのテーブルに**プライマリキーが必要**です。プライマリキーがないと、更新と削除は許可されません。また、主キー値を更新しないでください。
- シーケンスにはギャップがあり、更新順に番号が並ばない場合があります。シーケンスは複製されません。
 - グローバルシーケンス (pgactive.pgactive_snowflake_id_nextval) でユニークな値を作成することは可能
- DDL とラージオブジェクトは複製されません。
 - テーブル作成時のDDL も複製されません。この為、テーブルを追加した場合、ターゲット側にテーブルを先に作成しておく必要があります。
- セカンダリユニークインデックスはデータ相違の原因になることがあります。
- 照合順序はグループ内のすべてのノードで同一である必要があります。
- ノード間の負荷分散はアンチパターンです。
- **大量のトランザクションはレプリケーションラグの原因**になることがあります。

Amazon Aurora PostgreSQL が Optimized Reads をサポート

- ローカル NVMe を利用して、よりレイテンシーの小さい I/O を実現
 - Temp Object: ローカル NVMe 上に一時オブジェクトを置いてパフォーマンスを向上
 - Tiered Cache: ローカル NVMe のキャッシュにアクセスし、レイテンシを改善

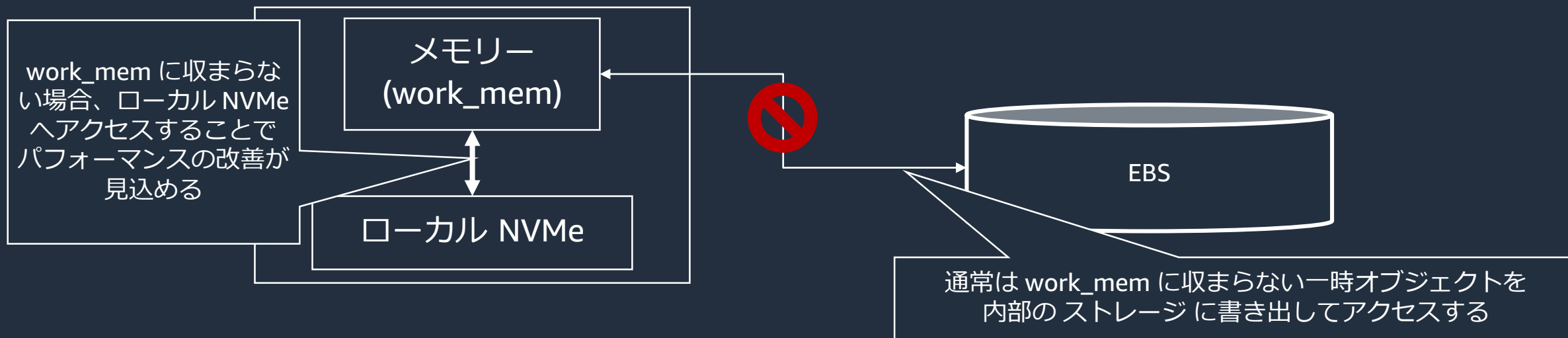


各種 DB の Optimized Reads サポート状況

DB エンジン	Optimized Reads		サポートバージョン
	Temp Object	Tiered Cache	
Aurora PostgreSQL	サポート	I/O 最適化ストレージのクラスタ構成で可能	14 系: 14.9 以上 15 系: 15.4 以上
Aurora MySQL	サポートされていない		
RDS for PostgreSQL	サポート	サポートされていない	13 系: 13.10 以上 14 系: 14.7 以上 15 系: 15.2 以上
RDS for MySQL	サポート	サポートされていない	8.0.28 以上

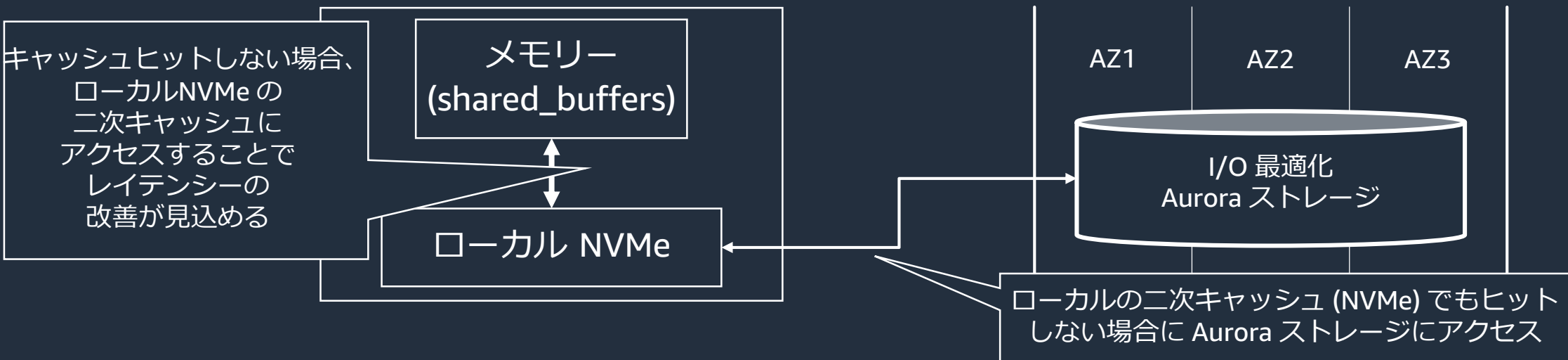
Aurora PostgreSQL Optimized Reads (Temp Object)

- work_mem に収まらないようなワークロードのチューニングに利用可能
- 読み取り最適化をサポートするインスタンスを利用することでクエリーの高速度化が期待できる



Aurora PostgreSQL Optimized Reads (Tiered Cache)

- I/O 最適化 Aurora ストレージ^(*1)を利用している場合、Tiered Cache が有効になる
 - キャッシュヒット率が低下するようなワークロードでも、より小さい I/O レイテンシー・大きな I/O 帯域で処理できるようになる^(*2)

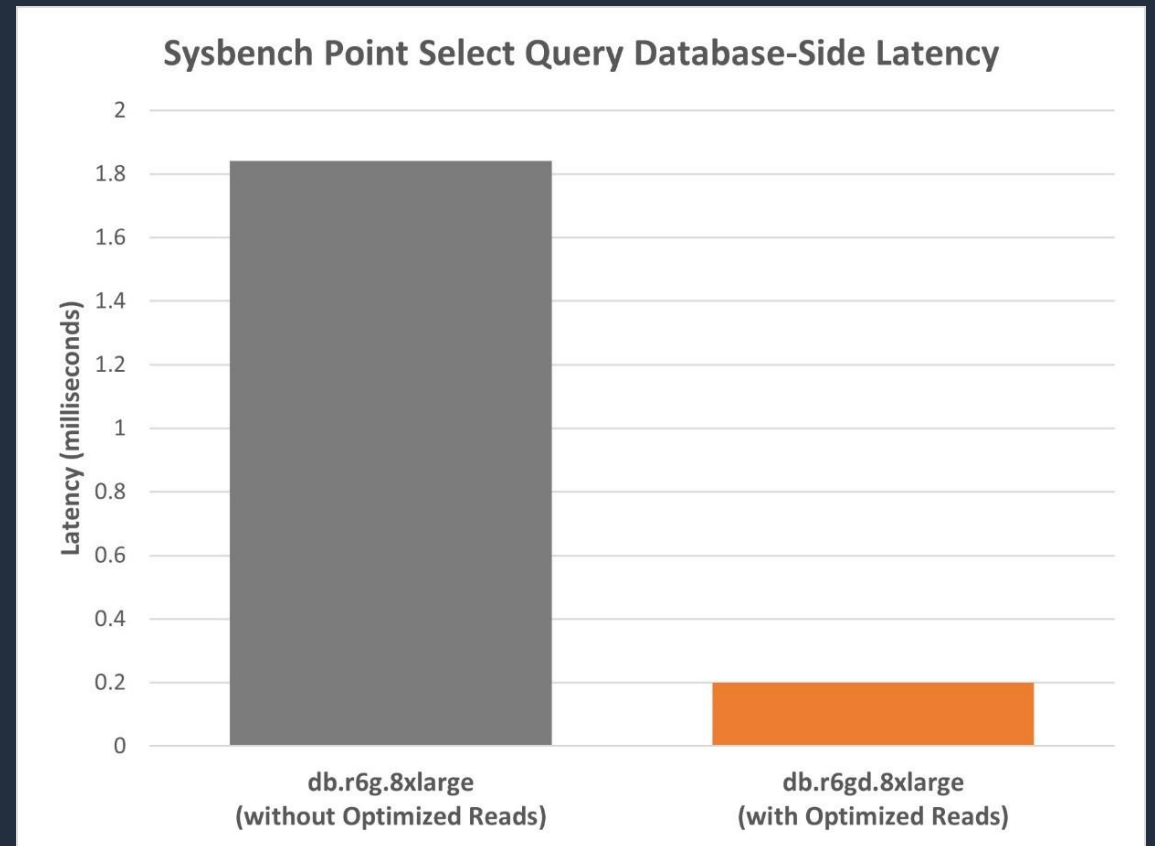
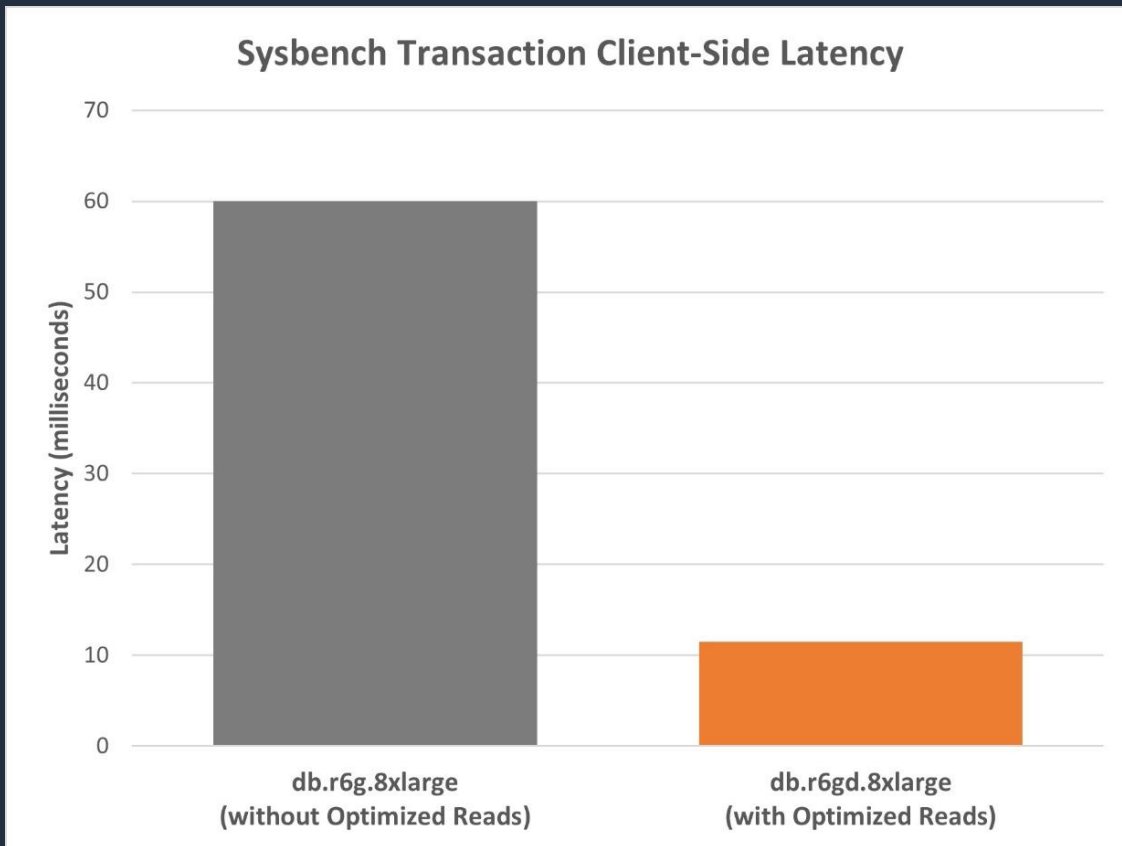


(*1) <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html#aurora-storage-type>

(*2) <https://aws.amazon.com/jp/blogs/database/new-amazon-aurora-optimized-reads-for-aurora-postgresql-with-up-to-8x-query-latency-improvement-for-i-o-intensive-applications/>

Optimized Reads の有無の比較(Tiered Cacheの利用)

- クライアントサイドも DB サイドも レイテンシが削減^(*1)



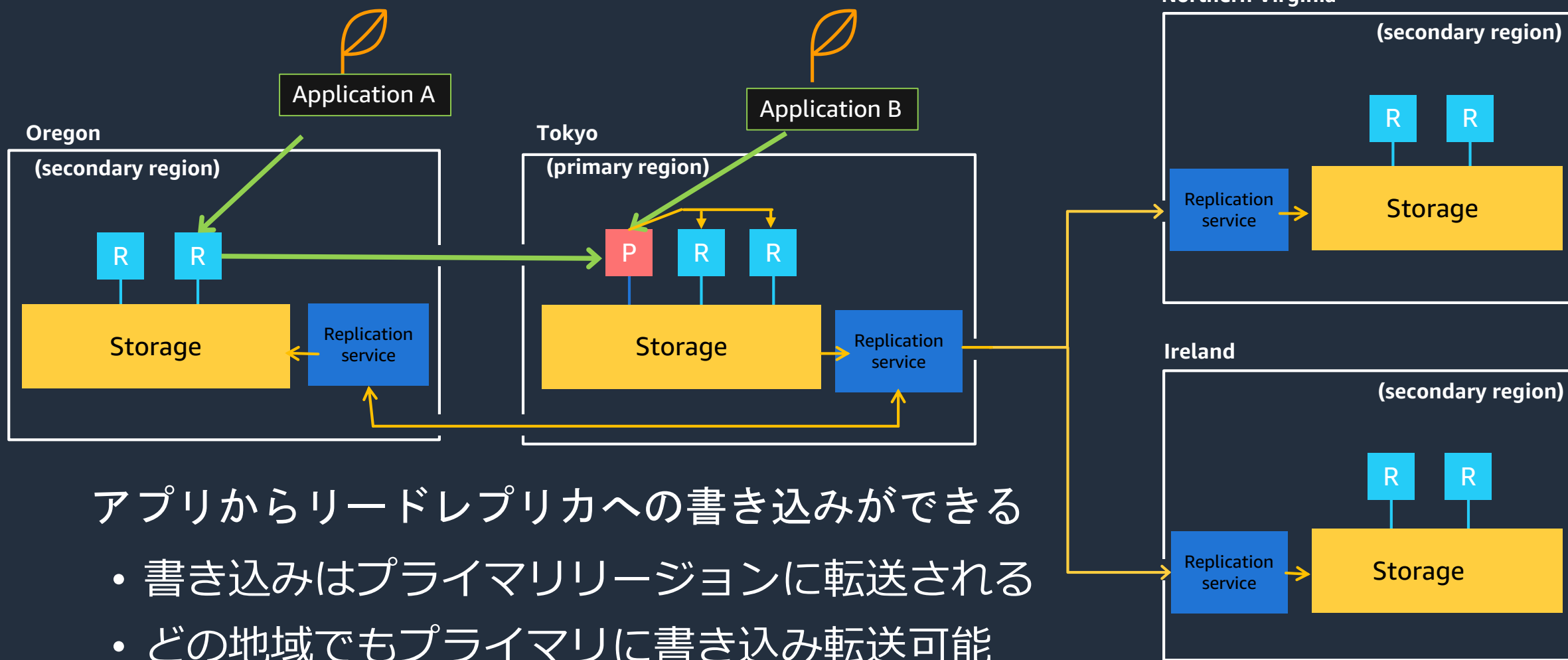
ユースケース

- 機械学習や生成 AI アプリケーション
 - pg_vector と Tierd Cache を併用したベクトル検索でクエリ処理件数が最大 9 倍に増加
 - pg_vector: PostgreSQL 向けのベクトル管理・検索を行う拡張モジュール
- クエリの実行を内部テーブルやファイルなどの一時オブジェクトに大きく依存するワークロード
 - 中間結果を格納するために大規模な一時テーブルを作成するアプリケーション
 - 複雑なテーブル共通式 (CTE)、派生テーブル、グループ化オペレーションを使用して分析クエリを実行するアプリケーション

Amazon Aurora Global Database for PostgreSQL が Write forwarding (書き込み転送) をサポート

- Amazon Aurora Global Database
 - Aurora を複数の AWS リージョンで運用し、リージョン全体停止からのディザスタリカバリや、グローバルに分散した DB のローカル読み取りが可能
- Write forwarding (書き込み転送)
 - セカンダリクラスターで受け取った書き込みオペレーションをプライマリクラスターに転送することができる
- Aurora MySQL は2020/6/19 にサポート済み

Aurora Global Database Write Forwarding



アプリからリードレプリカへの書き込みができる

- 書き込みはプライマリリージョンに転送される
- どの地域でもプライマリに書き込み転送可能

Write Forwarding の機能と制限

- プライマリクラスターにて最大転送同時セッション数とセッションタイムアウトを設定可能
- セカンダリクラスターの読み取り整合性のレベル



- サポート外^(*)

- DDL、ANALYZE、CLUSTER、COPY、Cursors
GRANT|REVOKE|REASSIGN OWNED|SECURITY LABEL、LOCK、
SAVEPOINT、SELECT INTO、SET CONSTRAINTS、TRUNCATE、VACUUM

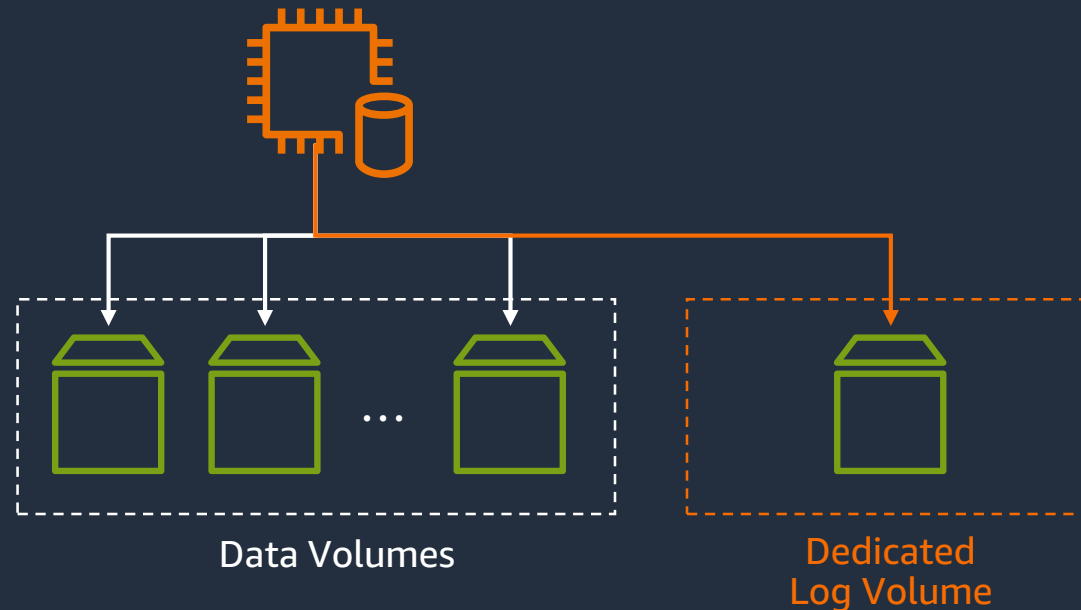
(*) https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding-apg.html#aurora-global-database-write-forwarding-compatibility-apg

Write Forwarding のメリット

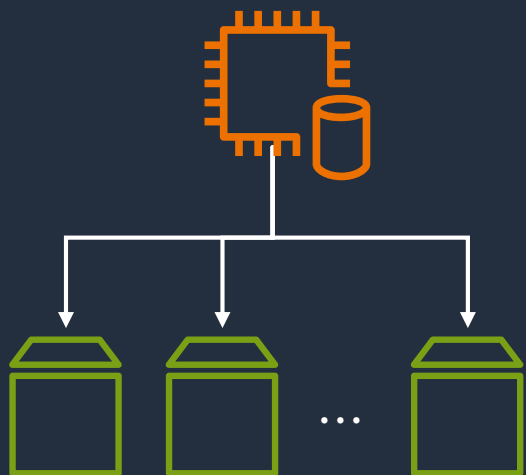
- マネージドソリューション: リモートクラスターで発行された書き込みは、透過的にプライマリクラスターに転送される
- 競合しない: すべての書き込みはプライマリクラスターによって適用されるため、レプリケーション関連の「更新競合」は発生しない
- 柔軟性: 複数の読み取り整合性レベルから選択して、一貫性と許容可能な読み取りレイテンシのバランスを取れる
 - 結果整合性
 - セッションの整合性
 - グローバルな整合性

Amazon RDS が Dedicated Log Volume (専用ログボリューム) をサポート

- Dedicated Log Volumeを使用することで、データベースで最もレイテンシーの影響を受けやすいコンポーネントであるトランザクションログを専用のボリュームに保存可能

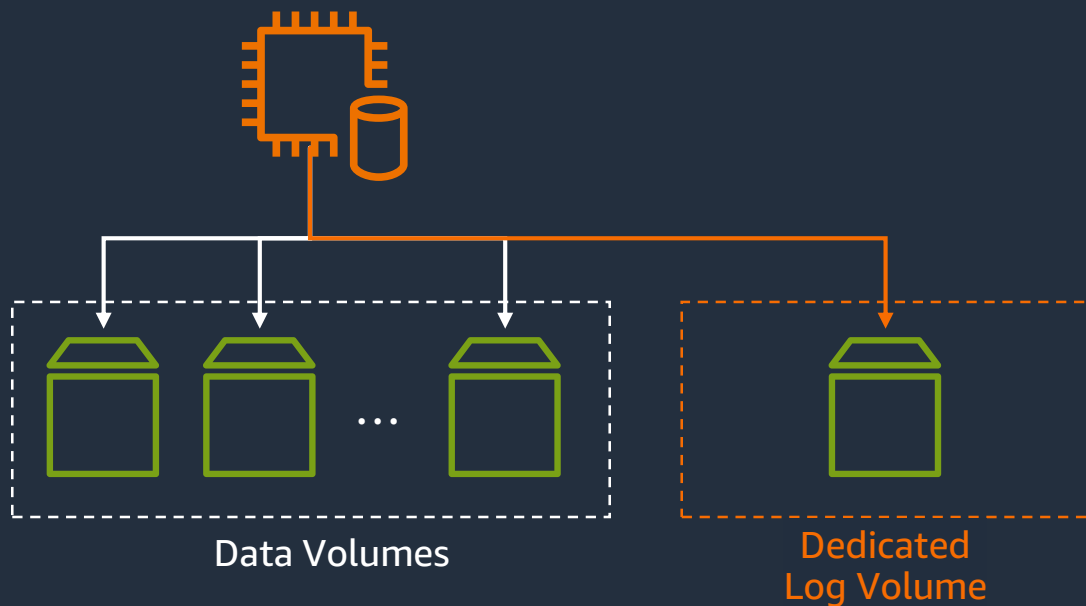


RDS のストレージシステムの概要



- 1 つの AZ につき、1 から 4 のボリュームで構成
- **ボリュームにはトランザクションログとデータページの両方が含まれる**

RDS Dedicated Log Volumes のメリット



- シーケンシャル・ログ書き込みからのI/Oマージの改善
- ディスク・キューでの競合の低減
- ジッターが発生する可能性の低減

DB エンジン	サポートバージョン
PostgreSQL	13 系: 13.10 以上 14 系: 14.7 以上 15 系: 15.2 以上
MySQL	8.0.28 以上
MariaDB	10.6.7 以上

RDS Dedicated Log Volumes の利用とユースケース

- プロビジョンド IOPS ストレージタイプ (io1) と連携
 - 固定サイズ 1,000 GiB、3,000 Provisioned IOPS で作成される
- 5,000 GiB 以上のストレージが割り当てられているデータベースでの使用を推奨
- Dedicated Log Volumes が適しているワークロード
 - データベースのストレージボリュームが大きい
 - IOPS 要件が高い
 - ワークロードがレイテンシーの影響を受けやすい

Amazon Aurora と Amazon RDS が MySQL と PostgreSQL データベースの延長サポートを提供

- MySQL 5.7、PostgreSQL 11、およびそれ以降のメジャーバージョンを実行している Aurora および RDS が Amazon RDS 延長サポートで、コミュニティのサポート終了後も継続してサポート可能に
 - Amazon RDS延長サポートにより、メジャーアップグレードに関連するテストやプロセスに最大3年間の追加工数を取ることができる

延長サポートの適用の例

- RDS for MySQL 5.7 の RDS 標準サポート終了日は 2024/2/29
- しかし、その日までに RDS for MySQL 8.0 に手動でアップグレードする準備ができていない
- 2024/2/29になると、自動的にデータベースを延長サポートに登録し、RDS for MySQL 5.7 を引き続き実行できるようになる
- 2024/3/1 以降、Amazon RDS は自動的に延長サポートの料金を請求する

延長サポートの注意点

- 標準サポートの期間内でのアップグレードを推奨しています
- サードパーティのアプリケーションがサポートしていない場合や、追加で更に検証する期間が必要な場合等に延長サポートを検討してください
- 標準サポート終了までにバージョンアップを実施しない場合は**自動**で延長サポートとなり料金が発生します
 - 延長サポートの料金が1~2年目と3年目で異なる点もご注意ください
- DBエンジン/リージョン毎の料金についてこちらを参照ください
 - Aurora: https://aws.amazon.com/jp/rds/aurora/pricing/#Amazon_RDS_Extended_Support_costs
 - RDS for MySQL: https://aws.amazon.com/jp/rds/mysql/pricing/#Amazon_RDS_Extended_Support_costs
 - RDS for PostgreSQL: https://aws.amazon.com/jp/rds/postgresql/pricing/#Amazon_RDS_Extended_Support_costs