AWS re:Invent Recap – Database

re:Invent 期間中に発表された Amazon RDS/Amazon Aurora 新機能まとめ Part 1

新久保 浩二 (しんくぼ こうじ)
シニア 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
アマゾン ウェブ サービス ジャパン 合同会社

アジェンダ

- **Limited Preview** Amazon Aurora zero-ETL integration with Amazon Redshift
- **New!** Amazon RDS Optimized Writes
- **New!** Amazon RDS Optimized Reads
- **New!** Amazon RDS Blue/Green Deployments

自己紹介

- 新久保　浩二 (しんくぼ　こうじ)
  - 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
    - AWSでデータベースを動かす際のお手伝い
- ライフワーク
    - データベースのパフォーマンス問題の分析や最適化
- お気に入りのAWSサービス
    - Amazon RDS / Amazon Aurora
    - Amazon RDS Performance Insights
    - Amazon DevOps Guru for RDS

限定プレビュー

Amazon Aurora zero-ETL integration with Amazon Redshift (Limited Preview)

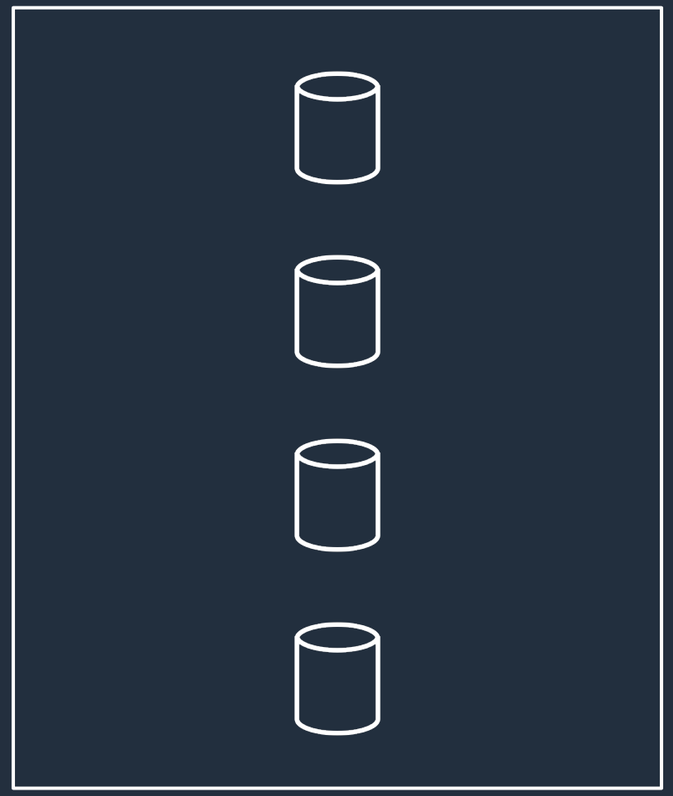
AWS RE:INVENT RE:CAP

限定プレビュー

既存の分析システムへのデータ連携は難しい

個別のシステムに複雑なETLパイプラインが必要

トランザクション系データベース



手動でのETLパイプライン

ETLジョブの構築とメンテナンスに**コストがかかる**

スキーマの変更に伴う**複雑なデータの再構築**

データの**不完全性、一貫性の欠如、陳腐化により、**インサイトが限定

分析系データベース

ETLパイプラインの例

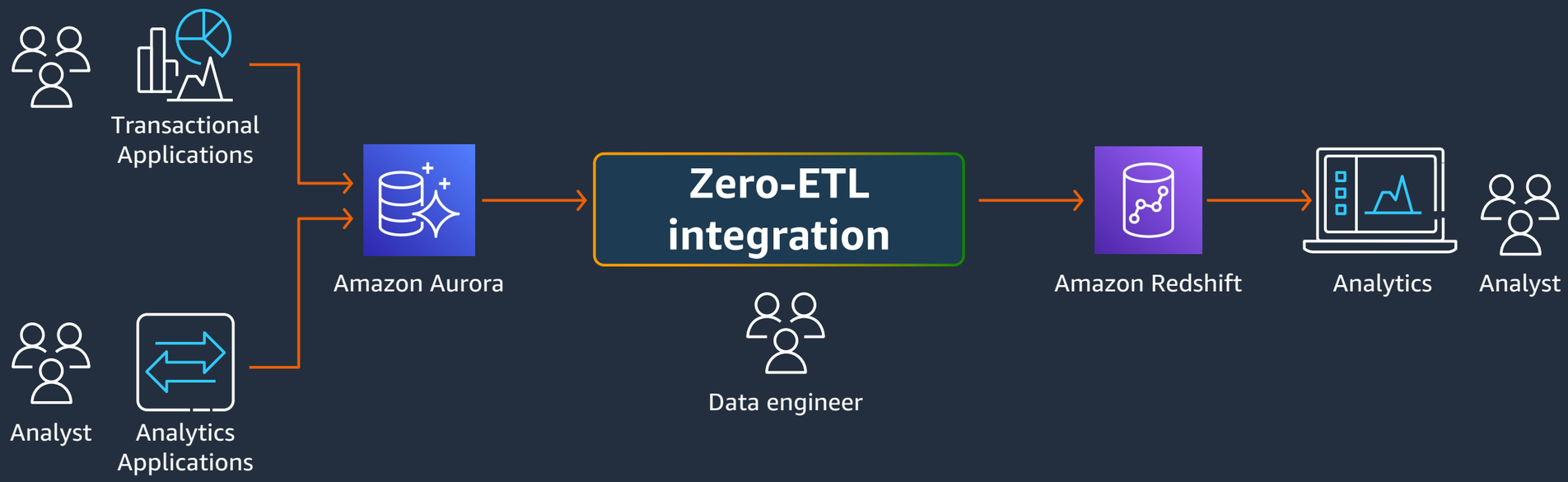
複雑なETLジョブの構築とメンテナンス

限定プレビュー

限定プレビュー

Limited Preview - Amazon Aurora zero-ETL integration with Amazon Redshift

ペタバイトクラスの**トランザクションデータ**を**簡単**、**安全**
かつ、**ニアリアルタイムで分析**することが可能



限定プレビュー

zero-ETL Integrationの利点

複雑なETLジョブを排除して、トランザクションデータからインサイトを得ることに注力

簡単で確実

複雑なETLパイプラインを構築・メンテナンスする必要がなくなる

低レイテンシーでのデータ統合

Amazon Auroraのペタバイトクラスのトランザクションデータに対して、Amazon Redshiftを使用してニアリアルタイムに分析、機械学習を実行可能

統一されたインサイト

複数のデータベースクラスタの統合されたデータから、Amazon Redshiftの高度な分析処理を使ってインサイトを導出可能

プレビュー時の制限とプレビューへの参加

限定プレビュー

- サポートされているデータベースはAurora MySQL 3.0 (MySQL 8.0互換)
- サポートされているリージョンは米国東部 (バージニア北部)

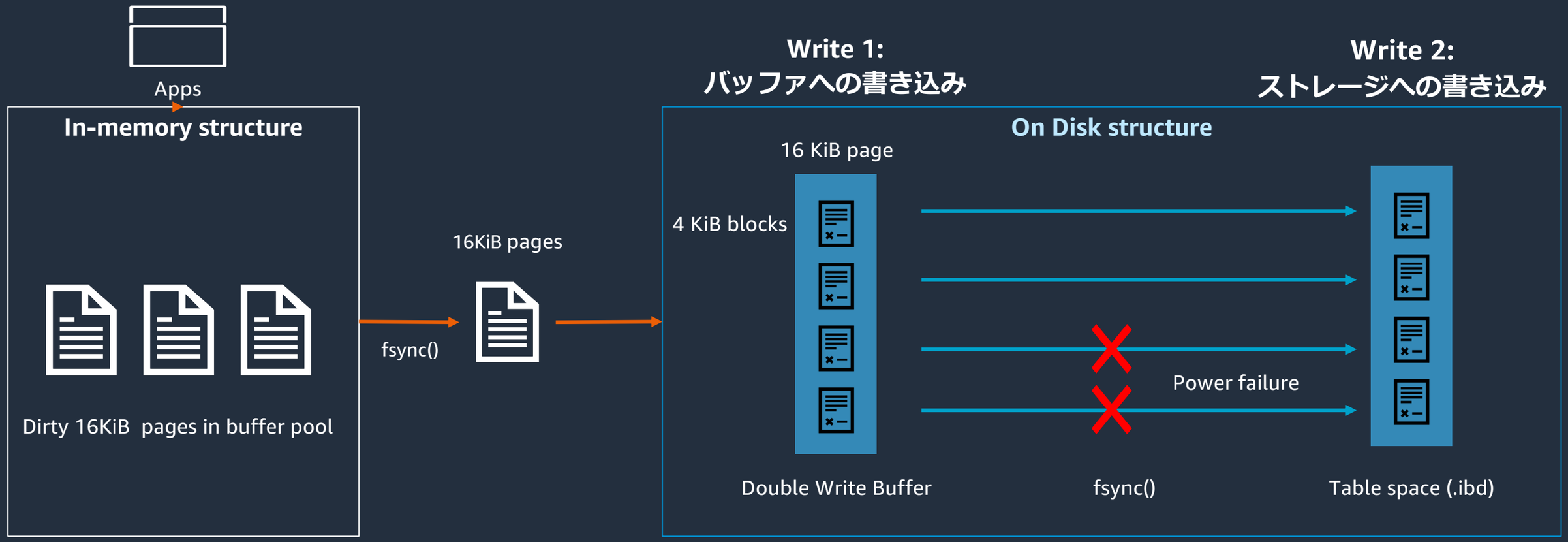
- プレビューの参加については、以下より申し込みをお願いします

  https://pages.awscloud.com/AmazonAurora-zeroETL-AmazonRedshift-Preview.html

Amazon RDS Optimized Writes

(RDS) MySQLが更新データを安全に書き戻す仕組み

チェックポイント時にMySQLがダブルライトバッファでページの破損を防止



ダブルライトバッファの課題

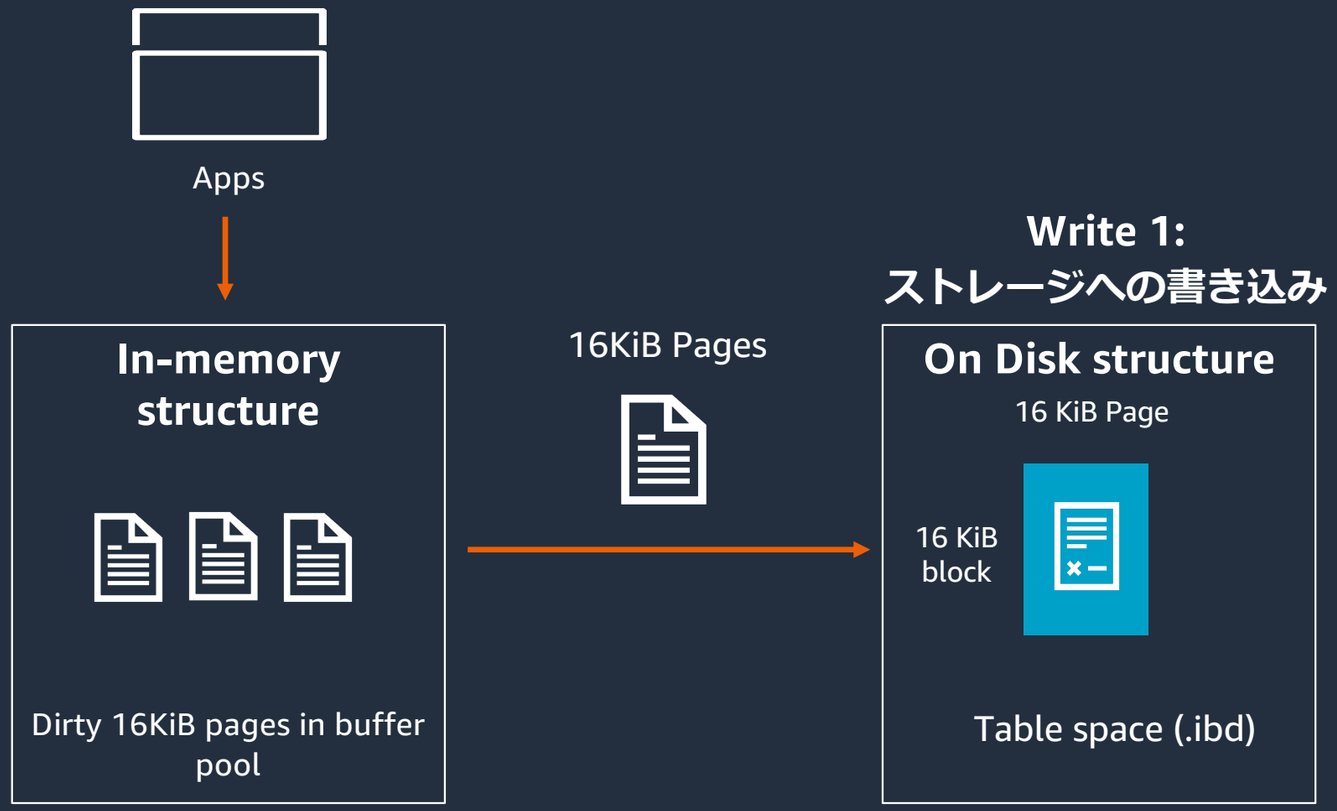
この書き込み方法だと、ユーザーのデータの消失を保護することができるのですが...

...遅い

...効率が悪い

...さらにコストがかかる

New! - Amazon RDS Optimized Writes



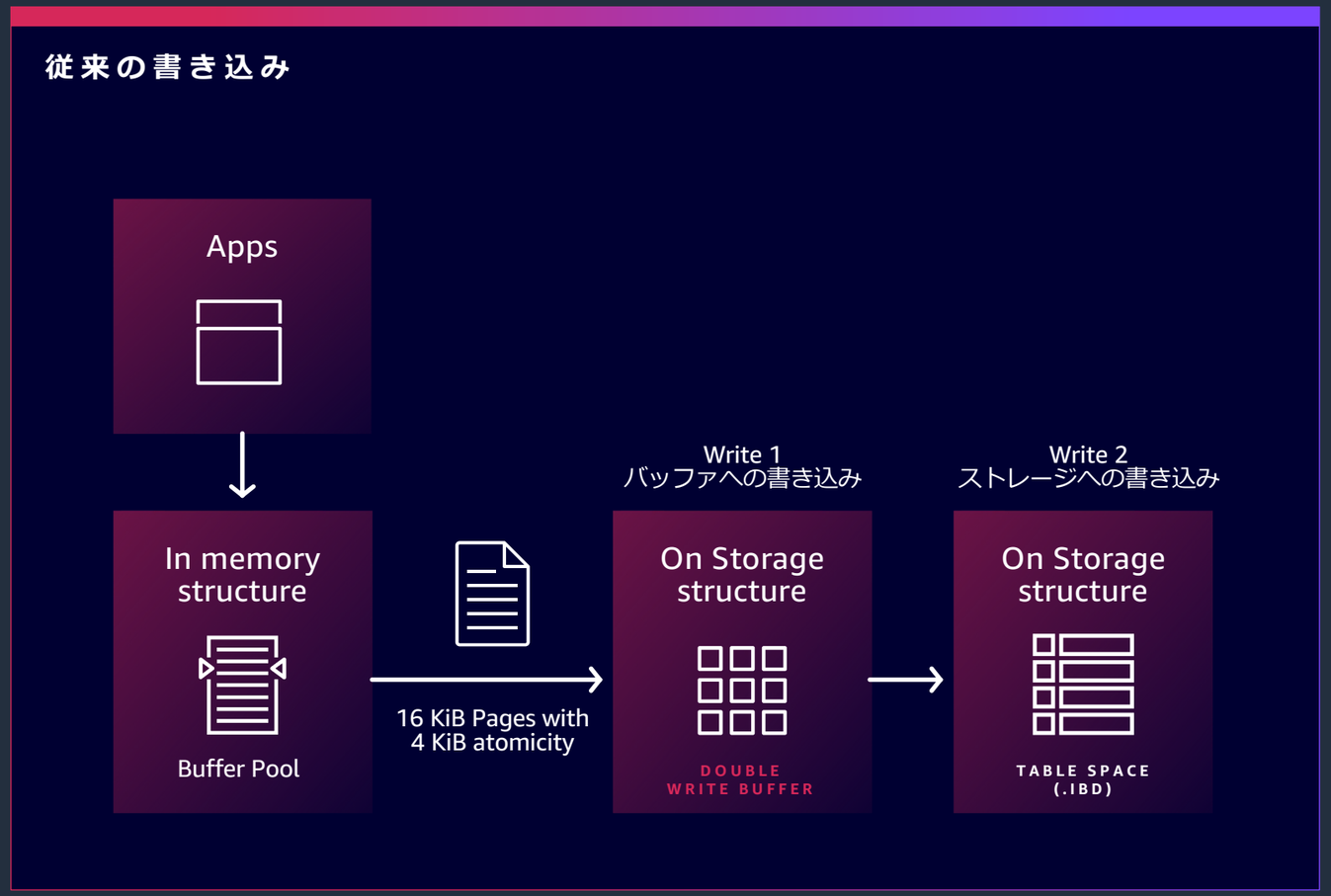
ストレージにおいて、ファイルシステムのページ(ディスク内のデータ)が16KiBページサイズにアライメントされていることを保証

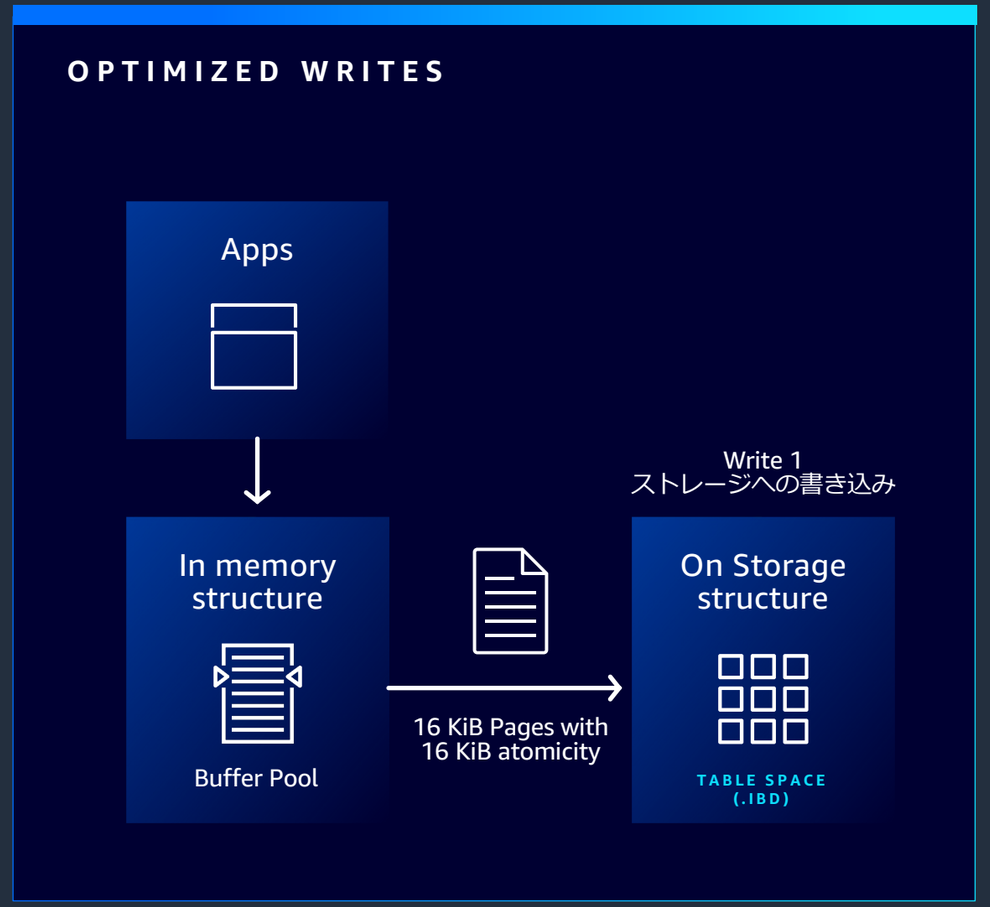
ストレージデバイス16KiBページのアトミックライトをサポートできることを保証 (停電などの障害に耐えられること)

メモリ(バッファプール)から16KiBページをストレージのテーブルデータに1回のアトミック操作で書き込む

従来の書き込みとOptimized Writes

Amazon RDS for MySQLで利用可能





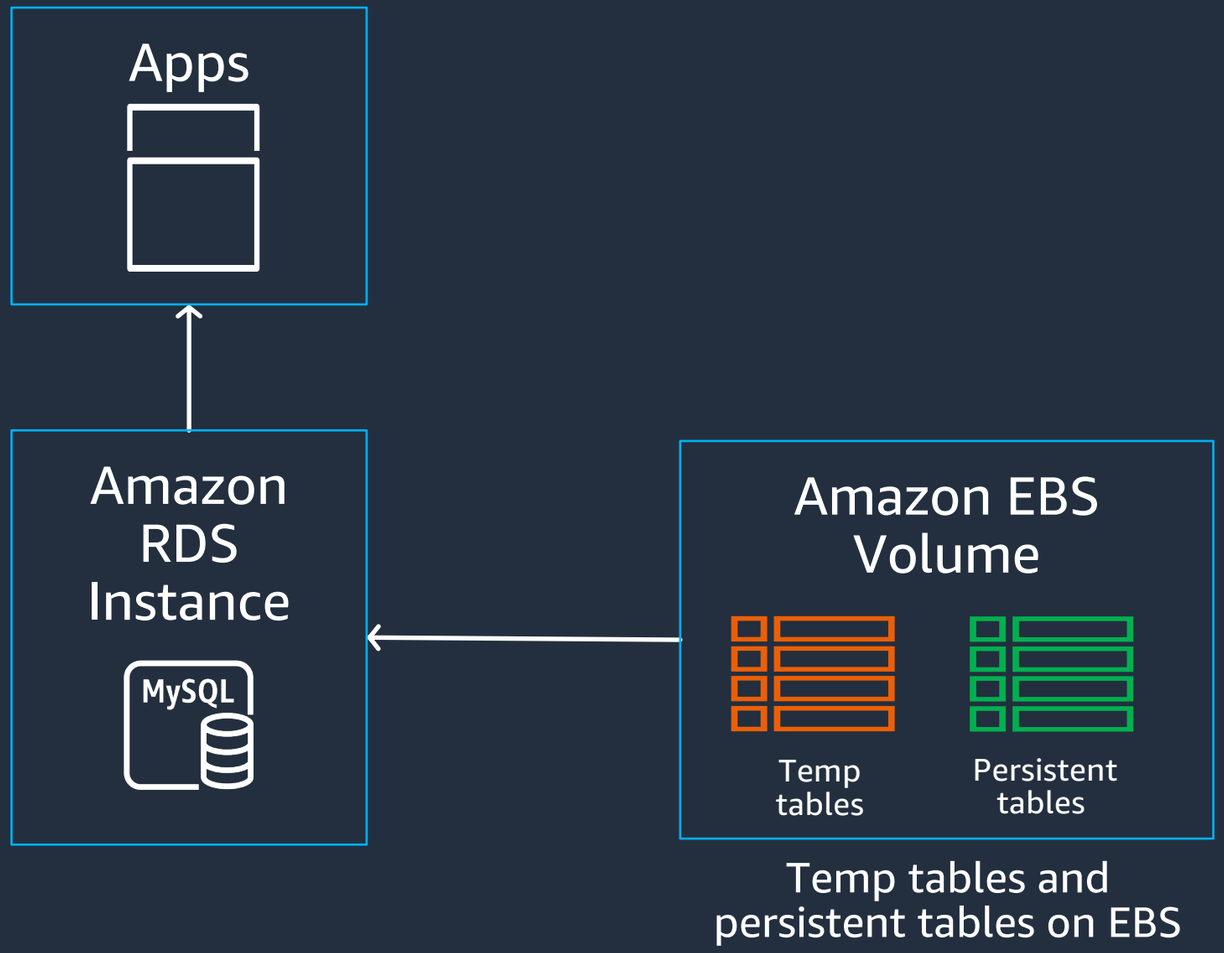
RDS Optimized Writesの注意点

- Optimized Writesがサポートされているバージョン、インスタンスクラスを指定して**新規にインスタンスを作成**し、かつ、パラメータのrds.optimized_writesがAUTO(デフォルト値)に設定されている場合に自動的にOptimized Writesが有効になります
  - Optimized Writesがサポートされているバージョン
    - RDS for MySQL 8.0.30以降
  - Optimized Writesがサポートされているインスタンスクラス
    - db.x2iedn, db.r6i, db.r6g, db.r6gd, db.r5b
  - スナップショットから復元する際は以下のすべての条件を満たす場合のみOptimized Writesが有効
    - Optimized Writesをサポートするインスタンスからスナップショットが作成されている
    - Optimized Writesがサポートされているバージョン、インスタンスクラスに復元
    - パラメータのrds.optimized_writesがAUTO(デフォルト値)に設定されている
- あくまでも、ダブルライトバッファのオーバーヘッドの排除であるため、**すべての書き込み処理が高速化されるわけではありません**

Amazon RDS Optimized Reads

RDS MySQL/MariaDBの一時データの配置場所

ソートや複雑のサブクエリーなどにおける一時データをEBSストレージに書き出す



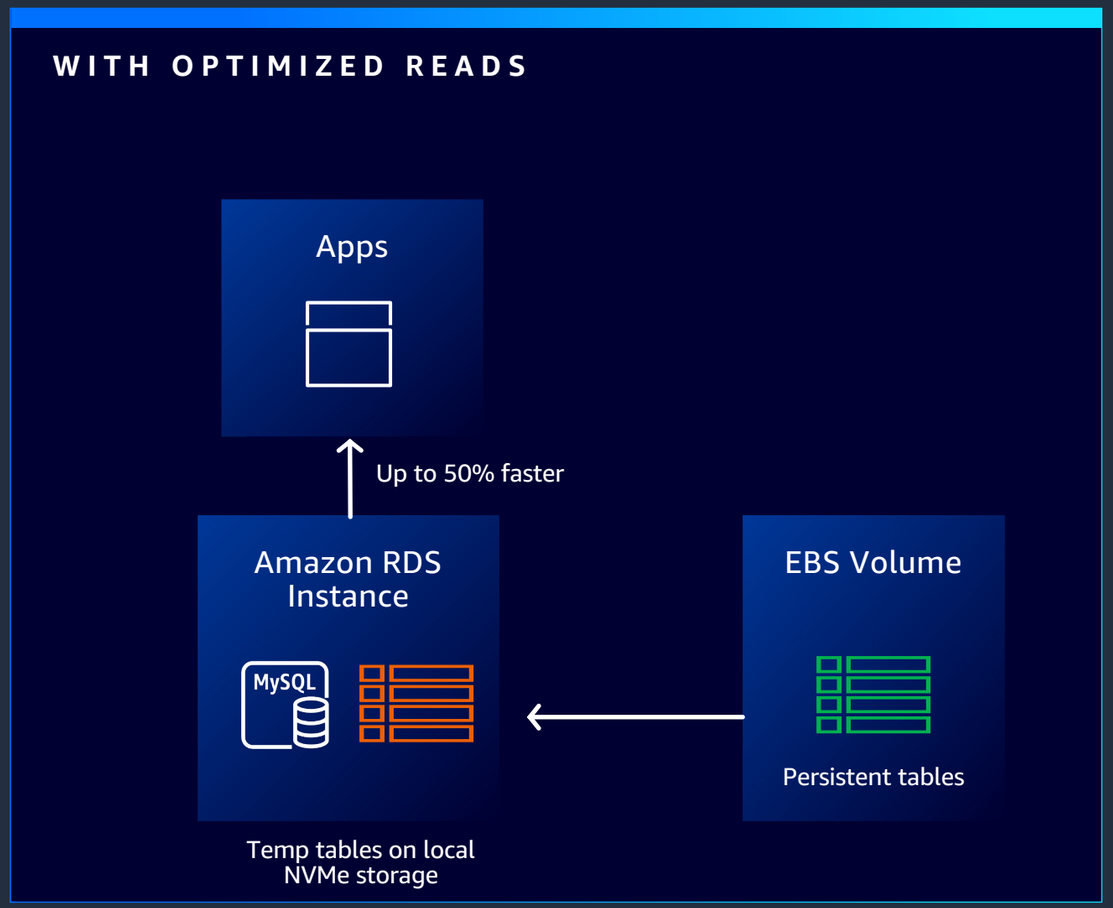
MySQLが複雑なクエリを処理する際に一時データを生成する

メモリに収まらない一時データは、ディスクストレージに書き出される

Amazon RDSでは、一時データは、Amazon EBSから書き込みと読み取りを行う

New! - Amazon RDS Optimized Reads

Amazon RDS for MySQL/MariaDBのワークロードにおいて、クエリーを最大50%高速化



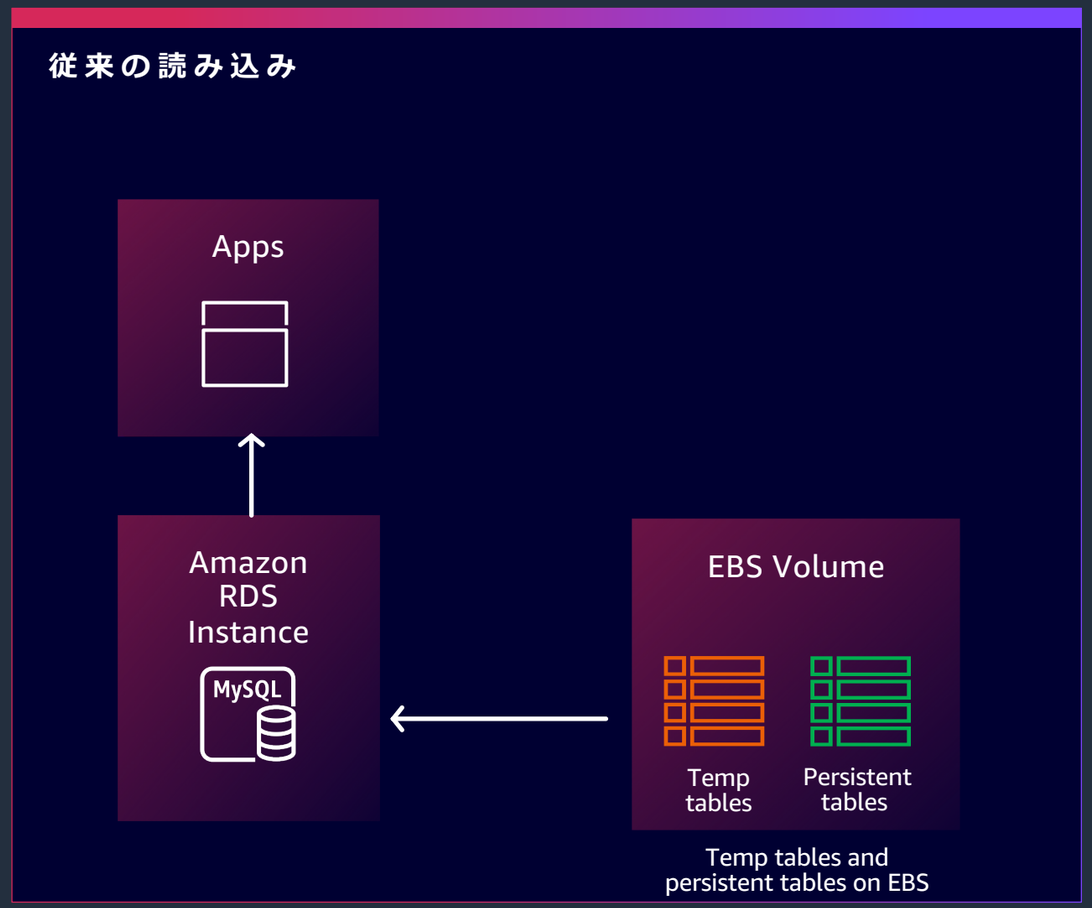
クエリーを最大50%高速化

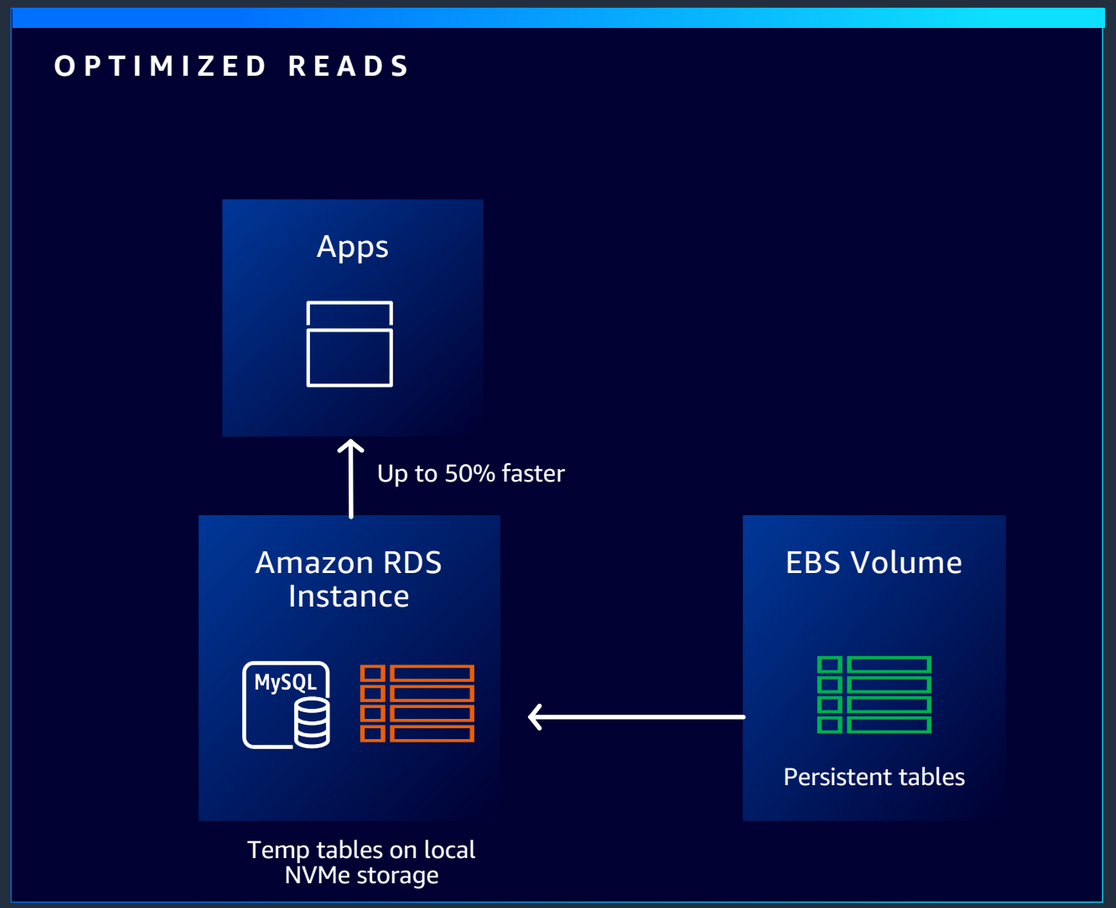
リードレプリカの高速化と複雑なクエリの最適化

Amazon RDSコンソールまたはAPIを使用して、IntelまたはGravitonインスタンスにデプロイ可能

従来の読み込みとOptimized Reads

Amazon RDS for MySQL/MariaDBで利用可能





RDS Optimized Readsの注意点

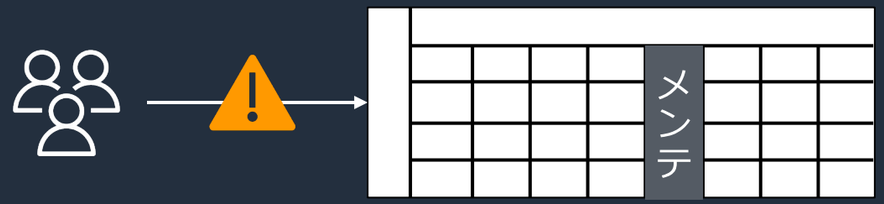
- Optimized Readsがサポートされているバージョン、インスタンスクラスを指定している場合に自動的にOptimized Readsが有効になります
  - Optimized Readsがサポートされているエンジン、バージョン
    - RDS for MySQL 8.0.28以降
    - RDS for MariaDB 10.4.25以降、10.5.16以降、10.6.7以降
  - Optimized Readsがサポートされているインスタンスクラス
    - db.x2idn, db.x2iedn, db.m6gd, db.r6gd, db.m5d, db.r5d
  - Optimized Readsが有効になっているインスタンスで一時データ(オブジェクト)の配置場所をインスタンスストアからEBSに変更することはできません
  - binlogが有効になっている場合、トランザクションサイズの最大はインスタンスストアのサイズに制限されます
  - インスタンスストアが一杯になるとトランザクションが失敗する可能性があります
- あくまでも、一時データ(オブジェクト)の配置先がインスタンスストアに変更される最適化であるため、**すべての読み込み処理が高速化されるわけではありません**

Amazon RDS Blue/Green Deployments

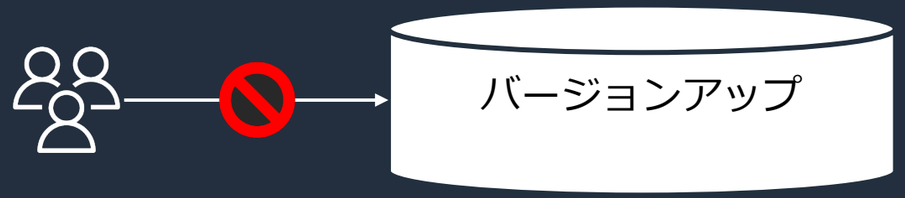
一般的なデータベースメンテナンスの課題

MySQLでの一般的なDBメンテナンス

- スキーマ変更などのメンテナンス作業におけるダウンタイム
  - オンラインDDLなどが導入されていますが、カラムの削除やデータ型の変更などデータが大幅に変更され、コストの高い作業の場合に大きなダウンタイムが発生する



- メジャー/マイナー/パッチ(*)のアップグレード時のダウンタイム
  - アップグレード関連のメンテナンスではダウンタイムが回避できない(*)



(*) Auroraのゼロダウンタイムパッチ適用を除く

(参考) RDS/Auroraのアップグレード方法

一般的なRDS/Aurora MySQLのアップグレード方法

インプレースアップグレード

シンプル – ワンクリックで実施可能な管理されたワークフロー

BUT

- 本番環境での(比較的長い)ダウンタイム
- 予測が難しいダウンタイム時間

Blue/Green Deployment(Self-Managed)

- **高速** – ワークロードに応じた短いダウンタイム時間

BUT

- (比較的)複雑な構築手順のためのナレッジ
- ステージング環境を維持する工数

(参考) RDS/Auroraのアップグレード方法

参考ドキュメント

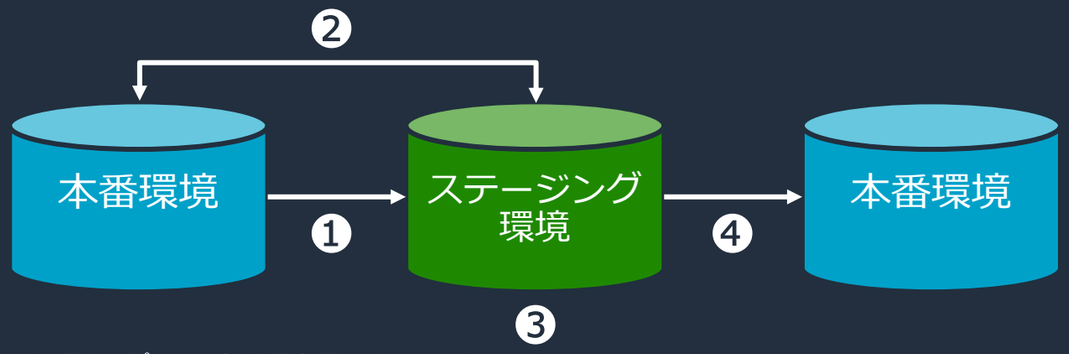
- インプレースアップグレード
  - RDS for MySQL/MariaDB
    - MySQL のメジャーバージョンのアップグレード
    - MariaDB のメジャーバージョンのアップグレード
  - Aurora MySQL
    - Aurora MySQL インプレースアップグレードのチュートリアル
- Blue/Green Deployment (Self-Managed)
  - RDS for MySQL/MariaDB
    - MySQL データベースのアップグレード時にリードレプリカを使用したダウンタイムの短縮
  - Aurora MySQL
    - 代替の Blue/Green のアップグレードテクニック

Blue/Green Deployment(Self-Managed)の概要

一般的なMySQLのbinlog replicationを用いたBlue/Green Deployment

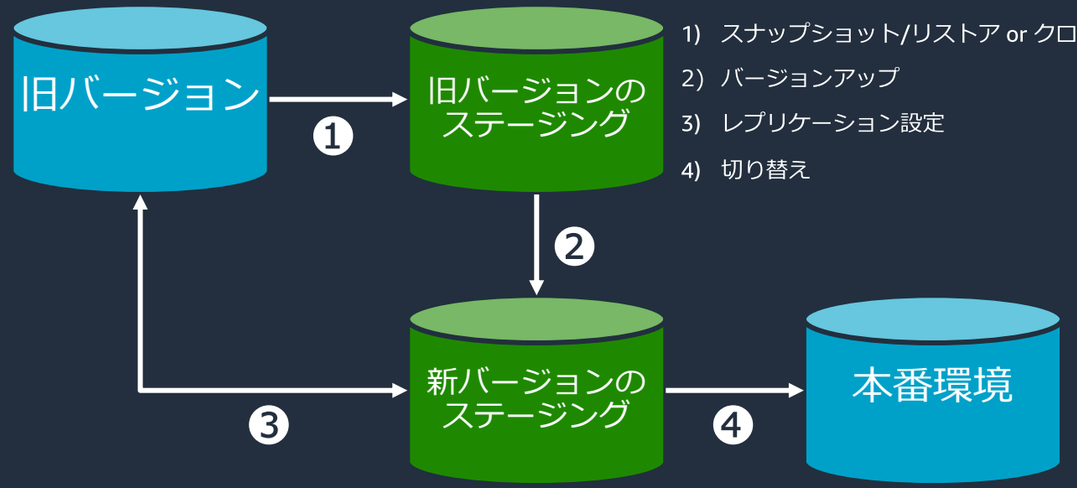
- RDS for MySQLやAurora MySQLではMySQLで一般的なbinlogを利用したレプリケーション環境を構築し切り替えることでダウンタイムを最小化できます
(* RDSの場合はステージング環境の作成/アップグレードはリードレプリカの作成/アップグレードとしてコンソール/APIから実施可能)

スキーマ変更の場合



1) スナップショット/リストア or クローン(Aurora)
2) レプリケーション設定
3) スキーマ変更
4) 切り替え

アップグレードの場合



1) スナップショット/リストア or クローン(Aurora)
2) バージョンアップ
3) レプリケーション設定
4) 切り替え

https://aws.amazon.com/jp/blogs/database/performing-major-version-upgrades-for-amazon-aurora-mysql-with-minimum-downtime/

https://aws.amazon.com/jp/blogs/news/how-to-upgrade-amazon-rds-from-mysql5-5-to-mysql5-7/

Blue/Green Deployment(Self-Managed)の概要

一般的なMySQLのbinlog replicationを用いたBlue/Green Deployment

- RDS for MySQLやAurora MySQLではMySQLで一般的なbinlogを利用したレプリケーション環境を構築し切り替えることでダウンタイムを最小化できます

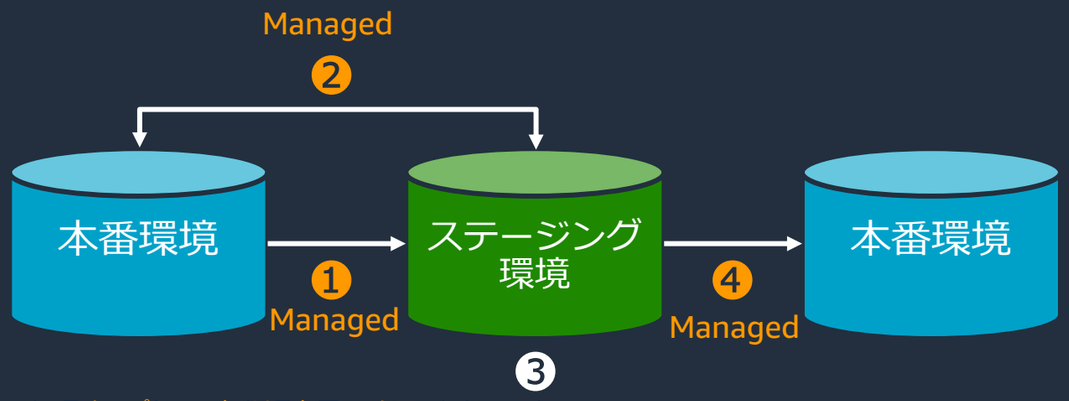
- Self-Managed Blue/Green Deploymentsの課題
  - Blue/Green環境の構築手順が複雑
    (RDSの場合はステージング環境の作成/アップグレードはリードレプリカの作成/アップグレードとしてコンソール/APIから実施可能)
  - 正しい手順で構築/切り替えをしないとデータロストのリスクが存在する
  - 切り替え後のエンドポイント変更など後処理にも手作業が発生する

New! - Amazon RDS Blue/Green Deployments

フルマネージによるBlue/Green Deploymentを実現

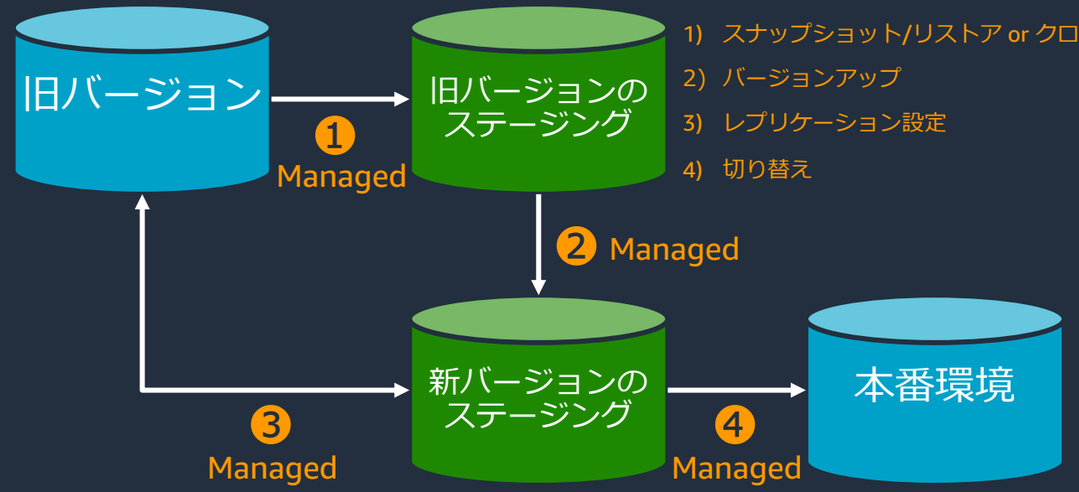
- RDS for MySQLやAurora MySQLで従来は手動で作成していたレプリケーション環境をコンソールやAPIから簡単に作成できるようになりました
- 切り替え時もコンソール/APIから実行でき、レプケーションラグがゼロ(完全同期)になるのを待って切り替え処理を実施します

スキーマ変更の場合



1) スナップショット/リストア or クローン(Aurora)
2) レプリケーション設定
3) スキーマ変更
4) 切り替え

アップグレードの場合



1) スナップショット/リストア or クローン(Aurora)
2) バージョンアップ
3) レプリケーション設定
4) 切り替え

Amazon RDS Blue/Green Deploymentsの利点

作業の複雑さやオペレーションミスによるリスクの回避

- セキュリティを常に最新の状態に保ち、データベースパフォーマンスを向上させ、より新しいデータベース機能を短時間で予測可能なダウンタイムで採用することが可能
- メジャー/マイナーのバージョンアップなどデータベースの更新に伴うリスクとダウンタイムを低減することが可能
  - 本番環境と同じステージング環境を簡単に構築可能にし自動でレプリケーション設定を実施
  - 本番環境に影響を与えることなく、安全なステージング環境でデータベースの変更をテストすることが可能
  - データベースのパッチやシステムの更新を最小のダウンタイムで実施することが可能
  - アプリケーションの変更なしに、ステージング環境を新しい本番環境に切り替えが可能
    - 組み込みのガードレールにより、安全に切り替え
    - 切り替え時のデータ損失を排除
    - ワークロードにもよりますが、通常1分以内という迅速な切り替えが可能

Amazon RDS Blue/Green Deploymentsの注意事項

MySQLのbinlogレプリケーションの仕組みを理解してステージング環境の運用が必要 (1)

```
mysql> desc blue_green;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| c1    | varchar(10) | YES  |     | NULL    |       |
| c2    | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> insert into blue_green values (1,'1','1');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from blue_green;
+----+------+------+
| id | c1   | c2   |
+----+------+------+
|  1 | 1    | 1    |
+----+------+------+
1 row in set (0.00 sec)

mysql> alter table blue_green drop c2;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

例えば、ステージング(Green)環境で破壊的な変更(アプリケーションで利用しているカラムの削除)を実施した場合や…

Amazon RDS Blue/Green Deploymentsの注意事項

MySQLのbinlogレプリケーションの仕組みを理解してステージング環境の運用が必要 (2)

レプリケーションに対して破壊的な
変更をおこなった場合、
レプリケーションはエラーにより
中断します。

```
mysql> insert into blue_green values (2,'2','2');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> show replica status\G
*************************** 1. row ***************************
               Replica_IO_State: Waiting for master to send event
                    Source_Host: 172.23.2.209
                    Source_User: rdsrepladmin
                    Source_Port: 3306
                  Connect_Retry: 60
                Source_Log_File: mysql-bin-changelog.000592
            Read_Source_Log_Pos: 154
                 Relay_Log_File: relaylog.000111
                  Relay_Log_Pos: 569
          Relay_Source_Log_File: mysql-bin-changelog.000591
             Replica_IO_Running: Yes
            Replica_SQL_Running: No
                Replicate_Do_DB:
            Replicate_Ignore_DB:
             Replicate_Do_Table:
         Replicate_Ignore_Table: innodb_memcache.cache_policies,innod
b_memcache.config_options,mysql.plugin,mysql.rds_configuration,mysq
l.rds_history,mysql.rds_monitor,mysql.rds_replication_status,mysql.
rds_sysinfo
       Replicate_Wild_Do_Table:
   Replicate_Wild_Ignore_Table:
                    Last_Errno: 1136
                    Last_Error: Coordinator stopped because there we
re error(s) in the worker(s). The most recent failure being: Worker
 1 failed executing transaction 'ANONYMOUS' at master log mysql-bin
-changelog.000591, end_log_pos 703. See error log and/or performanc
e_schema.replication_applier_status_by_worker table for more detail
s about this failure or others, if any.
                  Skip_Counter: 0
```

Amazon RDS Blue/Green Deploymentsの注意事項

MySQLのbinlogレプリケーションの仕組みを理解してステージング環境の運用が必要 (3)

Blue/Green Deploymentsの状況やステージング(Green)環境のイベントから確認可能

正常

DB 識別子	レプリケーションの状態	遅延
rds-mysql Blue	-	-
rds-mysql-green-59gch6 Green	レプリケーション中	-

異常

DB 識別子	レプリケーションの状態	遅延
rds-mysql Blue	-	-
rds-mysql-green-59gch6 Green	レプリケーション中	-1 秒

異常

最近のイベント (25)

フィルター条件 db イベント

1 2 3 4 5

時間	システムノート
December 02, 2022, 13:49 (UTC+09:00)	Read Replica Replication Error - SQLError: 1136, reason: Coordinator stopped because there were error(s) in the worker(s). The most recent failure being: Worker 1 failed executing transaction 'ANONYMOUS' at master log mysql-bin-changelog.000598, end_log_pos 413. See error log and/or performance_schema.replication_applier_status_by_worker table for more details about this failure or others, if any.
December 02, 2022, 13:50 (UTC+09:00)	Read Replica Replication Error - SQLError: 1136, reason: Coordinator stopped because there were error(s) in the worker(s). The most recent failure being: Worker 1 failed executing transaction 'ANONYMOUS' at master log mysql-bin-changelog.000598, end_log_pos 413. See error log and/or performance_schema.replication_applier_status_by_worker table for more details about this failure or others, if any.

正常(に復旧)

最近のイベント (25)

フィルター条件 db イベント

1 2 3 4 5

時間	システムノート
December 02, 2022, 14:13 (UTC+09:00)	Replication for the Read Replica resumed

Amazon RDS Blue/Green Deploymentsの要件

Amazon RDS Blue/Green Deploymentsがサポートしているリージョンとバージョン

- RDS (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RDS_Fea_Regions_DB-eng.Feature.BlueGreenDeployments.html)
  - エンジン
    - RDS for MySQL (全バージョン)
    - RDS for MariaDB (全バージョン)
  - リージョン
    - China(Beijing、Ningxia)以外の全てのリージョン
- Aurora (https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.Aurora_Fea_Regions_DB-eng.Feature.BlueGreenDeployments.html)
  - エンジン
    - Aurora MySQL (全バージョン) * Blue環境でAurora MySQLを利用する場合はbinlogを有効にする必要があります
  - リージョン
    - China(Beijing、Ningxia)以外の全てのリージョン

Amazon RDS Blue/Green Deploymentsの制限事項

- 制限事項
  - Amazon RDS Proxy
  - Cascading read replicas (RDS only)
  - Cross region read replicas
  - Multi-AZ DB clusters (RDS only)
  - AWS CloudFormation
  - Aurora Serverless v1 DB clusters (Aurora only)
  - グローバルデータベースの一部になっているクラスター(Aurora only)
  - 非暗号化環境から暗号化環境への変更
  - 暗号化環境から非暗号化環境への変更
  - 本番(Blue)環境をステージング(Green)環境より高いバージョンにできない

Amazon RDS Blue/Green Deploymentsの料金

- RDS/Aurora共にAmazon RDS Blue/Green Deploymentsに関する費用は不要
  - ただし、新たに作成するGreen環境の料金は別途必要になります
    - マネジメントコンソールからAmazon RDS Blue/Green Deploymentsを利用する場合は、Green環境の推定コストが提示されます

Amazon RDS Blue/Green Deploymentsの情報源

- What’s new
  - https://aws.amazon.com/about-aws/whats-new/2022/11/amazon-rds-blue-green-deployments-safer-simpler-faster-updates/
- Blog “New – Fully Managed Blue/Green Deployments in Amazon Aurora and Amazon RDS”
  - https://aws.amazon.com/jp/blogs/aws/new-fully-managed-blue-green-deployments-in-amazon-aurora-and-amazon-rds/
- User Guide (RDS)
  - https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-overview.html
- Users Guide (Aurora)
  - https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Thank you!