

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

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



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ましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)

自己紹介

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



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

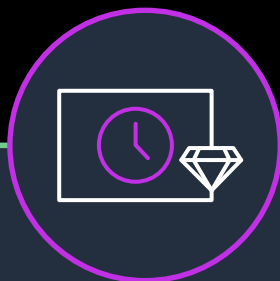
- クラウド向けに再設計されたMySQL, PostgreSQLと互換性のあるRDBMS
- コマーシャルデータベースの性能と可用性を1/10のコストで

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

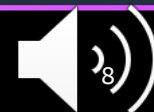


ネットワーク分離、保管時/通信の暗号化

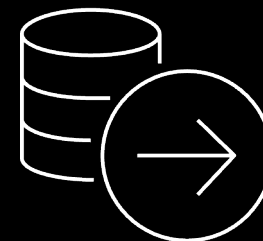
フルマネージド



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.

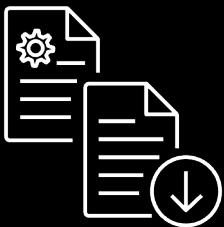
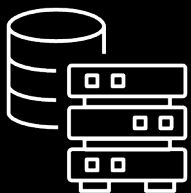


チームの生産性を高める

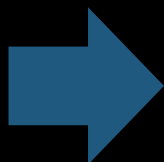
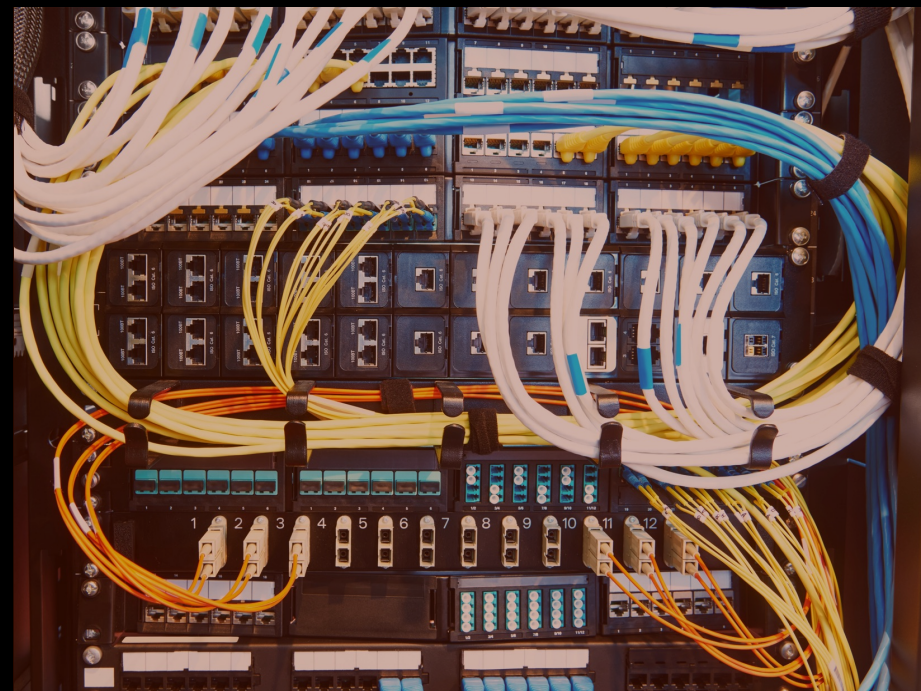
時間のかかるデータベース管理タスクや、複雑なライセンス管理から解放され、アプリケーションやビジネスに集中できる



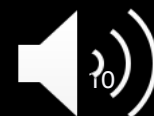
データベースの移行作業には多くの工数がかかる



- ✓ 経験や知見が不足している中で
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）

計画策定

解決すべき課題特定

- ビジネス成長への追従
- コスト構造の把握、削減
- 可用性/セキュリティの担保
- 人材不足への対応

アセスメント

- DB エンジン変更可否
- 移行先 DB サイジング
- コード変換の難易度調査

PoC

- コード変換
- 機能検証
- 性能検証
- データ移行検証
- 費用対効果検証

移行実施

コード変換

- DBオブジェクトの変換
(表や索引など)
- DBコードオブジェクトの変換
(プロシージャなど)
- アプリケーションSQLの変換

データ移行

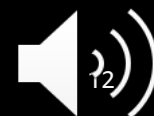
- データ移行
- 整合性の確認

テスト実施

- 単体テスト
- 結合テスト
- 性能テスト
- 負荷テスト/ストレステスト

切り替え

- 移行の正常性確認
- 切り替え



移行計画の策定に対する支援プログラム

計画
策定

解決すべき課題特定

- ビジネス成長への追従
- コスト構造の把握、削減
- 可用性/セキュリティの担保
- 人材不足への対応

アセスメント

- DB エンジン変更可否
- 移行先 DB サイジング
- コード変換の難易度調査

PoC

- コード変換
- 機能検証
- 性能検証
- データ移行検証
- 費用対効果検証

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
- ワークロードレポートに基づいたサイジング、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 Schema Conversion Tool (SCT)

スキーマ、ビュー、プロシージャ、ファンクションの大部分を自動的に変換

Oracle SQL Server IBM Db2
MySQL MariaDB PostgreSQL
Azure SQL Database Sybase etc...



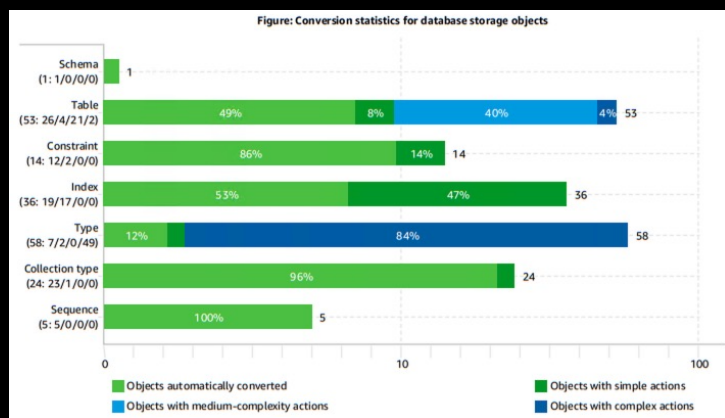
MySQL
PostgreSQL



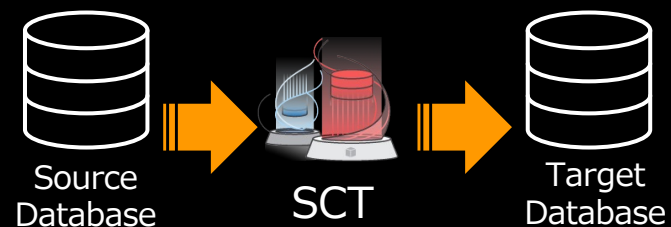
etc...



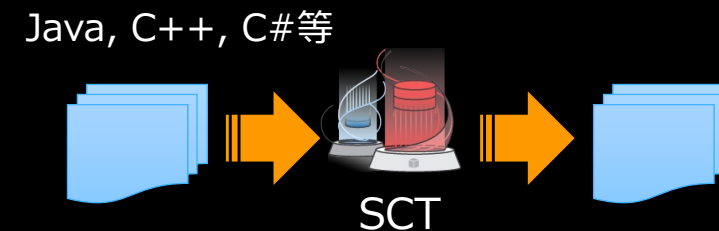
AWS SCTの使用用途



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



Configuration			
Oracle	Aurora PostgreSQL	Key Differences	Compatibility
Alerting	Alerting	• Use Event Notifications Subscription with Amazon Simple Notification Service (SNS)	🟢🟢🟢🟢🟢
Cache and Pools	Cache and Pools	• Different cache names, similar usage	🟢🟢🟢🟢🟢
Database Parameters	Database Parameters	• Use Cluster and Database/Cluster Parameters	🟢🟢🟢🟢🟢
Session Parameters	Session Parameters	• SET options are significantly different in PostgreSQL	🟢🟢🟢🟢🟢
Special Features			
Oracle	Aurora PostgreSQL	Key Differences	Compatibility
Character Set	Character Set	• UTF16 character and NCHAR/NVARCHAR data types are not supported	🟢🟢🟢🟢🔴
Database Links	Database Links	• Different paradigm and syntax	🟢🟢🟢🟢🔴
DBMS_SCHEDULER	DBMS_SCHEDULER	• Use Scheduler AWS Lambda	🟢🟢🟢🟢🔴
External Tables	External Tables	• PostgreSQL doesn't support EXTERNAL TABLES	🟢🟢🟢🟢🔴
Inline Views	Inline Views		🟢🟢🟢🟢🟢
JSON	JSON	• 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));

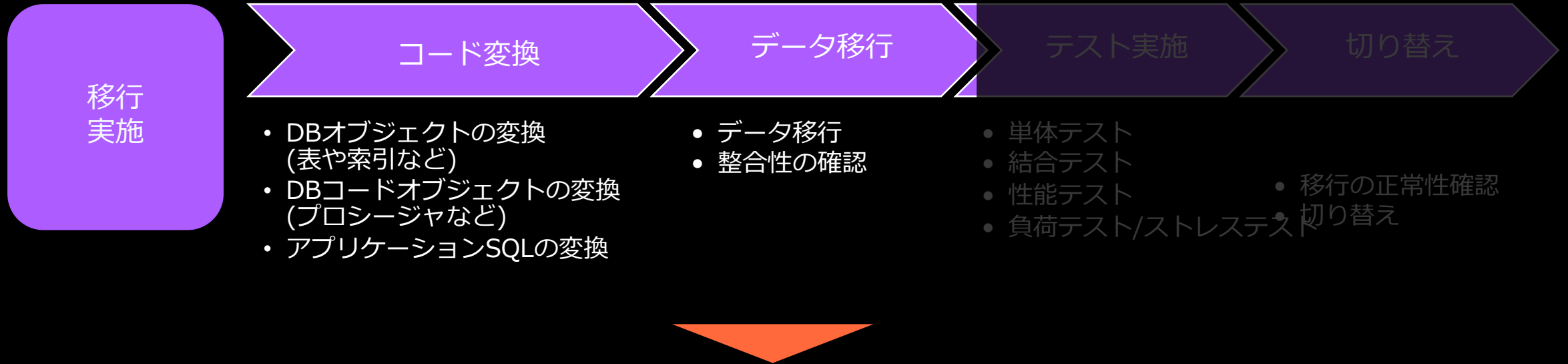
MERGE INTO EMP_BONUS E1
USING (SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEES) E2 ON (E1.EMPLOYEE_ID = E2.EMPLOYEE_ID) WHEN MATCHED THEN
UPDATE SET E1.BONUS = E2.SALARY * 0.5
DELETE WHERE (E1.SALARY >= 10000)
WHEN NOT MATCHED THEN
INSERT (E1.EMPLOYEE_ID, E1.BONUS_YEAR, E1.SALARY, 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_YEAR SALARY BONUS
-----
103 2017 9000 4500
104 2017 6000 3000
102 2017 4800 2400
```



Aurora へのデータ移行

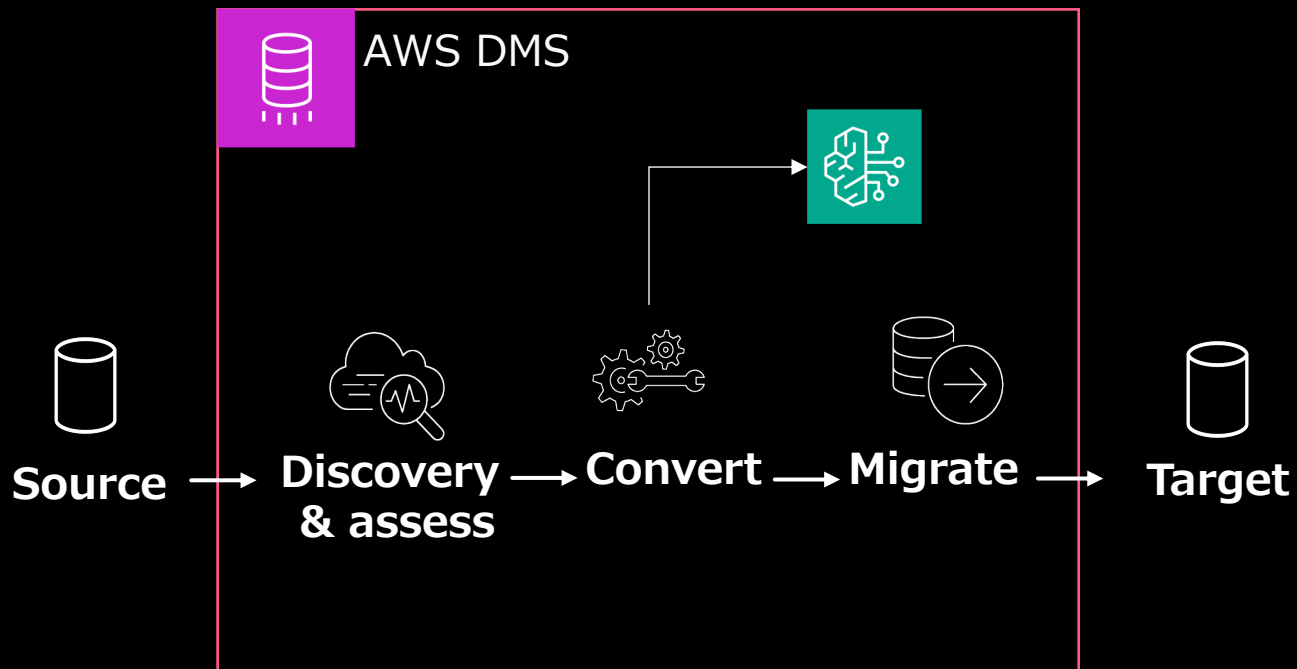


AWS Database Migration Service (DMS)

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

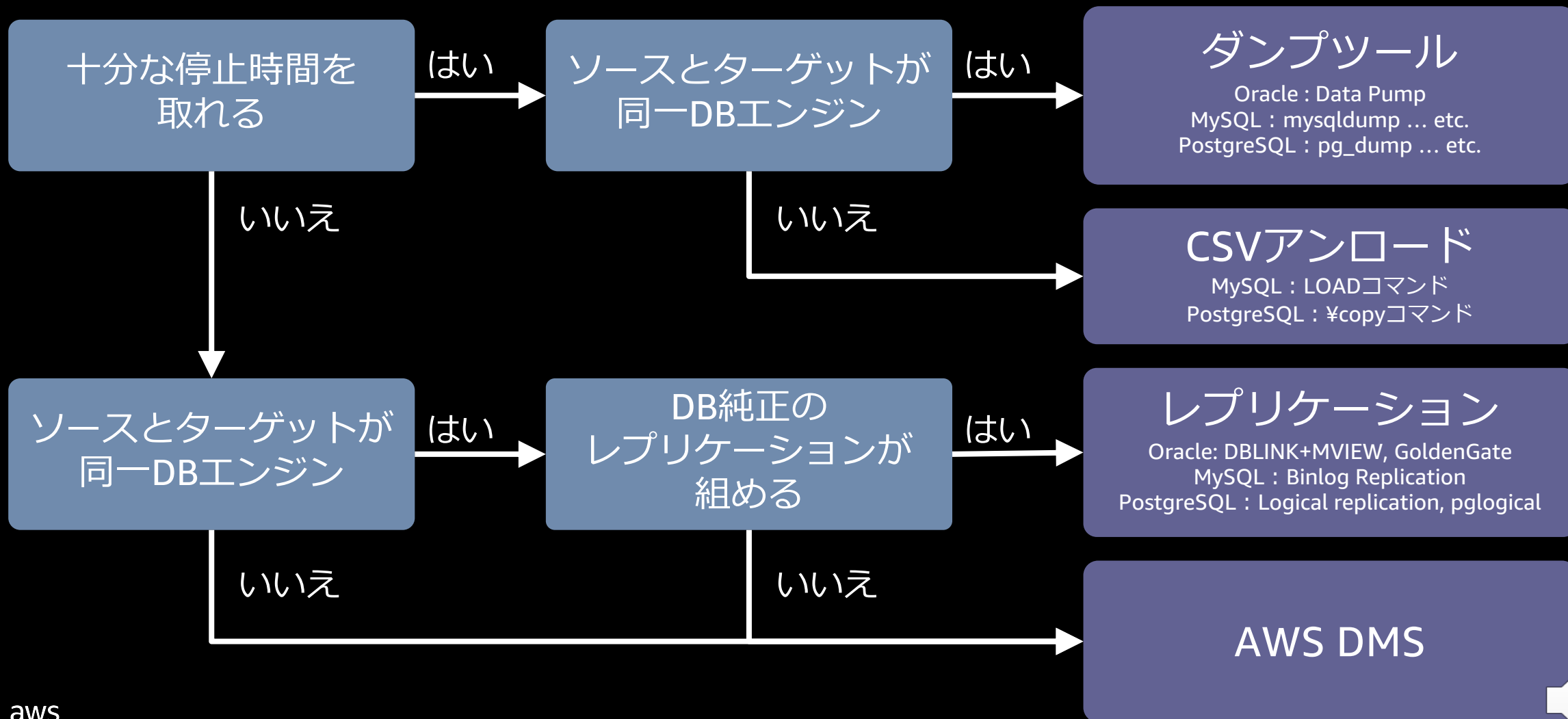
データ検証

- 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

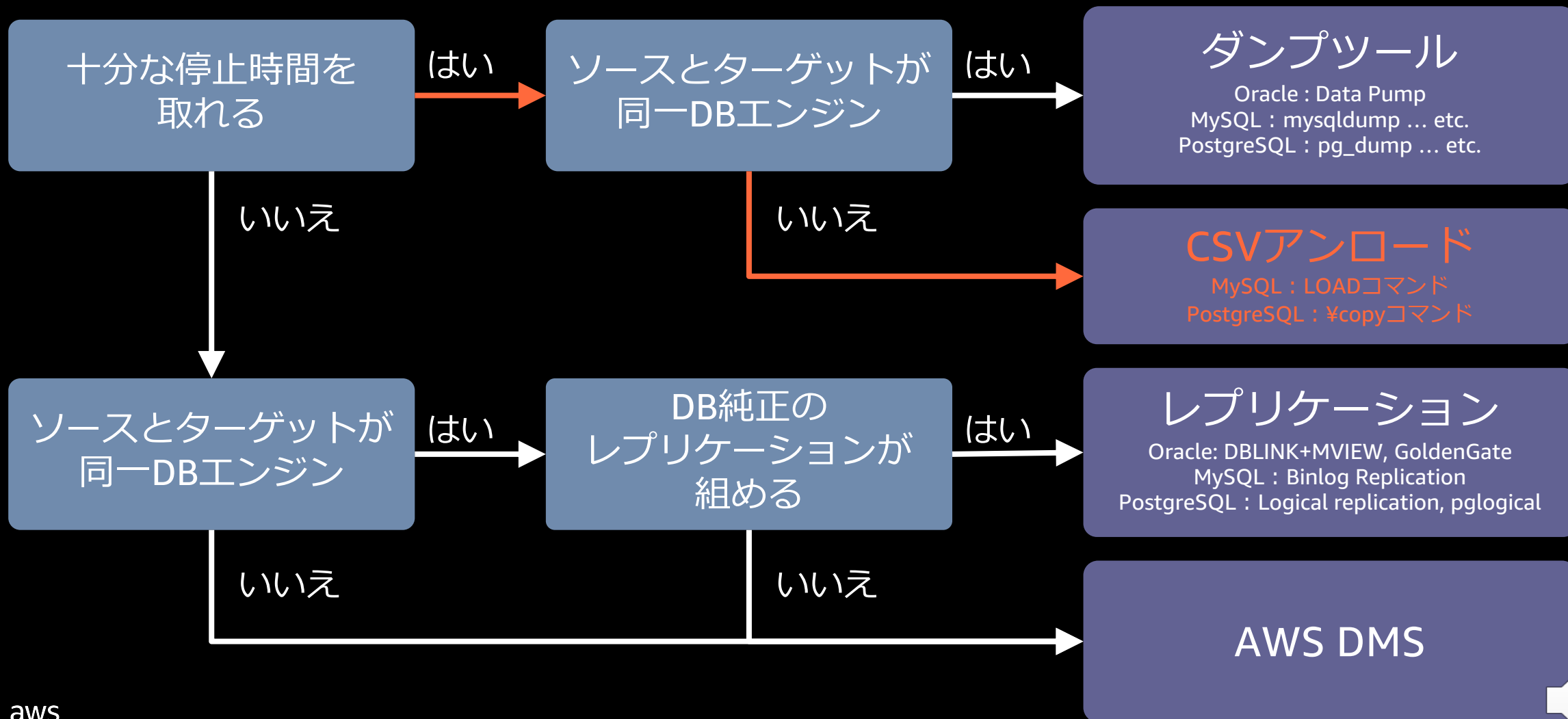
スキーマ変換 **New!**

- 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

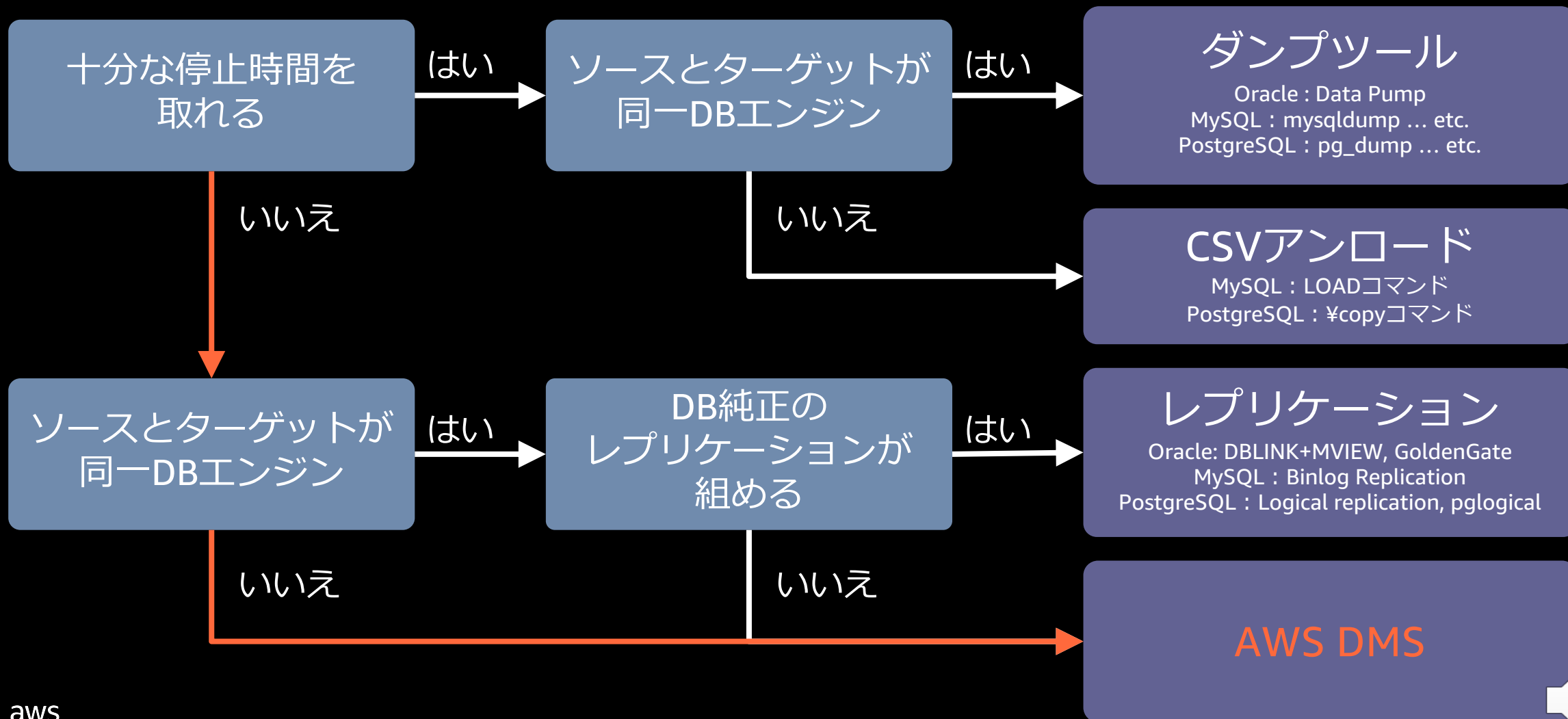
データ移行の選択肢と選定ポイント



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
<ul style="list-style-type: none">• 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

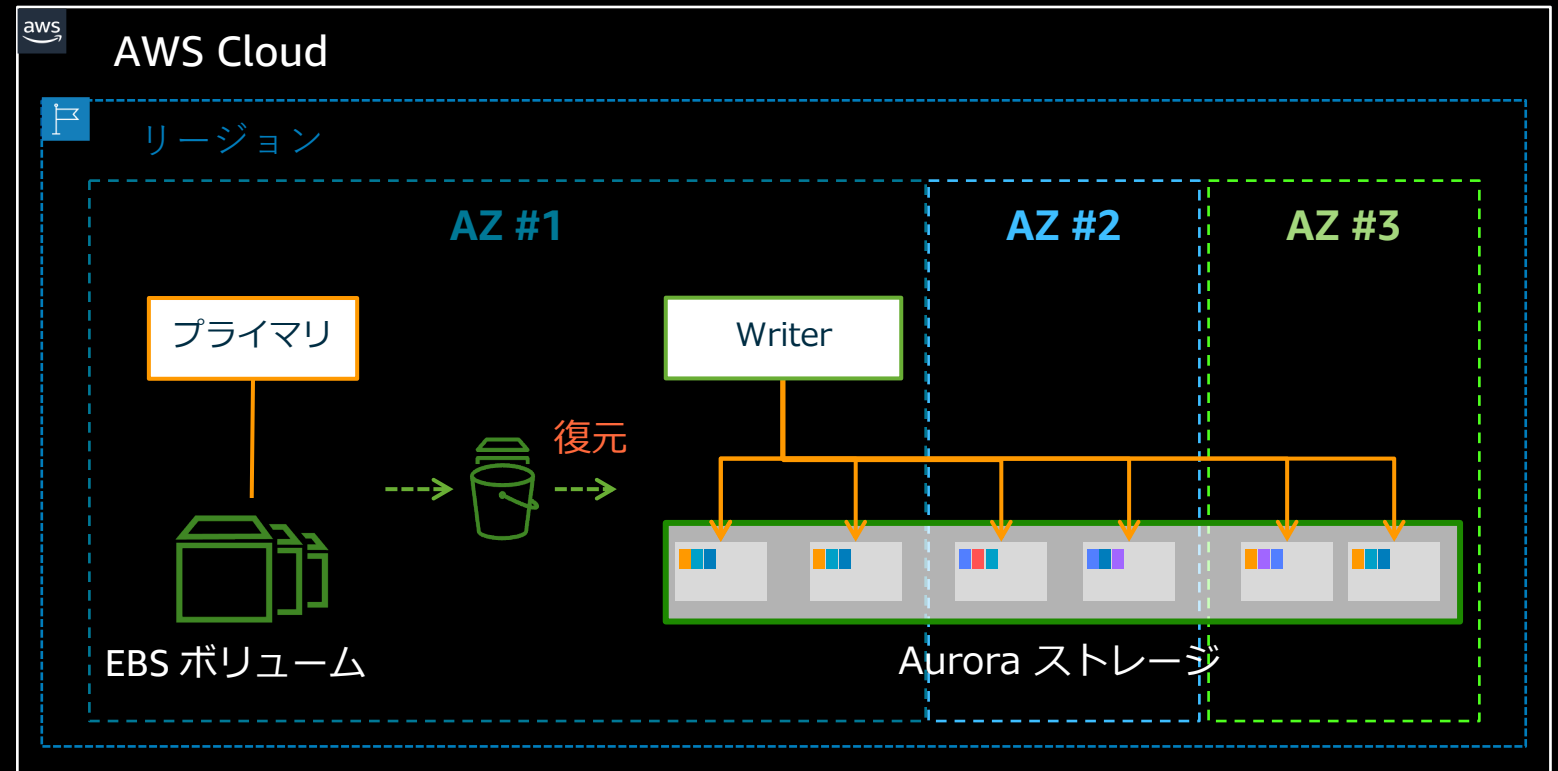
<https://docs.aws.amazon.com/dms/latest/userguide/dm-migrating-data-postgresql.html>

- 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

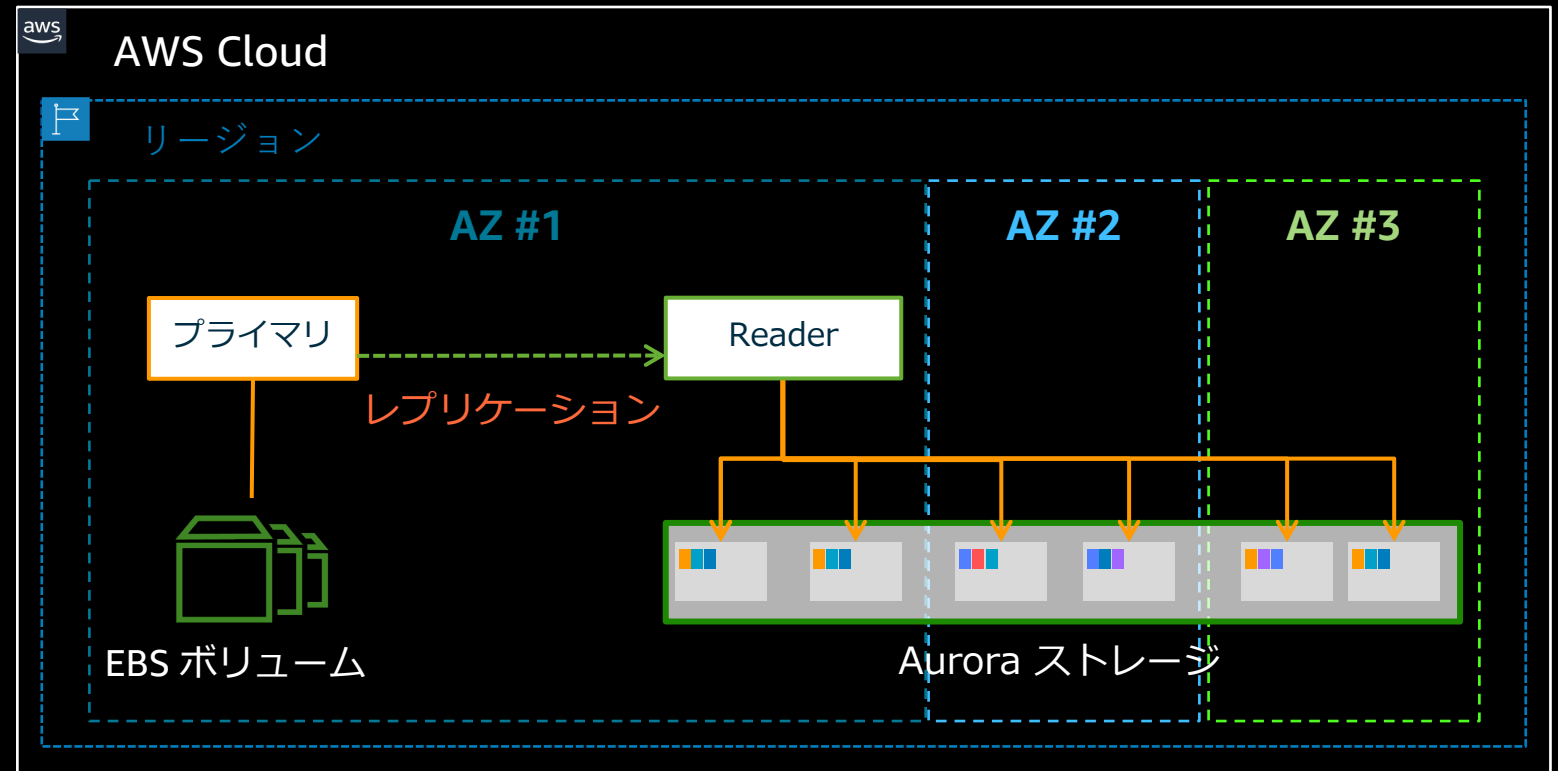
- シンプルに構成可能
- スナップショットから復元するために、一定のダウンタイムを要する



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!

