

---

# AWS の規範的ガイダンス

## リレーショナルデータベースの移行戦略



## AWS の規範的ガイダンス: リレーショナルデータベースの移行戦略

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、顧客に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

## Table of Contents

はじめに .....	1
概要 .....	1
移行の段階 .....	2
段階 1: 準備 .....	3
依存関係を特定する .....	3
ワークロードを適格化する .....	4
段階 2: 計画 .....	5
移行戦略を選択する .....	6
第 3 段階: 移行 .....	7
スキーマを変換する .....	7
AWS SCT .....	7
移行プレイブック .....	8
データを移行する .....	8
AWS DMS .....	9
オフライン移行オプション .....	9
アプリケーションの更新 .....	9
移行をテストする .....	9
カットオーバー .....	10
オフライン移行 .....	10
フラッシュカット移行 .....	10
アクティブ/アクティブデータベース構成 .....	11
増分移行 .....	11
AWS に関するベストプラクティスに従う .....	11
段階 4: 運用と最適化 .....	12
を使用するAWSパートナー .....	13
次のステップ .....	14
Resources .....	15
ドキュメント履歴 .....	16
用語集 .....	17
データベース用語 .....	17
移行の条件 .....	18
.....	xxiv

# リレーショナルデータベースの移行 戦略

Yaser Raja, Amazon Web Services (AWS)

2020 年 11 月([ドキュメント履歴 \(p. 16\)](#))

企業ポートフォリオには、複数のタイプのデータベースがある可能性があります。Amazon Web Services (AWS) の場合、データベースの「リフトアンドシフト」(リホスト)を行うか、次のように切り替えてアプリケーションをモダナイズするかを選択できますAWSマネージドデータベースサービス(リプラットフォーム)。

データベースを再ホストすることを選択した場合、AWSには、データの安全な移動、保存、分析に役立つさまざまなサービスとツールが用意されています。に切り替えることを選択した場合AWSマネージドデータベースサービス、AWSには多数のオプションが用意されているため、機能、パフォーマンス、拡張性をトレードオフする必要はありません。の詳細についてはAWSデータベースファミリ、を参照してください[上のデータベースAWS](#)上にAWSウェブサイト。

このドキュメントでは、リレーショナルデータベースをAWSクラウドは、オンプレミスのデータベースの移行を計画している IT およびプロジェクトマネージャー、製品オーナー、および運用/インフラストラクチャマネージャーを対象としていますAWS。

## 概要

最適なデータベース移行戦略により、次の利点を最大限に活用できます。AWS Cloud。これには、専用のクラウドネイティブデータベースを使用するようにアプリケーションを移行することが含まれます。オンプレミスで使用していたものと同じ古いガードデータベースだけに限定すべきではありません。代わりに、アプリケーションの最新化を検討し、アプリケーションのワークフロー要件に最も適したデータベースを選択してください。

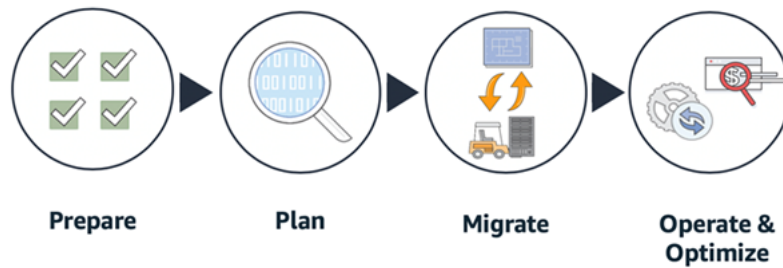
多くの企業がこのアプローチを採用しています。たとえば、Airbnbは毎日50 GBのデータを迅速に処理して分析する必要がありました。彼らは、パーソナライズされた検索を可能にするクイックルックアップのためのユーザー検索履歴を保存するためのキーバリュードータベース、より高速な(ミリ秒未満)サイトレンダリングのためにセッション状態を保存するインメモリデータストア、そして主要なトランザクションデータベースとしてのリレーショナルデータベースを必要としていました。彼らは選んだ[Amazon DynamoDB](#)キーバリュードータベースとして、[アマゾン ElastiCache](#)インメモリストアとして、[Amazon Relational Database Service](#)(Amazon RDS)のトランザクションデータベース。Airbnbの利用の詳細についてはAWSデータベースサービスについては、[を参照してください](#)。 [Airbnbケーススタディ](#)。

データベース移行戦略は、組織の包括的なクラウド戦略と密接に結びついています。たとえば、最初にアプリケーションを移行してから移行することを選択した場合、最初にデータベースをリフトアンドシフトすることに決めるかもしれません。あなたが完全にいるときAWSクラウドでは、アプリケーションの最新化に取り掛かります。この戦略は、現在のデータセンターをすぐに退出し、近代化に集中するのに役立ちます。

データベースの移行は、アプリケーションの移行と密接に結びついています。すべてのデータベース移行戦略には、それらのデータベースを使用するアプリケーションにある程度の変更が含まれます。これらの変更は、データベースの新しい場所を指し示すことまで多岐にわたります。AWSソースコードがないために変更できない場合、またはクローズドソースのサードパーティアプリケーションである場合は、クラウドでアプリケーションを完全に書き直します。

# 移行の段階

移行するデータベースを特定したら、データベースの準備、計画、移行、最適化の段階を経ます。



以下のセクションでは、各段階の詳細について説明します。

- [段階 1: 準備 \(p. 3\)](#)
- [段階 2: 計画 \(p. 5\)](#)
- [第 3 段階: 移行 \(p. 7\)](#)
- [段階 4: 運用と最適化 \(p. 12\)](#)

## 段階 1: 準備

データベース移行プロセスの第 1 段階は準備です。準備中に、アプリケーションとデータベース間の相互依存性を特定します。また、データベースのワークロードを分析して、単純なリホスト (同種) 移行からアーキテクチャの再設計 (異種) 移行まで、移行カテゴリを決定します。この段階を完了しないと、移行のスケジュールが遅れるリスクがあります。

これらのタスクについては、以降のセクションで説明します。

- [依存関係の特定 \(p. 3\)](#)
- [ワークロードの適格 \(p. 4\)](#)

## 依存関係を特定する

まず、次のような質問をして、アプリケーションおよびデータベースの依存関係を特定します。

- このデータベースは他のアプリケーションによって直接アクセスされますか？

される場合、データベースの移行がそのアプリケーションにどのように影響するかを判断する必要があります。データベースをリホストする場合は、アプリケーションが許容可能なパフォーマンスでデータベースにアクセスできることを確認する必要があります。

- アプリケーションは、他のデータベースに直接アクセスしますか？

する場合、他のデータベースの移行プランを決定します。移行もする場合は、それに応じてアプリケーションを更新する必要があります。移行しない場合は、アプリケーションが許容可能なレイテンシーで引き続き接続できることを確認する必要があります。

- データベースはデータベースリンクを使用して他のデータベースからデータを取得していますか？

前のポイントと同様、他のデータベースの移行プランを決定し、それに応じてリンクを処理します。

- アプリケーションはオンプレミスのソフトウェアに依存していますか？

している場合、そのソフトウェアの移行プランを決定する必要があります。移行する場合は、それに応じてアプリケーションを更新する必要があります。移行しない場合は、アプリケーションが引き続き接続でき、レイテンシーが許容可能であることを確認します。

- ハードウェアの依存関係はありますか？

ある場合、それらに対処する計画を考え出してください。

- 厳密な帯域幅やネットワーク要件はありますか？

その場合は、これらの要件を満たすのに役立つ AWS のサービスを選択します。

- アプリケーションでは、特別なデータベースエンジンのオプションや機能を使用していますか？

別のデータベースエンジンに移行する場合は、それに応じてアプリケーションを更新する必要があります。

これらの質問に対する回答が複雑な場合は、マイクロサービスを使用してデータベースをアプリケーションから疎結合化の方がよいでしょう。このようにして、アプリケーションはデータベースに直接接続するのではなく、マイクロサービスを呼び出すことによってデータを取得できます。

## ワークロードを適格化する

データベースに最適な移行戦略を決定するには、現在のデータベースのワークロードを理解することが重要です。データベースを分析して、現在使用している機能と、[Amazon Aurora PostgreSQL](#) のような別のクラウドネイティブデータベースエンジンへの移行に関係しているものを特定する必要があります。

AWS は、AWS ワークロード資格フレームワーク (AWSWQF) というワークロード認定ツールを提供します。このツールは、データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、パフォーマンス特性、および類似の入力を分析することで、Oracle および Microsoft SQL Server データベース移行の複雑さを特定するのに役立ちます。WQF は、ターゲットデータベースエンジンに関するレコメンデーションを提供します。また、関連する作業の種類と必要な作業のレベルも予測します。

WQF は移行ワークロードを評価し、次の表に示す 5 つのワークロードカテゴリのいずれかに配置します。

Category 1	ODBC/JDBC workloads	< 50 manual changes, easy to refactor
Category 2	Light, proprietary feature workloads	< 200 manual changes, medium complexity
Category 3	Heavy, proprietary feature workloads	> 200 manual changes, high complexity
Category 4	Engine-specific workloads	Not recommended for refactoring
Category 5	COTS or other non-portable workloads	Not recommended for refactoring

- カテゴリ 1: データベースに接続するワークロード独自のドライバーではなく、Open Database Connectivity (ODBC) または Java Database Connectivity (JDBC) を使用するワークロード。このカテゴリには通常、アクセス制御に使用される単純なストアードプロシージャがあります。変換に必要な手動変更は 50 件未満です。
- カテゴリ 2: 独自の機能を多少使用し、高度な SQL 言語機能を使用しないワークロード。このタイプのワークロードでは、手動で変更する必要が 200 未満です。
- カテゴリ 3: 独自の機能を多用するワークロード。このカテゴリの作業負荷は、高度なストアードプロシージャロジックまたは独自の機能によって完全に左右されます。このタイプのワークロードでは、データベース常駐のコードと機能を含む 200 件を超える手動による変更が必要です。
- カテゴリ 4: エンジン固有のワークロード。このカテゴリのワークロードは、特定の商用データベースエンジンでしか機能しないフレームワークを使用します。例えば、これらのフレームワークには、Oracle Forms、Oracle Reports、Oracle Application Development Framework (ADF)、Oracle Application Express (APEX) や、.NET ActiveRecord を広範囲で使用するアプリがあります。
- カテゴリ 5: 移動できない、許容できないリスク、または「リフトアンドシフト」ワークロード。このカテゴリのワークロードは、クラウドベース同等の機能を持たないデータベースエンジンに実装される可能性があります。お客様がこれらのプログラムのソースコードを持っていない場合もあります。

この分類は、「[第 2 段階: 計画 \(p. 5\)](#)」セクションで説明するように、アプリケーションの移行パスを決定するのに役立ちます。

AWS は、現在ダウンロードのための AWS WQF を提供していません。AWS WQF で AWS への移行の評価をサポートする必要がある場合、サポートチケットを使用することをお勧めします。AWS は、お客様と直接関わって、お客様のプロセスをお手伝いいたします。

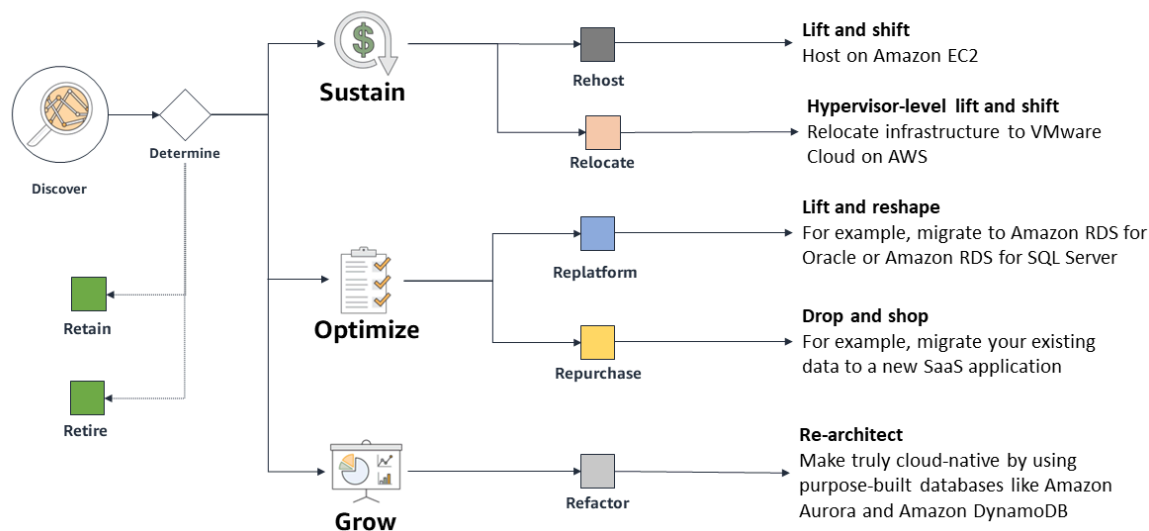
## 段階 2: 計画

この段階では、準備段階で収集した情報を使用して、移行戦略を考えます。移行プランの重要な側面は、リホスト、再プラットフォーム化、再配置、再購入、リファクタリング、使用停止、および保持という 7R の移行に対して収集した情報を合理化することです。

移行戦略の選択は、クラウド導入のビジネス原動力や、時間の考慮事項、ビジネスおよび財務上の制約、リソース要件によって異なります。クラウド内の現在のワークロードを維持する場合は、[rehosting] (リホスト) を選択します。ただし、ワークロードを最適化してスケーリングする場合は、他のオプションから 1 つを検討してください。

ここでは、7R のデータベース移行の概要を示します。これらを次の図で説明します。

### Migration Paths



- リホスト (リフトアンドシフト) — 変更を加えずに、アプリケーションをクラウドに移行します。例えば、お客様のオンプレミスの Oracle データベースを [クラウドの Amazon Elastic Compute Cloud \(Amazon EC2\)](#) インスタンス上の Oracle に移行します。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) — 新しいハードウェアの購入、アプリケーションの書き換え、既存のオペレーションの変更を行うことなく、インフラストラクチャをクラウドに移行できます。この移行シナリオは AWS の VMware Cloud に固有のもので、お客様のオンプレミス環境と、および AWS の間の仮想マシン (VM) 互換性とワークロードの移植性をサポートします。インフラストラクチャを AWS の VMware Cloud に移行するときに、オンプレミスのデータセンターから VMware Cloud Foundation テクノロジーを使用できます。例えば、AWS の Oracle データベースをホストしているハイパーバイザーを VMware Cloud 上に再配置します。
- 再プラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例えば、お客様のオンプレミスの Oracle データベースを AWS クラウドの [Amazon RDS for Oracle](#) に移行します。
- 再購入 (ドロップアンドショップ) — 通常、従来のアプリケーションから SaaS (Software-as-a-Service) 製品に移行し、オンプレミスアプリケーションから新製品にデータを移行して、別の製品に変更します。例えば、顧客データをオンプレミスの顧客関係管理 (CRM) システムから Salesforce.com に移行します。
- リファクタリング (アーキテクチャの再設計) — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更








します。例えば、オンプレミスの Oracle データベースを [Aurora PostgreSQL](#) へ移行します。この戦略には、AWS がさまざまなワークフローを提供する目的別データベースを使用するようにアプリケーションを書き直すことも含まれます。または、独自のデータベーススキーマにアクセスする小さなマイクロサービスに分割して、モノリシックアプリケーションをモダナイズすることもできます。

- 保持 (再アクセス) — アプリケーションをソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。
- 使用停止 — ソース環境で不要になったアプリケーションを停止または削除します。

## 移行戦略を選択する

ほとんどのデータベース移行では、リホスト、再プラットフォーム化、またはリファクタリングを選択できます。これらの戦略はどれも役に立ちます。ガイドの原則は、移行から最大限の利点を引き出す方法であるべきです。アプリケーションをリファクタリングして Aurora などのクラウドネイティブデータベースに移行することで、データベースアプリケーションを強化できます。ただし、ワークロードの複雑さによっては、データベースのリファクタリングに時間がかかり、リソースを大量に消費することがあります。

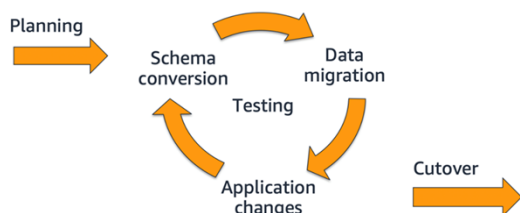
WQF の分類は、特定の移行戦略を検討すべきタイミングを決定するのに役立ちます。WQF カテゴリが高いほど、必要な移行作業が重要であることを意味します。したがって、リホストや再プラットフォーム化などの別のオプションを選択して、許容可能な期間内に移行を完了することをお勧めします。次の表は、WQF カテゴリに基づく推奨戦略を示しています。

カテゴリ	ワークロードの複雑さ	ワークロード	移行戦略
1		ODBC/JDBC ワークロード	リファクタリング候補
2		軽い、独自の機能ワークロード	リファクタリング候補
3		重い、独自の機能ワークロード	リファクタリングまたは再プラットフォーム化の候補
4		エンジン固有のワークロード	再プラットフォームまたはリホスト候補
5		非ポータブル、ハイリスク、またはリフトアンドシフトワークロード	再プラットフォームまたはリホスト候補

リファクタリングに伴う複雑さが高い場合は、リホストおよび再プラットフォーム化オプションが適しています。これらのシナリオでは、モダナイゼーションのニーズに基づいて、AWS クラウドへの移行を完了した後データベースのリファクタリングを検討できます。

## 第 3 段階: 移行

移行プランを完了し、移行戦略を特定すると、実際の移行が実行されます。この段階では、ターゲットデータベースが設計され、ソースデータがターゲットに移行され、データが検証されます。



これは、変換、移行、およびテストの複数のサイクルを含む反復プロセスです。機能テストとパフォーマンステストが完了したら、新しいデータベースに切り取ることができます。

移行段階は次の重要なステップで構成されています。詳細については、次のセクションで説明します。

- [スキーマの変換 \(p. 7\)](#)
- [データの移行 \(p. 8\)](#)
- [アプリケーションの更新 \(p. 9\)](#)
- [移行のテスト \(p. 9\)](#)
- [新しいデータベースへのカットオーバー \(p. 10\)](#)

## スキーマを変換する

データベース移行中の重要なタスクの 1 つは、スキーマをソースデータベースエンジンからターゲットデータベースエンジンに移行することです。リホストまたは再プラットフォーム化しても、データベースエンジンは変更されません。これは、同種のデータベースの移行と呼ばれ、ネイティブデータベースツールを使用してスキーマを移行できます。

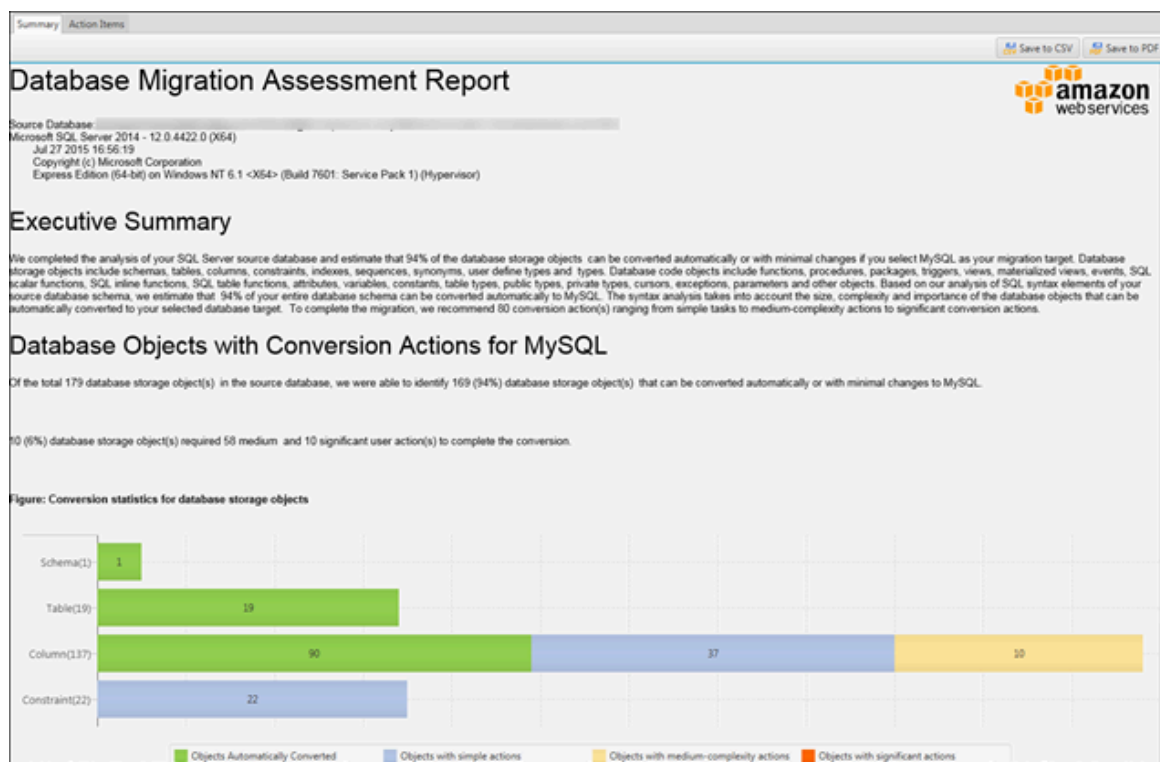
ただし、アプリケーションのアーキテクチャを再構築する場合は、スキーマ変換により多くの労力が必要になる場合があります。この場合は、異種混在データベースの移行で、ソースとターゲットのデータベースエンジンが異なる場所になります。現在のデータベーススキーマが、ターゲットデータベースエンジンに直接変換できないパッケージと機能を使用している可能性があります。一部の機能は、別の名前で利用できる場合があります。したがって、スキーマを変換するには、ソースとターゲットのデータベースエンジンを十分に理解する必要があります。現在のスキーマの複雑さによっては、このタスクは難しい場合があります。

AWS には、スキーマの変換に役立つ AWS Schema Conversion Tool (AWS SCT) と移行プレイブックの 2 つのリソースがあります。

## AWS SCT

AWS SCT は、既存のデータベースをあるエンジンから別のエンジンに変換するのに役立つ無料のツールです。AWS SCT は、Oracle、Microsoft SQL Server、MySQL、Sybase、IBM Db2 LUW など、多数のソースデータベースをサポートしています。Aurora MySQL や Aurora PostgreSQL などのターゲットデータベースから選択できます。

AWS SCT は、ソースデータベースとターゲットデータベースに直接接続して現在のスキーマオブジェクトを取得するグラフィカルユーザーインターフェースを提供します。接続すると、データベース移行評価レポートを生成して、変換作業とアクション項目の高レベルな概要を把握できます。次の図は、サンプルのデータベース移行評価レポートを示しています。



AWS SCT を使用すると、スキーマを変換してターゲットデータベースに直接デプロイするか、変換されたスキーマの SQL ファイルを取得できます。詳細については、AWS ドキュメントの「[AWS Schema Conversion Tool ユーザーインターフェースの使用](#)」を参照してください。

## 移行プレイブック

AWS SCT は、多くのソースオブジェクトを変換しますが、変換のいくつかの側面では、手動による操作と調整が必要です。このタスクを支援するために、AWS は、2 つのデータベース間の非互換性と類似性を詳細に説明した移行プレイブックを提供します。これらのプレイブックの詳細については、AWS ウェブサイトの「[AWS Database Migration Service リソース](#)」を参照してください。

## データを移行する

スキーマの移行が完了したら、データをソースデータベースからターゲットデータベースに移動できます。アプリケーションの可用性の要件に応じて、新しいデータベースへのソースデータの 1 回限りのコピーを実行する単純な抽出ジョブを実行できます。または、現在のデータをコピーし、新しいデータベースにカットオーバーする準備ができるまで、すべての変更をレプリケートするツールを使用することもできます。リホストおよび再プラットフォーム移行の場合、ネイティブのデータベース固有のツールを使用してデータを移行することをお勧めします。

データ転送に役立つツールには、次のようなものがあります。AWS Database Migration Service (AWS DMS)、オフライン移行ツールを使用できます。次のセクションにこれらのオプションの説明を示します。

## AWS DMS

AWS SCT を使用してスキーマオブジェクトをソースデータベースエンジンからターゲットエンジンに変換した後、AWS DMS を使用して、データを移行します。AWS DMS を使用すると、データがレプリケートされている間は、ソースデータベースを稼働させ続けることができます。データの 1 回限りのコピーを実行することも、連続レプリケーションを使用してコピーすることもできます。ソースデータベースとターゲットデータベースが同期されている場合、データベースをオフラインにして、操作をターゲットデータベースに移動できます。AWS DMS は、同種データベース移行c (オンプレミスの Oracle データベースから Amazon RDS for Oracle データベースへなど) および異種混在移行 (オンプレミスの Oracle データベースから Amazon RDS for PostgreSQL データベースへなど) に使用できます。AWS DMS の詳細な操作方法については、「[AWS DMS ドキュメント](#)」を参照してください。

## オフライン移行オプション

また、AWS DMS に加え、その他のオプションを使用して、ソースデータベースからデータを抽出し、ターゲットデータベースにロードできます。これらのオプションは、データ移行アクティビティ中にアプリケーションのダウンタイムが許容される場合に適しています。これらの方法の例には以下があります。

- ターゲットデータベースにロードされたソースデータベースからカンマ区切り値 (CSV) を抽出します。
- Oracle ソースデータベースの場合、PostgreSQL にデータをコピーする ora2pg ユーティリティ
- ソースからターゲットにデータをコピーするためのカスタム抽出、変換、ロード (ETL) ジョブ

## アプリケーションの更新

データベースの移行は、データベースのみの移行ではありません。データベースを使用しているアプリケーションを調べて、新しいデータベースで期待どおりに動作することを確認する必要があります。同じデータベースエンジンを再ホストまたは再プラットフォームするだけであれば、変更は最小限に抑えられますが、新しいデータベースエンジンに移行する場合には、より重要になります。

アプリケーションがデータベースと対話するためにオブジェクトリレーショナルマッピング (ORM) に依存している場合、新しいデータベースエンジンに移行する際に多くの変更を必要としません。ただし、アプリケーションにカスタムデータベースインタラクションまたは動的に構築された SQL クエリがある場合、変更は大きくなる可能性があります。アプリケーションが期待どおりに動作するように修正する必要があるクエリ形式に差異がある場合があります。

例えば、Oracle では、文字列を NULL に結びつけると元の文字列を返します。ただし、PostgreSQL では、文字列を NULL と結びつけると NULL を返します。もう 1 つ例は、NULL と空の文字列がどのように処理されるかを示しています。PostgreSQL では、NULL と空の文字列は 2 つの異なるものですが、Oracle のようなデータベースはこれらと同じ方法で扱います。Oracle では、列の値が NULL または空の文字列に設定されている行を挿入した場合、where 句 `where <mycolumn> is NULL` を使用して、両方のタイプの値を取得できます。PostgreSQL では、この where 句は、カラム値が実際に NULL である行を 1 つだけ返します。空の文字列値を持つ行は返されません。これらの差異の詳細については、[AWS Database Migration Service リソース](#) ウェブサイトに一覧表示されている移行プレイブックを参照してください。

## 移行をテストする

機能テストとパフォーマンステストは、データベースの移行に不可欠なパートです。詳細な機能テストにより、アプリケーションが問題なく新しいデータベースで動作していることを確認します。アプリケーションのワークフローをテストするためのユニットテストの開発に時間を費やす必要があります。

パフォーマンステストにより、データベースの応答時間が許容可能な時間範囲内であることを確認します。ボトルネックを特定し、最適化し、パフォーマンステストを繰り返すことができます。必要に応じてこのサイクルを繰り返して、目的のパフォーマンス結果が得られます。

テストは手動でも自動でもかまいません。テストには自動フレームワークを使用することをお勧めします。移行中は、テストを複数回実行する必要があります。そのため、自動テストフレームワークを使用すると、バグの修正と最適化のサイクルを高速化できます。

このテストでは、開発段階中に欠落した問題を明らかにできます。例えば、誤って変換されたクエリが失敗したり、誤った結果を返したりして、機能テストが失敗します。パフォーマンステストでは、インデックスの欠落によるクエリの応答時間が遅くなるなどの問題が明らかになります。また、ワークロードやクエリの変更に応じて、データベースエンジンのチューニングが必要なパフォーマンスの問題を明らかにすることもできます。

## カットオーバー

データベースのカットオーバー戦略は、通常、アプリケーションのダウンタイム要件と密接に結びついています。データベースカットオーバーに使用できる戦略には、オフライン移行、フラッシュカット移行、アクティブ/アクティブデータベース構成、増分移行などがあります。これらについては、以降のセクションで説明します。

### オフライン移行

書き込み操作中にアプリケーションを長期間オフラインにできる場合は、データ移行には AWS DMS フルロードタスク設定またはオフライン移行オプションの 1 つを使用できます。読み取りトラフィックは、この移行の進行中も継続できますが、書き込みトラフィックを停止する必要があります。すべてのデータをソースデータベースからコピーする必要があるため、I/O や CPU などのソースデータベースリソースが使用されます。

高いレベルで、オフライン移行には次のステップが含まれます。

1. スキーマの変換を完了します。
2. 書き込みトラフィックのダウンタイムを開始します。
3. オフライン移行オプションのいずれかを使用してデータを移行します。
4. データを検証します。
5. アプリケーションに新しいデータベースを参照させます。
6. アプリケーションのダウンタイムを終了します。

### フラッシュカット移行

フラッシュカット移行では、ダウンタイムを最小限に抑えることが主な目的です。この方法では、ソースデータベースからターゲットデータベースへの連続データレプリケーション (CDC) を使用します。データの移行中は、すべての読み取り/書き込みトラフィックが現在のデータベースで継続されます。すべてのデータをソースデータベースからコピーする必要があるため、I/O や CPU などのソースサーバーリソースが使用されます。このデータ移行アクティビティがアプリケーションのパフォーマンスの SLA に影響を与えないことをテストする必要があります。

高いレベルで、フラッシュカット移行には次のステップが含まれます。

1. スキーマの変換を完了します。
2. 連続データレプリケーションモードで AWS DMS をセットアップします。
3. ソースデータベースとターゲットデータベースが同期されている場合は、データを検証します。
4. アプリケーションのダウンタイムをスタートします。
5. 新しいデータベースを参照するアプリケーションの新しいバージョンをロールアウトします。
6. アプリケーションのダウンタイムを終了します。

## アクティブ/アクティブデータベース構成

アクティブ/アクティブデータベース構成では、両方のデータベースが書き込みトラフィックに使用されている間、ソースデータベースとターゲットデータベースを同期させるメカニズムを設定する必要があります。この戦略では、オフラインまたはフラッシュカット移行よりも多くの作業が必要になりますが、移行時の柔軟性も向上します。例えば、移行中のダウンタイムが最小限に抑えられるだけでなく、1 回限りのカットオーバーを実行するのではなく、小規模な制御バッチで本番トラフィックを新しいデータベースに移動できます。デュアル書き込み演算を実行して両方のデータベースに変更を加えるか、[HVR](#) のような双方向レプリケーションツールを使用して、データベースを同期できます。この戦略では、[セットアップ](#)とメンテナンスの面で複雑性が高まるため、データ整合性の問題を回避するには、より多くのテストが必要です。

高いレベルで、アクティブ/アクティブデータベースの設定には次のステップが含まれます。

1. スキーマの変換を完了します。
2. 既存のデータをソースデータベースからターゲットデータベースにコピーし、双方向のレプリケーションツールまたはアプリケーションからのデュアル書き込みを使用して、2 つのデータベースの同期を維持します。
3. ソースデータベースとターゲットデータベースが同期されている場合は、データを検証します。
4. トラフィックのサブセットを新しいデータベースに移動し始めます。
5. すべてのデータベーストラフィックが新しいデータベースに移動されるまで、トラフィックを移動し続けます。

## 増分移行

増分移行では、1 回限りのフルカットオーバーを実行するのではなく、小さなまとまりでアプリケーションを移行します。このカットオーバー戦略は、現在のアプリケーションアーキテクチャやアプリケーションで実行するリファクタリングに基づいて、さまざまなバリエーションを持つことができます。

[設計パターン](#)を使用して、新しい独立したマイクロサービスを追加して、既存のモノリシックレガシーアプリケーションの一部を置き換えます。これらの独立したマイクロサービスには、アプリケーションの他の部分によって共有またはアクセスされない独自のテーブルがあります。これらのマイクロサービスは、他のカットオーバー戦略を使用して、新しいデータベースに 1 つずつ移行します。移行されたマイクロサービスは、新しいデータベースを読み取り/書き込みトラフィックに使用し始め、アプリケーションの他のすべての部分は引き続き古いデータベースを使用します。すべてのマイクロサービスが移行されたら、レガシーアプリケーションを廃止します。この戦略は、移行をより小さく管理しやすい単位に分割し、1 つの大きな移行に関連するリスクを軽減できます。

## AWS に関するベストプラクティスに従う

前のセクションで説明した移行アクティビティに加えて、時間をかけて、AWS クラウドのデータベースをホストするためのベストプラクティスに従っていることを確認する必要があります。AWS のリレーショナルデータベースの使用のベストプラクティスについては、「[AWS ドキュメント](#)」を参照してください。

## 段階 4: 運用と最適化

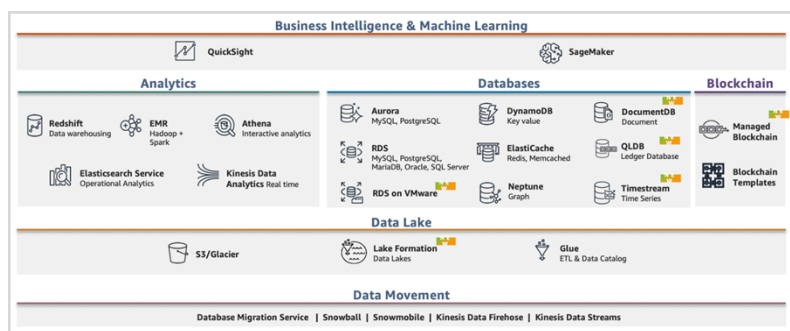
データベースが AWS にある場合、クラウドで運用する必要があります。モニタリング、アラート、バックアップ、高可用性などの分野のベストプラクティスに従っていることを確認する必要がある場合があります。リホストされたデータベースのオペレーションオーバーヘッドは、マネージド AWS データベースサービスを使用するために再プラットフォームまたはリファクタリングされたデータベースよりも高くなります。

- リホストされたデータベースは EC2 インスタンスで実行されます。バックアップ、高可用性、災害対策ソリューションの設定など、すべてのデータベース管理タスクをお客様が担当します。
- Amazon RDS でデータベースを再プラットフォーム化またはリファクタリングする場合、これらのデータベース管理タスクの設定に必要なのは数回のクリックのみです。つまり、EC2 インスタンスでリホストされたデータベースを管理する場合と比較して、データベース管理者は Amazon RDS でのデータベースの管理に費やす時間が短縮されます。Amazon RDS には、Amazon RDS Performance Insights というパフォーマンスモニタリングツールも提供しています。これにより、専門家でなくても、データベースの負荷を視覚化するわかりやすいダッシュボードを使用して、パフォーマンスの問題を検出できます。

どの移行オプションを選択しても、Amazon CloudWatch は CPU、メモリ、I/O 使用率などの主要なメトリクスを収集する上で非常に重要な役割を果たします。また、メトリクスにしきい値を設定し、指定されたしきい値を超えたときにアクションを開始する機能も提供します。例えば、Aurora PostgreSQL クラスターメトリクスでアラームを作成し、通知を設定し、未使用または使用率の低いリーダーインスタンスを検出してシャットダウンするアクションを実行できます。メトリクスとイベントにリアルタイムアラームを設定することで、ダウンタイムと潜在的なビジネスへの影響を最小限に抑えることができます。

運用と最適化段階では、AWS でアプリケーションをホストすることから得られる利点を最大化できます。最適化アクティビティは、アプリケーションスタックのコスト、パフォーマンス、セキュリティ、または回復力の懸念に対処できます。例えば、オートスケーリング機能を使用して、ピーク時にリードレプリカを追加し、オフピーク時にリードレプリカを削除してコストを削減できます。Amazon RDS データベースとシームレスに統合される多くの AWS のサービスを使用することもできます。例えば、分析のためにデータベースエンジンログを Amazon CloudWatch Logs に簡単に送信することができます。

AWS クラウドに移行すると、数回のクリックでスピンアップできる多数のサービスと機能を利用して、アプリケーションの最適化を開始できます。インフラストラクチャやデータセンターの管理の無差別化に重点を置くことなく、ビジネスを差別化し、ユーザーエクスペリエンスを変革するアプリケーションを開発する非常に価値の高い IT リソースに集中できるため、イノベーションを迅速に行うことができます。次の図は、AWS のサービスで提供されるオプションのいくつかを示しています。



さらに、数分でグローバルデプロイが可能です。例えば、数回のクリックで [Amazon Aurora Global Database](#) を作成できます。これにより、世界中のデータベース読み取り操作を簡単に拡張し、アプリケーションをユーザーの近くに配置できます。

同様に、統合を使用して、データからより多くの価値を引き出すことができます。例えば、[Aurora データベースアプリケーションでの機械学習 \(ML\) 機能](#) は簡単な数ステップで使用できます。

## を使用するAWSパートナー

データベースの移行は、専門知識とツールを必要とする困難なプロジェクトになる可能性があります。パートナーシップにより、移行期間を短縮し、結果を出すまでの時間を短縮できます。[AWS Database Migration Serviceパートナー](#)お客様がクラウドに簡単かつ安全に移行できるようにするために必要な専門知識を持っている。これらのパートナーは、OracleからOracleへの同種移行と、OracleからAmazon Aurora またはMicrosoft SQL ServerからMySQLなどの異なるデータベースプラットフォーム間の異種移行の両方に関する専門知識を持っています。

要件や好みに応じて、パートナーを利用して移行全体を処理したり、移行の一部のみを支援したりすることができます。さらに、次のツールやソリューションも使用できます。AWS移行を支援するパートナー。



## 次のステップ

Oracle Database および SQL Server ワークロードの移行の詳細については、AWS 規範的ガイダンスウェブサイトの次のガイドを参照してください。

- [Oracle データベースを AWS クラウドへ移行する](#)
- [SQL Server データベースを AWS クラウドへ移行する](#)

特定のリレーショナルデータベースを移行するためのステップバイステップの手順については、「[database migration patterns](#)」(データベースの移行パターン)を参照してください。そのページのフィルターを使用して、AWS のサービスごと (Aurora への移行など)、ワークロードごと (Oracle データベースの移行など)、計画的な使用ごと (本番または試行)、または移行戦略ごと (再構築、リホスト、再配置、または再プラットフォーム化) ごとのサービスのパターンを表示できます。

# Resources

- [Oracle データベースを AWS クラウドへ移行する](#)
- [SQL Server データベースを AWS クラウドへ移行する](#)
- [AWS DMS ドキュメント](#)
- [AWS SCT ドキュメント](#)
- [移行プレイブック](#)
- [AWS データベースのオプション](#)
- AWS マネージドデータベースサービスに関する一般的な情報:
  - [Amazon RDS](#)
  - [Amazon Aurora](#)
  - [Amazon RDS for MySQL](#)
  - [Amazon RDS for Oracle](#)
  - [Amazon RDS for PostgreSQL](#)
  - [Amazon RDS for SQL Server](#)
- [Amazon RDS ドキュメント](#)

# ドキュメント履歴

このガイドは、このドキュメントの大きな変更点をまとめたものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#)をサブスクライブできます。

変更	説明	日付
<a href="#">UpdatedAWSWQF 情報 (p. 16)</a>	更新しました「 <a href="#">ワークロードの検定</a> 」セクションについての最新情報付きAWSWQF。	2020 年 11 月 5 日
<a href="#">初版発行 (p. 16)</a>	—	2019 年 12 月 15 日

# AWS 規範的ガイダンスの用語集

以下は、AWS規範的ガイダンスで提供される戦略、ガイド、およびパターンでよく使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

## データベース用語

原子性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用の信頼性を保証するソフトウェアプロパティのセット。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「[エンディアン \(p. 17\)](#)」も参照してください。

ブルームフィルター

要素がセットのメンバーかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造です。

バッファキャッシュ

最も頻繁にアクセスされるデータが格納されるメモリ領域。

変更データキャプチャ (CDC)

データベーステーブルなどのデータソースへの変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムの変更を監査または複製して同期を維持するなど、さまざまな目的に使用できます。

コールドキャッシュ

空のバッファキャッシュ、データが正しくない、または古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必要があり、バッファキャッシュからの読み取りよりも時間がかかるため、これはパフォーマンスに影響します。

構成管理データベース (CMDB)

ハードウェアとソフトウェアの両方のコンポーネントとその構成を含む、データベースとそのIT環境に関する情報を保存および管理するリポジトリです。

データベース定義言語 (DDL)

データベース内のテーブルとオブジェクトの構造を作成または変更するためのステートメントまたはコマンド。

データベース操作言語 (DML)

データベース内の情報を変更 (挿入、更新、削除) するためのステートメントまたはコマンド。

エンディアン

コンピューターのメモリにバイトが格納される順序。ビッグエンディアンのシステムでは、最上位バイトが最初に格納されます。リトルエンディアンのシステムでは、最下位バイトが最初に格納されません。

### リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「[エンディアン \(p. 17\)](#)」も参照してください。

### クエリプランのリグレッション

データベースサービスオプティマイザーが、データベース環境が変更される前よりも最適化されていないプランを選択した場合。これは、統計、制約、環境設定、クエリパラメータバインディング、およびデータベースエンジンの更新の変更が原因である可能性があります。

### リードレプリカ

読み取り専用で使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースの負荷を軽減できます。

### バキューム中

インクリメンタルアップデート後にクリーンアップしてストレージを再利用し、パフォーマンスを向上させるデータベースメンテナンス操作です。

### ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから読み取ることができ、メインメモリやディスクから読み取るよりも高速です。

## 移行の条件

### 7 Rs

アプリケーションをクラウドに移行するための 7 つの一般的な移行戦略。これらの戦略は、ガートナーが 2011 年に特定した 5 Rs に基づいて構築され、以下で構成されています。

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行する。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの Oracle 用の Amazon Relational Database Service (Amazon RDS) に移行する。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行する。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの EC2 インスタンス上の Oracle に移行する。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) — 新しいハードウェアの購入、アプリケーションの書き換え、お客様の既存のオペレーションの変更を行うことなく、インフラストラクチャをクラウドに移行できます。この移行シナリオは AWS の VMware Cloud に固有のもので、お客様のオンプレミス環境と、および AWS の間の仮想マシン (VM) 互換性とワークロードの移植性をサポートします。インフラストラクチャを AWS の VMware Cloud に移行するときに、お客様のオンプレミスのデータセンターから VMware Cloud Foundation テクノロジーを使用できます。例: AWS の Oracle データベースをホストしているハイパーバイザーを VMware Cloud 上に再配置する。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを行き移るためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。

- 使用停止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

#### アクティブ-アクティブマイグレーション

ソースデータベースとターゲットデータベースを (双方向レプリケーションツールまたは二重書き込み操作を使用して) 同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法です。この方法では、1 回限りのカットオーバーを必要とせず、小規模で管理されたバッチでの移行が可能です。より柔軟性がありますが、より多くの作業が必要です [アクティブ/パッシブ移行 \(p. 19\)](#)。

#### アクティブ/パッシブ移行

ソースデータベースとターゲットデータベースは同期を保ちながら、データがターゲットデータベースに複製される間、接続アプリケーションからのトランザクションはソースデータベースのみが処理するデータベース移行方法です。ターゲットデータベースは、移行中のトランザクションを一切受け付けません。

#### アプリケーションポートフォリオ

アプリケーションの構築と維持にかかるコスト、およびそのビジネス価値を含む、組織が使用する各アプリケーションに関する詳細情報の集まり。この情報は、[ポートフォリオの検出と分析プロセス](#) の需要要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

#### AI オペレーション (AIOps)

機械学習技術を使用して運用上の問題を解決し、運用上のインシデントと人の介入を減らし、サービス品質を向上させるプロセス。AWS 移行戦略での AIOps の使用方法については、[オペレーション統合ガイド](#) を参照してください。

#### AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドに正常に移行するための効率的で効果的な計画を立てるのを支援する AWS からのガイドラインとベストプラクティスのフレームワーク。AWSCAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用の観点と呼ばれる 6 つの重点を置く分野にガイダンスを編成しています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、AWS CAF は、クラウドの導入を成功させるための組織の準備を支援するために、人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、[AWS CAF ウェブサイト](#) と [AWS CAF のホワイトペーパー](#) を参照してください。

#### AWS ワークロード資格フレームワーク (AWS WQF)

データベース移行ワークロードを評価し、移行戦略を推奨し、作業の見積もりを提供するツール。AWSWQF は AWS Schema Conversion Tool (AWS SCT) と共に含まれます。データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、およびパフォーマンス特性を分析し、評価レポートを提供します。

#### ビジネス継続性計画 (BCP)

大規模移行など、中断を伴うイベントが運用に与える潜在的な影響に対処し、ビジネスを迅速に再開できるようにする計画。

#### 変更データキャプチャ (CDC)

データベーステーブルなどのデータソースへの変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムの変更を監査または複製して同期を維持するなど、さまざまな目的に使用できます。

#### Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、AWS クラウドエンタープライズ戦略ブログの [CCoE の投稿](#) を参照してください。

## 導入のクラウドステージ

組織が、AWS クラウドへの移行時に通常実行する 4 つの段階。

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、CCoE の定義、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

これらのステージは Stephen Orban が AWS クラウドエンタープライズ戦略ブログの [クラウドファーストジャーニーと導入ステージ](#) というブログ記事で定義したものです。これらが、AWS 移行戦略とどのような関係があるかについては、[移行準備ガイド](#)を参照してください。

## 構成管理データベース (CMDB)

ハードウェアとソフトウェアの両方のコンポーネントとその構成を含む、データベースとそのIT環境に関する情報を保存および管理するリポジトリです。通常、CMDB のデータは、移行のポートフォリオの検出と分析の段階で使用します。

## エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、AWS CAF セキュリティエピックには、アイデンティティとアクセスの管理、検出型制御、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。AWS 移行戦略のエピックの詳細については、[プログラム実装ガイド](#)を参照してください。

## フラッシュカット移行

段階的なアプローチではなく、[変更データキャプチャ \(CDC\) \(p. 19\)](#) 継続的なデータ複製を使用して可能な限り短時間でデータを移行するデータベース移行方法。目的は、ダウンタイムを最小限に抑えることです。

## 異種混在データベースの移行

別のデータベースエンジンを使用するターゲットデータベースへお客様の出典データベースの移行 (例えば、Oracle から Amazon Aurora)。異種間移行は通常、アーキテクチャの再設計作業の一部であり、スキーマの変換は複雑なタスクになる可能性があります。[AWS は、スキーマの変換に役立つ AWS SCT を提供します](#)。

## 同種データベースの移行

お客様の出典データベースを、同じデータベースエンジンを共有するターゲットデータベース (Microsoft SQL Server から Amazon RDS for SQL Server など) に移行する。同種間移行は、通常、リホストまたはリプラットフォーム化の作業の一部です。ネイティブデータベースユーティリティを使用して、スキーマを移行できます。

## ハイパーケア期間

カットオーバー直後、移行チームが問題に対処するために移行されたアプリケーションをクラウドで管理および監視する期間。通常、この期間は1〜4日です。ハイパーケア期間の終わりに、移行チームは通常、アプリケーションの責任をクラウド運用チームに移します。

## アイドル状態のアプリケーション

90 日間の平均的な CPU およびメモリ使用率が 5~20% のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するか、オンプレミスに保持するのが一般的です。

## インクリメンタル・マイグレーション

1 回の完全なカットオーバーを行うのではなく、アプリケーションを少しずつ移行するカットオーバー戦略です。たとえば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移動できます。すべてが正常に動作していることを確認したら、レガシーシステムを廃止できるまで、追加のマイクロサービスまたはユーザーを段階的に移動できます。この戦略により、大規模な移行に伴うリスクが軽減されます。

## IT 情報ライブラリ (ITIL)

IT サービスを提供し、これらのサービスをビジネス要件に合わせるための一連のベストプラクティス。ITIL は ITSM の基盤を提供します。

## IT サービス管理 (ITSM)

組織の IT サービスの設計、実装、管理、およびサポートに関連する活動。クラウドオペレーションと ITSM ツールの統合については、「[オペレーション統合ガイド](#)」を参照してください。

## ランディングゾーン

ランディングゾーンは、Well-Architected の、スケーラブルで安全なマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロードとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、[安全でスケーラブルなマルチアカウント AWS 環境のセットアップ](#)を参照してください。

## 大規模な移行

300 台以上のサーバの移行。

## Migration Acceleration Program (MAP)

組織がクラウドへの移行のための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP には、組織的な方法でレガシー移行を実行するための移行方法論と、一般的な移行シナリオを自動化および高速化する一連のツールが含まれています。

## 移行ポートフォリオ評価 (MPA)

AWS クラウドに移行するためのビジネスケースを検証するための情報を提供するオンラインツール。MPA は、詳細なポートフォリオ評価 (サーバの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェブプランニング) を提供します。[MPA ツール](#) (ログインが必要) は、すべての人に無料で利用できる AWS コンサルタントと APN パートナーコンサルタントです。

## 移行準備状況評価 (MRA)

組織のクラウド対応状況に関するインサイトを獲得し、長所と短所を特定し、AWS CAF を使用して特定されたギャップを埋めるためのアクションプランを構築するプロセス。詳細については、[移行準備状況ガイド](#)を参照してください。MRA [AWSは移行戦略の第一段階です](#)。

## 大規模な移行

アプリケーションポートフォリオの大部分を次々にクラウドに移行し、各ウェブでより多くのアプリケーションを高速に移動させるプロセス。この段階では、以前の段階から学んだベストプラクティスと教訓を使用して、移行ファクトリーチーム、ツール、プロセスのうち、オートメーションとアジャイルデリバリーによってワークロードの移行を合理化します。これは、[AWS 移行戦略](#) の第 3 段階です。

## 移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行工場チームには通常、オペレーション、ビジネスアナリストとオーナー、移行エンジニア、開発者、DevOpsスプリントに携わる専門家が含まれます。エンタープライズアプリケーションポートフォリオの 20~50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、[このコンテンツセットのマイグレーションファクトリーの説明とクラウドマイグレーションファクトリーガイドを参照してください](#)。

## 移行メタデータ

移行を完了するために必要なアプリケーションおよびサーバに関する情報。移行パターンごとに、異なる一連の移行メタデータが必要です。移行メタデータの例として、ターゲットサブネット、セキュリティグループ、AWS アカウントが挙げられます。



## 移行パターン

移行戦略、移行先、および使用する移行アプリケーションまたはサービスを詳述する、反復可能な移行タスク。例: AWS Application Migration Service を使用して Amazon EC2 への移行を再ホストする。

## 移行戦略

ワークロードを AWS クラウドに移行するために使用するアプローチ。詳細については、この用語集の「[7 Rs \(p. 18\)](#) エントリー」と、「[組織を動員して大規模な移行を加速する](#)」を参照してください。

## オフライン移行

移行プロセス中にソースワークロードを停止する移行方法。この方法はダウンタイムが長くなるため、通常は小規模でクリティカルではないワークロードに使用されます。

## オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も引き続き機能します。この方法はダウンタイムがゼロまたは最小限に抑えられ、通常は重要な本番環境のワークロードに使用されます。

## オペレーショナルレベルアグリーメント (OLA)

サービスレベルアグリーメント (SLA) をサポートするために、どの機能的 IT グループが互いに提供することを約束するかを明確にする契約。

## オペレーション統合 (OI)

クラウドでオペレーションをモダナイズするプロセスには、準備計画、オートメーション、統合が含まれます。詳細については、[オペレーション統合ガイド](#) を参照してください。

## 組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の導入を加速し、移行問題に対処し、文化や組織の変化を推進することで、組織が新しいシステムと戦略の準備と移行するのを支援します。AWS 移行戦略では、クラウド導入プロジェクトに必要な変更のスピードから、このフレームワークは人材の高速化と呼ばれます。詳細については、[OCM ガイド](#) を参照してください。

## プレイブック

クラウドでのコアオペレーション機能の提供など、移行に関連する作業を取り込む、事前定義された一連のステップ。プレイブックは、スクリプト、自動ランブック、またはお客様のモダナイズされた環境を運用するために必要なプロセスや手順の要約などの形式をとることができます。

## ポートフォリオ評価

移行を計画するために、アプリケーションポートフォリオの検出、分析、優先順位付けを行うプロセス。詳細については、「[移行準備状況ガイド](#)」を参照してください。

## 実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

プロジェクト内のロールと責任を定義して割り当てるマトリックス。例えば、お客様が RACI を作成して、セキュリティコントロールの所有権を定義したり、移行プロジェクト内の特定のタスクの役割と責任を特定したりできます。

## ランブック

特定のタスクを実行するために必要な手動または自動化された一連の手順。これらは通常、エラー率の高い反復操作や手順を合理化するために構築されています。

## サービスレベルアグリーメント (SLA)

サービスのアップタイムやパフォーマンスなど、IT チームがお客様に提供すると約束したものを明示した合意書。

## タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要がある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

## ワークストリーム

特定のタスクセットを担当する移行プロジェクト内の機能グループ。各ワークストリームは独立していますが、プロジェクト内の他のワークストリームをサポートしています。たとえば、ポートフォリオワークストリームは、アプリケーションの優先順位付け、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

## ゾンビアプリケーション

平均 CPU およびメモリ使用率が 5% 未満のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するのが一般的です。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。