



AWS 大規模な移行のためのポートフォリオプレイブック

AWS 規範ガイド



AWS 規範ガイド: AWS 大規模な移行のためのポートフォリオプレイブック

Copyright © 2026 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 Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

序章	1
大規模な移行に関するガイド	2
ランブック、ツール、テンプレートについて	2
ステージ 1: 初期化	4
タスク 1: 最初の検出を実行し、移行戦略を検証する	5
ステップ 1: 検出データを検証する	5
ステップ 2: ビジネスと技術の推進要因を特定する	7
ステップ 3: 移行戦略を検証する	9
ステップ 4: 移行パターンを検証する	11
タスク終了条件	14
タスク 2: メタデータを識別、収集、保存するためのプロセスを定義する	14
ステップ 1: 必要なメタデータを定義する	15
ステップ 2: メタデータストレージとコレクションプロセスを構築する	25
ステップ 3: ランブックにメタデータ要件と収集プロセスを文書化する	32
タスク終了条件	33
タスク 3: アプリケーションの優先順位付けプロセスを定義する	33
複雑なスコアリング基準について	34
ステップ 1: アプリケーションの優先順位付けプロセスを定義する	41
ステップ 2: アプリケーションの優先順位付けルールを定義する	46
ステップ 3: アプリケーションの優先順位付けプロセスを完了する	47
タスク終了条件	47
タスク 4: アプリケーションのディープダイブプロセスを定義する	48
ステップ 1: アプリケーションワークショッププロセスを定義する	48
ステップ 2: アプリケーションマッピングプロセスを定義する	53
ステップ 3: (オプション) アプリケーションターゲットの状態を定義する	62
ステップ 4: アプリケーションのディープダイブプロセスを完了する	66
タスク 5: ウェーブプランニングプロセスの定義	67
ステップ 1: 移動グループプロセスを定義する	67
ステップ 2: ウェーブプランニングの選択基準を定義する	71
ステップ 3: ウェーブプランニングプロセスを完了する	73
タスク終了条件	74
ステージ 2: の実装	75
進行状況の追跡	75
タスク 1: アプリケーションの優先順位付け	76

タスク 2: アプリケーションの詳細な分析を実行する	77
タスク 3: ウェーブプランニングとメタデータ収集の実行	77
リソース	80
AWS 大規模な移行	80
その他のリファレンス	80
ツールとサービス	80
AWS 規範ガイド	80
動画	80
寄稿者	81
ドキュメント履歴	82
用語集	83
#	83
A	84
B	86
C	88
D	91
E	95
F	98
G	99
H	100
I	102
L	104
M	105
O	109
P	112
Q	115
R	115
S	118
T	122
U	123
V	124
W	124
Z	125
.....	CXXvi

AWS 大規模な移行のためのポートフォリオプレイブック

Amazon Web Services ([寄稿者](#))

2024 年 7 月 ([ドキュメント履歴](#))

Note

このプレイブックのタスクを完了するには、アプリケーションポートフォリオの初期の大まかな検出と評価を実行することが前提条件です。このプロセスを完了する方法の詳細については、[AWS クラウド「移行のためのアプリケーションポートフォリオ評価ガイド」](#)を参照してください。

大規模な移行では、ポートフォリオワークストリームは移行するアプリケーションのウェーブを計画し、移行ワークストリームはそれらのウェーブの移行に焦点を当てます。ウェーブを計画する場合、ポートフォリオワークストリームはポートフォリオを評価し、移行に必要なメタデータを収集し、アプリケーションの優先順位を付け、アプリケーションをウェーブに割り当てる責任があります。ウェーブは、移行ワークストリームの容量に応じてサイズ設定およびスケジュールする必要があります。アプリケーションの複雑さ、依存関係、予算、パフォーマンス目標、リソースの可用性、期限などのビジネス要因を考慮する必要があります。コアおよびサポートワークストリームの詳細については、[「大規模な移行のための Foundation プレイブック」の「大規模な移行におけるワークストリーム AWS」](#)を参照してください。

このプレイブックは、アプリケーション評価やウェーブプランニングなど、大規模な移行プロジェクトの詳細なポートフォリオ評価を実行するためのstep-by-stepアプローチを提供します。ここでは、大規模な移行、初期化、実装の両方の段階にまたがるポートフォリオワークストリームのタスクについて説明します。

- ステージ 1 で初期化し、最初のポートフォリオ検出と移行戦略を検証し、ポートフォリオ評価とウェーブプランニングに使用されるプロセスとルールを定義するランブックを作成します。ステージ 1 の最後に、独自のポートフォリオ、プロセス、インフラストラクチャに合わせてカスタマイズされたポートフォリオランブックと追跡ツールがあります。
- ステージ 2 では、ポートフォリオ評価とウェーブプランを完了するために、前のステージで作成したランブックを使用します。

さらに、次のツールを作成します。このツールは、進行状況の追跡や、決定やその他の重要な情報の文書化に使用します。

- アプリケーションの複雑さスコアシート
- アプリケーションターゲットの状態ワークシート
- ポートフォリオ評価の進行状況トラッカー
- アプリケーション所有者向けのアンケート
- ウェーブプランニングと移行ダッシュボード

[ポートフォリオプレイブックテンプレート](#)を使用し、ポートフォリオ、プロセス、環境に合わせてカスタマイズすることをお勧めします。このプレイブックの手順では、これらの各テンプレートをいつ、どのようにカスタマイズするかを説明します。このプレイブックには、次のテンプレートが含まれています。

- アプリケーションターゲット状態ワークシート – このテンプレートを使用して、アプリケーションまたは移行戦略が特に複雑な場合に、AWS クラウド内のアプリケーションの将来の状態を定義します。
- ウェーブ計画と移行用のダッシュボードテンプレート – このテンプレートを使用して、重要なメタデータの照合、アプリケーションポートフォリオの分析、依存関係の特定、移行ウェーブの計画を行います。
- ポートフォリオ評価の進行状況追跡テンプレート – このテンプレートを使用して、ポートフォリオワークストリームを通じて各アプリケーションの進行状況を追跡します。
- アプリケーション所有者向けのアンケートテンプレート – このテンプレートは、アプリケーション所有者から直接アプリケーションに関する情報を収集するために、アプリケーションのディープダイブプロセスで使用します。
- アプリケーションの優先順位付けのためのランブックテンプレート – このテンプレートは、独自のアプリケーションの優先順位付けと詳細プロセスを構築するための出発点です。
- メタデータ管理用のランブックテンプレート – このテンプレートは、独自のメタデータ識別と収集プロセスを構築するための出発点です。
- ウェーブプランニング用のランブックテンプレート – このテンプレートは、独自のウェーブプランニングプロセスを構築するための出発点です。
- アプリケーションの複雑さに関するスコアシートテンプレート – このテンプレートを使用して、各アプリケーションをクラウドに移行する複雑さを評価し、その結果のスコアをアプリケーションの優先順位付けプロセスで使用できます。

ステージ 1: 大規模な移行を初期化する

初期化段階では、実装段階で詳細なポートフォリオ評価とウェーブプランを完了するために使用するランブックを定義します。別のチームメンバーが大規模な移行プロジェクトでランブックの定義を担当している場合は、[ステージ 2: 大規模な移行の実装](#)に進み、ランブックを使用してアプリケーションとサーバーのウェーブを移行します。この段階で行われた決定を文書化することで、実用的なランブックを作成します。たとえば、以下の質問について決定を下すと、ポートフォリオランブックに文書化する標準手順につながります。

- 必要な移行メタデータとその収集方法
- アプリケーションに優先順位を付け、ディープダイブを実行するにはどうすればよいですか？
- ウェーブをどのように計画しますか？

ステージ 1 では、移行をサポートするためにランブックのアクティビティがステージ 2 で何度も繰り返されるため、ルールの定義とランブックの構築に多くの時間を費やします。

ステージ 1 は、以下のタスクとステップで構成されます。

- [タスク 1: 最初の検出を実行し、移行戦略を検証する](#)
 - [ステップ 1: 検出データを検証する](#)
 - [ステップ 2: ビジネスと技術の推進要因を特定する](#)
 - [ステップ 3: 移行戦略を検証する](#)
 - [ステップ 4: 移行パターンを検証する](#)
- [タスク 2: メタデータを識別、収集、保存するためのプロセスを定義する](#)
 - [ステップ 1: 必要なメタデータを定義する](#)
 - [ステップ 2: メタデータストレージとコレクションプロセスを構築する](#)
 - [ステップ 3: ランブックにメタデータ要件と収集プロセスを文書化する](#)
- [タスク 3: アプリケーションの優先順位付けプロセスを定義する](#)
 - [ステップ 1: アプリケーションの優先順位付けプロセスを定義する](#)
 - [ステップ 2: アプリケーションの優先順位付けルールを定義する](#)
 - [ステップ 3: アプリケーションの優先順位付けプロセスを完了する](#)
- [タスク 4: アプリケーションのディープダイブプロセスを定義する](#)
 - [ステップ 1: アプリケーションワークショッププロセスを定義する](#)

- [ステップ 2: アプリケーションマッピングプロセスを定義する](#)
- [ステップ 3: \(オプション\) アプリケーションターゲットの状態を定義する](#)
- [ステップ 4: アプリケーションのディープダイブプロセスを完了する](#)
- [タスク 5: ウェーブプランニングプロセスの定義](#)
 - [ステップ 1: 移動グループプロセスを定義する](#)
 - [ステップ 2: ウェーブプランニングの選択基準を定義する](#)
 - [ステップ 3: ウェーブプランニングプロセスを完了する](#)

タスク 1: 最初の検出を実行し、移行戦略を検証する

大規模な移行プロジェクトにおけるポートフォリオ評価の最初のステップは、現在持っている情報、ビジネスおよび技術上の推進要因、および既に行われている移行戦略の決定を理解することです。ポートフォリオ評価の結果は、移行メタデータ、ウェーブプラン、移行戦略を移行ワークストリームに継続的にフィードすることです。収集された情報に基づいてギャップを分析し、次のステップを決定します。分析とタスクをすでに完了している場合は、このプレイブックの一部のセクションをスキップできます。このタスクは、次のステップで構成されます。

- [ステップ 1: 検出データを検証する](#)
- [ステップ 2: ビジネスと技術の推進要因を特定する](#)
- [ステップ 3: 移行戦略を検証する](#)
- [ステップ 4: 移行パターンを検証する](#)

ステップ 1: 検出データを検証する

動員フェーズでは、最初のポートフォリオ評価を完了した可能性があります。完了している場合は、その検出データを移行フェーズで再利用できます。そうでない場合は、心配しないでください。このプレイブックでは、大規模な移行をサポートするために必要なものについて説明します。

大規模な移行には通常、大量のデータがあります。たとえば、以下があるとします。

- ソースサーバー、アプリケーション、データベースに関するメタデータ
- 設定管理データベース (CMDB) からの IT ポートフォリオに関する情報
- 現在の状態と依存関係をよりよく理解するのに役立つ検出ツールからのデータ
- ターゲット AWS リソースのメタデータ

メタデータのタイプについて

以下は、大規模な移行をサポートするために必要な 3 つの主なメタデータタイプです。

- ソースポータルメタデータ – ソースポータルメタデータは、ソースサーバー、アプリケーション、データベースに関するメタデータです。メタデータは、既存の CMDB、検出ツール、またはアプリケーション所有者から取得できます。このメタデータタイプの包括的なリストを以下に示します。
 - [Server name] (サーバー名)
 - サーバー IP アドレス
 - サーバーオペレーティングシステム (OS)
 - サーバーストレージ、CPU、メモリ、および 1 秒あたりの入出力オペレーション (IOPS)
 - アプリケーション名
 - アプリ所有者
 - Application-to-application 依存関係
 - ビジネスユニット
 - Application-to-server マッピング
 - Application-to-database マッピング
 - データベースのタイプとサイズ
 - ストレージタイプとサイズ
 - 依存関係メタデータ
 - パフォーマンスと使用状況のデータ
- ターゲット環境メタデータ – これは、サーバーをターゲット環境に移行するのに役立つメタデータタイプです。ターゲット環境を決定する必要があります。このメタデータの一部は、検出ツールから取得できます。このメタデータタイプの例を次に示します。
 - ターゲットサブネット
 - ターゲットセキュリティグループ
 - ターゲットインスタンスタイプ
 - Target AWS Identity and Access Management (IAM) ロール
 - ターゲット IP アドレス
 - ターゲット AWS アカウント ID
 - ターゲット AWS リージョン
 - ターゲット AWS サービス

- ターゲットアプリケーションアーキテクチャ設計
- ウェーブプランニングメタデータ – ウェーブプランニングメタデータは、移行の管理に役立つメタデータタイプです。このメタデータタイプの例を次に示します。
 - Wave ID
 - ウェーブ開始時刻
 - ウェーブカットオーバー時間
 - ウェーブ所有者
 - Wave から application/server/database/移動グループへのマッピング

検出データを検証する

決定を行う前に、現在の検出データを理解することが重要です。移行のこの段階では、すべての情報がない可能性があります。このプレイブックは、メタデータ要件を定義し、メタデータを効率的に収集するのに役立ちます。次の質問をして、現在利用可能なメタデータとその場所を特定します。

- Migration Evaluator などのツールを使用して移行評価を実施しましたか？
- Flexera One Cloud Migration や Modernization などの検出ツールを環境にデプロイしましたか？
- IT ポートフォリオに関する up-to-date がある CMDB がありますか？
- 動員フェーズで最初のポートフォリオ評価を完了しましたか？
- 初期ウェーブ計画を完了しましたか？
- 最初のターゲット環境設計を完了しましたか？
- 各メタデータタイプのソースは何ですか？
- すべてのメタデータにアクセスできますか？
- すべてのメタデータにアクセスするにはどうすればよいですか？
- メタデータにアクセスするプロセスを文書化していますか？

ステップ 2: ビジネスと技術の推進要因を特定する

ビジネスとテクノロジーの推進要因は、各アプリケーションの高レベルの移行戦略とパターンを考慮する際に重要です。移行に固有のドライバーを理解する必要があります。移行戦略を検証し、アプリケーションマッピングルールを定義するときは、これらのビジネスドライバーと技術ドライバーを使用します。

一般的なビジネスドライバー

ビジネスドライバーは、契約の有効期限切れ、急速な成長、予算など、大規模な移行を計画する際に考慮すべきビジネス目標や制限に関連する要因です。一般的なビジネス推進要因は次のとおりです。

- データセンターの終了 – できるだけ早くクラウドに移行する必要があります。たとえば、データセンター契約の有効期限が近づいています。
- 運用コストとリスクの削減 – オンプレミス環境の運用に関連するコストやリスクを軽減したいと考えています。
- 柔軟性 – ビジネスの未来の変化に備えるために、戦略的方向性としてクラウドに移行する必要があります。
- ビジネスの成長 – 開発とイノベーションを迅速に加速したり、急速な成長に対応したりする必要があります。
- データをインテリジェントに使用する – 企業の成長を予測し、顧客の行動に関するインサイトを提供できるクラウドベースの人工知能、機械学習、モノのインターネット (IoT) を活用したいと考えています。
- セキュリティとコンプライアンスの向上 – クラウド AWS インフラストラクチャに既に組み込まれているコンプライアンスプログラムを活用する必要があります。または、データに対する脅威の可能性を警告できるソフトウェアベースのセキュリティツールを活用する必要があります。
- リソースの可用性 – リソースが限られているか、内部経験が限られていると、変更せずにアプリケーションを移動する戦略を選択する可能性があります。

一般的なテクニカルドライバー

技術的な要因は、現在のアーキテクチャなど、大規模な移行を計画する際に考慮すべき技術的な目標や制限に関連する要因です。一般的な技術推進要因は次のとおりです。

- ハードウェアまたはソフトウェアのend-of-support – ハードウェアまたはソフトウェアはライフサイクルの終了に近づいており、ベンダーがサポートしなくなったため、更新する必要があります。
- テクノロジー統合 – グローバルインフラストラクチャにアクセスして、アプリケーションを迅速かつ戦略的にスケールアップできます。グローバルサービスとインフラストラクチャを活用して、迅速にグローバル化できます。
- ストレージとコンピューティングの制限 – データセンターにはより多くのストレージやサーバー用の容量がないため、拡張する別の場所を見つける必要があります。
- スケーラビリティと回復性の要件 – アプリケーションでは過去にダウンタイムが発生しており、クラウドを使用して目標復旧時点 (RPO) と目標復旧時間 (RTO) を改善したいと考えています。

- アプリケーションアーキテクチャのモダナイズ – クラウドを活用し、アプリケーションをクラウドネイティブに変更したいと考えています。
- パフォーマンスの向上 – ピーク時にはアプリケーションのパフォーマンスが低下します。需要に合わせて自動的にスケールアップ/ダウンしたいと考えています。

ランブックを更新する

1. [ポートフォリオプレイブックテンプレート](#)で、アプリケーションの優先順位付け用のランブックテンプレート (Microsoft Word 形式) を開きます。
2. ビジネスおよび技術ドライバーセクションで、大規模な移行プロジェクトで特定したドライバーを記録します。
3. アプリケーションの優先順位付けランブックを保存します。

ステップ 3: 移行戦略を検証する

大規模な移行には、移行戦略の選択が不可欠です。選択した移行戦略が組織の期待、制限、要件を満たしていることを確認する必要があります。利用可能な移行戦略の詳細については、[AWS 「大規模な移行用ガイド」](#)を参照してください。

動員フェーズまたは初期ポートフォリオ評価中に移行戦略を選択した可能性があります。このステップでは、ビジネスドライバーと技術ドライバーを使用して、ポートフォリオの移行戦略を選択および検証します。

ポートフォリオの評価を継続し、移行を開始すると、移行戦略が変わる可能性があります。この段階では、各移行戦略に対するポートフォリオの一般的な分布を理解することが目標です。移行戦略の選択は次のステップで重要であり、詳細な移行パターンを検証します。

移行戦略を選択して検証する

ポートフォリオを評価し、次のように移行戦略を選択します。

1. 前のステップで特定したすべての技術的およびビジネス上の推進要因を確認し、ビジネスニーズに基づいて推進要因に優先順位を付けます。
2. 各ビジネスドライバーと技術ドライバーを移行戦略にマッピングします。次の表は例です。

優先度	ビジネスドライバーまたはテクニカルドライバー	移行戦略
1	指定された日付までにデータセンターを終了する	できるだけ多くのアプリケーションをリホストし、リホストが不可能な場合にのみリプラットフォームとリファクタリングを行います。
2	運用コストとリスクを軽減する	移行を高速化するには、できるだけ多くのアプリケーションをリホストします。
3	ハードウェアまたはソフトウェアのend-of-support	サポートされているアプリケーションをリホストし、クラウド内の新しいハードウェアやソフトウェアではサポートされていないアプリケーションをリプラットフォームします。
4	リソースの可用性	(AWS Managed Services AMS) にリホストして、運用オーバーヘッドを削減します。

3. 各ビジネスドライバーと技術ドライバーを比較検討し、ポートフォリオを大まかに評価することで、各移行戦略間でアプリケーションを分散させる方法を見積もります。ドライバー間で競合が発生するのが一般的です。プロジェクトの関係者は協力して、競合を解決するための最終決定を行う必要があります。以下は、ポートフォリオを各移行戦略に分散する方法の例です。

- リホスト – 60%
- リプラットフォーム – 15%
- リタイア – 10%
- 保持 – 5%
- 再購入 – 5%
- リファクタリング – 5%

ポートフォリオに高レベルの移行戦略を選択するまで、移行を続行しないでください。

ランブックを更新する

1. アプリケーションの優先順位付けランブックを開きます。
2. 「移行戦略」セクションに、アプリケーションのワークロードを7つの移行戦略に分散する方法を記録します。例えば、次のようになります。
 - リホスト – 60%
 - リプラットフォーム – 15%
 - リタイア – 10%
 - 保持 – 5%
 - 再購入 – 5%
 - リファクタリング – 5%
3. アプリケーションの優先順位付けランブックを保存します。

ステップ 4: 移行パターンを検証する

移行パターンについて

移行パターンは、移行戦略、移行先、および使用される移行アプリケーションまたはサービスの詳細を示す反復可能な移行タスクです。例として、を使用した Amazon Elastic Compute Cloud (Amazon EC2) へのリホスト AWS Application Migration Serviceがあります。以下の AWS サービスとソリューションは、一般的な移行パターンで頻繁に参照されます。

- AWS App2Container
- AWS Application Migration Service (AWS MGN)
- AWS CloudFormation
- AWS Database Migration Service (AWS DMS)
- AWS DataSync
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic File System (Amazon EFS)
- AWS クラウド移行ファクトリーソリューション
- Amazon Relational Database Service (Amazon RDS)

- AWS Schema Conversion Tool (AWS SCT)
- AWS Transfer Family

移行戦略の選択と同様に、以前のフェーズで移行パターンを既に特定している可能性があります。ただし、それらを検証し、パターンが定義および文書化されていることを確認する必要があります。次の表に、一般的な移行戦略とパターンを示します。

ID	方針	パターン
1	リホスト	Application Migration Service または Cloud Migration Factory を使用して Amazon EC2 にリホストする
2	リプラットフォーム	AWS DMS と を使用して Amazon RDS にリプラットフォームする AWS SCT
3	リプラットフォーム	を使用した Amazon EC2 へのリプラットフォーム CloudFormation
<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>CloudFormation テンプレートは、に新しいインフラストラクチャを構築します AWS クラウド。</p> </div>		
4	リプラットフォーム	AWS DataSync または を使用して Amazon EFS にリプラットフォームする AWS Transfer Family

ID	方針	パターン
5	リプラットフォーム	AWS App2Container を使用して Amazon ECS にリプラットフォームする
6	リプラットフォーム	エミュレータを使用してメインフレームまたはミッドレンジサーバーを Amazon EC2 にリプラットフォームする
7	リプラットフォーム	Amazon EC2 での Windows から Linux へのリプラットフォーム
8	リタイア	アプリケーションを廃止します。
9	Retain	オンプレミスで保持する
10	再購入	SaaS の再購入とアップグレード
11	リファクタリングまたはリアーキテクト	アプリケーションを再設計する

ランブックを更新する

この時点で、ポートフォリオレベルでパターンを定義します。このプレイブックの後半では、各アプリケーションに対応する移行パターンにマッピングします。

1. アプリケーションの優先順位付けランブックを開きます。
2. 「移行パターン」セクションで、特定して検証した移行パターンを記録します。各パターンに一意の ID を割り当て、そのパターンの移行戦略を書き留めます。
3. アプリケーションの優先順位付けランブックを保存します。

移行パターンは、進行するにつれて変更される可能性があることに注意してください。後で新しい情報を見ついたり、ワークロードの範囲を変更したり、新しい AWS サービスを使用することを決定したりする際に、移行戦略やパターンを変更することができます。

タスク終了条件

高レベルのポートフォリオの観点から移行戦略とパターンをまだ特定していない場合は、次のタスクに進む前に技術チームと協力して定義することを強くお勧めします。ポートフォリオの評価とウェブプランニングは、移行戦略とパターンの理解に依存します。続行する前に、移行パターンの包括的なリストを用意する必要はありません。新しいパターンを追加したり、戦略を調整したりできます。

次のタスクを完了したら、次のタスクに進みます。

- 最新の検出データにアクセスして理解できます。
- 移行のビジネスおよび技術的な推進要因を特定しました。
- ビジネスと技術の推進要因に基づいて、移行戦略を選択し、検証しました。
- 移行パターンを選択して検証しました。
- アプリケーションの優先順位付けランブックに以下を文書化しました。
 - ビジネスと技術の推進要因
 - 移行戦略
 - 移行パターン

タスク 2: メタデータを識別、収集、保存するためのプロセスを定義する

前のタスクでは、最初の検出データ、移行戦略、大規模な移行の移行パターンを検証しました。このタスクでは、必要なメタデータを特定し、どのように収集するかを決定します。このタスクは、次のステップで構成されます。

- [ステップ 1: 必要なメタデータを定義する](#)
- [ステップ 2: メタデータストレージとコレクションプロセスを構築する](#)
- [ステップ 3: ランブックにメタデータ要件と収集プロセスを文書化する](#)

このセクションのステップを完了するときは、メタデータの観点から移行サイクル全体を考慮してください。ポートフォリオ評価、ウェブプランニング、移行、テスト、カットオーバー後のアクティ

ビティを検討し、考えられるすべてのユースケースと関連するユースケースを分析します。完全な移行プロセスを完了するために必要な情報について考えると、そのパターンのすべてのメタデータを識別するのに役立ちます。

ステップ 1: 必要なメタデータを定義する

必要なメタデータ属性を決定する前に、移行パターンを理解する必要があります。たとえば、サーバーを Amazon EC2 に移行したり、データベースを Amazon RDS に移行したりするには、異なるメタデータが必要です。ほとんどのパターンは、多くの小さなタスクで構成されています。移行パターンを実行するには、必要なメタデータ属性を確認してから、そのアプリケーションのメタデータを収集する必要があります。実装段階で移行を効率的かつ遅滞なく実行できるように、初期化段階で必要なメタデータを決定して収集する必要があります。

メタデータ属性を定義する人またはチームは、移行パターンの実行に必要なステップとタスクを定義することから始めます。タスクは必要なメタデータを決定するため、各タスクを操作すると、必要なメタデータの包括的なコレクションが構築されます。通常、必要なメタデータを決定する担当者は、移行パターンを完了する方法を包括的に理解する必要があります。移行ランブックの作成者との調整が必要になる場合があります。詳細については、[AWS「大規模な移行の移行プレイブック」](#)を参照してください。

大規模な移行中、メタデータに依存するすべてのワークストリームにまたがる多くのプロセスがあります。タイムリーで正確なメタデータを持つことは、大規模な移行の成功に幅広く大きな影響を与えます。

このステップでは、パターンまたはタスクを定義し、その定義を使用して必要なメタデータを識別します。

移行パターンとサポートタスクの主要なコンポーネントを特定する

このステップでは、移行パターンまたはサポートタスクごとに、アクション、ソースオブジェクト、ターゲットオブジェクト、使用するツールなどの主要なコンポーネントを定義します。次に、回答に基づいてパターンまたはタスクに名前を付けます。

サポートタスクには、ウェーブプランニング、アプリケーションの優先順位付け、依存関係分析、ガバナンス、ディザスタリカバリ、パフォーマンステスト、ユーザー受け入れテストなど、ポートフォリオと移行ワークストリームが移行中に実行する必要がある運用アクティビティが含まれます。これらのタスクをサポートするにはメタデータが必要なため、移行パターンとサポートタスクの両方に対してこれらのステップを実行します。

1. アクション – 移行戦略またはサポートタスクを特定します。1つのアクションには、他のアクションが関連付けられている可能性があることに注意してください。たとえば、移行のオペレーションを定義できません。アクションの例は次のとおりです。
 - リホスト、リプラットフォーム、再配置などの移行戦略
 - ウェーブプランニング
 - アプリケーションの優先順位付けと依存関係の分析
 - 運用
 - ガバナンス
 - ディザスタリカバリ
 - パフォーマンステストやユーザー受け入れテスト (UAT) などのテスト
2. ソースオブジェクト – アクションを実行するソースオブジェクトを特定します。ソースオブジェクトの例は次のとおりです。
 - ウェーブ
 - サーバー
 - データベース
 - ファイル共有
 - アプリケーション
3. ツール – アクションを実行するために使用されるサービスまたはツールを特定します。複数のツールまたはサービスを使用できます。ツールの例は次のとおりです。
 - AWS Application Migration Service
 - AWS DataSync
 - AWS Database Migration Service (AWS DMS)
 - AWS Backup
 - パフォーマンスモニタリングツール
4. ターゲットオブジェクト – アクションの完了時にソースが存在するターゲットオブジェクト、サービス、または場所を特定します。オブジェクト、サービス、または場所の例は次のとおりです。
 - Amazon Elastic Compute Cloud (Amazon EC2)
 - Amazon Relational Database Service (Amazon RDS)
 - Amazon Elastic File System (Amazon EFS)
 - Amazon Elastic Container Service (Amazon ECS)

- ウェーブプラン

5. パターン名 – 次のように前のステップに対する回答を結合します。

<tool> を使用した <action> <source object> on/to <target object>

次に例を示します。

- Application Migration Service または Cloud Migration Factory (ツール) を使用して、ウェブ、アプリケーション、またはサーバー (ソースオブジェクト) を Amazon EC2 (ターゲットオブジェクト) にリホスト (アクション) する
- DataSync (ツール) を使用して (アクション) ファイル共有 (ソースオブジェクト) を Amazon EFS (ターゲットオブジェクト) にリプラットフォームする
- (ツール) を使用して (アクション) データベース (ソースオブジェクト) を Amazon RDS (ターゲットオブジェクト) にリプラットフォーム AWS DMS する
- Amazon CloudWatch (ツール) を使用した Amazon EC2 (ターゲットオブジェクト) 上のアプリケーション (ソースオブジェクト) のパフォーマンスモニタリング (アクション)
- 移行後に (ツール) を使用して AWS Backup Amazon EC2 (ターゲットオブジェクト) で (アクション) サーバー (ソースオブジェクト) をバックアップする
- ウェーブプランを作成するためのウェーブプランニング (アクション) ウェーブ、アプリケーション、またはサーバー (ソースオブジェクト) (ターゲットオブジェクト)

以下は、移行パターンテーブルから Application Migration Service または Cloud Migration Factory を使用してパターン 1: Amazon EC2 へのリホストを記録する方法の例です。 [???](#)

Pattern ID	1
Pattern name	Application Migration Service または Cloud Migration Factory を使用して Amazon EC2 にリホストする
Action	リホスト移行
Source object	ウェブ、アプリケーション、またはサーバー
Tools	Application Migration Service または Cloud Migration Factory

Target object	Amazon EC2
---------------	------------

各パターンまたはタスクに必要なメタデータを決定する

パターンまたはタスクを定義したら、ソースオブジェクト、ターゲットオブジェクト、ツール、およびその他のビジネス情報に必要なメタデータを決定します。このプロセスを説明するために、このプレイブックでは、例として、[移行](#)パターンテーブルから Application Migration Service または Cloud Migration Factory を使用してパターン 1: Amazon EC2 にリホストします。一部のパターンやタスクでは、一部のステップが適用されない場合があります。

1. ターゲットオブジェクトを分析する – ターゲットオブジェクトから逆算し、オブジェクトを手動で作成して、オブジェクトをサポートするために必要なメタデータを特定します。次の表に示すように、メタデータをキャプチャします。

たとえば、EC2 インスタンスを作成するときは、インスタンスタイプ、ストレージタイプ、ストレージサイズ、サブネット、セキュリティグループ、タグを選択する必要があります。次の表に、ターゲットオブジェクトが EC2 インスタンスの場合に必要なメタデータ属性の例を示します。

属性名	オブジェクトの種類	説明または目的
target_subnet	ターゲット EC2 インスタンス	ターゲット EC2 インスタンスのサブネット
target_subnet_test	ターゲット EC2 インスタンス	ターゲット EC2 インスタンスのテストサブネット
target_security_group	ターゲット EC2 インスタンス	ターゲット EC2 インスタンスのセキュリティグループ
target_security_group_test	ターゲット EC2 インスタンス	ターゲット EC2 インスタンスのセキュリティグループをテストする
IAM_role	ターゲット EC2 インスタンス	AWS Identity and Access Management ターゲット

属性名	オブジェクトの種類	説明または目的
		EC2 インスタンスの (IAM) ロール
instance_type	ターゲット EC2 インスタンス	ターゲット EC2 インスタンスのインスタンスタイプ
AWS_account_ID	ターゲット EC2 インスタンス	AWS ターゲット EC2 インスタンスをホストする アカウント
AWS_Region	ターゲット EC2 インスタンス	AWS ターゲット EC2 インスタンスをホストするリージョン

2. ツールを分析する – ツールを使用してターゲットオブジェクトを作成し、違いをチェックします。次の表に示すように、ツール固有のメタデータをキャプチャし、移行ツールでサポートされていない場合は前の表から属性を削除します。たとえば、リホスト移行ツールはlike-for-likeしているため、Application Migration Service の OS タイプとストレージサイズをカスタマイズすることはできません。したがって、これらの属性が前の表に含まれている場合は、ターゲット OS とターゲットディスクサイズを削除します。前の例の表では、すべての属性がツールでサポートされているため、アクションは必要ありません。

次の表に、ツールに必要なメタデータの例を示します。

属性名	オブジェクトの種類	説明または目的
AWS_account_ID	ツール (アプリケーション移行サービス)	AWS の アカウント ID AWS Application Migration Service
AWS_Region	ツール (アプリケーション移行サービス)	AWS Application Migration Service のリージョン
replication_server_subnet	ツール (アプリケーション移行サービス)	Application Migration Service レプリケーションサーバーのサブネット

属性名	オブジェクトの種類	説明または目的
replication_server_security_group	ツール (アプリケーション移行サービス)	Application Migration Service レプリケーションサーバーのセキュリティグループ

3. ソースオブジェクトを分析する – 次のようにアクションを評価して、ソースオブジェクトに必要なメタデータを決定します。

- サーバーを移行するには、サーバーに接続するために、ソースサーバー名と完全修飾ドメイン名 (FQDN) を知る必要があります。
- サーバーとともにアプリケーションを移行するには、アプリケーション名、アプリケーション環境、および application-to-server 間のマッピングを知る必要があります。
- ポートフォリオ評価の実行、アプリケーションの優先順位付け、または移動グループの定義を行うには、application-to-server 間のマッピング、application-to-database マッピング、および application-to-application 依存関係を把握しておく必要があります。
- ウェーブを管理するには、ウェーブ ID とウェーブの開始時刻と終了時刻を知る必要があります。

次の表に、ソースオブジェクトに必要なメタデータの例を示します。

属性名	オブジェクトの種類	説明または目的
wave_ID	ソースウェーブ	ウェーブの ID (ウェーブ 10 など)
wave_start_date	ソースウェーブ	ウェーブの開始日
wave_cutover_date	ソースウェーブ	ウェーブのカットオーバー日
wave_owner	ソースウェーブ	ウェーブの所有者
app_name	ソースアプリケーション	ソースアプリケーション名
app_to_server_mapping	ソースアプリケーション	Application-to-server 関係
app_to_DB_mapping	ソースアプリケーション	Application-to-database 関係

属性名	オブジェクトの種類	説明または目的
app_to_app_dependencies	ソースアプリケーション	アプリケーションの外部依存関係
server_name	ソースサーバー	ソースサーバー名
server_FQDN	ソースサーバー	ソースサーバーの完全修飾ドメイン名
server_OS_family	ソースサーバー	ソースサーバーのオペレーティングシステム (OS) ファミリー (Windows や Linux など)
server_OS_version	ソースサーバー	ソースサーバーの OS バージョン (Windows Server 2003 など)
server_environment	ソースサーバー	ソースサーバーの環境 (開発、本番稼働、テストなど)
server_tier	ソースサーバー	ソースサーバーの階層 (ウェブ、データベース、アプリケーションなど)
CPU	ソースサーバー	ソースサーバー内の CPUs の数
RAM	ソースサーバー	ソースサーバーの RAM サイズ
disk_size	ソースサーバー	ソースサーバーのディスクサイズ

4. 他の属性を検討する – プライマリアクションに加えて、ターゲットオブジェクトまたはアプリケーションに関連する他のアクションや属性も考慮します。サンプルパターンのパターン 1: Application Migration Service または Cloud Migration Factory を使用して Amazon EC2 にリホストすると、アクションはリホストされ、ターゲットオブジェクトは Amazon EC2 です。このターゲットオブジェクトのその他の関連アクションには、Amazon EC2 へのバックアップ、移行後の

EC2 インスタンスのモニタリング、EC2 インスタンスに関連するコストの管理のためのタグの使用などがあります。また、アプリケーション所有者など、移行の管理に役立つ他のアプリケーション属性を検討することもできます。アプリケーション所有者は、質問やカットオーバーの目的で問い合わせる必要があります。

次の表に、一般的に使用される追加のメタデータの例を示します。このテーブルには、ターゲット EC2 インスタンスのタグが含まれています。タグとその使用方法の詳細については、[Amazon EC2 ドキュメントの「Amazon EC2 リソースのタグ付け」](#)を参照してください。Amazon EC2

属性名	オブジェクトの種類	説明または目的
Name	ターゲット EC2 インスタンス (タグ)	ターゲット EC2 インスタンスの名前を定義するタグ
app_owner	ソースアプリケーション	ソースアプリケーションの所有者
business_unit	ターゲット EC2 インスタンス (タグ)	ターゲット EC2 インスタンスのビジネスユニットを識別するタグ (人事、財務、IT など)
cost_center	ターゲット EC2 インスタンス (タグ)	ターゲット EC2 インスタンスのコストセンターを識別するタグ

5. テーブルの作成 – 前のステップで特定したすべてのメタデータを 1 つのテーブルに結合します。

属性名	オブジェクトの種類	説明または目的
wave_ID	ソースウェーブ	ウェーブの ID (ウェーブ 10 など)
wave_start_date	ソースウェーブ	ウェーブの開始日
wave_cutover_date	ソースウェーブ	ウェーブのカットオーバー日
wave_owner	ソースウェーブ	ウェーブの所有者

属性名	オブジェクトの種類	説明または目的
app_name	ソースアプリケーション	ソースアプリケーション名
app_to_server_mapping	ソースアプリケーション	Application-to-server関係
app_to_DB_mapping	ソースアプリケーション	Application-to-database関係
app_to_app_dependencies	ソースアプリケーション	アプリケーションの外部依存関係
AWS_account_ID	ツール (アプリケーション移行サービス)	AWS ターゲット EC2 インスタンスをホストする アカウント
AWS_Region	ツール (アプリケーション移行サービス)	AWS ターゲット EC2 インスタンスをホストするリージョン
replication_server_subnet	ツール (アプリケーション移行サービス)	Application Migration Service レプリケーションサーバーのサブネット
replication_server_security_group	ツール (アプリケーション移行サービス)	Application Migration Service レプリケーションサーバーのセキュリティグループ
server_name	ソースサーバー	ソースサーバー名
server_FQDN	ソースサーバー	ソースサーバーの完全修飾ドメイン名
server_OS_family	ソースサーバー	ソースサーバーのオペレーティングシステム (OS) ファミリー (Windows や Linux など)

属性名	オブジェクトの種類	説明または目的
server_OS_version	ソースサーバー	ソースサーバーの OS バージョン (Windows Server 2003 など)
server_environment	ソースサーバー	ソースサーバーの環境 (開発、本番稼働、テストなど)
server_tier	ソースサーバー	ソースサーバーの階層 (ウェブ、データベース、アプリケーションなど)
CPU	ソースサーバー	ソースサーバー内の CPUs の数
RAM	ソースサーバー	ソースサーバーの RAM サイズ
disk_size	ソースサーバー	ソースサーバーのディスクサイズ
target_subnet	ターゲットサーバー	ターゲット EC2 インスタンスのサブネット
target_subnet_test	ターゲットサーバー	ターゲット EC2 インスタンスのテストサブネット
target_security_group	ターゲットサーバー	ターゲット EC2 インスタンスのセキュリティグループ
target_security_group_test	ターゲットサーバー	ターゲット EC2 インスタンスのセキュリティグループをテストする
instance_type	ターゲットサーバー	ターゲット EC2 インスタンスのインスタンスタイプ

属性名	オブジェクトの種類	説明または目的
IAM_role	ターゲットサーバー	AWS Identity and Access Management ターゲット EC2 インスタンスの (IAM) ロール
Name	ターゲットサーバー (タグ)	ターゲット EC2 インスタンスの名前を定義するタグ
app_owner	ソースアプリケーション	ソースアプリケーションの所有者
business_unit	ターゲットサーバー (タグ)	ターゲット EC2 インスタンスのビジネスユニットを識別するタグ (人事、財務、IT など)
cost_center	ターゲットサーバー (タグ)	ターゲット EC2 インスタンスのコストセンターを識別するタグ

6. Repeat – 各パターンに必要なメタデータを記録するまで、このプロセスを繰り返します。

ステップ 2: メタデータストレージとコレクションプロセスを構築する

前のステップでは、移行のサポートに必要なメタデータを定義しました。このステップでは、メタデータを収集して保存するプロセスを構築します。このステップは 2 つのタスクで構成されます。

1. 前のステップの必要なメタデータを分析し、ソースを特定します。
2. メタデータを効率的に保存および収集するプロセスを定義します。

メタデータソースを分析する

一般的なメタデータソースは多数あります。通常、最初にアクセスできるのは高レベルのアセットインベントリで、通常は設定管理データベース (CMDB) または別の既存のツールからエクスポートされます。ただし、自動プロセスと手動プロセスの両方を使用して、他のソースからもメタデータを収集する必要があります。

次の表に、一般的なソース、そのソースの標準収集プロセス、そのソースから検索できる一般的なメタデータタイプを示します。

メタデータソース	コレクションタイプ	メタデータタイプ
Discovery tools	自動	ソースサーバー
CMDB	自動	ソースサーバー
の RVTools など、他のツールからのインベントリ VMware vSphere	自動	ソースサーバー
アプリケーション所有者アンケート	手動	ソースサーバー、ターゲットサーバー、ウェブ
アプリケーション所有者のインタビュー	手動	ソースサーバー、ターゲットサーバー、ウェブ
アプリケーション設計ドキュメント	手動	ターゲットサーバー
ランディングゾーン設計ドキュメント	手動	ターゲットサーバー、ツール

メタデータのすべてのソースを一覧表示したら、メタデータタイプを分析し、各ソースを前のステップで特定したメタデータ属性にマッピングします。

1. からメタデータ属性の完全なリストを取得します [ステップ 1: 必要なメタデータを定義する](#)。
2. 各メタデータタイプを分析し、自動プロセスを使用して取得できないタイプを特定します。これは通常、ターゲットサーバーメタデータとウェブメタデータタイプです。アプリケーション所有者の決定が必要なためです。たとえば、ターゲット EC2 インスタンスにはどのサブネットとセキュリティグループを使用しますか？
3. 各メタデータ属性を分析し、前の表のメタデータソースにマッピングします。複数のソースの組み合わせを持つことが一般的です。検出ツールを使用して、一部のソースサーバーメタデータを収集できます。検出ツールを使用してメタデータを収集する方法については、AWS「規範ガイド」ウェブサイトの [「自動ポートフォリオ検出の開始方法」](#) を参照してください。
4. メタデータ属性をタイプとソースにマッピングするテーブルを作成します。次の表は例です。

Metadata 属性	メタデータタイプ	メタデータソース
app_name	ソースアプリケーション	CMDB
app_owner	ソースアプリケーション	CMDB
app_to_server_mapping	ソースアプリケーション	CMDB、検出ツール、またはアプリケーション所有者のアンケート
app_to_DB_mapping	ソースアプリケーション	CMDB、検出ツール、またはアプリケーション所有者のアンケート
app_to_app_dependencies	ソースアプリケーション	CMDB、検出ツール、またはアプリケーション所有者のアンケート
server_name	ソースサーバー	CMDB
server_FQDN	ソースサーバー	CMDB
server_OS_family	ソースサーバー	CMDB
server_IP	ソースサーバー	Discovery tools
disk_size	ソースサーバー	Discovery tools
instance_type	ターゲットサーバー	Discovery tools
target_subnet	ターゲットサーバー	アプリケーション所有者アンケート
target_security_group	ターゲットサーバー	アプリケーション所有者アンケート
AWS_Region	ターゲットサーバー	アプリケーション所有者アンケート

Metadata 属性	メタデータタイプ	メタデータソース
AWS_account_ID	ターゲットサーバー	アプリケーション所有者アンケート
replication_server_subnet	ツール (アプリケーション移行サービス)	ランディングゾーン設計ドキュメント
replication_server_security_group	ツール (アプリケーション移行サービス)	ランディングゾーン設計ドキュメント
Name	ターゲットサーバー (タグ)	アプリケーション所有者アンケート
business_unit	ターゲットサーバー (タグ)	アプリケーション所有者アンケート
cost_center	ターゲットサーバー (タグ)	アプリケーション所有者アンケート
wave_ID	ウェーブプランニング	アプリケーション所有者のインタビュー
wave_start_date	ウェーブプランニング	アプリケーション所有者のインタビュー
wave_cutover_date	ウェーブプランニング	アプリケーション所有者のインタビュー

1 つのメタデータストアを定義する

各メタデータ属性をソースにマッピングした後、メタデータを保存する場所を定義します。メタデータの保存方法と保存場所にかかわらず、選択する必要があるリポジトリは 1 つだけです。これにより、信頼できる情報源が 1 つになります。メタデータを複数の場所に保存することは、大規模な移行でよくある間違いです。

オプション 1: 共有リポジトリのスプレッドシートにメタデータを保存する

このオプションは非常に手動のプロセスのように聞こえるかもしれませんが、大規模な移行では最も一般的なデータストアです。また、スプレッドシートを Microsoft SharePoint サイトなどの共有リポジトリに保存することも一般的です。

Microsoft Excel スプレッドシートはカスタマイズが簡単で、構築に時間がかかることはありません。欠点は、メタデータが多いと非常に複雑になり、サーバー、アプリケーション、データベースなどのアセット間の関係を管理するのが困難な場合があることです。もう 1 つの課題はバージョン管理です。書き込みアクセスを数人に制限するか、自動化されたプロセスを使用してスプレッドシートを更新する必要があります。

[ポートフォリオプレイブックテンプレート](#)では、独自のデータストアスプレッドシートを構築するための出発点として、ウェーブプランニングと移行用の Dashboard テンプレート (Excel 形式) を使用できます。

オプション 2: 専用ツールにメタデータを保存する

[TDS Transition Manager](#) (TDS ウェブサイト) などの構築済みのツールを使用してデータを保存するか、独自のツールを構築できます。独自のツールを構築する場合、オプション 1 の Excel スプレッドシートタブと同様にデータベーステーブルが必要です。例えば、次のようになります。

- サーバーテーブル
- アプリケーションテーブル
- データベーステーブル
- Application-to-server および application-to-database マッピングテーブル
- Wave-planning テーブル
- アプリケーション所有者アンケートテーブル

メタデータ収集プロセスを定義する

前のステップでは、メタデータをソースにマッピングし、メタデータを収集するデータストアを定義しました。このステップでは、メタデータを効果的に収集するプロセスを構築します。手動 copy-and-paste プロセスを最小限に抑え、自動化を使用して各ソースからメタデータを収集する必要があります。次の 3 つのステップがあります。

1. メタデータマッピングテーブルに基づいて、各メタデータソースの抽出、変換、ロード (ETL) スクリプトを構築します。

2. 各ソースからメタデータを自動的にインポートするスケジュールされたタスクを定期的に構築します。
3. エクスポートプロセスを構築するか、リポジトリに保存されているメタデータへのアプリケーションプログラミングインターフェイス (API) アクセスを提供します。

次の表は、各 ETL スクリプトによって収集されたメタデータ属性の例です。メタデータは、スプレッドシートや専用ツールなど、前のセクションで定義した場所に保存されます。

Metadata 属性	メタデータタイプ	メタデータソース	収集プロセス
app_name	ソースアプリケーション	CMDB	ETL スクリプト – CMDB
app_owner	ソースアプリケーション	CMDB	ETL スクリプト – CMDB
app_to_server_mapping	ソースアプリケーション	CMDB	ETL スクリプト – CMDB
app_to_DB_mapping	ソースアプリケーション	CMDB	ETL スクリプト – CMDB
app_to_app_dependencies	ソースアプリケーション	検出ツール	ETL スクリプト – 検出ツール
server_name	ソースサーバー	CMDB	ETL スクリプト – CMDB
server_FQDN	ソースサーバー	CMDB	ETL スクリプト – CMDB
server_OS_family	ソースサーバー	CMDB	ETL スクリプト – CMDB
server_OS_version	ソースサーバー	CMDB	ETL スクリプト – CMDB

Metadata 属性	メタデータタイプ	メタデータソース	収集プロセス
disk_size	ソースサーバー	検出ツール	ETL スクリプト – 検出ツール
instance_type	ターゲットサーバー	検出ツール	ETL スクリプト – 検出ツール
target_subnet	ターゲットサーバー	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
target_security_group	ターゲットサーバー	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
AWS_Region	ターゲットサーバー	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
AWS_account_ID	ターゲットサーバー	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
Name	ターゲットサーバー (タグ)	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
business_unit	ターゲットサーバー (タグ)	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
cost_center	ターゲットサーバー (タグ)	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
wave_ID	ウェーブプランニング	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者

Metadata 属性	メタデータタイプ	メタデータソース	収集プロセス
wave_start_date	ウェーブプランニング	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者
wave_cutover_date	ウェーブプランニング	アプリケーション所有者アンケート	ETL スクリプト – アプリケーション所有者

ステップ 3: ランブックにメタデータ要件と収集プロセスを文書化する

このタスクでは、決定事項をメタデータ管理ランブックに文書化します。移行中、ポートフォリオワークストリームは、メタデータを収集して保存するための標準手順として、このランブックに従います。

1. [ポートフォリオプレイブックテンプレート](#)で、メタデータ管理用のランブックテンプレート (Microsoft Word 形式) を開きます。これは、独自のランブックを構築するための出発点として機能します。
2. メタデータ属性セクションで、各移行パターンのメタデータ属性テーブルを作成し、で識別されるメタデータ属性をテーブルに入力します [ステップ 1: 必要なメタデータを定義する](#)。
3. 「ソースの場所」セクションで、で識別したソースを文書化します [メタデータソースを分析する](#)。
4. 「ソースロケーションアクセス手順」セクションで、メタデータソースロケーションにアクセスするためにユーザーが従う必要があるステップを文書化します。
5. メタデータストアセクションで、で作成したメタデータストアにアクセスするためにユーザーが従う必要があるステップを文書化します [1つのメタデータストアを定義する](#)。
6. 「データ収集タイプ」セクションで、各メタデータソースに使用するデータ収集プロセスを特定します。理想的には、自動化スクリプトを使用してすべてのメタデータ収集を自動化する必要があります。
7. メタデータ属性によるデータ収集セクションで、メタデータ属性ごとに、「」の指示に従って以下を特定します [メタデータ収集プロセスを定義する](#)。
 - a. メタデータタイプ
 - b. メタデータソース
 - c. メタデータストア

d. コレクションタイプ

8. 「メタデータの収集」セクションで、ユースケースに応じてプロセスを更新します。これは、ポートフォリオワークストリームがウェブのメタデータを収集する際に実装段階で従うプロセスです。
9. ランブックが完全で正確であることを確認します。このランブックは、移行中の信頼できるソースである必要があります。
10. レビューのためにメタデータ管理ランブックをチームと共有します。

タスク終了条件

次のタスクを完了したら、次のタスクに進みます。

- 収集されたメタデータを保存するための1つのリポジトリを準備しました。
- メタデータ管理ランブックでは、以下を定義して文書化しています。
 - 各移行パターンに必要なメタデータ属性
 - メタデータソースと各ソースへのアクセス方法の詳細な手順
 - メタデータストアとそのアクセス方法の詳細な手順
 - メタデータの収集に使用されるプロセス
 - メタデータ属性をメタデータソースとコレクションプロセスにマッピングするマッピングテーブル

タスク 3: アプリケーションの優先順位付けプロセスを定義する

アプリケーションの優先順位付けは、アプリケーションをクラウドに移行する順序を決定するプロセスです。優先度は、アプリケーションをクラウドに移行する複雑さと定義したルールに基づいて評価します。アプリケーションの優先順位付けについて議論する場合、優先度が高いと必ずしもビジネスにとってのアプリケーションの重要性と相関するとは限りません。実際、ビジネスクリティカルなアプリケーションはリスクが高いため、移行の優先順位は通常低くなります。大規模な移行では、ビジネスクリティカルではない低複雑さのアプリケーションを優先し、各ウェブでは、ますます複雑またはビジネスクリティカルなアプリケーションを移行します。

何百ものアプリケーションが移行用に並んでいる大規模な移行では、すべてのアプリケーションを一度に優先順位付けして計画することはお勧めしません。これは、大規模な移行プロジェクトにとってアプリケーションの優先順位付けプロセスを定義することが重要な理由の1つです。アジャイルな

方法で移行にアプローチするには、最も優先度の高いアプリケーション (3~10 個のアプリケーション) を選択するか、3~5 個のウェーブに十分なアプリケーションを選択できます。次に、選択したアプリケーションのみのアプリケーション検出とウェーブプランニングを完了します。このアプローチは、大規模な移行の過程でアプリケーションの優先度と波が変化することが多いため、かなりの時間を節約できます。

アプリケーションの優先度に関する一般的な誤解の 1 つは、最も優先度の高いアプリケーションが第 1 ウェーブにあることです。ウェーブプランニングを実行する場合、優先順位が最も高い 10 のアプリケーションのうち、準備が整っていないアプリケーションが最初のウェーブにあるのはわずかな可能性が高いです。これは、依存関係、ビジネス上の制約、リソースの可用性など、さまざまな有効な理由が考えられます。アプリケーションの優先度はウェーブプランニングの重要な要素ですが、考慮すべき唯一の要素ではありません。

このタスクでは、アプリケーションの優先順位付けプロセスとルールを定義します。このタスクは、次のステップで構成されます。

- [ステップ 1: アプリケーションの優先順位付けプロセスを定義する](#)
- [ステップ 2: アプリケーションの優先順位付けルールを定義する](#)
- [ステップ 3: アプリケーションの優先順位付けプロセスを完了する](#)

次のセクションでは、複雑さのスコアリングについて説明します。このプレイブックには、アプリケーションの優先順位付け方法に関する 3 つのプロセスオプションと、3 つのオプションのうちの 2 つが複雑なスコアリングを使用します。プロセスオプションの詳細については、「」を参照してください [ステップ 1: アプリケーションの優先順位付けプロセスを定義する](#)。アプリケーションの指名プロセスを使用する予定の場合は、複雑なスコアリング基準を定義する必要はなく、に直接進む必要があります [ステップ 1: アプリケーションの優先順位付けプロセスを定義する](#)。

複雑なスコアリング基準について

複雑さスコアリングは、アプリケーションを移行することの難しさを評価するために使用されるプロセスです。これは、アプリケーションを優先順位付けする際の重要な要素です。複雑さのスコアリングには、定義したのと同じ一連のビジネス基準と技術基準に照らしてすべてのアプリケーションを評価する必要があります。アプリケーションを評価するときは、各基準にスコアを割り当てます。ビジネス基準と技術基準のスコアを合計すると、そのアプリケーションの移行の全体的な複雑さを反映する複雑さスコアが得られます。その後、アプリケーションの優先順位付けやウェーブの計画時に複雑さスコアを使用できます。

複雑さスコアリング基準には 2 つのカテゴリがあります。

- **ビジネス基準** – このカテゴリの基準は、アプリケーションが使用できなくなった場合のリスク、セキュリティとコンプライアンスに関する考慮事項、リソースの可用性など、アプリケーションの移行に伴うビジネスの複雑さに関連しています。
- **技術的基準** – このカテゴリの基準は、オペレーティングシステムとそのバージョン、サーバーとユーザーの数、移行戦略など、アプリケーションの移行の技術的複雑さに関連しています。

ユースケースに適したスコアリング基準を決定する必要があります。アプリケーションの複雑さを手動でスコアリングする場合は、[ポートフォリオプレイブックテンプレート](#)で、アプリケーションの複雑さのスコアシートテンプレート (Microsoft Excel 形式) に基準とスコア値の標準セットが含まれています。これらの値から始めて、ユースケースに合わせてカスタマイズできます。アプリケーションの優先順位付けに検出ツールを使用している場合、これらのツールには通常標準の条件セットが含まれており、条件を追加、削除、または変更でき、必要に応じて重み付けできます。基準を確立するときは、次の2つのセクションの質問を使用して基準を絞り込みます。

ビジネス基準

以下は、複雑なスコアリングで一般的に使用されるビジネス基準です。

ビジネス基準	説明
ビジネスへの影響	<p>このアプリケーションが使用できなくなった場合のビジネスへの影響を評価します。</p> <ul style="list-style-type: none"> • 財務上の影響はありますか？ • オペレーションに影響はありますか？ • カスタマーエクスペリエンスに影響しますか？ • 製品または会社のイベントに影響しますか？
スタッフの可用性	<p>移行中に、アプリケーション所有者、対象分野のエキスパート (SME)、ネットワークまたはインフラストラクチャ管理者、テスター、開発者からのサポートが必要になる場合があります。移行中に役立つこれらのリソースの可用性を評価します。</p>

ビジネス基準	説明
	<ul style="list-style-type: none">• 移行チームを支援し、ガイダンスを提供するために、移行中にこのスタッフが対応できますか？• このスタッフは、移行後にアプリケーションをテストおよび検証できますか？• このスタッフは、ターゲット環境でアプリケーションを実行するために必要な IP アドレスまたはポートをプロビジョニングできますか？
ビジネスの複雑さ	<p>相互依存および相互接続されたステークホルダー、情報技術システム、組織構造が多数あると、ビジネスの複雑さが増す可能性があります。ビジネスの複雑さを次のように評価します。</p> <ul style="list-style-type: none">• ファイアウォールの変更や新しいインスタンスのプロビジョニングなど、インフラストラクチャやネットワークの変更を企業が承認するまでにどれくらいの時間がかかりますか？• 検出ツールなど、サーバーへの新しいソフトウェアまたはツールのインストールをビジネスが承認するまでにどれくらいの時間がかかりますか？

ビジネス基準	説明
準備状況	<p>アプリケーションを移行する準備ができているかどうかを次のように評価します。</p> <ul style="list-style-type: none">• アプリケーションは現在、テクノロジーの更新を受けているか、または更新が予定されているか。• メンテナンスはスケジュールされ、スケジュールされた移行と重複しますか？• アプリケーションは廃止される予定ですか？• アプリケーションは現在アップグレード中であり、新機能は開発または統合されていますか？
セキュリティ	<p>アプリケーションのセキュリティ要件とセキュリティポリシーの複雑さを次のように評価します。</p> <ul style="list-style-type: none">• アプリケーションにアクセスするために IP アドレスとポートをプロビジョニングする必要がありますか？• アプリケーションにはインフラストラクチャ保護が必要ですか？• アプリケーションにはデータ保護が必要ですか？• キー管理は必要ですか？• アプリケーションには特別なアクセス管理ポリシーが必要ですか？• アプリケーションにはモニタリングまたはログ記録が必要ですか？• アプリケーションにはインシデント対応プロセスと自動化が必要ですか？• このアプリケーションにはアラートと通知が必要ですか？

ビジネス基準	説明
コンプライアンス	<p>コンプライアンス要件は、州、業界、または企業ポリシーによって提供される法律、規制、ガイドラインなど、アプリケーションに適用される場合があります。アプリケーションのコンプライアンス要件の複雑さを次のように評価します。</p> <ul style="list-style-type: none">• データプライバシーとレジデンシーの要件はありますか？• アプリケーションに保存されているデータは暗号化する必要がありますか？• アプリケーションとの間で転送されるデータは暗号化する必要がありますか？• 監査ログ記録は必要ですか？• アプリケーションは、System and Organization Controls (SOC) などの会計および財務標準に準拠している必要がありますか？• アプリケーションは Payment Card Industry (PCI) などの支払いセキュリティ標準に準拠している必要がありますか？• アプリケーションは、医療保険の相互運用性と説明責任に関する法律 (HIPAA) などの患者の健康情報規制に準拠している必要がありますか？• アプリケーションは、情報システムセキュリティ管理および評価プログラム (ISMAP) などのパブリッククラウドセキュリティプログラムに準拠している必要がありますか？

ビジネス基準	説明
アプリケーションの知識	アプリケーション所有者などの組織内の誰かが、問題を維持、統合、トラブルシューティング、修正するための知識、スキル、経験を持っていますか？また、ビジネス需要に合わせてアプリケーションを拡張できますか？
移行スキル	組織のスタッフには、ワークロードをターゲット環境に移行するスキルがありますか？

技術基準

以下は、複雑なスコアリングで一般的に使用される技術基準です。

技術基準	説明
Storage	<p>アプリケーションの現在のストレージを次のように評価します。</p> <ul style="list-style-type: none"> アプリケーションは現在どこに保存されていますか？例としては、ネットワーク接続ストレージ (NAS)、ストレージエリアネットワーク (SAN)、ローカルドライブなどがあります。 現在のストレージの合計サイズを教えてください。
ユーザー数	このアプリケーションには何人のユーザーがいますか？実際のログまたは見積りを使用できません。
サーバー数	アプリケーションスタックにはサーバーがいくつありますか？
接続	このアプリケーションが組織内の他のユーザーにどのように接続されているかを次のように評価します。

技術基準	説明
アプリケーション OS とバージョン	<ul style="list-style-type: none">このアプリケーションに依存する他のアプリケーションはいくつありますか？このアプリケーションが使用できない場合、他のアプリケーションにはどのような影響がありますか？ <p>次のように、オペレーティングシステム (OS) とアプリケーションのサーバーのバージョンを評価します。</p> <ul style="list-style-type: none">サーバーの OS バージョンはサポートされなくなりましたか？サーバーは Unix または Linux OS を実行していますか？サーバーは Windows Server OS を実行していますか？アプリケーションはメインフレームまたはミッドレンジサーバーにありますか？
アプリケーション依存関係	<p>このアプリケーションが環境内の他のリソースにどのように依存しているかを評価します。</p> <ul style="list-style-type: none">このアプリケーションはどのリソースに依存していますか？ リソースには、他のアプリケーション、コンポーネント、OS 固有のサービス (レジストリやウェブサーバーなど)、ライブラリなどがあります。これらのリソースが 1 つ以上使用できない場合、このアプリケーションにはどのような影響がありますか？

技術基準	説明
データ移行	<p>このアプリケーションのデータまたはファイルを移行する必要があるかどうかを評価します。</p> <ul style="list-style-type: none"> データ移行はどの程度複雑ですか？ ファイル移行はどの程度複雑ですか？
移行戦略	<p>選択した移行戦略の複雑さを評価します。移行戦略の詳細については、AWS「大規模な移行ガイド」を参照してください。</p>
COTS またはカスタム	<p>アプリケーションがカスタム作成か商用off-the-shelf (COTS) かを次のように評価します。</p> <ul style="list-style-type: none"> ソースコードの最新バージョンはありますか？ アプリケーションベンダーはサポートされていますか？ アプリケーションは外部委託されていますか？

ステップ 1: アプリケーションの優先順位付けプロセスを定義する

このプレイブックには、アプリケーションを優先順位付けするための 3 つのプロセスオプションが含まれています。オプションのいずれかを選択するか、2 つ以上を組み合わせることでカスタムプロセスを構築できます。ユースケースを評価し、環境に最適なものを以下の中から判断します。

- [オプション 1: 手動の複雑さスコアリング](#) – これは、個人またはワークショップ形式のセッションで完了できる手動の優先順位付けプロセスです。このプロセスでは、複雑なスコアリング基準を使用して、各アプリケーションの移行の難しさを評価します。これは、アプリケーションの優先順位付けにおける重要な要素です。この手動プロセスは、大規模なアプリケーションポートフォリオに優先順位を付けるための一貫した定量的アプローチを提供するため、大規模な移行に適しています。ただし、定義された一連の基準に照らして各アプリケーションを評価すると、他の 2 つのオプションと比較してプロセスが遅くなる可能性があります。
- [オプション 2: アプリケーションのノミネーション](#) – これは、通常ワークショップ形式のセッションを完了する手動の優先順位付けプロセスです。このプロセスでは、アプリケーション所有者が移

行するアプリケーションを指名します。このプロセスを成功させるには、アプリケーション所有者がそれぞれのアプリケーションについて包括的な知識を持っている必要があります。このプロセスは、時間が要因であり、アプリケーションの迅速な優先順位付けが必要な場合に推奨されます。

- **オプション 3: 検出ツール** – これは自動優先順位付けプロセスです。環境内の検出ツールにアプリケーションの複雑さの自動スコアリングまたは優先順位付けのための自動機能がある場合、この機能を使用すると、時間を節約し、アプリケーションの優先順位付けプロセスを高速化できます。このプロセスでは、通常、検出ツールのパラメータ内で基準を定義し、ツールがアプリケーションを分析し、最終的な複雑さスコアを提供します。このオプションを選択する前に、検出ツールで利用できる機能を調べ、ユースケースのニーズに合わせてカスタマイズできることを確認します。

オプション 1: 手動の複雑さスコアリング

この手動アプリケーションの優先順位付けプロセスでは、ワークシートを使用して、定義された一連の複雑なスコアリング基準に照らしてアプリケーションを評価します。ワークシート形式のセッションでワークシートを完了するか、利害関係者と協力してワークシートを完了することをお勧めします。次に、最終的な複雑さスコアとアプリケーションの優先順位付けルールを使用して、アプリケーションの優先度を決定します。手動プロセスのうち、アプリケーションの複雑さを判断し、その情報を使用してアプリケーションの優先順位を付けるための最も一貫性のある定量的アプローチを提供します。

このプロセスの複雑なスコアリングステップでは、ポートフォリオプレイブックテンプレートで利用可能なアプリケーションの複雑さ (Excel 形式) にスコアシートテンプレートを使用することをお勧めします。 [samples/portfolio-playbook-templates.zip](#) このテンプレートには、事前定義されたビジネス基準と技術基準が含まれています。これらの基準を追加、削除、または変更したり、スコアリング値を調整したりできます。たとえば、1~5 ではなく 1~10 のスコア範囲を優先できます。提供されたテンプレートについては、次の点に注意してください。

- 各基準にカーソルを合わせると、その説明が表示されます。
- テンプレートに慣れたら、例を削除する必要があります。これらはデモンストレーションのみを目的としています。

移行の初期化と実装の段階で、複雑さスコアシートを最新の状態に保ちます。ポートフォリオ評価を進めるにつれてスコアが変わる場合があります。アプリケーションの詳細調査は、チームが各アプリケーションについて詳しく調べる際に、スコアシートを更新する一般的な機会です。アプリケーションの移行を妨げる未発見の依存関係や制限などの問題が発生した場合は、移行中にアプリケーションの優先度を変更することもできます。移行全体を通してスコアシートを維持することで、より正確にアプリケーションに優先順位を付けることができます。

アプリケーションの優先順位付けプロセスを次のように文書化します。

1. [ポートフォリオプレイブックテンプレート](#)で、アプリケーションの複雑さに合わせてスコアシートテンプレートを開きます。
2. アプリケーションシートで、移行に適した条件を追加、変更、または削除します。条件を変更するときは、次の操作を行います。
 - このプレイブックの [複雑なスコアリング基準について](#) セクションのガイダンスを確認してください。
 - 各基準が移行の期間、リソース、コストに与える影響を考慮してください。
 - 信頼できる複雑さスコアを得るには、組織内のさまざまなレベルの移行の複雑さを表す基準を含めます。
3. スコアリングガイドシートで、ユースケースに応じてデフォルト値と基準を更新します。
4. スコアシートを保存します。
5. アプリケーションの優先順位付けランブックを開きます。
6. アプリケーションの複雑さのスコアリング基準セクションで、スコアシートの場所を反映するようにセクションを更新します。
7. アプリケーションの優先順位付けプロセスセクションで、次の操作を行います。
 - a. オプション 1 を保持する: 手動の複雑さスコアリングを行い、他のオプションを削除します。
 - b. ユースケースに応じてプロセスを変更します。
 - c. Option という単語を含むこのセクションの見出しをすべて削除します。ランブックにこれらを残しておく、ユーザーがプロセスがオプションであるか、複数のオプションが利用可能であると考えることに混乱する可能性があります。
 - d. アプリケーションの優先順位付けランブックを保存します。

オプション 2: アプリケーションのノミネーション

この手動アプリケーションの優先順位付けプロセスは、アプリケーションの優先順位付けのための最も簡単で迅速なアプローチです。このプロセスでは、アプリケーション所有者に、クラウドに簡単に移行できるアプリケーションを指名するよう依頼します。その後、ユーザーとアプリケーション所有者は、指定されたアプリケーションに関する深い知識をすでに持っているため、アプリケーションに迅速に優先順位を付けることができます。ワークショップ形式のセッションでステークホルダーと協力することをお勧めしますが、Eメール、共有ドキュメント、その他のコミュニケーションプラットフォームでコラボレーションすることもできます。

ノミネーションプロセス中に、[ポートフォリオプレイブック](#)テンプレートに含まれるアプリケーションの複雑さ (Excel 形式) のスコアシートテンプレートに、ノミネーションされたアプリケーションを入力します。このテンプレートのすべてのスコアリングおよび基準機能を使用するわけではありませんが、このシートを使用して、ノミネーションと優先順位付けの決定を記録することをお勧めします。

状況によっては、アプリケーションの指名プロセスを使用して優先順位付けを高速化し、スコアシートを必要としない場合があります。たとえば、少数のアプリケーションのみを優先する場合や、アプリケーション所有者がアプリケーションについて非常に精通している場合、アプリケーション所有者と利害関係者は、その知識と経験に基づいてアプリケーションを優先できます。この場合、スコアシートの使用をスキップし、直接優先順位付けに進むことができます。

アプリケーションの優先順位付けプロセスを次のように文書化します。

1. アプリケーションの優先順位付けランブックを開きます。
2. アプリケーションの複雑さのスコアリング基準セクションを削除します。このプロセスでは、アプリケーションの複雑さのスコアリングは使用されません。
3. アプリケーションの優先順位付けプロセスセクションで、次の操作を行います。
 - a. オプション 2: アプリケーションのノミネーションを保持し、他のオプションを削除します。
 - b. ユースケースに応じてプロセスを変更します。
 - c. Option という単語を含むこのセクションの見出しをすべて削除します。ランブックにこれらを残しておく、ユーザーがプロセスがオプションであるか、複数のオプションが利用可能であると考えるように混乱する可能性があります。
4. アプリケーションの優先順位付けランブックを保存します。

オプション 3: 検出ツール

検出ツールに複雑なスコアリングやアプリケーションの優先順位付けの機能がある場合、この自動プロセスに必要なリソースは少なく、アプリケーションの優先順位付けプロセスを加速できます。ユースケースの検出ツールで基準をカスタマイズすると、検出ツールが自動的にアプリケーションを分析し、最終的な複雑さスコアを提供します。ツールにはすでにすべてのアプリケーションメタデータがあるため、入力する必要はありません。

例えば、Flexera One Cloud Migration and Modernization (旧 Flexera Foundation および CloudScape) 検出ツールには、Optimization Scorecard と呼ばれる複雑なスコアリング機能があります。この機能を使用すると、スコアリングに含める基準を選択し、好みに基づいて各基準を重み付け

できます。データ検出が完了すると、検出ツールは指定した加重基準に基づいてデータを分析し、ツール独自の式を使用して最終的な複雑さスコアを生成します。詳細については、[Foundation and CloudScape ユーザーガイド](#) (Flexera ドキュメント) および [Optimization Scorecard](#) (Flexera ドキュメント) を参照してください。

このプロセスの欠点は、環境内のエージェントレス検出ツール用にスキャンアプライアンスをセットアップしたり、対象範囲内のすべてのワークロードにエージェントをインストールしたりするのに (4~8 週間) かかることです。検出ツールでスコアリング機能を使用する前に、アプリケーションワークロードをスキャンしてアプリケーションスタック分析を実行することで、検出ツールがメタデータを収集するための追加の時間 (4~12 週間) を確保する必要があります。ただし、メタデータの収集とアプリケーションの優先順位付けに必要な時間とリソースを減らすことで、検出ツールの設定に必要な余分な時間が復旧される可能性があります。例えば、検出ツールのデータがまだ最新である場合、ポートフォリオワークストリームは検出ツールとそのデータを動員フェーズから再利用してパイロットアプリケーションを特定できます。

Note

検出ツールプロセスを使用している場合でも、手動のスコアシートテンプレートを使用してアプリケーションの複雑さを分析できます。この追加情報は、アプリケーションの優先順位を絞り込むのに役立ちます。

アプリケーションの優先順位付けプロセスを次のように文書化します。

1. まだ設定していない場合は、環境で検出ツールをセットアップします。詳細については、AWS「規範ガイド」ウェブサイトの [「自動ポートフォリオ検出の開始方法」](#) を参照してください。
2. ツールの指示に従って、検出ツールで複雑なスコアリングまたはアプリケーションの優先順位付け基準をカスタマイズします。条件の選択の詳細については、「」を参照してください [複雑なスコアリング基準について](#)。
3. アプリケーションの優先順位付けランブックを開きます。
4. アプリケーションの複雑さのスコアリング基準セクションで、セクションを更新して、スコアリング基準が検出ツールで定義されていることを反映させます。例: 複雑なスコアリング基準は <検出ツール> で定義されています。
5. アプリケーションの優先順位付けプロセスセクションで、次の操作を行います。
 - a. オプション 3: 検出ツールを保持し、他のオプションを削除します。

- b. ユースケースに応じてプロセスを変更します。複雑さスコアを含むレポートを生成する方法の step-by-step の手順を含めることが重要です。利用可能な場合は、ユーザーガイドへのリンクを含めることができます。
 - c. Option という単語を含むこのセクションの見出しをすべて削除します。ランブックにこれらを残しておく、ユーザーがプロセスがオプションであるか、複数のオプションが利用可能であると考えられるように混乱する可能性があります。
6. アプリケーションの優先順位付けランブックを保存します。

ステップ 2: アプリケーションの優先順位付けルールを定義する

このステップでは、アプリケーションの優先順位付けルールを定義し、アプリケーションの移行順序を決定するのに役立ちます。アプリケーションの複雑さスコアはアプリケーションの優先順位付けやウェブの計画において重要な要素ですが、ビジネスやテクノロジーの要素も考慮する必要があります。各アプリケーションの優先度を評価し、適切なウェブでアプリケーションをスケジュールするのに役立つルールを作成します。一般的なビジネスルールとテクノロジールールは次のとおりです。

- データセンターの移行順序とスケジュールの指定
- ビジネスユニットの優先順位付け
- 重要なビジネスアプリケーションの期限を把握する

アプリケーションの優先順位付けルールを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. 「アプリケーションの優先順位付けルール」セクションで、移行用のカスタムルールを追加します。
3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックのルールを維持します。ルールは、移行の進行状況、範囲の変更、またはスケジュールのシフトに応じて変更される可能性があります。

以下は、アプリケーションの優先順位付けルールの例です。

優先度	ルール
1	ニューヨークデータセンターのアプリケーションは、常にテキサスデータセンターのアプリケーションよりも優先度が高くなります。
2	IT 部門のアプリケーションは、マーケティング部門のアプリケーションよりも常に優先順位が高くなります。
3	複雑さスコアが高いアプリケーションは、優先順位が高くなります。
4	SAP アプリケーションは、年末までに移行する必要があります。

ステップ 3: アプリケーションの優先順位付けプロセスを完了する

次に、ポートフォリオワークストリームがルールとプロセスを使用してアプリケーションの優先順位を付ける方法を定義します。これは、ポートフォリオワークストリームが移行の実装段階で参照するプロセスです。

アプリケーション優先順位付けランブックでこのプロセスを次のようにカスタマイズします。

1. アプリケーションの優先順位付けランブックを開きます。
2. ステージ 2: アプリケーションの優先順位付けセクションで、ユースケースと環境に応じてプロセスを変更します。
3. アプリケーションの優先順位付けランブックを保存します。

タスク終了条件

次のタスクを完了したら、次のタスクに進みます。

- 使用可能なオプションからアプリケーションの優先順位付けプロセスを選択しました。
- アプリケーションの優先順位付けランブックに以下を文書化しました。
 - アプリケーションの複雑さのスコアリング基準 (該当する場合)

- アプリケーションの優先順位付けプロセス
- アプリケーションの優先順位付けルール
- アプリケーションランブックの「ステージ 2: アプリケーションの優先順位付け」セクションを更新しました。

タスク 4: アプリケーションのディープダイブプロセスを定義する

アプリケーションの優先順位付けのルールとプロセスの確立が完了したら、各アプリケーションの優先順位を絞り込むのに役立つアプリケーションの詳細な説明を実行します。アプリケーションのディープダイブは、優先度が最も高いものから低いものまで、一度に 1 つのアプリケーションで実行します。複数のポートフォリオチームを持つプロジェクトの場合、各チームは異なるアプリケーションに対して同時にアプリケーションのディープダイブを実行できます。

詳細な分析中に、依存関係など、アプリケーションの移行の複雑さに影響する予期しない問題が発生する可能性があります。この場合、前のステップで定義した複雑さスコア基準を変更し、それに応じてスコアシートを更新して、将来のアプリケーションのより正確な複雑さスコアを取得する必要があります。その後、新しい複雑さスコアを使用してアプリケーションの優先順位を更新できます。

このタスクは、次のステップで構成されます。

- [ステップ 1: アプリケーションワークショッププロセスを定義する](#)
- [ステップ 2: アプリケーションマッピングプロセスを定義する](#)
- [ステップ 3: \(オプション\) アプリケーションターゲットの状態を定義する](#)
- [ステップ 4: アプリケーションのディープダイブプロセスを完了する](#)

ステップ 1: アプリケーションワークショッププロセスを定義する

ワークショッププロセスは、アプリケーションのディープダイブに対する最も効率的なアプローチの 1 つです。このプロセスを使用して、ステークホルダー、アプリケーション所有者、ポートフォリオチームと協力してアプリケーションを評価および分析します。目標は、アーキテクチャ、ビジネス目的、依存関係、環境など、アプリケーションの現在の状態を明確に把握することです。次に、アプリケーションのサイズと複雑さに関するこの詳細情報を使用して、アプリケーションのターゲット状態を設計します。

各ワークショップは通常 1~8 時間かかりますが、複雑性の高いアプリケーションにはさらに時間が必要になる場合があります。また、リソースの可用性、好み、アプリケーションのサイズと複雑さに応じて、ワークショップを複数の会議に分割することもできます。

期待される成果を特定する

アプリケーションワークショップを実施する前に、アジェンダを設定し、ワークショップの期待される成果、またはワークショップで収集する必要がある情報を定義する必要があります。これにより、ワークショップ参加者はワークショップの準備ができ、会議を目標に維持し、ワークショップの終了までに、アプリケーションの優先順位付け、ウェーブプラン、移行に必要なすべての情報を確実に取得できます。

期待される結果の標準セットを定義し、それらをアプリケーションの優先順位付けランブックに記録することをお勧めします。ワークショップを準備するときは、標準的な期待される結果を使用し、特定のアプリケーションに新しい結果を追加します。

アプリケーションワークショップで期待される成果の標準セットを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. アプリケーションワークショップの期待される成果セクションで、アプリケーションワークショップの期待される成果の標準セットを確立します。ワークショップを準備するときは、アプリケーションの特定のニーズに合わせてカスタマイズできます。
3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックで期待される成果を維持します。アプリケーションワークショップを実施し、ポートフォリオ評価とウェーブプランニングを継続すると、期待される新しい成果を特定したり、既存の成果を改善したりできます。

以下は、アプリケーションワークショップで期待される成果の例です。

優先度	アプリケーションワークショップの期待される成果
1	関連するサーバー、依存関係、環境、アプリケーション層など、アプリケーションの現在のアーキテクチャを明確に理解します。

優先度	アプリケーションワークショップの期待される成果
2	<p>チームは、ターゲットアーキテクチャの設計をサポートするためにメタデータを収集しました。次のメタデータが必要です。</p> <ul style="list-style-type: none"> • ターゲット AWS アカウント ID • ターゲット AWS リージョン • ターゲットサブネット • ターゲットセキュリティグループ
3	<p>アプリケーション所有者アンケートは完了し、すべての主要な質問に回答します。</p>
4	<p>チームは、ユーザーガイド、アプリケーションアーキテクチャドキュメント、テストドキュメント、設計ドキュメント、アプリケーションプログラミングインターフェイス (API) ドキュメントなど、すべてのアプリケーションドキュメントを収集しました。</p>

アプリケーションワークショップルールを定義する

アプリケーションワークショップを実施する前に、ワークショップを管理するルールを定義する必要があります。一般的なルールには、ワークショップの長さ、ワークショップで必要になる可能性のあるツール、考慮すべきスケジューリング上の考慮事項や期限などがあります。これにより、会議を目標に保ち、ワークショップで行われた決定が移行スケジュールに沿ったものになります。

アプリケーションワークショップルールは、アプリケーションの優先順位付けランブックに文書化することをお勧めします。

アプリケーションワークショップルールを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. アプリケーションワークショップルールセクションで、ワークショップを管理するルールを定義します。

3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックのルールを維持します。アプリケーションワークショップを実施し、ポートフォリオ評価とウェブプランニングを継続すると、新しいルールを特定したり、既存のルールを改善したりできます。

アプリケーションワークショップのルールの例を次に示します。

優先度	ワークショップルール
1	ワークショップは、火曜日と木曜日のセッションごとに最大 2 時間スケジュールする必要があります。
2	12 月 1 日から 1 月 15 日の間、インフラストラクチャのフリーズが予定されています。
3	移行ツールの実践的なワークショップがあります。
4	データセンター契約は 3 月 31 日に期限切れになります。ペナルティやコストのかかる契約延長を避けるため、ワークロードは 3 月 31 日までに退避する必要があります。
5	生体認証アプリケーションと時間および出席アプリケーションは保持されます。

アプリケーションワークショッププロセスを定義する

アプリケーションワークショップを実施するための標準プロセスを定義することが重要です。これにより、一貫したエクスペリエンスが保証され、ワークショップ参加者に期待が設定されます。これにより、ワークショップの効率が向上します。

アプリケーションワークショッププロセスには 3 つのステージがあります。

- **ワークショップの準備** – ワークショップの準備は、セッションがスムーズに進行し、参加者が期待される内容を把握するのに役立ちます。ワークショップの準備として、アジェンダを作成し、期待される成果を定義し、ワークショップに必要な参加者、ツール、情報を特定し、ワークショップ

をスケジュールします。少なくとも 1 週間前にワークショップをスケジュールすると、チームはカレンダーをブロックし、ワークショップの準備を整え、有用な情報を収集する時間を確保できます。

- **ワークショップを実施する** –ワークショップを実施するときは、ディスカッションをアジェンダの項目に限定し、期待される成果を達成するようにします。役に立ちますが、アジェンダには含まれていないと思われるトピックに注意してください。ワークショップを記録すると便利です。
- **ワークショップの結果をまとめる** –ワークショップの後、チームはアプリケーションの現在の状態と、優先順位付けと移行に影響を与える可能性のある潜在的な問題点、リスク、課題、ブロック要因を明確に理解する必要があります。ワークショップ後の一般的なタスクには、アプリケーションの優先順位の変更、アプリケーションの将来の状態の作成、次のワークショップに役立つ可能性のある期待される結果、ルール、またはプロセスの変更を含むランブックの更新が含まれます。

アプリケーションの優先順位付け用のランブックテンプレートには、アプリケーションワークショップの準備、実施、最終化のための標準的なstep-by-stepのプロセスが含まれています。アプリケーションワークショッププロセスを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. アプリケーションワークショッププロセスセクションで、ユースケースのニーズに合わせて標準プロセスを変更します。
3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックでプロセスを維持します。アプリケーションワークショップを実施する際に、このプロセスに加えたい変更を特定する場合があります。

ステップ終了基準

- ワorkshopのアジェンダと、ワークショップのサポートに必要なリソースとアーティファクトを定義しました。
- ワorkshopの期待される成果を定義し、ワークショップで収集する必要がある情報を特定しました。
- ワorkshopプロセスのトライアルを実施し、アプリケーションマッピングをサポートし、ターゲットの状態を設計するために必要な情報を取得しました。
- アプリケーションの優先順位付けランブックに以下を文書化しました。
 - アプリケーションワークショップの期待される成果
 - アプリケーションワークショップのルール

- アプリケーションワークシヨッププロセス

ステップ 2: アプリケーションマッピングプロセスを定義する

アプリケーションマッピングは、各アプリケーションを移行パターンに割り当てるプロセスであり、で特定して検証します [ステップ 4: 移行パターンを検証する](#)。このステップでは、アプリケーションの評価に使用するルールを定義します。次に、各アプリケーションの評価に使用するプロセスを定義します。各アプリケーションを移行パターンのユースケースにマッピングすることで、移行ツール、移行を完了するために必要なスキル、移行パターンの複雑さを特定できます。

必ずしもアプリケーションを 1 つのパターンに移行するわけではありません。同じアプリケーションに複数のパターンが必要な場合の詳細については、このセクションの [アプリケーションマッピングプロセスを定義する](#) 後半の「」を参照してください。

アプリケーションマッピングルール

アプリケーションマッピングルールは、アプリケーションを評価し、適切な移行パターンを特定するのに役立ちます。各ルールは、アプリケーションの評価とパターン的一致基準に使用する必要がある情報で構成されます。

[ポートフォリオプレイブックテンプレート](#) では、アプリケーションの優先順位付け用のランブックテンプレートに、アプリケーションマッピングルールを文書化するためのセクションが含まれています。プロセスを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. 「アプリケーションマッピングルール」セクションで、アプリケーションマッピングルールを定義します。
3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックのルールを維持します。

次の表に、アプリケーションマッピングルールの例を示します。

Note

この表のパターン IDs と名前は、のサンプルパターンに対応しています [ステップ 4: 移行パターンを検証する](#)。アプリケーションの優先順位付けランブックで定義したパターン IDs と名前を使用します。

優先度	マッピングルール
1	<p>使用率データまたはモニタリングツールを使用して、アプリケーションがゾンビアプリケーションかアイドルアプリケーションかを特定します。アプリケーションがゾンビまたはアイドル状態のアプリケーションの場合は、パターン 8: アプリケーションの廃止を選択し、アプリケーションスタック内のサーバーをシャットダウンします。</p>
2	<p>このアプリケーションをクラウドに移行するとビジネス価値が得られるかどうかを特定します。オンプレミスでのみ使用され、時間や出席アプリケーションなど、ブランチや地理的な場所間で共有されていないアプリケーションは、通常、クラウドに移行する必要はありません。このアプリケーションを移行してもビジネス価値が得られない場合は、パターン 9: オンプレミスで保持を選択します。</p>
3	<p>アプリケーションのオペレーティングシステム (OS) が AWS、AWS、移行サービス、ベンダー、またはリホスト移行ツールでサポートされているかどうかを確認し、以下を実行します。</p> <ul style="list-style-type: none">OS がサポートされている場合は、パターン 1: Application Migration Service または Cloud Migration Factory を使用して Amazon EC2 にリホストするを選択します。OS がサポートされていない場合は、を使用してパターン 3: Amazon EC2 へのリプラットフォーム CloudFormationを選択します。
4	<p>アプリケーションに Software as a Service (SaaS) バージョンまたは同等のものがあるか</p>

優先度	マッピングルール
	どうかを特定し、この新しいプラットフォームに移行するメリットとコストを評価します。これらの基準を満たしている場合は、パターン 10: 再購入を選択し、SaaS にアップグレードします。
5	オンプレミスの Oracle データベースを Amazon RDS for Oracle に移行する、オンプレミスの MySQL データベースを Amazon Aurora MySQL 互換エディションデータベースに移行するなど、アプリケーションのオンプレミスデータベース (複数可) をクラウド内の同種サービスに移行できるかどうかを特定します。これらの基準が満たされた場合は、パターン 2: AWS DMS とを使用して Amazon RDS にリプラットフォーム AWS SCTします。
6	アプリケーションが Microsoft .NET Core (.NET 5 または .NET 6)、Java、PHP、またはその他のオープンソースプログラミング言語を使用しているかどうか、およびアプリケーションが Microsoft Windows Server でホストされているかどうかを確認します。リプラットフォームのコストが正当かどうかを評価します。これらの条件が満たされた場合は、パターン 7: Amazon EC2 で Windows から Linux へのリプラットフォームを選択します。

優先度	マッピングルール
7	<p>アプリケーションが依存するオンプレミスのローカルファイルストレージと共有ファイルストレージを特定し、移行に含める必要があるかどうかを判断します。ローカルファイルストレージと共有ファイルストレージを移行する必要がある場合は、パターン 4: AWS DataSync または を使用して Amazon EFS にリプラットフォームする AWS Transfer Familyを選択します。</p>
8	<p>アプリケーションのサーバーが IBM AS/400 や Apache Spark などのメインフレームサーバーかミッドレンジサーバーかを特定し、アプリケーションがエミュレータと互換性があることを確認します。これらの基準を満たしている場合は、エミュレータを使用してパターン 6: メインフレームまたはミッドレンジサーバーを Amazon EC2 にリプラットフォームします。</p>
9	<p>これがレガシーアプリケーション、モノリシックアプリケーション、メインフレームベースのアプリケーションのいずれかで、その制限によりビジネスの需要を満たせないかを特定します。例えば、アプリケーションがスケーリングできるか、関連アプリケーションと統合できるか、またはコストが高く保守が難しいかを特定します。アプリケーションがこれらの基準のいずれかを満たしている場合は、パターン 11: アプリケーションを再設計するを選択します。</p>

アプリケーションマッピングプロセスを定義する

アプリケーションを移行パターンにマッピングする場合、検出チームからアプリケーションの検出データをリクエストすると便利です。このデータには、通常、推奨される移行パターン (R パターンまたは R スコアと呼ばれることもあります)、使用率情報、アプリケーションの依存関係、およ

び評価で使用できるその他の情報が含まれます。このアプリケーションを詳しく調べる際に、以前に特定した移行パターンを変更する場合があります。

データを取得したら、アプリケーションと、で特定したビジネスおよび技術基準を比較できます [ステップ 2: ビジネスと技術の推進要因を特定する](#)。アプリケーションの優先順位付けランブックにドライバーを記録しました。基準に照らしてアプリケーションを評価すると、アプリケーションの複数の移行パターンを選択したり、コスト、スケジュール、その他の制限に基づいてパターンを排除したりする可能性があります。

以下は、1つのアプリケーションで複数の移行パターンを使用する必要があるビジネスドライバーの例です。オンプレミスの SQL Server データベースを Amazon DynamoDB に移行したいが、データセンターの契約が期限切れになるため、ビジネスは提案されたスケジュールよりも早く移行してリプラットフォームしたいと考えています。このビジネスドライバーに対処するには、アプリケーションの移行計画を2パターンのアプローチに修正します。まず、アプリケーションをクラウドにリホストして、データセンターから削除します。その後、アプリケーションがクラウド内にあると、提案されたスケジュールに従ってアプリケーションを再プラットフォームできます。

また、アプリケーションが複数の階層で構成されるアプリケーションである n 階層アプリケーションであるのかも考慮する必要があります。アプリケーション層は、アプリケーションの水平レイヤーをホストする物理サーバーのグループです。N 層アプリケーションは、各層が異なる戦略を必要とする場合があります。アプリケーション層を異なるウェブに移行することを選択できるため、より複雑になります。たとえば、プレゼンテーション、ビジネスサービス、データベース階層で構成されるアプリケーションがある場合、階層ごとに異なるパターンをマッピングできる可能性があります。

次に、アプリケーション優先順位付けランブックで定義したアプリケーションマッピングルールに対してアプリケーションを評価します。詳細については、このセクションの前半の [アプリケーションマッピングルール](#) を参照してください。

アプリケーションを1つ以上のパターンにマッピングしたら、この決定を確認してアプリケーション所有者に確認します。アプリケーション所有者は選択したパターンを確認し、移行の計画と実行に役立ちます。現時点では、アプリケーション所有者は、経験に基づいてインサイトを提供したり、移行で予想される問題を共有したりすることもできます。

アプリケーションを1つ以上の移行パターンにマッピングし、アプリケーション所有者にパターンを確認したら、アプリケーション、パターン ID、パターン名、および関連するドライバーをアプリケーションの優先順位付けランブックのアプリケーションマッピングテーブルに記録します。

[ポートフォリオプレイブックテンプレート](#) では、アプリケーションの優先順位付け用のランブックテンプレートに、アプリケーションマッピングの標準的な step-by-step プロセスが含まれています。プロセスを次のように定義します。

1. アプリケーションの優先順位付けランブックを開きます。
2. アプリケーションワークショッププロセスセクションで、ユースケースのニーズに合わせて標準プロセスを変更します。
3. アプリケーションの優先順位付けランブックを保存します。
4. アプリケーションの優先順位付けランブックでプロセスを維持します。

次の表は、アプリケーションマッピングテーブルの例です。アプリケーションの優先順位付け用に用意されているランブックテンプレートには、アプリケーションマッピングプロセスの結果を記録できる空のテーブルが含まれています。

Note

このテーブルのパターン IDs と名前は、のサンプルパターンに対応しています [ステップ 4: 移行パターンを検証する](#)。アプリケーションの優先順位付けランブックで定義したパターン IDs と名前を使用します。

アプリケーション名	パターン ID	パターン名	対処されたビジネスおよび技術の推進要因
企業ウェブサイト	1	Application Migration Service または Cloud Migration Factory を使用して Amazon EC2 にリホストする	<ul style="list-style-type: none"> • データセンターの終了 • 運用コストの削減
HR システム	8	アプリケーションを廃止します。	<ul style="list-style-type: none"> • 運用コストの削減
時間と出席申請	9	オンプレミスで保持する	<ul style="list-style-type: none"> • 運用コストの削減 • リスクと影響の軽減
PO システム	3	を使用した Amazon EC2 へのリプラットフォーム	<ul style="list-style-type: none"> • テクノロジー統合

アプリケーション名	パターン ID	パターン名	対処されたビジネスおよび技術の推進要因
		フォーム CloudFormation	<ul style="list-style-type: none"> ストレージとコンピューティングの制限 ハードウェアとソフトウェアのend-of-life セキュリティとコンプライアンスの向上
CRM システム	10	SaaS の再購入とアップグレード	<ul style="list-style-type: none"> 運用コストの削減 テクノロジー統合 ハードウェアとソフトウェアのend-of-life 開発、イノベーション、成長を加速する
固定アセットシステム	7	Amazon EC2 での Windows から Linux へのリプラットフォーム	<ul style="list-style-type: none"> 運用コストの削減
ERP ファイルストレージ	4	AWS DataSync または を使用して Amazon EFS にリプラットフォームする AWS Transfer Family	<ul style="list-style-type: none"> ストレージとコンピューティングの制限

アプリケーション名	パターン ID	パターン名	対処されたビジネスおよび技術の推進要因
台帳システム	6	エミュレータを使用してメインフレームまたはミッドレンジサーバーを Amazon EC2 にリホストする	<ul style="list-style-type: none">• データセンターの終了• テクノロジー統合• セキュリティとコンプライアンスの向上• ハードウェアとソフトウェアのend-of-life• ストレージとコンピューティングの制限• アプリケーションアーキテクチャのモダナイズ

アプリケーション名	パターン ID	パターン名	対処されたビジネスおよび技術の推進要因
総台帳	11	アプリケーションを再設計する	<ul style="list-style-type: none"> 運用コストの削減 テクノロジー統合 セキュリティとコンプライアンスの向上 ハードウェアとソフトウェアのend-of-life ストレージとコンピューティングの制限 アプリケーションアーキテクチャのモダナイズ スケーラビリティと耐障害性 開発、イノベーション、成長を加速する

ステップ終了基準

- アプリケーションの優先順位付けランブックに以下を文書化しました。
 - アプリケーションマッピングルール
 - アプリケーションマッピングプロセス
- 1 つ以上のproof-of-concept (POC) アプリケーションを使用して、マッピングルールとプロセスを検証しました。

ステップ 3: (オプション) アプリケーションターゲットの状態を定義する

このステップでは、アプリケーションのターゲット状態または to-be 状態をドキュメント化するために使用する属性とプロセスを定義します。ターゲットの状態は、移行後のアプリケーションがターゲットクラウド環境でどのように動作するかです。ターゲット環境は、ターゲットプラットフォームまたはサービスやビジネス要件によって異なります。ターゲット環境は、AWS クラウドまたは AWS Managed Services (AMS) です。

ターゲットの状態を定義すると、プロジェクトマネージャー、移行コンサルタント、アーキテクト、アプリケーション所有者、ステークホルダーが効果的にコラボレーションできるようになります。このプロセスを使用することで、チームは事前に問題を特定して解決し、ターゲット状態環境をより効率的に実装できます。

一部のアプリケーションでは、このステップはオプションです。移行するアプリケーションがスタンダードアロンまたは複雑さが低い場合は、このステップをスキップできます。リホストなど、アプリケーションを変更しない移行戦略では、このステップを必要としない場合があります。ただし、リプラットフォームや再設計など、より複雑な移行戦略の場合は、移行を開始する前にターゲットの状態を定義する必要があります。

このワークショップでは、アプリケーションの現在の状態を深く理解できるため、ワークショップの完了後にターゲット状態をドラフトすることをお勧めします。さらに、アプリケーションを移行パターンにマッピングすると、追加のインサイトが得られ、ターゲット状態の定義が必要かどうかを特定できます。例えば、Application Migration Service または Cloud Migration Factory を使用してアプリケーションを Amazon EC2 Rehost にマッピングすると、戦略がリホストであることが特定され、このアプリケーションのターゲット状態を定義する必要がない可能性があります。

ターゲット状態属性

ターゲット状態を定義し、アプリケーションに関する決定を行うときは、次のターゲット状態属性を検討することをお勧めします。

- AWS Well-Architected Tool — AWS Well-Architected フレームワークに対してアプリケーションターゲットの状態を確認し、クラウド内のアプリケーションのセキュリティ、パフォーマンス、耐障害性を向上させます。
- ターゲットランディングゾーン – 通常、[動員フェーズ](#)の終わりまでに、パイロットアプリケーションを実行する準備ができていない完全なランディングゾーンを構築しておく必要があります。ランディングゾーンは、マルチアカウントアーキテクチャ、ID とアクセスの管理、ガバナンス、データセキュリティ、ネットワーク設計、ログ記録で設定済みである必要があります。パイロットアプリケーションを使用して、ランディングゾーンが完了したことを確認します。既存のターゲット

トランディングゾーンでパイロットアプリケーションを起動して実行できることを確認します。アプリケーションのランディングゾーンを変更する必要がある場合は、ランディングゾーンチームに要件を通知します。たとえば、アプリケーションが別のアカウントでホストされているサービスへのアクセスを必要とする場合や、アプリケーションが Virtual Private Cloud (VPC) への特別なルーティングを必要とする場合があります。

- 依存関係 – アプリケーションが適切に機能するために依存しているアプリケーションを特定します。たとえば、アプリケーションがデータベース、ストレージ、または支払いゲートウェイや外部ウェブサービスなどのサードパーティーのサービスに依存している場合や、アプリケーションが環境内の他のアプリケーションに依存している場合などです。これらの依存関係にアクセスするには、認証のためにディレクトリサービスに接続するなど、特別なルーティングまたは設定が必要になる場合があります。
- 依存アプリケーション – 正常に機能するためにアプリケーションに依存するアプリケーションを特定します。移行中のダウンタイムを防ぐために、これらのアプリケーションを再設定して更新する必要があります。
- セキュリティとコンプライアンス – セキュリティとコンプライアンスチームとともにターゲット環境を確認し、ギャップを特定します。アプリケーションは、複数のコンポーネント、論理レイヤー、または複数の階層で構成されます。コンプライアンス要件によっては、これらのコンポーネントの一部をターゲット環境に移行できない場合や、ワークロードを移行するときに追加のセキュリティ対策が必要になる場合があります。セキュリティとコンプライアンスに関する一般的な要件は、データレジデンシー、転送中の暗号化、保管中の暗号化です。これには、ターゲット環境で追加の設定が必要です。例えば、通信を保護するために証明書を設定する必要がある場合や、保管中のデータを保護するために暗号化キーが必要な場合があります。また、一部のアプリケーション層がオンプレミスのままで、他の層がクラウドに移行されるように、アプリケーションの複数の移行パターンを選択する必要がある場合もあります。
- ストレージの依存関係 – アプリケーションストレージの依存関係を確認し、アプリケーションをターゲット環境に移行すると、これらの依存関係にどのように影響するかを判断します。たとえば、アプリケーションがネットワーク接続ストレージ (NAS) やストレージエリアネットワーク (SAN) などのネットワークストレージに依存している場合は、Amazon Simple Storage Service (Amazon S3) や Amazon FSx などのクラウド内のストレージサービスを計画する必要があります。また、アプリケーションの実行を維持するために、ターゲットクラウド環境にデータを移行する方法を計画する必要があります。
- データベース – アプリケーションが使用するデータベースの移行戦略を確認します。Amazon Relational Database Service (Amazon RDS)、Amazon Aurora、Amazon DynamoDB などの新しいデータベースサービスにリプラットフォームしますか？ ターゲット環境でデータベースをリホストしますか？ 場合によっては、特にモノリシックデータベースの場合、1 秒未満のレイテンシーなど

の技術的要件に対処したり、特定のタイプのデータベースの機能を活用したりするために、AWS データベースをリファクタリングする必要があります。データレジデンシーのコンプライアンス要件と同様に、オンプレミスで保持すべきデータとクラウドに移行すべきデータを特定する必要があります。たとえば、顧客情報用にオンプレミスのデータベーステーブルを保持する必要があり、残りのデータベースをクラウドに移行できます。

- **アプリケーションコンポーネント** – アプリケーションが依存するコンポーネントを確認します。ターゲット環境でサポートされていないコンポーネントにアプリケーションが依存しているかどうかを判断します。ターゲット環境がすべてのアプリケーションコンポーネントをサポートしていない場合は、問題を軽減するためにアプリケーションをリファクタリングする必要があります。たとえば、コンポーネントオブジェクトモデル (COM) Interop、COM+、Windows レジストリなどの Windows 専用コンポーネントに依存する .NET Framework アプリケーションがある場合、Linux オペレーティングシステムでアプリケーションをリプラットフォームするには、アプリケーションを .NET Core にリファクタリングする必要があります。
- **アプリケーション階層** – アプリケーション内の階層の数を特定します。アプリケーションは n 層、2 層、またはスタンドアロンですか？ 各階層の移行パターンを理解していることを確認します。たとえば、アプリケーションには、ユーザーインターフェイスをホストするプレゼンテーション (またはウェブ) 階層、ビジネスサービスをホストするアプリケーション階層、データベースをホストするデータベース階層があり、各階層には異なる移行パターンが必要になる場合があります。
- **ディザスタリカバリ** – 目標復旧時点 (RPO) や目標復旧時間 (RTO) など、アプリケーションの現在および将来のディザスタリカバリ (DR) 計画を特定します。既存のオンプレミス DR プランを使用するか、クラウドで新しい DR 戦略を検討するかを決定します。詳細については、[クラウドホワイトペーパーの「ワークロードのディザスタリカバリ」の「ディザスタリカバリオプション」と「リカバリ目標 \(RTO と RPO\)」](#)を参照してください。 AWS

ターゲット状態プロセスを定義する

アプリケーションターゲットの状態を定義するには、提供されたテンプレートであるアプリケーションターゲット状態ワークシート (Excel 形式) を使用することをお勧めします。このワークシートは、[ポートフォリオプレイブックテンプレート](#)で入手できます。テンプレートには、使用または変更できる標準属性が含まれています。アプリケーションターゲットの状態を次のように文書化するプロセスを定義します。

1. アプリケーションターゲットの状態ワークシートを開きます。
2. デフォルトの属性を確認し、ユースケースに変更を加えます。
3. ワークシートを保存します。

4. アプリケーションの優先順位付けランブックを開きます。
5. ターゲットアプリケーションの状態セクションで、次の操作を行います。
 - a. このプロセスをいつ完了するかセクションで、ポートフォリオチームがアプリケーションのターゲット状態を定義する必要があるかどうかを判断できるようにする基準を確立します。
 - b. 必要に応じて属性セクションを更新します。
 - c. ユースケースに応じてプロセスセクションを更新します。
6. アプリケーションの優先順位付けランブックを保存します。

アプリケーションターゲットの状態のサンプル

次の表は、アプリケーションターゲット状態ワークシートを使用してアプリケーションのターゲット状態を文書化する方法の例を示しています。

アプリケーション	例
ターゲットプラットフォーム	AWS クラウド
ランディングゾーン	オンプレミスディレクトリサービスへのアクセスが必要 組織全体の複数のアカウントとサービスの管理を一元化 AWS Control Tower する必要があります
依存関係	Active Directory、支払いゲートウェイ、インベントリシステム
依存アプリケーション	なし
セキュリティ	保管中と転送中の暗号化
コンプライアンス	PCI、SOC
ストレージの依存関係	ブートドライブのアタッチ、NAS、ネットワーク共有
データベース	最新: Oracle DB

アプリケーション	例
	クラウド: Amazon RDS for Oracle
アプリケーションコンポーネント	.NET Framework 4.5
アプリケーション層	N 層 フロントエンド、ビジネスサービス、データサービスとエージェント、データベース
ディザスタリカバリ	RPO: 1 分、RTO: 5 分 現在の DR 戦略はウォームスタンバイです 任意の米国リージョンの DR

ステップ終了基準

- アプリケーションターゲット状態ワークシートで、ターゲット状態プロセスに含める属性を定義しました。
- アプリケーションの優先順位付けランブックでは、以下を実行しました。
 - ポートフォリオチームがアプリケーションのターゲット状態を定義することが予想されるタイミングの基準を確立している。
 - ユースケースに基づいてターゲット状態を定義するプロセスを更新しました。

ステップ 4: アプリケーションのディープダイブプロセスを完了する

次に、ポートフォリオワークストリームが、このタスクで確立したワークシヨップ、ルール、プロセスを使用して、アプリケーションについて詳しく説明します。これは、ポートフォリオワークストリームが移行の実装段階で参照するプロセスです。

アプリケーション優先順位付けランブックでこのプロセスを次のようにカスタマイズします。

1. アプリケーションの優先順位付けランブックを開きます。
2. ステージ 2: アプリケーションの詳細セクションで、ユースケースと環境に応じてプロセスを変更します。
3. アプリケーションの優先順位付けランブックを保存します。

4. レビューのために、アプリケーションの優先順位付けランブックをチームと共有します。

タスク 5: ウェーブプランニングプロセスの定義

ウェーブプランニングは、大規模な移行における重要なマイルストーンです。ウェーブプランでは、インフラストラクチャとアプリケーションの依存関係 (共有データベースなど)、アプリケーションの優先度、アプリケーションアーキテクチャの類似性、ビジネス機能を考慮して、同様のアプリケーションをグループ化します。次に、アプリケーションチームおよびインフラストラクチャチームとともにウェーブプランを確認し、指定された移行およびカットオーバー期間中の可用性を確認します。

さまざまな AWS お客様の実際のデプロイに基づいて、ウェーブプランニングのベストプラクティスを以下に示します。

- 少なくとも 4~5 波の移行ウェーブを計画します。これにより、移行ワークストリームに常に十分なサーバーを確保できます。
- フェイルファスト。いくつかの低複雑さのアプリケーションから始めて、学習内容を後の波に適用する必要があります。
- 早期ウェーブ (ウェーブ 1~5) では、少数のサーバー (10 未満)、低複雑さのアプリケーション、および開発環境やテスト環境などの低頻度の環境のアプリケーションを選択します。進行するにつれて、段階的に複雑になり、より多くのサーバーがウェーブに導入されます。
- ウェーブプランニングは継続的なプロセスであり、1 回限りのタスクではありません。すべてのウェーブを一度に計画しないでください。
- ポートフォリオ検出ツールを使用していて、複雑なスコアリング機能がある場合は、ウェーブプランニングで使用します。複雑さが最も低いアプリケーションを最初に移行します。

このタスクは、次のステップで構成されます。

- [ステップ 1: 移動グループプロセスを定義する](#)
- [ステップ 2: ウェーブプランニングの選択基準を定義する](#)
- [ステップ 3: ウェーブプランニングプロセスを完了する](#)

ステップ 1: 移動グループプロセスを定義する

このステップでは、application-to-server 依存関係を特定し、移動グループとして一緒に移動するサーバーを決定するために使用されるルールを定義します。移動グループは、グループ内で一緒に移

動する必要があるサーバーまたはアプリケーションのブロックです。これは移行ウェブの構成要素であり、各ウェブは各移動グループのサーバー数に応じて 1 つ以上の移動グループで構成されます。

アプリケーションの依存関係を特定する

移動グループに相互依存するアプリケーションをグループ化する際の重要な考慮事項を次に示します。

- 次のようなインフラストラクチャの依存関係を考慮します。
 - アプリケーションには複数のデータベースがあり、それらのデータベースは他のアプリケーションと共有できます。
 - アプリケーションは別のアプリケーションに依存している可能性があります。
 - サーバーは、複数のアプリケーションのデータベースをホストする場合があります。
- 次のようなビジネスおよび運用上の依存関係を考慮します。
 - ビジネスへの影響や運用スケジュール (バックアップやパッチ適用など) により、アプリケーションは特定の期間にのみ移行できます。
 - アプリケーション所有者は 1 つの移行カットオーバーウィンドウにのみ使用できるため、所有者のすべてのアプリケーションが同じ移動グループに存在する必要があります。

アプリケーションワークショッププロセスまたはターゲット状態を定義したときに、インフラストラクチャの依存関係を特定しました。自動プロセスまたは手動プロセスを使用して、インフラストラクチャの依存関係を特定できます。インフラストラクチャの依存関係の識別を自動化するには、Flexera One Cloud Migration and Modernization や TDS TransitionManager などの検出ツールを使用できます。手動プロセスの場合は、アプリケーションチームとインフラストラクチャチームで CMDB 情報を検証します。

アプリケーションワークショッププロセスでビジネスと運用の依存関係を特定しました。

独自のウェブプランニングランブックを構築するための出発点として、ポートフォリオプレイブックテンプレートに含まれているウェブプランニング (Microsoft Word 形式) に Runbook テンプレートを使用することをお勧めします。 [samples/portfolio-playbook-templates.zip](#) 移行の依存関係を次のように文書化します。

1. ウェブプランニングランブックを開きます。
2. アプリケーションの依存関係 セクションで、依存関係を記録します。タイプ (インフラストラクチャ、ビジネス、または運用)、依存関係、依存関係の簡単な説明を特定します。

3. ウェーブプランニングランブックを保存します。
4. ウェーブプランニングランブックの依存関係を維持します。進行するにつれて、新しい依存関係を特定できます。

次の表は、依存関係の例を示しています。

タイプ	依存関係	説明
インフラストラクチャ	データベース	データベースが他のアプリケーションと共有されている
インフラストラクチャ	ファイルストア	アプリケーションは、デカップリングできる中央ファイルストアを使用するか、関連するすべてのアプリケーションと一緒に移行する必要があります。
インフラストラクチャ	アプリケーション	アプリケーションは、抽出、変換、ロード (ETL) ジョブなど、機能する他の 1 つ以上のアプリケーションに依存します。
ビジネス	ビジネスの停止	アプリケーションの特定の承認済み停止ウィンドウ
運用中	パッチウィンドウ	移行カットオーバーに影響を与える可能性のあるパッチ適用などのスケジュールされた運用タスク

移動グループルールを定義する

ウェーブプランニングランブックに依存関係を記録したら、それらの依存関係に基づいて移動グループルールを構築する必要があります。これらのルールは、サーバーを移動グループにグループ化する方法を制御します。ルールを構築するには、次の手順に従います。

1. 前のセクションで定義した依存関係を確認します。
2. アプリケーションを移動グループ内で一緒に移動する必要があるかどうかに影響する依存関係を選択します。すべての依存関係でアプリケーションを一緒に移行する必要があるわけではありません。たとえば、Microsoft Active Directory へのインフラストラクチャの依存関係は、すべてのアプリケーションに共通する依存関係であるため、移動グループを定義するときに考慮しないでください。アプリケーションを移行する前に、クラウドにドメインコントローラーを構築する必要があります。
3. アプリケーションを一緒に移動する必要がある依存関係を移動グループルールに変換します。

アプリケーションがいずれかのルールに一致する場合、関連付けられたすべてのサーバーを同じ移動グループに配置して、一緒に移行する必要があります。

移行の移動グループルールを次のように文書化します。

1. ウェブプランニングランブックを開きます。
2. グループルールの移動セクションで、移動グループルールを優先度順に記録します。
3. ウェブプランニングランブックを保存します。
4. ウェブプランニングランブックのルールを維持します。進行するにつれて、新しいルールを特定できます。

次の表は、移動グループルールの例を示しています。

ルール	グループルールの移動
1	共有データベースを持つアプリケーションは一緒に移行する必要があります。
2	同じアプリケーション所有者を持つアプリケーションは、一緒に移行する必要があります。
3	同じパッチウィンドウを持つアプリケーションは一緒に移行する必要があります。

ステップ 2: ウェーブプランニングの選択基準を定義する

移動グループを確立したら、移行ウェーブを形成するために、同様の移動グループをまとめて収集する必要があります。このステップでは、各ウェーブに対して 1 つ以上の移動グループを選択するために使用する基準を定義します。

ウェーブプランニングを成功させるには、各移動グループのサイズを理解することが不可欠です。目標は、移行が俊敏性を維持し、サーバーの正常なパイプラインを維持するために、各ウェーブのサイズを調整することです。大きすぎるウェーブは移行計画の変更に適応するのが難しく、小さすぎるウェーブは、必要な移行速度を実現するのに十分なサーバーを提供しない可能性があります。

ウェーブのサイズを設定するときは、次の基準を考慮することをお勧めします。

- **小さな最初の波** – 初期波は 10 台未満のサーバーで小さくする必要があります。その後、各波のサーバー数を徐々に増やすことができます。これにより、迅速に失敗し、学習した教訓に基づいて構築できます。たとえば、20 台のサーバーでアプリケーションを移行する前に、3 台のサーバーでアプリケーションを移行します。
- **リソース** – 移行チームが 1 つのウェーブで移行できるサーバーの数を特定します。標準的な対策は、4 人のアーキテクトの移行チームが、リホストパターンのために 1 週間に最大 50 台のサーバーを移行できることです。移動グループを組み合わせ、移行チームのキャパシティを超えない移行ウェーブを形成します。
- **俊敏性** – ウェーブは移行計画の変更に適応する必要があります。サーバーを再スケジュールする必要がある場合は、影響を受けるサーバーの移動グループ全体を再スケジュールできます。
- **ストレージサイズ** – 小さいアプリケーションを最初に移行します。たとえば、2 TB のアプリケーションの前に 100 GB のアプリケーションを移行します。
- **アプリケーション環境** – 開発環境やテスト環境などの下位環境のアプリケーションを、本番環境のアプリケーションに移行する前に移行します。
- **アプリケーションの複雑さ** – 外部依存関係が少なく、複雑さの少ないアプリケーションを最初に移行します。
- **アプリケーションの重要度** – ミッションクリティカルなアプリケーションの前に、重要でないアプリケーションを移行します。
- **ユーザーベース** – ユーザーベースが小さいアプリケーションを最初に移行します。たとえば、10,000 人のユーザーを持つアプリケーションの前に、10 人のユーザーを持つアプリケーションを移行します。

- ネットワーク帯域幅 – ウェーブのサイズはネットワーク帯域幅を超えることはできません。詳細については、[AWS「大規模な移行のための Foundation プレイブック」の指示に従って定義した移行原則](#)を参照してください。

ウェーブプランニングの選択基準を次のように文書化します。

1. ウェーブプランニングランブックを開きます。
2. 「Wave planning selection criteria」セクションで、移行に使用する基準を記録します。
3. ウェーブプランニングランブックを保存します。
4. ウェーブプランニングランブックの基準を維持します。進行するにつれて、条件を調整するか、新しい条件を追加する必要がある場合があります。

次の表は、ウェーブプランニング選択基準の例を示しています。

条件	説明
最も複雑でないアプリケーションを特定する	移動グループの複雑さスコアが高いアプリケーションを特定します。
まず環境を低くする	開発環境やテスト環境など、下位環境内の重要でないアプリケーションは、最初に移行する必要があります。収益を生み出すアプリケーションなど、本番環境内の重要なアプリケーションは、最後に移行する必要があります。
フェイルファスト	10 台未満のサーバーで初期ウェーブを形成します。
移行チームの強み	各移行チームがカットオーバーできるサーバーの数を特定します。
同様の移動グループを組み合わせる	共通性に基づいて移動グループを結合します。たとえば、移動グループは、同じアプリケーション所有者、ソースデータセンター、またはターゲット AWS アカウントを共有する場合があります。

条件	説明
波のサイズ	ウェーブの合計は 50 サーバーを超えることはできません。

ステップ終了基準

- ユースケースのウェーブプランニング基準を特定し、ウェーブプランニングランブックに文書化しました。

ステップ 3: ウェーブプランニングプロセスを完了する

移動グループの作成方法を定義し、移動グループを移行ウェーブに結合するために使用する基準を確立したので、ウェーブを計画するプロセスを定義する必要があります。このステップでは、ウェーブプランニングランブックを更新してウェーブプランニングプロセス全体を記録し、チームがウェーブ情報の記録に使用できるダッシュボードツールがあることを確認します。

このステップでは、提供された Dashboard テンプレートをウェーブプランニングと移行に使用することをお勧めします。このテンプレートは、[ポートフォリオプレイブックテンプレート](#)で入手できます。このテンプレートは、ポートフォリオチームを支援するように設計されており、データを照合するための出発点として機能し、アプリケーションポートフォリオの分析、application-to-server 依存関係の特定、最終的には移行ウェーブの計画に役立ちます。このテンプレートは、環境に応じて変更できます。

ウェーブプランニングプロセスを次のように文書化します。

1. ウェーブ計画と移行用の Dashboard テンプレートを開きます。
2. ユースケースに応じてダッシュボードを変更します。たとえば、サーバーインベントリを抽出するためのワークシートを追加したり、新しいピボットテーブルやグラフを追加したり、VLOOKUP関数を使用してソース情報をインポートしたりできます。
3. ダッシュボードテンプレートを保存します。
4. ウェーブプランニングランブックを開きます。
5. ステージ 2: ウェーブプランニングを実行するセクションで、ユースケースのニーズに合わせて提供される標準プロセスを変更します。
6. ウェーブプランニングランブックを保存します。
7. ウェーブプランニングランブックをチームと共有してレビューします。

8. ウェーブプランニングランブックでプロセスを維持します。このプロセスは、大規模な移行のウェーブを計画するための標準的な運用手順として機能します。

タスク終了条件

- ウェーブプランニングランブックに以下を文書化しました。
 - アプリケーション依存関係
 - 優先度順にリストされたアプリケーション移動グループルール
 - ウェーブプランニングの選択基準
 - ウェーブプランニングプロセス

ステージ 2: 大規模な移行の実装

大規模な移行を初期化するステージ 1 では、ポートフォリオ評価とウェーブプランニングのプロセスを定義し、ランブックに文書化しました。大規模な移行を実装するステージ 2 では、これらのプロセスを完了し、移行が完了するまでスプリントごとにこのプロセスを繰り返します。

ポートフォリオチームは、ステージ 2 で次のポートフォリオ評価とウェーブプランニングタスクを完了します。

- [タスク 1: アプリケーションの優先順位付け](#)
- [タスク 2: アプリケーションの詳細な分析を実行する](#)
- [タスク 3: ウェーブプランニングとメタデータ収集の実行](#)

Note

ポートフォリオの評価とウェーブプランニングは 1 回限りのタスクではありません。これは、移行をサポートする継続的なタスクです。移行が完了するまで、このステージのすべてのタスクを何度も繰り返します。

ポートフォリオの評価とウェーブ計画プロセスには、通常、ウェーブごとに 1~2 週間かかります。ポートフォリオワークストリームは通常、移行ワークストリームのサーバーの正常なパイプラインを維持するために、4~5 ウェーブを事前に計画します。ポートフォリオワークストリームは、初期化ステージ (ステージ 1) の最後にウェーブの計画を開始し、実装ステージ (ステージ 2) は、移行ワークストリームがアプリケーションの第 1 ウェーブの移行を開始したときに開始されます。ウェーブスケジュールの例については、[「大規模移行ガイド」の「ステージ 2: 大規模移行の実装 AWS」](#)を参照してください。

進行状況の追跡

移行のためのウェーブの準備を始める際には、ポートフォリオ評価プロセスを通じて各アプリケーションのステータスを追跡することをお勧めします。[ポートフォリオプレイブックテンプレート](#)では、ポートフォリオ評価に進行状況追跡テンプレートを使用できます (Microsoft Excel 形式)。このテンプレートを使用すると、各アプリケーションについて、複雑さスコア、ターゲットウェーブ、アプリケーション所有者、主要タスクのターゲット完了日 (アプリケーションの優先順位付け、詳細分

析、ウェーブ計画、データ収集)、移行のためのアプリケーションの全体的な準備状況を追跡できません。このプレイブックのガイドには、進行状況追跡シートを更新するタイミングに関する手順が含まれています。

タスク 1: アプリケーションの優先順位付け

このタスクでは、ポートフォリオ内の移行されていないアプリケーションのリストを確認し、残りのアプリケーションのサブセットにアプリケーションの複雑さスコアと優先度を割り当てます。移行プロジェクト全体でこのプロセスを何度も繰り返します。

このタスクを完了するには、次の情報が必要です。

Input	ソース
移行するポートフォリオ内のアプリケーションの完全なリスト	検出ツールまたは設定管理データベース (CMDB)
ターゲット移行戦略とパターンの概要	アプリケーションの優先順位付けランブックの移行戦略と移行パターン
ウェーブに含める予定のアプリケーションの数	ウェーブプランニングランブックのウェーブプランニング選択基準

「ステージ 2: アプリケーションの優先順位付け」セクションの「アプリケーションの優先順位付けランブック」の指示に従ってください。このプロセスは、このプレイブックの「」で定義しました [ステップ 3: アプリケーションの優先順位付けプロセスを完了する](#)。

このタスクの最後に、以下を完了しました。

Output	説明
優先アプリケーションのリスト	ウェーブに含める予定のアプリケーションの数の 2~3 倍を優先し、これらのアプリケーションを進行状況トラッカーに入力しました。

タスク 2: アプリケーションの詳細な分析を実行する

このタスクでは、前のタスクで優先順位付けした各アプリケーションについて詳しく説明します。これには、通常、アプリケーション所有者へのアンケートの送信、アプリケーションの依存関係の分析、アプリケーションワークショップのスケジュールが含まれます。

このタスクを完了するには、次の情報が必要です。

Input	ソース
優先アプリケーションのリスト	実装ステージの前半、に作成 タスク 1: アプリケーションの優先順位付け
ターゲット移行戦略とパターンの概要	アプリケーションの優先順位付けランブックの移行戦略と移行パターン

「ステージ 2: アプリケーションの詳細な分析を実行する」セクションの「アプリケーションの優先順位付けランブック」の指示に従います。このプロセスは、このプレイブックの「」で定義しました [ステップ 4: アプリケーションのディープダイブプロセスを完了する](#)。

このタスクの最後に、以下を完了しました。

Output	説明
移行パターンマッピング	各アプリケーションを移行パターンにマッピングしました。
アプリケーションターゲットの状態 (該当する場合)	アプリケーションに該当する場合は、クラウド内のアプリケーションの将来の状態を定義しています。

タスク 3: ウェーブプランニングとメタデータ収集の実行

これは、ポートフォリオ評価とウェーブプランニングの最終タスクです。このタスクでは、アプリケーション情報とターゲット移行パターンを使用して、移動グループを構築し、移動グループをウェーブに割り当て、移行のサポートに必要なすべてのメタデータを収集します。最後に、ウェーブの準備ができたことを移行ワークストリームに通知します。

このタスクを完了するには、次の情報が必要です。

Input	ソース
優先順位付けされたアプリケーションのリスト	実装ステージの前半、で作成 タスク 1: アプリケーションの優先順位付け
移行パターンマッピング	実装ステージの前半、で作成 タスク 2: アプリケーションの詳細な分析を実行する
アプリケーションターゲットの状態 (該当する場合)	でも作成 タスク 2: アプリケーションの詳細な分析を実行する

以下の操作を実行します。

- 「ステージ 2: ウェーブプランニングの実行」セクションの「ウェーブプランニングランブック」の指示に従ってください。このプロセスは、このプレイブックの「」で定義しました [ステップ 3: ウェーブプランニングプロセスを完了する](#)。
- 「ステージ 2: メタデータを収集する」セクションのメタデータ管理ランブックの指示に従います。このプロセスは、このプレイブックの「」で定義しました [ステップ 3: ランブックにメタデータ要件と収集プロセスを文書化する](#)。
- ウェーブプランが完了し、メタデータの準備が整ったことを移行ワークストリームに通知します。このコミュニケーションは、[AWS 大規模な移行のプロジェクトガバナンスプレイブック](#)で定義した [ガバナンス](#) に従う必要があります。

このタスクの最後に、以下を完了しました。

Output	説明
ウェーブプラン	ウェーブを計画し、そのウェーブ内のサーバー、アプリケーション、データベースを特定し、開始日とカットオーバー日時を定義しました。

Output	説明
ソースインフラストラクチャメタデータ	サーバー名やオペレーティングシステムなどのソースインフラストラクチャメタデータを収集しました。
ターゲットインフラストラクチャメタデータ	ターゲットサブネット、セキュリティグループ、AWS アカウントなどのターゲットインフラストラクチャメタデータを収集しました。
通知が完了しました	ウェーブプランとメタデータの準備ができたことを移行ワークストリームに通知しました。

ポートフォリオチームは、移行プロジェクトが完了するまで、スプリントごとにこのステージの3つのタスクをすべて繰り返します。

リソース

AWS 大規模な移行

大規模な移行に関する AWS 完全な 規範ガイドシリーズにアクセスするには、[「への大規模な移行 AWS クラウド」](#)を参照してください。

その他のリファレンス

ツールとサービス

- [AWS クラウド移行ファクトリーソリューション](#)
- [での無料のクラウド移行サービス AWS](#)
- [AWS Database Migration Service](#)
- [で移行する AWS](#)
- [Flexera One Cloud Migration and Modernization](#) (Flexera ウェブサイト)
- [TDS TransitionManager](#) (TDS ウェブサイト)

AWS 規範ガイド

- [クラウド移行ファクトリーによる大規模なサーバー移行の自動化](#)
- [への移行中に廃止されるアプリケーションを評価するためのベストプラクティス AWS クラウド](#)
- [移行の準備状況の評価](#)
- [自動ポートフォリオ検出の使用を開始する](#)
- [Mobilize your organization to accelerate large-scale migrations](#)
- [リレーショナルデータベースの移行戦略](#)
- [AWS クラウド 移行のためのアプリケーションポートフォリオ評価ガイド](#)

動画

- [への大規模な移行の実行 AWS](#) (AWS re:Invent 2020)
- [CloudEndure Migration Factory のベストプラクティス](#) (AWS re:Invent 2020)

寄稿者

このドキュメントの寄稿者は次のとおりです。

- Amazon Web Services、Principal Cloud Architect、Pratik Chunawala
- Amazon Web Services、Senior Cloud Application Architect、Dwayne Bordelon
- Amazon Web Services、Senior Application Architect、Rodolfo Jr. Cerrada
- Amazon Web Services、プリンシパルコンサルタント、Wally Lu

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。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
VMware Cloud on を削除 AWS	VMware Cloud on への参照を削除 AWS し、 一般的な移行戦略とパターン のリストを更新しました。	2024 年 7 月 5 日
AWS ソリューションの名前を 更新しました	参照される AWS ソリューションの名前を CloudEndure Migration Factory から Cloud Migration Factory に更新しました。	2022 年 5 月 2 日
初版発行	—	2022 年 2 月 28 日

AWS 規範ガイドの用語集

以下は、AWS 規範ガイドによって提供される戦略、ガイド、パターンで一般的に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。

数字

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を購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例: Microsoft Hyper-V アプリケーションをに移行します AWS。
- 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。
- 廃止 — 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。

A

ABAC

[「属性ベースのアクセス制御」](#)をご覧ください。

抽象化されたサービス

[「マネージドユーザー」](#)をご覧ください。

ACID

[「原子性、一貫性、分離性、耐久性 \(ACID\)」](#)をご覧ください。

アクティブ/アクティブ移行

(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) 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、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。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。[アクティブ/パッシブ移行](#)よりも柔軟な方法ですが、さらに多くの作業が必要となります。

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を同期させながら、データがターゲットデータベースにレプリケートされている間、接続しているアプリケーションからのトランザクションをソースデータベースのみで処理するデータベース移行方法。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

複数行に処理を行い、グループ全体を対象に単一の戻り値を計算する SQL 関数。集計関数の例としては、SUM や MAX などがあります。

AI

[「人工知能」](#)をご覧ください。

AIOps

[「AI オペレーション」](#)をご覧ください。

匿名化

データセット内の個人情報を完全に削除するプロセス。匿名化は個人のプライバシー保護に役立ちます。匿名化されたデータは、もはや個人データとは見なされません。

アンチパターン

繰り返し起こる問題に対して頻繁に用いられる解決策で、その解決策が逆効果であったり、効果がなかったり、代替案よりも効果が低かったりするもの。

アプリケーション制御

マルウェアからシステムを保護するために、承認されたアプリケーションのみを使用できるようにするセキュリティアプローチ。

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](#)の重要な要素であり、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。

人工知能 (AI)

コンピューティングテクノロジーを使用し、学習、問題の解決、パターンの認識など、通常は人間に関連づけられる認知機能の実行に特化したコンピュータサイエンスの分野。詳細については、「[人工知能 \(AI\) とは何ですか?](#)」をご覧ください。

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。

非対称暗号化

暗号化用のパブリックキーと復号用のプライベートキーから成る 1 組のキーを使用した、暗号化のアルゴリズム。パブリックキーは復号には使用されないため共有しても問題ありませんが、プライベートキーの利用は厳しく制限する必要があります。

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上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセス制御 (ABAC)

部署、役職、チーム名など、ユーザーの属性に基づいてアクセス許可をきめ細かく設定する方法。詳細については、AWS Identity and Access Management (IAM) ドキュメントの「[の ABAC AWS](#)」を参照してください。

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリーバージョンを保存する場所。匿名化、編集、仮名化など、データを処理または変更する目的で、信頼できるデータソースから他の場所にデータをコピーすることができます。

アベイラビリティゾーン (AZ)

他のアベイラビリティゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドへの移行を成功させるための効率的で効果的な計画を立てるための、こ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です。AWS CAF 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、運用という 6 つの重点分野にガイドランスを整理しています。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、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションに関するガイドランスを提供します。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 ワークロード認定フレームワーク (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、作業見積もりを提供するツール。AWS WQF は AWS Schema Conversion Tool (AWS SCT) に含まれています。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。

B

不正なボット

個人や組織に混乱や損害を与えることを目的とした [ボット](#)。

BCP

「[ビジネス継続性計画 \(BCP\)](#)」をご覧ください。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective の動作グラフを使用すると、失敗したログオンの試行、不審な API 呼び出し、その他同様のアクションを調べることができます。詳細については、Detective ドキュメントの「[動作グラフのデータ](#)」を参照してください。

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。「[エンディアン性](#)」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの一つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーである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。

ブルー/グリーンデプロイ

それぞれが独立しているが、同一の環境を 2 つ作成するデプロイ戦略。現在のアプリケーションバージョンを 1 つの環境 (ブルー) で実行し、新しいアプリケーションバージョンを別の環境 (グリーン) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティややり取りをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクローラーなど、一部のボットは有用または有益です。悪質なボットと呼ばれる他のボットの中には、個人や組織を混乱させたり、損害を与えたりすることを意図したものもあります。

ボットネット

[マルウェア](#)に感染しており、ボットハーダーまたはボットオペレーターと呼ばれる単一の当事者によって制御されている[ボット](#)のネットワーク。ボットネットは、ボットとその影響力を拡大する仕組みとして、非常によく知られています。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発した

り、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブランチと呼ばれます。機能をリリースする準備ができたなら、機能ブランチをメインブランチに統合します。詳細については、「[ブランチの概要](#)」(GitHub ドキュメント)を参照してください。

ブレイクグラスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが AWS アカウント 通常アクセス許可を持たないにすばやくアクセスできるようにします。詳細については、AWS Well-Architected ガイドの「[ブレイクグラス手順の実装](#)」インジケータを参照してください。

ブラウフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウフィールド戦略を導入する場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計します。既存のインフラストラクチャを拡張している場合は、ブラウフィールド戦略と[グリーンフィールド](#)戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

価値を生み出すためにビジネスが行うこと (営業、カスタマーサービス、マーケティングなど)。マイクロサービスのアーキテクチャと開発の決定は、ビジネス能力によって推進できます。詳細については、[AWSでのコンテナ化されたマイクロサービスの実行](#)ホワイトペーパーの「[ビジネス機能を中心に組織化](#)」セクションを参照してください。

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。

C

CAF

「[AWS クラウド導入フレームワーク](#)」を参照してください

カナリアデプロイ

エンドユーザーへのバージョンリリースを、時間をかけて段階的に行うこと。確信が持てたら新規バージョンをデプロイして、現在のバージョン全体を置き換えます。

CCoE

「[Cloud Center of Excellence](#)」を参照してください。

CDC

「[変更データキャプチャ](#)」を参照してください。

変更データキャプチャ (CDC)

データソース (データベーステーブルなど) 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での変更を監査またはレプリケート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。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストすること。[AWS Fault Injection Service \(AWS FIS\)](#) を使用して、AWS ワークロードにストレスを与え、その応答を評価する実験を実行できます。

CI/CD

「[継続的インテグレーションと継続的デリバリー](#)」を参照してください。

分類

予測を生成するのに役立つ分類プロセス。分類問題の機械学習モデルは、離散値を予測します。離散値は、常に互いに区別されます。例えば、モデルがイメージ内に車があるかどうかを評価する必要がある場合があります。

クライアント側の暗号化

ターゲットがデータ AWS のサービスを受信する前のローカルでのデータの暗号化。

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 投稿](#) を参照してください。

クラウドコンピューティング

リモートデータストレージと IoT デバイス管理に通常使用されるクラウドテクノロジー。クラウドコンピューティングは、一般的に、[エッジコンピューティング](#)に接続されています。

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、「[クラウド運用モデルの構築](#)」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、AWS クラウド エンタープライズ戦略ブログのブログ記事「[クラウドファーストへのジャーニー](#)」と「[導入のステージ](#)」で Stephen Orban によって定義されました。移行戦略との関連性については、AWS「[移行準備ガイド](#)」を参照してください。

CMDB

「[構成管理データベース \(CMDB\)](#)」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、GitHub や Bitbucket Cloud があります。コードの各バージョンはブランチと呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

空である、または、かなり空きがある、もしくは、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読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

めったにアクセスされず、通常は過去のデータです。この種類のデータをクエリする場合、通常は低速なクエリでも問題ありません。このデータを低パフォーマンスで安価なストレージ階層またはクラスに移動すると、コストを削減することができます。

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオといった、ビジュアル形式の情報を分析および抽出する [AI](#) の分野。例えば、Amazon SageMaker AI では、CV 用の画像処理アルゴリズムを利用できます。

設定ドリフト

ワークロードにおいて、設定が想定した状態から変化すること。これによって、ワークロードが非準拠になる可能性があります。この状態は、徐々に生じ、意図的なものではありません。

構成管理データベース (CMDB)

データベースとその IT 環境 (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設定を含む) に関する情報を保存、管理するリポジトリ。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。

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンの単一のエンティティとしてデプロイすることも、組織全体にデプロイすることもできます。詳細については、AWS Config ドキュメントの「[コンフォーマンスパック](#)」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「[継続的デリバリーの利点](#)」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「[継続的デリバリーと継続的なデプロイ](#)」を参照してください。

CV

[「コンピュータビジョン」](#) を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

ネットワーク内のデータを重要度と機密性に基づいて識別、分類するプロセス。データに適した保護および保持のコントロールを判断する際に役立つため、あらゆるサイバーセキュリティのリスク管理戦略において重要な要素です。データ分類は、AWS Well-Architected フレームワークのセキュリティの柱のコンポーネントです。詳細については、「[データ分類](#)」を参照してください。

データドリフト

実稼働データと ML モデルのトレーニングに使用されたデータとの間に有意な差異が生じたり、入力データが時間の経過と共に有意に変化したりすることです。データドリフトは、ML モデル予測の全体的な品質、精度、公平性を低下させる可能性があります。

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

非一元的で分散型のデータ所有権を持つとともに、一元的な管理およびガバナンスを行えるアーキテクチャフレームワーク。

データ最小化

厳密に必要なデータのみを収集し、処理するという原則。でデータ最小化を実践 AWS クラウドすることで、プライバシーリスク、コスト、分析のカーボンフットプリントを削減できます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「[でのデータ境界の構築 AWS](#)」を参照してください。

データの前処理

raw データをお客様の機械学習モデルで簡単に解析できる形式に変換すること。データの前処理とは、特定の列または行を削除して、欠落している、矛盾している、または重複する値に対処することを意味します。

データ出所

データの生成、送信、保存の方法など、データのライフサイクル全体を通じてデータの出所と履歴を追跡するプロセス。

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには、一般的に、大量の履歴データが含まれており、多くの場合、それらはクエリや分析に使用されます。

データベース定義言語 (DDL)

データベース内のテーブルや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。

DDL

「[データベース定義言語](#)」を参照してください。

ディープアンサンブル

予測のために複数の深層学習モデルを組み合わせます。ディープアンサンブルを使用して、より正確な予測を取得したり、予測の不確実性を推定したりできます。

深層学習

人工ニューラルネットワークの複数層を使用して、入力データと対象のターゲット変数の間のマッピングを識別する機械学習サブフィールド。

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリティの手法。この戦略をに採用するときは AWS、リソースの保護に役立つように、AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加します。たとえば、多層防御アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

では AWS Organizations、互換性のあるサービスが AWS メンバーアカウントを登録して組織のアカウントを管理し、そのサービスのアクセス許可を管理できます。このアカウントを、そのサービスの委任管理者と呼びます。詳細、および互換性のあるサービスの一覧は、AWS

Organizations ドキュメントの「[AWS Organizationsで利用できるサービス](#)」を参照してください。

トラブルシューティング

アプリケーション、新機能、コードの修正をターゲットの環境で利用できるようにするプロセス。デプロイでは、コードベースに変更を施した後、アプリケーションの環境でそのコードベースを構築して実行します。

開発環境

「[環境](#)」を参照してください。

検出管理

イベントが発生したときに、検出、ログ記録、警告を行うように設計されたセキュリティコントロール。これらのコントロールは副次的な防衛手段であり、実行中の予防的コントロールをすり抜けたセキュリティイベントをユーザーに警告します。詳細については、「AWSでのセキュリティコントロールの実装」の「[検出的コントロール](#)」を参照してください。

開発バリューストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、もともとリーンマニファクチャリング・プラクティスのために設計されたバリューストリームマッピング・プロセスを拡張したものです。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

建物、工場、産業機器、生産ラインなど、現実世界のシステムを仮想的に表現したものです。デジタルツインは、予知保全、リモートモニタリング、生産最適化をサポートします。

ディメンションテーブル

[スタースキーマ](#)において、ファクトテーブルの定量データに関するデータ属性が含まれる小さいテーブル。ディメンションテーブルの属性は、通常、テキストフィールド、またはテキストのように扱える個別の数値で示されます。これらの属性は、一般的に、クエリの制約、フィルタリング、結果セットのラベル付けに使用されます。

デザスタ

ワークロードまたはシステムが、導入されている主要な場所でのビジネス目標の達成を妨げるイベント。これらのイベントは、自然災害、技術的障害、または意図しない設定ミスやマルウェア攻撃などの人間の行動の結果である場合があります。

ディザスタリカバリ (DR)

[ディザスタ](#)によるダウンタイムとデータ損失を最小限に抑えるための戦略とプロセス。詳細については、AWS Well-Architected フレームワークの「[でのワークロードのディザスタリカバリ](#)」[AWS: クラウドでのリカバリ](#)」を参照してください。

DML

「[データベース操作言語](#)」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み) で紹介されています (ポストン: Addison-Wesley Professional、2003)。strangler fig パターンでドメイン駆動型設計を使用する方法の詳細については、「[コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET \(ASMX\) ウェブサービスを段階的にモダナイズ](#)」を参照してください。

DR

「[ディザスタリカバリ](#)」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、AWS CloudFormation を使用して[システムリソースのドリフトを検出](#)したり、を使用して AWS Control Tower、ガバナンス要件への準拠に影響する[ランディングゾーンの変更を検出](#)したりできます。

DVSM

「[開発バリューSTREAMマッピング](#)」を参照してください。

E

EDA

「[探索的データ分析](#)」を参照してください。

EDI

「[電子データ交換](#)」を参照してください。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。[クラウドコンピューティング](#)と比較すると、エッジコンピューティングは通信レイテンシーを短縮し、応答時間を改善できます。

電子データ交換 (EDI)

組織間で行う、ビジネスドキュメントの自動交換。詳細については、[「電子データ交換とは」](#)を参照してください。

暗号化

人間が読み取り可能なプレーンテキストデータを暗号文に変換するコンピューティング処理。

暗号化キー

暗号化アルゴリズムが生成した、ランダム化されたビットからなる暗号文字列。キーの長さは決まっておらず、各キーは予測できないように、一意になるように設計されています。

エンディアン

コンピュータメモリにバイトが格納される順序。ビッグエンディアンシステムでは、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システムでは、最下位バイトが最初に格納されます。

エンドポイント

[「サービスエンドポイント」](#)を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これらのアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの [「エンドポイントサービスを作成する」](#)を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (会計、[MES](#)、プロジェクト管理など) を自動化および管理するシステム。

エンベロープ暗号化

暗号化キーを、別の暗号化キーを使用して暗号化するプロセス。詳細については、AWS Key Management Service (AWS KMS) ドキュメントの「[エンベロープ暗号化](#)」を参照してください。

環境

実行中のアプリケーションのインスタンス。クラウドコンピューティングにおける一般的な環境の種類は以下のとおりです。

- 開発環境 — アプリケーションのメンテナンスを担当するコアチームのみが利用できる、実行中のアプリケーションのインスタンス。開発環境は、上位の環境に昇格させる変更をテストするときに使用します。このタイプの環境は、テスト環境と呼ばれることもあります。
- 下位環境 — 初期ビルドやテストに使用される環境など、アプリケーションのすべての開発環境。
- 本番環境 — エンドユーザーがアクセスできる、実行中のアプリケーションのインスタンス。CI/CD パイプラインでは、本番環境が最後のデプロイ環境になります。
- 上位環境 — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

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、ID とアクセスの管理、検出コントロール、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。

ERP

「[エンタープライズリソース計画](#)」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、統計の概要を計算し、データの可視化を作成することによって実行されます。

F

ファクトテーブル

[スタースキーマ](#)の中央にあるテーブル。ビジネスオペレーションに関する定量的データが保存されます。一般的に、ファクトテーブルは、2種類の列で構成されます。1つは測定値が含まれる列、もう1つはディメンションテーブルへの外部キーが含まれる列です。

フェイルファスト

開発ライフサイクルを短縮するために、頻繁かつ段階的にテストを行う哲学であり、アジャイルアプローチでは、この考え方がきわめて重要です。

障害分離境界

では AWS クラウド、障害の影響を制限し、ワークロードの耐障害性を高めるのに役立つアベイラビリティゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界。詳細については、「[AWS 障害分離境界](#)」を参照してください。

機能ブランチ

「[ブランチ](#)」を参照してください。

特徴量

お客様が予測に使用する入力データ。例えば、製造コンテキストでは、特徴量は製造ラインから定期的にキャプチャされるイメージの可能性もあります。

特徴量重要度

モデルの予測に対する特徴量の重要性。これは通常、Shapley Additive Deskonations (SHAP) や積分勾配など、さまざまな手法で計算できる数値スコアで表されます。詳細については、「[を使用した機械学習モデルの解釈可能性 AWS](#)」を参照してください。

機能変換

追加のソースによるデータのエンリッチ化、値のスケーリング、単一のデータフィールドからの複数の情報セットの抽出など、機械学習プロセスのデータを最適化すること。これにより、機械学習モデルはデータの恩恵を受けることができます。例えば、「2021-05-27 00:15:37」の日付を「2021年」、「5月」、「木」、「15」に分解すると、学習アルゴリズムがさまざまなデータコンポーネントに関連する微妙に異なるパターンを学習するのに役立ちます。

数ショットプロンプト

[LLM](#) に、タスクと望ましい出力を示す例を少数提示した後に、類似のタスクを実行させること。この手法は、プロンプトに記述された例(ショット)からモデルが学習する「インコンテキスト学

習」の一種です。数ショットプロンプトは、特定のフォーマット、推論、専門知識が必要なタスクに効果的です。「[ゼロショットプロンプト](#)」も参照してください。

FGAC

「[きめ細かなアクセス制御](#)」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

[変更データのキャプチャ](#)による継続的なデータ複製を利用して、段階的なアプローチではなく、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。

FM

「[基盤モデル](#)」を参照してください。

基盤モデル (FM)

大規模な深層学習ニューラルネットワークであり、一般化およびラベル付けされていないデータからなる大規模データセットでトレーニングされています。FMにより、言語理解、テキストおよび画像生成、自然言語での会話といった、一般的な各種タスクを実行できます。詳細については、「[基盤モデルとは何ですか?](#)」を参照してください。

G

生成 AI

[AI](#) モデルのサブセット。大量のデータでトレーニングされており、シンプルなテキストプロンプトを使用して、画像、動画、テキスト、オーディオなどの新しいコンテンツやアーティファクトを作成できます。詳細については、「[生成 AI とは何ですか?](#)」を参照してください。

ジオブロッキング

「[地理的制限](#)」を参照してください。

地理的制限 (ジオブロッキング)

特定の国のユーザーがコンテンツ配信にアクセスできないようにするための、Amazon CloudFront のオプション。アクセスを許可する国と禁止する国は、許可リストまたは禁止リスト

を使って指定します。詳細については、CloudFront ドキュメントの「[コンテンツの地理的ディストリビューションの制限](#)」を参照してください。

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローは古いと見なされている方法であり、[トランクベースのワークフロー](#)は推奨されている新しい方法です。

ゴールデンイメージ

システムまたはソフトウェアのスナップショットであり、システムまたはソフトウェアの新規インスタンスをデプロイするテンプレートとして使用されます。製造の例で言えば、ゴールデンイメージを使用すると、複数のデバイスにソフトウェアをプロビジョニングして、デバイス製造オペレーションの速度、スケーラビリティ、生産性を向上させることができます。

グリーンフィールド戦略

新しい環境に既存のインフラストラクチャが存在しないこと。システムアーキテクチャにグリーンフィールド戦略を導入する場合、既存のインフラストラクチャ (別名 [ブラウンフィールド](#)) との互換性の制約を受けることなく、あらゆる新しいテクノロジーを選択できます。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実装します。これらは、サービスコントロールポリシーと IAM アクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは AWS Config、AWS Security Hub CSPM、Amazon GuardDuty、AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「[高可用性](#)」を参照してください。

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。](#)

高可用性 (HA)

課題や災害が発生した場合に、介入なしにワークロードを継続的に運用できること。HA システムは、自動的にフェイルオーバーし、一貫して高品質のパフォーマンスを提供し、パフォーマンスへの影響を最小限に抑えながらさまざまな負荷や障害を処理するように設計されています。

ヒストリアンのモダナイゼーション

製造業のニーズによりよく応えるために、オペレーションテクノロジー (OT) システムをモダナイズし、アップグレードするためのアプローチ。ヒストリアンは、工場内のさまざまなソースからデータを収集して保存するために使用されるデータベースの一種です。

ホールドアウトデータ

[機械学習](#) モデルのトレーニング用データセットから保留される、ラベル付き履歴データの一部。ホールドアウトデータを使用すると、モデル予測をホールドアウトデータと比較して、モデルのパフォーマンスを評価できます。

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。

ホットデータ

リアルタイムデータや最近の翻訳データなど、頻繁にアクセスされるデータ。通常、このデータには高速なクエリ応答を提供する高性能なストレージ階層またはクラスが必要です。

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性が高いため、通常の DevOps のリリースワークフローからは外れた形で実施されます。

ハイパーケア期間

カットオーバー直後、移行したアプリケーションを移行チームがクラウドで管理、監視して問題に対処する期間。通常、この期間は 1~4 日です。ハイパーケア期間が終了すると、アプリケーションに対する責任は一般的に移行チームからクラウドオペレーションチームに移ります。

I

laC

「[Infrastructure as Code](#)」を参照してください。

ID ベースのポリシー

AWS クラウド 環境内のアクセス許可を定義する 1 つ以上の IAM プリンシパルにアタッチされたポリシー。

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。

IIoT

「[インダストリアル IoT](#)」を参照してください。

イミュータブルインフラストラクチャ

既存インフラストラクチャの更新、パッチ適用、変更などを行わずに、本番環境ワークロードに使用する新規インフラストラクチャをデプロイするモデル。本質的に、イミュータブルインフラストラクチャは、[ミュータブルインフラストラクチャ](#)よりも一貫性、信頼性、予測性に優れています。詳細については、AWS Well-Architected フレームワークにある「[イミュータブルインフラストラクチャを使用してデプロイする](#)」のベストプラクティスを参照してください。

インバウンド (受信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングする VPC。[AWS Security Reference Architecture](#) では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

I

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、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行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。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。

インダストリー 4.0

2016 年に [Klaus Schwab](#) 氏が提唱した用語で、接続、リアルタイムデータ、オートメーション、分析、AI/ML の進歩による、ビジネスプロセスのモダナイズを意味します。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

アプリケーションのインフラストラクチャを一連の設定ファイルを使用してプロビジョニングし、管理するプロセス。IaC は、新しい環境を再現可能で信頼性が高く、一貫性のあるものにするため、インフラストラクチャを一元的に管理し、リソースを標準化し、スケールを迅速に行えるように設計されています。

インダストリアル IoT (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、「[インダストリアル IoT \(IIoT\) デジタルトランスフォーメーション戦略の構築](#)」を参照してください。

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。 [AWS Security Reference Architecture](#) では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

IoT

インターネットまたはローカル通信ネットワークを介して他のデバイスやシステムと通信する、センサーまたはプロセッサが組み込まれた接続済み物理オブジェクトのネットワーク。詳細については、「[IoT とは](#)」を参照してください。

解釈可能性

機械学習モデルの特性で、モデルの予測がその入力にどのように依存するかを人間が理解できる度合いを表します。詳細については、[「を使用した機械学習モデルの解釈可能性 AWS」](#)を参照してください。

IoT

[「IoT」](#)を参照してください。

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。

ITIL

[「IT 情報ライブラリ」](#)を参照してください。

ITSM

[「IT サービス管理」](#)を参照してください。

L

ラベルベースアクセス制御 (LBAC)

強制アクセス制御 (MAC) の実装で、ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられます。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント 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。

大規模言語モデル (LLM)

大量のデータで事前トレーニングされた深層学習 [AI](#) モデル。LLM では、質問への回答、ドキュメントの要約、他言語へのテキスト翻訳、文を完成させるなど、さまざまなタスクを実行できます。詳細については、「[大規模言語モデル \(LLM\) とは何ですか?](#)」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。

LBAC

「[ラベルベースアクセス制御](#)」を参照してください。

最小特権

タスクの実行には必要最低限の権限を付与するという、セキュリティのベストプラクティス。詳細については、IAM ドキュメントの「[最小特権アクセス許可を適用する](#)」を参照してください。

リフトアンドシフト

「[7 Rs](#)」を参照してください。

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。「[エンディアン性](#)」もご覧ください。

LLM

「[大規模言語モデル](#)」を参照してください。

下位環境

「[環境](#)」を参照してください。

M

機械学習 (ML)

パターン認識と学習にアルゴリズムと手法を使用する人工知能の一種。ML は、モノのインターネット (IoT) データなどの記録されたデータを分析して学習し、パターンに基づく統計モデルを生成します。詳細については、「[機械学習](#)」を参照してください。

メインブランチ

「[ブランチ](#)」を参照してください。

マルウェア

コンピュータのセキュリティやプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスを招く可能性があります。マルウェアの例には、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービスはインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、エンドポイントにアクセスしてデータを保存および取得します。マネージドサービスの例として、Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB が挙げられます。このサービスは、抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するソフトウェアシステムであり、工場では、これによって、原材料から製品を完成させます。

MAP

[「Migration Acceleration Program」](#) を参照してください。

メカニズム

ツールを作成してその導入を推進し、導入結果を調べて調整を行うための包括的なプロセス。メカニズムとは、運用中にそれ自体を強化し改善するサイクルを意味します。詳細については、AWS 「Well-Architected フレームワーク」の [「メカニズムの構築」](#) を参照してください。

メンバーアカウント

組織の一部である管理アカウント AWS アカウント 以外のすべて AWS Organizations。アカウントが組織のメンバーになることができるのは、一度に 1 つのみです。

MES

[「製造実行システム」](#) を参照してください。

Message Queuing Telemetry Transport (MQTT)

[発行/サブスクリプション](#) のパターンに基づく、軽量のマシンツーマシン (M2M) 通信プロトコルであり、リソースに限りのある [IoT](#) デバイスに使用されます。

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス

機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、[AWS「サーバーレスサービスを使用したマイクロサービスの統合」](#)を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量 API を使用して、明確に定義されたインターフェイスを介して通信します。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およびスケーリングできます。詳細については、「[でのマイクロサービスの実装 AWS](#)」を参照してください。

Migration Acceleration Program (MAP)

組織がクラウドに移行する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。

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、各ウェー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。移行ファクトリーチームには、通常、運用、ビジネスアナリストおよび所有者、移行エンジニア、デベロッパー、およびスプリントで作業する DevOps プロフェッショナルが含まれます。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[移行ファクトリーに関する解説](#)と [Cloud Migration Factory 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ーに関する情報。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は、ターゲットサブネット、セキュリティグループ、AWS アカウントなどがあります。

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 への移行をリホストします。

Migration Portfolio Assessment (MPA)

オンラインツール。これによって、AWS クラウドに移行するビジネスケースの検証に必要な情報を得られます。MPA は、詳細なポートフォリオ評価 (サーバー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、およびウェーブプランニング) を提供します。[MPA ツール](#) (ログインが必要) は、すべての AWS コンサルタントと APN パートナー コンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

AWS CAF を使用して、組織のクラウド準備状況に関するインサイトを取得し、長所と短所を特定し、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 は、[AWS 移行戦略](#)の第一段階です。

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](#) エントリと、「[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。

ML

「[機械学習](#)」を参照してください。

モダナイゼーション

古い (レガシーまたはモノリシック) アプリケーションとそのインフラストラクチャをクラウド内の俊敏で弾力性のある高可用性システムに変換して、コストを削減し、効率を高め、イノベーションを活用します。詳細については、「[AWS クラウドでのアプリケーションのモダナイズ戦略](#)」を参照してください。

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定されたギャップに対処するためのアクションプランが得られます。詳細については、「[AWS クラウドでのアプリケーションのモダナイゼーションの準備状況を評価する](#)」を参照してください。

モノリシックアプリケーション (モノリス)

緊密に結合されたプロセスを持つ単一のサービスとして実行されるアプリケーション。モノリシックアプリケーションにはいくつかの欠点があります。1つのアプリケーション機能エクスペリエンスの需要が急増する場合は、アーキテクチャ全体をスケーリングする必要があります。モノリシックアプリケーションの特徴を追加または改善することは、コードベースが大きくなると複雑になります。これらの問題に対処するには、マイクロサービスアーキテクチャを使用できます。詳細については、「[モノリスをマイクロサービスに分解する](#)」を参照してください。

MPA

「[Migration Portfolio Assessment](#)」を参照してください。

MQTT

「[Message Queuing Telemetry Transport](#)」を参照してください。

多クラス分類

複数のクラスの予測を生成するプロセス (2 つ以上の結果の 1 つを予測します)。例えば、機械学習モデルが、「この製品は書籍、自動車、電話のいずれですか?」または、「このお客様にとって最も関心のある商品のカテゴリはどれですか?」と聞くかもしれません。

ミュータブルなインフラストラクチャ

本番ワークロードに使用する既存のインフラストラクチャを更新および変更するためのモデル。Well-Architected AWS フレームワークでは、一貫性、信頼性、予測可能性を向上させるために、[イミュータブルインフラストラクチャ](#)の使用をベストプラクティスとして推奨しています。

O

OAC

「[オリジンアクセス制御](#)」を参照してください。

OAI

「[オリジンアクセスアイデンティティ](#)」を参照してください。

OCM

「[組織変更管理](#)」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させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

「[オペレーション統合](#)」を参照してください。

Ola

「[オペレーショナルレベルアグリーメント](#)」を参照してください。

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、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

「[Open Process Communications - Unified Architecture](#)」を参照してください。

Open Process Communications - Unified Architecture (OPC-UA)

産業オートメーション用のマシンツーマシン (M2M) 通信プロトコル。OPC-UA により、相互運用の際に、データ暗号化、認証、認可の各スキームを標準化できます。

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。

運用準備状況レビュー (ORR)

質問と関連するベストプラクティスのチェックリスト。インシデントや起こり得る障害を理解、評価、防止したり、その範囲を縮小したりする際に役立ちます。詳細については、AWS Well-Architected フレームワークの「[Operational Readiness Reviews \(ORR\)](#)」を参照してください。

運用テクノロジー (OT)

産業オペレーション、機器、インフラストラクチャを制御するために物理環境と連携させるハードウェアおよびソフトウェアシステム。製造分野では、[Industry 4.0](#) への変革を進める上で、OT と情報技術 (IT) システムの統合に焦点が当てられています。

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。

組織の証跡

組織 AWS アカウント 内のすべてのイベント AWS CloudTrail をログに記録することによって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウントに作成され、各アカウントのアクティビティを追跡します。詳細については、CloudTrail ドキュメントの「[組織の証跡の作成](#)」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化のスピードにより、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、[OCM ガイド](#)を参照してください。

オリジンアクセス制御 (OAC)

Amazon Simple Storage Service (Amazon S3) コンテンツを保護するための、CloudFront のアクセス制限の強化オプション。OAC は AWS リージョン、すべての S3 バケット、AWS KMS (SSE-KMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETE リクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

CloudFront の、Amazon S3 コンテンツを保護するためのアクセス制限オプション。OAI を使用すると、CloudFront が、Amazon S3 に認証可能なプリンシパルを作成します。認証されたプリンシパルは、S3 バケット内のコンテンツに、特定の CloudFront ディストリビューションを介してのみアクセスできます。[OAC](#) も併せて参照してください。OAC では、より詳細な、強化されたアクセス制御が可能です。

ORR

「[運用準備状況レビュー](#)」を参照してください。

OT

「[運用テクノロジー](#)」を参照してください。

アウトバウンド (送信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されたネットワーク接続を処理する VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

ユーザーまたはロールが使用できるアクセス許可の上限を設定する、IAM プリンシパルにアタッチされる IAM 管理ポリシー。詳細については、IAM ドキュメントの[アクセス許可の境界](#)を参照してください。

個人を特定できる情報 (PII)

直接閲覧した場合、または他の関連データと組み合わせた場合に、個人の身元を合理的に推測するために使用できる情報。PII の例には、氏名、住所、連絡先情報などがあります。

PII

「[個人を特定できる情報](#)」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。

PLC

「[プログラマブルロジックコントローラー](#)」を参照してください。

PLM

「[製品ライフサイクル管理](#)」を参照してください。

ポリシー

次の操作を可能にするオブジェクト: アクセス許可を定義する ([ID ベースのポリシー](#)を参照)。アクセス条件を指定する ([リソースベースのポリシー](#)を参照)。AWS Organizations の組織における全アカウントにアクセス許可の上限を定義する ([サービスコントロールポリシー](#)を参照)。

多言語の永続性

データアクセスパターンやその他の要件に基づいて、マイクロサービスのデータストレージテクノロジーを個別に選択します。マイクロサービスが同じデータストレージテクノロジーを使用している場合、実装上の問題が発生したり、パフォーマンスが低下する可能性があります。マイクロサービスは、要件に最も適合したデータストアを使用すると、より簡単に実装でき、パフォーマンスとスケーラビリティが向上します。

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、「[移行の準備状況の評価](#)」を参照してください。

述語

true または false を返すためのクエリ条件。一般的に、WHERE 句に記述されます。

述語プッシュダウン

データベースクエリを最適化する手法。これによって、転送前にクエリ内のデータをフィルタリングします。この手法を取ると、リレーショナルデータベースから取得し処理する必要のあるデータの量が減少するため、クエリのパフォーマンスが向上します。

予防的コントロール

イベントの発生を防ぐように設計されたセキュリティコントロール。このコントロールは、ネットワークへの不正アクセスや好ましくない変更を防ぐ最前線の防御です。詳細については、「AWSでのセキュリティコントロールの実装」の「[予防的コントロール](#)」を参照してください。

プリンシパル

アクションを実行し AWS、リソースにアクセスできるのエンティティ。このエンティティは通常、IAM AWS アカウントロール、またはユーザーのルートユーザーです。詳細については、IAM ドキュメントの「[ロールに関する用語と概念](#)」にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通してプライバシーが考慮されているシステムエンジニアリングのアプローチ。

プライベートホストゾーン

1 つ以上の VPC 内のドメインとそのサブドメインへの DNS クエリに対し、Amazon Route 53 がどのように応答するかに関する情報を保持するコンテナ。詳細については、Route 53 ドキュメントの「[プライベートホストゾーンの使用](#)」を参照してください。

プロアクティブコントロール

非準拠リソースのデプロイ防止を目的とした[セキュリティコントロール](#)。このコントロールにより、プロビジョニング前にリソースをスキャンします。コントロールに準拠していないリソースは、プロビジョニングされません。詳細については、AWS Control Tower ドキュメントの「[コントロールリファレンスガイド](#)」および「[セキュリティコントロールの実装](#)」の「[プロアクティブコントロール](#)」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

製品の設計、開発、発売から、成長、成熟、衰退、廃棄に至る、製品のライフサイクル全体を通してデータとプロセスを管理すること。

本番環境

「[環境](#)」を参照してください。

プログラマブルロジックコントローラー (PLC)

製造分野で使用される、信頼性と適応性に優れたコンピュータであり、これによって、マシンをモニタリングするとともに、製造プロセスを自動化します。

プロンプトチェイニング

1 つの [LLM](#) プロンプトによる出力を次のプロンプトの入力に使用して、より良いレスポンスを生成します。この手法を使用すると、複雑なタスクをサブタスクに分割したり、事前レスポンスを繰り返し改良または拡張したりできます。これによって、モデルのレスポンスの精度と関連性が向上し、粒度の高いパーソナライズされた結果を得られます。

仮名化

データセット内の個人識別子をプレースホルダー値に置き換えるプロセス。仮名化は個人のプライバシー保護に役立ちます。仮名化されたデータは、依然として個人データとみなされます。

発行/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。これにより、スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの [MES](#) の場合、マイクロサービスは、他のマイクロサービスがサブスクライブ可能なチャンネルにイベントメッセージを発行できます。このシステムでは、発行サービスの変更なしに、新規マイクロサービスを追加できます。

Q

クエリプラン

手順などの一連のステップであり、SQL リレーショナルデータベースシステムのデータにアクセスするために使用されます。

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

「[実行責任者、説明責任者、協業先、報告先 \(RACI\)](#)」を参照してください。

RAG

「[検索拡張生成](#)」を参照してください。

ランサムウェア

決済が完了するまでコンピュータシステムまたはデータへのアクセスをブロックするように設計された、悪意のあるソフトウェア。

RASCI マトリックス

「[実行責任者、説明責任者、協業先、報告先 \(RACI\)](#)」を参照してください。

RCAC

「[行と列のアクセス制御](#)」を参照してください。

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への負荷を軽減できます。

リアーキテクト

「[7 Rs](#)」を参照してください。

目標復旧時点 (RPO)

最後のデータリカバリポイントからの最大許容時間です。これにより、最後の回復時点からサービスが中断されるまでの間に許容できるデータ損失の程度が決まります。

目標復旧時間 (RTO)

サービスが中断から復旧までの最大許容遅延時間。

リファクタリング

「[7 Rs](#)」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョンは、耐障害性、安定性、耐障害性を提供するために、他のから分離され、独立しています。詳細については、「[アカウントが使用できる AWS リージョンを指定する](#)」を参照してください。

リグレッション

数値を予測する機械学習手法。例えば、「この家はどれくらいの値段で売れるでしょうか?」という問題を解決するために、機械学習モデルは、線形回帰モデルを使用して、この家に関する既知の事実 (平方フィートなど) に基づいて家の販売価格を予測できます。

リホスト

「[7 Rs](#)」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「[7 Rs](#)」を参照してください。

リプラットフォーム

「[7 Rs](#)」を参照してください。

再購入

「[7 Rs](#)」を参照してください。

回復性

中断に抵抗または中断から回復するアプリケーションの機能。AWS クラウドでの回復力を計画する際には、一般的に、[高可用性](#)と[ディザスタリカバリ](#)が考慮されます。詳細については、「[AWS クラウドの耐障害性](#)」を参照してください。

リソースベースのポリシー

Amazon S3 バケット、エンドポイント、暗号化キーなどのリソースにアタッチされたポリシー。このタイプのポリシーは、アクセスが許可されているプリンシパル、サポートされているアクション、その他の満たすべき条件を指定します。

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

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R)、説明責任 (A)、協議 (C)、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートが含まれる場合は RASCI マトリックスと呼ばれ、含まれない場合は RACI マトリックスと呼ばれます。

レスポンスコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、「AWSでのセキュリティコントロールの実装」の「[レスポンスコントロール](#)」を参照してください。

保持

「[7 Rs](#)」を参照してください。

廃止

「[7 Rs](#)」を参照してください。

検索拡張生成 (RAG)

[生成 AI](#) の技術。これにより、[LLM](#) では、レスポンスの生成前に、トレーニングデータソースの外部にある信頼できるデータソースが参照されます。例えば、RAG モデルによって、組織のナレッジベースまたはカスタムデータのセマンティック検索を実行できる場合があります。細については、「[RAG \(検索拡張生成\) とは何ですか?](#)」を参照してください。

ローテーション

定期的に[シークレット情報](#)を更新して、攻撃者が認証情報にアクセスするのをより困難にするプロセス。

行と列のアクセス制御 (RCAC)

アクセスルールが定義された、基本的で柔軟な SQL 表現の使用。RCAC は行権限と列マスクで構成されています。

RPO

「[目標復旧時点](#)」を参照してください。

RTO

「[目標復旧時間](#)」を参照してください。

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。

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーテッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、AWS マネジメントコンソールにログインしたり AWS、API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「[SAML 2.0 ベースのフェデレーションについて](#)」を参照してください。

SCADA

「[監視制御とデータ取得](#)」を参照してください。

SCP

「[サービスコントロールポリシー](#)」を参照してください。

シークレット

暗号化された形式で保存する AWS Secrets Manager パスワードやユーザー認証情報などの機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値には、バイナリ、1 つの文字列、複数の文字列を指定できます。詳細については、Secrets Manager ドキュメントの「[Secrets Manager シークレットの概要](#)」を参照してください。

セキュリティバイデザイン

開発プロセス全体を通してセキュリティが考慮されているシステムエンジニアリングのアプローチ。

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、主に 4 つの種類があります。4 つとは、[予防](#)、[検出](#)、[レスポンス](#)、[プロアクティブ](#)です。

セキュリティ強化

アタックサーフェスを狭めて攻撃への耐性を高めるプロセス。このプロセスには、不要になったリソースの削除、最小特権を付与するセキュリティのベストプラクティスの実装、設定ファイル内の不要な機能の無効化、といったアクションが含まれています。

Security Information and Event Management (SIEM) システム

セキュリティ情報管理 (SIM) とセキュリティイベント管理 (SEM) のシステムを組み合わせたツールとサービス。SIEM システムは、サーバー、ネットワーク、デバイス、その他ソースからデータを収集、モニタリング、分析して、脅威やセキュリティ違反を検出し、アラートを発信します。

セキュリティレスポンスの自動化

セキュリティイベントへの自動レスポンスまたは自動修復を目的として、事前定義およびプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ[検出的](#)または[応答的](#)な AWS セキュリティコントロールとして機能します。自動レスポンスアクションの例には、VPC セキュリティグループの変更、Amazon EC2 インスタンスへのパッチ適用、認証情報の更新などがあります。

サーバー側の暗号化

送信先で、それ AWS のサービスを受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCP は、管理者がユーザーまたはロールに委任するアクションに、ガードレールを定義したり、アクションの制限を設定したりします。SCP は、許可リストまたは拒否リストとして、許可または禁止するサービスやアクションを指定する際に使用できます。詳細については、AWS Organizations ドキュメントの「[サービスコントロールポリシー](#)」を参照してください。

サービスエンドポイント

のエンドポイントの URL AWS のサービス。ターゲットサービスにプログラムで接続するには、エンドポイントを使用します。詳細については、「AWS 全般のリファレンス」の「[AWS のサービス エンドポイント](#)」を参照してください。

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。

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

[サービスレベルインジケータ](#)によって測定され、サービスの状態を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、 はクラウドのセキュリティを担当します。詳細については、「[責任共有モデル](#)」を参照してください。

SIEM

「[Security Information and Event Management システム](#)」を参照してください。

単一障害点 (SPOF)

特定のアプリケーションを構成する単一の重要なコンポーネントで発生し、システム稼働に支障をきたす可能性のある障害。

SLA

「[サービスレベルアグリーメント](#)」を参照してください。

SLI

「[サービスレベルインジケータ](#)」を参照してください。

SLO

「[サービスレベルの目標](#)」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、「[AWS クラウドでのアプリケーションをモダナイズするための段階的アプローチ](#)」を参照してください。

SPOF

「[単一障害点](#)」を参照してください。

スタースキーマ

データベースの編成構造を意味し、1つの大きいファクトテーブルにトランザクションデータまたは測定データが保存され、1つ以上の小さいディメンションテーブルにデータ属性が保存されます。この構造は、[データウェアハウス](#)やビジネスインテリジェンスを用途とするように設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に取って代わるイチジクのつるを例えています。そのパターンは、モノリシックシステムを書き換えるときのリスクを管理する方法として [Martin Fowler](#) により提唱されました。このパターンの適用方法の例については、「[コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET \(ASMX\) ウェブサービスを段階的にモダナイズ](#)」を参照してください。

サブネット

VPC 内の IP アドレスの範囲。サブネットは、1つのアベイラビリティゾーンに存在する必要があります。

監視制御とデータ取得 (SCADA)

製造分野において、ハードウェアとソフトウェアを使用して物理アセットと本番運用をモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーとのやり取りをシミュレートして、起こり得る問題を検出したり、パフォーマンスをモニタリングしたりすることで、システムをテストします。[Amazon CloudWatch Synthetics](#) を使用すると、こうしたテストを作成できます。

システムプロンプト

コンテキスト、指示、ガイドラインなどを提示して、[LLM](#) に動作を指示する手法。システムプロンプトは、コンテキストを設定して、ユーザーとやり取りするルールを確立するのに有用です。

T

タグ

AWS リソースを整理するためのメタデータとして機能するキーと値のペア。タグは、リソースの管理、識別、整理、検索、フィルタリングに役立ちます。詳細については、「[AWS リソースのタグ付け](#)」を参照してください。

ターゲット変数

監督された機械学習でお客様が予測しようとしている値。これは、結果変数のことも指します。例えば、製造設定では、ターゲット変数が製品の欠陥である可能性があります。

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。

テスト環境

「[環境](#)」を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパターンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

トランジットゲートウェイ

VPC とオンプレミスネットワークを相互接続するために使用できる、ネットワークの中継ハブ。詳細については、AWS Transit Gateway ドキュメントの「[トランジットゲートウェイとは](#)」を参照してください。

トランクベースのワークフロー

デベロッパーが機能ブランチで機能をローカルにビルドしてテストし、その変更をメインブランチにマージするアプローチ。メインブランチはその後、開発環境、本番前環境、本番環境に合わせて順次構築されます。

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定したサービスにアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要とときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、ドキュメントの「[Using AWS Organizations with other AWS services](#) AWS Organizations」を参照してください。

チューニング

機械学習モデルの精度を向上させるために、お客様のトレーニングプロセスの側面を変更する。例えば、お客様が機械学習モデルをトレーニングするには、ラベル付けセットを生成し、ラベルを追加します。これらのステップを、異なる設定で複数回繰り返して、モデルを最適化します。

ツーピザチーム

2 枚のピザを分け合えることができるくらい小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の 2 つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性は、データに固有のノイズとランダム性によって引き起こされます。

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

「[環境](#)」を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングできる、2 つの VPC 間の接続。詳細については、Amazon VPC ドキュメントの「[VPC ピア機能とは](#)」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

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、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

現在のレコードに何らかの形で関連している行のグループに計算を実行する SQL 関数。ウィンドウ関数は、移動平均を計算したり、現在の行の相対位置に基づいて他の行の値にアクセスするといったタスクの処理に役立ちます。

ワークロード

ビジネス価値をもたらすリソースとコード (顧客向けアプリケーションやバックエンドプロセスなど) の総称。

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。

WORM

「[Write-Once-Read-Many](#)」を参照してください。

WQF

「[AWS ワークロード資格フレームワーク](#)」を参照してください。

Write-Once-Read-Many (WORM)

データを 1 回のみ書き込むことで、データの削除や変更を防ぐストレージモデル。承認済みユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャは、[イミュータブル](#)と見なされます。

Z

ゼロデイエクスプロイト

[ゼロデイ脆弱性](#)を悪用した攻撃（一般的にマルウェアによる）。

ゼロデイ脆弱性

実稼働システムにおける未解決の欠陥または脆弱性。脅威アクターは、このような脆弱性を利用してシステムを攻撃する可能性があります。開発者は、よく攻撃の結果で脆弱性に気付きます。

ゼロショットプロンプト

[LLM](#) にタスク実行の手順は提示するが、実行のガイドとして役立つ例（ショット）は提示しない方法。LLM は、事前トレーニング済みの知識を使用してタスクを処理する必要があります。ゼロショットプロンプトの有効性は、タスクの複雑さとプロンプトの品質によって異なります。「[数ショットプロンプト](#)」も参照してください。

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。