



AWS クラウド 移行用のアプリケーションポートフォリオ評価ガイド

AWS 規範ガイド



AWS 規範ガイダンス: AWS クラウド 移行用のアプリケーションポートフォリオ評価ガイド

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
概要	1
Discovery アクセラレーションと初期計画	4
このステージの主な結果	4
初期評価データ要件について	4
データソースとデータ要件	4
検出ツールの必要性の評価	15
ビジネスドライバーと技術ガイドの原則	21
ビジネスドライバー	21
技術ガイドの原則	22
データ収集の開始	24
優先順位付けと移行戦略	25
アプリケーションの優先順位付け	26
移行用の R タイプの決定	28
方向性のあるビジネスケースの作成	31
方向性のあるビジネスケースの範囲の修正	31
フォーカス値ドライバー	32
データのニーズ	33
インフラストラクチャの TCO 比較の構築	34
運用コスト最適化の構築	35
完全な方向性のビジネスケースへの拡張	37
移行とモダナイゼーションプログラムのセットアップの見積もり	38
優先順位付けされたアプリケーション評価	49
詳細な評価データ要件について	49
アプリケーション	50
インフラストラクチャ	55
詳細なアプリケーション評価	58
General	60
アーキテクチャ	60
オペレーション	60
パフォーマンス	61
ソフトウェアのライフサイクル	61
移行	61
回復性	62

セキュリティとコンプライアンス	62
データベース	62
依存関係	63
AWS アプリケーション設計と移行戦略	63
アプリケーションの将来の状態	64
再現性	65
要件	65
To-Be アーキテクチャ	66
アーキテクチャ上の意思決定	68
ソフトウェアライフサイクル環境	69
Tagging	69
移行戦略	69
移行パターンとツール	69
サービス管理とオペレーション	70
カットオーバーに関する考慮事項	70
リスク、前提、問題、依存関係	70
実行コストの見積もり	71
ポートフォリオ分析と移行計画	72
完全な評価データ要件について	72
アプリケーション	72
インフラストラクチャ	76
ネットワーク	79
移行	80
アプリケーションポートフォリオのベースラインの確立	82
優先順位付け基準の反復	83
6 Rs 移行戦略の選択を繰り返す	86
ウェーブプランニング	86
ウェーブプランの作成	90
変更の管理	92
詳細なビジネスケース	93
ケースに必要なシナリオを決定する	94
インフラストラクチャと移行のコストモデルを検証して絞り込む	94
IT 生産性と IT 運用を改善し、効率バリューモデルをサポートする	95
レジリエンスバリューモデルを開発する	102
ビジネスの俊敏性バリューモデルを開発する	103
継続的な評価と改善	105

継続的な評価データ要件を理解する	106
詳細なウェブ評価	106
最適化とモダナイゼーションの評価	106
その他のリソース	107
ウェブプランの反復	107
ビジネスケースの進化と追跡	108
リソース	110
AWS リファレンス	110
AWS のサービス	110
ドキュメント履歴	112
.....	cxiii

AWS クラウド 移行用のアプリケーションポートフォリオ評価ガイド

German Goncalves と Mark Berner、Amazon Web Services

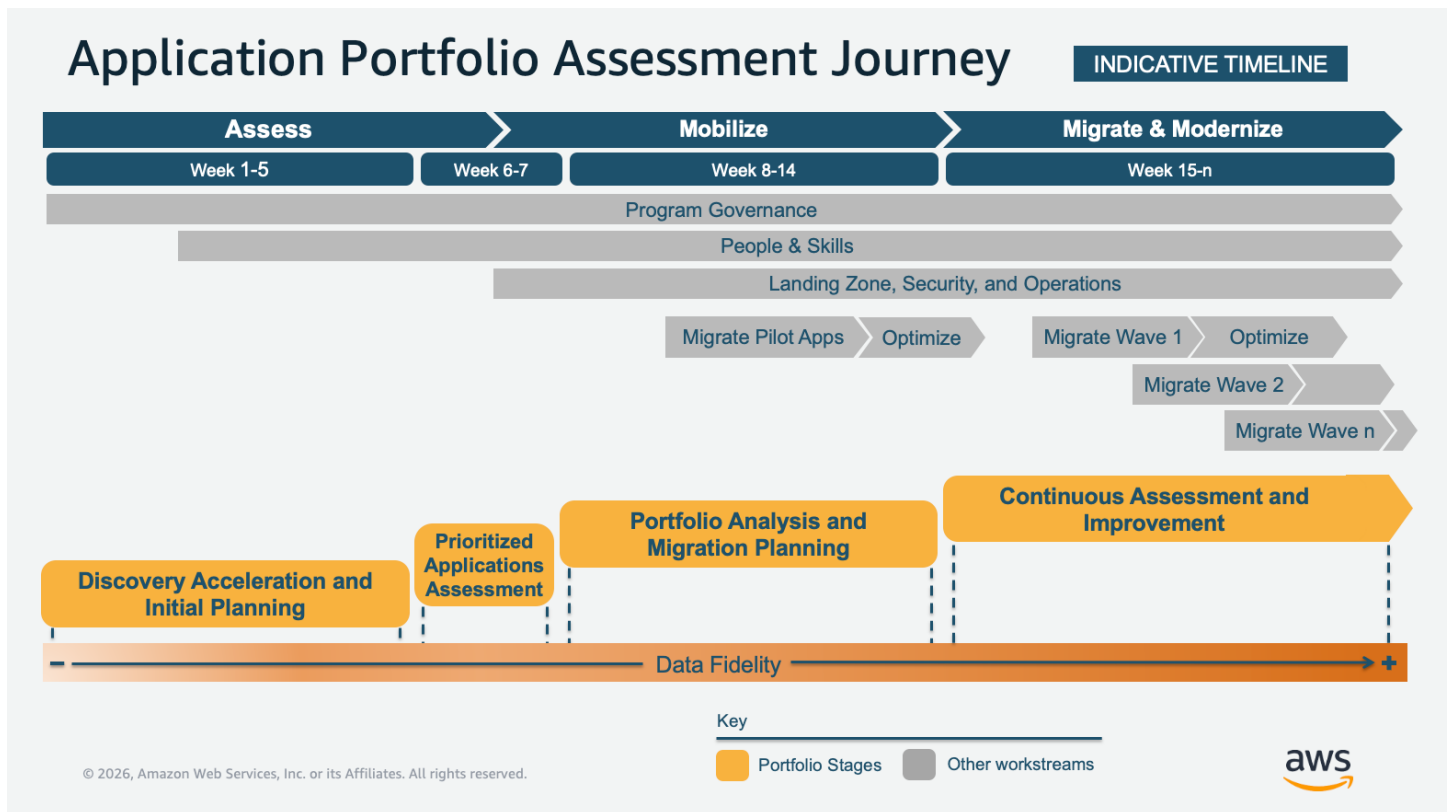
2026 年 6 月 ([ドキュメント履歴](#))

この Amazon Web Services (AWS) 規範ガイダンスドキュメントでは、[アプリケーションポートフォリオ評価戦略](#)の実装について詳しく説明します。このガイドは、アプリケーションポートフォリオと関連するインフラストラクチャの評価を開始して進めるのに役立ちます。評価には、検出、分析、計画が含まれます。インフラストラクチャには、コンピューティング、ストレージ、ネットワークが含まれます。

概要

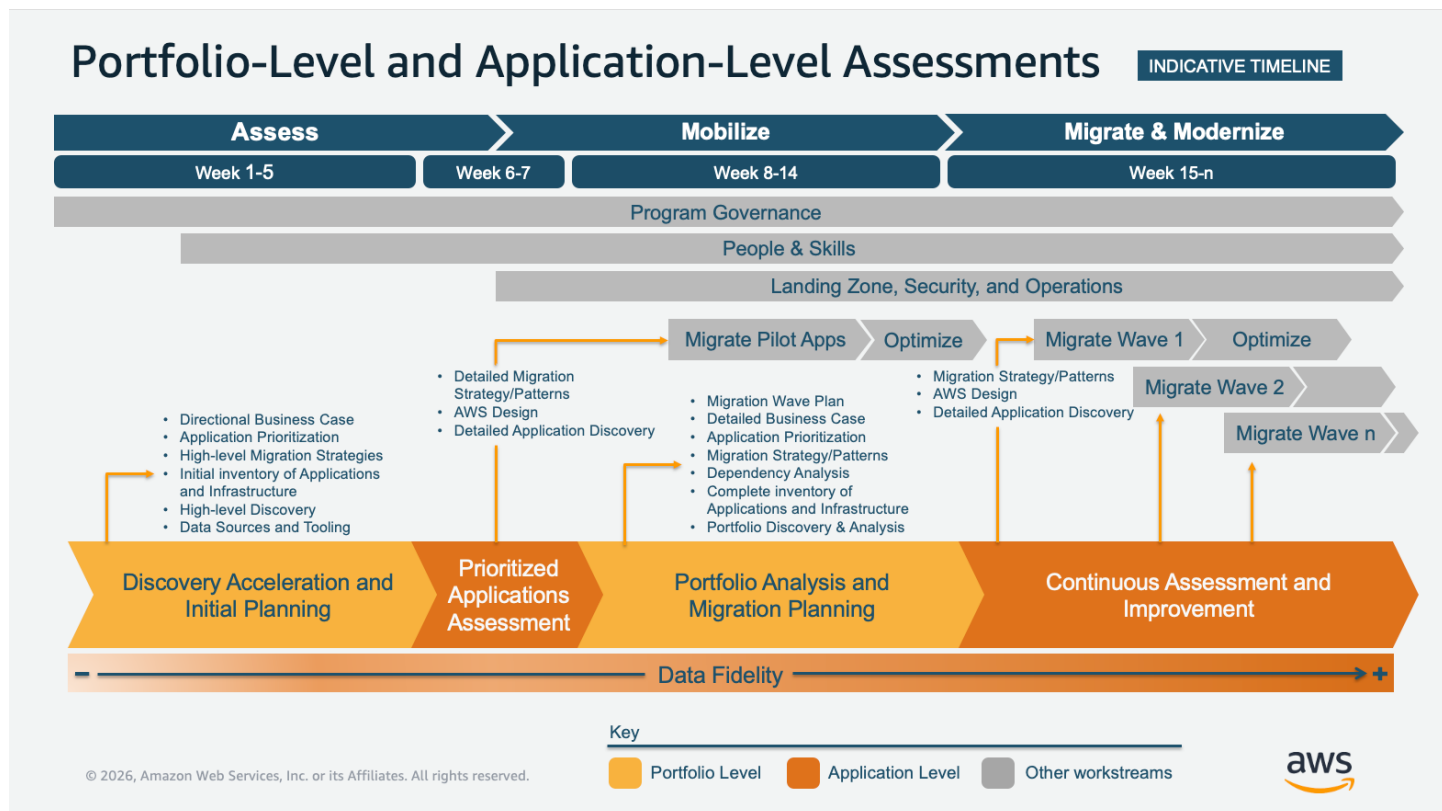
長時間実行されるクラウド移行プログラムでは、プログラムガバナンス、ランディングゾーン (セキュリティコントロールを備えた運用ターゲット環境)、移行、アプリケーションポートフォリオなど、複数のワークストリームを調整する必要があります。これらのワークストリームの名前は、移行プログラムの整理方法によって異なる場合があります。ワークストリームとして、アプリケーションポートフォリオ評価は、これらのプログラムのライフサイクル全体にわたる基本的なアクティビティを表します。評価を通じて得られるポートフォリオの理解は、継続的なアプリケーションポートフォリオ評価から得られたデータと分析に依存する他のワークストリームへの重要な入力を提供します。

次の図は、ポートフォリオ評価のステージが移行やその他のワークストリームの AWS フェーズにどのように対応しているかを示しています。ポートフォリオ検出と初期計画段階は、通常は最初の 5 週間に評価段階で開始されます。優先順位付けされたアプリケーション評価は、6 週間目と 7 週間目に評価フェーズと動員フェーズにまたがります。ポートフォリオ分析と移行計画の段階は、第 8～14 週の動員段階で行われます。継続的な評価と改善の段階は、移行とモダナイズの段階で、15 週から移行プログラムの終了まで行われます。このタイムラインは参考です。ステージの実際の期間は、プログラム組織全体によって異なります。ポートフォリオ評価ステージは、このフレームワークの外部でも有効であり、任意の移行プログラム構造に組み込むことができます。



- ディスカバリーアクセラレーションと初期計画は、ポートフォリオの現在の理解に焦点を当てています。これには、方向性のあるビジネスケースの作成、移行の基本合理化モデルの確立、初期移行候補の特定が含まれます。
- 優先順位付けされたアプリケーション評価は、詳細な評価、ターゲット状態アーキテクチャの初期設計、短期的に移動できるアプリケーションの特定を通じて、time-to-valueします。アプリケーションをすばやく移行することで、チームは移行の経験を持ち、初期ランディングゾーンやその他のインフラストラクチャコンポーネントなどのクラウド基盤を確立できます。
- ポートフォリオ分析と移行計画は、アプリケーションポートフォリオの完全かつup-to-dateビューを構築することに重点を置いています。このビューは、ポートフォリオデータセットの反復的な強化、データギャップの解消、ビジネスケースの進化、信頼性の高い移行ウェーブプランの作成によって構築されます。
- 継続的な評価と改善は、継続的なアクティビティとして各移行ウェーブの詳細なアプリケーションとテクノロジー評価を生成することで、大規模な移行をサポートします。このステージには、移行ウェーブプランの反復と、最適化とモダナイゼーションのための移行されたワークロードのさらなる分析が含まれます。

次の図は、評価の各ステージの主要なアクティビティと、ポートフォリオレベルの評価とアプリケーションレベルの評価の間でどのようにピボットするかを示しています。ポートフォリオレベルの評価は、ポートフォリオの高レベルの検出と全体的な分析に焦点を当てています。例えば、ポートフォリオデータのソース、アプリケーションとインフラストラクチャのインベントリ、優先順位付け、高レベルの移行戦略、方向性のあるビジネスケースなどです。アプリケーションレベルの評価では、1つ以上のアプリケーションの詳細な検出に焦点を当てます。例えば、アプリケーションのアーキテクチャとテクノロジーレベルでの詳細なアプリケーション検出、ターゲット AWS 設計、詳細な移行戦略などです。ポートフォリオレベルとアプリケーションレベルの評価は、必要な情報の幅と深さを表します。



Discovery アクセラレーションと初期計画

このポートフォリオ評価の最初の段階では、ポートフォリオレベルでデータを取得して分析する最初のステップに焦点を当てます。主な目的は、ビジネスドライバーを特定し、アプリケーションとインフラストラクチャから一般的なデータを収集して、ポートフォリオの初期ビューを取得することです。このデータには、「[データ要件](#)」セクションで説明されているように、アプリケーション名、環境、製品バージョン、重要度、パフォーマンス値などの高レベルの技術的属性とビジネス属性が含まれます。この段階を完了することは、プロジェクトの範囲を理解し、初期移行候補を特定し、ビジネスケースを通知する上で重要です。

このステージの主な結果

- 文書化されたビジネス推進要因、成果、目標、技術指針の原則。
- アプリケーションとインフラストラクチャの初期インベントリ、および特定されたデータギャップ。これは、さらに段階的に反復および改良されるポートフォリオの初期ビューです。
- アプリケーションごとの指標となる移行戦略。例えば、リホスト、リプラットフォームなどです。これは、指標となる労力とタイムラインを決定するのに役立ちます。
- 方向性のあるビジネスケースと移行にかかる推定コスト。
- 初期移行候補のリスト (例: 3-5 アプリケーション)。
- 次のステップを定義します。

初期評価データ要件について

データ収集にはかなりの時間がかかり、どのデータが必要でいつ必要かが明確でない場合、簡単にブロックされる可能性があります。重要なのは、このステージの結果には、少なすぎるデータと多すぎるデータのバランスを理解することです。このポートフォリオ評価の初期段階に必要なデータと忠実度レベルに焦点を当てるには、データ収集に反復的なアプローチを採用します。

データソースとデータ要件

最初のステップは、データソースを特定することです。まず、データ要件を満たすことができる組織内の主要な利害関係者を特定します。これらは通常、サービス管理、運用、キャパシティプランニング、モニタリング、サポートチームのメンバー、およびアプリケーション所有者です。これらのグループのメンバーとの作業セッションを確立します。データ要件を伝え、データを提供できるツールと既存のドキュメントのリストを取得します。

これらの会話をガイドするには、次の一連の質問を使用します。

- 現在のインフラストラクチャとアプリケーションのインベントリはどの程度正確で最新ですか？たとえば、会社設定管理データベース (CMDB) の場合、ギャップがどこにあるかはわかっていますか？
- CMDB (または同等のもの) を更新するアクティブなツールやプロセスはありますか？その場合、どのくらいの頻度で更新されますか？最新の更新日はいつですか？
- CMDB や同等のものなどの現在のドキュメントには、application-to-infrastructure間のマッピングが含まれていますか？各インフラストラクチャアセットはアプリケーションに関連付けられていますか？各アプリケーションはインフラストラクチャにマッピングされていますか？
- ドキュメントには、各製品のライセンスとライセンス契約のカタログが含まれていますか？
- ドキュメントには依存関係データが含まれていますか？サーバーからサーバー、アプリケーションからアプリケーション、アプリケーション、サーバーからデータベースへの通信データが存在することに注意してください。
- 環境では、アプリケーションとインフラストラクチャの情報を提供できる他のどのようなツールを利用できますか？データソースとして使用できるパフォーマンス、モニタリング、リポジトリ、ナレッジベース、管理ツールが存在することに注意してください。
- アプリケーションやインフラストラクチャをホストするデータセンターなど、さまざまな場所は何ですか？

これらの質問に回答したら、特定されたデータソースを一覧表示します。次に、忠実度レベルまたは信頼レベルをそれぞれのレベルに割り当てます。ツールなどのアクティブなプログラムソースから最近 (30 日以内に) 検証されたデータは、最高レベルの忠実度を持ちます。静的データは、忠実度が低く、信頼度が低いと見なされます。静的データの例としては、ドキュメント、ワークブック、手動で更新された CMDBs、プログラムで管理されていないその他のデータセット、または最終更新日が 60 日を超えているデータセットなどがあります。

次の表のデータ忠実度レベルを例として示します。適切な忠実度レベルを決定するために、前提とそれに関連するリスクに対する最大の許容度の観点から組織の要件を評価することをお勧めします。表では、組織の知識とは、文書化されていないアプリケーションとインフラストラクチャに関する情報を指します。

データソース	忠実度レベルの評価	ポートフォリオカバレッジ	コメント
--------	-----------	--------------	------

の専門知識	低 - 正確なデータの最大 25%、75% の想定値、またはデータが 150 日以上経過している。	低	重要アプリケーションに焦点を当てた希少
Knowledge base	中低 - 正確なデータの 35 ~ 40%、65 ~ 60% の想定値、またはデータが 120 ~ 150 日経過しています。	中	手動で管理され、詳細レベルに一貫性がない
CMDB	中 - 正確なデータの約 50%、約 50% の想定値、またはデータが 90 ~ 120 日経過しています。	中	混合ソースからのデータ、複数のデータギャップを含む
VMware vCenter のエクスポート	中 ~ 高 - 正確なデータの 75 ~ 80%、25 ~ 20% の想定値、またはデータが 60 ~ 90 日経過しています。	高	仮想化された資産の 90% をカバー
アプリケーションパフォーマンスのモニタリング	高 - ほぼ正確なデータ、約 5% の想定値、またはデータが 0 ~ 60 日経過しています。	低	重要な本番稼働システムに限定 (アプリケーションポートフォリオの 15% をカバー)

次の表は、各アセットクラス (アプリケーション、インフラストラクチャ、ネットワーク、移行) に必要なデータ属性とオプションのデータ属性、特定のアクティビティ (インベントリまたはビジネスケース)、およびこの評価ステージで推奨されるデータ忠実度を示しています。テーブルでは、次の略語を使用します。

- R、必須

- (D)、ディレクティブビジネスケースの場合、総所有コスト (TCO) の比較とディレクティブビジネスケースに必要
- (F)、TCO 比較に必要な全方向性ビジネスケースと、移行とモダナイゼーションのコストを含む方向性ビジネスケースの場合
- O、オプション
- 該当なし、該当なし

アプリケーション

属性名	説明	インベントリと優先順位付け	ビジネスケース	推奨される忠実度レベル (最小)
一意の識別子	たとえば、アプリケーション ID などです。通常、既存の CMDBs またはその他の内部インベントリと管理システムで使用できます。一意の IDs 組織で定義されていない場合は必ず作成することを検討してください。	R	R (D)	高
アプリケーション名	このアプリケーションが組織で認識される名前。必要に応じて、市販の (COTS) off-the-shelf ベンダーと製品名を含めます。	R	R (D)	やや高い

COTS ですか？	はいまたはいいえ。これが商用アプリケーションか内部開発か	R	R (D)	やや高い
COTS 製品とバージョン	商用ソフトウェア製品名とバージョン	R	R (D)	中
説明	プライマリアプリケーション関数とコンテキスト	R	O	中
緊急性	例えば、戦略的アプリケーションや収益を生み出すアプリケーション、重要な機能のサポートなど	R	O	やや高い
タイプ	データベース、顧客関係管理 (CRM)、ウェブアプリケーション、マルチメディア、IT 共有サービスなど	R	O	中
環境	例: 本番稼働前、本番稼働前、開発、テスト、サンドボックス	R	R (D)	やや高い

コンプライアンスと規制	ワークロードに適用されるフレームワーク (HIPAA、Sox、PCI-DSS、ISO、SOC、FedRAMP など) と規制要件	R	R (D)	やや高い
依存関係	内部および外部のアプリケーションまたはサービスへのアップストリームとダウンストリームの依存関係。運用要素 (メンテナンスサイクルなど) などの非技術的な依存関係	O	O	やや低い
インフラストラクチャマッピング	アプリケーションを構成する物理アセットや仮想アセットへのマッピング	R	O	中
ライセンス	商品ソフトウェアライセンスタイプ (Microsoft SQL Server Enterprise など)	O	R	やや高い

Cost	ソフトウェアライセンス、ソフトウェアオペレーション、メンテナンスのコスト	O	O	中
------	--------------------------------------	---	---	---

インフラストラクチャ

属性名	説明	インベントリと優先順位付け	ビジネスケース	推奨される忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID などです。通常、既存の CMDBs またはその他の内部インベントリと管理システムで使用できます。一意の IDs 組織で定義されていない場合は必ず作成することを検討してください。	R	R	高
ネットワーク名	ネットワーク内のアセット名 (ホスト名など)	R	O	やや高い
DNS 名 (完全修飾ドメイン名、または FQDN)	[DNS 名]	O	O	中

IP アドレスと ネットマスク	内部 IP アドレス および/またはパ ブリック IP アド レス	O	O	やや高い
アセットタイプ	物理サーバーま たは仮想サー バー、ハイパー バイザー、コン テナ、デバイ ス、データベ ースインスタンス など。	R	R	やや高い
製品名	商用ベンダーと 製品名 (VMware ESXi、IBM Power Systems、E xadata など)	R	R	中
オペレーティン グシステム	例: REHL 8、Windows Server 2019、AIX 6.1	R	R	やや高い
設定	割り当てられた CPU、コア数、 コアあたりの スレッド数、合 計メモリ、スト レージ、ネット ワークカード	R	R	やや高い

使用率	CPU、メモリ、ストレージのピークと平均。データベースインスタンスのスループット。	O	O	やや高い
ライセンス	商品ライセンスタイプ (RHEL Standard など)	O	R	中
アプリケーションマッピング	このインフラストラクチャで実行されるアプリケーションまたはアプリケーションコンポーネント	R	O	中
Cost	ハードウェア、メンテナンス、オペレーション、ストレージ (SAN、NAS、オブジェクト)、オペレーティングシステムライセンス、ラックスペース共有、データセンターのオーバーヘッドなど、ベアメタルサーバーのフルロードコスト	O	O	やや高い

ネットワーク

属性名	説明	インベントリと 優先順位付け	ビジネスケース	推奨される忠実 度レベル (最小)
パイプのサイズ (MB/秒)、冗長 性 (Y/N)	現在の WAN リ ンク仕様 (例: 1000 Mb/秒冗長)	O	R	中
接続された口 ヶーション	このリンクで接 続された名前付 きの場所	O	O	中
リンク使用率	ピーク使用率と 平均使用率、 アウトバウンド データ転送 (GB/ 月)	O	R	中
レイテンシー (ミ リ秒)	接続先間の現在 のレイテンシー	O	O	中
Cost	1 か月あたりの 現在のコスト	該当なし	O	中

移行

属性名	説明	インベントリと 優先順位付け	ビジネスケース	推奨される忠実 度レベル (最小)
リホストコスト	各ワークロー ド (人日) の顧客 とパートナーの 労力、1 日あた りの顧客とパー トナーのコスト 率、ツールコス	該当なし	R (F)	やや高い

	ト、ワークロード数			
リプラットフォームのコスト	各ワークロード (人日) の顧客とパートナーの労力、1 日あたりの顧客およびパートナーのコスト率、ワークロードの数	該当なし	R (F)	やや高い
リファクタリングコスト	各ワークロード (人日) の顧客とパートナーの労力、1 日あたりの顧客およびパートナーのコスト率、ワークロードの数	該当なし	R (F)	やや高い
リタイアコスト	サーバー数、平均廃止コスト	該当なし	O	やや高い
ランディングゾーン	既存の (Y/N)、必要な AWS リージョンのリスト、コストの再利用	該当なし	R (F)	やや高い

人と変化	クラウド運用と開発でトレーニングするスタッフ数、1人あたりのトレーニングコスト、1人あたりのトレーニング時間のコスト	該当なし	R (F)	やや高い
時間	対象範囲内のワークロード移行期間 (月)	O	R (F)	やや高い
並列コスト	移行中に現状のコストを削除できる時間枠とレート	該当なし	O	やや高い
並列コスト	移行中に AWS の製品やサービス、およびその他のインフラストラクチャコストが導入される時間枠とレート	該当なし	O	やや高い

検出ツールの必要性の評価

組織には検出ツールが必要ですか？ ポートフォリオ評価には、アプリケーションとインフラストラクチャに関する信頼性の高いup-to-dateデータが必要です。ポートフォリオ評価の初期段階では、仮定を使用してデータギャップを埋めることができます。ただし、進捗状況に応じて、忠実度の高いデータは、移行計画を成功させ、ターゲットインフラストラクチャを正しく見積もってコストを削減し、メリットを最大化するのに役立ちます。また、依存関係を考慮し、移行の落とし穴を回避する実装を有効にすることで、リスクを軽減します。クラウド移行プログラムにおける検出ツールの主なユースケースは、以下を通じてリスクを軽減し、データの信頼性を高めることです。

- 自動またはプログラムによるデータ収集により、検証済みで信頼性の高いデータが得られる
- データの取得速度の加速、プロジェクトの速度の向上、コストの削減
- 通常 CMDBs では利用できない通信データや依存関係など、データの完全性のレベルが向上
- 自動アプリケーション識別、TCO 分析、予測実行率、最適化レコメンデーションなどのインサイトの取得
- 信頼性の高い移行ウェーブプランニング

システムが特定の場所に存在するかどうか不明な場合、ほとんどの検出ツールはネットワークサブネットをスキャンし、ping または Simple Network Management Protocol (SNMP) リクエストに応答するシステムを検出できます。すべてのネットワークまたはシステム設定で ping または SNMP トラフィックが許可されるわけではありません。これらのオプションについて、ネットワークチームや技術チームと話し合います。

アプリケーションポートフォリオの評価と移行のさらなる段階は、正確な依存関係マッピング情報に大きく依存します。依存関係マッピングは、AWS に必要なインフラストラクチャと設定 (セキュリティグループ、インスタンスタイプ、アカウントの配置、ネットワークルーティングなど) を理解します。また、同時に移動する必要があるアプリケーション (低レイテンシーネットワーク経由で通信する必要があるアプリケーションなど) のグループ化にも役立ちます。

検出ツールを決定するときは、評価プロセスのすべての段階を考慮し、データ要件を予測することが重要です。データギャップはブロック要因になる可能性があるため、将来のデータ要件とデータソースを分析してそれらを予測することが重要です。フィールドの経験上、ほとんどの停止した移行プロジェクトには、スコープ内のアプリケーション、関連するインフラストラクチャ、およびそれらの依存関係が明確に識別されないデータセットが限られています。この識別の欠如は、誤ったメトリクス、決定、遅延につながる可能性があります。up-to-dateデータを取得することは、移行プロジェクトを成功させるための最初のステップです。

検出ツールを選択する方法

市場内のいくつかの検出ツールには、さまざまな機能が用意されています。要件を検討してください。また、組織に最適なオプションを決定します。移行用の検出ツールを決定する際の最も一般的な要因は次のとおりです。

セキュリティ

- ツールによって収集されるデータ
- ツールデータリポジトリまたは分析エンジンにアクセスするための認証方法は何ですか？

- データにアクセスできるユーザーと、ツールにアクセスするためのセキュリティコントロールは何ですか？
- ツールによるデータ収集方法 専用の認証情報が必要ですか？
- ツールがシステムにアクセスしてデータを取得するには、どのような認証情報とアクセスレベルが必要ですか？
- ツールコンポーネント間でのデータ転送方法
- このツールは、保管中および転送中のデータ暗号化をサポートしていますか？
- データは環境内外の 1 つのコンポーネントに一元化されていますか？
- ネットワークとファイアウォールの要件は何ですか？

セキュリティチームが検出ツールに関する早期の会話に関与していることを確認します。

データ主権

- データはどこに保存および処理されますか？
- このツールは Software as a Service (SaaS) モデルを使用していますか？
- 環境の境界内にすべてのデータを保持する可能性はありますか？
- データが組織の境界を離れる前に、データをスクリーニングできますか？

データレジデンシー要件の観点から、組織のニーズを考慮してください。

アーキテクチャ

- ツールをデプロイするために必要なインフラストラクチャと、さまざまなコンポーネントは何ですか？
- 複数のアーキテクチャまたはデプロイモデルを利用できますか？
- このツールは、エアロックされたセキュリティゾーンへのコンポーネントのインストールをサポートしていますか？

パフォーマンス

- データ収集がシステムに与える影響

互換性と範囲

- このツールは、私の製品とバージョンのすべてまたはほとんどをサポートしていますか？ ツールドキュメントを確認して、サポートされるプラットフォームをスコープに関する現在の情報と照らし合わせて確認します。
- ほとんどのオペレーティングシステムはデータ収集に対応していますか？ オペレーティングシステムのバージョンがわからない場合は、検出ツールのリストを、サポートされているシステムの範囲が広いものに絞り込もうとします。

収集方法

- ツールでは、各ターゲットシステムにエージェントをインストールする必要がありますか？
- エージェントレスデータ収集をサポートしていますか？
- エージェントとエージェントレスは同じ機能を提供しますか？
- 収集プロセスとは

機能

- 利用可能な機能は何ですか？
- 総所有コスト (TCO) と推定 AWS クラウド 実行率を計算できますか？
- 移行計画をサポートしていますか？
- パフォーマンスを測定しますか？
- ターゲット AWS インフラストラクチャを推奨できますか？
- 依存関係マッピングを実行しますか？
- どのレベルの依存関係マッピングが提供されますか？
- API アクセスを提供しますか？ (たとえば、データを取得するためにプログラムでアクセスできますか？)

強力なアプリケーションとインフラストラクチャの依存関係マッピング機能を持つツールと、通信パターンからアプリケーションを推測できるツールを検討してください。

Cost

- ライセンスモデルとは

- ライセンスの料金はいくらですか？
- 各サーバーの料金ですか？ 階層料金ですか？
- オンデマンドでライセンスできる機能が限られているオプションはありますか？

検出ツールは、通常、移行プロジェクトのライフサイクル全体で使用されます。予算が限られている場合は、少なくとも 6 か月を検討してください。ただし、検出ツールがない場合、通常、手動作業と内部コストが増加します。

サポートモデル

- デフォルトでは、どのレベルのサポートが提供されますか？
- サポートプランはありますか？
- インシデント対応時間はどれくらいですか？

プロフェッショナルサービス

- ベンダーは検出出力を分析するためのプロフェッショナルサービスを提供していますか？
- このガイドの要素について説明できますか？
- ツール + サービスの割引やバンドルはありますか？

Tip

検出ツールを検索して評価するには、[検出、計画、推奨事項](#)のサイトを使用します。

検出ツールの推奨機能

複数のツールからのデータのプロビジョニングと時間の経過に伴う組み合わせを回避するには、検出ツールで以下の最小機能をカバーする必要があります。

- ソフトウェア – 検出ツールは、実行中のプロセスとインストールされたランタイム、パッケージ、フレームワークを特定する必要があります。
- 依存関係マッピング – ネットワーク接続情報を収集し、サーバーと実行中のアプリケーションのインバウンドとアウトバウンドの依存関係マップを構築する必要があります。また、検出ツールは、通信パターンに基づいてインフラストラクチャのグループからアプリケーションを推測する必要があります。

- プロファイルと設定の検出 – CPU ファミリー (x86、PowerPC など)、CPU コア数、メモリサイズ、ディスク数とサイズ、ネットワークインターフェイスなどのインフラストラクチャプロファイルをレポートできる必要があります。
- ネットワークストレージ検出 – ネットワーク接続ストレージ (NAS) からネットワーク共有を検出してプロファイリングできる必要があります。
- データベース検出 – エンジンタイプやバージョンなど、データベース、インスタンス、スキーマを検出できる必要があります。
- パフォーマンス – CPU、メモリ、ディスク、ネットワークのピーク使用率と平均使用率をレポートできるはずです。
- ギャップ分析 – データ量と忠実度に関するインサイトを提供できるはずです。
- ネットワークスキャン – ネットワークサブネットをスキャンし、不明なインフラストラクチャセットを検出できる必要があります。
- レポート – 収集と分析のステータスを提供できる必要があります。
- API アクセス – 収集されたデータにアクセスするためのプログラムによる手段を提供できる必要があります。

考慮すべき追加機能

- TCO 分析により、現在のオンプレミスコストと予測コストのコストを比較できます AWS。
- リホストおよびリプラットフォームシナリオにおける Microsoft SQL Server および Oracle システムのライセンス分析と最適化の推奨事項。
- 移行戦略の推奨事項 (検出ツールは、現在のテクノロジーに基づいてデフォルトの移行 R タイプの推奨事項を作成できますか?)
- インベントリのエクスポート (CSV または同様の形式)
- 適切なサイズのレコメンデーション (たとえば、推奨されるターゲット AWS インフラストラクチャをマッピングできますか?)
- 依存関係の視覚化 (たとえば、依存関係マッピングをグラフィカルモードで視覚化できますか?)
- アーキテクチャビュー (たとえば、アーキテクチャ図を自動的に作成できますか?)
- アプリケーションの優先順位付け (移行の優先順位付け基準を作成するために、アプリケーション属性とインフラストラクチャ属性に重みや関連性を割り当てることはできますか?)
- ウェーブプランニング (アプリケーションの推奨グループや移行ウェーブプランの作成など)
- 移行コストの見積もり (移行にかかる労力の見積もり)

デプロイに関する考慮事項

検出ツールを選択して調達したら、以下の質問を検討して、ツールを組織にデプロイするチームとの会話を促進します。

- サーバーまたはアプリケーションはサードパーティーによって運用されていますか？これにより、チームが関与し、従うプロセスが指示される可能性があります。
- 検出ツールをデプロイするための承認を取得するための大まかなプロセスは何ですか？
- サーバー、コンテナ、ストレージ、データベースなどのシステムにアクセスするための主な認証プロセスは何ですか？サーバー認証情報はローカルですか、それとも一元管理されていますか？認証情報を取得するプロセスは何ですか？システムからデータを収集するには、認証情報が必要です(コンテナ、仮想サーバーまたは物理サーバー、ハイパーバイザー、データベースなど)。特にこれらのアセットが一元化されていない場合、各アセットに接続するための検出ツールの認証情報を取得するのは難しい場合があります。
- ネットワークセキュリティゾーンの概要は何ですか？ネットワーク図は利用できますか？
- データセンターでファイアウォールルールをリクエストするプロセスは何ですか？
- データセンターの運用(検出ツールのインストール、ファイアウォールリクエスト)に関連する現在のサポートサービスレベルアグリーメント(SLAs)は何ですか？

ビジネスドライバーと技術ガイドの原則

ビジネスドライバー

組織が既にクラウドへの移行を決定しているか、その決定に近づいているかにかかわらず、クラウド移行のビジネス推進要因を定義して文書化すると、移行の理由が明確になります。理由が文書化されたら、移行対象とその移行方法を定義できます。このアクティビティは重要です。次のステップを知らせ、ガイドするために、プロセスのできるだけ早い段階で行うことをお勧めします。

ディスカッションに参加すべきステークホルダーを特定し、推進要因を文書化します。通常、CxOs、シニアマネージャー、組織内の主要なテクノロジーリーダー、および独自の顧客です。顧客がこのディスカッションに参加する可能性は低いですが、組織内の1人以上の人物が顧客の見解と目標を表すように指定することをお勧めします。

ビジネスドライバーは、結果が達成されたかどうかを検証するために、移行ジャーニー全体で測定できるメトリクスにリンクする必要があります。会社の戦略的目標と年次レポートは、出発点となる可能性があります。

クラウドへの移行の結果として、既存のメトリクスと予測されたメトリクスに基づいて、会社のありたい場所に会話を集中させます。目標とビジネス成果を考慮します。また、クラウドの導入が増えるにつれて、どのような成功が得られるかを検討してください。

次に、各ドライバーの重要度レベルを確立します。優先順位は何ですか？ 予想される利点は何ですか？ メリットはビジネスの目標と成果にどのように役立ちますか？ アプリケーションポートフォリオ評価の文脈では、回答は移行するワークロードに優先順位を付け、技術的な指針を確立するのに役立ちます。ただし、ビジネスドライバーは移行プログラム全体を定義し、影響します。

技術ガイドの原則

テクニカルガイドの原則は、ポートフォリオ評価の後期段階での移行戦略の選択に役立ちます。現在のステージでは、それらを特定することに重点を置いています。

指針の原則は、ビジネス目標と成果から導き出される一般的なテクノロジー関連およびアプローチ関連の意思決定として確立できます。

例えば、ある企業にはコスト削減の主な目標があり、望ましい成果は 12 か月以内に特定の日付までにオンプレミスデータセンターを閉鎖することです。その結果の指針となる原則は、可能な限りリホスト移行戦略を使用して、すべてのアプリケーションをクラウドにリフトアンドシフトすることです。この場合、lift-and-shiftアプローチは短期的な移行成果を加速します。アプリケーションがオンプレミスデータセンターの外に移動した後、移行されたワークロードを最適化またはモダナイズするために、他のビジネスドライバーに集中できます。

テクニカルガイドの原則を確立するには、まずビジネス推進要因を分析します。ビジネス目標と成果を達成するテクノロジーとテクニックのリストを特定します。次に、リストを絞り込み、適合性または好みに基づいて関連性の順序を割り当てて、望ましい結果を達成します。

移行の計画と実行に関係する人々に指針を文書化し、伝えます。原則と実際の実装の間の懸念と潜在的な競合を強調します。

次の表は、ビジネスドライバーと技術ガイドの原則の例を示しています。

ビジネスドライバー	結果	メトリクス	テクニカルガイドの原則
イノベーションを加速します。	競争力の向上、ビジネスの俊敏性の向上	1 日または 1 か月あたりのデプロイ数、四半期ごとにリリースされ	マイクロサービスと DevOps 運用モデルを使用してアプリケーションの差別化をリ

		た新機能、顧客満足度スコア、実験数	ファクタリングし、新機能の俊敏性と市場投入までの時間を短縮します。
運用コストとインフラストラクチャコストを削減します。	需要と供給が一致する伸縮自在なコストベース (使用した分に対する支払い)	時間の経過に伴う支出の変動	1. インフラストラクチャの適切なサイズ設定でアプリケーションをリホストします。 2. 使用率が低い、または使用していないアプリケーションを廃止します。
運用の耐障害性を高めます。	稼働時間を改善し、平均復旧時間を短縮	SLAs、インシデントの数	1. アプリケーションを最新のサポートされているオペレーティングシステムバージョンにリプラットフォームします。 2. 重要なアプリケーションに高可用性アーキテクチャを実装します。
データセンターを終了します。	6～12 か月以内の日付でデータセンターを閉鎖する	サーバー移行の速度	を使用してアプリケーションをリホストします AWS Transform MGN 。

オンプレミスのまま で、俊敏性と耐障害 性を高めます。	オンプレミスのままの 競争力と稼働時間を改 善	1 日または 1 か月あた りのデプロイ数、四半 期あたりの新機能リ リース数、SLAs、イ ンシデント数	1. 機能をクラウドに 拡張することで、シ ステムをモダナイズ します。 2. へのリホストまた はリプラットフォーム について評価しま す AWS Outposts。
-----------------------------------	-------------------------------	---	---

また、移行戦略に役立つアーキテクチャ原則を定義することも重要です。たとえば、移行したすべてのアプリケーションがすべての通信を暗号化する必要がある AWS クラウドのアーキテクチャ原則を導入できます。この種の原則が現在オンプレミスで適用されていない場合、移行チームは移行の一環として変更を実装するよう強制され、労力と計画に影響します。すべてのアーキテクチャ原則を定義し、関連するチームと共有します。

データ収集の開始

データ収集は、アプリケーションとインフラストラクチャからメタデータを収集するプロセスです。このプロセスは、評価のすべての段階で繰り返されます。各ステージでは、データ量と忠実度が増加します。この段階では、初期インベントリの確立に役立つ一般的なデータを収集することに重点を置いています。インベントリは方向性のあるビジネスケースと計画の作成に使用され、初期移行候補の特定に使用されます。

現在のデータソースを特定したら、できるだけ多くのシステムから情報を収集することをお勧めします。詳細については、このステージの[データ要件](#)を参照してください。

このアプローチには、現在のポートフォリオビューと、アプリケーションとサービスに関する組織の知識の更新に役立つ利点があります。また、移動する対象を決定するのに役立ちます。推奨されるアプローチは、設定管理データベース (CMDB) 出力や情報技術サービス管理 (ITSM) システムなどの既存のデータを確認することです。次に、追加のデータ収集の対象となるアセットのリストを作成します。組織で移行の対象範囲と対象外の範囲が完全に明確である場合は、対象範囲内のシステムにデータ収集を制限する可能性があります。

ポートフォリオを構築するときは、アプリケーションとその環境またはソフトウェアリリースのライフサイクルを考慮してください。例えば、顧客関係管理 (CRM) アプリケーションを識別し、テスト環境、開発環境、本番稼働環境を指定する代わりに、3 つのアプリケーション (CRM-Test、CRM-

Dev、CRM-Prod など)を一覧表示します。または、CRM 名を使用して各環境に一意の ID を割り当て、データリポジトリに個別のレコードとして表示します。これは、これらの環境の移行を個別に計画および追跡するのに役立ちます。たとえば、非本番環境を最初に移行できます。環境に従ってアプリケーションのインスタンスを一覧表示することで、移行を明確に管理および管理できます。

データ収集に、どのアプリケーションまたはサーバーが特定のデータセンターまたはソースロケーションにあるかが不確実な場合があります。このような場合は、既存の管理ツールからベアメタルリストとハイパーバイザーリストを取得すると便利です。たとえば、ハイパーバイザーに接続して、データ収集の対象となる仮想マシンのリストを取得できます。

既存のデータソースを組み合わせると、最初の出力が不完全になる可能性があることに注意してください。重要なのは、このステージの[データ要件](#)と既存のソースから取得できる内容の観点からギャップ分析を実行することです。完全性の割合とデータ忠実度レベルを対比することが重要です。低忠実度ソースからのより高い完全性レベルには、分析の欠陥につながる可能性のあるいくつかの仮定が含まれます。この評価段階では最大データ忠実度は必要ありませんが、データソースは少なくとも中～中程度の忠実度にするをお勧めします。これらの数値を、データギャップを埋めるための仮定の使用を含め、リスクに対する組織の許容度と照らし合わせます。

ギャップ分析は、作業しているデータの量と品質を理解するのに役立ちます。この分析は、方向性のあるビジネスケースを作成し、移行のアプリケーションに優先順位を付けるために行う必要がある前提のレベルを確立するのに役立ちます。検出ツールは、ギャップを埋め、忠実度の高いデータを収集するのに役立ちます。データの信頼性を高め、移行の成果を加速させるには、できるだけ早く検出ツールをデプロイすることをお勧めします。新しいツールの内部調達、セキュリティ、実装プロセスが完了するまでに数週間かかる可能性があるため、早期のアクションも重要です。

この段階では、コミュニケーションプランまたはケイデンスとスコープ変更制御メカニズムを確立することをお勧めします。これにより、ステークホルダーが事前に計画を立て、リスクを軽減できるように、ステークホルダーに最新情報を伝えることができます。明確なコミュニケーションの重要な要素は、アプリケーションポートフォリオと関連インフラストラクチャの信頼できる単一のソースを定義することです。複数の記録システム、アプリケーションおよびインフラストラクチャリストを保持しないでください。バージョニングとオンラインコラボレーションをサポートするデータを 1 か所 (データベース、ツール、スプレッドシートなど) に保持し、所有者を割り当てます。

優先順位付けと移行戦略

移行計画の重要な要素は、優先順位付け基準を確立することです。この演習のポイントは、アプリケーションを移行する順序を理解することです。戦略は、優先順位モデルを進化させるために反復的かつ段階的なアプローチを取ることです。

アプリケーションの優先順位付け

この評価ステージでは、リスクと複雑さの低いワークロードを優先するための初期基準の確立に焦点を当てます。これらのワークロードは、パイロットアプリケーションに適しています。初期移行で低リスクで複雑さの低いワークロードを使用すると、リスクが軽減され、チームは経験を積むことができます。これらの基準は、移行ウェーブプランの作成時にビジネスの推進要因に合わせて、さらなる評価段階で進化します。

初期基準では、クラウドでサポートされているインフラストラクチャで実行され、非本番環境から実行される、少数の依存関係を持つアプリケーションを優先する必要があります。たとえば、開発環境またはテスト環境で 0~3 個の依存関係をそのままリホストできるアプリケーションです。これらの基準は、クラウド導入の成熟度と信頼度のレベルに応じて、パイロットアプリケーションと、最初と 2 番目の移行ウェーブを定義するために有効です。

使用する初期基準の決定

最初のワークロードの優先順位付けに使用するデータポイントを 2~10 個選択します。これらのデータポイントは、最初のデータ収集から取得されます ([データ収集](#) セクションを参照)。

次に、各データポイントの可能な値ごとにスコアまたは重みを定義します。たとえば、環境属性が選択されていて、可能な値が本番、開発、テストの場合、各値にスコアが割り当てられ、数値が大きいほど優先度が高くなります。オプションですが、各データポイントに重要度または関連性の乗数を割り当てることをお勧めします。このオプションのステップでは、より重要なことを強調する高レベルの差別化要因を提供します。これにより、値にスコアを割り当てる際に繰り返し基準を一致させることができます。

最初のいくつかの移行ウェーブで低リスクのシンプルなアプリケーションを優先する戦略に基づいて、次の表に属性の選択例とその値の割り当てを示します。

属性 (データポイント)	使用できる値	スコア (0~99)	重要性または関連性の乗算係数
環境	テスト	80	高 (1x)
	開発	50	
	本番稼働	20	
ビジネスの重要性	低	60	高 (1x)

	中	40	
	高	20	
規制またはコンプライアンスフレームワーク	なし	60	高 (1x)
	FedRAMP	10	
オペレーティングシステムのサポート	クラウド対応	60	中高 (0.8x)
	クラウドではサポートされていません	10	
コンピューティングインスタンスの数	1 ~ 3	60	中高 (0.8x)
	4-10	40	
	11 以上	20	
依存関係の数	0-3	70	高 (1x)
	4-10	30	
	11 以上	10	
移行戦略	リホスト	70	中 (0.6x)
	リプラットフォーム	30	
	リファクタリングまたは再設計	10	
運用チームのクラウド成熟度または準備状況	高	80	高 (1x)
	中	50	
	低	10	

アプリケーション間の主要な差別化要因として機能する属性を必ず選択してください。そうしないと、条件によって多くのワークロードが同じ優先度を共有します。モデルを適用したら、結果のラン

キングの上部と下部を見て、同意するかどうか確認することをお勧めします。一般的に同意しない場合は、ワークロードのスコアリングに使用した基準を再確認できます。

ランキングを取得したら、ポートフォリオ全体のスコアの分布を確認します。スコア自体は関係ありません。重要なのはスコアの差です。たとえば、上位の合計スコアが 8,000 で、下位スコアが 800 であることがわかります。結果のスコアをヒストグラムとしてプロットして、分布が良好であることを確認することを検討してください。理想的なディストリビューションは、非常に優先度の高いワークロードと非常に優先度の低いワークロードがいくつかあり、標準のベル曲線のようになります。アプリケーションの大部分は中間のどこかにあります。

初期の優先順位付けのもう 1 つの重要な側面は、クラウドの早期導入者になることに関心を示す内部チームまたはビジネスユニットを含めることです。これらは、特に初期に特定のアプリケーションを移行するためのビジネスサポートを取得するためのかなりの手段になる可能性があります。組織でその場合は、上の表にビジネスユニット属性を含めます。アプリケーションを進んで進めるビジネスユニットに高いスコアを割り当てます。ビジネスユニット属性を使用すると、これらのアプリケーションをリストの先頭に配置するのに役立ちます。

結果のランキングに同意したら、上位 5~10 のアプリケーションを選択します。これらは、最初のアプリケーション移行候補になります。3~5 個のアプリケーションを確認するようにリストを絞り込みます。これにより、詳細なアプリケーション評価を実行する際に、的を絞ったアプローチを取ることができます。詳細については、[「優先順位付けされたアプリケーションの評価」](#)を参照してください。

移行用の R タイプの決定

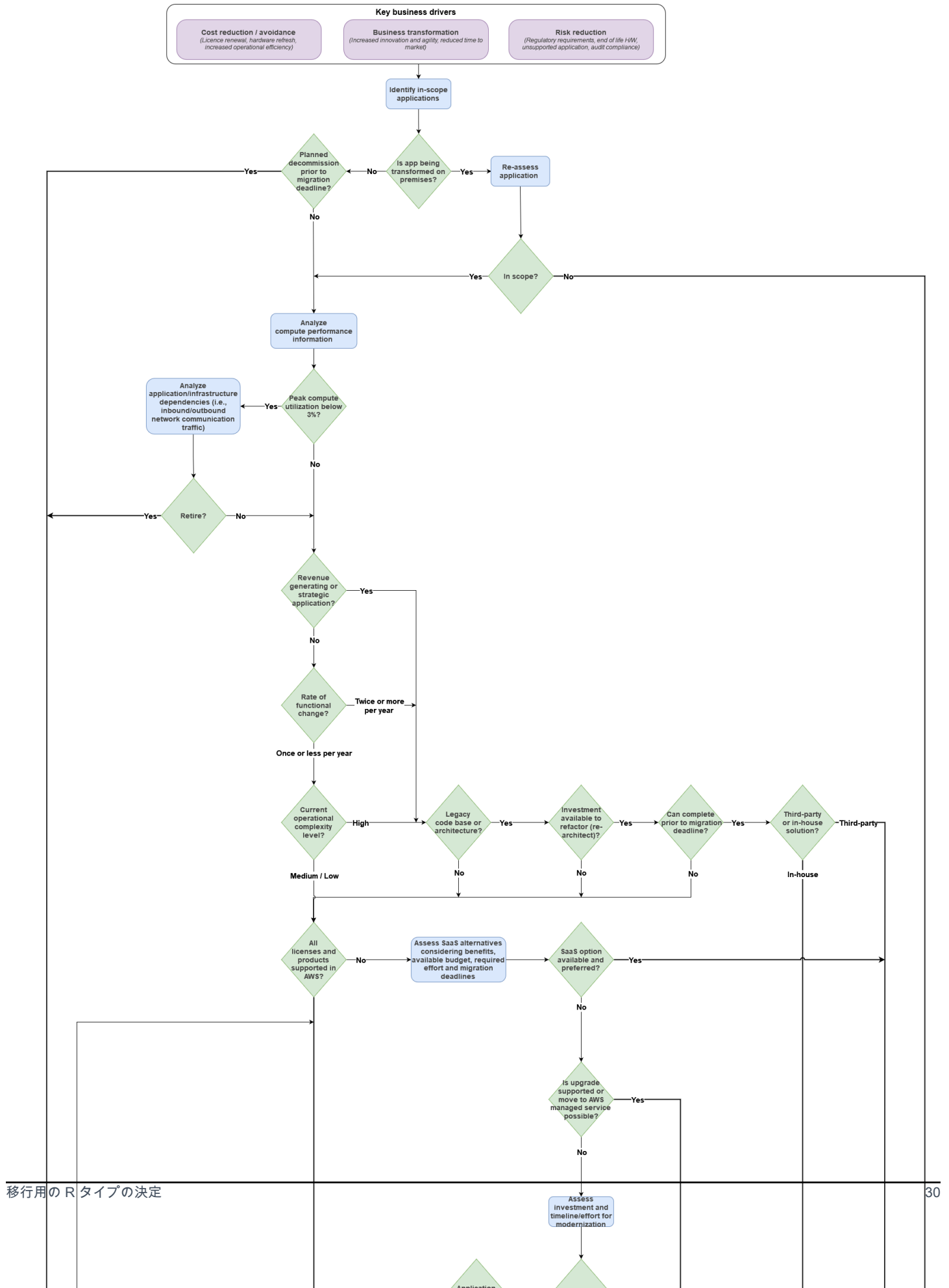
各アプリケーションおよび関連するインフラストラクチャの移行戦略を決定することは、移行速度、コスト、および利点のレベルに影響します。ビジネスドライバー、技術指針、優先順位付け基準、ビジネス戦略など、バランスの取れた要素の組み合わせに基づいて戦略を決定することが重要です。

これらの要因が競合するビューを作成することがあります。例えば、移行の主な推進要因はイノベーションと俊敏性です。同時に、コストを迅速に削減する必要がある場合があります。範囲内のすべてのアプリケーションをモダナイズすると、長期的にコストを削減できますが、事前により大きな投資が必要になります。この場合、1 つのアプローチは、リホストやリプラットフォームなど、労力の少ない戦略を使用してアプリケーションを移行することです。これにより、短期間で迅速な効率とコスト削減を実現できます。次に、コスト削減を後の段階でアプリケーションのモダナイズに再投資し、さらなるコスト削減を実現します。

ただし、すべてのアプリケーションの完全なリホストから始めると、モダナイゼーションのより大きな利点が遅れます。重要なのは、ビジネス戦略アプリケーションがモダナイゼーションに優先され、他のアプリケーションを最初にリホストまたはリプラットフォームしてからモダナイズできるように、移行戦略のバランスを見つけることです。

アプリケーションの移行戦略を決定する方法

この評価段階では、移行戦略の選択をガイドするための初期モデルを組み込むことに重点が置かれています。初期アプリケーションの移行戦略を検証するには、モデルをビジネスドライバーと優先順位付け基準と組み合わせて使用します。デシジョンツリーのデフォルトのロジックは、スコープの初期処理を決定するのに役立ちます。ツリーでは、リファクタリングやリアーキテクトなどの最も複雑なアプローチは、戦略的ワークロード用に予約されています。



初期モデルの最初のステップは、ツリーの上部にあるビジネスドライバーを、組織が定義したビジネスドライバーで更新することです。次に、アプリケーション全体ではなくアプリケーションコンポーネントにツリーを適用します。たとえば、3つのコンポーネント(フロントエンド、アプリケーションレイヤー、データベース)を持つ3層アプリケーションの場合、各コンポーネントはツリーを個別に転送し、特定の戦略とパターンを割り当てる必要があります。これは、場合によっては、特定の階層をリホストまたはリプラットフォームし、他の階層をリファクタリング(リアーキテクト)する必要があるためです。

デシジョンツリーを使用して移行戦略を確立する前に、いくつかのアプリケーションでロジックをテストし、一般的に結果に同意することを確認します。決定木は、その正確性を判断するために必要な分析を置き換えないガイドです。ツリーロジックは、特定のケースに適用されない場合があります。これらのケースを例外として扱い、ツリーロジックを変更するのではなく、オーバーライドの理論的根拠を文書化して、ツリーによって駆動される決定を上書きします。これにより、管理が困難になる可能性のある複数の決定木バージョンを防ぐことができます。一般的なガイダンスでは、ツリーはワークロードの少なくとも70~80%に対して有効である必要があります。残りは例外があります。評価のこの段階では、ツリーロジックの調整は、計画を可能にする初期モデルの確立に集中する必要があります。さらなる反復と改良は、[ポートフォリオ分析](#)や[移行計画](#)など、後の段階で行われます。

方向性のあるビジネスケースの作成

ビジネス全体のステークホルダーは、各ステップで変革のためのビジネスケースを理解し、受け入れる必要があります。

初期段階では、プログラムの計画と確立に必要なリソースを確保できるように、移行プログラムから十分な潜在的な価値をすばやく示すことが重要です。方向性のあるビジネスケースは、早期に収集できる限られたデータで、説得力のあるビジネス価値を達成するための合理的な信頼を提供するように設計されています。

プログラムが確立されると、ビジネスケースがさらに発展します。詳細なケースにより、精度が向上し、プログラムの価値をより完全に把握し、計画の優先順位を把握できます。組織が購入する計画されたビジネス成果を定義して定量化し、プログラムガバナンスオフィスがプログラムを運営し、その成果を測定するためのベースラインを設定します。

方向性のあるビジネスケースの範囲の修正

方向性のあるビジネスケースは通常、2~4週間以内に迅速に組み立てられます。コアチームを確立し、必要に応じてAWSパートナーを関与させ、少なくとも [優先順位付けされたアプリケーション評価](#)、[ポートフォリオ分析](#)、[移行計画](#)の各段階を完了できるように、十分な信頼を生み出す必要があります。

通常、ポートフォリオの移行をサポートする方向性のあるビジネスケースは、次のいずれかとして作成されます。

- 現状のインフラストラクチャ環境と移行後の AWS のサービス アーキテクチャの単純総所有コスト (TCO) の比較。比較は、特定のワークロードボリュームの予想される実行率の差を示しています。
- 現在の純価値 (NPV)、投資収益率 (ROI)、ペイバック期間、変更された内部収益率 (MIRR)、移行コスト AWS を含む への移行と現状維持のための 3~5 年間のキャッシュフロー分析 を示すビジネスケース。

方向性のあるビジネスケースの範囲は通常、次のいずれかに制限されます。

- インフラストラクチャテクノロジーコストの比較
- インフラストラクチャテクノロジーと運用コストの比較

一般的に、ポートフォリオが大きいほど、開発の必要なケースは少なくなります。これは、結果に大きな影響を与えることなく、より広範な仮定を行うことができるためです。ポートフォリオを小さくすると、変更の影響が大きくなるため、詳細が必要になります。

まず、基本インフラストラクチャのコスト比較を構築します。次に、続行する前に、比較が十分に説得力があるかどうかを判断します。通常、400 台を超えるサーバーのポートフォリオは、運用から 3 年以内 AWS、または 5 年以内の 250 台のサーバーで、インフラストラクチャのコスト削減だけでもプラスのビジネスケースになりますが、これは異なる場合があります。より小さなポートフォリオの場合は、より詳細な情報が必要になる場合があります。

逆に、移行範囲の合計が約 5 ワークロードまたは 50 サーバー未満でない限り、回復力の向上やビジネスの俊敏性から派生した値など、この段階で他のビジネス価値のコンポーネントを調べることはめったに役に立ちません。

フォーカス値ドライバー

インフラストラクチャテクノロジーの TCO 比較では、現状のインフラストラクチャコストのモデルと、同等のパフォーマンスと可用性でワークロードを実行するために必要な部品 AWS のサービス表の基本モデルを比較します。多くの最適化を行うことができます。ただし、この段階では、評価が容易で、通常は TCO を約 30% 削減できるため、次のリストに重点が置かれています。

- コンピューティングの伸縮性 – 8x5 (24% の使用)、10x5 (30%)、または 10x6 (36%) を実行している開発サーバーや UAT サーバー、2% を実行しているディザスタリカバリ (DR) サーバーなど、

使用量が 100% ではないサーバーを、使用時にのみ請求されるオンデマンドサービスにマッピングします。

- 削減計画による調達 – コストを最大 75% 削減するための適切な削減計画で、本番稼働用サーバーやその他の使用率の高いサーバー (36% 超) を調達する計画を立てます。オプションには、1 年契約と 3 年契約があり、割引を強化するために前払いレベルが異なります。
- ゾンビの削除 – CPU 使用率が 2% 未満のサーバーを特定し、不要になったことを確認して、コスト分析から削除します。
- コンピューティングの適切なサイズ設定 — CPU とメモリの使用率の時系列データを使用して、サーバーごとに必要なコンピューティング能力とメモリを評価します。次に、適合する Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを選択します。
- リレーショナルデータベース管理システム (RDBMS) ライセンスの適切なサイズ設定 – データベースサーバーでコンピューティングの適切なサイズ設定を行った後、RDBMS ライセンスのニーズを再評価し、Bring Your Own License (BYOL) と Procuring license from を比較し AWS、Amazon Relational Database Service (Amazon RDS) の可能性を調べて削減額を増やします。
- ストレージ – 必要な合計ストレージボリュームを適切なサイズに設定し、ポートフォリオ全体の 1 秒あたりの入出力オペレーション (IOPS) のニーズを特定します。SLAs とコストが異なるオブジェクトストレージに移動できる量を決定します。

データのニーズ

[「初期評価データ要件を理解する」](#)の表は、方向性のあるビジネスケースの各部分の構築に必要なデータと、必須かオプションかを示しています。

ケースを構築するには、初期計画データとコストデータのインフラストラクチャサブセットが必要です。含めるインフラストラクチャを特定する方法は、ビジネス目標によって異なります。

- プログラムの目的は、特定のアプリケーションを移行してモダナイズすることである場合は、共有されるインフラストラクチャを考慮して、アプリケーションが必要とするものに基づいてインフラストラクチャポートフォリオを構築します。
- リースの有効期限が切れるデータセンターから移行するなど、プログラムの目的がインフラストラクチャ中心である場合、インフラストラクチャの TCO 比較にアプリケーションマッピングは必要ありません。

オプションとしてマークされたデータ (サーバーの CPU やメモリのピーク使用率など) は通常、標準ベンチマーク値に置き換えることができます。これについては、AWS パートナーまたは AWS プロフェッショナルサービスにお問い合わせください。または、ポートフォリオの一部で利用可能なデー

タポイント (ハイパーバイザーによって収集されたデータなど) から値を推定することもできます。ポートフォリオが大きいほど、より正確です。

インフラストラクチャの TCO 比較の構築

インフラストラクチャの TCO 比較を構築するには、ツールが不可欠です。[AWS プロフェッショナルサービス](#)または [AWS パートナー](#)は、特により広範な移行プロセスを支援するためにそれらに関与させる予定がある場合は、あらゆる種類の方向性のケースを支援できます。

以下の操作に使用できるツールがあります。

- インベントリデータを収集します。
- 使用率データを収集します。
- インフラストラクチャのコストベンチマークデータをそのまま正確に提供します。
- ゾンビを特定して削除します。
- 適切なサイズ評価を行います。
- 購入オプションを推奨します。
- ソフトウェアライセンスオプションを比較します。
- シンプルなグラフィカルなキャッシュフロー分析を生成します。

AWS からの[移行エバリュエーター](#)は 1 つのオプションです。これらの機能はすべて、無料のマネージドサービスとして提供されます。Migration Evaluator は、AWS アカウントマネージャーまたは AWS Migration Competency Partner を通じて、またはオンラインでリクエストを送信[することでリクエスト](#)できます。Migration Evaluator は、インフラストラクチャテクノロジーの TCO 比較を迅速に生成するためのポイントソリューションとして特別に設計されています。

主な利点。

- 無料
- ツールベースの検出が制限されているインベントリデータのエージェントレス検出または手動設定
- デプロイ、設定、データ収集、ベースケースの構築、または方向性のあるビジネスケースを支援するための専用サポート
- SaaS オペレーションの利便性がありますが、顧客ネットワーク内でデータ収集を完全に実行して、分析エンジンにロードする前にスクラブをサポートできます。
- Microsoft ライセンスの適切なサイズ設定に対する強力なサポート
- 完全なデータエクスポート機能

主な制限事項:

- x86 アーキテクチャサーバーのみを評価する (Windows および Linux)
- ベンチマークをそのままコストデータとして設定またはキャリブレーションする限定オプション
- モデリングオペレーションのコスト最適化はサポートされていません
- 移行コストモデリングのサポートなし
- TCO 比較以外のビジネスケースの構築を直接サポートしていない

アプリケーションスタックや相互依存関係検出などのポートフォリオ検出および分析機能に商用検出ツールを使用することにした場合、通常、インフラストラクチャの TCO 比較も行われます。ポートフォリオの検出と評価のためのツールの使用に関するガイダンスについては、[「検出ツールの必要性の評価」](#)を参照してください。市場をリードするツールの主な機能を確認して比較するには、[「検出、計画、推奨事項の移行ツール」](#)を参照してください。

運用コスト最適化の構築

IT 運用の生産性向上は、多くの場合、移行に大きな価値をもたらします。アマゾン ウェブ サービスでビジネス価値を生み出すため、国際データ株式会社 (IDC) のホワイトペーパーによると、移行後、AWS IT 運用スタッフの生産性は平均 62% 向上します。<https://pages.awscloud.com/rs/112-TZM-766/images/AWS-BV%20IDC%202018.pdf?aliid=1614258770> ただし、サイズ設定と、これらの利点を方向性のあるケースに含めるには、2 つの課題があります。

まず、生産性の向上の全範囲を評価するには、広範なデータ収集が必要であり、[詳細なビジネスケース](#)に適しています。この課題は、単純なベンチマークデータでより簡単に評価およびサイズ設定できるが、それでも大きな利点を示すいくつかの要素に焦点を当てることで解決できます。

第 2 に、コスト削減のソースとして生産性に焦点を当てると、主要な顧客ステークホルダーとプログラムメンバーの間に懸念と否定が生じる可能性があります。メリットがどのように実現されるか、それが影響を受ける人々にとって何を意味するかを明確にしてください。このような問題は、チームの役割を強化するだけであることを明確にすることで回避できます。

- 移行プログラムには、内部運用スタッフを開発し、新しい役割に移行するためのトラックが含まれています。例えば、DevSecOps チームに参加してコードオートメーションとしてインフラストラクチャを構築し、チームの成長を促進するオートメーションをテストします。
- メリットは、オペレーションのアウトソーシング契約の範囲を変更およびサイズ変更することで実現できます。これにより、内部スタッフがより価値の高いアクティビティに集中できるようになります。

考慮すべきオペレーション変換に基づいて、このビジネスケース要素を構築するアプローチ:

- 既存の社内運用チームがある場合は、チームメンバーのスキルを高め、期待される生産性の向上を示します。
- または、現在の運用ソリューションから AWS Managed Services (AMS) または AWS パートナーが提供する代替マネージドサービスに移行します。

最初の変換では、ケースに含めることができる生産性の向上について控えめに財務見積もりを取得するには、以下をお勧めします。

1. 特にサーバー管理オペレーションの生産性に焦点を当てます。オペレーションの労力の大部分を占める傾向があり、より簡単に評価でき、後でより簡単に検証できます。
2. 各フルタイム同等の (FTE) 従業員が管理できるサーバー数のベンチマークに基づいて、必要な人員配置を計算します。オンプレミスでは、この数は約 150 サーバーです。AWS上のサーバー数は約 400 です。
3. これらのメトリクスを EC2 インスタンスの数と比較したオンプレミスサーバーの数に適用します。
4. 節約した時間を運用チーム全体のブレンドコスト率で乗算します。

その後、結果をいずれかの方法で確認するには、結果を次の表に示すロール (IDC ホワイトペーパー [Fostering Business and Organizational Transformation to Generate Business Value with Amazon Web Services](#) から取得したデータ) による平均生産性の向上を大幅に上回らないことを確認します。

ロール	効率の向上
IT インフラストラクチャ管理	62%
IT サポート	59%
アプリケーション管理	43%
データベース管理	19%
アプリケーション開発	25%

2 番目の変換では、対象範囲内のポートフォリオの現在の総オペレーションとサポートコストを、検討中のマネージドサービスのコストと直接比較することで、運用コストの削減を追加できます。

マネージドサービスのコストを取得するには、提案された部品 AWS 表、サービスレベルの選択 (Plus または Premium)、および AMS パッケージ (AMS Accelerate または AMS Advanced) を AWS アカウントマネージャーまたは [AWS Managed Services パートナー](#) に提供します。これにより、変換されたソリューションの AWS のサービス コンポーネントに対するマネージドサービスの合計コストが提供されます。同様に、独自のパラメータに基づいて独自のマネージドサービスパッケージを提供する AWS パートナーから料金を取得できます。

完全な方向性のビジネスケースへの拡張

一般的に、完全な方向性のビジネスケースを構築するには、IT 生産性要素の有無にかかわらず TCO 比較を構築し、すべての移行とモダナイゼーションのコストを見積もります。次に、migrate-and-modernize のシナリオのペアと t-migrate-and-modernize シナリオのペアをカバーするキャッシュフローを作成します。

最も基本的なケースは、移行 t-migrate-and-modernize シナリオが現在の状況であり、migrate-and-modernize シナリオには次の特性がある 1 組のシナリオの準備です。

- トランザクションボリューム、コンピューティング、ネットワーク容量の増加や縮小がない
- ストレージ要件の低ボリュームの着実な増加
- 既存のシステム機能に一致する Quality-of-service 機能 (可用性、耐久性、スループット、パフォーマンスなど)

非常に小さなポートフォリオを除くすべてのポートフォリオについて、これは方向性のあるケースをうまく構築する目的に適合します。これは、先に進むためのマンデートを得るために十分な価値を迅速に示します。

ポートフォリオが小さい場合は、migrate-and-modernize のペアを追加し、クラウド移行の付加価値の他の側面を示すシナリオ t-migrate-and-modernize 方が有益です。次に例を示します。

- その成長が予想されるワークロード全体で中程度と大容量の成長要件の組み合わせ
- 高可用性、DR、耐障害性などの強化された耐障害性を含める
- エッジコンピューティング、コンテンツ配信ネットワーク (CDN)、マルチリージョンデータベースレプリケーションでグローバルパフォーマンスが向上しました。
- プログラムのビジネス上の優先事項となったその他の特定の改善されたサービス品質

これらのシナリオでは、現在のクラウド以外のインフラストラクチャアーキテクチャを新しい仕様に合わせてアップグレードした場合のコストとキャッシュフローへの影響が正確に推定されていることを確認してください。この見積りを取得する最も直接的な方法は、特に移行コンピテンシーを持つ AWS コンサルティングパートナーで、migrate-and-modernizeのシナリオと移行t-migrate-and-modernizeのシナリオの両方をサポートできる場合、システムインテグレーターに見積りをリクエストすることです。

シナリオのペアごとに、以下で構成されるケースを組み立てます。

- 移行およびt-migrate-and-modernizeシナリオのコスト。最も基本的なケースには、以下が含まれます。
 - 現在のインフラストラクチャ設定のビジネスケース期間における総所有コスト
 - コンピューティング、ストレージ、ネットワークトラフィックの消費量の定期的な増加
- migrate-and-modernizeのコスト。シナリオには以下が含まれます。
 - 詳細な検出、移行計画、詳細なビジネスケース開発を含むプログラムのセットアップ、コアチームの確立とスキルの向上、まだ導入されていない場合のランディングゾーンの確立、移行されたワークロードのセキュリティ管理と運用の統合の確立
 - ワークロードの移行とモダナイゼーションのコスト
 - ネットワーク接続、[AWS Snowball Edge](#) や などのデータ移行サービス[AWS DataSync](#)、移行プロセス自体に必要なアーキテクチャの AWS ユーティリティコスト (テストなど) を含む移行インフラストラクチャのコスト
 - ウェーブが稼働するにつれて移行中の AWS ユーティリティコストが増加し、既存のインフラストラクチャコストが AWS ベースのサービスに置き換えられて廃止されるにつれて増加します。
- ストランドアセットの廃止コストと償却

移行とモダナイゼーションプログラムのセットアップの見積もり

プログラムを成功させるためにセットアップするには、ベースライン機能と詳細な計画を構築するための一連の基本的なアクティビティを以前に実行することが必要になる場合があります。これらの基本的なアクティビティには以下が含まれます。

1. ポートフォリオ分析と移行計画のセクションで説明されているように、詳細なポートフォリオ検出、[移行計画](#)、詳細なビジネスケース開発を実行するとともに、使用した検出ツールのコストを文書化します。

2. クラウドビジネスおよびテクニカルコアチームを確立し、トレーニングと雇用を通じて社内スキルを開発します。トレーニングが必要な IT 組織のメンバーを特定し、各メンバーにトレーニング予算を割り当てます。
3. [ランディングゾーン](#)を確立し、必要なコスト、運用、セキュリティガバナンス機能をサポートするように設定します。

AWS コンサルティングパートナーは、項目 1 と 3 の見積りを提供できます。

移行とモダナイゼーションのコストの見積もり

方向性のあるビジネスケースの目標を達成し、次のフェーズに進むのに十分な商用の可能性を示すには、移行とモダナイゼーションのコスト見積もりをできるだけ基本的なままにします。

そのためには、次の移行戦略に該当するアプリケーションに焦点を当てて、方向性のあるビジネスケースを準備することをお勧めします。

- リタイア
- 保持する
- リロケート
- リホスト
- リプラットフォーム
- 再購入

通常、ワークロードの約 70% はリホスト、再配置、またはリプラットフォームでき、さらに 5% はリタイアできます。移行戦略によるアプリケーションの評価は、通常、コスト削減ケースの中核をなします。

リファクタリングまたは再設計のコストの見積もりは複雑になる場合があります。方向性のあるビジネスケースを準備するために与えられた期間内にこれを試すことは実用的ではありません。前述したように、[移行とモダナイゼーション](#)の最初のフェーズでは、リホスト、再配置、またはリプラットフォーム戦略の使用を検討してください。これらの R 戦略は、初期ペイバックを加速し、実装リスクを軽減し、短期的にビジネスケースを改善する可能性があります。また、アプリケーションチームは、環境内で AWS 実行されているアプリケーションを、そうでないアプリケーションよりもモダナイズしやすくなります。[詳細なビジネスケース](#)の準備が整うと、リファクタリング (再設計) 固有のアプリケーションの見積もりが最適に追加されます。

戦略による移行の労力の見積もり

移行はそれぞれ異なります。予算や計画をコミットする前に、社内アプリケーションチーム、AWS プロフェッショナルサービス、パートナー AWS 組織など、プロジェクトを担当するチームからの移行アクティビティのワークロード見積もりを作成します。

ディレクショナルケースの構築に役立つように、次の表にさまざまな処理の労力範囲を示します。これらの範囲は、medium-to-largeのポートフォリオが移行中であり、移行チームがトレーニングを受け、経験していることを前提としています。小規模なポートフォリオの場合、方向性のあるケースであっても、移行を担当するチームに見積りを準備することをお勧めします。

移行戦略	見積りプロセス	[Elements] (要素)	人時	人時
保持する	何もせず、コストもメリットもテクノロジーの負債も削減しません。	—	—	—
リタイア	使用されているハードウェア機器がある場合は、廃止を見積もります。	—	—	—
リロケート	VMware ツールを使用して、VMware 内のワークロードのコピーを推定します。これには、データのコピー、検証のための煙テスト、ハードウェアの廃止が含まれます。VMs の再配置の労力は、	—	—	—

通常、複雑さの低いリホストパターンの場合よりも小さくなります。

リホスト	本番サーバーに適した場合は、イメージコピー、煙テスト、高可用性 (HA) およびディザスタリカバリ (DR) テスト、ハードウェアの廃止を使用して、ワークロードとデータのコピーを推定します。ベストプラクティスは、などのツールを使用することです AWS Transform MGN 。データベースまたは他のインフラストラクチャソフトウェアが実行されているかどうか、データベースの複雑さ、クラスター化されているかどうか、統合の複雑さ、データボリュームなどの要因に基づいて、ワークロードを低、中、高	サーバーあたりのアプリケーションあたりの労力	移行	HA/DR テスト
		低	10 ~ 14	3 ~ 5
		中	16 ~ 24	4 ~ 6
		高	26 ~ 38	8 ~ 12

の複雑さに分割
します。

リプラットフォーム	オペレーティングシステムまたは RDBMS バージョンへのアップグレードを含むリプラットフォーム移行の場合は、リホストの見積もりを取得し、新しいプラットフォームで再構築と煙テストを実行する時間を追加します。プラットフォームのテクノロジーの変更がリプラットフォームに含まれている場合は、 AWS Schema Conversion Tool や などの変換ツールを使用するための追加の時間を見積もり AWS Database Migration Service 、より完全なアプリケーションテストを行います。テクノロジーを変更する例として、	サーバーあたりのアプリケーションあたりの労力	バージョンアップ	テクノロジーの変更
		低	1～3 を追加する	10～15 を追加する
		中	2～5 を追加する	20～30 の追加
		高	4～8 を追加する	40～60 を追加する

	独自の商用データベースからオープンソースの置き換えへの移行が挙げられます。			
再購入	データ抽出、変換、新しく購入した SaaS サービスへのアップロード、およびハードウェアの廃止を見積もります。	–	–	–

移行インフラストラクチャのコストの見積もり

移行中に使用するインフラストラクチャの見積もりを含めます。通常、これらの見積もりは以下で構成されます。

- 現在の環境から へのワークロードとデータ移行のための接続とデータ交換サービスの予算 AWS
- 移行、テスト、カットオーバープロセス中に移行されたワークロードをホストするために必要な予算 AWS のサービス (特にコンピューティングとストレージ)
- 各移行ウェーブの完了に伴う AWS ユーティリティコストの増加
- 移行されたワークロードを実行しない既存のインフラストラクチャの廃止コスト

データ交換については、合計データボリュームを調べ、ネットワークの使用の実現可能性を評価します。移行後に運用上の使用のために [AWS Direct Connect](#) または [Site-to-Site VPN](#) から WAN 上のポイント AWS へのリンクを事前にプロビジョニングしている場合は、そのリソースをサービスクォータまで使用できます。

ネットワーク容量が不十分な場合、仮想プライベートネットワーク (VPN) によるインターネット帯域幅の短期的な増加は、多くの場合、費用対効果の高いソリューションです。そうでない場合、[AWS Snowball Edge](#) やなどの AWS メディア交換デバイスは、ほとんどの ソリューション [AWS Snowball Edge](#) を提供します AWS リージョン。また、非常に大量のデータ移行の場合は、 の予算を

含めることを検討してください。これにより [AWS DataSync](#)、使用するメディアに関係なく信頼性が向上し、転送が高速化されます。

ビジネスケースのキャッシュフロー分析要素では、のランプアップ AWS のサービスと既存のインフラストラクチャのランプダウンをモデル化することが重要です。この段階では、いつコストが発生するかを正確に判断するためのウェーブプランがある可能性は低くなります。次の構成を推奨します。

- のコストを移行よりも一定の割合 AWS で引き上げます。
- 同じ期間にわたって一定の割合で廃止する予定の既存のインフラストラクチャのコストを削減します。

既存のインフラストラクチャが縮小する 1~2 か月前に、AWS コストの増加を開始します。これにより、各ウェーブの移行を実行するために 1 か月間の AWS ユーティリティ使用量が提供されます。これには、テストの時間と、置き換えられたインフラストラクチャでコストが発生しないようにするために必要な廃止作業を完了する追加の時間が含まれます。

廃止コストの見積もり

再デプロイできない機器を廃止し、法的かつ環境に優しく廃棄すると、少額のコストが発生する可能性があります。ただし、方向性のあるビジネスケースでは、通常、潜在的に重要な合計は、置き換えられたアセットの残りの書籍価値を償却するコストです。

方向性のあるビジネスケースでは、以下を実行することをお勧めします。

- アセットリストを確認します。
- 廃止されるものを特定します。
- 書き込みオフを減らすには、リスト内の新しいデバイスを使用して、より完全に廃止された古いアセットを置き換えることができるように、デバイスを切り替えられる機会を調べます。
- その時点で廃止されるアセットの将来の書籍価値を評価します。
- これを廃止の移行コストとして含めます。

完全な方向性のビジネスケースの組み立てと調整

シナリオのペアごとにコストの完全なセットを準備したら、それぞれの割引キャッシュフローステートメントを作成し、グラフ化します。ハードウェアの更新サイクルと同じ期間に方向性のあるビジネス

スケースを構築することをお勧めします。サーバー、ストレージ、ネットワークデバイスの場合は、通常 5 年です。ハードウェアの更新サイクルと同じ期間を使用する場合、1 回のみの更新のコストは、各シナリオのそのままのコストに含まれます。

次に、プログラムの次のフェーズに移行するための承認を得るために必要な主要な財務メトリクスを計算します。通常、以下が含まれます。

- 評価されたコスト削減と生産性の向上の絶対値を測定する正味現在値 (NPV)
- リターンが十分に高速であることを確認するための月単位のペイバック期間
- プロセスが期間にわたって十分なコストを奪っているかどうかを確認する最後の実行レート比較
- 組織が優先する可能性のある資本に対する他の需要と比較して、プログラムの相対的な財務パフォーマンスを評価するための投資収益率 (ROI) と変更された投資収益率 (MIRR)

次の例のように、ケースの最初の反復を使用して、予想される財務パフォーマンスが絞り込む必要があるかどうかを判断します。

- 支払いが遅すぎる場合は、次のような移行のコストを高速化して削減するオプションを検討してください。
 - AWS パートナーまたは AWS プロフェッショナルサービスを使用して利用可能なリソースを拡張し、より基本的なパターンでワークロードの移行をさらに並列化します。
 - VMware で実行されているワークロードの場合、少なくとも初期フェーズでは、再配置戦略とリホストまたはリプラットフォーム戦略を比較します。再配置戦略を使用すると、移行コストを削減し、移行速度を向上させることができます。
 - 技術的に可能な場合は、より複雑なリプラットフォームまたはリファクタリング (リアーキテク) 戦略を必要とするワークロードを、最初のビジネスケースの範囲外の将来のフェーズにプッシュします。
- ROI と MIRR が低すぎる場合は、次の点を考慮してください。
 - 検討しているシナリオは保守的すぎますか？最も可能性の高い容量の増加と伸縮性のニーズを反映したシナリオはありますか？目標内のサービス品質の向上を含むコストを比較するシナリオはありますか？
 - 第 1 フェーズで移行するアプリケーションポートフォリオの範囲を絞り込んで、現在の使用率が低いワークロードや、ディザスタリカバリ (DR) のニーズが高いワークロードなど、より強力なリターンをもたらすワークロードに集中できますか？
 - アプリケーションポートフォリオの範囲を絞り込んで、最初に商用性が低い特定のワークロードを除外できますか？例えば、パブリッククラウドインフラストラクチャへのデプロイの条件が異

なるため、サードパーティーのソフトウェアライセンスのコストが高くなるワークロードを延期できますか？

- 最終的な実行レート比較が予想されるターゲットを満たさない場合は、以下を調べます。
- まず、他のメトリクスが期待を満たしていることを確認します。方向性のあるビジネスケースは、主に移行準備の次のフェーズを開始することを正当化する十分な財務機会があることを示すためのものです。
- 移行の初期フェーズ AWS 後も のコストパフォーマンスを継続的に改善する機会のリストを特定します。

詳細なビジネスケースを準備する際には、機会のリストの評価を含めます。さらに、ケースの継続的なメンテナンスと、移行完了後のmonth-to-monthのコスト最適化プロセスに機会評価を含めます。

優先順位付けされたアプリケーション評価

前のステージの主な成果の 1 つは、[ポートフォリオ検出と初期計画](#)であり、詳細な評価のために[アプリケーションのサブセットを優先](#)することでした。このセクションでは、アプリケーションの詳細な評価について説明します。

いくつかのアプリケーションの詳細を早期に見ると、加速が促進されます。評価と今後のアーキテクチャ設計のプロセスは、潜在的なブロック要因を明らかにし、より大きな範囲の移行に取って代わる重要なタスクを明確にします。これらのタスクには、ランディングゾーンなどの AWS 基盤を確立したり AWS、既存のランディングゾーンを拡張して検証したりするための要件の収集が含まれます。この評価は、移行のステップと戦略を検討するときでもあります。

このステージの主な成果は次のとおりです。

- 優先順位付けされたアプリケーションの検証済みリスト
- 文書化された現在の状態アーキテクチャ
- 移行候補の文書化されたターゲットアーキテクチャと移行戦略
- 特定された移行パターンとツール
- 文書化されたプラットフォーム要件 (セキュリティ、AWS インフラストラクチャ、オペレーション)
- 移行計画に関する文書化されたカットオーバーに関する考慮事項
- 推定 AWS 実行レート

詳細な評価データ要件について

次の表は、移行内のアプリケーションとその関連インフラストラクチャの完全なポートフォリオビューを取得するために必要な情報を示しています。

テーブルでは、次の略語を使用します。

- R、必須
- O、オプション
- 該当なし、該当なし

アプリケーション

属性名	説明	検出、設計、移行戦略	推定実行レート	推奨される忠実度レベル (最小)
一意の識別子	たとえば、アプリケーション ID などです。通常、既存の CMDBs またはその他の内部インベントリと管理システムで使用できます。一意の IDs組織で定義されていない場合は必ず作成することを検討してください。	R	O	高
アプリケーション名	このアプリケーションが組織で認識される名前。必要に応じて、市販の (COTS) off-the-shelf ベンダーと製品名を含めます。	R	R	高
COTS ですか？	はいまたはいいえ。これが商用アプリケーションか内部開発か	R	R	高

COTS 製品とバージョン	商用ソフトウェア製品名とバージョン	R	R	高
説明	プライマリアプリケーション関数とコンテキスト	R	O	高
緊急性	例えば、戦略的アプリケーションや収益を生み出すアプリケーション、重要な機能のサポートなどです。	R	O	高
タイプ	データベース、顧客関係管理 (CRM)、ウェブアプリケーション、マルチメディア、IT 共有サービスなど	R	O	高
環境	例: 本番稼働前、本番稼働前、開発、テスト、サンドボックス	R	R	高

コンプライアンスと規制	ワークロードに適用されるフレームワーク (HIPAA、Sox、PCI-DSS、ISO、SOC、FedRAMP など) と規制要件	R	O	高
依存関係	内部および外部のアプリケーションまたはサービスへのアップストリームとダウンストリームの依存関係	R	該当なし	高
インフラストラクチャマッピング	アプリケーションを構成する物理アセットや仮想アセットへのマッピング	R	R	高
ライセンス	コモディティソフトウェアライセンスタイプ (Microsoft SQL Server Enterprise など)	R	R	高
Cost	ソフトウェアライセンス、ソフトウェアオペレーション、メンテナンスのコスト	O	R	やや高い

ビジネスユニット	マーケティング、財務、営業など	R	O	高
所有者の詳細	アプリケーション所有者の連絡先情報	R	O	高
アーキテクチャタイプ	ウェブアプリケーション、2 層、3 層、マイクロサービス、サービス指向アーキテクチャ (SOA) など	R	R	高
目標復旧時点 (RPO)、目標復旧時間 (RTO)、/ サービスレベルアグリーメント (SLA)	現在のサービス - 管理属性	R	R	高
収益を生み出すアプリケーションがビジネス戦略アプリケーションか。	はい。アプリケーションが会社の収益に直接または間接的に影響を与えるか、ビジネスによって戦略的と見なされる場合。 。	R	O	やや高い

ユーザー数 (同時)	内部ユーザー、外部ユーザー、または内部ユーザーや外部ユーザー/顧客など	R	R	やや高い
ユーザーの場所	ユーザーセッションのオリジン	R	R	やや高い
リスクと問題	既知のリスクと問題	R	O	やや高い
移行に関する考慮事項	移行に関連する可能性のある追加情報	R	R	やや高い
移行戦略	たとえば、移行用の 6 R の 1 AWS つ	R	R	やや高い
データベースの詳細	例えば、パーティショニング、暗号化、レプリケーション、拡張機能、Secure Sockets Layer (SSL) サポートなど	R	R	高
サポートチーム	たとえば、アプリケーションオペレーションチーム名	R	O	やや高い

モニタリングソリューション	このアプリケーションをモニタリングするために使用される製品	R	O	やや高い
バックアップ要件	AWS で必要なバックアップスケジュール	R	R	やや高い
DR 情報	例えば、このアプリケーションのディザスタリカバリコンポーネント	R	R	やや高い
ターゲット AWS 要件	コンポーネント、アカウント、プレースメント、ネットワーク、セキュリティなど	R	R	高

インフラストラクチャ

属性名	説明	検出、設計、移行戦略	推定実行レート	推奨される忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID などで。通常、既存の CMDBs またはその他の内部インベントリと管理システムで使用できます。	R	O	高

	一意の IDs組織で定義されていない場合は必ず作成することを検討してください。				
ネットワーク名	ネットワーク内のアセット名 (ホスト名など)	R	O		高
DNS 名 (完全修飾ドメイン名、または FQDN)	[DNS 名]	R	O		やや高い
IP アドレスとネットマスク	内部 IP アドレスおよび/またはパブリック IP アドレス	R	R		高
アセットタイプ	物理サーバーまたは仮想サーバー、ハイパーバイザー、コンテナ、デバイス、データベースインスタンスなど	R	R		高
製品名	商用ベンダーと製品名 (VMware ESXi、IBM Power Systems、Exadata など)	R	R		高

オペレーティングシステム	例: RHEL 8、Windows Server 2019、AIX 6.1	R	R	高
設定	割り当てられた CPU、コア数、コアあたりのスレッド数、合計メモリ、ストレージ、ネットワークカード	R	R	高
使用率	CPU、メモリ、ストレージのピークと平均。データベースインスタンスのスループット。	R	R	高
ライセンス	商品ライセンスタイプ (RHEL Standard など)	R	R	高
アプリケーションマッピング	このインフラストラクチャで実行されるアプリケーションまたはアプリケーションコンポーネント	R	O	高
通信データ	例えば、プロセスレベルでのサーバー間	R	該当なし	やや高い

ターゲット AWS 要件	インスタンスタイプ、アカウント、サブネット、セキュリティグループ、ルーティングなど	R	R	高
移行戦略、パターン、ツール	例えば、移行用の 6 R の 1 つ、特定の技術パターン、移行ツール	R	O	高
リスクと問題	既知のリスクと問題	R	O	やや高い

詳細なアプリケーション評価

詳細なアプリケーション評価の目的は、ターゲットアプリケーションとその関連インフラストラクチャ (コンピューティング、ストレージ、ネットワーク) を完全に理解することです。落とし穴を回避するには、忠実度の高いデータが必要です。たとえば、組織がアプリケーションを完全に理解していることを前提とするのが一般的です。これは自然であり、多くの場合当てはまります。ただし、ビジネスへのリスクを最小限に抑えるには、プログラムによるデータを可能な限り取得して、組織の知識と静的ドキュメントを検証することが重要です。これにより、検出プロセスの負担が軽減されます。ビジネス固有の情報、戦略的ロードマップなど、代替ソースから得られるデータ要素に集中できます。

重要なのは、移行中および移行後の直前の変更を避けることです。例えば、移行時には、継続的な移行ウェーブにサーバーを含める必要がある可能性のある未確認の依存関係に基づいて、変更を最小限に抑えることが重要です。移行後すぐに、関連するプラットフォーム要件に基づいて変更を最小限に抑え、トラフィックを許可したり、追加のサービスをデプロイしたりすることが重要です。このような予期しない変更により、セキュリティや運用上の問題のリスクが高まります。プログラムによる検出ツールを使用して、詳細なアプリケーション評価を実行するときにトラフィックパターンと依存関係を検証することを強くお勧めします。

評価の最初に、アプリケーションのステークホルダーを特定する必要があります。これらは通常、次のようになります。

- ビジネスユニットリード
- アプリ所有者
- アーキテクト
- オペレーションとサポート
- クラウド対応チーム
- コンピューティング、ストレージ、ネットワークなどの特定のプラットフォームチーム

詳細な検出には 2 つのアプローチがあります。トップダウン検出は、アプリケーション、またはユーザーから始まり、インフラストラクチャまで続きます。これは、アプリケーションの識別が明確である場合に推奨されるアプローチです。逆に、ボトムアップ検出はインフラストラクチャから始まり、アプリケーションまたはサービスとそのユーザーまで続きます。このアプローチは、移行プログラムがインフラストラクチャチームによって推進されている場合や、application-to-infrastructure間のマッピングが不明な場合に役立ちます。一般的に、両方の組み合わせを使用する可能性があります。

アプリケーションを深く掘り下げるには、既存のアーキテクチャ図から始めることをお勧めします。これらが利用できない場合は、現在の知識に基づいて作成します。単純なリホスト移行戦略であっても、このタスクの重要性を過小評価しないでください。アーキテクチャ図をプロットすると、クラウド上の小さな変更で迅速に対処できる非効率性を特定できます。

トップダウンアプローチとボトムアップアプローチのどちらを実行しているかに応じて、最初の図にアプリケーションコンポーネントとサービス、またはサーバーやロードバランサーなどのインフラストラクチャコンポーネントがプロットされます。主要なコンポーネントとインターフェイスを特定したら、検出ツールとアプリケーションパフォーマンスモニタリングツールのプログラムデータを使用して検証します。ツールは依存関係分析をサポートし、コンポーネント間の通信情報を提供する必要があります。このアプリケーションを構成する各コンポーネントを特定する必要があります。次に、内部と外部の両方の他のアプリケーションとサービスへの依存関係を文書化します。

依存関係とマッピングを検証するためのツールがない場合は、手動アプローチが必要です。たとえば、インフラストラクチャコンポーネントにログインし、スクリプトを実行して、オープンポートや確立された接続などの通信情報を収集できます。同様に、実行中のプロセスとインストールされているソフトウェアを特定できます。手動検出に必要な労力を過小評価しないでください。[Kiro](#) などのエージェント AI ツールは、スクリプトの構築やデータ出力の分析に役立ちます。プログラムツールを使用すると、ほとんどの依存関係を数日でキャプチャしてレポートできます。ただし、より長い間

隔 (通常は少数の割合) で発生する依存関係は除きます。手動検出では、すべてのデータポイントの収集とマージに数週間かかる場合がありますが、それでもエラーや欠落データが発生しやすくなります。

次に、優先順位付けされた各アプリケーションとマッピングされたインフラストラクチャの[データ要件](#)セクションで指定された情報を取得します。次に、次のアンケートを使用して、詳細な評価プロセスについて説明します。特定されたステークホルダーとミーティングを行い、これらの質問に対する回答を取得して話し合います。

General

- このアプリケーションの重要度レベルはどのくらいですか？ 収益は発生しますか？ ビジネス戦略アプリケーションか、サポートビジネスアプリケーションか。他のシステムによって共有されるコアインフラストラクチャサービスですか？
- このアプリケーションに進行中の変換プロジェクトはありますか？
- これは内部向けか外部向けか。

アーキテクチャ

- 現在のアーキテクチャタイプ (SOA、マイクロサービス、モノリスなど) は何ですか？ アーキテクチャにはいくつの階層がありますか？ しっかりと結合されているか、緩く結合されているか。
- コンポーネント (コンピューティング、データベース、リモートストレージ、ロードバランサー、キャッシュサービスなど) は何ですか？
- テクノロジスタックとは オペレーティングシステム、ランタイム、パッケージ、フレームワーク、ミドルウェアについて説明します。
- APIs API 名、オペレーション、URLs、ポート、プロトコルなど、これらについて説明します。
- データフローとは バッチジョブ、ETL、パイプライン、ファイル転送について説明します。
- コンポーネント間、およびこのアプリケーションやサービスと他のアプリケーションやサービスとの間で許容される最大レイテンシーはどのくらいですか？

オペレーション

- このアプリケーションはどの場所で動作しますか？
- アプリケーションとインフラストラクチャを運用するユーザー これらは内部チームまたは AWS パートナーチームによって運営されていますか？

- このアプリケーションがダウンするとどうなりますか？ 影響を受けるのは誰ですか？ その影響は何ですか？
- ユーザーまたは顧客はどこにいますか？ アプリケーションへのアクセス方法 同時ユーザー数はいくつですか？
- テクノロジーを最後に更新したのはいつですか？ 更新は今後予定されていますか？ その場合、いつですか？
- このアプリケーションの既知のリスクと問題は何ですか？ 停止、中程度の重大度、重要度の高いインシデントの履歴は何ですか？
- 使用サイクル (営業時間) を教えてください。運用タイムゾーンは何ですか？
- 変更凍結期間とは何ですか？
- このアプリケーションをモニタリングするには、どのようなソリューションを使用しますか？

パフォーマンス

- 収集されたパフォーマンス情報には何が表示されますか？ 使用量は急増しているか、一定で予測可能か。
- 使用可能なパフォーマンスデータの時間枠、間隔、日付を教えてください。

ソフトウェアのライフサイクル

- 現在の変化率はどのくらいですか (毎週、毎月、四半期ごと、または毎年)？
- 開発ライフサイクル (テスト、開発、QA、UAT、本番前、本番など) は何ですか？
- アプリケーションとインフラストラクチャのデプロイ方法は何ですか？
- デプロイツールとは
- このアプリケーションまたはインフラストラクチャは、継続的インテグレーション (CI)/継続的デリバリー (CD) を使用していますか？ 自動化のレベルはどのくらいですか？ 手動タスクとは
- アプリケーションとインフラストラクチャのライセンス要件は何ですか？
- サービスレベルアグリーメント (SLA) とは
- 現在のテストメカニズムは何ですか？ テストステージとは

移行

- 移行に関する考慮事項は何ですか？

- 技術的な複雑さは何ですか？
- ビジネスリスクは何ですか？
- 収集されたデータの信頼度はどのくらいですか？

この時点で、このアプリケーションを移行する際の考慮事項に注意してください。より完全で正確な評価を行うには、さまざまな利害関係者からこの質問に対する回答を取得します。次に、知識と意見を対比します。

回復性

- 現在のバックアップ方法は何ですか？バックアップに使用される製品 バックアップスケジュールとは何ですか？バックアップ保持ポリシーとは
- 現在の目標復旧時点 (RPO) と目標復旧時間 (RTO) は何ですか？
- このアプリケーションにはディザスタリカバリ (DR) プランがありますか？その場合、DR シミュレーションとは何ですか？
- 最後の DR テストはいつ行われましたか？

セキュリティとコンプライアンス

- このアプリケーションに適用されるコンプライアンスと規制のフレームワークは何ですか？最終監査日と次回の監査日はいつですか？
- このアプリケーションは機密データをホストしていますか？データ分類とは？
- データは転送中、保管中、またはその両方で暗号化されていますか？暗号化メカニズムとは
- このアプリケーションは SSL 証明書を使用していますか？発行機関とは
- ユーザー、コンポーネント、その他のアプリケーションやサービスの認証方法は何ですか？

データベース

- このアプリケーションはどのデータベースを使用しますか？
- データベースへの同時接続の一般的な数を教えてください。接続の最小数と最大数を教えてください。
- 接続方法 (JDBC、ODBC など) は何ですか？
- 接続文字列は文書化されていますか？その場合、どこにありますか？

- データベーススキーマとは
- データベースはカスタムデータ型を使用していますか？

依存関係

- コンポーネント間の依存関係は何ですか？ 解決できず、コンポーネントと一緒に移行する必要がある依存関係に注意してください。
- コンポーネントは複数の場所に分かれていますか？ これらの場所 (WAN、VPN など) 間の接続は何ですか？
- このアプリケーションの他のアプリケーションやサービスへの依存関係は何ですか？
- 運用上の依存関係は何ですか？ 例えば、パッチ適用ウィンドウなどのメンテナンスとリリースのサイクルなどです。

AWS アプリケーション設計と移行戦略

アプリケーションの将来の状態を設計および文書化することは、移行の重要な成功要因です。シンプルでも複雑でも、あらゆる種類の移行戦略の設計を作成することをお勧めします。設計を作成すると、アーキテクチャが変更されることが予想されない場合でも、潜在的なブロック要因、依存関係、アプリケーションを最適化する機会が明らかになります。さらに、結果として得られる設計は、移行後のさらなる進化の基盤となります。

設計プロセスの重要性に加えて、よりブロックされ、移行速度に影響を与える可能性もあります。これは、オンプレミスシステム用に考案された従来のアーキテクチャレビュープロセスが原因で発生します。アーキテクチャのレビューと承認プロセスを確認し、事前承認できるアーキテクチャパターンやタイプの確立に集中することをお勧めします。数百または数千のアプリケーションのアーキテクチャを確認する労力を過小評価せず、プロセスをモダナイズする機会を得てください。

次のリストには、設計プロセスに役立つリソースが含まれています。

- [AWS アーキテクチャセンター](#)は、AWS Well-Architected フレームワークなどのツールとガイダンスを組み合わせています。また、アプリケーションに使用できるリファレンスアーキテクチャも提供します。
- [Amazon Builders' Library](#) には、Amazon がソフトウェアを構築および運用する方法に関するいくつかのリソースが含まれています。

- [AWS ソリューションライブ러리](#)は、AWS が検証したクラウドベースのソリューションを多数提供しています。これには、リファレンスアーキテクチャの大規模なコレクションが含まれています。
- [AWS 規範ガイダンス](#)は、設計プロセスと移行のベストプラクティスに役立つ戦略、ガイド、パターンを提供します。
- [AWS Documentation](#) には、ユーザーガイドや API リファレンスなどの AWS サービスに関する情報が含まれています。
- [入門リソースセンター](#)には、基礎を学習するための実践的なチュートリアルと詳細な演習がいくつか用意されています AWS。

クラウドジャーニーのどの段階にあるかに応じて、AWS 基盤が既に存在している可能性があります。これらの AWS 基盤には以下が含まれます。

- AWS リージョン が特定されました。
- アカウントが作成されているか、オンデマンドで取得できます。
- 一般的なネットワークが実装されました。
- 基本 AWS サービスは アカウント内にデプロイされています。

逆に、プロセスの早い段階で基盤がまだ確立 AWS されていない可能性があります。基盤が確立されていないと、アプリケーション設計の範囲が制限されたり、定義のためのさらなる作業が必要になる可能性があります。この場合、ランディングゾーンの基本的な設計をアプリケーション設計作業と並行して定義して実装することをお勧めします。アプリケーション設計は、AWS アカウント 構造、ネットワーキング、仮想プライベートクラウド (VPCs)、クラスレスドメイン間ルーティング (CIDR) 範囲、共有サービス、セキュリティ、クラウドオペレーションなどの要件を特定するのに役立ちます。

[AWS Control Tower](#) は、ランディングゾーンと呼ばれる安全なマルチアカウント AWS 環境を設定および管理するための最も簡単な方法を提供します。は、を使用してランディングゾーン AWS Control Tower を作成します。これにより AWS Organizations、何千人ものお客様とクラウドに移行する際に、AWS ベストプラクティスベースのエクスペリエンスを継続的に管理、ガバナンス、実装できます。

アプリケーションの将来の状態

まず、このアプリケーションの初期移行戦略を確立します。この時点で、戦略は将来の状態設計の一部として変更され、潜在的な制限を明らかにする可能性があるため、初期と見なされます。初期の前

提条件を検証するには、[6 Rs 決定ツリー](#)を参照してください。また、潜在的な移行フェーズを文書化します。たとえば、このアプリケーションは 1 つのイベントに移行されますか (すべてのコンポーネントは同時に移行されますか)。または、これは段階的な移行ですか (一部のコンポーネントは後で移行されます)?

特定のアプリケーションの移行戦略は一意ではない場合があることに注意してください。これは、複数の R タイプを使用してアプリケーションコンポーネントを移行できるためです。例えば、最初のアプローチは、変更なしでアプリケーションをリフトアンドシフトすることです。ただし、アプリケーションのコンポーネントは、さまざまな処理を必要とするさまざまなインフラストラクチャアセットに存在する場合があります。たとえば、アプリケーションは 3 つのコンポーネントで構成され、それぞれが別のサーバーで実行され、いずれかのサーバーはクラウドでサポートされていないレガシーオペレーティングシステムを実行します。そのコンポーネントにはリプラットフォームアプローチが必要ですが、サポートされているサーバーバージョンで実行されている他の 2 つのコンポーネントはリホストできます。移行する各アプリケーションコンポーネントおよび関連するインフラストラクチャに移行戦略を割り当てることが重要です。

次に、コンテキストと問題を文書化し、現在の状態を定義する既存のアーティファクトをリンクします。

- このアプリケーションが移行される理由
- 提案された変更は何ですか?
- 利点は何ですか?
- 重大なリスクやブロック要因はありますか?
- 現在の欠点は何ですか?
- スコープ内とスコープ外のものは何ですか?

再現性

設計作業全体を通して、このアプリケーションのこのソリューションとアーキテクチャを他のアプリケーションに再利用する方法を検討してください。このソリューションは一般化できますか?

要件

セキュリティを含め、このアプリケーションの機能要件と非機能要件を文書化します。これには、選択した移行戦略に応じて、現在と将来の状態の要件が含まれます。詳細なアプリケーション評価中に収集された情報を使用して、このプロセスをガイドします。

To-Be アーキテクチャ

このアプリケーションの将来のアーキテクチャについて説明します。ソース環境 (オンプレミス) とターゲット AWS 環境 (ターゲット、アカウント AWS リージョン、VPCs、アベイラビリティーゾーンなど) の構成要素を含む再利用可能な図テンプレートを作成することを検討してください。

移行するコンポーネントと新しいコンポーネントのテーブルを作成します。このアプリケーションとやり取りする他のアプリケーションとサービス (オンプレミスまたはクラウド) を含めます。

次の表に、コンポーネントの例を示します。リファレンスアーキテクチャや入念な設定を表すものではありません。

名前	説明	詳細
アプリケーション	外部サービス (インバウンド接続)	サービスは公開された API からのデータを消費します。
DNS	名前解決 (内部)	ベースラインアカウント設定の一部としてデプロイされた Amazon Route 53
Application Load Balancer	バックエンドサービス間でトラフィックを分散します	オンプレミスロードバランサーを置き換えます。プール A を移行します。
アプリケーションのセキュリティ	DdoS 保護	を使用して実装 AWS Shield
セキュリティグループ	仮想ファイアウォール	ポート 443 (インバウンド) のアプリケーションインスタンスへのアクセスを制限します。
サーバー A	フロントエンド	Amazon Elastic Compute Cloud (Amazon EC2) を使用してリホストします。
サーバー B	フロントエンド	Amazon EC2 を使用してリホストします。

サーバー C	アプリケーションロジック	Amazon EC2 を使用してリホストします。
サーバー D	アプリケーションロジック	Amazon EC2 を使用してリホストします。
Amazon Relational Database Service (Amazon RDS) – Amazon Aurora	データベース	サーバー E と F を置き換えます
モニタリングとアラート	変更管理	Amazon CloudWatch
監査ログ	変更管理	AWS CloudTrail
パッチ適用とリモートアクセス	メンテナンス	AWS Systems Manager
リソースアクセス	安全なアクセスコントロール	AWS Identity and Access Management (IAM)
認証	ユーザーアクセス	Amazon Cognito
証明書	SSL/TLS	AWS Certificate Manager
API 1	外部 API	Amazon API Gateway
オブジェクトストレージ	イメージホスティング	Amazon Simple Storage Service (Amazon S3)
認証情報	認証情報の管理とホスティング	AWS Secrets Manager
AWS Lambda 関数	データベース認証情報と API キーの取得	AWS Lambda
インターネットゲートウェイ	アウトバウンドインターネットアクセス	VPC へのインターネットゲートウェイ
プライベートサブネット 1	バックエンドと DB	アベイラビリティゾーン 1 – VPC 1

プライベートサブネット 2	バックエンドと DB	アベイラビリティゾーン 2 – VPC 1
パブリックサブネット 1	フロントエンド	アベイラビリティゾーン 1 – VPC 1
パブリックサブネット 2	フロントエンド	アベイラビリティゾーン 2 – VPC 1
バックアップサービス	データベースと EC2 インスタンスのバックアップ	AWS Backup
DR	Amazon EC2 の耐障害性	AWS Elastic Disaster Recovery

コンポーネントを特定したら、任意のツールを使用してそれらを図にプロットします。初期設計を、アプリケーション所有者、エンタープライズアーキテクト、プラットフォームチーム、移行チームなど、主要なアプリケーションステークホルダーと共有します。次の質問をすることを検討してください。

- チームは通常、設計に同意していますか？
- 運用チームはサポートできますか？
- 設計を進化させることはできますか？
- 他にオプションはありますか？
- 設計はアーキテクチャ標準とセキュリティポリシーに準拠していますか？
- 不足しているコンポーネント (コードリポジトリ、CI/CD ツール、VPC エンドポイントなど) はありますか？

アーキテクチャ上の意思決定

設計プロセスの一環として、アーキテクチャ全体またはその特定の部分に対してより多くのオプションが見つかる可能性があります。これらのオプションを、優先または選択したオプションの理論的根拠とともに文書化します。これらの決定は、アーキテクチャ上の決定として文書化できます。

メインオプションがリストされ、新しいリーダーが別のオプションを使用する決定の背後にあるオプションと理由を理解するのに十分な詳細で説明されていることを確認します。

ソフトウェアライフサイクル環境

現在の環境への変更を文書化します。たとえば、テスト環境と開発環境は で再作成され AWS 、移行されません。

Tagging

各インフラストラクチャコンポーネントの必須および推奨されるタグ付けと、この設計のタグ付け値について説明します。

移行戦略

設計のこの時点で、移行戦略に関する最初の前提を検証する必要があります。選択した R 戦略にコンセンサスがあることを確認します。全体的なアプリケーション移行戦略と個々のアプリケーションコンポーネントの戦略を文書化します。前述のように、異なるアプリケーションコンポーネントでは、移行に異なる R タイプが必要になる場合があります。

さらに、移行戦略を主要なビジネス推進要因と成果に合わせます。また、さまざまな移行イベントでのコンポーネントの移動など、移行に対する段階的なアプローチについても説明します。

6 R の決定の詳細については、[AWS Migration Hub 「戦略の推奨事項」](#)を参照してください。

移行パターンとツール

アプリケーションおよびインフラストラクチャコンポーネントの移行戦略を定義することで、特定の技術パターンを探索できるようになりました。たとえば、リホスト戦略は、などの移行ツールで実装できます[AWS Transform MGN](#)。状態やデータをレプリケートする必要がない場合は、Amazon マシンイメージ (AMI) とアプリケーションデプロイパイプラインを使用してアプリケーションを再デプロイすることで、同じ結果を得ることができます。

同様に、アプリケーションをリプラットフォームまたはリファクタリング (リアーキテクト) するには、[AWS App2Container](#)、[AWS Database Migration Service \(AWS DMS\)](#)、[AWS Schema Conversion Tool\(AWS SCT\)](#)、などのツールを使用できます[AWS DataSync](#)。コンテナ化には、[Amazon Elastic Container Service \(Amazon ECS\)](#)、[Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)、または [AWS Fargate](#)を使用できます。再購入するときは、特定の製品またはの Software as a Service (SaaS) ソリューションに AMI を使用できます[AWS Marketplace](#)。

目標を達成するために利用できるさまざまなパターンとオプションを評価します。長所と短所、および移行の運用準備を検討してください。分析に役立てるには、次の質問を使用します。

- 移行チームはこれらのパターンをサポートできますか？

- コストと利点のバランスはどれくらいですか？
- このアプリケーション、サービス、またはコンポーネントをマネージドサービスに移動できますか？
- このパターンを実装する労力は何ですか？
- 特定のパターンの使用を妨げる規制やコンプライアンスポリシーはありますか？
- このパターンは再利用できますか？ 再利用可能なパターンが推奨されます。ただし、パターンが 1 回だけ使用される場合があります。代替の再利用可能なパターンよりも、シングルユースパターンの労力のバランスを検討してください。

[AWS 規範ガイダンス](#)には、さまざまな移行パターンと手法が含まれています。

サービス管理とオペレーション

への移行のためのアプリケーション設計を作成するときは AWS、運用準備を検討してください。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準備状況要件を評価するときは、次の質問を考慮してください。

- 運用する準備はできていますか？
- インシデント対応手順は定義されていますか？
- 予想されるサービスレベルアグリーメント (SLA) とは何ですか？
- 職務の分離は必要ですか？
- さまざまなチームがサポートアクションを調整する準備ができていますか？
- 誰が何を担当しますか？

カットオーバーに関する考慮事項

移行戦略とパターンを考慮すると、アプリケーションが移行された時点で知っておくべき重要なことは何ですか？ カットオーバープランニングは、設計後のアクティビティです。ただし、予想されるアクティビティと要件に関する考慮事項があれば文書化します。例えば、該当する場合は概念実証を実行する要件を文書化し、テスト、監査、または検証の要件を概説します。

リスク、前提、問題、依存関係

未解決の未解決のリスク、前提、潜在的な問題を文書化します。これらの項目に明確な所有権を割り当て、全体的な設計と戦略の実装を承認できるように進捗状況を追跡します。さらに、この設計を実装するための主要な依存関係を文書化します。

実行コストの見積もり

ターゲット AWS アーキテクチャのコストを見積もるには、[AWS 料金見積りツール](#)を使用します。[AWS 料金見積りツール](#)。設計で定義されているインフラストラクチャコンポーネントを追加し、推定実行コストを取得します。アプリケーションコンポーネントに必要なソフトウェアライセンスと、AWS のサービス 使用する にまだ含まれていないソフトウェアライセンスを考慮します。

ポートフォリオ分析と移行計画

この評価ステージでは、ポートフォリオ検出と[初期計画セクションで開始されたポートフォリオレベルの検出](#)と分析を完了することに重点を置いています。目標は、アプリケーションとインフラストラクチャのポートフォリオの強固なベースラインを反復して確立することです。このベースラインには、すべての依存関係の特定、移行の合理化モデルの反復、詳細なビジネスケースの作成、移行ウェーブプランの概要が含まれます。その結果、必要なデータ忠実度は高くなります。この段階では、時間投資が必要になります。評価結果を高速化するには、検出ツールなどのプログラムによるデータソースをできるだけ多く使用することをお勧めします。

このステージの主な成果は次のとおりです。

- 忠実度の高いアプリケーションとインフラストラクチャのインベントリ
- アプリケーションごとに定義された移行戦略
- 信頼性の高い移行ウェーブプラン
- 詳細なビジネスケース

完全な評価データ要件について

次の表は、移行内のアプリケーションとその関連インフラストラクチャの完全なポートフォリオビューを取得するために必要な情報を示しています。

テーブルでは、次の略語を使用します。

- R、必須
- O、オプション
- 該当なし、該当なし

アプリケーション

属性名	説明	インベントリと 優先順位付け	詳細なビジネス ケース	推奨される忠実 度レベル (最小)
一意の識別子	たとえば、アプリケーション	R	R	高

ID などです。

通常、既存の CMDBs またはその他の内部インベントリと管理システムで使用できます。一意の IDs 組織で定義されていない場合は必ず作成することを検討してください。

アプリケーション名	このアプリケーションが組織で認識される名前。必要に応じて、市販の (COTS) off-the-shelf ベンダーと製品名を含めます。	R	R	高
COTS ですか？	はいまたはいいえ。これが商用アプリケーションか内部開発か	R	R	高
COTS 製品とバージョン	商用ソフトウェア製品名とバージョン	R	R	高
説明	プライマリアプリケーション関数とコンテキスト	R	R	高

緊急性	例えば、戦略的アプリケーションや収益を生み出すアプリケーション、重要な機能のサポートなどです。	R	R	高
タイプ	データベース、顧客関係管理 (CRM)、ウェブアプリケーション、マルチメディア、IT 共有サービスなど	R	R	高
環境	例: 本番稼働前、本番稼働前、開発、テスト、サンドボックス	R	R	高
コンプライアンスと規制	ワークロードに適用されるフレームワーク (HIPAA、Sox、PCI-DSS、ISO、SOC、FedRAMP など) と規制要件	R	R	高

依存関係	内部および外部のアプリケーションまたはサービスへのアップストリームとダウンストリームの依存関係。運用要素 (メンテナンスサイクルなど) などの非技術的な依存関係。	R	O	高
インフラストラクチャマッピング	アプリケーションを構成する物理アセットや仮想アセットへのマッピング	R	R	高
ライセンス	コモディティソフトウェアライセンスタイプ (Microsoft SQL Server Enterprise など)	R	R	やや高い
Cost	ソフトウェアライセンス、ソフトウェアオペレーション、メンテナンスのコスト	O	R	やや高い
ビジネスユニット	マーケティング、財務、営業など	R	R	高

所有者の詳細	アプリケーション所有者の連絡先情報	R	R	高
DR 情報	ディザスタリカバリコンポーネント	R	R	高
移行戦略	例えば、に移行するための 6 R のいずれか AWS	R	R	高
サポートチケット	停止、減速、トランザクションスロットリング、バッチウィンドウのオーバーランによる生産性と財務への影響を評価するのに役立つ 12 ~24 か月分のデータ	O	R	中

インフラストラクチャ

属性名	説明	インベントリと優先順位付け	ビジネスケース	推奨される忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID などです。通常、既存の CMDBs またはその他の内部インベントリと管理システムで	R	R	高

使用できます。
一意の IDs組織
で定義されてい
ない場合は必ず
作成することを
検討してくださ
い。

ネットワーク名	ネットワーク内 のアセット名 (ホ スト名など)	R	R	高
DNS 名 (完全修 飾ドメイン名、 または FQDN)	[DNS 名]	R	O	高
IP アドレスと ネットマスク	内部 IP アドレス および/またはパ ブリック IP アド レス	R	R	高
アセットタイプ	物理サーバーま たは仮想サー バー、ハイパー バイザー、コン テナ、デバイ ス、データベ ースインスタンス など	R	R	高
製品名	商用ベンダーと 製品名 (VMware ESXi、IBM Power Systems、E xadata など)	R	R	高

オペレーティングシステム	例: RHEL 8、Windows Server 2019、AIX 6.1	R	R	高
設定	割り当てられた CPU、コア数、コアあたりのスレッド数、合計メモリ、ストレージ、ネットワークカード	R	R	高
使用率	CPU、メモリ、ストレージのピークと平均。データベースインスタンスのスループット。	R	R	高
ライセンス	商品ライセンスタイプ (RHEL Standard など)	R	R	高
アプリケーションマッピング	このインフラストラクチャで実行されるアプリケーションまたはアプリケーションコンポーネント	R	R	高

Cost	ハードウェア、メンテナンス、オペレーション、ストレージ (SAN、NAS、オブジェクト)、オペレーティングシステムライセンス、ラックスペースの共有、データセンターのオーバーヘッドなど、ベアメタルサーバーのフルロードコスト	O	R	やや高い
データ転送の推定量 (入出力)	例えば、30 日間の 1 日あたりのインフラストラクチャセットあたり	O	R	中

ネットワーク

属性名	説明	インベントリと優先順位付け	ビジネスケース	推奨される忠実度レベル (最小)
パイプのサイズ (MB/秒)、冗長性 (Y/N)	現在の WAN リンク仕様 (1000 Mb/秒冗長など)	R	R	やや高い
接続されたロケーション	このリンクで接続された名前付きの場所	R	O	高

リンク使用率	ピーク使用率と平均使用率、アウトバウンドデータ転送 (GB/月)	R	R	やや高い
レイテンシー (ミリ秒)	接続された場所間の現在のレイテンシー。	R	O	高
Cost	現在の 1 か月あたりのコスト	該当なし	R	やや高い

移行

属性名	説明	インベントリと優先順位付け	ビジネスケース	推奨される忠実度レベル (最小)
リホストコスト	各ワークロード (人日) の顧客とパートナーの労力、1 日あたりの顧客とパートナーのコスト率、ツールコスト、ワークロード数	該当なし	R	やや高い
リプラットフォームのコスト	各ワークロードの顧客とパートナーの労力 (人日)、1 日あたりの顧客とパートナーのコスト率、ワークロードの数	該当なし	R	やや高い

リファクタリングコスト	各ワークロードの顧客とパートナーの労力 (人日)、1 日あたりの顧客とパートナーのコスト率、ワークロードの数	該当なし	R	やや高い
リタイアコスト	サーバー数、平均廃止コスト	該当なし	R	やや高い
ランディングゾーン	既存の (Y/N) の再利用、AWS リージョン 必要な のリスト、コスト	該当なし	R	やや高い
人材と変化	クラウド運用と開発でトレーニングするスタッフ数、1 人あたりのトレーニングコスト、1 人あたりのトレーニング時間のコスト	該当なし	R	やや高い
時間	対象範囲内のワークロード移行期間 (月)	O	R	やや高い
並列コスト	移行中に現状のコストを削除できる時間枠とレート	該当なし	R	やや高い

移行中に AWS の製品やサービ ス、およびその 他のインフラス トラクチャコス トが導入される 時間枠とレート	該当なし	R	やや高い
--	------	---	------

アプリケーションポートフォリオのベースラインの確立

信頼性の高い移行ウェブプランを作成するには、アプリケーションのポートフォリオとそれに関連するインフラストラクチャのベースラインを確立する必要があります。ポートフォリオベースラインは、技術的な依存関係や移行戦略など、移行範囲の包括的なビューを提供します。ポートフォリオベースラインは、移行の対象となるアプリケーションと、[「評価データ要件の理解」](#)セクションで概説されているデータポイントが収集されることを明確にします。同様に、関連するすべてのインフラストラクチャ (コンピューティング、ストレージネットワーク) が理解され、アプリケーションにマッピングされます。

技術的な依存関係は、次の 4 つのカテゴリで説明できます。

- Application-to-infrastructure の依存関係は、ソフトウェアと物理ハードウェアまたは仮想ハードウェア間のリンクを確立します。たとえば、CRM アプリケーションとインストールされている仮想マシンの間には依存関係があります。
- アプリケーションコンポーネントの依存関係は、さまざまなインフラストラクチャアセットで実行されているコンポーネントがどのように相互作用するかを示します。アプリケーションコンポーネントの依存関係の例としては、仮想マシンで実行されるウェブフロントエンド、別の仮想マシンで実行されるアプリケーションレイヤー、データベースクラスターで実行されるデータベースなどがあります。
- Application-to-application 依存関係は、さまざまなアプリケーションまたはそのコンポーネント間の相互作用に関連しています。application-to-application 依存関係の例は、支払い処理アプリケーションと株式管理アプリケーションです。これらのアプリケーションは独立していますが、定義された API オペレーションまたは共有データセットを使用して常にやり取りします。
- Application-to-infrastructure サービスの依存関係は、インフラストラクチャサービス自体がアプリケーションであるため、技術的には application-to-application 依存関係です。ただし、これらを個別に分類することをお勧めします。主な理由は、インフラストラクチャサービスは通常多くのアプ

リケーションによって共有されるため、依存関係の長い証跡があることです。また、通常、異なる移行戦略とパターンに従います。

たとえば、ロードバランサーには、複数のアプリケーションのバランシングプールを含めることができます。重要なのはプールへの依存関係です。これは、ロードバランサー自体が保持またはリタイアされている間、依存アプリケーションとともに個別に移行される可能性が高くなります。

さらに、application-to-infrastructureへのサービスの依存関係を個別化することで、誤った依存関係グループを回避できます。誤った依存関係グループとは、複数のビジネスアプリケーションがグループ化され、インフラストラクチャサービスへの共通の依存関係を持つことで、同時に移行されることを意味します。例えば、Active Directory などの認証サービスは、大規模なアプリケーショングループに関連付けられている可能性があります。重要なのは、インフラストラクチャサービスの依存関係をフィルタリングし、でこれらのサービスを有効にすることです AWS。

依存関係を使用して移行グループを作成する方法の詳細については、[「Wave Planning」](#) セクションを参照してください。

ポートフォリオのベースラインを確立するときは、各アプリケーションコンポーネントの移行戦略を確認することをお勧めします。移行戦略は、移行用の 6 R のいずれかになります ([「6 R の移行戦略の反復」](#) セクションを参照してください)。ポートフォリオベースラインでは、6 つの R のいずれかを各アプリケーションに関連付ける必要があります。6 R 戦略は、アプリケーションの各インフラストラクチャコンポーネントにも関連付ける必要があります。

依存関係や移行戦略など、ポートフォリオのベースラインバージョンを確立するには、自動検出ツールを使用します ([「検出ツールの必要性の評価」](#) を参照)。アプリケーション所有者やインフラストラクチャチームなどの主要な利害関係者から収集された情報でデータを補完します。このステージのデータ要件セクションで説明されている属性と忠実度レベルに一致する完全なポートフォリオインベントリを取得するまで、データを収集し続けます。結果として得られるデータセットは、移行の推進に役立ちます。

移行の範囲と使用可能なツールによっては、このアクティビティが完了するまでに数週間かかる場合があることに注意してください。

優先順位付け基準の反復

移行ウェーブプランを作成する前に、アプリケーションの優先順位付け基準を繰り返して、パイロットアプリケーションの選択から長期的なウェーブプランニングに移行することをお勧めします。

前のセクションでは、シンプルなクラウド対応アプリケーションを優先するデフォルトの優先順位付け基準を導入しました (「[アプリケーションの優先順位付け](#)」を参照)。これは、初期段階では、移行プロセスを改良し、学んだ教訓を組み込むために、重要ではないアプリケーションから始めることをお勧めします。ただし、この段階では、長期計画を作成するには、アプリケーションを移行する順序をビジネスドライバーに合わせて調整する必要があります。新しい基準を適用すると、ウェーブプランニングの重要な入力となるアプリケーションの新しいランキングが生成されます。

アプリケーションポートフォリオから利用可能なデータポイントを確認し、ビジネスドライバーに基づいてアプリケーションの優先順位付けを決定する属性を選択します。

まず、ビジネスドライバーを検証します (「[ビジネスドライバーと技術ガイドの原則](#)」を参照)。次に、ビジネスドライバーに基づいて、移行するアプリケーションの優先順位付けに役立つ属性を選択します。

次の表は、イノベーションのビジネス推進要因に沿った優先順位付け基準の例を示しています。

属性またはデータポイント	使用できる値	スコア (0 ~ 99)	重要性または関連性の乗算係数
オペレーティングシステム	AIX	80	高 (1x)
	Solaris	80	
	HP-UX	80	
	メインフレーム	70	
	Windows	50	
	Linux	20	
ビジネスの重要性	高	60	高 (1x)
	中	40	
	低	20	
アーキテクチャ	密結合	60	高 (1x)
	疎結合	20	

運用モデル	従来の - CI/CD なし	60	中高 (0.8x)
	基本的な CI/CD	40	
	完全な DevOps	20	
コンピューティング インスタンスの数	1 ~ 3	60	中高 (0.8x)
	4-10	40	
	11 以上	20	
移行戦略	リファクタリング (リ アーキテクト)	70	中 (0.6x)
	リプラットフォーム	40	
	再購入	30	
	リホスト	10	

次の表は、迅速なコスト削減のためのビジネス推進要因に沿った優先順位付け基準の例を示しています。

属性またはデータポイント	使用できる値	スコア (0 ~ 99)	重要性または関連性の乗算係数
データベース製品	Oracle	70	高 (1x)
	Microsoft SQL	70	
	その他	20	
オペレーティングシステム	Windows	70	高 (1x)
	Linux	70	
	その他	20	
CPU 使用率 (平均)	36% 以上	60	高 (1x)

	36% 未満	40	
コンピューティング インスタンスの数	11 以上	60	中高 (0.8x)
	4-10	40	
	1 ~ 3	20	
移行戦略	リタイア	80	中 (0.6x)
	リホスト	70	
	リプラットフォーム	50	
	リファクタリング (リ アーキテクト)	10	

優先順位付け基準をテストし、一般的に出力に同意するまで繰り返します。ベースラインバージョンを取得するには、少なくとも 3 ~ 4 回の反復が必要です。

6 Rs 移行戦略の選択を繰り返す

この段階では、6 Rs 決定ツリーを繰り返して進化させることをお勧めします。[移行用の R タイプの決定](#)セクションでは、デフォルトの決定ツリーが導入されました。ツリーを改訂し、初期パイロットアプリケーションの移行全体の学習を考慮し、ビジネスドライバー、優先順位基準、および独自の状況に引き続き一致させることをお勧めします。サンプルアプリケーションでディシジョンツリーを検証し、それでも期待される戦略が生成されることを確認します。それ以外の場合は、それに応じてロジックを更新します。結果のツリーは、アプリケーションのポートフォリオのベースラインを確立し、各アプリケーションコンポーネントの移行戦略を割り当てる上で重要です。

移行戦略が各アプリケーションコンポーネントおよび関連するインフラストラクチャに割り当てられていることを確認します。この情報は、必要な労力、キャパシティ、スキルを推定するとき、および移行ウェーブプランを作成するときに重要な要素です。

ウェーブプランニング

基本的に、ウェーブプランは移行スケジュールであり、他のプロジェクト計画アクティビティと似ています。管理可能なグループを作成し、リスクを軽減し、それらのグループを中心にアクティビティを整理する手段として、移行ウェーブを使用することをお勧めします。

効果的で信頼性の高い移行ウェーブプランを作成するには、アプリケーションポートフォリオ、関連するインフラストラクチャ (コンピューティング、ストレージ、ネットワーク)、依存関係マッピング、移行戦略の全体像を把握する必要があります。

ソフトウェアとインフラストラクチャコンポーネントのコレクションをグループ化する形式のビジネスアプリケーションに加えて、他のグループレベルを使用できます。ウェーブは最も高いグループレベルです。ウェーブ内で、依存関係グループを作成できます。このタイプのサブグループには、複数のアプリケーションを含めることができます。たとえば、低レイテンシーなどの技術的な依存関係のために同時に移行する必要がある 2 つ以上のアプリケーションなどです。次に、その依存関係グループ全体が管理されます。複数の依存関係グループをウェーブに割り当てることができます。次に、ウェーブ全体またはウェーブ内の個々の依存関係グループに移行日を割り当てることができます。移行日は、そのグループが現在の場所で停止され、アクティブになる日時です AWS。

移行ウェーブには複数のアクティビティがあります。ウェーブをフェーズごとに整理し、フェーズごとに予想される期間を設定することをお勧めします。以下のフェーズが例として役立ちます。

- 設計: このウェーブフェーズでは、ウェーブ内の各アプリケーションのターゲット設計が確認され、承認されます。
- カットオーバー計画: このウェーブフェーズには、カットオーバーランブックの作成または反復と、アプリケーションの切り替えに必要なすべてのステップの計画 AWS が含まれます (ロールバックシナリオを含む)。
- 移行前: このフェーズには、アカウントのプロビジョニング、設定、移行前テスト、移行ツールのセットアップ、データレプリケーションなどのランディングゾーンのデプロイアクティビティが含まれます。
- カットオーバー: このフェーズでは、実際の移行が行われます。この間、アプリケーションは現在の場所で停止され、データは最後に同期され、ビジネステストが実行され、移行が確定されます。このフェーズには、運用上の引き渡しが含まれます。
- ハイパーケアまたは移行後: このフェーズは、移行チームが問題が発生した場合に運用をサポートできる期間です。また、必要に応じて最適化を適用することもできます。
- ウェーブクローズ: このフェーズでは、学習したメトリクスと教訓を確認し、ウェーブを正式にクローズします。

移行ウェーブには事前定義された期間はなく、労力と複雑さのレベルによって異なります。移行ウェーブは 6 ~ 10 週間以内に維持することをお勧めします。アプリケーションコンポーネントを完全に書き換えるなど、より多くの時間が必要な場合、通常、移行ウェーブの外でより適切に処理されます。

成功を測定し、進捗状況を追跡するには、ウェーブを成果とビジネスドライバーに合わせる必要があります。これは、ウェーブ期間とウェーブに含まれる依存関係グループにも影響します。ウェーブの完了には、測定可能なアチーブメントを反映する必要があります。

移行ウェーブを整理する方法は複数あります。次の表に、最も一般的なウェーブ組織オプションを示します。これらは通常結合されます。

ウェーブ組織タイプ	説明	長所	短所
移行戦略またはテクノロジースタック別	共通の移行戦略またはパターンを持つアプリケーションをウェーブに割り当てます。たとえば、リホストアプリケーションのみを含むウェーブなどです。	パターンまたはスタックあたりの専有チームには、ウェーブ全体を割り当てることができ ます。 アクティビティの同種期間。	特にさまざまなパターンに従うアプリケーションでは、依存関係をより詳細に分析する必要があります。
ビジネスドメイン別	ビジネスドメインごとにウェーブを作成します。例えば、注文管理ウェーブや支払いウェーブなどです。	通常、特定のドメイン内の共有データ。 チームの一貫した関与。	ビジネスドメイン全体が影響を受けることによるリスクの増加。
技術的能力別	1 つ以上の機能を使用するアプリケーションをグループ化します。例えば、コンピューティング専用ウェーブやコンピューティング + 負荷分散ウェーブなどです。	技術機能が時間の経過とともに有効になるにつれて、移行はより速く開始されます。完全に動作するランディングゾーンの依存関係を削除します。	後の波で複雑なポケットを作成します。
環境別	ウェーブには、一連のアプリケーションの特定の環境が含まれます。例えば、開発	非本番稼働用ウェーブは、実行時の柔軟性からメリットを得ます。 本番稼働用移行のリスクが軽減されました。	非本番環境に存在しない依存関係が欠落しないように、依存関係分析に重点を置く必要があります。

ウェブや本番ウェブなどです。

ビジネスの優先度別	特定の優先順位付け基準に基づいて純粹にグループを作成します。	ビジネス成果に対処します。	通常、多くのチームが関与しており、調整が困難です。
-----------	--------------------------------	---------------	---------------------------

アプリケーションポートフォリオのベースラインの確立に関するセクションでは、技術的な依存関係の 4 つのカテゴリについて説明しました。これらの依存関係は、移行ウェブの作成と依存関係グループの定義に役立ちます。依存関係グループは、依存関係の重要度によって決まります。さらに、非技術的な依存関係も考慮する必要があります。例えば、アプリケーションのリリーススケジュール、メンテナンスウィンドウ、主要な営業日 (月末や四半期末の処理など) は、ウェブプランに影響を与える可能性があります。

依存関係がソフトかハードかを判断します。ソフト依存関係は、2 つ以上のアセット間の関係、またはアセットから制約への関係であり、コンポーネントの場所には依存しません。たとえば、同じローカルネットワーク (または同じインフラストラクチャ) で動作する 2 つのシステムを分割するには、一方のシステムをクラウドに移動し、もう一方をオンプレミスのままにします。ハード依存関係は、2 つ以上のアセット間の関係、またはアセットから制約への関係であり、場所によって異なります。例えば、同じローカルネットワークで動作し、アプリケーションサーバーとデータベースサーバー間の通信の低レイテンシーに大きく依存する 2 つのシステムには、強い依存関係があります。これらのシステムの 1 つだけをクラウドに移動すると、機能またはパフォーマンスの問題が発生し、解決できません。同様に、リソースの可用性 (移行を実行するチームなど) や運用上の制約 (2 つのシステムを特定の期間にのみ移行できるメンテナンスウィンドウなど) などの非技術的な理由によって、これらのアセットに厳しい依存関係が生じる可能性があります。

移行ウェブプランを作成するには、依存関係を分析して依存関係グループを決定します。理想的には、特殊な検出ツールなどの信頼性の高いデータソースから決定します。この情報とアプリケーションの優先順位付け基準および運用状況を組み合わせてください。

技術的な依存関係の特定は困難です。複数のデータポイントが必要であり、すべてを含むデータソースはありません。例えば、検出ツールから process-to-process 通信情報を取得することはできますが、ソフト依存関係とハード依存関係に分類するのは困難です。レイテンシー耐性は、ネットワークデータのみから判断することも困難です。

以下の手法は、実際の依存関係を決定する曖昧さに対処するのに役立ちます。

- 「[データ要件](#)」セクションの説明に従ってすべてのデータを収集し、必要に応じて考慮したその他のデータポイントも収集します。
- 依存関係情報 (または通信データ) をフィルタリングし、Active Directory、バックアップ、トラフィックのモニタリングなどの共有サービスを除外します。技術共有サービスは、スコープ全体をグルー化する傾向があります。
- すべての情報を分類します。使用可能な場合は、コンポーネント間でネットワーク周波数とデータ転送ボリュームを使用します。
- アプリケーション所有者、アーキテクト、サポートチームとミーティングを行います。接続のタイプについて説明します。同期か非同期か。最小レイテンシー要件を認識していますか？重要な接続とは何ですか？また、使用できない場合はどうなりますか？重要な接続がありませんか？バッチプロセスが散発的に発生し、データセットに欠落している可能性があることに注意してください。
- 検出ツールがデータグラフを提供する場合は、アプリケーションの大規模なクラスターをブリッジする単一のアプリケーションを探します。これらの単一の接続ポイントは、データを小さなグループに分割するのに役立ちます。

[AWS Transform](#) は、依存関係を分析し、ウェーブプランニングを実行するのに役立ちます。

ウェーブプランの作成

アプリケーションのウェーブを移行するための前提条件は、アプリケーションポートフォリオデータと、ウェーブで移行されるアプリケーションのグループの詳細なアプリケーション評価です。詳細な評価には、ウェーブ内のアプリケーションのリスト、関連するインフラストラクチャの詳細、ターゲット設計、各アプリケーションの移行戦略を含める必要があります。

ウェーブの所有権とガバナンスを確立することは、ウェーブ作業、プログラムの依存関係、変更管理、問題、リスクを管理および追跡するために重要です。計画を管理するためのガバナンスフレームワークが整っていることを確認します。

ウェーブプランの概要を表示するには、デフォルトのウェーブコンストラクトから始めます。ウェーブ内で何が起こりますか？初期入力定義されたら、ウェーブを開始できます。通常、アクティビティは次のようになります。

1. カットオーバープランを絞り込みます。このアクティビティでは、他の内部チームや外部チームとの調整など、移行時に実行する必要があるランブックとステップについて概説する必要があります。
2. ロールバックプランを絞り込みます。問題が発生した場合にアプリケーションをロールバックするには、何をする必要がありますか？

3. ターゲットインフラストラクチャを準備します。たとえば、AWS ランディングゾーン (AWS アカウント、セキュリティ、ネットワーク、インフラストラクチャサービス、その他のサポートインフラストラクチャ) を作成または拡張できます。
4. ターゲットインフラストラクチャをテストします。
5. 移行ツールを操作します。たとえば、レプリケーションエージェントをインストールし、データ転送を開始します。
6. カットオーバープランとランブックのドライランを実施します。参加しているチームメンバー全員をグループ化し、すべてのステップを事前に確認します。
7. データレプリケーションとインフラストラクチャのデプロイをモニタリングします。
8. データインフラストラクチャとアプリケーションの運用の準備が整っていることを確認します AWS。
9. セキュリティの準備状況を確認します。
- 10 該当する場合は、コンプライアンスと規制要件 (ワークロード検証の移行前と移行後など) を確認します。
11. アプリケーションを AWS に移行し、本番稼働前のテストを実行します。
- 12 運用チームと移行チームが問題を解決するために完全に対応できる 3 日間など、移行後のサポートを提供し、最適化を適用します。
- 13 移行後のレビューを実施します。学んだ教訓を文書化し、将来の波に組み込んでください。
- 14 運用上の引き渡しとレポート用のメトリクスの取得を確認して、ウェーブクロージャを実行します。

これらの各アクティビティにかかる時間は、スコープの複雑さ、波の容量、関係者、および固有の状況によって決まります。可能な限り、遅延や移行ブロッカーの影響を軽減するため、ウェーブを小さくすることをお勧めします。チームで、ウェーブのデフォルト期間を決定します。

次に、日付を分析して、空のウェーブの初期高レベル構造を作成します (まだアプリケーションが割り当てられていません)。以下の質問を検討してください。

- 移行プログラムの合計期間を教えてください。
- 期限は何ですか？
- データセンターの終了日は固定されていますか？
- コロケーション契約の終了日はありますか？
- アプリケーションとインフラストラクチャの更新サイクルは何ですか？
- アプリケーションのメンテナンスとリリースサイクルは何ですか？

- 移行を避ける日付 (リリースとメンテナンスのサイクル、年末、祝日、月末処理など) がありますか？

これらの考慮事項を考慮して、ウェーブを計画にプロットします。移行プロセスを高速化するには、可能な限り波を重ねることをお勧めします。波が重複する鍵は、波内で何が起こるかを定義して検討することです。通常、デプロイアクティビティ、ターゲットインフラストラクチャの検証、データ同期はウェーブの前半に行われます。後半では、実際の移行、テスト、運用の引き渡しに焦点を当てます。つまり、プロセスの半分ごとに異なるチームが関係し、ある程度の効率を得ることができます。例えば、ターゲットインフラストラクチャの準備に関与したチームが作業を完了するとすぐに、次のウェーブの要件に取り組むことができます。一般的に、移行に対するファクトリのようなアプローチを容易にするために、ほとんどの波の長さと構造が似ていることをお勧めします。ただし、ウェーブプランニングプロセスでは、特定のウェーブのサイズを拡張して、依存関係や運用要件を満たすことができます。

次に、特定された依存関係グループに基づいて、含めることができる依存関係グループの数の観点からウェーブの最大サイズを決定します。波のサイズは通常、リスク選好度 (許容できる並列変更の量など) とリソースの可用性 (使用可能なリソース、スキル、予算で実行できる並列変更の量など) によって決まります。ただし、早期計画中は、リソースの要件と可用性によって制限しないでください。複数の依存関係グループを含むウェーブは、将来の反復で小さなウェーブに分解できます。

特定のウェーブの依存関係グループを確認したら、ウェーブを移行するためのリソース要件を確認します。リソース要件に基づいてウェーブサイズ (含まれる依存関係グループの数) を調整することを検討してください。これにより、波が小さくなったり大きくなったりする可能性があります。必要に応じて、すべてのウェーブが定義されるまでウェーブプランを繰り返します。プロセス全体でサポートするスペシャリストを提供できる AWS プロフェッショナルサービスまたは AWS 移行コンピテンシーパートナーと連携することを検討してください。

変更の管理

アプリケーションと関連するインフラストラクチャのポートフォリオは、移行プログラムのライフサイクル中に変更されます。長時間実行される移行プログラムは、通常のビジネスの進化と変化と共存します。アプリケーションは、移行を待つにつれて進化し続けます。サーバーは追加または削除され、新しいインフラストラクチャはオンプレミスにデプロイされます。ウェーブグループまたは依存関係グループの範囲を変更する必要があります。特に、移行日が近づいた場合、以前に不明な依存関係が特定された場合、または新しいサーバーがインベントリに含まれている場合は、変更が必要です。これは、移行自体中に発生することがあります。

スコープの変更は、依存関係グループとウェーブに影響します。変更を処理し、影響を最小限に抑えるには、スコープ制御メカニズムを確立することが重要です。スコープ変更管理メカニズムには、スコープの単一の信頼できるソースの定義が必要です。これは、移行プログラムのガバナンスで定義されているように、スコープを管理するためのツール、または .csv ファイル、スプレッドシート、またはデータベースである場合があります。関係者がアクションを実行できるように、変更を特定し、影響を分析し、変更を関係者に伝える必要があります。ウェーブプランは結果として反復されます。

詳細なビジネスケース

この段階では、ビジネスケースの範囲を検証して拡張し、変換プログラムをサポートするためにより詳細なレベルを提供することをお勧めします。すぐに組み立てられる最初の方向性のあるビジネスケースは、基本的なステップと次のレベルの詳細な計画に投資するのに十分な信頼性を提供するように設計されています。

詳細なビジネスケースを作成すると、次の方法でこの計画プロセスがサポートされます。

- 移行およびモダナイズすべき内容、選択するオプション、作業を段階的に優先順位付けする方法に関する意思決定に役立つ財務分析を提供する
- 詳細な再検討により、元の方向性財務ケースを検証、改善、開発します。
 - インフラストラクチャのコスト削減の可能性
 - 社内の IT 生産性と外部委託された運用効率
 - プログラムのセットアップ、移行、モダナイゼーションに必要な投資の見積もり
- 移行がもたらすさらなる価値の推進要因を追跡するプロセスの特定、スケールの推定、設定

詳細なビジネスケースでは、以下を確立します。

- 少なくとも移行の第 1 フェーズを実装するためのマンデートと投資を確保する目標ベース
- プログラムに対するベースラインの最小財務パフォーマンスの期待値
- さまざまな移行設計と優先順位付けの決定が行われる財務基盤を明確にし、プログラムの過程で状況や人が変化したときに、新しいリーダーシップが情報に基づいた選択を行うことができます。
- ワークロードの移行と運用の開始時に最初の使用状況データが利用可能になった後に調査するコスト最適化の増分領域に関するインサイト
- 回復力と俊敏性の向上により、クラウドトランスフォーメーションがビジネスにもたらす価値を見積もる

- 回復力と俊敏性の向上による財務収益を見積もるために使用される、関連する KPIs、メトリクス、仮定。これにより、プログラムからの主なメリットの実現を促進するためのベースラインが形成されます。

ケースに必要なシナリオを決定する

詳細なビジネスケースを構築する場合、通常、ビジネスケースが使用されるさまざまな目的をサポートする複数のシナリオを開発する必要があります。

最小変更シナリオ – 財務パフォーマンスの最小期待値を評価するには、ステータスクォーへの最小予想変更を想定したシナリオを準備します。このシナリオは、最悪の場合のバックストップとして、移行に投資する権限を取得する際に役立つサポートを提供します。このシナリオでは、可用性や耐障害性など、他のquality-of-serviceのニーズに対して予想される最小の容量増加度と最小限の変更をモデル化します。最小の変更により、現在の運用モデルのコストとリソースの非効率性が最も低くなります。

最も可能性の高いシナリオ – プログラム戦略と優先順位付けの決定を通知するには、ビジネスが期待することを反映するシナリオを準備します。このシナリオには、ピーク時の使用率の増加または減少の可能性と、ビジネスからの高レベルのサービス品質 (特に可用性と回復力) の需要を満たすためのアップグレードコストを含める必要があります。

その他の特定のシナリオ – ビジネスケースに大きな影響を与える可能性のある仮定をまだ作成する必要がある場合は、その仮定が当てはまるかどうかを示すシナリオを開発します。ただし、これらの代替シナリオの数は絶対的に最小限にしておくことをお勧めします。合計 3 つを超えるシナリオは進行が遅く、コストがかかり、混乱し、維持が困難になります。可能な限り、実験を行い、より大きな仮定を削除してください。

インフラストラクチャと移行のコストモデルを検証して絞り込む

ポートフォリオ分析を完了し、ターゲットの設計とサイズ設定を準備したら AWS のサービス、シナリオ AWS ごとに の現在の運用モデル (COM) と将来の運用モデル (FOM) の運用コストの見積もりを絞り込みます。通常、以下の見積もりを絞り込む必要があります。

- ハイパーバイザーホストサーバー、ベアメタルサーバー、ストレージ、ネットワークデバイス、セキュリティアプライアンスハードウェアの更新、インストール、メンテナンスの COM インフラストラクチャコスト。シナリオに必要な容量の実際の料金と割引レベルで計算します。
- COM データセンターとコロケーションされた施設のコストには、スペース、冷却、電力、ラック、無停電電源装置 (UPS)、ケーブル、物理セキュリティシステム、成長に合わせてサイズ設定さ

れ容量を満たすように指定されたシステム、シナリオの高可用性とディザスタリカバリ (DR) レベルが含まれます。

- WAN リンク、コンテンツ配信ネットワーク、仮想プライベートネットワーク (VPNs) のコストを含む COM ネットワークサービスのコストは、シナリオの接続、帯域幅、スループット、レイテンシーのニーズに対して契約料金を使用して計算されます。
- 既存の契約に基づく COM アプリケーションとインフラストラクチャソフトウェアのコスト。シナリオの使用量の増加または削減を提供します。
- 洗練されたサービスアーキテクチャ、インスタンスサイズ、優先料金モデル、予想される使用量、使用状況の変動性に基づいて、必要に応じて技術サポートとマネージドサービスを含む FOM AWS ユーティリティコスト。
- 最終的なアプリケーション設計、アプリケーションを実行するインフラストラクチャの設定、時間の経過に伴う増加、ライセンスの移管可能性ルールに基づく FOM アプリケーションライセンス。
- FOM 移行とモダナイゼーションのコスト見積もり。シナリオのベースライン移行ウェーブプランを反映するように改良され、特にリプラットフォーム、再購入、またはリファクタリングされるワークロードごとにコストを提供するように詳述されています。
- アセットの償却コストと契約早期終了コストの見積もり、ベースライン移行ウェーブプランの廃止タイミングを反映するように改訂された FOM 廃止コスト、償却を最小限に抑えるために再利用できるアセットと切り替えることができるアセットの検証、物理アセットとメディアの廃棄コスト。
- 移行の並列実行コストは、各移行カットオーバーと既存の各サービス廃止のタイミングを反映するように調整されています。

IT 生産性と IT 運用を改善し、効率バリューモデルをサポートする

方向性のあるビジネスケースと同様に、IT 運用とサポートに関する価値モデルを改良および開発するには、主に 2 つのアプローチがあります。選択するアプローチは、COM が社内管理されているか、請負業者または外部委託サービスで管理されているかによって異なります。

社内チームの生産性向上

IT 運用とサポートが社内管理されている場合、ビジネスケースの焦点は次の通りです。

- 移行と対象範囲に含まれる運用自動化による生産性の向上を特定して定量化する
- 社内チームのために解放された時間が、通常より価値の高い他のアクティビティに簡単かつ生産的に適用できることを検証し、チームに進歩の機会とより大きな報酬を与え、組織により多くの価値をもたらす

チーム内の各ロールの各メンバーがさまざまな通常のアクティビティに費やした時間のメリットニーズの分析と、さまざまなアクティビティのワークロードの予想される削減に関するガイダンスを評価します。

次の表は、IT オペレーションの大部分を消費し、チーム内のさまざまなロールの労力をサポートするタスクのアクティビティ別のワークロード削減の一般的なレベルに関する初期ガイダンスを示しています。この表には、生産性の実現方法の説明が含まれています。

 Note

リストされているアクティビティは通常、複数の異なるロールのチームメンバーによって実行されるため、各タスクの生産性の節約は、チーム内のすべてのロールにわたって評価する必要があります。たとえば、インフラストラクチャタワー (コンピューティング、ストレージ、ネットワークなど) 別に編成された IT 運用チームでは、設備投資の計画と予算編成が各タワーのタワーリードに共通する場合があります。

運用およびサポートアクティビティ	削減額レベル	生産性ドライバー
インフラストラクチャ設計	中	設計が簡素化され、考慮するパラメータが少なくなります。
設備投資の計画と予算編成	高	OPEX 中心の Elastic サービスは、実質的にすべての予算と計画の問題を削除します。
購入	高	の確立後、調達 AWS アカウントが大幅に簡素化されます。
容量プランニング	中～非常に高い	ネットワーキングとコンピューティングキャパシティ管理のワークロードは通常すべて排除され、ストレージでは大幅に簡素化されます。

チューニング	非常に高い	インスタンスのサイズはいつでも変更できるため、マネージドサービスにチューニングは必要なく、他のサービスにはほとんど必要ありません。
ハードウェア障害の管理	非常に高い	クラウドでのハードウェア処理のすべての側面は、によって透過的に処理されます AWS。
サーバーの可用性と通信のモニタリング	高	AWS ツールのサポートと自動化により、モニタリングと通信は広範囲に簡素化されます。
セキュリティ管理	中	ワークロードは、AWS セキュリティ機能や、AWS クラウド ハードウェア、ソフトウェア、ネットワーク、施設の セキュリティ責任 AWS を持つことで大幅に削減されます。
ネットワークとストレージのアップグレード、メンテナンス、パッチ。	非常に高い	クラウドでのネットワークとストレージのメンテナンスのすべての側面は、によって透過的に処理されます AWS。
ラックとスタッキング — ハードウェアの物流	非常に高い	クラウドでのハードウェア管理のすべての側面は、によって透過的に処理されます AWS。
バックアップ	中	バックアップは、AWS ツール、柔軟なストレージシステム、自動化によって広範囲に簡素化されます。

マネージドサービス (Amazon S3、Amazon RDS AWS Lambda、 など AWS Fargate)	非常に高い	マネージドサービスは、 によって完全に管理されている環境で実行されるため AWS、メンテナンス、パッチ適用、モニタリング、プロビジョニングの管理アクティビティは必要ありません。
デバイスとサービスのセットアップとコミッショニング	非常に高い	に移行した AWS エステートのハードウェアセットアップのアクティビティは通常、VPN VPNs または AWS データセンターへの接続を確立するための WAN AWS Direct Connect 接続デバイスを除き、削減されます。
エンドポイント保護とウイルス対策保護	高	エンドポイント保護とウイルス対策サービスのアプリケーションとメンテナンスは、通常、移行設計の一環として広範囲に自動化されています。
脅威、脆弱性、リスク評価	高	AWS は、コアプラットフォームに重点を置いた、この要素のサポートを提供します。また、AWS が提供するメカニズムにより、安全なアーキテクチャでの評価が簡素化されます。

データセンターインフラストラクチャプロジェクト管理	高	インフラストラクチャサービスの拡張、更新、または廃止のためのインストール作業のためのプロジェクト管理。インフラストラクチャソフトウェアとサービスの一部の管理は残りますが、これはオンプレミスインフラストラクチャよりもはるかにシンプルで、ハードウェアアクティビティは排除されます。
データセンター施設管理	中～非常に高い	すべてのサーバー、ストレージデバイス、セキュリティアプライアンス、および関連するラックに起因する施設管理作業は、移行されるすべてのものに対して削除されます。ただし、通常、一部の作業は、WAN リンクネットワークデバイスとハイブリッドアーキテクチャでオンプレミスに保持されているインフラストラクチャの施設を提供するためのままです。
アプリケーションアーキテクチャ、開発、管理、テスト	低	アジャイル開発ツールチェーンを、必要に応じてテスト環境を構築するためのアプリケーションスタックのインスタンス化と破壊の自動化と組み合わせて使用すると、アプリケーション開発のリードタイムが短縮され、多くの手動テストステップが不要になります。

アプリケーションソフトウェア のインストールと設定	中	アプリケーションスタック の完全なインストールと設 定は、 などのサービスを使 用して簡単に自動化 AWS CloudFormation され、ラン ディングゾーンを使用して簡 素化されます。ランディング ゾーンは、 を使用して簡単 に設定できます AWS Control Tower。
IT サポート	中	L1 および L2 サポートの削減 は、セルフサービスのプロビ ジョニングに Service Catalog 機能を使用することで容量と パフォーマンスの問題を削減 し、低コストの高可用性アー キテクチャの使用を増やすこ とで実現されます (停止の削 減と自動スケーリングとエッ ジコンピューティングの設 定)。
データベース管理	最小低	これらのアクティビティはほ とんど変更されません。通 常、 のリソースは AWS オ ンプレミスインフラストラク チャと同じレベルで提供され ます。
インフラストラクチャとセ キュリティの要件のキャプ チャ、分析、設計	Minimal	
ドキュメント	Minimal	
アプリケーションとパフォー マンスのモニタリング	Minimal	
L3 テクニカルサポート、クエ リへの回答、トラブルシュー ティングと問題解決	Minimal	

アプリケーションソフトウェア
のインストールと設定 Minimal

アプリケーション L3 のサポー
ト (予算編成と長距離キャパシ
ティプランニングを除く) Minimal

次の表は、ワークロード削減の各レベルで予想される削減額を示しています。

レベル	予想
非常に高い	85% ~ 100%
高	60% ~ 90%
中	30% ~ 70%
低	10% ~ 35%
Minimal	0% ~ 10%

これらのメトリクスは、生産性の向上を評価し、詳細なビジネスケースに含めるための出発点となります。実際の生産性の向上は、特定の状況によって異なります。範囲の中間と下限の両方で生産性の節約を計算して、一般的なシナリオと保守的なシナリオを推定すると便利です。

プログラムが進むにつれて、各アクティビティに費やされた時間の実際のデータをロールごとにキャプチャすることが重要です。このデータは、運用を推定するための改善されたベースを構築し、新しいプロジェクトとサービスの拡張のコストをサポートします。

外部委託された IT 運用とサポートのコスト削減

IT 運用とサポートが主に請負業者に委託または管理されている場合、パートナー AWS 主導のを含むマネージドサービスソリューションを提供する AWS パートナーに見積りをリクエストすることで、将来の運用モデル (FOM) のコスト配分を準備することができます [AWS Managed Services](#)。 [「方向性のあるビジネスケースの作成」](#) セクションの [「運用コストの最適化の構築」](#) のサブセクションで説明されているように、AWS アカウントマネージャーに連絡して AMS の料金を直接リクエストすることもできます。

詳細なビジネスケースでは、ベンチマークの数値を、改訂された AWS のサービス 部品表と予想されるサービス消費量、AMS パッケージと必要なオプション、および必要なサービスレベルに基づく引用に置き換えます。コストには、1 回限りの実装コンポーネントと消費ベースの実行レートがあります。

残りの IT オペレーション、移行されないサービスで保持する必要があるサポート AWS、契約上のペナルティ (早期終了など) がある場合は 1 回限りのコストを含めます。

レジリエンスバリューモデルを開発する

では AWS、高可用性、ディザスタリカバリ、耐障害性に優れた幅広いアーキテクチャを構築できます。消費ベースの料金とは、サービスが使用時にのみ課金されることを意味します。これら 2 つの要因を組み合わせることで、耐障害性に優れたコストパフォーマンスが得られます。

さらに、AWS お客様はこれを使用してワークロードの耐障害性を向上させています。[IDC 2018 調査](#)では、参加している顧客が年間停止回数を 73% 削減し、平均復旧時間 (MTTR) を 58% 削減し、生産性の低下を 94% 削減した例を示しています。同じ調査では、レジリエンスの向上によって得られる財務上のメリットは、IT インフラストラクチャのコスト削減のメリットよりも 50% 大きいことがわかりました。

さらに、アプリケーションのソフトウェア開発ライフサイクルをモダナイズすることで、さらなる回復力が達成されます。テストオートメーションを備えた CI/CD パイプラインが導入されてビジネスの俊敏性が向上した場合、ソフトウェア欠陥は開発サイクルの早い段階で発見され、ソフトウェアメンテナンスコストが大幅に削減されます。

ビジネスケースでこの価値を評価して含めるには、まずアプリケーションのビジネスオーナーと協力して、移行する各ワークロードの全体的なメリットの機会を把握します。これには、次の項目が含まれる場合があります。

- サービスの中断の数、平均期間、性質
 - サービスの中断の例としては、停止、パフォーマンスの減速、計画的なバッチおよびメンテナンスウィンドウのオーバーラン、主要な関数のバグ、ピーク時のアクセススロットリングなどがあります。
- e コマースシステムなどの収益を生み出すサービスの中断による収益への影響
 - 中断時間とトランザクションレートに基づいて、サービスの中断によって完了できない可能性があるトランザクションの数
 - 影響を受ける各トランザクションの平均値

- サポートエンジニアが本番稼働システムの欠陥を解決する時間の追加コストと、開発プロセスの早い段階で発見するコストの比較
- 内部ユーザーの生産性と損失時間のコストへの影響

次に、予想されると、サービス中断による損失時間のより控えめな削減を評価し、耐障害性を高める必要があります。たとえば、次の項目を含めることを検討してください。

- 高可用性アーキテクチャを使用して停止回数と MTTR を減らし、目標復旧時間 (RTO) と目標復旧時点 (RPO) を改善
- 自動スケーリングなどの機能を使用した、スローダウンの削減、容量スロットリングの排除、バッチ処理のオーバーランの回避
- CI/CD パイプラインの実装とインフラストラクチャのスピンアップとスピンドアンの自動回帰テストにより、本番環境でのみ検出されるアプリケーションのバグの数を削減し、コストを最小限に抑えます。

これらを移行およびモダナイズするアプリケーションのポートフォリオにまとめ、ケースの毎年の予想およびより保守的なビジネス価値の数値を計算します。メリットは移行スケジュールに沿って拡大し、寄与するアプリケーションの使用量の増加の期待に沿ってボリュームを拡大する必要があります。

ビジネスの俊敏性バリューモデルを開発する

ビジネスの俊敏性は、AWS 顧客が移行する主な理由です AWS。 [IDC 2018 の顧客調査](#)では、ビジネス俊敏性のメリットは、測定される合計メリットの 47% を占め、インフラストラクチャのコスト削減によるメリットの 5 倍以上を占めていることを示しています。 AWS

あらゆるトランスフォーメーションから生じるビジネス俊敏性のメリットを正確に予測することは困難です。ただし、多数のユーザーをサポートするアプリケーションやビジネス上の差別化要因であるアプリケーションに焦点を当てることで、この利点の重要な部分をモデル化し、ベースラインの詳細なビジネスケースに含めることができます。

移行が進むにつれて、より多くのメリットが定量化可能になるにつれて、ビジネス俊敏性バリューモデルを段階的に改善および拡張します。これにより、ビジネスケースが関連付けられるため、プログラムを運営する主要な意思決定サポートツールとして使用できます。

ビジネスの俊敏性バリューモデルを構築するには、次のガイダンスを使用します。

- 次のような、ビジネスパフォーマンスを最大限に向上させる機会があるワークロードを選択します。
 - 収益を生成するワークロード
 - 効率の向上を推進し、ビジネスからコストを排除するスコープを持つビジネス運用ワークロード
 - 大規模なユーザーベースをサポートするビジネス生産性向上ツール
- 収益と効率を生成するワークロードについては、以下を実行します。
 - アプリケーションのメジャーアップグレードとマイナーアップグレードが推進されると予想される収益増加または運用効率について、現実的かつ保守的な評価を行います。
 - アプリケーション開発速度を高め、インフラストラクチャのデプロイ時間を短縮できる 1 年あたりのメジャーリリースとマイナーリリースの数 AWS の増加を見積もります。このベースラインメトリクスの一部は、IDC レポートに記載されています。
 - 現実的で保守的なメリットの期待値を計算します。ビジネスケースの期間にわたってマッピングし、それぞれのワークロードが移行されてからしばらくして、効率を最大に引き上げることができます。
- ビジネス生産性向上ツールの場合は、以下を実行します。
 - メジャーなアプリケーションアップグレードとマイナーなアプリケーションアップグレードが推進できることが予想される時間の節約について、現実的で保守的な評価を行います。
 - 影響を受けるユーザーベース全体の人々の時間と労力の平均コストを見積もります。
 - メジャーリリース頻度とマイナーリリース頻度の増加の数値を使用し、ビジネスケースの期間中のメリットを計算します。

デベロッパーの生産性の向上と起動時間の短縮により、追加のリソースが不要になるため、割引キャッシュフロー、NPV、ROI、MIRR、およびペイバック計算に含めるために、各ワークロードの純利益ラインをビジネスケースのキャッシュフローモデルに追加します。

継続的な評価と改善

この評価ステージでは、次の 2 つの側面に焦点を当てます。

- アプリケーションの波ごとに進行中の詳細なアプリケーション評価
- ポートフォリオの継続的な進化と改善

最初の側面である継続的な詳細なアプリケーション評価では、特定の波の各アプリケーション、提案された AWS 設計、移行戦略を完全に理解するために、アーキテクチャとテクノロジーレベルまで詳細な検出と分析に焦点を当てています。この移行準備状況の評価は、特定の移行ウェーブを開始するための前提条件です。

ポートフォリオの継続的な進化と改善という 2 つ目の側面は、ポートフォリオ管理と、ビジネスケースの進化と追跡など、時間の経過とともにアプリケーションを改善する計画を立てる方法に焦点を当てています。

このステージの主な移行結果は次のとおりです。

- 各ウェーブの検証済み移行スコープ
- 特定の移行ウェーブにおけるアプリケーションの文書化されたターゲットアーキテクチャと移行戦略
- 識別および検証された移行パターンとツール
- ウェーブごとの文書化された要件 (セキュリティ、AWS インフラストラクチャ、運用) と移行のカットオーバーに関する考慮事項

このステージの主な最適化結果は次のとおりです。

- ポートフォリオ合理化モデルとビジネス成果
- アーキテクチャとテクノロジーの変更案とその期待される利点
- プラットフォームの要件 (セキュリティ、AWS インフラストラクチャ、オペレーション)
- 実装計画

継続的な評価データ要件を理解する

アプリケーションポートフォリオの継続的な評価と改善のためのデータ要件は、前のセクションのデータ要件の組み合わせです。ポートフォリオの移行とその進化を継続的に管理するには、以下のセクションを参照してデータ要件を理解してください。

- ウェーブ評価とアプリケーションの最適化には、[「優先順位付けされたアプリケーション評価」](#)セクションのデータ要件を使用します。
- 継続的なポートフォリオ管理には、[ポートフォリオ分析と移行計画](#)セクションのデータ要件を使用します。
- ウェーブプランの定義については、[「ウェーブプランニング」](#)セクションを参照してください。

詳細なウェーブ評価

アプリケーションの詳細な評価は、移行ウェーブに先立って、移行の主要なイネーブラーとして、[優先順位付けされたアプリケーション評価](#)ステージと同じ要件と推奨事項があります。目標は、特定のウェーブにおけるアプリケーションの現在の状態を詳細に理解し、運用面、ツール、特定の移行パターンなど、将来のアーキテクチャ設計と移行戦略を作成することです。

[優先アプリケーション評価](#)を特定のウェーブ内のアプリケーションのグループに適用します。移行計画の各ウェーブの前にこのプロセスを繰り返します。重要なのは、詳細な評価からウェーブの開始までの十分な時間をスケジュールすることです。必要な時間は、ウェーブ要件を実装し、移行を実行するプラットフォームチームと移行チームの要件によって決まります。これらのチームと協力して、詳細なウェーブ評価とウェーブをスケジュールします。生産ラインをエミュレートする工場のようなモデルを実装することをお勧めします。

最適化とモダナイゼーションの評価

すでに移行されているワークロードの最適化とモダナイゼーションの評価プロセスは AWS、移行するワークロードの評価と似ています AWS。主な変更点は、評価を実行するデータソースです。には AWS、で実行されているアプリケーションに関する詳細情報を取得するために使用できる out-of-the-box 使えるツールやサービスがいくつかあります AWS。

アプリケーションを最適化およびモダナイズする内容と方法は、独自の推進要因と状況に基づきます。最適化は、現在のアーキテクチャとテクノロジーに変更を適用してコストを削減し、パフォーマンス要件を適応させ、学んだ教訓を組み込むことに重点を置いています。モダナイゼーションは、

サーバーレスモデルやマイクロサービスアーキテクチャの導入など、アプリケーションを次のレベルに進めることに重点を置いています。

[優先順位付けされたアプリケーション評価](#)のガイドラインに従ってください。最適化とモダナイゼーションの取り組みをさらに支援するには、次のリソースを参照してください。

- [AWS コスト最適化](#)は、IT 最適化と IT コストの節約に関する情報を提供します。
- [AWS Compute Optimizer](#) は、機械学習を使用して過去の使用率メトリクスを分析することで、コストを削減し、パフォーマンスを向上させるためにワークロードの AWS リソースを推奨します。
- [AWS コストと容量の最適化サービスとツール](#)は、コンピューティングリソースの管理に役立つため、コンピューティングコストの構築により多くの時間を費やし、管理時間を短縮できます。
- [Amazon S3 Storage Lens](#) は、オブジェクトストレージの使用状況とアクティビティの傾向を組織全体で可視化します。これにより、コスト効率を向上させ、データ保護のベストプラクティスを適用するための実用的なレコメンデーションが作成されます。
- [Database Freedom](#) は、AWS データベースおよび分析サービスへの移行を容易にします。
- [Amazon CodeGuru](#) は、コード品質を向上させ、アプリケーションの最も高価なコード行を特定するためのインテリジェントなレコメンデーションを提供する開発者ツールです。
- [AWS ハイブリッドクラウドサービス](#)は、クラウドからオンプレミス、エッジまで、必要な場所で一貫した AWS エクスペリエンスを提供します。

その他のリソース

- [コストの最適化とイノベーション: アプリケーションのモダナイゼーション入門](#) (ブログ記事)
- [サーバーレスウェブアプリケーションのコストの最適化](#) (ブログ記事)
- [Windows on AWS](#) (ブログ)
- [最新のアプリケーション](#)
- [アプリケーションのモダナイゼーション](#) (AWS re:Invent 2020)
- [AWS マイクロサービスガイド](#)

ウェーブプランの反復

移行プログラムが進み、より多くのウェーブが移行されるにつれて、学んだ教訓と変化するビジネスの優先順位に基づいて移行ウェーブプランを進化させることが重要です。特に、長期にわたる移行プ

プログラムでは、ビジネスドライバーと組織の変化を再評価し、移行ウェーブプランがまだ有効であることを確認することが重要です。

同様に、移行から学んだ教訓は、ウェーブプランの構成と各ウェーブの範囲に影響します。何が起きているかの可視性が失われないように、[ウェーブプラン](#)を最新の 状態に保つ必要があります。計画では、何が配信されているかを反映および追跡し、移行範囲の変更を管理および評価する必要があります。

ビジネスケースの進化と追跡

移行が進むにつれて、特に長時間実行されるプログラムでは、ビジネス上のプレッシャーによって移行とモダナイゼーションの優先順位が定期的に再検討されることは避けられません。

新しい情報が利用可能になるとビジネスケースを進化させ、詳細なビジネスケースに記載されている期待に照らして実際の商用パフォーマンスを追跡することをお勧めします。これらの推奨事項には以下が含まれます。

- ビジネスの優先順位に影響を与え、IT 戦略とそれに伴うアプリケーションポートフォリオに影響を与える組織における新しい構造変更
- アプリケーションポートフォリオの一部の商業上の重要性の高まり、または移行とモダナイゼーションが達成の対象となる変更
- サイズ調整、数値化、段階的なモダナイゼーションのケースの確認など、移行されたアプリケーションの実際のリソース使用率データの可用性
- IT 運用とサポート活動に費やされた労力に関するデータの可用性、および運用上の改善と自動化の可能性の分析
- ソフトウェア開発とメンテナンスのサイクル時間の変化、開発ステージとサービスの可用性情報別のソフトウェア不具合、さらに改善が見込める領域の根本原因分析を測定するデータの可用性

ビジネスケースに対するパフォーマンスを追跡することで、移行開始後により簡単に評価および定量化できるさらなる改善を含めるようにケースを進化させることができます。プログラムガバナンス組織は、変化するビジネス圧力に対応し、管理可能で許容可能なレベルのリスクで最大の価値をもたらす方向に変革を誘導するはるかに優れた能力を備えています。

これは、ケース内の IT 生産性、レジリエンス、ビジネスの俊敏性にとって特に重要です。これらは通常、事前に評価することが難しい大きな要因と難しい要因の両方です。これらのドライバーのパフォーマンスを追跡することで、チームはメリットの実現を妨げる問題を深く掘り下げて解決できま

す。または、ビジネスケースを調整して、最も進行中の財務パフォーマンスの最適化を実現するイニシアチブに優先順位を付けることもできます。

リソース

AWS リファレンス

- [Amazon Builders' Library](#)
- [アプリケーションのモダナイゼーション](#) (AWS re:Invent 2025)
- [アプリケーションポートフォリオ評価戦略](#)
- [AWS アーキテクチャセンター](#)
- [AWS Compute Optimizer](#)
- [AWS コストと容量の最適化サービスとツール](#)
- [AWS コスト最適化](#)
- [コストの最適化とイノベーション: アプリケーションのモダナイゼーション入門](#) (ブログ記事)
- 「[検出、計画、推奨の移行ツール](#)」
- [AWS Documentation](#)
- [リソースセンターのご利用開始にあたって](#)
- [AWS Marketplace](#)
- [AWS Managed Services](#)
- [AWS マイクロサービスガイド](#)
- [AWS 移行コンピテンシーパートナー](#)
- [最新のアプリケーション](#)
- [サーバーレスウェブアプリケーションのコストの最適化](#) (ブログ記事)
- [AWS 規範ガイダンス](#)
- [AWS \[プロフェッショナルサービス\]](#)
- [AWS ソリューションライブラリ](#)
- [Windows on AWS](#) (ブログ)

AWS のサービス

- [AWS App2Container](#)
- [AWS Transform MGN](#)

- [AWS Transform](#)
- [Kiro](#)
- [AWS Control Tower](#)
- [AWS Database Migration Service](#)
- [AWS DataSync](#)
- [AWS Direct Connect](#)
- [Amazon ECS](#)
- [Amazon EKS](#)
- [AWS Fargate](#)
- [AWS Managed Services](#)
- 「[Migration Evaluator](#)」
- [AWS ランディングゾーン](#)
- [AWS 料金見積りツール](#)
- [AWS Schema Conversion Tool](#)
- [Amazon S3 Storage Lens](#)

ドキュメント履歴

次の表に、この戦略の重要な変更点を示します。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
更新	ウェーブプランニング セクションを展開しました。 6 Rs 移行戦略 セクションを簡素化しました。	2026 年 6 月 2 日
更新	ポートフォリオ検出と初期計画セクション「検出の加速と初期計画」の名前を変更し、ディシジョンツリー図を更新しました。	2024 年 5 月 20 日
二	初版発行	2021 年 11 月 12 日

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。