



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

AWS 規範的ガイダンス



AWS 規範的ガイド: 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

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

Table of Contents

はじめに	1
概要	1
ポートフォリオの発見と初期計画	4
初期評価データ要件の理解	4
データソースとデータ要件	4
ディスクバリエーションの必要性の評価	16
ビジネス推進要因と技術的指針	21
ビジネスドライバー	21
技術指導原則	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
添付ファイル	31
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	32
フォーカス・バリュードライバー	33
データニーズ	34
ビルディングインフラストラクチャの TCO 比較	34
運用コストの最適化を組み込む	36
全面的なビジネスケースへの拡大	37
移行および近代化プログラムのセットアップの見積もり	39
優先アプリケーション評価	49
詳細な評価データ要件の理解	49
詳細なアプリケーションアセスメント	59
全般	61
アーキテクチャ	61
操作	61
パフォーマンス	62
ソフトウェアライフサイクル	62
移行	62
回復性	63
セキュリティとコンプライアンス	63
データベース	63

依存関係	64
AWSアプリケーション設計と移行戦略	64
アプリケーションのfuture 状態	65
再現性	66
要件	66
将来のアーキテクチャ	66
アーキテクチャに関する決定	69
ソフトウェア使用率環境	70
タグ付け	70
マイグレーション戦略	70
移行パターンとツール	70
サービスの管理と運用	71
カットオーバーに関する考慮事項	71
リスク、前提条件、問題、依存関係	72
実行コストの見積もり	72
.....	73
完全な評価データ要件の理解	73
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	84
優先順位付け基準の繰り返し	86
7 R の移行戦略選択の繰り返し	88
ウェーブプランニング	89
ウェーブプランの作成	91
変更管理	93
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コストモデルの検証と改良	95
ITの生産性とIT運用を改善し、効率化のバリューモデルをサポートします	96
レジリエンスバリューモデルの開発	103
ビジネスアジリティバリューモデルの開発	104
継続的な評価と改善	106
継続的評価データ要件の理解	107
詳細なウェーブアセスメント	107
最適化と最新化のための評価	107
ウェーブプランの繰り返し	108
ビジネスケースの進化と追跡	109
リソース	111

寄与要因	113
ドキュメント履歴	114
用語集	115
#	115
A	116
B	119
C	120
D	123
E	127
F	129
G	130
H	131
I	132
L	134
M	135
O	139
P	141
Q	143
R	143
S	146
T	149
U	151
V	151
W	152
Z	153
.....	cliv

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

Amazon Web Services ジャーマン・ ゴンカルベス、マーク・ バーナー、ザック・ ハンセン (AWS)

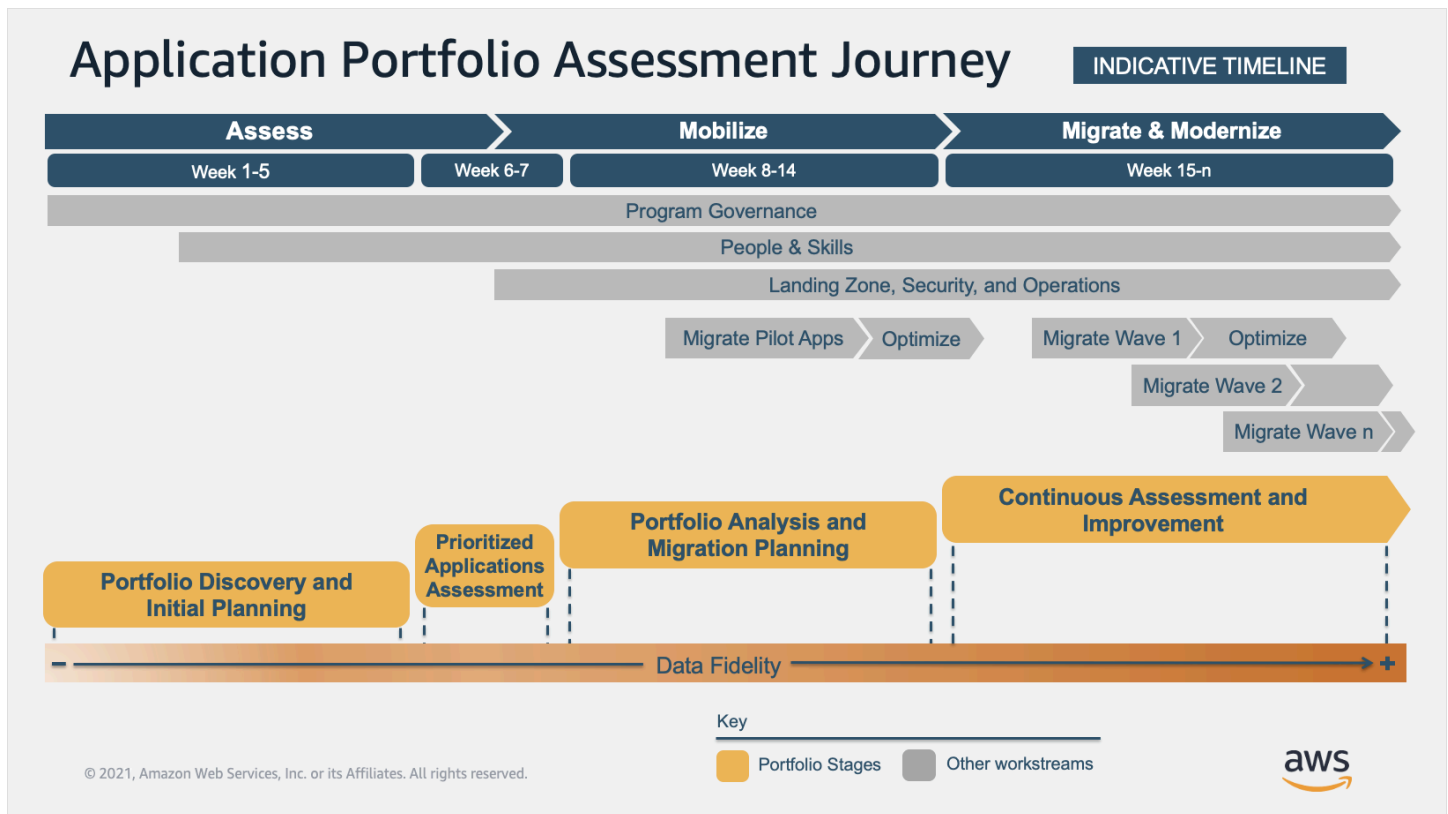
2021年11月 ([ドキュメント履歴](#))

このAmazon Web Services (AWS) 規範的ガイド文書では、[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ています](#)。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、発見、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

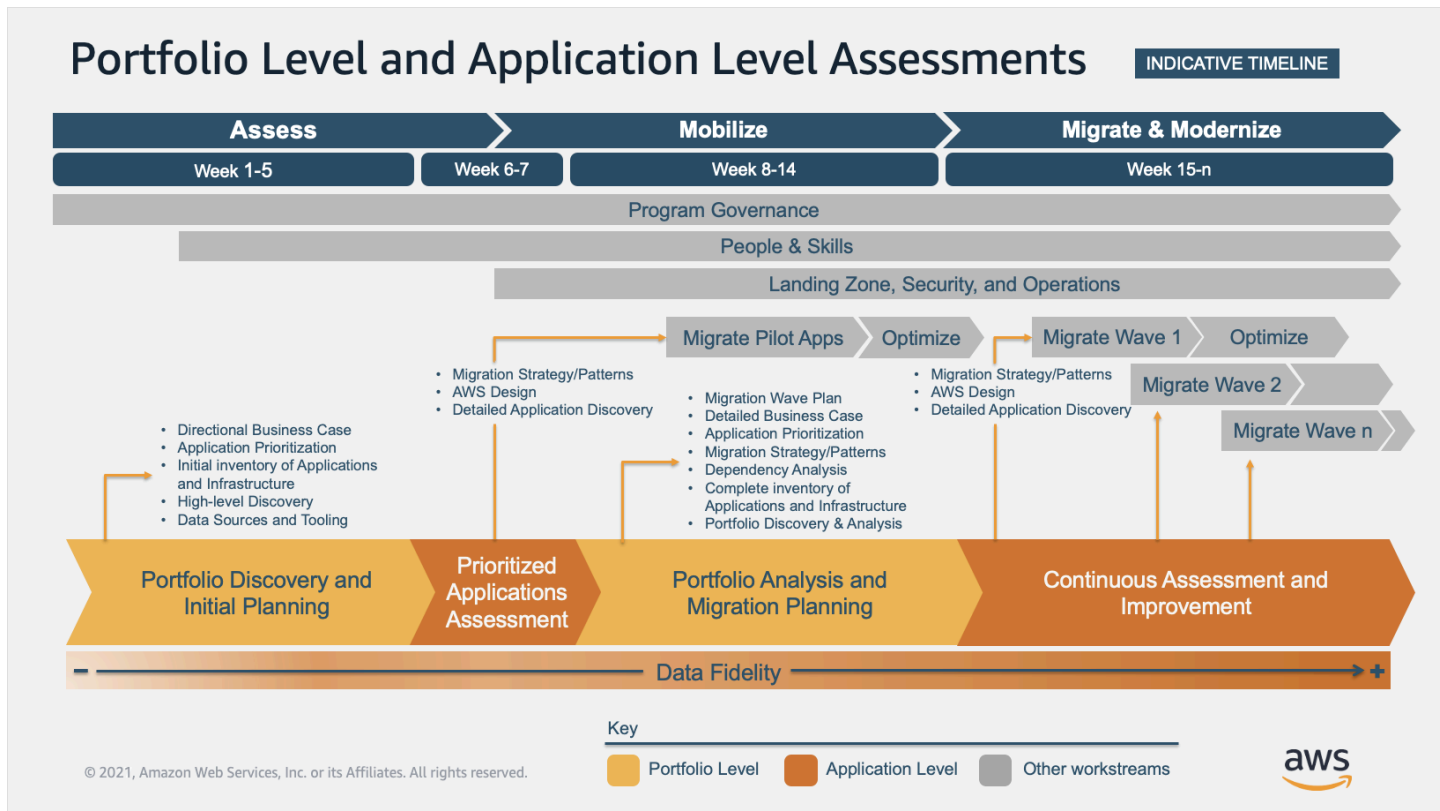
長期にわたるクラウド移行プログラムでは、プログラムガバナンス、landing zone (セキュリティ制御を備えた運用対象環境)、移行、アプリケーションポートフォリオなど、いくつかのワークストリームの調整が必要です。これらのワークストリームの名前は、移行プログラムの構成方法によって異なる場合があります。ワークストリームとして、アプリケーションポートフォリオ評価は、これらのプログラムのライフサイクル全体にわたる基本的なアクティビティです。評価を通じて得られるポートフォリオの理解は、継続的なアプリケーションポートフォリオ評価から得られるデータや分析に依存する他のワークストリームへの重要なインプットとなります。

次の図は、AWSポートフォリオ評価の段階が移行の段階にどのように対応するかを示しています。ポートフォリオの発見と初期計画の段階は、通常は最初の5週間に評価フェーズから始まります。優先順位付けされたアプリケーション評価は、6週目と7週目に行われ、評価フェーズとモビリティフェーズにまたがって行われます。ポートフォリオ分析と移行計画の段階は、8~14週目のモビライズ・フェーズで行われます。継続的な評価と改善段階は、15週目から移行プログラムの終了までの移行と最新化のフェーズで行われます。このタイムラインは目安です。ステージの実際の所要時間は、プログラム全体の構成によって異なります。ポートフォリオ評価段階は、このフレームワーク以外でも有効であり、どの移行プログラム構造にも組み込むことができます。



- ポートフォリオの発見と初期計画は、ポートフォリオに関する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、移行の基本的な合理化モデルの確立、および初期の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、ターゲット状態のアーキテクチャの初期設計、time-to-value および短期的に移行できるアプリケーションの特定を通じて、より迅速に行われます。アプリケーションを迅速に移行することで、チームは移行の経験を積むことができ、初期landing zone 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、up-to-date アプリケーションポートフォリオの全体像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を繰り返し強化し、データギャップを埋め、ビジネスケースを進化させ、信頼性の高い移行ウェーブプランを作成することによって構築されます。
- 継続的な評価と改善は、移行の波ごとにアプリケーションとテクノロジーの詳細な評価を継続的なアクティビティとして作成することにより、大規模な移行をサポートします。この段階では、移行ウェーブプランを繰り返し検討し、移行したワークロードをさらに分析して最適化と最新化を図ります。

次の図は、評価の各段階における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変化するかを示しています。ポートフォリオレベルの評価は、ポートフォリオのハイレベルな発見と全体的な分析に焦点を当てています。たとえば、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、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を詳細に発見することに重点を置きます。たとえば、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、AWSターゲット設計、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、必要な情報の幅広さと深さを表します。



ポートフォリオの発見と初期計画

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のデータの取得と分析の初期段階に焦点を当てます。主な目的は、ビジネス推進要因を特定し、アプリケーションや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状態を把握することです。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およびビジネス的属性が含まれます。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、最初の移行候補を特定し、ビジネスケースを伝える上で重要です。

この段階の主な成果

- ビジネス推進要因、成果、目標、および技術的指針を文書化しました。
- 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を作成し、データギャップを特定しました。これはポートフォリオの初期図であり、今後の段階で繰り返し改良していく予定です。
- 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 (たとえば、3 ~ 5 つのアプリケーション)。
- 次のステップの定義

初期評価データ要件の理解

データ収集にはかなりの時間がかかり、必要なデータとそのタイミングが明確でないと、簡単に障害になります。重要なのは、この段階の結果を得るには、データが少なすぎることと多すぎることのバランスを理解することです。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てください。

データソースとデータ要件

最初のステップでは、データソースを特定します。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することから始めます。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、監視、サポートチームのメンバーと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ワーキング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入手してください。

これらの会話の指針となるように、次の質問を参考にしてください。

- 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、会社の構成管理データベース (CMDB) の場合、ギャップがどこにあるかはすでにわかっていますか？
- CMDB (または同等のもの) を最新の状態に保つためのアクティブなツールやプロセスはありますか？もしそうなら、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 application-to-infrastructure などの現在のインベントリには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資産はアプリケーションに関連していますか？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？
- インベントリには、各製品のライセンスとライセンス契約のカatalogが含まれていますか？
- インベントリには依存関係データが含まれていますか？サーバー間、アプリケーション間、アプリケーション間、アプリケーションまたはサーバー対データベースなどの通信データの存在に注意してください。
- 環境内でアプリケーションやインフラストラクチャの情報を提供できるツールは他にありますか？データソースとして使用できるパフォーマンス、監視、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、どのような場所がありますか？

これらの質問に回答したら、特定したデータソースをリストアップします。次に、それぞれに忠実度、つまり信頼度を割り当てます。ツールなどのアクティブなプログラマティックソースから最近 (30 日以内に) 検証されたデータは、忠実度が最も高くなります。静的データは忠実度が低く、信頼性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 CMDB、プログラマ的に管理されていないその他のデータセットや、最終更新日が 60 日以上前の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。前提条件とそれに伴うリスクに対する最大限の許容度という観点から組織の要件を評価し、どの程度の忠実度が適切かを判断することをお勧めします。この表では、組織的な知識とは、文書化されていないアプリケーションやインフラストラクチャに関するあらゆる情報を指します。

データソース	忠実度レベル	ポートフォリオカバーレッジ	コメント
制度上の知識	低-最大 25% の正確なデータ、75% の想定値、またはデータが	低	希少で、重要なアプリケーションに集中している

データソース	忠実度レベル	ポートフォリオカバレッジ	コメント
	150 日以上経過しています。		
ナレッジベース	中～低-正確データの 35～40%、推定値またはデータが 120～150 日前のものが 65～60% です。	中	手作業で管理され、詳細レベルに一貫性がない
CMDB	中程度-精度の高いデータの 50%、推定値の約 50%、またはデータは 90～120 日前のものです。	中	ソースが混在し、データにギャップがいくつかある
VMware vCenter のエクスポート	中～高-データの 75～80% が精度、25～20% の想定値、またはデータは 60～90 日前のものです。	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モニタリング	高-ほぼ正確なデータ。推定値の約 5%、またはデータは 0～60 日前のものです。	低	重要な生産システムに限定 (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、各資産クラス (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) の必須および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)、および評価のこの段階で推奨されるデータ忠実度を示しています。表には次の略語が使用されています。

- R (必要な場合)
- (D)、方向性のあるビジネスケースの場合、総所有コスト (TCO) の比較や方向性のあるビジネスケースに必要

- (F) (方向性のあるビジネスケースの場合はTCOの比較に、移行と近代化のコストを含む方向性のあるビジネスケースの場合は必須)
- O、オプション用
- 該当しない場合は該当なし

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。通常、既存のCMDBまたはその他の内部インベントリおよび制御システムで利用できます。組織でこれらが定義されていない場合は必ず、ユニークな ID を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を組織に知らせる名前。該当する場合は、商用 off-the-shelf (COTS) ベンダーと製品名を含めてください。	R	R (D)	やや高い

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	推奨忠実度レベル (最小)
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	商用ソフトウェアの製品名、バージョン	R	R (D)	中
説明	主なアプリケーション機能とコンテキスト	R	O	中
緊急性	たとえば、戦略的または収益を生み出すアプリケーション、または重要な機能のサポートなど	R	O	やや高い
タイプ	たとえば、データベース、顧客関係管理 (CRM)、Web アプリケーション、マルチメディア、IT 共有サービス	R	O	中

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	推奨忠実度レベル (最小)
環境	たとえば、プロダクション、プリプロダクション、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 (HIPAA、SOX、PCI-DSS、ISO、SOC、FedRAMP など) および規制要件に適用可能なフレームワーク	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資産や仮想資産へのマッピング	O	O	中

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	推奨忠実度レベル (最小)
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 (例:Microsoft SQL Server エンタープライズ)	O	R	やや高い
コスト	ソフトウェアライセンス、ソフトウェア運用、およびメンテナンスのコスト	該当なし	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	推奨忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID。通常、既存のCMDBまたはその他の内部インベントリおよび制御システムで利用できます。組織でこれらが定義されていない場合は必ず、ユニークな ID を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 (例:ホスト名)	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	内部および/またはパブリック IP アドレス	R	O	やや高い
アセットタイプ	物理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	商用ベンダーおよび製品名 (VMware ESXi、IBM パワーシステムズ、Exadata など)	R	R	中
オペレーティングシステム	たとえば、REHL 8、Windows Server 2019、AIX 6.1	R	R	やや高い

構成	割り当てられた CPU、コア数、コアあたりのスレッド数、メモリの合計、ストレージ、ネットワークカード	R	R	やや高い
使用率	CPU やメモリ、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	O	やや高い
ライセンス	商品ライセンスの種類 (例:RHEL 標準)	R	R	中
インフラストラクチャは共有されていますか?	「はい」または「いいえ」は、認証プロバイダー、監視システム、バックアップサービス、および同様のサービスなどの共有サービスを提供するインフラストラクチャサービスを示します	R	R (D)	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	O	O	中
コスト	ハードウェア、メンテナンス、運用、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、ラックスペースのシェア、データセンターの諸経費など、ベアメタルサーバの全額負担コスト	該当なし	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	推奨忠実度レベル (最小)
パイプのサイズ (MB/s)、冗長性 (Y/N)	現在の WAN リンク仕様 (1000 MB/秒の冗長構成など)	O	R	中
リンク使用量	ピーク時および平均使用率、アウトバウンド	O	R	中

	データ転送 (GB/月)			
レイテンシー (ミリ秒)	接続された場所間の現在の待ち時間。	○	○	中
コスト	1 か月あたりの現在のコスト	該当なし	○	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	推奨忠実度レベル (最小)
リホスト	各作業負荷に対する顧客とパートナーの労力 (1 日あたり)、顧客とパートナーの 1 日あたりのコスト率、ツールコスト、ワークロード数	該当なし	R (F)	やや高い
プラットフォーム変更	各作業負荷に対する顧客とパートナーの労力 (1 人あたりの日数)、顧客とパートナーの 1 日あたりのコスト率、作業負荷数	該当なし	R (F)	やや高い
リファクタリング	各作業負荷に対する顧客とパートナーの労力	該当なし	○	やや高い

	(1人あたりの日数)、顧客とパートナーの1日あたりのコスト率、作業負荷数			
リタイア	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	既存の再利用 (Y/N)、AWS必要なリージョンのリスト、コスト	該当なし	R (F)	やや高い
人と変化	クラウドの運用と開発についてトレーニングするスタッフの数、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
[Duration] (所要時間)	対象ワークロード移行の期間 (月)	O	R (F)	やや高い
並列コスト	移行中に現状の費用を削減できる期間と速度	該当なし	O	やや高い

AWS移行中に製品やサービス、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期間と速度	該当なし	○	やや高い
--	------	---	------

ディスカバリーツールの必要性の評価

あなたの組織にはディスカバリーツールが必要ですか？ポートフォリオ評価には、up-to-date 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。

しかし、進歩が進むにつれて、忠実度の高いデータにより、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確に見積もることができるため、コスト削減とメリットの最大化が可能になります。また、依存関係を考慮した実装が可能になり、移行の落とし穴がなくなるため、リスクが軽減されます。クラウド移行プログラムにおけるディスカバリーツールの主な用途は、以下の方法でリスクを軽減し、データに対する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
- データ取得速度の向上、プロジェクトのスピードの向上、コストの削減
- 通信データやCMDBでは通常入手できない依存関係など、データの完全性レベルの向上
- アプリケーションの自動識別、TCO 分析、予測稼働率、最適化に関する推奨事項などの知見の取得
- 信頼性の高い移住ウェブプランニング

特定の場所にシステムが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て、ping または SNMP (簡易ネットワーク管理プロトコル) 要求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構成で ping または SNMP トラフィックが許可されるわけではないことに注意してください。これらのオプションについては、ネットワークチームや技術チームと話し合ってください。

アプリケーションポートフォリオの評価と移行の次の段階では、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により、必要となるインフラストラクチャと設定AWS (セキュリティグループ、インスタンスタイプ、アカウント配置、ネットワークルーティングなど) を理

解できます。また、同時に移動する必要があるアプリケーション (低遅延ネットワークを介して通信する必要があるアプリケーションなど) をグループ化するのにも役立ちます。さらに、依存関係マッピングは、ビジネスケースを進化させるための情報を提供します。

検出ツールを決定する際には、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。データギャップは障害になる可能性があるため、future データ要件とデータソースを分析してギャップを予測することが重要です。現場での経験から、行き詰まっている移行プロジェクトのほとんどはデータセットが限られており、対象とするアプリケーション、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特定されていないことがわかります。このような識別情報の欠如は、誤った指標、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第一歩です。

ディスカバリーツールを選ぶには？

市場に出回っているいくつかの検出ツールは、さまざまな機能を備えています。要件を検討してください。そして、あなたの組織に最も適したオプションを決めてください。移行のための検出ツールを決定する際によく見られる要素は次のとおりです。

セキュリティ

- 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？
- データにアクセスできるのは誰ですか？また、ツールにアクセスするためのセキュリティ管理はどのようなものですか？
- ツールはどのようにデータを収集しますか？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データはどのように転送されますか？
- このツールは、保管時と転送中のデータの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どのようなものですか？

セキュリティチームがディスカバリーツールに関する早期の話し合いに参加するようにしてください。

データ主権

- データはどこに保存され、処理されますか？
- このツールは、SaaS (SaaS) モデルを使用していますか？

- すべてのデータを自分の環境の境界内に保持することは可能ですか？
- データが組織の境界外に出る前にスクリーニングすることはできますか？

データレジデンシー要件の観点から組織のニーズを検討してください。

アーキテクチャ

- どのようなインフラストラクチャが必要で、どのようなコンポーネントが異なりますか？
- 複数のアーキテクチャが利用できますか？
- 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取り付けをサポートしていますか？

パフォーマンス

- データ収集はシステムにどのような影響を与えますか？

互換性と適用範囲

- 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？ ツールのドキュメントを確認して、サポートされているプラットフォームが対象範囲に関する現在の情報と照合されていることを確認してください。
- 使用している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？ 使用している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んでください。

収集方法

- このツールは、対象となる各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？
- エージェントレス展開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？
- 収集プロセスはどのようなものですか？

特徴

- どのような機能がありますか？

- 総保有コスト (TCO)AWS とクラウドの推定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？
- AWSターゲット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は実行されますか？
- どのレベルの依存関係マッピングが可能ですか？

アプリケーションやインフラストラクチャの依存関係マッピング機能が強いツールや、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。

費用

- ライセンスモデルとは
- ライセンスにかかるコストはどのくらいですか？
- サーバごとの価格設定ですか？ 段階的な価格設定ですか？
- オンデマンドでライセンスできる、機能が制限さ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を通じて使用されます。予算が限られている場合は、少なくとも6か月は検討してください。ただし、検出ツールがないと、通常、手作業と内部コストが高くなります。

Support 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ていますか？
- 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のくらいですか？

プロフェッショナルサービス

- ベンダーはディスカバリーアウトプットを分析するための専門サービスを提供していますか？
- このガイドの要素を網羅できますか？
- ツーリング+サービスの割引やバンドルはありますか？

ディスカバリーツールの推奨機能

時間が経つにつれて複数のツールからのデータをプロビジョニングして結合することを避けるため、検出ツールには以下の最低限の機能しか備えていない必要があります。

- 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ているソフトウェアを識別できる必要があります。
- 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作成できる必要があります。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できる必要があります。
- プロファイルと構成の検出 — CPU ファミリ (x86、PowerPC など)、CPU コアの数、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報告できるはずです。
- ネットワークストレージの検出 — 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。
- パフォーマンス — CPU やメモリ、ディスク、ネットワークの最大使用率と平均使用率をレポートできます。
- ギャップ分析 — データの量と正確性に関する洞察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て、未知のインフラストラクチャ資産を発見できるはずです。
- 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。

検討すべきその他の機能

- TCO 分析により、AWS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。
- リホストおよびリプラットフォームのシナリオにおけるMicrosoft SQL ServerおよびOracleシステムのライセンス分析と最適化に関する推奨事項。
- 移行戦略の推奨事項 (ディスクバリーツールは、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)
- 適切なサイズの推奨事項 (たとえば、AWS推奨されるターゲットインフラストラクチャをマッピングできるかどうか)
- 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?)
- 建築図 (建築図を自動作成できるかなど)
- API アクセス (たとえば、プログラムからアクセスして CMDB のデータを更新できますか?)

- アプリケーションの優先順位付け (アプリケーションやインフラストラクチャの属性に重み付けや関連性を割り当てて、移行の優先順位付け基準を作成できますか?)
- ウェーブプランニング (推奨アプリケーショングループや移行ウェーブプランの作成機能など)
- 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

発見ツールを選択して調達したら、次の質問を検討して、そのツールを組織に導入する担当チームと話し合ってください。

- サーバーやアプリケーションは第三者によって運営されていますか? これにより、関与するチームと従うべきプロセスが決まる可能性があります。
- ディスカバリーツール導入の承認を得るための大まかなプロセスはどのようなものですか?
- 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どのようなものですか? サーバーの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手順を教えてください。システム (コンテナ、仮想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) からデータを収集するには、認証情報が必要です。各資産に接続するための検出ツールの認証情報を取得することは、特にこれらの資産が一元化されていない場合には困難な場合があります。
- ネットワークセキュリティゾーンの概要を教えてください。ネットワーク図は入手できますか?
- データセンターのファイアウォールルールをリクエストするプロセスを教えてください。
- データセンターの運用 (検出ツールのインストール、ファイアウォールの要求) に関する現在のサポートサービスレベル契約 (SLA) はどのようなものですか?

ビジネス推進要因と技術的指針

ビジネスドライバー

組織がクラウドへの移行をすでに決定しているか、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ことで、移行の理由が明確になります。理由を文書化したら、何を移行し、どのように移行するかを定義できます。このアクティビティは重要です。次のステップを伝え、導くために、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。

議論に参加すべき利害関係者を特定し、推進要因を文書化します。一般的には CxOs、組織内の上級管理職、主要なテクノロジーリーダー、および自社の顧客です。お客様がこの議論に参加する可能性

は低いですが、お客様の意見や目標を代表する人物を組織内に 1 人以上指名することをお勧めします。

ビジネス推進要因は、移行プロセス全体を通して測定できる指標と結び付けて、成果が達成されたかどうかを検証する必要があります。会社の戦略目標と年次報告書が出発点となります。

クラウドへの移行の結果として、既存の指標と予測される指標に基づいて、会社が目指す方向性に焦点を当てて会話を進めます。目標とビジネス成果を検討してください。また、クラウドの採用が増えるにつれ、どのような成功を収めるかを考えてみましょう。

次に、各ドライバーの重要度を設定します。優先事項は何か？期待されるメリットはどのようなものですか？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という観点では、回答が移行するワークロードの優先順位付けや技術的な指針の確立に役立ちます。ただし、ビジネス推進要因が移行プログラム全体を定義し、影響を与えます。

技術指導原則

ポートフォリオ評価の後期段階における移行戦略の選択には、技術的な指針となる原則があります。現段階では、それらを特定することに重点が置かれています。

指針となる原則は、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。

たとえば、ある企業がコスト削減を第一の目標としており、望ましい結果はオンプレミスのデータセンターを 6 ~ 12 か月後に特定の日付までに閉鎖することです。そこから導かれる指針は、可能な限り再ホストまたは再配置の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移行することです。この場合、lift-and-shift このアプローチは短期的な移住の成果を加速させます。アプリケーションをオンプレミスのデータセンターから移動した後は、移行したワークロードを最適化または最新化するための主要なビジネス推進要因に集中できます。

技術的な指針を確立するには、まずビジネス推進要因を分析することから始めます。ビジネス目標と成果を達成するためのテクノロジーとテクニックのリストを挙げてください。次に、希望する結果が得られるように、リストを絞り込み、適合性や優先度に基づいて関連性の高い順序を割り当てます。

指針となる原則を文書化し、移行の計画と実施に関わる人々に伝えます。原則と実際の実施との間の懸念事項や潜在的な矛盾を浮き彫りにする。

次の表に、ビジネス推進要因と技術的指針の原則を示します。

ビジネスドライバー	結果	メトリクス	技術指導原則
イノベーションを加速します。	競争力の向上、ビジネスアジリティの向上	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る新機能、顧客満足度スコア、実験回数	DevOps マイクロサービスと運用モデルを使用して差別化を図るアプリケーションをリファクタリングし、新機能の市場投入までの俊敏性とスピードを高めます。
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。	需要と供給が一致した柔軟なコストベース (使用した分だけ支払う)	経時的な支出の変動	<ol style="list-style-type: none"> 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再ホストします。 2. 使用率が低い、またはまったくないアプリケーションは廃棄してください。
運用上の回復力を高めます。	稼働時間の向上、平均復旧時間の短縮	SLA、インシデント数	<ol style="list-style-type: none"> 1. アプリケーションを最新かつ最もサポートされ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。
データセンターを出ます。	6 ~ 12 か月以内の日付までのデータセンターの閉鎖	サーバー移行のスピード	1. クラウド移行ファクトリソリューションを使用してアプリ

ビジネスドライバー	結果	メトリクス	技術指導原則
オンプレミスにとどまって、俊敏性と回復力を高めましょう。	オンプレミスを維持しながら競争力と稼働時間を向上	1 日または 1 か月あたりの導入数、四半期ごとの新機能リリース、SLA、インシデント数	<p>ケーションを再ホストします。</p> <p>2. 上の VMware クラウドを使用して再配置しますAWS。</p> <p>1. 機能をクラウドに拡張してシステムを最新化します。</p> <p>2. AWSOutposts への再ホストまたはリプラットフォームを検討してください。</p>

データ収集の開始

データ収集は、アプリケーションや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し行われます。各段階で、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の収集に重点が置かれます。このインベントリは、方向性のあるビジネスケースの作成と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レビューと、アプリケーションやサービスに関する組織の知識を更新するのに役立つという利点があります。また、何を移動させるかを判断するのにも役立ちます。推奨されるアプローチは、構成管理データベース (CMDB) の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、データ収集の対象となる資産のリストを作成します。移行の対象と対象外の内容が組織で完全にわかっている場合は、データ収集の対象となるシステムに限定することもできます。

ポートフォリオを構築する際には、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。たとえば、カスタマー・リレーションシップ・マネジメント (CRM) アプリケーションを特定してそのアプリケーションにテスト、開発、および本番環境がある

こと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) を挙げてください。または、CRM 名を使用して環境ごとに固有の 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することもできます。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、非実稼働の環境を最初に移行できます。環境に応じ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制御できます。

データ収集、特定のデータセンターまたはソースの場所にあるアプリケーションまたはサーバーが不明な場合があります。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、初期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、[この段階のデータ要件と既存のソースから何が得られるかという観点からギャップ分析を行うことです](#)。完全性のパーセンテージとデータの忠実度を対比させることが重要です。忠実度の低いソースから得られる完成度が高いと、分析に欠陥が生じ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限のデータ忠実度は求められませんが、データソースは少なくとも中～中程度の忠実度であることが推奨されます。これらの数値を、データギャップを埋めるための前提条件の使用など、組織のリスクに対する許容度と比較してください。

ギャップ分析は、処理しているデータの量と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、移行するアプリケーションの優先順位を決定するために必要となる前提条件のレベルを確立するのにも役立ちます。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、移行の成果を早めるために、できるだけ早く検出ツールを導入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ら数か月かかる場合があるため、早期に行動することも重要です。

この段階で、コミュニケーション計画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、利害関係者に常に情報を提供して、事前に計画を立て、リスクを軽減することができます。明確なコミュニケーションのための重要な要素は、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唯一の情報源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、インフラストラクチャのリストを保持することは避けてください。バージョン管理とオンラインコラボレーションをサポートする 1 か所 (データベース、ツール、スプレッドシートなど) にデータを保管し、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、優先順位付けの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、反復的かつ漸進的なアプローチをとって優先順位付けモデルを進化させることです。

アプリケーションの優先順位付け

評価のこの段階では、リスクが低く複雑度の低いワークロードを優先するための初期基準を確立することに重点を置いています。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。初期移行でリスクが低く複雑性の低いワークロードを使用すると、リスクが軽減され、チームに経験を積む機会が与えられます。これらの基準は、移行ウェーブプランを作成する際には、ビジネス推進要因に合わせて優先順位を合わせるために、今後の評価段階でさらに発展させていく予定です。

最初の基準では、依存関係が少ないアプリケーション、クラウド対応のインフラストラクチャで実行されているアプリケーション、および非本番環境から実行されるアプリケーションを優先する必要があります。例としては、依存関係が 0 ~ 3 のアプリケーションを、開発環境またはテスト環境でそのまま再ホストできる場合が挙げられます。これらの基準は、クラウド導入の成熟度や信頼度にもよりますが、パイロットアプリケーションの定義や、場合によっては第 1、第 2 の移行の波を定義するうえで有効です。

どの初期基準を使用するか 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、アプリケーションおよびインフラストラクチャの初期インベントリから取得されます ([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、設定可能な値がプロダクション、開発、テストの場合、各値にはスコアが割り当てられ、数字が大きいほど優先度が高くなります。オプションですが、各データポイントの重要度や関連性を倍率で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、より重要な点を強調するためのより高いレベルの差別化要因となり、値へのスコアの割り当てを繰り返し行う際に基準を一致させるのに役立ちます。

移行の最初の数段階では、リスクの低いシンプルなアプリケーションを優先させるという戦略に基づいて、次の表は属性選択と値の割り当ての例を示しています。

属性 (データポイント)	設定可能な値	スコア (0-99)	重要度または関連性の倍率
環境	テスト	60	ハイ (1倍)
	開発	40	
	本番稼働用	20	

属性 (データポイント)	設定可能な値	スコア (0-99)	重要度または関連性の倍率
ビジネス重要度	低	60	ハイ (1倍)
	中	40	
	高	20	
規制またはコンプライアンスの枠組み	なし	60	ハイ (1倍)
	FedRAMP	10	
オペレーティング・システムサポート	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	
コンピューティンスタンスの数	1 ~ 3	60	中～高 (0.8x)
	4-10	40	
	11 個以上	20	
マイグレーション戦略	リホスト	70	ミディアム (0.6x)
	プラットフォーム変更	30	
	リファクタリング、または再設計	10	

アプリケーション間の主な差別化要因となる属性を選択してください。そうしないと、この基準によって多くのワークロードが同じ優先度を共有することになります。モデルを適用したら、結果のランキングの上位と下を見て、同意できるかどうかを確認することをお勧めします。おおむね同意できない場合は、作業負荷のスコアリングに使用した基準を再検討してください。

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違いです。たとえば、上位の合計スコアが 8,000 で、下位スコアが 800 である場合があります。結果のスコアをヒストグラムとしてプロットすることを検討してください。そうすれば、分布が良好であることを確認できます。理想的な分散は、優先順位が非常に高いワー

クロードがいくつかあり、優先順位が非常に低いワークロードがいくつかあるという、標準的なベルカーブのように見えます。アプリケーションの大半は中間のどこかにあります。

最初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に関心を示す社内チームまたはビジネスユニットを参加させることです。これらは、特に初期の段階では、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得る上で大きな手段となる可能性があります。これが組織に当てはまる場合は、前の表にビジネスユニット属性を含めてください。アプリケーションを積極的に提案してくれるビジネスユニットに高得点を割り当てます。ビジネスユニット属性を使用すると、それらのアプリケーションをリストの一番上に表示しやすくなります。

結果のランキングに同意したら、上位5〜10件のアプリケーションを選択します。これらが最初のアプリケーション移行候補になります。3〜5件の申請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施する際に、的を絞ったアプローチをとることができま

す。詳細については、「[優先アプリケーション評価](#)」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と関連インフラストラクチャの移行戦略を決定することは、移行のスピード、コスト、およびメリットのレベルに影響します。ビジネス推進要因、技術的指針、優先順位付け基準、ビジネス戦略など、さまざまな要因をバランスよく組み合わせて戦略を決定することが重要です。

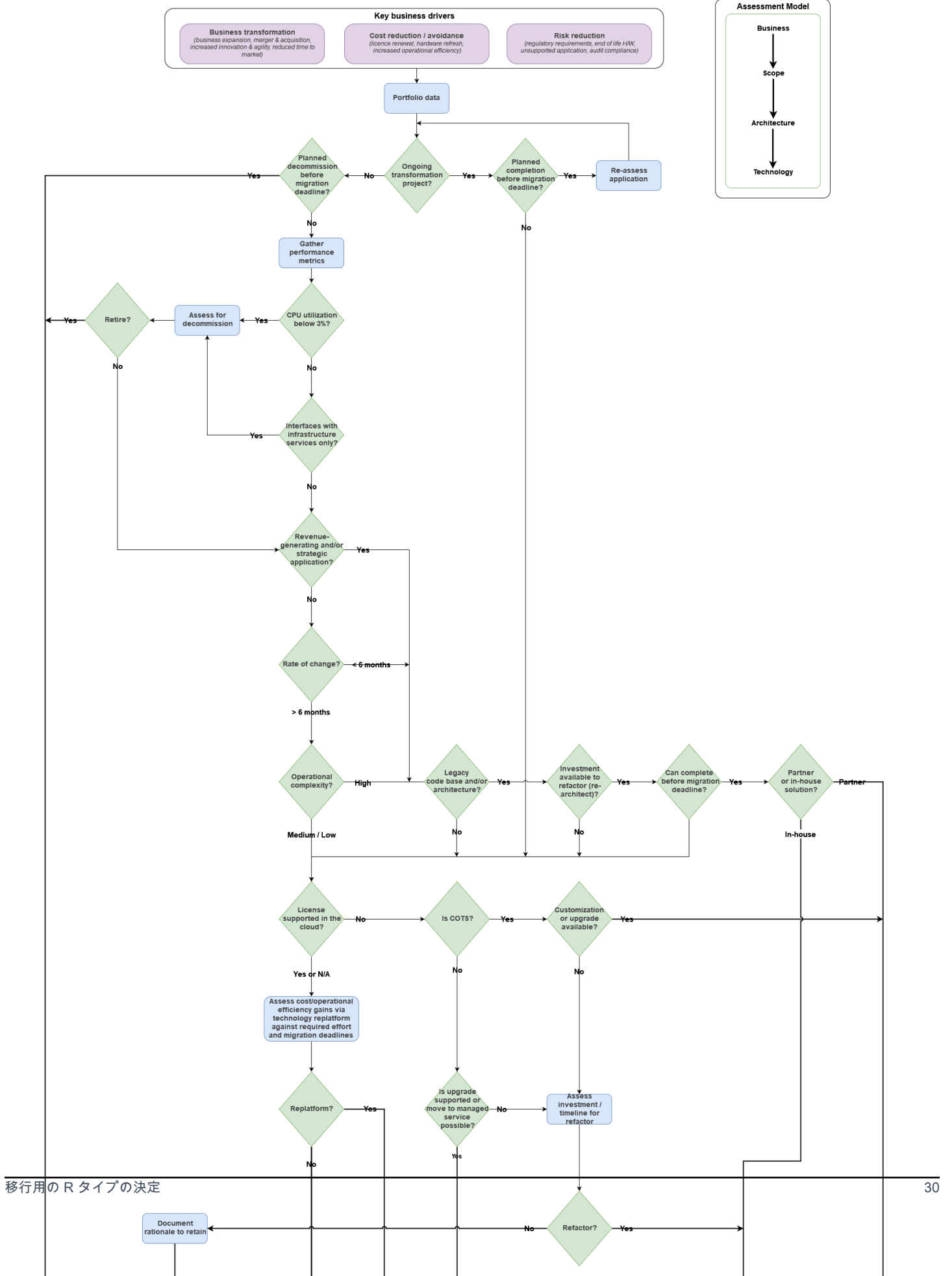
これらの要因が相反する見解を生み出すことがあります。たとえば、移行の主な推進力はイノベーションとアジリティかもしれませんが、同時に、コストを迅速に削減する必要があるかもしれません。対象範囲内のすべてのアプリケーションをモダナイズすることで、長期的にはコストを削減できますが、事前に多額の投資が必要になります。その場合の1つの方法は、再ホストやリプラットフォームなど、手間がかからない戦略を使用してアプリケーションを移行することです。これにより、短期間で迅速な効率化とコスト削減が可能になります。そして、節約した分を後の段階でアプリケーションの最新化に再投資し、さらなるコスト削減を実現します。

ただし、すべてのアプリケーションを完全に再ホストすることから始めると、モダナイゼーションの大きなメリットが遅れます。重要なのは、移行戦略のバランスをとることです。そうすることで、ビジネス戦略的なアプリケーションをモダナイゼーションの対象とし、他のアプリケーションを最初に再ホストまたは再プラットフォーム化してからモダナイズできるようにすることです。

アプリケーションの移行戦略を決定する方法は？

評価のこの段階では、移行戦略の選択を導くための初期モデルを組み込むことに重点が置かれます。初期アプリケーションの移行戦略を検証するには、モデルをビジネス推進要因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再設計などの最も複雑なアプローチは、戦略的ワークロード専用です。



[この図のカスタマイズ可能なdraw.ioバージョンは、添付ファイルセクションにあります。](#)

初期モデルの最初のステップは、ツリーの最上部にあるビジネスドライバーを、組織で定義されているもので更新することです。次に、ツリーを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、アプリケーション層、データベース）で構成される3層アプリケーションの場合、各コンポーネントは独立してツリーを移動し、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、特定の階層を再ホストまたはリプラットフォームして、他の層をリファクタリング（再構築）したい場合があるためです。

コンポーネントを個別に割り当てることで、関連するインフラストラクチャの移行戦略を定義できるようになります。インフラストラクチャ戦略は、サポートするアプリケーションコンポーネントと同じ戦略である場合もあれば、異なる場合もあります。たとえば、新しいオペレーティングシステムを搭載した新しい仮想マシンに再プラットフォームされるアプリケーションコンポーネントは、その再プラットフォーム戦略に従い、そのコンポーネントをホストしている現在の仮想マシンは廃止されます。インフラストラクチャの移行戦略は、アプリケーションコンポーネント用に選択された戦略に基づいて計算されます。

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、結果に概ね同意できるかどうかを確認してください。7R デシジョンツリーは、その正しさを判断するために必要な分析に代わるものではありません。ツリーロジックは特定のケースには適用されない場合があります。このようなケースは例外として扱い、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することで、ツリーによる決定をオーバーライドしてください。これにより、管理が困難になる可能性のある複数のデシジョンツリーバージョンを防ぐことができます。一般的なガイドンスでは、ツリーは少なくとも70～80%のワークロードで有効である必要があります。それ以外には例外があります。評価のこの段階でツリーロジックを調整する場合は、初期モデルの確立に重点を置く必要があります。[ポートフォリオ分析や移行計画など、後の段階でさらなる反復と改良が行われる予定です。](#)

添付ファイル

[attachment.zip](#)

方向性のあるビジネスケースの作成

ビジネス全体の利害関係者は、その過程の各段階で変革のビジネスケースを理解し、賛同する必要があります。

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得られる潜在的な価値をすぐに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実現できるという妥当な確信が得られるように設計されています。

プログラムが確立されると、ビジネスケースはさらに発展します。ケースを詳細に説明することで、プログラムの価値をより正確に把握し、計画の優先順位を把握できます。これにより、組織が賛同するビジネス上の成果の計画が定義および定量化され、プログラムガバナンスオフィスがプログラムの運営や成果の測定基準となるベースラインが設定されます。

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。コアチームを結成するためのリソースを確保し、AWS 必要に応じてパートナーと契約し、最低限、[優先順位の高いアプリケーション評価とポートフォリオ分析、移行計画の段階を完了できるように](#)、十分な信頼を得る必要があります。

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。

- AWS 現状のインフラストラクチャ環境と移行後のサービスアーキテクチャの簡単な総所有コスト (TCO) の比較です。比較すると、特定のワークロードボリュームでの予想実行レートの違いがわかります。
- 正味現在価値 (NPV)、投資収益率 (ROI)、投資回収期間、修正内部収益率 (MIRR)、3 ~ 5 年のキャッシュフロー分析を記載したビジネスケースです。AWS 移行費用込みへの移行と、現状のまま移行した場合の比較です。

方向性のあるビジネスケースの範囲は、通常、次のいずれかに限定されます。

- インフラ技術コストの比較
- インフラストラクチャ、テクノロジー、運用コストの比較

一般に、ポートフォリオが大きいほど、ケースの開発は少なく済みます。これは、結果に大きな影響を与えることなく、より幅広い仮定を行うことができるためです。ポートフォリオが小さい場合は、どんな変更も大きな影響をもたらすため、より詳細な情報が必要です。

まず、基本となるインフラストラクチャのコスト比較を作成します。次に、比較が十分に説得力があるかどうかを判断してから、次に進みます。通常、400 台を超えるサーバのポートフォリオでは、運用から 3 年以内、または 5 年以内に 250 台のサーバがインフラストラクチャコスト削減だけでプ

ラスのビジネスケースを示しますが、この数字はさまざまです。AWSポートフォリオが小さい場合は、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台のサーバー未満でない限り、回復力やビジネスアジリティの向上から得られる価値など、この段階で他のビジネス価値要素を検討することはほとんど役に立ちません。

フォーカス・バリュードライバー

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、AWS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サービス部品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、次のリストに注目します。これらのリストは評価が容易で、一般的に約 30% の TCO 削減が可能で、先に進むには十分だからです。

- コンピューティングの柔軟性 — 8x5 (使用率が 24%)、10x5 (30%)、10x6 (36%) で稼働する開発サーバーや UAT サーバー、2% で稼働するディザスタリカバリ (DR) サーバーなど、使用率が 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。
- 節約プランによる調達 — 運用サーバや使用率が高い (36% 以上) 他のサーバを、適切な節約プランで調達し、コストを最大 75% 削減することを計画します。オプションには、1 年契約と 3 年契約があり、前払いのレベルが異なり、大幅な割引が受けられます。
- ゾンビの排除 — 不要になったことが確認できる CPU 使用率が 2% 未満のサーバーを特定し、コスト分析から除外します。
- コンピューティングの適切なサイジング — CPU とメモリの使用率の時系列データを使用して、各サーバに必要な計算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ジング — 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を設定した後に、RDBMS ライセンスのニーズを再評価し、私物ライセンスの持ち込み (BYOL) とライセンス調達元を比較して、Amazon Relational Database Service (Amazon RDS) のコスト削減の可能性を探ります。AWS
- ストレージ — 必要な総ストレージ容量を適切なサイズに設定し、ポートフォリオ全体の 1 秒あたりの入出力操作数 (IOPS) のニーズを特定します。さまざまなSLAとコストで、オブジェクトストレージにどれだけ移動できるかを判断してください。

データニース

「[初期評価データ要件の理解](#)」の表には、方向性のあるビジネスケースの各部分を構築するために必要なデータと、それが必須かオプションかが示されています。

ケースを構築するには、初期計画データのインフラストラクチャサブセットとコストデータが必要です。含めるインフラストラクチャをどのように特定するかは、ビジネス目標によって異なります。

- プログラムの目的が特定のアプリケーションの移行と近代化である場合は、共有されてい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。
- リースの期限が切れるデータセンターからの移行など、プログラムの目的がインフラストラクチャ中心の場合、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、標準の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 パートナーまたはプロフェッショナルサービスと話し合うことができます。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、これはより正確になります。

ビルディングインフラストラクチャの TCO 比較

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パートナー](#)は、あらゆる種類の方向性のあるケースについて支援を提供できます。特に、より広範な移行プロセスを支援するために彼らを雇うことを計画している場合はそうです。

次のことを行うためのツールがあります。

- 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。
- 正確な「現状のまま」のインフラストラクチャコストベンチマークデータを提供します。
- ゾンビを特定して削除します。
- 適切なサイズの評価を行います。
- 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てください。
- シンプルなグラフィカルなキャッシュフロー分析を行います。

[AWS移行エバリュエーターからの移行も](#) 1つの選択肢です。これらの機能をすべて無料のマネージドサービスとして提供します。Migration Evaluator をリクエストするには、AWSアカウントマネージャーまたは移行コンピテンシーパートナーを通じて、[またはオンラインでリクエストを送信してください](#)。Migration Evaluatorは、インフラストラクチャテクノロジーのTCO比較を迅速に行う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
- エージェントレスでの検出または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手動設定
- 導入、設定、データ収集、ベースケースまたは方向性のあるビジネスケースの作成を支援する専用サポート
- SaaS運用は便利だが、分析エンジンに読み込む前のスクラビングをサポートするために、データ収集を顧客ネットワーク内で完全に実行できる
- Microsoft ライセンスの適切なサイズ設定への強力なサポート
- 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 (Windows および Linux) のみを評価します
- ベンチマークの現状のコストデータを設定または調整するためのオプションが限られている
- モデリング運用のコスト最適化はサポートされていない
- 移行コストモデリングはサポートされていません
- TCO 比較以外のビジネスケースの作成を直接サポートできない

アプリケーションスタックや相互依存関係の発見など、ポートフォリオの検出や分析機能に商用検出ツールを使用する場合は、通常、インフラストラクチャの TCO 比較も行えます。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ドンスについては、「[検出ツールの必要性の評価](#)」を参照してください。

運用コストの最適化を組み込む

IT運用の生産性の向上は、多くの場合、移行の価値に大きく貢献します。International Data Corporation (IDC) のホワイトペーパー「[Amazon Web Services 活用してビジネス価値を生み出すためのビジネスと組織の変革の促進](#)」によると、への移行後、移行によって IT 運用スタッフの生産性が平均で 62% 向上します。AWS だし、サイジングとこれらの利点をディレクショナルケースに含めることには、2 つの課題があります。

まず、生産性の向上を全面的に評価するには、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はより適切です](#)。この課題は、シンプルなベンチマークデータで簡単に評価およびサイジングできるものの、それでも大きな利点があるいくつかの要素に焦点を当てることで解決できます。

第二に、コスト削減の源泉として生産性に焦点を当てると、主要な顧客関係者やプログラムメンバーの間で懸念や否定的な意見が生じる可能性があります。そのメリットがどのように実現されるのか、またそれが影響を受ける人々にとってどのような意味を持つのかを明確にしてください。このような問題は、これによってチームの役割が強化され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、社内運用スタッフを育成して新しい役割に異動させるコースが含まれています。たとえば、DevSecOps チームに参加して、コード自動化やテスト自動化などの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などです。
- このメリットは、業務のアウトソーシング契約の範囲を変更してサイズを変更することで実現できます。これにより、社内スタッフはより価値の高い活動に集中できるようになります。

検討したい業務変革に基づいて、このビジネスケース要素を構築するアプローチは次のとおりです。

- すでに社内に運用チームがある場合は、チームメンバーのスキルを向上させ、期待される生産性の向上を示してください。
- または、AWS 現在の運用ソリューションから Managed Services (AMS) に移行するか、AWS パートナーが提供する別のマネージドサービスに移行してください。

最初の変革では、ケースに含めることができる生産性の向上を控えめに財務上の見積もりを得るには、次のことを推奨します。

1. 特にサーバー管理運用の生産性に重点を置いてください。運用作業の大部分を占める傾向があり、評価が容易で、後で検証しやすくなります。

- フルタイム相当 (FTE) の各従業員が管理できるサーバ数のベンチマークに基づいて、必要な人員数を計算します。オンプレミスでは、その数は約 150 台です。ほらAWS、約400台のサーバーがあります。
- EC2 インスタンスの数と比較して、これらのメトリックスをオンプレミスサーバーの数に適用します。
- 節約した時間に、運用チーム全体の混合コスト率を掛け合わせます。

次に、結果が、次の表に示されている役割別の平均生産性向上を大きく超えないことを確認して、どちらの方法でも結果を確認できます (IDC ホワイトペーパー「[Amazon Web Services によるビジネス価値創出のためのビジネスと組織の変革の促進](#)」からのデータ)。

[Role] (ロール)	効率性の向上
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つ目の変革では、対象ポートフォリオの現在の総運用コスト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、運用コスト削減額を追加できます。

マネージドサービスの費用を確認するには、AWS [AWS Managed Services アカウントマネージャー](#) または [パートナー](#) に、AWS 提案された部品表、選択したサービスレベル (プラスまたはプレミアム)、および AMS パッケージ (AMS Accelerate または AMS Advanced) を提出してください。これにより、AWS トランスフォーメーションされたソリューションのサービスコンポーネントのマネージドサービスの総コストが算出されます。同様に、AWS 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パートナーから価格を取得することもできます。

全面的なビジネスケースへの拡大

一般的に、全方位的なビジネスケースを組み立てるには、IT 生産性要素の有無にかかわらず、TCO の比較を行い、移行と近代化のコストをすべて見積もります。次に、migrate-and-modernize t-

migrate-and-modernize あるシナリオとそうでないシナリオの両方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、1組のシナリオを準備することです。「しない」 t-migrate-and-modernize シナリオは現在の状況であり、 migrate-and-modernize シナリオには次の特徴があります。

- トランザクション量、コンピューティング、ネットワーク容量の増加や縮小はありません
- ストレージ要件が少量で着実に増加している
- quality-of-service 既存のシステムの機能と一致するQ機能 (可用性、耐久性、スループット、パフォーマンスなど)

非常に小規模なポートフォリオを除くすべてのポートフォリオにとって、これは方向性のあるケースを構築するという目的に適しています。すぐにその価値が示され、前進する権限を獲得できます。

ポートフォリオが小さい場合は、 migrate-and-modernize t-migrate-and-modernize 次のようなクラウド移行の価値が高まる他の側面を示すシナリオとそうでないシナリオを2つ追加するとよいでしょう。

- 増加が予想されるワークロード全体で、中程度の容量増加要件と大容量の増加要件が混在している
- 高可用性、DR、耐障害性などの強化された耐障害性を組み込む
- 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により、グローバルパフォーマンスが向上しました。
- その他、プログラムにおいてビジネス上の優先事項として掲げた、サービス品質の向上に関する具体的事項

このようなシナリオでは、現在の非クラウド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見積もられていることを確認してください。この見積もりを得る最も直接的な方法は、システムインテグレーターに見積もりを依頼することです。特に、AWSシステムインテグレーターが移行資格を持つコンサルティングパートナーでもある場合は、 migrate-and-modernize t-migrate-and-modernize シナリオとそうでないシナリオの両方でサポートしてくれます。

シナリオのペアごとに、次の内容を含むケースを組み立てます。

- 「しない」 t-migrate-and-modernize シナリオのコスト 最も基本的なケースでは、これには以下が含まれます。
 - 現在のインフラストラクチャ構成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が定期的に増加する
- migrate-and-modernize; シナリオのコスト (以下を含む)
 - プログラムの設定 (詳細な検出、移行計画、詳細なビジネスケース開発、コアチームの設立とスキルアップ、まだ導入されていない場合はlanding zone 確立、移行したワークロードのセキュリティ管理と運用統合の確立を含む)
 - 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](#) などのデータ移行サービス[AWS DataSync](#)、AWS移行プロセス自体に必要なアーキテクチャの光熱費 (テストなど) などの移行インフラストラクチャコスト
 - AWS移行の過程で波が稼働するにつれて光熱費が上昇し、AWS既存のインフラがベースサービスに置き換えられて廃止されるにつれて増加する公共料金の増加

取り残された資産の廃止措置費用と償却

移行および近代化プログラムのセットアップの見積もり

プログラムを成功に導くには、一連の基礎的なアクティビティを実行してベースライン能力を構築し、詳細な計画を作成することが必要になる場合があります (これがまだ行われていない場合)。これらの基本的な活動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](#)」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行うほか、使用した検出ツールのコストを文書化します。
2. クラウドビジネスとテクニカルコアチームを設立し、トレーニングと採用を通じて社内スキルを開発します。トレーニングを必要とする 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. [landing zone](#) を設定し、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 の見積もりを提供できます。

移行と近代化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業的可能性を示すには、移行と近代化のコスト見積もりはできるだけ基本的なものにしてください。

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
- 再配置
- リホスト
- プラットフォーム変更
- 再購入

通常、ワークロードの約 70% は再ホスト、再配置、または再プラットフォームが可能で、残りの 5% は廃止できます。通常、移行戦略に基づいてアプリケーションを評価することが、コスト削減の要点となります。

リファクタリングや再構築のコストの見積もりは複雑な場合があります。方向性のあるビジネスケースを作成するために与えられた時間枠内でこれを試みることは現実的ではありません。前述の「[移行の R タイプの決定](#)」で説明したように、[移行とモダナイゼーションの最初のフェーズ](#)では、再ホスト、再配置、または再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、初期投資回収を加速し、導入リスクを軽減し、短期的には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にとって、AWS 環境内で実行されているアプリケーションをモダナイズする方が、そうでないアプリケーションよりも大幅に簡単です。特定のアプリケーションのリファクタリング（再設計）の見積もりは、[詳細なビジネスケースを準備するときに追加するのが最適です](#)。

戦略による移行作業の見積もり

移行はそれぞれ異なります。予算や計画を立てる前に、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、AWS パートナー組織など、プロジェクトを担当するチームから移行アクティビティの作業負荷を見積もってください。

方向性のあるケースを構築しやすくするために、次の表は、さまざまな治療法の目安となる範囲を示しています。これらの範囲は、medium-to-large 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、経験を積んでいることを前提としています。ポートフォリオが小さい場合は、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もりを準備してもらうのが最善です。

マイグレーション戦略	見積もりプロセス	[Elements] (要素)	個人勤務時間	個人勤務時間
Retain	Do nothing, with no cost, no	-	-	-

マイグレーション戦略	見積もりプロセス	[Elements] (要素)	個人勤務時間	個人勤務時間
	benefits, and no reduction in technology debt.			
Retire	Estimate decommissioning the hardware equipment used, if any.	—	—	—
Relocate	Estimate copying the workload within VMware using VMware tools. This includes copying the data, smoke testing to verify, and any hardware decommissioning. The effort to relocate VMs is typically less than for low-complexity rehost patterns.	—	—	—

マイグレーション戦略	見積もりプロセス	[Elements] (要素)	個人勤務時間	個人勤務時間
Rehost	Estimate copying the workload and data with an image copy, smoke testing, high availability (HA) and disaster recovery (DR) testing where appropriate for production servers, and any hardware decommissioning. The best practice is to use tools such as AWS Application Migration Service . Divide workloads into low, medium, and high complexity, based on factors such as whether a database or other infrastructure software is running, database complexity, whether	Effort per app per server Low Medium High	Migration 10–14 16–24 26–38	HA/DR test 3–5 4–6 8–12

マイグレーション戦略	見積もりプロセス clustered, integration complexity, and data volumes.	[Elements] (要素)	個人勤務時間	個人勤務時間
------------	---	-----------------	--------	--------

マイグレーション戦略	見積もりプロセス	[Elements] (要素)	個人勤務時間	個人勤務時間
Replatform	For replatform migration that include upgrades to operating system or RDBMS version, take the estimate for a rehost, and add time to run a rebuild and smoke test on the new platform. If the replatform includes changing the technology of the platform, estimate additional time for the use of the conversion tools, such as AWS Schema Conversion Tool and AWS Database Migration Service , and a more complete application test. An example of changing the	Effort per app per server Low Medium High	Version up Add 1–3 Add 2–5 Add 4–8	Technology change Add 10–15 Add 20–30 Add 40–60

マイグレーション戦略	見積もりプロセス	[Elements] (要素)	個人勤務時間	個人勤務時間
	technology is migrating away from a proprietary commercial database to an open source replacement.			
Repurchase	Estimate data extraction, transformation, and uploading into the newly purchased SaaS service replacement, and any hardware decommissioning.	–	–	–

移行インフラストラクチャコストの見積もり

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てください。通常、これらの推定値には次のものが含まれます。

- 現在の環境から、ワークロードとデータ移行のための接続およびデータ交換サービスの予算AWS
- 移行、テスト、AWSおよび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サービス (特にコンピューティングとストレージ) の予算
- AWS移行の波が押し寄せるたびに光熱費が上昇する
- 移行したワークロードを実行しなくなる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、総データ量を調べ、ネットワークを使用する可能性を評価してください。[AWS Direct Connect](#)[AWS VPN](#) AWS移行後の運用用にリンクまたはリンクを事前にプロビジョニングしておけば、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は、仮想プライベートネットワーク (VPN) を使用してインターネット帯域幅を短期間で増やすことが、費用対効果の高いソリューションであることがよくあります。そうでない場合は、[AWS Snowball](#) や [AWS Snowcone](#) [AWS](#) [AWS](#) などのメディア交換デバイスがほとんどの地域でソリューションを提供しています。また、大量のデータ移行の場合は、予算を含めることを検討してください。これにより[AWS DataSync](#)、信頼性が向上し、使用するメディアに関係なく転送を高速化できます。

AWSサービスの増加と既存のインフラの縮小をモデル化することは、ビジネスケースのキャッシュフロー分析要素にとって重要です。この段階では、コストが発生する時期を正確に決定するためのウェーブプランが策定されていない可能性があります。次の構成を推奨します。

- AWS移行中に一定の割合でコストが上昇する。
- 同じ期間に一定の割合で廃止を計画している既存のインフラストラクチャのコストを削減します。

AWS既存のインフラストラクチャが削減される1~2か月前にコスト増額を開始する。これにより、AWS各ウェーブの移行に必要なユーティリティを1か月分使用できます。これには、テストのための時間と、交換されたインフラストラクチャのコスト発生を防ぐために必要な廃止作業を完了するための追加の時間も含まれます。

廃止措置費用の見積もり

再配置できない機器を廃止し、合法的かつ環境に優しい方法で廃棄すると、少額の費用がかか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の場合、通常、重要な金額となる可能性があるのは、交換後の資産の残りの帳簿価額を償却するコストだけです。

方向性のあるビジネスケースでは、次の操作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てください。
- 廃止される予定のものを特定してください。
- 減価償却を減らすには、リストにある新しいデバイスを使用して、完全に減価償却された古い資産を置き換えることができるように、デバイスを切り替える機会を検討してください。
- その時点で廃止される予定の資産のfuture 帳簿価額を評価してください。
- これを廃止措置の移行コストとして含めてください。

全方位型ビジネスケースの組み立てと調整

各シナリオのコスト一式を準備したら、それぞれの割引キャッシュフロー計算書を作成してグラフ化します。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ケースを作成することをお勧めします。サーバ、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、各シナリオの現状のコストには、1回の更新のコストだけが含まれます。

次に、プログラムの次の段階に進むための承認を得るために必要な主要な財務指標を計算します。通常、次の内容が含まれます。

- 評価対象のコスト削減と生産性向上の絶対値を測定するための正味現在価値 (NPV)
- 返品が十分に迅速に行われたことを確認するための返済期間 (月単位)
- 最終実行率を比較して、そのプロセスで長期にわたって十分なコストが削減されているかどうかを確認する
- 投資収益率 (ROI) と修正投資収益率 (MIRR) は、組織が優先する可能性のある他の資本要求に対するプログラムの相対的な財務実績を評価するためのものです

次の例のように、ケースの最初の繰り返しを使用して、予想される財務実績から改善が必要かどうかを判断します。

- 投資回収が遅すぎる場合は、次のような移行を加速および削減するためのオプションを検討してください。
 - AWS/AWSパートナーやプロフェッショナルサービスを利用して、利用可能なリソースを増やし、移行するワークロードをより基本的なパターンでさらに並列化してください。
 - VMware で実行されているワークロードについては、少なくとも初期段階では、再配置戦略を再ホストまたは再プラットフォーム戦略と比較してください。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 (再構築) 戦略を必要とするワークロードを、当初のビジネスケースの範囲外のfuture 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？ 容量の増加と伸縮自在性に対するニーズが最も高いと思われるシナリオはありますか？ 目標の範囲内で、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 フェーズに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ものやディザスタリカバリ (DR) のニーズが高価なものなど、より大きな利益をもたらすワークロードに焦点を当てることはできますか？
- アプリケーションポートフォリオの範囲を絞り込んで、商業的にあまり成果が得られない特定のワークロードを最初に除外することはできますか？たとえば、パブリッククラウドインフラストラクチャへの導入条件が異なるためにサードパーティのソフトウェアライセンスの価格が高くなるワークロードを延期できますか？
- 最終的なランレート比較が期待した目標を達成できない場合は、次のことを検討してください。
 - まず、他の指標が期待どおりであ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、移行準備の次の段階の開始を正当化する十分な経済的機会があることを示すことです。
 - AWS 移行の初期段階以降も引き続きコストパフォーマンスを改善できる機会をリストアップしてください。

詳細なビジネスケースを作成する際には、機会リストの評価を含めてください。さらに、month-to-month ケースの継続的なメンテナンスと移行完了後のコスト最適化プロセスに機会評価を含めてください。

優先アプリケーション評価

前の段階であるポートフォリオの発見と初期計画の主な成果の1つは、詳細な評価のために一部のアプリケーションに優先順位を付けることでした。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。

早い段階でいくつかのアプリケーションの詳細を確認することで、加速が促進されます。評価と将来のアーキテクチャ設計のプロセスにより、潜在的な障害が明らかになり、より大規模な移行の前兆となる重要なタスクが明らかになります。これらのタスクには、landing zoneAWSなどの基盤を確立するための要件の収集やAWS、既存のランディングゾーン拡張と検証が含まれます。この評価は、移行の手順と戦略を検討する時でもあります。

この段階の主な成果は次のとおりです。

- 優先順位が付けられたアプリケーションの検証済みリスト
- 現在の状態メトリクスを構築する
- 移行候補者向けの初期ターゲットアーキテクチャと移行戦略の文書化
- 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、運用)
- 移行計画に関するカットオーバーの考慮事項の文書化
- AWS推定実行速度

詳細な評価データ要件の理解

次の表は、移行対象のアプリケーションとそれに関連するインフラストラクチャの完全なポートフォリオレビューを取得するために必要な情報を示しています。

表には次の略語が使用されています。

- R (必要な場合)
- O、オプション用
- 該当しない場合は該当なし

アプリケーション

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
一意の識別子	たとえば、アプリケーション ID 通常、既存の CMDB またはその他の内部インベントリおよび制御システムで利用できます。組織でこれらが定義されていない場合は必ず、ユニークな ID を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を組織に知らせる名前。該当する場合は、商用 off-the-shelf (COTS) ベンダーと製品名を含めてください。	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

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
COTS 製品とバージョン	商用ソフトウェアの製品名、バージョン	R	R	高
説明	主なアプリケーション機能とコンテキスト	R	O	高
緊急性	たとえば、戦略的または収益を生み出すアプリケーション、または重要な機能のサポートなど	R	O	高
タイプ	たとえば、データベース、顧客関係管理 (CRM)、Web アプリケーション、マルチメディア、IT 共有サービス	R	O	高
環境	たとえば、プロダクション、プリプロダクション、開発、テスト、サンドボックス	R	R	高

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
コンプライアンスと規制	ワークロード (HIPAA、SOX、PCI-DSS、ISO、SOC、FedRAMP など) および規制要件に適用可能なフレームワーク	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資産や仮想資産へのマッピング	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の種類 (例:Microsoft SQL Server エンタープライズ)	R	R	高

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
コスト	ソフトウェアライセンス、ソフトウェア運用、およびメンテナンスのコスト	該当なし	R	やや高い
ビジネスユニット	たとえば、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	たとえば、Web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)	R	R	高
目標復旧時点 (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

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
収益を生み出すアプリケーションか、ビジネス戦略アプリケーションか？	はい。応募が会社の収益に直接的または間接的に影響する場合や、事業部門が戦略的と見なしている場合は可能です。	R	O	やや高い
ユーザー数 (同時)	たとえば、内部ユーザー、外部ユーザー、または内部および/または外部のユーザー/顧客	R	R	やや高い
ユーザーの場所	ユーザーセッションの起源	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
マイグレーション戦略	たとえば、AWS 移行に関する7つのRのうちの1つ	R	R	やや高い

属性名	説明	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
データベース詳細	たとえば、パーティショニング、暗号化、レプリケーション、拡張、セキュア・ソケット・レイヤー (SSL) のサポート	R	R	高
Support チーム	たとえば、アプリケーション運用チーム名	R	O	やや高い
監視ソリューション	このアプリケーションを監視するために使用される製品	R	O	やや高い
Backup 要件	に必要なバックアップスケジュールAWS	R	R	やや高い
DR 情報	たとえば、このアプリケーションの災害復旧コンポーネント	R	R	やや高い
AWSターゲット要件	たとえば、コンポーネント、アカウント配置、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	発見、設計、移行戦略	推定実行速度	推奨忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID 通常、既存のCMDBまたはその他の内部インベントリおよび制御システムで利用できません。組織でこれらが定義されていない場合は必ず、ユニークな ID を作成することを検討してください。	R	O	高
ネットワーク名	ネットワーク内のアセット名 (例:ホスト名)	R	O	高
DNS 名 (完全修飾ドメイン名、FQDN)	DNS 名	O	O	やや高い
IP アドレスとネットマスク	内部またはパブリック 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	R	R	高

	ベースインスタンスなど			
製品名	商用ベンダーおよび製品名 (VMware ESXi、IBM パワーシステムズ、Exadata など)	R	R	高
オペレーティングシステム	たとえば、RHEL 8、Windows Server 2019、AIX 6.1	R	R	高
構成	割り当てられた CPU、コア数、コアあたりのスレッド数、メモリの合計、ストレージ、ネットワークカード	R	R	高
使用率	CPU、メモリ、ストレージ使用率などのものです。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	商品ライセンスの種類 (例:RHEL 標準)	R	R	高

インフラストラクチャは共有されていますか？	「はい」または「いいえ」は、認証プロバイダー、監視システム、バックアップサービス、および同様のサービスなどの共有サービスを提供するインフラストラクチャサービスを示します	R	O	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	たとえば、プロセスレベルでのサーバー間	R	該当なし	やや高い
AWSターゲット要件	たとえば、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	たとえば、移行に関する7つのRのうちの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アセスメント

詳細なアプリケーション評価の目標は、対象となるアプリケーションとそれに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) を完全に理解することです。落とし穴を避けるには、忠実度の高いデータが必要です。たとえば、組織はアプリケーションを完全に理解していると想定するのが一般的です。これは当然のことであり、多くの場合に当てはまります。ただし、ビジネスへのリスクを最小限に抑えるには、可能な限りプログラムデータを取得して、組織の知識と静的な文書を検証することが重要です。これにより、発見プロセスの面倒な作業が処理されます。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土壇場での変更を避けることです。たとえば、移行時には、進行中の移行の波にサーバーを含める必要があるような、未確認の依存関係に基づく変更を避けることが重要です。移行直後は、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ために、関連するプラットフォーム要件に基づく変更を避けることが重要です。このような計画外の変更は、セキュリティや運用上の問題のリスクを高めます。詳細なアプリケーション評価を行う際には、プログラマティックディスカバリーツールを使用して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。

評価の開始時に、アプリケーションの利害関係者を特定する必要があります。これらは通常次のとおりです。

- ビジネスユニットリード
- アプリケーション所有者
- 建築家
- オペレーションとサポート
- クラウド支援チーム

- 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

詳細を確認するには 2 つの方法があります。トップダウンの検出は、アプリケーションやユーザーから始まり、インフラストラクチャにまで及びます。これは、アプリケーションの識別が明確な場合に推奨される方法です。逆に、ボトムアップでの発見は、インフラストラクチャから始まり、アプリケーションやサービス、およびそのユーザーに至るまで続きます。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 マッピングが不明な場合に役立ちます。一般的には、両方を組み合わせて使用することが多いでしょう。

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とよいでしょう。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てください。単純なホスト変更や再配置の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、非効率性を特定するのに役立ちます。非効率な部分は、クラウドで少し変更するだけですぐに対処できます。

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、最初の図には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示されます。主要なコンポーネントとインターフェースを特定したら、検出ツールとアプリケーションパフォーマンス監視ツールからのプログラムデータでそれらを検証します。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やサービスへの依存関係を文書化します。

依存関係やマッピングを検証するツールがない場合は、手動による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、開いている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識別できます。手動による検出に必要な労力を過小評価しないでください。プログラマティックツールでは、間隔が長いもの（通常はごく一部）を除いて、ほとんどの依存関係を数日でキャプチャして報告できます。手動による検出では、すべてのデータポイントを収集して統合するのに数週間かかることがあり、それでもエラーやデータ欠落が発生しやすくなります。

続けて、[優先順位が付けら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](#)で指定されている情報を入手してください。次に、以下のアンケートを参考にして、詳細な評価プロセスを進めてください。特定された利害関係者と会って、これらの質問に対する答えについて話し合います。

全般

- このアプリケーションの重要度はどのくらいですか？ 収益は生み出されていますか？ ビジネス戦略的なアプリケーションなのか、それともビジネスを支えるアプリケーションなのか。他のシステムと共有してい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の変革プロジェクトは進行中ですか？
- これは社内向けのアプリケーションですか、それとも社外向けのアプリケーションですか？

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) はどのようなものですか？ 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) はどのようなものですか？
- API は何ですか？ API 名、オペレーション、URL、ポート、プロトコルなど、これらを説明してください。
- コンポーネント間、およびこのコンポーネント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れくらいですか？

操作

- 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営するのは誰か？ AWSこれらは社内チームまたは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した場合はどうなりますか？ 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？ アプリケーションにはどのようにアクセスしますか？ 同時接続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？ 今後、更新がfutureされていますか？ もしそうなら、いつ？
- このアプリケーションにはどのような既知のリスクと問題がありますか？ 障害や、重大度が中程度から高度のインシデントの履歴を教えてください。
- 使用サイクル (営業時間内) はどのくらいですか？ 営業時間は何時ですか？

- 変更凍結期間はどのくらいですか？
- このアプリケーションの監視にはどのようなソリューションが使用されていますか？

パフォーマンス

- 収集されたパフォーマンス情報は何を示していますか？ 使用量が急上昇しているのか、それとも一定で予測可能なのか？ 利用可能なパフォーマンスデータの期間、間隔、日付はどのくらいですか？
- このアプリケーションに含まれている、またはこのアプリケーションと連携するバッチジョブがスケジューリングされていますか？

ソフトウェアライフサイクル

- 現在の変化の割合 (毎週、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、プリプロダクション、プロダクションなど) はどのようになっていますか？
- アプリケーションとインフラストラクチャの導入方法にはどのようなものがありますか？
- 導入ツールとは何ですか？
- 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どのようなものですか？
- サービスレベルアグリーメント (SLA)
- 現在のテスト方法 テストステージはどのようになっていますか？

移行

- 移行に関する考慮事項はどのようなものですか？

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をすべて書き留めておきます。より完全に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入手してください。次に、彼らの知識と意見を対比してください。

回復性

- 現在のバックアップ方法 バックアップにはどのような製品が使用されていますか？ バックアップスケジュールはどのようになっていますか？ バックアップ保持ポリシーとは何ですか？
- 現在の目標復旧時点 (RPO) と目標復旧時間 (RTO)
- このアプリケーションには災害復旧 (DR) 計画がありますか？ もしそうなら、DR ソリューションとは何でしょうか？
- 最後の DR テストはいつでしたか？

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および規制の枠組みはどのようなものですか？ 最終監査日と次の監査日を教えてください。
- 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、保存中、あるい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Secure Sayer Layer Layer Layer Layer (SSL) 証明書を使用していますか？ 発行機関は何ですか？
- 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どのようなものですか？

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ていますか？
- データベースへの通常の同時接続数はどれくらいですか。最小接続数と最大接続数はいくつですか？
- 接続方法にはどのようなものがありますか (JDBC、ODBC など)。
- 接続文字列は文書化されていますか？ もしそうなら、どこで？
- データベーススキーマとは何ですか？
- 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どのようになっていますか？ 解決できない依存関係や、コンポーネントをまとめて移行する必要がある場合は書き留めておきます。
- 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) 間の接続はどのようになっていますか？
- このアプリケーションが他のアプリケーションやサービスにどのような依存関係にあるか
- 運用上の依存関係はどのようなものですか？ たとえば、パッチウィンドウなどのメンテナンスサイクルとリリースサイクル。

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future 状態を設計して文書化することは、移行を成功させるための重要な要素です。どんなに単純でも複雑でも、あらゆるタイプの移行戦略に合わせてデザインを作成することをお勧めします。設計を作成すると、アーキテクチャの変更が予想されない場合でも、潜在的な障害や依存関係、および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。

また、移行戦略の観点からアプリケーションのfuture 状態にアプローチすることもお勧めします。AWSこの段階では、AWSこの移行の結果としてのアプリケーションの外観を必ず定義してください。出来上がったデザインは、移行後のさらなる進化の基礎となります。

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ドンスを組み合わせたものです。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して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ラリ](#)には、さまざまな技術的問題やビジネス上の問題に対応するAWS、検証済みのクラウドベースのソリューションが揃っています。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規範的ガイド](#)には、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、パターンが記載されています。
- [AWSドキュメント](#)には、ユーザーガイドや API リファレンスなど、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、基礎を学ぶための実践的なチュートリアルや掘り下げたチュートリアルがいくつか用意されており、学習を開始できますAWS。](#)

クラウドへの移行のどの段階にいるかによっては、AWS基盤がすでに存在している場合があります。AWSこれらの基盤には次のものが含まれます。

- AWS地域が特定されました。
- 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。
- AWS基本的なサービスがアカウント内に導入されています。

逆に、プロセスの初期段階で、AWS基盤がまだ確立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、それらを定義するためにさらに作業が必要になったりする可能性があります。このような場合は、アプリケーションの設計作業とparallel、landing zone 基本設計を定義して実装することをお勧めします。アプリケーションの設計は、AWSアカウント構造、ネットワーク、仮想プライベートクラウド (VPC)、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、クラウド運用などの要件を特定するのに役立ちます。

[AWS Control Tower](#)は、landing zone と呼ばれる安全な複数アカウントのAWS環境を設定および管理する最も簡単な方法を提供します。AWS Control TowerAWSOrganizations を使用してlanding zone を作成します。これにより、継続的なアカウント管理とガバナンスを提供し、AWSクラウドに移行する何千もの顧客と連携しながらベストプラクティスに基づいたエクスペリエンスを実装できます。

アプリケーションのfuture 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、この戦略は初期段階とみなされます。これは、future 状態設計の一環として変更される可能性があり、潜在的な制限が明らかになる可能性があるためです。最初の仮定を検証するには、[7Rのディジションツリーを参照してください](#)。また、移行の可能性のあるフェーズも文書化してください。たとえば、このアプリケーションは1回のイベント (すべてのコンポーネントが同時に移行される) で移行されるのでしょうか。それとも、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ユニーク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。たとえば、最初のアプローチは、変更せずにアプリケーションを持ち上げたり移動したりすることです。た

だし、アプリケーションのコンポーネントは異なるインフラストラクチャ資産にある場合があり、さまざまな処理が必要になる場合があります。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、それぞれが別々のサーバー上で実行され、一方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再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再ホストできます。移行する各アプリケーションコンポーネントと関連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を移行する理由を教えてください。
- どのような変更が提案されていますか？
- どのようなメリットがありますか？
- 大きなリスクや阻害要因はありますか？
- 現在の欠点は何ですか？
- 対象範囲内と対象外の内容

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どのように再利用できるか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、現在およびfuture 州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参考にして、このプロセスを参考にしてください。

将来のアーキテクチャ

このアプリケーションのfuture アーキテクチャを説明してください。ソース環境 (オンプレミス)AWS とターゲット環境 (AWSターゲットリージョン、アカウント、VPC、アベイラビリティゾーンなど) の構成要素を含む再利用可能なダイアグラムテンプレートの作成を検討してください。

移行するコンポーネントと新しく追加されるコンポーネントの表を作成します。このアプリケーションと相互作用する他のアプリケーションやサービス (オンプレミスまたはクラウド) を含めてください。

次の表は、コンポーネントの例を示しています。リファレンスアーキテクチャや検証済みの構成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 (社内)	Amazon Route 53 が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ました
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	オンプレミスのロードバランサーを置き換えます。プール A を移行。
アプリケーションセキュリティ	DDoS からの保護	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 A	フロントエンドタイプ	Amazon Elastic Compute Cloud (Amazon EC2) を使用してリホストします。
サーバー B	フロントエンドタイプ	Amazon EC2 を使用したリホスト
サーバー C	アプリケーションロジック	Amazon EC2 を使用したリホスト

名前	説明	詳細
サーバー D	アプリケーションロジック	Amazon EC2 を使用したリホスト
Amazon Relational Database Service (Amazon RDS) — Amazon Aurora	データベース	サーバ E と F の置き換え
モニタリングとアラート	変更の管理	Amazon CloudWatch
監査ログ	変更の管理	AWS CloudTrail
パッチ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	画像ホスティング	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	アベイラビリティーゾーン 1 — VPC 1
プライベートサブネット 2	バックエンドと DB	アベイラビリティーゾーン 2 — VPC 1

名前	説明	詳細
パブリックサブネット 1	フロントエンドタイプ	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	フロントエンドタイプ	アベイラビリティゾーン 2 — VPC 1
Backup 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
博士	Amazon EC2 耐障害性	CloudEndure災害対策

コンポーネントが特定されたら、お好みのツールを使用してコンポーネント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、プラットフォームおよび移行チームなど、主要なアプリケーション関係者と共有します。次の質問をすることを検討してください。

- チームはおおむねデザインに賛成ですか？
- 運用チームはそれをサポートできますか？
- デザインを進化させることはできますか？
- 他に選択肢はありますか？
- 設計は建築基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？

アーキテクチャに関する決定

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ついて、より多くのオプションが見つかるでしょう。これらのオプションを、優先オプション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てください。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。

新しい読者が、あるオプションを他のオプションよりも優先して使用する決定の背後にあるオプションと理由を理解できるように、主なオプションが十分に詳細にリストされ、説明されていることを確認してください。

ソフトウェア使用率環境

現在の環境への変更をすべて文書化します。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。

タグ付け

各インフラストラクチャコンポーネントの必須タグ付けと推奨タグ付け、およびこの設計のタグ付け値について説明してください。

マイグレーション戦略

設計のこの時点で、移行戦略に関する最初の前提条件を検証する必要があります。選択した R 戦略について合意が得られてい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、アプリケーションコンポーネントが異なれば、移行に必要な R タイプも異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てください。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、段階的な移行アプローチについても説明してください。

自分の 7 R を決める方法については、[AWS Migration Hubおすすめの戦略をご覧ください](#)。

移行パターンとツール

アプリケーションとインフラストラクチャコンポーネントの移行戦略が定義されていれば、特定の技術パターンを調べることができます。たとえば、リホスト戦略は、[AWSアプリケーション移行サービスなどの移行ツールを使用して実装できます](#)。状態やデータを複製する必要がない場合は、Amazon Machine Image (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しても同じ結果が得られます。

同様に、アプリケーションを再プラットフォームまたはリファクタリング (再構築) するには、[AWSApp2Container](#)、[AWS Database Migration Service](#) (AWS DMS)、() などのツールを使用できます。[AWS DataSync](#)、[AWS Schema Conversion Tool](#) AWS SCT コンテナ化には、Amazon [エラスティックコンテナサービス](#) (Amazon ECS)、[Amazon エラスティック Kubernetes サービス](#) (Amazon EKS)、またはを使用できます [AWS Fargate](#)。再購入する際には、特定の製品用の AMI または [AWS Marketplace](#) のサービスとしてのソフトウェア (SaaS) ソリューションを使用できます。

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てください。長所と短所、および移行の運用準備状況を検討してください。分析に役立つように、次の質問を使用してください。

- 移行チームはこれらのパターンをサポートできますか？
- コストとメリットのバランスはどのようになっていますか？
- 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管理サービスに移動できますか？
- このパターンを実装するにはどのような努力が払われていますか？
- 特定のパターンの使用を禁止す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好ましい。ただし、パターンが 1 回しか使用されない場合もあります。1 回限りの使用パターンと別の再利用可能なパターンの労力のバランスを検討してください。

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

サービスの管理と運用

への移行のためのアプリケーション設計を作成する際にはAWS、運用準備が整っていることを考慮してください。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要件を評価する際には、次の質問を検討してください。

- 操作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？
- 期待されるサービスレベルアグリーメント (SLA) を確認します。
- 職務分掌は必要ですか？
- 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？
- 誰が何を担当しているのか？

カットオーバーに関する考慮事項

移行の戦略とパターンを考慮して、アプリケーションを移行する時点で知っておくべき重要なことは何ですか？ カットオーバー計画は設計後の作業です。ただし、予想されるアクティビティや要件に関する考慮事項はすべて文書化してください。たとえば、該当する場合は、概念実証を実施するための要件を文書化し、テスト、監査、または検証の要件を概説します。

リスク、前提条件、問題、依存関係

未解決のリスク、前提条件、およびまだ解決されていない潜在的な問題をすべて文書化します。これらの項目には明確なオーナーシップを割り当て、全体的な設計と戦略の実装が承認されるよう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てください。

実行コストの見積もり

AWSターゲットアーキテクチャのコストを見積もるには、[AWS価格計算ツールを使用してください](#)。設計で定義したとおりにインフラストラクチャコンポーネントを追加し、推定ランニングコストを算出します。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使用するサービスにまだ含まれていないソフトウェアライセンスを考慮に入れてください。

ポートフォリオ分析と移行計画

評価のこの段階では、「[ポートフォリオ発見と初期計画](#)」セクションで開始したポートフォリオレベルの発見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初期ポートフォリオのベースラインを繰り返し確立することです。このベースラインには、すべての依存関係の特定、移行のための合理化モデルの繰り返し、詳細なビジネスケースの作成、移行ウェブプランの概要作成が含まれます。その結果、必要なデータ忠実度が高くなります。この段階では時間の投資が必要です。評価結果を早めるために、ディスカバリーツールなどのプログラムティック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。

この段階の主な成果には以下が含まれます。

- 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
- 各アプリケーションのハイレベルな移行戦略
- 信頼度の高い移行ウェブプラン
- 詳細なビジネスケース

完全な評価データ要件の理解

次の表は、移行対象のアプリケーションとそれに関連する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まとめたものです。

表には次の略語が使用されています。

- R (必要な場合)
- O、オプション用
- 該当しない場合は該当なし

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 通常、既存の	R	R	高

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	推奨忠実度レベル (最小)
	CMDBまたはその他の内部インベントリおよび制御システムで利用できます。組織でこれらが定義されていない場合は必ず、ユニークな ID を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を組織に知らせる名前。該当する場合は、商用 off-the-shelf (COTS) ベンダーと製品名を含めてください。	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	商用ソフトウェアの製品名、バージョン	R	R	高

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	推奨忠実度レベル (最小)
説明	主なアプリケーション機能とコンテキスト	R	R	高
緊急性	たとえば、戦略的または収益を生み出すアプリケーション、または重要な機能のサポートなど	R	R	高
タイプ	たとえば、データベース、顧客関係管理 (CRM)、Web アプリケーション、マルチメディア、IT 共有サービス	R	R	高
環境	たとえば、プロダクション、プリプロダクション、開発、テスト、サンドボックス	R	R	高

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	推奨忠実度レベル (最小)
コンプライアンスと規制	ワークロード (HIPAA、SOX、PCI-DSS、ISO、SOC、FedRAMP など) および規制要件に適用可能なフレームワーク	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資産や仮想資産へのマッピング	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の種類 (例: Microsoft SQL Server エンタープライズ)	R	R	やや高い

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	推奨忠実度レベル (最小)
コスト	ソフトウェアライセンス、ソフトウェア運用、およびメンテナンスのコスト	該当なし	R	やや高い
Business	たとえば、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に関する情報	災害対策コンポーネント	R	R	高
移行戦略	たとえば、移行用の 7 つの R のうちの 1 つAWS	R	R	高
Support チケット	システム停止、速度低下、トランザクションスロットリング、バッチウィンドウオーバーランによる生産性と財務的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	推奨忠実度レベル (最小)
一意の識別子	たとえば、サーバー ID 通常、既存のCMDBまたはその他の内部インベントリおよび制御システムで利用できます。組織でこれらが定義されていない場合は必ず、ユニークな ID を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 (例:ホスト名)	R	R	高
DNS 名 (FQDN)	DNS 名	R	O	高
IP アドレスとネットマスク	内部またはパブリック 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	商用ベンダーおよび製品	R	R	高

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	推奨忠実度レベル (最小)
	名 (VMware ESXi、IBM パワーシステムズ、Exadata など)			
オペレーティングシステム	たとえば、REHL 8、Windows Server 2019、AIX 6.1	R	R	高
構成	割り当てられた CPU、コア数、コアあたりのスレッド数、メモリの合計、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	商品ライセンスの種類 (例:RHEL 標準)	R	R	高

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	推奨忠実度レベル (最小)
インフラストラクチャは共有されていますか？	「はい」または「いいえ」は、認証プロバイダー、監視システム、バックアップサービス、および同様のサービスなどの共有サービスを提供するインフラストラクチャサービスを示します	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

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	推奨忠実度レベル (最小)
コスト	ハードウェア、メンテナンス、運用、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、ラックスペースのシェア、データセンターの諸経費など、ベアメタルサーバの全額負担コスト	該当なし	R	やや高い
推定データ転送量 (イン/アウト)	たとえば、30 日間にわたる 1 日あたりのインフラストラクチャー資産あたり	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	推奨忠実度レベル (最小)
パイプのサイズ (MB/s)、冗長性 (Y/N)	現在の WAN リンク仕様 (1000 MB/秒の冗長構成など)	R	R	やや高い

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	推奨忠実度レベル (最小)
リンク使用量	ピーク時および平均使用率、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	接続された場所間の現在の待ち時間。	R	O	高
コスト	1 か月あたりの現在のコスト	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	推奨忠実度レベル (最小)
リHOST	各作業負荷に対する顧客とパートナーの労力 (1 日あたり)、顧客とパートナーの 1 日あたりのコスト率、ツールコスト、ワークロード数	該当なし	R	やや高い
プラットフォーム変更	各作業負荷に対する顧客とパートナーの労力 (1 人あたりの日数)、顧客とパートナーの 1 日あ	該当なし	R	やや高い

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	推奨忠実度レベル (最小)
	たりのコスト率、作業負荷数			
リファクタリング	各作業負荷に対する顧客とパートナーの労力 (1人あたりの日数)、顧客とパートナーの1日あたりのコスト率、作業負荷数	該当なし	R	やや高い
リタイア	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	既存の再利用 (Y/N)、AWS必要なリージョンのリスト、コスト	該当なし	R	やや高い
人と変化	クラウドの運用と開発についてトレーニングするスタッフの数、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
[Duration] (所要時間)	対象ワークロード移行の期間 (月)	O	R	やや高い

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	推奨忠実度レベル (最小)
並列コスト	移行中に現状の費用を削減できる期間と速度	該当なし	R	やや高い
	AWS移行中に製品やサービス、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期間と速度	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

信頼性の高い移行計画を作成するには、アプリケーション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、移行の範囲を包括的に把握できます。ポートフォリオベースラインにより、どのアプリケーションが移行の対象となるかが明確になり、「[評価データ要件全体を理解する](#)」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ていることがわかります。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 つのカテゴリに分類できます。

- application-to-infrastructure依存関係は、ソフトウェアと物理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それが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資産で実行されるコンポーネントがどのように相互作用するかを表します。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、Web フロントエンドが仮想マシンで実行され、アプリケーションレイヤーが別の仮想マシンで実行され、データベースがデータベースクラスターで実行される場合が挙げられます。

- application-to-application 依存関係は、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と他のアプリケーションまたはそのコンポーネントとの相互作用に関連します。application-to-application 依存関係の例としては、支払い処理アプリケーションや在庫管理アプリケーションがあります。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 オペレーションを使用して常に相互作用します。
- application-to-infrastructure 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ことを考えると、application-to-application サービスの依存関係は技術的には依存関係です。ただし、これらは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一般的に多くのアプリケーションで共有されるため、依存関係が長引くためです。また、通常は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性であり、ロードバランサー自体は保持または廃止されている間に、依存するアプリケーションと一緒に個別に移行される可能性があります。また、application-to-infrastructure 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ている場合です。つまり、インフラストラクチャサービスに共通の依存関係があるビジネスアプリケーションを同時に移行する必要があるということです。たとえば、Active Directory などの認証サービスは、アプリケーションの大規模な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、これらのアプリケーションに個別にアプローチし、Microsoft ActiveAWS Directory用のDirectory Service などのサービスをクラウド環境で有効にすることで依存関係に対処することです。

ポートフォリオのベースラインを設定する際には、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、移行の 7R のうちの 1 つになります ([「7R の移行戦略の繰り返し」](#) セクションを参照)。ポートフォリオのベースラインでは、7つのRのうちの1つを各アプリケーションに関連付ける必要があります。また、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は 7R 戦略を組み込む必要があります。

依存関係や移行戦略を含む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)。アプリケーションオーナー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した情報でデータを補完します。[この段階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と一致する完全なポートフォリオインベントリが得られるまで、データを収集し続けます](#)。生成されるデータセットは、移行を促進するのに役立ちます。

移行の範囲と利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を考慮してください。

優先順位付け基準の繰り返し

移行ウェ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ブプランニングに切り替えることをお勧めします。

前のセクションで、単純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紹介しました（「[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、重要度の低いアプリケーションから移行プロセスを改善し、学んだ教訓を取り入れることを推奨しているためです。ただし、この段階で長期計画を立てるには、アプリケーションを移行する順序をビジネスの推進要因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、ウェブプランニングの重要なインプット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入手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を決定する属性を選択します。

まず、ビジネス推進要因を検証します（「[ビジネス推進要因と技術指針](#)」を参照）。次に、ビジネスの推進要因に基づいて、移行するアプリケーションの優先順位を決めるの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合わせ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	設定可能な値	スコア (0-99)	重要度または関連性の倍率
オペレーティングシステム	AIX	80	ハイ (1倍)
	Solaris	80	
	HP-UX	80	
	メインフレーム	70	
	Windows	50	
	Linux	20	
ビジネス重要度	高	60	ハイ (1倍)
	中	40	
	低	20	

属性またはデータポイント	設定可能な値	スコア (0-99)	重要度または関連性の倍率
アーキテクチャ	緊密に結合	60	ハイ (1倍)
	緩く結合	20	
オペレーティングモデル	トラディショナル-CI/CD	60	中〜高 (0.8x)
	ベーシック CI/CD	40	
	フル 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	ミディアム (0.6x)
	プラットフォーム変更	40	
	再購入	30	
	リホスト	10	

次の表は、迅速なコスト削減を実現するためのビジネス推進要因に合わせ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	設定可能な値	スコア (0-99)	重要度または関連性の倍率
データベース製品	Oracle	70	ハイ (1倍)
	Microsoft SQL	70	

属性またはデータポイント	設定可能な値	スコア (0-99)	重要度または関連性の倍率
	その他	20	
オペレーティングシステム	Windows	70	ハイ (1倍)
	Linux	70	
	その他	20	
CPU 使用量	36% 以上	60	ハイ (1倍)
	36% 未満	40	
コンピュートインスタンスの数	11 以上	60	中〜高 (0.8x)
	4-10	40	
	1 ~ 3	20	
マイグレーション戦略	リタイア	80	ミディアム (0.6x)
	リホスト	70	
	プラットフォーム変更	50	
	リファクタリング (リアーキテクト)	10	

優先順位付けの基準をテストし、出力結果に概ね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。

7 R の移行戦略選択の繰り返し

この段階では、7 R のデシジョンツリーを繰り返し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、デフォルトのデシジョンツリーが導入されました。ツリーを改訂して、最初のパイロットアプリケーションの移行中の学習内容を考慮し、ビジネス推進要因、優先順位付け基準、およびお客様固有の状況に合わせることをお勧めします。サンプルアプリケーションを

使用してデジジョンツリーを検証し、それでも期待通りの戦略が作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てください。作成されたツリーは、アプリケーション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に移行戦略を割り当てる上で重要です。

[前の7Rのセクションで説明したように](#)、7Rはインフラにも当てはまるので、それに応じて割り当てることも同様に重要です。特定のアプリケーションコンポーネントには移行戦略がありますが、インフラストラクチャレベルでは、各インフラストラクチャ資産は特定の移行戦略に従います。この戦略は、サポートするアプリケーションコンポーネント用に確立された戦略とは異なる場合があります。

7Rのデジジョンツリーはアプリケーションコンポーネントにのみ適用されることに注意してください。インフラストラクチャの移行戦略は、アプリケーションに合わせて選択された戦略から導き出されます。たとえば、アプリケーションコンポーネントを再プラットフォーム化する場合、そのコンポーネントをホストしている現在のインフラストラクチャは廃止される可能性があります。

移行戦略が各アプリケーションコンポーネントとそれに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てください。この情報は、必要な労力、キャパシティ、スキルを見積もったり、移行ウェーブプランを作成したりする際の重要な要素となります。

自分の7Rを決める方法については、[AWS Migration Hubおすすめの戦略をご覧ください](#)。

ウェーブプランニング

ウェーブプランニングでは、依存関係グループとは、技術的な依存関係と技術以外の依存関係が解決できないアプリケーションとインフラストラクチャの集まりです。このような依存関係があるため、依存関係グループ内のアプリケーションとインフラストラクチャを同時に、または特定の日に移行する必要があります。たとえば、仮想マシンで実行されているアプリケーションと、別の仮想マシンで実行されているデータベースでは、レイテンシーが低い要件、トラフィック量が多く、クエリが複雑な場合は、一方のコンポーネントをクラウドで運用し、もう一方のコンポーネントをオンプレミスで運用するのではなく、一緒に移行する可能性があります。同様に、同様の低レイテンシー要件を持つAPIを介してやり取りする独立したアプリケーションも同時に移行されます。

通常、移行の波は4～8週間で、1つ以上の移行イベントが含まれることがあります。ディペンデンスグループはウェーブに結合されるため、1つのウェーブに1つ以上のディペンデンスグループを含めることができます。ウェーブには、移行に必要な他のアクティビティも含まれています。これらには、AWSインフラストラクチャの設定 (landing zone、セキュリティ、運用など)、移行ツール、データ複製、カットオーバー計画、テスト、移行後のサポートなどの移行アクティビティが含まれます。

成功を測定し、進捗状況を追跡するには、ウェーブを成果やビジネス推進要因に合わせて調整する必要があります。これはウェーブの持続時間やウェーブに含まれるディペンデンシーグループにも影響します。ウェーブの完成には、測定可能な成果が反映されているはずですが、ウェーブの計画には、技術的な指針となる原則など、他の要素を組み合わせることもできます。たとえば、ウェーブは環境（開発、テスト、プロダクションなど）または移行戦略（リホストウェーブ、リプラットフォームウェーブ）によって定義できます。

効果的で信頼性の高い移行計画を作成するには、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。

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、移行の波の生成と依存グループの定義の一因となります。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。たとえば、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、月末、四半期末処理などの主要な営業日は、ウェーブプランに影響します。

依存関係がソフトかハードかを判断します。ソフト依存関係とは、コンポーネントの位置に依存しない、2つ以上のアセット間、またはアセットからコンストレイントまでの関係です。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、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することで分離できます。別の例としては、メンテナンス作業に影響を与えずにメンテナンス期間中に移行できるシステムがあります。

ハードディペンデンシーとは、2つ以上のアセット間の関係、またはアセットとコンストレイントの関係で、場所によって異なります。たとえば、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レイテンシーが低いことに大きく依存している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うち1つだけをクラウドに移行すると、機能やパフォーマンスの問題が発生し、解決できなくなります。同様に、リソースの可用性（移行を実行するチームなど）などの非技術的な理由や、2つのシステムを特定の時間枠にしか移行できないメンテナンスウィンドウなどの運用上の制約により、これらの資産に強い依存関係が生じる可能性があります。

移行ウェー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、専用の検出ツールなどの信頼性の高いデータソースから取得し、この情報をアプリケーションの優先順位付け基準や運用状況と組み合わせます。優先度ランキングの上位にあるアプリケーションを、最初の移行のターゲットにする必要があります。リソースの可用性、リスク許容度、ビジネス上および技術上の制約、経験、スキルに基づいて、ウェーブ容量（ウェーブに含めることができるアプリケーションの数）を決定します。AWSAWSプロフェッショナルサービスまたは移行コンピテ

ンシーパートナーと連携することを検討してください。これらのパートナーは、プロセスを通じてサポートしてくれる専門家を派遣できます。

優先順位の基準は、アプリケーションをクラウドに移行する順序を最初に示すものです。ただし、特定の時点に移動するアプリケーションの実際の決定要因は、依存関係グループです。これは、優先度が高いとランク付けされたアプリケーションが、ランキングの中間または最下位にあるアプリケーションに強く依存する可能性があるためです。

移行戦略は波の構成にも影響します。たとえば、数週間から数か月かけて分析、設計、テスト、準備を要するリファクタリング戦略を必要とする優先度の高いアプリケーションは、後から導入される可能性があります。

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、そのウェーブで移行されるアプリケーション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、現在開発中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、および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。

ウェーブのオーナーシップとガバナンスを確立することは、ウェーブワーク、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上で重要です。計画を管理するためのガバナンスの枠組みが整っていることを確認してください。

ウェーブプランの概要を説明するには、デフォルトのウェーブコンストラクトから始めます。波の中では何が起こるのでしょうか？ 初期入力定義されると、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. カットオーバー計画を改善してください。このアクティビティでは、他の社内外のチームとの調整を含め、移行時に取る必要のあるランブックと手順の概要を説明する必要があります。
2. ロールバックプランを改良してください。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 landing zone (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. 移行ツールを運用します。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てデータ転送を開始します。

6. カットオーバー計画を実施し、ドライランを計画します。参加しているすべてのチームメンバーをグループ化し、すべてのステップを事前に確認してください。
7. データ複製とインフラストラクチャの展開を監視します。
8. のインフラストラクチャとアプリケーションの運用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する。
- 10 該当する場合は、コンプライアンスおよび規制要件 (移行前と移行後のワークロード検証など) を確認してください。
- 11 AWSアプリケーションをに移行し、本番稼働前のテストを実施します。
- 12 運用チームと移行チームが問題を解決し、最適化を適用できるようになったら、3 日間など、移行後のサポートを提供します。
- 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、future 波に取り入れましょう。
- 14 運用の引き渡しや報告用の指標の取得を確認して、ウェーブクロージングを行います。

これらの各アクティビティにかかる時間は、範囲の複雑さ、波の容量、関係する人々、および独自の状況によって決まります。可能な場合は、遅延や移動の妨げとなる要因の影響を軽減するため、波が小さい方が望ましいです。ウェーブのデフォルトの持続時間をチームで決定します。

次に、日付の分析に進み、(まだ用途が割り当てられていない状態で) 空の波の高レベルの初期構造を作成します。次の質問について考えます。

- 移行プログラムの合計期間はどれくらいですか？
- 締め切りはいつですか？
- データセンターの終了日は決まっていますか？
- 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どのようになっていますか？
- アプリケーションのメンテナンスとリリースのサイクルはどのようになっていますか？
- 移行を避けるべき日付 (リリースとメンテナンスのサイクル、年末、休日、月末処理など) はありますか？

これらの点を考慮して、波を平面図にプロットします。移行プロセスを迅速に進めるため、できる限りウェーブを重ねることをお勧めします。波を重ね合わせるための鍵は、波の中で何が起こるかを定義して考えることです。通常、導入作業、ターゲットインフラストラクチャの検証、およびデータ同期は、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、運用の引き継ぎに焦点を当てます。つまり、プロセスの各段階でさまざまなチームが関与し、ある程度の効率化を図ることができ

ます。たとえば、ターゲットとなるインフラストラクチャの準備に携わる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掛かることができます。一般に、ほとんどの波の長さと構造が同じであると、工場のような移行アプローチが容易になることが望ましいです。ただし、ウェーブプランニングのプロセス中に、依存関係や運用要件に合わせて特定のウェーブのサイズを拡張できます。

次に、特定された依存関係グループに基づいて、ウェーブに含めることができる依存関係グループの数という観点からウェーブの最大サイズを決定します。波の大きさは、通常、リスク選好 (たとえば、parallel する変更をどれだけ許容できるかなど) とリソースの可用性 (たとえば、利用可能なリソース、スキル、予算でどれだけのparallel 変更を実行できるか) によって決まります。ただし、早期の計画では、リソース要件や空き状況に制限されないようにしてください。複数のディペンデンスグループを含むウェーブは、future イテレーションでより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が確認されたら、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 (含まれる依存グループの数) を調整することを確認してください。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。すべての波形が定義されるまで、必要に応じて波形計画を繰り返します。

変更管理

アプリケーションのポートフォリオと関連インフラストラクチャは、移行プログラムのライフサイクル中に変化します。長期にわたる移行プログラムは、通常のビジネスの進化や変化と共存しています。アプリケーションは、移行を待つ間も進化し続けます。サーバーが追加または削除され、新しいインフラストラクチャがオンプレミスで導入されます。ウェーブやディペンデンスグループのスコップを変更する必要があることが予想されます。特に移行日が近づき、これまで知られていなかった依存関係が特定された場合や、新しいサーバーがインベントリに含まれる場合は、変更が必要です。これは移行中に発生することがあります。

スコップの変更は依存関係グループとウェーブに影響します。変化に対応して影響を最小限に抑えるには、スコップコントロールメカニズムを確立することが重要です。スコップ変更管理メカニズムでは、スコップに関する信頼できる唯一の情報源を定義する必要があります。これは、移行プログラムのガバナンスで定義されている範囲を管理するためのツールでも、.csv ファイル、スプレッドシート、データベースでもかまいません。変更を特定し、影響を分析し、関連する利害関係者に変更を伝えて、彼らが行動を起こせるようにする必要があります。その結果、ウェーブプランが繰り返されます。

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大し、トランスフォーメーションプログラムをサポートするためのより詳細な情報を提供することをおすすめします。すぐに組み立てられる最初の方

向性を示すビジネスケースは、基本的なステップや次の段階の詳細な計画に投資するのに十分な自信が持て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ことで、この計画プロセスを次のようにサポートできます。

- 何を移行して最新化すべきか、どのオプションを選択すべきか、どのように作業を段階的に進め、優先順位を付けるかについての意思決定に役立つ財務分析を提供する
- 詳細に再検討することにより、当初の方向性のある財務ケースを検証、改良、発展させます。
 - インフラコスト削減の可能性
 - 社内のIT生産性とアウトソーシングされた業務の効率性
 - プログラムのセットアップ、移行、近代化に必要な投資の見積もり
- 移行がもたらすさらなるバリュードライバーの特定、規模の見積もり、追跡プロセスの設定

詳細なビジネスケースでは、次の項目を設定します。

- 少なくとも移住の第一段階を実施するための権限と投資を確保するための客観的基盤
- プログラムの基準となる最低限の財務実績期待値
- さまざまな移住計画や優先順位付けの決定を行う際の財務基盤を明確にすることで、プログラムの進行中に状況や人が変化した場合に、新しい指導者が情報に基づいた選択を行えるようになります。
- ワークロードを移行して運用を開始するにあたり、初期使用状況データが利用可能になった後に検討すべきコスト最適化の段階的な領域についての洞察
- クラウドトランスフォーメーションがレジリエンスとアジリティの向上によってビジネスにもたらす価値の見積もり
- レジリエンスとアジリティの向上による経済的利益の見積もりに使用される関連KPI、指標、前提条件は、プログラムから主な利益の実現を促進するためのベースラインとなります。

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作成する場合、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ために、複数のシナリオを作成する必要があります。

最小限の変更シナリオ — 予想される最低限の財務実績を評価するには、現状に予想される最小限の変化を想定したシナリオを用意します。このシナリオは、最悪のシナリオとしては、移行への投資を義務付ける際に役立つサポートとなります。このシナリオでは、予想される容量の増加の度合いと、

quality-of-service 可用性や回復力などの他のニーズに対する最小限の変化をモデル化します。変更を最小限に抑えることで、現在の運用モデルにおけるコストとリソースの非効率性が最小限に抑えられます。

最も可能性の高いシナリオ — プログラムの戦略と優先順位付けの決定に役立つように、ビジネスが期待していることを反映したシナリオを準備します。このシナリオには、使用率がピーク時に増加または減少する可能性が高いこと、および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）に対する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。

その他の具体的なシナリオ — ビジネスケースに大きな影響を与える可能性がある前提をまだ立てる必要がある場合は、その仮定が当てはまる場合と当てはまらない場合の両方のシナリオを作成します。ただし、これらの代替シナリオの数は最小限に抑えることをお勧めします。合計で 3 ~ 4 つを超えるシナリオを作成すると、進行が遅くなり、コストがかかり、混乱が生じ、保守が困難になります。可能な限り、実験を行い、より大きな仮定を取り除くように努めてください。

インフラストラクチャと移行コストモデルの検証と改良

ポートフォリオ分析を完了し、AWSターゲットサービスの設計とサイジングを準備したら、シナリオごとに現在の運用モデル (COM) と future 運用モデル (FOM) のランニングコストの見積もりを調整します。AWS通常、次の項目について見積もりを絞り込む必要があります。

- ハイパーバイザホストサーバ、ベアメタルサーバ、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、設置、およびメンテナンスの COM インフラストラクチャコスト。シナリオに必要なキャパシティの実際の価格設定と discount レベルを合わせて計算します。
- COMデータセンターおよびコロケーション施設のコスト（スペース、冷却、電力、ラック、無停電電源装置（UPS）、配線、物理的セキュリティシステム（成長に合わせて容量に合わせて規定）、シナリオの高可用性とディザスタリカバリ（DR）レベルを含む）
- 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 (VPN) のコストを含む COM ネットワークサービスのコストは、シナリオに必要な接続、帯域幅、スループット、および遅延に関する契約価格に基づいて計算されます。
- COM アプリケーションおよびインフラストラクチャソフトウェアのコストは、シナリオに応じて使用量を増加または削減するための既存の契約に基づきます。
- AWSFOMのユーティリティコスト（必要に応じてテクニカルサポートやマネージドサービスを含む）。これは、改良されたサービスアーキテクチャ、インスタンスサイズ、優先価格モデル、予想される使用量、使用量の変動性に基づいています。

- FOMアプリケーションライセンスは、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構成、時間の経過に伴う増加、およびライセンス譲渡可能性のルールに基づいています。
- FOMの移行と近代化のコストの見積もり。シナリオのベースラインとなる移行ウェーブプランを反映するように調整し、各ワークロード、特に再プラットフォーム、再購入、またはリファクタリングするワークロードのコストを提供するように詳細に説明しました。
- FOMの廃止措置費用（資産償却費および契約早期解約費用の見積もりを含む）は、ベースラインの移行ウェーブプランにおける廃止措置のタイミング、償却を最小限に抑えるためにどの資産を再利用できるか、どの資産を切り替えることができるかの検証、および物理的資産と媒体の処分費用を反映するように改訂されました。
- 移行のparallel 実行コストは、各移行切り替えのタイミングと既存の各サービスの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の生産性とIT運用を改善し、効率化のバリューモデルをサポートします

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バリューモデルを洗練および発展させるには、主に2つのアプローチがあります。どのアプローチを選択するかは、COM を社内で管理するか、請負業者または外部委託サービスで管理するかによって異なります。

社内チームの生産性向上

ITの運用とサポートを社内で管理する場合、ビジネスケースの焦点は次の点にあります。

- 移行や対象範囲に含まれる運用の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空いた時間を、一般的に価値の高い他の活動に容易かつ生産的に活用できることを検証し、進歩の機会を与え、チームにはより大きな報酬を与え、組織にとってより大きな価値をもたらす

チーム内の各役割の各メンバーがさまざまな日常業務に費やしている時間を評価し、さまざまな活動で期待される作業負荷の削減に関するガイドンスを提供します。

次の表は、チーム内のさまざまな役割にわたるIT運用とサポート作業の大部分を消費するタスクについて、アクティビティごとに一般的な作業負荷削減レベルを示す最初のガイドンスを示しています。この表には、生産性がどのように達成されるかの説明が含まれています。

注:記載されているアクティビティは通常、複数の役割を担うチームメンバーによって実行されるため、各タスクの生産性節約効果は、チーム内の全役割で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、ネットワーキングなど）ごとに編成された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リーダーに共通している場合があります。

運用およびサポート活動	節約のレベル	生産性向上ドライバー
インフラストラクチャデザイン	中	設計が単純化され、考慮すべきパラメータが少なくなります。
資本支出の計画と予算編成	高	運用コストを重視したエラスティックサービスにより、予算や計画に関する問題が事実上すべて解消されます。
購入	高	AWS口座開設後は、調達が大幅に簡素化されます。
キャパシティプランニング	中～超高	ネットワークとコンピューティング能力管理のワークロードは通常ほとんど不要になり、ストレージに関しては大幅に簡素化されます。
チューニング	高い-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はチューニングは不要で、他のサービスではほとんど必要ありません。
ハードウェア障害管理	非常に高い	クラウドでのハードウェア処理のあらゆる側面は、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監視	高	AWSツールのサポートと自動化により、監視と通信が大幅に簡素化されます。

運用およびサポート活動	節約のレベル	生産性向上ドライバー
セキュリティ管理	中	AWSセキュリティ機能と、AWSクラウドのハードウェア、ソフトウェア、ネットワーク、 および施設のセキュリティ責任を担うこと で、作業負荷が大幅に軽減されます。AWS
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。
ラッキングとスタッキング-ハードウェアロジスティクス	非常に高い	クラウドでのハードウェア管理のあらゆる側面は、によって透過的に処理されますAWS。
バックアップ	中	Backup は、AWSツール、柔軟なストレージシステム、自動化によって大幅に簡素化されます。
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、監視、またはプロビジョニング管理アクティビティは必要ありません。

運用およびサポート活動	節約のレベル	生産性向上ドライバー
デバイスおよびサービスのセットアップと試運転	高い-非常に高い	VPN、AWS Direct Connect、AWS やデータセンターへの接続を確立するための WAN 接続デバイスを除き、AWS 移行先の不動産のハードウェアセットアップ作業は通常少なくなります。
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適用と保守は、通常、移行設計の一部として大幅に自動化されます。
脅威、脆弱性、リスク評価	高	AWS コアプラットフォームに重点を置いてこの要素をサポートし、AWS 安全なアーキテクチャを提供するメカニズムにより評価が簡単になります。
データセンターのインフラプロジェクト管理	高	インフラサービスの拡張、更新、廃止のための設置工事のプロジェクト管理。インフラストラクチャのソフトウェアとサービスの管理はある程度残っていますが、これはオンプレミスのインフラストラクチャよりもはるかに簡単で、ハードウェアアクティビティは不要です。

運用およびサポート活動	節約のレベル	生産性向上ドライバー
データセンターの施設管理	中～超高	すべてのサーバ、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ファシリティ管理作業は、移行対象すべて削除されます。ただし、通常、WAN リンクネットワークデバイスや、ハイブリッドアーキテクチャでオンプレミスで維持されているインフラストラクチャのための設備を提供するには、まだいくらかの作業が残っています。
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と、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棄の自動化を組み合わせることで、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全体の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できます。AWS CloudFormation または、を使用して簡単に設定できるランディングゾーンを使用することで簡素化できます AWS Control Tower。

運用およびサポート活動	節約のレベル	生産性向上ドライバー
IT サポート	中	L1とL2のサポートの削減は、セルフサービスプロビジョニングのためのサービスカタログ機能の使用、低コストの高可用性アーキテクチャの使用の増加（システム停止の削減、自動スケーリングとエッジコンピューティングの設定）による容量とパフォーマンスの問題の軽減によって達成されます。
データベース管理	ミニマム-ロー	これらの活動はほとんど変わりません。通常、オンプレミスのインフラストラクチャと同じレベルのリソースが提供されます。AWS
インフラストラクチャとセキュリティ要件の取得、分析、設計	最小限	
ドキュメント	最小限	
アプリケーションとパフォーマンスの監視	最小限	
L3 テクニカルサポート、質問への回答、トラブルシューティング、問題解決	最小限	
アプリケーションソフトウェアのインストールと設定	最小限	
アプリケーション L3 サポート (予算編成と長期的なキャパシティプランニングを除く)	最小限	

次の表は、各作業負荷削減の予想される削減を示しています。

レベル	期待
非常に高い	85%-100%
高	60%-90%
中	30%-70%
低	10%-35%
最小限	0%-10%

これらの指標は、生産性の向上を評価し、それを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、特定の状況に基づいて異なります。一般的なシナリオと保守的なシナリオを推定するには、範囲の中間点と下限の両方で生産性の節約額を計算すると便利です。

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役割別に把握しておくことで便利です。このデータは、運用を見積もるためのより良い基盤を構築し、新しいプロジェクトやサービスの拡大にかかるコストをサポートします。

IT業務のアウトソーシングとサポートコストの削減

ITの運用とサポートが主に外部委託または請負業者に管理されている場合、AWSパートナー主導 (AMS) AWSを含むマネージドサービスソリューションを提供するパートナーに見積もりを依頼することで、future 運用モデル [AWS Managed Services](#) (FOM) のコスト配分を作成できます。また、「[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直接連絡してAMSの価格をリクエストすることもできます。

詳細なビジネスケースについては、ベンチマークの数値を、AWS改訂された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見積もりに置き換えてください。コストには、1回限りの実装コンポーネントと、使用量に応じた稼働率が含まれます。

残っている IT 運用、移行しないサービスについて継続しなければならないサポート、契約上の罰則がある場合の1回限りの費用(たとえば、早期解約など)を含めてください。AWS

レジリエンスバリューモデルの開発

これでAWS、高可用性、災害復旧、耐障害性に優れたさまざまなアーキテクチャを構築できます。従量制の価格設定では、サービスは使用した場合にのみ課金されます。これら2つの要素が相まって、レジリエンスのための優れたコストパフォーマンスが実現します。

さらに、AWSお客様はこれを利用してワークロードの回復力を向上させています。[IDC 2018 の調査では](#)、参加しているお客様が、システム停止回数を年間 73% 減らし、平均復旧時間 (MTTR) を 58% 短縮し、生産性の低下を 94% 削減した例が示されています。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によるメリットよりも 50% 大きいことが示されました。

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最新化することで、レジリエンスがさらに高まります。ビジネスの俊敏性を高めるためにテスト自動化機能を備えた CI/CD パイプラインを導入すると、ソフトウェアの不具合が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。

この価値を評価してビジネスケースに含めるには、まずアプリケーション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メリット機会の全体像を把握します。これには以下の項目が含まれます。

- サービスの中断の数、平均継続時間、および性質:
 - サービスの中断の例としては、システム停止、パフォーマンスの低下、予定されているバッチやメンテナンスの時間枠の超過、主要機能のバグ、ピーク時のアクセス制限などがあります。
- eコマースシステムなどの収益創出サービスで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率に基づく、サービスでの中断により完了できなくなる可能性のあるトランザクションの数
 - 影響を受けた各トランザクションの平均値
- 開発プロセスの早い段階で欠陥を発見するコストに比べて、サポートエンジニアが生産システムの欠陥を解決するために費やす時間の追加コスト
- 内部ユーザーの生産性への影響と時間の損失によるコスト

次に、回復力の向上によって得られるはずの、サービス中断によって失われる時間の削減が予想される、より控えめな形で削減されるかどうかを評価します。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たシステム停止とMTR (RPO) と目標復旧時点 (RPO) と目標復旧時点 (RPO) と目標復旧時点 (RPO)
- 自動スケーリングなどの機能を使用して、処理速度低下の軽減、容量制限の解消、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、スピンアップ/スピンドウンしたインフラストラクチャでの自動リグレッションテストにより、本番環境でのみ発見されるアプリケーションバグの数を減らし、コストを最小限に抑えます。

これらをまとめて、移行と近代化の対象となるアプリケーションのポートフォリオを作成し、ケースの年ごとに予想される、より控えめなビジネスバリューの数値を計算します。メリットは移行スケジュールに合わせて増加し、その後、貢献するアプリケーションの使用量の増加が予想されるのに合わせて量を拡大する必要があります。

ビジネスアジリティバリューモデルの開発

ビジネスアジリティこそが、AWSお客様が移行する主な理由ですAWS。 [IDCAWS が実施した2018年の顧客調査では](#)、測定された利益全体の 47% がビジネスアジリティのメリットであり、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がわかりました。

トランスフォーメーションによって得られるビジネスアジリティのメリットをすべて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となるアプリケーションに焦点を当てることで、このメリットの大部分をモデル化し、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・アジリティ・バリュー・モデルを段階的に改良し、拡大していきます。これにより、ビジネスケースの関連性が保たれ、プログラムを運営するための主要な意思決定支援ツールとして使用できます。

次のガイクスを構築します。

- 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てください。
 - 収益を生み出すワークロード
 - 効率性の向上とビジネスコストの削減を目的とした業務運用のワークロード
 - 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ツール
- 収益と効率を生み出すワークロードについては、次の作業を実行してください。

- 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によってもたらされると予想される収益の伸びや運用効率について、現実的かつ慎重な評価を行います。
- アプリケーション開発速度が向上し、インフラストラクチャの導入時間が短縮されたことで、AWS年間のメジャーリリースとマイナーリリースの数の増加を見積もってください。これに関するいくつかのベースライン指標は IDC レポートに記載されています。
- より現実的で控えめな利益期待値を計算してください。ビジネスケースの期間にわたってそれらをマッピングし、それぞれのワークロードが移行された後しばらくして最大限の効率性まで引き上げられるように考慮します。
- ビジネス生産性向上ツールについては、次の作業を実行してください。
 - 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によってもたらされると予想される時間の節約について、現実的かつ控えめに評価してください。
 - 影響を受けたユーザーベース全体にわたる人々の時間と労力の平均コストを見積もります。
 - メジャーリリースとマイナーリリースの頻度の増加を数値として使用し、ビジネスケースの期間におけるメリットを計算します。

開発者の生産性が向上し、立ち上げまでの時間が短縮されるため、各ワークロードの純利益項目をビジネスケースのキャッシュフローモデルに追加して、割引キャッシュフロー、NPV、ROI、MIRR、投資回収の計算に含めます。

継続的な評価と改善

この評価段階では、次の2つの側面に焦点を当てます。

- アプリケーションの種類ごとに、継続的に詳細なアプリケーション評価を行います
- ポートフォリオの継続的な進化と改善

最初の側面は、継続的な詳細なアプリケーション評価であり、アーキテクチャとテクノロジーレベルに至るまでの詳細な発見と分析に重点を置いて、特定の波における各アプリケーション、AWS提案された設計、および移行戦略を完全に理解します。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。

2つ目の側面は、ポートフォリオの継続的な進化と改善であり、ポートフォリオ管理と、ビジネスケースの進化と追跡を含め、時間の経過とともにアプリケーションをどのように改善する予定であるかに重点を置いています。

この段階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範囲
- 特定の移行段階におけるアプリケーションのターゲットアーキテクチャと移行戦略を文書化
- 移行パターンとツールの特定と検証
- 各ウェーブにおける要件 (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を文書化

この段階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
- 提案されているアーキテクチャとテクノロジーの変更、およびそれらに期待されるメリット
- 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、運用)
- 実装計画

継続的評価データ要件の理解

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を組み合わせたものです。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。

- ウェーブアセスメントとアプリケーションの最適化には、「[優先アプリケーションアセスメント](#)」セクションのデータ要件を使用してください。
- 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てください。
- 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のセクション](#)を参照してください。

詳細なウェーブアセスメント

移行の波を先取りし、移行の鍵となるアプリケーションの詳細な評価には、[優先順位付けされたアプリケーション評価段階と同じ要件と推奨事項があります](#)。目標は、特定の波におけるアプリケーションの現状を詳細に理解し、運用面、ツール、特定の移行パターンなど、future アーキテクチャ設計と移行戦略を策定することです。

[優先順位付けされた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グループに適用します。移行計画の各段階の前にこのプロセスを繰り返してください。重要なのは、詳細な評価から波が始まるまでの間に十分な時間を確保することです。必要な時間は、Wave 要件を実装して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。それらのチームと協力して、詳細なウェーブアセスメントとウェーブのスケジュールを設定します。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。

最適化と最新化のための評価

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移行先のワークロードの評価に似ていますAWS。主な変更点は、評価を実施するためのデータのソースです。にはAWS、out-of-the-box 実行中の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ツールと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最新化する内容と方法は、お客様固有の推進要因と状況によって異なります。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調整し、学んだ教訓を取り入れることに重点を置いています。モダナイゼーシヨ

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採用など、アプリケーションを次のレベルに引き上げ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](#)。最適化と最新化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の最適化とITコストの削減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により、AWSワークロードのリソースを推奨してコストを削減し、パフォーマンスを向上させます。
- [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管理にかかる時間を短縮して、より多くの時間を構築に費やすことができます。
- [Amazon S3 Storage Lens](#) を使用すれば、オブジェクトストレージの使用状況とアクティビティについての傾向を、組織全体で確認できます。コスト効率を高め、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推奨事項を作成します。
- [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、コードの品質を向上させ、アプリケーションの最もコストのかかるコード行を特定するためのインテリジェントな推奨事項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、AWS必要な場所で一貫した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:アプリケーションモダナイゼーション入門](#) (ブログ記事)
- [サーバーレス Web アプリケーションのコストの最適化](#) (ブログ記事)
- [ウィンドウズオン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](#) (AWSre: Invent 2020)
- [AWSマイクロサービスガイド](#)

ウェーブプランの繰り返し

移行プログラムが進み、移行する波が増えるにつれて、教訓と変化するビジネス上の優先事項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ログラムでは、ビ

ビジネスの推進要因と組織の変化を再評価し、移行の波の計画が引き続き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を与えます。何が起きているかを把握できなくなることを避けるため、[ウェーブプランを最新の状態に保ってください](#)。計画には、提供されている内容を反映して追跡し、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、特に長期にわたるプログラムの場合は、ビジネス上の圧力によって移行と近代化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。

新しい情報が入手可能になったときにビジネスケースを進化させること、および詳細なビジネスケースに記録されている期待と照らし合わせて実際の商業パフォーマンスを追跡することの両方をお勧めします。これには以下が含まれます。

- 組織の新たな構造変化がビジネスの優先事項に影響を及ぼし、それに伴うIT戦略とアプリケーションポートフォリオにも影響を与える
- 移行と最新化が目指すアプリケーションポートフォリオの一部の商業的重要性の高まり、またはアプリケーションポートフォリオへの変更
- 移行後のアプリケーションに関する実際のリソース使用率データの可用性（サイズの調整、段階的なモダナイゼーションのケースの定量化と確認など）
- 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入手可能性、および可能な運用改善と自動化の分析
- ソフトウェア開発とメンテナンスのサイクルタイムの変化、開発段階ごとのソフトウェア不具合、サービスの可用性情報を測定するデータの可用性、さらに改善の余地がある領域の根本原因分析

ビジネスケースと照らし合わせてパフォーマンスを追跡することで、移行開始後に簡単に評価および定量化できるさらなる改善点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生み出す方向に変革を導くための準備がはるかに整っています。

これは、ケース内のIT生産性、耐障害性、およびビジネスアジリティのメリットにとって特に重要です。これらは通常、前もって評価するのが最も大きい要因であり、またより困難な要因でもあります。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ている

問題を深く掘り下げて解決することができます。または、ビジネスケースを調整して、最も継続的な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参照

- [Amazon Builders' Library](#)
- [アプリケーションのモダナイゼーション](#) (AWSre: Invent 2020)
- [アプリケーションポートフォリオ評価戦略](#)
- [AWSアーキテクチャーセンター](#)
- [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:アプリケーションモダナイゼーション入門](#) (ブログ記事)
- [AWS ドキュメント](#)
- [リソースセンターのご利用開始にあたって](#)
- [AWS Marketplace](#)
- [AWS Managed Services パートナー](#)
- [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 Web アプリケーションのコストの最適化](#) (ブログ記事)
- [AWS Prescriptive Guidance](#)
- [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](#)
- [ウィンドウズオンAWS](#) (ブログ)

AWS のサービス

- [AWSアプリ 2 コンテナ](#)
- [AWS Application Migration Service](#)
- [アマゾン CodeGuru](#)
- [AWS Control Tower](#)

- [データベースの自由](#)
- [AWS Database Migration Service](#)
- [AWS DataSync](#)
- [AWS Direct Connect](#)
- [Amazon ECS](#)
- [Amazon EKS](#)
- [AWS Fargate](#)
- [AWS Managed Services](#)
- [移行評価者](#)
- [AWS Migration Hub戦略に関する推奨事項](#)
- [AWSランディングゾーン](#)
- [AWS 料金計算機](#)
- [AWS Schema Conversion Tool](#)
- [Amazon S3 Storage Lens](#)
- [AWS Snowball](#)
- [AWS Snowcone](#)
- [AWS VPN](#)

その他のリソース

- [Amazon Web Services によるビジネス価値創出のためのビジネスおよび組織変革の促進](#)
- [IDC 2018 調査](#)

寄与要因

このガイドの執筆者は次のとおりです。

- シニア・アプリケーション・アーキテクト、ジャーマン・ゴンサルベス
- マーク・バーナー、シニア・アドバイザリー・コンサルタント
- プロダクトオーナー、ザック・ハンセン

以下の人々がこのガイドに貢献しました。

- アシッシュ・アメタ、シニア・アプリケーション・アーキテクト
- ロドニー・グリリ、プリンシパル・コンサルタント
- ノビ・インドリヤニ、シニア・アプリケーション・アーキテクト
- ルイス・サラザール、シニア・エンゲージメント・マネージャー
- ナヴィン・ワサン、シニア・アドバイザリー・コンサルタント

ドキュメント履歴

この戦略は、この方法の大きな変更点をまとめたものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#)をサブスクライブできます。

変更	説明	日付
二	初版発行	2021 年 11 月 12 日

AWS 規範的ガイドンスの用語集

以下は、AWS 規範的ガイドンス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の購入、アプリケーションの書き換え、お客様の既存のオペレーションの変更を行うことなく、インフラストラクチャをクラウドに移行できます。この移行シナリオは AWS の VMware Cloud に固有のもので、お客様のオンプレミス環境と、および AWS の間の仮想マシン (VM) 互換性とワークロードの移植性をサポートします。インフラストラクチャを AWS の VMware Cloud に移行するときに、お客様のオンプレミスのデータセンターから VMware Cloud Foundation テクノロジーを使用できます。例: AWS の Oracle データベースをホストしているハイパーバイザーを VMware Cloud 上に再配置する。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したい

アプリケーション、およびそれらを移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。

- 使用停止 — 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コントロール」](#)を参照してください。

抽象化されたサービス

[「マネージドサービス」](#)を参照してください。

ACID

[アトミック性、整合性、分離、耐久性](#)を参照してください。

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](#) よりも柔軟性がありますが、多くの作業が必要です。

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の方法。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

行のグループを操作し、グループの単一の戻り値を計算するための SQL 関数。集計関数の例には、SUMや などがありますMAX。

AI

[「人工知能」](#)を参照してください。

AI Ops

[「人工知能オペレーション」](#)を参照してください。

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

繰り返し発生する問題に対して、そのソリューションが逆効果であったり、効果がなかったり、代替ソリューションよりも効果が低かったりする場合によく使用されるソリューション。

アプリケーションコントロール

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](#)の需要要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

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?](#)」を参照してください。

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[AWS の ABAC とは](#)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

AWS リージョン 内の仕切られた場所は、他のアベイラビリティゾーンに障害が発生してもその影響を受けず、低コスト、低レイテンシーで同一リージョン内の他のアベイラビリティゾーンに接続できます。

AWS クラウド導入フレームワーク (AWS CAF)

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

AWS ワークロード資格フレームワーク (AWS WQF)

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

B

BCP

[「ビジネス継続性計画」](#)を参照してください。

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[Data in a behavior graph](#)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](#)も参照してください。

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、[「ブランチについて \(GitHub ドキュメント\)」](#)を参照してください。

ブレイククロスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが通常はアクセス許可AWS アカウントを持たないにアクセスするための迅速な手段を提供します。詳細については、AWS Well-Architected [ガイドンスの「Break Glass 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、ホワイトペーパー[AWS でのコンテナ化されたマイクロサービスの実行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。

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。

カオエンジニアリング

システムの耐障害性をテストするために、意図的に障害や破壊的なイベントを導入する。[AWS Fault Injection Service \(AWS FIS\)](#) を使用して、AWSワークロードに負荷を掛け、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

ターゲットの AWS のサービスが受け取る前に、データをローカルで暗号化すること。

Cloud Center of Excellence (CCoE)

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

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、エッジ [コンピューティング](#) テクノロジーに一般的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 Stephen Orban が AWS クラウドエンタープライズ戦略ブログの [クラウドファーストジャーニーと導入ステージ](#) というブログ記事で定義したものです。これらが、AWS 移行戦略とどのような関係があるかについては、[移行準備ガイド](#)を参照してください。

CMDB

[「設定管理データベース」](#)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 またはが含まれますAWS CodeCommit。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

画像内の人物、場所、物を人間と同等以上の精度で識別するために機械が使用するAIの分野です。多くの場合、深層学習モデルを使用して構築され、1 つの画像または一連の画像からの有用な情報の抽出、分析、分類、理解を自動化します。

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。

パフォーマンスパック

組み合わせることで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できる、AWS Config ルールと修復アクションのコレクション。パフォーマンスパックは、1つのエンティティとして AWS アカウント とリージョンに、または YAML テンプレートを使用して組織全体にデプロイできます。詳細については、AWS Config ドキュメントの[パ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。

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一要素です。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。AWS クラウド でデータ最小化を実践することで、プライバシーリスク、コスト、分析の二酸化炭素排出量を削減することができます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼できる ID のみが、期待されるネットワークから信頼できるリソースにアクセスしていることを確実にします。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。通常、データウェアハウスには大量の履歴データが含まれており、クエリや分析によく使用されます。

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る。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。

ディープラーニング

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。

defense-in-depth

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 AWS に採用すると、AWS Organizations 構造内の各層に複数のコントロールが追加され、リソースの安全を維持できます。例えば、defense-in-depth 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

AWS Organizations では、互換性のあるサービスは AWS メンバーアカウントを登録することで、組織のアカウントやそのサービスのアクセス許可を管理できるようになります。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 で使用できるサービス](#)を参照してください。

デプロイメント

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、Implementing security controls on AWSの[Detective controls](#)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](#) では、ファクトテーブル内の量子データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する離散数値です。これらの属性は、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よく使用されます。

ディザスタ

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。

での災害対策

[災害](#)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に使用する戦略とプロセス。詳細については、Well-Architected Framework ドキュメントの「Disaster recovery options in the cloud」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逸脱の追跡。例えば、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、AWS Control Towerを使用してガバナンス要件への準拠に影響を与える可能性のある[ランディングゾーンの変更を検出](#)したりできます。

DVSM

[「開発値ストリームマッピング」](#)を参照してください。

E

EDA

[「調査データ分析」](#)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、応答時間を短縮できます。

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プロセス。

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ず。

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。エンドポイントサービスは AWS PrivateLink を使って作成でき、アクセス許可を他の AWS アカウントまたは AWS Identity and Access Management (IAM) プリンシパルに付与することができます。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使用できる、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、アイデンティティとアクセスの管理、検出型制御、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。AWS 移行戦略のエピックの詳細については、[プログラム実装ガイド](#)を参照してください。

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

[星スキーマ](#) 内の中央テーブル。事業運営に関する量子データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 種類の列が含まれます。

フェイルファスト

頻繁な段階的テストを使用して開発ライフサイクルを短縮する哲学。これはアジャイルアプローチの重要な部分です。

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向上させる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、[「による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。

FGAC

[「きめ細かなアクセスコントロール」](#)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

段階的なアプローチを使用する代わりに、[変更データキャプチャ](#)を通じて継続的なデータレプリケーションを使用して、可能な限り短時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

G

Geo ブロック

[「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)

Amazon では CloudFront、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 ワークフローはレガシーと見なされ、[トランクベースのワークフロー](#)は最新の推奨アプローチです。

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施します。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、Amazon AWS Security Hub、GuardDuty、Amazon Inspector AWS Trusted Advisor、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

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) システムを近代化し、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。

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。その緊急性により、通常は一般的な DevOps リリースワークフローの外部で修正が行われます。

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している、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

「[産業分野におけるモノのインターネット](#)」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

既存のインフラストラクチャを更新、パッチ適用、または変更する代わりに、本番稼働用ワークロードに新しいインフラストラクチャをデプロイするモデル。イミュータブルなインフラストラ

クチャは、本質的に[ミュータブルなインフラストラクチャ](#)よりも一貫性、信頼性、および予測可能です。詳細については、AWS Well-Architected Framework の「[イミュータブルなインフラストラクチャのベストプラクティスを使用したデプロイ](#)」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内で、アプリケーション外部からの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。

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。

産業分野における 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、「[Building an industrial Internet of Things \(IIoT\) digital transformation strategy](#)」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内で、(同一または異なる AWS リージョンの) VPC、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 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。

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、「[AWS を使用した機械学習モデルの解釈](#)」を参照してください。

IoT

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

大規模な移行

300 台以上のサーバの移行。

LBAC

[「ラベルベースのアクセスコントロール」](#) 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。

マネージドサービス

AWS のサービスが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、プラットフォームをAWS運用し、ユーザーが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する対象になります。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象化されたサービスとも呼ばれます。

MAP

[「移行促進プログラム」](#)を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、結果を検査して調整を行う完全なプロセス。メカニズムとは、動作中に自身を強化および改善するサイクルです。詳細については、AWS Well-Architected フレームワークの「Building [mechanisms](#)」を参照してください。

メンバーアカウント

AWS Organizations の組織に含まれる管理アカウント以外の、すべての AWS アカウント。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。

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てマイクロサービスを統合する](#)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、[AWS でのマイクロサービスの実装](#)を参照してください。

Migration Acceleration Program (MAP)

組織がクラウドへの移行のための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP に

は、組織的な方法でレガシー移行を実行するための移行方法論と、一般的な移行シナリオを自動化および高速化する一連のツールが含まれています。

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、オペレーション、ビジネスアナリストと所有者、移行エンジニア、デベロッパー、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、ターゲットサブネット、セキュリティグループ、AWS アカウントが挙げられます。

移行パターン

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

Migration Portfolio Assessment (MPA)

AWS クラウドに移行するためのビジネスケースを検証するための情報を提供するオンラインツール。MPA は、詳細なポートフォリオ評価 (サーバ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、すべての人に無料で利用できる AWS コンサルタントと APN パートナーコンサルタントです。

移行準備状況評価 (MRA)

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

MPA

[「移行ポートフォリオ評価」](#) 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。

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。

変更可能なインフラストラクチャ

本番ワークロードの既存のインフラストラクチャを更新および変更するモデル。一貫性、信頼性、予測可能性を向上させるために、AWS Well-Architected フレームワークでは、[イミュータブルインフラストラクチャ](#)をベストプラクティスとして使用することをお勧めします。

O

OAC

[「オリジンアクセスコントロール」](#)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

[「組織の変更管理」](#)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

[「オペレーションの統合」](#)を参照してください。

OLA

[「運用レベルの契約」](#)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。

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 (TAK)

インシデントや障害の可能性の範囲を理解、評価、防止、または縮小するのに役立つ質問と関連ベストプラクティスのチェックリスト。詳細については、AWS Well-Architected フレームワークの「[運用準備状況レビュー \(TAK\)](#)」を参照してください。

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

AWS Organizations 内の一組織の、すべての AWS アカウント のイベントをすべてログ記録している AWS CloudTrail が作成した証跡。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、ドキュメントの「[組織の証跡の作成](#)」を参照してください。CloudTrail

組織変更管理 (OCM)

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

オリジンアクセスコントロール (OAC)

では CloudFront、Amazon Simple Storage Service (Amazon S3) コンテンツを保護するためにアクセスを制限するための拡張オプションです。OAC は、すべての AWS リージョン のすべての S3 バケット、AWS KMS (SSE-KMS) を使用したサーバー側の暗号化、S3 バケットへのダイナミックな PUT および DELETE リクエストをサポートしています。

オリジンアクセスアイデンティティ (OAI)

では CloudFront、Amazon S3 コンテンツを保護するためのアクセスを制限するオプションです。OAI を使用する場合は Amazon S3 が認証できるプリンシパル CloudFront を作成します。認証されたプリンシパルは、特定の CloudFront デイストリビューションを介してのみ S3 バケット内のコンテンツにアクセスできます。[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コントロールが可能です。

TAK

[「運用準備状況レビュー」](#)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、アプリケーションの内部から開始し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。

ポリシー

アクセス許可の定義 ([アイデンティティベースのポリシー](#) を参照)、アクセス条件の指定 ([リソースベースのポリシー](#) を参照)、または の組織内のすべてのアカウントに対する最大アクセス許可の定義 AWS Organizations ([サービスコントロールポリシー](#) を参照) が可能なオブジェクト。

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。詳細については、[マイクロサービスでのデータ永続性の有効化](#)を参照してください。

ポートフォリオ評価

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

述語

true または を返すクエリ条件。通常false、WHERE句にあります。

述語プッシュダウン

転送前にクエリ内のデータをフィルタリングするデータベースクエリ最適化手法。これにより、リレーショナルデータベースから取得および処理する必要があるデータの量が減少し、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、Implementing security controls on AWSの[Preventative controls](#)を参照してください。

プリンシパル

アクションを実行してリソースにアクセスできる AWS 内のエンティティです。このエンティティは、通常は AWS アカウント のルートユーザー、IAM ロール、ユーザーのいずれかになります。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](#)内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

エンジニアリングプロセス全体を通してプライバシーを考慮に入れた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

非準拠のリソースのデプロイを防ぐように設計された[セキュリティコントロール](#)。これらのコントロールは、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、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「[での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

本番環境

[「環境」](#)を参照してください。

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。

Q

クエリプラン

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る、手順などの一連のステップ。

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

[「責任、説明、相談、報告 \(RACI\)」](#)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

[「責任、説明、相談、報告 \(RACI\)」](#) を参照してください。

RCAC

[「行と列のアクセスコントロール」](#) 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。

アーキテクチャの再設計

[「7 Rs」](#) を参照してください。

目標復旧時点 (RPO)

RPO とは、データが最後に復旧した時点を開始とする経過時間の、許容される値のことです。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)

RTO とは、サービスが中断してから復旧するまでに経過した時間 (遅延) の、許容される値のことです。

リファクタリング

[「7 Rs」](#) を参照してください。

リージョン

地理的な領域内の 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、レジリエンスを実現するために他のリージョンと分離され、独立しています。詳細については、AWS 全般のリファレンスの [「Managing AWS リージョン」](#) を参照してください。

回帰

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。

プラットフォーム変更

「[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。

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。サポートを含めると、そのマトリックスは RASCI マトリックスと呼ばれ、サポートを除外すると 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、Implementing security controls on AWSの[Responsive controls](#)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。

ローテーション

定期的に[シークレット](#)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 (IdPs) が使用するオープンスタンダード。この機能ではフェデレーティッドシングルサインオン (SSO) が有効になるため、組織内の全員に IAM のユーザーを作成しなくても、ユーザーが AWS Management Console にログインしたり AWS API オペレーションを呼び出したりできます。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。

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

ではAWS Secrets Manager、暗号化された形式で保存する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された情報。シークレット値とそのメタデータで構成されます。シークレット値は、バイナリ、単一の文字列、または複数の文字列です。詳細については、[Secrets Manager](#) ドキュメントの「シークレット」を参照してください。

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、[主に予防的](#)、[検出的](#)、[応答性](#)、[プロアクティブ](#) の 4 つのタイプがあります。

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に自動的に応答または修正するように設計された、事前定義されたプログラムされたアクション。これらのオートメーションは、セキュリティのベストプラクティスの実装に役立つ [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。自動応答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、認証情報のローテーションなどがあります。

サーバー側の暗号化

データを受信した AWS のサービスによって、送信先でデータが暗号化されること。

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 \(SCP\)](#) を参照してください。

サービスエンドポイント

AWS のサービスのエン트리ポイントの URL。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、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベル目標 (SLO)

サービスレベルのインジケータによって測定される、サービスのヘルスを表すターゲットメトリクス。

責任共有モデル

ユーザーが、クラウドセキュリティとコンプライアンスに関する責任を AWS と共有するモデルのこと。AWS はクラウド自体のセキュリティに対して責任を負い、ユーザーはクラウド内のセキュリティに対して責任を負います。詳細については、[責任共有モデル](#)を参照してください。

SIEM

[「セキュリティ情報とイベント管理システム」](#)を参照してください。

単一障害点 (SPOF)

システムを中断させる可能性のあるアプリケーションの 1 つの重要なコンポーネントの障害。

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。

split-and-seed 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、[「」の「アプリケーションをモダナイズするための段階的なアプローチAWS クラウド」](#)を参照してください。

SPOF

[単一障害点](#)を参照してください。

star スキーマ

トランザクションデータまたは測定データを保存するために 1 つの大きなファクトテーブルを使用し、データ属性を保存するために 1 つ以上の小さなディメンションテーブルを使用するデータベースの組織構造。この構造は、[データウェアハウス](#)またはビジネスインテリジェンス目的で使用するよう設計されています。

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。

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

潜在的な問題を検出したり、パフォーマンスを監視したりするために、ユーザーインタラクションをシミュレートする方法でシステムをテストします。[Amazon CloudWatch Synthetics](#) を使用して、これらのテストを作成できます。

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「[Transit Gateway 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

AWS Organizations の組織およびそのアカウントで、ユーザーに代わって指定したサービスにタスクを実行させるためにアクセス許可を付与すること。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な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、AWS Organizations ドキュメントの[AWS Organizations を他の AWS サービスと併用する](#) 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つのピザを供給できる小規模な 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。詳細については、[深層学習システムにおける不確実性の定量化](#) ガイドを参照してください。

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

現在のレコードに関連する行のグループに対して計算を実行する SQL 関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなど、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 ワークロード認定フレームワーク」](#)を参照してください。

Write Once, Read Many (WORM)

データを 1 回書き込み、データの削除や変更を禁止するストレージモデル。認可されたユーザーは、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、[イミュータブルな](#)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](#) を利用する攻撃、通常はマルウェアです。

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。

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。