



AWS 大規模な移行の戦略とベストプラクティス

AWS 規範ガイド



AWS 規範ガイドンス: AWS 大規模な移行の戦略とベストプラクティス

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

序章	1
大規模な移行に関するガイド	1
範囲、戦略、タイムライン	3
スコープ — 何を移行しますか?	3
戦略 – 移行する理由	3
タイムライン – 移行はいつ完了する必要がありますか?	5
ベストプラクティス	6
People	6
エグゼクティブサポート	6
チームコラボレーションと所有権	7
トレーニング	9
テクノロジー	9
自動化、追跡、ツールの統合	10
前提条件と移行後の検証	12
プロセス	14
大規模な移行の準備	14
大規模な移行の実行	18
その他の考慮事項	22
結論	24
リソース	25
AWS 大規模な移行	25
関連する AWS 規範ガイドリソース	25
その他のリファレンス	25
動画	25
寄稿者	26
ドキュメント履歴	27
用語集	28
#	28
A	29
B	31
C	33
D	36
E	40
F	43

G	44
H	45
I	47
L	49
M	50
O	54
P	57
Q	60
R	60
S	63
T	67
U	68
V	69
W	69
Z	70
.....	lxxi

AWS 大規模な移行の戦略とベストプラクティス

Amazon Web Services ([寄稿者](#))

2022 年 5 月 ([ドキュメント履歴](#))

多くの AWS お客様は、ビジネスへの影響を最小限に抑えながら、多数のサーバーとアプリケーションを AWS クラウド できるだけ早く に移行したいと考えています。データセンターのリースが更新や終了に近づいているか、組織がテクノロジー変革の最初のステップを実行しているため、組織が大規模な移行プロジェクトを開始している可能性があります。ただし、大規模な は、範囲内のサーバーの数だけ定量化されません。また、人、プロセス、テクノロジー、優先順位などの複雑さを考慮して、移行によって生じる組織変革のレベルも考慮します。

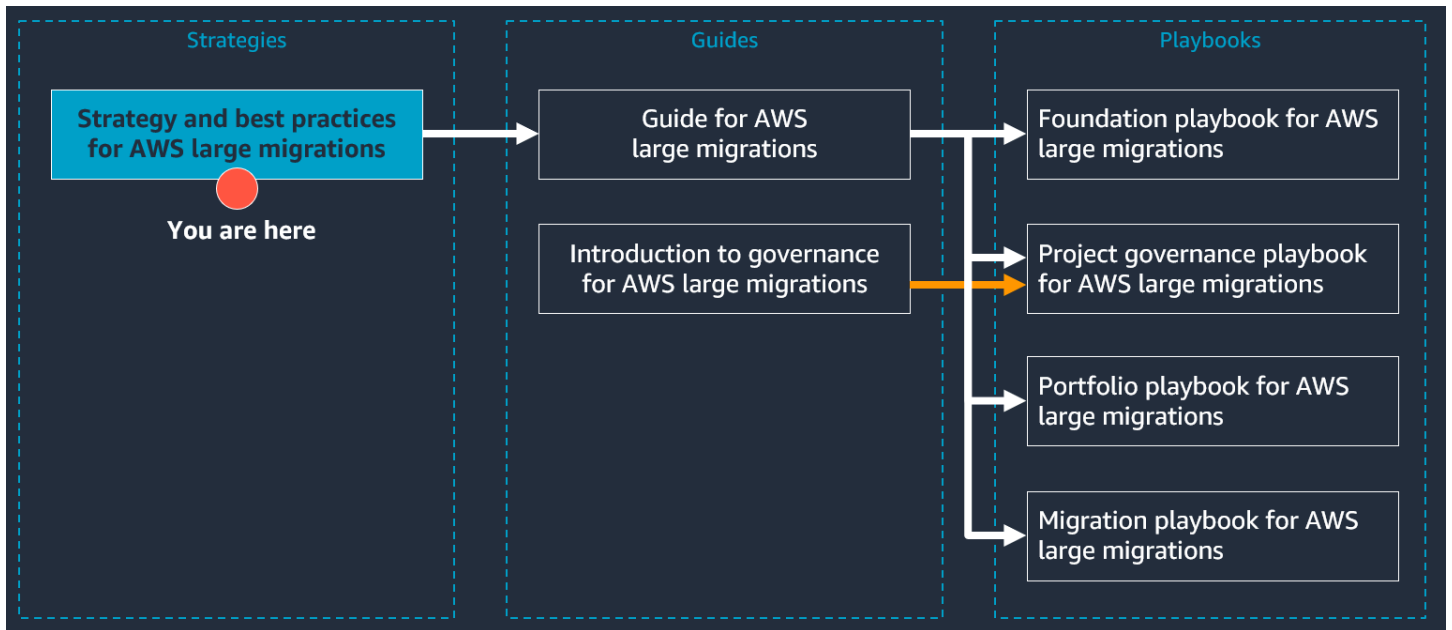
このガイドでは、大規模な移行能力に焦点を当てています AWS。既存のアプリケーションは、ほとんど、またはまったく変更せずに移行できます。クラウドを起動ポイントとして使用して、これらのアプリケーションをクラウドネイティブまたはサーバーレステクノロジーに移行できます。また、アプリケーションをモダナイズして、追加のビジネスメリットを引き出すことができます。

このガイドでは、大規模な移行のベストプラクティスについて説明し、金融サービスや医療など、さまざまなセグメントにわたる顧客からのユースケースを提供します。また、顧客の移行中に学んだ教訓の実例も示します AWS。このガイドの目的は、大規模な移行の初期段階にいるお客様を支援することです。ただし、このガイドのベストプラクティスと戦略は、移行ジャーニーのどの段階でも有益です。すでに に関する 100 レベルの知識があり、[AWS 移行の推奨プロセス](#)を理解していることを前提 AWS のサービス としています。

大規模な移行に関するガイド

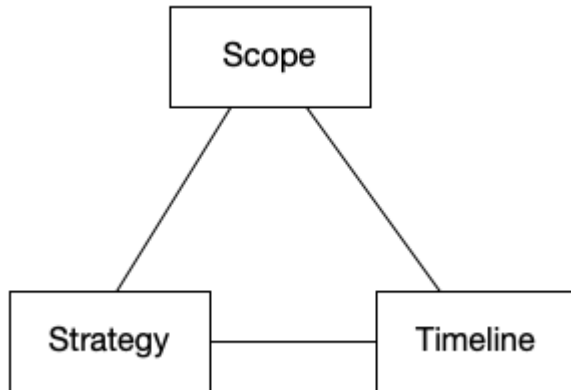
300 台以上のサーバを移行することは「大規模な移行」と見なされます。大規模な移行プロジェクトの人材、プロセス、テクノロジーの課題は、通常、ほとんどの企業にとって新しいものです。このドキュメントは、への大規模な移行に関する AWS 規範的ガイドシリーズの一部です AWS クラウド。このシリーズは、クラウドへの移行を合理化するために、最初から正しい戦略とベストプラクティスを適用するのに役立つように設計されています。

次の図は、このシリーズの他のドキュメントを示しています。最初に戦略を確認し、次にガイドを確認し、プレイブックに進みます。シリーズ全体にアクセスするには、[「への大規模な移行 AWS クラウド」](#)を参照してください。



範囲、戦略、タイムライン

すべてのプログラムの構成要素と大規模な移行における関連性を構成する 3 つの重要な要素は、スコープ、戦略、タイムラインです。



移行ジャーニーのステージを設定するには、移行プログラムの開始時からこれらの要素を調整して理解する必要があります。これらの要素のいずれかを変更すると、他の要素に影響します。再調整は、変更がどの程度基本的または賢明であるかにかかわらず、すべての変更を考慮する必要があります。

スコープ — 何を移行しますか？

移行の途中でも、プログラムの合計範囲が未定義になることが一般的です。これは、後のステージまでさまざまな要因が解凍されない可能性があるためです。たとえば、移行の途中で、設定管理データベース (CMDB) に記録されていないシャドウ IT のポケットを発見することがあります。または、計画では、これらのアプリケーションの実行に必要なサポートネットワークおよびセキュリティサービス (AWS パートナーへの VPN 接続、証明書に署名する認証機関など) を考慮せずに、サーバービューに焦点を当てている可能性があります。スコープの定義に少し時間を費やし、目標とするビジネス成果から逆算することをお勧めします。このガイドの後半で説明するベストプラクティスであるアセットを発見するために、検出ツールを使用することもあります。

大規模な移行には不明なものがあるため、スコープは変更されます。これらの未知は、関連性をほとんどまたはまったく理解せずに環境のアーキオロジーの一部となったシステム、または計画の遅延や移行を引き起こす本番環境のインシデントの形式である可能性があります。重要なのは、プログラムを前進させるための柔軟な対応と緊急時対応計画を整えることです。

戦略 — 移行する理由

次の AWS 1 つ以上の理由により、への移行を計画している場合があります。

- アプリケーションチームは、新しい CI/CD パイプラインを実装したり、最新のアプリケーションスタックをデプロイしたり、サポートされていないレガシープラットフォームをモダナイズしたりしたいと考えています。
- インフラストラクチャチームは、リースの有効期限が切れてプロバイダーが電源をオフにする前に、古いデータセンターからすぐに抜け出す必要があります。
- 取締役会は、戦略的方向性としてクラウドに移行する必要があると判断しました。これにより、ビジネスの未来における変化のペースが速くなります。

理由が何であれ、これらの理由などはすべて、ビジネス組織や IT 組織の頭の中にあります。ドライバーとは何かを理解し、伝達し、優先順位を付けることが重要です。ドライバーを追加するたびに、すでに大規模な移行に時間、コスト、範囲、リスクが加わる可能性があります。戦略がタイムラインとスコープに与える影響を完全に認識することが重要です。

移行戦略を定義した後、成功の鍵の 1 つは、さまざまな利害関係者やチームの要件の整合性です。移行を実行するには、インフラストラクチャ、セキュリティ、アプリケーション、オペレーションなど、組織全体のさまざまなチームが必要です。これらのチームには、個別の優先順位と、すでに開始されている可能性のある他のプロジェクトがあります。これらのチームがさまざまなタイムラインと優先順位に取り組んでいる場合、移行計画に同意して実装することはより困難です。移行チームと主要な利害関係者は、関係するすべてのチームが単一の目標に向かって作業し、優先順位を移行の単一のタイムラインに合わせる必要があります。

望ましいビジネス成果をさまざまなチーム間でどのように調整できるかを検討することをお勧めします。たとえば、に移行 AWS して AWS Key Management Service (AWS KMS) を使用して保管中のストレージを暗号化すると、移行とセキュリティの両方の目標を達成できます。

多くの場合、企業はアプリケーションをモダナイズしてインフラストラクチャをアップグレードしたいと考えていますが、インフラストラクチャチームはインフラストラクチャの変更を最小限に抑えたいと考えています。大規模な移行の考え方は、できるだけ基本的なものでなければなりません。関係するチームは、一度にすべてを実行しようとしないようにする必要があります。

これを実現するには、プロジェクトの早い段階で適切な期待を設定します。キーメッセージは「最初に移行してからモダナイズする」である必要があります。このアプローチにより、組織は技術的負債を減らし、最終的に大規模に運用できるだけでなく、が提供 AWS クラウド できるスケーラビリティと俊敏性を使用することで、さまざまなモダナイゼーションアプローチの道を開くことができます。長期検討は、インフラストラクチャチームがインフラストラクチャのデプロイと管理を合理化するのに役立ちます。その結果、ビジネスはより高速な機能リリースサイクルを実現できます。

タイムライン – 移行はいつ完了する必要がありますか？

ビジネスケースによっては、割り当てられた時間内に達成できる以上の作業を行っていないことを確認する必要があります。移行のドライバーが固定の完了日に基づいている場合は、そのタイムライン要件を満たす戦略を選択する必要があります。ほとんどの大規模な移行はこれらの時間ベースの制約に基づいているため、移行戦略は、拡張やオーバーランの余地をほとんど持たずに、定義された固定されたタイムラインと成果を持っている必要があります。

これらの時間的制約のあるタイプの移行では、「最初に移行してからモダナイズ」アプローチをお勧めします。これにより、期待値を設定し、チームが個々のプロジェクト計画と予算が移行目標全体と一致していることを確認することができます。プロジェクトでできるだけ早く意見の不一致を見つけ、迅速に失敗し、ステアリング委員会レベルで意見の不一致に対処し、適切な利害関係者を関与させて調整を整えることが重要です。

逆に、移行の主な目的がアプリケーションのモダナイゼーションのメリットを得ることである場合は、プログラムの早い段階でこれを呼び出す必要があります。多くのプログラムは、固定された期限に基づいて初期目標から始まり、未解決の問題や問題を解決したいステークホルダーの要件を計画していません。場合によっては、これらの問題はソースシステムに何年も存在していますが、現在は移行の人為的なブロック要因となっています。

移行中のモダナイゼーションアクティビティは、ビジネスアプリケーションの機能に影響を与える可能性があります。オペレーティングシステムのバージョン変更など、小規模なアップグレードと見なされる場合でも、プログラムのタイムラインに大きな影響を与える可能性があります。これらはささいなものと考えるべきではありません。

大規模な移行のベストプラクティス

組織の機能を管理する要因によっては、大規模な移行が困難になる可能性があります。このセクションでは、労力の初期段階で対処し、プロジェクト全体で追跡することで、大規模な移行を簡素化できるいくつかの主要な要因について説明します。

大規模な移行に関する以下のベストプラクティスは、他のお客様から取得したデータに基づいています。ベストプラクティスは 3 つのカテゴリに分かれています。

- People
- テクノロジー
- プロセス

人々の視点

このセクションでは、人的視点の以下の主要領域に焦点を当てます。

- エグゼクティブサポート – 意思決定を行う権限のあるシングルスレッドリーダーを特定する
- チームコラボレーションと所有権 – さまざまなチーム間のコラボレーション
- トレーニング – さまざまなツールに関するチームのプロアクティブトレーニング

エグゼクティブサポート

このセクションの内容:

- [シングルスレッドリーダーを特定する](#)
- [上級リーダーシップチームの調整](#)

シングルスレッドリーダーを特定する

大規模な移行を開始するときは、プロジェクトと説明責任に 100% 専念しているシングルスレッドのテクニカルリーダーを特定することが重要です。このリーダーは、一貫した優先順位を維持することで、意思決定を行い、サイロ化を回避し、ワークストリームを合理化することができます。

大規模な移行のグローバル顧客は、プログラムの開始時に毎週 1 台のサーバーから、2 か月目の開始時に毎週 80 台を超えるサーバーにスケールできました。シングルスレッドリーダーとしての CIO の

完全なサポートは、移行されるサーバーの急速なスケールアップにとって重要でした。CIO は、移行チームと毎週の移行カットオーバーコールに参加して、問題のリアルタイムのエスカレーションと解決を確保し、移行速度を加速しました。

上級リーダーシップチームの調整

移行の成功基準に関して、さまざまなチーム間で調整を行うことが重要です。移行の計画と実装は小規模な専有チームによって達成できますが、戦略を定義し、周辺機器のアクティビティを実行する際に課題が発生します。これらの潜在的な障害には、次のような IT 組織のさまざまな分野からのアクションやエスカレーションが必要になる場合があります。

- ビジネス
- アプリケーション
- ネットワーク
- セキュリティ
- インフラストラクチャ
- サードパーティーベンダー

アプリケーション所有者、リーダーシップ、調整、シングルスレッドリーダーへの明確なエスカレーションからの直接的なアクションが重要になります。

チームコラボレーションと所有権

このセクションの内容:

- [部門横断的なクラウド対応チームを作成する](#)
- [コア移行チーム外のチームや個人の要件を事前に定義する](#)
- [ワークロードの移行時にライセンスの問題がないことを検証する](#)

部門横断的なクラウド対応チームを作成する

大規模な移行プロジェクトの重要な最初のステップは、組織がクラウドで作業できるようにすることです。これを実現するには、[Cloud Enablement Engine](#) (CEE) を構築することをお勧めします。CEE は、移行のための組織の運用準備に重点を置いた、権限と説明責任を持つチームです AWS。CEE は、インフラストラクチャ、アプリケーション、運用、セキュリティからの代表者を含む部門横断的なチームである必要があります。チームには以下の責任があります。

- ポリシーの開発
- 組織のクラウド運用モデルを確立するツール、プロセス、アーキテクチャの定義と実装
- ステークホルダーが表すすべての領域でステークホルダーの連携を継続的に促進する

あるヘルスケアの顧客が CEE から始めませんでした。ただし、最初のパイロット移行により、ギャップが特定されました。最終移行カットオーバー日まで、厳しい期限が設定され、チームは移行の戦場を実装しました。移行の戦場では、インフラストラクチャ、セキュリティ、アプリケーション、ビジネスの関係者が問題の解決を支援できます。

コア移行チーム外のチームや個人の要件を事前に定義する

コアプログラム外のチームや個人を特定し、移行計画段階での関与を定義します。後の段階で移行の勢いを促進するには、アプリケーションチームの関与に特に注意してください。アプリケーションに関する知識、問題を診断する能力、カットオーバーにサインオフする要件が必要です。

移行はコアチームが主導しますが、アプリケーションチームはカットオーバー中の移行計画とテストの検証に関与している可能性があります。お客様は多くの場合、アプリケーション移行ではなくインフラストラクチャプロジェクトとしてクラウド移行にアプローチします。これにより、移行中に問題が発生する可能性があります。

移行戦略を選択するときは、アプリケーションチームに必要な関与を検討することをお勧めします。例えば、リホスト戦略では、リプラットフォーム戦略やリファクタリング戦略と比較して、アプリケーションとチームの関与が少なく済みます。リプラットフォーム戦略やリファクタリング戦略では、より多くのアプリケーション環境が変更されています。アプリケーション所有者の可用性が制限されている場合は、リファクタリング、再配置、再購入戦略ではなく、リホストまたはリプラットフォームの使用を検討してください。

ワークロードの移行時にライセンスの問題がないことを検証する

企業の既製の製品をクラウドに移行すると、ライセンスが変更される可能性があります。ライセンス契約は、オンプレミスの資産に焦点を当てている場合があります。たとえば、ライセンスは CPU によって、または特定の MAC アドレスにリンクされている場合があります。または、ライセンス契約には、パブリッククラウド環境でホストする権利が含まれていない場合があります。ただし、ベンダーとライセンスを再交渉すると、リードタイムが長くなり、移行が難しくなる可能性があります。

移行の範囲が定義されたら、すぐに調達チームまたはベンダー管理チームと協力することをお勧めします。ライセンスは、ターゲットアーキテクチャと移行パターンにも影響する可能性があります。

トレーニング

このセクションの内容:

- [新しいツールとプロセスについてチームをトレーニングする](#)

新しいツールとプロセスについてチームをトレーニングする

移行戦略が定義されたら、移行とターゲット運用モデルに必要なトレーニングを理解するために時間を費やしてください。移行中は、組織にとって AWS Database Migration Service 初めてのツールなどを使用する可能性があります。チームを積極的にトレーニングすることで、移行フェーズで発生する遅延を軽減できます。

実践的な方法でツールを試す機会を提供するアクティブな知識移転方法を探すことをお勧めします。例えば、AWS プロフェッショナルサービスは、大規模な移行を担当する 3 つのシステムインテグレーター (SI) AWS パートナー向けに、いくつかの Cloud Migration Factory トレーニングセッションを提供しました。これにより、チームは移行フェーズに移行する際に基本的な知識を持つことができました。また、各 SI AWS パートナーチーム内でファーストラインエスカレーションとして機能する可能性のある対象分野のエキスパート (SMEs) を特定するのにも役立ちました。

テクノロジーの視点

テクノロジーは、大規模な移行を加速するための優れた基盤を提供します。例えば、Cloud Migration Factory ソリューションは、移行に end-to-end 自動化を提供する方法に焦点を当てています。このセクションでは、スコープ、戦略、タイムラインに合わせて、必要なスケールと速度を達成するためにテクノロジーを使用するためのベストプラクティスをいくつか説明します。

包括的な原則は、可能な限り自動化の領域を確認することです。対象範囲内に何千ものサーバーがある場合、タスクを手動で実行すると、コストと時間のかかる作業になる可能性があります。

移行を実行するために、通常、次のようないくつかのツールが使用されます。

- 発見
- 移行の実装
- 設定管理データベース (CMDB)
- インベントリスプレッドシート
- プロジェクト管理

これらのツールは、評価から動員、実装まで、移行のさまざまな段階で使用されます。これらのツールの選択は、ビジネス目標とタイムラインによって決まります。

移行フェーズが計画されたら、次のステップは、移行チームが必要なツールを使用するスキルを持っていることを確認することです。チームにスキルや経験がない場合は、ターゲットを絞ったトレーニングを計画してスキルセットを強化します。可能であれば、チームが安全な環境で移行ツールの経験を積むことができるイベントを作成します。例えば、チームがツールを試すために移行できるサンドピットサーバーやラボサーバーはありますか？または、初期開発ワークロードを学習目的で使用することは可能ですか？

自動化、追跡、ツールの統合

このセクションの内容:

- [移行検出を自動化して必要な時間を短縮する](#)
- [反復タスクを自動化する](#)
- [追跡とレポートを自動化して意思決定を迅速化する](#)
- [移行を容易にするツールを調べる](#)

移行検出を自動化して必要な時間を短縮する

ほとんどの大規模な移行プログラムは、移行の範囲 (移行対象) を理解し、戦略 (移行方法) を策定することによって開始されます。検出は、この重要な側面です。必要なメタデータポイントは、移行戦略決定ツリーを形成するためにキャプチャされます。ワークロードをペースで移行するには、必要な移行メタデータを特定して、移行ファクトリーなどの実装プロセスにインポートする必要があります。移行メタデータを抽出、変換、ロード (ETL) する完全に自動化されたメカニズムにより、検出プロセスにかかる時間と労力が大幅に削減されます。

あるお客様は、移行ファクトリー用に完全に自動化されたデータ取り込みプロセスを開発しました。すべての移行メタデータを含む移行ウェブプランは、Microsoft SharePoint のスプレッドシートでホストされ、維持されました。ソースが変更されると、手動操作なしで移行ファクトリーにデータをロードする AWS Lambda 関数が開始されました。この自動データ取り込みプロセスにより、お客様は手動作業を減らし、人為的ミスをもっと抑え、スピードを加速できました。1,000 を超えるサーバーを に移行できました AWS。

反復タスクを自動化する

移行実装フェーズでは、多くの小規模なプロセスを頻繁に繰り返す必要があります。AWS Application Migration Service (MGN) を使用する場合、例えば、移行の対象となる各サーバーにエージェントをインストールする必要があります。

特定のビジネス要件と技術要件に適した移行ファクトリーを構築することは、大規模な移行を成功させるために必要な効率と速度を実現する最も効果的な方法です。移行ファクトリーは、標準化されたデータセットを使用して移行を高速化する統合およびオーケストレーションフレームワークを提供します。すべてのタスクを特定したら、規範的なランブックとともに自動化できるすべての手動タスクを自動化する時間を取ります。

[Cloud Migration Factory](#) ソリューションがその一例です。Cloud Migration Factory は、組織に固有の側面を自動化できる移行自動化基盤を提供するように設計されています。たとえば、CMDB のフラグを更新して、オンプレミスサーバーを廃止できることを強調できます。このシナリオでは、移行ウェブの最後にこのタスクを実行するオートメーションを作成できます。Cloud Migration Factory には、すべてのウェブ、アプリケーション、サーバーメタデータを含む一元化されたメタデータストアがあります。自動化スクリプトは Cloud Migration Factory に接続して、そのウェブ内のサーバーのリストを取得し、それに応じてアクションを実行できます。Cloud Migration Factory は [AWS Application Migration Service](#) をサポートしています。

追跡とレポートを自動化して意思決定を迅速化する

自動移行レポートダッシュボードを構築して、プログラムの重要業績評価指標 (KPIs) を含むライブデータを追跡およびレポートすることをお勧めします。移行プロジェクトには、以下を含む組織全体のステークホルダーが参加します。

- アプリケーションチーム
- テスター
- チームの廃止
- アーキテクト
- インフラストラクチャチーム
- リーダーシップ

ロールを実行するには、これらのステークホルダーにライブデータが必要です。例えば、オンプレミスリソースと 間の共有接続への影響を理解するには、ネットワークチームが今後の移行ウェブを

把握する必要があります AWS。リーダーシップチームは、移行の完了度を把握したいと考えています。信頼できる自動ライブフィードにより、ミスコミュニケーションが防止され、意思決定の基盤が提供されます。

大規模なヘルスケアのお客様が、データセンターの閉鎖に向けて、期限が近づいています。規模と複雑さを考慮すると、当初はステークホルダー間の移行ステータスの追跡と伝達にかなりの時間が費やされていました。その後、移行チームは [Amazon Quick Sight](#) を使用してデータを視覚化する自動ダッシュボードを構築し、移行速度を向上させながら追跡と通信を大幅に簡素化しました。

移行を容易にするツールを調べる

特に組織内の誰も大規模な移行を管理したことがない場合、移行に適したツールを選択することは簡単ではありません。

移行をサポートする適切なツールを選択する時間を取ることをお勧めします。この調査にはライセンスコストが伴う場合がありますが、より広範なイニシアチブを検討するとコスト上のメリットが得られます。または、組織に埋め込まれたツールでも同様の結果が得られる場合があります。たとえば、エステート全体にアプリケーションパフォーマンスモニタリングツールが既にデプロイされているため、豊富な検出情報を提供できます。

テクノロジーのお客様は、最初は知識不足のため、移行中に自動検出ツールの実行に消極的でした。その結果、SI AWS パートナーは、サーバー名、オペレーティングシステムのバージョン、依存関係など、エステートを手動で検出するために、アプリケーションごとに 5~10 時間の会議を実行する必要があります。検出ツールが使用されていた場合、検出作業は 1,000 時間以上短縮された可能性がありますと推定されました。

前提条件と移行後の検証

このセクションの内容:

- [移行前段階でランディングゾーンを構築する](#)
- [前提条件アクティビティの概要](#)
- [継続的な改善のために移行後のチェックを実装する](#)

移行前段階でランディングゾーンを構築する

移行ウェーブ中に AWS ターゲット仮想プライベートクラウド (VPCs) とサブネットを構築するのではなく、事前にターゲット環境またはランディングゾーンを構築することをお勧めします。適切に設

計されたランディングゾーンを構築することは、移行の前提条件です。ランディングゾーンには、モニタリング、ガバナンス、運用、セキュリティコントロールを含める必要があります。

移行前にランディングゾーンを構築して検証することで、新しい環境でワークロードを実行する際の不確実性を最小限に抑えることができます。ランディングゾーンを設定すると、ステークホルダーはアカウントまたは VPC レベルで管理される側面について心配することなく、ワークロードの移行に集中できます。

前提条件アクティビティの概要

ランディングゾーンに加えて、移行前に他の技術的前提条件、特にリードタイムが長いプロセスを調整することが重要です。たとえば、オンプレミスからデータをレプリケートするために必要なファイアウォールの変更を行います AWS。技術的な前提条件を早期に伝えることは、必要なリソースの準備と割り当てに役立ちます。前提条件が満たされていないため、移行が停止することが一般的です。これは進行中の移行の波に影響を与えるだけでなく、問題が修正されている間、将来のすべての移行の日付をプッシュバックする可能性があります。

複数のデータセンターを退避することを目指し AWSで、への一括移行を実行することを目的とした金融サービス会社。ただし、オンプレミスとの間で利用可能な帯域幅は、意図した速度では不十分 AWSでした。残念ながら、帯域幅を増やすには新しい接続が必要で、リードタイムは 3 か月でした。これは、移行速度が最初の 3 か月間制約されたことを意味します。

継続的な改善のために移行後のチェックを実装する

最後に、オペレーション統合、コスト最適化、ガバナンスとコンプライアンスチェックなどの移行後の検証を必ず実装してください。移行後の検証には、以前に移行されたワークロードを評価して、将来の波に適用すべき技術的教訓を明らかにすることが含まれます。

さらに、これはコスト管理オペレーションを実装する優れた機会です。たとえば、移行中に、パフォーマンステストの必要性を減らすために AWS、インスタンスのサイズをオンプレミスの資産と一致させることにしました。データセンター閉鎖のクリティカルパスでテストが行われなくなったので、Amazon CloudWatch を使用してインスタンス使用率を評価し、より小さいサイズのインスタンスが適切かどうかを判断できます。

このフェーズの重要性を説明するために、ある大規模なテクノロジーの顧客が大規模な移行を実行していましたが、当初は移行後の検証が含まれていませんでした。100 を超えるサーバーを移行した後、AWS Systems Manager エージェント (SSM Agent) が正しく設定されていないことがわかりました。以前に移行したすべてのサーバーを修正する必要があり、移行が停止しました。顧客は、インスタンスが最初の見積もりの 5 倍の大きさであることも特定したため、各移行ウェーブの最後にコストチェックポイントを実装しました。

プロセスの視点

プロセスは一貫性をもたらしますが、各プロジェクトが一意であるため、進化し、変更の影響を受けやすくなります。プロセスを繰り返し実行すると、障害発生時、学習時、導入時、反復時に大きなメリットをもたらす可能性のある改善のギャップと余地を特定できます。これらの変更は、プロジェクトとビジネスが将来活用できる新しいアイデアやイノベーションにつながる可能性があり、品質とチームの自信をもたらす成長の促進要因となります。

移行プロセスは、以前にリンクされていない可能性のあるテクノロジーや境界を越えるため、複雑になる可能性があります。この視点は、大規模な移行の特定の要件に関するプロセスとガイダンスを提供します。

大規模な移行の準備

以下のセクションでは、移行ジャーニーを明確な方向性で開始し、成功に不可欠なステークホルダーからの賛同を得るために必要な基本原則の概要を説明します。

このセクションの内容:

- [ビジネスドライバーを定義し、タイムライン、範囲、戦略を伝える](#)
- [ブロッカーの削除に役立つ明確なエスカレーションパスを定義する](#)
- [不要な変更を最小限に抑える](#)
- [end-to-endプロセスを早期に文書化する](#)
- [標準の移行パターンとアーティファクトを文書化する](#)
- [移行メタデータとステータスの信頼できる単一のソースを確立する](#)

ビジネスドライバーを定義し、タイムライン、範囲、戦略を伝える

大規模な移行に近づくと AWS、サーバーを移行する方法が多数あることがすぐにわかります。例えば、次の操作を実行できます。

- を使用してワークロードをリホストします [AWS Application Migration Service](#)。
- アプリケーションをコンテナ化し、[Amazon Elastic Container Service](#) (Amazon ECS) または [Amazon Elastic Kubernetes Service](#) (Amazon EKS) マネージドコンテナプラットフォームでホストします。
- ワークロードを完全なサーバーレスアプリケーションに再設計します。

正しい移行パスを決定するには、ビジネスドライバーから逆算することが重要です。最終的な目標がビジネスの俊敏性の向上である場合は、2番目の2つのパターンを優先できます。これには、より多くのレベルの変換が含まれます。年末までにデータセンターを退避することを目標とする場合は、リホストの速度によってワークロードをリホストすることを選択できます。

大規模な移行には、通常、次のような幅広い利害関係者が関係します。

- アプリ所有者
- ネットワークチーム
- データベース管理者
- エグゼクティブスポンサー

移行のビジネス推進要因を特定し、移行プログラムのメンバーがアクセスできるプロジェクト憲章など、そのリストをドキュメントに含めることが重要です。さらに、目標とするビジネス成果に密接に一致する主要業績評価指標 (KPIs) を作成します。

たとえば、ある顧客が、データセンターを退避するという目標ビジネス成果を達成するために、12か月以内に2,000台のサーバーを移行したいと考えています。ただし、セキュリティチームはこの目標に向けて調整されていませんでした。その結果、データセンターの閉鎖日を逃したが、アプリケーションをさらにモダナイズするか、最初にリホストしてタイムリーなデータセンターの閉鎖を可能にし、アプリケーションをモダナイズするかについて、数か月にわたる技術的な議論が行われました AWS。

ブロッカーの削除に役立つ明確なエスカレーションパスを定義する

大規模なクラウド移行プログラムには、通常、幅広い利害関係者が参加します。結局のところ、数十年にわたってオンプレミスでホストされているアプリケーションが変更される可能性があります。各利害関係者が競合する優先順位を持つことが一般的です。

すべての優先順位が価値をもたらす可能性がありますが、プログラムは予算の量が制限され、目標成果が定義されます。さまざまなステークホルダーを管理し、目標とするビジネス成果に焦点を当てるのは難しい場合があります。この課題は、移行の対象となる数百または数千のアプリケーションに乘算すると複雑になります。さらに、ステークホルダーは、他の優先事項を持つさまざまなリーダーシップチームに報告する可能性があります。これを念頭に置いて、目標とするビジネス成果を明確に文書化するとともに、明確なエスカレーションマトリックスを定義してブロッカーを排除することが重要です。これにより、かなりの時間を節約し、さまざまなチームを共通の目標に合わせることができます。

この例の 1 つは、12 か月以内にプライマリデータセンターを退避することを目標とする金融サービス会社です。明確なマンデートやエスカレーションパスがなかったため、時間と予算の制約に関係なく、ステークホルダーが希望する移行パスを作成しました。CIO へのエスカレーション後、明確な義務が設定され、必要な決定をリクエストするためのメカニズムが提供されました。

不要な変更を最小限に抑える

変化は良好ですが、変化が多いほどリスクが高くなります。大規模な移行のビジネスケースが承認されると、特定の日付までにデータセンターを退避するなど、このイニシアチブを推進する目標とするビジネス成果が生じる可能性が高くなります。技術者が AWS サービスを最大限に活用するためにすべてを書き換えることは一般的ですが、これはビジネス目標ではないかもしれません。

ある顧客は、会社のウェブスケールインフラストラクチャ全体を 2 年間移行することに重点を置いていました AWS。アプリケーションチームがアプリケーションを書き換えて数か月を費やすことを防ぐためのメカニズムとして、2 週間のルールを作成しました。2 週間のルールを使用することで、顧客は何百ものアプリケーションを複数年にわたって移動する必要があったとき、一貫した頻度で長期的な移行を維持できました。詳細については、ブログ記事「[The Two-week Rule: Refactor Your Applications for the Cloud in 10 Days](#)」を参照してください。

ビジネス成果と一致しない変更は最小限に抑えることをお勧めします。代わりに、将来のプロジェクトでこれらの追加変更を管理するメカニズムを構築します。

end-to-end プロセスを早期に文書化する

大規模な移行プログラムの初期段階で、移行プロセス全体と所有権の割り当てを文書化します。このドキュメントは、移行の実行方法とその役割と責任についてすべての利害関係者を教育するために重要です。このドキュメントは、問題が発生する可能性のある場所を理解し、移行を進めるにつれてプロセスの更新と反復を提供するのに役立ちます。

移行プロジェクトの開発中に、既存のプロセスを理解し、統合ポイントと依存関係が明確に文書化されていることを確認します。変更リクエスト、サービスリクエスト、ベンダーサポート、ネットワークとファイアウォールのサポートなど、外部プロセス所有者とのエンゲージメントが必要な場所を含めます。プロセスを理解したら、責任、説明責任、相談、情報 (RACI) マトリックスに所有権を文書化して、さまざまなアクティビティを追跡することをお勧めします。プロセスを確定するには、移行の各ステップに関連するタイムラインを特定してカウントダウン計画を作成します。カウントダウンプランは通常、ワークロード移行のカットオーバー日時から逆算して機能します。

このドキュメントアプローチは、1 年未満で AWS 正常に に移行し、4 つのデータセンターを終了した多国籍のホームアプライアンス企業にとってうまく機能しました。6 つの異なる組織チームと複数

のサードパーティーが関係していたため、管理オーバーヘッドが発生し、意思決定がback-and-forthし、実装が遅れました。AWS プロフェッショナルサービスチームは、顧客とそのサードパーティーと協力して、移行アクティビティの主要なプロセスを特定し、それぞれの所有者に文書化しました。結果として得られた RACI マトリックスは、関係するすべてのチームによって共有され、合意されました。RACI マトリックスとエスカレーションマトリックスを使用して、お客様は遅延の原因となっていたブロック要因と問題を軽減しました。その後、データセンターをスケジュールどおりに終了できました。

RACI とエスカレーションマトリックスを使用する別の例では、保険会社は 4 か月以内にデータセンターを出ることができました。お客様は責任共有モデルを理解して実装し、詳細な RACI マトリックスに従って、移行中の各プロセスとアクティビティの進行状況を追跡しました。その結果、実装の最初の 12 週間で 350 を超えるサーバーを移行できました。

標準の移行パターンとアーティファクトを文書化する

これは、実装用の Cookie カッターを作成することを前提としています。再利用可能なリファレンス、ドキュメント、ランブック、パターンがスケーリングの鍵です。これらのジャーナルには、将来の移行プロジェクトが再利用して回避できる経験、学習、落とし穴、問題、ソリューションが記されており、移行を大幅に加速します。パターンとアーティファクトは、プロセスを改善し、将来のプロジェクトを導くのに役立つ投資でもあります。

たとえば、ある顧客が、3 つの異なる SI AWS パートナーによってアプリケーションが移行される 1 年間の移行を実行していました。初期段階では、各 AWS パートナーは独自の標準、ランブック、アーティファクトを使用していました。これにより、同じ情報がさまざまな方法で顧客チームに提示される可能性があるため、顧客チームに多くのストレスがかかりました。このような初期の苦労の後、顧客は移行で使用するすべてのドキュメントとアーティファクトの一元的な所有権を確立し、推奨される変更を送信するプロセスを確立しました。これらのアセットには以下が含まれます。

- 標準の移行プロセスとチェックリスト
- ネットワーク図のスタイルと形式の標準
- ビジネス重要度に基づくアプリケーションアーキテクチャおよびセキュリティ標準

さらに、これらのドキュメントと標準への変更は毎週すべてのチームに送信され、各パートナーは変更の受信と準拠を確認する必要がありました。これにより、移行プロジェクトのコミュニケーションと一貫性が大幅に向上し、別のビジネスユニットで別の大規模な移行作業が開始されると、そのチームは既存のプロセスとドキュメントを採用でき、成功が大幅に加速しました。

移行メタデータとステータスの信頼できる単一のソースを確立する

大規模な移行を計画する場合、さまざまなチームを連携させ、データ駆動型意思決定を可能にするには、信頼できる情報源を確立することが重要です。このジャーニーを開始すると、設定管理データベース (CMDB)、アプリケーションパフォーマンスモニタリングツール、インベントリリストなど、使用できる多数のデータソースが見つかる場合があります。

または、データソースが少なく、必要なデータをキャプチャするメカニズムを作成する必要があります。例えば、検出ツールを使用して技術情報を明らかにし、IT リーダーを調査してビジネス情報を取得する必要がある場合があります。

移行に使用できる 1 つのデータセットにさまざまなデータソースを集約することが重要です。その後、単一の信頼できるソースを使用して、実装中の移行を追跡できます。たとえば、移行されたサーバーを追跡できます。

提供されたデータセットを使用した移行の計画に重点を置いて、AWS すべてのワークロードを移行したいと考えている金融サービスの顧客。このデータセットにはビジネス重要度や依存関係情報などの重要なギャップがあったため、プログラムは検出演習を開始しました。

別の例では、同じ業界の企業が、サーバーインフラストラクチャインベントリout-of-date理解に基づいて移行ウェブの実装に移行しました。データが正しくなかったため、すぐに移行数が減少し始めました。この場合、アプリケーション所有者は理解されなかったため、テスターを間に合うように見つけることができませんでした。さらに、データはアプリケーションチームが完了した廃止と一致していないため、サーバーはビジネス目的で使用されずに実行されていました。

大規模な移行の実行

ビジネス成果を確立し、ステークホルダーに戦略を伝えたら、大規模な移行の範囲を持続可能な移行イベントや波に取り込む方法の計画に進むことができます。次の例は、ウェブプランを作成するための主要なガイドランスを提供します。

このセクションの内容:

- [安定したフローを確保するために、移行ウェブを事前に計画する](#)
- [ウェブ実装とウェブ計画を個別のプロセスとチームとして維持する](#)
- [大きな成果を得るために小規模から始める](#)
- [カットオーバーウィンドウの数を最小限に抑える](#)
- [フェイルファスト、エクスペリエンスの適用、反復](#)
- [遡及情報を忘れないでください。](#)

安定したフローを確保するために、移行ウェーブを事前に計画する

移行の計画は、プログラムの最も重要なフェーズの 1 つです。「計画に失敗した場合、失敗する予定」と書かれています。移行ウェーブを事前に計画しておくことで、チームが移行状況に対してより積極的になるにつれて、プロジェクトが迅速に流れるようになります。これにより、プロジェクトのスケーリングが容易になり、プロジェクトの需要が増加し、複雑になるにつれて意思決定と予測が向上します。前もって計画することで、チームが変化に適応する能力も向上します。

例えば、ある大規模な金融サービスの顧客がデータセンターの出口プログラムに取り組んでいました。当初、お客様は移行ウェーブを順番に計画し、1 つのウェーブを完了してから次のウェーブの計画を開始します。このアプローチにより、準備時間が短縮されました。利害関係者がアプリケーションが移行されていることを知らされても AWS、移行を開始する前にいくつかのステップを実行する必要があります。これにより、プログラムに大幅な遅延が発生しました。顧客がこれを認識した後、移行ウェーブが数か月前に計画された、包括的で未来に焦点を当てた移行計画ストリームを実装しました。これにより、アプリケーションチームは、AWS パートナーへの通知、ライセンス分析などの移行前アクティビティを実行するのに十分な通知が提供されました。その後、プログラムの重要なパスからこれらのタスクを削除できます。

ウェーブ実装とウェーブ計画を個別のプロセスとチームとして維持する

ウェーブプランニングチームとウェーブ実装チームが別々の場合、2 つのプロセスは並行して機能します。通信と調整では、十分なサーバーやアプリケーションが期待速度を達成する準備ができていないため、移行が遅くなるのを回避できます。例えば、移行チームは毎週 30 台のサーバーを移行する必要があるかもしれませんが、現在のウェーブでは 10 台のサーバーしか準備ができていません。この課題は、多くの場合、以下が原因で発生します。

- 移行実装チームはウェーブプランニングに関与しておらず、ウェーブプランニングフェーズで収集されたデータも完了していません。移行実装チームは、ウェーブを開始する前により多くのサーバーデータを収集する必要があります。
- 移行実装は、ウェーブ計画の直後に開始される予定であり、間にバッファはありません。

事前にウェーブを計画し、準備とウェーブ実装の開始の間にバッファを作成することが重要です。また、ウェーブプランニングチームと移行チームが連携して適切なデータを収集し、再作業を避けることも重要です。

大きな成果を得るために小規模から始める

小規模から始めて、その後のウェーブごとに移行速度を上げることを計画します。初期ウェーブは、10 台未満の単一の小さなアプリケーションである必要があります。後続のウェーブにアプリ

ケーションとサーバーを追加し、移行速度を最大限に向上させます。複雑性やリスクの低いアプリケーションを優先し、スケジュールに従って速度を上げることで、チームは連携して作業し、プロセスを学習する時間を確保できます。さらに、チームはウェーブごとにプロセスの改善を特定して実装できるため、後のウェーブの速度を大幅に向上させることができます。

あるお客様は、年間 1,300 台を超えるサーバーを移行していました。パイロット移行といくつかの小さなウェーブから始めることで、移行チームは後の移行を改善する複数の方法を特定できました。例えば、以前に新しいデータセンターネットワークセグメントを特定しました。プロセスの早い段階でファイアウォールチームと協力して、移行ツールとの通信を可能にするファイアウォールルールを設定しました。これにより、将来の波での不要な遅延を防ぐことができます。さらに、チームは、各ウェーブで検出プロセスとカットオーバープロセスをさらに自動化するスクリプトを開発することができました。小規模から始めることで、チームは初期のプロセス改善に集中し、自信が大幅に高まりました。

カットオーバーウィンドウの数を最小限に抑える

大規模な移行には、スケールを推進するための統制のとれたアプローチが必要です。一部の領域で柔軟性が高すぎることは、大規模な移行にはアンチパターンです。毎週のカットオーバーウィンドウの数を制限することで、カットオーバーアクティビティに費やされる時間が高くなります。

たとえば、カットオーバーウィンドウの柔軟性が高すぎる場合、それぞれ 5 台のサーバーで 20 回のカットオーバーが発生する可能性があります。代わりに、それぞれ 50 台のサーバーで 2 回のカットオーバーを行うことができます。各カットオーバーの時間と労力は似ているため、カットオーバーが少なくても大きいほど、スケジュールリングの運用上の負担が軽減され、不要な遅延が制限されます。

ある大規模なテクノロジー企業が、契約の有効期限が切れる前にいくつかのリースデータセンターから移行しようとしていました。有効期限が切れると、費用がかかり、短期的な更新期間が発生します。移行の前半では、アプリケーションチームは、カットオーバーのわずか数日前に何らかの理由で移行をオプトアウトするなど、過去 1 分間の移行スケジュールを指示することができました。これにより、プロジェクトの初期段階で多数の遅延が発生しました。多くの場合、顧客は過去 1 分間に他のアプリケーションチームと交渉して入力する必要がありました。顧客は最終的に計画の規律を高めました。この初期のミスは移行チームに継続的なストレスをもたらしました。スケジュール全体の遅延により、一部のアプリケーションがデータセンターから間に合わなくなりました。

フェイルファースト、エクスペリエンスの適用、反復

すべての移行には、最初は落とし穴があります。早期失敗は、チームが学習し、ボトルネックを理解し、学んだ教訓をより大きな波に適用するのに役立ちます。移行の最初の数波は、次の理由で遅くなることが予想されます。

- チームメンバーはお互いとプロセスに適応しています。
- 大規模な移行には、通常、さまざまなツールや人が関係します。
- end-to-endのプロセスの統合、テスト、失敗、学習、継続的な改善には時間がかかります。

問題は一般的であり、最初の数回のウェーブ中に予期されます。一部のチームは新しいことを試して失敗することを好まない可能性があるため、これを理解し、組織全体に伝えることが重要です。失敗すると、チームが失望し、将来の移行の妨げになる可能性があります。最初の問題がジョブの一部であることを全員が理解し、全員が失敗するように促すことが、移行を成功させるための鍵となります。

ある会社では、24～36 か月で 10,000 台を超える サーバーを移行する予定です。この目標を達成するには、1 か月あたり約 300 台のサーバーを移行する必要がありました。ただし、1 日目から 300 台のサーバーを移行したわけではありません。最初の 2 つの波はウェーブを学習することでした。これにより、チームは物事がどのように機能し、誰が何を行う権限を持っているかを理解できます。また、CMDB や CyberArk との統合など、プロセスを改善する統合も特定しました。学習ウェーブを使用して失敗、改善、失敗し直し、プロセスと自動化を改善しました。6 か月後、毎週 120 を超えるサーバーを移行できました。

遡及情報を忘れないでください。

これはアジャイルプロセスの重要な部分です。チームがコミュニケーション、調整、学習、同意、前進する場所です。最も基本的なレベルでの遡及は、振り返って何が起こったのかを議論し、何がうまくいき、何を改善する必要があるのかを判断することです。その後、それらの議論に基づいて改善を構築できます。遡及的思考は、教訓を学ぶという考え方に形式やプロセスを巻き付けます。大規模な移行を成功させるには、プロセス、ツール、チームが常に進化し、改善する必要があるため、遡及性は重要です。遡及的分析は其中で重要な役割を果たします。

従来の教訓学習セッションはプログラムの最後まで行われなため、多くの場合、これらの教訓は次の移行ウェーブの開始時にレビューされません。大規模な移行では、学んだ教訓を次のウェーブに適用し、ウェーブ計画プロセスの重要な部分とする必要があります。

ある顧客については、カットオーバーから学んだ教訓について議論し、文書化するために、毎週の遡及調査が行われました。これらのセッションでは、プロセスの観点や自動化から合理化の範囲があ

る領域を発見しました。これにより、カットオーバー中のサードパーティーツールの検証や Amazon CloudWatch エージェントのインストールなど、手動タスクを最小限に抑えるために、特定のアクティビティ、所有者、自動化スクリプトを含むカウントダウンスケジュールが実装されました。

別の大規模なテクノロジー企業では、以前の移行ウェブの問題を特定するために、チームと定期的な遡及調査を実施しました。これにより、プロセス、スクリプト、自動化が改善され、プログラム全体で平均移行時間が 40% 短縮されました。

その他の考慮事項

多くの分野を大規模な移行プログラムに組み込む必要があります。以下のセクションでは、考慮すべき他の項目について説明します。

このセクションの内容:

- [移動中にクリーンアップする](#)
- [追加の変換のために複数のフェーズを実装する](#)

移動中にクリーンアップする

コストが予想の 10 倍で、移行に使用されるリソースがシャットダウンされてクリーンアップされるまでプロジェクトが完了していない場合、移行は成功とは見なされません。このクリーンアップは、移行後のアクティビティの一部である必要があります。これにより、未使用のリソースやサービスが環境内に残ってコストが増加することがなくなります。移行後のクリーンアップは、環境を公開する脅威や脆弱性を防ぐためにも優れたセキュリティプラクティスです。

への移行の主な成果 AWS クラウド は、コスト削減とセキュリティの 2 つです。未使用のリソースを残すと、クラウドへの移行というビジネス目的が損なわれる可能性があります。クリーンアップされていない最も一般的なリソースは次のとおりです。

- テストデータ
- データベースをテストする
- ファイアウォールルール、セキュリティグループ、ネットワークアクセスコントロールリスト (ネットワーク ACL) IP アドレスを含むアカウントのテスト
- テスト用にプロビジョニングされたポート
- Amazon Elastic Block Store (Amazon EBS) ボリューム
- スナップショット
- レプリケーション (オンプレミスから へのデータレプリケーションの停止など AWS)

- スペースを消費するファイル (移行に使用される一時データベースバックアップなど)
- 移行ツールをホストするインスタンス

不正なクリーンアッププラクティスの一例では、SI AWS パートナーは移行が成功した後にレプリケーションエージェントを削除しませんでした。AWS 監査では、すでに移行されていたレプリケーションサーバーと EBS ボリュームのコストは毎月 20,000 USD (USD) であることがわかりました。この問題を軽減するために、AWS Professional Services は、古いサーバーがまだレプリケートされているときに SI AWS パートナーに通知する自動監査プロセスを作成しました。その後、SI AWS パートナーは未使用の古いインスタンスに対してアクションを実行できます。

今後の移行では、移行後のハイパーケア期間として 48 時間を定義し、プラットフォームのスムーズな導入を保証するプロセスを採用しました。次に、お客様のインフラストラクチャチームがオンプレミスサーバーに対する廃止リクエストを送信しました。廃止リクエストが承認されると、それぞれのウェブのサーバーがアプリケーション移行サービスコンソールから削除されることが通知されました。

追加の変換のために複数のフェーズを実装する

大規模な移行を実行するときは、データセンターの閉鎖やインフラストラクチャの変革など、中心的な目標に集中し続けることが重要です。小規模な移行では、スコープのクリープの影響は最小限に抑えられます。ただし、数日の追加作業に数千のサーバーを掛けると、プログラムにかなりの時間がかかる可能性があります。さらに、追加の変更により、サポートチームのドキュメント、プロセス、トレーニングの更新が必要になる場合もあります。

潜在的なスコープクリープを克服するには、移行に複数フェーズのアプローチを実装できます。例えば、データセンターを退避することが目標である場合、フェーズ 1 にはワークロードをできるだけ AWS 速くリホストすることのみが含まれる場合があります。ワークロードがリホストされると、フェーズ 2 はターゲットのビジネス成果を危険にさらすことなく、変革的なアクティビティを実装できます。

たとえば、ある顧客が 12 か月以内にデータセンターを出る予定でした。ただし、移行には、新しいアプリケーションパフォーマンスモニタリングツールのロールアウトやオペレーティングシステムのアップグレードなど、他の変換アクティビティが含まれていました。1,000 台を超えるサーバーが移行対象であったため、これらのアクティビティにより移行が大幅に遅れました。さらに、このアプローチでは、新しいツールの使用に関するトレーニングが必要でした。顧客は後で、リホストに重点を置いた複数フェーズのアプローチを実装することを決定しました。これにより、移行速度が向上し、データセンターの閉鎖日を満たさないリスクが軽減されました。

結論

大規模な移行では、小規模な移行と比較してさまざまな課題があります。これは主に、スケールによって導入された複雑さが原因です。例えば、エージェントを単一のサーバーにインストールするのは非常に簡単で、約 5 分かかります。ただし、移行の対象となるサーバーが 5,000 台ある場合、これには約 416 時間かかり、次の課題が生じます。

- 異なるプロセスを必要とするオペレーティングシステムが複数ある可能性があります。
- 以前の合併や買収により、管理すべき個別の Microsoft Active Directory ドメインが存在する場合があります。
- ウェーブごとにエージェントのインストールを調整し、進行状況を追跡して報告するには、効果的なプロセスとツールが必要です。

この戦略は、幅広い顧客を支援する AWS プロフェッショナルサービスの経験に基づいて、大規模な移行のベストプラクティスの概要を示しています。これには、人、プロセス、テクノロジーの視点が含まれます。開始する場合や移行中の場合は AWS、AWS プロフェッショナルサービスのコンサルタントがお手伝いします。担当者に連絡して AWS 会話を開始します。

次のステップでは、への大規模な移行を計画して完了できるように設計された AWS 規範ガイドシリーズを確認することをお勧めします AWS クラウド。完全なシリーズについては、[「への大規模な移行 AWS クラウド」](#)を参照してください。

リソース

AWS 大規模な移行

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

関連する AWS 規範ガイドリソース

- [クラウド移行ファクトリーによる大規模なサーバー移行の自動化](#)
- [への移行中に廃止されるアプリケーションを評価するためのベストプラクティス AWS クラウド](#)
- [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](#)
- [移行の準備状況の評価](#)
- [Mobilize your organization to accelerate large-scale migrations](#)

その他のリファレンス

- [AWS クラウド移行ファクトリーソリューション](#)
- [での無料のクラウド移行サービス AWS](#)
- [AWS Database Migration Service](#)
- [AWSで移行](#)

動画

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

寄稿者

この戦略は、AWS プロフェッショナルサービス内のグローバル大規模移行タイガーチームによって作成されました。チームは、AWS お客様 AWS に代わって数千台のサーバーを に正常に移行しました。本ドキュメントの寄稿者は次のとおりです。

- Chris Baker、プリンシパル製品エンジニア
- Dwayne Bordelon、シニアクラウドアプリケーションアーキテクト
- Rodolfo Jr. Cerrada、シニアアプリケーションアーキテクト
- Pratik Chunawala、プリンシパルクラウドアーキテクト
- Bill David、Principal Customer Solutions Manager
- Dev Kar、シニアコンサルタント
- Wally Lu、プリンシパルコンサルタント
- Jon Madison、プリンシパルクラウドアーキテクト
- Abhishek Naik、シニアソリューションアーキテクト
- Damien Renner、シニア移行スペシャリスト
- Amit Rudraraju、シニアクラウドアーキテクト

ドキュメント履歴

次の表に、この戦略の重要な変更を示します。今後の更新に関する通知を受け取る場合は、[RSS フィード](#) をサブスクライブできます。

変更	説明	日付
CloudEndure Migration サービスを削除	CloudEndure Migration サービスへの参照を削除しました。AWS Application Migration Service は、へのlift-and-shift移行に推奨される主要な移行サービスです AWS クラウド。	2022 年 5 月 11 日
AWS ソリューションの名前を更新しました	参照される AWS ソリューションの名前を CloudEndure Migration Factory から Cloud Migration Factory に更新しました。	2022 年 5 月 2 日
更新されたリソース	「Introduction and Resources 」セクションを、大規模な移行シリーズの最新のドキュメントで更新しました。	2022 年 3 月 8 日
初版発行	—	2021 年 9 月 16 日

AWS 規範ガイドの用語集

以下は、AWS 規範ガイドによって提供される戦略、ガイド、パターンで一般的に使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

数字

7 Rs

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

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行する。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの Oracle 用の Amazon Relational Database Service (Amazon RDS) に移行する。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行する。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの EC2 インスタンス上の Oracle に移行する。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) — 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例: Microsoft Hyper-Vアプリケーションをに移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを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。
- 廃止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

「[属性ベースのアクセス制御](#)」をご覧ください。

抽象化されたサービス

「[マネージドユーザー](#)」をご覧ください。

ACID

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

アクティブ/アクティブ移行

(双方向レプリケーションツールまたは二重書き込み操作を使用して) ソースデータベースとターゲットデータベースを同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。[アクティブ/パッシブ移行](#)よりも柔軟な方法ですが、さらに多くの作業が必要となります。

アクティブ/パッシブ移行

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

集計関数

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

AI

「[人工知能](#)」をご覧ください。

AIOps

「[AI オペレーション](#)」をご覧ください。

匿名化

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

アンチパターン

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

アプリケーション制御

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

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

アベイラビリティゾーン (AZ)

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

AWS クラウド導入フレームワーク (AWS CAF)

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

AWS ワークロード認定フレームワーク (AWS WQF)

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

B

不正なボット

個人や組織に混乱や損害を与えることを目的とした [ボット](#)。

BCP

「[ビジネス継続性計画 \(BCP\)](#)」をご覧ください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

二項分類

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

ブルームフィルター

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

ブルー/グリーンデプロイ

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

ボット

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

ボットネット

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

ブランチ

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

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

ブレイクグラスアクセス

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

ブラウフィールド戦略

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

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

「[AWS クラウド導入フレームワーク](#)」を参照してください

カナリアデプロイ

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

CCoE

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

CDC

「[変更データキャプチャ](#)」を参照してください。

変更データキャプチャ (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムでの変更を監査またはレプリケートして同期を維持するなど、さまざまな目的に使用できます。

カオスエンジニアリング

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

CI/CD

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

分類

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

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、「[クラウド運用モデルの構築](#)」を参照してください。

導入のクラウドステージ

組織が、AWS クラウドへの移行時に通常実行する 4 つの段階。

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

これらのステージは、AWS クラウド エンタープライズ戦略ブログのブログ記事「[クラウドファーストへのジャーニー](#)」と「[導入のステージ](#)」で Stephen Orban によって定義されました。AWS 移行戦略との関連性については、「[移行準備ガイド](#)」を参照してください。

CMDB

「[構成管理データベース \(CMDB\)](#)」を参照してください。

コードリポジトリ

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

コールドキャッシュ

空である、または、かなり空きがある、もしくは、古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必要があり、バッファキャッシュから読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

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

コンピュータビジョン (CV)

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

設定ドリフト

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

構成管理データベース (CMDB)

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

コンフォーマンスパック

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

継続的インテグレーションと継続的デリバリー (CI/CD)

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

CV

[「コンピュータビジョン」](#) を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

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

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼できる 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できるようにします。詳細については、「[AWS でのデータ境界の構築](#)」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「[データベース定義言語](#)」を参照してください。

ディープアンサンブル

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

深層学習

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

多層防御

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

委任管理者

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

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

トラブルシューティング

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

開発環境

「[環境](#)」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

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

デザスタ

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

ディザスタリカバリ (DR)

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

DML

「[データベース操作言語](#)」を参照してください。

ドメイン駆動型設計

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

DR

「[ディザスタリカバリ](#)」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、AWS CloudFormation を使用して[システムリソースのドリフトを検出](#)したり、を使用して AWS Control Tower、ガバナンス要件のコンプライアンス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

「[開発バリューストリームマッピング](#)」を参照してください。

E

EDA

「[探索的データ分析](#)」を参照してください。

EDI

「[電子データ交換](#)」を参照してください。

エッジコンピューティング

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

電子データ交換 (EDI)

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

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

[「サービスエンドポイント」](#)を参照してください。

エンドポイントサービス

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

エンタープライズリソースプランニング (ERP)

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

エンベロープ暗号化

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

環境

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

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

エピック

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

ERP

「[エンタープライズリソース計画](#)」を参照してください。

探索的データ分析 (EDA)

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

F

ファクトテーブル

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

フェイルファスト

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

障害分離境界

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

機能ブランチ

「[ブランチ](#)」を参照してください。

特徴量

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

特徴量重要度

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

機能変換

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

数ショットプロンプト

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

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

FGAC

「[きめ細かなアクセス制御](#)」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

FM

「[基盤モデル](#)」を参照してください。

基盤モデル (FM)

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

G

生成 AI

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

ジオブロッキング

「[地理的制限](#)」を参照してください。

地理的制限 (ジオブロッキング)

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

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

Gitflow ワークフロー

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

ゴールデンイメージ

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

グリーンフィールド戦略

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

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。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、AWS Security Hub CSPM、Amazon GuardDuty、AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「[高可用性](#)」を参照してください。

異種混在データベースの移行

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

高可用性 (HA)

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

ヒストリアンのモダナイゼーション

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

ホールドアウトデータ

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

同種データベースの移行

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

ホットデータ

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

ホットフィックス

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

ハイパーケア期間

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

I

laC

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

ID ベースのポリシー

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

アイドル状態のアプリケーション

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

IIoT

「[インダストリアル IoT](#)」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (受信) VPC

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

I

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。この戦略により、大規模な移行に伴うリスクが軽減されます。

インダストリー 4.0

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

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

インダストリアル IoT (IIoT)

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

インスペクション VPC

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

IoT

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

解釈可能性

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

IoT

[「IoT」](#)を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

[「IT 情報ライブラリ」](#)を参照してください。

ITSM

[「IT サービス管理」](#)を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

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

大規模言語モデル (LLM)

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

大規模な移行

300 台以上のサーバの移行。

LBAC

「[ラベルベースアクセス制御](#)」を参照してください。

最小特権

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

リフトアンドシフト

「[7 Rs](#)」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

LLM

「[大規模言語モデル](#)」を参照してください。

下位環境

「[環境](#)」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「[ブランチ](#)」を参照してください。

マルウェア

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

マネージドサービス

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

製造実行システム (MES)

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

MAP

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

メカニズム

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

メンバーアカウント

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

MES

[「製造実行システム」](#) を参照してください。

Message Queuing Telemetry Transport (MQTT)

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

マイクロサービス

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

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

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

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

大規模な移行

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

移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行ファクトリーチームには、通常、運用、ビジネスアナリストおよび所有者、移行エンジニア、デベロッパー、およびスプリントで作業する DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20~50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの[移行ファクトリーに関する解説](#)と [Cloud Migration Factory ガイド](#)を参照してください。

移行メタデータ

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

移行パターン

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

Migration Portfolio Assessment (MPA)

オンラインツール。これによって、AWS クラウドに移行するビジネスケースの検証に必要な情報を得られます。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェーブプランニング) を提供します。[MPA ツール](#) (ログインが必要) は、すべての AWS コンサルタントと APN パートナー コンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

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

移行戦略

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

ML

「[機械学習](#)」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

モノリシックアプリケーション (モノリス)

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

MPA

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

MQTT

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

多クラス分類

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

ミュータブルなインフラストラクチャ

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

O

OAC

「[オリジンアクセス制御](#)」を参照してください。

OAI

「[オリジンアクセスアイデンティティ](#)」を参照してください。

OCM

「[組織変更管理](#)」を参照してください。

オフライン移行

移行プロセス中にソースワークロードを停止させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

「[オペレーション統合](#)」を参照してください。

Ola

「[オペレーショナルレベルアグリーメント](#)」を参照してください。

オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

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

Open Process Communications - Unified Architecture (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

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

運用テクノロジー (OT)

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

オペレーション統合 (OI)

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

組織の証跡

組織 AWS アカウント 内のすべてのイベント AWS CloudTrail をログに記録することによって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウントに作成され、各アカウントのアクティビティを追跡します。詳細については、CloudTrail ドキュメントの「[組織の証跡の作成](#)」を参照してください。

組織変更管理 (OCM)

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

オリジンアクセス制御 (OAC)

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

オリジンアクセスアイデンティティ (OAI)

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

ORR

「[運用準備状況レビュー](#)」を参照してください。

OT

「[運用テクノロジー](#)」を参照してください。

アウトバウンド (送信) VPC

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

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

「[個人を特定できる情報](#)」を参照してください。

プレイブック

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

PLC

「[プログラマブルロジックコントローラー](#)」を参照してください。

PLM

「[製品ライフサイクル管理](#)」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

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

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

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

プライバシーバイデザイン

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

プライベートホストゾーン

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

プロアクティブコントロール

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

製品ライフサイクル管理 (PLM)

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

本番環境

「[環境](#)」を参照してください。

プログラマブルロジックコントローラー (PLC)

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

プロンプトチェイニング

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

仮名化

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

発行/サブスクライブ (pub/sub)

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

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

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

RAG

「[検索拡張生成](#)」を参照してください。

ランサムウェア

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

RASCI マトリックス

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

RCAC

「[行と列のアクセス制御](#)」を参照してください。

リードレプリカ

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

リアーキテクト

「[7 Rs](#)」を参照してください。

目標復旧時点 (RPO)

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

目標復旧時間 (RTO)

サービスが中断から復旧までの最大許容遅延時間。

リファクタリング

「[7 Rs](#)」を参照してください。

リージョン

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

リグレッション

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

リホスト

「[7 Rs](#)」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「[7 Rs](#)」を参照してください。

リプラットフォーム

「[7 Rs](#)」を参照してください。

再購入

「[7 Rs](#)」を参照してください。

回復性

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

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンスコントロール

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

保持

「[7 Rs](#)」を参照してください。

廃止

「[7 Rs](#)」を参照してください。

検索拡張生成 (RAG)

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

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「[目標復旧時点](#)」を参照してください。

RTO

「[目標復旧時間](#)」を参照してください。

ランブック

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

S

SAML 2.0

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

SCADA

「[監視制御とデータ取得](#)」を参照してください。

SCP

「[サービスコントロールポリシー](#)」を参照してください。

シークレット

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

セキュリティバイデザイン

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

セキュリティコントロール

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

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

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

サーバー側の暗号化

送信先で、それ AWS のサービスを受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

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

サービスレベルアグリーメント (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

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

責任共有モデル

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

SIEM

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

単一障害点 (SPOF)

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

SLA

「[サービスレベルアグリーメント](#)」を参照してください。

SLI

「[サービスレベルインジケータ](#)」を参照してください。

SLO

「[サービスレベルの目標](#)」を参照してください。

スプリットアンドシードモデル

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

SPOF

「[単一障害点](#)」を参照してください。

スタースキーマ

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

strangler fig パターン

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

サブネット

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

監視制御とデータ取得 (SCADA)

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

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

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

システムプロンプト

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

T

タグ

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

ターゲット変数

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

タスクリスト

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

テスト環境

「[環境](#)」を参照してください。

トレーニング

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

トランジットゲートウェイ

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

トランクベースのワークフロー

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

信頼されたアクセス

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

チューニング

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

ツーピザチーム

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

U

不確実性

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

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

ワークロード

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

ワークストリーム

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

WORM

「[Write-Once-Read-Many](#)」を参照してください。

WQF

「[AWS ワークロード資格フレームワーク](#)」を参照してください。

Write-Once-Read-Many (WORM)

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

Z

ゼロデイ 익스プロイト

[ゼロデイ脆弱性](#)を悪用した攻撃（一般的にマルウェアによる）。

ゼロデイ脆弱性

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

ゼロショットプロンプト

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

ゾンビアプリケーション

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

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