



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: AWS 大規模な移行の戦略とベストプラクティス

Copyright © 2024 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 — 何を移行していますか?	3
戦略 — なぜ移行したいのですか?	4
タイムライン — いつ移行を完了する必要がありますか?	5
ベストプラクティス	6
人員	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
追加の考慮事項	22
結論	24
リソース	25
AWS大規模な移行	25
AWS関連する規範的ガイドのリソース	25
その他の参考資料	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	42

G	44
H	44
I	46
L	48
M	49
O	53
P	56
Q	58
R	59
S	61
T	65
U	66
V	67
W	67
Z	68
.....	lxx

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、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、移行によってもたらされる組織変革のレベルも考慮します。

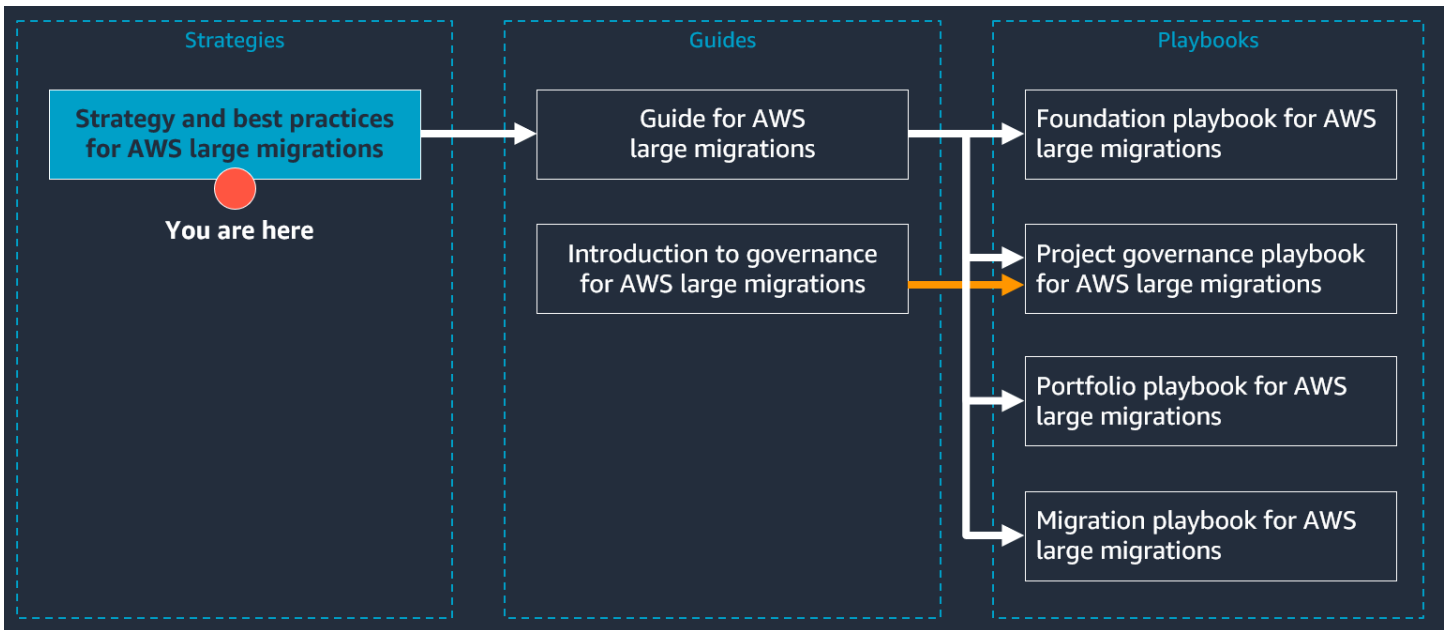
このガイドでは、 に大規模に移行する能力に焦点を当てています AWS。既存のアプリケーションをほとんど、またはまったく変更せずに移行できます。クラウドを起動ポイントとして使用して、これらのアプリケーションをクラウドネイティブまたはサーバーレステクノロジーに移行できます。また、アプリケーションをモダナイズして、追加のビジネスメリットを引き出すことができます。

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、金融サービスやヘルスケアなど、さまざまなセグメントの顧客からのユースケースを提供します。また、へ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な場合があります。既に 100 レベルの知識があり、 [AWS を移行するための推奨プロセス](#) を認識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

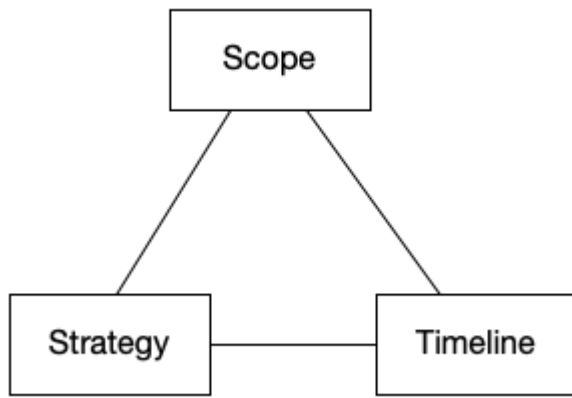
300 台以上のサーバを移行することは「大規模な移行」と見なされます。大規模な移行プロジェクトの人材、プロセス、テクノロジーの課題は、通常、ほとんどの企業にとって新しいものです。このドキュメントは、AWS への大規模な移行に関する規範的ガイドシリーズの一部です AWS クラウド。このシリーズは、クラウドへの移行を効率化するために、最初から正しい戦略とベストプラクティスを適用するのに役立つように設計されています。

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。



スコープ、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それらの関連性を構成する重要な要素は、スコープ、戦略、タイムラインの3つです。



移行の準備を整えるには、移行プログラムの開始時からこれらの要素を調整し、理解しておく必要があります。これらの要素のいずれかを変更すると、他の要素にも影響します。変更がどれほど基本的で賢明であるかにかかわらず、すべての変更にも再調整を考慮する必要があります。

スコープ — 何を移行していますか？

移行の途中であっても、プログラムの全範囲が未定義になることがよくあります。これは、さまざまな要素が後の段階まで解明されない可能性があるためです。たとえば、移行の途中で、構成管理データベース (CMDB) に記録されていないシャドーITのポケットが見つかるかもしれません。あるいは、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やセキュリティサービス (AWSパートナーへのVPN接続、証明書に署名するためのVPN接続など) を考慮せずに、サーバービューに焦点を当てた計画も考えられます。目標とするビジネス成果から逆算して、対象範囲の定義に時間をかけることをお勧めします。最終的には検出ツールを使用して資産を発見することになるかもしれません。ベストプラクティスについては、このガイドの後半で説明します。

大規模な移行には不明な点があるため、範囲は変わります。これらの未知数は、その関連性がほとんどまたはまったく理解されていないまま、環境の考古学の一部となったシステムや、計画に遅れや移行をもたらす生産上のインシデントなどの形である可能性があります。重要なのは、柔軟性を保ち、緊急時対応計画を立ててプログラムを進めることです。

戦略 — なぜ移行したいのですか？

AWSへの移行を計画する理由として次のものが考えられます (複数組み合わせる場合もあります)。

- アプリケーションチームは、新しい CI/CD パイプラインの実装、最新のアプリケーションスタックの導入、サポート対象外のレガシープラットフォームの最新化を求めています。
- インフラストラクチャチームは、リースの期限が切れてプロバイダーが電源を切る前に、老朽化したデータセンターからすぐに退出する必要があります。
- 取締役会は、ビジネスのfutureにおける迅速な変化を考慮して、戦略的な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。

理由が何であれ、ビジネス組織やIT組織は、これらすべての理由やその他の理由を思い浮かべましょう。ドライバーとは何かを理解し、伝え、優先順位を付けることが重要です。要因が増えるたびに、すでに大規模な移行に時間、コスト、範囲、リスクが追加される可能性があります。戦略がタイムラインと範囲に与える影響を十分に認識することが重要です。

移行戦略を定義したら、成功への主な鍵の1つは、さまざまな利害関係者やチーム間で要件を調整することです。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、運用など、組織全体でさまざまなチームが必要です。これらのチームには、個別の優先事項や、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。これらのチームがそれぞれ異なるスケジュールや優先事項に取り組んでいる場合、移行計画の合意と実施はより困難になります。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取り組み、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、保存中のストレージに移行してAWS Key Management Service (AWS KMS) を使用して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が達成される可能性があります。

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のアップグレードにつながる可能性があります。インフラストラクチャチームは質素で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。大規模な移行では、できるだけ基本的な考え方をする必要があります。関係するチームは、すべてを一度に行おうとすることを避けなければなりません。

そのためには、プロジェクトの早い段階で適切な期待値を設定してください。重要なメッセージは、「まず移行してから最新化する」ということです。このアプローチにより、組織は技術的負債を減らし、最終的には大規模に運営できるようになるだけでなく、AWS クラウドが提供できるスケラビ

リティと俊敏性を利用することで、さまざまな最新化アプローチへの道が開かれます。長期的に考えることは、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導入と管理を合理化するのに役立ちます。その結果、ビジネス部門は機能のリリースサイクルを短縮できます。

タイムライン — いつ移行を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わないようにする必要があります。移行の推進要因が決まった完了日を基準としている場合は、そのタイムラインの要件を満たす戦略を選択する必要があります。大規模な移行のほとんどは、こうした時間ベースの制約に基づいているため、移行戦略には、延長やオーバーランの余地がほとんどなく、明確なスケジュールと結果が必要です。

このような時間のかかる移行では、「まず移行してから最新化する」アプローチをお勧めします。これは期待値を設定するのに役立ち、チームが個々のプロジェクト計画と予算を全体的な移行目標と一致させるように促します。プロジェクトのできるだけ早い段階で意見の相違を見つけ出し、迅速に失敗して運営委員会レベルで意見の相違に対処し、適切な利害関係者を巻き込んで連携を図ることが重要です。

逆に、移行の主な目的がアプリケーションの最新化のメリットを享受することである場合は、プログラムの早い段階でそのことを説明する必要があります。多くのプログラムは、期限を定めた初期目標からスタートし、未解決の問題や問題を解決しようとする利害関係者からの要求に応える計画を立てていません。これらの問題がソースシステムに何年も存在していたケースもありますが、今では人為的に移行を妨げる原因となっています。

移行中の最新化アクティビティは、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ものでも、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。これらは些細なものとは見なす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れば、大規模な移行を簡素化できるいくつかの重要な要素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、他の顧客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。

- 人員
- 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分野に焦点を当てます。

- エグゼクティブサポート — 意思決定の権限を持つシングルスレッドリーダーの特定
- 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 — さまざまなツールでプロアクティブにチームをトレーニングする

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連携](#)

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を専有す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、サイロを回避し、ワークストリームを合理化する権限を持っています。

大規模な移行のグローバルカスタマーは、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アップできました。シングルスレッドリーダーとし

での CIO の完全なサポートは、移行中のサーバーの急速なスケールアップに不可欠です。CIO が毎週移行のカットオーバーを行うと、移行チームと が呼び出され、問題のエスカレーションと解決がリアルタイムで行われるため、移行速度が向上します。

上級リーダーシップチームの連携

移行の成功基準に関して、さまざまなチーム間で調整を作成することが重要です。移行の計画と実施は小規模な専用チームによって達成できますが、戦略を定義して周辺活動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による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有効化チームを作成する](#)
- [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確認する](#)

部門横断的なクラウド有効化チームを作成する

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、[クラウド有効化エンジン \(CEE\)](#) を構築することをお勧めします。CEE は、への移行に関する組織の運用準備状況に焦点を当てた、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、セキュリティの表現を含む部門横断的なチームである必要があります。チームには次の責任が課金されます。

- 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
- 表現するすべての分野にわたるステークホルダーの連携を促進することを続ける

1人の医療顧客が CEE から始めなかった。ただし、最初のパイロット移行では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ている状態で、チームは移行ウォールームを実装しました。移行のウォールームでは、インフラストラクチャ、セキュリティ、アプリケーション、ビジネスの利害関係者が問題の解決に役立つ場合があります。

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高めるには、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、カットオーバーへのサインオフ要件が必要です。

移行はコアチームによって主導されますが、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の検証とテストに関与している可能性があります。顧客は、多くの場合、クラウド移行に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の必要な関与を考慮することをお勧めします。例えば、リホスト戦略では、より多くのアプリケーションランドスケープが変更されている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、アプリケーションチームの関与が少なくなります。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確認する

企業の既製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集中している可能性があります。例えば、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限が含まれていない場合があります。ただし、ベンダーとのライセンスを再ネゴシエートすると、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して作業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を与え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関するチームのトレーニング](#)

新しいツールとプロセスに関するチームのトレーニング

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どのトレーニングが必要かを理解するために時間を費やす必要があります。移行中は、組織のAWS Database Migration Service新しいなどのツール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削減できます。

ツールを実践的に試す機会とな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。例として、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、複数の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、移行段階に移行したチームが基本的な知識を持つことが保証されました。また、各SI AWSパートナーチーム内のファーストラインエスカレーションとして機能する内容領域専門家(SMEs)の特定も支援します。

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、クラウド移行ファクトリーソリューションは、移行の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、技術を使用して必要な規模と速度を達成するためのベストプラクティスの一部を、スコープ、戦略、タイムラインに合わせて説明します。

最も重要な原則は、可能な限り自動化の分野を調べることです。対象範囲に数千台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が必要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)
- 在庫スプレッドシート
- プロジェクト管理

これらのツールは、評価から準備、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推進されます。

移行フェーズが計画されたら、次のステップは、移行チームが必要とする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、的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てください。可能であれば、チームが安全な環境で移行ツールを試すことができるイベントを作成します。例えば、チームがツールの使用経験に移行できるサンドベンチサーバーやラボサーバーはありますか？ または、初期の開発ワークロードを学習目的で使用することは許容できますか？

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では、移行の範囲 (移行する必要があるもの) を理解し、戦略 (移行方法) を策定します。検出は、この重要な側面です。移行戦略の決定木を形成するために必要なメタデータポイントがキャプチャされます。ワークロードを一定の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、Migration Factory 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を大幅に削減できます。

1 人のお客様が、マイグレーションファクトリー向けに完全に自動化されたデータ取り込みプロセスを作成しました。すべての移行メタデータを含む移行ウェブプランは、Microsoft のスプレッドシートでホストされ、維持されました SharePoint。ソースに変更が加えられると、AWS Lambda 関数が開始され、手動による介入なしで移行ファクトリーにデータがロードされます。この自動データインテイクプロセスにより、お客様は手動作業を減らし、人為的ミスをもっと抑え、作業速度を速めることができます。1,000 台以上のサーバーを 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、多くの小さ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や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。Migration Factory は、標準化されたデータセットを使用して移行を加速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一緒に自動化できるすべての手動タスクの自動化に時間を費やします。

[Cloud Migration Factory](#) ソリューションはその一例です。Cloud Migration Factory は、組織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。例えば、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、サーバーの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

プログラムの重要業績評価指標 (KPIs など、ライブデータを追跡してレポートするための自動移行レポートダッシュボードを作成することをお勧めします。移行プロジェクトには、組織全体のステークホルダーが参加し、以下が含まれます。

- アプリケーションチーム
- テスター
- 廃止チーム
- 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

役割を実行するには、これらの利害関係者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認識している必要があります。AWS。リーダーシップチームは、移行のどれだけ完了しているかを理解したい

と考えています。データの信頼できる自動ライブフィードを用意することで、通信ミスを防ぎ、意思決定を行えるようになります。

大規模な医療の顧客が、データセンターの退場に、期限が近づいています。規模と複雑さを考慮すると、最初はステークホルダー間の移行状況の追跡と伝達にかなりの時間が費やされていました。移行チームは後で Amazon を使用してデータを視覚化した自動ダッシュボード QuickSight を構築し、移行速度を向上させながら追跡とコミュニケーション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

移行に適したツールを選択することは、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、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設定することをお勧めします。この探索にはライセンスコストがかかる場合がありますが、より広範なイニシアチブを検討するとコスト面で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ことがあります。例えば、アプリケーションのパフォーマンスモニタリングツールが既にエステート全体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、最初は知識がないため、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 5TAK 10 時間の会議を実行する必要がありました。検出ツールを使用した場合、検出作業が 1,000 時間を超えて短縮された可能性があるかと推定されました。

前提条件と移行後の検証

このセクションの内容:

- [移行前の段階でランディングゾーンを構築する](#)
- [概要の前提条件アクティビティ](#)
- [移行後のチェックを実装して継続的な改善を実現する](#)

移行前の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。移行には、アーキテクチャが適切に設計されたランディングゾーンを構築することが前提条件です。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ことに伴う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導入することで、ステークホルダーはアカウントレベルまたは VPC レベルで管理される側面を気にすることなく、ワークロードの移行に集中できます。

概要の前提条件アクティビティ

ランディングゾーンとは別に、移行前に他の技術的前提条件、特にリードタイムの長いプロセスを調整することが重要です。例えば、オンプレミスからデータをレプリケートできるように、必要なファイアウォールを変更します AWS。技術的な前提条件を早期に伝達することで、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、移行が停止するのは一般的です。これは進行中の移行ウェーブに影響するだけでなく、問題が修正されている間、今後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。

いくつかのデータセンターを退避することを目指して AWS、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、意図した速度では十分ではありません 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。

移行後のチェックを実装して継続的な改善を実現する

最後に、オペレーションの統合、コストの最適化、ガバナンスとコンプライアンスのチェックなど、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、移行済みのワークロードを評価して、将来のウェーブに適用すべき技術的な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。例えば、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のサイズに合わせるすることができます。データセンター閉鎖の重要な経路でのテストが終了したので、Amazon CloudWatch を使用してインスタンスの使用率を評価し、より小さな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、大規模なテクノロジーのお客様は大規模な移行を実行していましたが、最初は移行後の検証が含まれていませんでした。100 台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を確認しました。以前に移行したサーバーはすべて修復する必要があり、移行は停止しました。また、お客様は、インスタンスのサイズが最初の見積もりの 5 倍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

プロセスには一貫性がありますが、各プロジェクトは一意であるため、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、学習、導入、反復など、大きなメリットをもたらす可能性のある改善点のギャップと余地を特定できます。これらの変化は、プロジェクトや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、品質とチームの信頼度をもたらす成長への期待につながります。

移行プロセスは、以前にリンクされていなかった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賛同してもらうために必要なコア原則の概要を説明します。

このセクションの内容:

- [ビジネス推進要因を定義し、タイムライン、範囲、戦略を伝える](#)
- [ブロッカーを削除するの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](#)

ビジネス推進要因を定義し、タイムライン、範囲、戦略を伝える

への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。ビジネスの俊敏性を高めることが最終的な目標である場合は、より多くの変革レベルを伴う2つ目のパターンを優先することができます。1年の終わりにデータセンターを退避することが目標であれば、リホストによって提供される速度のために、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。

- アプリケーション所有者
- 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会議など、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と密接に一致する重要業績評価指標 (KPIs) を作成します。

例えば、あるお客様がデータセンターから退避するという目標のビジネス上の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整合しませんでした。その結果、データセンターの閉鎖日を見逃すが、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データセンターをタイムリーに閉鎖し、でアプリケーションをモダナイズするかという技術的な議論が数か月発生しましたAWS。

ブロッカーを削除するの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。その結果、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、プログラムでは予算の量が制限され、目標成果が定義されている可能性があります。さまざまな利害関係者を管理し、目標とするビジネス成果に集中することは難しい場合があります。このチャレンジは、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乗算したときに複合されます。さらに、ステークホルダーは、他の優先順位を持つさまざまなリーダーシップチームにレポートする可能性があります。これを念頭に置いて、ターゲットとなるビジネス成果を明確に文書化しながら、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削除することが重要です。これにより、多大な時間を節約し、さまざまなチームを共通の目標に合わせるすることができます。

これを示す例としては、12か月以内に主要なデータセンターから退避することを目標とする金融サービス会社が挙げられます。明確な義務付けやエスカレーションの経路がなかったため、時間や予

算の制約にかかわらず、利害関係者が目的の移行経路を作成したことになります。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

変化は良いですが、変化が多いほどリスクが大きくなります。大規模な移行のビジネスケースが承認されると、データセンターを特定の日付までに退避するなど、このイニシアチブを促進するターゲットとなるビジネス成果がある可能性が最も高くなります。技術専門家が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のは一般的ですが、これはビジネス目標ではないかもしれません。

1人のお客様は、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何か月も書き換えられないようにする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させる必要があったときに、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「2週間のルール: クラウドのアプリケーションを10日以内にリファクタリングする」](#)を参照してください。

ビジネス成果に合わ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、今後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てください。

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の完全な移行プロセス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。また、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、移行を進める過程で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、既存のプロセスが理解され、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連携が必要な場所を含めます。プロセスが理解できたら、責任、説明、相談、報告 (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係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。カウントダウン計画は通常、ワークロード移行のカットオーバー日時とは逆に機能します。

このドキュメントアプローチは、1年未満でAWS正常に移行し、4つのデータセンターを終了した複数ある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が関係していたため、管理オーバーヘッドが発生し、back-and-forth 決定や実装の遅延が発生していました。AWS プロフェッショナルサービスチームは、顧客およびサードパーティ

とともに、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。結果の 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カーと問題を緩和しました。その後、スケジュールよりも先にデータセンターから退出できました。

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、保険会社がデータセンターを 4 か月以内に退出できました。お客様は責任共有モデルを理解し、実装し、詳細な RACI マトリックスに従って、移行全体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、お客様は最初の 12 週間で 350 台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 カットヤーを作成すると考えてください。再利用可能なリファレンス、ドキュメント、ランブック、パターンは、スケールするための鍵です。これらのジャーナルは、将来の移行プロジェクトで再利用して回避できる経験、学習、落とし穴、問題、ソリューションであり、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。

例えば、1 人の顧客が、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、同じ情報をさまざまな方法で提示できる可能性があるため、カスタマーチームに多くのストレスがかかります。これらの早期対応後、お客様は移行に使用するすべてのドキュメントとアーティファクトの中心的所有権を確立し、推奨される変更を送信するプロセスを導入しました。これらのアセットには以下が含まれます。

- 標準的な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
- ビジネスの重要性に基づくアプリケーションアーキテクチャとセキュリティ標準

さらに、これらのドキュメントや標準への変更は毎週すべてのチームに送信され、変更の受信と準拠を確認するために各パートナーが必要でした。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の個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、成功を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

大規模な移行の計画では、信頼できる情報源を確立することは、さまざまなチームが足並みをとってデータ主導の意思決定を可能に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

データベース (CMDB)、アプリケーションの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る場合もあります。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。

移行に使用できるさまざまなデータソースを 1 つのデータセットに集約することが重要です。その後、実装中の移行を追跡するために、信頼できる単一のソースを使用できます。例えば、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AWSに集中して、すべてのワークロードを に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、ビジネスの重要性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。

別の例として、同じ業界の企業が、サーバーインフラストラクチャのインベントリの 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。データが正しくないため、すぐに移行番号が減少するようになりました。この場合、アプリケーション所有者は理解できなかったため、テスターを期間内に見つけることができません。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に合わせていなかったため、サーバーはビジネス目的で使用せずに実行されました。

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ると、大規模な移行の範囲を持続可能な移行イベントや波にどのように導くかを計画することができます。次の例は、ウェブプランを作成するための主要なガイダンスを提供します。

このセクションの内容:

- [移行ウェブを事前に計画して、安定したフローを確保する](#)
- [ウェブの実装とウェブ計画を別々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](#)
- [遡及的な を忘れないでください](#)

移行ウェーブを事前に計画して、安定したフローを確保する

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。「計画に失敗すると、失敗することになります」という内容になります。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、プロジェクトを迅速に進めることができます。これにより、プロジェクトのスケーリングがより簡単になり、プロジェクトの要求が高まり、複雑になるにつれて意思決定と予測が向上します。事前に計画を立てておくことで、変更に対応するチームの能力も向上します。

例えば、大規模な金融サービスの顧客がデータセンターの終了プログラムに取り組んでいるとします。最初に、お客様は移行ウェーブを順番に計画し、次の計画を開始する前に 1 つのウェーブを完了しました。このアプローチでは、準備にかかる時間が短くなりました。ステークホルダーにアプリケーションが移行されていることが通知されても AWS、移行を開始する前にいくつかの手順を実行できました。これにより、プログラムの大幅な遅延が発生しました。顧客がこれを実現した後、移行ウェーブが数か月前に計画された、包括的で将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。

ウェーブの実装とウェーブ計画を別々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 つのプロセスが並行して動作します。通信と調整では、十分なサーバーやアプリケーションが予想される速度を達成できないため、移行が遅くなります。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できません。この課題は、多くの場合、次の原因によって引き起こされます。

- 移行実装チームはウェーブ計画に関与しておらず、ウェーブ計画フェーズで収集されたデータは完了していません。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するようにスケジュールされ、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、ウェーブ計画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

その後のウェーブごとに小規模から始めて移行速度を向上させることを計画してください。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。その後のウェーブにアプリケーションとサーバーを追加し、移行速度を最大化します。複雑さやリスクの少ないアプリケーションの優先順位付け、スケジュールでの速度の向上により、チームは作業に合わせて調整し、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。

1 人のお客様が、1 年間に 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の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許可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、将来のウェーブで不要な遅延を防ぐことができます。さらに、チームはウェーブごとに検出とカットオーバーのプロセスをより多く自動化するスクリプトを開発できました。小規模から始めると、チームは初期のプロセスの改善に集中し、信頼度が大幅に向上しました。

カットオーバーウィンドウの数を最小限に抑える

一括移行には、規模を拡大するための統制的な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、大規模な移行に対するアンチパターンです。週 1 回のカットオーバー期間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は価値が高くなります。

例えば、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 個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 個のカットオーバーを行うことができます。カットオーバーごとの時間と労力は似ているため、カットオーバーが少なくなり、カットオーバーが大きくなると、スケジュー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。

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リースされたデータセンターから移行しようとしていました。有効期限がないと、費用がかかり、短期更新条件が発生します。移行の早い段階で、アプリケーションチームは過去 1 分間の移行スケジュールを指示できました。これには、カットオーバーの直前の何らかの理由で移行をオプトアウトすることも含まれます。これにより、プロジェクトの初期段階で多くの遅延が発生しました。多くの場合、顧客は直近 1 分間に他のアプリケーションチームとネゴシエートして入力する必要がありました。顧客は最終的に計画分野を増やしましたが、この初期のミスにより、移行チームに一定の負荷がかかりました。全体的なスケジュールが遅れると、一部のアプリケーションがデータセンターから遅れることがなくなりました。

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。早い段階で失敗することで、チームは学習し、ボトルネックを理解し、学習した教訓をより大きなウェーブに適用できます。移行の最初のいくつかのウェーブは、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、互いに調整し、プロセスを進めています。
- 大規模な移行には、通常、さまざまなツールや人が含まれます。
- end-to-end プロセスの統合、テスト、失敗、学習、および継続的な改善には時間がかかります。

問題は一般的なものであり、最初のいくつかのウェーブで予想されます。一部のチームでは新しいことを試して失敗したくない場合があるため、これを組織全体に理解して伝達することが重要です。障害が発生すると、チームに報酬が減り、将来の移行の障害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認識し、全員が失敗するように促すことが、移行を成功させる鍵となります。

ある企業が 24~36 か月で 10,000 台以上のサーバーを移行する予定です。この目標を達成するために、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。最初のいくつかのウェーブはウェーブを学習することで、チームが何が機能し、誰が何を実行できるかを理解できるようにしました。また、CMDB やとの統合など、プロセスを改善する統合も特定しました CyberArk。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 台を超えるサーバーを移行できました。

遡及的な 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を取り、調整し、学習し、合意し、進める場所です。最も基本的なレベルの遡及的な考え方とは、さかのぼって何が起こったのかを説明し、何が成功し、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、これらの議論に基づいて改善を構築できます。遡及的は、教訓を学びながら、形式やプロセスをまとめます。大規模な移行を成功させるには、プロセス、ツール、チームを常に進化させ、改善する必要があるため、遡及的は重要です。遡及的は、その点で重要な役割を果たします。

従来の教訓によるセッションはプログラムの終了時までに行われないため、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は確認されません。大規模な移行では、学習した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なる必要があります。

1 人のお客様については、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、毎週の遡及が保持されました。これらのセッションでは、プロセスの観点から、または自動化から合理化

する範囲がある分野が明らかになりました。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の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。

別の大規模な技術会社では、以前の移行ウェーブの問題を特定するために、定期的にチームに遡及していました。これにより、プロセス、スクリプト、オートメーションの改善により、プログラムの過程で平均移行時間が 40% 短縮されました。

追加の考慮事項

多くの領域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、考慮する必要がある他の項目について説明します。

このセクションの内容:

- [作業中にクリーンアップする](#)
- [追加の変換に複数のフェーズを実装する](#)

作業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してクリーンアップされるまでプロジェクトが完了しない場合、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に残ってコストが増え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のセキュリティプラクティスでもあります。

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しておく、クラウドへの移行の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 アドレスなど、アカウントをテストする
- 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)
- 領域を消費するファイル (移行に使用される一時的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、SI AWSパートナーは移行が成功してもレプリケーションエージェントを削除しませんでした。すでに移行されたレプリケーションサーバーと EBS ボリュームのAWS監査では、毎月 20,000 USD (USD) のコストがかかることがわかりました。この問題を軽減するために、AWSプロフェッショナルサービスは、古いサーバーがまだレプリケートされたときに 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および古いインスタンスに対してアクションを実行できます。

将来の移行では、移行後のハイパーペイント期間を 48 時間として定義し、プラットフォームの採用を円滑に進めるプロセスを採用しました。その後、お客様のインフラストラクチャチームはオンプレミスサーバーに廃止リクエストを送信しました。廃止リクエストの承認時に、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をお勧めします。

追加の変換に複数のフェーズを実装する

大規模な移行を実施する場合、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、中核となる目標に集中することが重要です。小規模な移行では、スコープクリープによる影響は最小限に抑えられます。ただし、数日分の追加作業に数千台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ために、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的の場合、フェーズ 1 にはワークロードをできるだけ AWS速く に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た後、フェーズ 2 では、ターゲットとなるビジネス成果をリスクにさらすことなく、変革アクティビティを実装できます。

例えば、ある顧客が 12 か月以内にデータセンターを退出する予定です。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、他の変革活動も含まれていました。1,000 台以上の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に大幅な遅延が発生しました。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にしました。これにより、移行速度が向上し、データセンターの閉鎖日に間に合わないリスクが軽減されました。

結論

大規模な移行は、小規模な移行と比べてさまざまな課題をもたらします。これは主に、スケールによって生じる複雑さによるものです。たとえば、エージェントを1台のサーバにインストールするのは非常に簡単で、所要時間は約5分です。ただし、移行の対象となるサーバが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存在する可能性があります。
- 以前の合併や買収により、管理対象の Microsoft Active Directory ドメインが別々にある場合があります。
- エージェントインストールを段階ごとに調整し、進捗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、AWS幅広い顧客を支援するプロフェッショナルサービスの経験に基づいた大規模な移行のベストプラクティスを概説しています。これには、人、プロセス、テクノロジーの視点が含まれます。移行を開始したい、または移行中の場合はAWS、AWSプロフェッショナルサービスのコンサルタントが喜んでお手伝いします。AWS担当者に連絡して、会話を始めましょう。

次のステップについては、AWSへの大規模な移行を計画して完了するのに役立つ規範的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。シリーズ全体については、[「への大規模な移行」](#)を参照してくださいAWS クラウド。

リソース

AWS大規模な移行

AWS大規模な移行に関する規範的ガイドランスシリーズの全内容については、「[への大規模な移行](#)」を参照してくださいAWS クラウド。

AWS関連する規範的ガイドランスのリソース

- [クラウドMigration Factory 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](#)
- [AWS安全でスケーラブルなマルチアカウント環境のセットアップ](#)
- [移行準備状況の評価](#)
- [組織を動員して大規模な移行を加速させましょう](#)

その他の参考資料

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

動画

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

寄稿者

この戦略は、プロフェッショナルサービス内のグローバル大規模マイグレーションタイガーチームによって作成されました。AWS AWS このチームは顧客に代わって数千台のサーバーを移行することに成功しました。AWS 本ドキュメントの寄稿者は次のとおりです。

- クリス・ベイカー、主任プロダクトエンジニア
- シニアクラウドアプリケーションアーキテクト、Dwayne Bordelon 氏
- ロドルフォ・ジュニア セラーダ、シニア・アプリケーション・アーキテクト
- プラティック・チュナワラ、プリンシパル・クラウド・アーキテクト
- ビル・デイビッド、プリンシパル・カスタマー・ソリューション・マネージャー
- Dev Kar、シニアコンサルタント
- ウォーリー・ルー、プリンシパル・コンサルタント
- ジョン・マディソン、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
- ダミアン・レナー、シニア・マイグレーション・スペシャリスト
- アミット・ルドララジュ、シニアクラウドアーキテクト

ドキュメント履歴

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

変更	説明	日付
CloudEndure 移行サービスを削除しました	CloudEndure 移行サービスへの参照が削除されました。 。 AWS Application Migration Service lift-and-shift AWS クラウドへの移行に推奨される主な移行サービスです。	2022 年 5 月 11 日
AWSソリューション名の更新	AWS CloudEndure 参照先のソリューションの名前をマイグレーションファクトリからクラウドマイグレーションファクトリに更新しました。	2022 年 5 月 2 日
更新されたリソース	「 はじめに 」と「 リソース 」セクションを、大規模な移行シリーズの最新ドキュメントで更新しました。	2022 年 3 月 8 日
初回刊行物	—	2021 年 9 月 16 日

AWS 規範ガイド用語集

以下は、AWS 規範的ガイドが提供する戦略、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します。
- 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ある程度の最適化を導入します。例: オンプレミスの Oracle データベースをの Oracle 用 Amazon Relational Database Service (Amazon RDS) に移行します AWS クラウド。
- 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 に移行します。
- 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: オンプレミスの Oracle データベースをの EC2 インスタンスで Oracle に移行します AWS クラウド。
- 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: を移行する Microsoft Hyper-V へのアプリケーション 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。
- 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

[原子性、一貫性、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。柔軟性がありますが、[アクティブ/パッシブ移行](#)よりも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

行のグループで動作し、グループの単一の戻り値を計算SQLします。集計関数の例には、SUMおよびMAXが含まれます。

AI

[人工知能](#) を参照してください。

AIOps

[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。

人工知能オペレーション (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。移行戦略で AWS がどのように AIOps 使用されるかの詳細については、「[オペレーション統合ガイド](#)」を参照してください。

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、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コントロール (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[ABAC AWS](#)「」の「」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

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安価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)

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は、組織がクラウドへの移行を成功させるための効率的かつ効果的な計画を策定 AWS するのに役立ちます。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ダンスをまとめています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、AWS CAFは、クラウド導入を成功させるための準備に役立つ人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、[AWS CAFウェブサイト](#)と[AWS CAF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

個人または組織に混乱または害を与えることを目的とした[ボット](#)。

BCP

[事業継続計画](#) 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 で動作グラフを使用して、失敗したログオン試行、疑わしいAPI呼び出し、および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。 [「endianness」](#) 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。

ブルー/グリーンデプロイ

2 つの別々の環境を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は 1 つの環境 (青) で実行し、新しいアプリケーションバージョンは他の環境 (緑) 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。

ポット

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インタラクション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をインデックス化するウェブクローラーなど、一部のポットは有用または有益です。悪質なポットと呼ばれる他のポットの中には、個人や組織に混乱や害を与えることを意図したものもあります。

ポットネット

[マルウェア](#) に感染し、[ポット](#) ハーダーまたはポットオペレーターと呼ばれる 1 つの当事者によって制御されているポットのネットワーク。ポットネットは、ポットとその影響をスケールするための最もよく知られているメカニズムです。

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、[「ブランチについて」](#) (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アクセスするための簡単な手段を提供します。詳細については、「Well-Architected」ガイドの AWS [「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、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。

事業継続計画 (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 Cloud Adoption Framework](#) を参照してください。

Canary のデプロイ

エンドユーザーへのバージョンのスローリリースと増分リリース。自信が持てば、新しいバージョンをデプロイし、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

[データキャプチャの変更](#) を参照してください。

データキャプチャの変更 (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。は、同期を維持するために、ターゲットシステムの変更を監査したりレプリケートしたりするなど、CDCさまざまな目的で使用できます。

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します。[AWS Fault Injection Service \(AWS FIS \)](#) を使用して、AWS ワークロードに負荷をかけ、そのレスポンスを評価する実験を実行できます。

CI/CD

[継続的統合と継続的配信](#) 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 AWS のサービス 受信する前に、データをローカルで暗号化します。

Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、AWS クラウド エンタープライズ戦略ブログの[CCoE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](#)テクノロジー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 に移行するときに通常実行する 4 つのフェーズ AWS クラウド :

- 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
- 基盤 — クラウド導入を拡大するための基盤投資 (ランディングゾーンの作成、 の定義CCoE、オペレーションモデルの確立など)
- 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

[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以下が含まれます。GitHub または Bitbucket Cloud。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、デジタル画像や動画などのビジュアル形式から情報を分析および抽出する [AI](#) の分野。例えば、はオンプレミスのカメラネットワークに CV を追加するデバイス AWS Panorama を提供し、Amazon SageMaker は CV の画像処理アルゴリズムを提供します。

設定ドリフト

ワークロードの場合、設定は想定された状態から変更されます。ワークロードが非準拠になる可能性があり、通常、段階的かつ意図的ではありません。

設定管理データベース (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、移行の CMDB ポートフォリオ検出および分析段階でのデータを使用します。

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、または組織全体に単一のエンティティとしてデプロイできます。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 is commonly described as a pipeline. CI/CD は、プロセスの自動化、生産性の向上、コード品質の向上、および迅速な提供に役立ちます。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

一元的な管理とガバナンスで分散された分散データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが期待されたネットワークから信頼された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、「[「でデータ境界を構築する AWS」](#)」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには、通常、大量の履歴データが含まれ、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 に採用する場合 AWS、リソースの保護に役立つように、AWS Organizations 構造のさまざまなレイヤーに複数のコントロールを追加します。例えば、アプローチでは defense-in-depth、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、リーンな製造プラクティス用に最初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ます。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的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番号です。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一般的に使用されます。

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。

ディザスタリカバリ (DR)

[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 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。AWS

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 (ボストン: Addison-Wesley Professional、2003)。ストラングルラーの fig パターンでドメイン駆動型設計を使用する方法については、「[従来の Microsoft のモダナイズ](#)」を参照してくださいASP。NET (ASMX) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に更新する「」。

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。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、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化を検出](#)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。

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、レスポンス時間を向上させることができます。

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) 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、インターフェイスエンドポイントを作成することでVPC、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、アイデンティティとアクセスの管理、検出コントロール、インフラストラクチャのセキュリティ、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、サマリー統計を計算し、データ視覚化を作成することで実行されます。

F

ファクトテーブル

[星スキーマの中央テーブル](#)。ビジネスオペレーションに関する定量的なデータを保存します。通常、ファクトテーブルには 2 つのタイプの列が含まれます。つまり、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です。

フェイルファースト

開発ライフサイクルを短縮するために頻繁で増分的なテストを使用する哲学。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、アベイラビリティゾーン、コントロールプレーン AWS リージョン、データプレーンなどの境界が、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のに役立ちます。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 Explanations (SHAP) や統合勾配など、さまざまな手法で計算できる数値スコアとして表されます。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。

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コントロール (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](#)による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

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)

Amazon では CloudFront、特定の国のユーザーがコンテンツディストリビューションにアクセスできないようにするオプションです。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分散の制限](#)」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](#)はモダンで望ましいアプローチです。

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

組織全体のリソース、ポリシー、コンプライアンスを管理するのに役立つ大まかなルール (OUs)。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、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。](#)

ハイアベイラビリティ (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。

同種データベースの移行

ソースデータベースを、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に移行する (Microsoft SQL Server から Amazon RDS for SQL Server など)。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。緊急性のため、通常、ホットフィックスは一般的な DevOps リリースワークフローの外部で行われます。

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

IaC

[「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 日間の平均使用量 CPU とメモリ使用量が 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

[「産業用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、本番稼働ワークロード用に新しいインフラストラクチャをデプロイするモデル。イミュータブルインフラストラクチャは、本質的に [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 Framework」の [「イミュータブル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 (インGRESS) VPC

AWS マルチアカウントアーキテクチャでは、VPC が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、ルーティングします。[AWS セキュリティリファレンスアーキテクチャ](#) 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インターフェイス VPCs を保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

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。

インダストリー 4.0

接続、リアルタイムデータ、オートメーション、分析、AI/ML の進歩を通じて、製造プロセスのモダナイゼーションを指すために 2016 年に [Klaus Schwab](#) によって導入された用語。

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。

産業用モノのインターネット (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、[「産業用モノのインターネット \(IIoT\) 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。

検査 VPC

AWS マルチアカウントアーキテクチャでは、VPCs (同一または異なる 内の 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VPCを管理する一元化されたです。[AWS セキュリティリファレンスアーキテクチャ](#)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インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

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

[「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 の基盤を提供しますITSM。

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

ラベルベースのアクセスコントロール (LBAC)

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ている必須のアクセスコントロール (MAC) の実装。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「[最小権限の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。[「endianness」](#) 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。

マルウェア

コンピュータのセキュリティまたは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を混乱させたり、機密情報を漏洩したり、不正アクセスを受けたりする可能性があります。マルウェアの例としては、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 を操作し、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。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象サービスとも呼ばれます。

製造実行システム (MES)

原材料を作業現場の最終製品に変換する生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステム。

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、ツールの採用を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作時にそれ自体を強化して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[「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

の組織の一部である管理アカウント AWS アカウント を除くすべての 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。

メッセージキューイングテレメトリトランスポート (MQTT)

リソースに制約のある [IoT](#) デバイス向けの、machine-to-machine [パブリッシュ/サブスクライブ](#) パターンに基づく軽量 (M2M) 通信プロトコル。

マイクロサービス

明確に定義された上で通信APIsし、通常、小規模な自己完結型チームが所有する、小規模で独立したサービス。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APIs。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

びスケールリングできます。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。

移行促進プログラム (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、従来の移行を系統的な方法で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して高速化するための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、スプリントに携わる 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には、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 EC2を使用して Amazon への移行をリホストします。

移行ポートフォリオ評価 (MPA)

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 AWS クラウド。MPA は、詳細なポートフォリオ評価 (サーバーの適正サイズ、料金、TCO比較、移行コス

ト分析)と移行計画(アプリケーションデータ分析とデータ収集、アプリケーショングループ化、移行の優先順位付け、ウェーブプランニング)を提供します。[MPA ツール](#) (ロゲインが必要)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)

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 AWS CAF。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 移行戦略の最初のフェーズ](#) です。

移行戦略

ワークロードを に移行するために使用するアプローチ AWS クラウド。詳細については、この用語集の「[7 Rs エントリ](#)」および「[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、「」の「[アプリケーションのモダナイズ戦略 AWS クラウド](#)」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、「」の「[アプリケーションのモダナイゼーション準備状況の評価 AWS クラウド](#)」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

[「移行ポートフォリオ評価」](#)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。Well-Architected Framework AWS 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](#)をベストプラクティスとして使用することを推奨しています。

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

[「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。

OPC-UA

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

Open Process Communications - 統合アーキテクチャ (OPC-UA)

産業オートメーション用の (M2M) machine-to-machine通信プロトコル。OPC-UA は、データの暗号化、認証、認可スキームとの相互運用性標準を提供します。

運用レベルの契約 (OLA)

サービスレベルの契約 () をサポートするために、IT グループが相互にどのような機能を提供することを約束するかを明確にする契約SLA。

運用準備状況のレビュー (ORR)

インシデントや潜在的な障害の範囲を理解、評価、防止、または軽減するのに役立つ質問とそれに関連する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「[Operational Readiness Reviews \(ORR\)](#)」を参照してください。

運用テクノロジー (OT)

物理環境と連携して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ハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、[Industry 4.0](#) 変換の主要な焦点です。

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 のすべてのイベント AWS CloudTrail をログに記録する 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、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、変化の採用を加速し、移行に伴う問題に対処し、文化的および組織的な変化を推進することで、組織が新しいシステムや戦略の準備と移行を支援します。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から、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の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 S3 バケット AWS リージョン、AWS KMS (SSE-KMS) によるサーバー側の暗号化、および 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)

では CloudFront、Amazon S3 コンテンツを保護するためにアクセスを制限するオプションがあります。を使用する場合 OAI、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 ディ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。も参照してください。これにより [OAC](#)、より詳細で拡張されたアクセスコントロールが提供されます。

ORR

[「運用準備状況レビュー」](#)を参照してください。

OT

[運用技術](#)を参照してください。

アウトバウンド (出力) 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VPCを処理するです。[AWS セキュリティリファレンスアーキテクチャ](#)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インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

P

アクセス許可の境界

ユーザーまたはロールが持つことができる最大アクセス許可を設定するために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。例としてPIIは、名前、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

[「プログラム可能な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

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。一般的には 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して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 AWS、リソースにアクセスできる のエンティティ。このエンティティは通常、IAM ロール AWS アカウント、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「ロールの用語と概念」を参照してください。https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html#id_roles_terms-and-concepts

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

Amazon Route 53 が 1 つ以上の 内のドメインとそのサブドメインのDNSクエリに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VPCs。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、プロビジョニングされる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

製品ライフサイクル管理 (PLM)

設計、開発、発売から成長と成熟、減少と削除に至るまで、ライフサイクル全体にわたる製品のデータとプロセスの管理。

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)

製造では、マシンをモニタリングし、製造プロセスを自動化する、信頼性が高く適応性の高いコンピュータです。

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。

publish/subscribe (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で、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](#)、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できるチャンネルにイベントメッセージを公開できます。システムは、パブリッシュサービスを変更せずに新しいマイクロサービスを追加できます。

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す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、説明責任、相談、情報 \(RACI\)](#) 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、説明責任、相談、情報 \(RACI\)](#) 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。

再設計

[7 Rs](#) を参照してください。

復旧ポイントの目的 (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。耐障害性、安定性、耐障害性を提供するために、それぞれ AWS リージョン が分離され、他のものとは独立しています。詳細については、[AWS リージョン 「を使用できるアカウントを指定する」](#) 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。

回復性

中断に抵抗または回復するアプリケーションの機能。[高可用性](#)と[ディザスタリカバリ](#)は、で障害耐性を計画する際の一般的な考慮事項です AWS クラウド。詳細については、[AWS クラウド「レジリエンス」](#)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。

責任、説明責任、相談、情報 (RACI) 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、行列はRASCI行列と呼ばれ、除外すると行RACI列と呼ばれます。

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

攻撃者が認証情報にアクセスすることをより困難にするために、[シークレット](#)を定期的に更新するプロセス。

行と列のアクセスコントロール (RCAC)

アクセスルールが定義されている基本的で柔軟なSQL式の使用。RCAC は、行のアクセス許可と列マスクで構成されます。

RPO

[「復旧ポイントの目的」](#)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。この機能を使用すると、フェデレーテッドシングルサインオン (SSO) AWS Management Console が有効になるため、ユーザーはログインしたり、AWS API組織内のすべてのユーザーIAMに対してユーザーを作成したりすることなく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

シヨンの詳細については、IAMドキュメントの [「2.0 SAML 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

[「監視コントロールとデータ取得」](#) 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

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。シークレット値は、バイナリ、1つの文字列、または複数の文字列にすることができます。詳細については、[Secrets Manager ドキュメントの「Secrets Manager シークレットの内容」](#) 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、[予防的](#)、[検出的](#)、[応答的 ???](#)、[およびプロアクティブ](#) の4つの主なタイプがあります。

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。

セキュリティ情報とイベント管理 (SIEM) システム

セキュリティ情報管理 (SIM) システムとセキュリティイベント管理 (SEM) 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、その他の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、アラートを生成します。

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、認証情報のローテーションなどがあります。

サーバー側の暗号化

送信先で、それ AWS のサービスを受信する 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はガードレールを定義するか、管理者がユーザーまたはロールに委任できるアクションの制限を設定します。を許可リストまたは拒否リストSCPとしてを使用して、許可または禁止されるサービスまたはアクションを指定できます。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URLの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。

サービスレベル契約 (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。

サービスレベルの目標 (SLO)

サービス[レベルインジケータ](#)によって測定される、サービスの正常性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について共有する責任を説明するモデル。クラウドのセキュリティAWSはクラウドのセキュリティに責任があり、クラウドのセキュリティはユーザーの責任です。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)

システムを混乱させる可能性のある、アプリケーションの単一の重要なコンポーネントの障害。

SLA

[「サービスレベル契約」](#)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、「」の[「アプリケーションのモダナイズに対する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。

スタースキーマ

1つの大きなファクトテーブルを使用してトランザクションデータまたは測定データを保存し、1つ以上の小さなディメンションテーブルを使用してデータ属性を保存するデータベース組織構造。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の目的で使用するため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、宿主にと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。このパターンを適用する方法の例については、「[従来の Microsoft のモダナイズ](#)」を参照してください。ASP.NET (ASMX) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に更新する「」を参照してください。

サブネット

内の IP アドレスの範囲VPC。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。

監視コントロールとデータ取得 (SCADA)

製造では、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稼働をモニタリングするシステムです。

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

ユーザーインタラクションをシミュレートして潜在的な問題を検出したり、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

VPCs とオンプレミスの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トランジットハブ。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

組織内でタスクを実行するために指定したサービスに、ユーザーに代わってそのアカウント AWS Organizations でアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要とときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、ドキュメントの「[を他の AWS のサービス AWS Organizations で使用する 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つのピザを食べることができる小さな 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VPCsできる 2 つの間の接続。詳細については、Amazon VPCドキュメントの[VPC「ピアリングとは」](#)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の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

「書き込み」を1回参照し、多くのを読み取ります。

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。

1回書き込み、多数読み込む (WORM)

データを1回書き込み、データの削除や変更を防ぐストレージモデル。認定ユーザーは、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 を利用する攻撃、通常はマルウェア。

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

平均使用量CPUとメモリ使用量が 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。