



Amazon Neptune Analytics 用の AWS Well-Architected フレームワークの適用

AWS 規範ガイドンス



AWS 規範ガイド: Amazon Neptune Analytics 用の AWS Well-Architected フレームワークの適用

Copyright © 2025 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
対象者	1
目的	2
運用上の優秀性の柱	3
IaC アプローチを使用してデプロイを自動化する	3
運用のための設計	4
小規模で可逆的な変更を頻繁に行う	4
実用的なインサイトのためのオブザーバビリティを実装する	5
すべての運用上の障害から学ぶ	6
ログ記録機能を使用して、不正または異常なアクティビティをモニタリングする	6
セキュリティの柱	7
データセキュリティの実装	7
ネットワークを保護する	8
認証と認可を実装する	8
信頼性の柱	10
Neptune サービスクォータを理解する	10
Neptune のデプロイパターンを理解する	11
Neptune クラスターの管理とスケーリング	12
バックアップとフェイルオーバーイベントを管理する	13
パフォーマンス効率の柱	14
分析のグラフモデリングを理解する	14
クエリを最適化する	16
書き込みの最適化	17
適切なサイズのグラフ	18
コスト最適化の柱	19
必要な使用パターンとサービスを理解する	19
コストに注意してリソースを選択する	21
持続可能性の柱	23
AWS リージョン 選択を検討する	23
消費の最適化	23
ソフトウェア開発とアーキテクチャのパターンを最適化する	24
リソース	26
リファレンス	26
ブログ記事と動画	26

トレーニング	26
ドキュメント履歴	27
用語集	28
#	28
A	29
B	32
C	34
D	37
E	41
F	43
G	44
H	46
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

Amazon Neptune Analytics 用の AWS Well-Architected フレームワークの適用

Michael Havey, Amazon Web Services (AWS)

2024 年 12 月 ([ドキュメント履歴](#))

[Amazon Neptune](#) を使用して、Amazon Web Services (AWS) でグラフベースのソリューションを構築できます。Neptune には、大量のグラフデータをすばやく分析してインサイトを取得し、傾向を見つけることができるメモリ最適化グラフ分析エンジンである [Neptune Analytics](#) が含まれています。既存の [Neptune データベース](#) クラスター内のデータに対して分析を実行したり、外部データセットからデータをロードして分析したりできます。このガイドでは、Neptune Analytics のデプロイを計画するときに [AWS Well-Architected フレームワーク](#) の原則を適用するための規範的なガイダンスを提供します。[Amazon Neptune 用の AWS Well-Architected フレームワークの適用](#) では、Neptune データベースの同じトピックについて説明します。

AWS Well-Architected フレームワークは、さまざまなアプリケーションやワークロード向けに、安全で高性能、耐障害性、効率的なインフラストラクチャを構築するのに役立ちます。また、アーキテクチャを評価し、スケーラブルな設計を実装するための一貫したアプローチも提供します。

AWS Well-Architected フレームワークは、次の 6 つの柱を中心に構築されています。

- オペレーショナルエクセレンス
- セキュリティ
- 信頼性
- パフォーマンス効率
- コスト最適化
- 持続可能性

このガイドでは、Well-Architected フレームワークの設計の柱とベストプラクティス、および Neptune Analytics をデプロイする際の考慮事項について説明します AWS。

対象者

このガイドは、グラフを使用するソリューションを設計および実装するデータエンジニア、ソリューションアーキテクト、データアナリストを対象としています AWS。

目的

このガイドは、ユーザーと組織が以下を実行するのに役立ちます。

- サポートされているデプロイオプションから選択します。
- 耐障害性とセキュリティの向上に役立つ AWS Well-Architected 設計パターンに従ってください。
- 最適なパフォーマンスとコスト削減のためにクエリを設計します。
- 本番環境で Neptune Analytics グラフを管理するときに運用効率を高める方法について説明します。

運用上の優秀性の柱

AWS Well-Architected フレームワークの[運用上の優秀性の柱](#)は、システムの実行とモニタリング、プロセスと手順の継続的な改善に焦点を当てています。これには、開発をサポートし、ワークロードを効果的に実行し、運用に関するインサイトを得て、ビジネス価値を提供するためのサポートプロセスと手順を継続的に改善する機能が含まれます。自己修復ワークロードは、人間の介入なしにほとんどの問題を検出して修復することで、運用の複雑さを軽減できます。このセクションで説明されているベストプラクティスに従って、この目標を達成し、Amazon Neptune Analytics のメトリクス、APIs、メカニズムを使用して、ワークロードが予想される動作から逸脱した場合に適切に対応できます。

運用上の優秀性の柱に関するこの説明では、以下の主要分野に焦点を当てています。

- Infrastructure as code (IaC)
- 変更管理
- レジリエンシー戦略
- インシデント管理
- コンプライアンスに関する監査レポート
- ログ記録とモニタリング

IaC アプローチを使用してデプロイを自動化する

IaC を使用して Neptune へのデプロイを自動化するためのベストプラクティスは次のとおりです。

- IaC を適用して Neptune Analytics グラフと関連リソースをデプロイします。一貫した環境設定を行うには、[AWS CloudFormation](#)を使用して、グラフとプライベートエンドポイントをプロビジョニングします。
- CloudFormation を使用して、[Amazon SageMaker AI で Neptune ノートブックインスタンスをプロビジョニングします](#)。ノートブックを使用して、Neptune Analytics グラフ内のデータをクエリおよび視覚化できます。
- Neptune データベースクラスターやスナップショット、Amazon Simple Storage Service (Amazon S3) でステージングされたデータファイルなど、既存のソースから Neptune Analytics グラフを作成する場合は、[一括インポートタスク](#)をモニタリングします。
- [グラフのサイズ変更](#)、グラフの削除とスナップショット作成、スナップショットからのグラフの復元、グラフのリセットと再ロードなど、Neptune Analytics の運用手順を自動化します。[AWS](#)

[Command Line Interface \(AWS CLI \)](#) または [SDK](#) から利用可能な [Neptune Analytics API](#) を使用します。

- グラフに必要な稼働時間を評価します。分析は一時的なことがよくあります。グラフはアルゴリズムの実行に必要な期間のみ必要です。この場合、AWS CLI または SDKs を使用して、不要になったグラフを[スナップショットして削除](#)します。その後、必要に応じて[スナップショットから復元](#)できます。
- 接続文字列をクライアントから外部に保存します。接続文字列は[AWS Secrets Manager](#)、[Amazon DynamoDB](#)、または動的に変更できる任意の場所に保存できます。
- [タグを使用して Neptune Analytics リソースにメタデータを追加](#)し、タグに基づいて使用状況を追跡します。タグはリソースを整理するのに役立ちます。たとえば、特定の環境またはアプリケーションのリソースに共通のタグを適用できます。タグを使用してリソース使用量の請求を分析することもできます。詳細については、AWS 「請求ユーザーガイド」の[「コスト配分タグを使用した AWS コストの整理と追跡」](#)を参照してください。さらに、AWS Identity and Access Management (IAM) ポリシーの条件を使用して、そのリソースで使用されているタグに基づいてリソースへのアクセス AWS を制御できます。これを行うには、グローバル条件キー `aws:ResourceTag/tag-key` を使用します。詳細については、IAM ユーザーガイドの[AWS 「リソースへのアクセスの制御」](#)を参照してください。

運用のための設計

Neptune Analytics グラフの操作方法を改善するアプローチを採用します。

- 開発、テスト、本番稼働用に個別の Neptune Analytics グラフを維持します。これらのグラフには、データセット、ユーザー、運用コントロールが異なる場合があります。
- さまざまな用途に合わせて個別の Neptune Analytics グラフを維持します。たとえば、2 つの分析ユーザーのグループで、タイムライン、モデル、パフォーマンスと可用性の SLAs、使用パターンが異なる個別のグラフが必要な場合は、グループごとに個別のグラフを維持します。
- Neptune Analytics [メンテナンス更新](#)のためにユーザーと運用スタッフを準備します。

小規模で可逆的な変更を頻繁に行う

以下の推奨事項は、複雑性を最小限に抑え、ワークロードの中断の可能性を減らすために実行できる小規模で可逆的な変更に焦点を当てています。

- IaC テンプレートとスクリプトを GitHub や GitLab などのソースコントロールサービスに保存します。

Important

AWS 認証情報をソース管理に保存しないでください。

- IaC デプロイでは、[AWS CodeDeploy](#)や などの継続的インテグレーションおよび継続的デリバリー (CI/CD) サービスを使用する必要があります[AWS CodeBuild](#)。本番稼働用以外の Neptune Analytics 環境でコードをコンパイル、テスト、デプロイしてから、本番稼働用グラフに昇格させます。

実用的なインサイトのためのオブザーバビリティを実装する

ワークロードの動作、パフォーマンス、信頼性、コスト、ヘルスを包括的に把握できます。以下の推奨事項は、Neptune Analytics でそのレベルの理解に役立ちます。

- Neptune Analytics の Amazon CloudWatch メトリクスをモニタリングします。これらのメトリクスから、グラフのサイズ (ノード数、エッジ数、ベクトル数、合計バイトサイズ)、CPU 使用率、クエリリクエストとエラー率を判断できます。
- NumQueuedRequestsPerSec、NumOpenCypherRequestsPerSec、などの主要なメトリクスとGraphSizeBytes、アプリケーションログにある CPUUtilization Neptune GraphStorageUsagePercentクライアントレスポンスの CloudWatch ダッシュボードとアラームを作成します。
- 通知を設定して、グラフサイズ、リクエストレート、CPU 使用率がしきい値を超えたときなど、Neptune Analytics グラフの状態をモニタリングします。例えば、グラフでGraphStorageUsagePercentが 90% まで大幅に拡大する場合は、メモリ最適化 Neptune キャパシティユニット (m-NCU) 容量を増やすかどうかを決定します。現在の m-NCU が 128 の場合、256 に増やすとストレージが約 45% 減少します。NumQueuedRequestsPerSec が 0 より大きいことが多い場合は、m-NCU 容量を増やしてコンピューティング容量を増やすことを検討してください。または、クライアント側の同時実行数を減らすこともできます。

すべての運用上の障害から学ぶ

自己修復インフラストラクチャは、まれな問題が発生したり、対応が意図したほど効果的でなくなったりすると、反復して発生する長期的な労力です。以下のプラクティスを採用することで、その目標に焦点を当てることができます。

- すべての障害から学習することで、改善を推進します。
- チームや組織全体で学んだことを共有します。組織内の複数のチームが Neptune を使用している場合は、共通のチャットルームまたはユーザーグループを作成して、学習とベストプラクティスを共有します。

ログ記録機能を使用して、不正または異常なアクティビティをモニタリングする

ログ記録を使用して、異常なパフォーマンスとアクティビティパターンを監視します。以下のベストプラクティスを考慮します。

- Neptune Analytics は、を使用してコントロールプレーンアクションのログ記録をサポートします AWS CloudTrail。詳細については、「[を使用した Neptune Analytics API コールのログ記録 AWS CloudTrail](#)」を参照してください。この機能を使用すると、Neptune Analytics リソースの作成、更新、削除を追跡できます。堅牢なモニタリングとアラートのために、CloudTrail イベントを [Amazon CloudWatch Logs](#) と統合することもできます。Neptune Analytics サービスアクティビティの分析を強化し、のアクティビティの変更を特定するには AWS アカウント、[Amazon Athena を使用して CloudTrail ログをクエリ](#)できます。例えば、クエリを使用して傾向を識別したり、アクティビティを属性 (ソース IP アドレスやユーザーなど) でさらに分離したりできます。
- CloudTrail を使用して、クエリ実行などの [Neptune Analytics データプレーンアクティビティのログ記録を有効にする](#)こともできます。実行中のクエリ、その頻度、およびソースを表示できます。デフォルトでは、CloudTrail はデータイベントをログ記録しません。追加の変更がイベントデータに適用されます。詳細については、[AWS CloudTrail 料金表](#)を参照してください。
- コントロールプレーンまたはデータプレーンのいずれかで、Neptune Analytics へのアプリケーション呼び出しを記録することもできます。たとえば、を使用してクエリ [AWS SDK for Python \(Boto3\)](#)を行う場合、[デバッグレベルのログ記録を有効に](#)して、コンソールまたはファイルへのクエリのトレースを取得できます。これは開発時に便利です。また、アプリケーションから例外をキャッチしてログに記録することをお勧めします。

セキュリティの柱

クラウドセキュリティが最優先事項です AWS。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。セキュリティは、AWSとお客様の間で共有される責任です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ – AWS は、AWS のサービス で実行されるインフラストラクチャを保護する責任を担います AWS クラウド。AWS また、 は、安全に使用できるサービスも提供します。サードパーティーの監査者は、[AWS コンプライアンスプログラム](#)の一環として、AWS セキュリティの有効性を定期的にテストおよび検証します。Neptune に適用されるコンプライアンスプログラムの詳細については、[AWS 「コンプライアンスプログラムによる対象範囲内のサービス」](#)を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する によって決まり AWS のサービス ます。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。データプライバシーの詳細については、「[データプライバシーに関するFAQs](#)」を参照してください。欧州でのデータ保護の詳細については、責任[AWS 共有モデルと GDPR](#) ブログ記事を参照してください。

AWS Well-Architected フレームワークの[セキュリティの柱](#)は、Neptune Analytics を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目的を達成するために Neptune Analytics を設定する方法について説明します。また、Neptune Analytics リソースのモニタリングと保護 AWS のサービス に役立つ他の の使用方法についても説明します。セキュリティの柱には、以下の主要な重点領域が含まれています。

- データセキュリティ
- ネットワークセキュリティ
- 認証と認可

データセキュリティの実装

データ漏洩や違反は、顧客を危険にさらし、会社に大きな悪影響を与える可能性があります。以下のベストプラクティスは、不注意や悪意のある漏洩から顧客データを保護するのに役立ちます。

- グラフ名、タグ、IAM ロール、およびその他のメタデータには、機密情報や機密情報を含めないでください。データは請求ログや診断ログに表示される可能性があるためです。

- Neptune にデータとして保存されている外部サーバーへの URIs またはリンクには、リクエストを検証するための認証情報を含めないでください。
- Neptune Analytics グラフは保管時に暗号化されます。選択したデフォルトキーまたは AWS Key Management Service (AWS KMS) キーを使用して、グラフを暗号化できます。一括インポート中に Amazon S3 にエクスポートされるスナップショットとデータを暗号化することもできます。インポートが完了したら、暗号化を削除できます。
- openCypher 言語を使用する場合は、SQL インジェクションやその他の形式の攻撃を防ぐために、適切な入力検証と[パラメータ化](#)の手法を実践してください。ユーザーが指定した入力で文字列連結を使用するクエリを構築しないでください。パラメータ化されたクエリまたは準備済みステートメントを使用して、入力パラメータをグラフデータベースに安全に渡します。詳細については、Neptune ドキュメントの[openCypher パラメータ化されたクエリの例](#)」を参照してください。

ネットワークを保護する

パブリック接続用の Neptune Analytics グラフを有効にして、Virtual Private Cloud (VPC) の外部からアクセスできるようにします。この接続はデフォルトで無効になっています。グラフには IAM 認証が必要です。発信者は ID を取得し、グラフを使用するアクセス許可を持っている必要があります。たとえば、[openCypher クエリを実行するには](#)、呼び出し元に特定のグラフに対する読み取り、書き込み、または削除のアクセス許可が必要です。

グラフの[プライベートエンドポイントを作成して](#)、VPC 内からグラフにアクセスすることもできます。エンドポイントを作成するときは、VPC、サブネット、セキュリティグループを指定して、グラフを呼び出すアクセスを制限します。

転送中のデータを保護するために、Neptune Analytics は HTTPS 経由でグラフに SSL 接続を適用します。詳細については、[Neptune Analytics ドキュメントの「Neptune Analytics でのデータ保護」](#)を参照してください。

認証と認可を実装する

Neptune Analytics グラフを呼び出すには、IAM 認証が必要です。呼び出し元は ID を取得し、グラフでアクションを実行するための十分なアクセス許可を持っている必要があります。API アクションとその必要なアクセス許可の詳細については、[Neptune Analytics API ドキュメント](#)を参照してください。[条件チェックを適用](#)して、タグによるアクセスを制限できます。

IAM 認証では、[AWS 署名バージョン 4 \(SigV4\) プロトコル](#)を使用します。アプリケーションからの使用を簡素化するために、[AWS SDK](#)を使用することをお勧めします。たとえば、Python では、SigV4 を抽象化する [Neptune Graph の Boto3 クライアント](#)を使用します。

グラフにデータをロードすると、[バッチロード](#)は発信者の IAM 認証情報を使用します。呼び出し元には、Neptune Analytics が Amazon S3 ファイルからグラフにデータをロードするロールを引き受けられるように、信頼関係を設定して Amazon S3 からデータをダウンロードするアクセス許可が必要です。

[一括インポート](#)は、グラフの作成時 (インフラストラクチャチーム) または既存の空のグラフ上 (インポートタスクを開始する権限を持つデータエンジニアリングチーム) に実行できます。どちらの場合も、Neptune Analytics は発信者が入力として提供する IAM ロールを引き受けます。このロールは、入力データがステージングされている Amazon S3 フォルダの内容を読み取って一覧表示するアクセス許可を付与します。

信頼性の柱

AWS Well-Architected フレームワークの[信頼性の柱](#)には、ワークロードが意図した機能を正しく一貫して実行する能力が含まれます。これには、ライフサイクル全体を通じてワークロードを運用およびテストする機能が含まれます。

信頼性の高いワークロードの実現は、ソフトウェアとインフラストラクチャの両方について事前に設計を決定することから始まります。アーキテクチャの選択は、Well-Architected のすべての柱にわたってワークロードの動作に影響します。信頼性のために、このセクションで説明するように、従う必要がある特定のパターンがあります。

信頼性の柱は、次の主要分野に焦点を当てています。

- サービスクォータやデプロイパターンなどのワークロードアーキテクチャ
- 変更管理
- 障害管理

Neptune サービスクォータを理解する

AWS アカウントには、ごとにデフォルトのクォータ (以前は制限と呼ばれていました) があります。AWS のサービス。特に明記していない限り、クォータはリージョン固有です。一部のクォータの引き上げをリクエストできますが、すべてではありません。

Neptune Analytics のクォータを確認するには、[Service Quotas コンソール](#)を開きます。ナビゲーションペインで、 を選択しAWS のサービス、Amazon Neptune Analytics を選択します。グラフとスナップショットの数、グラフの最大プロビジョニングメモリ、API リクエストレートのクォータに注意してください。

最大プロビジョニングメモリがデータセットに十分でない場合は、意図した分析用途に不可欠なノードとエッジタイプを評価します。データのサブセットをロードして、許容されるプロビジョニングされた容量内で分析できるようにします。多くの分析ワークロード、特にグラフアルゴリズムを実行するワークロードでは、完全なトランザクショングラフではなく、プロパティのセットが制限されたトポロジのみが必要です。(トランザクションワークロードと分析ワークロードの違いについては、[パフォーマンス効率の柱](#)セクションを参照してください)。

グラフの最大数が意図した用途に十分でない場合：

- 同様の用途を持つグラフを組み合わせて検討してください。
- 一度に実行する必要があるグラフの数を評価します。エフェメラル分析のユースケースがある場合は、不要になったグラフをスナップショットして削除します。これにより、クォータに対するグラフの数が減少します。
- 異なる でグラフをプロビジョニングすることを検討してください AWS アカウント。

Neptune のデプロイパターンを理解する

Neptune Analytics グラフをデプロイする場合は、次の決定点を理解します。

- シード: Amazon S3、既存の Neptune データベースクラスター、または既存の Neptune データベーススナップショットのデータを使用して、空のグラフを作成するか、作成時にそのグラフにデータをロードするかを決定します。

推奨事項: ソースが Neptune クラスターまたはスナップショットの場合は、グラフ作成時にデータをロードする必要があります。ソースが Amazon S3 の場合、データをロードする労力が重要で、インフラストラクチャのプロビジョニングアクティビティとして最適である場合は、作成時にデータをロードします。データエンジニアリングまたはアプリケーションアクティビティとしてデータをロードする場合は、空のグラフを作成し、後で Amazon S3 からデータをロードします。

- 容量: データサイズと予想されるアプリケーション使用量を考慮して、グラフに必要なプロビジョニングされた容量を見積もります。

推奨事項: 作成時に、グラフサイズを制限する [最大プロビジョニングメモリを指定します](#)。この設定は必須です。必要に応じて、後で容量を変更できます。

- 可用性と耐障害性: 可用性にレプリカが必要かどうかを決定します。レプリカは、グラフに障害が発生した場合の復旧のためのウォームスタンバイとして機能します。レプリカを含むグラフは、レプリカを含まないグラフよりも速く回復します。また、グラフが必要な期間、エフェメラル分析専用かどうか、必要な場合はいつ削除されるかも考慮してください。

推奨事項: グラフを作成する前に、グラフを使用できない時間や削除できる時間などの可用性要件を決定します。

- ネットワークとセキュリティ: パブリック接続、プライベート接続、またはその両方が必要かどうか、およびデータを暗号化するかどうかを決定します。

推奨事項: グラフを作成する前に、パブリック接続が許可されているかどうか、グラフクライアントアプリケーションがデプロイされる場所など、組織の要件を理解します。

- バックアップとリカバリ: スナップショットを作成するかどうか、作成する場合は、いつ、どの条件でスナップショットを作成するかを決定します。組織にディザスタリカバリ (DR) 要件があるかどうかを検討します。

推奨事項: スナップショットの作成は手動アクティビティです。スナップショットを作成するタイミングを決定し、グラフを作成する前に DR 要件を検討します。

Neptune クラスターの管理とスケーリング

Neptune Analytics グラフは、単一のメモリ最適化インスタンスで構成されます。インスタンスの容量 (m-NCU) は作成時に設定されます。インスタンスは、[管理アクション](#)を通じてプロビジョニングされた容量を増やすことで垂直方向にスケーリングできます。また、プロビジョニングされた容量を減らすこともできます。レプリカはパッシブフェイルオーバーターゲットであるため、グラフのスケールは増加しません。この点で、グラフレプリカは、アプリケーションからの読み取りオペレーションを処理できる [Neptune クラスター内のアクティブなインスタンスである Neptune データベースリードレプリカ](#)とは異なります。

レプリカにはコストがかかります。レプリカの料金はグラフの m-NCU レートです。たとえば、グラフが 128 m-NCU に対してプロビジョニングされ、レプリカが 1 つある場合、コストはレプリカを持たない同等のグラフの 2 倍になります。

分析では、スケールアップの主な理由が 2 つあります。

- 分析クエリとアルゴリズムにより多くのメモリと CPU を提供するために、個々のクエリは高価であるため、実行するグラフアルゴリズムは本質的に複雑で、入力を考慮するとより多くのリソースを必要とするか、同時リクエストレートが高いです。このようなクエリで out-of-memory エラーが発生した場合は、スケールアップが妥当な解決策です。
- 計画よりも大きいグラフサイズをサポートするには。たとえば、現在のプロビジョニングされた容量が 60 GB のソースデータをサポートするために 128 m-NCU で、さらに 40 GB のソースデータが必要な場合は、256 m-NCU への引き上げが必要です。

NumQueuedRequestsPerSec、、、、などの Neptune Analytics

NumOpenCypherRequestsPerSec の CloudWatch GraphStorageUsagePercent メトリクスをモニタリングして GraphSizeBytesCPUUtilization、スケーリングが必要かどうかを判断します。グラフの設定は、コンソール AWS CLI、または SDKs を使用して更新できます。(例とベストプラクティスについては、[「オペレーショナルエクセレンスの柱」](#)セクションを参照してください)。

バックアップとフェイルオーバーイベントを管理する

レプリカを使用して、障害発生時にグラフを使用できるようにします。グラフはログベースの永続性を使用して、 のアベイラビリティゾーン間で変更をコミットします AWS リージョン。レプリカはウォームスタンバイとして機能し、このデータにアクセスできます。失敗した場合、グラフはレプリカのオペレーションを再開します。アプリケーションは引き続き同じエンドポイントを使用してグラフに接続します。失敗中の処理中のリクエストは、サービス利用不可の例外を含むエラーを生成します。アプリケーションコードの [バックオフパターンで再試行](#)してエラーをキャッチし、短い間隔で再試行することを検討してください。フェイルオーバー中に行われた新しいリクエストはキューに入れられ、レイテンシーが長くなる可能性があります。

レプリカが設定されておらず、グラフが失敗した場合、Neptune Analytics は耐久性のあるストレージから復旧しますが、Neptune がリソースを再初期化する必要があるため、復旧に時間がかかります。

グラフのスナップショットを作成します。(Neptune Analytics は自動スナップショットを作成しません)。作成後にグラフが定期的に変更される場合は、スナップショットを頻繁に作成して現在の状態をキャプチャします。以前の時点への復元が必要ない場合は、古いスナップショットを削除します。

スナップショットは、他の アカウントと共有できます AWS リージョン。DR 要件がある場合は、スナップショットから別のリージョンのグラフを復元することが、目標復旧時間 (RTO) と目標復旧時点 (RPO) の要件を満たしているかどうかを検討してください。

パフォーマンス効率の柱

AWS Well-Architected フレームワークの[パフォーマンス効率の柱](#)は、データの取り込みまたはクエリ中にパフォーマンスを最適化する方法に焦点を当てています。パフォーマンスの最適化は、以下の段階的かつ継続的なプロセスです。

- ビジネス要件の確認
- ワークロードのパフォーマンスの測定
- パフォーマンスの低いコンポーネントを特定する
- ビジネスニーズに合わせてコンポーネントを調整する

パフォーマンス効率の柱は、使用する適切なグラフデータモデルとクエリ言語を特定するのに役立つユースケース固有のガイドラインを提供します。また、Neptune Analytics にデータを取り込み、Neptune Analytics からデータを使用する際に従うべきベストプラクティスも含まれています。

パフォーマンス効率の柱は、次の主要分野に焦点を当てています。

- グラフモデリング
- クエリの最適化
- グラフの正しいサイズ設定
- 書き込みの最適化

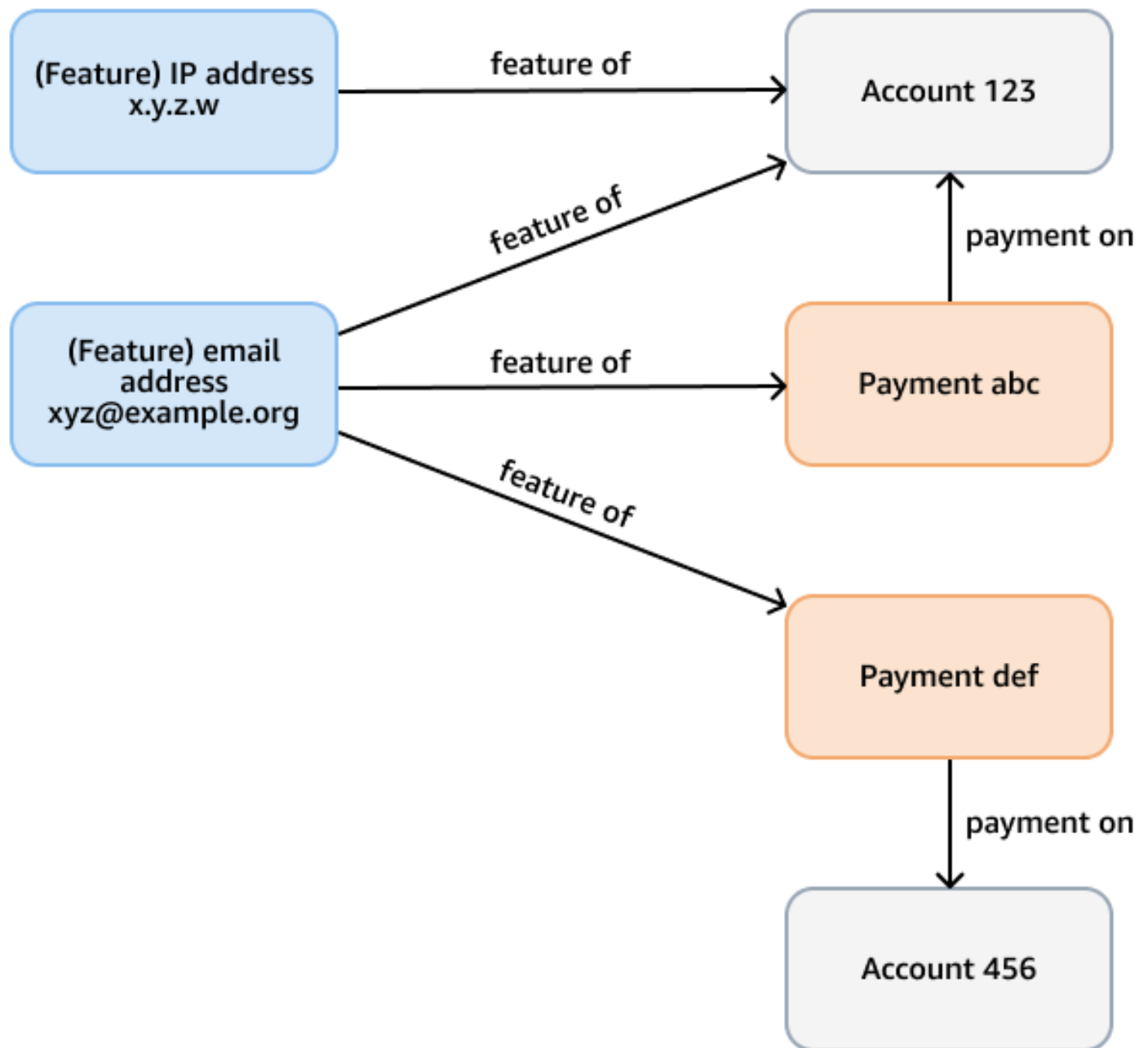
分析のグラフモデリングを理解する

Amazon Neptune 用の AWS Well-Architected フレームワークの適用」ガイドでは、[パフォーマンス効率のためのグラフモデリング](#)について説明します。パフォーマンスに影響するモデリングの決定には、必要なノードとエッジ、IDs、ラベルとプロパティ、エッジの方向、ラベルが汎用的か固有か、一般的にクエリエンジンがグラフをナビゲートして一般的なクエリを処理できる効率の選択が含まれます。

これらの考慮事項は Neptune Analytics にも適用されますが、トランザクション使用パターンと分析使用パターンを区別することが重要です。Neptune データベースなどのトランザクションデータベース内のクエリに効率的なグラフモデルは、分析のために再構築する必要がある場合があります。

たとえば、Neptune データベースの不正グラフで、クレジットカード支払いの不正パターンをチェックする目的があるとします。このグラフには、アカウントと支払いの両方のアカウント、支払

い、機能 (E メールアドレス、IP アドレス、電話番号など) を表すノードがある場合があります。この接続されたグラフは、特定の支払いから始まり、関連する機能やアカウントを見つけるためにいくつかのホップを取る可変長パスをトラバースするなどのクエリをサポートします。次の図は、このようなグラフを示しています。



分析要件は、特徴によってリンクされたアカウントのコミュニティを検索するなど、より具体的な場合があります。この目的には、[弱に接続されたコンポーネント \(WCC\)](#) アルゴリズムを使用できます。前の例のモデルに対して実行するには、いくつかの異なるタイプのノードとエッジを通過する必要があります。次の図のモデルはより効率的です。アカウント自体、またはアカ

アカウントからの支払いが機能を共有する場合、accountノードをshares featureエッジにリンクします。たとえば、Account 123にはEメール機能がありxyz@example.org、支払い()に同じEメールAccount 456を使用しますPayment def。



WCC の計算の複雑さは $O(|E|\log D)$ です。ここで $|E|$ はグラフ内のエッジ数、 D はノードを接続する直径 (最長パスの長さ) です。トランザクションモデルはイネスシャルノードとエッジを省略するため、エッジの数と直径の両方を最適化し、WCC アルゴリズムの複雑さを軽減します。

Neptune Analytics を使用する場合は、必要なアルゴリズムと分析クエリから戻してください。必要に応じて、モデルを再構築してこれらのクエリを最適化します。グラフにデータをロードしたり、グラフ内の既存のデータを変更するクエリを記述する前に、モデルを再構築できます。

クエリを最適化する

Neptune Analytics クエリを最適化するには、次の推奨事項に従ってください。

- [パラメータ化されたクエリ](#)と、デフォルトで有効になっている[クエリプランキャッシュ](#)を使用します。プランキャッシュを使用すると、クエリが 100 ミリ秒以内に完了すれば、エンジンは後でできるようにクエリを準備します。これにより、後続の呼び出しにかかる時間を節約できます。
- スロークエリの場合は、[説明プラン](#)を実行してボトルネックを特定し、それに応じて改善を行います。
- [ベクトル類似度検索](#)を使用する場合は、小さな埋め込みで正確な類似度の結果が得られるかどうかを決定します。小さな埋め込みをより効率的に作成、保存、検索できます。
- Neptune Analytics で [openCypher を使用するための文書化されたベストプラクティスに従います](#)。たとえば、[UNWIND 句でフラット化されたマップを使用し](#)、[可能な場合はエッジラベルを指定します](#)。
- [グラフアルゴリズム](#)を使用する場合は、アルゴリズムの入力と出力、計算の複雑さ、およびそれがどのように機能するかを広く理解します。
 - グラフアルゴリズムを呼び出す前に、MATCH句を使用して入力ノードセットを最小化します。例えば、[幅優先検索 \(BFS\)](#) を実行するノードを制限するには、Neptune Analytics ドキュメントに記載されている[例](#)に従います。

- 可能であれば、ノードラベルとエッジラベルでフィルタリングします。たとえば、BFS には、特定のノードラベル (vertexLabel) または特定のエッジラベル () へのトラバーサルをフィルタリングする入力パラメータがあります edgeLabels。
- 結果を制限する maxDepth には、などの境界パラメータを使用します。
- concurrency パラメータを試します。値 0 で試し、使用可能なすべてのアルゴリズムスレッドを使用して処理を並列化します。パラメータを 1 に設定して、シングルスレッド実行と比較します。アルゴリズムは 1 つのスレッドでより速く完了できます。特に、並列処理で実行時間が測定可能なほど短縮されず、オーバーヘッドが発生する可能性がある、幅が浅い最初の検索など、入力が小さい場合はなおさらです。
- 同様のタイプのアルゴリズムから選択します。たとえば、[Bellman-Ford とデルタステッピング](#) はどちらも単一ソースの最短パスアルゴリズムです。独自のデータセットでテストする場合は、両方のアルゴリズムを試し、結果を比較します。デルタステッピングは、計算の複雑さが低いため、通常、Bellman-Ford よりも高速です。ただし、パフォーマンスはデータセットと入力パラメータ、特に delta パラメータによって異なります。

書き込みの最適化

Neptune Analytics で書き込みオペレーションを最適化するには、次のプラクティスに従います。

- グラフにデータをロードする最も効率的な方法を探します。Amazon S3 のデータからロードする場合は、データが 50 GB を超える場合は [一括インポート](#) を使用します。小さいデータの場合は、[バッチロード](#) を使用します。バッチロードの実行時に out-of-memory エラーが発生した場合は、m-NCU 値を増やすか、ロードを複数のリクエストに分割することを検討してください。これを実現する 1 つの方法は、S3 バケット内の複数のプレフィックスにファイルを分割することです。その場合は、プレフィックスごとにバッチロードを個別に呼び出します。
- 一括インポートまたはバッチローダーを使用して、グラフデータの初期セットを入力します。トランザクション openCypher の作成、更新、削除オペレーションは、小さな変更に対してのみ使用します。
- 一括インポートまたは同時実行数が 1 (シングルスレッド) のバッチローダーを使用して、グラフに埋め込みを取り込みます。これらの方法のいずれかを使用して、埋め込みを事前にロードしてみてください。
- ベクトル類似度検索アルゴリズムの正確な類似度検索に必要なベクトル埋め込みのディメンションを評価します。可能であれば、より小さなディメンションを使用します。これにより、埋め込みのロード速度が向上します。

- 必要に応じて、ミューテーションアルゴリズムを使用してアルゴリズムの結果を記憶します。たとえば、[度可変中心性アルゴリズム](#)は、各入力ノードの度合いを検出し、その値をノードのプロパティとして書き込みます。これらのノードを囲む接続がその後変更されない場合、プロパティは正しい結果を保持します。アルゴリズムを再度実行する必要はありません。
- [グラフリセット管理アクション](#)を使用して、やり直す必要がある場合は、すべてのノード、エッジ、埋め込みをクリアします。openCypher クエリを使用してすべてのノード、エッジ、埋め込みを削除することは、グラフが大きい場合は実行できません。大規模なデータセットに対する 1 回のドロップクエリはタイムアウトする可能性があります。サイズが大きくなると、データセットの削除に時間がかかり、トランザクションサイズが大きくなります。対照的に、グラフのリセットを完了する時間はほぼ一定であり、アクションには、スナップショットを実行する前にスナップショットを作成するオプションがあります。

適切なサイズのグラフ

全体的なパフォーマンスは、Neptune Analytics グラフのプロビジョニングされた容量によって異なります。容量は、メモリ最適化 Neptune 容量単位 (m-NCUs)。グラフのサイズとクエリをサポートするのに十分なサイズがグラフにあることを確認してください。容量を増やしても、個々のクエリのパフォーマンスは必ずしも向上しないことに注意してください。

可能であれば、Amazon S3 や既存の Neptune クラスターまたはスナップショットなどの既存のソースからデータをインポートしてグラフを作成します。[制限は、最小容量と最大容量に設定できます](#)。既存のグラフで[プロビジョニングされた容量を変更](#)することもできます。

NumQueuedRequestsPerSec、NumOpenCypherRequestsPerSec、GraphStorageUsagePercent、などの CloudWatch メトリクスをモニタリングCPUUtilizationしてGraphSizeBytes、グラフのサイズが適切かどうかを評価します。グラフのサイズと負荷をサポートするために、より多くの容量が必要かどうかを判断します。これらのメトリクスの一部を解釈する方法の詳細については、「[オペレーショナルエクセレンスの柱](#)」セクションを参照してください。

コスト最適化の柱

AWS Well-Architected フレームワークの[コスト最適化の柱](#)は、不要なコストを回避することに重点を置いています。以下の推奨事項は、Neptune Analytics のコスト最適化設計原則とアーキテクチャのベストプラクティスを満たすのに役立ちます。

コスト最適化の柱は、次の主要分野に焦点を当てています。

- 時間の経過に伴う支出と資金配分の制御について
- 適切なタイプと数量のリソースの選択
- 過剰な支出なしでビジネスニーズを満たすスケーリング

必要な使用パターンとサービスを理解する

Neptune Analytics を採用する前に、ユースケースがグラフ分析に適しているかどうかを評価してください。

- **グラフデータベース:** Neptune などのグラフデータベースは、データモデルに識別可能なグラフ構造があり、クエリが関係を調べて複数のホップをトラバースする必要がある場合、ワークロードに適しています。グラフデータベースは、次のパターンには適していません。
 - 主にシングルホップクエリ。このユースケースでは、データがオブジェクトの属性としてより適切に表されるかどうかを検討してください。
 - プロパティとして保存される JSON またはバイナリラージオブジェクト (BLOB) データ。
- **グラフ分析:** Neptune Analytics は、大量のグラフデータをメモリ内ですばやく分析してインサイトを取得し、傾向を見つけることができるグラフ分析データベースエンジンです。Neptune データベースと Neptune Analytics グラフの両方にグラフデータを保存およびクエリできます。Neptune データベースは、スケーラブルなオンライントランザクション処理 (OLTP) のニーズに最適です。Neptune Analytics は、エフェメラル分析ワークロードに最適です。この 2 つを組み合わせるには、トランザクション指向の Neptune データベースから Neptune Analytics グラフにデータをロードして、そのデータの分析を実行します。分析が完了したら、Neptune Analytics グラフを削除できます。詳細については、[Neptune Analytics ドキュメントの「Neptune Analytics を使用するタイミング」](#)と「[Neptune Database を使用するタイミング](#)」を参照してください。

コストに注意して、Neptune Analytics グラフに入力する最適な方法を決定します。

- S3 バケットにステージングされたグラフデータを [一括インポート](#) します。このオプションは、データが以前に Neptune データベースへの一括ロード用にステージングされていた場合、または一括インポートに必要な [CSV またはその他のサポートされている形式](#) で分析されるデータがすでにあるか、簡単に生成できる場合にお勧めします。グラフ作成手順の一部として一括インポートを実行できます。 [最小容量と最大容量に境界を設定できます](#)。 [以前に作成した空のグラフでインポートを実行し](#)、実行中 [にインポートタスクをモニタリング](#) することもできます。
- 空のグラフを作成し、 [バッチロード](#) を使用して openCypher クエリで入力できます。このオプションは、ロードするデータが Amazon S3 でステージングされ、50 GB 未満である場合に最適です。
- [Neptune データベースクラスターのデータからグラフを入力](#) できます (Neptune データベースバージョン 1.3.0 以降でサポートされています)。このパターンの目的は、現在グラフデータベースにあるデータに対して分析を実行することです。データベースが最初の一括ロードで入力された場合でも、それ以後大幅に変更された可能性があります。データベースからインポートするために、Neptune Analytics はデータベースのクローンを作成し、クローンから S3 バケットにデータをエクスポートします。この手順にはコストが発生します。特に、クローンを実行するための Neptune データベースコストと、エクスポートされたデータを保存および消費するための Amazon S3 コストです。エクスポートが完了すると、クローンは削除されます。Amazon S3 でエクスポートされたデータを削除できます。
- [Neptune データベースクラスターのスナップショットからグラフを入力](#) できます。これは、ソースがデータベーススナップショットである点を除いて、前のオプションと似ています。スナップショットからインポートするには、Neptune Analytics はまずスナップショットを新しいデータベースクラスターに復元し、次にデータを S3 バケットにエクスポートします。この手順ではコストが発生します。特に、復元されたクラスターを実行するための Neptune データベースコストと、エクスポートされたデータの保存と消費のための Amazon S3 コストが発生します。
- openCypher クエリを実行して、グラフでアトミック性、整合性、分離、耐久性 (ACID) 準拠のトランザクションを使用してデータを作成、更新、または削除することもできます。このアプローチは、小さな更新を行う方法としてお勧めしますが、グラフをシードする方法としてはお勧めしません。

分析に必要なデータが Amazon S3 ですでにステージングされている場合は、一括インポートまたはバッチロードをお勧めします。これらは、Neptune データベースクラスターまたはスナップショットからグラフを入力するよりもコスト効率が高くなります。

コストに注意してリソースを選択する

[Neptune Analytics の料金](#)は、メモリ最適化 Neptune キャパシティーユニット (m-NCU) と呼ばれる単位を使用します。特定の m-NCU でグラフを実行する場合、時間単位の固定コストが発生します。グラフにはフェイルオーバー用のレプリカが含まれている場合があり、これらのレプリカには時間単位の m-NCU コストも発生します。

容量の見積もり、コストの制限、パフォーマンスに対するコストのモニタリングには、次のベストプラクティスをお勧めします。

- 可能であれば、既存のソースからデータをインポートしてグラフを作成します。Amazon S3 または既存の Neptune クラスターまたはスナップショットでステージングされたデータです。これにより、Neptune Analytics はグラフのシードの重労働を実行し、[バインドされた最大容量を指定できるため](#)、労力を節約できます。
- 既存のグラフで[プロビジョニングされた容量を変更できます](#)。
- グラフが不要になったら、[スナップショットを作成してグラフを削除](#)できます。再度使用する必要がある場合は、スナップショットからグラフを復元できます。
- グラフの作成時にレプリカの数を選択できます。分析の可用性要件に従って値を設定します。この設定を最小化してコストを節約します。最大値 2 では、別々のアベイラビリティゾーンに 2 つのレプリカインスタンスを使用できます。最小値 0 は、Neptune Analytics がレプリカを実行しないことを意味します。ただし、レプリカが利用可能な場合、復旧が速くなります。グラフの障害と復旧の説明については、「[信頼性の柱](#)」セクションを参照してください。
- を使用して、現在および過去の請求期間の Neptune Analytics の費用をモニタリングします[AWS Billing and Cost Management](#)。
- CloudWatch、特に NumQueuedRequestsPerSec、NumOpenCypherRequestsPerSec、GraphStorageUsagePercent および GraphSizeBytes の Neptune Analytics メトリクスをモニタリングして CPU Utilization、プロビジョニングされた容量がグラフに対して適切なサイズであるかどうかを評価します。小さい容量が観測されたリクエストレート、CPU 使用率、グラフサイズに対応できるかどうかを確認します。
- グラフにプライベートエンドポイントが必要な場合は、Elastic IPs、Virtual Private Cloud (VPC) エンドポイント、NAT ゲートウェイ、またはその他の VPC 関連のコストに注意してください。詳細については、「[Amazon VPC の料金](#)」と[Amazon EC2 の料金](#)を参照してください。
- 開発者やアナリストがグラフをクエリおよび視覚化するのに役立つクライアントインターフェイスを提供するために、1 つ以上の Neptune ノートブックインスタンスを実行することもできます（「[Neptune ワークベンチの料金](#)」を参照）。コストを最小限に抑えるには、ユーザー間でインス

タンスを共有し、ユーザーごとに個別のノートブックフォルダを作成します。インスタンスが使用されていない場合はシャットダウンします。シャットダウンを自動化する方法については、ブログ記事 [AWS 「リソースタグを使用して Amazon Neptune 環境リソースの停止と開始を自動化する」](#) を参照してください。

持続可能性の柱

AWS Well-Architected フレームワークの[持続可能性の柱](#)は、クラウドワークロードの実行による環境への影響を最小限に抑えることに重点を置いています。主なトピックには、持続可能性に関する責任共有モデル、影響の理解、必要なリソースの最小化とダウンストリームへの影響の軽減のための使用の最大化が含まれます。

持続可能性の柱には、以下の主要な重点分野が含まれています。

- 影響
- 持続可能性の目標
- 最大使用量
- より効率的な新しいソフトウェアの提供を予測して採用する
- マネージドサービスの使用
- ダウンストリームの影響軽減

このガイドでは、影響の理解に焦点を当てています。その他の持続可能性設計原則の詳細については、[AWS「Well-Architected フレームワーク」](#)を参照してください。

選択と要件は環境に影響します。炭素強度 AWS リージョン の低い を選択し、稼働時間と耐久性を最大化するだけでなく、実際のワークロードのニーズを要件に反映すると、ワークロードの持続可能性が向上します。次のセクションでは、ワークロード設計と継続的な運用に採用された場合に環境に悪影響を及ぼすベストプラクティスと考慮事項について説明します。

AWS リージョン 選択を検討する

Amazon 再生可能エネルギープロジェクトの近く AWS リージョン にあるものや、グリッドの炭素強度が他のものよりも低い場所にあるものもあります。ワークロードに対して実行可能なリージョンの[持続可能性への影響](#)を考慮し、[Neptune Analytics が利用可能なリージョンとリストを相互参照します。](#)

消費の最適化

以下を実践することで、Neptune Analytics の消費を最小限に抑えます。

- 多くの場合、分析は一時的なものです。グラフは、アルゴリズムを実行して結果を記録する時間にものみ必要です。この場合、不要になったグラフを[スナップショットして削除](#)します。必要に応じて、[後でスナップショットから復元](#)できます。
- ワークロードが一時的なもので、分析をいつ実行するかを決定する柔軟性がある場合は、電力消費量の day-to-day 傾向を考慮してください。電力の需要は、特定の時間帯に高くなります。米国にお住まいの場合は、米国エネルギー情報局 (EIA) のウェブサイトでの[毎日の電力消費量に関するメトリクス](#)を参照してください。可能であれば、リージョンのオフピーク期間中にワークロードを実行します。
- ワークロードがエフェメラルではなく、限られた期間のみ使用する必要がある場合は、グラフを削除し、必要に応じてスナップショットから復元します。可用性がスケジュールに従っている場合は、スクリプトを使用して復元プロセスを自動化し、スケジュールされた時刻にグラフの準備を整えます。
- データが読み取り専用であるか、前回のスナップショット以降に変更されていない場合は、削除前に再度スナップショットを作成しないでください。
- 使用していない Neptune ノートブックを停止します。
- NumQueuedRequestsPerSec、NumOpenCypherRequestsPerSec、GraphStorageUsagePercent などの CloudWatch メトリクスをモニタリング CPU Utilization して GraphSizeBytes、グラフのサイズが大きすぎるかどうかを評価します。小さいインスタンス容量が観測されたリクエストレート、CPU 使用率、グラフサイズに対応できるかどうかを確認します。

ソフトウェア開発とアーキテクチャのパターンを最適化する

無駄を防ぐには、モデルとクエリを最適化し、コンピューティングリソースを共有して、Neptune インスタンスとクラスターで利用できるすべてのリソースを使用します。具体的なベストプラクティスは次のとおりです。

- クエリとグラフアルゴリズムの呼び出しを最適化します。パラメータ化されたクエリを使用し、デフォルトで有効になっている[クエリプランキャッシュを使用します](#)。スロークエリの場合は、[説明プラン](#)を実行して改善を行います。[ベクトル類似度検索](#)を使用する場合は、小さい埋め込みの方がより効率的に作成、保存、検索できるため、小さい埋め込みの方が正確な類似度結果が得られるかどうかを判断します。[グラフアルゴリズム](#)を呼び出す前に、MATCH 句を使用して入力ノードセットを最小化します。可能であれば、ノードラベルとエッジラベルでフィルタリングします。
- グラフにデータをロードする最も効率的な方法を探します。Amazon S3 のデータからロードする場合は、データのサイズが 50 GB を超える場合は[一括インポート](#)を使用します。小さいデータには[バッチロード](#)を使用します。

- 開発者に、それぞれ独自のインスタンスを作成するのではなく、Neptune ノートブックインスタンスを共有するように依頼します。1 つの Jupyter インスタンスで開発者ごとに個別のノートブックフォルダを作成します。インスタンスが使用されていない場合はシャットダウンします。

リソース

リファレンス

- [AWS Well-Architected](#)
- [AWS Well-Architected フレームワークのドキュメント](#)
- [Amazon Neptune 用の AWS Well-Architected フレームワークの適用](#)
- [Neptune Analytics のベストプラクティス](#)

ブログ記事と動画

- [Neptune ブログ投稿](#) (AWS データベースブログ)
- [リソースタグを使用して Amazon Neptune 環境リソースの停止と開始を自動化する](#) (AWS データベースブログ)
- [Amazon Neptune の録音可能ファイル](#) (短い YouTube 動画)

トレーニング

- [Amazon Neptune の開始方法](#)
- [Amazon Neptune で構築する](#)
- [Amazon Neptune のデータモデリング](#)
- [Amazon Neptune 分析ワークショップ](#)

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#) をサブスクライブできます。

変更	説明	日付
初版発行	—	2024 年 12 月 20 日

AWS 規範ガイド用語集

以下は、AWS 規範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の最適化レベルを導入します。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: カスタマーリレーションシップ管理 (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 インスタンス上の 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。サーバーを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移行します。例: Microsoft Hyper-Vアプリケーションをに移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを行き移るためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。

- 使用停止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

[属性ベースのアクセスコントロール](#)を参照してください。

抽象化されたサービス

「[マネージドサービス](#)」を参照してください。

ACID

[アトミック性、一貫性、分離性、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。より柔軟ですが、[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する SQL 関数。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。

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。

アベイラビリティーゾーン

他のアベイラビリティーゾーンの障害から 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ー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)

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で効果的な計画を立て 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整理します。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、AWS CAF は、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションの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 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

個人や組織を混乱させたり、損害を与えたりすることを意図した[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 の動作グラフを使用すると、失敗したログオンの試行、不審な API 呼び出し、その他同様のアクションを調べることができます。詳細については、Detective ドキュメントの[Data in a behavior graph](#)を参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。[エンディアン性](#)も参照してください。

二項分類

バイナリ結果 (2 つの可能なクラスの中の 1 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 つの異なる同一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を 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を別の環境 (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。

ボット

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

ボットネット

[マルウェア](#)に感染し、[ボット](#)ハーダーまたはボットオペレーターとして知られる、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。ボットは、ボ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なります。詳細については、Well-Architected [ガイド](#)の「[ブレイクグラス手順の実装](#)」インジケータ AWS を参照してください。

ブラウンフィールド戦略

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ン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。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

[AWS 「クラウド導入フレームワーク」](#) を参照してください。

Canary 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。確信が持てば、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。

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します。[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](#) 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
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、CCoE の定義、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

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

CMDB

[「設定管理データベース」](#) を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、GitHub または が含まれます 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イメージやビデオ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、Amazon SageMaker AI は CV 用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、設定は想定状態から変化します。ワークロードが非標準になる可能性があり、通常は段階的かつ意図的ではありません。

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

一元管理とガバナンスを備えた分散型の分散型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、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「[でのデータ境界の構築 AWS](#)」を参照してください。

データの事前処理

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。データの事前処理とは、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。

データの出所

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。

データの種類

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ており、通常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。

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

ティの手法。この戦略を採用するときは AWS、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て、リソースの安全性を確保します。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。

デプロイ

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

開発環境

[「環境」](#)を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

[スタースキーマ](#)では、ファクトテーブル内の量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

離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに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、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 (ボストン: 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

「[開発値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、応答時間を短縮できます。

電子データ交換 (EDI)

組織間の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) 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力カテゴリー。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

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

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で段階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。

障害分離の境界

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

同様のタスクの実行を求める前に、タスクと必要な出力を示す少数の例を [LLM](#) に提供します。この手法は、プロンプトに埋め込まれた例 (ショット) からモデルが学習するコンテキスト内学習のアプリケーションです。少数ショットプロンプトは、特定のフォーマット、推論、またはドメインの知識を必要とする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コントロール」](#) を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

段階的なアプローチを使用する代わりに、[変更データキャプチャ](#) 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。

FM

[「基盤モデル」](#) を参照してください。

基盤モデル (FM)

一般化およびラベル付けされていないデータの大規模なデータセットでトレーニングされている大規模な深層学習ニューラルネットワーク。FMs は、言語の理解、テキストと画像の生成、自然言語の会話など、さまざまな一般的なタスクを実行できます。詳細については、[「基盤モデルとは」](#) を参照してください。

G

生成 AI

大量のデータでトレーニングされ、シンプルなテキストプロンプトを使用してイメージ、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る [AI](#) モデルのサブセット。詳細については、[「生成 AI とは」](#) 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](#)はモダンで推奨されるアプローチです。

ゴールデンイメージ

そのシステムまたはソフトウェアの新しいインスタンスをデプロイするためのテンプレートとして使用されるシステムまたはソフトウェアのスナップショット。例えば、製造では、ゴールデンイメージを使用して複数のデバイスにソフトウェアをプロビジョニングし、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、Amazon GuardDuty AWS Security Hub、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。

ホールドアウトデータ

[機械学習](#)モデルのトレーニングに使用されるデータセットから保留される、ラベル付きの履歴データの一部。モデル予測をホールドアウトデータと比較することで、ホールドアウトデータを使用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

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

IIoT

[「産業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、本番環境の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、予測性が高くなります。詳細については、AWS 「Well-Architected フレームワーク」の「[イミュータブルインフラストラクチャを使用したデプロイ](#)」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。

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](#) によって導入された用語で、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ゼーションを指します。

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。

産業分野における IoT (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、「[Building an industrial Internet of Things \(IIoT\) digital transformation strategy](#)」を参照してください。

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる 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

「[モノのインターネット](#)」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「[IT 情報ライブラリ](#)」を参照してください。

ITSM

「[IT サービス管理](#)」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

ランディングゾーンは、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)

大量のデータに対して事前トレーニングされた深層学習 [AI](#) モデル。LLM は、質問への回答、ドキュメントの要約、テキストの他の言語への翻訳、文の完了など、複数のタスクを実行できます。詳細については、[LLMs](#) を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

[「ラベルベースのアクセスコントロール」](#) を参照してください。

最小特権

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

リフトアンドシフト

[「7 Rs」](#) を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。 [エンディアン性](#) も参照してください。

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の中断、機密情報の漏洩、不正アクセスにつなが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。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象化されたサービスとも呼ばれます。

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、制御するためのソフトウェアシステムで、原材料を工場の完成製品に変換します。

MAP

[「移行促進プログラム」](#)を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、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、動作中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

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。

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](#) デバイス用の、[パブリッシュ/サブスクライブ](#)パターンに基づく軽量 machine-to-machine (M2M) 通信プロトコル。

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)

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 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」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、「」の「[アプリケーションをモダナイズするための戦略 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 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

[「オペレーションの統合」](#)を参照してください。

OLA

[「運用レベルの契約」](#)を参照してください。

オンライン移行

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

OPC-UA

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

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

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

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)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小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)

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、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のため、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)

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コントロールが可能です。

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

アクセス許可を定義 ([アイデンティティベースのポリシー](#)を参照)、アクセス条件を指定 ([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を定義 AWS Organizations ([サービスコントロールポリシー](#)を参照) できるオブジェクト。

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falseは 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 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](#)内にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通じてプライバシーを考慮するシステムエンジニアリングアプローチ。

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を防ぐように設計された[セキュリティコントロール](#)。これらのコントロールは、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、AWS Control Tower ドキュメントの「[コントロールリファレンスガイド](#)」および「[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まで、製品のデータとプロセスのライフサイクル全体にわたる管理。

本番環境

[「環境」](#)を参照してください。

プログラム可能な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、信頼性の高い適応可能なコンピュータです。

プロンプトの連鎖

1 つの [LLM](#) プロンプトの出力を次のプロンプトの入力として使用して、より良いレスポンスを生成します。この手法は、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善または拡張したりするために使用されます。これにより、モデルのレスポンスの精度と関連性が向上し、より詳細でパーソナライズされた結果が得られます。

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。

パブリッシュ/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。スケーラビリティと応答性を向上させます。たとえば、マイクロサービスベースの [MES](#) では、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発行できます。システムは、公開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される手順などの一連のステップ。

クエリプランのリグレッション

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

R

RACI マトリックス

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

RAG

[「取得拡張生成」](#) 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

[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。

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)

サービスの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、他のとは独立しています。詳細については、[AWS リージョン「アカウントで使用するを指定する」](#)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、で回復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。

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンスコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、Implementing security controls on AWSの[Responsive controls](#)を参照してください。

保持

[「7 Rs」](#)を参照してください。

廃止

[「7 Rs」](#)を参照してください。

取得拡張生成 (RAG)

[LLM](#) がレスポンスを生成する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を参照する[生成 AI](#) テクノロジー。たとえば、RAG モデルは、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する場合があります。詳細については、[「RAG とは」](#)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 Management Console 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

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。

設計によるセキュリティ

開発プロセス全体でセキュリティを考慮するシステムエンジニアリングアプローチ。

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、[予防的](#)、[検出的](#)、[応答的](#)、[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。

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修復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。自動応答アクションの例としては、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

のエントリポイントの 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 [レベルのインジケータ](#) によって測定される、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、お客様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

[セキュリティ情報とイベント管理システム](#) を参照してください。

単一障害点 (SPOF)

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

SLA

[「サービスレベルの契約」](#) 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。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、『』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 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よう設計されています。

strangler fig パターン

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

サブネット

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

監視コントロールとデータ収集 (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。

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。

システムプロンプト

[LLM](#) にコンテキスト、指示、またはガイドラインを提供して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、ユーザーとのやり取りのルールを確立するのに役立ちます。

T

tags

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 枚のピザで養う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、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

WORM

「[Write Once](#)」、「[Read Many](#)」を参照してください。

WQF

[AWS「ワークロード認定フレームワーク」](#)を参照してください。

write once, read many (WORM)

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

Z

ゼロデイエクスプロイト

[ゼロデイ脆弱性](#) を利用する攻撃、通常はマルウェア。

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](#) にタスクを実行する手順を提供しますが、タスクのガイドに役立つ例 (ショット) はありません。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。