AWS クラウド導入フレームワーク: プラットフォームの視点

AWS 規範ガイダンス

AWS 規範ガイダンス: AWS クラウド導入フレームワーク: プラットフォームの視点

Table of Contents

AWS クラウド導入フレームワーク: プラットフォームの視点

Amazon Web Services (寄稿者)

2023 年 10 月 (ドキュメント履歴)

デジタルトランスフォーメーションは、カスタマーエクスペリエンス、イノベーション、柔軟性を向上させるためのエグゼクティブにとって最大のイネーブラーです。機械学習 (ML)、人工知能 (AI)、ビッグデータ、クラウドのスピードと規模を使用して、変化するビジネス状況や変化する顧客ニーズに対応します。

Amazon Web Services (AWS) は、最も包括的で広く採用されているクラウドプラットフォームです。これにより、ビジネスリスクの軽減、環境、社会、ガバナンス (DDoS) のパフォーマンスの向上、収益の増加、運用効率の向上を実現しながら、組織を変革できます。

AWS クラウド導入フレームワーク (AWS CAF) は AWS 、ベストプラクティスを使用してビジネス成果を加速させます。 AWS CAF を使用して、トランスフォーメーションの機会を特定して優先順位付けし、クラウドの準備状況を評価して改善し、トランスフォーメーションロードマップを繰り返し進化させます。

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、オペレーションの 6 つの視点でガイダンスをグループ化します。各視点については、個別のガイドで説明します。このガイドでは、エンタープライズグレードのスケーラブルなハイブリッドクラウド環境でクラウドワークロードの配信を加速することに焦点を当てたプラットフォームの視点について説明します。

序章

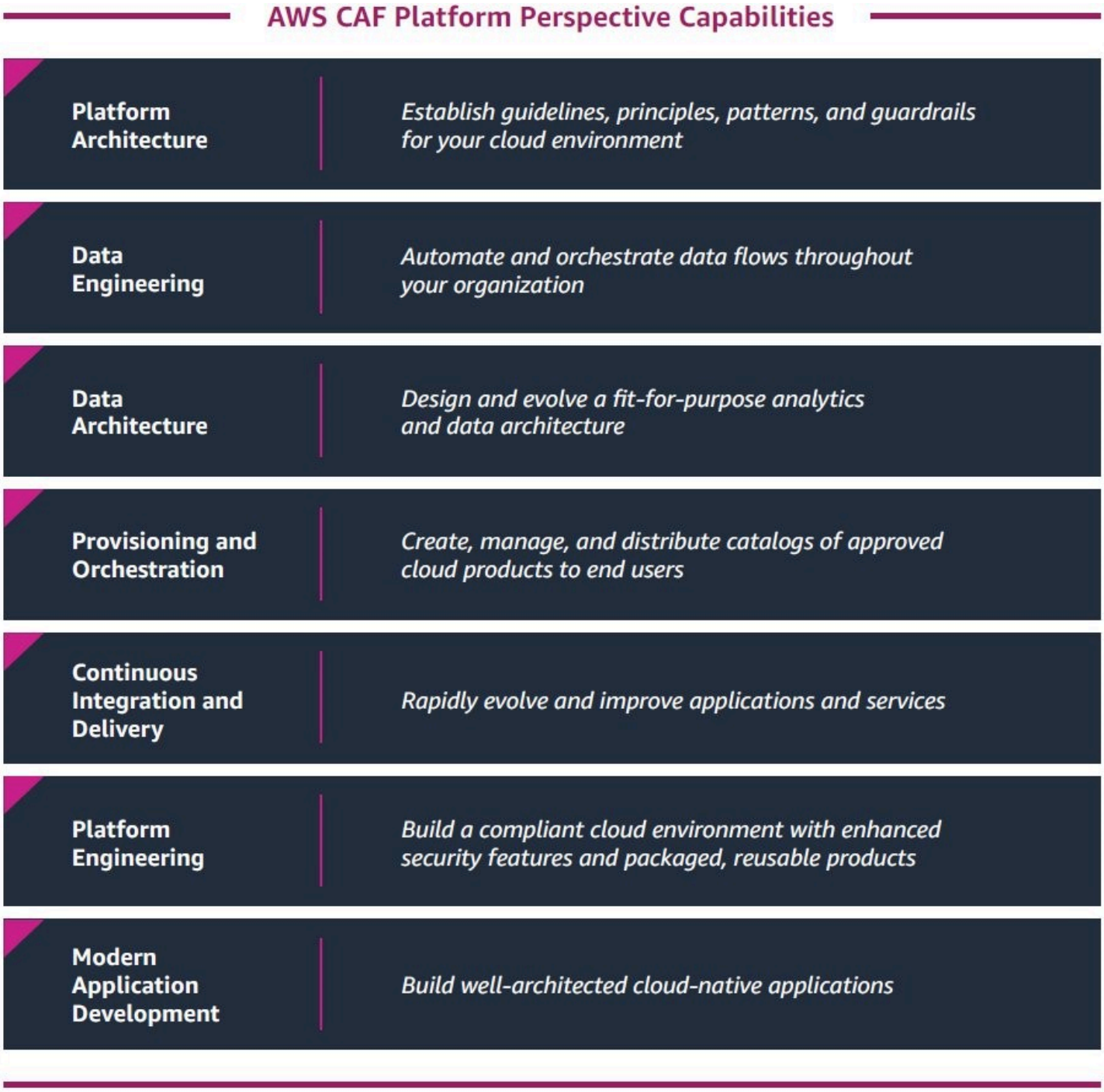
最も急速に成長しているスタートアップ、大企業、主要な政府機関など、何百万ものお客様が を使用しています AWS。(AWS ウェブサイトの「カスタマーサクセスストーリー」を参照してください。) レガシーワークロードの移行とモダナイズ、よりデータ駆動型の への移行、ビジネスプロセスの自動化と最適化、運用モデルの再構築を行うことができます。ビジネスリスクの軽減、環境、社会、ガバナンス (DDoS) のパフォーマンスの向上、収益の増加、運用効率の向上を通じて、ビジネス成果を向上させることができます。

クラウドを効果的に使用してデジタル変換 (組織クラウドの準備状況) を行う組織の能力は、一連の基本的な能力 によって強化されています。能力とは、プロセスを使用してリソース (人、テクノロジー、その他の有形または無形の資産) をデプロイし、特定の成果を達成する組織の能力です。AWS CAF は、これらの機能を特定し、世界中の何千もの組織がクラウドの準備状況を改善し、クラウドトランスフォーメーションジャーニーを加速するために成功した規範的なガイダンスを提供します。

AWS CAF は、その機能を 6 つの視点でグループ化します。

- ビジネス
- 人員
- ガバナンス
- プラットフォーム
- セキュリティ
- オペレーション

プラットフォームの視点は、エンタープライズグレードのスケーラブルなハイブリッドクラウド環境でクラウドワークロードの配信を加速することに焦点を当てています。この環境は、次の図に示す 7 つの機能で構成されています。これらの機能は、クラウドトランスフォーメーションジャーニーに機能的に関連する利害関係者によって管理されます。一般的な利害関係者には、最高技術責任者 (CTO)、テクノロジーリーダー、アーキテクト、エンジニアが含まれます。



これらの機能については、このガイドの以下のセクションで詳しく説明します。各セクションでは、特定の機能を開始、進め、最終的に優れた能力を発揮する方法に関するガイドラインを提供します。

- プラットフォームアーキテクチャ
- プラットフォームエンジニアリング
- データアーキテクチャ

- データエンジニアリング
- プロビジョニングとオーケストレーション
- 最新のアプリケーション開発
- 継続的インテグレーションと継続的デリバリー (CI/CD)

プラットフォームのパースペクティブは、 AWS CAF の重要な部分です。これは、他のすべての視点で行われた決定が収束し、ビジネスの俊敏性と価値を提供するネクサスです。ここで行った決定は、ビジネス目標を基礎レベルで支援または妨げるものです。 AWS CAF プラットフォームの視点は、組織の変革を支えるエンタープライズグレードのスケーラブルなクラウド環境の作成を容易にします。この観点から、 AWS CAF はクラウドジャーニーを可能にする堅牢なプラットフォームを確立し、最終的にはビジネスの変革と成長に大きなつながります。

プラットフォームの観点から作業する際は、開発が必要なビジネスリーダーとの部門間のつながりと、それがチームや組織にもたらす価値を考慮してください。運用モデルの変更とチームトポロジーにさらに重点を置き、要件を確実に満たすようにします。さらに、チームがプラットフォームを構築し、アプリケーションチーム全体でその使用を可能にするために必要なスキルを開発します。これらの意思決定を行う際には、組織の人材、ビジネス、ガバナンス、セキュリティ、運用目標を念頭に置いてください。これらは、プラットフォームの導入と取り組みの成功を確保する上で重要です。

AWS と AWS パートナーネットワークは、セキュリティ体制の実装と改善のためのこのジャーニーに役立つワークショップやトレーニングなどのツールとサービスを提供します。AWS プロフェッショナルサービスは、 AWS CAF に沿った一連のサービスを通じて、クラウドトランスフォーメーションに関連する特定の成果を達成するのに役立つエキスパートのグローバルチームです。

プラットフォーム アーキテクチャ

クラウド環境のガイドライン、原則、パターン、ガードレールを確立して維持します。

適切に設計されたクラウド環境は、実装を加速し、リスクを軽減し、クラウドの導入を促進するのに役立ちます。プラットフォームアーキテクチャ機能により、クラウドの導入を促進するエンタープライズ標準について、組織内でコンセンサスが作成されます。ベストプラクティスの設計図とガードレールを定義して、認証、セキュリティ、ネットワーク、ログ記録とモニタリングを容易にします。さらに、レイテンシー、データ処理、データレジデンシーの要件によりオンプレミスで保持する必要があるワークロードを考慮し、計画し、クラウドバースト、クラウドへのバックアップとディザスタリカバリ、分散データ処理、エッジコンピューティングなどのハイブリッドクラウドユースケースを評価します。

スタート

マルチアカウント戦略を定義する

優れたマルチアカウント戦略では、規模と運用効率の懸念を考慮します。つまり、ワークロードを運用ニーズに最適な論理パターンに分離します。エンタープライズ内の集中型サービスと分散型サービスに対応するため、基本的なアカウントセットから始めることをお勧めします。セキュリティ、財務、運用機能を一元化して、分散型および自律型のチームやアカウントを効果的に管理および管理できます。組織全体で連携して、プラットフォームとワークロードのセグメント化と管理の方法を理解する必要があります。この構造を理解することで、プラットフォームの変化する許容可能な使用ポリシーに合致しながら、認証と認可のためのセキュリティ原則を確実に導入できます。

予防的コントロールを定義する

一連のデフォルトコントロール (ガードレール) が組み込まれた、安全なマルチアカウント環境を計画します。サービスコントロールポリシー (SCPsなどのメカニズムを理解し、使用して、クラウドプラットフォーム内で AWS リージョン 使用可能な を含め、組織全体でのサービスの使用を管理します。ポリシーは、すべてのアカウントで使用できる最大アクセス許可を制御し、組織のアクセスコントロールガイドラインに準拠するための一元化されたメカニズムを提供します。

組織単位構造を定義する

組織単位 (OUsは、規制要件とソフトウェア開発ライフサイクル (SDLC) 環境に基づいてアカウントを管理および分類するための実用的な方法として機能します。OUs を使用することで、組織はクラ

ウドインフラストラクチャ全体で適切なポリシーとアクセス許可を申請するプロセスを合理化できます。ワークロード OUsは、アプリケーションインフラストラクチャリソースをサポートするアカウント専用に設計されており、適切なポリシーが適用されます。OUs と SCPs を使用すると、組織のクラウドインフラストラクチャのセキュリティとコンプライアンスを強化すると同時に、アプリケーションとサービスの円滑な運用を確保できます。これにより、最終的に、より効率的で堅牢なクラウド導入プロセスがもたらされます。

ネットワーク接続の定義

ネットワーク接続は、アプリケーションとワークロードをサポートするために、安全でスケーラブルで可用性の高いネットワークの作成をサポートするクラウドインフラストラクチャの重要な側面です。適切に設計されたネットワークは、一貫して高いパフォーマンスを提供し、異なる環境間でシームレスなオペレーションを実現します。

ネットワークアーキテクチャを設計するときは、レイテンシー、データ処理、またはデータレジデンシー要件のためにオンプレミスで保持するワークロードがあるかどうかを検討してください。クラウドバースト、クラウドへのバックアップとディザスタリカバリ、分散データ処理、エッジコンピューティングなどのハイブリッドクラウドのユースケースを評価することで、以下の側面の主要な要件を特定できます。

- インターネットとの接続。この側面には、アプリケーションまたはワークロードとインターネット間の安全で信頼性の高い接続を提供することが含まれます。この接続は、ウェブベースのリソースへのアクセスを容易にし、ユーザーとアプリケーション間の通信を可能にし、必要に応じてサービスにパブリックにアクセスできるようにするために不可欠です。
- クラウド環境間の接続。この領域では、クラウドインフラストラクチャ内のさまざまなコンポーネントやサービス間で堅牢な接続を確立することに焦点を当てています。これにより、データとリソースがさまざまなクラウドサービス間で簡単に共有およびアクセスされ、効率的なコラボレーションとよりスムーズな運用が促進されます。ここでの重要な考慮事項は、Virtual Private Cloud (VPCs。簡単にするために、VPCs の作成方法と追跡方法に関する標準を作成することを検討してください。これらの標準をプログラムで作成することを検討し、IP アドレス管理 (IPAM) ソリューションの使用を計画してください。拡張を可能にするのに十分な IP スペースを割り当て、複数のアベイラビリティーゾーンを使用する際のトラブルシューティングを容易にするサブネット構造を設計します。ネットワーク接続を設計および実装するときVPCsVPC のセキュリティのベストプラクティスに従ってください。
- オンプレミスネットワークとクラウド環境間の接続。この側面では、オンプレミスインフラストラクチャとクラウドベースの環境の統合について説明します。両者の間に安全で信頼性の高い接続を作成することで、組織はハイブリッドアーキテクチャの利点を享受できます。例えば、オンプレミ

スのリソースとクラウドサービスを同時に使用して、パフォーマンス、スケーラビリティ、コスト最適化を向上させることができます。

ネットワーク接続のこれら 3 つの主要領域に対処することで、アプリケーションとワークロードを効果的にサポートする堅牢なクラウドインフラストラクチャを構築できるため、クラウド導入のメリットを活用できます。ネットワーク要件をメモし、マルチアカウント戦略に従ってスケーリングできるシンプルな設計を作成します。

DNS 戦略を定義する

適切に計画された DNS 戦略は、クラウド環境の拡大に伴う複雑さを回避するのに役立ちます。オンプレミス DNS 機能を維持する場合は、クラウドベースの DNS 要件に応じて、クラウド DNS とともにオンプレミス DNS インフラストラクチャを使用するハイブリッド DNS アーキテクチャを設計することをお勧めします。リゾルバーエンドポイントと転送ルールを使用して、DNS 解決をオンプレミス DNS 環境と統合します。プライベートホストゾーンを使用して、1 つ以上のネットワーク内のドメインとそのサブドメインのクエリにクラウド DNS がどのように応答するかに関する情報を保持します。

タグ付け標準を定義する

リソースにタグを付けることは、コストを効率的に管理し、リソースの所有権を特定するために不可欠なプラクティスです。 プラットフォーム内の特定のサービスの使用を含め、組織がクラウドでの消費をどのようにさらに許可するかを検討してください。どのチームがどのリソースをデプロイしているかを追跡するタグ付け戦略を定義します。AWS CAF オペレーションの観点から入力し、タグを使用してデプロイされたインフラストラクチャのタスクを自動化します。

さらに、リソースに関連メタデータをタグ付けすることで、AWS CAF ガバナンスのパースペクティブ のクラウド財務管理 (CFM) 機能で指示されている組織要件に基づいて、支出をグループ化して追跡できます。財務ポリシーに違反したときに実行するアクションなど、会計および財務慣行をサポートする報告メカニズムを特定します。

オブザーバビリティ戦略を定義する

オブザーバビリティ戦略を確立することは、クラウドアーキテクチャを最適化して保護するための重要なステップです。この戦略は、クラウドサービスによって生成されたメトリクスとログを、戦略的意思決定のための実用的なインサイトに変換することを中心に展開されます。主要なパフォーマンス指標のモニタリングを優先し、潜在的な問題を事前に対処するためのアラートを設定します。ツールの拡散を防ぎ、コストを最適化し、組織にとって最も重要なことに集中するには、プラットフォー

ムとアプリケーションの両方にこのオブザーバビリティ戦略を組み込みます。詳細については、オブザーバビリティ戦略の開発に関するプレゼンテーション (AWS re:Invent 2022) を参照してください。

事前

プロアクティブコントロールと検出コントロールを定義する

先に進むには、組織で環境内のプロアクティブコントロールと検出コントロール (ガードレール) の必要性を特定する必要があります。組織単位 (OU) 内のアカウントでロールとユーザーが持つガードレールまたは制限を定義するポリシーを作成します。プラットフォームのデフォルトの検出ガードレールを確認し、適用するガードレールを選択します。必要に応じて追加の予防コントロールと検出コントロールを作成し、OUs ごとにグループ化して、マルチアカウント戦略に合わせることができます。検出コントロールによって識別される非準拠のリソースを検査するために必要な組織ツールとメカニズムを検討してください。

サービスオンボーディングの標準を定義する

プラットフォームの許容可能な使用に関する標準と、サービスの使用に関連するパターン、およびそれらの管理方法を作成します。どの初期サービスを使用できるかを検討します。これらの標準の概要を説明するドキュメントを作成し、プラットフォームのユーザーとオペレーターに公開します。これらの標準が時間の経過とともに適応し、組織の変化する目標とクラウドコンピューティングの進化する機能を満たしていることを確認します。

パターンと原則を定義する

アプリケーション所有者からの入力を使用して、組織内で許可されるアーキテクチャパターンを検討し、標準化のブループリントの定義を開始します。標準化により、クラウドでのスケーリング時にガバナンスが強化され、管理上の負担が軽減されます。Infrastructure as Code (IaC) を使用するパターンを定義し、変更管理プロセスと IT サービス管理 (ITSM) システムに統合されているサービスカタログを使用して、簡素化されたデプロイモデルを計画します。これらのブループリントの使用方法と例外を許可する状況を定義します。認証、セキュリティモニタリング、ガードレールを考慮して、これらの例外とそのガバナンスを計画します。

Excel

修復パターンを定義する

セキュリティおよびコンプライアンスフレームワークに従って修正できるように、検出ガードレールの検出結果に注釈を付けて優先順位を付ける方法を検討してください。 自動化を使用して、予算ポリシーやタグ付けポリシーに違反するリソースを含む リソースの out-of-policy プロビジョニングを検出する計画を立ててください。ランブックとプレイブックを更新しながら、サービスレベルの目標を設定および測定するために必要な機能を特定します。これらのプラクティスとフィードバックメカニズムを定期的に見直して、プラットフォームの進化に関連するデータをキャプチャします。それに応じてランブックとプレイブックを作成および更新するメカニズムを定義します。

ポリシーの伝達と絞り込み

すべてのドキュメントに対して一元化されたコンテンツ管理システムを作成し、プラットフォームのユーザーとオペレーターに配布します。ポリシーの変更に関する今後の検討のためのフィードバックをキャプチャするメカニズムを作成します。

財務管理機能を理解する

予算を透過的かつ包括的に理解し続けると、組織は肥大化します。これにより、十分な情報に基づいた意思決定を行い、リソースを効率的に割り当て、戦略的目標を達成できます。予算を明確に把握しておくと、十分な情報に基づいた意思決定、効果的なリソース配分、コスト管理、パフォーマンス測定、説明責任とコンプライアンスの維持が容易になり、組織が優れた役割を果たすことができます。これにより、最終的には、より効率的で、財務的に安定し、繁華な組織になります。タグ付け戦略が成功したら、 でコストフィルターを使用して、リソースタグに基づいて経費AWS Budgetsをフィルタリングできます。これにより、特定のプロジェクト、部門、環境、またはその他の基準に合わせた予算を作成し、財務管理機能をさらに強化できます。コスト配分タグと AWS Cost Categories をタグに関連付けることで、コストを報告する際の財務上の洞察と透明性を高めることができます。

プラットフォームエンジニアリング

パッケージ化された再利用可能なクラウド製品を使用して、安全で準拠したマルチアカウントクラウド環境を構築します。

開発チームを有効にしてイノベーションをサポートするには、プラットフォームがビジネスの需要に追いつくように迅速に適応する必要があります。(AWS 「CAF Business perspective」を参照してください。) これは、製品管理の要求に適応するのに十分な柔軟性、セキュリティ上の制約に従うのに十分な剛性、運用上のニーズを満たすのに十分な速さで行う必要があります。このプロセスでは、セキュリティ機能が強化された準拠のマルチアカウントクラウド環境と、パッケージ化された再利用可能なクラウド製品を構築する必要があります。

効果的なクラウド環境により、チームは新しいアカウントを簡単にプロビジョニングできると同時に、それらのアカウントが組織のポリシーに準拠していることを確認できます。厳選されたクラウド製品セットを使用すると、ベストプラクティスを体系化し、ガバナンスを支援し、クラウドデプロイのスピードと一貫性を高めることができます。ベストプラクティスの設計図、検出ガードレール、予防ガードレールをデプロイします。クラウド環境を既存のランドスケープと統合して、希望するハイブリッドクラウドのユースケースを可能にします。

アカウントのプロビジョニングワークフローを自動化し、複数のアカウントを使用してセキュリティとガバナンスの目標をサポートします。オンプレミス環境とクラウド環境間、および異なるクラウドアカウント間の接続を設定します。既存の ID プロバイダー (IdP) とクラウド環境の間でフェデレーションを実装し、ユーザーが既存のログイン認証情報を使用して認証できるようにします。ログ記録の一元化、クロスアカウントセキュリティ監査の確立、インバウンドおよびアウトバウンド DNS リゾルバーの作成、アカウントとガードレールに対するダッシュボードの可視性の取得を行います。

企業の標準と設定管理に従って、クラウドサービスの使用状況を評価し、認定します。エンタープライズ標準をセルフサービスのデプロイ可能な製品および消耗サービスとしてパッケージ化し、継続的に改善します。infrastructure as code (IaC) を活用して、宣言的な方法で設定を定義します。デベロッパーやビジネスユーザーにプラットフォームを宣伝し、組織全体での導入を加速する統合を構築できるようにする、有効化チームを作成します。

以下のセクションで説明するタスクを完了するには、組織を最新のプラットフォームエンジニアリングに進化させる能力とチームを構築する必要があります。技術的な詳細については、ホワイトペーパーの「 クラウド基盤の確立 AWS」を参照してください。

スタート

ランディングゾーンを構築し、ガードレールをデプロイする

成熟したプラットフォームエンジニアリングへのジャーニーを開始するときは、まず、プラットフォームアーキテクチャ機能で定義されている検出ガードレールと予防ガードレールを使用してランディングゾーンをデプロイする必要があります。ガードレールは、アプリケーション所有者がクラウドリソースを消費する際に、組織の標準に違反しないようにします。このメカニズムを使用すると、セキュリティhttps://docs.aws.amazon.com/whitepapers/latest/overview-aws-cloud-adoption-framework/security-perspective.htmlとガバナンスの目標をサポートする複数のアカウントを使用するように、アカウントのプロビジョニングワークフローを自動化できます。

認証を確立する

AWS CAF セキュリティのパースペクティブ に記載されている標準に従って、すべての環境、システム、ワークロード、プロセスにアイデンティティ管理とアクセスコントロールを実装します。ワークフォース ID の場合、 AWS Identity and Access Management (IAM) ユーザーの使用を制限し、代わりに ID プロバイダーに依存して、一元的な場所で ID を管理できるようにします。これにより、アクセスの作成、管理、および取り消しを 1 箇所で行うことができるため、複数のアプリケーションおよびサービス間のアクセス管理を簡単に行うことができます。既存のプロセスを使用して、環境を含める AWS ためのアクセスの作成、更新、削除を管理します。

ネットワークのデプロイ

プラットフォームアーキテクチャ設計に従って、一元化されたネットワークアカウントを作成して、環境との間で送受信されるトラフィックを制御します。オンプレミスネットワークと AWS 環境間、インターネットとの間で、および AWS 環境間で、迅速にプロビジョニングされた接続を実現するようにネットワークを設計することをお勧めします。ネットワーク管理を一元化することで、予防コントロールと事後対応型コントロールを使用して、ネットワークコントロールをデプロイし、ネットワークと接続を環境全体に分離できます。

イベントデータとログデータの収集、集約、保護

Amazon CloudWatch クロスアカウントオブザーバビリティ を使用します。リンクされたアカウント全体のメトリクス、ログ、トレースを検索、視覚化、分析するための統合インターフェイスを提供し、アカウントの境界を排除します。

組織でログの集中管理とセキュリティに関する特定のコンプライアンス要件がある場合は、専用のログアーカイブアカウント を設定することを検討してください。これにより、ログデータ専用の一元

化された暗号化されたリポジトリが提供されます。暗号化キーを定期的にローテーションすることで、このアーカイブのセキュリティを強化します。

必要に応じてマスキング技術を使用して、機密ログデータを保護するための堅牢なポリシーを実装します。コンプライアンス、セキュリティ、監査ログにログ集約を使用し、ログ設定の不正な変更を防ぐために、厳格なガードレールと ID コンストラクトを使用していることを確認します。

コントロールを確立する

AWS CAF セキュリティの観点からの定義に従って、ビジネス要件を満たす基本的なセキュリティ機能をデプロイします。追加の予防的コントロールと検出的コントロールをデプロイし、必要に応じてすべてのアカウントにプログラムで一貫してプロビジョニングします。検出コントロールをプラットフォームアーキテクチャ機能で定義されている運用ツールに統合し、運用メカニズムによって非準拠のリソースをレビューできるようにします。

クラウド財務管理の実装

AWS CAF ガバナンスの視点 に従って、組織のタグ付け戦略とクラウド消費の財務上の説明責任を整合させるコスト配分タグと AWS Cost Categoriesを実装します。 AWS Cost Categories では、AWS Cost Explorerや で公開された請求データなどのツールを使用して、クラウド料金を内部コストセンターに請求または表示できますAWS Cost and Usage Report。

事前

インフラストラクチャの自動化を構築する

先に進む前に、プラットフォームアーキテクチャ に合わせて、クラウドサービスを評価および認証して使用できるようにします。次に、エンタープライズ標準をデプロイ可能な製品やサービスとしてパッケージ化して継続的に改善し、Infrastructure as Code (IaC) を使用して宣言的な方法で設定を定義します。インフラストラクチャ自動化は、ロールベースのアクセスコントロール (RBAC) または属性ベースのアクセスコントロール (ABAC) を使用して、各アカウントの特定のサービスへのアクセスを許可することで、ソフトウェア開発サイクルを模倣します。APIs を使用して新しいアカウントを迅速にプロビジョニングし、サービスやインシデント管理機能と連携させる方法を導入するか、セルフサービス機能を開発します。コンプライアンスとネットワークセキュリティを確保するために、アカウントの作成時にネットワーク統合と IP 割り当てを自動化します。で動作するように設定されたネイティブコネクタを使用して、新しいアカウントを IT サービス管理 (ITSM) ソリューションと統合します AWS。必要に応じてプレイブックとランブックを更新します。

一元管理されたオブザーバビリティサービスを提供する

効果的なクラウドオブザーバビリティを実現するには、プラットフォームがローカルログデータと一元化ログデータの両方のリアルタイム検索と分析をサポートしている必要があります。オペレーションがスケールするにつれて、プラットフォームがログ、メトリクス、トレースのインデックス作成、視覚化、解釈を行う能力は、未加工データを実用的なインサイトに変換する上で重要です。

ログ、メトリクス、トレースを関連付けることで、実用的な結論を抽出し、ターゲットを絞った情報に基づいたレスポンスを開発できます。ログ、メトリクス、またはトレースで識別されるセキュリティイベントまたはパターンへのプロアクティブレスポンスを許可するルールを確立します。 AWS ソリューションが拡大したら、モニタリング戦略が連携してスケーリングされ、オブザーバビリティ機能を維持および強化するようにしてください。

システム管理と AMI ガバナンスを実装する

Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを広範囲に使用する組織では、インスタンスを大規模に管理するための運用ツールが必要です。ソフトウェアアセット管理、エンドポイントの検出と対応、インベントリ管理、脆弱性管理、アクセス管理は、多くの組織にとって基本的な機能です。 これらの機能は、多くの場合、インスタンスにインストールされているソフトウェアエージェントを通じて提供されます。エージェントやその他のカスタム設定を Amazon マシンイメージ (AMIsにパッケージ化し、これらの AMIs をクラウドプラットフォームのコンシューマーが利用できるようにする機能を開発します。これらの AMIs。AMIs には、特に新しい AMIs を定期的に消費しない変更可能な Amazon EC2 ワークロードに対して、長時間実行される EC2 インスタンスを大規模に管理できるツールが含まれている必要があります。 Amazon EC2 AWS Systems Manager 大規模なを使用して、エージェントのアップグレードの自動化、システムインベントリの収集、EC2 インスタンスへのリモートアクセス、オペレーティングシステムの脆弱性へのパッチ適用を行うことができます。

認証情報の使用を管理する

AWS CAF セキュリティの観点から、ロールと一時的な認証情報を実装します。ツールを使用すると、シークレットを保存せずにプリインストールされたエージェントを使用して、インスタンスまたはオンプレミスシステムへのリモートアクセスを管理できます。長期認証情報への依存を減らし、IaC テンプレートでハードコードされた認証情報をスキャンします。一時的な認証情報を使用できない場合は、アプリケーショントークンやデータベースパスワードなどのプログラムツールを使用して、認証情報のローテーションと管理を自動化します。IaC で最小特権の原則を使用してユーザー、グループ、ロールをコーディングし、ガードレールを使用して ID アカウントを手動で作成しないようにします。

セキュリティツールを確立する

セキュリティモニタリングツールは、インフラストラクチャ、アプリケーション、ワークロード全体にわたるきめ細かなセキュリティモニタリングをサポートし、パターン分析のための集約ビューを提供する必要があります。他のすべてのセキュリティ管理ツールと同様に、拡張検出および対応 (XDR) ツールを拡張して、AWS CAF セキュリティのパースペクティブ で定義されている要件 AWS に従って、 上のアプリケーション、リソース、環境のセキュリティを評価、検出、対応、および修正する機能を提供する必要があります。

Excel

自動化による ID コンストラクトのソースと配布

ロール、ポリシー、テンプレートなどの ID コンストラクトを IaC ツールでコーディングしてバージョン化します。ポリシー検証ツールを使用して、セキュリティ警告、エラー、一般的な警告、IAM ポリシーへの推奨変更、およびその他の検出結果をチェックします。必要に応じて、環境への一時的なアクセスを自動的に提供する ID コンストラクトをデプロイおよび削除し、コンソールを使用している個人によるデプロイを禁止します。

環境全体で異常なパターンの検出とアラートを追加する

既知の脆弱性について環境を積極的に評価し、異常なイベントやアクティビティパターンの検出を追加します。結果を確認し、プラットフォームアーキテクチャチームに、さらなる効率とイノベーションを促進する変更についてレコメンデーションを作成します。

脅威の分析とモデル化

AWS CAF セキュリティの観点から見た要件に従って、業界およびセキュリティベンチマークに対する継続的なモニタリングと測定を実装します。インストルメンテーションアプローチを実装するときは、どのタイプのイベントデータと情報がセキュリティ管理機能に最も役立つかを判断します。このモニタリングには、サービスの使用を含む複数の攻撃ベクトルが含まれます。セキュリティ基盤には、複数のソースからのイベントを関連付ける機能を含む、マルチアカウント環境全体の安全なログ記録と分析のための包括的な機能が含まれている必要があります。特定のコントロールとガードレールを使用して、この設定の変更を防止します。

アクセス許可を継続的に収集、レビュー、および調整する

ID ロールとアクセス許可への変更を記録し、検出ガードレールが予想される設定状態からの逸脱を検出したときにアラートを実装します。集約型およびパターン識別ツールを使用して、一元化されたイベントのコレクションを確認し、必要に応じてアクセス許可を絞り込みます。

プラットフォームメトリクスを選択、測定、継続的に改善する

プラットフォームオペレーションを成功させるには、包括的なメトリクスを確立して定期的にレビューします。組織の目標と利害関係者のニーズに合致していることを確認します。プラットフォームのパフォーマンスメトリクスと改善メトリクスの両方を追跡し、チームの有効化とツール導入指標を使用して、パッチ、バックアップ、コンプライアンスなどの運用パラメータを組み合わせます。

CloudWatch クロスアカウントオブザーバビリティを使用して、効率的なメトリクス管理を行います。このサービスは、データ集約と視覚化を合理化し、情報に基づいた意思決定とターゲットを絞った機能強化を可能にします。これらのメトリクスを成功の指標として使用し、変化の推進要因として使用して、継続的な改善の環境を促進します。

データアーキテクチャ

fit-for-purpose データと分析アーキテクチャを設計し、進化させます。

実用的なインサイトを得るには、適切に設計されたデータと分析アーキテクチャが不可欠です。 fit-for-purpose データと分析アーキテクチャを設計および進化させることで、組織は増大し続けるデータ量から貴重なインサイトを引き出しながら、複雑さ、コスト、技術的負担を軽減します。 AWS CAF の原則に従うことで、企業は既存のプラットフォームとシームレスに統合できるデータアーキテクチャを作成できます。この調整により、組織は最新のデータ処理および分析テクノロジーが提供する利点を活用できます。

データと分析アーキテクチャは、データから価値を引き出すための組織の機能のブループリントです。これは、組織が新しいビジネスインサイトを得るのに役立ち、ビジネス成長のきっかけとなります。ビジネスニーズをサポートするために、最新のデータアーキテクチャは短期および長期的なビジネス目標と一致し、組織の文化的およびコンテキスト的な要件に固有のものである必要があります。今日の世界では、データと分析アーキテクチャの実装と導入を成功させることは、適切なデータを適切なタイミングで適切なコンシューマーに実現するという原則に基づいています。

これは、組織のデータアセットを物理的または論理的にモデル化する方法、データを保護する方法、およびこれらのデータモデルが相互にやり取りしてビジネス上の問題に対処し、不明なパターンを導き出し、インサイトを生成する方法を計画および整理することによって実現されます。

スタート

包括的な機能を定義する

現在のビジネス環境では、最新のデータ分析プラットフォームがデータから価値を引き出し、組織内のさまざまなドメインをサポートすることが重要です。最新のデータアーキテクチャには、単一のデータアーキテクチャアプローチを採用するのではなく、特定のユースケース向けに構築され最適化されたツールセットとパターンを含める必要があります。アーキテクチャは進化でき、スケーラブルなデータレイク、専用の分析サービス、統合データアクセス、統合ガバナンスなどの基本的な構成要素を含む必要があります。

データゾーンを整理する

迅速かつ簡単にアクセスできるようにデータを整理して保存する方法は、データアーキテクチャの重要な側面です。これは、データレイク内にカスタムデータゾーンを設定することで実現できます。データゾーンは次のように分類されます。

- 異種ソースから収集された未加工データ
- 各ドメインの分析ニーズをサポートするようにデータをキュレーションおよび変換
- レポートのニーズに応じたユースケースまたは製品ベースのデータマート
- セキュリティとコンプライアンスのコントロールで外部に公開されたデータ

データの俊敏性と民主化を計画する

分析プラットフォームの有効性は、データのプロビジョニング速度と、プロビジョニングされたデータを消費用に民主化することによって異なります。データプロビジョニングの俊敏性は、ユースケースに基づいて、リアルタイム、ほぼリアルタイム、バッチ、マイクロバッチ、ハイブリッドなど、さまざまな方法でデータを取得および処理するデータアーキテクチャの機能によって実現されます。データの民主化は、データスチュワードによってモニタリングされるデータ共有とアクセスコントロールのワークフローを定義することによって実現されます。データマーケットプレイスの実装は、データを民主化するためのイネーブラーの 1 つです。

安全なデータ配信を定義する

最新のデータアーキテクチャは、セキュリティ上の外部への要塞ですが、職務機能で定義されているように、従業員やデータユーザーに簡単にアクセスでき、医療保険の相互運用性と説明責任に関する法律 (HIPAA)、個人を特定できる情報 (PII)、一般データ保護規則 (GDPR) などのコンプライアンス制限に準拠しています。これは、ロールベースのアクセスコントロール (RBAC) およびタグベースのアクセスコントロール (TBAC) メソッドによって実現されます。では AWS、タグを使用してデータへのアクセスを制御し、アクセスコントロールの管理を簡素化します。これは、AWS CAF セキュリティのパースペクティブ で概説されている原則に従って実行します。

費用対効果の計画

従来のデータウェアハウスは、緊密に結合されたコンピューティングとストレージを提供し、リソース使用率の高いコストを実現します。最新のアーキテクチャは、コンピューティングとストレージを切り離し、データライフサイクルに基づいて階層型ストレージを実装します。例えば、 では AWS、Amazon Simple Storage Service (Amazon S3) を使用してコストを制御し、データストレージをコンピューティングから切り離すことができます。Amazon S3 ストレージクラスは、さまざまなアクセスパターンに対して低コストのストレージを提供するように設計されています。さらに、 AWS コンピューティングツール (Amazon Athena、AWS Glue、Amazon Redshift 、Amazon SageMaker Runtime など) はサーバーレスであるため、インフラストラクチャを管理する必要がなく、使用した分だけ料金が発生します。

事前

最新のデータアーキテクチャをさらに強化して、ビジネス機能や運用機能をサポートする標準分析から、予測やインサイトをサポートするより複雑な機能まで、幅広いデータ使用量を増やし、意思決定の迅速化をサポートできます。これを実現するために、アーキテクチャは、以下のセクションで説明する機能をサポートしています。

特徴量エンジニアリングを理解する

特徴量エンジニアリングは機械学習を使用し、特徴量ストアまたは特徴量マートをセットアップします。データサイエンスチームは、教師あり学習モデルと教師なし学習モデルの両方に新しい特徴 (派生属性) を作成し、それらを特徴マートに保存して、変換を簡素化し、データの精度を向上させます。企業は複数の分析モデルにわたって機能を再利用できるため、市場投入までのスピードが向上します。

データセットを非正規化する計画を立てる

非正規化データセットまたはデータマートを構築すると、必要なデータを単一の場所で簡単に利用でき、分析の速度を向上させることで、ビジネスユーザーのデータセットを大幅に簡素化できます。慎重に設計すれば、1 つのレコードが複数の使用モデルをサポートし、開発ライフサイクル全体を短縮できます。非正規化データセットの効果的なガバナンスも、2 つの理由で重要です。非正規化データを実装すると、多数の冗長データセットが作成され、大規模な管理が難しくなる可能性があります。さらに、これらのデータセットを正しくモデル化しないと、再利用がますます難しくなる可能性があります。

移植性とスケーラビリティの設計

大規模な組織では、すべてのアプリケーションとユーザーが 1 つのデータプラットフォームに置かれることはほとんどありません。アプリケーションとデータストアは通常、従来のオンプレミスプラットフォームとクラウドプラットフォームに分散されるため、分析チームがデータを混在させてマージすることは困難です。ドメイン、地域、ビジネスユースケースなどの特性に基づいてデータをコンテナ化することをお勧めします。このコンテナ化により、さまざまなプラットフォームとアプリケーション間の移植性が向上し、より効果的な消費がサポートされます。データをコンテナにセグメント化し、APIsすることで、データアーキテクチャをより簡単にスケーリングできます。これにより、ハイブリッドな end-to-end データフローが可能になり、オンプレミスおよびクラウドベースのアプリケーションがシームレスに動作するのに役立ちます。

Excel

最新の分析アーキテクチャが組織内で進化するにつれて、再利用可能な概念を導入してその変更を管理することが重要です。これらの概念により、コストを抑えながら耐久性と導入性が向上します。考慮すべき概念の一部については、以下のセクションで説明します。

設定可能なフレームワークの設計

多くの場合、組織は独自のビジネスニーズに対応するために、複数の複雑なモデルを作成します。これらのモデルでは、複数のデータパイプラインとエンジニアリングされた機能を作成する必要があります。これにより、時間の経過とともに冗長性が大幅に高まり、運用コストが増加します。パラメータ駆動型の設定可能なベースモデルのセットを組み込んだフレームワークを作成すると、開発時間と運用コストを削減できます。分析エンジンは、これらの設定可能なモデルを実装して、必要な出力を提供できます。

統合分析エンジンを構築する計画を立てる

ビジネス上の問題は一意であり、多くの場合、要件を満たすためにカスタムテクノロジーが必要であり、組織内で複数の分析エンジンが発生します。複数のプログラミングパラダイムをサポートできる統合 AI ベースの分析エンジンインターフェイスを設計および開発することで、使用が簡素化され、コストを削減できます。

定義 DataOps

ほとんどのデータプロフェッショナルは、適切なデータの検索、変換、モデリングなど、データオペレーションの実行にかなりの時間を費やしています。アジャイルデータオペレーション (DataOps) を導入することで、データエンジニア、データサイエンティスト、データ所有者、アナリストのサイロを破壊することで、データアーキテクチャを大幅に強化できます。 DataOps はチーム間のコミュニケーションを改善し、サイクル時間を短縮し、高いデータ品質を確保します。データおよび分析アーキテクチャは、ビジネスニーズの変化と技術の進歩により、時間の経過とともに多くの変革を経験しています。組織は、時間の経過とともに進化し、ビジネスをサポートするデータと分析アーキテクチャを開発、実装、維持するよう努める必要があります。

データエンジニアリング

組織全体のデータフローを自動化およびオーケストレーションします。

メタデータを使用して、生データを処理し、最適化された出力を生成するパイプラインを自動化します。プラットフォームアーキテクチャとプラットフォームエンジニアリング機能、およびオペレーションの観点から定義されている AWS CAF既存のアーキテクチャガードレールとセキュリティコントロールを活用します。プラットフォームエンジニアリングイネーブルメントチームと協力して、パイプラインのデプロイを簡素化する一般的なパターンの再利用可能な設計図を作成します。

スタート

データレイクをデプロイする

構造化データと非構造化データに適したストレージソリューションを使用して、基本的なデータストレージ機能を確立します。これにより、さまざまなソースからデータを収集して保存し、データにアクセスして、さらなる処理と分析を行うことができます。データストレージは、データエンジニアリング戦略の重要な要素です。適切に設計されたデータストレージアーキテクチャにより、組織はデータを効率的かつ費用対効果の高い方法で保存、管理、アクセスできるようになります。 AWS は、特定のビジネスニーズを満たすためにさまざまなデータストレージサービスを提供します。

例えば、オブジェクトストレージには Amazon Simple Storage Service (Amazon S3)、リレーショナルデータベースには Amazon Relational Database Service (Amazon RDS) 、データウェアハウスには Amazon Redshift を使用して、基本的なデータストレージ機能を確立できます。これらのサービスは、データを安全かつ費用対効果の高い方法で保存し、データに簡単にアクセスして、さらなる処理と分析を行うのに役立ちます。また、パフォーマンスを向上させ、コストを削減するために、データのパーティショニングや圧縮などのデータストレージのベストプラクティスを実装することをお勧めします。

データ取り込みパターンの開発

データフローを自動化してオーケストレーションするには、データ取り込みプロセスを確立して、データベース、ファイル、 などのさまざまなソースからデータを収集しますAPIs。データ取り込みプロセスは、ビジネスの俊敏性をサポートし、ガバナンスコントロールを考慮する必要があります。

オーケストレーターは、クラウドベースのサービスを実行し、自動化されたスケジューリングメカニズムを提供できる必要があります。ポーリング機能とエラー処理機能に加えて、タスク間の条件付き

リンクと依存関係のオプションを提供する必要があります。さらに、パイプラインがスムーズに実行されるように、アラートシステムやモニタリングシステムとシームレスに統合する必要があります。

一般的なオーケストレーションメカニズムには、次のようなものがあります。

- 時間ベースのオーケストレーションは、再帰的な間隔と定義された頻度でワークフローを開始します。
- イベントベースのオーケストレーションは、ファイルの作成やAPIリクエストなどのイベントの発生に基づいてワークフローを開始します。
- ポーリングは、タスクまたはワークフローが (例えば、 を介してAPI) サービスを呼び出し、定義されたレスポンスを待ってから次のステップに進むメカニズムを実装します。

最新のアーキテクチャ設計では、クラウドでのインフラストラクチャ管理を簡素化し、デベロッパーやインフラストラクチャチームの負担を軽減するマネージドサービスを活用することを重視しています。このアプローチは、データエンジニアリングにも適用されます。必要に応じてマネージドサービスを使用してデータ取り込みパイプラインを構築し、データエンジニアリングプロセスを加速することをお勧めします。これらのタイプのサービスの 2 つの例は、Amazon Managed Workflows for Apache Airflow (Amazon MWAA) と です AWS Step Functions。

- Apache Airflow は、プログラムでワークフローを作成、スケジューリング、モニタリングするための一般的なオーケストレーションツールです。 は、Amazon Managed Workflows for Apache Airflow (Amazon MWAA) を、オーケストレーションツールのインフラストラクチャを管理するのではなく、開発者が構築に集中できるようにするマネージドサービスとして AWS 提供します。Amazon MWAAでは、Python スクリプトを使用してワークフローを簡単に作成できます。指向非循環グラフ (DAG) は、各タスクの関係と依存関係を示す方法で、ワークフローをタスクのコレクションとして表します。DAGs 必要な数だけ持つことができ、Apache Airflow は各タスクの関係と依存関係に従ってそれらを実行します。
- AWS Step Functions は、開発者が IT およびビジネスプロセスを自動化するためのローコードのビジュアルワークフローを構築するのに役立ちます。Step Functions で構築するワークフローはステートマシン と呼ばれ、ワークフローの各ステップはステート と呼ばれます。Step Functions を使用して、組み込みのエラー処理、パラメータの渡す、推奨されるセキュリティ設定、および状態管理のワークフローを作成できます。これにより、書き込みとメンテナンスが必要なコードの量が減少します。タスクは、オンプレミスまたはクラウド環境でホストする別の AWS サービスまたはアプリケーションと調整することで作業を実行します。

データ処理の高速化

データ処理は、現代の組織によって収集された膨大な量のデータを理解する上で重要なステップです。データ処理を開始するために、 は、強力な抽出AWS Glue、変換、ロード (ETL) 機能を提供するなどのマネージドサービス AWS を提供します。組織は、これらのサービスを使用して、データのクリーニング、正規化、集計など、未加工データの処理と変換を開始して、分析の準備をすることができます。

データ処理は、最初のデータ変換を実行するための集約やフィルタリングなどのシンプルな手法から始まります。データ処理のニーズが進化するにつれて、さまざまなソースからデータを抽出し、特定のニーズに合わせて変換し、一元化されたデータウェアハウスまたはデータベースにロードして統合分析できるようにする、より高度なETLプロセスを実装できます。このアプローチにより、データが正確で完全であり、タイムリーに分析できるようになります。

AWS マネージドサービスを使用してデータ処理を行うことで、組織はより高いレベルの自動化、スケーラビリティ、コスト効率のメリットを得ることができます。これらのサービスは、スキーマ検出、データプロファイリング、データ変換など、多くのルーチンデータ処理タスクを自動化し、より戦略的なアクティビティのために貴重なリソースを解放します。さらに、これらのサービスは自動的にスケーリングされ、増加するデータボリュームをサポートします。

データ視覚化サービスを提供する

データ視覚化を使用してデータを有意義かつ迅速に解釈する意思決定者がデータを利用できる方法を見つけます。視覚化により、技術的なスキルに関係なく、パターンを解釈し、さまざまな利害関係者のエンゲージメントを高めることができます。優れたプラットフォームにより、データエンジニアリングチームは、データ視覚化を迅速かつわずかなオーバーヘッドで提供するリソースをプロビジョニングできます。また、エンジニアリングの専門知識を必要とせずにデータストアに簡単にクエリできるツールを使用して、セルフサービス機能を提供することもできます。データビジュアルやインタラクティブダッシュボードを通じてサーバーレスのビジネスインテリジェンスを提供し、自然言語を使用してバックエンドデータをクエリできる組み込みツールの使用を検討してください。

アドバンス

ほぼリアルタイムのデータ処理を実装する

データ処理は、あらゆるデータエンジニアリングパイプラインに不可欠なコンポーネントであり、組織は生データを有意義なインサイトに変換できます。従来のバッチ処理に加えて、現在のペースの速

いビジネス環境では、リアルタイムデータ処理がますます重要になっています。リアルタイムデータ処理により、組織はイベントの発生時に対応し、意思決定と運用効率を向上させることができます。

データ品質の検証

データ品質は、データから導き出されるインサイトと決定の精度と信頼性に直接影響します。データの検証とクレンジングプロセスを実装することは、分析に高品質で信頼できるデータを確実に使用する上で不可欠です。

データ検証には、事前定義されたルールと基準に照らしてデータの精度、完全性、一貫性を確認することが含まれます。これにより、データ内の不一致や工ラーを特定し、目的に合ったデータを確保できます。データクレンジングには、データ内の不正確さ、不整合、または重複の特定と修正が含まれます。

データ品質プロセスとツールを実装することで、組織はデータから得られるインサイトの精度と信頼性を向上させることができ、意思決定と運用効率が向上します。これにより、組織のパフォーマンスが向上するだけでなく、生成されたデータと分析に対する利害関係者の信頼と信頼も高まります。

データ変換サービスの証明

データ変換は、高度な分析モデルと機械学習モデルのためのデータを準備します。これには、データの正規化、エンリッチメント、重複排除などの手法を使用して、データがクリーンで一貫性があり、分析の準備が整っていることを確認します。

- データ正規化には、データを標準形式に整理し、冗長性を排除し、異なるソース間でデータに一貫性を持たせることが含まれます。これにより、複数のソースからのデータの分析と比較が容易になり、組織がオペレーションをより包括的に理解できるようになります。
- データエンリッチメントには、属性データや市場トレンドなどの外部ソースからの追加情報を含む既存のデータの強化が含まれます。これにより、内部データソースだけでは明らかではない可能性がある顧客の行動や業界の傾向に関する貴重なインサイトが得られます。
- 重複排除には、重複するデータ入力を特定して削除し、データが正確で工ラーがないことを確認する必要があります。これは、少数の重複であっても分析の結果を歪める可能性がある大規模なデータセットを処理する場合に特に重要です。

高度なデータ変換技術を使用することで、組織はデータの品質、正確性、より複雑な分析の準備が整っていることを保証します。これにより、より良い意思決定、運用効率の向上、市場での競争上の優位性につながります。

データ民主化を有効にする

すべての従業員がデータにアクセスしやすく、理解しやすく、使用できるようにすることで、データ民主化の文化を促進します。データ民主化は、従業員がデータ駆動型の意思決定を行い、組織のデータ駆動型の文化に貢献するのに役立ちます。つまり、サイロを破壊し、意思決定を推進するためにすべての従業員がデータを共有して使用する文化を創造します。

全体として、データ民主化とは、組織内のすべての人がデータを評価し、アクセス可能で理解しやすい文化を創造することです。データ民主化を可能にすることで、組織はイノベーションを推進し、意思決定を改善し、最終的にはビジネスの成功につながるデータ駆動型の文化を育みます。

Excel

UI ベースのオーケストレーションを提供する

アジャイルで効果的なアプローチを使用する組織を構築するには、ビジネスライン全体の開発および運用リソースで使用される最新のオーケストレーションプラットフォームを計画することが重要です。目標は、単一のチーム、テクノロジー、サポートモデルに依存しずに、データパイプラインとワークフローを開発、デプロイ、共有することです。これは、UI ベースのオーケストレーションなどの機能によって実現されます。インタラクションなどの drag-and-drop機能により、技術的な専門知識がほとんどないユーザーは、マシンデータフローを構築DAGsしてステートできます。これらのコンポーネントは、データパイプラインをオーケストレーションする実行可能コードを生成できます。

DataOps は、データ管理の複雑さを克服し、組織全体のシームレスなデータフローを確保します。メタデータ駆動型のアプローチにより、組織のマンデートに従ってデータ品質とコンプライアンスが保証されます。マイクロサービス、コンテナ化、サーバーレス機能などのツールセットへの投資により、スケーラビリティと俊敏性が向上します。

データエンジニアリングチームに頼ってデータから価値を生み出し、インフラストラクチャタスクをオートメーションに任 day-to-dayせることで、組織はオートメーションとオーケストレーションの卓越性を実現できます。データフロー管理タスクのほぼリアルタイムのモニタリングと ログ記録は、即時修復アクションをサポートし、データフローパイプラインのパフォーマンスとセキュリティを向上させます。これらの原則は、セキュアなデータ共有モデルを確保しながらスケーラビリティとパフォーマンスを実現し、将来の成功に向けて組織をセットアップするのに役立ちます。

統合 DataOps

DataOps は、データパイプラインの作成、テスト、デプロイを合理化するための開発プロセスと運用プロセスの統合を強調する、データエンジニアリングに対する最新のアプローチです。 DataOps ベストプラクティスを実装するために、組織はInfrastructure as Code (IaC) と継続的統合と継続的配信 (CI/CD) ツールを使用します。これらのツールは、パイプラインの自動作成、テスト、デプロイをサポートしているため、効率が大幅に向上し、エラーが軽減されます。 DataOps チームはプラットフォームエンジニアリングの有効化チームと協力してこれらの自動化を構築し、各チームが最善を尽くすことに集中できるようにします。

DataOps 方法論を実装することで、データエンジニア、データサイエンティスト、ビジネスユーザーのコラボレーション環境を育成し、データパイプラインと分析ソリューションの開発、デプロイ、モニタリングを迅速に行うことができます。このアプローチにより、チーム間のコミュニケーションとコラボレーションがよりシームレスになり、イノベーションの迅速化と成果の向上につながります。

の利点を最大限に活用するには DataOps、データエンジニアリングプロセスを合理化することが重要です。これは、コードレビュー、継続的統合、自動テストなど、プラットフォームエンジニアリングチームのベストプラクティスを使用して実現されます。これらのプラクティスを実装することで、組織はデータパイプラインの信頼性、スケーラビリティ、安全性を確保し、ビジネスステークホルダーと技術ステークホルダーの両方のニーズを満たすことができます。

プロビジョニングとオーケストレーション

承認されたクラウド製品のカタログを作成、管理、およびユーザーに配布します。

組織の成長に合わせて、一貫性があり、スケーラブルで反復可能な方法でインフラストラクチャをプロビジョニングすることは、より困難になります。プロビジョニングとオーケストレーションを合理化することで、一貫したガバナンスを実現し、コンプライアンス要件を満たすと同時に、承認されたクラウド製品のみをデプロイできます。

組織内で事前承認された製品を再利用することで、デベロッパーは組織のセキュリティとガバナンスの要件を満たしながら、より迅速かつ一貫してアプリケーションを構築できます。

スタート

hub-and-spoke カタログモデルのデプロイ

ポートフォリオとしてサービスカタログで管理されるソフトウェアアセットは、 hub-and-spoke パターン内で 1 つ以上のアカウントのユーザーと共有されます。プライベートマーケットプレイスとプライベートオファーを使用して、サードパーティーソリューションのいくつかをキュレートし、Infrastructure as Code (IaC) テンプレートで配布できます。

ビルダーが事前承認された製品を消費できるようにするには、これらの製品を確認して承認し、ユーザーに公開するプロセスを定義します。まず、これらの事前承認済み製品を含む一元管理されたリポジトリを設計して実装します。組織のユーザーが各製品を消費する必要がある場合に、このリポジトリ内のライセンスと製品へのアクセスを許可するシステムを設計します。

組織内のビルダーが製品を公開メカニズムに送信して承認できるようにし、承認後に組織内のすべてのユーザーがこれらの製品を使用できるようにします。

再利用のためのテンプレートのキュレート

ソリューションの IaC テンプレートを体系化し、モデルを定義 hub-and-spokeしたら、スポークアカウントごとに 2 つのカテゴリのテンプレートを定義する必要があります。プロビジョニング/強制され、 を消費できます。プロビジョニング/強制テンプレートは、基本機能として管理アカウントから各メンバーアカウントに直接プロビジョニングされます。テンプレートを使用できるので、ビルダーはセルフサービス方式で参照およびプロビジョニングできます。

再利用のためにデフォルトパラメータを適用する

ビルダーが事前に選択できるデフォルトパラメータを含む IaC テンプレートを実装します。これにより、ビルダーは各パラメータの詳細を評価することなくガバナンスに合わせることができ、誤った選択をできなくなります。このアプローチでは、セットアップに必要なもののみが公開されます。例えば、 は、特定のポートフォリオの製品に適用されるルールを制御する制約機能を使用してこのアプローチAWS Service Catalogを実装します。このカスタマイズは、ビルダーチームがテンプレートのセルフサービスプロビジョニングを使用する場合に事前設定されます。

承認プロセスを確立する

製品を使用するビジネス上の根拠がある場合、ユーザーは承認されていない製品へのアクセスリクエストを送信できる必要があります。使用している製品の更新が利用可能になったときにユーザーに通知する通知システムを構築し、最新のセキュリティ更新プログラムに準拠できるようにします。

セルフサービスポータルを通じて、ビルダーがレビュー用の新製品を送信するためのワークフローを確立します。ビルダーはポータルを使用して製品の対象者を定義し、製品にアクセスできるユーザーグループを特定できます。送信ごとに、定義されたプロセスを使用して、製品をセルフサービスポータルにレビュー、承認、公開します。

事前

セルフサービスポータルを作成する

セルフサービスポータルを作成して、承認されたクラウド製品を配布、閲覧、消費します。組織内のユーザーは、このポータルを使用して、インフラストラクチャの構築や環境へのアプリケーションのデプロイに必要な製品を検索できます。ポータル内の製品にアクセスできるユーザーのアクセス許可の境界を設定し、ユーザーがライセンス製品を使用できる回数に制限を設定します。のカスタマイズAWS Control Towerなどのソリューションを使用してアカウントが作成されるため、各スポークアカウントで直接プロビジョニングまたはセルフサービスモデルとして利用できるリソースの基本セットを定義します。

プライベートマーケットプレイスを有効にする

プライベートマーケットプレイスは、購入した製品 (ソフトウェア、データ、プロフェッショナルサービス) の厳選されたカタログを提供し、スポークアカウントが承認されたソフトウェアのみをサブスクライブできるように hub-and-spoke パターン (1 つの管理アカウントと複数のメンバーアカウント) で実装されます。この製品ガバナンスは、ソフトウェアコストを管理し、法的および契約上の

レビューを合理化するのに役立ちます。管理アカウントレベルでプライベートマーケットプレイスを作成し、プライマリハブとして機能します。

使用権限の管理

承認されたユーザーとワークロードのみがベンダー定義の制限内でライセンスを使用できるようにするコントロールを有効にします。これにより、コストのかかる監査や予期しないライセンス調整のリスクが軽減されます。

Excel

調達システムとの統合

既存の調達プロセスを に統合して補完しますAWS Marketplace。これは、調達システム (Coupa または SAP Ariba) をプライベートマーケットプレイスに拡張して、ユーザーが既存の調達および承認プロセスに従ってソフトウェアを取得できるようにすることで行われます。適切な IAM 管理アクセス許可を作成し、 AWS Marketplace を使用して調達ソリューションの設定に必要な情報を生成し、調達ソリューションを設定して統合を完了します。例えば、パンチアウト をセットアップし、 AWS 請求書に発注書をアタッチし、調達プロセスを調整して標準プロビジョニングソリューションを使用できます。

ビルダーが内部 API を通じて事前承認された製品にアクセスできるようにすることで、ユーザーはその製品をアプリケーションに組み込むか、チームが製品を使用するためにパーソナライズされた独自のポータルを構築できます。新しい製品を作成するための送信と公開のプロセスを統合し、ユーザーが APIsを通じて新しいライセンスと製品へのアクセスをリクエストできるようにします。

ITSM ツールとの統合

該当する場合は、IT サービス管理 (ITSM) ツールに接続し、設定管理データベース (CMDB) の更新を自動化します。組織が使用する製品を評価するためのプロセスとメカニズムを確立します。コンプライアンスのために更新する必要があることを事前に承認された製品をユーザーに通知するためのメカニズムを確立します。ITSM ツールを使用して環境を分析し、重要な更新が必要な場合にセキュリティおよびコンプライアンスの更新を組織全体の製品にプッシュします。

ライフサイクル管理とバージョン分散システムの実装

IaC テンプレートのバージョンと、テンプレートからプロビジョニングされた のサービスのバージョンを開発ライフサイクルを通じて維持します。カタログに実装した hub-and-spoke モデルを使

用して、スポークレベルで強制更新が必要かどうか (例えば、セルフサービスプロビジョニングに同時バージョンが利用可能な場合) 、およびどのバージョンを廃止対象としてマークする必要があるかを定義できます。 hub-and-spoke カタログを使用すると、必要に応じて新しいバージョンの監査と配布を管理することもできます。

最新のアプリケーションの開発

適切に設計されたクラウドネイティブアプリケーションを構築します。

最新のアプリケーション開発プラクティスは、組織が適切に設計されたクラウドネイティブアプリケーションを構築し、競争力を維持するために不可欠です。企業は、コンテナやサーバーレスコンピューティングなどのクラウドネイティブテクノロジーを使用して、変化する市場需要に適応するスケーラブルでアジャイルなアプリケーションを作成できます。これらのテクノロジーにより、組織はリソースの使用率を最適化し、コストを削減し、アプリケーションのパフォーマンスを向上させることができます。

最新のアプリケーションを設計するときは、運用と開発のためのアジャイルソリューションを開発します。最新のアプリケーションは、顧客の需要の変化に自動的に反応し、障害に強いです。エンジニアは変更を迅速に開発してデプロイし、アプリケーションのパフォーマンスをモニタリングできます。最新のアプリケーションは、自己修復性を備え、必要に応じてトラフィックをゼロコストでなくすなど、大小両方のトラフィックにスケーリングできるように設計されています。

適切に設計されたクラウドネイティブアプリケーションを構築するには、基盤となるテクノロジーとそのベストプラクティスを深く理解する必要があります。組織はマイクロサービスアーキテクチャを採用し、アプリケーションをモジュール化して疎結合に設計し、独立したデプロイとスケーラビリティを実現する必要があります。このアプローチにより、組織はアプリケーションを、迅速かつ個別に開発、テスト、デプロイされる、より小さく管理しやすいコンポーネントに分割できます。

スタート

最新のアプローチを詳しく見る

まず、コンテナ、サーバーレステクノロジー、およびマイクロサービス の開発を可能にするその他のアプローチを調査します。これにより、リソースの効率が向上し、セキュリティが向上し、インフラストラクチャのコストが最小限に抑えられます。既存の差別化アプリケーションとエンタープライズアプリケーションをモダナイズして、効率を向上させ、既存の投資の価値を最大化することを選択します。価値主導の意思決定に基づいて、リプラットフォーム (セルフマネージドコンテナ、データベース、またはメッセージブローカーをマネージドクラウドサービスに移行する) とリファクタリング (クラウドネイティブアーキテクチャを採用するようにアプリケーションを再開発する) を検討してください。

既存のクラウドベースのアプリケーションを更新する場合、アプローチを成功させるには、strangler fig パターンを使用してアーキテクチャをマイクロサービスに徐々に分解する必要があります。この

手順は、従来のアプリケーション手法を採用するのに役立ちます。これにより、固有の利点を実現し、その価値を大規模な組織に実証できます。必要に応じて、イベント駆動型アーキテクチャを活用する個別のマイクロサービスとしてアプリケーションを構築することを検討してください。ワークロードのパフォーマンスや信頼性に影響を与えないように、変更不可能なサービスクォータと物理リソースがアーキテクチャで考慮されていることを確認してください。

クラウドネイティブなコンピューティング機能を採用する

クラウドネイティブなコンピューティング機能は、最新のアプリケーション開発の鍵です。このアプローチでは、コンピューティングユニットをどのようにホストするかを検討し、各ユースケースまたはサービスに最適なオプションを特定する必要があります。例えば、 AWS Lambdaはアプリケーションコードを実行するためのサーバーレスメカニズムを提供し、イベント駆動型アーキテクチャで重要な役割を果たします。Lambda 関数はオンデマンドで起動され、定義された最大同時実行数まで並行して実行されるため、さまざまなタスクを実行するようにスケーリングできます。

コンテナ化を使用する

最新のソフトウェア開発では、アプリケーションとその依存関係の管理は、特にさまざまな環境間で一貫性を維持する必要性を検討する場合に、ますます複雑になっています。これらの課題に対処するために、Docker などのコンテナ化テクノロジーは、アプリケーションとその依存関係をパッケージ化するための効果的なソリューションとして浮上しました。コンテナは、アプリケーションのランタイム環境に関係なく、一貫性と再現性のあるデプロイを保証するため、ローカル環境の開発は、クラウド環境の本稼働開発と同じ方法で動作します。このアプローチにより、環境またはその設定内の不一致によって引き起こされる可能性のあるエラーが軽減されます。

最新のデータベースを使用する

最新のデータベースを使用すると、アプリケーション内の各マイクロサービスが要件を満たす適切な専用データベースを使用できるため、俊敏性とパフォーマンスが向上し、コストを削減できます。例えば、あるマイクロサービスがセッションデータの保存時に NoSQL データベースを使用して高スループットを実現し、別のマイクロサービスがリレーショナルデータベースを使用して複雑なテーブル結合を行い、さらに別のマイクロサービスが量子台帳データベースを使用してブロックチェーンの変更を追跡する場合があります。

最新のデータベースはスケーラビリティと柔軟性を提供します。また、従来のデータベースよりもセキュリティ、コンプライアンス、信頼性の向上にも役立ちます。これにより、組織はデータをより効率的に保存および管理し、アプリケーションが適切なデータに適切なタイミングでアクセスできるようになり、パフォーマンスとユーザーエクスペリエンスが向上します。

最新のデータベースへの移行は、最新のアプリケーション開発の重要なコンポーネントです。適切なデータストレージソリューションを使用することで、組織はデータ管理機能を最適化し、より効率的で信頼性の高いアプリケーションを提供できます。各マイクロサービスを独立させ、各マイクロサービスに適したテクノロジーを選択することで、組織はデータ機能をさらに最適化し、コストを最小限に抑えながら最大限の効率とスケーラビリティを実現できます。

事前

最新のアーキテクチャを最適化する

さらなる最適化を実現するには、サーバーレステクノロジーの実装を改良し、Amazon API Gateway や などの AWS サービスを使用して個別にスケーリングおよびデプロイできるアーキテクチャを開発しますAWS Lambda。Amazon Route 53 と を使用してサービス検出を実装AWS Cloud Mapし、コンポーネント間のシームレスな通信を確保します。

API バージョニング、キャッシュ、レート制限を採用して、さまざまなアプリケーションバージョン間で互換性とパフォーマンスを維持します。AWS Identity and Access Management (IAM) とリソースポリシーを使用してセキュリティを強化します。これにより、インフラストラクチャが保護され、承認されたエンティティにのみアクセスが許可されます。

可能であれば、サーバーレスサービスを使用して、基盤となるインフラストラクチャを管理することなくコンテナを実行します。これにより、コアアプリケーションの開発に集中し、リソースの管理とパフォーマンスを向上させることができます。また、スケーラビリティ、柔軟性、コスト効率のメリットを最大限に活用するのに役立ちます。

サーバーレスアーキテクチャの複雑さをより深く掘り下げ、これらの高度なプラクティスを組み込むことで、組織は改善と微調整の機会を発見し、最終的にクラウドネイティブアプリケーションの可能性を最大化できます。この取り組みにより、より高度なアプリケーションパターンの導入が容易になり、全体的なユーザーエクスペリエンスがさらに向上します。また、組織がソフトウェア開発プロセスをより機敏かつ効率的に行えるようになります。

サービスメッシュテクノロジーを使用する

組織がアプリケーションの構築とデプロイにマイクロサービスアーキテクチャを採用するにつれて、これらのサービス間の複雑さ、セキュリティ、通信を管理することが重要になります。Istio、Linkerd、Consul などのサービスメッシュテクノロジーは、マイクロサービスのセキュリティ、オブザーバビリティ、信頼性を向上させる上で重要な役割を果たします。

可視性とトレーサビリティを確保する

最新のプラクティスでは、開発プロセスにおける可視性とトレーサビリティが向上し、業界標準とベストプラクティスへの準拠が容易になります。可視性とモニタリングは、最新のアプリケーション開発に不可欠です。モニタリングおよびログ記録ソリューションを実装してアプリケーションのパフォーマンスに関する貴重なインサイトを提供することで、組織は改善すべき分野を特定し、アプリケーションを最適化できます。プラットフォームエンジニアリングチームと協力して、アプリケーションエラー、パフォーマンス、コンプライアンスの可視性とモニタリングを提供する end-to-end ツールを利用できるようにすることをお勧めします。これにより、問題を迅速に検出、診断、解決できます。

Excel

マイクロサービスの導入

多くの組織にとって、最新のアプリケーション開発はビジネスの成功と同義です。マイクロサービスはこの変革の中核であり、組織はこれらの強力なアーキテクチャパターンを採用することでメリットを得ることができます。

マイクロサービスは、スケーラビリティ、耐障害性、俊敏性に優れたアプリケーションアーキテクチャを提供します。アプリケーションを小規模で個別にデプロイ可能なサービスに分割することで、組織はアプリケーションの他の部分に影響を与えることなく、特定のコンポーネントを迅速に反復処理することを選択できます。サーキットブレーカーやバルクヘッド などの高度な障害耐性パターンは、これらのアプリケーションの高可用性を確保する上で重要な役割を果たします。

サーキットブレーカーは、異常なサービスからの通信を一時的に停止またはシフトして、障害がカスケードするのを防ぐ安全メカニズムとして機能し、回復できます。バルクヘッドはリソースを分離し、潜在的な障害の影響範囲を制限します。これらのパターンを組み合わせることで、予期しない中断に耐え、最適なパフォーマンスを維持する堅牢なアーキテクチャを構築できます。

マイクロサービスを実装するもう 1 つの重要な点は、ドメイン駆動型設計 (DDD) の原則の導入です。DDD は、ビジネスドメインに関する共通の理解を確立し、それを適切に構造化されたソフトウェア設計に変換することに重点を置いています。このアプローチは、よりまとまりのある保守可能なマイクロサービスにつながり、アプリケーションが組織のニーズに合わせて進化することを確実にします。

マイクロサービスベースのアプリケーションでは、サービス間通信の最適化も重要です。gRPC や GraphQL などの高度なプロトコルを実装することで、組織はサービス間の通信効率を大幅に向上さ

せることができます。これらのプロトコルは、タイプの安全性、低レイテンシー、柔軟性などの機能を提供し、アプリケーション全体のパフォーマンスと保守性を向上させるのに役立ちます。

マイクロサービスを導入する組織は、イノベーション、俊敏性、コラボレーションを促進する環境を提供します。開発チームは通常、ビジネス機能を中心に編成されており、継続的インテグレーションと継続的デリバリー (CI/CD) プラクティスに重点を置いています。意思決定、実験、反復を迅速に行う権限があり、責任と説明責任を共有する文化を受け入れます。

継続的な統合と継続的な配信

従来のソフトウェア開発およびインフラストラクチャ管理プロセスを使用する組織よりも、アプリケーションとサービスを迅速に進化および改善します。

継続的な統合と継続的な配信によるDevOpsプラクティスの採用 (CI/CD) promotes a streamlined, automated, and efficient process for building, testing, and deploying applications. CI/CD により、ソフトウェアの迅速な配信が可能になり、デプロイエラーのリスクが軽減され、アプリケーションが常に最新の機能とバグ修正で最新状態になります。主な目的は、従来のソフトウェア開発およびインフラストラクチャ管理プロセスの使用から進化することで、アプリケーションとサービスをより迅速に進化および改善することです。

スタート

ソフトウェアコンポーネント管理を採用する

ソフトウェアコンポーネント管理は、ライブラリ、フレームワーク、ソースコードリポジトリ、モジュール、アーティファクト、サードパーティーの依存関係など、ソフトウェアの構築に使用されるすべての個々のコンポーネントを管理する練習です。Git や Apache Subversion などのバージョン管理システムを使用して、ソースコードの管理、コラボレーションの有効化、コード変更の履歴の維持を行うことをお勧めします。リポジトリの変更とイベントをモニタリングして、プロセスを自動化し、パイプラインを作成し、コードを管理し、必要に応じてワークフローを追加の サービスに統合できます。

CI/CD パイプラインを作成する

CI/CD pipelines are sets of automated instructions that are initiated by changes committed to the version control system. They typically include instructions for building the application, running automated tests, and deploying code to a specific environment. You can set up an automated CI/CD AWS CodePipeline、Jenkins GitLab、または CircleCI などのツールを使用してパイプラインを作成します。パイプライン生成をサポートするバージョン管理システムで直接セットアップすることもできます。

継続的統合のための最小限の実行可能なパイプラインから始め、さらに多くのアクションとステージを含む継続的デリバリーパイプラインに移行します。継続的配信設定をコードとして扱います。ブランチとチームごとに複数の個別のパイプラインを使用できるため、設定する必要がある設定変数と、パイプラインを使用するチームに最適なサポート方法を検討してください。

デプロイウィンドウ — コードをデプロイする日時を検討してください。システムの低需要時間を考慮すると、ロールバックする必要がある場合は、顧客への影響が最も少なくなります。その他のベストプラクティスには、金曜日のデプロイの回避や、ピーク時または祝祭日前のコードフリーズの実装などがあります。コミットの作成者が利用できない場合 (休暇中など) のコードのデプロイに関するルールの定義を検討してください。デプロイが失敗し、外部ヘルプに依存する必要がある場合があることに注意してください。インプレース、ローリング、イミュータブル、ブルー/グリーンデプロイなど、さまざまなデプロイ方法を評価します。可用性とセキュリティを向上させながら、複雑さと管理を最小限に抑えるために、継続的な配信ワークフローにフルマネージドサービスを使用することを検討してください。

自動テストのデプロイ

最新のプラクティスでは、リポジトリにコミットしてパイプラインを開始する前に、問題を検出して修正するために、左にシフト (デベロッパーと の近くでテストを に移動しIDE、ライフサイクルの前半) することをお勧めします。この練習では、開発者とのクイックフィードバックループを使用します。これは、開発者がコーディングしている間にエラーが検出されるためです。左へのシフトは、テストに実行中のパイプラインを必要としないため、コストの低下に関連しており、非同期フィードバックや運用コストの増加につながる可能性があります。

自動テストは、開発プロセスの早い段階でエラーをキャッチし、ユニットテスト、統合テスト、機能テストが含まれます。デベロッパーには、可能な限り早期にユニットテストを作成し、中央リポジトリにコードをプッシュする前に実行するツールを使用することをお勧めします。さらに、自動プロセスに静的コード分析、パフォーマンスベンチマーク、セキュリティアプリケーションテストが含まれていることを確認してください。

ドキュメントの作成

CI/CD pipeline to streamline development workflows, you should maintain clear and comprehensive documentation to ensure the pipeline's ongoing effectiveness, maintainability, and scalability. Documentation is a vital aspect of CI/CD パイプラインの実装に加えて、開発チームはパイプラインの設計、コンポーネント、プロセスを明確に理解できるためです。ドキュメントを作成するときは、パイプラインの概要から始めて、アーキテクチャと設計のトレードオフを説明し、使用されているツールとテクノロジーを説明し、初期設定と設定を指定し、セキュリティ対策とアクセスコントロールの概要を示し、トラブルシューティングとメンテナンスの情報を含めます。

インフラストラクチャをコードとして使用する

Terraform、Ansible、 などのツールを使用してインフラストラクチャAWS CloudFormationを管理し、一貫性と再現性のある環境を確保します。インフラストラクチャをコードとして扱い、インフラ

ストラクチャの変更を追跡し、コンソールで直接変更を行わないようにします。データベースプロビジョニングを含むすべてのインフラストラクチャをコードとして定義し、パイプラインを使用してこれらの変更をデプロイします。サニタイズされた本番データの小さなサブセットを使用して、パイプラインでデータベース統合をコードとして実行することを検討してください。可能であれば、変更を加え、コード内の変更を追跡します。

ソフトウェアコードと同様に、インフラストラクチャコードの以下のベストプラクティスに従ってください。

- バージョンコントロールを使用します。
- バグ追跡とチケットシステムを使用します。
- 適用する前に、同僚に変更を確認してもらいます。
- インフラストラクチャコードのパターンと設計を確立します。
- インフラストラクチャの変更をテストします。

標準メトリクスを維持および追跡する

高いレベルのパフォーマンスを維持するには、主要なメトリクスに対して開発および追跡して、パイプラインの健全性とビジネスへの影響を理解します。これには、以下が含まれます。

- ビルド頻度。ビルドの数は、チームの生産性と変更の複雑さに関するインサイトを提供します。
- デプロイ頻度。定期的なデプロイは、健全でアジャイルな開発プロセスを示しています。
- 変更のリードタイム。変更が本番稼働するまでの平均時間を測定すると、デプロイプロセスのボトルネックを特定するのに役立ちます。
- パイプラインの平均所要時間。初期パイプラインステージから後続の各ステージまでの平均時間は、ワークフローの最適化に役立ちます。
- 本番稼働変更ボリューム。本稼働環境に到達した変更の数を追跡することで、本稼働環境の安定性に関するインサイトを得ることができます。
- ビルド時間。平均ビルド時間は、コードベースまたはインフラストラクチャで潜在的な問題を示している可能性があります。

アドバンス

設定管理を使用する

設定管理ツールは、ソフトウェアとインフラストラクチャのデプロイ、設定、管理を自動化する上で重要な役割を果たします。さまざまな環境におけるインフラストラクチャ、ソフトウェア、設定の変更を処理し、望ましい状態を維持するための体系的なアプローチを提供します。これらのツールを使用すると、デベロッパーは宣言言語または必須言語を使用してシステムの望ましい状態を定義できます。次に、設定管理ツールは、これらの設定をターゲットシステムに適用するプロセスを自動化し、一貫性と再現性を確保します。

設定管理ツールを使用して、ソフトウェアとインフラストラクチャのデプロイ、設定、管理を自動化します。AWS Systems Manager State Manager は、マネージドノードやその他の AWS リソースを定義した状態に保つプロセスを自動化する、安全でスケーラブルな設定管理サービスです。

モニタリングとログ記録の統合

モニタリングとログ記録のソリューションを CD パイプラインに統合すると、開発チームやソフトウェア開発プロセス全体に多くの利点があります。これらのソリューションは、アプリケーションのパフォーマンスに関するリアルタイムのインサイトを提供し、問題のより迅速な特定と解決を可能にし、継続的な改善を促進し、アプリケーションのライフサイクルを通じて信頼性、パフォーマンス、スケーラビリティを維持するのに役立ちます。モニタリングおよびログ記録ソリューションへの投資は、堅牢で効率的な CD パイプラインを維持する上で重要な側面であり、最終的には高品質のソフトウェアの提供の成功に貢献します。

マージのケイデンスを作成する

メインライン (トランクまたはメイン) ブランチにコード変更を少なくとも 1 日に 1 回、または理想的には各タスクの後に 1 日に複数回コミットまたはマージします。このケイデンスは、複数の毎日のパイプライン呼び出しにつながります。プルベースの分岐ワークフローモデルは、このアプローチと一致します。機能フラグ 、ダーク起動 、および同様の手法を使用して、顧客が使用する機能をカスタマイズします。

デプロイ後の動作をキャプチャする

デプロイ後、自動合成テストを使用して本番環境の動作をキャプチャし、結果を継続的配信パイプラインと同期して、是正措置が迅速に行われるようにします。デベロッパーにとって最優先事項は、パイプラインで検出されたエラーをできるだけ早く修正し、ソースコードリポジトリにコード変更をコミットし、パイプラインでエラー解決を検証することです。

デプロイ後のベストプラクティスには、最も重要な主要業績評価指標 (KPIs) の観察と、本番環境にエラーがないことの検証が含まれます。デプロイ後のエラー処理と評価を自動化KPIsして、リリースの影響を定量化します。デベロッパーが改善に使用できる速度、セキュリティ、安定性のメトリクスを自動的に生成します。詳細については、 のソリューションDevOps モニタリングダッシュボードを参照してください AWS。

Excel

最適なパフォーマンスを実現するために、最先端のプラクティスとテクノロジーを採用します。CI/CD プロセスを継続的に改良することで、ソフトウェアの品質を向上させ、市場投入までの時間を短縮し、俊敏性を高めることができます。新しい手法とツールが継続的に出現するため、組織が常に情報を得て、競争力を維持するために適応することが不可欠です。

適応性を維持するには、以下を考慮してください。

- アプリケーション、設定、インフラストラクチャ、データ、 AWS アカウントと組織、デプロイパイプライン、ネットワーク、セキュリティとコンプライアンスのコントロールなど、すべてをコードとして定義します。
- コンピューティングイメージ、共有サービス、アプリケーションに対応するデプロイパイプラインを作成します。
- コードで説明されているように、プルベースのリクエストが既存のインフラストラクチャの状態を目的の状態と比較することで変更をデプロイするワークフローを開始する GitOps モデルを考えてみましょう。
- CD パイプラインを使用して、機械学習 (ML)、データ、モノのインターネット (IoT)、およびその他のワークロードをデプロイすることを検討してください。
- すべてのビルドアーティファクトにデジタル署名し、安全なリポジトリに保存します。
- 顧客にデプロイされたすべてのバージョニングアーティファクトとデジタル署名アーティファクトの記録を作成するソフトウェア部品表を自動的に生成して、ソフトウェアの出所を追跡します。
- ソフトウェア配信プロセスの手動アクティビティをすべて削除したら、手動レビューボードを削除します。

ソフトウェア配信プロセス全体を自動化したアプリケーションとサービスについては、チームがパイプライン内のすべてのチェックを本番環境の顧客に渡す変更をデプロイする継続的なデプロイを検討してください。視覚化については、ウェブサイトの「継続的配信とは AWS 」の最初の図を参照してください。

AI/ML テクノロジーの統合

人工知能 (AI) と機械学習 (ML) テクノロジーを CI/CD パイプラインに統合すると、次のような利点があります。

- 自動テスト生成
- インテリジェントなテストの優先順位付け
- 問題検出のための予測分析
- 異常検出と根本原因分析
- コードレビューと品質保証
- デプロイの最適化

詳細については、 AWS ウェブサイトの「デベロッパーオペレーションにインテリジェンスを追加する」を参照してください。

混沌エンジニアリングプラクティスを採用する

混沌エンジニアリングでは、システムへの障害の挿入を意図的に行い、予期しないイベントに耐えて回復する能力をテストします。弱点を特定し、積極的に対処することで、組織はシステム全体の信頼性を向上させ、潜在的な問題の影響を最小限に抑えることができます。

Gremlin、Chaos Monkey、Litmus などのツールを使用して、システムの耐障害性をテストするためのカオスエンジニアリングプラクティスを採用します。制御実験を定期的に実行して脆弱性を特定し、耐障害性を検証し、アプリケーションが予期しない障害を正常に処理することを確認します。このプロアクティブアプローチは、システムの信頼性を向上させ、より堅牢な CI/CD パイプラインに貢献します。

パフォーマンスの最適化

プロファイリングツール、リアルタイムモニタリング、フィードバックループを使用して、アプリケーションのパフォーマンスを継続的に最適化します。次のような手法を適用して、アプリケーションが増加したトラフィックと需要を処理できるようにします。

- コードの最適化
- プロファイリング
- リアルタイムモニタリング
- フィードバックループ

- キャッシュ
- 負荷分散
- スケーラビリティとパフォーマンステスト

高度なオブザーバビリティを実装する

クラウドインフラストラクチャのオブザーバビリティを高めることは、メトリクス、ログ、トレースの収集、集約、分析の基本にとどまりません。Amazon CloudWatch や などのツールを使用してオブザーバビリティを強化するとAWS X-Ray、継続的な配信とイノベーションを促進する戦略的プラクティスに進化します。

堅牢な CI/CD パイプラインでは、高度なオブザーバビリティにより、アプリケーションやインフラストラクチャだけでなく、パイプライン自体を含むシステム全体のパフォーマンスとヘルスに関するインサイトも見つけることができます。これらのインサイトは、以下に役立ちます。

- 潜在的な問題を迅速に特定、理解、対処して、アプリケーションの安定性を向上させ、ダウンタイムを短縮する
- CI/CD プロセスを合理化して、より高速で信頼性の高い配信を作成する
- コードの変更とデプロイの影響についてより深いインサイトを得て、情報に基づいた意思決定を推進します。
- リソース使用率を最適化して運用効率とコスト効率を向上させる

オブザーバビリティを向上させるには：

- オブザーバビリティをアプリケーションとインフラストラクチャのすべてのレイヤーに埋め込み、システムのパフォーマンス、動作、ヘルスを包括的に把握します。
- Amazon などのツールを使用してデータ収集、ストレージ、分析を一元化 CloudWatch し、オブザーバビリティデータを統合して、アクセスと解釈を容易にします。
- 分散トレース AWS X-Ray に を使用して、アプリケーションとその基盤となるサービスのパフォーマンスを把握します。
- 継続的改善のためのフィードバックループを確立し、オブザーバビリティデータを使用してシステムの反復的な強化を推進します。

高度なオブザーバビリティを採用することは、システムを維持するだけでなく、運用上の卓越性を達成し、組織内で継続的なイノベーションを推進するための戦略的動きです。

GitOps プラクティスを実装する

Git リポジトリを 1 つの情報源として使用して、インフラストラクチャとアプリケーションの設定を管理する GitOps プラクティスを実装します。このアプローチは、変更管理を簡素化し、トレーサビリティを強化し、環境間の一貫性を確保します。

結論

このガイドは、クラウド導入を成功させるための基盤の実装と管理を成功させるためのプレイブックとして機能します。ここでは、次の方法について説明します。

- プラットフォームアーキテクチャの技術的な課題や複雑さに直接対処し、クラウド環境とその中に存在するデータに関する堅牢なガイドラインと原則を確立します。
- 強力なプロビジョニングとオーケストレーションを使用してプラットフォームエンジニアリングを構築します。 ???
- 承認されたクラウド製品をスケーラブルで繰り返し可能な方法で管理し、ユーザーに配布する、準拠したマルチアカウントクラウド環境の使用を可能にします。
- データエンジニアリングがデータ駆動型の意思決定を推進するために必要なツールで、データアーキテクチャの決定をサポートします。
- これらの機能を最新のアプリケーション開発戦略や CI/CD プロセスと組み合わせて、組織内の俊敏性、効率性、イノベーションを促進します。
- 部門横断的な関係を構築し、他の AWS CAF の視点から独自の意思決定に情報を取り入れて、プラットフォームとその背後にあるチームの成功を確実にします。

詳細情報

AWS クラウド導入フレームワーク (AWS CAF) リソース：

- eBook
- オーディオブック
- インフォグラフィック
- AWS 人工知能、Machine LearningCAF
- ビジネスの視点
- 人材の視点
- ガバナンスの視点
- オペレーションの視点
- セキュリティの視点

その他のリソース：

- AWS アーキテクチャセンター
- AWS 導入事例
- AWS 全般のリファレンス
- 「AWS 用語集」
- AWS ナレッジセンター
- AWS 規範ガイダンス
- AWS パートナーソリューション (旧クイックスタート)
- AWS セキュリティドキュメント
- AWS ソリューションライブラリ
- AWS トレーニングと認定
- AWS Well-Architected
- AWS ホワイトペーパーとガイド
- の開始方法 AWS
- Amazon Web Services の概要

寄稿者

このガイドの寄稿者には以下が含まれます。

- Tony Santiago、シニアパートナーソリューションアーキテクト、 AWS
- Matias Undurraga、エンタープライズ技術者、 AWS
- Alex Torres、シニアソリューションアーキテクト、 AWS
- マイケル・リンドレス、シニア DevSecOps コンサルタント、 AWS
- プリンシパルソリューションアーキテクト、 CloudOps スペシャリスト、Alex livingstone AWS
- Bruce Cooper、プリンシパル SDE、 AWS
- Ravinder Thota、シニアアドバイザリコンサルタント、 AWS
- Sausan Yazji、シニアプラクティスマネージャー、 AWS
- Paul Duvall、ディレクター DevSecOps、 AWS
- ジェレミー・コンビナート、プリンシパル・クラウド・デリバリー・マネージャー、 AWS
- Sneh Shah、プリンシパルインフラストラクチャリード、 AWS
- Sasa Baskarada、 AWS クラウド導入フレームワーク、ワールドワイドリード AWS

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	—	2023 年 10 月 25 日

AWS 規範的ガイダンスの用語集

以下は、 AWS 規範的ガイダンスが提供する戦略、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に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: の移行 Microsoft Hyper-V 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。アクティブ/パッシブ移行よりも柔軟ですが、より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

行のグループに対して動作し、グループの単一の戻り値を計算する SQL 関数。集計関数の例としては、 SUMや などがありますMAX。

AI

人工知能を参照してください。

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。AIOps が移行戦略で AWS どのように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。

アトミック性、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 （ ABAC) ドキュメントの「 の AWS IAM」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 内の個別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドへの移行を成功させるための効率的で効果的な計画を立て AWS るのに役立つ、のガイドラインとベストプラクティスのフレームワークです。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを分類します。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や損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、不審な 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。エンディアンネス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 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 (ブルー) で実行し、新しいアプリケーションバージョンを他の環境 (グリーン) で実行します。この戦略は、影響を最小限に抑えながら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のインデックスを作成するウェブクローラーなど、有益または有益なボットもあります。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ているものがあります。

ボットネット

マルウェアに感染し、ボットのヘルダーまたはボットオペレーターとして知られる 1 人の当事者による管理下にあ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。

ブレークグラスアクセス

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 ガイダンスの「ブレー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。

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 「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低速かつ段階的なリリース。自信が持てたら、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。CDC 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またはレプリケートするなど、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、システムの回復力をテストします。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 を受信する前に、ローカルでデータを暗号化します。

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
- Foundation – クラウド導入を拡大するための基本的な投資 (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。これらが AWS 移行戦略とどのように関連しているかについては、「移行準備ガイド」を参照してください。

CMDB

「設定管理データベース」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、次のようなものがあります。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イメージ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、設定は想定状態から変化します。これにより、ワークロードが非準拠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。

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、移行のポートフォリオ検出および分析段階で CMDB 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 とリージョン、または組織全体に 1 つ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、より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

一元的な管理とガバナンスにより、分散型の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。信頼できる ID のみが、期待されるネットワークから信頼できるリソースにアクセスしていることを確実にします。詳細については、「 でのデータ境界の構築 AWS」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。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、クエリや分析によく使用されます。

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

ディープラーニング

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

defense-in-depth

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリ

ティの手法。この戦略を採用するときは 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。例えば、a defense-in-depth アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

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。

開発値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、もともとリーンな製造プラクティス向け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

離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けによく使用されます。

ディザスタ

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

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については、「コンテナと Amazon Word API Gateway を使用してレガシー Microsoft ASP.NET (ASMX) ウェブサービスを段階的にモダナイズする」を参照してください。

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

ベースライン設定からの逸脱の追跡。例えば、 AWS CloudFormation を使用してシステムリソースのドリフトを検出したり、 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

Virtual Private Cloud (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 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 Word) ドキュメントの「エンドポイントサービスの作成」を参照してください。 VPC

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

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、 AWS CAF セキュリティエピックには、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、サマリー統計を計算し、データの視覚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 種類の列が含まれます。

フェイルファスト

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

障害分離の境界

では AWS クラウド、アベイラビリティーゾーン、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で、障害の影響を制限し、ワークロードの耐障害性の向上に役立ちます。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) や積分勾配など、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。

数ショットプロンプト

同様のタスクを実行するように求める前に、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のではなく、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、可能な限り短い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)

一般化データとラベルなし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、自然言語での会話など、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、「基礎モデルとは」を参照してください。

G

生成 AI

大量のデータに対してトレーニングされ、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)

Amazon CloudFront では、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推奨されるアプローチです。

ゴールデンイメージ

システムまたはソフトウェアの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、デバイスの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 (OUs) 全体のリソース、ポリシー、コンプライアンスの管理に役立つ大まかなルール。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ている、ラベル付きの履歴データの一部。ホールドアウトデータを使用してモデル予測をホールドアウトデータと比較することで、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

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 (Microsoft SQL Server から Amazon RDS for SQL Server など) に移行します。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CPUおよびメモリ使用量が 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、または変更するのではなく、本番環境のワークロードに新しいインフラストラクチャをデプロイするモデル。イミュータブルなインフラストラクチャは、本質的にミュータブルなインフラストラクチャよりも一貫性、信頼性、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 (イングレス) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション外からのネットワーク接続を受け入れ、検査し、ルーティングする VPC。AWS 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VPCsを使用して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

2016 年に Klaus Schwab によって導入された用語は、接続、リアルタイムデータ、自動化、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、VPC (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。AWS 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VPCsを使用して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、AWS による機械学習モデルの解釈可能性」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。

大規模言語モデル (LLM)

大量のデータに基づい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 とは」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 R」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。エンディアンネス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ているソフトウェア。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 がインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、ユーザーがエンドポイントにアクセスしてデータを保存および取得する。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。これにより、生産現場の生産物が完成した製品に変換されます。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの採用を推進し、調整の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な 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 APIs を介して通信し、通常は小規模で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。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、軽量な APIs 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)

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。組織がクラウドへの移行のための強固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ちます。MAP には、従来の移行を体系的に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、運用、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、スプリントに取り組む 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 アプリケーション移行サービスを使用して Amazon EC2 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 (サーバーの適切なサイジング、価格設定、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」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

「組織の変更管理」を参照してください。

オフライン移行

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

OI

「 オペレーションの統合」を参照してください。

OLA

「運用レベルの契約」を参照してください。

オンライン移行

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

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 A 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。

運用レベルの契約 (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 IT グループが相互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のチェックリストと関連するベストプラクティス。詳細については、 AWS Well-Architected フレームワークの「運用準備状況レビュー (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運用、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 トランスフォーメーションの主要な焦点です。

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をログ AWS CloudTrail 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、変化の採用を加速し、移行に伴う問題に対処し、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や戦略に備え、移行するのに役立ちます。 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)

In CloudFront は、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 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、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)

In CloudFront は、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オプションです。OAI を使用すると、 CloudFront は Amazon S3 が認証できるプリンシパルを作成します。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。Word も参照してください。OACより詳細で強化されたアクセスコントロールを提供します。

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を処理する VPC。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VPCsを使用して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アクセス許可の上限を設定するための Word プリンシパルにアタッチされる IAM 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、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。通常は `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 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「ロールの用語と概念」の「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に入れた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

Amazon Route 53 が 1 つ以上の DNS 内のドメインとそのサブドメインの VPCs 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。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止する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、成長と成熟、辞退と削除まで、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、信頼性が高く適応性の高いコンピュータです。

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、事前対応を繰り返し調整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。

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。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 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内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

「責任者」、「説明責任者」、「相談先」、「通知先」 (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

「責任者」、「説明責任者」、「相談先」、「通知先」 (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

読み取り専用に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 R」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 R」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 とは分離され、独立しています。詳細については、AWS リージョン 「アカウントで使用できる 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 R」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 R」を参照してください。

プラットフォーム変更

「7 R」を参照してください。

再購入

「7 R」を参照してください。

回復性

中断に耐えたり、中断から回復したりするアプリケーションの機能。で障害耐性を計画する場合、高可用性とディザスタリカバリがよく考慮されます 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、情報提供 (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、マトリックスは RASCI マトリックスと呼ばれ、除外すると RACI マトリックスと呼ばれます。

レスポンシブコントロール

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 R」を参照してください。

廃止

「7 R」を参照してください。

取得拡張生成 (RAG)

レスポンスを生成する前に、LLM が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。例えば、RAG モデルは組織のナレッジベースや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

「目標復旧時点」を参照してください。

RTO

目標復旧時間を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdPs) が使用するオープンスタンダード。この機能により、フェデレーティッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーをAWS API で作成しなくても、 AWS Management Console にログインしたりIAM オペレーションを呼び出したりできます。SAML 2.0 ベースのフェデレーションの詳細については、Word 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収集」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、単一の文字列、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

開発プロセス全体を通じてセキュリティを考慮した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。

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、監視、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対応または修正するように設計された、事前定義され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の実装に役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動応答アクションの例としては、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

送信先にあるデータの、それ AWS のサービス を受け取る による暗号化。

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。SCPsは、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ガードレールを定義するか、制限を設定します。SCPs を許可リストまたは拒否リストとして使用して、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、ユーザー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)

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

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

split-and-seed モデル

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

SPOF

単一障害点を参照してください。

star スキーマ

トランザクションデータまたは測定データを保存するために 1 つの大きなファクトテーブルを使用し、データ属性を保存するために 1 つ以上の小さなディメンションテーブルを使用するデータベースの組織構造。この構造は、データウェアハウスまたはビジネスインテリジェンス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。このパターンを適用する方法の例については、「コンテナと Amazon ASP API Gateway を使用してレガシー Microsoft Word.NET (ASMX) ウェブサービスを段階的にモダナイズする」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ー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生産オペレーション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。

システムプロンプト

動作を指示するために、コンテキスト、指示、またはガイドラインを LLM に提供するための手法。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

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の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 ネットワーク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

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の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「 AWS Organizations を他の AWS のサービスで使用する 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 small 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 VPCs 間の接続。詳細については、Amazon VPC ドキュメントの「Word ピアリングとは」を参照してください。 VPC

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

「書き込み 1 回」、「読み取り多数」を参照してください。

WQF

AWS 「Word Workload Qualification Framework」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

タスクを実行するための指示を LLM に提供しますが、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

CPU とメモリの平均使用量が 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。