Amazon EKS インフラストラクチャのスケーリングによるコンピューティング、ワークロード、ネットワークパフォーマンスの最適化

AWS 規範ガイダンス

AWS 規範ガイダンス: Amazon EKS インフラストラクチャのスケーリングによるコンピューティング、ワークロード、ネットワークパフォーマンスの最適化

Table of Contents

Amazon EKS インフラストラクチャのスケーリングによるコンピューティング、ワークロード、ネットワークパフォーマンスの最適化

Aniket Dekate、Aniket Kurzadkar、Ishwar Chauthaiwale、Amazon Web Services (AWS)

2024 年 11 月 (ドキュメント履歴)

Amazon Elastic Kubernetes Service (Amazon EKS) はマネージド型 Kubernetes サービスです。Amazon EKS を使用すると、独自のコントロールプレーンをインストールして運用することなく、コンテナ化されたクラウド環境で Kubernetes ポッドを実行できます。コントロールプレーン AWS を管理することで、Amazon EKS は組織の運用管理を削減します。Amazon EKS を使用するその他の利点には、クラウド環境でのスケーリング、信頼性、セキュリティなどがあります。

このガイドは、組織が以下の分野にわたって Amazon EKS インフラストラクチャを最適化するのに役立つように設計されています。

- コンピューティングスケーリングは、動的な Kubernetes 環境でのアプリケーションパフォーマンスにとって重要なコンポーネントです。
  - 効率的なリソース割り当て – さまざまな需要に対応するために、計算されたリソースを動的に割り当てる手法について説明します。
  - 自動化ツール – コンピューティングスケーリングを自動化するツールとサービスの概要を取得し、手動による介入の必要性を減らします。
- ワークロードスケーリングは、アプリケーションがパフォーマンスを低下させることなくさまざまなワークロードを処理できるようにするのに役立ちます。
  - 水平ポッドオートスケーラー – HPA がリアルタイムメトリクスに基づくワークロードのスケーリングにどのように役立つかについて詳しく説明します。
  - Cluster Proportional Autoscaler – CPA がノードとレプリカ間の比例関係を自動的にスケーリングして維持し、クラスターサイズの変化に応じてワークロードをスケールアップまたはスケールダウンする方法について説明します。
  - イベント駆動型スケーリング – 特定のイベントまたはトリガーに応じてアプリケーションをスケーリングするための戦略を確認します。
- ネットワークスケーリングは、動的環境でサービスと効率的なデータフロー間のシームレスな通信を維持するのに役立ちます。

- Amazon VPC CNI プラグイン – VPC CNI プラグインが Amazon EKS クラスター内でスケーラブルなネットワーキングを有効にする方法について説明します。
- カスタムネットワーク - Amazon EKS クラスターの IP アドレス管理とネットワークトラフィックの分離を確認します。
- プレフィックスの委任 - 大規模でスケーラブルな Amazon EKS クラスターでの IP 管理の合理化の概要について説明します。
- Amazon VPC Lattice – VPC Lattice がシームレスなスケーリングのためにクロス VPC service-to-serviceネットワークを管理する方法の概要を説明します。

- コスト最適化は、企業がリソースが使用されている場所を確認し、部門やプロジェクトに経費を適切に割り当てるのに役立ちます。
  - 適切なサイジングリソース — ワークロードに合わせてクラウドリソースを適切にサイジングするための手法を検討します。
  - コストのモニタリングと制御 — クラウド費用を追跡および最適化するためのツールとベストプラクティスを確認します。

各セクションでは、信頼性が高く、効果的で、手頃な価格のクラウド環境を構築するために必要な特定の目標に焦点を当てています。

目的

このガイドは、お客様とお客様の組織が以下のビジネス目標を達成するのに役立ちます。

- リソース効率の向上 – リアルタイムの需要に基づいてコンピューティング、ワークロード、ネットワークリソースを動的にスケーリングすることで、最適なリソース使用率を実現します。

  この目標は、実際の使用パターンに応じてリソースをスケールアップまたはスケールダウンすることの重要性を強調しています。水平ポッドオートスケーラーや Amazon VPC CNI プラグインなどのツールを使用すると、組織は必要なリソースのみを使用し、無駄を最小限に抑えてパフォーマンスを最大化できます。

- アプリケーションパフォーマンスの向上 – 変動するワークロードやトラフィックパターンでも、アプリケーションの高いパフォーマンスと応答性を維持します。

  この目標は、アプリケーションがパフォーマンスを損なうことなくピークトラフィックと重いワークロードを処理できるようにする戦略に焦点を当てています。この目標を達成するには、イベント

駆動型のワークロードスケーリング、効率的なコンピューティング割り当て、スケーラブルなネットワークアーキテクチャなどの手法が重要です。

- シームレスなスケーラビリティ – インフラストラクチャコンポーネントのスムーズなスケーリングを可能にし、変化するビジネスニーズに簡単に成長と適応できるようにします。

  シームレスなスケーラビリティは、増加を予測したり、さまざまなトラフィックレベルを経験したりする組織にとって不可欠です。この目的は、スケーリングが自動、効率的、透過的になるように、コンピューティング、ワークロード、ネットワークリソース全体にスケーラブルなソリューションを実装することの重要性に対処します。

- コスト最適化 – パフォーマンスとスケーラビリティを維持または改善しながら、クラウドコストを最小限に抑えます。

  コスト最適化には、リソースの適切なサイズ設定、費用対効果の高いスケーリングソリューションの使用、支出のモニタリングなどのコストの削減が含まれます。目標は、コスト削減と、高いパフォーマンスとスケーラビリティの必要性のバランスを取ることです。

コンピューティングスケーリング

コンピューティングスケーリングは、動的な Kubernetes 環境でのアプリケーションパフォーマンスにとって重要なコンポーネントです。Kubernetes は、リアルタイムの需要に応じてコンピューティングリソース (CPU やメモリなど) を動的に調整することで、無駄を減らします。この機能は、過剰プロビジョニングや過少プロビジョニングを回避するのに役立ちます。これにより、運用コストも節約できます。Kubernetes は、ピーク時にインフラストラクチャを自動的にスケールアップし、オフピーク時にスケールダウンできるようにすることで、手動による介入を効果的に排除します。

Kubernetes の全体的なコンピューティングスケーリングはスケーリングプロセスを自動化するため、アプリケーションの柔軟性とスケーラビリティが向上し、耐障害性の動作が向上します。最終的に、Kubernetes の機能は運用上の優秀性と生産性を向上させます。

このセクションでは、次のタイプのコンピューティングスケーリングについて説明します。

- Cluster Autoscaler
- 過剰プロビジョニングによる Cluster Autoscaler
- Karpenter

クラスター AutoScaler

ポッドのニーズに応じて、Cluster Autoscaler ツールは、必要に応じてノードを追加するか、不要で十分に活用されていないノードを削除することで、サイズを自動的に変更します。

Cluster Autoscaler ツールは、需要が徐々に増加し、スケーリングのレイテンシーが大きな問題ではないワークロードのスケーリングソリューションとして考えてみましょう。

Cluster Autoscaler ツールには、次の主要な機能があります。

- スケーリング – 実際のリソース需要に応じてノードを動的にスケールアップおよびスケールダウンします。
- ポッドのスケジューリング — すべてのポッドが動作していて、機能するために必要なリソースがあることを確認し、リソースの不足を防ぎます。
- コスト効率 – 使用率の低いノードを排除することで、不要な運用コストを排除します。

過剰プロビジョニングによる Cluster Autoscaler

クラスターオートスケーラーは、クラスターオートスケーラーと同様に、ノードを効率的にデプロイし、ノードで優先度の低いポッドを実行することで時間を節約するという点で、過剰プロビジョニング機能を備えています。この手法では、需要の急増に応じてトラフィックがこれらのポッドにリダイレクトされるため、アプリケーションは中断することなく動作し続けることができます。

過剰プロビジョニングを備えた Cluster Autoscaler は、ワークロードが非常に大きく、レイテンシーが必要ではなく、スケーリングをすばやく行う必要がある場合に、ノードを簡単にデプロイして実行するために使用できるダミーポッドの機能を提供します。

オーバープロビジョニングの Cluster Autoscaler には、次の主要な機能があります。

- 応答性の向上 – 過剰な容量を常にアクセス可能にすることで、需要の急増に応じてクラスターをスケールアップする時間を短縮できます。
- リソース予約 – トラフィックの予期しない急増を管理すると、ダウンタイムをほとんど必要とせずに適切な管理を効果的に行うことができます。
- スムーズスケーリング – リソース割り当ての遅延を最小限に抑えることで、よりシームレスなスケーリングプロセスが容易になります。

Karpenter

Karpenter for Kubernetes は、オープンソース、パフォーマンス、カスタマイズ可能性の点で従来の Cluster Autoscaler ツールよりも優れています。Karpenter を使用すると、クラスターの需要をリアルタイムで処理するために必要なコンピューティングリソースのみを自動的に起動できます。Karpenter は、より効率的で応答性の高いスケーリングを実現するように設計されています。

迅速なスケーリングの決定が不可欠な、非常に可変または複雑なワークロードを持つアプリケーションは、Karpenter の使用から大きなメリットを得ます。と統合され AWS、デプロイとノード選択の最適化が改善されます。

Karpenter には以下の主要な機能があります。

- 動的プロビジョニング – Karpenter は目的に適したインスタンスとサイズを提供し、ポッドの特定の要件に基づいて新しいノードを動的にプロビジョニングします。
- 高度なスケジューリング – Karpenter は、賢明なポッド配置を使用して、GPU、CPU、メモリ、ストレージなどのリソースを可能な限り効果的に使用できるようにノードを配置します。

- クイックスケーリング – Karpenter はすばやくスケールでき、数秒で頻繁に反応します。この応答性は、突然のトラフィックのパターンや、ワークロードが即時スケーリングを必要とする場合に役立ちます。
- コスト効率 – 最も効果的なインスタンスを慎重に選択することで、運用コストを削減し、オンデマンドインスタンス AWS、スポットインスタンス、リザーブドインスタンスなど、 が提供する追加のコスト削減代替手段を活用できます。

ワークロードのスケーリング

Kubernetes でのワークロードスケーリングは、動的環境でアプリケーションのパフォーマンスとリソース効率を維持するために不可欠です。スケーリングは、アプリケーションがパフォーマンスを低下させることなくさまざまなワークロードを処理できるようにするのに役立ちます。Kubernetes では、リアルタイムのメトリクスに基づいてリソースを自動的にスケールアップまたはスケールダウンできるため、組織はトラフィックの変化に迅速に対応できます。この伸縮性により、ユーザーエクスペリエンスが向上するだけでなく、リソース使用率も最適化され、使用率の低いリソースや過剰にプロビジョニングされたリソースに関連するコストを最小限に抑えることができます。

さらに、効果的なワークロードスケーリングは高可用性をサポートし、需要のピーク時でもアプリケーションの応答性を維持します。Kubernetes でのワークロードスケーリングにより、組織は現在のニーズに合わせて容量を動的に調整することで、クラウドリソースをより有効に活用できます。

このセクションでは、次のタイプのワークロードスケーリングについて説明します。

- 水平ポッドオートスケーラー
- クラスター比例オートスケーラー
- Kubernetes ベースの Event Driven Autoscaler

Horizontal Pod Autoscaler

Horizontal Pod Autoscaler (HPA) は、観測された CPU 使用率やその他の選択されたメトリクスに基づいて、デプロイ、レプリケーションコントローラー、またはステートフルセット内のポッドレプリカの数を自動的に調整する Kubernetes 機能です。HPA を使用すると、アプリケーションは手動で介入することなく、変動するトラフィックとワークロードレベルを管理できます。HPA は、利用可能なリソースを効果的に活用しながら、最適なパフォーマンスを維持する手段を提供します。

ウェブアプリケーション、マイクロサービス、APIs など、ユーザーの需要が時間の経過とともに大幅に変動する可能性がある状況では、HPA が特に役立ちます。

Horizontal Pod Autoscaler には、次の主要な機能があります。

- 自動スケーリング – HPA は、リアルタイムメトリクスに応じてポッドレプリカの数を自動的に増減するため、アプリケーションはユーザーの需要に合わせてスケーリングできます。

- メトリクスベースの決定 – デフォルトでは、HPA は CPU 使用率に基づいてスケーリングされます。ただし、メモリ使用量やアプリケーション固有のメトリクスなどのカスタムメトリクスを使用することもできます。これにより、よりカスタマイズされたスケーリング戦略が可能になります。
- 設定可能なパラメータ – 最小レプリカ数と最大レプリカ数、および希望する使用率を選択できます。これにより、スケーリングの重要度に対する権限が付与されます。
- Kubernetes との統合 – リソースをモニタリングおよび変更するために、HPA は Metrics Server、Kubernetes API、カスタムメトリクスアダプターなど、Kubernetes エコシステムの他の要素と連携して動作します。
- リソース使用率の向上 – HPA は、ポッドの数を動的に変更することで、リソースを効果的に使用し、コストを削減してパフォーマンスを向上させるのに役立ちます。

クラスタープロポーショナルオートスケーラー

Cluster Proportional Autoscaler (CPA) は、使用可能なノードの数に基づいてクラスター内のポッドレプリカの数を自動的に調整するように設計された Kubernetes コンポーネントです。リソース使用率メトリクス (CPU やメモリなど) に基づいてスケールする従来のオートスケーラーとは異なり、CPA はクラスター自体のサイズに比例してワークロードをスケールします。

このアプローチは、CoreDNS やその他のインフラストラクチャサービスなど、クラスターサイズに対して特定のレベルの冗長性や可用性を維持する必要があるアプリケーションに特に役立ちます。CPA の主なユースケースには、次のようなものがあります。

- オーバープロビジョニング
- コアプラットフォームサービスのスケールアウト
- CPA はメトリクスサーバーや Prometheus Adapter を必要としないため、ワークロードをスケールアウトする

スケーリングプロセスを自動化することで、CPA はバランスの取れたワークロード分散を維持し、リソース効率を高め、ユーザーの需要を満たすようにアプリケーションを適切にプロビジョニングすることを支援します。

Cluster Proportional Autoscaler には、次の主要な機能があります。

- ノードベースのスケーリング – CPA は、スケジュールできるクラスターノードの数に応じてレプリカをスケーリングし、アプリケーションがクラスターのサイズに比例して拡張または縮小できるようにします。

- 比例調整 — アプリケーションがクラスターサイズの変化に応じてスケールできるように、オートスケーラーはノード数とレプリカ数の間に比例関係を確立します。この関係は、ワークロードに必要な数のレプリカを計算するために使用されます。
- Kubernetes コンポーネントとの統合 – CPA は Horizontal Pod Autoscaler (HPA) などの標準 Kubernetes コンポーネントと連携しますが、リソース使用率メトリクスではなくノード数に特に重点を置いています。この統合により、より包括的なスケーリング戦略が可能になります。
- Golang API クライアント – ノードの数とその使用可能なコアをモニタリングするために、CPA はポッド内で実行され、Kubernetes API サーバーと通信する Golang API クライアントを使用します。
- 設定可能なパラメータ – を使用すると`ConfigMap`、ユーザーは CPA が動作を変更するために使用するしきい値とスケーリングパラメータを設定し、意図したスケーリングプランに従っていることを確認できます。

Kubernetes ベースの Event-Driven Autoscaler

Kubernetes ベースの Event Driven Autoscaler (KEDA) は、処理する必要があるイベントの数に基づいて Kubernetes ワークロードをスケールできるようにするオープンソースプロジェクトです。KEDA は、さまざまなワークロード、特にイベント駆動型ワークロードに動的に応答できるようにすることで、アプリケーションのスケーラビリティを強化します。

イベントに基づいてスケーリングプロセスを自動化することで、KEDA は組織がリソース使用率を最適化し、アプリケーションのパフォーマンスを向上させ、オーバープロビジョニングに関連するコストを削減するのに役立ちます。このアプローチは、マイクロサービス、サーバーレス関数、リアルタイムデータ処理システムなど、さまざまなトラフィックパターンを経験するアプリケーションに特に役立ちます。

KEDA には以下の主要な機能があります。

- イベント駆動型スケーリング – KEDA では、メッセージキュー、HTTP リクエスト、カスタムメトリクスなどの外部イベントソースに基づいてスケーリングルールを定義できます。この機能は、アプリケーションがリアルタイムの需要に応じてスケールインするのに役立ちます。
- 軽量コンポーネント – KEDA は、既存の Kubernetes クラスターに簡単に統合するために多くのセットアップやオーバーヘッドを必要としない、単一用途の軽量コンポーネントです。
- Kubernetes との統合 – KEDA は、Horizontal Pod Autoscaler (HPA) などの Kubernetes ネイティブコンポーネントの機能を拡張します。KEDA は、これらのコンポーネントにイベント駆動型のスケーリング機能を追加し、置き換えるのではなく強化します。

- 複数のイベントソースのサポート – KEDA は、RabbitMQ、Apache Kafka などの一般的なメッセージングプラットフォームなど、さまざまなイベントソースと互換性があります。この適応性により、独自のイベント駆動型アーキテクチャに合わせてスケーリングをカスタマイズできます。
- カスタムスケーラー – カスタムスケーラーを使用すると、KEDA が特定のビジネスロジックまたは要件に応じてスケーリングアクションを開始するために使用できる特定のメトリクスを指定できます。
- 宣言型設定 – Kubernetes の原則に従って、Kubernetes カスタムリソースを使用してスケーリングの実行方法を定義することで、KEDA を使用してスケーリング動作を宣言的に記述できます。

ネットワークスケーリング

Kubernetes でのネットワークスケーリングは、サービス間のシームレスな通信を維持し、動的環境で効率的なデータフローをサポートするために不可欠です。ネットワークインフラストラクチャをスケーリングすると、ボトルネックやレイテンシーの問題が発生することなく、クラスターがさまざまなレベルのトラフィックを処理できるようになります。Kubernetes には、ネットワークリソースをスケールするためのツールとメカニズムが用意されているため、組織はトラフィックパターンの変化に応じて最適なパフォーマンスを維持できます。

このネットワークスケーリングの伸縮性により、高速で信頼性の高い接続を確保することで、全体的なユーザーエクスペリエンスが向上します。また、ネットワークスケーリングはネットワークリソースの使用を最適化し、使用率の低さや過負荷のネットワークコンポーネントに関連するコストを削減します。

さらに、高可用性と耐障害性をサポートするには、効果的なネットワークスケーリングが不可欠です。ネットワーク容量とルーティングを動的に調整することで、組織は、需要のピーク時や予期しないトラフィックの急増時でも、サービスへのアクセスと応答性を維持できます。このアプローチにより、クラウドネットワークリソースの使用率が向上し、インフラストラクチャが常に現在の要件と一致します。

このセクションでは、以下のタイプのネットワークスケーリングについて説明します。

- Amazon VPC CNI plugin for Kubernetes
- カスタムネットワーク
- プレフィックスの委任
- Amazon VPC Lattice

Amazon VPC CNI Kubernetes用プラグイン

Amazon VPC Container Network Interface (CNI) plugin for Kubernetes は、Amazon EKS の重要なコンポーネントです。VPC CNI プラグインは、Kubernetes ポッドを Amazon VPC と統合することで、高度なネットワーク機能を提供します。このプラグインを使用すると、各ポッドに Virtual Private Cloud (VPC) から一意の IP アドレスが割り当てられるため、ネットワークの分離とパフォーマンスが向上します。クラスターが増加し、ネットワーク需要が変動するにつれて、Amazon VPC CNI プラグインは効率的でスケーラブルなネットワークオペレーションを確保する上で重要な役割を果たします。

プラグインは、VPC 内の IP アドレスの割り当てとルーティングを自動的に管理し、ネットワーク管理を簡素化し、IP 競合のリスクを軽減します。プレフィックス委任などの機能をサポートしているため、より柔軟な IP 管理が可能になります。

VPC CNI プラグインは、組織がネットワークパフォーマンスを最適化し、セキュリティを強化し、IP が枯渇するリスクを軽減するのに役立ちます。これらの機能は、マイクロサービスアーキテクチャ、高密度ワークロード、マルチテナントアプリケーションなど、ネットワーク需要が変動する大規模で動的な環境で特に役立ちます。

Amazon VPC CNI プラグインには、次の主要な機能があります。

- 拡張ネットワーキング – VPC CNI プラグインを使用すると、各ポッドが VPC から直接独自の IP アドレスを受信できるため、強力な分離とネットワークパフォーマンスが得られます。このアプローチは、高いネットワークスループットと低レイテンシーを必要とするワークロードに不可欠です。
- プレフィックスの委任 – 大規模なクラスターの IP アドレスの枯渇の問題を克服するために、プレフィックスの委任はより大きな IPs ブロックをノードに動的に割り当て、それをポッド用に分割します。このアプローチにより、効率的な IP 使用率が確保され、ネットワークスケーリングが簡素化されます。
- カスタムネットワーク – ユーザーはポッドのカスタムネットワークインターフェイス (ENIs) を設定できます。これにより、ポッドトラフィックを複数のインターフェイスに分散し、ネットワークの輻輳を減らし、スケーラビリティを向上させることができます。
- IPv6 のサポート – Amazon EKS クラスターで IPv6 を有効にすることで、ユーザーは使用可能な IP アドレス空間を大幅に拡張できるため、IPv4 の制限なしに大規模な分散アプリケーションのスケーリングが容易になります。
- Kubernetes との統合 – VPC CNI プラグインは Kubernetes ネットワークコンポーネントとシームレスに連携し、IPs がポッド、サービス、外部エンドポイント間で効率的に管理されるようにし、ポッドのセキュリティグループなどの高度な機能をサポートします。

カスタムネットワーキング

Amazon EKS のカスタムネットワーキングにより、ポッドへの特定のネットワークインターフェイスの割り当てが可能になり、IP アドレス管理とネットワークトラフィックの制御が強化されます。このアプローチは、IP アドレスの枯渇が懸念される場合や、セキュリティ、コンプライアンス、パフォーマンス上の理由からネットワークトラフィックを分離する必要がある場合に特に役立ちます。カスタムネットワーキングは、組織が IP アドレス空間を効率的に管理し、トラフィックを分離し、スケーラブルなネットワークパフォーマンスを確保するのに役立ちます。

カスタムネットワーキングを使用すると、管理者はネットワークリソースをより効率的に管理できます。管理者はカスタムネットワークを使用して、ポッドに必要なネットワーク分離を確保し、IP アドレスの制限に遭遇することなくクラスターをスケーリングできます。

カスタムネットワーキングには、次の主要な機能があります。

- IP 管理の強化 – カスタムネットワークでは、特定のネットワークインターフェイス (ENIs) をポッドに割り当てることができるため、複数の ENIs にポッドトラフィックを分散することで IP アドレスの枯渇を管理できます。この機能は、高密度ワークロードを持つクラスターでは特に重要です。
- トラフィックの分離 – カスタムネットワークインターフェイスを使用すると、アプリケーションタイプやセキュリティ要件などの特定の基準に基づいてポッドトラフィックを分離できます。このアプローチにより、クラスター内外のトラフィックフローをより詳細に制御できます。
- IPv6 のサポート – Amazon EKS のカスタムネットワーキングは IPv6 もサポートし、IPv4 アドレスの制限に対するソリューションを提供します。大規模なデプロイでも、ネットワークは IP アドレスの競合なしで効率的にスケールできます。
- スケーラビリティと柔軟性 – クラスターがスケールすると、カスタムネットワーキングによりネットワークインターフェイスの動的な管理が可能になります。新しいポッドには、手動操作なしで適切なネットワークリソースが割り当てられます。このアプローチは、変化するワークロードに適応できる柔軟でスケーラブルなネットワーク環境を維持するのに役立ちます。

プレフィックスの委任

Kubernetes、特に Amazon EKS 内のプレフィックスの委任は、クラスターのスケーリングに合わせて IP アドレス管理を合理化および最適化するように設計されています。ノードにより大きな IP アドレス (プレフィックス) のブロックを動的に割り当てることで、プレフィックスの委任により IP 枯渇のリスクが軽減され、IP スペースの管理が簡素化されます。

このアプローチにより、ネットワーク効率が向上し、断片化が最小限に抑えられ、IP 範囲を手動で調整することなくクラスターをスムーズにスケールできます。プレフィックスの委任は、大規模なデプロイ、高密度ワークロード、およびネットワークのパフォーマンスとスケーラビリティを維持するために柔軟で動的な IP 管理が重要な環境にとって特に重要です。

プレフィックスの委任には、次の主要な機能があります。

- 効率的な IP アドレス管理 – プレフィックスの委任により、IP 範囲の動的な割り当てが可能になり、IP が枯渇するリスクを軽減し、使用可能な IP スペースを効率的に使用できます。

- ネットワーク管理の簡素化 – ノードが独自の IP 割り当てを処理できるようにすることで、プレフィックスの委任によりネットワークの断片化が最小限に抑えられ、ルーティングプロセスが簡素化されるため、必要に応じてクラスターを簡単にスケーリングできます。
- 大規模なデプロイのサポート – 高密度ワークロードを持つ大規模なクラスターでは、プレフィックスの委任により、新しいノードが手動で IP 範囲を調整することなくクラスターに参加できるようにすることで、シームレスなスケーリングが可能になります。

Amazon VPC Lattice

Amazon VPC Lattice は、特にマイクロサービスアーキテクチャにおいて、VPCs 間の効率的で安全なservice-to-service通信を可能にします。VPC Lattice は、きめ細かなアプリケーション認証のための AWS Identity and Access Management (IAM) 統合に加えて、セキュリティグループやネットワークアクセスコントロールリスト (ネットワーク ACLs) などのセキュリティ対策を使用します。VPC Lattice の中心にある Layer-7 プロキシサービスは、接続、ロードバランシング、認証、認可、オブザーバビリティ、トラフィック管理、サービス検出を提供します。

ネットワークとセキュリティの設定を簡素化することで、VPC Lattice は、組織がトラフィック管理を最適化し、アプリケーションのパフォーマンスを向上させ、複数の VPCsがってシームレスにスケーリングするのに役立ちます AWS リージョン。これは、マイクロサービス、クロスリージョンデプロイ、複雑なクラウドネイティブ環境など、一貫性のある信頼性の高いネットワークを必要とする分散アプリケーションに特に役立ちます。

Amazon VPC Lattice には、次の主要な機能があります。

- Service-to-serviceネットワーク — VPC Lattice は、マイクロサービスアーキテクチャ内のサービス間のネットワークとセキュリティの設定を簡素化します。通信を管理するための統合プラットフォームを提供するため、サービスは高いパフォーマンスとセキュリティを維持しながら個別にスケーリングできます。
- クロス VPC ネットワーク – VPC Lattice は、複数の VPCs またはリージョン間のトラフィックを管理するために不可欠です。これは、物理的な場所に関係なく、サービスがシームレスに通信できるようにする一貫したネットワークフレームワークを提供します。この機能は、複数の VPCs または地理的リージョンにまたがる大規模なアプリケーションでは特に重要です。
- セキュリティ管理の強化 – セキュリティポリシーをネットワークレイヤーに直接統合することで、VPC Lattice は安全で効率的なservice-to-service通信をサポートします。この機能により、分散環境全体のセキュリティ管理の複雑さが軽減され、スケーリングが容易になり、運用上のオーバーヘッドが軽減されます。

- トラフィック管理の簡素化 – VPC Lattice は、ルーティング、ロードバランシング、フェイルオーバーメカニズムなどの高度なトラフィック管理機能を提供します。これらの機能により、トラフィックは サービス間で効率的に分散され、ネットワークパフォーマンスが最適化され、アプリケーションのスケーラビリティが向上します。

コスト最適化

効果的なリソース制御をサポートするために、Kubernetes のコスト最小化は、このコンテナオーケストレーションテクノロジーを使用する企業にとって不可欠です。ポッドやノードなどの複数のコンポーネントを含む複雑なため、Kubernetes 設定の支出を適切に追跡することは困難です。コスト最適化手法を適用することで、企業はリソースがどこに費やされているかを確認し、部門やプロジェクトに経費を適切に割り当てることができます。

動的スケーリングには利点がありますが、適切に管理しないと、予期しない費用が発生する可能性があります。効率的なコスト管理は、本当に必要な場合にのみリソースを割り当て、予期しない支出の急増を防ぐのに役立ちます。

このセクションでは、コスト最適化の以下のアプローチについて説明します。

- Kubecost
- ゴールドロック
- AWS Fargate
- スポットインスタンス
- リザーブドインスタンス
- AWS Graviton インスタンス

Kubecost

Kubecost は、企業がクラウドインフラストラクチャへの支出を追跡、制御、最大化するのに役立つコスト管理ソリューションです。これは、Kubernetes クラスター専用に作成されています。Kubecost は、リソース使用率とリアルタイムのコスト認識に関するインサイトを提供し、クラウドリソースが使用されている場所と量をよりよく理解できるようにします。これらのインサイトにより、インフラストラクチャの支出を最適化し、リソース効率を向上させ、クラウド投資についてより多くの情報に基づいた意思決定を行うことができます。

Kubecost には以下の主要な機能があります。

- コスト配分 – Kubecost は、ワークロード、サービス、名前空間、ラベルなど、Kubernetes リソースのコスト配分を徹底的に提供します。この機能は、チームが環境、プロジェクト、チームごとにコストをモニタリングするのに役立ちます。

- リアルタイムのコストモニタリング – クラウドコストをリアルタイムでモニタリングできるため、組織は支出パターンをすぐに把握し、予期しないコスト超過を防ぐことができます。
- 最適化に関する推奨事項 – Kubecost は、アイドル状態のリソースの削減、ワークロードの適切なサイズ設定、ストレージ費用の最大化など、リソース使用率を最小限に抑えるための実用的な提案を提供します。
- 予算とアラート – Kubecost ユーザーは予算を作成し、支出が事前定義された基準に近づいたとき、または超えたときにリマインダーを受け取ることができます。この機能は、チームが財務上の制約に従うのに役立ちます。

ゴールドロック

Goldilocks は、ユーザーが Kubernetes ワークロードのリソースリクエストと制限を最適化できるように設計された Kubernetes ユーティリティです。Kubernetes クラスターで実行されているコンテナの CPU リソースとメモリリソースを設定する方法に関する推奨事項を提供します。これらのレコメンデーションは、アプリケーションを無駄なく効率的に実行するために、適切な数のリソースがあることを確認するのに役立ちます。この最適化により、コスト削減、パフォーマンスの向上、Kubernetes クラスターのより効率的な使用が可能になります。

Goldilocks には、次の主要な機能があります。

- リソースのレコメンデーション – Goldenilocks は、Kubernetes ワークロードの過去の CPU とメモリの消費量の統計を分析することで、リソースリクエストと制限の理想的な設定を決定します。これにより、プロビジョニング不足や過剰なプロビジョニングを回避しやすくなります。これにより、パフォーマンスの問題やリソースの無駄が発生する可能性があります。
- VPA 統合 – Goldilocks は、Kubernetes Vertical Pod Autoscaler (VPA) を活用してデータを収集し、レコメンデーションを提供します。これは「推奨モード」で実行されます。つまり、実際にリソース設定を変更するのではなく、それらの設定についてガイダンスを提供します。
- 名前空間ベースの分析 – Goldenilocks では、特定の名前空間を分析対象にすることで、最適化およびモニタリングされるワークロードを細かく調整できます。
- ビジュアルダッシュボード – ウェブベースのダッシュボードには、提案されたリソースリクエストと制限が視覚的に表示されるため、データを理解し、アクションを実行することが容易になります。
- 非侵入オペレーション – 推奨モードで動作するため、Goldilocks はクラスターの設定を変更しません。必要に応じて、レコメンデーションを確認した後に、推奨リソース設定を手動で適用できます。

AWS Fargate

Amazon EKS のコンテキストでは、 https://docs.aws.amazon.com/eks/latest/userguide/fargate.htmlAWS Fargate を使用すると、基盤となる Amazon EC2 インスタンスを管理せずに Kubernetes ポッドを実行できます。これは、インフラストラクチャを気にすることなく、コンテナ化されたアプリケーションのデプロイとスケーリングに集中できるサーバーレスコンピューティングエンジンです。

AWS Fargate には、次の主要な機能があります。

- インフラストラクチャ管理なし – Fargate は、Amazon EC2 インスタンスまたは Kubernetes ノードをプロビジョニング、管理、またはスケーリングする必要をなくします。 は、パッチ適用やスケーリングを含むすべてのインフラストラクチャ管理 AWS を処理します。
- ポッドレベルの分離 – Amazon EC2 に基づくワーカーノードとは異なり、Fargate はタスクまたはポッドレベルの分離を提供します。各ポッドは独自の独立したコンピューティング環境で実行されるため、セキュリティとパフォーマンスが向上します。
- 自動スケーリング – Fargate は、需要に基づいて Kubernetes ポッドを自動的にスケーリングします。スケーリングポリシーやノードプールを管理する必要はありません。
- 1 秒あたりの請求 – 各ポッドが実行する正確な期間に消費した vCPU とメモリリソースに対してのみ料金が発生します。これは、特定のワークロードに対して費用対効果の高いオプションです。
- オーバーヘッドの削減 – EC2 インスタンスを管理する必要がなくなるため、Fargate ではインフラストラクチャオペレーションではなくアプリケーションの構築と管理に集中できます。

スポットインスタンス

スポットインスタンスは、オンデマンドインスタンスの料金を大幅に節約でき、Amazon EKS クラスターで Amazon EC2 ワーカーノードを実行するための手頃なオプションです。ただし、 は、オンデマンドAWS インスタンスの容量が必要な場合にスポットインスタンスを中断できます。 は、容量が必要な場合に 2 分間の通知でスポットインスタンスを再利用 AWS できるため、重要でステートフルなワークロードの信頼性が低下します。

コストに敏感で中断に耐えられるワークロードの場合、Amazon EKS のスポットインスタンスが適しています。Kubernetes クラスターでスポットインスタンスとオンデマンドインスタンスを組み合わせて使用すると、重要なワークロードの可用性を犠牲にすることなくコストを削減できます。

スポットインスタンスには、次の主要な機能があります。

- コスト削減 – スポットインスタンスはオンデマンドインスタンスの料金よりも安価になる可能性があるため、コストの影響を受けやすいワークロードに最適です。
- 耐障害性のあるワークロードに最適 – バッチ処理、CI/CD ジョブ、機械学習、大規模なデータ処理などのステートレスで耐障害性のあるワークロードに最適です。大規模なデータ処理では、大規模な中断なしでインスタンスを置き換えることができます。
- Auto Scaling グループ統合 – Amazon EKS はスポットインスタンスを Kubernetes クラスターオートスケーラーと統合します。これにより、中断されたスポットインスタンスノードを他の利用可能なスポットインスタンスまたはオンデマンドインスタンスに自動的に置き換えることができます。

予約インスタンス

Amazon EKS では、リザーブドインスタンスは、Kubernetes ワークロードを実行する Amazon EC2 ワーカーノードの料金モデルです。リザーブドインスタンスを使用すると、オンデマンドインスタンスの料金と比較してコスト削減と引き換えに、特定のインスタンスタイプを 1 年間または 3 年間使用することをコミットできます。Amazon EKS でインスタンスを予約することは、Amazon EC2 ワーカーノードで一貫した長期ワークロードを実行するための手頃な方法です。

リザーブドインスタンスはAmazon EC2 で一般的に使用されます。ただし、ワークロードが長期的で予測可能な使用を必要とする場合、Amazon EKS クラスター (EC2 インスタンス) のワーカーノードにもこのコスト削減モデルのメリットがあります。

高可用性と一貫したパフォーマンスを必要とする本稼働サービス、データベース、およびその他のステートフルアプリケーションは、リザーブドインスタンスに適した安定したワークロードの例です。

リザーブドインスタンスには、次の主要な機能があります。

- コスト削減 – リザーブドインスタンスは、期間の長さ (1 年または 3 年) と支払いプラン (全額前払い、一部前払い、前払いなし) に応じて、オンデマンドインスタンスと比較して節約を提供します。
- 長期コミットメント – 特定のインスタンスタイプ、サイズ、および について、1 年または 3 年の期間をコミットします AWS リージョン。これは、安定していて、時間の経過とともに継続的に実行されるワークロードに最適です。
- 予測可能な料金 – リザーブドインスタンスは特定の期間にコミットされているため、予測可能な月額コストまたは前払いコストを提供するため、長期的なワークロードの予算編成が容易になります。

- インスタンスの柔軟性 – コンバーティブルリザーブドインスタンスでは、予約期間中にインスタンスタイプ、ファミリー、またはサイズを変更できます。コンバーティブルリザーブドインスタンスは、スタンダードリザーブドインスタンスよりも柔軟性が高く、変更はできません。
- 保証された容量 – リザーブドインスタンスは、予約が行われたアベイラビリティーゾーンで容量を利用できるようにします。これは、一貫したコンピューティング能力を必要とする重要なワークロードにとって不可欠です。
- 中断リスクなし – スポットインスタンスとは異なり、リザーブドインスタンスは によって中断されることはありません AWS。これにより、保証された稼働時間を必要とするミッションクリティカルなワークロードの実行に最適です。

AWS Graviton インスタンス

AWS Graviton は、クラウドワークロードのパフォーマンスとコスト効率を向上させる AWS ためにによって設計された ARM ベースのプロセッサのファミリーです。Amazon EKS のコンテキストでは、Graviton インスタンスをワーカーノードとして使用して Kubernetes ワークロードを実行し、大幅なパフォーマンスの向上とコスト削減を実現できます。

Graviton インスタンスは、x86 インスタンスよりも高い価格パフォーマンス比を提供するため、クラウドネイティブおよびコンピューティング集約型のアプリケーションに最適です。ただし、Graviton インスタンスの採用を検討する場合は、ARM の互換性を考慮してください。

AWS Graviton インスタンスには、次の主要な機能があります。

- ARM ベースのアーキテクチャ – AWS Graviton プロセッサは ARM アーキテクチャ上に構築されています。従来の x86 アーキテクチャとは異なりますが、多くのワークロードで非常に効率的です。
- コスト効率 – Graviton に基づく Amazon EC2 インスタンスは、通常、x86 ベースの EC2 インスタンスよりも優れた価格パフォーマンスを提供します。これにより、Amazon EKS を実行する Kubernetes クラスターにとって魅力的なオプションになります。
- パフォーマンス – Graviton の第 2 世代である Graviton2 プロセッサは、コンピューティングパフォーマンス、メモリスループット、エネルギー効率の点で大幅に改善されています。 AWS CPU を大量に消費し、メモリを大量に消費するワークロードに最適です。
- 多様なインスタンスタイプ – Graviton インスタンスには、t4g、m7g、c7g、r7g などのさまざまなファミリーがあり、汎用からコンピューティング最適化、メモリ最適化、バースト可能なワークロードまで、さまざまなユースケースに対応しています。

- Amazon EKS ノードグループ – Amazon EKS によって管理されるノードグループまたはセルフマネージド型ノードグループを設定して、Graviton ベースのインスタンスを含めることができます。このアプローチでは、x86 ベースのインスタンスとともに同じ Kubernetes クラスターで ARM アーキテクチャに最適化されたワークロードを実行できます。

次のステップ

このガイドでは、コンピューティングスケーリング、ワークロードスケーリング、ネットワークスケーリング、コスト最適化に関して Amazon EKS を最適化するのに役立つ情報を提供します。これらの概念を理解して適用することで、組織は動的なニーズを満たす非常に効率的でスケーラブル、費用対効果の高いクラウド環境を実現できます。

コンピューティングとワークロードのスケーリングを効果的に実装することで、リソースが効率的に使用され、アプリケーションがピーク時でも高いパフォーマンスを維持できます。カスタムネットワーキングやプレフィックス委任などのネットワークスケーリング手法を採用することで、ネットワークリソースの管理とシームレスなスケーラビリティがサポートされます。コスト最適化を強調することで、組織はパフォーマンスと財務効率のバランスを取ることができます。

このガイダンスをクラウド戦略に統合することで、インフラストラクチャのパフォーマンスとスケーラビリティを向上させ、コスト削減を促進することができます。この包括的なアプローチにより、組織の成長をサポートし、絶えず変化するビジネス需要に適応する堅牢なクラウド環境を構築できます。

リソース

AWS ブログ

- スポットインスタンスを使用した EKS のコスト最適化と耐障害性の構築
- Amazon EKS AWS を使用してコストと耐障害性を最適化するために Graviton と x86 CPUs を混在させる

AWS ドキュメント

- アマゾン VPC CNI
- Amazon Elastic Kubernetes Service (AWS ホワイトペーパー: でのデプロイオプションの概要 AWS)
- Amazon EKS ベストプラクティスガイド
- Karpenter
- Kubecost の詳細
- でコンピューティング管理を簡素化する AWS Fargate

その他のリソース

- クラスターの自動スケーリング (Kubernetes ドキュメント)
- Goldilocks: An Open Source Tool for Recommending Resource Requests (Fairwinds ブログ)
- 水平ポッドの自動スケーリング (Kubernetes ドキュメント)
- Kubecost (Kubecost ドキュメント)
- Kubernetes イベント駆動型自動スケーリング (KEDA ドキュメント)

ドキュメント履歴

次の表は、コンピューティング、ワークロード、ネットワークパフォーマンスを最適化するための Amazon EKS インフラストラクチャのスケーリングという、このガイドの重要な変更点を示しています。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
初版発行	—	2024 年 11 月 11 日

AWS 規範ガイダンスの用語集

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

数字

7 Rs

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

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

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

A

A2A (Agent-to-Agent)

タスクの委任と状態転送をサポートするagent-to-agentコラボレーションのためのステートフルプロトコル。

ABAC

「属性ベースのアクセス制御」をご覧ください。

抽象化されたサービス

「マネージドユーザー」をご覧ください。

ACID

「原子性、一貫性、分離性、耐久性 (ACID)」をご覧ください。

アクティブ/アクティブ移行

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

アクティブ/パッシブ移行

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

[エージェント]

目標を達成するためのツールを使用して、自律的に推論、計画、アクションを実行できる AI システム。

エージェントオペレーション

AI エージェントを本番環境で大規模に構築、テスト、デプロイ、実行するための運用プラクティス。

集計関数

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

AI

「人工知能」をご覧ください。

AIOps

「AI オペレーション」をご覧ください。

匿名化

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

アンチパターン

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

アプリケーション制御

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

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

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

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

アベイラビリティーゾーン (AZ)

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)

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 るための、 のガイドラインとベストプラクティスのフレームワークです。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを整理しています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、 AWS CAF は、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ため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

個人や組織に混乱や損害を与えることを目的としたボット。

BCP

「ビジネス継続性計画 (BCP)」をご覧ください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 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

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

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティややり取りをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクローラーなど、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ており、ボットハーダーまたはボットオペレーターと呼ばれる単一の当事者によって制御されているボットのネットワーク。ボットネットは、ボットとその影響力を拡大する仕組みとして、非常によく知られています。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといいます。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、「ブランチの概要」(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、 AWS 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、AWSでのコンテナ化されたマイクロサービスの実行ホワイトペーパーの「ビジネス機能を中心に組織化」セクションを参照してください。

ビジネス継続性計画 (BCP)

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

C

CAF

「AWS クラウド導入フレームワーク」を参照してください

カナリアデプロイ

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

CCoE

「Cloud Center of Excellence」を参照してください。

CDC

「変更データキャプチャ」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

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

CI/CD

「継続的インテグレーションと継続的デリバリー」を参照してください。

分類

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

シチズンデベロッパー

専門的な技術スキルを持たないノーコード/ローコードプラットフォームを使用して AI アプリケーションを作成するビジネスユーザー。

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が、 AWS クラウドへの移行時に通常実行する 4 つの段階。

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

これらのステージは、 AWS クラウド エンタープライズ戦略ブログのブログ記事「クラウドファーストへのジャーニー」と「導入のステージ」で Stephen Orban によって定義されました。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

「構成管理データベース (CMDB)」を参照してください。

コードリポジトリ

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

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

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

設定ドリフト

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

構成管理データベース (CMDB)

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

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンの単一のエンティティとしてデプロイ

することも、組織全体にデプロイすることもできます。詳細については、 AWS Config ドキュメントの「コンフォーマンスパック」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

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

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

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

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された 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。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

深層学習

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

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリティの手法。この戦略を採用するときは AWS、 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で使用できるサービス」を参照してください。

トラブルシューティング

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

開発環境

「環境」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

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

ディザスタ

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)

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

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み) で紹介されています (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 、ガバナンス要件への準拠に影響するランディングゾーンの変更を検出したりできます。

DVSM

「開発バリューストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

EDI

「電子データ交換」を参照してください。

エッジコンピューティング

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

電子データ交換 (EDI)

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

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

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

エンタープライズリソースプランニング (ERP)

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

エンベロープ暗号化

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

環境

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

- 開発環境 — アプリケーションのメンテナンスを担当するコアチームのみが使用できる、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。たとえば、 AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。 AWS 移行戦略のエピックの詳細については、プログラム実装ガイドを参照してください。

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

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

F

ファクトテーブル

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

フェイルファスト

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

障害分離境界

では AWS クラウド、アベイラビリティーゾーン 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で、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。詳細については、「AWS 障害分離境界」を参照してください。

機能ブランチ

「ブランチ」を参照してください。

特徴量

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

特徴量重要度

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

機能変換

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

数ショットプロンプト

LLM に、タスクと望ましい出力を示す例を少数提示した後に、類似のタスクを実行させること。この手法は、プロンプトに記述された例 (ショット) からモデルが学習する「インコンテキスト学習」の一種です。数ショットプロンプトは、特定のフォーマット、推論、専門知識が必要なタスクに効果的です。「ゼロショットプロンプト」も参照してください。

FGAC

「きめ細かなアクセス制御」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

FM

「基盤モデル」を参照してください。

基盤モデル (FM)

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

FM ゲートウェイ

基盤モデルへのアクセスを制御および正規化する一元化された仲介者。LLM ゲートウェイとも呼ばれます。

G

生成 AI

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

ジオブロッキング

「地理的制限」を参照してください。

地理的制限 (ジオブロッキング)

特定の国のユーザーがコンテンツ配信にアクセスできないようにするための、Amazon CloudFront のオプション。アクセスを許可する国と禁止する国は、許可リストまたは禁止リストを使って指定します。詳細については、CloudFront ドキュメントの「コンテンツの地理的ディストリビューションの制限」を参照してください。

Gitflow ワークフロー

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

ゴールデンイメージ

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

グリーンフィールド戦略

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

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーと IAM アクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは AWS Config、、Amazon GuardDuty AWS Security Hub CSPM、、 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。

ガードレール (AI)

エージェントの入出力をフィルタリング、検証、制約して、責任のある安全な AI の動作を確保するのに役立つ安全メカニズム。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

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

高可用性 (HA)

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

ヒストリアンのモダナイゼーション

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

ホールドアウトデータ

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

ヒューhuman-in-the-loop (HitL)

エージェントの実行が重要な決定時点で人間によるレビューと承認のために一時停止するワークフローパターン。

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

「インダストリアル IoT」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (受信) VPC

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

増分移行

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

インダストリー 4.0

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

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

インダストリアル IoT (IIoT)

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

インスペクション VPC

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

IoT

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

解釈可能性

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

IoT

「IoT」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

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

大規模言語モデル (LLM)

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

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースアクセス制御」を参照してください。

最小特権

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

リフトアンドシフト

「7 Rs」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

LLM

「大規模言語モデル」を参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「ブランチ」を参照してください。

マルウェア

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

マネージドサービス

AWS のサービス はインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、エンドポイントにアクセスしてデータを保存および取得します。

マネージドサービスの例として、Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB が挙げられます。このサービスは、抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

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

MAP

「Migration Acceleration Program」を参照してください。

MCP

「モデルコンテキストプロトコル」を参照してください。

モデルコンテキストプロトコル (MCP)

エージェントとツール間の通信のためのステートレスプロトコル。

MCP サーバー

モデルコンテキストプロトコルを通じて 1 つ以上のツールを公開するサービス。

メカニズム

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

メンバーアカウント

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

MES

「製造実行システム」を参照してください。

Message Queuing Telemetry Transport (MQTT)

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

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれ

る場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、AWS 「サーバーレスサービスを使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

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

大規模な移行

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

移行ファクトリー

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

移行メタデータ

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

移行パターン

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

Migration Portfolio Assessment (MPA)

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

移行準備状況評価 (MRA)

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

移行戦略

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

ML

「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定された

ギャップに対処するためのアクションプランが得られます。詳細については、「AWS クラウドでのアプリケーションのモダナイゼーションの準備状況を評価する」を参照してください。

モノリシックアプリケーション (モノリス)

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

MPA

「Migration Portfolio Assessment」を参照してください。

MQTT

「Message Queuing Telemetry Transport」を参照してください。

多クラス分類

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

ミュータブルなインフラストラクチャ

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

O

OAC

「オリジンアクセス制御」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

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

OI

「オペレーション統合」を参照してください。

Ola

「オペレーショナルレベルアグリーメント」を参照してください。

オンライン移行

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

OPC-UA

「Open Process Communications - Unified Architecture」を参照してください。

Open Process Communications - Unified Architecture (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

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

運用テクノロジー (OT)

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

オペレーション統合 (OI)

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

組織の証跡

組織 AWS アカウント 内のすべての 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により、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、OCM ガイドを参照してください。

オリジンアクセス制御 (OAC)

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

オリジンアクセスアイデンティティ (OAI)

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

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用テクノロジー」を参照してください。

アウトバウンド (送信) VPC

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

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

「個人を特定できる情報」を参照してください。

プレイブック

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

PLC

「プログラマブルロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

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

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

アクションを実行し AWS 、リソースにアクセスできる 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」にあるプリンシパルを参照してください。

プライバシーバイデザイン

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

プライベートホストゾーン

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

プロアクティブコントロール

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

製品ライフサイクル管理 (PLM)

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

本番環境

「環境」を参照してください。

プログラマブルロジックコントローラー (PLC)

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

プロンプトチェイニング

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

仮名化

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

発行/サブスクライブ (pub/sub)

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

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

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

RAG

「検索拡張生成」を参照してください。

ランサムウェア

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

RASCI マトリックス

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

RCAC

「行と列のアクセス制御」を参照してください。

リードレプリカ

読み取り専用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。

リアーキテクト

「7 Rs」を参照してください。

目標復旧時点 (RPO)

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

目標復旧時間 (RTO)

サービス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、「アカウントが使用できる AWS リージョン を指定する」を参照してください。

リグレッション

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

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 Rs」を参照してください。

リプラットフォーム

「7 Rs」を参照してください。

再購入

「7 Rs」を参照してください。

回復性

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

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンシブコントロール

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、「 AWSでのセキュリティコントロールの実装」の「レスポンシブコントロール」を参照してください。

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

検索拡張生成 (RAG)

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

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「目標復旧時点」を参照してください。

RTO

「目標復旧時間」を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーテ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 IAM で作成しなくても、 にログイン AWS マネジメントコンソール したり AWS 、API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「SAML 2.0 ベースのフェデレーションについて」を参照してください。

SCADA

「監視制御とデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

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

セキュリティバイデザイン

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

セキュリティコントロール

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

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

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

サーバー側の暗号化

送信先で、それ AWS のサービス を受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

のエントリポイントの 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、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

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

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、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。

シャドウ AI

組織内の管理対象チャネルの外部で構築または使用される認可されていない AI アプリケーション。

SIEM

「Security Information and Event Management システム」を参照してください。

単一障害点 (SPOF)

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

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケー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、「AWS クラウドでのアプリケーションをモダナイズするための段階的アプローチ」を参照してください。

SPOF

「単一障害点」を参照してください。

スタースキーマ

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

strangler fig パターン

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

サブネット

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

監視制御とデータ取得 (SCADA)

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

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

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

システムプロンプト

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

T

タグ

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

ターゲット変数

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

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要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。

tool

エージェントが外部システムでオペレーションを実行するために呼び出す???ことができる関数または API。

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「 を他の AWS のサービス AWS Organizations で使用する AWS Organizations 」を参照してください。

チューニング

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

ツーピザチーム

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

U

不確実性

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

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

「環境」を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングできる、2 つの VPC 間の接続。詳細については、Amazon VPC ドキュメントの「VPC ピア機能とは」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

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

ワークロード

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

ワークストリーム

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

WORM

「Write-Once-Read-Many」を参照してください。

WQF

「AWS ワークロード資格フレームワーク」を参照してください

Write-Once-Read-Many (WORM)

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

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性を悪用した攻撃 (一般的にマルウェアによる)。

ゼロデイ脆弱性

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

ゼロショットプロンプト

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

ゾンビアプリケーション

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

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