を使用したハイブリッドクラウドのミュータブルインスタンスの自動パッチ適用 AWS Systems Manager

AWS 規範ガイダンス

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 規範ガイダンス: を使用したハイブリッドクラウドのミュータブルインスタンスの自動パッチ適用 AWS Systems Manager

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

AWS Systems Manager を使用したハイブリッドクラウドにおけるミュータブル・インスタンスの自動パッチ適用

チャンドラ・アラカ、Amazon Web Services（AWS）

2020 年 6 月（「ドキュメント履歴」）

この規定ガイドでは、Amazon Web Services (AWS) Systems Managerを使用した自動パッチ適用ソリューションについて説明します。このソリューションを使用して、AWS 複数のアカウントとAWS リージョンにまたがる変更可能な（長時間稼働する）Amazon Elastic Compute Cloud (Amazon EC2) インスタンスとオンプレミスインスタンスの両方にパッチを適用できます。

このガイドは、アプリケーションチームが自社のパッチポリシーに準拠できるようにするためのハイブリッドクラウド環境における運用機能の設計と構築に携わっているユーザーを対象としています。事前に承認されたパッチをアプリケーションサーバーに導入するためのセルフサービスの仕組みを提供します。

本ガイドは、以下の AWS サービスとコンセプトを十分に理解していることを前提としている：

- 「Systems Manager」 – 複数の AWS サービスの運用データを閲覧し、AWS リソース全体で運用タスクを自動化するための統合されたユーザーインターフェイスを提供します。
- 「Systems Manager Inventory」 – Amazon EC2およびオンプレミスのコンピューティング環境を可視化します。インベントリを使用して、マネージドインスタンスからメタデータを収集できます。
- 「Systems Manager Patch Manager」 – セキュリティ関連およびその他の種類の更新を使用してマネージド インスタンスにパッチを適用するプロセスを自動化します。
- 「Systems Manager Maintenance Windows」 – オペレーティングシステムのパッチ適用、ドライバの更新、ソフトウェアやパッチのインストールなど、インスタンス上で破壊を引き起こす可能性のあるアクションを実行するスケジュールを定義できます。
- 「AWS Lambda」――サーバーをプロビジョニングまたは管理しなくてもコードを実行できます。
- 「Amazon QuickSight」――機械学習（ML）インサイトを含むインタラクティブなダッシュボードを簡単に作成して公開できます。どのデバイスからでもダッシュボードにアクセスし、アプリケーション、ポータル、ウェブサイトに埋め込むことができます。

- 「タグ付け」 – AWS リソースにメタデータを割り当てることができます。各タグは、ユーザー定義のキーと値で構成されるラベルです。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。

パッチ管理の概要

アプリケーションやインフラストラクチャの運用に携わっている方なら、アプリケーションチームからの多様な要件に対応できる柔軟性と拡張性を備えたオペレーティングシステム (OS) パッチソリューションの重要性を理解していることでしょう。一般的な組織では、不変インスタンスを含むアーキテクチャを使用するアプリケーションチームもあれば、可変インスタンスにアプリケーションをデプロイするアプリケーションチームもあります。

イミュータブルインスタンスのパッチ適用では、イミュータブル EC2 アプリケーションインスタンスのプロビジョニングに使用される Amazon マシンイメージ (AMI) にパッチを適用します。可変インスタンスパッチでは、スケジュールされたメンテナンス期間中に、実行中のインスタンスにパッチをインプレースでデプロイします。

この規範ガイドでは、AWS Systems Manager Patch Manager を使用して、アプリケーションチームがサーバー上でタグを使用して定義したメンテナンスウィンドウとパッチグループに基づいて、複数の AWS アカウントと AWS リージョンにまたがる可変インスタンスに自動的にパッチを適用する方法について説明します。

このガイドでは、AWS Lambda を使用し、パッチマネージャーとメンテナンスウィンドウを使用して、パッチの設定とスケジューリングを自動化する自動パッチソリューションについて説明します。Amazon QuickSight には、パッチコンプライアンスに関するレポートに必要なレポート機能とダッシュボード機能が備わっています。

また、このガイドでは、ハイブリッドクラウド環境のリファレンスアーキテクチャについても説明します。ハイブリッドクラウドのセットアップでアプリケーションを運用するユーザーは、AWS とオンプレミスのインフラストラクチャにまたがるパッチ管理業務を統合、簡素化、標準化、最適化する機会を求めています。このガイドでは、可変インスタンス用の自動パッチ適用ソリューションを拡張してハイブリッドクラウドシナリオをサポートする方法について説明しています。

このガイドでは、以下を取り上げています:

- パッチ管理に関する主なユーザーストーリー
- パッチの適用プロセス
- 単一アカウントと単一 AWS リージョンにおける可変インスタンスのパッチ管理、アーキテクチャ上の考慮事項と制限事項
- マルチアカウント、マルチリージョン環境におけるミュータブルインスタンスのパッチ管理；建築上の考慮事項と制限

- ハイブリッドクラウド環境におけるオンプレミスインスタンスのパッチ管理；アーキテクチャ上の考慮事項と制限事項
- 主要利害関係者、役割、責任

> **Note**
>
> このガイドでは、ミュータブルインスタンスのパッチ管理要件をサポートするために実装できる自動化ソリューション (自動化パッチソリューションと呼ばれます) のアーキテクチャについて説明します。ソリューションを構築するためのコードは提供されていません。

用語と概念

期間	定義
イミュータブルインスタンス	イミュータブルインスタンスとは、実行中も変更されない EC2 サーバーインスタンスのことです。変更が必要な場合は、更新されたサーバーイメージを使用して新しいインスタンスを作成し、そのインスタンスを再デプロイして、既存のサーバーイメージを破棄します。
パッチベースライン	パッチベースラインは OS タイプによって異なり、インスタンスへのインストールが承認されたパッチリストを定義します。詳細については、Systems Manager ドキュメントの「定義済みパッチベースラインとカスタムパッチベースラインについて」を参照してください。
パッチグループ	パッチグループは、特定のパッチベースラインの対象となるアプリケーション環境内のサーバーを表します。パッチグループは、正しい一連のインスタンスに、正しいパッチベースラインのデプロイを確認するのに役立ちます。また、適切なテストの完了前にパッチがデプロイされることも回避できます。パッチグループ

期間	定義
	は [パッチグループ] タグで表されます。詳細については、Systems Manager ドキュメントの「パッチグループについて」を参照してください。
メンテナンスウィンドウ	メンテナンスウィンドウでは、オペレーティングシステムのパッチ適用、ドライバの更新、ソフトウェアやパッチのインストールなど、インスタンスに対して潜在的に破壊的なアクションを実行するためのスケジュールを定義できます。各メンテナンスウィンドウには、スケジュール、最大期間、登録されたターゲットインスタンスのセット、登録されたタスクのセットがあります。メンテナンスウィンドウは [メンテナンスウィンドウ] タグで表されます。詳細については、Systems Manager ドキュメントの「メンテナンスウィンドウを使用したパッチ適用スケジュールについて」を参照してください。

主要ユーザーストーリー

一般的な OS パッチ処理には次の 3 つのタスクが含まれます。

1. EC2 インスタンスとオンプレミスサーバーをスキャンして、該当する OS パッチを探します。
2. 適切なタイミングでインスタンスをグループ化し、パッチを適用します。
3. サーバー環境全体にわたるパッチ適用コンプライアンスの報告。

次の表は、パッチ管理の主なユーザーストーリーをまとめたものです。

シナリオ	ユーザーロール	説明
パッチメカニズム	アプリケーション開発 / サポートチーム	OS のパッチ適用を担当するアプリケーションチームのメ

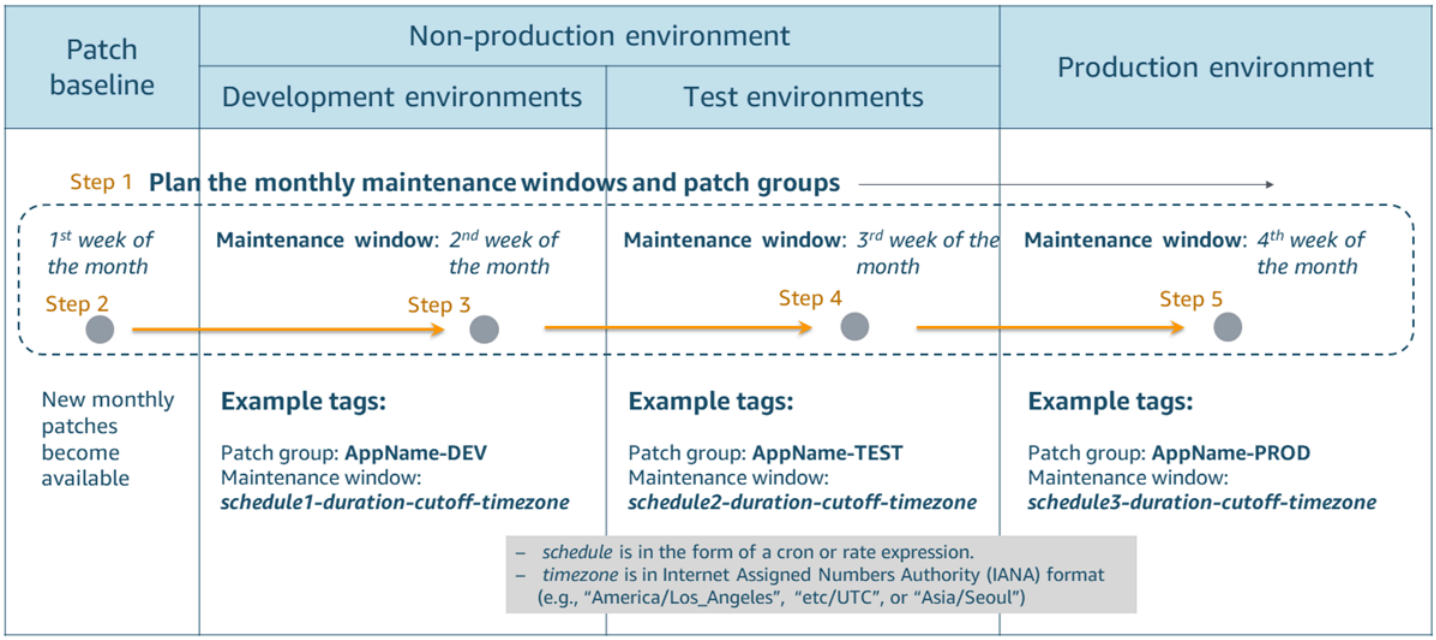
シナリオ	ユーザーロール	説明
		ンバーとして、OS のセキュリティの脆弱性を軽減し、インスタンスがセキュリティチームが定義したパッチベースラインに確実に準拠できるように、長時間実行されるインスタンスや可変インスタンスにパッチを適用するメカニズムが必要です。
パッチソリューション	クラウドサービスオーナー	アプリケーションチームへのクラウドサービスの提供を担当するクラウドサービスオーナーとして、複数の AWS アカウントや AWS リージョン、オンプレミスサーバーをサポートする OS パッチソリューションを構築する必要があります。これにより、アプリケーションチームは OS セキュリティの脆弱性を軽減し、セキュリティチームが定義したパッチベースラインへの準拠も維持できます。
パッチ適用コンプライアンスレポート	セキュリティオペレーションマネージャー	パッチコンプライアンスの確保を担当するセキュリティ運用マネージャーとして、パッチベースラインに準拠していないサーバーを特定し、必要な緩和策を実施するようチームに警告できるように、クラウド環境全体にわたる詳細なパッチコンプライアンスレポートと情報が必要です。

シナリオ	ユーザーロール	説明
役割と責任の定義	クラウドサービスオーナー	クラウドサービスオーナーとして、私は、私が構築したハイブリッドクラウドパッチングソリューションの管理において誰が何をするのかを説明する、明確に定義された役割と責任のマトリクスを構築する必要があります。

パッチ適用プロセス

パッチソリューションの主なユーザーは、アプリケーション開発チームと運用チームです。各アプリケーションは通常、開発、テスト（統合、ユーザー承認など）、本番環境など、複数の環境にデプロイされます。アプリケーションチームは、パッチが本番環境に適用された時点でテスト済みで、アプリケーションに悪影響がないことが確認されているように、環境ごとにパッチ適用スケジュールを計画する必要があります。

以下のワークフローは、複数の環境にデプロイされているアプリケーションのパッチ適用期間を計画する方法と、タグを設定する方法の例を示しています。



- ステップ 1。各アプリケーションチームは、さまざまな環境内のサーバーのメンテナンスウィンドウを計画し、それに応じてサーバーのパッチグループとメンテナンスウィンドウを表すタグを設定します。
  - [パッチグループ] タグは、特定のパッチベースラインの対象となるアプリケーション環境内のサーバーを表します。パッチグループは、正しい一連のインスタンスに、正しいパッチベースラインのデプロイを確認するのに役立ちます。パッチグループはまた、パッチが適切にテストされる前に運用環境に展開することを回避するのにも役立ちます。
  - アプリケーションサーバーに複数のオペレーティングシステムが含まれている場合、アプリケーションチームは環境とオペレーティングシステムの組み合わせに基づいてパッチグループを作成します。パッチグループは 1 つのパッチベースラインにのみ登録でき、インスタンスは 1 つのパッチグループにのみ属することができます。

例：*appname*-DEV-WIN および *appname*-DEV-RHEL

- [メンテナンスウィンド] ウタグは、サーバにパッチを適用するスケジュールを表します。パッチグループ内のすべてのサーバーが同じメンテナンスウィンドウ内にある必要があります。メンテナンスウィンドウタグは、定義した Lambda 関数が式を簡単に解析できるように、cron 式と rate 式の一貫した形式に従う必要があります。（このガイドでは、このラムダ関数を automate-patch と呼ぶことにする）。

  例：*schedule-duration-cutoff-timezone*

  cron(0 2 ? * SAT#3 *) は毎月第 3 土曜日の午前 2:00 を表します。cron 式と rate 式の詳細については、「Systems Manager ドキュメント」を参照してください。

- ステップ 2。Systems Manager Patch Manager は、定義されたコンフィギュレーションに基づき、オペレーティング・システム固有のパッチ・ベースラインを通じて、新しいパッチを定期的に提供します。
  - オペレーティングシステムごとに、クラウド環境全体のインスタンスに適用する必要がある承認ルールとパッチを含むカスタム パッチベースラインを定義できます。
- ステップ 3。カスタム自動化コードにより、[パッチグループ] と[メンテナンスウィンドウ] タグに基づいてパッチを適用するように Patch Manager を構成し、パッチを開発環境に適用します。
  - パッチ適用が完了すると、アプリケーション開発チームとサポートチームがアプリケーションをテストし、すべてが正しく動作することを確認します。
  - 新しいパッチでアプリケーションに問題が発生した場合、アプリケーションチームはクラウドサービスチームに、メンテナンスウィンドウを無効にするか、パッチタスク実行の登録を解除して、他のパッチグループや他の環境へのパッチ適用を停止するよう依頼します。
- ステップ 4。開発環境のパッチが成功したら、本番環境以外の環境にもパッチを適用します。開発環境と同様に、アプリケーションはテストされ、すべての非本番環境で正しく動作することが検証されます。問題があれば、アプリケーションチームはクラウドサービスチームに本番環境へのパッチ適用を中止するよう要請します。
- ステップ 5。すべての非本番環境のパッチが成功した後、本番環境にパッチが適用されます。

ミュータブル EC2 インスタンスのパッチソリューション設計

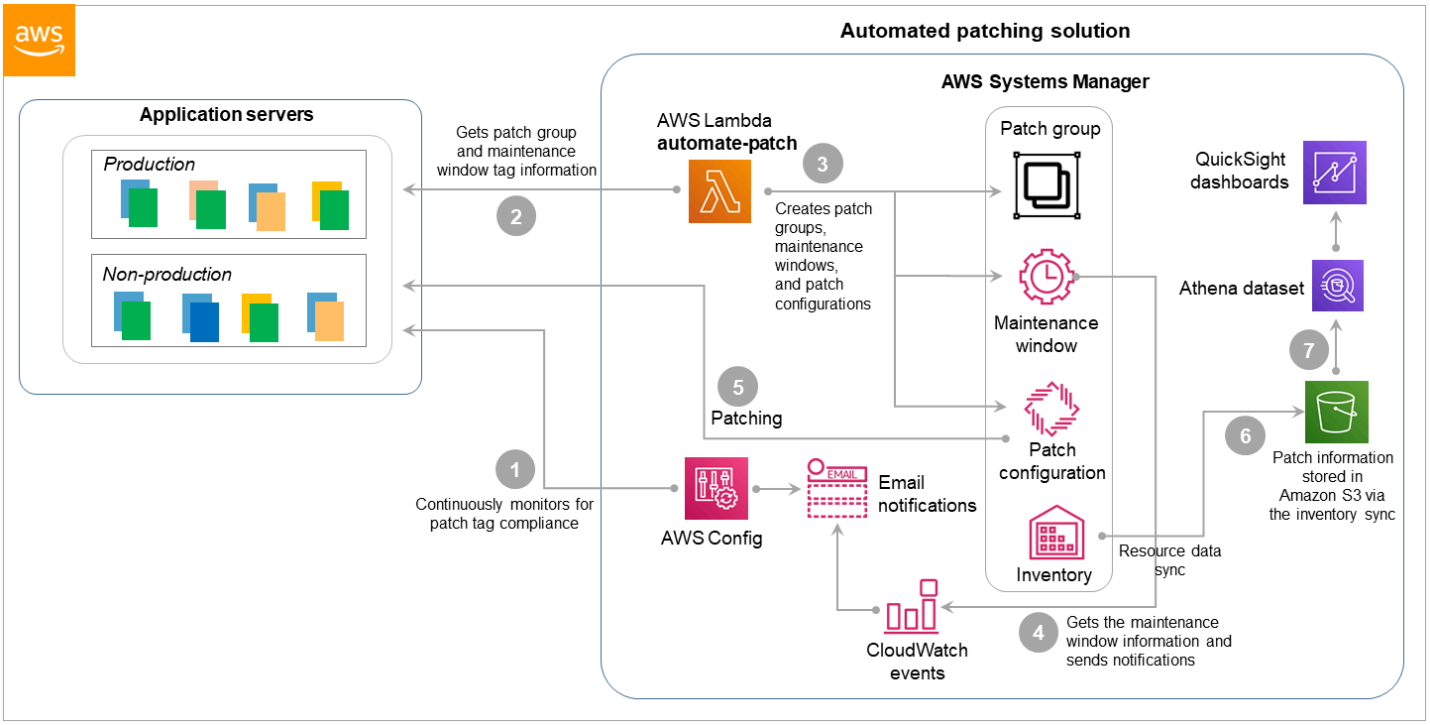
ミュータブルインスタンスのパッチ適用プロセスには、以下のチームとアクションが含まれます。

- アプリケーション (DevOps) チームは、アプリケーション環境、OSタイプ、またはその他の基準に基づいて、サーバーのパッチグループを定義します。また、各パッチグループ固有のメンテナンスウィンドウも定義します。この情報は EC2 アプリケーションインスタンスのパッチグループとメンテナンスウィンドウタグに保存されます。各パッチサイクル中、アプリケーションチームはパッチ適用の準備、パッチ適用後のアプリケーションのテスト、パッチ適用中のアプリケーションと OS の問題のトラブルシューティングを行います。
- セキュリティ運用チームは、アプリケーションチームが使用するさまざまな OS タイプのパッチベースラインを定義し、パッチを承認して、Systems Manager Patch Manager を通じてパッチを利用できるようにします。
- 自動パッチソリューションは定期的に実行され、ユーザー定義のパッチグループとメンテナンスウィンドウに基づいて、パッチベースラインに定義されているパッチを配布します。パッチコンプライアンス情報は Systems Manager インベントリ内のリソースデータ同期によって取得され、Amazon QuickSight ダッシュボードによるパッチコンプライアンスレポートに使用されます。
- ガバナンスチームとコンプライアンスチームは、パッチガイドラインを定義し、例外プロセスとメカニズムを定義し、Amazon QuickSight からコンプライアンスレポートを取得します。

OS パッチ管理ソリューションを成功に導いた主要な利害関係者とその責任の詳細については、本ガイドの後半にある主要な利害関係者、役割、責任セクションを参照してください。

自動化プロセス

自動パッチソリューションでは、連携して動作する複数の AWS サービスを使用して EC2 インスタンスにパッチをデプロイします。このプロセスには、AWS Config、AWS Lambda、Systems Manager、Amazon Simple Storage Service (Amazon S3)、および Amazon QuickSight が含まれます。以下の図は、リファレンスアーキテクチャとワークフローを示しています。



ワークフローには以下のステップが含まれており、ステップ番号は図のコールアウトと一致しています。

1. AWS Config は以下を継続的に監視し、準拠していないインスタンスの詳細と必要な設定を含む通知を送信します。
   - EC2 インスタンスでのパッチタグ付けコンプライアンス。AWS Config は パッチグループとメンテナンスウィンドウタグがないインスタンスをチェックします。
   - AWS Identity and Access Management (IAM) インスタンスプロファイルに Systems Manager ロールを設定し、Systems Manager がインスタンスを管理できます。
2. Lambda 関数 (ここでは `automate-patch` と呼びます) は、あらかじめ定義されたスケジュールで実行され、すべてのサーバーのパッチグループとメンテナンスウィンドウの情報を収集します。
3. 次に `automate-patch` 機能は、適切なパッチグループとメンテナンスウィンドウを作成または更新し、パッチグループとパッチベースラインを関連付け、パッチスキャンを設定し、パッチタスクを展開します。オプションで、`automate-patch` 関数は Amazon CloudWatch Events にイベントを作成して、差し迫ったパッチをユーザーに通知します。
4. メンテナンスウィンドウに基づき、イベントはアプリケーションチームに、間近に迫ったパッチ処理の詳細を示すパッチ通知を送信します。

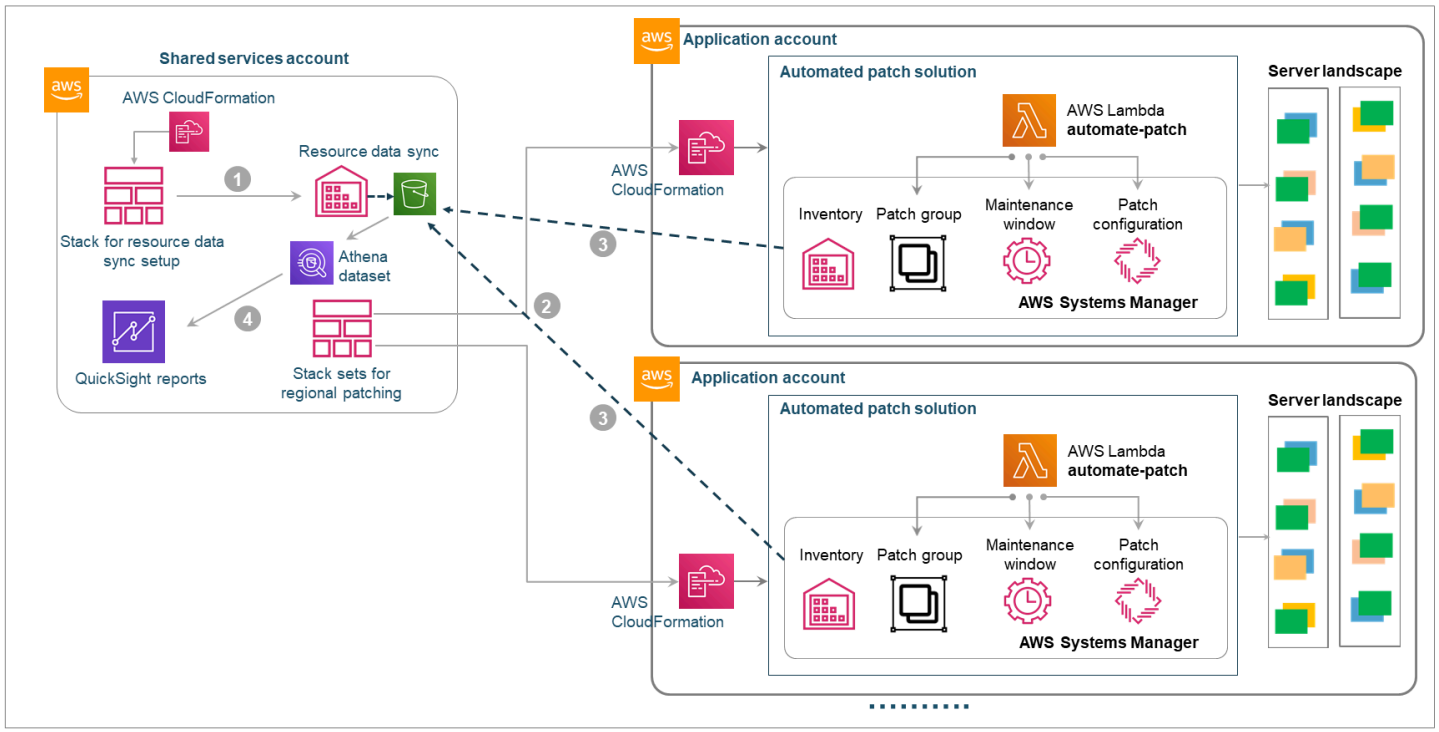
5. パッチマネージャーは、定義されたスケジュールとパッチグループに基づいてシステムパッチを実行します。
6. Systems Manager Inventory のリソースデータ同期が、パッチの詳細を収集し、S3 バケットにパブリッシュします。
7. パッチコンプライアンスレポートとダッシュボードは、S3 バケット情報から Amazon QuickSight に組み込まれています。

複数の AWS アカウントとリージョンのパッチソリューション設計

自動パッチ適用ソリューションを拡張して、複数の AWS アカウントと複数の AWS リージョンにまたがるサーバーをサポートすることができます。この拡張ソリューションでは、共有サービスアカウントの AWS CloudFormation StackSets を通じて、各 AWS アカウントにパッチ自動化ソリューションを設定し、共有サービスアカウントでアカウント間のリソースデータ同期を設定します。

自動化プロセス

次の図は、このシナリオのアーキテクチャーを示しています。このアーキテクチャには、AWS CloudFormation スタックセットと AWS 共有サービスアカウントが含まれます。



ワークフローは前のセクションで説明したプロセスと似ていますが、以下の追加ステップが含まれます。ステップ番号は図のコールアウトと一致します。

1. 共有サービスアカウントでは、AWS CloudFormation スタックセットを使用して、Systems Manager Inventory によるリソースデータの同期用に S3 バケットを設定します。

2. CloudFormation スタックセットは、automate-patch Lambda 関数を使用してスタックを作成し、パッチベースラインを設定し、アプリケーションアカウントの Systems Manager インベントリリソースデータ同期を設定して、共有サービスアカウントのリソースを同期します。
3. アプリケーションアカウントのリソース情報は、共有サービスアカウントのリソース情報と同期されます。
4. Amazon QuickSight は、同期されたリソース情報の Amazon Athena データセットを使用して、パッチコンプライアンスレポートを生成します。

アーキテクチャ上の考慮と制約事項

アカウントごとのメンテナンスウィンドウのクォータ

前のセクションで説明したアーキテクチャでは、パッチグループごとにメンテナンスウィンドウが作成されます。ただし、AWS アカウントごとのメンテナンス・ウィンドウ数のクォータは 50 です（サービス・クォータの増加をリクエストしていないことを前提とします）。パッチグループの数が1つの AWS アカウントで50グループを超えることが予想される場合、このアーキテクチャでは要件を満たすために拡張できません。

サービスクォータの増加だけでは十分でない場合、この課題を管理するための 2 つのオプションがある：事前定義されたメンテナンスウィンドウを使用することと、CloudWatch Eventsを使用することです。それぞれのアプローチの長所と短所は以下の通りです。

オプション 1. 定義済みのメンテナンスウィンドウを使用します

- さまざまなタイムウィンドウでメンテナンスウィンドウのリストを定義します（たとえば、アカウントごとに 15 ~ 20 のメンテナンスウィンドウ）。
- アプリケーションチームは事前に定義されたリストから自分に合ったメンテナンスウィンドウを選択し、それに応じてインスタンスにタグを付けます。
- 新しいメンテナンスウィンドウを作成する代わりに、自動パッチソリューションを更新して、選択したメンテナンスウィンドウにパッチグループをマッピングします。

メリット：

- 簡略化された管理

デメリット：

- カスタムのメンテナンスウィンドウを定義する柔軟性が低いです。
- 複数のパッチグループがメンテナンスウィンドウとパッチタスクを共有している場合、特定のパッチグループの特定のパッチタスクをキャンセルすると、追加の手動作業が必要になります。

オプション 2. メンテナンスウィンドウを使用する代わりに CloudWatch Events を使用してパッチタスクをトリガーします

- メンテナンスウィンドウを作成する代わりに、CloudWatch Events を使用してスケジュールとパッチグループに基づいてパッチタスクをトリガーします。
- このシナリオでは、各パッチグループはメンテナンスウィンドウではなく CloudWatch Events イベントに関連付けられます。
- 自動パッチ適用ソリューションを更新して、メンテナンスウィンドウではなくイベントを作成します。

メリット：

- スケーラブルなデザイン。
- カスタムメンテナンスウィンドウを柔軟に定義できます。

デメリット：

- メンテナンスウィンドウには、CloudWatch Events では利用できない追加機能（期間や締め切り時間など）が用意されています。

その他の考慮事項

- このセクションで説明する自動パッチソリューションは、シャットダウンされた EC2 インスタンスをサポートしません。
- このプロセスはパブリックサブネットの EC2 インスタンスをサポートします。プライベートサブネットのインスタンスにパッチを適用するには、「Windows Server Update Services（WSUS）などのローカルパッチリポジトリ」をデプロイする必要があります。
- パッチグループとメンテナンスウィンドウが必要なスケジュールに従って更新されるように、Lambda 関数の実行頻度を調整する必要があります。

ハイブリッドクラウド環境におけるオンプレミスインスタンスのパッチソリューション設計

このガイドで説明されているソリューションを拡張して、ハイブリッドクラウド環境のオンプレミスサーバーインスタンスにパッチを適用することもできます。

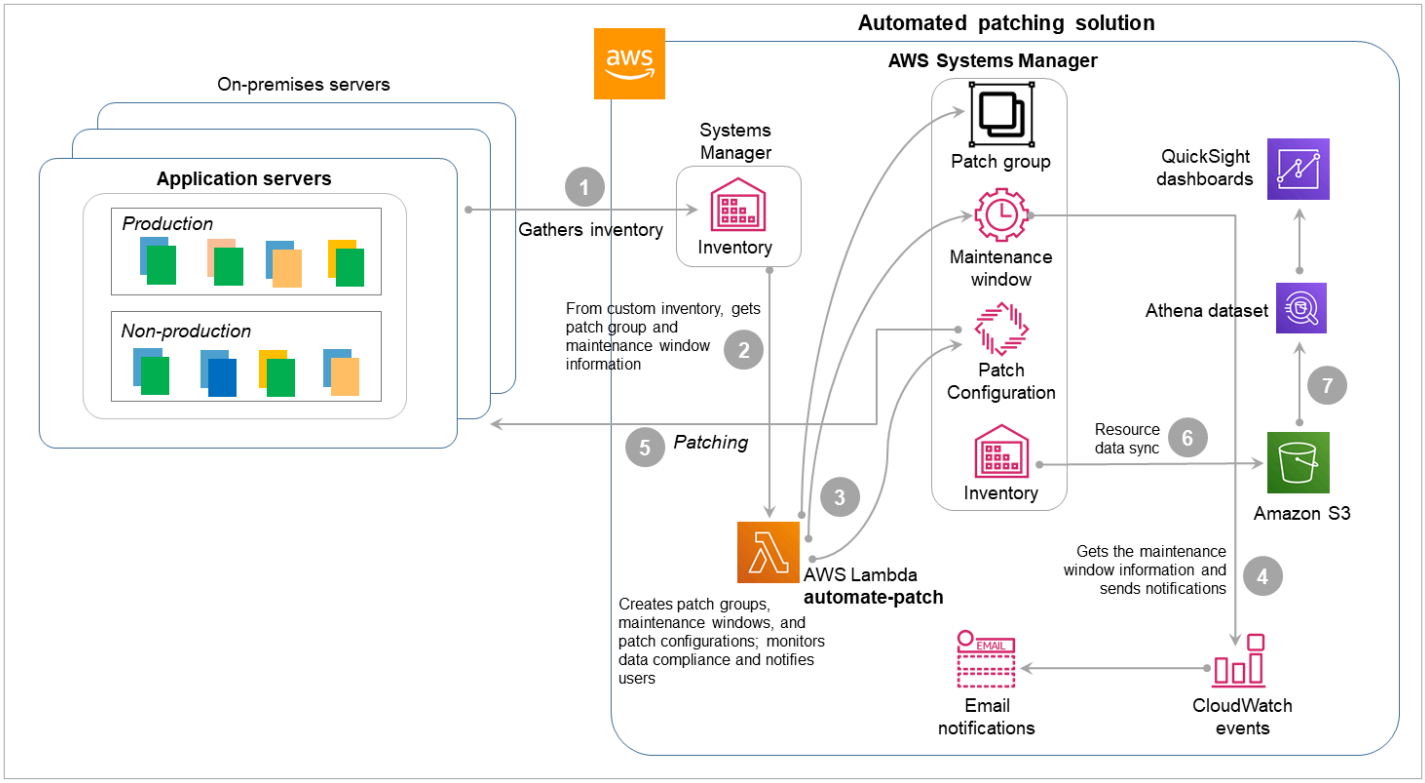
オンプレミス・インスタンスの標準的なパッチ適用プロセスは、次の 2 つのステップで構成されます。

- オンプレミスサーバーを Systems Manager によって管理されるように設定します。このプロセスの詳細については、Systems Manager ドキュメントの「ハイブリッド環境用の Systems Manager のセットアップ」を参照してください。
- AWS Command Line Interface（AWS CLI）の「add-tags-to-resourceコマンド」を使用して、これらのオンプレミスの管理対象インスタンスに適切な[パッチグループ] と[メンテナンスウィンドウ] タグを設定します。

しかし、この方法では、アプリケーションチームかクラウドチームのいずれかが、パッチグループやメンテナンスウィンドウに変更を加えたいときに、AWS CLIコマンドを手動で実行する必要があります。

自動化プロセス

次の図は、Systems Manager カスタムインベントリオプションを使用するオンプレミスインスタンスにパッチを適用する代替方法を示しています。このプロセスは、先ほど説明した変更可能な EC2 インスタンス用の自動パッチ適用ソリューションを拡張したものです。



1. Systems Manager は、タグを使用する代わりに、カスタムインベントリコレクションを通じてオンプレミスの管理対象インスタンスからパッチ情報 (パッチグループとメンテナンスウィンドウ) を取得します。

```
Sample custom inventory JSON file
{
    "SchemaVersion": "1.0",
    "TypeName": "Custom:PatchInformation",
    "Content": {
        "Patch Group": "<APP-PROD>",
        "Maintenance Window": "XXX"
    }
}
```

2. ラムダ automate-patch 機能は毎日実行され、オンプレミスのサーバーカスタムインベントリからパッチグループとメンテナンスウィンドウの情報を収集し、管理対象のインスタンスに[パッチグループ] と[メンテナンスウィンドウ] のタグを作成します。
3. 次にLambda automate-patch 関数は、収集したカスタムインベントリに基づいて、適切なパッチグループとメンテナンスウィンドウを作成または更新し、パッチグループをパッチベースラインに関連付け、パッチスキャンを設定し、パッチタスクを展開します。オプション

で、automate-patch 機能は CloudWatch Events にイベントを作成し、差し迫ったパッチをユーザーに通知します。

4. メンテナンスウィンドウに基づき、イベントはアプリケーションチームに、間近に迫ったパッチ処理の詳細を示すパッチ通知を送信します。
5. パッチマネージャーは、定義されたスケジュールとパッチグループに基づいてシステムパッチを実行します。
6. Systems Manager Inventory のリソースデータ同期が、パッチの詳細を収集し、S3 バケットにパブリッシュします。
7. パッチコンプライアンスレポートとダッシュボードは、S3 バケット情報から Amazon QuickSight に組み込まれています。

アーキテクチャ上の考慮と制約事項

前のセクションで説明したように、オンプレミスインスタンスにパッチを適用するには、カスタムインベントリを使用する方法とタグを使用する方法の 2 つがあります。それぞれのアプローチの長所と短所は以下の通りです。

オプション 1. カスタムインベントリをパッチ情報に使用する

- オンプレミスサーバーを操作するアプリケーションチームがカスタムインベントリファイルにパッチ情報を設定し、Systems Manager がその情報を選択します。
- 次に、カスタムインベントリパッチ情報を使用してパッチタスクが作成されます。

メリット

- ファイルの更新のみが必要なため、設定がはるかに簡単になります。

デメリット

- パッチ設定の変更は、インベントリ収集スケジュールに限定されます。

オプション 2. オンプレミスのマネージドインスタンスにタグを使用する

- オンプレミスサーバーを扱うアプリケーションチームは、適切なパッチ情報を持つ AWS CLI を使用して、[パッチグループ] と[メンテナンスウィンドウ] タグを作成します。
- タグ情報はパッチタスクの作成に使用されます。

メリット

- パッチの標準化と自動化を推進するため、AWS とオンプレミス全体およびオンプレミスでの一貫したアプローチです。

デメリット

- オンプレミスのインスタンスで作業するアプリケーションチームは、タグを作成または更新するために AWS CLI を学び、使用する必要があります。

パッチ管理における主な利害関係者、役割、責任

OS パッチ管理を成功させるには、自動パッチソリューションをサポートし、継続的に最適化するための役割と責任を明確に定義する必要があります。このセクションでは、あなたのニーズや組織構造に応じて変更できる、推奨される役割と責任について説明します。

ユーザーペルソナ

次の表では、自動パッチ適用ソリューションに関係するユーザーペルソナについて説明しています。

ユーザーペルソナ	説明
コンシューマー（C）	長期稼働インスタンス用のパッチ管理ソリューションは、OS 管理に携わるさまざまなチームによって使用されている： • フルスタックのアプリケーション環境を管理する開発チーム。 • アプリケーションサーバー OS を管理する運用チーム。
クラウドエンジニアリング（CE）	担当チーム： • パッチ管理ソリューションを継続的に最適化します。 • クラウドサービスの自動化の構築。 • 自動化をサポートします。
クラウド・ビジネス・オフィス（CBO）	関与チーム： • ソリューションのコンシューマーエクスペリエンスを管理します。 • イネーブルメントとユーザーエンゲージメント。 • パッチソリューションが消費者のニーズを満たしていることを確認します。

ユーザーペルソナ	説明
クラウドサービス / プロダクトオーナー（CPO）	責任者： • 消費者にクラウドサービスを提供します。 • リーダーシップチームと緊密に連携して、期待とガイドラインに沿ったサービスの提供を行います。 • プラットフォームに関する顧客の期待や エスカレーションをすべて管理します。 • プラットフォームロードマップを所有します。
セキュリティオペレーション（SO）	パッチベースラインと承認を管理するチーム。
セキュリティオペレーションマネージャ（SOM）	パッチのコンプライアンスを担当するマネージャー。

RACI マトリックス

以下の責任者、説明責任者、協議責任者、情報提供責任者（RACI）マトリックスは、パッチマネジメントソリューションに関わる活動を特定するものです。プロセスの各ステップについて、利害関係者とその関与が記載されている：

- R――ステップを完了する責任者
- A――作業の承認と承認を担当する責任
- C――タスクに意見を提供するために相談される
- I――進捗状況は知らされるが、タスクには直接関与しない

パッチ管理ソリューション	CPO	CBO	CE	SO	SOM	C
パッチ管理、製品ロ	A	C	R	C	C	I

パッチ管理ソリューションロードマップの実行	CPO	CBO	CE	SO	SOM	C
パッチ管理のアーキテクチャと設計	A	I	R	C	I	
パッチ管理の開発と設定	A		R	C		
パッチ管理の検証とテスト	A	I	R	I	I	
新規 AWS アカウント、アプリケーション、サーバーのオンボーディングによるパッチ適用	A	C	R	I		
ユーザーエンゲージメントとイネーブルメント	A	R	I	I	I	

パッチ管理ソリューション	CPO	CBO	CE	SO	SOM	C
ユーザーからのフィードバックとエスカレーション管理	A	R		I	I	
製品変更管理	A	R	C	I		
問題管理と解決	A		R	C		
サーバーのパッチ適用とパッチコンプライアンス			C	C		AR
パッチベースライン設定			C	R	A	C
パッチレポートとコンプライアンス			C	R	AR	I

次のステップ

本ガイドでは、ハイブリッドクラウド環境における、AWS およびオンプレミスインスタンス上のミュータブルインスタンスに対する自動パッチ適用ソリューションについて説明しました。ソリューションを構築するには、このガイドで説明されている AWS サービスのドキュメントを参照することをお勧めします。ご質問がある場合は、AWS アカウント· チームまでお問い合わせください。

詳細については、「追加リソース」のセクションを参照してください。

その他のリソース

AWS リソース

- 「AWS 規定ガイダンス」
- 「AWS ドキュメント」
- AWS general reference
- 「AWS 用語集」

AWS サービス

- AWS CloudFormation
- Amazon CloudWatch
- Amazon EC2
- IAM
- AWS Lambda
- Amazon QuickSight
- AWS Systems Manager

その他のリソース

- （AWS 管理とガバナンスに関するブログ）「AWS Systems Manager を使用してプライベートサブネット内の Amazon EC2 インスタンスにパッチを適用する方法」
- 「ムーディーズが複数のクラウドプロバイダーのサーバーに AWS Systems Manager を使ってパッチを適用する方法」（AWS 管理とガバナンスに関するブログ）
- 「ハイブリッド環境の AWS Systems Manager を設定する」（Systems Manager ドキュメント）
- AWS Systems Manager Automationでマルチアカウント、マルチリージョンのパッチ適用を一元化（AWS Management & Governance blog）
- 「AWS Systems Manager パッチマネージャーを使用して Amazon EC2 インスタンスにパッチを適用する」（AWS 管理とガバナンスに関するブログ）
- 「AWS でMicrosoft Windows ワークロードのパッチ、検査、保護を行う方法――第 1 部」（AWS セキュリティブログ）

ドキュメント履歴

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

変更	説明	日付
初版発行	—	2020 年 6 月 12 日

AWS 規範ガイダンス用語集

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 Postgre SQL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インスタンスで Oracle に移行します AWS クラウド。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例: を移行する Microsoft Hyper-V へのアプリケーション AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。
- 使用停止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

属性ベースのアクセスコントロール を参照してください。

抽象化されたサービス

「マネージドサービス」を参照してください。

ACID

原子性、一貫性、分離、耐久性 を参照してください。

アクティブ - アクティブ移行

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

アクティブ - パッシブ移行

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

集計関数

行のグループで動作し、グループの単一の戻り値を計算SQLします。集計関数の例には、 SUMおよび が含まれます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。移行戦略で AWS がどのようにAIOps使用されるかの詳細については、「 オペレーション統合ガイド」を参照してください。

非対称暗号化

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

原子性、一貫性、分離、耐久性 (ACID)

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

属性ベースのアクセスコントロール (ABAC)

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

信頼できるデータソース

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

アベイラビリティーゾーン

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安価で低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

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

AWS ワークロード認定フレームワーク (AWS WQF)

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

B

不正なボット

個人または組織に混乱または害を与えることを目的としたボット。

BCP

事業継続計画 を参照してください。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective で動作グラフを使用して、失敗したログオン試行、疑わしいAPI呼び出し、および同様のアクションを調べることができます。詳細については、Detective ドキュメントのData in a behavior graphを参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「endianness」も参照してください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 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 つの別々の環境を作成するデプロイ戦略。現在のアプリケーションバージョンは 1 つの環境 (青) で実行し、新しいアプリケーションバージョンは他の環境 (緑) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

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をインデックス化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、個人や組織に混乱や害を与えることを意図したものもあります。

ボットネット

マルウェアに感染し、ボットハーダーまたはボットオペレーター と呼ばれる 1 つ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 にアクセスするための簡単な手段を提供します。詳細については、「Well-Architected」ガイダンスの AWS 「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、ホワイトペーパー AWSでのコンテナ化されたマイクロサービスの実行 の ビジネス機能を中心に組織化 セクションを参照してください。

事業継続計画 (BCP)

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

C

CAF

AWS Cloud Adoption Framework を参照してください。

Canary のデプロイ

エンドユーザーへのバージョンのスローリリースと増分リリース。自信が持てば、新しいバージョンをデプロイし、現在のバージョン全体を置き換えます。

CCoE

Cloud Center of Excellence を参照してください。

CDC

データキャプチャの変更を参照してください。

データキャプチャの変更 (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。は、同期を維持するために、ターゲットシステムの変更を監査したりレプリケートしたりするなど、CDCさまざまな目的で使用できます。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストします。AWS Fault Injection Service （ AWS FIS ） を使用して、 AWS ワークロードに負荷をかけ、そのレスポンスを評価する実験を実行できます。

CI/CD

継続的統合と継続的配信 を参照してください。

分類

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

クライアント側の暗号化

ターゲットが AWS のサービス 受信する前に、データをローカルで暗号化します。

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が に移行するときに通常実行する 4 つのフェーズ AWS クラウド：

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

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

CMDB

設定管理データベース を参照してください。

コードリポジトリ

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

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

機械学習を使用して、デジタル画像や動画などのビジュアル形式から情報を分析および抽出する AI の分野。例えば、 はオンプレミスのカメラネットワークに CV を追加するデバイス AWS Panorama を提供し、Amazon SageMaker は CV の画像処理アルゴリズムを提供します。

設定ドリフト

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

設定管理データベース (CMDB)

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

コンフォーマンスパック

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

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD is commonly described as a pipeline. CI/CD は、プロセスの自動化、生産性の向上、コード品質の向上、および迅速な提供に役立ちます。詳細については、「継続的デリバリーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

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

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが期待されたネットワークから信頼されたリソースにアクセスしていることを確実にします。詳細については、「 でデータ境界を構築する AWS」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

データベース定義言語 を参照してください。

ディープアンサンブル

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

ディープラーニング

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

defense-in-depth

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

委任管理者

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

デプロイメント

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

開発環境

環境 を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、リーンな製造プラクティス用に最初に設計されたバリューストリームマッピングプロセスを拡張します。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

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

ディメンションテーブル

スタースキーマ では、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は、通常、テキストフィールドまたはテキストのように動作する離散番号です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに一般的に使用されます。

ディザスタ

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

ディザスタリカバリ (DR)

災害によるダウンタイムとデータ損失を最小限に抑えるために使用する戦略とプロセス。詳細については、「 Well-Architected Framework」の「 でのワークロードの災害復旧 AWS: クラウドでの復旧」を参照してください。 AWS

DML

データベース操作言語 を参照してください。

ドメイン駆動型設計

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

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

ベースライン設定からの偏差の追跡。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ランディングゾーンの変化を検出したりできます。

DVSM

「開発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。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。クラウドコンピューティング と比較すると、エッジコンピューティングは通信レイテンシーを短縮し、レスポンスタイムを向上させることができます。

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

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

エンタープライズリソースプランニング (ERP)

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

エンベロープ暗号化

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

環境

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

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

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、サマリー統計を計算し、データ視覚化を作成することで実行されます。

F

ファクトテーブル

星スキーマ の中央テーブル。ビジネスオペレーションに関する定量的なデータを保存します。通常、ファクトテーブルには 2 つのタイプの列が含まれます。つまり、メジャーを含む列と、ディメンションテーブルへの外部キーを含む列です。

フェイルファースト

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

障害分離境界

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が、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。詳細については、AWS 「障害分離境界」を参照してください。

機能ブランチ

ブランチ を参照してください。

特徴量

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

特徴量重要度

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

機能変換

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

FGAC

「きめ細かなアクセスコントロール」を参照してください。

きめ細かなアクセスコントロール (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

段階的なアプローチを使用する代わりに、変更データキャプチャによる継続的なデータレプリケーションを使用して、可能な限り短時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

G

ジオブロッキング

地理的制限 を参照してください。

地理的制限 (ジオブロッキング)

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

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローはレガシーと見なされ、トランクベースのワークフローはモダンで望ましいアプローチです。

グリーンフィールド戦略

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

ガードレール

組織全体のリソース、ポリシー、コンプライアンスを管理するのに役立つ大まかなルール (OUs) 。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーとIAMアクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは、 AWS Config、 AWS Security Hub、Amazon GuardDuty、、 AWS Trusted Advisor Amazon Inspector 、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

高可用性 を参照してください。

異種混在データベースの移行

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

ハイアベイラビリティ (HA)

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

ヒストリアンのモダナイゼーション

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

同種データベースの移行

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

ホットデータ

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

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性のため、通常、ホットフィックスは一般的な DevOps リリースワークフローの外部で行われます。

ハイパーケア期間

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

I

IaC

「Infrastructure as Code」を参照してください。

ID ベースのポリシー

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

アイドル状態のアプリケーション

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

IIoT

「産業用モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

既存のインフラストラクチャを更新、パッチ適用、または変更する代わりに、本番稼働ワークロード用に新しいインフラストラクチャをデプロイするモデル。イミュータブルインフラストラクチャは、本質的にミュータブルインフラストラクチャ よりも一貫性、信頼性、予測性に優れています。詳細については、 AWS 「 Well-Architected Framework」の「イミュータブルインフラストラクチャのベストプラクティスを使用したデプロイ」を参照してください。

インバウンド (イングレス) VPC

AWS マルチアカウントアーキテクチャでは、 VPCがアプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングします。AWS セキュリティリファレンスアーキテクチャでは、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

増分移行

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

インダストリー 4.0

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

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

産業用モノのインターネット (IIoT)

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

検査 VPC

AWS マルチアカウントアーキテクチャでは、 VPCs (同一または異なる 内の AWS リージョン) 、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査VPCを管理する一元化された です。AWS セキュリティリファレンスアーキテクチャでは、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

IoT

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

解釈可能性

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

IoT

「モノのインターネット」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースのアクセスコントロール (LBAC)

ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられている必須のアクセスコントロール (MAC) の実装。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

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

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースのアクセスコントロール」を参照してください。

最小特権

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

リフトアンドシフト

7 Rs を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「endianness」も参照してください。

下位環境

環境 を参照してください。

M

機械学習 (ML)

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

メインブランチ

ブランチ を参照してください。

マルウェア

コンピュータのセキュリティまたはプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムを混乱させたり、機密情報を漏洩したり、不正アクセスを受けたりする可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービス はインフラストラクチャレイヤー、オペレーティングシステム、プラットフォームを で AWS 運用し、エンドポイントにアクセスしてデータを保存および取得します。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象サービス とも呼ばれます。

製造実行システム (MES)

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

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの採用を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作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

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

MES

「製造実行システム」を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

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

マイクロサービス

明確に定義された上で通信APIsし、通常、小規模な自己完結型チームが所有する、小規模で独立したサービス。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、AWS 「サーバーレスサービス を使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量 を使用して、明確に定義されたインターフェイスを介して通信しますAPIs。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およ

びスケーリングできます。詳細については、「 でのマイクロサービスの実装 AWS」を参照してください。

移行促進プログラム (MAP)

組織がクラウドに移行するための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP には、従来の移行を系統的な方法で実行するための移行方法論と、一般的な移行シナリオを自動化して高速化するための一連のツールが含まれています。

大規模な移行

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

移行ファクトリー

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

移行メタデータ

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

移行パターン

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

移行ポートフォリオ評価 (MPA)

に移行するためのビジネスケースを検証するための情報を提供するオンラインツール AWS クラウド。MPA は、詳細なポートフォリオ評価 (サーバーの適正サイズ、料金、TCO比較、移行コス

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

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

移行戦略

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

ML

「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

モノリシックアプリケーション (モノリス)

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

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

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

多クラス分類

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

ミュータブルインフラストラクチャ

本稼働ワークロードの既存のインフラストラクチャを更新および変更するモデル。Well-Architected Framework AWS では、一貫性、信頼性、予測可能性を向上させるために、イミュータブルインフラストラクチャをベストプラクティスとして使用することを推奨しています。

O

OAC

「オリジンアクセスコントロール」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

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

OI

オペレーション統合 を参照してください。

OLA

「運用レベルの契約」を参照してください。

オンライン移行

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

OPC-UA

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

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

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

運用レベルの契約 (OLA)

サービスレベルの契約 () をサポートするために、IT グループが相互に提供することを約束する機能的な IT グループを明確にする契約SLA。

運用準備状況のレビュー (ORR)

インシデントや潜在的な障害の範囲を理解、評価、防止、または軽減するのに役立つ質問とそれに関連するベストプラクティスのチェックリスト。詳細については、 AWS Well-Architected フレームワークの「Operational Readiness Reviews (ORR)」を参照してください。

運用テクノロジー (OT)

物理環境と連携して産業オペレーション、機器、インフラストラクチャを制御するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が Industry 4.0 変換の主要な焦点です。

オペレーション統合 (OI)

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

組織の証跡

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

組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の採用を加速し、移行に伴う問題に対処し、文化的および組織的な変化を推進することで、組織が新しいシステムや戦略の準備と移行を支援します。 AWS 移行戦略では、クラウド導入プロジェクトに必要な変化のスピードから、このフレームワークは人材アクセラレーション と呼ばれます。詳細については、OCM「 ガイド」を参照してください。

オリジンアクセスコントロール (OAC)

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

オリジンアクセスアイデンティティ (OAI)

では CloudFront、Amazon S3 コンテンツを保護するためにアクセスを制限するオプションがあります。を使用する場合OAI、 は Amazon S3 が認証できるプリンシパル CloudFront を作成します。認証されたプリンシパルは、特定の CloudFront ディストリビューションを介してのみ S3 バケット内のコンテンツにアクセスできます。も参照してください。これによりOAC、より詳細で拡張されたアクセスコントロールが提供されます。

ORR

「運用準備状況の確認」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 (出力) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されたネットワーク接続VPCを処理する です。AWS セキュリティリファレンスアーキテクチャでは、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

P

アクセス許可の境界

ユーザーまたはロールが持つことができる最大アクセス許可を設定するためにIAMプリンシパルにアタッチされるIAM管理ポリシー。詳細については、IAMドキュメントの「アクセス許可の境界」を参照してください。

個人を特定できる情報 (PII)

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

PII

個人を特定できる情報を参照してください。

プレイブック

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

PLC

「プログラム可能なロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

アクセス許可の定義 (アイデンティティベースのポリシー を参照)、アクセス条件の指定 (リソースベースのポリシー を参照)、または の組織内のすべてのアカウントに対する最大アクセス許可の定義 AWS Organizations （ サービスコントロールポリシー を参照) が可能なオブジェクト。

多言語の永続性

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

ポートフォリオ評価

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

述語

`true` または を返すクエリ条件。一般的には `false`WHERE句にあります。

述語プッシュダウン

転送前にクエリ内のデータをフィルタリングするデータベースクエリ最適化手法。これにより、リレーショナルデータベースから取得して処理する必要があるデータの量が減少し、クエリのパフォーマンスが向上します。

予防的コントロール

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

プリンシパル

アクションを実行し AWS 、リソースにアクセスできる のエンティティ。このエンティティは通常、、IAMロール AWS アカウント、またはユーザーのルートユーザーです。詳細については、「 IAMドキュメント」の「ロールの用語と概念」を参照してください。 https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html#id_roles_terms-and-concepts

プライバシーバイデザイン

エンジニアリングプロセス全体を通してプライバシーを考慮に入れたシステムエンジニアリングのアプローチ。

プライベートホストゾーン

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

プロアクティブコントロール

非準拠のリソースのデプロイを防ぐように設計されたセキュリティコントロール。これらのコントロールは、プロビジョニングされる前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、 AWS Control Tower ド

キュメントの「 コントロールリファレンスガイド」および「 のセキュリティコントロールの実装」の「プロアクティブコントロール」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

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

本番環境

環境 を参照してください。

プログラマブルロジックコントローラー (PLC)

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

仮名化

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

publish/subscribe (pub/sub)

マイクロサービス間の非同期通信を可能にするパターンで、スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの ではMES、マイクロサービスは、他のマイクロサービスがサブスクライブできるチャネルにイベントメッセージを公開できます。システムは、パブリッシュサービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

ランサムウェア

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

RASCI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

「行と列のアクセスコントロール」を参照してください。

リードレプリカ

読み取り専用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 Rs を参照してください。

復旧ポイントの目的 (RPO)

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

目標復旧時間 (RTO)

サービス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。耐障害性、安定性、耐障害性を提供するために、それぞれ AWS リージョン が分離され、他のものとは独立しています。詳細については、AWS リージョン 「 を使用できるアカウントを指定する」を参照してください。

回帰

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

リホスト

7 Rs を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

7 Rs を参照してください。

プラットフォーム変更

7 Rs を参照してください。

再購入

7 Rs を参照してください。

回復性

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

リソースベースのポリシー

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

責任、説明責任、相談、情報 (RACI) マトリックス

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

レスポンシブコントロール

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 Rs を参照してください。

廃止

7 Rs を参照してください。

ローテーション

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

行と列のアクセスコントロール (RCAC)

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

RPO

「復旧ポイントの目的」を参照してください。

RTO

「目標復旧時間」を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdPs) が使用するオープンスタンダード。この機能を使用すると、フェデレーティッドシングルサインオン (SSO) AWS Management Console が有効になるため、ユーザーは にログインしたり、 AWS API組織内のすべてのユーザーIAMに対して でユーザーを作成したりすることなく オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレー

ションの詳細については、IAMドキュメントの「2.0 SAML ベースのフェデレーションについて」を参照してください。

SCADA

「監視コントロールとデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

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

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、予防的 、検出的 、応答的 ???、およびプロアクティブ の 4 つの主なタイプがあります。

セキュリティ強化

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

セキュリティ情報とイベント管理 (SIEM) システム

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

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修正するように設計された、事前定義されたプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動レスポンスアクションの例には、VPCセキュリティグループの変更、Amazon EC2インスタンスへのパッチ適用、認証情報のローテーションなどがあります。

サーバー側の暗号化

送信先で、それ AWS のサービス を受信する によるデータの暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCPs ガードレールを定義するか、管理者がユーザーまたはロールに委任できるアクションの制限を設定します。を許可リストまたは拒否リストSCPsとして を使用して、許可または禁止されるサービスまたはアクションを指定できます。詳細については、 AWS Organizations ドキュメントの「サービスコントロールポリシー」を参照してください。

サービスエンドポイント

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

サービスレベル契約 (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

サービスレベルインジケータ によって測定される、サービスの正常性を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について共有する責任を説明するモデル。 クラウドのセキュリティ AWS はクラウドのセキュリティに責任があり、クラウドのセキュリティはユーザーの責任です。詳細については、責任共有モデルを参照してください。

SIEM

セキュリティ情報とイベント管理システム を参照してください。

単一障害点 (SPOF)

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

SLA

「サービスレベル契約」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

split-and-seed モデル

モダナイゼーションプロジェクトのスケー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「アプリケーションのモダナイズに対する段階的なアプローチ AWS クラウド」を参照してください。

SPOF

単一障害点 を参照してください。

スタースキーマ

1 つの大きなファクトテーブルを使用してトランザクションデータまたは測定データを保存し、1 つ以上の小さなディメンションテーブルを使用してデータ属性を保存するデータベース組織構造。この構造は、データウェアハウスまたはビジネスインテリジェンスの目的で使用するために設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に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、「従来の Microsoft のモダナイズ」を参照してくださいASP。NET (ASMX) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に更新する「」を参照してください。

サブネット

内の IP アドレスの範囲VPC。サブネットは、1 つのアベイラビリティ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取得 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと本番稼働をモニタリングするシステムです。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーインタラクションをシミュレートして潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。Amazon CloudWatch Synthetics を使用して、これらのテストを作成できます。

T

タグ

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

ターゲット変数

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

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要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 とオンプレミスのネットワークを相互接続するために使用できるネットワークトランジットハブ。詳細については、 AWS Transit Gateway ドキュメントの「トランジットゲートウェイとは」を参照してください。

トランクベースのワークフロー

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

信頼されたアクセス

組織内でタスクを実行するために指定したサービスに、ユーザーに代わってそのアカウント AWS Organizations でアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要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 つのピザを食べることができる小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の 2 つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性 は、データに固有のノイズとランダム性によって引き起こされます。詳細については、深層学習システムにおける不確実性の定量化 ガイドを参照してください。

未分化なタスク

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

上位環境

環境 を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

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

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

現在のレコードに何らかの形で関連する行のグループに対して計算を実行するSQL関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなど、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」を 1 回参照し、多くの を読み取ります。

WQF

AWS 「ワークロード認定フレームワーク」を参照してください。

1 回書き込み、多数読み取り (WORM)

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

Z

ゼロデイエクスプロイト

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。

ゾンビアプリケーション

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

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