aws

での製品としてのインフラストラクチャの実装 (IaP) AWS

AWS 規範ガイダンス

AWS 規範ガイダンス: での製品としてのインフラストラクチャの実装 (IaP) AWS

Table of Contents

にiAPを実装するAWS

キルスティン・キスマイヤー、Amazon Web Services (AWS)

2023 年 1 月 (ドキュメント履歴)

このガイドでは、AWSインフラストラクチャを製品 (iAP) として管理する方法について説明します。IaP は、コードとしてのインフラストラクチャ (IaC) よりも高いレベルの抽象化と制御を実現しますが、その目的を達成するために IaC メソッドを使用します。このガイドでは、iAPを管理するためのツールについても説明し、各ツールがインフラストラクチャ管理の目標にどのように役立つかについても説明します。AWS サービスこのガイドの情報は、AWS Service Catalog非常に大規模な金融セクター企業の支援イニシアチブから得た教訓に基づいています。

このガイドは、さまざまな組織ユーザー、ビジネスユニット、AWS クラウドおよびサードパーティに必要に応じて簡単に割り当てて承認できる機能的なインフラストラクチャサービスを開発したいユーザーを対象としています。

インフラストラクチャを製品として管理する理由

インフラストラクチャリソースを製品として管理することの利点は、コンシューマー機能を標準化された定義と構成を持つリソースのセットとしてパッケージ化できることです。製品は、AWS組織が機能の割り当てと利用方法を管理および制御するための便利な方法を提供します。製品は、指定された組織単位 (OU) のみに制限される場合や、それらの機能を必要とする個人のみに制限される場合があります。AWS リージョン製品を特定のものに制限することもできます。

製品プロビジョニングモデルでは、製品の定義を一元的にカプセル化して更新することもできます。その後、製品のアップデートを 1 回限り、または定期的に配布できます。その実装は時間の経過とともに変化します。

ターゲットを絞ったビジネス成果

Organizations は常に、AWSインフラストラクチャを管理およびプロビジョニングするためのより良い方法を模索しています。目標には以下が含まれる場合があります。

- 高度な俊敏性、信頼性、フォールトトレランス、一元管理を実現し、単一の構成ポイントで進化する社内外の標準へのコンプライアンスを満たします。
- 特定のチームや個人が必要とするときにセルフサービスでアクセスできるようにしながら、インフラストラクチャを一元的に分散するためのロータッチまたはプッシュボタンメカニズム。

- 社内スタッフ、クライアントアカウント、AWSパートナーOUアカウントにインフラストラクチャとサービスをプロビジョニングできます。また、特定のリージョンの特定のインフラストラクチャコンポーネントにアクセスできる組織や組織を管理したい場合もあります。
- サードパーティのツール (など ServiceNow) やカスタムツールを使用して、企業資産やインフラストラクチャへのアクセスやプロビジョニングのリクエストを管理する場合、AWSインフラストラクチャとこれらのツールを簡単に統合できます。
- AWS数十または数百のターゲットアカウントに同時にインフラストラクチャをプロビジョニングできます。
- AWS複数のリソースをプロビジョニングして単一の機能を提供することをSupport します。
- 厳しいスケジュールの中で、必要なインフラストラクチャを備えた新しいアカウントを作成できます。
- プロビジョニングしたインフラストラクチャのインベントリにアクセスでき、インフラストラクチャコンポーネントを更新または削除できます。
- プロビジョニングとメンテナンスのプロセスをより簡単に、より速く、より安全で信頼性の高いものにするアプローチとテクノロジー。

AWS Service Catalogアプリ内課金の管理に使用

AWSは、AWS Service CatalogAWSインフラストラクチャの管理とプロビジョニングを製品としてサポートするというサービスを提供しています。Service Catalog を使用すると、プロビジョニングに必要なインフラストラクチャを製品セットとして迅速に定義し、それらの製品に対する権限を必要な人に付与し、個々の製品に必要なプロビジョニングと更新のパターンを実装できます。

Service Catalog はによってバックアップされていますAWS CloudFormation。Service Catalog のポートフォリオ、製品、 CloudFormation およびそれらのプロビジョニングテンプレートはスタックとして管理されます。これらのスタックは、次の 4 つの方法で定義できます。

- CloudFormation 標準テンプレートを使用する。
- AWS Cloud Development Kit (AWS CDK)およびService Catalog コンストラクトライブラリを、サポート対象の任意のプログラミング言語で使用します。
- サードパーティツールが提供するフレームワークを使用して、 CloudFormation スタックを記述する宣言型メタデータからスタック定義を生成します。
- Service Catalog API を使用する。この API は、製品のビルド以外のすべてのメソッドを提供します。ポートフォリオへの製品の追加、ポートフォリオからの製品の削除、製品とポートフォリオのタグ付け、管理上および運用上の製品サービスアクションの定義、ポートフォリオと製品定義の参照と検索を行うことができます。

Service Catalog 製品の核となるのは、(パラメータ化によって) まとめてカスタマイズ可能な機能を提供するように構成された 1AWS つ以上のリソースのセットです。たとえば、Service Catalog 製品を定義することで、ターゲットアカウントの Amazon Simple Storage Service (Amazon S3) バケットをプロビジョニングできます。S3 バケットは、バケット名、アクセスを許可するインターネットアドレスの範囲、バケットにアクセスできるユーザーセット、ライフサイクル階層化ポリシー、バケットのバージョン管理仕様などの入力パラメータを含む製品です。製品の一部としてバケットへのアクセスを提供するAWS Identity and Access Management (IAM) ロールを定義することもできます。

Service Catalog 製品は 1 つ以上のポートフォリオに追加できます。Service Catalog ポートフォリオは、一般的に同じ目的 (分析、開発、クライアントアクセスサービス、パートナーアクセスサービスなど) を果たすため、グループ化された製品の集まりです。

ユーザー、グループ、またはロールに、ポートフォリオレベルで製品をプロビジョニングするためのアクセス権を与える権限を与えます。プロビジョニングの場合、製品は起動 IAM ロール (ロールを引き受けることができる人なら誰でもセルフサービス方式で製品を起動する) か、製品をプロビジョニ

ングできる 1 つ以上のアカウントを定義するスタックセットに関連付けられます。スタックセットを使用するには、Service Catalog ハブアカウントでService Catalog 管理者ロールを定義し、スタックセットの各ターゲットアカウントでService Catalog 製品プロビジョニング実行ロールを定義する必要があります。

以下のセクションで、Service Catalog iAP (サービスカタログ) 機能がより詳しく説明されています。

トピック

- モジュール性とコード再利用のSupport
- Service Catalog で製品を定義するためのプログラミングオプション
- 外部のプロビジョニングプロセスとワークフローとの統合
- 製品プロビジョニング仕様
- DevSecOps ライフサイクルサポート
- カスタマイズされた再利用とアカウント固有のプロビジョニング
- Service Catalog 製品リソースをアプリケーションとして定義および管理する
- InInvenInven

モジュール性とコード再利用のSupport

AWSさまざまなリソースから製品を組み立てたり、他の製品から組み立てたりできます。理想的には、リソースをモジュール方式で定義して、複数の製品で再利用できるようにすることです。リソースレベルの再利用により、そのリソースタイプを使用するすべての製品ではなく、future 変更を 1 か所で行うことができます。

Service Catalog には、製品レベルでの再利用性をサポートするチェーニングと呼ばれる機能があります。製品を 1 つ以上の他の製品と連結できます。たとえば、S3 ロギングバケット製品をより高いレベルの監視製品にチェーン化したい場合があります。チェーニングはモジュール化をサポートしていますが、依存関係を管理する必要があるため、操作がいくらか複雑になります。Service Catalog はチェーン製品間のバージョン管理を自動的に管理しないため、1 つの製品を変更しても、その製品に依存する他の製品に影響が及ばないようにすることはできません。チェーニングは慎重に使用し、バージョン管理と依存関係の維持を確実にする独自のメカニズムを開発してください。

Service Catalog では、 CloudFormation CloudFormation製品プロビジョニングテンプレートをスタックとしてネイティブにデプロイします。ただし、Service Catalog CloudFormation では製品スタックの導入にいくつかの制限があります。特に、Service Catalog のプロビジョニングでは、再利

用可能なスクリプトセグメントを挿入したり、 CloudFormation ネストされたスクリプト (またはスタック) CloudFormation `include` を複数のレベルに参照したりするマクロはサポートされていません。これらのService Catalog の制限により、 CloudFormation 再利用可能なテンプレートまたはコンポーネントから製品を定義する機能が制限されます。これは、 CloudFormationスタックをでネイティブに定義する場合の標準的なベストプラクティスです。

> **Note**
>
> Service Catalog では、 CloudFormation これらの構成を使用するプロビジョニングテンプレートを使用して製品を適切に定義できます。ただし、`include`マクロを使用したり、Service Catalog CloudFormation テンプレートに複数のレベルのスクリプトをネストしたりすると、プロビジョニング時にエラーが発生します。

これらの制限により、モジュラー型の再利用可能な製品を Service Catalog に実装することが難しくなる可能性があります。モジュール性が必要な場合は、を使用して製品とそのプロビジョニングテンプレートを実装するか、AWSLabs Service Catalog ToolsAWS CDK プロジェクトのプロビジョニングワークフローとエンジンを使用することを検討してください。どちらの方法についても、このガイドで後述する。

Service Catalog で製品を定義するためのプログラミングオプション

Service CatalogAWS を使用してインフラストラクチャをプロビジョニングする場合のプログラミングオプションには、 CloudFormationテンプレートとの 2AWS CDK つがあります。現在、Service Catalog 製品を定義するための宣言型またはコードなしのメカニズムはありません。

CloudFormation スクリプティング

AWS CloudFormationは、AWSインフラストラクチャをプロビジョニングするための、実証済みの真の IaC ネイティブスクリプト言語です。 CloudFormation スクリプトは、AWS Management ConsoleまたはVisual Studio Code (またはシンプルなテキストエディタ) などの開発ツールやAWS Command Line Interface (AWS CLI) を使用して開発できます。

詳細については、「CloudFormation ドキュメント」を参照してください。 CloudFormation テンプレートを使用して Service Catalog 製品を指定する方法の詳細については、AWS::ServiceCatalog::CloudFormation CloudFormation ドキュメントの製品リソースを参照してください。

によるプログラマティックアプローチAWS CDK

は、AWSさまざまなプログラミング言語を使用してインフラストラクチャを定義および管理するための、AWS CDKエレガントで強力なオブジェクト指向プログラミングフレームワークを提供します。を使用して、AWS CDKAWSオブジェクト指向のきめ細かいカスタマイズやクラスフレームワークの拡張を開発できます。AWS CDKは、より高度なインフラストラクチャのニーズに合わせてカスタマイズしたいユーザーで、AWS サービス必要なプログラミングスキルと経験を持っているユーザー向けです。

を使用して Service Catalog ソリューションを実装するにはAWS CDK、組み込みの Service Catalog クラスを使用して製品とポートフォリオを定義します。これらのクラスは、AWS CDK aws-cdk-lib.aws_servicecatalog モジュールによって提供されます。

AWS CDKを使用して、さまざまな方法で製品を実装できます。製品のプロビジョニングテンプレートをに記述しなくても済むように、また再利用性を維持するために、AWS CDKProductStackプロビジョニングテンプレートを表すクラスを使用することをお勧めします。 CloudFormation `ProductStack`インスタンスは、AWS CDKプログラムでリソースを追加するスタックです。たとえば、S3 バケット、IAM ロール、Amazon CloudWatch ログを追加できます。を呼び出して、`ProductStack`servicecatalog.CloudFormationProduct定義済みのインスタンスにプロビジョニングテンプレートとしてインスタンスを追加するとservicecatalog.CloudFormationTemplate.fromProductStack (<ProductStack instance>)、AWS CDK CloudFormationが自動的にテンプレートを生成します。

Amazon S3 製品の JavaProductStack 実装の例を次に示します。

```
import * as s3 from 'aws-cdk-lib/aws-s3';
import * as cdk from 'aws-cdk-lib';

class S3BucketProduct extends servicecatalog.ProductStack {
  constructor(scope: Construct, id: string) {
    super(scope, id);

    new s3.Bucket(this, 'BucketProduct');
  }
}

const product = new servicecatalog.CloudFormationProduct(this, 'Product', {
  productName: "My Product",
  owner: "Product Owner",
  productVersions: [
    {
```

```
      productVersionName: "v1",
      cloudFormationTemplate:
  servicecatalog.CloudFormationTemplate.fromProductStack(new S3BucketProduct(this,
  'S3BucketProduct')),
    },
  ],
});
```

AWS CDKには、継続的な統合や継続的なデプロイ (CI/CD) パイプラインが組み込まれています。これらの組み込みパイプラインとソフトウェア開発ライフサイクル (SDLC) プロセスは、独自のプロセス標準と目的に合わせてカスタマイズできます。

AWS CDKカスタムクラスは他のクラスから継承して特殊な機能を提供でき、クラスは他のクラスのインスタンスから構成できます。AWS CDK共有クラスフレームワークを使用して複数の Service Catalog 製品を実装する場合は、特に複数の開発チームにわたるバージョン管理や互換性への影響を考慮してください。変更に後方互換性があることを確認するか、ある製品に対してクラスを変更しても別の製品に影響を与えないように、バージョン管理スキームに従っていることを確認する必要があります。

詳細については、「AWS CDK ドキュメント」を参照してください。

外部のプロビジョニングプロセスとワークフローとの統合

AWSSDK API またはを使用してService Catalog コンポーネントを操作できますAWS CLI。AWSSDK Service Catalog API を使用すると、Service Catalog API 呼び出しを統合できる任意のツールからService Catalog 製品を管理できます。API は、Service Catalog の作成と管理のすべての側面をカバーします。たとえば、Terraformは、Launch WizardAWS でSDKService Catalog APIを呼び出すことにより、Service Catalog 製品の起動 (プロビジョニング) をサポートします。詳細については、AWSドキュメントの「TerraformAWS Service Catalog による製品の起動」を参照してください。

Service Catalog コマンドを呼び出して、AWS CLI Service Catalog に対してアクションを実行することもできます。サポートされているコマンドの詳細については、AWS CLIコマンドリファレンスの servicecatalog を参照してください。

製品プロビジョニング仕様

Service Catalog は、 CloudFormation CloudFormation プロビジョニングテンプレートで指定されたリソースのスタックセットデプロイとしてプロビジョニングプロセスを開始します。(AWS

CloudFormationテンプレートは直接作成することも、AWS CDKProductStackコンストラクトで生成することもできます)。Service Catalog 製品のプロビジョニングはクローズドプロセスなので、カスタマイズして事前または後処理ステップを追加したり、調整したりすることはできません。ただし、プロビジョニングテンプレートを変更して、 CloudFormation リソース仕様の形式でステップを追加することはできます。これらにはAWS Step Functions、AWS Lambdaまたはまたは、準備段階 (プロビジョニング中に使用する要塞ホストをセットアップするためのカスタムブートストラップなど) と事後手順 (要塞ホストの解体など) を実行するLambdaベースのカスタムリソースなどがあります。プロビジョニング前とプロビジョニング後の手順を実装するこの方法には、includeプロビジョニングテンプレートと同じネストスタックの制限が適用されます。

ターゲットアカウントは、組織単位 (OU) としてではなく、個別のアカウントとして指定できます。この制限を回避するには、カスタムリソースまたは関数を記述できます。ほとんどの組織は、アカウントの生成を自動化し、アカウントリストを手動で管理したくないため、個々のアカウントではなくOUに製品ポートフォリオをプロビジョニングします。

DevSecOps ライフサイクルサポート

現在、Service Catalog CloudFormation スクリプトを使用してプロビジョニングされた製品には、CI/CD プロセスのサポートが組み込まれていません。開発、テスト、ステージ、プロダクションなどのライフサイクル環境を通じて製品を開発、テスト、リリースするには、CI/CDAWS CodePipeline DevOps プロセスをまたはその他のツールで作成することをお勧めします。

AWS CDKには、このガイドで前述したように、製品の CI/CD サポートが組み込まれています。

カスタマイズされた再利用とアカウント固有のプロビジョニング

製品は、できるだけ多くの異なるカスタマイズされた目的で再利用可能にする必要があります。Service Catalog は、製品パラメータによる再利用をサポートします。これらのパラメータは、プロビジョニング時に製品への入力として指定できます。

AWS Systems Manager CloudFormationこれらのパラメータをテンプレートレベルでパラメータストア値として指定して、アカウント固有の値と OU 固有の値を適用することもできます。CloudFormation これはプロビジョニングテンプレート設計のベストプラクティスです。ターゲットアカウント内の指定されたパラメータの値は、製品のプロビジョニング時に適用されます。たとえば、サブネットパラメータをパラメータストア値として指定し、そのサブネットを製品プロビジョニング時に特定の OU アカウントに適用できます。 CloudFormation テンプレートパラメータとしてのパラメータストア値の詳細については、AWS CloudFormationドキュメントの「ダイナミックリファレンスを使用してテンプレート値を指定する」を参照してください。

Service Catalog 製品リソースをアプリケーションとして定義および管理する

AWS Service Catalog AppRegistryアプリケーション検索、レポート作成、管理機能を一元的に提供します。 AppRegistry アプリケーションには、Service Catalog CloudFormation から独立したスタックだけでなく、1 つ以上のプロビジョニング済み製品スタックを含めることができます。AWS アカウントデプロイターゲットとして定義したすべてのアプリケーションリソースコレクションをグループ化して表示できます。これらのアカウントは、開発、テスト、本番稼働用ライフサイクルアカウントです。

AppRegistry を使用してメタデータ属性をアプリケーションに関連付けることもできます。属性のセットを含む再利用可能な属性グループを割り当てることができます。その後、 AppRegistry または統合サービスを使用して、指定された属性を持つアプリケーションリソースを検索して処理できます。これらの統合サービスには以下が含まれます。

- Application Manager:AWS Systems Manager アプリケーションとクラスターのコンテキストで AWSリソースに関する問題を調査および修復します。
- AWS Resource Access Manager、AWSアプリケーションや属性グループを組織のプリンシパルと共有するため
- AWS と連携する AWS Resource Groups のサービス
- AWS Resilience Hub製品構造の発見とレジリエンス評価用
- AWS Service Management ConnectorJIRA ServiceNow、その他の一般的なツールへの接続を宣言して設定する

の詳細については AppRegistry、以下を参照してください。

- AppRegistry 管理者ガイド
- AWS Service Catalog AppRegistryブログ投稿を使用して、アプリケーションの可視性とガバナンスを向上させます。この記事では、 AppRegistry インフラストラクチャガバナンスの使用方法の概要と、インフラストラクチャをアプリケーションとして登録するコマンドラインの例を紹介します AppRegistry。
- アプリケーションマネージャーのブログ投稿を使用して AppRegistry 、アプリケーションを一元管理します。この記事では、LAMP Web AppRegistry アプリケーションをに登録するための申請方法、および Application ManagerAWS Management Console を使用して管理する方法の概要とチュートリアルを紹介します。

InInvenInven

Service Catalog には独自の内部在庫管理機能があり、製品共有とセルフサービスによってプロビジョニングされた製品を登録できます。ただし、製品でプロビジョニングされるリソースを管理するには、AWS ConfigAppRegistryまたはおよび関連サービスを使用することをお勧めします。これらのツールを使用すると、プロビジョニングされた Service CatalogAWS 製品を他のインフラストラクチャと一緒に管理するための、より包括的で統合されたアプローチが可能になります。 AWS Config コンソールまたはAWS SDK API を使用して、プロビジョニングされた製品のインベントリを作成し、アクションを実行できます。 AppRegistryは Application Manager と統合されており、Service Catalog でプロビジョニングされた製品のインベントリ管理も行います。

AWS Service Catalogツールを使う

カスタマイズしたプロビジョニングワークフローで IaC 製品をより宣言的な方法でプロビジョニングしたい場合は、Service Catalog 機能の一部を補強するとよいでしょう。 AWSには、これらの要件をサポートするためのツールがいくつか用意されています。AWSLabs プロジェクトでは、Service Catalog Puppet とService Catalog ファクトリーという 2 つの一般的なツールが提供されています。

トピック

- Service Catalog
- Service Catalog

Service Catalog

Service Catalog パペットはAWS Boto3 API を使用して Python で実装されています。このツールには、Service Catalog 製品の設定とプロビジョニングを行うための強力な機能がいくつか用意されています。開発者は、マニフェストとして機能する YAML テンプレートを使用して Service Catalog の製品とポートフォリオのプロビジョニング情報を設定できます。Service Catalog Puppet プロビジョニングワークフローは、Service Catalog よりも複雑なデプロイプロセスを必要とする製品をサポートします。また、パフォーマンスの最適化もサポートしているため、厳しい時間枠内で製品を大規模にプロビジョニングできます。

Service Catalog Puppet は、デプロイ時に製品プロビジョニング用のService Catalog CloudFormation テンプレートにアクセスします。 CloudFormation 製品のプロビジョニングテンプレートスタックを直接呼び出してデプロイし、Service Catalog 独自のスタックセットプロビジョニングプロセスによって課される制限を回避します。 CloudFormation プロビジョニングテンプレートがマクロを使用して他のスクリプトを含めたり、 CloudFormationネストされたスクリプトを使用したりする場合は、プロビジョニングワークフローのブートストラップ部分でターゲットアカウントのそれらのスクリプトへのアクセスを提供する必要があります。

詳細については:

- Service Catalog Puppet GitHubのドキュメントとリポジトリを参照してください。
- Service Catalog Puppet SDK を使用してツールをプログラム的に操作し、製品およびポートフォリオのプロビジョニングを開始する場合は、SDK のドキュメントを参照してください。
- GitOpsは、Service Catalog Puppet 環境を管理するためのデフォルトのメカニズムです。

Service Catalog Puppet は開発者にとってかなり簡単に習得できます。製品プロビジョニングテンプレートの実装とマニフェストの実装には YAML テンプレートに精通している必要があります。CloudFormation自分のペースで進められるワークショップなど、新しい開発者を養うための優れたワークショップがあります。

プロビジョニングワークフローのSupport

Service Catalog Puppet は Python Luigi タスクオーケストレーションエンジンを使用して、ブートストラップとプロビジョニングのワークフローを実装します。これらのワークフローのすべてのステップは、Luigi ワークフロータスクとして実装されます。Luigiの概要と他の一般的なワークフローツールとの比較については、データ収益ブログの「Airflow vs Luigi、Argo、MLflow」の比較をご覧ください。 KubeFlow

Luigiを使用すると、Service Catalog Puppetはワークフロータスクに関連するワーカーの数を制御したり、ワークフローの他の側面を制御したりして、スケーリングとパフォーマンスを向上させることができます。Service Catalog Puppet には、製品とステップの依存関係を管理したり、製品のプロビジョニングを調整したりするための depends_on メカニズムも用意されています。この機能により、きめ細かな製品定義や複雑な依存関係を実装し、運用的に管理できます。

プロビジョニングモード

Service Catalog Puppet は、ハブ、スポーク、非同期の 3 つの実行モードをサポートしています。3 つのモードはすべて、Service Catalog ですでに定義されているポートフォリオ内の製品をプロビジョニングします。ターゲットアカウントとの Service Catalog 製品共有を利用し、Service Catalog 管理者とローンチロールを使用してそれらのターゲットでのプロビジョニングを実現します。Service Catalog Puppet は、YAML 設定ファイルで提供されるロール設定に基づいて、同じ組織内でブートストラップステップを実行します。このツールは、単一のハブアカウントから複数の組織へのプロビジョニングもサポートしています。このシナリオでは、Service Catalog Puppet が外部組織のアカウントで必要なプロビジョニングアクションを実行できるように、外部組織で手動でブートストラップを実行する必要があります。

すべてのプロビジョニングモードで、Service Catalog Puppet は Service Catalog のプロビジョニングプロセスを呼び出すことなく、製品のプロビジョニングを直接実装します。既存の Service Catalog スタックセット制約のロールとターゲットアカウントの仕様を使用するようにプロビジョニングマニフェストを設定できます。Service Catalog Puppet は、この情報を使用して Luigi ワークフローで独自のプロビジョニングを行います。

OU またはアカウントを直接指定することに加えて、アカウントタグ付けアプローチに基づいて製品ポートフォリオプロビジョニングのターゲットを定義できます。アカウントタグベースのプロビ

ジョニングでは、指定されたマニフェストプロビジョニングタグセットのすべてのタグを持つすべてのアカウントにポートフォリオ製品がプロビジョニングされます。たとえば、米国東部地域のすべての機関投資家にポートフォリオ商品を発行する場合は、、`type:prod partition:us-east`、というタグを指定できます`scope:institutional-client`。アカウントと OU の除外を宣言して、指定したタグが付いている OU やアカウント、または OU 指定のターゲットのメンバーであるアカウントへのプロビジョニングを禁止することもできます。アカウントタグ付けの詳細については、[Service Catalog ツールのドキュメントを参照してください](#)。

ハブモード

ハブプロビジョニングモードでは、スポークアカウントのすべての Luigi ワークフローは、指定された中央ハブアカウントから管理されます。ハブアカウントには、スポークアカウントでアクションを実行できる IAM ロールが割り当てられますが、タスクの管理はハブアカウント内から行われます。Hub アカウントは、すべてのスポークアカウントおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチおよびマルチ その後、最終ステータスを報告します。ハブアカウントモードは、最も古く、最も成熟したプロビジョニングモードです。ただし、プロビジョニングの同時実行性と速度を向上させるために、多くのユーザーがスポークプロビジョニングモードに移行しています。

スポークモード

スポークモードでは、Service Catalog ハブアカウントが Luigi ワークフローを開始し、指定されたブートストラップスポークアカウントで実行します。スポークワークフローが完了すると、ハブアカウントに通知されます。スポークアカウントで障害が発生すると、ハブアカウントまで上がります。ハブアカウントはスポークアカウントをポーリングして終了したかどうかを確認し、ステータスを判断します。

スポークモードでは、ほとんどすべてが別々のスポークアカウントで実行されるため、AWS サービスクォータを増やす必要はほとんどありません。また、スポークモードは、中央制御を維持しながら、ハブモードよりもはるかに優れた同時実行性を実現します。ハブモードに比べて、プロビジョニング速度を 800% 向上させることができます。スポークモードは、`DependsOn`製品間の関係による製品チェーニングをサポートします。これにより、依存している製品がすでにプロビジョニングされていることが保証されます。連鎖製品を含む製品では、コンポーネント連鎖製品をプロビジョニングすることもできます。AWS Lambda特殊な関数呼び出しを使用して必要な手順を実行することもできます。1つのスポークの障害は他のスポークから切り離されます。

スポークモードは、最大 7 つのリージョンに 980 を超えるアカウントを持つ企業で使用されます。これらの企業は通常、インフラストラクチャ内のすべてのリージョンとアカウントに1時間以内に製品をプロビジョニングできます。

> **Note**
>
> これらの結果は、ネットワークインフラストラクチャ、ワークロード、AWS組織のハブアンドスポークアカウントに設定されているクォータなどの要因によって異なる場合があります。また、プロビジョニングされる製品リソース、固有の作成時間、他のリソースへの依存関係にも依存します。

非同期モード

非同期モードでは、スポークアカウントでプロビジョニングワークフローが開始されますが、スポークからの完了応答を待ったり受信したりすることはありません。

キャッシュ

Service Catalog Puppet がワークフローの速度を最適化するために使用するもう 1 つのメカニズムは、ワークフローのステップを表す一般的なタスクをキャッシュすることです。キャッシュされたタスクは、その出力をAmazon Simple Storage Service (Amazon S3) に書き込みます。次回同じセッションで同じパラメーターを使用してタスクが呼び出されると、Service Catalog Puppet はタスクを再実行する代わりにキャッシュされた値を使用します。詳細については、Service vice vice vice vice Catalog Puppet ドキュメントの「キャッシュ」を参照してください。

DevSecOps ライフサイクルサポート

Service Catalog Puppet には、 DevSecOps パイプラインの管理に関するサポートが含まれています。Service Catalog Toolsのアクション（ Service Catalog Puppet の概要を参照 ）を使用してテストを自動化し、AWS推奨されるカナリアアカウントを含むライフサイクルアカウント全体で製品を宣伝できます。詳細については、Service Catalog Puppet ドキュメントの「環境の管理」を参照してください。

Service Catalog Puppet では、製品変更に関連する問題が本番環境で広く使用される前に確実に検出されるように、初期デプロイには少なくとも 1 つの Canary アカウントが必要です。新しいリリースをテストして自信がついたら、カナリア以外のプロダクションアカウントにプロモートできます。問題が見つかった場合は、リリースをロールバックして、問題が解決したら再導入できます。この方

法を使用する場合、問題のあるカナリアバージョンを本番アカウントにリリースすると、本番環境の問題が発生する可能性があります。別の方法として、変更を本番環境にリリースする前に、製品の変更ごとに完全なリグレッションテストを実行することもできます。これにより、CI/CD プロセスに追加のオーバーヘッドが発生しますが、本番環境の問題を回避するのに役立ちます。 DevSecOps 開発チームにとって最適な機能リリースシナリオとアプローチを決定するのは管理者の責任です。

Service Catalog Puppet を使用すると、複数のチームがService Catalog 製品ソリューションのプロビジョニングを同時に開発およびテストできます。ベストプラクティスとして、複数の開発者が同時に製品を変更しないでください。代わりに、製品をよりきめ細かいコンポーネントに分割して、個別に同時に修正することができます。

Service Catalog Puppet は、静的テスト機能とユニットテスト機能を提供するアサーションステートメントによるテストの自動化にも役立ちます。ポリシーシミュレーターを使用して、サービスコントロールポリシー (SCP) およびIAMポリシーをテストできます。 end-to-end これらは技術的にはテストですが、システム統合テスト (SIT) 環境でも使用できます。詳細については、Service Catalog Puppet ドキュメントの「ポリシーシミュレーションの使用」と「サービスコントロールポリシーの適用」を参照してください。

成熟度、完全性、サポート

Service Catalog Puppet は公式にはサポートされていませんがAWS サービス、広く採用されています。このツールはここ数年、大規模な組織で使用されており、希望するプロビジョニング期間内に数百の OU アカウントに製品を集中的にプロビジョニングすることに成功しています。フォールトトレラントな製品プロビジョニングを大規模に提供することが実証されています。Service Catalog Puppet で問題が発生したユーザーは、GitHub その問題をリポジトリに口グインさせて、このAWS Labs ソリューションへの貢献者が解決できるようにすることができます。

Service Catalog

Service Catalog ファクトリは、AWSラボが提供するもう1つのツールです。これに似ています。アカウントを生成し、Service Catalog を呼び出して (場合によっては Puppet を通じて) そのアカウント内でIAPをプロビジョニングします。AWS Control TowerService Catalog Puppet と同じメカニズムの多くを使用して機能を実装しています。Service Catalog ファクトリは、Service Catalog またはService Catalog Puppet を呼び出して、アカウント内の製品のインフラストラクチャをプロビジョニングできます。このツールは、AWS リージョン複数の組織でのアカウント生成もサポートしています。詳細については、Service Catalog GitHub ファクトリのドキュメントとリポジトリを参照してください。

レビューと以降のステップ

Service Catalog は、インフラストラクチャを製品として迅速かつ確実にプロビジョニングするのに役立ちます。定義済みの製品カタログからインフラストラクチャをセルフサービスすることも、hub-and-spoke モデル内の指定されたターゲットアカウントに製品をプッシュすることもできます。Service Catalog 製品とそのプロビジョニングテンプレートは、 CloudFormation スクリプトまたはを使用して定義できますAWS CDK。どちらの方法でも、Service Catalog CloudFormation は製品のプロビジョニングテンプレートを表すスタックを呼び出してデプロイすることで製品をプロビジョニングします。スタックは、 CloudFormation スタックセット内の指定されたすべてのターゲットアカウントにデプロイされます。

Service CatalogAWS CDK 開発のアプローチでは、定義済みの Service Catalog 製品クラスとポートフォリオクラス、および定義済みのリソースタイプを使用して製品とそのリソースを定義できるため、 CloudFormationより高度なモジュール化と再利用が可能になります。AWS CDK実装には、より高度なプログラミングスキルが必要です。これは、AWS組織がインフラストラクチャ開発の基盤として、標準化されたリソース構成と動作を備えた独自の再利用可能な製品フレームワークを確立したい場合に正当化される可能性があります。

Service Catalog Puppet とService Catalog ファクトリを使用して、主にプロビジョニングを目的としてService Catalog 機能を拡張できます。Service Catalog Puppet には、宣言型およびタグベースの製品プロビジョニング仕様、組み込み型のカスタマイズ可能な高性能な専用プロビジョニングワークフロー、カスタマイズ可能なアクションベースの CI/CD および SDLC パイプラインが組み込まれています。ワークフローの依存関係管理と組み込みのテスト自動化機能を使用することで、運用リスクを抑えて Service Catalog 製品をチェーン化できます。Service Catalog Puppetを使用すると、厳しい時間枠内で何百ものアカウントに確実に製品をプロビジョニングできます。Service Catalog ファクトリはに似ていますAWS Control Tower。アカウントを生成し、Service Catalog を呼び出して、それらのアカウント内でIAPをプロビジョニングします。

Service Catalog とService Catalog ツールには、AWSアプリ内課金の管理に役立つさまざまな機能があります。Service Catalog とこれらのツールは絶えず改善されています。最新の機能については、AWS Service CatalogAWS Service Catalog機能と製品リポジトリを参照してください。

リソース

リファレンス:

- Service Catalog 文書
- Service Catalog API
- AppRegistry
- AWS CloudFormation ドキュメント
- AWS CloudFormationスタックセット
- AWS::ServiceCatalog::CloudFormation製品リソース
- AWS Service Catalogテラフォームで製品を発売
- AWS Cloud Development Kit (AWS CDK)
- Service Catalog 構成ライブラリ
- AWS CDK ProductStackクラス
- AWS Organizations ドキュメント

Tools

- Service Catalog Puppet ドキュメンテーション
- Service Catalog Puppet GitHub リポジトリ
- Service Catalog ファクトリ文書
- Service Catalog GitHub ファクトリリポジトリ

AWS規範的なガイダンスパターン

- AWS Service CatalogAWS アカウント複数の製品を管理し、AWS リージョン
- AWS Service CatalogAWS アカウント製品を別の場所にコピーしてAWS リージョン
- AWS Service Catalogを使用してポートフォリオと製品の展開を自動化しますAWS CDK

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
初回刊行物	—	2023 年年年年年年年年年

AWS 規範的ガイダンスの用語集

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

数字

7 Rs

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

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

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

A

ABAC

「属性ベースのアクセスコントロール」を参照してください。

抽象化されたサービス

「 マネージドサービス」を参照してください。

ACID

「原子性、一貫性、分離性、耐久性」を参照してください。

アクティブ - アクティブ移行

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

アクティブ - パッシブ移行

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

集計関数

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

AI

「人工知能」を参照してください。

AIOps

「人工知能オペレーション」を参照してください。

匿名化

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

アンチパターン

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

アプリケーションコントロール

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

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

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)

組織がクラウドに正常に移行 AWS するための効率的で効果的な計画を立てるのに役立つ、 のガイドラインとベストプラクティスのフレームワークです。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを編成します。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、 AWS CAF は、組織が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 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

「事業継続計画」を参照してください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。エンディアンネス も参照してください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 1 つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーであるかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造。

ブルー/グリーンデプロイ

2 つの異なる同一の環境を作成するデプロイ戦略。現在のアプリケーションバージョンは 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、個人や組織に混乱を与えたり、損害を与えたりすることを意図しているものがあります。

ボットネット

マルウェアに感染し、ボットのヘルダーまたはボットオペレーター と呼ばれる、単一関係者の管理下にあるボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといいます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発したり、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブランチと呼ばれます。機能をリリースする準備ができたら、機能ブランチをメインブランチに統合します。詳細については、「ブランチについて (GitHub ドキュメント) 」を参照してください。

ブレークグラスアクセス

例外的な状況や承認されたプロセスを通じて、ユーザーが通常アクセス許可を持たない AWS アカウント にすばやくアクセスできるようにします。詳細については、 Well-Architected ガイダンスの「ブレークグラス手順の実装」インジケータ AWS を参照してください。

ブラウンフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウンフィールド戦略を導入する場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計します。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

AWS 「クラウド導入フレームワーク」を参照してください。

Canary デプロイ

エンドユーザーへのバージョンの低速かつ増分的なリリース。確信できたら、新しいバージョンをデプロイし、現在のバージョン全体を置き換えます。

CCoE

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

CDC

「データキャプチャの変更」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

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

CI/CD

「継続的インテグレーションと継続的デリバリー」を参照してください。

分類

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

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が に移行するときに通常実行する 4 つのフェーズ AWS クラウド :

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

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

CMDB

「設定管理データベース」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、 GitHub または が含まれます AWS CodeCommit。コードの各バージョンはブランチと呼ばれます。マイクロサー

ビスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオなどのビジュアル形式から情報を分析および抽出する AI の分野。例えば、 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、設定は想定した状態から変化します。これにより、ワークロードが非準拠になる可能性があり、通常は段階的かつ意図的ではありません。

構成管理データベース (CMDB)

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

コンフォーマンスパック

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

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性

の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「継続的デリバリーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

一元化された管理とガバナンスにより、分散型の分散型データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼できる ID のみが、期待されるネットワークから信頼できるリソースにアクセスしていることを確認できます。詳細については、「 でのデータ境界の構築 AWS」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

ディープラーニング

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

defense-in-depth

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

委任管理者

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

デプロイメント

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

開発環境

「環境」を参照してください。

検出管理

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

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

ディザスタ

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

ディザスタリカバリ (DR)

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

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

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

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

ベースライン設定からの偏差の追跡。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 ガバナンス要件への準拠に影響を与える可能性のあるランディングゾーンの変更を検出したりできます。

DVSM

「開発値ストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

エッジコンピューティング

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

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

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

エンタープライズリソースプランニング (ERP)

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

エンベロープ暗号化

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

環境

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

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

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、 AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。 AWS 移行戦略のエピックの詳細については、プログラム実装ガイド を参照してください。

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、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 の中央テーブル。事業運営に関する定量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 種類の列が含まれます。

フェイルファスト

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、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。

FGAC

「きめ細かなアクセスコントロール」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

段階的なアプローチを使用するのではなく、変更データキャプチャによる継続的なデータレプリケーションを使用して、可能な限り短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ディストリビューションの制限」を参照してください。

Gitflow ワークフロー

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

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 AWS Security Hub、GuardDuty、Amazon Inspector AWS Trusted Advisor、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

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。緊急性のため、通常、修正は一般的な DevOps リリースワークフローの外で行われます。

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

IIoT

「産業モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

既存のインフラストラクチャを更新、パッチ適用、または変更するのではなく、本番ワークロード用の新しいインフラストラクチャをデプロイするモデル。イミュータブルなインフラストラクチャは、本質的にミュータブルなインフラストラクチャ よりも一貫性、信頼性、予測性が高くなります。詳細については、 AWS Well-Architected フレームワークの「イミュータブルインフラストラクチャを使用したデプロイ」のベストプラクティスを参照してください。

インバウンド (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

接続、リアルタイムデータ、自動化、分析、AI/ML の進歩を通じて、 のビジネスプロセスのモダナイズを指すために 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。

産業分野における IoT (IIoT)

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

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。AWS 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、「AWS を使用した機械学習モデルの解釈」を参照してください。

IoT

「モノのインターネット」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロー

ドとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、安全でスケーラブルなマルチアカウント AWS 環境のセットアップ を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースのアクセスコントロール」を参照してください。

最小特権

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

リフトアンドシフト

「7 R」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。エンディアンネス も参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「ブランチ」を参照してください。

マルウェア

コンピュータのセキュリティまたはプライバシーを侵害するように設計されているソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスにつながる

可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

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

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステム。このシステムは、加工品を現場の完成製品に変換します。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整のために結果を検査する完全なプロセス。メカニズムとは、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 以外のすべて AWS Organizations。アカウントが組織のメンバーになることができるのは、一度に 1 つのみです。

MES

「製造実行システム」を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

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

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロ

イ、再利用可能なコード、回復力などがあります。詳細については、AWS 「サーバーレスサービスを使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

コンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。組織がクラウドへの移行のための強固な運用基盤を構築し、移行の初期コストを相殺するのに役立ちます。MAP には、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、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、スプリントに取り組む DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20～50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの移行ファクトリーに関する解説とCloud Migration Factory ガイドを参照してください。

移行メタデータ

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

移行パターン

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

Migration Portfolio Assessment (MPA)

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

移行準備状況評価 (MRA)

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

移行戦略

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

ML

「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

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

モノリシックアプリケーション (モノリス)

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

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

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

多クラス分類

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

変更可能なインフラストラクチャ

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

O

OAC

「オリジンアクセスコントロール」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

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

OI

「 オペレーション統合」を参照してください。

OLA

「運用レベルの契約」を参照してください。

オンライン移行

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

OPC-UA

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

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問および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「運用準備状況レビュー (ORR)」を参照してください。

運用テクノロジー (OT)

産業運用、機器、インフラストラクチャを制御するために物理環境と連携するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、Industry 4.0 トランスフォーメーションの主要な焦点です。

オペレーション統合 (OI)

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

組織の証跡

の組織 AWS アカウント 内のすべての のすべてのイベントを口グ AWS CloudTrail に記録する、によって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウントに作成され、各アカウントのアクティビティを追跡します。詳細については、 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 のすべての S3 バケット AWS リージョン、 AWS KMS (SSE-KMS) によるサーバー側の暗号化、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のアクセスを制限するオプションです。OAI を使用すると、 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。OACも併せて参照してください。OAC では、より詳細な、強化されたアクセスコントロールが可能です。

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用技術」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るネットワーク接続を処理する VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

個人を特定できる情報を参照してください。

プレイブック

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

PLC

「プログラム可能なロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

アクセス許可の定義 (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。`false`通常は `WHERE`句にあります。

述語のプッシュダウン

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に関する用語と概念内にあるプリンシパルを参照してください。

プライバシーバイデザイン

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

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

非準拠のリソースのデプロイを防止するように設計されたセキュリティコントロール。これらのコントロールは、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、 AWS Control Tower ドキュメントの「 コントロールリファレンスガイド」および「 でのセキュリティコントロールの実装」の「プロアクティブコントロール」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

設計、開発、発売から成長と成熟まで、製品のデータとプロセスのライフサイクル全体にわたる管理。

本番環境

「環境」を参照してください。

プログラム可能なロジックコントローラー (NAL)

製造では、マシンをモニタリングし、承認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。

パブリッシュ/サブスクライブ (pub/sub)

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

Q

クエリプラン

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

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設

定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

責任、説明責任、相談、通知 (RACI) を参照してください。

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。

再構築

「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 R」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョン は、耐障害性、安定性、耐障害性を提供するために、他の から分離され、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 R」を参照してください。

プラットフォーム変更

「7 R」を参照してください。

再購入

「7 R」を参照してください。

回復性

中断に耐えたり、中断から回復したりするアプリケーションの機能。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。

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R) 、説明責任

(A) 、協議 (C) 、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートを含めると、そのマトリックスは RASCI マトリックスと呼ばれ、サポートを除外すると RACI マトリックスと呼ばれます。

レスポンシブコントロール

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 (IdPs) が使用するオープンスタンダード。この機能により、フェデレーティッドシングルサインオン (SSO) が有効になるため、ユーザーは AWS Management Console

にログインしたり AWS API オペレーションを呼び出したりでき、組織内のすべてのユーザーを IAM で作成する必要はありません。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントのSAML 2.0 ベースのフェデレーションについてを参照してください。

SCADA

「監視コントロールとデータ収集」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

では AWS Secrets Manager、パスワードやユーザー認証情報など、暗号化された形式で保存する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 、プロアクティブ の 4 つの主なタイプがあります。

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修正するように設計された、事前定義されプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティス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

送信先にあるデータの、それを受け取る AWS サービス による暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。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 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 サービス エンドポイント」を参照してください。

サービスレベルアグリーメント (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

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

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、お客様はクラウドのセキュリティを担当します。詳細については、責任共有モデルを参照してください。

SIEM

「セキュリティ情報とイベント管理システム」を参照してください。

単一障害点 (SPOF)

システムを中断する可能性のあるアプリケーションの単一の重要なコンポーネントの障害。

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

split-and-seed モデル

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

SPOF

単一障害点 を参照してください。

star スキーマ

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

監視コントロールとデータ収集 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと生産オペレーションをモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

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

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

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

チューニング

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

ツーピザチーム

2 つのピザを食べることができる小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

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

未分化なタスク

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

上位環境

「環境」を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

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

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

現在のレコードに関連する行のグループに対して計算を実行する SQL 関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなどのタスクの処理に役立ちます。

ワークロード

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

ワークストリーム

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

WORM

「書き込み 1 回」を参照し、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。

Write Once, Read Many (WORM)

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