



で製造実行システム (MES) をモダナイズする AWS クラウド

AWS 規範ガイドンス



AWS 規範ガイド: で製造実行システム (MES) をモダナイズする AWS クラウド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

序章	1
アーキテクチャパターン	3
産業用エッジコンピューティング	3
アーキテクチャ	3
IIoT	4
アーキテクチャ	5
他のエンタープライズアプリケーションとのインターフェイス	6
アーキテクチャ	7
AI/ML	8
アーキテクチャ	9
データと分析	10
アーキテクチャ	11
コンピューティング用のコンテナ	13
アーキテクチャ	13
ステップの集約	14
MES をマイクロサービスに分解する	16
最適な専用テクノロジーの決定	19
コンピューティング	20
長時間稼働コンピューティング	21
コンテナ	21
イベント駆動型およびサーバーレスコンピューティング	21
データベース	22
リレーショナルデータベース	22
キー値、NoSQL データベース	22
時系列データベース	23
クラウドストレージ	23
ユーザーインターフェイス	24
マイクロサービスの統合アプローチの決定	25
同期通信	25
非同期通信	26
Pub/sub パターン	27
ハイブリッド通信	27
クラウドネイティブテクノロジーを使用してマイクロサービスを管理する	32
オーケストレーション	32

監査	33
回復性	35
可用性	35
ディザスタリカバリ	36
結論	38
リファレンス	39
AWS サービス	39
AWS サービスファミリー	40
その他の AWS リソース	40
作成者と寄稿者	41
ドキュメント履歴	42
用語集	43
#	43
A	44
B	47
C	49
D	52
E	56
F	58
G	60
H	61
I	62
L	65
M	66
O	70
P	73
Q	76
R	76
S	79
T	83
U	84
V	85
W	85
Z	86
.....	lxxxviii

での製造実行システム (MES) のモダナイズ AWS クラウド

Amazon Web Services ([寄稿者](#))

2024 年 4 月 ([ドキュメント履歴](#))

製造実行システム (MES) は、1970 年代に一連のデータ収集ツールと計画システムの拡張として始まりました。時間の経過とともに、これらは、加工品を現場の完成製品に変換する生産プロセスをモニタリング、追跡、文書化、制御するための包括的なソフトウェアソリューションに進化しました。MES は、プログラム可能なロジックコントローラー (PLCs)、監視制御およびデータ収集 (SCADA) システム、および履歴などの既存の現場システムと統合され、シームレスな生産管理を可能にします。また、エンタープライズリソースプランニング (ERP) や製品ライフサイクル管理 (PLM) システムなどのエンタープライズシステムと統合して、企業から現場への情報のシームレスなフローを可能にします。

クラウドコンピューティングでは、スケーラビリティ、柔軟性、パフォーマンス効率を向上させ、コストを削減するために、企業は MES をクラウドに移行することを検討しています。さらに、モノのインターネット (IoT)、人工知能と機械学習 (AI/ML)、マイクロサービスの出現により、MES の環境が破壊されています。従来のモノリシック MES をクラウドでホストすることに加えて、製造元や製造元にサービスを提供する独立系ソフトウェアベンダー (ISVs) には、マイクロサービスを使用してモジュラー MES を開発するオプションが追加されました。従来のモノリシック MES と最新の MES のどちらを選択するかは困難であり、組織の能力、予算配分、タイムラインの期待、ビジネス上の優先順位の徹底的な分析が必要です。APIs を使用する最新のクラウドネイティブのマイクロサービスベースの MES は、第 4 次産業改革 (Industry 4.0) の概念を活用する企業にとって推奨の選択肢です。これは、俊敏性、スケーラビリティ、柔軟性、価値実現までの時間を短縮し、IoT との互換性を実現するためです。

最新の MES にはいくつかの利点があります。

- アジャイル開発をサポートし、アプリケーション全体に影響を与えるのではなく、特定のサービスに変更を加えることで頻繁な更新をサポートし、進化するビジネスプロセスに適応します。
- マイクロサービスは、さまざまなプログラミング言語、データベース、ユーザーインターフェイステクノロジーを通じて、技術的な柔軟性を提供し、固有の要件に対応します。
- スケーラビリティを備えているため、地理的に分散したメーカーで、多様な生産プロセスがあるメーカーに適しています。
- 変化する顧客ニーズやサプライチェーンの中断に迅速に対応できるため、市場投入までの時間を短縮できます。

マイクロサービスベースの MES を採用することで、企業は Industry 4.0 の利点を活用できます。このガイドでは、サービスとテクノロジーを使用して AWS マイクロサービスベースの MES を実装するアプローチについて説明します。このアプローチでは、特定のビジネス成果に基づいてマイクロサービス構造を決定し、各成果に適したテクノロジーを選択します。このガイドでは、これらのマイクロサービスの統合、強化、モニタリング、管理の可能な方法を提案しています。マイクロサービスベースのアーキテクチャは運用が複雑になる傾向があります。したがって、このガイドでは、メーカーがマイクロサービスベースの MES の運用ガバナンスを簡素化する方法に関するベストプラクティスとアーキテクチャパターンについても説明します。利用可能なオプションを提示し、意思決定者に指示を与えます。意思決定の最終責任は、アーキテクト、アナリスト、テクノロジーリーダーにあります。これらのリーダーは、それぞれの状況、期待されるビジネス成果、利用可能なリソースに基づいて、最適なオプションを決定する必要があります。

このガイドの内容

- [最新のマイクロサービスベースの MES のアーキテクチャパターン](#)
- [MES をマイクロサービスに分解する](#)
- [MES に最適な専用テクノロジーの決定](#)
- [MES でのマイクロサービスの統合アプローチの決定](#)
- [クラウドネイティブテクノロジーを使用して MES のマイクロサービスを管理、オーケストレーション、モニタリングする](#)
- [MES の耐障害性](#)
- [結論](#)
- [リファレンス](#)
- [作成者と寄稿者](#)

最新のマイクロサービスベースの MES のアーキテクチャパターン

MES は、貴重なインサイトの活用、パターンの推測、イベントの予測、品質検査やデータ収集などの手動プロセスを自動化するために、産業用モノのインターネット (IIoT)、AI/ML、デジタルツインなどのクラウドネイティブテクノロジーを使用できます。最も一般的なユースケースとそのアーキテクチャパターンについては、以下のセクションで説明します。

- [産業用エッジコンピューティング](#)
- [IIoT](#)
- [他のエンタープライズアプリケーションとのインターフェイス](#)
- [AI/ML](#)
- [データと分析](#)
- [コンピューティング用のコンテナ](#)

これらのアーキテクチャに含まれるマイクロサービスの詳細については、このガイドの後半にある「[マイクロサービスへの MES の分解](#)」セクションを参照してください。

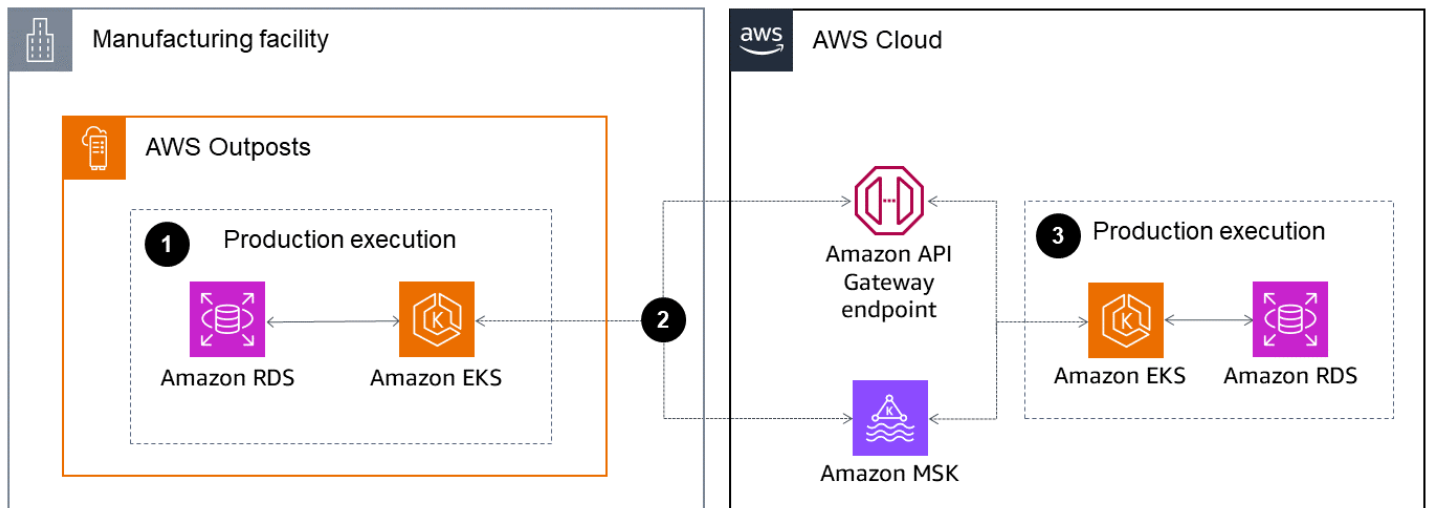
産業用エッジコンピューティング

MES は製造オペレーションにとって重要です。MES 内の一部のマイクロサービスまたは機能は低レイテンシーを必要とし、クラウドへの断続的な接続を許容できません。これらのマイクロサービスは、オンプレミスでの実行に適しています。[AWS エッジサービス](#)は、クラウドで提供されるインフラストラクチャ、サービス、APIs、ツールをオンプレミスのデータセンターまたはコロケーションスペースに拡張します。AWS エッジのサービスは、インフラストラクチャ、ストレージ、コンテンツ配信、堅牢で切断されたエッジ、ロボット、機械学習、IoT で利用できます。

アーキテクチャ

多くの MES トランザクションではレイテンシーが影響を受けます。このガイドで後述する例の 1 つは、本稼働実行サービスです。本稼働実行サービスの機能の 1 つは、work-in-progress フローをガイドすることです。これは機密性の高いアクティビティであるため、レイテンシーの許容度が低くなり、製造元がこのマイクロサービスのオンプレミスコンポーネントを必要とする可能性があります。

このユースケースのサンプルアーキテクチャを次に示します。



1. コンピューティング用の Amazon Elastic Kubernetes Service (Amazon EKS) とデータベース用の Amazon Relational Database Service (Amazon RDS) は、ローカルでホストされます AWS Outposts。セルフマネージドハードウェアを使用してエッジコンポーネントをホストすることもできます。Amazon EKS Anywhere などの一部の機能は、セルフマネージドハードウェアにも使用できます。

2. これらのサービスのエッジコンポーネントは、2 つのコンテナインスタンス間で Amazon API Gateway エンドポイントを介してクラウドコンポーネントと同期できます。

もう 1 つのオプションは、2 つのコンテナインスタンス間でサービスバスをセットアップして同期させることです。Amazon Managed Streaming for Apache Kafka (Amazon MSK) を使用して、このようなサービスバスを設定できます。

3. 製造元は、マイクロサービスのクラウドコンポーネントを使用して、プロセス改善のための PLM システムへの更新の送信、本番稼働用の ERP システムへの確認の送信、レポートと分析のためのデータレイクへのデータのエクスポートなど、レイテンシーの影響を受けにくいケースを処理できます。クラウドの経済性、スケール、ディザスタリカバリの利点により、メーカーはマイクロサービスのクラウドインスタンスにデータを長期間保存できます。

インダストリアル IoT (IIoT)

一般的な製造施設には、大量のデータを生成する何千ものセンサーとデバイスがあります。このデータのほとんどは使用されません。MES は、このデータをコンテキスト化し、クラウドネイティブサービスの助けを借りて使用できるようにします。MES は、マシンやデバイスに接続したり、プロセスパラメータやテスト結果などから情報を自動的に収集したり、イベントにリアルタイムで応答したり、時間を節約したり、手動入力によるエラーの可能性を排除したりすることもできます。例え

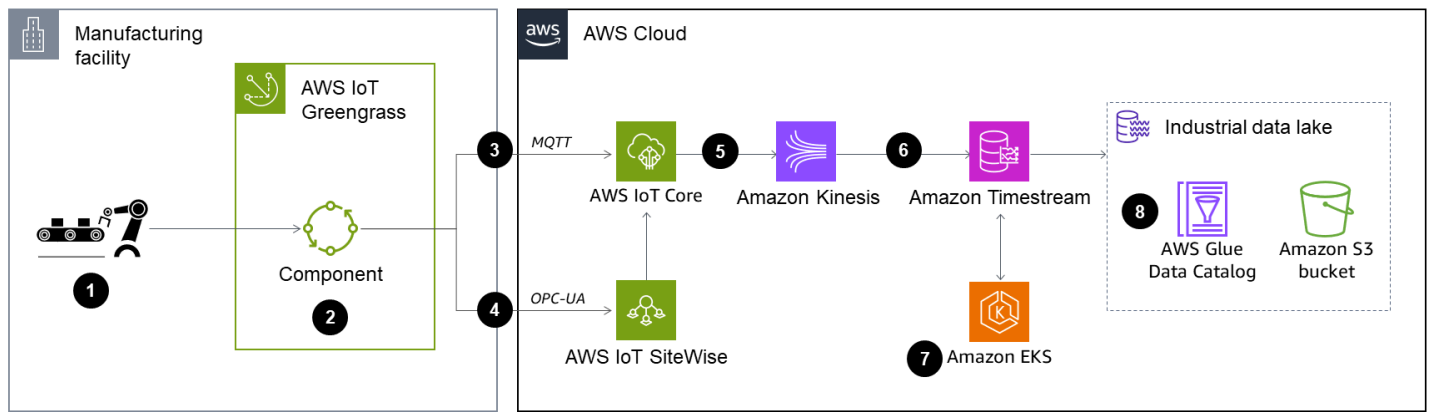
ば、手動データ入力なしで、テストマシンから結果を収集し、製品品質を判断し、不適合レコードまたは二次検査ワークフローを自動的に作成できます。時間の経過とともに、クラウドネイティブ IoT サービスは、欠陥の特定のパターンと根本原因を見つけるのに役立ち、製造プロセスを変更することで欠陥の発生を防ぐことができます。

AWS は、IoT データのロックを解除し、ビジネス成果を加速するための幅広いソリューションを提供しています。これらのソリューションには、顧客の固有のニーズに基づくアーキテクチャの構成要素である [AWS Partner ソリューション](#) と [AWS サービス](#) が含まれます。構成要素としてアーキテクチャに含めることができる AWS IoT サービスには以下が含まれます。

- [AWS IoT Greengrass](#) は、デバイスソフトウェアの構築、デプロイ、管理に役立つ IoT オープンソースのエッジランタイムおよびクラウドサービスです。エッジランタイムまたはクライアントソフトウェアはオンプレミスで実行され、さまざまなハードウェアと互換性があります。これにより、ローカル処理、メッセージング、データ管理、ML 推論が可能になり、アプリケーション開発を加速するための構築済みのコンポーネントが提供されます。は、レイテンシーの影響を受けやすいユースケースのために、MES のエッジコンポーネントとデータを交換 AWS IoT Greengrass できます。
- [AWS IoT Core](#) は、接続されたデバイスがクラウドアプリケーションやその他のデバイスと簡単かつ安全にやり取りできるようにするマネージドクラウドプラットフォームです。AWS IoT Core は、何十億ものデバイスと何兆ものメッセージを確実にサポートでき、それらのメッセージを処理して AWS エンドポイントやその他のデバイスにルーティングできます。を使用すると AWS IoT Core、アプリケーションは、接続されていない場合でも、すべてのデバイスを常に追跡し、通信できます。
- [AWS IoT SiteWise](#) は、産業企業が複数の産業施設にまたがる何千ものセンサーデータストリームを収集、保存、整理、視覚化できるようにするマネージドサービスです。には、施設内の現場にあるゲートウェイデバイスで実行され、履歴者や特殊な産業サービスから継続的にデータを収集し、クラウドに送信するソフトウェア AWS IoT SiteWise が含まれています。クラウドでこの収集されたデータをさらに分析し、ダッシュボードに使用することも、結果や傾向への対応のために MES にフィードすることもできます。

アーキテクチャ

一般的な IoT データインジェストおよび処理アーキテクチャは、固有の環境要因に基づいて多くの形状を取ることができます。最も一般的なユースケースは、ローカルネットワーク上のマシンからデータを収集し、このデータをクラウドに安全に送信することです。このユースケースのサンプルアーキテクチャを次に示します。



1. マシンまたはデータソース: これらは、ネットワークに接続され、独自のデータを共有できるスマートマシン、または PLCs や履歴などの他のデータソースである可能性があります。これらのソースから取得されるデータは、MQTT や OPC-UA などの異なるプロトコルにある可能性があります。
2. AWS IoT Greengrass は、データソースからデータを収集してクラウドに送信するコンポーネントを備えた Greengrass コアデバイスにインストールされます。
3. MQTT プロトコルのデータは に送られます AWS IoT Core。AWS IoT Core さらに、 は設定されたルールに基づいてこのデータをリダイレクトします。
4. OPC-UA プロトコルのデータは に送られます AWS IoT SiteWise。組織は、 AWS IoT SiteWise ポータルを使用してこのデータを視覚化できます。データは、コンテキスト化 AWS IoT Core のためにデータレイクに供給され、最終的には他のシステムのデータと結合されます。
5. Amazon Kinesis は、データを からストリーミング AWS IoT Core して保存します。AWS IoT Core には、他の とやり取りできる機能 ルール があります AWS のサービス。
6. Amazon Timestream データベースはデータを保存します。これは単なる例です。データの性質に応じて、他のタイプのデータベースを使用できます。
7. Amazon EKS は、マイクロサービス内の Kubernetes コントロールプレーンノードの可用性とスケーラビリティを管理します。
8. マシンやその他の運用技術 (OT) データソースから取り込まれたデータをデータレイクにフィードできます。

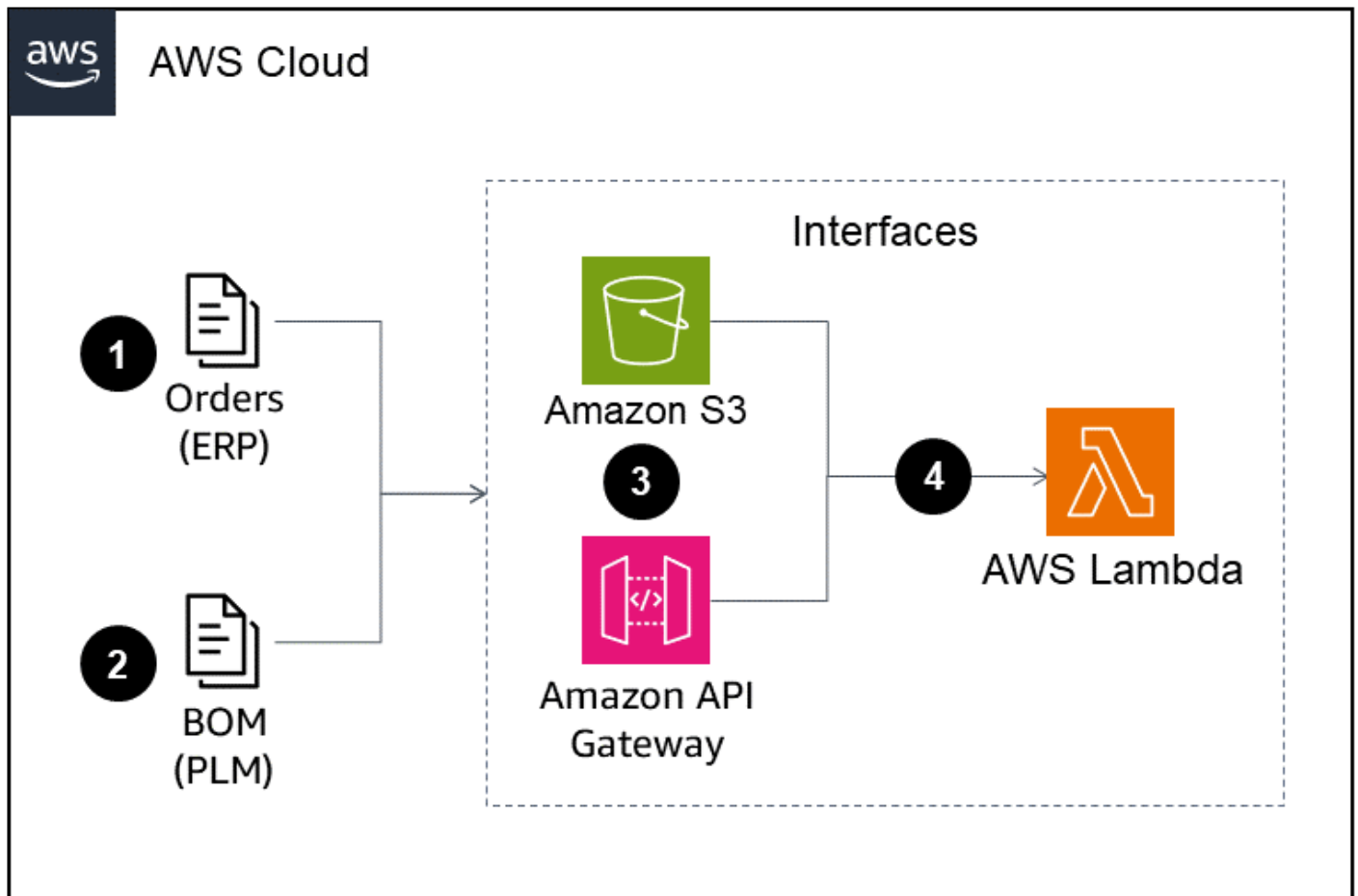
他のエンタープライズアプリケーションとのインターフェイス

MES は運用技術 (OT) と情報技術 (IT) のエッジにあるため、エンタープライズアプリケーションや OT データソースとやり取りする必要があります。組織のソリューションの状況に応じて、MES は

ERP とやり取りして、生産および発注書情報、部品と製品に関するマスターデータ、在庫状況、部品表を取得できます。MES は、注文のステータス、生産中の実際の品目と労働力の消費量、マシンのステータスについても ERP に報告します。PLM が存在する場合、MES は PLM を操作して、詳細な部品表 (BOP)、作業指示、場合によっては部品表 (BOM) を取得できます。MES は、プロセスの実行情報、不適合、BOM のバリエーションについても PLM に報告します。

アーキテクチャ

さまざまな PLM および ERP システムを考慮すると、このパターンの設計は MES が相互作用するシステムによって異なります。次の図は、アーキテクチャの例を示しています。



1. 組織には、AWS クラウド または他の場所に ERP インスタンスがある場合があります。
2. ERP と同様に、PLM システムは AWS クラウド または他の場所に存在する可能性があります。
3. 組織は、ERP と PLM から Amazon Simple Storage Service (Amazon S3) バケットにデータをインポートできます。これらのシステムがホストされている場合 AWS クラウド、ファイルポールの場合は別の S3 バケットである可能性があります、MES 用にレプリケートできます。これらのアプリ

ケーションに接続するもう 1 つの方法は、Amazon API Gateway を使用して API を使用することです。

4. 組織が ERP および PLM からデータをインポートする方法にかかわらず、AWS Lambda 関数は受信した情報を処理してマイクロサービスデータベースにデータをルーティングできます。これは、ERP および PLM インターフェイスと、このタイプのデータ処理が主にイベント駆動型であるためです。

人工知能と機械学習 (AI/ML)

MES、マシン、デバイス、センサー、その他のシステムによって生成されたデータに人工知能 (AI) と機械学習 (ML) を使用することで、製造オペレーションを最適化し、ビジネスに競争上の利点をもたらすことができます。AI/ML は、データをインサイトに変換します。このインサイトを使用して、プロアクティブにビジネスプロセスの最適化、マシンの予測メンテナンスの有効化、品質のモニタリング、検査とテストの自動化を行うことができます。AWS には、すべてのスキルレベルに対応する包括的な [AI/ML サービス](#) があります。機械学習へのアプローチには AWS 3 つのレイヤーがあります。やがて、重要なテクノロジー能力を持つほとんどの組織は 3 つすべてを使用します。

- 最下位レイヤーは、ML の専門家や実務者向けのフレームワークとインフラストラクチャで構成されています。
- 中間レイヤーは、データサイエンティストとデベロッパーに ML サービスを提供します。
- 最上位レイヤーは、ML モデルを構築したくないユーザー向けの人間の認識を模倣した AI サービスです。

以下に示しているのは、産業用の目立つ AWS ML サービスの一部です。

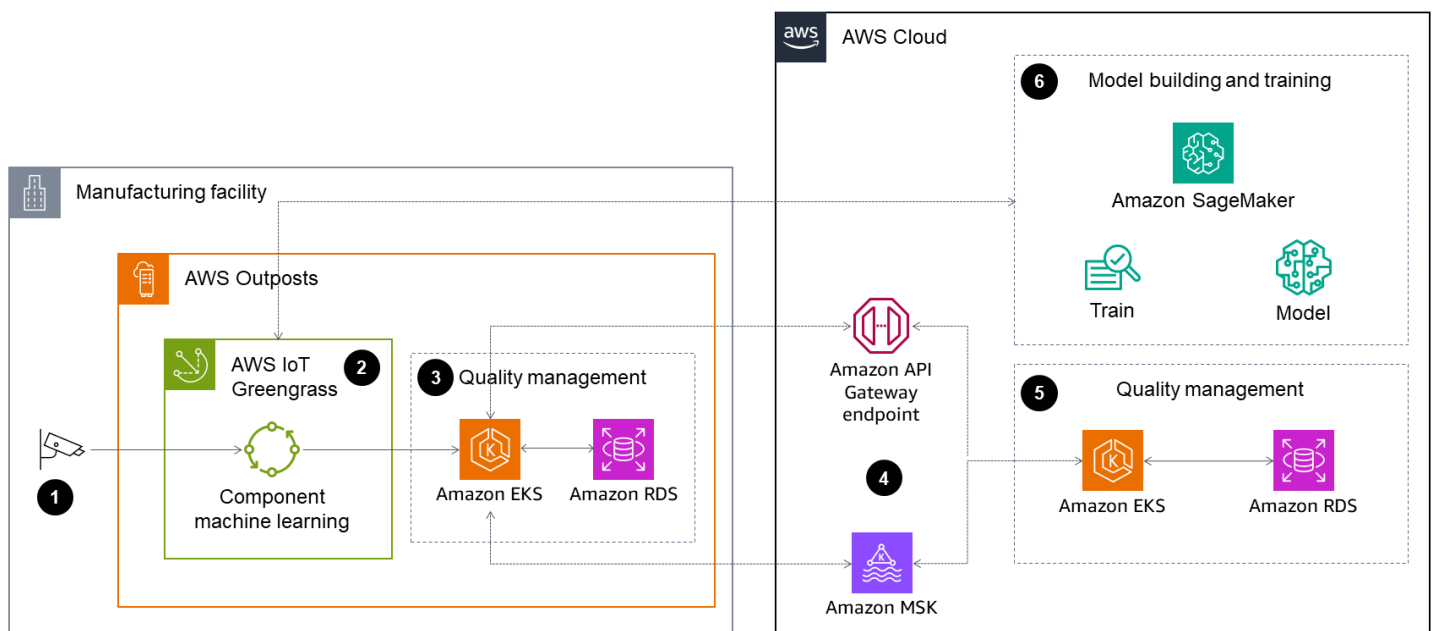
- [Amazon SageMaker AI](#) は、フルマネージド型のサービスであり、フルマネージド型のインフラストラクチャ、ツール、ワークフローを使用して、あらゆるユースケース向けにデータを準備し、ML モデルを構築、トレーニング、デプロイします。
- [AWS Panorama](#) は、オンプレミスカメラにコンピュータビジョン (CV) を追加して、高精度と低レイテンシーで自動予測を行う ML アプライアンスと SDK を提供します。を使用すると AWS Panorama、エッジのコンピュータパワー (ビデオをクラウドにストリーミングする必要なし) を使用してオペレーションを改善できます。は、製造品質の評価、産業プロセスのボトルネックの検出、施設内のワーカーの安全性の評価などのモニタリングタスクと視覚検査タスク AWS Panorama を自動化します。これらの自動タスクの結果を MES およびエンタープライズアプリケーションにフィード AWS Panorama して、プロセスの改善、品質検査計画、およびビルド時のレコードを取得できます。

① サポート終了通知

2026年5月31日、AWSはこのサポートを終了します AWS Panorama。2026年5月31日以降、AWS Panorama コンソールまたは AWS Panorama リソースにアクセスできなくなります。詳細については、[AWS Panorama 「サポート終了」](#) を参照してください。

アーキテクチャ

製造品質管理では、自動品質検査がコンピュータビジョンと機械学習の最も一般的なユースケースの1つです。メーカーは、コンベアベルト、ミキサーシュート、パッケージングステーション、ストックルーム、ラボなどの場所にカメラを配置して、ビジュアルを取得できます。カメラは、視覚的な欠陥や異常を高品質に把握し、メーカーがすべての部品や製品の最大 100% の検査を検査精度の向上で実行し、さらなる改善のためのインサイトを引き出すのに役立ちます。次の図は、自動品質検査の一般的なアーキテクチャを示しています。



1. ネットワーク上で通信できるカメラはイメージを共有します。
2. AWS IoT Greengrass はローカルでホストされ、イメージ内の異常を推測するコンポーネントを提供します。
3. 品質管理エッジサービスは、レイテンシーの影響を受けやすいユースケースのために、前のステップからの推論出力の結果をローカルで処理します。はコンピューティングリソースとデータベースリソースを AWS Outposts ホストします。製造元はこのコンポーネントアーキテクチャを

- 拡張して、推論結果に基づいてステークホルダーにアラートやメッセージを送信できます。製造元は、他の互換性のあるサードパーティーハードウェアを使用して、エッジでサービスをホストすることもできます。
- これらのサービスのエッジコンポーネントは、2つのコンテナインスタンス間で Amazon API Gateway エンドポイントを介してクラウドコンポーネントと同期できます。もう1つのオプションは、2つのコンテナインスタンス間でサービスバスをセットアップして同期させることです。Amazon Managed Streaming for Apache Kafka (Amazon MSK) を使用して、このようなサービスバスを設定できます。
 - 製造元は、マイクロサービスのクラウドコンポーネントを使用して、レイテンシーの影響を受けにくいケースを処理できます。例えば、品質検査を処理して履歴テーブルに入力したり、PLM システムに更新を送信して、将来のプロセスやパート設計の改善のための品質結果を取得したりできます。クラウドの経済性、スケール、ディザスタリカバリのメリットにより、お客様はクラウドマイクロサービスインスタンスにデータを長期間保存できます。
 - Amazon SageMaker AI などのクラウドネイティブ ML サービスを使用して、クラウドでモデルを構築およびトレーニングできます。最後にトレーニングされたモデルをエッジにデプロイして推論できます。エッジコンポーネントは、クラウドにデータを送り返してモデルを再トレーニングすることもできます。

データと分析

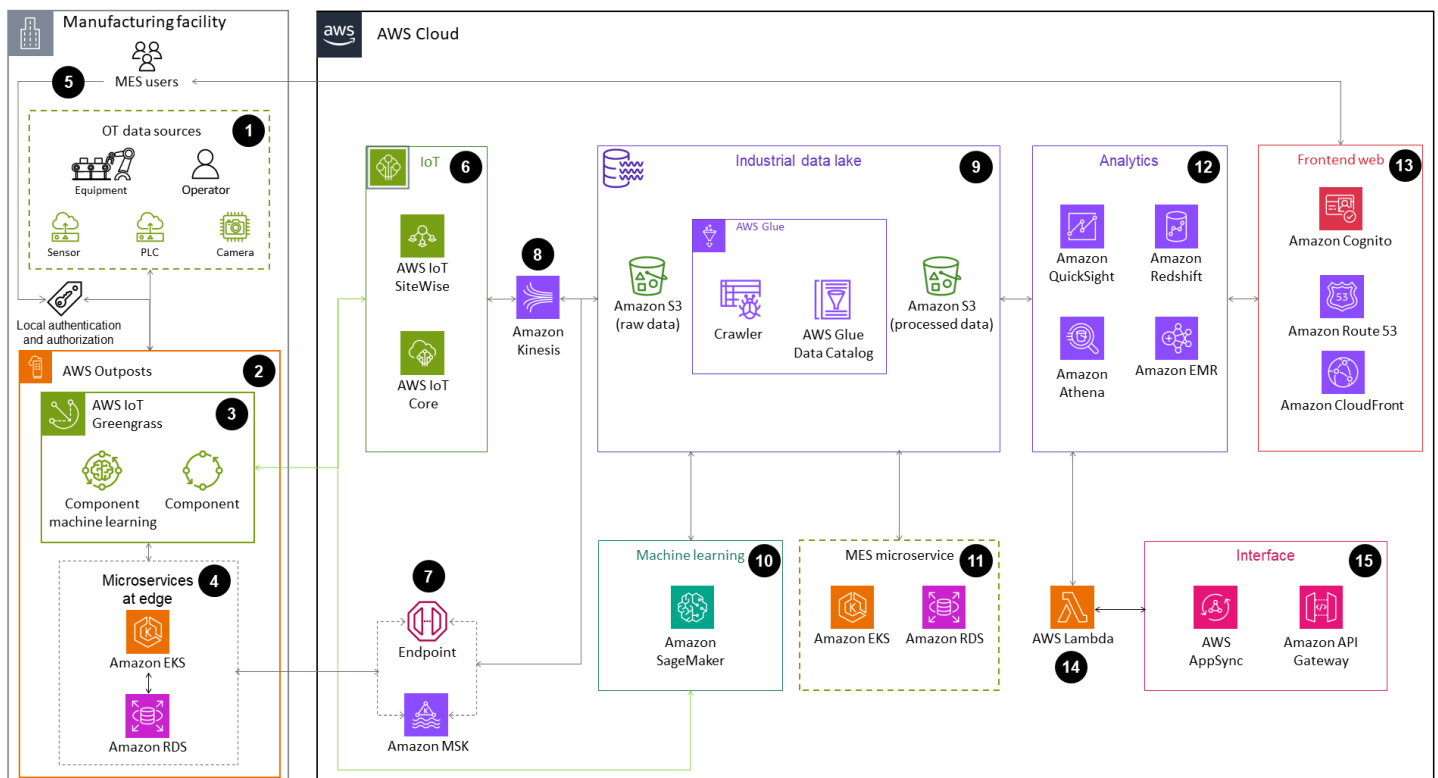
従来のモノリシック MES システムでは、分析機能が限られていたか、まったくありませんでした。メーカーは、毎日の生産、在庫レベル、品質結果などの基本的なレポートのために、高価なサードパーティー製ツールや、スプレッドシートへのバックエンドデータ抽出の複雑な方法に依存する必要がありました。分析のために MES データを他のアプリケーションやシステムデータと組み合わせる可能性はほとんどありませんでした。の AWS マイクロサービスベースの MES は、MES の一般的な分析課題を解決し、メーカーに競争上の優位性を与えるための追加の分析機能を提供します。AWS クラウドは、メーカーに一連の専用分析サービスおよび構築された分析プラットフォームからの選択肢を提供し、産業顧客向けに Industrial Data Fabric などの専用ソリューションも提供します。

- [AWS 分析サービス](#)は、ジョブに最適なツールを使用してデータインサイトをすばやく抽出することを目的として構築されており、ビジネスニーズに最適なパフォーマンス、スケール、コストを実現するように最適化されています。
- [Industrial Data Fabric](#) は、複数のデータソースからの大規模なデータ管理に役立ちます。企業は、MES データと製造全体のさまざまなシステムでサイロ化されたデータを組み合わせることで、バリューチェーンと関数全体のオペレーションを最適化できます。従来、製造内のシステムとアプリケーションは、階層に基づいて通信したり、強固に通信したりしません。たとえば、PLM

システムは SCADA や PLC などの OT システムと通信しません。したがって、本番稼働用とプロセス設計のデータは、これらのシステムが連携するように設計されていないため、結合されません。MES は 2 つを接続しますが、従来のモノリス MES もエンタープライズアプリケーションや OT システムとの通信が制限されています。の産業用データファブリックソリューションは、スケラブルで統合され、統合されたメカニズムがデータを効果的に使用できるようにするデータ管理アーキテクチャの作成 AWS に役立ちます。

アーキテクチャ

次の図は、IoT、MES、PLM、ERP のデータを組み合わせたデータと分析のサンプルアーキテクチャを示しています。このアーキテクチャは サービスでのみ構築されています AWS 。ただし、前述のように、データ分析に AWS Partner ソリューションを使用し、と AWS AWS パートナーのサービスを組み合わせることで、環境の固有の要件に対処できます。



1. 組み合わせる OT データソースは、ローカルネットワークで使用できます。
2. AWS Outposts はエッジハードウェアを提供します。
3. AWS IoT Greengrass サービスには、ローカル推論用の ML コンポーネントや、データインジェスト、処理、ストリーミングなどのその他のコンポーネントが含まれます。

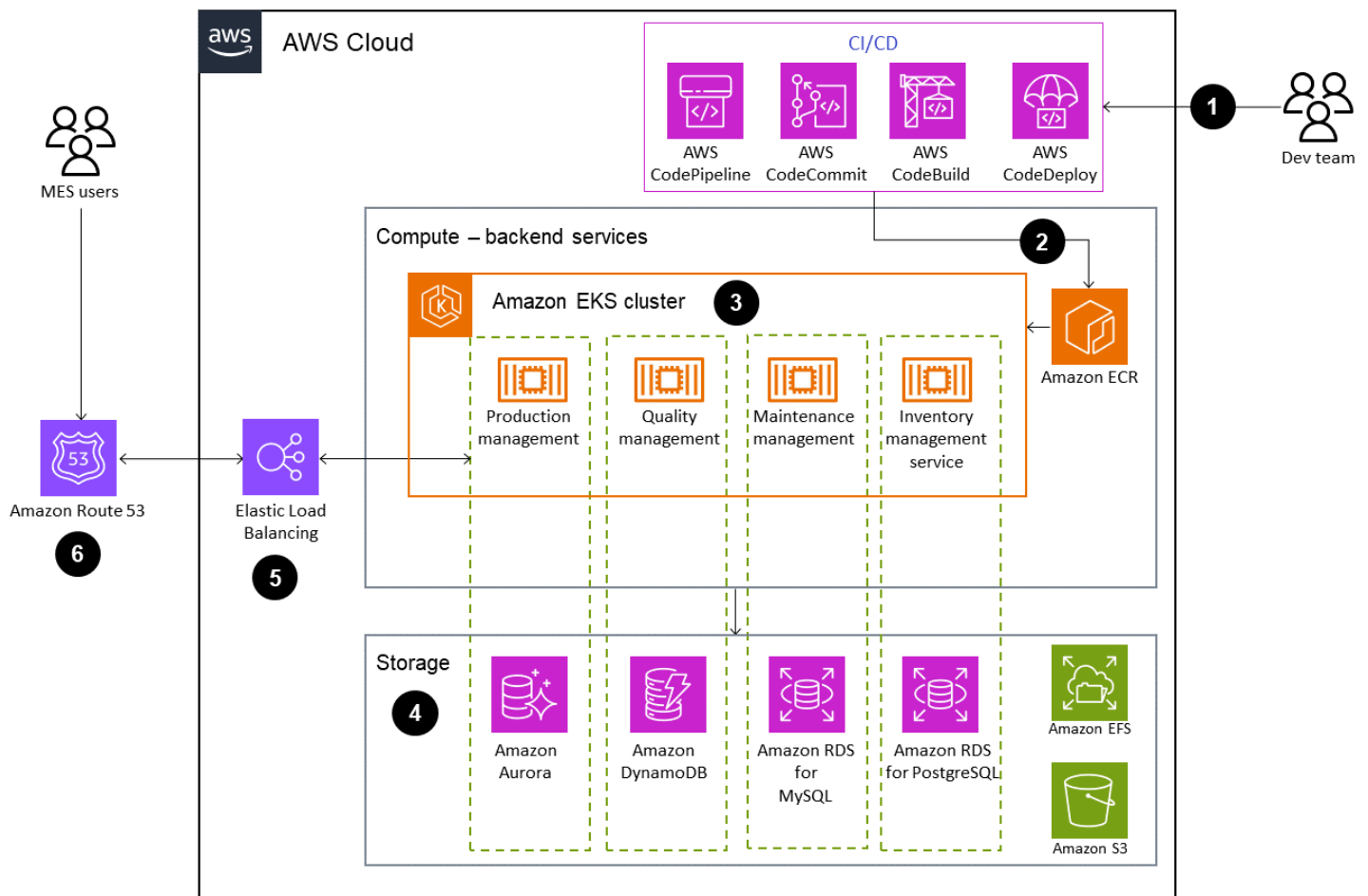
4. MES 用マイクロサービスのローカルインスタンスは、任意のマイクロサービスである可能性があり、要件に応じて、エッジに複数のマイクロサービスが存在する可能性があります。
5. ローカル認証と認可により、MES ユーザーは、リアルタイムの本稼働レポートなどのレイテンシーの影響を受けやすいユースケースや、接続が中断された場合に、ローカルマイクロサービスに安全にアクセスできます。
6. などの IoT サービスはクラウドでデータ AWS IoT Core を受信し AWS IoT SiteWise 、データを保存して処理します。
7. Amazon API Gateway エンドポイントと Amazon MSK オプションは、マイクロサービスのクラウドコンポーネントとエッジコンポーネントを同期させます。
8. Amazon Kinesis は IoT サービスから Amazon S3 バケットにデータをストリーミングします。Kinesis では、S3 バケットに保存する前にデータのバッファリングと処理を行うことができます。
9. 産業用データレイクには、S3 バケット、AWS Glue クローラ、および AWS Glue Data Catalog. AWS Glue crawlers が raw データを含む S3 バケットをスキャンしてスキーマとパーティション構造を自動的に推測し、処理されたデータを含む S3 バケットからの対応するテーブル定義と統計を Data Catalog に入力します。
- 10 Amazon SageMaker AI などの機械学習サービスは、データレイク内のデータを分析し、将来のイベントを予測するパターンを引き出すために使用されます。
- 11 MES マイクロサービスは、MES 内のマイクロサービスのクラウドコンポーネントで構成されません。
- 12 分析サービスは、データレイク、データウェアハウス (Amazon Athena)、ビジネスインテリジェンスサービスを使用したインタラクティブな視覚化 (Amazon Quick)、複雑なクエリを実行するオプションのクラウドデータウェアハウス (Amazon Redshift)、オプションの高度なデータ処理 (Amazon EMR) からのデータのサーバーレスクエリをサポートします。
- 13 フロントエンドウェブサービスには、ユーザーを認証する Amazon Cognito、DNS サービスとしての Amazon Route 53、低レイテンシーでエンドユーザーにコンテンツを配信する Amazon CloudFront などがあります。
- 14 AWS Lambda は、分析サービスと他のアプリケーション間のインターフェイスを有効にします。
- 15 インターフェイスサービスには、API を管理し APIs を統合してエンドポイントを作成 AWS AppSync するための APIs が含まれます。

コンピューティング用のコンテナ

コンテナは、マイクロサービスで構成される最新の MES の一般的な選択肢です。コンテナは、MES 開発者がアプリケーションをパッケージ化してデプロイするための強力な方法です。コンテナは軽量で、MES アプリケーションがどこでも実行およびスケーリングするための一貫性のあるポータブルソフトウェアを提供します。コンテナは、インターフェイス処理などのバッチジョブの実行、自動品質検査などのユースケース用の機械学習アプリケーションの実行、レガシー MES モジュールのクラウドへの移行にも推奨されます。ほとんどの MES モジュールは、コンピューティングにコンテナを使用できます。

アーキテクチャ

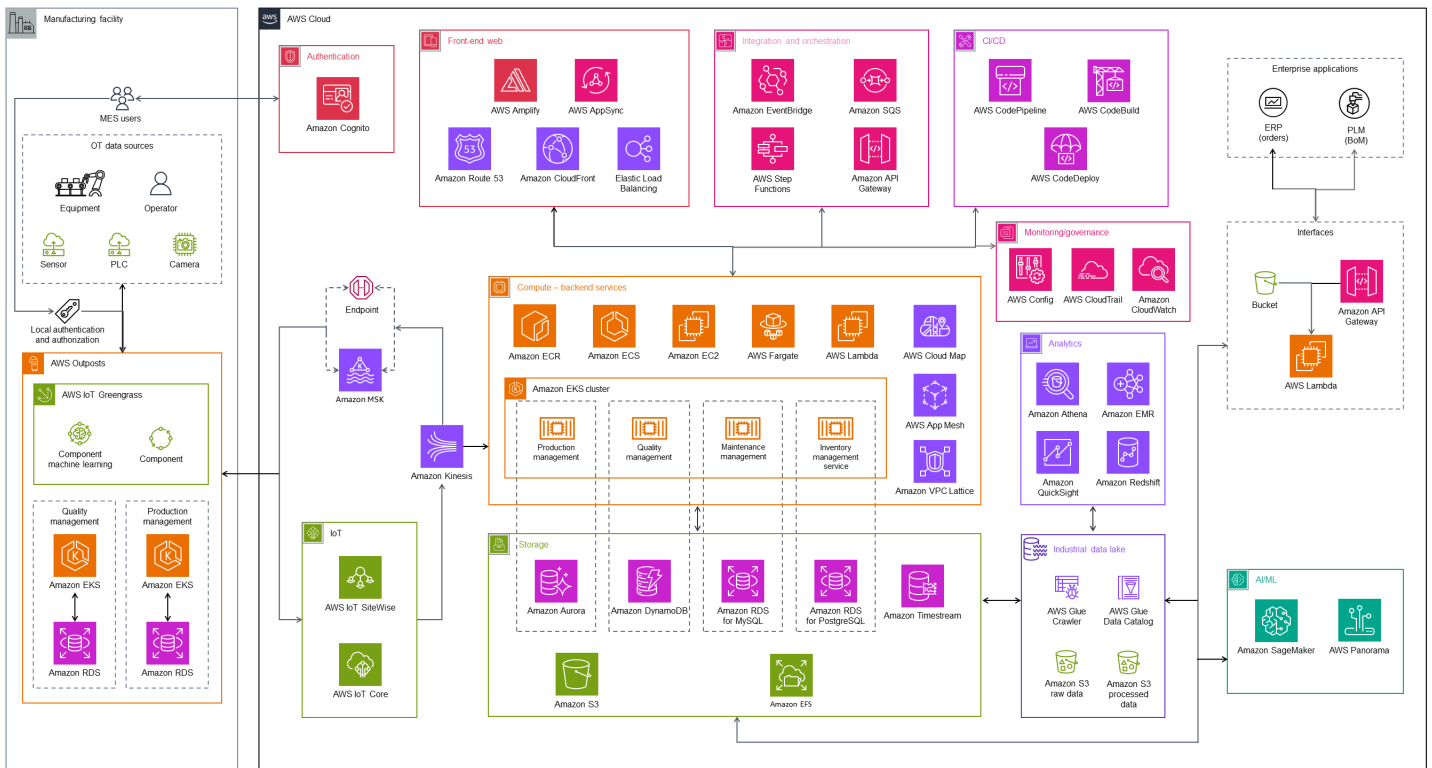
次の図のアーキテクチャは、DNS とロードバランシングを組み合わせ、バックエンドのコンテナ化されたコンピューティングで一貫したユーザーエクスペリエンスを実現します。また、継続的な更新のための継続的インテグレーションと継続的デプロイ (CI/CD) パイプラインも含まれています。



1. MES 開発チームは AWS CodePipeline を使用してコードを構築、コミット、デプロイします。
2. 新しいコンテナイメージは Amazon Elastic Container Registry (Amazon ECR) にプッシュされます。
3. フルマネージド Amazon Elastic Kubernetes Service (Amazon EKS) クラスターは、本番管理やインベントリ管理などの MES マイクロサービスのコンピューティング機能をサポートします。
4. AWS データベースとクラウドストレージサービスは、マイクロサービスの固有のニーズをサポートするために使用されます。
5. Elastic Load Balancing (ELB) は、1 つ以上のアベイラビリティーゾーンの複数のターゲットに MES モジュールの受信トラフィックを自動的に分散します。詳細については、Amazon EKS ドキュメントの「[ワークロード](#)」を参照してください。
6. Amazon Route 53 は、プライマリのロードバランサーへの受信リクエストを解決するための DNS サービスとして機能します AWS リージョン。

ステップの集約

成熟したマイクロサービスベースの MES アーキテクチャは、このガイドで説明されているすべてのユースケース、統合ツール、オーケストレーションサービスとアプローチを組み合わせています。ただし、アーキテクチャの詳細は、マイクロサービス、進化、MES の経時的な拡張の境界を決定するために使用される基準など、固有の環境要因によって異なる場合があります。次の図は、前のセクションで説明した使用シナリオを組み合わせた一般的なアーキテクチャを示しています。

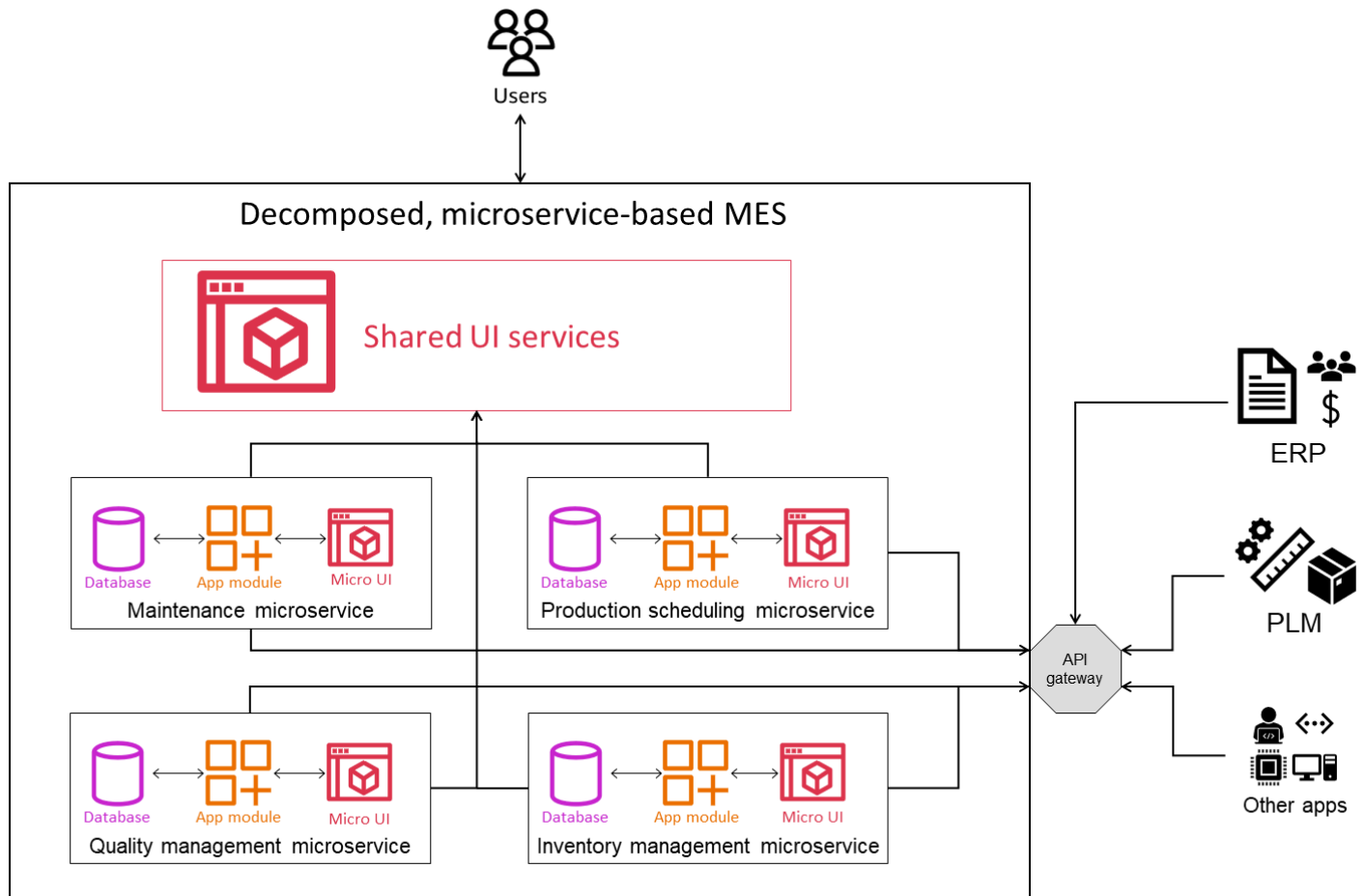


MES をマイクロサービスに分解する

通常、MES は組織のプロセスの固有の要件に合わせて広範なカスタマイズと設定を必要とするため、製造現場での MES のデプロイには数か月から数年の範囲があります。デプロイには、ワークフローのマッピングと設定、ユーザーロールとアクセス許可の定義、データ収集の設定、工場とエンタープライズシステムの統合、レポートと分析の要件の確立が含まれます。製造サイトは、作業プロセスを詳細に定義し、デジタル化して自動化できる構造にする必要があります。これには、大幅な組織変更、プロセスの再エンジニアリング、広範な再トレーニングが含まれる場合があります。問題や不一致を特定して対処するには、厳格なテストも必要です。これらの実装の課題、統合、および機能は、MES 実装を妨げる可能性があります。

all-in-one MES デプロイの実装上の課題を軽減するために、メーカーは段階的なアプローチを採用できます。まず、製造オペレーションに大きなメリットをもたらす限定された機能セットに優先順位を付けます。MES を、優先順位の高い要件に対応するようにカスタマイズされた、より小さく管理可能なマイクロサービスに分解します。その後、システムが成熟するにつれて、さらに多くの機能とマイクロサービスを徐々に追加します。このモジュラーアプローチは柔軟性を高め、製造ニーズに応じてターゲットを絞った改善を可能にします。これにより、よりスムーズで効果的な実装プロセスが可能になります。

次の図は、MES の必須マイクロサービスの例を示しています。



これらのマイクロサービスには以下が含まれます。

- 本番スケジューリングサービスは、作業指示を作成し、本番稼働をスケジュールします。他のシステムやマイクロサービスに接続して、本番稼働状況を追跡し、適切なリソース割り当てを確保できます。
- インベントリ管理サービスは、本番稼働に必要なインベントリレベルを追跡および管理します。また、本番稼働スケジューリングサービスに接続して、スケジュールされた本番稼働でインベントリが使用可能であることを確認することもできます。
- メンテナンス管理サービスは、機器の正常性のモニタリング、使用状況の追跡、予測メンテナンスアラートの作成、メンテナンスの追跡、メンテナンス履歴のキャプチャを行います。
- 品質管理サービスは、製品や品目の検査、品質保証などの品質管理活動を処理します。品質管理ワークフローの管理、テスト結果のキャプチャ、品質レポートの生成に役立ちます。また、本番稼働用スケジューリングサービスに接続して検査タスクをスケジュールし、在庫管理サービスに接続して品目の検査と追跡を行う場合もあります。

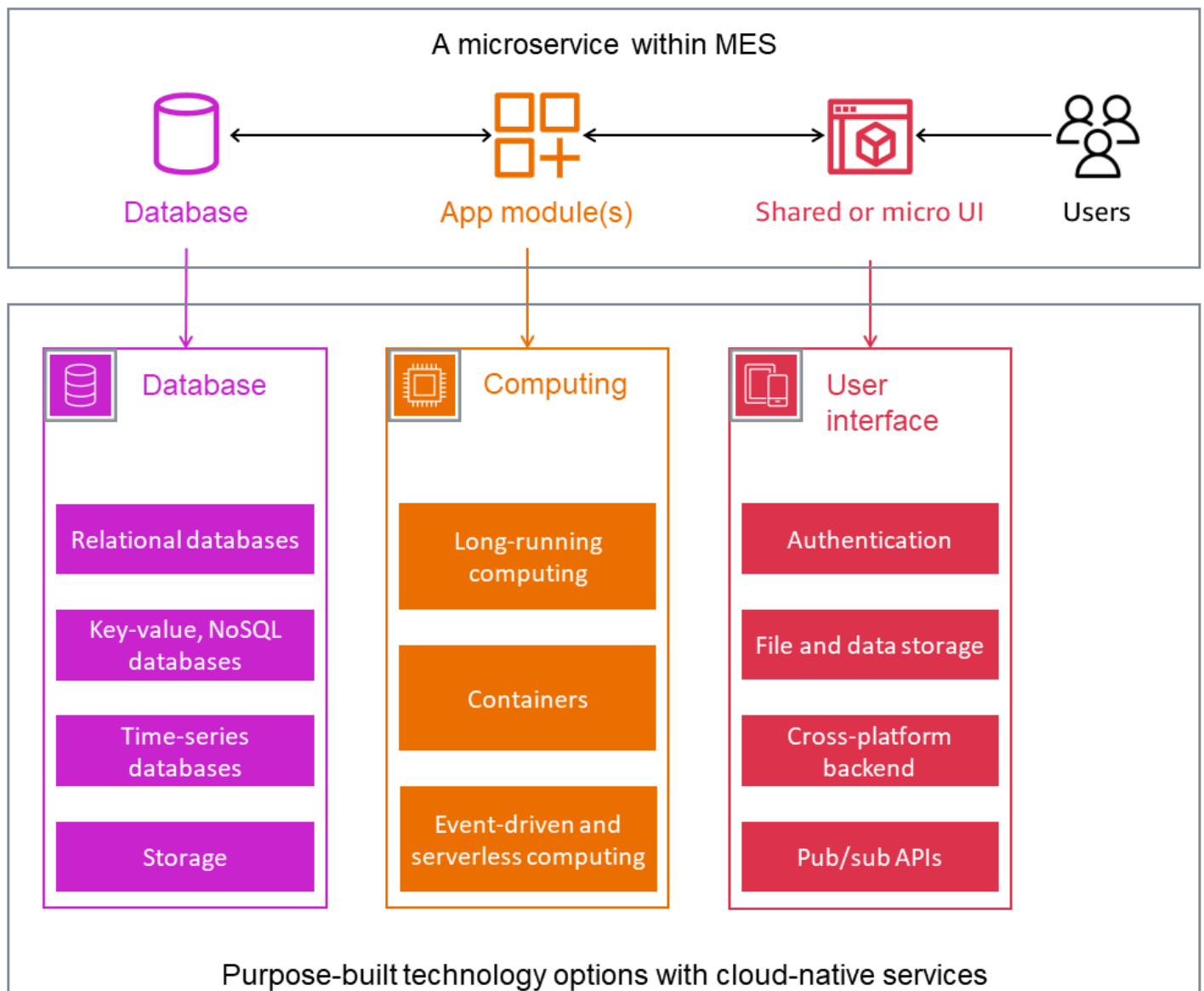
- 本番稼働サービスは、本番稼働注文の実行を管理し、本番稼働アクティビティを追跡します。マシン条件、オペレータアクション、マテリアル消費など、本番稼働に関連するすべてのデータをキャプチャします。また、生産注文に関する情報については本番稼働スケジュールサービス、品目の可用性と消費を追跡するためのインベントリ管理サービス、品質固有のワークフローについては品質管理サービスと接続することもできます。

製造オペレーション固有のサービスに加えて、サービススタック全体で共有関数を管理するには、標準サービスも必要です。共有サービスの例をいくつか示します。

- ユーザー管理サービスは、ユーザーの認証と認可を処理します。ユーザー関連のオペレーション用の API と、他のサービスのユーザーコンテキストを提供します。
- レポートおよび分析サービスは、他のサービスによって生成されたすべてのデータに対してレポートおよび分析機能を提供します。これにより、パフォーマンスのモニタリングが可能になり、メーカーはデータ主導型の意思決定を行うことができます。
- ユーザーインターフェイスサービスは、MES システムとやり取りするための標準ユーザーインターフェイスを提供します。他のサービスに接続してデータを取得し、コマンドを送信します。ユーザーがアプリケーションを設定して操作するためのダッシュボード、レポート、視覚化ツールを提供します。

MES に最適な専用テクノロジーの決定

MES をマイクロサービスに分解し、ビジネス成果への影響に基づいて開発に優先順位を付けた後、次のタスクは、特定のマイクロサービスとシステム全体のテクノロジースタックを決定することです。通常、MES、およびその本質的にマイクロサービスは、アプリケーションまたはコンピューティングレイヤー、永続性またはデータベースレイヤーを含む 2 層アプリケーションです。ユーザーインターフェイスは通常、すべてのマイクロサービス間で共有サービスです。UI のさまざまなコンポーネントは、マイクロサービスごとに一意にすることも、マイクロサービスごとに独自のマイクロ UI コンポーネントを持つこともできます。これらのマイクロサービスでは、コンピューティングとデータストレージの要件が異なるため、次の図に示すように、他のテクノロジースタックが必要になる場合があります。例えば、リレーショナルデータベースを使用した長時間実行型のコンピューティングは一部のマイクロサービスに最適ですが、イベント駆動型のオンデマンドコンピューティングや NoSQL データベースは他のマイクロサービスに適している場合があります。AWS は、各テクノロジーレイヤーに幅広いオプションを提供しているため、マイクロサービスの目的に基づいて最適なサービスを選択できます。



以下のセクションでは、コンピューティングとデータベースに使用できるオプションについて説明し、マイクロサービスの機能要件に基づいて適切なテクノロジーを選択する方法について説明します。

コンピューティング

従来、企業は常に インスタンス (長時間稼働コンピューティング) を使用してコンピューティングオペレーションを実行していました。インスタンスを使用すると、アプリケーションのすべてのリソースをボックスで取得できます。クラウドコンピューティングでは、複数のコンピューティング方法があります。従来の長時間実行されるコンピューティングに加えて、コンテナなど、より小さなコンピューティングユニットを使用できます。このユニットでは、より小さなマイクロサービスを構築し

で高速に移動してポータブルにしたり、サーバーとクラスターがすべてによって管理されるイベント駆動型のサーバーレスコンピューティングを実行したりできます AWS。

長時間稼働コンピューティング

MES 内のコンピューティング集約型で長時間実行されるマイクロサービスの中には、高パフォーマンスまたは永続的なコンピューティングリソースが必要です。例えば、PLM から受信した大規模な設計ファイルを処理する、機械学習モデルの品質検査の画像や動画を処理する、すべてのマイクロサービスからのデータを組み合わせることでデータ分析を実行する、履歴データに基づいてパターンを予測するために機械学習を使用するなどです。マイクロサービスが低レイテンシーのアプリケーションや、自動スケーラビリティ、幅広い OS サポート、ハードウェアサポートなどの機能に長時間実行されるコンピューティング能力を必要とする場合、[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) はクラウド内で安全でサイズ変更可能なコンピューティング容量を提供するサービスです。Amazon EC2 は、レガシーアプリケーションから継承され、すぐにモダナイズされることなくクラウドに移行されるアーキテクチャコンポーネントにも使用できます。

コンテナ

本番稼働スケジュール、本番稼働実行、品質管理など、MES 内のほとんどのマイクロサービスは、高性能コンピューティングを必要としません。これらのサービスはイベント駆動型ではありませんが、一貫して実行されます。このような場合、コンテナは、特に一貫したランタイム環境と効率的なリソース使用率が必要な場合に、移植性、分離性、スケーラビリティの利点により、マイクロサービスベースのアーキテクチャ内でリソースをコンピューティングするための最も一般的な選択肢の一つです。

コンテナがマイクロサービスのコンピューティング要件を満たすことができる場合は、Amazon Elastic Kubernetes Service (Amazon EKS) AWS や Amazon Elastic Container Service (Amazon ECS) などのコンテナ [オーケストレーションサービス](#) から使用できます。これらのサービスにより、基盤となるインフラストラクチャの管理が容易になり、安全なマイクロサービスの構築、適切なコンピューティングオプションの選択、信頼性の高い AWS 間の統合が可能になります。

イベント駆動型およびサーバーレスコンピューティング

マイクロサービスベースのアーキテクチャには、ERP および PLM からのデータの処理や、メンテナンスマネージャーまたはスーパーバイザーがフィールドにメカニズムをディスパッチするためのアラートの生成など、イベントに基づいて開始されるタスクが含まれます。は、オンデマンドでアプリケーションタスクを実行するイベント駆動型のサーバーレスコンピューティングサービスであるため、このような場合 [AWS Lambda](#) に適しています。Lambda では、ランタイムとサーバーの管

理や管理は必要ありません。Lambda 関数を作成するには、NodeJS、Go、Java、Python など、サポートされているいずれかの言語でコードを記述できます。サポートされている言語の詳細については、[Lambda ドキュメントの Lambda ランタイム](#)を参照してください。

データベース

従来のモノリシック MES では、主にリレーショナルデータベースが使用されていました。リレーショナルデータベースはほとんどのユースケースに適していますが、最適な選択肢はごくわずかです。マイクロサービスベースの MES を使用すると、マイクロサービスごとに最適な専用データベースを選択できます。では、[リレーショナルデータベース、時系列データベース、キーバリュースタイルデータベース、ドキュメントデータベース、インメモリデータベース、グラフデータベース、台帳データベース、現在 15 を超える専用データベースエンジンなど、8 つのデータベースファミリー](#) AWS を提供しています。以下は、MES 固有のマイクロサービスに適したデータベースの例です。

リレーショナルデータベース

一部の MES マイクロサービスは、データの整合性、アトミック性、一貫性、分離性、耐久性 (ACID) コンプライアンス、トランザクションデータの複雑な関係を維持する必要があります。たとえば、製品、BOMs、ベンダーなどとの作業指示の複雑な関係を保存するためにマイクロサービスが必要になる場合があります。リレーショナルデータベースは、このようなサービスに最適です。[Amazon Relational Database Service \(Amazon RDS\)](#) は、このようなすべてのニーズを満たすことができます。これは、クラウド内のデータベースのセットアップ、運用、スケーリングに役立つマネージドサービスのコレクションです。8 つの一般的なデータベースエンジン ([Amazon Aurora PostgreSQL 互換エディション](#)、[Amazon Aurora MySQL 互換エディション](#)、[Amazon RDS for PostgreSQL](#)、[Amazon RDS for MySQL](#)、[Amazon RDS for MariaDB for SQL Server](#)、[Amazon RDS for Oracle](#)、[Amazon RDS for Db2](#)) から選択できます。

キー値、NoSQL データベース

一部の MES マイクロサービスは、マシンまたはデバイスからの非構造化データとやり取りします。たとえば、フロアで実行されるさまざまな品質テストのテスト結果は、さまざまな形式で、合格/不合格値、数値、テキストなど、さまざまなタイプのデータが含まれる場合があります。一部のには、マテリアル分析でコンテンツまたはコンポジションテストをサポートするパラメータがある場合もあります。このような場合、リレーショナルデータベースの剛体構造が最適なオプションではない場合があります。NoSQL データベースの方が適している可能性があります。[Amazon DynamoDB](#) は、あらゆる規模で高性能アプリケーションを実行できるように設計された、完全マネージド型のサーバーレスのキーバリュースタイル NoSQL データベースです。

時系列データベース

機械やセンサーは、プロセスパラメータ、温度、圧力など、時間の経過とともに変化する値を測定するために、製造中に大量のデータを生成します。このような時系列データの場合、各データポイントはタイムスタンプ、1つ以上の属性、および時間の経過とともに変化する値で構成されます。企業はこのデータを使用して、アセットまたはプロセスのパフォーマンスと状態に関するインサイトを取得し、異常を検出し、最適化の機会を特定できます。企業は、このデータを費用対効果の高い方法でリアルタイムで収集し、効率的に保存する必要があります。これにより、データの整理と分析に役立ちます。従来のモノリシック MES では、時系列データを効果的に使用しません。時系列データ収集とストレージは、主に履歴やその他の下位レベルの OT システムの機能です。マイクロサービスとクラウドは、時系列データを使用し、他のコンテキスト化されたデータと組み合わせて、貴重なインサイトとプロセスの改善を引き出す機会を提供します。[Amazon Timestream](#) は、高速でスケーラブルなサーバーレスの時系列データベースサービスであり、リレーショナルデータベースのコストの 10 分の 1 のコストで、1 日あたり最大 1,000 倍速く、1 日あたり何兆ものイベントを簡単に保存および分析できます。時系列データを操作するもう 1 つのマネージドサービスは [AWS IoT SiteWise](#) です。これは、産業企業が複数の産業施設にまたがる何千ものセンサーデータストリームを収集、保存、整理、視覚化できるようにするマネージドサービスです。AWS IoT SiteWise には、施設内の現場にあるゲートウェイデバイスで実行され、履歴者または特殊な産業サーバーから継続的にデータを収集し、クラウドに送信するソフトウェアが含まれています。

クラウドストレージ

MES は、設計図、機械仕様、作業指示、製品のイメージと作業現場、トレーニングビデオ、オーディオファイル、データベースバックアップファイル、階層フォルダやファイル構造内のデータなど、多くの非構造化データ形式を処理します。従来、企業はこれらのタイプのデータを MES アプリケーションレイヤーに保存していました。クラウドストレージソリューションは、業界をリードするスケーラビリティ、データの可用性、セキュリティ、パフォーマンスを提供します。クラウドストレージの大きな利点は、実質的に無制限のスケーラビリティ、データの耐障害性と可用性の向上、ストレージコストの削減です。企業は、クラウドストレージサービスを使用して産業データレイク、分析、機械学習アプリケーションを強化することで、MES データをより適切に活用することもできます。は、[Amazon Simple Storage Service \(Amazon S3\)](#)、[Amazon Elastic Block Store \(Amazon EBS\)](#)、[Amazon Elastic File System \(Amazon EFS\)](#)、[Amazon FSx](#) などのストレージサービス AWS を提供します。マイクロサービスに適したストレージオプションの選択は、レイテンシーと速度、オペレーティングシステム、スケーラビリティ、コスト、使用状況、データ型の要件によって異なります。アーキテクチャの観点から、同じマイクロサービスに対して複数のオプションを選択することもできます。

ユーザーインターフェイス

MES ユーザーグループは多様です。これには、受領担当者、倉庫担当者、マテリアルハンドラー、マシンオペレーター、メンテナンスクルー、本番スケジューラ、本番マネージャーなどが含まれます。これらのユーザーとそのタスクは、MES のユーザーインターフェイス (UI) 設計に影響します。たとえば、オフィスのデスクから作業する店員の UI は、現場の手持ちデバイスを使用するマテリアルハンドラーの UI とは異なります。このさまざまな UI 要件によって、基盤となるテクノロジーの選択も決まります。マイクロサービスベースの MES アーキテクチャでは、UIs は頻繁にアップグレードされ、開発、配信、テスト、モニタリング、ユーザーエンゲージメントなど、独自のライフサイクルフェーズを経ます。AWS は、UI ライフサイクルフェーズの課題をサポートする [フロントエンドウェブとモバイル UI](#) の両方に幅広いサービスを提供します。UI ライフサイクルで使用される 2 つの目立つ AWS サービスは次のとおりです。

- [AWS Amplify](#) は、データストレージ、認証、ファイルストレージ、アプリホスティング、さらにはフロントエンドウェブやモバイルアプリでの AI や ML 機能のための一連のツールを提供します。リアルタイムおよびオフライン機能を使用して、iOS、Android、Flutter、ウェブ、または React Native アプリ用のクロスプラットフォームバックエンドを作成できます。
- [AWS AppSync](#) はサーバーレス GraphQL とパブリッシュ/サブスクライブ (pub/sub) APIs を作成し、単一のエンドポイントを通じてアプリケーション開発を簡素化し、データを安全にクエリ、更新、公開します。

MES でのマイクロサービスの統合アプローチの決定

マイクロサービスベースの MES では、データ交換、情報共有、シームレスなオペレーションを確保するために、service-to-service通信が不可欠です。MES マイクロサービスは、特定のイベントまたは一定の間隔でデータを交換できます。たとえば、ユーザーは、本番稼働確認トランザクション中に本番稼働数量を提供する場合があります。このようなトランザクションは、ERP への情報の送信、マシンの実行時間のキャプチャ、製品に関する品質情報のキャプチャ、労働時間の報告など、バックグラウンドで複数のトランザクションを開始できます。さまざまなマイクロサービスがこれらのタスクを担当する可能性があります。1つのイベントですべてのタスクが1つのマイクロサービスを通じて開始されます。

さらに、MES は外部システムと統合して、製造オペレーションの最適化、end-to-endデジタルスレッドの接続、プロセスの自動化を行います。マイクロサービスベースの MES を構築するときは、内部および外部サービスとの統合を処理する戦略を決定する必要があります。

次の機能パターンは、必要な通信のタイプに基づいて適切なテクノロジーを選択するためのガイドラインを提供します。

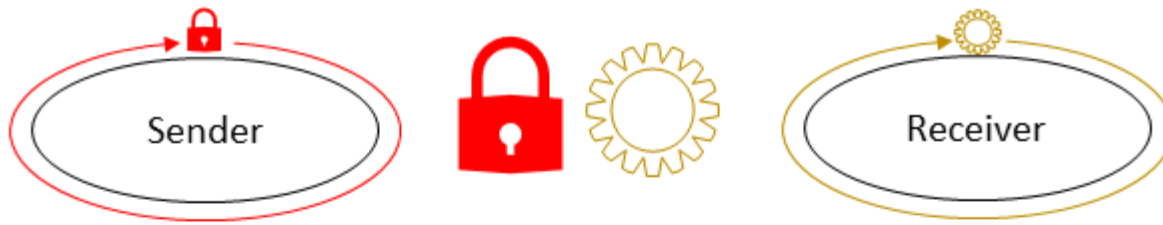
同期通信

同期通信パターンでは、呼び出し元のサービスはエンドポイントからレスポンスを受信するまでブロックされます。エンドポイントは通常、追加の処理のために他のサービスを呼び出すことができます。MES では、レイテンシーの影響を受けやすいトランザクションに同期通信が必要です。例えば、1人のユーザーが注文に対してオペレーションを完了する継続的な生産ラインを考えてみましょう。次のユーザーは、次のオペレーションでその注文がすぐに届くことを期待します。このようなトランザクションの遅延は、製品のサイクルタイムとプラントパフォーマンス KPIs に悪影響を及ぼし、追加の待機時間やリソースの使用率の低下を引き起こす可能性があります。

Stage 1: The sender sends a request to the receiver.



Stage 2: The sender remains blocked while the receiver is processing.



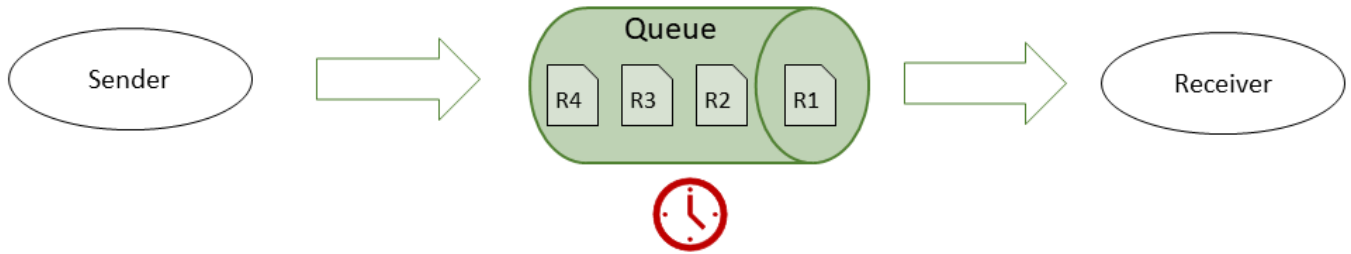
Stage 3: The sender is unblocked when the receiver sends a response.



非同期通信

この通信パターンでは、発信者はエンドポイントまたは別のサービスからの応答を待機しません。MES は、ビジネスランザクションに悪影響を及ぼさずにレイテンシーを許容できる場合、このパターンを採用します。たとえば、ユーザーがマシンを使用してオペレーションを完了すると、そのマシンの実行時間をメンテナンスマイクロサービスに報告できます。実行時間を更新してもすぐにイベントが開始されたり、オペレーションの完了に影響を与えたりしないため、この通信は非同期にすることができます。

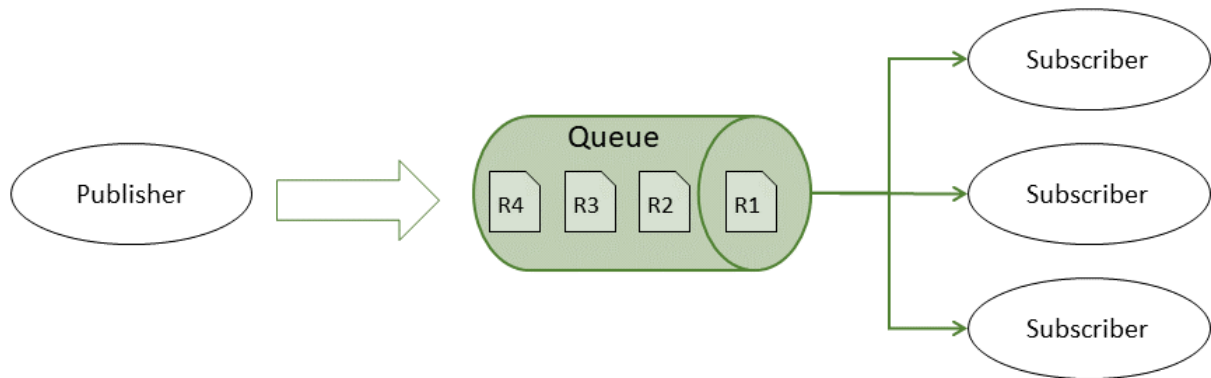
The sender sends a request to the queue and doesn't get blocked while the receiver is processing the request.



Pub/sub パターン

publish-subscribe (pub/sub) パターンは、非同期通信をさらに拡張します。相互依存通信の管理は、MES が成熟し、マイクロサービスの数が増えるにつれて困難になる可能性があります。リスンする必要がある新しいサービスを追加するたびに、発信者サービスを変更したくない場合があります。pub/sub パターンは、密接な結合なしで複数のマイクロサービス間の非同期通信を有効にすることでこれを解決します。このパターンでは、マイクロサービスはサブスクライバマイクロサービスがリスンできるチャンネルにイベントメッセージを発行します。したがって、新しいサービスを追加するときは、公開サービスを変更せずにチャンネルにサブスクライブします。たとえば、本稼働レポートやオペレーション完了トランザクションは、複数のログとトランザクション履歴レコードを更新する場合があります。マシン、労働、インベントリ、外部システムなどに新しいログ記録サービスを追加するたびにこれらのトランザクションを変更する代わりに、各新しいサービスを元のトランザクションのメッセージにサブスクライブし、個別に処理できます。

The sender sends a request to the queue. More than one receiver can subscribe to the queue.



ハイブリッド通信

ハイブリッド通信パターンは、同期通信パターンと非同期通信パターンを組み合わせたものです。

AWS は、複数の [サーバーレスサービス](#) を提供しており、さまざまな方法で組み合わせて目的の通信パターンを生成できます。次の表は、目立つ AWS サービスとその主要な機能の一部を示しています。

AWS サービス	説明	パターンをサポート		
		同期	非同期	Pub/sub
Amazon API Gateway	マイクロサービスが他のマイクロサービスからデータ、ビジネスロジック、または機能にアクセスできるようにします。API Gateway は、3 つの通信パターンすべてに対して同時 API コールを受け入れて処理します。	✓	✓	✓
AWS Lambda	サーバーを管理することなくコードを実行するサーバーレスのイベント駆動型コンピューティング機能を提供します。企業は Lambda を使用して、データベースやストレージ AWS サービスなどの他のサービス	✓	✓	✓

AWS サービス	説明	パターンをサポート		
		同期	非同期	Pub/sub
	間でデータを分離、処理、および渡すことができます。			
Amazon Simple Notification Service (Amazon SNS)	application-to-application (A2A) および application-to-person (A2P) メッセージングをサポートします。A2A は、分散システム、マイクロサービス、サーバーレスアプリケーション間で高スループットのプッシュベースのメッセージングを提供します。A2P 機能を使用すると、SMS テキスト、プッシュ通知、Eメールを使用してユーザーにメッセージを送信できます。		✓	✓

AWS サービス	説明	パターンをサポート		
		同期	非同期	Pub/sub
Amazon Simple Queue Service (Amazon SQS)	メッセージが失われたり、他のサービスを使用したりすることなく、任意のボリュームのソフトウェアコンポーネント間でメッセージを送受信できます。		✓	✓
Amazon EventBridge	コードを記述することなく、マイクロサービスまたはマイクロサービス内の AWS サービス内のデータの変更によって発生するイベントにリアルタイムでアクセスできます。その後、このイベントを受信、フィルタリング、変換、ルーティングし、ターゲットに配信できます。		✓	✓

AWS サービス	説明	パターンをサポート		
		同期	非同期	Pub/sub
Amazon MQ	メッセージブローカーのセットアップ、操作、管理を合理化するマネージドメッセージブローカーサービス AWS。メッセージブローカーを使用すると、多くの場合、さまざまなプラットフォームで異なるプログラミング言語を使用するソフトウェアシステムが、情報を通信および交換できます。			✓

詳細については、[「規範ガイド」ウェブサイトの 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。AWS

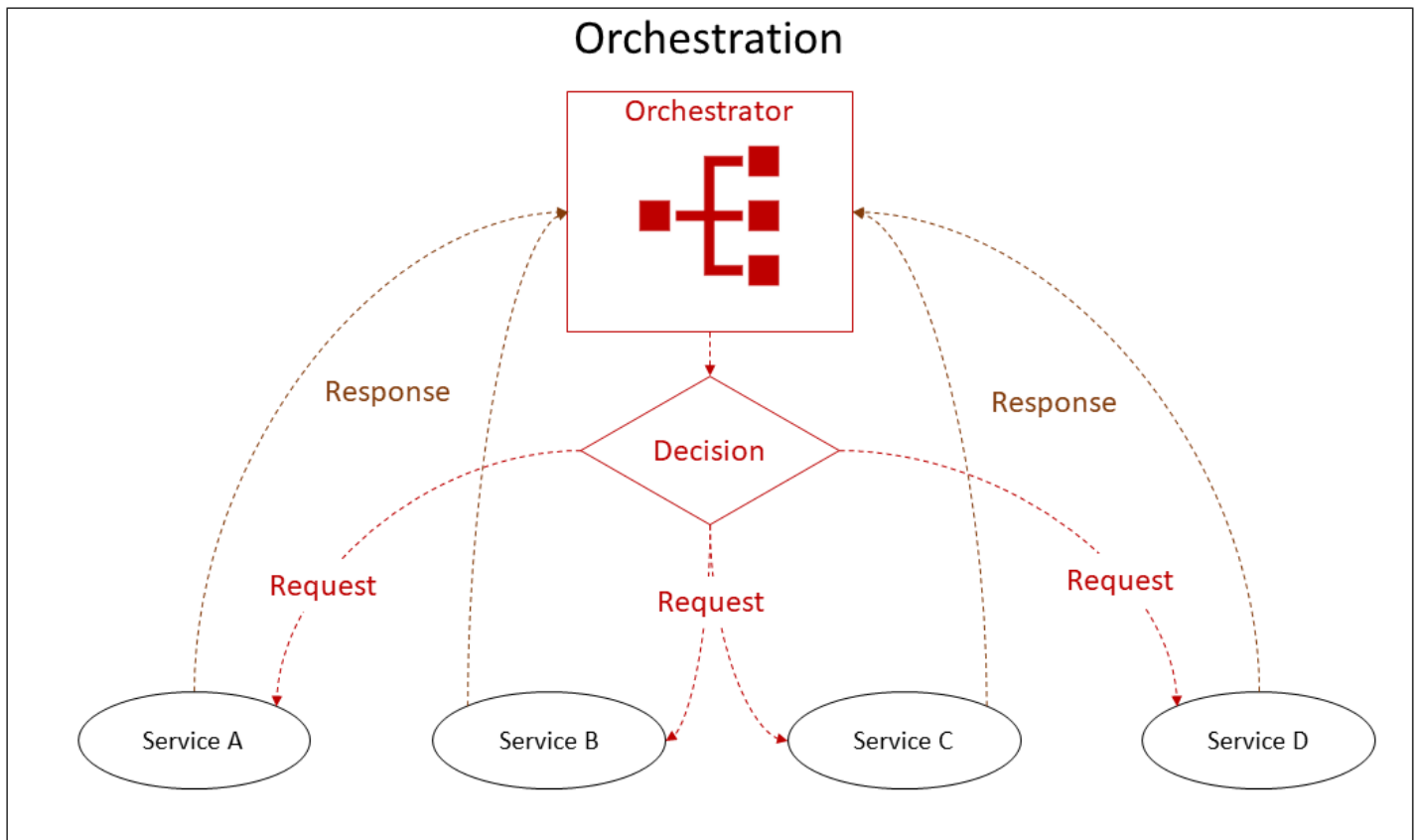
クラウドネイティブテクノロジーを使用した MES のマイクロサービスの管理、オーケストレーション、モニタリング

個々のマイクロサービスのアーキテクチャを設計したら、すべてのマイクロサービスがシームレスに動作することに焦点を当てる必要があります。マイクロサービスベースの MES は、コンテナイメージ、データベース、APIs、オブジェクトストア、キューなどの動的な分散コンポーネントを備えた、俊敏で進化し続けるシステムです。この継続的な変更は、これらの分散コンポーネントのオーケストレーション、モニタリング、管理におけるアーキテクチャ上の課題の別のセットをもたらします。

オーケストレーション

MES 内の一部のトランザクションには、オペレーションの完了の報告、発注書に対する在庫の受け取り、品質検査の完了などのタスクについて、本番稼働、品質、在庫、メンテナンス、その他の分野の複数のマイクロサービスが含まれる場合があります。これらのトランザクションには複数のサブトランザクションが含まれており、オーケストレーションが必要です。オーケストレーションコードは特定のマイクロサービス内に配置しないでください。ただし、上位のコントロールプレーンに表示されます。

このような複雑なオーケストレーションを簡素化するために、AWS は [AWS Step Functions](#) を提供します。このフルマネージドサービスを使用すると、ビジュアルワークフローを使用して、分散アプリケーションとマイクロサービスのコンポーネントを簡単に調整できます。次の図に示すように、アプリケーションのコンポーネントを一連のステップとして配置および視覚化するためのグラフィカルコンソールを提供します。視覚化された配置により、複数ステップのアプリケーションの構築と実行が容易になります。



監査

マイクロサービスベースの MES アーキテクチャは、継続的な変更と進化により動的です。組織は、コンプライアンスと規制のためにセキュリティおよびその他のエンタープライズポリシーを適用する必要があります。多くのユーザー、複数のマイクロサービス、各マイクロサービス内の多くのリソースを持つ MES などのシステム内でセキュリティポリシーとエンタープライズポリシーを確保するには、すべてのユーザーアクションとマイクロサービスのインタラクションを可視化する必要があります。

AWS は、監査とモニタリングの課題を解決するために以下のサービスを提供します。

- [AWS CloudTrail](#) は、ユーザーアクティビティと API の使用状況を追跡することで、監査、セキュリティモニタリング、運用上のトラブルシューティングを可能にします。CloudTrail ログは、AWS インフラストラクチャ全体のアクションに関連するアカウントアクティビティを継続的にモニタリングおよび保持し、ストレージ、分析、修復アクションを制御できます。
- [Amazon CloudWatch](#) は、AWS AWS クラウド リソースとアプリケーションのモニタリングサービスです。CloudWatch を使用して、リソース使用率、アプリケーションパフォーマンス、運用状

態をシステム全体で可視化できます。メトリクスの収集と追跡、ログファイルの収集とモニタリング、アラームの設定を行うことができます。

- [AWS Config](#) は、セキュリティとガバナンスのためのリソースインベントリ、設定履歴、および設定変更通知を提供します。を使用して、既存の AWS リソースを検出 AWS Config し、サードパーティリソースの設定を記録し、すべての設定の詳細を含むリソースの完全なインベントリをエクスポートし、リソースの設定方法をいつでも判断できます。
- [Amazon Managed Service for Prometheus](#) は、オープンソースの Prometheus データモデルとクエリ言語と互換性のあるメトリクスのサーバーレスモニタリングサービスです。ハイブリッドおよびマルチクラウド環境、オンプレミス、および AWS のコンテナワークロードをモニタリングしてアラートを生成します。

MES の耐障害性

耐障害性とは、MES システムがインフラストラクチャやサービスの中断から回復し、需要に合わせてコンピューティングリソースを動的に取得し、設定ミスや一時的なネットワーク問題などの中断を軽減できる機能です。Well-[AWS -Architected フレームワーク](#)の信頼性の柱が依存する主な要因は、回復性です。

障害耐性は、可用性とディザスタリカバリの 2 つの主要な要因に分けることができます。どちらの領域も、障害のモニタリング、複数の場所にデプロイする、自動フェイルオーバーなど、同じベストプラクティスの一部に依存しています。ただし、可用性は MES マイクロサービスのコンポーネントに重点を置いているが、ディザスタリカバリはマイクロサービス全体、または MES システム全体の個別のコピーに重点を置いています。

可用性

可用性は、次の式に示すように、マイクロサービスが使用できる時間の割合として定義されます。この割合は、1 か月、1 年、3 年後など、一定期間にわたって計算されます。

$$A = \frac{\textit{uptime}}{\textit{uptime} + \textit{downtime}}$$

この式では、製造と機器のメンテナンスに共通する 3 つのメトリクスを理解する必要があります。

- 平均障害間隔 (MTBF): マイクロサービスの通常のオペレーションの開始からその後の障害までの平均時間。
- 平均検出時間 (MTTD): 障害の発生から修復オペレーションの開始までの平均時間。
- 平均修復時間 (MTTR): サブシステムの障害によるマイクロサービスの利用不能と、その修復またはサービスへの復帰までの平均時間。MTTD は MTTR のサブセットです。

次の図は、これらの可用性メトリクスを示しています。



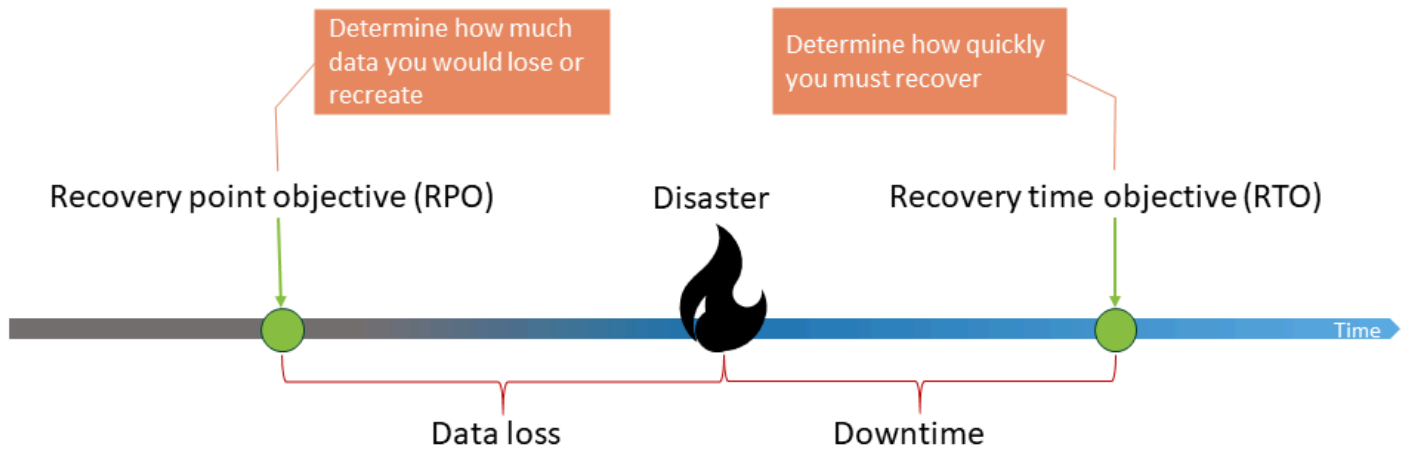
回復力があり可用性の高い MES は、MTTR と MTTD を削減し、MTBF を増やすことを目的としています。理想的な設計では障害は排除されますが、現実的ではありません。従来のモノリシック MES 障害は検出が困難で、修復に時間がかかりました。最新のクラウドネイティブな MES では、マルチ AZ 配置による迅速な検出、迅速な修復、ビジネス継続性を実現できます。関連 AWS サービスを備えた高可用性の最新のシステムのベストプラクティスについては、ホワイトペーパー「[可用性とそれを超越: での分散システムの耐障害性について AWS](#)」を参照してください。

ディザスタリカバリ

ディザスタリカバリとは、ハードウェアやソフトウェアの重大な障害など、テクノロジー関連の災害に備え、そこから復旧するプロセスを指します。マイクロサービスまたは MES が、デプロイされたプライマリロケーションでビジネス目標を達成できないようにするイベントは、災害と見なされます。ディザスタリカバリは可用性とは異なり、次の 2 つのメトリクスによって測定されます。

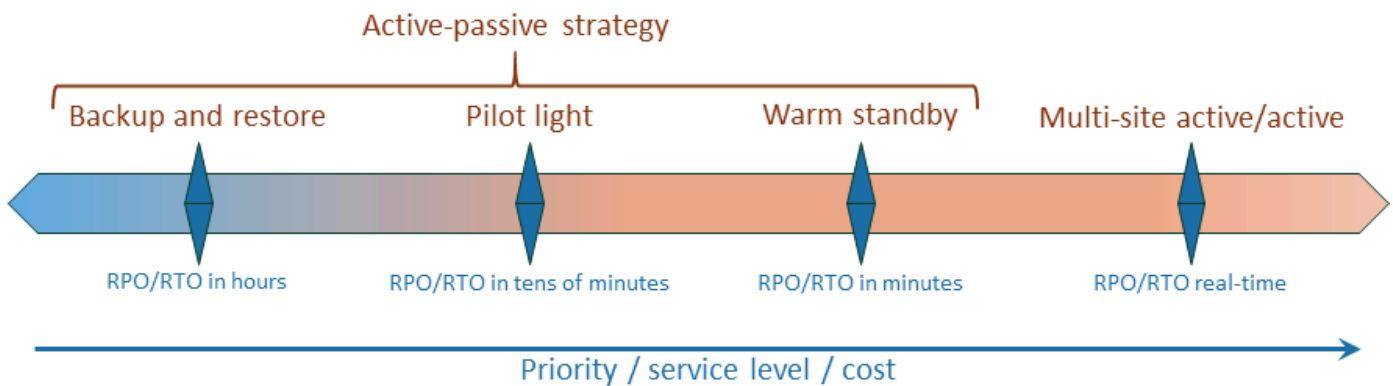
- 目標復旧時間 (RTO): マイクロサービスの中断とマイクロサービスの復元のための許容可能な遅延。RTO は、サービスが利用できないときに許容される時間枠と見なされるものを決定します。
- 目標復旧時点 (RPO): 最後のデータ復旧時点からの最大許容時間。RPO は、最後の復旧時点からマイクロサービスの中断までの間に許容されるデータ損失と見なされるものを決定します。

次の図は、これらのディザスタリカバリメトリクスを示しています。



次の図は、さまざまなディザスタリカバリ戦略を示しています。

Disaster recovery strategies



これらの戦略 AWS の実装に関する詳細なガイドは、Well-Architected フレームワークガイド「[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。

結論

マイクロサービスベースのアーキテクチャは、従来のモノリシック MES に伴う制限を克服するのに役立ちます。マイクロサービスベースのアプリケーションの構築には、アーキテクチャの複雑さや運用上のオーバーヘッドなどの課題があります。マイクロサービスベースの MES の可能性を最大限に引き出すには、以下の質問を検討することをお勧めします。

- 解決しようとしている現在のアーキテクチャにはどのような制限がありますか？
- ビジネスやアーキテクチャに関する意思決定を行うのに十分な専門知識はありますか？
- ガバナンス構造を持っているか、または構築する予定はありますか？
- テストとデプロイの自動化は実施されていますか？
- 変更管理とトレーニング計画はありますか？

AWS [モダナイゼーションアクセラレーション](#)、[評価](#)、[ワークショップ](#)、[ソリューションガイド](#)
[ス](#)、[イマージョンデー](#)などのリソースにより、メーカーはモダナイゼーションの取り組みから最大限のメリットを引き出すことができます。

リファレンス

AWS サービス

- [AWS Amplify](#) (フルスタックアプリケーション開発)
- [Amazon API Gateway](#) (API 管理)
- [AWS AppSync](#) (サーバーレス GraphQL APIs)
- [AWS CloudTrail](#) (API ログ)
- [Amazon CloudWatch](#) (APM ツール)
- [AWS Config](#) (マネージド設定サービス)
- [Amazon DynamoDB](#) (非リレーショナルデータベース)
- [Amazon EBS](#) (クラウドブロックストレージ)
- [Amazon EC2](#) (サイズ変更可能なコンピューティングウェブサービス)
- [Amazon EFS](#) (共有ファイルストレージ)
- [Amazon EventBridge](#) (イベントリスナー)
- [Amazon FSx](#) (マネージドファイルサーバー)
- [AWS IoT Core](#) (マネージド IoT クラウドプラットフォーム)
- [AWS IoT Greengrass](#) (オープンソースのエッジランタイムとクラウドサービス)
- [AWS IoT SiteWise](#) (IIoT データ収集、ストレージ、モニタリング)
- [AWS Lambda](#) (サーバーレス、イベント駆動型コンピューティング)
- [Amazon Managed Service for Prometheus](#) (マネージドコンテナモニタリング)
- [Amazon MQ](#) (メッセージブローカー)
- [Amazon RDS](#) (リレーショナルデータベース)
- [Amazon S3](#) (クラウドオブジェクトストレージ)
- [Amazon SageMaker AI](#) (ML モデリング)
- [Amazon SNS](#) (プッシュ通知)
- [Amazon SQS](#) (メッセージキューイング)
- [AWS Step Functions](#) (ワークフローオーケストレーション)

AWS サービスファミリー

- [での AI/ML AWS](#)
- [の分析サービス AWS](#)
- [のコンテナ AWS](#)
- [のデータベース AWS](#)
- [でのエッジサービス AWS](#)
- [でのフロントエンドウェブとモバイル AWS](#)
- [での IoT サービス AWS](#)
- [でのサーバーレス AWS](#)

その他の AWS リソース

- [AWS 評価ツール](#)
- [AWS IoT コンピテンシーパートナー](#)
- [AWS Migration Acceleration プログラム](#)
- [AWS ソリューションライブラリ](#)
- [AWS ソリューションに焦点を当てた没入日数](#)
- [AWS Well-Architected フレームワーク](#)
- [AWS ワークショップ](#)
- [AWS クラウドコンピューティングの概念ハブ](#)
- 出版物 :
 - [可用性以降: での分散システムの耐障害性について AWS](#) (AWS ホワイトペーパー)
 - [でのワークロードのディザスタリカバリ AWS: クラウドでの復旧](#) (AWS ホワイトペーパー)
 - [産業用データファブリック](#) (AWS パートナーソリューションとガイド) (AWS ホワイトペーパー)
 - [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](#) (AWS 規範ガイド)
 - [Amazon EKS でのロードバランシング](#) (Amazon EKS ドキュメント)
 - [AWS Outposts を使用した での AWS Lambda 関数の実行 AWS IoT Greengrass](#) (AWS ブログ記事)

作成者と寄稿者

以下の人がこのガイド AWS を作成し、寄稿しました。

作成者 :

- Ravi Soni、プリンシパル産業用製造ソリューションスペシャリスト
- スティーブ・ブラックウェル、製造担当ワールドワイドテクニカルリーダー
- Nishant Saini、プリンシパルパートナーソリューションアーキテクト
- Pratik Yeole、ソリューションアーキテクト

寄稿者 :

- Darpan Parikh、Composable App Solutions 責任者
- Jan Metzner、プリンシパル産業用製造ソリューションスペシャリスト
- Bhavisha Dawada、シニアソリューションアーキテクト

ドキュメント履歴

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

変更	説明	日付
更新	「データと分析」セクションの アーキテクチャ図と説明 を更新しました。	2024 年 4 月 2 日
初版発行	—	2024 年 2 月 23 日

AWS 規範ガイドの用語集

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

数字

7 Rs

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

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

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

A

A2A (Agent-to-Agent)

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

ABAC

「[属性ベースのアクセス制御](#)」をご覧ください。

抽象化されたサービス

「[マネージドユーザー](#)」をご覧ください。

ACID

「[原子性、一貫性、分離性、耐久性 \(ACID\)](#)」をご覧ください。

アクティブ/アクティブ移行

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

アクティブ/パッシブ移行

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

[エージェント]

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

エージェントオペレーション

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

集計関数

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

AI

「[人工知能](#)」をご覧ください。

AIOps

「[AI オペレーション](#)」をご覧ください。

匿名化

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

アンチパターン

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

アプリケーション制御

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

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

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリ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。

アベイラビリティゾーン (AZ)

他のアベイラビリティゾーンの障害から 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ゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

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

AWS ワークロード認定フレームワーク (AWS WQF)

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

B

不正なボット

個人や組織に混乱や損害を与えることを目的とした[ボット](#)。

BCP

「[ビジネス継続性計画 \(BCP\)](#)」をご覧ください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの一つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

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

ブルー/グリーンデプロイ

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

ボット

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

ボットネット

[マルウェア](#)に感染しており、ボットハーダーまたはボットオペレーターと呼ばれる単一の当事者によって制御されている[ボット](#)のネットワーク。ボットネットは、ボットとその影響力を拡大する仕組みとして、非常によく知られています。

ブランチ

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

ブレイクグラスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが AWS アカウント 通常アクセス許可を持たないにすばやくアクセスできるようになります。詳細については、AWS Well-Architected ガイダンスの「[ブレイ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н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н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нフィールド戦略と[グリーンフィールド](#)戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

「[AWS クラウド導入フレームワーク](#)」を参照してください。

カナリアデプロイ

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

CCoE

「[Cloud Center of Excellence](#)」を参照してください。

CDC

「[変更データキャプチャ](#)」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

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

CI/CD

「[継続的インテグレーションと継続的デリバリー](#)」を参照してください。

分類

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

シチズンデベロッパー

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

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が、AWS クラウドへの移行時に通常実行する 4 つの段階。

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

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

CMDB

「[構成管理データベース \(CMDB\)](#)」を参照してください。

コードリポジトリ

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

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

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

設定ドリフト

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

構成管理データベース (CMDB)

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

コンフォーマンスパック

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

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

継続的インテグレーションと継続的デリバリー (CI/CD)

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

CV

「[コンピュータビジョン](#)」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

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

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、[「でのデータ境界の構築 AWS」](#)を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

[「データベース定義言語」](#)を参照してください。

ディープアンサンブル

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

深層学習

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

多層防御

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

委任管理者

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

トラブルシューティング

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

開発環境

「[環境](#)」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

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

ディザスタ

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

ディザスタリカバリ (DR)

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

DML

「[データベース操作言語](#)」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計: ソフトウェアの中心における複雑さへの取り組み) で紹介されています (ポストン: Addison-Wesley Professional, 2003)。strangler fig パターンでドメイン駆動型設計を使用す

る方法の詳細については、「[コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET \(ASMX\) ウェブサービスを段階的にモダナイズ](#)」を参照してください。

DR

「[ディザスタリカバリ](#)」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、AWS CloudFormation を使用して[システムリソースのドリフトを検出](#)したり、を使用して AWS Control Tower、ガバナンス要件への準拠に影響する[ランディングゾーンの変更を検出](#)したりできます。

DVSM

「[開発バリューストリームマッピング](#)」を参照してください。

E

EDA

「[探索的データ分析](#)」を参照してください。

EDI

「[電子データ交換](#)」を参照してください。

エッジコンピューティング

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

電子データ交換 (EDI)

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

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

「[サービスエンドポイント](#)」を参照してください。

エンドポイントサービス

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

エンタープライズリソースプランニング (ERP)

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

エンベロープ暗号化

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

環境

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

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

- 上位環境 — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

エピック

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

ERP

「[エンタープライズリソース計画](#)」を参照してください。

探索的データ分析 (EDA)

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

F

ファクトテーブル

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

フェイルファスト

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

障害分離境界

では AWS クラウド、アベイラビリティーゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界で、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。詳細については、「[AWS 障害分離境界](#)」を参照してください。

機能ブランチ

「[ブランチ](#)」を参照してください。

特徴量

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

特徴量重要度

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

機能変換

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

数ショットプロンプト

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

FGAC

「[きめ細かなアクセス制御](#)」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

FM

「[基盤モデル](#)」を参照してください。

基盤モデル (FM)

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

FM ゲートウェイ

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

G

生成 AI

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

ジオブロッキング

「[地理的制限](#)」を参照してください。

地理的制限 (ジオブロッキング)

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

Gitflow ワークフロー

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

ゴールデンイメージ

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

グリーンフィールド戦略

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

ガードレール

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

ガードレール (AI)

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

H

HA

「[高可用性](#)」を参照してください。

異種混在データベースの移行

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

高可用性 (HA)

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

ヒストリアンのモダナイゼーション

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

ホールドアウトデータ

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

ヒューman-in-the-loop (HitL)

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

同種データベースの移行

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

ホットデータ

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

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性が高いため、通常の DevOps のリリースワークフローからは外れた形で実施されます。

ハイパーケア期間

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

|

laC

「[Infrastructure as Code](#)」を参照してください。

|

ID ベースのポリシー

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

アイドル状態のアプリケーション

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

IIoT

「[インダストリアル IIoT](#)」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (受信) VPC

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

増分移行

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

インダストリー 4.0

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

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

インダストリアル IoT (IIoT)

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

インスペクション VPC

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

IoT

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

解釈可能性

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

IoT

「[IoT](#)」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「[IT 情報ライブラリ](#)」を参照してください。

ITSM

「[IT サービス管理](#)」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

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

大規模言語モデル (LLM)

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

大規模な移行

300 台以上のサーバの移行。

LBAC

「[ラベルベースアクセス制御](#)」を参照してください。

最小特権

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

リフトアンドシフト

「[7 Rs](#)」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

LLM

「[大規模言語モデル](#)」を参照してください。

下位環境

「[環境](#)」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「[ブランチ](#)」を参照してください。

マルウェア

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

マネージドサービス

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

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

製造実行システム (MES)

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

MAP

「[Migration Acceleration Program](#)」を参照してください。

MCP

「[モデルコンテキストプロトコル](#)」を参照してください。

モデルコンテキストプロトコル (MCP)

[エージェント](#)と[ツール](#)間の通信のためのステートレスプロトコル。

MCP サーバー

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

メカニズム

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

メンバーアカウント

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

MES

「[製造実行システム](#)」を参照してください。

Message Queuing Telemetry Transport (MQTT)

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

マイクロサービス

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

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

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

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

大規模な移行

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

移行ファクトリー

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

移行メタデータ

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

移行パターン

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

Migration Portfolio Assessment (MPA)

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

移行準備状況評価 (MRA)

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

移行戦略

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

ML

「[機械学習](#)」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

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

モノリシックアプリケーション (モノリス)

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

MPA

「[Migration Portfolio Assessment](#)」を参照してください。

MQTT

「[Message Queuing Telemetry Transport](#)」を参照してください。

多クラス分類

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

ミュータブルなインフラストラクチャ

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

O

OAC

「[オリジンアクセス制御](#)」を参照してください。

OAI

「[オリジンアクセスアイデンティティ](#)」を参照してください。

OCM

「[組織変更管理](#)」を参照してください。

オフライン移行

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

OI

「[オペレーション統合](#)」を参照してください。

Ola

「[オペレーショナルレベルアグリーメント](#)」を参照してください。

オンライン移行

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

OPC-UA

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

Open Process Communications - Unified Architecture (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

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

運用テクノロジー (OT)

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

オペレーション統合 (OI)

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

組織の証跡

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

組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の導入を加速し、移行問題に対処し、文化や組織の変化を推進することで、組織が新しいシステムと戦略の準備と移行するのを支援します。AWS 移行戦略では、クラウド導入プロジェクトに必要な変化のスピードにより、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、[OCM ガイド](#)を参照してください。

オリジンアクセス制御 (OAC)

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

オリジンアクセスアイデンティティ (OAI)

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

ORR

「[運用準備状況レビュー](#)」を参照してください。

OT

「[運用テクノロジー](#)」を参照してください。

アウトバウンド (送信) VPC

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

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

「[個人を特定できる情報](#)」を参照してください。

プレイブック

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

PLC

「[プログラマブルロジックコントローラー](#)」を参照してください。

PLM

「[製品ライフサイクル管理](#)」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

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

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

アクションを実行し AWS、リソースにアクセスできるのエンティティ。このエンティティは通常、IAM AWS アカウントロール、またはユーザーのルートユーザーです。詳細については、IAM ドキュメントの「[ロールに関する用語と概念](#)」にあるプリンシパルを参照してください。

プライバシーバイデザイン

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

プライベートホストゾーン

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

プロアクティブコントロール

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

製品ライフサイクル管理 (PLM)

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

本番環境

「[環境](#)」を参照してください。

プログラマブルロジックコントローラー (PLC)

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

プロンプトチェイニング

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

仮名化

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

発行/サブスクライブ (pub/sub)

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

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

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

RAG

「[検索拡張生成](#)」を参照してください。

ランサムウェア

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

RASCI マトリックス

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

RCAC

「[行と列のアクセス制御](#)」を参照してください。

リードレプリカ

読み取り専用で使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースへの負荷を軽減できます。

リアーキテクト

「[7 Rs](#)」を参照してください。

目標復旧時点 (RPO)

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

目標復旧時間 (RTO)

サービスが中断から復旧までの最大許容遅延時間。

リファクタリング

「[7 Rs](#)」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョンは、耐障害性、安定性、耐障害性を提供するために、他のとは独立しています。詳細については、「[アカウントが使用できる AWS リージョンを指定する](#)」を参照してください。

リグレッション

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

リホスト

「[7 Rs](#)」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「[7 Rs](#)」を参照してください。

リプラットフォーム

「[7 Rs](#)」を参照してください。

再購入

「[7 Rs](#)」を参照してください。

回復性

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

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンスコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、「AWSでのセキュリティコントロールの実装」の「[レスポンスコントロール](#)」を参照してください。

保持

「[7 Rs](#)」を参照してください。

廃止

「[7 Rs](#)」を参照してください。

検索拡張生成 (RAG)

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

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「[目標復旧時点](#)」を参照してください。

RTO

「[目標復旧時間](#)」を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーテッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、にログイン AWS マネジメントコンソールしたり AWS、API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「[SAML 2.0 ベースのフェデレーションについて](#)」を参照してください。

SCADA

「[監視制御とデータ取得](#)」を参照してください。

SCP

「[サービスコントロールポリシー](#)」を参照してください。

シークレット

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

セキュリティバイデザイン

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

セキュリティコントロール

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

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

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

サーバー側の暗号化

送信先で、それ AWS のサービスを受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

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

サービスレベルアグリーメント (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

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

責任共有モデル

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

シャドウ AI

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

SIEM

「[Security Information and Event Management システム](#)」を参照してください。

単一障害点 (SPOF)

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

SLA

「[サービスレベルアグリーメント](#)」を参照してください。

SLI

「[サービスレベルインジケータ](#)」を参照してください。

SLO

「[サービスレベルの目標](#)」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お

お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、「[AWS クラウドでのアプリケーションをモダナイズするための段階的アプローチ](#)」を参照してください。

SPOF

「[単一障害点](#)」を参照してください。

スタースキーマ

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

strangler fig パターン

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

サブネット

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

監視制御とデータ取得 (SCADA)

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

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

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

システムプロンプト

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

T

タグ

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

ターゲット変数

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

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要がある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

テスト環境

「[環境](#)」を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパターンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

tool

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

トランジットゲートウェイ

VPC とオンプレミスネットワークを相互接続するために使用できる、ネットワークの中継ハブ。詳細については、AWS Transit Gateway ドキュメントの「[トランジットゲートウェイとは](#)」を参照してください。

トランクベースのワークフロー

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

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定したサービスにアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要とときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、ドキュメントの「[を他の AWS のサービス AWS Organizations で使用する AWS Organizations](#)」を参照してください。

チューニング

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

ツーピザチーム

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

U

不確実性

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

未分化なタスク

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

上位環境

「[環境](#)」を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

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

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

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

ワークロード

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

ワークストリーム

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

WORM

「[Write-Once-Read-Many](#)」を参照してください。

WQF

「[AWS ワークロード資格フレームワーク](#)」を参照してください

Write-Once-Read-Many (WORM)

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

Z

ゼロデイ 익스プロイト

[ゼロデイ脆弱性](#)を悪用した攻撃 (一般的にマルウェアによる)。

ゼロデイ脆弱性

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

ゼロショットプロンプト

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

ゾンビアプリケーション

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

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