で製造実行システム (MES) をモダナイズする AWS クラウド

AWS 規範ガイダンス

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

AWS 規範ガイダンス: 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
..... xliii

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、このガイダンスでは、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、貴重なインサイトを引き出し、パターンを推測し、イベントを予測し、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。

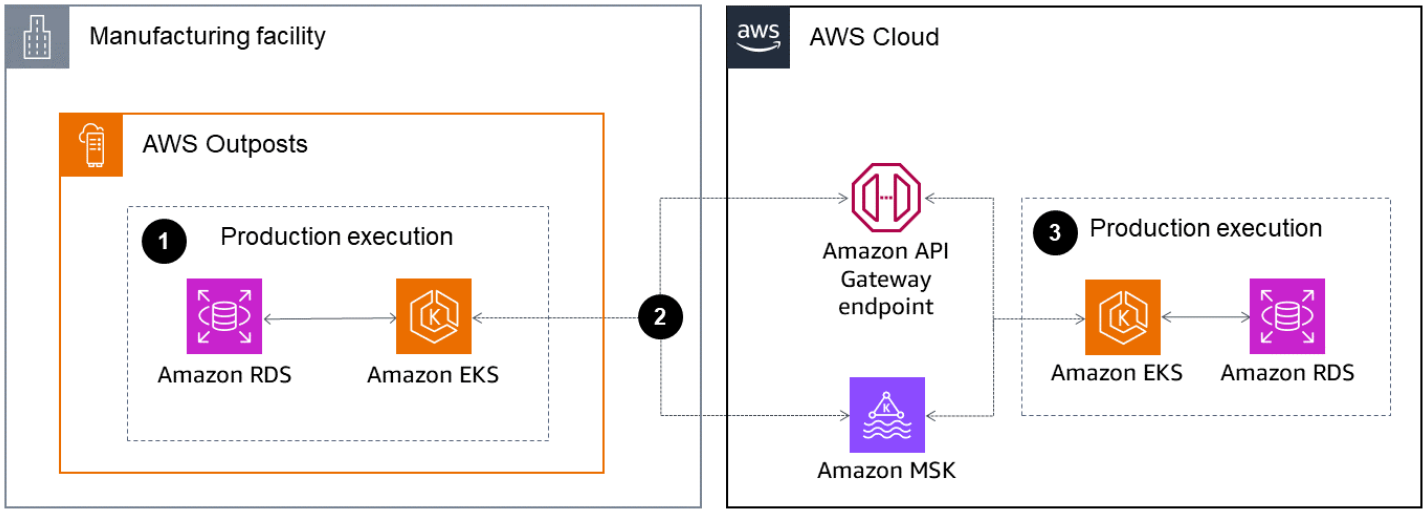
産業用エッジコンピューティング

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商品の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て、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

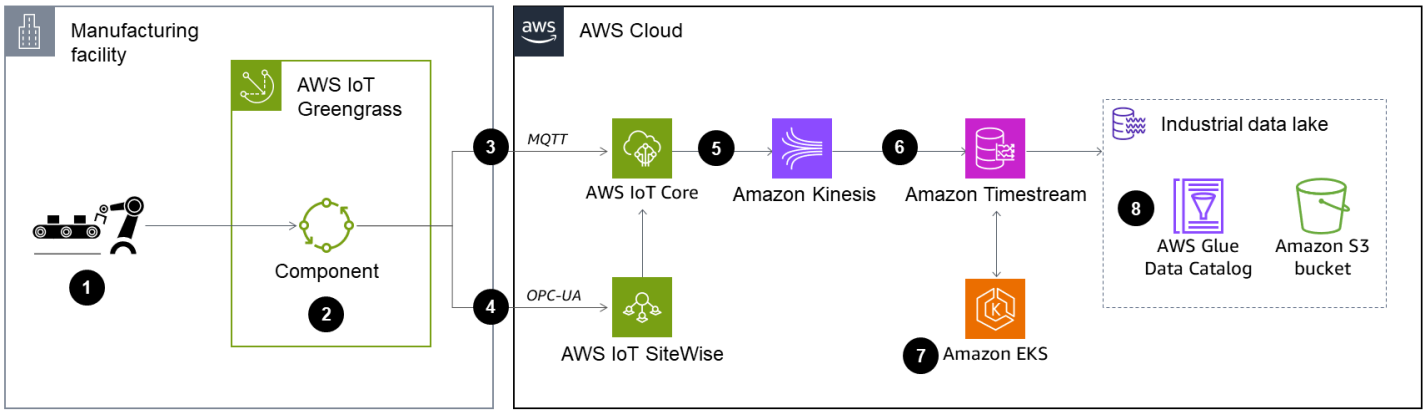
したり、時間を節約したり、手動入力によるエラーの可能性を排除したりすることもできます。例えば、手動データ入力なしで、テストマシンから結果を収集し、製品品質を判断し、不適合レコードや二次検査ワークフローを自動的に作成できます。時間の経過とともに、クラウドネイティブ IoT サービスは、欠陥の特定のパターンと根本原因を見つけるのに役立ち、製造プロセスを変更することで欠陥の発生を防ぐことができます。

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、お客様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かつ安全にサポートでき、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。

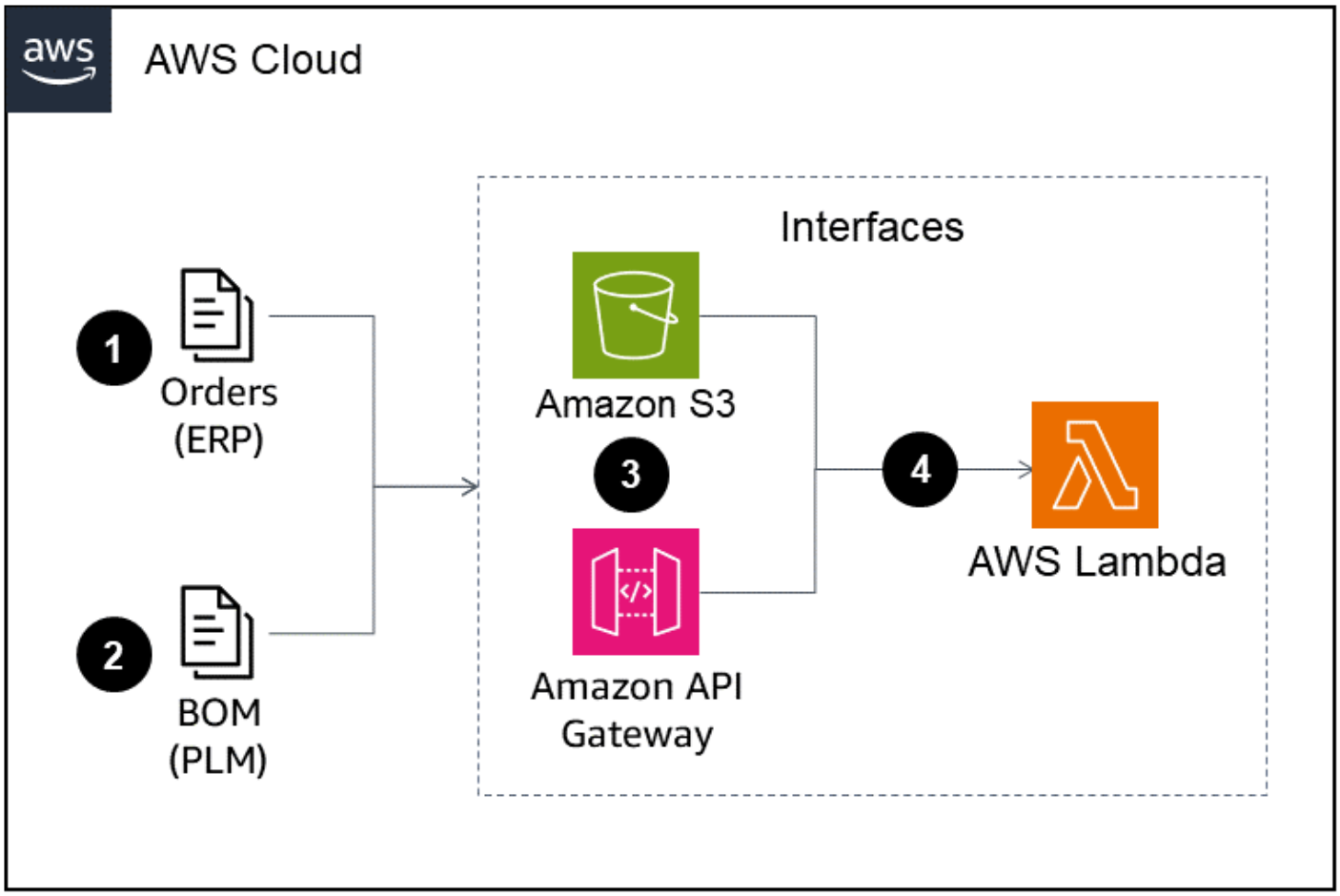
他のエンタープライズアプリケーションとのインターフェイス

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 でホストされている場合 AWS クラウド、ファイルボールトは別の S3 バケットである可能性があり、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。

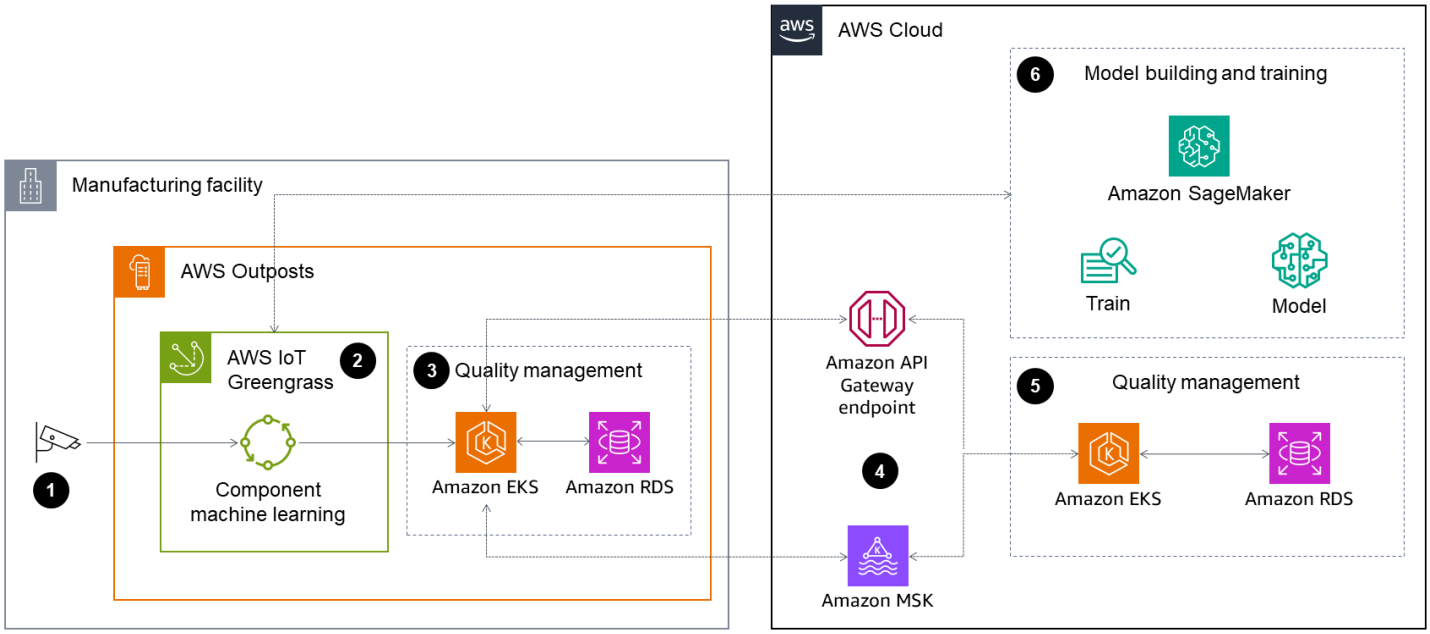
以下に示しているのは、産業用の目立つ AWS ML サービスの一部です。

- 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、高精度かつ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的検査タスク 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。

> **サポート終了通知**
>
> 2026 年 5 月 31 日、 AWS は 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。

4. 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。
5. 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。
6. 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

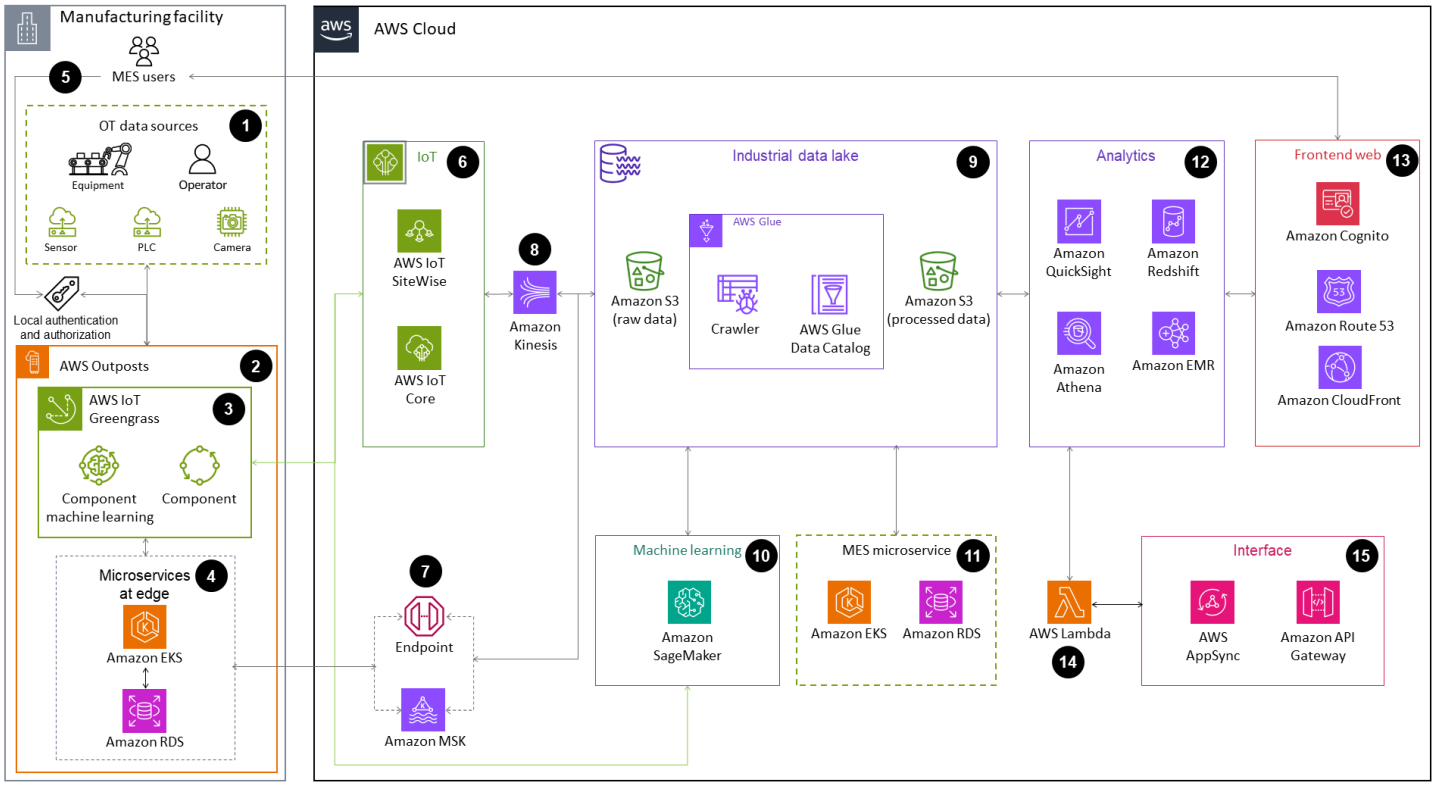
従来のモノリシック MES システムでは、分析機能が限られていたか、まったくありませんでした。メーカーは、毎日の生産、在庫レベル、品質結果などの基本的なレポートのために、高価なサードパーティー製ツールや、スプレッドシートへのバックエンドデータ抽出の複雑な方法に依存する必要がありました。分析のために MES データを他のアプリケーションやシステムデータと組み合わせる可能性はほとんどありませんでした。の AWS マイクロサービスベースの MES は、MES の一般的な分析課題を解決し、メーカーに競争上の優位性を与えるための追加の分析機能を提供します。 AWS クラウド は、メーカーに一連の専用分析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

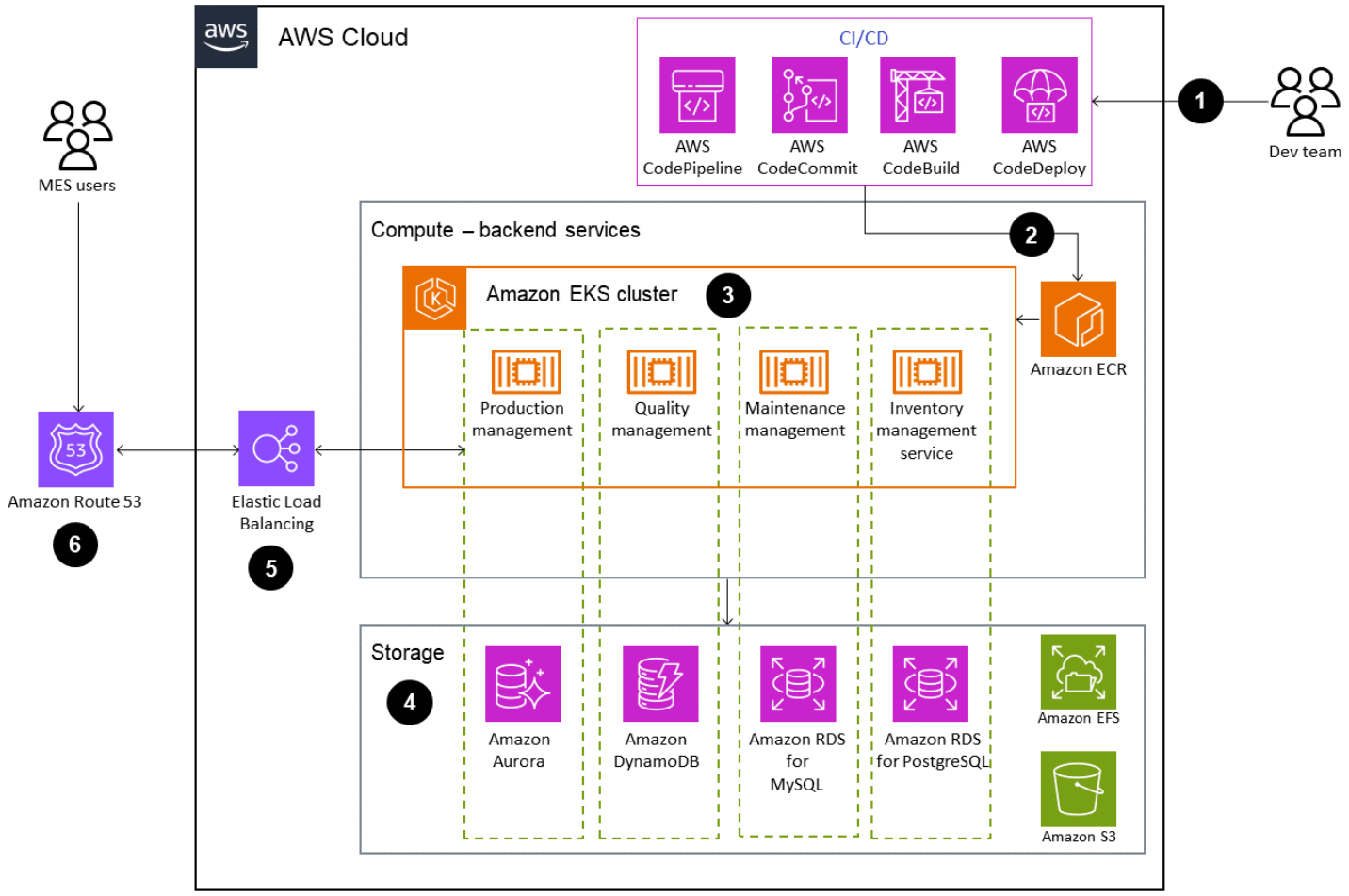
4. MES 用マイクロサービスのローカルインスタンスは、任意のマイクロサービスである可能性があり、要件に応じて、エッジに複数のマイクロサービスが存在する可能性があります。
5. ローカル認証と認可により、MES ユーザーは、リアルタイムの本稼働レポートなどのレイテンシーの影響を受けやすいユースケースや、接続が中断された場合に、ローカルマイクロサービスに安全にアクセスできます。
6. などの IoT サービスは、クラウドでデータ AWS IoT Core を受信し AWS IoT SiteWise 、データを保存して処理します。
7. Amazon API Gateway エンドポイントと Amazon 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導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す。
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で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

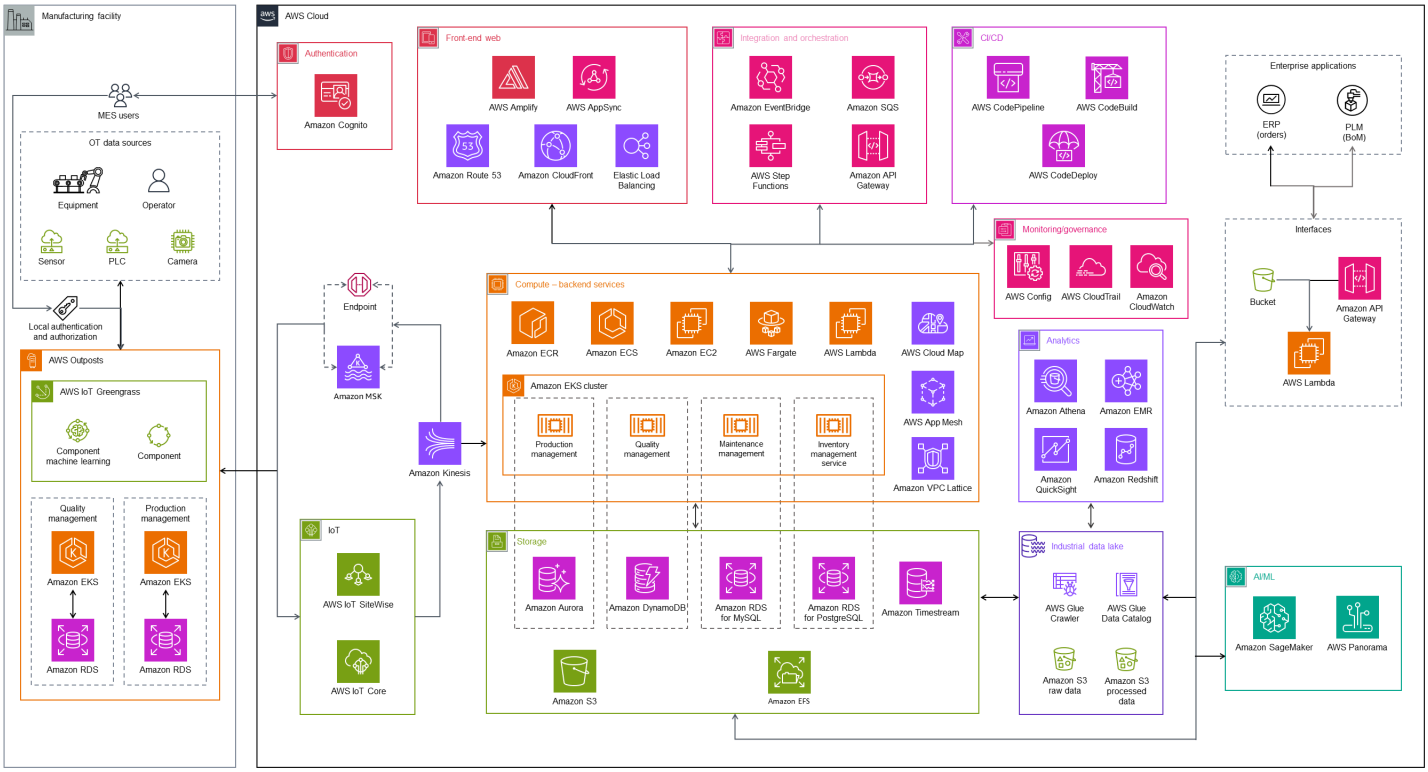
次の図のアーキテクチャは、DNS とロードバランシングを組み合わせて、バックエンドのコンテナ化されたコンピューティングで一貫したユーザーエクスペリエンスを実現します。また、継続的な更新のための継続的インテグレーションと継続的デプロイ (CI/CD) パイプラインも含まれています。



1. MES 開発チームは を使用してコード AWS CodePipeline 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。

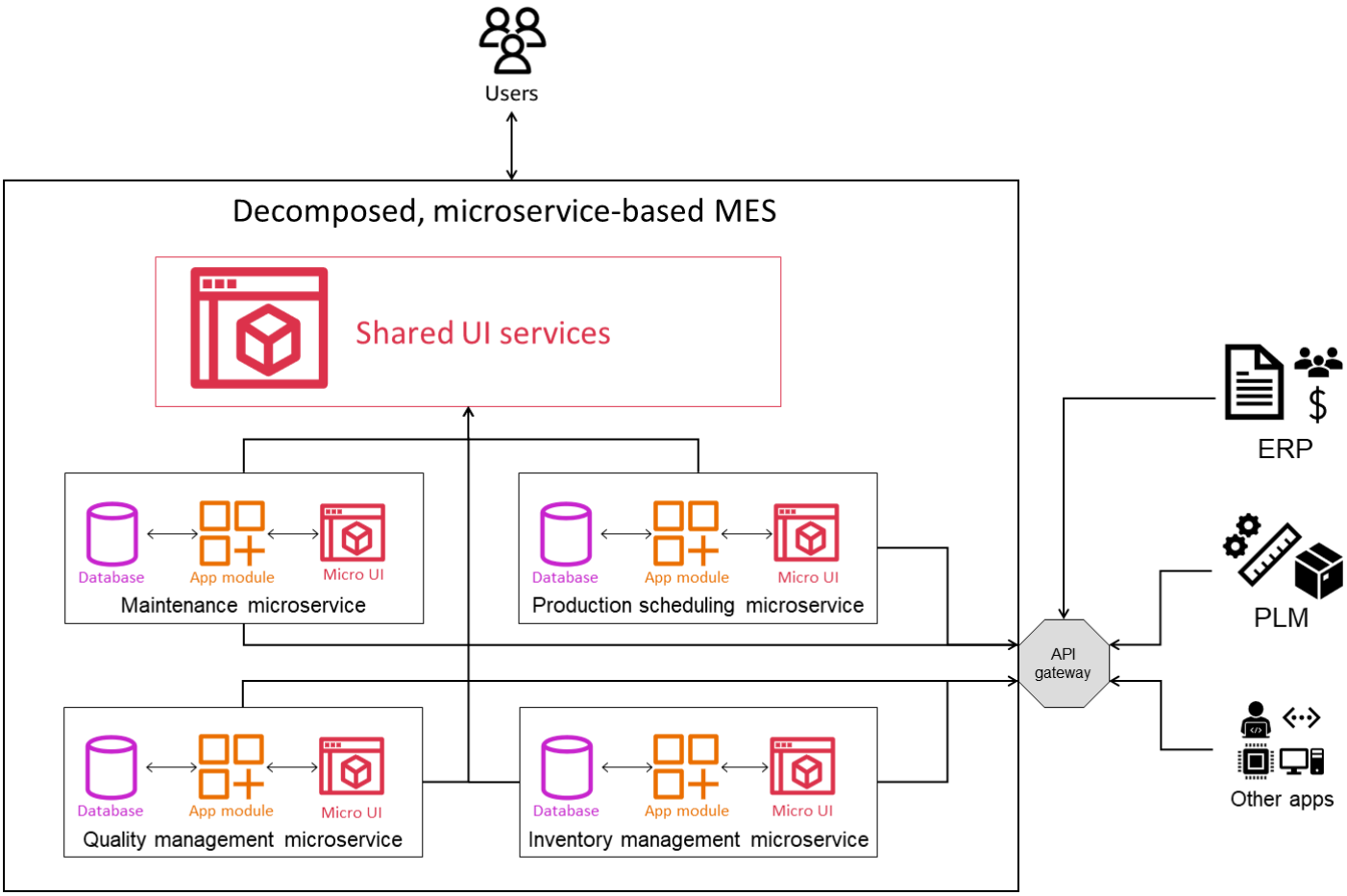

Manufacturing facility
MES users
OT data sources
Equipment
Operator
Sensor
PLC
Camera
Local authentication and authorization
AWS Outposts
AWS IoT Greengrass
Component machine learning
Component
Quality management
Production management
Amazon EKS
Amazon EKS
Amazon RDS
Amazon RDS
AWS Cloud
Authentication
Amazon Cognito
Front-end web
AWS Amplify
AWS AppSync
Amazon Route 53
Amazon CloudFront
Elastic Load Balancing
Integration and orchestration
Amazon EventBridge
Amazon SQS
AWS Step Functions
Amazon API Gateway
CI/CD
AWS CodePipeline
AWS CodeBuild
AWS CodeDeploy
Enterprise applications
ERP (orders)
PLM (BoM)
Interfaces
Bucket
Amazon API Gateway
AWS Lambda
Monitoring/governance
AWS Config
AWS CloudTrail
Amazon CloudWatch
Endpoint
Amazon MSK
Amazon Kinesis
Compute – backend services
Amazon ECR
Amazon ECS
Amazon EC2
AWS Fargate
AWS Lambda
AWS Cloud Map
Amazon EKS cluster
Production management
Quality management
Maintenance management
Inventory management service
AWS App Mesh
Amazon VPC Lattice
Analytics
Amazon Athena
Amazon EMR
Amazon QuickSight
Amazon Redshift
IoT
AWS IoT SiteWise
AWS IoT Core
Storage
Amazon Aurora
Amazon DynamoDB
Amazon RDS for MySQL
Amazon RDS for PostgreSQL
Amazon Timestream
Amazon S3
Amazon EFS
Industrial data lake
AWS Glue Crawler
AWS Glue Data Catalog
Amazon S3 raw data
Amazon S3 processed data
AI/ML
Amazon SageMaker
AWS Panorama

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、製造サイトへの MES のデプロイには数か月から数年かかることがあります。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、大規模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 の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を優先します。MES を、優先度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。次に、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。他のシステムまたはマイクロサービスに接続して、本番稼働ステータスを追跡し、適切なリソース割り当てを確保する場合があります。
- 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ル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、機器の状態のモニタリング、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、製品やマテリアルの検査、品質保証などの品質管理アクティビティを処理します。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、本番稼働スケジュールサービスに接続して検査タスクをスケジュールし、インベントリ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。

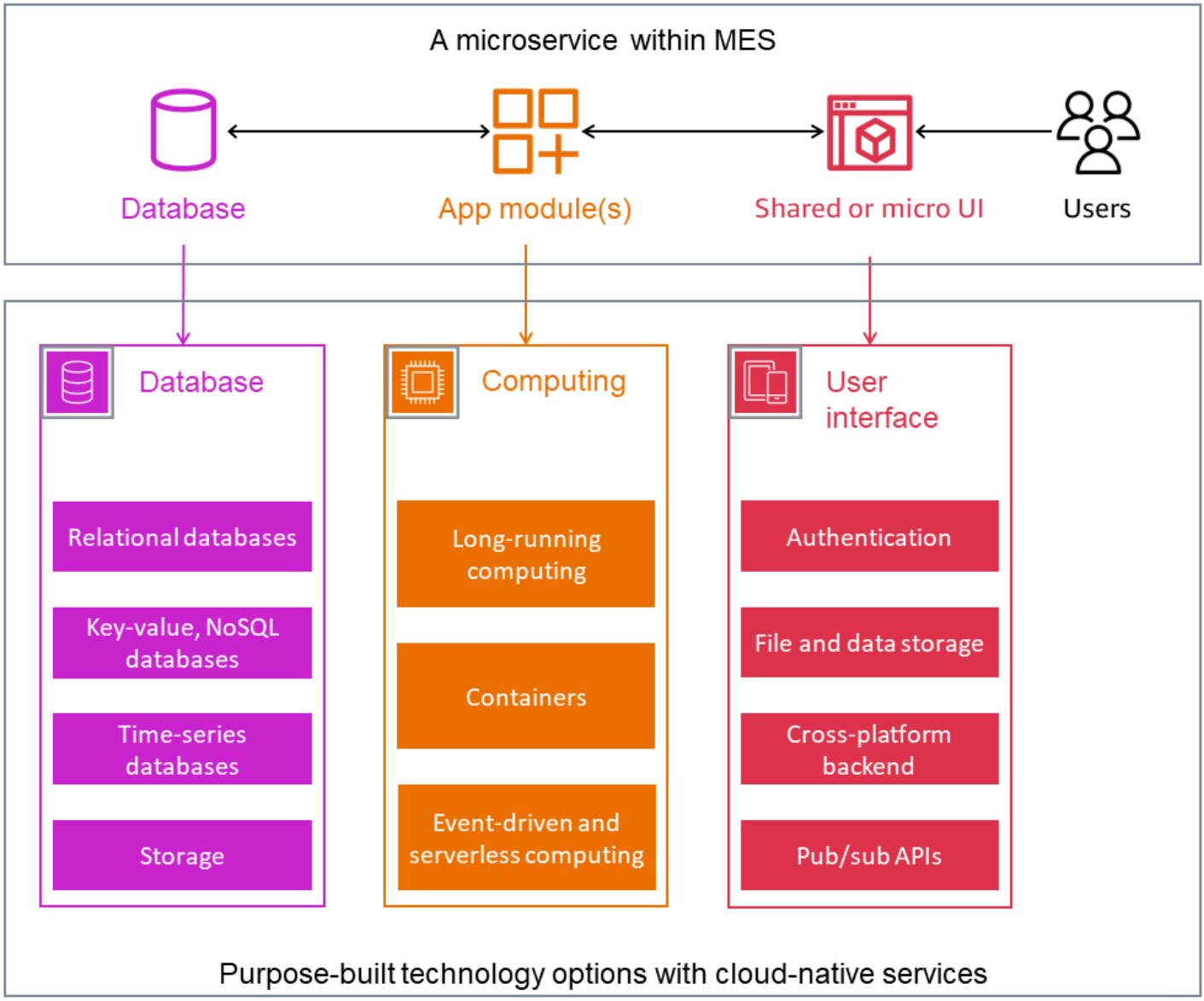
- 本番稼働サービスは、本番稼働用注文の実行を管理し、本番稼働用アクティビティを追跡します。マシンの状態、オペレータのアクション、マテリアルの消費など、本番稼働に関連するすべてのデータをキャプチャします。また、本番稼働用スケジューリングサービスに接続して、本番稼働用注文に関する情報、マテリアルの可用性と消費を追跡するインベントリ管理サービス、品質固有のワークフロー用の品質管理サービスに関する情報を取得することもできます。

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の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、製造元はデータ駆動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

て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 1 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を 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。 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を含む 8 つのデータベースファミリー AWS を提供します。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、整合性、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可能性があります。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型 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、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 日あたり数兆件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 ですAWS IoT SiteWise。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。例えば、オフィスのデスクから作業する店員の UI は、現場のハンドヘルド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の両方の UI 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 を作成し、パブリッシュ/サブスクライブ (pub/sub) APIs を作成します。これにより、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、データのクエリ、更新、公開を安全に行うことができます。

MES でのマイクロサービスの統合アプローチの決定

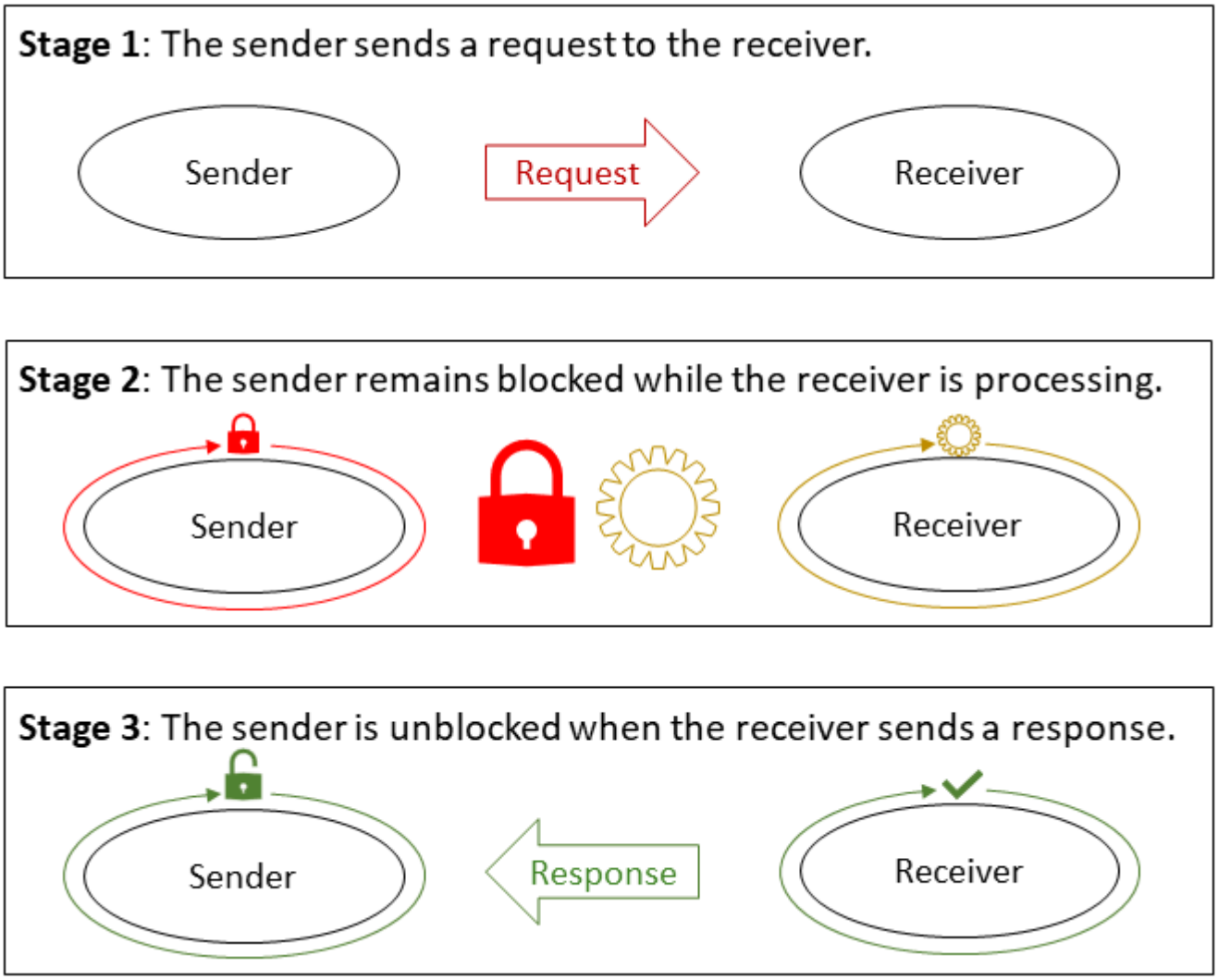
マイクロサービスベースの MES では、データ交換、情報共有、シームレスなオペレーションを確保するために、service-to-service通信が不可欠です。MES マイクロサービスは、特定のイベントまたは一定の間隔でデータを交換できます。たとえば、ユーザーは、本番稼働確認トランザクション中に本番稼働数量を提供する場合があります。このようなトランザクションは、ERP への情報の送信、マシンの実行時間のキャプチャ、製品に関する品質情報のキャプチャ、労働時間の報告など、バックグラウンドで複数のトランザクションを開始できます。さまざまなマイクロサービスがこれらのタスクを担当する可能性がありますが、1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

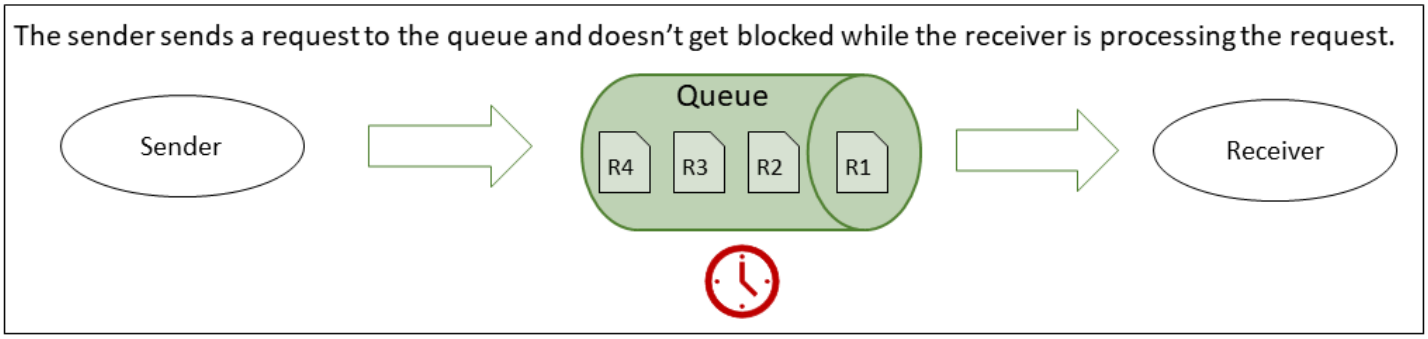
同期通信

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。たとえば、1 人のユーザーが注文に対するオペレーションを完了する連続した本番稼働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。



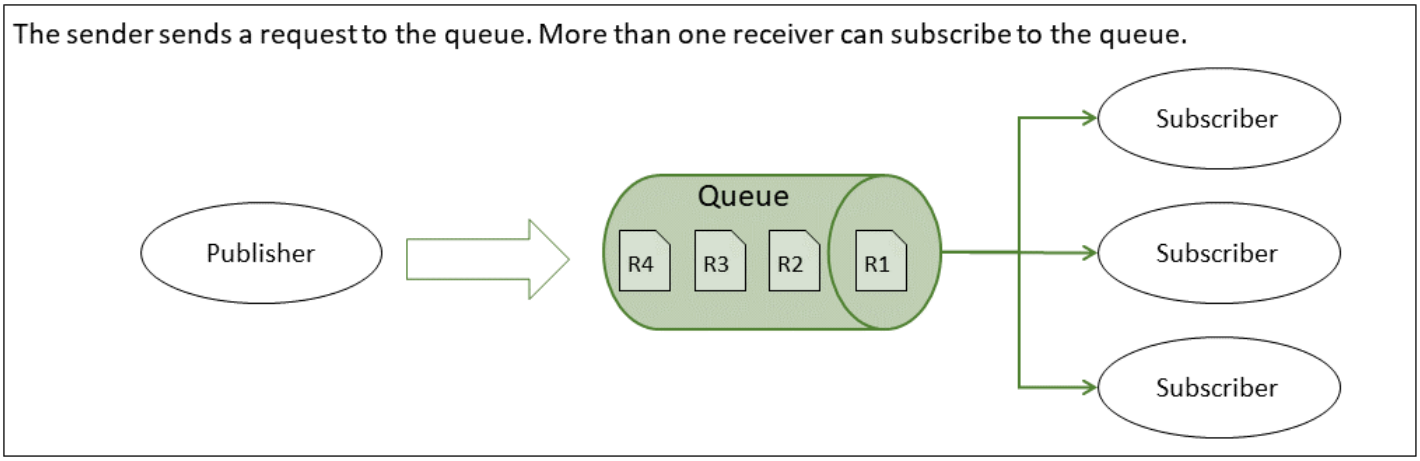
非同期通信

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、ビジネスト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。



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。リッ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ーマイクロサービスがリッスンできるチャ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。



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、「 規範ガイダンス」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

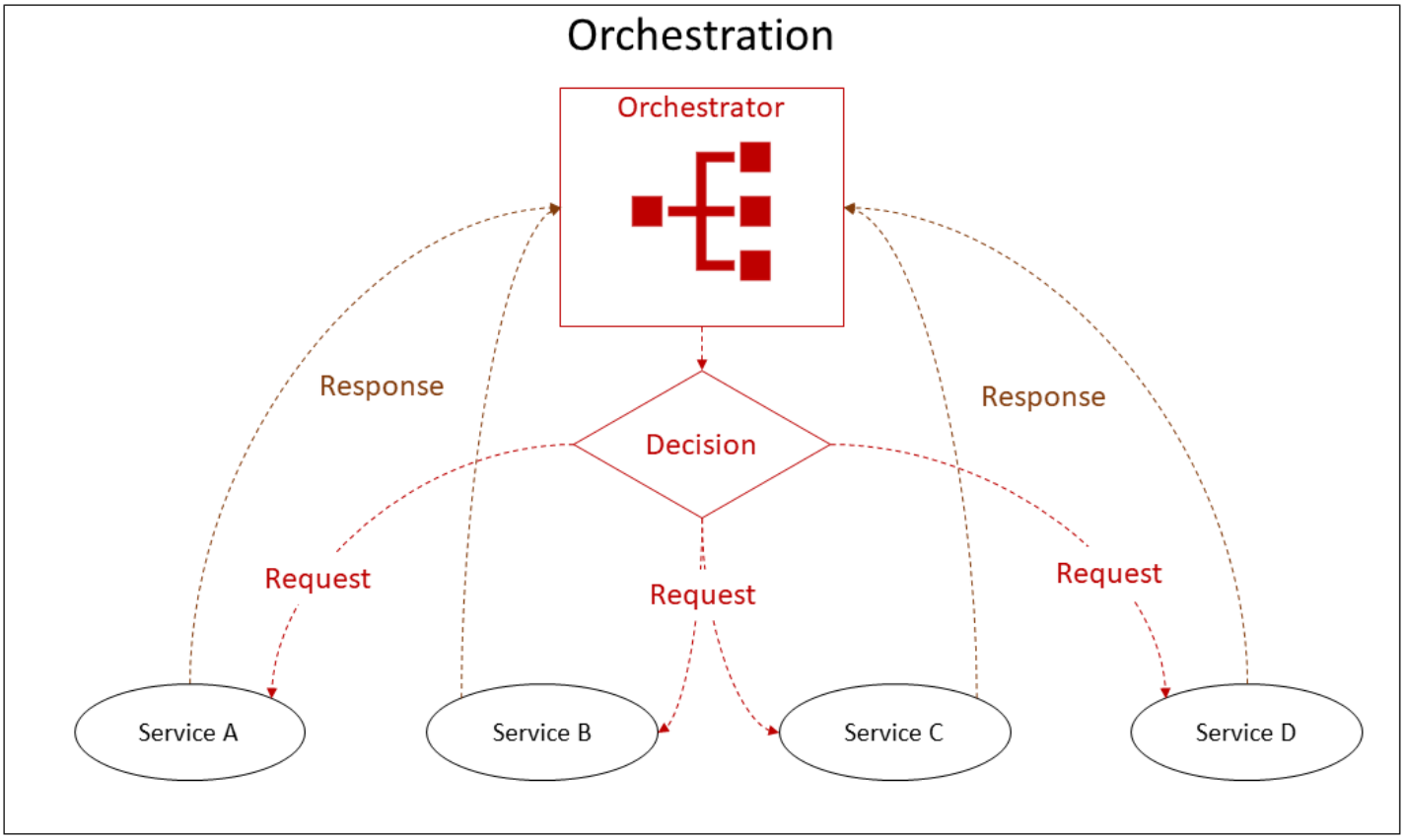
クラウドネイティブテクノロジーを使用した MES のマイクロサービスの管理、オーケストレーション、モニタリング

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 を提供します[AWS Step Functions](#)。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、サードパーティー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

回復力とは、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する機能です。WellAWS -Architected フレームワークの信頼性の柱が依存する主な要因は、回復力です。

回復力は、可用性とディザスタリカバリの 2 つの主な要因に分けることができます。どちらの分野も、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ますが、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

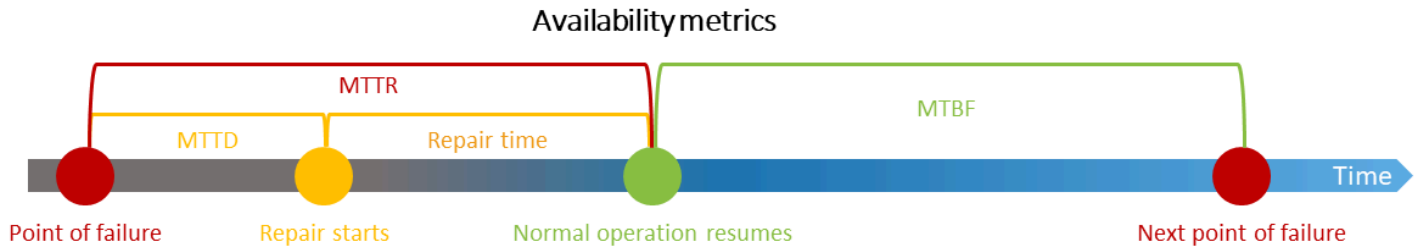
可用性は、次の式に示すように、マイクロサービスが使用できる時間の割合として定義されます。この割合は、1 か月、1 年、3 年後など、一定期間にわたって計算されます。

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

この式では、製造と機器のメンテナンスで一般的な 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から、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。



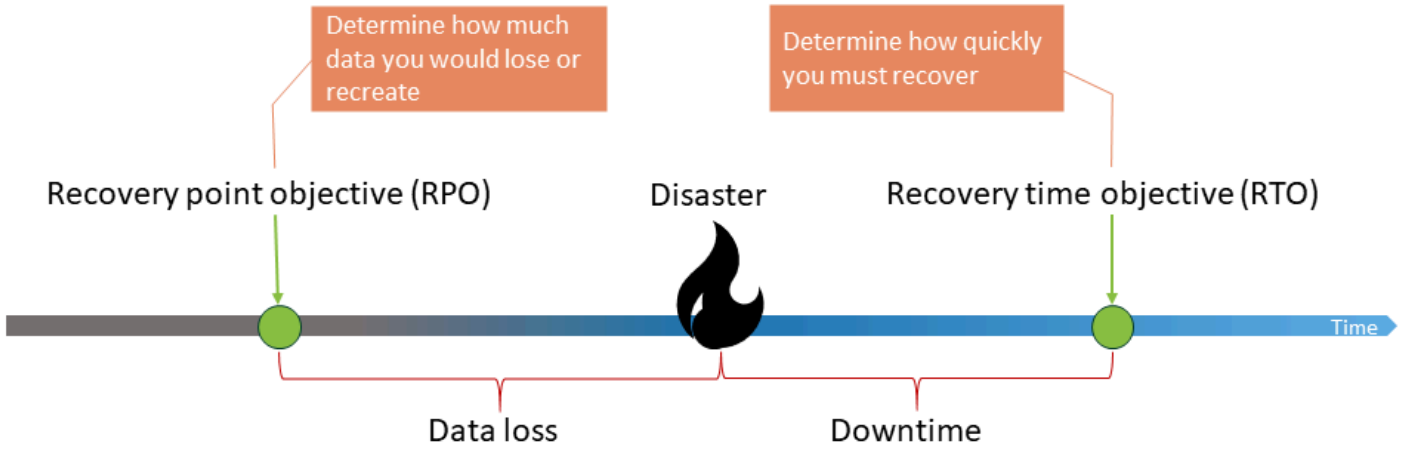
回復力があり可用性の高い MES は、MTTR と MTTD を減らし、MTBF を増やすことを目的としています。理想的な設計では障害を排除できますが、現実的ではありません。従来のモノリシック MES 障害は検出が難しく、修復に時間がかかりました。最新のクラウドネイティブ MES により、マルチ AZ 配置による迅速な検出、迅速な修復、ビジネス継続性が可能になります。関連 AWS サービスを備えた高可用性の最新システムのベストプラクティスについては、ホワイトペーパー「Availability and Beyond: Understanding and Improving the Resilience of Distributed Systems on AWS」を参照してください。

ディザスタリカバリ

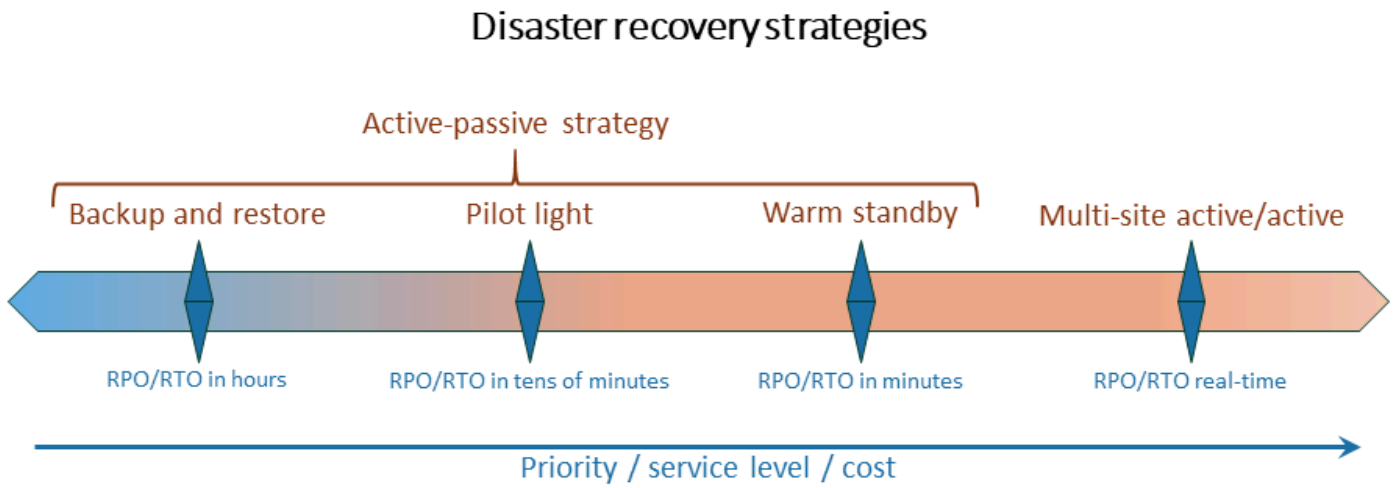
ディザスタリカバリとは、ハードウェアやソフトウェアの重大な障害など、テクノロジー関連の災害に備え、そこから復旧するプロセスを指します。マイクロサービスまたは MES が主要なデプロイ先でビジネス目標を達成できないようにするイベントは、災害と見なされます。ディザスタリカバリは可用性とは異なり、次の 2 つのメトリクスによって測定されます。

- 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間の許容可能な遅延。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。



これらの戦略 AWS の実装に関する詳細なガイダンスについては、「 Well-Architected Framework guide, Disaster Recovery of Workloads on AWS: Recovery in the Cloud」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ダンス、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ダンス)
  - 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 規範ガイダンス)
  - 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

の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

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。