aws

でのエージェント AI フレームワーク、プロトコル、ツール AWS

AWS 規範ガイダンス

AWS 規範ガイダンス: でのエージェント AI フレームワーク、プロトコル、ツール AWS

Table of Contents

でのエージェント AI フレームワーク、プロトコル、ツール AWS

Aaron Sempf と Joshua Samuel、Amazon Web Services (AWS)

2025 年 7 月 (ドキュメント履歴)

エージェント AI は、AI、分散システム、ソフトウェアエンジニアリングの共通部分における強力なパラダイムです。これは、自律的で非同期のソフトウェアエージェントで構成されるインテリジェントシステムのクラスです。エージェントは代理人として行動し、コンテキスト、目標に対する理由を把握し、意思決定を行い、ユーザーやシステムに代わって意図的なアクションを実行できます。これらのエージェントは、分散環境内で独立して、多くの場合共同で動作し、インテリジェンス、メモリ、インテントが組み込まれた委任された目標を追求するように設計されています。

では AWS、エージェント AI を活用して複雑なワークフローを自動化し、意思決定プロセスを強化し、より応答性の高いシステムを作成できます。このガイドでは、効果的なエージェント AI ソリューションを構築するために必要な主要なコンポーネントについて説明します。

- エージェント AI フレームワークは、利点とユースケースのレビューなど、現在のエージェント AI フレームワークをプロファイリングします。これらのフレームワークにより、パターン、プロトコル、ツール間で差別化されていない重労働がどのように軽減されるかについて説明します。主要な選択基準を理解し、要件に適したフレームワークを選択します。
- エージェントプロトコルは、エージェントとのやり取りに不可欠な標準化された通信プロトコルを探索します。Agent-to-agent プロトコルは、オープンソースの Model Context Protocol (MCP) や Agent2Agent (A2A) などの独自の実装とともに登場しています。一般的なプロトコルで、さまざまなプロトコルがシームレスにやり取りする方法について説明します。
- ツールには、プロトコルベースのツール (MCP など) 、フレームワークネイティブツール、メタツールに関する情報が用意されています。組織は、ワークフロー内のキーシステムと統合するツールキットを構築し、エンドユーザーとサーバーベースのエージェントワークフローの両方を実現できます。

対象者

このガイドは、最新のクラウドネイティブアプリケーション内で AI 駆動型ソフトウェアエージェントの能力を活用しようとしているアーキテクト、デベロッパー、テクノロジーリーダーを対象としています。

目的

このガイドは以下を行う際に役立ちます。

- さまざまなエージェント AI フレームワークを比較して、ユースケースに最も適したフレームワークを選択します。
- 持続可能なエージェント AI アーキテクチャを構築するためのオープンプロトコルの利点を理解します。
- エージェントシステムを構築するときに、適切なツール統合戦略を作成します。

このコンテンツシリーズについて

このガイドは、AI 駆動型ソフトウェアエージェントを構築するためのアーキテクチャ設計図と技術ガイダンスを提供する一連の出版物の一部です AWS。 AWS 規範ガイダンスシリーズには以下が含まれます。

- でのエージェント AI の運用 AWS
- でのエージェント AI の基礎 AWS
- でのエージェント AI のパターンとワークフロー AWS
- のエージェント AI フレームワーク、プロトコル、ツール AWS (このガイド)
- でのエージェント AI 用のサーバーレスアーキテクチャの構築 AWS
- でのエージェント AI 用のマルチテナントアーキテクチャの構築 AWS

このコンテンツシリーズの詳細については、「エージェント AI」を参照してください。

エージェント AI フレームワーク

のエージェント AI の基礎 AWSは、自律的で目標指向の動作を可能にするコアパターンとワークフローを調べます。これらのパターンを実装する中心にあるのは、フレームワークの選択です。フレームワークは、本番環境対応の自律型 AI エージェントの構築に必要な構造、ツール、オーケストレーション機能を提供するソフトウェア基盤です。

効果的なエージェント AI フレームワークは、未加工の大規模言語モデル (LLM) インタラクションを、独立したオペレーションが可能な堅牢で自律的なエージェントに変換するいくつかの重要な機能を提供します。

- エージェントオーケストレーションは、単一または複数のエージェント間の情報の流れと意思決定を調整し、人間の介入なしに複雑な目標を達成します。
- ツール統合により、エージェントは外部システム、APIs、データソースとやり取りして、言語処理を超えて機能を拡張できます。詳細については、 Strands Agentsドキュメントの「ツールの概要」を参照してください。
- メモリ管理は、永続的またはセッションベースの状態を提供し、インタラクション全体でコンテキストを維持します。これは、長時間実行される自律タスクに不可欠です。詳細については、LangChain ブログの「エージェントフレームワークについて考える方法」を参照してください。
- ワークフロー定義は、高度な自律推論を可能にするチェーン、ルーティング、並列化、リフレクションループなどの構造化パターンをサポートします。
- デプロイとモニタリングにより、自動システムのオブザーバビリティを備えた開発から本番稼働への移行が容易になります。詳細については、LangGraph「プラットフォーム GA の発表」を参照してください。

これらの機能は、フレームワーク環境全体でさまざまなアプローチと重点を置いて実装され、それぞれがさまざまな自律型エージェントのユースケースと組織コンテキストに異なる利点を提供します。

このセクションでは、エージェント AI ソリューションを構築するための主要なフレームワークをプロファイリングして比較し、その強み、制限、自律運用の理想的なユースケースに焦点を当てます。

- ストランドエージェント
- LangChain と LangGraph
- CrewAI
- Amazon Bedrock Agents

- AutoGen
- エージェント AI フレームワークの比較

> **Note**
>
> このセクションでは、AI の機関を特にサポートするフレームワークについて説明し、機関のないフロントエンドインターフェイスや生成 AI については説明しません。

Strands Agents

Strands Agents は、オープンソースAWS ブログで説明されているように AWS、 によって最初にリリースされたオープンソース SDK です。 Strands Agentsは、モデルファーストのアプローチで自律型 AI エージェントを構築するように設計されています。 とシームレスに連携するように設計された柔軟で拡張可能なフレームワークを提供し AWS のサービス 、サードパーティーのコンポーネントとの統合にオープンなままです。Strands エージェントは、完全自律型ソリューションの構築に最適です。

の主な機能 Strands Agents

Strands Agents には、次の主要な機能が含まれています。

- モデルファースト設計 – 基盤モデルがエージェントインテリジェンスの中核であるという概念に基づいて構築され、高度な自律型推論を可能にします。詳細については、 Strands Agentsドキュメントの「エージェントループ」を参照してください。
- MCP 統合 – Model Context Protocol (MCP) のネイティブサポートにより、LLMs、一貫した自律オペレーションが可能になります。
- AWS のサービス 統合 – Amazon Bedrock、 など AWS のサービス へのシームレスな接続により AWS Lambda AWS Step Functions、包括的な自律型ワークフローを実現します。詳細については、AWS 「週次ラウンドアップ (ブログ)」を参照してください。AWS
- 基盤モデルの選択 – Amazon Bedrock の Anthropic Claude、Amazon Nova (Premier、Pro、Lite、Micro) など、さまざまな基盤モデルをサポートして、さまざまな自律推論機能を最適化します。詳細については、 Strands Agentsドキュメントの「Amazon Bedrock」を参照してください。

- LLM API 統合 – Amazon Bedrock、OpenAI など、本番デプロイ用のさまざまな LLM サービスインターフェイスとの柔軟な統合。詳細については、 Strands Agentsドキュメントの「Amazon Bedrock の基本的な使用方法」を参照してください。
- マルチモーダル機能 – テキスト、音声、画像処理を含む複数のモダリティをサポートし、包括的な自律型エージェントインタラクションを実現します。詳細については、 Strands Agentsドキュメントの「Amazon Bedrock マルチモーダルサポート」を参照してください。
- ツールエコシステム – AWS のサービス 対話のための豊富なツールセット。自律機能を拡張するカスタムツールの拡張性を備えています。詳細については、 Strands Agentsドキュメントの「ツールの概要」を参照してください。

Strands Agents を使用する場合

Strands Agents は、次のような自律型エージェントのシナリオに特に適しています。

- 自動ワークフロー AWS のサービス のために とのネイティブ統合を必要とする AWS インフラストラクチャ上に構築される組織
- 本番稼働用自律システムのエンタープライズグレードのセキュリティ、スケーラビリティ、コンプライアンス機能を必要とするチーム
- 特殊な自律タスクのために異なるプロバイダー間でモデル選択に柔軟性を必要とするプロジェクト
- エンドツーエンドの自律プロセスのために既存の AWS ワークフローやリソースと緊密に統合する必要があるユースケース

の実装アプローチ Strands Agents

Strands Agents は、クイックスタートガイドで説明されているように、ビジネスステークホルダーに簡単な実装アプローチを提供します。このフレームワークにより、組織は次のことが可能になります。

- 特定のビジネス要件に基づいて、Amazon Bedrock の Amazon Nova (Premier、Pro、Lite、Micro) などの基盤モデルを選択します。
- エンタープライズシステムとデータソースに接続するカスタムツールを定義します。
- テキスト、画像、音声など、複数のモダリティを処理します。
- ビジネスクエリに自律的に応答し、タスクを実行できるエージェントをデプロイします。

この実装アプローチにより、ビジネスチームは AI モデル開発に関する深い技術的専門知識を必要とせずに、自律型エージェントを迅速に開発およびデプロイできます。

の実例 Strands Agents

AWS Transform for .NET Strands Agentsは、AWS Transform 「 for .NET」で説明されているように、 を使用してアプリケーションのモダナイゼーション機能を強化します。これは、.NET アプリケーションを大規模にモダナイズするための最初のエージェント AI サービスです (AWS ブログ)。この本稼働サービスは、複数の特殊な自律型エージェントを採用しています。エージェントは連携して、レガシー .NET アプリケーションを分析し、モダナイゼーション戦略を計画し、人間の介入なしにクラウドネイティブアーキテクチャへのコード変換を実行します。 AWS Transform for .NET は、エンタープライズ自律システムの Strands Agentsの本番稼働準備状況を示しています。

LangChain および LangGraph

LangChain は、エージェント AI エコシステムで最も確立されたフレームワークの 1 つです。 は、LangChain ブログで説明されているように、その機能LangGraphを拡張して、複雑でステートフルなエージェントワークフローをサポートします。これらを組み合わせることで、独立した運用のための豊富なオーケストレーション機能を備えた高度な自律型 AI エージェントを構築するための包括的なソリューションを提供します。

LangChain および の主な機能 LangGraph

LangChain および には、次の主要な機能LangGraphが含まれています。

- コンポーネントエコシステム – さまざまな自律型エージェント機能用の構築済みコンポーネントの広範なライブラリ。特殊なエージェントの迅速な開発を可能にします。詳細については、LangChain のドキュメントを参照してください。
- 基盤モデルの選択 – Anthropic Claude、Amazon Bedrock の Amazon Nova モデル (Premier、Pro、Lite、Micro) など、さまざまな推論機能のためのさまざまな基盤モデルをサポートします。詳細については、 LangChainドキュメントの「入出力」を参照してください。
- LLM API 統合 – Amazon Bedrock、 など、柔軟なデプロイOpenAIのための複数の大規模言語モデル (LLM) サービスプロバイダー向けの標準化されたインターフェイス。詳細については、LangChainドキュメントのLLMs」を参照してください。
- マルチモーダル処理 – テキスト、画像、オーディオ処理のサポートが組み込まれており、リッチなマルチモーダル自律型エージェントインタラクションを可能にします。詳細については、LangChainドキュメントの「マルチモダリティ」を参照してください。

- グラフベースのワークフロー – 複雑な自律型エージェントの動作をステートマシンとして定義LangGraphし、高度な決定ロジックをサポートできます。詳細については、LangGraph「プラットフォーム GA の発表」を参照してください。
- メモリ抽象化 – 短期および長期のメモリ管理のための複数のオプション。これは、時間の経過とともにコンテキストを維持する自律型エージェントに不可欠です。詳細については、 LangChainドキュメントの「チャットボットにメモリを追加する方法」を参照してください。
- ツール統合 – さまざまなサービスと APIs、自律型エージェント機能を拡張します。詳細については、 LangChainドキュメントの「ツール」を参照してください。
- LangGraph プラットフォーム – 長時間稼働の自律型エージェントをサポートする、本番環境向けのマネージドデプロイおよびモニタリングソリューション。詳細については、LangGraph「プラットフォーム GA の発表」を参照してください。

LangChain と を使用するタイミング LangGraph

LangChain と LangGraphは、次のような自律型エージェントのシナリオに特に適しています。

- 自律的な意思決定に高度なオーケストレーションを必要とする複雑な複数ステップの推論ワークフロー
- さまざまな自律機能のための構築済みコンポーネントと統合の大規模なエコシステムへのアクセスを必要とするプロジェクト
- 自動システムの構築を希望する既存の Pythonベースの機械学習 (ML) インフラストラクチャと専門知識を持つチーム
- 長時間実行される自律型エージェントセッション全体で複雑な状態管理を必要とするユースケース

LangChain と の実装アプローチ LangGraph

LangChain と は、 LangGraphドキュメントに詳述されているように、ビジネスステークホルダーに構造化された実装アプローチLangGraphを提供します。このフレームワークにより、組織は次のことが可能になります。

- ビジネスプロセスを表す高度なワークフローグラフを定義します。
- 決定ポイントと条件付きロジックを使用して、複数ステップの推論パターンを作成します。
- マルチモーダル処理機能を統合して、さまざまなデータ型を処理します。
- 組み込みのレビューおよび検証メカニズムを使用して、品質管理を実装します。

このグラフベースのアプローチにより、ビジネスチームは複雑な意思決定プロセスを自律型ワークフローとしてモデル化できます。チームは、推論プロセスの各ステップと、決定パスを監査する能力を明確に把握できます。

と の実際の例 LangChainLangGraph

Vodafone は、LangChainエンタープライズのケーススタディで説明されているように、 LangChain (および LangGraph) を使用して自動エージェントを実装し、データエンジニアリングとオペレーションのワークフローを強化しています。パフォーマンスメトリクスを自律的にモニタリングし、ドキュメントシステムから情報を取得し、自然言語インタラクションを通じて実用的なインサイトを提示する内部 AI アシスタントを構築しました。

このVodafone実装では、LangChainモジュラードキュメントローダー、ベクトル統合、および複数の LLMs (OpenAI、LLaMA3、および Gemini) のサポートを使用して、これらのパイプラインを迅速にプロトタイプ化およびベンチマークします。次に、モジュラーサブエージェントLangGraphをデプロイしてマルチエージェントオーケストレーションを構造化するために を使用しました。これらのエージェントは、収集、処理、要約、推論タスクを実行します。 APIs を通じてこれらのエージェントをクラウドシステムLangGraphに統合しました。

CrewAI

CrewAI は、特に GitHub で利用可能な自律型マルチエージェントオーケストレーションに焦点を当てたオープンソースフレームワークです。これは、人間の介入なしに複雑なタスクを解決するために協力する特殊な自律型エージェントのチームを作成するための構造化されたアプローチを提供します。 CrewAI は、ロールベースの調整とタスクの委任を強調します。

の主な機能 CrewAI

CrewAI には、次の主要な機能があります。

- ロールベースのエージェント設計 – 自律エージェントは、特殊な専門知識を実現するために、特定のロール、目標、バックストーリーで定義されます。詳細については、 CrewAIドキュメントの「Crafting Effective Agents」を参照してください。
- タスクの委任 – 機能に基づいて適切なエージェントにタスクを自動的に割り当てるための組み込みメカニズム。詳細については、 CrewAIドキュメントの「タスク」を参照してください。
- エージェントコラボレーション – 人間による仲介のない、エージェント間の自律的なコミュニケーションと知識共有のためのフレームワーク。詳細については、 CrewAIドキュメントの「コラボレーション」を参照してください。

- プロセス管理 – シーケンシャルおよび並列の自律タスク実行のための構造化ワークフロー。詳細については、 CrewAIドキュメントの「プロセス」を参照してください。
- 基盤モデルの選択 – Anthropic Claude、Amazon Bedrock の Amazon Nova モデル (Premier、Pro、Lite、Micro) など、さまざまな基盤モデルをサポートし、さまざまな自律推論タスクを最適化します。詳細については、 CrewAIドキュメントのLLMs」を参照してください。
- LLM API 統合 – Amazon Bedrock、、ローカルモデルのデプロイなどOpenAI、複数の LLM サービスインターフェイスとの柔軟な統合。詳細については、 CrewAIドキュメントの「プロバイダー設定の例」を参照してください。
- マルチモーダルサポート – テキスト、画像、その他のモダリティを処理して、包括的な自律型エージェントインタラクションを実現するための新しい機能。詳細については、 CrewAIドキュメントの「マルチモーダルエージェントの使用」を参照してください。

CrewAI を使用する場合

CrewAI は、次のような自律型エージェントのシナリオに特に適しています。

- 専門的なロールベースの専門知識を自律的に活用できる複雑な問題
- 複数の自律型エージェント間の明示的なコラボレーションを必要とするプロジェクト
- チームベースの問題分解が自律的な問題解決を改善するユースケース
- さまざまな自律エージェントロール間で懸念を明確に分離する必要があるシナリオ

の実装アプローチ CrewAI

CrewAI は、 CrewAIドキュメントの「開始方法」で説明されているように、ビジネスステークホルダー向けの AI エージェントアプローチのチームのロールベースの実装を提供します。このフレームワークにより、組織は次のことが可能になります。

- 特定のロール、目標、専門知識領域を持つ特殊な自律型エージェントを定義します。
- 特殊な機能に基づいてエージェントにタスクを割り当てます。
- タスク間に明確な依存関係を確立して、構造化ワークフローを作成します。
- 複数のエージェント間のコラボレーションを調整して、複雑な問題を解決します。

このロールベースのアプローチは人間のチーム構造を反映しており、ビジネスリーダーは直感的に理解して実装できます。組織は、ヒューマンチームの運営方法と同様に、ビジネス目標を達成するため

に協力する専門の専門分野を持つ自律型チームを作成できます。ただし、自律型チームは人間の介入なしに継続的に作業できます。

の実例 CrewAI

AWS は、CrewAI公開されたケーススタディで説明されているように、Amazon Bedrock と統合された CrewAI を使用して自律型マルチエージェントシステムを実装 AWS し、ベンダーに依存しない安全なフレームワークCrewAIを開発しました。CrewAI オープンソースの「フローとクルー」アーキテクチャは、Amazon Bedrock 基盤モデル、メモリシステム、コンプライアンスガードレールとシームレスに統合されます。

実装の主な要素は次のとおりです。

- 設計図とオープンソース – AWS および は、CrewAIエージェントを Amazon Bedrock モデルとオブザーバビリティツールにマッピングするリファレンスデザインをCrewAIリリースしました。https://aws.amazon.com/blogs/machine-learning/build-agentic-systems-with-crewai-and-amazon-bedrock/また、マルチエージェント AWS セキュリティ監査クルー、コードモダナイゼーションフロー、コンシューマーパッケージ製品 (CPG) バックオフィスオートメーションなどのサンプルシステムもリリースしました。
- オブザーバビリティスタックの統合 – このソリューションは、Amazon CloudWatch、AgentOps、および によるモニタリングを埋め込みLangFuse、概念実証から本番稼働までのトレーサビリティとデバッグを可能にします。
- 実証済みの投資収益率 (ROI) – 初期のパイロットでは、大規模なコードモダナイゼーションプロジェクトの実行が 70% 高速化され、CPG バックオフィスフローの処理時間が約 90% 短縮されました。

Amazon Bedrock Agents

Amazon Bedrock エージェントは、アプリケーションで自律型エージェントを構築および設定できるフルマネージドサービスです。基盤モデル、データソース、ソフトウェアアプリケーション、ユーザーとの会話間のやり取りをオーケストレーションできます。エージェントの作成に対する合理化されたアプローチでは、容量のプロビジョニング、インフラストラクチャの管理、カスタムコードの記述を行う必要はありません。

Amazon Bedrock エージェントの主な機能

Amazon Bedrock エージェントには、次の主要な機能があります。

- フルマネージドサービス – キャパシティをプロビジョニングしたり、基盤となるシステムを管理したりすることなく、インフラストラクチャ管理を完了します。詳細については、Amazon Bedrock ドキュメントの「AI エージェントを使用してアプリケーションのタスクを自動化する」を参照してください。
- API 駆動型開発 – モデル、手順、ツール、設定パラメータを指定して、シンプルな API コールを通じてエージェントを定義して実行します。詳細については、Amazon Bedrock ドキュメントの「エージェントを手動で作成および設定する」を参照してください。
- アクショングループ – API スキーマを使用してアクショングループを作成して、エージェントが実行できる特定のアクションを定義します。詳細については、Amazon Bedrock ドキュメントの「アクショングループを使用してエージェントが実行するアクションを定義する」を参照してください。
- ナレッジベースの統合 – Amazon Bedrock ナレッジベースにシームレスに接続して、組織のデータでエージェントのレスポンスを強化します。詳細については、Amazon Bedrock ドキュメントの「ナレッジベースを使用したエージェントの Augment レスポンス生成」を参照してください。
- 高度なプロンプトテンプレート – 前処理、オーケストレーション、ナレッジベースのレスポンス生成、後処理用のプロンプトテンプレートを使用してエージェントの動作をカスタマイズします。詳細については、Amazon Bedrock ドキュメントの「Amazon Bedrock の高度なプロンプトテンプレートを使用したエージェントの精度の向上」を参照してください。
- トレースとオブザーバビリティ – 組み込みトレース機能を使用して、エージェントのstep-by-stepの推論プロセスを追跡します。詳細については、Amazon Bedrock ドキュメントの「トレースを使用してエージェントのstep-by-stepの推論プロセスを追跡する」を参照してください。
- バージョニングとエイリアス – エージェントの複数のバージョンを作成し、制御されたロールアウトのエイリアスを介してデプロイします。詳細については、「Amazon Bedrock ドキュメント」の「アプリケーションで Amazon Bedrock エージェントをデプロイして使用する」を参照してください。

Amazon Bedrock エージェントを使用するタイミング

Amazon Bedrock エージェントは、次のような自律型エージェントのシナリオに特に適しています。

- インフラストラクチャを管理せずにエージェントを構築およびデプロイするためのフルマネージドエクスペリエンスを必要とする組織
- コードではなく設定を通じてエージェントの迅速な開発とデプロイを必要とするプロジェクト
- ナレッジベースやガードレールなどの他の Amazon Bedrock 機能との緊密な統合からメリットを得るユースケース

- エージェントをゼロから構築するための社内リソースがないが、本番環境対応の自律機能が必要なチーム

Amazon Bedrock エージェントの実装アプローチ

Amazon Bedrock エージェントは、ビジネスステークホルダー向けに設定ベースの実装アプローチを提供します。このサービスにより、組織は次のことが可能になります。

- 複雑なコードを記述せずに、 AWS Management Console または API コールを通じてエージェントを定義します。
- エージェントが実行できる APIsとオペレーションを指定するアクショングループを作成します。
- ナレッジベースを接続して、ドメイン固有の情報をエージェントに提供します。
- ビジュアルインターフェイスを使用して、エージェントの動作をテストして反復処理します。

このマネージド型アプローチにより、ビジネスチームは AI モデル開発やインフラストラクチャ管理に関する深い技術的専門知識を必要とせずに、自律型エージェントを迅速に開発およびデプロイできます。

Amazon Bedrock エージェントの実際の例

このAWS ブログ記事で説明されている財務オペレーション (FinOps) ソリューションは、Amazon Bedrock マルチエージェントフレームワークを使用して AI 主導のクラウドコスト管理アシスタントを作成します。費用対効果の高い Amazon Nova 基盤モデルは、中央の FinOps スーパーバイザーエージェントがタスクを専門エージェントに委任するソリューションを強化します。これらのエージェントは、 を使用して AWS 支出データを取得および分析 AWS Cost Explorer し、 を使用してコスト削減の推奨事項を生成します AWS Trusted Advisor。

このシステムには、Amazon Cognito を介した安全なユーザーアクセス、 でホストされるフロントエンド AWS Amplify、リアルタイム分析と予測のための AWS Lambda アクショングループが含まれています。財務チームは、「2025 年 2 月のコストはいくらだったか」などの自然言語クエリを聞くことができます。システムは、詳細な内訳、最適化の提案、予測で応答します。これらはすべて、 を使用してデプロイされたスケーラブルなサーバーレスアーキテクチャ内で行われます AWS CloudFormation。

AutoGen

AutoGen は、 によって最初にリリースされたオープンソースフレームワークですMicrosoft。 は、会話型および協調型の自律型 AI エージェントを有効にすることにAutoGen重点を置いています。これは、複雑な自律ワークフローのためにエージェント間の非同期のイベント駆動型インタラクションに重点を置いたマルチエージェントシステムを構築するための柔軟なアーキテクチャを提供します。

の主な機能 AutoGen

AutoGen には、次の主要な機能があります。

- 会話エージェント – 自律型エージェント間の自然言語会話を中心に構築され、対話を通じて高度な推論を可能にします。詳細については、 AutoGenドキュメントの「マルチエージェント会話フレームワーク」を参照してください。
- 非同期アーキテクチャ – ノンブロッキングの自律型エージェントインタラクションのためのイベント駆動型設計で、複雑な並列ワークフローをサポートします。詳細については、 AutoGenドキュメントの「非同期チャットのシーケンスでの複数のタスクの解決」を参照してください。
- Human-in-the-loop – 必要に応じて、他の自律型エージェントワークフローへのオプションの人間参加を強力にサポートします。詳細については、 AutoGenドキュメントの「 エージェントでのヒューマンフィードバックの許可」を参照してください。
- コードの生成と実行 — コードを記述して実行できるコードに重点を置いた自律型エージェント専用の機能。詳細については、 AutoGenドキュメントの「コード実行」を参照してください。
- カスタマイズ可能な動作 – さまざまなユースケースに対応する柔軟な自律型エージェント設定と会話制御。詳細については、 AutoGenドキュメントの agentchat.conversable_agent を参照してください。
- 基盤モデルの選択 – Amazon Bedrock の Anthropic Claude、Amazon Nova モデル (Premier、Pro、Lite、Micro) など、さまざまな基盤モデルをサポートし、さまざまな自律推論機能を提供します。詳細については、 AutoGenドキュメントの「LLM 設定」を参照してください。
- LLM API 統合 – Amazon Bedrock、、 などOpenAI、複数の LLM サービスインターフェイスの標準化された設定Azure OpenAI。詳細については、 AutoGen API リファレンスの「oai.openai_utils」を参照してください。
- マルチモーダル処理 – リッチなマルチモーダル自律型エージェントインタラクションを可能にするテキストおよび画像処理をサポートします。詳細については、 AutoGenドキュメントの「マルチモーダルモデルの使用: GPT-4VAutoGen」を参照してください。

AutoGen を使用する場合

AutoGen は、次のような自律型エージェントシナリオに特に適しています。

- 複雑な推論のために自律型エージェント間で自然な会話フローを必要とするアプリケーション
- 完全自律型オペレーションとオプションの人的監視機能の両方を必要とするプロジェクト
- 人間の介入なしでの自律的なコード生成、実行、デバッグを含むユースケース
- 柔軟で非同期の自律型エージェントの通信パターンを必要とするシナリオ

の実装アプローチ AutoGen

AutoGen は、「 AutoGenドキュメント」の「開始方法https://microsoft.github.io/autogen/docs/Getting-Started」で説明されているように、ビジネスステークホルダー向けの会話型実装アプローチを提供します。このフレームワークにより、組織は次のことが可能になります。

- 自然言語の会話を通じて通信する自律型エージェントを作成します。
- 複数のエージェント間でイベント駆動型の非同期インタラクションを実装します。
- 必要に応じて、完全自律型オペレーションとオプションの人間による監視を組み合わせます。
- 対話を通じてコラボレーションするさまざまなビジネス機能に特化したエージェントを開発します。

この会話型アプローチは、自律システムの推論をビジネスユーザーが透過的に利用できるようにします。意思決定者は、エージェント間の対話を観察して結論にどのように達するかを理解し、必要に応じて人間の判断が必要なときに会話に参加できます。

の実例 AutoGen

Magentic-One は、Microsoft AI Frontiers ブログで説明されているように、さまざまな環境で複雑なマルチステップタスクを自律的に解決するように設計されたオープンソースのジェネラリストマルチエージェントシステムです。その中核となるのはオーケストレーターエージェントです。オーケストレーターエージェントは、高レベルの目標を分解し、構造化台帳を使用して進捗状況を追跡します。このエージェントは、サブタスクを特殊なエージェント (WebSurfer、、FileSurfer、 などComputerTerminal) に委任しCoder、必要に応じて再計画することで動的に適応します。

システムはAutoGenフレームワーク上に構築され、モデルに依存しず、デフォルトで GPT-4o になります。タスク固有の調整WebArenaなしでAssistantBench、GAIA、、 などのベンチマーク全体で最

先端のパフォーマンスを実現します。さらに、提案によるモジュール式の拡張性と厳格な評価もサポートしていますAutoGenBench。

エージェント AI フレームワークの比較

自律型エージェント開発用のエージェント AI フレームワークを選択するときは、各オプションが特定の要件にどのように適合するかを検討してください。技術的な能力だけでなく、チームの専門知識、既存のインフラストラクチャ、長期的なメンテナンス要件など、組織の適合性も考慮してください。多くの組織は、自律型 AI エコシステムのさまざまなコンポーネントに複数のフレームワークを活用して、ハイブリッドアプローチの恩恵を受ける可能性があります。

次の表は、主要な技術的側面における各フレームワークの成熟度レベル (最強、強、適切、弱) を比較したものです。この表には、フレームワークごとに、本番デプロイオプションと学習曲線の複雑さに関する情報も含まれています。

Framework	AWS 統合	自動マルチエージェントサポート	自律ワークフローの複雑さ	マルチモーダル機能	基盤モデルの選択	LLM API 統合	本番稼働用デプロイ	学習曲線
Amazon BedrockAgents	最も強い	適切	適切	強力	強力	強力	フルマネージド型	低
AutoGen	弱い	強力	強力	適切	適切	強力	自分で行う (DIY)	急勾配
CrewAI	弱い	強力	適切	弱い	適切	適切	DIY	中
LangChain / LangGraph	適切	強力	最も強い	最も強い	最も強い	最も強い	プラットフォームまたは DIY	急勾配

Strands Agents	最も強い	強力	最も強い	強力	強力	最も強い	DIY	中

エージェント AI フレームワークの選択に関する考慮事項

自律型エージェントを開発するときは、次の主要な要因を考慮してください。

- AWS インフラストラクチャ統合 – に大きく投資されている組織は AWS 、自律型ワークフロー AWS のサービス のために Strands Agentsと のネイティブ統合から最も恩恵を受けます。詳細については、AWS 「週次ラウンドアップ (ブログ)」を参照してください。AWS
- 基盤モデルの選択 – 自律型エージェントの推論要件に基づいて、優先基盤モデル (Amazon Bedrock の Amazon Nova モデルや Anthropic Claude など) に最適なサポートを提供するフレームワークを検討します。詳細については、 Anthropicウェブサイトの「効果的なエージェントの構築」を参照してください。
- LLM API 統合 – 本番デプロイ用の任意の大規模言語モデル (LLM) サービスインターフェイス (Amazon Bedrock や などOpenAI) との統合に基づいてフレームワークを評価します。詳細については、 ドキュメントの Strands Agents「モデルインターフェイス」を参照してください。
- マルチモーダル要件 – テキスト、画像、音声を処理する必要がある自律型エージェントの場合は、各フレームワークのマルチモーダル機能を検討してください。詳細については、 LangChain ドキュメントの「マルチモダリティ」を参照してください。
- 自律型ワークフローの複雑さ — 高度な状態管理を備えたより複雑な自律型ワークフローは、 の高度なステートマシン機能を優先する可能性がありますLangGraph。
- 自律型チームコラボレーション – 専門エージェント間の明示的なロールベースの自律型コラボレーションを必要とするプロジェクトは、 のチーム指向アーキテクチャからメリットを得ることができますCrewAI。
- 自律型開発パラダイム – 自律型エージェントの会話型非同期パターンを好むチームは、 のイベント駆動型アーキテクチャを好むかもしれませんAutoGen。
- マネージド型またはコードベースのアプローチ – 最小限のコーディングでフルマネージド型のエクスペリエンスを希望する組織は、Amazon Bedrock エージェントを検討する必要があります。より詳細なカスタマイズを必要とする組織は、特定の自律型エージェントの要件により適した特殊な機能を備えた Strands Agentsまたは他のフレームワークを優先する場合があります。
- 自律システムの本番稼働準備 — 本番稼働用自律エージェントのデプロイオプション、モニタリング機能、エンタープライズ機能を検討します。

エージェントプロトコル

AI エージェントは、他のエージェントやサービスとやり取りするために標準化された通信プロトコルを必要とします。エージェントアーキテクチャを実装している組織は、相互運用性、ベンダーの独立性、投資の将来性に関して大きな課題に直面しています。

このセクションでは、柔軟性と相互運用性を最大化するオープンスタンダードに焦点を当ててagent-to-agentプロトコルランドスケープをナビゲートするのに役立ちます。(agent-to-toolプロトコルの詳細については、このガイドの後半にある「ツール統合戦略」を参照してください)。

このセクションでは、2024 Anthropic 年に によって最初に開発されたオープンスタンダードである Model Context Protocol (MCP) について説明します。現在、 はプロトコルの開発と実装に貢献することで MCP AWS を積極的にサポートしています。 AWS は、、LangGraph、 CrewAIなどの主要なオープンソースエージェントフレームワークと協力してLlamaIndex、プロトコルでのエージェント間通信の未来を形成しています。詳細については、「Open Protocols for Agent Interoperability Part 1: Inter-Agent Communication on MCP」(AWS ブログ) を参照してください。

このセクションの内容

- プロトコルの選択が重要な理由
- Agent-to-agentプロトコル
- エージェントプロトコルの選択
- エージェントプロトコルの実装戦略
- MCP の開始方法

プロトコルの選択が重要な理由

プロトコルの選択は、AI エージェントアーキテクチャを構築および進化する方法を根本的に形成します。エージェントフレームワーク間の移植性をサポートするプロトコルを選択することで、特定のニーズに合わせてさまざまなエージェントシステムとワークフローを柔軟に組み合わせることができます。

オープンプロトコルを使用すると、エージェントを複数のフレームワークに統合できます。たとえば、 LangChainを使用してラピッドプロトタイプを作成し、 を使用して本番稼働用システムを実装しStrands Agents、MCP や Agent2Agent (A2A) プロトコルなどの一般的なプロトコルを介して通信します。この柔軟性により、特定の AI プロバイダーへの依存を減らし、既存のシステムとの統合を簡素化し、時間の経過とともにエージェント機能を強化できます。

また、適切に設計されたプロトコルは、エージェントエコシステム全体で認証と認可のための一貫したセキュリティパターンを確立します。最も重要なのは、プロトコルの移植性により、新しいエージェントのフレームワークと機能の導入の自由が維持されることです。オープンプロトコルを選択すると、サードパーティーシステムとの相互運用性を維持しながら、エージェント開発への投資が保護されます。

オープンプロトコルの利点

独自の拡張機能を実装する場合やカスタムエージェントシステムを構築する場合、オープンプロトコルには魅力的な利点があります。

- ドキュメントと透明性 — 通常、包括的なドキュメントと透過的な実装を提供します。
- コミュニティサポート – トラブルシューティングとベストプラクティスのためのより広範な開発者コミュニティへのアクセス
- 相互運用性の保証 – 拡張機能がさまざまな実装で機能する保証を強化
- 将来の互換性 — 変更が中断されたり、廃止されたりするリスクを軽減
- 開発への影響 — プロトコルの進化に貢献する機会

Agent-to-agentプロトコル

次の表は、複数のエージェントがコラボレーション、タスクの委任、および情報の共有を可能にするエージェントプロトコルの概要を示しています。

[プロトコル]	に最適	考慮事項
MCP エージェント間通信	柔軟なエージェントコラボレーションパターンを求める組織	• エージェントagent-to-agent通信の既存の基盤を構築する AWS 、 によって提案されたモデルコンテキストプロトコル (MCP) の拡張 • OAuth ベースのセキュリティによるシームレスなエージェントコラボレーションを実現
A2A プロトコル	クロスプラットフォームエージェントエコシステム	• によるバックアップ Google

		• MCP と比較して導入が限られている新しい標準
AutoGen マルチエージェント	研究に焦点を当てたマルチエージェントシステム	• によってバックアップ Microsoft • 複雑なエージェントインタラクションに強い
CrewAI	ロールベースのエージェントチーム	• 独立した実装 • 組織構造のシミュレートに適しています

プロトコルオプション間の決定

agent-to-agent通信を実装する場合は、特定の通信要件を適切なプロトコル機能と一致させます。異なるインタラクションパターンには、異なるプロトコル機能が必要です。次の表は、一般的な通信パターンの概要を示し、各シナリオに最適なプロトコルの選択を推奨しています。

パターン:	説明	理想的なプロトコルの選択
シンプルなリクエストとレスポンス	エージェント間の 1 回限りのインタラクション	ステートレスフローを持つ MCP
ステートフルな対話	コンテキストを使用した継続的な会話	セッション管理による MCP
マルチエージェントコラボレーション	複数のエージェント間の複雑なやり取り	MCP エージェント間または AutoGen
チームベースのワークフロー	ロールが定義された階層エージェントチーム	MCP エージェント間、CrewAI、または AutoGen

コミュニケーションパターン以外にも、いくつかの技術的および組織的な要因がプロトコルの選択に影響を与える可能性があります。次の表は、特定の実装要件に最も近いプロトコルを評価するのに役立つ重要な考慮事項の概要を示しています。

考慮事項	説明	例
セキュリティモデル	認証と認可の要件	MCP の OAuth 2.0
デプロイ環境	エージェントが実行して通信する場所	分散マシンまたは単一マシン
エコシステムの互換性	既存のエージェントフレームワークとの統合	LangChain、または Strands Agents
スケーラビリティのニーズ	エージェントインタラクションの予想される増加	MCP のストリーミング機能

エージェントプロトコルの選択

本番稼働用エージェントシステムを構築するほとんどの組織では、モデルコンテキストプロトコル (MCP) は、agent-to-agent通信の最も包括的で十分にサポートされている基盤を提供します。MCP は、 AWS とオープンソースコミュニティからの積極的な開発貢献からメリットを得ます。

適切なエージェントプロトコルを選択することは、エージェント AI を効果的に実装したい組織にとって重要です。考慮事項は、組織のコンテキストによって異なります。

エンタープライズプロトコルに関する考慮事項

企業は以下のアクションを検討する必要があります。

- 戦略的で長期的なエージェント実装のために、MCP などのオープンプロトコルを優先します。
- 将来の移行を容易にするために独自のプロトコルを使用する場合は、抽象化レイヤーを実装します。
- プロトコルの進化に影響を与えるための標準開発に参加します。
- コアインフラストラクチャにはオープンプロトコルを使用し、特定のユースケースには独自のプロトコルを使用するハイブリッドアプローチを検討してください。

起動プロトコルと SMB プロトコルに関する考慮事項

スタートアップ企業やsmall-to-medium (SMB) 企業は、以下のアクションを検討する必要があります。

- 迅速な開発のために、十分にサポートされている独自のプロトコルから始めて、速度と柔軟性のバランスを取ります。
- ニーズが成熟するにつれて、よりオープンな標準を使用するように移行パスを計画します。
- プロトコル導入の傾向を評価し、規格の低下に投資しないようにします。
- プロトコルの複雑さを抽象化するマネージドサービスを検討してください。

政府の規制された業界プロトコルに関する考慮事項

政府および規制対象業界は、以下のアクションを検討する必要があります。

- オープンスタンダードを強調して、長期的なアクセスを確保し、ベンダーのロックインを回避します。
- 強力なセキュリティモデルと認証メカニズムを使用してプロトコルに優先順位を付けます。
- リモートデプロイモデルとローカルデプロイモデルによるデータ主権の影響を考慮します。
- コンプライアンスとガバナンスの要件に関するプロトコルの決定を文書化します。

エージェントプロトコルの実装戦略

組織全体でエージェントプロトコルを効果的に実装するには、以下の戦略的ステップを検討してください。

1. 標準の調整から始める – 可能な場合は、確立されたオープンプロトコルを採用します。
2. 抽象化レイヤーの作成 – システムと特定のプロトコルの間にアダプターを実装します。
3. オープンスタンダードに貢献 — プロトコル開発コミュニティに参加します。
4. プロトコルの進化をモニタリングする – 新しい標準と更新について常に情報を得ます。
5. 相互運用性を定期的にテストする — 実装に互換性があることを確認します。

MCP の開始方法

エージェントアーキテクチャにモデルコンテキストプロトコル (MCP) を実装するには、次のアクションを実行します。

1. Strands Agents SDK などのフレームワークでの MCP 実装について説明します。
2. Model Context Protocol の技術ドキュメントを確認してください。

3. エージェント相互運用性のオープンプロトコル パート 1: MCP でのエージェント間通信 (AWS ブログ) を読んで、エージェントの相互運用性について学びます。
4. MCP コミュニティに参加して、プロトコルの進化に影響を与えます。

MCP は、エージェントが外部データやサービスとやり取りできるようにする通信レイヤーを提供し、エージェントが他のエージェントとやり取りできるようにするのにも使用できます。プロトコルの Streamable HTTP トランスポート実装により、デベロッパーはホイールを再考案することなく、包括的な一連のインタラクションパターンを使用できます。これらのパターンは、ステートレスリクエスト/レスポンスフローと永続的 IDs を使用したステートフルセッション管理の両方をサポートします。

MCP などのオープンプロトコルを採用することで、AI テクノロジーの進化に合わせて柔軟性、相互運用性、適応性を維持するエージェントシステムを構築するように組織を配置できます。

ツール

AI エージェントは、外部ツール、APIs、データソースを操作して有用なタスクを実行することで価値を提供します。適切なツール統合戦略は、エージェントの機能、セキュリティ体制、長期的な柔軟性に直接影響します。

このセクションでは、自由と柔軟性を最大化するオープンスタンダードに焦点を当てて、ツール統合の状況をナビゲートするのに役立ちます。このセクションでは、ツール統合用の Model Context Protocol (MCP) に焦点を当て、エージェントワークフローを強化するフレームワーク固有のツールと特殊なメタツールを確認します。

このセクションの内容

ツールカテゴリ

エージェントシステムの構築には、主に 3 つのカテゴリのツールが含まれます。

プロトコルベースのツール

プロトコルベースのツールは、agent-to-tool間の通信に標準化されたプロトコルを使用します。

- MCP ツール – ローカル実行オプションとリモート実行オプションの両方でフレームワーク間で機能する標準ツールを開きます。
- OpenAI 関数呼び出し — OpenAIモデルに固有の独自のツール。
- Anthropic ツール – Anthropic Claude モデルに固有の独自のツール。

フレームワークネイティブツール

フレームワークネイティブツールは、特定のエージェントフレームワークに直接組み込まれています。

- Strands Agents Python ツール – Strands Agentsフレームワークに固有の軽量でquick-to-implementツール。
- LangChain ツール – LangChainエコシステムと緊密に統合された Pythonベースのツール。
- LlamaIndex ツール – 内のデータの取得と処理に最適化されたツールLlamaIndex。

メタツール

メタツールは、外部アクションを直接実行することなく、エージェントワークフローを強化します。

- ワークフローツール – エージェント実行フロー、分岐ロジック、状態管理を管理します。
- エージェントグラフツール – 複雑なワークフローで複数のエージェントを調整します。
- メモリツール – エージェントセッション全体で永続的なストレージと情報の取得を提供します。
- リフレクションツール – エージェントが独自のパフォーマンスを分析し、改善できるようにします。

プロトコルベースのツール

プロトコルベースのツールを検討する場合、モデルコンテキストプロトコル (MCP) はツール統合の最も包括的で柔軟な基盤を提供します。AWS エージェントの相互運用性に関するオープンソースブログ記事で述べたように、 AWS は MCP を戦略的プロトコルとして採用し、その開発に積極的に貢献しています。

次の表に、MCP ツールのデプロイのオプションを示します。

デプロイモデル	説明	に最適	実装
ローカル stdio ベース	エージェントと同じプロセスで実行されるツール	開発、テスト、シンプルなツール	ネットワークオーバーヘッドなしで迅速に実装

ローカルサーバー送信イベント (SSE) ベース	ツールはローカルで実行されますが、HTTP 経由で通信します	懸念を分離したより複雑なローカルツール	分離は向上するが、レイテンシーは低い
リモート SSE ベース	リモートサーバーで実行されるツール	本番環境と共有ツール	スケーラブルで一元管理

公式の Model Context Protocol SDKs は MCP ツールの構築に使用できます。

- Python SDK – 完全なプロトコルサポートによる包括的な実装
- TypeScript SDK – ウェブアプリケーションの JavaScript/TypeScript 実装
- Java SDK – エンタープライズアプリケーションの Java 実装

これらの SDKsは、プロトコル仕様の一貫した実装により、任意の言語で MCP 互換ツールを作成するための構成要素を提供します。

さらに、 AWS は Strands Agents SDK に MCP を実装しています。Strands Agents SDK を使用すると、MCP 互換ツールを簡単に作成して使用できます。包括的なドキュメントは、Strands AgentsGitHub リポジトリで入手できます。より簡単なユースケースやStrands Agentsフレームワーク外で作業する場合、公式 MCP SDKs は複数の言語でプロトコルを直接実装します。

MCP ツールのセキュリティ機能

MCP ツールのセキュリティ機能には以下が含まれます。

- OAuth 2.0/2.1 認証 – 業界標準認証
- アクセス許可のスコープ — ツールのきめ細かなアクセスコントロール
- ツール機能検出 — 使用可能なツールの動的検出
- 構造化エラー処理 – 一貫したエラーパターン

MCP ツールの開始方法

ツール統合用の MCP を実装するには、次のアクションを実行します。

1. 本番環境対応の MCP 実装については、 Strands Agents SDK をご覧ください。
2. MCP 技術ドキュメントを確認して、主要な概念を理解します。

3. このAWS オープンソースブログ記事で説明されている実用的な例を使用します。
4. リモートツールに進む前に、シンプルなローカルツールから始めます。
5. MCP コミュニティに参加して、プロトコルの進化に影響を与えます。

フレームワークネイティブツール

Model Context Protocol (MCP) は最も柔軟な基盤を提供しますが、フレームワークネイティブツールは特定のユースケースに利点を提供します。

Strands Agents SDK は、シンプルなオペレーションに最小限のオーバーヘッドを必要とする軽量設計を特徴とする Pythonベースのツールを提供します。これにより、迅速な実装が可能になり、開発者はわずか数行のコードでツールを作成できます。さらに、Strands Agentsフレームワーク内でシームレスに機能するように緊密に統合されています。

次の例は、 を使用してシンプルな気象ツールを作成する方法を示していますStrands Agents。開発者は、最小限のコードオーバーヘッドでPython関数をエージェントアクセス可能なツールにすばやく変換し、関数の docstring から適切なドキュメントを自動的に生成できます。

```
#Example of a simple Strands native tool

@tool

def weather(location: str) -> str:

"""Get the current weather for a location""" #

Implementation here

return f"The weather in {location} is sunny."
```

ラピッドプロトタイピングやシンプルなユースケースでは、フレームワークネイティブツールが開発を加速できます。ただし、本稼働システムでは、MCP ツールはフレームワークネイティブツールよりも相互運用性と将来の柔軟性に優れています。

次の表は、他のフレームワーク固有のツールの概要を示しています。

Framework	ツールタイプ	利点	考慮事項
AutoGen	関数定義	強力なマルチエージェントサポート	Microsoft エコシステム

LangChain	Python クラス	構築済みのツールの大規模なエコシステム	フレームワークのロックイン
LlamaIndex	Python 関数	データオペレーション用に最適化	に制限 LlamaIndex

メタツール

メタツールは外部システムと直接やり取りしません。代わりに、エージェントパターンを実装することでエージェントの機能を強化します。このセクションでは、ワークフロー、エージェントグラフ、メモリメタツールについて説明します。

ワークフローメタツール

ワークフローメタツールは、エージェント実行のフローを管理します。

- 状態管理 – 複数のエージェントインタラクションのコンテキストを維持する
- 分岐ロジック – 条件付き実行パスを有効にする
- 再試行メカニズム – 高度な再試行戦略で障害を処理する

ワークフローメタツールを使用したフレームワークの例には、 LangGraphおよび Strands Agents ワークフロー機能が含まれます。

エージェントグラフのメタツール

エージェントグラフのメタツールは、複数のエージェントを連携させて調整します。

- タスクの委任 – 専門エージェントにサブタスクを割り当てる
- 結果の集約 – 複数のエージェントからの出力を結合する
- 競合の解決 – エージェント間の不一致を解決する

AutoGen や などのフレームワークは、エージェントグラフの調整にCrewAI特化しています。

メモリメタツール

メモリメタツールは、永続的なストレージと取り出しを提供します。

- 会話履歴 – セッション間でコンテキストを維持する
- ナレッジベース – ドメイン固有の情報の保存と取得
- ベクトルストア – セマンティック検索機能を有効にする

MCP のリソースシステムは、さまざまなエージェントフレームワークで動作するメモリメタツールを実装するための標準化された方法を提供します。

ツール統合戦略

ツール統合戦略の選択は、エージェントが達成できることとシステムの進化の容易さに直接影響します。フレームワークネイティブツールとメタツールを戦略的に使用しながら、モデルコンテキストプロトコル (MCP) などのオープンプロトコルを優先します。これにより、AI テクノロジーの進歩に合わせて柔軟で強力なツールエコシステムを構築できます。

ツール統合に対する以下の戦略的アプローチは、組織の当面のニーズを満たしながら柔軟性を最大化します。

1. 基盤として MCP を採用する – MCP は、強力なセキュリティ機能を持つツールにエージェントを接続するための標準化された方法を提供します。MCP を主なツールプロトコルとして開始します。
   - 複数のエージェント実装で使用される戦略的ツール。
   - 堅牢な認証と認可を必要とするセキュリティ重視のツール。
   - 本番環境でリモート実行が必要なツール。
2. 必要に応じてフレームワークネイティブツールを使用する – 以下のフレームワークネイティブツールを検討します。
   - 初期開発中の迅速なプロトタイピング。
   - セキュリティ要件を最小限に抑えたシンプルな重要ではないツール。
   - 独自の機能を活用するフレームワーク固有の機能。
3. 複雑なワークフローにメタツールを実装する – メタツールを追加してエージェントアーキテクチャを強化します。
   - 基本的なワークフローパターンから簡単に開始します。
   - ユースケースが成熟するにつれて複雑さを追加します。
   - エージェントとメタツール間のインターフェイスを標準化します。
4. 進化の計画 – 将来の柔軟性を念頭に置いて構築します。

- 実装とは無関係にツールインターフェイスを文書化します。
- エージェントとツールの間に抽象化レイヤーを作成します。
- 独自のプロトコルからオープンプロトコルへの移行パスを確立します。

ツール統合のセキュリティのベストプラクティス

ツール統合は、セキュリティ体制に直接影響します。このセクションでは、組織で考慮すべきベストプラクティスの概要を説明します。

認証と認可

次の堅牢なアクセスコントロールを使用します。

- OAuth 2.0/2.1 を使用する – リモートツールに業界標準の認証を実装します。
- 最小特権を実装する – ツールに必要なアクセス許可のみを付与します。
- 認証情報のローテーション – API キーとアクセストークンを定期的に更新します。

データ保護

データを保護するために、以下の対策を実施してください。

- 入力と出力を検証する – すべてのツールインタラクションにスキーマ検証を実装します。
- 機密データの暗号化 – すべてのリモートツール通信に TLS を使用します。
- データ最小化を実装する – 必要な情報のみをツールに渡します。

モニタリングと監査

次のメカニズムを使用して、可視性と制御を維持します。

- すべてのツール呼び出しをログに記録する – 包括的な監査証跡を維持します。
- 異常をモニタリングする – 異常なツールの使用パターンを検出します。
- レート制限の実装 — 過剰なツール呼び出しによる不正使用を防止します。

MCP セキュリティモデルは、これらの懸念に包括的に対処します。詳細については、MCP ドキュメントの「セキュリティに関する考慮事項」を参照してください。

結論

エージェント AI の環境は急速に進化し続けており、組織はインテリジェントで自律的なシステムを構築するための強力な新しい方法を提供します。このガイドでは、実装を成功させるための 3 つの重要なコンポーネント、基盤を提供するフレームワーク、通信を可能にするプロトコル、機能を拡張するツールについて説明しました。

フレームワークが成熟するにつれて、相互運用性の向上、モデルコンテキストプロトコル (MCP) などのプロトコルの標準化、自律型エージェントのより高度なオーケストレーション機能が期待できます。現在、これらのフレームワークに関する専門知識を確立している組織は、ビジネスに大きな価値をもたらす、ますます自律的でインテリジェントなエージェントを構築する体制を整えることができます。

エージェントプロトコルの選択は、即時の開発ニーズと長期的な柔軟性および相互運用性のバランスを取る戦略的決定を表します。オープンプロトコルを優先し、適切な抽象化レイヤーを作成することで、組織は現在のビジネス要件を満たしながら、進化するテクノロジーに適応可能なエージェントシステムを構築できます。

ほとんどの組織にとって、MCP はオープンスタンダード、成長するエコシステム、agent-to-agent 通信パターンのサポート、ツール統合機能による強固な基盤です。 AWS は MCP を戦略的プロトコルとして採用し、Strands AgentsSDK などのサービス全体での開発と実装に積極的に貢献しています。MCP を適切なフレームワークネイティブツールやメタツールとともに使用することで、将来のイノベーションに適応しながらすぐに価値をもたらすエージェントシステムを構築できます。

リソース

次のリソース AWS と、自律型エージェント開発に関連するその他のリソースを使用します。

AWS ブログ

- Amazon Bedrock エージェントを使用して堅牢な生成 AI アプリケーションを構築するためのベストプラクティス – パート 1
- Amazon Bedrock エージェントを使用して堅牢な生成 AI アプリケーションを構築するためのベストプラクティス – パート 2
- オープンソース AI エージェント SDK Strands Agentsの紹介
- エージェント相互運用性のオープンプロトコル パート 1: MCP でのエージェント間通信
- AWS Transform for .NET、.NET アプリケーションを大規模にモダナイズするための最初のエージェント AI サービス
- AWS 毎週の切り上げ: Strands Agents

AWS 規範ガイダンス

- でのエージェント AI の運用 AWS
- でのエージェント AI の基礎 AWS
- でのエージェント AI のパターンとワークフロー AWS
- でのエージェント AI 用のサーバーレスアーキテクチャの構築 AWS
- でのエージェント AI 用のマルチテナントアーキテクチャの構築 AWS
- で拡張生成オプションとアーキテクチャを取得する AWS

AWS リソース

- Amazon Bedrock ドキュメント
- Amazon Nova ドキュメント
- AWS MCP サーバー (GitHub)

その他のリソース

- AutoGen ドキュメント (Microsoft)
- 効果的なエージェントの構築 (Anthropic)
- CrewAI GitHub リポジトリ
- LangChain ドキュメント
- LangGraph プラットフォーム
- Model Context Protocol ドキュメント
- Strands Agents ドキュメント
- Strands Agents ツールの概要
- Strands Agents クイックスタートガイド

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
初版発行	—	2025 年 7 月 14 日

AWS 規範ガイダンスの用語集

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

数字

7 Rs

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

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

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

A

ABAC

「属性ベースのアクセスコントロール」を参照してください。

抽象化されたサービス

「 マネージドサービス」を参照してください。

ACID

アトミック性、一貫性、分離性、耐久性を参照してください。

アクティブ - アクティブ移行

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

アクティブ - パッシブ移行

ソースデータベースとターゲットデータベースが同期されるデータベース移行方法。ただし、ソースデータベースのみが、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、接続アプリケーションからのトランザクションを処理します。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

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

AI

「人工知能」を参照してください。

AIOps

「人工知能オペレーション」を参照してください。

匿名化

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

アンチパターン

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

アプリケーションコントロール

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

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

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

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

アベイラビリティーゾーン

他のアベイラビリティーゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティーゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドへの移行を成功させるための効率的で効果的な計画を立て AWS るための、 のガイドラインとベストプラクティスのフレームワーク。 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に関するガイダンスを提供します。詳細については、AWS CAF ウェブサイト と AWS CAF のホワイトペーパー を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

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

B

不正なボット

個人や組織を混乱させたり、損害を与えたりすることを意図したボット。

BCP

事業継続計画を参照してください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。エンディアン性も参照してください。

二項分類

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

ブルームフィルター

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

ブルー/グリーンデプロイ

2 つの異なる同一の環境を作成するデプロイ戦略。現在のアプリケーションバージョンは 1 つの環境 (青) で実行し、新しいアプリケーションバージョンは別の環境 (緑) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティや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、ボットハーダーまたはボットオペレーターとして知られる 1 人の当事者が管理しているボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

ブランチ

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

ブレークグラスアクセス

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

ブラウンフィールド戦略

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

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

AWS 「クラウド導入フレームワーク」を参照してください。

Canary デプロイ

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

CCoE

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

CDC

「データキャプチャの変更」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストします。AWS Fault Injection Service （ AWS FIS ） を使用して、 AWS ワークロードにストレスを与え、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。

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テクノロジーに接続されています。

クラウド運用モデル

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

導入のクラウドステージ

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

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

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

「設定管理データベース」を参照してください。

コードリポジトリ

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

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオなどのビジュアル形式から情報を分析および抽出する AI の分野。例えば、Amazon SageMaker AI は CV 用の画像処理アルゴリズムを提供します。

設定ドリフト

ワークロードの場合、設定が想定状態から変化します。これにより、ワークロードが非準拠になる可能性があり、通常は段階的かつ意図的ではありません。

構成管理データベース (CMDB)

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

コンフォーマンスパック

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

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「継続的デリバ

リーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

一元管理とガバナンスを備えた分散型の分散型データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された 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、通常、大量の履歴データが含まれており、クエリや分析に使用されます。

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

ディープラーニング

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

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリ

ティの手法。この戦略を採用するときは AWS、 AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加して、リソースの安全性を確保します。たとえば、多層防御アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

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

デプロイ

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

開発環境

「環境」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

スタースキーマでは、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する

離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに一般的に使用されます。

ディザスタ

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

ディザスタリカバリ (DR)

災害によるダウンタイムとデータ損失を最小限に抑えるために使用する戦略とプロセス。詳細については、 AWS Well-Architected フレームワークの「 Disaster Recovery of Workloads on AWS: Recovery in the Cloud」を参照してください。

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

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

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

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

DVSM

「開発値ストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

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、応答時間を短縮できます。

電子データ交換 (EDI)

組織間の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プロセス。

暗号化キー

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

エンディアン

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

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink 、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これら

のアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの「エンドポイントサービスを作成する」を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (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

星スキーマの中央テーブル。事業運営に関する量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 つのタイプの列が含まれます。

フェイルファスト

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

障害分離の境界

では AWS クラウド、障害の影響を制限し、ワークロードの耐障害性を高めるのに役立つ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。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

同様のタスクの実行を求める前に、タスクと必要な出力を示す少数の例を LLM に提供します。この手法は、プロンプトに埋め込まれた例 (ショット) からモデルが学習するコンテキスト内学習のアプリケーションです。少数ショットプロンプトは、特定のフォーマット、推論、またはドメインの知識を必要とする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コントロール」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

FM

「基盤モデル」を参照してください。

基盤モデル (FM)

一般化およびラベル付けされていないデータの大規模なデータセットでトレーニングされている大規模な深層学習ニューラルネットワーク。FMs は、言語の理解、テキストと画像の生成、自然言語の会話など、さまざまな一般的なタスクを実行できます。詳細については、「基盤モデルとは」を参照してください。

G

生成 AI

大量のデータでトレーニングされ、シンプルなテキストプロンプトを使用してイメージ、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る AI モデルのサブセット。詳細については、「生成 AI とは」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 ワークフローはレガシーと見なされ、トランクベースのワークフローはモダンで推奨されるアプローチです。

ゴールデンイメージ

そのシステムまたはソフトウェアの新しいインスタンスをデプロイするためのテンプレートとして使用されるシステムまたはソフトウェアのスナップショット。例えば、製造では、ゴールデンイメージを使用して複数のデバイスにソフトウェアをプロビジョニングし、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、、 AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

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

ハイアベイラビリティ (HA)

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

ヒストリアンのモダナイゼーション

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

ホールドアウトデータ

機械学習モデルのトレーニングに使用されるデータセットから保留される、ラベル付きの履歴データの一部。モデル予測をホールドアウトデータと比較することで、ホールドアウトデータを使用してモデルのパフォーマンスを評価できます。

同種データベースの移行

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

ホットデータ

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

ホットフィックス

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

ハイパーケア期間

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

I

IaC

「Infrastructure as Code」を参照してください。

ID ベースのポリシー

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

アイドル状態のアプリケーション

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

IIoT

「産業用モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (受信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングする VPC。AWS Security Reference Architecture では、アプリ

ケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

増分移行

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

インダストリー 4.0

2016 年に Klaus Schwab によって導入された用語で、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ゼーションを指します。

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。

産業分野における IoT (IIoT)

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

インスペクション VPC

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

モノのインターネット (IoT)

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

解釈可能性

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

IoT

「モノのインターネット」を参照してください。

IT 情報ライブラリ (ITIL)

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

大量のデータに対して事前トレーニングされた深層学習 AI モデル。LLM は、質問への回答、ドキュメントの要約、テキストの他の言語への翻訳、文の完了など、複数のタスクを実行できます。詳細については、LLMs」を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースのアクセスコントロール」を参照してください。

最小特権

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

リフトアンドシフト

「7 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。エンディアン性も参照してください。

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またはプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスにつながる可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS 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 を操作し、エンドポイントにアクセスしてデータを保存および取得します。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステムで、原材料を工場の完成製品に変換します。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作時にそれ自体を強化および改善するサイクルです。詳細については、 AWS 「 Well-Architected フレームワーク」の「メカニズムの構築」を参照してください。

メンバーアカウント

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

MES

「製造実行システム」を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

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

マイクロサービス

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

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

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)

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

移行準備状況評価 (MRA)

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

移行戦略

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

ML

???「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

モノリシックアプリケーション (モノリス)

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

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

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

多クラス分類

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

ミュータブルインフラストラクチャ

本番ワークロードの既存のインフラストラクチャを更新および変更するモデル。Well-Architected AWS 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

「オリジンアクセスコントロール」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織の変更管理」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

「 オペレーションの統合」を参照してください。

OLA

「運用レベルの契約」を参照してください。

オンライン移行

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

OPC-UA

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

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問および関連するベストプラクティスのチェックリスト。詳細については、 AWS Well-Architected フレームワークの「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するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、Industry 4.0 変換の主な焦点です。

オペレーション統合 (OI)

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

組織の証跡

組織 AWS アカウント 内のすべての のすべてのイベント AWS CloudTrail をログに記録する、 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。

オリジンアクセスコントロール (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、すべての S3 バケット AWS リージョン、 AWS KMS (SSE-KMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETEリクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

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

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用テクノロジー」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

「プログラム可能なロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

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

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

アクションを実行し AWS 、リソースにアクセスできる のエンティティ。このエンティティは通常、、IAM 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内にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通じてプライバシーを考慮するシステムエンジニアリングアプローチ。

プライベートホストゾーン

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

プロアクティブコントロール

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

製品ライフサイクル管理 (PLM)

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

本番環境

「環境」を参照してください。

プログラム可能なロジックコントローラー (PLC)

製造では、マシンをモニタリングし、製造プロセスを自動化する、信頼性の高い適応可能なコンピュータです。

プロンプトの連鎖

1 つの LLM プロンプトの出力を次のプロンプトの入力として使用して、より良いレスポンスを生成します。この手法は、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善または拡張したりするために使用されます。これにより、モデルのレスポンスの精度と関連性が向上し、より詳細でパーソナライズされた結果が得られます。

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。

パブリッシュ/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。スケーラビリティと応答性を向上させます。たとえば、マイクロサービスベースの MES では、マイクロサービスは他のマイクロサー

ビスがサブスクライブできるチャネルにイベントメッセージを発行できます。システムは、公開サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

RAG

「取得拡張生成」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

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

「行と列のアクセスコントロール」を参照してください。

リードレプリカ

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

再設計

「7 Rs」を参照してください。

目標復旧時点 (RPO)

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

目標復旧時間 (RTO)

サービスの中断から復旧までの最大許容遅延時間。

リファクタリング

「7 Rs」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョン は、耐障害性、安定性、耐障害性を提供するために、他の から分離され、独立しています。詳細については、AWS リージョン 「アカウントで使用できる を指定する」を参照してください。

回帰

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

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 Rs」を参照してください。

プラットフォーム変更

「7 Rs」を参照してください。

再購入

「7 Rs」を参照してください。

回復性

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

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンシブコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、Implementing security controls on AWSのResponsive controlsを参照してください。

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

取得拡張生成 (RAG)

LLM がレスポンスを生成する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を参照する生成 AI テクノロジー。例えば、RAG モデルは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する場合があります。詳細については、「RAG とは」を参照してください。

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「目標復旧時点」を参照してください。

RTO

目標復旧時間を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーティッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、 AWS Management Console にログインしたり 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

「監視コントロールとデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

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

設計によるセキュリティ

開発プロセス全体でセキュリティを考慮するシステムエンジニアリングアプローチ。

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、予防的、検出的、応答的、プロアクティブの 4 つの主なタイプがあります。

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修復するように設計された、事前定義されたプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動応答アクションの例としては、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

送信先にあるデータの、それ AWS のサービス を受け取る による暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

のエントリポイントの URL AWS 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、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

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

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について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、 はクラウドのセキュリティを担当します。詳細については、責任共有モデルを参照してください。

SIEM

セキュリティ情報とイベント管理システムを参照してください。

単一障害点 (SPOF)

システムを中断する可能性のあるアプリケーションの 1 つの重要なコンポーネントの障害。

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

スプリットアンドシードモデル

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

SPOF

単一障害点を参照してください。

スタースキーマ

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

strangler fig パターン

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

サブネット

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

監視コントロールとデータ収集 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと本番稼働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潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。Amazon CloudWatch Synthetics を使用して、これらのテストを作成できます。

システムプロンプト

LLM にコンテキスト、指示、またはガイドラインを提供して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、ユーザーとのやり取りのルールを確立するのに役立ちます。

T

tags

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

ターゲット変数

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

タスクリスト

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

テスト環境

「環境」を参照してください。

トレーニング

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

トランジットゲートウェイ

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

トランクベースのワークフロー

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

信頼されたアクセス

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

チューニング

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

ツーピザチーム

2 枚のピザで養う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。詳細については、深層学習システムにおける不確実性の定量化 ガイドを参照してください。

未分化なタスク

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

上位環境

「環境」を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング接続

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

ワークロード

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

ワークストリーム

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

WORM

「書き込み 1 回」、「読み取り多数」を参照してください。

WQF

AWS 「ワークロード認定フレームワーク」を参照してください。

Write Once, Read Many (WORM)

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

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性を利用する攻撃、通常はマルウェア。

ゼロデイ脆弱性

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

ゼロショットプロンプト

LLM にタスクを実行する手順を提供しますが、タスクのガイドに役立つ例 (ショット) はありません。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。