aws

でのエージェント AI の運用 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 の運用 AWS

Table of Contents

でのエージェント AI の運用 AWS

Aaron Sempf、Brad Ryan、Bhargs Srivathsan、Akhil Bhaskar、Amazon Web Services

2025 年 8 月 (ドキュメント履歴)

エージェント AI は機能ではなく、新しい運用パラダイムです。統制のとれたアーキテクチャ、信頼フレームワーク、ビジネスに沿ったデプロイモデルに投資する組織は、次世代のアダプティブでインテリジェントな企業をリードします。

エージェント AI は、自律型ソフトウェアエージェントと生成 AI の収束を表します。エージェントの意思決定と目標指向の動作を、大規模言語モデル (LLMs。これらのエージェントは、動的なエンタープライズ環境全体で推論、行動、適応、コラボレーションを行うことができます。この可能性を実現するために、企業は考え方をモデルデプロイからエージェントインフラストラクチャに移行する必要があります。

このガイドでは、エージェント AI を独立した実験からエンタープライズ規模の価値を生み出すインフラストラクチャに変換する組織戦略を提供します。これは、ガバナンス、スケーラビリティ、ビジネスアラインメントを備えたワークフロー全体にインテリジェントエージェントを埋め込むのに役立ちます。

主な重点領域と推奨事項

このガイドでは、エージェント AI を運用する際の以下の基本領域に焦点を当てています。各重点分野について、組織およびビジネスの推奨事項が提供されます。

- フォーカスエリア 1: エージェントのインテントとスコープを明確にする – エージェントをビジネスの優先順位と認識のボトルネックに合わせます。エージェントをツールとしてだけでなく、デジタルチームメイトとして扱います。
- 重点領域 2: コンポジビリティとコラボレーションの設計 – モジュラーアーキテクチャ、セマンティックプロトコル、およびアービターエージェントによる動的委任を備えたマルチエージェントシステムを採用します。
- フォーカスエリア 3: マルチテナンシーと制御のためのアーキテクト – 共有エージェントサービス、一元化されたガバナンス、ロールベースのアクセスを使用して、スケーラブルなテナント対応インフラストラクチャを構築します。
- フォーカスエリア 4: ID、ガードレール、オブザーバビリティを通じて信頼を構築する – トレーサビリティ、ランタイムコントロール、説明可能性を適用して、ステークホルダーの信頼を獲得します。

- フォーカスエリア 5: ライフサイクルを管理する – 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、プロンプトバージョニング、テレメトリ、継続的再トレーニングを確立して、エージェント AI のパフォーマンスと効率をサポートします。
- 重点領域 6: エージェントモデルをビジネスモデルと整合させる – 使用状況ベースのモデル、内部 ROI メトリクス、商用サービスを通じてエージェントの機能を収益化します。

このガイドの推奨事項を使用して、エージェント AI を大規模に準備できます。エージェント向け DevOps (AgentOps) チームの構築、相互運用可能なシステム、導入を拡大する変更管理戦略など、組織がエージェント AI を中心に再構築する方法について説明します。Well-Architected AWS フレームワークとの決定優先の考え方と整合性を強調しています。

対象者

このガイドは、エージェントシステムを設計およびスケーリングし、AI をコアビジネスワークフローに組み込み、本番環境で LLMs と自律エージェントを運用しているエンタープライズアーキテクト、AI/ML エンジニアリングリーダー、デジタルトランスフォーメーション戦略家を対象としています。このガイドの概念と推奨事項を理解するには、最新のクラウドネイティブアーキテクチャと分散システム、大規模言語モデル、基盤モデル機能、AI ガバナンス、DevOps、プラットフォームエンジニアリングの原則に精通している必要があります。

目的

このガイドの推奨事項を実装することで、組織は次のビジネス成果を達成できます。

- 人間のボトルネックと認知負荷を軽減する、目標指向の自律型エージェントによる意思決定とワークフロー実行の高速化。
- 再利用可能なマルチテナントエージェントプラットフォームによる、ビジネスユニット間のインテリジェント機能のスケーラブルでコスト効率の高いデプロイ。
- AI システムの耐障害性、信頼、ガバナンスの向上により、規制対象、ミッションクリティカル、または顧客向け環境での確実な導入が可能になります。

このコンテンツシリーズについて

このガイドは、 でのエージェント AI に関するシリーズの一部です AWS。詳細およびこのシリーズの他のガイドについては、 AWS 「 規範ガイダンス」ウェブサイトの 「エージェント AI 」を参照してください。

エージェント AI の戦略的基盤

エージェントシステムは新しいものではありません。ロボットプロセスオートメーション (RPA) や決定エンジンなどのソフトウェアエージェントは、数十年前から存在していました。しかし、これらはシンプルで決定論的であり、事前定義されたルールとシンボリックロジックに従って反復的で変動の少ないタスクを実行するように設計されています。生成 AI の増加に伴い、ゲームは変わりました。大規模言語モデル (LLMs) は、複雑な入力を解釈し、レスポンスを動的に生成し、知識をすばやく合成できるようになりました。脆弱またはハードコードされたロジックなしでエージェンシーをスケールできるようになりました。これで、エージェントは推論を行い、意思決定を行い、ツールを呼び出し、コンテキストに適応し、ワークフロー全体で他のエージェントと調整できます。目標に向かって自律的に動作し、メモリを維持し、成果を反映できます。

ただし、raw 機能だけでは不十分です。統合のないインテリジェンスは、影響ではなく新規性をもたらします。強力な LLMs から価値を引き出すには、企業は独立した実験を超えてエンジニアリングされたエコシステムに移行する必要があります。エージェントは、あらゆるエンタープライズシステムと同じ分野で動作する本番稼働用サービスとして扱う必要があります。これには、ガバナンス、オブザーバビリティ、安全な ID モデル、ライフサイクル管理が含まれます。また、投機的な可能性ではなく、実際のビジネス成果をもたらす必要があります。これらのシステムは、意思決定と耐障害性のために明確な境界で設計する必要があります。自動復旧メカニズム、リアルタイムのパフォーマンスモニタリング、スケーラブルなリソース管理を組み込むことが重要です。これにより、エージェントとのやり取りの動的で非決定的な性質に対処しながら、エンタープライズワークフロー全体で一貫したサービスレベルを維持できます。

基本的なレベルでは、企業は運用の構造にインテリジェンスがどのように埋め込まれているかを再検討する必要があります。エージェントは、コアシステムと統合し、エンタープライズポリシーに準拠し、測定可能な価値を提供するように設計されている必要があります。部門、ドメイン、ユーザーコンテキストを横断して大規模に運用する必要があります。エージェント AI の運用は、最終的には使用に関するものです。分離されたタスクを実行する AI のデプロイと、ビジネスモデルを進化させるエージェントのデプロイの違いです。

エージェント AI は、組織全体でインテリジェンスをスケールするためのシステム、プロセス、人材へのアプローチを根本的に変える必要がある新しい運用哲学を表します。エージェントは、人間の能力を強化する戦略的アセットになります。エージェント AI を運用に統合することで、組織はビジネス価値を高め、人間の能力を強化し、複雑なワークフローを最適化するインサイトを引き出すことができます。

エージェント AI の戦略的重点分野

初期のプロトタイプから本番稼働用グレードのシステムや価値を生み出すシステムに移行するには、チームはアーキテクチャ、プロセス、製品思考をブレンドした一貫した戦略が必要です。

多くの組織は、まだツールファーストまたはモデル中心の考え方で AI にアプローチしています。生成 AI は実験が強化されていますが、多くの場合、ビジネス戦略や測定可能な成果との明確な整合性はありません。戦略的役割を定義しないと、エージェントはスケーラブルな価値を提供するのではなく、リソースを枯渇させる新しい実験になるリスクがあります。エージェント AI の戦略的役割を確立するには、組織はビジネスの優先順位から始める必要があります。自律性が緩和できる認知過負荷、決定のボトルネック、または断片化されたワークフローの領域を特定します。ドメイン固有の問題ステートメントを使用して、エージェントの責任を形成します。エージェントをツールではなくデジタルチームメイトとして扱います。エージェントは推論、委任、適応を行うことができます。

決定科学は、データサイエンス、分析、行動モデリングを組み合わせて意思決定を改善する分野です。これは、設計をビジネス成果に合わせるために、エージェントアーキテクチャプロセスの早い段階で統合する必要があります。決定パターンを特定し、トレードオフをシミュレートし、価値への影響を定量化することで、決定科学は、エージェントの自律性が最高の価値を提供できる場所を特定するのに役立ちます。決定科学は、意思決定を加速し、エラーを減らし、リアルタイムの適応を可能にします。このデータ情報に基づく基盤は、エージェント設計を測定可能なインサイトに基づいており、ルールエンジン、分析プラットフォーム、予測モデルなどの既存のエンタープライズテクノロジーと緊密に統合できます。

このセクションでは、エージェントの戦略的役割を確立するために、エージェント AI の運用のバックボーンを形成する基本的な重点分野を紹介します。各 は、エージェントの構想と設計に責任を持つ技術リーダー、アーキテクト、または製品所有者の観点からコアジョブにマッピングされます。これらの重点領域はシーケンシャルステップではありません。各 は、システムライフサイクル全体で見直して、回復力があり、スケーラブルで、収益化可能なエージェントエコシステムを育む価値があります。

このセクションでは、以下の重点領域について説明します。

- フォーカスエリア 1: エージェントのインテントとスコープを明確にする
- 重点領域 2: コンポジビリティとコラボレーションの設計
- フォーカスエリア 3: マルチテナンシーと制御のためのアーキテクト
- フォーカスエリア 4: ID、ガードレール、オブザーバビリティを通じて信頼を構築する
- フォーカスエリア 5: ライフサイクルを管理する

- 重点領域 6: エージェントモデルをビジネスモデルと整合させる

フォーカスエリア 1: エージェントのインテントとスコープを明確にする

完了すべきジョブ:「クールデモだけでなく、各エージェントが明確な境界で実際の問題を解決するのに役立つ」。

エージェント AI は、能力を構築することだけではありません。これは、適切な結果を適切な方法で解決することです。これは、エージェント AI ソリューションの意図を完全に明確にすることから始まります。

方針

多くの場合、組織はモデルができることから始め (APIs呼び出し、質問への回答、概要の生成など)、ユースケースを改良します。これにより、スコープクリープ、不十分な統合、および技術的には魅力的だが運用上は役に立たないエージェントが発生します。代わりに、次のような特定の質問を通じてエージェントのロールを定義することから始めます。

- エージェントが担当する具体的な成果は何ですか?
- 誰に代わって行動していますか?
- 誰がメリットを得られますか?
- エージェントの自律性はどこで開始および終了しますか?
- 失敗するとどうなりますか?

適切な範囲のエージェントには、明確な職務、定義された責任、測定可能な成功基準があります。エージェントをアシスタントまたはチャットボットと見なさないでください。代わりに、ジョブタイトルを付けます。これは、カスタマーサクセスエージェント、製品がハンドラーを返す、またはコンプライアンスモニターと考えることができます。

利害関係者や顧客を関与させるときは、エージェント AI システムのスケーラビリティと適応性を強調します。これらのエージェントはビジネスとともに進化し、学習とフィードバックを通じて継続的に改善されます。抵抗を減らし、導入を加速するには、ワーカーの感情を念頭に置いてエージェントツールがどのように設計されているかを強調します。透明性、制御、および信頼を構築するオプションのオーバーライドメカニズムを提供します。エージェントは人間を置き換えるのではなく、人間の

能力と意思決定を強化し、従業員がループにとどまり、価値の高いタスクに集中できるようにします。

実装を成功させるための鍵は、エージェント AI を特定の影響の大きいビジネス成果に合わせることです。チームやパートナーに、目に見える問題点を解決する集中的なパイロットプロジェクトから始めるよう促します。クイックウィンは、測定可能な投資収益率 (ROI) を生成し、内部賛同を構築し、より広範な導入の勢いを生み出します。

導入と成熟を導くために、組織は進化モデルに沿ってエージェント設計をフレーム化できます。エージェントの自律性、複雑さ、ビジネスへの影響は徐々に増加します。このモデルのステージは次のとおりです。

- オブザーバーエージェントはノイズからインサイトを表面化します。例としては、デジタルチャネル全体のブランド認識を追跡する市場感情エージェントがあります。
- アシスタントエージェントは人間の意思決定をサポートします。例としては、販売チームの競合データと市場状況を合成するディールアドバイザリーエージェントがあります。
- 自律エージェントは、定義された境界内で独立して動作します。例えば、需要に基づいてクラウドインフラストラクチャを動的に調整するリソース割り当てエージェントです。
- オーケストレーターエージェントは、マルチエージェントワークフローを調整します。例としては、在庫、物流、予測エージェント間のインタラクションを管理するサプライチェーン最適化エージェントがあります。
- イノベーターエージェントは、新しい戦略的可能性を生み出します。例としては、市場トレンドを分析し、新しい収益ストリームを提案するビジネスモデルイノベーションエージェントがあります。

これらの戦略的成果と成熟度レベルを中心にエージェントをフレーミングすることで、焦点が高まり、導入が加速し、ステークホルダーの信頼が構築されます。

Amazon Quick など AWS のサービス、この重点分野の調整をサポートするために、 はエージェント主導の結果にリンクされた主要業績評価指標 (KPIs) を視覚化できます。Amazon CloudWatch を使用して、エージェントの動作、パフォーマンスメトリクス、システムの状態をほぼリアルタイムでモニタリングできます。運用上のフィードバックを使用して、エージェントのインタラクションとリソースの使用を調整します。 は、初期の実験フェーズと改良フェーズでエージェントのアクティビティと統合パターンを可視化AWS CloudTrailできます。

インテントとスコープを定義するビジネス価値

エージェント AI の導入は、組織がデジタルトランスフォーメーションとオペレーショナルエクセレンスにどのようにアプローチするかにおける重要な変化を表しています。これは自動化に関するものではありません。これは、意思決定と価値の実現を加速するインテリジェントな自律性を可能にすることです。

主要なビジネス推進要因は次のとおりです。

- 競争上の利点 – 早期導入者は、より迅速なインサイト、より良いサービス、適応型オペレーションを通じて戦略的利点を得ることができます。
- カスタマーエクスペリエンスの強化 – エージェントは、満足度と口イヤルティを高める、パーソナライズされた常時稼働のリアルタイムサポートを提供します。
- 運用効率 – エージェント AI は、複雑で反復的な決定タスクを自動化することで、人間の認知負荷を大幅に軽減します。これにより、スタッフが価値の高いアクティビティに集中し、コストを削減できます。

さまざまな業界の実際のユースケースは次のとおりです。

- 金融サービス – AI エージェントは、パーソナライズされた財務アドバイスを提供し、不正を検出できます。
- ヘルスケア – トリアージと治療計画のエージェントは、臨床スループットを向上させることができます。
- 小売 – エージェントはインテリジェントなショッピングアシスタントとして機能したり、インベントリをリアルタイムで最適化したりできます。
- 製造 – エージェントは、予測メンテナンスを実行したり、サプライチェーンを調整したりできます。

重点領域 2: コンポジビリティとコラボレーションの設計

完了すべきジョブ:「サービスを構築するようにエージェントをモジュール化してテスト可能にし、必要に応じて構成してオーケストレーションできるようにします」

多くの AI の取り組みは、モノリシックでモデル中心のパイロットとして始まります。これらは便利ですが、ドメイン間でスケーリングしたり、複雑な問題に適応したりするのは困難です。これらのエージェントが相互運用するように設計されている場合のバリューコンパウンド。テクノロジーで

は、コンポーザビリティとは、モジュラーコンポーネントを組み合わせて、変化に適応できる柔軟でスケーラブルなソリューションを作成する行為です。コンポーザビリティがない場合、インテリジェンスは特定のワークフロー内でロックされます。さらに、エージェントのコラボレーションでは、オーケストレーション、状態管理、プロトコルネゴシエーションの複雑さが導入され、従来のオートメーションチームでは処理できない場合があります。

方針

マルチエージェントパラダイムを採用します。組織部門などのモデルエージェント: モジュラー、専門、相互運用可能。モデルコンテキストプロトコル (MCP) や Agent2Agent (A2A) などの明確なインターフェイス、共有コンテキスト形式、標準通信プロトコルを定義します。 Agent2Agent スワーム、グラフ、階層調整などのマルチエージェントオーケストレーションパターンを採用します。これらのパターンは、タスク構造と信頼レベルに応じて、エージェントが並列ワークフロー、シーケンシャルワークフロー、またはコンセンサス駆動型ワークフローのいずれかで、相互に機能を検出し、サービスを動的にリクエストするのに役立ちます。

スケーラブルで管理されたコラボレーションを促進するには、アービターエージェントを使用します。この種のエージェントは、既知の機能とフォールバック戦略に基づいてタスクの委任を容易にする中立的な権限です。一元化されたコントローラーではありませんが、アービターエージェントは信頼とコンプライアンスにおいて重要な役割を果たします。これにより、機密性の高いタスクや規制されたタスクは、アイデンティティとポリシーの要件を満たすエージェントにのみルーティングされます。これは、ポリシーバインドワークフローのゲートキーパーとして機能します。分離を強制し、説明可能な委任を有効にします。重要なのは、アービターエージェントはボトルネックではなく、水平方向のpeer-to-peer方式で動作する自己調整エージェントと共存することです。これらのエージェントは、サブタスクを委任し、コンテキストを共有し、依存関係を直接解決します。

このハイブリッドモデルは、決定論的割り当て (アービターエージェント経由) と緊急コラボレーションの両方をサポートします。構造と柔軟性をブレンドします。このアーキテクチャでは、エージェントを次の特殊なロールに分類できます。

- ポリシーエンサー、リソースアロケーター、リスクエバリュエーターなどの決定エージェント
- コンテキストアグリゲータ、パターンレコグナイザー、異常ディテクターなどのナレッジエージェント
- タスクエグゼキュター、品質コントローラー、統合マネージャーなどの実行エージェント

効果的に調整するには、マルチエージェントシステムが状態管理、障害復旧、および競合解決のための堅牢なインタラクションプロトコルをサポートしている必要があります。これにより、エージェントが独立して動作していても、安定性と説明責任が促進されます。

負荷ベースのエージェントのインスタンス化、コンテキスト対応リソースの割り当て、自動機能検出と登録など、スケーリングの明確なルールを確立します。これらの測定値は、需要や複雑さに応じてシステムが動的に成長するのに役立ちます。

分散メッセージングの基板内でready-to-useモジュールとなるようにエージェントを設計します。たとえば、サイロ化されたサービスではなくA2A または MCP で Amazon EventBridge を使用できます。バージョニング、CI/CD パイプライン、エージェントテンプレートを採用して、内部導入とライフサイクルの進化を加速しながら、システムの安定性をサポートします。コードの再利用と標準化を奨励して、統合の摩擦を軽減し、回復力のあるエコシステムを促進します。

コラボレーションは力の乗数です。マルチエージェント環境全体でスケール、特殊化、耐障害性を解放します。この動的なコラボレーションをサポートするために、組織はエージェントの調整のために軽量なコントロールプレーンを設計する必要があります。このコントロールプレーンには以下が含まれます。

- 各エージェントができることを定義し、ピア検出用のバージョン管理されたメタデータをサポートする機能レジストリ
- アービターエージェントまたはスーパーバイザーエージェントを使用して、コンテキスト、可用性、ポリシーに基づいてタスクをルーティングするタスクアービトレーションロジック
- リアルタイムの決定コンテキストと安全な引き渡しを可能にするライフサイクルと状態の追跡

コントロールプレーンは、権限を一元化したり操作を遅くしたりすることなく、マルチエージェントシステムが拡張可能、ポリシー整合性、耐障害性を維持できるようにします。

ただし、マルチエージェント環境には運用上の課題もあります。エージェントとのやり取り全体でコンテキストを維持し、共有状態を管理し、アクションを調整すると、複雑さとコストが増大する可能性があります。エージェント間の通信中にトークンを消費する LLMs を使用すると、コストが増加する可能性があります。これらのコストは、大規模なインテリジェントな自律性がもたらす複合的なビジネス上の利点と照らし合わせて検討する必要があります。

これらの課題に対処するには、次のような重要な懸念事項を抽象化するエージェントプラットフォームを検討してください。

- 標準化された通信プロトコルとセマンティック形式
- 組み込みオーケストレーションロジックと動的ルーティング
- エージェント間の共有コンテキストとメモリ管理
- フォールバック処理と障害時の正常な低下

マルチエージェント戦略を採用するチームにとって、最善のアプローチは小規模から始めて規模に合わせて設計することです。実際の問題を解決する、ターゲットを絞った単一エージェントソリューションから始めます。次に、これらのエージェントを協調システムに段階的に構成し、それぞれが共通の目標とシステム全体のコンテキストに基づいて検出、調整、委任できます。

重要なのは、堅牢なエラー処理と正常な機能低下が主要な設計原則であることです。マルチエージェントシステムは、エージェントが利用できない場合や失敗した場合に、部分的なワークフローを継続したり、バックアップロジックを開始したりできる必要があります。これにより、剛直な結合なしで信頼性が向上します。

AWS のサービス は、このアーキテクチャを大規模にサポートするための堅牢な機能を提供します。Amazon EventBridge と EventBridge Pipes は、マルチエージェントメッセージング用の構造化されたイベント駆動型バックボーンを提供します。モジュール動作を管理するために、 はエージェントインスタンス間で安全で動的な設定の切り替えAWS AppConfigを有効にします。共有コンテキストとメモリ管理をサポートするには、Amazon DynamoDB を使用して、テナント対応の軽量な状態永続化とエージェント間の高速コンテキスト取得を実現します。Amazon Simple Storage Service (Amazon S3) を使用して、構造化されたプロンプト履歴、共有アーティファクト、またはエージェント生成出力を保存できます。ステートフルな調整を必要とするより複雑なワークフローの場合、AWS Step Functionsはチェックポイントとエラー復旧ロジックを使用して長時間実行されるプロセスをオーケストレーションできます。これらのサービスを組み合わせることで、企業の需要に合わせてスケールする構成可能で回復力があり、意味的に接続されたマルチエージェントシステムを作成できます。

マルチエージェントシステムのビジネス価値

多くの組織は単一エージェントソリューションで AI ジャーニーを開始しますが、エージェント AI の可能性はスケーラブルなマルチエージェントシステムを通じて最大限に引き出されます。これらのシステムは、複雑で分散した問題を解決し、ビジネスニーズに応じて進化する堅牢で柔軟な AI エコシステムを作成する上で重要です。

マルチエージェントシステムの主なビジネス上の利点は次のとおりです。

- スケーラビリティ – タスクとワークロードを特殊なエージェントに分散して、容量とパフォーマンスを向上させることができます。
- 柔軟性 – エージェントは最小限の中断で追加、置換、または変更できるため、動的な環境で俊敏性を実現できます。
- 耐障害性 – 冗長なロールとインテリジェントなフェイルオーバーにより、個々のエージェントが失敗してもシステムの安定性は維持されます。

- 専門分野 – 専用のエージェントは、より高い効率と精度でタスクを実行します。
- コスト効率 – 再利用可能なエージェントコンポーネントは、開発を加速し、新機能のデプロイコストを削減します。

マルチエージェントシステムは、より多くの事前計画を必要としますが、長期的な俊敏性、スピード、イノベーション能力を提供します。柔軟なエージェントコラボレーションアーキテクチャに投資する企業は、新しい AI 機能を迅速にデプロイし、需要の変化に適応し、ますますエージェント主導の競争環境に導くことができます。

フォーカスエリア 3: マルチテナンシーと制御のためのアーキテクト

実行するジョブ:「管理、説明責任、可視性を失うことなく、複数の顧客にエージェントの使用状況をスケーリングするのに役立つ」

初期のプロトタイプは、単独で価値を証明するのに適していますが、ほとんどの企業は複数の顧客、部門、またはワークフローを同時にサポートする必要があります。つまり、各エージェントは明確に定義されたポリシー、データ、アイデンティティの境界内で動作する必要があります。マルチテナンシーがないと、オペレーションは脆弱でコストがかかり、ガバナンスはパッチワークになります。

方針

Software as a Service (SaaS) アーキテクチャの原則に従います。例えば、テナントの分離、ポリシーの適用、リソース制御の設計などです。テナント対応のメモリ、設定、アイデンティティを使用して、エージェントとオーケストレーションプラットフォームを設計します。境界を適用するには、タグ付け、ロールベースのアクセスコントロール (RBAC)、ID とアクセスの管理スコープを使用します。

エージェントテレメトリがテナントコンテキスト別に集約される、統一されたオブザーバビリティレイヤーを採用します。一元化されたポリシーエンジンと設定ベースの機能を実装して、動的な動作ルールを適用します。

サービスとしてのエージェントのデプロイを構築します。内部チームまたは顧客がエージェント機能をスケーラブルで管理された APIsとして消費できるようにします。 は、これらのパターンの強力な基盤 AWS を提供します。Amazon Cognito を使用して、ユーザーとテナントの ID を管理し、AWS Organizationsサービスコントロールポリシー (SCPs) を使用してクロスアカウントガバナンスを行

い、 AWS Resource Access Manager (AWS RAM) を使用して機能を安全に共有できます。さらに、AWS AppConfigはテナントまたは環境ごとにエージェントの動作を動的に管理できます。これらのサービスは、共有インフラストラクチャをサポートしながら境界とポリシーを適用するのに役立ちます。

この静的デプロイから動的プロビジョニングへの移行により、エージェント AI はエンタープライズ全体のプラットフォームになります。

マルチテナントエージェントプラットフォームのビジネス価値

マルチテナンシーは、単なるアーキテクチャ上の利便性ではなく、ビジネスアクセラレーターです。インテリジェントなエージェントが部門やチーム間で拡散するにつれて、組織はインフラストラクチャを複製したりガバナンスを断片化したりすることなく、成長をサポートする必要があります。

マルチテナントシステムの主なビジネス上の利点は次のとおりです。

- スケーラビリティ – マルチテナントエージェントプラットフォームを使用すると、社内チーム、ビジネスユニット、またはクライアントは、カスタム環境を必要とせずに AI 機能を迅速にオンボードできます。
- コスト効率 – 共有インフラストラクチャは、冗長なデプロイを最小限に抑え、運用コストを統合し、環境全体のメンテナンスを簡素化します。
- ガバナンスとリスク削減 – 一元化されたポリシーコントロール、アイデンティティモデル、オブザーバビリティは、エージェントがすべてのテナントでより安全かつコンプライアンスに従って運用するのに役立ちます。
- サービスの再利用可能性 – 再利用を促進し、重複を減らすために、テナント対応エージェントをエンリッチメント、コンプライアンス、要約などの内部サービスとして提供できます。

マルチテナントシステムのユースケースの例は次のとおりです。

- 子会社全体にデプロイされるコンプライアンスエージェントは、テナント固有の設定を通じて、そのロジックをローカル規制に適応させます。これにより、リージョンごとに個別のエージェントを構築する必要がなくなります。
- 内部ワークフロー自動化エージェントは、異なるデータ境界とアクセス許可を持つ複数の部門を提供します。分離を維持しながら、タスクのフルフィルメントを加速します。

エージェントをmulti-tenant-awareサービスとして設計することで、組織はサイロ化された AI イニシアチブのオーバーヘッドを回避できます。代わりに、統合インテリジェンスプラットフォームを促進

します。このアーキテクチャにより、スケーラブルなロールアウト、運用の一貫性、ROI の向上が可能になります。また、AI の導入を企業全体に拡大することが容易になります。

フォーカスエリア 4: ID、ガードレール、オブザーバビリティを通じて信頼を構築する

実行するジョブ:「特に誰も見ていないときに、エージェントが安全かつ予測可能な行動を取るという自信をください。」

自律型エージェントは、従来のコントロールモデルに挑戦します。適切な管理が行われていない場合、個別に推論して行動する能力はリスクをもたらします。明確な所有権、監査可能性、またはポリシーの制約がないと、意図した動作から逸脱する可能性があります。組織の信頼を構築するには、技術的な信頼性以上のものが必要です。これには説明可能性、説明責任、一貫性が必要です。

方針

信頼できる自律性のバックボーンとして、アイデンティティファーストの管理システムを構築します。各エージェントは、検証可能な ID、スコープ付きアクセス許可、追跡可能な実行履歴で動作する必要があります。エージェントは、テナントバインディング、コンテキストアクセス継承、ガードレールとポリシーエンジンによるランタイム適用を含むゼロトラストフレームワークに埋め込む必要があります。これにより、組織のルールとリスク体制に基づいて、エージェントのアクションを監査、取り消し、または制限できます。

インテリジェントなガードレールを使用して、実行時に信頼の適用を埋め込みます。これには、動作パターンまたはワークロード条件に基づくレート制御とスロットリング、自動スケーリングとともに適用されるリソース境界、リスクを評価するための決定スコアリングが含まれます。しきい値を超えたときにヒューhuman-in-the-loopワークフローをエンゲージするトリガーを構築します。

また、すべてのエージェントは透明で説明可能である必要があります。ログ記録、トレース、推論の概要を通じて構造化テレメトリを埋め込み、決定ロジックを公開します。決定証跡と影響追跡をサポートします。これにより、エージェントアクションを主要なメトリクスまたは結果に接続できます。予想される動作やポリシーからの逸脱を監視するドリフト検出メカニズムを実装します。

エージェントの動作とシステムパターンを継続的に監視する反射エージェントを紹介します。異常や不整合にリアルタイムでフラグを付ける必要があります。これらのエージェントは、機能の再検証、適応、または廃止を開始できるガバナンスフィードバックループに貢献します。

エージェントポリシーのレビュー、機能変更の承認、インシデント対応プロトコルの監督を行うガバナンスボードを設置します。信頼を獲得、測定し、継続的に強化する必要があります。

AWS は、この信頼フレームワークを実装するための強力な基盤を提供します。

- AWS Identity and Access Management (IAM) は、ロールベースの実行とアクセス許可の境界を適用します。
- Amazon CloudWatch と は、完全な可視性とトレーサビリティAWS X-Rayをサポートします。
- Amazon GuardDuty を使用して、セキュリティの異常やポリシードリフトAWS Configを検出します。

これらのサービスを組み合わせることで、アイデンティティの適用、ランタイムの安全性、信頼ベースのガバナンスを大規模に実現できます。これらは、自律システムを強化し、信頼性を高めるのに役立ちます。

信頼できる自律性のビジネス価値

エージェントの自律性が高まるにつれて、信頼は企業の採用、ガバナンス、運用パフォーマンスにとって重要な推進要因になります。ID、オブザーバビリティ、ガードレールの基盤を確立することで、組織はガバナンスやコントロールを犠牲にすることなく、エージェント AI を機密ドメインにスケールできます。

主要なビジネス推進要因は次のとおりです。

- ガバナンスの保証 – 強力なアイデンティティモデル、監査証跡、アクセス許可の境界により、コンプライアンスリスクを軽減し、規制の調整をサポートします。
- 運用継続性 – ランタイムガードレールと異常検出は、意図しない動作を防ぎ、エッジケース障害からの自己回復をサポートします。
- ステークホルダーの信頼 — 意思決定の説明可能性とテレメトリは、内部ステークホルダー、リスクマネージャー、外部監査人との信頼を構築します。
- インシデントレジリエンス – 組み込みオブザーバビリティは、問題が発生した場合の根本原因分析と応答時間を短縮します。

ユースケースの例を以下に示します。

- 金融サービスでは、不正検出エージェントは推論を公開し、追跡可能な ID を使用してすべてのアクションを記録し、厳密にスコープされた IAM ロールの下で運用する必要があります。
- ヘルスケアでは、自律型トリアージエージェントはランタイム安全チェックを実施し、しきい値が満たされたら人間によるレビューにエスカレーションし、臨床監視のための完全な口グを提供する必要があります。

信頼メカニズムをエージェントのライフサイクルに組み込むことで、組織はシステムが説明責任を持って自律的に動作することを許可できます。この基盤はリスクを軽減し、エージェントが透明性と完全性をもってビジネスに代わって行動できるようにします。

最終的に、信頼された自律性は、ユーザーとリーダーの両方に、コアオペレーション全体でインテリジェントエージェントをスケールする自信を与えることで、導入を加速させます。

フォーカスエリア 5: ライフサイクルを管理する

完了すべきジョブ:「カオスやヒロインなしで、チームが時間の経過とともにエージェントを改善できることを確認してください。」

コードのみで形成された従来のアプリケーションとは異なり、エージェントの動作はプロンプト、メモリ、ツール、トレーニングコンテキストによっても形成されます。これらの要因は時間の経過とともに変動します。ドリフトは信頼性を低下させ、コストを増大させ、デバッグをほぼ不可能にします。ライフサイクルコントロールがないと、エージェントは価値の提供を停止し、リスクの蓄積を開始します。

方針

練習としてエージェントの DevOps (AgentOps) を確立します。エージェントに合わせた CI/CD パイプラインを統合します。これらのパイプラインを使用して、プロンプト出力のテスト、ツール統合の検証、コストパフォーマンスの動作のプロファイリングを行います。プロンプト、ポリシー、モデルインタラクションのバージョン履歴を維持します。

オブザーバビリティデータからのフィードバックループを使用して、再トレーニング、プロンプト調整、またはエージェントの廃止を開始します。改善登録などのシステム全体のリフレクションメカニズムを組み込み、学習を制度化します。

決定精度、レイテンシー、コスト、信頼性を示すパフォーマンステレメトリダッシュボードを構築します。インフラストラクチャを使用した AWS ライフサイクル管理を合理化して高速化するために、チームはエージェントツールキットを使用できます。例として、Strands Agents SDK があります。これは、プロンプトのバージョニング、ツール登録、および AWS CodePipeline、 AWS のサービス、 AWS Cloud Development Kit (AWS CDK)などの との CI/CD 統合のための構造化ツールを提供しますAWS Lambda。さらに、Amazon S3 と Amazon Elastic File System (Amazon EFS) を使用して、エージェントアーティファクトとトレーニングデータを保存します。を使用してAWS Step Functions、複雑な再トレーニングまたは検証ワークフローを自動化します。エージェントが LLM オーケストレーション以外のカスタムモデル調整または微調整ワークフローを必要とする場合

は、Amazon SageMaker AI を使用できます。ライフサイクルの規律は、エージェントを実験から永続的で進化し続けるアセットに変換します。

時間の経過とともに、このライフサイクルシステムはイノベーションのバックボーンを形成します。これにより、機能の再構築、再トレーニング、再デプロイを俊敏に行うことができます。これにより、エージェントレイヤーは生きたシステムに変換され、フィードバックと機会の両方に応じて進化できます。

ライフサイクル管理のビジネス価値

効果的なライフサイクル管理は、エージェントのパフォーマンスとコスト効率の主要な推進要因です。これにより、インテリジェントエージェントは進化するにつれて、正確で信頼性が高く、価値に沿った結果を提供し続けることができます。エージェントはデフォルトでは価値を持ちません。ビジネス要件、ワークフロー、データ環境の変化に合わせて進化する必要があります。統制のとれた AgentOps チームは、エージェントが正確で効率的で、時間の経過とともに企業の目標に合わせるのに役立ちます。

主要なビジネス推進要因は次のとおりです。

- パフォーマンスの一貫性 – 継続的なテスト、プロンプトの検証、再トレーニングは、変化する条件やデータセットにわたってエージェントが決定品質を維持するのに役立ちます。
- コスト最適化 – テレメトリ駆動型プロファイリングは、非効率的なツール、高トークンプロンプト、または不要な実行を識別します。その後、 を調整して運用コストを削減できます。
- イテレーションの高速化 – CI/CD によるライフサイクル自動化は開発サイクルを加速し、チームが自信を持ってエージェントを実験、デプロイ、改善するのに役立ちます。
- リスク軽減 — 迅速なバージョニング、ロールバックサポート、構造化された評価メカニズムは、リグレッションを防ぎ、安全で信頼性の高い変更管理をサポートします。

ユースケースの例には次の内容が含まれます。

- カスタマーサポートエージェントは、レイテンシー、モデルコスト、ユーザーフィードバックについてモニタリングされます。オブザーバビリティはコストの急増を明らかにし、埋め込みプロンプトとフォールバックモデルロジックの再調整を促します。
- 契約要約エージェントは、リーガルチームからのフィードバックに基づいて更新されます。バージョン管理されたプロンプトは、本番リリース前にサンドボックス環境でテストされ、安全性と品質をサポートします。

構造化されたライフサイクル管理により、組織は事後対応型メンテナンスからプロアクティブで継続的な改善に移行します。エージェントは、ビジネス目標に対して測定、改良、再検証される適応型デジタルアセットになります。この手法は、エージェントエコシステムを高性能でコスト意識が高く、回復力のあるシステムに変換し、変化に遅れることなく永続的な価値を提供します。

重点領域 6: エージェントモデルをビジネスモデルと整合させる

実行するジョブ:「継続的な投資を正当化できるように、影響を表示します」。

技術的に能力のあるエージェントでも、ビジネス成果に結びついていない場合は責任が発生します。エージェントは、効率、収益化、または戦略的差別化を提供する必要があります。しかし、ほとんどの企業は、エージェントが価格、パッケージング、または使用モデルにどのように適合するかを定義するのに苦労しています。ビジネス価値に明確な整合性がなければ、スケーリングを正当化したり、投資を維持したりすることさえ困難です。

方針

製品管理プラクティスを採用します。エージェントを収益化可能なサービスとして測定可能な ROI で扱います。決定、セッション、または結果に基づいて料金戦略を定義します。次に、エージェント機能を顧客セグメントまたは内部ビジネスユニットに合わせた階層型サービスにパッケージ化します。

持続可能性を促進するには、組織はエージェントのデプロイを通じて直接的な価値と成長の両方の乗数をキャプチャする必要があります。次の ROI メトリクスを使用して即時値を測定することを検討してください。

- 決定あたりのコスト – エージェントの処理コストを人間の同等物とベンチマークします。
- 時間圧縮 – 販売や承認の迅速化など、加速サイクルの価値を定量化します。
- エラー削減 – 精度、一貫性、コンプライアンスの向上によるコスト削減を測定します。

これらの差し迫った利益を超えて、エージェントは次の長期的な成長機会を解き放つことができます。

- 機能スタッキング – エージェントサービスを組み合わせて、ドメイン固有の垂直的ソリューションを作成します。
- ネットワーク効果 – 調整複合が機能するマルチエージェントエコシステムを通じて価値を高めます。

- 市場拡張 – 外部で使用可能なエージェント対応サービスを通じて新しい収益ストリームを生成します。

ビジネスメトリクス (コスト削減、コンバージョンリフト、time-to-resolution) からフィードバックループを作成し、エージェントの継続的な進化を促進します。使用状況テレメトリとユーザー満足度スコアを分析して、価値の整合性とロードマップの優先順位を絞り込みます。エージェント機能をビジネスモデルに直接リンクすることで、組織は技術的成果だけでなく、持続可能で複合的な価値を捉えることができます。

以下は、堅牢な追跡および収益化フレームワークを提供することで、この調整 AWS のサービス をサポートします。

- AWS Cost Explorer と Amazon CloudWatch は、エージェントあたりのコストと運用効率に関するインサイトを提供します。
- Amazon API Gateway は、エージェントエンドポイントの従量制アクセス、レート制限、階層化料金を有効にします。
- AWS Marketplace は、エージェントとエージェントソリューションを商用製品として公開するためのチャネルを提供します。

これらのサービスは、エージェントの機能を、企業の成長と収益化戦略に沿ったスケーラブルで価値主導型のデジタルサービスに変換するのに役立ちます。

エージェント AI のソフトウェア配信の進化

最新のソフトウェア配信は、出荷するシステムを制御するという単純な前提によって形成されています。要件の定義、ロジックの記述、期待される成果に対するテスト、予測可能なサービスのデプロイを行います。アジャイルアプローチや DevOps アプローチでも、各スプリントが決定論的、検証可能、主に人間による監視の範囲内で何かを提供するという原則に依拠しています。

エージェント AI はその基盤をアップグレードします。エージェントシステムは、スクリプトに従うのではなく、解釈、理由、適応します。動作は、記述するコード、操作するコンテキスト、入力、アクセスできるツール、割り当てられた目標によって異なります。つまり、注文に従わず、結果を追求します。

これにより、配信は制御についてより少なくなり、調整についてより多くなります。指示を提供するのではなく、その動作を具体化する必要があります。つまり、従来のソフトウェア開発ライフサイクル (SDLC) はロジックベースのヒューマンコントロールシステム向けに設計されているため、適合しなくなりました。

このセクションは、以下のトピックで構成されます。

- エージェント AI のインテントゾーン
- エージェント AI の配信ライフサイクルの進化
- エージェント AI のためのチームの準備

エージェント AI のインテントゾーン

定義、構築、テスト、リリースなどのハードステージの代わりに、自律性、不確実性、出現を受け入れるモデルが必要です。代わりに、インテントゾーンを使用します。インテントゾーンは、エージェントが制約内で自律的に操作できる境界領域を定義します。目標は、すべてのタスクのマイクロ管理から、エージェントが安全に行動、学習、コラボレーションできる環境の設計に移行することです。何 (望ましい結果) 、理由 (インテント) 、ガードレール (制約、ポリシー、信頼境界) を指定します。これらの境界とこの情報を考慮すると、エージェントはその方法を特定します。

アセンブリラインの代わりに、環境を空域と考えてください。誰が入力できるのか、何ができるのか、どこへ向かうのかを自分で制御できます。ただし、内部に入ると、必要に応じて自由に移動できます。このようにエージェントシステムはカオスなしでスケールします。

これは単なる哲学的なシフトではなく、実用的なシフトです。エージェントベースのシステムの非決定的な出力は、ユニットテストでは完全にテストできません。静的バイナリのようにバージョニング

することはできません。エージェントは時間の経過とともに変化し、新しいデータに適応し、予測不可能な方法で他のシステムとやり取りします。従来のモデルを使用して配信しようとすると、脆弱でスケーラブルでないアーキテクチャになります。最悪の場合、実際には管理できないシステムに対する誤った信頼につながります。

チームがインテントベースのデリバリーを採用すると、次の 2 つの利点があります。

- 最も重要な場所を制御する – 出力ではなく境界を定義します。
- 委任によるスケーラビリティ – エージェントが人間がハードコードできない複雑さに対応できるようにします。

これは、分離されたプロトタイプから、価値を反復的かつ確実に提供できる実稼働グレードのエージェントシステムに移行する方法です。

エージェント AI の配信ライフサイクルの進化

インテリジェントでアダプティブな動作をサポートするには、SDLC を決定論的制御からアダプティブインテントに再構成する必要があります。エージェント AI の従来の SDLC を進化させるために必要な変更を次に示します。

- 計画はインテント設計になります。チームは、目標、制約、予想されるエージェントの行動を定義します。ポリシーと成功基準は、ロジックではなく整合性の観点からフレーム化されます。
- アーキテクチャが足場になります。チームは、すべての決定パスをスクリプト化するのではなく、ロール、インターフェイス、ガードレール、フォールバックメカニズム、オブザーバビリティの定義に焦点を当てます。
- テストは動作評価になります。チームは、特定の出力をアサートするのではなく、エージェントが許容範囲内に留まるかどうかを検証し、さまざまな入力でインテントを達成します。
- デプロイは継続的なオーケストレーションになります。エージェントシステムは、リアルタイム調整を可能にするランタイムコントロール、ライブモニタリング、フィードバックチャネルを使用してデプロイされます。
- 反復はフィードバックと適応になります。従来のコード変更パッチサイクルの代わりに、チームはエージェントの進化、成功場所、ドリフトを観察します。必要に応じて、チームは更新された制約、再トレーニング、コントロールメカニズムの追加または変更に介入します。

反復、実験、迅速なフィードバックに焦点を当てた既存のプラクティスは、その途中にあります。エージェントシステムへの移行は、アジャイルの原則を拒否するものではありません。実際、これ

は自然に進化したものです。アジャイル思考は、厳格な計画よりも適応性、フィードバック、作業ソリューションを重視します。これは、リアルタイムでコンテキストを学習、適応、応答するエージェントシステムの性質と完全に一致します。すでに短いサイクルを実行し、前提を迅速に検証し、継続的なデリバリーを通じて不確実性を管理している場合は、この移行を主導する準備が整います。

ただし、主な違いがあります。従来のアジャイルアプローチでは、配信されるモノが決定論的であることを前提としています。構築されると、モノは一貫して予測どおりに動作し、同じ入力に対して繰り返し可能な結果が得られることを前提としています。この再現性は、デバッグ、テスト、反復を自信を持って行うのに役立ちます。エージェントシステムは、そのモデルを破壊します。確率的であり、コンテキストに敏感で、独立して進化できます。つまり、ストーリーの完了に基づく速度追跡、厳格な承認基準、決定論的なスプリント計画など、一部のアジャイルプラクティスはあまり役に立たなくなります。

従来の SDLC の以下の側面がエージェント AI に適用されます。

- 反復開発と配信
- プライマリシグナルとしてのお客様のフィードバック
- 部門間のコラボレーション
- 継続的な統合とデプロイ

エージェント AI では、従来の SDLC の以下の側面を進化させる必要があります。

- インテントに合わせて再定義します。 エージェントの動作が、定義された制約内で意図した目標を満たしているかどうかに焦点を当てます。
- 許容基準から動作ガードレールへの移行。
- 継続的な学習と信頼をサポートするオブザーバビリティ、説明可能性、フィードバックメカニズムを含むランタイムの準備状況を含めるように、完了の定義を拡張します。
- 事前計画よりもリアルタイムのフィードバックループと動作追跡を優先する

良いニュースは、SDLC プレイブックを捨てる必要がないことです。コードの管理から行動の形成まで進化させるだけで済みます。エージェントシステムでは、成功とはソフトウェアが実行されるかどうかだけでなく、その動作に関するものです。

エージェント AI のためのチームの準備

ソフトウェアエンジニアリングは廃止されません。進化しています。ジョブは、関数の記述から、インテリジェントな動作のためのフレームワークと制御メカニズムの形成に移行します。エージェント AI の世界では、構築はもはやハードな部分ではなく、出現の管理です。ほとんどのエンジニアリングチームにとって、進化は技術的な飛躍ではなく考え方の変化のように感じられます。「システムは何をしますか？」と尋ねる代わりに 質問は「何を追求する権限を与えたか」になり、それが進行中かどうかはどのようにわかりますか？」になります。

エンジニアリングチームの場合、エージェント AI への移行には以下の変更が必要です。

- 文化的シフト – チームは、完全には制御できないシステムの不確実性と自律性に慣れる必要があります。
- 新しいロール – インテントデザイナー、動作テスター、オブザーバビリティエンジニアがデリバリーの中核となります。
- 共有言語 – チームは、これまで仕様やテストケースが必要だったのと同様に、目標、ガードレール、成功シグナルを明確に共有する必要があります。

生成 AI が成熟するにつれて、より多くのエージェントシステムが顧客、製品、オペレーションとやり取りするようになります。成功した組織は、最適なモデルを持つ組織ではありません。これは、エージェントを信頼、制御、速度で実際のワークフローに統合できるものです。つまり、デリバリーモデルとエンジニアリングチームは一緒に進化する必要があります。インテントゾーンを使用すると、抽象化できます。説明責任を放棄することなく、自律性を運用するのに役立ちます。また、チーム間で共有フレームワークを提供し、ハードコードできないシステムを管理するのに役立ちます。

エージェント AI のためのチームの準備の詳細については、このガイドの「大規模なエージェント AI のためのビジネスの準備」セクションを参照してください。

エージェント AI の大規模なビジネスの準備

このガイドで説明されている重点領域が収束すると、エージェント AI は分離された関数から、機能プラットフォームとして理解できる統合インテリジェンスレイヤーに移行します。このプラットフォームはタスクを実行するだけではありません。ドメイン間で進化、適応、調整を行います。エージェントは、イノベーションを加速し、認知負荷を軽減し、企業全体で測定可能な成果を促進するモジュール型、再利用可能な、発見可能なサービスになります。このプラットフォームビューは、運用モデル全体に埋め込まれたスケーラブルなインテリジェンスのステージを設定します。

エージェント AI の運用には、インテリジェントエージェントのデプロイ以上のものが必要です。これには、企業がチームを編成し、プロセスを設計し、テクノロジーを管理する方法の根本的な変革が必要です。クラウドまたは DevOps が運用モデルを再定義したのと同様に、エージェント AI は決定の自動化、継続的な学習、自律的な調整の新しい時代を導入します。成功は、この新しい運用哲学に関するシステム、人材、プロセスを調整することにかかっています。

このセクションは、以下のトピックで構成されます。

- チームおよび所有権モデルの連携
- 変更と組織の準備状況の管理
- 相互運用性とコラボレーションのためのアーキテクチャ
- エージェントファブリックへのガバナンスの構築
- 決定優先の運用マインドセットを採用する
- 目的とインテントを使用したスケーリング

チームおよび所有権モデルの連携

成熟に向けた最初のステップは、部門間の連携です。企業は、分散システムアーキテクト、ソフトウェアエンジニア、製品所有者、コンプライアンスリード、プラットフォームアーキテクトなどの AI/ML 実務者とドメインスペシャリストを含む AgentOps チームを確立する必要があります。これらのチームは、設計とデプロイから再トレーニングとモニタリングまで、エージェントのライフサイクル全体を共同で所有します。

エージェントのプロビジョニングとリリースは、Infrastructure as Code や自動デプロイ AWS CodePipeline に AWS Cloud Development Kit (AWS CDK) や を使用するなど、クラウドネイティブのプラクティスに従う必要があります。この構造は、説明責任の共有を促進し、反復を加速しま

す。DevOps が開発と運用を統合するのと同様に、AgentOps はインテリジェンスをガバナンスと実行に接続します。

効果的にするために、これらのチームには共有言語も必要です。ビジネス関係者は、エージェントとは何か、どのように運用するか、どのような成果をもたらすかを理解する必要があります。トレーニングと内部有効化が不可欠です。エージェントをわかりやすくし、このメンタルモデルを日常的な会話に組み込むことで、組織はより広範な参加とより整合性のあるイノベーションを引き出します。

を使用してエージェントの開発と統合を加速するために AWS のサービス、チームは Strands Agents SDK などのフレームワークを採用できます。これは、エージェントの足場、設定、パッケージングのための CLI ベースのツールを提供します。Strands Agents は、Amazon Bedrock、、Amazon EventBridgeAWS Lambda、、 AWS CDKなどのインフラストラクチャとシームレスに AWS 連携するように設計されています AWS CodePipeline。これにより、本番稼働用の標準を維持しながら、迅速なプロトタイプ作成とデプロイが可能になります。

しかし、構造とツールだけでは不十分です。エージェント AI のスケーリングには、組織全体で導入が根付くように、文化、教育、リーダーシップに関する慎重な準備が必要です。

変更と組織の準備状況の管理

エージェント AI を正常にスケーリングするには、インフラストラクチャやインテリジェントエージェントをデプロイするだけではありません。組織変革には構造化されたアプローチが必要です。これには、文化的準備状況、スキル開発、メトリクス主導のフィードバックループ、経営陣の連携が含まれ、導入が意図的かつ持続可能であることを確認します。

文化の進化を促進する

- エージェントを置き換えではなくチームメイトとして配置して、抵抗を減らし、信頼を構築します。
- エージェントの機能と制限について透過的にコミュニケーションをとり、現実的な期待を設定します。
- エージェントが決定をより高い機関にエスカレーションしたり、プロセスの一部を共同作業者に委任したりするタイミングについて、明確な引き渡しプロトコルを確立します。

スキル開発フレームワークを確立する

- エンジニア、製品マネージャー、ドメインリード、コンプライアンス責任者に合わせたロールベースのトレーニングを提供します。

- センターオブエクセレンスを作成して、ベストプラクティス、ツールパターン、再利用可能なアセットを共有します。
- 指導プログラムを通じて AI スペシャリストをドメインエキスパートとペアリングし、知識のギャップを埋めます。

メトリクスとフィードバックループを定義する

- 技術的およびビジネス上の KPIs戦略的価値にアンカーして、影響を評価します。価値の例としては、決定レイテンシー、解決精度、コスト削減などがあります。
- ユーザーのフィードバックを体系的かつ継続的にキャプチャして、摩擦ポイントと導入の課題を明らかにします。
- 定期的な遡及分析を実施して、エージェントのパフォーマンス、使用状況の傾向、改善の機会を評価します。

リーダーシップを上から調整する

- エージェントイニシアチブを戦略的成果と ROI にリンクして、エグゼクティブスポンサーシップを取得します。
- 技術リーダーシップとビジネスリーダーシップの両方を含む部門横断的なガバナンス委員会を編成します。
- コミュニケーション戦略を調整して、すべての組織レベルで明確さとエンゲージメントを実現します。

変更管理に対するこの体系的なアプローチにより、テクノロジーの実装が組織の成熟度と一致するようになります。これにより、信頼、導入、長期的なビジネス価値の基盤が構築されます。

相互運用性とコラボレーションのためのアーキテクチャ

分離されたエージェントのデプロイは、ローカルの勝ちを提供します。ただし、エージェントが互いに動的に検出、呼び出し、コラボレーションできると、エンタープライズ価値が生まれます。つまり、エージェントの登録、認証、機能交換の基準を定義します。アーキテクチャ上、これはモノリスからマイクロサービスへのシフトを反映しています。マイクロサービスは、複雑な問題を一緒に解決する、組み合わせ可能で再利用可能な疎結合ユニットです。

A2A や MCP などの新しいプロトコルは基盤となります。は、エージェント、ツール、メモリシステム間のセマンティック相互運用性を有効にします。A2A はピアレベルのインタラクションをサポー

トしているため、エージェントはタスクの所有権の交渉、コンテキストの共有、ワークフローの調整を行うことができます。MCP は、エージェントとその環境間でコンテキストデータを交換するための共有スキーマを提供することで、これを補完します。関数の呼び出し方法、APIsへのアクセス方法、状態の維持方法を標準化します。これらのプロトコルを組み合わせることで、エージェントエコシステム全体の拡張性、一貫性、長期的な保守性が向上します。

ガバナンスは依然として重要です。アービターエージェントなどのコントロールレイヤーは、一元化されたボトルネックを発生させることなく、ポリシー対応の委任を有効にします。これらのエージェントは信頼ブローカーとして機能します。他のエージェントが自己組織化できるようにしながら、境界を適用します。エージェントコラボレーションは、組織が俊敏性と信頼の両方でエージェント AI エコシステムをスケールするのに役立ちます。

エージェントファブリックへのガバナンスの構築

自律性が高いほど、リスクが高くなります。ガバナンスは、初日からエージェントアーキテクチャに埋め込む必要があります。これには、エージェントが実行できることの範囲を限定するポリシーの境界の定義、エージェントが誰に代わって行動するかを決定するアイデンティティモデルの適用、説明可能性とトレーサビリティの実装が含まれます。オブザーバビリティシステムは、Amazon CloudWatch や などのサービスを使用してエージェントの動作に関するテレメトリをキャプチャする必要があります。これによりAWS X-Ray、エージェントワークフロー全体で一元的なログ記録と分散トレースが可能になります。リフレクティブエージェントは、これらのテレメトリフィードに基づいてパフォーマンスを継続的に監査および評価できます。

エージェントエコシステムが成熟するにつれて、ガバナンスも進化する必要があります。エージェントの能力と自律性が高まるにつれて、監視メカニズムはより適応的になる必要があります。ポリシーの更新、機能ゲート、ランタイム動作の制約は、動的で大規模に実施可能である必要があります。信頼はボルトオン機能ではありません。アーキテクチャ、動作、プロセスを通じて継続的に強化されています。AWS Identity and Access Management (IAM) と AWS AppConfigは、安全な ID、ランタイムアクセス許可の境界、および環境固有の動作の切り替えをエージェント間で強制する上で重要な役割を果たします。

決定優先の運用マインドセットを採用する

従来の自動化では、事前定義されたスクリプトやワークフローをより迅速かつ確実に実行するプロセス効率に焦点を当てています。これとは対照的に、エージェント AI は決定優先の自動化を導入します。エージェントはコンテキストを評価し、オプションを比較検討し、リアルタイムで動作を適応させます。この実行優先の考え方から決定優先の考え方への移行には、成功のメトリクスと結果に関す

る新しい考え方が必要です。タスクの完了によってのみ成功を測定するのではなく、エージェント AI の成功は、決定がインテント、ポリシー、進化する条件とどの程度一致しているかによって測定されます。

組織は、タスクの完了やサイクル時間のみを測定するのではなく、決定の質、time-to-action、変化への応答性を評価する必要があります。KPIsには、次のようなメトリクスを含める必要があります。

- 決定品質 – エージェントは特定のユーザーまたはシナリオに対するレスポンスをどの程度パーソナライズしましたか? ビジネス目標とユーザーコンテキストに沿った微妙な意思決定が行われましたか?
- Time-to-action – エージェントが状況を評価して対応したのは、どのくらい迅速かつインテリジェントですか? レイテンシーは、アダプティブで人間らしく感じられるほど低くなりましたか?
- 認知オフロード – エージェントが人間に代わって処理できた手動分析、トリアージ、または日常的な意思決定の量。労力を減らしたのか、それとも単にシフトしたのか。

意思決定優先の考え方を採用する企業は、回復力、適応力、および新しいレベルの複雑さで運用できるようになります。

目的とインテントを使用したスケーリング

エージェント AI のスケーリングを成功させることは、より多くのツールを試すことではありません。これは、エンタープライズインテリジェンスの耐久性のあるレイヤーを構築することです。これには、プラットフォームインフラストラクチャ、運用文化、ガバナンスフレームワーク、戦略的調整への投資が必要です。企業は、意図的なアプローチを採用する必要があります。エージェントは、実験としてではなく、デジタル運用モデルのコアコンポーネントとして扱う必要があります。

AWS Well-Architected フレームワークと連携することで、システムは信頼性、セキュリティ、パフォーマンス効率、コスト最適化に関するエンタープライズ基準を満たすことができます。Strands Agents SDK などのツールは、構造化されたプロンプト、ツール登録、CI/CD の準備を提供することで、このジャーニーを加速できます。これにより、チームは使い慣れた AWS ワークフローを使用して実験からスケーラブルなデリバリーに移行できます。

エージェント AI はツールではなく、インテリジェンスをオペレーションに埋め込む方法の変化です。それに応じて準備する組織は、ますます複雑化する世界で、より多くの自動化、よりスマートな運用、より迅速な適応、永続的な利点を生み出すことができます。

エージェント AI の運用に関する結論

エージェント AI は単なる技術シフトではありません。これは、エンタープライズ向けの新しいオペレーティングシステムの出現を示しています。この変革を受け入れる組織は、自動化のユースケースを絞り込み、運用の基盤にインテリジェンスを構築します。このシフトでは、意思決定方法、システムの適応方法、大規模な成果の実現方法を再設計します。

複雑さの増大、リアルタイムの需要、情報過負荷によって定義される時代に、スクリプト化されたオートメーションの従来のモデルは制限に達しました。成功は、認識、理由、行動、進化するシステムを構築するために、インテリジェンスをワークフローに直接埋め込む能力にかかっています。エージェント AI は、自律性を目的に、意思決定をガバナンスに、適応性を説明責任に合わせることができます。

この移行には、実行優先から決定優先への移行が必要です。エージェントシステムは、単に指示に従うわけではありません。これらは、定義された制約内で目標の解釈、トレードオフの重み付け、結果の追及を行います。このコンテキストでは、タスクの完了だけでなく、成功も測定されます。また、リアルタイムで行われた決定の品質、俊敏性、説明可能性によっても測定されます。組織は、不確実性の下でインテリジェントに運用するエージェントをサポートするために、メトリクス、インセンティブ、システム設計を再検討する必要があります。

エージェント AI の運用は、plug-and-playアップグレードではありません。これは、アーキテクチャと文化の変換です。これには、ライフサイクル管理、信頼の適用、相互運用性、ビジネスモデルとの整合性に関する統制のとれたプラクティスが必要です。また、インテントゾーンの形成、ランタイムガードレールの埋め込み、エージェントの行動と戦略的成果の継続的な調整など、配信モデルの進化も求められます。チームは、エージェントのパフォーマンスと安全性に関する共有言語、共有所有権、共有説明責任を採用する必要があります。

エンタープライズの準備状況は、この新しい環境で誰が対処するかを決定できます。組織は、長期的な価値をスケールして創出する内部有効化、AgentOps 機能、ガバナンスフレームワークに投資する必要があります。成功したユーザーは、よりスマートなシステムを構築できます。また、より適応力があり、回復力があり、インサイト主導型のビジネスを構築することもできます。

このガイドでは、基盤を構築します。戦略を実行に接続し、組織がインテリジェントエージェントのスケーラブルなプラットフォームを構築する準備をします。のエージェント AI に関するより広範なコンテンツシリーズは、補完的なガイダンス AWS を提供します。このシリーズの他のガイドを表示するには、 AWS 「 規範ガイダンス」ウェブサイトの「エージェント AI」を参照してください。このコンテンツシリーズは、規律と意図を持って自律性を運用するためのロードマップを提供します。

開始するには、エージェントが速度、精度、または応答性において測定可能な改善を提供できる影響の大きい決定領域を特定します。次に、計測、ガバナンス、フィードバックループを持つ集中パイロットエージェントをデプロイします。これを使用して、値仮説を検証し、内部勢いを生成し、アプローチで信頼を構築します。学習によるモメンタムコンパウンド。

エージェント AI は送信先ではなく、ビジネスとともに進化する機能レイヤーです。これは、インフラストラクチャとしてのインテリジェンスへの長期的な移行を表します。この領域を主導する組織は、より多くの自動化、より迅速な対応、より適切な適応、エンタープライズ規模で複雑さをナビゲートできる運用モデルの構築を行うことができます。

エージェント AI を運用するためのリソース

AWS のサービス

以下の AWS のサービス および 機能は、 でエージェント AI システムを構築および運用するのに役立ちます AWS クラウド。

- Amazon API Gateway は、エージェントの機能をスケーラブルなものとして公開し、使用量ベースの料金を提供します。
- AWS AppConfig は、テナントまたは環境全体でエージェントのランタイム設定管理と機能の切り替えを提供します。
- Amazon Bedrock は、エージェントが推論、生成、プロンプト実行に使用できる基盤モデルサービスです。
- AWS Cloud Development Kit (AWS CDK) は、エージェントスタックのデプロイと管理に使用できるコードとしてのインフラストラクチャサービスです。
- AWS CloudTrail はイベント履歴を記録し、エージェントのアクティビティ、監査証跡、統合動作を追跡できるようにします。
- Amazon CloudWatch は、エージェントのパフォーマンスとマルチエージェントコラボレーションの動作をモニタリングするためのログ、メトリクス、アラームを管理できます。
- AWS CodePipeline には、エージェントコードをテスト、検証、デプロイするために使用できる CI/CD オートメーションが用意されています。
- Amazon Cognito は、マルチエージェントシステムのユーザーおよびテナント認証を管理するために使用できる ID サービスです。
- AWS Config は、エージェントポリシーと環境設定のコンプライアンスとドリフト検出を提供します。
- AWS Cost Explorer はエージェントレベルの使用状況を追跡し、コストを調整して ROI を最大化できます。
- Amazon DynamoDB は、エージェントメモリ、改善ログ、コンテキスト状態に使用できるストレージサービスです。
- Amazon Elastic File System (Amazon EFS) は、ワークフロー間のエージェントのコラボレーションや中間処理に使用できる共有ファイルシステムです。
- Amazon EventBridge は、タスクをルーティングし、エージェントファブリックで通信をオーケストレーションするために使用できるコアイベントバスです。

- Amazon EventBridge Pipes は、エージェントとサービスを接続するためのイベントの取り込みとルーティングを合理化できます。
- Amazon GuardDuty は、安全なエージェント実行をサポートできる脅威検出と異常モニタリングを提供します。
- AWS Identity and Access Management (IAM) は、エージェントの実行とデータアクセスのためのきめ細かなアクセス許可を定義するのに役立ちます。
- AWS Lambda は、エージェントロジックとスワームドローンを実行できるステートレスコンピューティングサービスです。
- AWS Marketplace は、エージェント機能を商用製品として提供するために使用できる外部ディストリビューションプラットフォームです。
- AWS Organizations は、マルチテナントエージェントインフラストラクチャの管理に役立つクロスアカウントガバナンスおよびポリシー適用サービスです。
- AWS Organizations サービスコントロールポリシーは、アカウントまたは組織単位レベルでアクセス許可を制御するためのガードレールとして機能します。
- Amazon Quick は、データの分析、視覚化の作成、ワークフローの自動化、組織全体の他のユーザーとのコラボレーションに役立つ生成 AI を活用したビジネスインテリジェンス (BI) プラットフォームです。
- AWS Resource Access Manager (AWS RAM) は、アカウントとエージェントサービス間で機能を共有するのに役立ちます。
- Amazon SageMaker AI は、基礎モデル以外のモデルトレーニング、ファインチューニング、推論に使用できるサービスです。
- Amazon Simple Storage Service (Amazon S3) は、プロンプトライブラリ、モデルアーティファクト、およびエージェント生成データ用のオブジェクトストレージを提供します。
- AWS Step Functions は、マルチエージェントフローの調整とパイプラインの再トレーニングに役立つワークフローエンジンです。
- AWS X-Ray には、エージェントの決定フローとサービスの依存関係を追跡するために使用できる分散トレースが用意されています。

その他の AWS リソース

- 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 フレームワーク、プロトコル、ツール 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

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 8 月 12 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する。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの Oracle 用の Amazon Relational Database Service (Amazon RDS) に移行する。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行する。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの EC2 インスタンス上の Oracle に移行する。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例： Microsoft Hyper-Vアプリケーションを に移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを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。

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

A

A2A (Agent-to-Agent)

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

ABAC

「属性ベースのアクセス制御」をご覧ください。

抽象化されたサービス

「マネージドユーザー」をご覧ください。

ACID

「原子性、一貫性、分離性、耐久性 (ACID)」をご覧ください。

アクティブ/アクティブ移行

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

アクティブ/パッシブ移行

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

[エージェント]

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

エージェントオペレーション

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

集計関数

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

AI

「人工知能」をご覧ください。

AIOps

「AI オペレーション」をご覧ください。

匿名化

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

アンチパターン

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

アプリケーション制御

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

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

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリ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 (AZ)

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です。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを整理しています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、 AWS CAF は、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、AWS CAF ウェブサイトと AWS CAF のホワイトペーパー を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

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

B

不正なボット

個人や組織に混乱や損害を与えることを目的としたボット。

BCP

「ビジネス継続性計画 (BCP)」をご覧ください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 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

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

ボット

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

ボットネット

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

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといいます。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、 AWS 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、AWSでのコンテナ化されたマイクロサービスの実行ホワイトペーパーの「ビジネス機能を中心に組織化」セクションを参照してください。

ビジネス継続性計画 (BCP)

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

C

CAF

「AWS クラウド導入フレームワーク」を参照してください

カナリアデプロイ

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

CCoE

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

CDC

「変更データキャプチャ」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

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

CI/CD

「継続的インテグレーションと継続的デリバリー」を参照してください。

分類

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

シチズンデベロッパー

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

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が、 AWS クラウドへの移行時に通常実行する 4 つの段階。

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、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

「構成管理データベース (CMDB)」を参照してください。

コードリポジトリ

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

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

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

設定ドリフト

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

構成管理データベース (CMDB)

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

コンフォーマンスパック

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

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

継続的インテグレーションと継続的デリバリー (CI/CD)

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

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

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

データ最小化

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

データ境界

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

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

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

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

深層学習

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

多層防御

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

委任管理者

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

トラブルシューティング

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

開発環境

「環境」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

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

ディザスタ

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

ディザスタリカバリ (DR)

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

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

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

「開発バリューストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

EDI

「電子データ交換」を参照してください。

エッジコンピューティング

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

電子データ交換 (EDI)

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

暗号化

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

暗号化キー

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

エンディアン

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

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

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

フェイルファスト

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

障害分離境界

では AWS クラウド、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

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

FGAC

「きめ細かなアクセス制御」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

FM

「基盤モデル」を参照してください。

基盤モデル (FM)

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

FM ゲートウェイ

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

G

生成 AI

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

ジオブロッキング

「地理的制限」を参照してください。

地理的制限 (ジオブロッキング)

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

Gitflow ワークフロー

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

ゴールデンイメージ

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

グリーンフィールド戦略

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

ガードレール

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

ガードレール (AI)

エージェントの入力と出力をフィルタリング、検証、制約して、責任ある安全な AI 動作を確保するのに役立つ安全メカニズム。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

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

高可用性 (HA)

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

ヒストリアンのモダナイゼーション

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

ホールドアウトデータ

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

ヒューhuman-in-the-loop (HitL)

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

同種データベースの移行

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

ホットデータ

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

ホットフィックス

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

ハイパーケア期間

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

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

「インダストリアル IoT」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (受信) VPC

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

増分移行

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

インダストリー 4.0

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

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

インダストリアル IoT (IIoT)

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

インスペクション VPC

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

IoT

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

解釈可能性

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

IoT

「IoT」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

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

大規模言語モデル (LLM)

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

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースアクセス制御」を参照してください。

最小特権

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

リフトアンドシフト

「7 Rs」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

LLM

「大規模言語モデル」を参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「ブランチ」を参照してください。

マルウェア

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

マネージドサービス

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

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

製造実行システム (MES)

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

MAP

「Migration Acceleration Program」を参照してください。

MCP

「モデルコンテキストプロトコル」を参照してください。

モデルコンテキストプロトコル (MCP)

エージェントツーツール通信のステートレスプロトコル。

MCP サーバー

Model Context Protocol を通じて 1 つ以上のツールを公開するサービス。

メカニズム

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

メンバーアカウント

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

MES

「製造実行システム」を参照してください。

Message Queuing Telemetry Transport (MQTT)

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

マイクロサービス

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

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)

オンラインツール。これによって、 AWS クラウドに移行するビジネスケースの検証に必要な情報を得られます。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェーブプランニング) 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 AWS クラウドに移行するために使用するアプローチ。詳細については、この用語集の 7 Rs エントリと、「組織を動員して大規模な移行を加速する」を参照してください。

ML

「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

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

モノリシックアプリケーション (モノリス)

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

MPA

「Migration Portfolio Assessment」を参照してください。

MQTT

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

多クラス分類

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

ミュータブルなインフラストラクチャ

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

O

OAC

「オリジンアクセス制御」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

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

OI

「オペレーション統合」を参照してください。

Ola

「オペレーショナルレベルアグリーメント」を参照してください。

オンライン移行

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

OPC-UA

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

Open Process Communications - Unified Architecture (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

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

運用テクノロジー (OT)

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

オペレーション統合 (OI)

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

組織の証跡

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

組織変更管理 (OCM)

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

オリジンアクセス制御 (OAC)

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

オリジンアクセスアイデンティティ (OAI)

CloudFront の、Amazon S3 コンテンツを保護するためのアクセス制限オプション。OAI を使用すると、CloudFront が、Amazon S3 に認証可能なプリンシパルを作成します。認証されたプリンシパルは、S3 バケット内のコンテンツに、特定の CloudFront デ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制御が可能です。

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用テクノロジー」を参照してください。

アウトバウンド (送信) VPC

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

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

「個人を特定できる情報」を参照してください。

プレイブック

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

PLC

「プログラマブルロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

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

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

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

プライバシーバイデザイン

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

プライベートホストゾーン

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

プロアクティブコントロール

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

製品ライフサイクル管理 (PLM)

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

本番環境

「環境」を参照してください。

プログラマブルロジックコントローラー (PLC)

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

プロンプトチェイニング

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

仮名化

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

発行/サブスクライブ (pub/sub)

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

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

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

RAG

「検索拡張生成」を参照してください。

ランサムウェア

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

RASCI マトリックス

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

RCAC

「行と列のアクセス制御」を参照してください。

リードレプリカ

読み取り専用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。

リアーキテクト

「7 Rs」を参照してください。

目標復旧時点 (RPO)

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

目標復旧時間 (RTO)

サービス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 とは独立しています。詳細については、「アカウントが使用できる AWS リージョン を指定する」を参照してください。

リグレッション

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

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 Rs」を参照してください。

リプラットフォーム

「7 Rs」を参照してください。

再購入

「7 Rs」を参照してください。

回復性

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

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

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

レスポンシブコントロール

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、「 AWSでのセキュリティコントロールの実装」の「レスポンシブコントロール」を参照してください。

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

検索拡張生成 (RAG)

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

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「目標復旧時点」を参照してください。

RTO

「目標復旧時間」を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーテ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 にログイン AWS マネジメントコンソール したり AWS API オペレーションを呼び出したりでき、組織内のすべてのユーザーを IAM で作成する必要はありません。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「SAML 2.0 ベースのフェデレーションについて」を参照してください。

SCADA

「監視制御とデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

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

セキュリティバイデザイン

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

セキュリティコントロール

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

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

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

サーバー側の暗号化

送信先にあるデータを、 AWS のサービス が受信する によって暗号化します。

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、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

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

責任共有モデル

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

シャドウ AI

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

SIEM

「Security Information and Event Management システム」を参照してください。

単一障害点 (SPOF)

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

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

スプリットアンドシードモデル

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

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

SPOF

「単一障害点」を参照してください。

スタースキーマ

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

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に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。

監視制御とデータ取得 (SCADA)

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

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

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

システムプロンプト

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

T

タグ

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

ターゲット変数

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

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要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。

tool

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

トランジットゲートウェイ

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

トランクベースのワークフロー

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

信頼されたアクセス

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

U

不確実性

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

未分化なタスク

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

上位環境

「環境」を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

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

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

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

ワークロード

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

ワークストリーム

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

WORM

「Write-Once-Read-Many」を参照してください。

WQF

「AWS ワークロード資格フレームワーク」を参照してください

Write-Once-Read-Many (WORM)

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

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性を悪用した攻撃 (一般的にマルウェアによる)。

ゼロデイ脆弱性

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

ゼロショットプロンプト

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

ゾンビアプリケーション

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

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