段階的な移行のために共有 IBM Db2 for z/OS データベースを使用してメインフレームアプリケーションをリプラットフォームする

AWS 規範ガイダンス

AWS 規範ガイダンス: 段階的な移行のために共有 IBM Db2 for z/OS データベースを使用してメインフレームアプリケーションをリプラットフォームする

Table of Contents

段階的な移行のために共有 IBM Db2 for z/OS データベースを使用してメインフレームアプリケーションをリプラットフォームする

Luis Gustavo Dantas と Andre Botura、Amazon Web Services (AWS)

2025 年 5 月 (ドキュメント履歴)

エンタープライズテクノロジーの絶えず変化する状況では、メインフレームのモダナイゼーションは、競争力と俊敏性を維持する必要がある組織にとって重要な要件となっています。この変換は、古いシステムを新しいシステムに置き換えるだけでなく、過去の堅牢で信頼性の高い基盤と、未来の動的で革新的な可能性とのギャップを埋める戦略的進化です。

メインフレームは、以前はエンタープライズコンピューティングの議論の余地のないリーダーでしたが、現在は転換点にあります。ピアレス処理能力とセキュリティ機能は数十年にわたって関連性を維持してきましたが、今日のビジネスには、クラウドサービスとシームレスに統合し、モバイルアプリケーションをサポートし、人工知能とビッグデータ分析の能力を活用できるシステムが必要です。

モダナイゼーションでは、必ずしもメインフレームから完全に移行する必要はありません。一部の組織は、メインフレーム環境とクラウド環境の両方の強みを活用するハイブリッドアプローチを選択しています。この戦略により、重要なレガシーアプリケーションはより最新のプラットフォームに徐々に移行しながら維持できます。この技術的な移行には、単なるシステム更新だけでなく、組織の文化とスキルを変革する必要があります。企業がモダナイズするにつれて、世代間のギャップを埋め、継続的な学習とイノベーションを促進することで、新しいテクノロジーとワークフォースの両方に投資します。

このガイドでは、メインフレームシステムの利点と最新のクラウドテクノロジーの利点のバランスを取る段階的な移行戦略について説明します。この段階的なリプラットフォームアプローチでは、まずアプリケーションレイヤーを移行し、既存の IBM Db2 for z/OS データベースへの接続を維持して、移行プロセスを合理化し、重要なビジネスオペレーションの中断を最小限に抑えながら、クラウドに新しい機能を導入します。

このガイドは、メインフレームのモダナイゼーションイニシアチブに関与する技術的な意思決定者と実装チームを対象としています。主な対象者には、エンタープライズアーキテクト、ソリューションアーキテクト、技術プロジェクトマネージャー、モダナイゼーションプログラムリーダーが含まれ、メインフレームリプラットフォームの戦略的側面と技術的側面の両方を理解する必要があります。コンテンツは、モダナイゼーションの実装を担当するメインフレームアプリケーション開発者、 AWS

クラウドエンジニア、データベース管理者、DevOps エンジニアなどの実装チームにとっても同様に重要です。

ビジネス上の成果

企業には、レガシーアプリケーションを更新する説得力のある理由が多数あります。このプロセスにより、業界間で緊急性が生まれます。古い専門家が退職すると、知識に大きなギャップが生じるため、この専門知識が失われる前にシステムをモダナイズすることが不可欠です。さらに、企業はコストを削減し、俊敏性を高め、急速に変化する市場状況に迅速に対応する必要性に支えられています。

デジタルトランスフォーメーションの推進は、新しいテクノロジーとカスタマーエクスペリエンスの向上への需要によってさらに強化されています。これらの要因は、複雑なシステムの維持に関連するリスクと相まって、組織が IT インフラストラクチャをモダナイズする際に迅速に行動するよう促しています。

特にメインフレームのモダナイゼーションは、デリケートなバランシング法を提示します。企業はメインフレームの安定性とセキュリティを維持する必要がありますが、最新のアーキテクチャが提供する柔軟性とスケーラビリティを採用する必要があります。このプロセスには、移行するアプリケーション、書き換えるアプリケーション、メインフレームに保持するアプリケーションに関する複雑な決定が含まれます。

モダナイゼーションの主な推進要因には、俊敏性とコスト削減があります。

- 俊敏性と市場投入までの時間。最新のシステムでは、調達プロセスが速くなり、変化する市場需要に迅速に対応できます。DevOps および SysOps プラクティスの導入により、生産性とデプロイの速度が大幅に向上します。
- コスト削減。モダナイゼーションは、多くの場合、以下を通じてインフラストラクチャコストを削減します。
  - 従Pay-as-you-goモデル。実際の使用量に合わせてコストを調整できます。
  - レガシーシステムに関連するライセンス料金を削減しました。
  - 伸縮性が向上し、リソースの割り当てが改善されました。
  - アクティブ/アクティブ、高可用性のセットアップ。リソース使用率を最適化しながらシステムの耐障害性を強化します。

これらのビジネスドライバーに基づいて、COBOL アプリケーションのリプラットフォームはモダナイゼーションへの戦略的アプローチと見なされます。共有データベースを使用して、モダナイゼー

ションの必要性とビジネス継続性を維持するための緊急性とのバランスが取れた段階的な移行パスをたどることができます。この方法により、COBOL アプリケーションの信頼性を維持しながら、最新のアーキテクチャの利点を活用できます。その結果、大規模な突然の移行に関連するリスクを軽減しながら、俊敏性、コスト効率、イノベーションを実現できます。このガイドで説明する共有 Db2 データベースアプローチは、レガシーシステムと最新のプラットフォームの架け橋となり、よりスムーズで制御されたモダナイゼーションプロセスを可能にします。

このガイドの内容

AWS Mainframe Modernization

> **Note**
>
> AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) は、新規のお客様に公開されなくなりました。 AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) に似た機能については、 AWS Mainframe Modernization サービス (セルフマネージドエクスペリエンス) をご覧ください。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「AWS Mainframe Modernization 可用性の変更」を参照してください。

このAWS Mainframe Modernization サービスを使用すると、レガシーメインフレームアプリケーションをクラウドネイティブ環境に移行し、既存のビジネスロジックと投資を維持し、自動化されたツールとマネージドランタイムサービスを使用し、アプリケーションのパフォーマンスを最適化し、運用コストを削減できます。このサービスはモダナイゼーションプロセスを合理化するため、コアメインフレームシステムの価値を維持しながらクラウドの能力を活用できます。 は、メインフレームのモダナイゼーションにリプラットフォームと自動リファクタリングという 2 つの主要なアプローチ AWS を提供します。

リプラットフォーム

AWS Mainframe Modernization Rocket Software (旧 Micro Focus) を使用したリプラットフォームは、メインフレームアプリケーションを最小限の中断でクラウドに移行したい企業に強力なリプラットフォームオプションを提供します。このソリューションを使用すると、コードを大幅に変更 AWS することなく、 で既存の COBOL および PL/I アプリケーションを再コンパイルして実行できます。

AWS Replatform with Rocket Software ソリューションの主な利点は次のとおりです。

- 既存のビジネスロジックと投資の保存
- リスクの軽減と市場投入までの時間の短縮
- AWS インフラストラクチャのスケーラビリティとパフォーマンスの向上
- 最新の開発ツールとプラクティスへのアクセス

このソリューションを使用すると、 の柔軟性、コスト効率、イノベーションを活用しながら、使い慣れたメインフレームプログラミング言語を維持できます AWS クラウド。

自動リファクタリング

リプラットフォームよりも変革的なアプローチとして、AWS Blu Age を使用できます。これにより、メインフレームアプリケーションを Java ベースのクラウドネイティブアプリケーションに自動的にリファクタリングできます。このソリューションは、レガシーシステムをより包括的にモダナイズし、クラウドネイティブテクノロジーを最大限に活用できるアプリケーションに変換するのに役立ちます。

AWS Blu Age の主な利点は次のとおりです。

- レガシーコードを最新の保守可能な Java アプリケーションに変換する
- 手動の労力と潜在的なエラーを減らす自動変換
- に最適化されたクラウドネイティブアプリケーションの作成 AWS のサービス
- 俊敏性が向上し、最新のテクノロジーとの統合が容易になりました

AWS Blu Age は、アプリケーションを移行してクラウドに備え、イノベーションと成長のための新しい可能性を開くのに役立ちます。このアプローチの詳細については、 ドキュメントの AWS Mainframe Modernization AWS 「Blu Age を使用してアプリケーションを自動的にリファクタリングする」を参照してください。

リプラットフォームの利点

このガイドでは、 でメインフレーム COBOL アプリケーションを再プラットフォームするためのアプローチについて説明します AWS。このアプローチは、IBM Db2 for z/OS を一時的に保持しながらレガシーシステムをモダナイズし、移行プロセスを合理化することを目指しています。既存のデータベース構造を最初に維持することで、移行中の複雑さとリスクを軽減できます。この段階的アプローチは、重要なデータ整合性 AWS クラウド を維持しながら、 のスケーラビリティとコスト効率のメリットを享受するのに役立ちます。段階的リプラットフォームの利点は次のとおりです。

- モダナイゼーションの加速: リプラットフォームとリファクタリングには通常、アプリケーション全体の書き換えを必要としないため、クラウド内のレガシーアプリケーションを再考するよりも時間とリソースが少なくて済みます。このアプローチは、組織が のスケーラビリティとコスト効率をすぐに享受しながら、自分のペースでモダナイズできるようにする、より段階的な移行もサポートします AWS クラウド。

- リスクの軽減: リプラットフォームには、多くの組織でリファクタリングよりもいくつかの利点があります。企業は、既存の COBOL および PL/I コードベースを維持し、長年のビジネスロジックを維持し、大規模なコード変更に関連するリスクを最小限に抑えることができます。
- データ継続性と段階的な移行: リプラットフォームの大きな利点は、最初に Db2 for z/OS のデータを元のデータ形式で保持することです。この戦略により、迅速で複雑でリスクの高いデータ移行プロセスが不要になります。初期段階でデータを元の環境に維持することで、データの整合性を維持し、ダウンタイムを減らし、モダナイゼーションプロセス中にデータ損失や破損のリスクを最小限に抑えることができます。2 番目のステップとして、アプリケーションがリプラットフォームされた環境で引き続き実行されている間、徹底的なテストと検証を伴うクラウドネイティブデータベースへの制御された段階的なデータ移行を計画できます。
- 柔軟性と将来性: メインフレームのスキルとアプリケーションに多大な投資を行っている企業にとって、リプラットフォームはイノベーションと継続性のバランスを取るモダナイゼーションへの実用的な道筋を提供します。重要なデータ構造とアクセスメソッドを最初に保持する柔軟性を提供すると同時に、完全なクラウドネイティブソリューションへの最終的なデータ移行など、将来のモダナイゼーションの取り組みのためのステージを設定します。

組織は、長期的なデジタルトランスフォーメーションの目標を計画しながら、リプラットフォームアプローチに従って自分のペースでモダナイズし、差し迫ったニーズに対応できます。このアプローチにより、企業はクラウドネイティブサービスについてスタッフをトレーニングする機会も得られます。

変換プロセス

メインフレームのモダナイゼーションは、貴重なレガシーアプリケーションを維持しながらクラウドコンピューティングの利点を活用する組織にとって重要なステップです。この変換には大きな課題があります。メインフレームアプリケーションは通常、高度に結合されており、数十年にわたる運用で進化してきた複雑な相互依存関係があります。この複雑さには、モダナイゼーションに対する慎重で体系的なアプローチが必要です。

組織は、移行を成功させるために、次の主要なフェーズをナビゲートする必要があります。

- 計画: このフェーズでは、既存のシステムの包括的な検出とモダナイゼーションの取り組みの優先順位付けを行います。組織は現在のインフラストラクチャを評価し、重要なアプリケーションを特定し、最初にモダナイズする必要があるシステムを決定します。
- 構築: この段階で、組織はアプリケーションを移行し、新しいシステムとインフラストラクチャを開発するプロセスを作成します。これには、モダナイズされたアーキテクチャの設計と実装、およびソースコードのコンパイルが含まれます。
- 実行中: このステップは、リプラットフォームされたアプリケーションをホストするランタイム環境の作成で構成されます。これには、モダナイズされたシステムをサポートし、新しい環境で効率的に動作するために必要なハードウェア、ソフトウェア、クラウドインフラストラクチャのセットアップが含まれます。
- テスト: このフェーズには、モダナイズされたシステムの厳格な検証が含まれ、すべての機能要件とパフォーマンス要件が満たされていることを確認します。データの整合性、システムの互換性、新しい環境の全体的なパフォーマンスを検証するために、広範なテストが実施されます。
- カットオーバー: 最終フェーズでは、スムーズな移行のための戦略の実装と、レガシーメインフレームからモダナイズされた環境への移行の制御に焦点を当てます。これには、事業運営の中断を最小限に抑えるための移行スケジュールと緊急時対応計画の慎重な計画が含まれます。

以下のセクションでは、これらのフェーズについて詳しく説明します。

- 計画
- 構築
- Running (実行中)
- テスト
- カットオーバー

計画

メインフレームレガシーアプリケーションの要件を効果的にナビゲートするために、組織は多くの場合、メインフレーム環境の包括的な評価から始めます。

アプリケーション検出

この初期フェーズの強力なツールは、メインフレームアプリケーションの構造、依存関係、複雑さに関する深い洞察を提供する Rocket Enterprise Analyzer です。このツールは、モダナイゼーションの取り組みの範囲、潜在的なリスク、最適化の機会を決定するのに役立ちます。

発見すべき重要な側面の 1 つは、メインフレームシステム内のデータの依存関係の複雑なウェブです。これらの依存関係はレガシーコードのレイヤーの下に隠れることが多く、モダナイゼーションの取り組みに大きな影響を与える可能性があります。さまざまなアプリケーションやモジュールがさまざまなデータソースとどのように相互作用するかをマッピングすることで、実装する予定の変更の潜在的な影響をよりよく理解できます。

データ依存関係

データ依存関係を徹底的に評価することで、メインフレーム環境内のデータフロー、データ品質、データガバナンスに関する重要な情報を明らかにすることができます。この知識は、データ移行戦略の計画、モダナイゼーション中のデータ整合性の確保、データ最適化の機会の特定に非常に役立ちます。データを明確に把握することで、どのモダナイゼーションアプローチが既存のオペレーションに対して最も効果的で、最も破壊的でないかについて、より多くの情報に基づいた意思決定を行うことができます。

トランザクションまたはジョブコントロール言語 (JCL) ジョブによってテーブルの使用を識別するトップダウン分析は、ウェーブ計画と優先順位付けを作成する上で重要です。このアプローチは、メインフレームシステムのさまざまなコンポーネント間の関係を明確にし、モダナイゼーションへの戦略的で段階的なアプローチを開発するのに役立ちます。最も頻繁にアクセスされるテーブルとプロセスを特定することで、モダナイゼーションの取り組みに優先順位を付けることができます。まず影響の大きい領域に集中し、重要なビジネスオペレーションへの中断を最小限に抑えながら、よりスムーズな移行を確保できます。

Rocket Enterprise Analyzer を使用してデータの依存関係を検出するだけでなく、多くの組織は独自のカスタムビルドソリューションを使用してメインフレーム環境に関するより深い洞察を得ています。これらの社内ツールは、IBM Db2 カタログと System Management Facility (SMF) レコードで利用可能な豊富な情報を悪用することがよくあります。

キャパシティベンチマーク

メインフレームの再プラットフォームプロジェクトを計画する 1 つのステップは、現在のワークロードの消費量に関する詳細情報を収集することです。このデータは、ターゲットクラウド環境で必要な初期容量を正確に予測してプロビジョニングするのに役立ちます。例えば、IBM Customer Information Control System (CICS) ジョブまたは Information Management System (IMS) ジョブとジョブコントロール言語 (JCL) ジョブから、オンライントランザクションとバッチトランザクションの両方について、1 秒あたり 100 万回の命令 (MIPS) 消費データを収集することをお勧めします。

IBM は、メインフレームコンピューティングにおける MIPS のさまざまな料金モデルを提供しており、これらのモデルの多くはピーク時の使用量を中心としています。これらのピークベースのモデルの中で、最も一般的なのはローリング 4 時間のピークです。

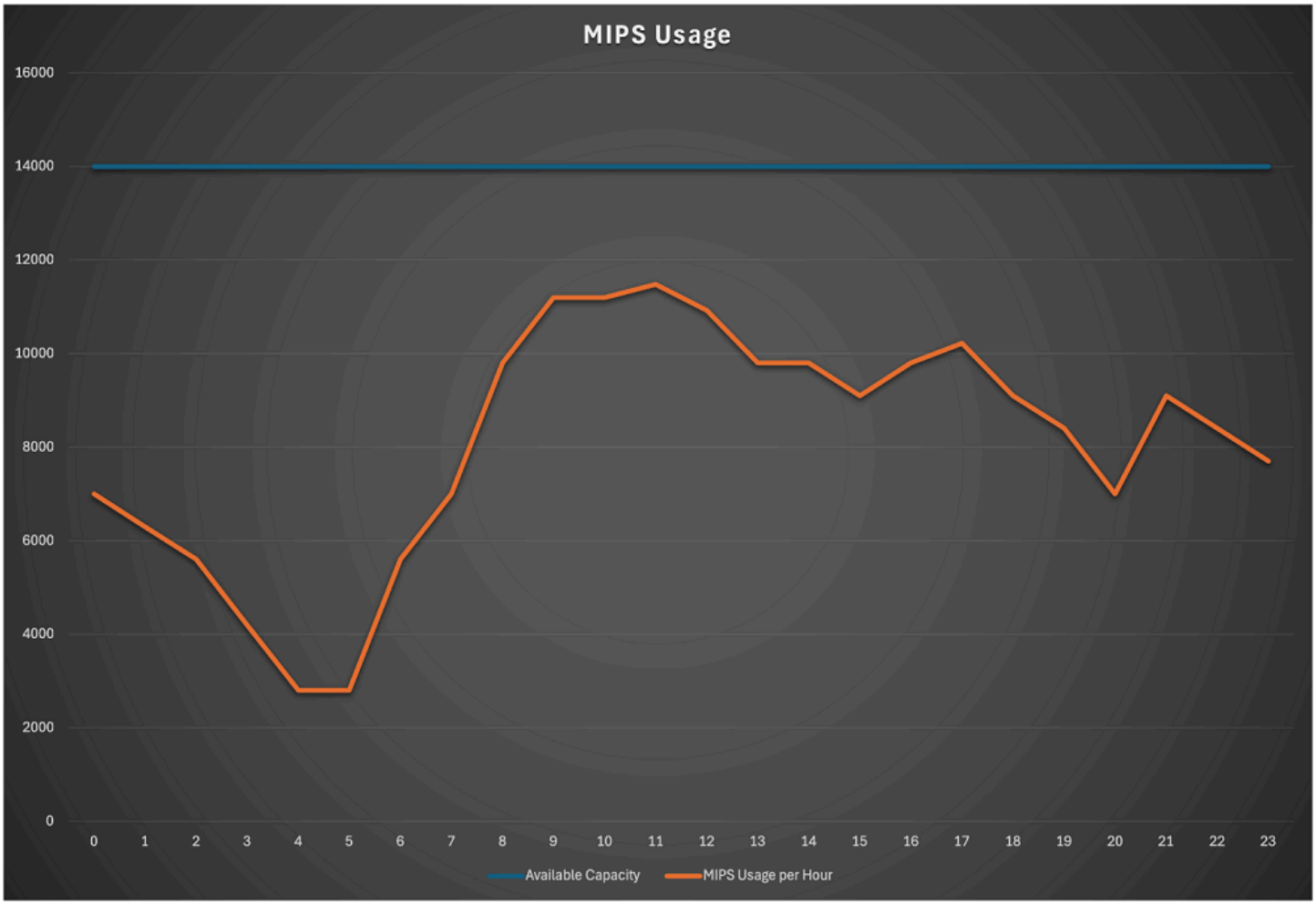
メインフレームコストには、全体的なコストに大きな影響を与える 5 つの主要領域が含まれます。

- 多くの場合、ソフトウェアライセンスは主要なコンポーネントです。オペレーティングシステム、ミドルウェア、データベース、およびさまざまなアプリケーションを対象としており、コストはマシンの容量や使用量に関連する場合があります。
- ハードウェア費用には、メインフレーム機器の初回購入またはリース、継続的なメンテナンス、アップグレードが含まれます。
- 大量のマネージドデータを管理し、ディスクシステム、テープライブラリ、および関連する管理ソフトウェアが含まれるため、ストレージコストは高額になる可能性があります。
- 人的支出は、システムプログラマーやデータベース管理者などの特殊なメインフレームプロフェッショナルの給与をカバーします。
- バックアップシステム、冗長ハードウェア、オフサイト復旧施設などのディザスタリカバリとビジネス継続性対策は、高可用性と迅速な復旧を確保する上で大きな投資となります。

これら 5 つのコストカテゴリは、MIPS ベースの料金と組み合わせて、ほとんどのメインフレーム予算の中核を形成します。ただし、相対的な比率は、組織のサイズ、業界、特定のメインフレーム使用率パターンによって大きく異なる場合があります。

時間単位の MIPS データは、メインフレームのワークロードパターンとパフォーマンスを包括的に理解するために不可欠です。日単位または月単位の平均とは異なり、時間単位のデータは、1 日を通してシステムのリソース使用率の微妙な変動を明らかにする詳細なインサイトを提供します。この詳細レベルは、クラウド内のアプリケーションのパフォーマンスと容量のニーズを正確に評価するために非常に重要です。

時間単位の MIPS データを分析することで、次の図に示すように、ピーク使用期間を特定し、傾向を特定し、集計データで隠されている可能性のあるボトルネックを特定できます。この詳細度により、より正確な容量計画が可能になり、リソースの割り当てを最適化し、コスト削減とシステム効率の向上につながる可能性があります。



時間単位の MIPS データは、重要なパフォーマンスベンチマークツールとしても役立ちます。これにより、システムのパフォーマンスの詳細なベースラインが確立されます。これは、移行やアップグレードなどのシステム変更を計画または評価する場合に特に役立ちます。変更前と変更後の時間単位の MIPS データを比較することで、これらの変更がシステムのパフォーマンスに与える影響を正確に測定し、メインフレームが引き続き組織のニーズを満たしていることを確認することができます。

時間単位の MIPS データを収集するには、いくつかのオプションがあります。1 つの方法は、SMF レコードを直接使用することです。これらのレコードは、システムアクティビティとリソースの使用状況に関する豊富な情報を提供します。または、IBM Sub-Capacity Reporting Tool (SCRT) などの特殊なツールを使用して、MIPS データの収集と分析のプロセスを簡素化することもできます。

選択した方法にかかわらず、長期間、理想的には数か月にわたってデータを収集することが重要です。この収集期間の延長により、end-of-monthの処理の急増や季節的な変動など、ワークロードの周期的な変動を考慮できます。これらの長期パターンをキャプチャすることで、メインフレームのパフォーマンス特性をより正確かつ包括的に把握できるため、情報に基づいた意思決定とより効果的なキャパシティ管理が可能になります。

ウェーブプランニング

収集した情報を使用して、メインフレームのリプラットフォームイニシアチブに戦略的に優先順位を付けることができます。賢明なアプローチは、非中核的なビジネストランザクションやバッチジョブなど、重要度の低いワークロードから始めて、チームが経験を積み、重要な運用のリスクを最小限に抑えながらプロセスを改良できるようにすることです。さらに、読み取り専用ワークロードを移行の早期候補として考慮すると、これらのワークロードには通常、複雑さが少なく、データの不整合のリスクが低いため、有利です。このアプローチにより、リプラットフォームの取り組みに自信と勢いをつけることができます。

さらに、書き込みまたは更新オペレーションのために Db2 テーブルを共有するワークロードをグループ化すると、移行プロセスを合理化できます。これらの相互接続されたワークロードを特定することで、データの整合性を維持し、複雑な中間ソリューションの必要性を最小限に抑える、まとまりのある移行ウェーブを計画できます。この戦略は、データ競合のリスクを減らすだけでなく、関連するコンポーネントに同時に対処することで、全体的なリプラットフォームタイムラインを最適化します。最終的に、このデータ駆動型の優先順位付けアプローチにより、重要度、複雑さ、相互依存のバランスがとれ、メインフレームのモダナイゼーションプロセスがより効率的かつ成功します。

Building

共有 Db2 データベースを使用すると、メインフレーム環境とクラウド環境の両方で、同一または一貫性のあるアプリケーションを同時に実行できます。このアプローチは、両方のプラットフォームで同じアプリケーションバージョンを維持する場合、いくつかの利点があり、オペレーションの柔軟性と信頼性が向上します。

この戦略の主な利点の 1 つは、効果的なロールバックプランを実装できることです。移行またはデプロイ中に問題が発生した場合、同じアプリケーションバージョンを使用すると、以前の状態にシームレスに戻ることができ、ダウンタイムと潜在的なデータの不整合を最小限に抑えることができます。

アプリケーションの整合性

アプリケーションコンポーネントを分散ソースコントロールマネージャーからメインフレームにミラーリングすることは、リプラットフォームプロセス中の戦略的アプローチです。このメソッドは、メインフレーム環境との同期を維持しながら、最新のソースコード管理ツールの使用をサポートします。このミラーリングプロセスは一時的なもので、ワークロードが分散プラットフォームの本番環境で完全に機能するまで続きます。

リプラットフォームされたアプリケーションのソースコードを分散変更管理ツールに移行することで、最新のソースコードマネージャーが提供するいくつかの利点を活用できます。具体的には次のとおりです。

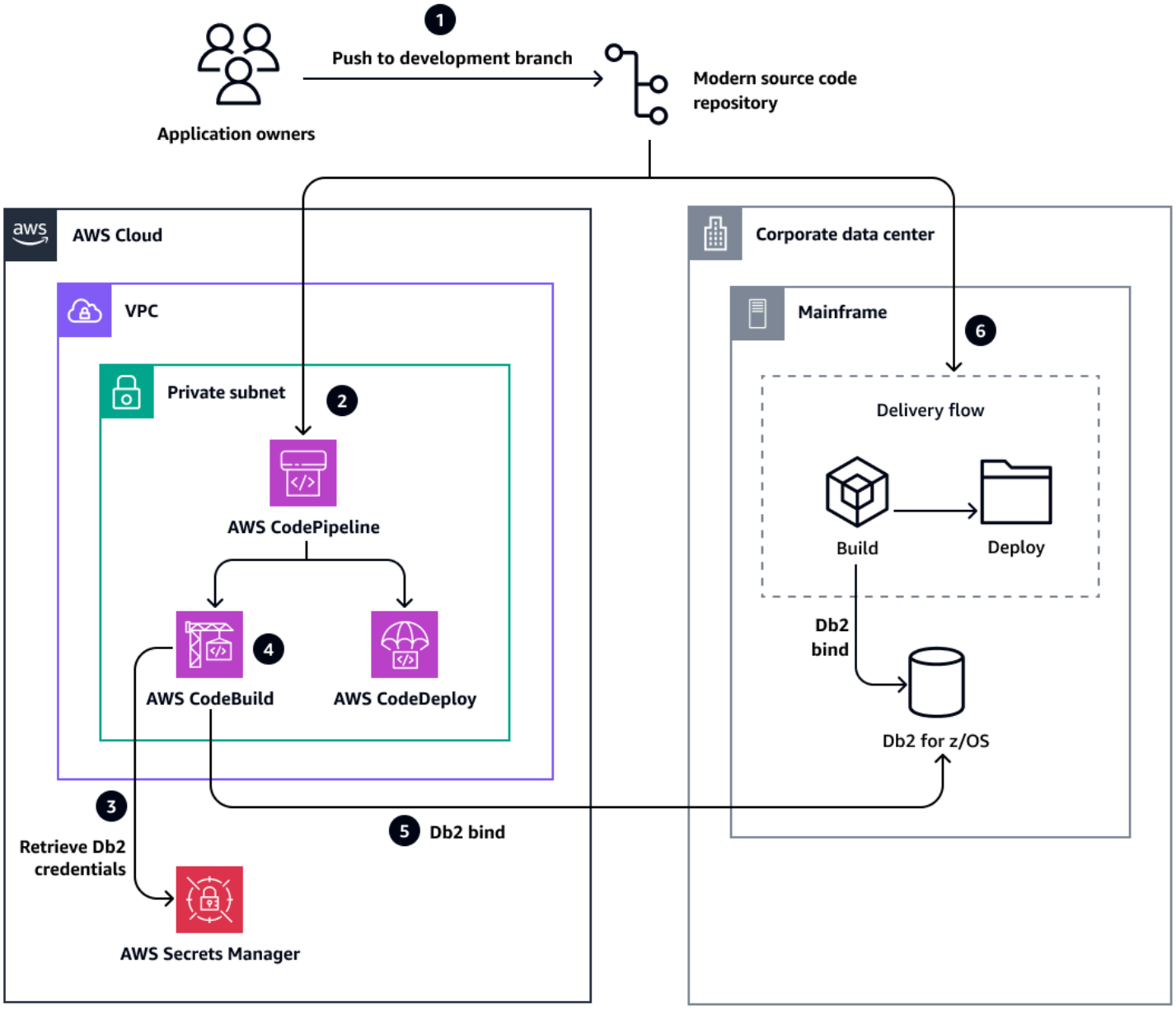
- コラボレーションの強化: 分散ツールは、多くの場合、プルリクエスト、コードレビュー、分岐戦略などの機能を含めることで、チームコラボレーションのサポートを強化します。
- バージョン管理の改善: 最新のシステムでは、より詳細なバージョン管理が提供され、変更の追跡やコードのさまざまなバージョンの管理が容易になります。
- CI/CD パイプラインとの統合: 多くの分散ツールは、継続的インテグレーションおよび継続的デプロイ (CI/CD) パイプラインとシームレスに統合され、開発プロセスを合理化します。
- 可視性とトレーサビリティの向上: これらのツールは、多くの場合、優れたダッシュボードとレポート機能を提供し、開発プロセスに関するより深い洞察を提供します。
- 最新の開発プラクティスのサポート: 分散システムは、通常、アジャイルな方法論や DevOps プラクティスに適しています。

ミラーリングプロセスでは、分散ソースコントロールマネージャーからメインフレームにコードを同期します。これにより、移行期間中も両方の環境の一貫性が保たれます。ただし、双方向ではなく分散システムからメインフレームに更新が流れる一方向同期としてミラーリングを実装する必要があります。このアプローチは一貫性を維持し、両方の環境での同時更新によって発生する可能性のある競合を防ぎます。

このミラーリング戦略を採用することで、メインフレーム環境をup-to-date状態に保ちながら、開発作業を分散プラットフォームに徐々に移行できます。これにより、リプラットフォームプロセス中にスムーズな移行と安全ネットが提供されます。ワークロードが分散本番環境で完全に機能し、安定している場合は、ミラーリングプロセスを段階的に廃止し、最新のソースコード管理システムへの移行を完了できます。

アーキテクチャ

次の図は、分散ソースコード管理システムがアプリケーションコンポーネントを AWS クラウドミラーリングし、 とメインフレーム環境間の同期を維持する方法を示しています。 AWS クラウド 環境では、AWS CodeBuild、、 などの CI/CD サービスを使用してアプリケーションAWS CodePipelineAWS CodeDeployを構築およびデプロイします。



このワークフローでは、次の操作を行います。

1. アプリケーション所有者は、ソースコードリポジトリの開発ブランチに新しいアプリケーションリリースを配信します。

2. 新しいリリースがトリガーされます AWS CodePipeline。
3. AWS CodeBuild は から Db2 認証情報を取得しますAWS Secrets Manager。
4. CodeBuild はアプリケーションをコンパイルします。
5. CodeBuild は Db2 for z/OS を使用してアプリケーションをバインドします。
6. メインフレーム配信フローは、アプリケーションも構築およびデプロイします。

実行中

クラウドベースのアプリケーションとオンプレミスデータベース間の最適なパフォーマンスと低レイテンシーを確保するために、 を実装することをお勧めしますAWS Direct Connect。このサービスは、 AWS と組織のデータセンター間の専用ネットワーク接続を提供し、インターネットベースの接続と比較して、より一貫したネットワークパフォーマンスとレイテンシーの低減を実現します。これは、迅速な応答時間を必要とするデータベースオペレーションにとって特に重要です。

で実行されているアプリケーションの高可用性 (HA) と伸縮性を実現するには AWS、次のコンポーネントを使用して堅牢なアーキテクチャを実装できます。

- Elastic Load Balancing (ELB): ロードバランサーをデプロイして、アプリケーションが実行される複数の Amazon Elastic Compute Cloud (Amazon EC2) インスタンスに受信トラフィックを分散できます。これにより、ワークロードが均等に分散され、クライアントリクエストのエントリポイントが 1 つになります。
- Auto Scaling グループ: アプリケーションをホストする EC2 インスタンスを Auto Scaling グループに整理できます。これにより、インフラストラクチャは CPU 使用率やネットワークトラフィックなどの事前定義されたメトリクスに基づいてインスタンスの数を自動的に調整できます。ピーク時には、負荷の増加に対応するために追加のインスタンスを起動できますが、クワイエット時には、不要なインスタンスを終了してコストを最適化できます。
- EC2 インスタンス: アプリケーションを Auto Scaling グループ内の EC2 インスタンスにデプロイできます。これらのインスタンスは、耐障害性を強化し、高可用性を確保するために、複数のアベイラビリティーゾーンに分散する必要があります。
- マルチ AZ 配置: アプリケーションインスタンスを複数のアベイラビリティーゾーンに分散することで、システムは 1 つのアベイラビリティーゾーンの障害に耐えられ、全体的な可用性に大きな影響を与えません。

このアーキテクチャにより、アプリケーションは高可用性を維持しながら、需要に応じてシームレスにスケーリングできます。ロードバランサーは、トラフィックが正常なインスタンスに均等に分散さ

れるようにし、Auto Scaling グループは実際のワークロードに基づいてインスタンスの数を管理します。

信頼性をさらに高めるために、Amazon CloudWatch を使用してパフォーマンスの問題や障害を迅速に検出して対応できるようにすることで、堅牢なモニタリングおよびアラートシステムを実装できます。さらに、自動スケーリング機能とフェイルオーバーシナリオを定期的にテストすることで、さまざまな負荷状況や潜在的な障害時にシステムが期待どおりに動作することを確認できます。

このアプローチを採用することで、オンプレミスの Db2 データベースへの安全な接続 AWS クラウド を維持しながら、 のスケーラビリティと柔軟性を活用できます。このハイブリッドセットアップは、完全なクラウド移行への優れたパスとして機能し、プロセス全体で段階的な移行とリスク軽減を提供します。

2 フェーズコミット (2PC)

AWS Mainframe Modernization Rocket Software とのリプラットフォームは、拡張アーキテクチャ (XA) の実装を通じて 2 フェーズコミット (2PC) トランザクションをサポートします。この機能は、分散システム全体、特に複雑なトランザクションが複数のリソースにまたがることが多いメインフレーム環境でデータの整合性を維持する上で不可欠です。

XA アーキテクチャは、Rocket Software で AWS Replatform に統合されており、データベースやメッセージキューなどのさまざまなリソース間でトランザクションを調整できます。この統合により、分散トランザクションのすべての部分がコミットまたはロールバックされ、システム全体の一貫性が維持されます。

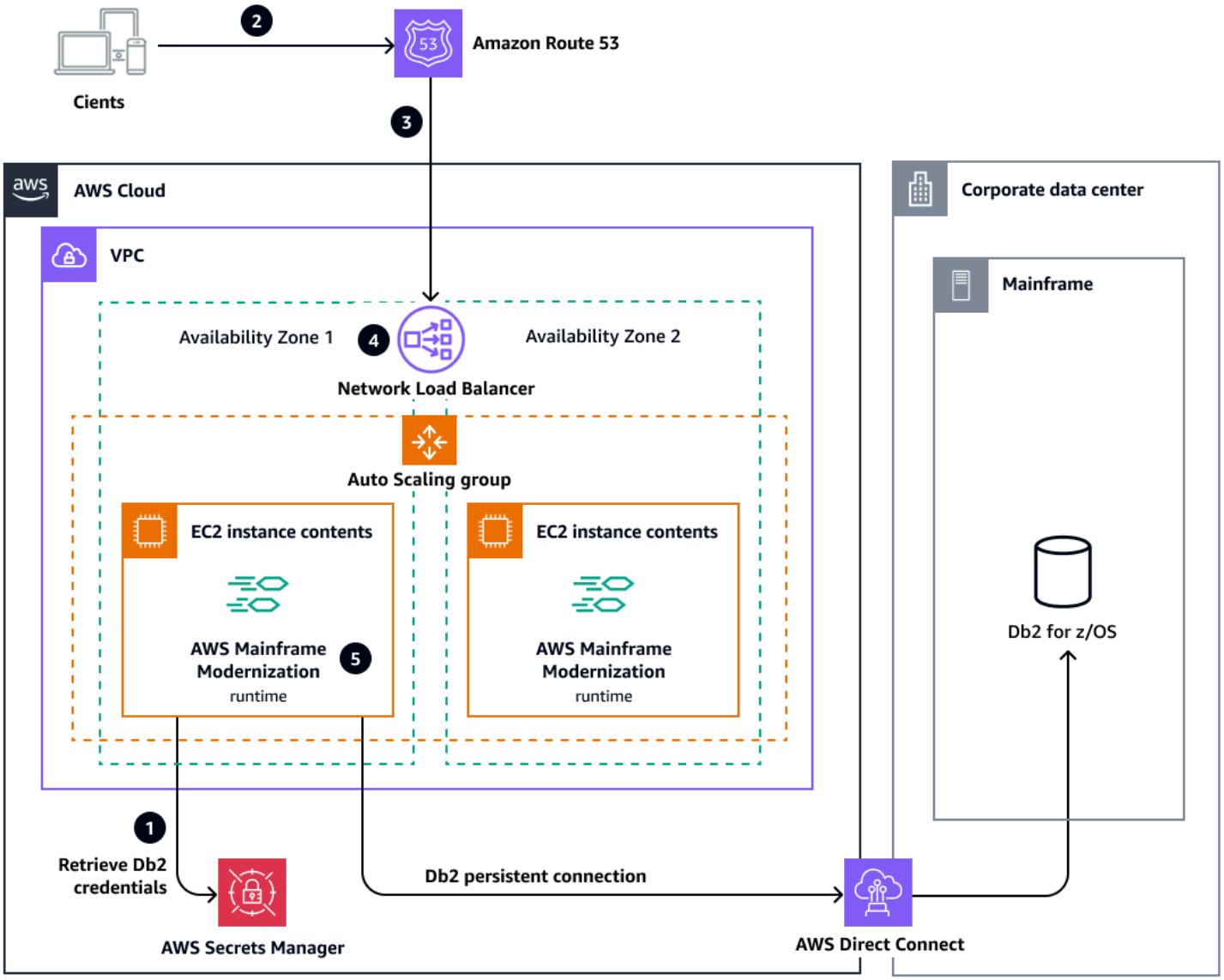
2PC プロセスは 2 つのフェーズで構成されます。

- 準備フェーズ: トランザクションマネージャーは、トランザクションに関係するすべてのリソースマネージャーにクエリを実行して、コミットする準備ができていることを確認します。
- コミットフェーズ: すべてのリソースマネージャーが肯定的に応答した場合、トランザクションマネージャーは変更をコミットするように指示します。いずれかのリソースマネージャーがコミットできない場合、すべてのマネージャーに変更をロールバックするように指示されます。

XA を使用することで、Rocket Software AWS で Replatform は、モダナイズされたメインフレーム環境で複雑な分散トランザクションを管理するための信頼性が高くスケーラブルなソリューションを提供します。この機能は、トランザクションの整合性やパフォーマンスを損なうことなくメインフレームアプリケーションをクラウドに移行したい組織にとって不可欠です。

ランタイムインフラストラクチャ

次の図は、2 つのアベイラビリティーゾーン、Auto Scaling グループ内の EC2 インスタンス、Network Load Balancer、 を介した AWS とメインフレーム環境間の専用接続 AWS クラウド を含む の高可用性で伸縮自在な環境を示しています AWS Direct Connect。



このアーキテクチャの詳細は以下のとおりです。

1. AWS Mainframe Modernization ランタイムが開始されると、 から Db2 認証情報を取得しAWS Secrets Manager、Db2 for z/OS との永続的な接続を開きます。

> **Note**
>
> AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) は、新規のお客様に公開されなくなりました。 AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) に似た機能については、 AWS Mainframe Modernization サービス (セルフマネージドエクスペリエンス) をご覧ください。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「AWS Mainframe Modernization 可用性の変更」を参照してください。

2. クライアントは Amazon Route 53 の Network Load Balancer アドレスをバインドします。
3. Route 53 は、トランザクションを Network Load Balancer にリダイレクトします。
4. Network Load Balancer は、複数の EC2 インスタンスにトランザクションを分散します。
5. で実行されているワークロードは、 を介した永続的な接続を使用して Db2 for z/OS と AWS Mainframe Modernization やり取りします AWS Direct Connect。

テスト

Db2 for z/OS を共有データベースとして維持しながら COBOL アプリケーションを再プラットフォームする場合は、新しいシステムが元の と同等に機能するようにすることが重要です。このハイブリッド環境には、独自の課題とテストの機会があります。次の戦略は、機能同等性テストへの包括的なアプローチの概要を示し、リプラットフォームされたアプリケーションのパフォーマンス、データ整合性、既存の Db2 for z/OS データベースとのシームレスな統合を検証するように設計されています。

まず、システム間で比較する必要がある重要なビジネスプロセスとトランザクションを特定します。次に、これらのトランザクションの機能同等性を効果的に評価する特定のシナリオを含む詳細なテストプランを作成します。最後に、特定されたすべてのシナリオをカバーする包括的なテストデータセットを作成し、正確な比較を可能にするために両方のシステムで同一であることを確認します。

ソース環境

- 初期スナップショット (最初のスナップショット):
  - 同等性テストに影響する可能性があるため、テスト中にデータテーブルが他のアプリケーションで使用されていないことを確認してください。

  - テストを実行する前に、トランザクションで使用される Db2 for z/OS テーブルのスナップショットを作成します。
- ソースシステムのテスト:
  - 元の COBOL アプリケーションでテストの完全なスイートを実行します。
  - すべてのトランザクション、入力、出力を記録します。
  - システムパフォーマンスとリソース使用率をモニタリングします。
- ソーステスト後のスナップショット (2 番目のスナップショット):
  - ソースシステムのテストが完了したら、Db2 for z/OS データベースの別のスナップショットを作成します。

ターゲット環境

- データベースのリセット:
  - 最初のスナップショットを使用して、データベースを初期状態に復元します。
- ターゲットシステムテスト (リプラットフォーム環境):
  - リプラットフォームされたアプリケーションで同じテストスイートを実行します。
  - すべてのターゲットシステムテストで、ソースシステムテストと同じ入力が使用されていることを確認します。
  - システムパフォーマンスとリソース使用率をモニタリングします。
- ターゲットテスト後のスナップショット (3 番目のスナップショット):
  - ターゲットシステムテストが完了したら、Db2 for z/OS データベースの最終スナップショットを作成します。

分析

- 比較と分析:
  - 2 番目と 3 番目のスナップショットを比較して、データの不一致を特定します。
  - テスト結果を分析し、ソースシステムとターゲットシステムからの出力を比較します。
  - 2 つの環境間のパフォーマンスメトリクスを評価します。
- 統合テスト:
  - リプラットフォームされたアプリケーションと残りの COBOL コンポーネントの両方を含むテストを実行します。

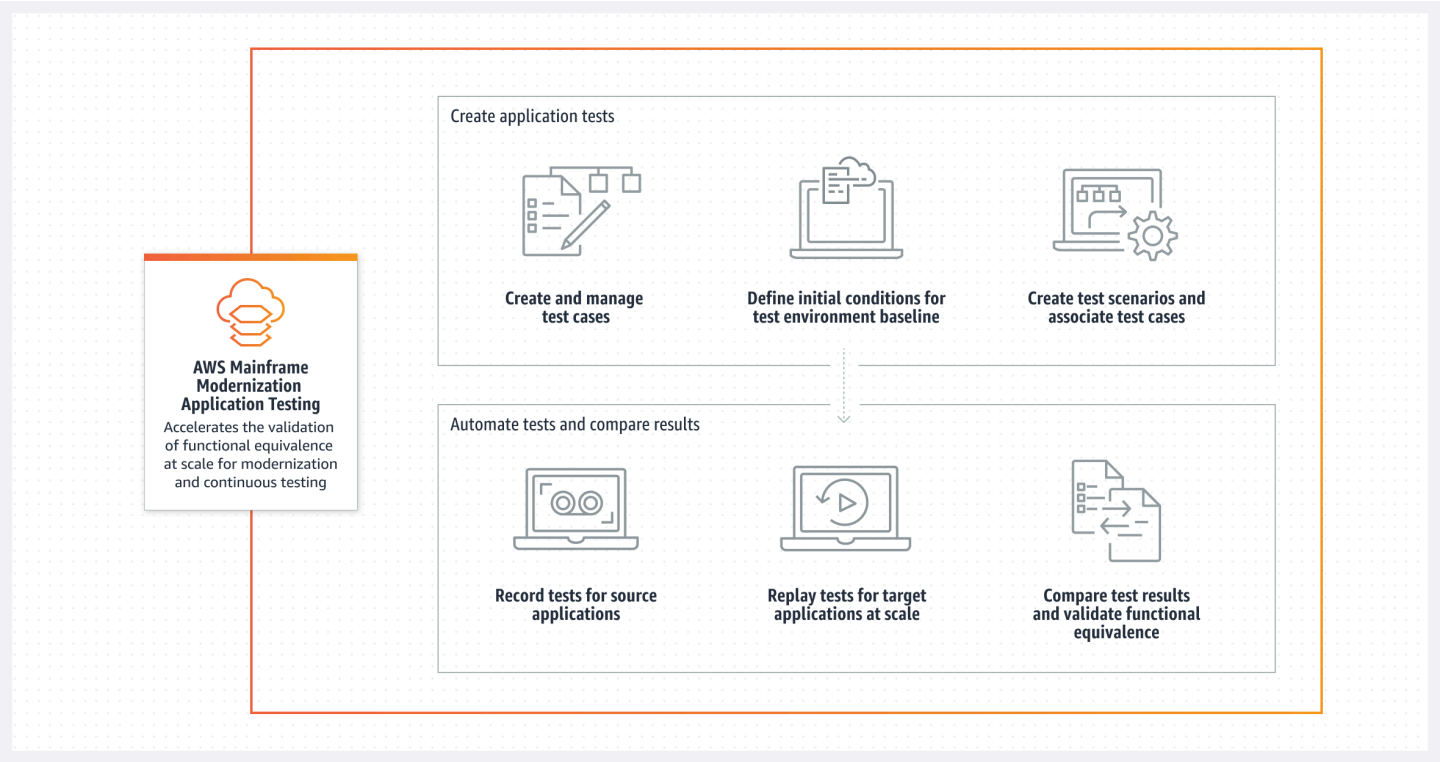
  - 2 つの環境間のシームレスな相互作用を検証します。
- フェイルオーバーと復旧のテスト:
  - 一方の環境が失敗し、もう一方の環境が引き継ぐシナリオをテストします。
  - フェイルオーバー時にデータの整合性と整合性を確保します。
- 負荷テストとストレステスト:
  - さまざまな負荷でテストを行い、ハイブリッドシステムがストレス下でどのように動作するかを評価します。
  - どちらの環境でもボトルネックやパフォーマンスの問題を特定します。
- ドキュメントとレポート:
  - すべてのテスト結果、不一致、パフォーマンスメトリクスを文書化します。
  - ソースシステムとターゲットシステムを比較する包括的なレポートを作成します。

でのアプリケーションのテスト AWS Mainframe Modernization

このAWS Mainframe Modernization Application Testingサービスは、大規模なアプリケーションテストの実行を自動化します。 AWS Application Testing これにより、メインフレームアプリケーションのモダナイゼーションとテストプロジェクトのコストを最適化して削減できます。

> **Note**
>
> AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) は、新規のお客様に公開されなくなりました。 AWS Mainframe Modernization サービス (マネージドランタイム環境エクスペリエンス) に似た機能については、 AWS Mainframe Modernization サービス (セルフマネージドエクスペリエンス) をご覧ください。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「AWS Mainframe Modernization 可用性の変更」を参照してください。

次の図は、 が高レベルで AWS Application Testing どのように機能するかを示しています。



このプロセスは次の 4 つの手順で構成されます。

1. テストアクションの最小単位であるテストケースを作成および管理します。ソースシステムとターゲットシステム間の機能的同等性を最もよく表すデータ型を特定します。
2. テンプレートと追加の属性を指定 CloudFormation して、テスト環境の設定を定義します。
3. テストケースのコレクションであるテストスイートを作成します。
4. データセットをアップロードして再生する: メインフレームで入出力データセットをキャプチャし、 にアップロードしてから AWS、ターゲットシステムでテストシナリオを再生します。
5. ソースデータセットとターゲットデータセットを比較します。 は、ソースシステムとターゲットシステムからの出力データセット AWS Application Testing を自動的に比較します。これらを確認して評価し、不一致を特定します。

詳細については、AWS Mainframe Modernization ドキュメントを参照してください。

カットオーバー

メインフレームのモダナイゼーションにおける最も重要な課題の 1 つは、新しいプラットフォームへの移行中のダウンタイムとリスクを最小限に抑えることです。Blue/Green デプロイ戦略は、システム移行に対する強力で柔軟なアプローチを提供します。

ブルー/グリーンデプロイは、ブルーとグリーンという 2 つの同一の本番環境を実行することで、ダウンタイムとリスクを軽減する手法です。メインフレームのモダナイゼーションコンテキストでの仕組みは次のとおりです。

- ブルー環境: これは、すべての本番トラフィックを処理する現在のメインフレームシステムです。
- グリーン環境: これは、引き継ぐ準備ができてい AWS る の新しいモダナイズされたプラットフォームです。

ブルー/グリーンカットオーバー戦略には、プロビジョニング、本番稼働、問題が発生した場合のロールバック、終了のステップが含まれます。

プロビジョニング

この段階では、以下の手順に従って に新しい (グリーン) AWS 環境をプロビジョニングします。

1. 環境をリプラットフォームする: Route 53 ホストゾーンには、メインフレーム環境 (青) を指す DNS レコードが含まれている必要があります。
2. 接続の確認: AWS アカウント とオンプレミスのトランザクションマネージャーと Db2 for z/OS データベース間の適切な接続を確認します。
3. 煙テストを実行する: AWS ロードバランサーアドレスを使用してリプラットフォームされた環境にアクセスし、包括的な煙テストを実行して以下を確認します。
   - 予想されるすべてのワークロードを使用できます。
   - 3270 件のトランザクションが正しく処理されています。
   - Db2 for z/OS とのデータインタラクションは期待どおりに機能しています。

稼働

この段階では、トラフィックをグリーン環境にシフトし、変更をモニタリングします。

1. Route 53 のトラフィックルーティングポリシーを使用して、トラフィックをシフトします。
   - オプション A: トラフィックをすべて一度にシフトできます。
   - オプション B: または、段階的な加重分散を使用することもできます。
2. モニタリングと検証:
   - トラフィックの変化に応じて AWS 環境を注意深く監視します。
   - 3270 トランザクション処理を確認します。

- Db2 for z/OS 通信を確認します。
- パフォーマンスの問題を監視します。
- ユーザーにトランザクション結果の検証を依頼します。

ロールバック

問題が発生した場合は、Route 53 をすばやく更新して、トラフィックをオンプレミスのメインフレーム (ブルー) 環境にリダイレクトできます。

別のカットオーバーを試みる前に、問題を調査して解決する必要があります。

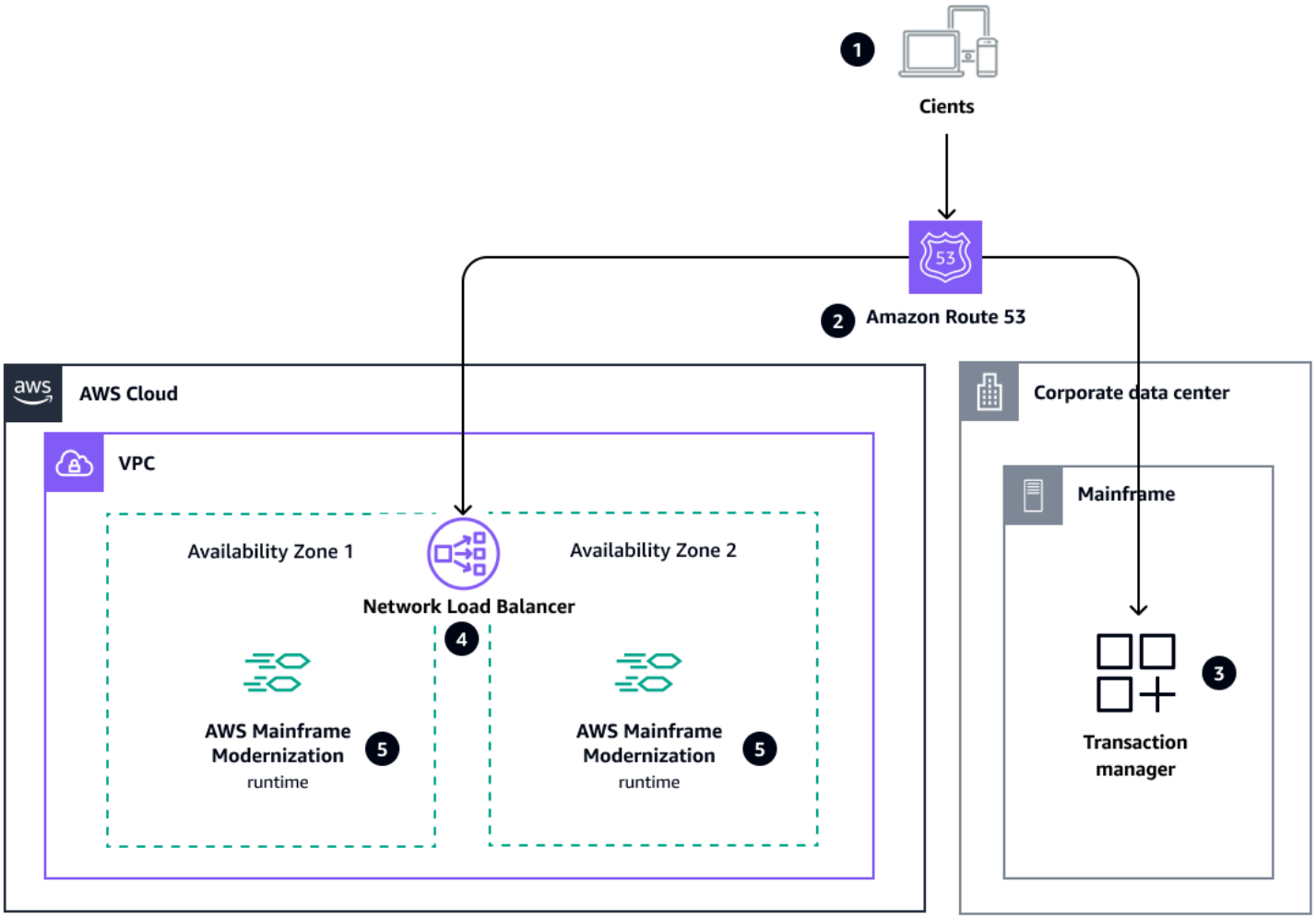
終了

トラフィックをモニタリングし、グリーン環境が正しく機能していることを検証したら、アプリケーショントラフィックを に徐々に増やすことができます AWS。

安定した期間を過ぎると、メインフレームトランザクション (ブルー) 環境を廃止し、Db2 for z/OS データベースをオンプレミスに保持できます。

アーキテクチャ

次の図は、カットオーバーフローを示しています。



カットオーバープロセスは、以下で構成されます。

1. フロントエンド (BFFs) のクライアントアプリケーション、フロントエンド、バックエンドは、Route 53 ドメイン名にトランザクションを送信します。
2. Route 53 は、定義されたルーティングポリシーに応じて、メインフレームトランザクションマネージャーまたは Network Load Balancer に接続をルーティングします。
3. トランザクションマネージャーは、メインフレームに送信されるトランザクションを処理します。
4. Network Load Balancer は、トランザクションを利用可能なリプラットフォーム環境に分散して処理します。
5. AWS Mainframe Modernization リプラットフォーム環境はリクエストを処理します。

ベストプラクティス

このセクションでは、データベースを Db2 for z/OS に維持しながら、メインフレームワークロードをクラウド環境に再プラットフォームする際の主要な課題に対処するための一連のベストプラクティスの概要を説明します。

ネットワークのレイテンシー

リプラットフォーム作業中にアプリケーションを Db2 データベースから分離することによるレイテンシーへの影響を正確に予測するには、トランザクションとバッチプロセスの両方で Db2 呼び出しの数を徹底的に評価することをお勧めします。この評価はトレースデータを使用して実行し、次のステップを含める必要があります。

- トレースデータの収集: 代表的なトランザクションとバッチジョブの詳細なトレースを収集し、そのトレースがエントリと終了を含むすべての Db2 インタラクションをキャプチャしていることを確認します。
- トレースデータを分析する: 各トランザクションとバッチジョブの Db2 エントリと終了の数をカウントし、トランザクションとバッチプロセスあたりの Db2 インタラクションの平均数を計算します。
- 現在の応答時間を測定する: Db2 アクセスがアプリケーションのサービスレベルアグリーメント (SLA) と一致しているかどうかを確認します。
- ネットワークレイテンシーの推定: リプラットフォームされたアプリケーションと Db2 データベース間の予想されるネットワークレイテンシーを決定します。物理的な距離、ネットワークインフラストラクチャ、潜在的なボトルネックなどの要因を考慮してください。
- 潜在的な影響を計算する: トランザクションとバッチプロセスごとに、Db2 エントリと終了の数に推定ネットワークレイテンシーを掛けます。この計算時間を現在の応答時間に追加して、新しい合計処理時間を予測します。
- 結果を評価する: 予測されるレイテンシーの増加がビジネス要件に適しているかどうかを評価し、レイテンシーの問題を軽減するために最適化または再設計を必要とする可能性のあるトランザクションやプロセスを特定します。
- 緩和戦略を検討する: 接続プーリング、キャッシュ、バッチデータの取得などのオプションを調べて、個々の Db2 インタラクションの数を減らします。頻繁にアクセスされるデータをアプリケーション層の近くに移動する可能性を評価します。

これらのステップに従うことで、リプラットフォーム戦略の実現可能性についてデータ駆動型の意思決定を行い、潜在的なパフォーマンス問題が本番環境に影響を与える前に特定できます。このアプローチは、データベースに依存するアプリケーションの許容可能なパフォーマンスレベルを維持しながら、スムーズな移行を確保するのに役立ちます。

セキュリティ

- アプリケーションビルドを保護する: Virtual Private Cloud (VPC) のプライベートサブネットを使用して を実行し AWS CodeBuild 、分離とセキュリティの強化を確実にします。ビルドプロセス中に安全なデータベースアクセスのために、CodeBuild サブネット CIDR から Db2 の信頼されたコンテキストを実装します。
- ランタイム環境を保護する: ランタイムサブネット CIDR の Db2 信頼されたコンテキストを使用して、データベース接続を保護します。
- データベース認証情報を安全に管理する： 認証情報の定期的なローテーションスケジュールを実装して、不正アクセスのリスクを最小限に抑えます。Db2 認証情報を安全に保存します AWS Secrets Manager。
- ネットワークセキュリティを確立する: 強力なネットワークセグメンテーションとファイアウォールルールを実装して、ビルド環境とランタイム環境の両方を保護します。 AWS Direct Connect との適切な組み合わせ AWS Site-to-Site VPN を使用して、必要なレベルのセキュリティを実現します。
- 暗号化を強制する: アプリケーションと Db2 for z/OS 間で転送中のデータの暗号化を強制します。

アプリケーションガバナンス

- 信頼できるソースを確立する: GitHub などの新しいソフトウェア設定管理 (SCM) を、移行されたアプリケーションコードの信頼できる単一のソースとして確立します。これにより、整合性が確保され、移行期間中のクラウド環境とメインフレーム環境間のバージョン不一致がなくなります。
- 変更管理プロセスを更新する: 変更管理プロセスを更新して、この新しいデュアル環境パラダイムのコード変更を検討します。このプロセスには以下が含まれます。
  - コード変更の承認ワークフローをクリアします。
  - メインブランチにコードをマージする前に、コードレビューを行う必要があります。
  - 両方の環境が同時に更新を受信できるようにするための同期されたデプロイ手順。
  - どちらの環境でも問題が発生した場合のロールバックメカニズム。

伸縮性

クラウドコンピューティングの伸縮性により、メインフレームのコスト構造とリソース管理を大幅に変更するパラダイムシフトが導入されます。固定容量とピークベースの料金モデルを持つ従来のメインフレーム環境とは異なり、クラウドプラットフォームは動的なスケーラビリティとpay-as-you-goのアプローチを提供し、大幅なコスト削減と運用効率の向上につながる可能性があります。

クラウド環境では、組織は実際の需要に基づいてコンピューティングリソースをリアルタイムでスケールアップまたはスケールダウンできるため、ピーク時の負荷に対応するためにオーバープロビジョニングを行う必要がなくなります。この伸縮性により、企業は使用量の急増に対応するために高価なハードウェアおよびソフトウェアライセンスに投資するのではなく、消費したリソースに対してのみ支払うことができます。

料金の仕組みの詳細については AWS、AWS 「 料金表」を参照してください。

次のステップ

メインフレームのモダナイゼーションは、専門知識と高度なソリューションを必要とする複雑で重要なイニシアチブです。以下のタスクに役立つ戦略的パートナーシップを通じて、モダナイゼーションプロセスを加速し、より迅速なビジネス成果を達成できます。

- 評価と優先順位付け: メインフレームアプリケーションを確認し、データベースを Db2 for z/OS に保持しながら、リプラットフォームに適したアプリケーションを特定します。複雑さ、ビジネス重要度、潜在的な投資収益率 (ROI) などの要因を考慮します。
- 移行戦略の策定: タイムライン、リソース割り当て、リスク軽減戦略など、選択したアプリケーションをリプラットフォームするための詳細な計画を作成します。
- ツールとテクノロジーを評価する: アプリケーションのモダナイゼーションプラットフォームやコード変換ツールなど、リプラットフォームプロセスを容易にするために、適切なツールとテクノロジーを調査して選択します。
- エキスパートと連携する: プロジェクトのリプラットフォームの経験があるメインフレームモダナイゼーションスペシャリストやコンサルティング会社と提携することを検討してください。
- 概念実証: 小規模な概念実証から始めてアプローチを検証し、大きなアプリケーションにスケールアップする前に潜在的な課題を特定します。
- テストと検証: 包括的なテスト戦略を策定して、リプラットフォームされたアプリケーションが正しく機能し、既存の Db2 for z/OS データベースとデータの整合性を維持します。
- トレーニングと知識の移転: リプラットフォームされたアプリケーションと導入された新しいツールやテクノロジーに関するトレーニングを提供することで、新しい環境に向けてチームを準備します。
- 段階的な実装: リプラットフォームへの段階的なアプローチを検討してください。パフォーマンスをモニタリングし、発生した問題に対処しながら、アプリケーションを徐々に移行します。
- 継続的な最適化: リプラットフォーム後、アプリケーションのパフォーマンスと Db2 for z/OS データベースとのやり取りを継続的にモニタリングして最適化し、長期的な成功を確保します。
- ペースでモダナイズする: ワークロードが実行され AWS 、すでにクラウドを活用できたので、モダナイゼーションの再考フェーズの計画を開始します。

リソース

メインフレームの移行とモダナイゼーションの詳細については、以下のリソースを参照してください。

AWS ドキュメント

- DNS サービスとしての Amazon Route 53 の設定
- ELB ロードバランサーへのトラフィックのルーティング
- 加重ルーティング
- Rocket Software を使用したアプリケーションのリプラットフォーム

Rocket Software リファレンス

- Micro Focus 外部通話インターフェイス (ECI)
- CICS ウェブサービス

IBM リファレンス

- 信頼できるコンテキスト (IBM Db2 for z/OS ドキュメント)

ツール

- Rocket Enterprise Server

AWS 規範ガイダンスのパターンとガイド

- AWS Mainframe Modernization と を使用して COBOL Db2 プログラムを構築する AWS CodeBuild
- の DevOps AWS Mainframe Modernization
- メインフレームのモダナイゼーション: アプリケーションコードを移行するためのデカップリングパターン

- 信頼されたコンテキストを使用して、 上の Db2 フェデレーションデータベース AWS 内のユーザーアクセスを保護および合理化する

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 5 月 7 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

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。

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、組織がクラウド導入を成功させるための準備を支援します。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。

クライアント側の暗号化

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

Cloud Center of Excellence (CCoE)

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

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

組織が、 AWS クラウドへの移行時に通常実行する 4 つの段階。

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、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。AWS 移行戦略との関連性については、「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。信頼できる ID のみが、期待されるネットワークから信頼できるリソースにアクセス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、リソースの保護に役立つように、 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 、ガバナンス要件のコンプライアンスに影響を与える可能性のあるランディングゾーンの変更を検出したりできます。

DVSM

「開発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。例えば、 AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。 AWS 移行戦略のエピックの詳細については、プログラム実装ガイドを参照してください。

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

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

F

ファクトテーブル

スタースキーマの中央にあるテーブル。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。

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、、 AWS Security Hub CSPM、Amazon GuardDuty、、 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。

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。

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。

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 から分離され、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、ユーザーは組織内のすべて

のユーザーを IAM で作成しなくても、 AWS マネジメントコンソール 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

送信先で、それ AWS のサービス を受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

のエントリポイントの 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、 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

「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。

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「Using AWS Organizations with other AWS services 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。詳細については、深層学習システムにおける不確実性の定量化ガイドを参照してください。

未分化なタスク

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

上位環境

「環境」を参照してください。

V

バキューミング

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

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

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

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

現在のレコードに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。