AWS マルチリージョンの基本

AWS 規範ガイダンス

AWS 規範ガイダンス: AWS マルチリージョンの基本

Table of Contents

AWS マルチリージョンの基本

John Formento、Amazon Web Services (AWS)

2025 年 9 月 (ドキュメント履歴)

この高度な 300 レベルのガイドは、 でワークロードを構築し、マルチリージョンアーキテクチャを使用してワークロードの耐障害性を向上させることに関心のあるクラウドアーキテクト AWS やシニアリーダーを対象としています。このガイドでは、インフラストラクチャとサービスに関する AWS 基本的な知識を前提としています。マルチリージョンの一般的なユースケースについて概説し、設計、開発、デプロイに関するマルチリージョンの基本的な概念と影響について共有し、マルチリージョンアーキテクチャがワークロードに適しているかどうかを判断するのに役立つ規範的ガイダンスを提供します。

このガイドの内容

Well-Architected の実現状況の確認

AWS Well-Architected フレームワークは、クラウドでシステムを構築する際の意思決定の長所と短所を理解するのに役立ちます。フレームワークの 6 つの柱は、信頼性、安全性、効率性、コスト効率、持続可能なシステムを設計および運用するためのアーキテクチャのベストプラクティスを提供します。で無料で利用できる を使用してAWS Well-Architected Tool、各柱の一連の質問に答えることでAWS マネジメントコンソール、これらのベストプラクティスに照らしてワークロードを確認できます。

リファレンスアーキテクチャのデプロイ、図、技術ガイドなど、クラウドアーキテクチャに関する追加のエキスパートガイダンスとベストプラクティスについては、AWS 「アーキテクチャセンター」を参照してください。

序章

各 AWS リージョン は、1 つの地理的エリア内における、複数の独立した、物理的にも分離されたアベイラビリティーゾーンで構成されています。各リージョンのソフトウェアサービス間の論理的な分離は、厳密に管理されています。この目的に沿った設計により、あるリージョンでインフラストラクチャやサービスに障害が発生しても、別のリージョンで障害が相関することはありません。

ほとんどの AWS ユーザーは、複数のアベイラビリティーゾーンまたはリージョンを使用することで、1 つのリージョンのワークロードの耐障害性目標を達成できます AWS のサービス。ただし、ユーザーのサブセットは、次の 3 つの理由でマルチリージョンアーキテクチャを追求します。

- 最上位のワークロードに対して高可用性と運用継続性の要件があり、単一リージョンのリソースに影響を与える障害からの制限された復旧時間を確立したいと考えています。
- 特定の管轄区域内でワークロードを運用する必要があるデータ主権要件 (現地の法律、規制、コンプライアンスの遵守など) を満たす必要があります。
- エンドユーザーに最も近い場所でワークロードを実行することで、ワークロードのパフォーマンスとカスタマーエクスペリエンスを向上させる必要があります。

このガイドでは、高可用性と運用継続性の要件に焦点を当て、ワークロードにマルチリージョンアーキテクチャを採用する際の考慮事項について説明します。マルチリージョンワークロードの設計、開発、デプロイに適用される基本的な概念について説明し、マルチリージョンアーキテクチャが特定のワークロードに適しているかどうかを判断するのに役立つ規範的なフレームワークを提供します。これらのアーキテクチャは困難であり、マルチリージョンアーキテクチャが正しく構築されていない場合、ワークロードの全体的な可用性が低下する可能性があるため、マルチリージョンアーキテクチャがワークロードに適した選択であることを確認する必要があります。

単一リージョンでの耐障害性を実現するエンジニアリングとオペレーション

マルチリージョンの概念を検討する前に、まずワークロードが 1 つのリージョンで可能な限り回復力があることを確認します。これを実現するには、 AWS Well-Architected フレームワークの信頼性の柱と運用上の優秀性の柱に照らしてワークロードを評価し、トレードオフとリスク評価に基づいて必要な変更を加えます。Well-Architected AWS フレームワークでは、以下の概念について説明します。

- ドメイン境界に基づくワークロードのセグメント化
- 明確に定義されたサービス契約
- 依存関係の管理と結合
- 障害、再試行、バックオフ戦略の処理
- べき等操作とステートフルトランザクションとステートレストランザクション
- オペレーショナルレディネスと変更管理
- ワークロードの状態の理解
- イベントへの対応

単一リージョンのレジリエンスをさらに高めるには、ホワイトペーパー「Advanced Multi-AZ Resilience Patterns: Detecting and Mitigating Gray Failures」で説明されている概念を確認して適用します。このホワイトペーパーでは、各アベイラビリティーゾーンでレプリカを使用して障害を封じ込め、 AWS Well Architected フレームワークで導入されたマルチ AZ 概念を拡張するためのベストプラクティスについて説明します。マルチリージョンアーキテクチャは、アベイラビリティーゾーンにバインドされる障害モードを軽減できますが、考慮すべきマルチリージョンアプローチに伴うトレードオフがあります。そのため、マルチ AZ アプローチから始めて、マルチリージョンアーキテクチャの基本と照らし合わせて特定のワークロードを評価し、マルチリージョンアプローチがワークロードの耐障害性を高めることができるかどうかを判断することをお勧めします。

マルチリージョンの基本 1: 要件の理解

前述のように、高可用性とオペレーションの継続性が、マルチリージョンアーキテクチャを追求する一般的な理由です。可用性メトリクスは、定義された期間にワークロードを使用できる時間の割合を測定しますが、オペレーションの継続性メトリクスは、大規模で通常はより長い期間イベントの復旧時間を測定します。

可用性の測定はほぼ継続的なプロセスです。特定の測定値は異なる場合がありますが、通常はターゲット可用性メトリクスを中心に結合されます。ほとんどの場合、9 と呼ばれています (99.99% の可用性など) 。可用性の目標は、常に同じではありません。すべてのワークロードに単一の目標を適用するのではなく、ワークロードレベルで可用性目標を設定し、重要でないコンポーネントと重要なコンポーネントを分離する必要があります。

オペレーションを継続するためには、通常、以下のポイントインタイム測定が使用されます。

- 目標復旧時間 (RTO) – RTO は、サービスの中断とサービスの復旧の間の最大許容遅延です。この値によって、サービスが停止される許容時間が決まります。
- 目標復旧時点 (RPO) – RPO は、最後のデータ復旧時点からの最大許容時間です。これにより、最新の復旧時点からサービス中断までの間のデータ損失がどの程度まで許容されるかが決まります。

可用性目標の設定と同様に、RTO と RPO もワークロードレベルで定義する必要があります。より積極的な運用の継続性または高可用性を実現するには、投資を増やす必要があります。とはいえ、すべてのアプリケーションが同じレベルの耐障害性を要求したり、必要とするわけではありません。ビジネスオーナーと IT オーナーを調整して、ビジネスへの影響に基づいてアプリケーションの重要度を評価し、それに応じて階層化することで、出発点を提供することができます。次の表は、階層化の例を示しています。

この表は、サービスレベルアグリーメント (SLAs。

耐障害性階層	アベイラビリティー SLA	許容ダウンタイム/年
プラチナ	99.99%	52.60 分
ゴールド	99.90%	8.77 時間
シルバー	99.5%	1.83 日間

次の表は、RTO と RPO の耐障害性階層化の例を示しています。

耐障害性階層	最大 RTO	最大 RPO	条件	Cost
プラチナ	15 分	5 分	ミッションクリティカルなワークロード	$$$
ゴールド	15 分 ~ 6 時間	2 時間	重要だがミッションクリティカルではないワークロード	$$
シルバー	6 時間 – 数日	24 時間	重要ではないワークロード	$

回復力のあるワークロードを設計する場合は、高可用性と運用の継続性の関係を考慮してください。たとえば、ワークロードが 99.99% の可用性を必要とする場合、1 年間に許容されるダウンタイムは 53 分以下です。障害を検出するには少なくとも 5 分、オペレーターがエンゲージして復旧手順を決定し、これらの手順を実行するにはさらに 10 分かかる場合があります。1 つの問題から復旧するまでに 30 ~ 45 分かかることは珍しくありません。この場合、相関する影響を排除する独立したインスタンスを提供するマルチリージョン戦略を持つことが有益です。これにより、初期障害を個別にトリアージしながら、制限された時間内にフェイルオーバーすることで、オペレーションを継続できます。ここでは、適切な制限された復旧時間を定義し、調整する必要があります。

マルチリージョンアプローチは、極端な可用性のニーズ (99.99% 以上の可用性など) や、別のリージョンにフェイルオーバーすることによってのみ満たすことができる厳格な運用継続性要件を持つミッションクリティカルなワークロードに適しています。ただし、これらの要件は通常、数分または数時間で測定される制限された復旧時間を持つエンタープライズのワークロードポートフォリオの小さなサブセットにのみ適用されます。アプリケーションが数分または数時間の復旧時間を必要とする場合を除き、影響を受けるリージョン内でアプリケーションのリージョンの中断が修正されるのを待つことをお勧めします。このアプローチは通常、低層ワークロードと一致します。

マルチリージョンアーキテクチャを実装する前に、ビジネス意思決定者と技術チームは、運用コストやインフラストラクチャのコストドライバーなどのコストへの影響について調整する必要があります。一般的なマルチリージョンアーキテクチャでは、単一リージョンアプローチの 2 倍のコストが発生する可能性があります。ホットスタンバイ、ウォームスタンバイ、パイロットライトを使用した

実行など、ビジネス継続性には複数のマルチリージョンパターンがありますが、復旧目標を達成するリスクが最も低いパターンにはホットスタンバイの実行が含まれ、ワークロードのコストが 2 倍になります。

主なガイダンス

- RTO や RPO などのオペレーションの可用性と継続性の目標は、ワークロードごとに設定し、ビジネスと IT の利害関係者と調整する必要があります。
- 可用性とオペレーションの継続性に関する目標は、ほとんどの場合 1 つのリージョン内で達成できます。1 つのリージョン内で達成できない目標については、コスト、複雑さ、利点のトレードオフを明確に把握したマルチリージョンを検討してください。

マルチリージョンの基本 2: データについて

マルチリージョンアーキテクチャを採用する場合、データの管理は簡単な問題ではありません。リージョン間の地理的距離により、リージョン間でデータをレプリケートするのにかかる時間として明らかになる不可避のレイテンシーが課されます。可用性、データ整合性、およびマルチリージョンアーキテクチャを使用するワークロードへのレイテンシーの増加のトレードオフが必要になります。非同期レプリケーションと同期レプリケーションのどちらを使用するかにかかわらず、レプリケーションテクノロジーが課す動作の変更を処理するようにアプリケーションを変更する必要があります。データ整合性とレイテンシーに関する課題により、単一のリージョン用に設計された既存のアプリケーションを取得し、マルチリージョンにすることは非常に困難です。特定のワークロードのデータ整合性要件とデータアクセスパターンを理解することは、トレードオフを比較検討するのに重要です。

2.a: データ整合性要件を理解する

CAP 定理は、データ整合性、可用性、ネットワークパーティション間のトレードオフについて推論するためのリファレンスを提供します。これらの要件のうち、ワークロードで同時に満たすことができるのは 2 つだけです。定義上、マルチリージョンアーキテクチャにはリージョン間のネットワークパーティションが含まれているため、可用性と整合性を選択する必要があります。

リージョン間でデータの可用性を選択した場合、トランザクション書き込みオペレーション中に大きなレイテンシーが発生することはありません。リージョン間でコミットされたデータの非同期レプリケーションに依存すると、レプリケーションが完了するまでリージョン間の整合性が低下するためです。非同期レプリケーションでは、プライマリリージョンで障害が発生した場合、書き込みオペレーションがプライマリリージョンからのレプリケーションを保留中になる可能性が高くなります。これにより、レプリケーションが再開されるまで最新のデータが使用できず、中断が発生したリージョンからレプリケートされなかった処理中のトランザクションを処理するための調整プロセスが必要になるシナリオが発生します。このシナリオでは、ビジネスロジックを理解し、トランザクションを再生したり、リージョン間でデータストアを比較したりするための特定のプロセスを作成する必要があります。

非同期レプリケーションが優先されるワークロードでは、 Amazon Aurora や Amazon DynamoDB などのサービスを非同期クロスリージョンレプリケーションに使用できます。 Amazon Aurora グローバルデータベース と Amazon DynamoDB グローバルテーブル の両方に、レプリケーションラグのモニタリングに役立つデフォルトの Amazon CloudWatch メトリクスがあります。Aurora グローバルデータベースは、データが書き込まれる 1 つの プライマリリージョンと、最大 5 つの読み取り専用 セカンダリ リージョンで構成されます。DynamoDB グローバルテーブルは、データの書き

込みと読み取りを行う任意の数のリージョンにわたるマルチアクティブレプリカテーブルで構成されます。

イベント駆動型アーキテクチャを活用するようにワークロードをエンジニアリングすることは、マルチリージョン戦略の利点です。これは、ワークロードがデータの非同期レプリケーションを受け入れ、イベントを再生することで状態を再構築できることを意味します。ストリーミングおよびメッセージングサービスはメッセージペイロードデータを 1 つのリージョンでバッファするため、リージョンフェイルオーバーまたはフェイルバックプロセスには、クライアント入力データフローをリダイレクトするメカニズムを含める必要があります。このプロセスでは、中断が発生したリージョンに保存されている処理中または未配信のペイロードも調整する必要があります。

CAP 整合性要件を選択し、リージョン間で同期的にレプリケートされたデータベースを使用して、複数のリージョンから同時に実行されるアプリケーションをサポートする場合は、データ損失のリスクを排除し、リージョン間でデータを同期させます。ただし、書き込みは複数のリージョンにコミットする必要があり、リージョンは互いに数百または数千マイルになる可能性があるため、レイテンシー特性は高くなります。アプリケーション設計では、このレイテンシー特性を考慮する必要があります。さらに、同期レプリケーションでは、書き込みを成功させるために複数のリージョンにコミットする必要があるため、相関する障害が発生する可能性があります。1 つのリージョン内に障害がある場合は、書き込みを成功させるためのクォーラムを作成する必要があります。これには、通常、3 つのリージョンにデータベースを設定し、3 つのリージョンのうち 2 つのクォーラムを確立する必要があります。Paxos などの テクノロジーは、データを同期的にレプリケートしてコミットするのに役立ちますが、開発者の多大な投資が必要です。

強固な整合性要件を満たすため、書き込みに複数のリージョン間での同期レプリケーションが含まれる場合、書き込みレイテンシーは桁違いに大きくなります。書き込みレイテンシーを長くすることは、通常、アプリケーションのタイムアウトや再試行戦略を再検討するなど、大きな変更を加えることなくアプリケーションに改良できるものではありません。理想的には、アプリケーションを最初に設計するときに考慮する必要があります。同期レプリケーションが優先されるマルチリージョンワークロードでは、 AWS Partner ソリューションが役立ちます。

2.b: データアクセスパターンを理解する

ワークロードデータアクセスパターンは、読み込み多用または書き込み多用です。特定のワークロードのこの特性を理解することは、適切なマルチリージョンアーキテクチャを選択するのに役立ちます。

完全に読み取り専用の静的コンテンツなどの読み取り負荷の高いワークロードでは、書き込み負荷の高いワークロードと比較して、エンジニアリングの複雑さが少ないアクティブ/アクティブマルチ

リージョンアーキテクチャを実現できます。コンテンツ配信ネットワーク (CDN) を使用してエッジで静的コンテンツを提供すると、エンドユーザーに最も近いコンテンツをキャッシュすることで可用性を確保できます。Amazon CloudFront 内でオリジンフェイルオーバーなどの機能セットを使用すると、これを実現できます。もう 1 つのオプションは、ステートレスコンピューティングを複数のリージョンにデプロイし、DNS を使用してユーザーを最も近いリージョンにルーティングしてコンテンツを読み込ませることです。これを実現するには、位置情報ルーティングポリシーで Amazon Route 53 を使用できます。

読み込みトラフィックの割合が書き込みトラフィックよりも大きい読み込み負荷の高いワークロードでは、 読み込みローカル書き込みグローバル 戦略を使用できます。つまり、すべての書き込みリクエストは特定のリージョンのデータベースに送信され、データは他のすべてのリージョンに非同期的にレプリケートされ、読み取りは任意のリージョンで実行できます。このアプローチでは、書き込みのクロスリージョンレプリケーションのレイテンシーが増加するとローカル読み取りが古くなる可能性があるため、ワークロードは結果整合性を受け入れる必要があります。

Aurora グローバルデータベース は、すべての 読み取りトラフィックをローカルでのみ処理できるスタンバイリージョンでリードレプリカ をプロビジョニングし、書き込みトラフィックを処理するために特定のリージョンに単一のプライマリデータストアをプロビジョニングするのに役立ちます。データはプライマリデータベースからスタンバイデータベース (リードレプリカ) に非同期的にレプリケートされ、オペレーションをスタンバイリージョンにフェイルオーバーする必要がある場合、スタンバイデータベースをプライマリに昇格させることができます。このアプローチでは DynamoDB を使用することもできます。DynamoDB グローバルテーブル は、リージョン間で レプリカテーブルをプロビジョニングできます。リージョンごとにスケーリングして、任意のボリュームのローカル読み取りまたは書き込みトラフィックをサポートできます。アプリケーションが 1 つのリージョンのレプリカテーブルにデータを書き込むと、DynamoDB はその書き込みを他の リージョンの他のレプリカテーブルに自動的に伝搬します。この設定では、データは定義されたプライマリリージョンからスタンバイリージョンのレプリカテーブルに非同期的にレプリケートされます。どのリージョンのレプリカテーブルでも書き込みを受け付けることができるため、スタンバイリージョンのプライマリへの昇格はアプリケーションレベルで管理されます。ここでも、ワークロードは結果整合性を受け入れる必要があります。そのため、最初から設計されていない場合は書き直す必要がある場合があります。

書き込み負荷の高いワークロードの場合、プライマリリージョンを選択し、スタンバイリージョンにフェイルオーバーする機能をワークロードに組み込む必要があります。アクティブ/アクティブアプローチと比較して、プライマリスタンバイアプローチには追加のトレードオフがあります。これは、アクティブ/アクティブアーキテクチャの場合、リージョンへのインテリジェントなルーティングを処理し、セッションアフィニティを確立し、べき等なトランザクションを確保し、潜在的な競合を処理するためにワークロードを書き直す必要があるためです。

回復力のためにマルチリージョンアプローチを使用するほとんどのワークロードは、アクティブ/アクティブアプローチを必要としません。シャーディング戦略を使用すると、クライアントベース全体で障害の影響範囲を制限することで、耐障害性を高めることができます。クライアントベースを効果的にシャードできる場合は、シャードごとに異なるプライマリリージョンを選択できます。例えば、クライアントの半分がリージョン 1 に、半分がリージョン 2 に整列するように、クライアントをシャードできます。リージョンをセルとして扱うことで、マルチリージョンセルアプローチを作成できます。これにより、ワークロードへの影響の範囲が縮小されます。詳細については、このアプローチに関する AWS re:Invent プレゼンテーションを参照してください。

シャーディングアプローチとプライマリスタンバイアプローチを組み合わせて、シャードのフェイルオーバー機能を提供できます。フェイルオーバー後のデータストアのトランザクション整合性を確保するために、テスト済みのフェイルオーバープロセスをワークロードに設計し、データ調整のプロセスも設計する必要があります。これらについては、このガイドの後半で詳しく説明します。

主なガイダンス

- 障害が発生すると、レプリケーション待ちの書き込みがスタンバイリージョンにコミットされない可能性が高くなります。レプリケーションが再開されるまで、データは使用できません (非同期レプリケーションの場合)。
- フェイルオーバーの一環として、非同期レプリケーションを使用するデータストアでトランザクション整合性のある状態を維持するために、データ調整プロセスが必要になります。これには特定のビジネスロジックが必要であり、データストア自体によって処理されるものではありません。
- 強力な整合性が必要な場合は、同期的にレプリケートするデータストアの必要なレイテンシーを許容するようにワークロードを変更する必要があります。

マルチリージョンの基本 3: ワークロードの依存関係を理解する

特定のワークロードには、使用されるリージョン、内部依存関係、サードパーティー依存関係、ネットワーク依存関係、証明書、キー、シークレット、パラメータなど AWS のサービス 、複数の依存関係がある場合があります。障害シナリオ中にワークロードを確実に運用するには、プライマリリージョンとスタンバイリージョンの間に依存関係がない必要があります。それぞれが他のリージョンとは独立して運用できる必要があります。これを実現するには、ワークロード内のすべての依存関係を精査して、各リージョン内で利用可能であることを確認します。これは、プライマリリージョンの障害がスタンバイリージョンに影響を与えないために必要です。さらに、依存関係が低下した状態または完全に利用できない場合のワークロードの動作を理解し、これを適切に処理するソリューションを設計できるようにする必要があります。

3.a: AWS のサービス

マルチリージョンアーキテクチャを設計するときは、 AWS のサービス 使用する 、それらのサービスのマルチリージョン機能、およびマルチリージョンの目標を達成するために設計する必要があるソリューションを理解することが重要です。例えば、Amazon Aurora と Amazon DynamoDB は、スタンバイリージョンにデータを非同期的にレプリケートできます。すべての AWS のサービス 依存関係は、ワークロードを実行するすべてのリージョンで利用できる必要があります。使用するサービスが目的のリージョンで利用できることを確認するには、AWS のサービス リージョン別の リストを確認します。

3.b: 内部依存関係とサードパーティー依存関係

すべてのワークロードの内部依存関係が、それらが運用されているリージョンで使用可能であることを確認します。たとえば、ワークロードが多数のマイクロサービスで構成されている場合は、ビジネス機能を構成するすべてのマイクロサービスを特定し、それらのすべてのマイクロサービスがワークロードが動作する各リージョンにデプロイされていることを確認します。または、使用できなくなったマイクロサービスを正常に処理するための戦略を定義します。

ワークロード内のマイクロサービス間のクロスリージョン呼び出しは推奨されず、リージョン分離を維持する必要があります。これは、クロスリージョンの依存関係を作成すると、相関する障害のリスクが高まり、ワークロードの分離されたリージョン実装の利点が相殺されるためです。オンプレミスの依存関係もワークロードの一部である可能性があるため、プライマリリージョンが変更された

場合、これらの統合の特性がどのように変化する可能性があるかを理解することが重要です。たとえば、スタンバイリージョンがオンプレミス環境から遠くにある場合、レイテンシーが増加すると悪影響が及ぶ可能性があります。

Software as a Service (SaaS) ソリューション、Software Development Kit (SDKs)、およびその他のサードパーティー製品の依存関係を理解し、これらの依存関係が低下または利用できないシナリオを実行できると、さまざまな障害モードでのシステムチェーンの動作や動作をより深く把握できます。これらの依存関係は、 を使用して外部でシークレットを管理するなど、アプリケーションコード内にある場合やAWS Secrets Manager、フェデレーションログインAWS IAM アイデンティティセンターで に依存するサードパーティーのボールトソリューション (HashiCorp など) や認証システムが含まれる場合があります。

依存関係に関して冗長性があると、耐障害性を高めることができます。SaaS ソリューションまたはサードパーティーの依存関係 AWS リージョン がワークロードと同じプライマリを使用している場合は、ベンダーと協力して、その耐障害性体制がワークロードの要件と一致するかどうかを判断します。

さらに、サードパーティー製アプリケーションなど、ワークロードとその依存関係との間で共有される運命にも注意してください。フェイルオーバー後にセカンダリリージョンで (またはセカンダリリージョンから) 依存関係が利用できない場合、ワークロードは完全には回復しない可能性があります。

3.c: フェイルオーバーメカニズム

DNS は、トラフィックをプライマリリージョンからスタンバイリージョンにシフトするためのフェイルオーバーメカニズムとして一般的に使用されます。フェイルオーバーメカニズムが取るすべての依存関係を慎重に確認し、精査してください。たとえば、ワークロードが Amazon Route 53 を使用している場合、コントロールプレーンが でホストされていることを理解することは、その特定のリージョンのコントロールプレーンに依存しているus-east-1ことを意味します。プライマリリージョンも単一障害点を作成するus-east-1ため、フェイルオーバーメカニズムの一部として使用することはお勧めしません。別のフェイルオーバーメカニズムを使用する場合は、フェイルオーバーが期待どおりに機能しないシナリオを深く理解し、必要に応じて緊急時に備えるか、新しいメカニズムを開発する必要があります。Amazon Application Recovery Controller (ARC) リージョンスイッチは、フェイルオーバーメカニズムとして使用できるフルマネージド型のマルチリージョンリカバリサービスです。

前のセクションで説明したように、ビジネス機能の一部であるすべてのマイクロサービスは、ワークロードがデプロイされる各リージョンで利用できる必要があります。フェイルオーバー戦略の一

環として、ビジネス機能の一部であるすべてのマイクロサービスがフェイルオーバーして、クロスリージョン呼び出しの可能性を排除する必要があります。または、マイクロサービスが個別にフェイルオーバーした場合、マイクロサービスがクロスリージョン呼び出しを行うなど、望ましくない動作が発生する可能性があります。これによりレイテンシーが発生し、クライアントのタイムアウト中にワークロードが使用できなくなる可能性があります。

3.d: 設定の依存関係

証明書、キー、シークレット、Amazon マシンイメージ (AMIs)、コンテナイメージ、パラメータは、マルチリージョンアーキテクチャの設計に必要な依存関係分析の一部です。可能な限り、各リージョン内でこれらのコンポーネントをローカライズして、これらの依存関係についてリージョン間で運命を共有しないようにすることをお勧めします。たとえば、証明書の有効期限を変更して、期限切れの証明書 (アラームを「事前通知」に設定) が複数のリージョンに影響を与えるシナリオを防ぐ必要があります。

暗号化キーとシークレットもリージョン固有のものでなければなりません。これにより、キーまたはシークレットのローテーションにエラーが発生した場合、その影響は特定のリージョンに限定されます。

最後に、ワークロードが特定のリージョンで取得できるように、すべてのワークロードパラメータをローカルに保存する必要があります。

主なガイダンス

- マルチリージョンアーキテクチャは、リージョン間の物理的および論理的な分離からメリットを得られます。アプリケーションレイヤーでクロスリージョンの依存関係を導入すると、このメリットは損なわれます。このような依存関係は避けてください。
- フェイルオーバー制御は、プライマリリージョンに依存することなく機能する必要があります。
- フェイルオーバーは、クロスリージョン呼び出しのレイテンシーと依存関係が増加する可能性を排除するために、ユーザージャーニー全体で調整する必要があります。

マルチリージョンの基本 4: 運用準備について

マルチリージョンワークロードの運用は、マルチリージョンアーキテクチャに固有の運用上の課題を伴う複雑なタスクです。これには、 AWS アカウント 管理、デプロイプロセスの更新、マルチリージョンオブザーバビリティ戦略の作成、復旧プロセスの作成とテスト、コストの管理が含まれます。運用準備状況レビュー (ORR) は、1 つのリージョンで実行されているか、複数のリージョンで実行されているかにかかわらず、チームが本番稼働用のワークロードを準備するのに役立ちます。

4.a: AWS アカウント 管理

ワークロードを にデプロイするには AWS リージョン、アカウント内のすべてのAWS のサービスクォータにリージョン間でパリティがあることを確認します。まず、アーキテクチャの一部 AWS のサービス であるすべての を特定し、スタンバイリージョンで計画された使用量を確認し、計画された使用量を現在の使用量と比較します。場合によっては、スタンバイリージョンを以前に使用したことがない場合は、デフォルトのサービスクォータを参照して開始点を理解できます。次に、使用するすべてのサービスで、Service Quotas コンソール (ログインが必要) または APIs。

各リージョンで AWS Identity and Access Management (IAM) ロールを設定し、オペレータ、自動化ツール、スタンバイリージョン内のリソースへの AWS のサービス 適切なアクセス許可を付与します。マルチリージョンアーキテクチャのリージョン分離を実現するには、リージョンごとにロールを分離します。スタンバイリージョンを使用する前に、アクセス許可が設定されていることを確認してください。

4.b: デプロイのプラクティス

マルチリージョン機能を使用すると、ワークロードを複数のリージョンにデプロイするのが複雑になる可能性があります。一度に 1 つのリージョンにデプロイする必要があります。例えば、アクティブ/パッシブアプローチを使用する場合は、まずプライマリリージョンにデプロイし、次にスタンバイリージョンにデプロイする必要があります。 は、インフラストラクチャを単一または複数のリージョンにデプロイするAWS CloudFormationのに役立ちます。また、必要に応じて調整できます。 は、パイプラインが存在するリージョンとは異なるリージョンへのデプロイを許可するクロスリージョンアクションを持つ継続的インテグレーション/継続的デリバリー (CI/CD) パイプラインを構築するAWS CodePipelineのに役立ちます。これをブルー/グリーンなどの堅牢なデプロイ戦略と組み合わせると、最小限の場合、ダウンタイムがゼロのデプロイが可能になります。

ただし、アプリケーションまたはデータの状態が永続的ストアに外部化されていない場合、ステートフル機能のデプロイがより複雑になる可能性があります。このような場合は、ニーズに合わせてデプ

ロイプロセスを慎重にカスタマイズしてください。複数のリージョンに同時にデプロイするのではなく、一度に 1 つのリージョンにデプロイするデプロイパイプラインとプロセスを設計します。これにより、リージョン間で相関する障害が発生する可能性が低くなります。Amazon がソフトウェアデプロイを自動化するために使用するテクニックについては、 AWS 「ビルダーズライブラリ」の記事「安全なハンドオフデプロイの自動化」を参照してください。

4.c: オブザーバビリティ

マルチリージョンを設計するときは、各リージョンのすべてのコンポーネントのヘルスをモニタリングして、リージョンのヘルスを包括的に把握する方法を検討してください。これには、レプリケーションラグのメトリクスのモニタリングが含まれる場合がありますが、これは単一リージョンのワークロードに関する考慮事項ではありません。

マルチリージョンアーキテクチャを構築するときは、スタンバイリージョンからのワークロードのパフォーマンスも監視することを検討してください。これには、スタンバイリージョンからヘルスチェックと Canary (合成テスト) を実行して、プライマリリージョンの状態を外部で確認することが含まれます。さらに、Amazon CloudWatch Internet Monitor を使用して、エンドユーザーの観点から外部ネットワークの状態とワークロードのパフォーマンスを把握できます。スタンバイリージョンをモニタリングするには、プライマリリージョンに同じオブザーバビリティを設定する必要があります。

スタンバイリージョンの Canary は、ワークロードの全体的な状態を判断するために、カスタマーエクスペリエンスメトリクスをモニタリングする必要があります。これは、プライマリリージョンに問題がある場合、プライマリのオブザーバビリティが損なわれ、ワークロードの状態を評価する能力に影響する可能性があるため、必要です。

その場合、そのリージョンを外部から観察することでインサイトが得られます。これらのメトリクスは、各リージョンで利用可能なダッシュボードと、各リージョンで作成されるアラームにロールアップする必要があります。CloudWatch はリージョンのサービスであるため、両方のリージョンにアラームがあることが要件です。このモニタリングデータは、呼び出しをプライマリリージョンからスタンバイリージョンにフェイルオーバーするために使用されます。

4.d: プロセスと手順

質問には、「いつフェイルオーバーすべきですか？」と答えるのが最適です。必要になるよりずっと前です。問題が発生する前に、人、プロセス、テクノロジーを含む復旧計画を定義し、定期的にテストします。リカバリの意思決定のフレームワークを決定しておきます。十分に練習された復旧プロセ

スがあり、復旧までの時間が十分に理解されている場合は、RTO ターゲットを満たすフェイルオーバーを使用して復旧プロセスを開始できます。この時点は、プライマリリージョンのアプリケーションの問題が特定された直後か、リージョンのアプリケーション内の復旧オプションが使い果たされたときにさらにイベントになる可能性があります。

フェイルオーバーアクション自体は 100% 自動化する必要がありますが、フェイルオーバーをアクティブ化する決定は、通常は組織内の少数の事前定義された個人が行う必要があります。これらの担当者は、データ損失とイベントに関する情報を考慮する必要があります。また、フェイルオーバーの基準を明確に定義し、組織内でグローバルに理解する必要があります。これらのプロセスを定義して完了するには、Amazon Application Recovery Controller (ARC) リージョンスイッチを使用できます。これにより、end-to-endの自動化が可能になり、テストとフェイルオーバー中に実行されるプロセスの一貫性が確保されます。

リージョン切り替えプランを作成すると、プランがプライマリリージョンとスタンバイリージョンに自動的にレプリケートされ、単一のリージョンに依存しなくなります。この自動化が整ったら、定期的なテスト頻度を定義して実行します。これにより、実際のイベントが発生した場合に、組織が信頼する明確に定義された実践的なプロセスがレスポンスに確実に従います。また、データ調整プロセスに対して確立された許容度を考慮することも重要です。提案されたプロセスが確立された RPO/RTO 要件を満たしていることを確認します。

4.e: テスト

未テストの復旧アプローチを持つことは、復旧アプローチがないことと同じです。テストの基本レベルは、復旧手順を実行してアプリケーションの運用リージョンを切り替えることです。これは、アプリケーションローテーションアプローチと呼ばれることがあります。リージョンを通常の運用体制に切り替える機能を構築することをお勧めしますが、このテストだけでは不十分です。

アプリケーションの復旧アプローチを検証するには、耐障害性テストも重要です。これには、特定の障害シナリオを挿入し、アプリケーションと復旧プロセスがどのように反応するかを理解し、テストが計画どおりに進まなかった場合に必要な緩和策を実装することが含まれます。エラーがない場合に復旧手順をテストしても、障害が発生したときにアプリケーションがどのように動作するかはわかりません。予想される障害シナリオに対して復旧をテストする計画を立てる必要があります。 AWS Fault Injection Serviceには、開始するためのシナリオのリストが増えています。

これは、ビジネス継続性の目標を確実に達成するために厳格なテストが必要な高可用性アプリケーションにとって特に重要です。復旧機能をプロアクティブにテストすることで、本番環境で障害が発生するリスクが軽減され、アプリケーションが望ましい目標復旧時間を達成できるという信頼が構築されます。定期的なテストでは運用上の専門知識も構築されるため、チームは停止が発生したときに

迅速かつ確実に復旧できます。復旧アプローチの人的要素またはプロセスを実行することは、技術的な側面と同様に重要です。

4.f: コストと複雑さ

マルチリージョンアーキテクチャのコストに与える影響は、インフラストラクチャの使用、運用のオーバーヘッド、およびリソース時間の上昇によって駆動されます。前述のように、スタンバイリージョンのインフラストラクチャコストは、事前プロビジョニング時のプライマリリージョンのインフラストラクチャコストと似ているため、総コストが 2 倍になります。日常的なオペレーションには十分ですが、需要の急増を許容するために十分なバッファ容量を確保できるように容量をプロビジョニングします。次に、各リージョンで同じ制限を設定します。

さらに、アクティブ/アクティブアーキテクチャを採用する場合は、マルチリージョンアーキテクチャでアプリケーションを正常に実行するために、アプリケーションレベルの変更が必要になる場合があります。これらの変更は、設計と運用に時間がかかり、リソースを大量に消費する可能性があります。少なくとも、組織は各リージョンの技術とビジネスの依存関係を理解し、フェイルオーバーとフェイルバックプロセスを設計するために時間を費やす必要があります。

また、チームは通常のフェイルオーバーとフェイルバックの演習を行って、イベント中に使用されるランブックに慣れる必要があります。これらの演習は、マルチリージョン投資から期待される成果を得るために不可欠ですが、機会コストを表し、他のアクティビティから時間とリソースを奪います。

4.g: 組織のマルチリージョンフェイルオーバー戦略

AWS リージョン は、相関障害を防ぎ AWS のサービス 、障害発生時の影響を 1 つのリージョンに含む障害分離境界を提供します。これらの障害境界を使用して、各リージョンで独立した障害分離レプリカで構成されるマルチリージョンアプリケーションを構築し、共有された運命シナリオを制限できます。これにより、マルチリージョンアプリケーションを構築し、バックアップと復元からパイロットライト、アクティブ/アクティブまで、さまざまなアプローチを使用してマルチリージョンアーキテクチャを実装できます。ただし、アプリケーションは通常単独では動作しないため、フェイルオーバー戦略の一環として使用するコンポーネントとその依存関係の両方を検討してください。一般的に、複数のアプリケーションが連携してユーザーストーリーをサポートします。これは、ソーシャルメディアアプリに写真や字幕を投稿したり、e コマースサイトでチェックアウトしたりするなど、エンドユーザーに提供される特定の機能です。このため、アプローチを成功させるために必要な調整と一貫性を提供する組織のマルチリージョンフェイルオーバー戦略を開発する必要があります。

マルチリージョンアプローチの指針として、組織が選択できる 4 つの大まかな戦略があります。これらは、最も詳細なアプローチから最も広範なアプローチまで一覧表示されます。

- コンポーネントレベルのフェイルオーバー
- 個々のアプリケーションのフェイルオーバー
- 依存関係グラフのフェイルオーバー
- アプリケーションポートフォリオ全体のフェイルオーバー

各戦略にはトレードオフがあり、フェイルオーバーの意思決定の柔軟性、フェイルオーバーの組み合わせをテストする能力、モーダル動作の存在、計画と実装への組織投資など、さまざまな課題に対処します。各戦略の詳細については、 AWS ブログ記事「組織のマルチリージョンフェイルオーバー戦略の作成」を参照してください。

主なガイダンス

- すべての AWS のサービス クォータを確認して、ワークロードが動作するすべてのリージョンで同等であることを確認します。
- デプロイプロセスは、複数のリージョンを同時に含めるのではなく、一度に 1 つのリージョンをターゲットにする必要があります。
- レプリケーションラグなどの追加のメトリクスは、マルチリージョンシナリオに固有であり、モニタリングする必要があります。
- プライマリリージョンを超えてワークロードのモニタリングを拡張します。各リージョンのカスタマーエクスペリエンスメトリクスをモニタリングし、ワークロードが実行されている各リージョンの外部からこのデータを測定します。
- フェイルオーバーとフェイルバックを定期的にテストします。フェイルオーバープロセスとフェイルバックプロセス用に 1 つのランブックを実装し、テストイベントとライブイベントの両方に使用します。テストイベントとライブイベントのランブックは、異なるものであってはなりません。
- フェイルオーバー戦略のトレードオフを理解します。依存関係グラフまたはアプリケーションポートフォリオ戦略全体を実装します。

結論とリソース

このガイドでは、マルチリージョンアーキテクチャの一般的なユースケース、これらのアーキテクチャの実装の基礎、このアプローチの影響について説明しました。これらの基本を任意のワークロードに適用し、その情報をフレームワークとして使用して、マルチリージョンアーキテクチャがビジネスに適したアプローチかどうかを判断するのに役立ちます。

詳細については、以下のリソースを参照してください。

- AWS アーキテクチャセンター
- AWS Well-Architected フレームワーク
- AWS Well-Architected Tool
- 組織のマルチリージョンフェイルオーバー戦略の作成 (AWS ブログ記事)
- AWS マルチリージョン機能 (AWS re:Post 記事)

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
ARC リージョンスイッチの更新	セクション 3.c および 4.d で、復旧タスクを処理するための Amazon Application Recovery Controller (ARC) リージョンスイッチに関する情報を追加しました。	2025 年 9 月 29 日
更新	ガイド全体の更新。	2024 年 12 月 27 日
初版発行	—	2022 年 12 月 20 日

AWS 規範ガイダンスの用語集

以下は、 AWS 規範ガイダンスが提供する戦略、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まとめています。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。

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 に採用するときは 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。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、、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。

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パスワードやユーザー認証情報などの AWS Secrets Manager機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値には、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の、それ AWS のサービス を受け取る による暗号化。

サービスコントロールポリシー (SCP)

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

サービスエンドポイント

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

サービスレベルアグリーメント (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、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。

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「 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。