Amazon DynamoDB グローバルテーブルを使用する

AWS 規範ガイダンス

AWS 規範ガイダンス: Amazon DynamoDB グローバルテーブルを使用する

Table of Contents

序章 1
概要 2
重要な事実 2
ユースケース 3
書き込みモード 5
任意のリージョンへの書き込みモード (プライマリなし) 5
1 つのリージョンへの書き込みモード (単一プライマリ) 7
自分のリージョンへの書き込みモード (混合プライマリ) 9
ルーティング戦略 12
クライアント主導のリクエストルーティング 13
コンピューティングレイヤーのリクエストルーティング 14
Route 53 のリクエストルーティング 16
Global Accelerator のリクエストルーティング 17
避難プロセス 19
ライブリージョンからの退避 19
オフラインリージョンからの退避 19
スループット・パシティ・プランニング 22
準備チェックリスト 24
よくある質問 26
グローバルテーブルの料金はいくらですか? 26
グローバルテーブルはどのリージョンでサポートされていますか? 26
GSI はグローバルテーブルでどのように処理されますか? 26
グローバルテーブルのレプリケーションを停止するにはどうしたらいいですか? 27
Amazon DynamoDB Streams はグローバルテーブルとどのようにやり取りするのですか? 27
グローバルテーブルはトランザクションをどのように処理するのですか? 27
グローバルテーブルは DynamoDB Accelerator (DAX) キャッシュ とどのようにやり取りするのですか? 28
テーブルのタグは伝播されますか? 28
すべてのリージョンのテーブルをバックアップすべきですか、それとも 1 つのリージョンだけでよいですか? 28
AWS CloudFormation を使用してグローバルテーブルをデプロイするにはどうすればいいですか? 28
結論とリソース 30
ドキュメント履歴 31

用語集 32
..... 32
A 33
B 36
C 38
D 41
E 45
F 47
G 48
H 49
I 50
L 52
M 53
O 57
P 60
Q 62
R 63
S 65
T 69
U 70
V 71
W 71
Z 72
..... lxxiv

Amazon DynamoDB グローバルテーブルを使用する

Amazon Web Services (AWS)、Jason Hunter

2024 年 3 月 (ドキュメント履歴)

グローバルテーブルは、グローバルな Amazon DynamoDB フットプリントに基づいて、フルマネージド、マルチリージョン、マルチアクティブのデータベースを提供し、大規模にスケールされたグローバルアプリケーションの高速なローカルの読み取り/書き込みパフォーマンスを実現します。グローバルテーブルは、選択した 全体で DynamoDB テーブルを自動的にレプリケートします AWS リージョン。グローバルテーブルは既存の DynamoDB API を使用するため、アプリケーションを変更する必要はありません。グローバルテーブルの使用に伴う前払い料金やコミットメントはなく、使用したリソース分のみの支払いで済みます。

このガイドでは、DynamoDB グローバルテーブルを効果的に使用する方法について説明します。ここでは、グローバルテーブルに関する重要な情報、機能の主なユースケース、考慮すべき 3 種類の書き込みモデルの分類基準、実装可能な 4 つの主なリクエストルーティング、稼働しているリージョンまたはオフラインのリージョンを退避させる方法、スループットキャパシティプランニングに関する考え方、グローバルテーブルをデプロイする際の考慮事項のチェックリストを見ていきます。

このガイドは、 AWS 「マルチリージョンAWS の基礎」ホワイトペーパーと「ビデオによるデータの耐障害性設計パターン」で説明されているように、マルチリージョンデプロイのより大きなコンテキストに対応しています。 AWS

目次

- 概要
- 書き込みモード
- ルーティング戦略
- 退避プロセス
- スループットキャパシティプランニング
- 準備チェックリスト
- よくある質問
- 結論とリソース

グローバルテーブルの概要

重要な事実

- グローバルテーブルには、バージョン 2017.11.29 (レガシー) (v1 と呼ばれることもあります) とバージョン 2019.11.21 (最新) (v2 と呼ばれることもあります) の 2 つのバージョンがあります。このガイドは、現在のバージョンにのみ焦点を当てています。
- DynamoDB (グローバルテーブルなし) はリージョンサービスです。つまり、可用性が高く、アベイラビリティーゾーン全体の障害を含むインフラストラクチャの障害に対して本質的に回復力があります。単一リージョンのDynamoDBテーブルは、99.99% の可用性を実現するように設計されています。詳細については、DynamoDB サービスレベル契約 (SLA) を参照してください。
- DynamoDB グローバルテーブルは、2 つ以上のリージョン間でデータを複製します。マルチリージョン DynamoDB テーブルは 99.999% の可用性を保証します。適切な計画を立てれば、グローバルテーブルは地域の障害に強いアーキテクチャを構築するのに役立ちます。
- グローバルテーブルは、アクティブ-アクティブレプリケーションモデルを採用しています。DynamoDB の観点から見ると、各リージョンのテーブルは読み取りと書き込みのリクエストを同じように受け入れることができます。書き込み要求を受け取ると、ローカルレプリカテーブルはバックグラウンドで他の参加リモートリージョンに書き込み操作を複製します。
- 項目は個別にレプリケートされます。1 つのトランザクション内で更新されたアイテムは、まとめてレプリケートされない場合があります。
- ソースリージョンの各テーブルパーティションは、他のすべてのパーティションとparallel 書き込み操作を複製します。リモートリージョン内の書き込み操作の順序は、ソースリージョン内で発生した書き込み操作の順序と一致しない場合があります。テーブルパーティションの詳細については、ブログ記事「DynamoDB のスケーリング: パーティション、ホットキー、ヒートに応じた分割がパフォーマンスに与える影響」を参照してください。
- 新しく書き込まれた項目は、通常、1 秒以内にすべてのレプリカテーブルに伝播されます。近くのリージョンにはより速く伝播される傾向があります。
- Amazon CloudWatch `ReplicationLatency` は各リージョンのペアについてメトリックスを提供しています。到着したアイテムを見て、その到着時間を最初の書き込み時間と比較し、平均を計算することによって計算されます。CloudWatchタイミングはソースリージョン内に保存されます。平均タイミングと最大タイミングを表示すると、レプリケーションラグの平均と最悪のケースを判断するのに役立ちます。このレイテンシーには SLA はありません。
- 2 つの異なるリージョンで個々の項目が (`ReplicationLatency`このウィンドウ内で) ほぼ同時に更新され、最初の書き込み操作が複製される前に 2 回目の書き込み操作が行われると、書き込み

が競合する可能性があります。グローバルテーブルは、書き込み操作のタイムスタンプに基づいて、最終書き込みが優先されるメカニズムを使用することで、このようなコンフリクトを解決します。1 番目の操作は 2 番目の操作に「負け」ます。CloudWatchこれらのコンフリクトはまたはには記録されませんAWS CloudTrail。

- 各項目には、最終書き込みタイムスタンプがプライベートシステムプロパティとして保持されます。ラスト・ライター・ウィンズ・アプローチは、入力するアイテムのタイムスタンプが既存のアイテムのタイムスタンプよりも大きいことを要求する条件付き書き込み操作を使用して実装されます。
- グローバルテーブルは、すべての項目をすべての参加リージョンに複製します。異なるレプリケーションスコープを使用する場合は、複数のグローバルテーブルを作成し、各テーブルに異なる参加リージョンを割り当てることができます。
- ローカルリージョンは、レプリカリージョンがオフラインであったり、`ReplicationLatency`規模が大きくなったりしても、書き込み操作を受け入れます。ローカルテーブルは、各項目が成功するまで、リモートテーブルへの項目のレプリケーションを試行し続けます。
- 万が一、リージョンが完全にオフラインになり、後でオンラインに戻ったときに、保留中のすべてのアウトバウンドレプリケーションとインバウンドレプリケーションが再試行されます。テーブルを同期状態に戻すために特別なアクションは必要ありません。ラスト・ライター・ウィンズ・メカニズムにより、最終的にデータの一貫性が保たれます。
- DynamoDB テーブルには、いつでも新しいリージョンを追加できます。DynamoDB は初期同期と継続的なレプリケーションを処理します。リージョン (元のリージョンも含む) を削除すると、そのリージョンのローカルテーブルも削除されます。
- DynamoDB にはグローバルエンドポイントはありません。すべてのリクエストは、そのリージョンのローカルなグローバルテーブルインスタンスにアクセスするリージョナルエンドポイントに対して行われます。
- DynamoDB への呼び出しはリージョンをまたがってはいけません。ベストプラクティスは、あるリージョンの本拠地であるアプリケーションが、そのリージョンのローカル DynamoDB エンドポイントにのみ直接アクセスすることです。リージョン (DynamoDB レイヤーまたは周囲のスタック) で問題が検出された場合、エンドユーザーのトラフィックは別のリージョンでホストされている別のアプリケーションエンドポイントにルーティングする必要があります。グローバルテーブルにより、すべてのリージョンを拠点とするアプリケーションが同じデータにアクセスできるようになります。

ユースケース

グローバルテーブルには、次のような一般的な利点があります。

- 低レイテンシーの読み取り操作。データのコピーをエンドユーザーの近くに配置して、読み取り操作中のネットワーク遅延を減らすことができます。ReplicationLatencyデータは値と同じくらい新鮮に保たれます。
- 低レイテンシーの書き込み操作。エンドユーザーは近くのリージョンに書き込むことで、ネットワークの待ち時間を減らし、書き込み操作を完了するまでの時間を短縮できます。競合が発生しないように、書き込みトラフィックは慎重にルーティングする必要があります。ルーティングのテクニックについては、後のセクションで説明します。
- 回復力とディザスタリカバリの向上。リージョンのパフォーマンスが低下したり、完全に停止したりした場合は、そのリージョンを退避させ（そのリージョンへのリクエストの一部または全部を移動）、秒単位で測定されたリカバリポイント目標（RPO）と目標復旧時間（RTO）を達成できます。グローバルテーブルを使用すると、DynamoDB SLA の月間稼働率が 99.99% から 99.999% に上がります。
- シームレスなリージョンの移行。新しいリージョンを追加してから古いリージョンを削除して、データレイヤーでダウンタイムを発生させることなく、あるリージョンから別のリージョンにデプロイを移行できます。

たとえば、フィデリティ・インベストメンツは re: Invent 2022 で、注文管理システムに DynamoDB グローバルテーブルをどのように使用しているかについて発表しました。彼らの目標は、アベイラビリティーゾーンとリージョンの障害に対する回復力を維持しながら、オンプレミス処理では達成できない規模で確実に低レイテンシーの処理を実現することでした。

グローバルテーブルの書き込みモード

グローバルテーブルは、テーブルレベルでは常にアクティブ/アクティブです。ただし、書き込みリクエストのルーティング方法を制御して、アクティブ/パッシブとして扱うこともできます。例えば、書き込みが競合しないように、書き込みリクエストを単一のリージョンにルーティングすることを決定する場合があります。

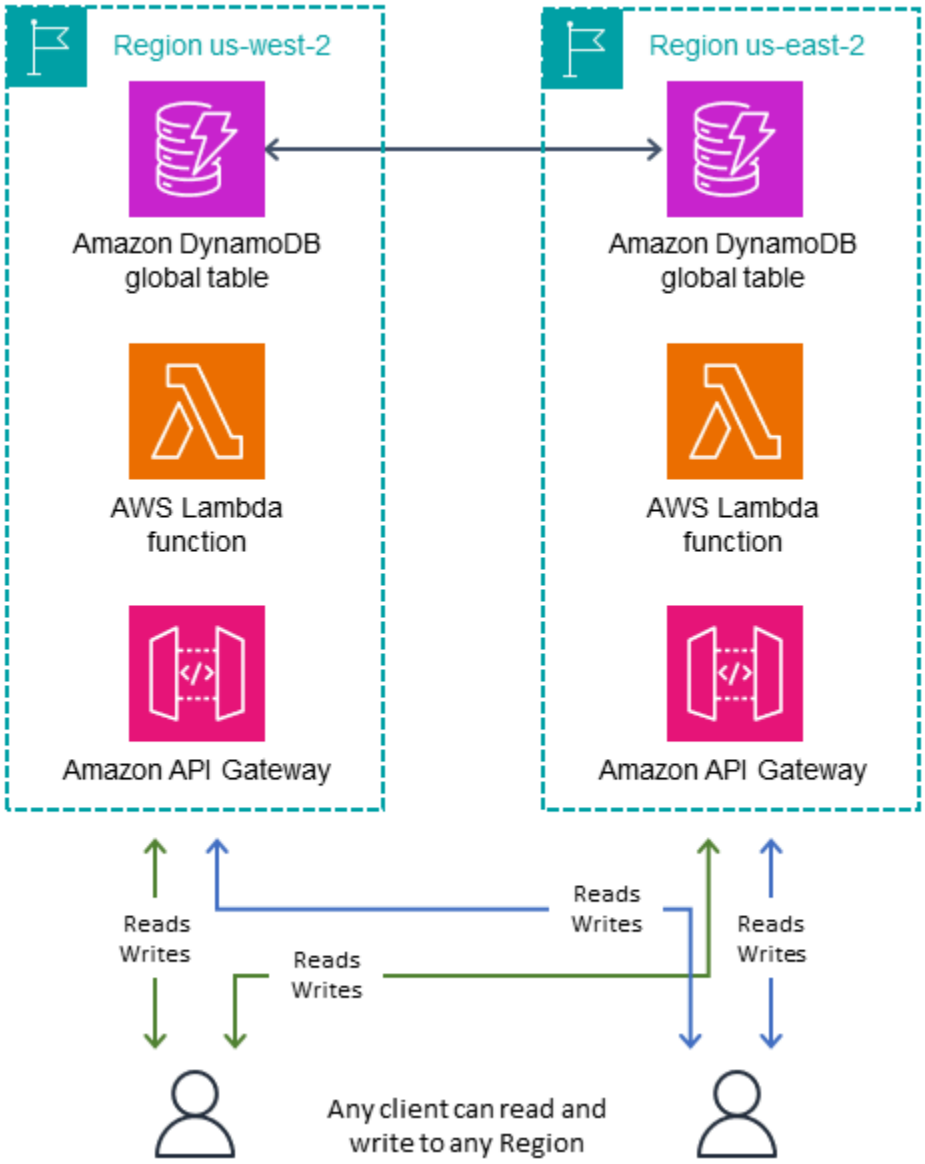
次の 3 つのセクションで説明するように、マネージド書き込みには主に 3 つのパターンがあります。どの書き込みパターンがユースケースに適しているかを検討する必要があります。この選択は、リクエストのルーティング、リージョンの退避、ディザスタリカバリの処理方法に影響します。後のセクションのガイダンスは、アプリケーションの書き込みモードによって異なります。

トピック

- 任意のリージョンへの書き込みモード (プライマリなし)
- 1 つのリージョンへの書き込みモード (単一プライマリ)
- 自分のリージョンへの書き込みモード (混合プライマリ)

任意のリージョンへの書き込みモード (プライマリなし)

どのリージョンへの書き込みモードでも完全にアクティブ/アクティブであり、書き込み操作が行われる場所に制限はありません。どのリージョンでも書き込みリクエストをいつでも受け付けることができます。これは最も単純なモードですが、一部の種類のアプリケーションでのみ使用できます。すべての書き込み操作が不能な場合に適しています。Idempotentとは、ユーザーが連絡先データを更新するときなど、複数のリージョンで同時に実行される書き込み操作や繰り返される書き込み操作が競合しないように、安全に繰り返し実行できることを意味します。また、すべての書き込み操作が決定論的な主キーの下での一意の挿入である追加専用データセットの場合にも適しています。これは、不能であるという特殊なケースです。最後に、このモードは、書き込み操作が競合するリスクが許容できる場合に適しています。



任意のリージョンへの書き込みモードは、実装が最も簡単なアーキテクチャです。いつでも任意のリージョンを書き込み先にすることができるため、ルーティングが容易になります。最近の書き込み操作を任意のセカンダリリージョンに何度でもリプレイできるため、フェイルオーバーが容易になります。可能な限り、この書き込みモード向けの設計を行うようにします。

たとえば、いくつかのビデオストリーミングサービスでは、ブックマーク、レビュー、視聴ステータスフラグなどを追跡するためにグローバルテーブルを使用しています。これらのデプロイでは、すべての書き込み操作が非効率であることが保証されている限り、どのリージョンへの書き込みモードでも使用できます。たとえば、最新のタイムコードの設定、新しいレビューの割り当て、新しいウォッチステータスの設定など、更新のたびにユーザーに新しいステータスが直接割り当てられ、項目の次の正しい値が現在の値に依存しない場合がこれに該当します。万が一、ユーザーの

書き込みリクエストが別のリージョンにルーティングされても、最後の書き込み操作が持続し、最後の割り当てに従ってグローバル状態が落ち着きます。このモードでの読み取り操作は、最終的には、ReplicationLatency最新の値の分だけ遅延して一貫性が保たれます。

別の例として、ある金融サービス会社では、システムの一部としてグローバルテーブルを使用し、各顧客のデビットカード購入の累計を管理して、その顧客のキャッシュバック特典を計算しています。新しいトランザクションは世界中から流入し、複数のリージョンに送られます。この会社は、慎重に再設計することで、どのリージョンへの書き込みモードでも使用できるようになりました。最初のデザインスケッチでは、RunningBalance顧客ごとに1つのアイテムしか管理していませんでした。お客様のアクションにより、(新しい正しい値は現在の値に依存するため)ADD等価ではない式で残高が更新され、異なるリージョンで同じ残高に対してほぼ同時に2つの書き込み操作が行われた場合、残高が同期しなくなります。再設計では、追加のみのワークフローを備えた台帳のように機能するイベントストリーミングを使用しています。顧客のアクションごとに、その顧客用に管理されている項目コレクションに新しい項目が追加されます。(アイテムコレクションとは、主キーを共有していてもソートキーが異なるアイテムのセットです)。各書き込み操作は、カスタマー ID をパーティションキー、トランザクション ID をソートキーとして使用する一意の挿入です。この設計では、項目をプルしてからクライアント側で計算する必要があるため、残量の計算が難しくなりますが、すべての書き込み操作が非効率になり、ルーティングとフェイルオーバーが大幅に簡略化されます。Query(これについては、このガイドの後半で詳しく説明します)。

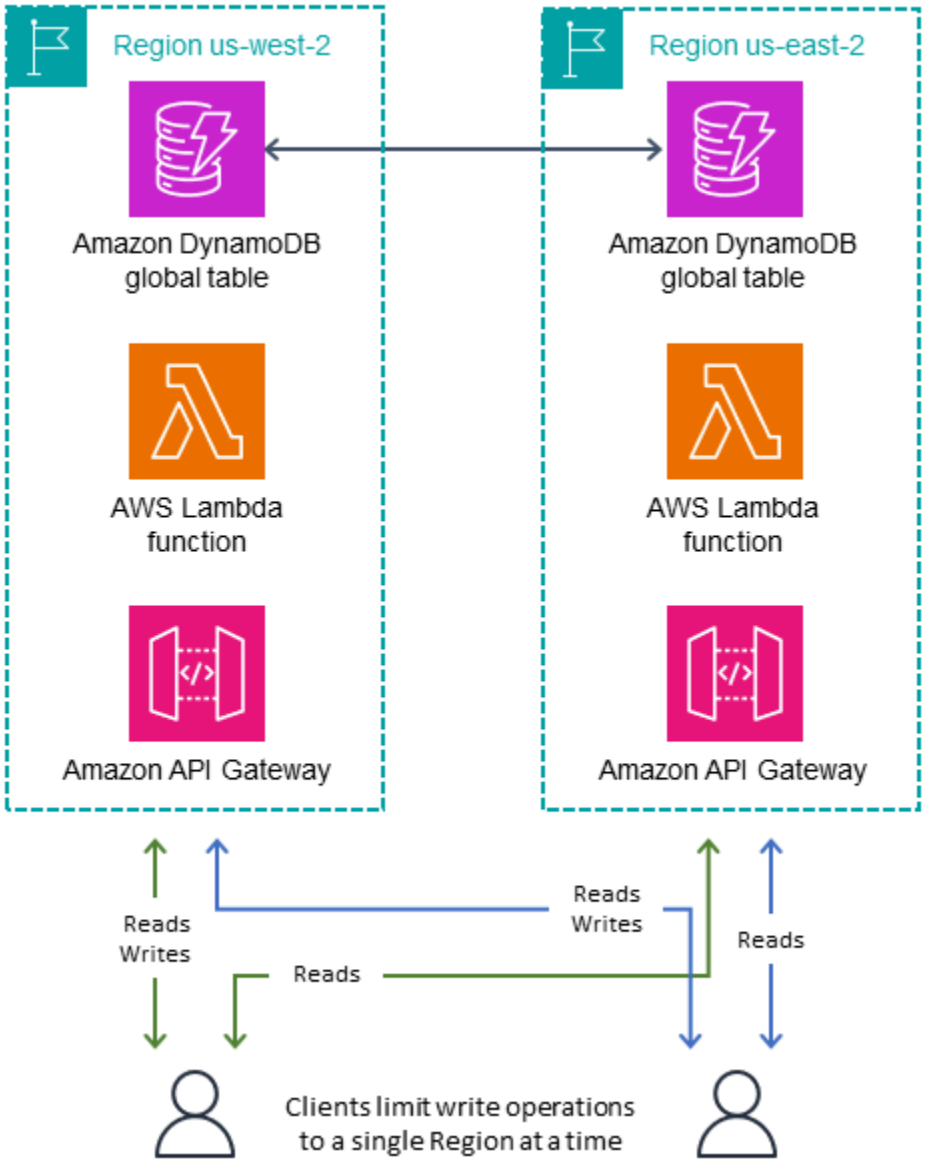
3つ目の例としては、オンライン広告掲載サービスを提供する会社があります。この会社は、どのリージョンモードにも書き込むという設計の簡略化を実現するには、データ損失のリスクを低く抑えても問題ないと判断しました。広告を配信する際、わずか数ミリ秒で十分なメタデータを取得して表示する広告を決定し、広告インプレッションを記録して、同じ広告がすぐに繰り返されないようにします。グローバルテーブルを使用して、世界中のエンドユーザー向けの低レイテンシーの読み取り操作と低レイテンシーの書き込み操作の両方を実現しています。ユーザーのすべての広告インプレッションを1つのアイテムに記録し、リストは増え続けています。アイテムコレクションに追加する代わりに 1 つのアイテムを使用するため、各書き込み操作の一部として古い広告インプレッションを削除できます。削除操作の費用は発生しません。この書き込み操作は同一ではありません。同じエンドユーザーが複数のリージョンからほぼ同時に配信された広告を見た場合、ある広告インプレッションの書き込み操作が別の書き込み操作を上書きする可能性があります。リスクは、ユーザーに広告がたまに繰り返されることです。彼らはこれが許容できると判断しました。

1 つのリージョンへの書き込みモード (単一プライマリ)

1 つのリージョンへの書き込みモードはアクティブ/パッシブで、すべてのテーブル書き込み操作を 1 つのアクティブリージョンにルーティングします。(DynamoDB には 1 つのアクティブリージョンという概念はありません。DynamoDB の外側のレイヤーがこれを管理します)。1 つのリージョンへの

書き込みモードは、書き込み操作が一度に 1 つのリージョンにのみ流れるようにすることで、書き込みの競合を回避します。この書き込みモードは、条件式やトランザクションを使用する場合に役立ちます。これらの式は、最新のデータに対してアクションを実行していることがわかっていない限り使用できないため、すべての書き込みリクエストを最新のデータを含む 1 つのリージョンに送信する必要があります。

最終的には、どのレプリカリージョンでも一貫性のある読み取り操作を行って、レイテンシーを低くすることができます。一貫性のある読み取り操作は、単一のプライマリリージョンに対して行われる必要があります。



後で説明するように、リージョンの障害に対応してアクティブリージョンを変更する必要がある場合があります。一部のユーザーは、follow-the-sunデプロイメントの実装など、現在アクティブなリー

ジョンを定期的に変更します。これにより、アクティブなリージョンが最もアクティビティの多い地域 (通常は昼間、つまり名前) の近くに配置され、読み取り/書き込み操作のレイテンシーが最も低くなります。また、リージョンを変更するコードを毎日呼び出して、障害復旧の前に十分にテストされていることを確認できるという副次的な利点もあります。

パッシブリージョンは、アクティブリージョンになった場合にのみ構築される DynamoDB を取り巻くダウンスケールされたインフラストラクチャを維持する場合があります。このガイドでは、パイロットライトとウォームスタンバイの設計については説明していません。詳細については、ブログ記事「ディザスタリカバリ (DR) アーキテクチャ (パート III : パイロットライトとウォームスタンバイ) 」を参照してください。AWS

グローバルテーブルを使用して低レイテンシーでグローバルに分散された読み取り操作を行う場合は、1 つのリージョンへの書き込みモードを使用すると効果的です。一例として、世界中のすべての地域で同じ参照データを利用できるようにする必要がある大規模なソーシャルメディア企業があります。データを頻繁に更新することはありませんが、更新する場合は書き込みの競合を避けるため、1 つのリージョンのみに書き込みます。読み取り操作は、どのリージョンからでも常に許可されます。

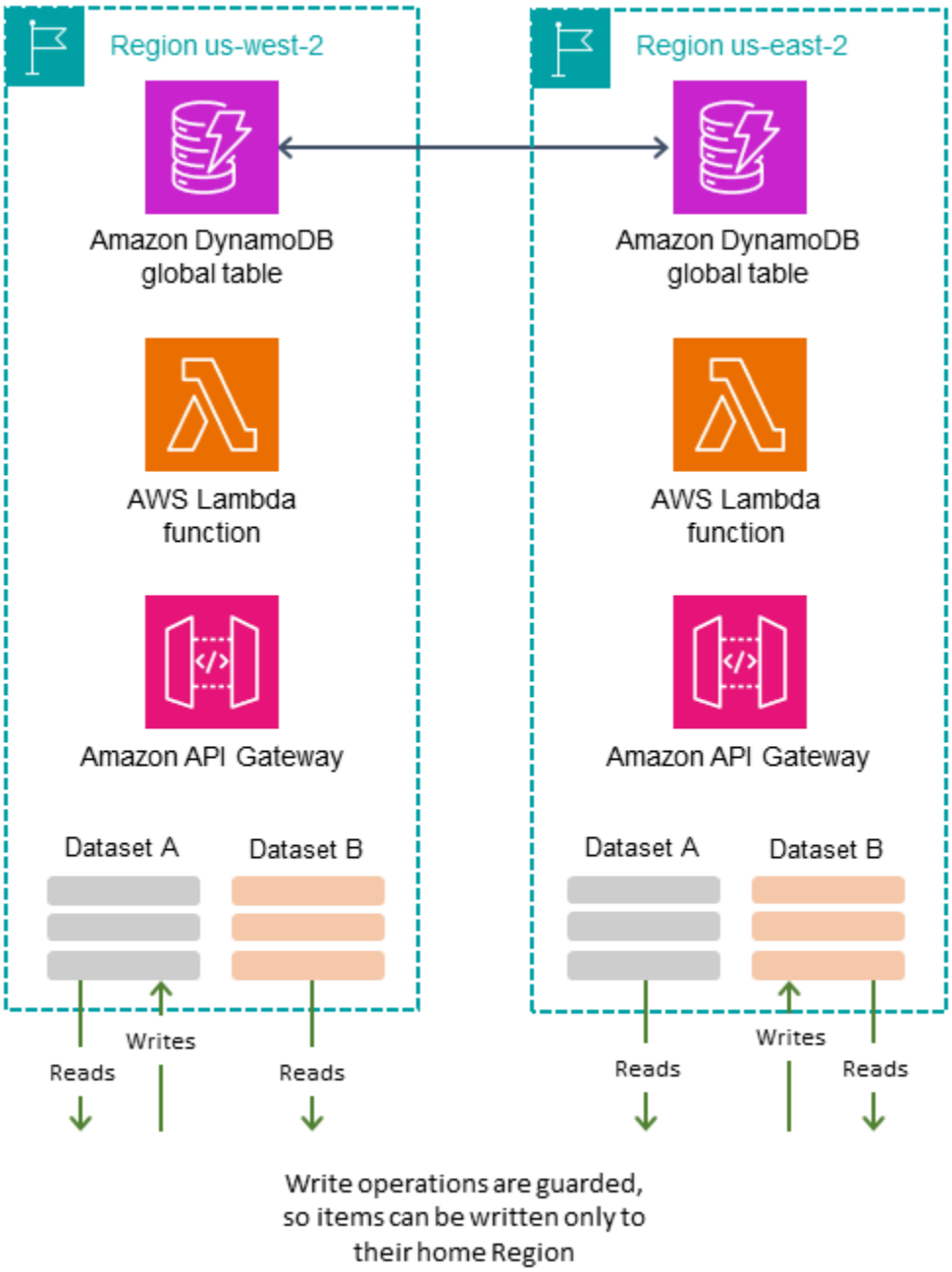
別の例として、先ほど説明した日次キャッシュバック計算を実装した金融サービス会社を考えてみましょう。残高の計算には任意の地域への書き込みモードを使用していましたが、キャッシュバックの支払いを追跡するには1つの地域への書き込みモードを使用していました。10 ドル使うごとに 1 セントの報酬を得たい場合は、前日のすべての取引について支出総額を計算し、キャッシュバックの決定を新しいテーブルに書き込み、クエリした項目セットを削除して消費済みとしてマークし、`Query`翌日の計算に必要な残りを格納する単一の項目に置き換える必要があります。この作業にはトランザクションが必要なので、1 つのリージョンへの書き込みモードの方が効果的です。ワークロードが重複しない限り、アプリケーションは同じテーブルであっても書き込みモードを混在させることができます。

自分のリージョンへの書き込みモード (混合プライマリ)

リージョンへの書き込み書き込みモードでは、さまざまなデータサブセットがさまざまなホームリージョンに割り当てられ、ホームリージョンを介してのみアイテムへの書き込み操作が可能になります。このモードはアクティブ/パッシブですが、アイテムに基づいてアクティブリージョンを割り当てます。各リージョンは重複しない独自のデータセットを主とするため、書き込み操作は適切な局所性を確保するために保護する必要があります。

このモードは 1 つのリージョンへの書き込みと似ていますが、各ユーザーに関連するデータをそのユーザーに近いネットワークに配置できるため、書き込み操作のレイテンシーが低くなる点が異なります。また、すべてのリージョンのインフラストラクチャがすでにアクティブになっているため、周

囲のインフラストラクチャがリージョン間でより均等に分散され、フェイルオーバーシナリオ中のインフラストラクチャの構築に必要な作業が少なくなります。



アイテムのホームリージョンはいくつかの方法で決定できます。

- 内在性:特別な属性やパーティションキーに埋め込まれた値など、データの一部がホームリージョンを明確にします。この手法については、ブログ記事「リージョンピニングを使用して Amazon DynamoDB グローバルテーブル内のアイテムのホームリージョンを設定する」で説明されています。

- ネゴシエーション:各データセットのホームリージョンは、アサインメントを管理する個別のグローバルサービスなど、外部的な方法でネゴシエーションされます。アサインメントの期間には限りがあり、その後は再交渉の対象となる場合があります。
- テーブル指向:レプリケートするグローバルテーブルを 1 つ作成する代わりに、レプリケートするリージョンの数と同じ数のグローバルテーブルを作成します。各テーブルの名前はホームリージョンを示します。標準オペレーションでは、すべてのデータがホームリージョンに書き込まれ、他のリージョンは読み取り専用のコピーを保持します。フェイルオーバー中、別のリージョンがそのテーブルの書き込み権限を一時的に引き継ぎます。

たとえば、あなたがゲーム会社で働いていると想像してみてください。世界中のすべてのゲーマーに、低レイテンシーの読み取りと書き込み操作が必要です。各ゲーマーを自分に最も近い地域に割り当てます。そのリージョンが読み取りと書き込みの操作をすべて引き受けるため、read-after-write 一貫性が保たれます。ただし、ゲーマーが旅行したり、ホームリージョンが停止したりした場合は、データの完全なコピーが別のリージョンで利用可能になり、ゲーマーを別のホームリージョンに割り当てることができます。

別の例として、ビデオ会議会社で働いていると想像してみてください。各電話会議のメタデータは特定の地域に割り当てられます。発信者は、最も近いリージョンを利用してレイテンシーを最小限に抑えることができます。リージョンが停止した場合、システムが呼び出しの処理をデータの複製コピーがすでに存在する別のリージョンに移動できるため、グローバルテーブルを使用すると迅速に回復できます。

グローバルテーブルのルーティング戦略

グローバルテーブルのデプロイで最も複雑な部分は、おそらくリクエストルーティングの管理です。リクエストは、まずエンドユーザーから、何らかの方法で経路が選定されたリージョンに送る必要があります。リクエストでは、 AWS Lambda 関数、コンテナ、または Amazon Elastic Compute Cloud (Amazon EC2) ノードによってバックアップされたロードバランサーと、別のデータベースを含むその他のサービスで構成される可能性のあるコンピューティングレイヤーなど、そのリージョンの一部のサービスのスタックが発生します。このコンピューティングレイヤーは DynamoDB と通信します。これは、そのリージョンのローカルエンドポイントを使用して行う必要があります。グローバルテーブルのデータは、参加している他のすべてのリージョンにレプリケートされ、各リージョンには DynamoDB テーブルを中心に同様のサービスのスタックがあります。

グローバルテーブルは、さまざまなリージョンの各スタックに同じデータのローカルコピーを提供します。1 つのリージョンの 1 つのスタックを対象とした設計を行い、ローカルの DynamoDB テーブルに問題が発生した場合は、セカンダリリージョンの DynamoDB エンドポイントにリモート呼び出しを行うということも考えられます。これはベストプラクティスではありません。リージョンをまたいだ場合のレイテンシーは、ローカルアクセスの場合より 100 倍も高くなる可能性があります。back-and-forth 一連の 5 つのリクエストは、ローカルで実行されるとミリ秒、地球を横断すると数秒かかる場合があります。エンドユーザーを別のリージョンにルーティングして処理することをお勧めします。耐障害性を確保するには、複数のリージョンにまたがるレプリケーションが必要です。コンピューティングレイヤーとデータレイヤーのレプリケーションです。

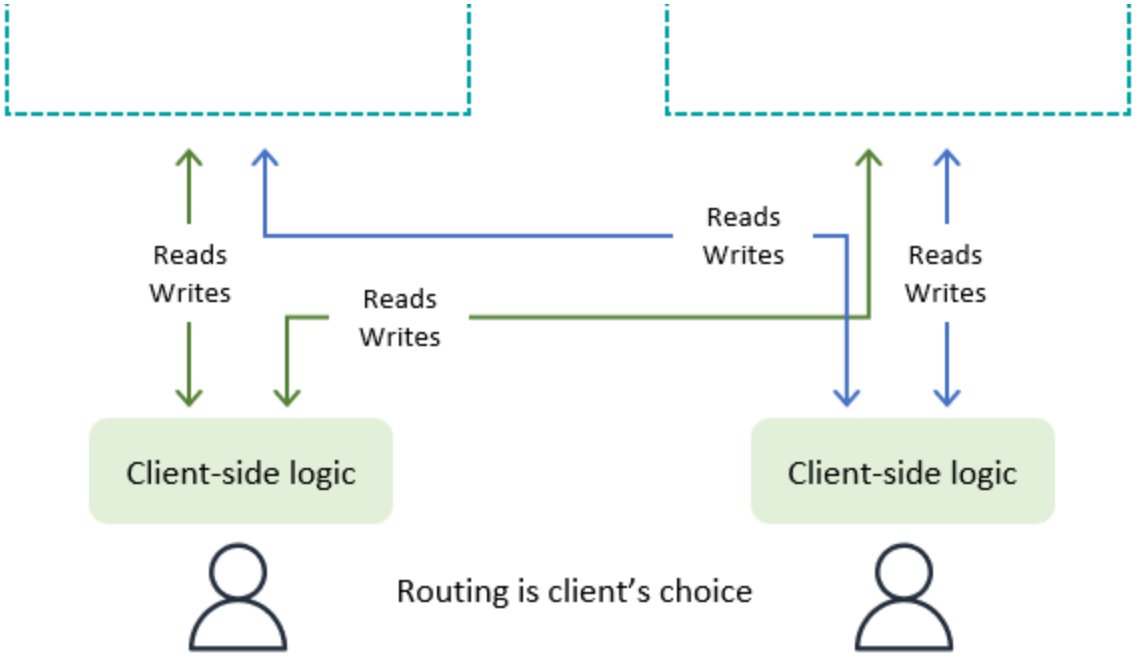
処理のためにエンドユーザーリクエストをリージョンにルーティングする方法は多数あります。適切な選択肢は、書き込みモードとフェイルオーバーに関する考慮事項によって異なります。このセクションでは、クライアント駆動型、コンピューティングレイヤー、Amazon Route 53、 の 4 つのオプションについて説明します AWS Global Accelerator。

トピック

- クライアント主導のリクエストルーティング
- コンピューティングレイヤーのリクエストルーティング
- Route 53 のリクエストルーティング
- Global Accelerator のリクエストルーティング

クライアント主導のリクエストルーティング

クライアント主導のリクエストルーティングでは、エンドユーザークライアント (アプリケーション、 を含むウェブページ JavaScript、または別のクライアント) は、有効なアプリケーションエンドポイント (リテラル DynamoDB エンドポイントではなく Amazon API Gateway エンドポイントなど) を追跡し、独自の埋め込みロジックを使用して通信するリージョンを選択します。ランダム選択、観測された最小レイテンシー、観測された最大帯域幅測定値、またはローカルで実行されたヘルスチェックに基づいて選択できます。



利点として、クライアント主導のリクエストルーティングは、実際のパブリックインターネットトラフィックの状況などに適応して、パフォーマンスの低下に気づいた場合にリージョンを切り替えることができます。クライアントは、すべての潜在的なエンドポイントを認識している必要がありますが、新しいリージョンエンドポイントの立ち上げは頻繁に起こることではありません。

任意のリージョンへの書き込みモードでは、クライアントは優先エンドポイントを一方的に選択できます。1 つのリージョンへのアクセスが損なわれた場合、クライアントは別のエンドポイントにルーティングできます。

1 つのリージョンへの書き込みモードでは、クライアントは現在アクティブなリージョンに書き込みリクエストをルーティングするメカニズムが必要です。これは、現在書き込みリクエストを受け入れているリージョンを経験的にテストするなど、基本的なメカニズムである可能性があります (書き込みの拒否に注意し、別のリージョンにフォールバックする) 。または、グローバルコーディネーターを使用して現在のアプリケーションの状態をクエリするなど、複雑なメカニズムでもかまいません

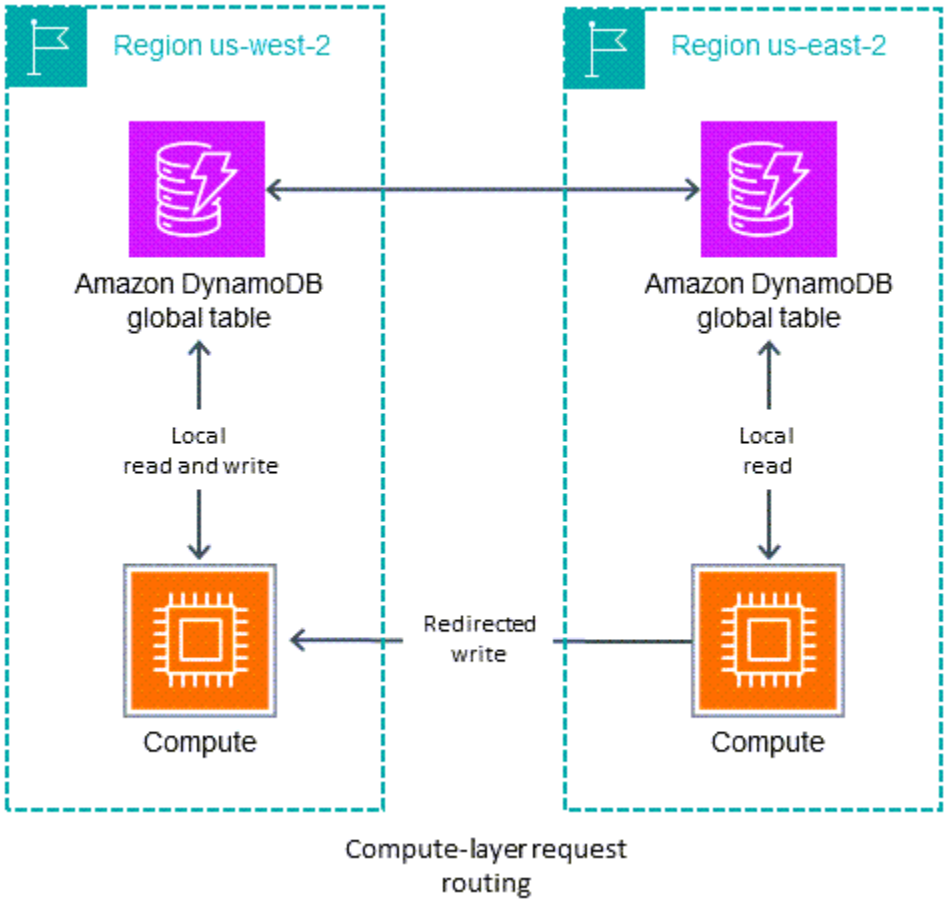
(おそらく、Amazon Route 53 Application Recovery Controller (ARC) ルーティングコントロール上に構築され、このようなニーズに対してグローバル状態を維持するためのクォーラム駆動型システムを提供します)。クライアントは、読み込みリクエストが結果整合性のために任意のリージョンに送信できるか、または強力な整合性のためにアクティブなリージョンにルーティングする必要があるかを決定できます。

リージョンへの書き込みモードでは、クライアントは作業するデータセットのホームリージョンを決定する必要があります。例えば、クライアントがユーザーアカウントに対応し、各ユーザーアカウントがリージョンにホームされている場合、クライアントはグローバルログインシステムから認証情報とともに使用する適切なエンドポイント割り当てをリクエストできます。

例えば、ユーザーがウェブ経由でビジネス財務を管理できるようにする金融サービス会社では、リージョンへの書き込みモードでグローバルテーブルを使用します。各ユーザーは中央サービスにログインする必要があります。このサービスは、認証情報と、その認証情報が機能するリージョンのエンドポイントを返します。返されるリージョンは、ユーザーのデータセットが現在ホームになっている場所に基づいています。認証情報の有効期間は短期です。その後、ウェブページは新しいログインを自動的にネゴシエートします。これにより、ユーザーのアクティビティを新しいリージョンにリダイレクトする可能性があります。

コンピューティングレイヤーのリクエストルーティング

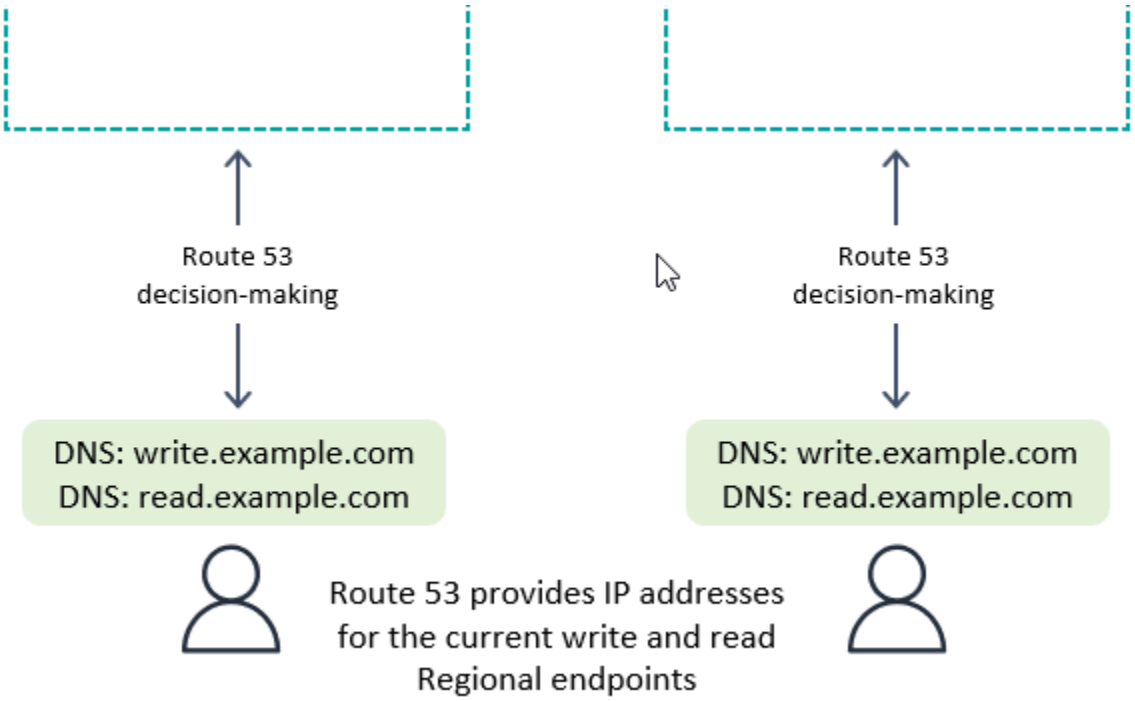
コンピューティングレイヤーのリクエストルーティングでは、コンピューティングレイヤーで実行されるコードによって、リクエストをローカルで処理するか、別のリージョンで実行されているそれ自体のコピーに渡すかが決まります。あるリージョンへの書き込みモードを使用すると、コンピューティングレイヤーは、それがアクティブなリージョンではないことを検出し、すべての書き込みオペレーションを別のリージョンに転送しながらローカル読み取りオペレーションを許可する場合があります。このコンピューティングレイヤーコードは、データトポロジとルーティングルールを認識して、どのリージョンをどのデータに対してアクティブにするかを指定する最新の設定に基づいて、確実に適用する必要があります。リージョン内の外側のソフトウェアスタックは、読み取りリクエストと書き込みリクエストがマイクロサービスによってどのようにルーティングされるかを認識する必要はありません。堅牢な設計では、受信側リージョンが書き込みオペレーションの現在のプライマリであるかどうかを検証します。プライマリでない場合は、グローバル状態を修正する必要があることを示すエラーが生成されます。プライマリリージョンが変更中の場合、受信側リージョンが書き込みオペレーションをしばらくバッファリングすることもあります。いずれの場合も、リージョン内のコンピューティングスタックはローカルの DynamoDB エンドポイントにのみ書き込みますが、コンピューティングスタック間で相互に通信する可能性があります。



「re:Invent 2022」で説明されているように、Vanguard Group は、このルーティングプロセスにグローバルオーケストレーションおよびステータスツール (GOaST) と呼ばれるシステムと、グローバルマルチリージョンライブラリ (GMRlib) と呼ばれるライブラリを使用します。1 follow-the-sun つのプライマリモデルを使用します。GOaST は、前のセクションで説明した Route 53 ARC ルーティングコントロールと同様に、グローバル状態を維持します。グローバルテーブルを使用して、どのリージョンがプライマリリージョンであるか、次のプライマリスイッチがいつスケジュールされるかを追跡します。すべての読み取りおよび書き込みオペレーションは、GOaST と連携する GMRlib を経由します。GMRlib を使用すると、読み取りオペレーションを低レイテンシーでローカルで実行できます。書き込みオペレーションの場合、GMRlib はローカルリージョンが現在のプライマリリージョンであるかどうかをチェックします。プライマリリージョンである場合、書き込みオペレーションは直接完了します。そうでない場合、GMRlib は書き込みタスクをプライマリリージョンの GMRlib に転送します。受信側のライブラリは、自身もプライマリリージョンであると認識したことを確認します。そうでない場合は、エラーを生成します。これにより、グローバル状態の伝播が遅延します。このアプローチでは、リモート DynamoDB エンドポイントに直接書き込まないため、検証上の利点があります。

Route 53 のリクエストルーティング

Amazon Route 53 はドメインネームサービス (DNS) のテクノロジーです。Route 53 では、クライアントはよく知られている DNS ドメイン名を検索してエンドポイントをリクエストし、Route 53 は最も適切であると判断したリージョンのエンドポイントに対応する IP アドレスを返します。Route 53 には、適切なリージョンを決定するために使用するルーティングポリシーの長いリストがあります。また、フェイルオーバールーティングを実行して、ヘルスチェックに失敗したリージョンからトラフィックをルーティングすることもできます。



任意のリージョンへの書き込みモード、またはバックエンドのコンピューティングレイヤーリクエストルーティングと組み合わせると、Route 53 は、最も近いネットワークや地理的近接のリージョンの選択、またはその他の選択肢など、複雑な内部ルールに基づいてリージョンを返すように完全に自由にできます。

1 つのリージョンへの書き込みモードでは、(Route 53 ARC を使用して) 現在アクティブなリージョンを返すように Route 53 を設定できます。クライアントがパッシブリージョン (読み取りオペレーションなど) に接続する場合、別の DNS 名を検索する可能性があります。

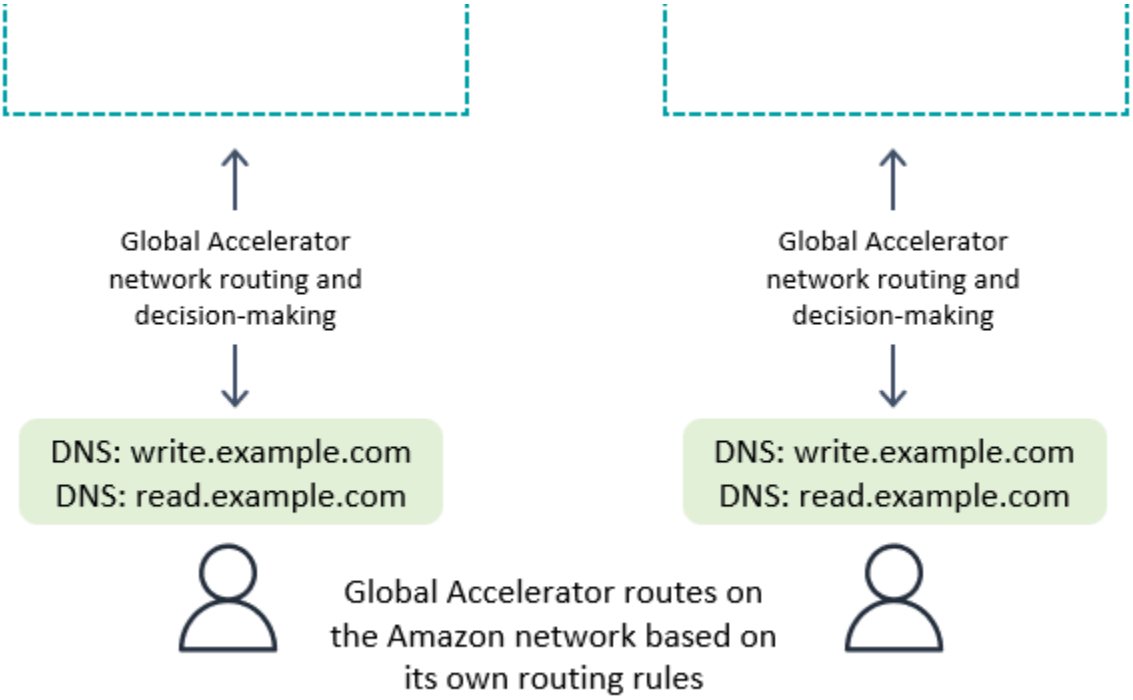
> **Note**
>
> クライアントは、ドメイン名の有効期間 (TTL) 設定で指定された期間にわたって、Route 53 からの応答に含まれている IP アドレスをキャッシュします。TTL を長くすると、すべての

クライアントが新しいエンドポイントを認識するまでの目標復旧時間 (RTO) が長くなります。フェイルオーバーを使用する場合は通常 60 秒です。すべてのソフトウェアが DNS TTL の有効期限に完全に準拠しているわけではなく、オペレーティングシステム、仮想マシン、アプリケーションなど、複数のレベルの DNS キャッシュが存在する可能性があります。

リージョンへの書き込みモードでは、コンピューティングレイヤーのリクエストルーティングも使用していない限り、Route 53 を避けることをお勧めします。

Global Accelerator のリクエストルーティング

クライアントAWS Global Acceleratorを使用して、Route 53 でよく知られているドメイン名を検索します。ただし、クライアントはリージョンのエンドポイントに対応する IP アドレスを返す代わりに、最も近い AWS エッジロケーションにルーティングするエニーキャスト静的 IP アドレスを返します。そのエッジロケーションから、すべてのトラフィックはプライベート AWS ネットワークで、Global Accelerator 内で維持されるルーティングルールによって選択されたリージョンの一部のエンドポイント (Network Load Balancer、Application Load Balancer、EC2 インスタンス、または Elastic IP アドレス) にルーティングされます。Global Accelerator のリクエストルーティングは、Route 53 ルールに基づくルーティングと比較して、パブリックインターネットのトラフィック量を削減するため、レイテンシーが短縮されます。さらに、Global Accelerator はルーティングルールを変更するために DNS TTL の有効期限に依存しないため、ルーティングをより迅速に調整できます。



任意のリージョンへの書き込みモード、またはバックエンドのコンピューティングレイヤーリクエストルーティングと組み合わせると、Global Accelerator はシームレスに動作します。クライアントは最も近いエッジロケーションに接続するため、どのリージョンがリクエストを受け取るかについて心配する必要はありません。

1 つのリージョンへの書き込みモードでは、Global Accelerator ルーティングルールは現在アクティブなリージョンにリクエストを送信する必要があります。ヘルスチェックを使用すると、グローバルシステムによってアクティブなリージョンとは見なされていないリージョンの障害を人為的に報告できます。DNS と同様に、読み取りリクエストが任意のリージョンからのものである場合は、別の DNS ドメイン名を使用して読み取りリクエストをルーティングできます。

リージョンへの書き込みモードでは、コンピューティングレイヤーのリクエストルーティングも使用していない限り、Global Accelerator を避けることをお勧めします。

グローバルテーブルの退避プロセス

リージョンの退避とは、アクティビティ (通常は書き込みアクティビティ、場合によっては読み取りアクティビティ) をそのリージョンから移行するプロセスです。

ライブリージョンからの退避

ライブリージョンを退避する理由はさまざまです。たとえば、通常のビジネス活動の一環として (たとえば、a follow-the-sun を使用している場合は 1 つのリージョンに書き込む)、現在アクティブなリージョンを変更するというビジネス上の意思決定、DynamoDB 外のソフトウェアスタックの障害への対応、リージョン内で通常よりも高いレイテンシーなどの一般的な問題が発生している場合などです。

任意のリージョンへの書き込みモードでは、ライブリージョンから簡単に退避できます。任意のルーティングシステムを使用してトラフィックを代替リージョンにルーティングし、退避したリージョンですでに行われた書き込み操作を通常どおり複製できます。

あるリージョンへの書き込みモードとリージョンへの書き込みモードでは、新しいアクティブリージョンで書き込み操作を開始する前に、アクティブリージョンへのすべての書き込み操作が完全に記録され、ストリーム処理され、グローバルに伝達されていることを確認する必要があります。これにより、future 書き込み操作が最新バージョンのデータに対して処理されるようになります。

例えば、リージョン A がアクティブ、リージョン B がパッシブだとします (リージョン A をホームとするテーブル全体または項目のいずれかが対象)。退避を実行する一般的なメカニズムは、A への書き込みオペレーションを一時停止し、そのオペレーションが B に完全に伝播されるまで待ち、アーキテクチャスタックを更新して B がアクティブであることを認識してから、B への書き込みオペレーションを再開することです。リージョン A がリージョン B にデータを完全にレプリケートしたことを確実に示すメトリクスはありません。リージョン A が正常であれば、リージョン A への書き込みオペレーションを一時停止し、`ReplicationLatency` メトリクスの最新の最大値の 10 倍を待てば、通常はレプリケーションが完了したことを十分に確認できます。リージョン A に異常があり、他の領域でのレイテンシーの増加も示している場合は、待機時間の倍数を大きくします。

オフラインリージョンからの退避

考慮すべき特別なケースがあります。リージョン A が予告なしに完全にオフラインになった場合はどうなるでしょうか。これは非常にありそうもないことですが、それでも考慮する必要があります。これが発生した場合、リージョン A でまだ伝播されていない書き込みオペレーションはすべて保持

され、リージョン A がオンラインに戻った後に伝播されます。書き込みオペレーションは失われませんが、その伝播は無期限に遅延します。

このイベントの処理方法は、アプリケーションの決定によります。ビジネスの継続性を確保するために、書き込みオペレーションを新しいプライマリリージョン B に移行することが必要になる場合があります。ただし、リージョン A の項目に対する書き込みオペレーションの伝播が保留されているときに、リージョン B でその項目が更新を受け取ると、最終書き込み者優先モデルに従ってその伝播は抑制されます。リージョン B での更新は、着信する書き込みリクエストを抑制する可能性があります。

いずれのリージョンへの書き込みモードでも、リージョンAのアイテムは最終的にリージョンBに伝播され、リージョンAがオンラインに戻るまでアイテムが見つからない可能性があることを認識して、リージョンBで読み取りと書き込みの操作を続行できます。同じ書き込み操作を行う場合など、可能な場合は、(たとえばアップストリームのイベントソースを使用して) 最近の書き込みトラフィックをリプレイすることを検討してください。これにより、欠落している可能性のある書き込み操作のギャップを埋め、最後の書き込みが競合解決され、入力された書き込み操作の最終的な伝播が抑制されます。

他の書き込みモードでは、out-of-date少しだけ世界観を持って作業をどの程度続けられるかを考える必要があります。リージョン A がオンラインに戻るまで、ReplicationLatency で追跡する書き込みオペレーションの一部の短い期間は失われます。ビジネスを継続できるでしょうか。一部のユースケースでは可能ですが、他のユースケースでは追加の緩和メカニズムがないと難しい場合があります。

たとえば、あるリージョンが完全に停止した後でも、利用可能なクレジット残高を中断することなく維持する必要があるとします。残高を地域Aと地域Bの2つの異なる項目に分割し、それぞれ利用可能な残高の半分から始めることができます。この場合は、自分のリージョンへの書き込みモードを使用します。各リージョンで処理されたトランザクションの更新は、残高のローカルコピーに対して書き込まれます。リージョン A が完全にオフラインになっても、リージョン B でトランザクション処理を続行でき、書き込みオペレーションはリージョン B に保持されている残高部分に制限されます。このように残高を分割すると、残高が少なくなった場合やクレジットの再調整が必要になった場合に複雑になりますが、保留中の書き込みオペレーションが不安定でも安全にビジネスを回復できる一例となります。

別の例として、Web フォームのデータをキャプチャしているとします。オプティミスティック・コンカレンシー・コントロール (OCC) を使用してデータ項目にバージョンを割り当て、最新バージョンを隠しフィールドとして Web フォームに埋め込むことができます。送信するたびに、データベース内のバージョンがフォーム作成時のバージョンとまだ一致する場合にのみ、書き込みオペレーションが成功します。バージョンが一致しない場合、データベース内の現在のバージョンに基づいてウェ

ブフォームを更新 (または慎重にマージ) することで、ユーザーは再度続行できます。OCC モデルは通常、別のクライアントがデータを上書きして新しいバージョンを生成するのを防ぎますが、フェイルオーバー中にクライアントが古いバージョンのデータに遭遇する可能性がある場合にも役立ちます。タイムスタンプをバージョンとして使用しているとします。フォームは最初に 12:00 にリージョン A に対して作成されましたが、(フェイルオーバー後に) リージョン B への書き込みを試みたところ、データベースの最新バージョンが 11:59 であることがわかります。このシナリオの場合、クライアントは 12:00 バージョンがリージョン B に伝播されるのを待ってからそのバージョンに書き込むか、11:59 バージョンに基づいて構築し、新しい 12:01 バージョンを作成することができます (書き込み後、リージョン A の回復後に着信したバージョンは抑制されます)。

3 番目の例として、ある金融サービス会社が顧客アカウントとその金融取引に関するデータを DynamoDB データベースに保持しています。リージョン A が完全に停止した場合、そのアカウントに関連する書き込みアクティビティがすべて、リージョン B で完全に利用可能であることを確認するか、リージョン A がオンラインに戻るまでアカウントを部分的に隔離する必要があります。すべての業務を一時停止するのではなく、伝播されていないトランザクションがあると判断したごく一部のアカウントに対してのみ業務を一時停止することにしました。これを実現するために、同社は 3 番目のリージョン (リージョン C と呼びます) を使用しました。リージョン A で書き込みオペレーションを処理する前に、保留中のオペレーション (アカウントの新規トランザクション数など) の簡潔な要約をリージョン C に配置しました。この要約は、リージョン B のビューが完全に最新であるかどうかを判断するのに十分でした。このアクションにより、リージョン C での書き込み時点から、リージョン A が書き込みオペレーションを受け入れてリージョン B がそれを受け取るまでの間、アカウントは事実上ロックされました。リージョン C のデータは、フェイルオーバープロセスの一部として以外は使用されませんでした。その後、リージョン B はリージョン C とデータを相互チェックして、古いアカウントがないかどうかを確認できました。これらのアカウントは、リージョン A のリカバリによって部分的なデータがリージョン B に伝達されるまで、隔離されたとマークされます。リージョン C に障害が発生した場合、代わりに新しいリージョン D を起動して使用することができます。リージョン C のデータは非常に一時的なもので、up-to-date数分後には処理中の書き込み操作に関する十分な記録がリージョン D に蓄積され、十分に役に立ちます。リージョン B に障害が発生しても、リージョン A はリージョン C と協力して書き込みリクエストを引き続き受け入れることができます。同社は、より高いレイテンシーの書き込み (C に続けて A という 2 つのリージョンへの書き込み) をあえて受け入れ、アカウントの状態を簡潔に要約できるデータモデルを持つことができたことを喜んでいます。

グローバルテーブルのスループットキャパシティプランニング

リージョン間でトラフィックを移行する場合は、キャパシティに関する DynamoDB テーブルの設定を慎重に考慮する必要があります。

書き込みキャパシティの考慮事項は以下のとおりです。

- グローバルテーブルは、オンデマンドモードにするか、自動スケーリングを有効にしてプロビジョニングする必要があります。
- 自動スケーリングでプロビジョニングした場合、書き込み設定 (最小使用率、最大使用率、およびターゲット使用率) はリージョン間でレプリケートされます。自動スケーリングの設定は同期されますが、実際にプロビジョニングされた書き込みキャパシティは、リージョン間で独立して変動する可能性があります。
- プロビジョニングされた書き込みキャパシティが異なっている理由の 1 つは、有効期限 (TL) 機能によるものです。DynamoDB で TL を有効にすると、項目の有効期限を示す値を含む属性名を指定できます。値を含む属性名は、秒単位の Unix エポックタイムです。その後、DynamoDB は書き込みコストを発生させずに項目を削除できます。グローバルテーブルを使用すると、任意のリージョンで TTL を設定でき、その設定はグローバルテーブルに関連付けられている他のリージョンに自動的にレプリケートされます。項目が TTL ルールで削除の対象となった場合、削除はどのリージョンでも実行できます。削除操作はソーステーブルの書き込みユニットを消費せずに実行されますが、レプリカテーブルにはその削除操作の複製書き込みが行われるため、複製書き込み単位のコストが発生します。
- 自動スケーリングを使用する場合は、プロビジョニングされた最大書き込みキャパシティの設定が、すべての書き込みオペレーションとすべての潜在的な TTL 削除オペレーションを処理するのに十分な大きさであることを確認してください。自動スケーリングは、書き込み消費量に応じて各リージョンを調整します。オンデマンドテーブルには、プロビジョニングされた最大書き込みキャパシティの設定はありませんが、テーブルレベルの最大書き込みスループット制限により、オンデマンドテーブルが許可する最大持続書き込みキャパシティが指定されます。デフォルトの制限は 40,000 ですが、調整可能です。オンデマンドテーブルに必要となるすべての書き込みオペレーション (TTL 書き込みオペレーションを含む) を処理できるように、この値を十分に高く設定することをお勧めします。グローバルテーブルを設定する場合、この値は参加しているすべてのリージョンで同じでなければなりません。

読み込みおよび管理での考慮事項は以下のとおりです。

- 読み取りキャパシティの管理に関する設定は、リージョンごとに読み取りパターンが異なることが想定されているため、リージョン間で異なることが許容されます。グローバルレプリカを初めてテーブルに追加すると、ソースリージョンのキャパシティが反映されます。作成後、読み取りキャパシティの設定は調整できますが、この調整した設定は反対側には転送されません。
- DynamoDB 自動スケーリングを使用するときは、プロビジョニングされた最大読み取りキャパシティの設定が、すべてのリージョンのすべての読み取りオペレーションを処理するのに十分な大きさであることを確認してください。標準的なオペレーション時には、読み取りキャパシティがリージョン全体に分散される可能性がありますが、フェイルオーバー時には、テーブルが読み取りワークロードの増大に自動的に対応できなければなりません。オンデマンドテーブルにはプロビジョニングされた最大読み取りキャパシティの設定はありませんが、テーブルレベルの最大読み取りスループット制限により、オンデマンドテーブルが許容する最大持続読み取りキャパシティが指定されます。デフォルトの制限は 40,000 ですが、調整可能です。すべての読み取りオペレーションをこの単一のリージョンにルーティングする場合、テーブルが必要とするすべての読み取りオペレーションを処理できるように、この値を十分に高く設定することをお勧めします。
- あるリージョンのテーブルが通常は読み取りトラフィックを受信しないが、フェイルオーバー後に大量の読み取りトラフィックを吸収しなければならない場合は、テーブルのプロビジョニングされた読み取りキャパシティを増やし、テーブルの更新が完了するのを待ってから、テーブルを再度プロビジョニングできます。テーブルをプロビジョニングモードのままにすることも、オンデマンドモードに切り替えることもできます。これにより、テーブルが事前にウォームアップされ、より高いレベルの読み取りトラフィックを受け入れることができます。

Route 53 ARC には準備状況チェック機能があり、Route 53 を使用してリクエストをルーティングするかどうかにかかわらず、DynamoDB リージョンのテーブル設定とアカウントクォータが類似していることを確認するのに役立ちます。これらの準備状況チェックは、アカウントレベルのクォータを調整して一致させるのにも役立ちます。

グローバルテーブルの準備チェックリスト

グローバルテーブルをデプロイするときの決定事項とタスクには、次のチェックリストを使用してください。

- グローバルテーブルに参加するリージョンの数と各リージョンを決定します。
- アプリケーションの書き込みモードを決定します。
- 書き込みモードに基づいてルーティング戦略を計画します。
- 書き込みモードとルーティング戦略に基づいて、避難計画を定義します。
- リージョンごとのヘルス、レイテンシー、エラーに関するメトリクスをキャプチャします。DynamoDB メトリクスのリストについては、AWSブログ記事「Amazon DynamoDB をモニタリングして運用状況を把握する」を参照してください。また、合成カナリア（障害を検出するための人工的なリクエスト）や顧客トラフィックのライブ監視も使用する必要があります。DynamoDB メトリックスにすべての問題が表示されるわけではありません。
- `ReplicationLatency` の継続的な増加に対するアラームを設定します。この増加は、グローバルテーブルの書き込み設定がリージョンごとに異なるという偶発的な設定ミスを示している可能性があり、その結果、レプリケートされたリクエストが失敗し、レイテンシーが高くなります。また、リージョンに混乱が生じていることを示している可能性もあります。良い例としては、最近の平均が 180,000 ミリ秒を超えた場合にアラートを生成することが挙げられます。また、レプリケーションが停止していることを示す 0 `ReplicationLatency` に下がるのにも注意してください。
- 各グローバルテーブルに十分な最大読み取り/書き込み設定を割り当てます。
- 地域から避難する条件を特定してください。決定に人間の判断が関与する場合は、すべての考慮事項を文書化します。この作業は、必要に迫られて行うのではなく、事前に慎重に行う必要があります。
- リージョンから退避する際に実行する必要があるすべてのアクションのランブックを用意しておきます。通常、グローバルテーブルに関係する作業はほとんどありませんが、スタックの残りの部分を移動するのは複雑な作業になる場合があります。

> **Note**
>
> フェイルオーバー・プロシージャでは、リージョンの障害時に一部のコントロール・プレーンの操作が低下する可能性があるため、コントロール・プレーンの操作には依存せず、データ・プレーンの操作のみに頼るのがベスト・プラクティスです。詳細について

は、AWS ブログ記事「Amazon DynamoDB グローバルテーブルを使用した回復性の高いアプリケーションの構築: パート 4」を参照してください。

- リージョンの退避を含め、ランブックのあらゆる側面を定期的にテストします。テストしないと、ランブックの信頼性が低下します。
- アプリケーション全体 (グローバルテーブルを含む) AWS Resilience Hubの回復力を評価するために使用することを検討してください。アプリケーションのポートフォリオのレジリエンシー状態の包括的なビューを提供するサービス。
- Route 53 ARC 準備状況チェックを使用してアプリケーションの現在の構成を評価し、ベストプラクティスからの逸脱を追跡することを検討してください。
- Route 53 またはグローバルアクセラレータで使用するヘルスチェックを作成するときは、データベースフロー全体をカバーする一連の呼び出しを行います。チェック対象を DynamoDB エンドポイントが稼働していることだけを確認すると、AWS Identity and Access Management (IAM) 設定エラー、コードデプロイの問題、DynamoDB 外部のスタックの障害、平均よりも高い読み取りまたは書き込みレイテンシーなど、多くの障害モードに対応できなくなります。

グローバルテーブルに関するよくある質問

このセクションでは、DynamoDB グローバルテーブルのよくある質問に回答します。

グローバルテーブルの料金はいくらですか?

- 従来の DynamoDB テーブルへの書き込みオペレーションは、プロビジョニングされたテーブルの場合、書き込みキャパシティユニット (WCU)、オンデマンドテーブルの場合、書き込みリクエストユニット (WRU) で料金が設定されます。5 KB のアイテムを書き込むと、5 ユニットの料金が発生します。グローバルテーブルへの書き込みは、プロビジョニングされたテーブルの場合、レプリケートされた書き込みキャパシティユニット (rWCU)、オンデマンドテーブルの場合、レプリケートされた書き込みリクエストユニット (rWRU) で料金が設定されます。
- rWCU と rWRU には、レプリケーションの管理に必要なストリーミングインフラストラクチャのコストが含まれます。そのため、WCU や WRU よりも料金が 50% 高くなります。リージョン間のデータ転送料金が適用されます。
- rWCU および rWRU 料金は、項目が直接書き込まれるか、レプリケーションを通じて書き込まれるすべてのリージョンで発生します。
- グローバルセカンダリインデックス (GSI) への書き込みはローカル書き込みオペレーションと見なされ、通常の書き込みユニットを使用します。
- 現在、rWCU に利用できるリザーブドキャパシティはありません。WCU 用のリザーブドキャパシティの購入は、GSI が書き込みユニットを消費するテーブルでは依然として有益な場合があります。
- グローバルテーブルに新しいリージョンを追加すると、DynamoDB は新しいリージョンを自動的にブートストラップし、テーブルの GB サイズに基づいてテーブルを復元したかのように料金が発生します。リージョン間のデータ転送料金もかかります。

グローバルテーブルはどのリージョンでサポートされていますか?

グローバルテーブルはすべての AWS リージョン でサポートされています。

GSI はグローバルテーブルでどのように処理されますか?

グローバルテーブル (現在、バージョン 2019) では、あるリージョンで GSI を作成すると、他の参加リージョンでも自動的に作成され、自動的にバックフィルされます。

グローバルテーブルのレプリケーションを停止するにはどうしたらいいですか?

レプリカテーブルは、他のテーブルを削除するのと同じ方法で削除できます。グローバルテーブルを削除すると、そのリージョンへのレプリケーションが停止し、そのリージョンに保持されているテーブルのコピーが削除されます。ただし、テーブルのコピーを独立したエンティティとして保持している間は、レプリケーションを停止または一時停止することはできません。

Amazon DynamoDB Streams はグローバルテーブルとどのようにやり取りするのですか?

各グローバルテーブルは、書き込み元に関係なく、すべての書き込みオペレーションに基づいて独立したストリームを生成します。DynamoDB ストリームを 1 つのリージョンで使用するか、すべてのリージョンで (個別に) 使用するかを選択できます。レプリケートされた書き込みオペレーションではなく、ローカル書き込みオペレーションを処理する場合は、各項目に独自のリージョン属性を追加して、書き込みリージョンを識別できます。次に、AWS Lambda イベントフィルターを使用して、ローカルリージョンでの書き込みオペレーションに対してのみ Lambda 関数を呼び出すことができます。これは挿入および更新オペレーションには役立ちますが、削除オペレーションには役立ちません。

グローバルテーブルはトランザクションをどのように処理するのですか?

トランザクションオペレーションは、書き込みオペレーションが最初に発生したリージョン内でのみ、アトミック性、整合性、分離性、耐久性 (ACID) を保証します。グローバルテーブルのリージョン間では、トランザクションはサポートされていません。例えば、米国東部 (オハイオ) および米国西部 (オレゴン) リージョンにレプリカを持つグローバルテーブルがあり、米国東部 (オハイオ) リージョンで `TransactWriteItems` オペレーションを実行すると、変更がレプリケートされたときに米国西部 (オレゴン) では部分的に完了したトランザクションが観察されることがあります。変更は、ソースリージョンでコミットされた後でのみ、他のリージョンにレプリケートされます。

グローバルテーブルは DynamoDB Accelerator (DAX) キャッシュとどのようにやり取りするのですか?

グローバルテーブルは DynamoDB を直接更新することで DAX をバイパスするため、DAX はそれが古いデータを保持していることを認識しません。DAX キャッシュは、キャッシュの TTL が期限切れになったときにのみ更新されます。

テーブルのタグは伝播されますか?

いいえ、タグは自動的には伝播されません。

すべてのリージョンのテーブルをバックアップすべきですか、それとも 1 つのリージョンだけでよいですか?

答えはバックアップの目的によって異なります。

- データの耐久性を確保したい場合、DynamoDB には既に保護手段が用意されています。このサービスにより、耐久性が保証されます。
- 履歴記録のスナップショットを保持する場合 (規制要件を満たすためなど)、1 つのリージョンでバックアップすれば十分です。AWS Backup を使用してバックアップを別のリージョンにコピーできます。
- 誤って削除または変更されたデータを復元する場合は、1 つのリージョンで DynamoDB point-in-time リカバリ (PITR) を使用します。

AWS CloudFormation を使用してグローバルテーブルをデプロイするにはどうすればいいですか?

- CloudFormation は、DynamoDB テーブルとグローバルテーブルを と の 2 つの個別のリソースとして表`AWS::DynamoDB::Table`しますAWS::DynamoDB::GlobalTable。1 つの方法としては、`GlobalTable` コンストラクトを使用してグローバルになる可能性のあるすべてのテーブルを作成します。これらのテーブルは、最初はスタンドアロンテーブルのままにし、必要に応じて後でリージョンを追加します。
- では CloudFormation、レプリカの数に関係なく、各グローバルテーブルは 1 つのリージョンの 1 つのスタックによって制御されます。テンプレートをデプロイすると、 は 1 つの

スタックオペレーションの一部としてすべてのレプリカ CloudFormation を作成および更新します。同じ AWS::DynamoDB::GlobalTable リソースを複数のリージョンにデプロイしないでください。これは、エラーとなるため、サポートされていません アプリケーションテンプレートを複数のリージョンにデプロイする場合は、条件を使用して単一リージョンに AWS::DynamoDB::GlobalTable リソースを作成できます。または、アプリケーションスタックとは別のスタック内に AWS::DynamoDB::GlobalTable リソースを定義し、それが単一リージョンにのみデプロイされるようにします。

- 通常のテーブルがあり、それを で管理しながらグローバルテーブルに変換する場合 CloudFormation: 削除ポリシーを に設定しRetain、そのテーブルをスタックから削除し、コンソールでテーブルをグローバルテーブルに変換してから、グローバルテーブルを新しいリソースとしてスタックにインポートします。詳細については、AWS GitHub リポジトリ amazon-dynamodb-table-to-global-table-cdk を参照してください。
- クロスアカウントレプリケーションは現在サポートされていません。

結論とリソース

DynamoDB グローバルテーブルにはコントロールがほとんどありませんが、それでも慎重に検討する必要があります。書き込みモード、ルーティングモデル、および退避プロセスを決定する必要があります。すべてのリージョンにわたってアプリケーションをインストルメントし、グローバルヘルスを維持するためのルーティングの調整や退避に備える必要があります。その見返りは、99.999% の可用性を実現するように設計された、低レイテンシーの読み取りおよび書き込み操作を備えたグローバルに分散されたデータセットがあることです。

DynamoDB グローバルテーブルの詳細については、次のリソースを参照してください。

- Amazon DynamoDB ドキュメンテーション
- Amazon Route 53 Application Recovery Controller
- Route 53 ARC 準備状況チェック (AWSドキュメンテーション)
- Route 53 ルーティングポリシー (AWSドキュメント)
- AWS Global Accelerator
- DynamoDB サービスレベルアグリーメント
- AWSマルチリージョンの基礎 (ホワイトペーパー) AWS
- AWS (AWSre: Invent 2022プレゼンテーション) によるデータレジリエンシー設計パターン
- フィデリティ・ インベストメンツと Reltio が Amazon DynamoDB でモダナイズした方法 (re: Invent 2022 プレゼンテーション) AWS
- マルチリージョンのデザインパターンとベストプラクティス (AWSre: Invent 2022 プレゼンテーション)
- ディザスタリカバリ (DR) アーキテクチャの導入AWS、パート III: パイロットライトとウォームスタンバイ (AWSブログ記事)
- リージョンピニングを使用して Amazon DynamoDB グローバルテーブル内のアイテムのホームリージョンを設定する (AWSブログ記事)
- 運用意識向上のための Amazon DynamoDB のモニタリング (AWSブログ投稿)
- DynamoDB のスケーリング:ヒート用のパーティション、ホットキー、スプリットがパフォーマンスに与える影響 (ブログ記事) AWS

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
更新された AWS Global Accelerator 情報	Global Accelerator リクエストルーティング のエンドポイントを修正しました。	2024 年 3 月 14 日
更新された AWS リージョンサポート情報	よくある質問を更新して、グローバルテーブルがすべての AWS リージョンでサポートされるようになったことを示しました。	2023 年 11 月 15 日
初版発行	—	2023 年 5 月 19 日

AWS 規範的ガイダンスの用語集

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

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

A

ABAC

「属性ベースのアクセスコントロール」を参照してください。

抽象化されたサービス

「 マネージドサービス」を参照してください。

ACID

「原子性、一貫性、分離性、耐久性」を参照してください。

アクティブ - アクティブ移行

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

アクティブ - パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、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の方法。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

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

AI

「人工知能」を参照してください。

AIOps

「人工知能オペレーション」を参照してください。

匿名化

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

アンチパターン

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

アプリケーションコントロール

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

アプリケーションポートフォリオ

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

人工知能 (AI)

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

AI オペレーション (AIOps)

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

非対称暗号化

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

原子性、一貫性、分離性、耐久性 (ACID)

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

属性ベースのアクセス制御 (ABAC)

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

信頼できるデータソース

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

アベイラビリティーゾーン

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

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドに正常に移行 AWS するための効率的で効果的な計画を立てるのに役立つ、 のガイドラインとベストプラクティスのフレームワーク。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを編成します。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

個人または組織に混乱や損害を与えることを目的としたボット。

BCP

事業継続計画を参照してください。

動作グラフ

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

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。エンディアンネス も参照してください。

二項分類

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

ブルームフィルター

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

ブルー/グリーンデプロイ

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

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティやインタラクション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、個人や組織に混乱を与えたり、損害を与えたりすることを意図しているものがあります。

ボットネット

マルウェアに感染し、ボットのヘルダーまたはボットオペレーター と呼ばれる、単一関係者の管理下にあるボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

ブランチ

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

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

ブラウンフィールド戦略

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

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

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

ビジネス継続性計画 (BCP)

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

C

CAF

AWS 「クラウド導入フレームワーク」を参照してください。

Canary デプロイ

エンドユーザーへのバージョンの低速かつ増分的なリリース。確信できたら、新しいバージョンをデプロイし、現在のバージョン全体を置き換えます。

CCoE

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

CDC

「データキャプチャの変更」を参照してください。

変更データキャプチャ (CDC)

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

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストします。AWS Fault Injection Service （ AWS FIS ） を使用して、 AWS ワークロードに負荷をかけ、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 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 の投稿」を参照してください。

クラウドコンピューティング

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

クラウド運用モデル

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

導入のクラウドステージ

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

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

これらのステージは、 AWS クラウド エンタープライズ戦略ブログのブログ記事「クラウドファーストへのジャーニー」と「導入のステージ」で Stephen Orban によって定義されました。移行戦略とどのように関連しているかについては、 AWS 「移行準備ガイド」を参照してください。

CMDB

「設定管理データベース」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、 GitHub または が含まれます AWS CodeCommit。コードの各バージョンはブランチと呼ばれます。マイクロサー

ビスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオなどのビジュアル形式から情報を分析および抽出する AI の分野。例えば、 AWS Panorama はオンプレミスのカメラネットワークに CV を追加するデバイスを提供し、Amazon SageMaker は CV の画像処理アルゴリズムを提供します。

設定ドリフト

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

構成管理データベース (CMDB)

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

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンで、または組織全体に 1 つのエンティティとしてデプロイできます。詳細については、 AWS Config ドキュメントの「コンフォーマンスパック」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性

の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「継続的デリバリーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

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

データドリフト

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

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

一元化された管理とガバナンスにより、分散型の分散型データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼できる 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していることを確認できます。詳細については、「 でのデータ境界の構築 AWS」を参照してください。

データの前処理

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

データ出所

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

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには通常、大量の履歴データが含まれており、クエリや分析によく使用されます。

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

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

ディープラーニング

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

defense-in-depth

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、 AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加して、リソースの安全性を確保します。例えば、 defense-in-depth アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

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

デプロイメント

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

開発環境

「環境」を参照してください。

検出管理

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

開発バリューストリームマッピング (DVSM)

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

デジタルツイン

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

ディメンションテーブル

スタースキーマ では、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに一般的に使用されます。

ディザスタ

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

ディザスタリカバリ (DR)

災害によるダウンタイムとデータ損失を最小限に抑えるために使用する戦略とプロセス。詳細については、 AWS Well-Architected フレームワークの「 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の追跡。例えば、 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

「開発値ストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

エッジコンピューティング

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

暗号化

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

暗号化キー

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

エンディアン

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

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink 、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これらのアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの「エンドポイントサービスを作成する」を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (アカウンティング、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 の中央テーブル。事業運営に関する定量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 種類の列が含まれます。

フェイルファスト

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

障害分離境界

では AWS クラウド、障害の影響を制限し AWS リージョン、ワークロードの耐障害性を向上させるアベイラビリティーゾーン、コントロールプレーン、データプレーンなどの境界です。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。

FGAC

「きめ細かなアクセスコントロール」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

G

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)

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

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローはレガシーと見なされ、トランクベースのワークフローはモダンで推奨されるアプローチです。

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 AWS Security Hub、GuardDuty、Amazon Inspector AWS Trusted Advisor、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

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

ハイアベイラビリティ (HA)

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

ヒストリアンのモダナイゼーション

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

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。緊急性のため、通常、修正は一般的な DevOps リリースワークフローの外で行われます。

ハイパーケア期間

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

I

IaC

「Infrastructure as Code」を参照してください。

ID ベースのポリシー

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

アイドル状態のアプリケーション

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

IIoT

「産業モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

既存のインフラストラクチャを更新、パッチ適用、または変更するのではなく、本番ワークロード用の新しいインフラストラクチャをデプロイするモデル。イミュータブルなインフラストラクチャは、本質的にミュータブルなインフラストラクチャ よりも一貫性、信頼性、予測性が高くなります。詳細については、 AWS Well-Architected フレームワークの「変更不可能なインフラストラクチャを使用したデプロイ」のベストプラクティスを参照してください。

インバウンド (受信) VPC

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

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

増分移行

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

インダストリー 4.0

接続、リアルタイムデータ、自動化、分析、AI/ML の進歩を通じて、 のビジネスプロセスのモダナイゼーションを指すために 2016 年に Klaus Schwab によって導入された用語。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

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

産業分野における IoT (IIoT)

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

インスペクション VPC

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

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、「AWS を使用した機械学習モデルの解釈」を参照してください。

IoT

「モノのインターネット」を参照してください。

IT 情報ライブラリ (ITIL)

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

IT サービス管理 (ITSM)

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

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

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

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロー

ドとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、安全でスケーラブルなマルチアカウント AWS 環境のセットアップ を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースのアクセスコントロール」を参照してください。

最小特権

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

リフトアンドシフト

「7 R」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。エンディアンネス も参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

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

メインブランチ

「ブランチ」を参照してください。

マルウェア

コンピュータのセキュリティまたはプライバシーを侵害するように設計されているソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスにつながる

可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

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

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステム。これにより、加工品を現場の完成製品に変換します。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整のために結果を検査する完全なプロセス。メカニズムとは、動作中にそれ自体を強化して改善するサイクルです。詳細については、 AWS Well-Architected フレームワークの「メカニズムの構築」を参照してください。

メンバーアカウント

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

MES

「製造実行システム」を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

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

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロ

イ、再利用可能なコード、回復力などがあります。詳細については、AWS 「サーバーレスサービスを使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

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

Migration Acceleration Program (MAP)

コンサルティングサポート、トレーニング、サービスを提供し、組織がクラウドへの移行のための強固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 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、オペレーション、ビジネスアナリストと所有者、移行エンジニア、デベロッパー、スプリントに取り組む 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には、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: Application Migration Service を使用して Amazon EC2 AWS への移行をリホストします。

Migration Portfolio Assessment (MPA)

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

移行準備状況評価 (MRA)

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

移行戦略

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

ML

「機械学習」を参照してください。

モダナイゼーション

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

モダナイゼーション準備状況評価

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

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

モノリシックアプリケーション (モノリス)

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

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

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

多クラス分類

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

変更可能なインフラストラクチャ

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

O

OAC

「オリジンアクセスコントロール」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」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

「運用レベルの契約」を参照してください。

オンライン移行

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

OPC-UA

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

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

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

オペレーショナルレベルアグリーメント (OLA)

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

運用準備状況レビュー (ORR)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問および関連するベストプラクティスのチェックリスト。詳細については、 AWS Well-Architected フレームワークの「運用準備状況レビュー (ORR)」を参照してください。

運用テクノロジー (OT)

産業運用、機器、インフラストラクチャを制御するために物理環境と連携するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、Industry 4.0 トランスフォーメーションの主要な焦点です。

オペレーション統合 (OI)

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

組織の証跡

の組織 AWS アカウント 内のすべての のすべてのイベントを口グ AWS CloudTrail 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、 ドキュメントの「組織の証跡の作成」を参照してください。 CloudTrail

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から、このフレームワークは人材アクセラレーション と呼ばれます。詳細については、OCM ガイド を参照してください。

オリジンアクセスコントロール (OAC)

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

オリジンアクセスアイデンティティ (OAI)

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

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用技術」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るネットワーク接続を処理する VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

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

個人を特定できる情報 (PII)

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

PII

個人を特定できる情報を参照してください。

プレイブック

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

PLC

「プログラム可能なロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

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

多言語の永続性

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

ポートフォリオ評価

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

述語

`true` または を返すクエリ条件。`false`通常は `WHERE`句にあります。

述語のプッシュダウン

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

予防的コントロール

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

プリンシパル

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

プライバシーバイデザイン

エンジニアリングプロセス全体を通してプライバシーを考慮に入れた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

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

製品ライフサイクル管理 (PLM)

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

本番環境

「環境」を参照してください。

プログラム可能なロジックコントローラー (NAL)

製造では、マシンをモニタリングし、承認プロセスを自動化する、信頼性が高く、適応性の高いコンピュータです。

仮名化

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

パブリッシュ/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの MES では、マイクロサービスは他のマイクロサービスがサブスクライブできるチャネルにイベントメッセージを発行できます。システムは、公開サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

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

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設

定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

責任、説明責任、相談、通知 (RACI) を参照してください。

ランサムウェア

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

RASCI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

「行と列のアクセスコントロール」を参照してください。

リードレプリカ

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

再構築

「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 R」を参照してください。

リージョン

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

回帰

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

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 R」を参照してください。

プラットフォーム変更

「7 R」を参照してください。

再購入

「7 R」を参照してください。

回復性

中断に耐えたり、中断から回復したりするアプリケーションの機能。高可用性とディザスタリカバリは、 で障害耐性を計画する際の一般的な考慮事項です AWS クラウド。詳細については、AWS クラウド 「レジリエンス」を参照してください。

リソースベースのポリシー

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

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R) 、説明責任

(A) 、協議 (C) 、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートを含めると、そのマトリックスは RASCI マトリックスと呼ばれ、サポートを除外すると RACI マトリックスと呼ばれます。

レスポンシブコントロール

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

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

ローテーション

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

行と列のアクセス制御 (RCAC)

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

RPO

「目標復旧時点」を参照してください。

RTO

「目標復旧時間」を参照してください。

ランブック

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

S

SAML 2.0

多くの ID プロバイダー (IdPs) が使用するオープンスタンダード。この機能により、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 AWS Management Console

にログインしたり AWS API オペレーションを呼び出したりでき、組織内のすべてのユーザーを IAM で作成する必要はありません。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントのSAML 2.0 ベースのフェデレーションについてを参照してください。

SCADA

「監視コントロールとデータ収集」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

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

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、予防的 、検出的 、???応答的 、プロアクティブ の 4 つの主なタイプがあります。

セキュリティ強化

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

Security Information and Event Management (SIEM) システム

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

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修正するように設計された、事前定義され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、認証情報のローテーションなどがあります。

サーバー側の暗号化

送信先にあるデータの、それを受け取る AWS サービス 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 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 サービス エンドポイント」を参照してください。

サービスレベルアグリーメント (SLA)

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

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

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

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について 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

「セキュリティ情報とイベント管理システム」を参照してください。

単一障害点 (SPOF)

システムを中断させる可能性のあるアプリケーションの単一の重要なコンポーネントの障害。

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

split-and-seed モデル

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

SPOF

単一障害点 を参照してください。

star スキーマ

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

strangler fig パターン

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

サブネット

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

監視統制とデータ収集 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと生産オペレーション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

ユーザーインタラクションをシミュレートして潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。Amazon CloudWatch Synthetics を使用してこれらのテストを作成できます。

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するサービスへの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「 AWS Organizations を他の AWS のサービスで使用する 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 つのピザを食べることができる小さな 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

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

ワークロード

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

ワークストリーム

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

WORM

「書き込み 1 回」を参照し、多くの を読み取ります。

WQF

「AWS ワークロード認定フレームワーク」を参照してください。

Write Once, Read Many (WORM)

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

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性 を利用する攻撃、通常はマルウェア。

ゼロデイ脆弱性

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

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。