から Amazon DynamoDB RDBMSに移行してアプリケーションをモダナイズする

AWS 規範的ガイダンス

AWS 規範的ガイダンス: から Amazon DynamoDB RDBMSに移行してアプリケーションをモダナイズする

Table of Contents

はじめに 1
DynamoDB の概要 2
特徴と利点 2
パーティションキー 3
インデックス 3
有効期限 4
料金モデル 4
トランザクション 6
大きなコンポーネント 8
バックアップと復元 8
言語とSDKサポート 9
サンプルアプリケーション 10
古いデータアクセスパターン 10
新しいデータアクセスパターン 11
RDBMS テーブルスキーマとマッピング 12
1 つのテーブルデザイン 14
グローバルセカンダリインデックス 15
アクセスパターン 16
オブジェクト永続性インターフェイス 16
ドキュメントインターフェイス 17
低レベル API 18
コンバーター 19
ベストプラクティス 21
オブジェクト永続性アクセスパターンを使用する 21
適切なキャパシティプロビジョニングモードを選択してください 21
キャッシュを使う 21
スキャンの代わりにクエリを使う 21
データ整合性を検証する 22
よくある質問 23
DynamoDB で作成できる最大テーブルサイズはどのくらいですか? 23
アカウントごとにいくつのテーブルを作成できますか? 23
DynamoDB テーブルにはいくつのグローバルセカンダリインデックスを作成できますか? 23
1 回の取引で追加または変更できる項目はいくつですか? 23
次のステップとリソース 24

ドキュメント履歴 25
用語集 26
..... 26
A 27
B 29
C 31
D 34
E 38
F 40
G 42
H 42
I 44
L 46
M 47
O 51
P 54
Q 56
R 57
S 59
T 63
U 64
V 65
W 65
Z 66
..... lxviii

RDBMS から Amazon DynamoDB への移行によるアプリケーションのモダナイゼーション

Ramkumar Ramanujam と Mahesh Kumar Vemula、Amazon Web Services (AWS)

2023 年 2 月 (ドキュメント履歴)

組織が事業を拡大するにつれて、情報システムへの負荷は大幅に増加します。パフォーマンスの最適化対策は、この増加する需要への対応にはある程度しか役立ちません。多くの場合、需要の増加により、データベースが負荷を処理できなくなることがあります。この問題は、リレーショナルデータベース管理システム (RDBMS) 上に構築されたアプリケーションで特に発生します。主要な RDBMS プロバイダは、リードレプリカ、データベースミラーリング、プライマリ/セカンダリ構成などの機能を通じてシステム負荷を処理する方法を提供していますが、これらの機能のコストとライセンス要件が問題になる可能性があります。データストレージの代替案を検討している組織向けに、このガイドでは RDBMS から Amazon DynamoDB への移行に焦点を当てています。

このガイドの最初の部分では、DynamoDB の機能と利点の概要を説明します。このガイドの第 2 部は、RDBMS (Microsoft SQL Server) から DynamoDB に移行されたアプリケーションのケーススタディに基づいています。リレーショナルデータを DynamoDB のドキュメント構造とキー値の収集にマッピングすること、もう 1 つは DynamoDB での作成、読み取り、更新、削除 (CRUD) 操作をサポートするようにアプリケーションのデータアクセスレイヤーを変更することの 2 つの移行上の課題に対処するコード例を示しています。

このガイドは、RDBMS システムを DynamoDB に移行してラピッドアプリケーション開発 (RAD) と高性能の要件に対応することを計画しているプログラムまたはプロジェクトマネージャー、データベース管理者、データベースアーキテクトを対象としています。このガイドは、リレーショナルデータベースと NoSQL の概念の基本的な理解を前提としていますが、DynamoDB のスキルや経験は必要ありません。

DynamoDB の概要

Amazon DynamoDB は、シームレスなスケーラビリティで高速で予測可能なパフォーマンスを提供するキーSQLバリューなしのドキュメントデータベースです。フルマネージド、マルチリージョン、マルチアクティブ、耐久性のあるデータベースです。(DynamoDB のグローバルテーブル機能は、1 つの AWS リージョンで行われた変更を他のすべての選択したリージョンに自動的に同期し、マルチアクティブサポートを提供します。) DynamoDB には、セキュリティ機能、バックアップおよび復元オプション、インターネット規模のアプリケーション用のインメモリキャッシュが組み込まれています。

NoSQL Database のスキーマレスな性質により、データベース (リードスキーマ) の変更を本番環境にプッシュするために必要な時間とプロセスが短縮され、迅速なアプリケーション開発 () が可能になりますRAD。DynamoDB などのSQLデータベースは、高性能の読み取り/書き込みオペレーション用に構築されていません。

特徴と利点

DynamoDB には、以下の機能と利点があります。

- 管理するサーバーなし – DynamoDB はフルマネージドの NoSQL データベースサービスです。つまり、サーバーのメンテナンスオーバーヘッドは発生しません。
- Schemaless – DynamoDB は、迅速なアプリケーション開発とデプロイをサポートします。
- 大規模なパフォーマンス – DynamoDB は、シームレスなスケーラビリティを備えた高速で予測可能なパフォーマンスを提供します。
- ACID サポート – DynamoDB は、データの正確性を維持するために、原子性、整合性、分離、耐久性 (ACID) トランザクションをサポートしています。
- 高可用性と耐久性 – データはソリッドステートディスク (SSDs) に保存され、リージョン AWS 内の複数のアベイラビリティーゾーンに自動的にレプリケートされるため、高可用性とデータの耐久性が組み込まれています。
- 自動スケーリング – DynamoDB は AWS Application Auto Scaling サービスを使用して、トラフィックパターンに応じてプロビジョニングされたスループット容量を動的に調整します。
- 柔軟な料金オプション – DynamoDB は、オンデマンドキャパシティモードとプロビジョニングされたキャパシティモードの 2 つのキャパシティモードに特定の請求オプションを提供します。

- Point-in-time リカバリ – リカバリを使用して point-in-time継続的なバックアップを有効にして、テーブルを誤って書き込みまたは削除操作から保護し、過去 35 日間の任意の時点にテーブルを復元できます。
- Time to Live (TTL) – 指定した期間後に DynamoDB テーブルから項目を自動的に削除できます。
- グローバルテーブル – 独自のレプリケーションソリューションを構築することなく、複数のレプリカを AWS リージョンにデプロイできます。
- グローバルセカンダリインデックス – DynamoDB テーブルのクエリは、テーブル自体のパーティションキーとソートキーとは異なるパーティションキーとソートキーを使用して実行できます。
- DAX – DynamoDB Accelerator (DAX) キャッシュサービスは、読み取りオペレーションにミリ秒未満の応答時間を提供します。
- DynamoDB Streams – この機能は、DynamoDB テーブルの変更のリアルタイム追跡と通知をサポートするために、ログに項目レベルの変更の時系列を提供します。

これらの利点の詳細については、 AWS ウェブサイトの「Amazon DynamoDB の機能」を参照してください。以下のセクションでは、 から DynamoDB RDBMSに移行することで、データベースワークロードのモダナイズに関連する機能の一部について説明します。

パーティションキー

DynamoDB はスキーマレスであるため、テーブルのすべての属性を定義する必要はありません。パーティションキー属性は必須です。ソートキーはオプションです。残りの属性は任意で、項目ごとに異なる場合があります。頻繁にアクセスされる項目が同じパーティションに存在しないように、カーディナリティの高いパーティションキーを選択することをお勧めします。この練習は、データアクセスの不均衡やホットパーティションを回避するのに役立ちます。詳細については、DynamoDB ドキュメントの「パーティションキーを効果的に設計して使用するためのベストプラクティス」を参照してください。

インデックス

インデックスを使用すると、代替クエリパターンにアクセスでき、クエリを高速化できます。リレーショナルデータベースと DynamoDB のどちらを使用しているかにかかわらず、インデックスは慎重に作成する必要があります。テーブルに対して書き込みオペレーションが発生するたびに、そのテーブルのすべてのインデックスを更新する必要があります。

グローバルセカンダリインデックスには、ベーステーブルから選択した属性が含まれていますが、テーブル自体のプライマリキーとは異なるプライマリキーによって編成されます。DynamoDB では、グローバルセカンダリインデックスはデフォルトでスパースです。つまり、ソートキーは

オプションであり、すべてのテーブル項目に表示されるわけではありません。この機能を利用するには、必要な属性のみを保存および投影するグローバルセカンダリインデックスを作成できます。DynamoDB テーブルには、最大 20 個のグローバルセカンダリインデックスを設定できます。この機能の詳細については、DynamoDB ドキュメントの「DynamoDB でのグローバルセカンダリインデックスの使用」を参照してください。 DynamoDB

有効期限

DynamoDB テーブルの Time to Live (TTL) プロパティを設定して、項目ごとに (レコード) タイムスタンプを定義し、項目が不要になったタイミングを指定できます。指定されたタイムスタンプの直後、DynamoDB は追加のキャパシティユニットを消費することなく、テーブルから項目を削除します。この機能の詳細については、DynamoDB ドキュメントの「DynamoDB Time to Live」を使用してアイテムの有効期限が切れる「」を参照してください。 DynamoDB

料金モデル

DynamoDB には、プロビジョニングされた容量とオンデマンド容量の 2 つの料金モデルがあります。料金モデルの選択は、予測されるワークロードによって異なります。

料金モデル	ワークロードタイプ	コスト	読み取り/書き込みスループット
プロビジョンドキャパシティー	予測可能	Lower	読み取りキャパシティーユニット (RCUs) と書き込みキャパシティーユニット () の点で、1 秒あたりの読み取り/書き込みオペレーションの数を指定しますWCUs。例: • 最大 4 KB の項目の場合、1 RCUは最終的に 1 秒あたり 2 回の一貫した読み取りを実行できます。

料金モデル	ワークロードタイプ	コスト	読み取り/書き込みスループット
			• 最大 1 KB の項目の場合、1 WCUは最終的に 1 秒あたり 2 回の一貫した読み取りを実行できます。 自動スケーリングを有効にして、トラフィックの変化に応じて容量を調整できます。

料金モデル	ワークロードタイプ	コスト	読み取り/書き込みスループット
オンデマンドキャパシティ	動的	より高い	スループット要件は指定しません。DynamoDB はワークロードを自動的に対応します。 読み取りリクエストユニットと書き込みリクエストユニットの観点から、アプリケーションがテーブルで実行する読み取りと書き込みには料金がかかります。例: • 8 KB 項目には、最終的に一貫した読み取りには 1 つの読み取りリクエストユニット、トランザクション以外の書き込みオペレーションには 8 つの書き込みリクエストユニットが必要です。

これらの 2 つのモデルの詳細については、DynamoDB ドキュメントの「読み取り/書き込み容量モード」を参照してください。

トランザクション

DynamoDB は、1 つの AWS アカウントと AWS リージョン内の 1 つ以上のテーブルにわたる原子力、整合性、分離、耐久性 (ACID) トランザクションをサポートします。

テーブル内およびテーブル間で複数の項目に対する変更を管理するには、DynamoDB トランザクション TransactWriteItems と TransactGetItems を使用できますAPIs。

- TransactWriteItems は、1 つ以上の PutItem、、および DeleteItemアクションを含む書き込みセットを含むバッチオペレーションです。 TransactWriteItemsはオプションでUpdateItem、更新を行う前に満たす必要がある前提条件をチェックできます。これらの条件には、書き込みセット内の項目と同じ項目、または異なる項目が含まれる場合があります。条件が満たされない場合、トランザクションは拒否されます。
- TransactGetItems は、1 つ以上のGetItemアクションを含むリードセットを含むバッチオペレーションです。アクティブな書き込みトランザクションの一部である項目に対してTransactGetItemsリクエストを発行すると、読み取りトランザクションはキャンセルされます。以前にコミットされた値を取得するには、標準の読み取りオペレーションを使用できます。

これらの の詳細についてはAPIs、DynamoDB ドキュメント」の「Amazon DynamoDB トランザクション: その仕組み」を参照してください。 DynamoDB

制約事項

DynamoDB トランザクションAPIオペレーションには以下の制約があります。

- トランザクションは 100 個を超える一意の項目を更新することはできません。
- トランザクションには、4 MB を超えるデータを含めることはできません。
- トランザクション内の 2 つのアクションを、同じテーブルの同じ項目に対して実行することはできません。例えば、1 つのトランザクションで同じ項目に対して ConditionCheck と の両方のUpdateアクションを実行することはできません。
- トランザクションは、複数の AWS アカウントまたはリージョンのテーブルでは動作できません。
- 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実行される AWS リージョン内でのみACID保証を提供します。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。
- オブジェクト永続性モデルはトランザクションをサポートしていません。トランザクション機能を使用するには、DynamoDB 低レベル APIを使用してデータベースとテーブルにアクセスする必要があります。

大きなコンポーネント

DynamoDB のサイズ制限は、項目ごとに 400 KB です。この制限には、属性名 (UTF-8 エンコーディングのバイナリ長) と属性値 (バイナリ長) の両方が含まれます。属性名はサイズ制限にカウントされます。例えば、値「IN」を持つ「国コード」という名前の属性と値「91」を持つcountry-phone-prefix「」という名前の属性の 2 つの属性を持つ項目を考えてみましょう。その項目の合計サイズは 36 バイトです。

回避策

項目が多くの属性とプロパティ、または大量のデータに関連付けられている場合、そのサイズは 400 KB を超える可能性があります。この場合、シリアル化された項目を Amazon Simple Storage Service (Amazon S3) JSON形式で保存し、Amazon S3 の場所を項目に属性 (`S3Location`) として保存できます。その項目の読み取りおよび書き込みオペレーションは S3 オブジェクトを取得し、JSON文字列を更新します。ローカルインデックスとグローバルセカンダリインデックスで使用されるプライマリキー、ソートキー、およびすべての属性は、 `S3Location` 属性とともにテーブルに保存する必要があります。これには、 `S3Location` 属性をチェックし、Amazon S3 から完全な項目データを取得するために、アプリケーション (データアクセスレイヤー) に追加のロジックが必要です。

バックアップと復元

バックアップと復元のサポートは、どのデータベースでも一般的な機能です。DynamoDB は、同じアカウント内のバックアップおよび復元オペレーションをネイティブにサポートしますが、他のオプションまたはプロセスを使用して、複数のアカウント間でテーブルコピーを実行できます。これらのプロセスでは、読み取り/書き込みキャパシティーユニットは消費されません。詳細については、AWS 「規範ガイダンス」ウェブサイトの「Amazon DynamoDB のクロスアカウントフルテーブルコピーオプション」ガイドを参照してください。

制約事項

DynamoDB は現在、 を使用してクロスアカウントバックアップと復元をサポートしていますがAWS Backup、アカウントは同じ組織の一部である必要があります。この制限に対処するには、次のいずれかのソリューションを使用します。

- を使用してNET、任意のプログラミング言語 (.、Java、Python など) でカスタム実装しますAWS SDK。アカウント A のソーステーブルから項目をスキャンし、アカウント B のテーブルに項目 (`BatchWrite`) を書き込むことができます。このコードは、サーバー、オンプレミスコンピュータ、または で実行できます AWS Lambda （ データベースが小さく、スクリプトの実行に 15 分未

満の時間がある場合) 。詳細については、 AWS 「規範ガイダンス」ウェブサイトの「カスタム実装を使用したアカウント間の Amazon DynamoDB テーブルのコピー」パターンを参照してください。

- の使用 AWS Glue。このオプションの詳細については、 AWS 「規範ガイダンス」ウェブサイトの「Amazon DynamoDB のクロスアカウントフルテーブルコピーオプション」ガイドを参照してください。

言語とSDKサポート

は、 AWS .NET、Java、、Node.js JavaScript、Python、、および Ruby のサービスPHPおよびサポートへのシンプルなプログラミングインターフェイスAWS SDKsを提供します。

オブジェクト永続性モデル (高レベルインターフェイス) 、ドキュメントインターフェイス、低レベルインターフェイス AWS SDKの 3 つのパターンから選択して、DynamoDB テーブルにアクセスできます。詳細については、このガイドの後半の「アクセスパターン」を参照してください。

サンプルアプリケーション

このセクションでは、リレーショナルデータベース管理システム (RDBMS) から NoSQL データベースへの移行を評価しているチーム向けのガイダンスを提供し、ターゲット NoSQL データベースとして Amazon DynamoDB に焦点を当てています。Microsoft SQL Server から DynamoDB に移行したアプリケーションのケーススタディに基づいて、次の 2 つの課題に対処しています。

- RDBMS 内の複数のテーブルのリレーショナルデータを DynamoDB のドキュメント構造とキーバリューコレクションにマッピングする
- DynamoDB での作成、読み取り、更新、削除 (CRUD) のオペレーションを実行するためのアプリケーション内のデータアクセスレイヤーの変更

ディスカッションとガイダンスには、AWS SDK for .NET を使用して C# で記述されたコード例が含まれています。

サンプル Web アプリケーションには、各アプリケーションで許可されているユーザーとホスト (Web、モバイル、デスクトップ)、メタデータ、検索キーワードなど、組織で使用される何百ものアプリケーションの構成が管理されています。このアプリケーションは、組織で使用されるさまざまなアプリケーションのさまざまなバージョンの構成、管理、および検索機能を提供します。構成の変更は、監査テーブルを使用して追跡されます。サンプルアプリケーションの一般的なワークフローは次のとおりです。

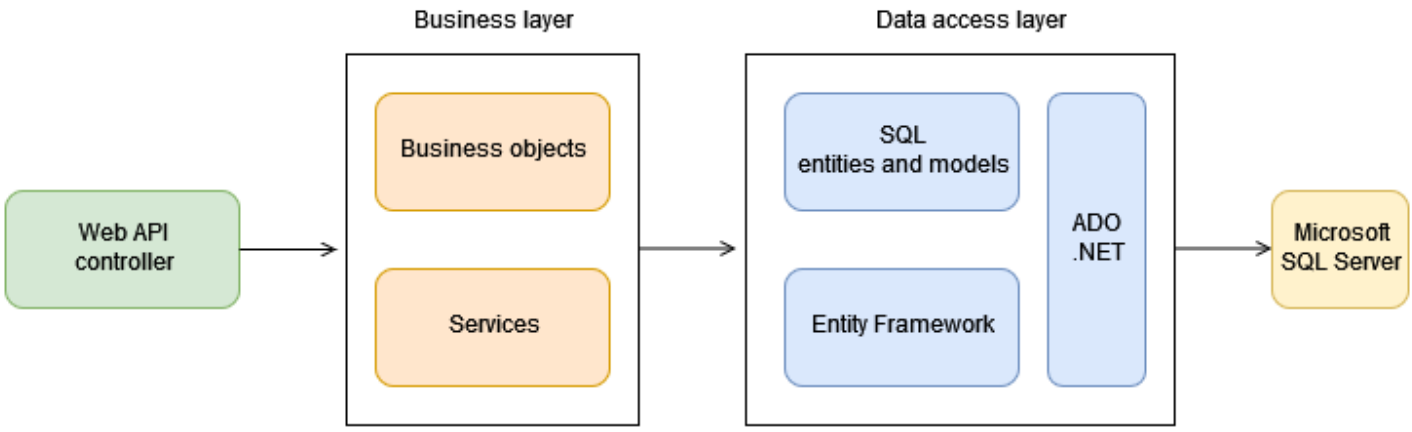
1. テストアプリケーションの設定を作成します。
2. テストアプリケーションの構成を本番環境に昇格させます (つまり、本番アプリケーションの構成を作成します)。
3. 変更の更新と監査 (監査レコードの作成、変更されたアプリケーション構成の呼び出し)

古いデータアクセスパターン

ソーステクノロジスタックは、次のもので構成されます。

- ASP.NET Web API Controller
- ビジネスオブジェクト
- ASP.NET エンティティフレームワーク (EF)
- ADO.NET データサービス

- Microsoft SQL Server 2016

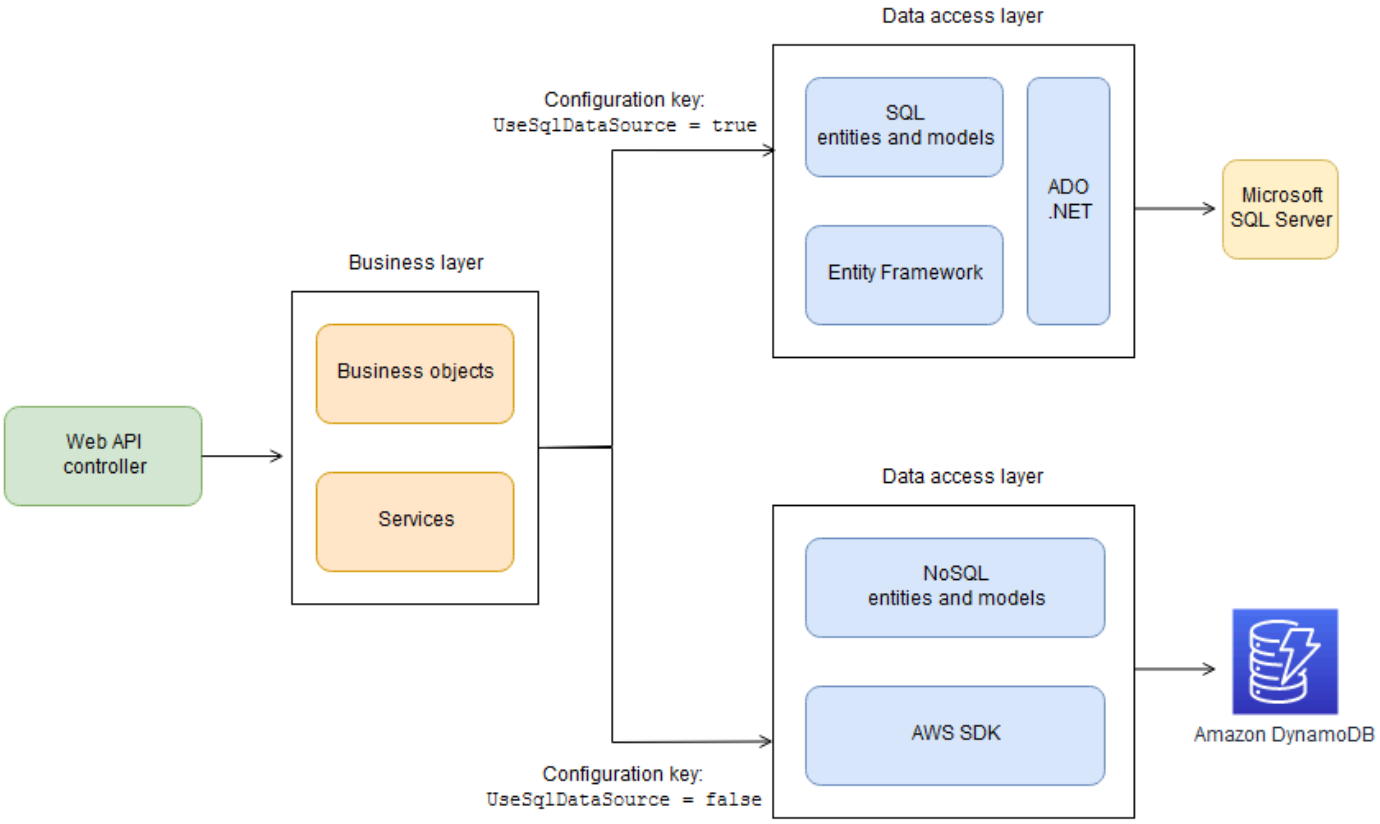


新しいデータアクセスパターン

移行したアプリケーションは、設定ファイルで提供される設定キー (`UseSqlDataSource`) に基づいて SQL Server と DynamoDB の両方をサポートします。次の図に示すように、`UseSqlDataSource`の値がの場合`true`、アプリケーションは SQL Server に接続します。値がの場合`false`、アプリケーションは DynamoDB に接続します。

新しいテクノロジースタックは、次のもので構成されます。

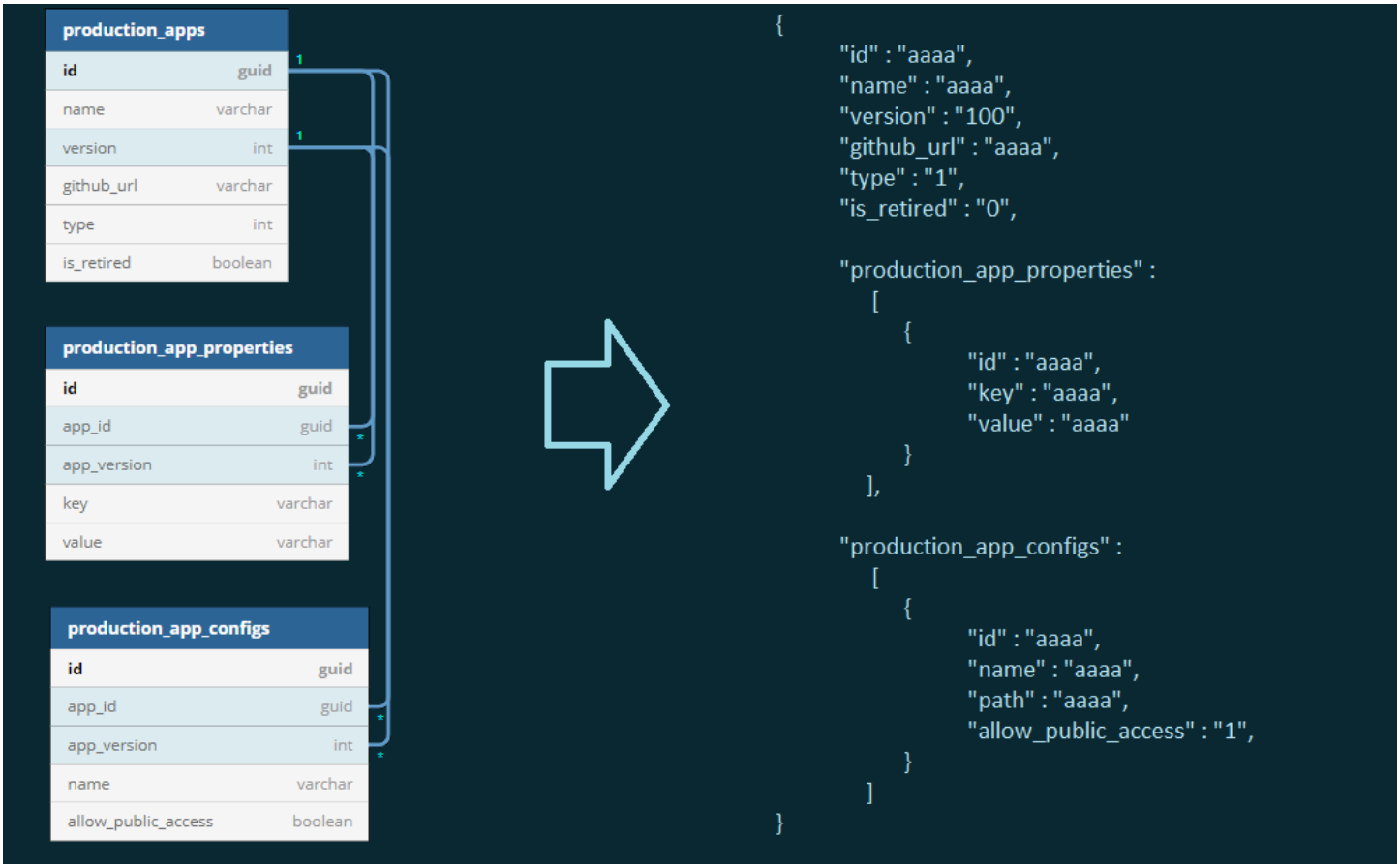
- ASP.NET ウェブ API コントローラー — さまざまな API エンドポイント経由の HTTP リクエストを受け付けます。
- ビジネスオブジェクトとサービス — データベースから取得した入力とデータを処理するビジネスロジックを含むクラスとオブジェクト。
- NoSQL エンティティとモデル — DynamoDB に保存されているアイテムにマップするクラス。
- AWSSDK — DynamoDBAWS やその他のサービスへのプログラムによるアクセスを提供します。
- DynamoDB — アプリケーションデータを保存するためのデータベース。



RDBMS テーブルスキーマとマッピング

次の図は、ソース RDBMS スキーマのテーブルと関係を示しています。

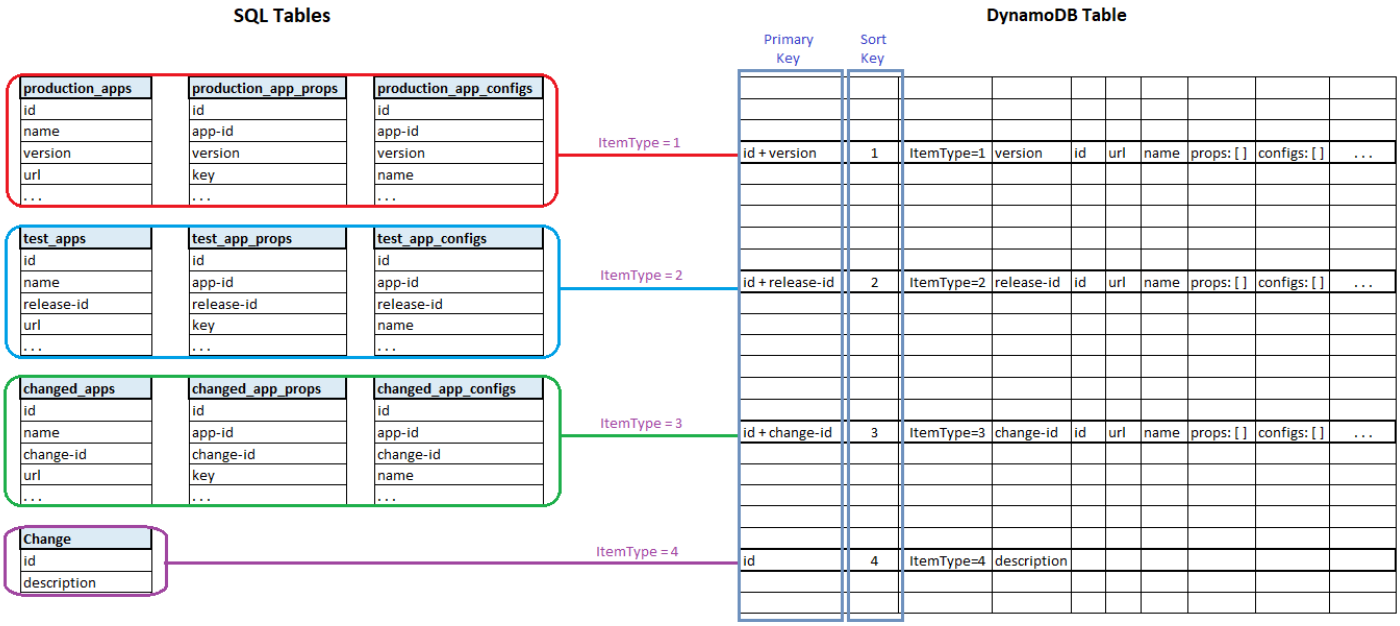
前の図で示したように、production_appsidversionテーブルにはと列があり、 one-to-many production_app_propertiesproduction_app_configsそれらはとのテーブルと関係があります。そのため、DynamoDB 設計では、次の JSON コードに示すようにproduction_app item、production_app_propertiesproduction_app_configsとテーブルがに埋め込まれています。production_app_configsとは複数の値を持つことができるためproduction_app_properties、これらのテーブルは JSON コードに配列として追加されます。changed_appstest_appsとテーブルも同様にマッピングされます。



1 つのテーブルデザイン

DynamoDB はリレーションシップを維持しません。固定テーブルスキーマをサポートします。そのため、さまざまなタイプのアイテム (SQL テーブルなど) を、アイテムのタイプを識別する属性 (`ItemType`) を使用して 1 つの DynamoDB テーブルに格納できます。

DynamoDB では、パーティションキー (PK) とソートキー (SK) の組み合わせは一意でなければならないため、これらのキーはアイテムタイプによって異なります。



グローバルセカンダリインデックス

インデックスは、データをより迅速に取得し、アプリケーションのパフォーマンスを向上させるのに役立ちます。サンプルアプリケーションでは、次のインデックスが作成されました。PKとSKは、どのように異なるアイテムを識別できるかに基づいて選択されました。

索引名	説明	パーティションキー (PK)	ソートキー (SK)	プロジェクト属性
`Version-index`	version特定のプロダクションアプリケーションをすべて取得します。	`version`		`id, name`
`Release-index`	release-id 特定のテストアプリケーションをすべて取得します。	`release-id`		`id, name`

索引名	説明	パーティションキー (PK)	ソートキー (SK)	プロジェクト属性
Change-index	に関連付けられた (変更された)change-id アプリケーションをすべて取得します。	change-id		id, modified-by, date

アクセスパターン

このガイドで前述したように、DynamoDB テーブルに対する作成、読み取り、更新、削除 (CRUD) オペレーションを実行する際の変更には、オブジェクト永続性インターフェイス、ドキュメントインターフェイス、低レベル API インターフェイスの 3 つのアクセスパターンから選択できます。以下のセクションでは、各インターフェイスについて説明します。SQL Server から DynamoDB へのユースケースでは、シンプルさ、読みやすさ、メンテナンスのしやすさを考慮して、オブジェクト永続性インターフェイスを選択しました。

オブジェクト永続性インターフェイス

オブジェクト永続性インターフェイスは、Entity Framework エンティティと同様に、.NET モデルを使用して DynamoDB アイテムの CRUD 操作を実行するための高レベルの抽象化されたアクセスメカニズムを提供します。インターフェイスプロパティは DynamoDB アイテム属性にマップされます。AWSSDK for .NET は、このモデルのカスタムプロパティ属性をサポートして、個々のプロパティのシリアル化と逆シリアル化のカスタマイズ、NULL 値の処理、および型変換を行います。

アプリケーションで使用されているサンプルモデル:

```
[DynamoDBTable(“AppLibrary")]
public class ProdApp
{
      [DynamoDBHashKey]
      public string PK { get; set; }     //Partition key

      [DynamoDBRangeKey]
      public string SK { get; set; }     //Sort key
```

```
      [DynamoDBGlobalSecondaryIndexRangeKey("Version-index")]
      [DynamoDBProperty]
      public int Version { get; set; }
       . . .
      [DynamoDBProperty]
      public Int64 TTL { get; set; }
}
```

アイテムへのアクセス:

```
var _dynamoDbClient = new AmazonDynamoDBClient(AWSCredentials);
var _context = new DynamoDBContext(_dynamoDbClient);

public ProdApp GetProdAppById (Guid id, int version)
{
     var pk = $"{id}-{version}";
     return _context.Load<ProdApp>(pk, ItemType.ProductionApplication);
}
```

詳細については、DynamoDB ドキュメントの「オブジェクト永続性インターフェイス」を参照してください。

ドキュメントインターフェイス

ドキュメントインターフェイスモデルでは、DynamoDBXMLDocument アイテムへのドキュメントベースのアクセス (.NET と同様) が提供されます。このモデルは上位レベルのプログラミングインターフェイスを提供しますが、その呼び出しを低レベルの API に変換して操作を実行します。

```
var _dynamoDbClient = new AmazonDynamoDBClient(AWSCredentials);
var _table = Table.LoadTable(_dynamoDbClient, "AppLibrary");

public ProdApp GetProdAppById (Guid id, int version)
{
      var pk = $"{id}-{version}";
     var doc = _table.GetItem(pk, ItemType.ProductionApplication);
      var app = new ProdApp {
          PK = doc["PK"],
          SK = doc["SK"],
          Version = doc["Version"],
          . . .
      };
      return app;
```

```
 }
```

詳細については、DynamoDB ドキュメントの「ドキュメントインターフェイス」を参照してください。

低レベル API

DynamoDB 用AWS SDK では、、、DeleteItemメソッドを使用して CRUD オペレーションを実行するための低レベルの API アクセスも提供します。PutItemGetItemUpdateItemこのモデルでは、属性マッピングとタイプ変換を完全に制御できます。これらの呼び出しに対する応答は、キーと値のペアの辞書です。

```
[DynamoDBTable(“AppLibrary")]
public class ProdApp
{
        [DynamoDBHashKey]
        public string PK { get; set; }     //Partition key

        [DynamoDBRangeKey]
         public string SK { get; set; }     //Sort key

        [DynamoDBGlobalSecondaryIndexRangeKey(“Version-index")]
        [DynamoDBProperty]
        public int Version { get; set; }
              . . .
        [DynamoDBProperty]
        public ProdConfig Config  { get; set; }
}

var _dynamoDbClient = new AmazonDynamoDBClient(AWSCredentials);

public ProdApp GetProdAppById (Guid id, int version)
{
    var pk = $”{id}-{version}”;
    var resp = _dynamoDbClient.Query(queryRequest);
    var item = resp.Items[0];
    var app = new ProdApp {
         PK = item[“PK”].S,
         SK = item[“SK”].S,
         Version = Convert.ToInt32(item[“Version”].S),
         . . .
         Config = new ProdConfig {
```

```
                Name = item[“Config”].M[“Name”].S,
                Id = Conver.ToInt32(item[“Config”].M[“Id”].S)
            }
    };
    return app;
}
```

詳細については、DynamoDB ドキュメントの「低レベルインターフェイス」を参照してください。

コンバーター

DynamoDB データベースを保存または読み取る際に、データを変更または変換しなければならない場合があります。このようなシナリオでは、Amazon.DynamoDBV2 の IPropertyConverter インターフェイスを使用できます。 DataModel以下のようなコードを使用して、名前空間

```
// Converts the null values of a string property to a valid string and vice versa.
 public class NullOrStringConverter : IPropertyConverter
 {
     // Called when creating the JSON / DynamoDB item from the model
     public DynamoDBEntry ToEntry(object value)
     {
         var entry = new Primitive
         {
               value = new DynamoDBNull()
         };
         if(value != null)
         {
               entry.Value = value.ToString();
         }
         return entry;
     }
    // Called when populating the model from the JSON / DynamoDB item
     public object FromEntry(DynamoDBEntry entry)
     {
         if(entry is DynamoDBNull)
         {
               return string.Empty;
         }
         else
         {
               return entry.ToString();
```

```
            }
        }
    }
```

モデルでのコンバーターの使用方法:

```
[DynamoDBTable(“AppLibrary")]
public class ProdApp
{
        . . .

        [DynamoDBProperty (typeof(NullOrString))]
        public string AppConfigId { get; set; }
         . . .
}
```

ベストプラクティス

このセクションでは、前のセクションで説明したベストプラクティス (400 KB を超えるアイテムを Amazon S3 に保存したり、インデックス、単一テーブルデザイン、トランザクションを使用したりするなど) に加えて推奨事項をまとめます。

オブジェクト永続性アクセスパターンを使用する

このガイドの前半で説明したように、Amazon DynamoDB には 3 つのアクセスパターンがあります。 easy-to-maintain オブジェクト・パーシスタンス・インターフェースはクリーンでコード化も容易です。読み取り/書き込み操作中にモデルプロパティ値をカスタマイズまたは変換する必要がない限り、オブジェクト永続性インターフェイスを使用することをお勧めします。

適切なキャパシティプロビジョニングモードを選択してください

オンデマンドキャパシティプロビジョニングは、ワークロードの増減に合わせて読み取り操作と書き込み操作を自動的にスケーリングします。ワークロードが予測できない場合は、このモードを使用することをお勧めします。通常、コストはプロビジョニングキャパシティモードよりも高く、 pay-as-you-use ベースで課金されます。ワークロードが予測可能で、キャパシティ要件を予測できる場合は、プロビジョニングキャパシティモードを使用することをお勧めします。詳細については、このガイドの前半の「価格モデル」セクションを参照してください。

キャッシュを使う

DynamoDB を使用する場合は、各読み取り/書き込み操作に関連するコストを削減するためにキャッシュを使用することをお勧めします。キャッシュされたデータが古くなった場合は、適切な無効化ロジックを使用してキャッシュからアイテムを削除します。キャッシュを実装するために最も頻繁に使用されるエンドポイントを特定します。

スキャンの代わりにクエリを使う

DynamoDB スキャンは可能な限り避けてください。DynamoDB クエリは、スキャン操作よりも効率的でコストもかかりません。クエリはパーティションキー (PK) とソートキー (SK) の値に基づいてアイテムをフィルタリングしますが、スキャンでは指定されたパラメータに基づいてアイテムをフィルタリングするためにすべてのレコードを読み取る必要があります。DynamoDB の料金はデー

タの読み取り/書き込み操作の量に基づいているため、スキャンの方がクエリよりもコストがかかります。クエリも高速になり、最終的にはアプリケーションのパフォーマンスが向上します。

データ整合性を検証する

DynamoDB は NoSQL データベースであるため、リレーションシップデータを維持したり、データ整合性の制約を含めたりしません。各項目には、主キーとソートキーの組み合わせのみが必要です。DynamoDB テーブル内の関連項目間のデータ整合性を確保するために、システムのアプリケーションまたはビジネスレイヤーで厳密な検証を実行することをお勧めします。

よくある質問

このセクションでは、DynamoDB の使用についてよく寄せられる質問への回答を提供します。

DynamoDB で作成できる最大テーブルサイズはどのくらいですか?

テーブルのサイズや作成できる列の数に制限はありません。

アカウントごとにいくつのテーブルを作成できますか?

AWS リージョンアカウントごとに、最大 2500 のテーブルを作成できます。さらにテーブルを作成したい場合は、https://aws.amazon.com/support でサービスクォータ増加を申請できます。

DynamoDB テーブルにはいくつのグローバルセカンダリインデックスを作成できますか?

初期クォータとして、テーブルごとに 20 個のグローバルセカンダリインデックスがあります。さらにインデックスを作成したい場合は、https://aws.amazon.com/support でサービスクォータ増加を申請できます。

1 回の取引で追加または変更できる項目はいくつですか?

1 回のトランザクションで最大 100 項目 (または 4 MB のデータ) を追加または変更できます。テーブルに 100 件を超えるレコードを書き込む場合は、バッチ書き込み操作を使用できます。

クォータの詳細なリストについては、DynamoDB ドキュメントの「Amazon DynamoDB のサービス、アカウント、テーブルのクォータ」を参照してください。

次のステップとリソース

Amazon DynamoDB は、ハイパフォーマンスな NoSQL データベースとして設計されています。低コスト、高性能、自動スケーリングなどの機能を備えているため、リレーショナルデータベースシステム (RDBMS) に代わる優れた選択肢です。ベストプラクティスに従い、インデックスを使用し、適切なパーティションキーを選択し、テーブル構造を慎重に設計することで、DynamoDB を最大限に活用できます。DynamoDB アクセラレータ (DAX) は高頻度の読み取り操作に使用でき、高速なインメモリパフォーマンスを活用できます。RDBMS に代わるものをお探しの場合は、コストとパフォーマンスのメリットから DynamoDB を検討してください。

DynamoDBの使用を開始するには、次のリンクを参照してください。

DynamoDB ドキュメンテーション

- NoSQL
- グローバルセカンダリインデックス
- トランザクション
- データアクセス — 低レベル API
- データアクセス — 文書モデル (中間レベル)
- データアクセス — オブジェクト永続性モデル (高レベル)
- 仕組み — 読み取りの一貫性
- 仕組み — 読み込み/書き込みモード
- 任意データのマッピング
- ベストプラクティス

AWS規範的ガイダンスの出版物

- Amazon DynamoDB によるデータのモデリング (ガイド)
- Amazon DynamoDB のクロスアカウント完全テーブルコピーオプション (ガイド)
- カスタム実装 (パターン) を使用してアカウント間で Amazon DynamoDB テーブルをコピーする

ドキュメント履歴

このガイドは、このドキュメントの大きな変更点をまとめたものです。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
更新された情報	トランザクション API 操作、バックアップと復元、よくある質問に関するセクションを更新しました。	2023年2月24日
初回刊行物	—	2021 年 9 月 30 日

AWS 規範ガイダンス用語集

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 Postgre SQL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にある程度の最適化を導入します。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: 顧客関係管理 (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インスタンスで Oracle に移行します AWS クラウド。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例: を移行する Microsoft Hyper-V へのアプリケーション 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、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。柔軟性はありますが、アクティブ/パッシブ移行よりも多くの作業が必要です。

アクティブ - パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、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

行のグループで動作し、グループの 1 つの戻り値を計算するSQL関数。集計関数の例には、SUMおよび が含まれます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。

人工知能オペレーション (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。移行戦略で AWS がどのようにAIOps使用されるかの詳細については、「 オペレーション統合ガイド」を参照してください。

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、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コントロール (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ABAC AWS「」の「」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安価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かつ効果的な計画を策定 AWS するのに役立つ 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または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 で動作グラフを使用して、失敗したログオン試行、疑わしいAPI呼び出し、および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。「endianness」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 つの別々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をインデックス化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、個人や組織に混乱や害を与えることを意図したものもあります。

ボットネット

マルウェアに感染し、ボットハーダーまたはボットオペレーターと呼ばれる 1 つ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 にアクセスするための簡単な手段を提供します。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 AWS 」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、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。

事業継続計画 (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 Cloud Adoption Framework を参照してください。

Canary のデプロイ

エンドユーザーへのバージョンのスローリリースと増分リリース。自信が持てば、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。

データキャプチャの変更 (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。は、同期を維持するために、ターゲットシステムの変更を監査したりレプリケートしたりするなど、CDCさまざまな目的で使用できます。

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、そのレスポンスを評価する実験を実行できます。

CI/CD

「継続的統合と継続的配信」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 AWS のサービス 受信する前に、データをローカルで暗号化します。

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

組織は通常、 に移行するときに 4 つのフェーズを実行します 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
- 基盤 – クラウド導入を拡大するための基盤投資 (ランディングゾーンの作成、 の定義CCoE、オペレーションモデルの確立など)
- 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旅」と「導入のステージ」で、Stephen Orban によって定義されました。 AWS 移行戦略との関連性については、「移行準備ガイド」を参照してください。

CMDB

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以下が含まれます。GitHub または Bitbucket Cloud。 コードの各バージョンはブランチ と呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

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

コールドデータ

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

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージや動画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、 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 AWS Panorama 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、設定は想定された状態から変更されます。ワークロードが非準拠になる可能性があり、通常、段階的かつ意図的ではありません。

設定管理データベース (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、移行のCMDBポートフォリオ検出および分析段階で のデータを使用します。

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、または組織全体に単一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 is commonly described as a pipeline. CI/CD は、プロセスの自動化、生産性の向上、コード品質の向上、迅速な提供に役立ちます。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

一元的な管理とガバナンスで分散された分散データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

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

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「 でデータ境界を構築する 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、クエリや分析に使用されます。

データベース定義言語 (DDL)

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

データベース操作言語 (DML)

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

DDL

データベース定義言語 を参照してください。

ディープアンサンブル

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

ディープラーニング

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

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 に採用する場合 AWS、リソースの保護に役立つように、 AWS Organizations 構造のさまざまなレイヤーに複数のコントロールを追加します。例えば、アプローチでは defense-in-depth、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、リーンな製造プラクティス用に最初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ます。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番号です。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、「 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。 AWS

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)。ストラングルラーの fig パターンでドメイン駆動型設計を使用する方法については、「従来の Microsoft のモダナイズ」を参照してくださいASP。NET (ASMX) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に更新する「」。

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。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化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。

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、レスポンスタイムを向上させることができます。

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) 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、インターフェイスエンドポイントを作成することでVPC、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、アイデンティティとアクセスの管理、検出コントロール、インフラストラクチャのセキュリティ、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、サマリー統計を計算し、データ視覚化を作成することで実行されます。

F

ファクトテーブル

星スキーマ の中央テーブル。ビジネスオペレーションに関する定量的なデータを保存します。通常、ファクトテーブルには 2 つのタイプの列が含まれます。つまり、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です。

フェイルファースト

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

障害分離境界

では AWS クラウド、アベイラビリティーゾーン、コントロールプレーン AWS リージョン、データプレーンなどの境界が、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。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 Explanations (SHAP) や統合勾配など、さまざまな手法で計算できる数値スコアとして表されます。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コントロール (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

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

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分散の制限」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望ましいアプローチです。

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

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

H

HA

高可用性 を参照してください。

異種混在データベースの移行

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

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

ソースデータベースを、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に移行する (Microsoft SQL Server から Amazon RDS for SQL Server など) 。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、ホットフィックスは一般的な DevOps リリースワークフローの外部で行われます。

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

IIoT

「産業用モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

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

インバウンド (イングレス) VPC

AWS マルチアカウントアーキテクチャでは、 VPCが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、ルーティングします。AWS セキュリティリファレンスアーキテクチャでは、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インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

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、オートメーション、分析、AI/ML の進歩を通じて、製造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。

産業用モノのインターネット (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、「産業用モノのインターネット (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。

検査 VPC

AWS マルチアカウントアーキテクチャでは、 VPCs (同一または異なる 内の 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VPCを管理する一元化された です。AWS セキュリティリファレンスアーキテクチャでは、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インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

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

「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 の基盤を提供しますITSM。

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

ラベルベースのアクセスコントロール (LBAC)

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ている必須のアクセスコントロール (MAC) の実装。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「最小権限の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。「endianness」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たソフトウェア。マルウェアは、コンピュータシステムを混乱させたり、機密情報を漏洩したり、不正アクセスを受けたりする可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービス は、インフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を操作し、エンドポイントにアクセスしてデータを保存および取得します。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象サービス とも呼ばれます。

製造実行システム (MES)

原材料を作業現場の最終製品に変換する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。

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、ツールの採用を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作時にそれ自体を強化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

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 デバイスのパブリッシュ/サブスクライブパターンに基づく軽量 machine-to-machine (M2M) 通信プロトコル。

マイクロサービス

明確に定義された上で通信APIsし、通常、小規模な自己完結型チームが所有する、小規模で独立したサービス。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 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APIs。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。

移行促進プログラム (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、レガシー移行を系統的な方法で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して高速化するための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、スプリントに携わる 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 EC2を使用して Amazon への移行をリホストします。

移行ポートフォリオ評価 (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 (サーバーの適正サイズ、料金、TCO比較、移行コス

ト分析) と移行計画 (アプリケーションデータ分析とデータ収集、アプリケーショングループ化、移行の優先順位付け、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)

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 AWS CAF。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 の最初のフェーズです。

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「アプリケーションのモダナイズ戦略 AWS クラウド」を参照してください。

モダナイゼーション準備状況評価

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

モノリシックアプリケーション (モノリス)

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

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

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

多クラス分類

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

ミュータブルインフラストラクチャ

本稼働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 Framework AWS 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をベストプラクティスとして使用することを推奨しています。

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。

Open Process Communications - 統合アーキテクチャ (OPC-UA)

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

運用レベルの契約 (OLA)

サービスレベル契約 () をサポートするために、IT グループが相互に提供することを約束する機能を明確にする契約SLA。

運用準備状況のレビュー (ORR)

インシデントや潜在的な障害の範囲を理解、評価、防止、または軽減するのに役立つ質問とそれに関連するベストプラクティスのチェックリスト。詳細については、 AWS Well-Architected フレームワークの「Operational Readiness Reviews (ORR)」を参照してください。

運用テクノロジー (OT)

物理環境と連携して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ハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が Industry 4.0 変換の主要な焦点です。

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

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 を口グに記録する 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、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、変化の採用を加速し、移行に伴う問題に対処し、文化的および組織的な変化を推進することで、組織が新しいシステムや戦略の準備と移行を支援します。 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に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 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、および 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にアクセスを制限するオプションがあります。を使用する場合OAI、 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。「」も参照してください。これによりOAC、より詳細で拡張されたアクセスコントロールが提供されます。

ORR

「運用準備状況の確認」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 (出力) 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VPCを処理する です。AWS セキュリティリファレンスアーキテクチャでは、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インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、および検査でネットワークアカウントを設定することをお勧めします。

P

アクセス許可の境界

ユーザーまたはロールが持つことができる最大アクセス許可を設定するために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。例としてPIIは、名前、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。一般的には falseWHERE句にあります。

述語プッシュダウン

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

予防的コントロール

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

プリンシパル

アクションを実行し AWS 、リソースにアクセスできる のエンティティ。このエンティティは通常、、IAM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「ロールの用語と概念」を参照してください。 https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html#id_roles_terms-and-concepts

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

Amazon Route 53 が 1 つ以上の 内のドメインとそのサブドメインのDNSクエリに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VPCs。詳細については、Route 53 ドキュメントの「プライベートホストゾーンの使用」を参照してください。

プロアクティブコントロール

非準拠のリソースのデプロイを防ぐように設計されたセキュリティコントロール。これらのコントロールは、プロビジョニングされる前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、 AWS Control Tower ド

キュメントの「 コントロールリファレンスガイド」および「 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)

製品の設計、開発、発売から、成長と成熟、減少と削除に至るまでのライフサイクル全体にわたるデータとプロセスの管理。

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)

製造では、マシンをモニタリングし、製造プロセスを自動化する、信頼性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。

publish/subscribe (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で、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 ではMES、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公開できます。システムは、発行サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

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

クエリプランのリグレッション

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

R

RACI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

ランサムウェア

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

RASCI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

行と列のアクセスコントロール を参照してください。

リードレプリカ

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

再設計

7 Rs を参照してください。

復旧ポイントの目的 (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。耐障害性、安定性、耐障害性を提供するために、それぞれ AWS リージョン が分離され、他のものとは独立しています。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 Rs を参照してください。

プラットフォーム変更

7 Rs を参照してください。

再購入

7 Rs を参照してください。

回復性

中断に抵抗または復旧するアプリケーションの機能。高可用性とディザスタリカバリは、 で回復性を計画する場合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。

責任、説明責任、相談、情報 (RACI) 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、行列はRASCI行列 と呼ばれ、除外すると行RACI列 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。

行と列のアクセスコントロール (RCAC)

アクセスルールが定義されている基本的で柔軟なSQL式の使用。RCAC は、行のアクセス許可と列マスクで構成されます。

RPO

「復旧ポイントの目的」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) AWS Management Console が有効になるため、ユーザーは にログインしたり、 AWS API組織内のすべてのユーザーIAMに対して でユーザーを作成したりすることなく 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「2.0 SAML ベースのフェデレーションについて」を参照してください。

SCADA

「監視コントロールとデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

では AWS Secrets Manager、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 ???、およびプロアクティブ 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。

セキュリティ情報とイベント管理 (SIEM) システム

セキュリティ情報管理 (SIM) システムとセキュリティイベント管理 (SEM) 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、その他の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、アラートを生成します。

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、事前定義されたプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動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

送信先で、それ AWS のサービス を受信する によるデータの暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCPs ガードレールを定義するか、管理者がユーザーまたはロールに委任できるアクションの制限を設定します。を許可リストまたは拒否リストSCPsとして を使用して、許可または禁止されるサービスまたはアクションを指定できます。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URLの 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。

サービスレベル契約 (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)

サービスレベルインジケータ によって測定される、サービスの正常性を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について共有する責任を説明するモデル。 クラウドのセキュリティ AWS はクラウドのセキュリティに責任があり、クラウドのセキュリティはユーザーの責任です。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)

システムを混乱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

「サービスレベル契約」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「アプリケーションのモダナイズに対する段階的なアプローチ AWS クラウド」を参照してください。

SPOF

単一障害点 を参照してください。

スタースキーマ

1 つの大きなファクトテーブルを使用してトランザクションデータまたは測定データを保存し、1 つ以上の小さなディメンションテーブルを使用してデータ属性を保存するデータベース組織構造。この構造は、データウェアハウスまたはビジネスインテリジェンスの目的で使用するため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。このパターンを適用する方法の例については、「従来の Microsoft のモダナイズ」を参照してくださいASP。NET (ASMX) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に更新する「」を参照してください。

サブネット

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

監視コントロールとデータ取得 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと本番稼働をモニタリングするシステムです。

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

VPCs とオンプレミスの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トランジットハブ。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

組織内でタスクを実行するために指定したサービスに、ユーザーに代わって AWS Organizations および アカウントで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 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VPCsできる 2 つの間の接続。詳細については、Amazon VPCドキュメントのVPC「ピアリングとは」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

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

ウォームデータ

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

ウィンドウ関数

現在のレコードに何らかの形で関連する行のグループに対して計算を実行するSQL関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなど、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」を 1 回参照し、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。

1 回書き込み、多数読み取り (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。認定ユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャは、イミュータブル と見なされます。

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性 を利用する攻撃、通常はマルウェア。

ゼロデイ脆弱性

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

ゾンビアプリケーション

平均使用量CPUとメモリ使用量が 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。