AWS による優れた設計のフレームワーク

2015 年 10 月

注意

本書は情報提供のみを目的としています。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本文書の情報および AWS 製品の使用について独自に評価する責任を負うものとします。これらの情報は、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されるものです。本文書内のいかなるものも、AWS、その関係者、サプライヤ、またはライセンサーからの保証、表明、契約的なコミットメント、条件や確約を意味するものではありません。お客様に対する AWS の責任は AWS 契約によって規定されています。また、本文書は、AWS とお客様との間の契約に属するものではなく、また、当該契約が本文書によって修正されることもありません。

目次

要約

本書では **AWS による優れた設計のフレームワーク**について説明し、これによってお客様がそのクラウドベースのアーキテクチャを評価、改善し、またそのデザイン決定がビジネスに及ぼす影響をより良く理解できるようになります。ここでは、AWS による優れた設計のフレームワークの柱とされる 4 つの概念領域における一般的な設計の原則と、特定のベストプラクティスおよびガイダンスを紹介します。

はじめに

アマゾン ウェブ サービス (AWS) において、クラウドに信頼性、セキュリティ、効率、コスト効果が高いシステムを設計するために、アーキテクチャのベストプラクティスについてお客様に学習していただくことが重要だと考えます。この一環として、AWS でシステムを構築する際の選択肢の長所と短所の理解に役立つように、AWS による優れた設計のフレームワークを開発しました。システムを上手に構築することで、ビジネスが成功する可能性が飛躍的に増加すると確信しています。

AWS ソリューションアーキテクトは、多様なビジネスにおいて垂直およびユースケースにおけるソリューションの設計に長年の経験を持ち、何千という AWS のお客様のアーキテクチャの設計および検証のお手伝いをしてきました。この経験から、クラウドにおけるシステム設計のベストプラクティスおよび核となる戦略を見つけ出しました。AWS による優れた設計のフレームワークには、特定のアーキテクチャがクラウドのベストプラクティスにうまく合致しているかどうかを理解するための基礎質問集が記載されています。フレームワークは、現代のクラウドベースのシステムに期待する品質について整合性の面からシステムを評価するアプローチ方法、およびその品質を達成するために必要な対処を提供します。AWS プラットフォームは進化し続け、お客様との共同作業で学ぶことも尽きないため、優れた設計の定義も改良され続けます。

本書は技術責任者 (CTO)、設計者、開発者、オペレーションチームメンバーなどの技術担当者を対象としています。本書を読むことで、クラウドアーキテクチャを設計する際の AWS のベストプラクティスおよび取るべき戦略が理解できます。本書では、アーキテクチャパターンの実装については詳しく紹介しませんが、この情報に関する適切な資料へのリファレンスを含んでいます。

AWS による優れた設計のフレームワークの定義

AWS の専門家は、日々、お客様がクラウドのベストプラクティスの利点を活かせるようなシステムを構築できるようにお手伝いしています。設計が進化するにつれて発生するアーキテクチャとのトレードオフをお客様とともに考えてきました。ライブ環境にシステムをデプロイするたびに、システムがどの程度うまく機能するか、トレードオフの影響はどうかが判ります。

このように学んできたことをベースに、AWS による優れた設計のフレームワークは作成されました。アーキテクチャが AWS ベストプラクティスにどの程度合致しているかを評価できる質問集です。

AWS による優れた設計のフレームワークは、セキュリティ、信頼性、パフォーマンス効率、コスト最適化という 4 つの柱を基本としています。それぞれは以下のように定義されます。

柱名	説明
セキュリティ	リスクの評価と軽減戦略を通して、ビジネス価値を提供しながら、情報、システム、資産を保護する能力です。
信頼性	インフラストラクチャまたはサービスの障害から復旧したり、必要に応じて動的にコンピューティングリソースを獲得したり、設定ミスや一時的なネットワークの問題などによる障害を軽減したりといったシステムの能力です。
パフォーマンス効率	システムの要求に合わせてコンピューティングリソースを効率的に使用し、需要の変化や技術の進歩に合わせてこの効率を維持する能力です。
コストの最適化	不要なコストや最適でないリソースを回避または排除する能力です。

一般的な設計の原則

AWS による優れた設計のフレームワークでは、クラウドにおける適切な設計を可能にする一般的な設計の原則が判ります。

- **必要キャパシティーの推測をやめる**:必要なインフラストラクチャキャパシティーを予測する必要がなくなります。システムのデプロイに先立って性能を決定すると、高価で無駄なリソースが発生したり、機能が制限されてパフォーマンスに影響したりします。クラウドコンピューティングにはこのような問題はありません。必要な分のみキャパシティーを使用し、自動的にスケールアップまたはスケールダウンできます。
- **本稼働スケールでシステムをテストする**:従来の非クラウド環境では、通常、コストの問題でテストのためだけに複製環境を作成することはできません。従って、ほとんどのテスト環境は本稼働が要求するライブレベルではテストされていません。クラウドでは、複製環境を作成し、テストを完了してから、リソースを破棄できます。支払いはテスト環境を実行しているときのみ発生するため、僅かなコストでオンプレミスでのテストを行いライブ環境をシミュレートできます。
- **アーキテクチャ変更のリスクを低減する**:本稼働設定をエミュレートするテスト環境の作成を自動化できるため、簡単にテストを行うことができます。また、テストリソースを使用するためにチームが順番待ちをしなくてはならないオンプレミス環境で発生していたテストのシリアル化もなくなります。
- **自動化によってアーキテクチャ実験を簡単にする**:自動化によって、システムを低コスト（手動作業なし）で作成およびレプリケートできます。自動化に対する変更を追跡し、影響を監査して、必要な場合は以前のパラメーターに戻すことができます。
- **発展するアーキテクチャの許可**:従来の環境では、アーキテクチャは決定されると固定され、実装は 1 回きりであり、その寿命の間にいくつかシステムのメジャー・バージョンアップがあるだけです。ビジネスとそのコンテキストは変化し続けているのに、最初に決めたこれらのために、システムがビジネスの需要の変化に対応できない場合があります。クラウドでは、オンデマンドによる自動化とテスト機能によって、設計変更によって生じる影響のリスクを低減できます。これによって、システムが何度でも進化でき、新しいイノベーションの利点をスタンダードプラクティスとしてビジネスに活かすことができます。

AWSによる優れた設計のフレームワークの4 本柱

ソフトウェアシステムの作成はビルの建築に似ています。基礎がしっかりしていなければ、ビルの健全性や機能を損なう構造の問題が発生することがあります。技術ソリューションを設計する場合、セキュリティ、信頼性、パフォーマンス効率、コスト最適化の 4 本の柱を疎かにすると、意図したとおりにまた要件に従って稼働するシステムの構築が難しくなるでしょう。これらの柱をアーキテクチャに組み込むことで、安定した効率的なシステムを作成することができます。こうすることで、要求される機能など設計の他の要素に集中できます。

このセクションでは、4 本柱のそれぞれについて、定義、ベストプラクティス、疑問点、考慮点、および関連する AWS サービスなどを説明します。

セキュリティの柱

セキュリティの柱には、リスクの評価と軽減戦略を通して、ビジネス価値を提供しながら、情報、システム、資産を保護する能力が含まれます。

設計の原則

クラウドでは、システムセキュリティを強化する役に立つ数多くの原則があります。

- **すべてのレイヤーでセキュリティを適用**:インフラストラクチャの水際でセキュリティアプライアンス（例: ファイアウォール）を実行するだけではなく、ファイアウォールや他のセキュリティ制御をすべてのリソース（例: 各仮想サーバー、ロードバランサー、ネットワークサブネット）でも使用します。
- **追跡可能性を有効にする**:すべてのアクションと環境に対する変更を記録し監査します。
- **セキュリティイベントに対する応答を自動化する**:イベント駆動型または条件駆動型アラートをモニターし、応答を自動的にトリガーします。

- **システムの保護に集中する**:AWS 責任共有モデルを使用すると、AWS が安全なインフラストラクチャやサービスを提供するため、お客様はアプリケーションデータやオペレーティングシステムの保護に集中できます。
- **セキュリティのベストプラクティスを自動化する**:ソフトウェアベースのセキュリティメカニズムによって、より手早くコスト面で効果的なスケーリングをより安全に実行できます。仮想サーバーのカスタムベースラインイメージを作成して保存し、そのイメージを自動的にお客様が起動する各新規サーバーに使用します。テンプレートで定義され管理されるインフラストラクチャ全体を作成します。

定義

クラウドでのセキュリティは、次の 4 つの領域で構成されます。

1. データ保護
2. 権限管理
3. インフラストラクチャの保護
4. 発見的コントロール

AWS 責任共有モデルを使用して、クラウドを使用する企業がセキュリティおよびコンプライアンスの目的を達成できます。AWS では、クラウドサービスをサポートするインフラストラクチャを物理的に保護するため、AWS のお客様はサービスを使用して目的を達成することに集中できます。また、AWS クラウドでは、セキュリティデータへのアクセス機能が強化され、セキュリティイベントへの応答方法が自動化できます。

ベストプラクティス

データ保護

システムを構築する前に、セキュリティに影響を及ぼす基礎的なプラクティスが適切な場所に配置される必要があります。たとえば、データ分類は、企業のデータを重要度に基づいてカテゴリー分けする方法です。最小権限は、通常の機能の使用を許可しながらアクセスを可能な限り最低レベルに制限します。また、暗号化は、認証されていないアクセスでは解読できないようにデータをレンダリングすることでデータを保護します。これらのツールや技術は、財政的損失の防止や規則上の義務の遵守などの目的をサポートするため、重要です。

データ保護には、データの健全性を保持しながら機密に保つように設計されたコントロールやパターンの使用や、必要なときには確実にデータを使用できることも含まれます。

AWS では、次のようなプラクティスでデータの保護が実行されます。

- AWS のお客様は、お客様のデータに対する完全な統制を維持します。
- AWS ではデータの暗号化や定期的なキーローテーションを含むキーの管理がより簡単になります。これらは AWS によってネイティブに自動化することも、お客様が保守することもできます。
- ファイルアクセスや変更などの重要な内容を含む詳細なログ記録を使用できます。
- AWS は、例外に対する弾力性に対応したストレージシステムを設計しました。たとえば、Amazon Simple Storage Service (S3) はほぼ完全な耐久性を求めた設計になっています。(たとえば、Amazon S3 を使用して 10,000 のオブジェクトを保存する場合、平均で、10,000,000 年間に 1 つのオブジェクトを消失する可能性があります)。
- バージョニングは、より大容量データのライフサイクル管理処理の一部であり、不慮の上書き、削除などの障害に対してデータを保護できます。
- AWS ではリージョン間のデータの移動は実行しません。1 つのリージョンに置かれたコンテンツは、お客様が明示的に機能を有効にするか、その機能を提供するサービスを利用しないかぎり、そのリージョンに残り続けます。

次の質問はデータセキュリティの考慮事項に関するものです（セキュリティに関するすべての質問、回答、およびベストプラクティスの一覧については、付録を参照してください）。

SEC 1. **保管時のデータをどのように暗号化、保護していますか。**

SEC 2. **伝送中のデータをどのように暗号化、保護していますか。**

AWS では、保管時および伝送時のデータの暗号化用に複数の手段を用意しています。これらの機能を製品とサービスに組み込み、データの暗号化を容易にします。たとえば、サーバー側の暗号化（SSE）をAmazon S3 に実装し、暗号化されたフォームでのデータ保存をより簡単にします。また、HTTPS 全体の暗号化および復号化処理（一般には SSL 停止と呼ばれます）を Elastic Load Balancing で担当するように調整することもできます。

権限管理

権限管理は情報セキュリティプログラムの主要部分です。権限を持つユーザーおよび認証されたユーザーのみが、意図した方法でのみ、確実にリソースにアクセスできるようにします。たとえば、アクセスコントロールリスト (ACL) はオブジェクトに添付するアクセス許可のリストです。ロールベースアクセスコントロール (RBAC) はエンドユーザーのロールまたは機能に合わせたアクセス権限セットであり、パスワード管理には複雑性の条件および変更間隔が含まれています。これらの権限管理要素はユーザーの認証および許可の主要概念を表すものであるため、情報セキュリティアーキテクチャにおいて非常に重要です。

AWS では、権限管理は AWS Identity and Access Management (IAM) サービスで優先的にサポートされています。これを使用して AWS のサービスやリソースへのユーザーのアクセスを安全に管理できます。アクセス権限をユーザー、グループ、ロール、リソースに割り当てるポリシーは細かく適用できます。複雑性、再使用、Multi-Factor Authentication (MFA) など、強力なパスワードプラクティスを要求することもできます。また、既存のディレクトリサービスでフェデレーションを使用できます。

次の質問は、セキュリティのための権限管理の考慮事項に関するものです。

SEC 3.	**AWS ルートアカウントの認証情報へのアクセスと使用をどのように保護していますか。**
SEC 4.	**AWS マネジメントコンソールと API への人間によるアクセスを制御するために、システムユーザーのロールと責任をどのように定義していますか。**
SEC 5.	**AWS リソースへの自動化されたアクセス（アプリケーション、スクリプト、サードパーティーツールまたはサービスからのアクセスなど）をどのように制限していますか。**
SEC 6.	**キーと認証情報をどのように管理していますか。**

ルートアカウント認証情報を保護し続けることは非常に重要です。このため、AWS では MFA をルートアカウントに添付し、MFA 付きの認証情報を物理的に安全な場所にロックすることをお勧めします。IAM サービスでは、その他の（ルートではない）ユーザーアクセス権限を作成して管理し、またリソースへのアクセスレベルを確立できます。

インフラストラクチャの保護

インフラストラクチャの保護には深層防御や Multi-Factor Authentication などのコントロール方法が含まれ、ベストプラクティスおよび業界や規制上の義務に合致している必要があります。クラウドでもオンプレミスでも、オペレーションの実行を成功させるには、これらの手段を使用することが非常に重要です。

AWS では、AWS のネイティブ技術を使用するか、AWS Marketplace で入手できるパートナー製品およびサービスを使用して、ステートフルおよびステートレスパケットインスペクションを実装できます。また、Amazon Virtual Private Cloud (VPC) を使用してプライベートで安全なスケーラブル環境を作成し、ゲートウェイ、ルーティングテーブル、公開サブネットおよびプライベートサブネットを含むトポロジを定義できます。

次の質問は、セキュリティのためのインフラストラクチャ保護の考慮事項に関するものです。

SEC 7.	**ネットワークおよびホストレベルの境界保護をどのように実施していますか。**
SEC 8.	**AWS サービスレベルの保護をどのように実施していますか。**
SEC 9.	**Amazon EC2 インスタンス上のオペレーティングシステムの整合性はどのように保護していますか。**

多層防御はどのタイプの環境にもお勧めです。インフラストラクチャ保護の場合は、クラウドおよびオンプレミスモデルで多くの概念および方法が有効です。境界保護の徹底、送受信のモニタリング、広範囲のログ記録、モニタリングおよびアラートはすべて、効果的な情報セキュリティ計画に不可欠です。

前述の「設計の原則」セクションで説明したとおり、AWS のお客様は EC2 インスタンスの設定をカスタマイズまたは強化して、この設定を Amazon Machine Image (AMI) に配置できます。その後、Auto Scaling によるトリガーにしろ手動起動にしろ、この AMI で起動するすべての新規仮想サーバー（インスタンス）には、強化された設定が適用されます。

発見的コントロール

発見的コントロールはセキュリティ違反の検出または識別に使用できます。これらは管理フレームワークの通常の一部分であり、品質プロセス、法律上の準拠義務、脅威の識別や応答処理のサポートに使用できます。発見的コントロールには、さまざまな種類があります。たとえば、資産およびその詳細属性を目録化して、より効果的に意思決定（およびライフサイクル統制）を促進し、オペレーションの基礎を確立します。あるいは、情報システムに関連するコントロールを精査する内部監査を使用し、実態がポリシーおよび要件に一致しているか、定義した条件に基づいたアラート通知の自動化が正しく設定されているかを確認できます。これらのコントロールは重要なリアクティブファクターであり、企業が変則的なアクティビティの範囲を識別して理解する役に立ちます。

AWS では、次のサービスが発見的コントロールをサポートします。

- **AWS CloudTrail** – 呼び出しのアイデンティティ、呼び出し時刻、ソース IP アドレス、パラメーター、および応答要素を含む API コールをログ記録するウェブサービスです。
- **Amazon CloudWatch** – Amazon Elastic Compute Cloud (EC2) の CPU、ディスク、ネットワークのアクティビティや、Amazon Relational Database Service (RDS) データベースインスタンス、Amazon Elastic Block Store (EBS) ボリュームなどのログを記録する AWS リソース用のモニタリングサービスです。CloudWatch を使用してこれらのログや他のメトリックスを利用してアラームを通知することができます。
- **AWS Config** – 経時的なインフラストラクチャの設定と変更についての情報を提供する、インベントリーおよび設定の履歴を管理するサービスです。
- **Amazon Simple Storage Service (S3)**– Amazon S3 データアクセス監査を使用して、アクセス要求のタイプ、リソース、日付および時刻などを含む詳細を記録するように Amazon S3 バケットを設定できます。
- **Amazon Glacier**– ボールトロック機能を利用することで、監査可能な長期保存をサポートするように設計されたコンプライアンス上のミッションクリティカルなデータを保存することができます。

次の質問は、セキュリティのための発見的コントロールの考慮事項に関するものです。

SEC 10. AWS ログをどのように取得して分析していますか。

セキュリティやフォレンジックに関する規制・法令の要件のため、優れた設計においてはログ管理が重要です。AWS には、お客様がデータ保持ライフサイクルを定義したり、データを保管、アーカイブ、場合によっては削除する場所を定義したりすることで、ログ管理をより簡単に実装できる機能があります。この機能により、予測可能で信頼できるデータ操作がよりシンプルになり、かつコスト効果も高くなります。

関連する AWS サービス

セキュリティに不可欠な AWS サービスは AWS Identity and Access Management (IAM) です。AWS のサービスやリソースへのユーザーのアクセスを安全に管理できます。以下のサービスと機能は、セキュリティの 4 つの領域をサポートします。

データ保護:Elastic Load Balancing、Amazon Elastic Block Store (EBS)、Amazon Simple Storage Service (S3)、および Amazon Relational Database Service (RDS) などのサービスには、伝送中や保管中のデータを保護する暗号化機能があります。AWS Key Management Service (KMS) では、データの暗号化に使用するコントロールキーの作成や制御を簡単に行うことができます。

権限管理:IAM を利用すると、AWS のサービスおよびリソースに対するアクセスを安全にコントロールすることができます。Multi-Factor Authentication (MFA) は、ユーザー名およびパスワードに加え、拡張された保護レイヤーを追加します。

インフラストラクチャの保護:Amazon Virtual Private Cloud (VPC) では、AWS クラウド上にプライベートで隔離されたネットワークをプロビジョニングすることができます。ここでは、仮想ネットワークで AWS リソースを起動することができます。

発見的コントロール:AWS CloudTrail は AWS API コールを記録します。AWS Config は AWS リソースおよび設定の詳細なインベントリーを提供し、Amazon CloudWatch は AWS リソースのサービスをモニタリングします。

リソース

セキュリティに関連するベストプラクティスの詳細については、以下のリソースを参照してください。

ドキュメント & ブログ

- AWS セキュリティセンター
- AWS コンプライアンス
- AWS セキュリティブログ

ホワイトペーパー

- AWS セキュリティの概要
- AWS セキュリティのベストプラクティス
- AWS リスクおよびコンプライアンス

動画

- AWS クラウドのセキュリティ
- 責任共有モデルの概要

信頼性の柱

信頼性の柱には、インフラストラクチャまたはサービスの障害からの復旧、必要に応じた動的なコンピューティングリソースの獲得、設定ミスや一時的なネットワークの問題などによる障害軽減などのシステムの能力が含まれています。

設計の原則

クラウドでは、信頼性を高める役に立つ数多くの原則があります。

- **復旧手順のテスト:**オンプレミス環境では、テストはたいてい特定のシナリオによってシステムが機能することを証明するために行われます。復旧戦略を検証するためにテストが使用されることはあまりありません。クラウドでは、システム障害の発生過程をテストし、復旧手順を検証できます。自動化により、異なる障害をシミュレートしたり以前に障害が発生したシナリオを再作成したりできます。これらによって障害経路が洗い出され、実際に障害が発生する前にテストし修正することで、テストしていないコンポーネント障害のリスクを軽減できます。
- **障害から自動的に復旧する:**システムの主要なパフォーマンスインジケーター (KPI) をモニタリングすることで、しきい値を超過した場合に自動処理をトリガーできます。それにより自動的に障害を通知および追跡し、障害を回避または修正する復旧プロセスを自動化できます。より高度な自動化を使用して、障害が発生する前にそれを予測し修正することも可能です。

- **水平方向にスケールして集約システム能力を強化する:**1 つの大きなリソースを複数の小さなリソースに置換して、1 つの障害がシステム全体に及ぼす影響を軽減します。複数のより小さなリソースに要求を分散させて、障害の共通点を共有しないようにします。
- **キャパシティーの推測をやめる:**オンプレミスシステムで発生する障害の一般的な原因はリソースの飽和です。システムに対する要求がそのシステムの能力を超えたときに発生します (よくサービス妨害攻撃の目標になります)。クラウドでは、需要とシステム使用率を監視し、需要を満たすために最適なレベルを維持するためのリソースの追加や削除を自動化することができます。

定義

クラウドでの信頼性は、次の 3 つの領域で構成されます。

1. 基礎
2. 変更管理
3. 障害管理

信頼性を確保するには、システムによく計画された基礎があり適切な箇所をモニタリングし、また必要や要件に応じた変更を行う仕組みが必要です。障害を検出し自動的に自己修復するようにシステムを設計する必要があります。

ベストプラクティス

基礎

どのようなシステムでも、設計前に、信頼性に影響を及ぼすような基本的な要件が適切に設定されている必要があります。たとえば、データセンターへの十分なネットワーク帯域幅確保などです。これらの要件は時として放置されがちです (それぞれのプロジェクトスコープ外のため)。これらを放置すると、信頼性の高いシステムを構築する際に重大な影響を与える可能性があります。オンプレミス環境では、依存関係の影響によりこれらの要件を満たすには長い時間を必要とします。そのため計画の初期に組み込まれる必要があります。

AWS では、ほとんどの基礎要件は組み込み済みであるか、必要に応じて利用できます。クラウドは基本的に制限がないように設計されています。そのため、十分なネットワーキングおよびコンピューティングキャパシティの要件を満たすのは AWS の責任であり、お客様はオンデマンドでストレージデバイスのサイズなどのリソースサイズおよび割り当てを自由に変更できます。

次の質問は信頼性のための基礎的な考慮事項に関するものです（信頼性に関するすべての質問、回答、およびベストプラクティスの一覧については、付録を参照してください）。

REL 1. アカウントの AWS サービス制限はどのようにモニタリングしていますか。

REL 2. AWS でのネットワークトポロジをどのように計画していますか。

REL 3. 技術的な問題に対応するためのエスカレーションパスがありますか。

AWS は、お客様がリソースを意図せずよけいにプロビジョニングしてしまう事故を防止するため、サービス制限（チームが要求できる各リソースの数の上限）を設定しています。これらの制限をモニタリングしビジネスのニーズに合わせて変更するために、適切なガバナンスおよびプロセスを設ける必要があります。クラウドを取り入れるにあたって、既存のオンプレミスリソースとの統合（ハイブリッドアプローチ）を計画しなければならない場合があります。ハイブリッドモデルでは、段階的にオール・インのクラウド使用に移行できます。そのため、AWSとオンプレミスのリソースが相互作用する方法をネットワークトポロジとして設計することが重要です。最後に、ITチームには、パブリッククラウドの利用サポートのために、トレーニングと最新のプロセスを提供し、適切な場合にはパートナーまたはサポート契約を用意するのが良いでしょう。

変更管理

変更がシステムに及ぼす影響を把握すれば事前に計画を立てることができます。モニタリングによって容量の問題や SLA 未達を引き起こす傾向をすばやく見つけることができます。旧来の環境では、変更制御のプロセスは手動であることが多く、変更を行う担当者や変更実施時期を効果的に制御するには、監査を受けながら注意深く調整しなければなりません。

AWSを使用することで、システムの挙動をモニタリングし KPI の変化への対応を自動化できます。たとえば、システムのユーザーが増えた場合に追加サーバーを追加するなどです。誰にシステムを変更する権限を持たせるかを制御し、これらの変更の履歴を監査できます。

次の質問は、変更管理に関連する信頼性のための考慮事項に関するものです。

REL 4. システムは需要の変化にどのように対応できますか。

REL 5. AWS リソースをどのようにモニタリングしていますか。

REL 6. 変更管理をどのように実行していますか。

需要の変更に対応してリソースを自動的に追加および削除するようにシステムを設計すると、信頼性が向上するだけでなく、ビジネスの成功が重荷ではなくなります。適切にモニタリングしていれば、KPI が予想した水準を逸脱した場合、チームは自動的にアラートを受信します。環境に対する変更を自動的にログ記録することで、それを監査し信頼性に影響を与えるアクションをすばやく識別できます。変更管理を統制することで、必要とされる信頼性を達成するためのルールを強化できます。

障害管理

合理的な複雑性を持つシステムでは、障害は発生するものです。このような障害を感知し、対応し、再発を防止する方法には、誰しも関心を持っています。

AWS では、モニタリングデータに対応するために自動化を活用できます。たとえば、特定のメトリックスがしきい値を超えたとき、自動化されたアクションをトリガーして問題を修正できます。また、本番環境の一部である、障害が発生したリソースを診断して修正するのではなく、新しいものに置換して、障害が発生したリソースを本番環境外で分析できます。クラウドでは低価格で一時的なバージョンとしてシステム全体を立ち上げることができるため、自動化されたテストを使用して復旧プロセス全体を検証できます。

次の質問は、障害管理に関する信頼性のための考慮事項に関するものです。

REL 7. データをどのようにバックアップしていますか。

REL 8. システムはコンポーネントの障害にどのように対応しますか。

REL 9. 復旧についてどのように計画していますか。

論理エラーおよび物理エラーの両方から確実に復旧するように、定期的にデータをバックアップし、バックアップファイルをテストしてください。障害管理の要は、障害から復旧までの自動化されたシステムのテストを頻繁に実行することです（定期スケジュールに加えて、大規模なシステム変更後にも実行されるのが理想です）。システムの適合性（特に障害テストシナリオ内で）を評価して、単一障害点を識別し修正する手がかりにするために、目標復旧時間 (RTO) および目標復旧時点 (RPO) などの KPI を能動的にトラッキングしてください。目標は、システム復旧プロセスを徹底的にテストし、問題が解決できない局面においても、すべてのデータが復旧可能で顧客へのサービスを継続できる確信を持ち続けることです。復旧プロセスは通常の本番稼働プロセスと同様に習熟されていなければなりません。

関連する AWS サービス

信頼性を確実にする要となる AWS サービスは、実行中のメトリックスをモニタリングする Amazon CloudWatch です。信頼性の 3 つの領域をサポートするその他のサービスと機能は、次のとおりです。

基盤サービス:AWS Identity and Access Management (IAM) により、AWS サービスおよびリソースへのアクセスを安全にコントロールすることができます。Amazon VPC を使用すると、AWS クラウドの隔離されたプライベートなセクションをプロビジョニングし、仮想ネットワークで AWS リソースを起動することが可能になります。

変更管理:AWS CloudTrail は、アカウントの AWS API コールを記録し、アカウント保持者に監査用のログファイルを送信します。AWS Config は AWS リソースのおよび設定の詳細インベントリーを提供し、継続して設定の変更を記録します。

障害管理:AWS CloudFormation を使用すれば AWSリソースをテンプレートを使用して作成し、整然とした予測可能な方法でプロビジョニングできます。

リソース

信頼性に関連するベストプラクティスの詳細については、以下のリソースを参照してください。

ビデオおよびアナリストレポート

- 障害の受け入れ:障害の意図的な生成とサービスの信頼性
- クラウドでの可用性と信頼性のベンチマーク

ドキュメントおよびブログ

- サービス上限についてのドキュメント
- サービス上限についてのブログ 投稿記事

ホワイトペーパー

- AWS を用いたバックアップおよびリカバリのアプローチ ホワイトペーパー
- 大規模環境での AWS インフラストラクチャの管理 ホワイトペーパー
- 災害対策目的での AWS の使用 ホワイトペーパー
- AWS Amazon VPC のネットワーク接続オプションホワイトペーパー

AWS サポート

- AWS プレミアムサポート
- Trusted Advisor

パフォーマンス効率の柱

パフォーマンス効率化の柱は、コンピューティングリソースを要求に合わせて効率的に使用し、需要の変化や技術の進化に合わせてその効率を維持することです。

設計の原則

クラウドでは、パフォーマンス効率を達成するいくつかの原則があります。

- **先端技術の開放:**その知識や複雑性をクラウドベンダーの領域に押し込むことで、実装が難しかった技術の利用がより簡単になります。IT チームに新技術のホストおよび実行方法を学習してもらうよりも、単純にサービスとして使用してもらうことができます。たとえば、NoSQL データベース、メディア変換、機械学習はどれも専門知識を必要とする技術ですが、この専門知識は技術コミュニティに広く行き渡っているわけではありません。クラウドでは、これらの技術はチームが使用できるサービスとなり、リソースのプロビジョニングや管理よりも製品開発に注力できます。
- **グローバル化を即座に達成:**わずか数クリックで、世界中の複数のリージョンにシステムを簡単にデプロイしてください。こうすることで、簡単に最小限のコストで、より低いレイテンシーとより良いエクスペリエンスを顧客に提供できます。

- **サーバーレスアーキテクチャの使用:**クラウドでは、サーバーレスアーキテクチャにより、従来のコンピューティング活動を実行するサーバーを稼働させ維持する必要がなくなります。たとえば、ストレージサービスは静的ウェブサイトとして動作でき、ウェブサーバーの必要がなくなります。イベントサービスならお客様のプログラムコードをホストできます。サーバーを管理するオペレーション上の負担がなくあるだけではなく、トランザクションコストが低くなります。なぜなら、これらのマネージドサービスは、クラウドのスケールで実行されるからです。
- **実験の頻度の増加:**仮想化された自動化可能なリソースを使用して、異なったタイプのインスタンス、ストレージ、設定を使用した比較テストを簡単に実行できます。

定義

クラウドでの**パフォーマンス効率**は、次の 4 つの領域で構成されます。

1. コンピューティング
2. ストレージ
3. データベース
4. 容量と時間のトレードオフ

これらの領域それぞれでの考慮事項には、a) 適切なアプローチおよびリソースの選択方法 b) 現在のアプローチを維持しながらクラウドの能力を進化させる方法 c) 実行中のパフォーマンスを予測と比較してモニタリングする方法 d) 需要に合わせてリソースをスケールする方法が含まれます。

ベストプラクティス

コンピューティング

特定のアーキテクチャに向けた適切なサーバー設定はアプリケーション設計、使用パターン、構成設定によって異なります。多くのシステムでは、パフォーマンスを向上させるために、さまざまなコンポーネントに対して異なるサーバー設定を使用し、異なる機能を活動化しています。ユースケースに対して適切でないサーバー設定を選択すると、パフォーマンス効率が低下する可能性があります。

AWS では、サーバーは仮想化されているため、ボタンクリックまたは API コールで機能を変更できます。リソースの決定が変更できないものではなくなったため、異なるサーバータイプで実験できます。AWS では、これらの仮想サーバーインスタンスは異なるファミリーおよびサイズで使用でき、SSD や GPU など幅広い機能を提供します。AWS では、サーバーレスのコンピューティングを実行することもできます。たとえば、AWS Lambda ではインスタンスを実行せずにコードを実行できます。

次の質問例はコンピューティングの考慮事項に関するものです（パフォーマンス効率に関するすべての質問、回答、およびベストプラクティスの一覧については、付録を参照してください）。

PERF 1. **システムに対して適切なインスタンスタイプはどのように選択していますか。**

PERF 2. **新しいインスタンスタイプや機能の提供が開始される中で、どのようにして最適なインスタンスタイプを使い続けていることを担保していますか。**

PERF 3. **起動後のインスタンスが期待どおりの性能を出すように、インスタンスをどのようにモニタリングしていますか。**

PERF 4. **どのようにしてインスタンスの数量を需要に一致させていますか。**

使用するインスタンスタイプを選択するときは、そのワークロードに最適なインスタンスタイプ（またはサーバーレスの手法）を示すテストデータがあることが重要です。新しいインスタンスタイプや機能が利用可能になったときに簡単にテストできるように、これらのテストは繰り返し可能（理想的には継続的デリバリ (CD) パイプラインの一部として）であるべきです。運用の観点からは、いかなるパフォーマンスの低下も通知するモニタリングが実行されている必要があります。

ストレージ

特定のシステムの最適なストレージソリューションは、アクセス方法の種類（ブロック、ファイル、またはオブジェクト）、アクセスのパターン（ランダムまたはシーケンシャル）、必要なスループット、アクセス頻度（オンライン、オフライン、アーカイブ）、更新頻度（ワーム、動的）、および可用性と耐久性の制約によって異なります。優れた設計システムでは、複数のストレージソリューションを使用し、さまざまな機能を有効にしてパフォーマンスを向上させます。

AWS では、ストレージは仮想化され、さまざまなタイプで利用できます。これにより、データ保管方法をニーズに一層忠実に一致させることがより簡単になるとともに、オンプレミスのインフラストラクチャでは簡単に実現できないストレージの選択肢も得られます。たとえば、Amazon S3 は、イレブンナイン (99.999999999%) の堅牢性を実現するよう設計されています。また、マグネティックハードドライブ (HDD) の使用をソリッドステートドライブ (SSD) に変更することも、仮想ドライブを1つのインスタンスから別のインスタンスに数秒で簡単に移動することもできます。

次の質問例は、パフォーマンス効率を向上させるためのストレージの考慮事項に関するものです。

PERF 5. **システムに最適なストレージソリューションをどのように選択していますか。**

PERF 6. **新しいストレージソリューションや機能が提供開始される中で、最適なストレージソリューションを使い続けていることを、どのように担保していますか。**

PERF 7. **ストレージソリューションが想定通りの性能を示していることを確実にするために、どのようにモニタリングしていますか。**

PERF 8. **どのようにしてストレージソリューションのキャパシティーとスループットが需要に一致するようにしていますか。**

ストレージソリューションを選択するときには、そのワークロードに必要なコストと価値のマージンをどのストレージソリューションが提供するか示すテストデータを持つことが重要です。新しいストレージソリューションや機能が利用可能になったときに簡単にテストできるように、これらのテストは繰り返し可能（理想的には継続的デリバリ (CD) パイプラインの一部として）であるべきです。異なるインスタンスに使用されるストレージのタイプ（EBS 対インスタンスストア、あるいは HDD 対 SSD）は、システムのパフォーマンス効率に大きな影響を与えます。運用の観点からは、いかなるパフォーマンスの低下も通知するモニタリングが実行されている必要があります。

データベース

特定のシステムに最適なデータベースソリューションは、整合性、可用性、分断耐性、およびレイテンシーについての要件によって異なります。多くのシステムでは、パフォーマンスを向上させるために、さまざまなサブシステムに対して異なるデータベースソリューションを使用し、異なる機能を活動化しています。システムに対して適切でないデータベースソリューションや機能を選択すると、パフォーマンス効率が低下する可能性があります。

AWS では、Amazon Relational Database Service (RDS) が、完全マネージド型リレーショナルデータベースサービスを提供します。Amazon RDS を使用すれば、データベースのコンピューティングリソースやストレージリソースをスケールすることができ、多くの場合、ダウンタイムは発生しません。また、AWS は、その他のデータベースソリューションやストレージソリューションも提供しています。Amazon DynamoDB は、どのような規模でも、数ミリ秒台に抑えられたレイテンシーを提供する、フルマネージド型 NoSQL データベースです。Amazon Redshift は、パフォーマンスまたはキャパシティーのニーズが変わったときにノードの数やタイプを変更することができる、マネージド型でペタバイトスケールのデータウェアハウスです。

次の質問例は、パフォーマンス効率を向上させるためのデータベースの考慮事項に関するものです。

PERF 9. **システムに最適なデータベースソリューションをどのように選択していますか。**

PERF 10. **新しいデータベースソリューションや機能が提供開始される中で、最適なストレージソリューションを使い続けていることを、どのように担保していますか。**

PERF 11. **データベースが想定通りの性能を示していることを確実にするために、どのようにモニタリングしていますか。**

PERF 12. **どのようにしてデータベースのキャパシティーとスループットが需要に一致するようにしていますか。**

組織のデータベース手法（RDBMS、NoSQL など）はシステムのパフォーマンス効率に大きな影響を与えますが、これは評価を通じてではなく、組織のデフォルトに従って選択されることがしばしばある領域です。データベースソリューションのビルドおよびデプロイ中は、1回限りの変えられない決定事項ではなく、時間の経過とともに進化できるように、データベースをコードとして扱ってください。各ワークロードに最適なデータベースソリューションを特定するには、テストデータを使用してください。新しいデータベースソリューションや機能が利用可能になったときに簡単にテストできるように、これらのテストは繰り返し可能（理想的には継続的デリバリ (CD) パイプラインの一部として）であるべきです。たとえば、読み取り専用レプリカは、その他の非機能要件に反することなく、パフォーマンスの効率を改善するかどうか評価してください。運用の観点からは、いかなるパフォーマンスの低下も通知するモニタリングが実行されている必要があります。

容量と時間のトレードオフ

ソリューションを構築する際には、処理時間（コンピューティング）を減らすために容量（メモリまたはストレージ）を使用するか、容量を減らすために時間を使うかに関する一連のトレードオフがあります。また、リソースやキャッシュされたデータをエンドユーザーのより近くに配置して時間を減らす方法もあります。

AWS を使用すれば、即座にグローバル化を達成し、世界各地の複数の場所でリソースをデプロイして、エンドユーザーに近づけることができます。また、読み取り専用レプリカをデータベースシステムなどの情報ストアに動的に追加し、プライマリデータベース上の負荷を減らすこともできます。

低レイテンシーおよび高スループットを実現するためにAWSのグローバルなインフラストラクチャを使用し、同時にデータが指定したリージョン内にのみ保存されることを確実にしてください。AWS Direct Connect などのネットワーキングソリューションは、オンプレミスネットワークと AWS インフラストラクチャ間の予測可能なレイテンシーを提供するよう設計されています。AWS は、効率を高める Amazon ElastiCache や、静的コンテンツのコピーをエンドユーザーの近くにキャッシュする Amazon CloudFront などのキャッシュソリューションも提供しています。

次の質問例は、パフォーマンス効率を向上させるための容量と時間のトレードオフに関するものです。

PERF 13. システムに最適な近接性およびキャッシュソリューションをどのように選択していますか。

PERF 14. 新しいソリューションが提供開始される中で、最適な近接性およびキャッシュソリューションを使い続けていることを、どのように担保していますか。

PERF 15. パフォーマンスを予期どおりのものとするには、近接性およびキャッシュソリューションをどのようにモニタリングしていますか。

PERF 16. どのようにして近接性およびキャッシュソリューションのキャパシティーとスループットが需要に一致するようにしていますか。

パフォーマンス効率を実現するには容量と時間のトレードオフが必要であり、そのワークロードに最も一致するトレードオフを示すテストデータがあることが重要です。新しい手法や機能が利用可能になったときに簡単にテストできるように、これらのテストは繰り返し可能（理想的には継続的デリバリ (CD) パイプラインの一部として）であるべきです。たとえば、ライトアサイドキャッシュとして Amazon ElastiCache を使用することは、その他の非機能要件に違反することなく、パフォーマンス効率を向上できるかどうかテストしてください。運用の観点からは、いかなるパフォーマンスの低下も通知するモニタリングが実行されている必要があります。アーキテクチャは需要に応じてスケールし、その需要とのマージンを保つ必要があります。

関連する AWS サービス

パフォーマンス効率を高めるための主要な AWS サービスは、Amazon CloudWatch です。これはリソースやシステムをモニタリングし、パフォーマンス全体および運用健全性を可視化します。以下のサービスは、パフォーマンス効率の 4 つの領域において重要です。

コンピューティング: Auto Scaling は、需要を満たし応答性を維持するために十分なインスタンスを確保するための鍵となります。

ストレージ: Amazon EBS は、ユースケースにあわせて最適化するための広範囲なストレージオプション（SSD や PIOPS など）を提供します。Amazon S3 は、低冗長化ストレージ、Amazon Glacier（アーカイブストレージ）へのライフサイクルポリシー、サーバーレスのコンテンツ配信機能も提供します。

データベース: Amazon RDS は、ユースケースにあわせて最適化できる広範囲なデータベース機能（プロビジョンド IOPS やリードレプリカなど）を提供します。Amazon DynamoDB は、どのような規模でも、数ミリ秒台に抑えられたレイテンシーを提供します。

容量と時間のトレードオフ: AWS は世界各地にリージョンを持っており、リソース、データ、および処理に最適な場所を選ぶことができます。Amazon CloudFront を使用すると、さらにユーザーに近い場所でコンテンツをキャッシュすることができます。

リソース

パフォーマンス効率に関連するベストプラクティスの詳細については、以下のリソースを参照してください。

動画

- Performance Channel
- AWSにおけるパフォーマンスベンチマーキング

ドキュメント

- Amazon S3 パフォーマンスの最適化ドキュメント
- Amazon EBS ボリュームのパフォーマンスのドキュメント

コストの最適化の柱

不要なコストや最適でないリソースを回避または排除することが出来るか評価するために**コスト最適化**の柱を使ってください。それによる節約をビジネス上の差別化された利点のために使用してください。コストが最適化されたシステムでは、ビジネスの目標を達成し、その他の優れたアーキテクチャの柱の主要な要件を満たすかそれを超えつつ、可能な限り低い料金を支払うだけで良いことになります。適切なアーキテクチャの選択、使用していないリソースの削減、および最も経済的な手法の選択を行うテクニックを使用してコストの最適化を達成できるのです。

設計の原則

クラウドでは、コストの最適化を達成できる数多くの原則に従うことができます。

- **透明性のある費用賦課:**クラウドにより、システムのコストを識別し、個別のビジネスオーナーに IT コストを割り当てることが容易になります。これは投資効果の確認する助けとなり、その結果としてリソースを最適化して、コストを削減化するためのインセンティブをビジネスオーナーに与えることになります。
- **マネージドサービスを使用して所有コストを減らす:**クラウドでは、マネージドサービスは E メールの送信といったタスクを行うサーバーの保守や、データベースの管理のような運用面の負荷を取り除きます。さらに、マネージドサービスはクラウドスケールで実行されるため、トランザクションまたはサービスあたりのコストが低くなります。

- **固定の償却コストを変動コストに転換:**使用量もわからないうちにデータセンターやサーバーに多額の投資を行うのではなく、リソースを使用した時に、使用した分だけお支払いください。たとえば、開発環境やテスト環境では、通常は、週のうち稼働日だけ、1 日あたり 8 時間のみ使用されますので、使用しないときはこれらのリソースを停止して最大 75% のコストを節約できます(40 時間対 168 時間)。
- **スケールによる大きなコストメリット:**クラウドコンピューティングを使用すると、自社環境よりも低い変動コストを実現できます。AWS には高いスケールメリットがあるためです。数十万単位のユーザーの使用がクラウドに集約されるため、従量課金制の料金も低くなります。
- **データセンターの運用への投資が不要に:**サーバーのラッキング、スタック、電源供給といった手間のかかる作業は AWS が行うため、お客様は IT インフラストラクチャではなく自社の顧客やビジネスプロジェクトに集中することができます。

定義

クラウドでの**コストの最適化**は、次の 4 つの領域で構成されます。

1. 供給と需要の一致
2. コスト効果が高いリソース
3. 費用の把握
4. 継続した最適化

他の柱と同じように、市場までの時間を最適化するのか、コストを最適化するのかなど、考慮すべきトレードオフがあります。場合によっては、先行コストの最適化に投資するのではなく、市場までの時間、新機能の提供、または単純に期日の順守といった、スピードの最適化が最善であることもあります。設計上の決定は、経験的データを使用せずにあわてて行われる場合があります。コスト最適化したデプロイのベンチマークに時間を費やすよりも、「念のため」に過度な補償を行いたくなるためです。これにより、過度にプロビジョニングしすぎ、最適化されていないデプロイがしばしば生まれます。以下のセクションでは、デプロイの初期コストの最適化および継続的なコストの最適化についての技術的および戦略的なガイダンスを示します。

ベストプラクティス

供給と需要の一致

需要と供給を最適に一致させることで、システムのコストは最少となりますが、プロビジョニングの時間と個別のリソースの障害を考慮して、十分な供給の余力を確保しておく必要もあります。需要は固定または可変の場合があるため、管理が大きなコストとならないようにメトリックスと自動化が必要になります。

AWS では、需要に合わせて自動的にリソースをプロビジョニングできます。Auto Scaling、時間、イベント駆動、キューに基づく手法により、必要に応じてリソースを追加または削除できます。需要の変動を予想できる場合、もっと費用を節約し、リソースをシステムのニーズに一致させることができます。

次の質問例は、コストの最適化のための需要と供給の一致に関するものです（コストの最適化に関するすべての質問、回答、およびベストプラクティスの一覧については、付録を参照してください）。

COST 1. **キャパシティーが必要量を満たしているが大幅に超えていないことをどのように実現していますか。**

COST 2. **AWS サービスの使用量をどのようにして最適化していますか。**

モニタリングツールと定期的なベンチマークにより、リソースの使用率を大幅に高めることができます。オンデマンドコンピューティングや Auto Scaling、その他の自動化デプロイ機能など、柔軟性の高い機能を活用してより高度な最適化を進めることで、必要なリソースのみをプロビジョニングして、スケールアウトできるようになります。

コスト効果が高いリソース

システムに対して適切なインスタンスとリソースを使用することが、コスト削減の鍵となります。たとえば、小規模なサーバーでレポートを作成する処理に 5 時間かかるところ、コストが 2 倍の大規模なサーバーでは、これを 1 時間で実行できるとします。どちらのジョブでも結果は同じですが、小規模なサーバーでは時間とともにより多くのコストが発生します。

優れた設計のシステムでは、最もコスト効率の高いリソースが使用され、それにより大きなコスト効果が得られます。また、コストを削減するために、マネージドサービスを使用することもできます。たとえば、E メール配信のために複数のメールサーバーを維持する代わりに、メッセージごとに課金するサービスを使用することができます。

AWS は柔軟でコスト効率が高いさまざまな料金オプションを提供し、ニーズに最も合った方法で Amazon EC2 インスタンスを利用できるようにします。オンデマンドインスタンスは、最低契約金なしに、時間単位で、コンピューティング性能に対して料金をお支払いいただくものです。リザーブドインスタンス (RI) では、キャパシティーを予約し、オンデマンド料金の最大 75 パーセントを節約できます。スポットインスタンスでは、使用されていない Amazon EC2 キャパシティーを、大幅な割引で価格を指定して利用することができます。スポットインスタンスは、HPC やビッグデータのように、サーバー群の中の個々のサーバーが、動的に追加/削除されるような状況での利用に適しています。

次の質問例は、コスト最適化のために、コスト効率の高いリソースを選択する方法に関するものです。

COST 3. コスト目標を達成するために適切なリソースタイプを選択しましたか。

COST 4. コスト目標を達成するために適切な料金モデルを選択していますか。

COST 5. ROI を高めるために使用できるマネージドサービス（Amazon EC2、Amazon EBS、および Amazon S3 よりもハイレベルなサービス）はありますか。

AWS Trusted Advisor などのツールを使用して定期的に AWS の使用量を確認することで、使用率をアクティブにモニタリングし、それに応じてデプロイを調整できます。また、項目あたりのコストや管理コストを低減できる、Amazon RDS、Amazon Elastic MapReduce (EMR)、Amazon DynamoDB などのマネージド AWS サービスを活用することもできます。Amazon CloudFront などの CDN ソリューションで、ネットワークトラフィックに関連するコストの低減を検討してください。

費用の把握

クラウドで柔軟性と俊敏性が高まることにより、イノベーションと速いペースの開発およびデプロイが容易になります。ハードウェア仕様の確認、価格見積りの交渉、発注書の管理、配送のスケジュール、リソースのデプロイなど、オンプレミスインフラストラクチャのプロビジョニングに関連する手動のプロセスと時間が排除されます。ただし、使いやすさと事実上無制限のオンデマンドキャパシティーでは、支出に関する新しい考え方が必要になる場合があります。

多くのビジネスは、さまざまなチームによって実行される複数のシステムで構成されています。それぞれのビジネスオーナーまたは製品オーナーにリソースのコストを割り当てるようにすると、リソースを効率的に利用するようになり、無駄を低減できます。正確なコストを特定することで、実際に利益率の高い製品を把握することができ、予算の配分先についてより多くの情報に基づいた決定ができるようになります。

次の質問例は、コストの最適化のための費用の把握に関するものです。

COST 6. AWS のコストを管理するためにどのようなアクセス制御と手順を実行していますか。

COST 7. 使用状況と支出をどのようにモニタリングしていますか。

COST 8. 必要なくなったリソースをどのように廃棄していますか。または、一時的に必要ないリソースをどのように停止していますか。

COST 9. アーキテクチャを設計するときに、データ転送料金についてどのように考えていますか。

コスト配分タグを使用して、AWS コストをカテゴライズおよび追跡できます。使用している AWS リソース（たとえば Amazon EC2 インスタンスや Amazon S3 バケット）にタグを適用すると、使用量とコストをタグごとに集計したコスト配分レポートが生成されます。自社のカテゴリ（たとえばコストセンター、システム名、所有者）を表すタグを適用すると、複数のサービスにわたってコストを分類することができます。

タグ付けされたリソースに対してコストを可視化することで、ビジネス上の価値を生み出しておらず、削除する必要のある孤立したリソースやプロジェクトの確認が容易になります。予測される費用超過を通知する請求アラートを設定できます。また、AWS 簡易見積りツールによりデータ転送コストを計算することもできます。

継続した最適化

AWS が新しいサービスや機能を公開するときに、既存のアーキテクチャ上の決定を再評価し、それらが引き続き最もコスト効果が高いことを確認するのがベストプラクティスです。要件が変わったら、必要なくなったリソースとサービス全体またはシステムを積極的に廃棄します。

AWS のマネージドサービスにより、しばしばソリューションを大幅に最適化できるため、利用可能になった新しいマネージドサービスについて確認することをお勧めします。たとえば、Amazon RDS データベースの実行は、Amazon EC2 で独自のデータベースを実行するよりも安価になる場合があります。

次の質問例は、コスト最適化のためのコストの再評価に関するものです。

COST 10. 新しいサービスの採用をどのように管理または検討していますか。

デプロイを定期的に再評価することで、新しい AWS サービスを利用してコストを下げることがしばしば可能になります。また、新しいサービスの適用性を評価して費用を節約することができます。たとえば、Aurora 用の AWS RDS では、リレーショナルデータベースのコストを下げることができます。

関連する AWS サービス

コストの最適化をサポートする主な AWS 機能はコスト配分タグで、システムのコストを理解するうえで役立ちます。以下のサービスと機能は、コスト最適化の 4 つの領域に関して重要です。

供給と需要の一致: Auto Scaling では、費用の超過を抑えて、需要に合わせてリソースを追加または削除できます。

コスト効果が高いリソース: リザーブドインスタンスと前払いのキャパシティを使用してコストを削減できます。AWS Trusted Advisor を使用してお客様の AWS 環境を検査し、コスト削減の機会を見つけることができます。

費用の把握: Amazon CloudWatch アラームと Amazon Simple Notification Service (SNS) では、予算額を超えた場合や、予算額を超える見通しの場合に警告を通知することができます。

継続した最適化:AWS ブログと AWS ウェブサイトの「最新情報」セクションは、新機能や新しいサービスについて学習するためのリソースです。AWS Trusted Advisor は AWS 環境を検査し、使用されていないリソースやアイドル状態のリソースを排除したり、リザーブドインスタンス容量をコミットしたりすることで、コスト削減の可能性を見つけます。

リソース

コストの最適化に関連する AWS ベストプラクティスの詳細については、以下の参考資料を参照してください。

動画

- AWS でのコスト最適化

ドキュメント

- AWS エコノミクスセンター

ツール

- AWS 総所有コスト (TCO) 計算ツール
- AWS 詳細な請求レポート
- AWS 簡易見積りツール
- AWS コストエクスプローラー

まとめ

AWS による優れた設計のフレームワークは、クラウド上の信頼性、セキュリティ、効率、コスト効果が高いシステムを設計するための 4 つの柱についてアーキテクチャのベストプラクティスを提供します。このフレームワークには、既存のアーキテクチャまたは提案されたアーキテクチャを評価するための質問と、それぞれの柱についての AWS ベストプラクティスが用意されています。アーキテクチャでフレームワークを使用すると、安定した効率的なシステムを構築することができ、より機能的な要件に注目することができます。

寄稿者

本書の執筆に当たり、次の人物および組織が寄稿しました。

- Philip Fitzsimons、ソリューションアーキテクチャマネージャー、アマゾン ウェブ サービス
- Erin Rifkin、上級プログラムマネージャー、アマゾン ウェブ サービス
- Callum Hughes、ソリューションアーキテクト、アマゾン ウェブ サービス
- Max Ramsay、プリンシパルセキュリティソリューションアーキテクト、アマゾン ウェブ サービス
- Scott Paddock、セキュリティソリューションアーキテクト、アマゾン ウェブ サービス

ドキュメント履歴

2015 年 11 月 20 日: 最新の Amazon CloudWatch ログ情報で付録を更新しました。

付録: 優れた設計のための質問、回答、ベストプラクティス

この付録には、ベストプラクティスを含む優れた設計に関する質問と回答の完全な一覧が、柱ごとに整理されて含まれています。

セキュリティの柱

SEC 1. 保管時のデータをどのように暗号化、保護していますか。

伝統的なセキュリティ統制では、保管時のデータを暗号化します。AWS は、クライアント側（SDK サポート、OS サポート、Windows Bitlocker、dm-crypt、Trend Micro SafeNet など）およびサーバー側（Amazon S3 など）の両方を使用してこれをサポートしています。サーバー側の暗号化 (SSE) や Amazon Elastic Block Store 暗号化ボリュームなどを使用することもできます。

ベストプラクティス:

- 保管時のデータは、AWS サービス固有の統制（Amazon S3 SSE、Amazon EBS 暗号化ボリューム、Amazon Relational Database Service (RDS) Transparent Data Encryption (TDE) など）を使用して暗号化します。
- 保管時のデータは、クライアント側の手法を使用して暗号化します。
- AWS Marketplace または APN パートナーからのソリューション。

SEC 2. 伝送中のデータをどのように暗号化、保護していますか。

暗号化を使用することによって、伝送中のデータを保護するのがベストプラクティスです。AWS は、サービス API に対する暗号化エンドポイントの使用をサポートします。さらに、お客様は Amazon EC2 インスタンス内でさまざまな技術を使用できます。

ベストプラクティス:

- SSL が有効な AWS API が適切に使用されている。
- SSL または同等のものが通信に使用されている。
- VPN ベースのソリューション。
- プライベート接続（AWS Direct Connect など）。
- AWS Marketplace ソリューションが使用されている。

SEC 3. AWS ルートアカウントの認証情報へのアクセスと使用をどのように保護していますか。

AWS ルートアカウントの認証情報は、他のオペレーティングシステムのルートまたはローカル管理者と似ており、非常に注意して使用する必要があります。最新のベストプラクティスでは、AWS Identity and Access Management (IAM) ユーザーを作成し、それらのユーザーを管理者グループに関連付けて、IAM ユーザーを使ってアカウントを管理します。AWS ルートアカウントには API キーを持たせず、強力なパスワードを設定し、ハードウェアの Multi-Factor Authentication (MFA) デバイスの認証を有効にしてください。これにより、ルート ID は AWS マネジメントコンソールを通じてのみ使用でき、アプリケーションプログラミングインターフェイス (API) コールに使用することはできません。一部のリセラーやリージョンでは、AWS ルートアカウント認証情報が配布またはサポートされていないことに注意してください。

ベストプラクティス:

- AWS ルートアカウントの認証情報は、必要最小限の作業にのみ使用される。
- AWS ルートアカウントと関連付けられた MFA ハードウェアデバイスがある。
- AWS Marketplace ソリューションが使用されている。

SEC 4. AWS マネジメントコンソールと API へのユーザーによるアクセスを制御するために、システムユーザーのロールと責任をどのように定義していますか。

お客様にとって最新のベストプラクティスは、ユーザーグループを作成してシステムユーザーの定義されたロールと責任を分離することです。ユーザーグループは、いくつか異なる技術を使用して定義できます。Identity and Access Management (IAM) グループ、クロスアカウントアクセス用の IAM ロール、Security Assertion Markup Language (SAML) 統合によるウェブ認証（例: Active Directory でのロール定義）、または SAML や AWS Security Token Service (STS) により統合されるサードパーティーソリューションの使用 (Okta、Ping Identity、別のカスタムテクニック）などです。共有アカウントは使用しないことを強くお勧めします。

ベストプラクティス:

- IAM ユーザーとグループ
- SAML との統合
- ウェブ ID フェデレーション
- AWS Security Token Service (STS)
- クロスアカウントアクセス用の IAM ロール
- AWS Marketplace（Okta、Ping Identity など）または APN パートナーからのソリューション。
- 従業員のライフサイクルポリシーが定義、実施されている
- ユーザー、グループ、およびロールが明確に定義され、ビジネス要件を達成するために必要な最小の権限のみが付与されている

SEC 5. AWS リソースへの自動化されたアクセスをどのように制限していますか（アプリケーション、スクリプト、サードパーティーのツールやサービスなど）。

システムによるアクセスは、ユーザーグループの作成と同様の方法で定義する必要があります。Amazon EC2 インスタンスの場合、これらのグループは EC2 用の IAM ロールと呼ばれます。最新のベストプラクティスは、EC2 用の IAM ロールと AWS SDK または CLI を使用することです。これらには、EC2 認証情報のための IAM ロールを取得する機能が組み込まれています。従来、ユーザーの認証情報は EC2 インスタンス内に設定されていましたが、スクリプトやソースコードへの認証情報のハードコーディングはお勧めできません。

ベストプラクティス:

- Amazon EC2 用の IAM ロール
- IAM ユーザー認証情報が使用されているが、スクリプトやアプリケーションにハードコーディングされていない
- SAML との統合
- AWS Security Token Service (STS)
- EC2 インスタンス用に OS 固有のコントロールが使用されている
- AWS Marketplace ソリューションが使用されている

SEC 6. キーと認証情報をどのように管理していますか。

キーと認証情報は、秘密情報であるため保護される必要があり、適切なローテーションポリシーを定義して管理する必要があります。ベストプラクティスは、これらの秘密情報を管理スクリプトやアプリケーションにハードコーディングしないことです。しかし、これはしばしば行われています。

ベストプラクティス:

- 適切なキーと認証情報のローテーションポリシーが使用されている。
- AWS CloudHSM を使用する。
- AWS 管理型のキーに AWS サーバーサイドの手法が使用されている（Amazon S3 SSE、Amazon EBS 暗号化ボリュームなど）。
- AWS Marketplace ソリューション（SafeNet、TrendMicro など）。

SEC 7. ネットワークおよびホストレベルの境界保護をどのように実施していますか。

オンプレミスデータセンターで、DMZ 手法ではファイアウォールを使用して異なるシステムを信頼済みゾーンと信頼されていないゾーンに分離します。AWS で、ステートフルなファイアウォールとステートレスのファイアウォールの両方が使用されます。ステートフルなファイアウォールはセキュリティグループと呼ばれ、ステートレスなファイアウォールは、Amazon Virtual Private Cloud (VPC) のサブネットを保護するネットワークアクセスコントロールリスト (ACL) と呼ばれます。最新のベストプラクティスは、VPC でシステムを実行し、セキュリティグループでロールベースのセキュリティ（ウェブ層、アプリケーション層など）を定義して、ネットワーク ACL で場所ベースのセキュリティ（アベイラビリティーゾーンごとに 1 つのサブネットでの Elastic Load Balancing 層、アベイラビリティーゾーンごとに別のサブネットでのウェブ層など）を定義することです。

ベストプラクティス:

- 最小限の許可を持つセキュリティグループが、ロールベースのアクセスに使用されている。
- システムは 1 つ以上の VPC で実行される。
- 信頼された VPC アクセスは、保護されたアクセス方法（仮想プライベートネットワーク (VPN)、IPsec トンネル、AWS Direct Connect、AWS Marketplace ソリューション）を通じて行われる。
- サブネットとネットワーク ACL が適切に使用されている。
- ホストベースのファイアウォールで最小限の権限のみが許可されている。
- サービス固有のアクセス制御が使用されている（バケットポリシーなど）。
- VPC へのプライベート接続が使用されている（VPN、AWS Direct Connect、VPC ピア接続など）。
- インスタンスの管理に踏み台ホストが使用されている。
- セキュリティテストが定期的に実行されている。
- AWS Trusted Advisor チェックが定期的に確認されている。

SEC 8. AWS サービスレベルの保護をどのように実施していますか。

別のベストプラクティスは、リソースへのアクセスを制御することです。AWS Identity and Access Management (IAM) により、さまざまなリソースレベルの制御（例: 暗号化の使用、時刻、ソース IP など）を定義することができます。また、サービスごとに追加のアクセス制御を使用することもできます（例: Amazon S3 bucket ポリシーなど）。さらに、お客様は Amazon EC2 インスタンス内でさまざまな技術を使用できます。

ベストプラクティス:

- 最小の権限で認証情報を設定する。
- 職務分掌。
- アクセス権限の定期的な監査。
- 重要な API コールに対してリソース要件が定義されている（MFA 認証と暗号化が必要であるなど）。
- サービス固有の要件が定義、使用されている。
- AWS Marketplace ソリューションが使用されている。

SEC 9. Amazon EC2 インスタンス上のオペレーティングシステムの整合性はどのように保護していますか。

別の従来の制御として、オペレーティングシステムの整合性の保護があります。これは、従来のホストベースの手法（OSSEC、Tripwire、Trend Micro Deep Security など）を使用して EC2 で簡単に行うことができます。

ベストプラクティス:

- EC2 インスタンス用にファイル整合性のコントロールが使用されている。
- EC2 インスタンス用にホストベースの侵入検出コントロールが使用されている。
- AWS Marketplace または APN パートナーからのソリューションが使用されている。
- カスタム AMI またはデフォルトでセキュリティ保護される設定管理ツール（Puppet または Chef）を使用している。

SEC 10. AWS ログをどのように取得して分析していますか。

ログの取得は、パフォーマンスからセキュリティインシデントまですべてを調査するために重要です。最新のベストプラクティスは、ビジネスニーズに基づいて、ログを定期的にソースから直接ログ処理システム（CloudWatch Logs、Splunk、Papertrail など）に移動するか、後で処理するために Amazon S3 バケットに保存することです。ログの一般的なソースには、AWS API およびユーザー関連ログ（AWS CloudTrail など）、AWS サービス固有のログ（Amazon S3、Amazon CloudFront など）、オペレーティングシステム生成ログ、およびサードパーティー製アプリケーション固有のログがあります。Amazon CloudWatch ログを使用して、Amazon EC2 インスタンス、AWS CloudTrail、およびその他のリソースをモニタリング、保存、アクセスすることができます。

ベストプラクティス:

- AWS CloudTrail。
- Amazon CloudWatch logs。
- Elastic Load Balancing (ELB) ログ。
- Amazon Virtual Private Cloud (VPC) フィルタログ。
- Amazon S3 バケットログ。
- その他の AWS サービス固有のログソース。
- OS またはサードパーティー製アプリケーションのログ。
- AWS Marketplace ソリューションが使用されている。

信頼性の柱

REL 1. アカウントの AWS サービス制限はどのようにモニタリングしていますか。

AWS アカウントは、新しいユーザーが誤って必要以上のリソースをプロビジョニングしないように、デフォルトのサービス制限でプロビジョニングされます。AWS のお客様は AWS サービスに関するニーズを評価し、使用している各リージョンについて制限の適切な変更をリクエストする必要があります。

ベストプラクティス:

- **制限のモニタリングと管理** AWS 使用量の可能性について評価し、リージョンごとの制限を適切に引き上げ、使用量の計画的な成長を可能にします。
- **自動化されたモニタリングのセットアップ** SDK などのツールを実装して、しきい値に近づいたときに警告を発します。
- **サービスの固定の上限に注意** 変更できないサービスの上限に注意し、それらを考慮した設計を行います。

REL 2. AWS でのネットワークトポロジをどのように計画していますか。

アプリケーションは、EC2-Classic、VPC など複数の環境で実行することができます。デフォルトでは VPC 環境で実行されます。システムの接続、EIP/パブリック IP アドレス管理、VPC/プライベートアドレス管理、名前の解決などのネットワークの考慮事項は、クラウドでのリソースの活用において必須です。よく計画され、文書化されたデプロイは、重複や競合のリスクを減らすために必要不可欠です。

ベストプラクティス:

- **AWS への可用性の高い接続** 複数の DX接続、複数の VPN トンネル、AWS Marketplace アプライアンスの利用。
- **システムへの可用性の高い接続** 可用性の高い負荷分散またはプロキシ、DNS ベースのソリューション、AWS Marketplace アプライアンス等の利用。
- **重複のないプライベート IP 範囲** 仮想プライベートクラウドの IP アドレス範囲とサブネットは、他のクラウド環境やオンプレミス環境と重複しないようにする必要があります。
- **IP サブネットの割り当て** Amazon VPC の IP アドレス範囲は、将来の拡張や Multi-AZ 構成でのサブネット割り当てを考慮し、アプリケーションの要件を満たすため十分な大きさである必要があります。

REL 3. 技術的な問題に対応するためのエスカレーションパスがありますか。

お客様は AWS サポートまたは AWS パートナーを活用してください。定期的な情報交換により、既知の問題、知識のギャップ、および設計の問題に対応することができます。これにより、実装の失敗や大規模な停止のリスクを低減することができます。

ベストプラクティス:

- **計画** AWS サポートまたは APN パートナーとの継続的なエンゲージメント/関係。
- **AWS サポート API の活用** AWS サポート API を企業内のモニタリングシステムやチケットシステムと統合します。

REL 4. システムは需要の変化にどのように対応できますか。

スケーラブルなシステムは、いつの時点でも最新の需要に合わせて、リソースを自動的に追加・削除する柔軟性を備えています。

ベストプラクティス:

- **自動スケーリング** Amazon S3、Amazon CloudFront、Auto Scaling、Amazon DynamoDB、AWS Elastic Beanstalk など、自動的にスケールするサービスを使用します。
- **ロードテスト** ロードテスト手法を採用し、規模の拡大や縮小がアプリケーションの要件に合うかどうか測定します。

REL 5. AWS リソースをどのようにモニタリングしていますか。

ログとメトリックスは、アプリケーションの健全性についての洞察を得るための強力なツールです。ログとメトリックスをモニタリングし、しきい値を超えるか重要なイベントが発生したときに通知を送信するようシステムを設定できます。低パフォーマンスのしきい値を超えるか、障害が発生した場合、システムが自動的に自己修復するか、それに応じてスケールするよう構築されていることが理想的です。

ベストプラクティス:

- **モニタリング** Amazon CloudWatch またはサードパーティー製ツールでアプリケーションをモニタリングします。
- **通知** 重要なイベントが発生した場合に通知を受け取るように計画します。
- **自動応答** 自動化を使用して、障害が検出されたときに、失敗したコンポーネントを置き換えるなどのアクションを実行します。
- **レビュー** 重要なイベントに基づいてシステムのレビューを頻繁に行い、アーキテクチャを評価します。

REL 6. 変更管理をどのように実行していますか。

プロビジョニングされた AWS リソースとアプリケーションの変更管理は、アプリケーションと運用環境で既知のソフトウェアが実行されており、管理された方法でパッチを適用または置換できることを確認するために必要です。

ベストプラクティス:

- **変更管理の自動化** デプロイ/パッチ適用を自動化します。

REL 7. データをどのようにバックアップしていますか。

データ、アプリケーション、および運用環境（アプリケーションで設定されたオペレーティングシステムと定義される）をバックアップし、平均修復時間 (MTTR) および目標復旧時点 (RPO) の要件を満たします。

ベストプラクティス:

- **データのバックアップ** Amazon S3、Amazon EBS スナップショット、またはサードパーティー製ソフトウェアを使用して、RPO を満たすように重要なデータをバックアップします。
- **自動バックアップ** AWS の機能、AWS Marketplace ソリューション、またはサードパーティー製ソフトウェアを使用してバックアップを自動化します。

- **バックアップがセキュリティ保護または暗号化されている** 『AWS セキュリティのベストプラクティス』ホワイトペーパーを参照してください。
- **定期的な復旧テスト** 復旧テストを通じて、バックアップの実装が RTO および RPO を満たすことを確認します。

REL 8. システムはコンポーネントの障害にどのように対応しますか。

アプリケーションが、高可用性と短い平均修復時間 (MTTR) を満たす、暗黙または明示の要件がありますか。そのような要件がある場合は、弾力性を持つようアプリケーションを設計し、機能停止に対処できるようにアプリケーションを配置します。より高いレベルの可用性を達成するため、アプリケーションは、複数の物理的なロケーションに分散して配置される必要があります。弾力性を持つように個別の Layer（ウェブサーバー、データベースなど）を構築します。これにはイベントの重要な中断と障害のモニタリング、自己修復、および通知が含まれます。

ベストプラクティス:

- **負荷分散** リソースプールのフロントエンドにロードバランサーを使用します。
- **複数の AZ/リージョン** 複数のアベイラビリティーゾーン/リージョンにまたがってアプリケーションを配置します。
- **自動修復** 自動的に障害を検出し、復旧対応を実行します。
- **モニタリング** システムの状態を継続的にモニタリングします。
- **通知** 重要なイベントについて通知を受け取るように計画します。

REL 9. 復旧についてどのように計画していますか。

データ復旧は、バックアップからデータを復元する際に不可欠です。このデータに対して、RTO および RPO の目標と一致するように目標、リソース、場所、および機能が定義され、実行される必要があります。

ベストプラクティス:

- **目標の定義** RTO および RPO を定義します。
- **災害対策** DR 戦略を確立します。
- **構成情報の乖離** Amazon Machine Images (AMI) およびシステム設定状態が、DR サイト/リージョンで最新であることを確認します。
- **サービス制限** フェイルオーバーに対応するため、DR サイトにサービスの上限の引き上げをリクエストします。
- **DR のテストと確認** DR サイトへのフェイルオーバーを定期的にテストして、RTO と RPO が満たされていることを確認します。
- **自動復旧の実装** AWS またはサードパーティー製ツール（またはその両方）を使用してシステム復旧を自動化します。

パフォーマンスの柱

PERF 1. システムに対して適切なインスタンスタイプはどのように選択していますか。

Amazon EC2 は、さまざまなユースケースに合うように最適化された、広範なインスタンスタイプをご用意しています。インスタンスタイプはさまざまな CPU、メモリ、ストレージ、ネットワーキングキャパシティーの組み合わせによって構成されているため、使用するアプリケーションに合わせて適切なリソースの組み合わせを柔軟に選択できます。各インスタンスタイプには 1 つ以上のインスタンスサイズが含まれており、対象のワークロードの要件に応じてリソースのスケール調整が可能です。AWS は、AWS Lambda など、ワークロードのパフォーマンス効率を大きく変えることができる、サーバーレスアーキテクチャをサポートしています。

ベストプラクティス:

- **ポリシー/リファレンスアーキテクチャ** 企業内の統制標準で予測されたリソースのニーズに基づいて、インスタンスタイプとサイズを選択します。
- **コスト/予算** 企業内のコスト管理で予測されたリソースのニーズに基づいて、インスタンスタイプとサイズを選択します。

- **ベンチマーク** AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を行います（既知のパフォーマンスベンチマークと既知のワークロードのテスト）。
- **AWS または AWS パートナーネットワーク (APN) のメンバーからのガイダンス** ベストプラクティスのアドバイスに基づいて選択を行います。
- **ロードテスト** さまざまなインスタンスタイプとサイズを使用して AWS でシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 2. 新しいインスタンスタイプや機能の提供が開始される中で、どのようにして最適なインスタンスタイプを使い続けていることを担保していますか。

AWS はお客様からのフィードバックに耳を傾け、新しいインスタンスタイプとサイズでイノベーションを継続し、CPU、メモリ、ストレージ、およびネットワーキングキャパシティーの新しい組み合わせを提供します。これはつまり、最初に選択したものよりも高いパフォーマンス効率を提供する新しいインスタンスタイプがリリースされる可能性があることを意味します。

ベストプラクティス:

- **レビュー** 予測されたリソースのニーズに基づいて、新しいインスタンスタイプとサイズの選択を定期的に見直します。
- **ベンチマーク** 新しいインスタンスタイプがリリースされたら、AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を行います。
- **ロードテスト** 新しいインスタンスタイプがリリースされたら、AWS でシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 3. 起動後のインスタンスが期待どおりの性能を出すように、インスタンスをどのようにモニタリングしていますか。

内部または外部（またはその両方）の要因により、時間とともにシステムパフォーマンスが低下する場合があります。システムのパフォーマンスをモニタリングすることで、パフォーマンスの低下を検知し、内部または外部の要因（OS またはアプリケーションのロードなど）を修正することができます。

ベストプラクティス:

- **Amazon CloudWatch によるモニタリング** CloudWatch を使用してインスタンスをモニタリングします。
- **サードパーティー製品によるモニタリング** サードパーティー製ツールを使用してシステムをモニタリングします。
- **定期的な確認** モニタリングダッシュボードを定期的に確認します。
- **アラームベースの通知** メトリックスが安全な境界を超えた場合に、モニタリングシステムから自動的なアラートを受け取ります。
- **トリガーベースのアクション** アラームにより、自動化されたアクションで問題を修正またはエスカレーションします。

PERF 4. どのようにしてインスタンスの数量を需要に一致させていますか。

システムに対する需要は、サイクルによってしばしば異なります。リリース時や成長期などの製品ライフサイクル、時刻や曜日、月などの時間によるサイクル、ソーシャルメディアでの人気のような予測できないサイクル、テレビ放映のような予測可能なサイクルなどです。ワークロードに合った十分なインスタンスがないと、ユーザーエクスペリエンスが低下し、最悪の場合システム障害につながります。

ベストプラクティス:

- **計画化** メトリックスまたは計画されたイベント（またはその両方）に基づいて計画します。
- **スクリプトによる自動化** 自動管理用のツールを使用します。
- **Auto Scaling による自動化** Auto Scaling を使用して自動管理を行います。

PERF 5. システムに最適なストレージソリューションをどのように選択していますか。

AWS は、耐久性と可用性が高く低コストのデータストレージを提供するよう設計されています。AWS は、ブロックストレージ、ファイルストレージ、オブジェクトストレージに加えて、バックアップ、アーカイブ、災害対策に適したストレージも提供しています。

ベストプラクティス:

- **ポリシー/リファレンスアーキテクチャ** 企業内の統制標準によるリソースのニーズの予測に基づいて、ストレージソリューションと機能を選択します。
- **コスト/予算** 企業内のコスト管理で予測されたリソースのニーズに基づいて、ストレージソリューションと機能を選択します。
- **ベンチマーク** AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を行います（既知のパフォーマンスベンチマークと既知のワークロードのテスト）。
- **AWS または APN パートナーからのガイダンス** ベストプラクティスのアドバイスに基づいてソリューションを選択します。
- **ロードテスト** さまざまなストレージソリューションを使用して AWS でシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 6. 新しいストレージソリューションや機能が提供開始される中で、最適なストレージソリューションを使い続けていることを、どのように担保していますか。

AWS はお客様からのフィードバックに耳を傾け、新しいストレージソリューションと機能でイノベーションを継続し、キャパシティー、スループット、および耐久性の新しい組み合わせを提供します。これはつまり、最初に選択したものよりも高いパフォーマンス効率を提供する新しいストレージソリューションがリリースされる可能性があることを意味します。

ベストプラクティス:

- **レビュー** 予測されたリソースのニーズに基づいて、循環的に新しいストレージソリューションと機能を再選択します。
- **ベンチマーク** 新しいストレージソリューションと機能がリリースされたら、AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を予測します。
- **ロードテスト** 新しいストレージソリューションがリリースされたら、AWS でシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 7. ストレージソリューションが想定通りの性能を示していることを確実にするために、どのようにモニタリングしていますか。

内部または外部（またはその両方）の要因により、時間とともに、または一定の期間、システムパフォーマンスが低下する場合があります。システムのパフォーマンスをモニタリングすることで、この低下を確認し、内部または外部の要因を修正することができます。

ベストプラクティス:

- **Amazon CloudWatch によるモニタリング** CloudWatch を使用してストレージシステムをモニタリングします。
- **サードパーティー製品によるモニタリング** サードパーティー製ツールを使用してストレージシステムをモニタリングします。
- **定期的な確認** モニタリングダッシュボードを定期的に確認します。
- **アラームベースの確認** メトリックスが安全な境界を超えた場合に、モニタリングシステムから自動的なアラートを受け取るよう計画します。
- **トリガーベースのアクション** アラームにより、自動化されたアクションで問題が修正またはエスカレートされるよう計画します。

PERF 8. どのようにしてストレージソリューションのキャパシティーとスループットが需要に一致するようにしていますか。

システムに対する需要は、サイクルによってしばしば異なります。起動や成長などの製品ライフサイクル、時刻、曜日、月などの一時的なサイクル、ソーシャルメディアの可視性などの予期できないサイクル、テレビのエピソードなどの予測できるサイクルなどです。ワークロードに合った十分なストレージキャパシティーまたはスループットがないと、ユーザーエクスペリエンスが低下し、最悪の場合システム障害につながります。

ベストプラクティス:

- **リアクティブ** メトリックスに基づいて手動で管理します。
- **計画化** メトリックスまたは計画されたイベント（またはその両方）に基づいて、将来のキャパシティーおよびスループットを計画します。
- **自動化** メトリックスに対して自動化します。

PERF 9. システムに最適なデータベースソリューションをどのように選択していますか。

特定のシステムに最適なデータベースソリューションは、整合性、可用性、分断耐性、およびレイテンシーについての要件によって異なります。多くのシステムでは、さまざまなサブシステムに対して異なるデータベースソリューションを使用し、さまざまな機能を有効にしてパフォーマンスを向上させます。システムワークロードに対して誤ったデータベースソリューションや機能を選択すると、パフォーマンス効率が低下する可能性があります。

ベストプラクティス:

- **ポリシー/リファレンスアーキテクチャ** 内部の統制標準に基づいて予測されたリソースのニーズに基づいて、データベースソリューションと機能を選択します。
- **コスト/予算** 内部のコスト管理で予測されたリソースのニーズに基づいて、データベースソリューションと機能を選択します。
- **ベンチマーク** AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を行います（既知のパフォーマンスベンチマークと既知のワークロードのテスト）。
- **AWS または APN パートナーからのガイダンス** ベストプラクティスのアドバイスに基づいてソリューションを選択します。
- **ロードテスト** さまざまなデータベースソリューションと機能を使用してAWS上にシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 10. 新しいデータベースソリューションや機能が提供開始される中で、最適なストレージソリューションを使い続けていることを、どのように担保していますか。

AWS はお客様からのフィードバックに耳を傾け、新しいデータベースソリューションと機能でイノベーションを継続し、整合性、アベイラビリティー、分断耐性、およびレイテンシーの新しい組み合わせを提供します。これはつまり、最初に選択したものよりも高いパフォーマンス効率を提供する新しいデータベースソリューションまたは機能がリリースされる可能性があることを意味します。

ベストプラクティス:

- **レビュー** 予測されたリソースのニーズに基づいて、循環的に新しいデータベースソリューションと機能を再選択します。
- **ベンチマーク** 新しいデータベースソリューションと機能がリリースされたら、AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を予測します。
- **ロードテスト** 新しいデータベースソリューションと機能がリリースされたら、AWS 上にシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 11. データベースが想定通りの性能を示していることを確実にするために、どのようにモニタリングしていますか。

内部または外部の要因により、時間とともにシステムパフォーマンスが低下する場合があります。システムのパフォーマンスをモニタリングすることで、この低下を確認し、内部または外部の要因を修正することができます。

ベストプラクティス:

- **Amazon CloudWatch によるモニタリング** CloudWatch を使用してデータベースをモニタリングします。
- **サードパーティー製品によるモニタリング** サードパーティー製ツールを使用してデータベースをモニタリングします。

- **定期的な確認** モニタリングダッシュボードを定期的に確認します。
- **アラームベースの通知** メトリックスが安全な境界を超えた場合に、モニタリングシステムから自動的なアラートを受け取るよう計画します。
- **トリガーベースのアクション** アラームにより、自動化されたアクションで問題が修正またはエスカレートされるよう計画します。

PERF 12. どのようにしてデータベースのキャパシティーとスループットが需要に一致するようにしていますか。

システムに対する需要は、サイクルによってしばしば異なります。起動や成長などの製品ライフサイクル、時刻、曜日、月などの一時的なサイクル、ソーシャルメディアなどに見られる予期できないサイクル、テレビのエピソードなどの予測できるサイクルなどです。ワークロードに合った十分なデータベースキャパシティーおよびスループットがないと、ユーザーエクスペリエンスが低下し、最悪の場合システム障害につながります。

ベストプラクティス:

- **計画化** メトリックスまたは計画されたイベント（またはその両方）に基づいて、将来のキャパシティーおよびスループットを計画します。
- **自動化** メトリックスに対して自動化します。

PERF 13. システムに最適な近接性およびキャッシュソリューションをどのように選択していますか。

物理的な距離、ネットワークの距離、または長時間実行されているリクエストは、システムの遅延を発生させる可能性があります。レイテンシーに対応しないと、必要以上に長くシステムリソースを拘束し、内部および外部のパフォーマンスの低下を引き起こす可能性があります。レイテンシーを減らすには、エンドユーザーの観点からシステム全体のエンドツーエンドのパフォーマンスを考慮し、リソースまたはキャッシュソリューションを物理的に近づけるように調整する機会を探します。

ベストプラクティス:

- **ポリシー/リファレンスアーキテクチャ** 内部の統制標準に基づいて予測されたリソースのニーズに基づいて、近接性およびキャッシュソリューションを選択します。
- **コスト/予算** 内部のコスト管理で予測されたリソースのニーズに基づいて、近接性およびキャッシュソリューションを選択します。
- **ベンチマーク** AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を行います（既知のパフォーマンスベンチマークと既知のワークロードのテスト）。
- **AWS または APN パートナーからのガイダンス** ベストプラクティスのアドバイスに基づいて近接性およびキャッシュソリューションを選択します。
- **ロードテスト** さまざまな近接性およびキャッシュソリューションを使用して AWS 上にシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択を行います。

PERF 14. 新しいソリューションや機能が開始される中で、最適な近接性およびキャッシュソリューションを使い続けることを、どのように担保していますか。

AWS はお客様からのフィードバックに耳を傾け、新しい近接性およびキャッシュソリューションと機能でイノベーションを継続し、近接性、キャッシュ、分断耐性、およびレイテンシーの新しい組み合わせを提供します。これはつまり、最初に選択したものよりも高いパフォーマンス効率を提供する新しい近接性およびキャッシュソリューションがリリースされる可能性があることを意味します。システム全体でレイテンシーを減らし、パフォーマンスを向上させる機会を探します。たとえば、1 回のみの最適化を完了したのか、時間とともに需要が変わるにつれてシステムの最適化を継続するかです。

ベストプラクティス:

- **レビュー** 予測されたリソースのニーズに基づいて、循環的に新しい近接性およびキャッシュソリューションを再選択します。
- **ベンチマーク** 新しい近接性およびキャッシュソリューションがリリースされたら、AWS で既知のワークロードのロードテストを行い、それを使用して最適な選択を予測します。
- **ロードテスト** 新しい近接性およびキャッシュソリューションがリリースされたら、AWS 上にシステムの最新バージョンをデプロイし、モニタリングを使用してパフォーマンスメトリックスをキャプチャしてから、パフォーマンス/コストの計算に基づいて選択します。
- **積極的モニタリング – Amazon Cloud Watch によるモニタリング** Amazon CloudWatch を使用して近接性およびキャッシュソリューションをモニタリングします。
- **積極的モニタリング – サードパーティー製品によるモニタリング** サードパーティー製のツールを使用して近接性およびキャッシュソリューションをモニタリングします。
- **アラームベースの通知** メトリックスが安全な境界を超えた場合に、モニタリングシステムから自動的なアラートを受け取るよう計画します。
- **トリガーベースのアクション** アラームにより、自動化されたアクションで問題が修正またはエスカレートされるよう計画します。

PERF 15. パフォーマンスを予期どおりのものとするには、近接性およびキャッシュソリューションをどのようにモニタリングしていますか。

内部または外部の要因により、時間とともにシステムパフォーマンスが低下する場合があります。システムのパフォーマンスをモニタリングすることで、この低下を確認し、内部または外部の要因を修正することができます。

ベストプラクティス:

- **Amazon CloudWatch によるモニタリング** CloudWatch を使用してインスタンスをモニタリングします。
- **サードパーティー製品によるモニタリング** サードパーティー製ツールを使用してシステムをモニタリングします。
- **定期的な確認** モニタリングダッシュボードを定期的に確認します。
- **アラームベースの通知** メトリックスが安全な境界を超えた場合に、モニタリングシステムから自動的なアラートを受け取るよう計画します。
- **トリガーベースのアクション** アラームにより、自動化されたアクションで問題が修正またはエスカレートされるよう計画します。

PERF 16. どのようにして近接性およびキャッシュソリューションを需要に一致させていますか。

システムに対する需要は、サイクルによってしばしば異なります。起動や成長などの製品ライフサイクル、時刻、曜日、月などの一時的なサイクル、ソーシャルメディアなどに見られる予期できないサイクル、テレビのエピソードなどの予測できるサイクルなどです。ワークロードに合っていない近接性およびキャッシュソリューションがあると、ユーザーエクスペリエンスが低下し、最悪の場合システム障害につながります。これは特に、グローバルユーザーベースを持っているか、持つ計画の場合に該当します。

ベストプラクティス:

- **計画化** メトリックスまたは計画されたイベント（またはその両方）に基づいて、将来の近接性またはキャッシュソリューションを計画します。
- **モニタリング** 時間とともにキャッシュの使用量と需要をモニタリングします。
- **定期的な確認** 時間と共に変化するキャッシュの使用量と需要を確認します。

コストの最適化の柱

COST 1. キャパシティーが必要量を満たしているが大幅に超えていないことをどのように実現していますか。

料金とパフォーマンスの点でバランスが取れたアーキテクチャを構築するには、支払ったすべてのものが使用されるようにし、著しく使用率が低いインスタンスを避けます。いずれかの方向に偏った使用率のメトリックスは、運用コスト（過度の使用率によるパフォーマンスの低下）または無駄な AWS 支出（オーバープロビジョニングによる）によるビジネスへの悪影響につながります。

ベストプラクティス:

- **デマンドベースの手法** Auto Scaling を使用して変化する需要に対応します。
- **キューベースの手法** 個別に Amazon Simple Queue Service (SQS) キューを実行し、需要に基づいてインスタンスを回転/シャットダウンします。
- **タイムベース手法**の例:24 時間 365 日の運用において、週末や四半期、年間スケジュールに基づいて開発/テストのインスタンスを停止します（たとえば年末年始の休暇、ブラックフライデーなど）。
- **適切にプロビジョニング** Amazon DynamoDB、Amazon EBS（プロビジョンドIOPS）、Amazon RDS、Amazon EMR のようなサービスのスループット、サイジング、ストレージについて適切にプロビジョニングを行います。

COST 2. AWS サービスの使用量をどのようにして最適化していますか。

アプリケーションレベルのサービスを使用する場合は、これらが適切に使用されているのか確かめます。たとえば、ライフサイクルポリシーを導入して Amazon S3 使用率を管理したり、Amazon RDS や Amazon DynamoDB などのサービスを利用して柔軟性を大幅に高めます。適切な使用率の確認には、Amazon RDS 用のマルチ AZ 配置の確認や、Amazon DynamoDB テーブルでプロビジョンド IOPS が該当することの確認が含まれます。

ベストプラクティス:

- **サービス固有の最適化** 例には、Amazon EBS 用の I/O の最小化、小さなファイルを大量に Amazon S3 にアップロードすることの回避、Amazon EMR に対するスポットインスタンスの広範な使用などがあります。

COST 3. コスト目標を達成するために適切なリソースを選択していますか。

選択した Amazon EC2 インスタンスが、タスクに対して適切であることを確認します。AWS では、ベンチマーク評価の使用によって、選択したインスタンスタイプがそのワークロードに対して最適化されていることを確認します。

ベストプラクティス:

- **ニーズに基づいてインスタンスプロファイルを一致させます。** たとえば、ワークロードやインスタンスの特性（コンピューティング、メモリ、またはストレージに最適化）に基づいた一致があります。
- **サードパーティー製品** たとえば、CopperEgg または New Relic などのサードパーティ製品を使用して、適切なインスタンスタイプを判断します。
- **Amazon CloudWatch** CloudWatch を使用してプロセッサの負荷を判断します。
- **カスタムメトリックス** カスタムメモリスクリプトをロードし、CloudWatch を使用してメモリ使用量を検査します。
- **プロファイルされたアプリケーション** アプリケーションをプロファイルし、Amazon EBSのどのタイプ（マグネティック、汎用 (SSD)、プロビジョンド IOPS）をいつ使用するかについて理解します。EBS 最適化インスタンスは、必要なときのみ使用します。

COST 4. コスト目標を達成するために適切な料金モデルを選択していますか。

ワークロードで費用を最小化するために最適な料金モデルを使用します。デプロイを最適化すると、全てオンデマンドインスタンス、オンデマンドとリザーブドインスタンスの組み合わせとするか、そして、可能な場合はスポットインスタンスを含めることもできます。

ベストプラクティス:

- **スポット** ワークロードによってスポットインスタンスを使用します。
- **使用量の分析** 定期的に使用量を分析して、それに応じてリザーブドインスタンスを購入します。
- **リザーブドインスタンスの販売** ニーズの変化に合わせて、必要なくなったリザーブドインスタンスをリザーブドインスタンス Marketplace で販売し、他のインスタンスを購入します。
- **自動化されたアクション** アーキテクチャで、使用されていないインスタンスをオフにします（たとえば、Auto Scaling を使用して非稼働時間中にスケールダウンします）。
- **コスト要因** リージョンの選択でコストを考慮します。

COST 5. ROI を改善するためにマネージドサービス（Amazon EC2 や Amazon EBS、Amazon S3 よりも高いレベルのサービス）を選択していますか。

Amazon EC2、Amazon EBS、および Amazon S3 は AWS サービスの基本的な「構成要素」です。Amazon RDS や Amazon DynamoDB などのマネージドサービスは、「高レベル」の AWS サービスです。これらのマネージドサービスを使用することで、管理オーバーヘッドまたは運用オーバーヘッドの大部分を減らすか排除することができ、アプリケーションやビジネス関連のアクティビティに労力を向けることができます。

ベストプラクティス:

- **サービスの分析** アプリケーションレベルのサービスを分析して、使用できるサービスを確認します。
- **適切なデータベースの検討** 該当する場合は、Amazon Relational Database Service (RDS) (Postgres、MySQL、SQL Server、Oracle Server) または Amazon DynamoDB（またはその他のキーと値のストア、NoSQL 代替策）を使用します。
- **その他のアプリケーションレベルのサービスの検討** 該当する場合は、Amazon Simple Queue Service (SQS)、Amazon Simple Notification Service (SNS)、Amazon Simple Email Service (SES) を使用します。
- **AWS CloudFormation、AWS Elastic Beanstalk、または AWS Opsworks の検討** AWS CloudFormation テンプレート / AWS Elastic Beanstalk /AWS OpsWorks を使用して、標準化とコスト管理の利点を達成します。

COST 6. AWS の使用状況を管理するためにどのような管理と手順を実行していますか。

目標の達成のために適切なコストとなるようなポリシーとメカニズムを確立します。タグ付けと IAM コントロールを通じてチェックとバランスの手法を使用すると、浪費することなくイノベーションが可能になります。

ベストプラクティス:

- **グループおよびロールの確立**（例: 開発/テスト/本稼働）IAMのような AWS ガバナンスメカニズムを使用して、各々のグループのインスタンスとリソースをスピンアップできるのが誰かを管理します。（これは AWS サービスまたはサードパーティーのソリューションに適用されます）。
- **プロジェクトライフサイクルの追跡** プロジェクト、チーム、および環境のライフサイクルを追跡、測定、監査して、不要なリソースの使用と支払いを避けます。

COST 7. 使用状況と支出をどのようにモニタリングしていますか。

コストをモニタリング、管理、適切に割り当てるためのポリシーと手順を確立します。AWS 提供のツールを活用して、だれが何をどのぐらいのコストで使用しているのかを明らかにします。これにより、ビジネスニーズとチームのオペレーションについてより深く理解することができます。

ベストプラクティス:

- **すべてのリソースにタグ付け** 請求の変化をインフラストラクチャと使用量の変更に関連付けられるようにします。
- **詳細な請求レポートの確認** 詳細な請求レポートをロードし、解釈できるための標準プロセスを用意します。
- **コスト効率が高いアーキテクチャ** 使用量と費用の両方について計画を持ちます（ユーザー単位、ギガバイトのデータ単位など）。
- **モニタリング** Amazon CloudWatch またはサードパーティー製品（例：Cloudability、CloudCheckr）を使用して定期的に使用量と費用をモニタリングします。
- **通知** 費用が明確に定義された限度を超える場合に、チームの主要メンバーに知らせます。
- **AWS コストエクスプローラーの使用**
- **請求の配賦方法** これを使用して、インスタンスとリソースをコストセンターに割り当てます（タグ付けなど）。

COST 8. 必要なくなったリソースは廃棄していますか。または、一時的に必要ないリソースを停止していますか。

使用しているサービスについてのみ支払いを行っていることを確認します。プロジェクトの開始から終了まで変更管理とリソース管理を実装し、適切な場合は必要なプロセス変更または機能強化を識別できるようにします。AWS サポートと連携して、ワークロードに対してプロジェクトを最適化する方法の推奨事項を得ます。たとえば、いつ Auto Scaling、AWS OpsWorks、AWS Data Pipeline、または別の Amazon EC2 プロビジョニング手法を使用するかについてです。

ベストプラクティス:

- 重要でないインスタンスや不要なインスタンス、または使用率が低いリソースを確認して廃棄する際に、インスタンスの終了を適切に処理するようシステムを設計します。
- 既に使われていないリソースを確認して廃棄するプロセスを導入します。
- システムまたはプロセスに基づいて、廃棄したリソースを照合します。

COST 9. アーキテクチャを設計するときに、データ転送料金について考慮しましたか。

データ転送料金をモニタリングし、それらのコストの一部を軽減できるようなアーキテクチャの決定を行います。たとえば、コンテンツプロバイダーがエンドユーザーに対して Amazon S3 バケットから直接コンテンツを提供している場合、コンテンツを Amazon CloudFront CDN にプッシュすれば、大幅なコスト削減が可能になる可能性があります。小規模でも効果的なアーキテクチャ変更によって、運用コストを大幅に削減できる場合があることを覚えておいてください。

ベストプラクティス:

- CDN を使用します
- データ転送を最適化するよう設計します (アプリケーション設計、WAN アクセラレーションなど)。
- 状況を分析し、AWS Direct Connect を使用して費用を節約し、パフォーマンスを向上させます。
- 高可用性 (HA) および信頼性のニーズでアーキテクチャのデータ転送コストのバランスをとります。

COST 10. 新しいサービスの採用をどのように管理または検討していますか。

AWS の目標は、お客様が可能な限り最適に、高いコスト効率でアーキテクチャを作成できるよう支援することです。新しいサービスや機能によって直接コストを削減できる場合もあります。この良い例として Amazon Glacier があります。これは頻繁にアクセスされないが、ビジネスまたは法律上の理由で維持する必要があるデータに対する、低コストの「コールド」ストレージソリューションを提供します。別の例として、Amazon S3 用低冗長化ストレージがあります。これにより、Amazon S3 オブジェクトのコピー数を減らし（冗長化レベルを下げて）、コストを削減できます。これらの決定に際しては、考慮すべき影響があります。たとえば、「データのコピー数を減らすことは何を意味するのか」や、「このデータには、自分が考えているよりも頻繁にアクセスするのか」などです。

ベストプラクティス:

- AWS ソリューションアーキテクト、コンサルタント、またはアカウントチームと定期的に会って、費用を節約するために採用できる新しいサービスや機能について検討します。