モジュール2、3レポート

* **Amazon Elastic Compute Cloud(Amazon EC2) -** 仮想サーバーへのアクセスに使用するサーバー、安全でサイズ調整が可能なコンピューティング容量をAmazon EC2 インスタンスとしてクラウドで提供します。
* オンプレミス リソースを使用する場合
  + 事前にハードウェアを購入する必要があります。
  + サーバーが配達されるまで待たなければなりません。
  + 物理データセンターにサーバーを設置する必要があります。
  + 必要なすべての構成を実行する必要があります。
* Amazon EC2 インスタンスを使用する場合 - AWSクラウドで仮想サーバーを使用してアプリケーションを実行することができます。
  + 数分でAmazon EC2 インスタンスをプロビジョニングして始められます。 (プロビジョニング - ユーザが要請したITリソースを使用可能な状態で準備すること)
  + ワークロードの実行が完了した場合は、インスタンスの使用を中止することができます。(ワークロード - 顧客対面のアプリケーションやバックエンドプロセスなどのビジネス価値を生み出すリソースとコードのコレクション)
  + インスタンスが実行中に使用したコンピューティング時間に対してのみ費用を支払い、インスタンスが中止または終了した状態では、費用は支払いません。
  + 必要なサーバーの容量に対してのみコストを支払うことでコストを削減できます。
* Amazon EC2 作動 方式
  + 始め - デフォルトの構成インスタンスが含まれているテンプレートを選択して開始します。 これらの構成には、オペレーティングシステム、アプリケーションサーバー、またはアプリケーションが含まれます。 (インスタンス - awsクラウドで使用する仮想コンピュータ、CPU、メモリ、グラフィックカードなど演算のためのハードウェア部分を担当)
  + 連結 - 次に、インスタンスに 連結します。 プログラムとアプリケーションには、インスタンスに直接接続し、データを交換するいくつかの方法があります。
  + 使用 - コマンドを実行して、ソフトウェアのインストール、ストレージの追加、ファイルのコピー、およびクリーンアップなどのタスクを実行できます。
* Amazon EC2 インスタンスタイプ - インスタンスタイプを選択するときは、ワークロードとアプリケーションの具体的な要求事項を考慮します。これには、コンピューティング、メモリ、またはストレージ機能の要求事項が含まれる場合があります。
  + 汎用インスタンス - コンピューティング、メモリ、ネットワーキングリソースをバランスよく提供します。 次のようなさまざまなワークロードに使用できます。
    - アプリケーションサーバー
    - ゲームサーバー
    - エンタープライズアプリケーション用のバックエンドサーバー
    - 中小規模データベース
  + コンピューティング最適化インスタンス - 高性能プロセッサを活用するコンピューティング集約的なアプリケーションに適しています。
  + メモリ最適化インスタンス - メモリから大規模なデータセットを処理するワークロードのための高速なパフォーマンスを提供するために設計されました。
  + アクセラレーテッドコンピューティングインスタンス - ハードウェアアクセラレータまたはコプロセッサを使用して、一部の機能をCPUで実行されているソフトウェアでより効率的に実行します。 (ハードウェアアクセラレータ - データ処理を高速で向上させるハードウェア) (コプロセッサ - CPUの機能を補うために使用されるコンピュータプロセッサ)
  + ストレージ最適化インスタンス - ローカルストレージの大規模データセットに対する高いシーケンシャル読み取りおよび書き込みアクセスを必要とするワークロード用に設計されています。
* Amazon EC2料金 - Amazon EC2 では、使用したコンピューティング時間に対してのみ費用を支払います。
  + オンデマンド - 中断できない不規則な短期ワークロードを持つアプリケーションに最適です。
  + Amazon EC2 Savings Plans - 1年または3年間、一定のコンピューティング使用量を約定することで、コンピューティングコストを削減できます。この期間約定により、オンデマンド料金に比べて最大72%までコストを削減できます。
  + 予約インスタンス - アカウントからオンデマンド インスタンスを使用する場合に適用される支払い割引オプションです。
  + スポットインスタンス - 開始および終了時間が自由であるか、中断に耐えられるワークロードに適しています。
  + 専用ホスト - ユーザー専用のAmazon EC2 インスタンス容量を備えた物理的サーバーです。
* Amazon EC2 Savings Plansとスポットインスタンスの違いは何ですか？ - Amazon EC2 Savings Plansとは異なり、スポットインスタンスには契約や一定のコンピューティング使用量に対する約定は必要ありません。
* スケーラビリティ- スケーラビリティのためには、必要なリソースだけで開始し、スケーラビリティと縮小によって需要の変化に自動的に対応するようにアーキテクチャを設計する必要があります。 その結果、使用したリソースに対してのみ費用を支払います。
* Amazon EC2 Auto Scaling - 変化するアプリケーションの需要に応じて、Amazon EC2 インスタンスを自動的に追加または削除できます。 必要に応じてインスタンスを自動的に調整することで、アプリケーションの可用性を効果的に維持できます。
  + 動的スケーリング - 需要の変化に対応します。
  + 予測スケーリング - 予測された需要に応じて、適切な数のAmazon EC2 インスタンスを自動的に予約します。
* Auto Scaling グループ
  + 最小キャパシティ - 最小グループサイズを表します。
  + 希望するキャパシティ - Auto Scaling グループ作成時の初期キャパシティを表します。
  + 最大キャパシティ - 最大グループサイズを表します。
* Elastic Load Balancing - 入ってくるアプリケーショントラフィックをAmazon EC2 インスタンスなどの複数のリソースに自動的に分散するAWS サービスです。 入ってくるトラフィックの量に合わせて Amazon EC2 インスタンスを追加または削除するため、これらのリクエストがロード バランサに最初にルーティングされます。 (ルーティング - あるネットワーク内で通信データを送信する際に最適な経路を選択するプロセス)
* モノリシック·アプリケーション - コンポーネントが密結合されたアプリケーションです。 あるコンポーネントで障害が発生すると、別のコンポーネントで障害が発生し、アプリケーション全体で障害が発生する可能性があります。
* マイクロサービス - アプリケーション コンポーネントが小結合されます。 この場合、単一のコンポーネントに障害が発生しても、他のコンポーネントは互いに通信するため、引き続き動作します。
* Amazon Simple Notification Service(Amazon SNS) - 発行/受信サービスです。発行者は、Amazon SNSのトピックを使用して、受信者にメッセージを発行します。 Amazon SNSでは、 受信 者はウェブサーバー、メールアドレス、AWS Lambda関数、またはその他の複数のオプションになることがあります。
* Amazon Simple Queue Service(Amazon SQS) - メッセージキュー サービスです。 メッセージの損失やその他のサービスを使用することなく、ソフトウェア コンポーネント間でメッセージを送信、保存、受信できます。 Amazon SQSでは、アプリケーションがメッセージをキューに転送、 ユーザーまたはサービスは、キューでメッセージを検索して処理し、キューから削除します。
* サーバーレス・コンピューティング - コードはサーバーで実行されますが、これらのサーバーをプロビジョニングまたは管理する必要はありません。 サーバレスコンピューティング用のAWSサービスはAWS Lambdaです。
* AWS Lambda - サーバーをプロビジョニングしたり、管理したりする必要がなく、コードを実行できるサービスです。 AWS Lambdaを使用する場合、使用したコンピューティング時間に対してのみ費用を支払います。 コードを実行している間だけ料金がかかります。 事実上、あらゆるタイプのアプリケーションまたはバックエンドサービスコードを実行することができ、これを管理する必要はまったくありません。
* AWS Lambda作動方式
  + コードをLambdaにアップロードします。
  + AWS サービス、モバイルアプリケーション、またはHTTPエンドポイントなどのイベントソースでトリガーされるようにコードを設定します。 (トリガー - ラムダ関数を実行できるイベント)
  + Lambdaはトリガーされた場合にのみコードを実行します。
  + 使用したコンピューティング時間の料金のみを支払います。
* コンテナ - アプリケーションのコードと依存性を1つのオブジェクトにパッケージングする標準的な方法を提供します。コンテナオーケストレーションサービスは、コンテナ式アプリケーションの配布、管理、拡張に役立ちます。
  + Amazon Elastic Container Service(Amazon ECS) - AWSでコンテナ式アプリケーションを実行し、拡張できる拡張性に優れた高性能コンテナ管理システムです。Amazon ECSはDockerコンテナをサポートします。 (Docker - アプリケーションを迅速に構築、テスト、配布できるソフトウェアプラットフォーム)
  + Amazon Elastic Kubernetes Service(Amazon EKS) - AWSでKubernetesを実行するに使用できる完全管理型サービスです。 (Kubernetes - コンテナ式アプリケーションの大規模な配布と管理に使用できるオープンソースソフトウェアです。)
  + AWS Fargate - コンテナ用のサーバーレスコンピューティングエンジンで、Amazon ECSとAmazon EKSで動作します。AWS Fargateを使用している場合、サーバーをプロビジョニングまたは管理する必要はありません。
* Docker, Kubernetes違い - Dockerはコンテナ化技術を提供するプラットフォームであり、 Kubernetesはコンテナオーケストレーションツールです。
* AWSリージョン - AWSで提供されるクラウドコンピューティングリソースは、世界各国でホスティングされます。 この時、このホスティングされる位置を地理的観点から領域を区分したのがリージョンです。
* リージョン選択の4つの重要な要因
  + 規定遵守 - 会社と場所によっては、特定の領域でデータを実行する必要がある場合があります。
  + 近接性 - 顧客に近いリージョンを選択すると、顧客にコンテンツをより迅速に提供するのに役立ちます。
  + 機能可用性 - 場合によっては、お客様に提供しようとする機能が最寄りのリージョンにないこともあります。開発者は、すでにサービスを提供しているリージョンの 1 つでこのプラットフォームを実行する必要があります。
  + 価格 - サービスの費用は、ビジョンによって異なる場合があります。
* アベイラビリティーゾーン - リージョン内の単一のデータセンターまたはデータセンターグループです。アベイラビリティーゾーンは互いに数十マイル離れています。この間隔は、アベイラビリティーゾーン間の遅延時間が短いほど十分に近いです。しかし、リージョンの一部分で災害が発生した場合、複数のアベイラビリティーゾーンが影響を受ける可能性を減らすほど遠く離れています。
* エッジロケーション - Amazon Cloud Frontがより迅速なコンテンツ配信のために、お客様に近い位置にコンテンツのコピーをキャッシュするために使用するサイトです。 (Amazon Cloud Front - データ、動画、アプリケーション、APIを世界中のお客様に短い遅延時間で迅速に配信するサービス)
* AWSサービスと相互作用する方法
  + AWS Management Console - Amazonサービスへのアクセスと管理のためのウェブベースのインターフェースです。
  + AWSコマンドラインインターフェイス(AWS CLI) - AWSサービスを管理する統合ツールです。 1つのツールだけをダウンロードして構成すると、複数のAWSサービスをコマンドラインでコントロールし、スクリプトを通じて自動化することができます。
  + ソフトウェア開発キット(SDK) - SDKを使用すると、プログラミング言語またはプラットフォーム用に設計されたAPIを通じて、AWSサービスをより簡単に使用することができます。
* AWS Elastic Beanstalk - Amazon EC2 ベースの環境プロビジョニングをサポートするサービスです。AWS Elastic Beanstalkでは、ユーザーがコードと構成設定を提供すると、Elastic Beanstalkが次の作業を実行するのに必要なリソースを配布します。
  + 容量のプロビジョニング
  + ロードバランシング
  + スケーリング
  + アプリケーション状態モニタリング
* AWS CloudFormation - AWS リソースのモデル化およびセットアップに役立つサービスです。リソース管理に割く時間を減らし、AWSで実行するアプリケーションにさらに注力できるようになります。