

## AWS Skill Builder 2週目の勉強内容まとめ

**AWSには非管理型サービスであるEC2の他にも、ユーザーの責任を軽減するさまざまな管理型サービスが存在する。**

EventBridgeやSNS、SQSも管理型サービスの一種であり、Lambdaという顧客の固定責任以外のすべての部分をAWSが責任を持つ完全管理型サービスも存在する。

### **Lambdaとは？**

ユーザーがサーバーをプロビジョニング・管理する必要がなく、イベントにตอบสนองしてコードを実行するサーバーレスコンピューティングサービス(サービス型関数)である。

Lambdaはインフラの管理が完全に自動で(リソースもボリュームに応じて自動調整され)、使用したコンピューティング時間に対して料金が発生する。

そのため、Lambdaを使用する場合、ユーザーはコードに集中できる利点がある。

また、Lambdaはすべてのプログラミング言語をサポートしており、関数の最大サポート時間は15分である。

### **Lambdaはどうやって動くのか？**

1. まずユーザーがLambdaにコードをアップロードすると、自動的にLambda関数にアップロードされる。
2. 次に、AWSサービスやHTTPなどのリクエストイベントでトリガーされるようにコードを構成する。
3. トリガーが発動するとコードが実行され、Lambdaサービスがサーバー、インフラ、スケーリングなどすべての管理を担当し、効率的に実行する。

その後、ユーザーには使用したコンピューティング時間に対してのみ料金が課金される。

※料金はユーザーが関数に割り当てるメモリ容量によって異なる場合がある。

### **Lambdaの使用例**

SNSアプリのリアルタイム画像処理: アップロードされた画像に応じて自動的にスケールを調整し、各画像の処理時間に対してのみ料金が発生するため、合理的である。

ニュース推薦サービスのパーソナライズされたコンテンツ推薦: ユーザー数に応じてトラフィック量を調整し、ユーザーがサービスを利用する時だけコードが実行されるため合理的である。

オンラインゲームのリアルタイムイベント処理: 運営者がサーバーを管理する必要なく、数千件のイベントリクエストをリアルタイムで処理し、使用量に応じて規模を自動調整するため特にアクセス率が上がるピーク時に有用である。

## コンテナとは？

コンテナはどこでも同じ環境を提供し、移植性が保証されるクラウドシステムの一つである。  
また、コンテナはホストコンピュータのOSを共有するため、ハイパーバイザーを通じて全体のオペレーティングシステムが実行される仮想マシン(VM)に比べて  
軽量化されており、起動時間が短い。ただし、物理的に空間が隔離されているVMに比べて、コンテナは論理的に隔離されているだけなので  
セキュリティ上の問題が生じる可能性がある。

## AWSではコンテナコンピューティングサービスも提供している。

Amazon Elastic Container Service(ECS):AWSベースのコンテナオーケストレーションサービス(例:Docker)

Amazon Elastic Kubernetes Service(EKS):AWS上でKubernetesを実行するためのフルマネージドシステム

\*Kubernetes(クーバネティス):Googleが開発したオープンソースのコンテナオーケストレーションサービス

Amazon Elastic Container Registry(ECR):AWSが提供する完全管理型コンテナプライベートリポジトリ、  
CLIを通じてコンテナイメージを管理し、ECSやEKSを実行する際にここからイメージを取得する

Fargate:AWSが提供するサーバーレスコンピューティングエンジン

## その他にも、AWSにはElastic Beanstalk、AWS Batch、Lightsail、Outpostsなどのコンピューティングサービスが提供されている。

Elastic Beanstalk:ウェブアプリケーションのデプロイとスケーリングのためのマネージドサービス

AWS Batch:バッチコンピューティングワークロード向けのフルマネージドサービス

Lightsail:仮想プライベートサーバー(VPS)、ストレージ、ネットワーキングを備えた簡素化されたサービス

Outposts:AWSをオンプレミスに拡張するハイブリッドクラウドサービス

## AWSを通じてグローバル展開を計画しているなら、リージョンを選ぶ際にいくつか注意すべき点がある。

\*リージョンまたはサービスを提供する地域の規制を遵守する必要があります(例:EUのGDPR)

\*ターゲットユーザー層に物理的に近いリージョンを選択することで、応答性とシステム効率を向上させる

ことができる。

\*リージョンによっては、AWSの一部機能が制限または提供されない場合がある。

\*リージョンが位置する地域の税制などにより、合理的な料金体系が提供されるなど、料金が異なる場合がある。

**高可用性アーキテクチャを構成するためには、さまざまなリージョンに展開するだけでなく、さまざまなアベイラビリティゾーン(AZ)に展開することも考慮する必要がある。**

**マルチリージョンおよびアベイラビリティゾーンを利用することで得られる利点は？**

\*高可用性:どの可用領域で問題が発生しても、すぐに他の可用領域に置き換わることで、スムーズな顧客体験を提供し

ユーザーはシステムを停止させることなく障害処理を行えるようになる。

\*敏捷性:複数のグローバルリージョンとアベイラビリティゾーンを活用することで、急速に変化するグローバル市場の状況に迅速に対応できる。

\*弾力性:AWSでは需要に応じてリソースを自動的に拡大または縮小するサービスがあるため、弾力性が良い。

また、必ずしもリージョンを構築しなくても、エッジロケーションを活用することで、地域外の顧客に対してキャッシュを利用したサービス提供の遅延時間を短縮できる。

エッジロケーションはAZに直接接続されず、AWSのCDN(コンテンツデリバリーネットワーク)コンテンツ配信ネットワークの一部であり、コンテンツ配信に特化している。

**AWSのAPIを呼び出す方法にはCloudFormationも存在する。Management Consoleと共にIac(Infrastructure as Code)コード型インフラと呼ばれており複雑なインフラ設定をコードで定義し、開発・テスト・本番といった様々な環境でのデプロイの一貫性を保証する。**