●演習問題・解答

問題

問題は選択式（単一もしくは複数）です。

Q 演習問題

1 MySQLデータベースがボトルネックになるのを防ぐために効果的な方法は、次のうちどれですか（2つ選択）。

A ELBのClassic Load BalancerをWebアプリケーション層置する

B Amazon RDSのリードレプリカを利用する

解答

解答番号は問題番号と同一です。正解の内容は当該節で解説されています。

A 解答

1 B、C

Amazon RDSのリードレプリカを利用することで、読み取り頻度の高いデータベースを増設できるため、読み取りスループットを増やし、パフォーマンスを向上させることができます。Amazon ElastiCacheはマネージド型のインメモリデータベースです。メモリ上で処理を実行するため、データをキャッシュすることで、データベースの負荷軽減を実現します。したがって、**B**、**C**が正解です。

- 本書に記載されているAWSサービスの名称や内容、料金、グローバル インフラストラクチャの数などは、特に注記がない限り執筆時点のものです。
- 本書の説明文中に記載された設定値などの最大数はデフォルト値のため、上限緩和申請によって変更できる場合があります。
- 読者特典の模擬問題は、AWSの公式模擬試験問題ではありません。

●解説の図

解説には多数の図を掲載し、システムやアーキテクチャのイメージをわかりやすく示しています。図中に登場するサービスアイコンの名称は、略称で表記しています。同一のグループアイコンやサービスアイコンが頻出したり類似する図が連続する場合は、初出時に表記した名称を省略していることがあります。また、グループが含まれる図では、名称が記載されなくても説明できる場合は、一部のグループを省略しています。

なお、本文中の図で使用しているAWSのグループとサービスのアイコンおよび名称は、「AWS Architecture Icons (Release 14-2022.07.31)」に準拠しています。

目次
CONTENTS

まえがき 3
AWS認定について 4
AWS認定ソリューションアーキテクトーアソシエイトについて 5
本書の活用方法 6

第1章 AWSサービス全体の概要 11

1-1 AWS Well-Architectedフレームワーク 12
1-2 ネットワークサービス 16
1-3 アクセス制御サービス 38
1-4 コンピューティングサービス 43
1-5 ストレージサービス 71
1-6 データベースサービス 90
1-7 データ処理・分析サービス 119
1-8 構成管理サービス・開発サービス 129
1-9 運用管理サービス 138

第2章 AWSにおけるセキュリティ設計 149

2-1 AWSにおけるセキュリティ設計の考え方 150
2-2 アイデンティティ管理とアクセス管理 155
2-3 ネットワークセキュリティ 171
2-4 データの保護 184
2-5 セキュリティ監視 195

第3章 AWSにおける高可用アーキテクチャ 205

3-1 高可用性の定義 206
3-2 ネットワークにおける高可用性の実現 212
3-3 コンピューティングにおける高可用性の実現 221
3-4 ストレージにおける高可用性の実現 243
3-5 データベースにおける高可用性の実現 254

第4章　AWSにおけるパフォーマンス　261

4-1　AWSにおけるパフォーマンスの考え方 262
4-2　ネットワークサービスにおけるパフォーマンス 265
4-3　コンピューティングサービスにおけるパフォーマンス 274
4-4　ストレージサービスにおけるパフォーマンス 283
4-5　データベースサービスにおけるパフォーマンス 288

第5章　AWSにおけるコスト最適化　299

5-1　AWSにおけるコスト最適化の考え方 300
5-2　コスト効果が高いリソースの選定 305
5-3　需要と供給のマッチングによるコスト最適化 318
5-4　コストの管理 326

索引 334

AWS
Solutions Architect–Associate

第1章

AWSサービス全体の概要

1-1　AWS Well-Architectedフレームワーク
1-2　ネットワークサービス
1-3　アクセス制御サービス
1-4　コンピューティングサービス
1-5　ストレージサービス
1-6　データベースサービス
1-7　データ処理・分析サービス
1-8　構成管理サービス・開発サービス
1-9　運用管理サービス

1-1 AWS Well-Architected フレームワーク

アマゾン ウェブ サービス（AWS：Amazon Web Services）では「AWS Well-Architectedフレームワーク」と呼ばれる、クラウド活用のベストプラクティスをまとめたホワイトペーパーが無償で提供されています。本節では、AWS Well-Architectedフレームワークの概要を説明します。

1 AWS Well-Architectedフレームワークの概要

AWSが最初にクラウドサービスを開始したのは2006年に遡ります。当初はコンピューティングサービスのAmazon Elastic Compute Cloud(EC2)とストレージサービスのAmazon Simple Storage Service(S3)からサービスを開始しましたが、現在ではユーザーのニーズに応じた200以上の多様なサービスを提供しており、その種類はさらに増え続けています。

AWS Well-Architectedフレームワークは、AWSがサービス開始してから十数年、数多くのユーザーにさまざまなクラウドサービスを提供してきた経験をベストプラクティス集として整理したものです。

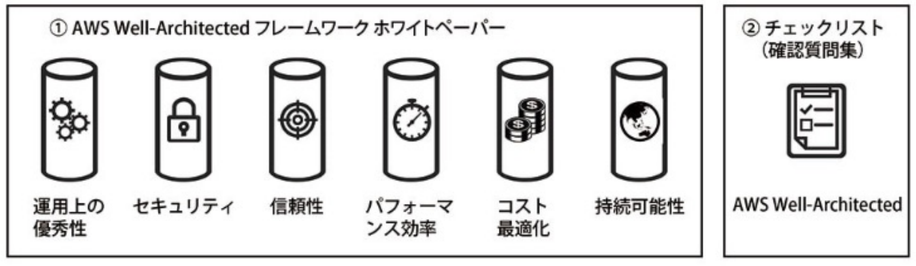
これにより、AWSのユーザーが設計・構築・運用の各工程で、AWSにおけるクラウド活用のノウハウを効率よく体現できるようになりました。

●AWS Well-Architectedフレームワークの構成

このフレームワークは、「運用上の優秀性」「セキュリティ」「信頼性」「パフォーマンス効率」「コスト最適化」「持続可能性」の6つを柱として、各項目のクラウド活用時の設計原則および、AWS上で構築するシステムがベストプラクティスに沿っているかを確認するためのチェックリスト(質問と回答)で構成されています。

この6つの柱に沿ってAWSを活用することが、安全かつ高パフォーマンスで、耐障害性を備えた、効率的なインフラストラクチャの実現への近道となります。

【AWS Well-Architectedフレームワークの構成】



●AWS Well-Architectedフレームワークの6つの柱の概要

6つの柱それぞれの概要は、以下のとおりです。

● 運用上の優秀性

運用上の優秀性の柱では、AWS上で構築したシステムがビジネスにおいて価値を提供し続けるために、運用面でどのような点に考慮すべきかを記載しています。たとえば、システムのモニタリングや変更管理、継続的な運用プロセス・手順の改善、通常・障害時の運用業務などのトピックが取り上げられています。

● セキュリティ

セキュリティの柱では、AWS上で構築したシステムとデータをどのように保護するかを記載しています。たとえば、データの機密性と整合性の担保、権限管理、セキュリティイベントの監視や制御のトピックが取り上げられています。

● 信頼性

信頼性の柱では、AWSにおける障害の防止、障害時の復旧について記載しています。たとえば、障害時における動的なコンピューティングリソースの獲得など、可用性の高いアーキテクチャに関するトピックが取り上げられています。

● パフォーマンス効率

パフォーマンス効率の柱では、AWSにおけるコンピューティングリソースの効率的な使用について記載しています。たとえば、性能要件や需要の変化に応じた適切なリソースタイプの選定や、パフォーマンスのモニタリングなどのトピックが取り上げられています。

● コスト最適化

コスト最適化の柱では、AWS上での不要なコストの回避や最適化について記載しています。たとえば、適切なコストの把握や、最適なリソースタイプの選定、需要に応じたシステムのスケーリングなどのトピックが取り上げられています。

● 持続可能性

持続可能性の柱では、エネルギー消費量を削減し地球環境への影響を最小限にするために必要な設計・実装について記載しています。たとえば、サーバーリソースの稼働・消費時間の最適化や、定期的なデータ削除による無駄なリソースの最適化などのトピックが取り上げられています。

試験対策

AWS Well-Architectedフレームワークは、AWSを利用するうえで重要なベストプラクティスです。実際の試験では、AWSがこれらのベストプラクティスに基づいて最適と考えるアーキテクチャを問う設問が、出題される傾向にあります。ベストプラクティスであるこれらフレームワークの概要をまずは押さえたうえで、試験範囲の個々のトピックを重点的に勉強するのが効率的です。また、AWS上でシステムの構築・運用を行う際にも参照し、ベストプラクティスに沿っているか定期的にチェックしましょう。

参考

AWS Well-Architectedフレームワーク（日本語版）
https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/framework/wellarchitected-framework.pdf

AWS Well-Architectedフレームワークに加えて、個々の技術トピックに対するホワイトペーパーも公開されています。

AWS ホワイトペーパーとガイド
https://aws.amazon.com/jp/whitepapers/

さまざまなテクニカル ホワイトペーパーや技術ガイド、参考資料、リファレンス アークテクチャ図などが無償で公開されており、学習の補足として活用できます。

Q 演習問題

1 **ある会社では、リリースしたアプリケーションが継続的に正しく動作し、障害時においても素早く復旧可能な運用を実現するクラウドアーキテクチャを設計したいと考えています。次のAWS Well-Architectedフレームワークのうち、どのベストプラクティスを指していますか？ 正しい項目を1つ選びなさい。**

A　パフォーマンス

B　コスト最適化

C　運用上の優秀性

D　セキュリティ

A 解答

1 C

AWS Well-Architectedフレームワークでは、リリース後の運用時におけるオペレーションのベストプラクティスとして「運用上の優秀性」の柱を定義しています。このなかで、継続性を重視した運用機能や、障害時の適切な運用方法などのトピックが取り上げられています。したがって、**C**が正解です。

1-2 ネットワークサービス

AWSは世界中にデータセンターを配備しており、AWSのインフラストラクチャを理解するうえで、ネットワークサービスの理解は不可欠です。本節では、AWSにおけるインフラストラクチャの概要と、ネットワークの基本となるVirtual Private Cloud（VPC）やVPCを構成するサービス、AWSと外部システムや外部サービスを接続する方法について説明します。

1 AWSインフラストラクチャの概要

●リージョンとアベイラビリティーゾーン

AWSは、世界中に**リージョン**と**アベイラビリティーゾーン**（**AZ**：Availability Zone）を保有しています。リージョンは、地理的に離れた領域のことを指し、世界中に30カ所あります（2022年12月時点）。

リージョンは、「東京」「シンガポール」「バージニア北部」などの地名や地域名で呼ぶことが多いですが、「ap-northeast-1」などのようにコード表記して使うこともあります。

AZは、リージョン内にある複数のデータセンターの集合体を指し、96カ所あります（2022年12月時点）。各AZ間は、冗長化された高速なネットワークで接続されており、それぞれ地理・電源・ネットワークが独立した施設で構成されているため、地震や火災などの災害発生時に1つのAZが正常に動作しなくなったとしても、ほかのAZでシステムを継続できます。AZは、リージョンに「a」や「b」など記号を付けて表記するため、たとえば「東京リージョンのAZ-a」や「ap-northeast-1a」といった形で表現します。

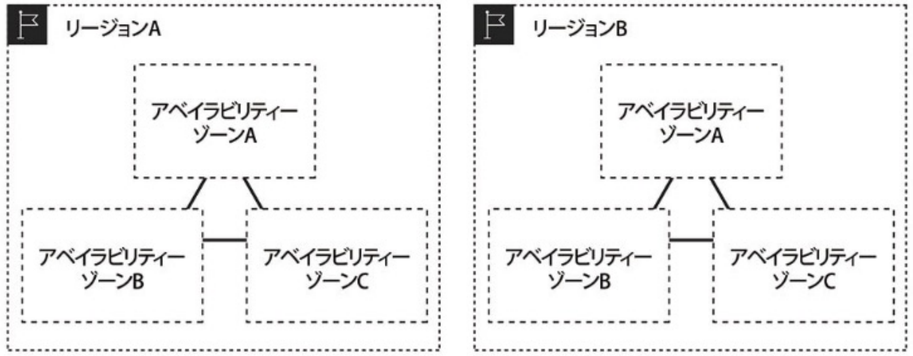
システムを設計するうえで、高可用性を実現するためには、複数のAZを使う**マルチAZ構成**が推奨されており、**災害復旧**（**DR**）[※1]サイトを構築する場合は、複

※1 【災害復旧（DR：Disaster Recovery）】：地震などの広範囲にわたる災害が発生した場合でも、遠隔地でシステムを復旧・再開すること。または、その災害に備えた対策や体制。

数のリージョンを使う**マルチリージョン**でシステムを構成することがあります。

AWSのアーキテクチャを構成するにあたり、複数のAZを利用することや、「**AWSマネージドサービス**」と呼ばれるAWSがインフラストラクチャを管理しているサービスを利用することで、パブリッククラウドの恩恵を最大限に享受することができます。

【リージョンとアベイラビリティーゾーン】



●エッジロケーション

CDN（Content Delivery Network）[※2]などのグローバルインフラストラクチャサービスを提供するための拠点のことで、AZを上回る数の拠点が全世界に設置されています。ユーザーは地理的に近いエッジロケーションから配布されるコンテンツを利用できるため、ネットワークの物理的な距離が短くなることで低レイテンシーの通信が可能になり、効率よくコンテンツを配布するシステムを構築できます。

試験対策

本項のロケーションに関する用語は、各AWSサービスを理解するうえで非常に重要です。特に第3章の「弾力性に優れたアーキテクチャの設計」や第4章の「高性能アーキテクチャの設計」では、内容を理解するための前提になります。

※2 【CDN】Content Delivery Network：世界中に配置されているCDNネットワークから効率的にコンテンツを配信する仕組み。閲覧者に最も近い拠点から配信するため、非常に効率がよい。

AWSでは、AWSが提供するインフラストラクチャ以外に、ユーザーが用意したインフラストラクチャにAWSを拡張する「AWS Outposts」や、電気通信事業ネットワーク内に5G（第5世代）通信デバイス向けのAWSサービスを構築できる「Wavelength Zone」などのサービスも提供されています。

AWSの各サービスは、利用できる範囲によって以下のように分類できます。
・グローバルサービス
リージョンに依存しない共通のサービスを指します。AWS Identity and Access Management（IAM） やAmazon CloudFront、Amazon Route 53などが該当します。
・リージョンサービス
特定のリージョン内でのみ利用するサービスを指します。 Amazon Virtual Private Cloud（VPC） やAmazon DynamoDB、AWS Lambdaが該当します。
・アベイラビリティーゾーン サービス
特定のアベイラビリティーゾーン（AZ）内で利用するサービスを指します。VPCのサブネットやAmazon Elastic Compute Cloud（EC2）、Amazon Relational Database Service（RDS）などが該当します。

たとえば、AWS IAMやRoute 53といったグローバルサービスは、AWSがすでにグローバルレベルでシステムを運用しているため、災害復旧（DR：Disaster Recovery）サイトを構築する際も、サービスを継続するための対策を考慮する必要がありません。
一方で、Amazon EC2などのアベイラビリティーゾーン サービスについては、特定のAZ内でしかサービスが稼働しないため、DRサイトを構築する際にはマルチAZまたはマルチリージョンの構成を検討する必要があります。
なお、Amazon S3は、バケット名がグローバルで一意でなければならないためグローバルサービスに分類されますが、データは特定のリージョンに保管されます。

2 Amazon Virtual Private Cloud（VPC）

世界中のユーザーが利用するAWSにおいて、特定のユーザーだけが利用できるネットワーク環境を構築することは、セキュリティの観点から非常に重要です。

AWSには、このネットワーク環境を実現するサービスとしてAmazon

Virtual Private Cloud（VPC）があります。Amazon VPCは、AWS上の特定のリージョンに独立したプライベートネットワーク空間を作成できるサービスです。

通常、AWS上のハードウェアやネットワークインフラストラクチャはユーザー全体で共有されていますが、「VPC」と呼ばれる仮想ネットワークを作成することで、論理的に分割された、特定のユーザーだけが利用できるプライベートネットワークを構築することができます。これにより、ネットワークのセキュリティを確保することができます。

VPCは、ユーザーが自由に作成することができますが、作成にあたってはいくつか注意点があります。

- IPv4のVPCは、プライベートIPアドレスの範囲内で、16ビット以上28ビット以内のCIDR※3を指定する必要がある
- VPC間を接続したりオンプレミス環境と接続する場合（後述）は、IPアドレスの範囲が重複しないように注意する必要がある
- IPv6のVPCは、56ビットのCIDRに固定されている

以下に、VPCに関するサービスと機能を説明します。

●サブネット

サブネットとは、VPC内に構成するネットワークセグメントです。ユーザーはサブネットを作成する際に、どのアベイラビリティーゾーン（AZ：Availability Zone）で構成するかを指定します。サブネットは、1つのVPCに対して複数作成することができますが、VPCのIPアドレス範囲内でCIDRを設定する必要があります。

サブネットを構成する際は、大きく分けてセキュリティの観点と可用性の観点を考えることが重要です。

まずセキュリティの観点では、パブリックサブネットとプライベートサブネットのどちらのサブネットとして機能させるかを検討する必要があります。

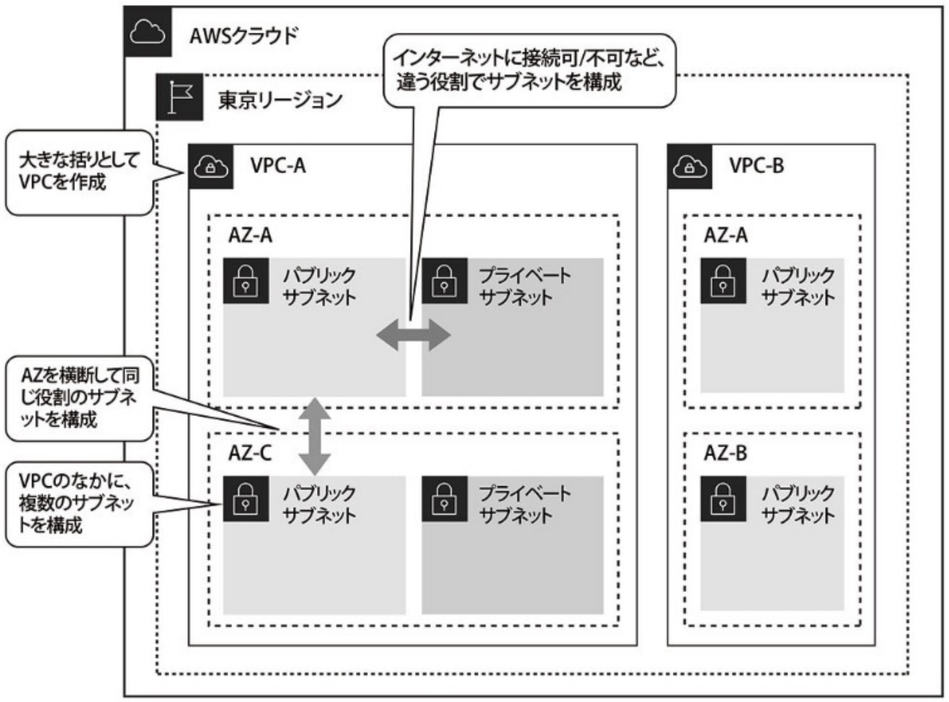
※3　【CIDR】Classless Inter-Domain Routing：IPアドレスのクラス分類を考慮せず、IPアドレスを割り当てる仕組み。IPアドレス空間を効率的に利用できる。

パブリックサブネットは、インターネットとの通信が可能なサブネットで、インターネットと通信する必要があるサーバーを配置します。また、**プライベートサブネット**には、インターネットと接続する必要がないサーバーを配置します。サブネットをインターネット接続の可／不可で分けることにより、ネットワークのセキュリティを向上させることができます。なお、パブリックサブネットとプライベートサブネットの用語の定義は、後述する「ルートテーブル」で説明します。

次に可用性の観点では、同じ用途のサブネットを複数のAZに作成することが重要です。サブネットは、システム単位やセキュリティの観点から複数作成することになりますが、同じ用途でもそれぞれのAZごとに作成することで、より可用性の高いネットワークを構築できます。

Amazon RDSなどのマルチAZ機能（95ページを参照）を利用するためには、異なるAZで作成した複数のサブネットを用意することが前提となります。

【VPCとサブネット】



試験対策

VPCはプライベートネットワーク空間でIPアドレスを自由に決めることができます。サブネットは、VPC内に構成するネットワークセグメントで、サブネットの役割やAZごとに作成することが重要です。

●インターネットゲートウェイ（IGW：Internet Gateway）

VPC内のリソースからインターネットにアクセスするためのゲートウェイで、IPv4およびIPv6のトラフィックをサポートしています。インターネットゲートウェイをVPCにアタッチすることで、VPC内のリソースからインターネットへアクセスできます。

●ルートテーブル

サブネット内のAWSリソースに対する静的ルーティングを定義するもので、設定はサブネット単位で行います。ルートテーブル内のデフォルトゲートウェイ（0.0.0.0/0）へのルーティングに、インターネットゲートウェイを指定したサブネットがパブリックサブネット、インターネットゲートウェイを指定していないサブネットがプライベートサブネットと定義されています。

試験対策

インターネットゲートウェイへルーティングされるサブネットをパブリックサブネット、インターネットゲートウェイへルーティングされないサブネットはプライベートサブネットと定義されています。

参考

VPC内の通信では、デフォルトでルートテーブルに「local」の経路が定義されており、同じVPC内にあるサブネット間で相互通信が可能になっています。これには異なるAZに作成されたサブネット間も含まれます。以下は、パブリックサブネットにおけるルートテーブルの定義例です。

【VPC（192.168.0.0/16）におけるルートテーブルの例】

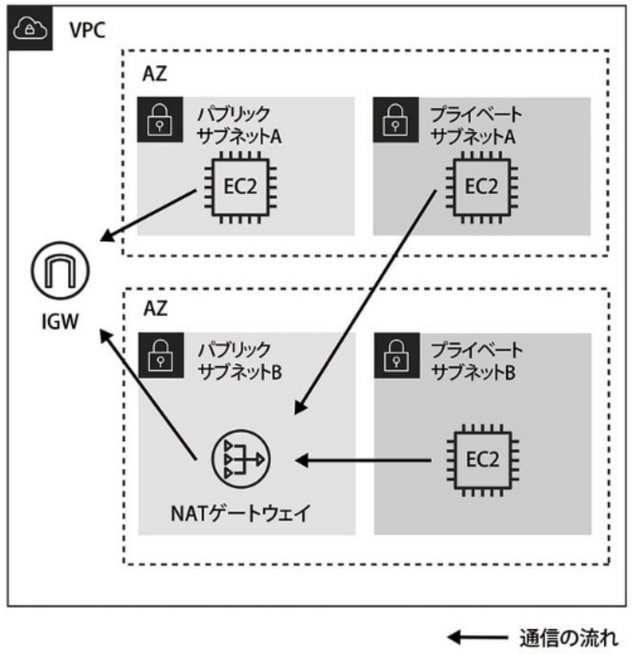
送信先	ターゲット
192.168.0.0/16	local
0.0.0.0/0	igw

●NATゲートウェイ

NATゲートウェイは、プライベートサブネットからインターネットへ接続するためのネットワークアドレス変換(NAT：Network Address Translation)機能を提供します。AWSではマネージドサービスとして提供されており、AZ内で冗長化されているため、NATインスタンス(後述)を利用する方法よりも高い可用性が期待できます。

パブリックサブネットとプライベートサブネットに配置されたEC2インスタンスから、インターネットへ接続する際の通信の流れを次の図に示します。

【パブリックサブネットとプライベートサブネット】



0.0.0.0/0 宛ての
ルートテーブル

サブネット	宛先
パブリックA	igw
パブリックB	igw
プライベートA	nat
プライベートB	nat

パブリックサブネットAのEC2インスタンスは、IGWからインターネットへ直接接続していますが、プライベートサブネットAとBのEC2インスタンスはインターネットに直接接続できないため、パブリックサブネットBにあるNATゲートウェイを経由してインターネットに接続しています。

主なユースケースとして、プライベートサブネットに配置されたAmazon EC2内のOSにパッチを適用する場合や、ソフトウェアライセンス認証を行う場合に利用されています。

NATゲートウェイを利用しない場合、「**NATインスタンス**」と呼ばれるNAT機能を持つEC2インスタンスを利用してインターネットにアクセスします。これには、「送信元／送信先チェック」と呼ばれる機能を無効化し、自身へのトラフィックを破棄する設定が必要になります。

このNATインスタンスはEC2インスタンスをNATサーバーとして利用するため、単一障害点とならないように冗長化の仕組みを検討する必要があります。

> **試験対策**
> プライベートサブネット内のEC2からインターネットに接続するには、NATゲートウェイまたはNATインスタンスが必要です。

3　ネットワークアクセス制御

●セキュリティグループ

EC2インスタンスなどに適用するファイアウォール機能です。EC2インスタンスから出る通信を制御する**アウトバウンド**、EC2インスタンスに入る通信を制御する**インバウンド**の2つを設定します。

セキュリティグループの主な特徴は、以下のとおりです。なお、セキュリティグループの詳細については、「2-3　ネットワークセキュリティ」を参照してください。

- デフォルトの設定値では、アウトバウンド通信はすべて許可、インバウンド通信はすべて拒否
- インバウンド／アウトバウンドの種別、プロトコル(TCPやUDP)、ポート範囲(80番など)、アクセス元またはアクセス先IPなど許可するルールのみを定義する
- EC2インスタンスに複数のセキュリティグループ(複数のルールの集合体)を適用できる
- セキュリティグループの設定追加・変更は即座に反映される
- **ステートフル**な制御が可能(ルールで許可された通信の戻りの通信も自動的に許可される)

●ネットワークアクセスコントロールリスト(ACL)

サブネット単位で設定するファイアウォール機能です。セキュリティグループと同様に、アウトバウンド通信とインバウンド通信の2つを設定します。たとえば、パブリックサブネットからデータベースサーバーが配置されたサブネットへの通信を明示的に拒否する、あるサブネットから外部へのSMTP通信を拒否するなど、主に異なるサブネット間の通信を制御する場合に利用します。

ネットワークACLの主な特徴は、次のとおりです。なお、ネットワークACLの詳細については、「2-3　ネットワークセキュリティ」を参照してください。

- VPC内に構成したサブネットごとに1つのネットワークACLを設定可能
- VPC作成時に、デフォルトのネットワークACLが1つ準備されており、初期設定ではすべてのトラフィックを許可
- 新規にネットワークACLを作成することもでき(「カスタムネットワークACL」と呼ぶ)、その場合の初期設定はすべてのトラフィックを拒否
- インバウンドとアウトバウンドそれぞれに対して、許可または拒否を明示した通信制御が可能
- **ステートレス**(セキュリティグループとは異なり、インバウンドとアウトバウンドに対する通信制御が必要)
- 各ルールに番号を割り当て、番号順に許可または拒否のルールを適用する

【セキュリティグループとネットワークACLの主な違い】

項　目	セキュリティグループ	ネットワークACL(デフォルト)
適用範囲	インスタンス単位	サブネット単位
デフォルト動作	インバウンド:すべて拒否 アウトバウンド:すべて許可 ENI※4単位で設定	インバウンド:すべて許可 アウトバウンド:すべて許可
ルールの評価	すべてのルールが適用される	ルールの順番で適用される
ステータス	ステートフル	ステートレス

試験対策

セキュリティグループはステートフルな制御が可能で、許可した通信のみ通すことができます。ネットワークACLはステートレスであるため、インバウンド通信とアウトバウンド通信の双方で通信制御を行う必要があります。

※4　【ENI】Elastic Network Interface：VPC内のネットワークインターフェイスを提供するサービス。利用可能なAWSサービスにアタッチすることでENIに紐付いたIPアドレスを利用できる。

4　VPCと外部との接続

●AWS Direct Connect

オンプレミス環境とAWSの間を専用線で接続するサービスです。専用線でプライベート接続することに加え、高速なネットワーク帯域で安定した通信を行うことができます。

また、Direct Connectの追加機能として、**Direct Connectゲートウェイ**があります。通常、Direct Connectはオンプレミス環境とVPCを「1対1」でしか接続できませんが、Direct Connectゲートウェイを利用することで、オンプレミス環境と全リージョン内の複数のVPCを「1対多」で接続できます。

参考

AWS Direct Connectは、「DX」という略称で呼ばれることが多くあります。近年では、デジタル トランスフォーメーションもDXと呼ばれているため、混同しないように注意しましょう。

●AWS Virtual Private Network(VPN)

AWS VPNは、オンプレミスネットワークやクライアントデバイスとAWSネットワーク間をVPN接続するサービスです。AWS VPNは、AWS Site-to-Site VPNとAWS Client VPNで構成されています。

● AWS Site-to-Site VPN

Direct Connect以外にも、オンプレミス環境とAWSの間を接続する方法として、IPSec※5を使用してセキュアに通信できる**AWS Site-to-Site VPN**があります。インターネット回線を利用しているため、Direct Connectに比べるとスループットや品質は低下しますが、比較的低コストかつ短期間で導入できる点がメリットです。

● AWS Client VPN

AWS Client VPNは、クライアントデバイスからAmazon EC2などのVPCの

※5　【IPSec】Security Architecture for Internet Protocol：IPパケットを暗号化し、盗聴や改ざんを防ぐ技術のこと。オンプレミス環境のVPNとしても多く利用されている。

リソースへのVPN接続を提供するサービスです。フルマネージド型のサービスであるため、従来のようにVPNサーバーを構築することなく、ユーザー向けのVPN環境を構築できます。

● 仮想プライベートゲートウェイ（VGW：Virtual Private Gateway）

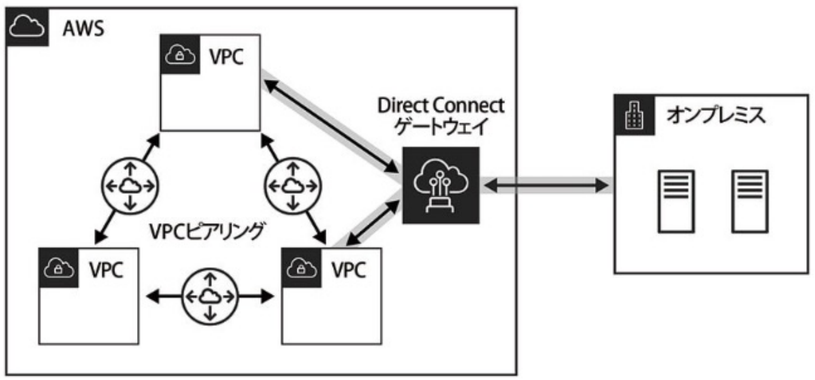
オンプレミス環境とAWSを接続する際に利用するゲートウェイです。Site-to-Site VPNやDirect Connect接続の際にあらかじめ用意しておき、VPCへアタッチして利用します。

> **試験対策**
> オンプレミス環境とAWS環境をプライベート接続するには、Direct ConnectまたはSite-to-Site VPNを利用します。Direct Connectは、Site-to-Site VPNより高いスループットや安定性を維持できますが、導入までに時間がかかり、コストが高くなる傾向にあるという点が重要です。

●VPCピアリング

異なるVPC間をプライベート接続するサービスです。VPCは、AWS上に独立したプライベートネットワーク空間を作成するため、本来はVPC間での直接的な通信は行えません。しかし、VPC間をVPCピアリング接続することで、インターネットを経由せずにAWSのプライベートネットワーク内で直接通信できるようになります。VPCピアリングは異なるAWSアカウント間でも接続できますが、ネットワークアドレスが一致または重複する場合は接続できません。

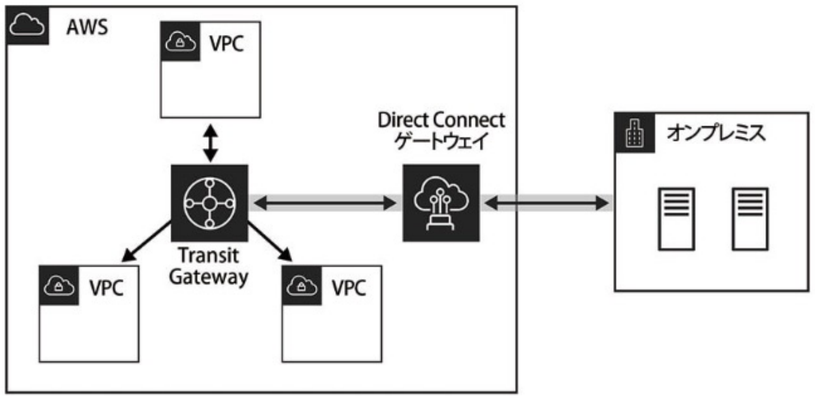
【VPCピアリングとDirect Connectゲートウェイの接続例】



●AWS Transit Gateway

VPC内にハブの機能を備えたゲートウェイを配置するサービスです。ある特定のVPCが外部のネットワークと接続する必要がある場合、前述した Direct ConnectやVPN、VPCピアリングなどそれぞれのサービスで、設定や運用を行う必要があります。当然、接続先が増えるほど管理が煩雑になり、ネットワーク構成が複雑化しますが、Transit Gatewayを利用することでネットワークを簡素化し、一元管理することができます。

【Transit Gatewayの接続例】



●VPCエンドポイント

VPC内のリソースから各種のAWSサービスへのアクセスやAPIコールは、通常はインターネットを経由して行いますが、**VPCエンドポイント**を利用することで、AWSのプライベートネットワークからアクセスできるようになります。

VPCエンドポイントには、ネットワークレイヤーのゲートウェイ型とアプリケーションレイヤーのインターフェイス型の2種類があります。

ゲートウェイ型では、**ルートテーブルに指定されたターゲットを追加**することで、Amazon Simple Storage Service（S3）やAmazon DynamoDBなどのAWSサービスへプライベート接続することができます。

インターフェイス型は、「**AWS PrivateLink**」とも呼ばれ、Amazon CloudWatchやAmazon Simple Queue Service（SQS）などの**APIコール**に対して、インターネットを経由せずにプライベート接続することができます。

ゲートウェイ型は無料で利用できますが、インターフェイス型は料金が発生するので注意しましょう。

試験対策

VPCエンドポイントには、ゲートウェイ型とインターフェイス型があり、インターネットを経由せずにAWSサービスにアクセスできます。

5 Elastic Load Balancing（ELB）

Elastic Load Balancing(ELB)は、Amazon EC2や特定のIPアドレスへのトラフィックを分散するロードバランシングサービスで、EC2インスタンスを登録するだけで利用できます。ここで登録されたEC2インスタンスを「バックエンドインスタンス」と呼びます。

ELBには以下の4種類があり、それぞれ特性が異なります。

- Classic Load Balancer(CLB)：標準的なロードバランシングを提供する。CLBに対するTCPリクエストをバックエンドインスタンスに振り分け、指定されたポートへトラフィックを転送する。
- Application Load Balancer(ALB)：リクエストレベル(レイヤー7)で動作し、リクエストの内容に応じて、あらかじめ設定したターゲットにルーティングして処理を振り分ける。そのため、Webサーバーやアプリケーションサーバーなどで利用するHTTPやHTTPSトラフィックを、特定のサーバーに振り分けることができる。
- Network Load Balancer(NLB)：レイヤー4で動作し、低レイテンシーで高いスループットを実現する場合に利用する。送信元のアドレスを保持するため、レスポンスはクライアントへ直接返す。
- Gateway Load Balancer(GLB)：レイヤー3で動作し、ファイアウォールや侵入検知などの仮想アプライアンスの可用性をより高められる。

以下にELBにおける重要な特徴を説明します。

●高可用性

ELBは、トラフィックを複数のAZへ分散できるため、地理的に離れたデータセンターを利用することで可用性の向上が見込めます。また、トラフィックは正常なバックエンドインスタンスのみに振り分けられるため、インスタンスや

AZに障害が発生しても、正常なインスタンスとの通信を続けることができます。

●自動スケーリング

ELBは、トラフィックの負荷に応じて自動でスケーリングする機能を備えており、ELB自体にも冗長性が確保されています。ELBにはVPCサブネットのIPアドレスが自動的に割り振られます。IPアドレスは変動するため、通常、ELBへの接続は「エンドポイント」と呼ばれる、ELBに割り当てられたDNS名を使用します。

●セキュリティ機能

ELBは、SSL復号の機能を備えています。後述するAWS Certificate Manager(ACM)と連携し、SSL復号をELBが担うことで、バックエンドインスタンスの負荷軽減や証明書の一元管理が可能になります。また、ELB自体にもセキュリティグループが設定できるので、ELBのみでアクセス制御を行うこともできます。

●ヘルスチェックとモニタリング

ELBは、バックエンドインスタンスが正常に動作しているか、ヘルスチェックを行います。ヘルスチェックの結果、異常なEC2インスタンスには振り分けを停止し、正常なEC2インスタンスのみと通信を行います。このとき、「スティッキーセッション」という機能を使用することで、一度セッションを確立したバックエンドインスタンスにユーザーのリクエストを送信できるようになります。

また、バックエンドインスタンスをELBから登録解除する際に、サーバーが何らかの処理を行っていると、その処理を中断してしまうことが懸念されます。この場合、「Connection Draining」という機能を利用すると、サーバーの処理が完了するまで解除を遅延させることができます。ELBは、応答時間やリクエスト数の記録だけでなく、通信のログをAmazon S3に保管することができます。

●クロスゾーン負荷分散

クロスゾーン負荷分散を使用すると、ELBは複数のAZに登録されたすべてのインスタンスに対してリクエストを均等に分散します。一方、クロスゾーン負荷分散を無効にした場合、ELBはAZごとにリクエストを均等に分散するため、AZによってインスタンス数が異なるとすべてのインスタンスでリクエストを均等に分散できなくなります。

●外部ELBと内部ELB

ELBは、外部ELB(インターネット公開向け)のInternet-facingか、内部ELBとして利用するInternalのいずれかで動作します。Internet-facingの場合、ELB自体がパブリックサブネットに配置されているため、バックエンドインスタンスをパブリックサブネットに配置する必要がなく、プライベートサブネットに配置することで、セキュリティを高めることができます。Internalは、VPC内やオンプレミス環境からのみ利用することができます。

●AWS Certificate Manager(ACM)

AWS Certificate Managerは、複数のAWSサービスで利用するSSL証明書を一括で管理することができます。また、AWSで利用するSSL証明書の新規発行だけでなく、既存SSL証明書をインポートして管理することも可能です。このACMで管理されたSSL証明書は、主にELBやCloudFront、API GatewayでHTTPS通信を行う場合に利用されます。

試験対策

ELBは、複数のAZに設定することで高可用なネットワークを構築できます。ELBをパブリックサブネットに配置し、バックエンドインスタンスをプライベートサブネットに配置することで、サーバーのセキュリティを担保しつつSSL証明書の管理を簡素化できます。

6 Amazon CloudFront

Amazon CloudFrontは、エッジロケーションからコンテンツを配信するCDN(Content Delivery Network)サービスです。

CloudFrontを利用することで、閲覧者は地理的に最も近いキャッシュサーバーからコンテンツを受け取れるため、通信の伝送距離が短くなることで低レイテンシーでWebサービスを提供することができます。

CloudFront自体には、高可用性、高パフォーマンス、低レイテンシーなネットワークが用意されており、オリジンとしてELBやAmazon EC2、Amazon S3などを指定することができます。**オリジン**とは、CloudFrontへキャッシュするコンテンツの提供元を指します。

CloudFrontには、主に次のような機能があります。詳細については、「4-2 ネットワークサービスにおけるパフォーマンス」を参照してください。

●SSLによる通信の暗号化

CloudFrontでは、SSL証明書を利用したSSL暗号化通信が可能です。ユーザーが用意した独自のSSL証明書を導入することができます。また、CloudFrontからバックエンドのAWSサービスまでをSSL暗号化通信にすることも可能です。

●署名付きURL

一定時間だけアクセスを許可するためのURLを発行し、URLを通知した限定的なユーザーにのみ公開できる機能です。

●カスタムエラーページ

オリジンからエラーコードが返された場合、あらかじめ設定したエラーページを表示することができます。

●地域制限(地理的ブロッキング)

CloudFrontに接続するユーザーの地域情報に基づいて、アクセスを許可または拒否することができます。

●ストリーミング配信

Webサービスのコンテンツ配信に加えて、映像や音声のストリーミング配信にも対応しています。

試験対策

CloudFrontは、ELBやAmazon EC2、Amazon S3などオリジンのコンテンツをキャッシュして配信することで、Webサービスのパフォーマンスを向上することができます。また、全世界にコンテンツを効率よく配布できますが、署名付きURLや地域制限を行うことで、一定のアクセス制御ができる点が重要です。

7 Amazon Route 53

Amazon Route 53は、可用性の高いDNSを提供するマネージドサービスです。ホストゾーンの設定として、外部向けDNSの**パブリックホストゾーン**と、VPC内DNSの**プライベートホストゾーン**があります。パブリックホストゾーンは公開Webサーバーの名前解決などに利用し、プライベートホストゾーンは社内システムなど、インターネットを介さない通信における名前解決に利用します。

Route 53は、「**ALIASレコード**」と呼ばれるRoute 53独自のDNS機能を利用することができます。ALIASレコードは、ELBやCloudFrontのエンドポイントを、CNAMEレコード[※6]ではなくAレコード[※7]として指定することができます。そのため、Zone Apexレコード[※8]をAレコードとして扱うことができます。

また、単にDNSの機能だけでなく、通信を制御する**ルーティングポリシー**を使用することもでき、レコードを作成する際にRoute 53がクエリにどのように応答するかを設定できます。

次の表にRoute 53の主なルーティングポリシーを示します。

※6 【CNAMEレコード】ドメイン名から別のドメイン名(別名)を参照するDNSレコード。

※7 【Aレコード】ドメイン名からIPアドレスを参照するDNSレコード。

※8 【Zone Apexレコード】サブドメインを含まないドメイン名のこと。通常、Zone ApexレコードはNSレコードとして利用されるため、CNAMEレコードが使えないという制限がある。

【主なルーティングポリシー】

ルーティング ポリシー	説 明
レイテンシーベースルーティング(レイテンシールーティング)	レイテンシーが最も低いリソースにルーティングする
加重ルーティング	複数のリソースに加重度を設定し、指定した比率に応じて処理を分散するようにルーティングする
位置情報ルーティング	接続元のクライアントの位置から、地理的に近い場所にルーティングする
フェイルオーバールーティング	ルーティング先の対象になるリソースをヘルスチェックし、利用できるリソースにルーティングする
シンプルルーティング	設定されたレコードの情報に従ってルーティングする
地理的近接性ルーティング	ユーザーとリソースの地理的場所に基づいてルーティングする
複数値回答ルーティング	最大8個からランダムに選ばれた正常なレコードを使用して、Route 53がDNSクエリに応答する

試験対策

Route 53のルーティングポリシーは押さえておきましょう。特に、フェイルオーバー ルーティングを設定することで高可用性ネットワークを構築でき、レイテンシーベースルーティングを設定することで高パフォーマンス ネットワークを構築できる点が重要です。

参考

Amazon Route 53 は、特定のルーティングポリシーだけでなく、複数のルーティングポリシーを組み合わせて設定することもできます。

8 その他のネットワークサービス

●AWS Global Accelerator

AWSが管理するネットワーク網を利用することにより、アプリケーションを高パフォーマンスで提供できるサービスです。

ユーザーからのリクエストは、まずエッジロケーションを経由し、Global Acceleratorによってトラフィックが最適化されます。最適化されたトラフィックに従って、AWSが管理するネットワーク網からNLBやAmazon EC2などのエンドポイントに転送されます。

リクエストからエンドポイントまでの通信の大部分をAWSネットワーク網で行うことにより、インターネットを経由した場合よりも高いパフォーマンスが期待できます。

●VPCフローログ

VPC内のネットワークインターフェイス(ENI：Elastic Network Interface)で通信するトラフィック情報をキャプチャするサービスです。キャプチャした情報は、Amazon CloudWatch Logsへ転送されます。設定は、ENI単位、サブネット単位、VPC単位で指定することができます。

●Elastic IP(EIP)

固定のグローバルIPアドレスを提供するサービスです。通常、EC2インスタンスを停止すると、それまで使用していたIPアドレスは保持できなくなります。IPアドレスが変わるたびにアプリケーションやDNSを変更するのは、運用上の手間がかかるため、対応策としてEIPを利用します。

EIPは、起動しているEC2インスタンスにアタッチされている場合には料金が発生しませんが、EC2インスタンスにアタッチされていない場合や、アタッチされているEC2インスタンスが停止中の場合には時間単位で料金が発生します。

●AWS Resource Access Manager(RAM)

AWSアカウントやAWS Organizations(168ページを参照)組織内で、サブネットやAWS Transit Gatewayなどのリソースを共有するサービスです。

管理やコスト分離の観点から複数のAWSアカウントを利用している場合、たとえ用途が同じであっても、それぞれのAWSアカウントごとにリソースを作成する必要があります。

AWS RAMを利用することで、すべてのAWSアカウントで共通して利用するリソースを集約し、運用の負荷やコスト効率を向上させることができます。

● AWS DataSync

オンプレミス環境のストレージとAWSのストレージサービスとの間で、データを高速に転送できるサービスです。DataSyncは、専用のエージェントを導入することで利用でき、主にAWSへのデータ移行を効率的に行う用途で利用されます。

また、データ移行のほかにも、機械学習用の分析基盤をAWS上に構築している場合など、オンプレミス環境で稼働する業務システムとは別の目的でAWSを利用する際のAWSへのデータ転送を効率化するためにも利用されています。

転送先のストレージサービスとしては、主にAmazon S3、Amazon EFS、Amazon FSx for Windows File Serverなどがあります。

●AWS Directory Service

Directory Serviceは、マネージド型でMicrosoft Active Directory(AD)機能を提供するサービスで、これによりAWSリソースからMicrosoft ADの機能が利用できます。また、新規にディレクトリを作成するだけでなく、Microsoft ADの信頼関係を利用して既存のMicrosoft AD環境と統合することができます。詳細は、「2-2 アイデンティティ管理とアクセス管理」を参照してください。

●AWS Network Firewall

AWS Network Firewallは、ネットワークトラフィックに応じて自動で拡張するVPCネットワークの侵入検知やフィルタリングを行うステートフルなマネージドサービスです。プロトコルやIPアドレス、ポートなどを制御することで、主にVPC間やオンプレミス間、インターネットへのトラフィックを保護します。

Q 演習問題

1 **新たにAWS環境で構築したシステムを、オンプレミス環境で動作しているシステムと連携する必要があります。次のうち、高品質かつ低レイテンシーでプライベート接続が可能なサービスはどれですか。**

A　AWS Direct Connect

B　AWS Site-to-Site VPN

C　Elastic Load Balancing

D　NATゲートウェイ

2 **プライベートサブネットのAmazon EC2からインターネットへ接続するために利用するAWSサービスは、次のうちどれですか。**

A　VPCピアリング

B　Amazon Route 53

C　Amazon CloudFront

D　NATゲートウェイ

3 **Amazon EC2からAmazon DynamoDBにインターネットを経由しないで接続するためのAWSサービスは、次のうちどれですか。**

A　インターネットゲートウェイ

B　AWS Transit Gateway

C　VPCエンドポイント

D　AWS DataSync

A 解答

1 A

AのDirect Connectは、オンプレミス環境とAWSの間を専用線で接続するサービスで、専用線でプライベート接続することに加え、高速なネットワーク帯域で安定した通信を行うことができます。
BのSite-to-Site VPNは、オンプレミス環境とAWSの間をセキュアに通信することができますが、インターネット回線を利用しているため、スループットや品質は低下します。
CのElastic Load Balancingは、EC2や特定のIPアドレスへのトラフィックを分散するロードバランシングサービスで、AWSとオンプレミス間をプライベート接続するサービスではありません。
DのNATゲートウェイは、プライベートサブネットからインターネットへ接続するためのNAT機能を提供するサービスで、AWSとオンプレミス間をプライベート接続するサービスではありません。
したがって、**A**が正解です。

2 D

AのVPCピアリングは、異なるVPC間をプライベート接続するサービスです。BのRoute 53は、DNSを提供するマネージドサービスです。CのCloudFrontは、エッジロケーションからコンテンツを配信するCDNサービスです。DのNATゲートウェイは、プライベートサブネットからインターネットへ接続するためのNAT機能を提供するサービスです。したがって、**D**が正解です。

3 C

Aのインターネットゲートウェイは、VPC内のリソースからインターネットへアクセスするためのゲートウェイです。BのTransit Gatewayは、VPC内にハブの機能を持ったゲートウェイを配置するサービスです。CのVPCエンドポイントは、VPC内のリソースから各種AWSサービスへのアクセスやAPIコールを、AWSのプライベートネットワークからアクセス可能にするサービスです。DのDataSyncは、オンプレミス環境のストレージとAWSのストレージサービスとの間で、データ転送を高速に行えるようにするサービスです。したがって、**C**が正解です。

1-3 アクセス制御サービス

AWSを利用するには、まず最初にAWSのユーザー登録を行う必要があります。
本節では、AWS Identity and Access Management（IAM）によるユーザー管理をはじめとした、アクセス制御に関連するサービスの概要を紹介します。

1 IAMサービスの概要

AWS Identity and Access Management(IAM)サービスとは、AWSを利用するユーザーに対してAWSへのアクセスを安全に制御するための仕組みです。なお、IAMサービスについての詳細は、「2-2　アイデンティティ管理とアクセス管理」を参照してください。

AWSを最初に利用する際は、以下の流れで管理者となるAWSアカウントと、操作するユーザーをIAMサービスで登録します。

●AWSアカウントの登録

AWSを最初に利用する際には、管理者のメールアドレスやプロフィールなどの個人情報、料金支払いに必要なクレジットカード情報などを登録し、AWSを利用するためのアカウント(**AWSアカウント**)を取得する必要があります。

最初に登録したメールアドレスで、AWSの操作の入口となる**AWSマネジメントコンソール**にログインすることができます。

このユーザーを「**ルートユーザー**」と呼び、すべての管理権限が付与されます。

●IAMユーザーおよびIAMグループの登録

ルートユーザーではAWSのすべての操作を行うことができるため、誤操作などの危険を伴います。そこで、通常のAWSの操作は、操作用のユーザーやグループを作成して行います。

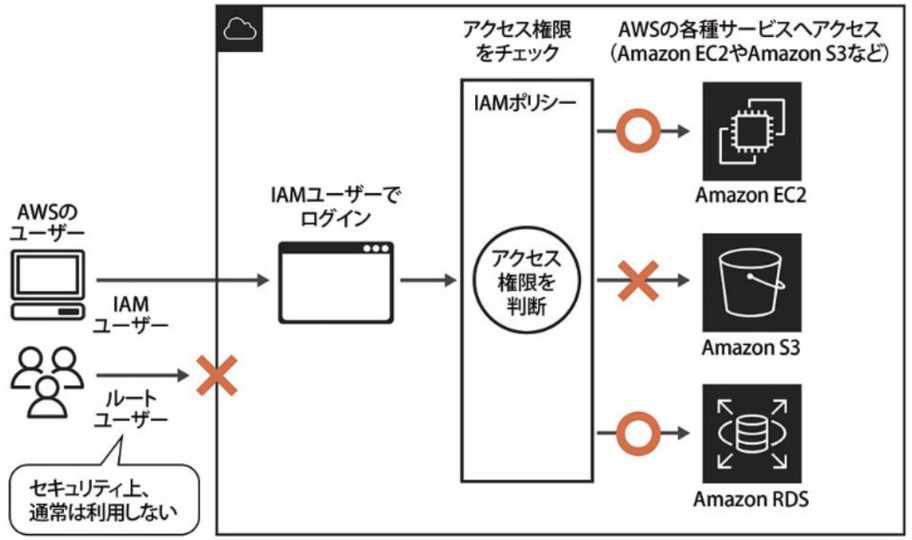
AWSではIAMサービスを使用して**IAMユーザー**および**IAMグループ**を作成することで、実際に利用するユーザーやグループを登録できます。

●IAMポリシーによる権限の付与

IAMユーザーおよびIAMグループは、作成直後には権限が付与されていません。

IAMユーザーまたはIAMグループに対して、**IAMポリシー**で適切な権限を付与することで、AWSの各種サービス、ストレージなどのリソースへのアクセス制御を行うことができます。たとえば、開発者のIAMグループに所属するIAMユーザーに対しては、AWSの仮想サーバーサービスであるAmazon Elastic Compute Cloud(EC2)への操作許可のみを与えるといったアクセス制御が可能となります。

【IAMによるアクセス制御のイメージ】



2 IAMサービスを通じたAWSの操作方法

AWSにアクセスして各種サービスやリソースを操作するには、主に以下の3種類の方法があります。

●WebブラウザでAWSマネジメントコンソールにログインする

登録したIAMユーザーのユーザー名とパスワードを用いて、WebブラウザからAWSが提供する**AWSマネジメントコンソール**の画面にアクセスします。

●AWS CLIでWindowsやLinuxからコマンド操作する

AWS Command Line Interface(AWS CLI)は、AWSサービスを管理するためのコマンド群です。

AWS CLIをインストールすることで、WindowsコマンドプロンプトやLinuxシェルから各種のコマンドを利用できます。たとえば、Linuxシェルからコマンドを実行して、AWSサービスを操作できます(次の例を参照)。

コマンド操作には、操作先のリージョンおよびIAMユーザーごとに作成できる**アクセスキーID**と**シークレットアクセスキー**を事前に作成しておく必要があります。この2つのキーペアは、AWS CLIやAWS SDK(後述)を利用する際の認証情報として使われます。

例 EC2インスタンスのIDを指定して起動するコマンド

```
$ aws ec2 start-instances --instance-ids <instance_id>
{
    "StartingInstances": [
        {
            "InstanceId": "i-xxxx",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

←インスタンスIDを指定して、AWSのEC2インスタンスを起動するコマンドを実行

←起動コマンドの実行結果を表示

←i-xxxxというインスタンスIDのEC2インスタンスが起動準備中であることを示す

←起動前のインスタンス状態を表示（この例では停止状態）

アクセスキーID、シークレットアクセスキーは、aws configureコマンドまたはLinuxやWindows環境変数で事前に設定しておくことが可能です。

●AWS SDKでプログラムからAPIを利用する

AWS SDKは、Javaなどの主要なプログラミング言語向けにAWSから提供されているAPI群です。JavaなどのプログラムからAWS SDKで提供されているAPIを指定することで、AWSの各種サービスを操作できます。

AWS SDKを利用する場合も、アクセスキーIDとシークレットアクセスキーによる認証情報が必要になります。

なおアクセスキーIDとシークレットアクセスキーを利用した認証は、キー流出のリスクが懸念されるため、現在はIAMロールによる認証が推奨されています。詳細は、「2-2　アイデンティティ管理とアクセス管理」で説明します。

Q 演習問題

1 **IAMユーザーを作成した直後に設定されているアクセス権限に関する説明として、正しい項目は次のうちどれですか。**

A　すべての権限が付与されている

B　Amazon EC2サービスを操作する権限のみ付与されている

C　権限は付与されていない

D　IAMサービスを操作する権限のみ付与されている

2 **AWS CLIやAWS SDKを利用するために必要な認証情報は、次のうちどれですか。**

A　IAMユーザーのログイン名とパスワード

B　IAMユーザーに付与されたIAMポリシー

C　IAMユーザーのアクセスキーIDとシークレットアクセスキー

D　ワンタイムパスワード

A 解答

1 C

IAMユーザーは、作成直後には権限が付与されていないため、IAMポリシーの割り当てが必要です。

2 C

IAMユーザーごとに作成可能なアクセスキーIDとシークレットアクセスキーを使用することで、AWS CLIとAWS SDKが利用できます。

1-4 コンピューティングサービス

AWSでは、サーバーなどのITリソースを少ない手間で作成するためのサービスがいくつも提供されています。
本節では、サーバーを提供する主要なコンピューティングサービスであるAmazon EC2と、その他の主要なサービスを説明します。また、EC2と合わせて利用されることが多い移行サービスやコンテナサービスについても紹介します。

1 Amazon Elastic Compute Cloud（EC2）

Amazon EC2は、AWSクラウド上で仮想サーバーを提供する、AWSの代表的なコンピューティングサービスです。このサービスを利用して作成した仮想サーバーのことを**EC2インスタンス**といいます。EC2ではハイパーバイザー型の仮想化が利用されていますが、仮想化ソフトウェアの機能によって、ほかの利用者の仮想サーバーにアクセスできないように設計されています。

ユーザーは、AWSマネジメントコンソール上で行う簡単な操作によって、必要な台数、必要なスペックのEC2インスタンスを容易に構築することができます。

スペックは、システムの利用状況に応じて柔軟に変更できます。また、EC2インスタンスの利用料金は原則として従量課金制のため、利用した分だけのコスト負担で済みます。

次に、実際のEC2インスタンスの構築時の主な設定項目を説明します。

●Amazon Machine Image（AMI）の選択

EC2インスタンスは、「**AMI**」と呼ばれるイメージファイルから起動します。

イメージファイルにはインスタンスの起動に必要な**OSイメージ**と**ソフトウェア設定**が含まれているため、インスタンス起動後すぐにサーバーとして利用することができます。一般的にAWSで標準提供されているAMIを利用することが多いですが、ベンダーなどから提供されているAMIをAWS Marketplace（46ページを参照）から利用することもできます。

●インスタンスタイプの選択

EC2インスタンスで利用するサーバースペックを**インスタンスタイプ**と呼びます。インスタンスタイプは、vCPU(仮想CPU)とメモリ容量の組み合わせがパッケージ化されており、多数の組み合わせから自由に選択できます。また、vCPUとメモリ容量のバランスによって、**インスタンスファミリー**と呼ばれるグループに分かれています。

詳細は、「5-2　コスト効果が高いリソースの選定」を参照してください。

●インスタンスの詳細設定

インスタンスの詳細設定では、EC2インスタンスを構築するネットワーク情報やIAMロールを設定します。

詳細は、「2-2　アイデンティティ管理とアクセス管理」を参照してください。

●ストレージの設定

EC2インスタンスに接続するストレージを設定します。ストレージは、Amazon EBSやインスタンスストアを選択することができます。

詳細は、「1-5　ストレージサービス」を参照してください。

●タグの追加

EC2インスタンスや設定したストレージに対して、タグ(AWSリソースに付けるラベル)を追加することで、運用の効率化や課金の振り分けを行うことができます。

詳細は、「5-4　コストの管理」を参照してください。

●セキュリティグループの設定

セキュリティグループは、EC2インスタンスへのアクセスを制限するファイアウォール機能です。

ネットワークACLのサブネット単位での通信制御に加えて、セキュリティグループを使用することで、EC2インスタンス単位でアクセス制御を行うことができます。AWSでは、セキュリティグループを使ってアクセス制御を行うのが一般的です。

詳細は、「2-3　ネットワークセキュリティ」を参照してください。

●キーペアの設定

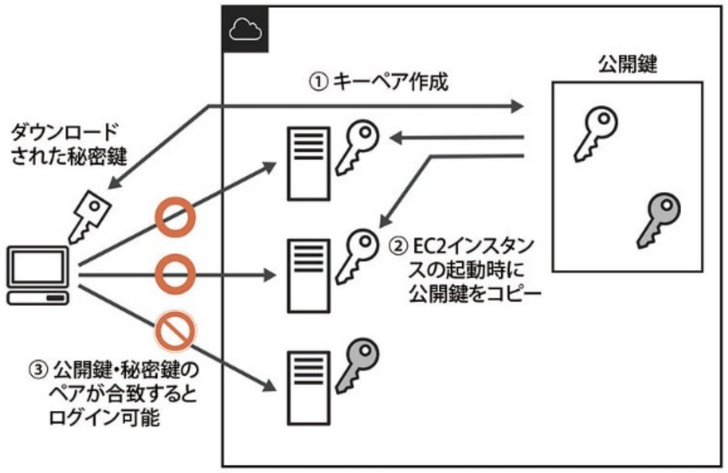
EC2インスタンスへのログインは公開鍵認証方式をとっており、鍵認証を用いてログインします。

一般的には**キーペア**というサービスで公開鍵と秘密鍵を発行します。

公開鍵はAWSで管理され、EC2インスタンス起動時にコピーします。秘密鍵はダウンロードしてユーザーが保管し、EC2インスタンスへのアクセス時に使用します。秘密鍵を紛失した場合は、新しいキーペアを発行して別のEC2インスタンスを起動します。Amazon EBS-Backedインスタンス(後述)を利用している場合は、リカバリを実施する必要があります。

なお、秘密鍵を紛失すると、構築したEC2インスタンスにログインすることができなくなるため、取り扱いには注意が必要です。

【キーペアを利用した通信】



> **参考**
> 2022年4月5日にEC2インスタンスの新しい起動ウィザードが発表されました。これまで複数ページに分割されていたAMIやインスタンスタイプなどの設定項目が、1つのページにまとめられ、AMIの選択がより容易になりました。主要な設定項目は大きく変わっていませんが、詳細な設定や機能が増えているので、今後新しい起動ウィザードでEC2インスタンスを作成する場合は、注意が必要です。

●Amazon Machine Image(AMI)

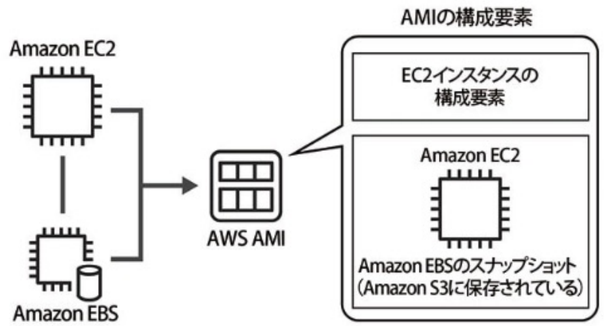
AMIはEC2インスタンスを作成する際に使用する仮想マシンイメージで、

Amazon EBSのスナップショットとEC2インスタンスの構成情報から成っています。OSには、Red Hat Enterprise LinuxやUbuntuなどの各種Linuxディストリビューションと Microsoft Windows Server、AWSが提供しているAmazon Linuxが利用できます。

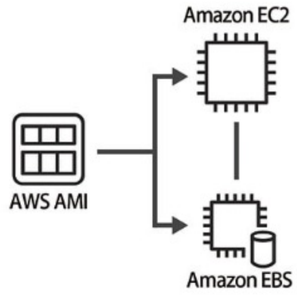
AWS Marketplaceには、ミドルウェアがインストール済みのサードパーティー製AMIが用意されています。また、ユーザー自身が作成したAMIを利用することもできます。

AMIをAWS Auto Scalingなどのサービスで指定することで、同じ設定のEC2インスタンスを、必要なときに必要な分だけ自動で構築することができます。

【ユーザー自身がAMIを作成する場合のイメージ】



【ユーザーが作成したAMIや、AWS Marketplaceで入手したAMIからEC2環境を構築するイメージ】



AMIは、Amazon EBS-BackedもしくはInstance Store-Backedのどちらかに分類されます。EBS-BackedのAMIでEC2インスタンスを起動した場合、EBSがOSのルート領域として利用されます。EBSはEC2インスタンスを停止してもデータは残り続けます(不揮発性)。

Instance Store-BackedのAMIでEC2インスタンスを起動した場合、インスタ

ンスストアがOSのルート領域に利用されます。インスタンスストアはEC2インスタンスを停止するとデータが削除されます(揮発性)。

すでに動作しているEC2インスタンスと同じ環境を別のリージョンにも複製したい場合は、以下の作業を行います。

① EC2インスタンスのAMIをコピーし、複製先として別のリージョンを指定する
② 新しいリージョンに複製されたAMIからEC2インスタンスを起動する

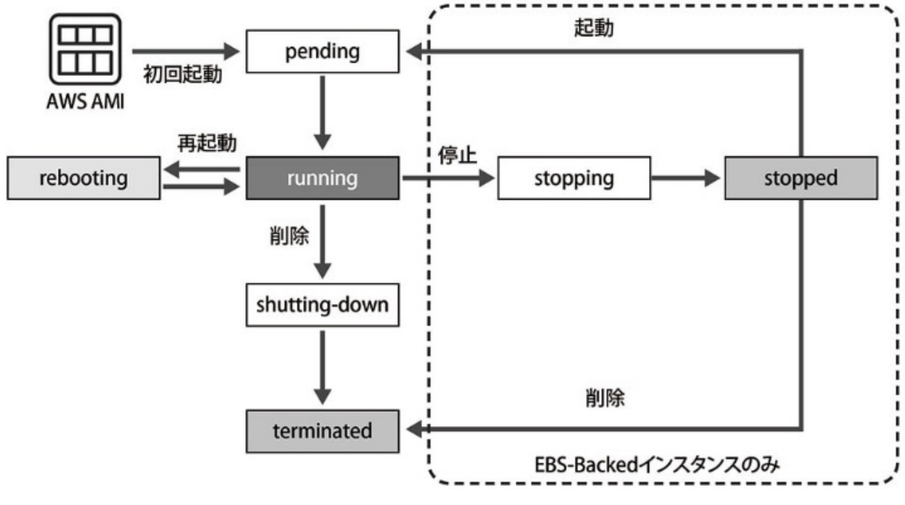
Amazon EBSとインスタンスストアの詳細は、「3-4　ストレージにおける高可用性の実現」を参照してください。

試験対策
現在のEC2インスタンスと同じ環境を別リージョンに複製する方法を理解しておきましょう。

●EC2インスタンスのライフサイクル

EC2インスタンスの状態は以下のように遷移します。

【EC2インスタンスのライフサイクル】



【EC2インスタンスの状態】

状　態	内　容	課金対象
pending	running状態の前段階。初回起動時またはstoppedから起動する場合の状態	対象外
running	EC2インスタンスが起動した状態	対象
stopping※	インスタンス停止準備中の状態	対象外
stopped※	シャットダウンされているため利用不可の状態。インスタンスの再開はいつでも可能	対象外
shutting-down	インスタンス削除準備中の状態	対象外
terminated	インスタンスが完全に削除されているため再開不可	対象外

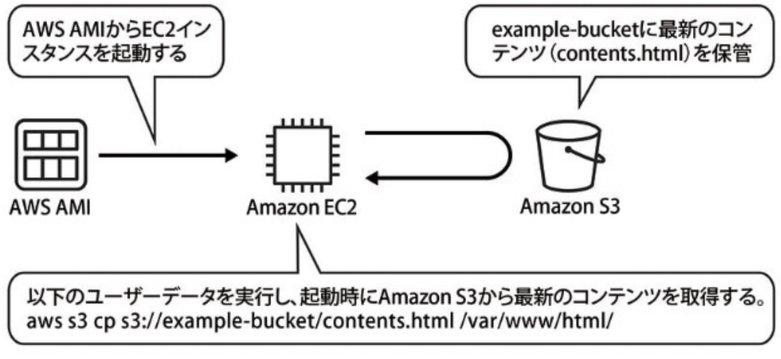
※ Amazon EBS-Backedインスタンスでは停止処理が可能。Instance Store-Backedインスタンスでは停止状態に遷移できない。

●ユーザーデータとインスタンスメタデータ

ユーザーデータとは、EC2インスタンスの初回起動時に1回だけスクリプトを実行できる機能です。AMIからEC2インスタンスを起動する際に、最新のコンテンツを反映する場合などに利用されます。

以下に、AMIからEC2インスタンスを起動した際に、S3バケットに保管されている最新のコンテンツを配布する例を示します。

【ユーザーデータの利用例】



インスタンスメタデータとは、EC2インスタンス自身に関するデータで、実行中のインスタンスを設定または管理するために使用されます。

インスタンスメタデータは、リンクローカルアドレス[※9]「http://169.254.169.254/latest/meta-data/」にブラウザやcURLコマンド[※10]でアクセスして取得することができます。

※9　【リンクローカルアドレス】169.254.0.0/16で定義されたローカルネットワーク上で使われる特別なIPアドレス。AWSの管理用としてバックエンドで利用されている。

具体的には、以下のような情報が取得できます。

- インスタンスID
- プライベートIPv4アドレス
- パブリックIPv4アドレス
- ローカルホスト名
- 公開鍵

試験対策

インスタンスメタデータで取得できる情報と、ユーザーデータの用途を覚えておきましょう。

新しい世代のEC2基盤のプラットフォームとして、AWS Nitro Systemが提供されるようになりました。EC2サービスは、AWS社が保有するデータセンターのホストサーバー上で、EC2用ソフトウェアを稼働させて提供されています。Nitro Systemでは、このソフトウェアをAWSが独自開発したハードウェアにオフロード（機能の一部をハードウェアで処理すること）することで、io1やio2などのEBSボリュームの利用時にさらに高いパフォーマンスを発揮します。

●Session Manager

EC2インスタンスへのログイン方法として、キーペアを使用した公開鍵認証方式がありますが、秘密鍵を紛失してしまうとログインができなくなります。

そのため、秘密鍵の管理をせずにEC2インスタンスにログインする仕組みとして、AWS Systems ManagerのSession Managerという機能を使ったログイン方法があります。Session Managerでは、鍵の代わりにIAMポリシーを使用してアクセス制御を行い、コンソールやCLIからワンクリックでEC2インスタンスにログインできます。また、AWS Systems ManagerのVPCエンドポイントを作成することで、プライベートサブネット上に作成されたEC2インスタンスに対しても、セキュアにアクセスできます。

※10　【cURLコマンド】さまざまな通信プロトコルでデータを送受信できるコマンド。

試験対策

Session Managerを使用したEC2ログインの設定方法について理解しておきましょう。

●Auto Scaling

Auto Scalingは、リソースの使用状況をモニタリングし、その状況に応じてEC2インスタンスなどのAWSリソースを、自動でスケールアウトまたはスケールインするサービスです。Auto Scalingは、対象サービスと機能によって次の3つのサービスに分類されます。

【Auto Scalingの種類】

種　類	対象サービス	機　能
EC2 Auto Scaling	EC2	スケーリングポリシーに従いEC2インスタンスをスケーリングする。
Application Auto Scaling	EC2以外	Amazon ECS サービスやAmazon EMRクラスター、Amazon Auroraレプリカなど、Amazon EC2以外のリソースを、使用状況に応じて自動スケーリングする。対象になるサービスは、以下を参照。 https://docs.aws.amazon.com/autoscaling/application/userguide/what-is-application-auto-scaling.html
AWS Auto Scaling	EC2、EC2以外	自動スケーリングと予測スケーリングの機能を持つ。予測スケーリングでは、過去のメトリクス（CPUの使用率などのリソース使用状況）を分析して、その結果に基づいて今後の利用量を予測し、適切なシステム利用状況になるよう管理する。EC2インスタンスやECSサービス、AuroraレプリカなどのAWSサービスが対象。対象になるサービスは、以下を参照。 https://docs.aws.amazon.com/autoscaling/application/userguide/integrated-services-list.html

EC2 Auto Scalingでは、あらかじめ設定したAmazon Machine Image（AMI）からEC2インスタンスを起動するため、AMIを常に最新にしておくことが重要です。アプリケーションなどの設定は、ユーザーデータを利用してAmazon S3やGitリポジトリからソースやスクリプトを取得することで、EC2インスタンスを最新の状態にすることができます。

EC2 Auto Scalingを実行するには、次の3つの設定を行います。

●スケーリングプラン

いつ、どのような条件でAuto Scalingを実行するかを定義します。具体的には以下の条件で実行が可能です。

- 正常なインスタンス数を維持するように実行
- 手動でスケーリングを実行
- スケジュールを指定して実行
- CloudWatchのメトリクス（CPUの使用率などのリソース使用状況）に応じて自動実行

このうち、自動実行の場合は以下のポリシータイプが設定可能です。

【ポリシータイプ】

ポリシータイプ	説　明
シンプルスケーリング	1つのスケーリングポリシーに基づいて実行
ステップスケーリング	Amazon CloudWatchアラームの設定に応じた一連のスケーリングポリシーに基づいて実行
ターゲットトラッキングスケーリング	特定のメトリクスのターゲット値を維持するように実行
予測スケーリング	過去のAmazon CloudWatch履歴データから予測データを作成し、その予測に基づいて実行 ※事前に履歴データが作成されている必要有

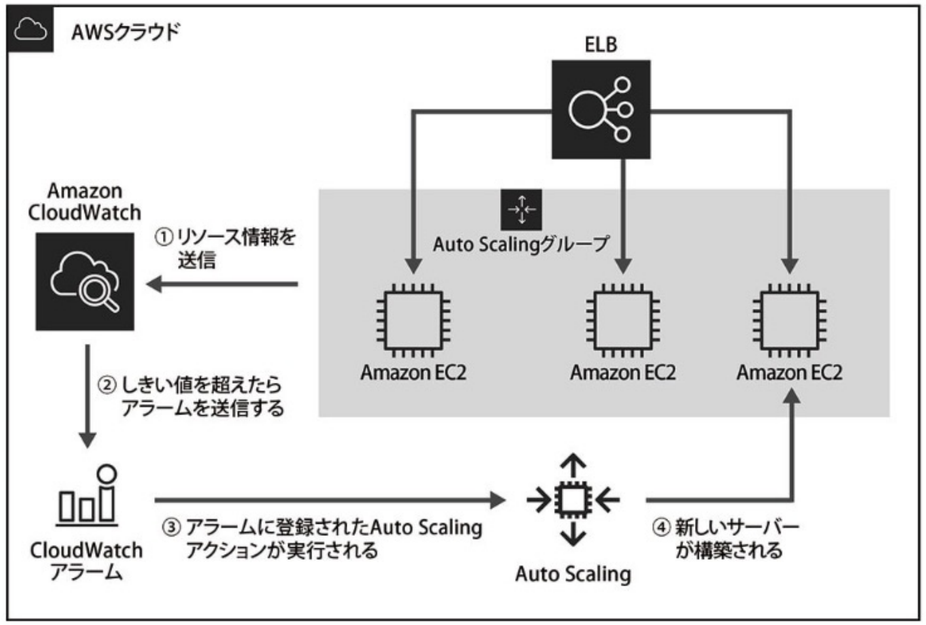
●起動設定

Auto Scalingの実行時に起動するEC2インスタンスの情報を定義します。具体的には、使用するAMI、インスタンスタイプ、セキュリティグループ、キーペアなど、Amazon EC2を構築する際に必要なパラメータを定義します。

●Auto Scalingグループ

EC2インスタンスの管理を行う範囲のことを指し、Auto Scalingで実際に起動するEC2インスタンスの最小数、最大数、希望数を定義します。ここで定義した範囲内でEC2インスタンス数が増減します。

【Auto Scalingのイメージ】



試験対策

スケーリングプラン、起動設定、Auto Scalingグループが、それぞれ何を設定する機能なのかを覚えておきましょう。

試験対策

Amazon EC2以外のAuto Scaling（Application Auto Scaling、AWS Auto Scaling）の設定方法についても確認しておきましょう。

2　コンテナ関連のサービス

AWSには、コンテナ関連のサービスとして以下のものがあります。

●Amazon Elastic Container Service（ECS）

Amazon ECSはフルマネージド型のコンテナオーケストレーションサービスです。ECSを利用することで、Dockerコンテナを簡単に実行、停止、管理することができます。

コンテナとは、ホストOS上に作成された論理的な領域で、その領域が通常のサーバーのように動作する仮想環境となります。また、**コンテナオーケストレーション**は、複数のコンテナの管理・起動・削除・監視などが行えるツールです。

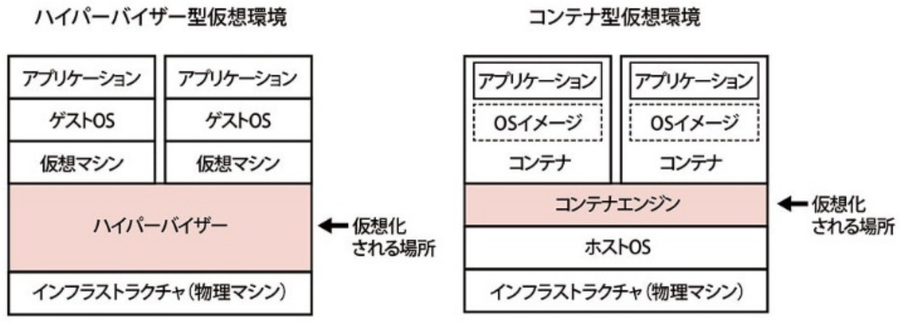
Docker[※11]は、ドッカー社が開発している「コンテナ型の仮想環境を作成、配布、実行するプラットフォーム」です。

「仮想環境」というと、ヴイエムウェア社のvSphereなどの仮想マシン環境をイメージするかもしれませんが、Dockerはそうした技術とは少し異なります。

VMwareをはじめとするハイパーバイザー型の仮想化ソフトウェアの場合は、次の図のように、物理マシン上でハイパーバイザーを利用し、その上でゲストOSを動作させます。

これに対してコンテナは、物理マシンのホストOS上にコンテナ仮想化ソフトウェアをインストールし、そのソフトウェア上で仮想環境を動作させます。

【ハイパーバイザー型仮想化とコンテナ型仮想化の違い】



※11　【Docker】コンテナ仮想化によってアプリケーションを迅速に開発・展開・実行するためのオープンプラットフォーム。読み方は「ドッカー」。次のWebページを参照。
https://docs.docker.com/

コンテナはOS部分の機能を共有しますが、CPUやメモリ、ネットワークなどの各リソースを論理的に分割することで、個々のコンテナの独立性を確保しています。そのため、1コンテナあたりのリソース消費を抑えることができ、1台のサーバーで多くのコンテナを稼働させることができます。

1台のサーバーでコンテナを1台だけ稼働させるのであれば、その管理は簡単です。しかし、業務で利用する場合は数十台や数百台、場合によってはそれ以上の規模のサーバー上でコンテナを稼働させることもあります。

これらをすべて手動で管理することは困難なため、実際に利用する際には複数のコンテナをシステムが管理するような機能が必要となります。この機能を持つサービスが、**Amazon ECS**です。

ECS環境を構築するには、以下のような作業を行います。

① クラスター作成

ECSはDockerコンテナ管理サービスですが、コンテナ自体はEC2インスタンスやオンプレミスのサーバーや仮想マシン(VM)、AWS Fargateなどのインフラストラクチャ上で動作します。それらのインフラストラクチャを論理グループ化して管理するタスクまたはサービスを「**クラスター**」といい、ECS環境の構築では、まずこのクラスターを作成します。

インフラストラクチャにEC2インスタンスを利用する場合は、クラスター作成時にEC2インスタンスタイプやインスタンス数、EBSストレージの容量などを設定できます。EC2インスタンス以外のインフラストラクチャを利用する場合は、クラスター作成後に紐付け設定する必要があります。

② タスク定義

ECSの実行環境を定義するために、タスク定義という設定を作成します。この定義では、Dockerイメージの格納先やメモリ使用量の制限、ネットワーク、ストレージ、ヘルスチェックに関する項目の設定を行います。

コンテナイメージを取得するタスク実行ロールや、コンテナから他のAWSサービスへのAPIリクエストに使われるタスクロールもここで指定します。

③ ELB設定

トラフィックをコンテナ全体に分散して処理できるようにするために、Elastic Load Balancing(ELB)の設定を行います。

④ サービス設定

クラスター上で起動するコンテナ数や起動するコンテナの最大数などを設定します。

以上の設定を行うと、設定された内容どおりにコンテナ環境が動作するようECSが管理してくれます。

試験対策

Amazon ECSで利用されるタスクロールと、タスク実行ロールの違いを覚えておきましょう。

●Amazon Elastic Container Service Anywhere(ECS Anywhere)

ECSをユーザーが所有するインフラストラクチャ環境上で実行させたい場合は、Amazon ECS Anywhereを利用することで実現できます。

ただし、既存のインフラストラクチャ環境でコンテナを実行できることはメリットですが、対応しているOSが限定されていたり、実行するインフラストラクチャ環境からAWS APIに接続できる必要があるなど、いくつかの制限事項があるため注意が必要です。

●Amazon Elastic Kubernetes Service(EKS)

EKSはECSと同様のフルマネージド型のコンテナオーケストレーションサービスの1つですが、Kubernetes向けのサービスであることがECSとは異なります。

Kubernetes[※12]は、分散されたホスト上でDockerなどのコンテナを利用するためのオーケストレーションツールです。ユーザーからは、分散されているホストを意識することなく、1台のマシンで動作しているコンテナにアクセスするような感覚で利用できます。

ECSと同様に、ユーザーがインスタンス数やEBSストレージの容量を定義すると、状況をモニタリングしながらその状態になるように自動管理され、負荷の増減についても自動でスケールして対応できる仕組みになっています。さらに、多くのコンテナを効率よく管理できる仕組みも備わっています。

※12 【Kubernetes】Dockerなどのコンテナを運用管理、自動化するためのオープンソース・ソフトウェア。読み方は「クーベネティス」または「クバネティス」など。次のWebページを参照。
https://kubernetes.io/ja/

Amazon ECSとAmazon EKSとの違いは、コンテナのオーケストレーションにあります。ECSはAWS独自のコンテナオーケストレーションを採用していますが、EKSはKubernetesを採用しています。

●Amazon Elastic Kubernetes Service Distro（EKS Distro）

EKSの機能を使用しながらもKubernetesの管理をユーザー側で行いたい場合は、オープンソースとして公開されているEKS Distroを使用すると実現できます。

EKS DistroはKubernetesディストリビューションとして提供されているソフトウェアで、インストールする環境をAWSに限定しないため、ユーザーが保有しているインフラストラクチャ環境にインストールができます。

ただし、EKS Distroをインストールした環境ではAWSのEKSコンソールは使用できないため、EKSの機能を使用してKubernetesクラスターの管理やライフサイクル管理などを行うことができます。

●Amazon Elastic Kubernetes Service Anywhere（EKS Anywhere）

EKS Distroを使用してAWS以外の環境に構築されたKubernetesクラスターをAWSのEKSコンソールで管理したい場合は、Amazon EKS Anywhereを使用します。

EKS AnywhereのEKS Connectorという機能を利用すると、EKSコンソールからEKS Anywhereのコマンドで構築したKubernetesクラスターやKubernetesリソースを確認することができます。

●AWS Fargate

コンテナ向けの**サーバーレスコンピューティングエンジン**で、ECSとEKSの両方で動作します。

手動でコンテナの管理を行う場合は、たとえば次のような作業が必要になります。

- コンテナイメージのビルド
- ECSインスタンスの定義とデプロイ
- コンピューティングリソースおよびメモリリソースのプロビジョニングと管理
- アプリケーションを別々の仮想マシンに分離させる
- アプリケーションとインフラの両方を別々に実行して管理

これらの作業は手間がかかりますが、Fargateを利用することでユーザーは、コンピューティングリソースとメモリリソースの定義やアプリケーションの実行・管理を行うだけでよく、スケールを含むその他のコンテナの管理をFargateに任せることができます。

試験対策

最近ではコンテナ関連のサービスがよく利用されています。コンテナ関連のサービス名とその機能を覚えておきましょう。

Amazon ECSやAmazon EKSは、コンテナを管理するサービスに分類されます。ただし、これらのサービスでコンテナそのものが動作するわけではありません。実際にコンテナが動作する環境は、EC2インスタンスやFargateです。
EC2インスタンスを利用すると詳細な設定ができるようになりますが、その分だけEC2インスタンスの管理が必要になります。
また、Fargateはサーバーレスなサービスであるため、EC2のような動作環境の管理が不要になります。

●Amazon Elastic Container Registry（ECR）

Amazon ECRはフルマネージド型のコンテナイメージレジストリサービスです。Dockerコンテナイメージをセキュアに管理することができます。

Dockerコンテナイメージとは、Dockerコンテナを作成するためのOSやコマンド、設定などをまとめたファイルで、コンテナはこのコンテナイメージに記載されているとおりに構築、起動されます。ECSやEKSでは、ECRやDocker Hubなどのレジストリサービスに登録されたイメージを元にコンテナが起動されます。

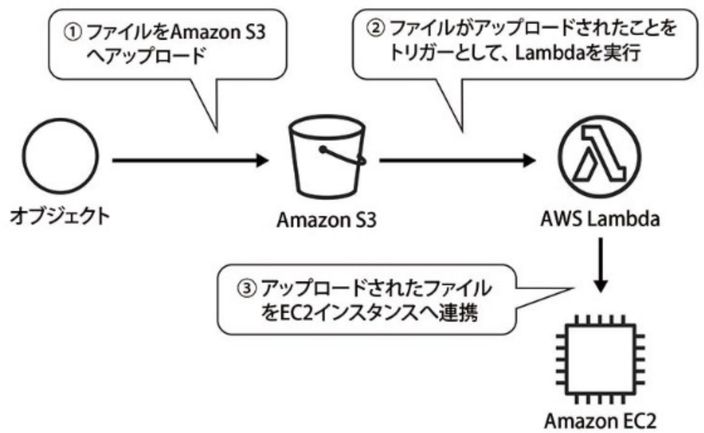
3　AWS Lambda

AWS Lambdaは、サーバーなどのコンピューティングリソースを意識することなく、「**Lambda関数**」と呼ばれるアプリケーションコードをデプロイしただけで実行することができるサーバーレスなサービスです。アプリケーションを実行するインフラストラクチャはAWSが管理するため、スケーリングも不要で、

実行する際のリソース(メモリ容量)と実行時間に対して課金されます。ただし、Lambdaは実行時間に制限があるため、時間を要する処理には不向きです。

LambdaはさまざまなAWSサービスと連携が可能で、イベント駆動型のアプリケーションを実行することができます。たとえば、Amazon S3にコンテンツファイルがアップロードされたことをトリガーとして、Lambdaでアップロードされたファイルを EC2インスタンスへ連携することができます。

【Lambdaへの連携イメージ】



参考

AWS Lambdaがサポートしているサービスの詳細は、以下を参照してください。
https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/lambda-services.html

参考

AWS Lambdaがサポートしているプログラミング言語については、以下を参照してください。
https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/lambda-runtimes.html

●環境変数

Lambdaではアプリケーションコードとは別に、関数ごとに環境変数を設定することができます。

環境変数はキーと値のペアで構成され、キーには文字、数字、アンダースコア「_」のみ設定できます。設定された環境変数はアプリケーションコードからキー名で呼び出すことができるため、キー名に紐付く値が変更されたとしても、アプリケーションコードを書き換える必要がありません。たとえば、開発環境のLambda関数から接続するデータベース名を環境変数に設定すれば、本番環境にデプロイする場合は、環境変数のデータベース名のみを書き換えるだけで済みます。

試験対策

ここでは、一例としてデータベースの認証情報を環境変数に設定する方法を説明しましたが、セキュリティの観点から、認証情報などの情報漏洩対策が必要なパラメータは、環境変数での設定を推奨されていません。認証情報をアプリケーションコードで使用する際は、AWS Secrets ManagerやAWS Systems Manager Parameter Storeと組み合わせて設定する方法が推奨されていることを、理解しておきましょう。

●ExecutionRole

LambdaにアタッチされたIAMロールのことです。

Lambdaは、IAMロールの権限に従って各AWSサービスへアクセスするため、IAMロールによるアクセス制御設計が必要になります。そのために、このExecutionRoleを利用します。

●ロギング

すべてのLambda関数の処理結果は、CloudWatch Logsに保存されます。

●Lambda@Edge

Lambdaという名前が付いていますが、実際にはAmazon CloudFrontの機能です。

Lambda関数は、アプリケーションのユーザーに最も近い場所でコードが実行されるため、アプリケーション実行時のパフォーマンスが向上し、待ち時間が短縮されます。

Lambda@Edgeの詳細は、「4-2　ネットワークサービスにおけるパフォーマンス」を参照してください。

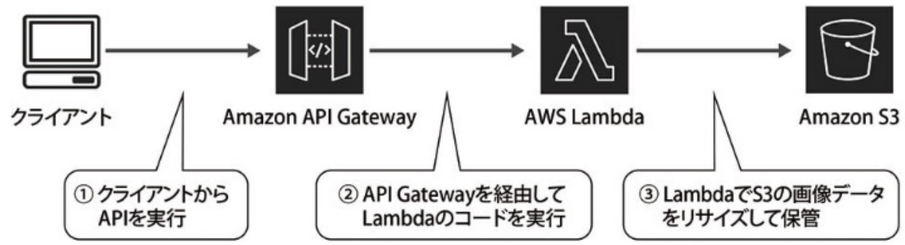
4 Amazon API Gateway

Amazon API GatewayはAPIの作成、配布、保守、監視、保護を簡単に行えるサービスです。

EC2インスタンスやLambdaと組み合わせることで、Webアプリケーションのバックエンドやデータ、ビジネスロジックにアクセスすることができます。API Gateway自体はサーバーを管理する必要がないため、Lambdaと組み合わせることでサーバーレスアプリケーションを実装することができます。

次の図は、クライアントからAPIを実行するだけで、Amazon S3バケットに保管された画像ファイルを自動でリサイズすることを想定したサーバーレスなシステムの例です。

【API GatewayとLambdaの連携イメージ】



●APIエンドポイントタイプ

API Gatewayを作成すると、特定のリージョンにAPIがデプロイされます。その際にデプロイされたAPIのホスト名のことを、**APIエンドポイント**と呼びます。

APIエンドポイントにはAPIトラフィックの発信元によって、3つのタイプが存在しています。また、各APIによって使用可能なエンドポイントタイプが異なります。各APIエンドポイントタイプの違いと使用可能なAPIを、次の表に示します。

【APIエンドポイントタイプ】

エンドポイント名	トラフィックの発信元	使用可能なAPI
リージョンAPIエンドポイント	API Gatewayと同一リージョン内のクライアント	HTTP API、REST API
エッジ最適化APIエンドポイント	地理的に分散されたクライアント	REST API
プライベートAPIエンドポイント	VPC内のクライアント	REST API

APIエンドポイントタイプの違いについて理解しておきましょう。

5 アプリケーションデプロイ関連のサービス

AWSには、アプリケーションデプロイ関連として以下のサービスがあります。

●AWS Amplify

AWS Amplifyは、Webアプリケーションやモバイルアプリの開発、デプロイ、管理を簡単に行うための開発プラットフォームです。Amplifyには以下のようなサービスやツールが用意されており、これらを利用することで、数回のコマンドや画面操作でバックエンドのAWSサービスが構築できたり、フロントエンドのアプリケーションとの連携を簡単に実装することができます。

- Amplify CLI
- Amplify Framework
- Amplify Developer Tools

たとえば、Amplify CLIで構築する場合、以下の構文でコマンドを実行します。

カテゴリは、認証、API、ストレージなど構築したいサービスに応じて選択します。

【Amplify CLIの構文】

コマンド	コマンド内容
amplify add [カテゴリ名]	設定の追加
amplify update [カテゴリ名]	設定の更新
amplify remove [カテゴリ名]	設定の削除
amplify status [カテゴリ名]	設定ステータス表示
amplify push [カテゴリ名]	バックエンド構築

コマンドによってバックエンドが構築されると、構築したAWSサービスの情報が記載された設定ファイルが生成されます。

生成された設定ファイルをフロントエンドのアプリケーションに読み込ませることで、アプリケーションからバックエンドとして構築したAWSサービスに接続させることができます。

試験対策

AWS Amplifyの各サービスと使い方について理解しておきましょう。また、Amplifyで構築可能なAWSサービスも確認しておきましょう。

参考

GUIで提供されるグラフィカルな開発環境として、Amplify Studioと呼ばれる機能も用意されています。

●AWS Elastic Beanstalk

AWS Elastic Beanstalkは、少ない手順でWebアプリケーションやサービスをサーバーにデプロイでき、また、実行環境も管理できるサービスです。アプリケーションを実行しているインフラストラクチャについて深く理解していなくても、アプリケーションのデプロイとその管理を簡単に行うことができます。

開発したWebアプリケーションをアップロードすると、Elastic Beanstalkが自動的に容量のプロビジョニング、負荷分散、拡張、アプリケーション状態の監視を行います。このほかにも、アップロードしたWebアプリケーションのライフサイクル管理や時間ベースのスケーリング設定などにも対応しているため、構築したいシステムに合わせた柔軟な設定が可能です。

Elastic Beanstalkが対応しているアプリケーションには、Java、.NET、PHP、Node.js、Python、Ruby、Goがあります。また、コンテナ(Docker)にも対応しています。

アプリケーションをデプロイするサーバーとしては、Apache HTTP Server、Nginx、Passenger、Puma、Microsoft Internet Information Services(IIS)などがあります。

試験対策

AWS Elastic Beanstalkで使用可能なデプロイポリシーについても確認しておきましょう。

参考

AWS AmplifyとAWS Elastic Beanstalkはどちらもアプリケーションのデプロイと管理の機能を提供しますが、それぞれのサービスを利用する目的が異なります。
AmplifyはAWS上で開発を行う際のプラットフォームの提供を行うため、アプリケーション開発の簡略化と迅速化が利用する目的になります。
一方のElastic Beanstalkは、アプリケーションをアップロードするのみで開発機能は提供されていないため、インフラストラクチャのデプロイ・管理の簡略化と迅速化が目的になります。どちらのサービスが最適かの判断は、構築するアプリケーションやシステムによって異なるため、事前に目的や用途を確認しておく必要があります。

6　移行関連のサービス

AWSには、次のような移行関連のサービスがあります。

●VM Import/Export

VM Importは、オンプレミス環境に作成した仮想サーバーのマシンイメージを、AWSのEC2インスタンスにインポートすることができるEC2サービスの機能の1つです。

VM Exportは、VM ImportによってAWS上にインポートされたEC2インスタンスを、オンプレミス環境で動作する仮想サーバーのマシンイメージへエクスポートする機能です。

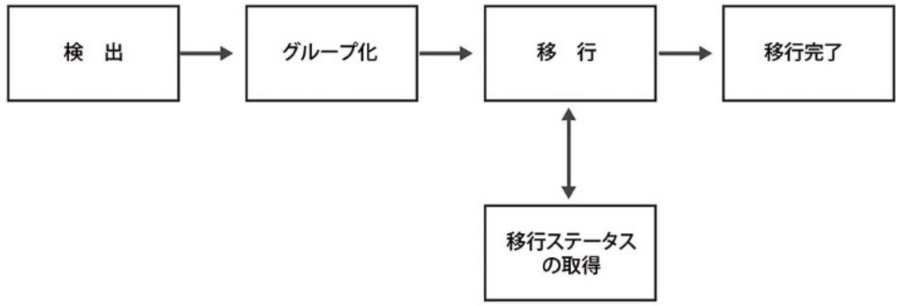
これらの機能によって、既存の仮想マシンを無駄にすることなく容易にAWS上に移行することが可能となります。

●AWS Migration Hub

AWS Migration Hubは、オンプレミス環境などで稼働する既存サーバーをAWSに移行する際の移行対象や移行計画、移行ステータスを、一元的に管理するための機能を提供するサービスです。

Migration Hubを使用する際の移行フローを、次の図に示します。また、検出とグループ化を使用せずに、直接移行することもできます。

【移行フロー】



フローの検出と移行の機能はMigration Hubでは提供されておらず、別のAWSサービスを利用する必要があります。

以下にそのAWSサービスを記載します。

● 検出

- Migration Hubインポート
- Migration Evaluator Collector
- AWS Agentless Discovery Connector
- AWS Application Discovery Agent

● 移行

- AWS Application Migration Service（AWS MGN）
- AWS Server Migration Service（AWS SMS）
- AWS Database Migration Service（AWS DMS）

> 参考
> AWS Server Migration Service（SMS）は2022年3月31日で廃止され、2022年4月1日以降はAWS Application Migration Service（AWS MGN）が推奨されています。

●AWS Application Discovery Service

AWS Application Discovery Serviceは、オンプレミス環境にある移行対象サーバーの使用状況と構成データなどを収集するサービスです。収集されたデータはMigration Hubのコンソールに集約・管理されます。

データの収集方法として、エージェントレス検出（AWS Agentless Discovery Connector）とエージェントベース検出（AWS Application Discovery Agent）の2つがあります。VMWare仮想マシンからの収集であればエージェントレス検出、それ以外のVMまたは物理サーバーからの収集であればエージェントベース検出を利用します。

●AWS Application Migration Service（AWS MGN）

AWS Application Migration Serviceは、オンプレミス環境やVMware仮想マシンなどからAWSへ、アプリケーションを移行するためのサービスです。移行元の既存サーバーにレプリケーションエージェントをインストールし、AWS上にデータをレプリケーションして、AWS MGNコンソールから行う移行テストで問題なければ、カットオーバー作業を行うことで移行ができます。また、AWS Migration Hubと事前に接続しておくことで、移行の進行状況とステータスをMigration Hubのコンソールから確認できます。

> 参考
> AWS Application Migration Service（MGN）は、一部のAWSリージョンではサポートされていないため、利用する際は、まずサポートされたリージョンか確認する必要があります。
> 利用したいリージョンでサポートされているAWSサービスは、次のWebページから確認することができます。
> https://aws.amazon.com/jp/about-aws/global-infrastructure/regional-product-services/

7 その他のコンピューティングサービス

AWSには、以下のようなコンピューティングサービスも用意されています。

●Amazon Lightsail

Amazon Lightsailは、オールインワンのプラットフォームとして設計されたコンピューティング サービスです。アプリケーションやWebサイトをすばやく構築できるように、仮想サーバーやストレージ、データベース、ネットワークなどの環境を簡単に構成することができます。

●AWS Batch

AWS Batchは、インフラストラクチャの管理が不要なバッチ処理の実行環境を提供するサービスです。定期的に特定コマンドを実行したい場合や、GPUが必要な計算を手軽に行いたい場合などに利用できます。

AWS Batchの作業単位をジョブと呼びます。ジョブはコンテナで実行されるため、事前にDockerイメージの準備が必要です。ジョブを実行するために必要なパラメータや実行環境などは、ジョブ定義で指定できます。

●AWS Serverless Application Repository

AWS Serverless Application Repositoryは、開発したサーバーレスアプリケーションをデプロイ・管理するためのサービスです。デプロイしたアプリケーションは公開することもでき、公開したアプリケーションは他のユーザーが利用できます。アプリケーションを公開するには、AWS Serverless Application Modelテンプレートのマニフェストファイルをアップロードする必要があります。

> 参考
> AWS Serverless Application Model（SAM）とは、サーバーレスアプリケーションを開発する際に使用できるOSSのフレームワークの1つです。
> AWS SAMでは、ローカルでのデバッグ機能やCloudFormation拡張のデプロイ機能などが提供されているため、アプリケーション開発の速度を高めながら管理されたインフラストラクチャを構築することが容易になります。

●VMware Cloud on AWS

VMware Cloud on AWSは、AWSのインフラストラクチャ上で稼働するVMwareソフトウェアベースのクラウドサービスです。オンプレミス環境で稼働するVMware環境をAWSに移行することで、既存のVMware環境を大幅に変更することなく、AWSの耐障害性や拡張性といったメリットを享受することができます。

●AWS Device Farm

AWS Device Farmは、モバイルアプリやWebアプリケーションをテストするために、実際の電話やタブレットの環境を提供するサービスです。自動アプリテストとリモートアクセス操作が提供されており、ユーザーはデバイススロットと呼ばれる単位でスロット数を購入し、購入したスロットの分だけデバイスを利用できます。

> 参考
> AWS Device Farmはオレゴン地域（us-west-2）のみで利用できます。

●Amazon Pinpoint

Amazon Pinpointは、ユーザーが指定した特定の対象者に複数の方法でメッセージを送り、その対象者の行動や反応の分析などができるサービスです。また、セグメントと呼ばれる単位で対象者をグルーピングすることで、送信先の管理を行うこともできます。

Amazon Pinpointで利用可能なメッセージ送信方法には、以下のような方法があります。

- プッシュ通知
- メール
- SMS
- 音声
- アプリケーション内メッセージ
- カスタムチャンネル経由メッセージ

●AWS AppSync

AWS AppSyncは、GraphQLをマネージドで提供するサービスです。GraphQLとは、API向けに作られたクエリ言語のことで、1つのエンドポイントを使用して必要なデータのみ取得できるため、APIリクエストの回数を削減できるなどの特徴があります。

Amplifyにも統合されており、フロントエンドのアプリケーションからバックエンドのAWSサービスにクエリを実行する際に利用されるケースもあります。

●AWS Compute Optimizer

AWS Compute Optimizerは、Amazon EC2やAmazon EBSなどのAWSリソースについて、設定項目のチェックやメトリクス分析を実行し、ワークロードを最適化するための推奨事項を作成するサービスです。対象リソースは、EC2インスタンス、EC2 Auto Scalingグループ、EBSボリューム、Lambda関数の4つで、推奨事項を作成するためにはメトリクスデータがある程度取得されている必要があります。

有料機能を有効化することで、過去3カ月間のメトリクス分析結果を閲覧できます。デフォルトでは過去14日間分の結果を見ることができます。

●Amazon Elastic Transcoder

Amazon Elastic Transcoderは、Amazon S3上に保存されたメディアファイルを変換するためのサービスです。変換処理はジョブというコンポーネントで実行し、ジョブはパイプラインと呼ばれるグループにまとめられ、実行順序などを含めて管理されます。変換処理にはデフォルトでプリセットがいくつかあり、カスタマイズした独自のプリセットを作成することもできるようになっています。

Q 演習問題

1 **ある会社では、システムの担当者はSSH鍵を使用して複数のEC2インスタンスを管理していますが、管理のオーバーヘッドを極力減らしたいと考えています。次の選択肢のうち、要件を満たすために使用するソリューションを1つ選択してください。**

A　AWS Security Token Service（AWS STS）

B　AWS Systems Manager

C　AWS Command Line Interface（AWS CLI）

D　AWS Key Management Service（AWS KMS）

2 **ある企業のアプリケーションはAPI Gateway、Lambda、DynamoDBで構築されており、毎日かなりの量のリクエストが発生しています。アプリケーションを世界中に展開し、常にミリ秒のレイテンシーでアクセスできるようにするためには、どのAPI Gatewayエンドポイントタイプでデプロイしたらよいですか。**

A　リージョンAPIエンドポイント

B　グローバルAPIエンドポイント

C　エッジ最適化APIエンドポイント

D　プライベートAPIエンドポイント

3 **オンプレミス環境のサーバーで実行中のアプリケーションとデータベースを、最小限の作業でAWSに構築したいと考えています。また、AWSに構築後も、アプリケーションで発生した更新を本番環境に反映させる際に、ダウンタイムを発生させないソリューションを、以下から1つ選択してください。**

A　AWS CodeDeploy

B　AWS Amplify

C　AWS Elastic Beanstalk

D　AWS CloudFormation

A 解答

1 B

EC2へのログイン方法の知識を問う問題です。SSH鍵を使用せずにEC2インスタンスにログインするためには、AWS Systems ManagerのSession Manager機能を使用します。SSH鍵の管理が不要になるので、管理オーバーヘッドを減らすことができます。
問題文では言及されていませんが、Session Manager機能を使用するほかに、AWS Systems ManagerのVPCエンドポイントの作成が必要です。
B以外の選択肢では、EC2のログインはできません。

2 C

API GatewayのAPIエンドポイントは、APIトラフィックの発信元によって選択します。
問題文では、世界中にアプリケーションを展開するとあるので、地理的に分散されたクライアントからのアクセスに適したエッジ最適化APIエンドポイントでデプロイします。
AとDのエンドポイントは、アプリケーションと同一リージョン、または同VPC内からのアクセスに適したもので、Bのエンドポイントは存在しません。

3 C

AWSのデプロイサービスについての知識を問う問題です。
問題文から、アプリケーションとインフラのどちらもデプロイ可能なサービスを選択する必要があるため、DのAWS CloudFormationは適当ではありません。
また、AのAWS CodeDeployは構築作業に多くの手順が必要で、BのAWS Amplifyではデータベースのデプロイは提供されていません。
アプリケーション更新時のダウンタイムの最小化はいずれも実現可能ですが、最小限の構築作業かつデータベースもデプロイ可能なサービスは、**C**のAWS Elastic Beanstalkのみです。

1-5 ストレージサービス

AWSでは、代表的なストレージサービスのAmazon S3やAmazon EBSをはじめ、さまざまなストレージサービスが提供されています。本節では、AWSにおける各ストレージサービスの概要を説明します。

1 Amazon Simple Storage Service（S3）

Amazon S3は、高耐久・大容量のオブジェクトストレージサービスです。**オブジェクトストレージ**とは、データをオブジェクトとして扱い、IDとメタデータによって管理する方式です。S3では「**バケット**」と呼ばれる単位で保管するオブジェクトを管理します。

S3に保存したオブジェクトは、デフォルトで**同一リージョン内の3カ所のアベイラビリティーゾーン（AZ：Availability Zone）**へ自動的に複製されます。

オブジェクトの耐久性は99.999999999％（イレブン・ナイン）に達し、オブジェクトストレージとしてだけでなくバックアップストレージとしても利用できます。また、利用できる容量は実質的に制限がなく、利用した分だけの料金が発生するため、非常に導入しやすいことが特徴です。ただし、1ファイルあたりのサイズには制限があるので注意が必要です。

以下では、S3の機能について詳細を説明します。

●データの復元性と大規模災害対策に関するS3の機能

S3は非常に高い耐久性を備えており、データ損失の可能性は限りなく低くなっています。しかし、誤ってデータを削除してしまうケースや、大規模災害の影響で特定のリージョン内に保管したS3のデータが利用できなくなるケースなども考えられます。

誤ってデータを削除するケースでは、オブジェクトロック機能やバージョニング機能を有効化することで対策できます。また、大規模災害などの影響でS3が利用できないケースでは、クロスリージョンレプリケーションで対策できます。

●オブジェクトロック機能

オブジェクトロック機能は、オブジェクトの上書きまたは削除の操作を防止できる機能です。オブジェクトロックを設定することで、オペレーションミスなど意図しないオブジェクトの削除を防止することができます。ロック期間として、一定期間または無期限を設定することができます。

●バージョニング機能

バージョニングとは、オブジェクトの世代管理を提供する機能です。保管しているオブジェクトを誤って更新・削除した際に、設定に基づいて以前の世代へ戻すことが可能です。バージョンは、オブジェクト単位でバージョンIDにより管理されます。

S3では、同名のオブジェクトをアップロードすると、標準では上書きされます。また、MFA Deleteを有効化すると、特定のバージョンを削除したりバケットのバージョニング状態を変更する際に、多要素認証を要求するように設定することもできます。

●クロスリージョンレプリケーション

S3に保存したデータは、デフォルトで同一リージョン内の3カ所のAZへ自動的に複製されますが、クロスリージョンレプリケーションを有効化すると、オブジェクトを別リージョンのS3バケットに自動的に複製できます。複製先には、別のAWSアカウントも指定できます。

Amazon S3は、高耐久性ストレージとして、データストレージをはじめバックアップやアーカイブ用途、データレイクなどで利用します。標準で高い可用性を備えていますが、オブジェクトロック機能やバージョニング機能を利用して、意図しないデータロストからデータを保護できることが重要です。

クロスリージョンレプリケーションとは別に、同じAWSリージョン内のバケット間でオブジェクトを自動的に複製するセイムリージョンレプリケーションという機能が用意されています。クロスリージョンレプリケーションは主にDR（災害復旧）などで利用しますが、セイムリージョンレプリケーションは、さまざまなログを集約したり、アクセス権限を変更して読み取り専用から外部からの閲覧を可能にする、などの用途で主に利用されます。

●S3におけるセキュリティ対策

S3には、重要なデータや閲覧者を限定したデータが保管されることもあるため、データのセキュリティを確保するための施策が非常に重要です。

●アクセス制御

S3は、主に以下の4つの方法でアクセス制御を行います。

● IAM ポリシーによるアクセス制御

IAMポリシーによる制御を行います。IAMユーザーやIAMロールからRead、Get、Put操作に対する制御などを行うことが可能です。

● バケットポリシーによるアクセス制御

バケットポリシーとは、あるS3バケットに対して、JSON[※13]コードを記載してバケット全体のアクセス制御を行う機能です。バケットポリシーを設定することで、IPアドレスによる制御やIAMユーザーに対して、S3バケットのアクセス権を細かく制御することができます。

● ACL によるアクセス制御

S3バケットに対して、AWSアカウントレベルのアクセス制御を行います。ACLからパブリックアクセスの設定を行うことで、不特定多数のユーザーへファイルを公開することも可能です。

● ブロックパブリックアクセスによるアクセス制御

前述したアクセス制御のうち、特にバケットポリシーとACLによるアクセス制御では、S3の各バケットや各オブジェクトデータを監視しておかないと、意図せず外部に公開されてしまうという問題がありました。そのような場合は、**ブロックパブリックアクセス**によるアクセス制御を行うことで、バケットポリシーやACLの設定に対して一括で制限を設けることができます。

●暗号化

S3バケットに保管されたデータは暗号化されます。暗号化には、クライアントサイドの暗号化と、サーバーサイドの暗号化の2種類があります。サーバーサ

※13 【JSON】JavaScript Object Notation：データ記述言語の1つ。キーとバリューの構成でデータを表現する。

イドでは、以下の3種類の方式で暗号化を行います。

・S3のデフォルトキーを使用したAES-256[※14]暗号化
・AWS Key Management Service(KMS)で管理されている鍵によるAES-256暗号化
・ユーザーの任意の鍵によるAES-256暗号化

S3のデフォルトキーまたはAWS KMSを利用した暗号化は、AWS側で鍵の管理を行いますが、ユーザーの任意の鍵を使用する場合は、暗号化鍵と暗号化したオブジェクトのマッピングは、ユーザーが管理する必要があります。

●S3のコスト最適化

S3は、容量あたりの料金が非常に安価に抑えられていますが、用途に応じた適正な機能を利用することで、さらにコストを最適化できます。S3では、ユースケースに応じて「**ストレージクラス**」と呼ばれる利用レベルを選択することができます。

主なストレージクラスは、以下のとおりです。

● 標準(Standard)

デフォルトのストレージクラスで、99.999999999%の高耐久性を備えています。

● 標準 - 低頻度アクセス(Standard - Infrequent Access)

スタンダードと同じ耐久性を備えたストレージクラスで、スタンダードと比較して低コストで利用可能です。ただし、データの読み取りに対して課金されるため、長期保管やバックアップ用途向けに利用されます。

● 1 ゾーン - 低頻度アクセス(One Zone - Infrequent Access)

標準低頻度アクセスほど耐久性を必要としない場合に利用するストレージクラスです。1カ所のAZのみにデータを保存するため、標準低頻度アクセスよりさらに安価で利用することができます。

※14 【AES-256】Advanced Encryption Standard 256bit：暗号化鍵に256ビット長を使用する暗号化方式。「AES」と呼ばれる。DES(Data Encryption Standard)よりも暗号化強度が高いとされる。

● 低冗長化ストレージ(Reduced Redundancy Storage)

スタンダードでは3カ所のAZにレプリケーションされるのに対して、2カ所にすることで、耐久性は低下するもののコストを抑えて利用できるストレージクラスです。

● Intelligent-Tiering

低頻度・高頻度の2階層のストレージ層を用意し、S3上に格納したオブジェクトへのアクセス頻度に応じて、コストに見合った最適なストレージ層を自動的に使い分けるストレージクラスです。S3がオブジェクトを最適な階層に格納するため、コスト最適化が自動的に行えます。

● Amazon S3 Glacier

アーカイブを目的としたストレージで、大容量のデータをより安価に保管することができます。Glacierには3種類のストレージクラスがあります。

Glacier Instant Retrieval

Glacier Instant Retrievalは、おおよそ四半期に1回程度しかアクセスされることがなく、S3スタンダードと同等のミリ秒単位の取り出しが必要な長期間保管用データに最適なアーカイブストレージクラスです。医療画像やゲノムデータなど、普段は利用しないものの急に必要とされるデータの利用に適しています。

Glacier Flexible Retrieval

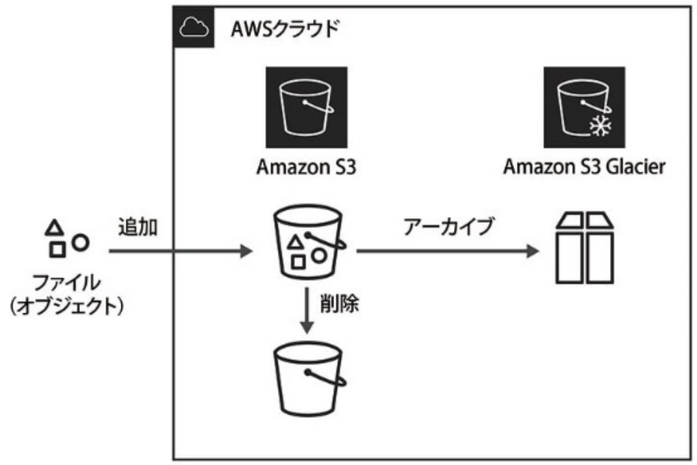
Glacier Flexible Retrievalは、従来ではGlacierと呼ばれていたクラスを指し、年に1、2回程度しかアクセスされることがないものの、時々取り出す必要があるデータに最適なアーカイブストレージクラスです。データの取り出しに数分～数時間の時間が許容されるバックアップデータなどの利用に適しています。

Glacier Deep Archive

Glacier Deep Archiveは、S3のなかで最も低コストに利用できるストレージクラスで、年に1、2回程度しかアクセスされないようなデータに最適なアーカイブストレージクラスです。データの取り出しには数時間以上を要しますが、データ保管要件が厳しい業界や数年～十数年以上にわたってデータを保管する要件に適しています。

また、S3では、ライフサイクルポリシーを設定することで、設定した期間が経過したあと、オブジェクトを自動で削除したりアーカイブすることができます。これは、バックアップデータやログデータなど、保存期間の要件を過ぎたデータの削除／アーカイブに手を加えることなく自動で実行できるため、コストの抑制につながります。

【ライフサイクルポリシー】



試験対策

Amazon S3では、保管するデータの特性に応じてストレージクラスやライフサイクルポリシーを使い分けることで、よりコストを抑えることができます。保管するゾーンを減らしてコストを最適化するのか、アクセス頻度が少ないデータに対してコストを最適化するのかなど、観点を整理することが重要です。

●S3のデータ転送効率化を実現する機能

S3へオブジェクトをアップロードする際に、**マルチパートアップロード**という機能が利用できます。これは、大容量のオブジェクトを「パート」と呼ばれる複数の単位に分割してアップロードすることで、効率よく転送できるようにします。すべてのパートのアップロードが完了すると、S3側で自動的にオブジェクトの再構築が行われます。

このほか、**Amazon S3 Transfer Acceleration**という機能を有効化すると、クライアントとS3間の通信を高速化させることができます。この機能では、データ転送時にエッジロケーションとAWSネットワークを利用することで、通信の高速化を図っています。

●ストレージ以外でのS3の用途

S3はオブジェクトストレージだけではなく、以下のような用途でも利用されています。

● Web サイトホスティング

S3は、**静的コンテンツのWebサイトホスティング**の機能も備えています。静的なコンテンツを配信するWebサイトであれば、サーバーを構築することなく公開できます。

また、簡易的なWebサイト以外に、Route 53のDNSフェイルオーバー機能と組み合わせて、WebサイトのSorryページ[※15]としてもよく利用されます。

● 署名付き URL

S3では、AWS CLIやAWS SDKを利用して、署名付きURLを発行することができます。**署名付きURL**とは、S3上のデータに対して一定時間だけアクセスを許可するためのURLを発行する機能で、これによりAWSへログインできないユーザーでもデータにアクセスすることができます。アクセス可能な時間は自由に決めることができるため、ファイルの受け渡しをセキュアに行うことができます。

試験対策

Amazon S3のWebサイトホスティングを利用することで、サーバーを管理することなく静的コンテンツを配信できます。S3だけでも構築できますが、Amazon CloudFrontと連携するケースが多くあります。

2　Amazon Elastic Block Store（EBS）

Amazon EBSは永続可能なブロックストレージサービスで、EC2インスタンスにアタッチすることで利用できます。**ブロックストレージ**とは、データを「**ブロック**」と呼ばれる細かい単位で分割して保管する方式です。

EBSは、Amazon S3のAZ間での複製とは異なり、AZ内で自動的に複製され

※15 【Sorryページ】Webサイトのメンテナンスや障害発生時に表示するページのこと。