

Column



AWSの歴史

パブリッククラウドの先駆者であるAWSのサービスが開始されたのは2006年です。当時はまだクラウドという言葉もなく、米国アマゾン・ドット・コム社が自社のIT設備の余剰リソースを、一般向けに提供したことが始まりです。

その構想は、2003年に当時のアマゾン・ドット・コム社のネットワークエンジニアであったベンジャミン・ブラック氏が、その上司のクリス・ピンカム氏とまとめた論文がきっかけになりました。

論文には、自社のIT設備を完全に仮想化・自動化し、サービスとして提供する現在のパブリッククラウドに近い姿が描かれています。現在、AWSには200を超えるサービスがありますが、当初はAmazon S3とAmazon SQSの2つでした。その後、Amazon EC2やAmazon RDSがリリースされ、徐々に利用者が増えていきました。

現在では、Microsoft Azure やGoogle Cloud Platform、Alibaba Cloud、Oracle Cloud などの競合も台頭し、クラウド戦国時代ともいえる状況になっています。たとえば、マイクロソフト社は自社の強みでもあるソフトウェア技術をAzureクラウドに実装し、Office製品も含めてすべての機能をクラウドサービス化して企業ニーズに幅広く対応しようとしています。また、グーグル社はBigQueryに代表されるように、AIなどの先進的な分析に特化したサービスで差別化を図っています。

ただし、こうした競合によって、パブリッククラウドのサービス事業者がどこか1社に淘汰されるようなことは起こりそうにありません。むしろ、それぞれの特徴や特性を理解し、適材適所でサービスを使い分ける“マルチクラウド活用”が望ましいとも考えられます。

まずは、先駆者であるAWSを勉強しておくことで、パブリッククラウドならではの共通的な技術や設計の勘所を押さえることができます。AWS資格を取得したうえで、他のクラウドも利用しながらマルチクラウド活用のスキルを高めていくことが、より重要になるでしょう。



第2章

AWSにおけるセキュリティ設計

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

2-1

AWSにおけるセキュリティ設計の考え方

AWSなどのパブリッククラウドは、インターネットからアクセスして手軽に利用できる反面、不正アクセスや情報流出などのリスクがあるため、セキュリティに十分配慮して設計する必要があります。本節では、AWSにおけるセキュリティ設計の考え方について説明します。

1

AWS責任共有モデルによるセキュリティ方針

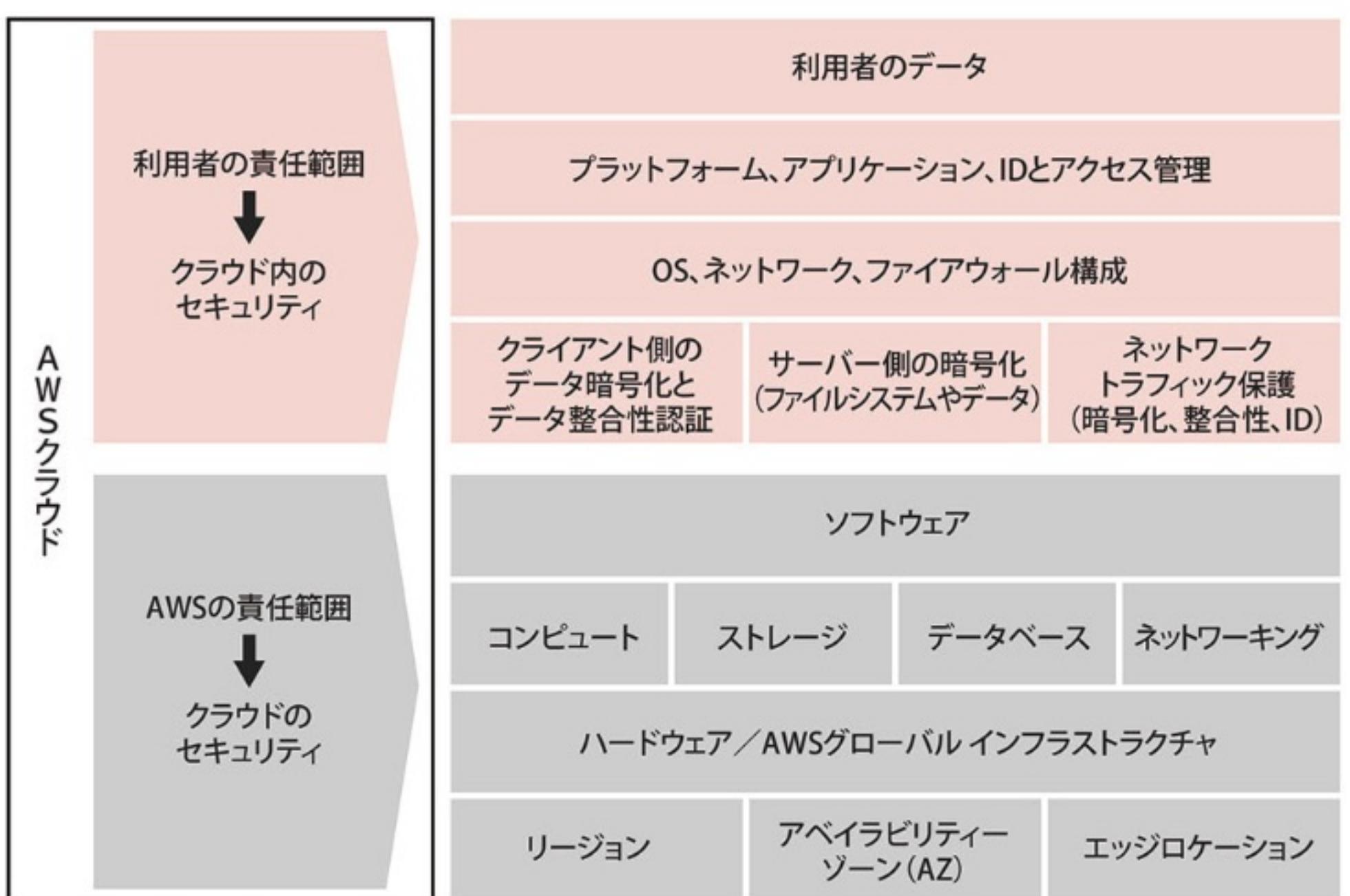
AWSに代表されるパブリッククラウドの利用を検討する際に、一番の懸念事項として取り上げられるのがクラウドの「セキュリティ」です。

たとえば、AWSは世界中に多くのデータセンターと物理リソースを持っていますが、それらがどのように配置・構成されているかなどシステムの詳細は原則として公開されていません。

「どのようにユーザーから見えない部分もあるため、「データをクラウドに置いて大丈夫か」など、さまざまな不安が生じます。

これに対してAWSでは、クラウド事業者であるAWSと利用者であるユーザーとがセキュリティ対策の責任境界を明確にすることで、分担・協力しながらセキュリティを強化していくという方針をとっています。この考え方は「**AWS責任共有モデル**」と呼ばれています。

【AWS責任共有モデルの概要図】



2

AWSのセキュリティ責任・対策

では、AWSが責任を持ってセキュリティを担保している部分はどこでしょうか。AWSは主に、データセンター・物理ハードウェア、ネットワーク・仮想化インフラストラクチャの管理・運用に責任を負っています。

以下に、セキュリティにおけるAWSの代表的な責任範囲を説明します。

●ハードウェア、AWSグローバルインフラストラクチャ

AWSは、世界中に多くのデータセンターと物理的なハードウェアリソースを配置しています。

データセンターに対しては、その場所を秘匿にしたり、監視カメラや侵入検知システムで施設への入退を厳密に管理・制御するなどのセキュリティ対策を実施しています。

また、装置や機器類のセキュリティにも配慮されており、たとえば、ストレージ装置を破棄する際にはデータが流出しないように規定の工程に従うなど、セキュリティを意識した運用が行われています。

●ネットワーキング

分散型サービス拒否(DDoS : Distributed Denial of Service)攻撃やIPスルーフィング(なりすまし)、パケット盗聴などの一般的なネットワークセキュリティ侵害に対しては、AWSが保護対策を実施しています。

●コンピュート

Amazon Elastic Compute Cloud(EC2)に代表されるコンピュートについては、仮想化インフラストラクチャで実現しています。仮想化を実現するハイパーバイザー(ホストOS)については、AWSがアップデートやパッチ管理、アクセス管理、ログ監査などのセキュリティ対策を講じて運用しています。

3

ユーザーのセキュリティ責任・対策

AWSを利用するユーザーは、基本的にOSから上位にあるレイヤーの管理・運用に責任を負う必要があります。

たとえば、OSおよびミドルウェアのパッチ適用やアカウント・権限管理、データ暗号化、仮想ネットワークのセキュリティ設定などが該当します。

本章では、AWS Well-Architectedフレームワークの「セキュリティ」でベストプラクティスとして定義されている以下の項目に沿って、以降の節からユーザーが担うべきセキュリティ対策のポイントを説明します。

●アイデンティティ管理とアクセス管理

必要最低限の権限をユーザーに付与する「**最小権限の原則**」の考え方について、AWSのサービスやリソースへの論理的アクセス制御をどのように実現するかを説明します。

ここでは、**AWS Identity and Access Management(IAM)**による権限管理について重点的に説明します。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。

●ネットワークセキュリティ

AWSの仮想ネットワーク上で構築されるサーバーやアプリケーションに対して、考慮すべきセキュリティ設計を説明します。

セキュリティを考慮したVPC(Virtual Private Cloud)のセグメンテーション、セキュリティグループやネットワークACL、Webアプリケーション保護によるネットワークの論理的なアクセス制御について重点的に説明します。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。

●データの保護

ユーザーが保持するさまざまなデータを、AWSでどのように保護するかを説明します。

Amazon Simple Storage Service(S3)などのAWSのサービス側で暗号化するServer Side Encryption(SSE)、クライアント側で暗号化するClient Side Encryption(CSE)、AWS上で暗号化鍵を管理するAmazon Key Management Service(KMS)など、ユーザーのさまざまなデータセキュリティ要件に応じた対応方法について重点的に説明します。詳細は、「2-4 データの保護」を参照してください。

●セキュリティ監視

AWSのサービスやリソースの構成変更、操作ログなどの把握・追跡、セキュリティに関連するログの収集・監視方法などを説明します。

Amazon CloudWatchやAWS CloudTrailによるログ収集・監視、AWS Configによるリソース構成の追跡、AWS Trusted Advisorによるセキュリティチェックなどを重点的に説明します。詳細は、「2-5 セキュリティ監視」を参照してください。



演習問題

1 AWS責任共有モデルにおいて、セキュリティに関するAWSの責任範囲は次のうちどれですか。

- A 物理ハードウェアの管理
- B 利用者のデータ管理
- C IDや権限などのアクセス管理
- D OSのパッチ管理



解答

1 A

AWSでは、クラウド事業者であるAWSと利用者であるユーザーとがセキュリティ対策の責任境界を「AWS責任共有モデル」という考え方で明確化しています。

たとえば、クラウド事業者であるAWSの責任範囲として、データセンター・物理ハードウェアの管理があります。したがって、Aが正解です。

2-2

アイデンティティ管理とアクセス管理

AWSでは、AWSを利用するユーザーの管理や認証、サービスやリソースへのアクセス制御を「AWS Identity and Access Management (IAM)」と呼ばれるサービスで管理します。本節では、IAMによるアクセス制御の概要を説明します。

1

AWSにおけるアカウント

AWSを利用するには、最初にAWSアカウントを作成する必要があります。アカウント作成時に指定したメールアドレスとパスワードを使用することで、AWSを管理する画面(AWSマネジメントコンソール)にログインできるようになります。

最初に作成するAWSアカウントは「ルートユーザー」とも呼ばれ、アカウント内のすべてのAWSのサービスやリソースに対するフルアクセス権限を持っています。たとえば、以下のような操作はルートユーザーが持つルート権限が必要となります。

- ・ AWSアカウントのメールアドレスやパスワードの変更
- ・ IAMユーザーの課金情報に対するアクセス許可・拒否、など

このルートユーザーの情報が乗っ取りや盗難で第三者に漏洩すると、すべての権限が奪われてしまいます。したがって、次の2つのセキュリティ対策を実施することが重要です。

- ・ ルートユーザーに対して、強度の高いパスワードや、ログイン時の多要素認証(MFA : Multi-Factor Authentication)を設定する
- ・ AWSの通常利用時には原則としてルートユーザーは使用せず、IAMユーザーとIAMグループを作成し、適切な権限(ポリシー)を与えて利用する



ルートユーザーの通常利用を避けるだけでなく、強度の高いパスワード設定やアカウント自体へのログインを多要素認証にするなどのセキュリティ強化策が重要です。

2 IAMユーザーとIAMグループの概要

AWSのユーザーに対しては、AWSアカウントのルートユーザーの権限で**IAMユーザー**や**IAMグループ**を作成し、適切な権限を**IAMポリシー**として付与することで、適切なアクセス制御の下で利用を許可します。以下に、それぞれの特徴を説明します。

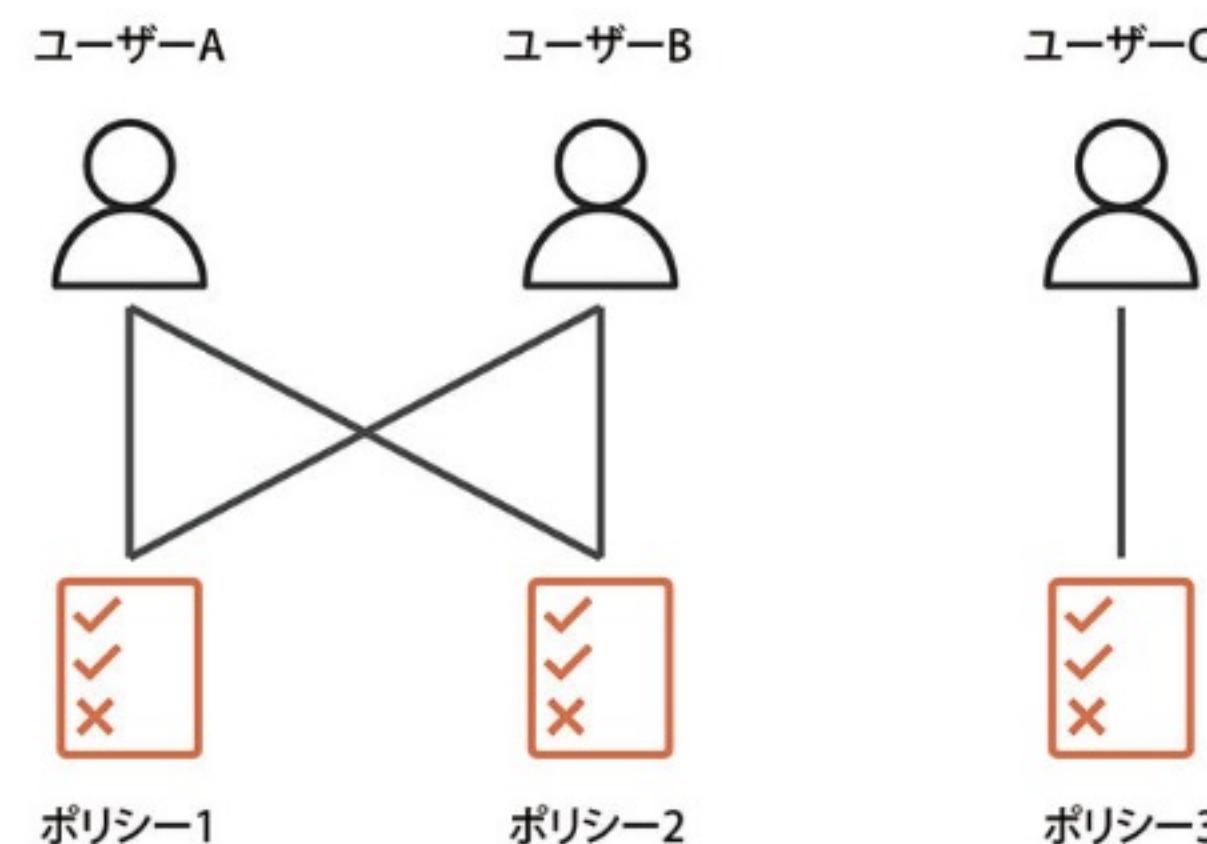
IAMユーザー

AWSを操作するユーザーをIAMサービスから作成します。1つのAWSアカウントで5,000ユーザーまで作成できます。

ユーザー名とパスワードを使用することで、AWSマネジメントコンソールの画面にログインできます。

IAMユーザーには、後述するIAMポリシーで権限を直接割り当てることもできますが、IAMグループに所属させてグループに対して権限を付与すると、管理しやすくなります。

IAMユーザーに権限を直接付与



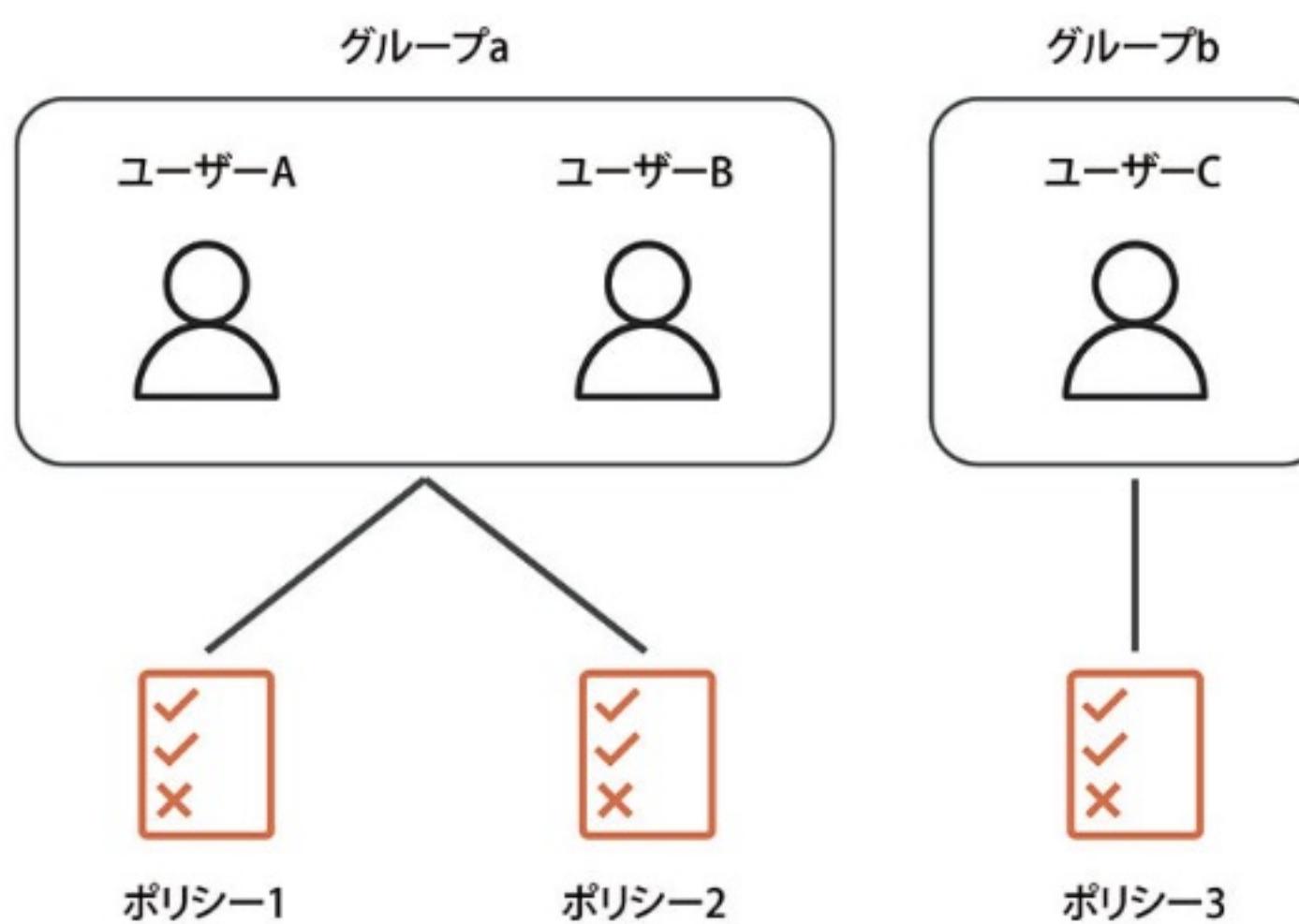
IAMグループ

IAMユーザーを束ねるグループを作成することで、ユーザーや権限などの管理を統合的に行うことができます。

1つのAWSアカウントで最大300グループを作成でき、IAMユーザーあたり10までのグループに所属することができます。

IAMグループに対してIAMポリシーを割り当てることで、権限管理が簡素化されます。

IAMグループにIAMポリシーを付与



IAMユーザーあたりのグループ所属数には上限があることに注意しましょう。

3 IAMポリシーによるアクセス権限管理

新規に作成したIAMユーザーやIAMグループには、最初は何も権限が付与されていません。このため、最低限必要となる権限のみを、IAMポリシーを利用してIAMユーザーやIAMグループに付与します。

この手法のように、最低限必要な権限のみを付与することでアクセスの範囲を制限してセキュリティを高めることを、AWS Well-Architectedフレームワークでは「**最小権限の原則**」と呼んでいます。

IAMポリシーの記述方法

IAMポリシーは、どのユーザーやグループが、どのAWSサービスまたはリソースに対して、どの操作を許可または拒否するかをJSON(JavaScript Object Notation)形式で記述します。

例 IAMポリシーのJSON記述例(Amazon S3の操作を、ある特定のアクセス元IPアドレスからのみに制限する)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "arn:aws:s3:::test-bucket/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "11.22.33.44/32"
          ]
        }
      }
    }
  ]
}
```

←対象となるAWS操作を指定
←許可の設定ならば"Allow"、拒否の設定ならば"Deny"
←対象となるAWSリソースを指定
←このアクセス制御が有効になる条件の設定

この例の場合、アクセス元IPアドレスが11.22.33.44ならば、S3のtest-bucketというバケット内のオブジェクト例に対してGetObjectの操作を許可する。

例 IAMポリシーのJSON記述例(ある特定のIPアドレスから、Amazon EC2のある特定リージョンのEC2インスタンスの削除を許可)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "10.10.10.0/24"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "ec2:Region": "ap-northeast-1"
        }
      }
    }
  ]
}
```

この例の場合、1つ目のポリシーで、10.10.10.0/24からのEC2インスタンスの削除操作のみ許可し、2つ目のポリシーでap-northeast-1リージョン以外に対するEC2の操作をすべて拒否する。

前述したとおり、最小権限の原則に則り、デフォルト設定ではユーザーは何も許可されていません。したがって、IAMポリシーで明示的に許可を設定しな

い限り、すべての操作が拒否されます。これを「**暗示的な拒否**」と呼びます。

一方で、IAMポリシーで許可や拒否の権限を付与するには、IAMポリシーに明示的に記述します。これを「**明示的な許可**」あるいは「**明示的な拒否**」と呼びます。

権限の優先度としては、優先度の高いものから「明示的な拒否」「明示的な許可」「暗示的な拒否」の順となります。



試験対策

IAMポリシーでは、権限の優先度は以下のようになります。重要ですので、覚えておきましょう。
明示的な拒否 > 明示的な許可 > 暗示的な拒否



試験対策

JSONで定義されているIAMポリシーの内容から、どのような操作に対するアクセスが許可、あるいは拒否されているかを判断できるようにしておきましょう。



最小権限の原則は、AWSに限らず一般的にITシステム設計・構築時に考慮すべきセキュリティ対策における重要な考え方です。

●IAMポリシーの種類

IAMポリシーには、以下の3種類の設定方法が提供されています。

- ・ AWS管理ポリシー
- ・ カスタマー管理ポリシー
- ・ インラインポリシー

AWS管理ポリシーは、AWSが管理する事前に定義されたポリシーのテンプレート群です。

一般的なユースケースに基づいたポリシーが用意されており、ユーザーは条件に合うポリシーを選択するだけで利用できます。たとえば、代表的なポリシーとして**AdministratorAccess**と**PowerUserAccess**が用意されています。

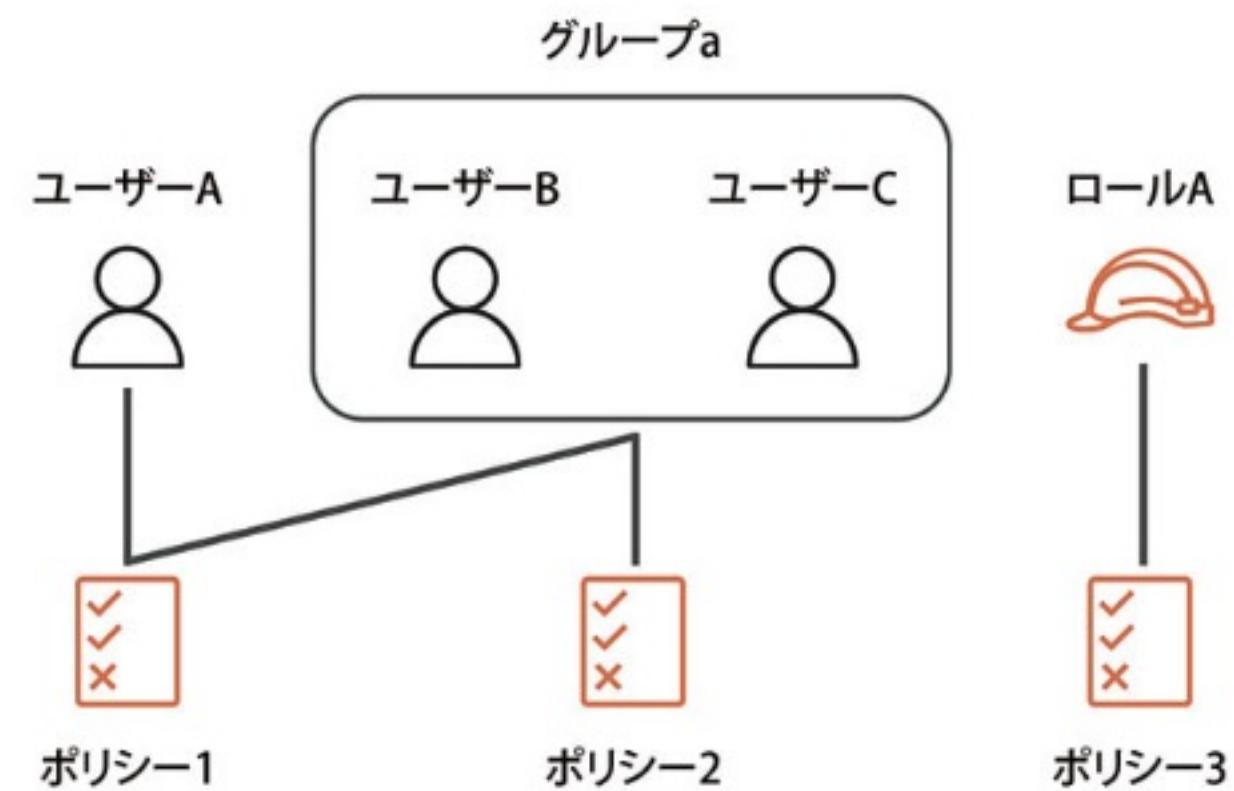
【代表的なAWS管理ポリシー】

AdministratorAccess	PowerUserAccess
<ul style="list-style-type: none"> ・すべてのアクセス権限を提供するポリシー ・AWSアカウントを利用しない代わりに、管理者相当のIAMユーザー或はIAMグループに付与する 	<ul style="list-style-type: none"> ・IAMサービスを除くその他すべてのアクセス権限を提供するポリシー ・通常業務ではユーザー管理やポリシー管理を必要としないIAMユーザー(例: 開発者など)やIAMグループに付与する

カスタマー管理ポリシーは、AWSのユーザーが自身で作成・カスタマイズできるポリシーです。

標準提供されているAWS管理ポリシーではセキュリティ要件を満たせない場合に利用します。

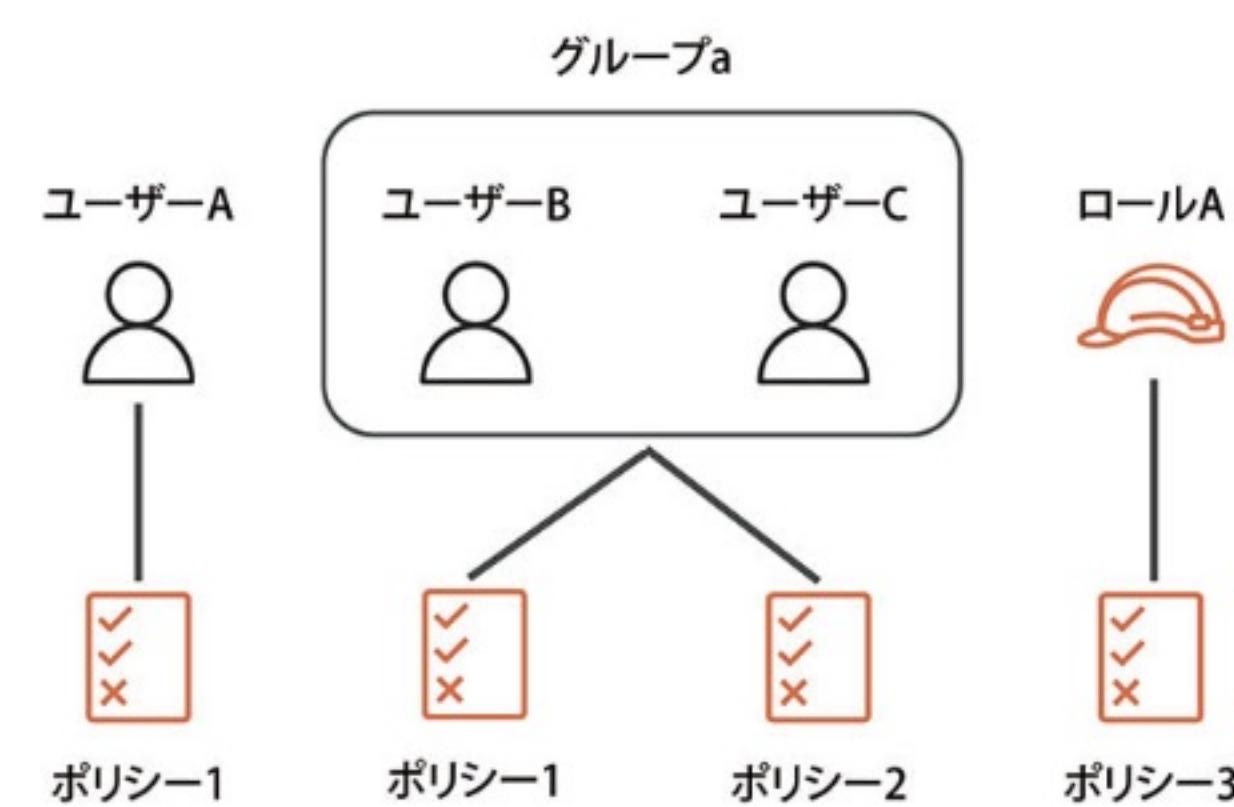
【カスタマー管理ポリシー】



インラインポリシーでは、IAMユーザー或はIAMグループ、IAMロールに埋め込まれているポリシーに対し、ポリシーを1つ設定できます。

前述した2つの管理ポリシー(AWS管理ポリシーとカスタマー管理ポリシー)は複数のIAMユーザー或はIAMグループ間で共有できますが、インラインポリシーは個別に埋め込まれているため共有できません。

【オンラインポリシー】



●Amazon S3におけるアクセス制御(バケットポリシー)

Amazon S3には、4つのアクセス制御があることを「1-5 ストレージサービス」で説明しました。ここでは、**バケットポリシー**によるアクセス制御の例を示します。バケットポリシーもIAMポリシーと同様に、JSON形式で記述します。

例 特定のIAMユーザーのみS3の操作が可能

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": "arn:aws:iam::111122223333:user/userA"
      },
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::test-bucket"
    }
  ]
}
```

AWSアカウント(111122223333)の IAMユーザー(userA)以外の操作を拒否

S3バケット(test-bucket)に対するすべてのS3操作

この例の場合、IAMユーザー「userA」以外のユーザーに対して、Amazon S3バケットの操作が拒否される(AWSマネジメントコンソールからAmazon S3を操作する場合を想定したポリシー記述例)。



AWS管理ポリシーで提供されている代表的なポリシーは重要です。
覚えておきましょう。

4

IAMによる認証方式

ここまで、IAMユーザーとIAMグループに対して、最低限必要なアクセス権限をIAMポリシーで付与する方法を説明しました。IAMユーザーから実際にAWSサービスの操作を行うためには、**認証情報**も必要です。

●IAMユーザー名とパスワード

各種のAWSサービスを操作する場合、たとえば「EC2インスタンスを起動する」といった操作を実行するには、WebブラウザからAWSマネジメントコンソールを利用します。

このAWSマネジメントコンソールを利用する際には、**IAMユーザー名とパスワード**の認証情報が使用されます。



通常は、「強度の高いパスワードのみを許可する」「パスワードの有効期限を設定する」などのパスワードポリシーを設定することで、ログイン時のセキュリティを強化します。

●多要素認証(MFA : Multi-Factor Authentication)

パスワード流出による“なりすまし”を防ぐため、通常は**多要素認証(MFA)**により認証時のセキュリティを強化します。

たとえば、IAMユーザー名とパスワードに加え、Google Authenticatorなどで生成されるワンタイムパスワードをもう1つの認証情報として利用します。

●アクセスキーIDとシークレットアクセスキー

IAMユーザーは**アクセスキーID**と**シークレットアクセスキー**のペアを作成することができます。

これらの認証情報は、コマンドラインで利用するAWS Command Line

Interface(AWS CLI)、またはプログラムからAPI経由で利用するAWS SDKの認証情報として使うことができます。

たとえば、ある自作のプログラムから「EC2インスタンスを起動する」といった操作を記述したい場合は、プログラムのソースコードにアクセスキーIDとシークレットアクセスキーを指定する必要があります。

ただし、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、流出によるリスクもあるため現在は推奨されていません。たとえば、プログラムのソースコードにキーを直接記述していると、ソースコード流出に伴うキー流出のリスクがあるため非常に危険です。したがって、現在では次に説明するIAMロールによる認証が推奨されています。

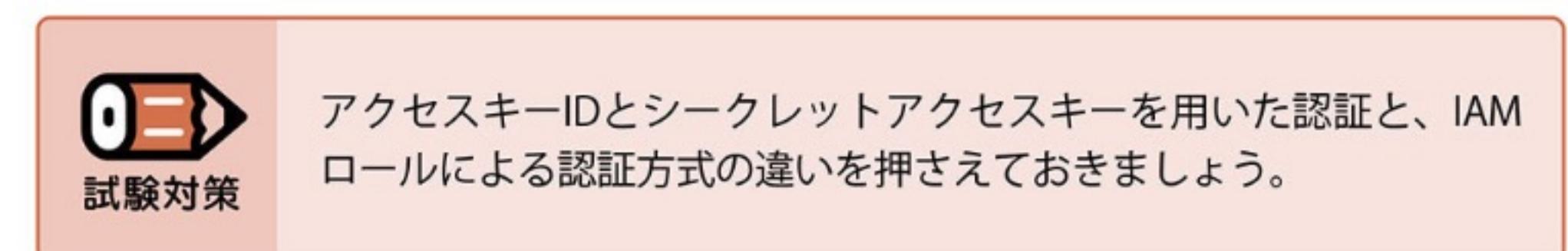
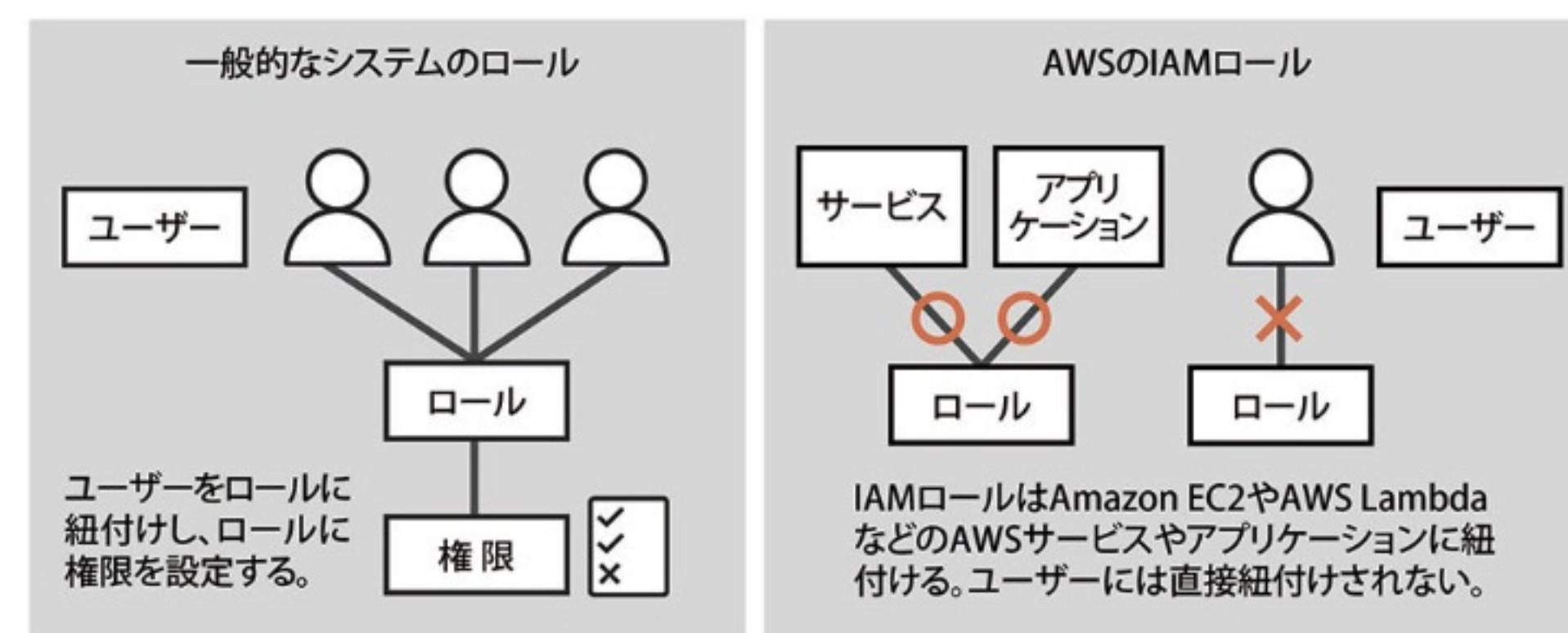
● IAMロール

IAMロールは、AWSサービスやアプリケーションにAWSの操作権限を付与する仕組みです。

IAMユーザーとIAMグループとは紐付けせず、**IAMロールに対してIAMポリシーを直接付与**することで権限を管理します。

たとえば、EC2インスタンスにIAMロールを直接割り当てることで、その上で稼働するアプリケーションから、AWS SDK経由でアクセスキーIDとシークレットアクセスキーなしで、IAMロールで許可されている操作(Amazon EC2やAWS LambdaからAmazon S3やAmazon DynamoDBへの書き込みなど)を行うことができます。

【ロール利用時のイメージ】



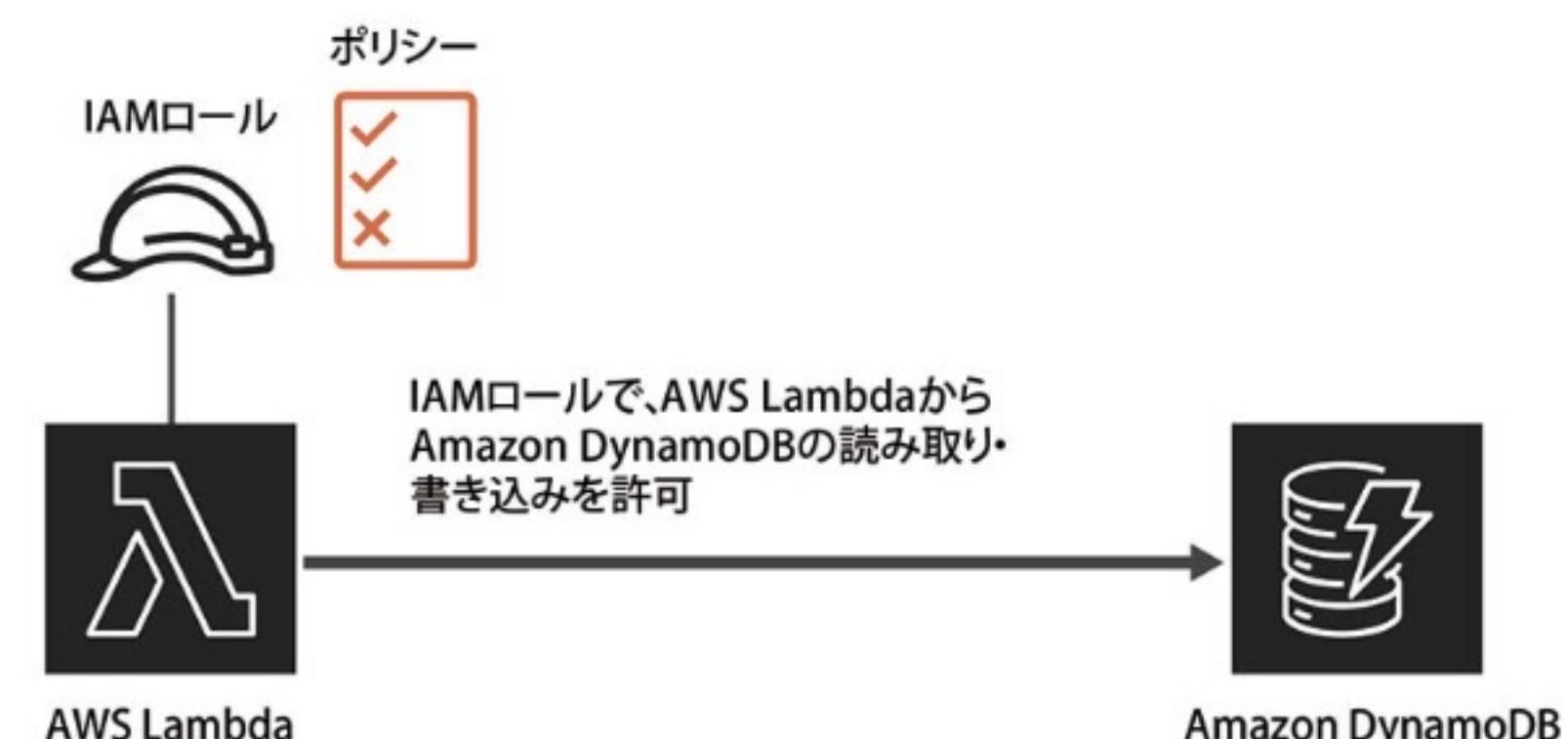
次の例では、IAMロールをEC2インスタンスに割り当て、Amazon EC2からAmazon S3上のバケットやファイルへの読み取り操作を許可しています。

【IAMロールの割り当てと操作の許可の例(1)】



次の例では、IAMロールをAWS Lambdaに割り当て、LambdaからAmazon DynamoDBへの読み取りや書き込みを許可しています。

【IAMロールの割り当てと操作の許可の例(2)】



5

IAMによるIDフェデレーション

IDフェデレーションとは、企業や組織などすでに導入されている認証の仕組みとAWSの認証を紐付けし、**シングルサインオン**を実現する機能です。

この機能により、全従業員のユーザー認証情報をAWSのIAMに登録する必要がなくなり、すでに導入されているLDAP^{*1}などの認証の仕組みと連携して、AWSサービスを簡単に利用できます。

たとえば、企業で利用中のLDAP認証基盤で認証された社員に対しては、AWSのAmazon S3へのアップロードも許可するといった認証連携が可能になります。

また、GoogleやFacebookなどのSNS(ソーシャルネットワーキングサービス)で使われている認証情報と連携することもできます。

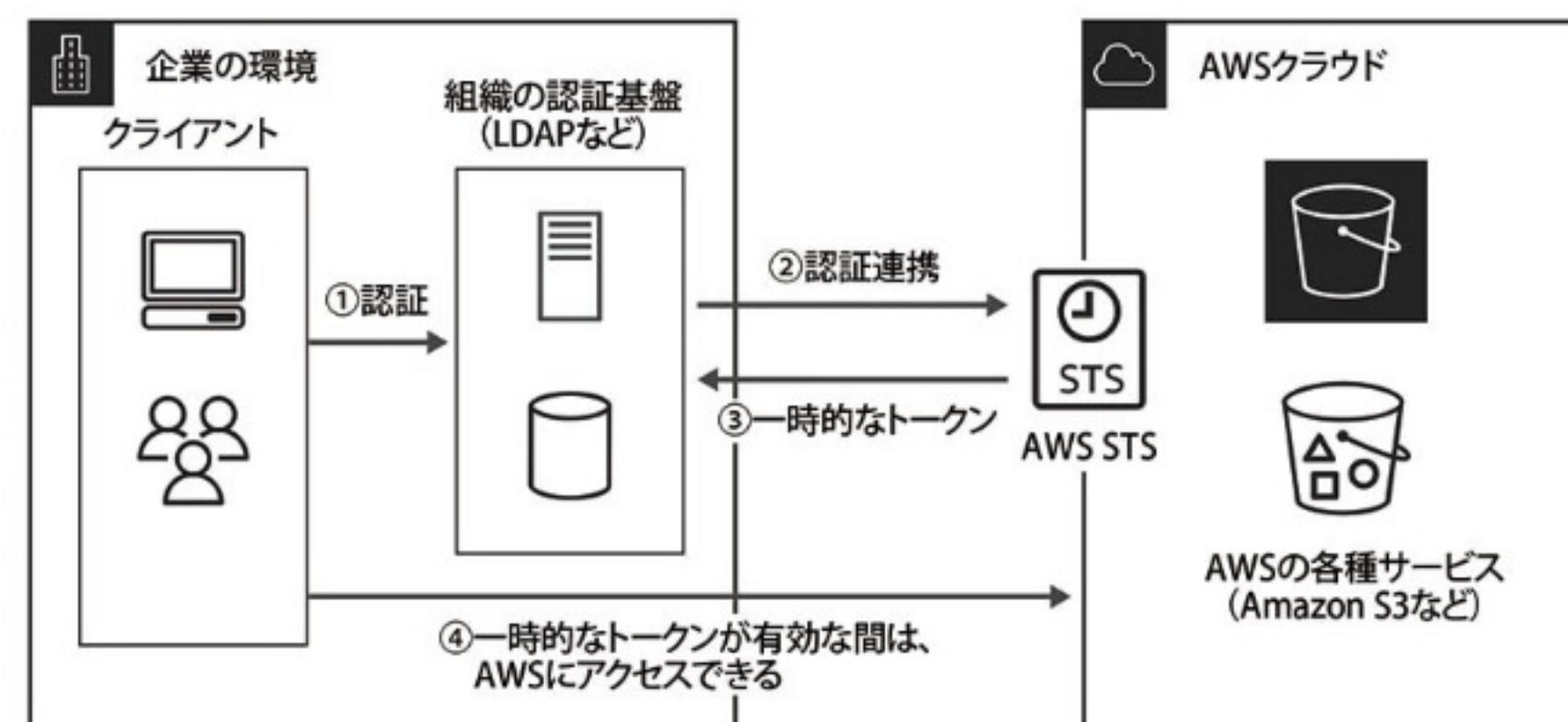
●SAML 2.0を使用したIDフェデレーション

IAMでは、**SAML(Security Assertion Markup Language)2.0**や**OpenID Connect**と互換性のある認証プロトコルをサポートしています。

SAML 2.0やOpenID Connectは、IDフェデレーションによるシングルサインオンを簡単に実現するためのプロトコルです。認証したユーザーごとに一時的なアクセキー(Temporary Security Token)を発行することで認証連携を実現します。

AWSへの一時的なアクセキーは、AWS Security Token Service(STS)というサービスで管理・発行されます。

【SAML 2.0による企業の認証基盤と連携したIDフェデレーションの概念フロー】



*1 【LDAP】Lightweight Directory Access Protocol：ユーザーの認証情報などを格納したディレクトリデータベースへアクセスするためのプロトコル。

●Web IDフェデレーションによるSNSとの連携

GoogleやFacebookなど、OpenID ConnectをサポートしているSNSをIDフェデレーションで連携させると、AWSにシングルサインオンすることができます。これを「Web IDフェデレーション」と呼びます。

なお、Webアプリケーションやモバイルアプリから認証する場合は、**Amazon Cognito**というサービスを利用することで、各SNSとの認証連携が簡単に実現できます。

●Directory ServiceによるMicrosoft ADとの認証連携

AWS Directory Serviceは、AWSのクラウド内で管理されるマネージド型のMicrosoft AD^{*2}です。

主に以下の3つのディレクトリタイプを提供します。

- Microsoft AD
- Simple AD
- AD Connector

Microsoft ADでは、AWS上でActive Directoryサービスを利用できます。

Simple ADでは、AWS上でActive Directory互換のSambaサービスを利用できます。

AD Connectorは、既存のActive Directory環境へ接続するためのプロキシサービスです。たとえば、企業や組織内の認証基盤としてすでにMicrosoft ADを使用している場合は、Directory Serviceが提供するAD Connectorを利用することで、AWSで必要となる認証を既存のActive Directoryに接続して連携することができます。



Directory ServiceのAD Connectorを利用してことで、企業内にある既存のActive Directoryとの認証連携を行うことができます。

*2 【Microsoft AD】Microsoft Active Directory：マイクロソフト社が開発したディレクトリサービスで、主にユーザー情報など企業内のさまざまな情報を管理する。

6

AWSアカウントのサービスアクセス管理

ここまで、AWSアカウント内のルートユーザーやIAMユーザーの管理・アクセス権限について説明してきました。ここからは、AWSアカウント自体が複数ある場合のアカウント管理方法とアクセス制限について説明します。

● AWS Organizations

1つの組織内で複数のAWSアカウントを保持するケースがあります。たとえば、提供するビジネスのシステム・サービスごと、あるいは保持する環境ごと(開発環境やテスト環境、本番環境)など、用途ごとに複数のAWSアカウントを管理しなければならない場合です。

そのような場合、通常であればAWSアカウントごとに権限設定やコスト管理を行いますが、アカウント数が増えてくると管理が煩雑かつ複雑になります。したがって、複数のAWSアカウントに対して統合的なアカウント管理を行うことは効率化の観点で非常に重要です。

AWSでは、**AWS Organizations**というサービスが提供されています。Organizationsを利用することで、複数のAWSアカウント作成の自動化やグループ化による集中管理、またそれらのグループにポリシーを適用したアクセス管理が実現できます。

さらに、複数のAWSアカウントで発生した使用料を一括請求にすることができます(詳細は「5-4 コストの管理」を参照)。

● サービスコントロールポリシー(SCP)

Organizationsでは、「**サービスコントロールポリシー(SCP)**」という重要な機能が提供されています。SCPは、Organizationsで管理されている複数のAWSアカウントに対して、IAMポリシーのような権限制御を統合的に管理・適用する機能です。

たとえば、複数のAWSアカウントに対して、あるAWSサービスの使用を禁止したい場合があるとします。SCPを利用すれば、複数のAWSアカウントに対して、特定のAWSサービスの操作を禁止する共通のポリシーを設定することができます。これにより、複数のAWSアカウントに対する統合的なセキュリティやガバナンスの強化を図ることができます。



Organizationsのサービスコントロールポリシー(SCP)の特徴を押さえておきましょう。



演習問題

1

EC2インスタンス上のアプリケーションから、アクセスキーIDとシークレットアクセスキーを使わず安全にほかのAWSサービス(Amazon S3など)にアクセスしたいと考えています。次のうち、適切な方法はどれですか。

- A Amazon S3にアクセスキーIDとシークレットアクセスキーを保存する
- B IAMロールをEC2インスタンスに割り当てる
- C IAMユーザーをEC2インスタンスに割り当てる
- D 多要素認証(MFA)を使う
- E IAMポリシーでIAMユーザーのパスワードを複雑に設定する

2

あなたの会社では、Microsoft Active Directory ドメインコントローラを運用しています。このActive DirectoryをAWSへ移行し、運用負荷を軽減することを検討しています。移行先のサービスとして適切な項目はどれですか。

- A Amazon Elastic Compute Cloud(EC2)
- B AWS Identity and Access Management(IAM)
- C Amazon Cognito
- D AWS Directory Service

3 複数のAWSアカウントで利用するAWSサービスのうち、使用できるサービスを制限したいと考えています。次のうち、どのサービスを使用して制限すればよいでしょうか。

- A AWS Organizations
- B AWS WAF
- C AWS Directory Service
- D IAMロール

A 解答

1 B

EC2インスタンスにIAMロールを直接割り当てることで、その上で稼働するアプリケーションからAWS SDK経由でアクセスキーIDとシークレットアクセスキーなしで、IAMロールで許可されている操作(Amazon EC2やAWS LambdaからAmazon S3やAmazon DynamoDBへの書き込みなど)を行うことができます。

2 D

AWS Directory Serviceは、AWS上でActive Directoryを利用するサービスです。マネージドサービスとして提供されているため、サーバーの構築や運用負荷が軽減できます。D以外の選択肢は、移行先のサービスとして適切ではありません。

3 A

AWS Organizationsでは、複数のAWSアカウントを一元管理することができます。複数のAWSアカウントで利用できるサービスを制限するには、AWS Organizationsが提供する「サービスコントロールポリシー(SCP)」という機能を利用します。

2-3

ネットワークセキュリティ

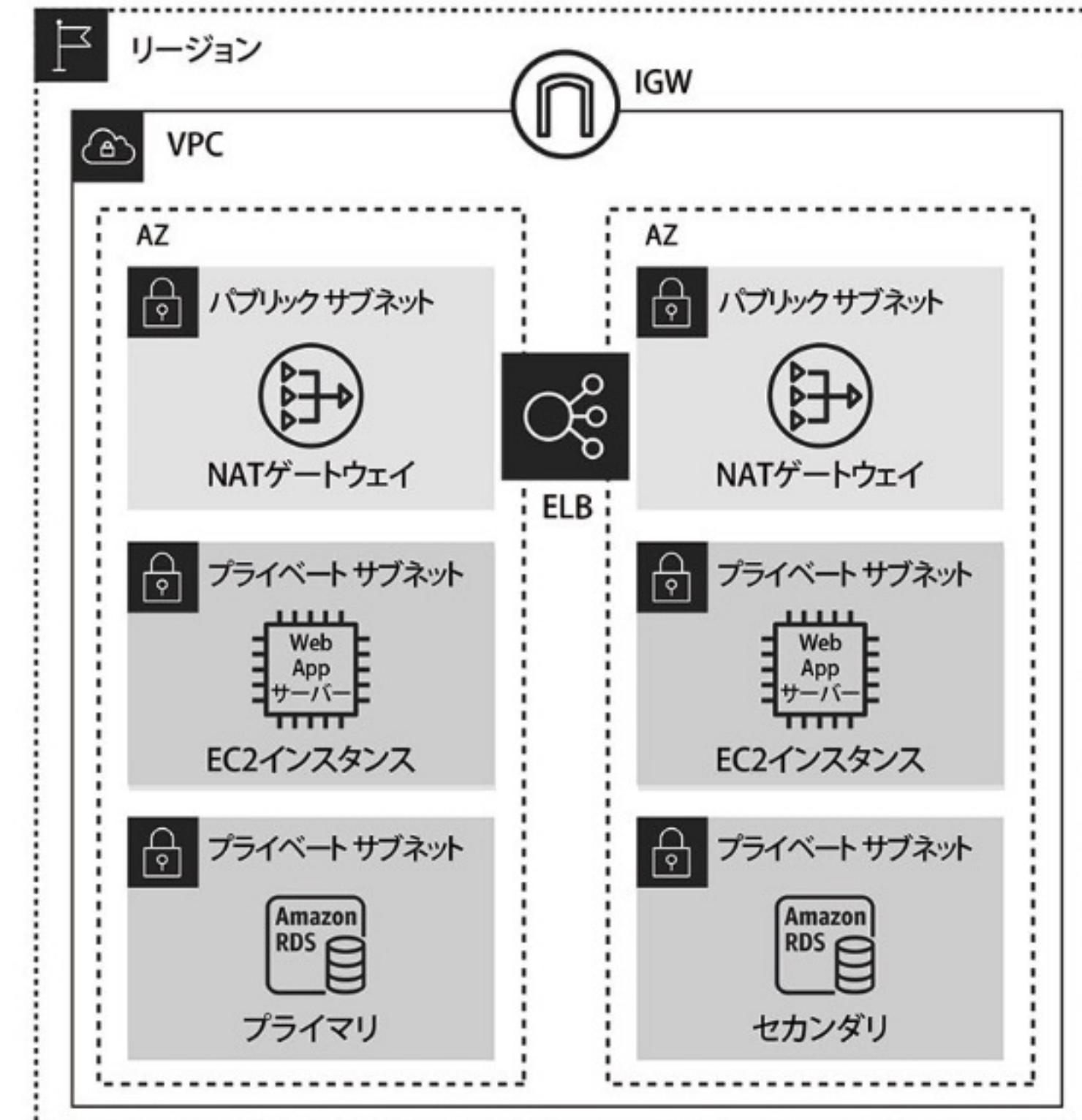
AWSではVirtual Private Cloud(VPC)により仮想的なプライベートネットワークを構築できます。本節では、VPCにおけるネットワークセキュリティの考え方を説明します。

1

VPCによるネットワーク構成

VPCにより、AWS上でプライベートな仮想ネットワークを構築することができます。VPCをリージョン内に構築することで、Amazon EC2やAmazon RDSなどのさまざまなAWSサービスを配置できます。WebアプリケーションをAWS上で構築する場合の代表的なVPCネットワークの構成例を、次の図に示します。

【WebアプリケーションのVPC構成例】



前ページで示した図【WebアプリケーションのVPC構成例】の主な特徴は以下のとおりです。

● NAT ゲートウェイ

ネットワークアドレス変換(NAT)を提供するゲートウェイサービス。パブリックサブネットに配置し、プライベートサブネットに配置されたサーバーからインターネットにアクセスできるようにする

● Elastic Load Balancing (ELB)

インターネットからのリクエストを、複数のWebアプリケーションサーバー(EC2インスタンス)に振り分けるサービス

● Web アプリケーションサーバーと RDS

- インターネットから直接アクセスされないよう、プライベートサブネットに配置
- EC2インスタンス上にWebアプリケーションサーバーを構築
- データベースインスタンスは正(プライマリ)と副(セカンダリ)の冗長構成でデータを格納

このような一般的なVPC構成に基づいて、ユーザーが設定する必要があるネットワークセキュリティについて、次項から説明します。

2

セキュリティグループとネットワークACLによるアクセス制御

VPCでは、標準で次の2つのネットワークセキュリティ機能が提供されています。

●セキュリティグループ

セキュリティグループは、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。

VPCに配置したEC2インスタンスから出入りするトラフィックを制御します。たとえば、「EC2インスタンスへのSSH(TCPポート22番)アクセスを許可する」といった制御が可能です。

主な特徴は以下のとおりです。

- デフォルトではEC2インスタンスから発信する(アウトバウンド)通信はすべて許可、受信する(インバウンド)通信はすべて拒否
- インバウンドかアウトバウンドかの区別、プロトコル(TCPやUDP)、ポート範囲(80番など)、IPアドレスなどの項目で、許可するルールのみを最大60まで定義できる
- 複数のセキュリティグループ(複数のルールの集合体)をEC2インスタンスに適用できる
- 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は戻りの通信も自動的に許可される)

【セキュリティグループによるファイアウォール機能のイメージ】



セキュリティグループのインバウンド通信のルール設定例

タイプ	プロトコル	ポート範囲	ソース	説明
SSH (22)	TCP	22	192.168.1.0/24	192.168.1.0/24から22番ポートへのアクセスを許可
HTTP (80)	TCP	80	0.0.0.0/0	すべての送信元IPから80番ポートへのアクセスを許可



セキュリティグループの特徴を押さえておきましょう。特に、設定が即時反映される点、またステートフルな制御が可能である点は重要です。



実際のセキュリティグループの適用では、インバウンド通信は最低限必要な通信のみを許可し、アウトバウンド通信は特別な要件がない限りはすべて許可しておくケースが一般的です。

●ネットワークACL

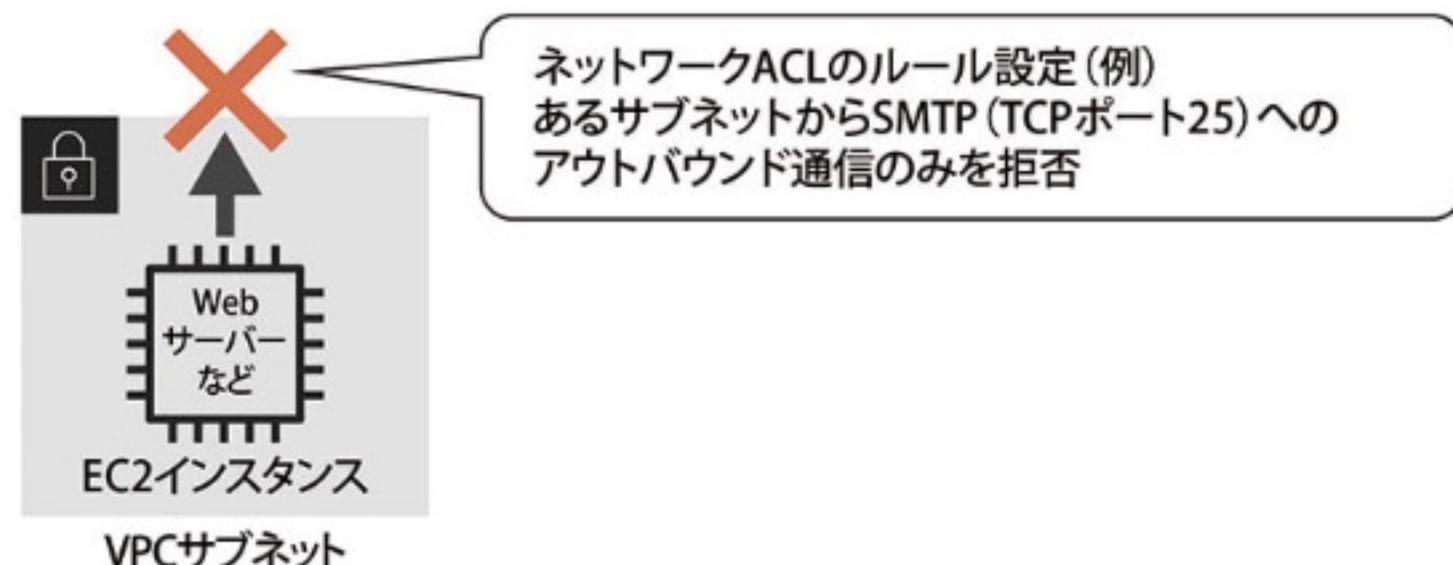
VPC内に構成されたサブネットに対するファイアウォール機能です。

サブネットを出入りするトラフィックを制御します。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、主にサブネットを横断する通信を制御する場合に利用します。

主な特徴は以下のとおりです。

- ・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を設定できる
- ・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する
- ・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ではすべてのトラフィックを拒否する
- ・インバウンドとアウトバウンドのそれぞれに対して、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)

【ネットワークACLによるサブネットでの通信制御のイメージ】



ネットワークACLによるアウトバウンド通信のルール設定例

ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可・拒否
90	SMTP (25)	TCP	25	0.0.0.0/0	拒否
100	すべてのトラフィック	すべて	すべて	0.0.0.0/0	許可
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	拒否



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ことが重要です。



ネットワークACLのデフォルト状態では、

- ・ルール番号100で「すべてのトラフィックを許可」
- ・ルール番号*(最後の行)で「この行より上に記載したルールに一致しないすべての通信を拒否」となります。

ルールは番号順に適用されるため、明示的な拒否の場合は100より小さな数字を指定します。

●セキュリティグループとネットワークACLの両方を適用している場合の注意点

セキュリティグループとネットワークACLの両方に通信ルールが適用されている場合は、**両方のルールで許可されないと拒否になる**ため注意が必要です。

たとえば、セキュリティグループでSMTPのアウトバウンド通信を許可、ネットワークACLでアウトバウンドのSMTPを拒否の場合は、結果的に通信は拒否となります。なお、セキュリティグループとネットワークACLの主な違いについては、「1-2 ネットワークサービス」でも説明しています。

【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 ^{※3} 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



セキュリティグループとネットワークACLの両方で許可されていない場合は、すべて拒否になります。

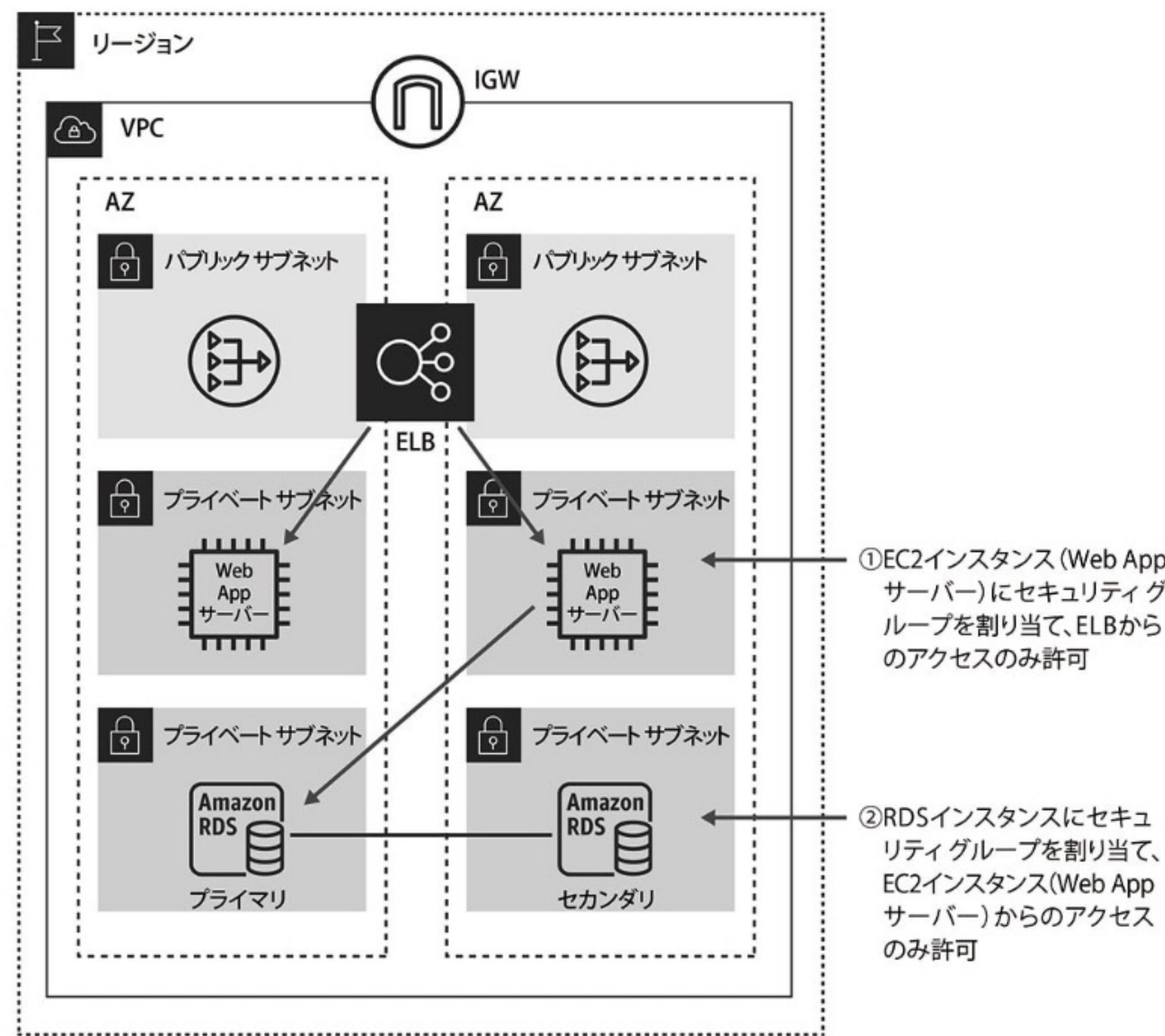
※3 【ENI】Elastic Network Interface : VPC内のネットワークインターフェイスを提供するサービス。利用可能なAWSサービスにアタッチすることでENIに紐付いたIPアドレスを利用できる。

3

VPC内のネットワークセキュリティの設定例

前述したセキュリティグループとネットワークACLをどのように使い分けるかを、一般的なVPCの構成例に基づいて説明します。

【一般的なVPC構成に対するセキュリティグループの適用例】

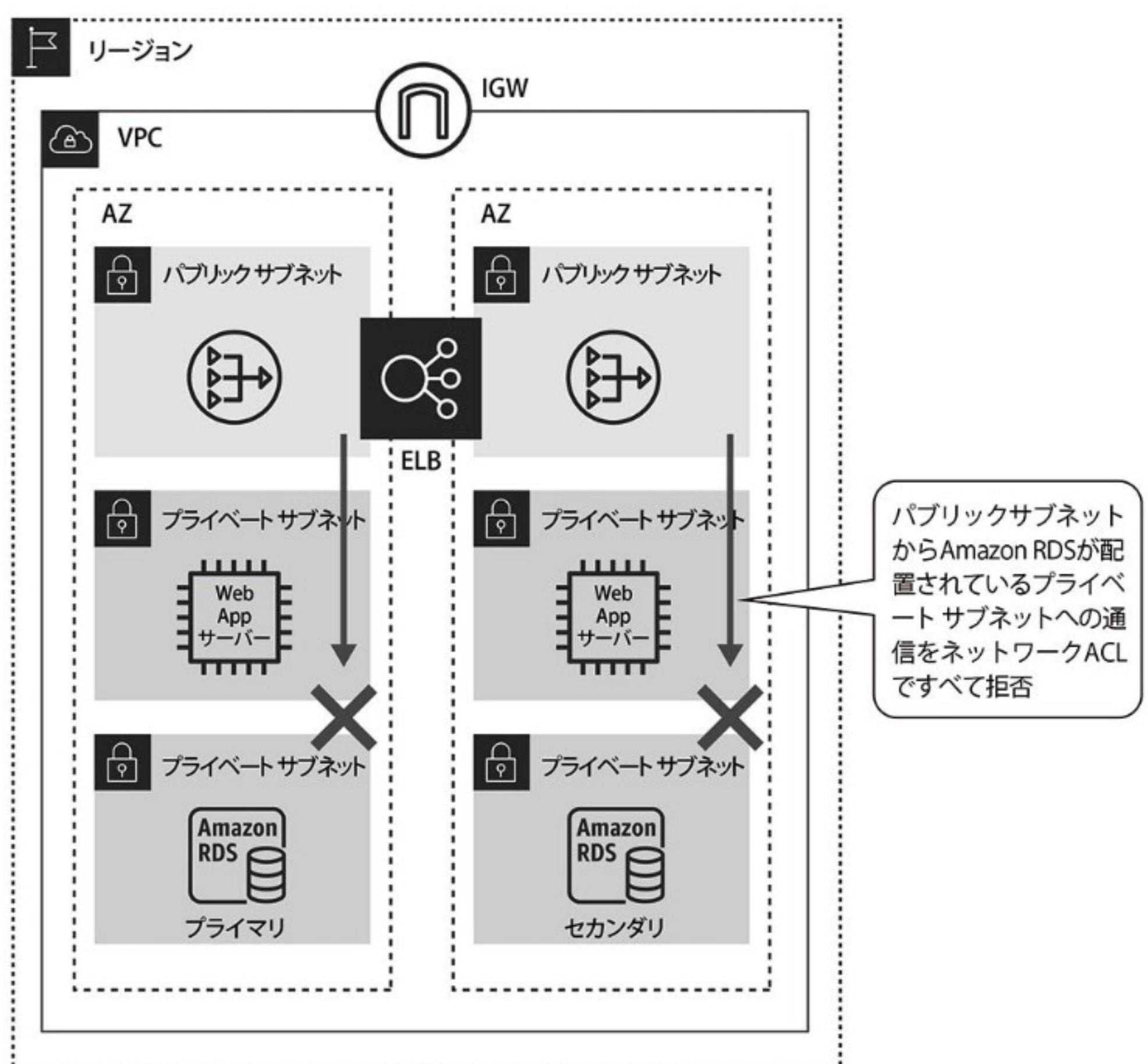


この図のセキュリティグループの適用ポイントは以下のとおりです。

- 1 Webアプリケーションサーバーが配置されたEC2インスタンスへのインバウンド通信は、ロードバランサーであるELBからのHTTP通信(ポート80番)のみを許可することで、外部からWebアプリケーションサーバーへの不必要的アクセス(たとえば、SSHなど)を制限する

- 2 データベースサーバーであるAmazon RDS上のデータベースインスタンスには、Webアプリケーションサーバー群からの通信をすべて許可することで、外部からの通信などデータベースサーバー自体への不必要的アクセスを制限する

【ネットワークACLの適用例】



上図に示した【ネットワークACLの適用例】の例では、パブリックサブネットからAmazon RDS(データベースサーバー)が配置されているプライベートサブネットへの直接的な通信は、すべて拒否されます。

たとえば、悪意のある第三者がインターネット上の無防備なパブリックサブネットに侵入した場合などで、データベースサーバーからのデータ流出などを防ぐため、データベースサーバーが配置されているプライベートサブネットへのインバウンド通信を明示的に禁止するケースなどが想定されます。

4

踏み台サーバーによるセキュアなリモートアクセス制御

VPC内のEC2インスタンスに、リモートからセキュアにアクセスする方法として、**踏み台サーバー**を利用する方法があります。踏み台サーバーは、英語で「Bastion(砦)」とも呼ばれています。

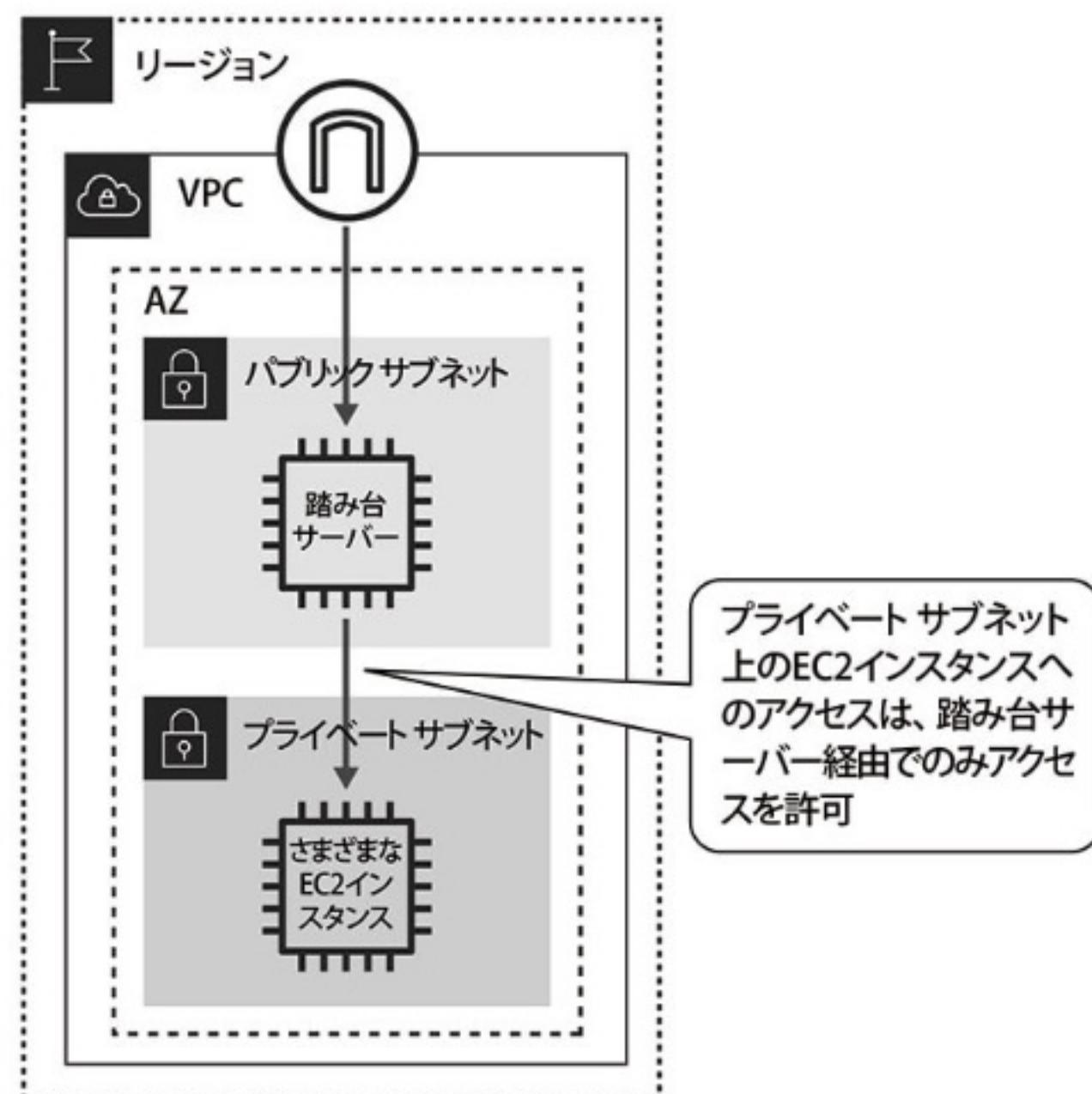
たとえば、VPC内に配置されたEC2インスタンス上のサーバーに、メンテナンスなどの運用のために**SSH(Secure Shell)**や**RDP(Remote Desktop Protocol)**を利用したリモートからのアクセスを許可したいケースがあります。

VPC内のすべてのサーバーのセキュリティグループで、SSHやRDPをインターネットから常時許可しておくのはセキュリティ上好ましくありません。

そのため、メンテナンス時に一時的に踏み台としてアクセス可能な踏み台サーバーを用意しておき、踏み台サーバーにログインしたあとで、各サーバーにアクセスすることでセキュリティリスクを低減します。

踏み台サーバーによるアクセス制御のイメージを以下に示します。

【踏み台サーバーによるアクセス制御のイメージ】



踏み台サーバーの構成のポイントは、以下のとおりです。

- メンテナンスなどでSSHやRDPなどのリモートアクセスが必要となる場合のみ、踏み台サーバーを起動する（通常は停止しておく）
- 踏み台サーバーはパブリックサブネットに設置し、パブリックIPを設定する
- 踏み台サーバーに適用するセキュリティグループには、インターネットから踏み台サーバーに対するSSHやRDPアクセスの許可ルールを設定する
- その際、インターネットからのアクセス元IPは、不特定多数(0.0.0.0/0)ではなく、メンテナンスに利用するPCなどの特定のIPやサブネットを指定することで、よりセキュアなアクセス制御が可能になる
- 踏み台サーバー経由でアクセスされるEC2インスタンスには、必要に応じて「踏み台サーバーからのSSHやRDPのみ許可」をセキュリティグループに設定しておく



試験対策

踏み台サーバー(Bastion)の役割と構成は重要です。



通常、LinuxサーバーへのリモートアクセスにはSSH(TCPポート番号22)、WindowsサーバーへのリモートアクセスにはRDP(TCP/UDPポート番号3389)のプロトコルがそれぞれ使用されます。

5

Webアプリケーションの保護

ネットワークセキュリティに関する重要な対策として、Webアプリケーションの保護があります。

AWS上のサーバーに対しては、インターネットなどを経由してさまざまなセキュリティ攻撃が行われていますが、そのなかでもWebアプリケーションの脆弱性を標的とした攻撃が多く見受けられます。こうした攻撃からWebアプリケーションを保護するサービスとして、**AWS Web Application Firewall (WAF)**と**AWS Shield**が提供されています。

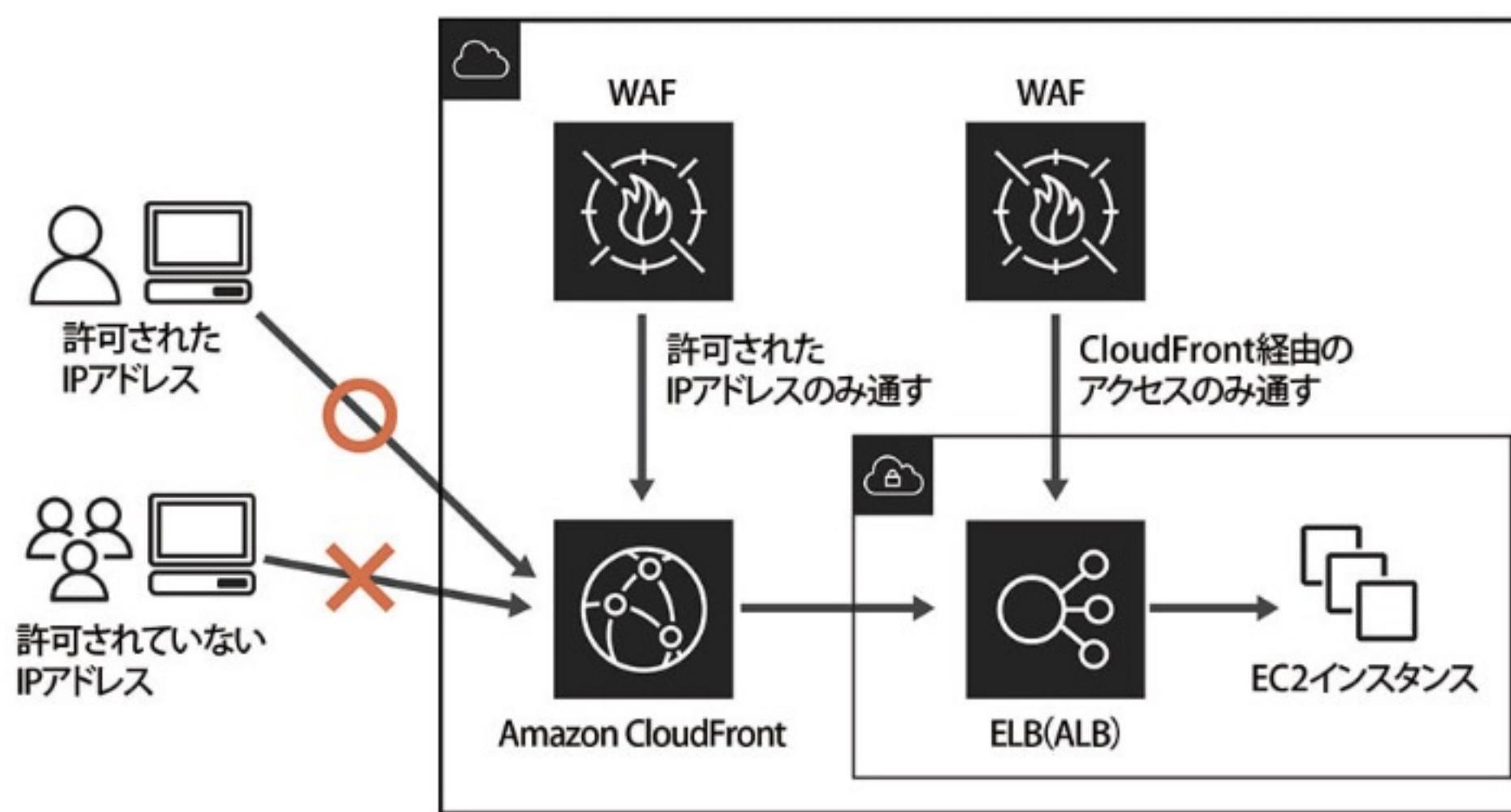
● AWS Web Application Firewall(WAF)

AWS WAFは、Webアプリケーションに対する攻撃のうち、SQLインジェクションやクロスサイトスクリプティングのような一般的な攻撃を防御する機能を提供しています。

送信元のIPアドレスに基づくアクセス制限や、HTTPプロトコル情報(HTTPヘッダや本文、URI文字列など)に対してフィルタリングを設定し、特定の攻撃パターンからWebアプリケーションを保護します。

Amazon CloudFront や Application Load Balancer(ALB)、Amazon API GatewayなどのAWSのサービスにWAFをアタッチ(適用)し、各サービスへのアクセスを制限することができます。

【WAFをCloudFrontとALBに設定した場合のイメージ】



AWS WAFが適用可能なサービス(CloudFront、ALB、API Gateway)は重要です。送信元IPアドレスに基づいてアクセス制限できる点も押さえておきましょう。

● AWS Shield

Webアプリケーションへの攻撃の中でも、特にDoS(Denial of Service)やDDoS(Distributed Denial of Service)などに代表される一斉攻撃を防御するには、AWS Shiledが適しています。たとえば、DoS攻撃で一般的なTCP SYNフラッドなどからの保護が可能です。

無償版のShield Standardと有償版のShield Advancedが提供されています。

有償版は、Amazon EC2やELB、Amazon CloudFrontなどで実行されるアプリケーションを標的とした攻撃に対して、高レベルな保護や、DDoS攻撃発生時に24時間365日のサポート対応が提供されるなど、エンタープライズ向けのサービスになっています。

● AWS Firewall Manager

AWS Firewall Managerは、セキュリティグループをはじめ、AWS WAFやAWS Shield、AWS Network Firewallなどのルールや設定を一元管理できるサービスです。複数のAWSアカウントや環境で、セキュリティ設定を集中管理したい場合に有効です。

たとえば、AWS Organizationsと連携することで、複数アカウントのALBやAmazon CloudFrontのディストリビューションに対して、横断的にAWS WAFルールを簡単に設定管理・適用することができます。



AWS WAFとAWS Shiled Standard/Advancedの違いを押さえておきましょう。特に、SQLインジェクションやクロスサイトスクリプティングにはWAF、DDoS攻撃にはAWS Shieldが有効です。



演習問題

1 10.1.3.2/32から、あるEC2インスタンスへのSSHアクセスを許可したいと考えています。次のうち正しい設定はどれですか。

- A ネットワークACLのインバウンド通信で、アクセス元IPが10.1.3.2/32からのSSHを許可する
- B ネットワークACLのアウトバウンド通信で、アクセス元IPが10.1.3.2/32からのSSHを許可する
- C セキュリティグループのインバウンド通信で、アクセス元IPが10.1.3.2/32からのSSHを許可する
- D セキュリティグループのアウトバウンド通信で、アクセス元IPが10.1.3.2/32からのSSHを許可する
- E 何も設定しない

2 会社のネットワークからインターネットを経由してVPC内のWebサーバーにSSHでリモートアクセスし、メンテナンスを行いたいと考えています。最も適切な方法はどれですか。

- A 踏み台サーバー(Bastion)をVPC内のパブリックサブネットに構成し、セキュリティグループで会社のネットワークからのSSHのみ許可する
- B Webサーバーのセキュリティグループで、インターネットから不特定多数のSSHを常時許可する
- C ネットワークACLでWebサーバーに対するSSHのインバウンド通信を拒否する
- D 踏み台サーバー(Bastion)をVPC内のプライベートサブネットに構成し、セキュリティグループで会社のネットワークからのSSHのみ許可する
- E NATゲートウェイをパブリックサブネットに設置する

3

Amazon CloudFrontとAmazon S3を組み合わせて、静的Webサイトをインターネットに公開しています。悪意のある第三者による攻撃からWebサイトを保護する方法として適切なものはどれですか。

- A AWS WAFを設定し、Amazon CloudFrontに適用する
- B AWS WAFを設定し、Amazon S3に適用する
- C Amazon CloudFrontにセキュリティグループを適用する
- D Amazon S3にセキュリティグループを適用する

A

解答

1

C

セキュリティグループはステートフルな通信制御が可能なため、インバウンド通信のみ許可することでSSHの戻りの通信も許可されます。C以外の選択肢では、設問のとおりには動作しないため、適切ではありません。

2

A

踏み台サーバーにより、リモートからのアクセスを一時的に許可することができます。

なお、選択肢DのようにVPC内のプライベートサブネット上に配置した場合は、インターネット上から踏み台サーバーにアクセスできません。

3

A

AWS WAFは、Webアプリケーションを保護する機能を提供しています。WAFは、AWSサービスのAmazon CloudFrontやALB、Amazon API Gatewayに適用することができます。

2-4 データの保護

AWSでは、ユーザーがクラウド上に保管するさまざまな種類のデータを保護するため、窃盗や改ざんを防止する手段を提供しています。本節では、暗号化に代表されるデータ保護の概要について説明します。

1 暗号化によるデータの保護

AWSに代表されるパブリッククラウドの特性上、データはインターネット経由で伝送され、AWSが管理するデータセンターに保管されます。したがって、ユーザーとAWSの双方でデータを保護する工夫をしなければ、データの窃盗や流出、改ざんによって甚大な被害を受けるリスクがあります。

データ保護の基本的な考え方は「暗号化」です。一般的には、以下の2種類の暗号化を検討する必要があります。

●通信の暗号化

ユーザーとAWSの間を通る通信経路(インターネットなど)における情報窃盗からデータを保護します。

具体的には、**SSL/TLS**などの暗号化の仕組みをユーザーとAWSの間で導入します。

AWSでは、以下に代表されるさまざまなサービスにおいて、データのアップロード／ダウンロードを保護するため、SSL/TLSによる**通信の暗号化**をサポートしています。

- ・ Elastic Load Balancing(ELB)
- ・ Amazon Relational Database Service(RDS)
- ・ Amazon CloudFront
- ・ Amazon API Gateway

たとえば、ロードバランサーであるELBの配下に複数のEC2インスタンスで構成されるWebサーバーなどがある場合は、ELBがクライアントとWebサー

バーの間でSSL/TLSの終端となり、暗号化処理を行います。

またAWSでは、SSL/TLS証明書の購入や登録・更新、証明書の期限切れに対する事前通知などが一元管理できる**AWS Certificate Manager(ACM)**サービスを提供しており、ELBなどへの証明書設定が簡単に行えます。



試験対策

ELBやAmazon RDSなどのサービスと通信する際には、SSL/TLSによる通信の暗号化が利用できます。また、ACMによってSSL/TLS証明書の購入・更新や期限が管理できます。



SSL/TLSの利用時にはSSL/TLS証明書が必要となります。ユーザーが証明書を持ち込むことも可能ですが、AWS Certificate Manager(ACM)を利用することで、AWSサービスで利用する証明書の作成・管理が容易になります。

●保管するデータ自体の暗号化

AWSのAmazon EBSやAmazon S3などに保管されるユーザーデータが、悪意ある第三者からアクセスされることを防ぎます。具体的には、クライアントサイドまたはサーバーサイドでの**ファイル暗号化**などにより実現します。

また、暗号化や復号を行うには「鍵」の作成と管理も重要となります。この暗号化や復号を実行する際に使用する鍵のことを、AWSでは**カスタマーマスターキー(CMK)**と呼びます。

データ暗号化の方式と鍵の作成・管理について、次項から詳細を説明します。

2 データ暗号化の方式と場所

ユーザーとAWSとの間でデータのやり取りを行う場合、暗号化によるセキュリティ強化を検討します。暗号化の方式を検討する際には、「どこで暗号化するか」「鍵の管理はどこで行うか」の2つが重要です。

「どこで暗号化するか」については、クライアントサイドとサーバーサイドの2つの方式があります。

●クライアントサイドでの暗号化(CSE : Client Side Encryption)

AWSにデータを送信・保存する前に、**ユーザーの環境**でデータを暗号化します。

Amazon S3にユーザーが保持するファイルをアップロードする場合は、ユーザー側でファイルに暗号化やパスワード処理などを施してからアップロードします。

また、AWS SDKを利用して、プログラムからAmazon S3へのアップロード時にファイルを暗号化することもできます。

Amazon EBSでは、たとえばWindowsやLinuxなどのOSが提供するファイルシステムの暗号化機能などを用いて、ユーザーがファイルシステム全体を暗号化してからデータを保管します。

●サーバーサイドでの暗号化(SSE : Server Side Encryption)

AWS側でファイルを暗号化します。Amazon S3やAmazon EBS、Amazon Redshift、Amazon S3 Glacierなどの**サービス**で暗号化機能を提供しています。

Amazon S3では、バケットに**デフォルト暗号化オプション**を設定することで、S3バケット内に保存されるファイルが自動的に暗号化されます。

また、AWS SDKやAWS CLIを利用することで、サーバーサイドのファイル暗号化を指定できます。

暗号化に必要な鍵には、次のいずれかを利用します。

- ・ ユーザーが管理している鍵
- ・ Amazon S3で自動生成された鍵
- ・ 後述するAWS Key Management Service(KMS)などのサービスと連携して生成された鍵

Amazon EBSでは、ボリュームの作成時に暗号化を設定できます。

ただし、既存のEBSボリュームを暗号化ボリュームに変更するには、以下の作業が必要になります。

- ① 既存のボリュームのスナップショットを作成
- ② 作成したスナップショットを複製する際に暗号化オプションを指定
- ③ 暗号化されたスナップショットからEBSボリュームを再作成
- ④ EC2インスタンスから既存のEBSボリュームをデタッチ
- ⑤ EC2インスタンスへ暗号化されたEBSボリュームをアタッチ



クライアントサイドでの暗号化(CSE)とサーバーサイドでの暗号化(SSE)の違いを覚えましょう。特に、Amazon S3とAmazon EBSのユースケースは重要です。

3

暗号化に必要な鍵の管理

データの暗号化・復号には、鍵の管理と保管が必要になります。

鍵の管理とは、鍵自体の作成、有効化や無効化、定期的なローテーションなどを指します。

主な鍵の管理方式は、鍵の管理・保管をユーザーが自身の責任で行うか、AWS側で行うかによって以下の3種類に分類されます。

【鍵(CMK)の管理方式】

鍵の管理	鍵の保管	暗号化処理	利用するAWSサービス	概要
ユーザー	ユーザー	CSE	ユーザー側で実施するため、特になし	ユーザーが自らの責任・環境において、鍵を管理し保管する
ユーザー	AWS	CSE、またはSSE	AWS KMS、またはAWS CloudHSM	ユーザーは自ら鍵を作成・管理するが、AWS KMSやAWS CloudHSMサービス上で鍵を保管する
AWS	AWS	SSE	各AWSサービスの暗号化機能(Amazon S3やAmazon EBSなど)	鍵の管理・保管などがすべてAWS側のサービス上で透過的に行われる

4

AWS KMSとCloudHSMによる鍵の管理・保管

AWSでユーザーの鍵の管理と保管を行うには、AWS Key Management Service(KMS)とAWS CloudHSMを利用します。

●AWS KMSによる鍵の管理・保管

AWS KMSは、AWS上で鍵管理を提供するマネージドサービスで、主に暗号化鍵の作成や有効・無効の管理、ローテーション、削除などを行なうことができます。

また、鍵自体はAWS上に保存されます。たとえば、ユーザーが作成した鍵をAWS KMSで管理・保管することで、Amazon S3上のファイルのサーバーサイドでの暗号化(SSE)や、データ送信前にクライアントサイドでの暗号化(CSE)も簡単に行えます。

AWS KMSと連携して暗号化処理できる主なAWSサービスには、以下のものがあります。

- AWS SDKやAWS CLIを利用したクライアント アプリケーション
- Amazon S3、Amazon EBS、Amazon RDS、Amazon Redshiftなどのストレージサービスやデータベースサービス



ユーザーの鍵をAWS KMSで管理することで、サーバーサイドでの暗号化(SSE)やクライアントサイドでの暗号化(CSE)が利用できます。

● AWS CloudHSMによる鍵の管理・保管

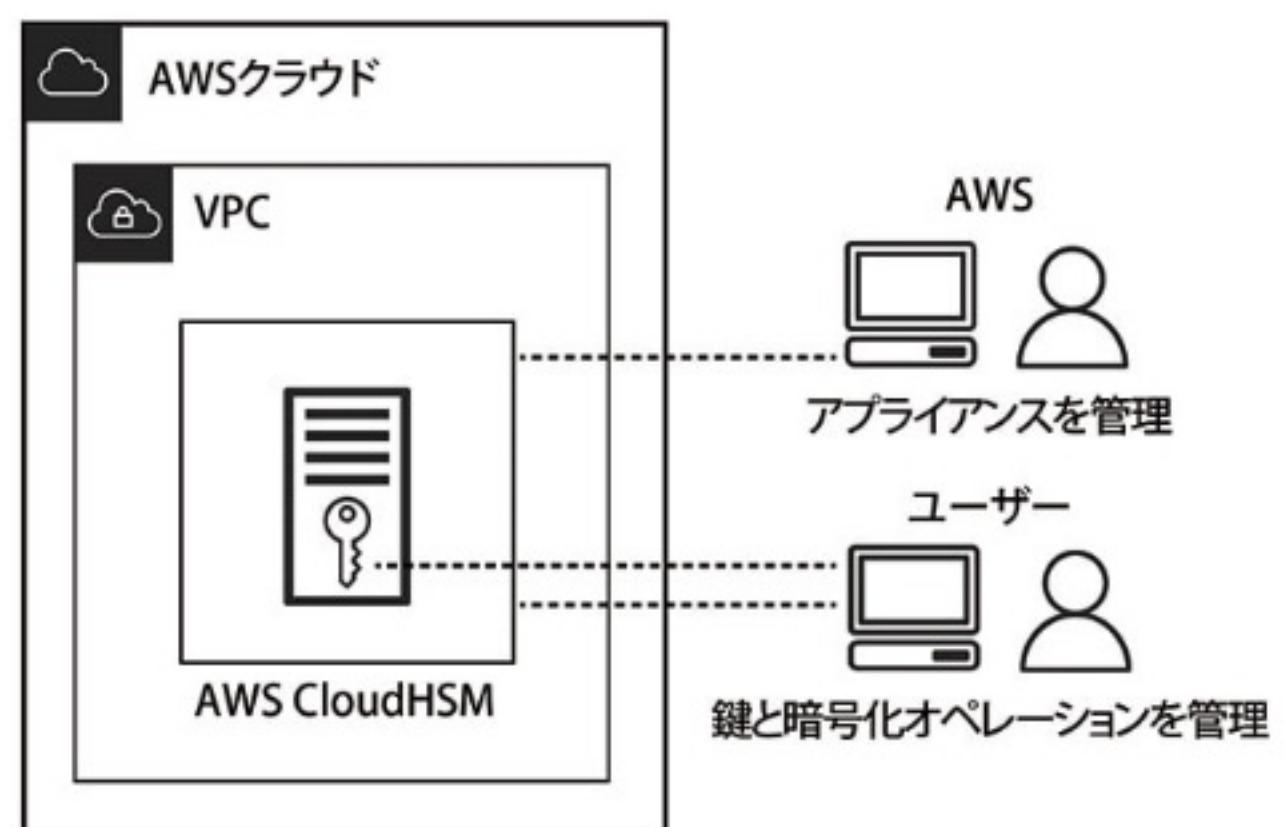
鍵を管理するもう1つの手段として、**AWS CloudHSM**サービスが利用できます。

HSM(Hardware Security Module)は、AWSのデータセンター内に配置されるユーザー占有のハードウェア アプライアンスです。

AWSがHSMのアプライアンス自体を管理し、HSMを占有するユーザーだけが、そのアプライアンスに保存される鍵を管理できます。

HSMは、それ自体がユーザーのVPC内に配置され、ほかのネットワークから隔離されることや、国際的なセキュリティ基準(NIST FIPS140-2など)に準拠していることなどから、セキュリティのコンプライアンス要件が厳しい場合に適用します。

【CloudHSMによる鍵管理のイメージ】



HSMと連携して暗号化処理できる主なAWSサービスには、以下のものがあります。

- Amazon Redshift
- Amazon RDS for Oracle



セキュリティ要件が厳しい場合には、AWS CloudHSMの適用を検討します。

5

AWS Secrets Managerによる認証情報の管理・保管

AWS Secrets Managerは、Amazon RDSやAmazon Redshiftなどのデータベースの認証情報(ユーザーやパスワード)を暗号化して集中管理・保管するサービスです。

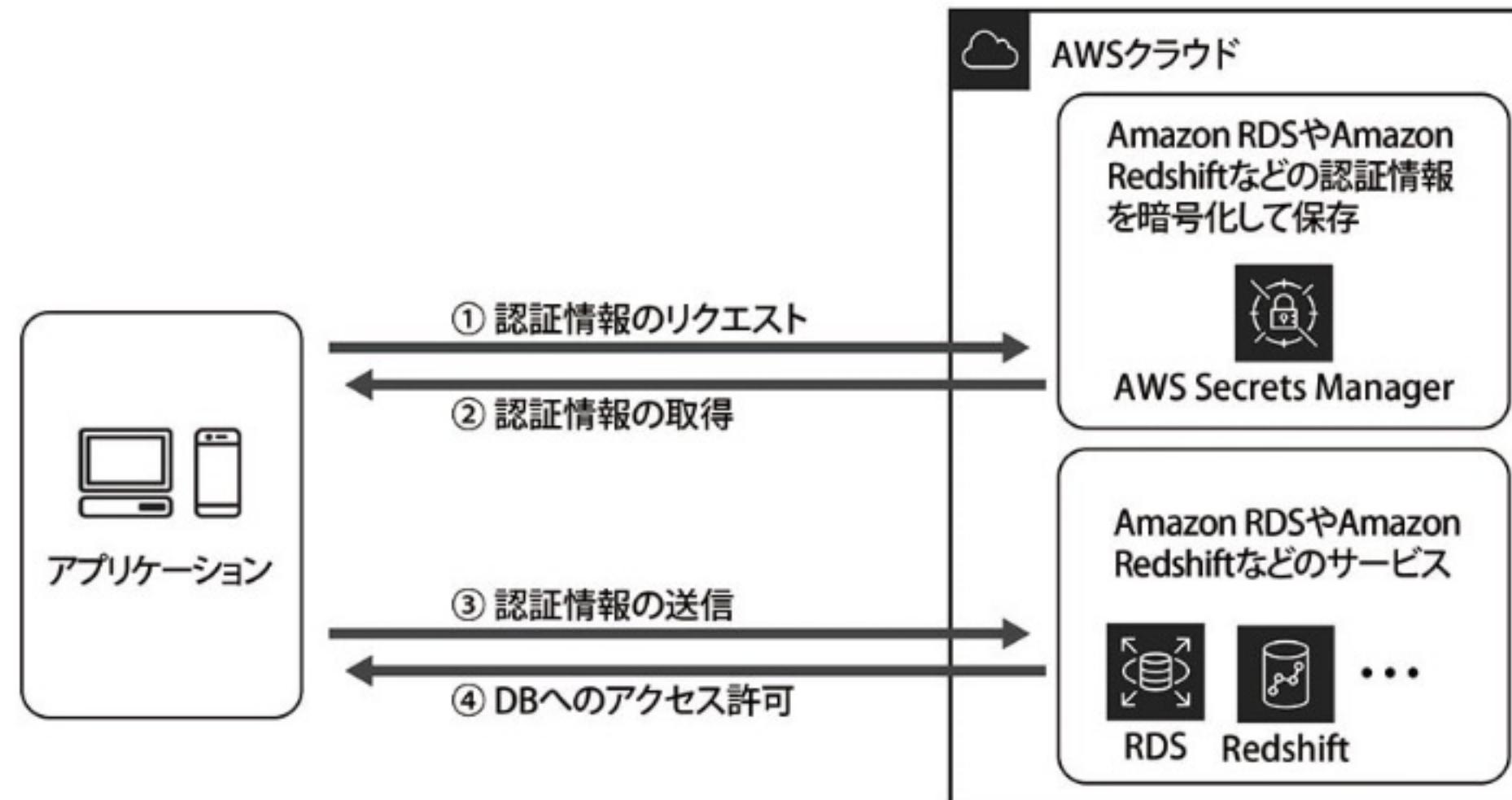
通常、アプリケーションからデータベースにアクセスするためには、アプリケーション内にデータベースの認証情報を保持していかなければなりませんが、漏洩などによるセキュリティ侵害のリスクがあります。

アプリケーションからSecrets ManagerのAPIを実行することで、保管されているデータベースの認証情報をセキュアに取得できます。

そのほかにも、データベースの認証情報を自動的に更新できるという特徴があります。たとえば、定期的かつ自動的にAmazon RDSの認証情報を更新すれば、認証情報漏洩のリスクを低減できます。

また、AWS KMSなどと連携し、認証情報を暗号化する際に鍵の管理をKMSで行うこともできます。

【Secrets Managerによる認証情報取得のイメージ】



AWS Secrets ManagerでAmazon RDSやAmazon Redshiftなどの認証情報を管理することで、アプリケーションからセキュアに認証情報を利用できます。

【S3における暗号化方式】

暗号化方式	暗号化処理	鍵の管理
CSE	クライアントサイド(CSE)	クライアント(ユーザー)が提供・管理
SSE-C	サーバーサイド(SSE)	クライアント(ユーザー)が提供・管理
SSE-S3	サーバーサイド(SSE)	Amazon S3が提供・管理
SSE-KMS	サーバーサイド(SSE)	AWS KMSが提供・管理



SSE-S3やSSE-KMSの用語と、各処理の違いを押さえておきましょう。

6

Amazon S3におけるデータ暗号化

「1-5 ストレージサービス」で説明したとおり、Amazon S3に保管されたデータはさまざまな方式で暗号化できます。

鍵の管理については、クライアント(ユーザー)が独自に作成した鍵を持ち込む、AWS KMSで管理する、Amazon S3がデフォルトで提供する鍵を使うなど、いくつかのオプションがあります。

各オプションは、「データを暗号化(処理)する場所」と「暗号化鍵を提供・管理するサービス」という点で異なります。



演習問題

• • •

1 複数のEC2インスタンスにWebサーバーを構築しています。SSL証明書を利用してSSL通信を可能にする方式として適切な項目はどれですか。

- A AWSの各サービスでは、SSL/TLSをサポートしていない
- B EBSでサーバーサイド暗号化(SSE)処理を行う
- C AWS Certificate ManagerでSSL/TLS通信を許可する
- D S3でサーバーサイド暗号化(SSE)処理を行う
- E EC2インスタンスの手前にELBを作成し、SSL証明書をインストールしてSSLの通信設定を行う

2 Amazon EBS上に保存されるファイルを暗号化したいと考えています。適切な方式はどれですか(3つ選択)。

- A 既存のEBSボリュームのスナップショットを作成し、コピー時に暗号化設定を行う。暗号化されたスナップショットからEBSボリュームを作成する
- B EC2インスタンスを停止してから、EBSボリュームの暗号化設定の変更を行う
- C OSのファイルシステム暗号化機能を利用する
- D EBSボリュームの新規作成時に暗号化設定を指定する
- E EC2インスタンスを起動したまま、EBSボリュームの暗号化設定の変更を行う

3 Amazon S3上に保存されるファイルを暗号化したいと考えています。次の方のうち、適切ではない項目はどれですか。

- A Amazon S3との通信にSSL/TLSを利用する
- B クライアントサイドでファイルを暗号化してからAmazon S3にアップロードを行う

- C Amazon S3のバケットでデフォルト暗号化オプションを設定する
- D AWS KMSで管理しているユーザーの鍵を自動連携し、Amazon S3上でサーバーサイド暗号化を行う
- E ユーザーの鍵を用いてAmazon S3上でサーバーサイド暗号化を行う

4 あなたの会社では、データの暗号化に必要な鍵の管理をAWS上で行いたいと考えています。鍵の管理・運用負荷を軽減する方法として、適切な項目はどれですか。

- A Amazon S3を利用し、暗号化鍵を保存する
- B IAMポリシーを利用し、IAMユーザーによる暗号化鍵へのアクセスを制限する
- C AWS KMSを使用し、暗号化鍵を管理する
- D 多要素認証(MFA)を使用し、暗号化鍵を保護する

A**解答****1****E**

Amazon ELBやAmazon RDSなどのサービスとの通信時には、SSL/TLSによる通信の暗号化が利用できます。E以外の選択肢は、SSL証明書を利用してSSL通信を可能にする方式としては、適切ではありません。

2**A、C、D**

Amazon EBSでは、たとえばWindowsやLinuxなどのOSが提供するファイルシステムの暗号化の機能などを用いて、ユーザーがファイルシステム全体を暗号化してからデータを保管します。ボリュームの作成時に暗号化設定が可能です。ただし、既存のEBSボリュームを暗号化ボリュームに変更するには、以下の作業を行う必要があります。

- ① 既存のボリュームのスナップショットを作成
- ② 作成したスナップショットを複製する際に暗号化オプションを指定
- ③ 暗号化されたスナップショットからEBSボリュームを再作成
- ④ EC2インスタンスから既存のEBSボリュームをデタッチ
- ⑤ EC2インスタンスへ暗号化されたEBSボリュームをアタッチ

3**A**

Amazon S3は、クライアントサイドでの暗号化(CSE)とサーバーサイドでの暗号化(SSE)によるファイル暗号化をサポートしています。SSL/TLSによる通信経路の暗号化も可能ですが、これだけではAmazon S3上に保存されるファイルが暗号化されるわけではないため、クライアントサイドでの暗号化(CSE)とサーバーサイドでの暗号化(SSE)を行う必要があります。

3**C**

AWS KMSにより、AWS上で鍵の管理を行うことができます。主に鍵の作成や有効・無効の管理、ローテーション、削除などを行うことができ、鍵の管理・運用負荷を軽減することができます。

2-5**セキュリティ監視**

AWSでは、セキュリティを監視するためのさまざまな手段をサービスとして提供しています。

本節では、セキュリティに関連するインシデントやログ監視などのサービスについて説明します。

1**セキュリティ監視の関連サービス**

ここまでこの節で、AWSにおけるさまざまなセキュリティ対策やサービスを紹介しました。セキュリティ対策で重要なことは、単にセキュリティを設定するだけではなく、日々の運用でセキュリティ状況のチェックや監視を継続的に行うことです。そうすることにより、思わぬ対策漏れやインシデントの早期発見が可能になります。

AWSのセキュリティ監視に関するサービスとして押さえておくべき代表的なものは、以下の11のサービスです。

- AWS CloudTrail
- VPCフローログ
- Amazon GuardDuty
- Amazon CloudWatch Logs
- AWS Config
- AWS Trusted Advisor
- Amazon Inspector
- AWS Artifact
- AWS AuditManager
- Amazon Macie
- AWS Security Hub

● AWS CloudTrail

AWS CloudTrailは、AWSアカウントで利用された操作(APIコール)を、ログとして記録するサービスです。さまざまな操作ログを蓄積することで、AWSへの不審なアクセスや操作がなされていないか、意図しない設定変更が行われていないかなど、さまざまなセキュリティ監視や監査に活用できます。主な特徴は以下のとおりです。

- ・ AWSアカウントを取得した時点で有効化され、過去90日間のサービスに対する操作を表示する
- ・ 取得されたログはデフォルトでAmazon S3に保存され、後述のCloudWatch Logsへの連携も可能
- ・ AWSのさまざまなサービスをサポート・連携してログ記録を行う
- ・ 記録内容は、サービスのAPIコール元、時間、送信元IPアドレス、呼び出したAPI、対象となるリソースなど

例として、以下のようなサービスのログやイベントが記録できます。

- ・ 管理コンソールで、AWSアカウントのルートユーザーによるログイン履歴を記録
- ・ EC2インスタンスの操作履歴(削除など)
- ・ AWS KMSで管理されている鍵の使用や削除履歴



AWS CloudTrailは、さまざまなサービス(Amazon EC2やAWS IAM、AWS KMS、ELBなど)の操作ログを収集することで、セキュリティインシデントに関連する操作を監視できる点が重要です。

● VPCフローログ

VPCフローログは、VPC内のネットワークインターフェイス間で行き来する通信の内容をキャプチャする機能です。意図しない通信が行われていないか、などの監視・監査に利用します。VPCフローログを有効化することで、後述するCloudWatch Logsを使用してログを保存・可視化できます。

記録されるログとしては送信元IP・ポート、宛先IP・ポート、プロトコル番号など、ネットワーク通信に関する詳細な内容です。

● Amazon GuardDuty

Amazon GuardDutyは、AWS内の各種ログ(AWS CloudTrailやVPCフローログ、Amazon Route 53のクエリログなど)を監視し、悪意のある第三者による攻撃や不正操作などのセキュリティ脅威を検知するサービスです。

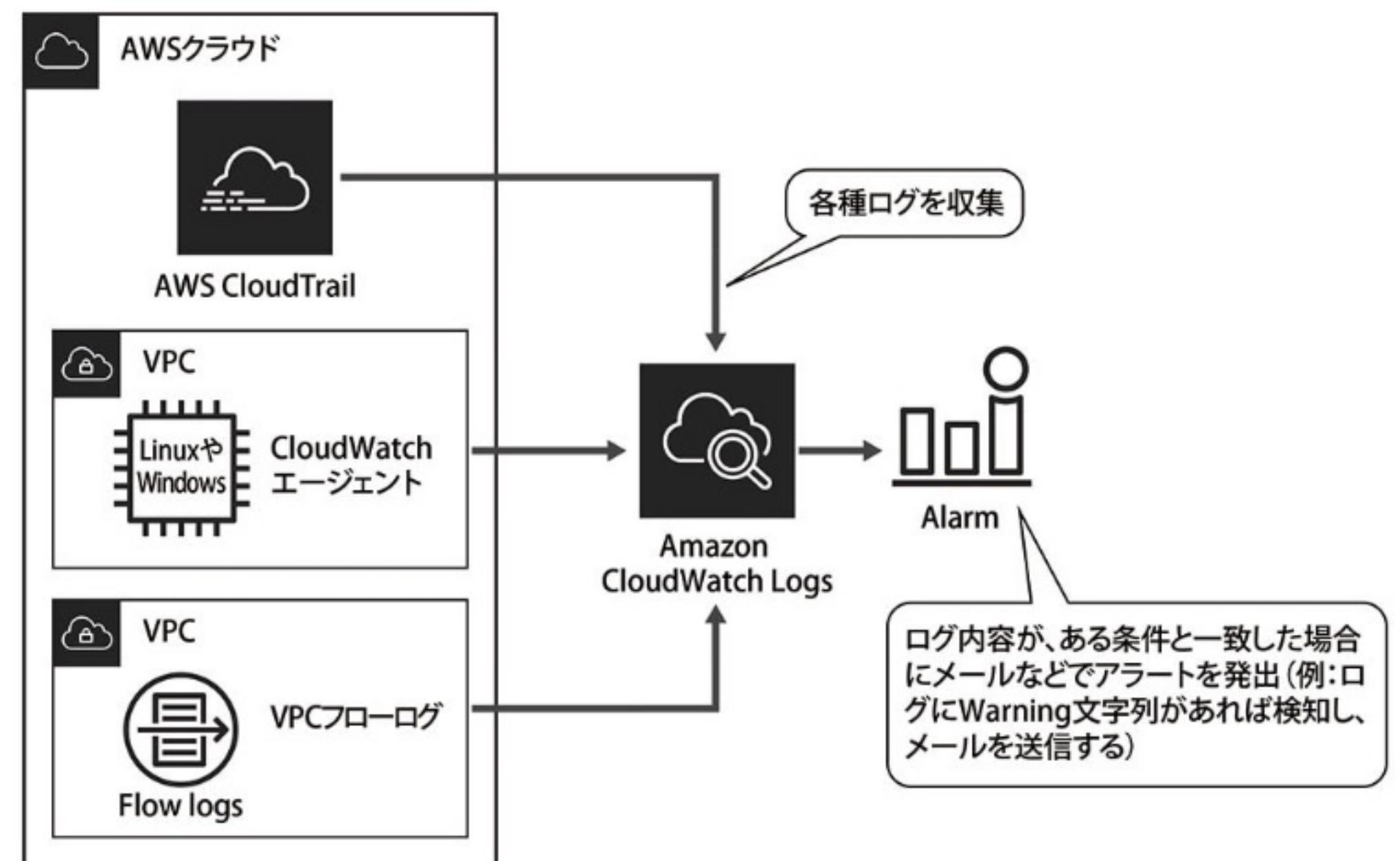
たとえば、マルウェアによって動作が不審なEC2インスタンスの検出や、これまでにあまり利用されていないリージョンでのEC2インスタンス起動など、脅威の可能性がある動作を検知します。これらの脅威を継続的に検知するために、機械学習による不正な動作の学習や異常検知の仕組みが利用されています。

● Amazon CloudWatch Logs

Amazon CloudWatch Logsは、AWS CloudTrailやVPCフローログなど、AWSのさまざまなログを統合的に収集するサービスです。AWSのログだけでなく、LinuxなどのサーバーOSにログを収集・通知するエージェントをインストールすることで、さまざまなログをCloudWatch Logsに送信することができます。

また、収集したログはセキュリティ監視に利用できます。たとえば、あるログ内のWarning文字列をフィルタリングして監視し、CloudWatchのアラーム機能でアラートをメール通知する、といった運用が可能になります。

【CloudWatch Logsによるログ収集とログ監視・通知の連携イメージ】





Amazon GuardDutyにより、悪意のある第三者による攻撃や不正操作をAWS内で監視・検知することで、セキュリティを強化できる点は重要です。

● AWS Config

AWS Configは、AWSのサービスで管理されているリソースの構成変更を追跡するサービスです。

たとえば、EC2インスタンスの作成や削除などの構成変更の履歴を取得できます。さらに、変更をメールなどで通知することで、意図しないリソース変更の監視・追跡が可能になります。



AWS CloudTrailはユーザー(ヒト)の操作(APIコール)を追跡するサービスですが、AWS Configはリソース(モノ)の構成変更履歴に特化した監視サービスです。

● AWS Trusted Advisor

AWS Trusted Advisorは、AWSのベストプラクティスに基づいて、コスト最適化、セキュリティ、耐障害性、パフォーマンス、サービスの制限の5項目でユーザーのAWS利用状況をチェックし、改善すべき事項を推奨するサービスです。

Trusted Advisorのセキュリティ機能では、たとえば、以下の観点でのセキュリティ診断が実行できます。

● セキュリティグループ—開かれたポート

特定のポートに対して、無制限アクセス(0.0.0.0/0)を許可しているセキュリティグループのルールをチェックします。

● ルートアカウントの多要素認証

AWSアカウントのルートユーザーで、多要素認証(MFA)が有効にされていない場合にアラートを表示します。

● Amazon Inspector

Amazon Inspectorは、EC2インスタンスやAmazon ECRに登録されている

コンテナイメージのセキュリティを高めるサービスです。たとえば、EC2インスタンスの場合は、Inspectorのエージェントをインストールすることで、EC2インスタンス上にあるアプリケーションの脆弱性などを診断します。

● AWS Artifact

AWS Artifactは、コンプライアンス関連の情報を一元管理するサービスです。

AWS Artifactから、AWSのセキュリティおよびコンプライアンスレポート(AWS Artifact Report)や、秘密保持契約(NDA)などのAWSとの関連契約(AWS Artifact Agreements)を、オンラインで管理・参照することができます。

たとえば、AWS Artifact Reportsでは、Service Organization Control(SOC)、Payment Card Industry(PCI)レポートなど、第三者監査機関のコンプライアンスレポートをオンデマンドで参照し、ダウンロードすることができます。

さらに、AWSが準拠しているコンプライアンス関連情報を、利用者がセルフサービスで即座に入手できるため、AWSの利用者はガバナンスへの準拠状況を即座に顧客に開示したり共有することができます。

● AWS AuditManager

AWS AuditManagerは、AWSアカウント内のリソースがコンプライアンスに違反していないか、継続的にチェック・監査する仕組みです。

通常、これらのコンプライアンス準拠に関わる監査は、業界や企業のルールが一致していない状況で、第三者機関等に依頼して定期的に実施する必要がありました。AWS AuditManagerは、これらの定期監査を仕組化し、インターネット・セキュリティ標準化フレームワークの1つであるCIS(Center for Internet Security)ベンチマークや、PCI DSS、GDPRなどの各種規格に沿って、継続的なチェックを自動化することができます。

● Amazon Macie

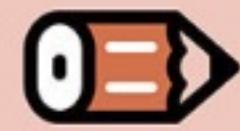
Amazon Macieは、機械学習とパターンマッチングを使用し、Amazon S3などに保存されている機密データを検出して保護するサービスです。

個人識別情報(PII)などのデータを特定できます。たとえば、Amazon S3に保存されたファイルから氏名や電話番号、クレジット番号などが含まれるデータを検出し、アラートを発出することができます。

● AWS Security Hub

AWS Security Hubは、AWS内のセキュリティの状態が、セキュリティ標準およびベストプラクティスに準拠しているかどうかを包括的に把握し、アラートを一元化します。たとえば、前述したAmazon GuardDutyやAmazon Inspector、Amazon Macieなどのサービスから発生するセキュリティアラートを一元的に集約し、1つの管理画面でわかりやすく把握することができます。

また、AWS Security Hubにより、複数のAWSのセキュリティ関連サービスから発生する膨大な数のアラートを個別に確認・対応するといった手間を減らすことができます。



試験対策

- Amazon GuardDuty、Amazon Inspector、Amazon Macieのサービスの違いと特徴を押さえておきましょう。
- GuardDuty … AWSのさまざまなログを監視し、悪意のある第三者などによる攻撃や不正操作などのセキュリティ脅威を検知
 - Inspector … EC2インスタンスなどに対する脆弱性などのセキュリティチェックを実施
 - Macie … Amazon S3などに保存されているデータを機械学習で解析し、個人情報などの機密データを検出

2

AWS上でのセキュリティ侵入テスト

継続的なセキュリティ監視の手段として、AWSを利用するユーザー側で定期的に脆弱性のスキャンやペネトレーションテスト(侵入テスト)などのセキュリティ診断を行うことも重要です。

AWSでは、「AWS脆弱性／侵入テストリクエストフォーム」から申請し許可を得ることで、ユーザー側でこれらのテストを実施することが可能ですが。ただし、AWSのポリシーとして、以下のリソースなどに対するテストのみが許可されています。

- EC2インスタンス、NATゲートウェイ、Elastic Load Balancing(ELB)
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora

- Amazon API Gateway
- AWS Lambda関数およびLambda Edge関数
- Amazon Lightsailリソース
- Amazon Elastic Beanstalk環境
- AWS Fargate



ユーザー側でAWSへの脆弱性スキャンや侵入テストを行うには、事前申請が必要です。



演習問題

1

EC2インスタンスで稼働しているLinuxサーバーのログを収集して監視したいと考えています。適切なサービスは、次のうちどれですか。

- A AWS Config
- B VPCフローログ
- C AWS Trusted Advisor
- D AWS CloudTrail
- E Amazon CloudWatch Logs

2

あなたの会社では、EC2インスタンスのセキュリティに関わる一般的な脆弱性を定期的に診断したいと考えています。適切なサービスはどれですか。

- A Amazon GuardDuty
- B Amazon Inspector
- C Amazon Macie
- D AWS Config

A

解答

1

E

Amazon CloudWatch Logsは、AWS CloudTrailやVPCフローログなどさまざまなAWSのログを統合的に収集するサービスです。AWSのログだけでなく、LinuxなどのサーバーOSにログを収集・通知するエージェントをインストールして、各種のログをCloudWatch Logsに送信することも可能です。

2

B

EC2インスタンスなどのセキュリティ診断を行うサービスは、Amazon Inspectorです。EC2インスタンスにInspectorのエージェントをインストールすることで、EC2インスタンス上のアプリケーションの脆弱性などを診断可能です。

Column



AWS認定試験のアップデートと新試験について

AWSのサービスは、機能の拡充や使いやすさの向上などを目的に日々更新されており、数ヶ月前には実現できなかった構成が、直近のアップデートで可能になることもあります。そのため、AWS認定試験で出題される問題内容についても定期的に更新され、最近では以下のような改定が発表されています。(2022年12月現在、予定を含む)

2022年4月 AWS Certified SAP on AWS—Specialty (PAS-C01)

2022年7月 AWS Certified Advanced Networking—Specialty (ANS-C01)

2022年8月 AWS Certified Solutions Architect—Associate (SAA-C03)

2022年11月 AWS Certified Solutions Architect—Professional (SAP-C02)

2023年 DVA (AWS Certified Developer—Associate認定)

2023年 DOP (AWS Certified DevOps Engineer—Professional認定)

このなかで特に注目したいのが、2022年4月にリリースされたSAP on AWS認定試験(PAS-C01)です。この試験はアップデートではなく、新しい認定試験として登場しました。試験ガイドには、「AWSでSAPワークロードを最適に設計、実装、移行、運用するための高度な技術スキルと経験を検証します」とあり、ERPであるSAPをAWS上で構成するための知識が問われます。

試験ガイドを読んでも、SAPの知識がないと問題文を読み解くことができないと思うかもしれません、AWSの知識があれば、解答のきっかけをつかむことができます。SAPというサードパーティの要素が含まれているとはいえ、求められる知識は、AWS Well-Architectedで説明されているベストプラクティスなどがベースになっているからです。

この観点は、前述した新試験であることが理由ではなく、AWSを扱ううえで全般的に必要とされる観点であり、本書の冒頭(第1章第1節)でAWS Well-Architectedについて説明しているのも、それが理由です。

認定試験の勉強をしていると、サービスの機能や仕組みに注目しがちですが、AWSの観点を理解し、ユーザーが何を求めているのか(試験では問題の要件)を、正しく把握できるようにしておきましょう。



第3章

AWSにおける 高可用アーキテクチャ

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

3-1

高可用性の定義

コンピュータ上で動作するアプリケーションには、システム要件に応じてさまざまな可用性が求められます。AWSでは、オンプレミス環境よりも低コストかつ多様なレベルで、可用性をシステムに組み込むことができます。

本節では、可用性について一般的な考え方とAWSにおける考え方の概要を説明します。

1

一般的な可用性の定義

可用性とは、システムが正常に継続して動作し続ける能力を指します。

可用性の指標として「**稼働率**」が用いられ、多くの場合、パーセンテージで表されます。稼働率を高めるためには、サーバーを**冗長化**^{*1}し、万一、障害が発生しても、すぐに別のサーバーへフェイルオーバーするアーキテクチャを設計するのが一般的です。

フェイルオーバーとは、稼働中のサーバーで障害が発生し、正常に動作しなくなったときに、待機しているサーバーへ自動的に切り替える仕組みのことです。基幹業務系システムの多くで採用されています。

次の表に、稼働率と年間停止時間の目安を示します。

【稼働率に対する年間停止時間の目安】

稼働率	年間停止時間
99%	3日15時間36分
99.90%	8時間46分
99.95%	4時間23分
99.99%	52分34秒

*1 【冗長化】システムやサーバーを単一ではなく複数で構成している状態のこと。冗長化することによって、システムの一部に障害や異常が発生してもシステム全体としては処理を継続することができる。



AWSサービスのなかには、SLA^{*2}で稼働率が公表されているサービスがありますが、この稼働率はあくまでも努力目標値としての位置づけです。詳しくは、以下のURLを参照してください。
<https://aws.amazon.com/jp/legal/service-level-agreements/>

2

AWSにおける可用性向上策

システム障害の原因の多くはハードウェアによるものです。AWSにおいてもハードウェアは存在するため、システム障害が起きる可能性があります。また、AWSサービスの特性に応じて、どのように高可用性を実現するかはユーザー側で設計する必要があります。

AWSでは、**Design for Failure**(障害発生を前提としたシステム構築)という考え方があり、障害を回避するシステムを設計するのではなく、障害が発生してもシステムが継続できるように設計することが重要だとしています。

では、AWSにおいて可用性を向上させるためには、どのようなことを考慮すべきでしょうか。以下に、そのポイントをあげます。

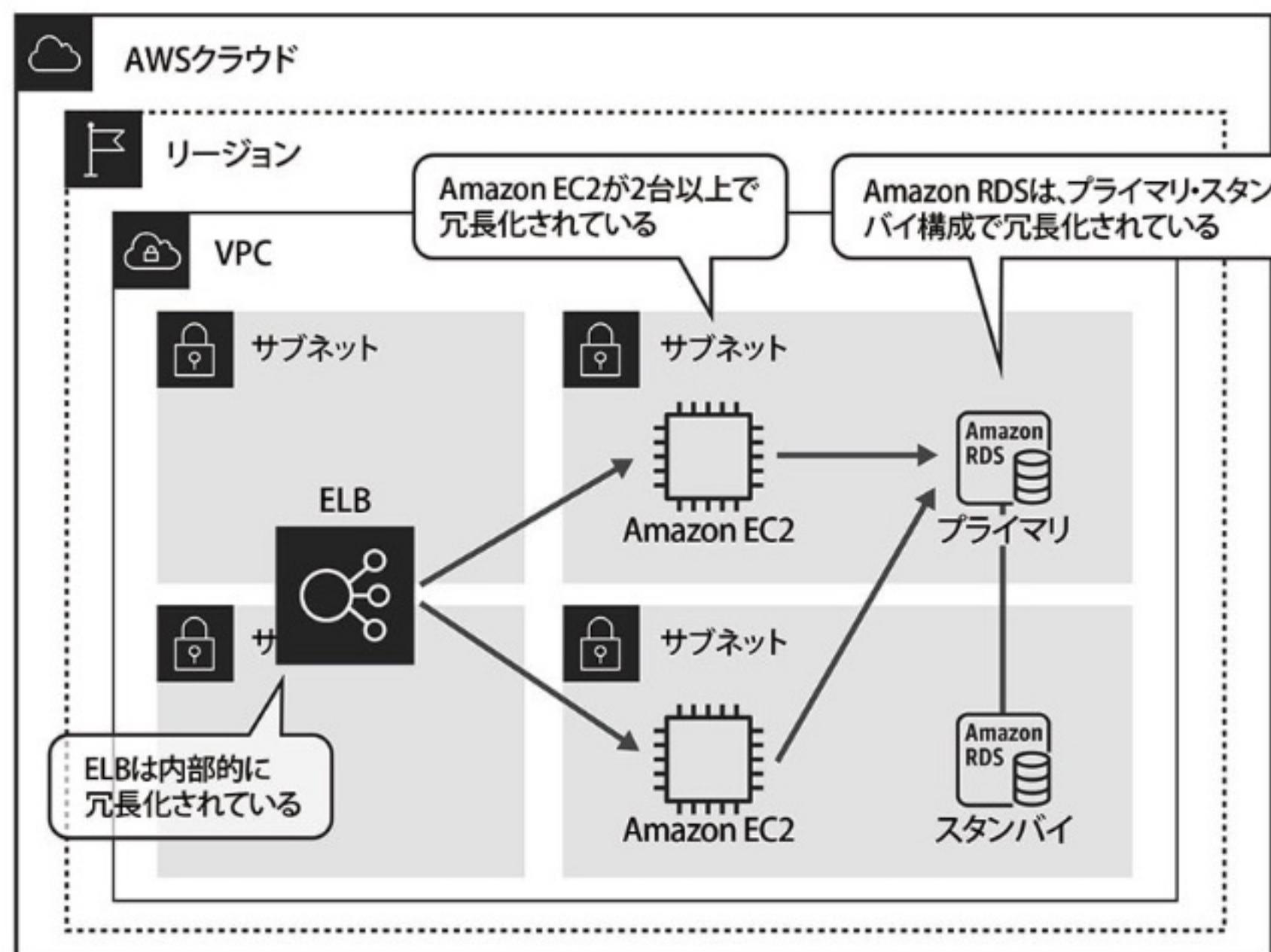
●リソースを冗長化する

オンプレミス環境と同様に、**单一障害点(SPOF)** : Single Point of Failureを作らないことが重要です。单一障害点とは、障害が発生するとシステム全体が使用不能になる、もしくはシステムの動作結果の正しさを保証できなくなる箇所を指します。

AWSサービスのなかには内部的に冗長化されているリソースもありますが、一部のサービスはユーザー側で冗長構成を設計する必要があります。また、コンポーネントを連結してシステムを構成している場合、システムに1カ所でも单一障害点が存在すると、システム全体の可用性に影響します。したがって、それぞれのコンポーネントで冗長性を持たせているかどうかを確認する必要があります。

*2 【SLA】Service Level Agreement : サービス提供者とサービス利用者の間で結ばれる品質基準のこと。たとえば、サーバーの場合は稼働率の保証をSLAとして規定する。

【リソースの冗長化例】



●地理的に離れた場所で冗長化する

オンプレミス環境では、地理的に離れた場所で冗長化する場合は拠点ごとにハードウェアを調達し、それぞれのデータセンターでシステムを構築する必要があります。

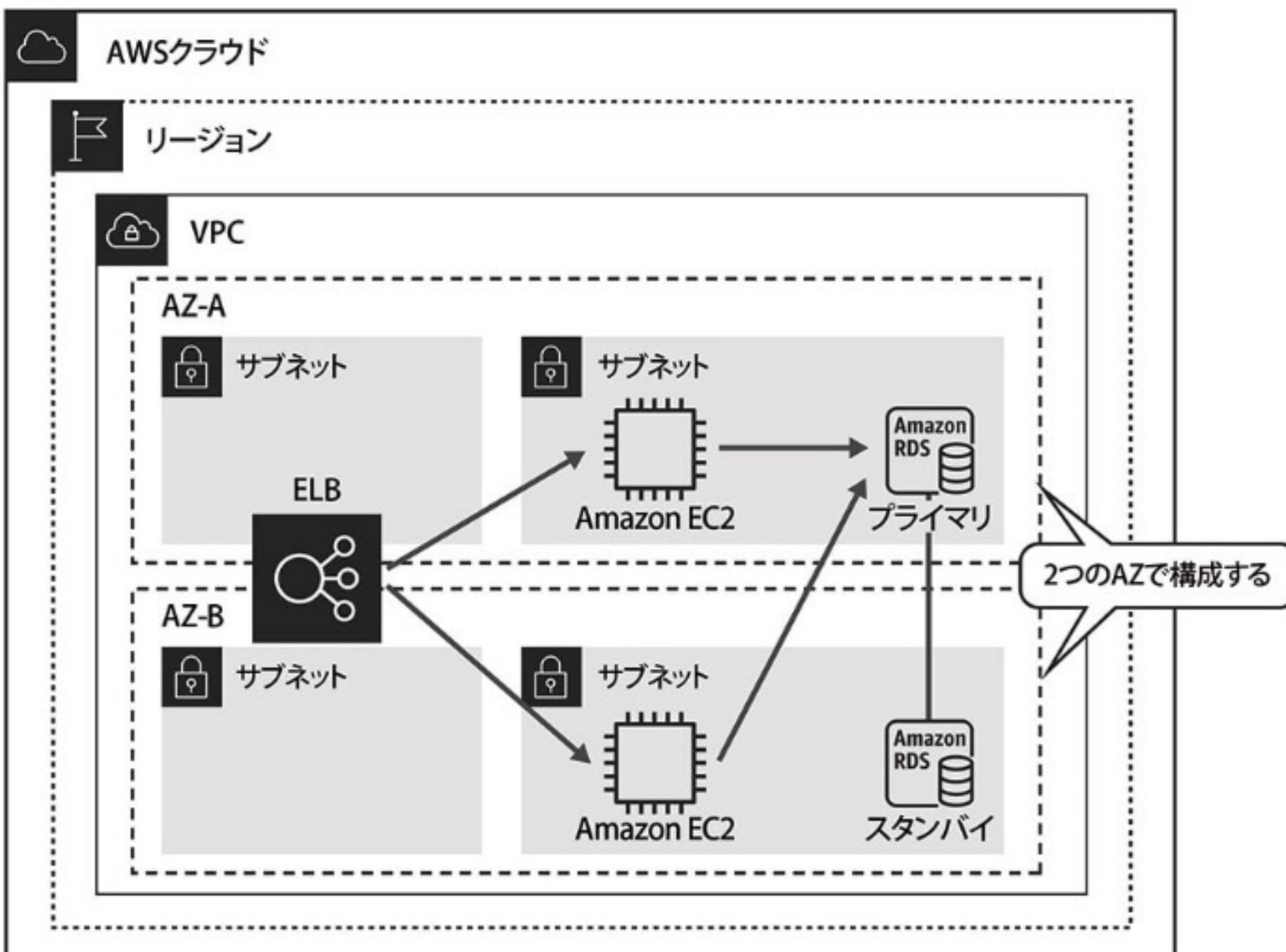
一方で、AWSのようなクラウド環境では、管理画面から設定を行うだけで、地理的に離れた場所で即座にリソースを用意することができます。

次の図は、AZ-AとAZ-Bの2つのアベイラビリティーゾーン(AZ)に、それぞれEC2インスタンスとRDSのデータベースインスタンスを構成する冗長化の例です。

EC2インスタンスはELBによって複数のAZに冗長化し、RDSはマルチAZを有効化して冗長化しています。

AWSでは、このような構成でどのサブネットにリソースを配置するかを、設定画面で選択するだけで実現できます。

【AZを横断する地理的な冗長化例】



●システムを疎結合で構成する

可用性を高めるうえで、システムやコンポーネントを独立させる(疎結合にする)考え方非常に重要です。

クラウドはシステムの疎結合化と相性がよく、オンプレミス環境よりも自由にコンポーネント単位でリソースを調達できます。疎結合化することで、システムの一部分が故障したり動作不良を起こした場合に、その部分だけを切り離して復旧作業を行うことが可能になり、システム全体への影響を軽減することができます。また、システムが拡張しやすくなるというメリットもあります。

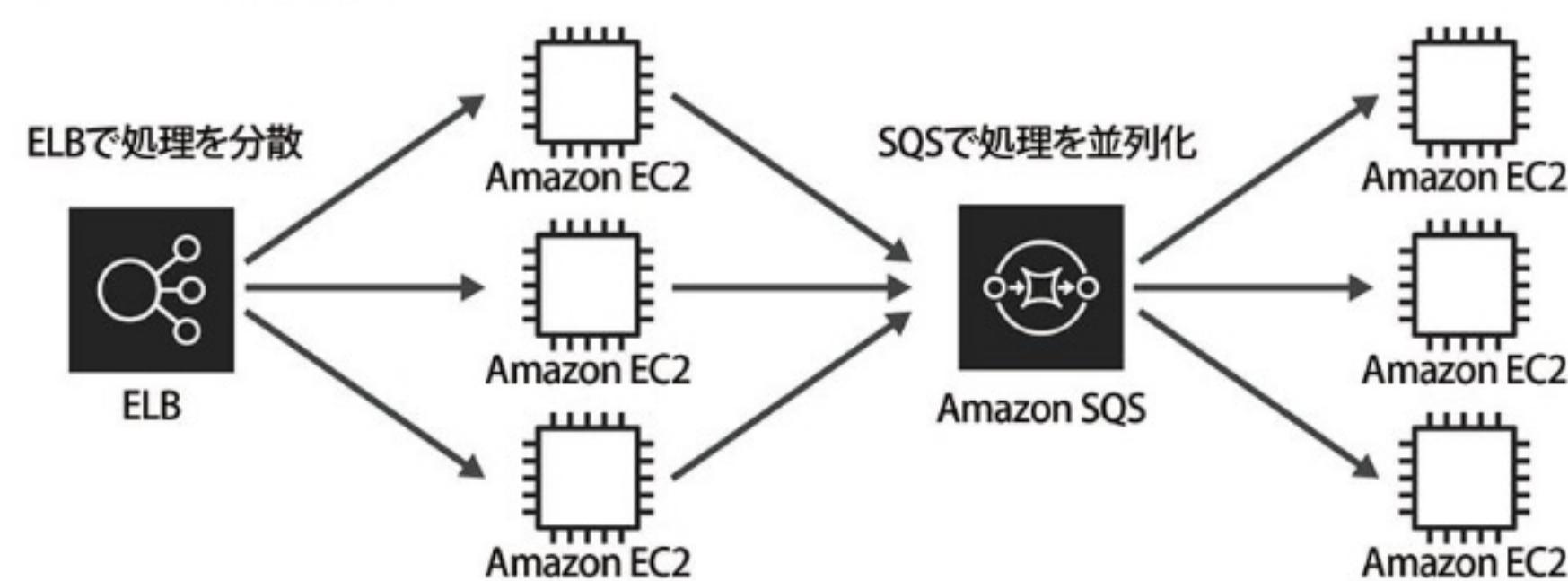
次の図に、Elastic Load Balancing(ELB)とAmazon Simple Queue Service(SQS)を利用した疎結合な構成の例を示します。

ELBを利用して、ELBのエンドポイントから複数のEC2インスタンスに処理を分散できるため、ほかのEC2インスタンス同士が干渉しない疎結合な構成を実現できます。

Amazon SQSを利用することで、大量の処理を行う必要がある場合でもSQS側でバッファリングしてEC2インスタンスへの処理を並列化することができます。

なお、ELBとSQSはAWS内部で冗長されているため、单一障害点になることはありません。

【疎結合な構成例】



これらの考慮点を踏まえたうえで、次節よりネットワーク、コンピューティング、ストレージ、データベースの観点から、高可用性を実現する方法を説明します。

Q 演習問題

1

ある会社では、Amazon EC2とAmazon RDSを利用したアプリケーションを運用しています。ビジネスの加速にともなって要求される稼働率が高まってきたため、冗長化を検討しています。次のうち、EC2とRDSを冗長化する場合の適切な構成はどれですか。

- A EC2は同じAZに2台配置し、RDSはマルチAZを有効化する
- B EC2は同じAZに2台配置し、RDSは2インスタンス構築する
- C EC2は異なるAZに2台配置し、RDSはマルチAZを有効化する
- D EC2は異なるAZに2台配置し、RDSは2インスタンス構築する

A 解答

1

C

EC2のようなコンピューティングリソースを冗長化する場合、同じAZよりも異なるAZで構築したほうが、AZ障害が発生した場合でも別のAZでシステム運用が継続できます。

RDSはマルチAZを有効化することで、複数のAZにインスタンスを構築でき、プライマリに障害が発生した場合でも、自動でスタンバイへ切り替えることができます。

一方、RDSを2インスタンス構築した場合は、プライマリに障害が発生すると手動で切り替えるか、自動で切り替える仕組みを用意する必要があります。

したがって、Cが正解です。

3-2

ネットワークにおける高可用性の実現

システムの可用性を高めるためには、アプリケーションだけでなくネットワークの可用性も考慮する必要があります。本節では、AWSにおけるネットワークサービスから、各サービスにおける高可用性の実現方法までを説明します。

1

ネットワークサービス

まず最初に、各AWSサービスごとに高可用性ネットワークを実現する方法について説明します。

●Amazon Virtual Private Cloud(VPC)

Amazon Virtual Private Cloud(VPC)は、Amazon EC2やAmazon RDSなど、アプリケーションで利用頻度の高いサービスを構成するうえで必須のサービスです。

オンプレミスのネットワーク環境のように、LANケーブルの冗長化やネットワーク機器の冗長化を考慮する必要はほとんどありませんが、以下の事項については、しっかりと検討する必要があります。

- データセンターレベルの障害や特定地域の自然災害などに対応するため、複数のアベイラビリティーゾーン(AZ)でサブネットを構成する
- ネットワークセキュリティの境界を明確にするため、サブネットはパブリックサブネットとプライベートサブネットを構成する
- 将来利用するIPアドレス数を見越してIPアドレスを設計する

次の図に、基本的なサブネットの構成例を示します。

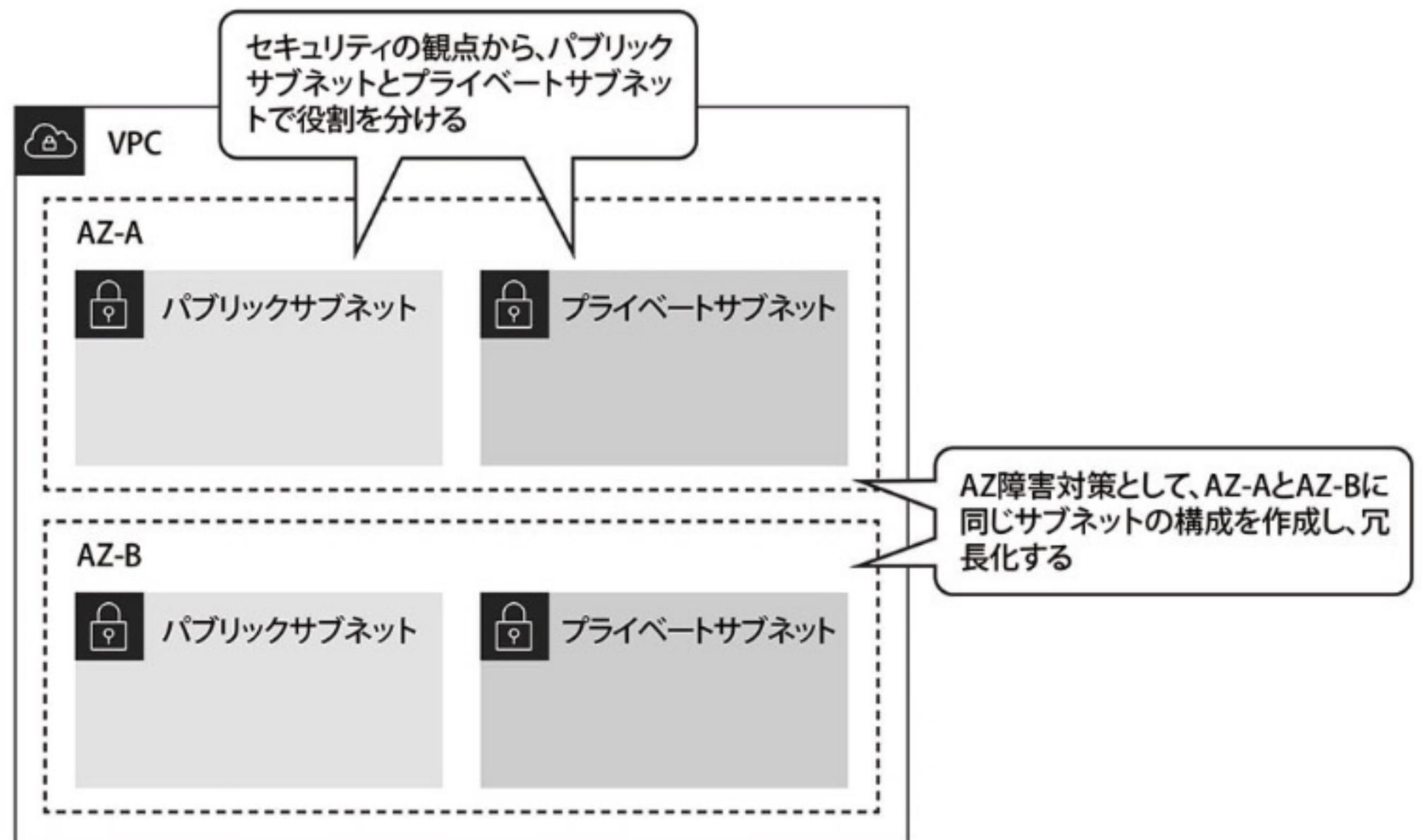
この例では、システムの通信要件に従ってリソースが配置できるように、**パブリックサブネット**と**プライベートサブネット**の、2つの役割でサブネットを構成します。

また、それぞれのサブネットで地理的に離れた場所で冗長化するため、もう

1つのAZでも同じ構成を組んでいます。

それぞれのAZでパブリックサブネットとプライベートサブネットのセットを構成することで、AZに障害が発生した場合でも、もう一方のAZのネットワークでシステムを継続させることができます。

【VPCの構成例】



●NATゲートウェイ

プライベートサブネット内のリソースからインターネットへ接続するためには、**NATゲートウェイ**(22ページを参照)を配置することになります。

NATゲートウェイは、AZ内では冗長化されていますが、複数のAZ間では冗長化されていません。そのため、複数のAZを横断するVPCネットワークを構成する場合、それぞれのAZでNATゲートウェイを配置するかどうかを検討する必要があります。

●AWS Direct Connect

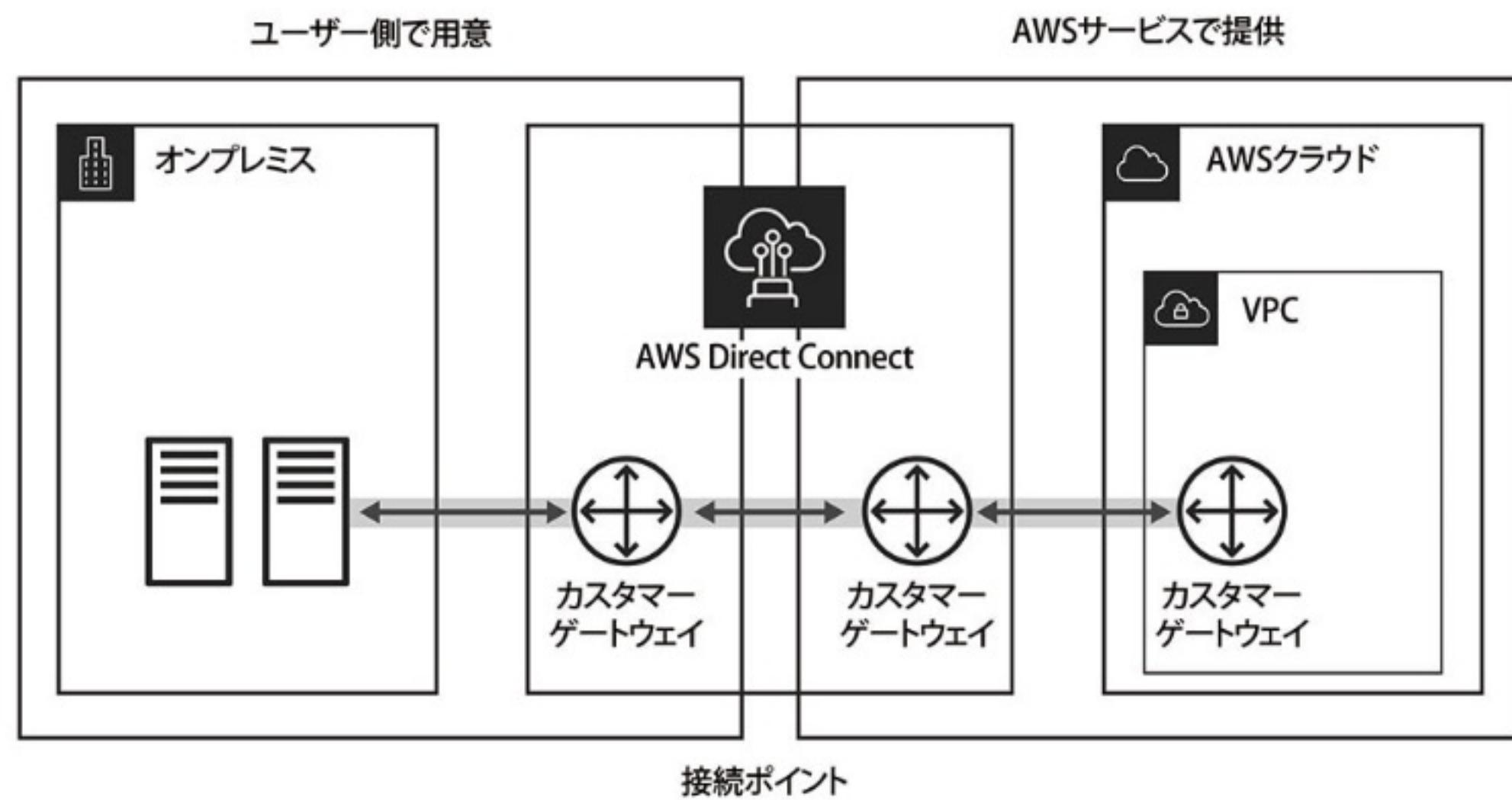
AWS Direct Connectは、「接続ポイント」と呼ばれるロケーションを経由して、オンプレミス環境とAWSとの間を専用線で接続するサービスです。高速のネットワーク帯域で安定した通信を実現することができます。

オンプレミス環境と接続ポイント間の可用性は、ユーザーが考慮する必要があります。この回線の可用性は通信キャリアの提供プランによって異なるため、

Direct Connectを契約する前に確認しておく必要があります。

なお、接続ポイントとAWS間の可用性は、AWSが提供しています。

【Direct Connectの接続例】



●Amazon Route 53

Amazon Route 53は、ドメインネームシステム(DNS)のマネージドサービスです。AWS側で高い可用性を提供しているため、Route 53の可用性を利用して、より高可用なネットワークを構築することができます。

たとえば、Amazon EC2やAmazon RDSを使用するアプリケーションは、複数のAZ間の高可用性を実現するためにELBを活用するケースがほとんどです。

一方で、複数リージョン間の高可用性を実現する場合には、DNSフェイルオーバー機能を備えたフェイルオーバールーティング(33ページを参照)を活用することで、特定のリージョンに障害が発生した場合でもDNSをすぐに切り替えて、別リージョンでアプリケーションを稼働できます。

2

高可用ネットワークの構築

それでは、これまでに説明したサービスを利用して高可用ネットワークを構築するには、どうすればよいでしょうか。ここからは、オンプレミス環境とAWS間のネットワークや、AWS内のネットワークで高可用性を実現する方法を説明します。

●オンプレミス環境-AWS間の接続

オンプレミス環境とAWSの間の接続は、原則としてインターネットを経由して行いますが、セキュアな接続が必要となるケースでは、AWS Direct ConnectまたはAWS Site-to-Site VPNを使用します。

Direct Connectは、安定した高速通信環境が必要とされる場合に利用しますが、応分のコストが必要になります。また、契約から実際に回線が開通して使用開始するまで、手続き期間を含めて時間を要するため、すぐに利用したい場合には向きません。

Site-to-Site VPNは、各メーカーのルーターに最適化された設定ファイルをAWSマネジメントコンソールからダウンロードし、その設定ファイルに基づいてオンプレミス環境のルーターを設定するだけで利用できるため、低コストで迅速に利用を開始できます。しかし、あくまでも回線はインターネットを経由するため、Direct Connectに比べると速度や安全性の点で劣ります。

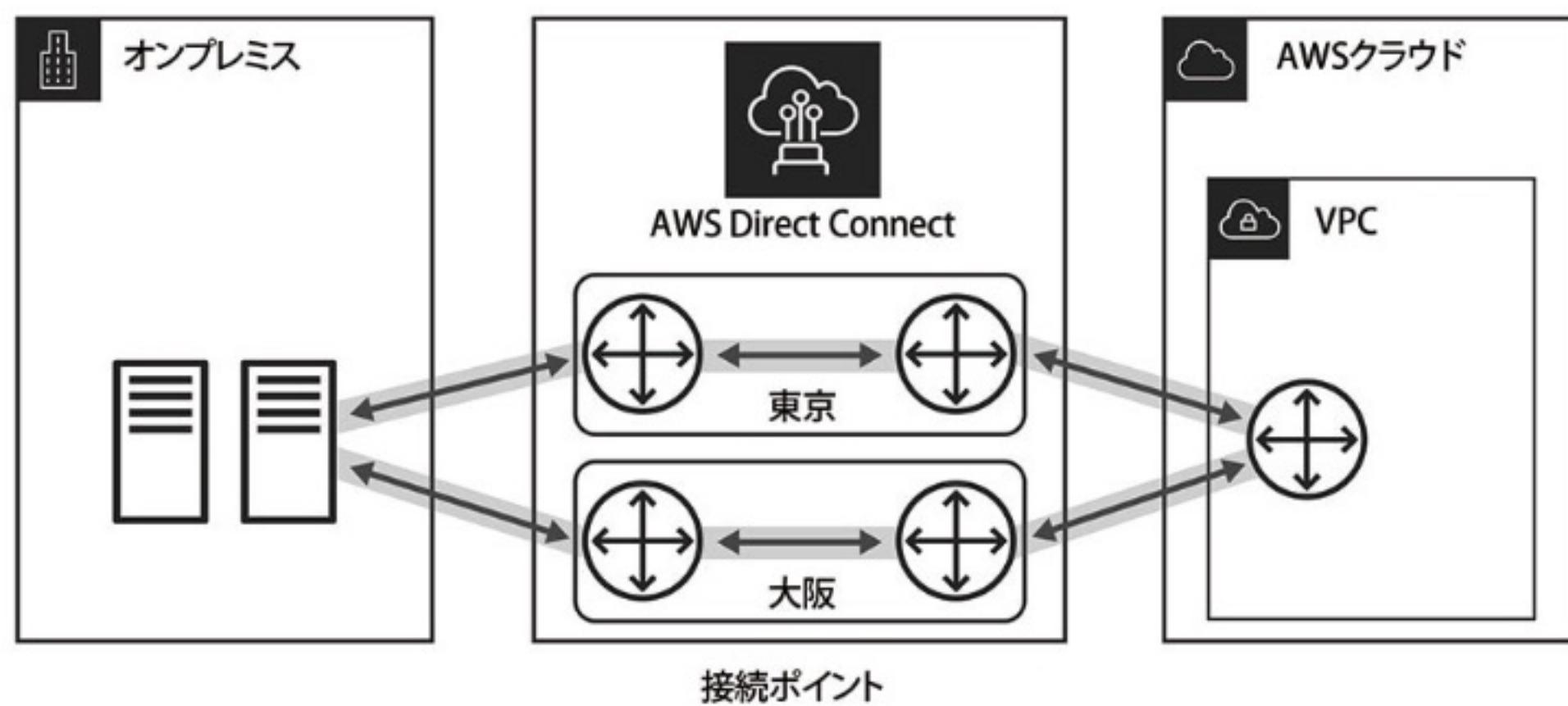
これら両者の特性を踏まえたうえで、可用性の高い回線環境を構築する例を次に示します。

●Direct Connect冗長化パターン

Direct Connectを2回線用意して冗長化することで、ネットワークの高可用性を実現します。

また、接続ポイントを東京・大阪のように別拠点にすることで、接続ポイントにおける障害にも対応することが可能ですが、回線にかかるコストは高くなります。

【Direct Connect冗長化の構成】

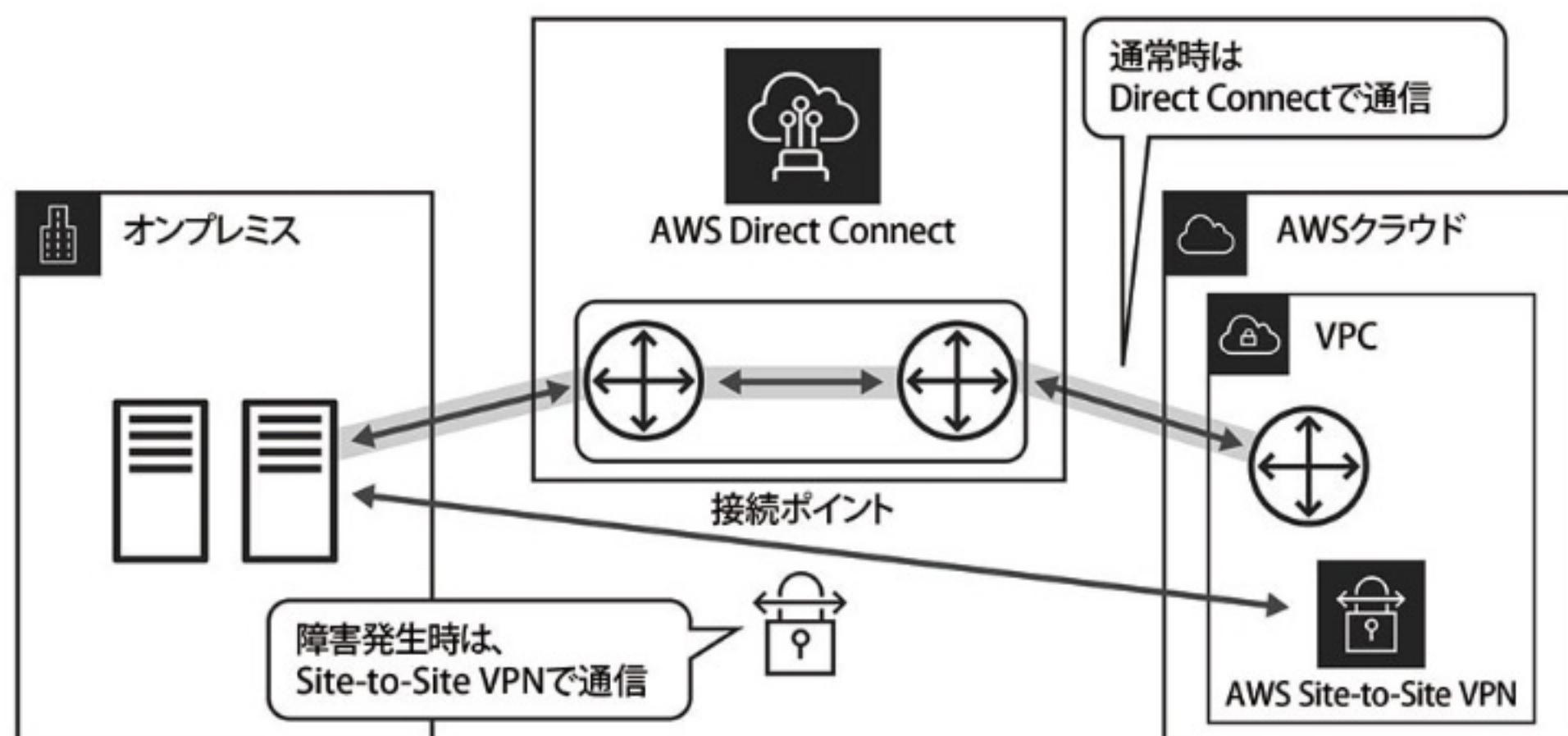


●Direct ConnectとSite-to-Site VPNの併用パターン

Direct Connect障害時のバックアップ回線としてSite-to-Site VPNを採用することで、ネットワークの可用性を確保できます。

ただし、Direct Connectに障害が発生してSite-to-Site VPNへフェイルオーバーした場合、通信品質や帯域が異なる回線に切り替わるため、パフォーマンスに影響が出る可能性があります。

【Direct ConnectとSite-to-Site VPNの構成】



以上、2パターンの高可用ネットワークの要点を、次の表にまとめます。

【高可用ネットワークの要点】

構成	メリット	デメリット
Direct Connect冗長化の構成	<ul style="list-style-type: none"> 地理的に離れた高可用性ネットワークを構築できる 同等のネットワークを2回線用意することで、切り替え前と同等の通信品質を切り替え後も担保できる 	・回線の運用コストが高くなる
Direct ConnectとSite-to-Site VPNの構成	<ul style="list-style-type: none"> 主回線の障害時でも、システムの稼働を最低限継続できる 片系をVPNにすることで、運用コストを抑えることができる 	・Direct ConnectからVPNへの切り替え後に通信品質が低下する



試験対策

オンプレミス環境とAWS間のネットワーク冗長化のパターンを覚えておきましょう。
試験問題の内容が、コストを重視しているのか通信品質を重視しているのかを読み解くことで、どのパターンを採用すべきかわかりやすくなります。



AWS Direct Connectには、占有型と共有型の2つの接続タイプがあります。
占有型は物理接続に対して契約するため、ユーザー側で自由に論理接続を作成することができます。共有型は物理接続をキャリアが保有しており、論理接続単位での契約が必要になります。

●VPC内リソースの可用性

オンプレミス環境では、ネットワーク機器やネットワークインターフェイスなど障害ポイントとなる箇所が多数存在しますが、AWS内の物理機器は責任共有モデルに従い、原則としてAWSで管理されています。

たとえば、インターネット接続に必要なインターネットゲートウェイ(IGW)は、AWS内部で透過的に冗長化されており、1つのゲートウェイを1カ所のVPCで利用するため、ある特定のAZで障害が発生したとしても、残りのAZで利用する通信には影響がありません。

一方で、NATゲートウェイはAZ内では冗長化されていますが、AZ間の冗長化はされていません。そのため、AZに障害が発生してもシステムの稼働を継続する構成を検討する場合は、NATゲートウェイをAZごとに配置するかどうかを考える必要があります。

VPC内で利用するリソースに関しては、VPCに対して利用するリソースなのか、任意のAZまたはサブネットに配置するリソースかを理解しておきましょう。