合格対策

AWS認定 ソリューションアーキテクト アソシエイト

大塚 康徳（日立インフォメーションアカデミー）著

リックテレコム

注　意

1. 本書は著者が独自に調査した結果を出版したものです。
2. 本書は万全を期して作成しましたが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、出版元まで書面にてご連絡ください。
3. 本書の内容に関して運用した結果の影響については、上記にかかわらず責任を負いかねますので、あらかじめご了承ください。
4. 本書の内容は、2016年6月の執筆時点のものです。本書で紹介したサービスの内容に関しては、将来予告なしに変更されることがあります。

商　標

はじめに

AWS認定プログラムは、必ずしも「普段からAWSをよく利用している人」が合格できる認定試験ではありません。なぜならば、AWS上に仮想サーバを起動し、オンプレミスと同様のシステムを構築することは驚くほど簡単にできてしまうのですが、認定試験で問われるのはオンプレミスと同じようにAWSを利用することではありません。認定試験で問われるのは「いかにAWSの各種サービスや特徴を活用し、可用性が高く、コスト効率に優れたシステムを設計・構築するか」ということだからです。

オンプレミスのシステムにほとんど変更を加えず、シンプルにAWSに移行し、安定稼働させることは可能です。ただし、その場合はオンプレミスでも実施していた可用性を担保するための冗長構成などをAWS上でも構築する必要があり、構築やその後の運用管理に多くの工数がかかります。また、サーバスペック／ストレージサイズもシステムの負荷に対して対応できるよう、オンプレミスと同程度をデプロイ（用意）すると、ITリソースのコストもさほど削減することができません。一方で、本書で解説するAWSの7つのベストプラクティスを順守し、AWSの各種サービスとその特徴を活かしてシステムを構築することで、可用性の高いシステムを低コストで構築／運用管理することができます。

AWSには「Design for Failure（障害に耐えうる設計）」という考え方があります。「個々のサーバなどに障害が発生してもシステムに問題が生じないようにシステムを設計しましょう」ということをサービス提供側であるAWSが言っているのです。「なぜ個々のサーバなどの可用性をより高めることをしないのか？」と思われる方もいるかもしれませんが、「形あるものはいつかは壊れる」と言われるように、世の中、壊れないものはありません。ならば、壊れても問題ないようにシステムを設計／構築することこそ、究極の可用性なのではないでしょうか。オンプレミスの限られたリソースの中で、そのようなシステムを実現することは難しいかもしれませんが、AWSの持つリソース／各種サービスとその特徴を活かすことでオンプレミスでは実現困難なシステムの構築が可能になります。

AWS 認定プログラムの 1 つである、AWS 認定ソリューションアーキテクト－アソシエイトはそのエッセンスを問う認定プログラムです。是非、この認定を取得して AWS 上のシステムの設計、デプロイ、管理に必要なスキルと技術知識を有する IT プロフェッショナルであることを証明するとともに、AWS を最大限に活用するためのエッセンスを習得しましょう。

2016 年 7 月　大塚 康徳

目次

第1章 AWS と認定プログラム

AWS クラウドとは何か、そして認定プログラムとは何か？

AWS クラウドとは、Amazon Web Services, Inc. が提供するクラウドサービスです。そのサービスは多岐にわたり、日々新しい機能やサービスが生まれています。そして、AWS 認定プログラムは、この AWS クラウドに関する知識とスキルを有していることを認定するものです。

本章では、まず AWS クラウドと認定制度の概要について解説します。

1-1 AWS(Amazon Web Services) クラウド

サーバやストレージといった IT リソースを利用者が所有せず、必要なときにネットワークを介して使いたいだけ使い、必要がなくなれば解放するという IT リソースの利用形態をクラウドコンピューティング（以下**クラウド**）といいます。利用者が IT リソースを所有する**オンプレミス型**の運用形態と異なり、クラウドでは、利用にかかる料金は使った分だけ（従量課金）という課金体系が多く、利用者は IT リソースの購入・維持・管理費用などを抑えることができます。

AWS クラウド（以下 AWS）とは、Amazon Web Services, Inc. が提供するクラウドサービスで、ネットワークやサーバ、ストレージといったインフラストラクチャからアプリケーションまで、様々な IT リソースを利用者がセルフサービスで利用できます。その特徴としては、IT リソースの柔軟かつ俊敏な伸縮自在性が挙げられます。必要なときに使いたいだけ IT リソースを使い、必要がなくなれば解放するというのがクラウドの特徴ですが、それに加えて AWS では、システムの負荷などの利用状況に応じて、サーバ台数やストレージ容量を増やしたり、減らしたりといったことを利用者が自動・手動でリアルタイムに行えるようになっています。

AWS 上にシステムを構築する上で、オンプレミスでの構築とは異なる、AWS ならではの **7 つのベストプラクティス**（推奨される設計時の考慮点）があります。

1. 故障に備えた設計で障害を回避
2. コンポーネント間を疎結合で柔軟に
3. 伸縮自在性を実装
4. すべての層でセキュリティを強化
5. 制約を恐れない（IT リソース量の制限などオンプレミスとは考え方を変える）
6. 処理の並列化を考慮
7. さまざまなストレージの選択肢を活用

例えば、「1．故障に備えた設計で障害を回避」はオンプレミスと大きく異なるポイントの 1 つで、オンプレミスのシステムではインフラ側でサーバがダウンせず継続して使えるようにする、いわゆる**可用性**を実現しようとしますが、AWS では構成設計側（アプリ側）で高い信頼性・可用性を実現しようとします。そのポイントを押さえることで、AWS 上で高信頼・高可用なシステムを構築できるため、AWS を利用する上では極めて重要なポイントとなります。

これらの 7 つのベストプラクティスは、AWS 認定プログラムでもよく問われるポイントになっています。詳細は、後の章で解説します。

1-2 AWS 認定プログラム

AWS 認定プログラムは、AWS に関する知識とスキルを有していることを認定するものです（https://aws.amazon.com/jp/certification/）。AWS 導入にあたるエンジニアの役割を「ソリューションアーキテクト」「デベロッパー」「システムオペレーション（SysOps）アドミニストレーター」という 3 つの役割に分類し、その役割ごとに認定資格があります。各認定資格には、習熟度によって「アソシエイト」「プロフェッショナル」の 2 つのレベルがあります（**図 1-2-1**）。

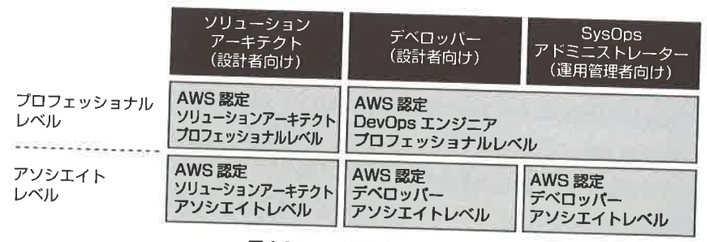

図 1-2-1　AWS 認定資格体系

AWS 認定プログラムは、AWS を使用してセキュアで信頼性のあるクラウドベースのアプリケーションを構築するためのベストプラクティスに関するスキルと知識を認証することを目的にしています。AWS 認定資格は 2 年ごとに

更新する必要があり、アソシエイトレベルの方は、再認定試験を 2 年ごとに更新するか、またはその上位のプロフェッショナルレベルを取得する必要があります。

本書は、ソリューションアーキテクトーアソシエイトレベルを対象としており、その概要と出題割合は次のとおりになります。

● 概要

AWS 認定ソリューションアーキテクトーアソシエイトレベル資格は、認定者の次の能力を認定するものです。

- システム要件の洗い出しと定義能力を有し、AWS アーキテクチャーのベストプラクティスに基づき AWS 上にシステムを構築することができる
- AWS アーキテクチャーのベストプラクティスを、アプリケーション開発者およびシステム管理者に対してプロジェクトのライフサイクルを通じて助言できる

● 出題分野と割合

分野	出題割合
高可用性、コスト効率、対障害性、スケーラブルなシステムの設計	60%
実装／デプロイ	10%
データセキュリティ	20%
トラブルシューティング	10%

試験時間：80 分
問題数：非公表
回答方法：択一選択／複数選択
合格ライン：非公表

第 2 章

リージョン／アベイラビリティーゾーンと AWS サービス

リージョンと AZ、そして各種サービスの提供レベルについて

AWS を利用する上で、必ず押さえなくてはいけない概念（用語）が、「リージョン」と「アベイラビリティーゾーン」です。認定試験においても、リージョンとアベイラビリティーゾーンについて正しく理解していなければ、ほとんどの問題を解くことができません。本章では、まず、リージョンとアベイラビリティーゾーンについて概要を説明し、その後に主要なサービスの提供レベルについて説明します。

2-1 リージョンとアベイラビリティーゾーン

重要！

アベイラビリティーゾーン：物理的なデータセンタ群で AZ と略される
リージョン：複数の AZ が存在する世界 13ヶ所の地域

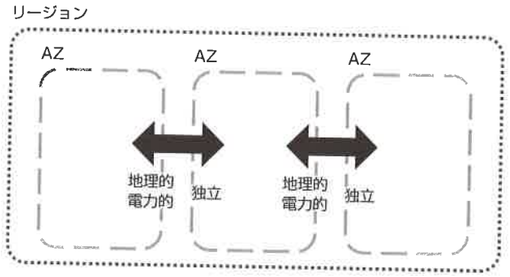

図 2-1-1　リージョンと AZ

2016 年 7 月現在、AWS は世界中に 13ヶ所[注1]の**リージョン**（Region）を用意しており、2017 年までにさらに 4ヶ所のリージョンを追加する予定です。1 つのリージョンの中には複数の**アベイラビリティーゾーン**（Availability Zone：以下 **AZ**）があり、それぞれの AZ は地理的にも電力的にも独立しています。そのため、落雷によるサージ電流や大雨による浸水などでリージョン内の AZ の 1 つに避けられない障害が発生したとしても、他の AZ には影響が及ばないように設計されています（**図 2-1-1**）。これにより、サーバやデータを AZ 間で冗長的に配置することで、可用性の高いシステムを構築することができ、これは 1 章 1-1 で説明した“7 つのベストプラクティス”の 1 つである「故障に備えた設計で障害を回避」にも関係する、非常に重要なポイントです（**図 2-1-2**）。各リージョン内の AZ 間は専用線で接続されており、各 AZ に配置されたサーバ間は低レイテンシー（遅延）で通信することができます。

注 1　13ヶ所のうち、2ヶ所（GovCloud と北京）は通常の利用者は使えない特殊なリージョンになります。

試験のポイント！

サーバやデータは AZ 間で冗長的に配置する

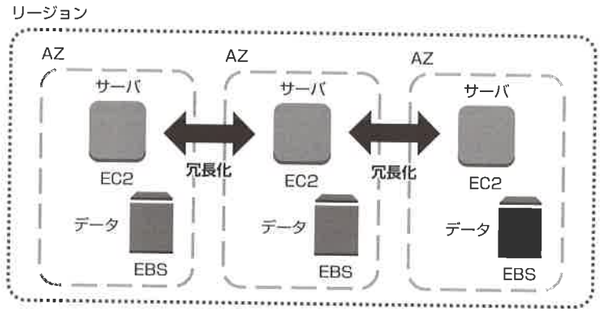

図 2-1-2　AZ にまたがるサーバ配置

補足　Amazon Route 53 という DNS サービスを利用し、複数のリージョンにまたがってサーバやデータを配置することで、より高可用なシステムを構築できます。詳しくは 10 章で説明します。

2-2 AWS サービスとリージョン／AZ

AWS には非常に数多くのサービスがあります。そして、これらのサービスを組み合わせて使用することにより、システムを構築することができます。各サービスにはそれぞれ対応するシンプルなアイコン[注2]が用意されており（**図 2-2-1**）、それらのアイコンを配置してシステム構成図を作成することができます。

注 2　「シンプルアイコン」と呼ばれています。

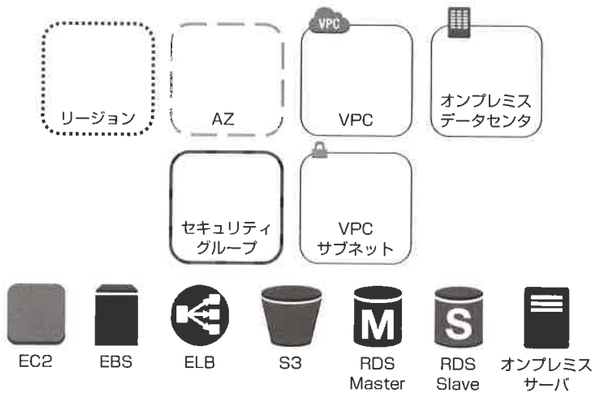

図 2-2-1　主要なシンプルアイコン

例えば、仮想サーバのサービスである Amazon EC2（以下 EC2）インスタンス[注3]を負荷分散サービスである Elastic Load Balancing（以下 ELB）の配下に配置して、トラフィックを分散させるシステムを構築するとします。EC2 インスタンスのバックエンドにはリレーショナルデータベースのサービスである Amazon RDS（以下 RDS）インスタンスを配置してデータベースを管理し、静的なデータや大きな読み取りデータは耐久性の高いストレージである Amazon S3（以下 S3）バケット[注4]に保存するようにします。これらのシステム構成についてシンプルアイコンを使ってまとめると、**図 2-2-2** に示したようになります。

注 3　EC2 や RDS などはサービス名であり、仮想サーバ（マシン）1 台 1 台の実体はインスタンスと呼ばれます（EC2 インスタンスや RDS インスタンス）。

注 4　リージョンごとに作成するデータの格納先で、詳しくは 6 章で紹介します。

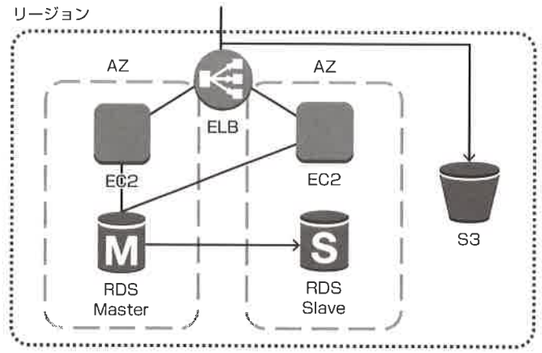

図 2-2-2　シンプルアイコンを用いたシステム構成図

AWS の各種サービスには、次の 3 つのサービスレベルがあります。

- リージョンごとに作成・管理される**リージョンサービス**
- AZ ごとに作成・管理される **AZ サービス**
- どこのリージョンからでも共通のサービスとして利用できる**グローバルサービス**

AWS 上のほとんどシステムがこれらのサービスを組み合わせてシステムを構成しますが、組み合わせる際には注意が必要です。1 つのリージョン内の AZ サービス間であればプライベート IP アドレスで接続できますが、リージョンサービスの場合、基本的にはグローバル IP アドレスで接続しなければいけません。このようなサービスレベルの違いは、システムを設計する上で意識しなければいけない重要事項です。代表的なリージョンサービスと AZ サービス、そしてグローバルサービスを**表 2-2-1** 及び**図 2-2-3** に示します。

表 2-2-1　代表的な AWS サービスと概要

	サービス名	サービスの概要
リージョンサービス	Amazon S3	ストレージ
	Amazon DynamoDB	NoSQL
	Amazon SQS	キュー
	Amazon CloudSearch	検索
AZ サービス	Amazon EC2	仮想マシン
	Amazon RDS	リレーショナル DB
	ELB	負荷分散
	Amazon ElastiCache	キャッシュ
グローバルサービス	AWS IAM	認証・アクセス制限
	Amazon Route 53	DNS
	Amazon CloudFront	コンテンツ配信

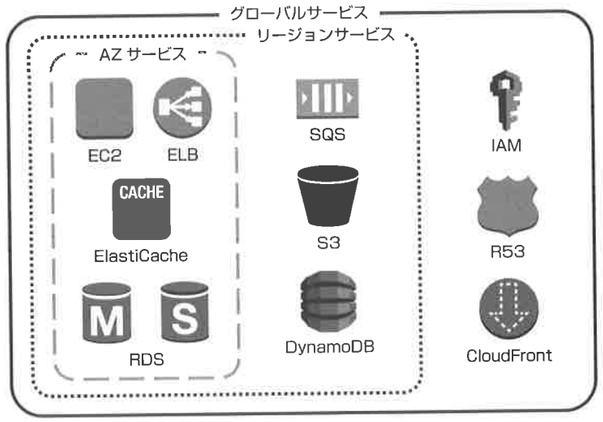

図 2-2-3　サービスレベル

章末問題

Q1 次の構成のうち、最も可用性が高くなる構成はどれか？

- ○ **A** 1 つのリージョン内の 1 つの AZ に 4 台の Web サーバ (EC2) を配置し、ELB を用いて負荷分散する
- ○ **B** 1 つのリージョン内の 2 つの AZ に各 2 台の Web サーバ (EC2) を配置し、ELB を用いて負荷分散する
- ○ **C** 2 つのリージョン内の各 1 つの AZ に 2 台ずつ Web サーバ (EC2) を配置し、ELB を用いて負荷分散する
- ○ **D** 2 つのリージョン内の各 2 つの AZ に 1 台ずつ Web サーバ (EC2) を配置し、ELB を用いて負荷分散する

答え

A1 B

ELB は AZ サービスであり、リージョンをまたいで負荷分散することはできません。複数の AZ に EC2 を配置し、負荷分散する B の構成が最も可用性が高くなります。

第 3 章

責任分担セキュリティモデルと AWS における認証（IAM）

AWS におけるセキュリティの考え方と、認証について

利用者と AWS が協力してセキュリティを高める考え方を**責任分担セキュリティモデル**といいます。また、AWS の各種サービスを利用する上での認証とアクセス制御を提供する AWS サービスを IAM（Identity and Access Management）といいます。IAM は、AWS を利用する上でまず初めに理解しなくてはいけないサービスであり、認定試験においても、セキュリティに関する出題は 20% を占めます。本章では、AWS を利用する上でのセキュリティの考え方と IAM について説明します。

3-1 責任分担セキュリティモデル

AWS 上のシステムを不正アクセスなどから保護するには、利用者と AWS が協力してセキュリティを高める必要があります。この考え方を AWS では**責任分担（共有）セキュリティモデル**と呼んでいます。

重要！

AWS 上のシステムは、利用者と AWS が責任を共有してセキュリティを高める

利用者と AWS の間で、システムのどの階層について責任を分担するかは、利用する AWS サービスによって異なります。例えば、EC2 のようなハードウェア部分まで AWS が管理する**インフラストラクチャサービス**の場合は、図 3-1-1 のような分担になります。

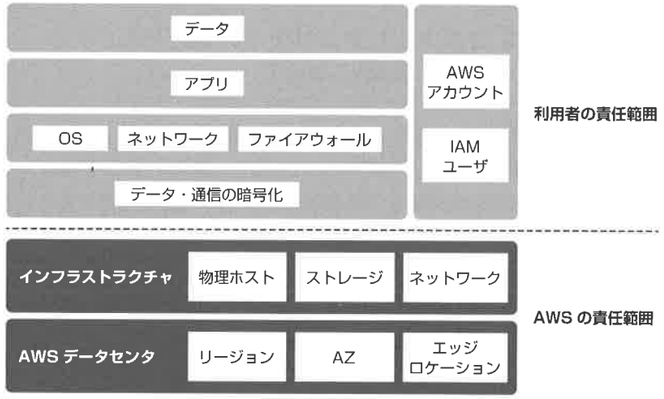

図 3-1-1　インフラストラクチャサービスの責任分担セキュリティモデル

一方、RDS のようなハードウェア部分から OS やミドルウェア部分まで AWS が管理する**コンテナサービス**の場合は、図 3-1-2 のような分担になります。

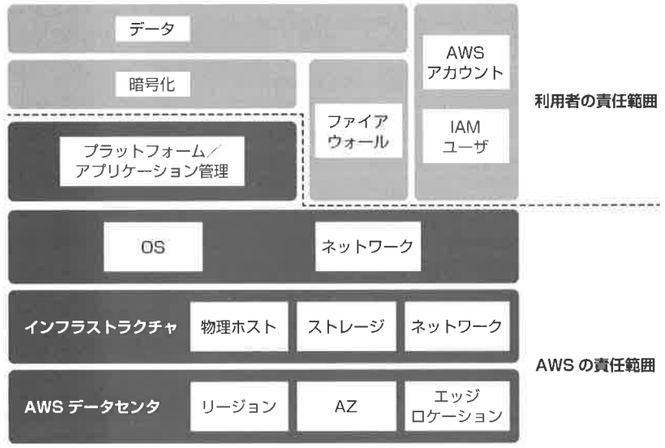

図 3-1-2　コンテナサービスの責任分担セキュリティモデル

S3 や DynamoDB のようなハードウェア部分からソフトウェア部分まで AWS が管理する**アブストラクトサービス**の場合は、図 3-1-3 のような分担になります。

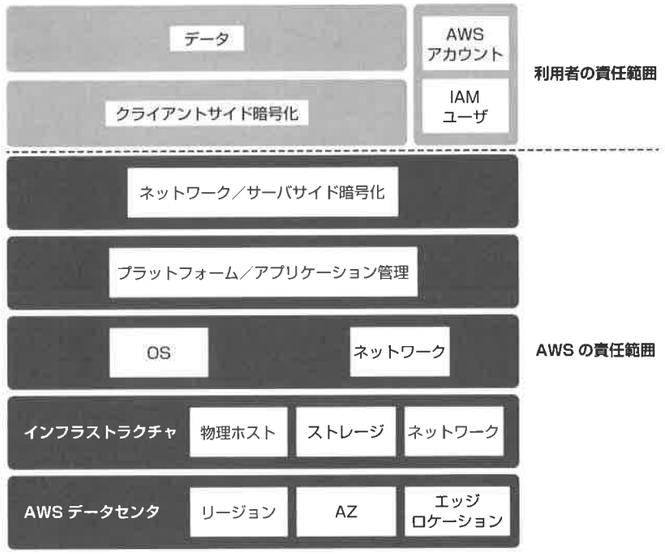

図 3-1-3　アブストラクトサービス責任分担セキュリティモデル

インフラストラクチャサービスである EC2 については、OS のファイアウォールを利用する他、セキュリティ対策ソフトを導入するなどして、利用者の責任でセキュリティ上の脅威からシステムを守る必要があります。ただし、セキュリティ対策後のテストについては注意が必要で、侵入テストなどを行う際には AWS に事前申請する必要があり、申請せずに実施すると利用規約違反となります。

試験のポイント！

インフラストラクチャサービス、コンテナサービス、アブストラクトサービスの各サービスについて、利用者の責任範囲を明確にする

3-2 AWS における認証とアクセス制御（IAM）

AWS を利用するには、アカウントを取得する必要があります。メールアドレス・パスワードなどのログイン情報、名前・住所・電話番号などの連絡先情報、クレジットカード情報などの支払い情報を登録すると、12 桁のアカウント番号が発行され、アカウントを取得できます。登録したメールアドレスとパスワードでマネージメントコンソールにサインインすると、EC2 を始めとした様々な AWS サービスを利用することができます。このメールアドレスのことを**ルートアカウント**と呼び、ルートアカウントではすべての操作を行うことができます。ただし、ルートアカウントの権限は制御することができないため、操作ミスやパスワード漏えいに備え、日常の操作にはルートアカウントを使用せず、アカウント内に**ユーザ**を作成し、このユーザを使用します。ユーザは、1 アカウント内に複数作成することができます。

試験のポイント！

日常の操作にはルートアカウントを使用せず、ユーザを使用する

AWS 内のユーザ管理や AWS リソースに対するアクセス制御を行うためのサービスを AWS Identity and Access Management（以下 **IAM**）といいます。AWS アカウントを取得したら、まず、IAM でグループ（以下 **IAM グループ**）やユーザ（以下 **IAM ユーザ**）を作成します。そして、各 IAM グループや IAM ユーザごとに、AWS の各種リソースに対するアクセスの可否（**IAM ポリシー**）を設定します。

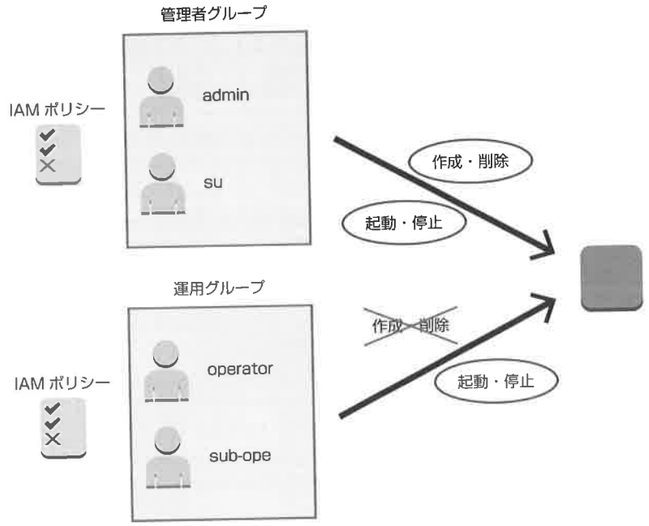

図 3-2-1　IAM グループ／ユーザ／ポリシー

新たに作成した IAM グループや IAM ユーザには何の権限も与えられていないため、アクセス権限を割り当てていきますが、その際には必要最低限のアクセス権限を割り当てるようにします。アクセス許可と拒否の IAM ポリシーが相反する場合、拒否の IAM ポリシーが優先されます。

試験のポイント！

各 IAM グループ・IAM ユーザには、最小権限のアクセス権を与える。
IAM ポリシーは最も厳しいポリシー（拒否）が優先される。

「EC2 インスタンスを起動する」「S3 バケットにファイルをアップロードする」といった AWS サービスを利用（操作）する方法には、表 3-2-1 に示した 3 種類があり、どの方法を利用するにも認証が必要です。

表 3-2-1　AWS サービスを操作する方法と認証情報

利用（操作）方法	認証情報
マネージメントコンソール（Web ブラウザ）	ユーザ名／パスワード
AWS CLI（コマンド）	アクセスキー／シークレットアクセスキー
AWS SDK（プログラム）	アクセスキー／シークレットアクセスキー

各 IAM ユーザは「アクセスキー」と「シークレットアクセスキー」のペアを作成・保持することができ、ユーザ ID とパスワードのように、そのペアを AWS CLI（コマンド）や AWS SDK（プログラム）の認証情報として利用することができます。

アクセスキーの例：AKIABCDEFGHIJKLMNOPQ
シークレットキーの例：zyxwvutsrqponmlkjihgfedcba123456789ABCDE

環境変数や認証ファイルにアクセスキーとシークレットキーの値を格納しておくと、コマンドラインで次のようなコマンドを実行できます。

図 3-2-2　実行例　aws s3 ls の例

```
$ aws s3 ls

2015-12-15  09:28:47  my-bucket1
2016-01-07  16:53:12  my-bucket2
```

アクセスキーとシークレットアクセスキーは、SDK（プログラム）の認証情報として利用することができますが、認証情報の更新の問題や流出の危険性などから推奨されていません。その代わりに、**IAM ロール**の利用が推奨されています。IAM ロールには、IAM グループや IAM ユーザと同様に、AWS の各種リソースに対するアクセス可否（IAM ポリシー）を設定します。IAM ロールは、EC2 インスタンスなどに割り当てることができ、IAM ロールを割り当てられた EC2 インスタンス上のプログラムは、アクセスキーとシークレットキーがなくとも IAM ロールに許可されている AWS のリソースにアクセスできます。

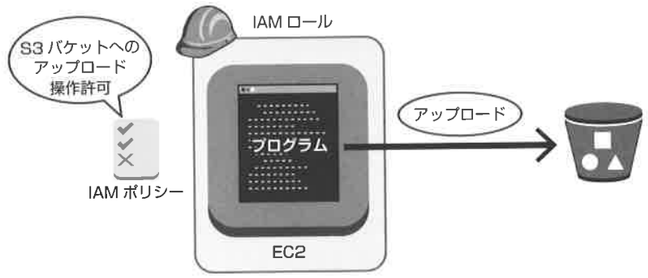

図 3-2-3　IAM ロール

試験のポイント！

EC2 インスタンス上で実行されるプログラムの認証には IAM ロールを割り当てる。

3-3 ID フェデレーション

自社の従業員に、各自の業務レポートを毎月末に S3 バケットにアップロードさせるために、S3 バケット（AWS サービス）へのアクセス権を付与したいという要望があったとします。その際、各従業員の毎月 1 回だけの S3 バケットへのアクセスのために、従業員一人一人を IAM ユーザとして登録するのはたいへん非効率です。AWS には Security Token Service（以下 STS）という一時的に認証情報を付与するサービスがあり、その STS と ID ブローカー（ID プロバイダー）を利用して、自社の認証基盤で認証が通れば、そのユーザから S3 バケット（AWS サービス）へのアップロードを一時的に許可することができます。

これを **ID フェデレーション**と呼びます。

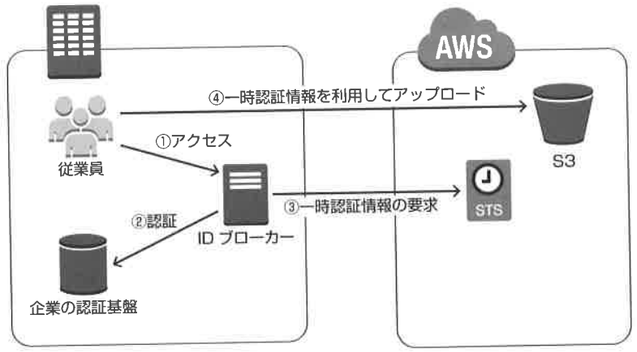

図 3-3-1　ID ブローカーを利用したフェデレーション

① ユーザが社内の ID ブローカーにアクセス
② ID ブローカーが社内の ID ストア（Active Directory や LDAP）でユーザ認証
③ ID ブローカーが STS から一時的な認証情報を取得する
④ 一時的な認証情報を使ってユーザが S3 バケットにファイルをアップロード

AWS では、図 3-3-1 のように ID ブローカーを使用する他、Security Assertion Markup Language（SAML）を使用したシングルサインオンや、Google や Facebook といったウェブ ID プロバイダーを使用したシングルサインオンなどにも対応しています。

試験のポイント！

AWS の使用頻度が低いユーザは、ID フェデレーションで社内の認証基盤と IAM を連携する。

章末問題

Q1 次のうち、利用者の責任で実施しなければいけないセキュリティ対策はどれか？2 つ選べ。

- □ **A** EC2 インスタンスの物理ホスト上のハイパーバイザのセキュリティパッチの適用
- □ **B** S3 上のデータの暗号化
- □ **C** 物理ディスクの適切な廃棄
- □ **D** EC2 インスタンス上の OS のセキュリティパッチの適用

Q2 AWS アカウント／認証情報の推奨される運用はどれか？

- ○ **A** ルートアカウントには複雑なパスワードを割り当てて、定期的に更新しながら利用する
- ○ **B** S3 バケットへのファイルのアップロードを行うプログラムのソースコードに、S3 バケットへのファイルアップロードが許可された IAM ユーザのアクセスキーとシークレットアクセスキーを記載する
- ○ **C** S3 バケットへのファイルのアップロードを行うプログラムを EC2 インスタンスで実行する場合、S3 バケットへのファイルアップロードが許可された IAM ロールを EC2 インスタンスに割り当てて EC2 インスタンスを作成する
- ○ **D** AWS の使用頻度に関わらず、全従業員（約 1,000 人）をそれぞれ IAM ユーザとして登録し、各自のアクセスキーとシークレットキーを用いて AWS を利用させる

答え

A1 B, D

インフラストラクチャサービスである EC2 は、OS 以上が利用者の責任です。

A2 C

ルートアカウントは使用しないことが推奨されています。
使用頻度が低いユーザについては、ID ブローカーなどを利用して社内の認証基盤と IAM を統合します。

第 4 章

AWS におけるネットワーク（VPC）

AWS におけるネットワークについて

AWS でネットワーク環境を提供しているサービスを VPC といいます。AWS の各種サービスを利用する上で、VPC は欠かすことのできないサービスであり、認定試験においても、全ての分野にまたがって出題されます。本章では、AWS を利用する上でのネットワーク（VPC）について説明します。

4-1 VPC の機能と設定

EC2 インスタンスに AWS 外からアクセスできるようにするには、EC2 インスタンスに IP アドレスが適切に割り振られ、外部ネットワークから EC2 インスタンスに到達できるよう適切にルーティングされていなければいけません。AWS でこのようなネットワーク環境を提供しているサービスを Amazon Virtual Private Cloud（以下 **VPC**）といいます。VPC はその名の通り、AWS 上に利用者ごとのプライベートなネットワーク空間を提供し、インターネットやオンプレミスのイントラネットなどの外部ネットワークと接続できます。

3 つの AZ からなるリージョンに、インターネットからアクセスされる Web サーバの EC2 インスタンスと、その Web サーバからアクセスされる DB の EC2 インスタンス、そしてオンプレミスの基幹システムにアクセスする必要がある EC2 インスタンスからなるシステムを AWS 上に構築するとします。そのようなシステムの VPC を構築するステップは、次の通りです。

① VPC（プライベートネットワーク空間）の作成

ある特定のリージョンを選択してプライベートネットワーク空間を作成します。プライベートネットワーク空間は /16 から /28 の CIDR ブロック範囲で作成でき、このプライベートネットワーク空間自体も VPC と呼びます。ネットワークアドレスは、一般的に次のようなクラス A から C のプライベートネットワークのいずれかの値を使用します。

クラス A：10.0.0.0〜10.255.255.255
クラス B：172.16.0.0〜172.31.255.255
クラス C：192.168.0.0〜192.168.255.255

ここでは、「10.100.0.0/16」の範囲の VPC を作成する想定で進めます（図 4-1-1）。

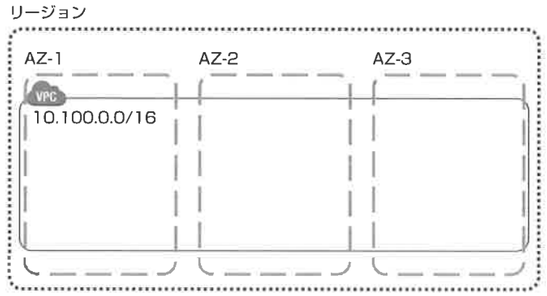

図 4-1-1　「10.100.0.0/16」VPC の作成

② サブネットの作成

①で作成した VPC の中にサブネットを作成します。サブネットは、その中に配置するサーバの役割（機能）に応じて作成するのが一般的です。また、サブネットは複数の AZ にまたがって作成することはできないので、必ずある 1 つの AZ を指定して作成します。このとき、AWS の 7 つのベストプラクティスの 1 つ「故障に備えた設計で障害を回避」を実践するため、同じ役割のサブネットを複数の AZ に作成し、サーバを各 AZ に冗長的に配置するのが一般的ですが、ここでは便宜的に各 AZ に役割ごとに 1 つずつ作成することにします。図 4-1-2 のように、AZ-1 に DB 用のサブネット（10.100.1.0/24）、AZ-2 に Web サーバ用のサブネット（10.100.2.0/24）、AZ-3 に基幹システムにアクセスするサーバ用のサブネット（10.100.3.0/24）を作成します。

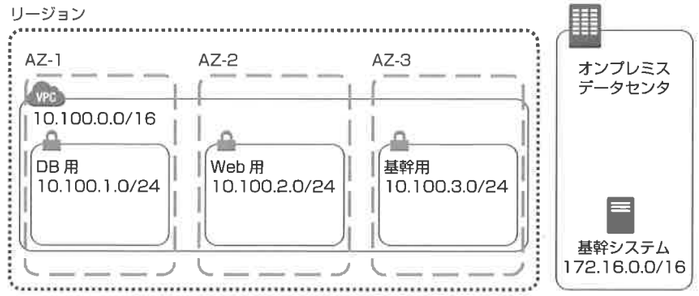

図 4-1-2　サブネットの作成

重要！

サブネットは AZ をまたがることができない。サブネットを選択することは AZ を選択することと同じ。

③ ゲートウェイの作成

図 4-1-3 のように、VPC と外部ネットワークの間で通信を行うための出入口となるゲートウェイを作成し、VPC にアタッチします。インターネットとの出入口になるゲートウェイをインターネットゲートウェイ（以下 **IGW**）、オンプレミスと VPN や専用線で通信するための出入口となるゲートウェイをバーチャルプライベートゲートウェイ（以下 **VGW**）といいます。

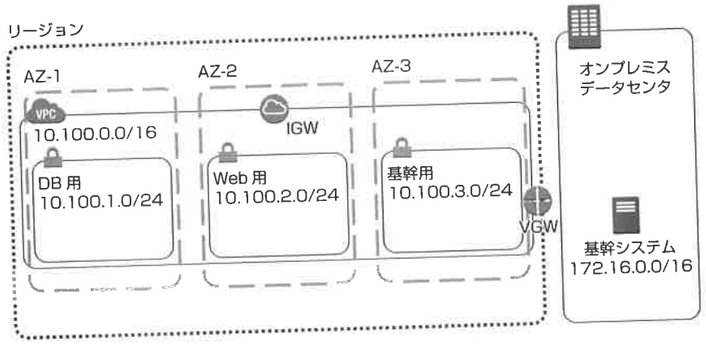

図 4-1-3　ゲートウェイのアタッチ

④ ルートテーブルの設定

サブネットを作成する際、中に配置するサーバの役割に応じて作成したのは、サブネットごとに、インターネットとのアクセスを許可する／しないのアクセス制限をかけることができるためです。サブネットのアクセス制限は、AWS 上のシステムのセキュリティを守る重要な要素の 1 つです。AWS では、これらのサブネットをそれぞれ次のように呼びます。

重要！

インターネットとのアクセスを許可するサブネット：パブリックサブネット
インターネットとのアクセスを許可しないサブネット：プライベートサブネット

各サブネットがパブリックサブネットなのか、あるいはプライベートサブネットなのかは、そのサブネットに適用されている**ルートテーブル**によって決まります。デフォルトゲートウェイ（送信先：0.0.0.0/0）のターゲットとして IGW が設定されたルートテーブルがサブネットに適用されていれば、そのサブネットはパブリックサブネットです。一方、デフォルトゲートウェイ（送信先：0.0.0.0/0）のターゲットとして IGW が設定されていないルートテーブルがサブネットに適用されていれば、そのサブネットはプライベートサブネットです。ここでは、図 4-1-4 のように、DB 用のサブネットはデフォルトのままのプライベートサブネットとし、Web サーバ用のサブネットをパブリックサブネットに設定します。また、オンプレミスの基幹システムと通信するサーバが配置されるサブネットもプライベートサブネットとし、オンプレミスのデータセンタへのルーティングルールを設定します。

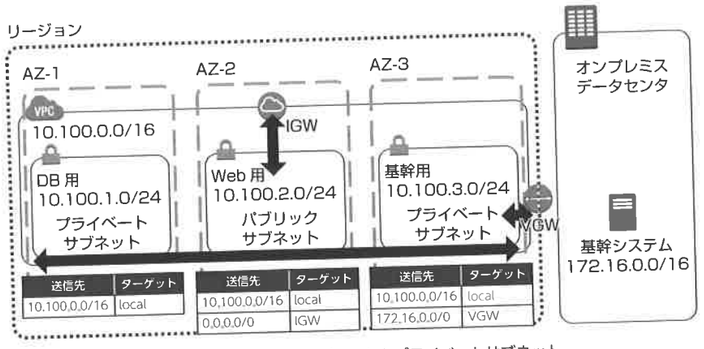

図 4-1-4　パブリックサブネットとプライベートサブネット

ルートテーブル内の「10.100.0.0/16　local」という設定は、デフォルトの設定で変更することも削除することもできません。このデフォルト設定

が意味するところは、VPC 内の通信はルートテーブルでは制御できないということです。通常のネットワークでは、サブネットでネットワークを区切ってしまえば、ルータがルーティングしない限り、サブネット間の通信は発生しません。ところが、VPC の場合は同じ VPC 内のサブネットであればサブネット間の通信が可能になっています。

⑤ NAT インスタンスの作成

サブネットをプライベートサブネットとして作成すれば、インターネットからのアクセスを受け付けないため、その中に配置するサーバのセキュリティレベルを高めることができます。一方で、プライベートサブネット内に配置したサーバがパッチのダウンロードのためにインターネットにアクセスしたい場合や、リージョンサービスである DynamoDB にアクセスしたい場合、デフォルトのルートテーブルの設定ではアクセスすることができません。このような場合は、**NAT**（Network Address Translation）**インスタンス**と呼ばれるインスタンスを利用することで、インターネットからはアクセスを受け付けないまま、プライベートサブネット内からインターネットやリージョンサービスにアクセスさせることができます。

NAT インスタンスの実体は EC2 インスタンスで、プライベートサブネット内の EC2 インスタンスからのトラフィックを受け付け、その EC2 インスタンスのプライベート IP アドレスを NAT インスタンスに割り振られたグローバル IP アドレスに変換し、インターネットへのアクセスを可能にします。ただし、EC2 インスタンスはデフォルトで、流れてきたトラフィックを自身の IP アドレス宛てかどうかをチェックし、宛先が自身の IP アドレスでなければトラフィックを破棄する設定になっています。この機能を**「送信元／送信先チェック」**といい、NAT インスタンスとして利用するためには、この機能を無効化する必要があります。

NAT インスタンスをパブリックサブネットに作成できたら、プライベートサブネットに適用しているルートテーブルのデフォルトゲートウェイ（送信先：0.0.0.0/0）のターゲットとして、NAT インスタンスを指定します（図 4-1-5）。こうして、プライベートサブネット内の EC2 インスタンスでもインターネットやリージョンサービスにアクセスが可能になります。

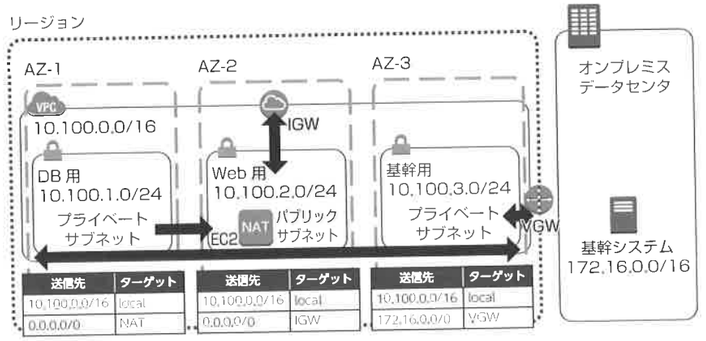

図 4-1-5　プライベートサブネットのルートテーブル

試験のポイント！

プライベートサブネット内のインスタンスがインターネットにアクセスするための設定を押さえる！

補足　2015 年 12 月に NAT ゲートウェイという AWS のマネージド型の NAT サービスが利用できるようになりましたが、本書では割愛します。

4-2 EC2 インスタンスの IP アドレス

2013 年 12 月 4 日以降に AWS アカウントを取得した場合、そのアカウントの EC2 インスタンスは必ず VPC のサブネット内で起動します。2013 年 12 月以前から AWS アカウントを取得している場合は、EC2-Classic という環境で EC2 インスタンスを起動できます。現在の AWS のネットワーク環境のデフォルトは VPC 環境であり、EC2-Classic から様々な機能拡張がされていることもあって、2013 年 12 月以前から AWS をお使いの方も VPC 環境を利用することが推奨されています。

VPC のサブネット内に起動する EC2 インスタンスには、そのサブネット内のプライベート IP アドレスが少なくとも 1 つ割り振られます。その EC2 イン

スタンスにインターネットからアクセスしたい場合、さらにグローバル IP アドレスを割り振る必要があります。AWS のグローバル IP アドレスには、次の 2 種類があります。

Public IP： EC2 インスタンスが起動した際にランダムに割り振られる動的なグローバル IP アドレス

Elastic IP： アカウントに割り当てられる固定のグローバル IP アドレスで、EC2 インスタンスにアタッチ／デタッチが可能

Public IP は EC2 インスタンスを停止して起動した場合にもランダムに変更されるため、固定のグローバル IP アドレスでの運用が求められる場合には Elastic IP を使用します。Elastic IP はそのアドレスを明示的に解放するまで、アカウントで保持されます。

EC2 インスタンスのプライベート IP アドレスとグローバル IP アドレスの紐付けは VPC の仮想ネットワークで行われているので、EC2 インスタンスの OS にログインし ipconfig コマンド（Windows）や ifconfig コマンド（Linux）を実行しても、プライベート IP アドレスの値しか表示されません。

4-3 セキュリティグループとネットワーク ACL

VPC が提供するファイアウォール機能に**セキュリティグループ**と**ネットワーク ACL（NACL）**があります。

セキュリティグループは、EC2 や ELB、RDS などインスタンスごとのファイアウォールで、受信（以下インバウンド）と送信（以下アウトバウンド）のアクセス制御ができます。各インスタンスには少なくとも 1 つのセキュリティグループを適用する必要があります。インバウンドでは、送信元の IP アドレスと、アクセスを受け付けるポート番号へのアクセスを許可します。アウトバウンドでは、送信先の IP アドレスと、アクセス先のポート番号へのアクセスを許可します。デフォルトでインバウンドは許可されているルールがないため、どこからのアクセスも受け付けません。一方、アウトバウンドは、デフォルトで全ての宛先／ポート番号に対するアクセスを許可するルールが設定されています。

ネットワーク ACL は、サブネットごとのファイアウォールで、セキュリティグループと同様にインバウンドとアウトバウンドのアクセス制御ができます。指定した送信元／送信先の IP アドレスとポート番号のアクセスを許可するだけでなく、拒否することも可能で、各ルールに優先順位をつけて設定します。デフォルトでは全てのインバウンドとアウトバウンドを許可するルールが設定されています。

セキュリティグループとネットワーク ACL の違いを次の表 4-3-1 でまとめます。

表 4-3-1　セキュリティグループとネットワーク ACL の違い

	セキュリティグループ	ネットワーク ACL
適用単位	EC2 や RDS、ELB など、インスタンス単位	サブネット単位
作成（追加）可能なルール	許可のみ	許可／拒否
デフォルトルール（作成時）	インバウンド：すべて拒否 アウトバウンド：すべて許可	インバウンド：すべて許可 アウトバウンド：すべて許可
特徴	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 の違いの 1 つに、ステートフル／ステートレスがあります。セキュリティグループはステートフルなファイアウォールであり、アウトバウンドで許可されて送出したトラフィックの情報を保持しているため、その戻りのトラフィックはインバウンドで許可しなくとも受け付けます。その逆も同様で、インバウンドで許可して受信したトラフィックの戻りのトラフィックはアウトバンドで許可しなくとも送出できます。これに対して、ネットワーク ACL はステートレスであるため、戻りのトラフィックを通すには、インバウンド／アウトバンドの設定で許可しておく必要があります。

試験のポイント！

セキュリティグループとネットワーク ACL の違いを押さえて、ファイアウォールによるトラブルシューティングに対応できるようにする

補足　セキュリティグループのインバウンド／アウトバウンドの送信元／送信先として、セキュリティグループの ID を指定することができます。セキュリティグループ ID を指定すると、そのセキュリティグループが適用されているインスタンスから／へのアクセスを許可できます。

4-4 VPC ピア接続

VPC ピア接続とは、2 つの VPC を接続する機能です。たとえば、本番環境と開発環境で異なる VPC にシステムを構築する場合があります。本番環境と開発環境の VPC を分けているものの、本番環境と開発環境の間で通信する必要がある場合には、VPC ピア接続を利用し、プライベート IP で通信を行います。2 つの VPC 間で VPC ピア接続を確立すると、双方の VPC に PCX というゲートウェイに相当するものが作成されます。そして、ルートテーブルの設定で送信先のターゲットとして PCX を設定することにより、各 VPC 内のサブネット間でプライベート IP での通信が可能になります（図 4-4-1）。

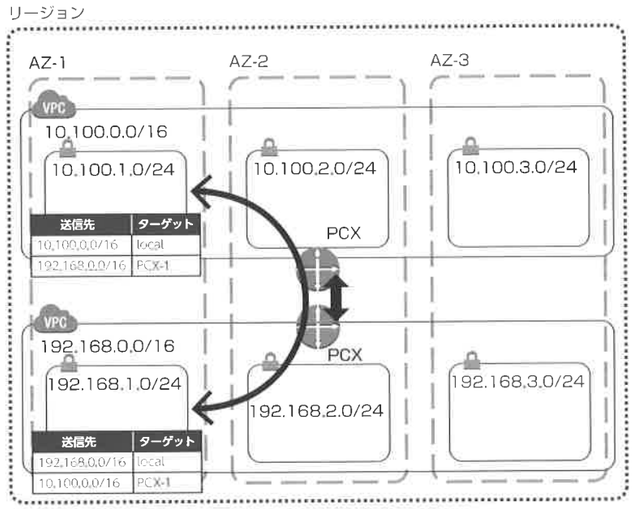

図 4-4-1　VPC ピア接続とルートテーブル

なお、VPC ピア接続には、次の制約があります。

- 接続する VPC は同じリージョンに存在する必要がある
- 接続する VPC のプライベートネットワークアドレス空間は重複していない
- 1 対 1 の接続である

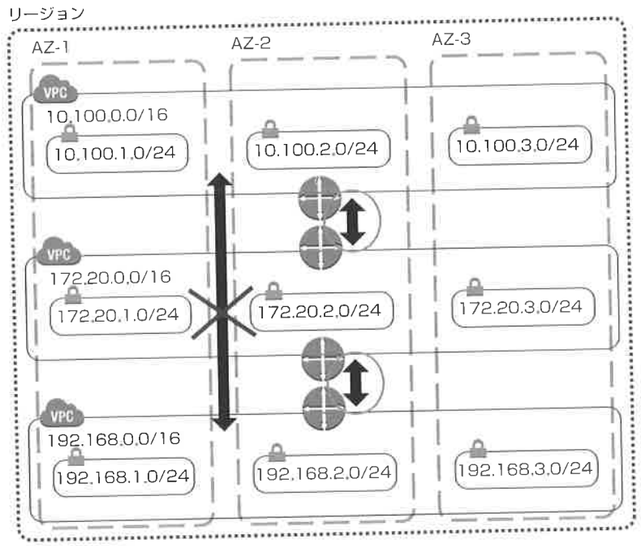

図 4-4-2　3 つの VPC ピア接続

図 4-4-2 において、VPC-A、VPC-B、VPC-C の 3 つの VPC は、すべて同じリージョンに存在しています。また、各 VPC のネットワークアドレス空間同士は重複していません。このとき、VPC-A と VPC-B、VPC-B と VPC-C の間で VPC ピア接続を確立しても、VPC-A の EC2 インスタンスから VPC-C の EC2 インスタンスにプライベート IP アドレスで直接アクセスすることはできません。VPC-A の EC2 インスタンスから VPC-C の EC2 インスタンスにプライベート IP アドレスでアクセスするには、VPC-A と VPC-C の間で直接 VPC ピア接続を確立する必要があります。

試験のポイント！

VPC ピア接続の特徴／制約を押さえる

章末問題

Q1　サブネットの特徴についての次の記述のうち、正しいものはどれか？

○ **A**　7 つのベストプラクティスの 1 つである「故障に備えた設計で障害を回避」を実践するために、サブネットは複数の AZ にまたがって作成することが推奨されている。

○ **B**　サブネットを作成する際、パブリックサブネット機能を有効化することで、インターネットと通信が可能なパブリックサブネットを作成することができる。

○ **C**　異なる AZ に作成されたサブネット間の通信も、デフォルトのルーティングルールで許可されており、ルーティングルールでは通信を制限することができない。

○ **D**　パブリックサブネット内のインスタンスとプライベートサブネット内のインスタンスの通信を許可するには、セキュリティグループ、ネットワーク ACL とルートテーブルの 3 つ全ての設定を見直す必要がある。

Q2　プライベートサブネット内の EC2 インスタンスがインターネットにアクセスするのに不要な手順はどれか？

○ **A**　インターネットゲートウェイを VPC にアタッチする。

○ **B**　NAT インスタンスを作成し、NAT インスタンスの送信先／送信元チェックを無効化する。

○ **C**　インターネットにアクセスさせるプライベートサブネット内の EC2 インスタンスに、Elastic IP をアタッチする。

○ **D**　プライベートサブネットのルートテーブルの送信先 0.0.0.0/0 のターゲットに NAT インスタンスを設定する。

Q3 パブリックサブネット内のWebサーバのEC2インスタンスにインターネットからHTTPアクセス（80番ポート）ができない。確認不要な設定はどれか？

- ○ **A** Webサーバのセキュリティグループのインバウンドで、80番ポートへのアクセスが許可されていることを確認する。
- ○ **B** Webサーバのセキュリティグループのアウトバウンドで、戻りのトラフィックの通過が許可されていることを確認する。
- ○ **C** パブリックサブネットのネットワークACLのインバウンドで、80番ポートへのアクセスが許可されていることを確認する。
- ○ **D** パブリックサブネットのネットワークACLのアウトバウンドで、戻りのトラフィックの通過が許可されていることを確認する。

Q4 VPCピア接続の正しい利用方法はどれか？

- ○ **A** 災害対策で2つのリージョンに同じシステムを構築した。リージョン間のデータの同期のために、各VPCをピア接続で接続した。
- ○ **B** 本番環境と開発環境を同一のネットワーク環境（同じプライベートCIDRブロック）としたいため、VPCを分けて作成した。本番環境で発生した障害を開発環境で検証するために、本番環境と開発環境をVPCピア接続で接続した。
- ○ **C** オンプレミスのデータセンタとVPC-AがVPN接続されている。VPC-B内のECインスタンスにオンプレミスのデータセンタからVPNを介してセキュアに接続するために、オンプレミスのデータセンタ内のルーティングルールでVPC-B宛を追加し、VPC-AとVPC-BをVPCピア接続で接続した。
- ○ **D** プライベートネットワークアドレスが10.200.0.0/16のVPC-Aと192.168.0.0/16のVPC-BをVPCピア接続で接続したところ、PCX-1が作成された。そこで、VPC-Aのルートテーブルに「送信先192.168.0.0/16のターゲットとしてPCX-1」を追加し、VPC-Bのルートテーブルに「送信先10.200.0.0/16のターゲットとしてPCX-1」を追加した。

答え

A1 C

サブネットはAZをまたがって作成することはできません。
パブリックサブネットかプライベートサブネットかを決定するのは、ルートテーブルの設定です。
VPC内のすべてのサブネット間の通信はデフォルトのルーティングルールで許可されており、変更や削除はできません。

A2 C

プライベートサブネット内のEC2インスタンスにグローバルIPアドレスをアタッチしても、ルーティングルールにより、そのグローバルIPアドレスにアクセスすることはできません。

A3 B

セキュリティグループはステートフルのため、戻りのトラフィックについてはルールを確認する必要はありません。

A4 D

VPCピア接続は接続する2つのVPCが同一リージョンに存在する必要があります。また、プライベートネットワークアドレス空間は重複できません。VPCピア接続は1対1であり、あるVPCを経由して別のVPCとピア接続することはできません。