第 5 章

AWS におけるコンピューティング (EC2／AMI／EBS／インスタンスストア)

AWS におけるコンピューティングについて

AWS における仮想サーバ（マシン）を、EC2 と言います。EC2 は AWS の主要サービスの 1 つで、AWS を使う上で最も基本的で AWS の核となるサービスといっても過言ではありません。認定試験においても、EC2 は様々な分野にまたがって関わり、出題されます。

本章では、EC2 と EC2 の仮想マシンイメージである AMI、そしてデータを格納するブロックデバイスである EBS／インスタンスストアについて解説します。

5-1 EC2 の初回起動と設定

Amazon Elastic Compute Cloud（以下 EC2）は、AWS における仮想サーバ（マシン）のことです。EC2 の特徴の 1 つとして、必要なときに必要な台数を起動し、必要がなくなれば解放することが挙げられます。これにより、負荷の増減に対応でき、コンピューティングリソースコストの削減が実現されます。このサービスにおける費用の支払い方法としては、起動していた時間に応じた金額を支払うオンデマンドの従量課金制と、長期利用契約でオンデマンドよりも安く利用できるリザーブドインスタンス、入札形式のスポットインスタンスの 3 種類がありますが、これらについては後の章で説明します。

EC2 インスタンスは、VPC のサブネット内で起動する AZ サービスです。

マネージメントコンソールから EC2 インスタンスを初回起動（作成）する手順（ステップ）は、次のとおりです。

① Amazon Machine Image（以下 AMI）の選択
② インスタンスタイプの選択
③ ネットワーク／IAM ロール／ユーザデータなどの設定（インスタンスの詳細設定）
④ ストレージの設定
⑤ タグ付け
⑥ セキュリティグループの設定
⑦ ここまでの設定の確認
⑧ キーペアの選択

① AMI の選択

AMI は、EC2 インスタンスを初回起動（作成）する際に必要となる仮想マシンイメージです。EC2 インスタンスの OS が格納されるボリュームを AWS ではルートボリュームといいますが、AMI にはルートボリュームのテンプレート（OS データ）や、ルートボリューム以外のボリュームのマッピング情報などが含まれています。AMI には、AWS から提供される Windows Server や各種 Linux ディストリビューションの他、AWS

Marketplace で購入できる各種ソフトウェアのインストール済みのイメージや、利用者が作成したカスタム AMI を利用できます。利用者は、任意のタイミングで、現在利用している EC2 インスタンスのカスタム AMI（仮想マシンのバックアップ）を取得できます。

② インスタンスタイプの選択

インスタンスタイプは、EC2 インスタンスのマシンスペックを規定する仮想 CPU コア数やメモリ容量などの組み合わせで、そ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けられます。インスタンスファミリーと各ファミリーに属する 2016 年 1 月時点での現行世代のインスタンスタイプは、表 5-1-1 のとおりです。

表 5-1-1　インスタンスファミリーと現行世代インスタンスタイプ

インスタンスファミリー	インスタンスタイプ
汎用（バランス重視）	t2.nano、t2.micro、t2.small、t2.medium、t2.large m4.large、m4.xlarge、m4.2xlarge、m4.4xlarge、m4.10xlarge m3.medium、m3.large、m3.xlarge、m3.2xlarge
コンピューティングの最適化（CPU 重視）	c4.large、c4.xlarge、c4.2xlarge、c4.4xlarge、c4.8xlarge c3.large、c3.xlarge、c3.2xlarge、c3.4xlarge、c3.8xlarge
GPU	g2.2xlarge、g2.8xlarge
メモリの最適化	r3.large、r3.xlarge、r3.2xlarge、r3.4xlarge、r3.8xlarge
ストレージの最適化	i2.xlarge、i2.2xlarge、i2.4xlarge、i2.8xlarge d2.xlarge、d2.2xlarge、d2.4xlarge、d2.8xlarge

t や m などの後の数字はインスタンスタイプの世代を示しており、その後の文字はマシンスペックの規模を示しています。例えば、m4.large は仮想 CPU コア数が 2、メモリが 8GiB のスペックになります。また、m4.xlarge は仮想 CPU コア数が 4、メモリが 8GiB のスペックです。

③ ネットワーク／IAM ロール／ユーザデータなどの設定（インスタンスの詳細設定）

このステップでは、EC2 インスタンスを起動する VPC やサブネット（4 章）を選択し、必要に応じて動的なグローバル IP アドレスである Public IP（4 章）や IAM ロール（3 章）を割り当てたり、ユーザデータを設定したりします。

ユーザデータは OS の起動スクリプトのようなもので、EC2 インスタンスの初回起動時（作成時）に実行したい処理を設定します。例えば、Web サーバとして利用する EC2 インスタンスを初回起動（作成）する際に、ユーザデータで「Apache Web サーバをインストールして、Web サービスを起動し、OS 再起動時にも Web サービスが起動する」ように設定することができます。

ユーザデータの中で、固定のグローバル IP アドレスである Elastic IP（4 章）を初回起動してくる EC2 インスタンスに関連付ける設定もできます。その場合、「aws ec2 associate-address」というコマンドを使用しますが、コマンドの引数には EC2 インスタンスのインスタンス ID またはネットワークインタフェースの ID が必要になります。インスタンス ID は、EC2 インスタンスが作成されてから割り振られる固有の ID で、ユーザデータを設定する時点では値が決まっていないため、記述することができません。この問題を解決する手段として、**メタデータ**を利用することができます。メタデータは、インスタンス ID や IP アドレス、ホスト名など EC2 インスタンス自身に関するデータで、実行中の EC2 インスタンスは、設定や管理のためにメタデータを利用することができます。主なメタデータとして、表 5-1-2 のようなものがあります。

表 5-1-2　主なメタデータ

メタデータ	説明
ami-id	インスタンスの作成に使用された AMI ID
hostname	ホスト名
iam/security-credentials/role-name	IAM ロール名
instance-id	インスタンスの ID
local-ipv4	プライベート IP アドレス
public-ipv4	Public／Elastic IP アドレス

試験のポイント！

ユーザデータおよびメタデータの用途と、メタデータで参照できる主要なデータを押さえる！

④ 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デバイス（ブロックストレージ）を設定します。デフォルトで EC2 インスタンスに接続しているストレージデバイスには、OS がインストールされます。ストレージデバイスには、EBS とインスタンスストアの 2 種類があり（5-3 参照）、これらを追加で接続することができます。EBS の追加は EC2 インスタンスの初回起動後でも可能ですが、インスタンスストアを追加できるのは EC2 インスタンスの初回起動時のみです。

EBS ボリュームは、EBS のタイプ（5-4 参照）によりますが、1GiB から 16TiB のサイズのものを作成できます。インスタンスストアについては、インスタンスタイプによって作成できるボリュームサイズと本数が決まっており、インスタンスストアが接続できないインスタンスタイプもあります。

⑤ タグ付け

EC2 インスタンスなど、AWS 上に作成したリソースには、**タグ**を付けることができます。タグは、キーと値のペアで構成され、例えば、「Name」キーに「Web Server」値というタグを EC2 インスタンスに付けると、そのタグを元に検索をかけたり、コマンドで操作する際の絞込み条件として指定することができます。

⑥ セキュリティグループの設定

EC2 インスタンスのファイアウォールである**セキュリティグループ**の設定をします。EC2 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、このステップで新しいセキュリティグループを作成することも、既存のセキュリティグループを設定することもできます。

⑦ ここまでの設定の確認

ステップ①から⑥までの手順で設定した内容を確認します。

⑧ キーペアの選択

AWS では、EC2 インスタンスにログインするためにデフォルトではキーペアを利用します。図 5-1-1 のようにあらかじめキーペアを作成しておく

ユーザデータは OS の起動スクリプトのようなもので、EC2 インスタンスの初回起動時（作成時）に実行したい処理を設定します。例えば、Web サーバとして利用する EC2 インスタンスを初回起動（作成）する際に、ユーザデータで「Apache Web サーバをインストールして、Web サービスを起動し、OS 再起動時にも Web サービスが起動する」ように設定することができます。

ユーザデータの中で、固定のグローバル IP アドレスである Elastic IP（4 章）を初回起動してくる EC2 インスタンスに関連付ける設定もできます。その場合、「aws ec2 associate-address」というコマンドを使用しますが、コマンドの引数には EC2 インスタンスのインスタンス ID またはネットワークインタフェースの ID が必要になります。インスタンス ID は、EC2 インスタンスが作成されてから割り振られる固有の ID で、ユーザデータを設定する時点では値が決まっていないため、記述することができません。この問題を解決する手段として、**メタデータ**を利用することができます。メタデータは、インスタンス ID や IP アドレス、ホスト名など EC2 インスタンス自身に関するデータで、実行中の EC2 インスタンスは、設定や管理のためにメタデータを利用することができます。主なメタデータとして、表 5-1-2 のようなものがあります。

表 5-1-2　主なメタデータ

メタデータ	説明
ami-id	インスタンスの作成に使用された AMI ID
hostname	ホスト名
iam/security-credentials/role-name	IAM ロール名
instance-id	インスタンスの ID
local-ipv4	プライベート IP アドレス
public-ipv4	Public／Elastic IP アドレス

試験のポイント！

ユーザデータおよびメタデータの用途と、メタデータで参照できる主要なデータを押さえる！

④ 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デバイス（ブロックストレージ）を設定します。デフォルトで EC2 インスタンスに接続しているストレージデバイスには、OS がインストールされます。ストレージデバイスには、EBS とインスタンスストアの 2 種類があり（5-3 参照）、これらを追加で接続することができます。EBS の追加は EC2 インスタンスの初回起動後でも可能ですが、インスタンスストアを追加できるのは EC2 インスタンスの初回起動時のみです。

EBS ボリュームは、EBS のタイプ（5-4 参照）によりますが、1GiB から 16TiB のサイズのものを作成できます。インスタンスストアについては、インスタンスタイプによって作成できるボリュームサイズと本数が決まっており、インスタンスストアが接続できないインスタンスタイプもあります。

⑤ タグ付け

EC2 インスタンスなど、AWS 上に作成したリソースには、**タグ**を付けることができます。タグは、キーと値のペアで構成され、例えば、「Name」キーに「Web Server」値というタグを EC2 インスタンスに付けると、そのタグを元に検索をかけたり、コマンドで操作する際の絞込み条件として指定することができます。

⑥ セキュリティグループの設定

EC2 インスタンスのファイアウォールである**セキュリティグループ**の設定をします。EC2 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、このステップで新しいセキュリティグループを作成することも、既存のセキュリティグループを設定することもできます。

⑦ ここまでの設定の確認

ステップ①から⑥までの手順で設定した内容を確認します。

⑧ キーペアの選択

AWS では、EC2 インスタンスにログインするためにデフォルトではキーペアを利用します。図 5-1-1 のようにあらかじめキーペアを作成しておく

と、公開鍵と秘密鍵のペアのうち、公開鍵は AWS 上に保管され、秘密鍵はローカルにダウンロードされます。

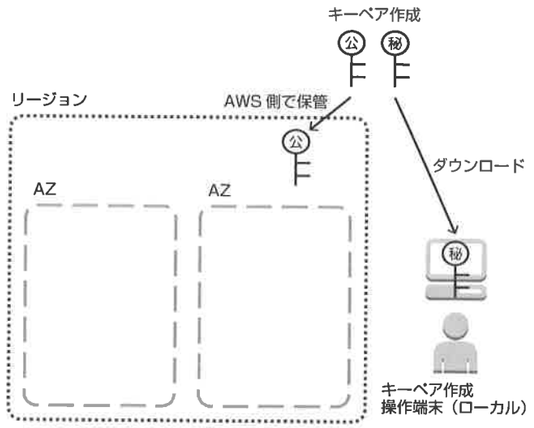

図 5-1-1　キーペアの作成

EC2 インスタンスを初回起動する際には、キーペアを選択でき、AWS 上に保管されている公開鍵がその EC2 インスタンスに埋め込まれます。キーペアは、図 5-1-2 に示すように、EC2 インスタンスの OS が Linux の場合 SSH の認証に利用し、Windows の場合は暗号化されている Administrator ユーザのパスワードの復号化に利用します。

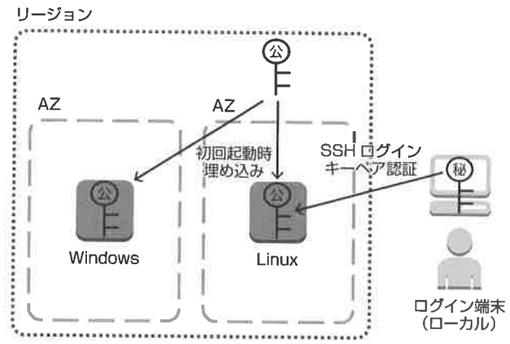

図 5-1-2　キーペア認証とパスワード復号化

5-2 EC2 インスタンスのライフサイクル

EC2 インスタンスは、図 5-2-1 のように 7 つの状態を遷移します。

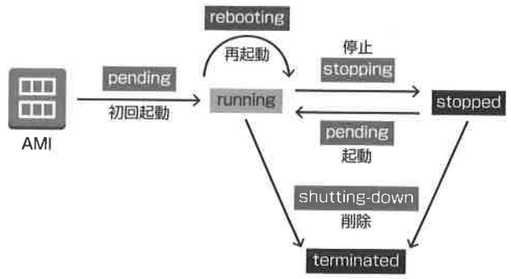

図 5-2-1　EC2 インスタンスの状態遷移

EC2 インスタンスは、初回起動をかけると pending 状態になり、起動処理が終了すると running 状態になります。ただ、running 状態になっても、次の 2 種類のステータスチェックがかかってその間は EC2 インスタンスにアク

セスできない場合があります。

- System Status Checks：インフラストラクチャ（HW、ハイパーバイザ）のチェック
- Instance Status Checks：OS のチェック

この 2 つのステータスチェックに通ると、「2/2 checks passed」となり、EC2 インスタンスが正常起動していることがわかります。

オンデマンドの EC2 インスタンスの利用料金は、running になった時点から発生し、stopped あるいは terminated になるまで発生します。

5-3 EBS とインスタンスストア

EC2 インスタンスに接続できるブロックストレージには、次の 2 種類があります。

- Amazon EBS（Elastic Block Store）
- インスタンスストア（インスタンスストレージ）

EBS は、AZ 内に作成されるネットワーク接続型のブロックストレージで、**不揮発性**（永続的なデータボリューム）という特徴があります。一方、**インスタンスストア**は、EC2 インスタンスの物理ホストの内蔵ストレージで、**揮発性**（一時的なデータボリューム）という特徴があり、EC2 インスタンスを停止すると保存されていたデータは削除されます。

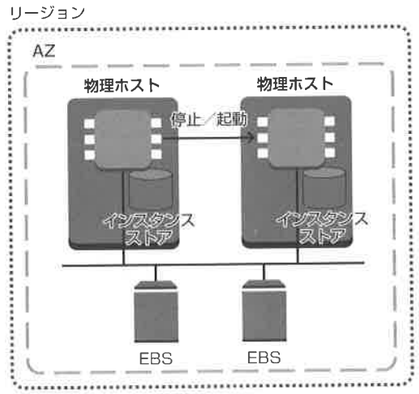

図 5-3-1　EBS とインスタンスストア

デフォルトで EC2 インスタンスを一度停止し、再び起動すると、図 5-3-1 のように物理ホストが変わるため、インスタンスストアは揮発性という特徴があります。EBS とインスタンスストアには他にも表 5-3-1 のような違いがあります。性能を表す IOPS は Input Output Per Second の略で、1 秒あたりの処理できる読み書きの回数です。

表 5-3-1　EBS とインスタンスストアの違い

	EBS	インスタンスストア
データ特性	不揮発性	揮発性
性能	数百～20,000 IOPS	最大 300,000 IOPS
価格	$/GB	無料（EC2 の料金に含まれる）

重要！

ブロックストレージには EBS とインスタンスストアの 2 種類があり、不揮発性と揮発性という違いがある！

EBS とインスタンスストアの違いに起因して、EC2 インスタンスには「**EBS-backed インスタンス**」と「**instance store-backed インスタンス**」という 2 つのタイプがあります。両者の違いは、OS がインストールされる

ルートボリュームがEBSか、あるいはインスタンスストアかです（図5-3-2）。

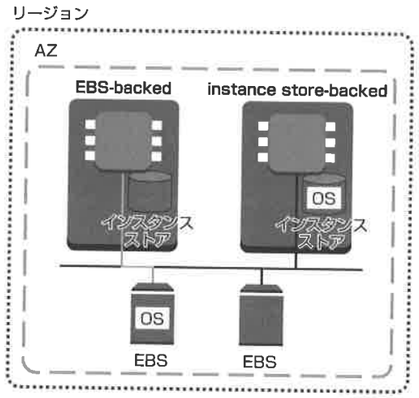

図5-3-2　EBS-backedインスタンスとinstance store-backedインスタンス

EBSの特徴から、EBS-backedインスタンスは停止と起動、再起動、削除ができます。これに対して、インスタンスストアの特徴から、instance store-backedインスタンスは再起動と削除しかできません。EBS-backedインスタンスとinstance store-backedインスタンスはAMIが異なっており、AMIごとにEBS-backedインスタンス用のものか、instance store-backedインスタンス用のものかが決まっています。instance store-backedインスタンスは、「s3-backedインスタンス」という別名で呼ばれることもあります。

試験のポイント！

EBS-backedインスタンスとinstance store-backedインスタンスの特徴を押さえる！

5-4 EBSのタイプ

EBSには、General Purpose SSD、Provisioned IOPS SSD、Magnetic（磁気ディスク）という3つのタイプがあります。これらは表5-4-1に示すように、性能面や費用面、それに伴う用途などに違いがあります。

表5-4-1　EBSの3タイプ

	General Purpose SSD	Provisioned IOPS SSD	Magnetic
ボリュームサイズ	1GiB～16TiB	4GiB～16TiB	1GiB～1TiB
IOPS	3 IOPS/GBのベースパフォーマンス 最大10,000 IOPS ベースパフォーマンスが3,000 IOPS未満の場合、3,000 IOPSまでのバースト機能	容量（GB）の30倍までのIOPSを指定 最大20,000 IOPS	平均100 IOPS
価格	・容量のみ	・容量 ・指定した性能（IOPS）	・容量 ・発生したIO数
ユースケース	一般（汎用）	10,000 IOPSを超える性能が求められるアプリ・DBなど	IOがあまり発生せず、コストが重視されるマシン

Provisioned IOPS SSDはEBSディスク性能を高めることができますが、EBSはネットワーク接続型のストレージのため、ネットワークがボトルネックになります。通常のEC2インスタンスでは、業務ネットワークの帯域とEBSのディスクI/Oの帯域が競合した状態になっています。この対策として、EC2インスタンスを「**EBS最適化インスタンス**」というタイプで起動すれば、EBSのディスクI/O専用の帯域が確保され、EBSのディスクI/Oが安定化します（図5-4-1）。Provisioned IOPS SSDでは、EBSの高いディスクI/O性能を活かすためにも、接続するEC2インスタンスをEBS最適化インスタンスとすることが推奨されています。

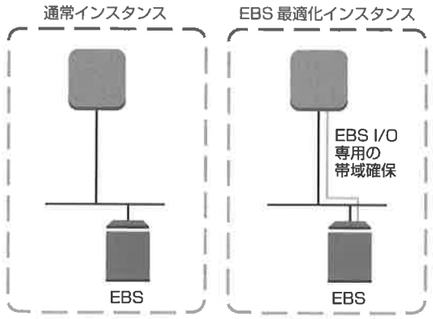

図 5-4-1　通常の EC2 インスタンスと EBS 最適化インスタンス

試験のポイント！

EBS ボリュームタイプの性能の違いと EBS 最適化インスタンスの使いどころを押さえる！

補足　2016 年 4 月にスループット最適化 HDD と Cold HDD というボリュームタイプが利用できるようになりましたが、本書では割愛します。

5-5 EBS スナップショット

EBS ボリュームは、任意のタイミングで**スナップショット**を作成することができます。スナップショットは S3 に保存される EBS 内のデータのバックアップで、耐久性の高い S3 にバックアップをとることで、EBS 内のデータ喪失を防ぐことができます。スナップショットで S3 に保存されるデータは圧縮されており、また差分のデータのみが保存されていくため、毎日スナップショットを取得したとしても、バックアップストレージに要する費用を低く抑えることができます。

スナップショットを取得する際、データの整合性を保つためにディスク I/O を停止する必要があり、Linux であれば対象の EBS ボリュームをアンマウントしてからスナップショットを取得することが推奨されています。ただし、スナップショットは取得開始時点の EBS ボリューム内のデータをすべてキャ

プチャして、それ以降の書き込みはキャプチャ対象外となるため、スナップショット取得開始後は、取得完了を待たずに再びマウントして使用することができます。

スナップショットから EBS ボリュームを復元（作成）する際は、元となった EBS ボリュームとは異なる AZ や EBS のタイプ、また元となった EBS ボリュームのサイズよりも大きいディスクサイズを指定して復元（作成）することができます。

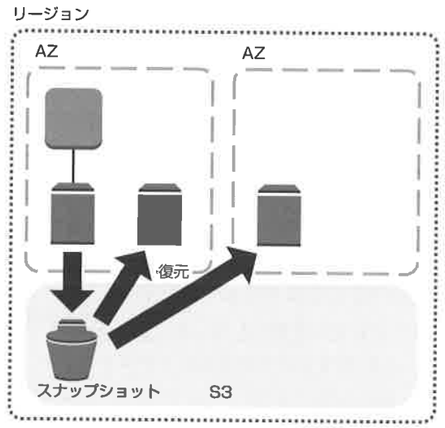

図 5-5-1　スナップショットの作成と復元

試験のポイント！

EBS スナップショットの特徴を押さえる！

EBS ボリュームは AZ サービスであり、ある AZ 内に作成された EBS ボリュームは図 5-5-2 のように、同じ AZ 内の EC2 にのみアタッチすることができます。

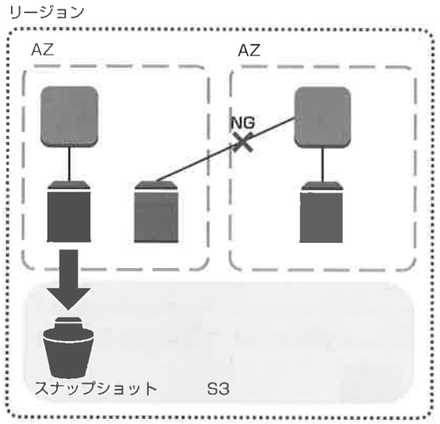

図 5-5-2　AZ をまたいだ EBS ボリュームの接続不可

別の AZ やリージョンで同じデータが格納されている EBS ボリュームを使用したい場合は、スナップショットを利用します。図 5-5-3 のように、スナップショットから別の AZ に EBS ボリュームを復元したり、あるいはスナップショットを別のリージョンにコピーして、そこから EBS ボリュームを復元し、その EBS ボリュームを EC2 インスタンスにアタッチします。

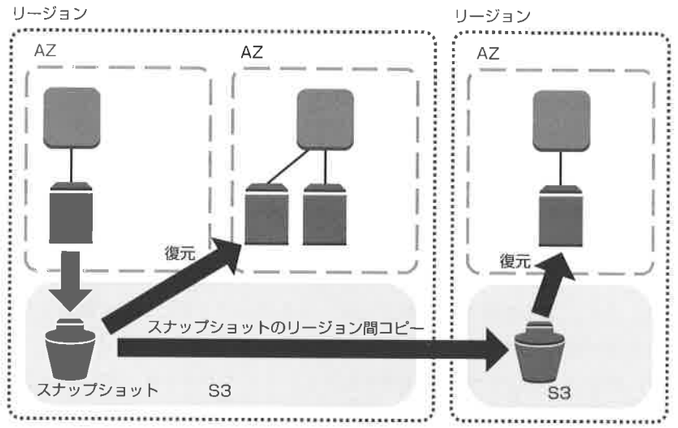

図 5-5-3　AZ／リージョンをまたいだ EBS ボリュームの復元

試験のポイント！

EBS スナップショットを介した AZ／リージョン間の EBS ボリュームの複製の流れを押さえる。

5-6 プレイスメントグループ

AWS のリージョンにある各 AZ は、自然災害などに対しても冗長性を担保するために、互いに地理的に離れた場所に存在しています。そのため、リージョン内の AZ 間は専用線で接続されているものの、異なる AZ 内の EC2 インスタンスへのアクセスには多少の遅延が発生します。そこで、ある単一 AZ に**プレイスメントグループ**というものを作成し、その中に EC2 インスタンスを作成すると、プレイスメントグループ内の EC2 インスタンス間のネットワーク接続を高速化することができます。ただし、プレイスメントグループではネットワークの高速化のために冗長性を犠牲にしているので、使用用途について注意が必要です。

試験のポイント！
プレイスメントグループ内にEC2インスタンスを起動することで、EC2インスタンス間のネットワーク接続を高速化できる。

5-7 Dedicatedインスタンス

EC2インスタンスは、デフォルトではAZの中の任意の物理ホストの上で起動します。ある物理ホストの上には、ハイパーバイザによって厳格に分離された形で複数の異なるアカウントのEC2インスタンスが同時に起動しています。EC2インスタンス上で利用するソフトウェアのライセンスによっては、ソフトウェアをインストールするサーバのハードウェアを利用者が専有していることを求めている場合があります。また、コンプライアンス上、利用しているEC2インスタンスと他者のEC2インスタンスが同じ物理ホストで共存することを許していない場合もあります。このような要件には、Dedicatedインスタンス（ハードウェア専有インスタンス）を利用することで対応が可能になります。Dedicatedインスタンスは、EC2インスタンスを起動する物理ホストに、別のアカウントのEC2インスタンスが起動しないことを保証します。これにより、上記の要件を満たすことができますが、EC2インスタンスの時間あたりの利用料金に加えて、リージョンごとの専有料金が発生します。

補足　2015年11月にDedicatedホストという、物理ホストをアカウントに割り当てておき、その中にEC2インスタンスを起動していく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。

章末問題

Q1 EC2インスタンスの初回起動時にソフトウェアをインストールしたり、サービスを起動したりすることを指定するEC2のデータはどれか？

- ○ **A** メタデータ
- ○ **B** 起動スクリプト
- ○ **C** rc.ec2
- ○ **D** ユーザデータ

Q2 インスタンスIDやIPアドレスなど、EC2インスタンス自身に関する情報が格納されており、EC2インスタンスの初回起動時の設定などに用いられるEC2のデータはどれか？

- ○ **A** メタデータ
- ○ **B** 起動スクリプト
- ○ **C** rc.ec2
- ○ **D** ユーザデータ

Q3 EBSボリュームに格納しているデータが削除されるタイミングとして正しい選択肢はどれか？

- ○ **A** EC2インスタンス再起動時
- ○ **B** EC2インスタンス停止時
- ○ **C** EC2インスタンス削除時（EBSボリュームのDelete on Terminationはオフ）
- ○ **D** OSからデータの削除処理を行った時

Q4 インスタンスストアに格納しているデータが削除されるタイミングとして正しい選択肢はどれか？正しい選択肢を**全て**選べ。

- □ **A** EC2インスタンス再起動時
- □ **B** EC2インスタンス停止時
- □ **C** EC2インスタンス削除時（EBSボリュームのDelete on Terminationはオフ）

□ **D** OSからデータの削除処理を行った時

Q5 EBS-backedインスタンスにはできるが、instance store-backedインスタンスにはできない操作はどれか？

○ **A** EC2インスタンス再起動
○ **B** EC2インスタンス停止
○ **C** EC2インスタンス削除
○ **D** OSからのシャットダウン（停止）

Q6 Provisioned IOPSタイプのEBSボリュームを利用して、安定したIO性能をアプリケーションに提供したい。推奨される構成として正しい選択肢はどれか？

○ **A** Provisioned IOPSのEBSボリュームを4本以上用意し、ソフトウェアRAIDによってRAID5を組む。
○ **B** Provisioned IOPS最適化EC2インスタンスを起動して、Provision IOPSのEBSボリュームをそのEC2インスタンスにアタッチする。
○ **C** EBS最適化EC2インスタンスを起動して、Provisioned IOPSのEBSボリュームをそのEC2インスタンスにアタッチする。
○ **D** Provisioned IOPSのEBSボリュームを2本用意し、ソフトウェアRAIDによってRAID1を組む。

Q7 EBSデータボリューム（非ルートボリューム）のスナップショット取得方法として、適切な選択肢はどれか？

○ **A** 対象となるEBSボリュームへのディスクI/Oは気にせずに、スナップショットの取得を開始する。
○ **B** 対象となるEBSボリュームへのディスクI/Oが少なくなる時間帯に、スナップショットの取得を開始する。
○ **C** 対象となるEBSボリュームをアンマウントし、スナップショットの取得を開始する。開始後はスナップショットの取得完了を待たずにEBSボリュームを再びマウントして、ディスクI/Oを再開する。
○ **D** 対象となるEBSボリュームをアンマウントし、スナップショットの取得を開始する。スナップショットの取得完了を待ってEBSボリュームを再びマウントして、ディスクI/Oを再開する。

Q8 EBSボリュームの特徴（取り扱い）として、正しい選択肢はどれか？

○ **A** EBSボリュームは同じVPC内のEC2インスタンスであれば、どのEC2インスタンスにでもアタッチ（接続）することができる。
○ **B** あるEBSボリュームを他のリージョンのEC2インスタンスにアタッチ（接続）したい場合、EBSボリュームのリージョン間コピーを利用して、該当リージョンにコピーする。
○ **C** あるEBSボリュームを同じリージョン内の別のAZのEC2インスタンスにアタッチしたい場合は、EBSボリュームのAZ間コピーを利用して、該当AZにコピーする。
○ **D** あるEBSボリュームを同じリージョン内の別のAZのEC2インスタンスにアタッチしたい場合は、一度スナップショットを作成し、そのスナップショットから新たなEBSボリュームを作成する。

Q9 プレイスメントグループの特徴を示した選択肢はどれか？

○ **A** 単一のAZ内に作られたグループで、そのグループ内に起動したEC2インスタンス間のネットワークアクセスを高速化する。
○ **B** 単一のリージョン内に作られたグループで、そのグループ内に起動したEC2インスタンス間のネットワークアクセスを高速化する。
○ **C** 単一のAZ内に作られたグループで、そのグループ内に起動したEC2インスタンス間の通信が自動的に暗号化される。
○ **D** 単一のリージョン内に作られたグループで、そのグループ内に起動したEC2インスタンス間の通信が自動的に暗号化される。

Q10 Dedicatedインスタンスの特徴を示した選択肢はどれか？

○ **A** ネットワーク接続型のストレージであるEBSボリュームとの間に、ディスクI/O専用の帯域を確保し、ディスクI/Oを安定化させるEC2インスタンス。
○ **B** 特定のAZ内で、特定のインスタンスタイプを1年あるいは3年の契約で低価格で利用できるEC2インスタンス。
○ **C** 特定のAZにおける市場価格に対して、その市場価格を上回る価格で入札し、低価格で利用できるEC2インスタンス。
○ **D** EC2インスタンスを起動する物理ホストにおいて、自アカウント以外のEC2インスタンスが起動しないことを保証されたEC2インスタンス。

答え

A1 D

A　メタデータは、EC2インスタンス自身のデータです。
B　起動スクリプトでもソフトウェアのインストールやサービスの起動はできますが、これはEC2ではなく、OS内の機能（スクリプト）です。
C　rc.ec2という機能はありません。

A2 A

Q1.と同じです。
ユーザデータには、EC2インスタンスの初回起動時に実行させたい処理を記述することができます。

A3 D

EBSボリュームは、永続的なデータ格納ボリュームであり、OSから明示的な削除処理を行わない限り削除されません。Delete on Terminationオプションがオンであれば、EC2インスタンス削除時にEBSボリュームも同時に削除しますが、そうでなければEC2の存続に関係なく、EBSボリュームはデータを保持したまま残ります。

A4 B、C、D

インスタンスストアは、揮発性のボリュームのため、EC2インスタンスを停止あるいは削除するとデータは失われます。インスタンスストアの存続にEBSボリュームのDelete on Termination設定は関係ありません。

A5 B

Instance store-backedインスタンスは、ルートボリュームがインスタンスストア（揮発性のボリューム）のため、EC2インスタンスを停止することができません。OSからシャットダウン（停止）操作が行えますが、シャットダウン（停止）した場合は、停止と同時にEC2インスタンスが削除されます。

A6 C

A　RAID5（分散パリティ）は、性能に影響を与えるため、非推奨となっています。
B　Provisioned IOPS最適化EC2インスタンスというものはありません。
D　EBSは内部的に冗長化されているので、RAID1（ミラーリング）は不要です。

A7 C

A、B　スナップショットを取得する際は、データの静止点を設けて取得します。
C、D　スナップショット取得開始後は、ディスクアクセスしても問題ありません。

A8 D

A　EBSボリュームは、同じAZ内のEC2インスタンスにのみアタッチすることができます。VPCは複数のAZにまたがって作成されるため、誤りです。
B　EBSボリュームには、リージョン間コピー機能はありません。別リージョンでも同じデータが格納されたEBSボリュームを利用したければ、スナップショットを取得して、スナップショットをリージョン間コピーし、そのコピーからEBSボリュームを復元します。
C　EBSボリュームには、AZ間コピー機能もありません。別AZで同じデータが格納されたEBSボリュームを利用したければ、スナップショットを取得して、スナップショットからEBSボリュームを復元します。

A9 A

プレイスメントグループは、単一のAZ内に作られるグループで、その中に起動したEC2インスタンス間の通信を高速化します。

A10 D

A　EBS最適化インスタンスのことです。
B　リザーブドインスタンスのことです。
C　スポットインスタンスのことです。
D　Dedicatedインスタンスは、自アカウントで物理ホストを専有し、他アカウントのEC2インスタンスが同じ物理ホストで起動しないことを保証することで、ソフトウェアライセンスやコンプライアンスに対応できます。

第 6 章

オブジェクトストレージ (S3 ／ Glacier)

オブジェクトストレージについて

AWS において、ストレージの中心的な役割を果たしているサービスは S3 というオブジェクトストレージです。S3 にはシステムの様々なデータを保存することができ、また S3 を介して別のシステムやデータ分析処理に受け渡すこともできます。認定試験においては、S3 の用途や特徴について出題されます。

また、AWS のオブジェクトストレージには、この他に Glacier があり、S3 との使い分けについても押さえておく必要があります。

本章では、オブジェクトストレージである S3 と Glacier について解説します。

6-1 S3 バケット／オブジェクトとストレージクラス

Amazon Simple Storage Service（以下 **S3**）は、安価で非常に耐久性があるオブジェクトストレージ[注1]で、AWS 上のストレージの中で中心的な役割を果たしています。S3 にデータを保存するには、特定のリージョンにバケットと呼ばれる格納先を作成し、その中に Key-Value Store 形式でファイルをアップロードします。S3 バケットにファイルをアップロードすると、キーを付与したオブジェクトとして保存されます。各オブジェクトには URL が付与され、適切なアクセス権限を設定することによって HTTPS によるアクセスが可能になります。1 オブジェクトあたり最大 5TB まで[注2]のサイズ制限がありますが、バケットに格納できるオブジェクトの数およびデータ総量は無制限です。なお、バケット名は AWS の全アカウント（全世界）で一意とする必要があり、既に存在しているバケットと同じバケット名を用いて新たに作成することはできません。

S3 には、格納したデータ（オブジェクト）の利用用途ごとに**ストレージクラス**があり、それぞれ冗長性や料金が異なります。**スタンダード（標準）クラス**のオブジェクトの耐久性は 99.999999999% と極めて高く、スタンダードクラスのオブジェクトとして格納すれば、そのオブジェクトが失われることはほぼないと考えることができます。その理由は、スタンダードクラスのオブジェクトは、バケットにオブジェクトがアップロードされると自動的にそのリージョン内の 3 か所のデータセンタに複製され、同時に 2 か所でデータロストが発生しても復元できる仕組みになっているからです。失うことが許されないオリジナルのデータなどは、このスタンダードクラスとしてデータを格納します。

注1　オブジェクトストレージは、従来のファイルシステムにおける階層構造によるデータ（ファイル）の管理とは異なり、データ（オブジェクト）にキー（ユニークな ID）を付与してフラットに格納します。データを Key-Value 形式でフラットに格納することで、大量の、かつ大容量なデータの保存に適しています。

注2　マルチパートアップロード機能という、大容量のデータを分割して並列アップロードを実行した際の最大サイズです。マルチパートアップロード機能を使用しない場合の 1 オブジェクトあたりの最大サイズは 5GB までです。

一方、オリジナルデータから加工されたデータなど、再作成可能なデータを格納する場合には、そこまでの耐久性は必要ない場合もあります。そのような利用用途に対して、低冗長化（Reduced Redundancy Storage；RRS）ストレージクラスが用意されています。低冗長化ストレージクラスの耐久性は 99.99% とスタンダードクラスの S3 に比べて低いですが、利用料金を低く抑えることができます。

試験のポイント！
S3 のストレージクラスには失われることが許されないデータを格納する用途に適したスタンダードクラスと、失われても再作成可能なデータを格納する用途に適した低冗長化クラスがある！

補足　スタンダードクラスの S3 と同等の耐久性は必要だがアクセスされる頻度の低いオブジェクト向けに、低頻度アクセス S3 というストレージクラスが 2015 年 9 月から利用できるようになりましたが、本書では割愛します。

6-2 S3 の整合性

S3 は、格納したデータを複数のデータセンタに複製することで非常に高いデータ耐久性を実現しています。しかし、そのためにデータの整合性については注意が必要になります。S3 のデータの整合性は、データに対しどの操作を行うかによって異なります。

① **新しいオブジェクトの書き込み（PUT）**
【書き込み後の読み取り整合性】
新しいオブジェクトを S3 バケットに書き込み（新規アップロード）、すぐにバケット内のオブジェクトの一覧表示操作を行うと、そのオブジェクトが表示されないことがあります。S3 から「完了」が返されると、新しく書き込んだオブジェクトもバケット内の一覧に正しく表示されるようになり、そのデータにアクセスすれば正しいデータが返されます。

② **既存オブジェクトの上書き（PUT）**
【結果整合性】

既存のオブジェクトを上書きし、「完了」が返された後でも、そのデータにアクセスした際に古いデータ（上書き前のデータ）が返されることがあります。時間が経てば結果的に正しいデータ（上書き後のデータ）が返されます。

③ **オブジェクトの削除（DELETE）**

【結果整合性】

オブジェクトを削除し、「完了」が返された後でも、削除したはずのデータがバケットの一覧に表示されたり、データにアクセスできたりすることがあります。時間が経てば結果的にバケットの一覧から削除され、アクセスできなくなります。

以上のことから、S3 はオンラインで頻繁に更新されるデータの格納先には向いておらず、格納されている静的なデータを何度も読み取るような用途に向いているといえます。

試験のポイント！

S3 の各操作とデータの整合性について押さえ、整合性を考慮した S3 の利用用途を押さえる！

6-3 S3 のアクセス制限とセキュリティ

S3 バケットやオブジェクトは、デフォルトではそのリソースを作成したアカウントだけにアクセス権限が与えられています。S3 バケットとオブジェクトには適切なアクセス制限をかける必要があり、アクセス管理の方法には、次の 3 つがあります。

- **アクセスコントロールリスト（ACL）**
- **バケットポリシー**
- **IAM（ユーザ）ポリシー**

① **アクセスコントロールリスト（ACL）**

バケットとオブジェクトそれぞれについて、読み取り／書き込みの許可を、他の AWS アカウントに与えることができます。また、オブジェクトに付与されている URL についても、HTTPS アクセスの許可を与えることができます。条件付きアクセス許可を与えることや、アクセス拒否を設定することはできず、自アカウント内の IAM ユーザやグループのアクセス権を制限することもできません。

② **バケットポリシー**

バケットごとに、自アカウント内の IAM ユーザやグループ、他アカウントのユーザに対してアクセス（様々な操作）許可を与えることができます。また、条件付きのアクセス許可を与えることや、アクセス拒否を設定することもできます。

③ **IAM（ユーザ）ポリシー**

IAM ポリシー（S3 のアクセス管理ではユーザポリシーと呼ばれることが多い）は、3 章で説明した AWS リソースに対するアクセス可否です。ここでは、その対象となる AWS リソースが S3 になります。S3 へのアクセス（様々な操作）許可を設定した IAM ポリシーを自アカウント内の IAM ユーザやグループ、ロールに割り当てます。バケットポリシーと同様に条件付きのアクセス許可を与えることや、アクセス拒否の設定をすることもできますが、他アカウントを指定したアクセス権の設定はできません。

アクセス制限には、これらの 3 種類のアクセス管理のほかに、アクセス許可設定をしていない特定のオブジェクトを指定した期間に限定して HTTPS アクセスで公開する「**署名（期限）付き URL**」という方法があります。

ここで、S3 バケットに格納された特定のオブジェクトを、限られたエンドユーザにのみアクセスさせたい要望があったとします。例えば、ある商品を購入したエンドユーザに限り、S3 バケットに格納されている特典映像の動画オブジェクトにアクセスできるようにするという場合を想定します。商品を購入するのは当然ながら IAM ユーザではないので、IAM ポリシーで動画オブジェクトへのアクセス制限をすることができません。また、商品購入者のアクセス元の IP アドレスなどを事前に特定することもできないため、バケットポリシーでもアクセス制限はできません。アクセスコントロールリスト（ACL）

で動画オブジェクト URL への HTTPS アクセスを全てのエンドユーザに公開することはできますが、動画オブジェクト URL がわかれば商品を購入していないエンドユーザも動画オブジェクトにアクセスできることになり、購入特典ではなくなってしまいます。そこで、エンドユーザが商品を購入したときに、10 分だけ有効な URL が生成され、その URL を使って動画オブジェクトにアクセスできるように設定します。そうすれば、たとえその URL が商品購入者以外に漏えいしたとしても、生成から 10 分経過すると URL の有効期限が切れ、動画オブジェクトにアクセスすることはできません。この有効期限がついた URL は、SDK によって生成され、URL の中に署名が入るため、「**署名付き URL**」といいます（図 6-3-1）。

オリジナルのオブジェクト URL の例：
https://s3-ap-northeast-1.amazonaws.com/my-bucket/sp-movie.mp4
署名付き URL の例：
https://s3-ap-northeast-1/amazonaws.com/my-bucket/sp-movie.mp4?Expires=1234567890&AWSAccessKeyId= AKIABCDEFGHIJKLMNOPQ&Signature=0123456789ABCDEFGHIJKLMNOPQRST

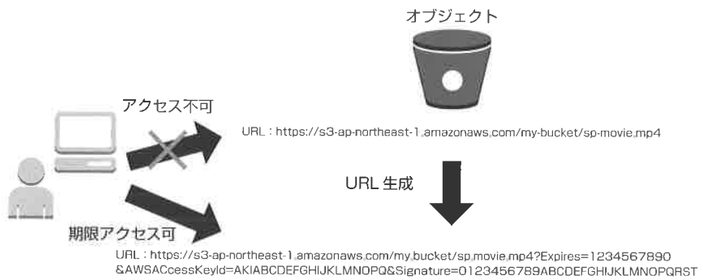

図 6-3-1　署名付き URL へのアクセス

試験のポイント！

S3 のバケットとオブジェクトのアクセス制限や、署名付き URL の利用用途を押さえる！

6-4 オブジェクトの暗号化とアクセスログ

S3 バケットに格納されているオブジェクトを任意で暗号化して、データを保護することができます。オブジェクトの暗号化は、AWS が管理する鍵やユーザが管理する鍵を使って S3 上で暗号化するサーバサイド暗号化と、クライアント側で事前に暗号化したデータを S3 バケットにアップロードするクライアントサイド暗号化の両方が可能です。

S3 バケットへのアクセスログを任意で同じ／異なる S3 バケットに取得することができます。アクセスログの取得はベストエフォートで記録されるため、完全性は保証されず、また、アクセスログ格納バケットへのログの格納は、実際のアクセスから時間を置いて行われます。

試験のポイント！

S3 の暗号化やアクセスログの取得はデフォルトではなく、ユーザの責任の元に実施する！

6-5 S3 の静的 Web サイトホスティング機能

S3 で Web サイトをホスティングすることができます。ただし、ホスティングできるのは静的な Web サイトに限られ、PHP や JSP、ASP.NET など Web サーバ側のプログラム実行により動的なページをユーザに提供する動的 Web サイトは S3 でホスティングすることができません。静的な Web サイトには、JavaScript などクライアント側で実行されるプログラムを含んだページも含まれます。S3 の **Web サイトホスティング機能**を利用することで、EC2 を利用するよりも運用の負荷やコストを抑えることができます。

なお、S3 の Web サイトホスティング機能を利用した際のアクセス先のエン

ドポイントは

バケット名 .s3-website- リージョン名 .amazonaws.com

になります。

エンドポイント例

mybucket.s3-website-ap-northeast-1.amazonaws.com

このエンドポイントを所有しているドメインでアクセスさせるためには、後述する Route 53 などの DNS サービスにより、名前解決する必要があります。

6-6 S3 のバージョニング機能

S3 には**バージョニング機能**（図 6-6-1）があり、S3 バケット単位で有効にすることができます。バージョニング機能を有効にしたバケットでは、オブジェクトを誤って上書きしたり、削除した後でも、操作前のオブジェクトを復元することができます。

バージョニング機能を有効にしたバケットに格納されるオブジェクトには、キーの他にバージョン ID が付与されます。オブジェクトを上書きアップロードする際に、図 6-3-1 のように上書き前のオブジェクトと異なるバージョン ID の付与されたオブジェクトが別に格納されます。オブジェクトを削除する場合も、異なるバージョン ID と削除マーカーが付与されたオブジェクトが生成され、削除前のオブジェクトが保持されます。

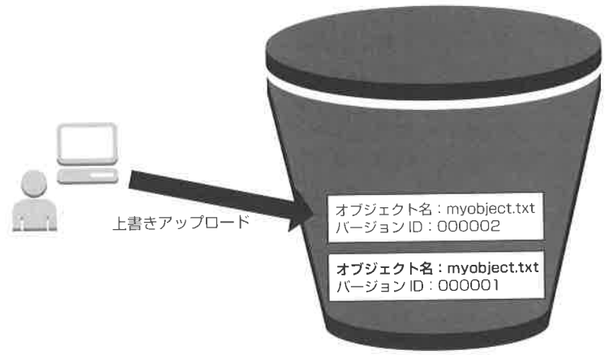

図 6-6-1　S3 のバージョニング

試験のポイント！

S3 のバージョニング機能を利用すれば、誤操作などにより上書きや削除をしてしまっても、元のデータを復元できる！

6-7 S3 のライフサイクル機能と Glacier へのアーカイブ

S3 は容量無制限で、かつ非常に耐久性の高いストレージのため、AWS におけるストレージの中心となり、様々なデータが格納されます。格納されるデータの中には、ユーザの課金ログや、ある生物のゲノムのシーケンスデータなど、削除はしたくない／できないが、普段アクセスすることはほとんどないデータも存在します。そのようなデータは、S3 と同じ耐久性を持ちながら、より低価格で利用できる Amazon Glacier（以下 Glacier）ストレージに格納することで、コストを低く抑えることができます。データのアーカイブや長期バックアップ先などの用途には Glacier が適しています。

Glacier にデータを格納する方法には、S3 の**ライフサイクル機能**を利用するものと、SDK を利用して直接格納するものがあります。S3 のライフサイクル

機能は、S3 バケットに格納したオブジェクトを、指定した日数が経過した後に Glacier に移行したり、削除したりすることができる機能です。この機能によって、「ユーザの課金ログを格納から 1 年後に Glacier に移行し、5 年後に削除する」といったジョブを自動化できます。

Glacier に格納したままのデータを参照することはできないため、監査などで格納したデータを参照したい場合には、Glacier からそのデータを取り出す必要があります。Glacier からデータを取得するのに要する時間は、データの大小にかかわらず、3〜5 時間にもなります。また、データの取り出しは Glacier に保管しているデータ量（月平均）の 5% までは無料ですが、それを超える場合は取出し料金が発生します。このような理由から、Glacier は、参照（取出し）がほとんど行われないデータを長期間保管しなければならない場合のデータの格納先として適しています。

試験のポイント！

Glacier は参照する頻度の少ないデータを長期間保管するのに適している！

章末問題

Q1 RAW 画像データを様々な形式に変換する処理を行っている。データの耐久性やコストの最適化を考慮した場合、それぞれのデータに適した格納先はどれか？

- ○ **A** RAW 画像：EBS　変換後画像：スタンダード S3
- ○ **B** RAW 画像：EBS　変換後画像：RRS S3
- ○ **C** RAW 画像：スタンダード S3　変換後画像：RRS S3
- ○ **D** RAW 画像：スタンダード S3　変換後画像：Glacier

Q2 S3 に保存すべきデータとして適していないものはどれか？

- ○ **A** 社内の従業員に公開する動画
- ○ **B** 世界中のエンドユーザに公開する動画
- ○ **C** ゲノムのシーケンスデータ
- ○ **D** DB のオンライントランザクションログ

Q3 S3 に対してオブジェクトの格納／上書き／削除を行った際、発生する可能性がある選択肢はどれか？正しい選択肢を**全て**選べ。

- □ **A** S3 バケットに新規にオブジェクトを格納する操作をし、「完了」と表示された。その後すぐにバケット内のオブジェクトの一覧表示をしたが、格納したはずのオブジェクトが表示されなかった。
- □ **B** S3 バケットに格納済みのオブジェクトを上書きする操作をし、「完了」と表示された。その後すぐに上書きしたオブジェクトを開くと以前のデータが参照された。
- □ **C** S3 バケットに格納済みのオブジェクトを削除する操作をし、「完了」と表示された。その後すぐにバケット内のオブジェクトの一覧表示をしたが、削除したはずのオブジェクトが表示された。
- □ **D** 上記はすべて発生する可能性がない。

Q4 S3 バケットに格納しているオブジェクトを特定の AWS アカウントの IAM ユーザにのみ参照させたい。利用する S3 のアクセス制限はどれか？

- ○ **A** アクセスコントロールリスト（ACL）
- ○ **B** バケットポリシー
- ○ **C** IAM ポリシー（ユーザポリシー）
- ○ **D** 署名付き URL

Q5 S3 のデフォルトで有効な機能はどれか？

- ○ **A** オブジェクトの暗号化
- ○ **B** アクセスログの取得
- ○ **C** オブジェクトのバージョニング
- ○ **D** オブジェクトのリージョン内複製

Q6 S3 オブジェクトの誤削除に対する有効な機能／設定はどれか？正しい選択肢を**全て**選べ。

- □ **A** オブジェクトのバージョニング
- □ **B** オブジェクトのリージョン内複製
- □ **C** アクセスコントロールリスト（ACL）
- □ **D** バケットポリシー

Q7 Glacier に格納すべきデータとして適切なものはどれか？

- ○ **A** 社内の従業員に公開する動画
- ○ **B** 世界中のエンドユーザに公開する動画
- ○ **C** ゲノムのシーケンスデータ
- ○ **D** DB のオンライントランザクションログ

答え

A1 C

スタンダード S3 は、耐久性が非常に高いストレージのため、失ってはならないオリジナルデータの格納先として適しています。加工データについては、オリジナルデータから再加工（再作成）可能なため、RRS（低冗長化）S3 ストレージに格納することで、コストの低減を図れます。

A2 D

S3 は、オブジェクトの上書き／削除という操作に対して結果整合性の問題があります。そのため、頻繁に上書きされるオンライントランザクションログの格納先として利用するのは適切ではありません。

A3 B、C

S3 は、新規オブジェクトの格納については、書込み後の読込み整合性があるため、「完了」と表示されれば正しくデータを表示／取得できます。
一方、既存オブジェクトの上書き／削除については、結果整合性の問題があるため、「完了」と表示されても以前のデータが表示／取得されることがあります。

A4 B

他の AWS アカウントと IAM ユーザを指定したアクセス制限ができるのは、バケットポリシーです。

A5 D

オブジェクトの暗号化やアクセスログの取得、バージョニングは、ユーザが任意で有効にする機能です。
S3 バケットに格納したオブジェクトは、自動的にリージョン内で複製されます。

A6 A、D

S3 のバージョニング機能を有効にすることで、オブジェクトを誤って削除しても復元することができます。
オブジェクトは自動的にリージョン内で複製されていますが、削除操作に対する防衛策となるものではないため、たとえ誤った削除であっても、複製されているすべてのデータが削除されます。
アクセスコントロールリスト（ACL）で削除操作を禁止することはできません。

A7 C

Glacier に格納されているデータは、そのままでは参照することができないため、公開するデータの格納先として適していません。
Glacier は、変更されるデータの格納先ではなく、あまり参照されることのないデータのアーカイブ／長期バックアップ先として利用します。

第 7 章

データベース (RDS／ElastiCache／DynamoDB)

データベースについて

AWS には、マネージド型のデータベースサービスがあり、これらのサービスでは OS や DBMS のインストールなど様々な運用管理作業は不要です。マネージド型のデータベースサービスには、利用者の負荷を軽減しながら冗長性などを簡単に確保できる機能が豊富で、認定試験においても、そのメリットや特徴について出題されます。

本章では、このマネージド型の 3 つのサービス、具体的にはリレーショナルデータベースのサービスである RDS、キャッシュのサービスである ElastiCache、そして NoSQL のサービスである DynamoDB について解説します。

7-1 マネージドサービス

マネージドサービスとは、利用者が自身でOSやミドルウェア／ソフトウェアをインストールすることなくサービスを利用でき、サービスの可用性や拡張性、バックアップやパッチ適用といった管理作業の多くをAWSが管理してくれるサービスのことです。サービスの差別化につながらない構築／管理作業をAWSに任せることにより、利用者はコアとなる作業に集中することができます。

例えば、本章で紹介するマネージド型のリレーショナルデータベースサービスであるAmazon Relational Database Service（以下**RDS**）であれば、利用者はデータベースのインスタンスタイプ（スペック）とデータベースエンジン（OracleやMySQLなど）、フェイルオーバーの有無、自動バックアップの取得時刻などの設定を選択するだけで、各種設定の済んだRDSインスタンスが起動します。起動したRDSインスタンスに接続すると、すぐにSQLを実行してテーブルなどを作成していくことができ、設定した時刻に自動バックアップが取得されたり、障害が発生したときには自動でフェイルオーバーします。

一方、利用者が自身でEC2インスタンス上にDBMSをインストールする場合、バックアップや冗長性の担保について、利用者自身で設計／設定し、運用後の管理もすべて利用者が行う必要があります。

マネージドサービスは、データベース以外にも負荷分散サービスの**ELB**（Elastic Load Balancing）やキューサービスのAmazon **SQS**（Simple Queue Service）など、様々なものが存在します。マネージドサービスをうまく活用することで、システムの構築／運用／管理コストを抑えることができます。

7-2 マネージド型データベースサービス

AWSが提供するマネージド型データベースサービスには、次の4種類あります。

① リレーショナルデータベースサービス：Amazon **RDS**
② NoSQLデータベースサービス：Amazon **DynamoDB**
③ インメモリキャッシュサービス：Amazon **ElastiCache**
④ データウェアハウスサービス：Amazon **Redshift**

データ形式やデータサイズ、データベースへのクエリの頻度や応答性能など、データベースに格納されるデータは様々です。そのため、用途に応じてデータベースを使い分ける必要があり、AWSでも上記のように4種類のデータベースサービスを提供しています。このうち、本書では、RDSとDynamoDB、そしてElastiCacheを扱います。

7-3 RDS

RDSは、リレーショナルデータベースのマネージドサービスで、差別化につながらない構築／管理作業をAWSに任せて、可用性の高いリレーショナルデータベースを利用できます。RDSで選択できるデータベースエンジンは、次の6種類です。

- **Amazon Aurora**
- **MySQL**
- **MariaDB**
- **PostgreSQL**
- **Oracle**
- **Microsoft SQL Server**

Amazon Aurora（以下**Aurora**）は、MySQLと互換性のあるAWS独自のリレーショナルデータベースエンジンで、RDSが持つ様々な機能を活かすことができます。また、MariaDBは、MySQLからフォーク（分岐）して作られているオープンソースのリレーショナルデータベースエンジンです。

RDSには様々な機能／特徴があり、データベースエンジンそれぞれの特徴もあります。ここでは、認定試験でも出題されるRDSの主要な機能／特徴を

説明します。

① **マルチ AZ 配置**

マルチ AZ 配置は、その名の通り、複数の AZ に RDS インスタンスを配置して可用性を高める機能です。Aurora 以外のマルチ AZ 配置は、図 7-3-1 のように、RDS インスタンスのマスターが存在する AZ とは異なる AZ に同期スタンバイのスレーブを配置します。

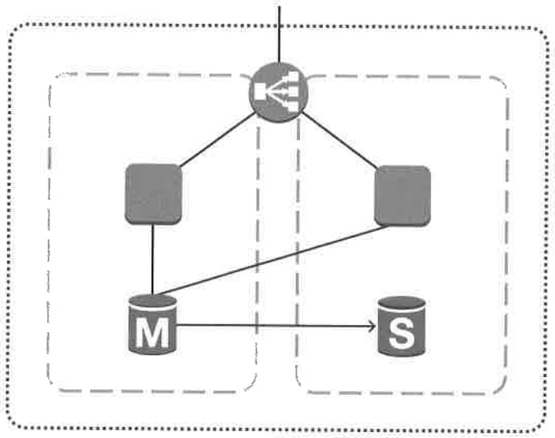

図 7-3-1　RDS のマルチ AZ 配置

MySQL、MariaDB、PostgreSQL、Oracle では、**同期物理レプリケーション**という仕組みを使い、スレーブのデータをマスターに合わせて最新の状態に維持します。

一方、SQL Server では、SQL Server のミラーリング機能である**同期論理レプリケーション**を使用して、その他のデータベースエンジンと同様に、スレーブのデータをマスターに合わせて最新の状態に維持しています。データの読み書きはマスターのみ可能で、スレーブについては読み取りもできない完全なスタンバイになるため、データベースの読み取り性能を上げたい場合はリードレプリカ（図 7-3-2）を作成するか、あるい

はデータベースキャッシュサービスである ElastiCache を配置します。ElastiCache については、7-5 で述べます。

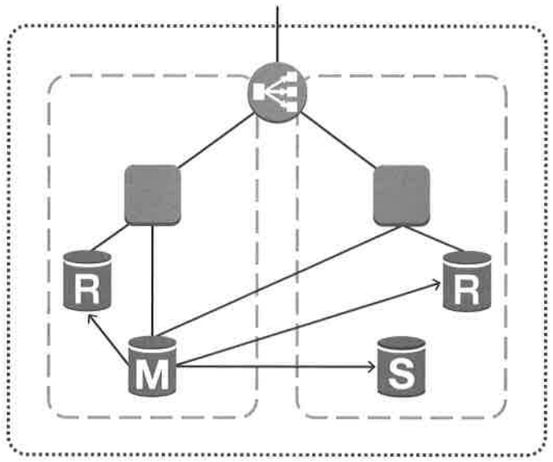

図 7-3-2　RDS のリードレプリカ

マルチ AZ 配置の RDS インスタンスのマスターが停止あるいは障害が発生した場合は、自動的にスレーブへのフェイルオーバーが開始されます。フェイルオーバーの過程で RDS インスタンスの CNAME がマスターからスレーブに付け替えられます。そのため、アプリ側ではデータベースと再接続するだけで済み、RDS インスタンスへの接続設定を変更する必要はありません。マスターからスレーブへのフェイルオーバーのタイミングは、マスターの障害時だけではなく、パッチ適用などのメンテナンス時や、手動の RDS インスタンスの再起動時に発生します。フェイルオーバーは、通常は数分で完了します。

Aurora のマルチ AZ 配置は、マスター－スレーブ構成ではなく、3 つの AZ にまたがるクラスターボリュームが作成され、各 AZ にクラスターデータのコピーが格納されます。図 7-3-3 のように、ある AZ に読み書きが可能なプライマリインスタンスが作成され、他の AZ には読み取り専用

のリードレプリカが作成されます。プライマリインスタンスに障害が発生しても、プライマリインスタンスとは独立したキャッシュを利用しつつ、プライマリインスタンスが瞬時に障害から回復するよう設計されています。

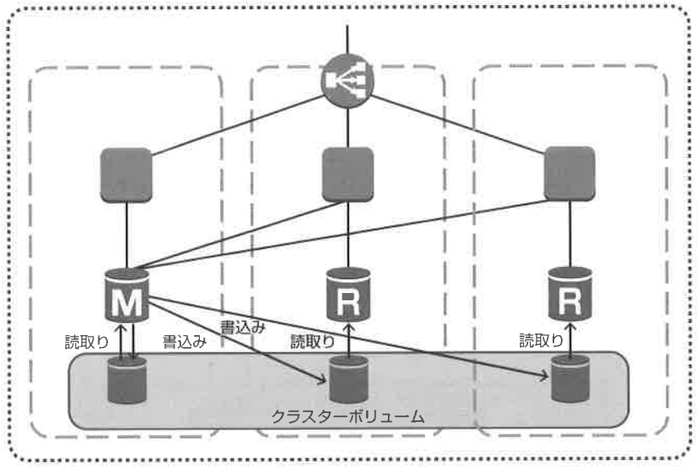

図 7-3-3　Aurora のクラスター

試験のポイント！

RDS のマルチ AZ 配置の特徴、フェイルオーバー時の挙動を押さえる！

② 自動バックアップ機能

自動バックアップ機能は、RDS の標準機能です。これは、1 日 1 回自動的にデータのバックアップ（スナップショット）を取得するものです。バックアップの取得中は、多少の読み書き遅延が発生する場合があり、利用者はバックアップウィンドウと呼ばれる設定項目でバックアップが取得される時間帯を指定できます。自動バックアップの保持期間は、デフォルト 7 日ですが、0〜35 日の間で指定できます（0 日を指定すると、バックアップは取得されません）。RDS では、1 日 1 回の自動バックアップの他

に、トランザクションログを自動的に取得しており、1 日 1 回の自動バックアップとトランザクションログを利用して、設定している保存期間（1〜35 日）の特定時点のデータを持つ RDS インスタンスを復元することができます。トランザクションログは 5 分に 1 回永続ボリュームに書き込まれているため、復元できる最新時刻は復元作業時点から過去 5 分以内の任意の時刻です。なお、利用者が任意のタイミングでバックアップ（スナップショット）を取得することもでき、このバックアップは利用者が明示的に削除するまで保持されます。

試験のポイント！

RDS の自動バックアップ機能のメリットを押さえる！

③ パッチ適用

RDS には自動パッチ適用の機能があり、この機能を有効にしておくことで、メンテナンスウィンドウと呼ばれる設定項目で指定された曜日／時間帯にパッチが適用されます。パッチ適用時に数分のダウンタイムが生じることがありますが、RDS をマルチ AZ 配置にすることで、先にスタンバイにパッチが適用され、フェイルオーバーしたのちに旧マスターでパッチが適用されるため、その影響を軽減することができます。自動パッチ適用は、利用者が有効／無効を設定できますが、重要なセキュリティ脆弱性が発生した場合には、自動パッチ適用を無効化していても、自動的に適用されることがあります。

④ ストレージ

RDS のストレージも EC2 のストレージである EBS と同様に、General Purpose SSD、Provisioned IOPS SSD、そして Magnetic（磁気ディスク）の 3 タイプがあります。ストレージのサイズは、Aurora 以外のデータベースエンジンでは、最小が Magnetic の 5GB、最大が General Purpose SSD／Provisioned IOPS SSD の 6TB ですが、データベースエンジンによってもそれぞれ異なります。Aurora では最小が 10GB で、データベースの使用量に応じて 10GB 単位で最大 64TB まで拡張されます。性能については、Aurora 以外のデータベースエンジンでは EBS と

同様で、General Purpose SSD は 1GB あたり 3IOPS のベースラインパフォーマンスがあり（最大 10,000IOPS）、3,000IOPS までのバースト機能があります。Provisioned IOPS SSD は容量／データベースエンジンによって指定できる IOPS の値が異なりますが、最大 30,000IOPS までの値を指定できます。

RDS は AZ サービスであり、EC2 インスタンスと同様に VPC のサブネット内に RDS インスタンスを起動し、セキュリティグループとサブネットのルーティングルール（プライベートサブネットに配置する）によってアクセスを制限します。

7-4 DynamoDB

DynamoDB は、マネージド型の NoSQL データベースサービスで、利用者はソフトウェアをインストール（構築）／管理することなく利用できます。DynamoDB には、次のような特徴があります。

- ストレージ容量が必要に応じて自動的に拡張
- 秒間あたりの I/O 性能（スループット）を指定できる
- ストレージは SSD のみで安定した I/O 性能を提供
- データを 3 つのデータセンタに複製することで高可用性と高い耐久性を提供
- 読み込み整合性の強弱を指定することで、性能と整合性のバランスを選択可能

NoSQL データベースの特徴として、リレーショナルデータベースでは難しい拡張性が挙げられます。DynamoDB では、ストレージについては事前の容量を指定する必要がなく、利用者がテーブルに項目（データ）を書き込んでいけば、必要に応じてストレージ容量が自動的に拡張し、利用した分だけの課金になります。また、性能については、テーブルごとに秒間あたりの読み書きスループットを指定でき、この値はデータベース使用中に変更するこ

とができます。また、AWS マネージドサービスの特徴ともいえる冗長性ですが、DynamoDB ではテーブルのデータを地理的に離れた 3 か所のデータセンタに複製するため、利用者側で冗長性を考慮する必要はありません。また、DynamoDB の整合性は結果整合性になります。つまり、あるデータを書き込み、すぐに読み込んだとき、最新の書込み結果が反映されない場合があり、時間が経つと最新の結果が反映されます。アプリケーションの要件によっては、結果整合性よりも強力な整合性が求められることもあり、DynamoDB で「強い整合性」オプションを指定すると、最新の書込み結果がすべて反映されたデータを読み取るようになります。ただし、「強い整合性」オプションを有効にすると、読込みスループットが半減してしまうため、できるだけデフォルトの結果整合性で要件を満たすようにアプリケーションを設計します。

DynamoDB は、テーブルに格納できる項目数やデータ容量の制限がなく、拡張性が非常に高いのですが、1 つ 1 つの項目のサイズは 400KB を超えることはできません。そのため、ユースケースとしては、1 つ 1 つの項目のデータサイズは小さいものの、項目数が多くなる次のようなケースが挙げられます。

- セッションデータ
- ゲームの点数
- 買い物リスト（買い物かご）
- センサーデータ

1 つ 1 つの項目に対応する実データサイズが大きくなるような場合には、実データを S3 に格納し、DynamoDB には S3 の格納先 URL や格納（作成）日付といったメタデータを格納します。

試験のポイント！

DynamoDB のメリットとユースケースを押さえる！

DynamoDB はリージョンサービスであり、プライベートサブネットから DynamoDB のテーブルに新しい項目を追加したり、既存の項目を読み取ったりという操作を行うには NAT インスタンスが必要です。テーブルへの書き込みなど、DynamoDB に対する操作（図 7-4-1）は、IAM によりテーブルや項

目レベルのアクセス管理を行います。DynamoDB にアクセスするアプリケーションが EC2 インスタンス上で動作する場合は、DynamoDB へのアクセスが許可された IAM ポリシーが設定された IAM ロールを EC2 インスタンスに割り当てることで、DynamoDB へのアクセスを安全に制御できます。

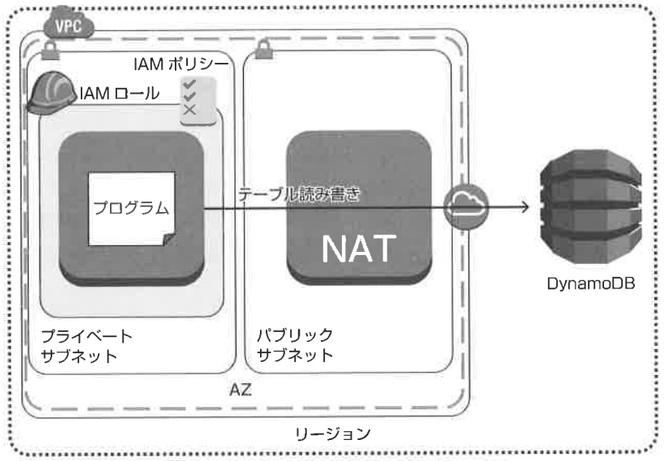

図 7-4-1　DynamoDB への読み書き

試験のポイント！

DynamoDB のアクセス制御は IAM で行い、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 ロールを活用する

7-5 ElastiCache

リレーショナルデータベースへのアクセス負荷が原因でアプリケーションのパフォーマンスが低下している場合、読取りであれば RDS のリードレプリカを利用することによりパフォーマンスを改善できます。また、その他に、ここで紹介する **ElastiCache** を利用する方法もあります。ElastiCache は、インメモリキャッシュのマネージドサービスで、**Memcached** と **Redis** の

2 つのキャッシュエンジンから選択して利用できます。

Memcached は、Key-Value Store 形式のインメモリキャッシュで、マルチノードのキャッシュクラスタを構成します。一方の Redis も Key-Value Store 形式のインメモリキャッシュですが、こちらはマスター－スレーブ構成になります。

ElastiCache は AZ サービスで、VPC のサブネットをグループ化したサブネットグループに配置します。アクセス制御は、セキュリティグループとサブネットのルーティングルール（プライベートサブネットに配置する）によってアクセスを制限します。

ElastiCache のユースケースとしては、図 7-5-1 のように RDS へのクエリ結果をキャッシングして RDS へのアクセス負荷を軽減することによる読書き性能向上や、DynamoDB と同様のセッションデータの格納があります。

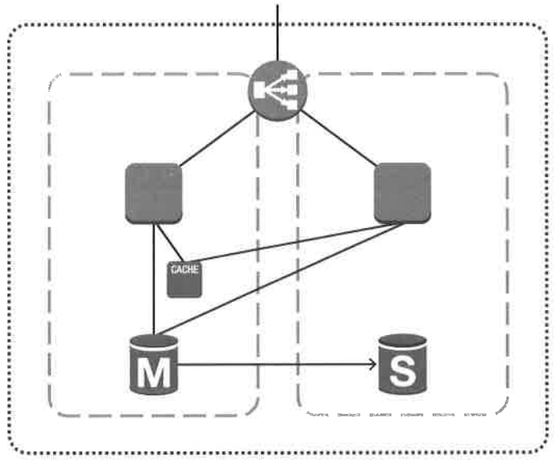

図 7-5-1　ElastiCache のユースケース

試験のポイント！

ElastiCache のメリット／ユースケースを押さえる！

章末問題

Q1 マルチAZ配置したRDSインスタンスでフェイルオーバーが発生した際に、利用者側で実施しなければいけない操作として正しい選択肢はどれか？

- ○ **A** マスターとして動作していたインスタンスに割り振られていたプライベートIPアドレスを、スレーブとしてスタンバイしているインスタンスに割り振る。
- ○ **B** マスターとして動作していたインスタンスを再起動し、スレーブとしてスタンバイするように設定する。
- ○ **C** RDSインスタンスへの読み書きを行っていたアプリケーションのRDSインスタンスの接続先を、マスターのプライベートIPアドレスからスレーブのプライベートIPアドレスに変更し、再度デプロイする。
- ○ **D** 特に何も操作する必要はない。

Q2 オペレーションミスによるデータ損失に対する有効なRDSの機能／設定はどれか？

- ○ **A** マルチAZ配置
- ○ **B** 自動バックアップ
- ○ **C** パッチ適用
- ○ **D** Provisioned IOPS

Q3 DynamoDBの特徴／メリット／ユースケースとして正しい選択肢を**全て**選べ。

- □ **A** DynamoDBは、データを3カ所のデータセンタに冗長的に格納するため耐久性が非常に高く、Key-Value Store形式に対応したNoSQLデータベースである。その上、拡張性があり、格納したデータに合わせて自動的に拡張するため、S3の代替用途として利用できる。
- □ **B** DynamoDBテーブルへのアクセスはデフォルトですべてのアクセスが拒否されているため、セキュリティグループでアクセスを受け付ける必要がある。
- □ **C** DynamoDBは拡張性があり、格納したデータに合わせて自動的に拡張する上、高いI/O性能を有している。そのため、頻繁に書き込みが発生する大量のデータの格納先として適している。
- □ **D** DynamoDBは拡張性があるものの、項目の最大サイズは限られている。そのため、大きなデータサイズのデータを扱うには、実データについてはS3に保管し、DynamoDBにはメタデータを格納する。

Q4 ElastiCacheの特徴／メリット／ユースケースとして正しい選択肢はどれか？

- ○ **A** ElastiCacheはインメモリキャッシュとして高速に読み書きできるため、ログインが必要な会員Webサイトのユーザのセッション情報の格納先として適している。
- ○ **B** ElastiCacheに対してデータを読み書きするアプリケーションがEC2インスタンス上で動作している場合、ElastiCacheへのアクセスを許可したIAMポリシーを設定したIAMロールをEC2インスタンスに割り当てることで、ElastiCacheへのアクセス制御を安全に管理することができる。
- ○ **C** ElastiCacheはリージョン内の3か所のデータセンタのサーバにデータをキャッシングすることで、利用者は3か所のうちの最も近いデータセンタからデータを低レイテンシーでダウンロードすることができる。
- ○ **D** ElastiCacheはインメモリキャッシュとして高速に読み書きできるため、Webサーバの前段に配置することにより、キャッシングしているデータを高速に利用者に応答するメリットがあるほか、Webサーバへの負荷を軽減することができる。

答え

A1 D

マルチAZ配置しているRDSインスタンスがフェイルオーバーする際、RDSインスタンスのCNAMEがマスターからスレーブに自動的に付け替えられるため、利用者側で特に操作することはありません。

A2 B

RDSの自動バックアップ機能により、自動バックアップの保持期間内の任意の時刻のデータを復元することができます。

A3 C、D

A　DynamoDBは1つ1つの項目のサイズが400KBまでに限られているため、S3の代替として利用することはできません。大きな実データ自体はS3に保管し、DynamoDBにはそのメタデータを格納するという利用用途が適しています。

B　DynamoDBはリージョンサービスであり、アクセス制御はセキュリティグループではなく、IAMで行います。

A4 A

B　ElastiCacheはセキュリティグループでアクセス制限します。

C　ElastiCacheはAZサービスであり、VPC環境ではEC2からのアクセスのみ受け付けます。

D　ElastiCacheはRDSの負荷を軽減するメリットがあります。

第8章

AWSにおける監視と通知（CloudWatch／SNS）

AWSにおける監視について

AWSにはCloudWatchという監視（モニタリング）サービスがあり、EC2インスタンスを始めとした様々なAWSリソースをモニタリングすることができます。CloudWatchによるモニタリングの結果を受けて、運用者にメールで通知をしたり、EC2インスタンスの増減アクションを発生させたりするなど、AWSの特徴の1つである伸縮自在性／柔軟性（アジリティ）を活かす上でも、CloudWatchのモニタリングは重要です。

本章では、AWSにおける監視と通知について、モニタリングサービスのCloudWatchと通知のサービスであるSNSを説明します。

8-1 CloudWatch によるモニタリング

あらゆるシステムにおいて、**監視（モニタリング）**は必要不可欠な要素です。通常の死活／性能監視などの他、AWS の特徴の 1 つである**伸縮自在性／柔軟性（アジリティ）**を活かす上でもモニタリングは重要です。負荷の増減に応じて EC2 インスタンスの数を増減させることで、無駄なリソースを省きコスト削減を図ったり、必要になったときにはリソースを増やして機会損失を避けたりすることができます。AWS には、Amazon **CloudWatch**（以下 CloudWatch）というモニタリングサービスがあり、CloudWatch のモニタリング結果から、EC2 インスタンスの数の増減アクションを発生させることができます。

CloudWatch によるモニタリングを理解する上で、必ず押さえなくてはいけない「**メトリックス**」という用語があります。メトリックスは「監視項目」という意味で、例えば次のようなメトリックスがデフォルトで集計されています。

- EC2 インスタンスの CPU 利用率
- EBS のディスク I/O
- S3 の格納オブジェクト総数
- RDS インスタンスの CPU 利用率
- RDS インスタンスのメモリ空き容量
- RDS インスタンスのストレージ空き容量
- DynamoDB に書き込まれたユニット数

など

CloudWatch は、各種 AWS リソースから送られてきたモニタリングデータを保存し、メトリックスごとにグラフ化して表示することができます。モニタリングデータの保持期間は 2 週間で、それ以降のデータは破棄されてしまうため、月次のモニタリングレポートが必要な場合は、保持期間内に CloudWatch からモニタリングデータをダウンロードしておく必要があります。

8-2 EC2 のモニタリング

CloudWatch は、AWS リソースから「送られてきた」モニタリングデータを保存／可視化するサービスであるため、CloudWatch にデータを送る仕組み／機能を EC2 インスタンスや RDS インスタンスなどの AWS リソース側で用意する必要があります。RDS はマネージドサービスであり、デフォルトで様々なモニタリングデータを収集して CloudWatch に送信するエージェントがインスタンスに導入されています。一方の EC2 はマネージドサービスでないため、デフォルトではハイパーバイザが収集できるモニタリングデータのみを収集して CloudWatch に送信しています。このことから、マネージドサービスではない EC2 のメトリックスは、次の 2 種類に大別できます。

標準（デフォルト）メトリックス：ハイパーバイザが取得して CloudWatch に送信するメトリックス
カスタムメトリックス：OS にインストールしたエージェントが取得して CloudWatch に送信するメトリックス

EC2 の標準メトリックスは、次の 12 種類です。（CPU クレジットに関するメトリックスは、t タイプのインスタンスファミリーのみです）

- CPU クレジット利用数（CPUCreditUsage）
- CPU クレジット累積数（CPUCreditBalance）
- CPU 利用率（CPUUtilization）
- 1 秒あたりの Disk 読込み回数（DiskReadOps）
- 1 秒あたりの Disk 書込み回数（DiskWriteOps）
- インスタンスストレージの読取りバイト数（DiskReadBytes）
- インスタンスストレージの書込みバイト数（DiskWriteBytes）
- 受信したバイト数（NetworkIn）
- 送信したバイト数（NetworkOut）
- OS／インフラストラクチャステータスチェックの成功（0）／失敗（1）（StatusCheckFailed）

- OS ステータスチェックの成功 (0)／失敗 (1)（StatusCheckFailed_Instance）
- インフラストラクチャステータスチェックの成功 (0)／失敗 (1)（StatusCheckFailed_System）

メモリの利用率など、上記以外にも取得したいメトリックスがある場合は、カスタムメトリックスとして取得する必要があります。

CloudWatch はモニタリングデータをグラフ化したり、ダウンロードしたりできますが、そのデータのプロット間隔は 1 分間隔や 5 分間隔など、AWS リソースによって異なります。EC2 では、次の 2 種類の間隔で取得可能で、それぞれ名前が付いています。

基本モニタリング：3 種類のステータスチェックは 1 分間隔、その他は 5 分間隔
詳細モニタリング：標準メトリックスをすべて 1 分間隔（ただし追加料金が必要）

試験のポイント！
EC2 の標準メトリックスや基本／詳細モニタリングを押さえる！

8-3 アラームとアクション

CloudWatch の各メトリックスに対して、アラームを設定することができます。アラームとは、事前に設定しておいた閾値（しきい値）を超えたときに所定の動作（アクション）を呼び出す状態遷移のことで、例えば EC2 インスタンスの CPU 利用率が 70% という閾値を超えたらアラーム状態に遷移する、あるいは EC2 インスタンスのステータスチェックで 1（失敗）を 1 回でも検出したら（1 回という閾値を超えたら）アラーム状態に遷移するという使い方ができます。CloudWatch で発生したアラームを受けて、次のようなアクションを呼び出すことができます。

- メールなどの通知（Simple Notification Service（詳細は後述します）
- Auto Scaling ポリシー（EC2 インスタンス数の増減。詳細は後述します）
- EC2 アクション（停止／削除／再起動／復旧）

このような CloudWatch アラームのアクションを利用して、7 つのベストプラクティスの 1 つであり、AWS の特徴／メリットでもある、「伸縮自在性を実装」を実現できます。また、ステータスチェック（死活監視）のアラームを受けた EC2 インスタンスの復旧など、システムの可用性を高めるためにも、CloudWatch アラームのアクションを利用できます。

アラームには、次の 3 つの状態があり、たとえば図 8-3-1 のようにその状態が遷移します。利用者が設定した期間アラームが持続すると、アクションが呼び出されます。

- OK
- アラーム（ALARM）
- 不足（INSUFFICIENT_DATA）

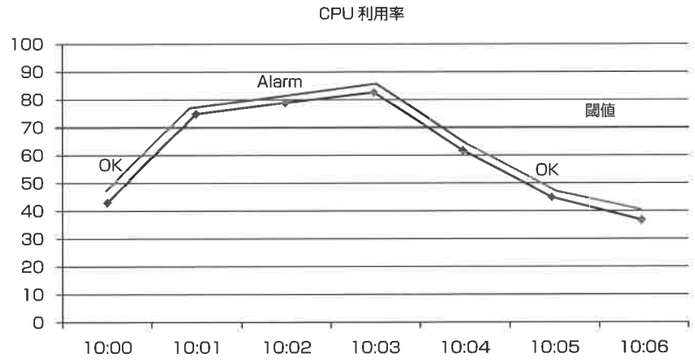

図 8-3-1　アラームの状態

例えば、EC2 インスタンスの 1 分間の CPU 利用率の平均が閾値の 70% を 3 期間連続（3 分間連続）で上回っている場合に、EC2 インスタンスを 2 台増やす Auto Scaling ポリシー（詳細は後述します）のアクションを呼び出すとい

う設定ができます。あるいは、4 台の EC2 インスタンスの 1 分間の CPU 利用率の平均が閾値の 30% を 3 期間連続（3 分間連続）で下回っている場合に、EC2 インスタンスを 2 台減らす Auto Scaling ポリシーのアクションを呼び出す設定も同時に行えます。この 2 つの設定は、それぞれ閾値が異なるため、異なるアラームになります。

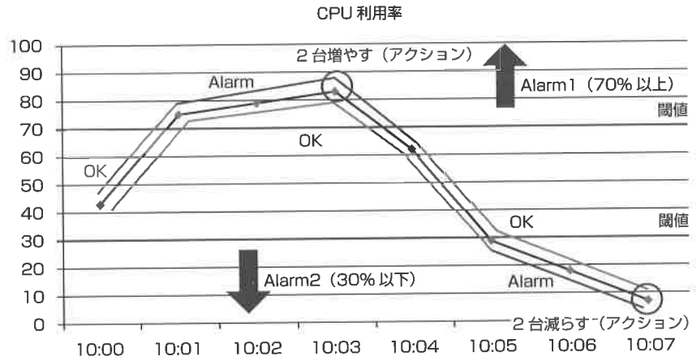

図 8-3-2　Auto Scaling ポリシーとアクションの呼び出し

後述する Auto Scaling により伸縮自在性を実装した場合、1 つのメトリックスに対して 2 つのアラームが設定される場合が多く、両者のアラームの状態が図 8-3-2 のように遷移します。

重要！

CloudWatch のアラームとアクションについて、特徴と代表的な利用の流れを押さえる！

8-4 SNS

AWS には、Amazon Simple Notification Service（以下 **SNS**）という通知のサービスがあり、これによりユーザやアプリケーションにメッセージを送信することができます。送信は任意のタイミングで行えるほか、前述の

CloudWatch アラームのアクションとしてメッセージを通知することができます。

SNS を利用するには、次の 3 つの用語を押さえる必要があります。

- **メッセージ**：通知するメッセージ
- **サブスクライバ**：受信者を指し、サポートされているプロトコルは次のとおり
  - E メール
  - SMS（ショートメッセージサービス）
  - モバイルプッシュ
  - HTTP／HTTPS
  - SQS（Simple Queue Service の略称で詳細は後述します）
  - Lambda（サーバ無しのプログラムコード実行サービス）
- **トピック**：単一／複数のサブスクライバをまとめたもの

例えば、前述の CloudWatch のアラームのアクションで SNS を利用する流れは、次のようになります。

① SampleTopic というシステムのアラートが通知されるトピックを作成し、運用管理者の E メールアドレス／メーリングリストをサブスクライバとして SampleTopic に登録
② CloudWatch でシステムの EC2 インスタンスをモニタリングし、1 分間の平均 CPU 利用率が 80% を 1 回超えたというアラームが発生すると、SampleTopic にメッセージを通知するアクションを設定
③ 該当する EC2 インスタンスの CPU 利用率が 80% を超えてアラームが発生すると、「CPU 利用率が 80% を超えてアラームの状態が OK から ALARM に遷移した」というメッセージを SampleTopic に送信
④ SNS は SampleTopic に登録されている運用管理者の E メールアドレス／メーリングリスト（サブスクライバ）にメッセージを送信

章末問題

Q1 Web サーバとして利用している EC2 インスタンスの標準メトリックスとして正しい選択肢はどれか？

- ○ **A** メモリ使用率
- ○ **B** Web ページへのロード時間
- ○ **C** Web サーバのプロセス／スレッド数
- ○ **D** Netowrk I/O

答え

A1 D

EC2 インスタンスの標準メトリックスは、ハイパーバイザが取得できる値で、Network I/O などがあります。これに対し、メモリの使用率は OS で収集する必要があります。

第 9 章

AWS における拡張性と分散／並列処理 (ELB／Auto Scaling／SQS／SWF)

AWS における拡張性と分散／並列処理について

AWS の特徴の 1 つに、伸縮自在性／柔軟性（アジリティ）があります。AWS では、必要なときに必要なだけ IT リソースを用意し、必要が無くなれば解放することで、コストメリットが図れるほか、経営判断から開発／サービス提供開始までの時間を短縮することができます。

そして、この特徴を活かす上で欠かせないのが、本章で紹介する ELB、Auto Scaling、SQS、SWF といったサービス／機能です。これらのサービス／機能は、7 つのベストプラクティスを実践する上でも欠かせないもので、認定試験において様々な分野にまたがって出題されます。

本章では、AWS における拡張性と分散／並列処理について、そこに関わるサービス／機能である ELB と Auto Scaling、SQS そして SWF を中心に説明します。