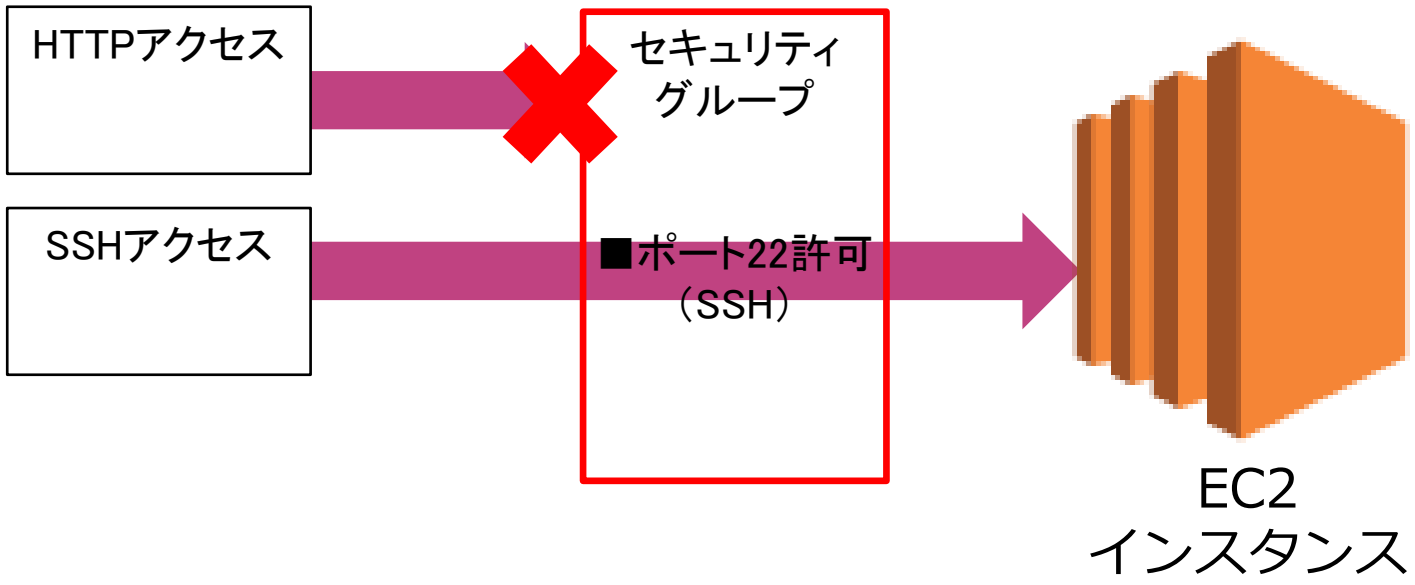
セキュリティグループの出題範囲

セキュリティグループとは何か？

インスタンスへのトラフィックのアクセス可否を設定するファイアーウォール機能を提供



セキュリティグループの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

セキュリティグループの特徴	✓ セキュリティグループがどのトラフィック通信を制御するのかなどの特徴に関する質問が出題される。
デフォルト設定	✓ デフォルトのセキュリティグループの設置内容が問われる
SSH接続	✓ EC2インスタンスの基本的な構成であるSSH接続のトラフィック制御設定に関して問われる。
カスタムソースの利用	✓ セキュリティグループのソースの設定方法が問われる。
ELBの設定	✓ ELBとEC2インスタンスを構成する際のセキュリティグループの設定方法が問われる。

セキュリティグループの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

RDSのセキュリティグループの設定	✓ RDSとEC2インスタンスを構成する際のセキュリティグループの設定方法が問われる。

[Q]セキュリティグループの特徴

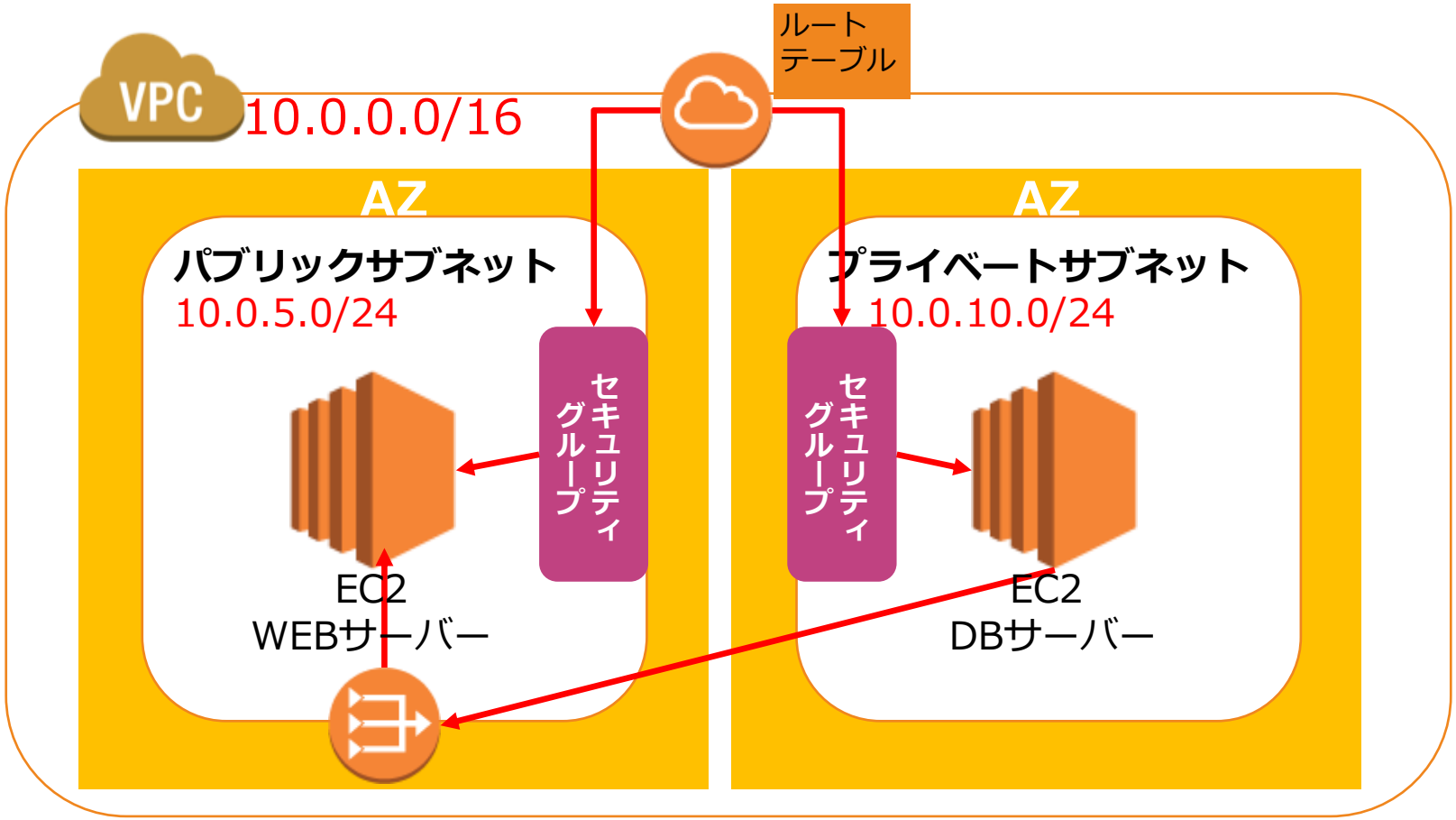
A社はWebサーバとデータベースサーバからなるWebアプリケーションを構築しているところです。 Webサーバとデータベースサーバはそれぞれ異なるサブネットに設置された異なるEC2インスタンスを利用して設定します。 セキュリティのために、データベースサーバがWebサーバからのトラフィックのみを許可する必要があります。

この要件を満たすための対応を選択してください。

1) VPCエンドポイントでトラフィックを制御する
2) セキュリティグループでトラフィックを制御する
3) ネットワークACLでトラフィックを制御する
4) IAMロールでWEBサーバにデータベースサーバへのアクセスを許可する。

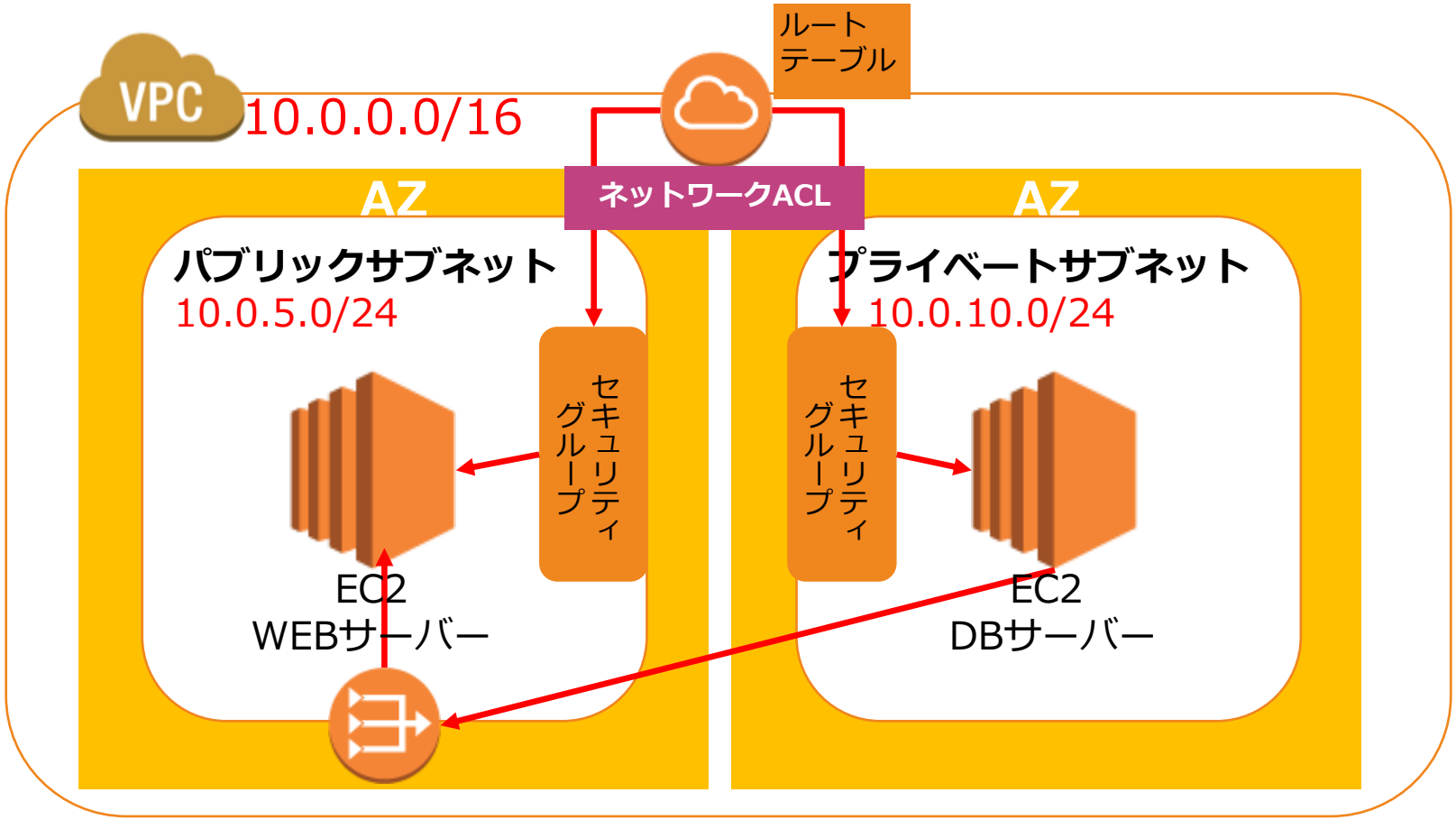
セキュリティグループ

EC2インスタンスに対するトラフィック制御を実施するファイル機能を提供



ネットワークACL

サブネットに対するトラフィック制御を実施する



セキュリティグループとネットワークACL

トラフィック設定はセキュリティグループまたはネットワークACLを利用する

セキュリティグループ設定

- サーバー単位で適用
- ステートフル：インバウンドのみ設定すればアウトバウンドも許可される。（状態を維持）
- 許可のみをIn/outで指定
- デフォルトでは同じセキュリティグループ内通信のみ許可
- 全てのルールを適用

ネットワークACLs設定

- VPC／サブネット単位で適用
- ステートレス：インバウンド設定だけではアウトバウンドは許可されない。
- 許可と拒否をIn/outで指定
- デフォルトでは全ての通信を許可する設定
- 番号の順序通りに適用

[Q]デフォルト設定

ソリューションアーキテクトが新規にAWSアカウントを作成し、アジアパシフィック（シドニー）リージョンを選択しました。 デフォルトのVPC内には、デフォルトのセキュリティグループがあります。

このデフォルトのセキュリティグループの初期設定はどれでしょうか？ （2つ選択してください）

1) インバウンドルールにおいて、全てのアドレスからの全てのトラフィックを許可している。
2) アウトバウンドルールにおいて、全てのアドレスからの全てのトラフィックを許可している。
3) アウトバウンドルールにおいて、同じセキュリティグループからの全てのトラフィックを許可している。
4) VPCのみにトラフィックを許可するアウトバウンドルールがある
5) インバウンドルールにおいて、同じセキュリティグループからの全てのトラフィックを許可している。

デフォルト設定

セキュリティグループを指定しない場合、Amazon EC2 はデフォルトのセキュリティグループを使用

デフォルト セキュリティ グループ	✓ セキュリティグループを指定しないでEC２インスタンスなどを起動するとデフォルトセキュリティグループが指定される。 ✓ 自身のセキュリティグループを設定されたリソースのみから全てのインバウンドアクセスを許可する設定がされている。
カスタム セキュリティ グループの 初期設定	✓ EC2インスタンスでインスタンスを起動時に最初に指定されるセキュリティグループでは、Linuxインスタンスならば SSH設定が0.0.0.0/0で設定されている。 ✓ DBインスタンスでインスタンスを起動時に最初に指定されるセキュリティグループでは、特定のIPアドレスが指定されている。

デフォルト設定

VPCのデフォルトセキュリティグループは同じセキュリティグループIDへの許可設定がされており、最初は通信不可

Inbound			
Source	**Protocol**	**Port range**	**Description**
セキュリティグループ ID (sg-*xxxxxxxx*)	すべて	すべて	同じセキュリティグループに割り当てられているネットワークインターフェイス（および関連付けられているインスタンス）からのインバウンドトラフィックを許可します。
Outbound			
Destination	**Protocol**	**Port range**	**Description**
0.0.0.0/0	すべて	すべて	すべての発信 IPv4 トラフィックを許可する
::/0	すべて	すべて	すべての発信 IPv6 トラフィックを許可するこのルールは、IPv6 CIDR ブロックを持つ VPC を作成するか、既存の VPC と IPv6 CIDR ブロックを関連付けている場合にデフォルトで追加されます。

Reference: https://docs.aws.amazon.com/ja_jp/vpc/latest/userguide/VPC_SecurityGroups.html#DefaultSecurityGroup

[Q] SSH接続

大手ECサイトではオンデマンドEC2インスタンスを利用してWEBサーバーを構築しています。このEC2インスタンスはパブリックサブネットに配置して、SSH接続を介して、特定のIPアドレス（130.178.101.46）からのみアクセスできることを確認する必要があります。

このアクセスを可能にするセキュリティグループの設定はどれでしょうか？

1) SSHプロトコルをUDPとポート22で選択し、ソースに130.178.101.46/32を設定する。
2) SSHプロトコルをUDPとポート22で選択し、ソースに130.178.101.46/0を設定する。
3) SSHプロトコルをTCPとポート22で選択し、ソースに130.178.101.46/32を設定する。
4) SSHプロトコルをTCPとポート22で選択し、ソースに130.178.101.46/0を設定する。

SSH接続

EC2インスタンスにSSH接続する際の設定はTCPプロトコルの22番ポート設定を利用する。

[Q]カスタムソースの利用

大手ECサイトではオンデマンドEC2インスタンスを利用してWEBサーバーを構築しています。 パブリックサブネットにWebサーバーを配置し、プライベートサブネットにデータベースサーバーを配置して、セキュリティグループでトラフィック制御を設定することが必要です。あなたはソリューションアーキテクトとして、セキュリティグループのインバウンドルールの設定をしているところです。

セキュリティグループをセットアップする際に無効なオプションはどれでしょうか？

1) インバウンドルールのカスタムソースとしてセキュリティグループを使用する。
2) インバウンドルールのカスタムソースとして、CIDRブロック表記のIPアドレスの範囲を使用する。
3) インバウンドルールのカスタムソースとしてIPアドレスを使用する。
4) インバウンドルールのカスタムソースとしてインターネットゲートウェイIDを使用する。

カスタムソースの利用

セキュリティグループのソースにはCIDRまたはセキュリティグループを指定して、アクセス対象を制御できる。

IPアドレスまたはCIDRを指定する。
or
セキュリティグループを指定する。

[Q] ELBのセキュリティグループの設定

大手ECサイトではオンデマンドEC2インスタンスを利用してWEBサーバーを構築しています。 パブリックサブネットにWebサーバーを配置し、プライベートサブネットにデータベースサーバーを配置して、ELBによるトラフィック分散を実施します。。ELBとWEBサーバーにセキュリティグループを設定して、ELBにはインターネットからのパブリックなアクセスを可能とし、WEBサーバーはELBからのアクセスのみに限定します。

この要件を満たすことができるELBのセキュリティグループの設定方法はどれでしょうか？（2つ選択してください）

1) ELBのセキュリティグループにHTTP / HTTPSを許可するインバウンドルールを追加し、ソースに0.0.0.0/0 “を指定する。
2) ELBのセキュリティグループにすべてのTCPを許可するアウトバウンドルールを追加し、ソースにインターネットゲートウェイを指定する
3) ELBのセキュリティグループにHTTP / HTTPSを許可するアウトバウンドルールを追加し、ソースにWebサーバーセキュリティグループを指定する。
4) ELBのセキュリティグループにHTTP / HTTPSを許可するインバウンドルールを追加し、ソースにWebサーバーセキュリティグループを指定する。
5) ELBのセキュリティグループにHTTP / HTTPSを許可するアウトバウンドルールを追加し、ソースを0.0.0.0/0を指定する。

ELBのセキュリティグループの設定

WEBサーバーとなるEC2インスタンスとELBを構成する場合は、WEBサーバーからのアウトバウンドアクセスに限定する。

インバウンド	✓ インターネットからのHTTP／HTTPSのトラフィックを許可 ✓ 誰でもアクセス可能なWEBサイトの場合はソースに0.0.0.0/0を指定
アウトバウンド	✓ WEBサーバーからのHTTP／HTTPSのトラフィックのみを許可 ✓ WEBサーバーからのアクセスに限定するため、ソースにWEBサーバーのIPアドレスを指定するか、WEBサーバーに設定されたセキュリティグループを指定する。

[Q] RDSのセキュリティグループ設定

大手ECサイトでは複数のオンデマンドEC2インスタンスを利用してWEBサーバーを構築しています。 パブリックサブネットにWebサーバーを配置し、プライベートサブネットにRDS PostgreSQLデータベースを配置して、セキュリティグループを設定しています。インターネットからのトラフィックはALBによってEC2インスタンスに分散しており、インターネットからはHTTPSからのアクセスのみを許可します。SSLの設定はALBを終端とする設定を実施します。

安全性を高めるためにセキュリティグループをどのように構成する必要がありますか？ （3つ選択してください）

1) RDSのセキュリティグループには、ポート5432のEC2インスタンスのセキュリティグループからのインバウンドルールを設定する。
2) EC2インスタンスのセキュリティグループには、ポート80のALBのセキュリティグループからのインバウンドルールを設定する。
3) ALBのセキュリティグループには、ポート443またはポート80のインバウンドルールに、ソースを0.0.0.0/0で設定する。
4) RDSのセキュリティグループには、ポート443またはポート80のEC2インスタンスのセキュリティグループからのインバウンドルールを設定する。
5) EC2インスタンスのセキュリティグループには、ポート443のALBのセキュリティグループからのインバウンドルールを設定する。
6) ALBのセキュリティグループには、ポート443のインバウンドルールに、ソースを0.0.0.0/0で設定する。

RDSのセキュリティグループの設定

WEBサーバーとなるEC2インスタンスとELBを構成する場合は、WEBサーバーからのアウトバウンドアクセスに限定する。

ALBのセキュリティグループ	✓ インターネットからのHTTPSまたはHTTPでトラフィックを許可して、ソースに0.0.0.0/0を指定する。
EC2のセキュリティグループ	✓ EC2インスタンスはALBからのインバウンドアクセスをHTTPSまたはHTTPで許可して、ソースにALBのセキュリティグループを指定する。
RDSのセキュリティグループ	✓ WEBサーバーとなるEC2インスタンスからPostgreSQLとの通信のためにはポート5432を許可する。 ✓ ソースにはEC2インスタンスのセキュリティグループを指定する。

RDSのセキュリティグループの設定

RDSの代表的なデータベースエンジンのポート番号は以下の通り

- MySQL： TCP IP通信　ポート番号3306
- PostgreSQL： TCP IP通信　ポート番号5432
- リモートデスクトップ：TCP IP通信　ポート番号3389

Kinesisの出題範囲

Kinesisとは何か？

ストリームデータ処理用の分析システムやアプリケーションを構築するサービス

Kinesisの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Kinesisの選択	✓ シナリオに基づいて、要件にあっているデータ処理サービスとしてKinesisを選択するという問題が問われる。
Kinesisの特徴	✓ Kinesisのデータ保持期間など、その特徴に関する問題が問われる。
Kinesisの基本構成	✓ シナリオに基づいて、Kinesisを利用したアプリケーションの構成やストリーム処理の構成が問われる。
Kinesisと連携するサービス	✓ Kinesis Data FirehoseとKinesis Data Streamsと連携することができるサービス対象が問われる。
アプリケーションの構築	✓ Kinesisを利用したアプリケーション構築方法に関する質問が問われる。

Kinesisの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Kinesisのスケーリング	✓ シナリオに基づいて、Kinesisのスケーリング方法に関する問題が問われる。

[Q]Kinesisの選択

大手メディア企業は、ニュースメディアを利用して広告収益を得たいと考えており、AWSを利用してメディアサイトを構築しました。広告のリアルタイム処理をするために、アクセス行動データを取得してリアルタイムデータ処理することでサービスを実現します。ソースからクリックストリームイベントをキャプチャして、データストリームをダウンストリームアプリケーションに同時にフィードする仕組みが必要です。

この要件を満たすためには、どのサービスを利用するべきでしょうか？

1）Amazon Kinesis Data Streamsを使用する。

2）Amazon SQSを使用する。

3）AWS Step Functionsを使用する。

4）Amazon SimpleWorkFlow（SWF）を使用する。

Kinesis

ストリームデータを収集・処理するためのフルマネージド型サービスで主に3つのサービスで構成される

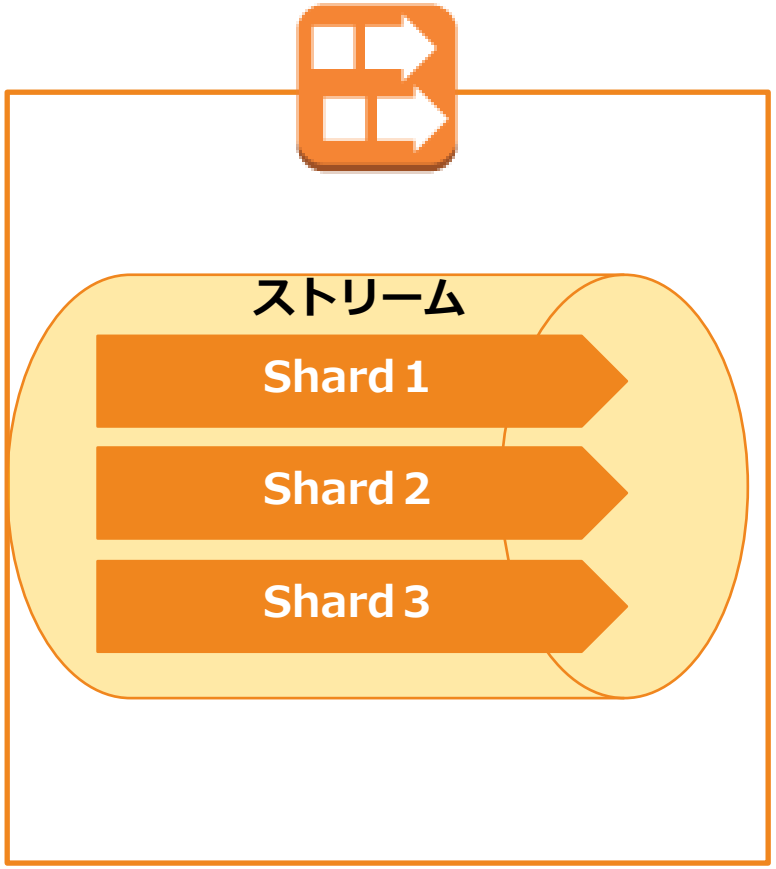
Amazon Kinesis Data Streams	Amazon Kinesis Data Firehose	Amazon Kinesis Data Analytics
ストリームデータを処理するアプリケーションを構築	ストリームデータをS3やRedshiftなどへ簡単に配信	ストリームデータを標準的なSQLクエリでリアルタイムに可視化・分析

Amazon Kinesis Data Streams

ストリームデータ処理用の分析システムやアプリケーションを構築するサービス

Amazon Kinesis Data Streams

ストリーミング処理をシャードに分けて分散させて実行するため高速処理が可能



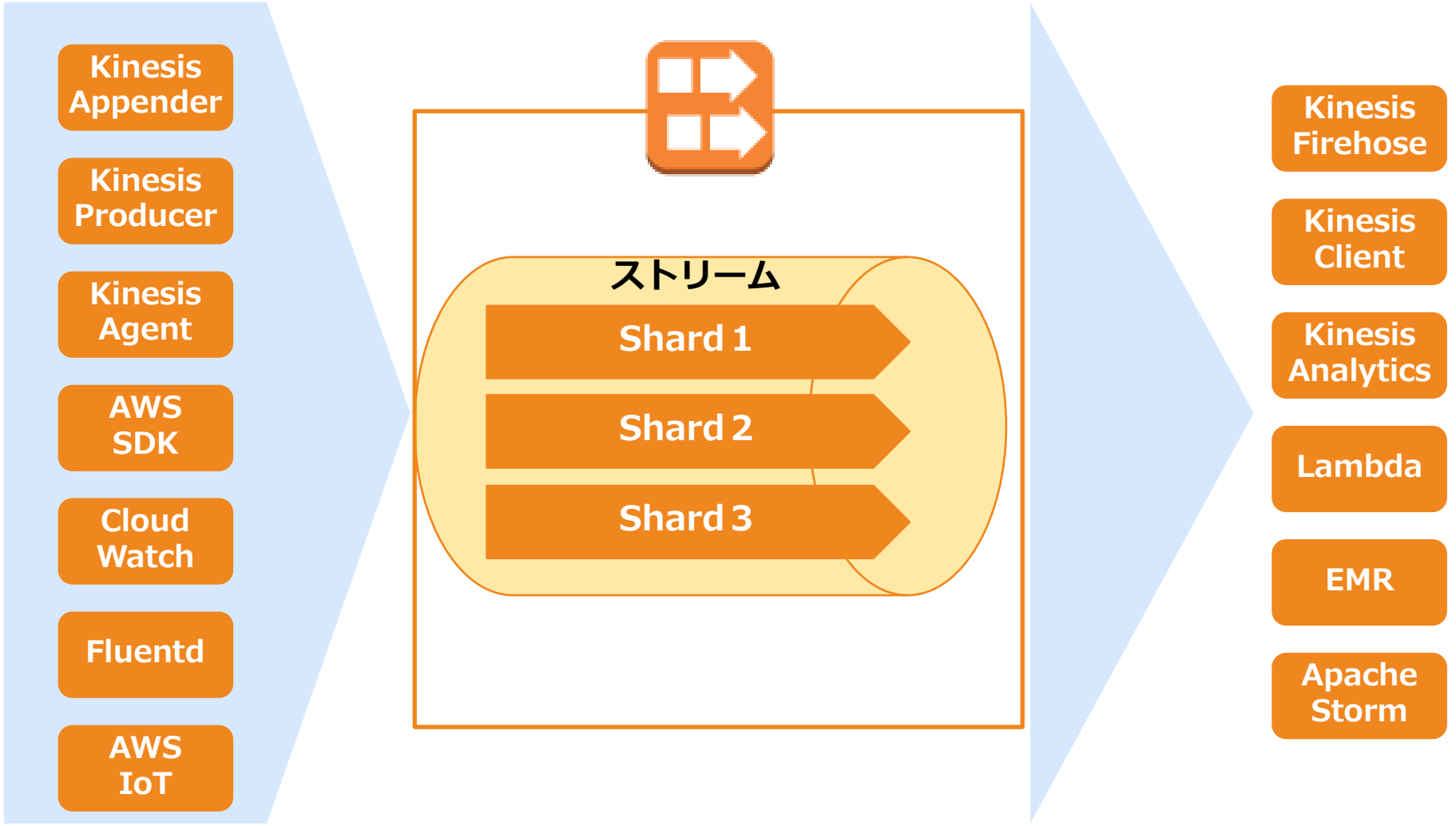
Amazon Kinesis Data Streams

Kinesis Data Streamsは以下の要素で成り立っている。シャード単位でパフォーマンスを向上させる。

シャード	Amazon Kinesis データストリームの基本的なスループットの単位 1 シャードは1 MB/秒のデータ入力と 2 MB/秒のデータ出力能力を提供 1 つのシャードは 1 秒当たり最大 1,000 件の PUT レコードをサポート
レコード	レコードはAmazon Kinesis データストリームに保存されるデータの単位 レコードはシーケンス番号、パーティションキー、データ BLOB で構成
データBLOB	データ BLOB はデータプロデューサーがデータストリームに追加する、処理対象のデータ 最大サイズは、1 メガバイト (MB)
パーティションキー	パーティションキーは、レコードを分離してデータストリームの異なるシャードにルーティングするために使用
シーケンス番号	シーケンス番号とは各レコードの一意の識別子

Amazon Kinesis Data Streams

Kinesis Streamsのデータ提供側（プロデューサー）とデータ利用側（コンシューマー）に様々なサービスが利用可能



[Q]Kinesisの特徴

IoTソリューションメーカーはセンサーを利用した交通量調査システムをAWS上に構築しています。IoTセンサーデータはAmazon Kinesis Data Streamsを利用して収集され、24時間ほどでAmazon Kinesis Data Firehoseを介してS3バケットにデータを蓄積します。しかしながら、データを検証したところ、S3バケットがKinesisストリームに送信されているすべてのデータを受信していないようです。センサーデバイスからのデータ送信には問題がないようです。

この問題の最も可能性が高い原因はどれでしょうか？

1) Amazon Kinesis Data Streamsのデータ保持期間がデフォルト設定となっている。
2) Amazon Kinesis Data Streamsからの配信設定が無効化されている。
3) Amazon Kinesis Data Firehoseにより、一部の不十分なデータを排除する処理設定を有効化している。
4) Amazon Kinesis Data Firehoseのデータ保持期間がデフォルト設定となっている。

Kinesisの特徴

Kinesisはデータ量やデータ保持期間、バッチ間隔や暗号化を調整することが可能

- バッチ間隔を60秒に設定など、バッチサイズまたはバッチ間隔を指定可能
- Kinesis data stream のデータ保持期間はデフォルト24 時間で、最大値は168時間
- 配信ストリームは自動的にスケーリングされ1 つのシャードは 1 秒あたり最大 1 MB のデータを取り込むことができ (パーティションキーを含む)、書き込みは 1 秒あたり 1,000 レコードを取りこみ可能
- 送信先にアップロードされたデータをKMS暗号化キーを指定して、自動的に暗号化を実施
- コンソールや Amazon CloudWatch からメトリクスを参照可能
- 送信データ量とデータ形式の変換に対してのみ料金が発生

[Q] Kinesis Data Firehoseの利用

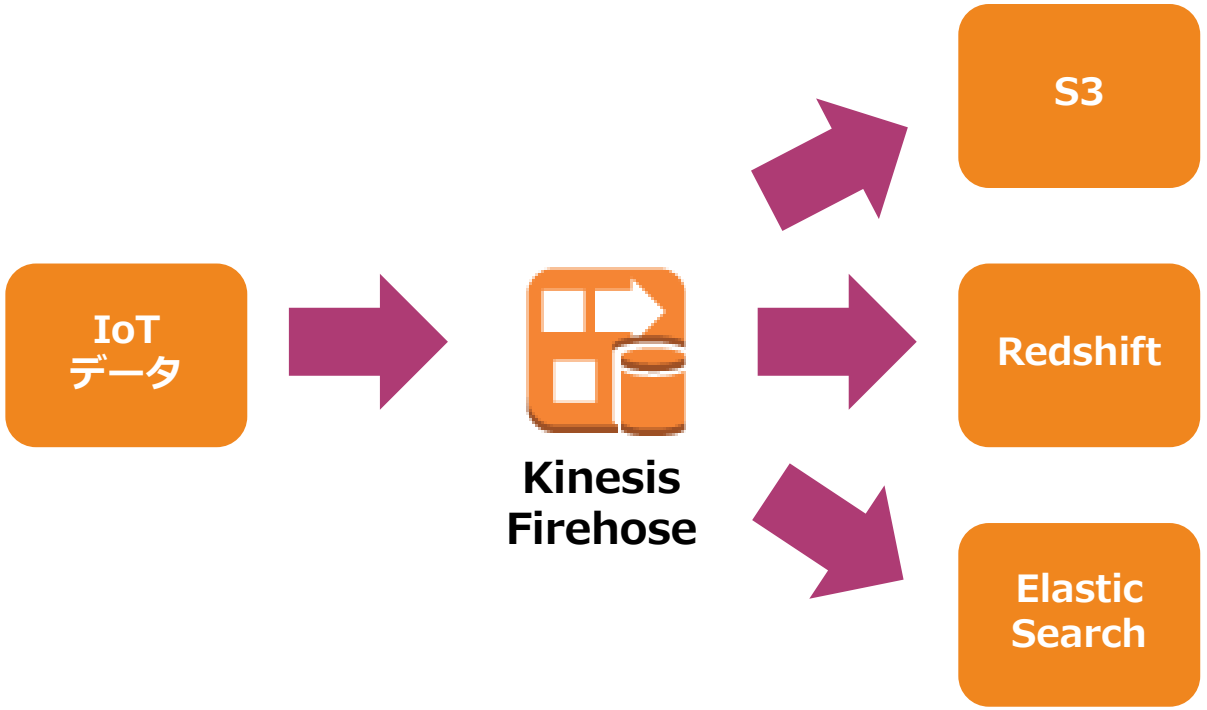
IoTソリューションメーカーはセンサーを利用した交通量調査システムをAWS上に構築しています。IoTセンサーデータを収集して、それらのデータを交通量予測モデルに利用します。データの速度は1分あたり1GBであり、予測モデルを構築するために、最も関連性の高い属性のみを含むデータに絞ってS3に保存することが必要です。

この要件を満たすために、最も費用効果の高いサービスの組合せはどれでしょうか？

1) Kinesis Data Streamsにデータを取り込み、Lambda関数を使用してデータ出力範囲を絞った上で、S3に保存する。
2) Kinesis Data Firehoseにデータを取り込み、Firehoseのフィルタリング機能でデータ出力範囲を絞った上で、S3に保存する。
3) Kinesis Data Streamsにデータを取り込み、Kinesis Data Analyticsを使用してデータ出力範囲を絞った上で、S3に保存する。
4) Kinesis Data Firehoseにデータを取り込み、Lambda関数を使用してデータ出力範囲を絞った上で、S3に保存する。

Amazon Kinesis Data Firehose

ストリームデータを各種DBに配信するためのサービス。
Lambdaと連携してETLとしても機能する



[Q] Kinesisの基本構成

IoTソリューションメーカーはセンサーを利用した交通量調査システムをAWS上に構築しています。IoTセンサーデータを収集して、交通量予測モデルに利用します。データはリアルタイムでAWSに送信されます。サービスの品質のためには、IoTデバイスごとにデータを確実に受信し、テータを処理することが不可欠です。

次の中で最も費用効果の高いサービスの組合せはどれでしょうか？

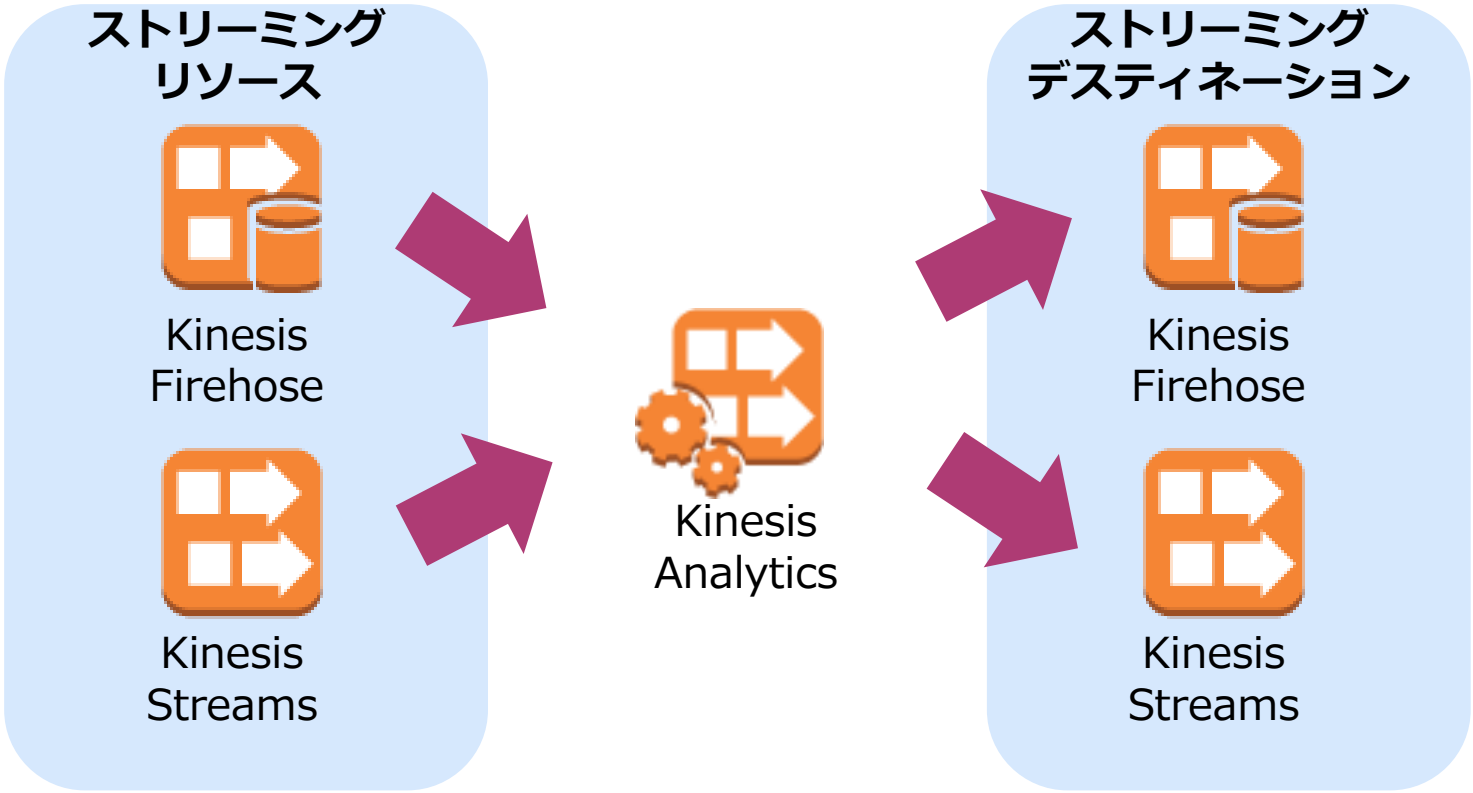
1) 各デバイスのパーティションキーを使用して、Amazon Kinesis Data Streamsでデバイス毎にデータを収集し、Amazon Kinesis Data Firehoseを使用してデータをAmazonS3に保存する
2) 各デバイスのシャードを指定して、Amazon Kinesis Data Streamsでデバイス毎にデータを収集し、Amazon Kinesis Data Firehoseを使用してデータをAmazonS3に保存する
3) 各デバイスの１つ標準キューを使用して、Amazon SQSでデバイス毎にデータを収集し、Lambda関数を使用してデータをAmazonS3に保存する
4) 各デバイスの１つFIFOキューを使用して、Amazon SQSでデバイス毎にデータを収集し、Lambda関数を使用してデータをAmazonS3に保存する

Amazon Kinesis Data Firehose

Kinesis Data Streamsによりリアルタイムにデータを収集し、
Kinesis Data Firehoseがデータを変換・格納する。

Amazon Kinesis Data Analytics

ストリームデータを標準的なSQLクエリでリアルタイムに分析



[Q] Kinesisと連携するサービス

自動車メーカーは、最新のモデルにはリアルタイムの位置データを取得して、MaaSプラットフォームを展開する方針です。 同社のソリューションアーキテクトは、Kinesis Data Firehoseを使用して、固有のストリーミングデータをダウンストリーム分析のターゲットに配信する計画をしています。

次のターゲットのうち、Kinesis Data Firehoseの配信先としてサポートされていないサービスはどれでしょうか？

1) Amazon EMR
2) Amazon RedShift
3) S3
4) Amazon Elasticsearch

Kinesisと連携するサービス

Kinesisを他のサービスと連携して、データ処理したり、データを保存する

Kinesis Data Streams	• Lambda関数と連携して、ストリームデータをポーリングして処理する。 • EC2にデータ処理を実施する。
Kinesis Data Firehorse	ストリームの配信先として以下の保存先を指定可能 • Amazon S3 • Amazon Redshift • Amazon Elasticsearch Service

[Q]アプリケーションの構築

IoTベンチャー企業は店舗解析IoTソリューションを運用しています。 店舗のセンサーなどのIoTデータはKinesis Data Streamsに送られ、Kinesis Data Firehoseで配信処理します。ソリューションアーキテクトはFirehose配信ストリームにIoTデータを送信するようにKinesis Agentを設定しましたが、データが期待どおりにFirehoseに到達していないようです。

この問題の最も妥当な根本原因はどれでしょうか？

1) Kinesis Agentは、Kinesisデータストリームに設定することが必要である。
2) Kinesis Data Firehoseの配信ストリーム処理が上限に達している。
3) Kinesis Data Streamsのシャードが不足している。
4) Kinesis Data Firehoseの配信ストリームソースがKinesis Data Streamsに設定されている

アプリケーションの構築

Kinesis Streamsは次の関連機能を活用してストリーミング処理アプリケーションを構築する

Amazon Kinesis Agent	Kinesisサービスにデータを簡単に収集して取り込むOSSのスタンドアロンJavaアプリケーション
Amazon Kinesis Producer Library (KPL)	Kinesis Streamsにデータを送信するOSSの補助ライブラリ
Fluent plugin for Amazon Kinesis	Kinesis StreamsとKinesis Firehoseにイベントを送信するOSSのFluentd出力プラグイン
Amazon Kinesis Data Generator (KDG)	Kinesis Data Generator (KDG)を利用してKinesis StreamsまたはKinesis Firehoseにテストデータを簡単に送信できる
Amazon Kinesis Client Library (KCL)	KCLを利用してKinesisアプリケーションを作成する。OSSのクライアントライブラリで、EC2インスタンスなどにデプロイして利用する **ワーカー**がシャード数に応じて、レコードプロセッサのライフサイクル管理（生成／終了）を実施

[Q]Kinesisのスケーリング

大学発の農業ベンチャーは農業IoTソリューションを運用しています。 農地に設置したセンサーデバイスからセンサーデータをリアルタイムに取得して、解析するIoTアプリケーションをKinesis Data Streamsを利用して実装しています。データストリームのプロデューサーとコンシューマー間のデータ配信速度のパフォーマンスに遅れが発生しており、あなたはソリューションアーキテクトとしては、スループット性能を向上させるよう依頼されました。

現状のパフォーマンスを向上させるために、何をするべきでしょうか？

1) Amazon Kinesis Data Streamsの拡張モニタリング機能を使用する。
2) Amazon Kinesis Data Streamsの最大配信設定を大きくする。
3) Amazon Kinesis Data Streamsのスケーリング機能を使用する。
4) Amazon Kinesis Data Streamsの拡張ファンアウト機能を使用する。

[Q]Kinesisのスケーリング

農業ベンチャーは農業IoTソリューションを運用しています。 あなたはソリューションアーキテクトとして、農地に設置したセンサーデバイスからセンサーデータをリアルタイムで取得して、解析するIoTアプリケーションをKinesis Data Streamsを利用して実装しています。複数のコンシューマーアプリケーションの合計読み取り数がシャードごとの制限を超えており、パフォーマンスが限界にきているようです。

現状のパフォーマンスを向上させるために、何をするべきでしょうか？

1) Kinesisデータストリームに利用するインスタンスを増加する。
2) シャードごとの読み取りトランザクションの数を増加する。
3) Kinesisデータストリーム内のシャードを分割する。
4) Kinesisデータストリーム内のシャードを結合する。

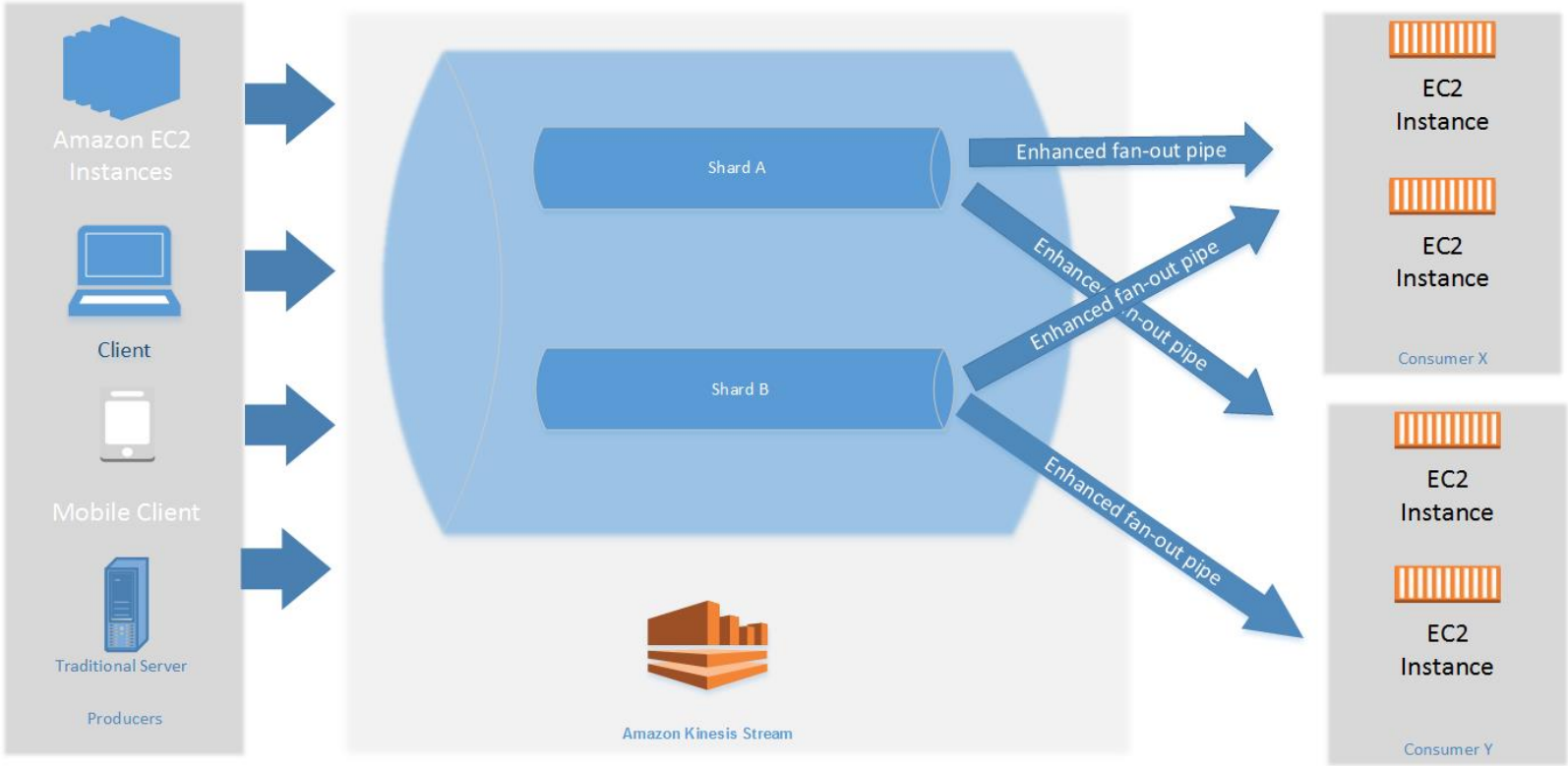
Kinesisのスケーリング

Kinesisのスケーリングでは、リシャーディングによりシャード数を増加させる。

リシャーディング	• 分割：シャード数を増加することでパフォーマンスを向上させる。 • 結合：シャード数を減少させることでコストを削減する。 • 1シャードに対して１インスタンスまで対応できる。
拡張ファンアウト機能	• スループット専用のコンシューマーの開発機能 • コンシューマーは、シャードあたり 1 秒間に最大 2 MB のデータのスループットで、ストリームからレコードを受け取ることができる。

Kinesisのスケーリング

この構成ではシャードを持つストリームから、拡張ファンアウト機能を使ったコンシューマーがシャードあたり最大 2 MB/秒のデータを処理している。

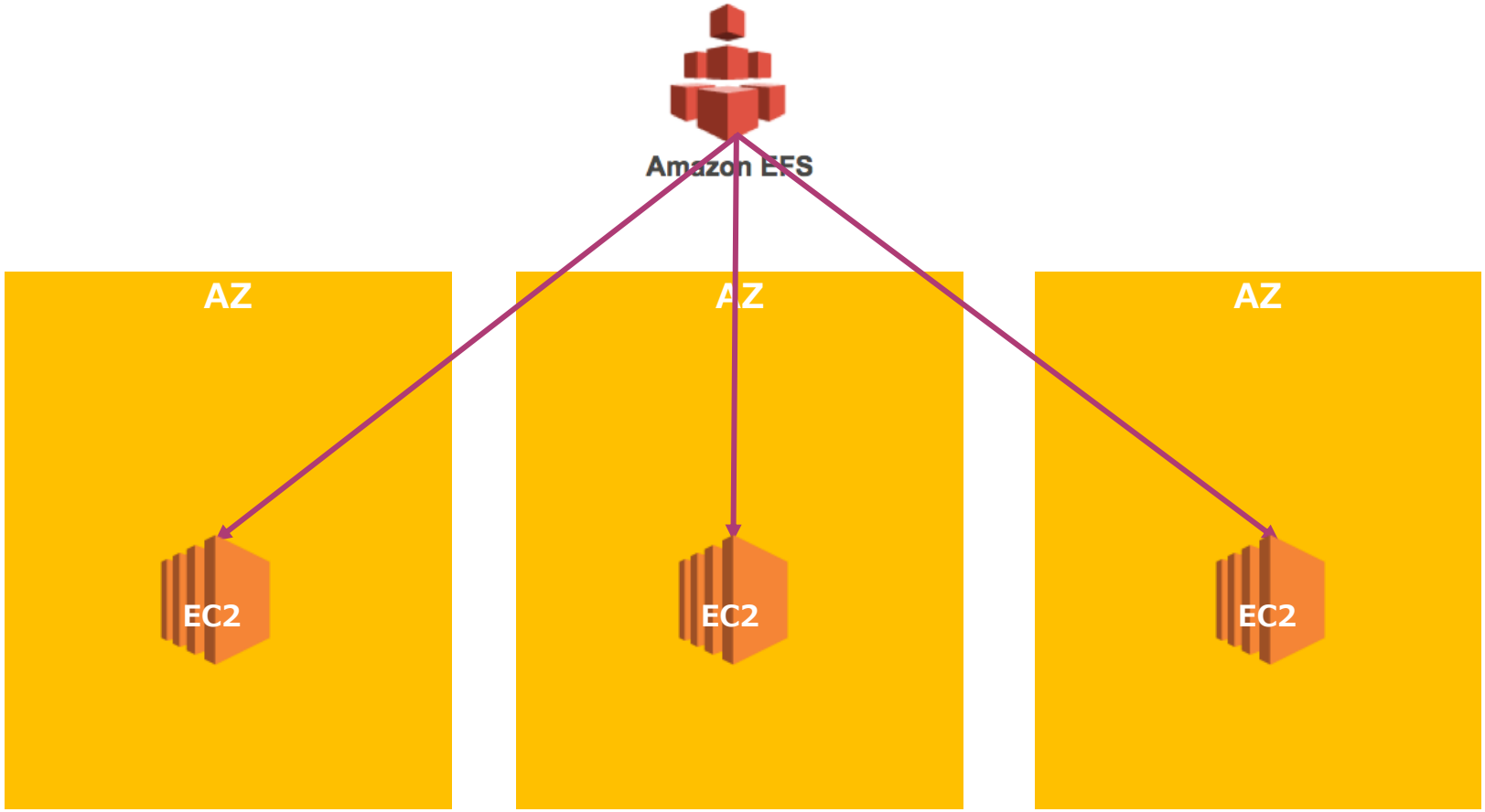


Reference: https://docs.aws.amazon.com/ja_jp/streams/latest/dev/enhanced-consumers.html

EFSの出題範囲

EFSとは何か？

複数インスタンスで共有して利用できるファイルストレージ



EFSの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

EFSの選択	✓ シナリオに基づいてストレージ要件が提示されてEFSを選択する問題が出題される。
EFSの設定	✓ EFSを利用する際の設定方法が問われる。
EFSの構成	✓ EFSを利用した複数EC2インスタンスを利用した構成方法が問われる。
EFSのパフォーマンスモード	✓ シナリオに基づいて、EFSのパフォーマンスモードの相違や設定方法が問われる。
EFS IAの利用	✓ EFSのライフサイクル管理を実施して、Infrequency Accessを利用したコスト削減の設定が出題される。

[Q]EFSの選択

あなたはソリューションアーキテクトとして、複数のLinux EC2インスタンスを利用したアプリケーションを構築しています。このアプリケーションは、POSIX準拠の共有ネットワークファイルシステムにアクセスすることが必要です。

どのストレージサービスを選択するべきでしょうか？

1) EBS
2) S3
3) Amazon FSx for Windwos
4) EFS

EFS（Elastic File System）

複数のEC2インスタンスからアクセス可能な共有ストレージ

S3	◘ オブジェクトストレージでリージョンに設置 ◘ HTTPによるAPI経由でアクセス ◘ 大容量のデータを長期保存するためのもの
EBS	◘ ブロックストレージでAZに設置 ◘ EC2インスタンスのディスクボリュームとして利用するが、物理的ではなくネットワーク経由で利用 ◘ 複数のEC2インスタンスにアタッチできない
EFS	◘ NASに似たファイルストレージ ◘ ファイルシステムとして利用し、複数のEC2インスタンスでの共有アクセスが可能 ◘ S3と異なりインターネットから直接アクセスができない

EFS（Elastic File System）

その特徴はシンプルでスケーラブルで柔軟に利用できるファイルストレージであること

シンプル	◘ フルマネージド型サービス ◘ ネットワークファイルシステムバージョン 4 (NFS v4) プロトコルを利用して、関連ツールや標準プロトコル／APIでアクセス可能（ POSIX準拠）
スケーラブル	◘ ペタバイトまでスケーラブルにデータを蓄積 ◘ スループット／IOPS性能は自動的にスケーリングし、低レイテンシーを維持
柔軟性	◘ ファイルの減少に合わせて自動で拡張・縮小 ◘ 事前に容量を設定する必要なし ◘ 使った分だけの従量課金

基本性能

何千もの同時アクセスが実現可能という性能が特徴的

基本性能	制限
◘ スループットを 100 MiB/秒（バースト込み） ◘ ファイル名は255バイト ◘ 1ファイルの最大容量48TB ◘ インスタンス当たり128ユーザーまでの同時オープンが可能 ◘ POSIX準拠 ◘ 何千もの同時アクセスが実現可能	◘ アカウント当たりのファイルシステム数：1000 ◘ AZごとのファイルシステムあたりのマウントターゲット： 1 ◘ ファイルシステムあたりのタグ：50 ◘ マウントターゲットあたりのセキュリティグループ： 5 ◘ ファイルシステムあたりのVPC数： 1 ◘ 各 VPC のマウントターゲットの数：400

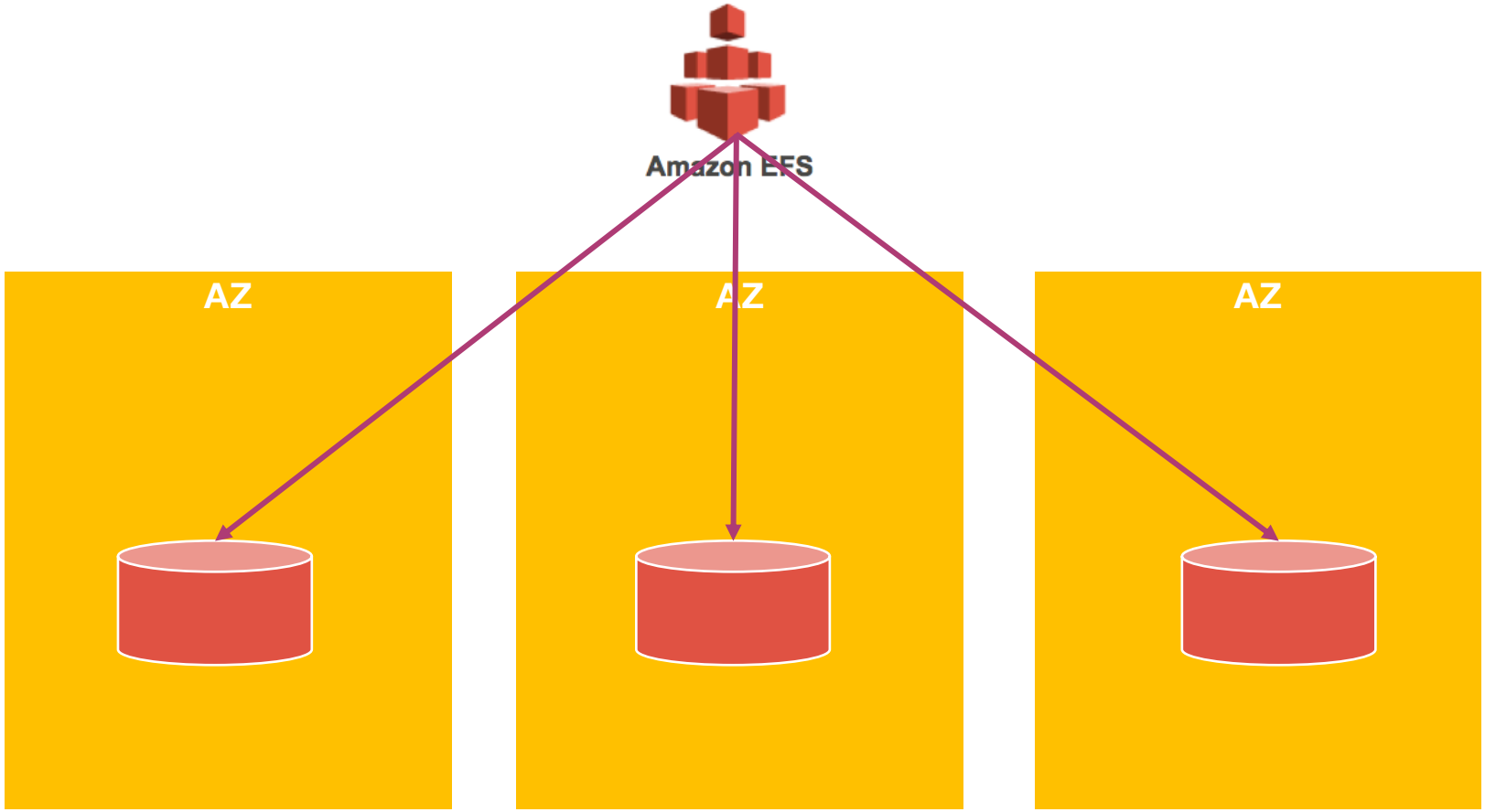
ユースケース

複数EC2インスタンスでデータ共有する際はEFSを利用する。

利用方針	利用シーン
◘ EBS（IOPS以外）では構成できない複数インスタンスからの同時アクセス構成が可能 ◘ 数秒単位でのデータ追記が必要 ◘ フルマネージドで運用して簡易に利用していきたい	◘ アプリケーションの共有ディレクトリとして利用 ◘ ビッグデータなどの分散並列処理環境における共有データアクセスストレージとして利用 ◘ コンテンツの共有リポジトリとして利用

EFSのデータ保存

EFSのデータファイルは複数AZに分散して保存されている



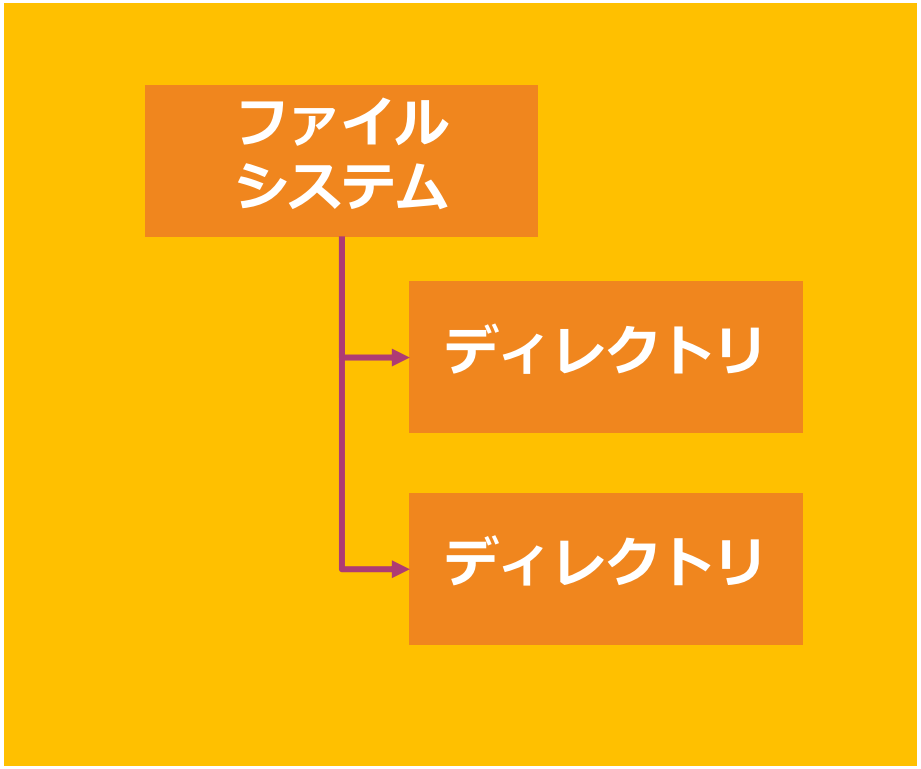
EFSの設定

EFS構築では次の設定を実施していく

- ファイルシステムを作成
- 接続先のマウントターゲットの作成
- セキュリティグループの作成
- パフォーマンスモードの選択

ファイルシステム

EFSの管理単位でファイルやディレクトリの保管場所。1つのAWSアカウントで複数のファイルシステムを作成できる



[Q] EFSの設定

大手IT企業はWEBアプリケーションをAWS上で構築しています。このアプリケーションは複数AZに展開された複数のEC2インスタンスにホストされており、Amazon EFSを使用してユーザーのホームディレクトリを構成します。 ユーザーがファイルをEFSファイルシステムに保存できるようにする設定が必要です。

この要件を満たすためにEFSをどのように設定しますか？（2つ選択してください。）

1) ユーザーごとにサブディレクトリを作成し、ユーザーに読み取り/書き込み-実行権限を付与する。 次に、サブディレクトリをユーザーのホームディレクトリにマウントする。
2) 各EC2インスタンスが展開されている各AZにマウントターゲットを構成して、EFSへのアクセスを設定する。
3) 各EC2インスタンスが展開されているリージョンにマウントターゲットを構成して、EFSへのアクセスを設定する
4) 各EC2インスタンスが展開されているVPCにマウントターゲットを構成して、EFSへのアクセスを設定する。
5) ユーザーごとに個別のEFSファイルシステムを作成し、ルートディレクトリに対する読み取り/書き込み-実行権限をそれぞれのユーザーに付与する。次に、ファイルシステムをユーザーのホームディレクトリにマウントする。

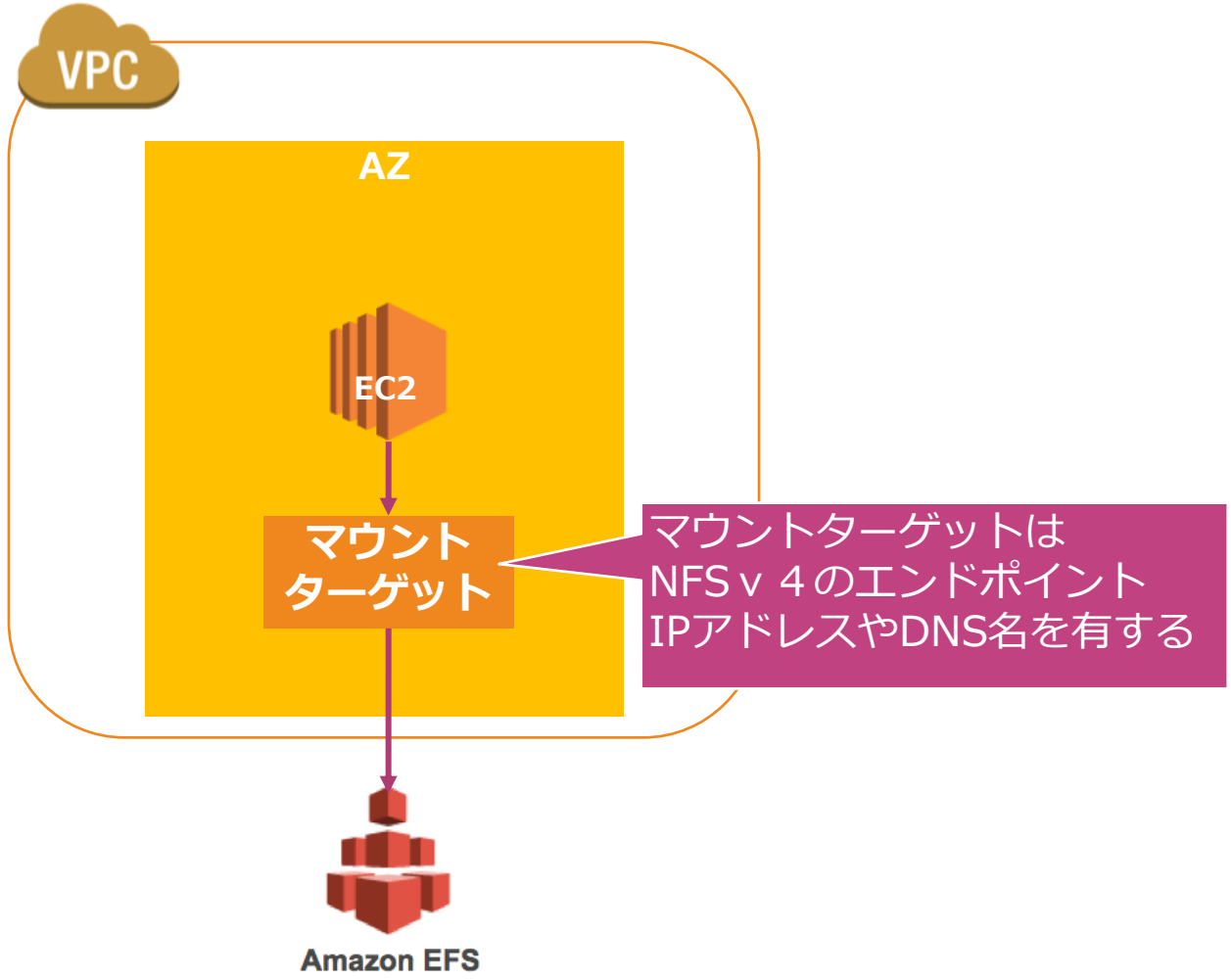
マウントターゲット

EC2インスタンスからの接続先のマウントターゲットを設定

- VPC内のAZにある接続先
- EC2インスタンスは同じAZ内にあるマウントターゲットから接続する
- 固定のDNS名とIPアドレスを有している
- ファイルシステムDNS名を使用してマウントすることで自動でIPアドレスを付与

マウントターゲット

EC2インスタンスからマウントターゲットを介して、AZ外にあるEFSにアクセスできる



[Q] EFSの構成

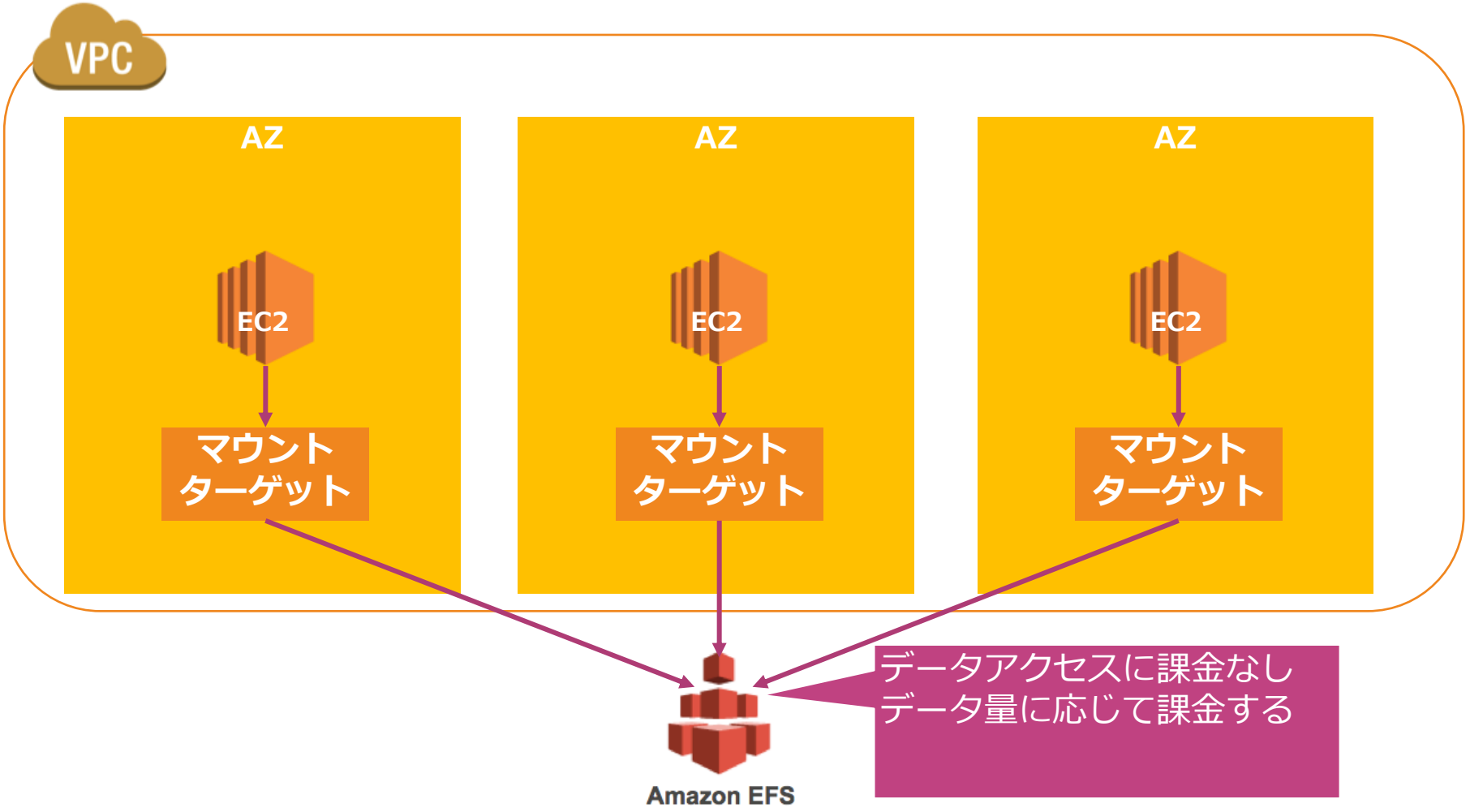
大手IT企業はWEBアプリケーションをAWS上で構築しています。複数のEC2インスタンスを利用してデータを共有することが必要です。このアプリケーションは障害が発生した場合に備えて、アプリケーションの復元力を高めることが必要です。

これらの要件を満たすために、どのようにソリューションを構成しますか？

1) EC2インスタンスに対してELBのターゲットグループを設定して複数AZにまたがるAuto Scalingグループを構成する。 EFSにデータを保存し、各インスタンスにターゲットをマウントする。
2) EC2インスタンスに対してELBのターゲットグループを構成する。 EFSにデータを保存し、各インスタンスにターゲットをマウントする。
3) EC2インスタンスに対してELBのターゲットグループを設定して複数AZにまたがるAuto Scalingグループを構成する。 EBSにデータを保存し、各インスタンスにターゲットをマウントする。
4) EC2インスタンスに対してELBのターゲットグループを構成する。 EBSにデータを保存し、各インスタンスにターゲットをマウントする。

EFSの構成

複数AZにある複数ECインスタンスからEFSにアクセスできる



[Q]EFSのパフォーマンスモード

データ分析企業はAWSを利用してビッグデータ解析ワークロードを実装しています。複数のAZにわたる数千のEC2インスタンスフリートを利用して大量のデータ処理が必要となります。 データは、すべてのEC2インスタンスから同時にマウントおよびアクセスできる共有ストレージレイヤーを利用します。

スループット性能を最大化したい場合のストレージを選択してください。

1) EBS プロビジョンドIOPS
2) Amazon S3
3) 最大I / OモードのAmazon EFS
4) 汎用モードのAmazon EFS

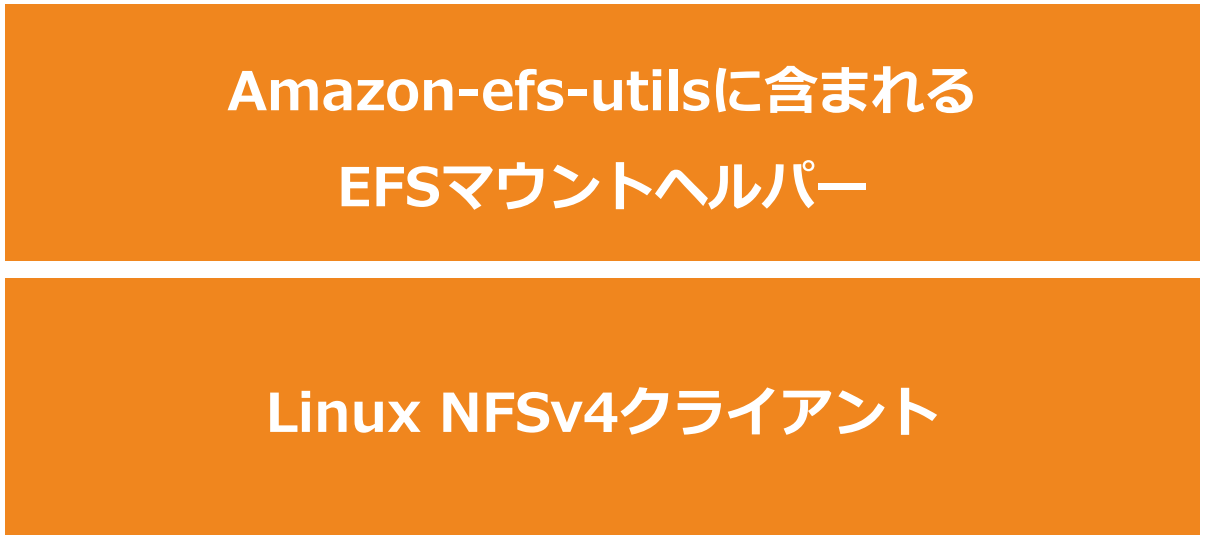
EFSのパフォーマンスモード

汎用モードと最大I/Oモードから選択。基本は汎用モード

汎用モード	◘ 一般的な用途を想定したモード ◘ デフォルトでは汎用モードとなり、推奨されている ◘ レイテンシーが最も低い ◘ 1秒あたりのファイルシステム操作を7000に制限
最大I/Oモード	◘ 何十～何千というクライアントからの同時アクセスが必要な大規模構築に利用 ◘ 合計スループットを優先してスケールする ◘ レイテンシーが多少長くなる

EFSクライアント

EFSをEC2インスタンスから操作する際に専用のクライアントソフトウェアを利用する



プロビジョンドスループット

ユースケースに応じてプロビジョンドスループットも利用

バーストのスループット	プロビジョンド スループット
ピーク時にクレジットを消費してバーストを実行して一時的な性能を向上させる方式 最大スループットとバースト時間に制限がある スループット性能向上にはストレージ容量の増大が必要	一貫したスループットを事前に設定する方式 API／AWS CLI／マネジメントコンソールにより制御 1日に1回だけスループット性能を減少できる

[Q] EFS IAの利用

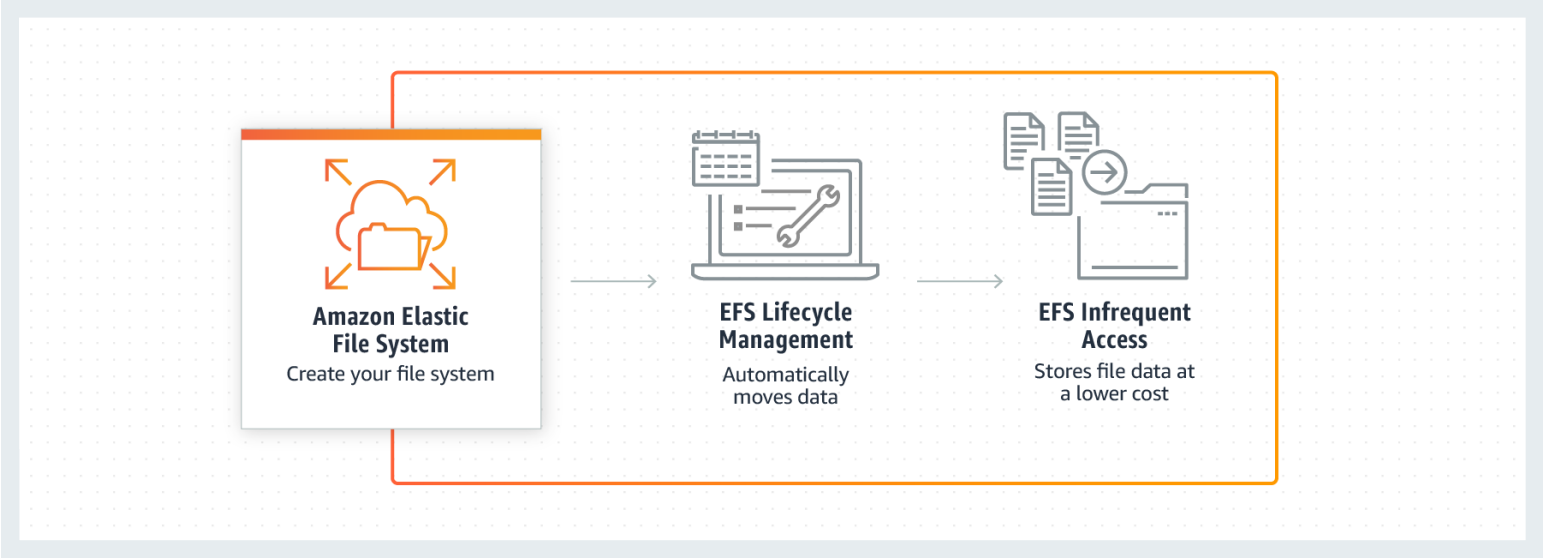
大手IT企業はWEBアプリケーションをAWS上で構築しています。このアプリケーションは複数AZに展開された複数のEC2インスタンスがデータを共有ストレージに保存します。このデータは内部的なディレクトリ管理に利用されるファイルであり、EC2インスタンスからのみ制御します。ファイルは最初は頻繁に利用されますが、その後はアクセス頻度が低下することが見込まれています。

最も費用効果の高いソリューションは何ですか？

1) EFSのライフサイクル管理を利用する。
2) EFSのストレージ最適化を利用する。
3) S3のライフサイクル管理を利用する。
4) EBSのライフサイクル管理を利用する。

EFS IAの利用

アクセス頻度が低いデータをIAストレージに格納することでコストを削減可能



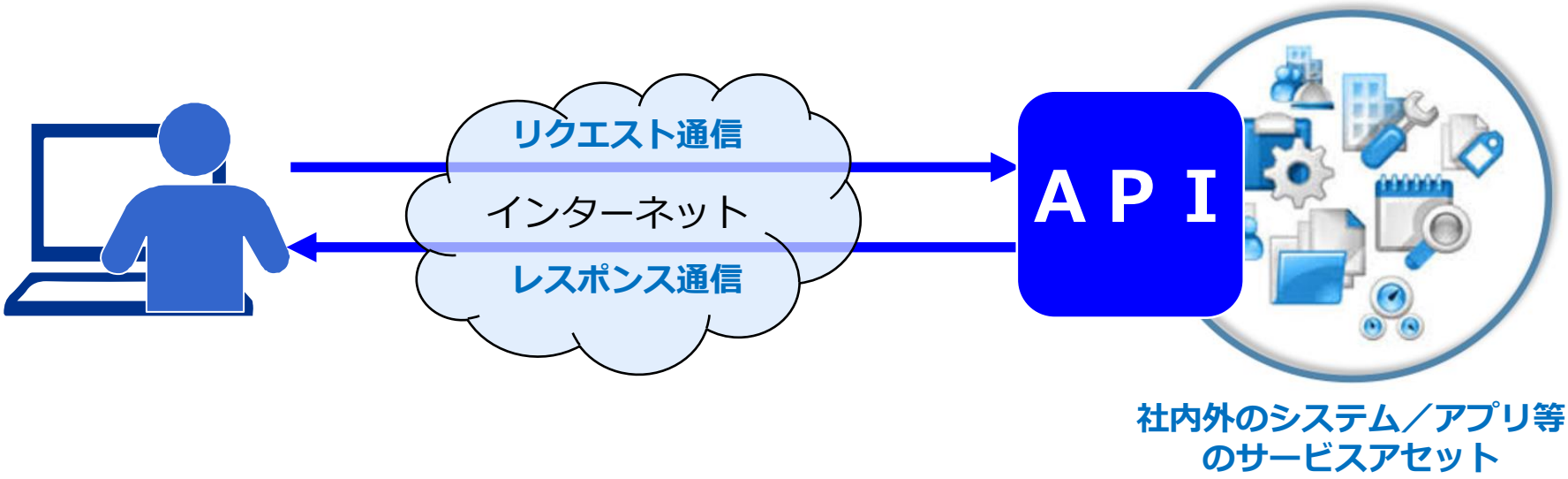
- ✓ ニーズに一致するライフサイクルポリシーを選択して、ファイルシステムのEFSライフサイクル管理を有効にする
- ✓ Amazon EFS Infrequent Access（EFS IA）は、毎日アクセスされないファイルに対してコスト最適化された価格/パフォーマンスを提供するストレージクラス
- ✓ 価格は最大92%割引：0.025ドル（GB月単位）

Reference: https://aws.amazon.com/jp/efs/features/infrequent-access/

API Gatewayの出題範囲

API Gatewayとは何か？

APIを通してリクエストとレスポンスにより、他サービスの機能やデータを呼び出すことができる



API Gatewayの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

API Gatewayの選択	✓ シナリオに基づいて、要件を達成するためにAPI Gatewayを選択する質問が問われる。
API Gatewayの特徴	✓ API Gatewayが使用するAPIタイプなどの特徴をこたえる問題が出題される。
API Gatewayの料金	✓ API Gatewayに料金の発生要因が問われる。
API Gatewayの構成	✓ API Gatewayを利用したアプリケーション構成のソリューションが問われる。
API Gatewayの認証方式	✓ API Gatewayへのアクセス許可設定などの認証方式が問われる。

API Gatewayの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

キャッシュ機能の利用	✓ シナリオに基づいて、API Gatewayのパフォーマンス向上に利用されるキャッシュ機能やTTLなどの設定方法が問われる。
スロットリングの利用	✓ シナリオに基づいて、スロットリングを利用したパフォーマンス調整の方法を選択する問題が出題される。

[Q]API Gatewayの選択

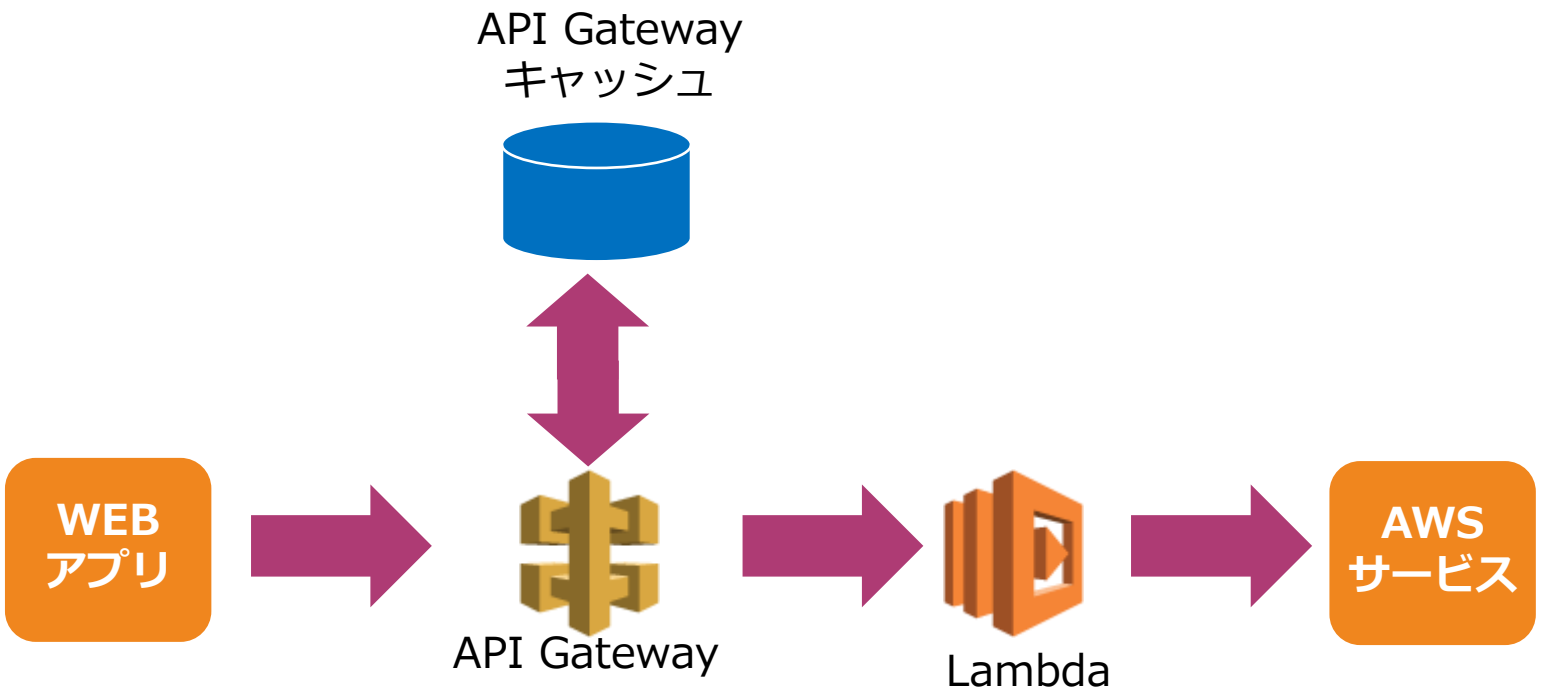
あなたはソリューションアーキテクトとして、モバイルアプリケーションをAWS上に構築しています。モバイルアプリケーションはユーザーインターフェイスにデータを入力するために、いくつかのアプリケーションサービスからデータを取得して利用します。 実装では、クライアントインターフェイスと連携アプリケーションとを分離したアーキテクチャ構成が必要となります。

この要件を満たすことができるAWSサービスはどれでしょうか？

1) AWS Lambda
2) AWS Device Farm
3) API Gateway
4) AWS Transit Gateway

ユースケース

API Gatewayを連携口として外部アプリとの連携を実現する



[Q] API Gatewayの特徴

AWSを利用した新規WEBアプリケーションはマイクロサービスアーキテクチャに基づいて開発されています。あなたはソリューションアーキテクトとして、様々な機能を有したアプリケーションを柔軟に連携させるため、API Gatewayを使うことを決定しました。

マイクロサービスを構築する際にAPI Gatewayを利用するべき理由を選択してください。（2つ選択してください。）

1) RESTfulAPIを利用できる。
2) スパイラルAPIが利用できる。
3) ブループリントに実装パターンが提供されている。
4) API Gatewayのプロビジョニング数に応じて課金される。
5) APIコールと転送したデータ量に応じて課金される。

API Gateway

API Gatewayは以下のタイプのAPI作成・管理をフルマネージド型サービスで提供

APIの作成管理	• RESTful APIの作成、デプロイ・管理 • AWS Lambda 関数、その他のAWSサービスを公開するためのWebSocket API の作成・デプロイ・管理 • フロントエンド HTTP および WebSocket エンドポイントによって公開された API メソッドの呼び出しを実行する
基本性能	• 最大数十万個のAPI同時呼び出し・受付が可能 • DDoS攻撃対応やスロットリングによるバックエンド保護 • EC2／Lambda／任意のウェブアプリケーションのワークロード処理を実行する際に利用 • Lambdaと密接に統合されている

API Gatewayの料金

APIのタイプに応じて料金形式が異なる。

HTTP API	使用した API コールの分だけ料金が発生
Restful API	受信した API コールと、転送データ量に対してのみ料金が発生
WebSocket API	受送信したメッセージ数および分単位の接続合計時間から料金が発生

[Q] API Gatewayの構成

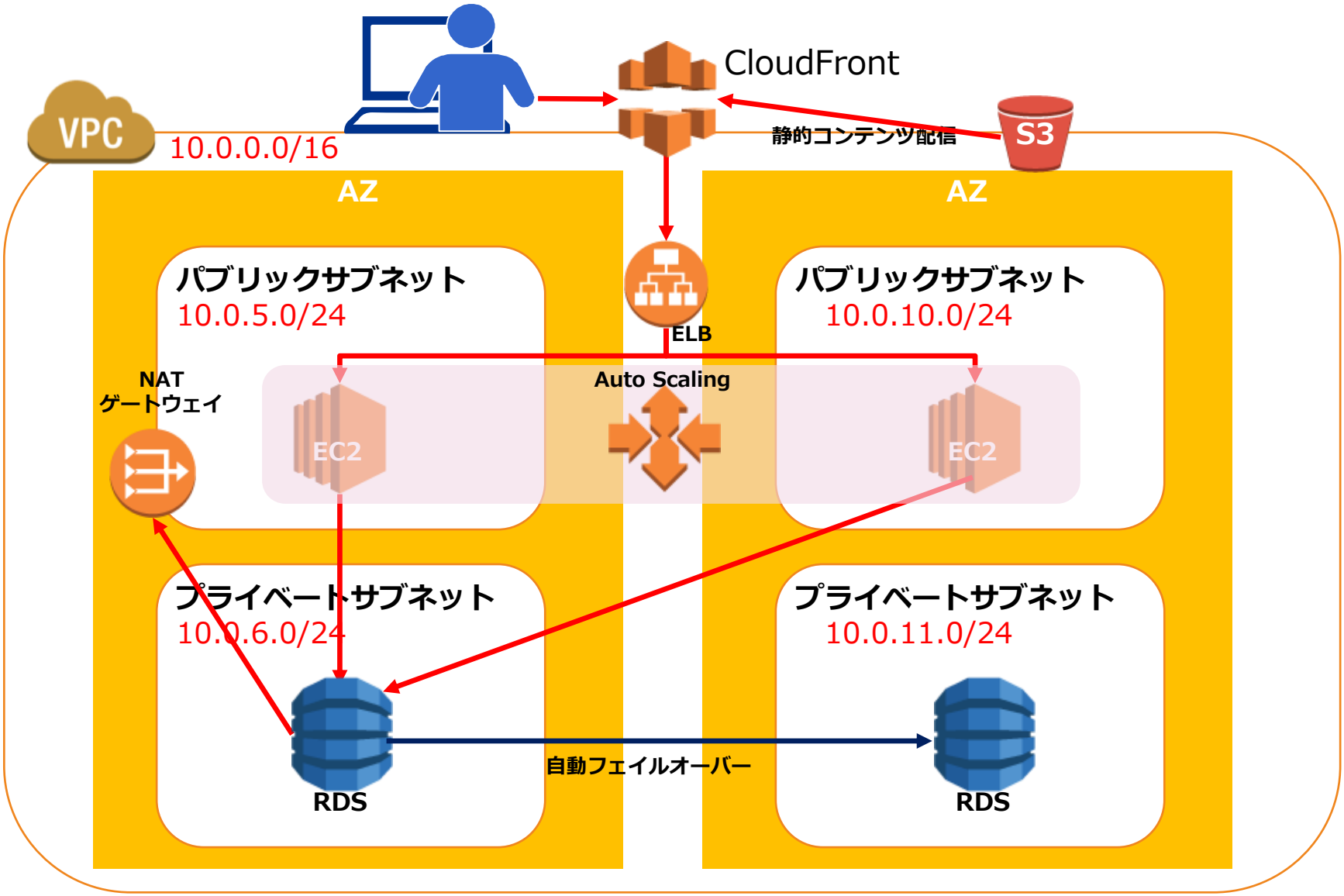
A社はマイクロサービスアーキテクチャに基づいてWEBアプリケーションを開発しています。このアプリケーションでは、ユーザーがAPIを呼び出すことで簡単なデータ処理プロセスを実行することができます。非機能要件として毎日約1,000のリクエストを受信し、平均応答時間は50ミリ秒が必要となります。

最小コストで高可用なアーキテクチャとなる構成はどれでしょうか？

1) API GatewayでAPIを作成してマイクロサービス間を連携し、サービスバックエンド処理にLambdaを使用する。
2) EC2インスタンスによりWEBサイトを構築して、SQSを介して、バックエンド処理用のEC2インスタンスと連携する。
3) 最大2つのインスタンスでAuto scaling グループを設定して、アプリケーションロードバランサーを使用してトラフィック分散する。
4) API GatewayによるAPIを作成してマイクロサービスの連携に利用し、サービスバックエンド処理にEC2インスタンスを使用する。

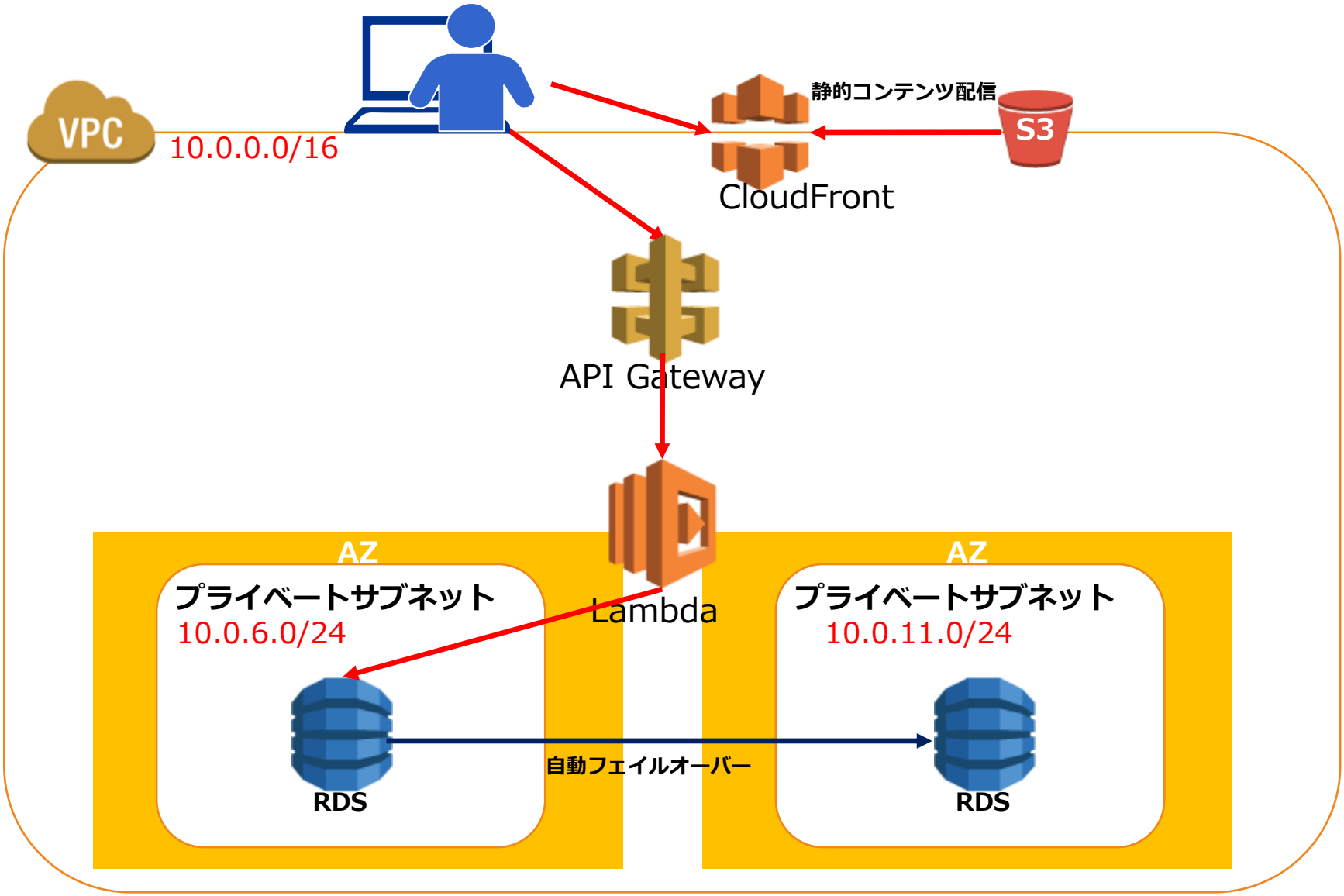
API Gatewayの構成

EC2中心のWEBアプリケーションの構築をサーバレス化する



API Gatewayの構成

EC2中心のWEBアプリケーションの構築をサーバレス化する



[Q]API Gatewayの認証方式

あなたの会社では様々なアプリケーションを利用しています。そのため、APIゲートウェイを導入してアプリケーション間連携を構築することになりました。このAPI gatewayを利用する開発者やIT管理者が複数存在しており、API Gatewayに対して最適な権限をユーザー毎に設定する必要があります。

API gatewayの権限管理を実施するために最適な設定方法を選択してください。

1) 認証キーを利用して、異なるユーザーにAPI Gatewayに対するアクセス権限許可を実施する。
2) IAMポリシーを利用して、異なるユーザーにAPI Gatewayに対するアクセス権限許可を設定する。
3) APIキーを利用して、異なるユーザーにAPI Gatewayに対するアクセス権限許可を実施する。
4) アクセスキーを利用して、異なるユーザーにAPI Gatewayに対するアクセス権限許可を実施する。

API Gatewayの認証方式

API Gatewayへのアクセス認証には様々なタイプを利用可能

リソースポリシー（REST APIのみ）	JSON形式のリソースポリシーを定義することで、API Gatewayのリソースからのアクションの許可または拒否を設定する。
IAM認証	APIのアクセス権限を設定したIAMポリシーを作成し、IAMユーザやIAMロールに付与してAPIへのアクセスを制御する。 APIメソッドでIAM認証を有効化する
Lambda オーソライザー	Lambda関数を作成することで、認証プロバイダーでの認証結果を元に、APIへのアクセス制御をメソッド単位で実施
Cognito オーソライザー	認証プロバイダとしてCognitoユーザープールを用いて、APIへのアクセス制御をメソッド単位で実施

[Q]キャッシュ機能の利用

あなたの会社ではマイクロサービス化されたアプリケーションを構築しています。あなたはソリューションアーキテクトとして、データ処理機能を連携するために、API Gateway、AWS Lambda、Auroraデータベースサービスを活用する新しいRestful APIを開発しました。このAPIは読み取りが非常に多いですが、 データが変更されることは滅多にありません。

API処理のパフォーマンスを向上させながらコストを削減するにはどうすればよいでしょうか？

1) Auroraにリードレプリカを追加する。
2) API Gatewayキャッシングを有効にする
3) API Gatewayの読み込みリクエストの上限緩和申請を実施する。
4) AuroraデータベースをAuroraサーバレスの切り替える。

[Q]キャッシュ機能の利用

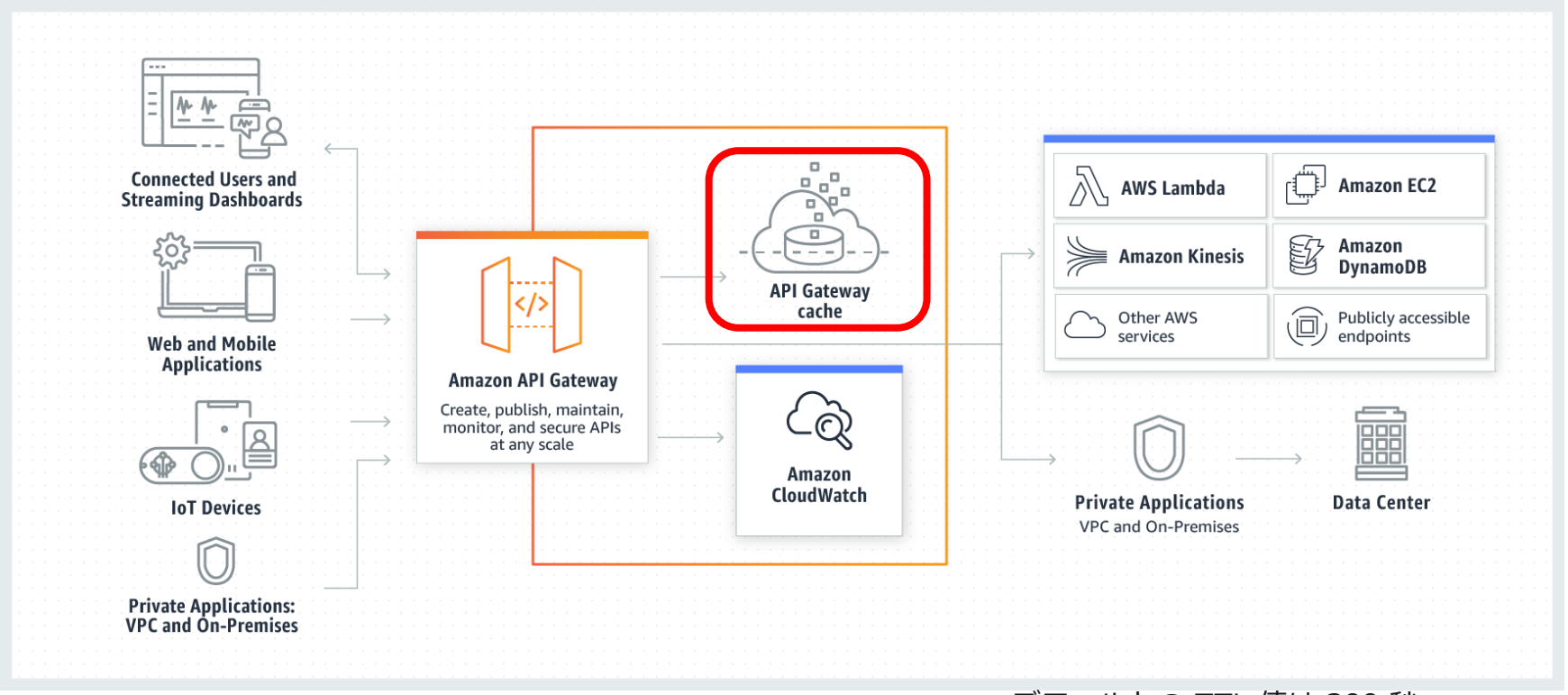
あなたの会社ではマイクロサービス化されたアプリケーションを構築しています。あなたはソリューションアーキテクトとして、データ処理機能を連携するために、API Gateway、AWS Lambda、Auroraデータベースサービスを活用する新しいRestful APIを開発しました。 キャッシュを制御して、パフォーマンスを向上させ、バックエンドサービスの負荷を軽減することが必要です。

キャッシュの制御に利用する機能はどれでしょうか？

1) スロットル機能を構成する
2) バーストを有効にする
3) 存続可能時間（TTL）設定の使用する。
4) キャッシュリクエスト機能を有効化する。

キャッシュ機能の利用

キャッシュを有効にすると、エンドポイントへの呼び出しの数を減らし、API へのリクエストのレイテンシーを短くできる



デフォルトの TTL 値は 300 秒
最大 TTL 値は 3600 秒
TTL=0 は、キャッシュを無効化

Reference: https://aws.amazon.com/jp/api-gateway/

[Q]スロットリングの利用

あなたの会社ではマイクロサービス化されたアプリケーションを構築しています。あなたはソリューションアーキテクトとして、データ処理機能を連携するために、API Gateway、 Amazon API Gatewayを使用してAPIをデプロイしました。 運用を開始したところ、ある特定の顧客からのリクエストが過剰となっており、API処理負荷が高まっているようです。

この問題に対して最適なソリューションを選択してください。

1) クライアントごとのスロットリング制限を構成する。
2) サーバー側のキャッシュ制限を構成する
3) メソッドごとのキャッシュ制限を構成する
4) サーバー側のTTLを制限する。

スロットリングの利用

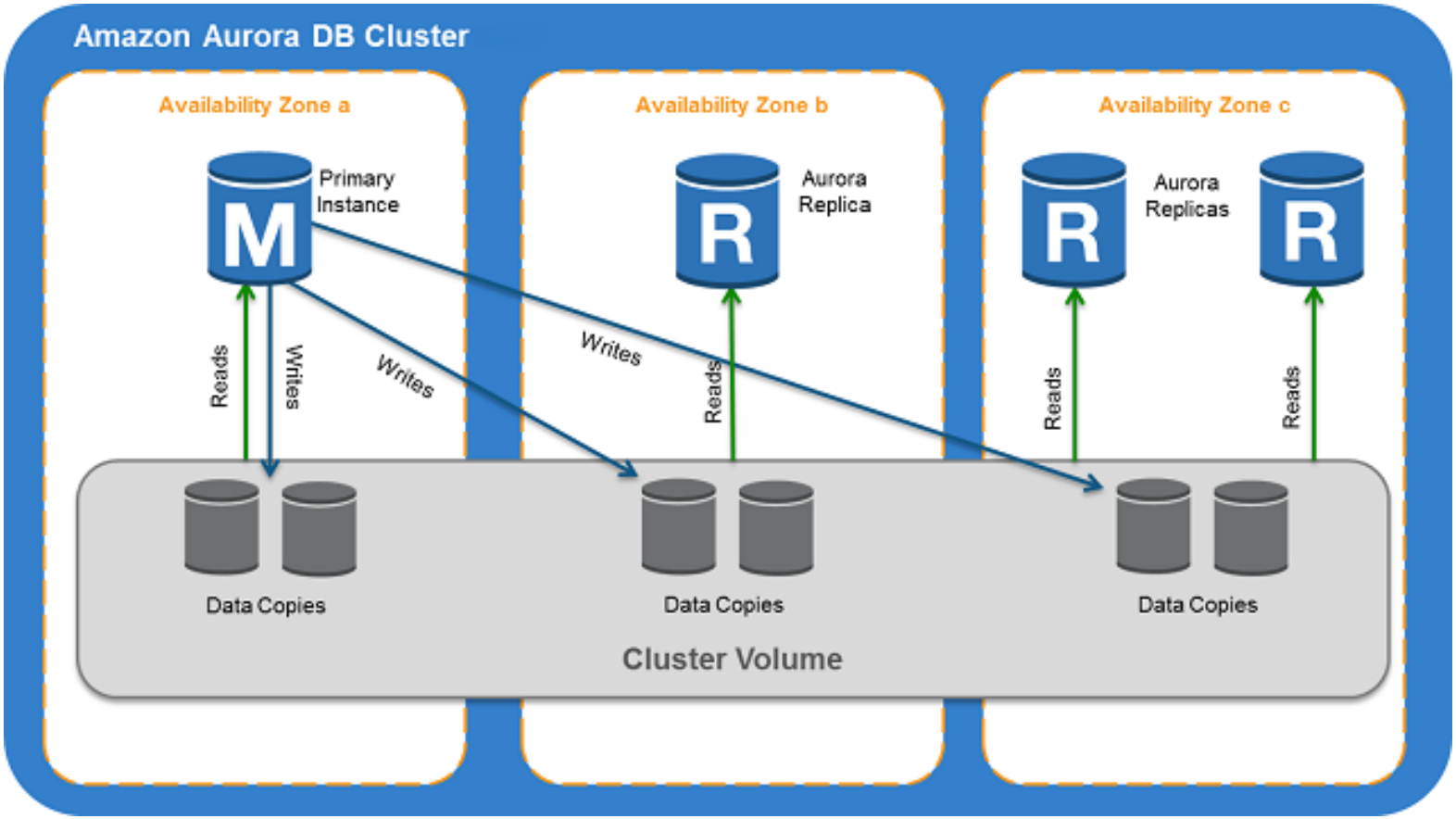
リクエスト数が多すぎる場合、制限をかけることで、トラフィックの急増に対してバックエンドサービスを守る。

サーバー側のスロットリング制限	全てのクライアントに対するリクエストを制限する。全体のリクエストが多すぎるために バックエンドサービスが処理しきれなくなることを防ぐことができる。
クライアントあたりのスロットリング制限	クライアントごとに「使用量プラン」に応じて制限を行う。 特定のユーザーからのリクエストが多い場合に有効

Auroraの出題範囲

Auroraとは何か？

マルチAZで分散されたクラスター構成により、高速・高性能なリレーショナルデータベース



Reference: https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.html

Auroraの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Auroraの特徴	✓ Auroraの特徴からAuroraを選択する問題が出題される。 ✓ Auroraを選択するべきメリットなどの特徴に関する問題が出題される。
リードレプリカ	✓ シナリオに基づいて、Auroraリードレプリカを設定する問題が出題される。 ✓ AuroraのリードレプリカとRDSのリードレプリカとの違いが問われる。
フェールオーバー構成	✓ Auroraのフェールオーバー構成のティア設定などの方法が出題される。
Auroraサーバレス	✓ シナリオに基づいて、要件を満たすためにAuroraサーバレスを選択する問題が出題される。
グローバル構成	✓ Auroraをグローバルにリードレプリカを展開できるアーキテクチャ構成が問われる。

Auroraの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

エンドポイントの選択	✓ Auroraの様々なインスタンスタイプに接続を実施するためのエンドポイントの選択が出題される。

[Q] Auroraの特徴

あなたはソリューションアーキテクトとして、データベースシステムをAWS上で構築しています。既存のオンプレミスデータベースではMySQL5.6を利用して、業務システム向けの顧客データを管理しています。最近になってデータ処理量が多くなり、高性能なデータベースをAWSに構築することになりました。あなたはAmazon Auroraが最適ではないかと検討しています。

次のうちAuroraを選択する際の利点として間違っている内容はどれでしょうか。

1) PostgreSQL10.4と互換性がある
2) MySQL5.7と互換性がある
3) 可用性は99.99%である
4) 最大5つのリードレプリカを利用した高速読み込みが可能

Aurora

クラウド時代の新しい分散型のリレーショナルデータベースとして誕生

- AmazonがクラウドDB時代に適したリレーショナルDBをーから考えて構築された新RDB
- その特徴はNoSQL型の分散高速処理とRDBとしてのデータ操作性を両立させたこと

Aurora

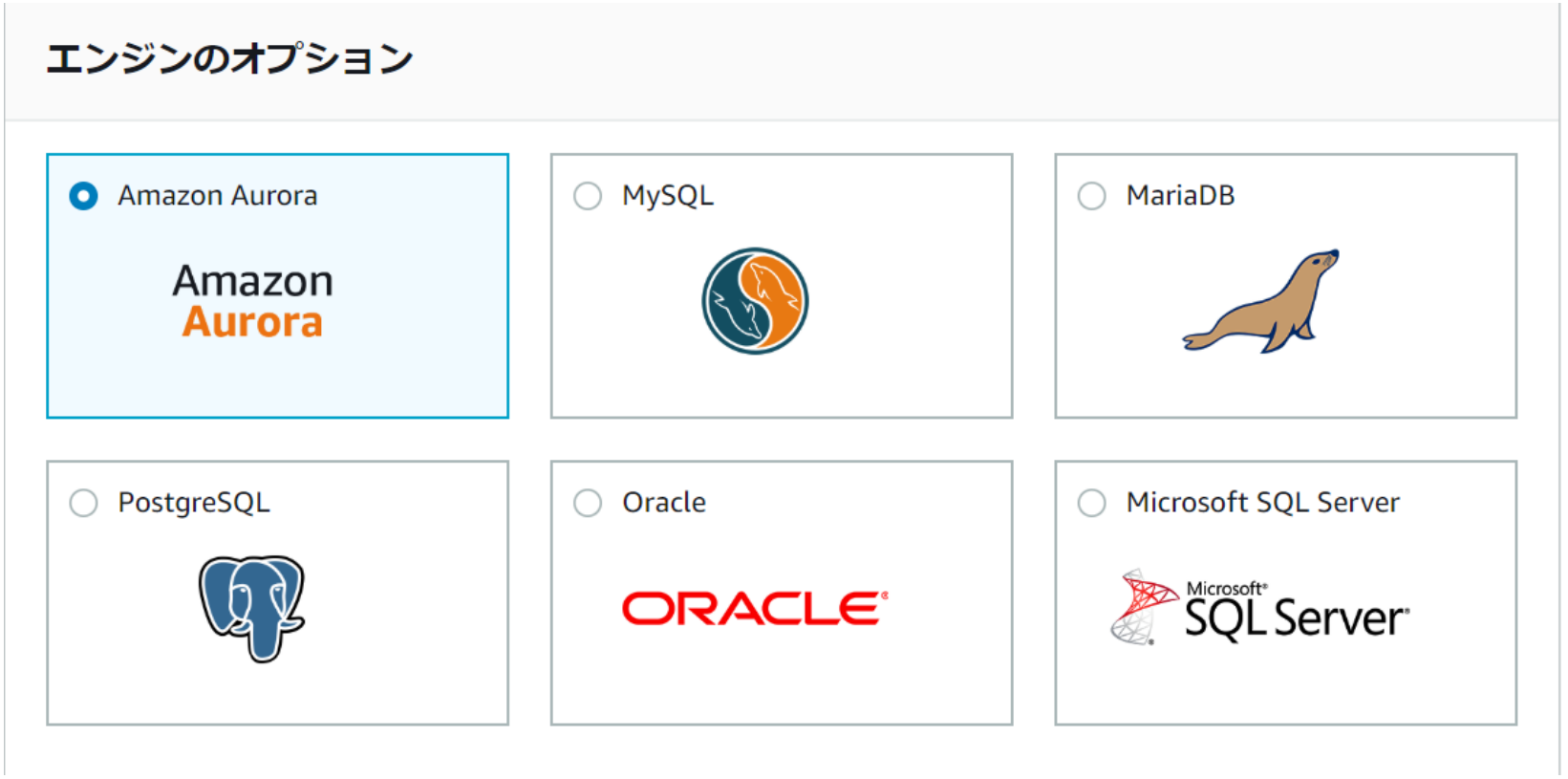
MySQLと2.5～5倍の性能を商用データベースの10分の1の価格で提供する高性能・低コストDB

性能5倍
Sysbench4インスタンスと r3.8xlargeAuroraとの比較

性能約2.5～5倍
TPC-CをR3.8xlargeのAuroraとの比較

Aurora

RDSにおいてデータベースソフトウェアの一つとして選択



Auroraの特徴

高い並列処理性能によって大量の読み書きをするのに適したDB

- 高い並列処理によるストレージアクセスによってクエリを高速処理することが可能
- Auroraは大量の書き込みや読み込みを同時に扱うことができる
- データベースの集約やスループット向上が見込まれる
- ただし、すべてが5倍高速というわけではなく、適用すべき領域を見つけて利用する

Auroraの特徴

MySQL／PostgreSQLと互換性があり、同じ操作方法とそのコミュニティを利用することが可能

MySQL5.6互換
を選択可能

PostgreSQL互換
を選択可能

Auroraの特徴

分散型で耐障害性と自己回復性を備えたスケーラブルな新しいタイプのフルマネージド型RDB

耐障害性／自己回復性

- 3つのAZに2つのコピーを設置可能で合計6つのコピーを保持
- 過去のデータがそのままS3に継続的バックアップ
- リストアも差分適用がなく高速
- どのタイミングでも安定したリストア時間を実現
- 99.99%の高可用性・高耐久性

スケーラビリティ

- 10GBから最大64TBを提供するSSDデータプレーンを利用してシームレスに拡張可能
- Auto-Scalingなどのクラウド独自のスケーラブルが可能
- 最大15のリードレプリカを利用した高速読み込みが可能

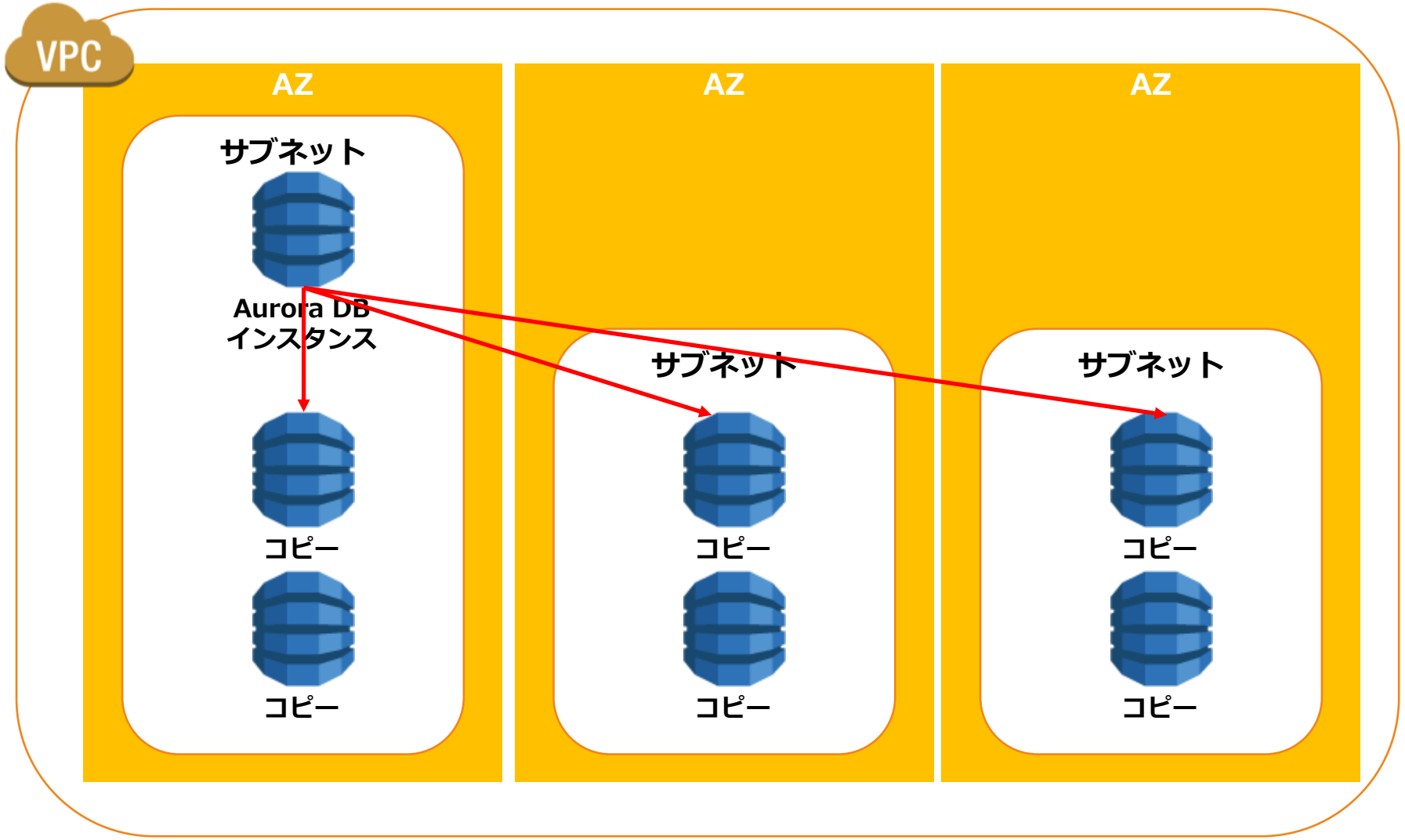
Auroraのユースケース

大規模なクエリ処理が発生するRDB環境などはAuroraへの移行を検討すべし

大規模なクエリ データ処理	◘ 書込み量が多くでトランザクション量が多い ◘ クエリ並行度が高い、データサイズが大きいケースで効果を発揮する ◘ コネクション数やテーブル数が多いデータベース処理
運用の容易さ を活用する	◘ スケーラビリティの高さやデータ容量が無制限に拡張できる ◘ レプリケーションなどの性能の高さ

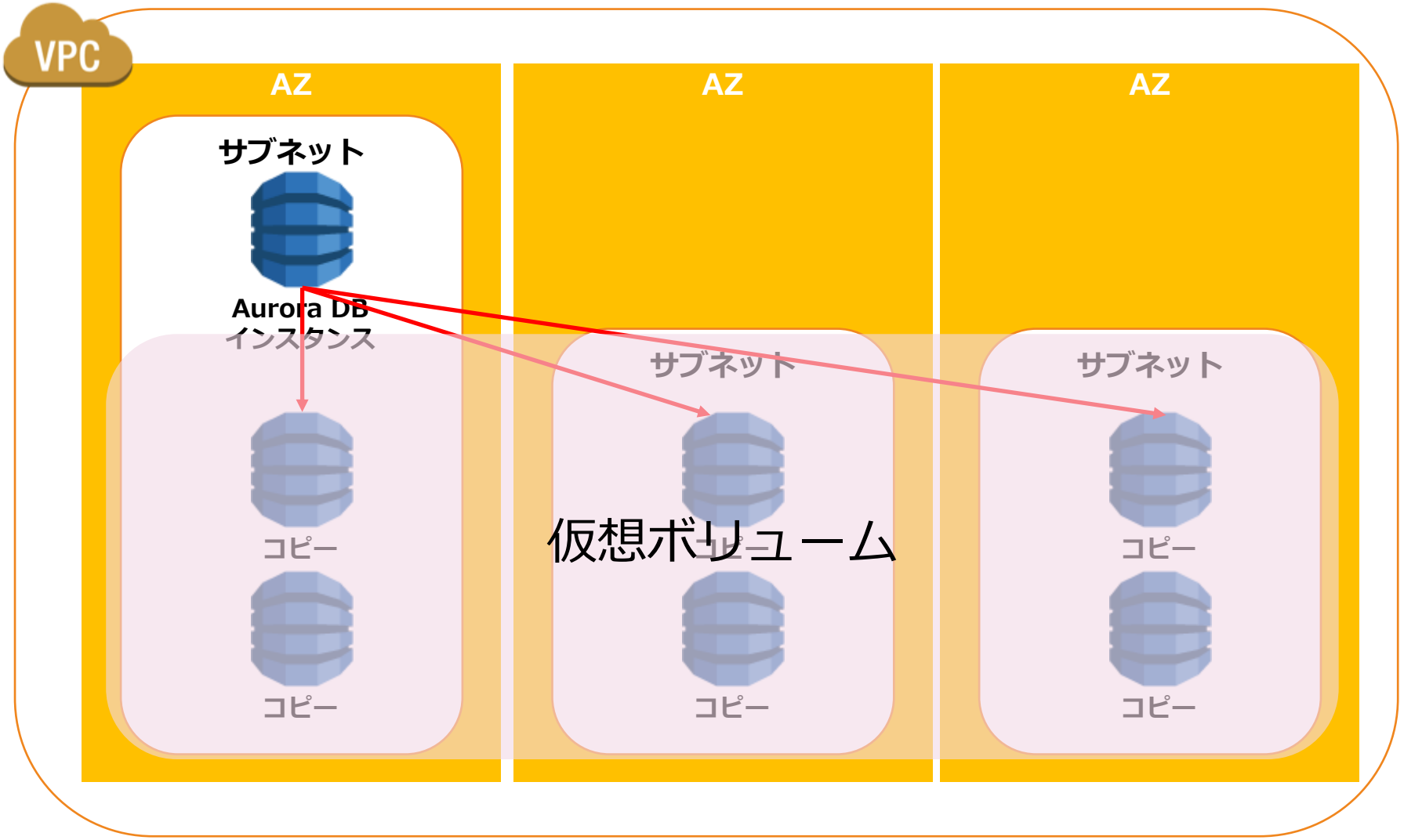
DBクラスタの仮想ボリューム

Auroraは1つのDBインスタンスと1つのDBクラスタボリュームで構成される。3つのAZにコピーされたクラスタを単一と認識



DBクラスタの仮想ボリューム

Auroraは1つのDBインスタンスと1つのDBクラスタボリュームで構成される。3つのAZにコピーされたクラスタを単一と認識



[Q]リードレプリカ

大手ニュースメディアはAWS上にニュース配信用のWebアプリケーションを運用しています。 このアプリケーションは、ALBの背後にあるAuto Scalingグループ内のAmazonEC2インスタンスフリートで実行されてます。データベースにはAuroraデータベースを利用しています。このアプリケーションの読み込みリクエストが次第に増大傾向にあるため、パフォーマンスが低下しつつあります。

この要件を満たすことができるソリューションを選択してください。（2つ選択してください。）

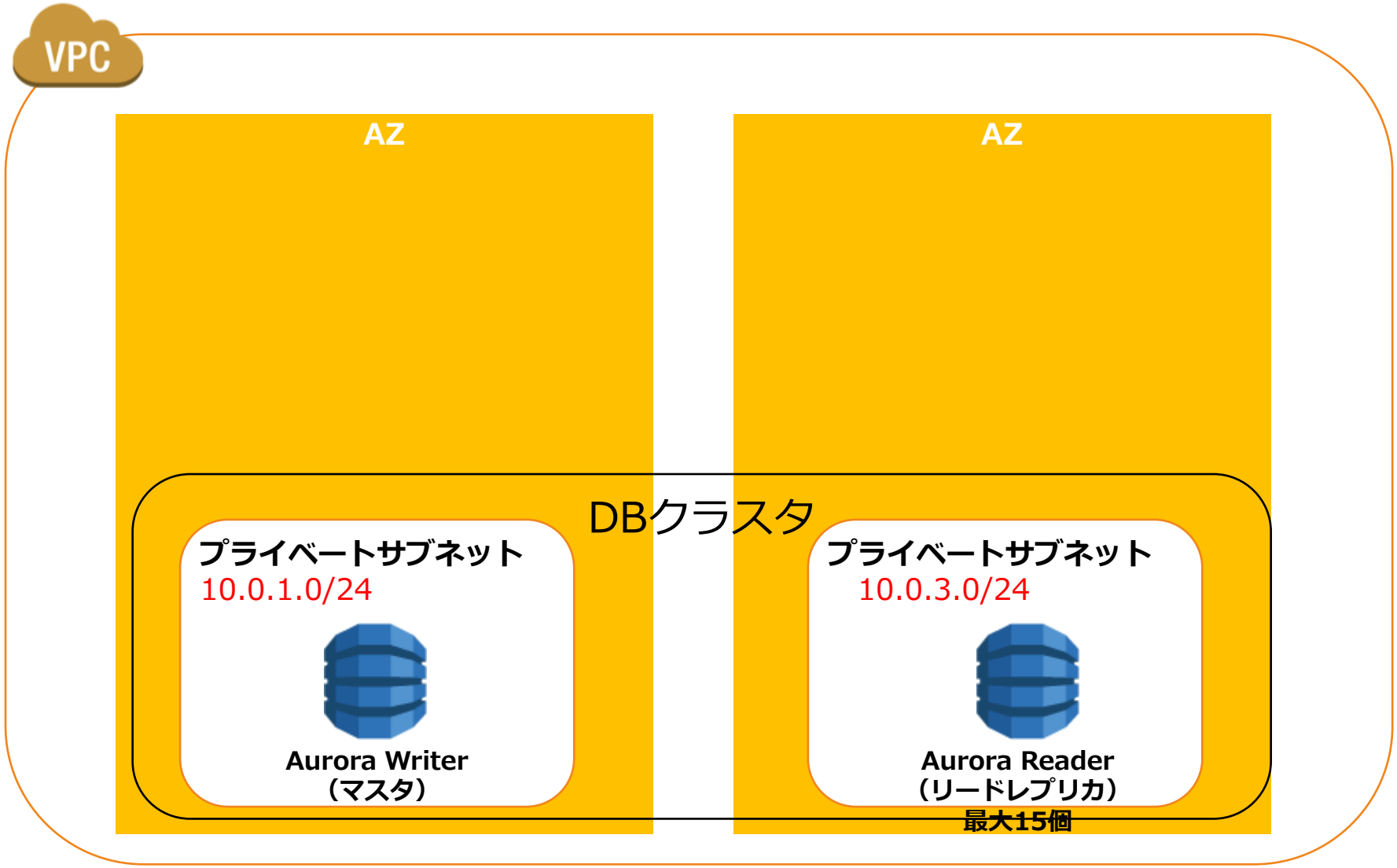
1) Auroraマルチマスター構成へ移行する。
2) ALBをNLBに変更する。
3) Auroraサーバレスに移行する。
4) Amazon Auroraレプリカを追加する
5) CloudFrontのWEBディストリビューションを追加する。

DBクラスタ構成

クラスタ構成を割愛してDBインスタンスのみで次を説明する

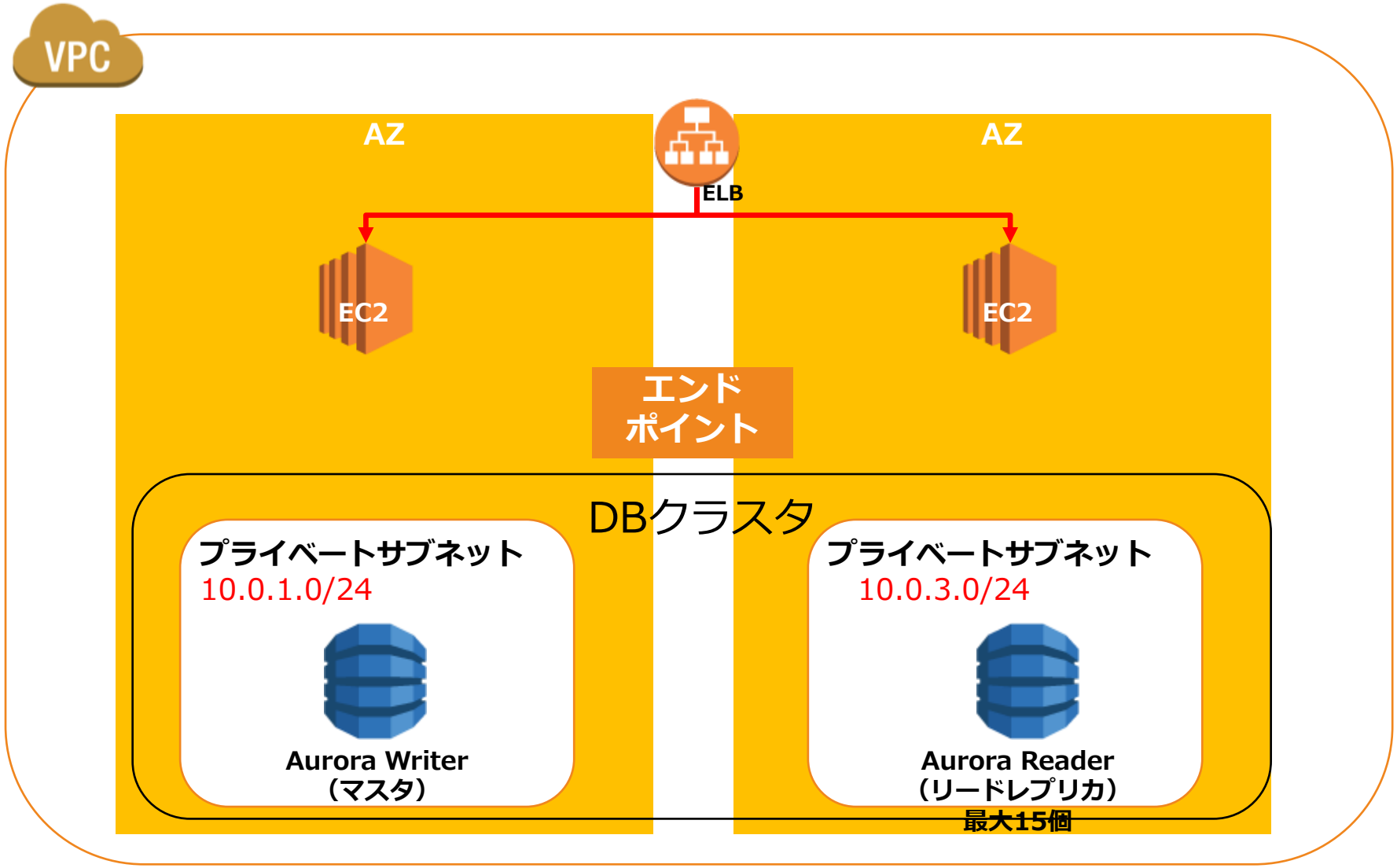
DBクラスタ構成

AuroraはマスタとリードレプリカをまとめたDBクラスタを構成



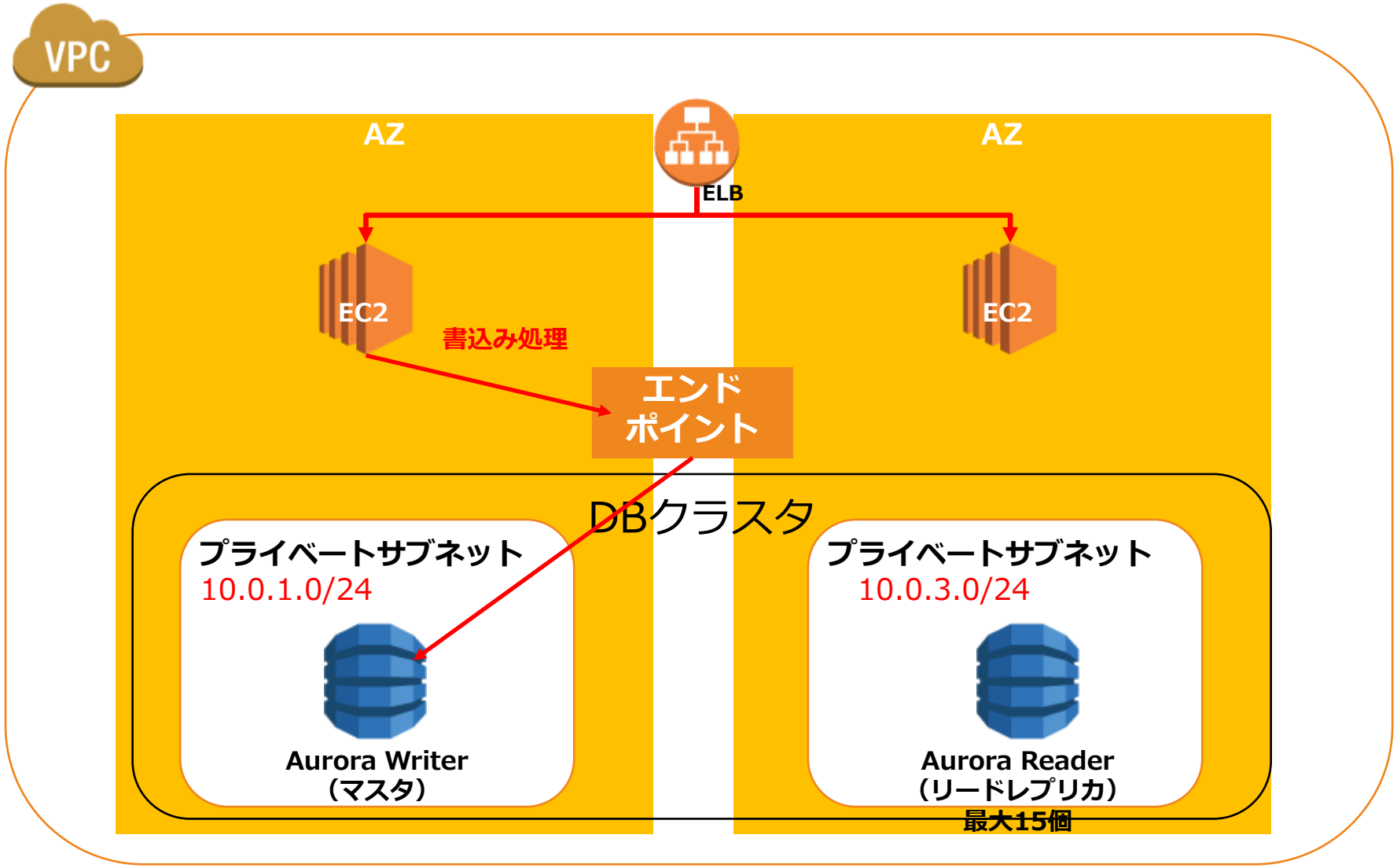
DBクラスタ構成

これらのマスタとレプリカはエンドポイントから接続する



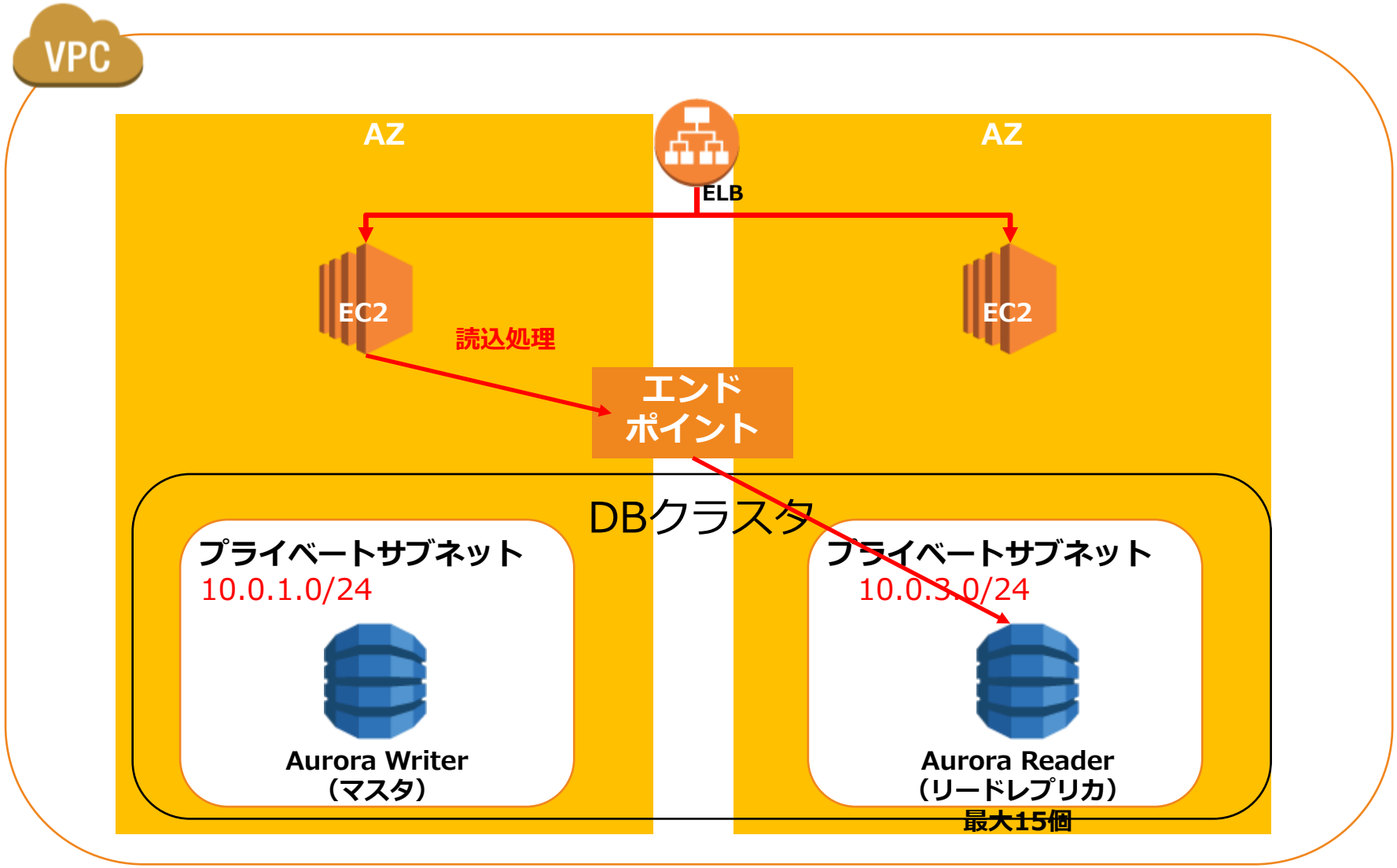
DBクラスタ構成

書込み処理はエンドポイントからWriterが指定される



DBクラスタ構成

読込処理はエンドポイントによりReaderを指定される



[Q]フェールオーバー構成

大手ニュースメディアはAWS上にニュース配信用のWebアプリケーションをホストしています。データベースにはAuroraデータベースを利用しています。同社は現在、読み取りスループットを向上させ、フェイルオーバーターゲットとして使用するために、4つのリードレプリカを複数AZに展開しています。レプリカの構成は以下ように設定しています。

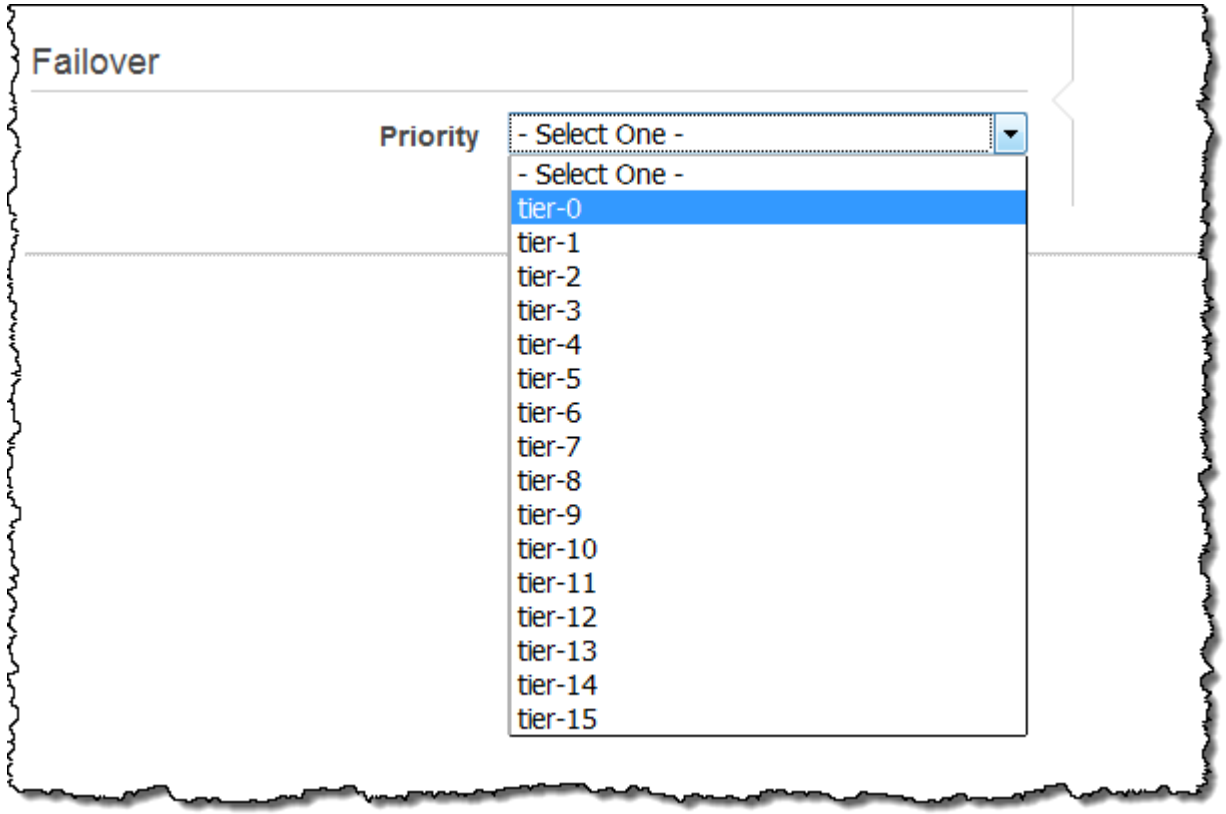
ティア1（8TB）
ティア1（16TB）
ティア15（16TB）
ティア15（ 32TB）

フェールオーバー時に実行されるティアはどれでしょうか？

1) ティア1（8TB）
2) ティア1（16TB）
3) ティア15（16TB）
4) ティア15（ 32TB）

フェールオーバー構成

Auroraはティア番号が小さい最大サイズが多きい順番にリードレプリカをプロモートして、フェールオーバーを実施する。

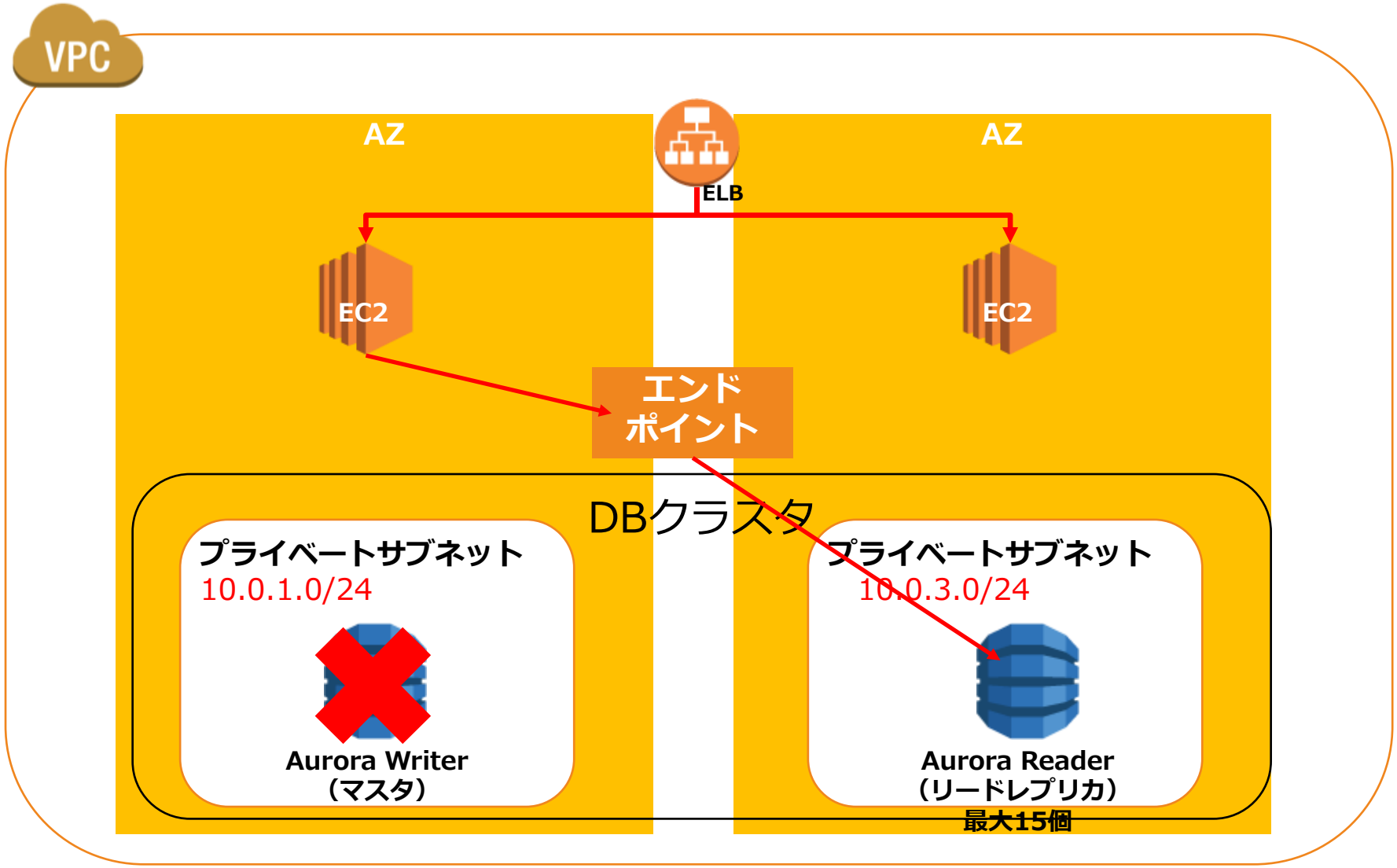


1. Amazon Auroraは、優先度が高い（番号が最も小さいティア）リードレプリカからプロモートする。
2. 2つ以上のAuroraレプリカが同じ優先度を共有している場合、最大サイズのレプリカをプロモートする。

Reference: https://aws.amazon.com/jp/blogs/aws/additional-failover-control-for-amazon-aurora/

フェールオーバーの実行

マスタに障害が発生するとReaderにフェイルオーバーする

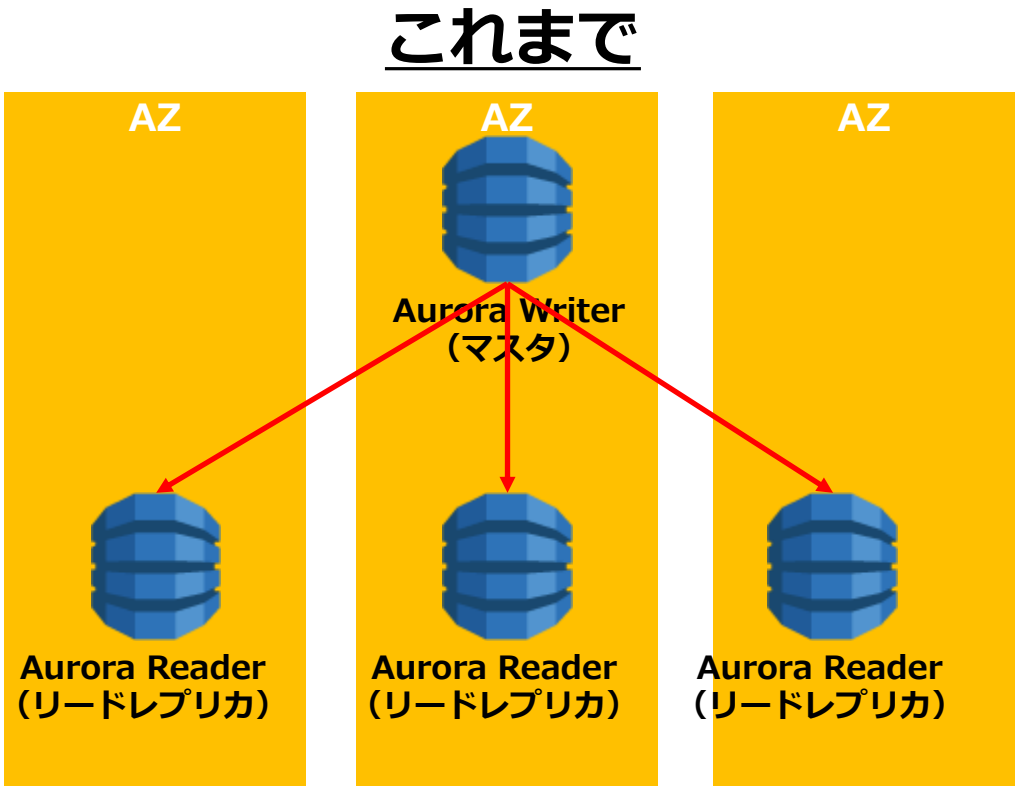


マイグレーション

MySQLとPostgreSQLのスナップショットからAuroraへとマイグレーションが可能

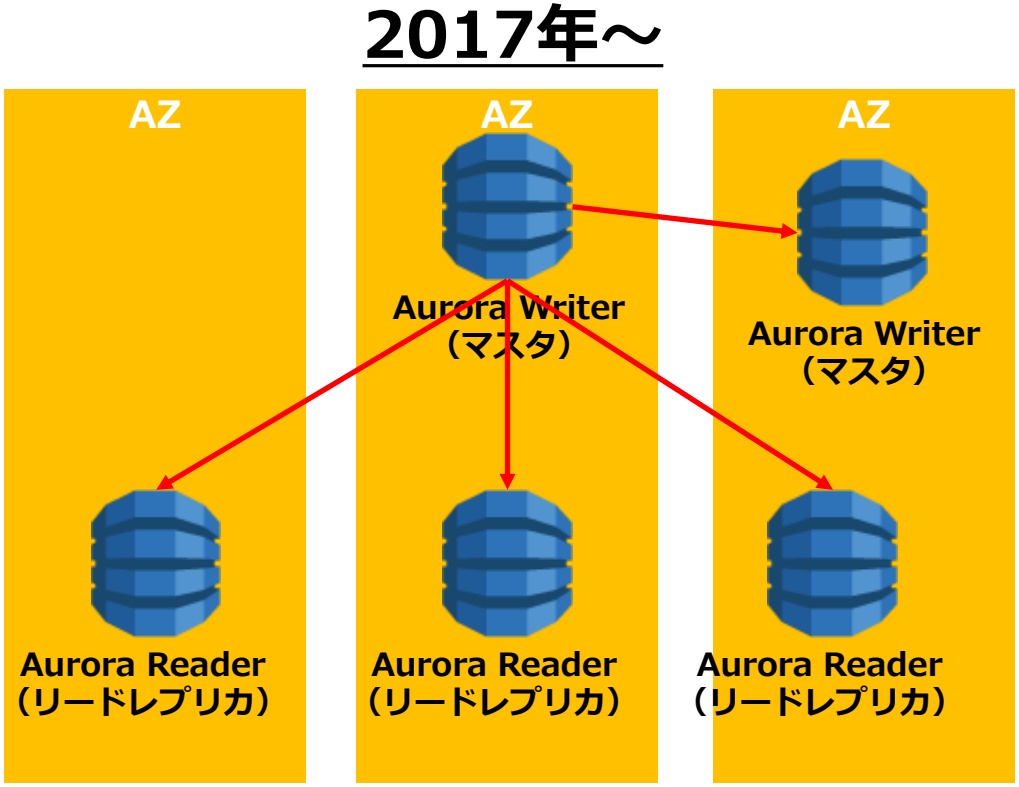
Auroraマルチマスター

マスターデータベースを複数構築してWrite性能もスケーラブルに構築可能



Auroraマルチマスター

マスターデータベースを複数構築してWrite性能もスケーラブルに構築可能



- どのノードが落ちてもダウンタウンがゼロに
- どのAZが落ちてもダウンタイムがゼロに
- Write性能のスケーリング

[Q] Auroraサーバレス

大手IT企業はAWS上にWebアプリケーションを構築しています。このアプリケーションではユーザー数の急増が予測されており、現段階ではどのくらいのパフォーマンスが必要か判断できていません。また、需要減が激しい可能性があり、不規則な処理負荷に対処する必要があります。しかしながら、その予測も事前にできないことが難点となっています。

この要件を満たすためには、どのデータベースが最適でしょうか？

1) RDSのオートスケーリング設定
2) Auroraサーバーレス
3) Auroraマルチマスター構成
4) オンデマンドでオートスケーリングするDynamoDB

Auroraサーバレス

予測困難なアプリケーションワークロードに対応したAuroraのオンデマンド自動スケーリング構成

アプリケーションのニーズに基づいて実行される

・自動的に起動／シャットダウン

・自動でスケールアップ／スケールダウン

[Q]グローバル構成

B社はAWS上にWebアプリケーションを構築しています。このアプリケーションは世界中にユーザーがおり、グローバルで多くのリクエストが発生していることで、Amazon RDS for MySQLにおいてリードレプリカを使用しているにもかかわらずパフォーマンスが低下しています。RDSの基本性能ではパフォーマンスに限界があるようです。

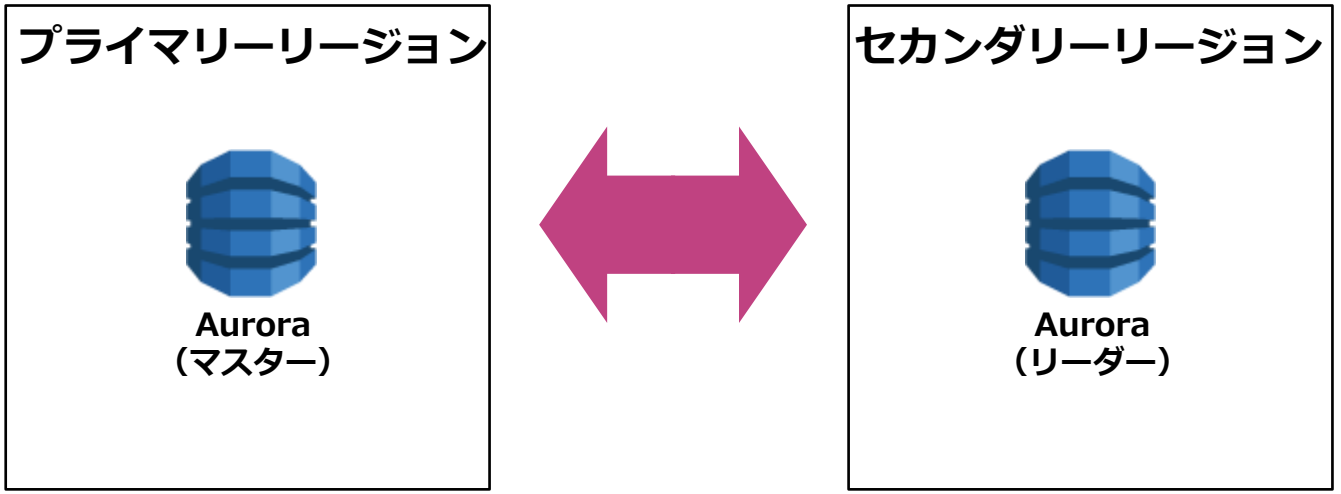
この問題を改善するために、最も費用効果が高く高性能なソリューションを選択してください。

1) Amazon RDSグローバルのリードレプリカを新設して、各リージョンで低レイテンシで高速なローカル読み取りを可能にする。
2) Amazon Auroraグローバルデータベースに移行して、各リージョンで低レイテンシで高速なローカル読み取りを可能にする。
3) Amazon Auroraサーバレスに移行して、各リージョンで低レイテンシで高速なローカル読み取りを可能にする。
4) Amazon DynamoDBグローバルテーブルを移行して、各リージョンで低レイテンシで高速なローカル読み取りを可能にする。

Auroraグローバルデータベース

他リージョンに対する高性能なリードレプリカ作成機能

- ログ転送ではなく、ストレージレベルのレプリケーション機能を利用してレプリケーションを実施
- 概ね1秒以下／最大でも5秒でレプリケーションを実行する低レイテンシーレプリケーションを実現



[Q]エンドポイントの選択

B社はAWS上にWebアプリケーションを構築しています。 このアプリケーションは、ALBにAuto Scalingグループを設定したAmazonEC2インスタンスのフリートで実行されます。データベースとしてAmazon Aurora PostgreSQLを使用しています。あなたはソリューションアーキテクトとして、クラスタ内のデータベースワークロードを最適化する対応を依頼されました。本番トラフィックの書き込み操作を大容量のインスタンスに転送しつつ、レポート作成などの読み込みクエリは低容量のインスタンスに向ける必要があります。

この要件を達成するために、Auroraのエンドポイントの最適設定方式はどれでしょうか？（２つ選択してください。）

1) 本番トラフィックはカスタムエンドポイントに構成する。
2) 読み込みリクエストはカスタムエンドポイントに構成する。
3) 本番トラフィックはインスタンスエンドポイントに構成する。
4) 読み込みリクエストはインスタンスエンドポイントに構成する。
5) 本番トラフィックはクラスターエンドポイントに構成する。
6) 読み込みリクエストは読込エンドポイントに構成する。

エンドポイントの選択

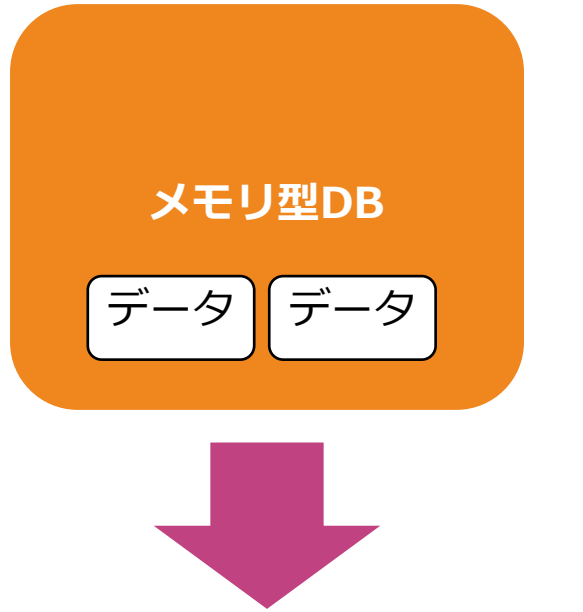
利用したいインスタンスタイプに応じてエンドポイントを選択

クラスター（書き込み）エンドポイント	◘ Auroraクラスターの書き込み処理に利用できるエンドポイント ◘ Writer専用
読み込みエンドポイント	◘ リードレプリカへのアクセスに利用するReaderエンドポイント ◘ リードレプリカ専用
インスタンスエンドポイント	◘ 特定のインスタンスへアクセスするエンドポイント ◘ 個別のインスタンスに設定
カスタムエンドポイント	◘ インスタンスの組み合わせを自由に行えるエンドポイント ◘ WriterとReaderを組み合わせるなど

ElastiCacheの出題範囲

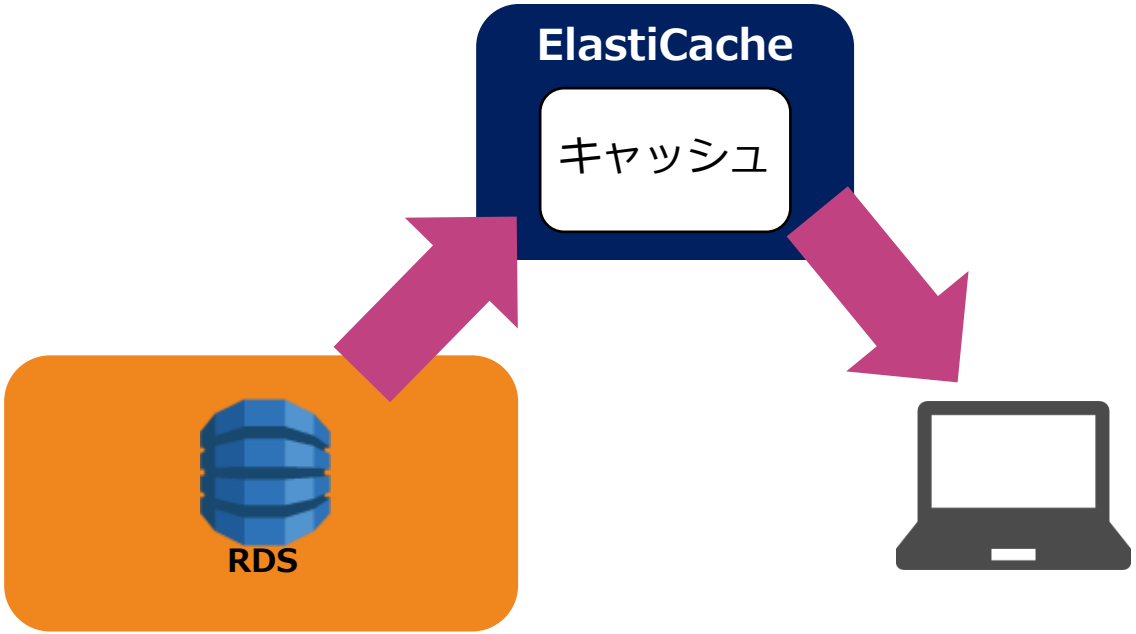
メモリ型DB

メモリDBをキャッシュを保持してデータの高速処理を実現する



ElastiCacheとは何か？

ElastiCacheはメモリにキャッシュを保持して、高速処理を実施するインメモリデータベース



次にアクセスした際はキャッシュからデータを取得する

ElastiCacheの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

ElastiCacheの選択	✓ シナリオに基づいて、データベース要件と合致しているElastiCacheを選択する問題が出題される。
タイプの選択	✓ ElastiCacheのRedisとMemcachedの選択が問われる。 ✓ RedisとMemcachedの違いや特徴に関する問題が出題される。
ElastiCacheの構成	✓ シナリオに基づいて、ElastiCacheの構成したソリューションが問われる。
セキュリティ対応	✓ ElastiCache Redisにおけるセキュリティ設定の方法が問われる。

[Q]ElastiCacheの選択

あなたはゲーム会社のシステム開発担当として、開発中のゲームに利用するデータベースを構築しています。このゲームではユーザー行動データの記録に応じてアイテムが出現する機能を実装する必要があり、ユーザー行動データのリアルタイム高速処理が求められています。

この要件を満たすためのサービスを選択してください。

1) ElastiCache
2) Redshift
3) Aurora
4) RDS

ElastiCache

分散インメモリキャッシュサービスの構築・管理及びスケーリングを容易に実施することができるサービス

- キャッシュクラスタを数クリックで起動
- フルマネージド型でモニタリング、自動障害検出、復旧、拡張、パッチ適用、バックアップに対応し高可用性を実現
- 広く利用されている2種類のエンジンmemcached,／redisから選択可能

ユースケース

データアクセスを高速にしたいケースがあればキャッシュの活用を検討する

【ユースケース】

- セッション管理
- IOT処理とストリーム分析
- メタデータ蓄積
- ソーシャルメディアのデータ処理／分析
- Pub/Sub処理
- DBキャッシュ処理

ユースケース

アプリケーションでデータの即時反映が必要なケースなどに活用する

【ユースケース】

- ユーザーのマッチング処理
- レコメンデーションの結果処理
- 画像データの高速表示
- ゲームイベント終了時のランキング表示

[Q]タイプの選択

あなたはゲーム会社のシステム開発担当として、開発中のゲームにおいて利用するデータベースを構築しています。データベースは、マルチスレッドのメモリ内キャッシュレイヤーを使用して、繰り返されるクエリのパフォーマンスを向上させることが必要です。

このデータベースキャッシュに利用するべきサービスを選択してください。

1) Amazon DynamoDB DAX.
2) Amazon RDS MySQL
3) Amazon ElastiCache Memcached"
4) Amazon ElastiCache Redis

[Q]タイプの選択

あなたはソリューションアーキテクトとして、データを高速処理する仕組みを構築しています。セッションデータ処理においてリアルタイム処理が必要であり、これらのデータ高速処理を実現するためにはElasticCacheが最適であると判断していますが、MemcachedとRedisのどちらを選択するべきか比較することになりました。

ElastiCacheにおけるMemcachedの特徴を選択してください。（2つ選択してください。）

1) シングルスレッドで動作するインメモリキャッシュDBである。
2) pub/sub機能を提供する。
3) スナップショット機能がない。
4) 自動的なフェイルオーバーが可能である。
5) キーストアの永続性は必要ない。

ElastiCache

オープンソースのRedisとMemcachedを利用可能で汎用性あり

Redis

- 高速に値をRead/Writeできるインメモリキャッシュ型DB
- シングルスレッドで動作するインメモリキャッシュDBで全てのデータ操作は排他的
- スナップショット機能がある
- データを永続化できる

Memcached

- 高速に値をRead/Writeできるインメモリキャッシュ型DB
- マルチスレッドで動作するインメモリキャッシュDB
- スナップショット機能がない
- データを永続化できない
- フェイルオーバーや復元ができない。

ElastiCache

シンプルに利用する場合はMemcachedを利用するが、それ以外はRedisを利用する場合が多い

Redis

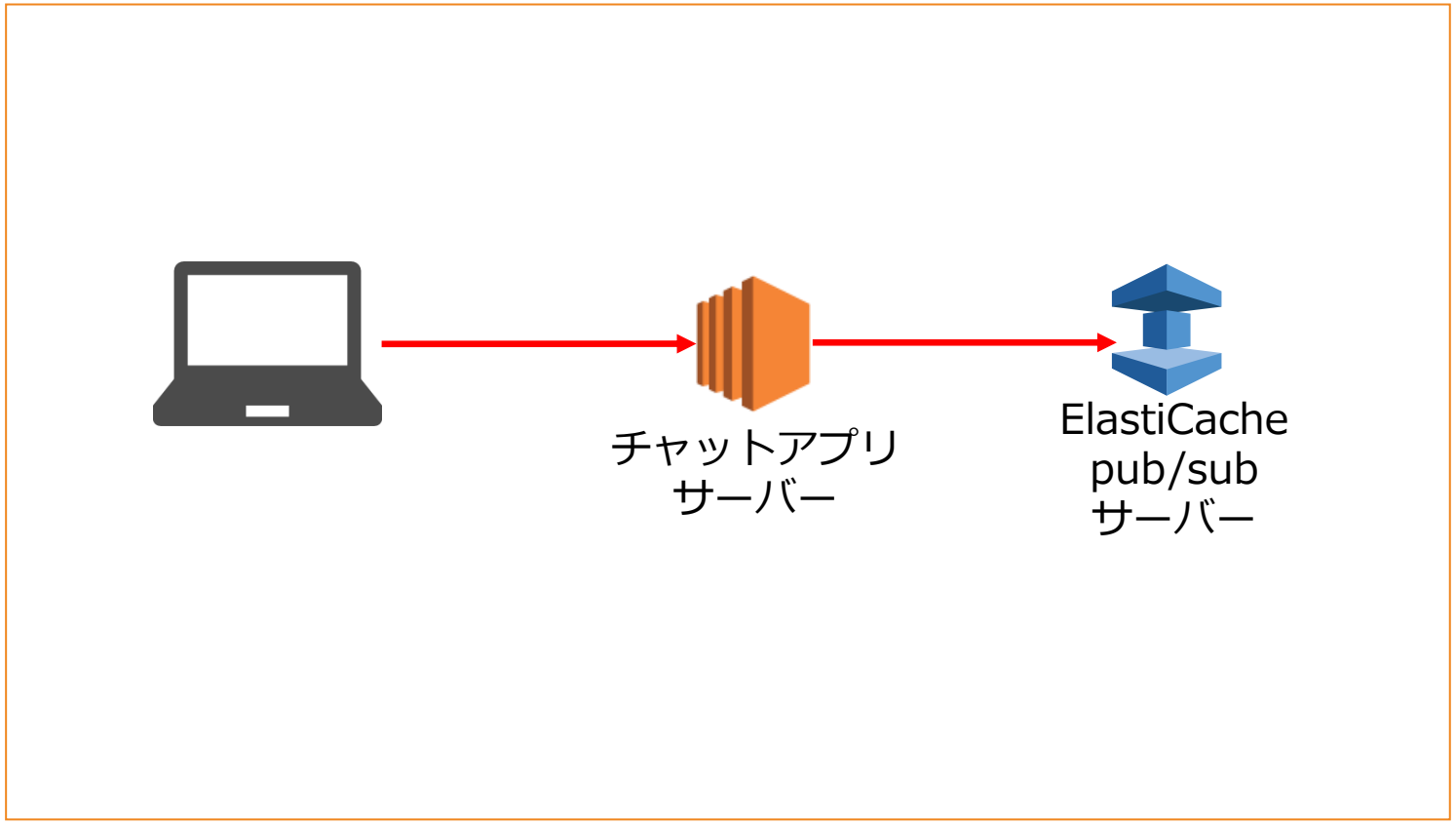
- 複雑なデータ型が必要である。
- インメモリデータセットをソートまたはランク付けする必要がある。
- 読込処理の負荷に対して、リードレプリカにレプリケートする必要がある。
- pub/sub機能が必要
- 自動的なフェイルオーバーが必要である
- キーストアの永続性が必要である。
- バックアップと復元の機能が必要である。
- 複数のデータベースをサポートする必要がある。

Memcached

- シンプルなデータ型が必要である
- 複数のコアまたはスレッドを持つ大きなノードを実行する必要がある。
- システムでの需要の増減に応じてノードを追加または削除するスケールアウトおよびスケールイン機能が必要である。
- データベースなどのオブジェクトをキャッシュする必要がある。
- キーストアの永続性は必要ない
- バックアップと復元の機能が必要でない
- 複数のデータベースを利用できない

ユースケース（チャットアプリ）

ElastiCacheのpub/sub機能を活用したチャットアプリ



ElastiCache with Redis

その他に位置情報クエリ／Luaスクリプトによる操作やpub/subモデルを活用可能

Luaスクリプト	◘ 移植性が高く、高速な実行速度などの特徴を持っているスクリプト言語
位置情報クエリ	◘ 経度・緯度などの位置情報をクエリ処理することが可能 ◘ 検索距離や検索範囲の指定可能
pub/subモデルの利用	◘ 「イベントを起こす側」と「イベント処理を行う側」を分離するのがpub/subモデル ◘ メッセージ処理やイベント処理で活用

[Q]ElastiCacheの構成

あなたはAWSでホストされているアプリケーションで負荷テストを行っています。Amazon RDS MySQL DBインスタンスをテストしているときに、CPU使用率が100%に達する事象が発生し、アプリケーションが応答しなくなるケースがあることが判明しています。このアプリケーションは読取処理が多いようです。

読み取り負荷が高いリクエストに対して高速な処理を実行する構成はどれでしょうか？

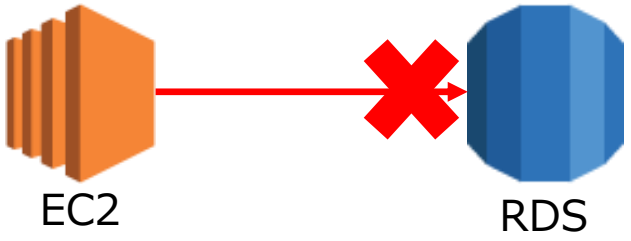
1) RDSへのアクセス集中を軽減するためにSQSによるキューイング処理を利用する
2) RDSインスタンスにAuto Scalingを導入して、負荷に応じてスケーラビリティを高める
3) DynamoDB（DAXクラスター）をRDSの前に設置して、キャッシュ処理を導入する
4) ElasticCacheをRDSの前に設置して、キャッシュ処理を導入する

ElastiCacheの構成

キャッシュすべきデータを特定して、他のDBと合わせて利用するのが標準的な構成方法

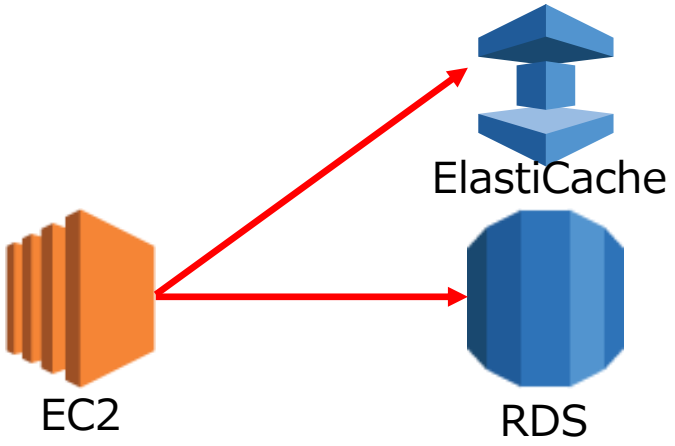
キャッシュ未使用パターン

DBアクセス負荷が増大すると処理能力が低下し可用性が低下する



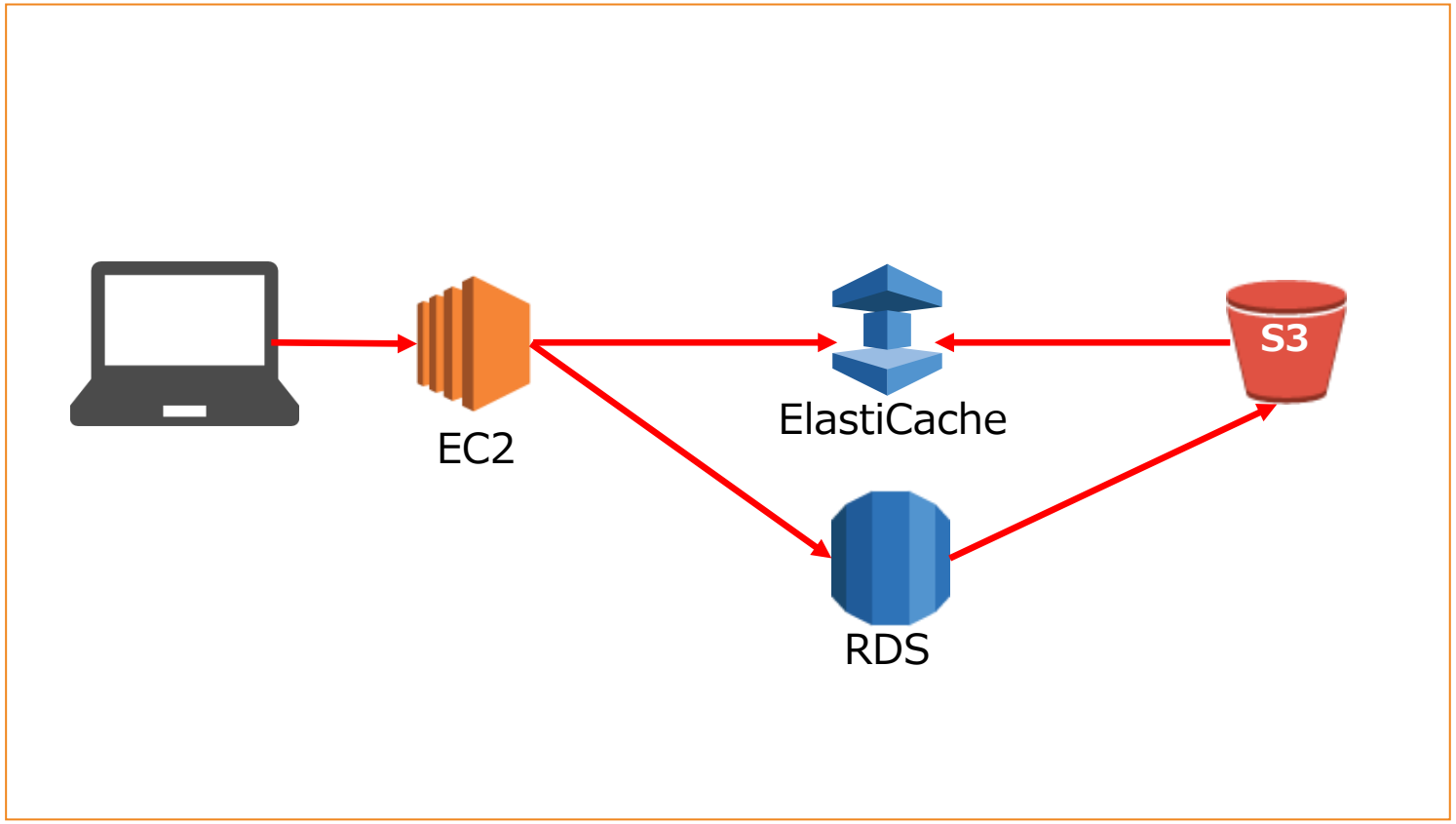
インメモリキャッシュ利用パターン

アクセス頻度の高いデータをキャッシュに配置して可用性を高める



ElastiCacheの構成

ElastiCacheを活用したシンプルなアーキテクチャパターン



[Q]セキュリティ対応

あなたはソリューションアーキテクトとして、データ高速処理の仕組みを構築しています。セッションデータ処理にリアルタイム処理が必要であり、 キャッシングにElastiCache Redisクラスターを使用しています。ユーザー名とパスワードの組み合わせを利用して、Lambda関数からRedisへの認証のセキュリティを強化したいと考えています。

この要件を満たすことができるRedisの設定方法を選択してください。

1) IAM Authを使用して認証する。
2) Lambdaオーソライザーを利用する。
3) Cognitoオーソライザーを利用する。
4) RedisAuthを使用して認証する。

[Q]セキュリティ対応

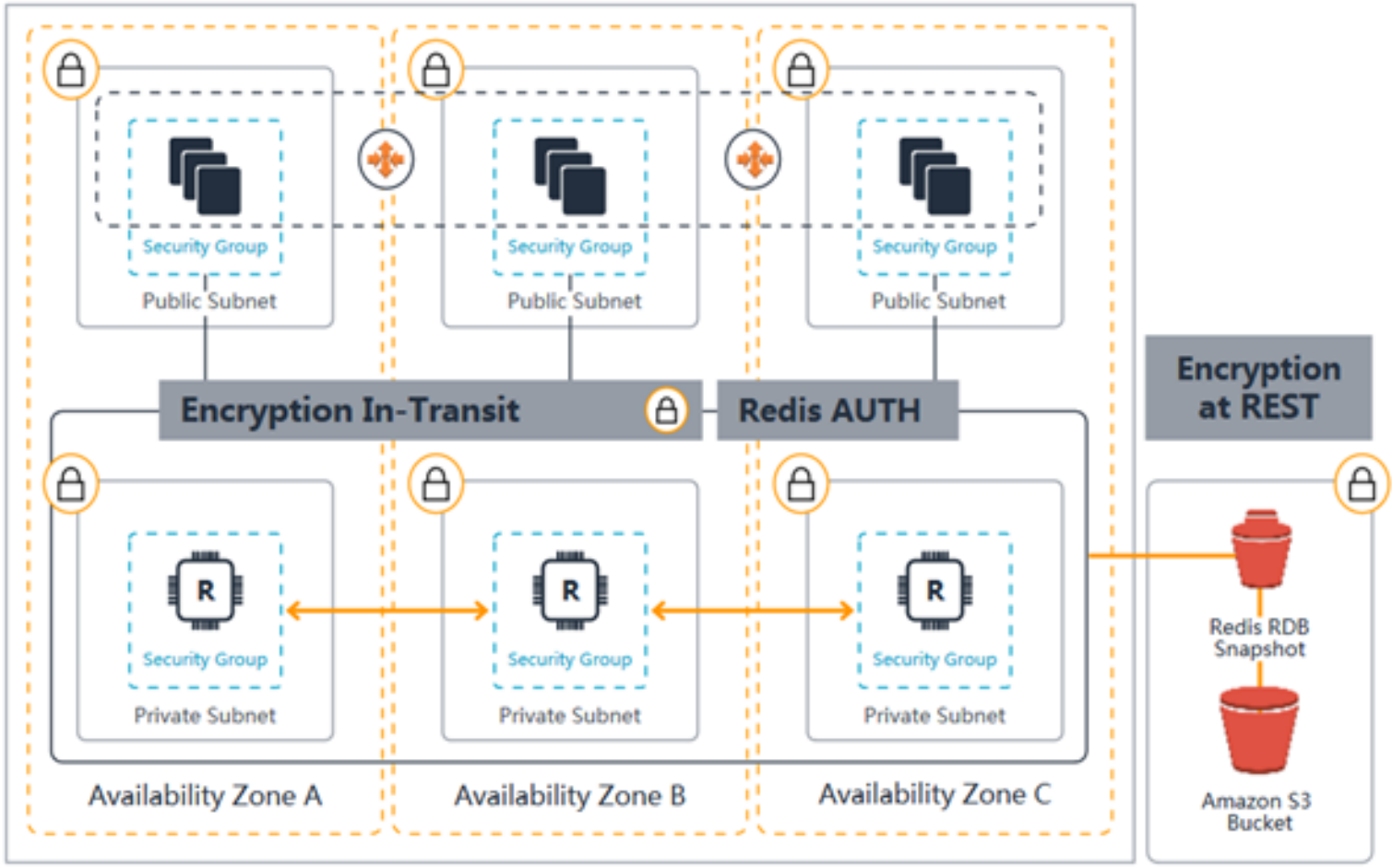
あなたはソリューションアーキテクトとして、データ高速処理の仕組みを構築しています。セッションデータ処理にリアルタイムな処理が必要であり、 キャッシングは、ElastiCacheクラスターを使用しています。会社のセキュリティポリシーに準拠するため、利用されるデータに対して保護が必要です。

データ保護のためのソリューションを選択してください。

1) ElastiCache Redisでデータ転送中の暗号化を有効にする
2) ElastiCache RedisでRedisAUTHコマンドを発行する。
3) ElastiCache Redisで保存時の暗号化を有効にする
4) ElastiCache Memcachedで保存時の暗号化を有効にする
5) ElastiCache Memcachedでデータ転送中の暗号化を有効にする

セキュリティ対応

ElastiCache Redisはデータ転送時の暗号化、データ保管時の暗号化およびRedis Authによる認証を実施することが可能



Reference: https://docs.aws.amazon.com/ja_jp/AmazonElastiCache/latest/red-ug/encryption.html

セキュリティ対応

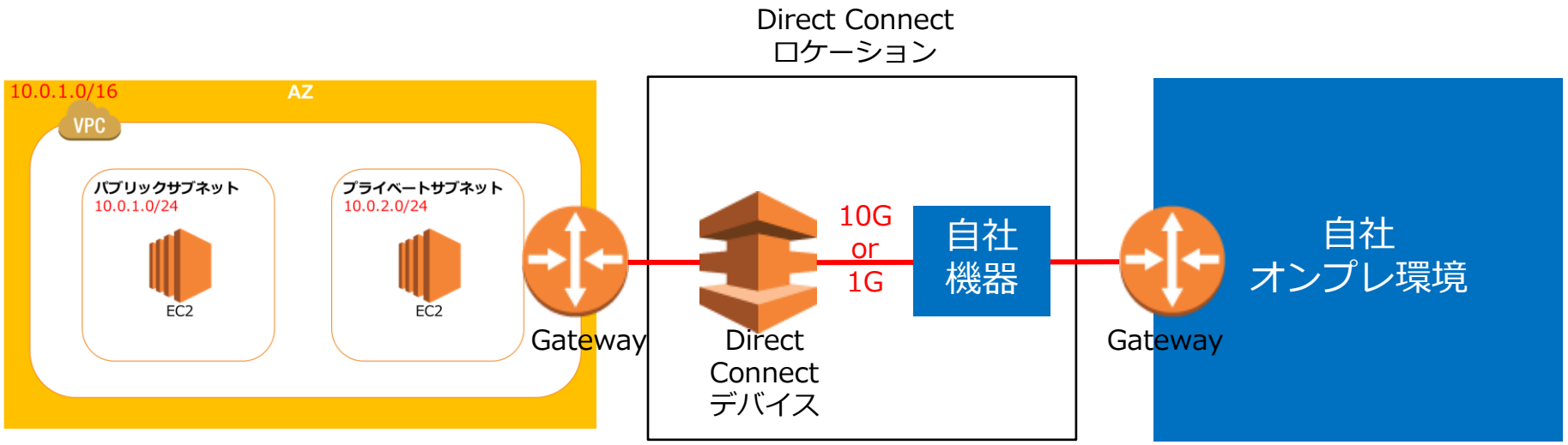
ElastiCache Redisはデータ転送時の暗号化、データ保管時の暗号化およびRedis Authによる認証を実施することが可能

保管時の暗号化	◘ AWS KMSを利用して同期やバックアップオペレーションの実行中にオンディスクデータが暗号化される。 ◘ バックアップ対象は同期・バックアップ・スワップ操作中のディスクとAmazonS3に保存されているバックアップ
通信の暗号化	◘ ある場所から別の場所に移動するデータ (クラスターのノード間、クラスターとアプリケーション間など) に暗号化を実施 ◘ Redisレプリケーショングループの作成時にのみ有効化が可能となる。
Redis Auth	◘ Redis認証トークンを使用して、クライアントがコマンドを実行できるようになる前にトークン(パスワード)を要求して認証を実施する。

サイト間接続の出題範囲

サイト間接続のユースケース

オンプレミスとAWSクラウド間の接続には専用線であるDirect ConnectまたはサイトVPNを利用する



サイト間接続の出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

接続方式の選択	✓ オンプレミス環境とAWSクラウドを接続するサービスや機能を選択する問題が出題される。
Direct Connect構成	✓ Direct Connect構成に利用する各種ゲートウェイやインターフェースの設定方法に関する質問が問われる。
リージョン間の接続	✓ リージョン間の接続する際のDirect Connectゲートウェイの構成方法が問われる。
サイト間VPN接続の設定	✓ サイト間VPNを設定する際のゲートウェイの構成方法が問われる。
VPN CloudHub	✓ VPN CloudHubを利用した複数のVPNをまとめる設定方法の活用が問われる。

サイト間接続の出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Direct Connectの冗長化	✓ Direct Connectの冗長性を担保する構成方法が問われる。

[Q]接続方式の選択

あなたはソリューションアーキテクトとして、社内のオンプレミスのインフラストラクチャをAWSクラウドネットワークに接続するための対応を行っています。

この要件を満たすための方法を選択してください。（2つ選択してください。）

1) Direct Connect
2) VPC Peering
3) VPN
4) Snowball
5) AWS Storage Gateway

VPCとのオンプレミス接続

VPN接続

専用線接続
（Direct connect）

VPNとのDirect Connect

VPNの方が安く素早く利用できるが、信頼性や品質は専用線が勝る

	VPN	専用線
コスト	✓ 安価なベストエフォート回線が利用可能	✓ キャリアの専用線サービス契約が必要となりVPNより割高
リードタイム	✓ クラウド上での接続設定で可能なため即時	✓ 物理対応が必要なため数週間
帯域幅	✓ 暗号化のオーバーヘッドにより制限がある	✓ ポートあたり1G/10Gbps
品質	✓ インターネット経由のためネットワーク状態の影響を受ける	✓ キャリアにより高い品質が保証される
障害切り分け	✓ インターネットベースのため自社で保持している範囲以外の確認は難しい	✓ 物理的に経路が確保されているため比較的容易

[Q]Direct Connect構成

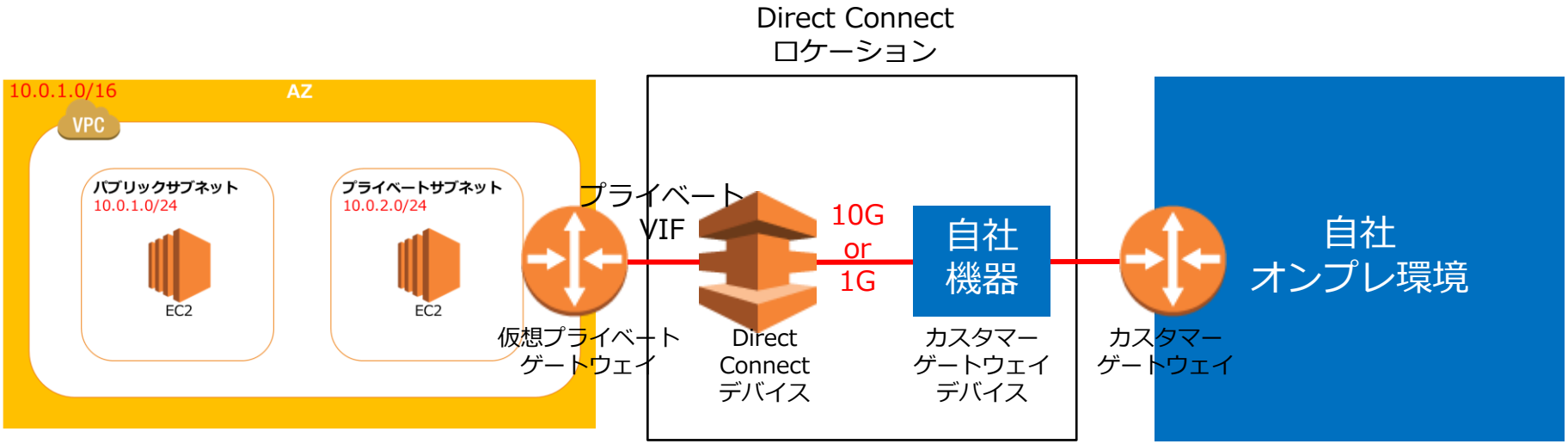
あなたが働いている会社は現在、インフラストラクチャとアプリケーションをAWSクラウドに移行しています。あなたはソリューションアーキテクトとして、オンプレミス環境との接続にDirect Connectを利用した接続構成を実施しています。

Direct Connectの構成をどのように実施すべきか選択してください。（2つ選択してください。）

1) オンプレミス環境にカスタマーゲートウェイデバイスを設置して、Direct Connectデバイスと接続する。
2) Amazon VPC側に仮想プライベートゲートウェイを設置して、 Direct Connectデバイスと接続する。
3) オンプレミス環境に仮想プライベートゲートウェイを設置して、Direct Connectデバイスと接続する。
4) Amazon VPC側にカスタマーゲートウェイデバイスを設置して、 Direct Connectデバイスと接続する。
5) オンプレミス環境にプライベート仮想インターフェースを設置して、 Direct Connectデバイスと接続する。

Direct Connect構成

Direct Connectロケーションに物理的に自社オンプレ環境を接続することでAWS環境との専用線接続を実現する



[Q]リージョン間接続

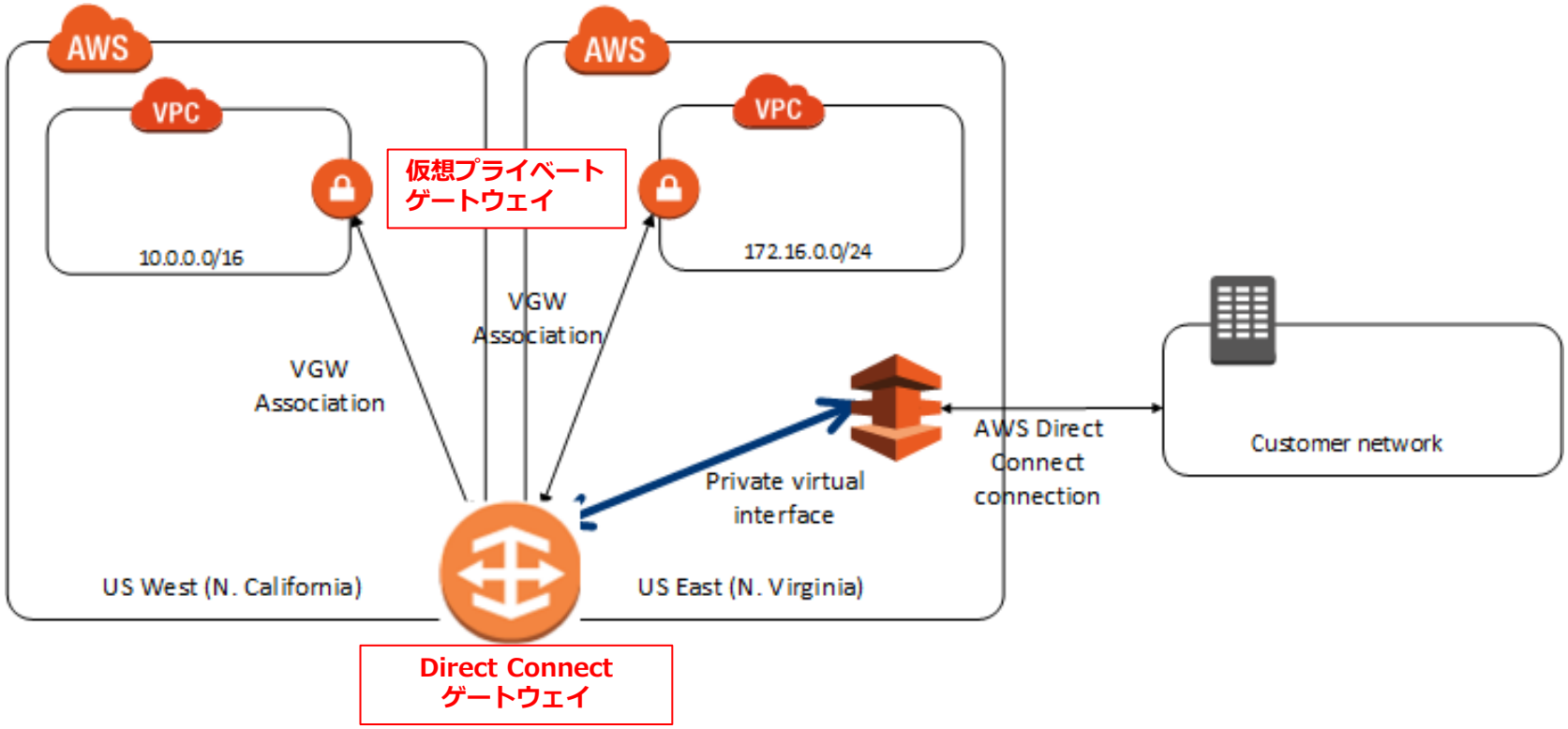
グローバルコンサルティングファームは世界中にオフィスをもっています。AWSを利用したドキュメント共有システムを構築しており、各国の知見を共有する計画をしています。この仕組みを実装するためには、同じアカウント内の複数のリージョンにある複数のVPCへの高帯域幅で低遅延の接続を実装する必要があります。 VPCにはそれぞれ固有のCIDR範囲があります。

この要件を満たすことができる最適なソリューション設計はどれでしょうか？（2つ選択してください）

1) DirectConnectゲートウェイを作成し、各リージョンにカスタマー仮想インターフェースを作成する。
2) オフィスからAWSリージョンへのDirect Connect接続を構成する。
3) オフィスからAWSリージョンへのVPN接続を構成する。
4) それぞれのAWSリージョンにDirect Connect接続を実装する
5) DirectConnectゲートウェイを作成し、各リージョンへのプライベート仮想インターフェースを作成する。

リージョン間接続

Direct Connect gatewayにより、同一アカウントに所属する複数リージョンの複数AZから複数リージョンの複数VPCに接続



Reference: https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways-intro.html

[Q]サイト間VPN接続の設定

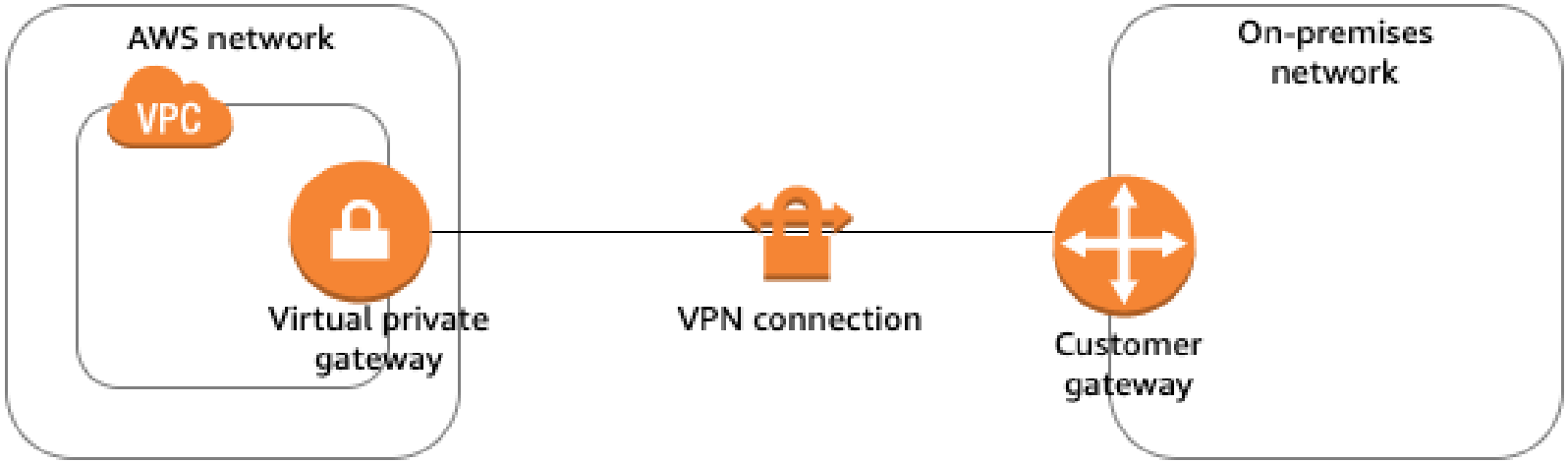
小売企業は、東京リージョンを利用してAWSクラウドへと移行する計画をしています。そのためには、オフィスとAWSクラウドとを接続する構成が必要です。あなたはソリューションアーキテクトとして、リモートのオンプレミスネットワークとインターネットを介したVPCの間にAWSマネージドIPSec VPN接続をセットアップしました。

次のうち、IPSec VPN接続の正しい構成を表すものはどれですか。

1) VPNのAWS側に仮想プライベートゲートウェイを作成し、VPNのオンプレミス側にカスタマーゲートウェイを作成する。
2) VPNのオンプレミス側に仮想プライベートゲートウェイを作成し、VPNのAWS側にカスタマーゲートウェイを作成する。
3) VPNのAWS側に仮想カスタマーゲートウェイを作成し、VPNのオンプレミス側にカスタマーゲートウェイを作成する。
4) VPNのオンプレミス側に仮想カスタマーゲートウェイを作成し、VPNのAWS側にカスタマーゲートウェイを作成する。

サイト間VPN接続の設定

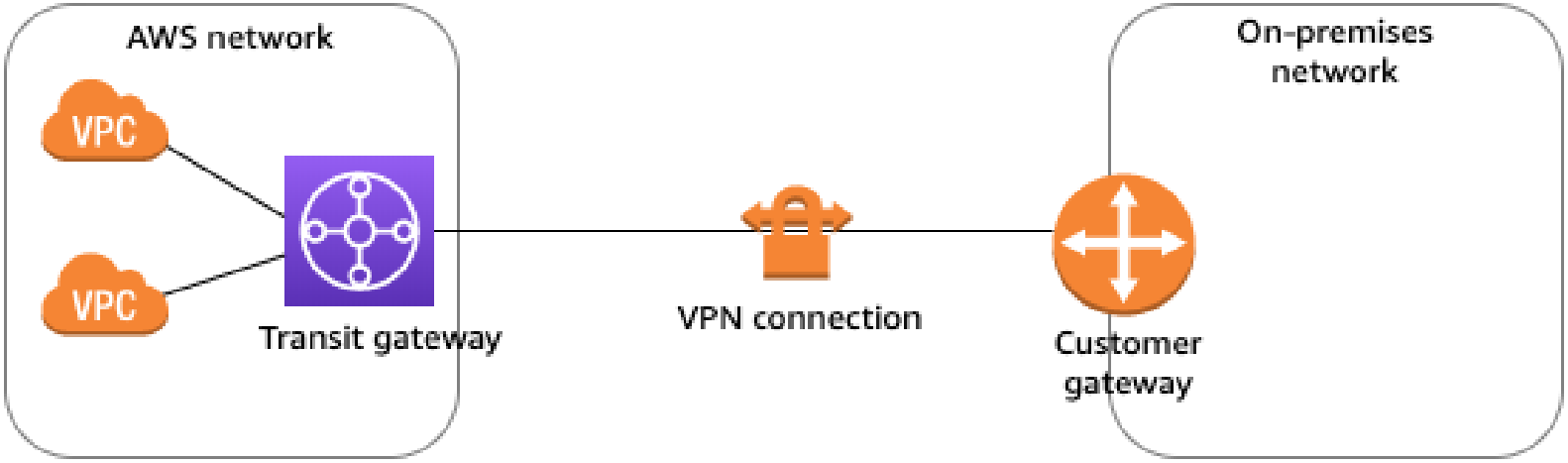
オンプレミス環境のカスタマーゲートウェイデバイスとAWS側の仮想プライベートゲートウェイを接続する。



Reference: https://docs.aws.amazon.com/ja_jp/vpn/latest/s2svpn/how_it_works.html

サイト間VPN接続の設定

AWS側の仮想プライベートゲートウェイをTransit Gatewayにすることも可能



Reference: https://docs.aws.amazon.com/ja_jp/vpn/latest/s2svpn/how_it_works.html

[Q] VPN CloudHub

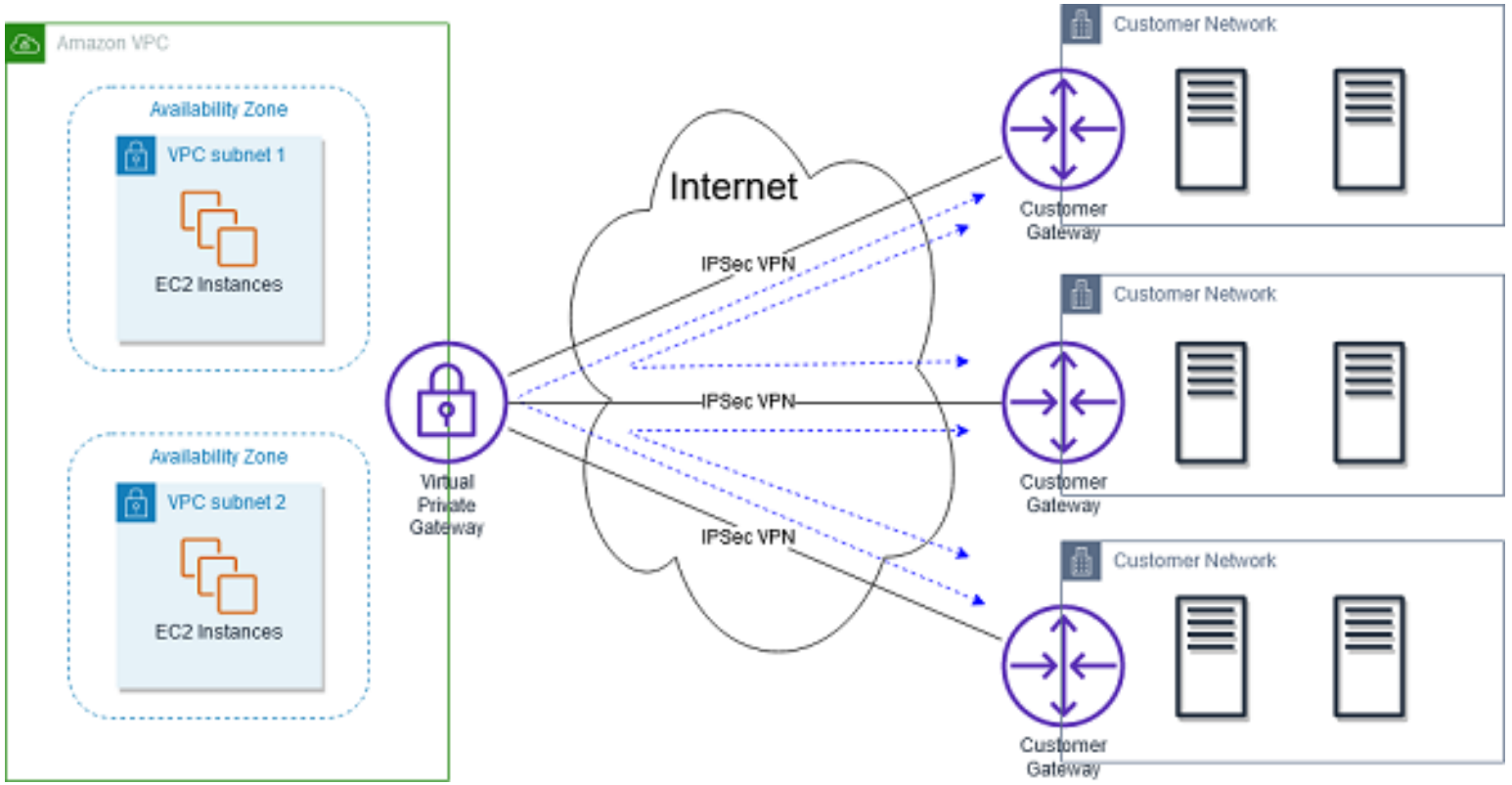
メディア企業は東京リージョンにおいてDirect Connectを利用してオフィスとAWSクラウドを接続しています。シンガポールとシドニーの各支店は別リージョンを利用しており、サイト間VPN接続を使用してVPCに接続しています。 同社は、支店が相互に、および本社とデータを送受信するためのソリューションを探しています。

この要件を満たすことができるAWSサービスを選択してください。

1) VPN CloudHub
2) VPC カスタマーゲートウェイ
3) VPCエンドポイント
4) AWS Transit Gateway

VPN CloudHub

複数の サイト間VPN 接続をまとめて、安全なサイト間通信を提供することができる。



Reference: https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-vpn-cloudhub.html

[Q]Direct Connectの冗長化

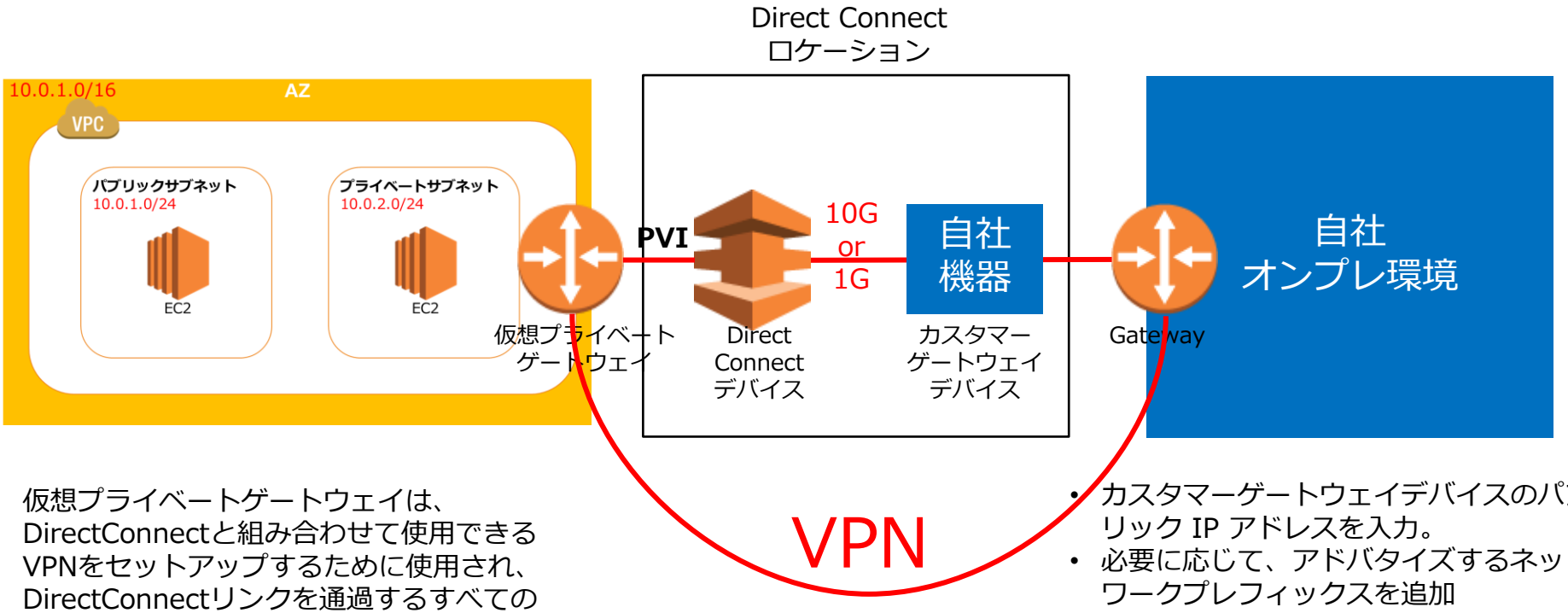
あなたの会社でDirect Connectを利用してオフィスとAWSクラウドを接続しています。ただし、Direct Connect接続 1 つだけが構成されており、冗長性が担保されていないことが問題となっています。あなたはソリューションアーキテクトとして、Direct Connect 接続の冗長性を高めるように依頼されました。その際にコスト最適な構成が求められています。

この要件を満たすことができるソリューションはどれでしょうか？

1) Direct Connectを二重に設定して冗長構成を実施する。
2) サイト間VPNをバックアップ接続として使用する。
3) プライマリ接続としてサイト間VPNを使用する。
4) 出力専用インターネットゲートウェイをバックアップ接続として使用する。

Direct Connectの冗長化

拠点間をeBGPにてピア接続を行った上で、同一拠点間をiBGPにてピア接続を行うことで、VPN接続の冗長化を実施する。



仮想プライベートゲートウェイは、DirectConnectと組み合わせて使用できるVPNをセットアップするために使用され、DirectConnectリンクを通過するすべてのデータを暗号化する。

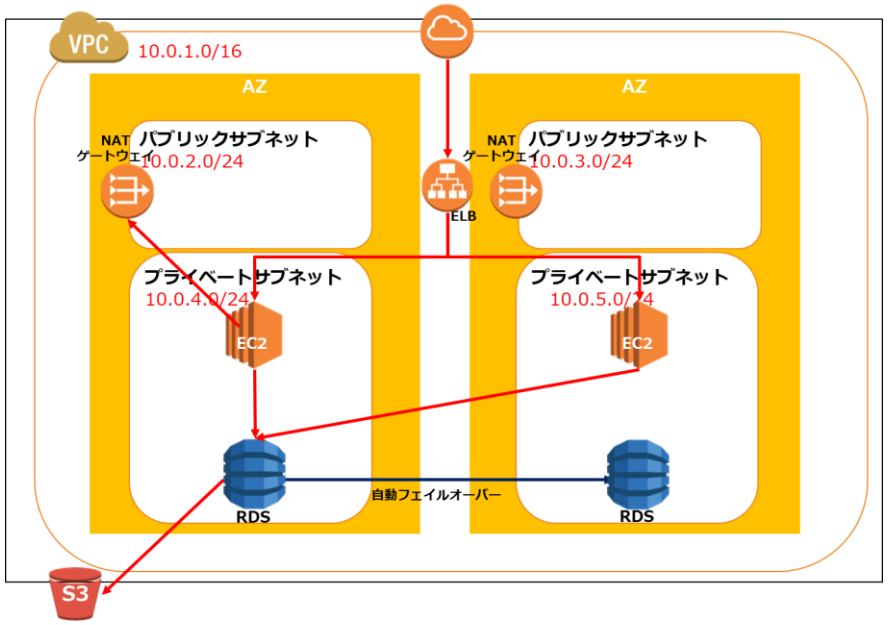
- カスタマーゲートウェイデバイスのパブリック IP アドレスを入力。
- 必要に応じて、アドバタイズするネットワークプレフィックスを追加

CloudFormation
の出題範囲

CloudFormationとは何か？

テンプレートに基づいて、AWSインフラ構成をデプロイする環境自動化サービス

```
EC2Instance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        commands:
          1_pvcreate:
            command: pvcreate /dev/xvdf
          2_vgcreate:
            command: vgcreate vg0 /dev/xvdf
          3_lvcreate:
            command: lvcreate -l 100%FREE -n myapp vg0
          4_mkfs:
            command: mkfs.ext4 /dev/vg0/myapp
          5_mkdir:
            command: mkdir /var/myapp
          6_fstab:
            command: echo "/dev/mapper/vg0-myapp /var/myapp ext4 defaults 0 2" >> /etc/fstab
          7_mount:
            command: mount -a
  Properties:
    BlockDeviceMappings:
      <snip>
    UserData:
      Fn::Base64: !Sub |
       #!/usr/bin/env bash
       set -e errexit
       yum -y update aws-cfn-bootstrap
       /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --region ${AWS::Region}
       /opt/aws/bin/cfn-signal --exit-code $? --stack ${AWS::StackName} --resource EC2Instance --region
```



CloudFormationの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

CloudFormationの選択	✓ AWSクラウド上で実施したい要件が提示されており、CloudFormationを選択する問題が問われる。
CloudFormationの機能	✓ スタックセットを活用して、CloudFormationを複数アカウントにまたがって展開する方法が問われる。
CloudFormation テンプレートの要素	✓ CloudFormationテンプレートの要素への理解が問われる。
CloudFormation テンプレート内容	✓ CloudFormationテンプレートの記述内容の理解が問われる。

[Q]CloudFormationの選択

会社ではインフラ構成を標準化するためのガイドラインを作成しました。あなたはソリューションアーキテクトとして、AWSリソースを利用する際のEC2インスタンスやVPCなどの構成をガイドラインに沿った展開を共有するための仕組みを整備しています。

この要件に適したテクノロジーの選択はどれですか？

1) CloudFormation
2) AWS Elastic Beanstalk
3) AWS Systems Manager
4) CodeDeploy

CloudFormation

環境構築を正確に実施しかつ効率的に展開したいときにCloudFormationを活用できる

ユースケース

- AWSリソースの構築を効率化したい
- 開発・テスト・本番環境で利用するインフラを標準化したい
- 毎回同じリソースやプロビジョニング設定を正確に利用したい
- ソフトウェアと同じように環境構成を管理したい

CloudFormation

AWSクラウド環境内の全インフラリソースを記述してテンプレート化して展開する環境自動設定サービス

- プロビジョニングされたリソースの変更・削除が可能
- JSON／YAMLで記述する
- クロスリージョンとクロスアカウントで管理
- 直接サポートされていないリソースや機能を利用する場合はカスタムリソースでスタック作成の一部に独自ロジックを組み込むことが可能

CloudFormationの構成

テンプレートで定義された内容をよみこんでAWSリソースの集合であるスタックを作成する

【テンプレート】

JSON/YAMLでリソースとパラメーターを定義

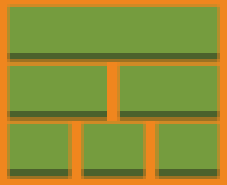
【Cloud Formation】

スタックの作成・変更・削除
エラー検知とロールバック

リソース間依存関係を自動判定

【スタック】

AWSリソースの集合
スタック単位で管理可能で
スタックを削除すると紐づいたリソースも削除される



[Q] CloudFormationの機能

B社ではCloudFormationテンプレートを作成して、インフラ環境設定を標準化しています。あなたはソリューションアーキテクトとして、テンプレートを展開後にインフラ構成を変更しているため、その変更点を確認することが必要です。

この要件を満たすCloudFormationの機能を選択してください。

1) AWS CloudFormation スタックセットを使用して、変更点を確認する。
2) AWS CloudFormationテンプレートを使用して、変更点を確認する。
3) AWS CloudFormation変更セットを使用して、変更点を確認する。
4) AWS CloudFormationドリフトを使用して、変更点を確認する。

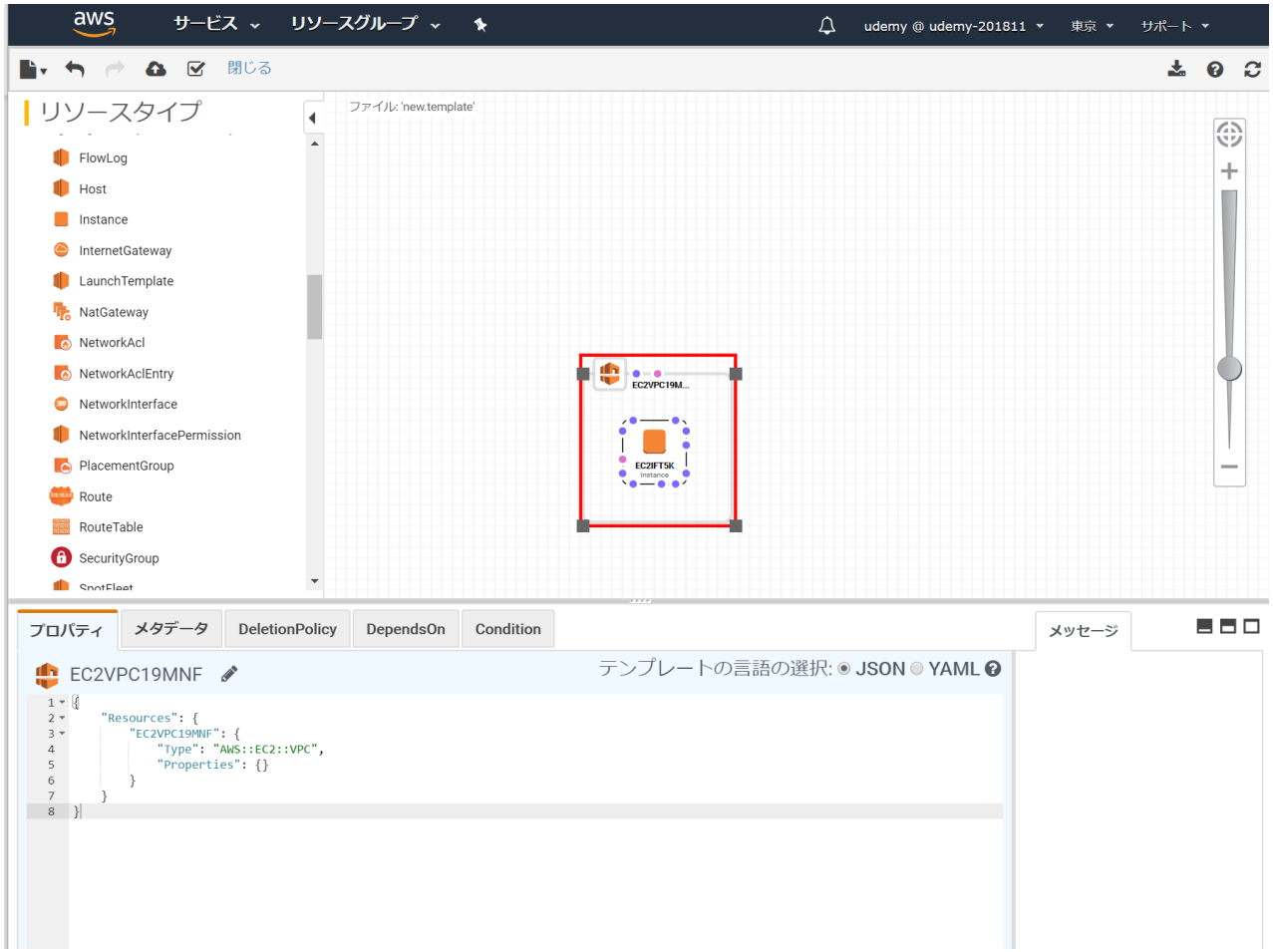
CloudFormationの機能

CloudFormationテンプレート自体を管理・便利に使うための機能を提供

変更セット	スタックの更新をおこなう時の概要が変更セットで、変更による影響度を確認するためのスタック スタック変更は直接更新と変更セットの実行で可能
ドリフト	テンプレートによって展開したAWSリソースを展開後に変更した場合に、元テンプレートとの差分を検出するチェック機能
スタックセット	複数のAWSアカウントと複数のリージョンに対してスタックを作成できる機能
スタック間のリソース参照機能	被参照テンプレートの参照値をエクスポートして値を抽出し、その後参照先のテンプレートのインポートによりリソース参照を行うことで連携したインフラ展開が可能になる機能

CloudFormationの機能

CloudFormationデザイナーを利用することで視覚的にテンプレートを作成できる



[Q] CloudFormationテンプレートの要素

あなたはソリューションアーキテクトとして、CloudFormationテンプレートを作成して、環境設定内容を標準化する対応をしています。AWSスタックの作成時に他のテンプレートで参照できるように一部の設定内容を出力することが必要です。

テンプレートのどのセクションを設定する必要がありますか？

1) Value
2) Outputs
3) Properties
4) Mappings

[Q] CloudFormationテンプレートの内容

あなたはソリューションアーキテクトとして、CloudFormationテンプレートを作成して、環境設定内容を標準化する対応をしています。

ーーーーこれより上は省略ーーーー

```
Mappings:
  RegionMap:
    ap-northeast-1:
      hvm: "ami-0792756bc9edf3e63"
    ap-southeast-1:
      hvm: "ami-0162da29310cc18f6"
Description: Create EC2 Instance
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap [RegionMap, !Ref 'AWS::Region', hvm]
      InstanceType: !Ref InstanceType
      Tags:
      - Key: Name
        Value: myInstance
```

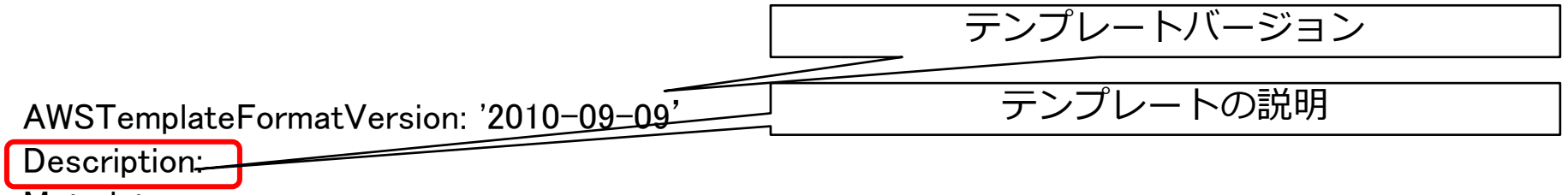
このCloudFormationテンプレートはどのような設定となりますか？ （3つ選択してください）

テンプレート

テンプレートバージョン

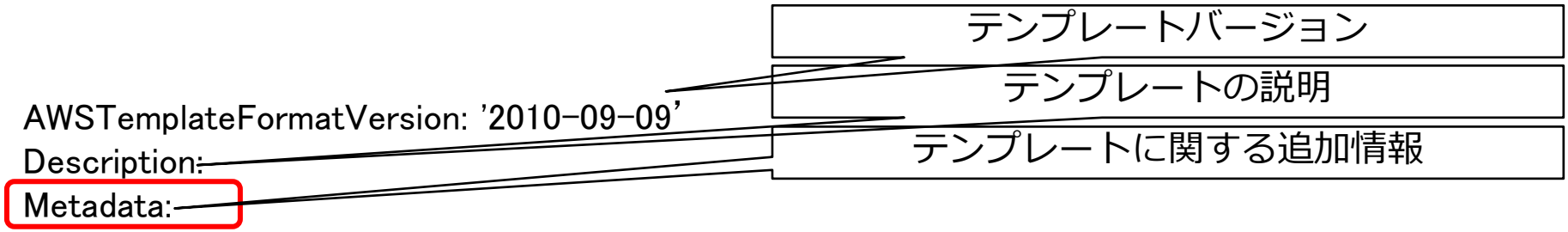
```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート




```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート




```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート

テンプレートバージョン

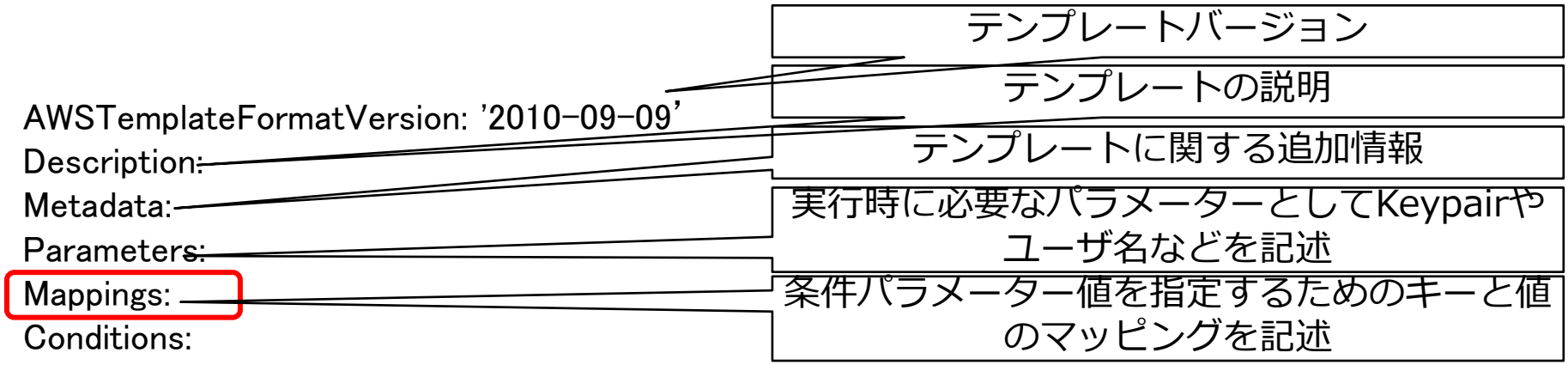
テンプレートの説明

テンプレートに関する追加情報

実行時に必要なパラメーターとしてKeypairやユーザ名などを記述

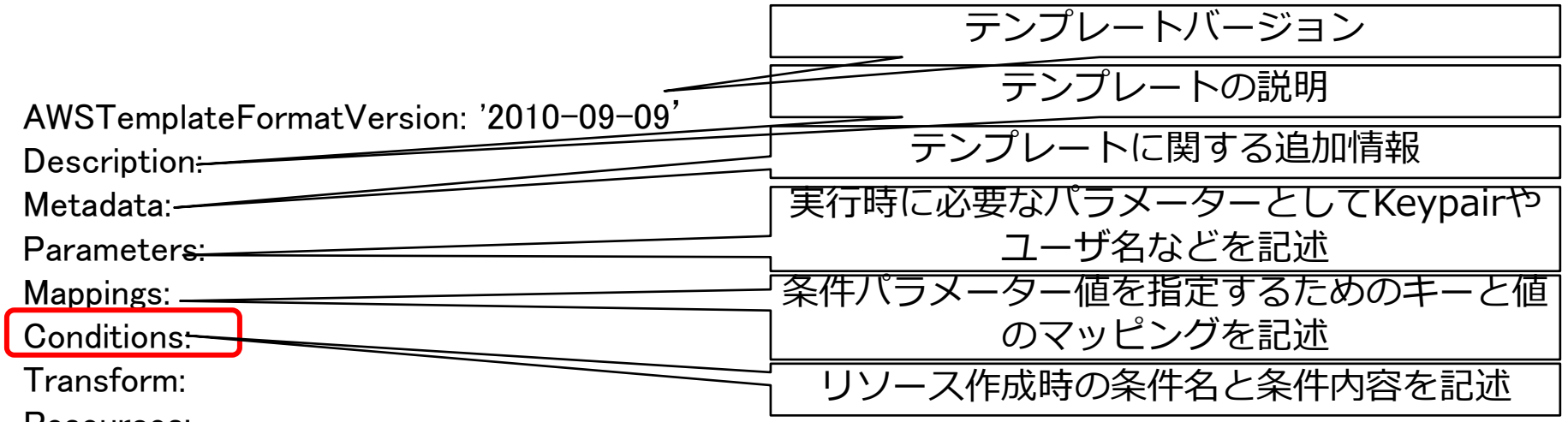
```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート

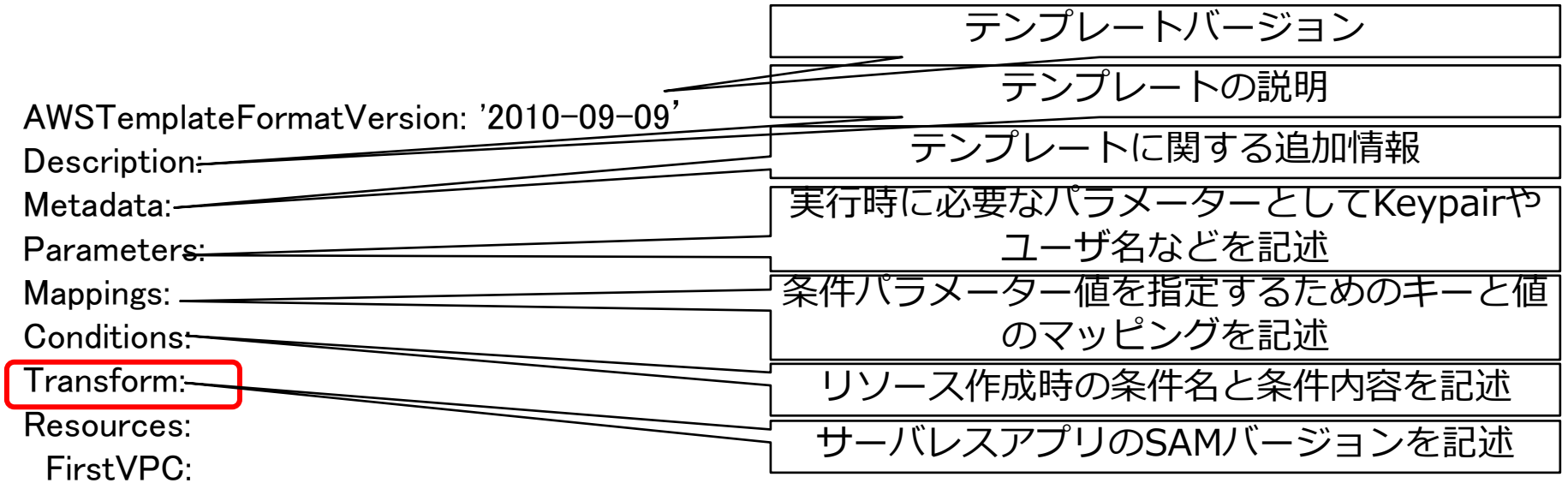



```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート


AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
FirstVPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
Tags:
- Key: Name
Value: FirstVPC
AttachGateway:
Type: AWS::EC2::VPCGatewayAttachment
Properties:
DependOn:
VpcId: !Ref FirstVPC
InternetGatewayId: !Ref InternetGateway
Outputs:
テンプレートバージョン
テンプレートの説明
テンプレートに関する追加情報
実行時に必要なパラメーターとしてKeypairやユーザ名などを記述
条件パラメーター値を指定するためのキーと値のマッピングを記述
リソース作成時の条件名と条件内容を記述

テンプレート




```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレート

```
AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
  FirstVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Name
        Value: FirstVPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      DependOn:
      VpcId: !Ref FirstVPC
      InternetGatewayId: !Ref InternetGateway
Outputs:
```

テンプレートバージョン

テンプレートの説明

テンプレートに関する追加情報

実行時に必要なパラメーターとしてKeypairやユーザ名などを記述

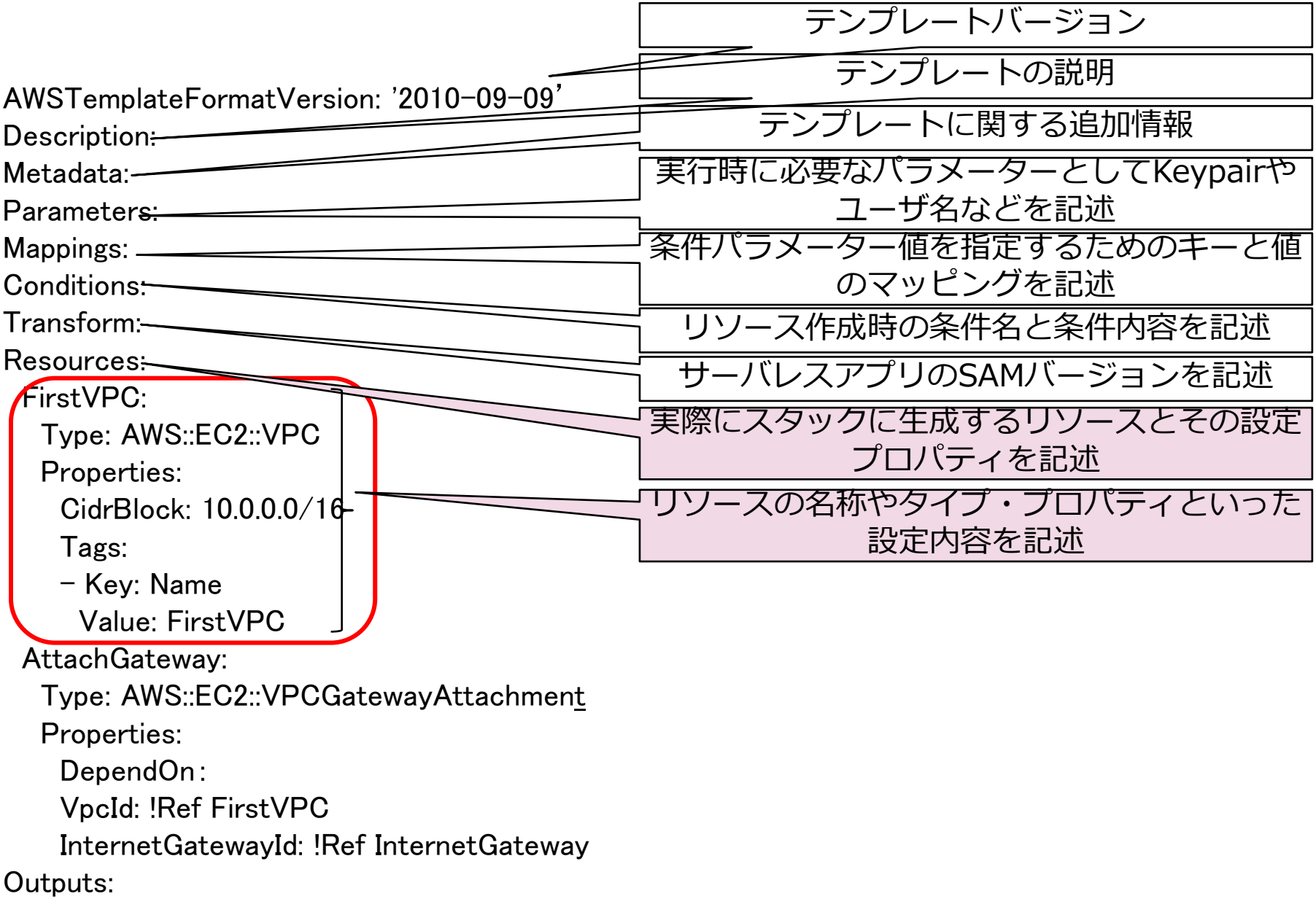
条件パラメーター値を指定するためのキーと値のマッピングを記述

リソース作成時の条件名と条件内容を記述

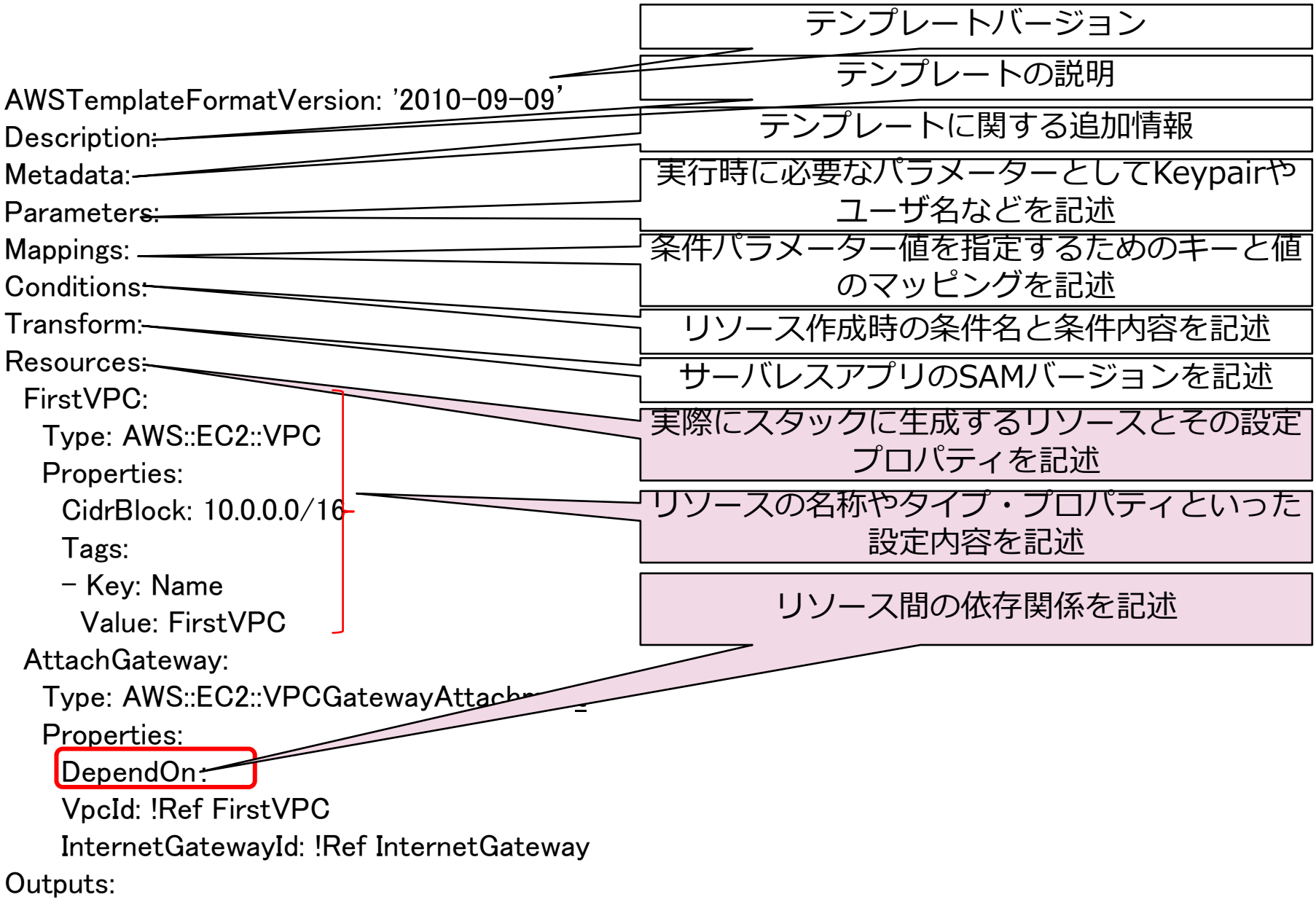
サーバレスアプリのSAMバージョンを記述

実際にスタックに生成するリソースとその設定プロパティを記述

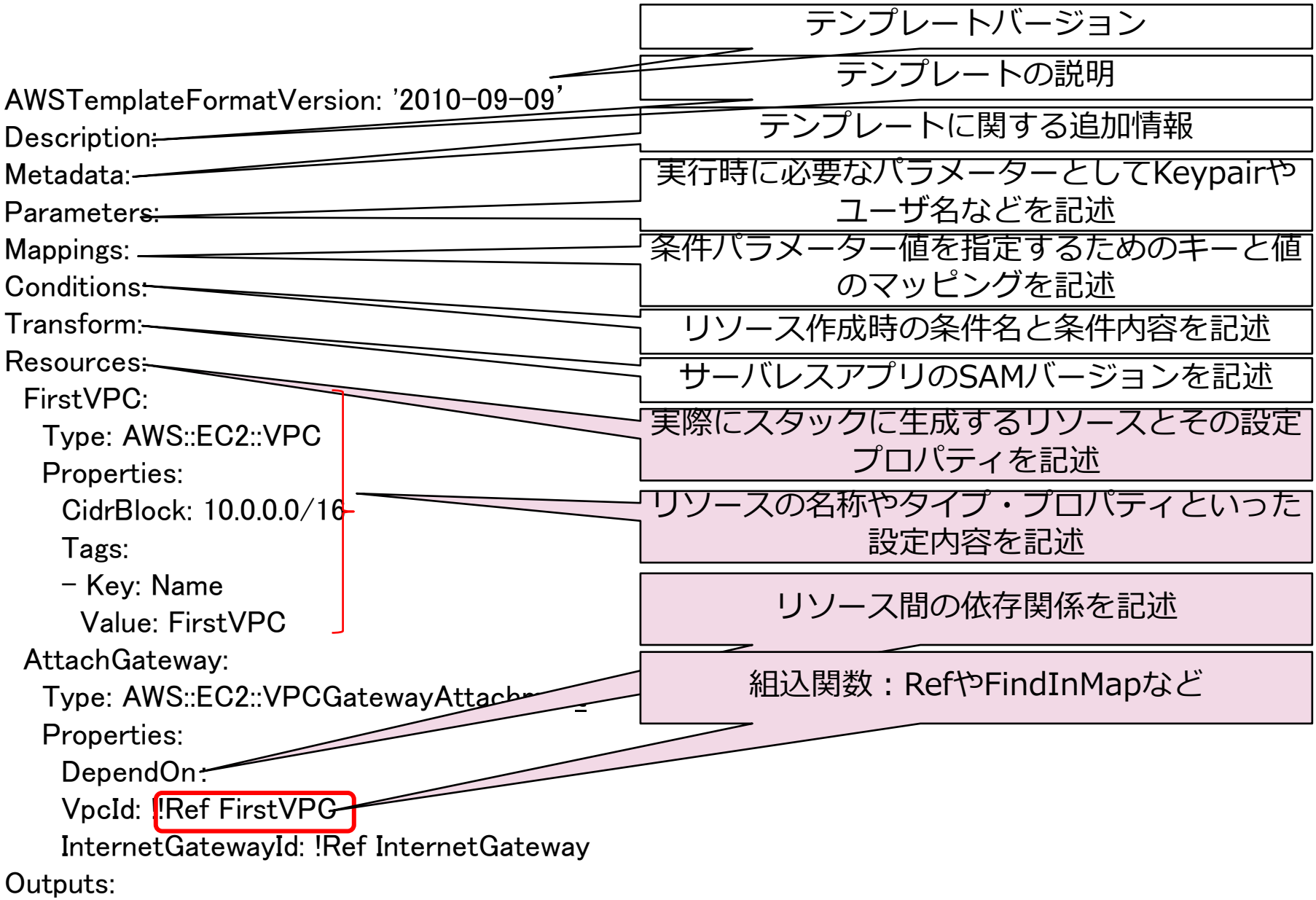
テンプレート


AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
FirstVPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
Tags:
- Key: Name
Value: FirstVPC
AttachGateway:
Type: AWS::EC2::VPCGatewayAttachment
Properties:
DependOn:
VpcId: !Ref FirstVPC
InternetGatewayId: !Ref InternetGateway
Outputs:
テンプレートバージョン
テンプレートの説明
テンプレートに関する追加情報
実行時に必要なパラメーターとしてKeypairやユーザ名などを記述
条件パラメーター値を指定するためのキーと値のマッピングを記述
リソース作成時の条件名と条件内容を記述
サーバレスアプリのSAMバージョンを記述
実際にスタックに生成するリソースとその設定プロパティを記述
リソースの名称やタイプ・プロパティといった設定内容を記述

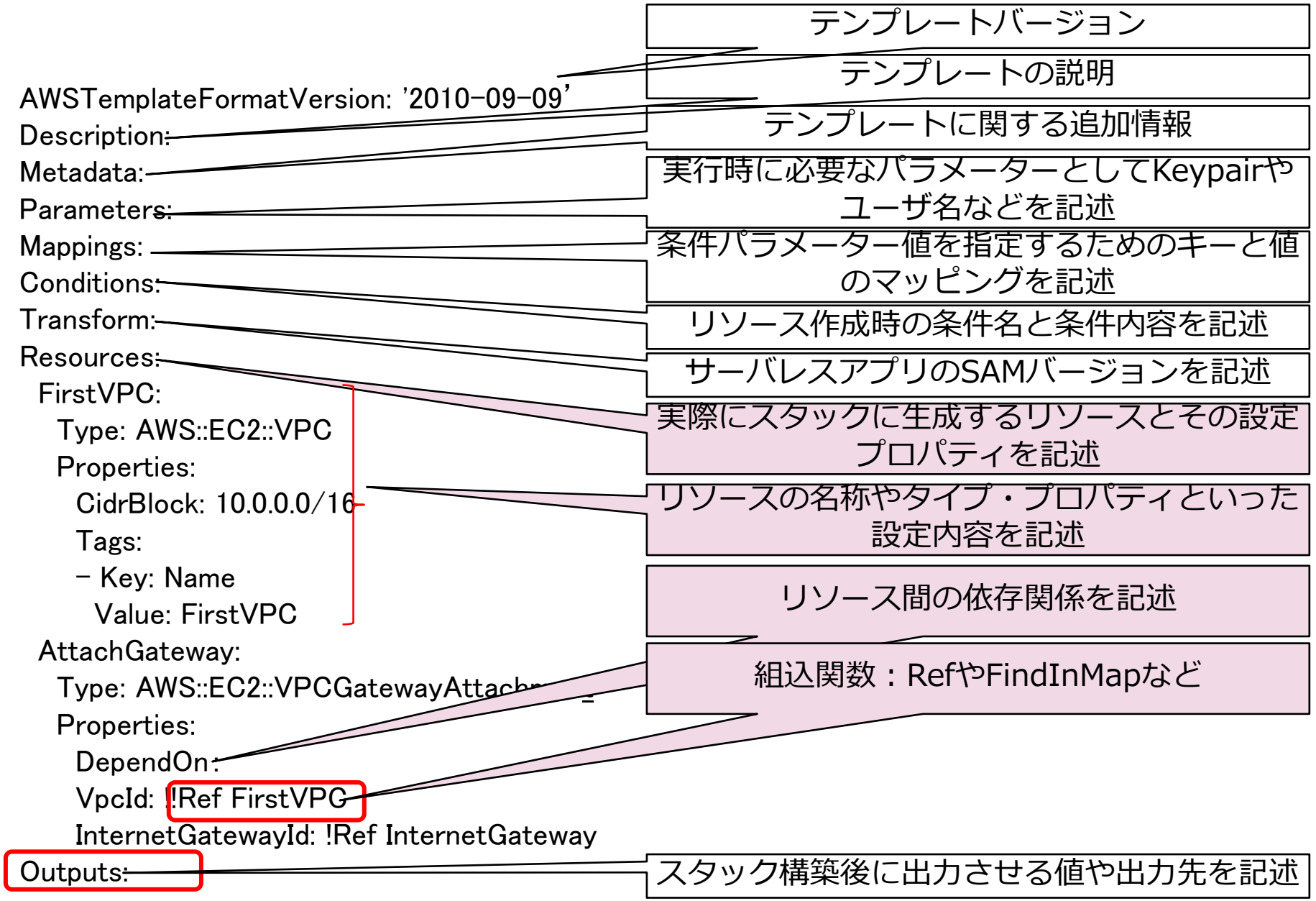
テンプレート


AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
FirstVPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
Tags:
- Key: Name
Value: FirstVPC
AttachGateway:
Type: AWS::EC2::VPCGatewayAttach
Properties:
DependOn:
VpcId: !Ref FirstVPC
InternetGatewayId: !Ref InternetGateway
Outputs:
テンプレートバージョン
テンプレートの説明
テンプレートに関する追加情報
実行時に必要なパラメーターとしてKeypairやユーザ名などを記述
条件パラメーター値を指定するためのキーと値のマッピングを記述
リソース作成時の条件名と条件内容を記述
サーバレスアプリのSAMバージョンを記述
実際にスタックに生成するリソースとその設定プロパティを記述
リソースの名称やタイプ・プロパティといった設定内容を記述
リソース間の依存関係を記述

テンプレート


AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
FirstVPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
Tags:
- Key: Name
Value: FirstVPC
AttachGateway:
Type: AWS::EC2::VPCGatewayAttach
Properties:
DependOn:
VpcId: !Ref FirstVPC
InternetGatewayId: !Ref InternetGateway
Outputs:
テンプレートバージョン
テンプレートの説明
テンプレートに関する追加情報
実行時に必要なパラメーターとしてKeypairやユーザ名などを記述
条件パラメーター値を指定するためのキーと値のマッピングを記述
リソース作成時の条件名と条件内容を記述
サーバレスアプリのSAMバージョンを記述
実際にスタックに生成するリソースとその設定プロパティを記述
リソースの名称やタイプ・プロパティといった設定内容を記述
リソース間の依存関係を記述
組込関数：RefやFindInMapなど

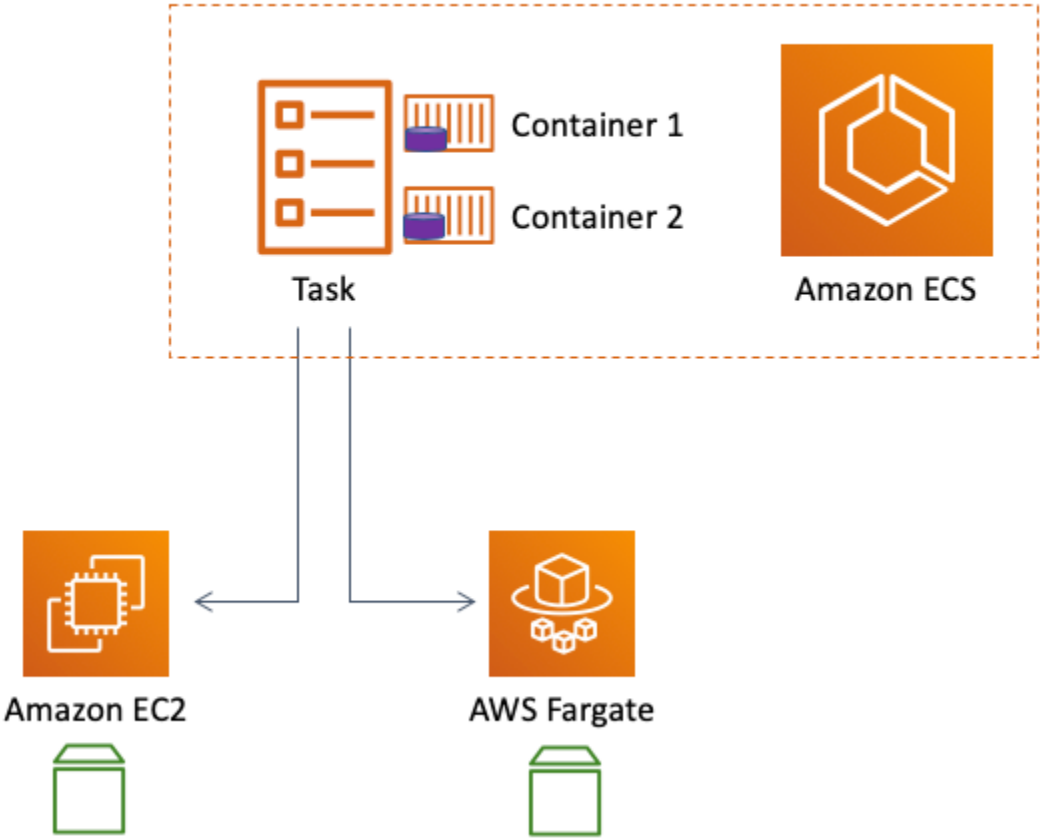
テンプレート


AWSTemplateFormatVersion: '2010-09-09'
Description:
Metadata:
Parameters:
Mappings:
Conditions:
Transform:
Resources:
FirstVPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
Tags:
- Key: Name
Value: FirstVPC
AttachGateway:
Type: AWS::EC2::VPCGatewayAttach
Properties:
DependOn:
VpcId: !Ref FirstVPC
InternetGatewayId: !Ref InternetGateway
Outputs:
テンプレートバージョン
テンプレートの説明
テンプレートに関する追加情報
実行時に必要なパラメーターとしてKeypairやユーザ名などを記述
条件パラメーター値を指定するためのキーと値のマッピングを記述
リソース作成時の条件名と条件内容を記述
サーバレスアプリのSAMバージョンを記述
実際にスタックに生成するリソースとその設定プロパティを記述
リソースの名称やタイプ・プロパティといった設定内容を記述
リソース間の依存関係を記述
組込関数：RefやFindInMapなど
スタック構築後に出力させる値や出力先を記述

Amazon ECSの出題範囲

Amazon ECSとは何か？

AWSでDockerを利用したコンテナーアプリケーション構築を可能にするサービス



Reference: https://aws.amazon.com/jp/blogs/containers/developers-guide-to-using-amazon-efs-with-amazon-ecs-and-aws-fargate-part-1/

Amazon ECSの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

ECSの選択	✓ Dockerを利用するためのサービスを選択するという問題が出題される。
起動タイプの選択	✓ シナリオに基づいて、要件にそったECSの設定時に適切な起動タイプの選択が問われる。
ECSの利用コスト	✓ ECSを利用した際に、料金が発生する要素が問われる。
タスクの定義	✓ Amazon ECSを利用した実装の際に、タスク定義の利用方法が出題される。
ECSの権限設定	✓ ECSのタスクが他のAWSリソースを利用する際の権限設定方法が問われる。

Amazon ECSの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

ALBの構成	✓ ECSのコンテナーをALBと構成する際の設定方法が問われる。
ECSの構成	✓ ECSを利用して複数のジョブを実行する際の基本的なECSの構成方法が問われる。

[Q] ECSの選択

B社ではAWSを利用したアプリケーション構築を実施する計画をしています。B社ではこれまでにDockerを利用したCI/CD環境により、Dockerアプリケーションを構築するような体制を整備しています。したがって、AWSでも同じようにDockerを利用する予定です。B社ではDockerのオープンソースの仕組みは利用していないため、今後も利用しない予定です。

この要件を満たすことができるソリューションを選択してください。

1) Amazon ECS
2) Amazon EKS
3) Amazon ECR
4) Amazon Fargate

Amazonのコンテナサービス

レジストリ	コントロールプレーン	データプレーン
コンテナエンジンに実行されるイメージが保管される場所 **Amazon ECR**	コンテナを管理するサービス **Amazon ECS** **Amazon EKS**	コンテナが実行される環境 **AWS Fargate**

Elastic Container Service (ECS)

Dockerコンテナをサポートする拡張性とパフォーマンスに優れたコンテナオーケストレーションサービス

- コンテナ化されたアプリをAWSにおいて簡単に実行およびスケールできる
- Fargateを利用することでコンテナのデプロイと管理にサーバーのプロビジョニングや管理は不要
- あらゆる種類のコンテナ化されたアプリケーションを簡単に作成できる。
- Dockerコンテナの数が数十であっても数万であっても数秒で簡単に起動
- ELB／ VPC／ IAM／ECR／CloudWatch／CloudFormation/CloudTrailなどのAWSサービスを利用可能
- VPCネットワークモードでTask毎にENIを自動割り当てSecurity GroupをTask毎に設定可能。VPC内の他のリソースへPrivate IPで通信が可能
- Fargate起動タイプとEC2起動タイプという2種類のモードがある

Amazon Elastic Kubernetes Service (EKS)

コンテナ化されたアプリケーションのデプロイ、管理、スケールをオープンソースのKubernetesを使って実行するサービス

- Kubernetesは自動デプロイ、スケーリング、アプリ・コンテナの運用自動化のために設計されたオープンソースのプラットフォーム
- Kubernetesのパートナーやコミュニティが作成した既存のプラグインやツールを使用可能
- マネージド型サービスでありコントロールプレーンの管理が不要
- ワーカーノードとマネージドコントロールプレーンとの間に、暗号化処理された安全な通信チャネルを自動的にセットアップする
- Kubernetes環境で管理されるアプリケーションとの完全な互換性がある

Elastic Container Registry (ECR)

フルマネージド型のレジストリサービスでDockerコンテナイメージを簡単に保存、管理、デプロイが可能

- ECSとDocker CLIに統合されており、開発から本稼働までのワークフローを簡略化する
- IAMによる強力な認証管理機構
- エンドポイントにアクセスできるならAWS外からでも利用可能
- ライフサイクルポリシーでイメージの自動クリーンアップできる
- VPCネットワークモードでタスク毎にENIを自動割り当てして、セ更にキュリティグループをタスク毎に設定可能

[Q]起動タイプの選択

あなたはソリューションアーキテクトとして、Dockerコンテナで複数のコンポーネントによって構成された新しいアプリケーションを構築することになりました。コンテナーの実行ではインスタンスタイプの選択、クラスタースケジューリングの管理などは実施しなくても良い方法が必要です。

どのようにDockerコンテナーを構成すればよいでしょうか？（2つ選択してください）

1）Amazon EKSでEC2起動タイプを使用する

2）Amazon ECSでEC2起動タイプを使用する

3）AWS Elastic BeansStalkによりDockerアプリケーションを構成する。

4）Amazon ECSでFargate起動タイプを使用する

5）コンテナイメージをAmazon ECRに配置する

起動タイプの選択

サーバーやクラスターの管理なしにコンテナを実行するECSに対応したコンピューティングエンジン

EC2起動モード

- ECSでEC2インスタンスを起動する
- コンテナアプリケーションを実行するインフラストラクチャに対して、サーバーレベルの詳細なコントロールを実行可能
- サーバークラスターを管理し、サーバーでのコンテナ配置をスケジュール可能
- サーバークラスターでのカスタマイズの幅広いオプションが利用できる

Fargate起動モード

- ECSで設置できる専用のコンピューティングエンジン
  ※2019年12月よりEKSにも対応
- EC2インスタンスのクラスターを管理する必要がない
- インスタンスタイプの選択、クラスタースケジューリングの管理、クラスター使用の最適化は不要
- CPU、メモリなどのアプリ要件を定義すると、必要なスケーリングやインフラはFargateが管理する
- 秒で数万個のコンテナを起動

[Q] ECSの利用コスト

あなたはソリューションアーキテクトとして、Dockerコンテナで複数のコンポーネントで構成された新しいアプリケーションを構築することになりました。Amazon ECSを利用したFargate起動タイプとEC2起動タイプのどちらを利用するのか検討しています。

Amazon ECSの課金方式として正しい内容を選択してください。

1) Fargate起動タイプはCPU数とメモリリソースに基づいて課金される。
2) EC2起動タイプはCPU数とメモリリソースに基づいて課金される。
3) Fargate起動タイプとEC2起動タイプはどちらもCPU数とメモリリソースに基づいて課金される。
4) EC2起動タイプは使用されたEC2インスタンスとEBSボリュームに基づいて課金されます。

ECSの利用コスト

ECSはEC2インスタンスの利用料金またはFargate向けの料金が発生する。

EC2起動モードの料金

- AWS リソース (EC2 インスタンス、EBS ボリュームなど) に対してのみ、料金が発生する。

Fargate起動モードの料金

- コンテナ化されたアプリケーションに必要な vCPU とメモリリソースに対する料金が発生する。
- コンテナイメージを取得した時点から Amazon ECS タスク* が終了するまでを対象として計算され、最も近い秒に切り上げられます。1 分の最低料金が適用される。

※ECRはリポジトリに保存するデータとインターネットに転送するデータの量に対して課金

[Q]タスク定義

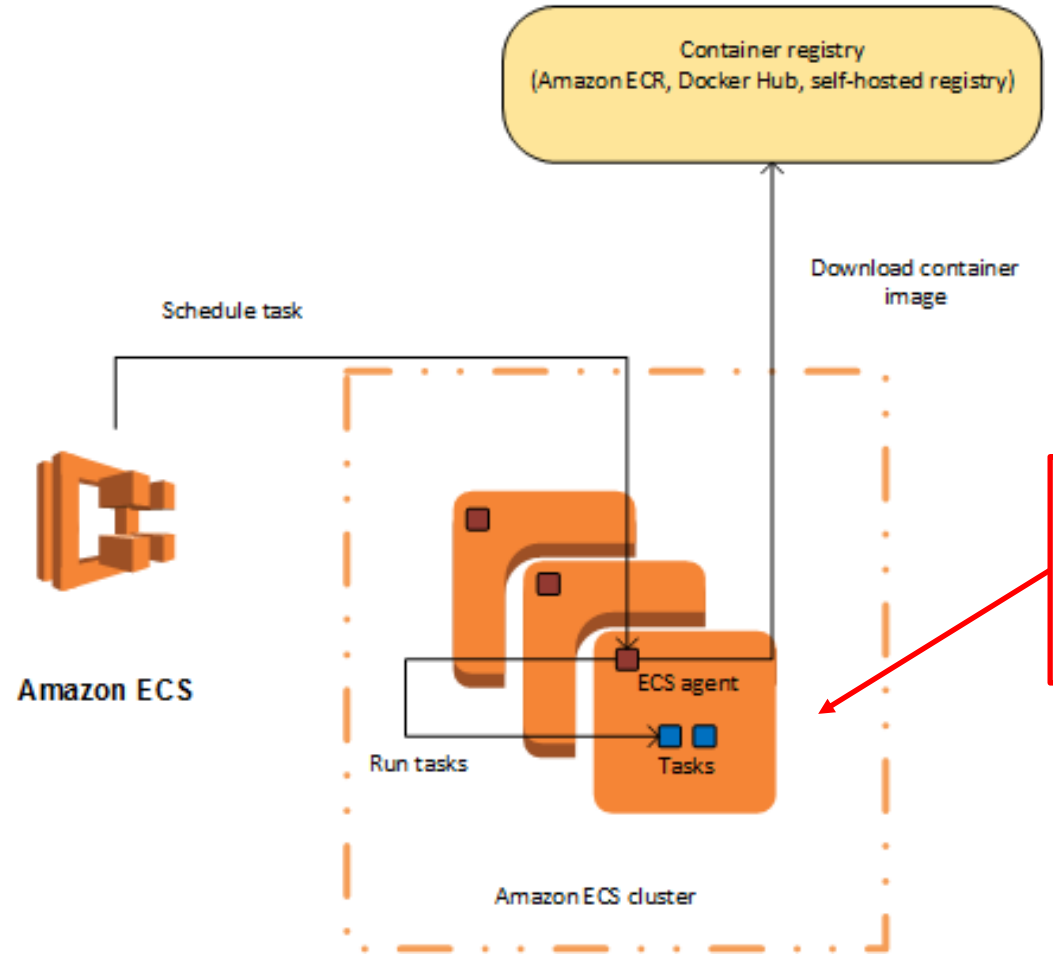
会社ではDockerアプリケーションを構築しています。既にデプロイしたECSクラスター上のDockerアプリケーションコンテナーに対して追加のアクセス許可を付与して、新しいタスクを追加したいと考えています。

この要件を満たすことができるAmazon ECSの設定方法はどれでしょうか？

1) 同じタスク定義でコンテナに個別のタスクロールを定義する。
2) ECSに設定されたEC2インスタンスにIAMロールを設定する。
3) ECSによる別のコンテナークラスターを立上げて、タスクロールを設定する。
4) 別のタスクロール用のコンテナーには別のタスク定義を作成する。

タスク定義

ECSでDockerコンテナを動かす際に、タスクを定義してコンテナを実行する。



タスク定義
タスクのリソース使用量や、タスクで実行するdockerコンテナの情報などを決める

Reference: https://docs.aws.amazon.com/ja_jp/AmazonECS/latest/developerguide/Welcome.html

[Q] ECSへの権限設定

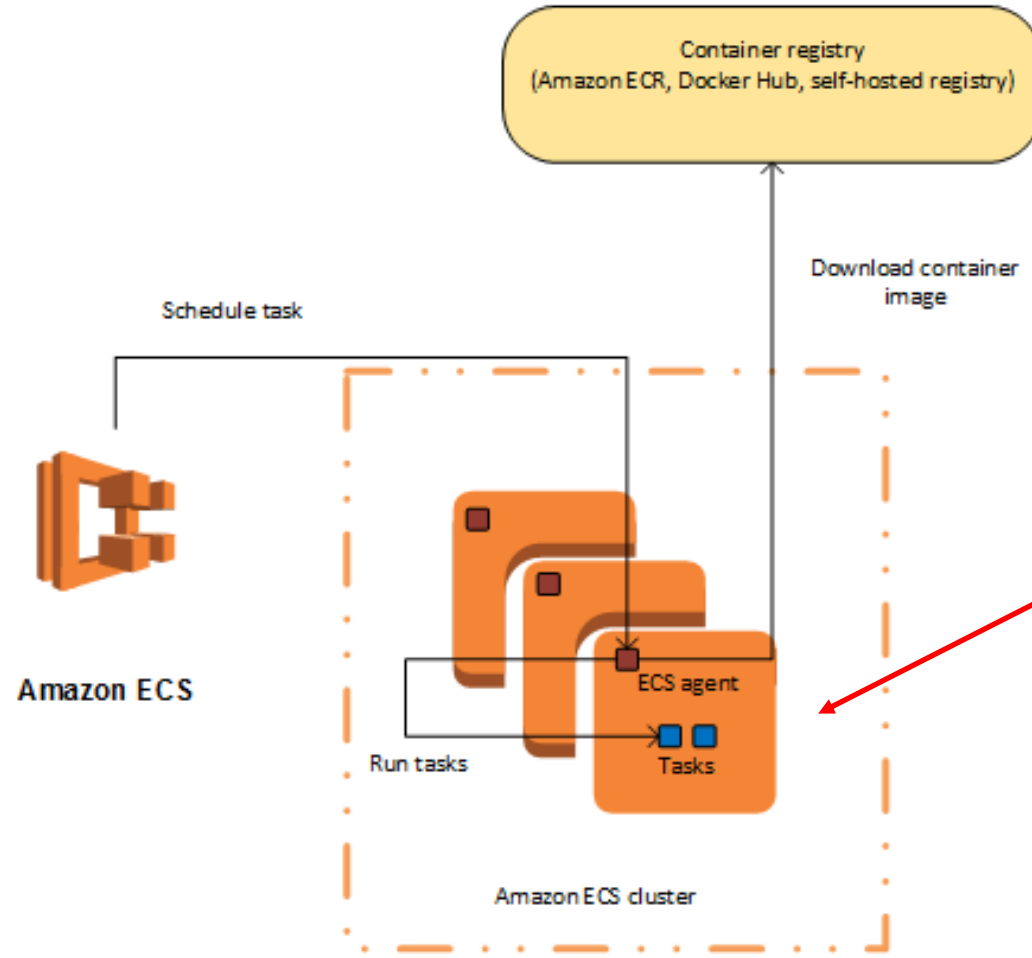
あなたはソリューションアーキテクトとして、Amazon ECSのEC2起動タイプを利用してアプリケーションを実装しています。このアプリケーションにはAmazon DynamoDBにデータを書き込むためのパーミッションが必要です。

特定のECSタスクにのみDynamoDBへのアクセス許可を割り当てるにはどうすればよいですか？

1) DynamoDBへのアクセス許可を持つIAMポリシーを作成し、それをコンテナーインスタンスにアタッチする。
2) DynamoDBへのアクセス許可を持つIAMポリシーを作成し、IAMロールに割りあて、そのIAMロールをECSに設定する。
3) DynamoDBへのアクセス許可を持つIAMポリシーを作成し、ECRクラスターに割り当てる。
4) DynamoDBへのアクセス許可を持つIAMポリシーを作成し、taskRoleArnパラメーターを使用してタスクに割り当てる。

ECSへの権限設定

タスクの実行に必要な権限をIAMポリシーによってタスク毎に割り当てる必要がある。



IAMポリシー
タスクに対してIAMポリシーを割り当てて、タスク毎にリソースへのアクセス権限を付与する。

Reference: https://docs.aws.amazon.com/ja_jp/AmazonECS/latest/developerguide/Welcome.html

[Q] ALBの構成

あなたはソリューションアーキテクトとして、Dockerを利用したWEBアプリケーションを構築しています。タスク定義で複数のコンテナークラスターにわけてタスクを実行するアプリケーションを実装していますが、それらを１つのALBによってトラフィック制御することが必要です。

最小限の労力でこれを達成するのに役立つ機能はどれですか？（2つ選択してください。）

1）ALB+動的ポートマッピング

2）ALB+パスルーティング

3）CLB+動的ポートマッピング

4）NLB+動的ポートマッピング

5）NLB+パスルーティング

ALBの構成

ALBを構成する際はパスルーティングや動的ポートマッピングによりECSと連携することが可能

パスルーティング構成

- ALBのパスベースルーティングを利用すると、URLに従ったターゲットグループへのルーティング可能
- ECSのパスルーティングにコンテナー指定して、コンテナーに対してパスルーティングを実装可能

動的ポートマッピング

- ECSで起動したEC2をターゲットグループに分ける際に複数のポートをALBのターゲットとして登録することが可能
- ECSのタスク定義で動的ポート番号を登録して、ポート番号に応じてトラフィック先を分散する。

[Q] ECSの構成

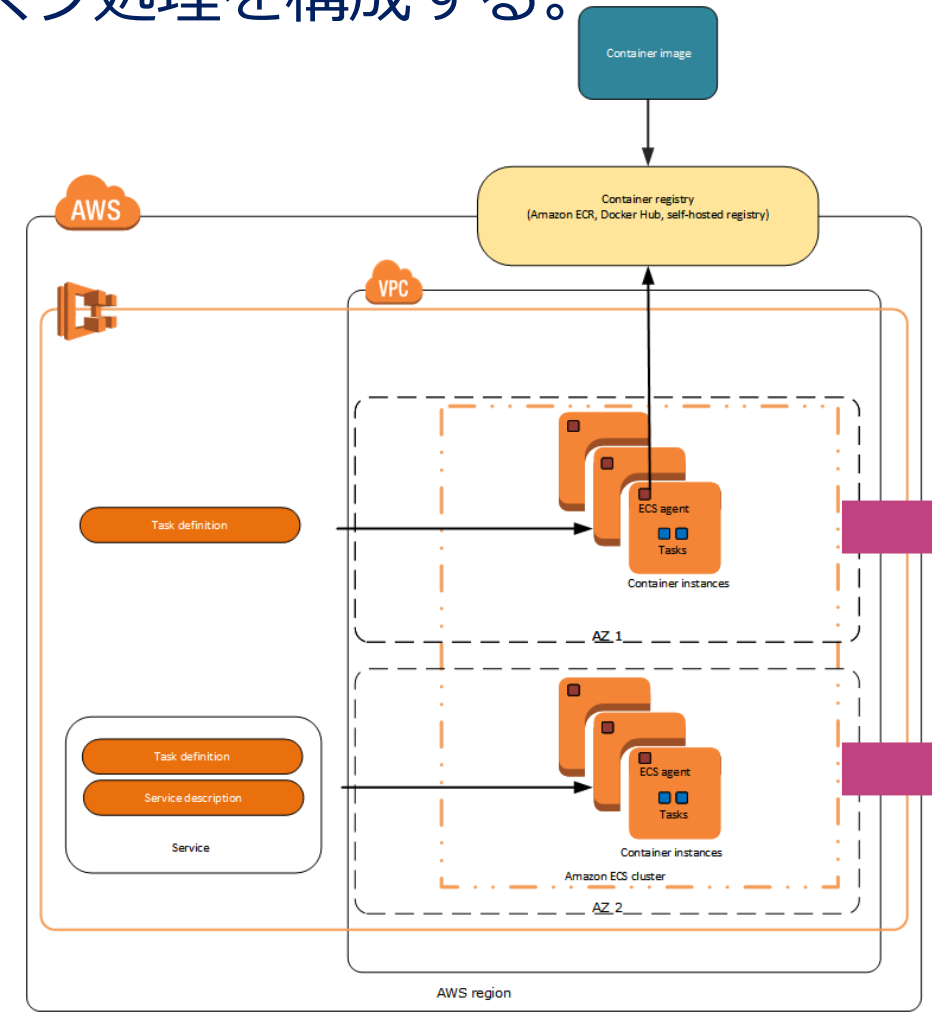
会社ではDockerアプリケーションを構築しています。既にデプロイしたECSクラスター上のDockerアプリケーションコンテナーに対して追加のアクセス許可を付与して、新しいタスクを追加したいと考えています。非常に重要なデータ処理ジョブと、いつでも実施可能なバッチジョブの両方を処理するために使用されます。

この要件を満たすために最も費用効果の高いオプションは次のうちどれですか？

1) 重要なデータ処理ジョブに対して、リザーブドEC2インスタンスを、重要ではないジョブにスポットEC2インスタンスを設定する。
2) 重要なデータ処理ジョブと、重要ではないジョブとに別々のタスク定義を割り当てることで、実行するコンテナーをわけて処理する。
3) Amazon SQSと連携して、重要なデータ処理ジョブに優先キューを設定して、重要ではないジョブは標準キューを設定する。
4) Lambdaと連携して、重要なデータ処理ジョブに優先ジョブを設定して、重要ではないジョブには標準ジョブを設定する。

ECSの構成

ECSのタスク定義で実行するジョブ内容を定義して、複数のタスク処理を構成する。



ミッションクリティカルな
バッチジョブを定義

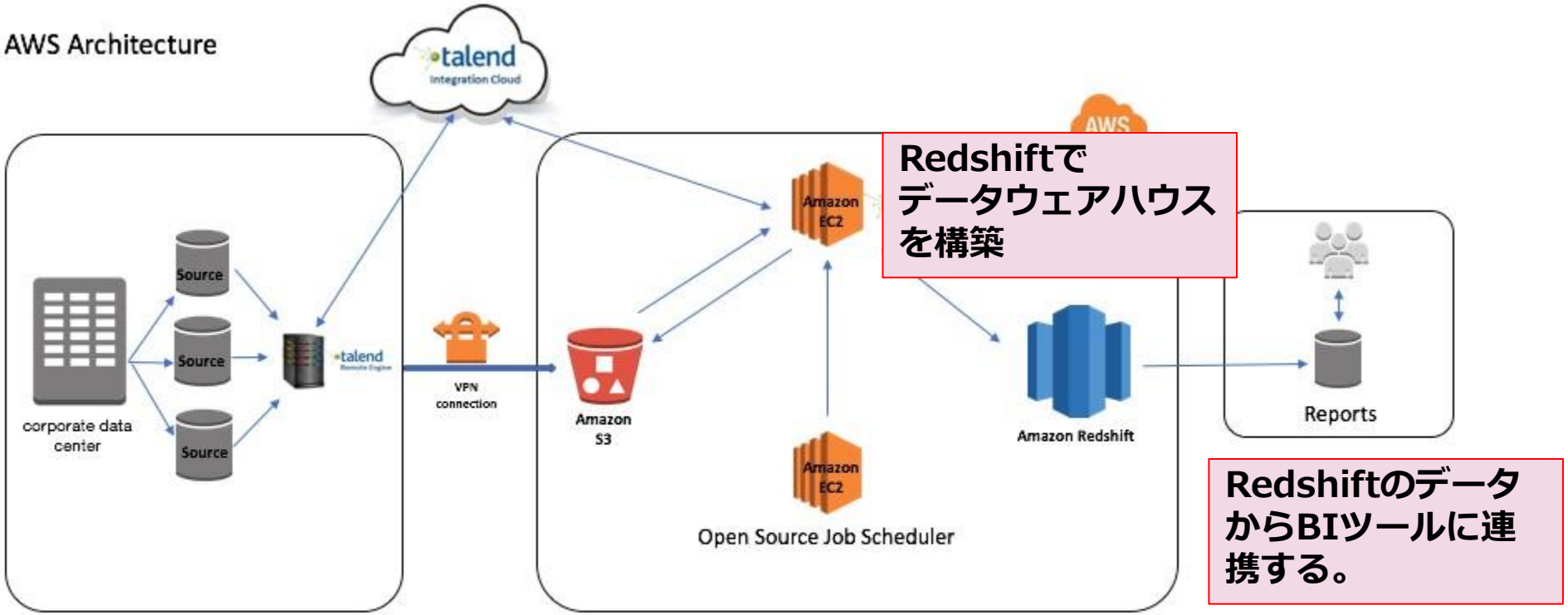
ミッションクリティカルでない
バッチジョブを定義

Reference: https://docs.aws.amazon.com/ja_jp/AmazonECS/latest/developerguide/launch_types.html

Redshiftの出題範囲

Redshiftとは何か？

AWS上にデータウェアハウスを構築することができるマネージド型サービス



Reference: https://aws.amazon.com/jp/blogs/database/using-amazon-redshift-for-fast-analytical-reports/

Redshiftの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Redshift の選択	✓ シナリオに基づいて、データベースの要件が提示されて Redshift を選択する問題が出題される。
Redshift の構成	✓ Redshiftを利用したAZやリージョンを利用した構成方法が問われる。
トラフィック制御	✓ トラフィック制御やモニタリング時にVPCを介したトラフィック制御を可能にする方法が問われる。
暗号化	✓ Redshiftにおける暗号化方法が問われる。
WLMの活用	✓ Redshift処理にキューを設定してワークロード管理する方法が問われる。

Redshiftの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Redshift Spectrum	✓ Redshiftを利用してS3ストレージに直接クエリを実行する方法が問われる。
リザーブドノードの利用	✓ Redshiftのノード利用をコスト最適化する方法が問われる。

[Q]Redshiftの選択

あなたの会社ではリレーショナルデータベースを利用した2つのデータ処理オペレーションを実装しています。1つ目の処理では、完了するまでに数時間かかるデータウェアハウスで複雑なクエリを実行します。 2つ目の処理では、顧客データ分析を実施して、ダッシュボードで可視化します。

これらの要件を満たすことができる最適なデータベースはどれでしょうか？

1) Redshiftを利用してデータウェアハウス処理を実行し、RDSを利用して顧客データ分析を実施する。
2) Redshiftを利用して両方のオペレーション処理を実装する。
3) RDSを利用して両方のオペレーション処理を実装する。
4) Auroraを利用して両方のオペレーション処理を実装する。

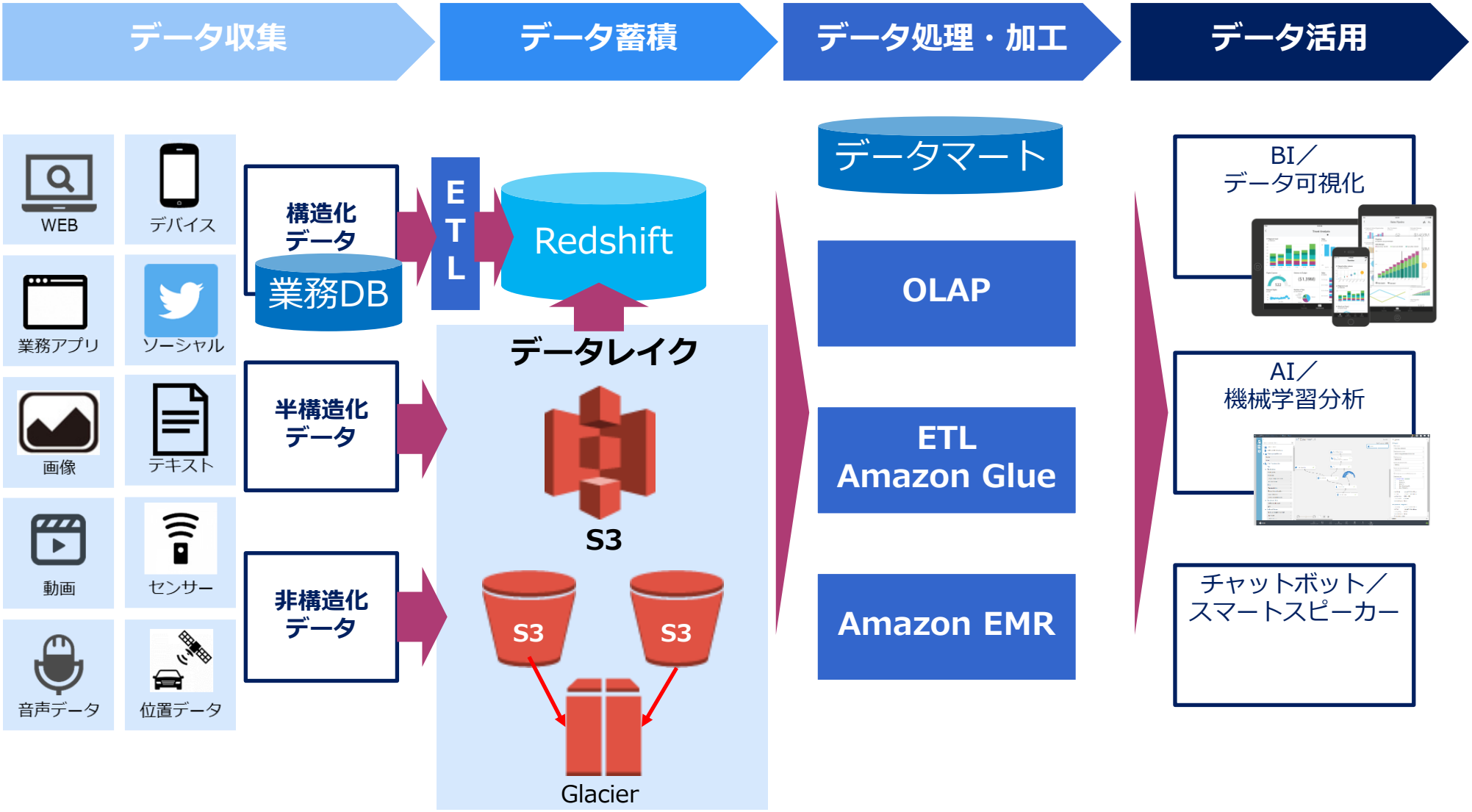
Redshift

高速でスケーラブルな費用対効果の高いマネージド型のDWH／データレイク分析サービス

- 数百ギガバイトのデータから開始して、ペタバイト以上まで拡張
- 1 テラバイトあたり年間 1,000 USD 以下で利用可能
- 自動ワークロード管理など自動テーブルメンテナンスなど多くのメンテナンスタスクやデータ配置が自動化されているフルマネージド型
- PostgreSQL互換の列指向データモデル
- 複数ノードをまとめたクラスター構成。単一AZで起動し、マルチAZ構成は不可
- RA3インスタンスで最大で他のクラウドデータウェアハウスの 3 倍に達するパフォーマンス
- AQUAによる分散キャッシュで、Redshift が他のクラウドデータウェアハウスに比べて最大 10 倍の速度で動作

データレイク

S3はデータレイクとしてデータ活用のハブとして利用できる



インスタンスタイプ

利用するデータサイズと増加予測に応じて２つのインスタンスタイプから選択

RA3インスタンス

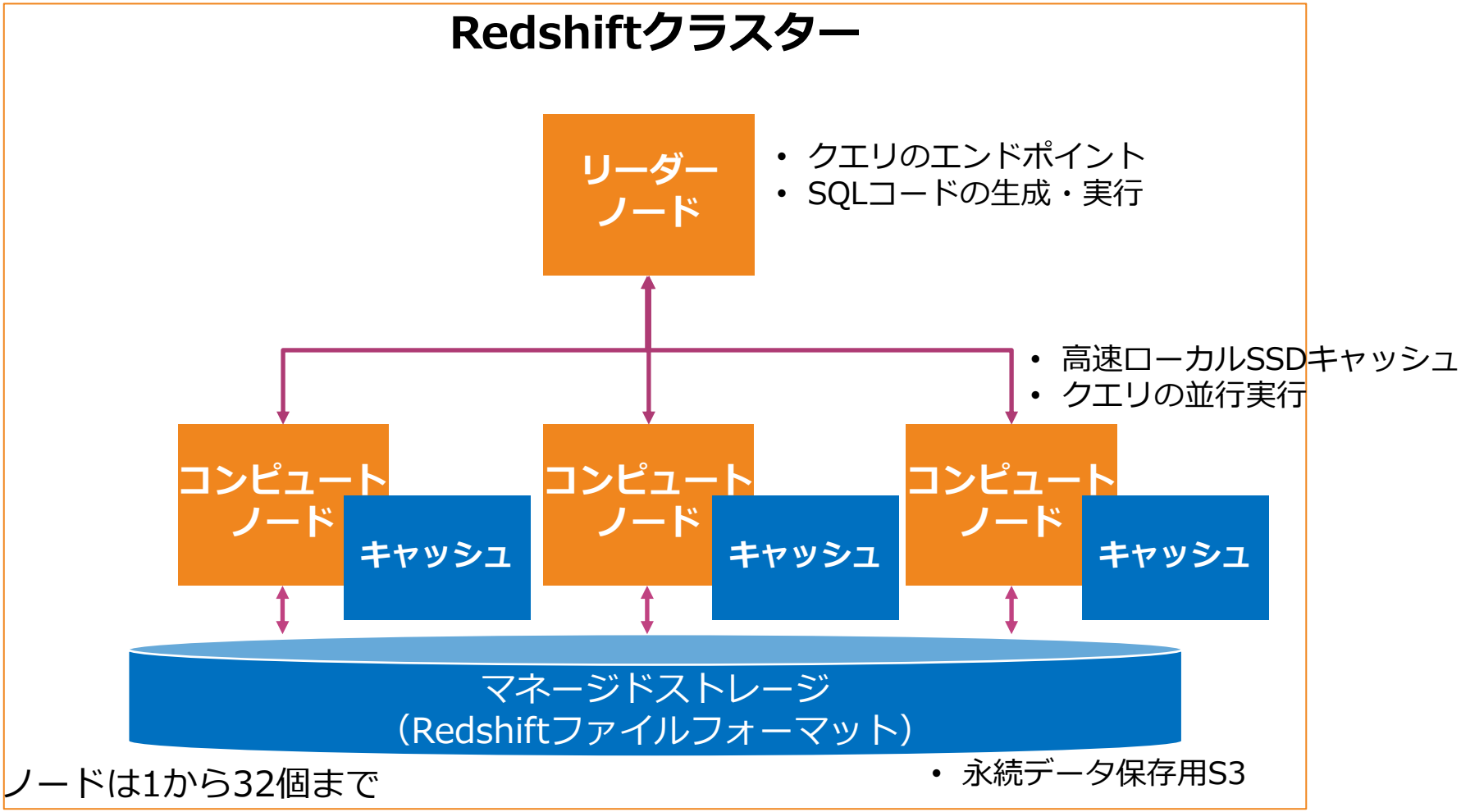
- コンピューティング性能とマネージドストレージのスケーリングと支払いを独立させることで、データウェアハウスを最適化
- データ量の増大が予想される場合は、RA3 ノードのご利用を推奨
- 最低2ノード必要
- 最安で$3.836/ノード/時

DC2インスタンス

- 固定ローカル SSD ストレージを使用してデータウェアハウス
- データのサイズ増加に対し、ノードを追加して、クラスターのストレージ容量を増強
- 未圧縮で 1 TB 未満のデータセットではDC2 ノードタイプの利用を推奨
- 最低1ノード必要
- 最安で$0.314/ノード/時

Redshiftの構成

クラスターというグループ単位で、複数ノードによってデータ処理を実行する構成



[Q] Redshiftの構成

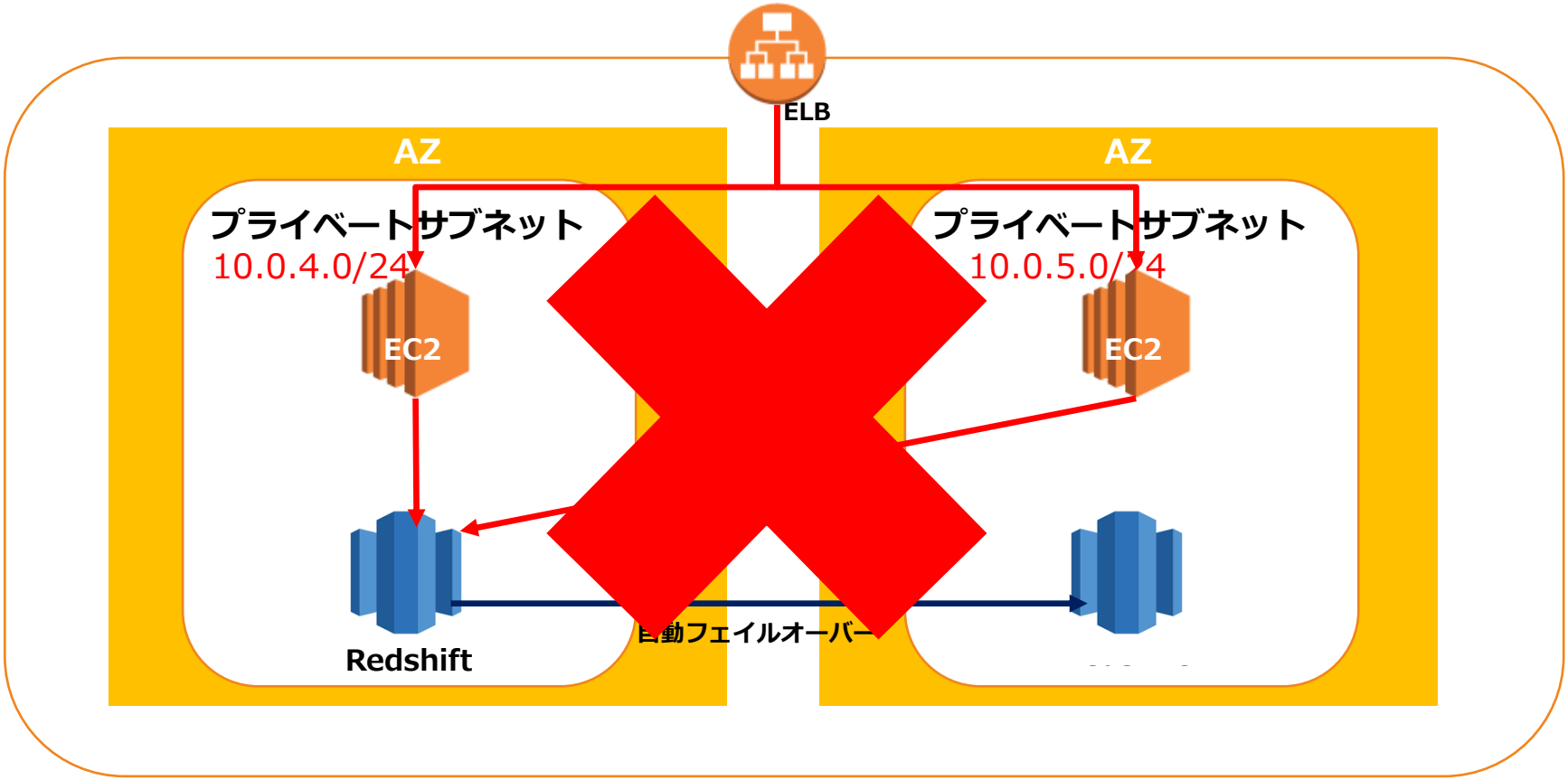
データ分析会社は、AWS上でデータウェアハウスとなるRedshiftクラスターを使用しています。 会社はビジネス継続性を確保するために、災害復旧計画を整備しました。あなたはソリューションアーキテクトとして、リージョンが停止した際のRedshiftの回復性を高める実装が求められています。

この要件を満たすための最良のアプローチは次のうちどれですか？

1) クロスリージョン構成を有効化する。
2) クロスリージョンのリードレプリカを起動して、フェールオーバー時にマスターにプロモートする。
3) クラスターのスナップショットを別のリージョンにコピーする。
4) グローバルクラスターを実施する。

Redshiftの構成

RDSとは異なり、マルチAZのフェールオーバー構成や、マルチリージョン対応はない。



- スナップショットを別リージョンにコピーして、万が一に備えることが必要

運用の自動化

自動的なメンテナンス機能と詳細のモニタリングによる簡易な運用が可能

CloudWatchとの連動	◘ 初期設定でCloudWatchメトリクス取得が自動で実施され、Redshiftコンソール内で確認可能
バックアップ	◘ 実行時間を設定して自動でバックアップを定期取得する ◘ スナップショットを手動で取得することも可能
自動メンテナンス	◘ パッチ適用も自動で実施 ◘ メンテナンスウィンドウでパッチ適用時間を指定可能

機械学習によるクエリ効率化

機械学習によってクエリ実行を調整し、効率的な自動実行を補助してくれる。

テーブルメンテナンスの自動化	◘ テーブルの分散スタイルの自動最適化 ◘ 統計情報の自動更新 ◘ データの再編成の自動実行
自動ワークロード管理	◘ 複数クエリの実行をワークロード管理で設定する際に、機械学習でクエリ実行の優先順位決めを自動化する
ショートアクセルレーション	◘ 機械学習アルゴリズムを使用して対象のクエリを １つ１つ分析し、クエリの実行時間を予測し、実行時間が短いクエリを、実行時間が長いクエリよりも優先して実行 ◘ WLMキューを削減可能
設定のレコメンデーション	◘ 自動でクラスターパフォーマンスなどを分析し、最適化やコスト削減に対するレコメンデーションを実施

[Q]トラフィック制御

あなたの会社ではリレーショナルデータベースを利用したデータ分析オペレーションを実装しています。そのためには、Redshiftを利用してデータウェアハウスを構成して、VPCエンドポイントにトラフィックを制御してAWSリソースと連携することが必要です。

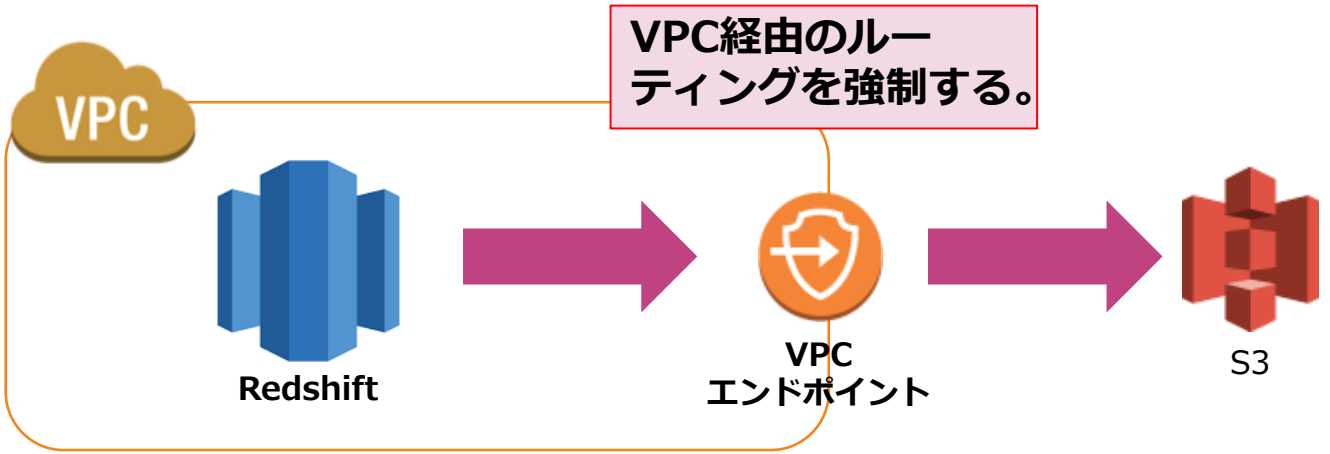
このシナリオで実装するのに最も適したソリューションは次のうちどれですか？

1) Amazon Redshiftの設置したサブネットのネットワークACLにおいて、 VPCエンドポイントへのルートを許可する。
2) Amazon Redshiftが利用するゲートウェイのルートテーブルにおいてVPCエンドポイントへのルートを設定する。
3) Amazon Redshiftの拡張VPCルーティングを使用する。
4) Amazon Redshiftにセキュリティグループを設定して、VPCエンドポイントへのルートを許可する。

トラフィック制御

拡張VPCルーティングによってVPCにトラフィックを強制しつつ、モニタリングが可能

拡張VPCルーティング	◘ Amazon Redshift はクラスターとデータリポジトリ間のすべての COPY と UNLOAD トラフィックが Amazon VPC を経由するよう強制する。 ◘ VPC フローログを使って COPY と UNLOAD トラフィックを監視する ◘ 有効にすることでVPCにて設定しているルートテーブルに従った通信を行う



[Q]暗号化

ある企業はRedshiftクラスターを使用してデータウェアハウスを構築しています。あなたは運用担当者として内部のセキュリティチームから、Redshiftデータベースのデータを確実に暗号化するよう求められました。

この要件を満たす最適な暗号化方法を選択してください。（2つ選択してください。）

1) SSL/TSLを利用して暗号化を実施する。
2) AWS KMSを利用して暗号化を実施する。
3) Amazon Redshift と HSM との間で信頼された接続を設定する
4) CSEを利用して暗号化を実施する。
5) Redshift Authによるキー作成を実施する。

暗号化の実施

RedshiftにおいてもKMSやACMを利用した暗号化を実施可能

保存データの暗号化	◘ 他のデータベースと同様にAWS KMSを利用して暗号化 ◘ 暗号化対象はディスク内に保存されているデータとスナップショット
通信の暗号化	◘ Amazon Redshift クラスターと SQL クライアントの間の送信時のデータを JDBC/ODBC 経由で暗号化する ◘ ACMによる証明書に基づいてSSL通信を実施

[Q] WLMの活用

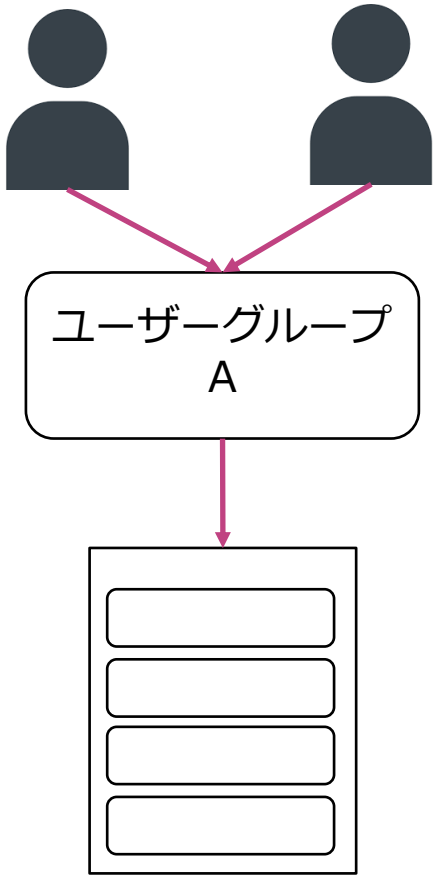
ある企業では、大規模なデータセットに対する複雑なクエリを処理するオンライン分析処理（OLAP）アプリケーションにRedshiftを使用しています。 これらのクエリ処理を実施する際に、照会内容をキューに経路指定する方法を定義しなければならないという要件があります。

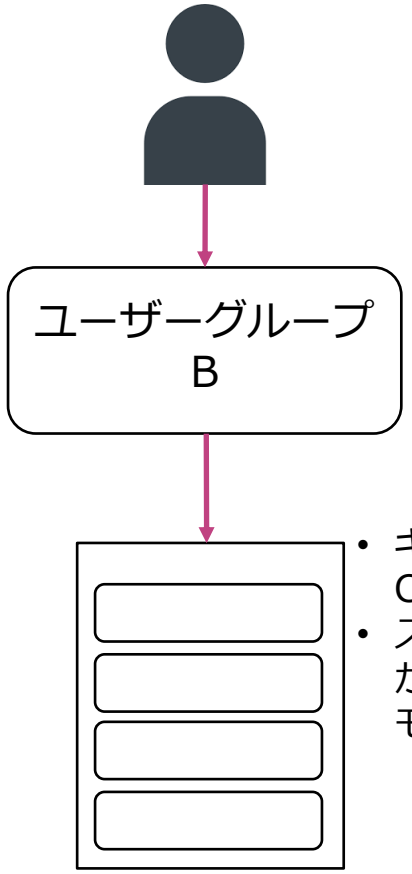
次のどのサービスによってこの要件を満たすことができますか。

1) Redshiftの拡張VPCルーティングでキューの経路を指定する。
2) RedshiftのWLM(Work Load Management)でキューの経路を指定する。
3) RedshiftのDLMを利用してキューの経路を指定する。
4) RedshiftにSQSを連携して、キューの経路を指定する。

ワークロード管理（WLM）の活用

ワークロードに応じて複数のキューを設定し、クエリ割り当てルールに基づいてキューを設定して、優先順位を設定可能





- キューにスロットを設定し、CPUとメモリの割り当てる
- スロットを増やすと並列度が向上するが、割り当てメモリが減少

ロングクエリ用キュー

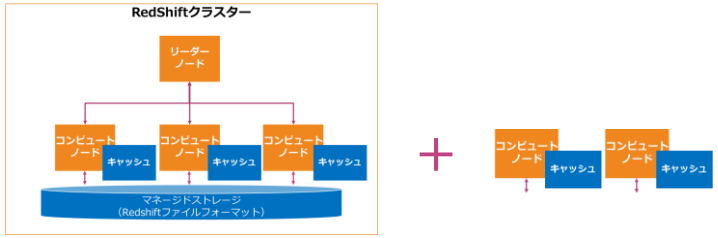
ショートクエリ用キュー

スケーリング

RedShiftはノードのタイプ変更・追加とクラスターの追加によってスケーリング可能

ノードの追加

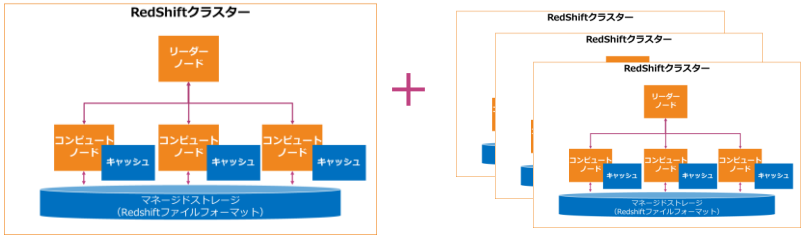
コンピューティングノードを追加することでパフォーマンスを向上



クラスターの追加

Concurrency Scalingにより急な同時実行リクエストに対応するために 一時的なクラスタを自動的に数秒で追加し、 一貫して高速なパフォーマンスを発揮

（追加クラスターは1~10）



[Q] Redshift Spectrum

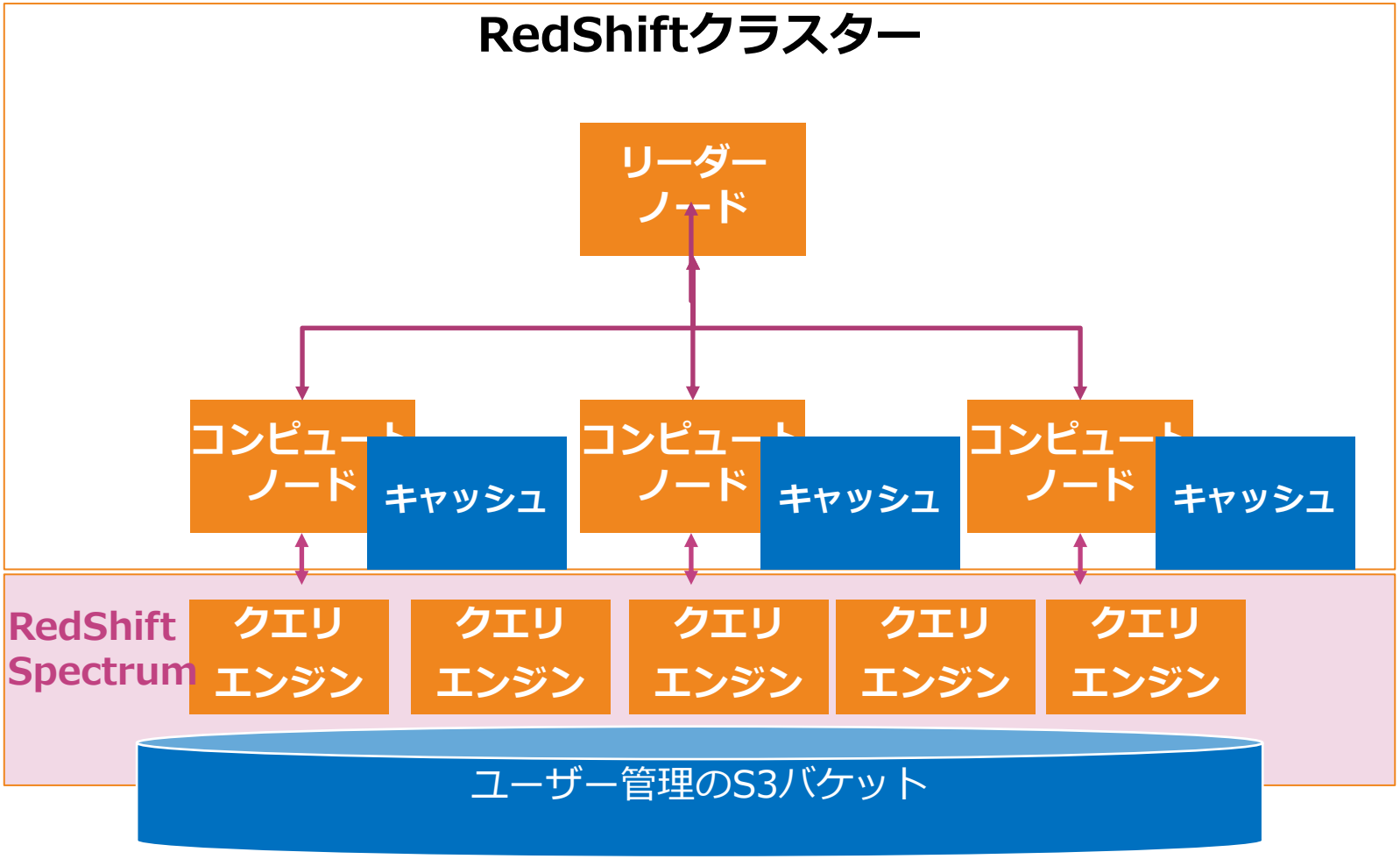
IoTソリューションを提供するビッグデータ分析企業は、AWS上に車両データを解析するためのデータ解析ソリューションを構築しています。車両データは大量に送信されるため、Kinesis Data StreamsからFirehoseを介してS3にデータを保存します。このS3データに直接クエリを実行して、大規模で複雑なクエリを実行することが必要となります。

次の中で、どのソリューションが最適でしょうか？

1) Athena
2) S3 Select
3) QuickSight
4) Redshift Spectrum

RedShift Spectrum

RedShift Spectrumにより、ユーザーが管理するS3バケットに対して直接データ解析を実行可能



データ連携（To Redshift）

Redshiftへとデータを移動させることで、DWHとしての解析基盤を集約化することが重要

S3	◘ 最も頻繁に使われるデータ連携先であり、S3からデータを取得してRedshiftで解析することも可能であるし、S3内部のデータ解析を直接実行することも可能
Kinesis	◘ Kinesis data Firehoseを利用してストリーミングデータの格納先としてRedshiftを指定してデータを保存して、解析に利用することが可能
RDS	◘ RDSとの直接接続はないが、AWS Data PipelineやDMSを利用してデータ移行を実施可能
DynamoDB	◘ DynamoDBからRedshiftにデータコピーを実行可能
Amazon EMR	◘ EMRからRedshiftにデータコピーを実行可能

データ連携（From Redshift）

RedshiftからはQuickSightを利用したデータ可視化に加え、S3へとデータ抽出も可能

Amazon QuickSight	◘ Redshiftに接続して、データの可視化を実施可能
S3	◘ UNLOADコマンドを実行することで、RedshiftからS3へとデータを抽出することが可能
Amazon Machine Learning	◘ RedShiftを機械学習の学習データとして設定して利用可能
RDS	◘ 直接に連携はできないが、PostgreSQLの機能を利用してデータをRedShiftからRDSと連携可能

[Q]リザーブドノードの利用

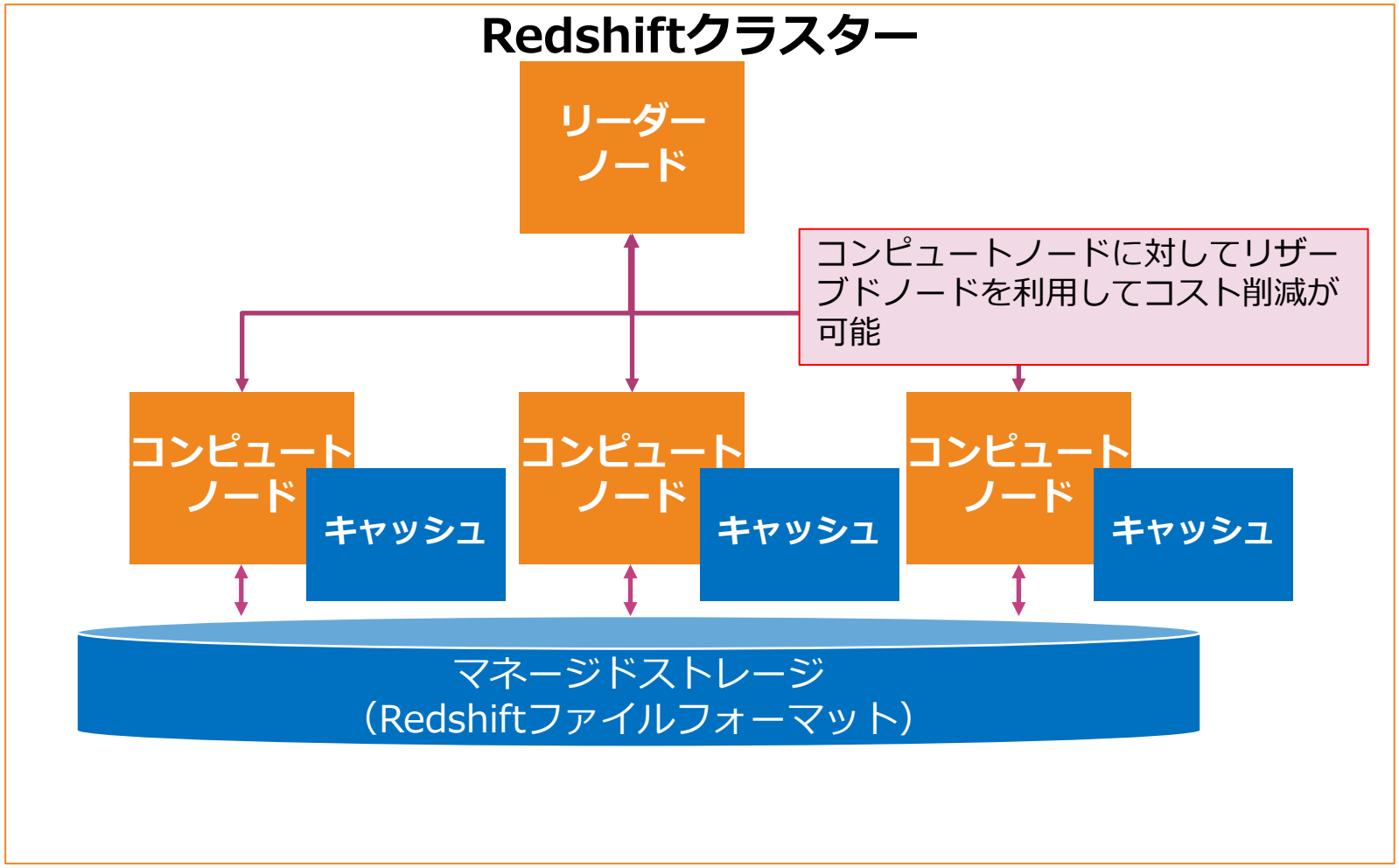
B社では６つのノードを持つAmazon Redshiftクラスターを利用したデータウェアハウスをAWS上で実行しています。このシステムを利用して様々な業務解析を日々行っており、このシステムは今後１年は利用される予定です

このRedshift構成のコスト削減が可能な、インスタンス構成を選択してください。

1) コンピュートノードに対してリザーブドノードを購入して前払い割引を適用する
2) リーダーノードに対してリザーブドノードを購入して前払い割引を適用する
3) コンピュートノードに対してスポットノードを購入して前払い割引を適用する
4) リーダーノードに対してスポットノードを購入して前払い割引を適用する

リザーブドノードの利用

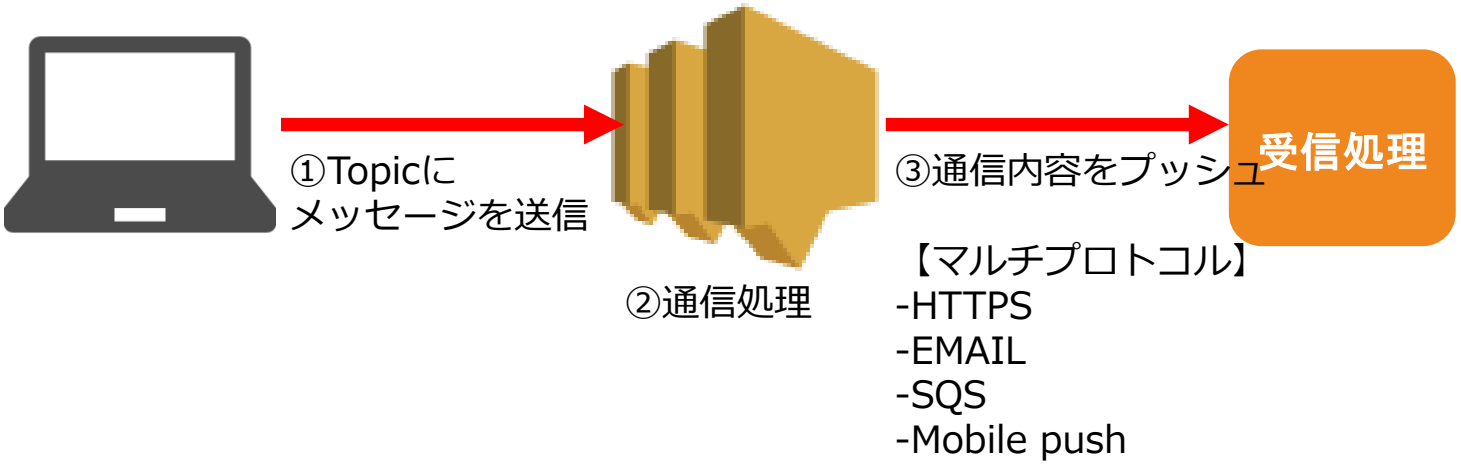
クラスターというグループ単位で、複数ノードによってデータ処理を実行する構成



SNSの出題範囲

SNSとは何か？

Amazon SNSはフルマネージド型のプッシュ型通知サービスで他のサービスとの非同期通信を可能にする



SNSの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

SNSの選択	✓ シナリオに基づいて、要件を達成するためにAmazon SNSを選択する問題が出題される。
SNSの特徴	✓ SNSによって達成できる機能や性能などの特徴に関する問題が出題される。
SNSの構成	✓ SNSを利用したソリューション構築に関する質問が出題される。

[Q]SNSの選択

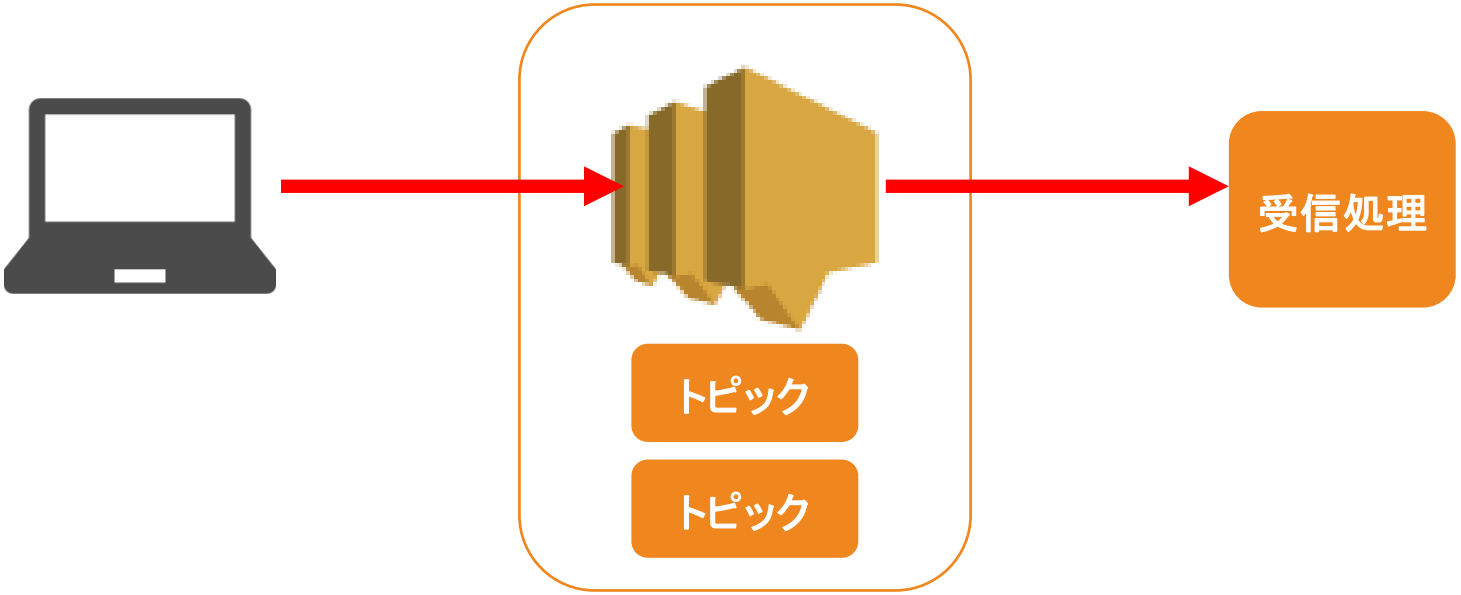
大手企業はEC2インスタンスを利用してWEBアプリケーションを構築しています。このアプリケーションは、データ処理を実行するためにAWS Lambda関数を非同期的に呼び出す必要があります。EC２とLambda関数とは分離して実行する必要があるため、コンポーネント間を連携をプッシュ通知によって実行する仕組みが不可欠となります。

アプリケーション間を分離するために使用できるサービスはどれですか？

1) Amazon SNS
2) Amazon SQS
3) Amazon SES
4) Amazon MQ

Amazon Simple Notification Service(SNS)

送信側がトピックを作成して受信側をポリシー指定することで
制御された非同期通信を実現する



SNSとSQS

SNSとSQSはその処理方式が異なるため利用シーンに応じて使い分ける

SNS	SQS
• メッセージは永続ではない • プッシュ型配信方式 • プロデューサーが発行 • コンシューマーがサブスクライブ	• メッセージは永続性あり • ポーリング型配信方式 • プロデューサーが送受信 • コンシューマーが送受信

[Q]SNSの特徴

大手企業はEC2インスタンスを利用してWEBアプリケーションを構築しています。このアプリケーションは他のアプリケーションに通知することで、バックエンド処理を実行する機能が必要です。あなたはソリューションアーキテクトとして、通知方式を検討しています。

次の中でAmazon SNSでサポートされている通知プロトコルはどれでしょうか？（３つ選択してください）

1) SSH
2) FTP
3) SMS
4) HTTPS
5) Email
6) MQ

SNSの特徴

AWSの様々なサービスと連携して通知可能で、疎結合アーキテクチャに利用できる

- 単一発行メッセージ
- メッセージ通信順番は保証されない
- 取り消し不可
- 配信ポリシーによる再試行を実施
- メッセージサイズは最大256KB

SNSの特徴

SNSでは以下のようにHTTP／HTTPS／JSON形式のメッセージを利用している。

- HTTP/HTTPS ヘッダー
- HTTP/HTTPS 受信登録の確認の JSON 形式
- HTTP/HTTPS 通知の JSON 形式
- HTTP/HTTPS 受信登録の解除の JSON 形式
- SetSubscriptionAttributes 配信ポリシー JSON 形式
- SetTopicAttributes 配信ポリシー JSON 形式

SNS連携

AWSの様々なサービスと連携して通知可能で、疎結合アーキテクチャに利用できる

- Amazon CloudWatch：Billing Alertの通知
- Amazon SES ：Bounce/Complaintのフィードバック通知
- Amazon SQS：SNS通知によりキューを配信して、処理を実行
- Amazon S3 ：ファイルがアップロードされた時の通知
- Amazon Elastic Transcoder ：動画変換処理完了/失敗時の通知
- AWS Lambda：SNSをトリガーとして処理を起動する。

[Q] SNSの構成

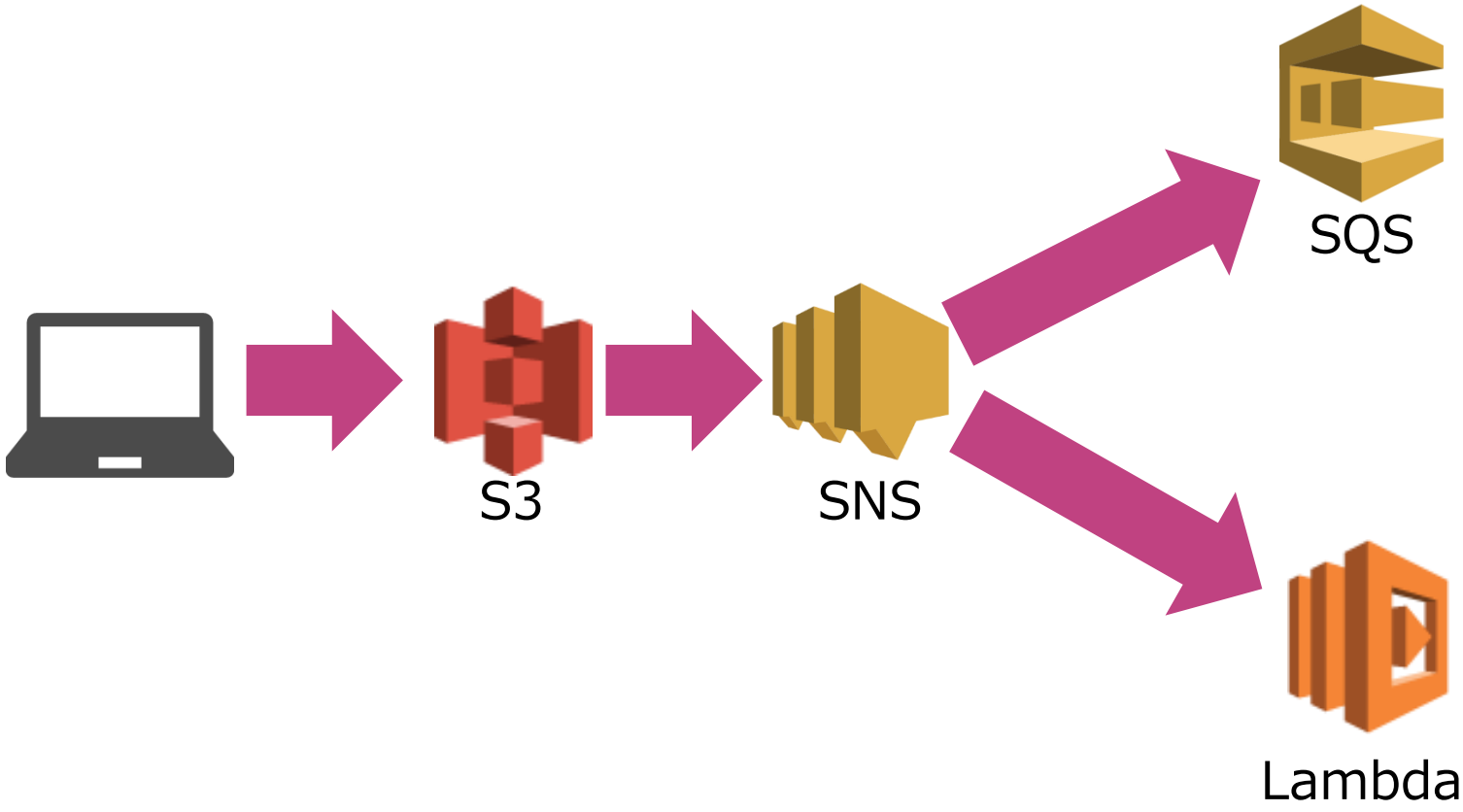
大手企業はEC2インスタンスを利用してWEBアプリケーションを構築しています。このアプリケーションは他のアプリケーションに通知することで、バックエンド処理を実行する機能が必要です。この通知は、バックエンド処理を実行するLambda関数によって処理されますが、ピーク時に1秒あたり約5000リクエストに達しています。あなたがテストを実施したところ、Lambda関数の処理の一部が実行されない問題が発生していることが分かりました。

この問題を解決策するために最適なソリューションを選択してください。

1) Amazon SNSが通知上限に達したため、制限を引き上げる必要がある。
2) Amazon SNSメッセージ配信がLambdaのアカウント同時実行クォータを超えたため制限を引き上げる必要がある。
3) Amazon SNSからLambda関数に連携する際のIAMポリシーが不適切です。
4) Lambda関数側でAmazon SNSのサブスクリプションを認証する必要がある。

Lambda関数との連携

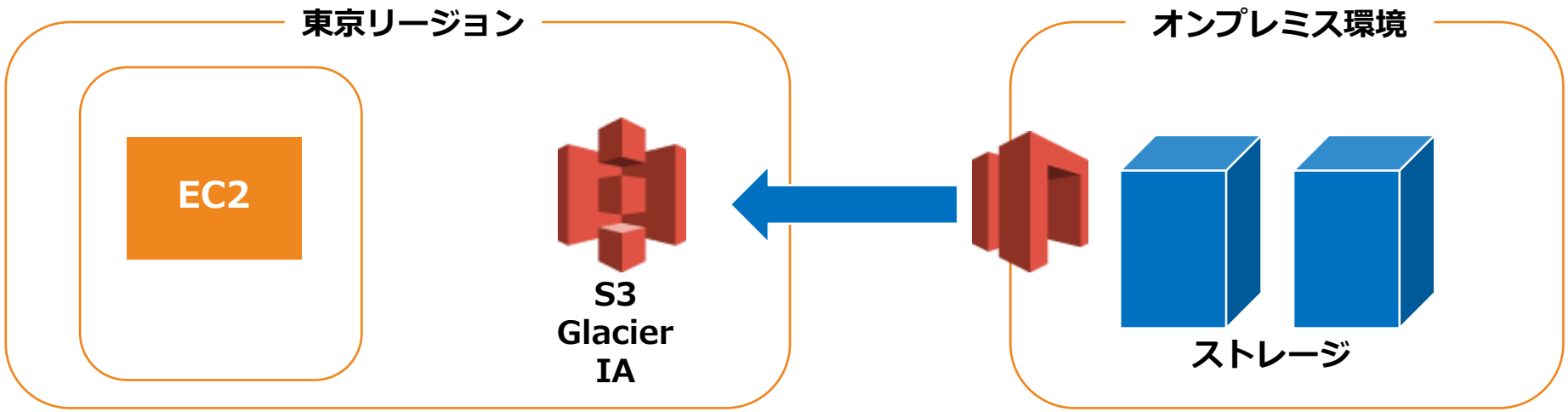
SNSのプッシュ通知からSQSやLambda関数などを実行する構成が可能



AWS Storage Gateway の出題範囲

AWS Storage Gatewayとは何か？

オンプレミス環境のストレージをAmazon S3に接続して拡張するサービス



AWS Storage Gatewayの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

ストレーレージゲートウェイの選択	✓ オンプレミス環境のストレージを拡張して、ハイブリッド構成を実現する方法を選択する問題が出題される。
ストレージゲートウェイタイプの選択	✓ オンプレミス環境のストレージを拡張する方法として、ストレージゲートウェイのタイプを選択する問題が出題される。

[Q]ストレーレージゲートウェイの選択

あなたはソリューションアーキテクトとして、AWS上でクラウドファンディングのアプリケーションを構築しています。このアプリケーションは、様々な社会貢献プロジェクトのためにお金を集めることを可能にします。 利用者に安心してもらうためには、このアプリケーションの集金機能が安全であることが大前提となります。セキュリティ要件としては、別途設定することなく、デフォルトで保存データを暗号化しているサービスを利用することになりました。

この要件を満たすサービスを選択してください（2つ選択してください。）

1) AWS Storage Gateway
2) Amazon Glacier
3) AWS RDS
4) AWS Lambda
5) Amazon ECS

AWS Storage Gatewayの利点

AWSが有する機能や性能を活用できることが大きな利点

- 業界標準のプロトコルを利用したシームレスな統合
- キャッシュを活用した低レイテンシーアクセスが可能
- AWSストレージサービスの堅牢性・低コスト・拡張性
- 効率的なデータ転送
- AWSのモニタリング・管理・セキュリティと統合されており、
  自動的に暗号化を実施している。

AWS Storage Gatewayの用途

データ移転や保存などAWSストレージを利用したい場合に用いる

- ビッグデータ処理／クラウドバースティング／システム移行のためにデータをAWSストレージに移動させたいケース
- バックアップ・アーカイブ・災害対策としてAWSにデータを保持
- オンプレミス環境で容易にAWSストレージを活用

[Q]ストレージゲートウェイタイプの選択

B社は3TBボリュームデータをオンプレミスのリポジトリ内に所有しており、大量の印刷ファイルを保存しています。このリポジトリは年間500GBほどの容量が増加しており、単一の論理ボリュームとして利用していく必要があります。 あなたはソリューションアーキテクトとして、頻繁にアクセスされるデータへの最適な応答時間を維持しつつ、ローカルストレージ容量の制約を回避するために、このレポジトリーをS3ストレージに拡張することになりました。S3をプライマリーに利用していく予定です。

要件を達成するために最適なAWS Storage Gateway構成はどれでしょうか？

1) S3への移転スケジュールが設定されたスナップショットを利用するキャッシュ型ボリューム
2) S3への移転スケジュールが設定されたスナップショットを利用する保管型ボリューム
3) Glacier（迅速アクセス）への移転スケジュールが設定されたスナップショットを利用するキャッシュ型ボリューム
4) S3への移転スケジュールが設定されたスナップショットを利用する仮想テープライブラリー

Storage Gatewayのタイプ

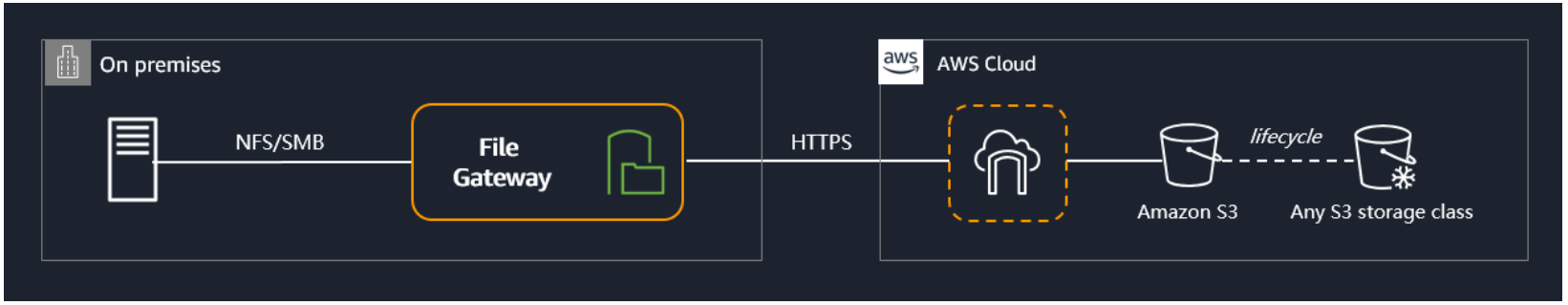
利用するデータタイプに応じて3つのゲートウェイを利用する

ファイルゲートウェイ	• シームレスにAWSクラウドに接続して、データファイルやバックアップイメージをAmazon S3 クラウドストレージに保存するバックアップソリューションを提供する。
ボリュームゲートウェイ	• オンプレミスアプリケーションにクラウドバックアップの iSCSI ブロックストレージボリュームを提供 • キャッシュ型ボリュームまたは保管型ボリュームのどちらかを使用
テープゲートウェイ	• Amazon S3とGlacierにデータを保管する仮想テープストレージとVTL管理を提供する。

ファイルゲートウェイ

オンプレミスのファイルデータをAWS Storage Gateway経由でAmazon S3上のオブジェクトとして格納

- 仮想アプライアンスでSMB または NFS のインターフェースを利用してデータを転送
- S3 標準、S3 低頻度アクセス、S3 Glacier、S3 Glacier Deep Archive のような低コストの S3 ストレージクラスにバックアップする。
- S3のライフサイクルポリシー／バージョニング／クロスリージョンレプリケーション等が利用可能
- 更新データは非同期でAWSに転送



Reference: https://aws.amazon.com/jp/storagegateway/file/

ボリュームゲートウェイ

オンプレミス環境のディスクデータに対するS3とのハイブリッド構成を実現

- 最新のアクセスデータのキャッシュ、またはボリューム全体のコピーのいずれかをオンプレミスで維持し、アプリケーションからデータへの高速アクセスが可能
- iSCSIでブロックストレージとしてインターフェースを提供
- オンプレミスのローカルディスクのバックアップを自動的にAWS側で実施
- 更新データは非同期でAWSに転送
- Amazon EBS スナップショット、Storage Gateway ボリュームクローン、AWS Backupを利用したデータ保護・復旧が可能

Reference: https://aws.amazon.com/jp/storagegateway/volume/?nc=sn&loc=2&dn=4

ボリュームゲートウェイのタイプ

プライマリーをオンプレミスとS3のどちらに設定するかで2つのタイプから選択する。

キャッシュ型ボリューム ゲートウェイ	• **プライマリーはS3ストレージ** • オンプレミス環境のストレージをS3に拡張する • 頻繁にアクセスされるデータはローカルのストレージゲートウェイに保持して、Amazon S3 をプライマリデータストレージとして使用 • 頻繁にアクセスするデータはオンプレミス環境にキャッシュ保持することで、低レイテンシーなアクセスが可能
保管型ボリューム ゲートウェイ	• **プライマリーはオンプレミスストレージ** • プライマリデータをローカルに保存する一方で、そのデータを非同期にAmazon S3にバックアップする。 • オンプレミスのアプリケーションがそのデータセット全体に低レイテンシーでアクセスが可能となる。

テープゲートウェイ

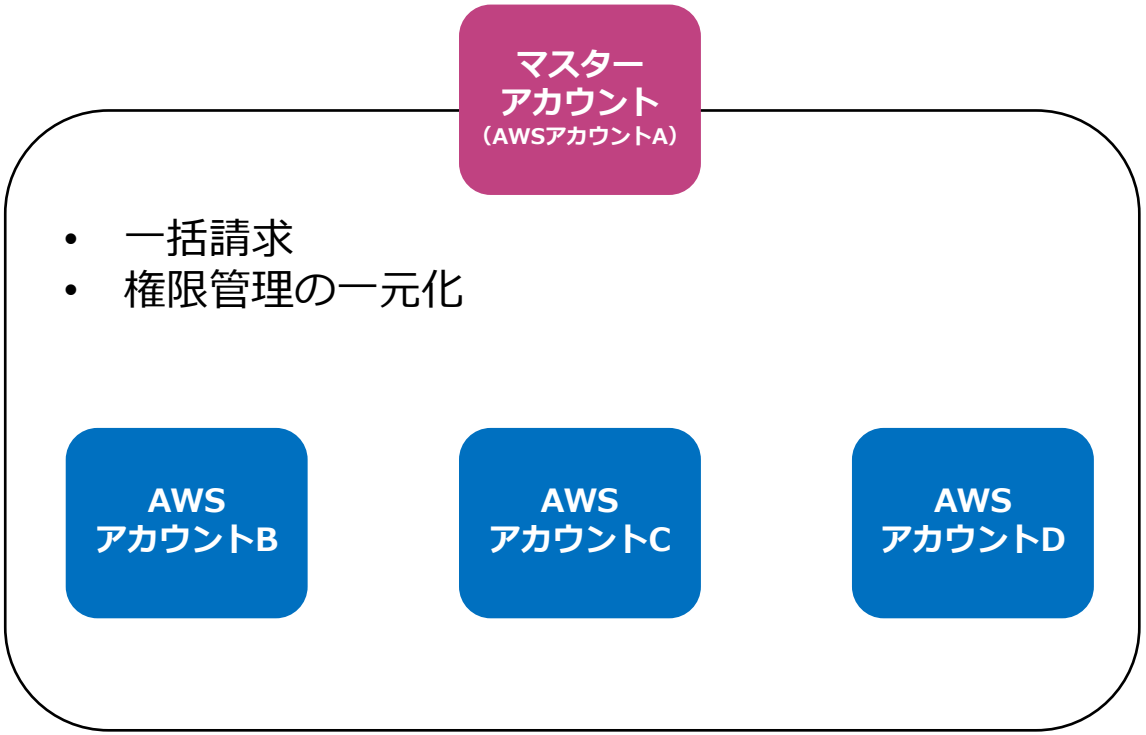
Storage Gatewayを仮想テープライブラリとして利用することで堅牢性の高い外部保管バックアップストレージを実現

- VTL(Virtual Tape Library)対応バックアップソフトウェアを利用し、Storage Gatewayを経由して、バックアップデータをS3およびGlacierに格納
- オンプレミスおよびAWSのEC2環境で利用可能
- バックアップソフトウェアによりテープ取り出しオペレーションを行うことで、安価なアーカイブストレージ（S3／Glacier）を利用
- 主要なバックアップソフトウェアをサポート

AWS Organizations
の出題範囲

AWS Organizationsとは何か？

複数のAWSアカウントを利用している場合に、統合管理を実施することができる



AWS Organizationsの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

AWS Organizations の選択	✓ AWSの複数アカウント管理を実施する要件に基づき適切な管理サービスを選択する問題が出題される。
アカウントの設定	✓ マスターアカウントとメンバーアカウントの設定方法や削除方法に関する質問が出題される。
一括請求のメリット	✓ 一括請求設定におけるメリットが問われる。
SCP	✓ Service Control Policyの設定方法やその設定目的が問われる。 ✓ またシナリオに基づいて、実際にSCPの設定結果がとわれる。
リソースのシェア	✓ AWS Organizationsを設定することで可能なメンバー間のリソースシェアの仕組みが問われる。

[Q]AWS Organizationsの選択

ある企業ではAWSアカウントを3つ社内で利用しています。部署ごとにばらばらに請求処理をしているため、CIO室が中心となってITコスト管理と運用を統括することが決まりました。AWSの複数アカウントをまとめる仕組みが必要となります。

AWSの複数アカウントをまとめるために利用できるサービスはどれでしょうか？

1) AWS Organizations
2) IAM
3) AWS Trusted Advisor
4) AWS Systems Manager

AWS Organizations

AWS OrganizationsはIAMのアクセス管理を大きな組織でも楽に実施できるようにするマネージド型サービス

複数アカウントの一元管理	新規アカウント作成の自動化	一括請求
AWSアカウントをグループ化してポリシーを適用して一元的に管理する	コンソール／SDK／CLIでAWSアカウントを新規作成して、作成内容をログ管理できる	複数AWSアカウントの請求を一括化する

[Q]アカウントの設定

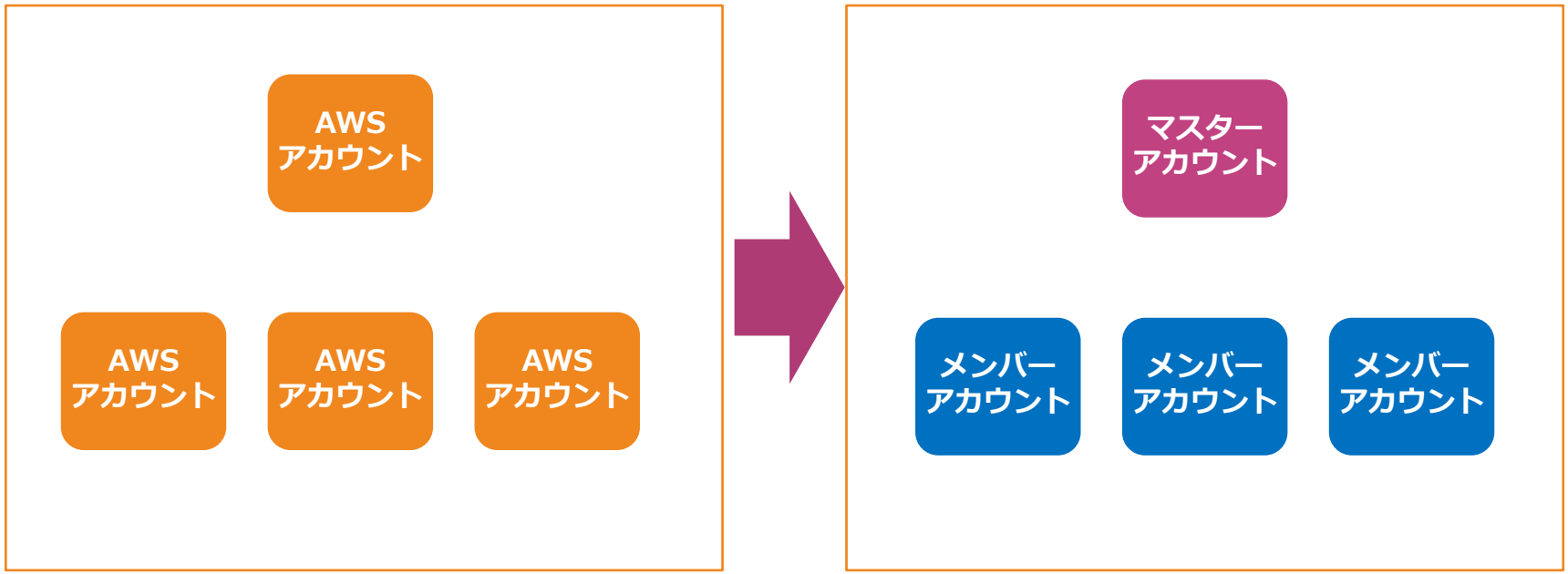
ある企業ではAWSアカウントを3つ社内で利用しています。部署ごとにばらばらに請求処理をしているため、AWS Organizationsを利用してITコスト管理と運用を統括しています。管理の都合上、２つのマスターアカウントを設定して、2つのOrganizationsの設定をしています。現在マスター―アカウントAのOrganizationzに所属しているメンバーアカウントBを別のアカウントCが管理するOrganizationsに移転させる要件が発生しました。

メンバーアカウントを移動する方法はどれでしょうか？（2つ選択してください。）

1) 既存のOrganizationsから削除したいメンバーアカウントBに必要な設定を付与した上で、マスターアカウントAによって削除を実行する。
2) マスターアカウントCからの新しいOrganizationsへの招待をメンバーアカウントBが承諾をするとアカウントAのOrganizationからアカウントBが削除される。
3) アカウントAから新しいOrganizationsへの権限移行を実施して、アカウントCが承諾すると、アカウントBが新しいOrganizationsに登録される。
4) アカウントAから新しいOrganizationsへの権限移行を実施して、アカウントCが承諾すると、アカウントBがアカウントAのOrganizationから削除される。
5) 新しいOrganizationsへの招待をマスターアカウントCからアカウントBに対して実施する。アカウントBから承諾を得るとメンバーアカウントに追加される。
6) Organizationsから削除したいメンバーアカウントBのルートアカウントが自ら脱退処理を実行する。

アカウントの設定

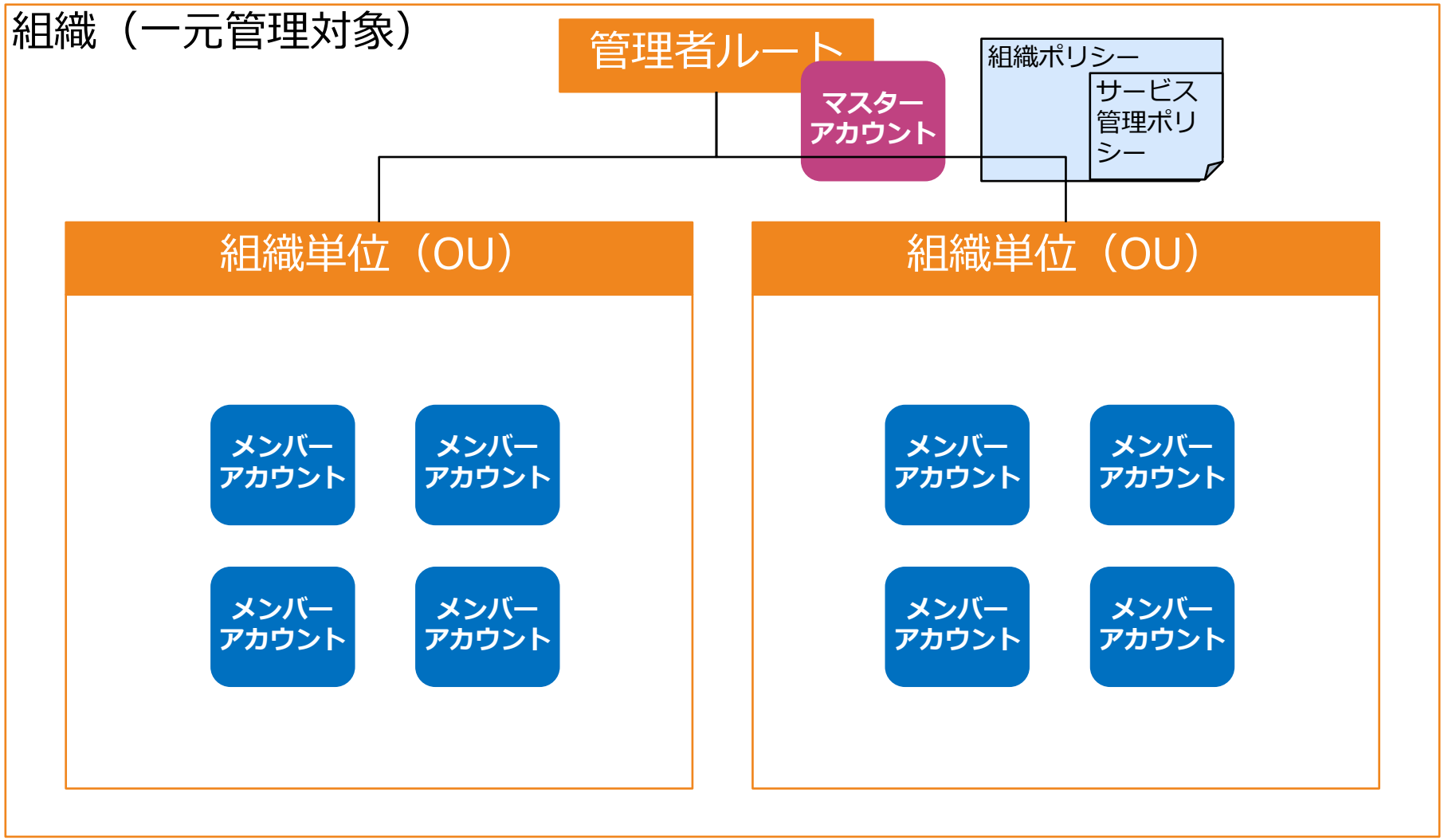
AWSアカウントの中からマスターアカウントを選定する



- ✓ メンバーアカウントはマスターアカウントから招待を承認するとメンバーアカウントとして登録される。
- ✓ メンバーアカウントから削除する際は独立したアカウントとして請求処理などの権限が整備されている必要がある。

AWS Organizations

組織という単位を構成して、マスターアカウントがメンバーアカウントを管理するという仕組み



[Q]一括請求のメリット

ある企業ではAWSアカウントを3つ社内で利用しています。部署ごとにばらばらに請求処理をしているため、AWS Organizationsを利用してITコスト管理と運用を統括しています。そこで、あなたはソリューションアーキテクトとして、AWS Organizationsを利用して3つのAWSアカウントに一括請求を設定するべきか検討することになりました。

AWS Organizationsを利用するべきコストメリットを選択してください。

1) 各アカウントがS3を利用している場合は、S3コストのボリュームディスカウントが適用され、コスト削減が確実に可能である。
2) 各アカウントがS3を利用している場合は、S3コストにOrganizations専用の価格帯が適用される。
3) 各アカウントがS3を利用している場合は、S3コストの利用ボリュームが増大する。
4) 各アカウントがS3を利用している場合は、S3コストを削減できる可能性がある。

機能セットの選択

支払一括代行とアカウントの全体管理の2つの方式を選択する

Consolidated Billing Only

- ✓ 支払一括代行のみを実施する場合に選択
- ✓ ボリュームディスカウントを統合できるため、コストメリットが発生する。

All Feature

- ✓ 支払一括代行も含めて、企業内の複数アカウントを統制したい場合に選択

[Q] SCP

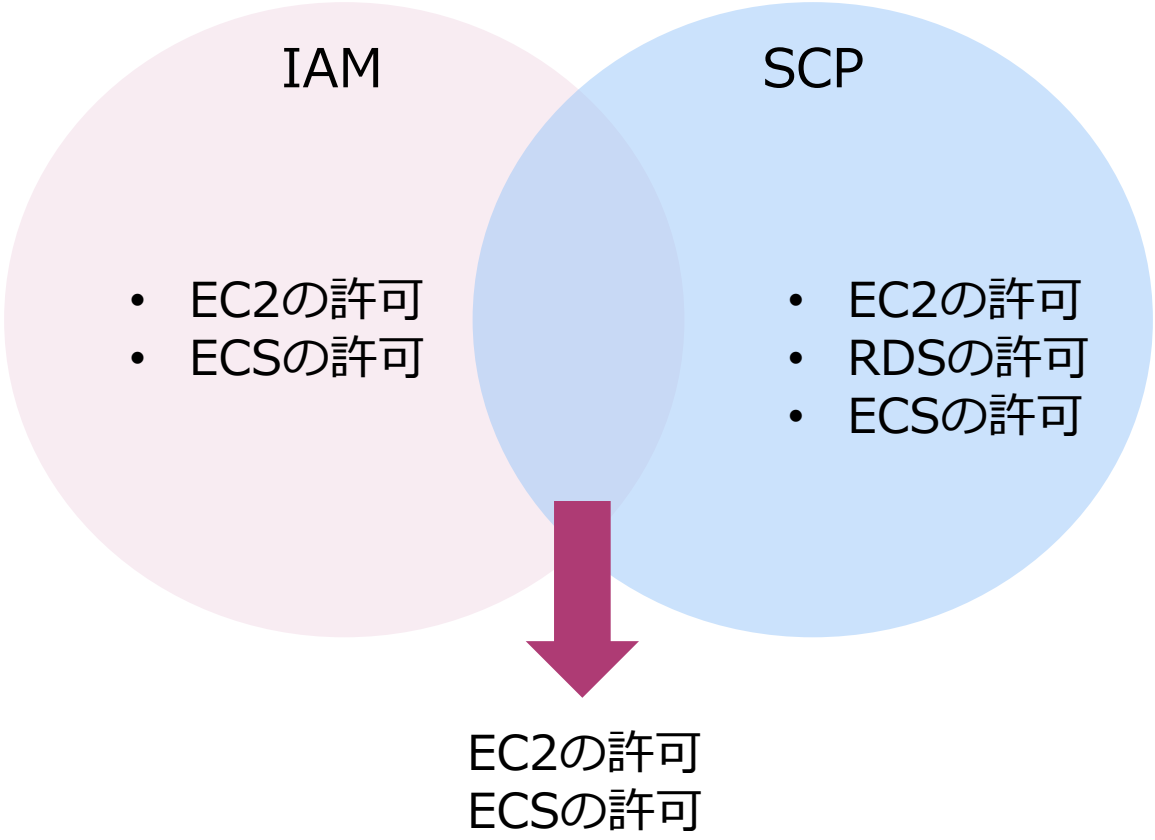
ある企業ではAWSアカウントを3つ社内で利用しています。部署ごとにばらばらに請求処理をしているため、AWS Organizationsを利用してITコスト管理と運用を統括しています。サービスコントロールポリシー（SCP）を使用して、組織内のすべてのアカウントでアクセス許可を一元管理しています。

SCPの利用としてどれが正しいですか？ （3つ選択してください）

1) SCPによりEC2へのアクセスを設定されたメンバーアカウントのIAMユーザーはEC2の操作権限が付与されている。
2) SCPはルートユーザーを含む、設定されたメンバーアカウントのすべてのユーザーとロールに影響する。
3) SCPはルートユーザー以外の設定されたメンバーアカウントのすべてのユーザーとロールに影響する。
4) SCPはサービスにリンクされたロールに影響を与える。
5) SCPはサービスにリンクされたロールに影響を与えない。
6) SCPによりEC2へのアクセスを設定されたメンバーアカウントのIAMユーザーはEC2の操作権限が付与されているわけではない。

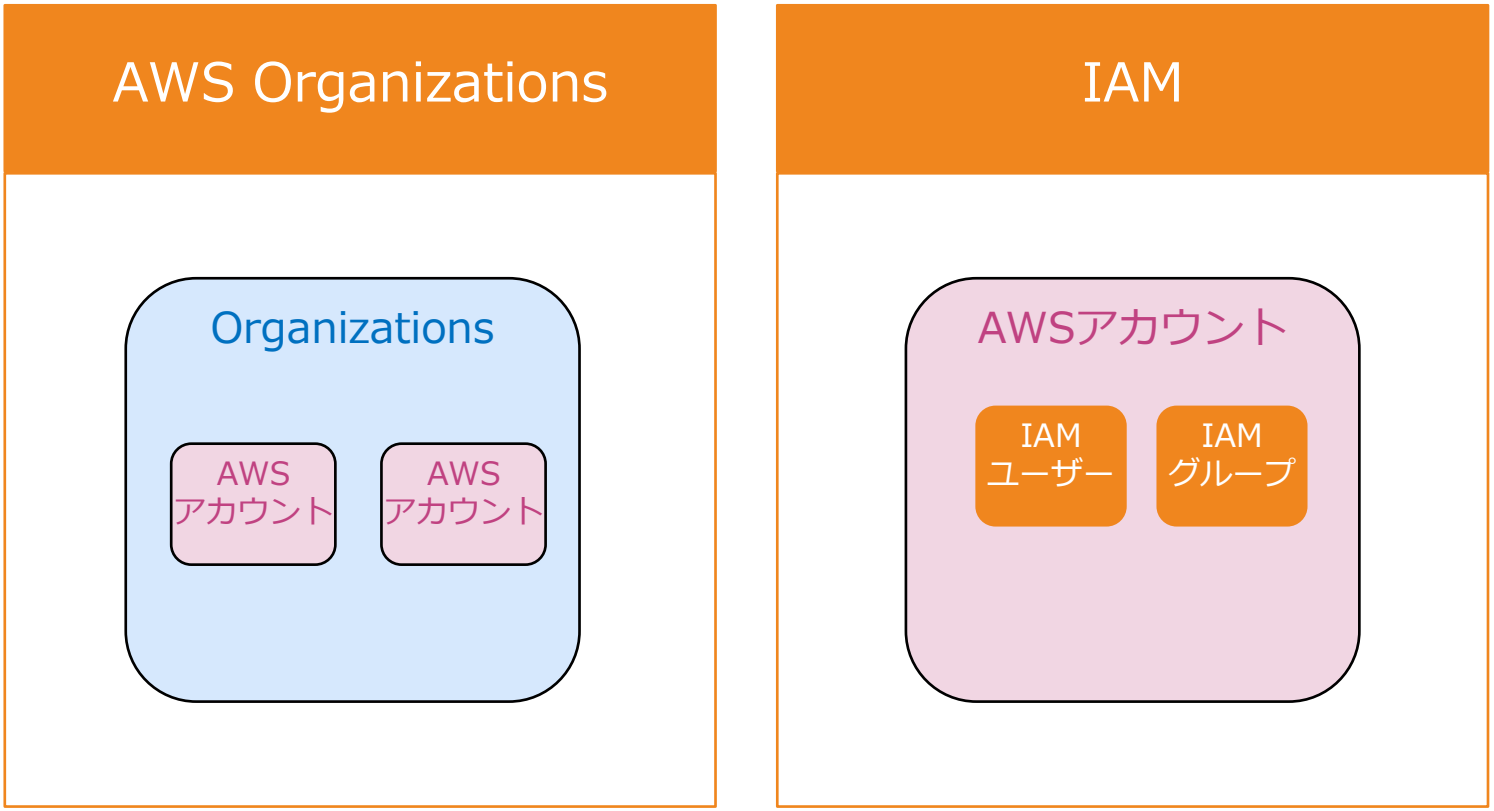
SCP

SCPというポリシーを利用して、OU内のメンバーに対して権限境界を設定することができる。



IAMとAWS Organizations

IAMはAWSアカウント内のユーザー管理を実施。
Organizationsは複数のAWSアカウント自体の管理を実施。



[Q]リソースのシェア

大手ニュースサイトを運営しているA社はAWSクラウドを使用してITインフラストラクチャを管理しています。 同社は複数のAWSアカウントを利用しているため、AWS Organizationsを利用してアカウント管理を実施することになりました。同社は、高度な相互接続性を必要とするアプリケーションを運用しており、メンバーアカウント間でVPCを共有することが必要です。

VPCを共有するためには、どのような設定が必要でしょうか？

1) AWS OrganizationsのVPC共有機能をオンにして、複数のメンバーアカウント間でVPC共有を使用する。
2) AWS Organizationsのデフォルト設定で、複数のメンバーアカウント間でVPC共有することができる。
3) AWS RAMと連携して複数のメンバーアカウント間でVPC共有を使用する。
4) IAMと連携して複数のメンバーアカウント間でVPC共有を使用する。

リソースのシェア

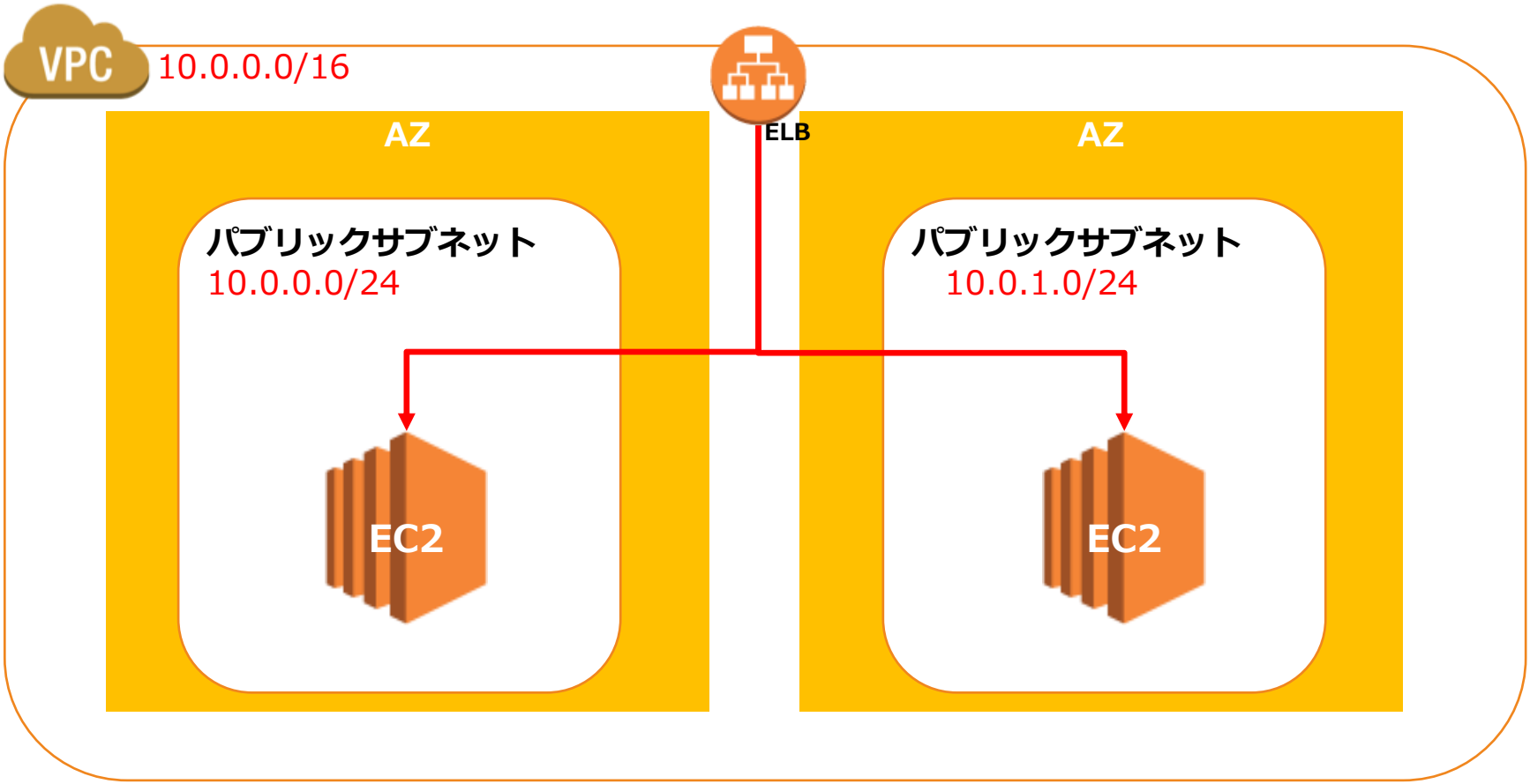
AWS Organizationsのメンバーアカウント間でリソースを共有できる。

AWS Resource Access Manager（RAM）との連携	リザーブドインスタンスの共有
✓ RAMと連携して、組織または組織単位とリソースを共有する	✓ アカウントでリザーブドインスタンスの共有がオンになっている ✓ AWS Organizationsを利用した一括請求でリザーブドインスタンスが共有される。

マルチAZ構成の出題範囲

マルチAZ構成とは何か？

複数AZにAWSリソースを冗長化して、可用性を高めるインフラ構成のこと



マルチAZ構成の出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

EC2のマルチAZ構成	✓ EC2を利用したマルチAZ構成が問われことが多い。 ✓ ELBとAuto Scalingを加えた冗長化構成が求められる。 ✓ DBサーバーやRDSを利用したマルチAZ構成も問われる
適切なサブネット構成	✓ パブリックサブネットやプライベートサブネットなどの適切なサブネット構成が問われることもある。

[Q]EC2のマルチAZ構成

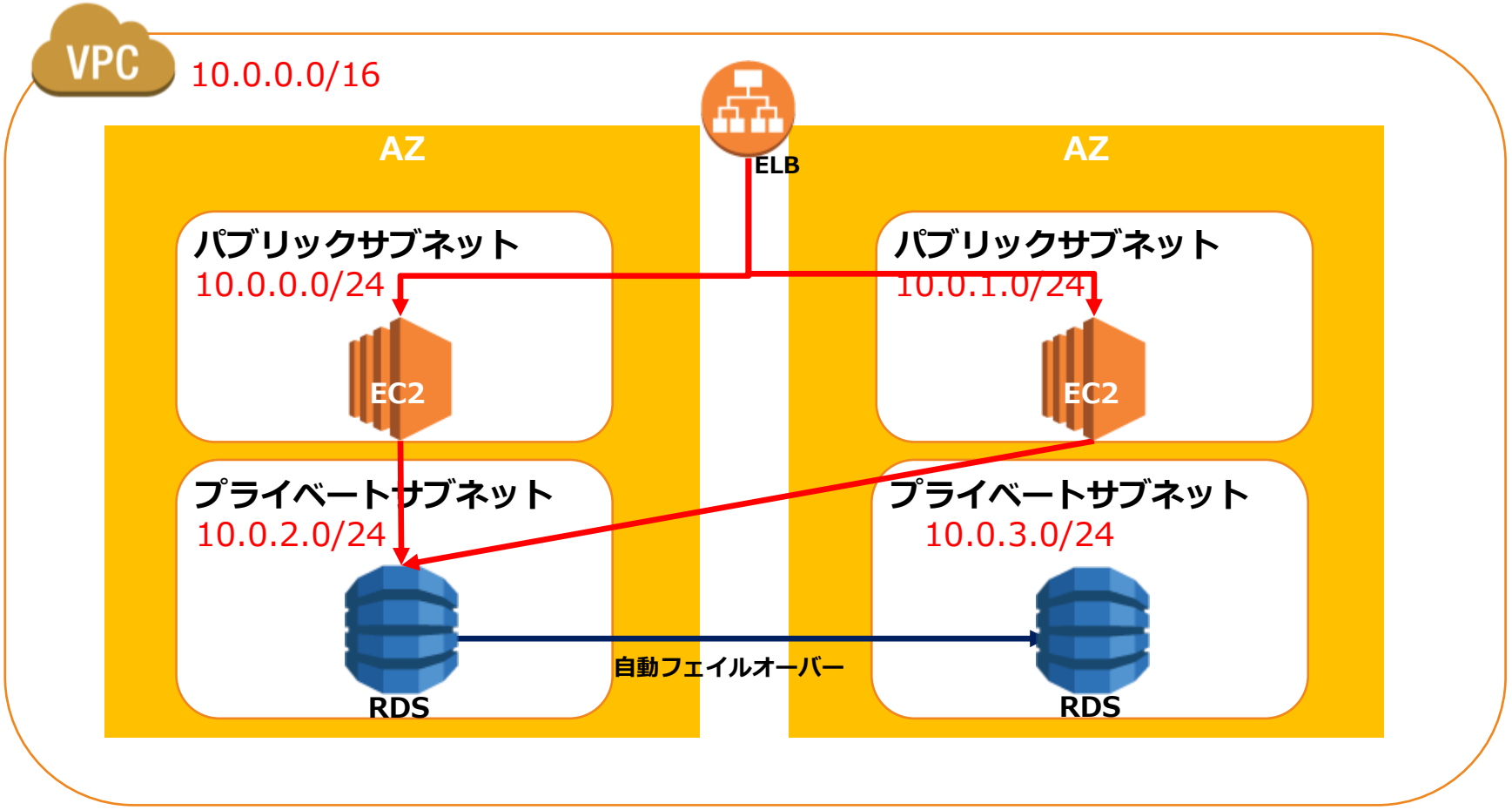
IT企業はAWSを利用してWEBアプリケーションを構築してます。アプリケーションのウェブ層はEC2インスタンスで実行され、データベース層はAmazon RDSMySQLを利用します。 現在、すべてのリソースが単一のアベイラビリティーゾーンにデプロイされています。 開発チームは運用開始前にアプリケーションの可用性を向上させたいと考えています。

このWEBアプリケーションを冗長化するためのアーキテクチャ構成はどれでしょうか？（2つ選択してください。）

1) ELBの背後にある2つのAZにWeb層のEC2インスタンスをデプロイする。
2) ELBの背後にある2つのリージョンにWeb層のEC2インスタンスをデプロイする。
3) ELBの背後にある2つのVPCにWeb層のEC2インスタンスをデプロイする。
4) マルチAZ構成でAmazon RDS MySQLデータベースをデプロイする。
5) グローバルデータベース構成でAmazon RDS MySQLデータベースをデプロイする。

マルチAZ構成

WEBサーバーのパブリックサブネットで冗長化して、RDSをフェールオーバー構成とするのが基本構成



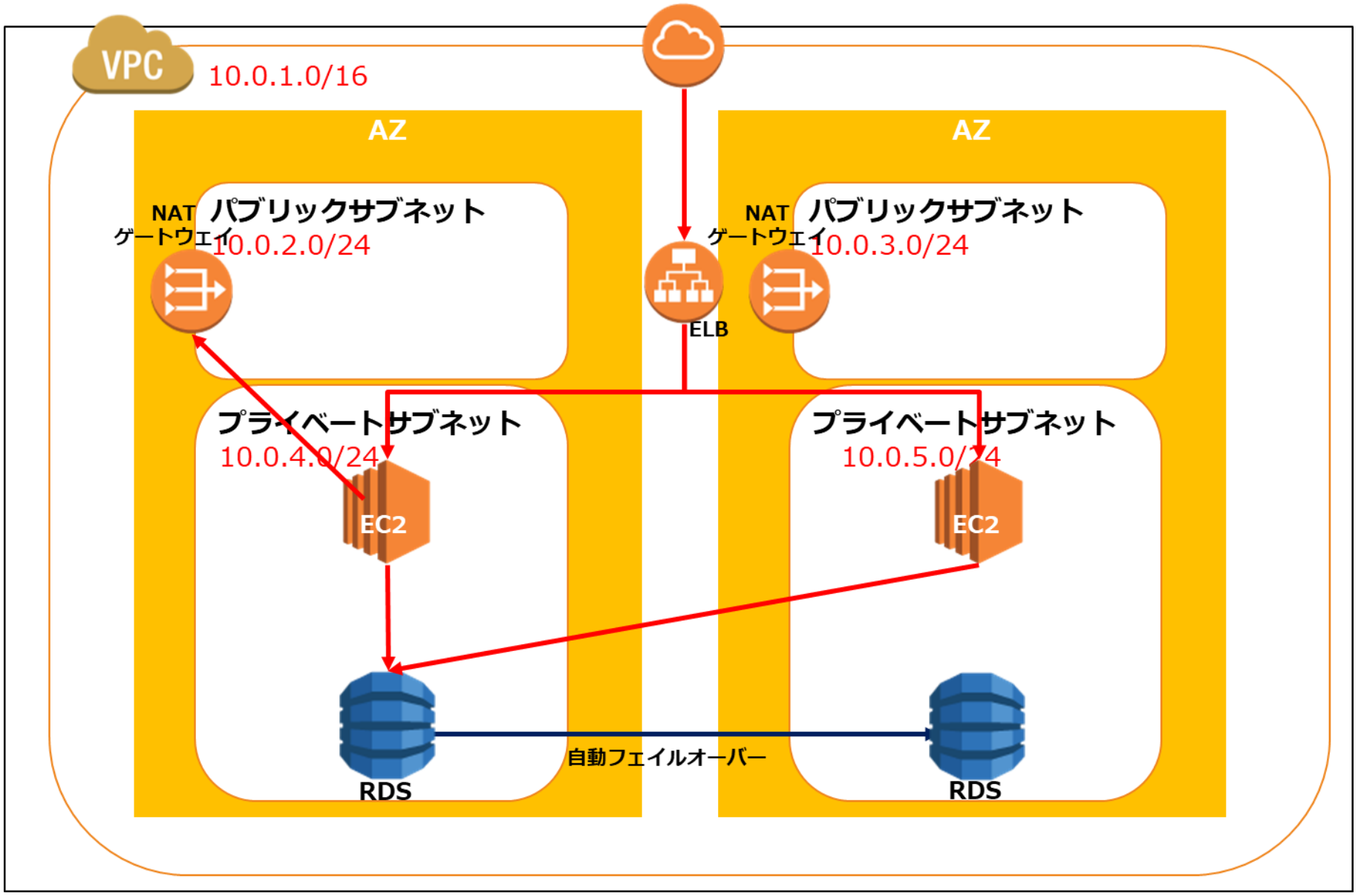
[Q]DBのマルチAZ構成

IT企業はAWSを利用してWEBアプリケーションを構築してます。アプリケーションのウェブ層はEC2インスタンスで実行され、データベース層はAmazon RDSMySQLを利用します。各インスタンスはセキュリティを重視して、プライベートサブネットに配置されています。WEBサーバーはインターネットからインスタンスにソフトウェアパッチをダウンロードすることが必要です。NATゲートウェイで障害が発生した場合にアクセスができなくなる問題に対処することが求められています。

どのように可用性と費用効果を高めることができますか？

1) 各アベイラビリティーゾーンにNATインスタンスを作成する。 NATインスタンスをELBでトラフィック分散する。
2) 各アベイラビリティーゾーンにNATゲートウェイを作成する。 NATゲートウェイをELBでトラフィック分散する。
3) 各アベイラビリティーゾーンにNATゲートウェイを作成する。 インスタンスが同じアベイラビリティーゾーンでNATゲートウェイを使用するように、各プライベートサブネットでルートテーブルを構成する。
4) 各アベイラビリティーゾーンにNATゲートウェイを作成する。各NATゲートウェイがヘルスチェックに基づいてEC2インスタンスへのトラフィックを制御する。

マルチAZ構成


VPC
10.0.1.0/16
AZ
AZ
NAT
ゲートウェイ
パブリックサブネット
10.0.2.0/24
NAT
ゲートウェイ
パブリックサブネット
10.0.3.0/24
ELB
プライベートサブネット
10.0.4.0/24
プライベートサブネット
10.0.5.0/24
EC2
EC2
RDS
自動フェイルオーバー
RDS

Amazon FSx
の出題範囲

Amazon FSxとは何か？

業界標準のファイルストレージを提供するフルマネージド型のストレージサービス

Amazon FSx for Windowsの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Amazon FSx for Windowsの選択	✓ シナリオに基づいて、ストレージの要件が提示されるため Amazon FSx for Windowsを選択する問題
Amazon FSx for Lustreの選択	✓ シナリオに基づいて、ストレージの要件が提示されるため Amazon FSx for Lustreを選択する問題

3つのファイルストレージ

EFS以外にもユースケースに応じてFSxタイプのファイルストレージが2タイプ利用可能

EFS	◘ NASに似たファイルストレージ ◘ ファイルシステムとして利用し、複数のEC2インスタンスでの共有アクセスが可能 ◘ S3と異なりインターネットから直接アクセスができない
Amazon FSx For Windows File Server	◘ Windows File Serverと互換性のあるストレージ ◘ Windows上に構築され、Windows AD、OSやソフトウェアとの連携が豊富に可能
Amazon FSx For Lustre	◘ 分散型ファイルストレージであるオープンソースLusterと互換性のある分散型の高速ストレージ ◘ 機械学習などの高速コンピューティングのデータレイヤーに利用する一時保存用の処理用ストレージ

[Q]Amazon FSx for Windowsの選択

A銀行ではMicrosoftの分散ファイルシステムを利用したWEBアプリケーションを
AWS上に構築しています。あなたはソリューションアーキテクトとして、この分散
ファイルシステムに適合した最適なストレージを選択することが必要です。

次のAWSサービスのうち、どのサービスが最適でしょうか？

1) Amazon FSx for Windows File Server
2) Amazon FSx for Lustre
3) EFS
4) AWS Managed Microsoft AD

Amazon FSx For Windows File Server

Windows File ServerをAWSクラウド上で利用したい場合に利用するストレージ

特徴・ユースケース

- Windows File Serverのクラウド移行
- Active Directory (AD) 統合などの幅広い管理機能
- SMB プロトコルによりAmazon EC2、VMware Cloud on AWS、Amazon WorkSpaces、Amazon AppStream 2.0 インスタンスなど幅広く接続可能
- 最大数千台のコンピューティングインスタンスからアクセス可能

アーキテクチャ構成

- ENI経由でアクセス
- VPCセキュリティグループでの制御
- 単一AZの単一サブネットを指定して構成する。
- 複数インスタンスでの共有や他AZ内のインスタンスからのアクセスも可能
- マルチAZ構成を実施することもできる。

[Q] Amazon FSX for Lustreの選択

大手航空関連企業はエンジン開発に向けてシミュレーションシステムをAWS上に構築しています。これはエンジンのパフォーマンスと障害予測をシミュレートするために使用される高性能ワークフローです。 分析時に「ホットデータ」は、並列分散方式で迅速に処理および保存する必要があります。 「コールドデータ」は、低コストで読み取りと更新にすばやくアクセスできるように、参照用に保持する必要があります。

このような高度なシミュレーションを実施するために最適なストレージを選択してください。

1) Amazon EMR
2) EFS
3) Amazon FSx for Lustre
4) Amazon FSx for Windows File Server

Amazon FSx For Lustre

高速コンピューティング処理を実現する分散・並列処理専用の超高性能ストレージを提供

特徴・ユースケース

- 多くのスーパーコンピューターに利用される分散ファイルシステム
- フルマネージド型で安全にLustre利用
- 最適容量3600GB
- 最大数百GB/秒のスループット
- 数百万IOPSまでスケール可能

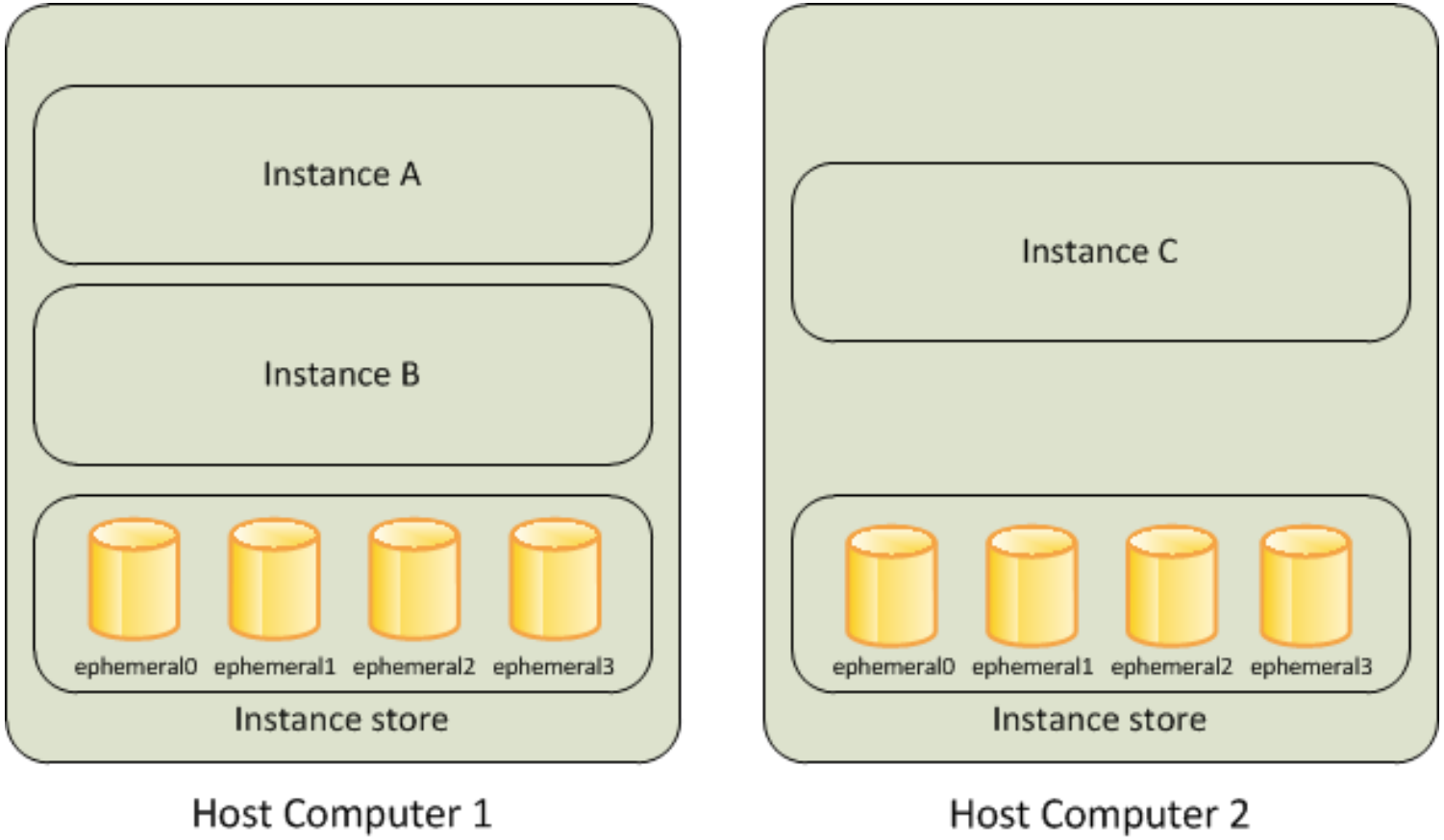
アーキテクチャ構成

- ENI／エンドポイント経由でアクセス
- セキュリティグループで制御
- 単一AZの単一サブネットを指定して構成する。
- Amazon S3とシームレスな統合によりデータレイク型のビッグデータ処理や解析ソリューションに組み込む

インスタンスストア
の出題範囲

インスタンスストアとは何か？

EC2インスタンスに物理的にアタッチされている一時データ保存用のストレージ



Reference: https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/InstanceStorage.html

インスタンスストアの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

インスタンスストアの選択	✓ シナリオに基づいて、ストレージ要件を提示されインスタンスストアを選択する出題
インスタンスストアの特徴	✓ インスタンスストアの特徴に基づいて、ストレージの設定方法などが問われる。

[Q]インスタンスストアの選択

データ処理アプリケーションは50GB EBS汎用ボリュームを備えたG4.largeEC2インスタンスで実行されています。 アプリケーションは、EBSルートボリュームにある小さなデータベース（30 GB未満）に一時データを利用して、I / O速度を向上させることが必要となります。

データベースの応答時間を改善するための最も費用効果の高い方法は何ですか？

1) 一時データベースをインスタンスストレージに移動する。
2) 一3000IOPSが割り当てられた新しい50GBのプロビジョンドIOPSを利用する。
3) ストレージ最適化インスタンスへと変更する。
4) インスタンスサイズをより大きなものへと変更数r。

インスタンスストアの選択

EC2で直接利用するストレージは不可分なインスタンスストアと自分で設定するEBSの2つ

インスタンスストア	✓ ホストコンピュータに内蔵されたディスクでEC2と不可分のブロックレベルの物理ストレージ ✓ EC2の一時的なデータが保持され、EC2の停止・終了と共にクリアされる ✓ 無料
Elastic Block Store（EBS）	✓ ネットワークで接続されたブロックレベルのストレージでEC2とは独立して管理される ✓ EC2をTerminateしてもEBSは保持可能で、SnapshotをS3に保持可能 ✓ 別途EBS料金が必要

[Q]インスタンスストアの選択

データ処理アプリケーションは50GB EBS汎用ボリュームを備えたG4.largeEC2インスタンスで実行されています。 アプリケーションは、EBSルートボリュームにある小さなデータベース（30 GB未満）に一時データを利用して、I / O速度を向上させることが必要となります。

データベースの応答時間を改善するための最も費用効果の高い方法は何ですか？

1) 一時データベースをインスタンスストレージに移動する。
2) 一3000IOPSが割り当てられた新しい50GBのプロビジョンドIOPSを利用する。
3) ストレージ最適化インスタンスへと変更する。
4) インスタンスサイズをより大きなものへと変更数r。

[Q]インスタンスストアの特徴

あなたはソリューションアーキテクトとして、EC2インスタンスを起動してWEBアプリケーションを構築する予定です。一部のEC2インスタンスには高性能エフェメラルストレージを利用するという要件が出てきました。

新しいインスタンスストアボリュームをどのように追加すれば良いでしょうか？

1) 新規にインスタンスストアを起動して、インスタンスにアタッチする。
2) インスタンスの起動時にのみ、インスタンスのインスタンスストアボリュームを指定できる。
3) インスタンスを停止して、インスタンスストアをアタッチする。
4) インスタンスの起動時にブロックデバイスマッピングを使用して追加のインスタンスストアボリュームを指定する。

インスタンスストアの特徴

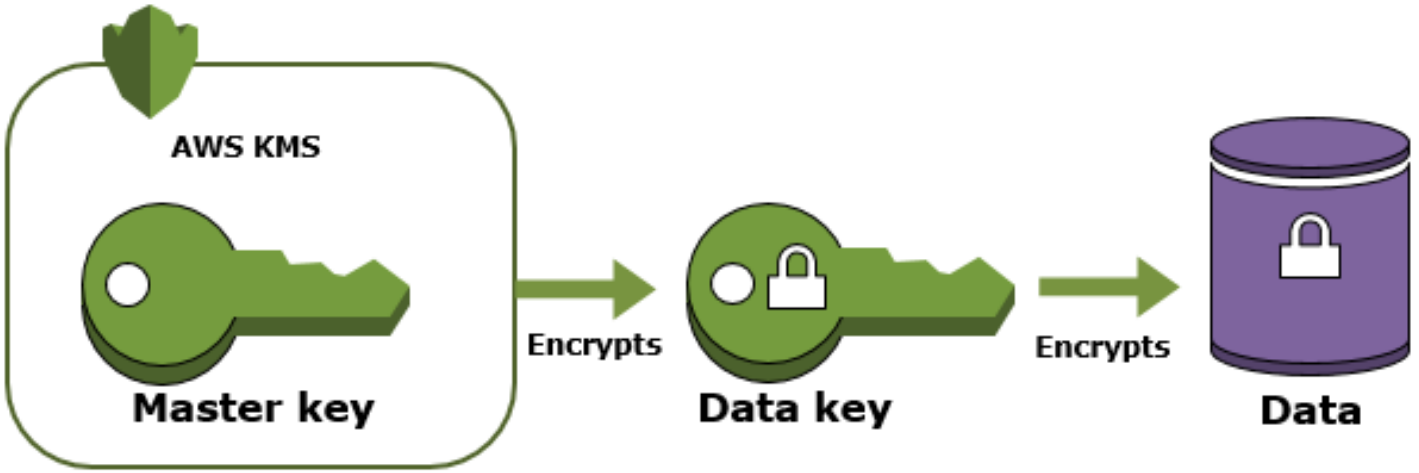
EC2で直接利用利用する不可分なストレージであるインスタンスストア

- インスタンスを起動する場合にのみインスタンスストアボリュームを指定可能
- インスタンスからデタッチして別のインスタンスにアタッチすることができない。
- インスタンスストア上のデータは関連付けられたインスタンスの運用中のみ維持
- 基盤となるディスクドライブで障害が発生した場合とインスタンスが停止・休止・終了時にデータは喪失する。
- インスタンスタイプにより、使用できるインスタンスストアのサイズ、およびインスタンスストアボリュームで使用されるハードウェアの種類が決まる。
- NVMe または SATA ベースのソリッドステートドライブ (SSD) を使用して、高いランダム I/O パフォーマンスを実現するものがある。
- ブロックデバイスマッピングを使用して、インスタンスの EBS ボリュームとインスタンスストアボリュームを指定
- インスタンスストアボリュームの仮想デバイスは ephemeralディスク

AWS KMSの出題範囲

KMSとは何か？

AWS KMSは暗号化に利用するマスターキーを作成・管理するサービス



Reference: https://docs.aws.amazon.com/crypto/latest/userguide/awscryp-service-kms.html

KMSの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

KMSの選択	✓ 暗号化を実施する手段として要件が提示され、KMSを選択する問題が出題される。
CMKの管理	✓ KMSで作成されるCMKの管理における基本的な知識が問われる。
KMSの設定	✓ KMSをアプリケーションで利用する際などの個別要件に合わせた設定方式が問われる。

[Q]KMSの選択

あなたはソリューションアーキテクトとして、複数のEBSボリュームを持つ大規模な専用EC2インスタンスを使用して医療情報共有アプリケーションを構築しています。EBSボリュームは、処理するデータの機密性を考慮し、HIPAA標準に準拠するために暗号化する必要があります。

EBS暗号化では、AWSは保存されているボリュームデータを保護するためにどのサービスを使用しますか？

1）AWS Key Management Service（KMS）でキー管理を実施する

2）SSE-EBSを利用したAmazonが管理するキーを使用する。

3）ACMを利用したキー管理を実施する。

4）AWS Certificate Manager（ACM）によって提供されるSSL証明書の使用する。

AWS KMS

AWS KMSはデータを暗号化するためのマネージド型暗号キーの作成・管理サービス

- ✓ 暗号鍵の作成・管理・運用を実施するマネージドサービスでAWS マネジメントコンソール、AWS SDK またはCLI を使用して、キーを作成、インポート、ローテーション、削除、管理する。
- ✓ IAMと連携して鍵のアクセス管理を実施
- ✓ カスタマーマスターキー（CMK）の無効化・有効化・削除を実施し、１年ごとの自動キーローテーションすることが可能
- ✓ CMKを外部から持ち込んで管理することも可能
- ✓ キーを保護するために FIPS 140-2 の検証済みまたは検証段階のハードウェアセキュリティモジュールを使用
- ✓ AWS CloudTrail と統合されており、すべてのキーの使用ログを表示
- ✓ RDSやS3などの多数のAWSサービスに適用可能
- ✓ KMS SDKを利用することで、アプリケーションにおける暗号化も可能

[Q] CMKの管理

あなたはソリューションアーキテクトとして、S3バケットにデータを保存するEC2インスタンスを使用してWEBアプリケーションを構築しています。 EBSボリュームはデータの機密性を考慮し、独自のカスタマーマスターキー（CMK）で暗号化を実施しています。メンバーが誤ってCMKを削除したため、ユーザーデータを復元できなくなりました。

この問題を解決するためにどうすればいいですか？

1) CMKを削除してしまうと、回復させることは不可能であり、データへのアクセス権限を失ってしまう。
2) AWSサポートに連絡することでCMKを復元することができる。
3) ルートアカウントユーザーからCMKを復元できる。
4) CMKは削除された直後であるため、CMKの削除をキャンセルして、キーを回復することができる。

AWS KMS

RDSでは保存されるデータ・リソースの暗号化と接続の暗号化を実施可能

カスタマーマスターキー（CMK）	✓ 暗号化を実行する上で最初に作成されるマスターキーで紛失するとデータにアクセスできなくなる根本的なもの ✓ 暗号化キーを暗号化する ✓ ローテーションされる。
カスタマーデータキー（暗号化キー）	✓ 実際のデータの暗号化に利用するキー ✓ KMSで生成されてCMKで暗号化される
エンベロープ暗号化	✓ マスターキーで暗号化をせずに、暗号化キーを利用して暗号化する暗号化方式

[Q] KMSの設定

ソリューションアーキテクトは暗号化ソリューションを開発しています。 このソリューションでは、データキーをディスクに書き込む前に、エンベロープ保護を使用して暗号化する必要があります。

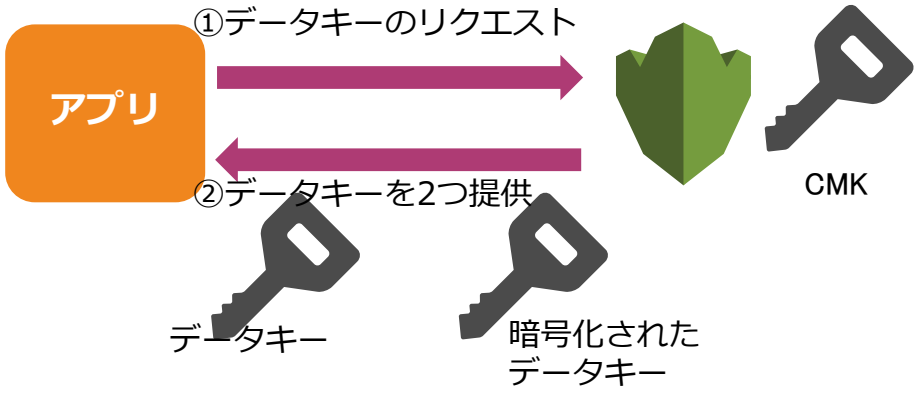
この要件を支援できるソリューションはどれですか？

1) AWS KMS API
2) API Gateway with STS
3) IAM Access Key
4) AWS Certificate Manager

エンベロープ暗号化

AWS KMSは簡単にデータを暗号化するためのマネージド型暗号化サービス

- データキーとカスタマーマスターキーによる暗号化を実施
- カスタマーマスターキー（CMK）を利用してデータキーを暗号化する（暗号化キーの作成）



③アプリ／ユーザーは暗号化されたデータキーを利用して暗号化する

④アプリ／ユーザーは暗号化データキーと暗号化データを送付する

AWS Snowファミリー
の出題範囲

AWS Snowファミリー

物理ストレージデバイスを使用し、インターネットを迂回して
AWSに直接大容量データを転送するサービス

Snowball

ペタバイト規模のデータ移動
（現在は非推奨）

Snowball Edge

ペタバイト規模のデータ移動
＋
コンピューティング
とストレージ機能

Snowmobile

エクサバイト規模の
データ移動

Snowファミリーの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Snowballタイプの選択	✓ データ移行に関するシナリオに基づいて要件が提示されて、利用するべきSnowballのタイプと台数を選択する。

Snowballタイプの選択

C社は現在はオンプレミスネットワークでホストされているインフラとアプリケーション全般をAWSクラウドに移行することを決定しました。現在、タイムリーかつ費用対効果の高い方法でS3バケットに移動する必要がある合計80TBのデータがあります。 既存のインターネット接続の空き容量を使用してデータをAWSにアップロードするのには1週間以上かかると推定しました。

データ移行において最も早く実現可能な、費用対効果の高い方法を選択してください。

1) Snowballを2つ利用してデータ移行を実施する。
2) Snowball Edge Compute Optimizedを１つ利用してデータ移行を実施する。
3) Snowball Edge Storage Optimizedを１つ利用してデータ移行を実施する。
4) Direct Connectを一から接続して、データ移行を実施する。
5) Snowball Edge Storage Optimizedを２つ利用してデータ移行を実施する。

Snowball

ペタバイト規模のデータ移行の際に利用するアプライアンスで、現在はSnowball Edgeに世代交代済

- オンプレミスのデータストレージロケーションと Amazon S3 との間でデータのインポートおよびエクスポートができる。
- Snowball ではすべてのリージョンで 80 TB モデルを使用可能
- 暗号化が強制、保管中や輸送中のデータを保護
- AWS Snowball マネジメントコンソール を使用
- オンプレミスのデータセンターと Snowball 間でローカルデータ転送を実行
- Snowball はそれ自体が配送コンテナ

【ユースケース】
移行／災害対策のデータ移行／データセンター統合／コンテンツ配信に伴うデータ移行

参照：https://docs.aws.amazon.com/ja_jp/snowball/latest/ug/receive-device.html

SnowballとSnowball Edge

Snowball EdgeはSnowball＋コンピューティングという高性能な機能を有し、現在AWSはSnowball Edge利用を推奨

Snowball	Snowball Edge
• クライアント側で暗号化を実施 • クライアント側にリッチなリソースが必要で、クライアント側のソフトウェアによるデータ転送を実施 • 容量：80TB／アプライアンス • 用途：データ移行 • クラスタリング：不可 • ラックマウント：不可 • 最大保持日数：90日	• Edge側で暗号化を実施 • 書込時にLambda関数を利用したデータ処理可能 アプライアンスに組み込まれたS3 Adapter For Snowballによりデータ転送を実施 • 容量：100TB（80TB利用）／アプライアンス • 用途：データ移行＋ローカルプロセッシング・ストレージとして利用 • クラスタリング：可 • ラックマウント：可 • 最大保持日数360日 • Snowball Edge Compute Optimized（42TB）のSnowball Edge Storage Optimized（80TB）の2タイプ

Snowmoblieの利用

超大容量データを AWS に移動するために使用できるエクサバイト規模のデータ転送サービス

参照：https://docs.aws.amazon.com/ja_jp/snowbal/latest/ug/receive-device.html

- セミトレーラートラックが牽引する長さ 14 m の丈夫な輸送コンテナ
- Snowmobile 1 台あたり 100 PB まで転送可能
- Snowmobileがデータセンターに輸送され直接にデータを取得
- データのロードが完了すると、Snowmobile は AWS に返送され、データは Amazon S3 にインポートされる
- データは 256 ビットの暗号化キーで暗号化
- 専門のセキュリティ担当者、GPS 追跡、アラームモニタリング、24 時間年中無休の監視カメラ、および輸送中に警護するセキュリティ車両 (オプション) など、データを保護する複数のセキュリティレイヤーを使用

【ユースケース】
ビデオライブラリや画像リポジトリ、またはデータセンター全体まで、膨大な量のデータを移行

Glacierの出題範囲

Amazon S3 Glacier

バックアップなど中長期保存用のS3よりも安価なストレージ

S3と同じ耐久性で
値段が安い！

データ取得などの
迅速性がない！

Glacierの出題範囲

1625問から質問出題範囲を分析した結果は以下の通り

Glacierの選択	✓ シナリオに基づいてGlacierを選択する問題が出題される。 ✓ S3の問題と同じもの
Glacierの特徴	✓ ストレージ選択の際にGlacierの特徴を答えさせるという問題が出題される。
Glacierのデータ取り出し	✓ Glacierのデータ取り出しの際に選択できるデータ取り出し方法のタイプを選択する問題 ✓ プロビジョンドキャパシティの利用が問われることもある
ボールトロックの利用	✓ コンプライアンス強化の要件が提示されてボールトロックの利用が求められる問題が出題される。

[Q]Glacierの特徴

あなたはソリューションアーキテクトとして、S3を利用してライフサイクル管理を利用して、Amazon Glacierにアーカイブする計画をしています。 あなたは担当上司にデータの回復力を理解させる必要があります。

Amazon Glacierストレージについて正しい説明は次のうちどれですか？ （2つ選択してください）

1) アーカイブには99.999999999%の耐久性を提供する。
2) アーカイブには99.999%の耐久性を提供する
3) アーカイブを保存するためのコンテナとして「ボールト」を使用する。
4) アーカイブを保存するためのコンテナとして「バケット」を使用する。
5) アーカイブには99.99%の可用性を提供する。

Glacierの特徴

バックアップなど中長期保存用のS3よりも安価なストレージ

- ✓ Amazon S3 Glacier では、データは「アーカイブ」に保存される
- ✓ 1 つのアーカイブの最大サイズは 40 TB
- ✓ 保存可能なアーカイブ数とデータ量に制限なし
- ✓ 各アーカイブには作成時に一意のアーカイブ ID が割り当てられ、作成後はアーカイブを更新できない。
- ✓ アーカイブを保存するためのコンテナとして「ボールト」を使用（1 つの AWS アカウントでは、最大 1,000 個のボールトを使用）
- ✓ Amazon S3 のライフサイクルルールと連携させることにより、Amazon S3 データのアーカイブを自動化し、全体的なストレージコストを削減
- ✓ Advanced Encryption Standard (AES) 256 ビット対称鍵を使用してデフォルトで自動的に暗号化
- ✓ S3と違って直接データをアップロード・取得という処理ができないため、S3ライフサイクル管理からか、プログラム処理によるアップロード／ダウンロードが必要
- ✓ Glacierの最低保持期間は90日

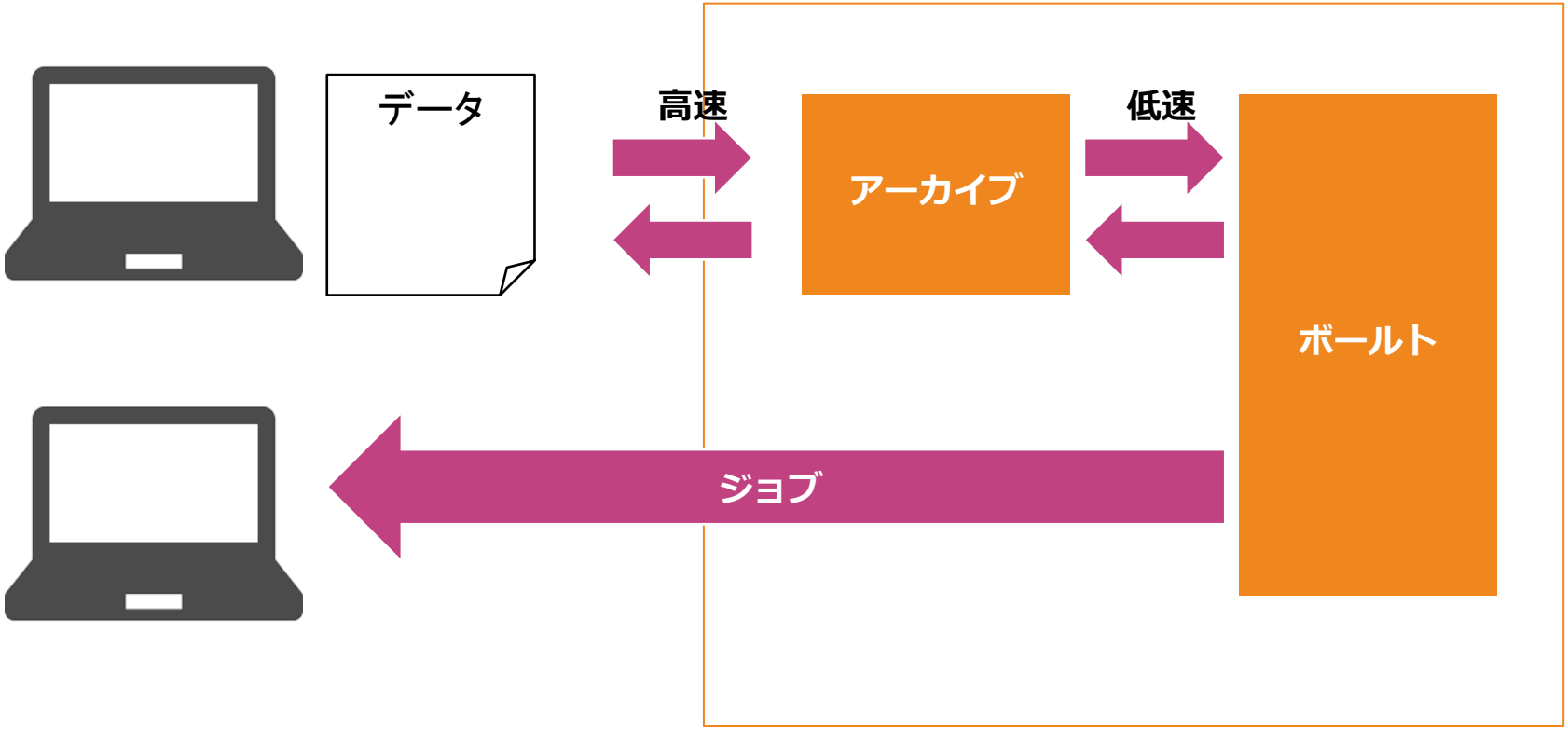
Glacierの仕組み

S3と異なり、ボールトとアーカイブという単位でデータを保存

管理方式	特徴
ボールト	✓ ボールトはアーカイブを格納するコンテナ ✓ ボールトはリージョンに作成
アーカイブ	✓ アーカイブは、写真、動画、ドキュメントなどの任意のデータで、S3 Glacier でのストレージの基本単位 ✓ 各アーカイブは一意のアドレスを持ちます。
ジョブ	✓ アーカイブに SELECT クエリを実行したり、アーカイブを取得したり、ボールトのインベントリを取得したりする実行単位
通知設定	✓ ジョブの完了には時間がかかるため、ジョブの完了時にSNSと連携した通知設定が可能

Glacierの仕組み

アーカイブに一時的にデータをアーカイブ処理して、ボールトに長期保存するという仕組み



[Q] Glacierのデータ取り出し

あなたはソリューションアーキテクトとして、AWSを利用して企業文書を管理・保存するためのソリューションを構築しています。データは一度保存されると滅多に利用されることはありませんが、必要に応じて管理者の指示に従って10時間以内に取得することが求められています。 あなたはAmazon Glacierを利用することを決定して、その設定方式を検討しています。

Glacierをどのように設定する必要がありますか。

1) 迅速取り出し
2) 標準取り出し
3) 大容量取り出し
4) ボールトロック

Glacierのデータ取出タイプ

Glacierのデータ取得タイプの設定に応じてデータ取得時間と取得時の料金が変わる

タイプ	特徴
迅速	✓ 迅速取り出しでは、アーカイブのサブセットが迅速に必要になった場合にデータにすばやくアクセスするモード。通常 1～5 分以内で使用可能になる
プロビジョニングキャパシティ	✓ プロビジョンドキャパシティーは、迅速取り出しの取得容量を必要なときに利用できることを保証する仕組み
標準	✓ 標準取り出しでは、数時間以内にすべてのアーカイブにアクセスできるデフォルト設定。通常、標準取り出しは 3～5 時間で完了
大容量	✓ 大容量取り出しは、最も安価な取り出しオプションであり、大量のデータ (ペタバイトのデータを含む) を 1 日以内に低コストで取得できます。通常、大容量取り出しは 5～12 時間で完了

[Q]ボールトロックの利用

ヘルスケアのスタートアップは、新規サービスにおいて医療情報共有アプリケーションを構築しており、AmazonS3に患者の健康記録を保存します。 あなたはソリューションアーキテクトとして、Amazon S3 Glacierに基づくアーカイブソリューションを実装して、データアクセスの規制およびコンプライアンス管理を実施する必要があります。

ソリューションアーキテクトとして、どのソリューションを選択するべきでしょうか？

1) S3 Glacierボールトを使用して機密性の高いアーカイブデータを保存してから、ボールトロックポリシーを使用する。
2) S3 Glacierアーカイブを使用して機密性の高いアーカイブデータを保存してから、アーカイブポリシーを使用する。
3) S3 Glacierボールトを使用して機密性の高いアーカイブデータを保存してから、ライフサイクルポリシーを使用する。
4) S3 Glacierアーカイブを使用して機密性の高いアーカイブデータを保存してから、リソースポリシーを使用する。

アクセス管理

Glacierのアクセス管理は用途に応じて方式を使い分ける

管理方式	特徴
IAMポリシー	✓ IAMユーザーやリソースに対してS3サービスへのアクセス権限を設定する ✓ 一元的にリソースへのアクセス権限を管理
ボールトポリシー	✓ ボールトで直接アクセスポリシーを定義して、組織内のユーザーや社外ユーザーに対してもボールトへのアクセス権を付与
データ取り出しポリシー	✓ データ取り出しに関する制限を定義 ✓ [無料利用枠のみ] に制限。または無料利用枠を超える量を取り出したい場合は、[最大取得率] を指定すると、取り出し速度を制限して、取り出しコストの上限を設定
ボールトロックポリシー	✓ ロックによって変更を禁止することにより、コンプライアンス管理を強力に実施することが可能
署名	✓ 認証保護のために、全リクエストに署名が必要

Amazon Glacierの料金

バックアップなど中長期保存用のS3よりも安価なストレージ

容量当たりの料金	GB/月 あたり 0.005USD（0.5円ほど） →S3は標準で0.025USD／One zoneで0.0152USD/GB
データ取り出し料金	迅速：0.033USD/GB 標準：0.011USD/GB 大容量：0.00275USD/GB
データ取り出しリクエスト料金	迅速：11.00USD/リクエスト 1,000 件 標準：0.0571USD/リクエスト 1,000 件 大容量：0.0275USD/リクエスト 1,000 件
プロビジョニングされた迅速取り出し	110.00USD/プロビジョンド容量単位
データ転送料金	データ転送 (イン)は無料 インターネットへのデータ転送 (アウト)は1 GB/月まで無料。それ以上は有料

※2020年7月あたりのお値段です。値段は変動する可能性があります。

Glacier Deep Archive

Glacierよりも値段が安くデータ保存が可能だが、データ取得はさらに遅くなる中長期保存用ストレージタイプ

Glacierよりさらに値段が安い！

Glacierよりさらにデータ取得が遅い！

Glacier Deep Archive

Glacierよりも値段が安くデータ保存が可能だが、データ取得はさらに遅くなる中長期保存用ストレージタイプ

- ✓ 基本的なデータモデル・管理はGlacierと同じ
- ✓ 1 GB あたりの月額料金 0.00099 USD から利用可能でAWSの最低**価格**
- ✓ データは 3 つ以上の AWS アベイラビリティゾーンにまたがって保存され、S3と同様に99.999999999% の耐久性を実現
- ✓ 標準取り出しで、データは12 時間以内に取りだすことが可能
- ✓ 大量取り出しで、48 時間以内にデータを取り出す大容量取り出しをすることで取得コストを低減できる。

その他の出題範囲

AWS DataSyncの利用

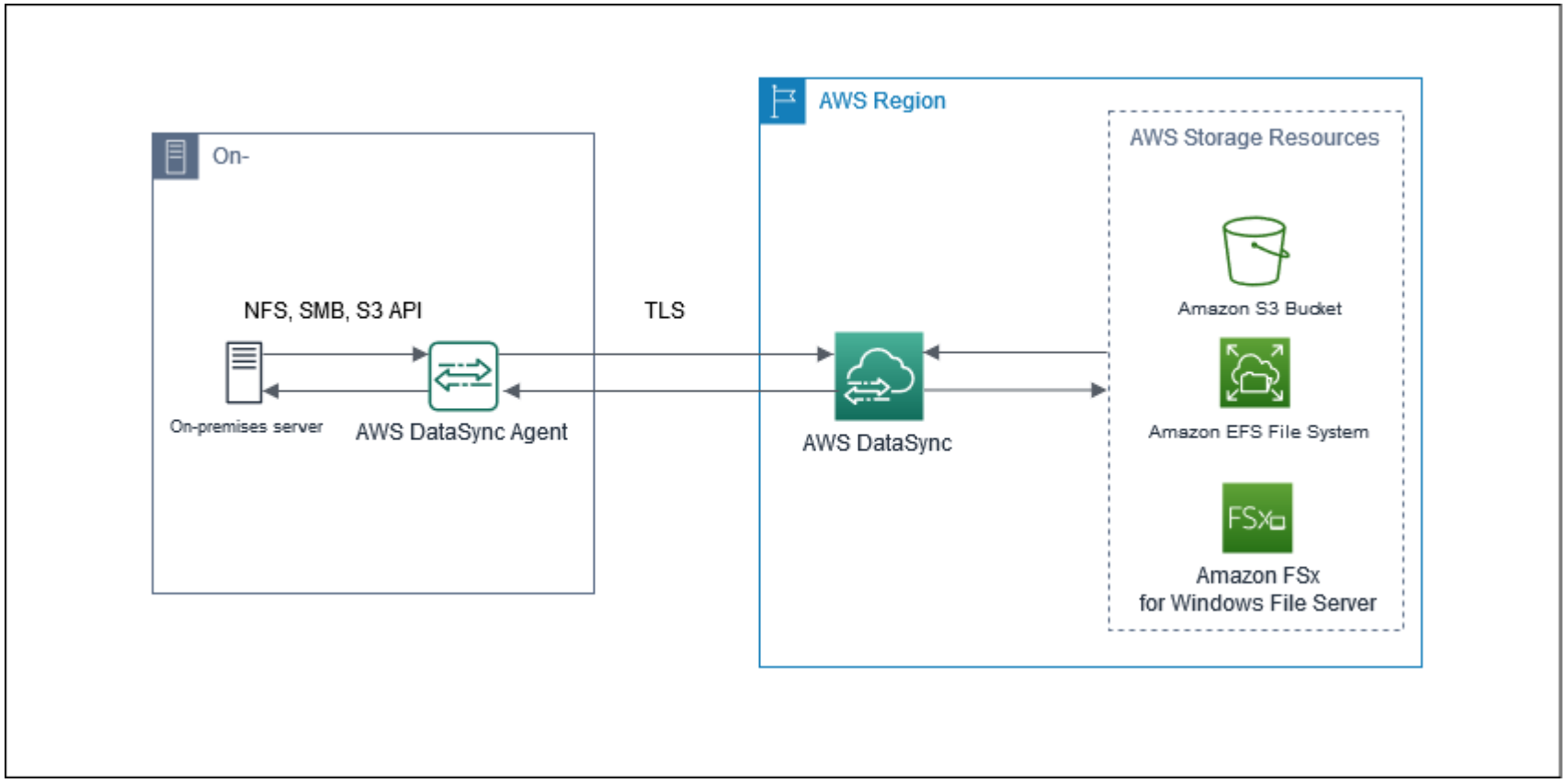
A社はAWSへとインフラストラクチャーを移行する決定をしました。あなたはソリューションアーキテクトとして、オンプレミスのほとんどデータをAmazon S3とAmazon EFSに移動する対応を行っています。これらのAWSストレージサービスへのオンラインデータ転送を自動化させる必要があります。

次のうち、最適なソリューションを選択してください。

1) AWS DataPipelineを使用して、特定のAWSストレージサービスへのオンラインデータ転送を自動化する。
2) AWS Snowball edgeを使用して、特定のAWSストレージサービスへのオンラインデータ転送を自動化する。
3) Amazon DMLを使用して、特定のAWSストレージサービスへのオンラインデータ転送を自動化する。
4) AWS DataSyncを使用して、特定のAWSストレージサービスへのオンラインデータ転送を自動化する。

AWS DataSyncの利用

AWS DataSyncはS3やEFSにストレージデータを移行する際に利用するサービス



Reference: https://docs.aws.amazon.com/ja_jp/datasync/latest/userguide/how-datasync-works.html

DR構成

金融サービス会社は、事業継続性計画の中で災害復旧プランを作成しています。あなたはソリューションアーキテクトとして、完全に機能する環境の縮小バージョンが常にAWSクラウドで実行されていることを確認し、災害が発生した場合の復旧時間を最小限に抑えたいと考えています。

この要件を満たすことができる方式はどれでしょうか？

1) ウォームスタンバイ
2) バックアップ＆リストア
3) パイロットライト
4) マルチサイト

DR構成

障害復旧対応向けの方法は用途に応じて様々な方式を準備する

ホットスタンバイ	◘ 本番サーバ機に問題が発生した際に、瞬時に予備サーバに切り替えられる用に稼働状態でスタンバイする方式
コールドスタンバイ	◘ 予備サーバとして必要なデータや機材を一通り事前に用意するが、稼働されていない状態でスタンバイする方式
ウォームスタンバイ	◘ 予備サーバーが本番サーバと同じ形で用意されているものの、切り替え（復旧）までに何らかの作業が必要
バックアップ＆リストア	◘ バックアップを定期的に実施して、本番機器に異常が発生した場合にリストアできるようにする方式
パイロットライト	◘ 停止した状態のサーバーを別のリージョンに用意しておき、障害発生時に立ち上げる。
マルチサイト	◘ マルチAZ構成やマルチリージョン展開などの複数サイトにインフラを構成する

DRにおけるリージョンの活用

金融サービス会社は、事業継続性計画の中で災害復旧プランを作成しています。あなたはソリューションアーキテクトとして、EC2インスタンス構成とRDSとをリージョンを跨いで災害対応ができるソリューションを検討しています。この対応はコスト最適が最も重用されます。

ディザスタリカバリ戦略としてどれを選択するべきでしょうか？（2つ選択してください。）

1) EC2インスタンスのAMIを作成して、別リージョンにコピーする。
2) EC2インスタンスのAMIを作成して、別リージョンにシェアする。
3) EC2インスタンスを別リージョンに移転する。
4) RDSを別リージョンに構成する。
5) RDSのスナップショットを作成して、別リージョンにコピーする。

DRにおけるリージョンの活用

AMIやスナップショットをコピーして別リージョンに取得する
別リージョンにストレージやDBをレプリケーションする