

# セクションの内容

レクチャー	レクチャーで学ぶ内容
IAMの出題範囲	AWSでユーザー管理を実施する主要サービスであるIAMにおける出題問題を確認して、その範囲の知識を詳細に学習します。
S3の出題範囲	AWSでストレージを構築する主要サービスであるS3における出題問題を確認して、その範囲の知識を詳細に学習します。
EC2の出題範囲	AWSで仮想サーバーを構築する主要サービスであるEC2における出題問題を確認して、その範囲の知識を詳細に学習します。
VPCの出題範囲	AWS内のネットワーク領域を切り出す主要サービスであるVPCにおける出題問題を確認して、その範囲の知識を詳細に学習します。

## IAMの出題範囲

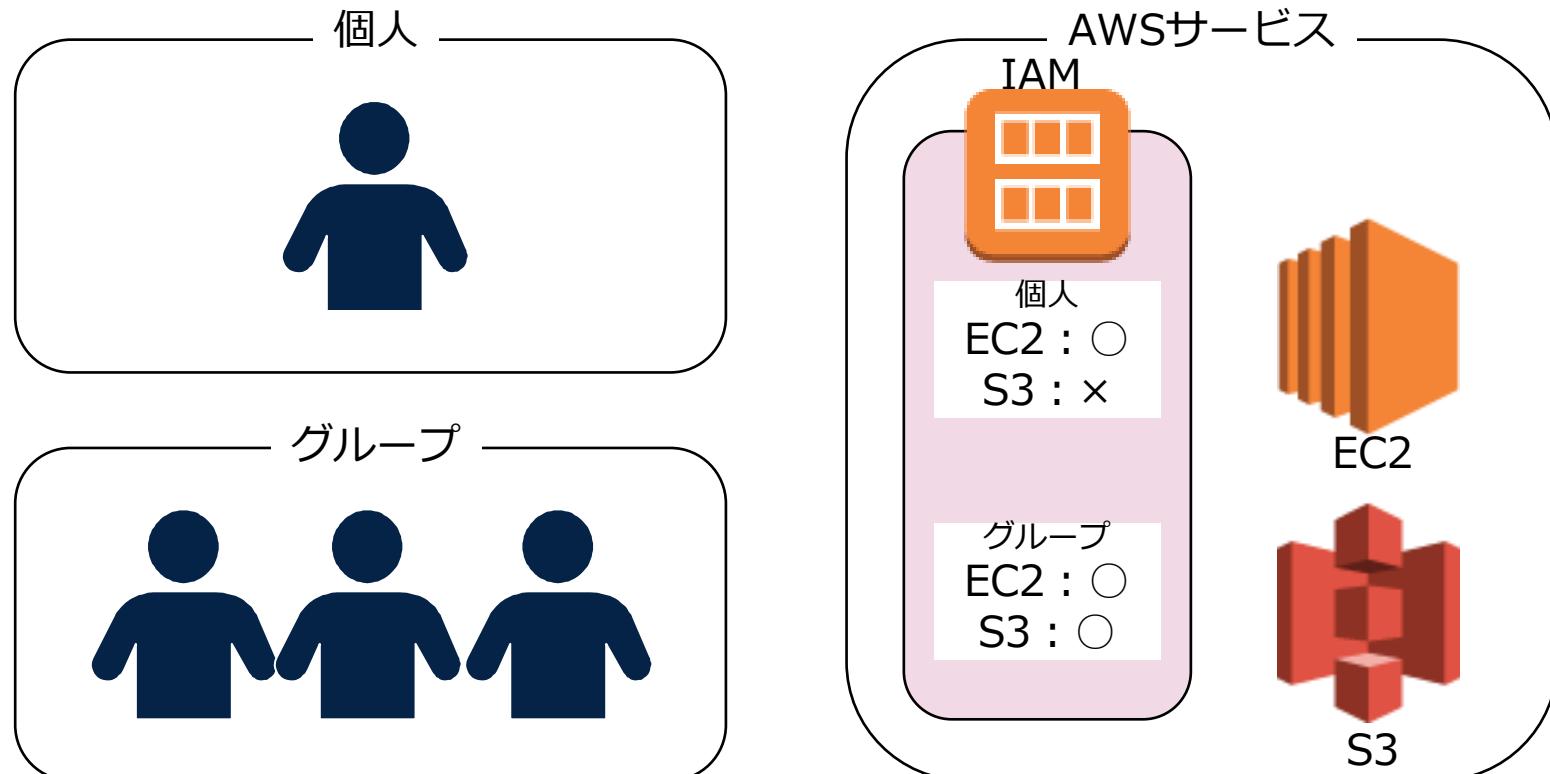
# IAMとは

AWS Identity and Access Management (IAM)は安全にAWS操作を実施するための認証・認可の仕組み

- AWS利用者認証の実施
- アクセスポリシーの設定
- 個人またはグループに設定

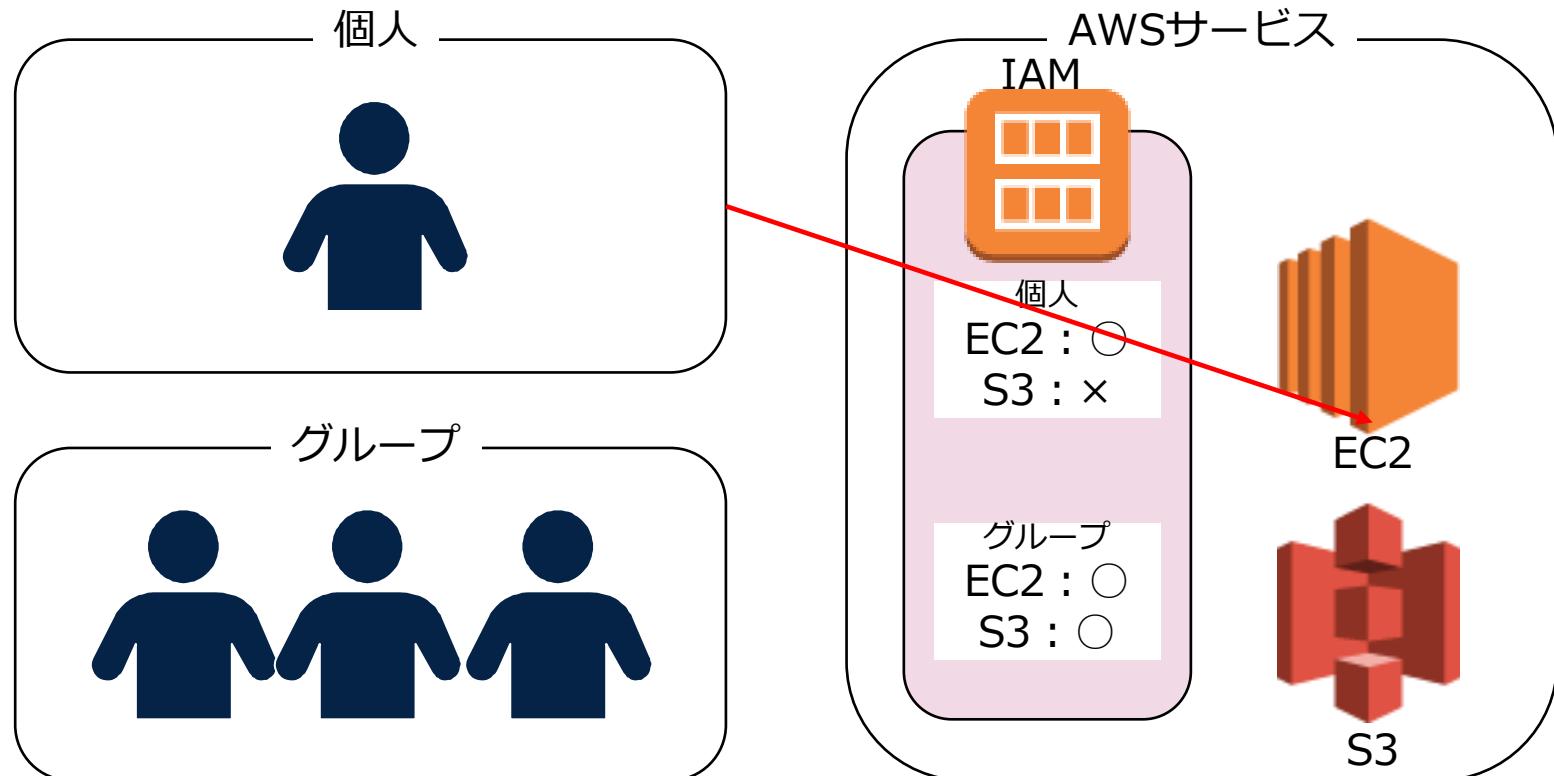
# IAMとは

AWS Identity and Access Management (IAM)は安全にAWS操作を実施するための認証・認可の仕組み



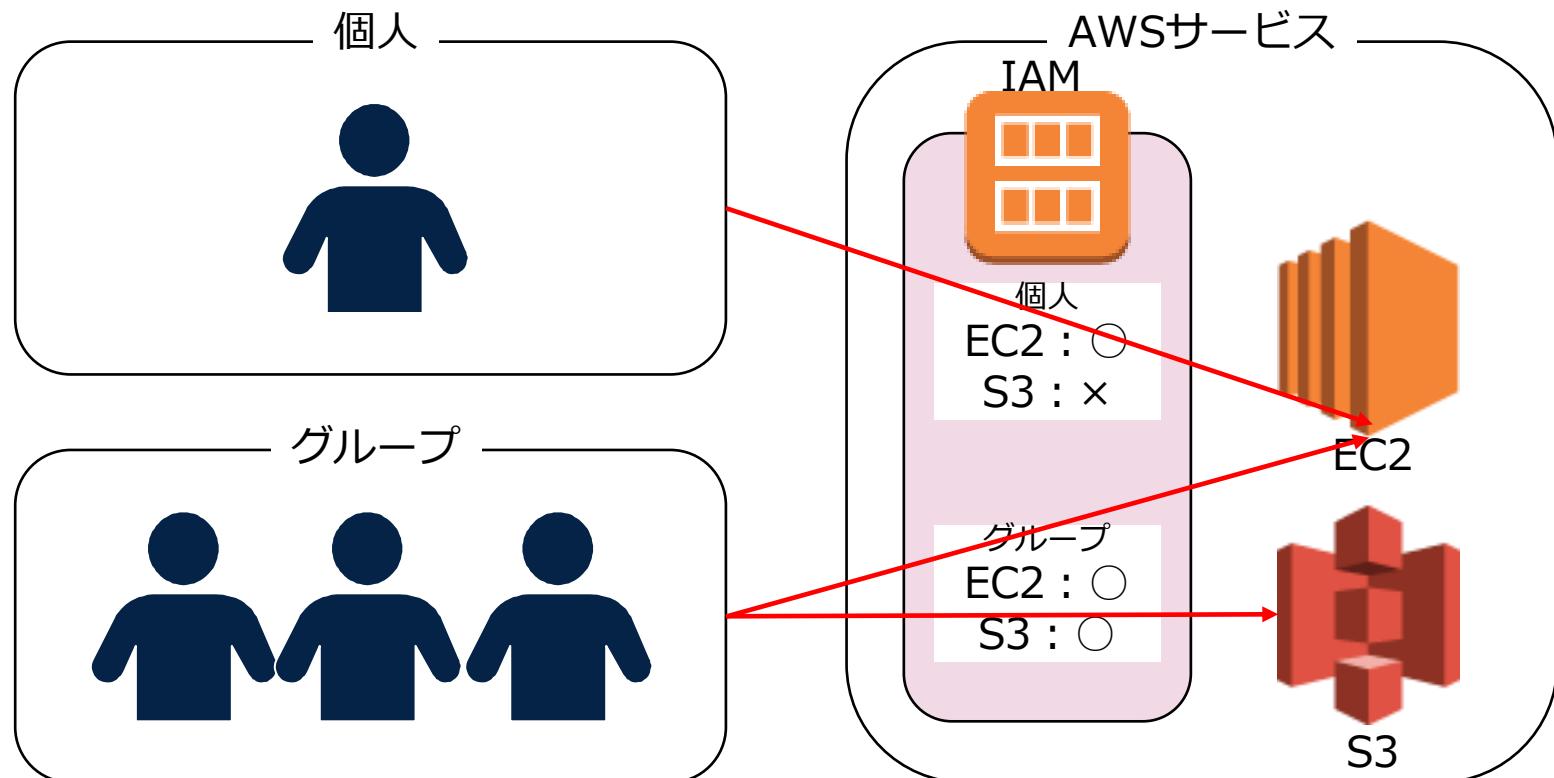
# IAMとは

AWS Identity and Access Management (IAM)は安全にAWS操作を実施するための認証・認可の仕組み



# IAMとは

AWS Identity and Access Management (IAM)は安全にAWS操作を実施するための認証・認可の仕組み



# IAMの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

IAMユーザー	✓ IAMユーザーの設定方法や利用方法が問われる。
ルートアカウント	✓ ルートアカウントの権限範囲が問われる。 ✓ ルートアカウントの利用制限に関するベストプラクティスが問われる。
IAMグループ	✓ IAMグループの利用目的や設定方法が問われる。
IAMポリシー	✓ IAMポリシーのドキュメントを読み込んで、その設定が示す許可状況が問われる。
IAMポリシーのタイプ	✓ IAMポリシーの各タイプの内容や利用目的が問われる。 ✓ また、MFAの利用、パスワードの強化などの基本的な推奨事項が問われる。

# IAMの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

IAMロールの設定	✓ IAMロールを設定する場合のケース内容が問われる。
IAMの認証方式	✓ アクセスキーとシークレットアクセスキーが必要な認証ケースが問われる。 ✓ MFAの認証が必要なベストプラクティスが問われる。
IAMデータベース認証	✓ 主にRDSの認証設定に使われる方式としてIAMデータベース認証の利用が問われる。
ユーザーのアクティビティの記録	✓ IAMユーザーのアクティビティなどの記録管理に利用されるツールの利用方法が問われる。
IAM権限のベストプラクティス	✓ 主に最小権限に基づいた権限設定などのベストプラクティスが問われる。

# IAMの主要機能

ユーザー、グループ、ポリシー、ロールを利用したアカウントやリソースのアクセス管理を実施する。

ユーザー	AWSアカウント内に追加するユーザー
グループ	ユーザーをグループ化してまとめる単位
ロール	リソースにアクセス権限を設定するもの
ポリシー	上記 3 つのエンティティに対して、権限を設定するドキュメント

# [Q] IAMポリシー

次のIAMポリシーでAWSリソースに対する権限設定を行っています。

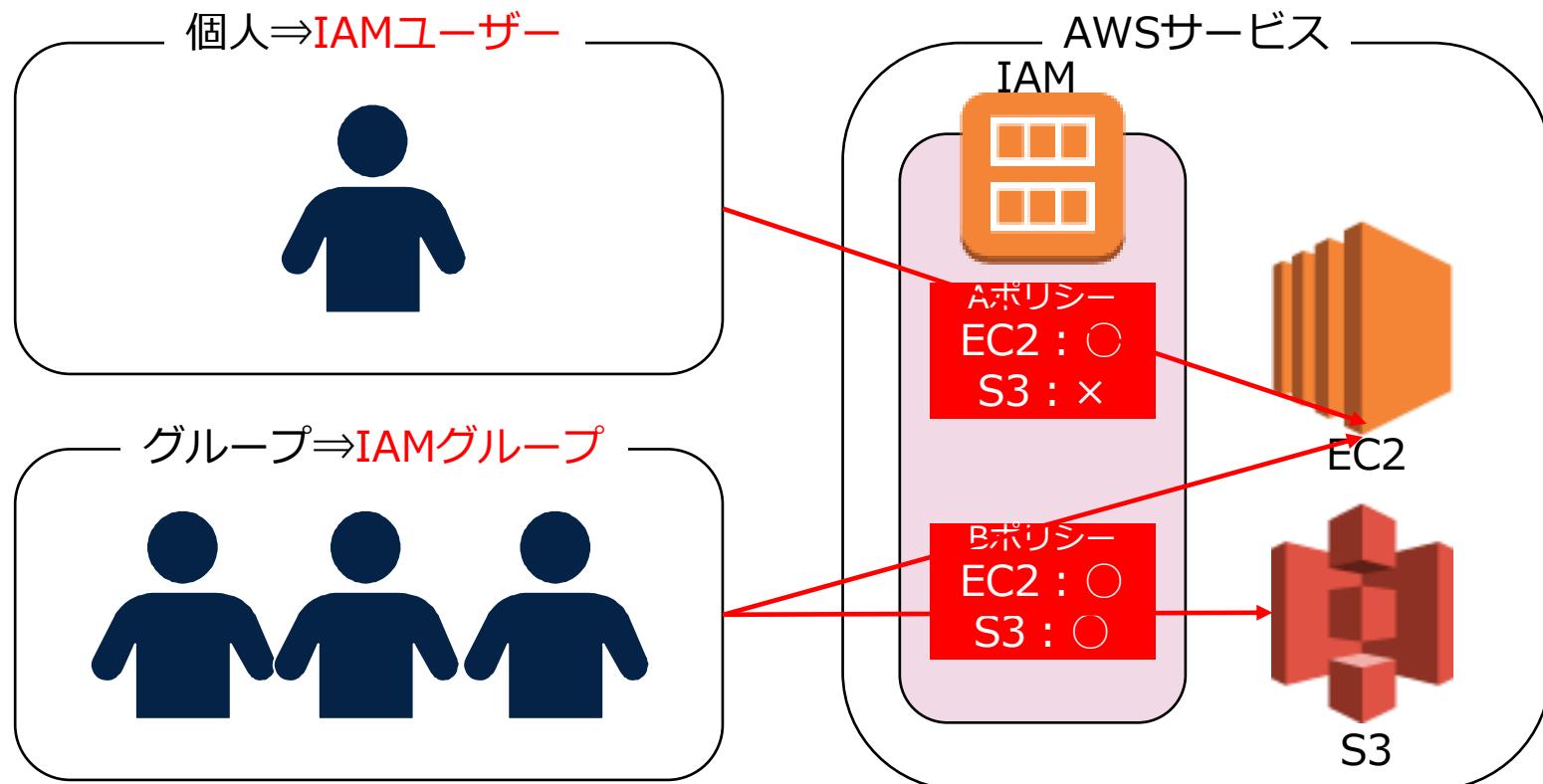
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "*",  
            "Resource": "*",  
            "Condition": {  
                "NotIpAddress": {  
                    "aws:SourceIp": [  
                        "172.103.1.38/24"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

この設定内容として正しい内容を選択してください。

- 1) IPアドレス（172.103.1.38）以外は全てのリソースのアクセス権限を拒否されている。
- 2) IPアドレス（172.103.1.0）は全てのリソースのアクセス権限を有している。
- 3) IPアドレス（172.103.1.3）は全てのリソースのアクセス権限を拒否されている。
- 4) IPアドレス（172.103.1.3）は全てのリソースのアクセス権限を有している。

# IAMポリシー

ユーザーなどへのアクセス権限を付与するための設定ドキュメントのこと（JSON形式の文書）



# IAMポリシー

IAMポリシーはJSON形式で設定される

{ "Effect": "Allow", "Action": [ "s3>ListBuckets", "s3:Get *" ], "Resource": [ "arn:aws:s3:::mybucket" ], "Condition": { "IpAddress": { "aws:SourceIP": ["176.32.92.49/32"] } } }	Effect	"Allow"⇒許可 "Deny"⇒拒否
	Action	対象のAWSサービス 例："s3:Get"
	Resource	対象のAWSリソース ARNで記述
	Condition	アクセス制御 が有効となる条件

# [Q] IAMユーザー

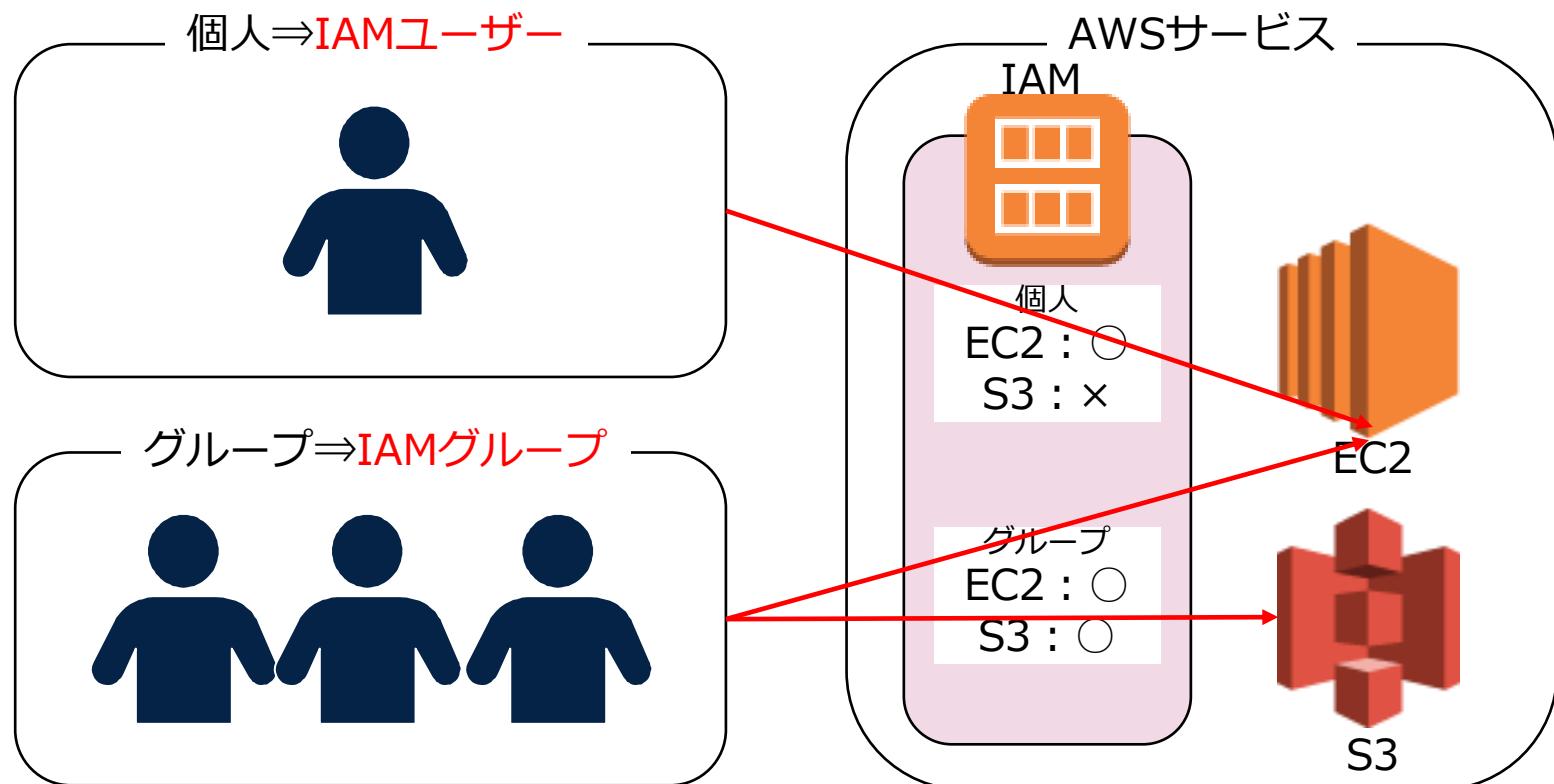
あなたはソリューションアーキテクトとして、部署内で新人のAWS担当者にAWSへのアクセス権限を設定しているところです。IAMユーザーを複数作成しましたが、作成されたIAMユーザーにはデフォルトでどのような権限が含まれているか確認することが必要です。

次の中でIAMユーザーのデフォルト権限として正しい説明はどれでしょうか？

- 1) 制限的な許可が設定されている。
- 2) 管理者権限以外のリソースへのアクセス許可が設定されている。
- 3) 何も権限を有していない。
- 4) デフォルトで基本リソースへの許可設定が付与されている。

# IAMユーザー

IAMポリシー内でAWSサービスを利用するユーザー。基本操作はIAMユーザーで実施することになる



# IAMユーザー

AWS上の利用者はIAMユーザーという権限を付与されたエンティティとして設定される。

ルートユーザー (IAMではない)	<ul style="list-style-type: none"><li>• AWSアカウント作成時に作られるIDアカウント</li><li>• 全てのAWSサービスとリソースを使用できる権限を有する</li><li>• 日常的なタスクはルートユーザーを使用しないことが強く推奨される</li></ul>
管理者権限 ( IAMユーザー )	<ul style="list-style-type: none"><li>• 管理者権限の許可が付与されたIAMユーザーのこと</li><li>• IAMの操作権限まであり。</li><li>• ルートアカウントしかできない権限は付与されない。</li></ul>
パワーユーザー (IAMユーザー)	<ul style="list-style-type: none"><li>• パワーユーザーはIAM以外の全てのAWSサービスにフルアクセス権限を有するIAMユーザー</li><li>• IAMの操作権限なし</li></ul>

# [Q]ルートアカウント

あなたは新規にAWSの利用を開始しました。AWSにアカウント登録するとルートアカウントと呼ばれるアカウントが1つ作成されて、AWS操作を実行できます。AWSではIAM管理者権限を有するIAMユーザーを利用して管理を実施することが推奨されていますが、ルートアカウントでしか実行できない操作があるようです。

ルートアカウントのみに実施可能な対応を選択してください。（2つ選択してください。）

- 1) IAMユーザーに対して課金情報へのアクセス許可を設定する。
- 2) AWSアカウント内のユーザー管理を実施する。
- 3) Route53を利用したドメイン登録を実施する。
- 4) AWSサポートへの連絡を行う。
- 5) AWS Organizationsにおいてメンバーアカウントになれる。

# ルートアカウント

ルートアカウント（ルートユーザー）にしかできない操作権限が存在する

## 【ルートユーザーのみの実施権限】

- AWSルートアカウントのメールアドレスやパスワードの変更
- IAMユーザーの課金情報へのアクセスに関するactivate/deactivate
- 他のAWSアカウントへのRoute53のドメイン登録の移行
- AWSサービス（サポート等）のキャンセル
- AWSアカウントの停止
- コンソリディテッドビリングの設定
- 脆弱性診断フォームの提出
- 逆引きDNS申請

# [Q] IAMグループ

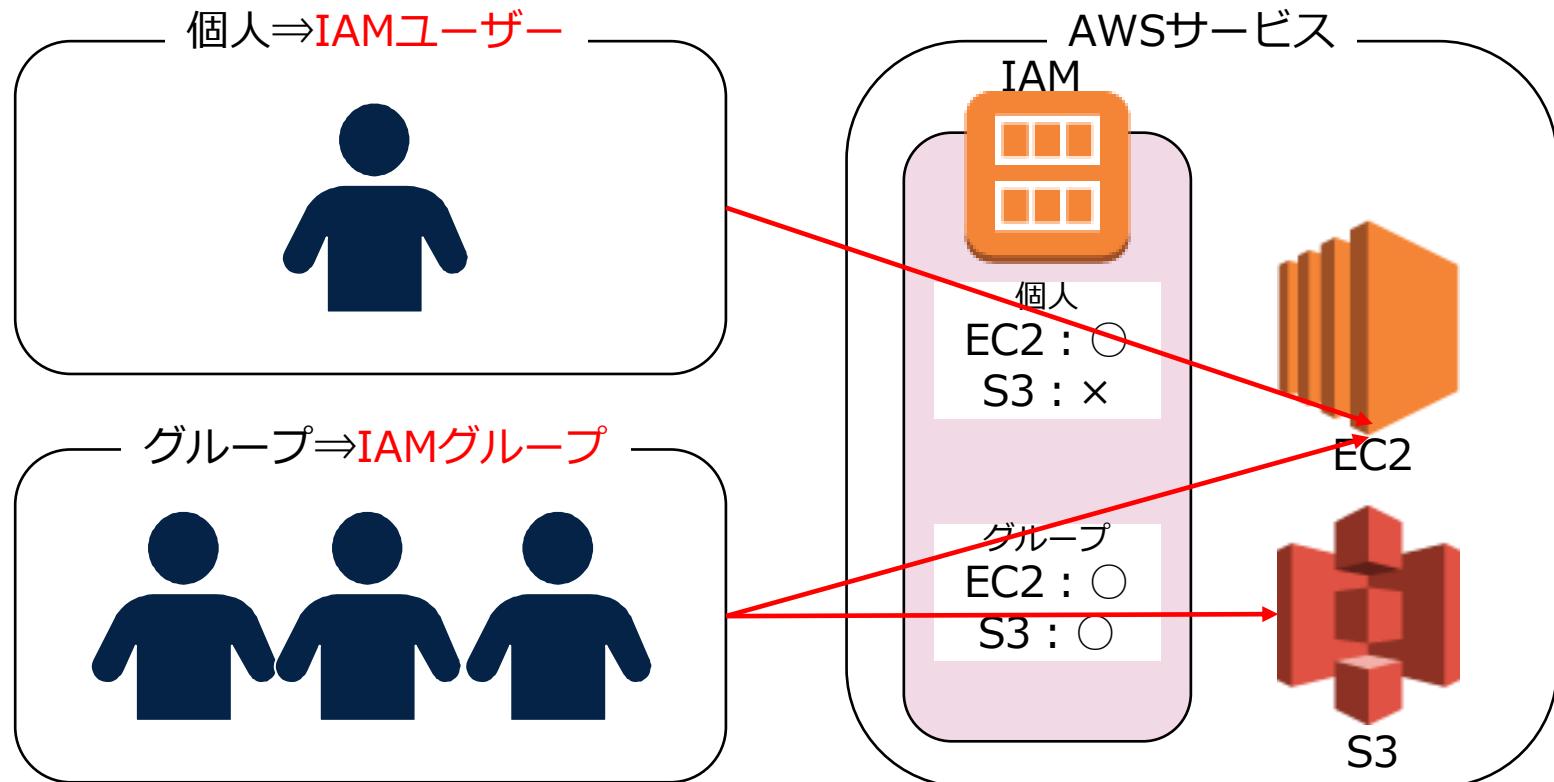
会社は300人以上のAWS利用者を設定することが必要です。これらのユーザーは3つの部署に分かれており、各部署の担当部門ごとに利用するAWSリソースが異なります。あなたはソリューションアーキテクトとして、これらのユーザーへの最適な権限設定を検討するように依頼されました。

最小権限の原則に基づいて、どのように権限を設定するべきでしょうか？

- 1) 各ユーザーに必要な最小権限を設定したIAMポリシーを作成して、IAMユーザーに設定する。
- 2) 各ユーザーに必要な最小権限を設定したIAMポリシーを作成して、IAMグループに設定する。IAMユーザーを各IAMグループに配置する。
- 3) 各ユーザーに必要な最小権限を設定したIAMポリシーを作成して、IAMユーザーに設定する。さらに、これらのIAMユーザーをIAMグループに配置する。
- 4) 各部署に対してIAMグループを作成して、各ユーザーに必要な最小権限を設定したIAMポリシーを設定する。

# IAMグループ

グループとして権限をまとめて設定された単位のこと。グループには通常は複数のIAMユーザーが設定される



## [Q]IAMロール

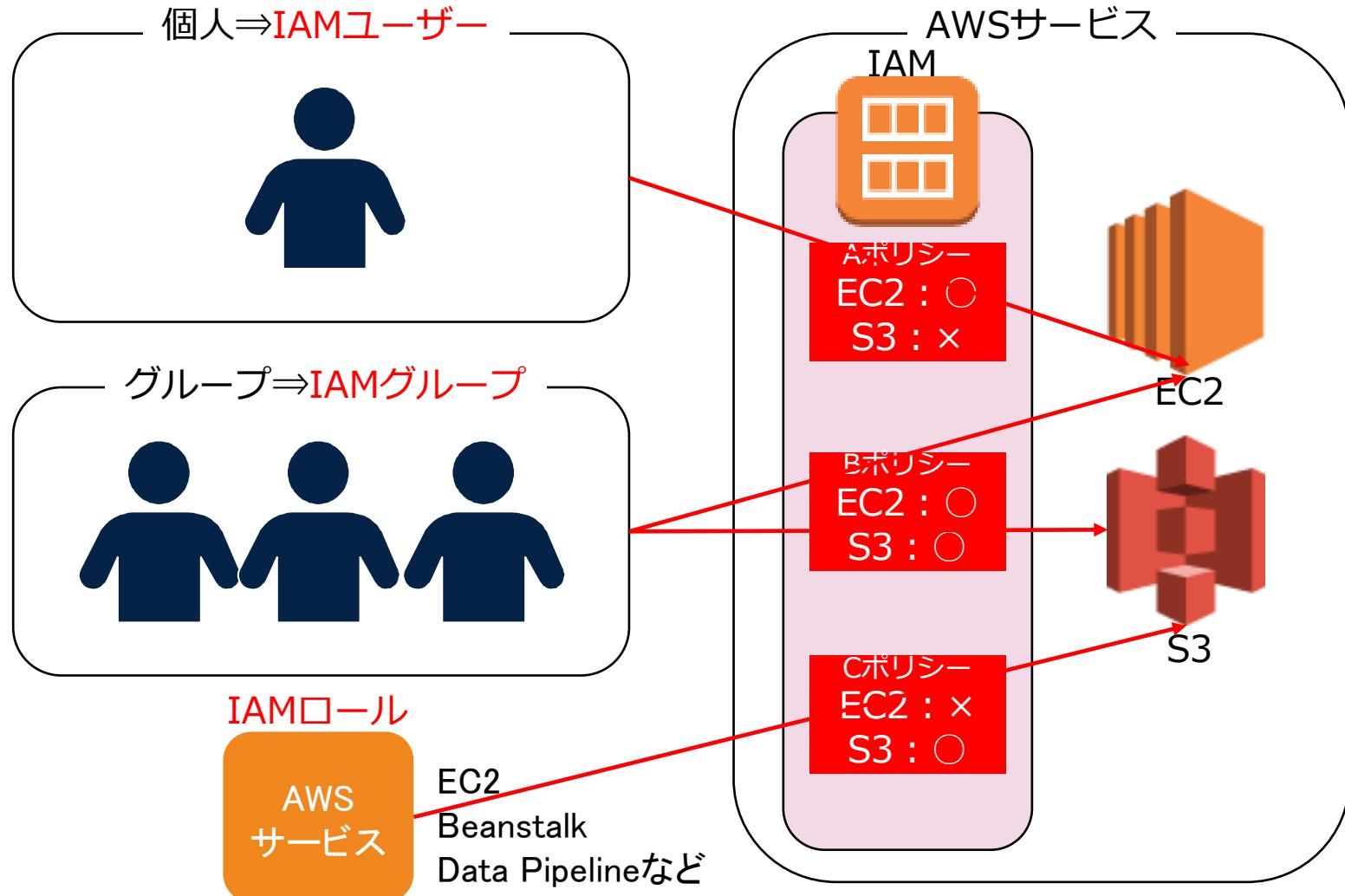
ソリューションアーキテクトは、Lambda関数を利用したデータベースオペレーションを実行するアプリケーションを構築しています。このサーバレスアプリケーションは、Amazon DynamoDBテーブルにアクセスして、データを取得して加工する処理を実行します。

Lambda関数にDynamoDBテーブルへのアクセスを許可する最も安全な手段は何ですか？

- 1) DynamoDBテーブルにアクセスするために必要な権限を持つIAMロールを作成し、そのロールをLambda関数に割り当てる。
- 2) DynamoDBテーブルにアクセスするために必要な権限を持つIAMポリシーを作成し、そのポリシーをLambda関数に割り当てる。
- 3) DynamoDBテーブルにアクセスするために必要な権限を持つIAMグループを作成し、そのグループをLambda関数に割り当てる。
- 4) DynamoDBテーブルにアクセスするために必要な権限を持つIAMユーザーを作成し、そのユーザーをLambda関数に割り当てる。

# IAMロール

AWSリソースに対してアクセス権限をロールとして付与できる



## [Q] IAMポリシーのタイプ

大手IT企業では、1つのAWSアカウントを利用して開発者グループにパワーアクセス権限のあるユーザーを用意して開発を進めていました。しかしながら、1人の開発担当者が本番環境にあるRoute53を不用意に削除したことで、重要なアプリケーションが長時間ダウンするトラブルが発生してしまいました。このインシデントが報告された後、セキュリティのベストプラクティスによる制御を実施するように依頼されました。各グループごとにIAM管理者が権限管理をしているため、あらかじめグループ毎に付与される権限を制限することが必要です。

このようなインシデントが再発しないように適切な対応を選択してください。

- 1) ルートアカウントを利用して、管理者権限をルートアカウントのみに制限する。
- 2) SCPを利用して、開発担当者がIAMアイデンティティに付与できる最大権限を制御する。
- 3) IAMグループを利用して、開発者グループの担当者の権限設定を制限する。
- 4) アクセス許可の境界を使用して、開発担当者のIAMアイデンティティに付与できる最大権限を制御する。

# IAMポリシーのタイプ

IAMポリシーはユーザーベースのポリシーと呼ばれるポリシーであり、他にも多数のポリシーが存在する。

## ユーザーベースの ポリシー

- ✓ 管理ポリシーとインラインポリシーを IAM エンティティ(ユーザー、ユーザーのグループ、ロール) にアタッチされるポリシー。
- ✓ ユーザー ポリシーのアクセス許可はエンティティに付与される。

## リソースベースの ポリシー

- ✓ バケットポリシーなどの JSON 形式のドキュメントで定義されたインラインポリシーをリソースにアタッチするポリシー
- ✓ 例は Amazon S3 バケットポリシー や IAM ロールの信頼ポリシー

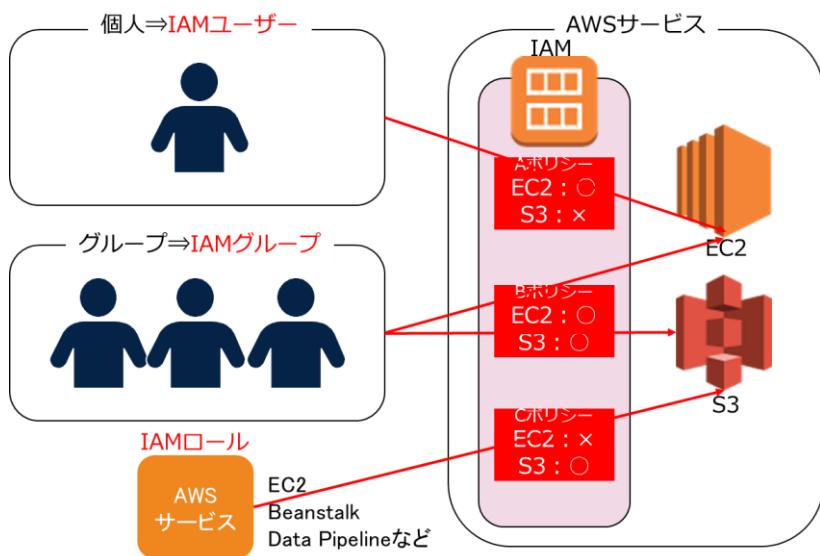
## アクセス許可 の境界

- ✓ アクセス許可の境界はユーザーベースポリシーが IAM エンティティ に付与できるアクセス許可の上限を設定する。アクセス許可自体は付与しない。
- ✓ IAM エンティティはユーザーベースポリシーとアクセス許可境界の両方で許可されているアクションのみ許可される。

# ユーザーベースとリソースベース

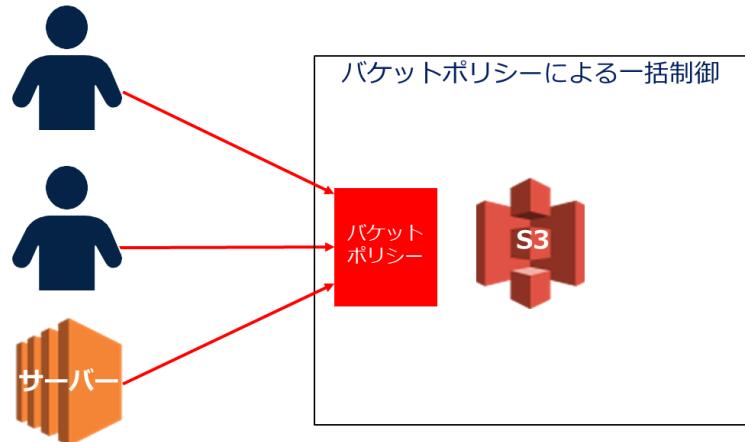
ユーザーやリソースにIAMポリシーで許可を与えるのがユーザーベースのポリシー。リソース側の機能で制御するのがリソースベースのポリシー。

ユーザー側のポリシーでアクセスを管理



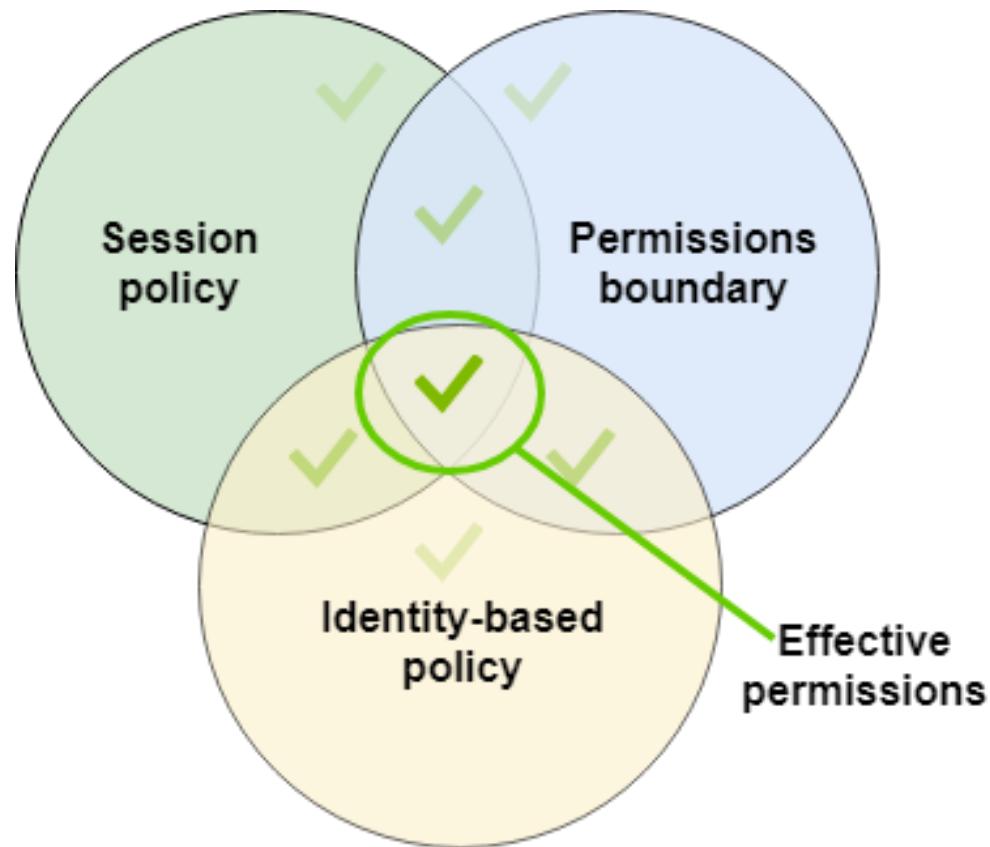
リソース側のポリシーでアクセスを管理

S3バケットポリシーによるアクセス制御



# アクセス許可の境界

アクセス許可の境界で許可設定の上限を設定してから、他のポリシーで許可を設定することができる。



# IAMポリシーのタイプ

IAMポリシーはユーザーベースのポリシーと呼ばれるポリシーであり、他にも多数のポリシーが存在する。

## SCP

- ✓ 組織または組織単位 (OU) のメンバーアカウントのアクセス許可の上限を定義するポリシー
- ✓ アクセス許可の境界と同様にこれ自体はアクセス許可は付与しない。
- ✓ メンバーアカウント内のIAMユーザーはSCPとIAMポリシーの両方で許可されているアクションのみ実行できる。

## ACL

- ✓ ACL がアタッチされているリソースへのアクセス許可・拒否を制御
- ✓ JSON ポリシードキュメント構造を使用しない点がリソースベースのポリシーと異なる。
- ✓ プリンシパルにアクセス許可を付与するクロスアカウントのアクセス許可ポリシー

## セッションポリシー

- ✓ ロールまたはフェデレーティッドユーザーの一時セッションをプログラムで作成する際にパラメータを渡す機能
- ✓ 作成したセッションのアクセス許可が制限されますが、アクセス許可は付与されない。

## [Q] ユーザーベースのポリシータイプ

あなたはソリューションアーキテクトとして、AWSを利用したアカウント管理を実施しています。まずはIAMユーザーを作成して、AWS利用者へのアカウント権限を発行する必要があります。そのためには、AWSの管理者となる2人のユーザーに対して、管理者権限のIAMポリシーが必要です。

この権限管理を実施するために、最も容易に利用できるIAMポリシータイプを選択してください。

- 1) AWS管理ポリシーの管理者権限を利用する
- 2) インラインポリシーを利用する
- 3) サードパーティのポリシーを利用する
- 4) カスタマー管理ポリシーを利用する

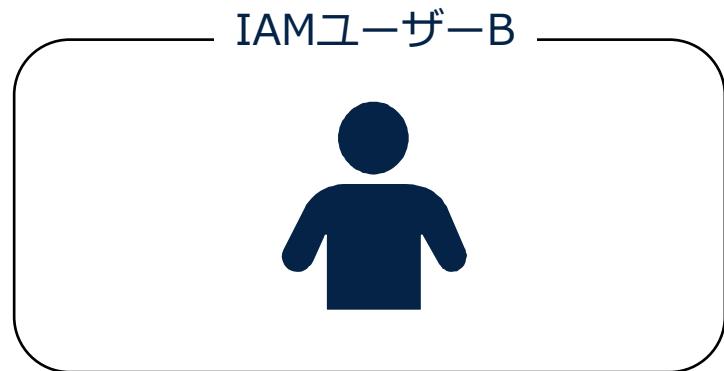
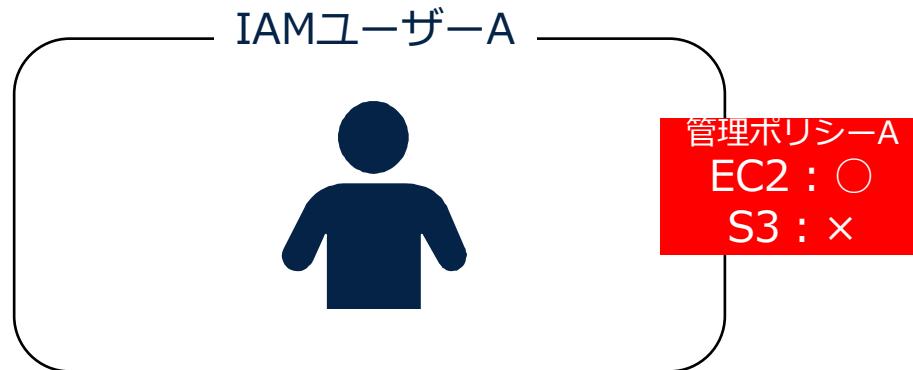
# ユーザーベースのポリシータイプ

AWSが用意するAWS管理ポリシーとユーザーが設定するカスタマーマネジメントポリシーと、再度転用ができないインラインポリシーがある。

AWS管理ポリシー	AWSが作成および管理する管理ポリシー 管理者権限やパワーユーザー権限などの標準的な設定のポリシーが用意されている。
カスタマーマネジメントポリシー	AWSアカウントが独自に作成する管理ポリシー。 インラインポリシーと異なり、同じポリシーを複数のIAMエンティティに何度も利用できる。
インラインポリシー	一つのプリンシパルエンティティ（ユーザー、グループ、またはロール）に直接組み込むポリシー 何度も利用できない。

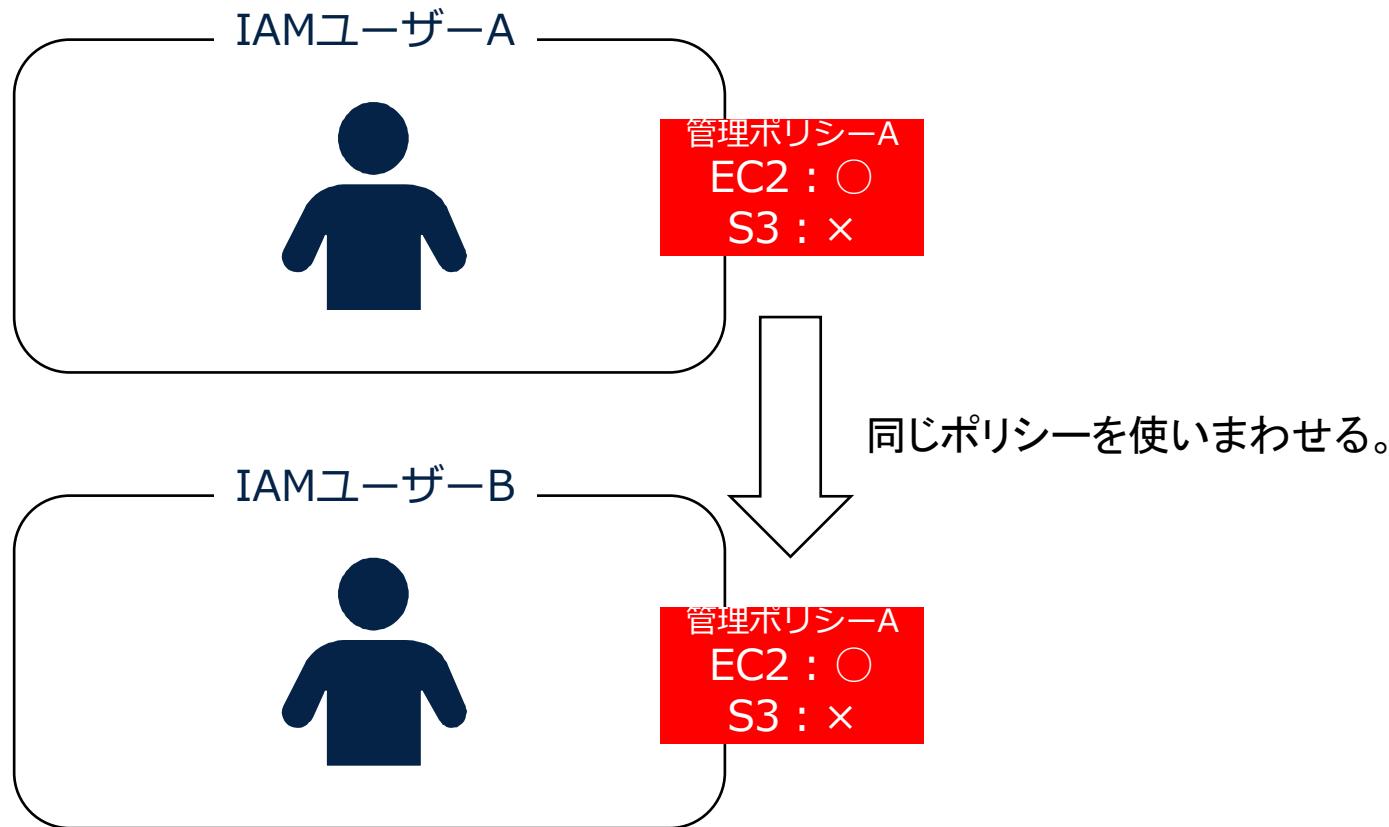
# 管理ポリシー

管理ポリシーは1つのポリシーを様々なユーザーやリソースに使いまわせるポリシー



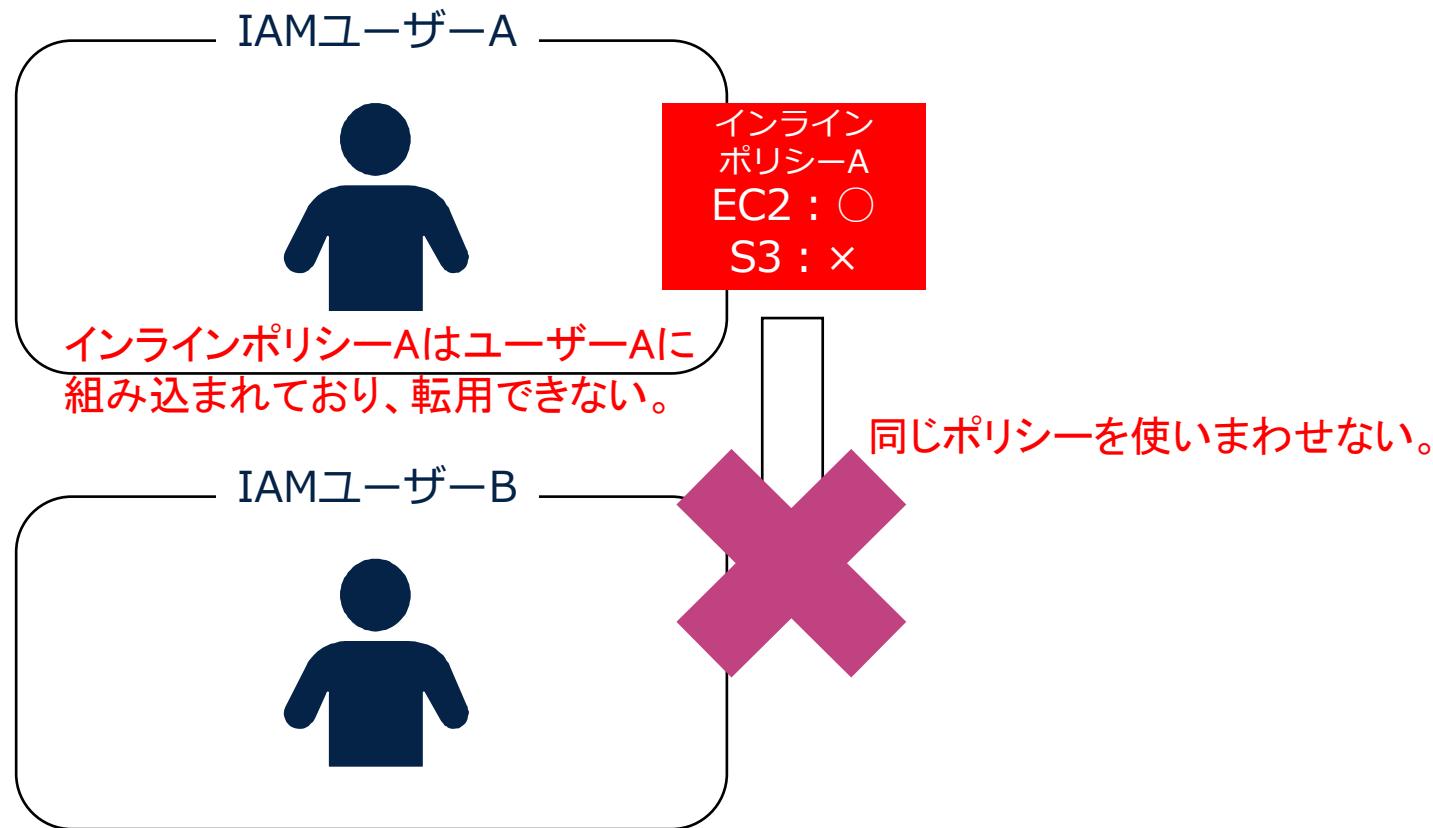
# 管理ポリシー

管理ポリシーは1つのポリシーを様々なユーザーやリソースに使いまわせるポリシー



# インラインポリシー

インラインポリシーは他のユーザーやリソースに使いまわせない、特定のユーザーやリソースに組み込まれるポリシー。



# [Q] IAMロールの信頼ポリシー

あなたの会社はAWSを利用して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にベンダーA社のソリューションを組み込むために、その会社の担当者に対して一時的なアクセスが必要です。あなたはソリューションアーキテクトとして、この担当者にアクセス権限を委任して、一部のリソースにアクセスできるようにしたいと考えています。

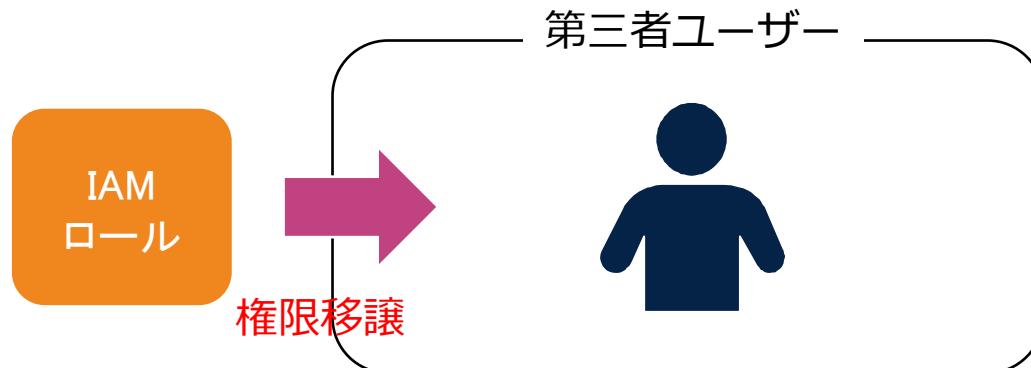
ソリューションアーキテクトとして、次のどの対応を実施するべきでしょうか？

- 1) 新しいIAMユーザーを作成して、必要なAWSリソースへの権限を設定した上で、A社の担当者に付与する。
- 2) 一時認証用の仕組みであるSTSを作成して、必要なAWSリソースへの権限を設定した上で、A社の担当者に付与する。
- 3) アクセスキーを発行して、必要なAWSリソースへの権限を設定した上で、A社の担当者に付与する。
- 4) 新しいIAMロールを作成して、必要なAWSリソースへの権限を設定した上で、A社の担当者に付与する。

# IAMロールの信頼ポリシー

IAMロールは監査人などに一時的な権限を委譲する際にも利用される。

- ✓ IAMロールの権限移譲操作に特化したポリシー
- ✓ この信頼ポリシーに関連づけられたIAMロールが保有する権限を、操作主体であるPrincipalに移譲(許可権限の貸与)することができる。



# [Q] IAMの認証方式

あなたはソリューションアーキテクトとして、EC2インスタンスベースのWEBアプリケーションをAWS上で開発しています。このアプリケーションはRestful APIを使用してAWS Rekognitionsを呼び出して画像処理を実施します。したがって、APIによってAWSリソースに直接アクセスして、連携する機能が必要です。

アプリケーションがAPIで連携するための必要な認証方法を選択してください。

- 1) 開発者のIAMユーザーのIDとパスワードを利用して、API連携を認証する。
- 2) EC2インスタンスのIAMロールのIDとパスワードを利用して、API連携を認証する。
- 3) 開発者のIAMユーザーのアクセスキーとシークレットアクセスキーを利用して、API連携を認証する。
- 4) Cognitoをアプリケーション上に実装して、STSを発行して連携する。

# ユーザー認証方式

ユーザー認証ではパスワード認証またはアクセスキーを利用した認証が行われ、追加でMFAも設定できる。

## パスワード認証

- ユーザーがAWSアカウントやIAMユーザーにログインする際に利用する。

## MFA(多要素認証)

- 物理デバイスなどを利用したピンコードによる追加の認証方式。ルートアカウントなどはMFAを付与してセキュリティを強化することが推奨される。

## アクセスキーとシークレットアクセスキー

- AWS CLI やAPI利用する際などのツールやアプリケーション経由の認証に利用する。

# パスワード認証

ルートアカウントはメールアドレスとパスワードでログインする。IAMユーザーはユーザーIDとユーザー名とパスワードを利用してコンソールにログインする。

## パスワードを取得

以下のユーザーのパスワードを表示およびダウンロードするか、AWS マネジメントコンソールにサインインするための手順を E メールでユーザーに送信できます。これは、このパスワードを表示およびダウンロードできる唯一の機会です。

### コンソールサインインの詳細

E メールでのサインイン手順 

#### コンソールサインイン URL

<https://udemy201111111.signin.aws.amazon.com/console>

ID

#### ユーザー名

udemy

#### コンソールパスワード

\*\*\*\*\* [表示](#)

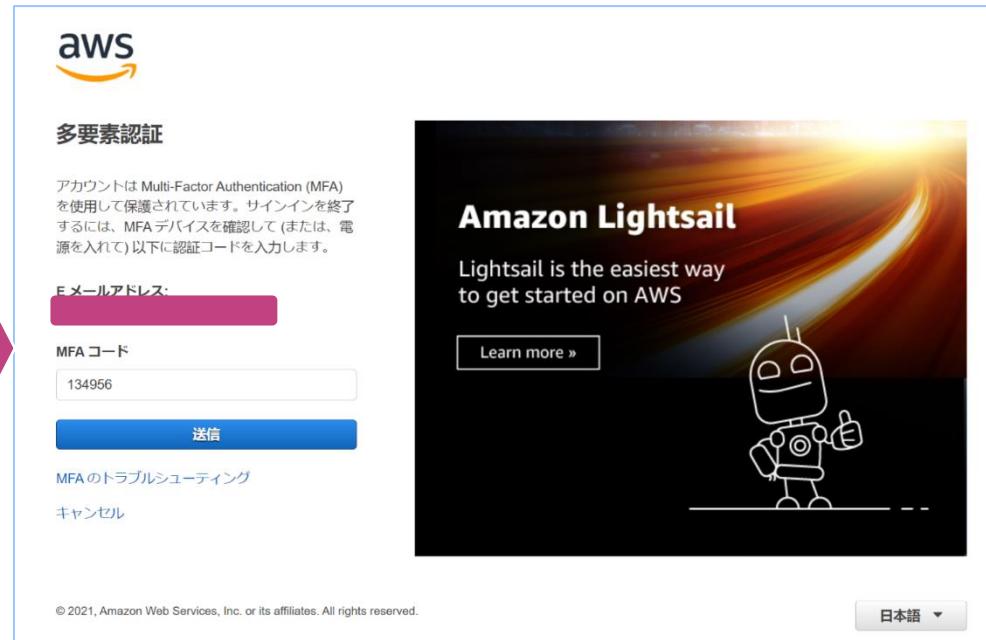


# 多要素認証 (MFA)

MFAはパスワード認証に追加して、ピンコードや顔認証などの多要素の認証を必要とする方法。



The screenshot shows the AWS sign-in interface. It starts with a 'サインイン' (Sign In) screen where the user selects 'ルートユーザー' (Root User). Below this, there's a placeholder for an IAM user. The user has entered their email address 'username@example.com'. A large red arrow points from this screen to the next one.



The second screenshot shows the '多要素認証' (Multi-Factor Authentication) step. It explains that MFA is used for account protection. The user is prompted to enter an MFA code sent via email. The code '134956' is entered in the field, and a '送信' (Send) button is visible. Below the form, there are links for 'MFA のトラブルシューティング' (MFA troubleshooting) and 'キャンセル' (Cancel). The footer includes copyright information and a Japanese language selection option.



# アクセスキーとシークレットアクセスキー

MFAはパスワード認証に追加して、ピンコードや顔認証などの多要素の認証を必要とする方法。

## アクセスキー

シークレットアクセスキーを紛失または失念した場合、それを取得することはできません。代わりに、新しいアクセスキーを作成し、古いキーを非アクティブにします。

### アクセスキー

AKIA4Q3XANMPS55OPU62

### シークレットアクセスキー

RW4Mb6VgisAcGGhr2Uj2livhw+iyVe4pVFSupdwA 非表示



```
[root@awscli ~]#  
[root@awscli ~]# aws  
aws                  aws.cmd          aws_zsh_completer.sh  
aws_bash_completer  aws_completer  
[root@awscli ~]# ls -ltrah .aws/c  
config      credentials  
[root@awscli ~]# cat .aws/credentials  
[default]  
aws_access_key_id = AKIAJJKK7VB6TXPTOHNNTA  
aws_secret_access_key = t5DSysqgCfee26s+nmUa9wu12kMnTWBsdk2C/LeR  
[root@awscli ~]# cat .aws/config  
[default]  
region = us-west-2  
#region = ap-south-1  
[root@awscli ~]#
```



## [Q] IAMデータベース認証

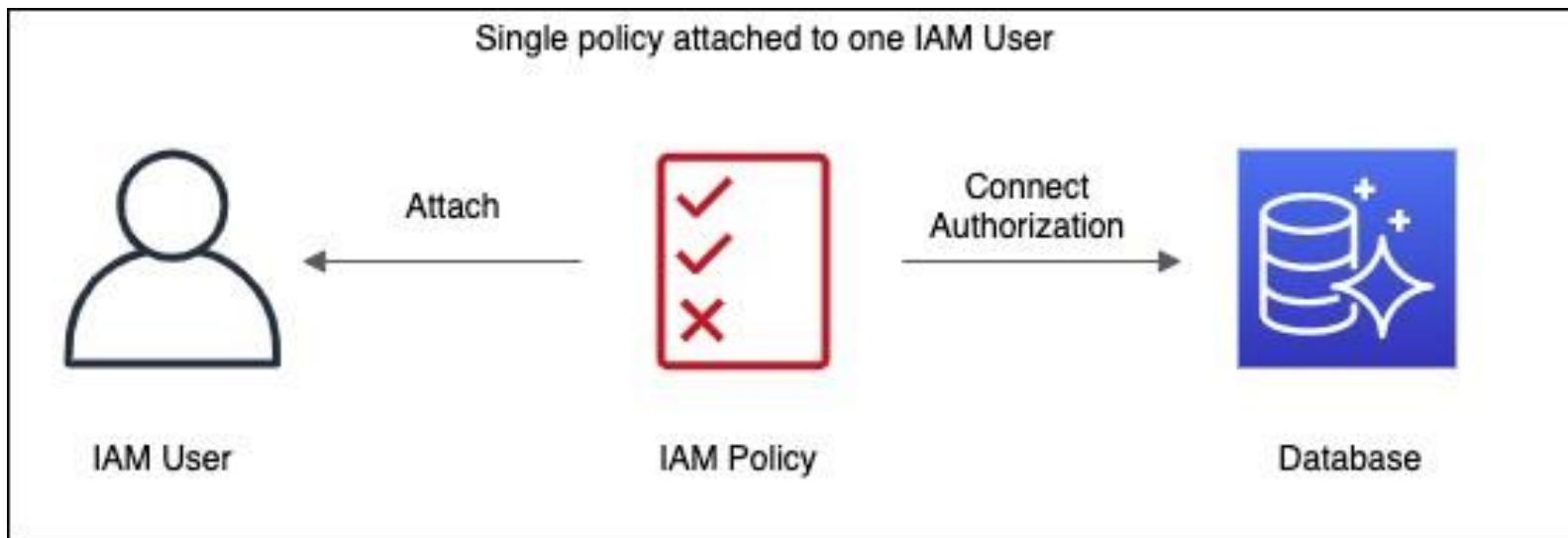
あなたの会社ではAWSを利用したデータベースソリューションを構築しており、複数のAmazon RDS MySQLデータベースを利用しています。現在、ユーザーIDによるMySQL認証方式を利用してEC2インスタンスベースのアプリケーションからデータベースに接続をしていますが、コード上でパスワードを実行する方式はセキュリティ上の懸念があります。

セキュリティを向上させる、より安全な認証方法はどれでしょうか？

- 1) Amazon RDS MySQLデータベースが有するIAMデータベース認証を利用して、EC2インスタンスへのアクセスを許可する。
- 2) AWS STSを使用して、EC2インスタンスからMySQLデータベースへのアクセスを許可する。
- 3) AUTHコマンドをEC2インスタンス上で実行して、データベース認証を実行する。
- 4) IAMロールを利用した一時的な認証をアプリケーション上で実行する。

# IAMデータベース認証

IAM DB認証を利用してIAM ユーザーまたはIAMロール認証と認証トークンを使用して Amazon RDS DBに接続可能  
(通常DBはユーザーIDとパスワードで認証する)



Reference: <https://aws.amazon.com/jp/blogs/news/using-iam-authentication-to-connect-with-pgadmin-amazon-aurora-postgresql-or-amazon-rds-for-postgresql/>

# [Q]ユーザーのアクティビティの記録

あなたの会社では一部のS3バケットについてIAMポリシーによって、外部の第三者のアプリケーションによるファイル読み込みを許可しています。したがって、これらのアクセスが想定された外部利用者に正しく利用されており、想定外の利用がされていないかを確認することが必要です。

この確認する仕組みとして最適な方法を選択してください。

- 1) IAM Access Advisorを利用して、信頼ゾーン外からのアクセスを検証して、不正なアクセスがあれば権限設定を見直す。
- 2) IAM Policy Simulatorを利用して、信頼ゾーン外からのアクセスを検証して、不正なアクセスがあれば権限設定を見直す。
- 3) AWS Configを利用して、信頼ゾーン外からのアクセスを検証して、不正なアクセスがあれば権限設定を見直す。
- 4) IAM Access Analyzerを利用して、信頼ゾーン外からのアクセスを検証して、不正なアクセスがあれば権限設定を見直す。

# ユーザーのアクティビティの記録

目的に応じて様々なツールを利用して記録を取得できる。

## IAMアクセス アナライザー

S3 バケットや IAM ロールなどリソースベースのポリシーを確認して、信頼ゾーンの外からのアクセスの有無を特定する。

## IAMアクセス アドバイザー

IAMユーザーのアクセス可能なリソースと最終アクセス日時を確認することができる。

## Credential Report

すべてのユーザーの認証情報が記載されたレポート

## AWS Config

AWS ConfigはIAMのユーザー、グループ、ロール、ポリシーの変更履歴、構成変更を管理するサービス

## AWS CloudTrail

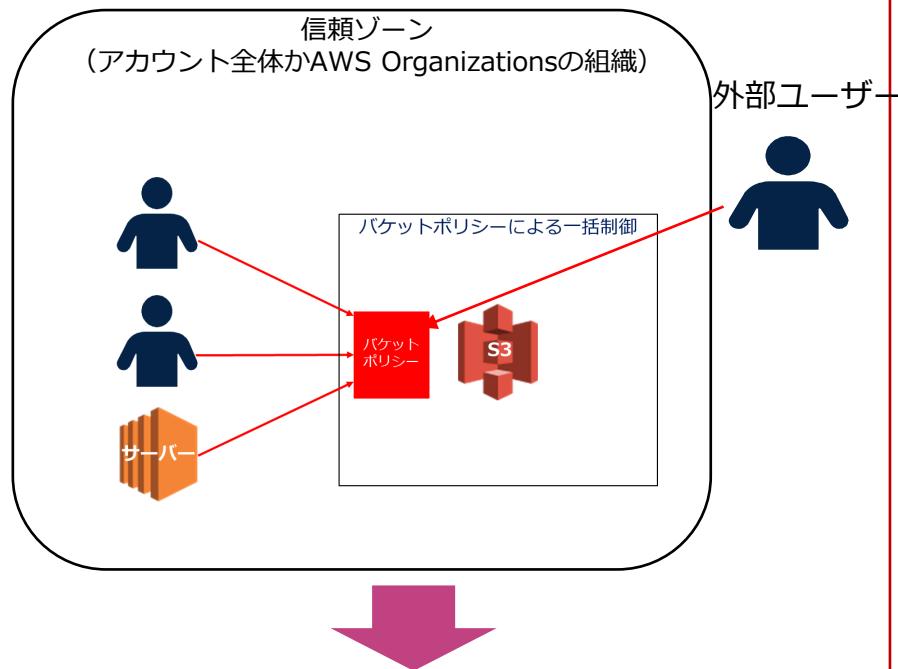
AWS CloudTrailは各種アカウントアクティビティやAPIコールをログに記録し、モニタリングするサービス



# ユーザーのアクティビティの記録

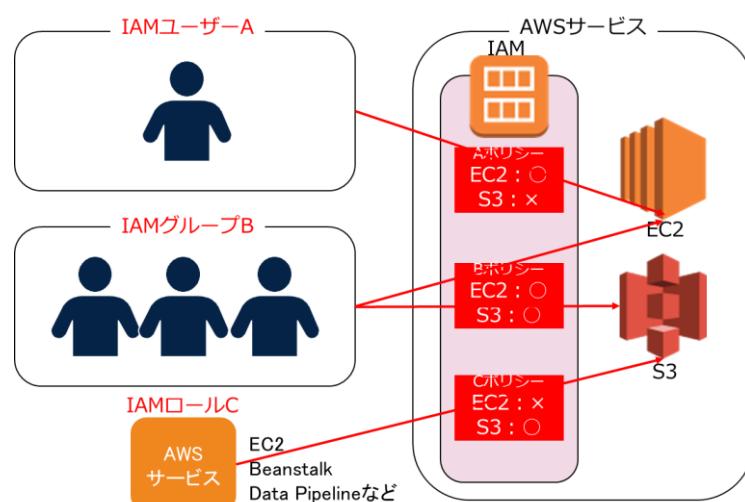
アクセスアナライザーは部外者アクセスの発見機能。アクセスアドバイザーは内部ユーザーのアクセス範囲の確認機能。

IAMアクセスアナライザー



外部ユーザーからのS3バケットへのアクセスが許可されている。

IAMアクセスアドバイザー



IAMユーザーAのアクセス範囲はEC2で○日○時にアクセスあり  
IAMグループBのアクセス範囲はEC2とS3で○日○時にアクセスあり  
IAMロールのアクセス範囲はS3で○日○時にアクセスあり



# [Q] IAM権限のベストプラクティス

あなたはAWSアカウントを新規に作成して、 AWSの設定を行っているところです。 AWSでは新規に作成したアカウントに対して設定すべきベストプラクティスが定義されています。これは実行しなければAWSを利用できないわけではありませんが、 実行が推奨されているため、 あなたは対応することになりました。

ソリューションアーキテクトとして、 AWSの初期に対応するべき事項を選択してください。 ( 3つ選択してください。 )

- 1) 全てのユーザーに対してMFA認証を有効化する。
- 2) アクセスキーを定期的にローテーションする。
- 3) アクセスキーは実施後に削除して、 不正アクセスを防ぐ。
- 4) Configのモニタリングを有効化する。
- 5) IAM アクセスアナライザーを利用して検証を実施する。
- 6) 強度の高いパスワードポリシーを設定する。

# IAM権限のベストプラクティス

IAMを利用する際はベストプラクティスに沿った運用をする。

- ✓ 人間のユーザーが一時的な認証情報を使用して AWS にアクセスするには、ID プロバイダーとのフェデレーションの使用が必要です
- ✓ AWS にアクセスするには、ワークフローが IAM ロールを使用して一時的な資格情報を使用する必要があります
- ✓ 多要素認証 (MFA) が必要です
- ✓ 長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする
- ✓ ルートユーザーの認証情報を保護し、日常的なタスクには使用しない
- ✓ 最小特権アクセス許可を適用する
- ✓ AWS 管理ポリシーの開始と最小特権のアクセス許可への移行
- ✓ IAM Access Analyzer を使用して、アクセスアクティビティに基づいて最小特権ポリシーを生成する
- ✓ 未使用のユーザー、ロール、アクセス許可、ポリシー、および認証情報を定期的に確認して削除する
- ✓ IAM ポリシーで条件を指定して、アクセスをさらに制限する
- ✓ IAM Access Analyzer を使用して、リソースへのパブリックアクセスおよびクロスアカウントアクセスを確認する
- ✓ IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的なアクセス許可を確保する
- ✓ 複数のアカウントにまたがるアクセス許可のガードレールを確立する
- ✓ アクセス権限の境界を使用して、アカウント内のアクセス許可の管理を委任します。

## S3の出題範囲

# S3とは何か？

S3は耐久性と可用性が非常に高くデータの中長期保存に最適なストレージ

ファイルやフォルダのアップロードや、バケットのバージニング、タグ、デフォルトの暗号化など、バケットの追加設定を行うには、[詳細の表示] を選択します。

Amazon S3

① S3 コンソールの新しいバージョンは引き続き改善されますが、バケットの以前のコンソールエクスペリエンスに一時的に切り替えることができます。エクスペリエンスの向上に役立てるため、フィードバックをお寄せください。

バケット (4)

バケットは S3 に保存されたデータのためのコンテナです。詳細

Q バケットを名前で検索

名前	リージョン	アクセス	作成日
elasticbeanstalk-ap-northeast-1-860853660447	アジアパシフィック (東京) ap-northeast-1	オブジェクトは公開することができます	2020/06/17 04:59:48 PM JST
test20200714-2	アジアパシフィック (東京) ap-northeast-1	非公開のバケットとオブジェクト	2020/07/14 08:09:04 PM JST
udemv-vpc111111	アジアパシフィック (東京) ap-northeast-1	非公開のバケットとオブジェクト	2020/07/01 10:35:38 PM JST
udemv2020108	アジアパシフィック (東京) ap-northeast-1	オブジェクトは公開することができます	2019/12/08 06:39:46 PM JST

Amazon S3 > test20200714-2

概要 プロパティ アクセス権限 管理 アクセスポイント

Q ブレフィックスを入力し、Enter キーで検索します。ESC を押してクリアします。

▲ アップロード + フォルダの作成 ダウンロード アクション ▾

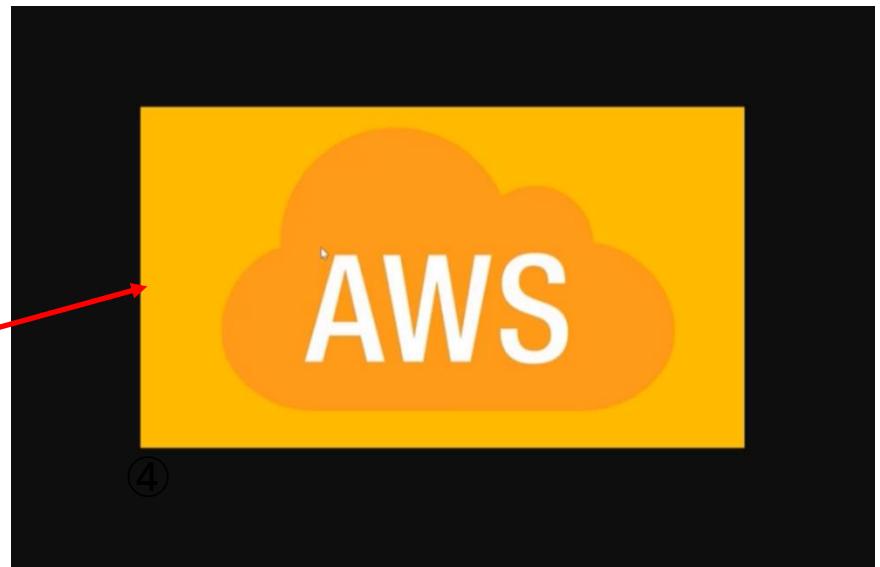
アジア

名前	最終更新日時	サイズ	ストレ
2594890_d1eb_2.jpg	7月 14, 2020 8:10:50 午後 GMT+0900	17.6 KB	スタン

# S3とは何か？

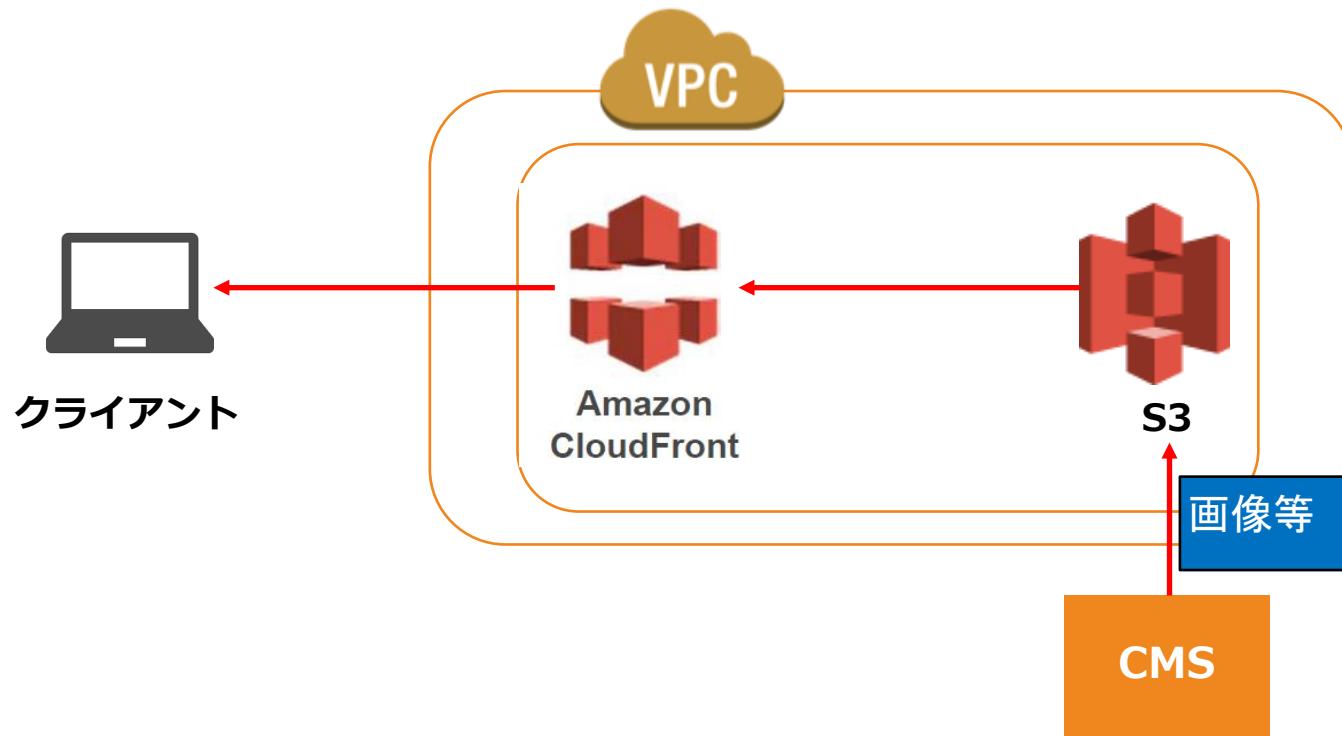
S3は耐久性と可用性が非常に高くデータの中長期保存に最適なストレージ

The screenshot shows the Amazon S3 console interface. The file path is `test20200714-2 > 2594890_d1eb_2.jpg`. The file name is `2594890_d1eb_2.jpg`, and the version is `最新バージョン`. The file was created by `shingoshibata` on `7月 14, 2020 8:10:50 午後 GMT+0900`. The ETag is `34e638997f57f722d3c4b34b1bd9c61`. The storage class is `スタンダード`. There is no server-side encryption. The size is `17.6 KB`. The key is `2594890_d1eb_2.jpg`. A red box highlights the `オブジェクト URL` field, which contains the value `https://test20200714-2.s3-ap-northeast-1.amazonaws.com/2594890_d1eb_2.jpg`. A red arrow labeled ③ points from this URL to the AWS logo in the adjacent image.



# S3のユースケース

コンテンツ配信用の画像データなどをS3に保存して、CloudFrontを利用して配信する。



# S3の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

S3ストレージの特徴	<ul style="list-style-type: none"><li>✓ シナリオのストレージ要件を満たすストレージを選択する質問</li><li>✓ S3ストレージの特徴を回答させる質問</li></ul>
S3のデータ容量制限	<ul style="list-style-type: none"><li>✓ S3のデータ容量に関するシンプルな質問</li></ul>
ストレージクラスの選択	<ul style="list-style-type: none"><li>✓ シナリオのストレージ要件を満たすS3のストレージクラスを選択する。</li><li>✓ ライフサイクル管理と一緒に出題されるパターンも多い。</li></ul>
S3の利用コスト	<ul style="list-style-type: none"><li>✓ S3におけるコストが発生する要素が質問として出題される。</li><li>✓ リクエストに応じた課金設定が可能な機能が問われることも。</li></ul>
ライフサイクル管理	<ul style="list-style-type: none"><li>✓ ライフサイクル管理によってデータ保存期間に応じて、ストレージクラスを移動させたり、削除させる適切な設定パターンが出題される。出来る組合せ／出来ない組合せがある。</li></ul>

# S3の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

バージョン管理	<ul style="list-style-type: none"><li>✓ S3ストレージ内のデータを誤って削除してしまった場合の予防策が問われる。</li><li>✓ MFA削除がセットで回答されるパターンが多い。</li></ul>
S3のアクセス管理	<ul style="list-style-type: none"><li>✓ バケットポリシー、ACL、IAMの利用方法と使い分けの問題</li><li>✓ バケットポリシー自体の設定内容を問う問題</li><li>✓ 事前署名付きURLによるアクセス制限に関する問題</li></ul>
ブロック パブリックアクセス	<ul style="list-style-type: none"><li>✓ オブジェクトのインターネットへの公開設定方法の問題が出題される。</li></ul>
クロスアカウント アクセス	<ul style="list-style-type: none"><li>✓ 他のアカウントにバケットを利用させる設定方法の問題が出題される。</li></ul>
S3アクセスポイント	<ul style="list-style-type: none"><li>✓ S3アクセスポイントの利用目的を問う問題が出題される。</li></ul>

# S3の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

静的WEBホスティング	✓ 静的WEBホスティングを実行するための設定方法や、静的コンテンツを実施する機能を選択させる質問が出題される。
Route53によるドメイン設定	✓ 静的WEBホスティングに対してRoute53によるドメインを設定する方法が問われる。
クロスオリジンリソースシェアリング(CORS)	✓ オリジンとしてドメインが設定されたS3バケットをオリジンとして複数アクセスできる設定が問われる。
S3イベント	✓ S3イベントが利用可能なサービスを選択する質問が出題される。 ✓ S3イベントを利用した実装方法が出題される。
S3の暗号化	✓ S3で利用できる暗号化方式が問われる。

# S3の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

レプリケーション	✓ S3でのレプリケーション方式やその設定方法が問われる。
S3のデータ解析	✓ S3と連携してデータ解析するサービスの選択が問われる。
S3の利用状況の確認	✓ S3のデータ利用状況やアクセス状況を確認・分析する方法が問われる。
S3の整合性モデル	✓ S3の読み込みや書き込みの整合性モデルに起因した問題が問われる。
マルチパートアップロード	✓ 大きなファイルをアップロードする際の最適な手法が問われる。

# S3の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

S3 Transfer Acceleration	<ul style="list-style-type: none"><li>✓ S3へのデータアップロードをグローバルに最適化するために必要な対応として出題される。</li></ul>
パフォーマンスの向上	<ul style="list-style-type: none"><li>✓ データの取得リクエストなどを効率化する方法が問われる。</li><li>✓ オブジェクトを多数アップロードする際に大量リクエストを処理する効率的な設定が問われる。</li></ul>

# [Q] S3ストレージの特徴

ベンチャー企業は複数の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アクセスやAPIコールに応じたログファイルを作成し続けるため、大量のログファイルを保存するストレージを必要としています。ストレージには頻繁にアクセスが発生し、大量のデータを安価に保存することが求められています。

次の中で、どのストレージサービスが最も費用効果が高いですか？

- 1) Amazon EFSのスタンダードストレージクラスを利用して、ファイルシステムマネージャーによるログ解析を実施する。
- 2) Amazon EBSの汎用SSDを利用して、Amazon EMRを連携したログ解析を実施する。
- 3) Amazon S3の標準ストレージタイプを利用して、Amazon EMRを連携したログ解析を実施する。
- 4) Amazon EC2インスタンスマーケットを利用して、EC2インスタンスからのログ解析を実施する。

# S3ストレージの特徴

AWSは3つの形式のストレージサービスを提供

## ブロックストレージ

- ✓ EC2にアタッチして活用するディスクサービス
- ✓ ブロック形式でデータを保存
- ✓ 高速・広帯域幅
- ✓ 例：EBS、インスタンスストア

## オブジェクトストレージ

- ✓ 安価かつ高い耐久性をもつオンラインストレージ
- ✓ オブジェクト形式でデータを保存
- ✓ デフォルトで複数AZに冗長化されている。
- ✓ 例：**S3**、Glacier

## ファイルストレージ

- ✓ 複数のEC2インスタンスから同時にアタッチ可能な共有ストレージサービス
- ✓ ファイル形式でデータを保存
- ✓ 例：EFS

# S3ストレージの特徴

S3はデータをオブジェクトとして保存。オブジェクトは以下の要素で構成されている

■Key

オブジェクトの名前であり、バケット内のオブジェクトは一意に識別

■Value

データそのものであり、バイト値で構成される

■バージョンID

バージョン管理に用いるID

■メタデータ

オブジェクトに付随する属性の情報

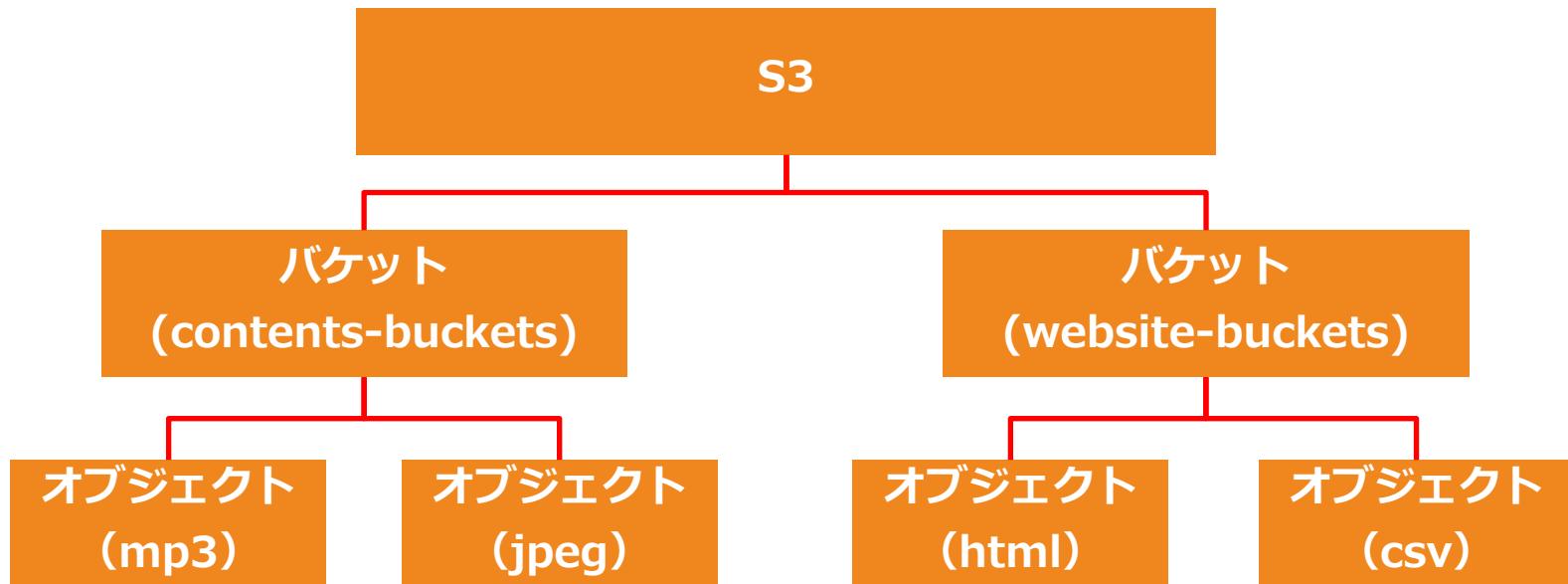
■サブリソース

バケット構成情報を保存および管理するためのサポートを提供

例：アクセスコントロールリスト（ACL）

# S3ストレージの特徴

S3はバケット単位で保存スペースを区分し、オブジェクトでデータを格納する



# [Q] S3のデータ容量制限

あなたは大手製造企業のエンジニアとして働いています。現在、あなたは社内に大量にある製造ドキュメントを効率的に保存・共有するための文書管理アプリケーションを構築しています。このソリューションではS3を利用してデータを保存することが決まっていますが、あなたは要件を定義するために保存データに関する制限を確認することが必要です。

次の中で、Amazon S3のデータ保存の制約として正しい説明はどれですか？（2つ選択してください。）

- 1) S3のストレージ容量はバケット作成時に設定し、その後自動でスケーリングする。
- 2) ストレージのデータ量と保存できるオブジェクトの数は無制限である。
- 3) 1つのPUTでアップロードできる最大のオブジェクトは5GBである
- 4) 1つのPUTでアップロードできる最大のオブジェクトは5TBである
- 5) S3は、ファイルシステムアクセスセマンティクス（ファイルロックなど）と、同時にアクセス可能なストレージを提供する。
- 6) S3のアクセスにはマウントヘルパーを利用する。

# S3のデータ容量制限

S3のストレージ容量は無制限であり、 0KBから5TBまでのデータを保存可能

## S3のデータ容量制限

### ■バケット

オブジェクトの保存場所。リージョンに設置されるため、名前はグローバルでユニークにする。**データ保存容量は無制限であり、自動でストレージ容量が拡張される。**

### ■オブジェクト

S3に格納されるファイル形式で、オブジェクトに対してURLが付与される。バケット内に**保存可能なオブジェクト数は無制限**

### ■保存可能なオブジェクトサイズの制限

オブジェクトあたりのデータサイズは**0KBから5TBまで保存可能**

# [Q] ストレージクラスの選択

世界4大監査法人の1つであるA社は、様々な監査レポートを作成しています。これらの監査レポートはセキュリティを強固にした上で、一定期間保存することが必要となります。また、これらの監査レポートを作成するための基礎となるデータはS3に保存され、数百テラバイトに達します。監査レポートの元データと監査レポートは頻繁にアクセスが発生します。

このユースケースに最適な最も費用効果の高いストレージクラスはどれですか？

- 1) S3 Standard-IA
- 2) S3 Standard
- 3) S3 Intelligent Tiering
- 4) S3 Glacier

# ストレージクラスの選択

S3の用途に応じてストレージタイプを選択する

タイプ	特徴	性能
<b>STANDARD</b>	<ul style="list-style-type: none"><li>✓ 複数個所にデータを複製するため耐久性が非常に高い。</li><li>✓ 頻繁に利用するデータを大量に保存するのに向いている。</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.99%</li></ul>
<b>STANDARD-IA</b>	<ul style="list-style-type: none"><li>✓ IAはInfrequency Accessの略であり、低頻度アクセスデータ用のストレージ。 One Zone-IAより重要なマスターデータ向け。データ取得は早い</li><li>✓ Standard に比べて安価だが、One Zone-IAよりは高い。</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.9%</li></ul>
<b>One Zone-IA</b>	<ul style="list-style-type: none"><li>✓ 低頻度アクセス用のストレージだが、マルチAZ分散されていないため可用性が低く、重要ではないデータ向け。その分Standard IAよりも値段が安い</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.5%</li></ul>
<b>S3 Intelligent Tiering</b>	<ul style="list-style-type: none"><li>✓ アクセス頻度ごとに自動的に保存先を割り振るストレージクラス。アクセス頻度が高いファイルは高頻度（標準クラス）に維持しつつ、アクセスがないファイルは低頻度（標準IAクラス）またはアーカイブインスタンスアクセス層に自動で移動する。</li><li>✓ アクセスパターンがわからない場合に利用</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.99%</li></ul> 

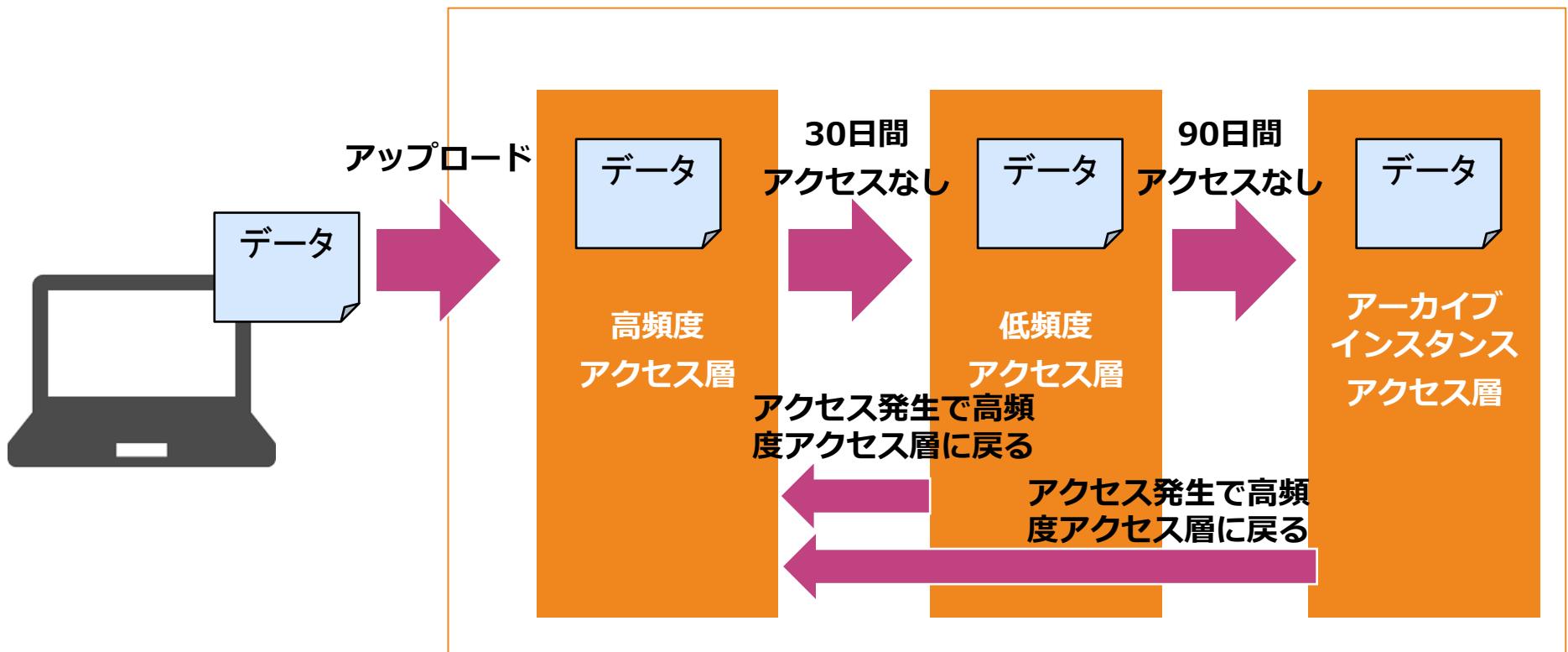
# ストレージクラスの選択

Glacierでは3つのストレージタイプから選択する。

タイプ	特徴	性能
<b>S3 Glacier Flexible Retrieval (通常のGlacier)</b>	<ul style="list-style-type: none"><li>✓ 1年に1~2回アクセスされ、非同期で取り出されるアーカイブデータ向け</li><li>✓ 標準は3~5時間でデータ取り出し可能</li><li>✓ 迅速取り出しは2~5分でデータ取り出し可能</li><li>✓ 一括検索は5~12時間で無料でデータ取り出し可能</li><li>✓ ライフサイクル管理で利用指定できる</li><li>✓ ボールトロック機能によりデータを保持・保護</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.99%</li></ul>
<b>S3 Glacier Instant Retrieval</b>	<ul style="list-style-type: none"><li>✓ アクセスされることがほとんどなく、ミリ秒単位の取り出しが必要な長期間有効なデータ向け</li><li>✓ 医療画像やニュースメディアなど</li><li>✓ S3 Standardと同じパフォーマンスのミリ秒単位でのデータの取り出し</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.9%</li></ul>
<b>Amazon Glacier Deep Archive</b>	<ul style="list-style-type: none"><li>✓ 最安のアーカイブ用ストレージ</li><li>✓ 7~10年以上保持される長期間使用されるものの、めったにアクセスされないデータ向け</li><li>✓ 標準の取り出し速度は12時間以内にデータを取得</li><li>✓ 大容量取り出しは48時間以内にデータを取得</li><li>✓ ライフサイクル管理で利用指定できる</li></ul>	<ul style="list-style-type: none"><li>■ 耐久性 99.99999999%</li><li>■ 可用性 99.99%</li></ul> 

# S3 Intelligent-Tiering

アクセス頻度に応じてオブジェクトを自動的に低コストのアクセス層に移動することでコストを削減する。



さらに180日間アクセスなしで  
ディープアーカイブアクセス層へ

# [Q] S3の利用コスト

あなたはAWS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では、EC2インスタンスからストレージにアクセスして、データを保存したり、取得するといった処理が多数発生する予定です。ストレージに必要なI/O性能やレイテンシーなどを比較したところ、S3、EBS、EFSのどれでも対応が可能なようです。そのため、あなたはソリューションアーキテクトとして、最もコストが安いストレージを選択することにしました。また、いづれのストレージにおいても標準ストレージまたは汎用ストレージを利用します。

これら3つのストレージを、コストが安い順番で左から並べてください。

- 1) S3標準 < EBS汎用ボリューム < EFS標準
- 2) S3標準 < EFS標準 < EBS汎用ボリューム
- 3) EBS汎用ボリューム < EFS標準 < S3標準
- 4) EBS汎用ボリューム < S3標準 < EFS標準

# S3の利用コスト

ストレージのコストを比較するとインスタンスストアを除けば、最も値段が安いのはS3およびGlacier

S3のデータ容量  
に応じたコスト

- ✓ 標準 : 1 GBあたり 0.025USD／月
- ✓ S3 Intelligent Tiering:標準と標準IAの組合せ
- ✓ 標準IA : 1 GBあたり 0.019USD／月
- ✓ One Zone IA : 1 GBあたり 0.0152USD／月
- ✓ Glacier : 1 GBあたり 0.005USD／月
- ✓ Glacier deep archive : 1 GBあたり 0.002USD／月

EBSの汎用  
ストレージのコスト

- ✓ 汎用 : 1 GBあたり 0.12USD／月
- ✓ コールドHDD:1 GBあたり 0.03USD／月

EFS  
ストレージのコスト

- ✓ 標準 : 1 GBあたり 0.36USD／月
- ✓ 低頻度アクセス : 0.0272USD／月

インスタンスストア

- ✓ EC2インスタンスに含まれる。

## [Q] S3の利用コスト

ある企業は日本に多数のオフィスを展開しており、各オフィスにあるデバイスから多数のデータを収集しています。この会社は、これらのデータをAmazon S3バケットに保存しています。現在のデータ量は10TBに及びます。同社はS3バケットを使用しているヨーロッパの会社とデータを共有することになりました。その際は、同社側のデータ共有にかかるコストを最小限にする必要があります。

会社にとって最も費用対効果が高いソリューションはどれでしょうか？

- 1) S3バケットのクロスアカウントアクセスを有効化して、S3バケットのリクエスト支払い機能を有効化する。画像のアップロードにS3 Transfer Accelerationは無料で利用できる。
- 2) ヨーロッパ企業にS3バケットのアクセスを許可するバケットポリシーを設定する。
- 3) S3バケットのクロスリージョンレプリケーションを有効化して、ヨーロッパ企業のS3バケットとレプリケーションする。
- 4) ヨーロッパ企業にS3バケットのアクセスを許可するクロスアカウントアクセス権限を設定する。

# S3の利用コスト

S3はデータ量とリクエストとデータ転送に対して料金が発生

リージョン	<ul style="list-style-type: none"><li>✓ リージョン：リージョン毎に価格が異なる。</li></ul>
データ容量	<ul style="list-style-type: none"><li>✓ データ容量：データ量と保存期間に応じて料金がかかる。 (GBあたり)</li><li>✓ S3 Intelligent Tiering、IAストレージには、最低 30 日間の料金</li></ul>
リクエストとデータ取得	<ul style="list-style-type: none"><li>✓ データに対するリクエストに応じて料金がかかる。 (1000リクエストあたり)</li><li>✓ データを取得した量に応じて料金がかかる (GBあたり)</li><li>✓ リクエスタ支払でリクエスト側に課金してもらえる。</li></ul>
データ転送	<ul style="list-style-type: none"><li>✓ データ転送イン：無料</li><li>✓ インターネットへのデータ転送アウト (GBあたり)</li><li>✓ S3からAWS内のデータ転送アウト (GBあたり)</li></ul>

# S3の利用コスト

## S3はボリュームディスカウントの価格帯が設定されている

ストレージ料金表

**S3 標準** - 頻繁にアクセスするデータに一般的に使用される、あらゆるタイプのデータの汎用ストレージ

最初の 50 TB/月	0.025USD/GB
次の 450 TB/月	0.024USD/GB
500 TB/月以上	0.023USD/GB

**S3 Intelligent - Tiering \*** - アクセスパターンが不明または変化するデータの自動コスト削減

高頻度アクセスティア、最初の 50 TB/月	0.025USD/GB
高頻度アクセスティア、次の 450 TB/月	0.024USD/GB
高頻度アクセスティア、500 TB/月を超える	0.023USD/GB
低頻度アクセスティア、すべてのストレージ/月	0.019USD/GB
モニタリングおよびオートメーション、すべてのストレージ/月	オブジェクト 1,000 件あたり 0.0025USD

**S3 標準 - 低頻度アクセス \*** - ミリ秒単位のアクセスが必要な、長期保管だがアクセス頻度の低いデータの場合

すべてのストレージ/月	0.019USD/GB
-------------	-------------

**S3 1 ゾーン - 低頻度アクセス \*** - ミリ秒単位のアクセスが必要な、再作成可能なアクセス頻度の低いデータの場合

すべてのストレージ/月	0.0152USD/GB
-------------	--------------

**S3 Glacier \*\*** - 1 分から 12 時間の取り出しオプションを使用した長期バックアップとアーカイブの場合

すべてのストレージ/月	0.005USD/GB
-------------	-------------

**S3 Glacier Deep Archive \*\*** - 1 年に 1~2 回アクセスされ、12 時間以内に復元できる長期のデータアーカイブの場合

すべてのストレージ/月	0.002USD/GB
-------------	-------------

## [Q] ライフサイクル管理

あなたはソリューションアーキテクトとして、自社のドキュメント管理ストレージを設定・管理しています。保存しているデータ量が大変多いためにS3ストレージの利用コストが高いことが問題となっており、あなたはコスト削減を実施するようにボスより依頼されました。ライフサイクルルールを新規に設定して、時間の経過とともにオブジェクトをより安いストレージクラスへと移行する設定が必要です。しかしながら、いくつかのライフサイクルルールは設定することができませんでした。

次の中で、設定することができないライフサイクルルールはどれでしょうか？（2つ選択してください）

- 1) S3 Standard ⇒ S3 Intelligent-Tiering
- 2) S3 Standard-IA ⇒ S3 Intelligent-Tiering
- 3) S3 Standard-IA ⇒ S3 One Zone-IA
- 4) S3 Intelligent-Tiering ⇒ S3 Standard
- 5) S3 One Zone-IA ⇒ S3 Standard-IA
- 6) S3 Glacier ⇒ S3 Standard-IA

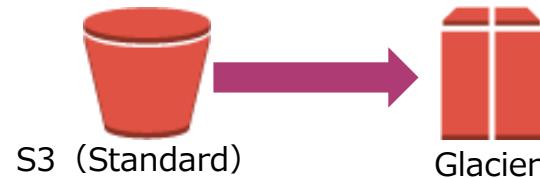
# ライフサイクル管理

時間に応じてオブジェクトのストレージクラスの変更や削除を自動的に行うルールを設定できる。

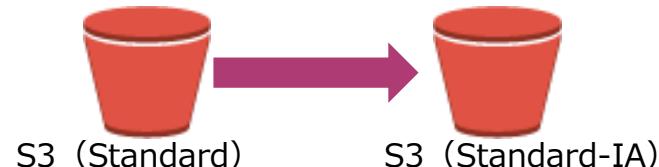
## 設定方法

- バケット全体やPrefixに設定
- オブジェクト更新日を基準にして日単位で指定し、毎日0:00UTCにキューを実行
- 最大1000ルール
- IAに移動できるのは128KB以上のオブジェクト
- MFA Deleteが有効だと設定不可

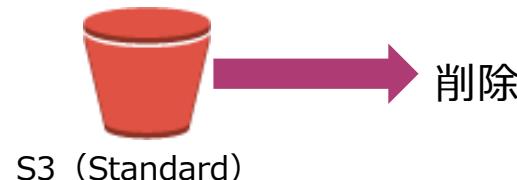
### 一定期間で自動アーカイブ



### 一定期間で自動で安価な保存場所へ

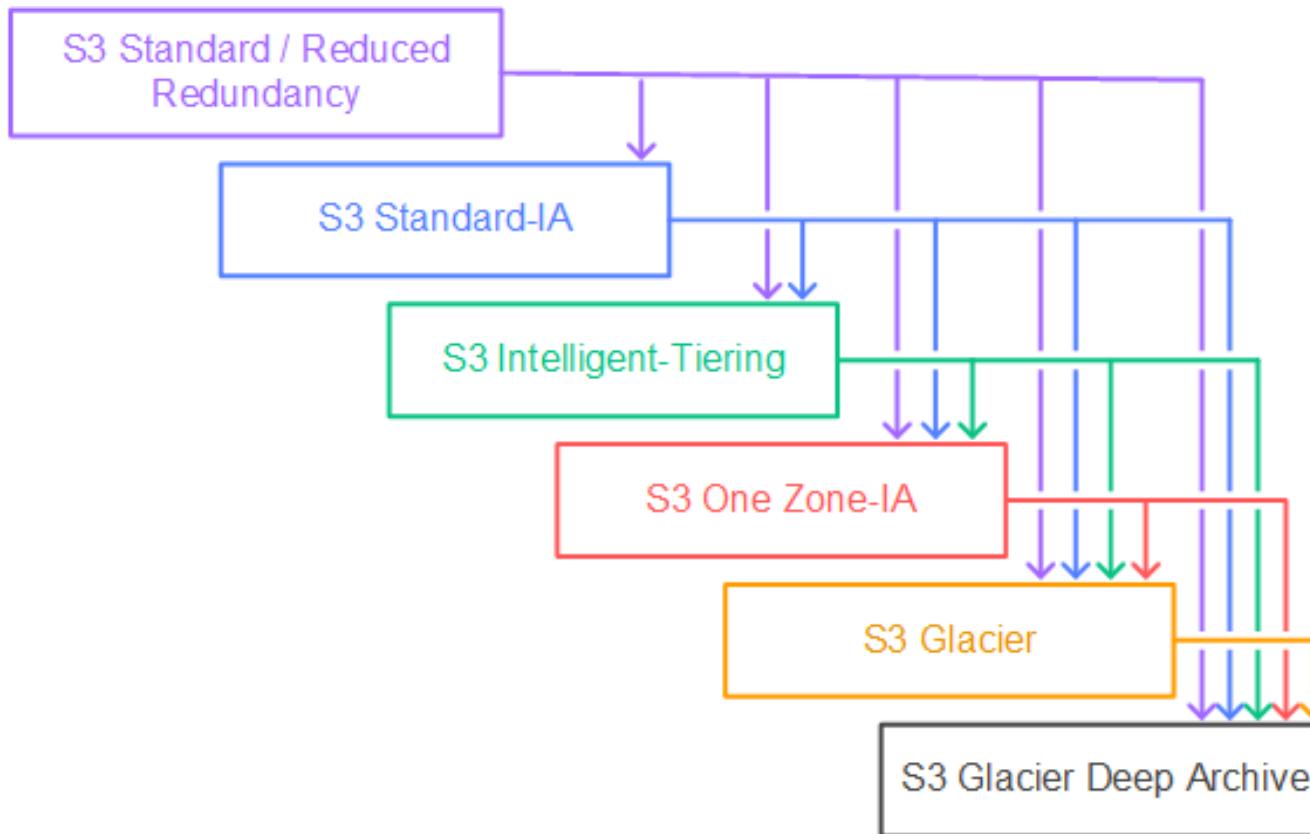


### 一定期間で自動で削除



# ライフサイクル管理

ライフサイクルポリシーを設定可能なパスは以下の通り



Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html)

## [Q]バージョン管理

シリコンバレーのベンチャー企業はAmazonS3を利用してデータを従業員間で共有しています。これらのデータが誤って削除したりしないように、オブジェクトを保護する設定が必要です。

ソリューションアーキテクトとして、要件を満たすことができる対応を選択してください。（2つ選択してください）

- 1) バケットでバージョン管理を有効にする。
- 2) S3オブジェクトを削除したときにイベントトリガーを作成して、SNSによる通知を設定する。
- 3) バケットでライフサイクルルールを有効にする。
- 4) MFA削除を有効にする。
- 5) バケットの設定でデータ削除不可を有効にする。

# バージョン管理

ユーザーによる誤操作でデータ削除などが発生してもバージョンから復元できる

## 設定

- バケット全体をバージョン管理する
- バージョンごとにオブジェクトが保管される。
- ライフサイクル管理によりバージョンが保存される期間を設定
- オブジェクトとは別に古いバージョン削除を実施する必要がある。

【現在】  
バージョンID  
00011

データA

データB

データC

【過去分】  
バージョンID  
00010

データA

データB

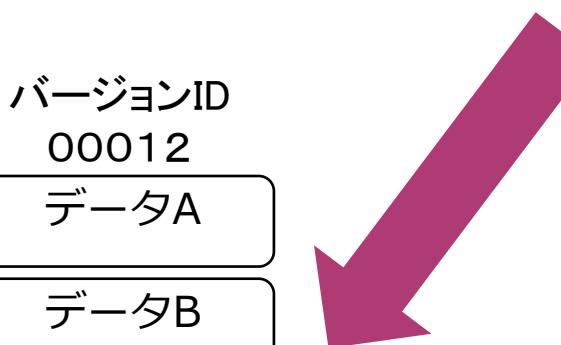
データC

バージョンID  
00012

データA

データB

データC



# S3 MFA Delete

バージョニング機能のオプションとして、オブジェクト削除時にMFA認証を必須にできる。

The screenshot shows the AWS IAM 'Your Security Credentials' page. The left sidebar lists navigation options: Dashboard, Search IAM, Details, Groups, Users, Roles, Policies, and Identity Providers. The main content area is titled 'Your Security Credentials' and contains instructions for managing credentials. It mentions using this page for AWS accounts and the IAM Console for IAM users. It also links to the AWS General Reference for more information on AWS credentials. A list of credential types is shown, with 'Multi-Factor Authentication (MFA)' being expanded. Below this, a note explains that AWS MFA increases security by requiring both a user name and password plus an authentication code from an AWS MFA device. A blue 'Activate MFA' button is visible at the bottom of this section.

AWS Services Edit Laurence Gellert Global Support

Dashboard

Search IAM

Details

Groups

Users

Roles

Policies

Identity Providers

Activate MFA

## Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

+ Password

- Multi-Factor Authentication (MFA)

You use AWS MFA to increase the security of your AWS environments when you sign in AWS websites. When AWS MFA is enabled, you must provide not only a user name and password but also an authentication code from an AWS MFA device.

# [新Q]オブジェクトロック

あなたはソリューションアーキテクトとして、AWS上でドキュメント共有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ユーザーがWEBインターフェイスまたはモバイルアプリ経由でドキュメントをアップロードします。厳しいセキュリティ要件に従って、新しいドキュメントは保存後に変更も削除もできないようにする必要があります。

この要件を満たすために、ソリューションアーキテクトはどうするべきでしょうか。

- 1) S3オブジェクトロックを有効にしたS3バケットを作成して、その中にドキュメントを保存する。
- 2) アップロードされたドキュメントをAmazon S3バケットに保存する。さらに、アーカイブをGlacierに移行してボールトロックを適用する。
- 3) S3バケットを作成した後、ストレージタイプをS3 Glacierに変更する。このGlacierにボールトロックを適用してから、ドキュメントを保存する。
- 4) S3バケットを作成した後、プロパティ変更画面でS3オブジェクトロックを有効にする。そのバケットの中にドキュメントを保存する。



# オブジェクトロック

オブジェクトロックを有効化することで、オブジェクトの削除・更新を一定時間ブロックすることができる。

- オブジェクトの削除・更新を一定時間ブロックする機能。Write Once Read Manyモデルを適用して、最初のアップロード後は読み取りしか許可されない。
- バケット作成時にしか適用できない。
- バージョンの保護
  - 保持期間：一定期間のオブジェクトバージョンの上書きと削除を防ぐ
  - リーガルホールド：オブジェクトが削除されるまでオブジェクトバージョンの上書きと削除を防ぐ
- モード
  - ガバナンスモード：特別なアクセス許可を持たないユーザーはオブジェクトのバージョンの上書きや削除、ロック設定を変更することができない。管理者やルートユーザーは可能
  - コンプライアンスマード：ルートユーザーを含め、全てのユーザーが、保護されたオブジェクトのバージョンを上書きまたは削除できない。

## バケット作成時に有効化して編集する

Amazon S3 > バケット > test20231123 > オブジェクトロックを編集

オブジェクトロックを編集 [Info](#)

**オブジェクトロック**  
Write-Once-Read-Many (WORM) モデルを使用してオブジェクトを保存すると、オブジェクトが一定期間または無期限に削除または上書きされるのを防げます。[詳細](#)

① Amazon S3 オブジェクトロックを有効にすると、オブジェクトロックを無効にしたり、バケットのバケットバージョニングを停止したりすることはできません。

オブジェクトロック  
有効

デフォルトの保持期間  
このバケットに配置された新しいオブジェクトが削除または上書きされないように自動的に保護します。

無効にする  
 有効にする

デフォルトの保持セード  
 ガバナンス  
特定の IAM アクセス許可を持つユーザーは、保持期間中に、保護されたオブジェクトを上書きまたは削除することができます。

コンプライアンス  
保持期間中は、どのユーザーも、保護されたオブジェクトバージョンを上書きまたは削除することはできません。

デフォルトの保持期間  
300 日

正の整数である必要があります。

キャンセル [変更の保存](#)

## バケット作成時に無効になると編集不可

オブジェクトロック

Write-Once-Read-Many (WORM) モデルを使用してオブジェクトを保存すると、オブジェクトが一定期間または無期限に削除または上書きされるのを防げます。[詳細](#)

オブジェクトロック  
無効

① Amazon S3 は現在、バケット作成後のオブジェクトロックの有効化をサポートしていません。このバケットのオブジェクトロックを有効にするには、カスタマイリポート[\[詳細\]](#)に連絡してください。

# [Q] S3のアクセス管理

あなたはソリューションアーキテクトとして、AWS上でドキュメント共有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、AmazonS3バケットに保存されたデータを読み込むプロセスをEC2インスタンス上で実行しています。これらのデータは、このWEBアプリケーションを介してのみ特定のユーザーにのみ閲覧できるように制限することが必要です。

WEBアプリケーションからのみデータにアクセスするための設定はどれでしょうか？

- 1) バケットポリシーにより、Webアプリケーション上のURLからの参照のみを許可する設定を行う。
- 2) IAMロールにより、WebアプリケーションのみがS3バケットへのアクセスを許可する設定を行う。
- 3) ACLにより、WebアプリケーションのみがS3バケットへのアクセスを許可する設定を行う。
- 4) 署名付きURLにより、Webアプリケーション上のURLからの参照のみを許可する設定を行う。

# [Q] S3のアクセス管理

あなたはAWSでデータ解析システムを構築しています。このシステムはIoTセンサーからデータを取得してKinesis Data Streamsがストリーミング処理したデータをKinesis Data Firehoseを介してAmazonS3バケットに保存します。その後、S3バケット内のデータに対してSQLクエリを使用して暗号化されたデータを簡易にクエリ処理して、結果をS3バケットに書き戻す必要があります。データは機密性が高いため、S3バケットへのアクセスに対してきめ細かい制御を実装する必要があります。

これらの要件を満たすことができるソリューションの組合せを選択してください。  
(2つ選択してください)

- 1) Athenaによりデータをクエリ処理して、結果をバケットに保存する。
- 2) Redshiftによりデータをクエリ処理して、結果をバケットに保存する。
- 3) バケットACLを使用して、バケットへのアクセスを制限する。
- 4) Amazon EMRによりデータをクエリ処理して、結果をバケットに保存する。
- 5) バケットポリシーを使用して、バケットへのアクセスを制限する。
- 6) IAMポリシーを使用して、バケットへのアクセスを制限する。

# S3のアクセス管理

S3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 ユーザー policy	<ul style="list-style-type: none"><li>✓ IAMユーザーに対してAWSリソースとしてのS3へのアクセス権限を設定</li><li>✓ 内部のIAMユーザーやAWSリソースへの権限管理</li></ul>
バケットポリシー	<ul style="list-style-type: none"><li>✓ バケットのアクセス権をJSONで設定</li><li>✓ 外部のユーザーも含めたアクセス管理</li></ul>
ACL	<ul style="list-style-type: none"><li>✓ バケット／オブジェクト単位でのアクセス権限をXMLで設定することができる</li><li>✓ オブジェクトに個別に設定可能</li></ul>
事前署名付きURL	<ul style="list-style-type: none"><li>✓ AWS SDKで生成した事前署名付きURLでS3オブジェクトURLにアクセスできる権利を一定期間付与する。</li><li>✓ インターネット上の第三者にURLを閲覧させる。</li></ul>

# [Q] S3バケットポリシー

次のバケットポリシーでS3バケットに対する権限設定を行っています。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadForGetBucketObjects",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::mybucket/*"  
        }  
    ]  
}
```

この設定内容として正しい内容を選択してください。

- 1) このS3バケットの全てのアクションが許可されている。
- 2) このS3バケットを利用した静的ホスティングを有効化できる。
- 3) このS3バケットは該当バケットの所有者は削除以外の操作が全て実行できる。
- 4) このS3バケットはオブジェクトのアップロードが可能である。

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

ポリシーのバージョン。  
必ず先頭に記載する。

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

Statementがポリシー内容を記述する部分

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

Sid (ステートメント ID) は、ユーザーが  
ポリシーに与える任意の識別子

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": ["AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]],  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

許可するポリシーか、拒否するポリシーかを決める。

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

対象となるプリンシパル(IAMユーザー  
やルートアカウントなど)を指定

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"],  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

Effectを適用するアクションを指定

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": "public-read"}}  
    }  
  ]  
}
```

ポリシーを適用する対象バケットを指定

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# S3バケットポリシー

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AddCannedAcl",  
      "Effect": "Allow",  
      "Principal": {"AWS": ["arn:aws:iam::11122223333:root", "arn:aws:iam::444455556666:root"]},  
      "Action": ["s3:PutObject", "s3:PutObjectAcl"],  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}  
    }  
  ]  
}
```

ポリシーを適用する場合の条件を指定

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-bucket-policies.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-bucket-policies.html)

# [Q]事前署名付きURL

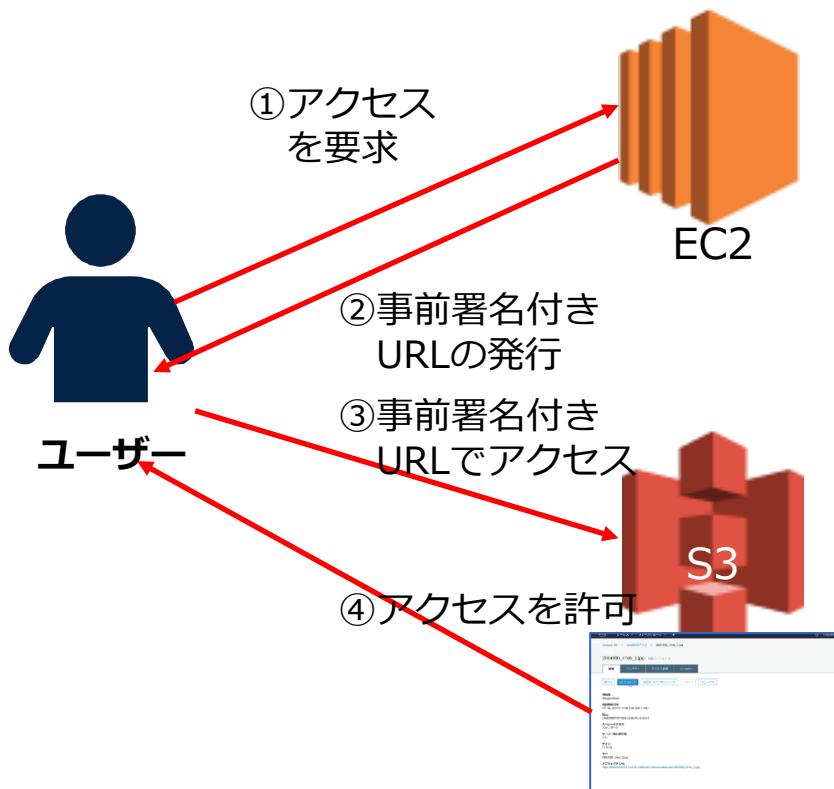
あなたはソリューションアーキテクトとして、動画共有アプリケーションを構築しています。このアプリケーションでは、S3バケットにビデオファイルを大量に保存して、EC2インスタンスを介してユーザーに一時的に共有されます。その際に、許可されたユーザーのみが動画データにアクセスできるようにする必要があります。

このアクセスを有効にするためのS3の設定を選択してください。

- 1) S3バケットのブロックパブリックアクセスを無効にして、URLが閲覧できるようになる。
- 2) CloudFrontを使用して、キャッシュに基づいて画像を配布する。
- 3) ACLを利用して、動画が共有されるユーザーからのアクセスを許可する。
- 4) 事前署名URLを生成し、動画が共有されるユーザーに配布する。

# 事前署名付きURL

事前署名付きURLにより、特定のユーザーのみがアクセスできる特別なURLが利用可能になる。



- 事前署名付きURLにアクセスできる外部ユーザーは、オブジェクトに対するGET/PUT権限を付与される。
- URLには有効期限が設定されていて、一時的なオブジェクトアクセス権限を付与するのに利用される。有効期限はデフォルトでは3600秒間
- SDKやCLIを使ってプログラム処理時などにURLを作成する仕組みを組み込むことができる。

## [Q]パブリックアクセス

あなたはメディア会社のソリューションアーキテクトです。現在、WEBメディアをAWS上で運用しており、このWEBメディアの静的コンテンツを提供するためにAmazon S3バケットを設定しています。この設定では、S3バケットにアップロードされたすべてのオブジェクトをインターネットに公開することが必要です。

ソリューションアーキテクトは、どのように設定すれば良いでしょうか？（2つ選択してください。）

- 1) ブロックパブリックアクセスを無効化する。
- 2) パブリックアクセス設定を有効化する。
- 3) バケットポリシーでインターネットからのアクセスを許可する。
- 4) ACLでインターネットからのアクセスを許可する。
- 5) IAMでインターネットからのアクセスを許可する。

# ブロックパブリックアクセス

インターネットからのアクセスをブロックする機能で、バケット作成時に初期設定で有効化されている。

The screenshot shows the AWS S3 Bucket Properties interface. The top navigation bar has tabs: 概要 (Overview), プロパティ (Properties), アクセス権限 (Access Permissions), and 管理 (Management). The Management tab is selected. Below it, there are four sub-tabs: ブロックパブリックアクセス (selected), アクセスコントロールリスト (Access Control List), バケットポリシー (Bucket Policy), and CORS の設定 (CORS Settings). The main content area is titled "ブロックパブリックアクセス (バケット設定)" (Block Public Access (Bucket Settings)). It contains a section titled "パブリックアクセスをすべてブロック" (Block all public access) with the status "オフ" (Off). A "編集" (Edit) button is located in the top right corner of this section. Below this, there are four items listed under "新しいアクセスコントロールリスト (ACL) を介して許可されたバケットとオブジェクトへのパブリックアクセスをブロックする" (Block public access through new ACLs):

- オフ
- 任意のアクセスコントロールリスト (ACL) を介して許可されたバケットとオブジェクトへのパブリックアクセスをブロックする  
オフ
- 新しいパブリックバケットポリシーを介して許可されたバケットとオブジェクトへのパブリックアクセスをブロックする  
オン
- 任意のパブリックバケットポリシーを介して、バケットとオブジェクトへのパブリックアクセスとクロスアカウントアクセスをブロックする  
オン

# [Q]クロスアカウントアクセス

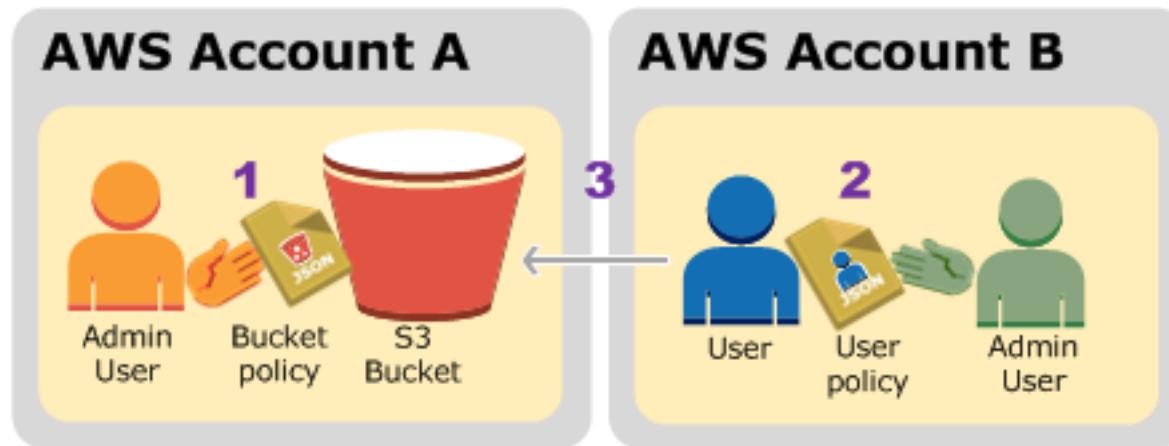
大手製造企業は5000人の従業員を有する大企業です。複数の部門がAWSアカウントを使用しているため、多数のAWSアカウントを管理することが必要です。Aアカウントが所有するS3バケット内のオブジェクトを、Bアカウントに属する別のS3バケットにコピーする要件が発生しました。あなたはソリューションアーキテクトとして、コピーされたオブジェクトを送信先アカウントが所有する設定を行っています。

この要件を満たすための設定方法を選択してください。

- 1) AアカウントのS3バケットでリクエスタ支払機能を有効にして、Bアカウントへのコピーを実施する。
- 2) AアカウントのオブジェクトをBアカウントにコピーできるようにするIAMカスタマーマネジメントポリシーを作成して、S3でクロスオリジンリソースシェアリングを設定する。
- 3) Aアカウントのバケットから、Bアカウントのバケットへとレプリケーションを設定して、S3でクロスオリジンリソースシェアリングを設定する。
- 4) AアカウントのオブジェクトをBアカウントにコピーできるようにするIAMカスタマーマネジメントポリシーを作成して、S3でクロスアカウントアクセスを許可して、IAMユーザーに設定する。

# クロスアカウントアクセス

アカウントAの所有するバケットに対して、アカウントBへのアクセス許可を与えることが可能



Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/dev/example-walkthroughs-managing-access-example2.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/example-walkthroughs-managing-access-example2.html)

# クロスアカウントアクセス

クロスアカウントアクセスを許可する設定は3つの方式がある

設定方式	詳細
バケットポリシーと IAMポリシーによる 許可	<ul style="list-style-type: none"><li>✓ S3バケットにアクセスを許可するIAMポリシーを設定する。</li><li>✓ S3バケットへのクロスアカウントアクセスを許可する場合はバケットポリシーでアカウントを指定して許可を行う。</li><li>✓ IAMユーザーとロールに設定</li></ul>
ACLとIAMポリシー による許可	<ul style="list-style-type: none"><li>✓ S3バケットにオブジェクトへの操作を許可するIAMポリシーを設定する。</li><li>✓ S3バケットの特定オブジェクトへのクロスアカウントアクセスを許可する場合はACLでアカウントを指定して許可を設定</li><li>✓ AWSアカウントに設定</li></ul>
IAMロールによる 許可	<ul style="list-style-type: none"><li>✓ AssumeRoleを利用してS3バケットオブジェクトへのプログラムによるアクセスまたはコンソールアクセス用のクロスアカウントの IAM ロールを設定する。</li><li>✓ ユーザAからAssumeRoleの実行を許可したロールBに対して権限を付与する</li></ul>

# [Q] S3アクセスポイント

ある会社ではAWS上にドキュメント管理システムを構築しているところです。このストレージはグローバルに複数部門や複数のアプリケーションが利用するため、様々なアクセス制御ルールを設定することが必要です。したがって、あなたはソリューションアーキテクトとして、S3 の共有データセットへの大規模なデータアクセス管理を簡素化する設定を検討しています。

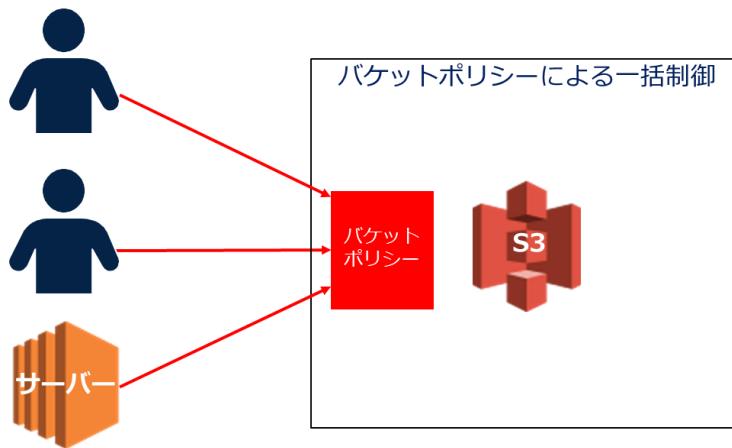
この要件を満たすことができるアクセス設定を選択してください。

- 1) Amazon S3 Transfer Accelerationを利用して、複数のアプリケーションと連携して、アクセスパフォーマンスを向上させる。
- 2) S3 アクセスポイントを利用して、複数のアプリケーションへのアクセスを制御する。
- 3) VPC エンドポイントを利用して、ルートテーブルへのルートを構成することで複数のアプリケーションへのアクセスを制御する。
- 4) マルチパートアップロードを利用して、複数のアプリケーションへのアクセスを制御する。

# S3アクセスポイント

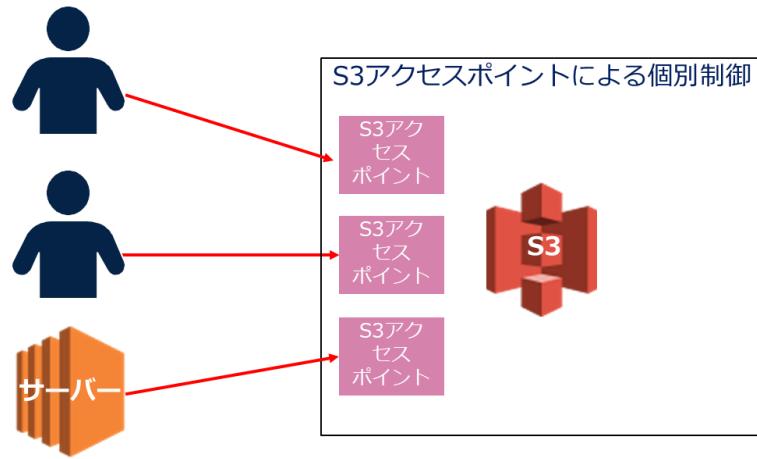
アクセス先に応じてアクセスポイントを作成して、ポリシーを適用してアクセス設定が可能になる。

## バケットポリシー



- 今までではS3に対する複数のアクセス可否を1つのバケットポリシーで管理

## アクセスポイントポリシー



- アクセス先ごとにアクセスポイントを作成して管理を分割できる。
- 別のアカウントにある S3 バケットに対してアクセスポイント（クロスアカウントアクセスポイント）の作成可能
- 複数の AWS リージョンにある S3 バケットからのリクエストを実行するために使用できるグローバルエンドポイントを作成可能

# [新Q] 静的WEBホスティング

ある企業が、オンプレミス環境上の静的ウェブサイトをAWSに移行することを決定しました。このウェブサイトは世界中のユーザーに閲覧されているグローバルなサイトとなっており、できるだけ迅速に世界中のユーザーが閲覧できる必要があります。また、これにはコスト最適なアーキテクチャが求められています。

この要件を満たすために、ソリューションアーキテクトは何を実施するべきでしょうか。

- 1) 静的WEBサイトをAmazon S3から配信するように静的ウェブホスティングをS3バケットに構成する。S3バケットを複数のAWSリージョンにレプリケートして、各リージョンにて配信をする。
- 2) 静的WEBサイトをAmazon S3から配信するように静的ウェブホスティングをS3バケットに構成する。Amazon CloudFrontを構成し、S3バケットをオリジンとして設定する。
- 3) 静的WEBサイトをAmazon S3から配信するように静的ウェブホスティングをS3バケットに構成する。Route53を利用して、各リージョンにルーティングするように構成する。
- 4) 静的WEBサイトをAmazon S3から配信するように静的ウェブホスティングをS3バケットに構成する。S3バケットを複数のAWSリージョンにレプリケートして、Route53を利用して、各リージョンにルーティングするように構成する。

# 静的WEBホスティング

静的サイトを構築する場合は、静的WEBホスティングによる安価なWEBページを構築可能

## 静的WEBホスティング メリット

- サーバーなしにWEBサイトをホスティング可能。
- サーバーが必要ないため値段が安い。
- マルチAZの冗長化を勝手にしてくれており、運用いらず
- Route53で独自ドメインを設定可能
- CloudFrontによる配信可能

## 静的WEBホスティング デメリット

- サーバーサイドスクリプト言語を実行するなどの動的サイト不可
- 単独ではSSLが利用できず、SSL設定にはCloudFrontが必要

## WEBサイト エンドポイント

使用しているリージョンに応じて、Amazon S3 ウェブサイトエンドポイントは以下の 2 つの形式のいずれかになる。

- ✓ <http://bucket-name.s3-website-Region.amazonaws.com>
- ✓ <http://bucket-name.s3-website.Region.amazonaws.com>

# 静的WEBホスティング

静的サイトを構築する場合は、静的WEBホスティングによる安価なWEBページを構築可能

ロックパブリックアクセスを無効化する。

バケットポリシーでバケットの読み取り許可を設定する。

Index.htmlなどのインデックスドキュメントをバケット内に保存する。

静的WEBホスティングの設定画面で  
Index.htmlなどのインデックスドキュメントを  
設定し、有効化する。

# [Q] Route53によるドメイン設定

あなたは会社のコーポレイトサイトをAWS上に構築しています。このサイトはシンプルな静的WEBサイトであり、なるべくコストを抑えるためにAmazonS3にデプロイしました。あなたは更に、Route 53を使用して新しいドメイン名をこのWEBサイトに設定したいと考えています。

Route53を使用してS3静的Webサイトにトラフィックをルーティングする設定を選択してください。（2つ選択してください。）

- 1) バケットとドメインと同じ名前に設定する。
- 2) CNAMEレコードを利用してドメインを設定する。
- 3) エイリアスレコード（Aレコード相当）を利用してドメインを設定する。
- 4) エイリアスレコード（AAAAレコード相当）を利用してドメインを設定する。
- 5) オブジェクトとドメインと同じ名前に設定する。

# Route53によるドメイン設定

S3の静的WEBホスティングのサイトにドメインを設定できる。

- トライフィック先としてS3 Webサイトエンドポイントへのエイリアス[Region(地域)]を選択します。
- レコードタイプとしてエイリアスレコードのA レコード（IPv4）タイプを利用してドメインを設定する。
- ターゲットの正常性の評価にはデフォルト値を設定する。
- バケット名とドメイン名またはサブドメイン名と同じにすることが必要

# [Q] クロスオリジンリソースシェアリング(CORS)

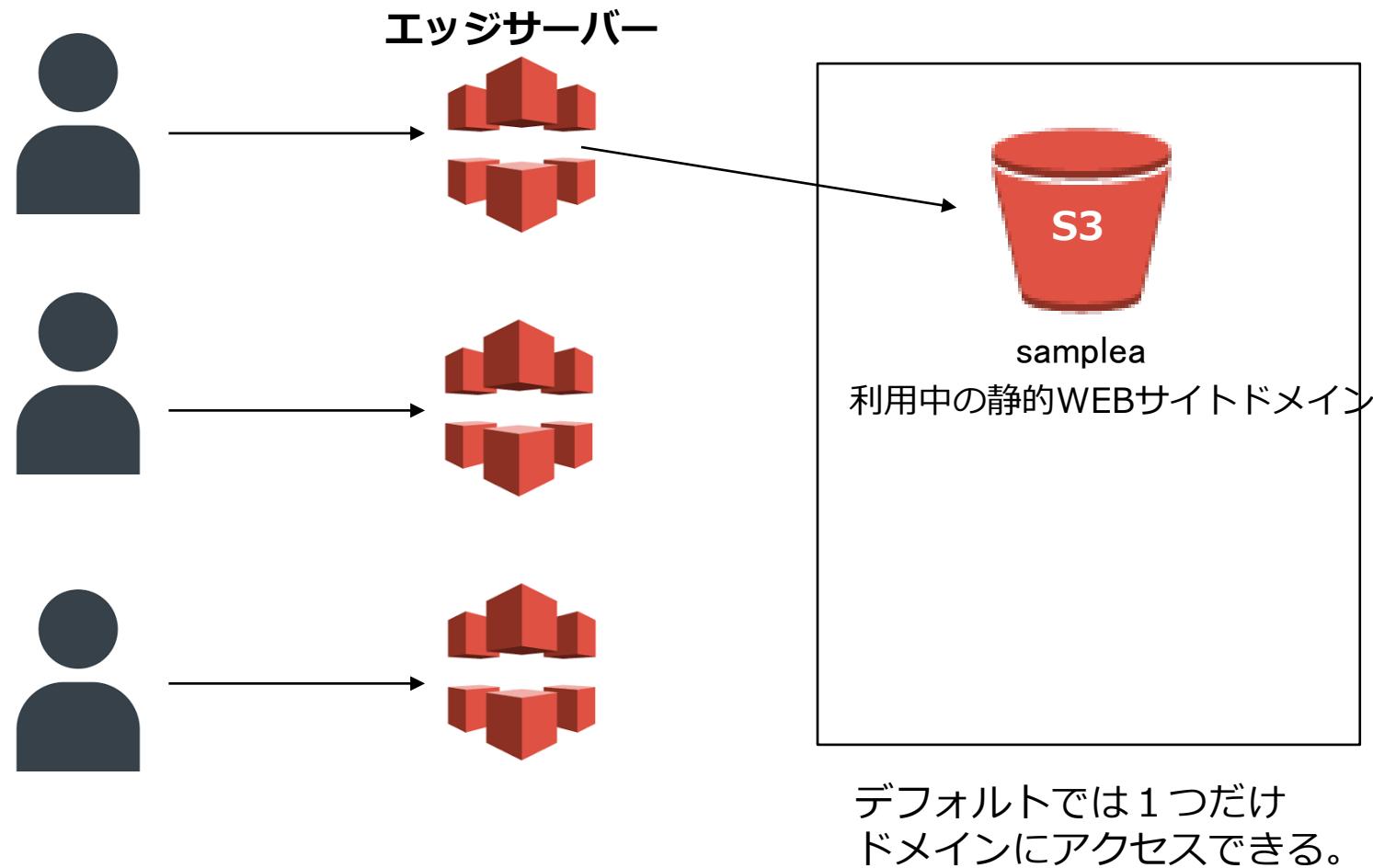
あなたの会社ではS3を利用したドキュメント管理システムを構築しています。このシステムはドメインを利用してユーザーからアクセスされていますが、ファイルを他のドメインから連携して、利用する機能が必要です。

この要件を満たすためのソリューションを選択してください。

- 1) レプリケーション
- 2) クロスアカウントアクセス
- 3) クロスオリジンリソースシェアリング (CORS)
- 4) S3アクセスポイント

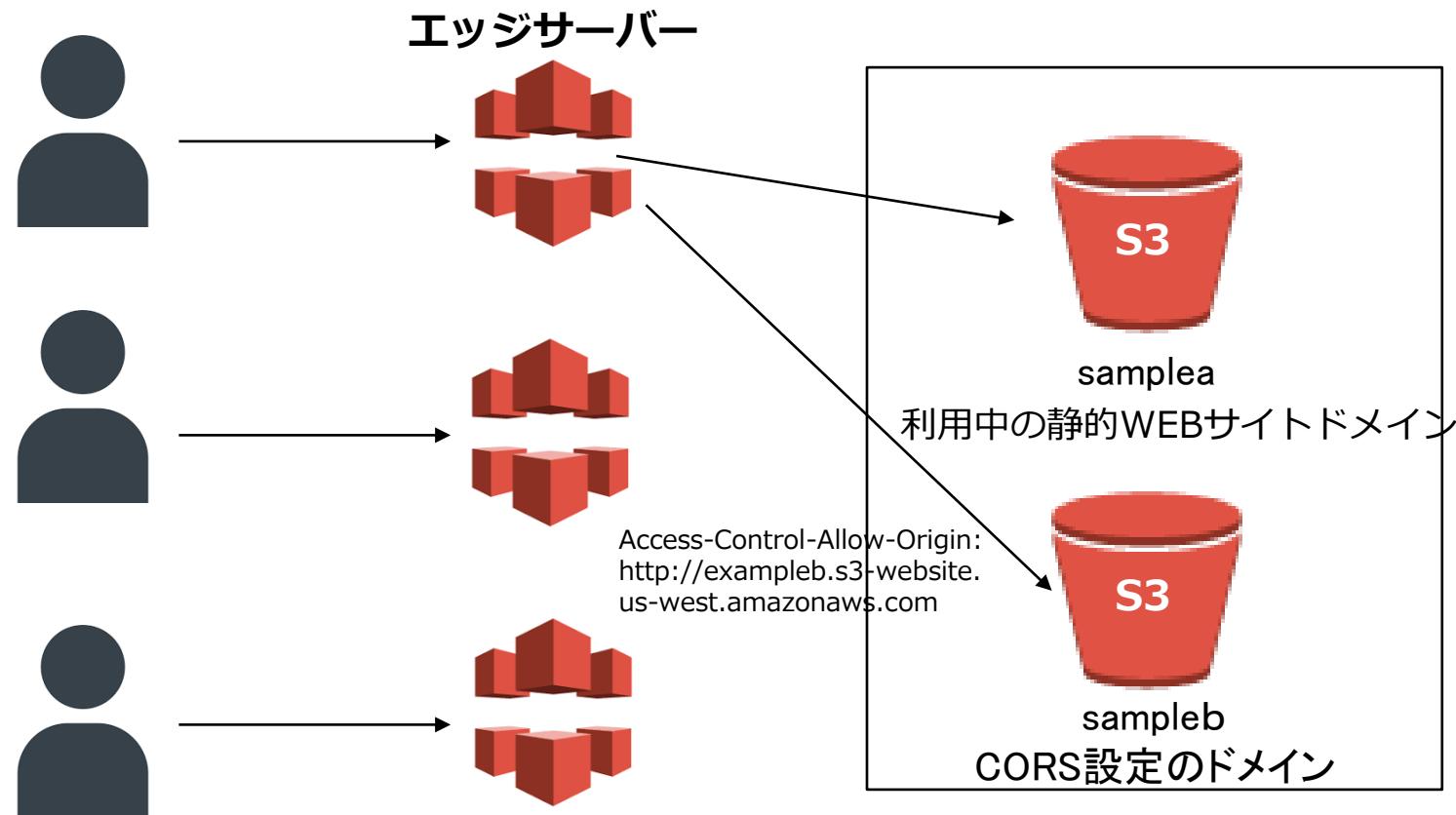
# CORS

クロスオリジンリソースシェアリング (CORS) により、特定のドメインにロードされたアプリケーションが異なるドメイン内のリソースと通信する方法を定義



# CORS

クロスオリジンリソースシェアリング (CORS) により、特定のドメインにロードされたアプリケーションが異なるドメイン内のリソースと通信する方法を定義



CORSヘッダーを付けてリクエストすることで複数ドメインにアクセスできる。



# [Q] S3イベント

あなたは写真共有アプリケーションをAWS上に構築しています。写真はS3バケットに保存され、画像処理を複数のEC2インスタンスにホストされたアプリケーションが実行します。ソリューションアーキテクトは、アップロードされたデータに応じて、EC2インスタンスのうちの1つで画像処理を実行する仕組みを構成しています。

要件を満たすために、どのようにS3と他のAWSサービスを構成するべきでしょうか？

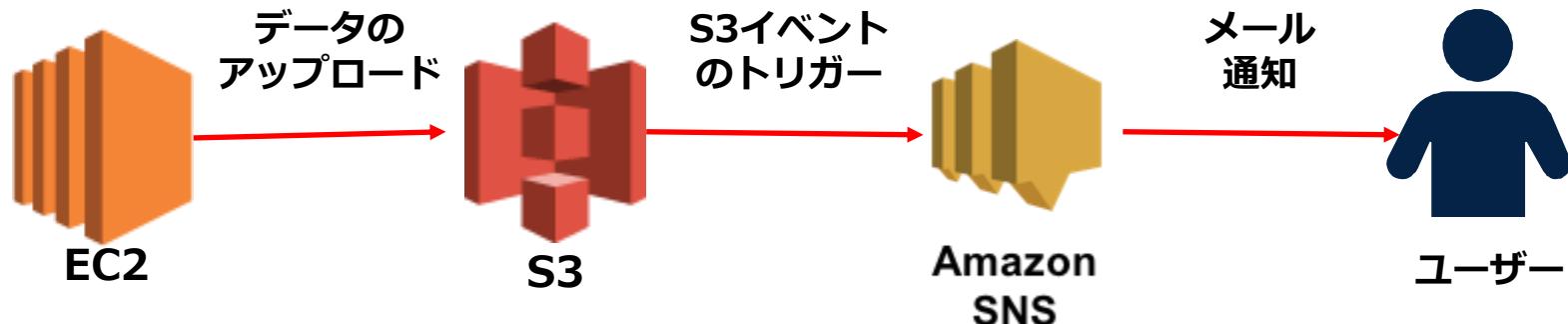
- 1) データアップロードをトリガーとするS3イベント通知を作成して、SQSを起動する。SQSキューからの処理メッセージをEC2インスタンスがポーリングして、画像処理を並行処理する。
- 2) データアップロードをトリガーとするS3イベント通知を作成して、SNSを起動する。SNSメッセージをトリガーにしてEC2インスタンスが画像処理を並行処理する。
- 3) データアップロードをトリガーとするS3イベント通知を作成して、Lambda関数を起動する。Lambda関数をトリガーにしてEC2インスタンスが画像処理を並行処理する。
- 4) データアップロードをトリガーとするS3イベント通知を作成して、SWFを起動する。SWFをトリガーにしてEC2インスタンスが画像処理を並行処理する。

# S3イベント

S3オブジェクト操作と連動したシステム連携処理を実現

## S3のイベント通知

- バケット内イベントの発生をトリガーにして、SNS／SQS／Lambda/Amazon EventBridgeに通知設定が可能
- S3オブジェクト操作と連動したシームレスなシステム連携処理を実現
  - S3へのデータアップロードをSNSでメッセージ通知
  - S3オブジェクトのアップロードをトリガーにLambda関数を実行



# [新Q] S3の暗号化

ある企業が、AWSを利用したデータ共有システムを構築しています。このシステムでは、機密データをAmazon S3バケットに保存します。コンプライアンス上の理由により、保管中のデータを暗号化する必要があります。その際は、セキュリティを高めるために暗号用のキーをロギングしつつ、毎年ローテーションする必要があります。その際は、キーの操作権限を有した上で、運用の手間をできる限りなくす必要があります。

最も費用対効果が高いソリューションはどれでしょうか。

- 1) 顧客提供のキーを使用したサーバーサイド暗号化を実施する。キーは1年に一回ローテーションを実施する。
- 2) Amazon S3マネージドキーを使用したサーバーサイド暗号化を実施する。キーはAWS側でローテーションされる。
- 3) AWS KMSキーを使用したサーバーサイド暗号化を実施する。キーには自動ローテーションを設定する。
- 4) AWS KMSキーを使用したサーバーサイド暗号化を実施する。キーは1年に一回ローテーションを実施する。

# S3の暗号化

S3へのデータ保管時に暗号化形式として以下の4つの形式から選択する

暗号化方式	特徴
SSE-S3	<ul style="list-style-type: none"><li>✓ S3の標準暗号化方式で簡易に利用可能</li><li>✓ 暗号化キーの作成・管理をS3側で自動で実施</li><li>✓ ブロック暗号の1つである256ビットのAdvanced Encryption Standard (AES-256) を使用してデータを暗号化</li></ul>
SSE-KMS	<ul style="list-style-type: none"><li>✓ AWS KMSに設定した暗号化キーを利用した暗号化を実施</li><li>✓ ユーザー側でAWS KMSを利用して暗号化キーを作成・管理することが可能</li><li>✓ クライアント独自の暗号キーを利用可能</li></ul>
SSE-C	<ul style="list-style-type: none"><li>✓ ユーザーが指定したキーによるサーバー側の暗号化 (SSE-C) を使用することが可能</li><li>✓ 利用設定や管理が煩雑になるのがデメリット</li></ul>
クライアントサイド 暗号化 (CSE)	<ul style="list-style-type: none"><li>✓ クライアント側の暗号化では、Amazon S3に送信する前にデータを暗号化する方式</li><li>✓ AWS KMSなどを利用して暗号化キーを作成・実施</li><li>✓ アプリケーション内に保存したマスターキーを使用</li></ul>



## [Q]レプリケーション

あなたの会社は複数リージョンを利用してAWSリソースを利用しています。現在シンガポールリージョンにAmazonS3バケットを設置し、大量のデータを保存していますが、このデータをシドニーリージョンにレプリケーションして、データのバックアップを実施したいと考えています。

次の中で、レプリケーションの正しい構成方法はどれでしょうか？（2つ選択してください。）

- 1) 両方のリージョンのバケットでバージョン管理を有効化する。
- 2) シンガポールリージョンにレプリケーション対象の新しいバケットを作成する。
- 3) シンガポールリージョンのバケットからのクロスオリジンリソースシェアリングを構成する。
- 4) シドニーリージョンに新たにS3バケットを作成し、クロスオリジンリソースシェアリングを構成する。
- 5) シドニーリージョンに新たにS3バケットを作成し、リージョン間のレプリケーションを構成する

# クロスリージョンレプリケーション

S3はリージョン間を跨ぐクロスリージョンレプリケーションにより耐障害性を高める

## レプリケーションのトリガー

- ✓ バケットにおけるオブジェクトの作成・更新・削除をトリガーにレプリケーションを実行する
- ✓ レプリケーション設定前のデータはレプリケートされない。

## 設定

- ✓ 事前にバージョニング機能を有効にする必要がある。
- ✓ レプリケーション先となるバケットは別リージョンに設置
- ✓ 双方向レプリケーションも可能
- ✓ データ転送費用が発生



# [新Q] S3データの解析

ある企業では、複数のEC2インスタンスで実行されているWEBアプリケーションを有しています。あなたはアプリケーションログファイルの蓄積と解析を実施する仕組みを構築しているところです。サーバー数を考慮するとログファイルは大量になる予定で、大量データに対する処理性能が求められます。また、このデータは中長期に保存することが必要ですが、データが必要となった場合にはミリ秒単位のアクセスが必要となります。

この要件を満たすためには、次のどのサービスを使用するのが最適ですか。（2つ選択してください。）

- 1) S3 Glacier Instant Retrievalにアプリケーションログファイルを保存して、Redshiftによって処理する。
- 2) S3にアプリケーションログファイルを保存して、Amazon EMRによって処理する。
- 3) S3にアプリケーションログファイルを保存して、S3 Selectによって処理する。
- 4) S3 Glacier Flexible Retrievalにアプリケーションログファイルを保存して、Lambda関数を利用して処理する。
- 5) S3にアプリケーションログファイルを保存して、Redshiftによって処理する。

# S3データの解析

S3内のデータ検索・解析には用途に応じて複数サービスから選択が可能

分析サービス	特徴
<b>S3 Select (Glacier Select)</b>	<ul style="list-style-type: none"><li>✓ S3の内部機能として有している検索機能で、S3内で直接にクエリを実行し、データを取得できる</li><li>✓ GZIP圧縮データやCSVやJSONに対して実行可能</li></ul>
<b>Amazon Athena</b>	<ul style="list-style-type: none"><li>✓ Amazon S3 内のデータを直接、簡単に分析できるようにするインタラクティブなクエリサービス</li><li>✓ Athena SQL クエリで SageMaker 機械学習モデルを呼び出し、機械学習による推論も実行可能</li></ul>
<b>Amazon Macie</b>	<ul style="list-style-type: none"><li>✓ 機械学習によりAmazon S3 の機密データを検出、分類、保護する、フルマネージド型サービス</li><li>✓ 機密データ検出や調査を実施する</li></ul>
<b>Amazon Redshift Spectrum</b>	<ul style="list-style-type: none"><li>✓ Amazon S3の格納データに対して、Amazon Redshiftから直接クエリを実行出来る機能</li><li>✓ Redshiftクラスターが起動されている前提であるため、Redshiftを利用している場合にお勧め</li></ul>



# S3データの解析

S3にビッグデータを蓄積して、EMRでビックデータ解析を実施



行動履歴データやログ  
ファイルゲノムデータ  
などを蓄積

Apacheによるビッグ  
データ解析を実施

解析結果をS3に保存

# S3 Select

SQLクエリを実行してS3バケット内のファイルを抽出・操作することができる。

```
Python
import boto3

s3 = boto3.client('s3')

resp = s3.select_object_content(
    Bucket='s3select-demo',
    Key='sample_data.csv',
    ExpressionType='SQL',
    Expression="SELECT * FROM s3object s where s.\"Name\" = 'Jane'",
    InputSerialization = {'CSV': {'FileHeaderInfo': "Use"}, 'CompressionType': 'NONE'},
    OutputSerialization = {'CSV': {}},
)

for event in resp['Payload']:
    if 'Records' in event:
        records = event['Records'][ 'Payload'].decode('utf-8')
        print(records)
    elif 'Stats' in event:
        statsDetails = event['Stats'][ 'Details']
        print("Stats details bytesScanned: ")
        print(statsDetails[ 'BytesScanned'])
        print("Stats details bytesProcessed: ")
        print(statsDetails[ 'BytesProcessed'])
        print("Stats details bytesReturned: ")
        print(statsDetails[ 'BytesReturned'])
```

```
Bash
python jane.py
```

以下の出力が得られます。

```
Jane,(949) 555-6704,Chicago,Developer

Stats details bytesScanned:
326
Stats details bytesProcessed:
326
Stats details BytesReturned:
38
```

- サーバー側でSQLクエリを実行することで必要なデータのみを取得する。
- ネットワークの転送量を削減して、処理コストや転送コストを削減する。

# [新Q] S3データの検索

あるコンサルティング企業は、グローバルにオフィスを展開しています。同社は多数のユーザーからの知見を共有するためにAWS上にナレッジデータベースを構築しようとしています。

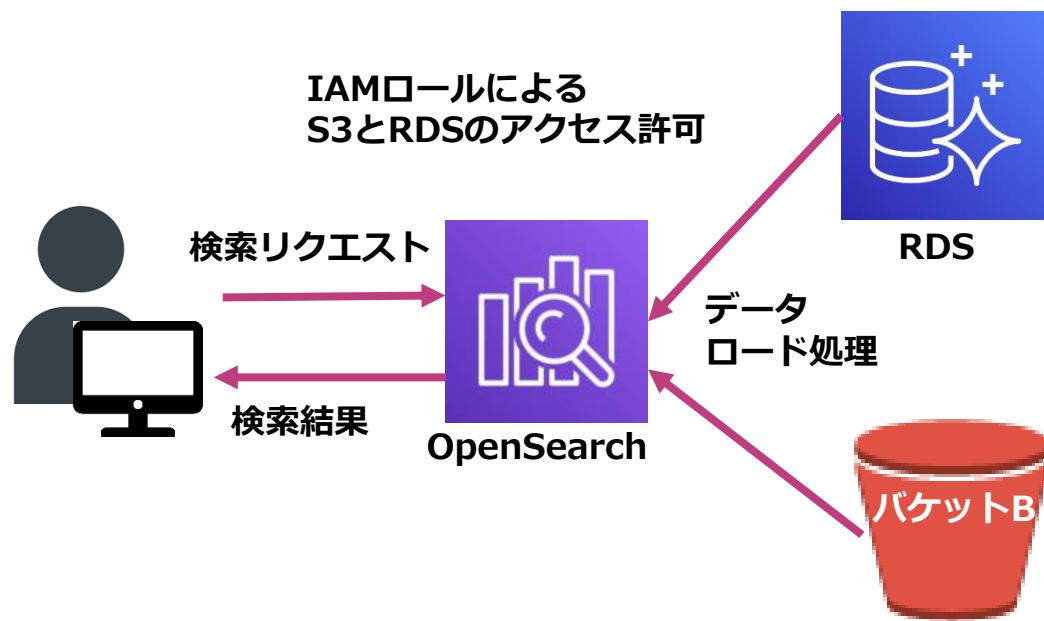
同社は設立から50年分のコンサルティングレポートがPDFファイルに記録されています。これらはグローバルにファイルをアップロードとダウンロードができる必要があります。グローバルに利用されるために、このデータベースシステムは24時間365日利用されるようです。また、ファイルを効率的に検索できるフレーズでの検索機能が必要です。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。

- 1) Amazon S3バケットにドキュメントデータを保存して共有し、Amazon OpenSearch Serviceを利用した検索機能を連携する。
- 2) Amazon S3バケットにドキュメントデータを保存して共有し、Amazon Athenaを利用した検索機能を連携する。
- 3) Amazon S3バケットにドキュメントデータを保存して共有し、Amazon S3 Selectを利用した検索機能を連携する。
- 4) Amazon S3バケットにドキュメントデータを保存して共有し、Amazon Redshiftを利用した検索機能を連携する。

# S3データの検索

Amazon OpenSearch Serviceを連携させて、Amazon S3バケット内のデータの検索機能を実装できる。



- OpenSearchは検索技術を提供するオープンソフトで、ログ分析、リアルタイムのアプリケーションモニタリング、クリックストリーム分析を提供
- Amazon OpenSearch Service はAWS上にOpenSearchクラスターのデプロイ、オペレーション、スケーリングを容易にするマネージドサービス
- 一方でAmazon CloudSearchはAWS独自の検索エンジンを提供
- OpenSearchはオープン検索結果
- S3バケット内のデータをOpenSearch側にロード処理して検索させる。
- S3にLambda関数を連携させて、リアルタイムにストリーミングデータをOpenSearchにロードさせることも可能

## [Q]利用状況の確認

あなたはソリューションアーキテクトとして、S3バケットを利用したドキュメント管理アプリケーションを構築しています。現在、ドキュメントデータを用いてレポートを生成する機能を追加開発しており、S3バケットへのすべてのリクエストアクセスとバケットのオブジェクトレベルの操作を詳細に把握できるようにする必要があります。

要件を満たすことができる最適な方法はどれでしょうか？

- 1) Amazon S3バケットにCloudWatchログを設定する。
- 2) Amazon S3バケットにS3アクセスアナライザーを有効にする。
- 3) Amazon S3バケットのサーバーアクセスログを有効にする。
- 4) Amazon S3バケットにCloudTrailを設定する。

# 利用状況の確認

S3の利用状況やS3のイベント発生を確認することができる

## S3の分析

- データのアクセスパターンの簡易可視化
- CSV形式で出力可能
- バケット内の分析を実施
- アクセス頻度の低いデータや保存期間を確認して、ライフサイクルポリシー設定に活かしていく

# サーバーアクセスログ

S3にアクセスした際のログを取得することが可能。バケットと  
プレフィックスをターゲットに設定する。

<input type="checkbox"/>	名前	最終更新日時
<input type="checkbox"/>	2019-05-10-17-55-948B7CEB7E063A7D	5月 10, 2019 7:17:5 GMT+0900
<input type="checkbox"/>	2019-05-10-18-11-DDD3C1EB69550551	5月 10, 2019 7:18:1 GMT+0900
<input type="checkbox"/>	2019-05-10-19-46-5B472ED82D8B552A	5月 10, 2019 7:19:4 GMT+0900
<input type="checkbox"/>	2019-05-10-20-00-1F137BA23771B806	5月 10, 2019 7:20:0 GMT+0900

- 監査対応などS3バケットのアクセスを記録する場合などに利用
- S3へのリクエストをすべてログとして記録して、ロギング用のバケットに保存する。
- ログ対象バケットとログ保存バケットは同じにしてはいけない。
- Amazon Athenaなどのデータ分析ツールでログ解析可能

# S3アクセスアナライザー

S3のアクセス状況がアクセスポリシーに沿っているか確認し、不正なアクセスの有無を監視する

- ✓ IAM アクセスアナライザーと連動したS3向けの機能
- ✓ バケットポリシー／ACLに沿ってポリシー違反がないかをモニタリング
- ✓ パブリックバケットまたは共有バケットアクセスを解析して、その解析結果を表示する
- ✓ バケットアクセスのソースを検証する場合は、列の情報を使用して、迅速で正確な措置を実行する
- ✓ バケットの実際のアクセス状況を確認する。



# [Q] S3の読み取り整合性モデル

あなたの会社はWEBアプリケーションをAWS上で運用しています。このアプリケーションではログファイルをAmazonS3に保存しています。このログファイルは広告表示のリアルタイム処理で利用されているため、頻繁に読み取り処理が発生していますが、ログファイルに変更が発生した際に、古いログファイルが読み取られてしまうようです。

この問題の最も可能性がある原因はどれでしょうか？

- 1) S3バケットでは既存のオブジェクトを置換し、すぐにそのオブジェクトの読み取ると、変更が完全に反映されるまで古いデータを返すことがある。
- 2) S3バケットでは既存のオブジェクトを置換し、すぐにそのオブジェクトの読み取ると、読み取工ラーが発生することがある。
- 3) S3バケットは強い整合性モデルを利用しているため、更新中のオブジェクトデータを読み取ることができないため、古いデータが表示されてしまう。
- 4) S3バケットはオブジェクトの共有を設定しないと、更新中のオブジェクトデータを読み取ることができないため、古いデータが表示されてしまう。

# S3の整合性モデル

S3はデータ登録・更新・削除などの処理時に強い整合性モデルを採用している。

データ処理	整合性モデル
新規登録	<ul style="list-style-type: none"><li>✓ Consistency Read</li><li>✓ 登録後即時にデータが反映される</li></ul>
更新	<ul style="list-style-type: none"><li>✓ 2020年12月より強い整合性モデルに変更された。そのため、齟齬は発生しない。</li></ul>
削除	<ul style="list-style-type: none"><li>✓ 2020年12月より強い整合性モデルに変更された。そのため、齟齬は発生しない。</li></ul>

# [Q]アップロード時のデータ整合性確認

AIベンチャー企業はAIベースの顔認識アプリケーションを構築しています。顔認証を実現するためにS3バケットに数百万の画像を保存して、これを利用した顔認識の学習を行います。新規に顔認証対象ユーザーを登録するには、S3バケットに対象ユーザーの顔写真を追加することが必要です。その際にアップロードされた画像が変更されることなく、整合性を保ったまま保存されていることが重要です。

オブジェクトが正常に保存されたことを示すために、何を実施すれば良いでしょうか？

- 1) S3バケットの整合性チェックを有効化する。
- 2) S3 API呼び出しで、HTTP200結果コードとMD5チェックサムを取得する。
- 3) S3イベントを設定して、アップロード後にAmazon SNSによるメッセージ通知を実施する。
- 4) S3のプレフィックスにおいてハッシュ値を設定して、整合性を確認する。

# アップロード時のデータ整合性確認

Content-MD5 ヘッダーを使用してアップロードされたオブジェクトの整合性を確認することができる。

1. オブジェクトの base64 でエンコードされた MD5 チェックサム値を取得します。
2. アップロード中のオブジェクトの整合性を確認します。

ただし、アップロードが AWS 署名バージョン 4 で署名されている場合、代わりに x-amz-content-sha256 ヘッダーを使用する必要があります。

# [Q] アップロードの高速化

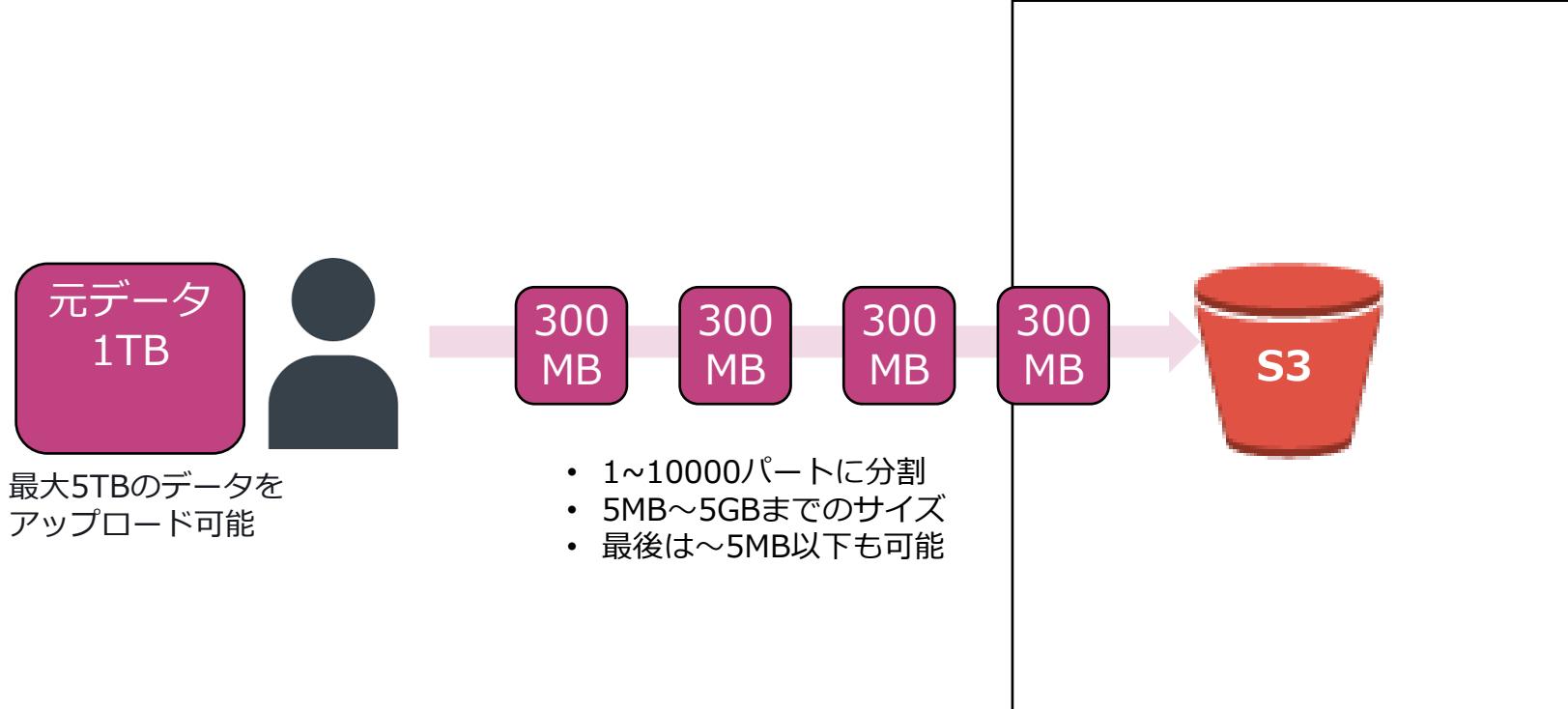
あなたはソリューションアーキテクトとして、AWS上で動画共有アプリケーションを構築しています。このアプリケーションはAmazon S3バケットに保存されたビデオデータを利用する動画処理アプリケーションをEC2インスタンスにホストする構成をとっています。利用ユーザーはグローバルに存在しており、大容量なデータがアップロードされます。そのために、大きなビデオファイルを宛先のS3バケットにアップロードするのが大幅に遅れており、クレームが発生しています。

S3へのファイルのアップロード速度を向上させる方法を選択してください。（2つ選択してください。）

- 1) Amazon S3 Transfer Accelerationを使用して、宛先S3バケットへのファイルのアップロードを高速化する。
- 2) Direct Connectを利用してS3へのファイルのアップロードを高速化する。
- 3) AWS Global Acceleratorを使用して、宛先S3バケットへのファイルのアップロードを高速化する。
- 4) AWS Transit Gatewayを使用して、宛先S3バケットへのファイルのアップロードを高速化する
- 5) マルチパートアップロードを利用してアップロードを高速化する。

# マルチパートアップロード

大容量オブジェクトをいくつかに分けてアップロードする機能



## 【失敗した場合】

- アップロードを中止するとパートデータが残る
- ライフサイクル管理でクリーンアップ設定が可能



# S3 Transfer Acceleration

地理的に一番近いエッジロケーションを利用して高速にデータアップロードを実施する。



## [Q]パフォーマンスの向上

リクエストは数百から2000程度まで同時に実行される可能性があり、パフォーマンスを効率化する処理が必要です。

ソリューションアーキテクトとして、最適なパフォーマンス向上策を選択してください。 (2つ選択してください)

- 1) 単一のバケット内に固有のカスタムプレフィックスを作成し、それらのプレフィックス付きの日次ファイルをアップロードする。
- 2) 単一のバケット内でTransfer Accelerationを有効化した上で、ファイルをアップロードする。
- 3) S3のマルチパートアップロードを有効化して、アップロード処理を実行する。
- 4) 単一のバケット内にハッシュを利用したランダムなカスタムプレフィックスを作成したファイルをアップロードする。

# パフォーマンスの向上

## 並列リクエストとカスタムプレフィックスでパフォーマンスを向上させる

### 並列リクエストの実行

- 並列リクエストを Amazon S3 サービスエンドポイントに水平にスケールすることでリクエストを分散し、ネットワーク経由で複数のパスに負荷を分散する
- 複数の接続で データを同時に GET または PUT するアプリケーションを使用することで高スループット転送が可能

### カスタム プレフィックスの利用

- パフォーマンスを最適化するためにカスタムプレフィックスを設定して、日付ベースの順次命名を使用する。
- 1 秒あたり 3,500 回以上の PUT/COPY/POST/DELETE リクエストと 5,500 回の GET/HEAD リクエストを送信可能
- 区切り記号としてスラッシュ (/) を使用できます。以下の例ではスラッシュ (/) 区切り記号を使用してプレフィックスを設定
  - Europe/France/Nouvelle-Aquitaine/Bordeaux
  - North America/Canada/Quebec/Montreal
  - 北米/米国/ワシントン州/ベルビュー
  - 北米/米国/ワシントン州/シアトル

# バックアップ

Glacierを利用してバックアップと復元が実施可能

## アーカイブ

- S3オブジェクトデータをライフサイクル設定によりGlacierに移動  
【データ紐づけ】
- S3 : 8KBオブジェクト/メタデータ
- Glacier : 32KBオブジェクト/メタデータ

## リストア

- オブジェクト毎に復元が可能
- 一時的に指定日数間複製する
- 復元に要する時間を選択
- 復元期間はGlacierで課金

# バッチオペレーション

S3 オブジェクトの大量データに対して一括処理を実行することが可能

## ジョブ

- ✓ ジョブはS3 バッチオペレーション の機能の基本単位で、ジョブを作成することでバッチオペレーションを作成
- ✓ ジョブにはオブジェクトのリストに対して指定された操作を実行するために必要なすべての情報を登録
- ✓ S3バッチオペレーション にオブジェクトのリストを渡し、それらのオブジェクトに対して実行するアクションを指定

## マニフェスト

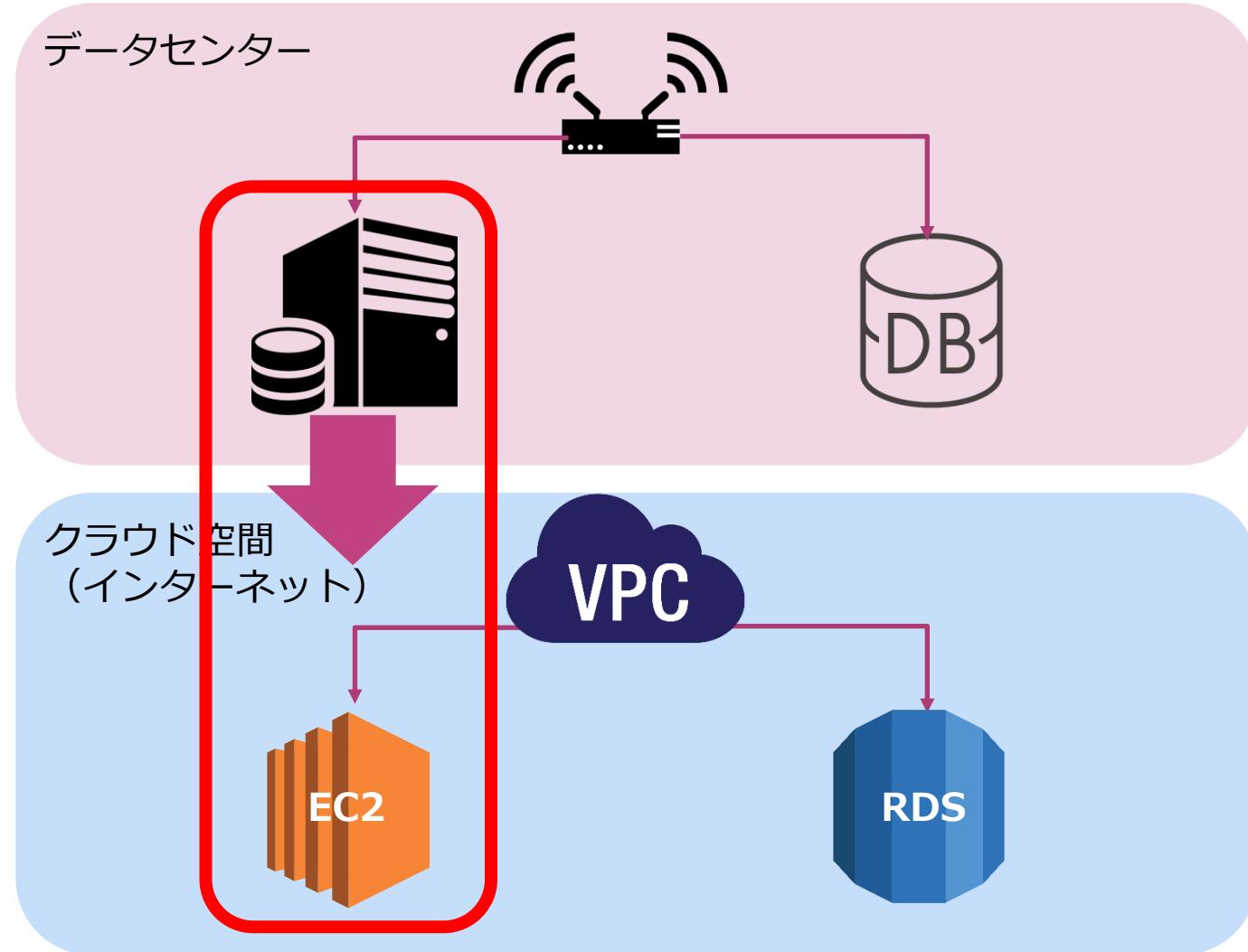
- ✓ マニフェストとは、Amazon S3 が作用するオブジェクトキーをリストする Amazon S3 オブジェクト
- ✓ マニフェストオブジェクトキー、ETag、およびオプションでバージョン ID を指定
- ✓ Amazon S3 インベントリレポート／CSVファイルの2つの形式で設定



## EC2の出題範囲

# EC2とは何か？

オンプレミス環境にあるサーバーと同じ性能を持ったサーバーをインターネット上で瞬時に作成することができるサービス



# EC2の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

EC2の特徴	✓ EC2を利用するケースが問われるシンプルな質問が問われる。
EC2の利用コスト	✓ EC2の利用コストを削減するための対応が問われる
AMIの利用	✓ AMIを別リージョンで利用するための方法が問われる。 ✓ AMIを利用して最適なEC2インスタンスを効率的に起動する方法が問われる。
インスタンスタイプの選択	✓ シナリオで利用したいインスタンスの要件が提示され、最適なインスタンスタイプが問われる。
ユーザーデータの利用	✓ EC2インスタンスを起動時にスクリプトを利用した自動設定を実行する方法が問われる。

# EC2の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

タグ設定	<ul style="list-style-type: none"><li>✓ EC2インスタンスに追加の情報を付与する機能を選択する質問が出題される。</li></ul>
キーペアの利用	<ul style="list-style-type: none"><li>✓ EC2インスタンスへのアクセスする認証方式が問われる。</li><li>✓ キーペアを他のアカウントやリージョンで使用する方法が問われる。</li></ul>
インターネットアクセス	<ul style="list-style-type: none"><li>✓ 起動したEC2インスタンスにインターネット経由でアクセスする際に必要な設定や方法が問われる。</li></ul>
インスタンスの購入形式	<ul style="list-style-type: none"><li>✓ インスタンスのコスト効率が良い購入形式の選択が問われる。</li><li>✓ シナリオに基づいて最適なインスタンスの選択が問われる。</li></ul>
リザーブドインスタンスの特徴	<ul style="list-style-type: none"><li>✓ リザーブドインスタンスのタイプや特徴に基づいて、タイプを選択する質問が問われる。</li><li>✓ リザーブドインスタンスの販売や属性変更に関する内容が問われる。</li></ul>

# EC2の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

スポットインスタンス の特徴	✓ シナリオに基づいてスポットインスタンスの特徴を選択する質問が出題される。
スポットフリート の利用	✓ スpotトフリートを利用してスポットインスタンスを購入する構成方法が問われる。
スポットブロック の利用	✓ シナリオに基づいて要件を満たすために、スポットブロックを選択する質問が出題される。
EC2フリート	✓ シナリオに基づいて要件を満たすために、EC2フリートを選択する質問が出題される。
プレイスメントグループ の利用	✓ シナリオに基づいて要件を満たすために、クラスタープレイスマントグループを選択する質問が出題される。 ✓ プレイスマントグループのタイプを選択する必要が問われる。

# EC2の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

拡張ネットワーキング	✓ EC2インスタンスのネットワークを高パフォーマンスする設定方法が問われる。
Elastic Fabric Adapter の利用	✓ HPCワークロードなどのユースケースを実現するために、Elastic Fabric Adapterを選択する質問が出題される。
EC2の自動リカバリー	✓ シナリオに基づいて要件を満たすために、EC2インスタンスがリカバリーした際のステータス状況に関する質問が出題される。 ✓ EC2のバックアップの方法が問われる。
インスタンスの停止と起動	✓ インスタンス起動時のトラブルに関する質問が出題される。 ✓ インスタンスの停止・起動に関するインスタンスのステータス状況に関する質問が出題される。
ハイバネーション	✓ EC2インスタンスでハイバネーションを実行する際の目的が問われる。

# [Q]EC2の特徴

ベンチャー企業はAWS上にWEBアプリケーションを有しています。このWEBアプリケーションはRDSをデータベースとして利用して、毎日午前7時にバッチジョブを実行しています。その際に、過去1日の業務オペレーションのログファイルを処理して、シェルスクリプトを介してバッチジョブで多数のレコードを順次実行することが必要です。この処理には1時間以上かかるため、バッチジョブの負荷は高いです。

このバッチジョブを実行するために、どのコンピューティングエンジンを利用するべきでしょうか？

- 1) AWS Lambda
- 2) Amazon EC2
- 3) Amazon EMR
- 4) Fargate

# EC2の特徴

数分で利用可能となる従量課金（時間～秒単位）で利用可能な  
仮想サーバー

- 起動・ノード追加・削除・マシンスペック変更できる仮想サー  
バーや提供する。
- WindowsやLinuxなどのほとんどのOSをサポート
- OSまでは提供されているタイプを選択することで自動設定さ  
れ、OSより上のレイヤーを自由に利用可能
- 独自のAmazon Machine Image (AMI) にバックアップして、  
再度起動させることが可能
- EC2はEBS・ELB・Auto Scalingを含む



# EC2の特徴

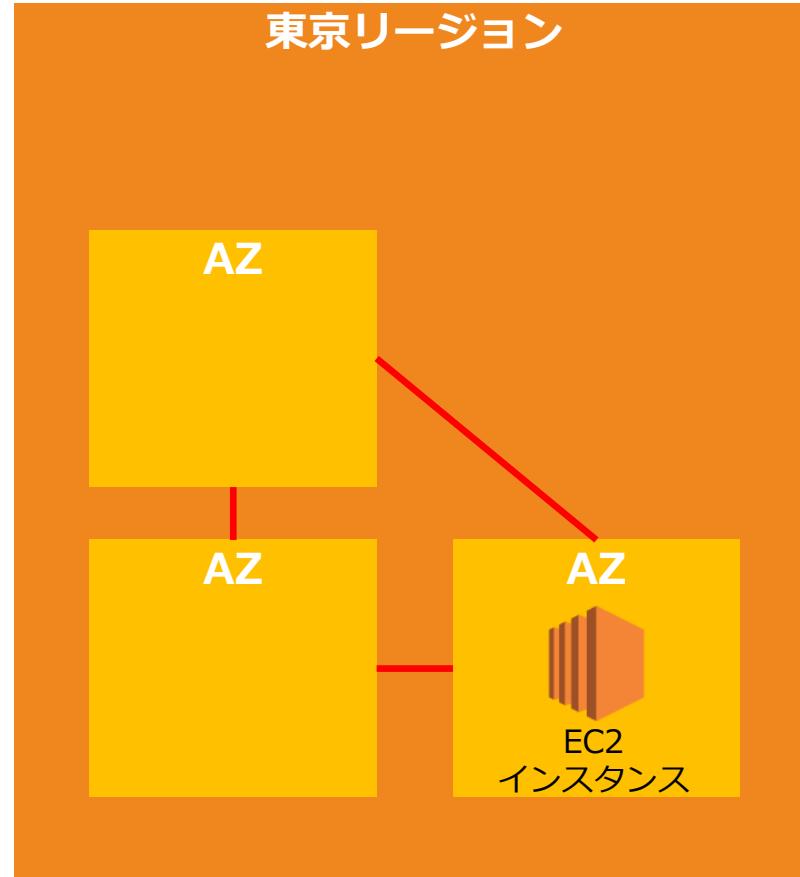
EC2ではOSからCPU、メモリ、ストレージなどの構成を選択して、仮想サーバーを起動することができる。

- オペレーティングシステムとしてのイメージの選択
- インスタンスタイプの設定
  - CPUのクロック数とコア数
  - メモリ容量
- ストレージタイプと容量
  - ネットワークディスク (EBS)
  - 物理ディスク (インスタンスストア)
- セキュリティグループでのトラフィック管理



# EC2の特徴

EC2の利用する単位をインスタンスと呼び、任意のAZにインスタンスを立ち上げてサーバーとして利用する



# [Q] EC2の利用コスト

大手ECマース企業は多数のEC2インスタンスを利用してECサイトや業務処理などを実現しています。そのために、EC2インスタンスの利用コストが甚大となっており、あなたはソリューションアーキテクトとして、コスト最適化を実現するように依頼されました。あなたはEC2インスタンスの請求方式を確認して、最適な対応を検討することが必要です。

EC2のコスト発生に関する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) オンデマンドインスタンスが保留状態になってもコストが発生する。
- 2) スポットインスタンスが停止準備中にもコストが発生する。
- 3) オンデマンドインスタンスが休止中でもコストが発生する。
- 4) リザーブドインスタンスが終了状態となってもコストが発生する。
- 5) オンデマンドインスタンスが停止状態または休止状態になる準備をしている際はコストが発生する。

# EC2の利用コスト

EC2の利用コストはインスタンスタイプや購入方式に応じて価格帯が決定する。

## 購入方式

購入形式に応じて様々な利用料金が設定されている。

- ✓ オンデマンド：通常価格
- ✓ リザーブド／Saving Plan：予約と事前支払の割引を適用
- ✓ スポットインスタンス：最大90%割引を適用

## インスタンスタイプ に応じた料金設定

- ✓ インスタンスタイプおよび利用時間によって価格が決定される。
- ✓ a1.mediumは0.0255USD/時間

## 時間課金の設定

- ✓ 時間単位または秒単位 (最低 60 秒) での課金
- ✓ Linux インスタンスは秒単位での課金
- ✓ その他のインスタンスは時間単位での課金

# EC2の利用コスト

リージョンに応じて価格が異なり、利用時間に加えてデータ転送アウトにも課金される。

リージョン	<ul style="list-style-type: none"><li>✓ リージョン：リージョン毎に価格が異なる。</li></ul>
データ転送	<ul style="list-style-type: none"><li>✓ データ転送イン：無料</li><li>✓ インターネットへのデータ転送アウト（GBあたり）</li><li>✓ S3からAWS内のデータ転送アウト（GBあたり）</li></ul>
ボリューム	<ul style="list-style-type: none"><li>✓ アタッチされたEBSでのデータ容量にも課金される。インスタンスを停止してもEBS分は課金が継続されるために注意が必要。</li><li>✓ インスタンスストアには課金されない。</li></ul>

# EC2の利用コスト

EC2インスタンスの状態に応じて課金発生が異なる。



# EC2の利用コスト

EC2インスタンスを停止することで課金を抑えることが可能

Running／開始／再起動

- ✓ 実行時間に応じて料金が発生する。
- ✓ 利用中のEBSボリュームの料金が発生する。

停止／Stop

- ✓ EC2の料金発生は停止する。
- ✓ 利用中のEBSボリュームの料金が発生する。

終了／Terminate

- ✓ EC2の料金発生は停止する。
- ✓ デフォルト設定ではルートボリュームに設定されたEBSボリュームも削除され、料金発生は停止する。

# EC2の起動方法

EC2の起動は以下のステップで実行します。

AMI (OSセッティング) を選択

インスタンスタイプを選択

インスタンスタイプの詳細の設定

ストレージを選択

タグの追加

セキュリティグループを選択

キーペアを設定

# [新Q]Amazon Machine Image (AMI)

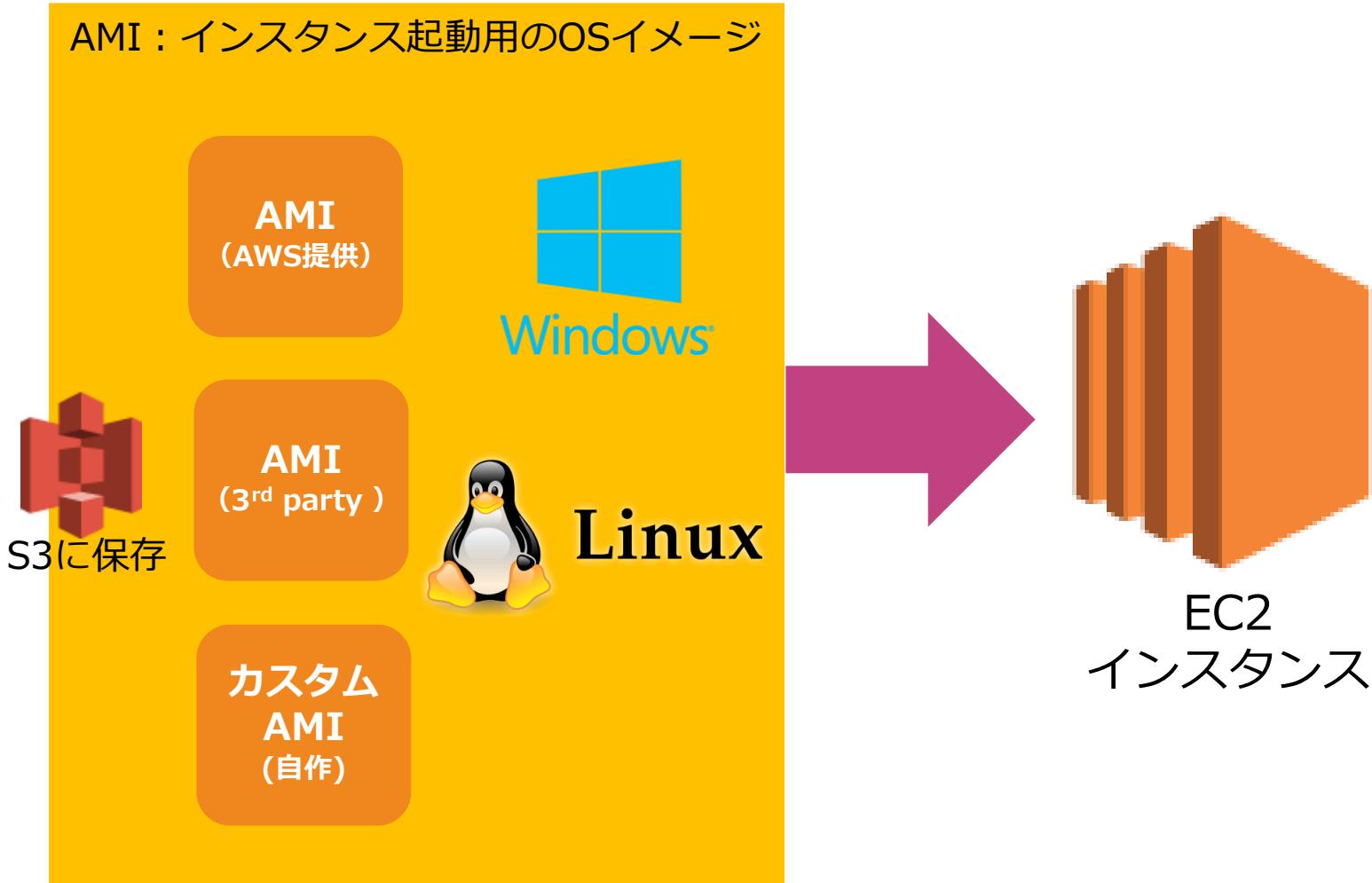
ある企業は、オンプレミス環境をAWSに移行することを決定しました。同社は、この移行作業を支援してもらうためにコンサルティング会社を雇いました。この移行作業のためには、同社のAWSアカウントが有するAmazon EC2のAMIをコンサルティング会社のAWSアカウントと共有する必要があります。このAMIはAmazon S3バケットにバックアップされており、AWS KMSのカスタマー管理キーを利用してスナップショットの暗号化を実施しています。ソリューションアーキテクトはこのAMIを共有する設定を行っています。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。

- 1) 暗号化されたAMIを公開して、コンサルティング会社のAWSアカウントがアクセスできるようにする。AWS KMSのキー policy を変更して、コンサルティング会社のAWSアカウントがKMSのキーを使用できるようにする。
- 2) AMIをコンサルティング会社のAWSアカウントとのみ共有できるようにAMIのLaunchPermissionプロパティを変更する。AWS KMSのキー policy を変更して、コンサルティング会社のAWSアカウントがKMSのキーを使用できるようにする。
- 3) AMIをコンサルティング会社のAWSアカウントとのみ共有できるようにAMIのLaunchPermissionプロパティを変更する。AWS KMSのキー policy を変更して、コンサルティング会社の所有する新しいKMSキーを許可して、このAMIに適用する。
- 4) 暗号化されたAMIを公開して、コンサルティング会社のAWSアカウントがアクセスできるようにする。AWS KMSのキー policy を変更して、コンサルティング会社の所有する新しいKMSキーを許可して、このAMIに適用する。

# AMI (OSセッティング) を選択

AMIはOSセッティング方式を選択すること



# AMI (OSセッティング) を選択

AMIはOSセッティング方式を選択すること

▼ アプリケーションおよび OS イメージ (Amazon マシンイメージ) [情報](#)

AMI は、インスタンスの起動に必要なソフトウェア設定 (オペレーティングシステム、アプリケーションサーバー、アプリケーション) を含むテンプレートです。お探しのものが以下に表示されない場合は、AMI を検索または参照してください。

[最新](#) [クイックスタート](#)



[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

Amazon マシンイメージ (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-Offac3e16de16665e (64 ビット (x86)) / ami-0b6ffdb4868a0bcac (64 ビット (Arm))  
仮想化: hvm ENA 有効: true ルートデバイスタイプ: ebs

無料利用枠の対象

説明

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230207.0 x86\_64 HVM gp2

Architecture: 64 ビット (x86)

AMI ID: ami-Offac3e16de16665e

検証済みプロバイダー

## [Q] AMIの活用

あなたはAWSに大量のEC2インスタンスを起動するタスクを依頼されました。効率的にタスクを実行するためには、同じ構成と同じ状態となる新しいコンピューティングリソースを展開するプロセスを自動化することが求められています。その際は、インスタンス内のソフトウェア構成は同じものにする必要がありますが、セキュリティグループなどの設定は個別に設定する必要があります。

この要件を満たすために、どのアプローチが適切ですか？（2つ選択してください。）

- 1) Bashスクリプトによるブートストラップを実行して、インスタンス起動時に最適な構成を自動化する。
- 2) AWSが提供するAMIの中から、最適なAMIを選択して、インスタンス構成を複製する。
- 3) マーケットプレイスで業者が提供する最適なAMIを購入して、インスタンス構成を複製する。
- 4) EC2 Image Builderによりゴールデンイメージを設定し、最適なインスタンス構成を複製する。
- 5) 起動テンプレートを活用して、最適なインスタンス構成を複製する。

# AMIの活用

EC2インスタンスはAMIを利用して起動・バックアップ・共有することができる。

OSの選択	<ul style="list-style-type: none"><li>✓ 利用したいサーバーのOSの選択としてAMIを利用</li><li>✓ 利用していたサーバーを復元する際にAMIを利用</li></ul>
EC2のバックアップ	<ul style="list-style-type: none"><li>✓ 既存のEC2インスタンスからAMIを作成できる。</li><li>✓ EC2インスタンスをバックアップとして構成内容を保存する。EBSボリュームのスナップショットも含まれる。</li></ul>
ゴールデンイメージ	<ul style="list-style-type: none"><li>✓ 最適なEC2インスタンスの構成をAMIとした上で、構成を複数利用することができる。</li><li>✓ 最適なEC2インスタンス構成を反映したAMIをゴールデンイメージと呼ぶ。</li></ul>
AMIの共有	<ul style="list-style-type: none"><li>✓ 共有したいユーザーのAWS アカウント番号を指定して、共有の設定することで他アカウントと共有できる</li></ul>
リージョンの移動	<ul style="list-style-type: none"><li>✓ AMIはリージョン内でのみ利用可能</li><li>✓ 別リージョンにコピーできる。その場合のAMIはそのリージョン固有AMIとして別のAMIとなる。</li></ul>

# [Q]インスタンスタイプの選択

大手ECマース企業はAWS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では、顧客情報を解析して最適な商品を提示する機能を作る必要があります。その際には、ローカルストレージ上の非常に大きなデータセットに対して高いシーケンシャルな読み取りおよび書き込みアクセスを必要とするワークフローを実行します。

このシナリオで使用するのに最適なインスタンスタイプは次のうちどれですか？

- 1) ストレージ最適化インスタンス
- 2) メモリ最適化インスタンス
- 3) コンピューティング最適化インスタンス
- 4) 汎用インスタンス

# インスタンスタイプの選択

インスタンスタイプの選択では、CPU・メモリ、ストレージ、ネットワークキャパシティなどのサーバーリソースを選択する

Amazon マシンイメージ (AMI)

検索

インスタンスタイプ	アーキテクチャ	CPU	メモリ	ストレージ	ネットワーク	オペレーティングシステム	料金
t1.micro	アマゾン Linux	1 vCPU	0.612 GiB	10 GiB SSD	1 Gbps	SUSE, Windows, RHEL, Linux	0.026 USD /時間
t2.nano	アマゾン Linux	1 vCPU	0.5 GiB	10 GiB SSD	1 Gbps	SUSE, Windows, RHEL, Linux	0.0099 USD /時間
t2.micro	アマゾン Linux	1 vCPU	1 GiB	10 GiB SSD	1 Gbps	SUSE, Windows, RHEL, Linux	0.0198 USD /時間
t2.small	アマゾン Linux	1 vCPU	2 GiB	10 GiB SSD	1 Gbps	SUSE, Windows, RHEL, Linux	0.0396 USD /時間
t2.micro	アマゾン Linux	1 vCPU	1 GiB	10 GiB SSD	1 Gbps	SUSE, Windows, RHEL, Linux	0.0198 USD /時間

無料利用枠の対象

バイダー

インスタンスタイプを比較

# インスタンスタイプ

t2.nano

ファミリーと世代 インスタンスの容量

# インスタンスファミリー

## ユースケースに応じてインスタンスタイプを選択する。

汎用	ファミリー：A1、M5、T3など バランスの取れたコンピューティング、メモリ、ネットワークのリソースを提供し、多様なワークロードに使用。ウェブサーバーやコードリポジトリなど、インスタンスのリソースを同じ割合で使用するアプリケーションに最適なインスタンス
コンピューティング最適化	ファミリー：C5、C6gなど 高パフォーマンスプロセッサが必要なコンピューティングバウンドなアプリケーションに利用。ユースケースはバッチ処理ワークロード、メディアトランスクード、高性能ウェブサーバー、ハイパフォーマンスコンピューティング（HPC）、科学モデリング、専用ゲームサーバーおよび広告サーバーエンジン、機械学習推論
メモリ最適化	ファミリー：X1、R5、ハイメモリ、z1dなど メモリ内の大きいデータセットを処理するワークロードに対して高速なパフォーマンスに最適なインスタンス
ストレージ最適化	ファミリー：H1、D2、I3、I3enなど ローカルストレージの大規模データセットに対する高いシーケンシャル読み取りおよび書き込みアクセスを必要とするワークロード用。ストレージ最適化インスタンスは、数万 IOPS の低レイテンシーなランダム I/O オペレーションに最適
高速コンピューティング	ファミリー：P3、Inf1、G4（GPU）、F1（FPGA）など 高速コンピューティングインスタンスはハードウェアアクセラレーター（コプロセッサ）を使用して、浮動小数点計算、グラフィックス処理、データパターン照合などの機能をCPUで実行するソフトウェアに最適

# [Q]ユーザーデータの利用

ある企業はAWS上でEC2インスタンスを利用したWEBアプリケーションを構築しています。これらのEC2インスタンスを起動する際に、全てのインスタンスで利用することになるApacheサーバーを自動的に設定することが必要です。

この要件を満たすために利用するべきEC2インスタンスの機能を選択してください。

- 1) ユーザーデータを利用して、インスタンス起動時に最適な構成を自動化する。
- 2) メタデータを利用して、インスタンス起動時に最適な構成を自動化する。
- 3) タグを利用して、インスタンス起動時に最適な構成を自動化する。
- 4) 自動設定機能を有効化して、インスタンス起動時に最適な構成を自動化する。

# ユーザーデータの利用

ユーザーデータを利用してEC2インスタンス起動時に実行されるスクリプトを設定できる。

ユーザーデータ

- ✓ EC2インスタンスの詳細設定の自動化に利用
- ✓ Bashスクリプトなどを設定して、インスタンス起動時に実行されるように準備できる。

ブートストラップ

- ✓ インスタンスにユーザーデータを渡すことで、起動時に実行される処理のこと

# ユーザーデータの利用

インスタンスの詳細設定において、高度な詳細にあるユーザーデータを利用して実行スクリプトを設定することが可能

The screenshot shows the 'Create Function' wizard in the AWS Lambda console. The 'User Data' field is highlighted with a large red circle. The field contains the placeholder text 'Enter user data in the field.' Below the field, a note states 'ユーザーデータは既に base64 エンコードされています' (User data is already base64 encoded).

アクセス可能なメタデータ [情報](#)  
選択

メタデータのバージョン [情報](#)  
選択

メタデータレスポンスのホップ制限 [情報](#)  
選択

メタデータのタグを許可 [情報](#)  
選択

ユーザーデータ - optional [情報](#)  
Enter user data in the field.

ユーザーデータは既に base64 エンコードされています

インスタンス数 [情報](#)  
1

ソフトウェアイメージ (AMI)  
Amazon Linux 2 Kernel 5.10 AMI...[続きを読む](#)  
ami-0ffac3e16de16665e

垂直サーバータイプ (インスタンスタイプ)  
t2.micro

ファイアウォール (セキュリティグループ)  
新しいセキュリティグループ

ストレージ (ボリューム)  
1 ボリューム - 8 GiB

① 無料利用枠: 最初の 1 年には、1 か月あたりの無料利用枠による AMI での t2.micro (または t2.micro が利用できないリージョンでは t3.micro) インスタンスの 750 時間の使用、30 GiB の EBS ストレージ、200 万の I/Os、1 GB のスナップショット、インターネットへの 100 GB の帯域幅が含まれます。 [×](#)

キャンセル [インスタンスを起動](#)

# ストレージの選択

EC2で直接利用するストレージを追加する。

▼ ストレージ (ボリューム) [情報](#) シンプル

EBS ボリューム [詳細を非表示](#)

▼ ボリューム 1 (AMI ルート)

ストレージタイプ <a href="#">情報</a> EBS	デバイス名 - <i>required</i> <a href="#">情報</a> /dev/xvda	スナップショット <a href="#">情報</a> snap-01a8a2055ed38ca72
サイズ (GiB) <a href="#">情報</a> <input type="text" value="8"/>	ボリュームタイプ <a href="#">情報</a> <input type="text" value="gp2"/>	IOPS <a href="#">情報</a> 100 / 3000
終了時に削除 <a href="#">情報</a> <input type="text" value="はい"/>	暗号化済み <a href="#">情報</a> <input type="text" value="暗号化なし"/>	KMS キー <a href="#">情報</a> <input type="text" value="選択"/> KMS キーは、このボリュームで暗号化が設定されている場合にのみ適用されます。

[新しいボリュームを追加](#)

ファイルシステム [詳細を非表示](#)

EFS  FSx

現在、このインスタンスにはファイルシステムがありません。EFS ファイルシステムを追加する前に、サブネットを選択する必要があります。



# ストレージの選択 (レートボリューム)

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・終了と共にデータは消去される
- ✓ 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はデータを保持可能で、SnapshotをS3に保存する。
- ✓ 別途EBS料金が必要



# ストレージの選択（オプション）

オプションでファイルシステム（ファイル形式のストレージ）を構成することもできる。

## Amazon 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

## 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



# [新Q]タグ設定

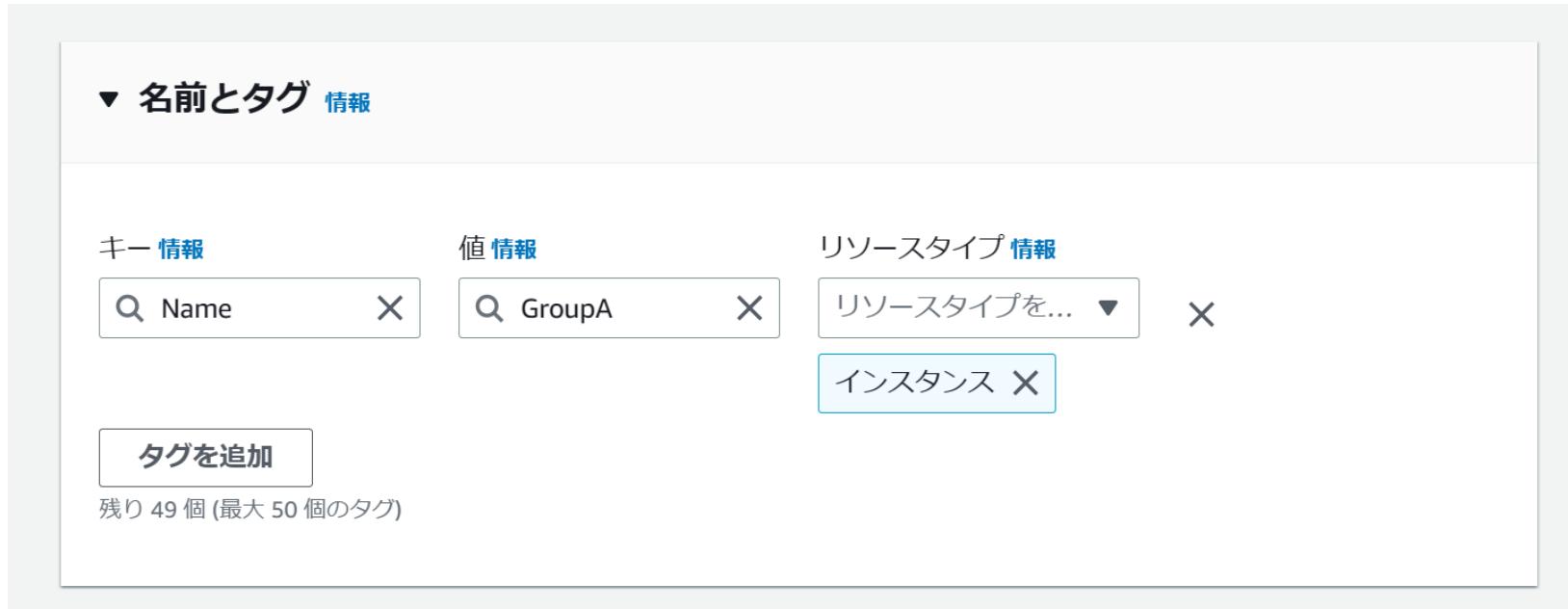
製造業のA社はAWS上にエンタープライズシステムをホストしています。最近になって、そのシステムが突如停止するという障害が発生しました。ソリューションアーキテクトが調査したところ、一人のエンジニアが本番環境のEC2インスタンスを誤って終了してしまい、サービス中断を引き起こしたことが判明しました。また、実稼働するアプリケーション向けのインフラ構成を操作できる開発者が多数存在するという事実も同時に判明しました。

この種の障害が再び発生するのを防ぐための、適切な対応はどれでしょうか？（2つ選択してください。）

- 1) すべてのEC2インスタンスに適切なタグ設定を行い、タグ情報に基づいて本番環境向けインスタンスの削除を拒否する権限を設定する。それによって、開発者の権限を所属するグループタグ内のリソース操作に限定する。
- 2) 開発者向けのIAMポリシーを修正して、EC2インスタンスの停止権限の許可ポリシーを削除する。
- 3) 開発者グループのIAMグループのIAMポリシーを修正して、EC2インスタンスへの操作権限を削除する。
- 4) AWS Organizationsを利用して開発者グループとそれ以外を分割する組織ルールを適用して、開発者がアクセスできるインスタンスを制限する。
- 5) 開発環境向けのVPCを新たに設置して、開発者のアクセス権限を開発者向けVPC内に制限する。その上で、IAMポリシーによってVPCごとに権限設定を割り振ることで、開発者グループが利用できるVPC内リソースを制限する。

# タグ設定

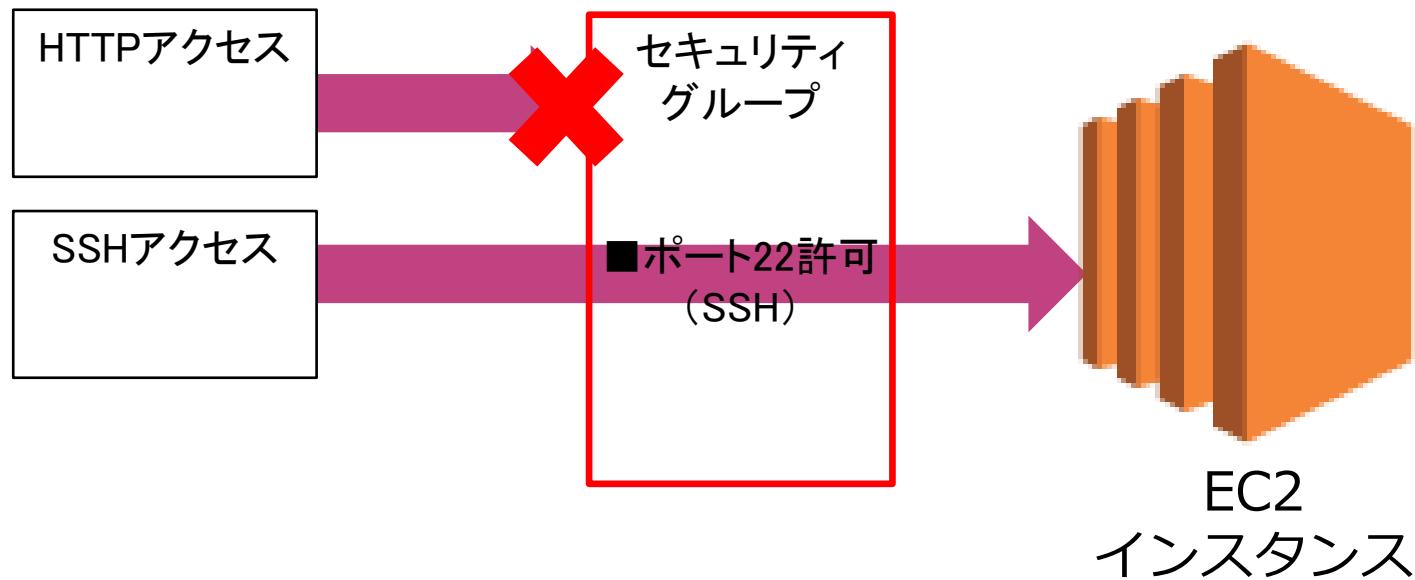
タグを追加で設定して名前を付与することができ、EC2などのリソースのグループ分けや権限分けに利用する。



- ・ タグ=キー+値のセット
- ・ 加えて、リソースタイプを選択して、どのリソースに設定したタグなのかを分類できる。

# セキュリティグループ

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



# [Q] キーペアの利用

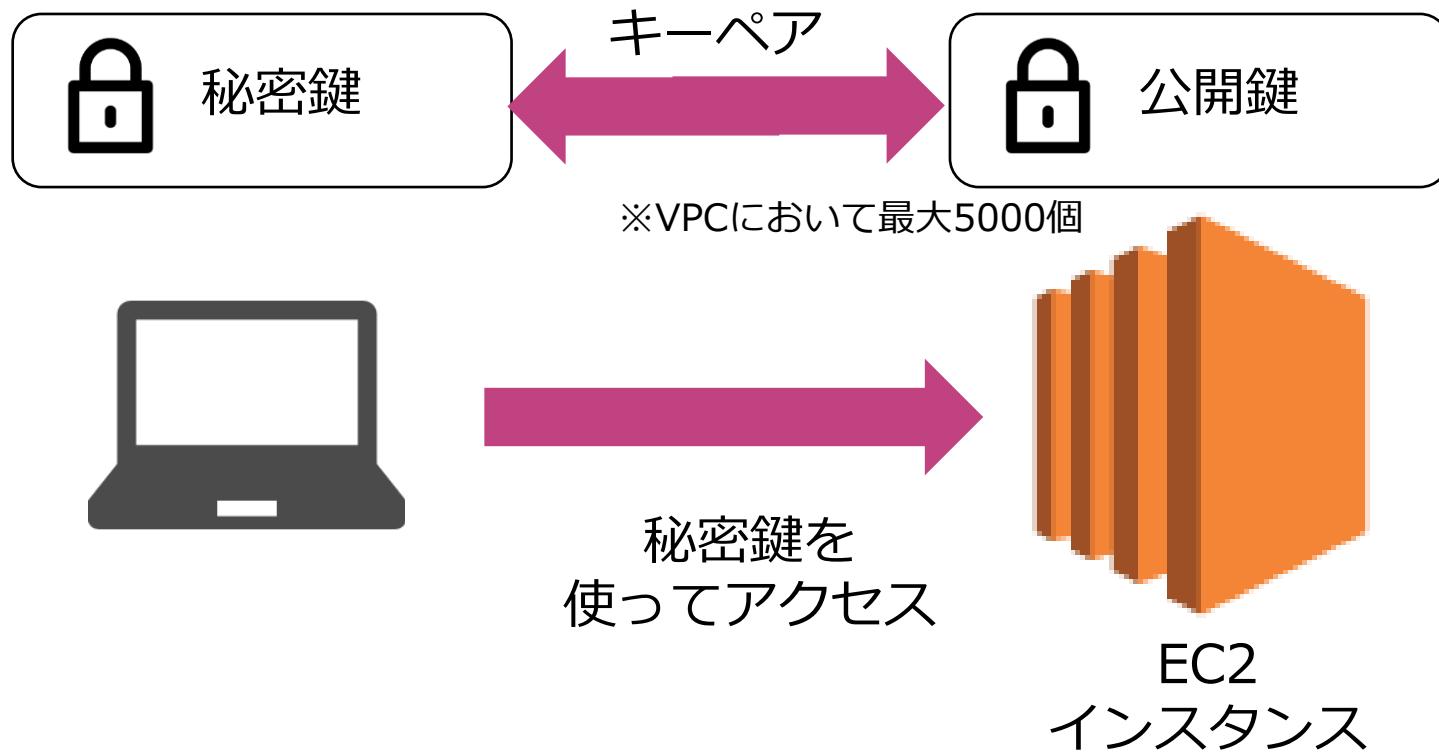
あなたはAWSアカウントを作成して、初めてLinux EC2インスタンスを起動しました。このインスタンスにアクセスしてサーバーソフトウェアをインストールして、WEBサーバーとして設定する対応が必要です。ローカル端末からインスタンスにアクセスして設定することになります。

インスタンスに安全にアクセスする方法を選択してください。

- 1) SSHソフトウェアを利用して、キーペアによる認証を実施してLinux EC2インスタンスにアクセスする。
- 2) SSHソフトウェアを利用して、アクセスキーとシークレットアクセスキーによる認証を実施してLinux EC2インスタンスにアクセスする。
- 3) SSHソフトウェアを利用して、ユーザー名にec2-userとして、インスタンス起動に設定したパスワードによりLinux EC2インスタンスにアクセスする。
- 4) SSHソフトウェアを利用して、AWSアカウントのユーザー名とパスワードによりLinux EC2インスタンスにアクセスする。

# キーペアの利用

キーペアを利用して自身がダウンロードした秘密鍵とマッチした公開鍵を有するインスタンスにアクセスする



## [Q]起動テンプレート

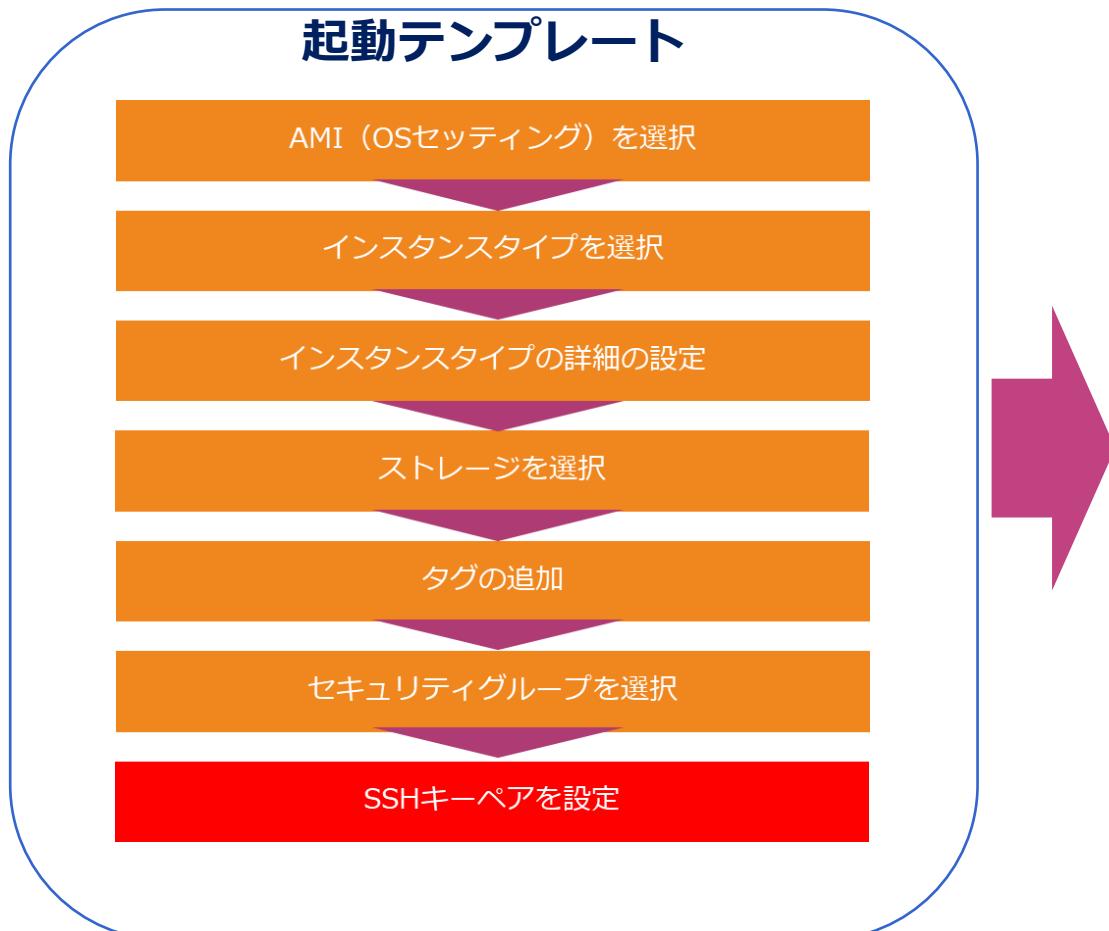
あなたはソリューションアーキテクトとして、社内のAWS利用を標準化する対応を実施しています。通常利用するEC2インスタンスの構成を事前に設定することで、EC2インスタンスを定期的に手動で起動し、プロセスを合理化して管理オーバーヘッドを削減する対応を検討しています。その際には、EC2インスタンスのAMIの選択、インスタンスタイプ、キーペア、セキュリティグループなどの設定を保存することが必要です。

この要件を満たすEC2インスタンスの機能を選択してください。

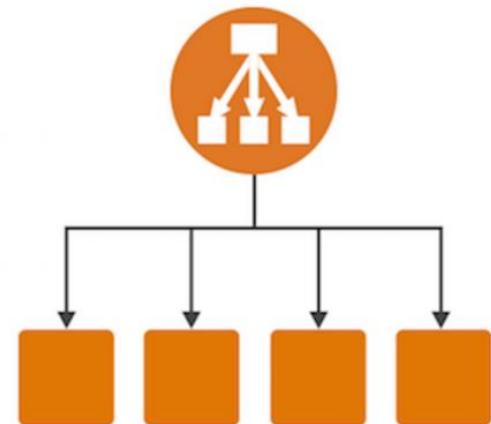
- 1) 起動テンプレートを利用する。
- 2) 起動設定を利用する。
- 3) AMIを利用する。
- 4) 起動グループを利用する。

# 起動テンプレート

起動テンプレートは起動の詳細な設定内容をテンプレート化して保存することができる。



Auto Scaling



# [Q]インターネットアクセス

ソリューションアーキテクトはAWSアカウントを新規に作成して、ITインフラストラクチャーを構成しています。Amazon VPCに新しいサブネットを作成し、そのサブネットにAmazonEC2インスタンスを起動しました。あなたはEC2インスタンスの設定をするために、インターネットからEC2インスタンスに直接アクセスを試みましたが、接続ができないようです。

EC2インスタンスへの接続失敗に対処するために、どの手順を確認する必要がありますか？（2つ選択してください）

- 1) パブリックサブネットにNATゲートウェイが設置されている。
- 2) セキュリティグループにアウトバウンドトラフィックのルールが適切に設定されている。
- 3) インスタンスにパブリックIPアドレスが設定されている。
- 4) インスタンスにプライベートIPアドレスが設定されている。
- 5) サブネットに関連付けられているルートテーブルにインターネットゲートウェイにインターネットへのアクセスルートが適切に構成されている。

# インターネットアクセス

起動したインスタンスにアクセスする際はパブリックIPを利用してアクセスする。

The screenshot shows the AWS Management Console interface for the EC2 service. At the top, there's a navigation bar with tabs for 'Instancesの作成', '接続', and 'アクション'. Below it is a search bar and a filter section for 'Name' and 'インスタンス ID'. A table lists one instance: 'i-0b02b822d692a0499' (t2.micro, ap-northeast-1c, running, 2/2のチェック, なし, ec2-54-199-94-166.ap-northeast-1.compute.amazonaws.com, 54.199.94.166). The main content area displays detailed information for this instance. The '説明' tab is selected. Key details shown include:

項目	値
インスタンス ID	i-0b02b822d692a0499
インスタンスの状態	running
インスタンスタイプ	t2.micro
プライベート DNS	ip-172-31-4-132.ap-northeast-1.compute.internal
プライベート IP	172.31.4.132
セカンダリプライベート IP	
VPC ID	vpc-940724f3
プラットフォーム	Amazon Linux
パブリック DNS (IPv4)	ec2-54-199-94-166.ap-northeast-1.compute.amazonaws.com
IPv4 パブリック IP	54.199.94.166
IPv6 IP	-
Elastic IP	
アベイラビリティーゾーン	ap-northeast-1c
セキュリティグループ	launch-wizard-5, インバウンドルールの表示, アウトバウンドルールの表示
予定されているイベント	予定されているイベントはありません
AMI ID	amzn2-ami-hvm-2.0.20200917.0-x86_64-gp2 (ami-0ce107ae7af2e92b5)
サブネット ID	subnet-51ffa00a

# パブリックIPとプライベートIP

インターネットと通信できるのがパブリックIPアドレス。内部ネットワーク内だけで通信できるのがプライベートIPアドレス。

インターネット

VPC

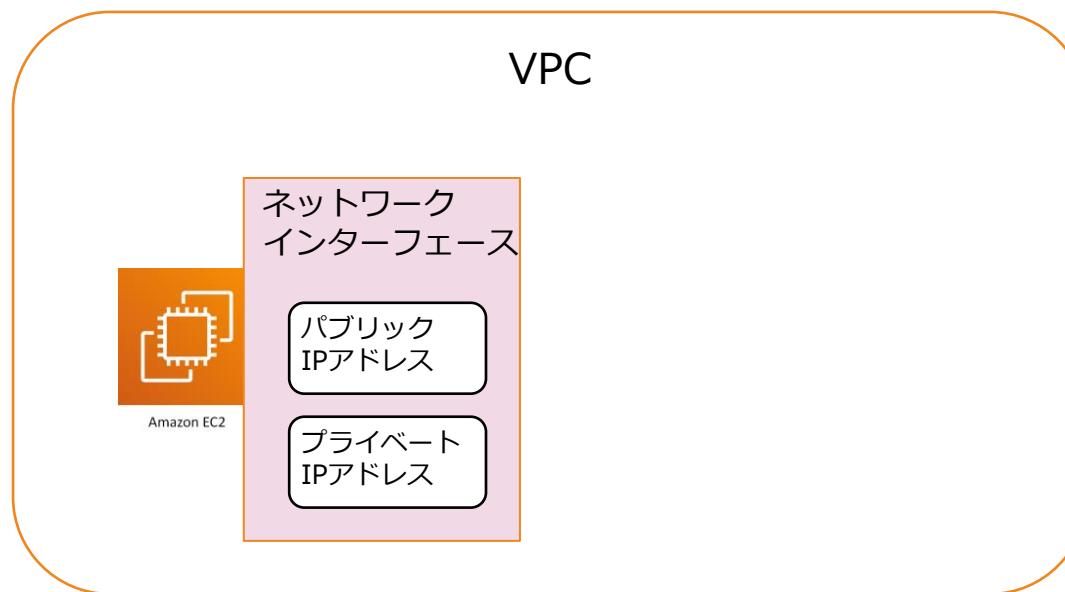
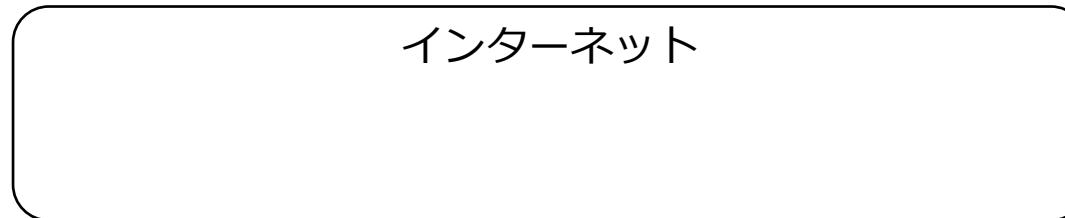


Amazon EC2



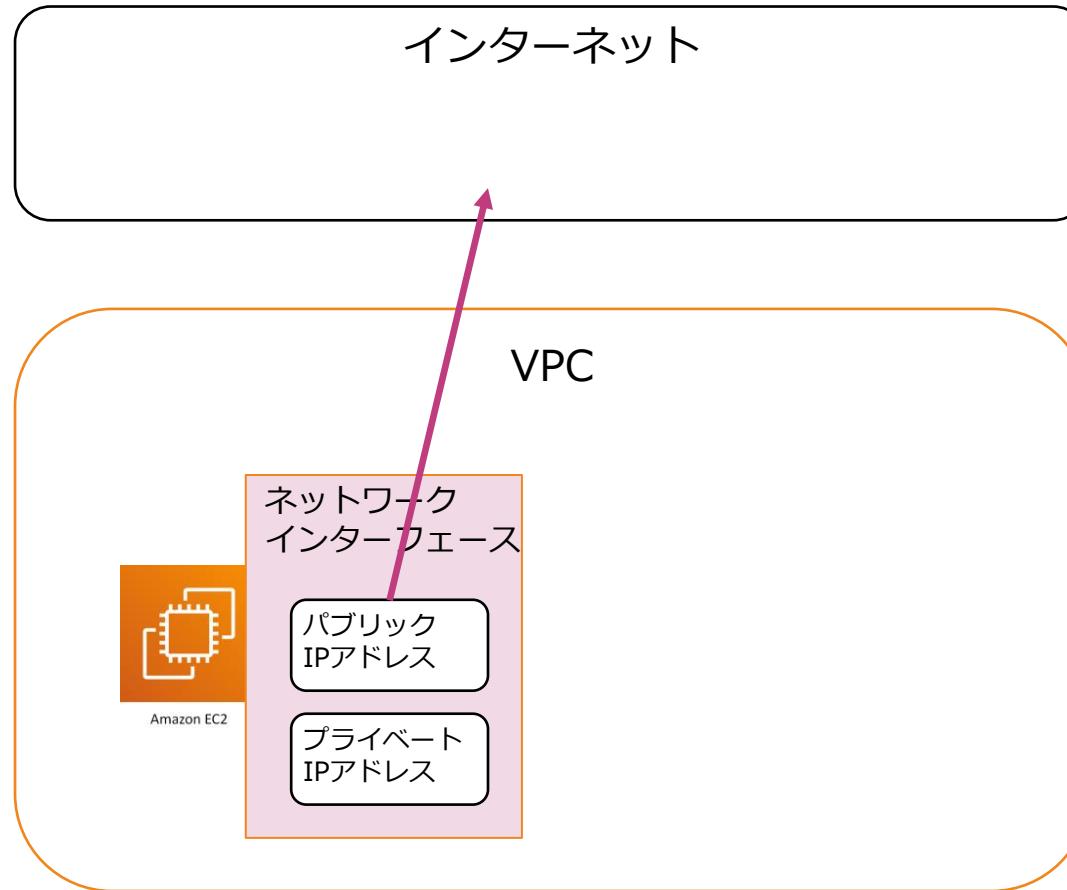
# パブリックIPとプライベートIP

インターネットと通信できるのがパブリックIPアドレス。内部ネットワーク内だけで通信できるのがプライベートIPアドレス。



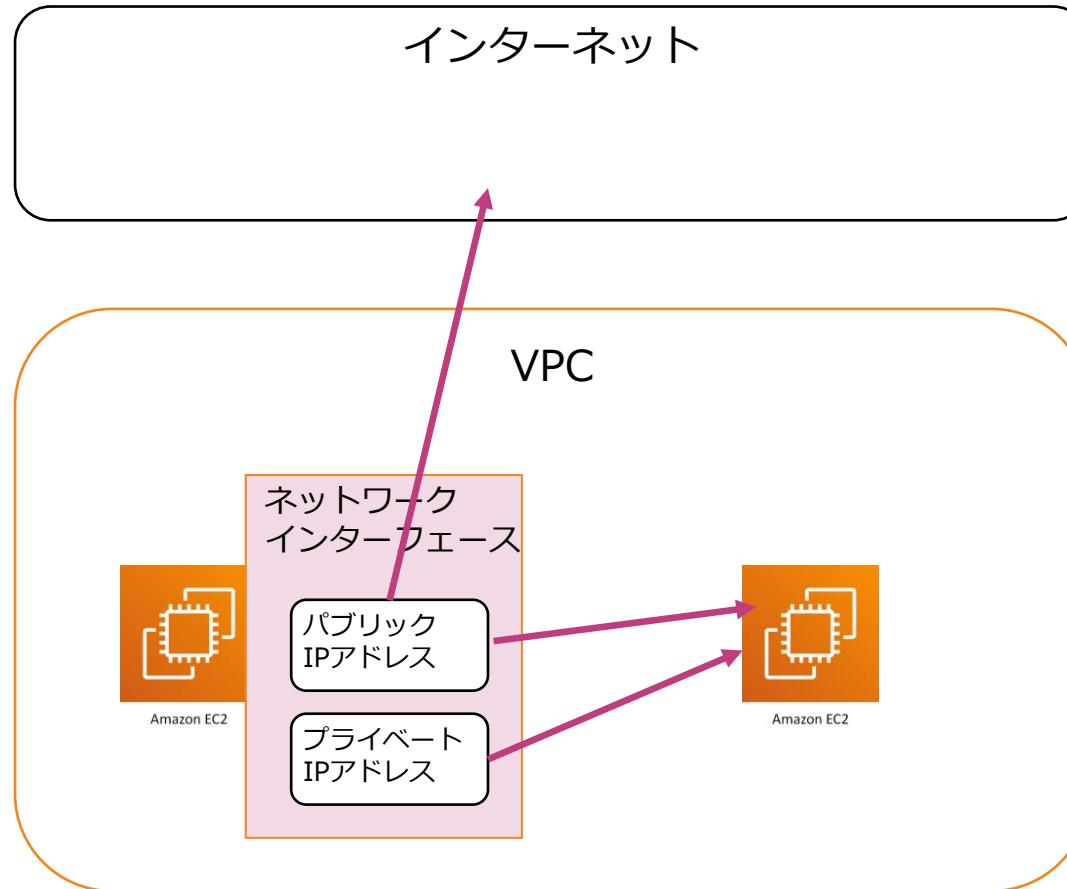
# パブリックIPとプライベートIP

インターネットと通信できるのがパブリックIPアドレス。内部ネットワーク内だけで通信できるのがプライベートIPアドレス。



# パブリックIPとプライベートIP

インターネットと通信できるのがパブリックIPアドレス。内部ネットワーク内だけで通信できるのがプライベートIPアドレス。



# インターネットアクセス

インターネットからのEC2インスタンスへのアクセスが不能な場合は以下のような原因が考えられる。

パブリック  
IPアドレスがない

- ✓ デフォルトサブネットでインスタンスを起動するとパブリックIPアドレスが自動で付与される。
- ✓ ユーザーが作成するサブネットをデフォルト設定で利用すると「パブリックIPの自動割り当て設定」が有効化されてない。
- ✓ 割り当てがされていないとインスタンスを作成しなおすか、EIPを利用する。

アクセス許可設定

- ✓ セキュリティグループまたはネットワークACLにより適切なアクセス許可が設定されていない。
- ✓ 利用しているオンプレミス側のネットワーク環境の問題

ネットワーク  
構成のミス

- ✓ パブリックサブネットにインスタンスを配置していない（サブネットとVPCにインターネットゲートウェイが設定されていない）

# [Q]インスタンスの購入方式

あなたの会社はデータセンターにホストされているITインフラストラクチャをAWSクラウドに移行しようとしています。会社はアプリケーションで利用するサーバーソフトウェアのライセンスを所有しており、AWSに移行されても、それらのライセンスを引き続き利用したいと考えています。あなたはソリューションアーキテクトとして最適なサーバーの移行先を検討しています。

最も費用効果の高いインスタンスの購入方式を選択してください。

- 1) ベアメタルインスタンスを利用する。
- 2) ハードウェア専有インスタンスを使用する。
- 3) オンデマンドインスタンスを使用する。
- 4) Dedicated Hostを使用する。

# インスタンスの購入方式

インスタンスの購入方式に応じて割引価格が提供されるため、用途に応じて割引価格を利用するすることが重要となる。

オンデマンドインスタンス	<ul style="list-style-type: none"><li>✓ 通常のインスタンス購入方式</li><li>✓ 長期契約なしで、コンピューティング性能に対して秒単位で支払う。そのライフサイクルを完全に制御できるため、いつ起動、停止、休止、開始、再起動、または終了するかを決定できる。</li></ul>
リザーブドインスタンス	<ul style="list-style-type: none"><li>✓ Amazon EC2 リザーブドインスタンス (RI) は、1年または3年の期間利用を予約することで、通常のオンデマンド料金に比べて大幅な割引価格 (最大 75%) が適用されるインスタンスの購入形式。</li><li>✓ 特定のアベイラビリティーゾーンまたはリージョンで使用するキャパシティーを予約できる2つのタイプがある。</li></ul>
スケジュールドリザーブドインスタンス <b>(利用停止)</b>	<ul style="list-style-type: none"><li>✓ 1年間にわたり毎日、毎週、または毎月ベースの指定された開始時間および期間で繰り返しキャパシティー予約を購入する。あらかじめキャパシティーを予約しておき、必要なときに使用できる。</li><li>✓ 繙続的には実行されないが定期的なスケジュールで実行されるワークロードに利用する。<b>2021年に利用停止となった。</b></li></ul>
スポットインスタンス	<ul style="list-style-type: none"><li>✓ オンデマンド価格より低価で利用できるAWS管理用に保持されているが未使用的 EC2 インスタンス。ユーザーは未使用的 EC2インスタンスを静止状態割引 (最大 90% 割引ほど) でリクエストできる。</li><li>✓ 実行時間に柔軟性がある場合や、中断できる処理に利用する。</li></ul>

# Savings Plans

1~3 年の期間に一定の使用量を守ることにより Amazon EC2 のコストを削減する

- リザーブドインスタンスと同様に、1 年または 3 年の期間に特定の量の処理能力 (USD/時間で測定) を使用する契約を結ぶことで適用される割引契約
- AWS コンピューティング使用料金を最大 72% 節約できる
- インスタンスタイプの属性の指定に柔軟性がある。
- 支払う金額を事前にコミットする必要がある。

## 【タイプ】

- EC2 インスタンスSavings Plan
  - Amazon EC2
- Compute Savings Plan
  - AWS Fargate、AWS Lambda に適用可能



# キャパシティの予約

キャパシティを事前に予約して購入形式に適用する。Savings Planなどとセットで利用する。

## キャパシティ予約

- ✓ インスタンスタイプが起動する期間を予約する（開始日と終了日を設定）
- ✓ 予めキャパシティを確保しておくことで実行時のキャパシティ不足エラーを抑制する。

## ゾーン リザーブド インスタンス

- ✓ 指定したアベイラビリティーゾーン(AZ)内で1年間または3年間の間のキャパシティを予約する
- ✓ AZは指定した場所のみ利用可能

## リージョン リザーブド インスタンス

- ✓ 指定したリージョン内で1年間または3年間の間のキャパシティを予約する
- ✓ AZはどこでも利用可能



# キャパシティーの予約

特定のアベイラビリティーゾーンの EC2 インスタンスに対して任意の期間キャパシティーを予約する

	キャパシティーの予約	リザーブド インスタンス	Savings Plans
期間	コミットメントは不要で、必要に応じて作成およびキャンセル可能	固定の 1 年または 3 年のコミットメントが必要	
キャパシティーの利点	特定のAZのキャパシティを予約して利用可能	特定のAZ またはリージョンで予約して利用可能	なし
請求割引	なし	有	有
インスタンスの制約	リージョンごとの オンデマンドインスタンス数に制限	AZまたはリージョンあたり 20 の制限 制限引上げ申請可	なし

# 物理対応可能なインスタンス

物理サーバーにインスタンスを起動して制御が可能なタイプのインスタンス

## ハードウェア専有インスタンス

- ✓ 専用HWのVPCで実行されるEC2インスタンス
- ✓ ホストHWのレベルで、他のAWSアカウントに属するインスタンスから物理的に分離する
- ✓ 同じAWSアカウントのインスタンスとはHWを共有する可能性がある

## Dedicated Host

- ✓ EC2インスタンス容量を完全にユーザー専用として利用できる物理サーバー
- ✓ サーバーにバインドされた既存のソフトウェアライセンスを利用可能

## Bare Metal

- ✓ アプリケーションが基盤となるサーバーのプロセッサーとメモリーに直接アクセス可能なインスタンス
- ✓ AWSの各種サービスとの連携が可能でOSが直接下層のハードウェアにアクセス可能

## [Q]リザーブドインスタンスの特徴

あなたの会社では3年間利用する予定でリザーブドインスタンスのスタンダードを前払いして購入してWEBアプリケーションを構築しました。しかしながら、このアプリケーションには不具合が多くなったため、急遽利用を取りやめることが決定されました。リザーブドインスタンスの利用期間はまだ2年以上残っています。あなたはソリューションアーキテクトとして、リザーブドインスタンスの料金の発生をできるだけ早く停止する必要があります。

この状況でどのような費用削減が実行できますか？

- 1) Amazonマーケットプレイスにおいて、リザーブドインスタンスを販売する。
- 2) リザーブドインスタンスマーケットプレイスにおいて、リザーブドインスタンスを販売する。
- 3) リザーブドインスタンスは3年契約で前払い購入しているため、料金は既に発生しており、このまま利用するしかない。
- 4) AWSに連絡してAWSサブスクリプションをキャンセルする。

# リザーブドインスタンスの特徴

利用期間を長期指定して利用する形式で、オンデマンドに比較して最大75%割安になる

	スタンダード	コンバータブル
利用期間	1年 (40%割引) 3年 (60%割引)	1年 (31%割引) 3年 (54%割引)
AZ／インスタンスサイズ／ネットワークタイプ変更可否	有	有
インスタンスファミリー／OS／テナント／支払オプションの変更可否	なし	有
リザーブドインスタンスマーケットプレイスでの販売可否	可能	今後可能となる予定
ユースケース	<ul style="list-style-type: none"><li>□ 一定した状態または使用量が予測可能なワークフロー</li><li>□ 災害対策などキャパシティ予約が可能なアプリケーション</li></ul>	

# [Q]スポットインスタンスの特徴

B社ではWEBアプリケーションをAWSサービスを利用して構築しました。このWEBアプリケーションでは最近になって負荷が高まっており、処理が滞るトラブルが発生しています。あなたはソリューションアーキテクトとして、一時的な負荷向上に対応するためAuto Scalingを設定して、スポットインスタンスを利用する構成を行いました。

スポットインスタンスの機能に関して、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) スポットリクエストが永続的である場合は、スポットインスタンスが中断された後に再びスポットインスタンスを起動する。
- 2) アクティブなスポットリクエストをキャンセルすると、関連するインスタンスも終了する。
- 3) スポットリクエストが永続的である場合は、スポットインスタンスを停止した後に再びスポットインスタンスを起動する。
- 4) スポットブロックはスポットインスタンスと同じように、中断される可能性がある。
- 5) アクティブなスポットリクエストをキャンセルしても、関連付けられたインスタンスは終了しない。

# [新Q]スポットインスタンスの特徴

ある会社は、長期間利用する予定のAmazon EC2インスタンスにホストされたWEBアプリケーションを構築しています。各サーバーは、開発、テストおよび本番環境に分割して実行されます。EC2インスタンスは通常時にCPU使用率が10%となり、ピーク時間中は平均50%ほどになります。本番EC2インスタンスは1日24時間実行されますが、開発およびテスト用のEC2インスタンスは毎日8時間ほど実行されています。

同社は、開発時に使用していない開発用インスタンスは適時停止してコストを節約します。またインスタンス数は抑制して、Auto Scalingを構成してインスタンス数を調整することにします。また、それぞれの環境におけるインスタンスタイプを最適化して、コストを削減したいと考えています。

同社の要件を満たすためのインスタンスタイプの購入方法を選択してください。

- 1) 本番用のEC2インスタンスにスポットインスタンスを使用する。開発およびテスト用のEC2インスタンスにリザーブドインスタンスを使用する。
- 2) 本番用のEC2インスタンスにリザーブドインスタンスを使用する。開発およびテスト用のEC2インスタンスにオンデマンドインスタンスを使用する。
- 3) 本番用のEC2インスタンスにスポットブロックを使用する。開発およびテスト用のEC2インスタンスにリザーブドインスタンスを使用する。
- 4) 本番用のEC2インスタンスにオンデマンドブロックを使用する。開発およびテスト用のEC2インスタンスにスポットブロックを使用する。

# スポットインスタンスの特徴

予備のコンピューティング容量を、オンデマンドインスタンスに比べて割引（最大90%引き）で利用できるEC2インスタンス

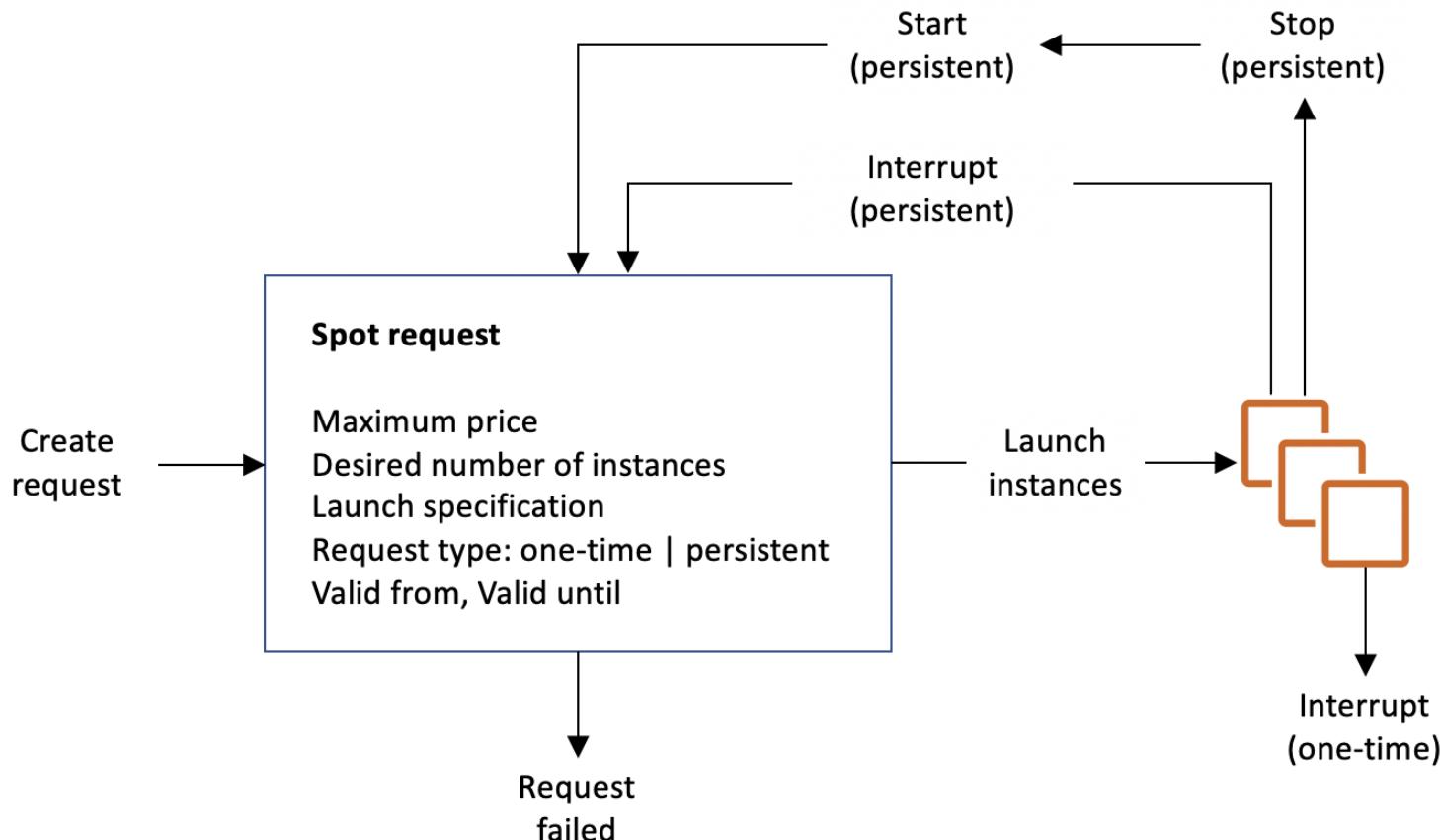
- 予備容量を利用するためとても安いが、予備用のため途中で削除される可能性がある一時利用用のインスタンス
- 事前に上限価格とインスタンスタイプを設定してリクエストすると、その価格以内のインスタンスを起動する。
- 1時間ごとの価格が需要と供給で変動して、最大90%割引となる。価格が上回った場合は停止または終了を選択できる。  
⇒一時的な拡張などの用途で利用

## 【リクエストの中斷などの挙動】

- リクエストを削除しないと、インスタンスは停止しても、再度起動をし続ける。

# スポットインスタンスの特徴

予備のコンピューティング容量を、オンデマンドインスタンスに比べて割引（最大90%引き）で利用できるEC2インスタンス



# [Q]スポットフリートの利用

あなたの会社には毎週実行されるバッチ処理のワークロードがあり、約2時間実行されます。このワークロードの処理はコスト効率を高めるために、インスタンスタイプや入札価格や価格上限などの容量ターゲットを指定することで、自動で最安値のインスタンスを選択して起動することが必要です。

この要件を満たすことができる最もコストが最適なソリューションはどれですか？

- 1) スポットインスタンスでワークロードを実行する
- 2) リザーブドインスタンスでワークロードを実行する
- 3) スケジュールドリザーブドインスタンスでワークロードを実行する
- 4) スポットフリートでワークロードを実行する

# スポットフリートの利用

スポットインスタンスのタイプと価格をフリート指定することで、自動でスポットインスタンスのリクエストを最適化する。

- スpotトインスタンスのフリートでオプションでオンデマンドも設定可能
- 上限価格内で目標容量のインスタンスを起動できるよう調整する。
- 配分戦略を選択して設定する。
  - lowestPrice  
スポットインスタンスは、最低価格のプールから取得。デフォルト戦略
  - Diversified  
スポットインスタンスは全プールに分散
  - capacityOptimized  
起動中のインスタンスの数に最適な容量を持つプールから選択

## 【スポットフリートの設定例】

- ✓ インスタンス数： 10台
- ✓ 入札価格： 1ドル
- ✓ インスタンスタイプ：  
c4.16xlarge, c3.8xlarge



c4.16xlargeとc3.8xlargeのインスタンスタイプから10インスタンスを自動で入札・起動する。



# [Q]EC2フリートの利用

大手ECマース企業はECマース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グローバルから沢山のユーザーからアクセスされる予定です。パフォーマンス要件を試算したところ、中長期的に20個のインスタンスが必要であり、不定期に実施されるバッチジョブなどのバックグラウンドジョブ用に追加で5個のインスタンスを利用することが必要となります。このバッチジョブは30分から2時間ほどで完了し、実行に失敗すると再処理を実行します。

インスタンス購入オプションの最適な組み合わせを選択してください。

- 1) リザーブドインスタンス20個とスケジュールドインスタンス5個でスポットフリートを構成する。
- 2) オンデマンドインスタンス20個とスケジュールドリザーブドインスタンス5個でEC2フリートを構成する。
- 3) リザーブドインスタンス20個とスポットインスタンス5個でEC2フリートを構成する。
- 4) オンデマンドインスタンス20個とスケジュールドリザーブドインスタンス5個でスポットフリートを構成する。

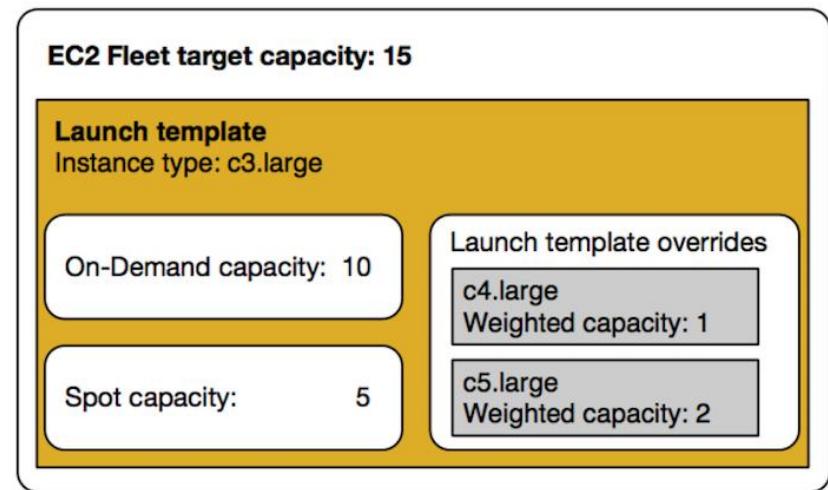
# EC2フリートの利用

オンデマンド、リザーブド、スポットインスタンスで構成されるインスタンスセットを定義する仕組み

- オンデマンドとスポットのターゲット容量を別個に定義して、1時間あたりの支払い上限料金を定義する
- アプリケーションに最適なインスタンスタイプを指定
- 各購入オプション内でフリート容量をAmazon EC2で分散する方法を指定する

## 【フリートの動作】

- リクエストで指定したターゲットキャパシティーを満たすために必要なインスタンス数の起動
- 1時間あたりの上限の合計料金を指定すると、支払いの上限料金に達するまで、容量を起動
- 支払い上限料金に達すると、ターゲットキャパシティーに満たない場合でも、フリートはインスタンスの起動を停止
- リザーブドインスタンス(RI)がありオンデマンドインスタンスを指定した場合、RIを使用



## 【設定内容】

- インスタンスタイプ
- オンデマンドとスポットの組合数
- 利用料金の上限



# [Q]プレイスメントグループの利用

大学ではゲノムデータの分析をAWS上で実行することになりました。ゲノム解析には高性能なサーバー処理が求められており、パフォーマンスコンピューティングに対応した複数のEC2インスタンスを利用した高パフォーマンスなネットワーク処理も不可欠となっています。

このアプリケーションを実行する際に利用するべきEC2インスタンスの構成はどれでしょうか？

- 1) EC2インスタンスでパーティションプレイスメントグループを構成する。
- 2) EC2インスタンスでEC2フリートを構成する。
- 3) EC2インスタンスでスポットフリートを構成する。
- 4) EC2インスタンスでスプレッドプレイスメントグループを構成する。
- 5) EC2インスタンスでクラスタープレイスメントグループを構成する。

# プレイスメントグループの利用

単一のアベイラビリティーゾーン内のインスタンスのパフォーマンスを向上させるために論理的にグループ化する機能

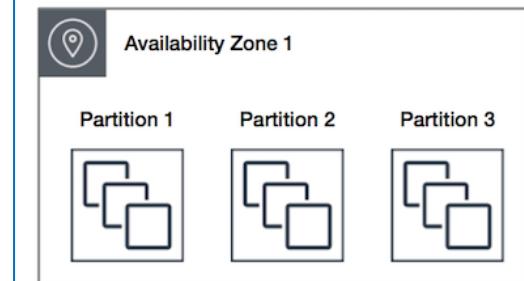
## クラスター プレイスメント グループ

- ✓ 単一AZ内のインスタンスを論理的にグループ化した構成
- ✓ 同じリージョン内の複数のピア VPC にまたがることも可能
- ✓ グループ内のインスタンスは、TCP/IP トラフィックのフローあたりのスループット上限が高くなり、ネットワークの二分帯域幅の広い同じセグメントに配置されインスタンス間通信が向上する
- ✓ 低いネットワークレイテンシー、高いネットワークスループットを実現するアプリケーション向けの構成



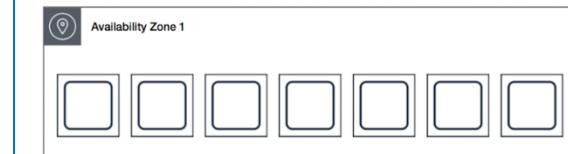
## パーティション プレイスメント グループ

- ✓ Amazon EC2 は各グループをパーティションと呼ばれる論理的なセグメントに分割した構成
- ✓ プレイスマントグループ内の各パーティションにそれぞれ一連のラックがあり、プレイスメントグループ内のパーティションどうしが同じラックを共有しない。
- ✓ ラックを分離することで、アプリケーション内でのハードウェア障害による影響を隔離して、軽減する。



## スプレッド プレイスメント グループ

- ✓ それぞれに独自のネットワークおよび電源がある異なるラックに別々に配置できるインスタンスのグループ
- ✓ 1 つのAZ内の、スプレッドプレイスメントグループに配置された 7 つのインスタンスは、7 つの異なるラックに配置される。
- ✓ 少数の重要なインスタンスが互いに分離して保持できる。インスタンスが同じラックを共有するときに発生する可能性のある同時障害のリスクを軽減する。



# [Q]拡張ネットワーキング

大学ではゲノムデータの分析をAWS上で実行することになりました。ゲノム解析には高性能なサーバー処理が求められており、パフォーマンスコンピューティングに対応した複数のEC2インスタンスを利用した高パフォーマンスなネットワーク処理も不可欠となっています。あなたはソリューションアーキテクトとして、EC2インスタンスの最適な構成により高ネットワークスループットを確保する必要があります。

この要件を達成するのに必要な構成はどれでしょうか？（3つ選択してください）

- 1) EC2インスタンスの拡張ネットワーキングを使用する。
- 2) EBSのプロビジョンドIOPSボリュームを使用する。
- 3) EC2インスタンスのコンピューティング最適化インスタンスを利用する。
- 4) クラスター・プレイスメント・グループを使用する。
- 5) Dedicated Hostを使用する。

# 拡張ネットワーキング

高い帯域幅、1秒あたりのパケット(PPS)の高いパフォーマンス、常に低いインスタンス間レイテンシーを実現する。3つのタイプを利用する。

アダプター	インスタンスタイプの例	カーネルモジュール	Windows ドライバ	パフォーマンス
VIF	すべて	xen-netfront	Citrix または AWS PV	低～中
Intel 82599 VF	C3、C4、D2、I2、R3、M4 (m4.16xlarge は除く)	ixgbevf	Intel 82599 VF	最大 10 Gbps
Elastic Network Adapter	C5、C5d、F1、G3、H1、I3、m4.16xlarge、M5、M5a、M5d、P2、P3、R4、R5、R5a、R5d、T3、u-6tb1.metal、u-9tb1.metal、u-12tb1.metal、X1、X1e、および z1d	ena	ena	最大 25 Gbps

Reference: <https://aws.amazon.com/jp/premiumsupport/knowledge-center/enable-configure-enhanced-networking/>

# [Q] Elastic Fabric Adapterの利用

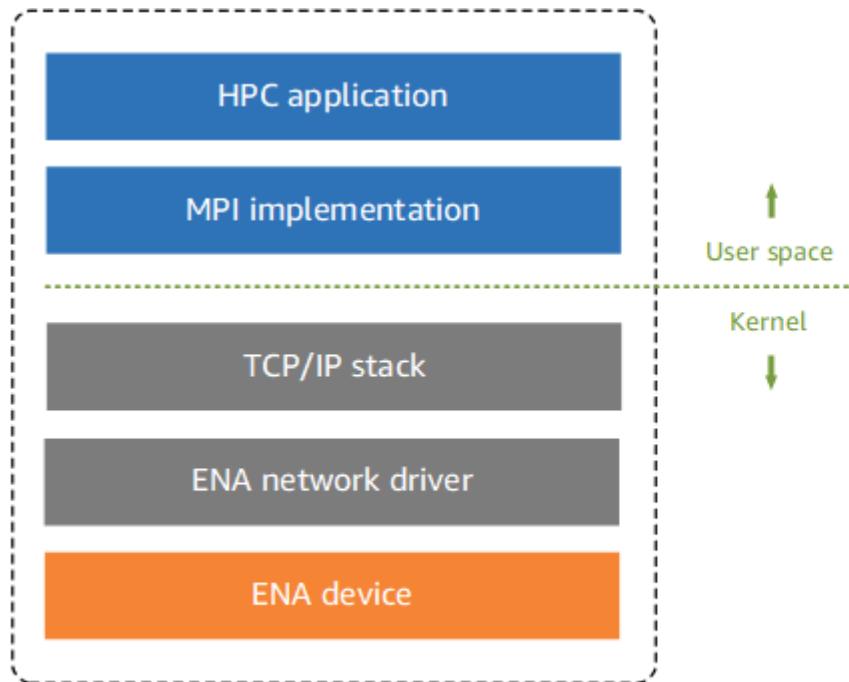
大学ではゲノムデータの分析をオンプレミス環境で実行しています。ゲノム解析には高性能なサーバー処理が求められており、パフォーマンスコンピューティング(HPC)を利用しています。あなたはソリューションアーキテクトとして、これらのワークフローをオンプレミスインフラストラクチャからAWSクラウドに移行することを検討しています。

HPCワークフローを実行するEC2インスタンスで利用されるネットワークコンポーネントはどれでしょうか？

- 1) Elastic Network Interface
- 2) Elastic Fabric Adapter
- 3) Elastic Network Adapter
- 4) Elastic IP Address

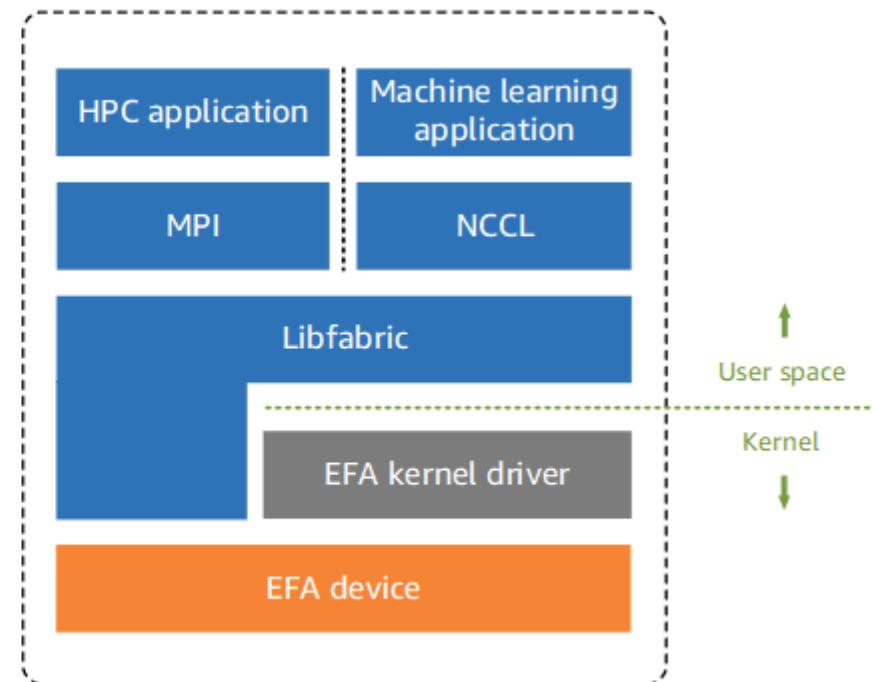
# Elastic Fabric Adapterの利用

ハイパフォーマンスコンピューティング (HPC) と機械学習アプリケーションを高速化するためのEC2用ネットワークデバイス



Traditional HPC software stack in EC2

ENAは、VPC のサポートに必要な従来の IP ネットワーキング機能を提供



HPC software stack in EC2 with EFA

EFAはENAsの機能に加えてOSバイパス機能がある。  
Libfabric APIを利用してHPCと機械学習アプリケーションはオペレーティングシステムのカーネルをバイパスしてEFAデバイスと直接通信できる

# [Q]EC2の自動リカバリー

あなたの会社は30台以上のEC2インスタンスを利用して大規模な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なるべく自動的に運用をする必要があります。あなたはソリューションアーキテクトとして、Amazon CloudWatchアラームを使用して、EC2インスタンスが障害になった場合に自動的に回復されます。

この自動回復されたインスタンスのステータスとして正しい説明はどれでしょうか？

- 1) インスタンスに設定されたパブリックIPv4アドレスは、インスタンスが復元時には別のアドレスに変更される。
- 2) インスタンスに設定されたパブリックIPv4アドレスは、リカバリ後も維持される。
- 3) 復元されたインスタンスは、インスタンスID、プライベートIPアドレス、Elastic IPアドレス、全てのメタデータは保持される。
- 4) インスタンス復元前のメモリ内にあるデータはすべて保持される。

# EC2のリカバリー

EC2インスタンスは定期的にバックアップすることが重要

- 定期的にバックアップ（AMI／スナップショット）をとる
- 定期的にリカバリプロセスを確認する
- 複数のAZに重要なアプリケーションをデプロイする
- CloudWatchによりインスタンスのステータスをモニタリングする
  - チェック結果が失敗になった場合、CloudWatch アラームアクションを使用してインスタンスを自動的に復旧させる
  - 自動復旧後のステータスとIPアドレスは元のインスタンスと同じ
- インスタンス起動時に動的IPアドレス処理の設定を行う

# [Q]インスタンスの停止と起動

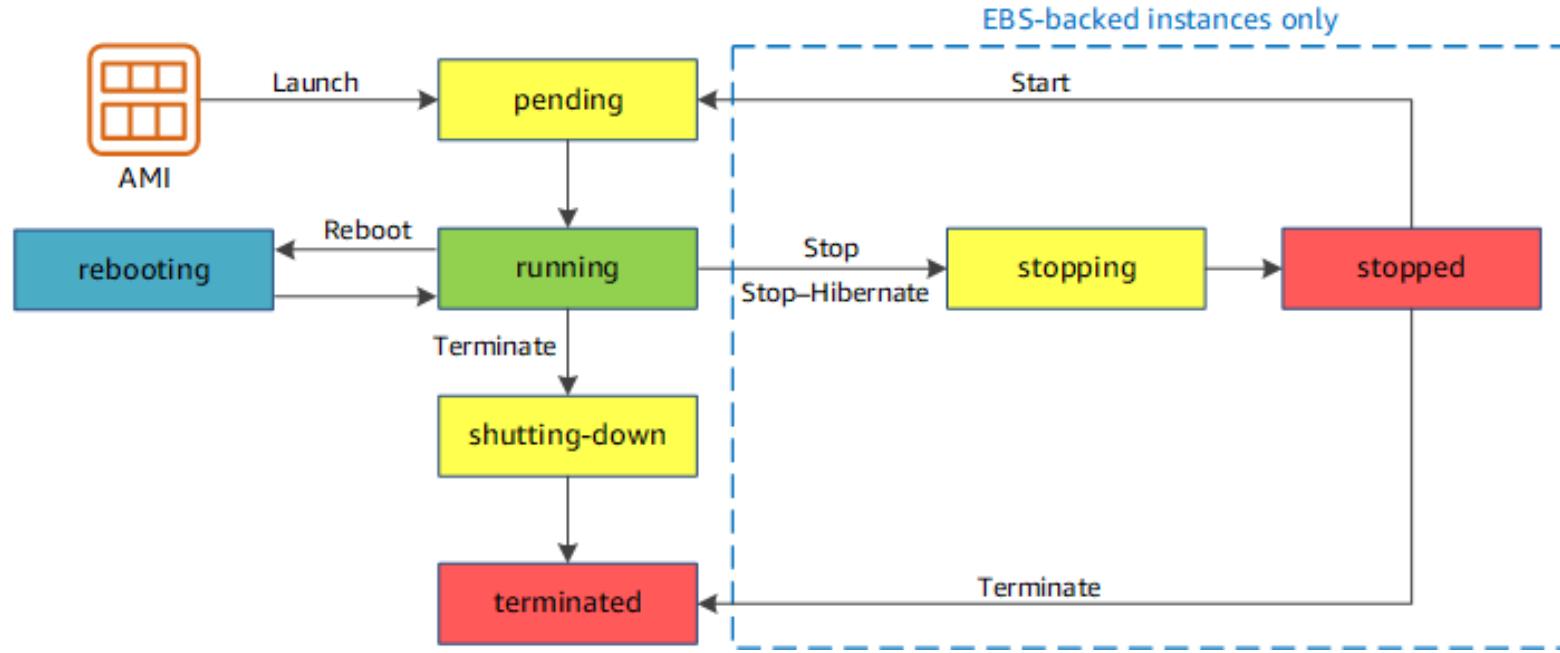
あなたはソリューションアーキテクトとしてEC2インスタンスのメンテナンスを実施しています。停止したEC2インスタンスを再起動しようとしましたが、すぐに保留状態から終了状態に変わりました。

最も可能性の高い原因はどれでしょうか？

- 1) EBSのスナップショットが壊れている。
- 2) EBSのスナップショットが暗号化されている。
- 3) EBSのスナップショットがコピーされたものである。
- 4) EBSボリュームが不足している。

# インスタンスの再起動

インスタンスのステータスは以下のように遷移



Reference: [https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html)

- ✓ 再起動時にはデータが消失され、ホストが変更される可能性あり
- ✓ 以下のような場合は起動に失敗する。
  1. スナップショットが壊れている
  2. EBSボリューム制限を超過している
  3. 暗号化されたスナップショットのキーを有していない。
  4. インスタンスストア型のAMIが必要なパートが失っている。

# インスタンスの再起動

インスタンスのステータスは以下のような内容を示している。

pending	インスタンスは running 状態への移行準備中です。 初めて起動する場合、または pending 状態になってから起動する場合、インスタンスは stopped 状態になります。	課金されない
running	インスタンスは実行中で、使用できる状態です。	課金される
stopping	インスタンスは停止または停止休止の準備中です。	停止準備中は無課金 休止準備中は課金
stopped	インスタンスは停止されているため、使用できません。 インスタンスはいつでも起動できます。	課金されない
shutting-down	インスタンスは削除準備中です。	課金されない
terminated	インスタンスは完全に削除されているため、起動することはできません。	課金されない

# [Q]ハイバネーションの利用

あなたのEC2インスタンスを起動しました。このEC2インスタンスはメンテナンス時に一時的に停止させる必要がありますが、その際にメモリ内のデータなどを維持することが求められています。

この要件を満たすためにEC2インスタンスで設定するべき機能はどれでしょうか？

- 1) AMIを使用する。
- 2) EC2インスタンスのリブート設定を利用する。
- 3) EC2インスタンスの再起動を実施する。
- 4) ハイバネーションを使用する。

# ハイバネーションの利用

ハイバネーションにより、再起動時に停止前の状態を維持することが可能

## ハイバネーションの機能

シャットダウン前にメインメモリ（RAM）の内容をハードディスク等に退避することで、次回起動時にまたメインメモリに読み込んで、シャットダウン前と同じ状態で起動する。再起動時に停止前の状態を保持して、再起動を高速化する。

## インスタンスタイプ に応じて設定

インスタンスタイプに応じてハイバネーションの実施可否が決まる。  
初期ではAmazon Linux 1を実行しているM3、M4、M5、C3、C4、C5、R3、R4、R5のみで可能であったが、現在はAmazon Linux 2やWindowsなども対応

# ハイバネーションの利用

インスタンスの詳細設定時に有効化する必要がある。

ステップ 3: インスタンスの詳細の設定

ドメイン結合ディレクトリ	<input type="text" value="ディレクトリなし"/>	<a href="#">新しいディレクトリの作成</a>
IAM ロール	<input type="text" value="なし"/>	<a href="#">新しい IAM ロールの作成</a>
CPU オプション	<input type="checkbox"/> CPU オプションを指定	
シャットダウン動作	<input type="text" value="停止"/>	
停止 - 休止動作	<input checked="" type="checkbox"/> 停止動作に休止動作を追加する	
休止動作を有効にする際、インスタンスマемリ (RAM) を格納するための容量がルートボリュームが RAM の内容を保存し、予想される使用量 (OS、アプリケーションなど) に対応ください。休止状態を使用するには、ルートボリュームが暗号化された EBS ボリューム		
<a href="#">タグ付けのための URL</a>		

## VPCの出題範囲

# VPCとは何か？

VPCはAWSクラウドのネットワークからユーザー専用の領域を切り出すことができる仮想ネットワークのサービス

AWSクラウドのネットワーク空間

# VPCとは何か？

VPCはAWSクラウドのネットワークからユーザー専用の領域を切り出すことができる仮想ネットワークのサービス

AWSクラウドのネットワーク空間



# VPCの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

<b>VPCの設定 (デフォルトVPC)</b>	✓ デフォルトVPCの構成状況を問う質問が出題される。
<b>サブネットマスク の設定</b>	✓ サブネットマスクを利用したCIDRの設定内容に関する質問が出題される。
<b>ゲートウェイの設定</b>	✓ VPCとサブネットに設置する各種ゲートウェイの使い分けに関する質問が問われる。
<b>インターネット ゲートウェイ</b>	✓ インターネットゲートウェイの設定方法や活用に関する質問が問われる。

# VPCの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

NATゲートウェイ	<ul style="list-style-type: none"><li>✓ NATゲートウェイの設定方法に関する質問が問われる。</li><li>✓ NATインスタンスとNATゲートウェイとの違いや特徴について問われる。</li></ul>
VPCエンドポイント	<ul style="list-style-type: none"><li>✓ VPCエンドポイントを利用したAWSサービスとの連携方法が問われる。</li></ul>
VPCピアリング	<ul style="list-style-type: none"><li>✓ VPCとVPCとを接続するVPCピアリングの活用や設定方法が問われる。</li></ul>
ネットワークACL	<ul style="list-style-type: none"><li>✓ ネットワークACLとセキュリティゲートウェイの違いなど、その特徴が問われる。</li><li>✓ ネットワークACLの設定内容を確認する質問が出題される。</li></ul>
VPC内サービスへの接続	<ul style="list-style-type: none"><li>✓ VPCに設置したAWSサービスへとアクセスする接続方式が問われる。</li><li>✓ また、接続方式の設定方法が問われる。</li></ul>

# VPCの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

<b>サブネットによる構成</b>	<ul style="list-style-type: none"><li>✓ サブネットの構成方法が問われる。</li><li>✓ パブリックサブネットとプライベートサブネットを利用した最適なAWSリソースの配置構成が問われる。</li></ul>
<b>踏み台サーバー</b>	<ul style="list-style-type: none"><li>✓ 踏み台サーバーを設置した、プライベートサブネット内のリソースへのアクセス構成が問われる。</li></ul>
<b>VPCフローログ</b>	<ul style="list-style-type: none"><li>✓ VPCフローログの役割に関する質問が出題される。</li></ul>
<b>VPCにおけるDNSの使用</b>	<ul style="list-style-type: none"><li>✓ VPCにおいてDNSの名前解決が適用されるための設定が問われる。</li></ul>
<b>Elastic IP</b>	<ul style="list-style-type: none"><li>✓ Elastic IPの役割や課金方式が問われる。</li></ul>

# VPCの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

IPフローティング	✓ EC2インスタンスを切り替える際にダウンタイムを抑える仕組みとしてIPフローティングの利用方法が問われる。
ENI	✓ ENIの役割とアタッチ方式について問われる。

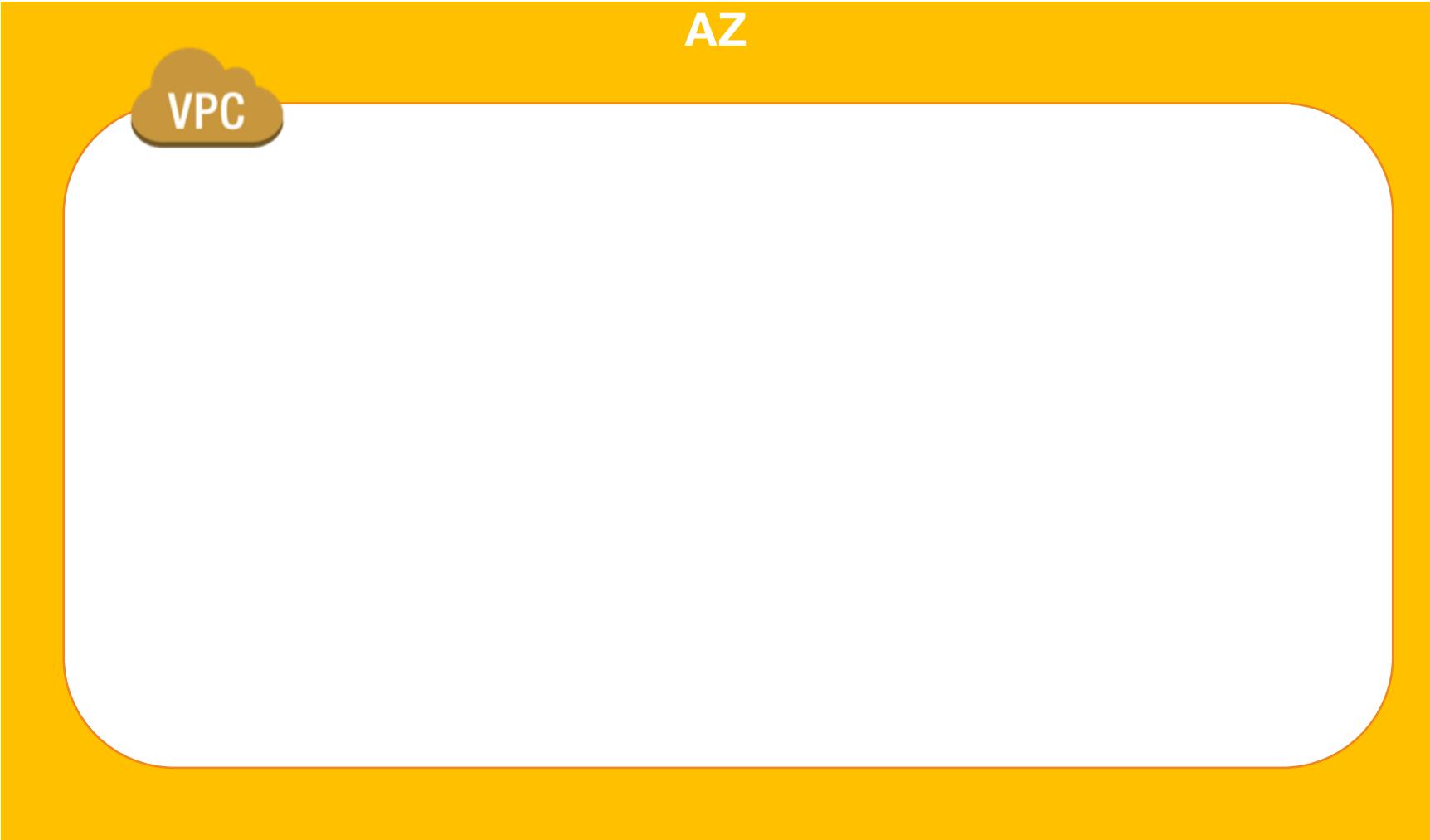
# Virtual Private Cloud (VPC)

VPCはAWSクラウド内に論理的に分離されたセクションを作り、ユーザーが定義した仮想ネットワークを構築するサービス

- ✓ 任意の IP アドレス範囲を選択して仮想ネットワークを構築する
- ✓ サブネットの作成、ルートテーブルやネットワークゲートウェイの設定などにより、仮想ネットワーキング環境を完全に制御できる。
- ✓ 必要に応じてクラウド内外のネットワーク同士を接続することも可能
- ✓ 複数の接続オプションが利用可能
  - インターネット経由
  - VPN/専用線(Direct Connect)

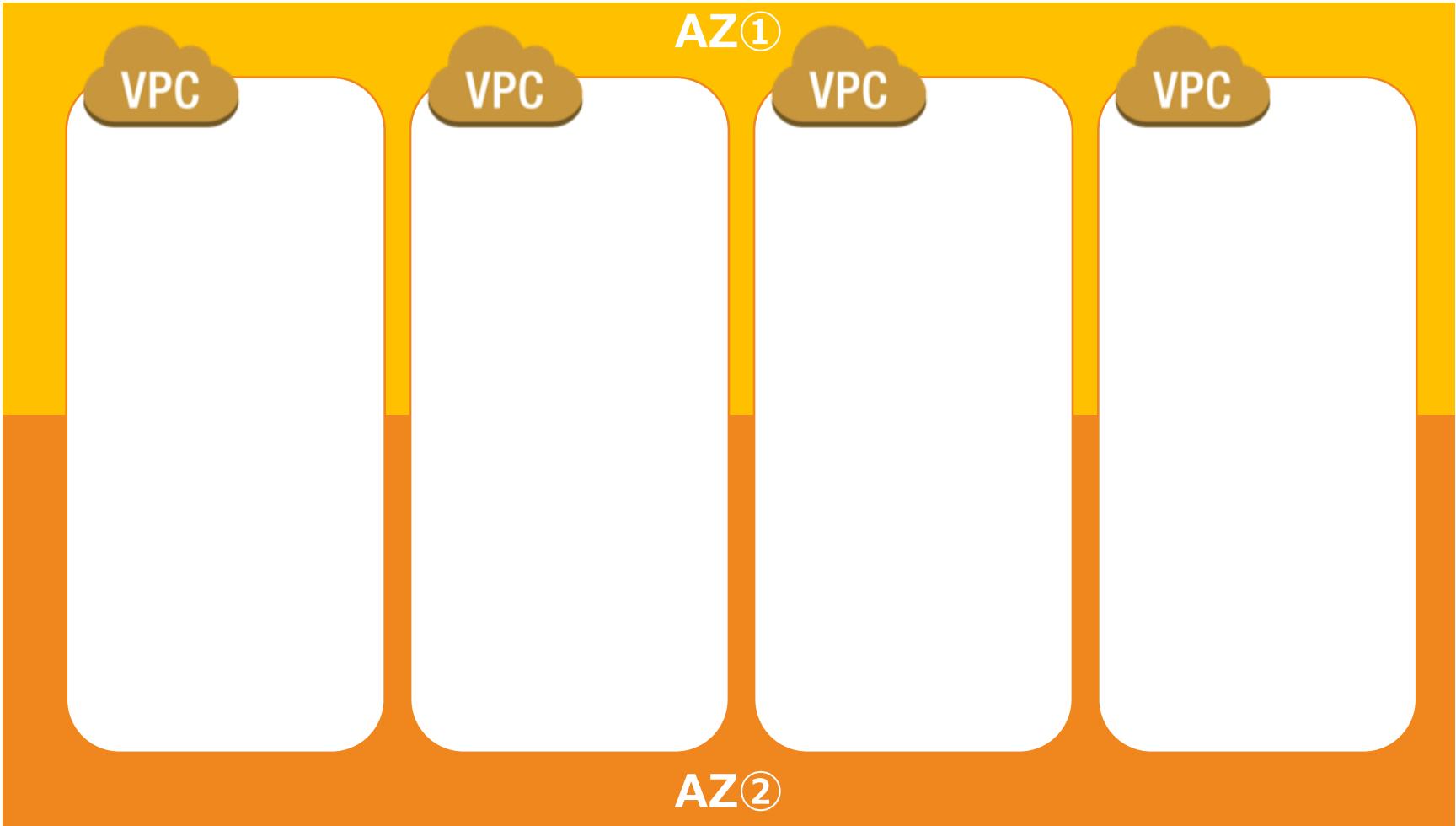
# Virtual Private Cloud (VPC)

単一のVPCを構築すると単一AZの範囲に設定される。



# Virtual Private Cloud (VPC)

同一リージョン内ではVPCは複数のAZにリソースを含めることができます



# サブネットとVPC

VPCとサブネットの組合せでネットワーク空間を構築する  
VPCはサブネットとのセットが必須



## [Q] VPCの設定（デフォルトVPC）

あなたはAWSアカウントを新規に開設して、まずはEC2インスタンスを起動させました。VPCの構成をしていなかったため、このEC2インスタンスにはデフォルトVPCが設定されています。インスタンスにプライベートDNSホスト名とパブリックDNSホスト名の両方があることを確認する必要があります。

VPCを利用した場合にDNSホスト名はどのように割り当てられますか？（2つ選択してください）

- 1) デフォルト以外のVPCでは、初期設定ではプライベートDNSホスト名が割り当てられるが、パブリックDNSホスト名は割り当てられない。
- 2) デフォルト以外のVPCでは、パブリックDNSホスト名とプライベートDNSホスト名が割り当てられる。
- 3) デフォルト以外のVPCでは、必ずプライベートDNSホスト名が割り当てられるが、パブリックDNSホスト名は割り当てられない。
- 4) デフォルトVPCではパブリックDNSホスト名とプライベートDNSホスト名が割り当てられる。
- 5) デフォルトVPCではパブリックDNSホスト名とプライベートDNSホスト名が割り当てられない。

# VPCの設定（デフォルトVPC）

AWSアカウントを作成すると、自動的に各リージョンに1つずつデフォルトVPCとデフォルトサブネットが生成される

- ✓ サイズ /16 の IPv4 CIDR ブロック (172.31.0.0/16) の VPC を作成する。これは、最大 65,536 個のプライベート IPv4 アドレスを提供する。
- ✓ 各アベイラビリティーゾーンに、サイズ /20 のデフォルトサブネットを作成する。この場合は、サブネットあたり最大 4,096 個のアドレスが作成され、その中のいくつかは Amazon が使用するように予約されている。
- ✓ インターネットゲートウェイを作成して、デフォルトVPCに接続する。
- ✓ デフォルトのセキュリティグループを作成し、デフォルトVPCに関連付ける。
- ✓ デフォルトのネットワークアクセスコントロールリスト (ACL) を作成し、デフォルトVPCに関連付ける。
- ✓ デフォルトVPC を備えた AWS アカウントにはデフォルトDHCPオプションセットを関連付ける
- ✓ パブリックとプライベートのDNSホスト名が付与される。

# VPCの設定：通常の設定

VPCウィザードを利用しない場合は、VPCを作成、サブネットを作成と1つずつ作成する。

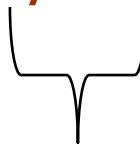


# CIDR (Classless Inter-Domain Routing)

サブネットマスクの値を設定し、同じネットワーク範囲として扱うIPアドレスの個数を調整できるIPアドレスの設定方法

## 【表記方法】

196.51.XXX.XXX/16



サブネット

左から16桁目までのIPアドレスを固定

# [Q]サブネットマスクの設定

あなたは新しくVPCを設定して、パブリックサブネットを2つ、プライベートサブネットを2つ設定してITインフラを設置しようと考えています。単一のVPC内でのIPv4アドレス指定およびサブネット作成に対して、CIDRを設定する必要があります。CIDRの設定として200個のIPアドレスを利用できるようにする必要があります。

CIDRのサブネットマスク指定として、多すぎず最適なIPアドレス数となる設定を選択してください。

- 1) /21
- 2) /22
- 3) /23
- 4) /24

# CIDR

VPCは/16 ~ /28のCIDR範囲を使用できる

/16 ~ /28

# CIDR

CIDRに/16を設定した際に設定可能となるサブネット数とIPアドレス数の組合せ（AWS管理IPの5つを引いたもの）

サブネットマスク	サブネット数	サブネット当たりのIPアドレス数 (AWSで利用可能な)
/18	4	16379
/20	16	4091
/22	64	1019
/24	256	251
/26	1024	59
/28	4096	11

# CIDR

既にAWS側で利用されており、設定できないアドレスもある  
(/24の例)

ホストアドレス	用途
.0	ネットワークアドレス
.1	VPCルータ
.2	Amazonが提供するDNSサービス
.3	AWSで予約されているアドレス
.255	ブロードキャストアドレス

# [Q]サブネットの作成

あなたはAWSアカウントを新規に開設して、まずはVPCを構成することにしました。VPCウィザードを使用することでよく利用される構成を迅速に設定することが可能です。パブリックなアクセスが必要なWebサーバー、セキュリティを高めるためにプライベートなアクセスに限定したデータベースサーバーを設置するためのネットワーク構成が必要です。

Amazon VPCのサブネットに関し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) 各サブネットは単一のアベイラビリティーゾーンに設定される。
- 2) 各サブネットは複数のアベイラビリティーゾーンに設定可能である。
- 3) 各サブネットは、VPCのメインルートテーブルに自動的に関連付けられる。
- 4) 各サブネットは、サブネットのメインルートテーブルが設定され、そのルートテーブルがVPCに自動的に関連付けられる。
- 5) 各サブネットはインターネットゲートウェイがデフォルトで構成される。

# サブネット

サブネットはCIDR範囲で分割したネットワークセグメント

パブリックサブネット  
10.0.1.0/24



トラフィックがインターネットゲートウェイにルーティングされるサブネット

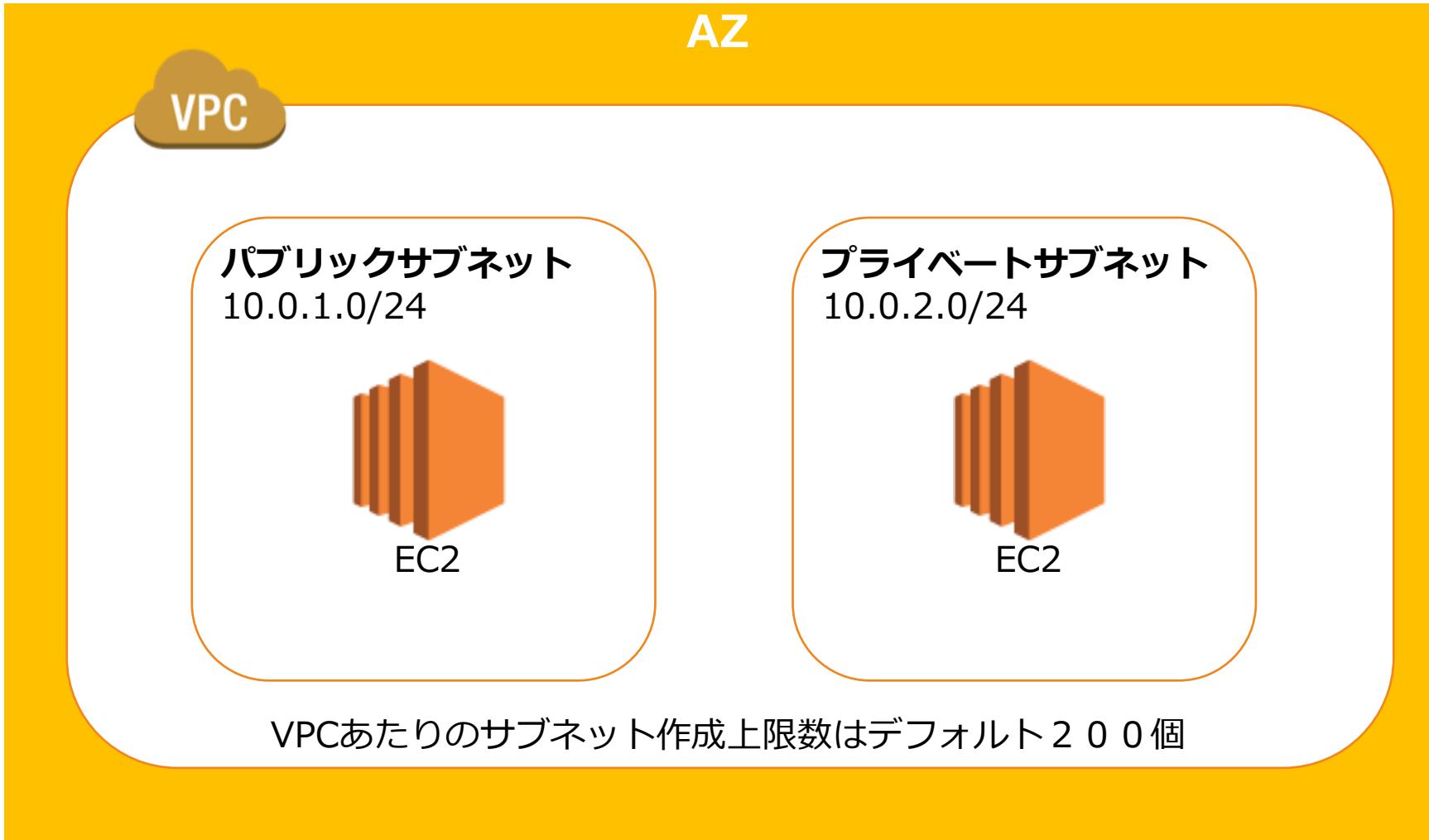
プライベートサブネット  
10.0.2.0/24



インターネットゲートウェイへのルートがないサブネット

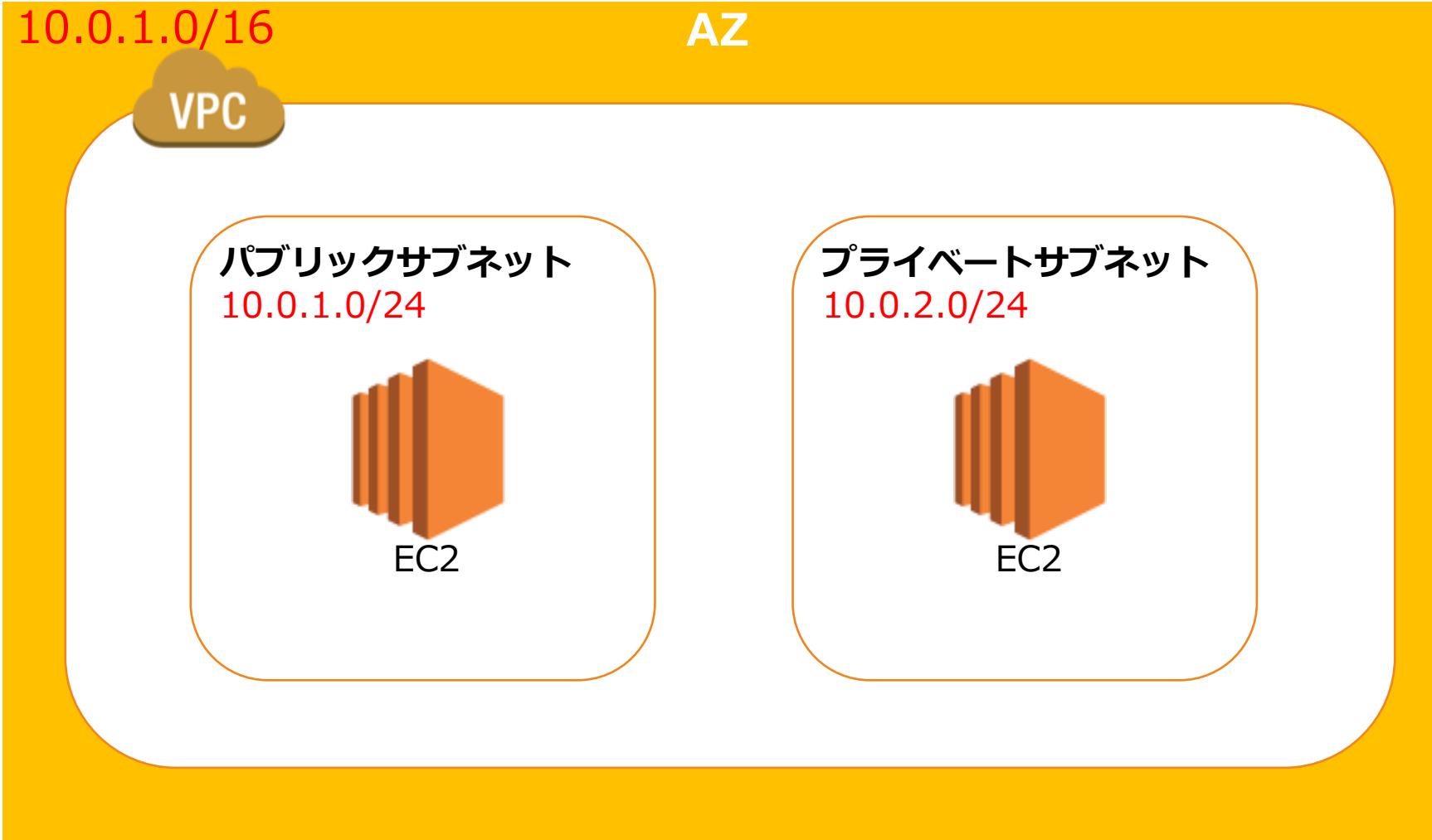
# サブネット

サブネットはVPC内の複数設置でき、1つのAZを指定して配置される。パブリックとプライベートがある。



# CIDRの付与

VPCとサブネットにはCIDR（IPアドレス範囲）が付与され、ネットワークレンジが決まる。



# サブネット

インターネットゲートウェイへのルーティング有無でサブネットのタイプが分かれる

パブリックサブネット  
10.0.1.0/24



トラフィックがインターネットゲートウェイにルーティングされるサブネット

プライベートサブネット  
10.0.2.0/24



インターネットゲートウェイへのルートがないサブネット

# サブネット

インターネットゲートウェイへのルーティング有無でサブネットのタイプが分かれる

ルートテーブル > ルートの編集

## ルートの編集

送信先	ターゲット	ステータス	伝播済み
172.31.0.0/16	local	active	いいえ
0.0.0.0/0	igw-80b29de4	active	いいえ

[ルートの追加](#)

\* 必須

[キャンセル](#) [ルートの保存](#)

# [Q] ゲートウェイの設定

あなたは新しくVPCを設定して、パブリックサブネットを2つ、プライベートサブネットを2つ設定してインフラを設置しようと考えています。構成したプライベートサブネットにあるインスタンスはIPv6プロトコルを使用してインターネットからホストに接続する必要があります。その際に、インターネットからのアクセスは拒否しつつ、インターネット側へのトラフィックは通せるようにします。

この接続を有効にするには、どの仕組みを設定することが必要ですか？（2つ選択してください。）

- 1) パブリックサブネットにインターネットゲートウェイを作成する。パブリックサブネット内のトラフィックがインターネットゲートウェイにルーティングするようにルートテーブルを設定する。
- 2) パブリックサブネットにNATゲートウェイを作成する。プライベートサブネットから送信されたトラフィックを、このNATゲートウェイ経由でルーティングするようにルートテーブルを設定する。
- 3) プライベートサブネットにNATゲートウェイを作成する。プライベートサブネットから送信されたトラフィックを、このNATゲートウェイ経由でルーティングするようにルートテーブルを設定する。
- 4) プライベートサブネットにEgress-Onlyインターネットゲートウェイを作成する。プライベートサブネット内のトラフィックが Egress-Onlyインターネットゲートウェイにルーティングするようにルートテーブルを設定する。
- 5) パブリックサブネットにEgress-Onlyインターネットゲートウェイを作成する。パブリックサブネット内のトラフィックが Egress-Onlyインターネットゲートウェイにルーティングするようにルートテーブルを設定する。

# ゲートウェイの設定

VPCコンソールで作成・管理できるゲートウェイは以下の通り

インターネット ゲートウェイ	✓ インターネットへの出入り口となるゲートウェイで、デフォルトゲートウェイとして利用されることが多い
NATゲートウェイ	✓ プライベートサブネットのリソースからインターネットへのトラフィックを可能にするためのゲートウェイ
Egress-Only Internet Gateway	✓ IPv6向けのインターネットゲートウェイ ✓ IPv6 経由での VPC からインターネットへの送信を可能にし、インターネットからのインスタンスへの接続は防ぐ
カスタマーゲートウェイ	✓ オンプレミス環境と接続する際に利用するゲートウェイ ✓ カスタマーゲートウェイデバイスまたはソフトウェアアプリケーションに関する情報を AWS に提供する
仮想プライベート ゲートウェイ	✓ 仮想プライベートゲートウェイは、VPN トンネルの Amazon 側にあるルーター ✓ VPN接続時に利用する

## [Q]インターネットゲートウェイ

あなたは新しくVPCを設定して、パブリックサブネットを1つ、プライベートサブネットを1つ設定してインフラを設置しようと考えています。IPv4アドレスを利用してインターネットへのアクセスを許可するためには、サブネットにパブリックサブネットとして機能させる設定が必要です。

パブリックサブネットに必要な設定はどれでしょうか？

- 1) Egress-Onlyインターネットゲートウェイへのルートを設定する。
- 2) インターネットゲートウェイへのルートを設定する。
- 3) NATゲートウェイへのルートを設定する。
- 4) カスタマーゲートウェイへのルートを設定する。

# インターネットゲートウェイ

パブリックサブネットからインターネットに接続するにはインターネットゲートウェイが必要



# インターネットゲートウェイ

ルートテーブルによりインターネットゲートウェイへのルートを確立する。

- インターネットゲートウェイをVPCに設置する。
- パブリックサブネットのルートテーブルにインターネットゲートウェイへの経路を設定する。

# [Q] NATゲートウェイ

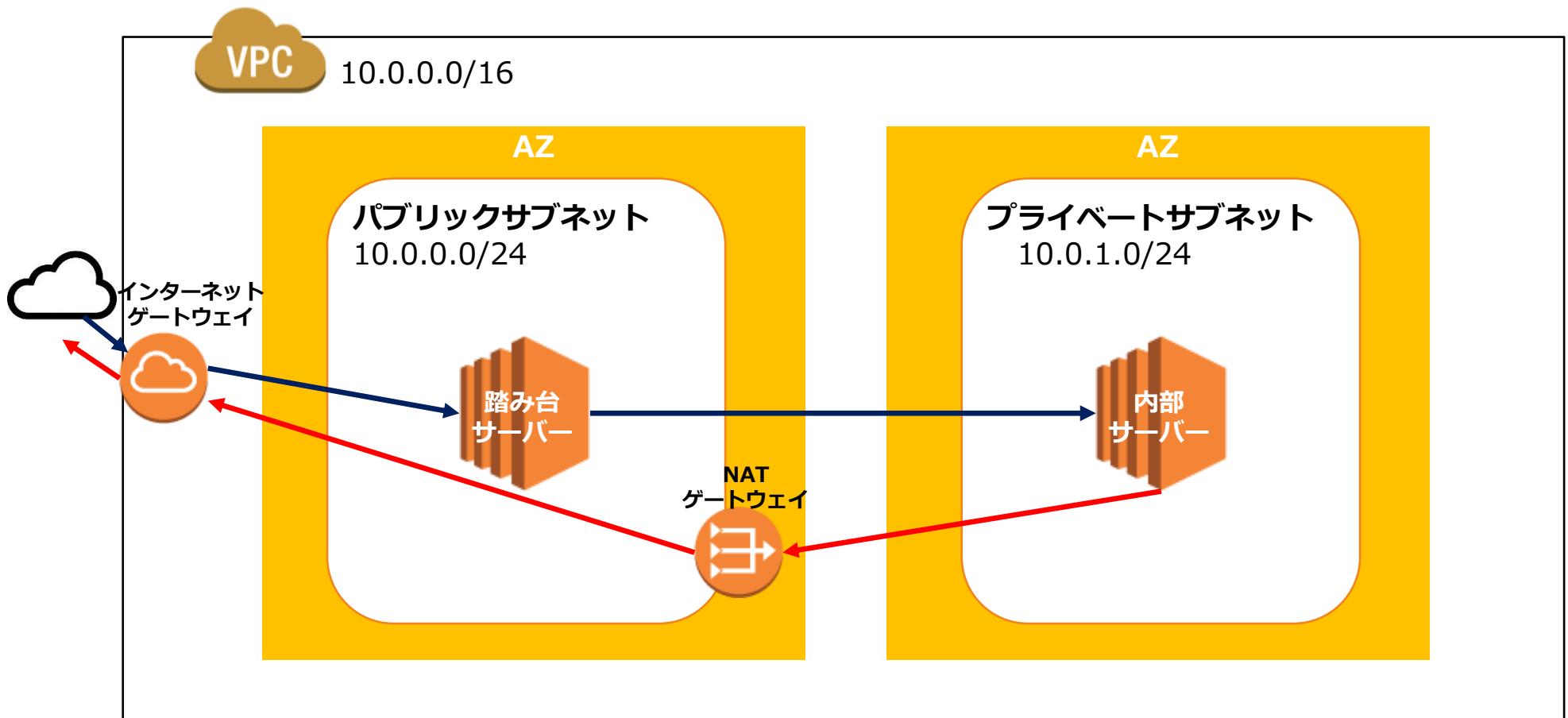
あなたの会社はニュースメディア配信アプリケーションをAWSにホストしています。このアプリケーションはバックエンドサーバーが1つのAZで利用されていることや、NATゲートウェイも同じAZのみに展開されていることが問題となっています。

この問題を解決する最適なAWSアーキテクチャ構成を選択してください。（2つ選択してください）

- 1) 1つのAZにおいてパブリックサブネットとプライベートサブネットを構成して、NATゲートウェイを各パブリックサブネットに設置する。
- 2) 2つのAZにおいてパブリックサブネットとプライベートサブネットを構成して、NATゲートウェイを各パブリックサブネットに設置する。
- 3) 2つのAZにおいてパブリックサブネットとプライベートサブネットを構成して、NATインスタンスを各パブリックサブネットに設置する。
- 4) 各AZにおいてプライベートサブネットからNATゲートウェイ（またはNATインスタンス）にルートを設定する。
- 5) 1つのプライベートサブネットに1つのNATゲートウェイ（またはNATインスタンス）をセットにして、ルートを設定する。

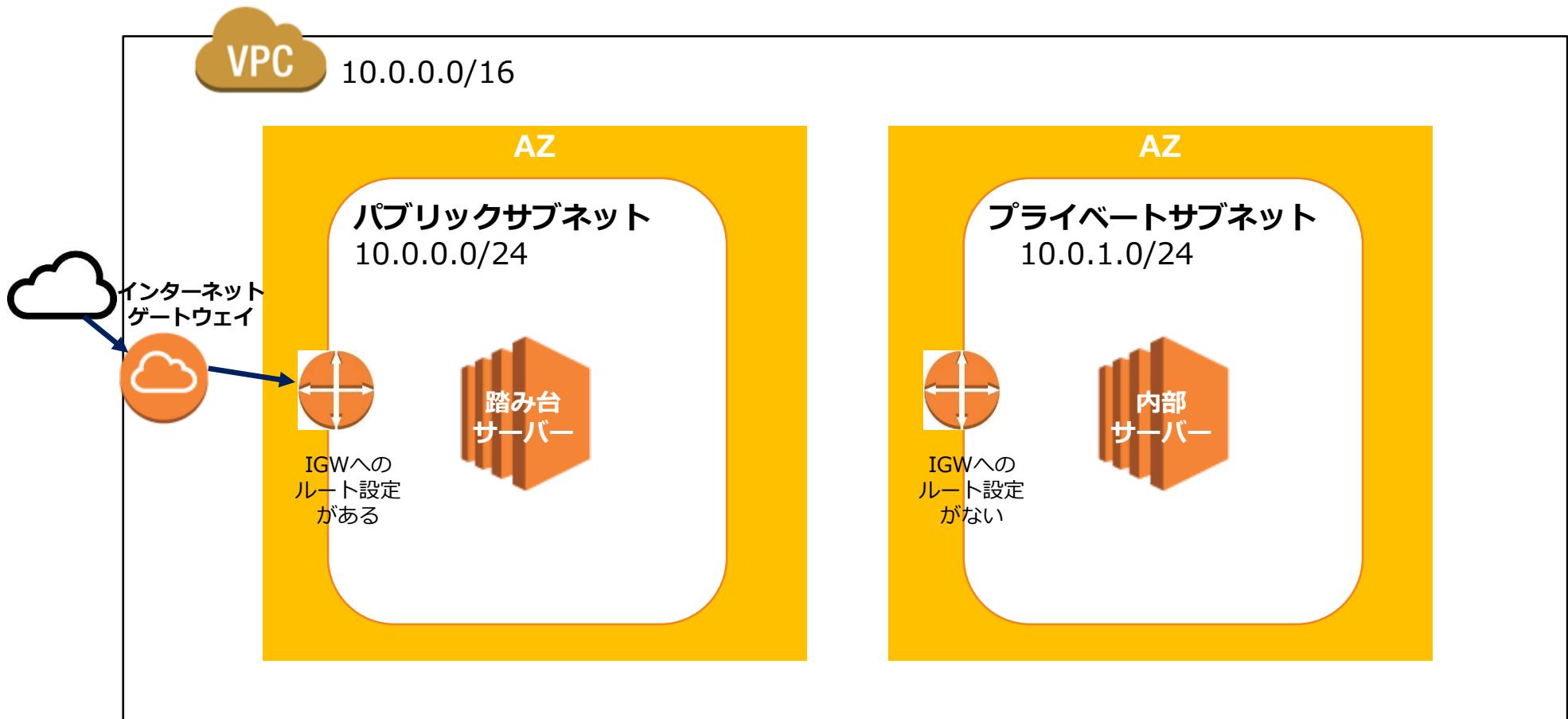
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



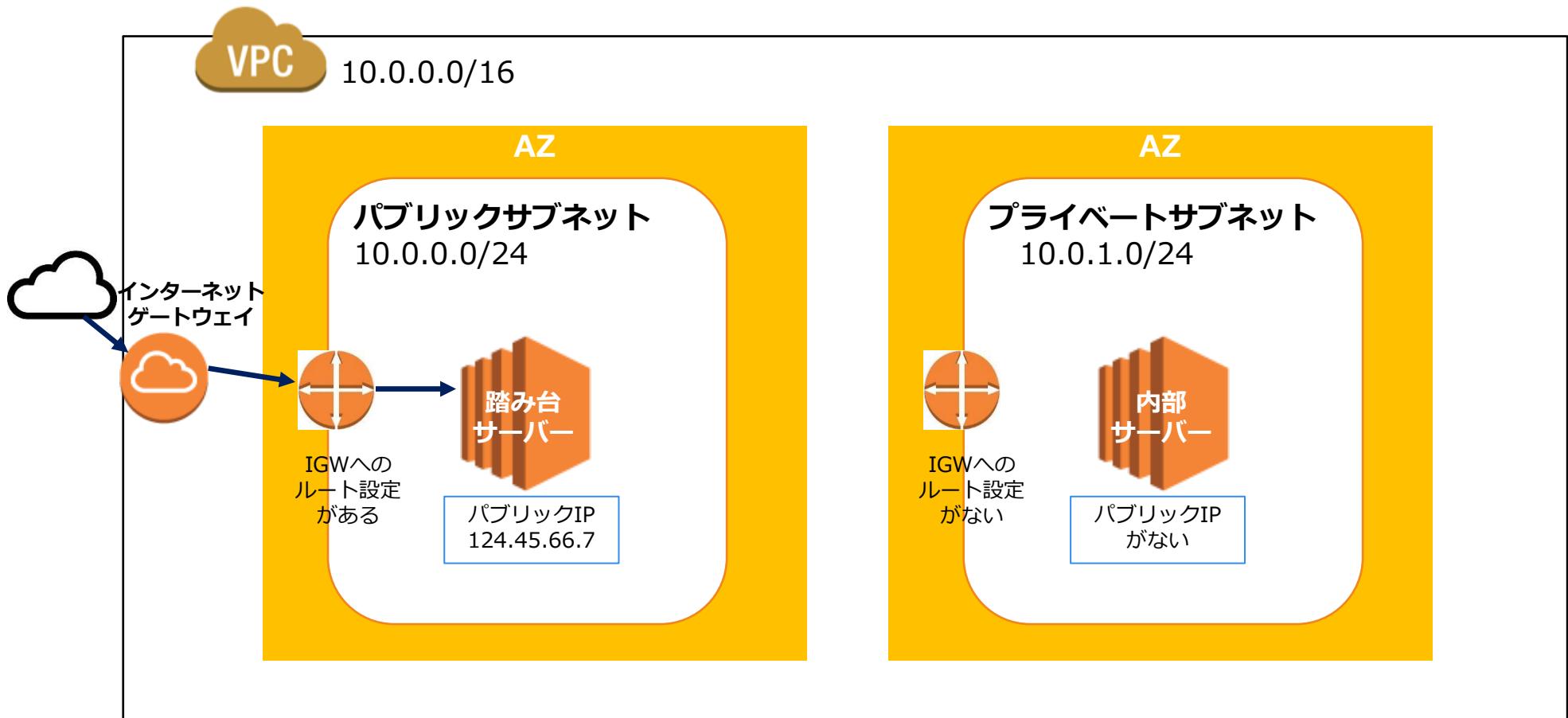
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



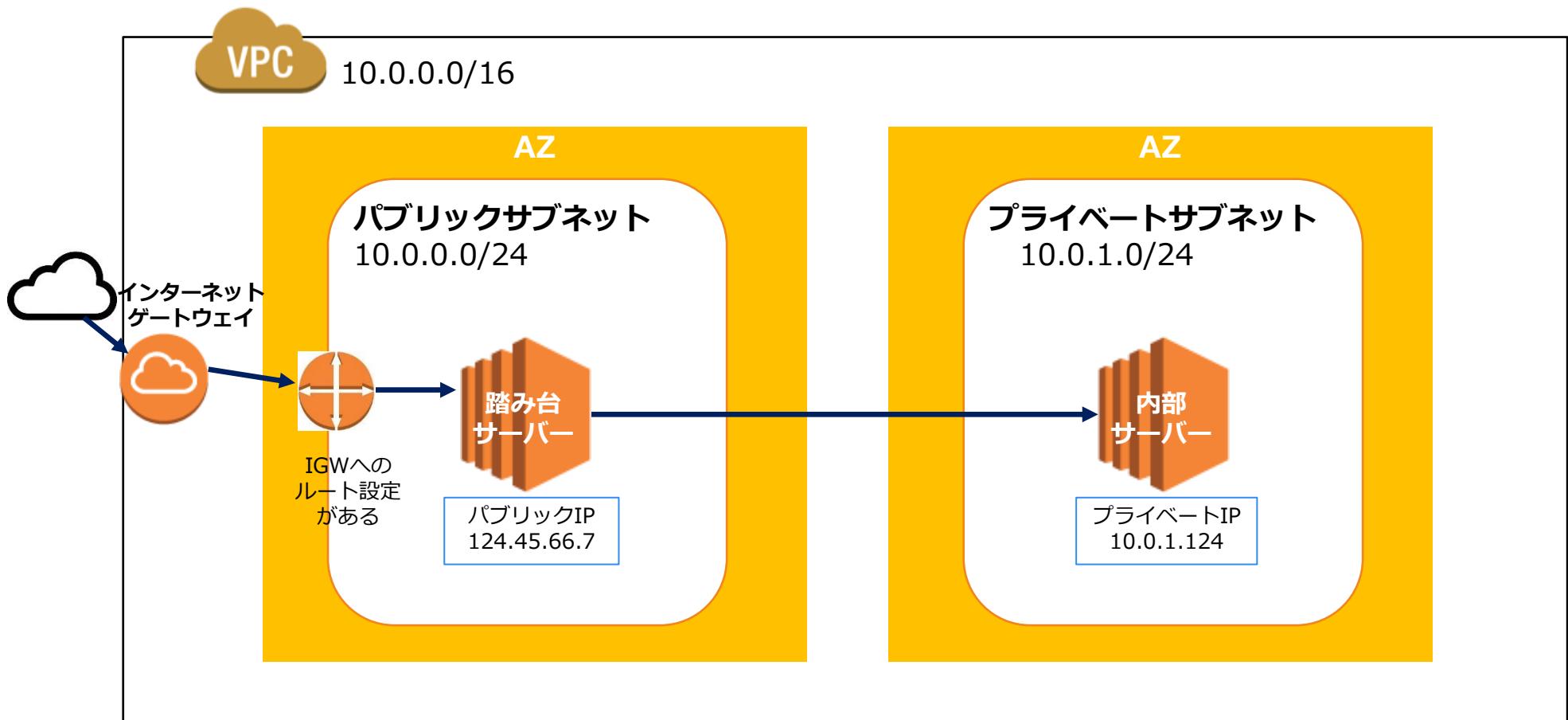
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



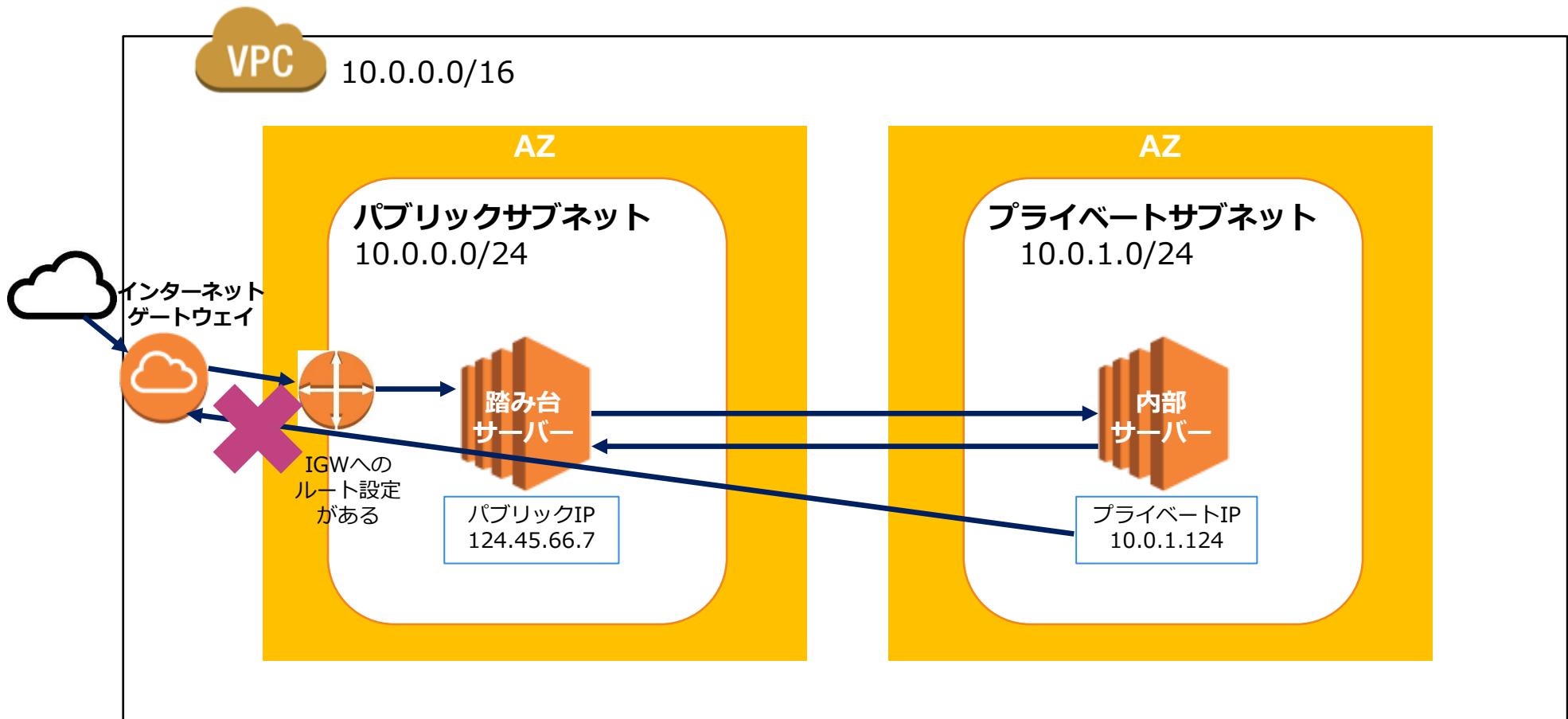
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



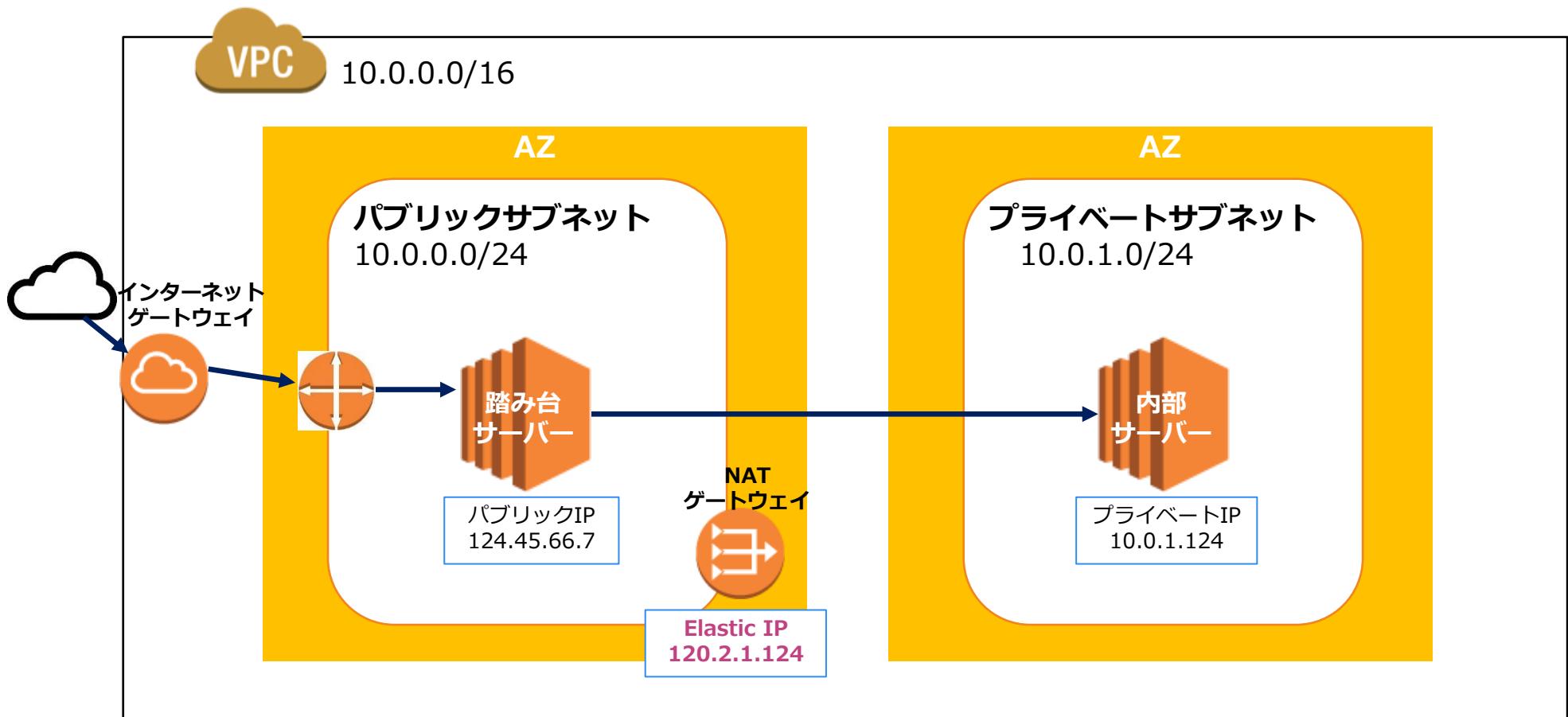
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



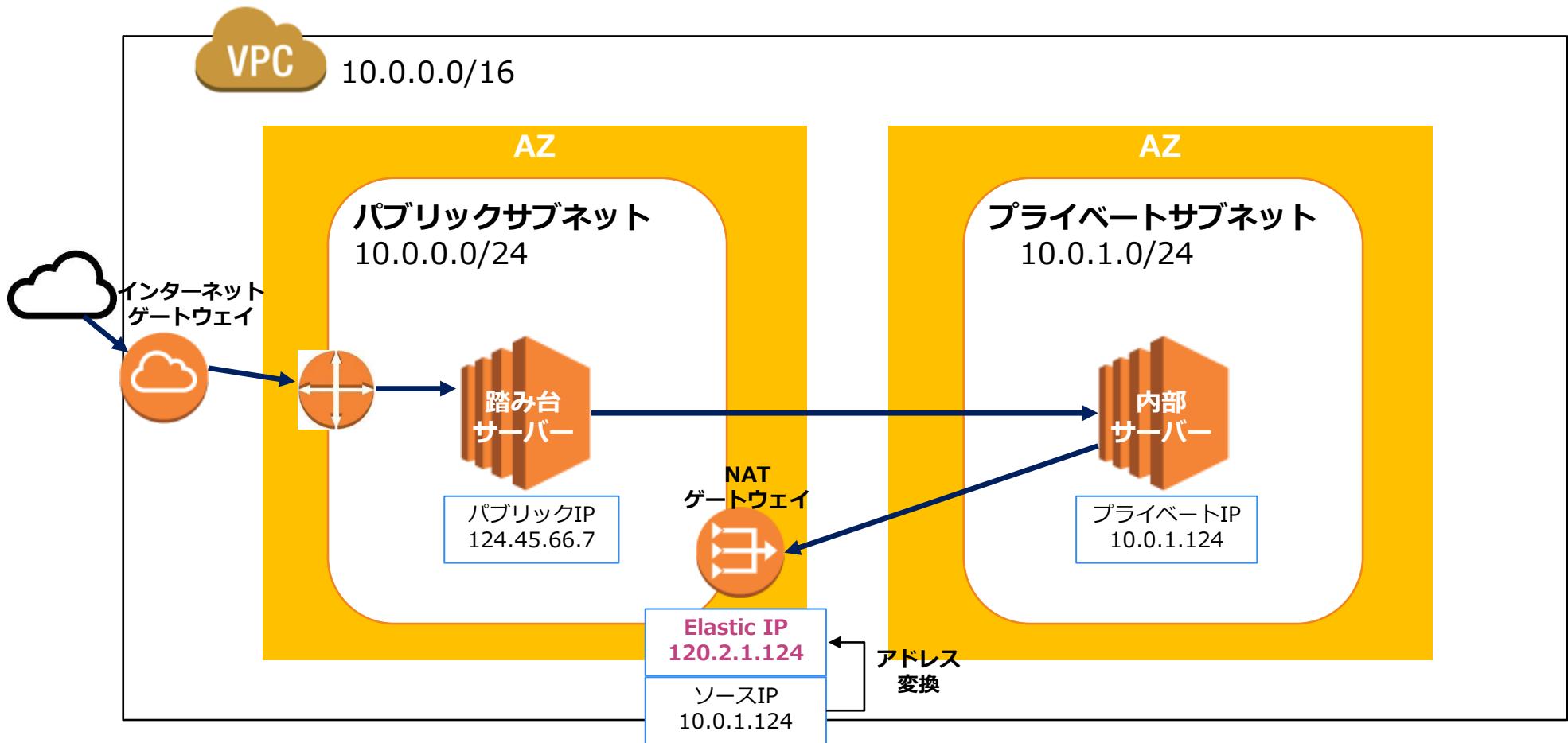
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



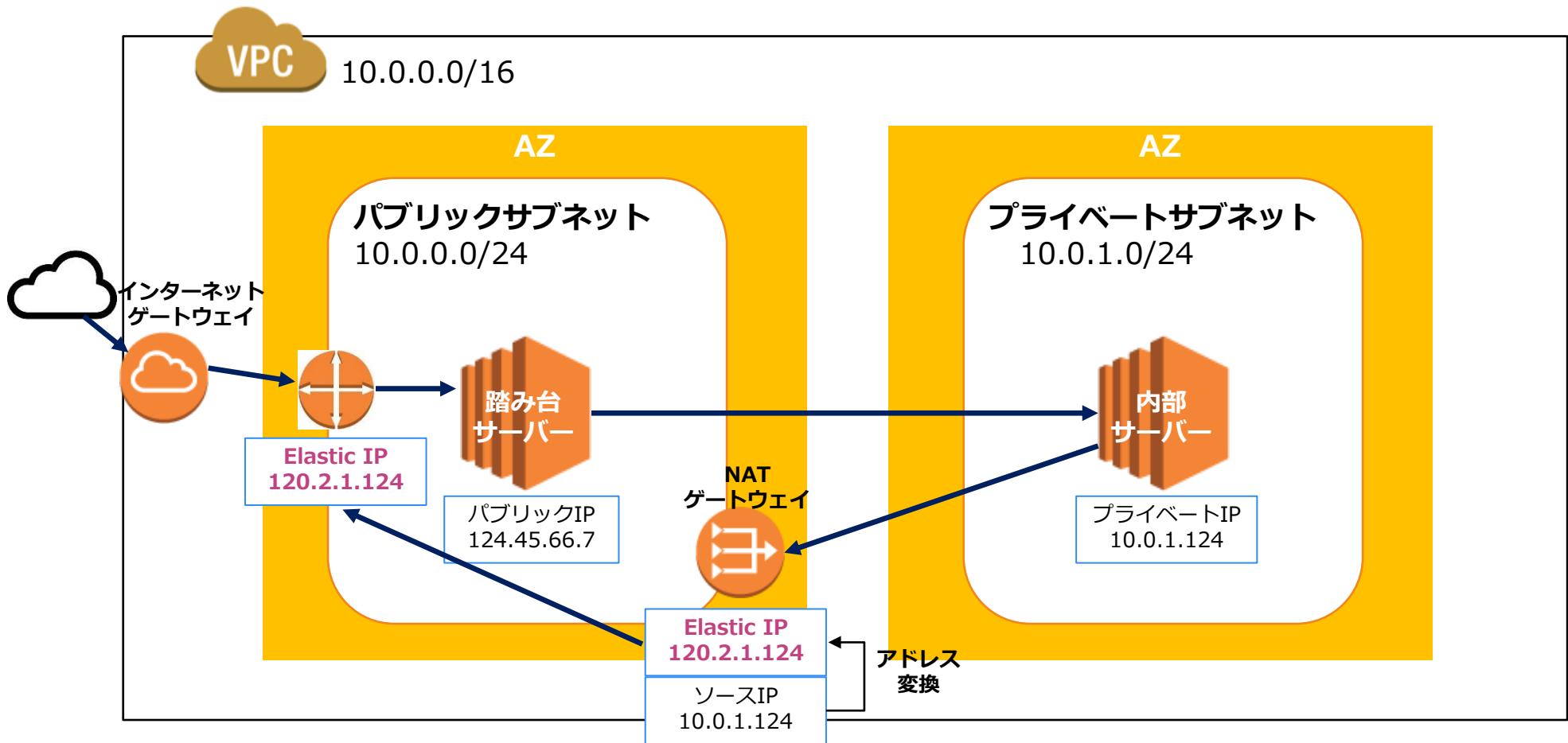
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



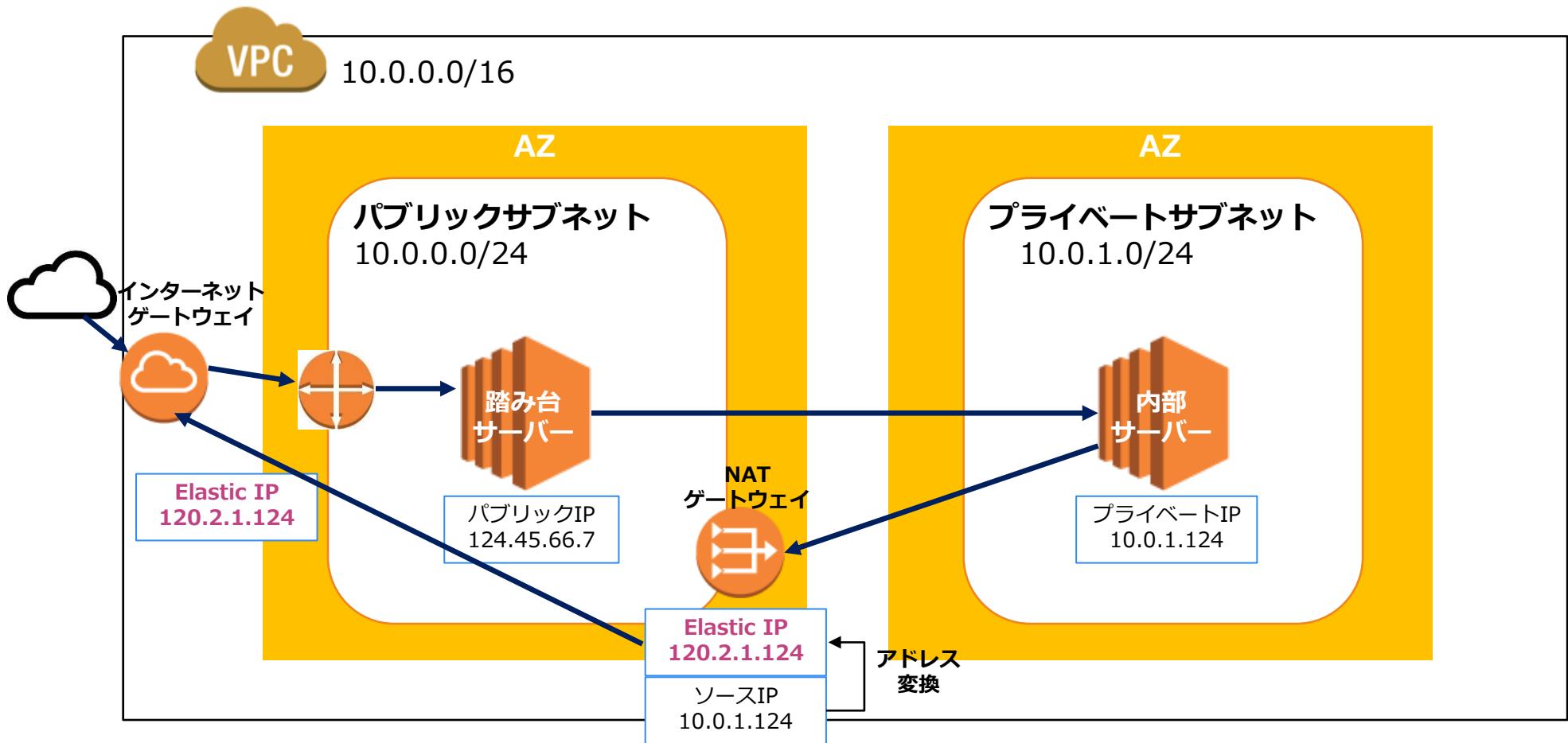
# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



# プライベートサブネットへの外部アクセス

プライベートサブネット内のインスタンスがインターネットに返信を返すにはNATゲートウェイが必要



# [Q] NATインスタンス

あなたは新しくVPCを設定して、パブリックサブネットを2つ、プライベートサブネットを2つ設定してインフラを設置しようと考えています。現在、プライベートサブネットのインスタンスがインターネットへのアウトバウンドIPv4トラフィックを開始できる構成を設定しているところです。そのためにNATインスタンスを構成することが必要です。

NATインスタンスの特徴として正しい説明はどれでしょうか？（3つ選択してください）

- 1) セキュリティグループによりNATインスタンスのトラフィックを制御できる。
- 2) ネットワークACLによりNATインスタンスのトラフィックを制御できる。
- 3) NATインスタンスはポート転送が利用できる。
- 4) NATインスタンスはAWS側で管理されている。
- 5) NATインスタンスはインスタンスタイプは選択できない。

# NATインスタンス

NATゲートウェイはAWS側でマネージド型で提供されており、NATインスタンスに比較して冗長性も高く、管理が楽である

ポイント	NATゲートウェイ	NATインスタンス
現在利用できるリージョン	高可用性。各アベイラビリティーゾーンの NAT ゲートウェイは冗長性を持たせて実装されます。アベイラビリティーゾーンごとに NAT ゲートウェイを作成し、ゾーンに依存しないアーキテクチャにします。	スクリプトを使用してインスタンス間のフェイルオーバーを管理します。
帯域幅	45 Gbps まで拡張できます。	インスタンスタイプの帯域幅に依存します。
メンテナンス	AWS によって管理されます。	ユーザーが管理します
パフォーマンス	ソフトウェアは NAT トラフィックを処理するように最適化されます。	一般的な Amazon Linux AMI が NAT を実行するように設定されます。
Cost	NAT ゲートウェイの使用数、使用期間、NAT ゲートウェイを通じて送信するデータの量に応じて課金されます。	NATインスタンスの使用数、使用期間、インスタンスタイプとサイズに応じて課金されます。
タイプおよびサイズ	一律提供で、タイプやサイズを決める必要はありません。	予測されるワークロードに応じて適切なインスタンスタイプとサイズを選択
パブリック IP アドレス	作成時に NAT ゲートウェイに関連付ける Elastic IP アドレスを選択します。	NAT インスタンスで Elastic IP アドレスまたはパブリック IP アドレスを使用します。
プライベート IP アドレス	ゲートウェイの作成時にサブネットの IP アドレス範囲から自動的に選択	インスタンスの起動時にサブネットの IP アドレス範囲から特定のプライベート IP アドレスを割り当てます。
セキュリティグループ	NAT ゲートウェイに関連付けることはできません。	セキュリティグループでトラフィックをコントロール可能
ネットワーク ACL	ネットワーク ACL を使用して、設置されたサブネットに出入りするトラフィックをコントロールします。	
フローログ	フローログを使用してトラフィックをキャプチャします。	
ポート転送	サポート外。	設定を手動でカスタマイズしてポート転送をサポートします。
踏み台サーバー	サポート外。	踏み台サーバーとして使用します。
タイムアウト動作	接続がタイムアウトになると、NAT ゲートウェイは、NAT ゲートウェイの背後で接続を継続しようとするリソースすべてに RST パケットを返します (FIN パケットは送信しません)。	接続がタイムアウトになると、NAT インスタンスは、接続を閉じるために、NAT インスタンスの背後にあるリソースに FIN パケットを送信します。
IP フラグメント化	UDP プロトコルの IP フラグメント化されたパケットの転送をサポートします。	TCP、UDP、ICMP プロトコルの IP フラグメント化されたパケットの再アセンブルをサポートします。

# [新Q] VPCエンドポイント

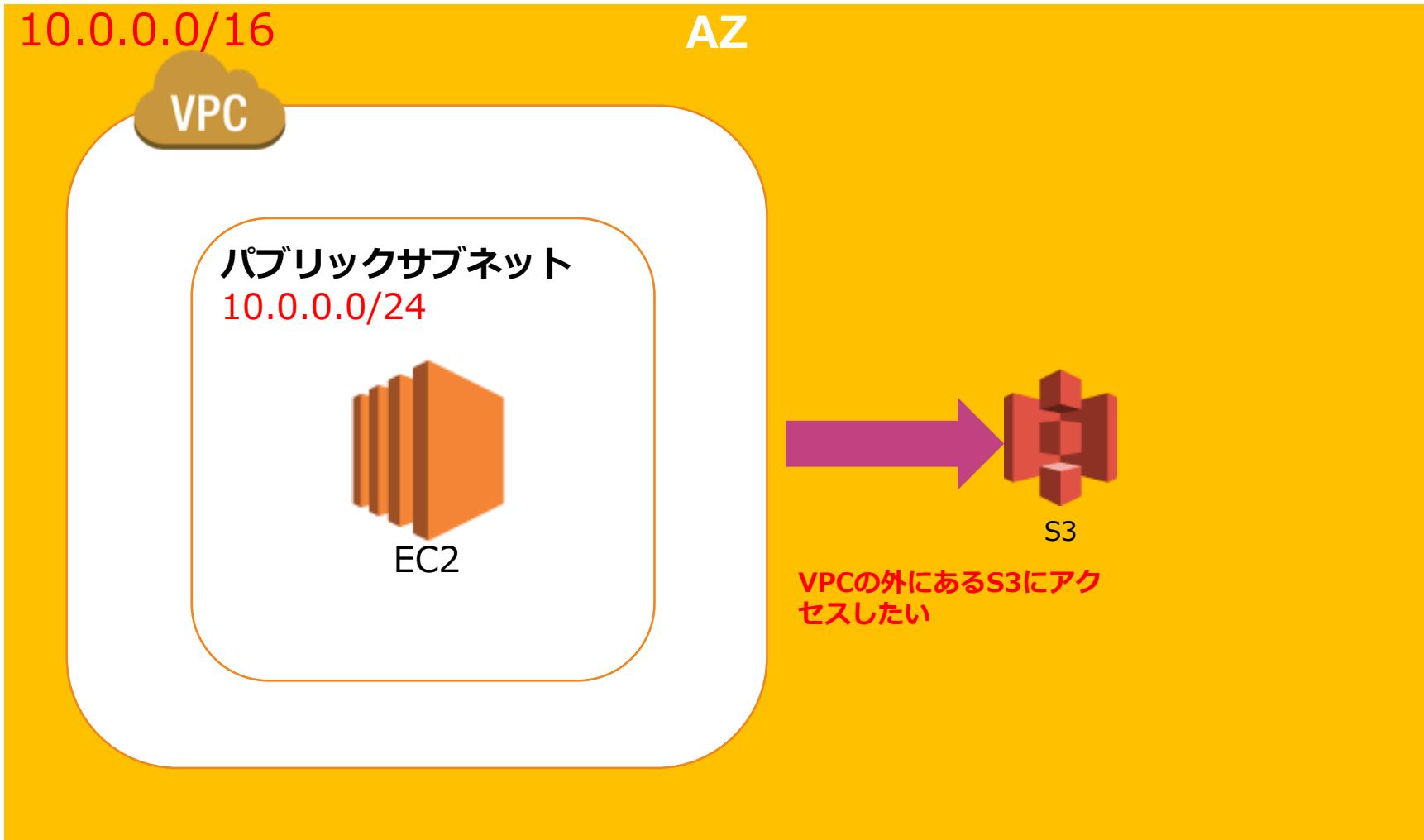
ある企業はAWSを利用して画像共有アプリケーションを運用しています。このアプリケーションはEC2インスタンスを利用して構成されています。このEC2インスタンスには同じAWSリージョン内にある複数のAmazon S3バケット間で、画像を共有する仕組みが必要です。画像をアップロードしたり、ダウンロードが繰り返されるため、データ転送コストが増加してしまいました。あなたはソリューションアーキテクトとして、コスト削減を実施する必要があります。

この要件を満たすために、ソリューションアーキテクトは何を実施するべきでしょうか。 (2つ選択してください。)

- 1) インターフェースエンドポイントをパブリックサブネットにデプロイして、エンドポイント経由でルーティングしてAmazon S3バケットにアクセスするようにルートテーブルを設定する。
- 2) インターフェースエンドポイントをVPCに配置する。そこに、Amazon S3バケットへのアクセスを許可するエンドポイントポリシーをアタッチする。
- 3) アプリケーションをパブリックサブネットにデプロイして、インターネットゲートウェイ経由でルーティングしてAmazon S3バケットにアクセスする。
- 4) ゲートウェイエンドポイントをVPCに配置する。そこに、Amazon S3バケットへのアクセスを許可するエンドポイントポリシーをアタッチする。
- 5) ゲートウェイエンドポイントをVPCに配置する。エンドポイントのプライベート IP アドレスを使用して Amazon S3バケットにアクセスする。

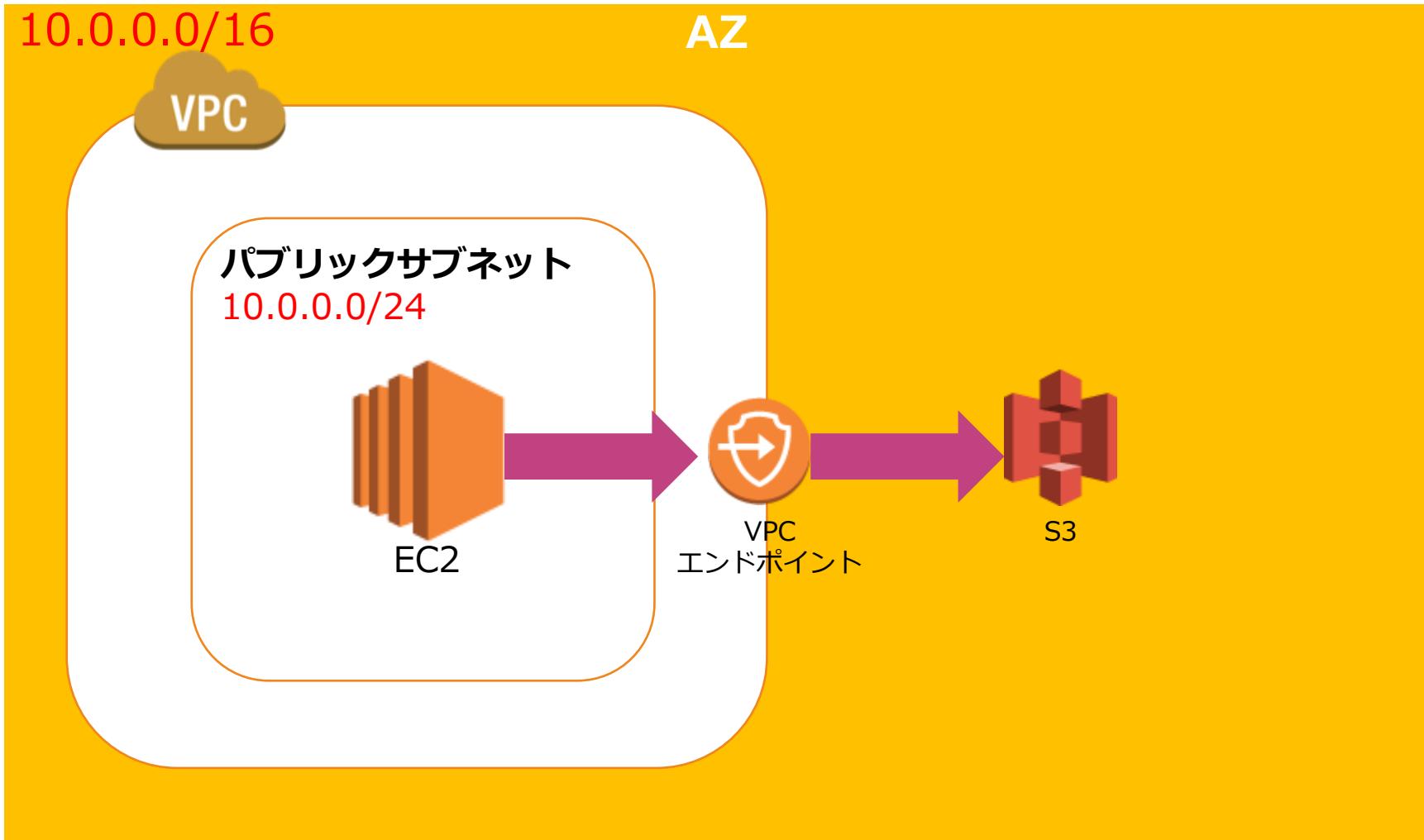
# VPCエンドポイント

VPCエンドポイントはグローバルIPを持つAWSサービスに対して、VPC内から直接アクセスするための出口



# VPCエンドポイント

VPCエンドポイントはグローバルIPを持つAWSサービスに対して、VPC内から直接アクセスするための出口



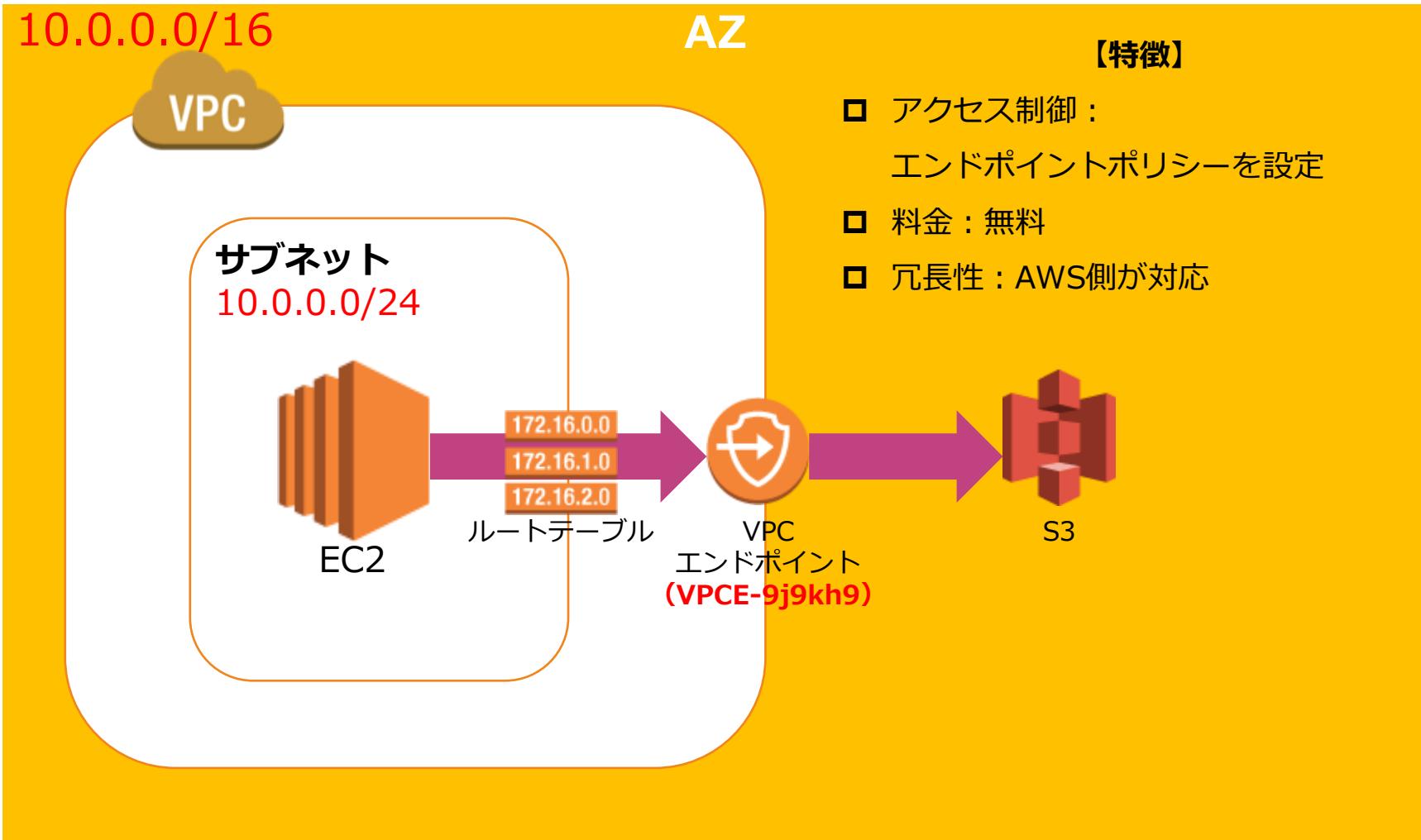
# VPCエンドポイント

ゲートウェイ型はS3とDynamoDBのみに適用され、多くのサービスはプライベートリンク（インターフェース）を利用

ゲートウェイ型 エンドポイント	<ul style="list-style-type: none"><li>✓ サポートされる AWSサービスを宛先とするトラフィックのルートテーブルの宛先として指定できるゲートウェイ</li><li>✓ <b>DynamoDBとAmazon S3のみに適用可能</b></li></ul>
プライベートリンク型 エンドポイント (インターフェース型)	<ul style="list-style-type: none"><li>✓ サポートされるサービスを宛先とするトラフィックのエントリーポイントとして機能するサブネットの IP アドレス範囲のプライベート IP アドレスを持つ Elastic Network Interface</li><li>✓ プライベート IP アドレスを使用してサービスにプライベートにアクセスする。</li><li>✓ AWS PrivateLink は、VPC とサービス間のすべてのネットワークトラフィックを Amazon ネットワークに制限</li><li>✓ RDS、EC2などの多くのAWSサービスに適用可能</li><li>✓ <b>2022年よりAmazon S3に対応。DynamoDBは未対応</b></li></ul>

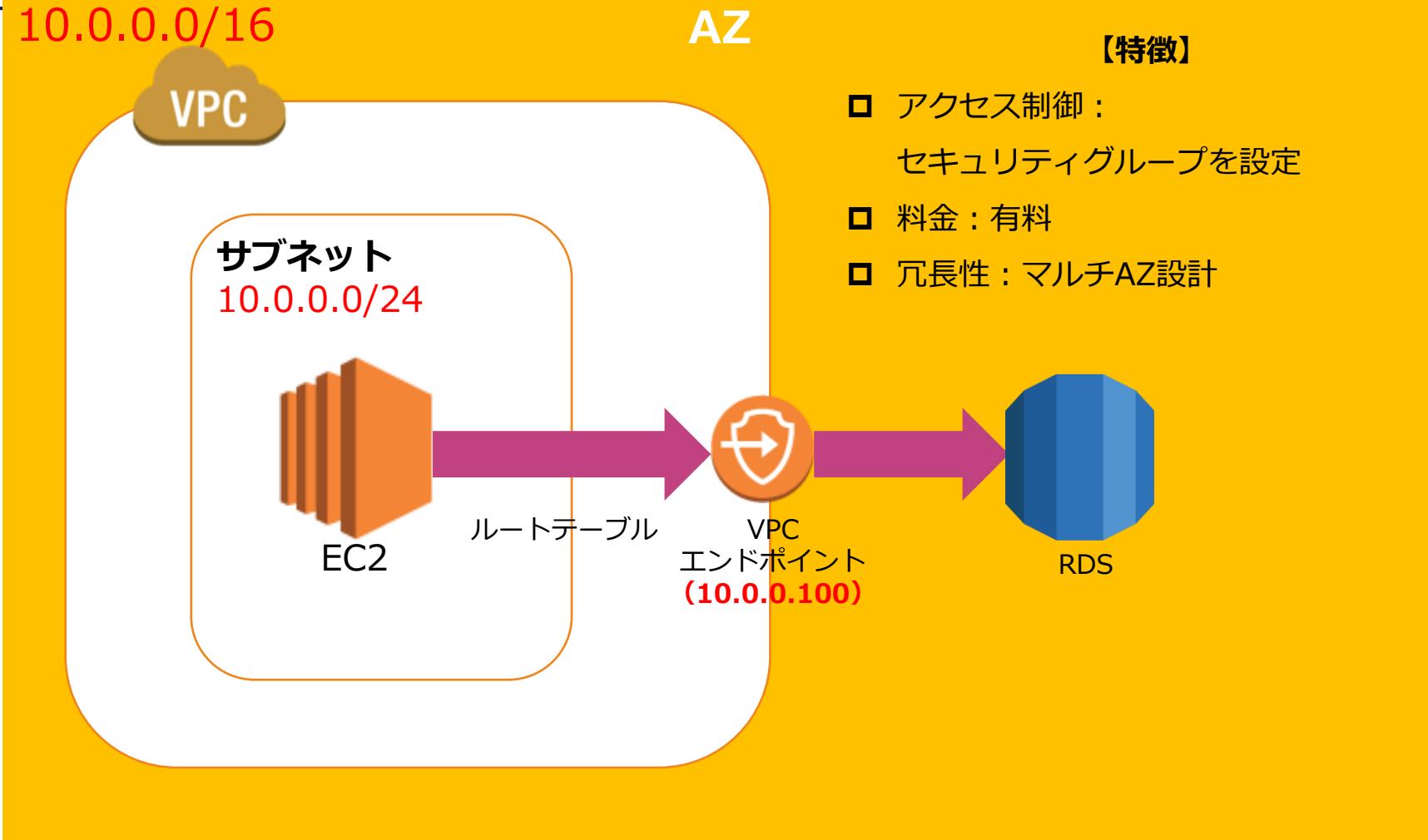
# VPCエンドポイント

ゲートウェイ型はサブネットに特殊なルーティングを設定し、VPC内部から直接外のサービスと通信する



# VPCエンドポイント

プライベートリンク型はサブネットにエンドポイント用のプライベートIPアドレスを生成し、DNSが名前解決でルーティング



# [Q] VPCピアリング

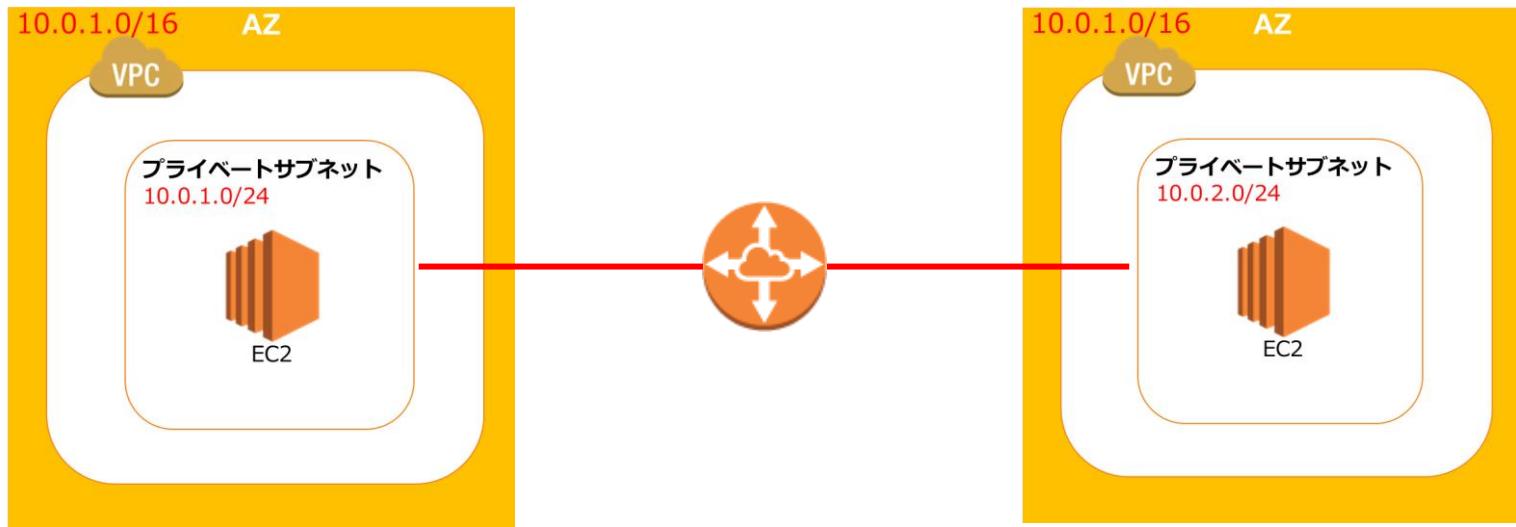
あなたの会社は複数のリージョンに複数のアプリケーションを展開しています。それぞれ別々のVPCを利用していますが、アプリケーション同士を連携することが必要となっています。VPCが異なるアプリケーションが相互に通信できるように、これらのVPCを接続する必要があります。

最適なソリューションは次のうちどれですか？

- 1) VPCピアリング接続を使用する。
- 2) インターネットゲートウェイを使用する。
- 3) VPN接続を使用する。
- 4) Direct Connectを使用する。

# VPC Peering

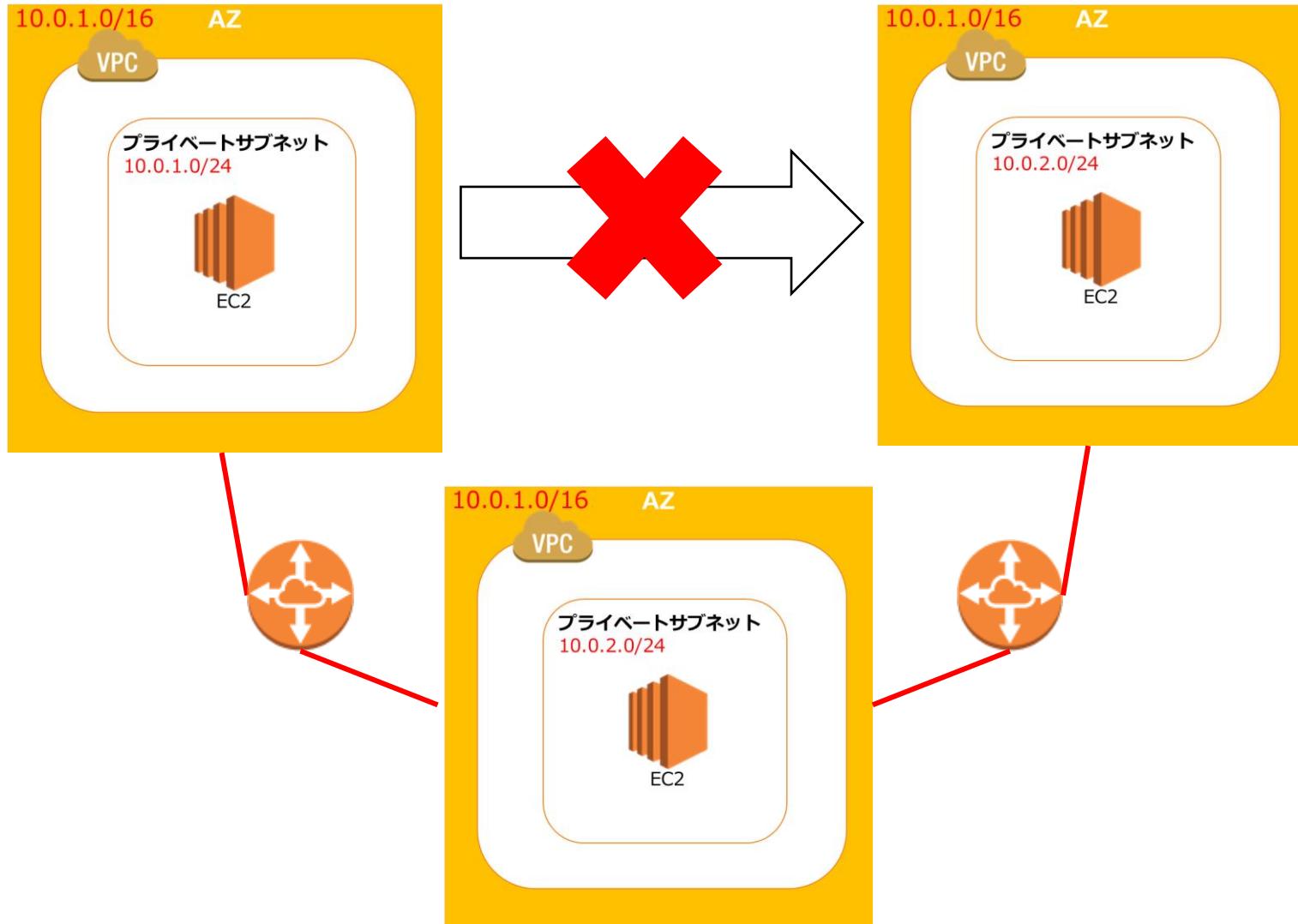
VPC peeringにより2つのVPC間でのトラフィックルーティングが可能



- 異なるAWSアカウント間のVPC間をピア接続可能
- 一部のリージョン間の異なるVPC間のピア接続も可能
- 単一障害点や帯域幅のボトルネックは存在しない

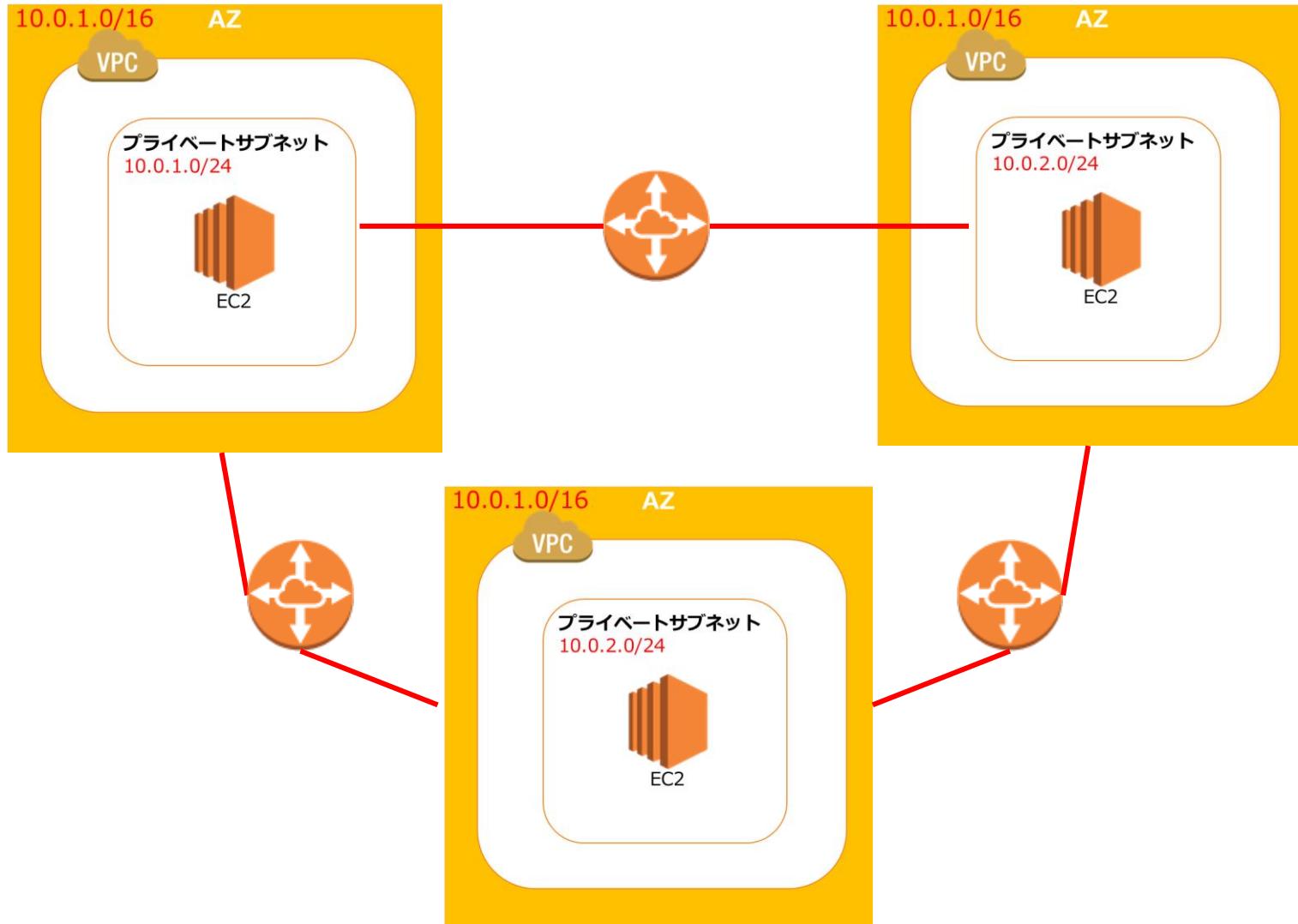
# VPC Peering

VPC peeringにより2つのVPC間でのトラフィックルーティングが可能



# VPC Peering

VPC peeringにより2つのVPC間でのトラフィックルーティングが可能



# [Q]ネットワークACL

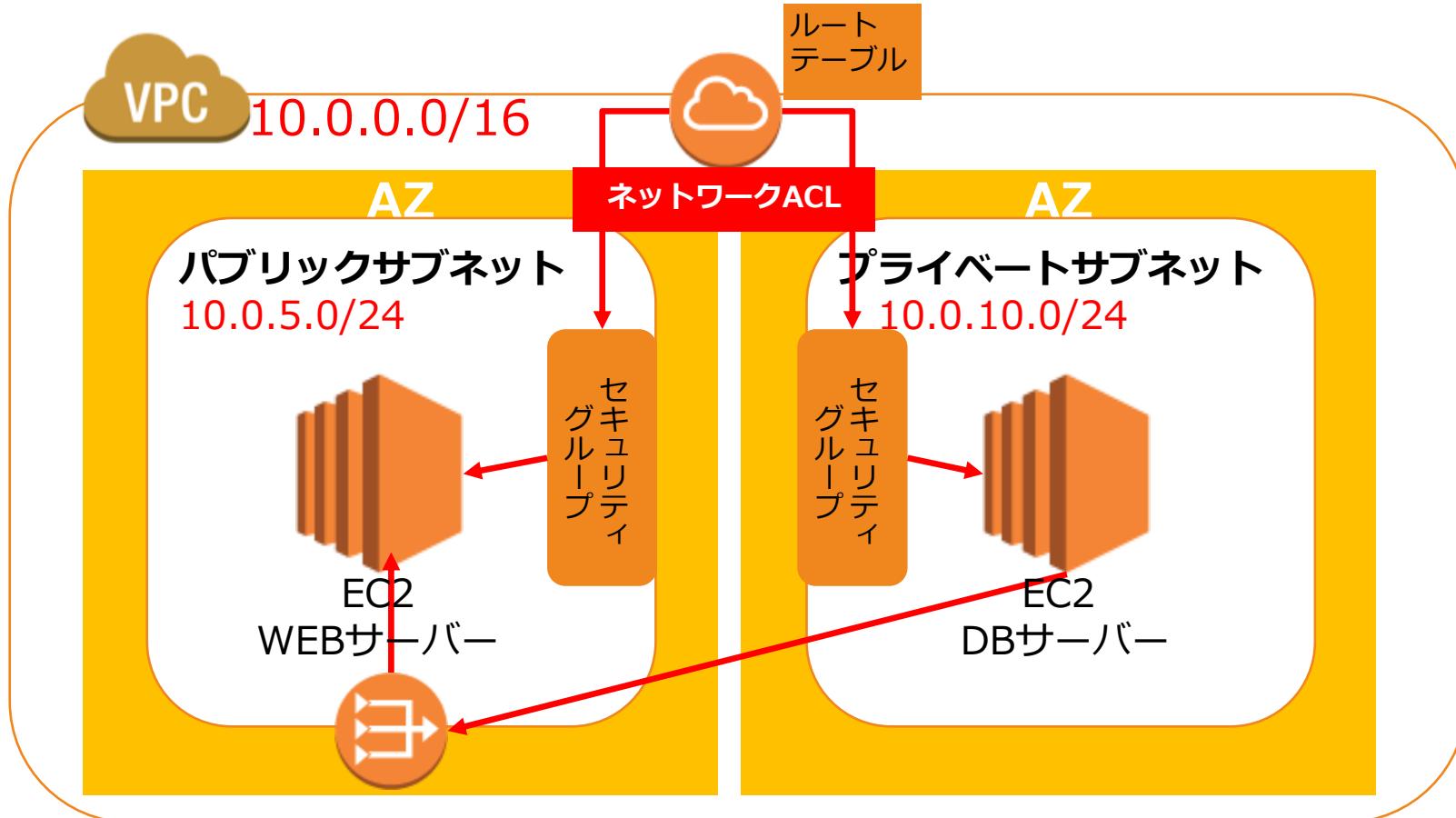
あなたの会社はAWSを利用したWEBアプリケーションを運用しています。最近になって不正アクセスを試みるようなトラフィックが急増しています。いくつかの固定されたIPアドレスから不正アクセスが試みられているようです。このリクエストは同じCIDR範囲内の異なるIPアドレスから実施されているようです。

このアクセスに対して直接効果がある保護策を選択してください。

- 1) ネットワークACLのインバウンドテーブルにおいて、他のルールよりも小さいルール番号で該当するCIDRを拒否する。
- 2) ネットワークACLのアウトバウンドテーブルにおいて、他のルールよりも小さいルール番号で該当するCIDRを拒否する。
- 3) セキュリティグループのインバウンドテーブルにおいて、他のルールよりも小さいルール番号で該当するCIDRを拒否する。
- 4) セキュリティグループのアウトバウンドテーブルにおいて、他のルールよりも小さいルール番号で該当するCIDRを拒否する。

# ネットワークACL

ネットワークACLによるアクセス制御を追加する



# 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]ネットワークACL

あなたはVPCを構築して、2つのサブネットを作成しました。現在は、ネットワークACLの設定を行っているところです。その際には、VPCを設置した際に設定されるデフォルトのネットワークACLを利用する予定です。

ネットワークACLのデフォルト設定に関する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) すべてのトラフィックを拒否するデフォルトのインバウンドルールが設定されている。
- 2) すべてのトラフィックを拒否するデフォルトのアウトバウンドルールが設定されている。
- 3) すべてのトラフィックを許可するデフォルトのインバウンドルールが設定されている。
- 4) すべてのトラフィックを許可するデフォルトのアウトバウンドルールが設定されている。
- 5) インターネットゲートウェイへのトラフィックを許可するデフォルトのアウトバウンドルールがある。

# ネットワークACL

ネットワークのデフォルトの構成はデフォルトとカスタムで異なる

VPCに最初に設定される  
デフォルトNACL

- ✓ 全てのインバウンドトラフィックを許可する設定がされている。
- ✓ 全てのアウトバウンドトラフィックを許可する設定がされている。

カスタムで作成する  
NACLのデフォルト設定

- ✓ 全てのインバウンドトラフィックを拒否する設定がされている。
- ✓ 全てのアウトバウンドトラフィックを拒否する設定がされている。

# [Q]ネットワークACLの設定

あなたはVPCを構築して、2つのサブネットを作成しました。現在は、ネットワークACLの設定を行っているところです。

The screenshot shows the AWS Network ACL inbound rules editor. The title bar says "ネットワーク ACL > インバウンドのルールの編集". The main heading is "インバウンドのルールの編集". Below it, it says "ネットワーク ACL acl-326cbd54". A table lists three rules:

ルール #	タイプ	プロトコル	ポート範囲	送信元	許可 / 拒否
98	HTTP (80)	TCP (6)	80	121.103.215.159/32	DENY
99	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW

At the bottom left is a "ルールの追加" button, and at the bottom center is a note "\* 必須".

このネットワークACLが適用されたサブネットにあるWEBサーバーに121.103.215.159からアクセスがあった場合にどのようにになりますか？

- 1) 121.103. 215.159からのSSH接続が許可される。
- 2) 121.103. 215.159からのSSH接続が拒否される。
- 3) 121.103. 215.159からHTTP経由でのWEBサイトにアクセスできる。
- 4) 121.103. 215.159からHTTP経由でのWEBサイトにアクセスできない。
- 5) 121.103. 215.159からHTTPS経由でのWEBサイトにアクセスできる。

# ネットワークACLの設定

トライ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

ネットワーク ACL > インバウンドのルールの編集

## インバウンドのルールの編集

ネットワーク ACL acl-326cbd54

ルール #	タイプ	プロトコル	ポート範囲 <small>i</small>	送信元 <small>i</small>	許可 / 拒否
98	HTTP (80)	TCP (6)	80	121.103.215.159/32	DENY
99	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW

[ルールの追加](#)

\* 必須

# [Q]サブネットによる構成

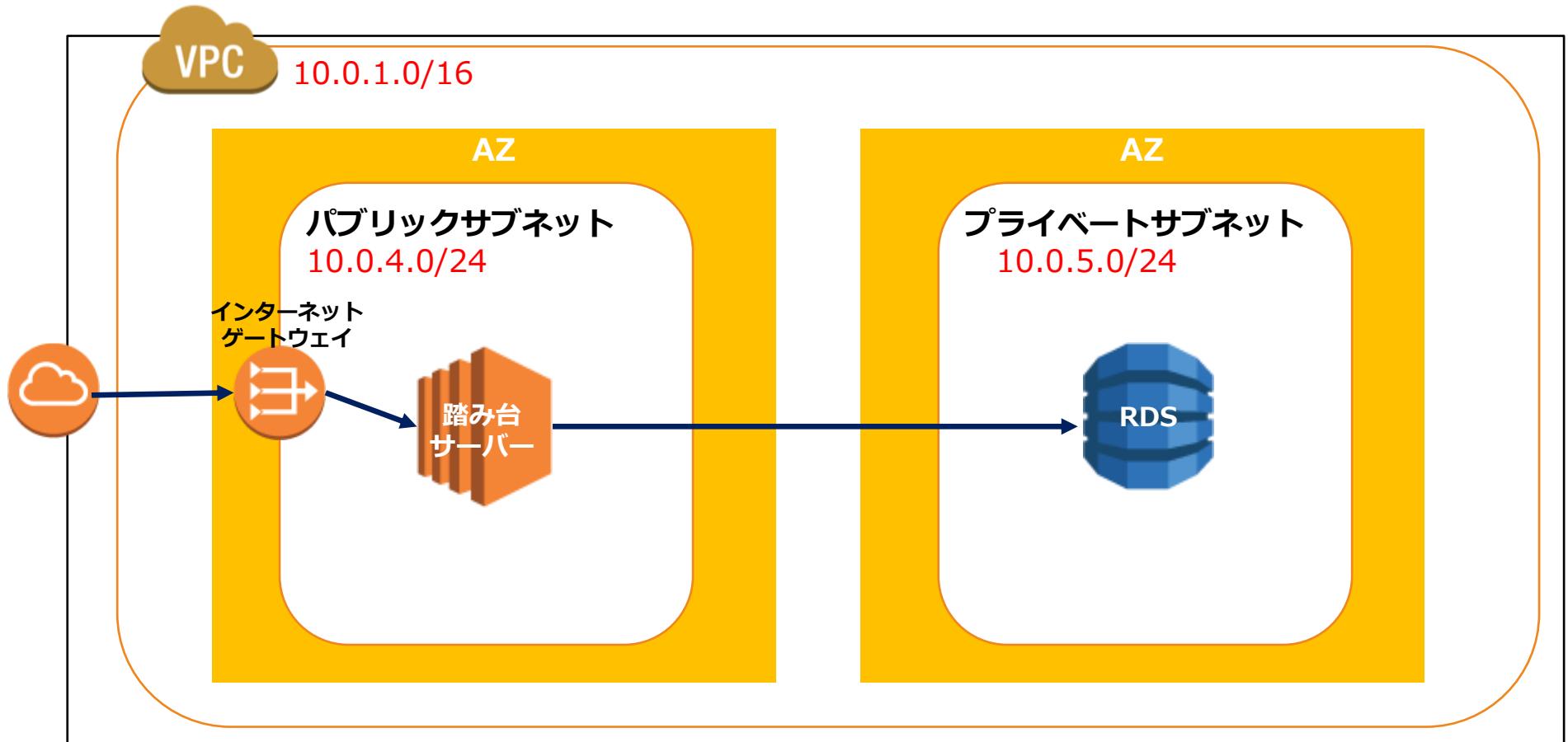
あなたはAWSにWEBアプリケーションをホストすることを計画しています。まずはVPCを作成して、パブリックサブネットにWEBサーバーとなるEC2インスタンスを起動しました。さらにMySQLデータベースをホストする別のEC2インスタンスを別のサブネットに設置して、WEBサーバーから接続します。

安全性を考慮すると、どのようにデータベースをセットアップするべきでしょうか。  
(2つ選択してください。)

- 1) データベースサーバーをプライベートサブネットに配置する。
- 2) データベースサーバーをパブリックサブネットに配置する。
- 3) セキュリティグループでWEBサーバーのIPアドレスを指定して、MySQLからのポート番号のみを許可する設定をDB側のインスタンスに設定する。
- 4) セキュリティグループでWEBサーバーのIPアドレスを指定して、MySQLからのポート番号のみを許可する設定をWEBサーバー側のインスタンスに設定する。
- 5) IAMデータベース認証でWEBサーバーのIPアドレスを指定して、MySQLからのポート番号のみを許可する設定をWEBサーバー側のインスタンスに設定する。

# サブネットによる構成

セキュリティを高めたいサービスはプライベートサブネットに設置する



# [Q] VPC内サービスへの接続：SSH

あなたはAWSアカウントを新規に開設して、まずはVPCを構成することにしました。VPCウィザードを使用することでよく利用されるVPC構成を迅速に設定することが可能です。セキュリティを高めるためにプライベートなアクセスに限定したデータベースサーバーを設置するためのネットワーク構成が必要です。パブリックサブネットに踏み台サーバーを設定して、SSH経由で企業データセンターからのみアクセスする必要があります。

これを達成するための最適な方法はどれでしょうか？（2つ選択してください。）

- 1) パブリックサブネットにEC2インスタンスを起動する。
- 2) プライベートサブネットにEC2インスタンスを起動する。
- 3) 企業データセンターのIPアドレスを介したポート22でのアクセスのみを許可するセキュリティグループをインスタンスに付与して、Pemキーでアクセスを実施する。
- 4) 企業データセンターのIPアドレスを介したポート22でのアクセスのみを許可するセキュリティグループをインスタンスに付与して、アクセスキーでアクセスを実施する。
- 5) 企業データセンターのIPアドレスを介したポート22でのアクセスのみを許可するセキュリティグループをインスタンスに付与して、ユーザーIDとパスワードでアクセスを実施する。

# [Q] VPC内サービスへの接続：RDP

あなたはAWSアカウントを新規に開設して、まずはVPCを構成することにしました。セキュリティを高めるためにプライベートなアクセスに限定したWEBサーバーを設置する予定です。Microsoftリモートデスクトッププロトコル（RDP）アクセスを備えた踏み台サーバーを利用して、すべてのインスタンスへの管理アクセスを制限したいと考えています。

次の踏み台サーバーの設定をどのように実施するべきでしょうか？（2つ選択してください。）

- 1) パブリックサブネットにElasticIPアドレスを設定したEC2インスタンスを起動する。
- 2) プライベートサブネットにElasticIPアドレスを設定したEC2インスタンスを起動する。
- 3) プライベートサブネットにプライベートIPアドレスを設定したEC2インスタンスを起動する。
- 4) セキュリティグループで企業IPアドレスからのみEC2インスタンスへとRDPアクセスを22ポートで許可する設定を付与する。
- 5) セキュリティグループで企業IPアドレスからのみEC2インスタンスへとRDPアクセスを3389ポートで許可する設定を付与する。

# VPC内サービスへの接続

VPC内のサービスに接続する際はネットワークACLとセキュリティグループでの許可が必要

## SSH接続

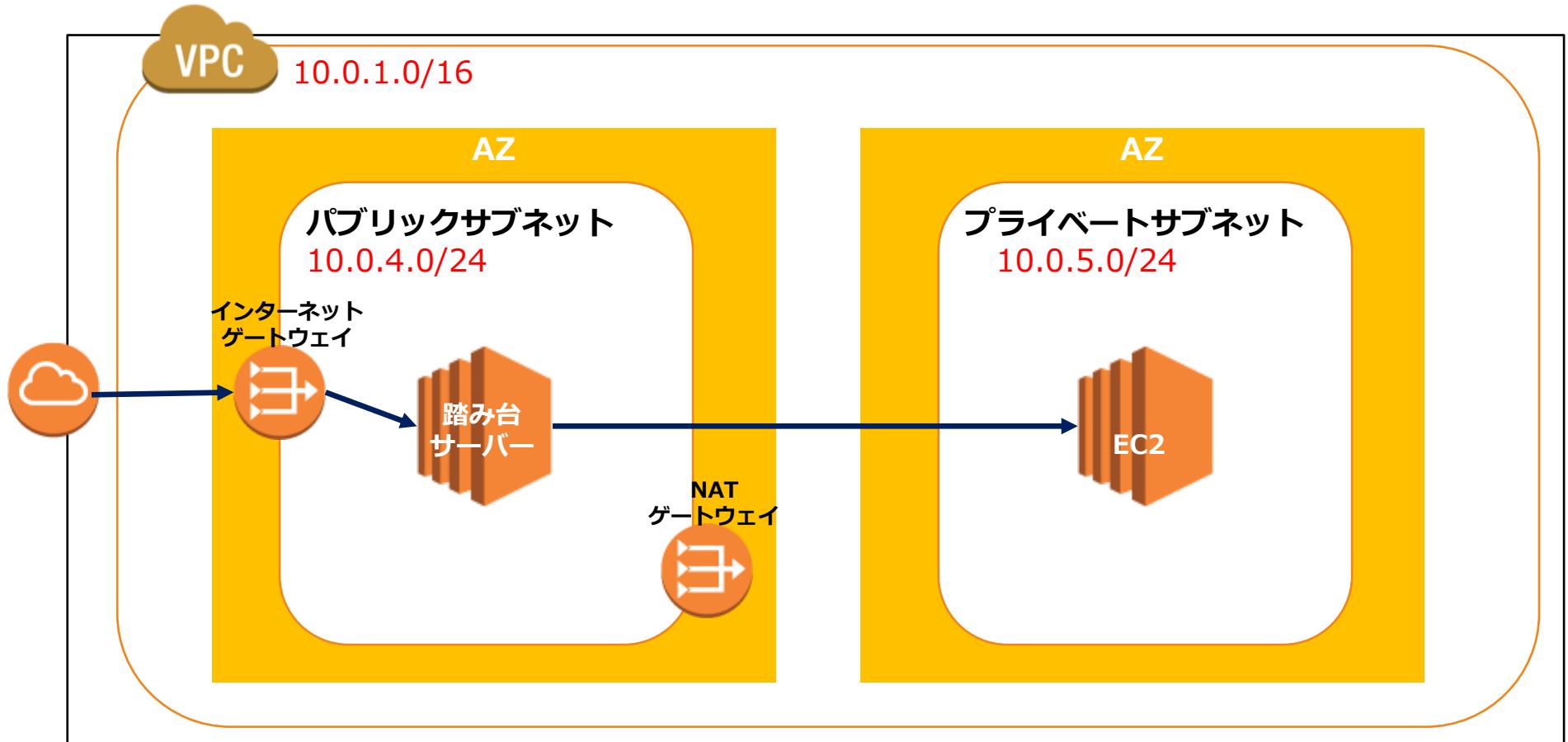
- ✓ SSHはインスタンスへの標準的な接続に利用するプロトコル
- ✓ セキュリティグループ／ネットワークACLで接続するIPアドレスを指定してポート22番のSSHを許可する。
- ✓ パブリックIPアドレス／EIPを指定してPEMキーを利用してインスタンスにアクセスを実施する。

## RDP接続

- ✓ リモートデスクトップによる接続方式
- ✓ RDPはリモートデスクトップ用の接続プロトコル
- ✓ パブリックサブネットに踏み台サーバー（Bastionサーバー）を設置して、Elastic IPを付与する。
- ✓ セキュリティグループ／ネットワークACLで接続するIPアドレスを指定してポート3389番のRDPを許可する。

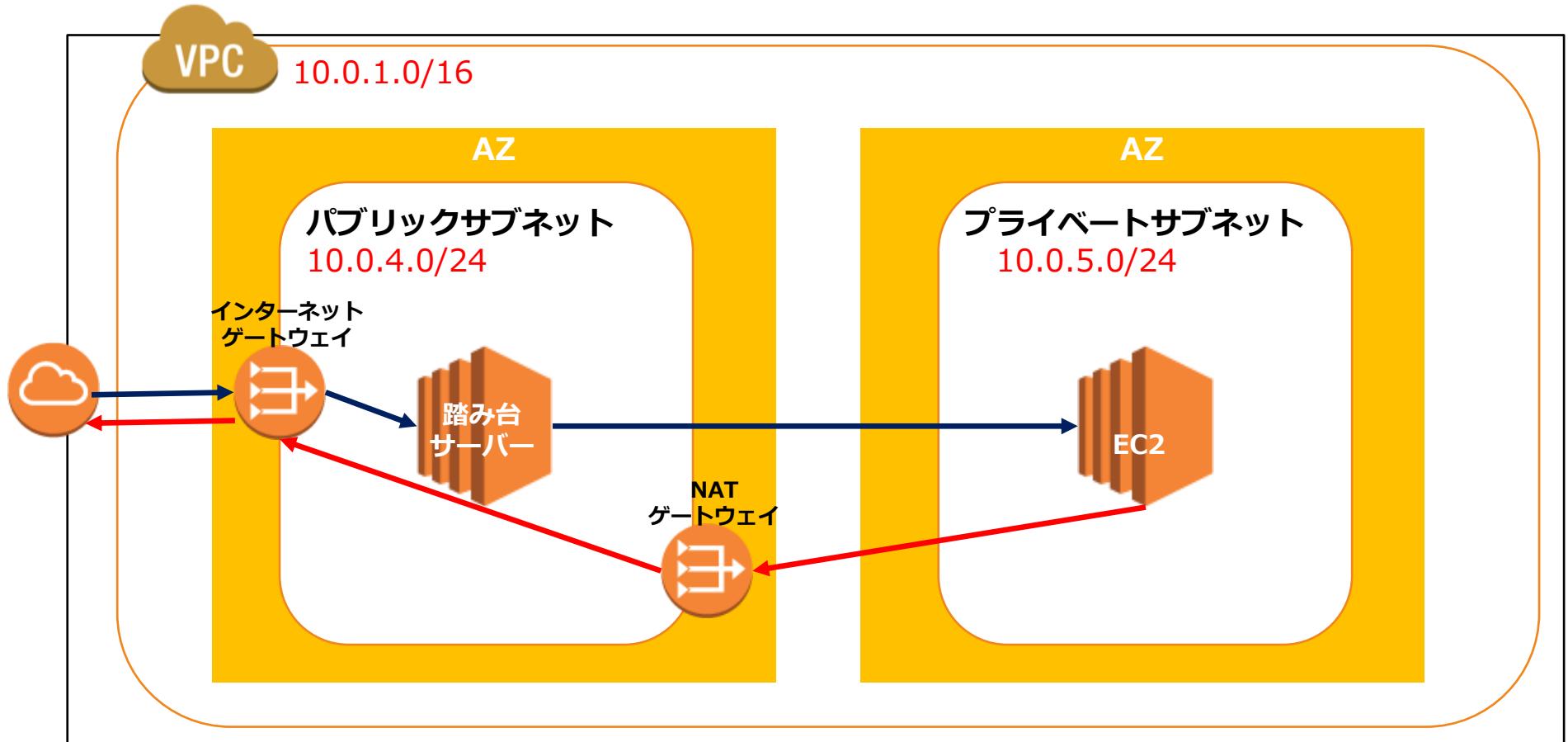
# 踏み台サーバー

プライベートサブネット内のインスタンスに接続するには踏み台サーバーが必要。戻りトラフィックにはNATゲートウェイが必要



# 踏み台サーバー

プライベートサブネット内のインスタンスに接続するには踏み台サーバーが必要。戻りトラフィックにはNATゲートウェイが必要



## [Q] VPCフロー-ログ

あなたはVPCをセッティングしてAWSリソースを利用しています。WEBアプリケーション用にVPC内に複数のEC2インスタンスを起動しており、ELBによるトラフィック分散を実行しています。モニタリングの一環として、ELBに到達するトラフィックに関する情報をキャプチャする必要があります。

このデータを収集するための最適な方法を選択してください。

- 1) ELBが関連付けられたEC2インスタンスのVPCフロー-ログを有効化する。
- 2) Amazon CloudWatch Logsを使用して、ELBからのログを確認する。
- 3) ELBに関連付けられたネットワークインターフェイスにVPCフロー-ログを有効化する。
- 4) ELBが実行されているサブネットに対してVPCフロー-ログを有効化する。

# VPCフロー・ログ

VPCフロー・ログはネットワークトラフィックを取得し  
CloudWatchでモニタリングできるようにする機能

- ネットワークインターフェースを送信元/ 送信先とするトラフィックが対象となる。
- セキュリティグループとネットワークACLのルールでaccepted/rejectされたトラフィックを取得する
- キャプチャウインドウと言われる時間枠 (約10分間)で収集・プロセッシング・保存する
- RDS、Redshift、ElasticCache、WorkSpacesのネットワークインターフェーストラフィックも取得できる。
- 追加料金はなし

# [Q]VPCにおけるDNSの使用

あなたはAWSアカウントを新規に開設して、EC2インスタンスを起動させました。このEC2インスタンスにはカスタムVPCが設定されています。このEC2インスタンスをWEBサーバーとして利用してPintor.comというカスタムドメインを設定したいと考えています。あなたはソリューションアーキテクトとして、これを実現するためRoute53のプライベートホストゾーン機能を使用したいと考えています。

次のVPC設定のどれを有効にする必要がありますか？（2つ選択してください）

- 1) enableDnsHostnames
- 2) enableDnsSupport
- 3) enableVpcSupport
- 4) enableVpcHostnames
- 5) enableDnsDomain

# VPCにおけるDNSの使用

VPC 内で起動したインスタンスがパブリック IP アドレスに対応するパブリック DNS ホスト名を受け取るための設定が必要

## enableDnsHostname S

- ✓ パブリック IP アドレスを持つインスタンスが、対応するパブリック DNS ホスト名を取得可能か否かを示す。
- ✓ この属性が true で enableDnsSupport 属性も true の場合、VPC 内のインスタンスは DNS ホスト名を取得する。

## enableDnsSupport

- ✓ DNS 解決がサポートされているかどうかを示す。
- ✓ この属性が false の場合、パブリック DNS ホスト名を IP アドレスに解決する Amazon Route 53 Resolver サーバーが機能しない。
- ✓ この属性が true の場合、Amazon が提供する DNS サーバー (IP アドレス 169.254.169.253) へのクエリ、またはリザーブド IP アドレス (VPC IPv4 ネットワークの範囲に 2 をプラスしたアドレス) へのクエリは成功する。

## [Q] Elastic IP

あなたはソリューションアーキテクトとして、AWSのコスト削減を検討しています。Cost Explorerを利用してコスト内容を確認すると、無料で利用できるはずのElastic IPアドレスに課金されていることが判明しました。

Elastic IPアドレスに課金されてしまった理由は何でしょうか？

- 1) Elastic IPを解放していないが、Elastic IPがEC2インスタンスにアタッチしていない。
- 2) Elastic IPを解放せずにElastic IPがEC2インスタンスにアタッチしている。
- 3) Elastic IPの無料利用時間を超過している。
- 4) Elastic IPの無料利用数を超過している。

# Elastic IP

Elastic IPは静的に利用できる追加のIPアドレス。インスタンスがインターネットへとアクセスするためには、パブリックIPかElastic IPを利用する。

## パブリックIP

- ✓ 動的なパブリック IPv4 アドレス
- ✓ インスタンスが停止した場合はIPアドレスが変更される。
- ✓ VPCでパブリックIPアドレスの割当が有効化されれば自動的にVPC内リソースに割り当てられる。
- ✓ 無料

## Elastic IP

- ✓ 静的なパブリック IPv4 アドレス
- ✓ インスタンスが停止してもIPアドレスは変更されない。
- ✓ VPCコンソールにおいてElastic IPを作成してから、必要なサービスにアタッチする。
- ✓ 利用時は無料。解放せずに利用しないと有料になる。

## [Q] IPフローディング

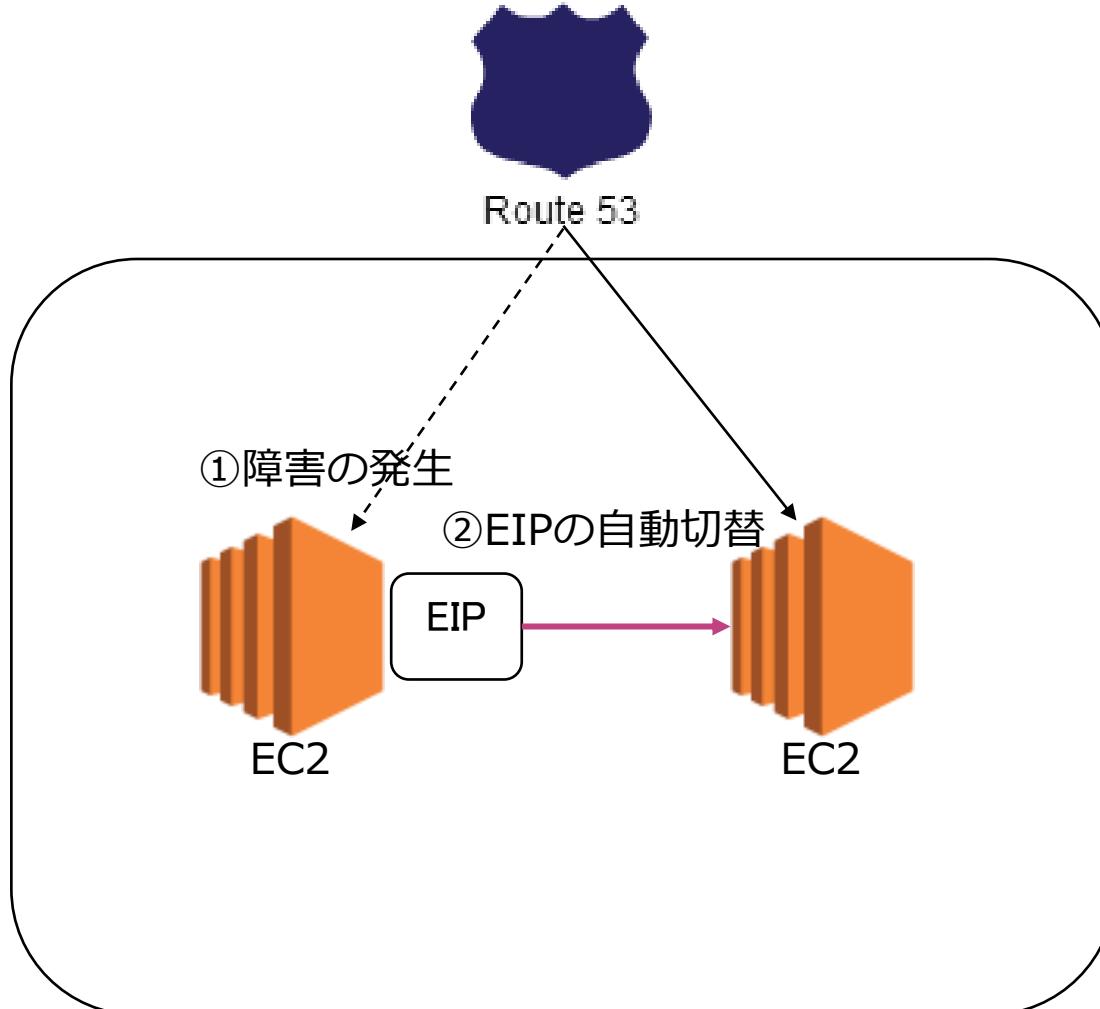
あなたはEC2インスタンスにホストされたアプリケーションを構築しています。このアプリケーションの非機能要件として、EC2インスタンスに障害が発生した場合に別のEC2インスタンスへとトラフィックを変更することで処理を継続させる必要があります。アプリケーションの運用が開始されるとEC2インスタンスに障害が発生し、トラフィックを別インスタンスに切り替えることができましたが、ダウントIMEが発生してしまいます。

この問題を解決するために実施するべき方法を選択してください。

- 1) ENIを利用してIPフローディングを利用する。
- 2) EFAを利用してIPフローディングを利用する。
- 3) ELBを利用してIPフローディングを利用する。
- 4) Elastic IPを利用してIPフローディングを利用する。

# IPフローティング

障害発生時にダウンタイムをなくすため、Elastic IPを自動で付け替える機能



## [Q] ENI

あなたはEC2インスタンスにホストされたアプリケーションを構築しています。このインスタンスにはプライベートIPアドレスとMACアドレスを利用した構成を実施しており、プライマリインスタンスが終了した場合は、ENIをスタンバイセカンダリインスタンスに接することが必要です。これにより、トラフィックフローを数秒以内に再開できます。その際に、EC2インスタンスへのENIアタッチメントで「ウォームアタッチ」を利用します。

ウォームアタッチの正しい説明を選択してください。

- 1) 停止中のインスタンスにENIをアタッチする。
- 2) 起動プロセス中のインスタンスにENIをアタッチする。
- 3) 実行中のインスタンスにENIをアタッチする。
- 4) インスタンスがアイドル状態のときにENIをアタッチする。

# ENI

Elastic Network Interface は、仮想ネットワークカードを表す VPC 内の論理ネットワーキングコンポーネント。インスタンスへのIPアドレスの割り当て時に利用

## 【ENIが保持するネットワークの属性情報】

- ✓ VPC の IPv4 アドレス範囲からのプライマリプライベート IPv4 アドレス
- ✓ VPC の IPv4 アドレス範囲からの 1 つ以上のセカンダリプライベート IPv4 アドレス
- ✓ プライベート IPv4 アドレスごとに 1 つの Elastic IP アドレス (IPv4)
- ✓ 1 つのパブリック IPv4 アドレス
- ✓ 1 つ以上の IPv6 アドレス
- ✓ 1 つ以上のセキュリティグループ
- ✓ MAC アドレス
- ✓ 送信元/送信先チェックフラグ

# ENI

ENIはインスタンスにアタッチして利用する。以下の3つのアタッチ方法がある。

## ホットアタッチ

- ✓ ENI をインスタンスの実行中にアタッチすること

## ウォームアタッチ

- ✓ ENI をインスタンスの停止中にアタッチすること

## コールドアタッチ

- ✓ ENI をインスタンスの起動中にアタッチすること

# セクションの内容

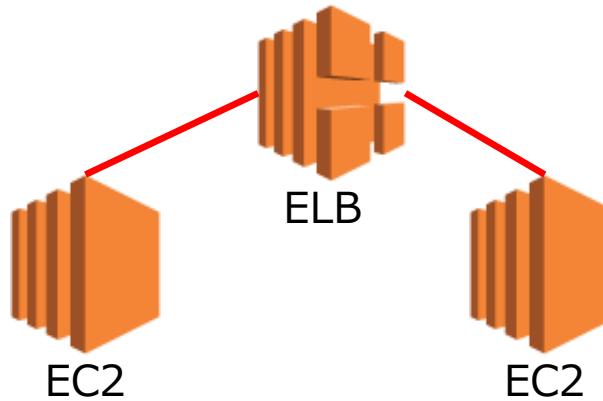
レクチャー	レクチャーで学ぶ内容
Auto Scalingの出題範囲	AWSのアーキテクチャ構成では欠かせないAuto Scalingにおける出題問題を確認して、その範囲の知識を詳細に学習します。
RDSの出題範囲	AWSの代表的なリレーショナルデータベースサービスであるRDSにおける出題問題を確認して、その範囲の知識を詳細に学習します。
EBSの出題範囲	EC2インスタンスと併に利用するストレージであるEBSにおける出題問題を確認して、その範囲の知識を詳細に学習します。
ELBの出題範囲	AWSのアーキテクチャ構成では欠かせないELBにおける出題問題を確認して、その範囲の知識を詳細に学習します。

## Auto Scalingの出題範囲

# Auto Scalingとは何か？

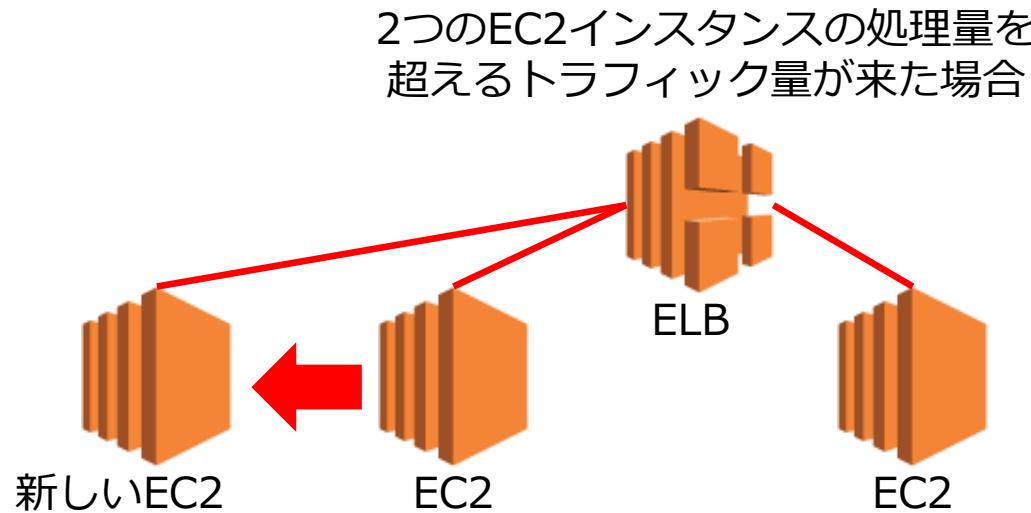
インスタンスへのアクセスが高まったときに、新しいインスタンスを増設して、パフォーマンスを向上させる機能

2つのEC2インスタンスの処理量を  
超えるトラフィック量が来た場合



# Auto Scalingとは何か？

インスタンスへのアクセスが高まったときに、新しいインスタンスを増設して、パフォーマンスを向上させる機能



# スケーリングのタイプ

スケーリングタイプは垂直スケーリングと水平スケーリングの2タイプ。Auto-scalingは水平スケーリング

## 垂直スケーリング

## 水平スケーリング

### 【拡張方法】

スケールアップ：メモリやCPUの追加・増強

### 【拡張方法】

スケールアウト：処理する機器／サーバー台数を増加する

### 【低減方法】

スケールダウン：メモリやCPUの削減・低性能化

### 【低減方法】

スケールイン：処理する機器／サーバー台数を低減する

# Auto Scalingの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

## 起動テンプレートの作成

- ✓ Auto Scalingグループを設定する際に利用するインスタンスの構成内容を決める設定方式が問われる。
- ✓ 起動設定との違いが問われる。

## Auto Scalingの構成

- ✓ シナリオに基づいて、Auto Scalingを利用したアーキテクチャの構成が問われる。

## Auto Scaling構成の設定

- ✓ シナリオに基づいてAuto Scalingを利用した設定上のトラブルや確認が問われる。

## グループサイズの設定

- ✓ Auto Scalingグループを設定する際のグループサイズの設定方法が問われる。

# Auto Scalingの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

スケーリングポリシーの設定	<ul style="list-style-type: none"><li>✓ Auto Scalingグループを設定する際に選択するスケーリングポリシーの設定方法が問われる。</li><li>✓ スケーリングポリシーのタイプの選択方法が問われる。</li></ul>
ヘルスチェック	<ul style="list-style-type: none"><li>✓ Auto Scalingグループ上のヘルスチェック方式の選択と、その効果に関する質問が出題される。</li></ul>
終了ポリシー	<ul style="list-style-type: none"><li>✓ Auto Scalingのスケールイン時にインスタンス削除順序を決定する終了ポリシーの選択方法が問われる。</li><li>✓ デフォルトの削除順序やAZの選択順序が問われる。</li></ul>
クールダウン期間	<ul style="list-style-type: none"><li>✓ スケールイン時に設定できるクールダウン期間の設定方法や用途が問われる。</li></ul>
Auto Scalingの挙動	<ul style="list-style-type: none"><li>✓ Auto Scaling実行時に不均衡が発生した場合や、インスタンスが終了したり異常が発生した場合の挙動が問われる。</li></ul>

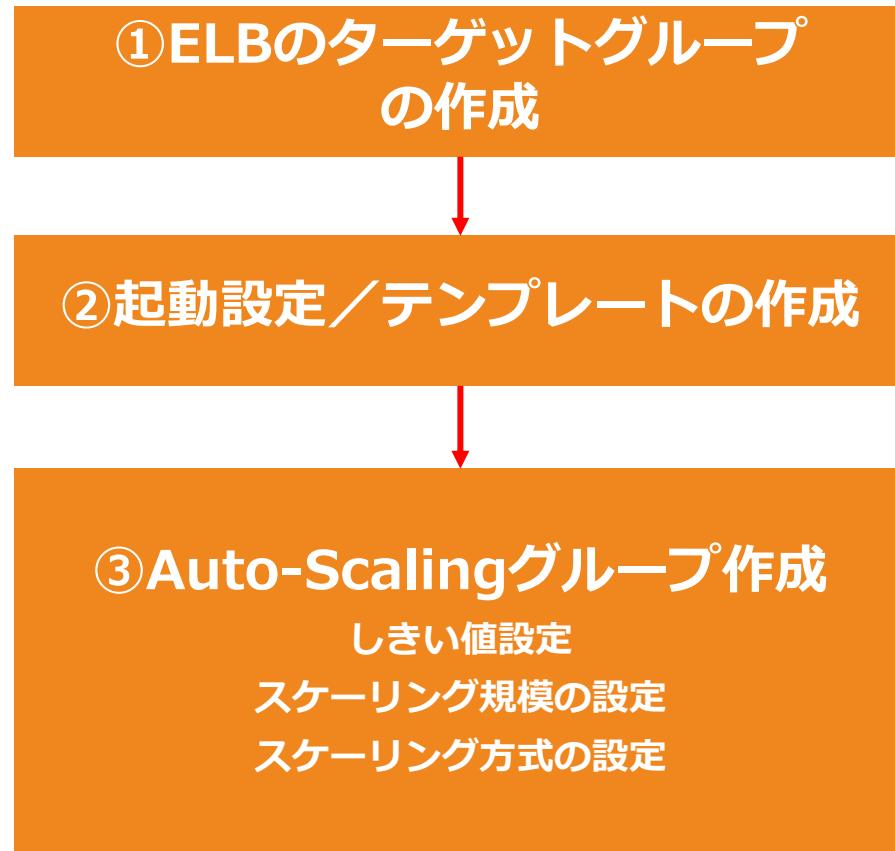
# Auto Scalingの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

ライフサイクルフック	✓ Auto Scalingグループによるインスタンス起動または削除時に実行されるカスタムアクションであるライフサイクルフックの用途や挙動に関する質問が出題される。
トラブルシューティング	✓ Auto Scaling実行時にトラブルが発生した場合の適切なトラブルシューティングの実行方法が問われる。

# Auto-Scalingの設定プロセス

Auto Scaling設定にはELBと起動テンプレートを事前に準備する必要がある



# ELBとの連携

Auto-Scalingで起動するインスタンスをELBのターゲットグループ内に配置することが可能

ロードバランシング - 省略可能 [Info](#)

ロードバランシングの有効化

Application Load Balancer または Network Load Balancer

Classic Load Balancer

ロードバランサーのターゲットグループを選択

ターゲットグループの選択 [▼](#) [C](#)

udemy-elb-target [X](#)

[ターゲットグループを作成する](#)

ヘルスチェック - 省略可能

ヘルスチェックのタイプ [Info](#)

EC2 Auto Scaling は、ヘルスチェックに合格しなかったインスタンスを自動的に置き換えます。ロードバランシングを有効にした場合、常に有効になっている EC2 ヘルスチェックに加えて、ELB ヘルスチェックを有効にすることができます。

EC2  ELB

ヘルスチェックの猶予期間

新しいインスタンスの運用が開始されてから、EC2 Auto Scaling が最初のインスタンスのヘルスチェックを実行するまでの時間です。

300 秒

# Auto-Scalingの要素

起動テンプレートによりインスタンスの設定を準備した後、Auto Scalingグループを設定する。

## 起動テンプレート

- ✓ Auto Scalingによって起動するインスタンスタイプなどの起動設定
- ✓ 起動テンプレートにスケーリングするインスタンスを設定する。
- ✓ 起動テンプレートはインスタンス起動全般で利用可能であり、バージョニングなどの機能が充実している。

## Auto Scaling グループ

- ✓ Auto Scalingのグループサイズ（起動するインスタンス数）の設定
- ✓ 実行時のしきい値の設定
- ✓ スケールリングポリシーを選択して、スケールアウトとスケールインの方法を設定
- ✓ ターミネーションポリシーを設定

# [Q]起動テンプレートの作成

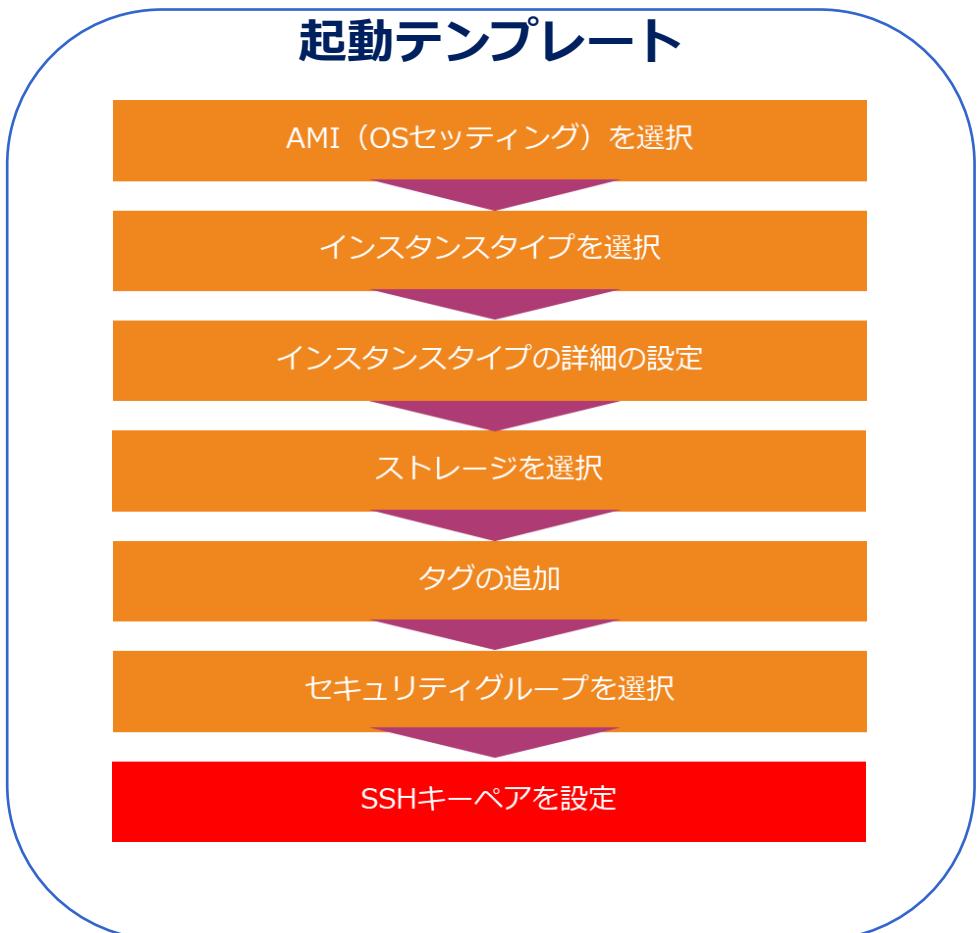
ある会社はWEBアプリケーションをAWS上で構築しています。需要増などによって一時的にアプリケーションの負荷が高まることに備えるためにAuto Scalingの仕組みをEC2インスタンスに導入することになりました。ソリューションアーキテクトはインスタンス構成を適切に管理する仕組みを選択して、Auto Scalingグループを設定することが求められています。

インスタンスを管理するために利用するべき機能はどれでしょうか？

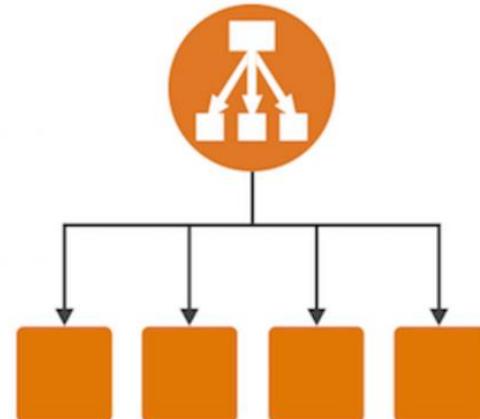
- 1) 起動テンプレートを使用してAuto Scalingグループを作成する
- 2) ゴールデンイメージを使用してAuto Scalingグループを作成する
- 3) スポットフリートリクエストを使用して、Auto Scalingグループを作成する
- 4) 起動設定を使用してAuto Scalingグループを作成する

# 起動テンプレートの作成

起動テンプレートはEC2で説明した起動設定をテンプレート化して、自動で起動する際に利用する仕組み



Auto Scaling



- ✓ 現在は起動設定ではなく起動テンプレートの利用が推奨されている。
- ✓ AMIを更新した場合は作成しなおす必要がある。
- ✓ 起動テンプレートの構成通りにインスタンスを選択して起動する。
- ✓ 起動設定はAuto Scaling専用
- ✓ 起動テンプレートは広くE2インスタンスの起動時に利用

# [Q] Auto Scalingの構成

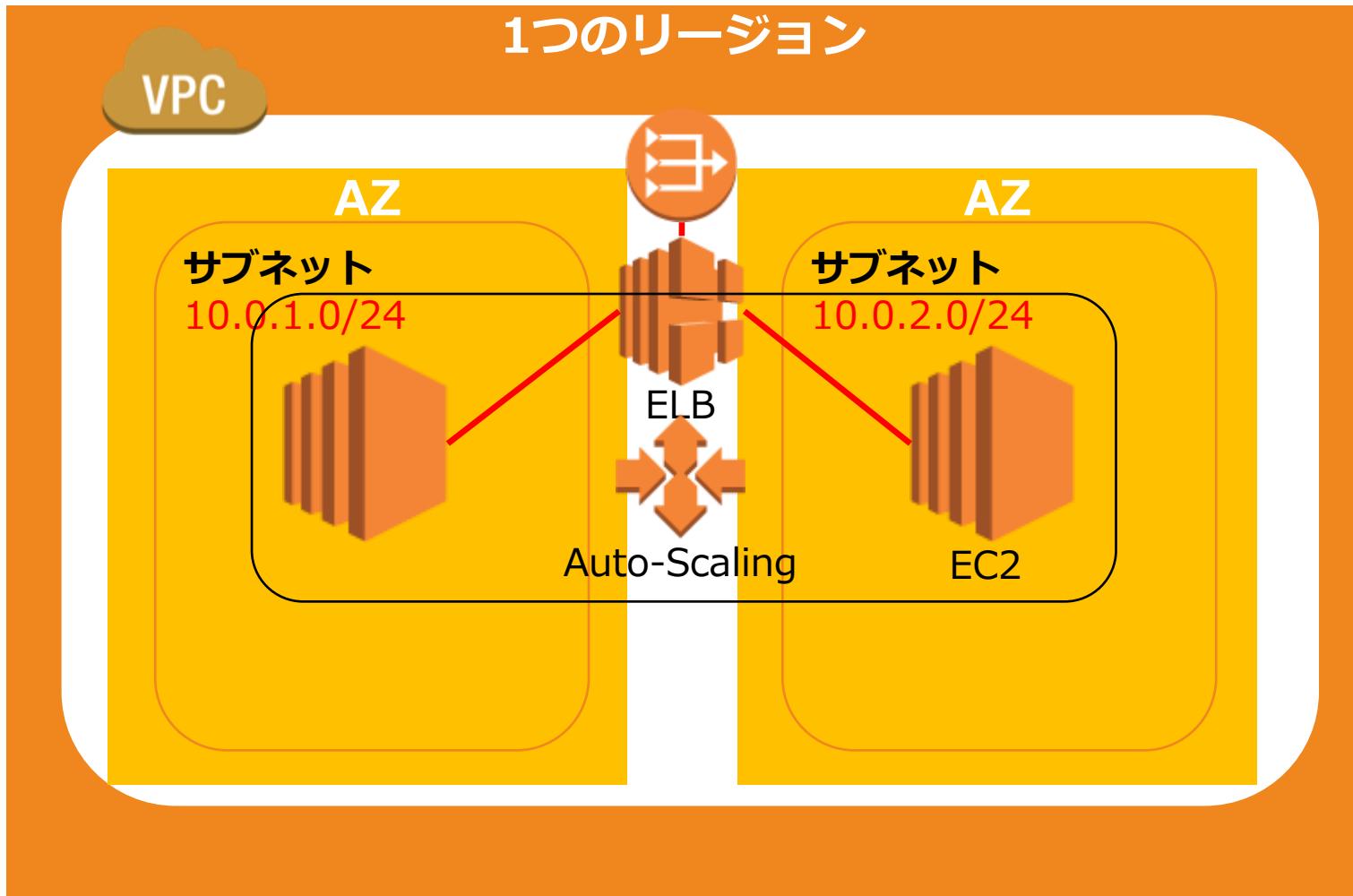
B社ではAWS上にWEBアプリケーションを構築して、コンテンツを配信する仕組みを構築しています。データ層では、オンライントランザクション処理（OLTP）データベースを利用しています。WEB層では柔軟でスケーラブルなアーキテクチャ構成を実現する必要があり、負荷分散や一時的な負荷に対する対策が必要不可欠です。

この要件を満たすための最適な方法を選択してください。

- 1) RDSのマルチAZ構成を構成する。
- 2) EC2インスタンスに対してAuto ScalingとELBを設定する。
- 3) EC2インスタンスをマルチAZに展開してRoute53によるフェイルオーバーティングを実施する。
- 4) EC2インスタンスを予測キャパシティよりも多く設置する。

# 基本アーキテクチャー

ELB構成で冗長化した上でAuto-Scalingを設定して自動拡張できるようにする



# [新Q] Auto Scaling構成の設定

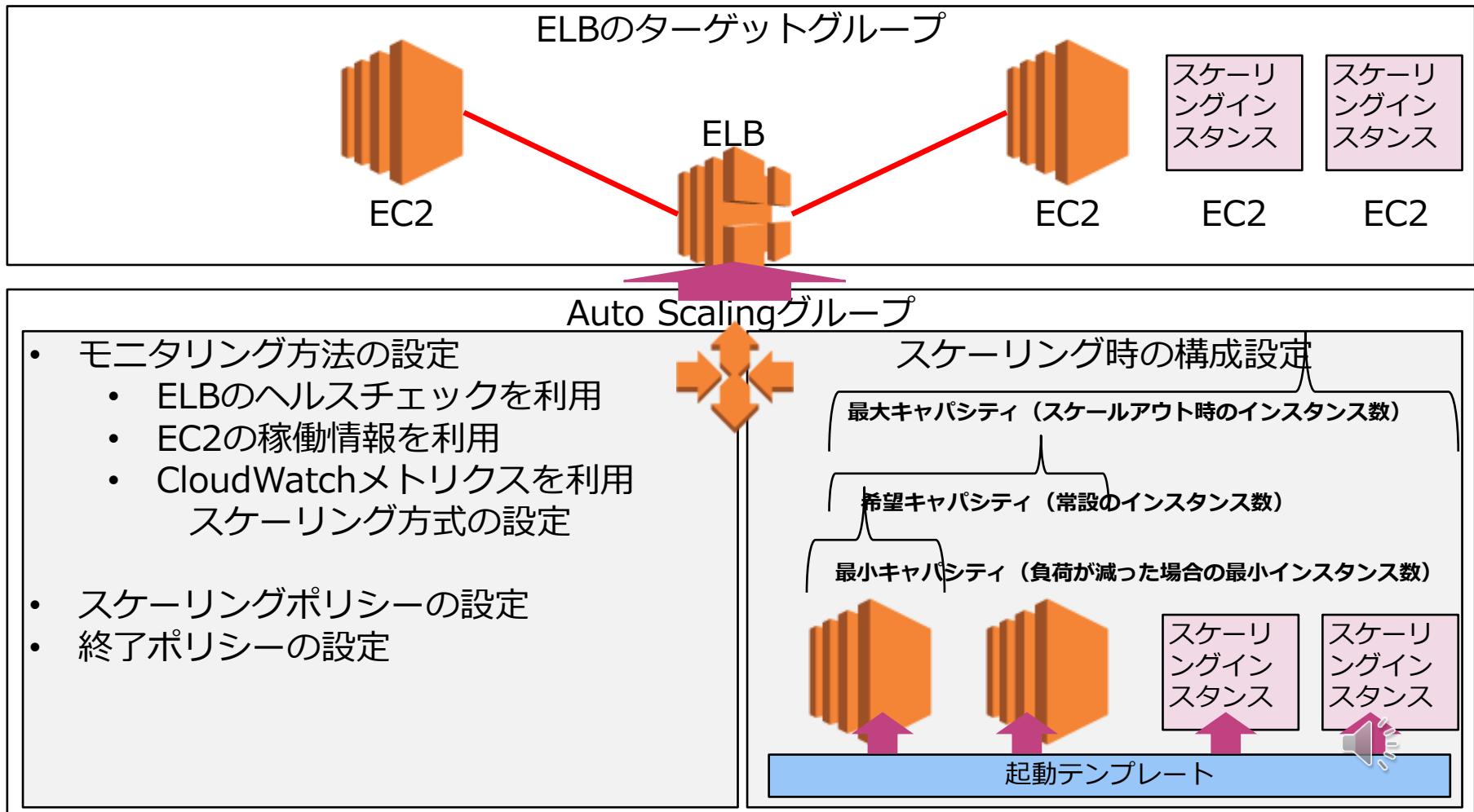
ソリューションアーキテクトはAWSを利用した新規アプリケーションのアーキテクチャ構成を設計しています。このアプリケーションでは処理されるジョブの負荷に応じてアプリケーションインスタンスを増減することが必要です。このジョブはステートレス処理であり、かつ疎結合に構成される必要があります。

こ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) Amazon SNSトピックを作成して、EC2インスタンスにジョブをメッセージとして送信するように構成する。このインスタンスのAMIを使用した起動テンプレートを作成する。この起動テンプレートを使用して、CPU使用率に基づいてインスタンス数を増減するスケーリングポリシーを設定したAuto Scalingグループを構成する。
- 2) Amazon SNSトピックを作成して、EC2インスタンスにジョブをメッセージとして送信するように構成する。このインスタンスのAMIを使用した起動設定を作成する。この起動設定を使用して、CPU使用率に基づいてインスタンス数を増減するスケーリングポリシーを設定したAuto Scalingグループを構成する。
- 3) Amazon SQSキューを作成して、EC2インスタンスにジョブをメッセージとして送信するように構成する。このインスタンスのAMIを使用した起動テンプレートを作成する。この起動テンプレートを使用して、CPU使用率に基づいてインスタンス数を増減するスケーリングポリシーを設定したAuto Scalingグループを構成する。
- 4) Amazon SQSキューを作成して、EC2インスタンスにジョブをメッセージとして送信するように構成する。このインスタンスのAMIを使用した起動設定を作成する。この起動設定を使用して、CPU使用率に基づいてインスタンス数を増減するスケーリングポリシーを設定したAuto Scalingグループを構成する。

# Auto Scalingグループの設定

スケーリング対象やスケーリング方法を示したAuto Scalingグループを設定する。



# [Q] グループサイズの設定

あなたはWebアプリケーションをAWS上に構築しています。このWEBアプリケーションは単一のEC2インスタンスで構成されています。冗長化を達成するために複数のインスタンスを利用して、1つのAZに通常は1台ずつインスタンスを起動し、負荷の上昇とともに4台のインスタンスで処理を継続する予定です。

この要件を満たすことができる最も費用対効果の高いスケーリング方法はどれですか？

- 1) min = 2、max = 4、desired = 2として、2つのAZにまたがる自動スケーリンググループを作成する。
- 2) min = 2、max = 4、desired = 4として、2つのAZにまたがる自動スケーリンググループを作成する。
- 3) min = 1、max = 4、desired = 2として、2つのAZにまたがる自動スケーリンググループを作成する。
- 4) min = 1、max = 2、desired = 1として、2つのAZにまたがる自動スケーリンググループを作成する。

# グループサイズの設定

グループサイズの設定において、インスタンスの増減値を設定することができる。

## 希望するキャパシティ

- ✓ Auto Scaling が実行されない状態でのインスタンス数を設定する。
- ✓ この数値を変更することで、手動でスケーリングさせることも可能

## 最小キャパシティ

- ✓ スケールイン時にインスタンスを削減する際の下限のインスタンス数を設定する。
- ✓ 希望する容量より大きい数値は設定できない。

## 最大キャパシティ

- ✓ 最大キャパシティはスケールアウト時に起動するインスタンスの最大数を設定する。
- ✓ 希望する容量より少ない数値は設定できない。

# [Q]スケーリングポリシーの設定

あなたは2層WebアプリケーションをAWS上に実装しました。このアプリケーションは、Amazon EC2インスタンスとAmazon ELBによりマルチAZ構成となっています。さらにAuto Scalingを追加して、EC2インスタンスを自動的に追加し、着信リクエストの一時的な負荷増加に対応する設定が必要です。EC2インスタンスは60%のCPU使用率までが正常に処理を実行できますが、それ以上の使用率になるとパフォーマンスが低下するようです。

どのようなスケーリングポリシーを設定するべきでしょうか？

- 1) Auto Scalingグループで平均合計CPU使用率を60%をしきい値としたターゲット追跡スケーリングポリシーを設定する。
- 2) Auto Scalingグループで平均合計CPU使用率を60%をしきい値としたステップスケーリングポリシーを設定する。
- 3) Auto Scalingグループで平均合計CPU使用率を60%をしきい値としたスケジュールドスケーリングポリシーを設定する。
- 4) Auto Scalingグループで平均合計CPU使用率を60%をしきい値とした手動スケーリングポリシーを設定する。

# スケーリングポリシーの設定

スケーリングポリシーを設定して、スケーリングを実施する。

動的 スケーリング	簡易 スケーリング ポリシー	ターゲット追跡スケーリングポリシーの通常の設定 アラーム設定に基づいて1段階のスケーリングを実施
	ステップ スケーリング ポリシー	アラーム超過のサイズに基づいてインスタンス数を動的にスケーリングする1つ以上のステップ調整値を指定して複数回の段階的なスケーリングを実施
	手動スケーリング	希望する容量を調整して、手動でスケーリングを実施する。
スケジュールされた スケーリング		スケーリングを実施する日時を指定して、スケーリングを実行する。

# ターゲット追跡スケーリングポリシー

CloudWatchのモニタリングメトリクスを利用したスケーリングを実施する。

スケーリングポリシー - 省略可能

需要の変化に対応するために、スケーリングポリシーを使用して Auto Scaling グループのサイズを動的に変更するかどうかを選択します。 [Info](#)

ターゲット追跡スケーリングポリシー  
希望する結果を選択して、スケーリングポリシーが結果を達成するために必要に応じてキャパシティを追加および削除するようにします。

なし

スケーリングポリシー名

メトリクスタイプ

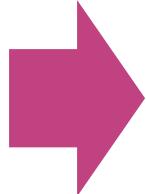
ターゲット値

インスタンスには以下のものが必要です  
 メトリクスに含める前にウォームアップする秒数

スケールインを無効にしてスケールアウトポリシーのみを作成する

# 複数のスケーリングポリシーの設定

スケーリングポリシーの設定は複数組み合わせて利用することができる



# [Q]ヘルスチェックの利用

現在WEBアプリケーションはAWS上で実行されています。このWEBアプリケーションはELBの背後にあるAmazonEC2インスタンスにAuto Scalingグループが構成されています。本日、1つのEC2インスタンスに異常が発生し、ELBがそれをターゲットからはずしましたが、インスタンスがまだ実行中となっています。

このような挙動の最も可能性が高い原因はどれでしょうか？

- 1) ELBヘルスチェックタイプがAuto Scalingで利用されていない。
- 2) EC2ヘルスチェックタイプがAuto Scalingで利用されていない。
- 3) Auto Scalingグループにクールダウン期間が設定されている。
- 4) Auto Scalingグループにタイムアウト猶予時間が設定されている。

# ヘルスチェックの利用

Auto-Scaling配下のEC2のヘルスチェックにはEC2のステータス情報またはELBのヘルスチェックのどちらかを利用する

## EC2ステータス

インスタンスのステータスがrunning以外の状態異常を判断する。  
EC2が稼働時の異常には対応できない。

## ELB

ELBのヘルスチェック機能を活用する  
ヘルスチェックのメトリクスをCloudWatchに連携させて動的スケーリングを設定できる。

# ヘルスチェック

ELBのヘルスチェックやCloudWatchのアラート機能をトリガーとして利用できる



# [Q]終了（ターミネーション）ポリシー

現在WEBアプリケーションはAWS上で実行されています。このWEBアプリケーションはELBの背後にあるAmazonEC2インスタンスにAuto Scalingグループが構成されています。負荷が高まるとAuto Scalingグループによって新規インスタンスが2つのアベイラビリティーゾーン（AZ）にまたがって生成されます。スケーリングが実行されると、ap-northeast-1aには3つのEC2インスタンスが配置され、ap-northeast-1cには4つのEC2インスタンスが配置されています。

スケーリング時に、どのようにインスタンスが削除されますか？

- 1) 最も古い起動構成のインスタンスがap-northeast-1cで終了する。
- 2) 最も古い起動構成のインスタンスがap-northeast-1aで終了する。
- 3) ap-northeast-1aでランダムにインスタンスを終了する。
- 4) ap-northeast-1cでランダムにインスタンスを終了する。
- 5) 均衡を保つためにap-northeast-1aのインスタンスが1つ作成される。

# 終了（ターミネーション）ポリシー

需要減に基づくスケールインの際に、どのインスタンスから終了するかを設定

デフォルト	AZの選択	<ul style="list-style-type: none"><li>✓ 複数AZにインスタンスがあるか確認し、一番多いインスタンスが配置されているAZのインスタンスを削除する。</li><li>✓ 全AZで同数のインスタンスが配置されている場合は、ランダムでAZを選択してインスタンスを削除する。</li></ul>
	インスタンスの選択	<ul style="list-style-type: none"><li>✓ 起動設定/テンプレートが一番古いインスタンスを削除</li><li>✓ 古いインスタンスが複数ある場合は、次の課金発生が短いインスタンスを削除する。</li><li>✓ 次の課金時間に近いインスタンスが複数ある場合は、ランダムで削除する。</li></ul>
カスタム	AZの選択	<ul style="list-style-type: none"><li>✓ 複数AZにインスタンスがあるか確認し、一番多いインスタンスが配置されているAZのインスタンスを削除する。</li><li>✓ 全AZで同数のインスタンスが配置されている場合は、ランダムでAZを選択してインスタンスを削除する。</li></ul>
	インスタンスの選択	<ul style="list-style-type: none"><li>✓ 選択したAZのカスタムポリシーに従い削除する。</li></ul>

# 終了（ターミネーション）ポリシー

需要減に基づくスケールインの際にどのインスタンスから終了するかを設定

**OldestInstance**

最も古いインスタンスから順番に終了

**NewestInstance**

最も新しい起動時刻のインスタンスから終了

**OldestLaunch Configuration**

最も古い起動設定により起動しているインスタンスから終了

**ClosestTo NextInstanceHour**

次の課金が始まるタイミングが最も近いインスタンスから終了

# [Q]クールダウン期間

現在WEBアプリケーションはAWS上で実行されています。このWEBアプリケーションはELBの背後にあるAmazonEC2インスタンスにAuto Scalingグループが構成されています。最近、Auto Scalingが同じ時間にインスタンスを増やしたり、削除したりと短期間に実行しており、スケーリングイベントの発生が多くなっています。

このようなスケーリング状況を改善するために、何をするべきでしょうか？（3つ選択してください）

- 1) Auto Scalingグループサイズを変更して、希望する容量を増加させる。
- 2) スケジュールされたスケーリングアクションを使用してスケーリングを設定する。
- 3) Auto ScalingスケールダウンポリシーをトリガーするCloudWatchアラーム期間を変更する。
- 4) Auto ScalingスケールダウンポリシーをトリガーするCloudWatchアラームのしきい値を変更する。
- 5) Auto Scalingグループのクールダウン期間を変更する。

# クールダウン期間

スケールイン時のインスタンスの終了においてクールダウン時間を設定することが可能

## クールダウン期間

- ✓ 終了するインスタンスの前のアクティビティの影響前に、Auto Scaling グループが追加のインスタンスを起動または終了するのを防ぐ
- ✓ クールダウン期間はデフォルトで設定されている
- ✓ クールダウン期間は変更可能

## クールダウン期間 の例外

- ✓ クールダウン期間中、スケジュールされたアクションがスケジュールされた時間に開始されるか、ターゲット追跡またはステップスケーリングポリシーによりスケーリングアクティビティが開始されると、クールダウン期間が終了するのを待たずに、それらのスケーリングアクティビティを実行する。
- ✓ インスタンスが正常でなくなった場合、Amazon EC2 Auto Scaling はクールダウン期間の完了を待つことなく、異常のあるインスタンスを置き換る。

# [Q] Auto Scalingの挙動

現在WEBアプリケーションはAWS上で実行されています。このWEBアプリケーションはELBの背後にあるAmazonEC2インスタンスにAuto Scalingグループが構成されています。このAuto Scaling Groupは2つのAZを使用しており、現在、グループ内で6つのAmazonEC2インスタンスが実行されています。

EC2インスタンスの1つの不具合が発生した場合に、Auto Scalingはどのようなアクションを実行しますか？（2つ選択してください。）

- 1) 不均衡を是正するため、3つのEC2インスタンスが実行されているAZ内のインスタンスを終了する。
- 2) 障害が発生したインスタンスがあるAZ内において新規に1つインスタンスを起動する。
- 3) 障害が発生したインスタンスがないAZ内において新規に1つインスタンスを起動する。
- 4) 最初に障害が発生したインスタンスを削除してから、同じAZ内に新しいインスタンスを起動する。
- 5) 新しいインスタンスを起動した後に、障害が発生したインスタンスを終了する。

# Auto Scalingの挙動

Auto Scalingが実行されると、最適なAZに適切に分散されるようにインスタンス数が調整される。

## 基本的な挙動

- ✓ インスタンスが最も少ないAZでインスタンスを起動する
- ✓ インスタンス起動が失敗した場合は、起動が成功するまで別AZで起動する

## AZ間にアンバランスが発生した場合の挙動

### 再分散の実施

- ✓ AZ間でインスタンス数の不均衡があると数を調整する。
- ✓ グループが不均等になった原因のインスタンスを停止して、リソースが不足していたAZに新規インスタンスを起動する。

### 再分散時の挙動

- ✓ 古いインスタンスを終了する前に新しいインスタンスを起動することで、パフォーマンス低下を防ぐ。
- ✓ Auto Scalingの最大容量に近づくと、再分散処理が遅くなったり、完全に停止する可能性がある。これを回避するために、一時的に最大容量を増やす（最大容量の10%または+1の容量を追加する。）

## [Q]ライフサイクルフック

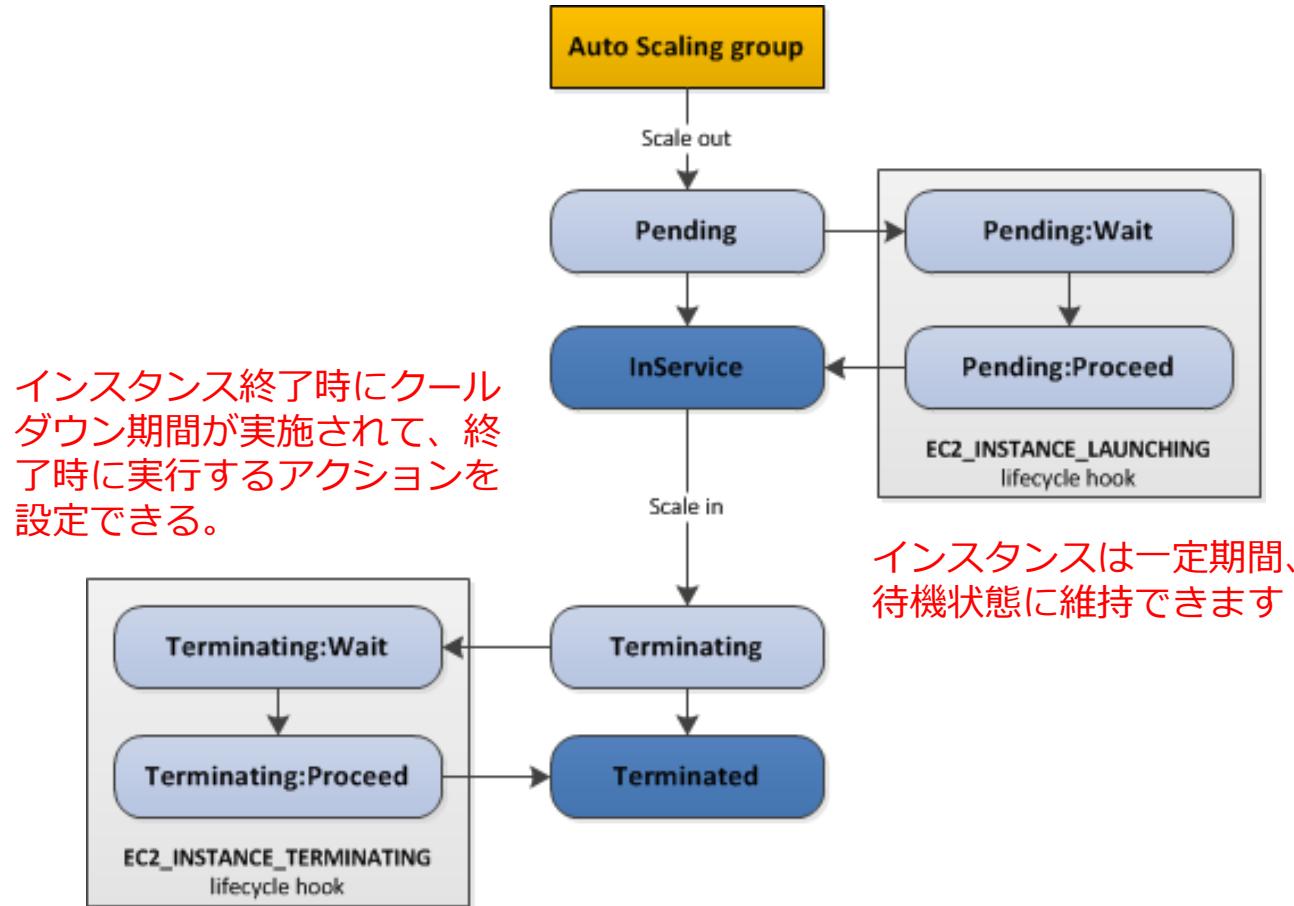
現在WEBアプリケーションはAWS上で実行されています。このWEBアプリケーションはELBの背後にあるAmazonEC2インスタンスにAuto Scalingグループが構成されています。スケールインを実行する際に、インスタンス停止の影響を調べるために、停止されるインスタンスのログファイルをダウンロードできるようにしたいと考えています。

このカスタムアクションを有効にするために使用できる機能は次のうちどれですか？

- 1) Auto ScalingグループのEC2フリート構成
- 2) Auto Scalingグループの終了ポリシー
- 3) Auto Scalingグループのスケジュールされたスケーリングポリシー
- 4) Auto Scalingグループのライフサイクルフック

# ライフサイクルフック

スケーリングされたインスタンスの起動時または削除時に、そのインスタンスを一時停止してカスタムアクションを実行する。Lambdaと連携した処理も可能



# [Q] トラブルシューティング

あなたの会社では、EC2インスタンスにELBを設定してトラフィック分散した上で、Auto Scalingグループを設定しました。負荷が向上した際の挙動を確かめるために、負荷テストツールを利用してAuto Scalingグループを実行させます。しかしながら、Auto Scalingによって起動された EC2インスタンスのステータスチェックにおいて、Impairedと表示されているようです。

Auto Scalingはどのようなアクションを実行しますか？

- 1) インスタンスが回復するまで数分待機し、回復しない場合はインスタンスを終了してから、別のインスタンスへと置換する。
- 2) 即時にインスタンスを終了してから、別のインスタンスへと置換する。
- 3) ELBが別のインスタンスへとターゲットを切り替える。
- 4) Auto Scalingは障害が発生していないAZ間でのリバランスを実行する。

# トラブルシューティング

インスタンスのメンテナンスや調査時にはAuto Scalingを一時中斷して、対応することが必要

## インスタンスの起動失敗

- ✓ Auto Scalingはインスタンスの起動を繰り返し実施し、24時間失敗し続けるとAmazon側で停止される可能性がある。

## インスタンスの障害

- ✓ インスタンスの状態が“Impaired”となると、数分間リカバリーされるかチェックする
- ✓ リカバリーされない場合は新しいインスタンスを起動して、Impairedのインスタンスを終了する。

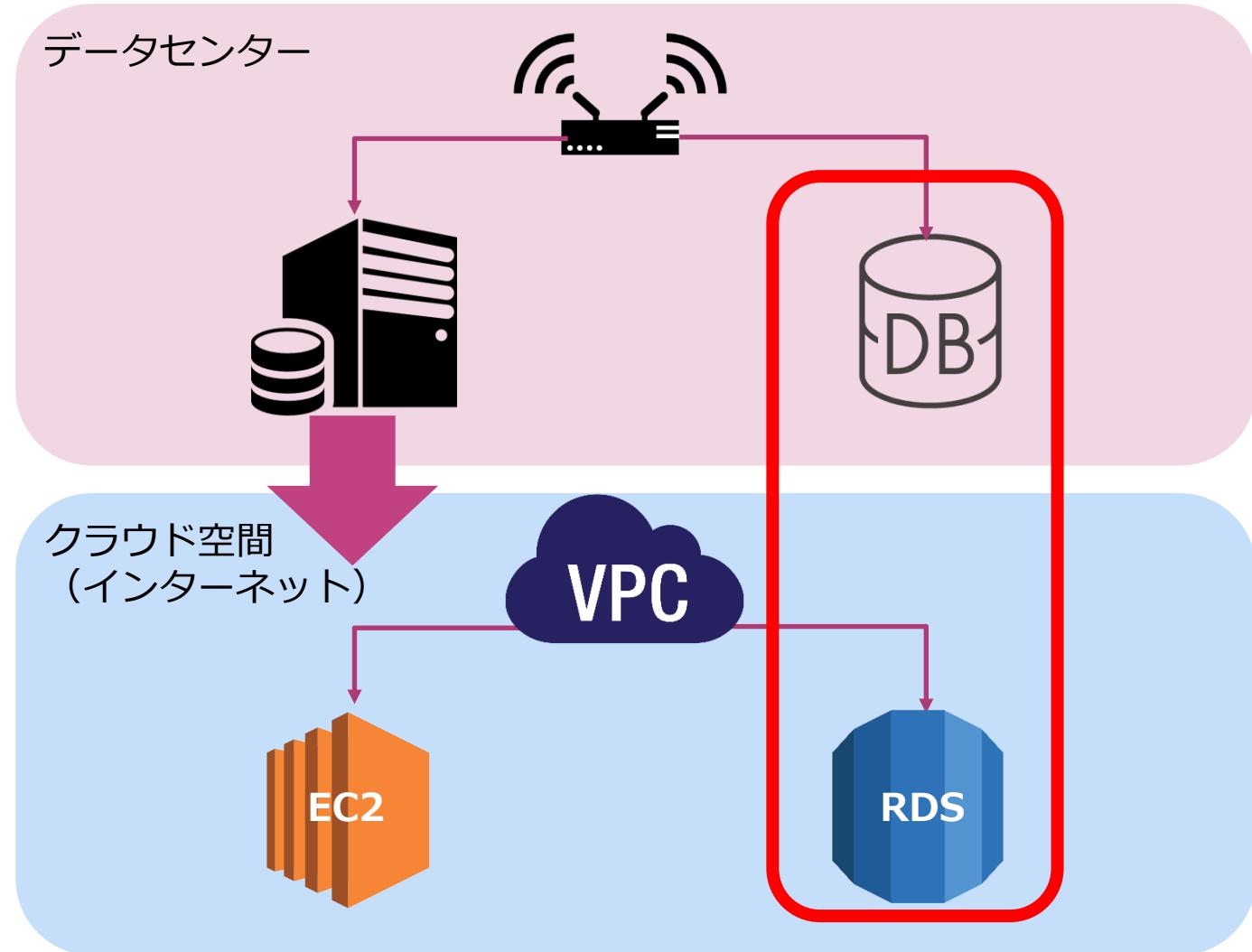
## トラブルシューティングのステップ

- ✓ Auto Scalingグループを一時的に停止しないでインスタンスを停止すると新規インスタンスが起動してしまう。
- ✓ Auto Scalingを停止して、調査・復旧し、Auto Scalingを再開することが基本的な実施方法

## RDSの出題範囲

# RDSとは何か？

RDSはリレーショナルデータベースをクラウド上で即時に起動して、利用することができるサービス



# RDSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

RDSの選択	✓ AWSが提供するデータベースサービスの中で、最適なデータベースとしてRDSを選択する質問
RDSの特徴	✓ RDSの特徴やMySQLやPostgreSQLなどのデータベースエンジンに関する特徴が出題される。
ストレージタイプの選択	✓ DBインスタンスを利用するストレージタイプの特徴とユースケースに関する質問が出題される。
パブリックアクセス構成	✓ RDSのDBインスタンスに対してインターネットから直接アクセスする構成が質問される。
リードレプリカ	✓ RDSを利用したスケーリング方式であるリードレプリカの特徴が出題される。 ✓ Auroraとの違いが出題される。

# RDSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

クロスリージョン レプリカ	✓ RDSを利用してクロスリージョンでリードレプリカを構成する際のユースケースが出題される。
RDSのスケーリング	✓ RDSのスケーリング方法としてスケールアップの方式とスケールアウトの方式が問われる。 ✓ コスト最適やパフォーマンス向上などの要件に応じて最適なスケーリング方式を選択する質問が出題される。
RDSの暗号化	✓ RDSの暗号化の実施方法が問われる。 ✓ 暗号化されていないRDSのDBインスタンスを途中から暗号化する方法が問われる。
メンテナンス	✓ RDSのメンテナンス方法が問われる。 ✓ RDSのメンテナンスウィンドウの設定方法や、その影響が問われる。
バックアップ	✓ RDSのバックアップ方法とその復元方法が問われる。

# [Q]RDSの選択

B社はAWSを利用したデータベースを構築するための要件を確認しています。あなたはソリューションアーキテクトとして、データベース要件から最適なAWSサービスを選択することになりました。この会社ではデータベース環境を自社内で管理することが要件となっています。

この要件を満たすデータベース構築方法を選択してください。

- 1) EC2
- 2) RDS
- 3) Aurora
- 4) DynamoDB

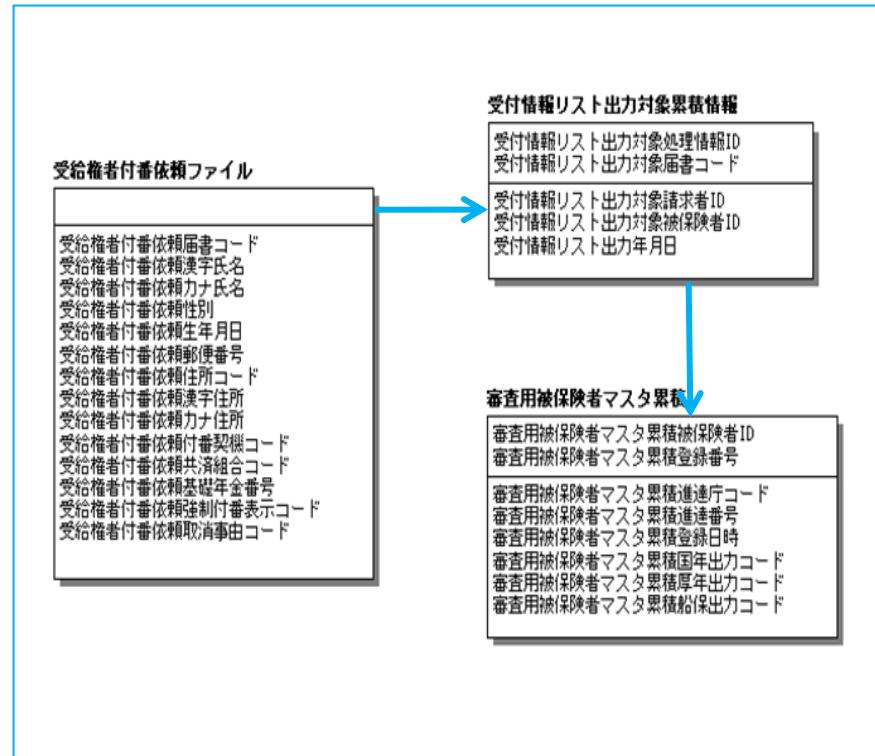
# データモデル

データベースには様々なデータモデルが存在し、利用目的に応じて使い分ける

- リレーションナルモデル
- グラフモデル
- キーバリューストア
- オブジェクト
- ドキュメント
- ワイドカラム
- 階層型

# リレーションナルモデル

データベースはリレーションナルモデルが基本的なデータモデルとなっている。



# トランザクション：ACID

ACIDは信頼性のあるトランザクションシステムの持つべき性質のこと

- Atomicity (原子性)

トランザクションが「すべて実行される」か「一つも実行されない」のどちらかの状態になるという性質

- Consistency (整合性)

トランザクションの前後でデータの整合性が保たれ、矛盾の無い状態が継続される性質

- Isolation (独立性)

トランザクション実行中の処理過程が外部から隠蔽され、他の処理などに影響を与えない性質

- Durability (耐久性)

トランザクションが完了したら、その結果は記録され、クラッシュしても失われることがないという性質

## [Q] RDSの特徴

あなたはソリューションアーキテクトとして、AWS上にデータベースを構築しています。現在、オンプレミスではMySQLを利用しているため、RDSのMySQLを利用すれば容易に移行ができると判断しました。RDSを利用する場合はその特徴を踏まえて移行する必要があります。

次のうちでRDSに推奨されていない方法を選択してください。

- 1) 自動バックアップを有効化する
- 2) MySQLのストレージエンジンとしてMyISAMを利用する。
- 3) MySQLのストレージエンジンとしてInnoDBを利用する。
- 4) 大きなテーブルのパーティションは16TBを超えないようにする。

# RDSの特徴

RDSは様々なデータベースソフトウェアに対応したフルマネージドなリレーショナルデータベース

以下のような標準ソフトウェアを利用したデータベースを構築できる

- MySQL
- ORACLE
- Microsoft SQL Server
- PostgreSQL
- MariaDB
- Amazon Aurora

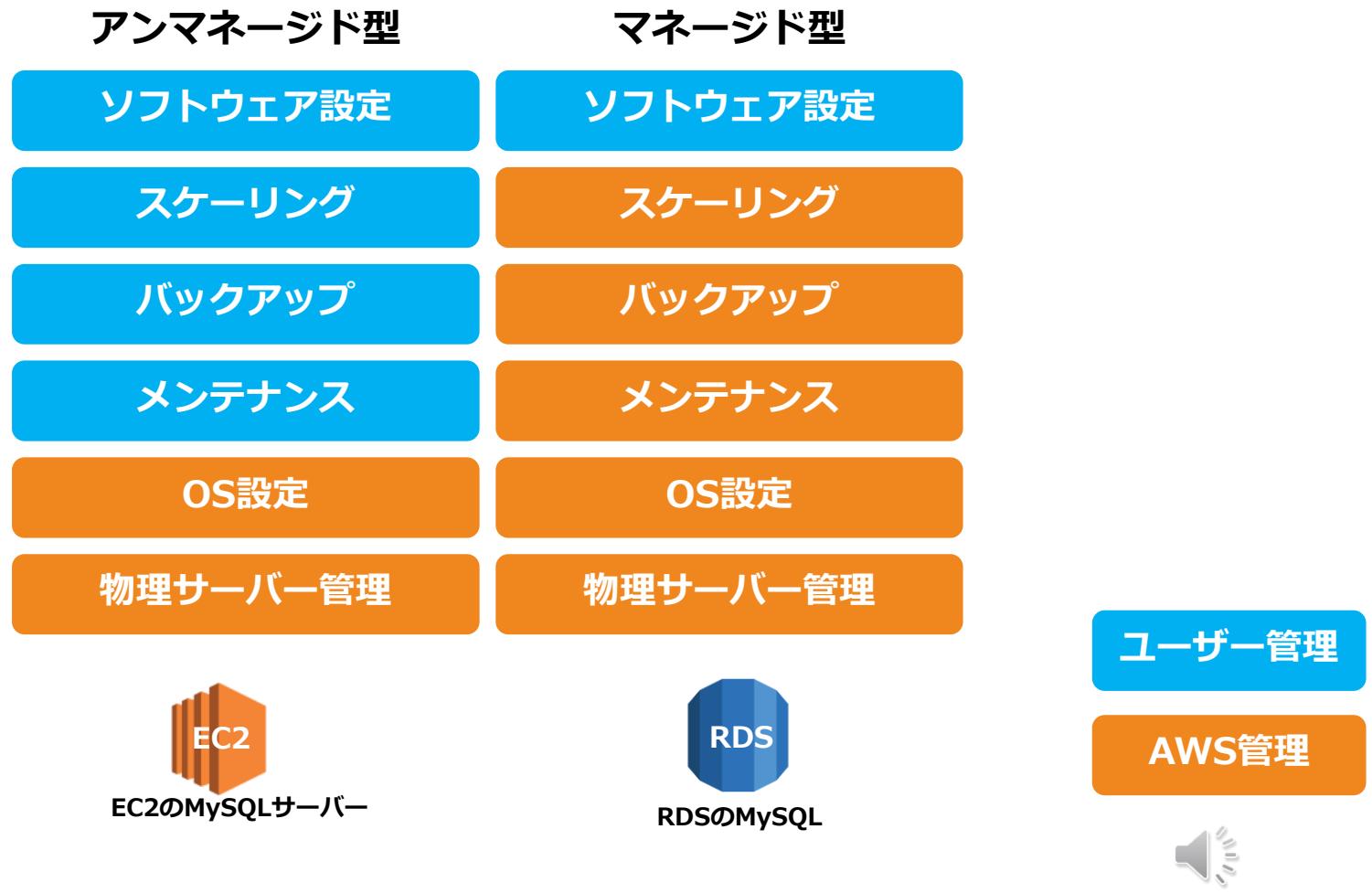
# RDSのベストプラクティス

RDSは主だったベストプラクティスとして以下のような内容が推奨されている。

- メモリ内に保持できるように十分な RAM を割り当てる
- 拡張モニタリングを使用したオペレーティングシステムの問題の特定
- 特定のメトリクスしきい値に対して Amazon CloudWatch アラームを設定
- MySQLのストレージエンジンにはInnoDBを利用する。
- 大きなテーブルのパーティションは16TBを超えないようにする。

# マネージド型DBの特徴

RDSはマネージド型であるため、クラウド側が多くの管理タスクを実施している。



# マネージド型DBの特徴

RDSはマネージド型であり管理が楽であるものの、AWSから提供される機能範囲しかユーザーは利用できない。

## RDSの主な制限事項

- ・ バージョンが限定される
- ・ キャパシティに上限がある
- ・ OSへのログインができない
- ・ ファイルシステムへのアクセスができない
- ・ 一部の機能が使えない
- ・ 個別パッチは適用できない

# [Q]ストレージタイプの選択

あなたはMySQLデータベースを利用した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では多数のトランザクション処理が発生することが予想されており、ランダムI/O遅延が発生することが懸念されています。あなたはソリューションアーキテクトとして、データベース設定によって性能を向上させるように依頼を受けました。その際は、運用負荷をかけないで実行することが求められます。

運用負荷や時間をかけずに実施できる最適なソリューションはどれでしょうか？

- 1) Amazon RDS for MySQLにElastiCacheのキャッシュレイヤーを連携させて、キャッシュ処理により高速処理を実現する。
- 2) EC2インスタンスにプロビジョンドIOPSを設定して、MySQLをインストールして利用する。
- 3) Amazon RDS for MySQLを利用してストレージタイプをプロビジョンドIOPSに変更する
- 4) Amazon RDS for MySQLを利用してインスタンスタイプを最適なものに変更する。

# ストレージタイプの選択

ストレージタイプは汎用とプロビジョンドIOPSから選択する。  
マグネティックは古いタイプであり、あまり利用しない

汎用	<ul style="list-style-type: none"><li>✓ SSDタイプ</li><li>✓ GBあたりの容量課金を実施</li><li>✓ 通常のパフォーマンスに加えてバーストを実施し、100～10,000IOPSを実現可能（サイズによって変わる）</li></ul>
プロビジョンドIOPS	<ul style="list-style-type: none"><li>✓ SSDタイプ</li><li>✓ GBあたりの容量課金を実施+プロビジョンド済みIOPS単位の課金</li><li>✓ 通常のパフォーマンスに加えてバーストを実施し、1,000～30,000IOPSを実現可能（サイズによって変わる）</li></ul>
マグネットイック	<ul style="list-style-type: none"><li>✓ ハードディスクタイプ</li><li>✓ GBあたりの容量課金を実施+IOリクエスト課金</li><li>✓ 平均100～最大数百のIOPS</li></ul>

# [Q] パブリックアクセス構成

顧客管理部門ではCRM用にAWS上にデータベースソリューションを運用しています。部門では機能追加に伴って、Amazon RDS MySQLを利用して新規にデータベースを構築する予定です。非機能要件に対応するため、このデータベースはインターネットから直接にアクセスできるようにすることが必要です。

インターネット経由でRDS MySQLに接続する正しい設定方法はどれでしょうか？  
(3つ選択してください)

- 1) RDSインスタンスのパブリックアクセスを有効化する。
- 2) RDSインスタンスをパブリックサブネットに配置する。
- 3) RDSインスタンスをプライベートサブネットに配置する。
- 4) インターネットからRDSインスタンスへのアクセスを許可するセキュリティグループを作成し、RDSインスタンスに割り当てる。
- 5) NATゲートウェイを構成して、RDSデータベースがあるサブネットにルートを設定する。
- 6) RDSインスタンスにおいてインターネットアクセスを有効化する。

# パブリックアクセス構成

パブリックアクセスを有効化して、セキュリティグループでアクセスを許可する必要がある。

The screenshot shows the AWS RDS console interface for configuring public access. It includes sections for Security Groups, Authentication, and Public Accessibility.

**セキュリティグループ**  
この DB インスタンスに関連付ける DB セキュリティグループの一覧。

セキュリティグループの選択 ▾

default (sg-a418d7d8) (vpc-940724f3) X

**認証機関**  
この DB インスタンスの認証機関。

rds-ca-2019 ▾

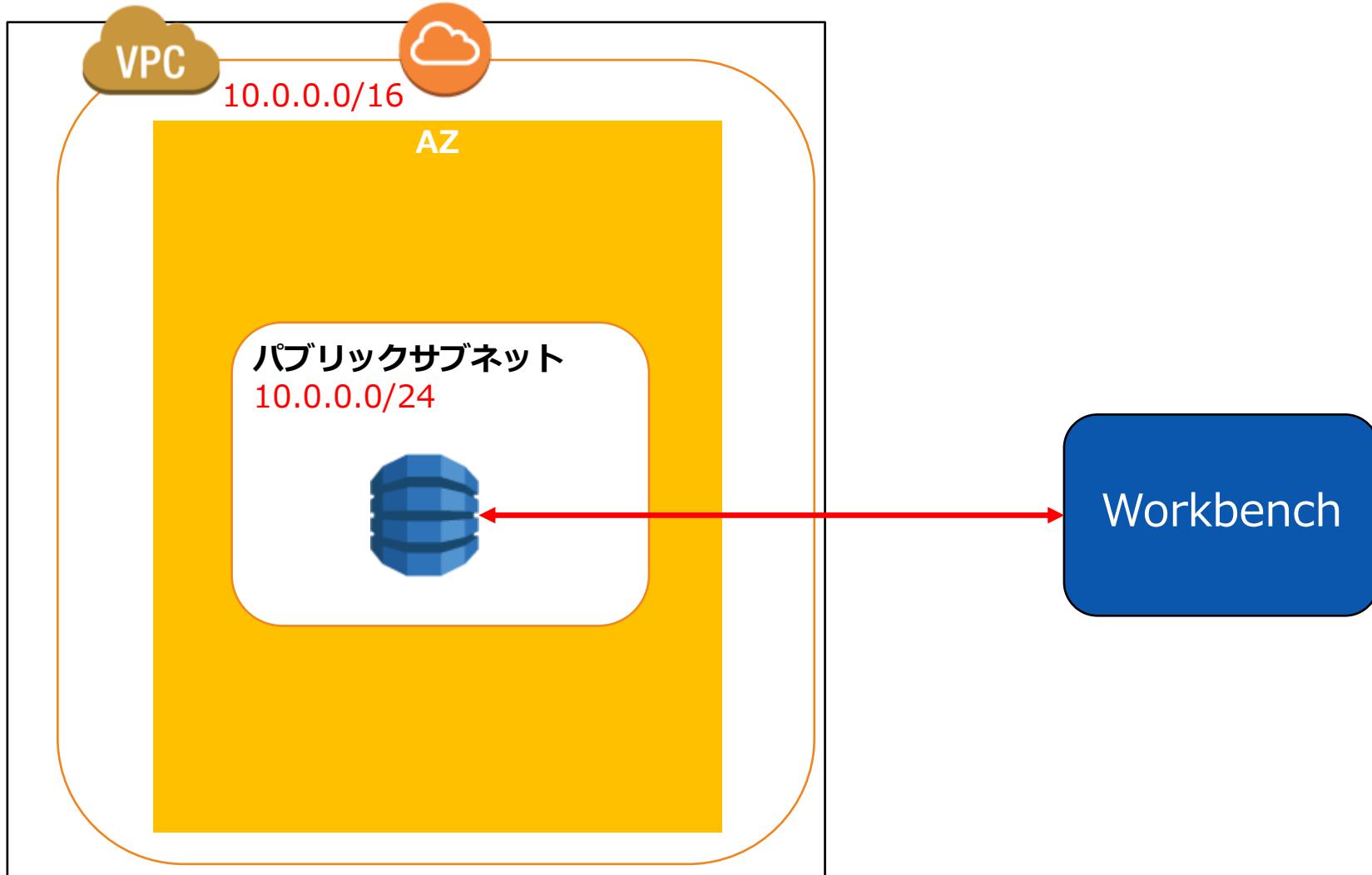
**パブリックアクセシビリティ** [info](#)

はい  
DB インスタンスをホストしている VPC 外部の EC2 インスタンスとデバイスは、DB インスタンスに接続します。DB インスタンスに接続できる EC2 インスタンスおよびデバイスを指定する 1 つ以上の VPC セキュリティグループも選択する必要があります。

いいえ  
DB インスタンスにはパブリック IP アドレスが割り当てられていません。VPC 外部のいずれかの EC2 インスタンスあるいはデバイスも接続できません。

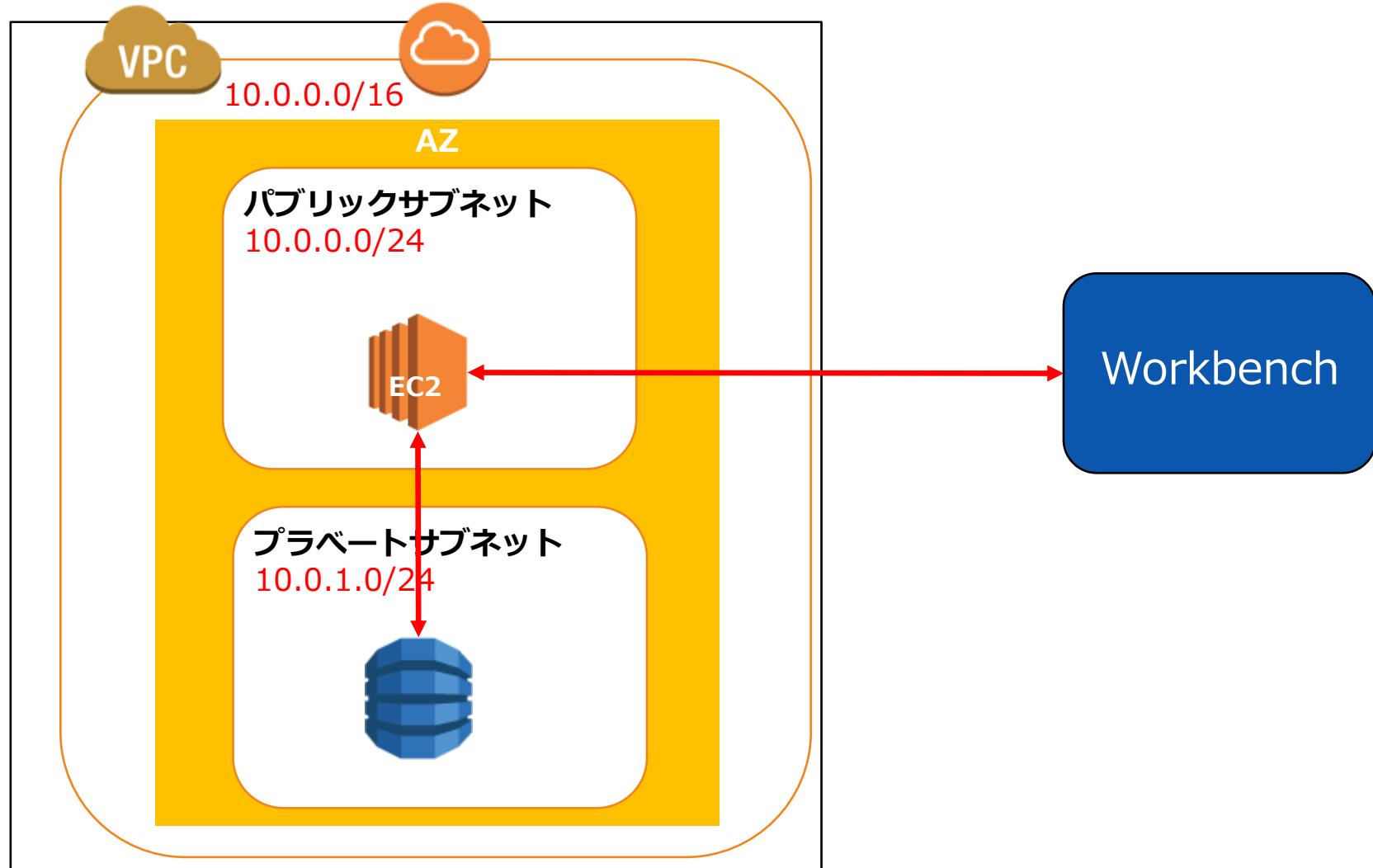
# パブリックアクセス構成

パブリックサブネットにRDSを設置し、直接にSQLソフトウェアで接続して操作する。



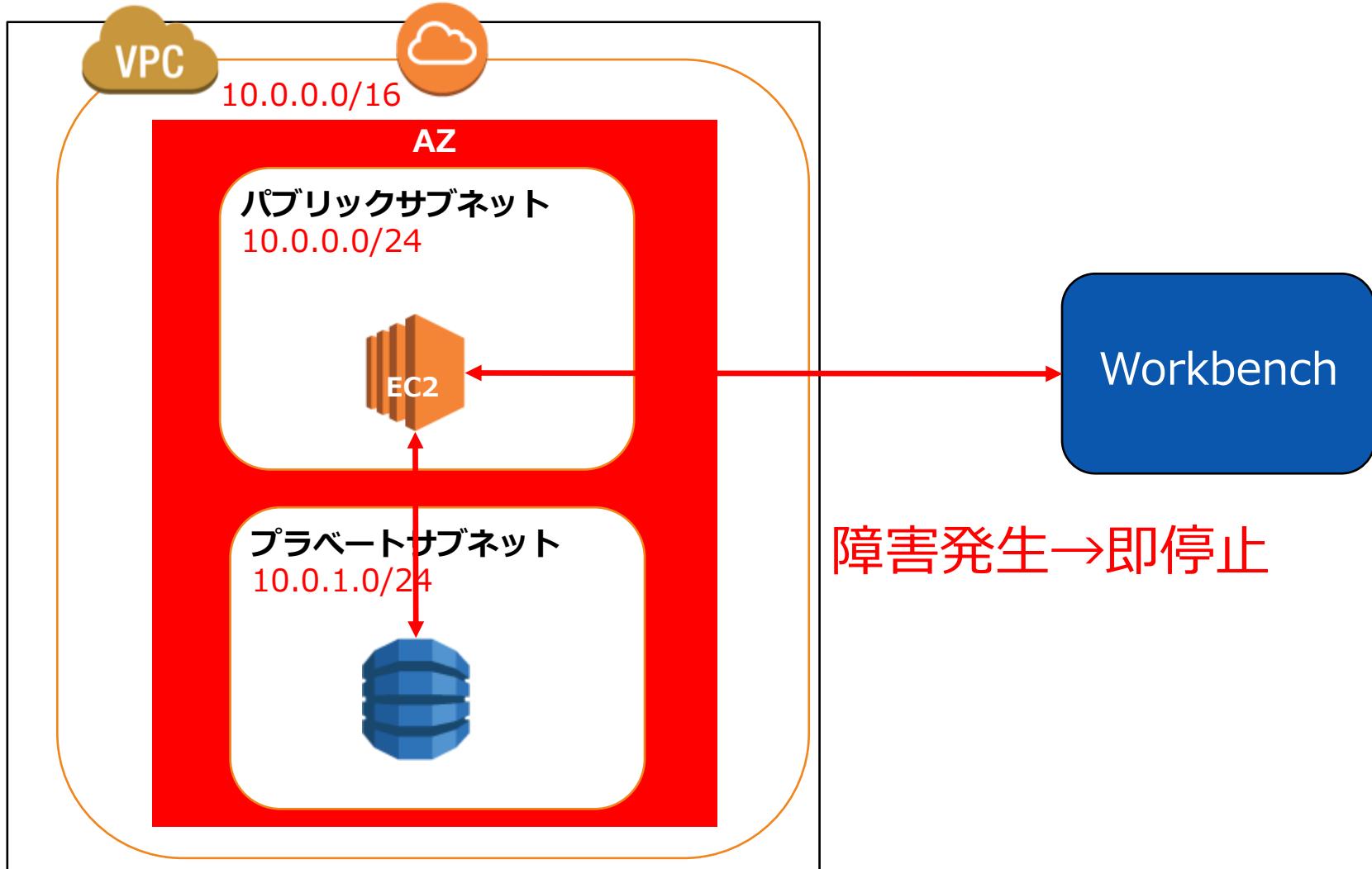
# 一般的な構成

RDSをプライベートサブネットに設置して、EC2インスタンスを踏み台にしてアクセスする。



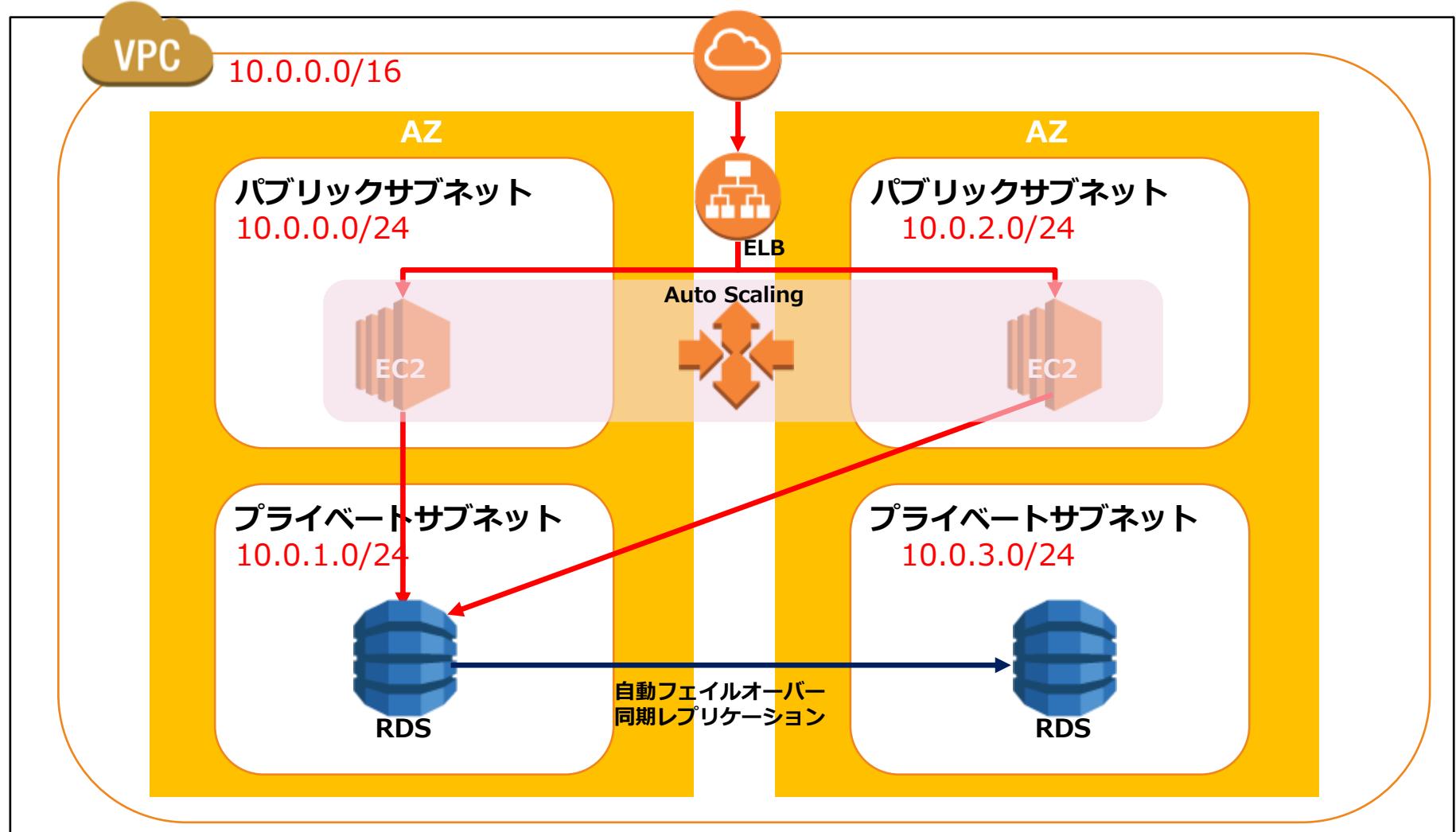
# 一般的な構成

この構成は1つのAZに依存しているため、AZ障害が発生するとダウンタイムが発生するリスクが高い。



# マルチAZ構成

マルチAZ構成にすることで、AZ障害が発生しても停止しない構成をとる必要がある。



# [Q]マルチAZ構成による効果

ある会社は自社のエンタープライズシステムの可用性を向上させるためにマルチAZ配置で構成されたRDSデータベースを利用しています。このRDSのプライマリーデータベースに障害が発生しました。

障害後にRDS上で自動でどのような対応が実施されているでしょうか？

- 1) CNAMEレコードがプライマリーデータベースからセカンダリーデータベースに移行する。
- 2) プライマリーデータベースがリブートする
- 3) RDSのセカンダリーデータベースが構成される。
- 4) スケーリングが実行される。

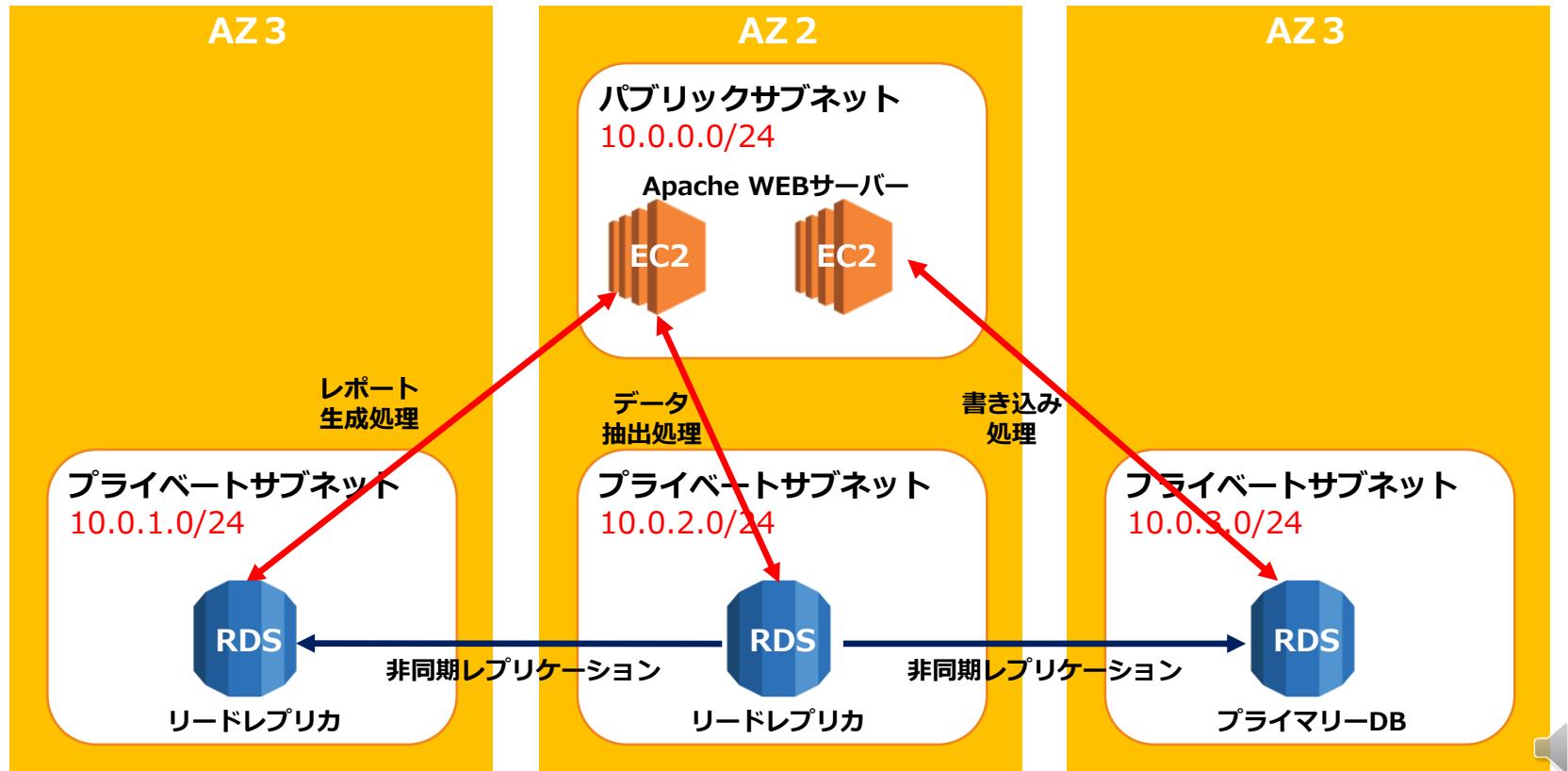
# マルチAZ構成による効果

フェールオーバー設定を有効化するだけで、非常に簡単にフェールオーバーが利用可能となる。

- ✓ プライマリーデータベースとセカンダリーデータベースの構成
- ✓ 2つのデータベースは同期レプリケーションを実施し、常に同じデータ内容を維持
- ✓ プライマリー側に障害が発生した場合、自動でフェールオーバーが実行されセカンダリーデータベースがプライマリーに昇格する。
- ✓ フェールオーバー時にCNAMEレコードがプライマリーからセカンダリーに移行する。
- ✓ スタンバイ状態のDBはアクセス不可

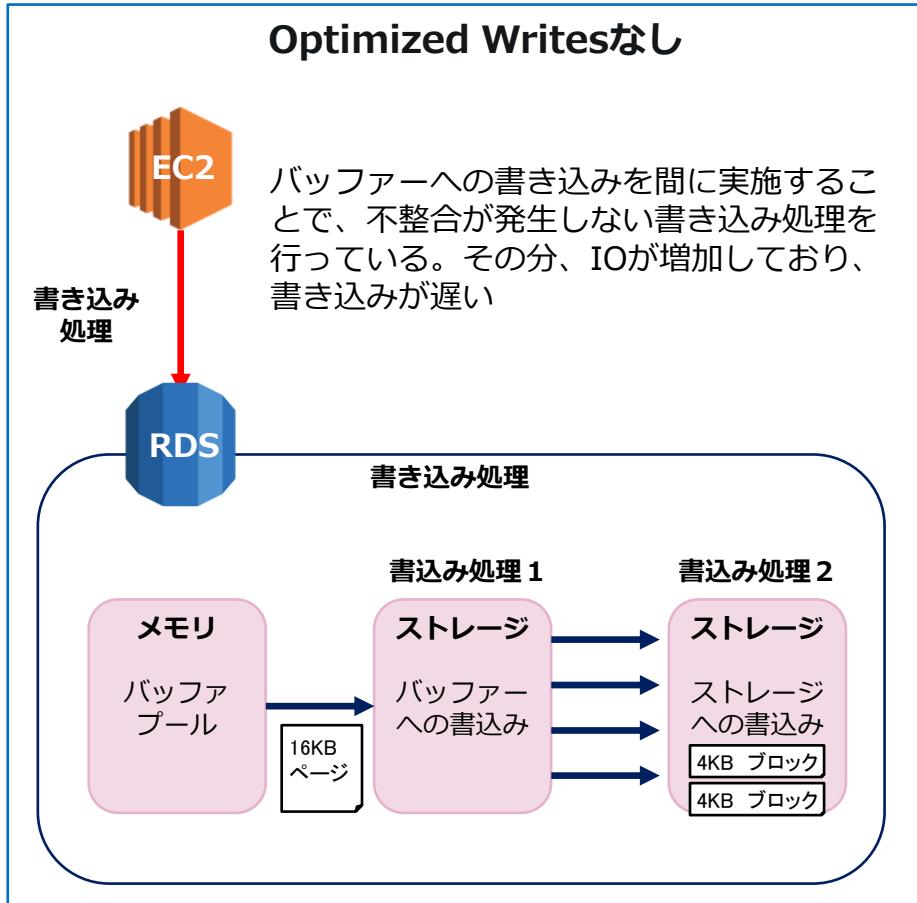
# マルチAZクラスター配置

2つの読み取り可能なスタンバイDB インスタンスを持つ、3つのAZを利用した高可用性の配置モード



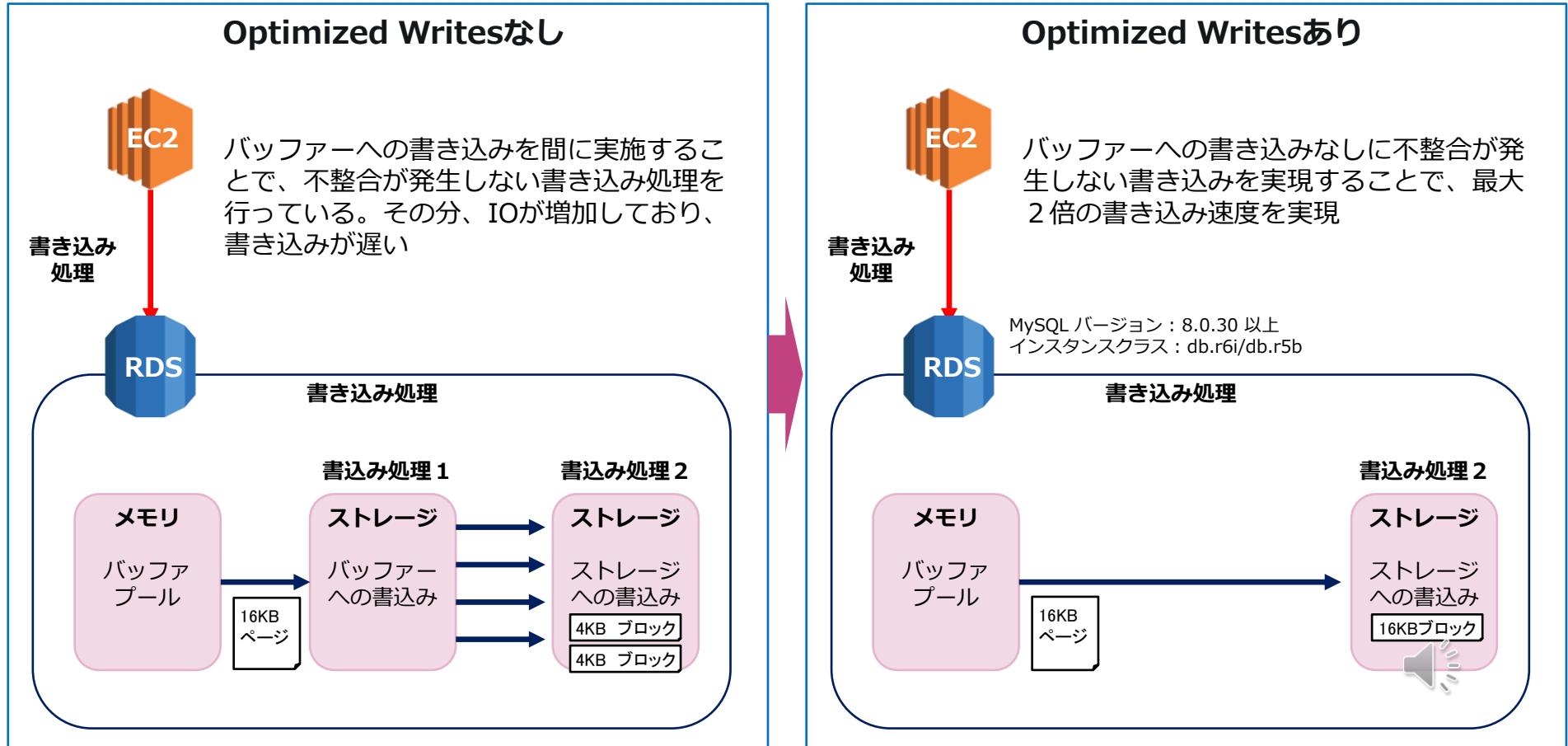
# RDS最適化書き込み (Optimized Writes)

書き込みスループットが最大で2倍にする書き込み処理技術を適用したDBエンジンとインスタンスタイプの構成



# RDS最適化書き込み (Optimized Writes)

書き込みスループットが最大で2倍にする書き込み処理技術を適用したDBエンジンとインスタンスタイプの構成



# [新Q]リードレプリカ

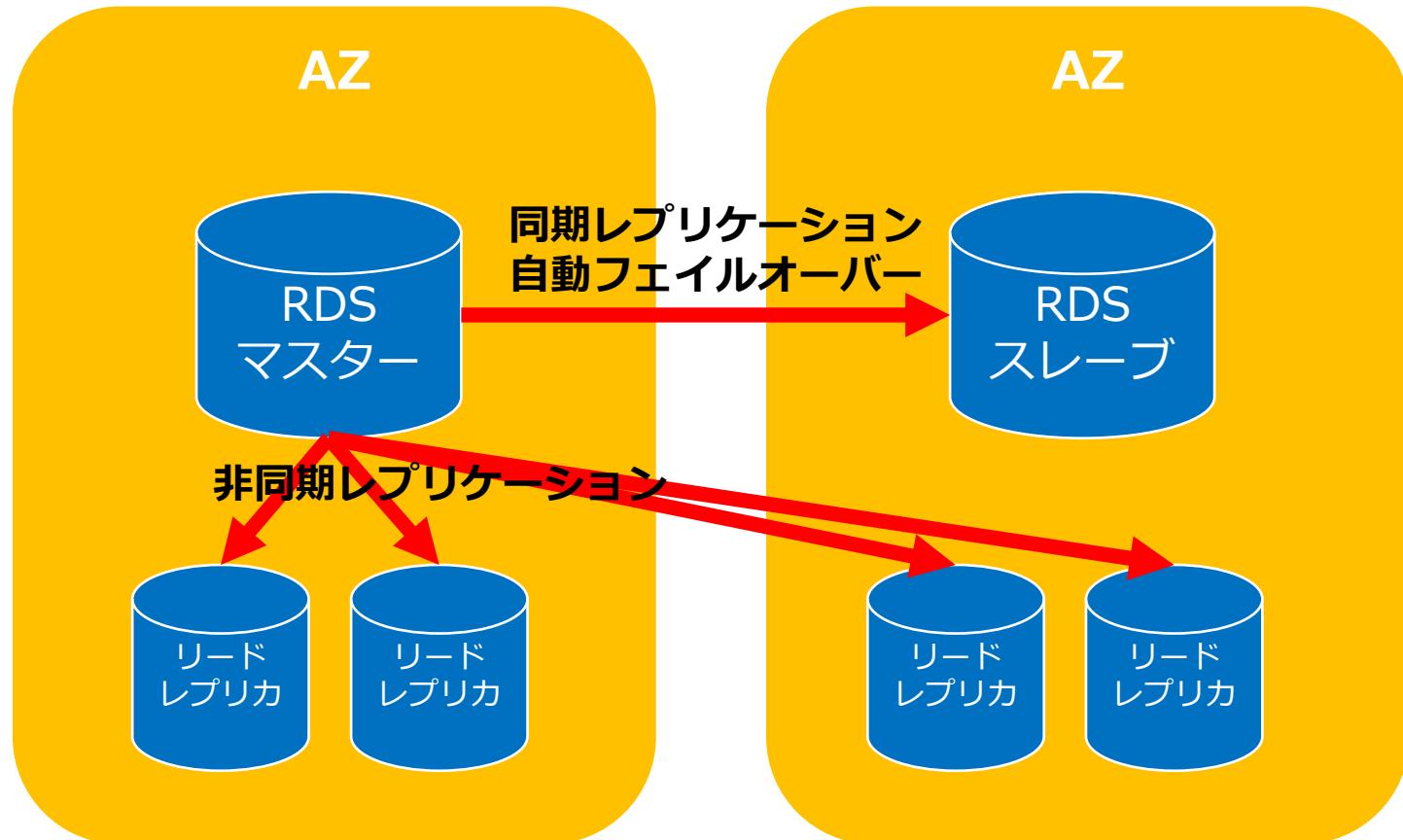
あなたの会社は、サッカーのプレミアリーグの試合に対するデータ分析システムを開発しています。このシステムは平均ゴール、平均パス、その他多くのプレーに関する詳細なデータを記録して、レポートを生成して、サッカーのプレミアリーグファンやスポーツメディアなどに提供します。これらのデータは永続的に保存する必要があるため、保存先は可用性・拡張性に優れている必要があります。リリース時はサッカーのプレミアリーグ開幕直前を予定しており、試合日には世界中のユーザーから20万件を超えるクエリが発生すると予測されています。

これらの要件を満たす最も費用対効果の高いソリューションを選択してください。

- 1) レポート作成用のリードレプリカ付のマルチAZ配置のAurora MySQLデータベースを設置して、レポートを保存する場所としてS3標準ストレージを利用する。さらにCloudFrontによる配信を実施する。
- 2) レポート作成用のリードレプリカ付のマルチAZ配置のRDS MySQLデータベースを設置して、レポートを保存する場所としてS3 標準-IAストレージを利用する。配信のためにAPI Gatewayからのアクセスを可能にする。
- 3) レポート作成用のリードレプリカ付のマルチAZ配置のAurora MySQLデータベースを設置し、レポート生成と配信処理にElastiCacheを利用したキャッシング処理を実施する。
- 4) レポート作成用のリードレプリカ付のマルチA配置のRDS MySQLデータベースを設置し、レポート生成と配信処理にDynamoDBを利用したキャッシング処理を実施する。

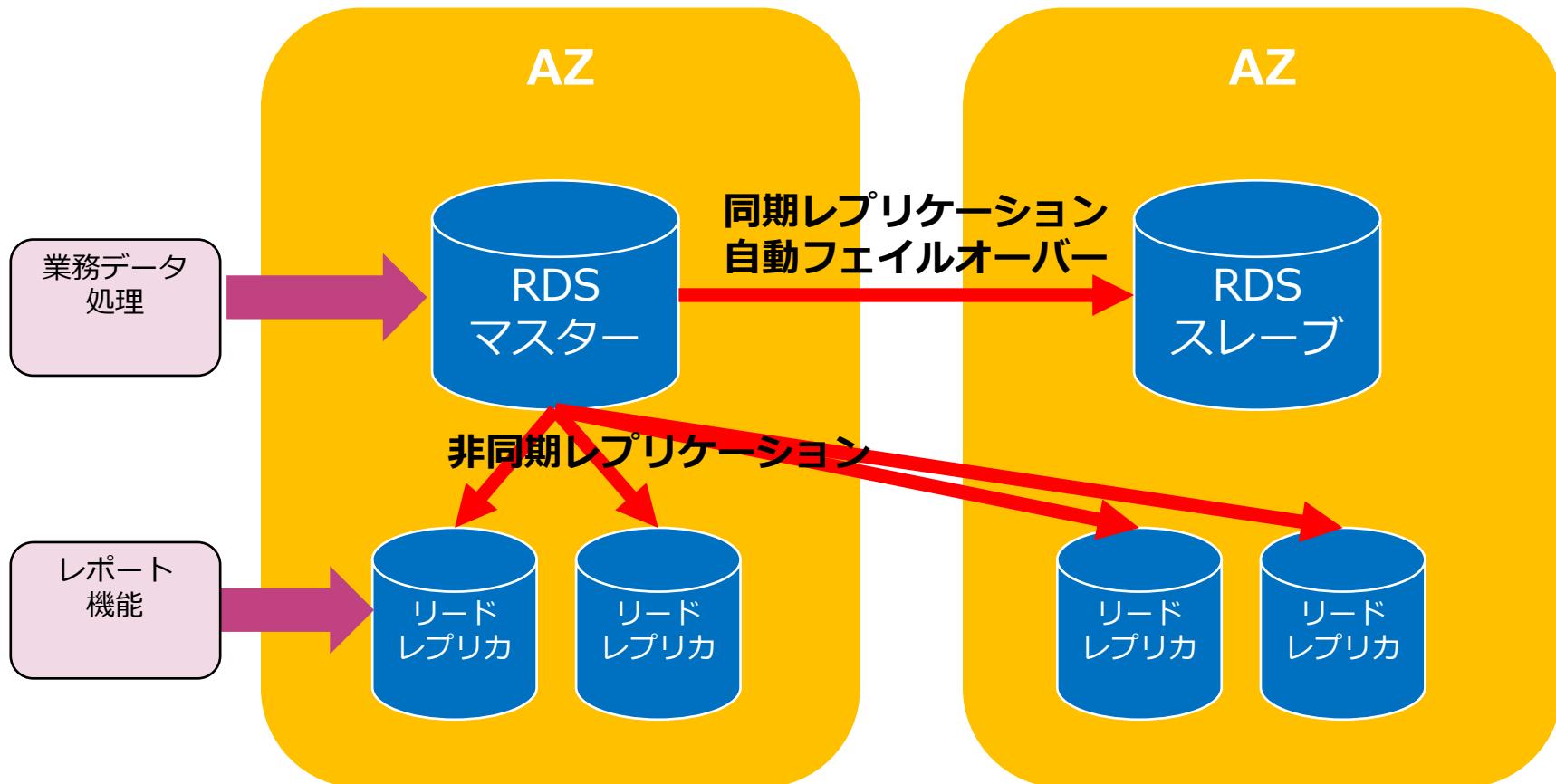
# リードレプリカ

読み取り専用のレプリカを最大15台設置し、DBの読み取り処理をスケールアウトできる



# リードレプリカ

読み取り専用のレプリカを最大15台設置し、DBの読み取り処理をスケールアウトできる



# [Q]クロスリージョンの構成

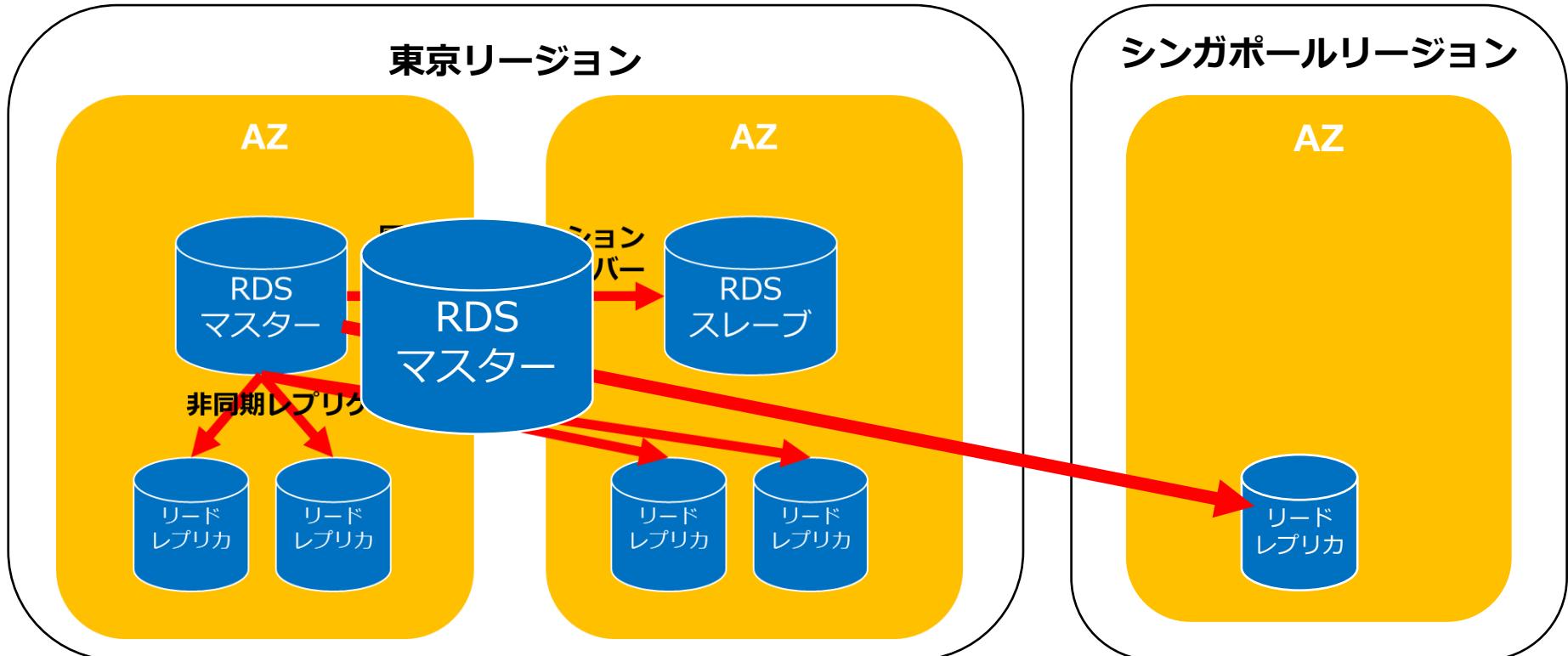
大手ECマース企業は自社のECサイトにRDS PostgreSQLデータベースを使用してます。このECサイトはアジア各国で展開されており、マスターデータベースはシンガポールリージョンに配置されていますが、データベースは他の国でのローカルの読み取りトラフィックを効果的に処理するためにパフォーマンスを増強することが必要です。

この要件を満たす最適なソリューションを選択してください。

- 1) RDSのフェールオーバー構成
- 2) クロスリージョン構成のRDS
- 3) マルチマスター構成のRDS
- 4) クロスリージョンリードレプリカを使用したRDS

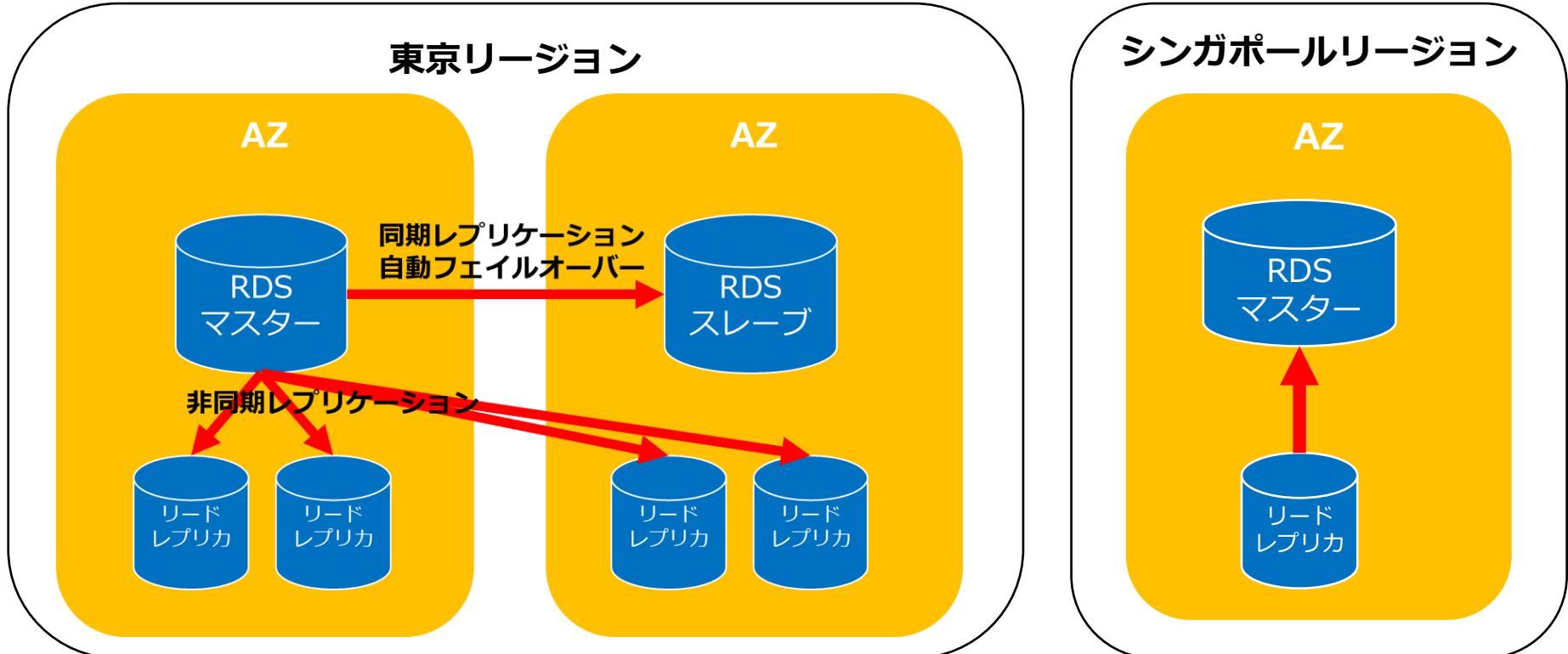
# クロスリージョンの構成

クロスリージョンでリードレプリカを構成可能



# クロスリージョンの構成

クロスリージョンでリードレプリカを構成可能



リードレプリカをマスターに昇格（プロモーション）できる。



# [Q]RDSのスケーリング

あなたはEC2インスタンスとAmazon RDSを利用して、2層アプリケーションを構築しています。現段階ではアプリケーションに必要となるワークロード要件は明確ですが、データベース処理の予想されるリクエスト数などのパフォーマンス要件が不明です。したがって、データベースを起動後にスケーリングすることが必要です。

DBインスタンスをデプロイ後に実施するべきスケーリング方法はどれでしょうか？

- 1) リードレプリカを追加する。
- 2) マルチAZ構成を有効化する。
- 3) 拡張ネットワーキングを有効化する。
- 4) より大きなインスタンスサイズを選択する。

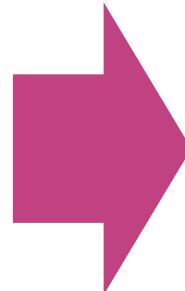
# RDSのスケーリング

データベースのパフォーマンス低下に対してスケーリング対応を実施することが求められる

RDSのパフォーマンス低下



- ✓ 読み込み処理が遅い
- ✓ 書き込みが止まる etc



- ✓ スケーリングによってパフォーマンスを向上させる

# RDSのスケールアップ

インスタンスタイプやサイズを変更することでスケールアップによるパフォーマンス向上を実施

## インスタンスサイズの変更

現在のDBインスタンスタイプに対してサイズを高性能なものに変更することで、パフォーマンスを向上させる。

## インスタンスタイプの変更

現在のDB利用方式に適したDBインスタンスタイプがある場合は、そのタイプに変更する。

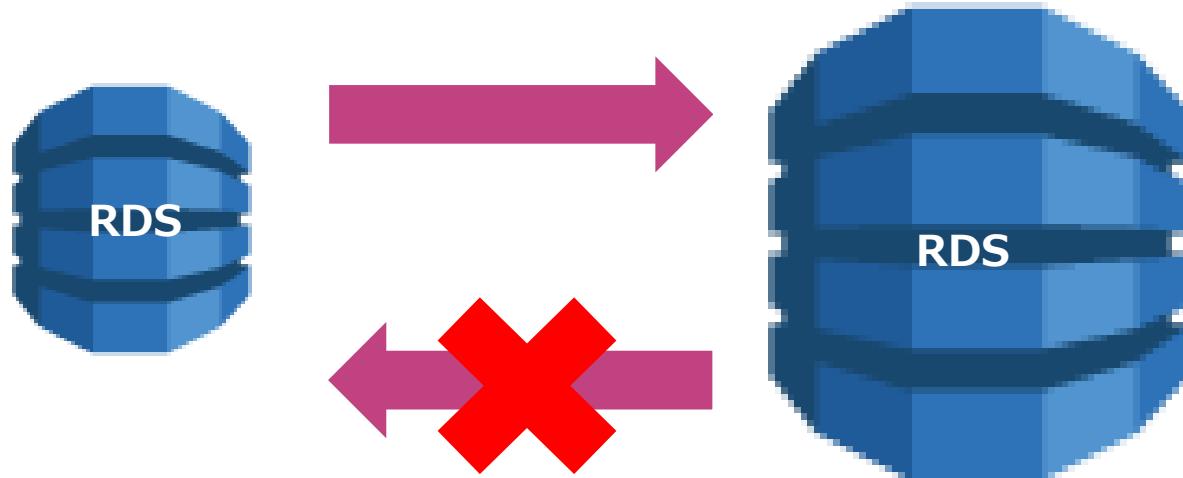
## ストレージタイプの変更

ストレージタイプを高性能なタイプ（I/O処理が多い場合はIOPS）に変更する。

# ストレージの容量変更

ストレージ容量は設定変更で増加させることはできるが、減少させることはできない。

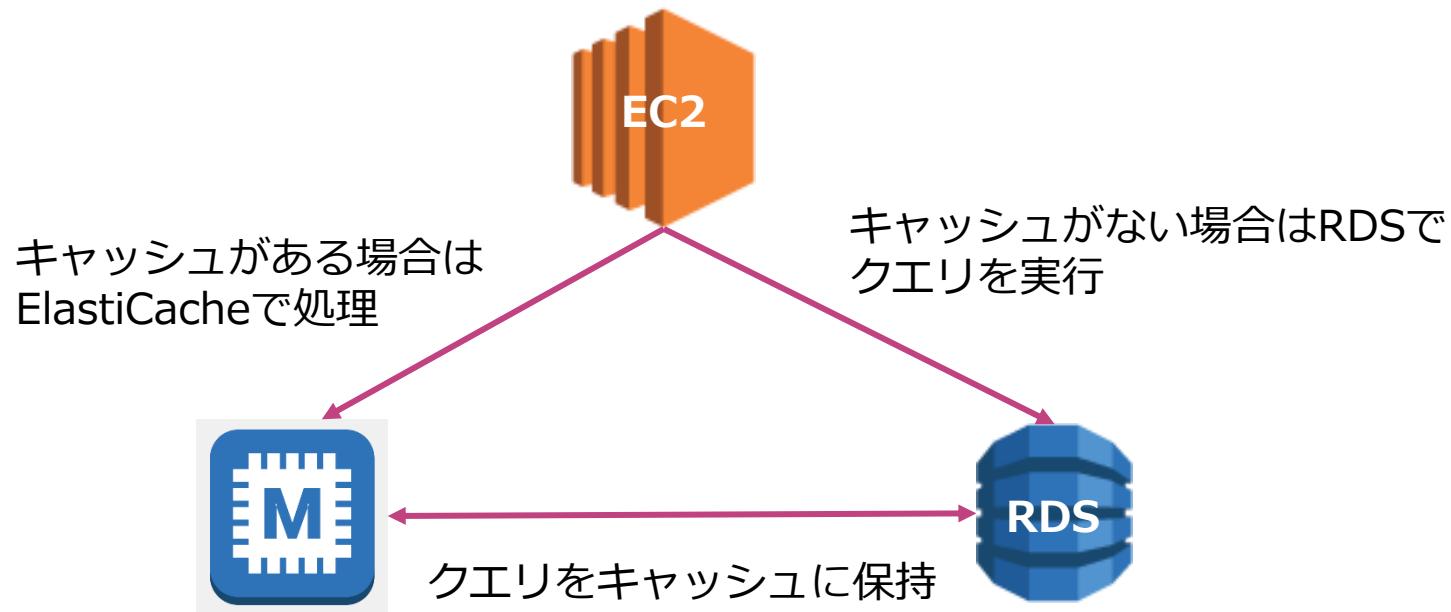
- ✓ ストレージ容量の増加は可能



- ✓ ストレージ容量の減少変更はできない

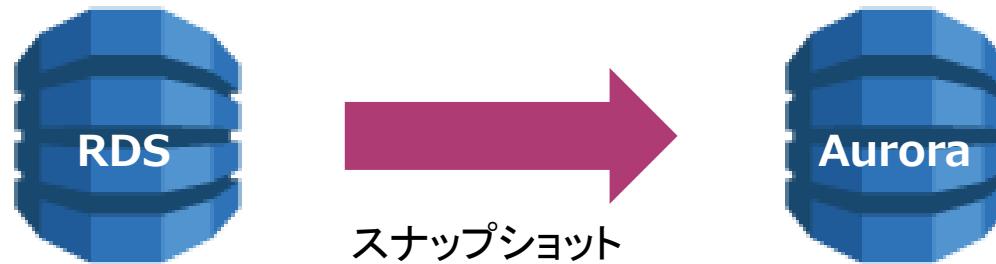
# ElastiCacheの利用

読み込み処理の一部をキャッシュに保持して、高速クエリ処理を実現する構成が可能



# Auroraへの移行

RDSのMySQLとPostgreSQLはAuroraと互換性があるバージョンは  
容易に移行が可能で、パフォーマンスを向上させることができる



# [Q] RDSの暗号化

大手ECマース企業は自社のECサイトにRDS PostgreSQLデータベースを使用してます。 最近になってIT監査が実施されたところ、RDSデータベースが暗号化されていないことを指摘されました。

暗号化手順として正しい説明を選択してください。 (2つ選択してください。)

- 1) 既存のRDSデータベースのアクション操作において、暗号化オプションの有効化を実施する。
- 2) RDSデータベースのスナップショットを作成し、暗号化されたスナップショットをコピーし、暗号化されたスナップショットからデータベースを復元する。
- 3) 既存のRDSデータベースは暗号化できないので終了する。
- 4) 既存のRDSデータベースの設定変更画面において、暗号化オプションの有効化を実施する。
- 5) RDSデータベースの暗号化されたリードレプリカを作成し、これをマスターデータベースに昇格させる。

# RDSの暗号化

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

## 通信の暗号化

- ✓ SSL/TLSを使用してDB インスタンスへの接続を暗号化する。
- ✓ SSL/TLS証明書はデフォルトで自動で構成されて暗号化される。

## 保管データの暗号化

- ✓ AWS KMSの暗号化キーを利用して、保管時のデータリソースを暗号化する。

# RDSの暗号化

保管時のデータ/DBインスタンスとスナップショットを暗号化

## 暗号化対象

- DBインスタンス
- 自動バックアップ
- リードレプリカ
- スナップショット

## 暗号化方式

- AES-256暗号化
- AWS KMSの暗号化キーを利用して暗号化される。
- リードレプリカも同じ鍵を利用して暗号化される。
- インスタンス作成時にのみ暗号化を実施する。途中で暗号化することはできない。
- スナップショットのコピーの暗号化する。
- スナップショットからDBをリストアする際に暗号化キーの権限が必要となる。

## [Q]メンテナンス

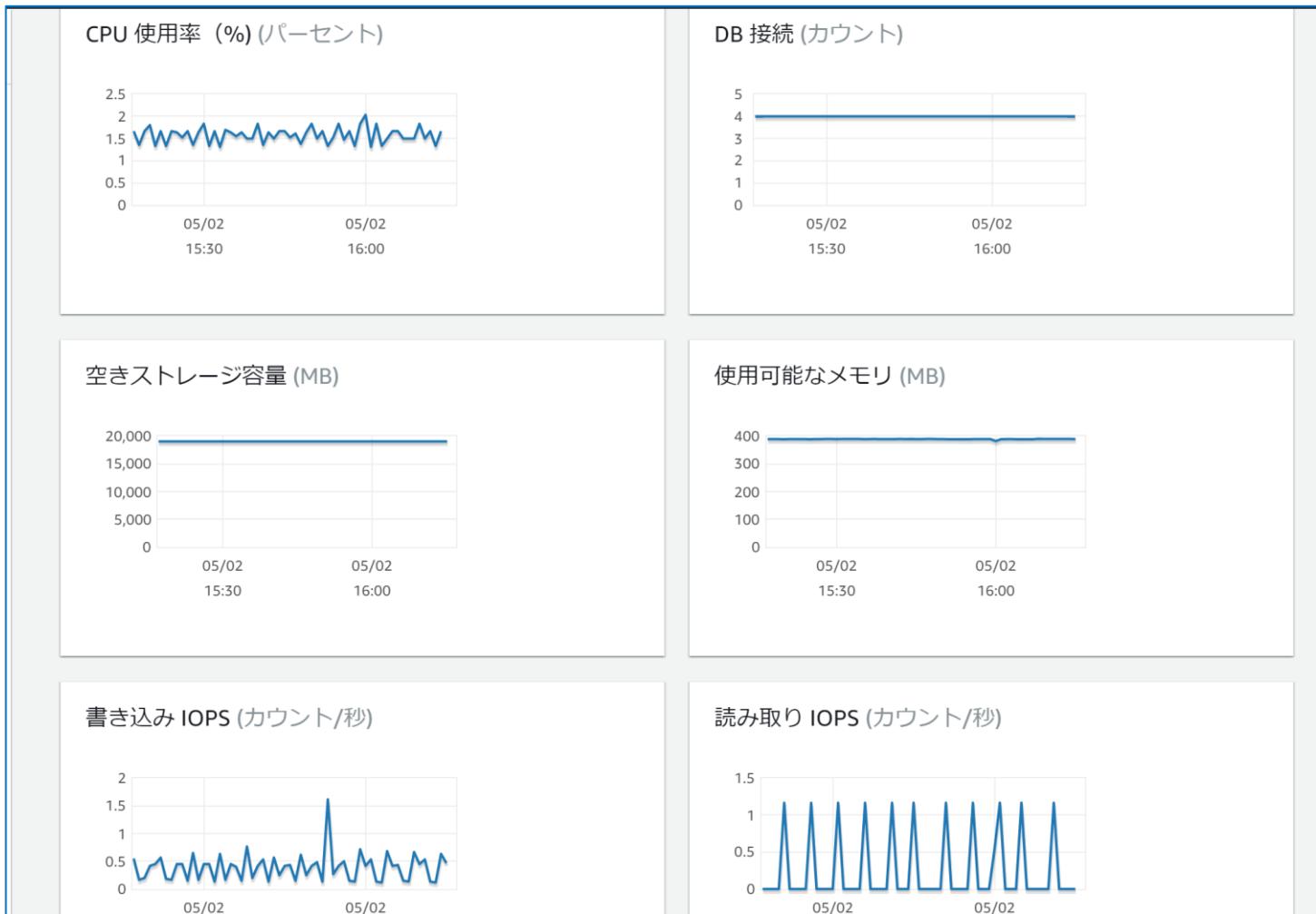
B社はAWS上にデータベース環境を整備しようと計画しています。RDSを利用することを検討していますが、メンテナンスの多くがマネージド型サービスで提供されるため、その内容を把握することが必要です。特に、メンテナンスウィンドウ中にDBインスタンスが強制的にオフラインになると影響が大きいため、その期間を回避することが必要です。

データベースのダウンタイムが発生するイベントを選択してください。（2つ選択してください。）

- 1) セキュリティパッチの適用
- 2) マルチAZ機能の適用
- 3) データベースのアップデート
- 4) DBパラメーターグループの更新
- 5) オプショングループの更新

# AWSコンソールダッシュボード

RDSのコンソールダッシュボードには、RDSインスタンスの状態が一目でわかるビューが表示される。



# ログの確認

ダッシュボードでログの確認・ダウンロードを実施可能

DBエンジン	ログのタイプ	ログの保持期間 (デフォルト)
MySQL/MariaDB	<ul style="list-style-type: none"><li>一般クエリログ</li><li>エラーログ</li><li>スロークエリ</li></ul>	<ul style="list-style-type: none"><li>24時間</li></ul>
Oracle	<ul style="list-style-type: none"><li>アラートログ</li><li>監査ログ</li><li>トレースログ</li></ul>	<ul style="list-style-type: none"><li>アラートログは30日</li><li>監査ログとトレースログは7日</li></ul>
SQL Server	<ul style="list-style-type: none"><li>エラーログ</li><li>エージェントログ</li><li>トレースログ・ダンプログ</li></ul>	<ul style="list-style-type: none"><li>7日</li></ul>
PostgreSQL	<ul style="list-style-type: none"><li>クエリログ</li><li>エラーログ</li></ul>	<ul style="list-style-type: none"><li>3日</li></ul>

# CloudWatchとの連携

CloudWatchと連携して、集中的にRDSのメトリクスに基づいた運用管理を実行することが可能となる。

<b>CloudWatch メトリクス</b>	Amazon RDS のアクティブな各データベースのメトリクスを 5 分間隔で取得して、ダッシュボードに表示する。
<b>拡張モニタリング</b>	送信間隔を秒単位でメトリクスが取得され、ほぼリアルタイムでのモニタリングが可能となる。 モニタリングコストが有料になる。
<b>CloudWatchアラーム</b>	特定の期間にわたって単一の Amazon RDS メトリクスを監視し、指定したしきい値に関連するメトリクス値に基づいて 1 つ以上のアクションを実行できる
<b>CloudWatch Logs</b>	CloudWatch Logs のデータベースログファイルの監視、保存、およびアクセスが可能になる。
<b>Amazon EventBridge (旧CloudWatch イベント)</b>	Amazon EventBridgeでは、イベントパターンを使用して受信イベントをフィルタリングし、ターゲットをトリガーするルールを作成することができる。

# AWS コンソールダッシュボード

メンテナンスとバックアップではメンテナスウィンドウとバックアップウィンドウの設定が確認できる

The screenshot shows the AWS RDS maintenance and backup configuration interface. It includes sections for maintenance windows, reserved maintenance, and backup windows.

**メンテナンス (Maintenance)**

マイナーバージョン自動アップグレード 有効	メンテナスウィンドウ mon:19:55-mon:20:25 UTC (GMT)	メンテナスの保留中	保留中の変更
--------------------------	---	-----------	--------

**保留中のメンテナンス (0) (Reserved Maintenance (0))**

C	今すぐ適用	次のメンテナスウィンドウで適用					
保留中のメンテナンス のフィルタリング							
説明	▼	タイプ	▼	ステータス	▼	日付の適用	▲
保留中のメンテナンスはありません							

**バックアップ (Backups)**

自動バックアップ 有効 (10 日) スナップショットにタグをコピー ー 有効	復元可能な最も早い時刻 May 2nd 2020, 4:10:00 pm UTC--9 (local)	バックアップウィンドウ 13:50-14:20 UTC (GMT)
---	--	--------------------------------------

必要なオペレーティングシステムやデータベースのパッチの適用時にはDBインスタンスを一時的にオフラインにするので注意が必要

# [Q] バックアップ

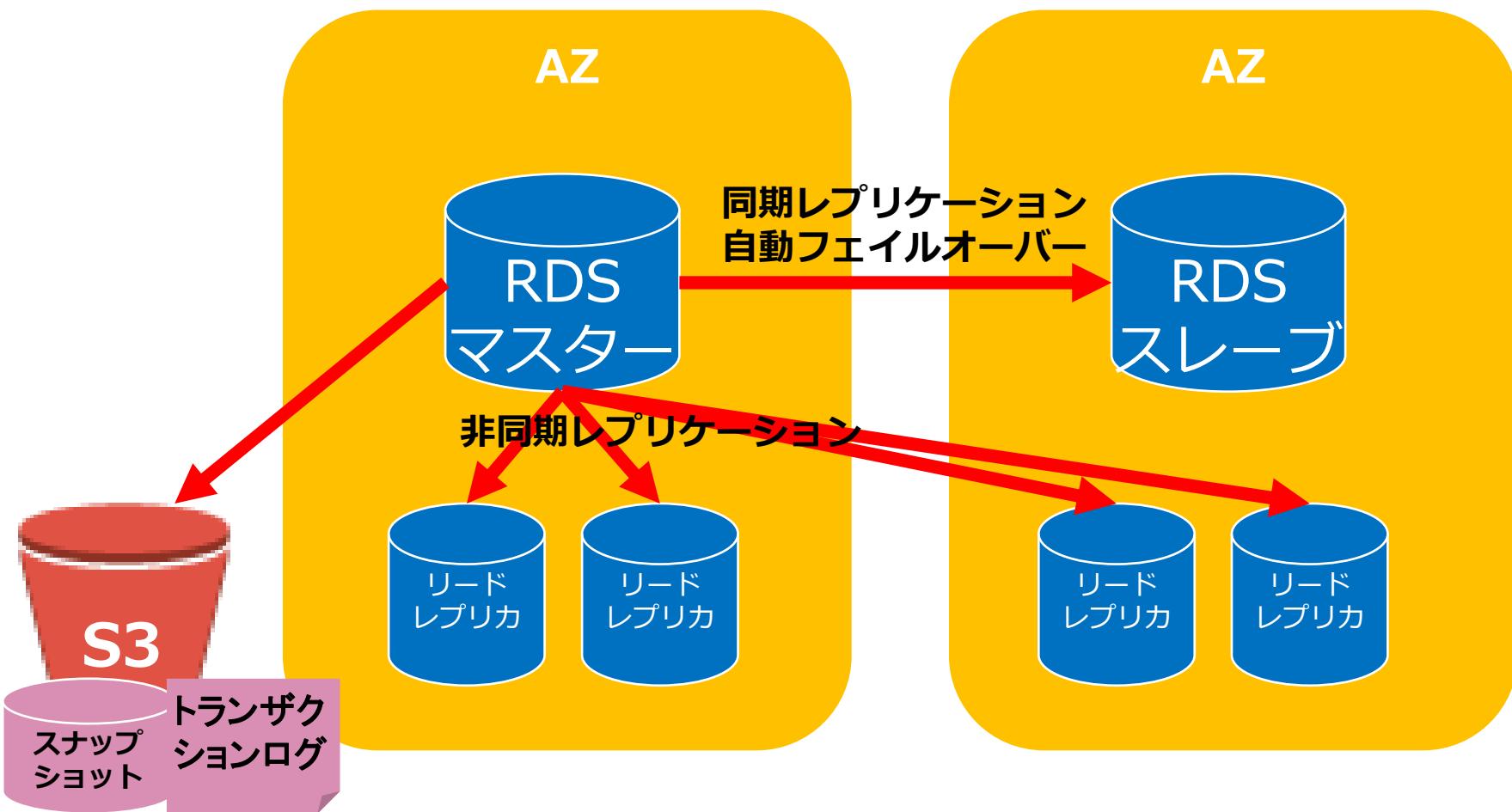
B社はAWS上にRDSを利用してデータベース環境を利用しています。しかしながら、データベースが障害によって破損したため、復元が必要となりました。あなたはソリューションアーキテクトとして、ポイントインタイムリカバリを使用して、データの最新な構成にリカバリすることになりました。

RDSのDBインスタンスを特定時点に復元する正しい方法はどれでしょうか？

- 1) スナップショットとトランザクションログを利用して、DBを5分前の状態に復元できる。
- 2) スナップショットを利用して、DBを5分前の状態に復元できる。
- 3) トランザクションログを利用して、DBを10分前の状態に復元できる。
- 4) スナップショットとトランザクションログを利用して、DBを10分前の状態に復元できる。

# バックアップ

スナップショットを取得することでデータを保存し、耐障害性を確保することができる。



# バックアップ

RDSのバックアップはスナップショットで取得され、2つの方法が提供されている。

## 自動バックアップ

自動バックアップ有効化されると、Amazon RDS は毎日、データのスナップショットを自動的に作成するポイントタイムリカバリが可能

## スナップショットの取得

ユーザーによって指定された頻度でスナップショットを取得することが可能

# 自動バックアップ

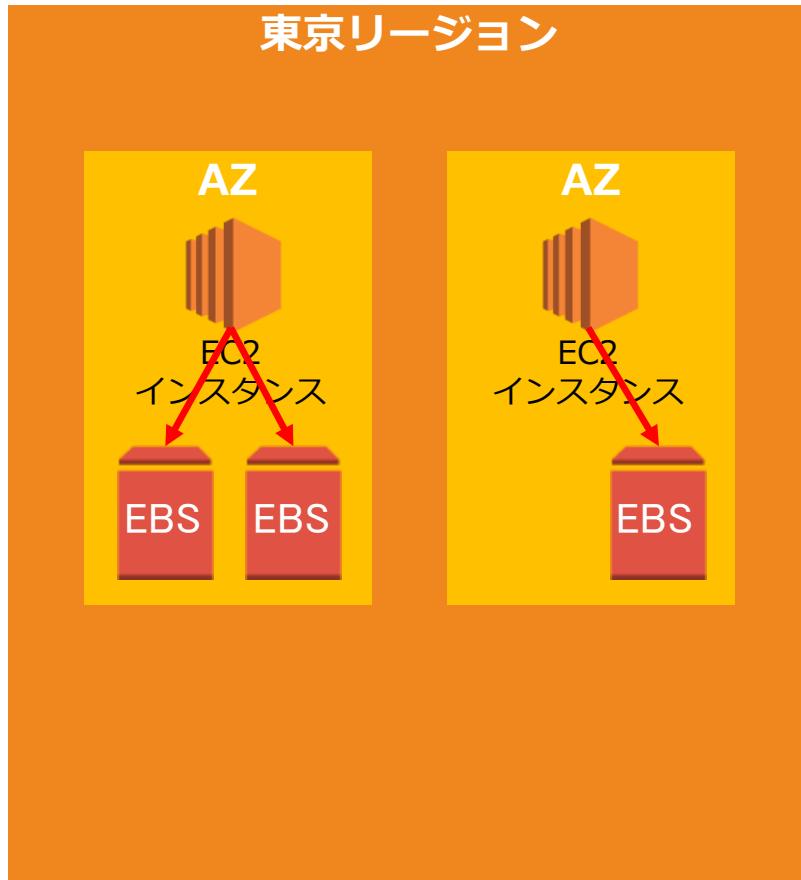
自動バックアップはRDS側で管理された定期的スナップショット取得を自動で実施する。

- ✓ 自動でのスナップショット取得とトランザクションログを取得
- ✓ 最も適切なデイリーバックアップとトランザクションログを利用して、DB インスタンスを特定の時刻の状態に復元することができる。
- ✓ バックアップサイクルは「1日1回」で固定されている。
- ✓ 5分毎にトランザクションログのアーカイブを自動で行っており、これにより、ポイントインタイムリカバリが可能となる。
- ✓ バックアップの保存期間はデフォルト7日で最大35日まで設定できる。
- ✓ 増分バックアップを実施する。
- ✓ RDSのバックアップはAWSが管理するS3ストレージに保存される。
- ✓ DBインスタンスを削除した場合や自動バックアップを無効にした場合に、スナップショットは削除される。

## EBSの出題範囲

# EBSとは何か？

EBSはEC2インスタンスと共に利用されるブロックストレージ。  
インスタンス上のワークロードなどに利用



# EBSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

EBSの選択	✓ シナリオのストレージ要件を満たすストレージを選択する質問
EBSの特徴	✓ EBSの特徴を回答させる質問 ✓ EC2インスタンスにEBSのアタッチ方式やインターネットからのアクセスの有無などに関する質問
EBSボリュームタイプの選択	✓ シナリオに基づいてワークロードの要件が提示され、EBSボリュームタイプを選択する質問が出題される。
スナップショットの特徴	✓ EBSのスナップショットの機能や特徴に関する質問が出題される。
スナップショットの管理	✓ スナップショットを利用して定期的なバックアップ取得などの設定方法が出題される。

# EBSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

スナップショットの共有	<ul style="list-style-type: none"><li>✓ スナップショットを別アカウントと共有する方法が問われる。</li><li>✓ スナップショットを別リージョンと共有する方法が問われる。</li></ul>
EBSボリュームの削除	<ul style="list-style-type: none"><li>✓ EBSボリュームが削除される設定状況が問われる。</li><li>✓ EC2インスタンスが削除された際のEBSの挙動が問われる。</li></ul>
EBSの暗号化	<ul style="list-style-type: none"><li>✓ EBSの暗号化設定の方法が問われる。</li><li>✓ EBSの暗号化対象範囲が問われる。</li><li>✓ 暗号化されたスナップショットの利用上の制約などが問われる。</li></ul>
EBSのステータス	<ul style="list-style-type: none"><li>✓ EBSのステータスに応じた特徴が問われる。</li></ul>
EBSのRAID構成	<ul style="list-style-type: none"><li>✓ EBSを利用したRAID0とRAID1の構成と利用方法が問われる。</li></ul>

# [Q] EBSの選択

大手金融機関はAWSを利用してFintech事業用の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、EC2インスタンスを利用してアプリケーションのデータ処理を実施する必要があり、データへの最小遅延アクセスを提供できるストレージサービスが必要です。データ容量は10TBまで増大することが予想されています。

この要件を満たす最も適切なストレージサービスは次のうちどれですか？

- 1) EFS
- 2) インスタンスストア
- 3) EBS
- 4) S3

# EBSの選択

AWSは3つの形式のストレージサービスを提供

## ブロックストレージ

- ✓ EC2にアタッチして活用するディスクサービス
- ✓ ブロック形式でデータを保存
- ✓ 高速・広帯域幅
- ✓ 例：EBS、インスタンスストア

## オブジェクトストレージ

- ✓ 安価かつ高い耐久性をもつオンラインストレージ
- ✓ オブジェクト形式でデータを保存
- ✓ デフォルトで複数AZに冗長化されている。
- ✓ 例：**S3**、Glacier

## ファイルストレージ

- ✓ 複数のEC2インスタンスから同時にアタッチ可能な共有ストレージサービス
- ✓ ファイル形式でデータを保存
- ✓ 例：EFS

# EBSの選択

EC2が利用するのはインスタンスストアとEBSの2タイプのストレージ

## 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・終了と共にデータがクリアされる
- ✓ 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とは独立管理
- ✓ EC2を終了してもEBSデータは保持可能
- ✓ SnapshotをS3に保持可能
- ✓ 別途EBS料金が必要

# [Q] EBSの特徴

B社では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、WEBサーバー用のストレージとしてEBSボリュームの汎用SSDを利用することにしました。複数のEC2インスタンスと複数のEBSボリュームを連携して利用するつもりですが、どのような設定が可能か調べています。

EBSボリュームの正しい説明は次のうちどれですか？

- 1) EBSボリュームは複数インスタンスにアタッチできる。
- 2) EBSボリュームは同じリージョンのインスタンスであればアタッチできる。
- 3) EBSボリュームは同じVPCのインスタンスにのみアタッチできる。
- 4) EBSボリュームは同じAZ内のインスタンスにのみアタッチできる。

# EBSの特徴

EC2にアタッチされるブロックレベルのストレージサービス



## 【基本】

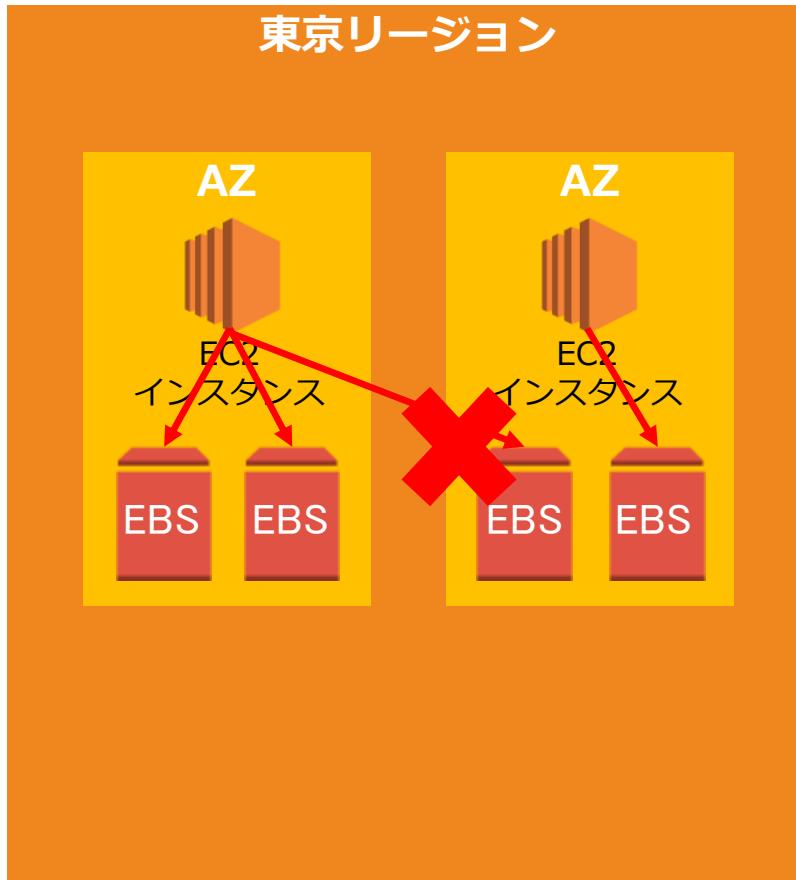
- ✓ OSやアプリケーション、データを保存するサーバー用の容量
- ✓ ネットワーク接続されたストレージ
- ✓ 99.999%の可用性
- ✓ サイズは1 GB～16TB
- ✓ プロビジョニングされたサイズと利用時間に応じて課金される。
- ✓ AZ固有のリソース

## 【特徴】

- ✓ ボリュームデータはAZ内で複数のHWにデフォルトでレプリケートされており、冗長化されている。
- ✓ セキュリティグループによる通信制御対象外であり、全ポートを閉じてもEBSは利用可能である。
- ✓ データは永続的に保存される。

# EBSの特徴

他のAZのインスタンスにはアタッチできない。

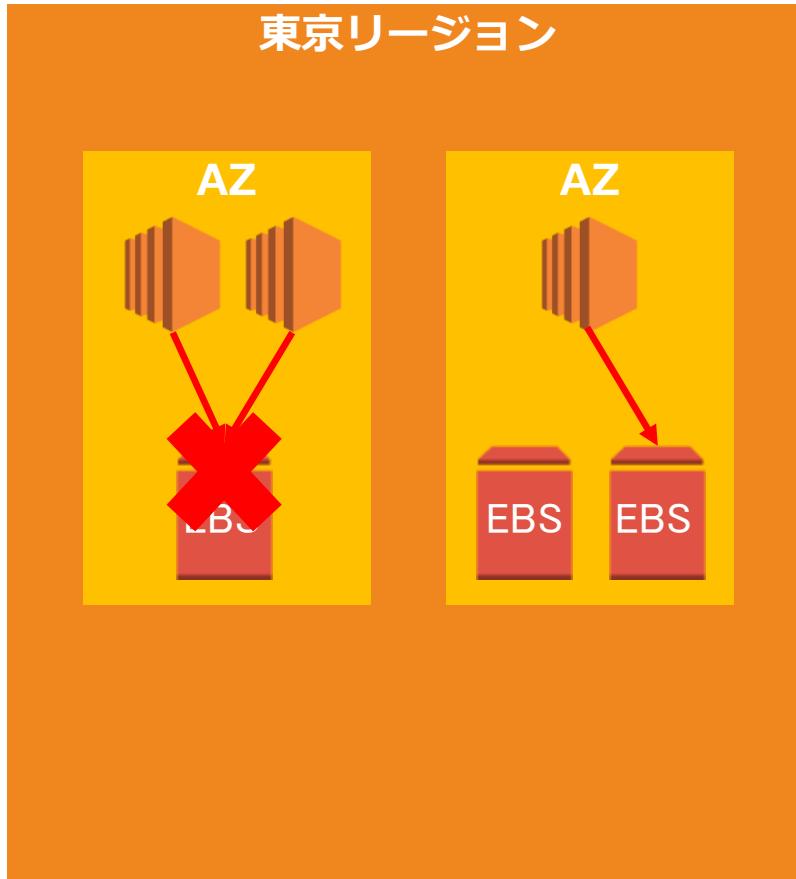


## 【特徴】

- ✓ EC2インスタンスは他のAZ内のEBSにアクセスできない

# EBSの特徴

1つのEBSを複数のインスタンスで共有することはできない。



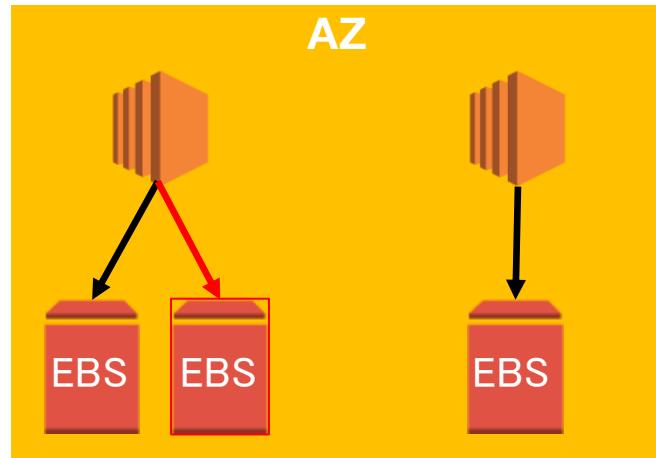
## 【特徴】

- ✓ EC2インスタンスに複数のEBSを接続することはできるが、EBSを複数のインスタンスで共有することはできない
- ✓ ただし、**プロビジョンドIOPSは複数インスタンスで共有するマルチアタッチ機能**を有している。



# EBSの特徴

同じAZ内のインスタンスのみ付け替えが可能

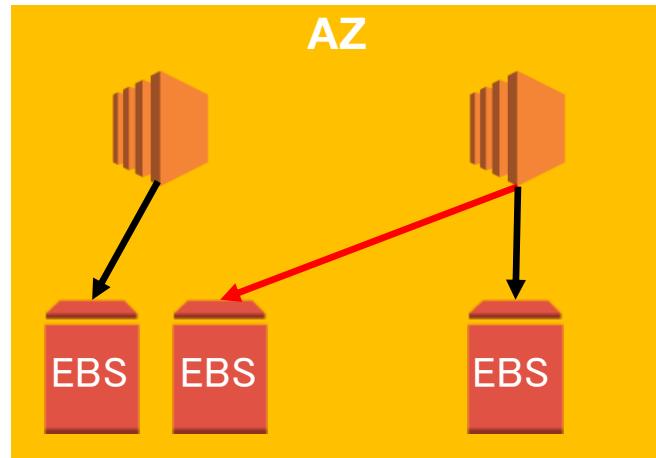


## 【特徴】

- ✓ 他のインスタンスに付け替えできる

# EBSの特徴

同じAZ内のインスタンスのみ付け替えが可能



## 【特徴】

- ✓ 他のインスタンスに付け替えできる

# [新Q] EBSボリュームタイプの選択

ある企業は、オンプレミス環境にあるアプリケーションをAWSに移行することを計画しています。このアプリケーションでは、100,000 IOPS以上の高性能なI/O処理が必要であるため、それに最適なEC2インスタンスとストレージを選択する必要があります。また、データ処理後に大量のデータを保存することになるため、耐久性が非常に高いストレージも必要です。これらの保存されたデータへのアクセス頻度は予測がつきません。

この要件を満たすために、どのようなAWSサービスを利用しますか。

- 1) 高性能なI/O処理用にAmazon EBSのプロビジョンド IOPS SSD (io2) Block Express ボリュームを利用する。耐久性の高いデータ保存用にAmazon S3 Intelligent-Tiering クラスを利用する。
- 2) 高性能なI/O処理用にAmazon EBSのプロビジョンド IOPS SSD (io1) Block Expressボリュームを利用する。耐久性の高いデータ保存用にAmazon S3 Standard IFクラスを利用する。
- 3) 高性能なI/O処理用にAmazon EBSのプロビジョンド IOPS SSD (io2) ボリュームを利用する。耐久性の高いデータ保存用にAmazon S3 Intelligent-Tiering クラスを利用する。
- 4) 高性能なI/O処理用にAmazon EBSのプロビジョンド IOPS SSD (io3) ボリュームを利用する。耐久性の高いデータ保存用にAmazon S3 Standardクラスを利用する。

# EBSのボリュームタイプ

ユースケースに応じて性能やコストが異なるボリュームタイプから選択する。

	ユースケース	ボリュームあたり最大I/O 最大スループット	サイズ
SSD	<b>汎用SSD (gp2, gp3)</b>	✓ 仮想デスクトップ ✓ 低レイテンシーを要求するアプリ ✓ 小～中規模のデータベース ✓ 開発/テスト環境	16,000IOPS Gp2は1,000 MiB/秒 Gp1は250 MiB/秒 1GiB～16TiB
	<b>プロビジョント IOPS (io1, io2)</b>	✓ ミリ秒未満のレイテンシー ✓ 持続的な IOPSパフォーマンス ✓ 64,000 IOPS や1,000 MiB/秒スループット ✓ マルチアタッチが可能	64,000IOPS 1,000 MiB/秒 4GiB～16TiB
	<b>プロビジョント IOPS io2 Block Express</b>	✓ 超高性能なデータ処理用 ✓ 持続的な IOPSパフォーマンス ✓ 256,000 IOPS や4,000MiB/秒スループット ✓ マルチアタッチが可能	256,000IOPS 4,000 MiB/秒 4GiB～64TiB
HDD	<b>スループット 最適化HDD</b>	✓ ビッグデータ処理 ✓ DWH ✓ 大規模なETL処理やログ分析 ✓ ルート（ブート）ボリュームには利用不可	500IOPS 500 MiB/秒 125GiB～16TiB
	<b>コールドHDD</b>	✓ ログデータなどアクセス頻度が低いデータ ✓ バックアップやアーカイブ ✓ ルート（ブート）ボリュームには利用不可	250IOPS 250 MiB/秒 125GiB～16TiB
	<b>マグネティック (Magnetic)</b>	✓ 旧世代のボリュームで基本利用しない ✓ データのアクセス頻度が低いワークロード ✓ ルート（ブート）ボリュームには利用不可	40～200IOPS 40～90 MiB/秒 1GB～1TB

# [Q]スナップショットの特徴

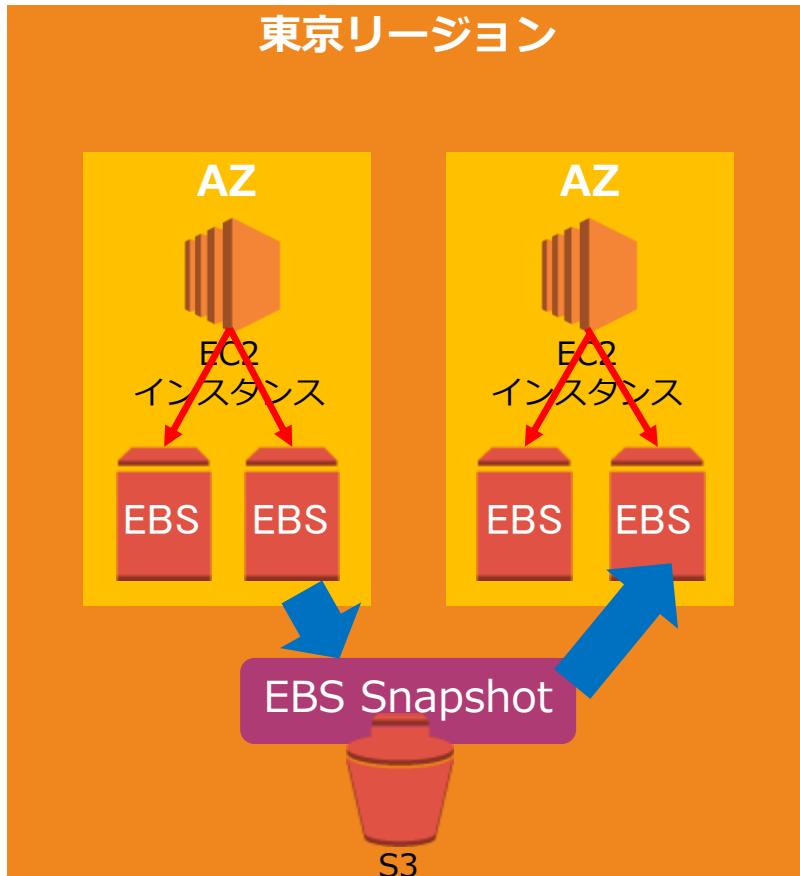
あなたはソリューションアーキテクトとして、会社内でAWSの管理を任せています。この会社ではAWSの利用コストが増大しており、 AWS Trusted Advisorを利用してコスト最適の余地を検証しました。 AWS Trusted Advisorによるとスペースとコストを節約するために、未使用のEBSボリュームとスナップショットをクリーンアップすることによってコストを削減できるようです。

スナップショットを削減する際の注意点として正しい説明はどれでしょうか？

- 1) 増分スナップショットであるため一連のスナップショットのいづれかを削除すると他のスナップショットが利用できなくなる。
- 2) 増分スナップショットであるが、最新のスナップショット以外を削除しても最新のスナップショットだけでEBSを復元できる。
- 3) 増分スナップショットであるため最初のスナップショットと最新のスナップショットだけは保持する必要がある。
- 4) 増分スナップショットであるため最初のスナップショット以外は削除できる。

# スナップショットの特徴

EBSはスナップショットを利用してバックアップを取得する



## 【特徴】

- ✓ スナップショットでバックアップ
- ✓ スナップショットからEBSを別AZにも復元ができる。
- ✓ スナップショットはAWS管理のS3バケットに保存される。
- ✓ スナップショットの2世代目以降は増分データを保存する増分バックアップとなる（1世代目を削除しても復元可能）
- ✓ スナップショット作成時にブロックレベルで圧縮して保管するため、圧縮後の容量に対して課金が行われる。
- ✓ スナップショット作成時でもEBSは利用可能である。

# [Q]スナップショットの管理

会社ではEBSを複数利用したWEBアプリケーションを利用しています。セキュリティ規定によるバックアップを定期的に実行することが必要ですが、現在は手動で行っており非常に手間がかかります。そのため、EBSボリュームのバックアップの作成、保持、および削除を自動化する方法を実装したいと考えています。

EBSのこれらのタスクを自動化する最も簡単な方法は何ですか？

- 1) S3でバックアップを作成するようにEBSボリュームレプリケーションを構成する
- 2) AWS CLIコマンドでスナップショットの実行スクリプトを定義する。
- 3) データライフサイクルマネージャー（DLM）を使用して、ボリュームのスナップショットを管理する。
- 4) EBSコンソール画面で自動バックアップを有効化する。

# スナップショットの管理

スナップショットの作成時には静止点が推奨されているものの、いつでも実行可能でEBS操作に影響を与えない。DLMにより取得期間を設定可能

- スナップショット作成時はデータ整合性を保つため静止点の設定を推奨
  - ソフトウェアの機能を利用
  - ファイルシステムの機能を利用
  - バックアップソフトウェアの機能を利用
  - アプリケーションの停止
  - ファイルシステムのアンマウントなど
- 保存期間や世代数は無制限
- 世代管理が必要な場合はAWS CLIやAPI等で自動化する
- DLMを利用してスナップショット取得をスケジューリングできる。

# [Q]スナップショットの共有

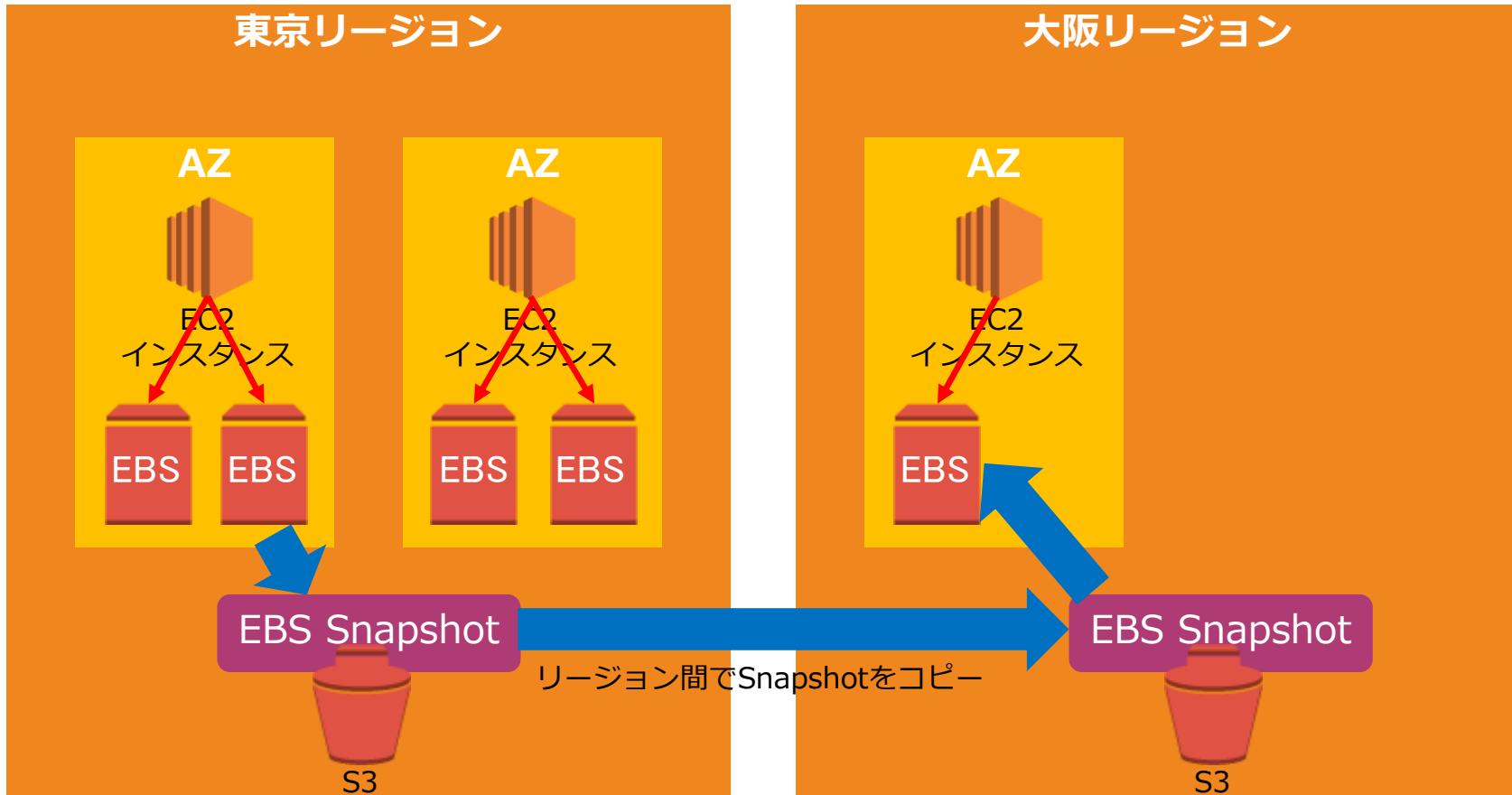
会社は複数部門でAWSアカウントを有してAWSリソースを様々な用途に利用しています。A部門のAアカウントにあるEBSをB部門のBアカウントでも利用することが必要となっており、あなたはソリューションアーキテクトとして、対応を求められています。このスナップショットは、カスタムキーで暗号化されたEBSボリュームから取得されました。

暗号化されたEBSスナップショットを共有する手順の正しい組合せはどれでしょうか？（2つ選択してください）

- 1) EBSボリュームのコピーを別アカウントにコピーする設定を行う。
- 2) EBSスナップショットの暗号化を非有効化する。
- 3) EC2コンソール画面でBアカウントIDを指定したスナップショットの共有設定を行う。
- 4) ボリュームの暗号化に使用されるカスタマーキーを共有する
- 5) EC2コンソール画面で暗号化されたスナップショットの権限を変更して、Bアカウントに設定する。

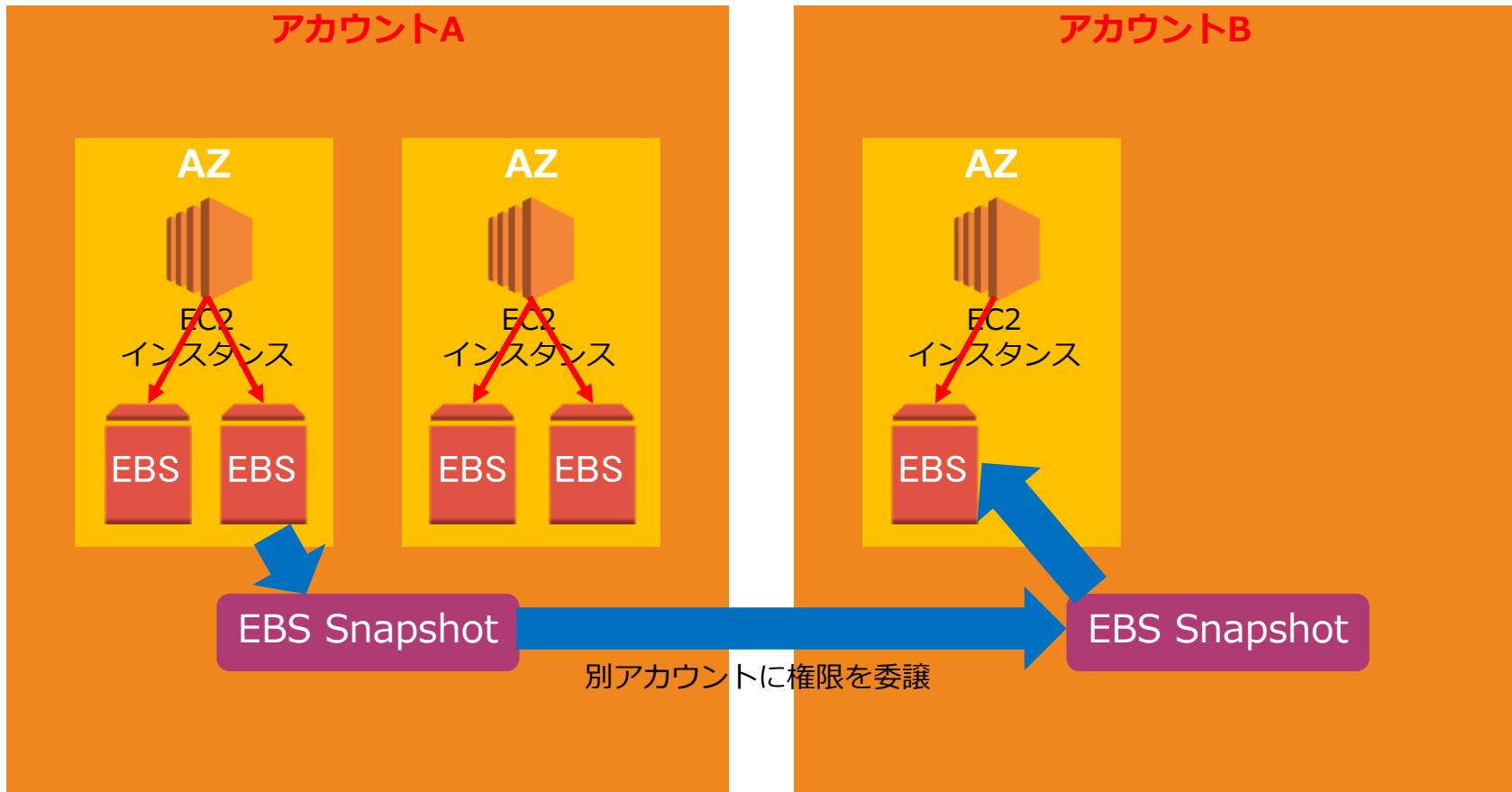
# スナップショットの共有

スナップショットはリージョン間を跨いで利用可能



# スナップショットの共有

スナップショットは権限を変更することで、他のアカウントに移譲することが可能



# スナップショットとAMI

Amazon Machine ImageはOS設定のイメージであり、  
Snapshotはストレージのバックアップとなる

AMI

- ✓ EC2インスタンスのOS設定などをイメージとして保持して、新規インスタンス設定に転用するもの
- ✓ 仮想サーバーのバックアップ

Snapshot

- ✓ ストレージ（EBS）のその時点の断面のバックアップとして保持するもの
- ✓ ストレージの復元や複製に利用

# [Q] EBSボリュームの削除

研究チームではデータ解析にEC2インスタンスを利用しています。日々収集されるデータをEBSボリュームがアタッチされたEC2インスタンスでバッチジョブとして分析ワークフローを実施します。分析の実行中に、チームはEC2インスタンスを終了すると、接続されているEBSボリュームも失われることを発見しました。

この問題に関する最も可能性が高い原因は何でしょうか？

- 1) EC2インスタンスがインスタンストアベースAMIで作成されているため、ルートボリュームがデータを一時的にしか保存できない。
- 2) EBSボリュームのスナップショット取得を実施していないため、終了時にデータを保持できなかった。
- 3) EC2インスタンスの終了時に、EBSボリュームの保護をチェック入れていないため、EBSボリュームを同時に削除してしまった。
- 4) EBSボリュームがEC2インスタンスのルートボリュームとして設定されているとインスタンスの終了時のデフォルトの動作では、接続されているルートボリュームも終了する。

# EBSボリュームの削除

EC2インスタンスの削除と共にEBSは削除されるため、データを保持したい場合は設定変更が必要

## ルートボリュームのEBS

- ✓ EBS-backed AMIインスタンスにはルートボリュームにEBSが利用されている。
- ✓ デフォルト設定ではEC2インスタンスの削除と共にEBSボリュームも削除される。

## DeleteOnTermination 属性

- ✓ DeleteOnTermination属性を有効化しているとEC2インスタンスの削除に応じてEBSも削除される
- ✓ 非有効化することでEBSボリュームのみ保持可能

### ▼ ブロックデバイス

ブロックデバイス					
プロトコル					
Amazon EBS	Amazon EBS	Amazon EBS	Amazon EBS	Amazon EBS	Amazon EBS
8	アタッチ済み	Sat Oct 15 2022 22:39:02 G...	いいえ	-	終了時に削除 はい

# [Q] EBSの暗号化

研究機関ではデータ解析にEC2インスタンスを利用しています。日々収集されるデータをEBSボリュームがアタッチされたEC2インスタンスでバッチジョブとして実施します。これらのデータは非常に機密性が高いため、EBSに保存されている機密データはHIPAAコンプライアンス基準を満たす必要があります。

暗号化されたEBSボリュームの正しい説明はどれでしょうか？（3つ選択してください）

- 1) ボリューム内に保存されるデータは暗号化される
- 2) ボリュームのスナップショットは暗号化される
- 3) ボリュームとインスタンス間を移動するデータは暗号化されていない。
- 4) ボリュームとインスタンス間を移動するデータ暗号化にはSSL証明書が必要である。
- 5) ボリュームのスナップショットは別途スナップショットの暗号化を実施する必要がある。

# EBSの暗号化

EBSはKMSのCMKを利用して、ボリューム作成時とスナップショット作成時に暗号化を実施する。

## EBSの暗号化

- ✓ EBSボリュームやスナップショット作成時 AWS KMSの カスタマーマスターキー (CMK) を使用して暗号化を実施
- ✓ インスタンスとそれに接続された EBSストレージ間のデータ転送と保存データの両方に対して暗号化を実施する。

## 暗号化対象

- ✓ ボリューム内の保存データ
- ✓ ボリュームとインスタンスの間の転送データ
- ✓ ボリュームから作成されたすべてのスナップショット
- ✓ それらのスナップショットから作成されたすべてのボリューム

# [Q] EBSのステータス

あなたはAWSアカウントを作成して、新規にEC2インスタンスを起動しました。起動したEC2インスタンスを確認すると、EC2のステータスチェックが不十分なデータ（Insufficient Data）と表示されています。

最も可能性の高い原因はどれでしょうか？（2つ選択してください。）

- 1) ボリュームチェックが進行中である。
- 2) EBSがボリューム制限を超過している。
- 3) ボリュームのチェックに失敗した。
- 4) ボリュームには十分なデータがない。

# EBSのステータス

EBSは次の4つのステータス表示を理解することが必要

ボリュームのステータス	I/O 有効ステータス	I/O パフォーマンスステータス (プロビジョンド IOPS ボリュームでのみ使用可能)
ok	Enabled (I/O Enabled または I/O Auto-Enabled)	Normal (ボリュームパフォーマンスは想定どおり)
warning	Enabled (I/O Enabled または I/O Auto-Enabled)	Degraded (ボリュームのパフォーマンスが想定を下回っている) Severely Degraded (ボリュームのパフォーマンスが想定をかなり下回っている)
impaired	Enabled (I/O Enabled または I/O Auto-Enabled)  Disabled (ボリュームがオフラインで復旧の保留中、またはユーザーによる I/O の有効化待ち)	Stalled (ボリュームのパフォーマンスは致命的な影響を受けている)  Not Available (I/O が無効なため、I/O パフォーマンスの判定不能)
insufficient-data	Enabled (I/O Enabled または I/O Auto-Enabled)  Insufficient Data	Insufficient Data ステータスチェックがまだ進行している場合も

Reference: [https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/monitoring-volume-status.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/monitoring-volume-status.html)

# [Q]EBSのRAID構成

あなたはソリューションアーキテクトとして、EC2インスタンスのWebサーバーとデータベースを利用して業務アプリケーションを構築しました。リレーショナルデータベースをホストするために、1つの500 GB EBSボリュームを持つ大規模なEC2インスタンスを使用しています。パフォーマンスを確認すると、データベースへの書き込みスループットを向上させる必要があることが判明しました。

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) EC2インスタンスのサイズを増やす。
- 2) 2つ以上のEBSボリュームを利用したRAID0構成を設定する
- 3) 2つ以上のEBSボリュームを利用したRAID1構成を設定する
- 4) EC2インスタンスをクラスター・プレイスメント・グループに設置する。
- 5) PV AMIを利用してEC2インスタンスを再起動して、拡張ネットワークを有効化する。

# EBSのRAID構成

パフォーマンス向上と冗長化を高める目的で、EBSでは主に RAID0とRAID1の構成が実施される

## RAID 0 (ストライピング)

- ✓ 目的：パフォーマンスを向上させる。
- ✓ RAID0は、2つ以上のボリュームを1台のボリュームに合わせてパフォーマンスを向上させる。
- ✓ ボリュームを合わせたパフォーマンスになる。

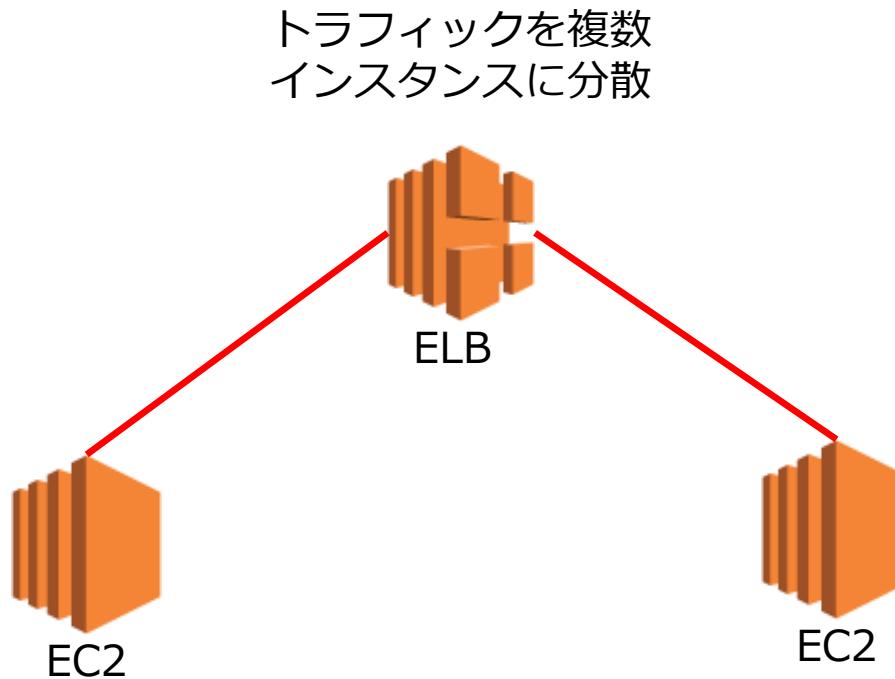
## RAID 1 (ミラーリング)

- ✓ 目的：ボリュームの冗長性を高める。
- ✓ RAID 1 では複数のボリュームに同時にデータを書き込むため、片方が壊れても、片方にデータが残る。

## ELBの出題範囲

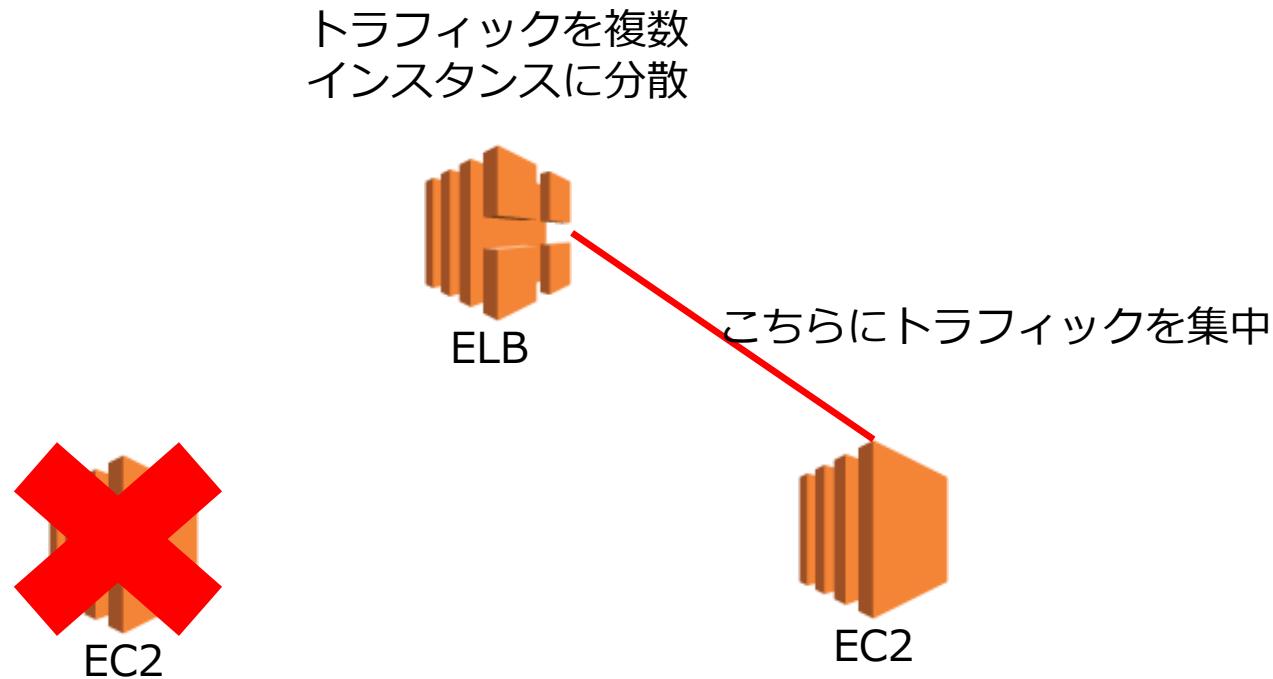
# ELBとは何か？

ELBは複数のEC2インスタンスで処理を可能にするロードバランサーを提供するサービス



# ELBとは何か？

EC2インスタンスのヘルスチェックを行い、正常なインスタンスのみを利用することも



# ELBの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

ELBの特徴	✓ Route53との違いも含めて、 ELBの利用方法や特徴に関する質問が問われる
ELBの構成	✓ ELBを利用した基本的なアーキテクチャ構成に関する質問が出題される。 ✓ セキュリティ要件に基づいてインターナルELBを利用した構成方法が問われる。
ELBタイプの選択	✓ シナリオに基づいてELBを利用するべき要件が説明されて、 どのELBタイプを利用するべきかが問われる。
ALBの特徴	✓ ALBの機能や他のELBタイプとの違いに関する質問が出題される。
NLBの特徴	✓ NLBの機能や他のELBタイプとの違いに関する質問が出題される。

# ELBの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

クロスゾーン負荷分散	✓ ELBのクロスゾーン負荷分散に関する特徴やユースケースに関する質問が出題される。
暗号化	✓ ELBの暗号化の設定方法に関する質問が出題される。
ステイッキーセッション	✓ ELBのステイッキーセッションに関する特徴やユースケースに関する質問が出題される。
Connection Draining	✓ ELBのConnection Drainingに関する特徴やユースケースに関する質問が出題される。

# [Q] ELBの特徴

あなたの会社は複数部門でAWSを利用しておあり、アプリケーション毎にVPCを設定しています。あなたはソリューションアーキテクトとして、アプリケーション間連携機能を実装しています。その実装には、それぞれのVPCをピアリング接続して単一のELBを使用して、同じリージョン内のピアリングされたVPC内の複数のEC2インスタンスにトラフィックをルーティングしたいと考えています。

このようなELBの構成をどのように達成することができますか？

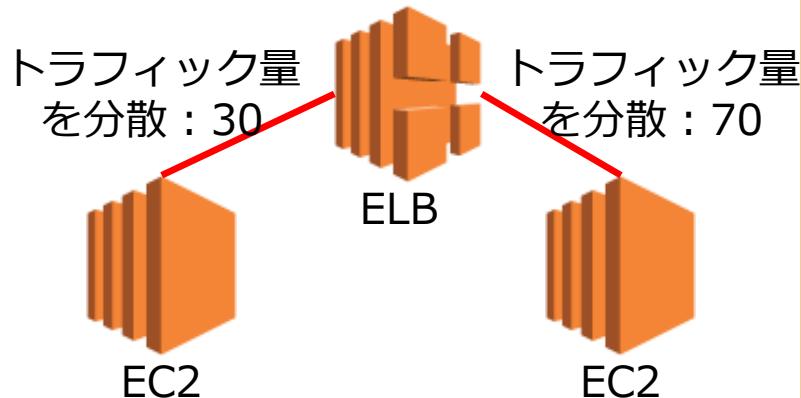
- 1) VPCピアリングを構成した上で、全てのELBタイプでVPCを跨いだ構成を実現できる。
- 2) この要件にはELBではなくRoute53を使用する必要がある。
- 3) ELBは複数のVPC間を跨いだトラフィック分散構成ができない。
- 4) VPCピアリングを構成した上で、NLBまたはALBによりIPアドレスをターゲットとして利用して、別VPCのインスタンスを指定できる。

# ELBの特徴

負荷分散によるスケーラビリティとヘルスチェックによる高可用性を実現

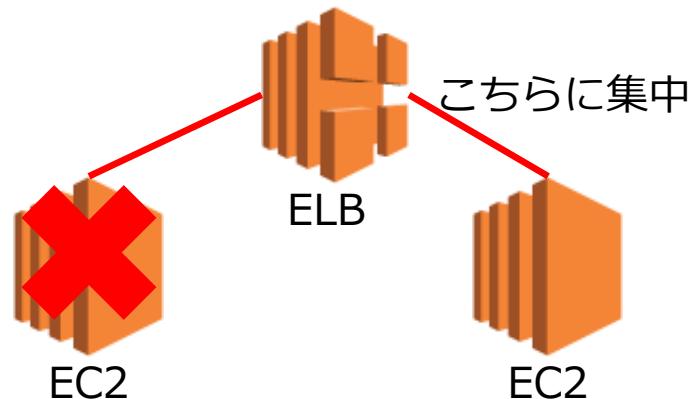
## スケーラビリティの確保

複数のEC2インスタンス/ECSコンテナの負荷分散



## 高可用性

複数のアベイラビリティゾーンにある複数のEC2インスタンスの中から正常なターゲットに振り分ける。



# ELBの特徴

EC2インスタンスの処理を分散する際に標準的に利用するマネージド型のロードバランシングサービス

- インスタンス間の負荷を分散するサービス。インスタンスに限らずIPアドレスをターゲットにした負荷分散も可能である。
- ヘルスチェックにより異常なインスタンスを認識してトラフィックを正常なインスタンスのみに分散させる。
- パブリックサブネット／プライベートサブネットで使用可能
- 負荷に応じてキャパシティを自動増減するスケーリングを実施するが、これはAWS側でマネージドサービスとして実施される。
- 時間に応じたロードバランサー・キャパシティ・ユニット (LCU) 使用量に課金 (CLBのみ転送データ単位)
- セキュリティグループで通信可能なプロトコルを制御する。
- ログを有効化することで、ログをS3バケットに保存する。

# [Q] ELBの構成

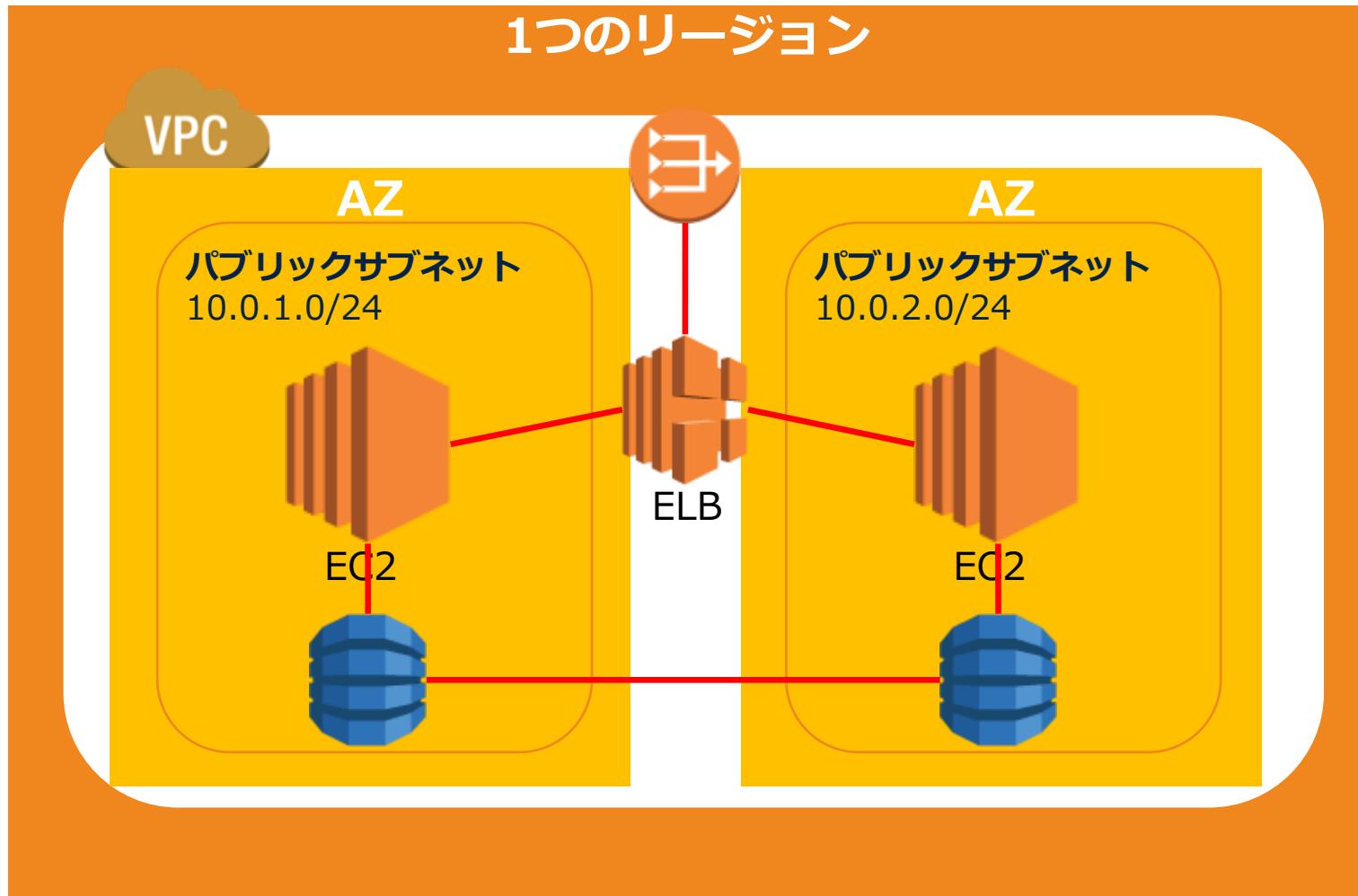
シンガポールにあるベンチャー企業はAWSを利用して新サービス用のアプリケーションを構築しています。このアプリケーションはWEBサーバーに4つのEC2インスタンスを設置して、さらにELBのターゲットグループを構成することが必要です。

次の中で実現できない構成はどれでしょうか？

- 1) ELBのターゲットグループにおいて、4つのインスタンスを全てシンガポールリージョンの2つのアベイラビリティゾーンにデプロイする。
- 2) ELBのターゲットグループにおいて、4つのインスタンスをシンガポールリージョンのAZ-aにデプロイする。
- 3) ELBのターゲットグループにおいて、4つのインスタンスは全てシドニーリージョンのAZ-bにデプロイする。
- 4) ELBのターゲットグループにおいて、2つのインスタンスをシンガポールリージョンのAZ-aにデプロイし、他の2つのインスタンスは、シドニーリージョンのAZ-bにデプロイする。

# ELBの構成

ELBを利用したマルチAZにインスタンスへのトラフィックを分散する構成が利用される。ELBはリージョンを跨げない



## [Q] ELBの構成

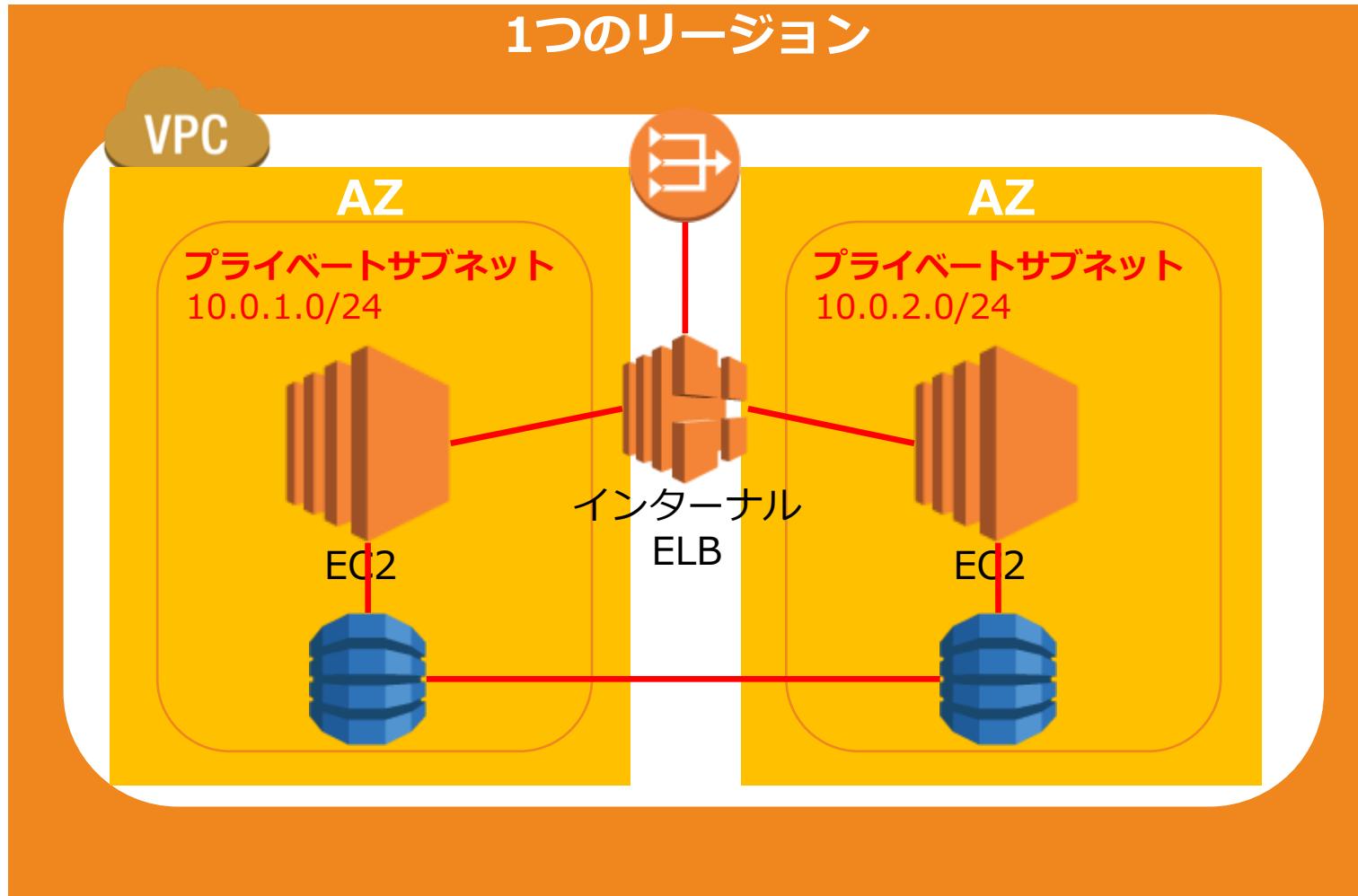
医療企業はAWSを利用して新サービス用の医療データ共有アプリケーションを構築しています。このアプリケーションはWEBサーバーにEC2インスタンスを利用して、データレイヤーにはS3とRDSを利用しています。負荷を分散するには、インターネット向けのALBを構成する必要があります。医療データを取り扱うためパブリックなアクセスを制限することが必要です。

この構成を機能させるために必要な構成を選択して下さい。 (2つ選択してください)

- 1) 同じAZに対応するパブリックサブネットを作成してALBに関連付ける。
- 2) インターネットゲートウェイをプライベートサブネットに接続する
- 3) プライベートサブネット内の各EC2インスタンスにElastic IPアドレスを追加する。
- 4) プライベートサブネットにNATゲートウェイを設置する。
- 5) プライベートサブネットにRDSとEC2インスタンスを設置してアプリケーション用のサーバーとする。

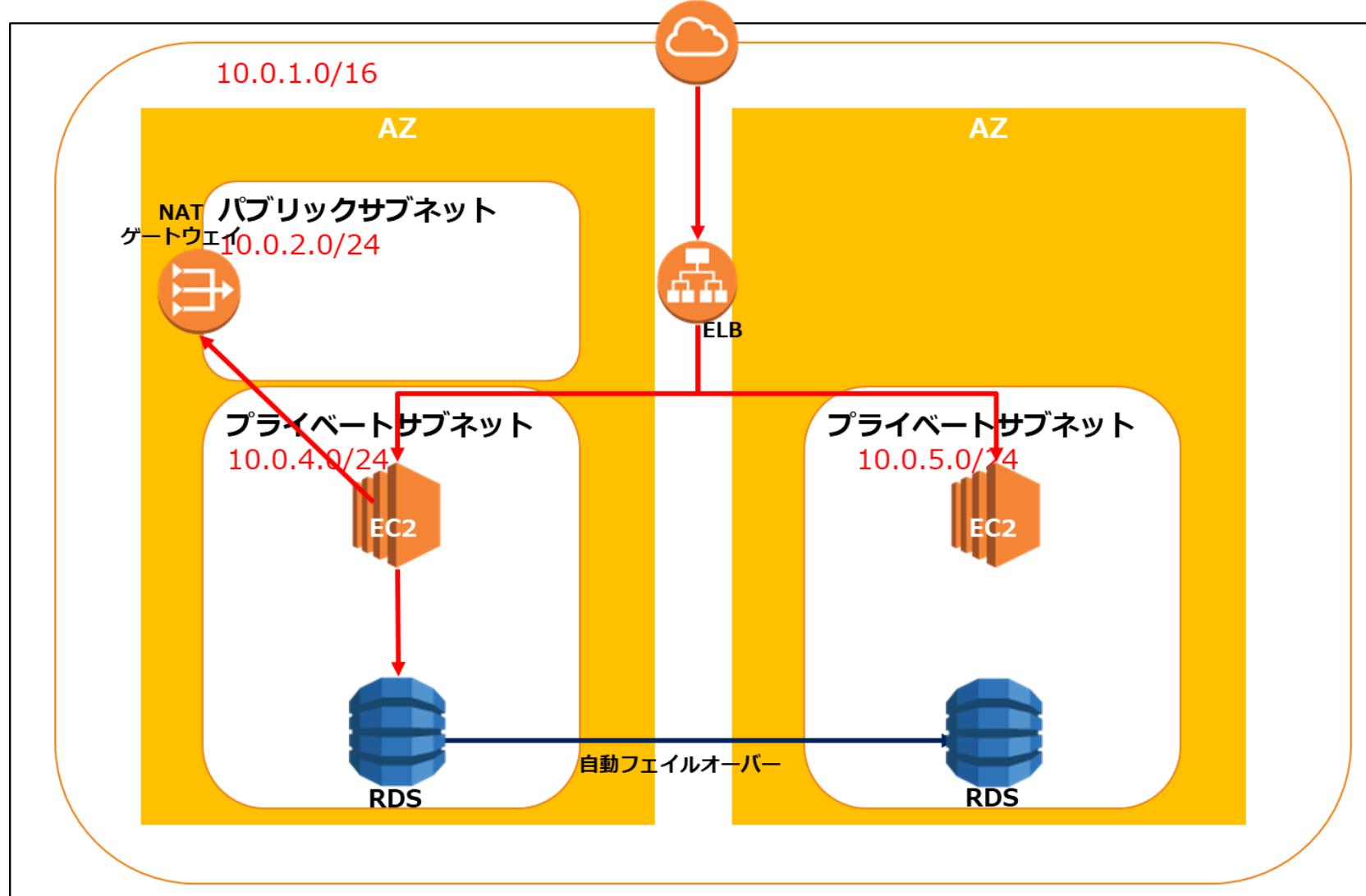
# ELBの構成

プライベートサブネット空間にもELBを利用することが可能



# ELBの構成

プライベートサブネットに対してパブリックネットワークとつながったELBを構成してトラフィック分散させることも可能



# [新Q] ELBタイプの選択

動画配信サイトを展開しているA社は、コンテンツを世界中のユーザーに配信するためにAWSクラウドを利用することを検討しています。この動画配信サイトは世界中にユーザーを抱えており、毎秒少なくとも100万件のリクエストをサポートすることが要件となっています。

この要件を満たすために、どのELBタイプを利用するべきでしょうか？

- 1) Application Load Balancer
- 2) Classic Load Balancer
- 3) Gateway Load Balancer
- 4) Network Load Balancer

# ELBのタイプ

現在利用できるロードバランサーは3タイプで用途に応じて使い分ける

## Classic Load Balancer (CLB)

レイヤー4と7に対応しており、TCP,SSL,HTTP,HTTPSリスナーを利用  
古いタイプなのでALB/NLBの利用を優先する。  
データ転送（GB単位）に応じて課金される。  
IPアドレスが可変であるため、指定時にDNSのみ利用可能

## Application Load Balancer (ALB)

レイヤー7に対応しHTTP／HTTPSリスナーに対応  
パスルーティングが利用可能  
時間に応じたロードバランサーキャパシティーユニット (LCU) の使用量で  
課金される。  
IPアドレスが可変であるため、指定時にDNSのみ利用可能  
デフォルトでクロスゾーン負荷分散が有効

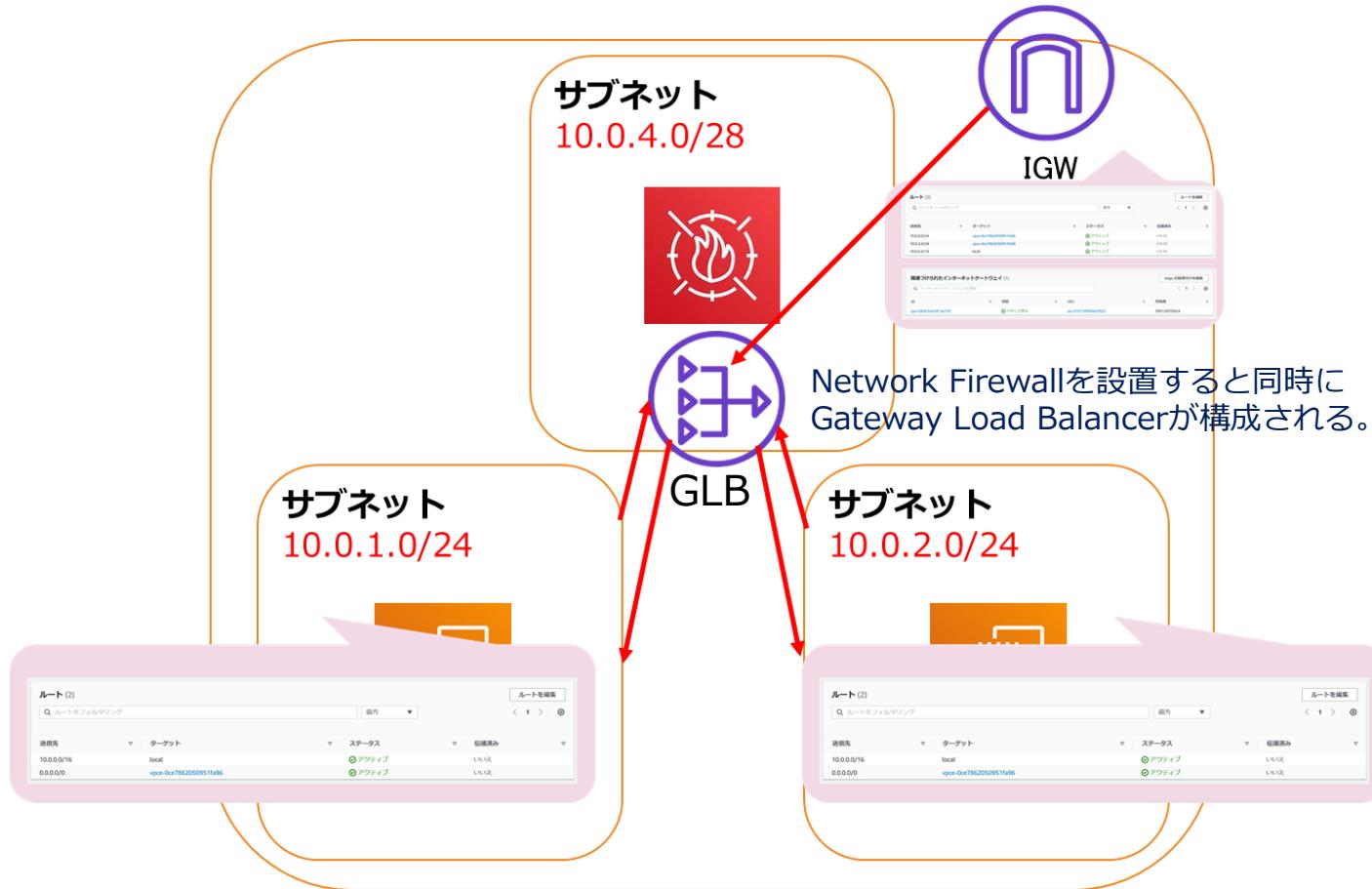
## Network Load Balancer (NLB)

- L4 NATロードバランサでTCPリスナーに対応（戻りトラフィックがNLBを  
経由しない）

時間に応じたLCU の使用量で課金される。  
NLB のサブネット拡張サポート（サブネットを追加できる）  
固定IPのためDNSとIPのどちらも利用可能  
ALBよりも高パフォーマンス処理が可能  
デフォルトでクロスゾーン負荷分散が無効

# Gateway Load Balancer

Gateway Load BalancerはNetwork Firewall用の特別なポートバランサーで、ファイアーウォールへのルートを管理する。



# [Q] ALBの特徴

ベンチャー企業はAWSを利用して新サービス用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WEBサーバーに4つのEC2インスタンスを利用して、ALBのターゲットグループを構成することが必要です。さらに開発チームは、HTTPヘッダーのURLパスに基づいて、トラフィックを複数のバックエンドサービスにルーティングして次のルーティングを設定します。

https://www.pintor.com/indexのリクエストをマイクロサービスAへ

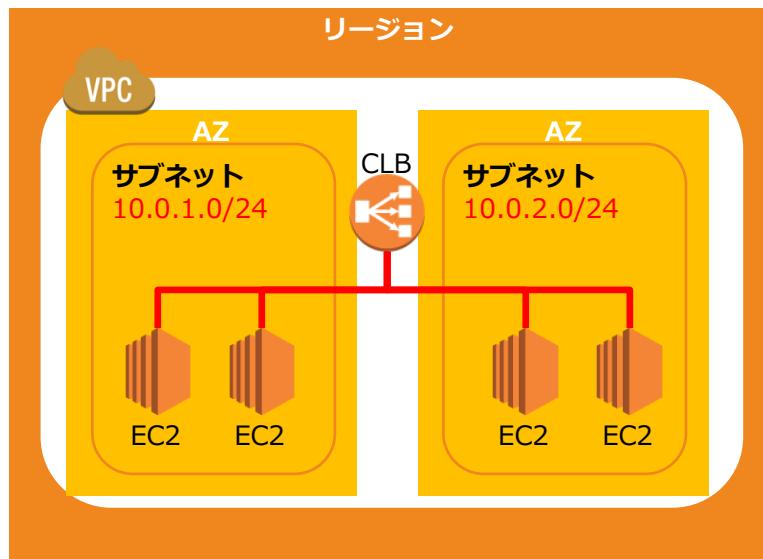
https://www.example.com/headのリクエストをマイクロサービスBへ

このような要件を満たす設定方法はどれでしょうか？

- 1) NLBのクエリ文字列パラメータベースのルーティングを利用する
- 2) ALBのHTTPヘッダーベースのルーティングを利用する
- 3) Route53の加重ルーティングを利用する。
- 4) ALBのパスベースルーティングを使用する。

# CLB (Classic Load Balancer )

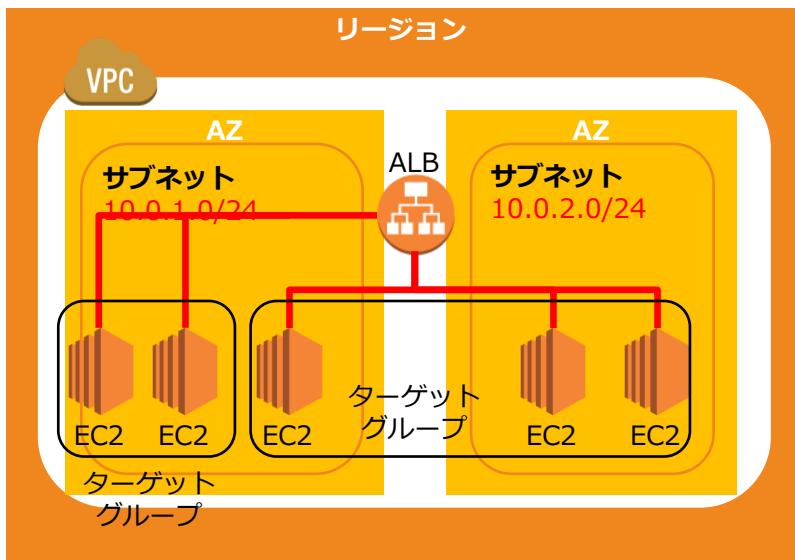
初期のELBタイプであり、標準的なL4／L7におけるロードバランシングが可能だが、複雑な設定はできない



- HTTP/HTTPSとTCP/SSLプロトコルのL4とL7に対応
- Proxyプロトコルによる発信元IPアドレス識別
- ELBとバックエンドのEC2インスタンス間でHTTPS/SSL使用時にサーバ証明書認証を実施
- CLB配下のインスタンスは、全て同一の機能を持ったインスタンスである必要がある。
- リクエスト内容を確認して分散先を振り分ける  
コンテンツベースルーティングは出来ない

# ALB (Application Load Balancer )

レイヤー7の対応が強化された単一ロードバランサーで、異なるアプリケーションへリクエストをルーティングが可能

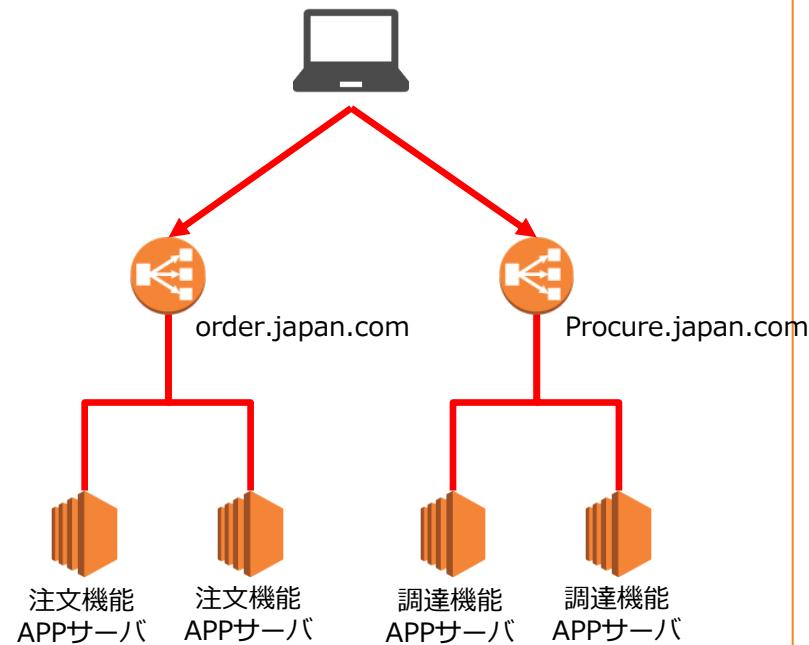


- レイヤー7に対応しHTTP／HTTPSリスナー対応
- WebSocketとHTTP/2のリクエストを受付
- 1インスタンスに複数ポートを登録可能
- 複数ポートを個別のターゲットとして登録するこ  
とが可能なため、ポートを利用するECSなどのコ  
ンテナをロードバランシング可能
- ターゲットグループでのヘルスチェックが可能
- EC2と同様に削除保護が可能
- 加重ロードバランシングが利用可能
- リクエスト内容を確認して分散先を振り分けるコ  
ンテントベースルーティングが可能
- URLのパスに基いてルーティングが可能なパス  
ベースルーティングが可能

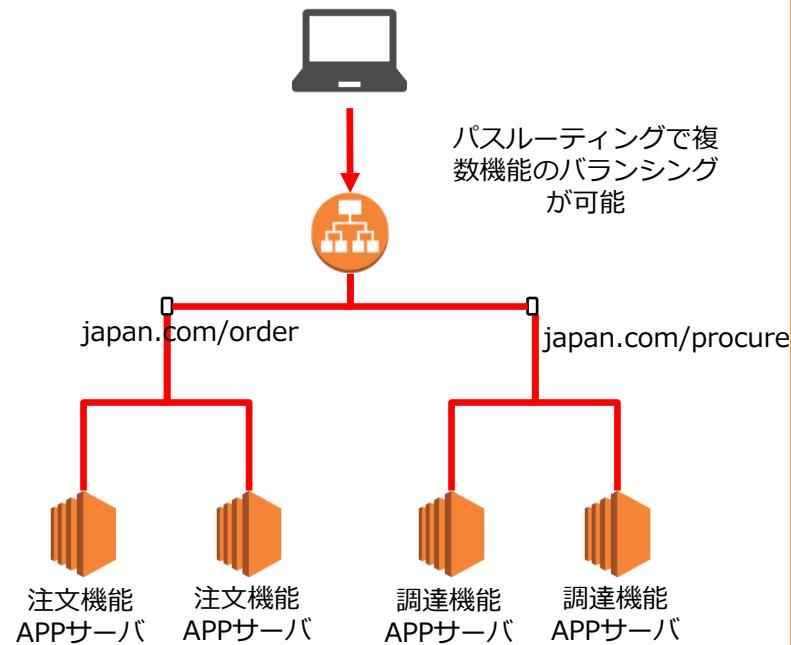
# CLBとALB

ALBはパスベースルーティングによりリクエスト内容に応じて機能毎にバランスシングすることが可能

## CLBの複数機能バランスシング



## ALBの複数機能バランスシング



# [Q] NLBの特徴

動画配信サイトを展開しているA社は、コンテンツを世界中のユーザーに配信するためにAWSクラウドを利用することを検討しています。この動画配信サイトは世界中にユーザーを抱えており、毎秒少なくとも100万件のリクエストをサポートすることが要件となっています。エンジニアリングチームは、パブリックサブネットに複数のインスタンスをプロビジョニングし、これらのインスタンスIDをNLBのターゲットとして指定しました。

NLBに設定したターゲットインスタンスの正しいルーティング方式を説明して下さい。

- 1) トラフィックはプライマリプライベートIPアドレスを使用してインスタンスをルーティングされる。
- 2) トラフィックはプライマリパブリックIPアドレスを使用してインスタンスにルーティングされる。
- 3) トラフィックはDNS名を使用してインスタンスにルーティングされる。
- 4) トラフィックはElastic IPアドレスを使用してインスタンスにルーティングされる。
- 5) トラフィックはインスタンスIDを使用してインスタンスにルーティングされる。

# NLB (Network Load Balancer)

NLBは超低遅延で高スループットを維持しながら秒間何百万リクエストを捌けるように設計された高性能ロードバランサー

- L4 NATロードバランサでTCPリスナーに対応（戻りトラフィックがNLBを経由しない）
- 振発性ワークロードを処理し、毎秒数百万のリクエストに対応できる能力
- VPC外のターゲットを含めたIP アドレスや静的IPアドレスでの登録可能
- 複数のポートで各インスタンスまたは IP アドレスを同ターゲットグループに登録可能
- 大規模アクセスが予測される際にCLBやALBでは必要な事前申請が不要
- ALBやCLBはX-Forwarded-Forでアクセス元IPアドレスを判断するが、NLBは送信元IPアドレスと送信元ポートを書き換えないため、パケットからアクセス元を判断可能
- NLBはフルトトレランス機能を内蔵したコネクション処理を持ち、数カ月から数年のオープンなコネクションを処理できる
- ECSなどによりコンテナ化されたアプリケーションのサポート
- 各サービスの個別のヘルステータスのモニタリングのサポート
- NLB のサブネット拡張サポート（サブネットを追加できる）

# ELBの主要機能

## ELBのロードバランシングの際に様々な機能を利用

ヘルスチェック	EC2インスタンスの正常／異常を確認し、利用するEC2の振り分けを行う
クロスゾーン 負荷分散	配下のEC2の負荷に応じて、複数のAZに跨るEC2インスタンスに均等に負荷分散を行う
リスナー設定	ELBはリスナー設定によって、通信するプロトコルタイプや通信ルールを設定することができる。
暗号化通信	SSL/TSL証明書をELBに設定することでHTTPSまたはTLS通信を実施することができる。
ステイツキー セッション	セッションを継続して、同じクライアント端末からのリクエストを継続して同じEC2インスタンスに送信する。
Connection Draining	インスタンスに異常が発生した場合に、そのバックエンドインスタンスへの指定した秒数の間は通信が切れずに、処理中のリクエストが終わるまで一定期間待ってくれる



# [新Q] ヘルスチェックの設定

ある会社は複数のAmazon EC2インスタンスにホストされているウェブアプリケーションを構築しています。これらのEC2インスタンスはNLBのターゲットグループに属しており、さらにAuto Scalingグループを使用するように構成されています。

ソリューションアーキテクトはカスタムスクリプトやカスタムコードを記述せずに、このアプリケーションの可用性を向上させるように依頼されました。また、アプリケーションに対するHTTPエラーをNLB上で検出して、これらのエラーが発生した場合、ウェブサービスを実行するEC2インスタンスを自動で再開する必要があります。

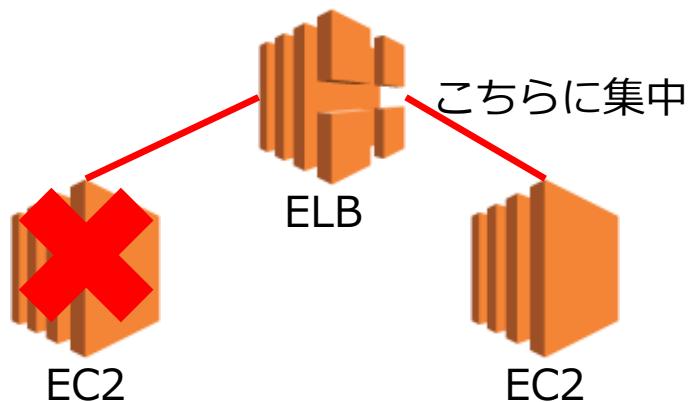
この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。

- 1) アプリケーションのログを1分ごとに評価するRoute53のヘルスチェックを有効化する。HTTPエラーが検出された場合、アプリケーションが再開始する。
- 2) アプリケーションのログを1分ごとに評価するcronジョブをEC2インスタンスに追加する。HTTPエラーが検出された場合、インスタンスが再開始するように設定する。
- 3) このアプリケーションのURLを指定して、NLBのHTTPヘルスチェックを有効化する。異常状態のインスタンスを置換するように、Auto Scalingアクションを構成する。
- 4) NLBのUnhealthyHostCountメトリクスをモニタリングするAmazon CloudWatchアラームを作成する。アラームがALARM状態になると異常状態のインスタンスを置換するように、Auto Scalingアクションを構成する。

# ヘルスチェックの設定

EC2インスタンスの正常/異常をチェックする機能。これにより、正常なインスタンスのトラフィックを振り分ける。

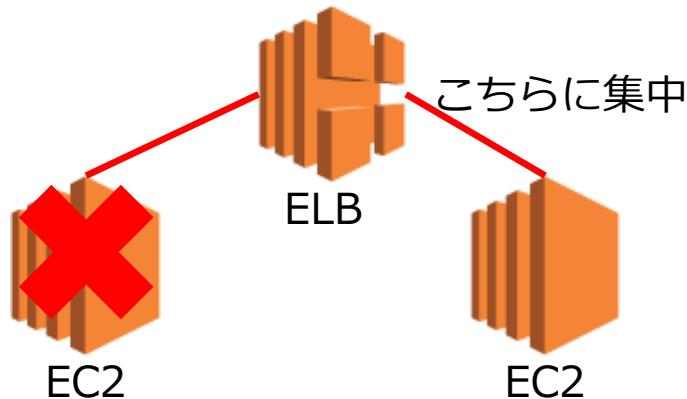
ヘルスチェックによる正常なインスタンスへのトラフィック分散



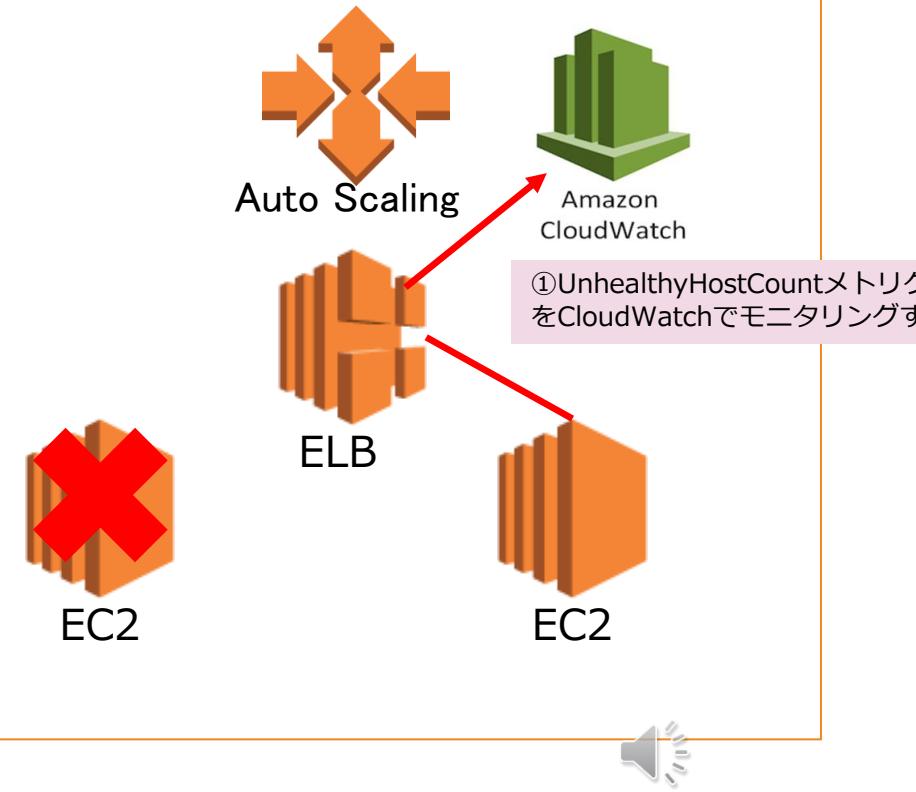
# ヘルスチェックの設定

ヘルスチェックに基づいて負荷分散やAuto Scalingなどを実施できる。

ヘルスチェックによる正常なインスタンスへのトラフィック分散



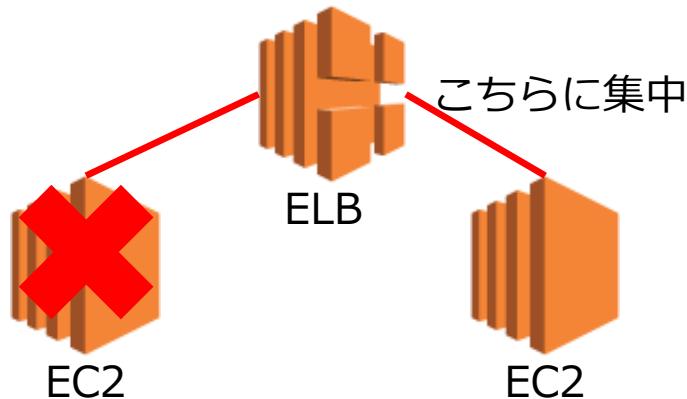
ヘルスチェックをCloudWatchに連携して、Auto Scalingを実施する。



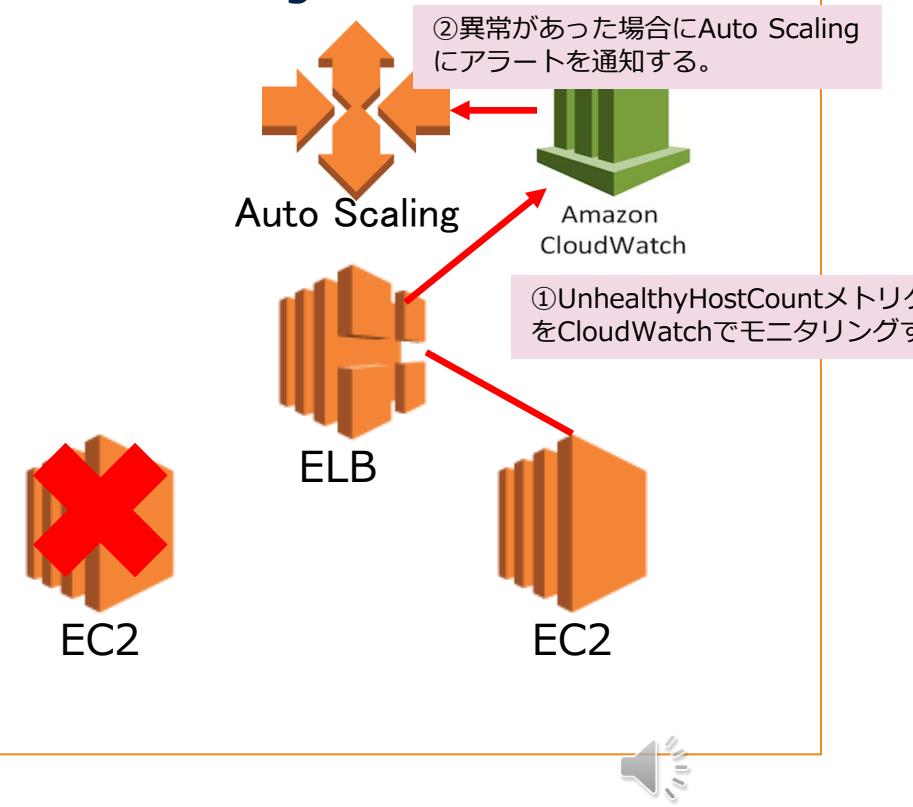
# ヘルスチェックの設定

ヘルスチェックに基づいて負荷分散やAuto Scalingなどを実施できる。

ヘルスチェックによる正常なインスタンスへのトラフィック分散



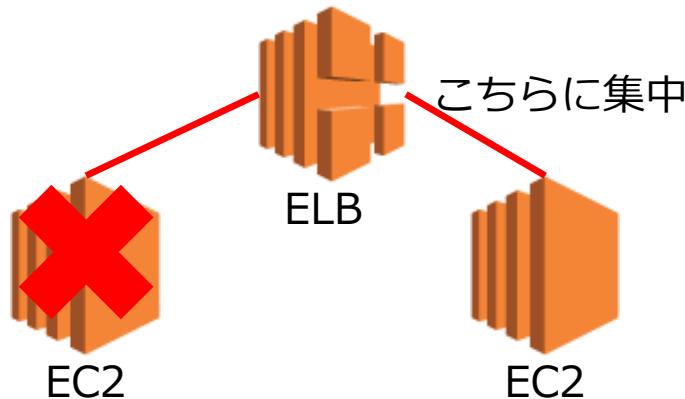
ヘルスチェックをCloudWatchに連携して、Auto Scalingを実施する。



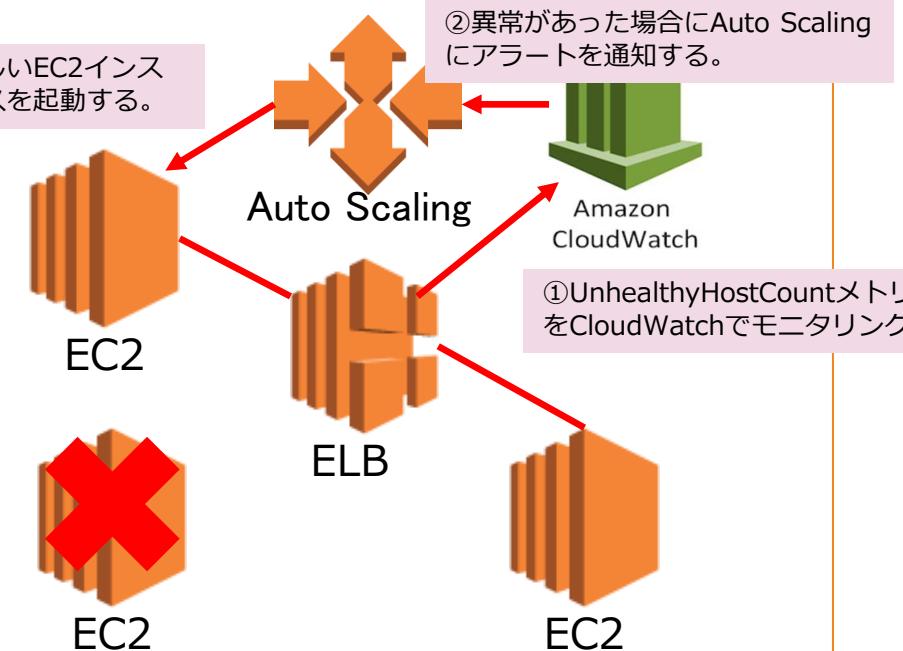
# ヘルスチェックの設定

ヘルスチェックに基づいて負荷分散やAuto Scalingなどを実施できる。

ヘルスチェックによる正常なインスタンスへのトラフィック分散



ヘルスチェックをCloudWatchに連携して、Auto Scalingを実施する。



# [Q]クロスゾーン負荷分散

大手スーパー・マーケットチェーンはECアプリケーションを運用しています。冗長構成をするために4つのEC2インスタンスをAZ-aに1つのインスタンスをAZ-bに3つのインスタンスを展開して、ELBを利用したトラフィック制御を行っています。

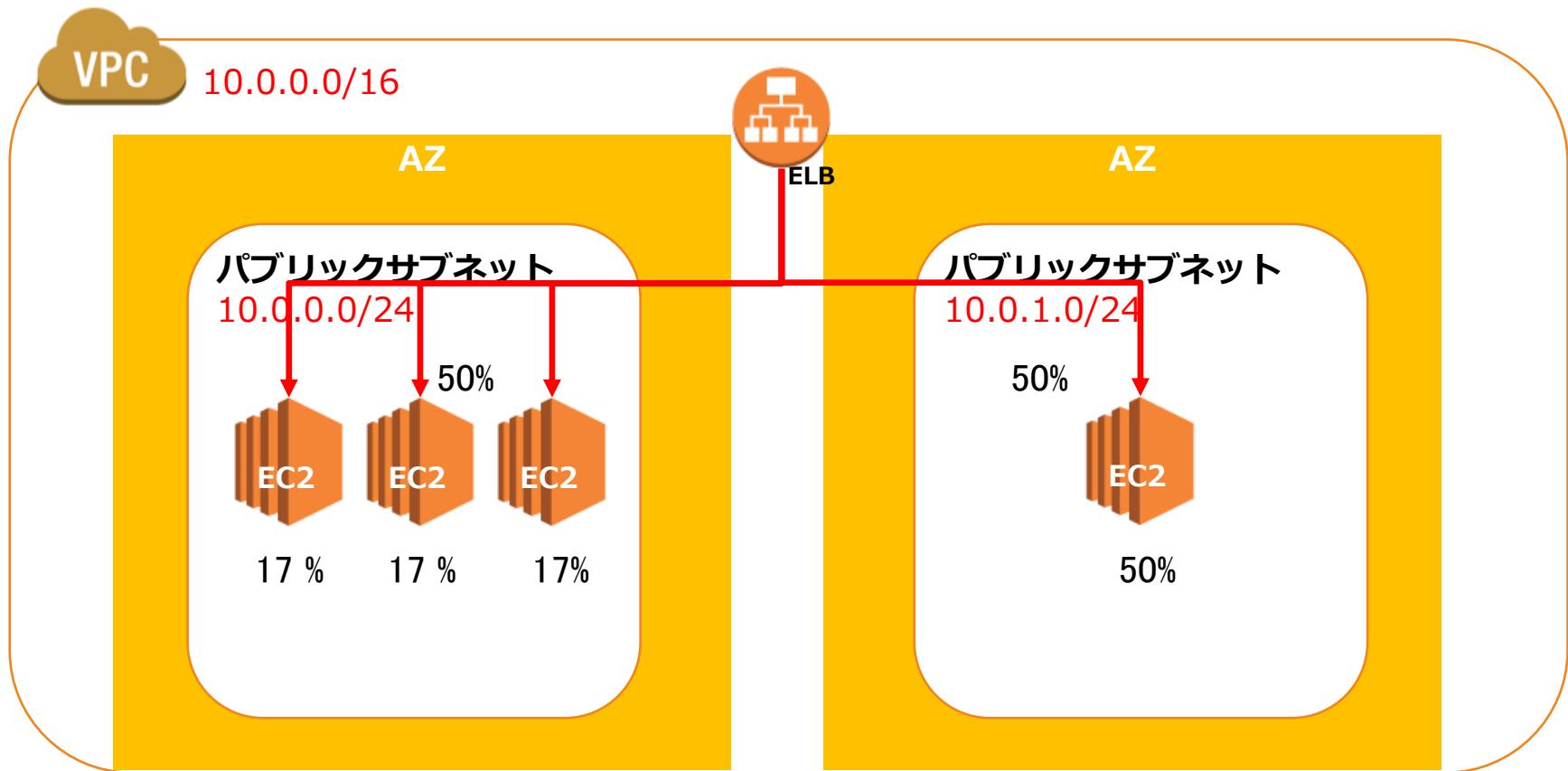
この構成でクロスゾーン負荷分散を実施している場合と、実施していない場合のラフィック分散の結果はどうなりますか？

- 1) クロスゾーン負荷分散を有効にすると、AZ-aの1つのインスタンスが50%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ17%のトラフィックを受信します。クロスゾーン負荷分散を無効にすると、AZ-aの1つのインスタンスが25%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ25%のトラフィックを受信します。
- 2) クロスゾーン負荷分散を有効にすると、AZ-aの1つのインスタンスが25%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ17%のトラフィックを受信します。クロスゾーン負荷分散を無効にすると、AZ-aの1つのインスタンスが25%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ25%のトラフィックを受信します。
- 3) クロスゾーン負荷分散を有効にすると、AZ-aの1つのインスタンスが25%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ25%のトラフィックを受信します。クロスゾーン負荷分散を無効にすると、AZ-aの1つのインスタンスが50%のトラフィックを受信し、AZ-bの4つのインスタンスがそれぞれ約17%のトラフィックを受信します。
- 4) クロスゾーン負荷分散を有効にすると、AZ-aの1つのインスタンスが90%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ10%のトラフィックを受信します。クロスゾーン負荷分散を無効にすると、AZ-aの1つのインスタンスが10%のトラフィックを受信し、AZ-bの3つのインスタンスがそれぞれ30%のトラフィックを受信します。



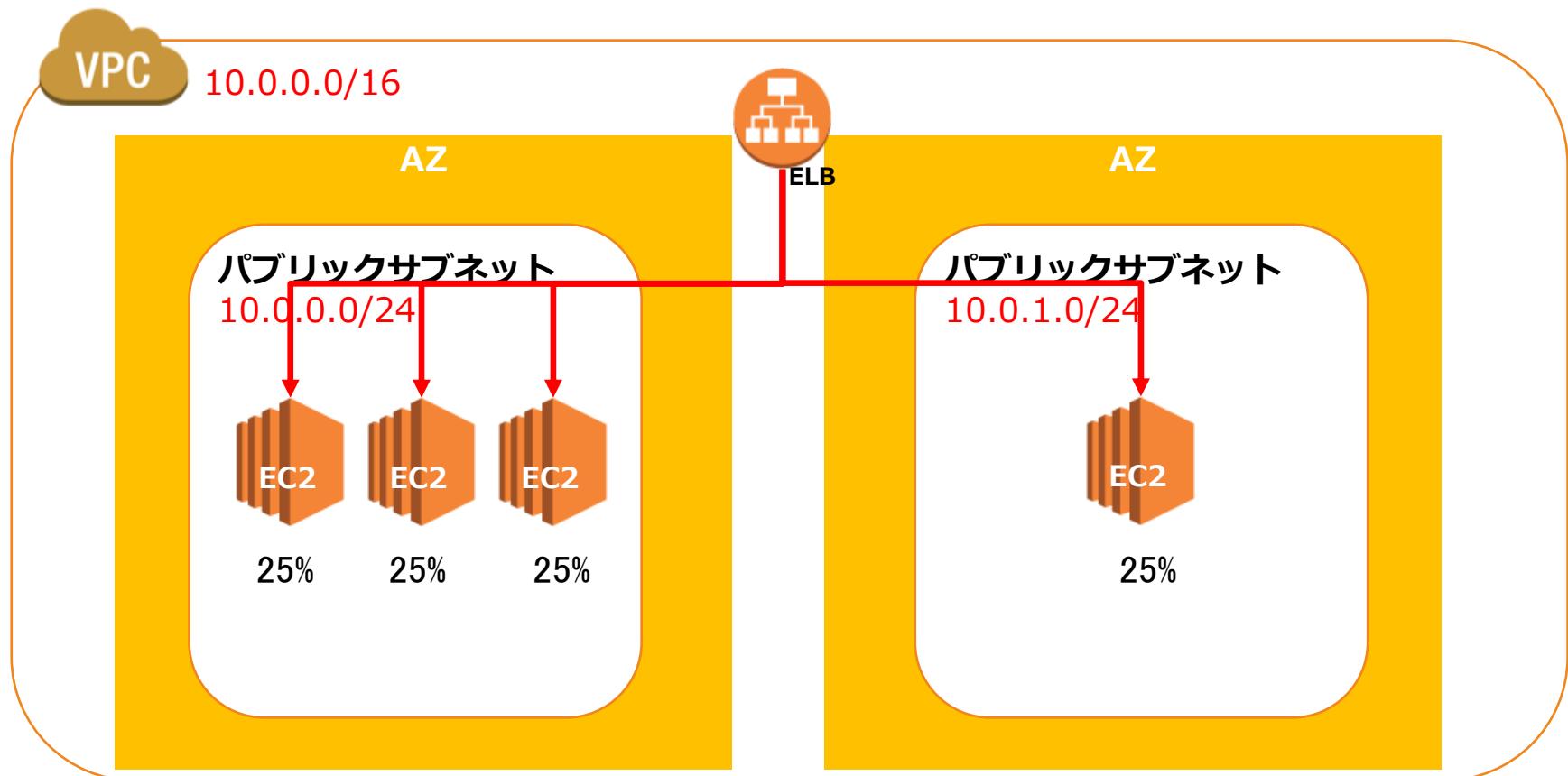
# クロスゾーン負荷分散

クロスゾーン負荷分散が無効にされていると、ゾーン毎に均等に負荷が分散される。



# クロスゾーン負荷分散

クロスゾーン負荷分散が有効化されると、ゾーンを跨いでインスタンスに均等に負荷が分散される。

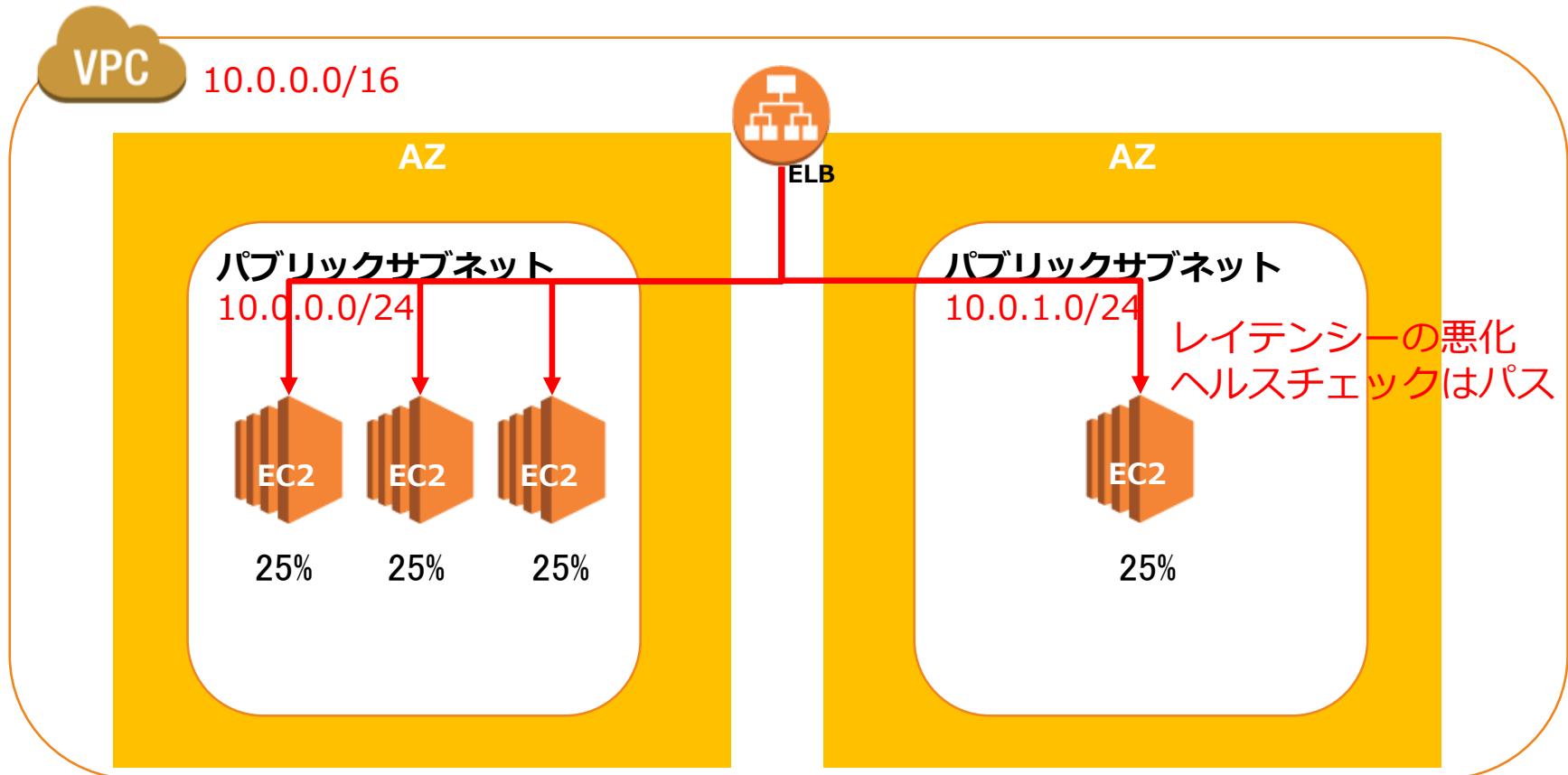


- ✓ ALBではデフォルト有効（無効化も可能）/有効時のAZ間のデータ転送無料
- ✓ NLBではデフォルト無効/有効時のAZ間のデータ転送有料
- ✓ デメリットは、ゾーンごとに偏りが発生すること。スケーリングのアンバランス化



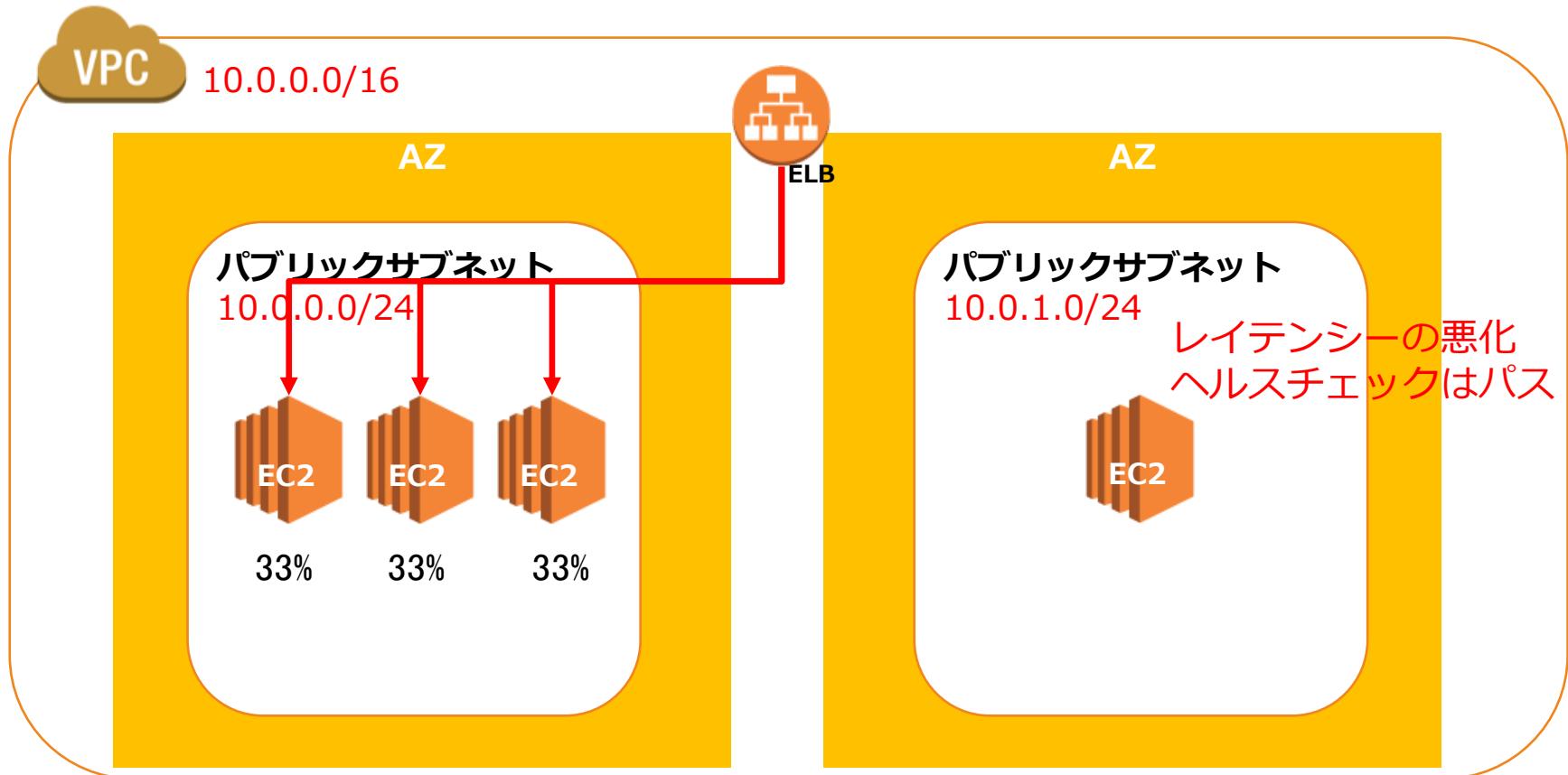
# ゾーンシフト機能 (ALB/NLB)

単一ゾーン障害が発生した際に、別のAZにトラフィックをシフトさせる機能。クロスゾーン負荷分散無効時の利用を推奨



# ゾーンシフト機能 (ALB/NLB)

単一ゾーン障害が発生した際に、別のAZにトラフィックをシフトさせる機能。クロスゾーン負荷分散無効時の利用を推奨



# [新Q] リスナールールの設定

ある会社はAmazon EC2インスタンスにホストされているウェブサイトを運用しています。このウェブサイトのEC2インスタンスはHTTPトラフィックとHTTPSトラフィックを別々に処理するALBのターゲットグループに構成されています。この企業は、すべてのリクエストをこのウェブサイトに転送して、リクエストがHTTPSを使用するようにしたいと考えています。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。

- 1) HTTPSトラフィックだけを許可するようにALBのネットワークACLルールを設定する。
- 2) HTTPをHTTPSに変換するリスナールールをALBで設定する。
- 3) HTTPトラフィックをHTTPSトラフィックにリダイレクトするリスナールールをALBで設定する。
- 4) ALBをNLBに置き換えて、HTTPトラフィックをHTTPSトラフィックにリダイレクトするリスナールールを設定する。



# リスナールールの設定

ELBはリスナー設定によって、通信するプロトコルタイプや通信ルールを設定することができる。

Add listener

▶ Details  
arn:aws:elasticloadbalancing:ap-northeast-1:860853660447:loadbalancer/app/realestate-ec2-elb/1b9213040f1f8c54

**Listener details**  
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Protocol	Port
HTTP ▾	: 81 1-65535

**Default actions** [Info](#)  
Specify the default actions for traffic on this listener. Default actions apply to traffic that does not meet the conditions of rules on your listener. Rules can be configured after the listener is created.

▼ 1. Redirect [Info](#)

[Remove](#)

Itemized URL	Full URL
--------------	----------

Protocol	Port
HTTPS ▾	: 443 1-65535 or to retain the original port enter #{port}



# [Q]暗号化通信

大手スーパー・マーケットチェーンはECアプリケーションを運用しています。冗長構成のために複数のEC2インスタンスに対してELBを利用したトラフィック制御とAuto Scalingを利用したスケーリングを設定しています。ELBを介した転送中のすべてのデータは暗号化する必要があります。

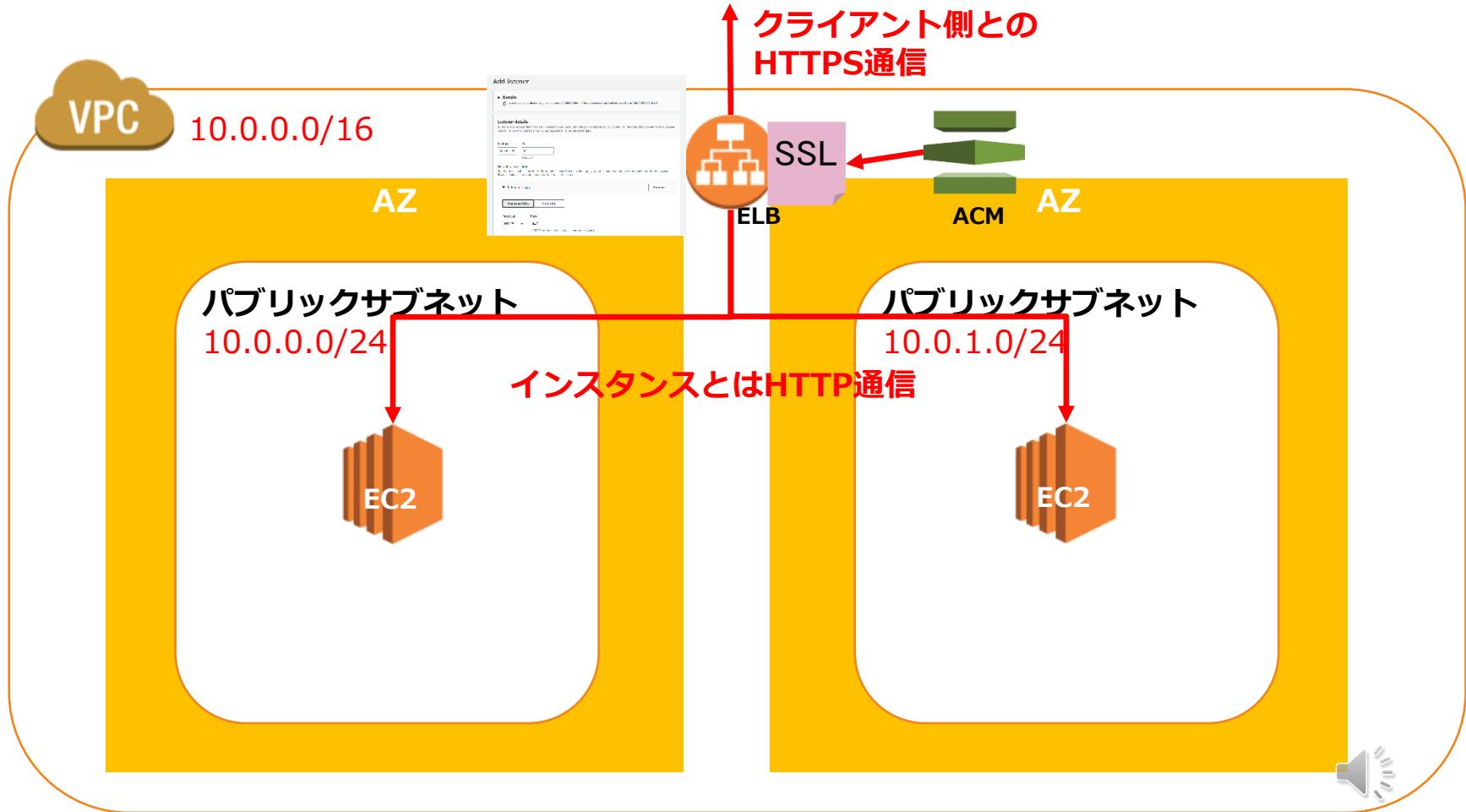
どのように暗号化要件を実現することができますか？（2つ選択してください）

- 1) NLBでTLSリスナーを構成してEC2インスタンスでSSLを終了する。
- 2) ALBでHTTPSリスナーを構成してALBにSSL証明書をインストールする。
- 3) NLBでHTTPSリスナーを構成してALBにSSL証明書をインストールする。
- 4) ALBでパススルーモードを使用して、EC2インスタンスでSSLを終了する。
- 5) ALBでTLSリスナーを構成してALBにSSL証明書をインストールする。



# 通信暗号化

ELBにACMが管理するSSL/TLS証明書を設定して、HTTPSリスナーを設定することで、クライアント型との通信の暗号化(HTTPS)する。



## [Q]ステイッキーセッション

大手スーパー・マーケット・チェーンはECアプリケーションを運用しています。冗長構成するために複数のEC2インスタンスに対してELBを利用したトラフィック制御とAuto Scalingを利用したスケーリングを設定しています。このシステムでは同じユーザーから断続的にシステム処理が発生することが多いため、同じユーザーには同じEC2インスタンスからのトラフィックを継続することが要件となっています。

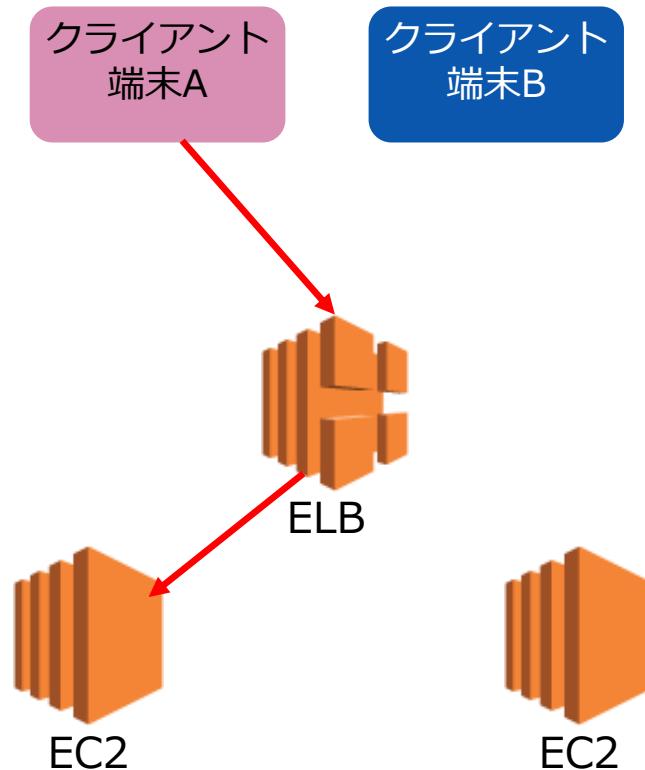
この要件を満たすためのELBの設定方法を選択してください。

- 1) 負荷分散機能を利用して、セッション中に、同じユーザから来たリクエストを全て、同じEC2インスタンスに送信する
- 2) Connection Drainingを利用して、セッション中に、同じユーザから来たリクエストを全て、同じEC2インスタンスに送信する
- 3) スティッキーセッションを利用して、セッション中に、同じユーザから来たリクエストを全て、同じEC2インスタンスに送信する
- 4) SSL Terminationを利用して、セッション中に、同じユーザから来たリクエストを全て、同じEC2インスタンスに送信する



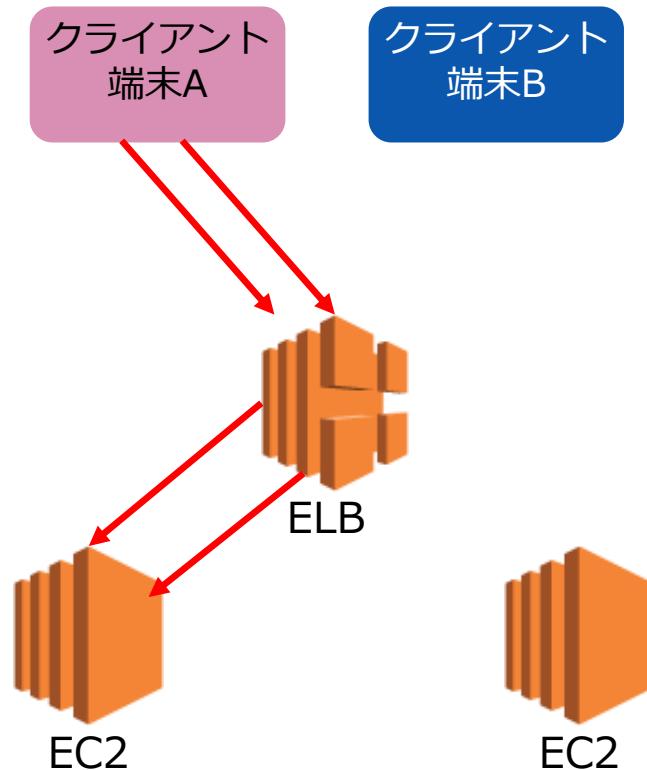
# ステイッキーセッション

セッションを継続して、同じクライアント端末からのリクエストを継続して同じEC2インスタンスに送信する



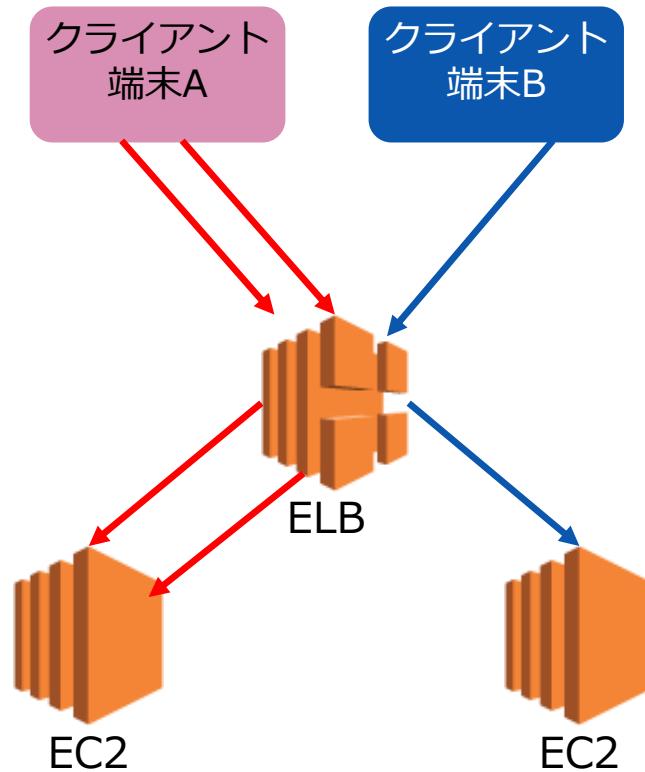
# ステイッキーセッション

セッションを継続して、同じクライアント端末からのリクエストを継続して同じEC2インスタンスに送信する



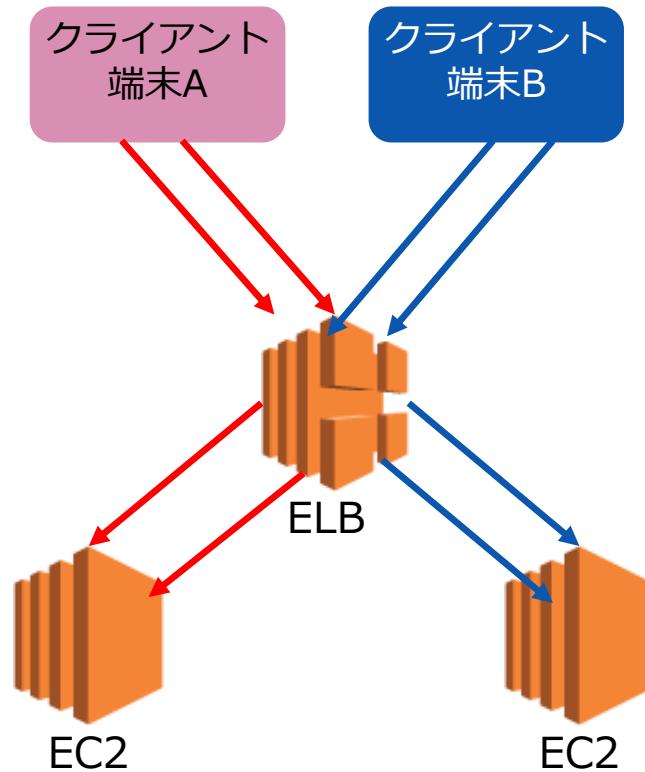
# ステイッキーセッション

セッションを継続して、同じクライアント端末からのリクエストを継続して同じEC2インスタンスに送信する



# ステイッキーセッション

セッションを継続して、同じクライアント端末からのリクエストを継続して同じEC2インスタンスに送信する



# [Q]Connection Draining

大手スーパー・マーケットチェーンはALBを設定したEC2インスタンスをマルチAZに構成したEC2アプリケーションを運用しています。開発チームは、インスタンスが異常になったときにELBからEC2インスタンスへの処理中のリクエストがドロップされるという問題を繰り返し発生しており、対応に追われています。

この問題に対処するために利用するべき機能はどれでしょうか？

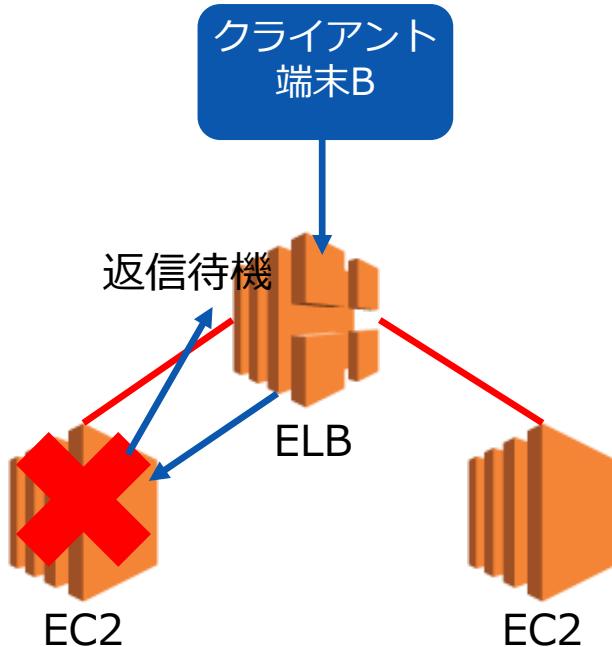
- 1) コネクションドレイニング
- 2) クロスゾーン負荷分散
- 3) スティックィセッション
- 4) ヘルスチェックの有効化



# Connection Draining

インスタンスが登録解除されるか異常が発生した場合に、そのバックエンドインスタンスへの指定した秒数の間は通信が切れずに、処理中のリクエストが終わるまで一定期間待ってくれる

障害時にリクエストがいきなり切れずに、一定期間返信を待ってくれる。



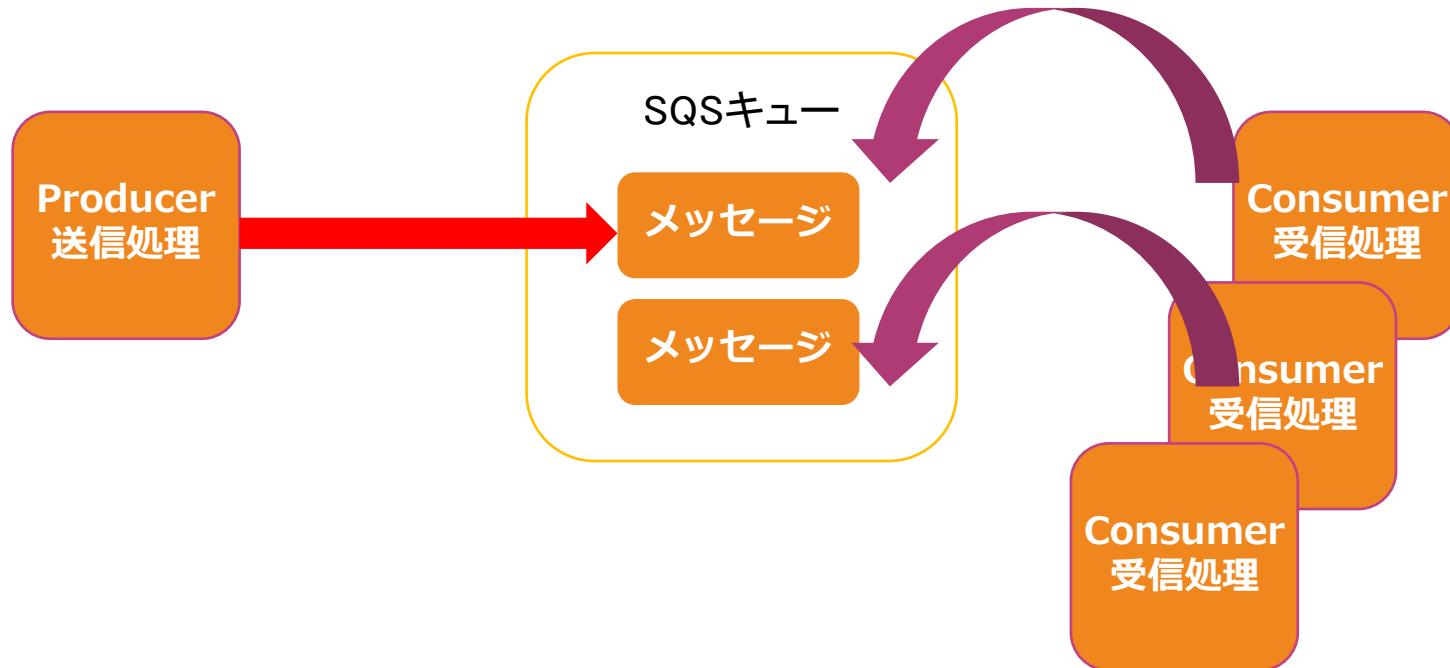
# セクションの内容

レクチャー	レクチャーで学ぶ内容
SQSの出題範囲	キューイングによるタスク管理を実施するSQSにおける出題問題を確認して、その範囲の知識を詳細に学習します。
CloudFrontの出題範囲	AWSのCDNサービスであるCloudFrontにおける出題問題を確認して、その範囲の知識を詳細に学習します。
DynamoDBの出題範囲	代表的なNoSQL型のデータベースであるDynamoDBにおける出題問題を確認して、その範囲の知識を詳細に学習します。
Lambdaの出題範囲	代表的なサーバレスコンピューティングであるLambdaにおける出題問題を確認して、その範囲の知識を詳細に学習します。
Route53の出題範囲	AWSにDNSサーバー機能を提供するRoute53における出題問題を確認して、その範囲の知識を詳細に学習します。

## SQSの出題範囲

# SQSとは何か？

タスクのトリガーとなるキューを複数管理することで、ワークロードの並列実行を実現するキューイングサービス



# SQSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

SQSの選択	✓ シナリオに基づいて、Amazon SNSやSESなどと比較して、SQSを選択する出題が問われる。
SQSの特徴	✓ SQSキューのポーリング処理などの特徴および制約に関する問題が出題される。 ✓ SQSの挙動や設定内容に関する問題が出題される。
SQSキュータイプ	✓ SQSで選択できる標準キューとFIFOキューの特徴とユースケースに関する質問が問われる。
SQSの識別子	✓ SQSの識別子として利用されるIDの特徴や使い方についての問題が出題される。
SQSの構成	✓ SQSをEC2インスタンスやECSなどと構成する際の基本的な構成方法が問われる。

# SQSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

SQSとAuto Scaling	✓ SQSをAuto Scalingと連動して利用する際のスケーリング設定などが問われる。
可視性タイムアウト	✓ 可視性タイムアウトの特徴とユースケースなどが問われる。
ポーリング方式	✓ ショートポーリングとロングポーリングの違い方とユースケースが問われる。
遅延キュー	✓ 遅延キューの特徴とユースケースなどが問われる。
優先度付キュー	✓ 優先度付キューの特徴とユースケースなどが問われる。

# SQSの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

メッセージタイマー	✓ メッセージタイマーの特徴とユースケースなどが問われる。
メッセージ重複排除ID	✓ メッセージ重複排除IDの特徴とユースケースなどが問われる。
デッドレターキュー	✓ デッドレターキューの特徴とユースケースなどが問われる。
SQSのバッチアクション	✓ SQSキューでまとめてメッセージを送付する際の設定方式が問われる。

# [Q]SQSの選択

あなたの会社はユーザーが投稿したビデオのアップロード・処理・公開用の動画管理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、ユーザーによってアップロードされたビデオを処理するために複数のEC2インスタンスを利用しています。ビデオを処理し公開するEC2ベースのワーカープロセスを有しており、Auto Scalingグループが設定されています。

ワーカープロセスの信頼性を高めるため利用すべきサービスを選択してください。

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

# SQSの選択

SQSはポーリング処理型のキューイングサービスで、タスクの並行実施などに利用される。

Amazon SNS	完全マネージド型 pub/sub メッセージングを実施するサービス。メール通知やプッシュ通知による連携処理に利用する。
Amazon SQS	完全マネージド型のキューイングサービス ポーリング処理によるタスクの並列実施に利用する。
Amazon SES	Eメール機能を可能にするサービス。アプリケーション上にEメール送受信機能を実装する際に利用する。安全、グローバル、大規模に E メールを送信が可能になる。
Amazon MQ	JMS、NMS、AMQP、STOMP、MQTT、WebSocket などの業界標準 API やメッセージング用プロトコルを使用するApache ActiveMQ 向けのマネージド型メッセージブローカーサービス
Amazon Kinesis Data Streams	5MBまでの容量のあるデータをストリーム処理することが可能で、シャード単位でデータを順番通りに送信することができる。



# [Q] SQSの特徴

あなたはソリューションアーキテクトとして、EC2インスタンスにホストされているEコマースサイトを構築しています。このサイトの注文はSQSキューからのメッセージによって処理サーバーが処理する構成となっています。SQSキューの可視性タイムアウトは30分に設定しています。注文が完了すると注文担当者にメッセージが通知される構成となっていますが、注文に対してメッセージ通知のいくつかが配信されないトラブルが発生しています。

この問題の最も可能性が高い原因はどれでしょうか？

- 1) 注文を処理するサーバーがメッセージを処理後に、SQSキュー内のメッセージを削除していない。
- 2) 標準キューを利用しているためメッセージに重複が発生している。
- 3) キューはショートポーリングに設定されているため、空のメッセージ取得が増加している。
- 4) いくつかの注文メッセージがデットレターキューへと移行している。

# SQSの特徴

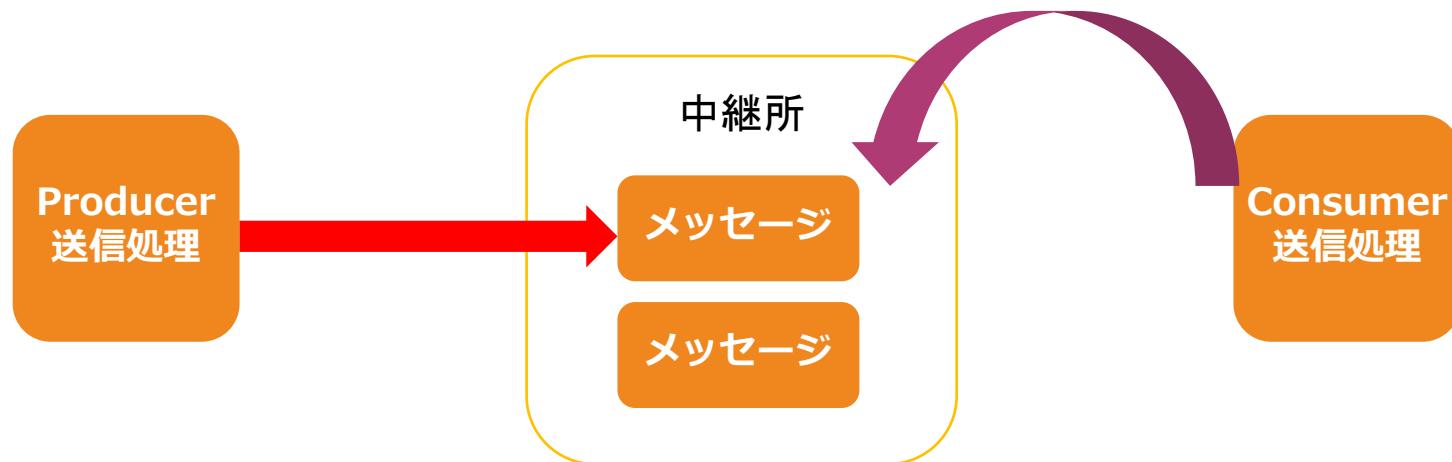
AWSの様々なサービスと連携して通知可能で、疎結合アーキテクチャに実現する。

## 【基本的な機能】

- 単一発行メッセージをキューとして利用
- ポーリング処理型のキューイングサービス
- 標準キューはメッセージ通信順番は保証されないが、FIFO  
キューは順番を保証する。
- 優先キューは他のキューよりも優先的に処理させることが可能
- メッセージ保持期間の間はメッセージを保持するが、超過するとメッセージを削除する。
- 発行したメッセージは取り消し不可
- 配信ポリシーによるキューの再試行を実施する

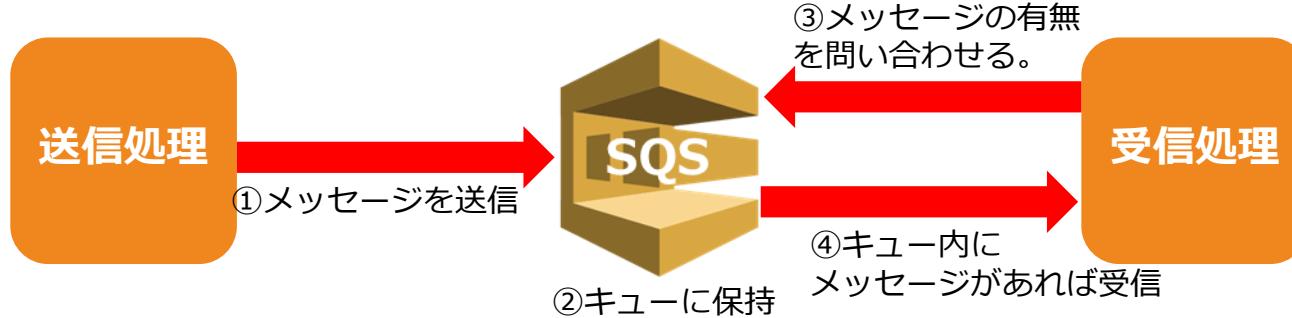
# SQSの特徴

キューとはProducerが送信したメッセージをキュー内に蓄積して、Consumerがプルすることで処理が始まるメッセージ方式



# SQSの特徴

SQSはキューを発行・蓄積して、ポーリング処理を管理する。



# SQSキューの特徴

無制限にメッセージを利用可能だが、メッセージ保持期間をうまく設定することが必要

## メッセージの制約

メッセージ数は無制限に利用可能

メッセージサイズは最大256KB

ただし、拡張クライアントライブラリーを利用すると2GBまでのメッセージのやり取りが可能となる。

## メッセージの保持期間

SQSのキューメッセージは保持期間の間は保存される。

デフォルト4日間（最小60秒～最大14日で設定可能）

APPLICATION上でメッセージを削除する処理を実施しないと、期間を超過するまでキューが滞留してしまう。

## [Q] SQSのキュータイプ

グローバルコンサルティングファームではコンサルティングの知見をグローバルで共有するための情報共有システムをAWS上に構築しています。このシステムはストレージレイヤーにS3を利用していますが、S3にデータがアップロードされるたびに、イベントをトリガーによってメッセージを配信する処理を追加します。その際は、メッセージの順序が正しく送信される必要があります。

要件を満たすための最適なソリューションはどれでしょうか？

- 1) S3イベント通知にAmazon SQSの標準キューを設定して、データアップロードをトリガーにして、メッセージを通知する。
- 2) S3イベント通知にAmazon SQSのFIFOキューを設定して、データアップロードをトリガーにして、メッセージを通知する。
- 3) S3イベント通知にAmazon SNSの標準トピックを設定して、データアップロードをトリガーにして、メッセージを通知する。
- 4) S3イベント通知にAmazon SNSのFIFOトピックを設定して、データアップロードをトリガーにして、メッセージを通知する。

# SQSのキュータイプ

SQSでは標準キューとFIFOキューのどちらかを選択して、SQSを初期設定することになる。

## 標準キュー

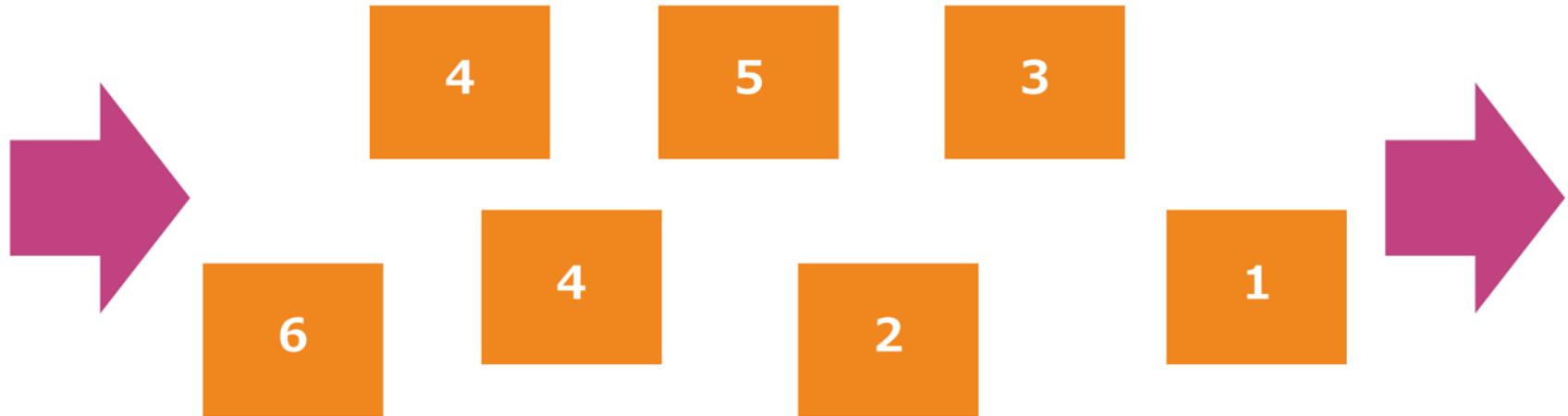
- ✓ メッセージの1つ以上のコピーが順序どおりに配信できないことがある。
- ✓ メッセージが少なくとも1回配信される方式であり、キューには重複が発生する可能性がある。
- ✓ 1秒あたりのトランザクション数はほぼ無制限
- ✓ 標準キューは、アプリケーションが1回以上に順序が正確ではなくても配信されれば良いケースで利用する。

## FIFOキュー

- ✓ 先入れ先出し方式 (FIFO) により配信順番を守る。
- ✓ メッセージが1回だけ配信され、コンシューマがプロセスを処理して削除するまで使用可能なキューの状態を保つため、キューに重複がない。
- ✓ 1秒あたり300トランザクションに制限
- ✓ FIFO キューは、操作やイベントの順序が重要である場合や、重複を許容できないユースケースに利用する。

# 標準キュー

標準キューは「順番通りの処理」と「1回だけのメッセージング」を“なるべく”実施するキュー方式



# FIFOキュー

その名の通り、最初に入ったキューを最初に処理する順番を守るキュー方式



# [Q] SQSの識別子

あなたはソリューションアーキテクトとして、IoTデバイスからのストリーミングデータを分析するワークフローを構築しています。このストリーミング処理では、データは1分ごとにAWSに送信されます。各IoTデバイスのデータは順番に個別に処理することが求められています。また、IoTデバイスは同じ場所に設置された2～5個のグループ単位になっており、グループでまとめてデータを解析することも必要です。

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) SQSのFIFOキューを使用し、IoTデータのデバイスIDの値を表すグループID属性を付与してメッセージを送信する。
- 2) SQSの標準キューを使用し、IoTデータのデバイスIDの値を表すグループID属性を付与してメッセージを送信する。
- 3) Kinesis Data Streamsを使用し、IoTデータのデバイスIDの値を表すグループID属性を付与してメッセージを送信する。
- 4) Kinesis Data Streamsを使用し、IoTデータのデバイスIDの値を表すグループID属性を付与してシャードごとに分離してデータを処理する。

# SQSの識別子

SQSではキューを利用する際に様々な機能を利用することが可能。ユースケースに応じて使い分ける必要がある。

## キューURL

- キューに割り当てられるURL

## メッセージID

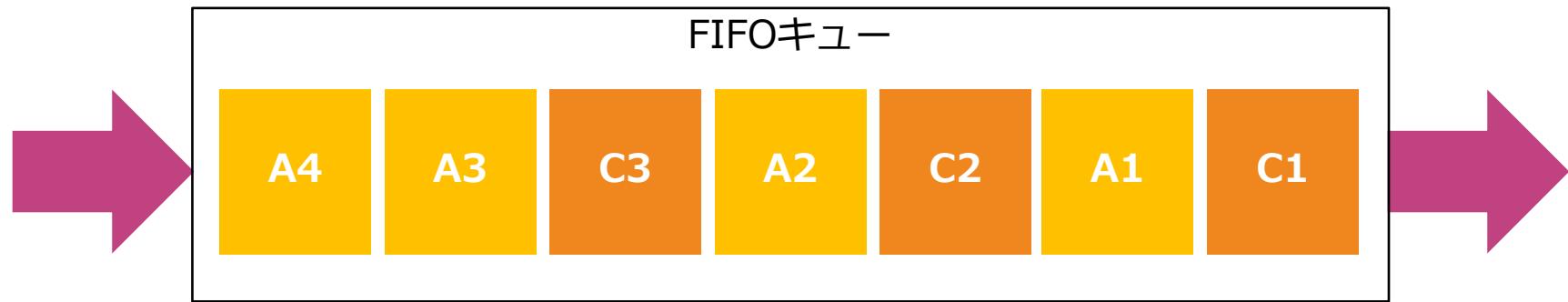
- メッセージに対して割り当てられたID

## メッセージグループID

- メッセージグループ ID は 特定のメッセージグループに属するメッセージを指定するタグ
- 同じメッセージグループに属するメッセージは、メッセージグループに相対的な厳密な順序で 1 つずつ処理される、
- 単一の FIFO キュー内で複数の順序付きメッセージグループをインターリーブするには、メッセージグループ ID 値を使用する。

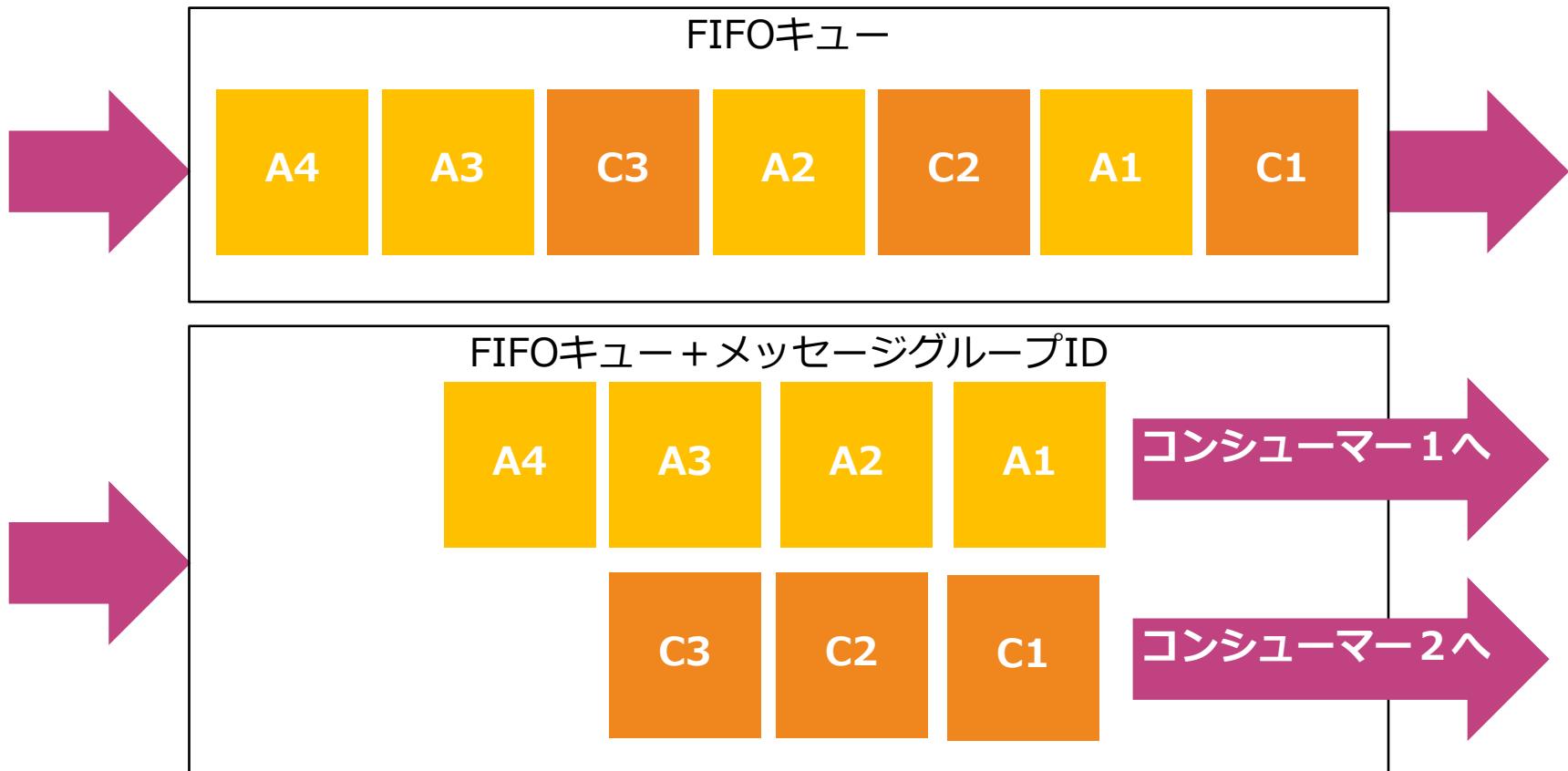
# メッセージグループIDによるグループ化

メッセージIDを利用して同じグループのメッセージはまとめて送信されるようにFIFOキューを調整する。



# メッセージグループIDによるグループ化

メッセージIDを利用して同じグループのメッセージはまとめて送信されるようにFIFOキューを調整する。



# [Q] SQSの構成

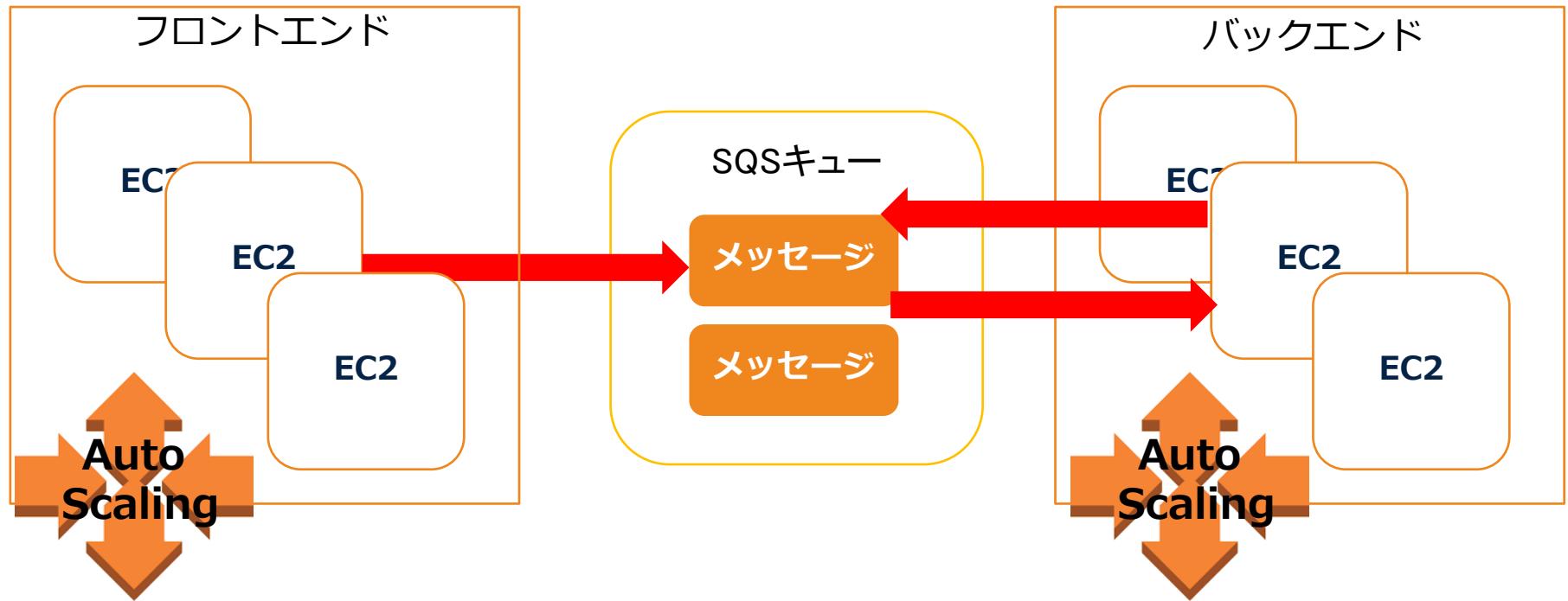
マーケティング会社はAmazon ECSを利用してデータ分析アプリケーションを構築しています。このアプリケーションは複数のAmazon ECSタスクで実行されます。フロントエンドアプリケーションがデータの前処理を実施し、そのデータをバックエンドのECSタスクに渡してデータ解析を実行します。これらの分析処理が並列で実行されることで高パフォーマンスを達成しつつ、障害が他のコンポーネントに影響を与えないように、相互依存性を減らす必要があります。

この要件を満たすことができるコスト最適なAWSアーキテクチャ構成はどれでしょうか？

- 1) Amazon SQSキューを作成し、キューにメッセージを追加するようにフロントエンドを設定し、メッセージについてキューをポーリングするようにバックエンドを設定する。
- 2) Amazon SQSキューを作成し、キューにメッセージを追加するようにバックエンドを設定し、メッセージについてキューをポーリングするようにフロントエンドを設定する。
- 3) Amazon SNSを作成し、キューにメッセージを追加するようにフロントエンドを設定し、メッセージについてキューをポーリングするようにバックエンドを設定する。
- 4) Amazon SNSを作成し、キューにメッセージを追加するようにバックエンドを設定し、メッセージについてキューをポーリングするようにフロントエンドを設定する。

# SQSの基本構成

フロントエンドサーバーからキューがトリガーされて、バックエンドの処理サーバーが並列処理するのがSQSの基本構成



# [Q] SQSとAuto Scaling

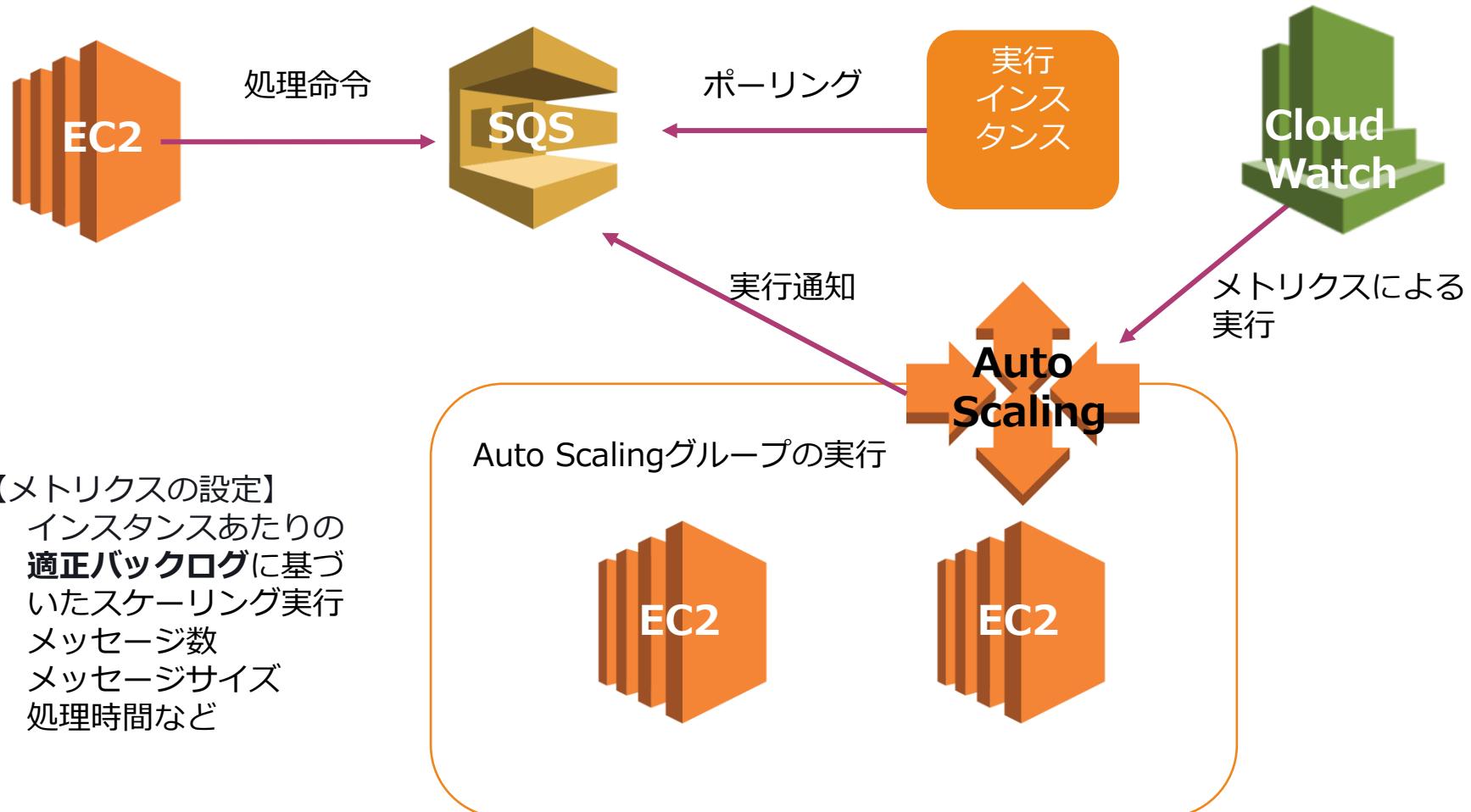
B社ではAWS上に動画処理を実行するワークフローを構築しました。このシステムはデータ処理を並列処理するためのキューを利用した分散構成が必要となります。このジョブは不定期に実行され、処理変更も多いため実行期間が不明確です。また負荷の増減も多いようです。この動画処理システムは中長期稼働させる予定であり、1つ1つの編集処理は1分から30分ほどで完了します。

この要件を満たすことができる最もコスト最適なAWSアーキテクチャ構成の組合せはどれでしょうか？（2つ選択してください。）

- 1) 動画処理サーバーにリザーブドインスタンスを利用して、SQSによる並列処理を設定する。
- 2) 動画処理サーバーにスポットインスタンスを利用して、SQSによる並列処理を設定する。
- 3) 動画処理サーバーにスポットインスタンスを利用して、Lambdaによる並列処理を設定する。
- 4) Auto Scalingにスポットインスタンスを利用したスケーリングを構成して、SQSの適正バックログをしきい値に設定してスケーリングを実行する。
- 5) Auto Scalingにスポットインスタンスを利用したスケーリングを構成して、SQSのメッセージ数をしきい値に設定してスケーリングを実行する。

# SQSとAuto Scaling

SQSとAuto Scalingを構成する際は、CloudWatchメトリクスに基づいてキューの処理量に応じたスケーリングを設定する。



# [新Q]可視性タイムアウト

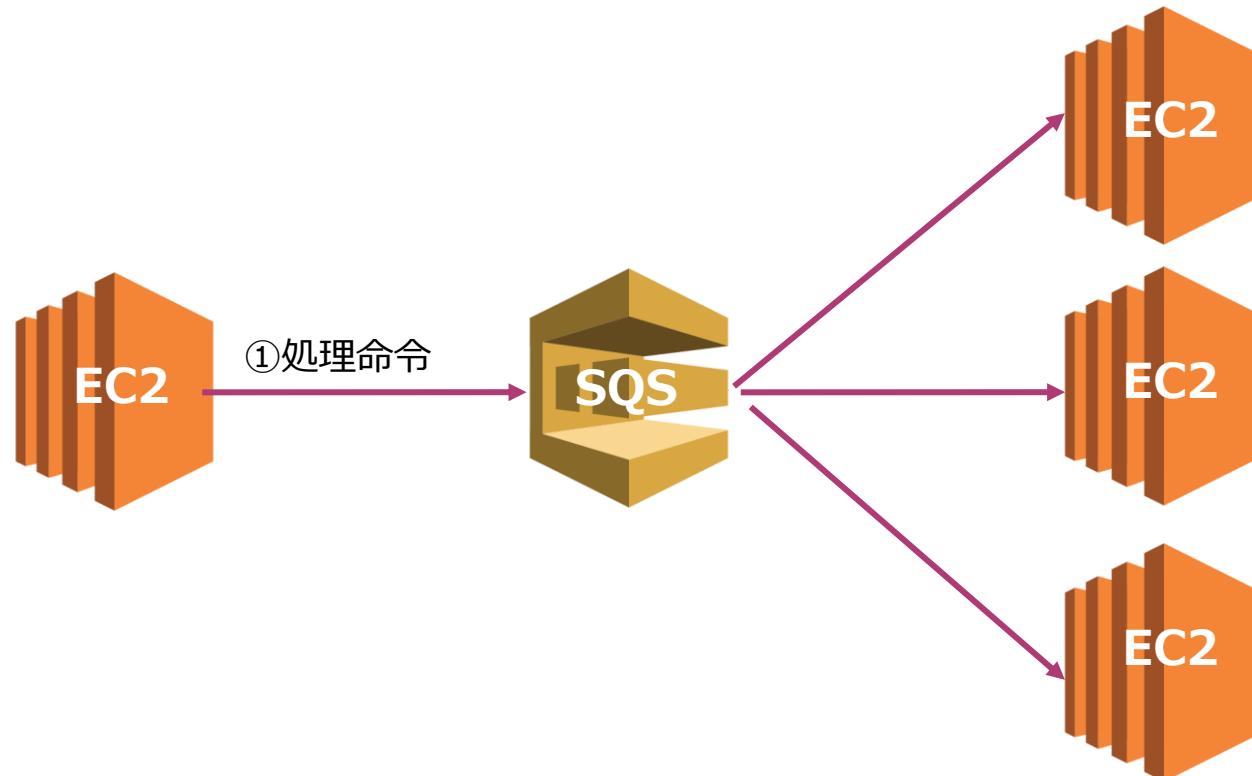
ある企業は、AWS上でウェブアプリケーションを構築しています。このアプリケーションは複数のEC2インスタンス上にホストされています。このアプリケーションは、Amazon SQSキュー内からメッセージを取得して、EC2インスタンスがメッセージを処理して、Amazon RDSデータベースに処理結果を書き込みます。処理が完了するとキューからメッセージを削除します。Amazon SQSキュー内のメッセージには重複はありませんが、RDS内の保存データには重複レコードが稀に見つかります。

重複メッセージが発生しないように、ソリューションアーキテクトは何を実施するべきでしょうか。

- 1) ChangeMessageVisibility APIを使用して、適切な可視性タイムアウト値を設定する。
- 2) AddPermission APIを使用して、適切な権限を付与する。
- 3) CreateQueue APIを使用して、新しいキューを作成する。
- 4) ReceiveMessage APIを使用して、適切な待機時間を設定する。

# 可視性タイムアウト

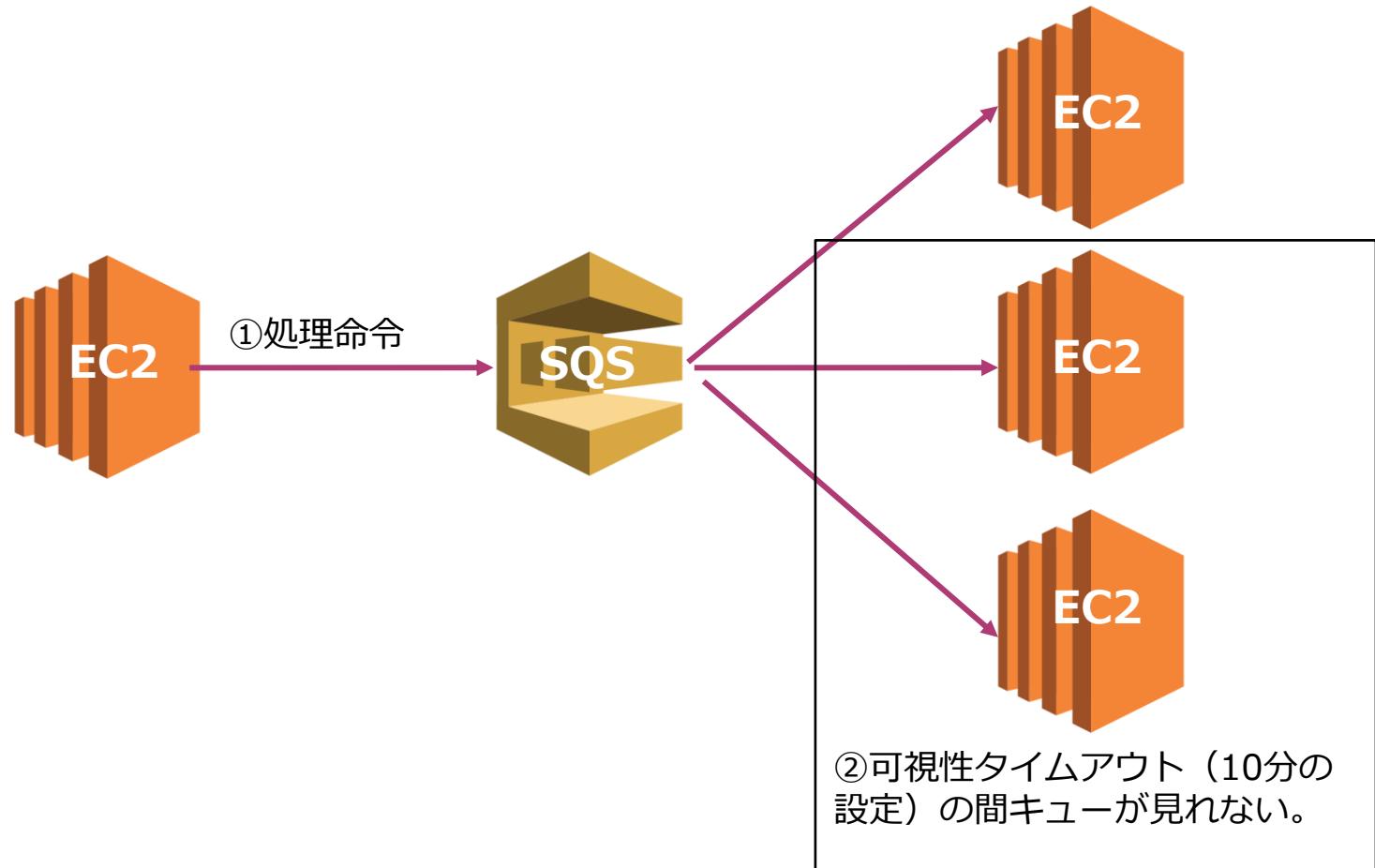
可視性タイムアウトは処理担当のインスタンス以外からは一定時間（30秒～12時間）キューが見えなくなる機能



メッセージが受信された直後は、メッセージはキューに残ったままとなる。他のコンシューマーが同じメッセージを再処理しないように、Amazon SQSは可視性タイムアウトを設定することで、重複処理を防ぐことができる。

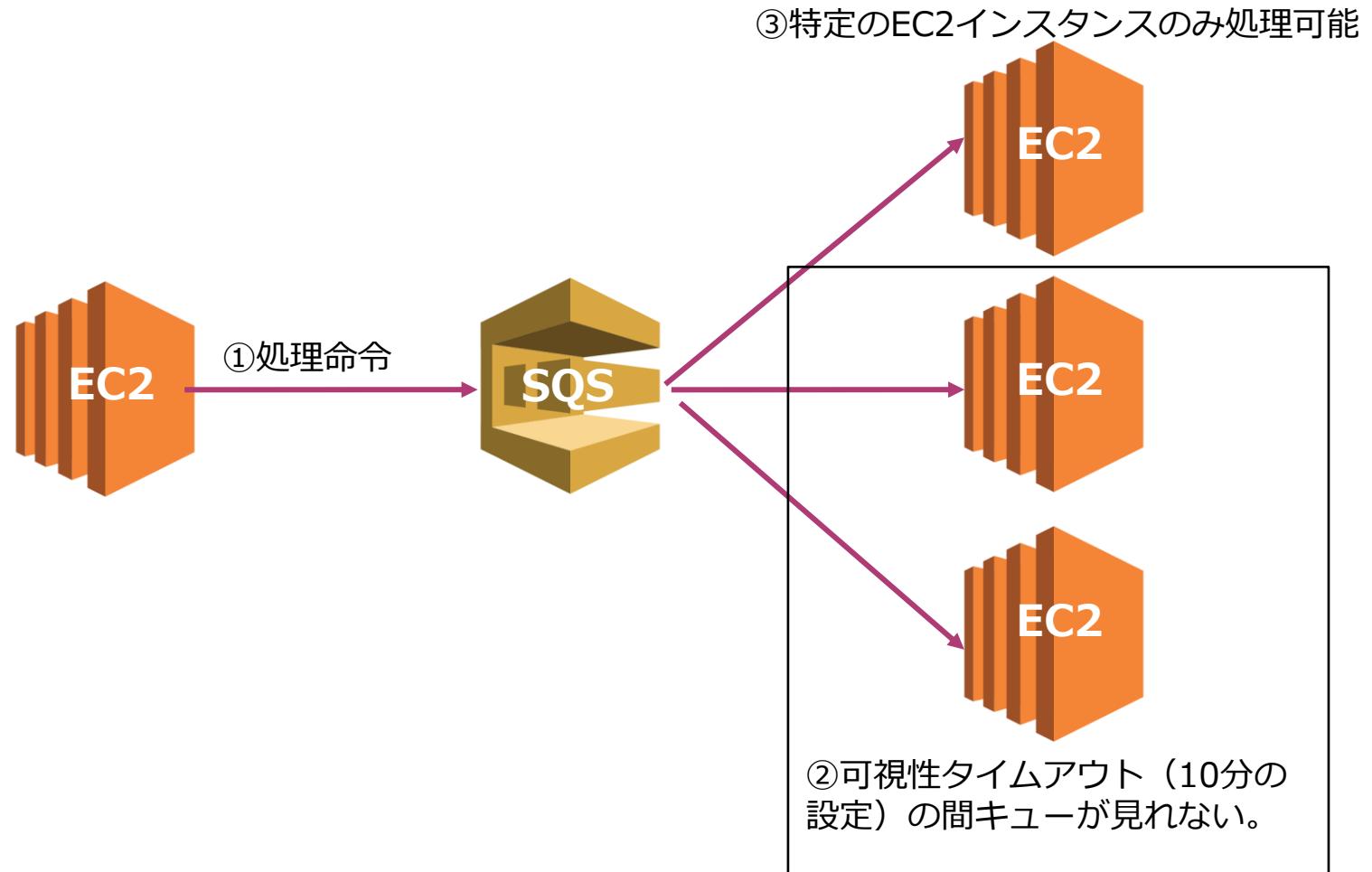
# 可視性タイムアウト

可視性タイムアウトは処理担当のインスタンス以外からは一定時間（30秒～12時間）キューが見えなくなる機能



# 可視性タイムアウト

可視性タイムアウトは処理担当のインスタンス以外からは一定時間（30秒～12時間）キューが見えなくなる機能



# [Q]ポーリングの方式

B社はAWS上に動画処理を実行するワークフローを構築しました。このアプリケーションでは、EC2インスタンスによる動画編集処理をAmazon SQSキューを使用して、並行処理する構成としています。開発チームは、動画編集中に動画リストが更新されると動画リストが未処理となり、メッセージ処理が失敗する事象を発見しました。

メッセージ処理が失敗するケースで利用するべきAmazon SQSの機能はどれでしょうか？

- 1) 遅延キューを使用して、メッセージ処理の失敗を処理する。
- 2) ショートポーリングを使用して、メッセージ処理の失敗を処理する。
- 3) ロングポーリングを使用して、メッセージ処理の失敗を処理する。
- 4) デッドレターキューを使用して、メッセージ処理の失敗を処理する。

# ポーリングの方式

ポーリング処理の方式でショートポーリングとロングポーリングの2通りがある。

## ロングポーリング

問い合わせの結果が空であった場合に、指定したメッセージ受信待機時間はSQSは待機してから応答を返す。メッセージ受信待機時間は0秒から20秒で設定  
空のレスポンス数を削減することができる。

## ショートポーリング

キューが空の場合にすぐに空のメッセージが返される

# [Q]遅延キュー

あなたはソリューションアーキテクトとして、マイクロサービスを利用したアプリケーションを構築しています。マイクロサービス間のコンポーネントを分離するためにSQSキューを使用しています。各コンポーネントはSQSメッセージを処理するためには一定の時間を必要とするため、新しいメッセージのキューへの配信を10秒間停止してから、処理を開始することがキュー設定として必要不可欠です。

この要件を満たすことができるキューの設定方法を選択してください。

- 1) 遅延キューを使用して、キューへの新しいメッセージの配信を10秒間延期する。
- 2) ショートポーリングを使用して、キューへの新しいメッセージの配信を10秒間延期する。
- 3) メッセージタイマーを使用して、キューへの新しいメッセージの配信を10秒間延期する。
- 4) 可視性タイムアウトを使用して、キューへの新しいメッセージの配信を10秒間延期する。

# [Q]メッセージタイマー

あなたはソリューションアーキテクトとして、マイクロサービスを利用したアプリケーションを構築しています。マイクロサービス間のコンポーネントを分離するためにSQSキューを使用しています。現在、あなたは特定のメッセージのキューへの配信を10秒延期し、他のすべてのメッセージはすぐにキューに配信する設定を実施しているところです。

この要件を満たすことができるキューの設定方法を選択してください。

- 1) 遅延キューを使用して、キューへの特定のメッセージの配信を10秒間延期する。
- 2) ショートポーリングを使用して、キューへの特定のメッセージの配信を10秒間延期する。
- 3) メッセージタイマーを使用して、キューへの特定のメッセージの配信を10秒間延期する。
- 4) 可視性タイムアウトを使用して、キューへの特定のメッセージの配信を10秒間延期する。

# [Q]優先度付キュー

B社ではAWS上に動画編集アプリケーションを構築しました。この動画処理アプリケーションはEC2インスタンスから送信されたAmazon SQSキューからのメッセージにより動画編集を実行して、処理後の動画をS3に保存します。ユーザーは無料ユーザーと有料ユーザーとに分かれます。有料ユーザーから提出されたファイルは優先的に処理される必要があります。

このような要件を満たすことができる実装方法を選択してください。

- 1) SQSを利用して、有料ユーザーには優先的に処理するメッセージを設定し、無料ユーザーにはデフォルトメッセージを利用する。
- 2) Lambdaファンクションを利用して有料ユーザーのメッセージ処理を優先的に処理するポーリング処理を設定し、無料ユーザーにはデフォルトメッセージを利用する。
- 3) SNSを利用して有料ユーザーのメッセージ処理を優先的に処理するポーリング処理を設定し、無料ユーザーにはデフォルトメッセージを利用する。
- 4) Amazon MQを利用して、有料ユーザーには優先的に処理するメッセージを設定し、無料ユーザーにはデフォルトメッセージを利用する。

# キューの詳細設定

SQSではキューを利用する際に様々な機能を利用することが可能。ユースケースに応じて使い分ける必要がある。

## 遅延キュー

キューへの新しいメッセージの配信を数秒間遅延させることができる機能（0秒から15分で設定）

可視性タイムアウトとの違いは、キューが発行された直後から見えなくなるということ。またキュー全体に効果がある。

## 優先度付きキュー

キューの処理順序に優先度をつけることができる。

これにより、優先対応があるタスクを最初に処理するようにワークフローを設定できる。

## デッドレターキュー

このキューは、正常に処理（消費）できないメッセージを別のキューへと移動させる。

処理不能なキューが蓄積されるのを防ぎつつ、処理できなかつた理由を後で解析できる。

# キューの詳細設定

SQSではキューを利用する際に様々な機能を利用することが可能。ユースケースに応じて使い分ける必要がある。

## メッセージ重複排除ID

- 送信されたメッセージの重複排除に使用するトークン
- 同一の重複排除IDが設定されたメッセージをキューへ送っても5分間の間は受け付けられないように設定できる。
- 個別のメッセージグループではなくキュー全体に適用される。
- FIFOのみで利用する

## 暗号化

- AWS Key Management Service (AWS KMS)を使用して、送信データを暗号化する。

## メッセージタイマー

- メッセージタイマーはメッセージが発出された瞬間から、キューに追加されたメッセージが表示されないようにする機能。45秒のタイマーでメッセージを送信すると、キューの最初の45秒間は表示されない。
- 個々のメッセージではなくキュー全体に対して遅延の秒数を設定するには、遅延キューを使用する。
- 個々のメッセージのメッセージタイマー設定はキュー全体よりも優先される。

## [Q] SQSのバッチアクション

あなたの会社はAWS上でワークフローを実行する業務システムを構築しています。あなたのソリューションアーキテクトとして、SQSを利用したキューイングによる高可用で高性能なフローを実装しています。SQSを介して処理される1秒あたり約1000メッセージのピークレートが期待されており、メッセージが順番に処理されることが重要です。

このSQSの実装の際に利用するべき機能はどれでしょうか？

- 1) 操作ごとに4メッセージのバッチモードでFIFOキューを使用する。
- 2) 操作ごとに2メッセージのバッチモードでFIFOキューを使用する。
- 3) 操作ごとに4メッセージのバッチモードで標準キューを使用する。
- 4) 操作ごとに2メッセージのバッチモードで標準キューを使用する。

# SQSのバッチアクション

バッチアクションは1回のアクションで複数のメッセージを操作するなどのバッチ処理が設定可能

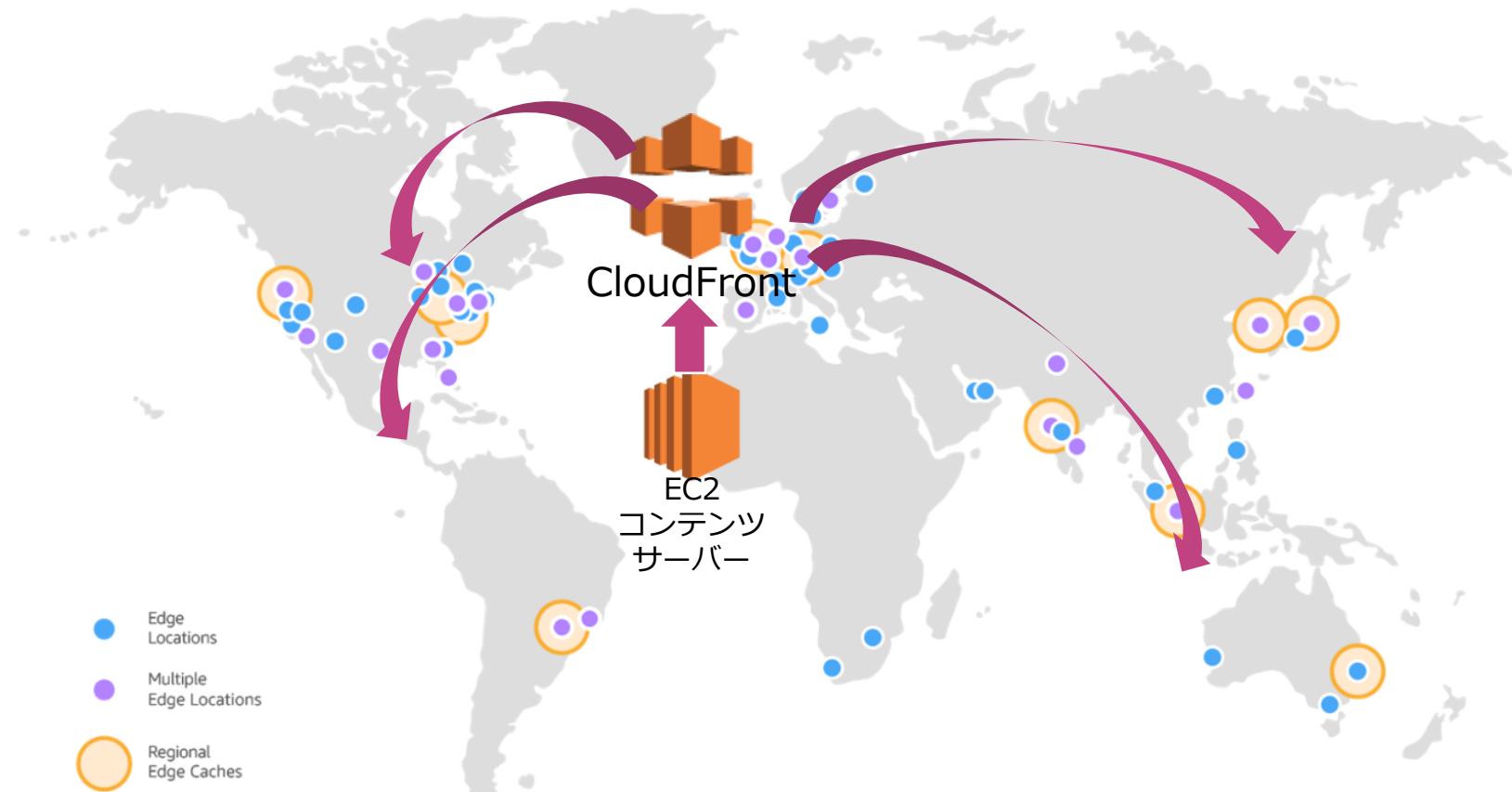
Amazon SQS バッチアクションをサポートする AWS SDK を使用して、バッチ機能を活用できる。

- SendMessageBatch
- DeleteMessageBatch
- ChangeMessageVisibilityBatch

## CloudFrontの出題範囲

# CloudFrontとは何か？

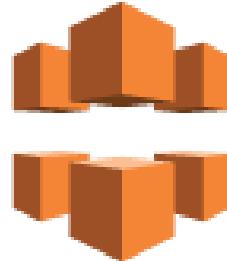
コンテンツ配信をグローバルロケーションを使って効率的に実施するサービス



参照 : <https://aws.amazon.com/jp/cloudfront/features/?nc=sn&loc=2>

# CloudFrontとは何か？

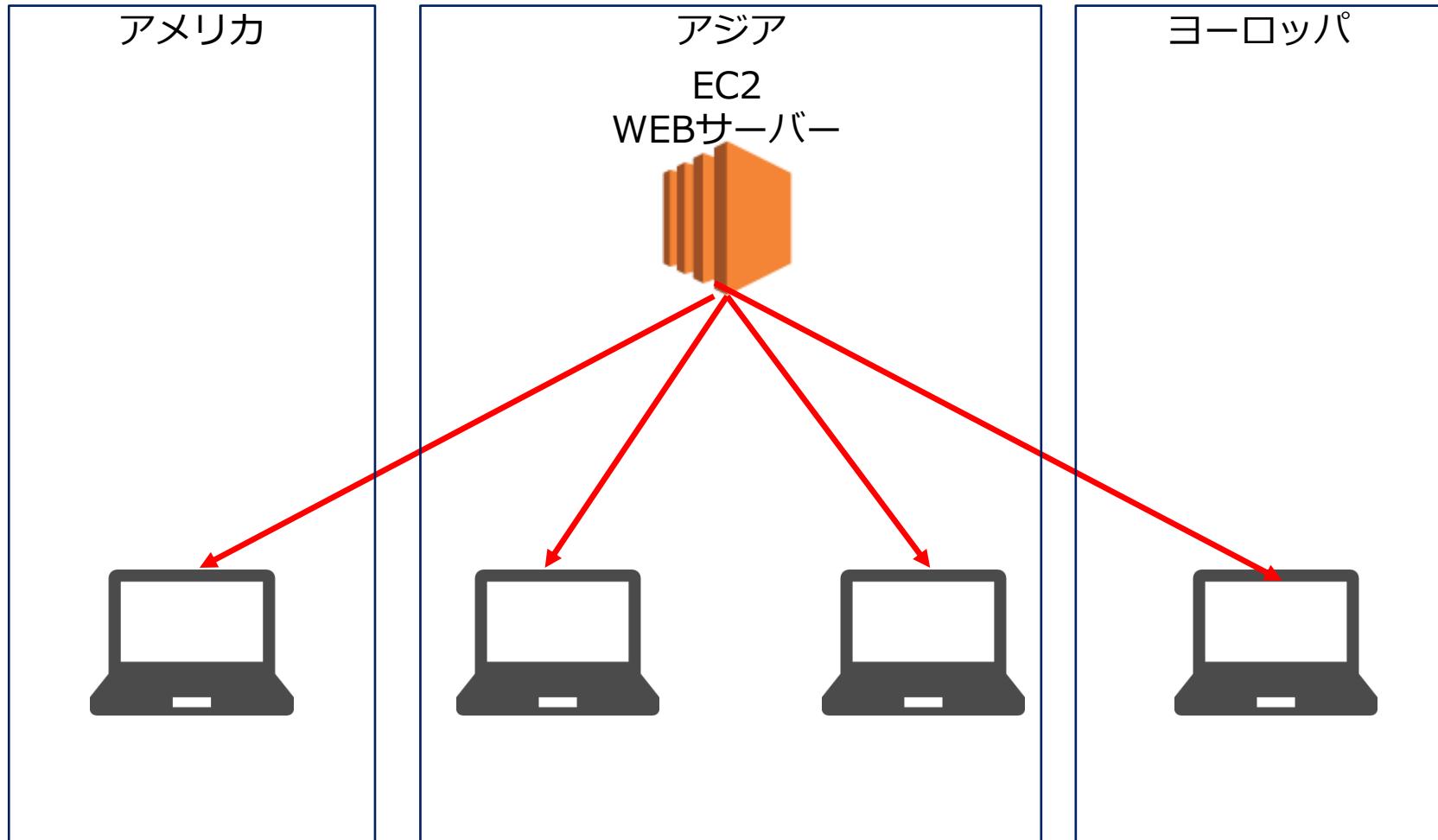
CloudFrontはAWSが提供するCDN（Content Delivery Network）サービス



CloudFront

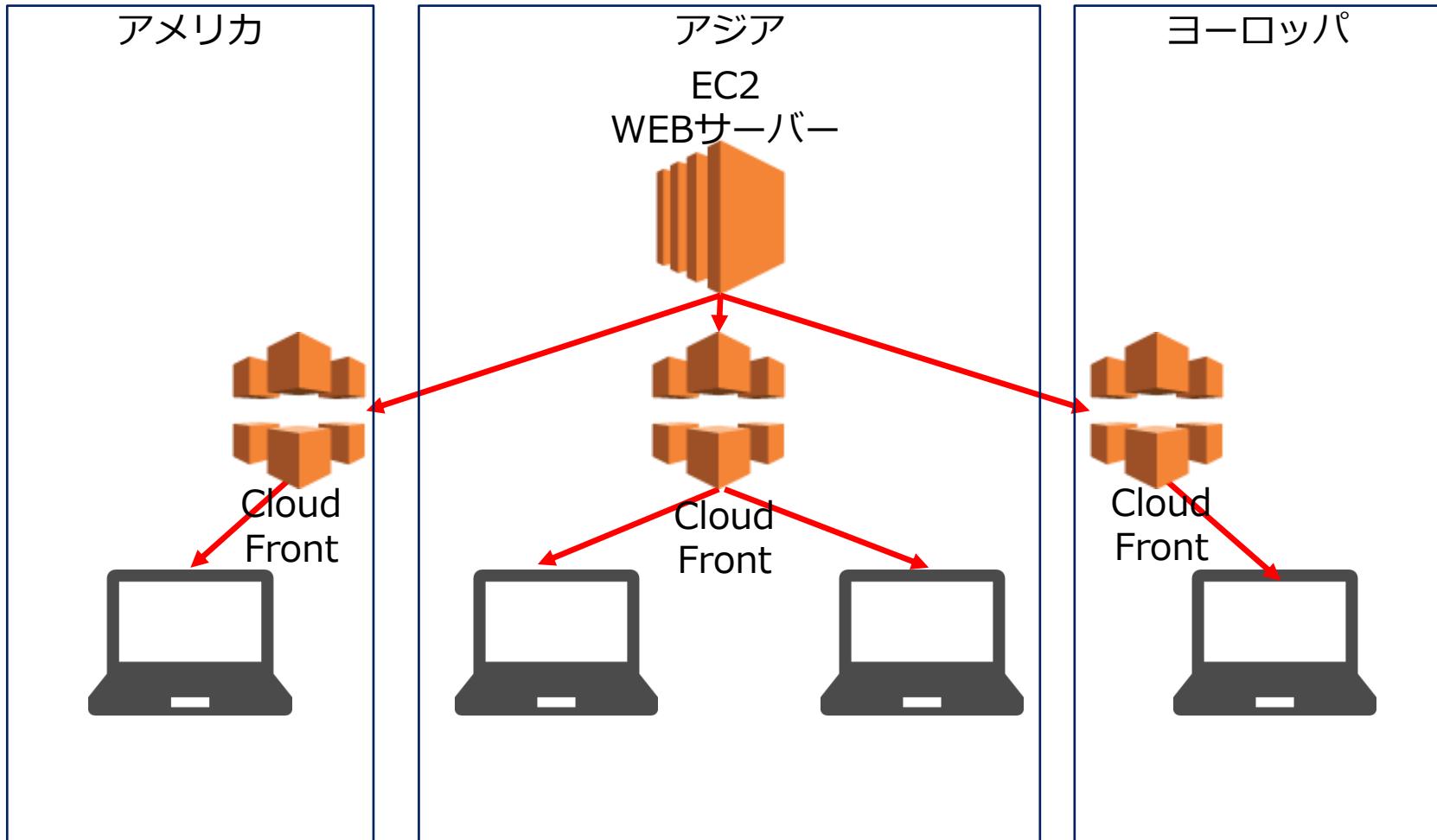
# CloudFrontとは何か？

CDNはWEBコンテンツ配信処理を高速化するためのサービス



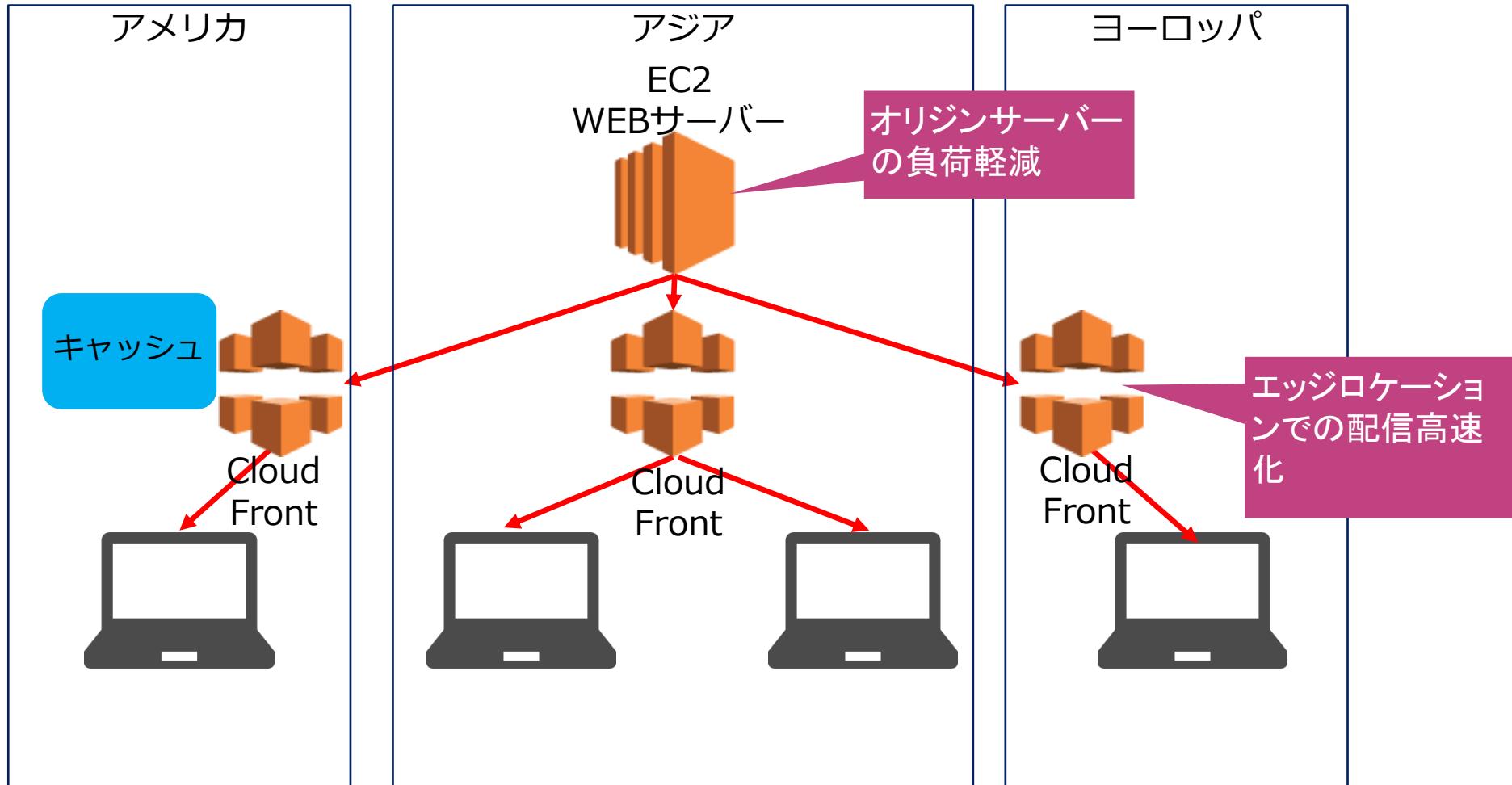
# CloudFrontとは何か？

CDNはWEBコンテンツ配信処理を高速化するためのサービス



# CloudFrontとは何か？

CDNはWEBコンテンツ配信処理を高速化するためのサービス



# CloudFrontの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

CloudFrontのS3構成	✓ コンテンツ配信を高パフォーマンス化するなどのシナリオに基づいてCloudFrontを利用した構成が問われる。
CloudFrontのカスタムオリジン構成	✓ EC2やELBなどをカスタムオリジンとした場合のCloudFrontの構成が問われる。
オリジンの冗長化	✓ オリジンサーバーの冗長化が必要とされるシナリオに基づいて、CloudFrontの冗長化構成が問われる。
エッジロケーション	✓ CloudFrontが配信の際に利用するエッジロケーションの利用方法が問われる。
リージョナルエッジキャッシュ	✓ CloudFrontが配信の際に利用するリージョナルエッジキャッシュの利用方法が問われる。

# CloudFrontの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

CloudFrontの挙動	✓ CloudFrontが最初にキャッシュを取得する挙動や、キャッシュデータが存在しない場合のCloudFrontの挙動が問われる。
キャッシュの保持期間の設定	✓ CloudFrontの配信設定時におけるキャッシュの保持期間の設定方法が問われる。 ✓ Cache-Control ヘッダーを利用した設定方法が問われる。
キャッシュの活用	✓ キャッシュを活用した配信処理設定や、細かい制御をする方法や効果が問われる。
CloudFrontの利用料	✓ CloudFrontにおいてコストが発生する要因が問われる。
Gzip圧縮機能	✓ Gzip圧縮機能の活用方法が問われる。

# CloudFrontの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

オリジンへのアクセス制御	✓ CloudFrontを迂回してオリジンにアクセスすることを制限する設定方法が問われる。
キャッシュのアクセス制御	✓ ユーザーがキャッシュデータにアクセスする際の利用制限をする方法が問われる。
CloudFront地域制限	✓ 特定の国や地域からCloudFront配信へのアクセスを制限する設定が問われる。
暗号化	✓ CloudFrontにおける通信の暗号化の設定方法が問われる。 ✓ CloudFrontにおけるフィールドレベル暗号化の用途が問われる。
ログ取得	✓ CloudFrontにおけるログ取得方法とその使い方が問われる。

# CloudFrontの特徴

世界中にあるエッジロケーションを活用してグローバルに効率的かつ高速にコンテンツ配信することができるCDNサービス

- 400以上のエッジロケーションにより、グローバルな分散配信を実施
- ユーザーに近いロケーションからキャッシュを使って配信することで高いパフォーマンスを達成
- AWS WAF/AWS Shieldによるエッジ側でのセキュリティ強化
- AWS Certificate Managerによる暗号化の実施
- S3バケットをオリジンに設定可能。また、EC2などをカスタムオリジンとして設定可能
- オリジンに対してHeader/Cookie/Query Stringsによるフォワード指定で、動的なページ配信が可能
- エッジロケーションでLambda関数を利用したコード処理も可能



# [Q]CloudFrontのS3構成

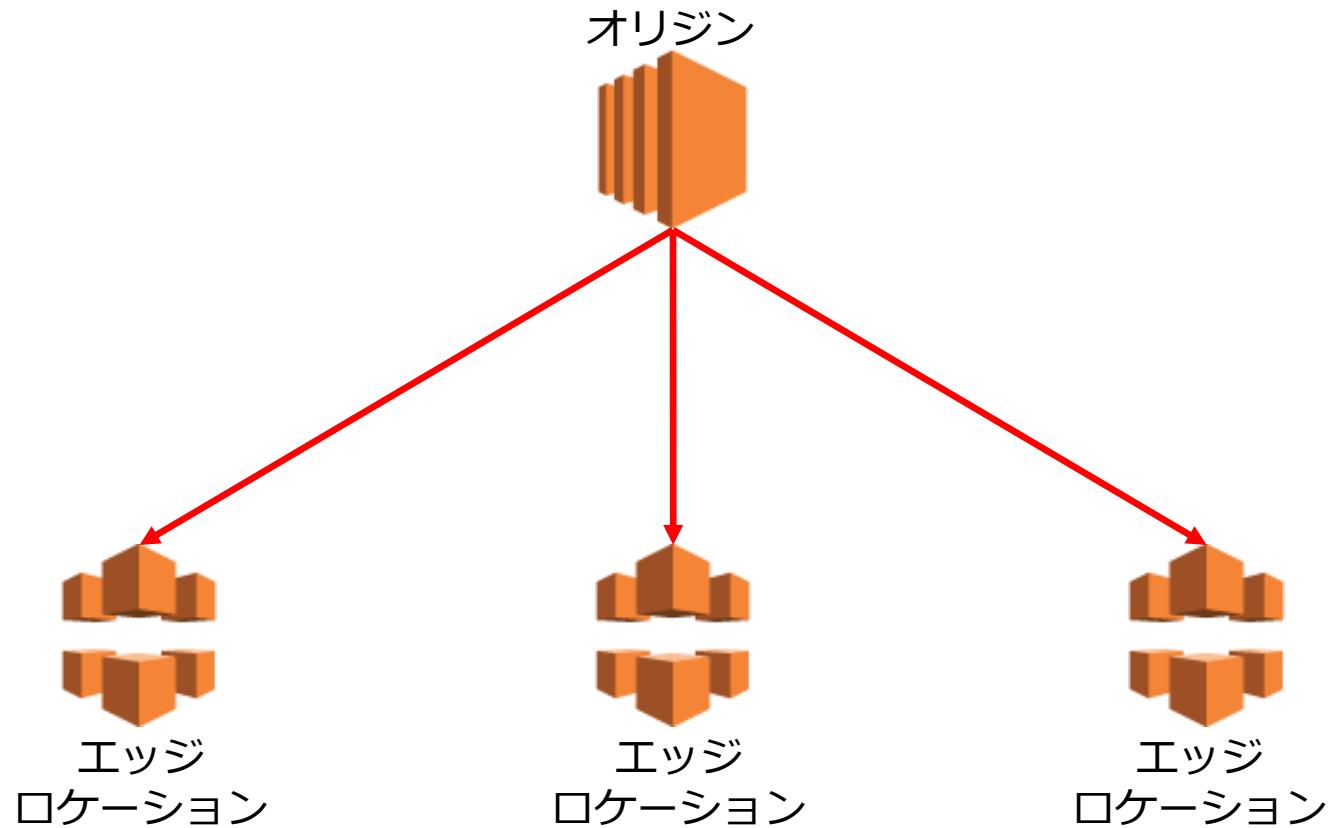
大手ニュースメディアは、AWSを利用したニュース配信アプリケーションを構築しています。アプリケーションはEC2インスタンスとS3を利用して構成されており、S3バケットに蓄積したビデオデータに基づいてストリーミング配信を実施します。このソリューションではビデオデータのアップロードと配信リクエストが頻発するため、あなたはソリューションアーキテクトとして、リクエスト処理のパフォーマンスを向上させたいと考えています。

この問題に対処するために実施すべきソリューションは次のうちどれですか？（2つ選択してください）

- 1) S3バケットをオリジンとしてCloudFrontディストリビューションを構成する。
- 2) Route53による地域制限設定を導入して、地域ごとの配信を最適化する。
- 3) S3バケットのS3 Transfer Accelerationを有効にする。
- 4) ELBを追加してクロスゾーン負荷分散を有効化する。
- 5) EC2インスタンスをストレージ最適化インスタンスに変更する。

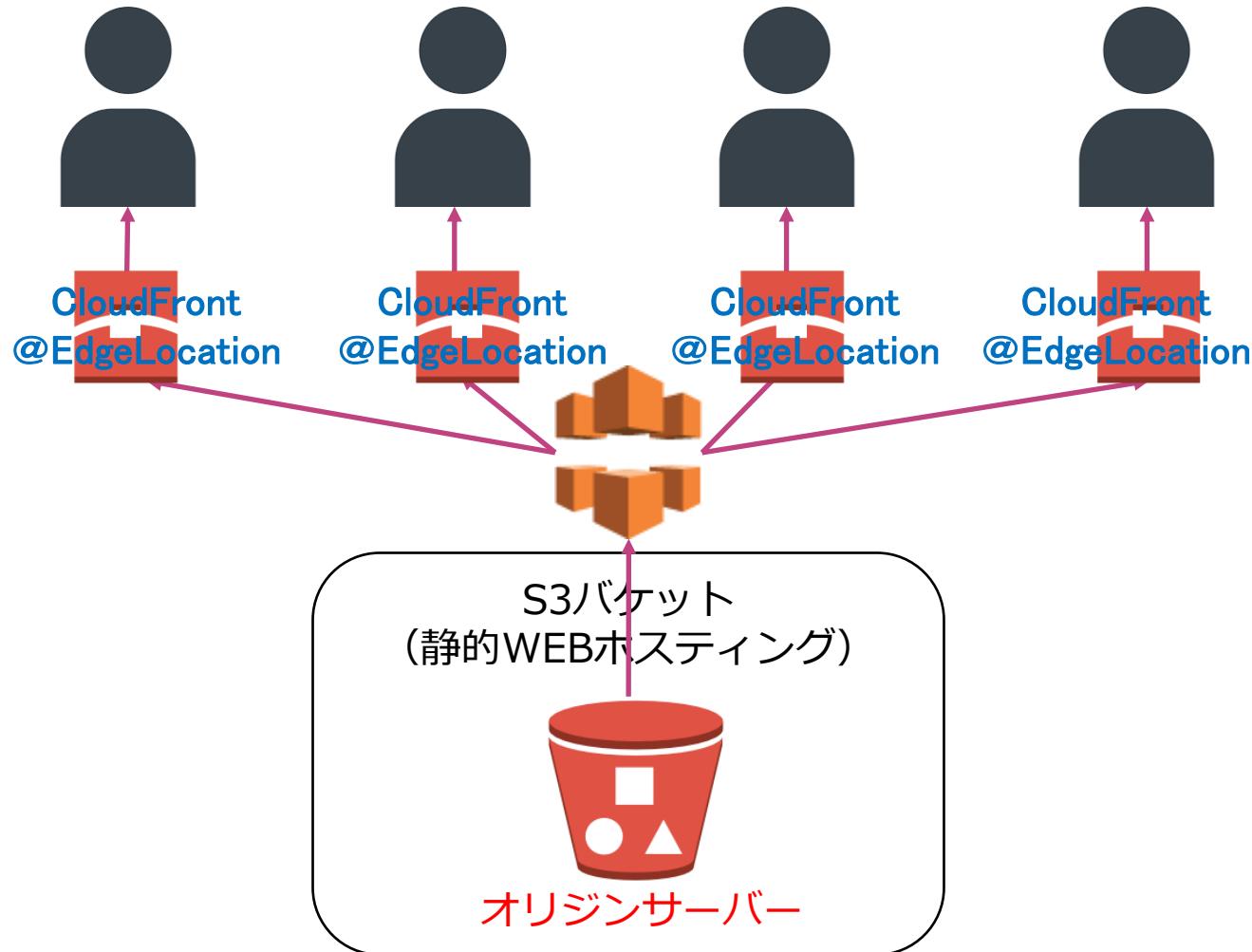
# CloudFrontの構成

ユーザーに近い位置にあるエッジロケーションから配信する  
シンプルなアーキテクチャ



# CloudFrontの構成

S3の静的WEBホスティングなどに対してCloudFront配信を構成するのが基本構成の1つ



# [新Q] CloudFrontのカスタムオリジン構成

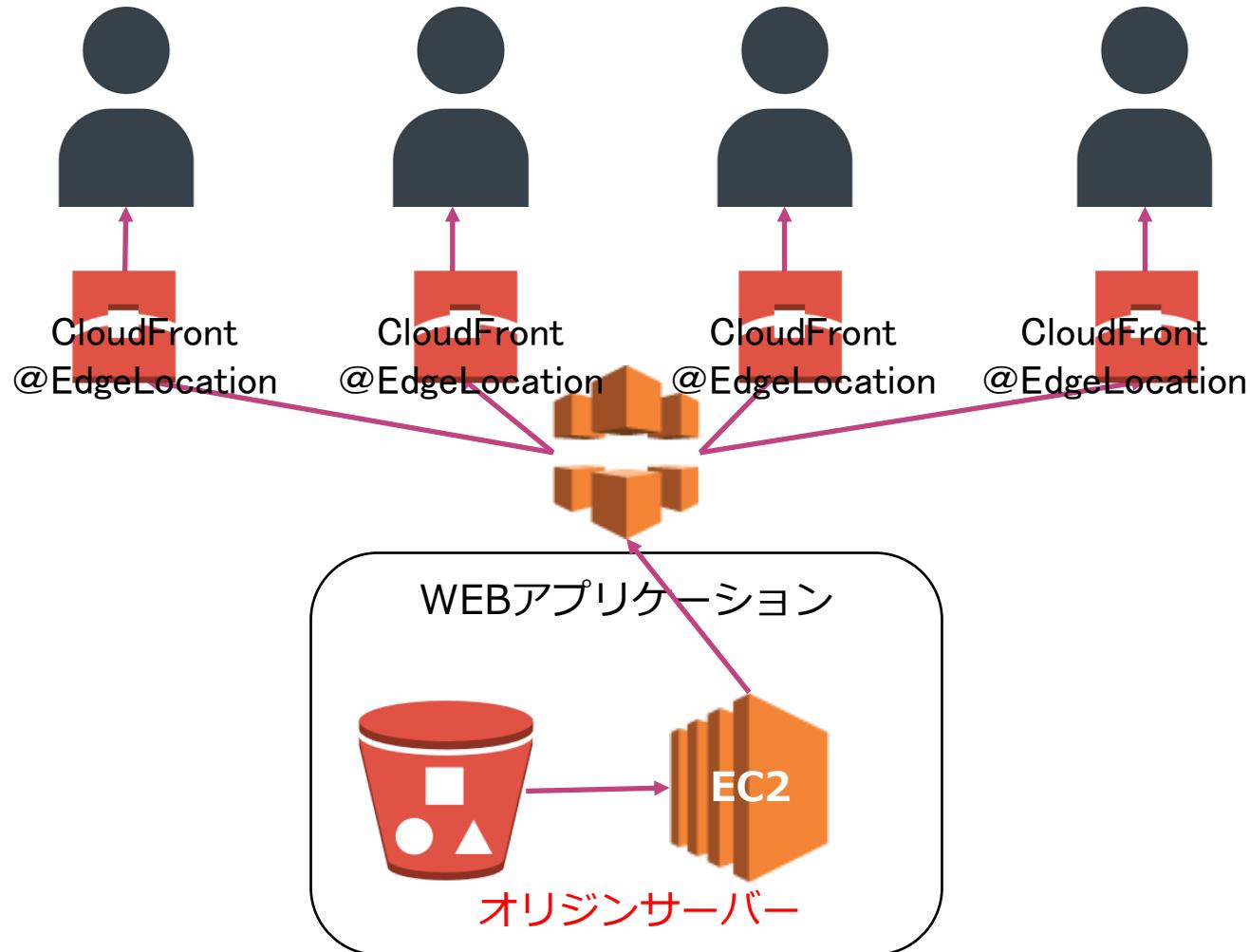
ある企業の、eコマースサイトはアメリカにあるオンプレミスサーバーを利用してホストされています。新規事業としてヨーロッパでもeコマースサイトを展開して製品を販売する予定であり、ヨーロッパのユーザーへのサイト提供を最適化したいと考えています。その際は、データセキュリティの観点で、サイトのバックエンドはアメリカのサーバーを継続的に利用します。このヨーロッパ向けのeコマースサイトは数日中に展開する必要があります。

この要件を満たすために、ソリューションアーキテクトはどうするべきでしょうか。

- 1) ヨーロッパのリージョンにAmazon EC2インスタンスを起動して、クロスリージョンレプリケーションを実施する。
- 2) オンプレミスサーバーをポイントするRoute53を設定して、位置情報ルーティングを実施する。
- 3) オンプレミスサーバーをポイントするカスタムオリジンとして、Amazon CloudFrontを設定する。
- 4) オンプレミスサーバーをポイントするRoute53を設定して、地理的近接ルーティングを実施する。

# CloudFrontの構成

WEBアプリケーションのEC2インスタンスをオリジンサーバーとする構成も基本



# [Q]オリジンの冗長化

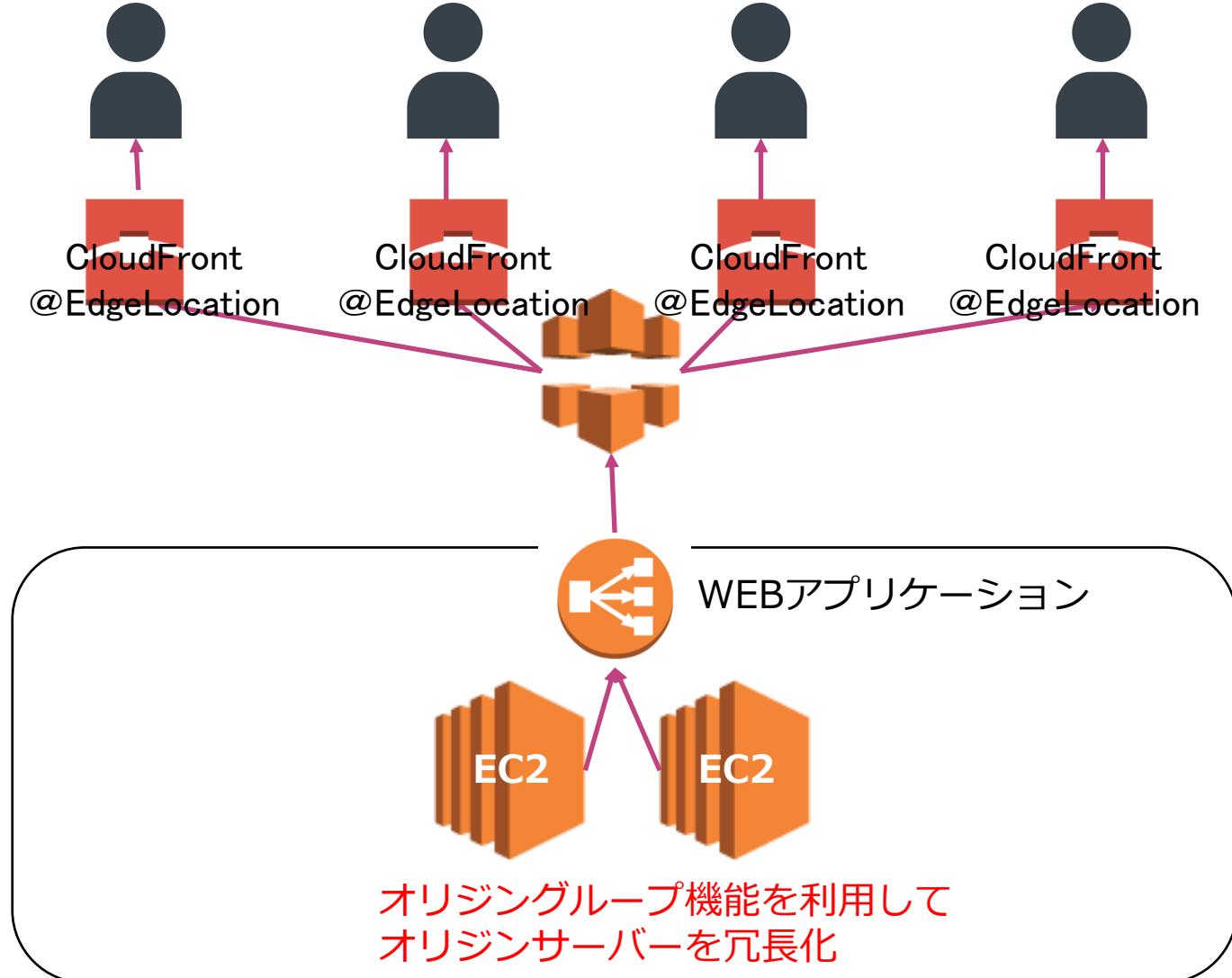
大手ニュースメディアは、AWSを利用した音楽配信アプリケーションを構築しています。アプリケーションは単一AZにある単一のEC2インスタンスを利用して構成されており、各楽曲は、EC2インスタンスをオリジンサーバーとして構成したCloudFrontを使用して配信されています。このアプリケーションの配信処理をさらに高可用にすることが求められています。

ソリューションアーキテクトとして、どのように対応しますか？

- 1) 既存のEC2インスタンスにELBを接続してELBをオリジンサーバーとして構成する。
- 2) Amazon S3を使用してWebアプリケーションの動的コンテンツを提供し、S3バケットをオリジンサーバーとして構成する。
- 3) 異なるアベイラビリティーゾーンにデプロイされた2つ以上のEC2インスタンスをオリジンサーバーとして構成する。
- 4) 既存のEC2インスタンスにAuto Scalingグループを追加して、Auto scaling をオリジンサーバーとして構成する。

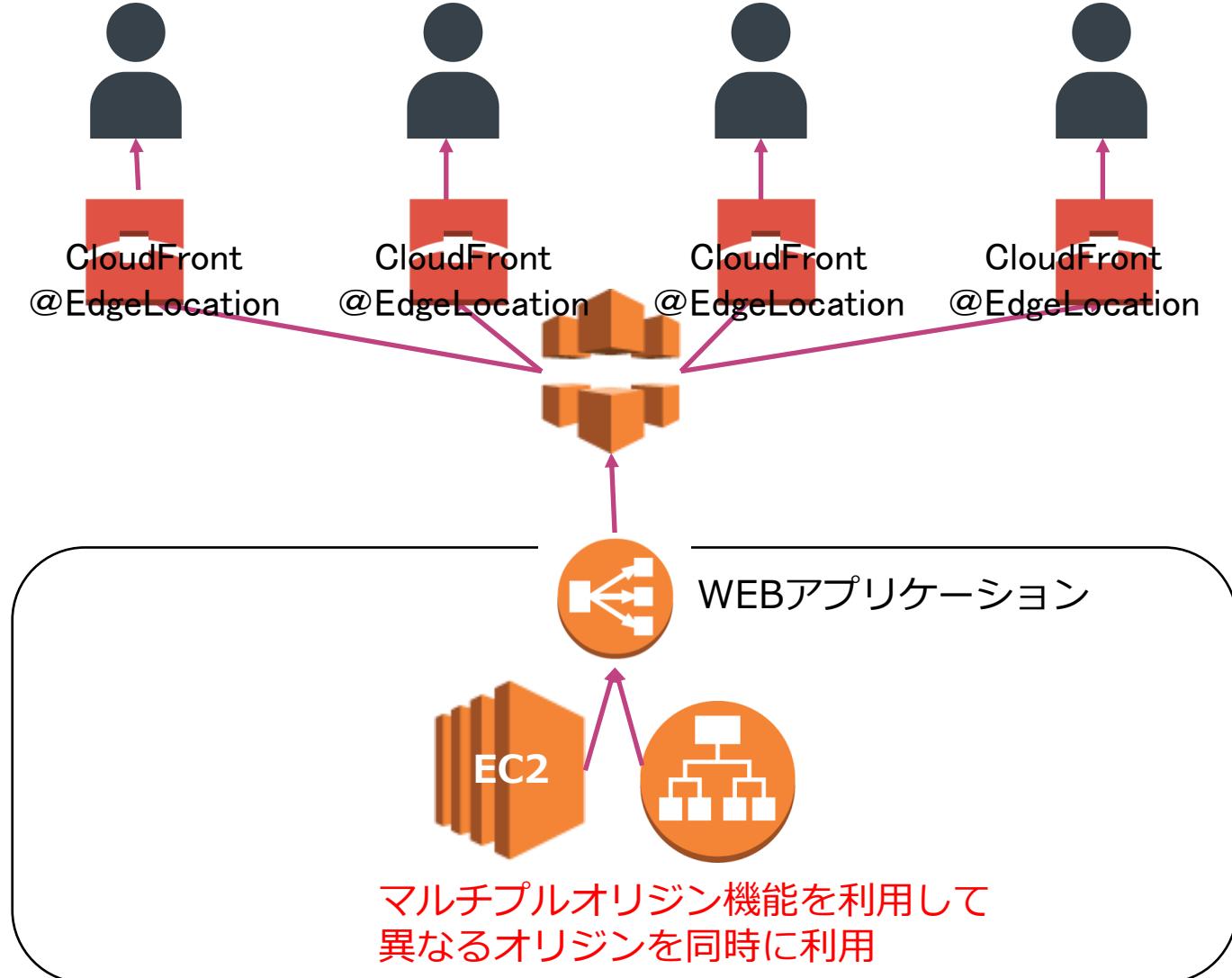
# CloudFrontの構成

オリジングループで同じオリジンサーバーを複数利用して、冗長化することができる。



# CloudFrontの構成

マルチプルオリジン機能を利用して、異なるオリジンを同時に設定することも可能



## [Q]エッジロケーション

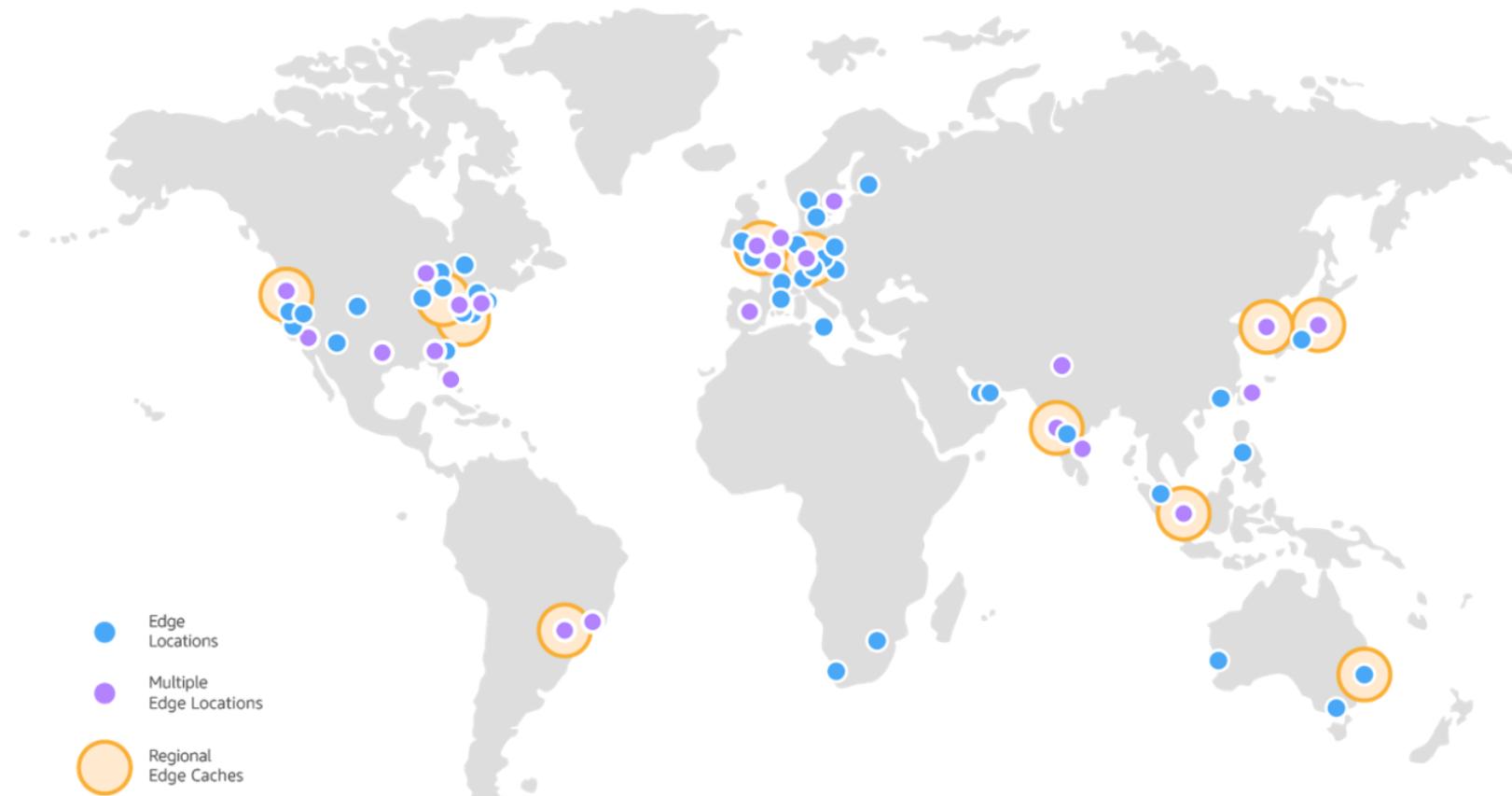
大手メディア企業は、Amazon S3 バケットにあるビデオデータにもとづいて顧客にニュースを提供しています。同社の顧客は世界中にあり、ピーク時には高い需要が発生します。欧州の各リージョンでは、ピーク時のダウンロード速度が遅く HTTP500 エラーが多発しているとのクレームが多発しており、あなたはソリューションアーキテクトとして、改善策を依頼されました。

この問題に対応するための最適なソリューションを選択してください。

- 1) Amazon Route 53 加重ルーティングポリシーを使用して、欧州地域へのルーティングの加重比率を高める。
- 2) DynamoDB の DAX クラスターを S3 バケットの前に配置して、高速な配信処理を可能にする。
- 3) ElastiCache クラスターを S3 バケットの前に配置して、高速な配信処理を可能にする。
- 4) CloudFront を使用してウェブコンテンツをキャッシュして、コンテンツ配信にすべてのエッジロケーションを使用する

# エッジネットワーク

AWSは世界中のエッジロケーションを利用したグローバルなコンテンツ配信ネットワークを利用することができる



参照 : <https://aws.amazon.com/jp/cloudfront/features/?nc=sn&loc=2>

# [Q]リージョナルエッジキャッシュ

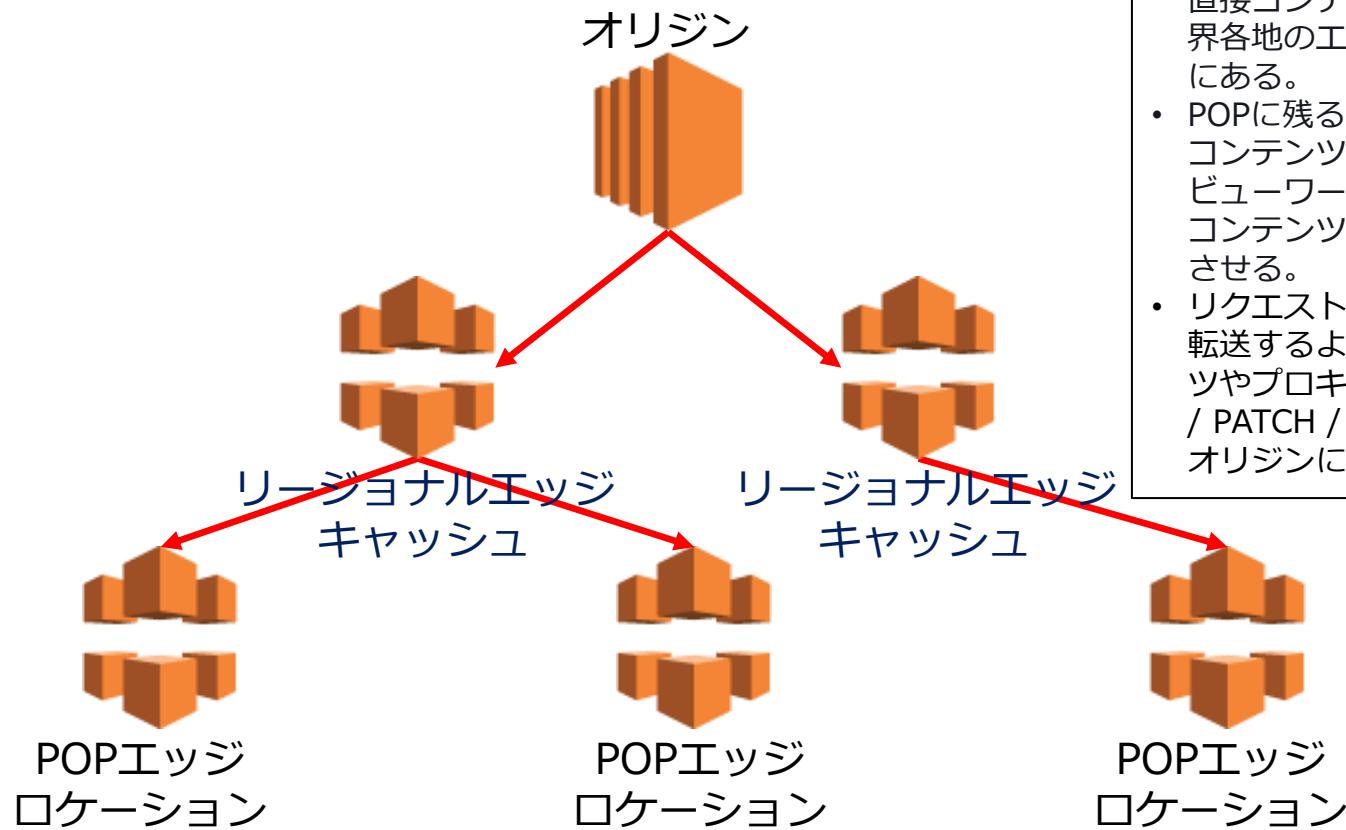
大手メディア企業は、Amazon S3 バケットにビデオデータを蓄積して、CloudFront 配信を構成して、顧客にニュースを提供しています。同社の顧客は世界中にあり、ピーク時には高い需要があります。AWS のコンテンツ配信ネットワーク（CDN）は、デフォルトで多層キャッシュを提供します。リージョナルエッジキャッシュは、オブジェクトがエッジにまだキャッシュされていない場合に、レイテンシーを改善し、オリジンサーバーの負荷を軽減してくれます。しかしながら、一部のコンテンツはリージョナルエッジキャッシュを利用していないようです。

リージョナルエッジキャッシュではなくオリジンに直接移動するコンテンツタイプはどれですか？（2つ選択してください）

- 1) リクエスト時にすべてのヘッダーを転送するように構成されたコンテンツ
- 2) ユーザー側でオリジンへの直接アクセスリクエストが発せられたコンテンツ
- 3) カスタムヘッダーを利用したアクセス制御がされている全てのコンテンツ
- 4) プロキシメソッドPUT / POST / PATCH / OPTIONS / DELETEはオリジンに直接移動する。
- 5) 全てのTTLが0に設定されたコンテンツ

# CloudFrontの構成

リージョナルエッジキャッシュが追加されより効率的な配信処理が可能になった



- リージョナルエッジキャッシュは、オリジンサーバーと、ビューワーに直接コンテンツを提供するPOP（世界各地のエッジロケーション）の間にある。
- POPに残るような人気が十分にないコンテンツでも、中間地点としてビューワーの近くに配置して、そのコンテンツのパフォーマンスを向上させる。
- リクエスト時にすべてのヘッダーを転送するように構成されたコンテンツやプロキシメソッドPUT / POST / PATCH / OPTIONS / DELETEはオリジンに直接移動する。

CloudFront ポイントオブプレゼンス (POP) は、人気のあるコンテンツをなるべくユーザーの近くに配置されたエッジロケーション



# Distribution設定

CloudFrontは配信設定により要件に応じた最適な配信設定を実施することが必要

- S3エンドポイントやIPアドレスをCloudFrontに設定する。
- マネジメントコンソールやAPIによりCloudFrontを構成する。
- WEB Distribution (RTMP Distributionは廃止) を選択する
- 使用量が最大40Gbps／10万RPS超は上限緩和申請を実施する
- Route53により独自ドメインを指定してURLを構成できる。



# Distribution設定

現在はWEBディストリビューションのみを利用して構成する

## WEB Distribution

- 通常のHTTPプロトコルを利用したWEB配信をする際に利用
- HTTP1.0/ HTTP1.1/ HTTP2に対応
- オリジンはS3バケット／MediaPackage チャネル／HTTP サーバーを設定
- HTTPやHTTPSを使用した静的および動的なダウンロードコンテンツ配信
- Apple HTTP Live Streaming (HLS)や Microsoft Smooth Streamingなど、さまざまな形式のビデオオンデマンド

## RTMP Distribution (廃止)

- RTMP形式配信の際に利用
- Adobe Media ServerとAdobe Real-Time Messaging Protocol (RTMP)を使用してメディアファイルをストリーミング
- S3バケットをオリジン設定
- クライアントはメディアファイル／メディアプレーヤー (JW Player、Flowplayer、Adobe Flash)を利用



# [Q]キャッシュ保持期間の設定

あなたはソリューションアーキテクトとして、WEBアプリケーションの運用管理を行っています。このアプリケーションはグローバルに利用されているため、CloudFrontによる配信処理を行っています。現在、キャッシュされるべきオブジェクトがエッジロ케ーションにないため、オリジンサーバーへのアクセスが頻繁しています。この問題は、一般的に頻繁に利用されるオブジェクトに対しても発生します。

次のうち、この問題の最も可能性の高い原因を選択してください。

- 1) キャッシュすべきオブジェクト設定の指定範囲が狭い
- 2) Cache-Controlのmax-ageディレクティブが低い値に設定されている
- 3) キャッシュするべきファイルサイズがCloudFront標準を超過している。
- 4) SSL証明設定でキャッシュできていない。

# キャッシュ保持期間の設定

キャッシュ対象を決定した上で、キャッシュの利用頻度を予測してキャッシュ保持期間を設定することが重要

<b>Minimum TTL (最小 TTL)</b>	<ul style="list-style-type: none"><li>CloudFront がオリジンに別のリクエストを送るまでに、オブジェクトを CloudFront キャッシュに保持する最小期間 (秒) を指定</li><li>デフォルト値は 0 (秒)</li></ul>
<b>Maximum TTL (最大 TTL)</b>	<ul style="list-style-type: none"><li>オブジェクトが更新されたかどうかを CloudFront がオリジンにクエリするまでに、オブジェクトを CloudFront キャッシュに保持する最大期間 (秒) を指定する。</li><li>デフォルト値は 31,536,000 (秒)つまり 1 年</li></ul>
<b>Default TTL (デフォルト TTL)</b>	<ul style="list-style-type: none"><li>CloudFront がオリジンに別のリクエストを送るまでオブジェクトを CloudFront キャッシュに保持するデフォルト期間 (秒) を指定</li><li>デフォルト値は 86,400 (秒)、つまり 1 日</li></ul>

# キャッシュ保持期間の設定

TTLとCache-Control および Expires ヘッダーを使用して、オブジェクトをキャッシュに保持する期間を制御できます

キャッシュ対象設定		<ul style="list-style-type: none"><li>□コンテンツ利用データ分析などを実施して、静的コンテンツ／動的コンテンツへのキャッシュ対象URLを設定する</li></ul>
キャッシュの有効期限	TTL	<ul style="list-style-type: none"><li>□CloudFront配信時に設定されるキャッシュ保持期間</li></ul>
	Expires ヘッダー	<ul style="list-style-type: none"><li>□Cache-Control ヘッダーのExpires ヘッダー</li><li>□キャッシュの期限切れ日を設定する</li><li>□【例】Expires: Thu, 01 Dec 1994 16:00:00 GMT</li></ul>
Cache-Control max-age ヘッダー		<ul style="list-style-type: none"><li>□CloudFront がオリジンサーバーからオブジェクトを再度取得するまでにオブジェクトをキャッシュに保持する期間(秒)を指定できる。</li><li>□最小の有効期限切れ時間は、ウェブディストリビューションで 0 秒、RTMP ディストリビューションで 3600 秒。最大値は 100 (年)</li></ul>

# キャッシュ保持期間の設定

キャッシュ期限を複数の要素で設定した場合に、複雑な結果となるため、矛盾する設定はなるべく避ける。

## 【設定が混在する場合の反映シナリオ】

- [Maximum TTL (最大 TTL)] に 5 分 (300 秒) を設定し、Cache-Control max-age ヘッダーに 1 時間 (3600 秒) を設定した場合、CloudFront は 1 時間ではなく 5 分間、オブジェクトをキャッシュする。
- Cache-Control max-age ヘッダーに 3 時間を設定し、Expires ヘッダーを 1 か月に設定した場合、CloudFront は 1 か月ではなく 3 時間オブジェクトをキャッシュする。
- [Default TTL (デフォルト TTL)]、[Minimum TTL (最小 TTL)]、および [Maximum TTL (最大 TTL)] に 0 秒を設定した場合、CloudFront は常にオリジンからの最新コンテンツがあることを確認する。

Reference: [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudFront/latest/DeveloperGuide/Expiration.html#expiration-individual-objects](https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/Expiration.html#expiration-individual-objects)

# [Q]キャッシュの活用

A社はAWSで多言語ウェブサイトをホストしています。WebサイトはCloudFrontを使用して提供されます。言語はHTTPリクエストはhttp://pintor.cloudfrontで指定されて以下のように表示されます。

```
http://pintor.cloudfront.net/main.html?language=de  
http://pintor.cloudfront.net/main.html?language=en  
http://pintor.cloudfront.net/main.html?language=jp
```

net / main.html ? language = jpキャッシュデータは日本語表示サイトとして表示されるようにCloudFront側で設定する必要があります。

この要件を達成するための設定方法を選択してください。

- 1) クエリ文字列パラメータを設定して、net / main.html ? language = jpキャッシュデータは日本語表示サイトとして表示される設定を行う。
- 2) 動的コンテンツ設定を利用して、net / main.html ? language = jpキャッシュデータは日本語表示サイトとして表示される設定を行う。
- 3) キャッシュオリジン設定を利用して、net / main.html ? language = jpキャッシュデータは日本語表示サイトとして表示される設定を行う。
- 4) フォワードクッキーを利用して、net / main.html ? language = jpキャッシュデータは日本語表示サイトとして表示される設定を行う。

# キャッシュの活用

キャッシュコントロールによりキャッシュヒット率を上昇させて効果的なキャッシュ活用を可能にする

## パラメーター値の完全一致

- URLとフォワードオプション機能 (header/Cookie/Query Strings) のパラメーター値の完全一致でキャッシュが指定される仕組み
- 単一ファイルのキャッシュは最大20GB
- GET/HEAD/OPTIONリクエストを対象

## キャッシュの無効化

- キャッシュが期限切れになる前に無効化することが可能
- 必要のないキャッシュを無効化することで効果的な利用を可能にする
- コンテンツ毎に最大3000個まで無効化パスを指定できる
- ワイルドカードを利用して最大15個まで無効化パスリクエストが指定可能

# [Q] CloudFrontの利用料

大手画像配信サイトはAWS上に構築されています。サイトの運営会社は画像配信の仕組みを効率化するためにCDNの利用を検討しています。そこで、あなたはソリューションアーキテクトとして、CloudFrontを利用したコンテンツ配信に必要なコストを算出して報告することになりました。

次のうち、CloudFrontのコスト算出の要素を選択してください。（2つ選択してください。）

- 1) リージョン数
- 2) グローバルエッジロケーションの数
- 3) データ転送アウト
- 4) リクエスト数
- 5) 設定されたキャッシュ数

# CloudFrontの利用コスト

主にリクエストとデータ転送アウトに対して料金が発生

## リクエスト

- HTTP/HTTPS リクエスト
- ORIGIN SHIELD リクエスト
- 無効リクエスト
- フィールドレベル暗号化リクエスト
- リアルタイムログリクエスト

## データ転送アウト

- インターネットへのリージョンデータ転送アウト (GB 単位)
- オリジンへのリージョン内データ転送アウト (GB 単位)

## 専用 IP カスタム SSL

- CLOUDFRONT ディストリビューションに関連する、専用 IP カスタム SSL 証明書を利用したSSL利用料金
- 1 つ以上の CloudFront ディストリビューションに関連付けられた各独自 SSL 証明書ごとに、毎月600 USD

## [Q] Gzip圧縮機能

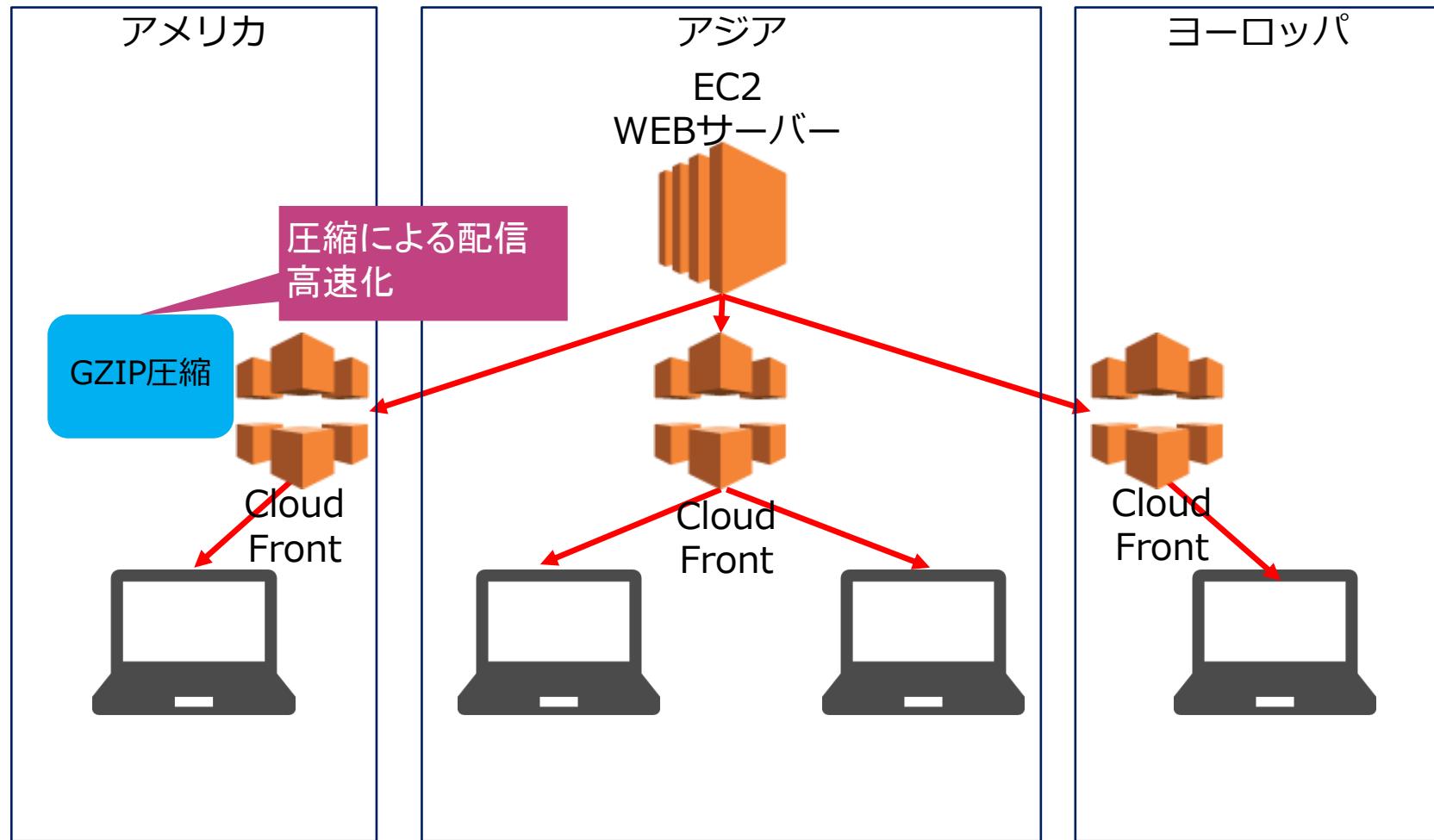
S3に静的コンテンツを保存した上で、CloudFrontを利用したグローバル配信を実施しています。CloudFrontは配信先が多いことで、利用料金が想定より高くなっています。これが問題となっています。

CloudFrontのコスト削減効果のある方法はどれでしょうか？

- 1) エッジロケーションによるファイル圧縮処理を実施する。
- 2) CloudFrontによるキャッシュ保持期間を短縮する。
- 3) Lambda@エッジによるファイル圧縮処理を実施する。
- 4) オリジンサーバーに設定したS3による配信コンテンツの圧縮処理を実施する。

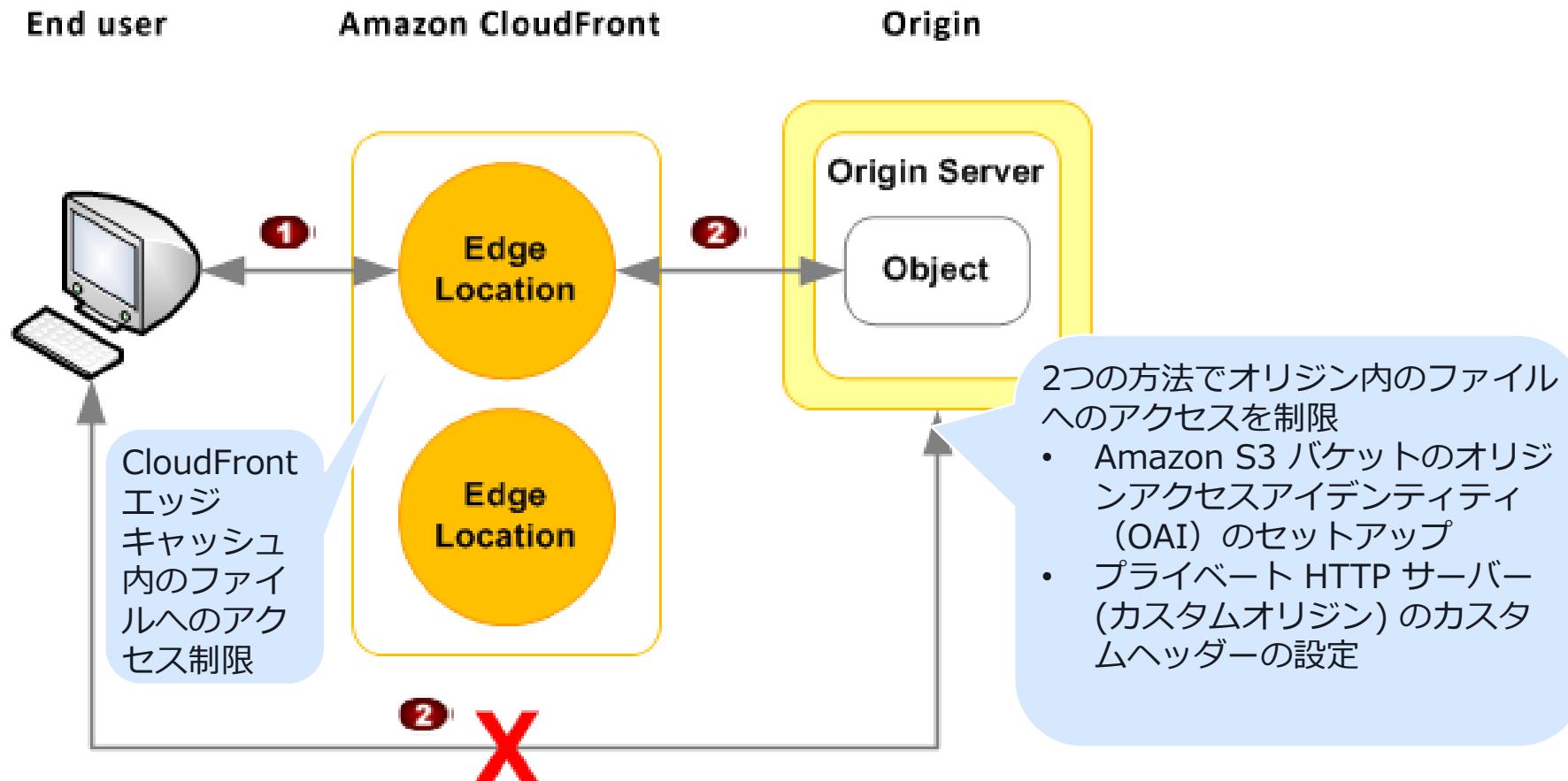
# Gzip圧縮機能

エッジ側でコンテンツをGZIP圧縮してより高速に配信可能



# アクセス制限

署名付きURLと署名付きCookieにより配信コンテンツへのアクセスを詳細に制御する。



# [新Q]オリジンへのアクセス宣言

ある企業は、AWSにホストされた画像共有アプリケーションを運用しています。ストレージ層にAmazon S3バケットを使用して、Amazon CloudFrontディストリビューションを使って、すべてのファイルを配信したいと考えています。その際は、S3 オブジェクトURLに直接アクセスできないようにする必要があります。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。（2つ選択してください。）

- 1) 各S3バケットにCloudFrontディストリビューションエンドポイントへのアクセスのみに読み取り権限を付与するバケットポリシーを設定する。
- 2) S3バケットのオブジェクトに対する読み取り権限を付与したIAMロールを作成して、CloudFrontに割り当てる。
- 3) CloudFrontディストリビューションIDをプリンシパルとしたS3バケットポリシーを設定する。
- 4) オリジンアクセスアイデンティティ（OAI）をCloudFrontディストリビューションに割り当てて、OAIのみに読み取り権限があるようにS3バケットポリシーを設定する。
- 5) オリジンアクセスコントロール（OAC）をCloudFrontディストリビューションに割り当てて、OAIのみに読み取り権限があるようにS3バケットポリシーを設定する。

# オリジンへのアクセス制限

OA1でS3バケットへの、カスタムヘッダーでカスタムオリジンへのアクセスを制限する。

## OA1

- OA1はS3 バケットへのアクセスを CloudFront からのリクエストに絞るための仕組み
- オリジンアクセスアイデンティティ (OA1) と呼ばれる特別な ユーザーを作成し、そのユーザーに限定してアクセスを許可
- CloudFrontはOA1 を使用してバケット内のファイルにアクセスする。

## カスタムヘッダー

カスタムヘッダーをオプションで設定して、カスタムオリジンへのアクセスを制限する仕組み

## ビューワープロトコル ポリシー

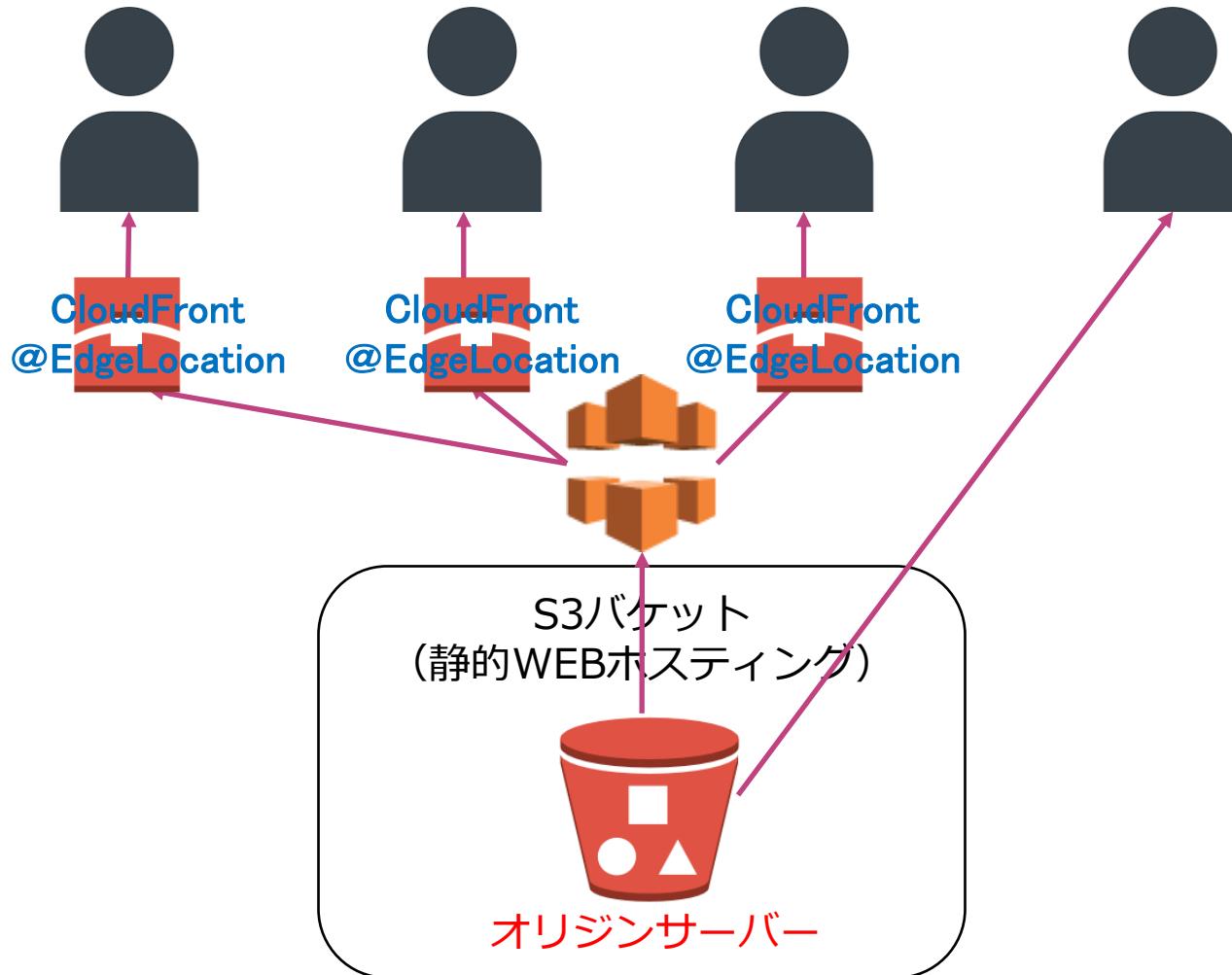
ビューワーが CloudFront にアクセスするのに HTTPS を使用しなければならないようにディストリビューションを設定

## オリジンプロトコル ポリシー

CloudFront がビューワーと同じプロトコルを使用してリクエストをオリジンに転送するように、ディストリビューションを設定

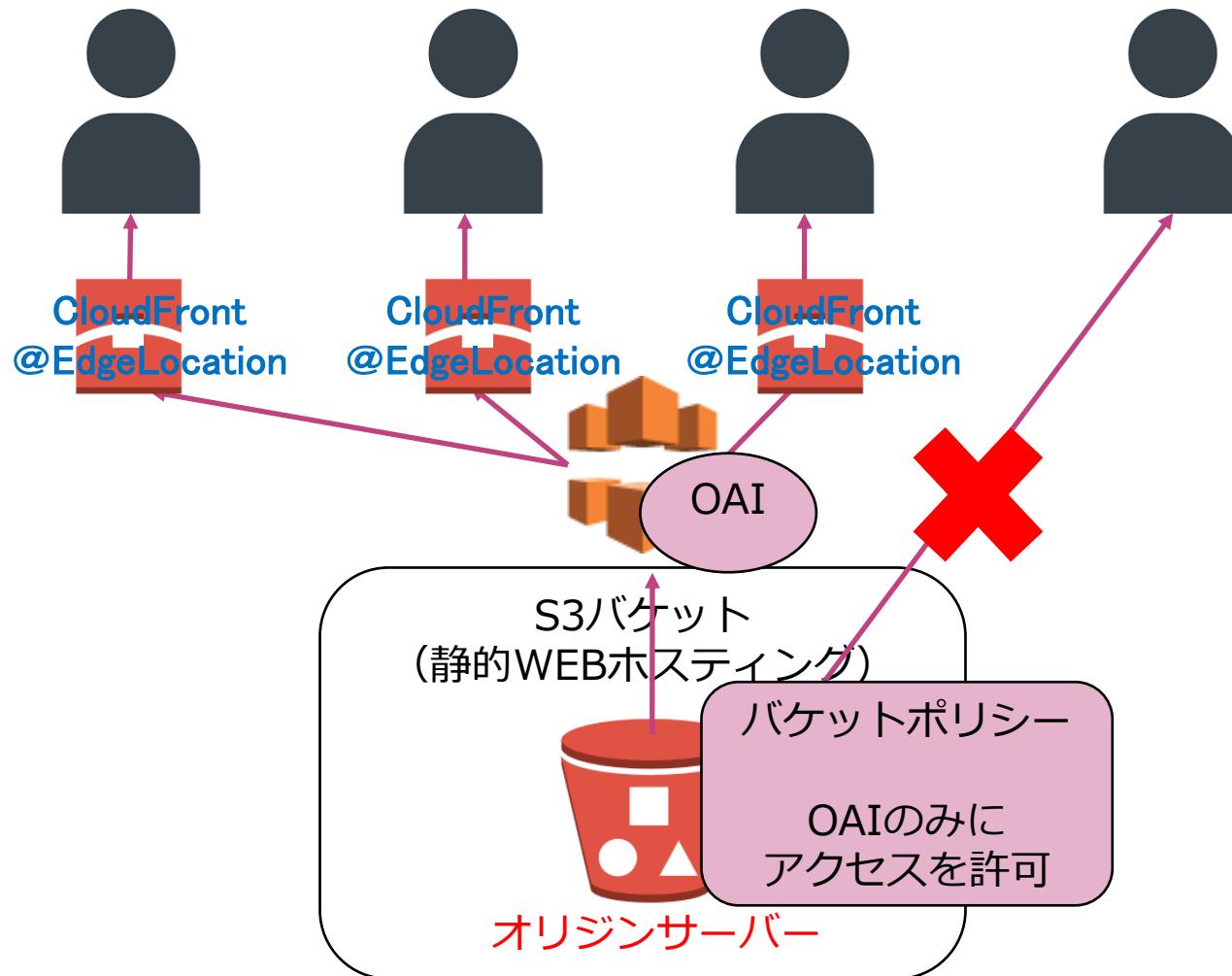
# オリジンへのアクセス制限

オリジンアクセスアイデンティティ (OAI) でS3バケットへの、カスタムヘッダーでカスタムオリジンへのアクセスを設定する。



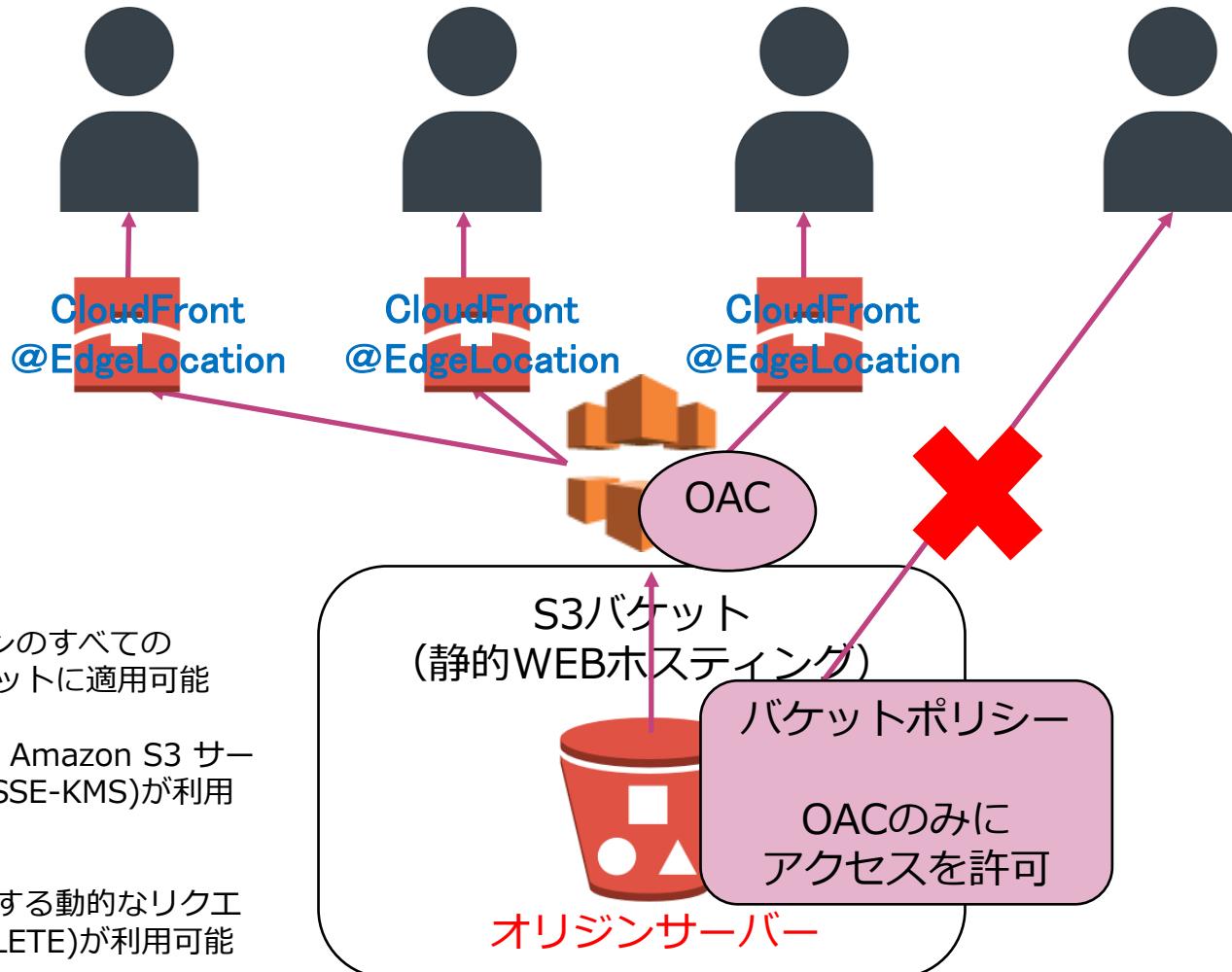
# オリジンへのアクセス制限

オリジンアクセスアイデンティティ (OAI) でS3バケットへの、カスタムヘッダーでカスタムオリジンへのアクセスを設定する。



# オリジンへのアクセス制限

オリジンアクセスコントロール (OAC) でS3バケットへの、カスタムヘッダーでカスタムオリジンへのアクセスを設定する。



# [Q]キャッシュのアクセス制限

大手画像配信サイトはAWS上に構築されています。画像配信の仕組みを効率化するためにCDNの利用を検討しています。あなたはソリューションアーキテクトとして、CloudFrontを使用して、効率的にコンテンツを配信する計画をたてました。その際には会員登録されたエンドユーザーのみがコンテンツが利用できるようにする必要があります。

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) CloudFrontの署名付きURLを使用する
- 2) CloudFrontで署名されたCookieを使用する
- 3) CloudFrontとカスタムオリジン間の通信にHTTPSが必要
- 4) CloudFrontとS3オリジン間の通信にHTTPSが必要
- 5) CloudFrontでOAIを使用する

# キャッシュのアクセス制限

署名付きURLと署名付きCookieでキャッシュに保持したコンテンツにアクセスできるユーザーを制限する

## 署名付きURL

- ・コンテンツに直接アクセスするURLではなく、署名付きURLからのみアクセスさせる。
- ・署名付きCookieはRTMPディストリビューションではサポートされていないため署名付きURLを利用する。
- ・**個別のファイル**(アプリケーションのインストールダウンロード)へのアクセスを制限する場合に利用する。
- ・ユーザーがCookieをサポートしていないクライアント(カスタムHTTPクライアントなど)を使用している場合に利用する。

## 署名付きCookie

- ・コンテンツに直接アクセスするURLではなく、署名付きCookieからのみアクセスさせる。
- ・**複数の制限されたファイル**(HLS形式の動画のすべてのファイルやウェブサイトの購読者の領域にあるすべてのファイルなど)へのアクセスを提供する場合に利用する。
- ・現在のURLを変更したくない場合に利用する。

# [Q] CloudFront地域制限

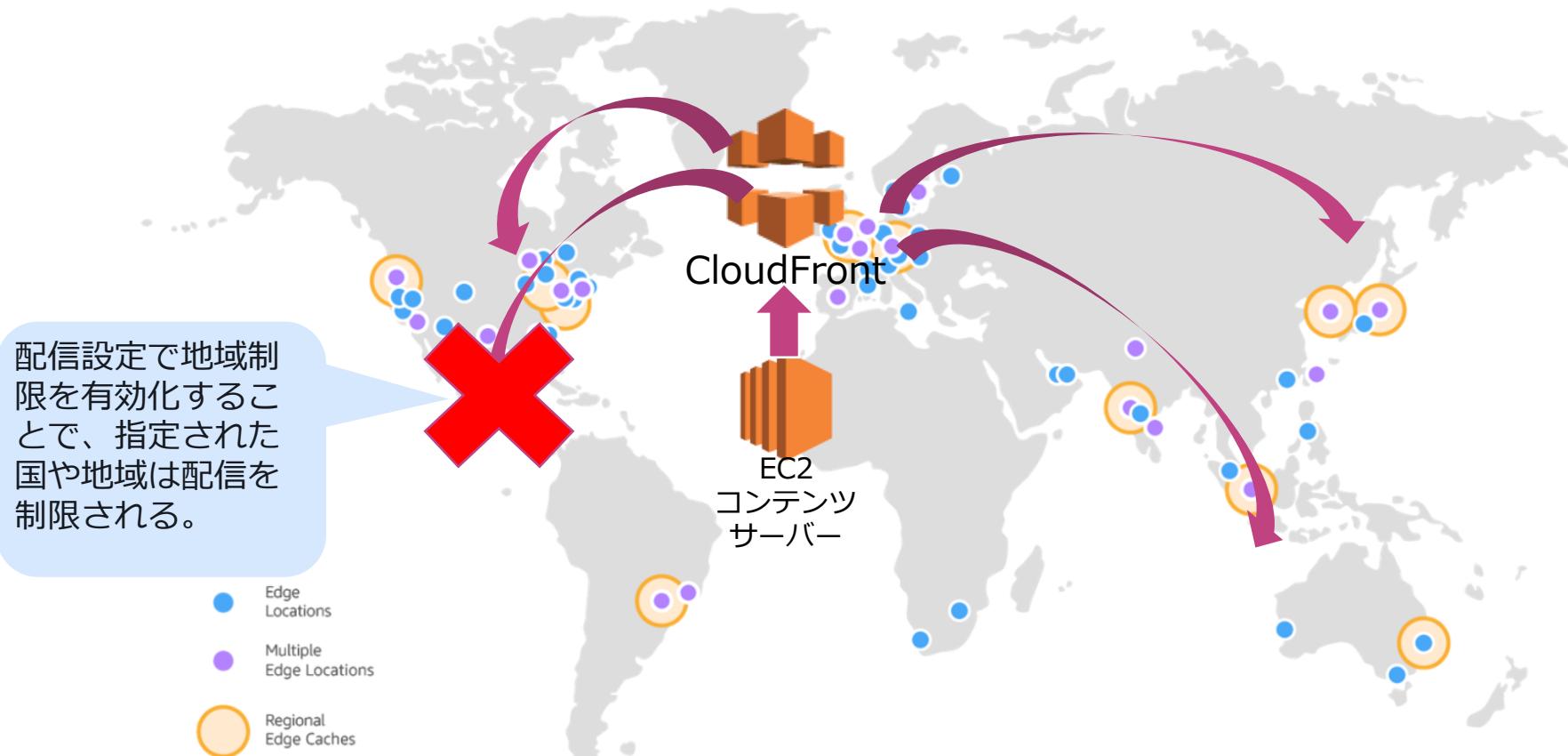
大手ニュース配信企業はニュース配信アプリケーションをAWS上に構築しています。ユーザーはグローバルに存在しており、グローバルにコンテンツを配信します。アプリケーションは、ALBの背後にあるプライベートサブネットに設置されたEC2インスタンスのフリートを使用しています。中国からの情報制限があり、中国からのアクセスをブロックする必要があります。

この要件を満たすための最も簡単な方法は何ですか？

- 1) ネットワークACLを使用して、特定の国に関連付けられたIPアドレス範囲をブロックする。
- 2) ELBのセキュリティグループを変更して、ブロックされた国からの着信トラフィックを拒否する。
- 3) CloudFrontを使用してコンテンツを提供し、特定の国からのアクセスをブロックする。
- 4) EC2インスタンスのセキュリティグループを変更して、ブロックされた国からの着信トラフィックを拒否する。

# CloudFront地域制限

地域制限機能を利用して、特定の場所にいるユーザーからのアクセスを制限する。



参照 : <https://aws.amazon.com/jp/cloudfront/features/?nc=sn&loc=2>

# [Q] ELBへのアクセス制限

ニュースメディアアプリケーションではCloudFrontを使用してWEBニュースを配信しています。このアプリケーションは、Elastic Load Balancer (ELB) の背後にあるEC2インスタンスで実行されています。ユーザーがCloudFrontを回避し、ELBを介してコンテンツに直接アクセスする機能を制限する必要があります。

この要件を満たすことができるソリューション要素の組み合わせを選択してください（3つ選択してください。）

- 1) CloudFront内部サービスのIPアドレスに変更があった場合はセキュリティグループに対して、Lambda関数によるIPアドレス設定を実施する。
- 2) ELBにVPCセキュリティグループを作成して、Lambda関数のアクセスを許可する。
- 3) Lambda関数にIAMロールを設定して、ELBへのアクセスを許可する。
- 4) CloudFront内部サービスのIPアドレスが変更されたときに自動的に更新する。
- 5) オリジンアクセスID (OAI) を作成し、それをディストリビューションに関連付ける。
- 6) ネットワークACLを使用してELBへのアクセスを制限する。

# ELBへのアクセス制限

CloudFrontではなくオリジンELBに直接アクセスするのを回避する設定も可能

## CloudFrontのIPレンジを利用

- CloudFrontのIPアドレスを指定して、指定したIPのみELBへのアクセスを許可する設定を行う方式
- CloudFrontのIPレンジを取得し、IPアドレスに変更があった場合はセキュリティグループのインバウンドルールを更新するLambda関数がIPを作成して、Lambda関数によるIPアドレス設定を実施する。
- IPアドレス上限の緩和申請が必要

## CloudFrontのカスタムヘッダーを利用

- 指定した文字列がカスタムヘッダに入ってない場合にELBへのアクセスを制限する方式
- CloudFrontのカスタムヘッダを利用して、任意のヘッダをELBオリジンに渡す

# [Q]暗号化

Webメディア企業は、CloudFrontを使用してWebサーバーをオリジンとして設定して、読み取りパフォーマンスを改善することにしました。最近になって、IT監査を実施し、CloudFrontを利用した配信処理が安全ではないため、OriginサーバーとCloudFrontへのデータ通信をセキュアにすることを要求されました。このOriginサーバーはELBではないことに留意が必要です。

この要件に対応するための最適な方法を選択してください。

- 1) AWS Certificate Manager (ACM) をオリジンとCloudFront側に利用して、HTTPSによるデータ通信を可能にする。
- 2) サードパーティのCA証明書をビューアーとCloudFront側に利用して、HTTPSによるデータ通信を可能にする。
- 3) サードパーティのCA証明書をオリジンとCloudFront側の両方に利用して、HTTPSによるデータ通信を可能にする。
- 4) AWS Certificate Manager (ACM) をビューアーとCloudFront側に利用して、HTTPSによるデータ通信を可能にする。

# 暗号化

CloudFrontはSSL/TLS暗号化とフィールドレベル暗号化を利用

## SSL/TLS

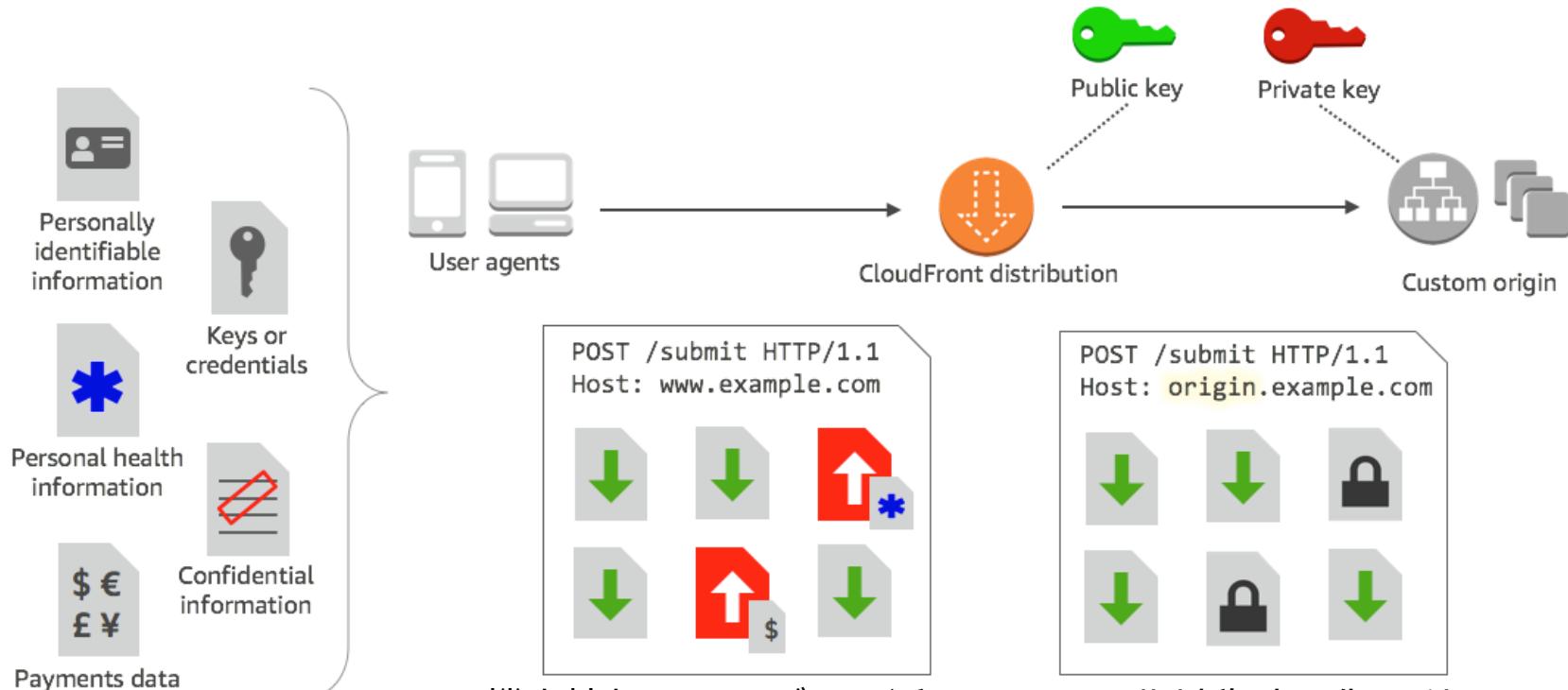
- AWS ACMと連携して証明を発行し、設定することが可能
- SSL証明書を設定してコンテンツ配信をHTTPSとする。
- CloudFront がビューアーと通信する際はビューアーが HTTPS を使用してファイルをリクエストするように Amazon CloudFront を設定
- オリジンからファイルを取得する際に CloudFront が HTTPS を使用するように設定して、CloudFront とオリジンとの通信を暗号化する
- SSLはPerfect Forward Secrecy (PFS) に対応

## フィールドレベル 暗号化

- HTTPS と共にセキュリティのレイヤーが追加される。
- システムの処理中に特定のデータに特定のアプリケーションのみがアクセスできるようにエンドツーエンドでデータを保護
- CloudFront のフィールドレベル暗号化では、公開鍵認証方式を利用した暗号化を実施

# フィールドレベル暗号化

CloudFront のフィールドレベル暗号化では、公開鍵認証方式を利用した暗号化を実施



- 機密情報をユーザーに近いエッジで非対称暗号化を利用して保護
- 暗号化したいPOSTリクエストのフィールドにおいて、暗号化するための公開鍵を指定して、リクエストを実施する。

# [Q]ログ取得

あなたの会社はCloudFrontを使用してホストされているWEB配信サービスを展開しています。ITセキュリティ部門はこのWeb配信を使用するアプリケーションのPCIコンプライアンスへの対応状況を監査しています。

コンプライアンス目標を確実に満たすための適切な対応を選択してください。（2つ選択してください。）

- 1) VPCフローログをCloudFrontに設定する。
- 2) CloudTrailをCloudFrontに設定する。
- 3) CloudFrontのキャッシュログを有効化する。
- 4) CloudFront APIに送信されるリクエストを取得する。
- 5) CloudFrontアクセスログを有効化する。

## その他のセキュリティ機能

様々な外部サービスと連携することで、セキュアなコンテンツ配信やアクセス管理が可能

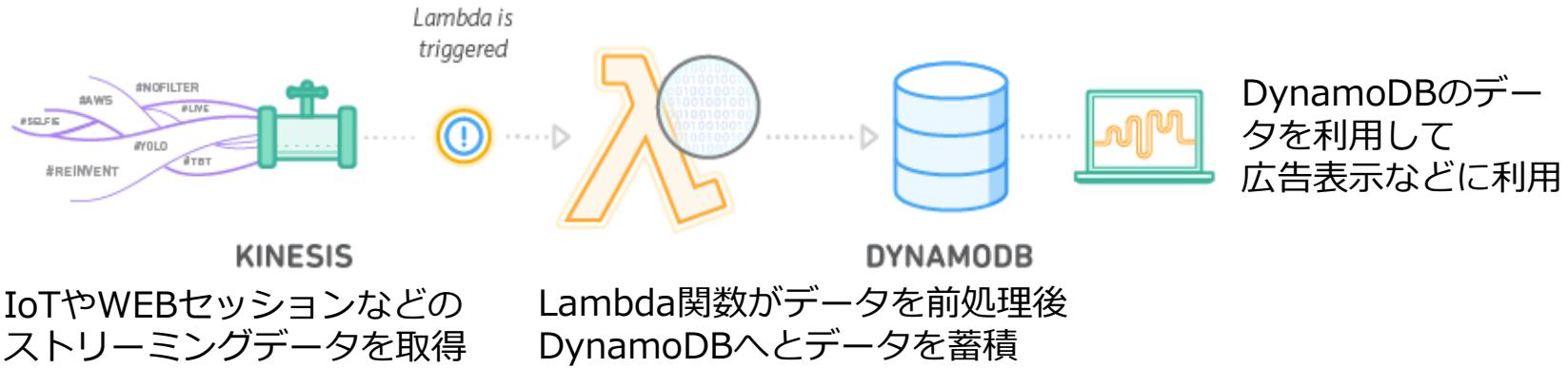
- AWS WAFによるファイアーウォールと連携し、ディストリビューションに対するウェブリクエストを許可、ブロックが可能。また、Referrer制限によるリンク参照禁止も可能
- AWS ShieldによるDDoS対応
- CloudTrailは、ユーザー、ロール、または AWS のサービスにより CloudFront で実行されたアクションレコードを提供
- CloudFront アクセスログは、ディストリビューションに対して行われたリクエストに関する詳細なレコードを提供

## DynamoDBの出題範囲

# DynamoDBとは何か？

ストリーミングデータを利用したリアルタイムデータ処理などに最適なデータベースとして利用するNoSQL型データベース

## DynamoDBの活用例



Reference: <https://aws.amazon.com/jp/dynamodb/>

# DynamoDBの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

DynamoDBの選択	✓ データベース要件が提示され、それに合った最適なデータベースとしてDynamoDBを選択する質問が出題される。
DynamoDBの特徴	✓ DynamoDBの性能や制約などを含めた特徴が問われる。 ✓ DynamoDBが利用できるユースケースが問われる。
整合性モデル	✓ DynamoDBの整合性モデルに基づく影響など、整合性モデルに関わる問題が出題される。
DynamoDBのインデックス	✓ DynamoDBで利用されるキーのタイプや設定方法が問われる。 ✓ DynamoDBの2つのセカンダリーアインデックスの用途と違いが問われる。
DynamoDBストリーム	✓ DynamoDBストリームの効果やユースケース、ストリームを利用したアーキテクチャ構成に関する問題が出題される。

# DynamoDBの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

スケーリング	✓ DynamoDBのスケーリングの設定方法やその効果が問われる。
DAX	✓ DynamoDBを利用したスケーリング方法の1つとしてDAXの利用が問われる。
グローバルテーブル	✓ DynamoDBグローバルテーブルの設定方法や利用目的が問われる。
キャパシティモードの設定	✓ DynamoDBの2つのキャパシティモードの違いや目的に関する質問が出題される。

# [Q] DynamoDBの選択

B社ではIoTソリューションを提供しています。この会社ではIoTデバイスから収集するストリーミングデータを利用して、リアルタイムのデータ処理を実行しています。このデータ処理には複雑なデータスキーマの設定などは必要なく、複雑なトランザクション処理も必要としませんが、リアルタイムでの高パフォーマンスな処理が求められています。

このデータベース処理に最適なAWSのデータベースサービスを選択してください。

- 1) Amazon Aurora
- 2) DynamoDB
- 3) Amazon EMR
- 4) RedShift

# NoSQL型データベース

データベースはリレーショナルDBかそうでないDBかの大きく2つの種類がある

これまでのDB

リレーショナル  
DB

ビッグデータ向けDB

NoSQL

# KVS : キーバリュー型

リレーションナルなしにバリュー一行にデータをまとめることで、高速処理を可能にする

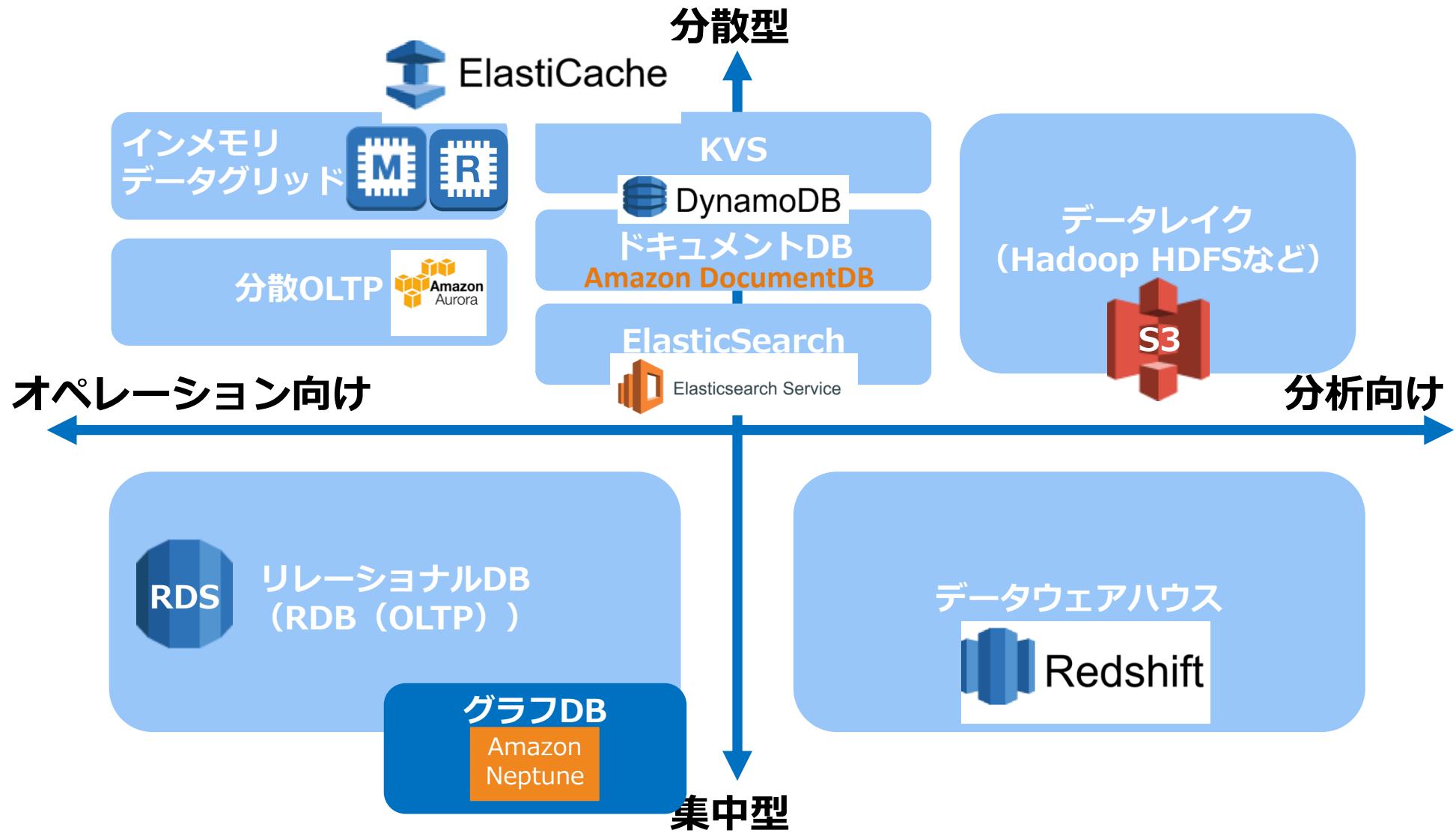
## SQLのテーブル

ID	Data1	Data2	Data3
0001	XXXX	AAAA	BBBB
0002	XXXX	AAAA	BBBB

## キーバルストア型DBのテーブル

Key	Value
0001	XXXX, AAAA, BBBB
0002	XXXX, AAAA, BBBB

# AWSのデータベースサービス



# DynamoDBの出来る事

キー バリュー（ワイドカラム型）でデータを簡易に操作することができる。

## 出来る事

- キーに対するバリュー（値）のCRUD操作
- 簡易なクエリやオーダー
- 例えば、数万人以上が同時アクセスして処理が必要になるアプリケーションのセッションデータ処理などが得意

## 出来ない事／向いていない事

- JOIN／TRANSACTION／COMMIT／ROLLBACKは不可
- 詳細なクエリやオーダー（データの検索や結合処理などには向いていない）
- 大量のデータ読み書きにはコストがかかる

# [Q] DynamoDBの特徴

B社ではAWS上に構築されたアプリケーションを利用してC to Cの売買ソリューションを提供しています。現在、WEBセッションデータ、顧客情報、商品情報を利用して、顧客への最適な商品を recommendationする機能を実装しているところです。このアプリケーションでは様々なデータ処理が必要となりますが、どのデータ処理に DynamoDBを使用するべきか検討しています。

DynamoDBの最適な利用方法は次のうちどれですか？（2つ選択してください）

- 1) 商品画像などの400KBを超えるオブジェクトをS3に保存し、DynamoDBにはメタデータを格納する。
- 2) アイテムごとに個別のローカルセカンダリインデックスを使用して、高速処理を可能にする。
- 3) BLOBデータはDynamoDBに保存する。
- 4) アクセス頻度の高いデータとアクセス頻度の低いデータを別々のテーブルに保存する
- 5) レコメンデーションを高速処理するために顧客管理情報をDynamoDBに格納する。

# DynamoDBのユースケース

ビッグデータ処理向けか大量データ処理が必要なアプリケーション向けに利用する

## ビッグデータ

- IoTデータなどKey Value型のシーケンシャルなデータを収集・蓄積・分析するのに最適
- Amazon EMRのHadoop処理と連携してビッグデータ処理が可能

## アプリケーション

- セッションデータやメタデータなどのアプリケーション上でシンプルでデータを蓄積
- 高パフォーマンスな処理が必要なデータを保存

# DynamoDBのユースケース

大量に発生しうるWEB行動データやログデータの保存には  
DynamoDBを利用する。

## ユーザー行動 データ管理

- ゲームのセッションデータやWEBサイトのユーザー行動データを保存・処理する。
- ユーザー毎の行動履歴管理などに利用する。

## バックエンド データ処理

- モバイルアプリのバックエンド／バッチ処理のロック管理／フラッシュマーケティング／ストレージのインデックス

# DynamoDBのユースケース

DynamoDBとElastiCacheはNoSQL型であるためユースケースが似ている。

比較項目	RDS	DynamonDB	ElastiCache	Redshift
リレーションナルデータベース	○	×	×	○
データベースキャッシュ	△	○	○	×
メタデータ検索	○	○	○	×
セッションなどの状態管理	△	○	△	×
大容量データ分析	△	×	×	○
リアルタイムデータ分析	△	○	○	×
低レイテンシー	△	○	○	×
モバイルバックエンドデータベース	△	○	△	×

参照 <https://qiita.com/leomaro7/items/e48d9941dab5b5f2a718>

# DynamoDBの性能

完全マネージド型のNoSQLデータベースサービスであり、テーブルサイズは無制限だが、1つのデータは400KBに制限

## 【パフォーマンス】

- ハイスケーラブルで無制限に性能を拡張できる
- 負荷が高くなっても応答速度が低下しない低レイテンシー
- 高可用性（SPOFなしでデータは3箇所のAZに保存）
- マネージド型のためメンテナンスフリー：CloudWatchで運用

## 【データ容量の制限】

- **ストレージの容量制限がない**  
テーブルのサイズには実用的な制限はない。  
テーブルは項目数やバイト数について制限がない
- **データ項目には制限あり**  
項目のサイズ制限は400 KBであり、大きなデータを格納できない。

# DynamoDBの性能

1行台のミリ秒レイテンシーを安定して実現しつつ、DAXを利用すればマイクロセカンド単位でのリクエスト処理が可能

## DynamoDBテーブル

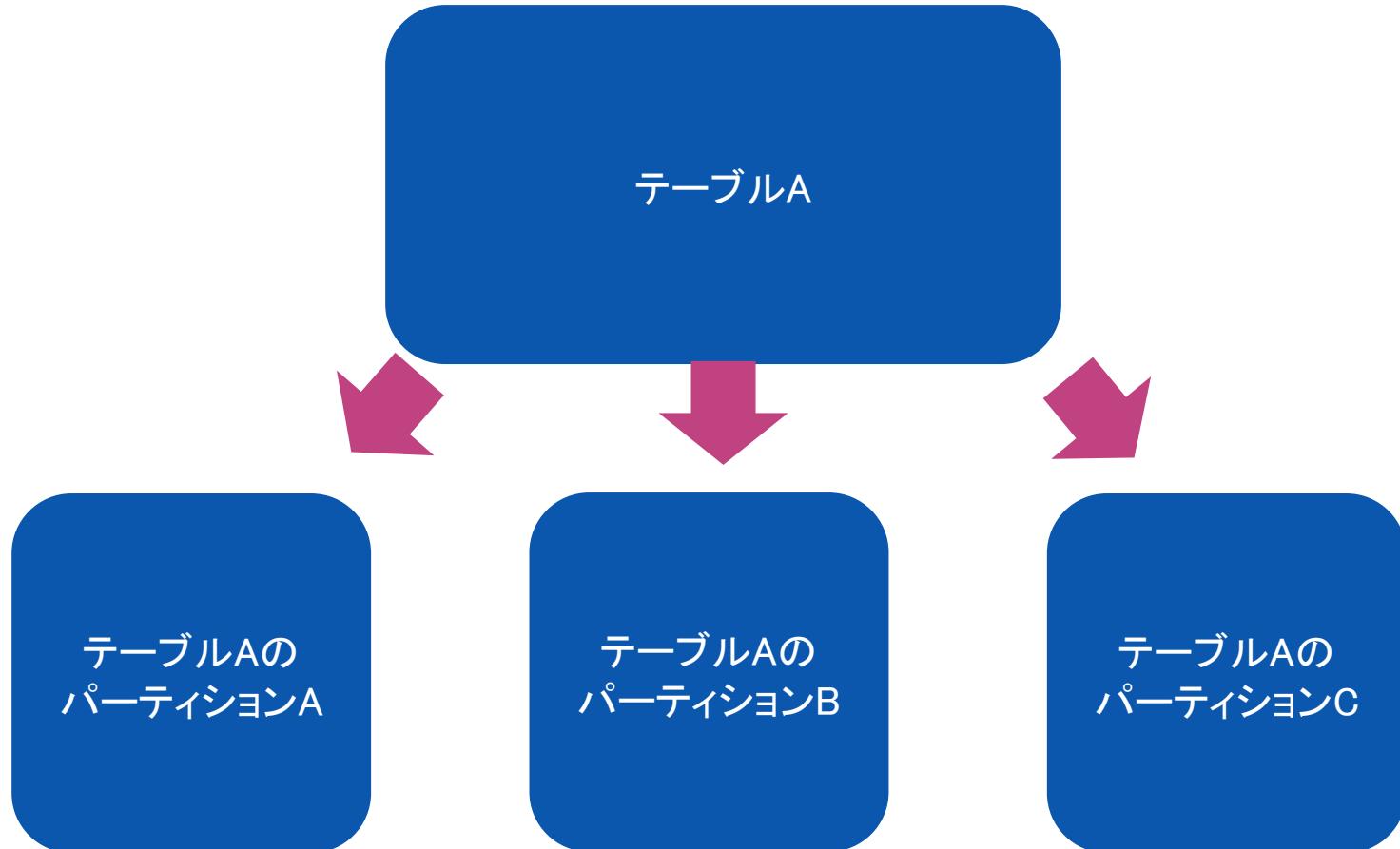
1 行のミリ秒レイテンシーを実現

## DAX

1 秒あたりのリクエスト数が数百万件になる場合でも、ミリセカンドからマイクロセカンドへ向上

# パーティショニング

大量データを高速処理するためにパーティショニングによる分散処理を実施している



## [Q]整合性モデル

ある企業ではDynamoDBを利用して顧客のセッションデータを管理しています。ユーザーがデータベースにアクセスした際に陳腐化したデータが表示されるというクレームが届いています。

あなたは運用担当者として原因を確認して、解決策を選択してください。

- 1) DynamoDBのデータロケーションへのデータ同期が遅れているため、DynamoDBのレプリケーションの設定を有効化する。
- 2) DynamoDBの読み取り処理負荷が高くなっているため、書き込み処理の反映が遅れているため、DynamoDBのDAXを有効化する。
- 3) DynamoDBのデフォルトのデータ読み取り処理では結果整合性モデルを利用しているため、データ読み取り処理時にデータ整合性モデルを利用するクエリを設定する。
- 4) DynamoDBの読み取り処理負荷が高くなっているため、書き込み処理の反映が遅れているため、DynamoDBのクラスターを増強する。

# DynamoDBの整合性モデル

デフォルトで結果整合性モデルであり、一部処理に強い整合性モデルを利用している

## Write

少なくとも2つのAZでの書き込み  
完了が確認された時点で完了

## Read

### □ デフォルト：結果整合性モデル

最新の書き込み結果が即時読み取り  
処理に反映されない可能性がある

### □ オプション：強い整合性モデル

GetItem/Query/Scanでは強い  
整合性のある読み込みオプション  
が指定可能

# テーブル設計

DynamoDBはテーブル単位から利用が開始され、テーブル→項目→属性と設計する

## テーブル

DynamoDBはテーブルはデータのコレクションのこと。他のDBと同様にテーブル単位にデータを保存する

## 項目（アイテム）

各テーブルの中に項目を作ってデータを作成する。項目間で一意に識別可能な属性グループとなる。Personalという項目を作成すれば、名前やIDなどが属性として付属する

## 属性

各項目は 1 つ以上の属性で構成される。属性はそれ以上分割する必要がない最小のデータ単位。例えばPersonal項目には、姓名といった名前の属性を設定する

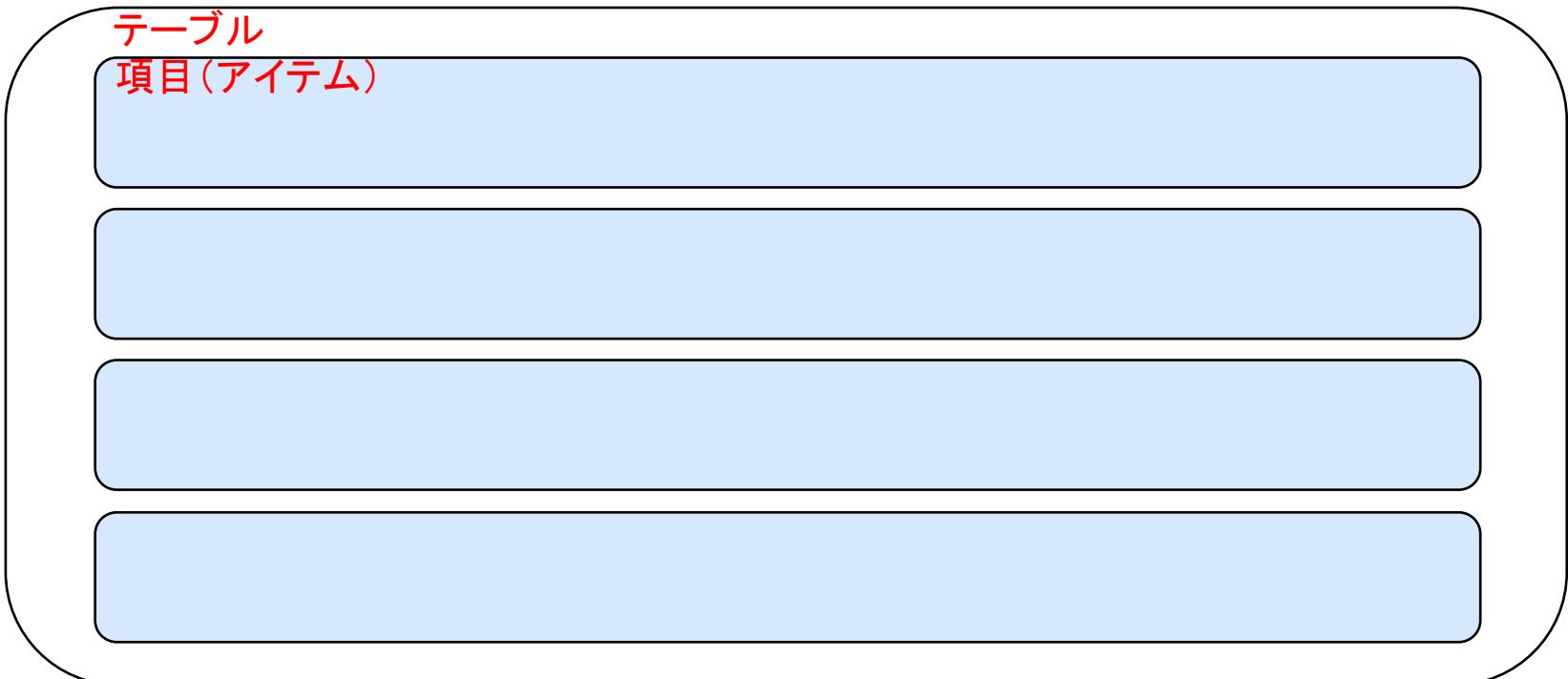
# テーブル設計

テーブルと項目と属性の関係性を入れ子状にしてテーブルを設計する



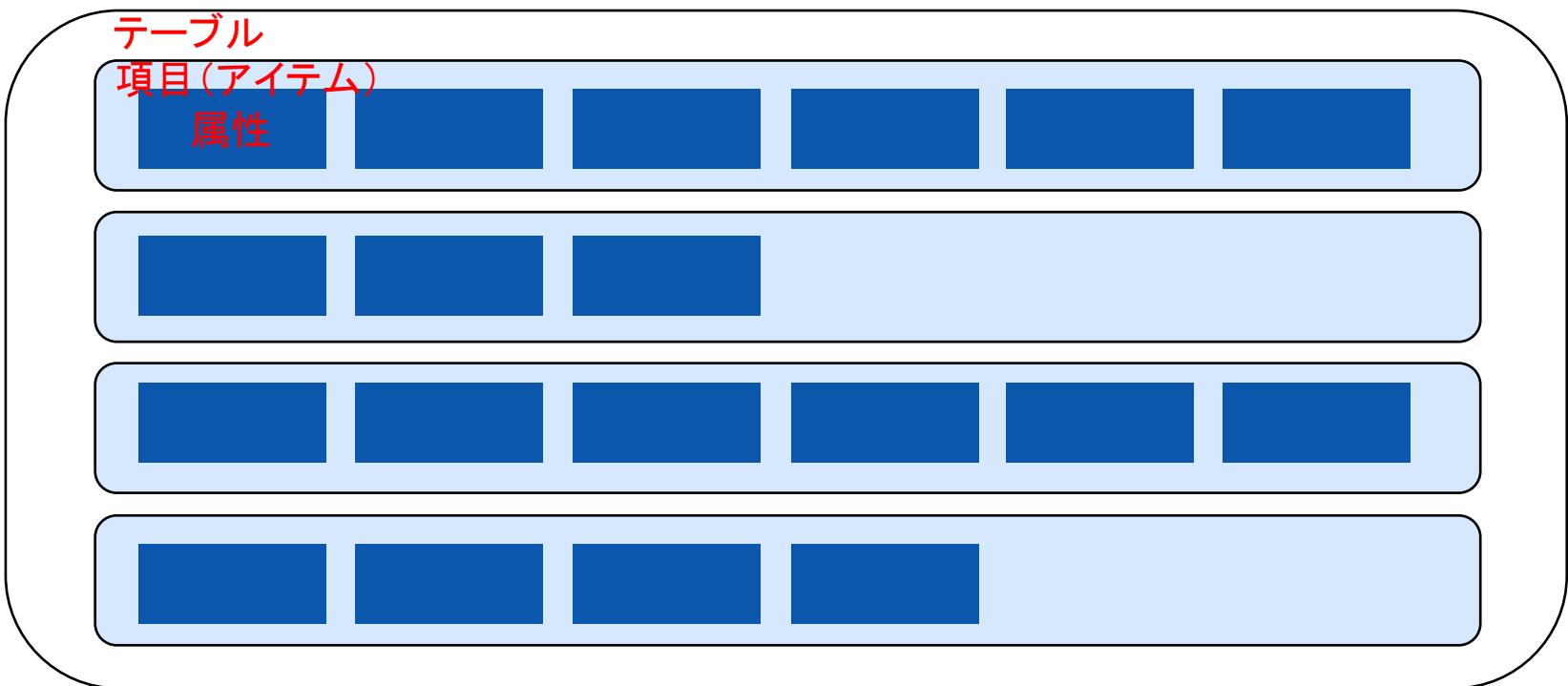
# テーブル設計

テーブルと項目と属性の関係性を入れ子状にしてテーブルを設計する



# テーブル設計

テーブルと項目と属性の関係性を入れ子状にしてテーブルを設計する



属性はVALUE型やJSON型など不ぞろいであっても構わない

# [新Q]キャパシティモードの設定

ある企業は、AWSを利用して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のデータレイヤーにはAmazon DynamoDBテーブルを使用する予定です。このアプリケーションによるデータ処理は午前中はほとんど発生しません。しかしながら、午後以降に利用が増え始めて、夕方ごろは予測不能な読み込みと書き込みのトラフィックが発生します。その際は短時間で、トラフィックのスパイクが発生する可能性があります。

コスト最適に要件を達成するために、ソリューションアーキテクトは何を実施すべきでしょうか。

- 1) オンデマンドモードのDynamoDBテーブルを作成する。さらに、このテーブルをグローバルテーブルとして設定する。
- 2) オンデマンドモードのDynamoDBテーブルを作成する。
- 3) プロビジョンドスループットモードでDynamoDBテーブルを作成する。さらに、このテーブルにオートスケーリングを有効化する。
- 4) プロビジョンドスループットモードでDynamoDBテーブルを作成する。さらに、このテーブルをグローバルテーブルとして設定する。

# キャパシティモードの設定

利用するキャパシティが予測できるか否かでモードを選択

## オンデマンドモード

- 利用するキャパシティが予測できないときに選択するモード
- トラフィック量の予測が困難な場合にリクエストの実績数に応じて課金
- オンデマンドでRead／Write処理に自動スケーリングを実施

## プロビジョニング モード

- 利用するキャパシティが事前予測できるときに選択するモード
- 事前に予測した書き込みキャパシティユニット（WCU）と読み込みキャパシティユニット（RCU）を設定する。
- 設定したキャパシティに基づいて課金
- UpdateTable オペレーションを使用して、必要な回数だけ ReadCapacityUnits または WriteCapacityUnits を増やすことができる。
- キャパシティ容量に近づくとHTTP 400コード（不正なリクエスト）とProvisionedThroughputExceededExceptionが発せられる。

24時間ごとに1回、読み込み/書き込みキャパシティモードを切り替えることができる。

# DynamoDBの料金

キャパシティ設定の方式と利用する機能に応じて課金される。

## オンデマンド

- ストレージ容量（GB単位）
- 書き込み単位
- 読み込み単位

## プロビジョンド

- ストレージ容量（GB単位）
- 読み込みキャパシティーユニット (RCU)
- 書き込みキャパシティーユニット (WCU)

## その他

- グローバルテーブル：レプリケート書き込みキャパシティーユニット (rWCU)
- DynamoDB Accelerator (DAX)：ノード時間単位
- DynamoDB ストリーム：ストリーム読み込みリクエスト単位

# インデックス

DynamoDBは暗黙的に設定するKVSにおけるKeyに値するものと、明示的に設定するキーがインデックスとして利用できる

## 暗黙的なキー

データを一意に特定するために暗黙的にキー（ハッシュキーやレンジキー）として宣言して検索に利用するインデックスで、1テーブルに1つ宣言する

## 明示的なキー

ローカル・セカンダリ・インデックス（LSI）はプライマリキーのタイプがハッシュキーやレンジキーの場合に追加で別のレンジキーを増やすように利用できる  
1テーブルに5つ作成可能／テーブル作成時に作成

グローバル・セカンダリ・インデックス（GSI）は別のハッシュキーを設定することができる。全データに対してグローバルに検索を実施する。  
1テーブルに5つ作成可能／テーブル作成後に作成

# プライマリーキー

DynamoDBはハッシュキーとレンジキーという2種類のプライマリーキーを利用する

## ハッシュキー

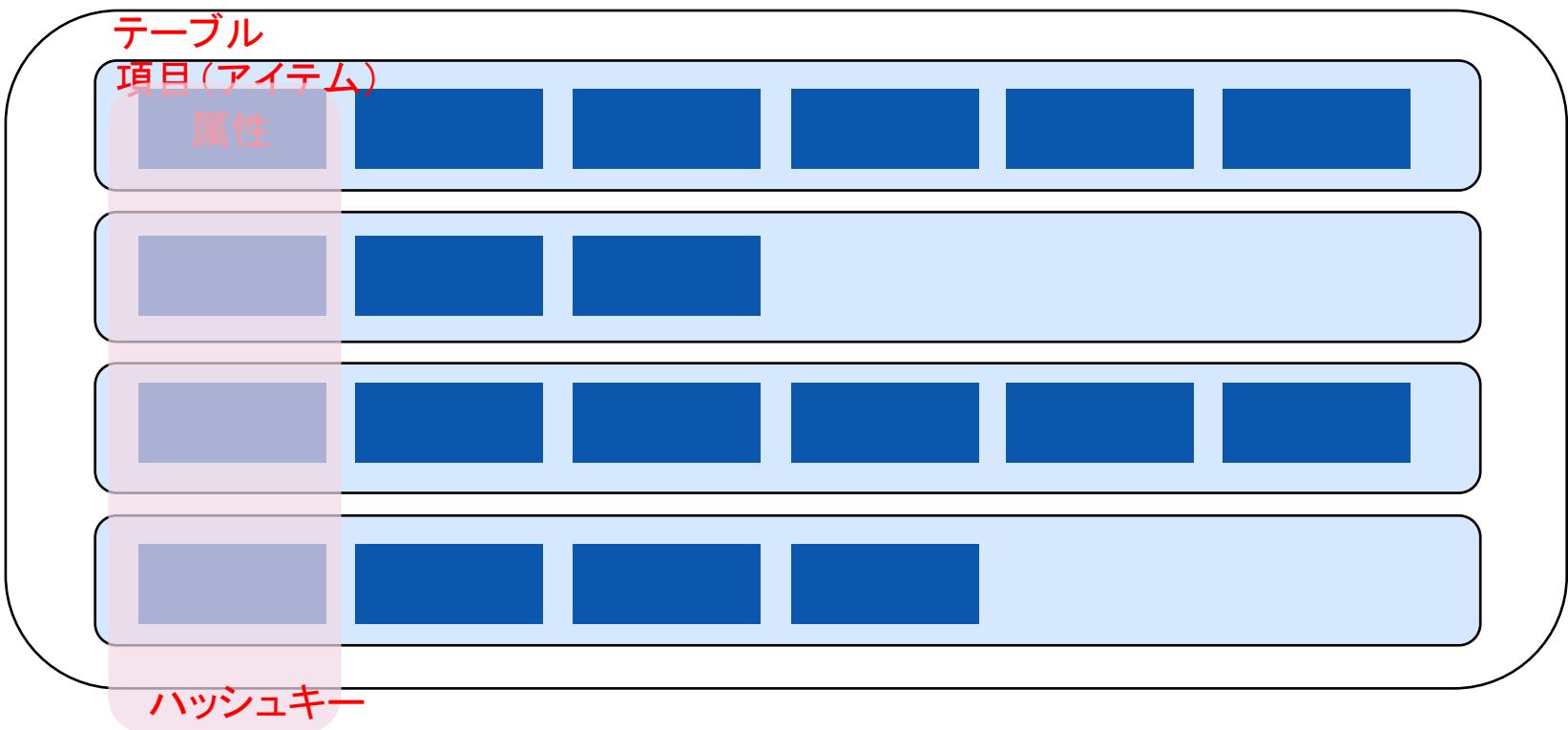
- KVSにおけるキーに相当するデータを一意に特定するためのIDなどのこと
- テーブル作成時に1つの属性を選び、ハッシュキーとして宣言
- ハッシュ関数によってパーティションを決定するためハッシュキーと呼ぶ
- ハッシュキーは単独での重複を許さない

## レンジキー

- ハッシュキーにレンジを加えたものをレンジキーまたは複合キーと呼ぶ
- テーブル作成時に2つの属性を選び、1つをハッシュキーとして、もう一つをレンジキーと呼ばれるキーとして宣言
- 2つの値の組み合わせによって、1つの項目を特定
- 複合キーは、単独であれば重複が許される

# プライマリーキー

テーブルと項目と属性の関係性を入れ子状にしてテーブルを設計する



# プライマリーキー

テーブルと項目と属性の関係性を入れ子状にしてテーブルを設計する



# セカンダリインデックス

ハッシュキーやレンジキーだけでは検索要件が満たせない場合にLSIとGSIを追加する。

## Local Secondary Index (LSI)

- ソートキー以外にインデックスを作成できる検索方式。
- 複合キーテーブルにのみ設定可
- 複合キーによって整理されている項目に対して、パーティションキーを指定した上で、別の規則のインデックスとなりクエリ検索に利用できる。

## Global Secondary Index (GSI)

- GSIはインデックス用に新たにパーティションキーとソートキーを指定する検索方式。
- ハッシュキーテーブル及び複合キーテーブルどちらにでも設定可能
- ハッシュキーの代わりになるため、ハッシュキーをまたいで物理パーティションに囚われない検索が可能

スループットやストレージ容量を追加で必要で書き込みも増大するため、多様すべきではない。

# テーブル操作

テーブル操作としては以下のようなコマンドを利用する

<b>.GetItem</b> ハッシュキーを条件に一定の項目（アイテム）を取得	<b>Query</b> ハッシュキーとレンジキーにマッチする項目を取得（最大 1 MB）
<b>PutItem</b> 1件のアイテムを書き込む	<b>Scan</b> テーブルを全件検索する（最大 1 MB）
<b>Update</b> 1件のアイテムを更新	<b>BatchGetitem</b> 複数のプライマリーキーに対してマッチする項目を取得
<b>Delete</b> 1件のアイテムを削除	

# [Q] DynamoDBストリーム

B社ではAWS上に構築されたアプリケーションを利用してC to Cの売買ソリューションを提供しています。現在、WEBセッションデータ、顧客情報、商品情報を利用して、顧客への最適な商品をレコメンデーションする新機能を開発しています。顧客からの注文情報がテーブルの保存される度に、そのデータに対して前処理を実施して、レコメンデーション用機能へとデータを引き渡す簡易なプログラム処理が必要です。

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) DynamoDB Streamsを有効化して、Lambda関数を実行する。
- 2) DynamoDB DAX を有効化して、 API Gatewayからデータを連携する。
- 3) DynamoDBにAmazon SQSを連携して、DynamoDBからのデータをキューに格納して、 Lambda関数がポーリングして処理を実行する。
- 4) DynamoDBとAmazon EventBridgeを連携して、DynamoDBからのデータ登録に応じてLambda関数を実行する。

# DynamoDBストリーム

DynamoDB テーブルに保存された項目の追加・変更・削除の発生時の履歴をキャプチャできる機能

## データの保存

- 過去24時間以内のデータ変更の履歴を保存し、24時間を経過すると消去される
- データ容量はマネージド型で自動的に管理

## データ保存の順番

- 操作が実施された順番に応じてデータはシリアル化される
- 特定のハッシュキーに基づいた変更は正しい順番で保存されるが、ハッシュキーが異なる場合は受信した順番が前後される可能性がある

# DynamoDBストリームのユースケース

データ更新をトリガーとして処理を実行するアプリケーション機能や、DynamoDBテーブルのレプリケーションに活用できる

## クロスリージョンレプリケーション

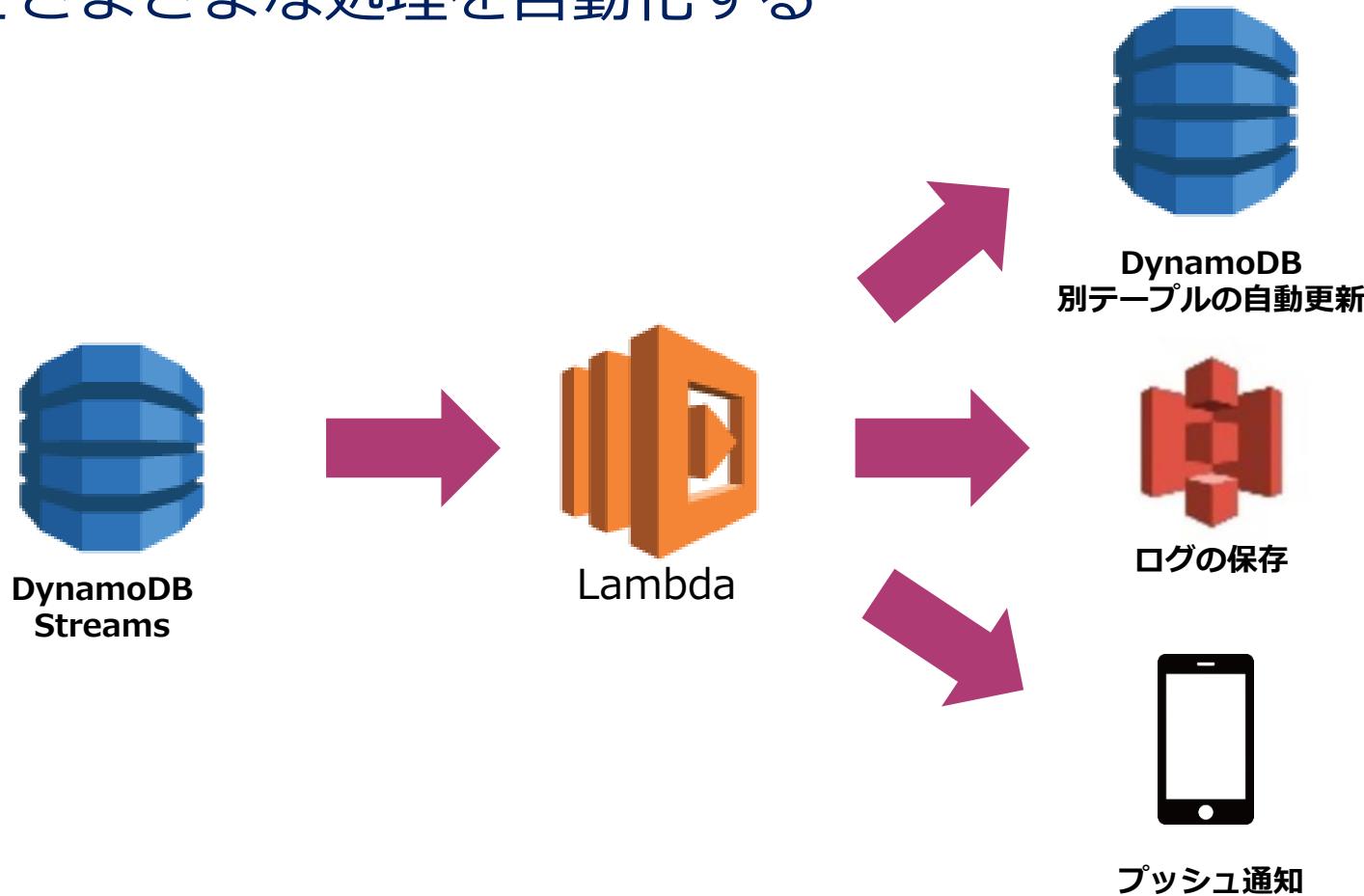
- ストリームによるキャプションをトリガーとしてクロスリージョンレプリケーションを実施することが可能

## データ更新をトリガーとしたアプリケーション機能

- データ更新に応じた通知処理などのアプリケーション処理の実行 など

# DynamoDBストリームのユースケース

DynamoDBの書き込み処理をトリガーにしてLambda関数によってさまざまな処理を自動化する



# [Q]スケーリング

B社ではAWS上に構築されたアプリケーションを利用してC to Cの売買ソリューションを提供しており、セッションデータ処理にはプロビジョニングモードのAmazon DynamoDBテーブルを利用してます。Amazon DynamoDBテーブルの負荷変動が激しく、ある日は頻繁に使用されますが、ほとんど利用されない日もあります。プロビジョニングされたスループット容量は、スロットルが発生しないように、重い負荷を考慮して構成されており、コスト効率が悪いことが問題となっています。

コストを最適化するための最も効率的なソリューションは何でしょうか？

- 1) DynamoDBオートスケーリングポリシーを使用する。
- 2) プロビジョニングされたスループット数を削減する。
- 3) DynamoDBテーブルのキャパシティに基づいてCloudWatchアラームを作成し、アラームに基づいてLambda関数により、 WCUとRCUを自動調整する。
- 4) DynamoDB DAXを使用してデータベースのパフォーマンスを向上させる。

# DynamoDB Auto Scaling

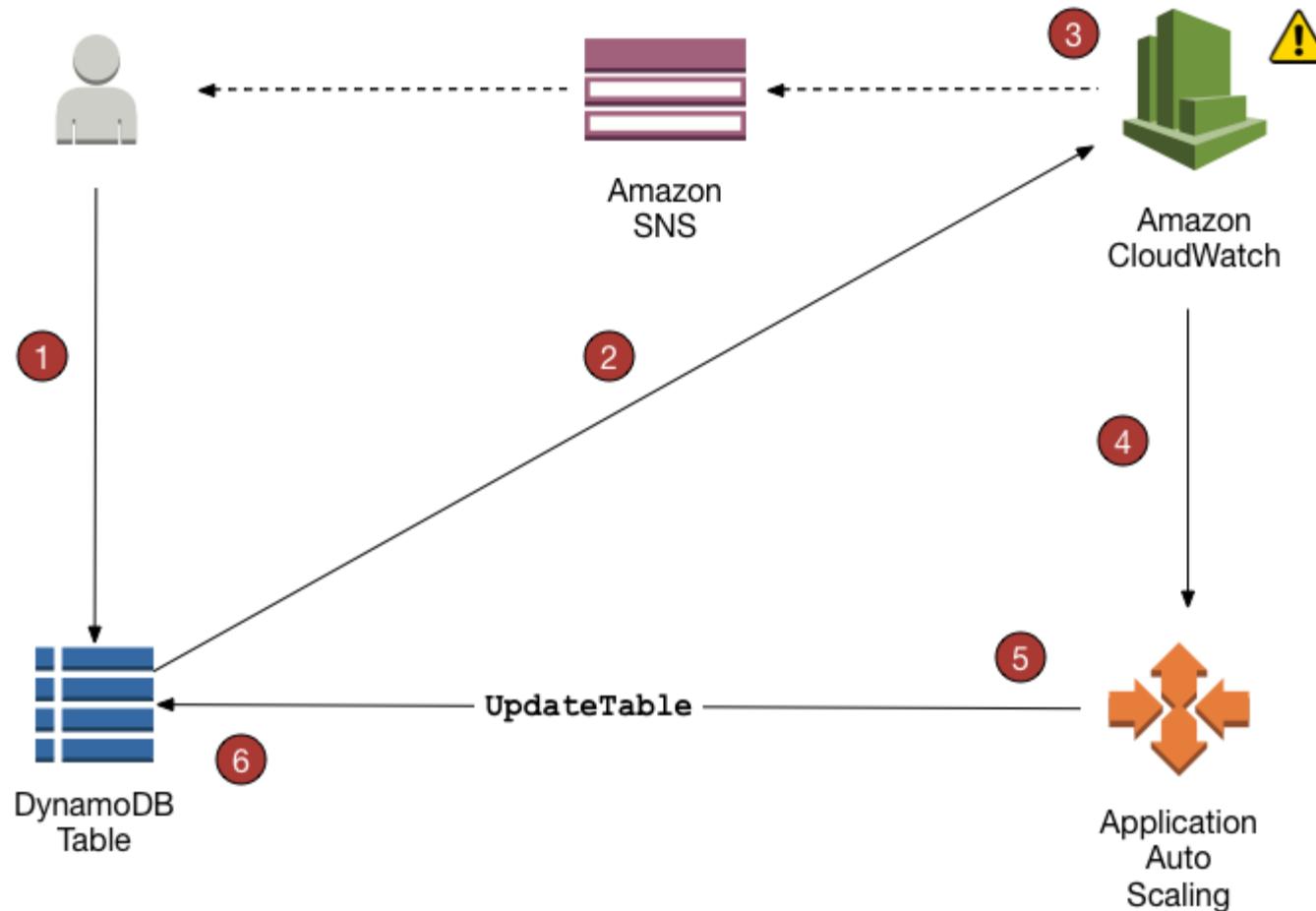
スケーリングポリシーに基づいてテーブルまたはGSIを自動でスケーリングする。

## DynamoDB Auto Scaling

- AWS Application Auto Scaling サービスを使用して Application Auto Scaling ポリシーを設定する。
- CloudWatchのモニタリングに基づいてトラフィックパターンに応じてプロビジョンドスループット性能をユーザーに代わって動的に調節する。
- テーブルまたは グローバルセカンダリインデックスはプロビジョニングされた読み込みおよび書き込みキャパシティーを増やし、急激なトラフィック増加をスロットリングなしに処理できる。
- DynamoDB テーブルを作成すると、Auto Scaling がデフォルトで有効化されている。

# DynamoDBのスケーリング

スケーリングポリシーに基づいてテーブルまたはGSIを自動でスケーリングする。



## [Q] DAX

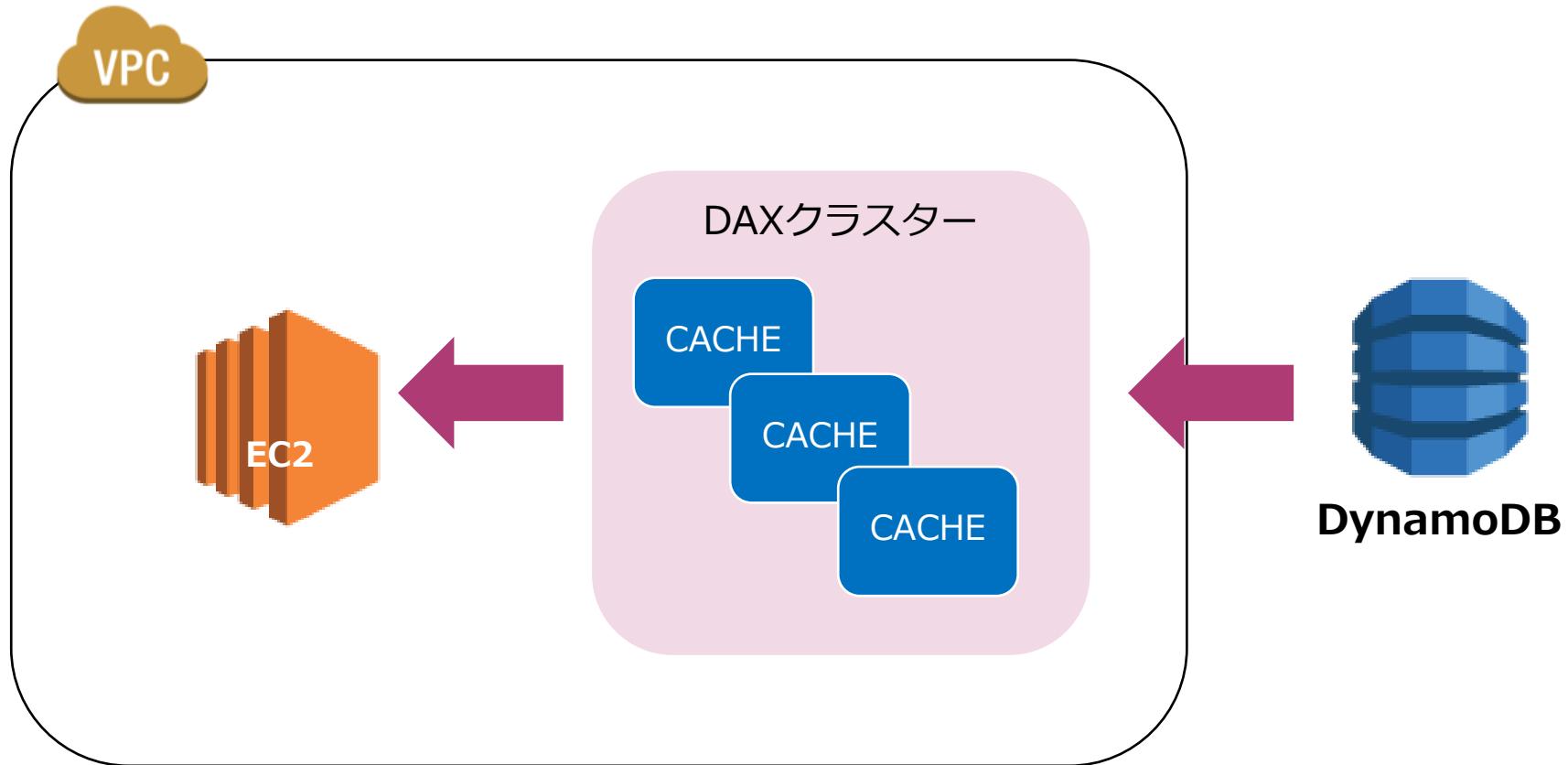
B社ではAWS上に構築されたアプリケーションを利用してC to Cの売買ソリューションを提供しており、セッションデータ処理にはプロビジョニングモードのDynamoDBテーブルを利用しています。最近になって、アプリケーションのリクエスト数が急増しており、あなたは担当のソリューションアーキテクトとして、DynamoDBのRCUを増やしました。それでも、ホットキーでホットパーティションの問題が発生しています。

このホットキーの問題を解消するにはどうすればよいですか？

- 1) DynamoDB グローバルテーブルを利用する。
- 2) DynamoDB Streamsを利用する。
- 3) DynamoDB Accelerator (DAX)を利用する。
- 4) DynamoDBのGSIを利用する。

# DynamoDB Accelerator (DAX)

DAXはDynamoDBにインメモリキャッシュ型の機能を付加する



# DynamoDB Accelerator (DAX)

DynamoDBにおいて高速なインメモリパフォーマンスを可能にする。

- インメモリキャッシュとして 1桁台のミリ秒単位からマイクロ秒単位まで結果整合性のある読み込みワークロードの応答時間を短縮。マルチAZ DAXクラスターは、1秒間に数百万件のリクエストを処理できる。
- DAXはDynamoDB APIと互換性を持つマネージド型サービスであり、運用上そしてアプリケーションの複雑性を減少させて容易に導入可能
- 読み取りの多いワークロードや急激に増大するワークロードに対して、DAXはスループットを強化したり、読み込みキャパシティユニットを必要以上にプロビジョニングしないよう設計することで運用コストを節約できる

## [Q]グローバルテーブル

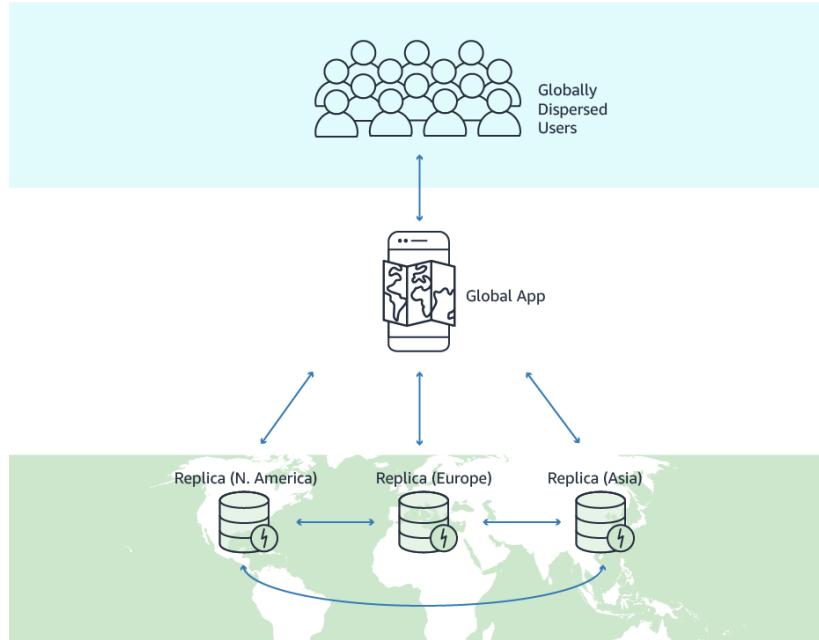
B社はAWS上に構築されたアプリケーションを利用してC to Cの売買ソリューションを提供しており、セッションデータ処理にはプロビジョニングモードのDynamoDBテーブルを利用しています。同社は、アクティブ-アクティブ構成の3つの異なるAWSリージョンにアプリケーションをデプロイしたいと考えています。情報の同期を維持するには、データベースを複製する必要があります。

これらの要件に最適なデータベースソリューションはどれですか？

- 1) グローバルテーブルを備えたAmazon DynamoDB
- 2) ElastiCacheのグローバルレイヤー
- 3) AmazonS3クロスリージョンレプリケーション
- 4) Amazon Auroraのグローバル構成されたマルチマスター構成

# グローバルテーブル

リージョン間で同期されるマルチマスター作成可能



- DynamoDBの性能のまま、世界中で複数のリージョンにエンドポイントを持つことができる
- 読み書きのキャパシティに加えて、クロスリージョンレプリケーションのデータ転送料金に課金される。
- オプションで実施できた強い整合性は使用できない。

# [新Q]バックアップとリカバリ

ある企業は、AWSにホストされたeコマース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では、Amazon DynamoDBを使用して顧客情報を保存しています。データベースに障害が発生した場合は迅速に復旧する必要があります。その際には目標復旧時点（RPO）を15分として、目標復旧時間（RTO）を1時間にすることが求められています。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか？

- 1) DynamoDBグローバルテーブルを設定して、アプリケーションを異なるAWSリージョンにポイントする。
- 2) DynamoDBポイントタイムリカバリを設定する。目的のポイントタイムにリストアする。
- 3) 毎日DynamoDBデータをAmazon S3 Glacierにアーカイブする。データをAmazon S3 GlacierからDynamoDBにインポートする。
- 4) DynamoDBテーブルのスナップショットを15分ごとに取得する。このスナップショットを使って、DynamoDBテーブルをリストアする。



# バックアップとリカバリ

パフォーマンスに影響なく数百TBのバックアップを実行可能

- オンデマンドバックアップ

- 任意のタイミングでテーブルの完全なバックアップを作成する。
  - 長期間の保存とアーカイブを実施

- ポイントインタイムリカバリ

- 連続バックアップを有効化して、バックアップを継続的に実施する。そのため、任意のタイミングでリカバリできる。
  - 過去 35 日間の任意の時点にテーブルを復元することができる。



Lambdaの出題範囲

# Lambdaとは何か？

サーバーを起動せずにプログラミングコードを実行する仕組み。  
簡易なアプリケーション処理を構築することができる。



# Lambdaとは何か？

サーバーを起動せずにプログラミングコードを実行する仕組み。  
簡易なアプリケーション処理を構築することができる。



# Lambdaの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

Lambdaの特徴	✓ Lambdaの特徴や、利用する上での制限設定に関する質問が出題される。
Lambdaの処理タイミング	✓ Lambdaが処理されるタイミングとして、同期処理と非同期処理の設定内容が問われる。
LambdaとVPC	✓ LambdaがVPC内のリソースにアクセスして処理を実行するための設定方法が問われる。
Lambdaレイヤー	✓ Lambdaレイヤーの利用目的が問われる。

# Lambdaの出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

Lambdaの構成	✓ シナリオに基づいて要件が提示されて、それを実現するためのLambdaを利用したアーキテクチャの設計方法が問われる。
API Gatewayとの連携	✓ API Gatewayから、APIコールに基づいてLambda関数を動作させる場合のアーキテクチャ方式が問われる。
Lambdaエッジ	✓ CloudFrontと連携したLambda関数の実行方法が問われます。
RDSとの連携	✓ Lambdaを利用してRDSのデータベース処理を実施する際に、RDSプロキシを利用する構成が問われる。

# [Q] Lambdaの特徴

あなたはソリューションアーキテクトとして、AWS Lambdaを使用して、バッチジョブのワークロードを実装しています。このLambda関数は、Amazon S3からデータを取得して処理した上で、処理結果をDynamoDBに保存します。しかしながら、このLambda関数を実行すると、15分後にLambda関数にエラーが発生していました。

この問題の原因と解決策はどれでしょうか？

- 1) Lambda関数には非同期処理が設定されているため、同期的処理に設定を変更する。
- 2) Lambda関数のメモリが不足しているため、設定するメモリ量を増やす。
- 3) Lambda関数の同時実行数が上限に達しているため、上限緩和申請を実施する。
- 4) Lambda関数の実行時間を超過しているため、Lambda関数のパフォーマンスを向上させる。



# Lambdaの特徴

Lambda関数は様々なコードを利用可能で、AWS側で実行環境が管理されている。

- 実行基盤は全てAWS側で管理されているマネージド型サービス
- AWSサービスと連携させることでLambda関数（ファンクション）と呼ばれる簡単にイベントドリブンなアプリケーションを実装可能
- **Java、Go、PowerShell、Node.js、C#、Python、Ruby** のランタイムをサポート

## 【関数の内容】

- コード

関数のコードと依存関係を作る。スクリプト言語の場合は、組み込みエディタで関数コードを編集が可能。ライブラリを追加するには、またはエディタでサポートされていない言語の場合は、デプロイパッケージをアップロードする。デプロイパッケージのサイズが 50 MB を超える場合はS3からアップロードする。
- ランタイム - 関数を実行する Lambda ランタイムのこと
- ハンドラー - 関数の呼び出し時にランタイムで実行されるメソッド

# Lambdaの課金

Lambdaはリクエスト数とコードの実行期間で算出されて課金される。

- コード実行時間に対して課金されるため、サーバーを保持して処理コードを実行するよりもコスト効率が非常に高い。
- リクエストの数とコードの実行時間に基づいて課金
- 実行時間はコードの実行が開始された瞬間から処理が返されるか、中止されるまでの時間で計算される。値は100 ミリ秒単位で切り上げられる。
- 1 か月ごとに 100 万件の無料リクエスト、および 40 万 GB-秒のコンピューティング時間が無料枠になっている。

# Lambdaの制限

Lambda関数は効率的な処理を可能にするために、データ量や実行時間や同時実行数に制限がある。

- 関数のタイムアウト時間は**デフォルト値は3秒で、許容されている最大値は900秒（15分）**。タイムアウトに達すると、関数が停止される。
- 関数の最大同時実行数はデフォルトは100を最大は1000（申請によって数十万まで引き上げ可能）
- 関数の実行時に使用できるメモリの量。メモリの量を 128 MB ~ 10,240MB の範囲
- /tmp ディレクトリのストレージの保存可能容量は 512 MB~10,240 MB
- Lambdaレイヤーを最大5つまで設定可能

[https://docs.aws.amazon.com/ja\\_jp/lambda/latest/dg/configuration-console.html](https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/configuration-console.html)  
[https://docs.aws.amazon.com/ja\\_jp/lambda/latest/dg/gettingstarted-limits.html](https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/gettingstarted-limits.html)



# Lambdaの仕組み

利用方法もシンプルでWEBアプリやモバイルアプリから簡単に利用可能

Lambdaファンクションを用意する  
(コーディング)

Lambdaを呼び出す

# Lambdaの実装：ブループリント

Lambdaファンクションをコーディングする際にサンプルコード集を利用することが可能

Lambdaを利用する  
ユースケース  
を設計

ブループリントに  
てサンプルコード  
を探す

サンプルコードを  
修正してファンク  
ションを作成する

# [新Q] Lambdaの処理タイミング

ある企業は、サーバレス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Amazon API Gateway APIによって呼び出されるAWS Lambda関数で構成されています。このLambda関数は顧客データを取得して、Amazon Aurora MySQLデータベースに保存します。Amazon Aurora MySQLデータベースのアップグレード時は、アップグレード処理が完了するまではLambda関数がデータベースと接続できなくなり、その間に発生したデータは記録されません。あなたはソリューションアーキテクトとして、その間のデータを保存する仕組みを検討しています。

この要件を満たすために、ソリューションアーキテクトは何を実施するべきでしょうか。

- 1) Lambda関数が顧客データをLambdaのローカルストレージに保持し、このローカルストレージをスキャンして、顧客データをAuroraデータベースに保存する。
- 2) Lambda関数が取得した顧客データをAmazon SQSキューに保存する。別のLambda関数がこのキューにポーリングして、顧客データをAuroraデータベースに保存する。
- 3) Lambda関数の実行時間を上限まで増やす。その上で、このLambda関数でデータベースへの接続に失敗した場合の再実行処理を実装する。
- 4) Amazon RDSプロキシをプロビジョニングし、このRDSプロキシに接続するようにLambda関数を構成する。

# Lambdaの処理タイミング

他のAWSサービスやSDKを利用したアプリケーションからの呼び出して実行することが可能

## 非同期呼び出し

- 関数を非同期的に呼び出してイベントを処理する。
- 関数を非同期的に呼び出す場合は、関数コードからのレスポンスを待機しない。

## 同期呼び出し

- 関数を同期的に呼び出すと、Lambdaが関数を実行し、レスポンスを待つ。
- 実行完了時に、実行された関数のバージョンなどの追加データとともに、Lambda関数内でセットしたレスポンスが返ってくる

# スケジュール機能

特定時刻をトリガーにしてLambdaファンクションを実行する

特定時刻に毎回ファンクション  
を実行したい処理

Lambdaが定期的に実行

# [Q] LambdaとVPC

ソリューションアーキテクトは、AWS Lambda関数を使用するコードを作成しています。Lambda関数は実行されるとElastiCacheクラスターにストリーミングデータを格納します。ElastiCacheは同じアカウントのVPC内に設置されているため、Lambda関数にはVPC内のリソースにアクセスする設定が必要です。

Lambda関数には必要なVPC固有の情報はどれでしょうか？（2つ選択してください）

- 1) VPCサブネットID
- 2) VPCセキュリティグループID
- 3) VPCのARN
- 4) VPC論理ID
- 5) VPCルートテーブルID

# VPCアクセス

インターネットを経由せずにVPC内のAWSリソースへとアクセス可能になる

## VPC内のリソースへのアクセス

- VPC内リソースにインターネットを経由せずにアクセスが可能
- VPCを指定する際にサブネットIDとVPCセキュリティグループIDを指定して、ENIを作成する。ENI経由で接続
- ENIには指定したサブネットのIPがDHCPで動的に割り当てられる

## アクセス設定

- ファンクションに割り当てるIAM Roleに"AWSLambdaVPCAccessExecutionRole"というポリシーをアタッチしておくこと

# [Q] Lambdaレイヤー

あなたはサーバレス構成を利用したアプリケーションでコスト最適化を目指しています。複数のLambda関数を利用したアプリケーションを運用していますが、その処理の部分的に重複していることがわかりました。このLambda関数処理を効率化して、パフォーマンスを向上させる必要があります。

このLambda関数処理を改善するための方法を選択してください。

- 1) Lambdaエッジを利用して、エッジロケーションに処理を分割することで、パフォーマンスを向上させる。
- 2) Lambda Layerを利用して、共通機能をレイヤーに集約して、Lambda関数単体の処理を分散させてパフォーマンスを向上させる。
- 3) Invocationを利用して、共通機能をレイヤーに集約して、Lambda関数単体の処理を分散させてパフォーマンスを向上させる。
- 4) API Gatewayキャッシュを利用して、キャッシュに処理を蓄積して、パフォーマンスを向上させる。

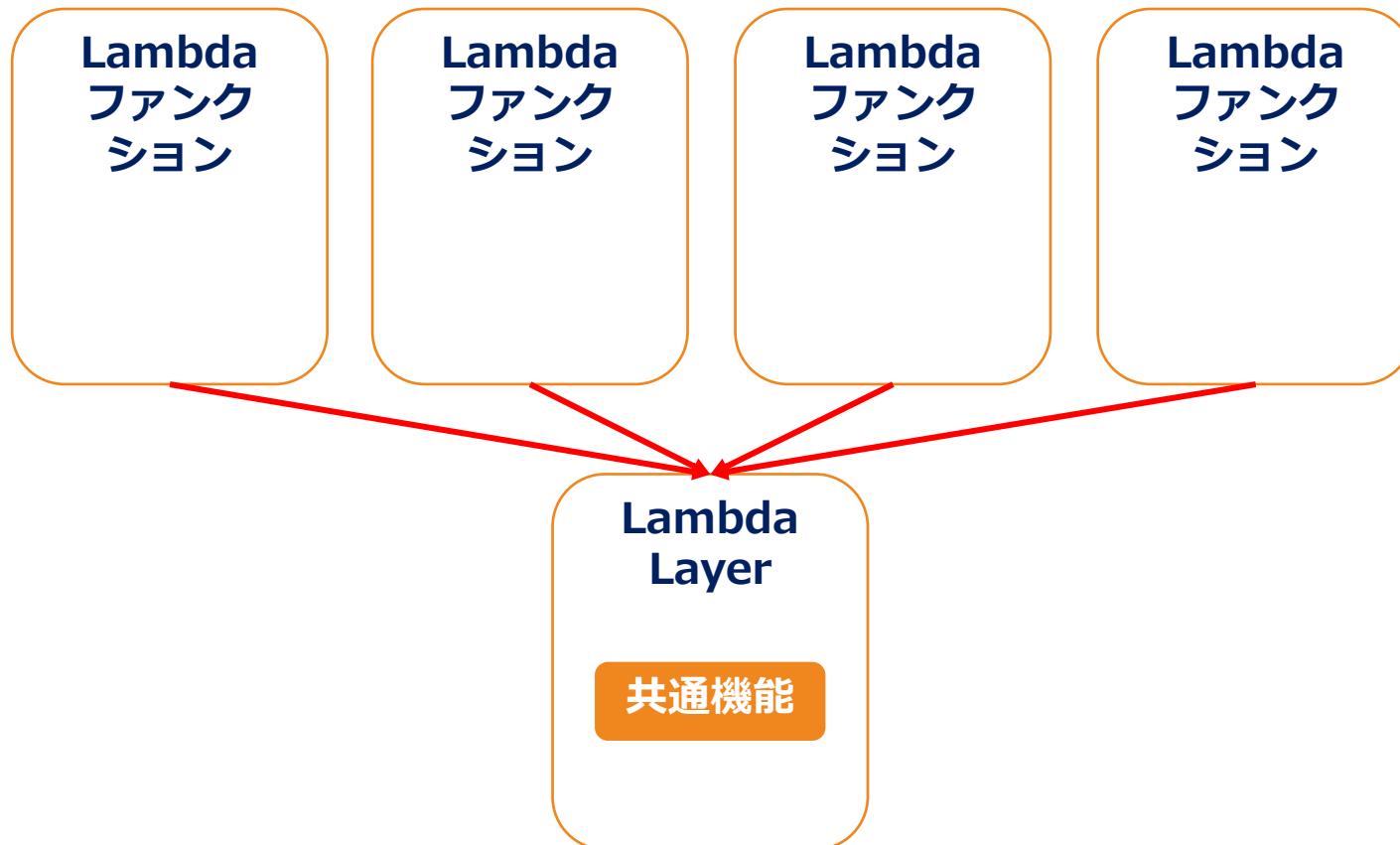
# Lambdaレイヤー

Lambdaファンクション間で共通するコンポーネントをLambda Layerとして定義し参照できる（5つまで）



# Lambdaレイヤー

Lambdaファンクション間で共通するコンポーネントをLambda Layerとして定義し参照できる（5つまで）



# [新Q] Lambdaのスケーリング

ある企業がAWSにホストされた多層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は、REST APIサービス経由で相互通信するアプリケーション層とMySQLデータベースを利用したデータベース層で構成されています。トランザクションが多発して高負荷になると、それらのトランザクションは破棄されないように処理を継続する必要があります。

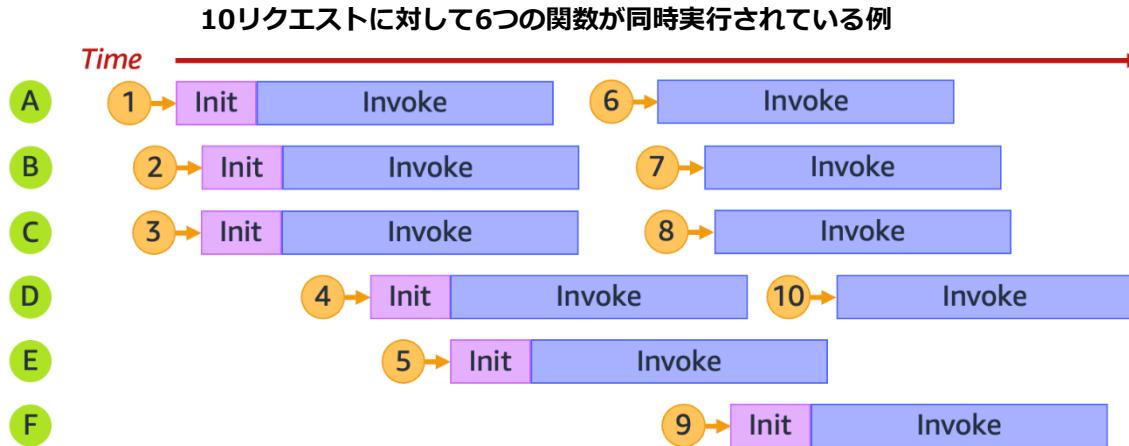
この要件を満たすために、どのソリューションを利用すればよいでしょうか。

- 1) Amazon API GatewayとAWS Lambda関数を統合して、トランザクションを処理する。  
Amazon SQSをアプリケーションサービス間の通信レイヤーに使用して、トランザクション処理を負荷分散する。
- 2) Amazon API GatewayとAWS Lambda関数を統合して、複数のLambda関数でトランザクションを処理する。トランザクション負荷に応じて、Lambda関数が自動でスケーリングされるように設定する。
- 3) Amazon SNS FIFOトピックによって、Amazon EC2インスタンスサーバー群のトランザクション処理を負荷分散する。Amazon CloudWatchによってSNSメッセージ数を監視して、Amazon EC2インスタンスにAuto Scalingグループによるスケーリングを設定する。
- 4) Amazon SQSキューによって、Amazon EC2インスタンスサーバー群のトランザクション処理を負荷分散する。Amazon CloudWatchによってSQSキューサイズを監視して、Amazon EC2インスタンスにAuto Scalingグループによるスケーリングを設定する。



# Lambdaのスケーリング

Lambda関数はオートスケーリングとキャパシティー予約を利用可能



- Lambda は、同時実行リクエストごとに、実行環境の個別のインスタンスをプロビジョニングして自動でスケールする（デフォルトで1,000まで）
- 予約同時実行では、関数の同時インスタンスの最大数を予め予約しておいて、保証する。メリットは、他の関数が関数のスケーリングを妨げないことと、関数のスケーリングが制御不能にならないこと。
- プロビジョニング済み同時実行では、リクエストされた数の実行環境を初期化して、関数の呼び出しに即座に応答するために予め起動に向けた準備がされている。



# [Q] Lambdaの構成

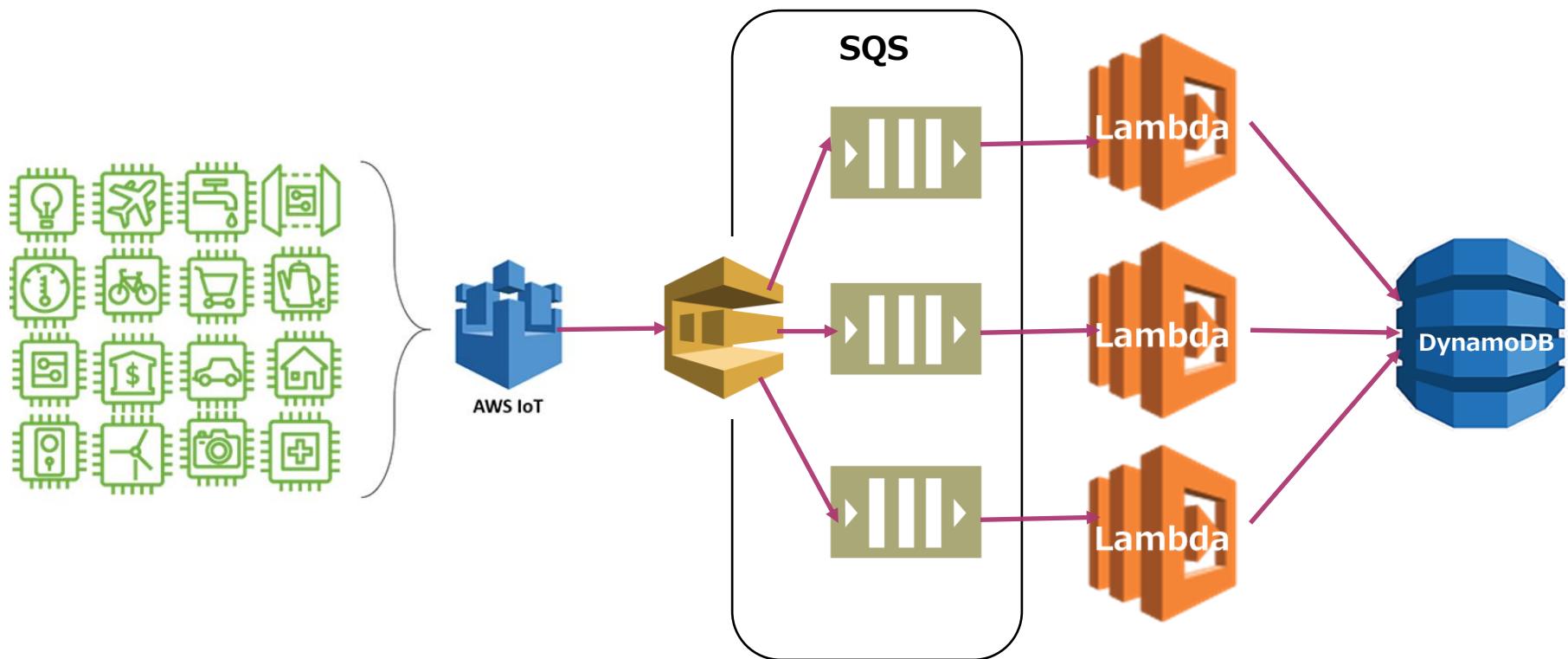
自動車メーカーはAWS上に自動車に設置されたセンサーデータを取得して、データを加工した上で、 DynamoDBに格納するデータ処理アプリケーションのワークフローを実装しています。アプリケーションはデータの保存が成功したことをユーザーに通知を返す必要があります。これらのイベント処理は自動で実行されることが必要です。

この要件を満たすことができるLambda関数の実装方法を選択してください（2つ選択してください。）

- 1) センサーデータをAmazon SQS FIFOキューに取り込み、 Lambda関数によって処理した上で、 DynamoDBテーブルに書き込む。
- 2) センサーデータをKinesis Data Streamsに取り込み、 Lambda関数によって処理した上で、 DynamoDBテーブルに書き込む。
- 3) センサーデータをAmazon SQS 標準キューに取り込み、 Lambda関数によって処理した上で、 DynamoDBテーブルに書き込む。
- 4) DynamoDBストリームを処理するLambda関数を連携して、 Lambda関数からAmazon SNS通知を設定する。
- 5) DynamoDBストリームによってデータ登録時のイベントストリームに基づいて別のLambda関数が通知を実行する。

# Lambdaユースケース

SQSとLambdaを組み合わせてIoTセンサーデータを  
DynamoDBに格納する処理プログラムを作ることが可能



# Lambdaの連携

様々なAWSサービスをトリガーとして起動するなどの連携処理が可能

- Amazon S3
- Amazon Kinesis
- Amazon DynamoDB Streams
- Amazon Cognito(Sync)
- Amazon SNS
- Amazon SQS
- Alexa Skills Kit
- Amazon SWF
- Amazon EventBridge

# [新Q] API Gatewayとの連携

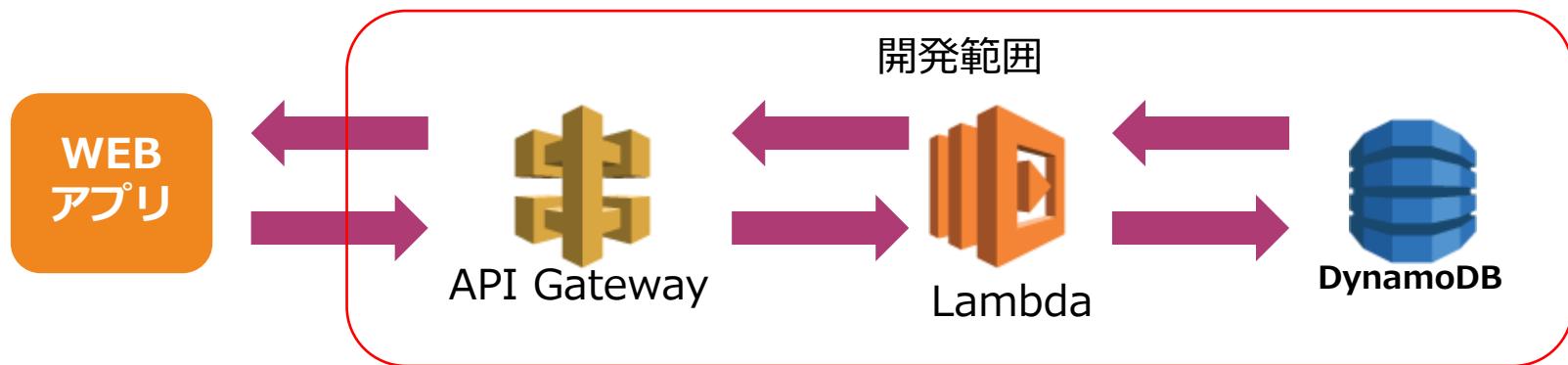
あなたはソリューションアーキテクトとして、予測できないワークロードをサポートするために拡張性がある新しいWebアプリケーションを開発しています。このアプリケーションは、実行回数が限られており、その実行時間も短いため、外部からのHTTPSコールに応じて実行されるシンプルなワークロードを実行します。

このユースケースに最も適したソリューションはどれですか？

- 1) Amazon S3とAmazon API Gatewayを統合して、APIからアクセス可能なアプリケーションを構成する。
- 2) AWS LambdaとAmazon API Gatewayを統合して、APIからアクセス可能なアプリケーションを構成する。
- 3) EC2インスタンスとAmazon API Gatewayを統合して、APIからアクセス可能なアプリケーションを構成する。
- 4) Amazon Kinesis Data StreamsとAmazon API Gatewayを統合して、APIからアクセス可能なアプリケーションを構成する。

# API Gatewayとの連携

API Gatewayと統合することで、Lambda関数をAPIから実行することができる。



# [新Q] Lambdaの権限設定

ある企業は、サーバレス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Amazon EventBridgeルールによって呼び出されるAWS Lambda関数で構成されています。ソリューションアーキテクトは最小権限の原則に従って、AWS Lambda関数を実行するのに必要なアクセス許可を設定する必要があります。

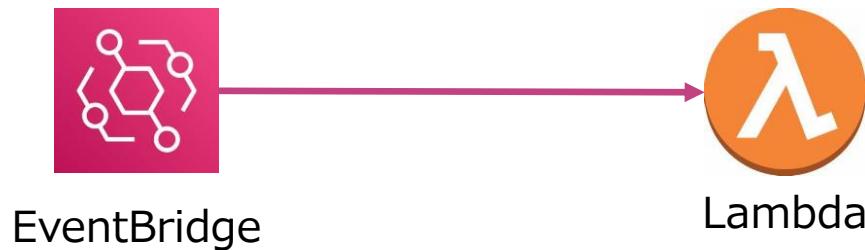
この要件を満たすために、ソリューションアーキテクトはどうするべきでしょうか。

- 1) lambda:InvokeFunctionをアクションに、\*をプリンシパルに設定した実行ルールを付与する。
- 2) lambda:InvokeFunctionをアクションに、events.amazonaws.comをプリンシパルに設定した実行ルールを付与する。
- 3) lambda:LaunchFunctionをアクションに、\*をプリンシパルに設定したリソースベースのポリシーを関数に付与する。
- 4) lambda:InvokeFunctionをアクションに、プリンシパルにlambda.amazonaws.comに設定したリソースベースのポリシーを関数に付与する。



# Lambdaの権限設定

EventBridgeからLambda関数を実行する場合はadd-permissionコマンドにおいて プリンシパルにevents.amazonaws.comを、アクションに‘lambda:InvokeFunctionを設定する。



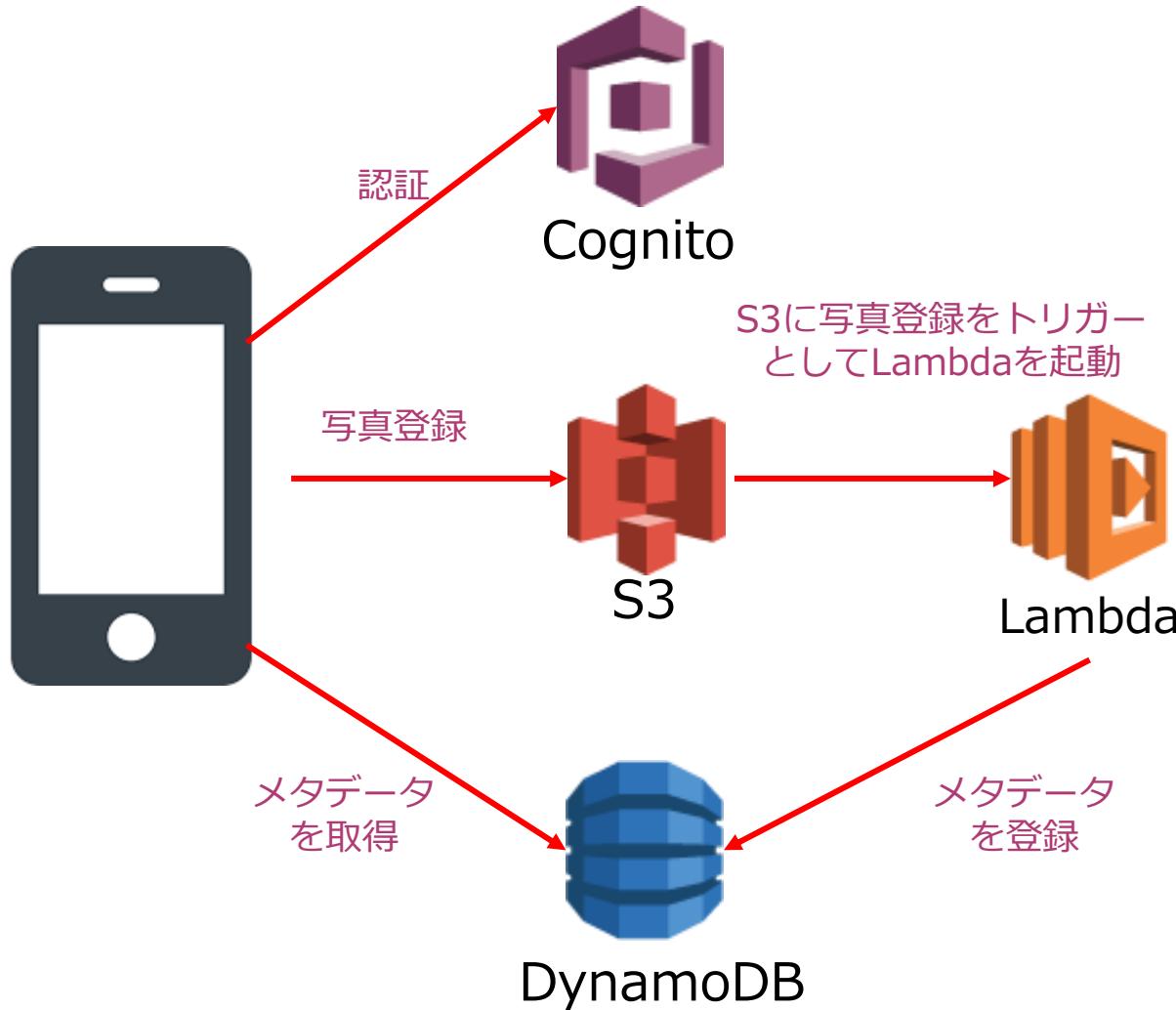
## 【アクセス許可設定の例】

```
aws lambda add-permission  
--function-name LogScheduledEvent  
--statement-id my-scheduled-event  
--action 'lambda:InvokeFunction'  
--principal events.amazonaws.com  
--source-arn arn:aws:events:us-east-  
1:123456789012:rule/my-scheduled-rule
```



# Lambdaモバイルアプリ

モバイルからの写真管理をLambdaを通して実施するなどモバイル連携も容易



# [Q] Lambdaエッジ

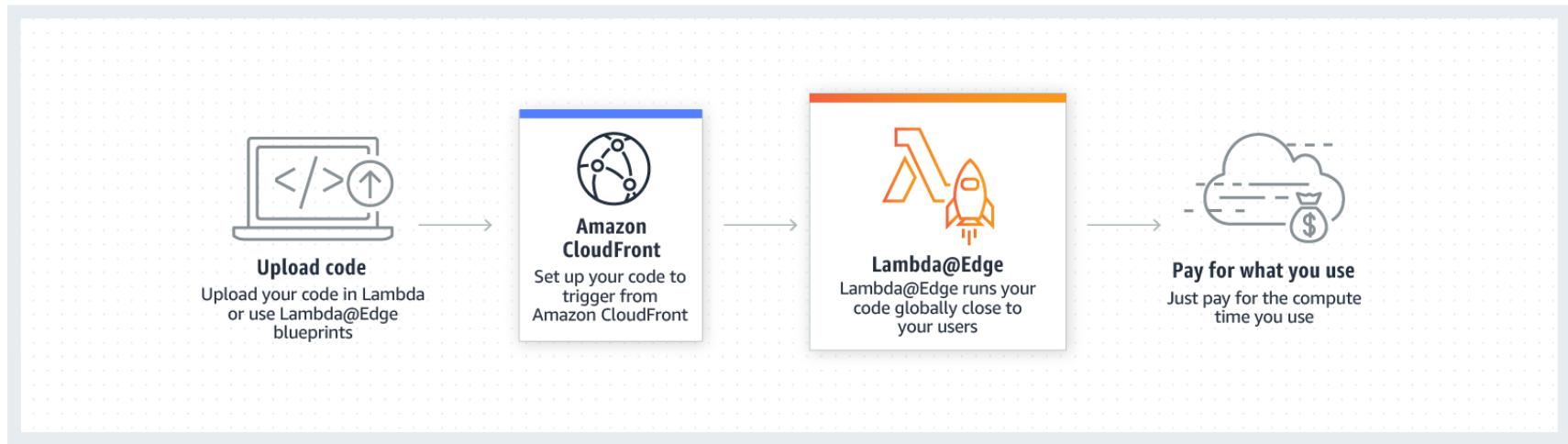
大手ニュースサイトはCloudFrontのWEBディストリビューションを使用して、静的コンテンツを世界中のユーザーに提供しています。URIに対応したHTMLファイルが存在しないため、ブラウザリロード時などのリクエストがエラーになってしまうことがあります。その際に、403／404などのエラーページをindex.htmlにリダイレクトすることでこの問題を回避する設定が必要です。

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) リージョナルエッジキャッシュを利用して、レスポンスをリダイレクトする。
- 2) ローカリゼーションを利用して、レスポンスをリダイレクトする。
- 3) CloudFrontのリダイレクト設定を有効化する。
- 4) Lambda @ Edgeを使用して、CloudFront Webディストリビューションがユーザーに配信するコンテンツをカスタマイズする。

# Lambdaエッジ

CloudFrontの配信コンテンツをLambda関数によってエッジロケーションで処理することが可能

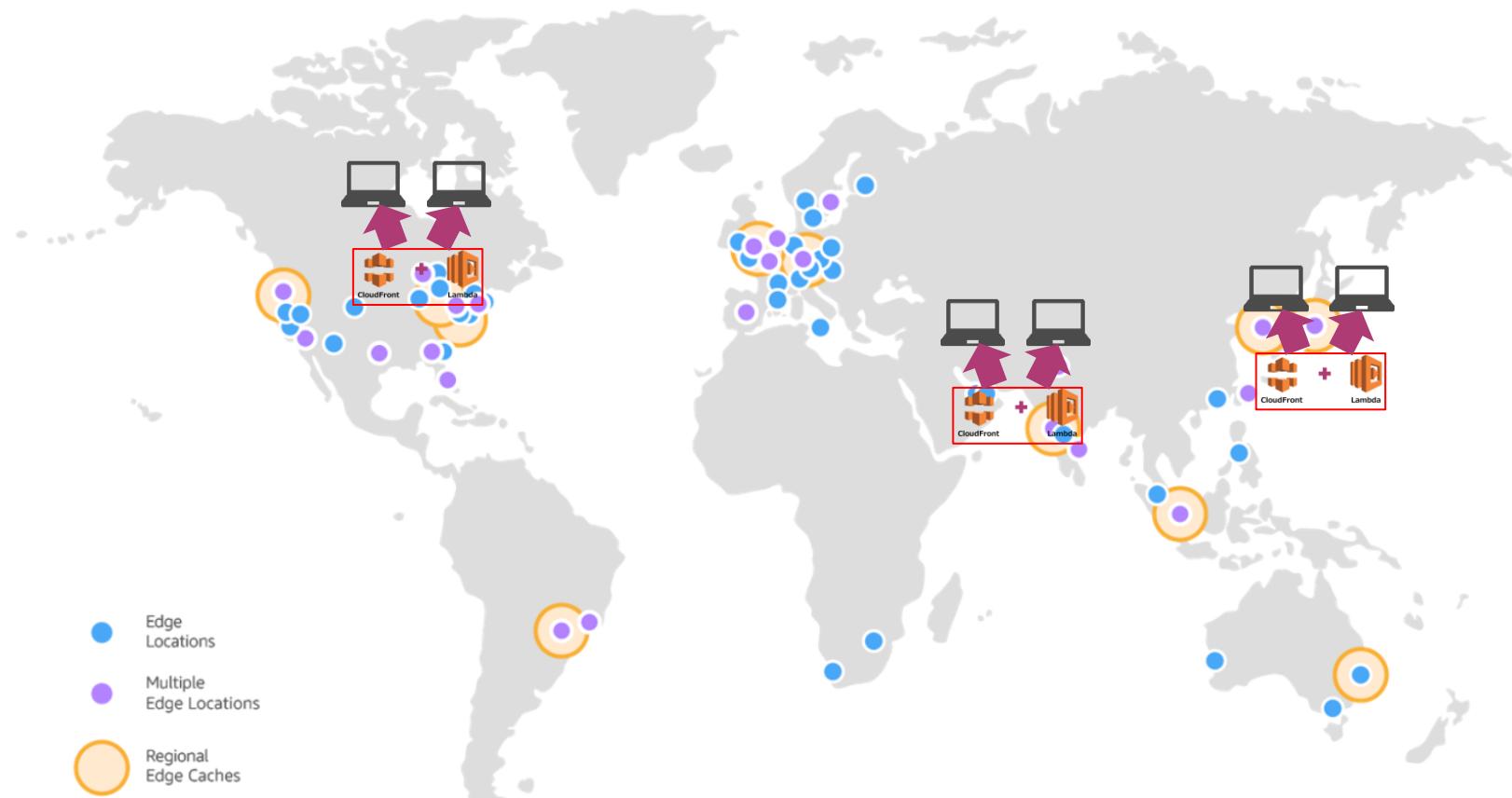


Reference: <https://aws.amazon.com/jp/lambda/edge/>

- グローバルにLambda関数を並行実施したい場合
- ユーザーに近いロケーションでコンテンツ配信などのフィルタリング処理などを実施する場合

# Lambdaエッジ

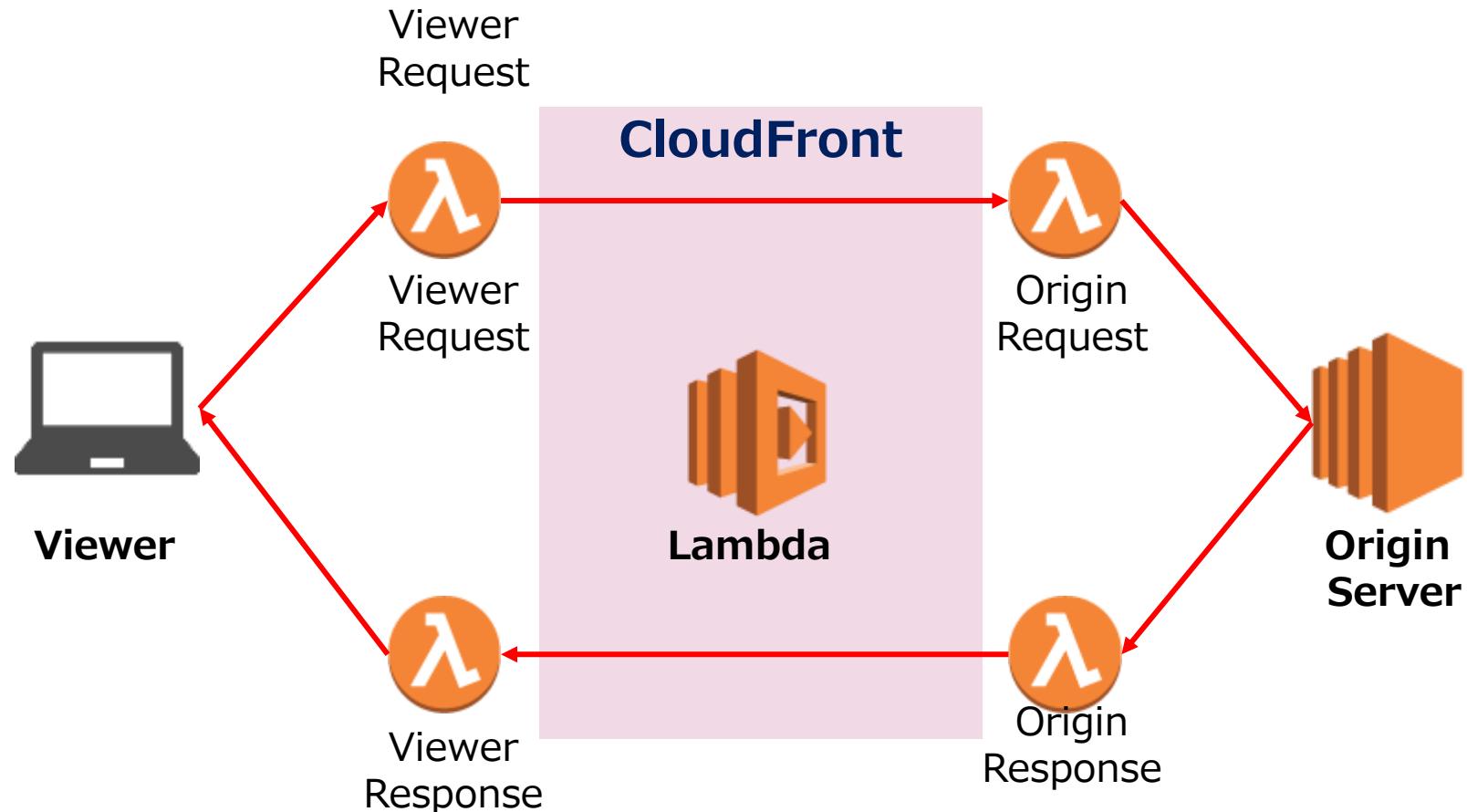
CloudFrontにLambda機能を連携することで、世界中でユーザーに近いロケーションにおいてコードを実行できる



参照 : <https://aws.amazon.com/jp/cloudfront/features/?nc=sn&loc=2>

# Lambdaエッジ

イベントに関連付けられてLambdaファンクションがエッジロケーションで実行されて実行結果を返答する



- CloudFrontのレスポンス/リクエスト処理時にフィルタリングなどの関数処理を実行することができる。

# [新Q] RDSとの連携

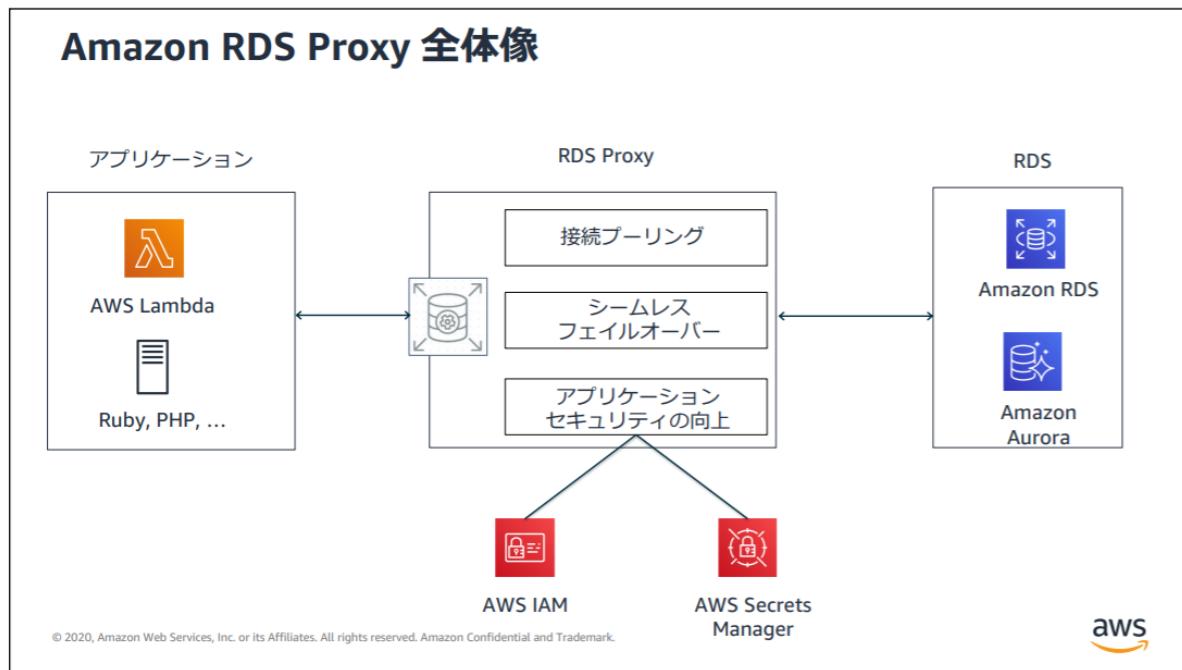
あるベンチャー企業は、Amazon RDS MySQLを利用した顧客管理データベースを構築しています。そこでは、多数の顧客情報やユーザーによる売買記録が保存されており、今後のデータ分析に利用される予定です。あなたはソリューションアーキテクトとして、RDSデータベース上に保存されたデータにアクセスしてデータ処理を実行するLambdaベースのサーバレスアプリケーションを開発しています。その際にはRDSへのコネクション接続を最適にする必要があります。

この要件を満たす最適なソリューションを選択してください。

- 1) Lambda関数をSQSによるキューと連携して、RDSのデータ処理を分散化する。
- 2) Lambda関数からRDSエンドポイントに接続してデータ処理を実施することで、効率的な非同期並列実行を可能にするデータ処理アプリケーションを構築する。
- 3) RDSのスティックィセッションを有効化する。これによって、Lambda関数による効率的な非同期並列実行を可能にするデータ処理アプリケーションを構築する。
- 4) Lambda関数からRDS Proxyに接続してデータ処理を実施することで、Lambda関数による効率的な非同期並列実行を可能にするデータ処理アプリケーションを構築する。

# RDSプロキシ

Lambdaを利用してRDSのデータベースに接続する際は、RDSプロキシをエンドポイントの代わりに利用して、接続をキャッシングすることで効率化できる。



- プロキシを利用して接続をキャッシングして無駄なコネクションを削減
- RDS Proxyへの接続はIAM認証を利用する
- TLS/SSLによる暗号化を実施
- フェールオーバーによる高可用な接続をマネージド型で提供

Reference [https://pages.awscloud.com/rs/112-TZM-766/images/EV\\_amazon-rds-aws-lambda-update\\_Jul28-2020\\_RDS\\_Proxy.pdf](https://pages.awscloud.com/rs/112-TZM-766/images/EV_amazon-rds-aws-lambda-update_Jul28-2020_RDS_Proxy.pdf)

# Route53の出題範囲

# Route53とは何か？

IPアドレスを人が読みやすいURLに変換して、住所として利用できるようにしてくれるDNSサーバーの役割を提供



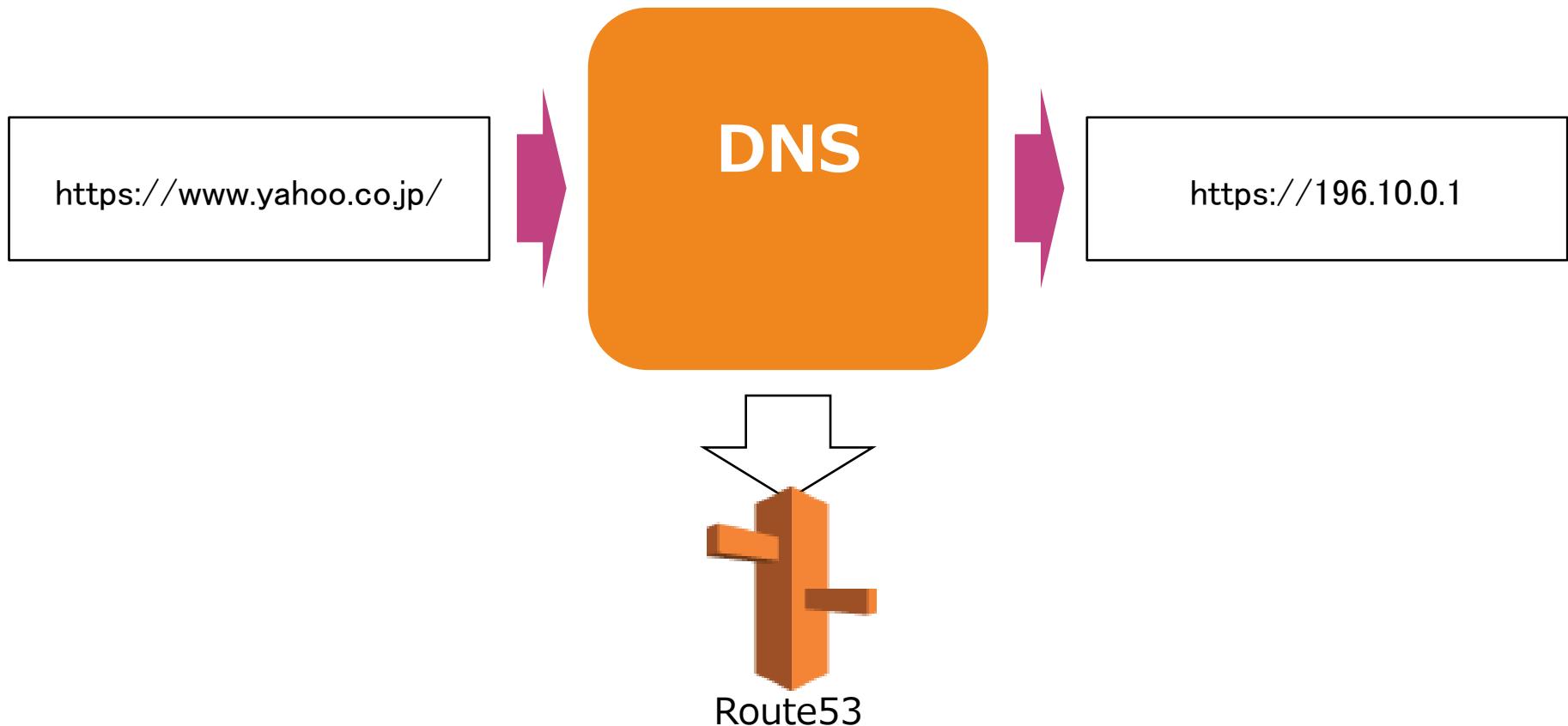
# Route53とは何か？

DNSはインターネットにおける人向けのURLをシステム向けの住所となるIPアドレスに変換するための仕組み



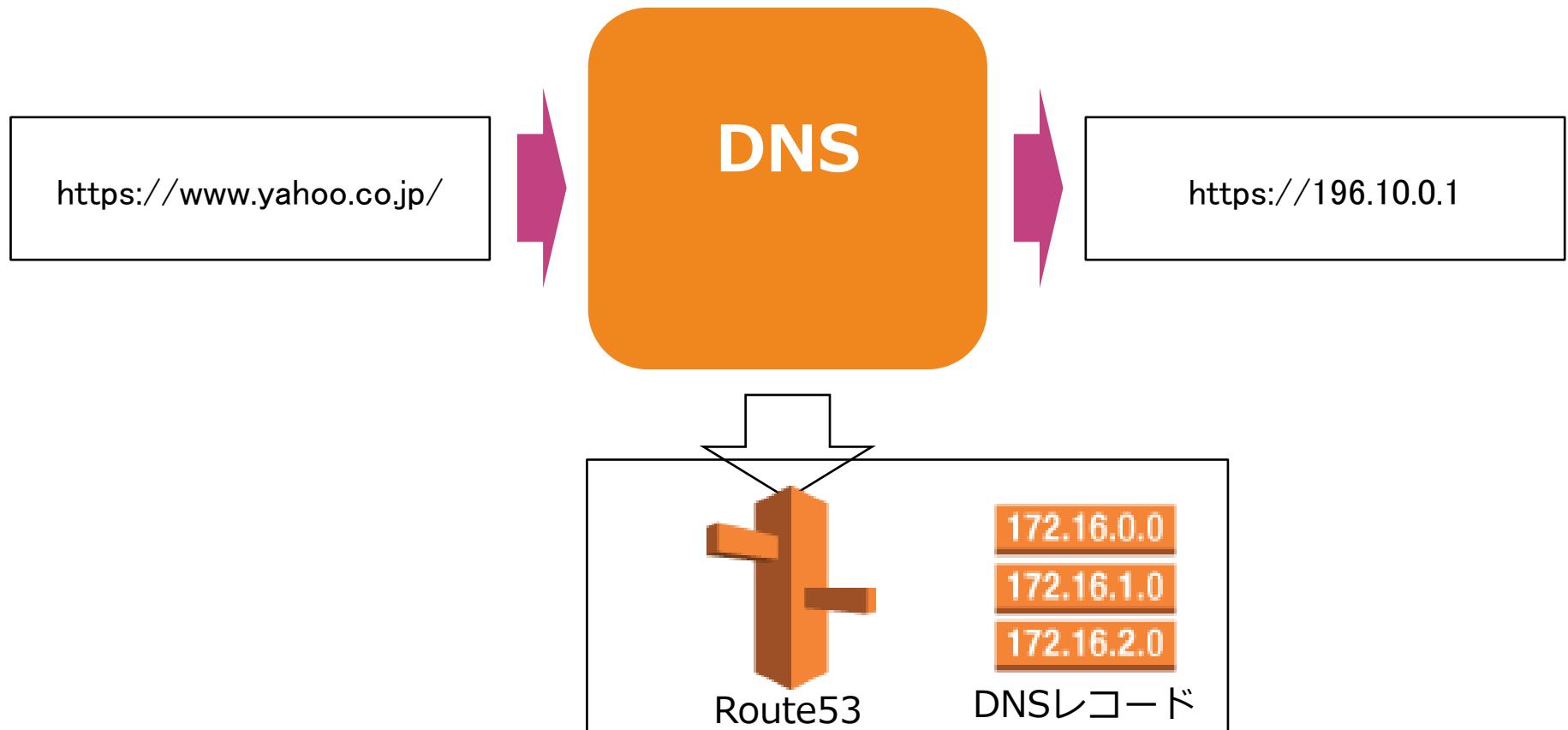
# Route53とは何か？

Route53はAWSが提供する権威DNSサーバーで、ポート53で動作することからRoute53と呼ばれる



# Route53とは何か？

DNSレコードというIPアドレスとURLを紐づけた表を確認してルーティングする



# Route53の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

ホストゾーン	✓ Route53のドメイン設定の際に一番最初に実施するホストゾーンの作成について、プライベートホストゾーンとパブリックホストゾーンの使い分けなどの特徴に関する問題が出題
レコードタイプ	✓ Route53の設定においてレコードタイプを選択する問題が出題される。 ✓ レコードタイプの違いに関する質問が出題される。
ルーティングポリシーの選択	✓ Route53を設定するシナリオが提示されて、適切なルーティングポリシーを選択する問題が出題される。
フェールオーバー構成	✓ Route53を利用してフェールオーバー構成を実現する際の設定方法が問われる。

# Route53の出題範囲

1625問から質問出題範囲を抽出した頻出問題は以下の通り

Route53による 地域制限	✓ Route53を利用して配信先の地域を限定するための設定方法が 問われる。
トラフィックフロー	✓ Route53のルーティングポリシー設定におけるトラフィックフ ローの利用方法が問われる。
TTL	✓ Route53にDNS名前解決におけるTTL設定に関する質問が出題 される。
オンプレミス環境 への適用	✓ Route53を利用してオンプレミス環境への名前解決の適用方法 に関する質問が出題される。

# Route53

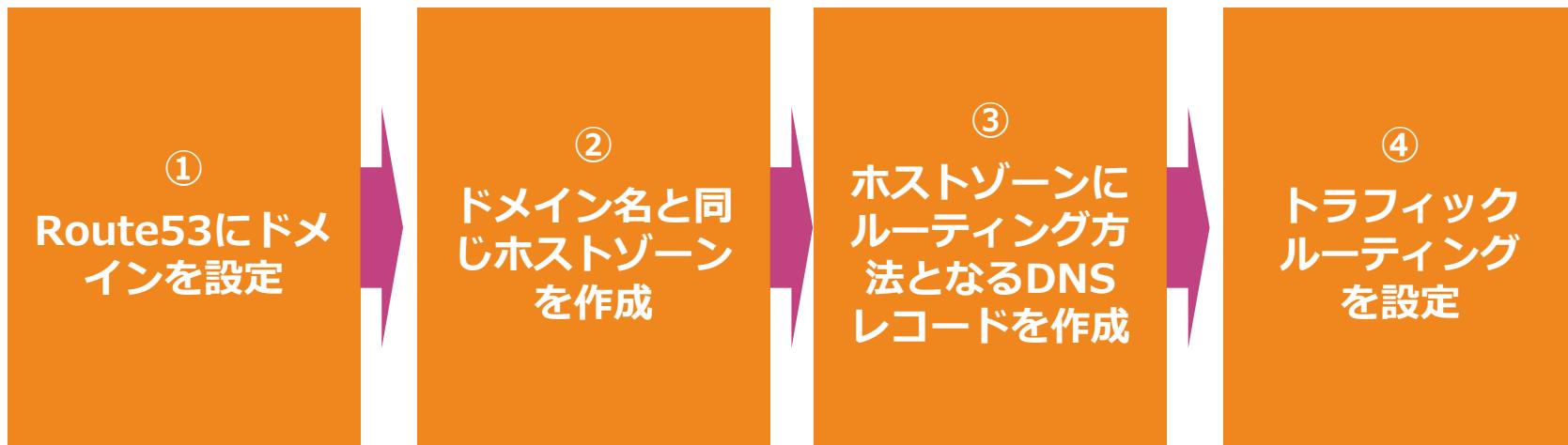
Route53は権威DNSサーバーの機能をマネージド型で簡単に利用できるサービス

- 主要機能はドメイン登録／DNSルーティング／ヘルスチェックの3つ
- ポリシーによるルーティング設定
  - トラフィックルーティング／フェイルオーバー／トラフィックフローに基づく様々な条件のルーティング設定が可能
- AWS側で100%可用性を保証するSLA
- マネージドサービスとして提供しており、ユーザー側で冗長性などを考慮する必要がない
- ドメインを購入・登録・管理するレジストラーとしても機能する。
- 他のレジストラーで購入したドメインを移管することも可能



# Route53の利用方法

Route53の利用を開始してドメインを登録すると自動でホストゾーンを自動生成し、そこにルーティングを設定する。



# [Q]ホストゾーン

ある企業では2つのEC2インスタンスを利用して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、EC2インスタンスをDNSルーティングによる冗長構成とすることで、異常が発生しているインスタンスへのトラフィックを回避できるように設定しようとしています。マルチリージョンにも対応できる構成とするため、Route53を利用したルーティングを利用することにしました。そのためにはパブリックホストゾーンを設定することが必要です。

パブリックホストゾーンの特徴として正しい内容を選択してください。（2つ選択してください。）

- 1) VPCが相互アクセス可能であれば複数リージョンのVPCでも、同じホストゾーンを利用可能である。
- 2) プライベートサブネット内にあるドメインをルーティングすることが可能である。
- 3) インターネット上に公開されたDNSドメインレコードを管理するコンテナである。
- 4) インターネットのDNSドメインに対するトラフィックのルーティング方法を定義する。

# ホストゾーン

ドメイン (example.com) とそのサブドメイン (sub.example.com) のトラフィックのルーティングする方法についての情報を保持するコンテナ

## パブリックホストゾーン

- インターネット上に公開されたDNSドメインレコードを管理するコンテナ
- インターネットのDNSドメインに対するトラフィックのルーティング方法を定義

## プライベートホストゾーン

- VPCに閉じたプライベートネットワーク内のDNSドメインのレコードを管理するコンテナ
- VPC内のDNSドメインに対して、どのようにトラフィックをルーティングするかを定義
- 1つのプライベートホストゾーンで複数VPCに対応
- VPCが相互アクセス可能であれば複数リージョンのVPCでも、同じホストゾーンを利用可能

## [Q]レコードタイプ

あなたはソリューションアーキテクトとして、AWS上でWEBアプリケーションを構築しています。この構成に対して、example.comのドメイン名を利用したいと考えています。Route53のレコードへの設定が必要となります。あなたは登録したホスト名を別のドメイン名に転送したいと考えています。

どのようにRoute53を設定するべきでしょうか？

- 1) エイリアスレコードを利用して、転送先ドメイン名を指定する。
- 2) AAAAレコードを利用して、転送先ドメイン名を指定する。
- 3) CNAMEレコードを利用して、転送先ドメイン名を指定する。
- 4) DNSSECレコードを利用して、転送先ドメイン名を指定する。

# レコードタイプ

ルーティング方法を設定するためにDNSレコードを作成し、各種レコードを設定する

SOA	ドメインのDNSサーバー／ドメイン管理者のメール・アドレス／シリアル番号などを保持して、ゾーン転送時に情報が更新されているかの判断に利用する
A	ホスト名とIPv4アドレスの関連づけを定義するレコード
MX	メールの配送先（メールサーバ）のホスト名を定義するレコード
CNAME	正規ホスト名に対して別名を定義するレコード。特定のホスト名を別のドメイン名に転送する時などに利用する

他のレコードタイプは以下を参照

[https://docs.aws.amazon.com/ja\\_jp/Route53/latest/DeveloperGuide/ResourceRecordTypes.html](https://docs.aws.amazon.com/ja_jp/Route53/latest/DeveloperGuide/ResourceRecordTypes.html)

# エイリアスレコード

CloudFrontやELBなどのAWSリソースをドメインと関連付ける際にはAWS専用のエイリアスレコードを利用する。

- エイリアスレコードはDNSクエリにAWSサービスのエンドポイントのIPアドレスを返答することで、AWSリソースにドメイン名を設定することができる。
- 以下のサービスに利用
  - 静的ウェブサイトとして設定されたS3バケット
  - CloudFront
  - ELB
  - AWS Elastic Beanstalk 環境
- IPアドレスバージョンに応じたタイプ
  - エイリアスターゲットの IP アドレスを伴う A レコード (IPv4 アドレス)
  - エイリアスターゲットの IP アドレスをAAAA レコード (IPv6 アドレス)

# [Q]ルーティングポリシーの選択

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冗長構成を高めるためにELBの背後に複数のEC2インスタンスを利用しています。このアプリケーションに対して、Route53を利用して、通信の遅延発生を最小限に抑える構成が必要です。

このシナリオでAWS Route 53をどのように構成する必要がありますか？

- 1) フェイルオーバールーティングポリシーを使用する。
- 2) レイテンシールーティングを利用する。
- 3) 加重ルーティングポリシーを使用する。
- 4) シンプルルーティングを利用する。

# [Q]ルーティングポリシーの選択

様々なルーティング方式を選択して設定することが可能

シンプルルーティング	<ul style="list-style-type: none"><li>□ レコードセットにおいて事前に設定された値のみに基づいてDNSクエリに応答するルーティング方式</li><li>□ 静的マッピングによりルーティングを決定する。</li></ul>
加重ルーティング	<ul style="list-style-type: none"><li>□ 複数エンドポイントに重みを設定して、重みに応じてDNSクエリに応答するルーティング方式</li><li>□ 重みづけの高いエンドポイントに多くルーティングする。</li></ul>
フェールオーバー ルーティング	<ul style="list-style-type: none"><li>□ ヘルスチェックに基づいて、利用可能なリソースにDNSクエリを応答するルーティング方式</li><li>□ 利用可能なリソースにルーティングされる。</li></ul>
複数値回答ルーティング	<ul style="list-style-type: none"><li>□ 複数のリソースにトラフィックをルーティングさせる方式。</li><li>□ ランダムに選ばれた最大8つの別々のレコードにIPアドレスを設定して、複数の値を返答するルーティングさせる。</li><li>□ IPアドレス単位でヘルスチェックを実施してルーティングすることで、正常なリソースの値を返す。ELBに代わるものではないが、正常を確認して複数のIPアドレスを返す機能により、DNSを使用してアベイラビリティとロードバランシングを向上させることができる。</li></ul>

# [Q]ルーティングポリシーの選択

様々なルーティング方式を選択して設定することが可能

## レイテンシールーティング

- リージョンのレインテンシーに応じて、DNSクエリに応答するルーティング方式。ユーザーの最寄りのリージョンになることが多い。
- リージョン間のレインテンシーが低い方へルーティングされる。

## 位置情報ルーティング

- ユーザーのIPアドレスにより位置情報を特定して、地域ごとに異なるレコードを返すルーティング方式
- ネットワーク構成に依拠しない精度の高いレコード返答の区分けが可能となる。

## 地理的近接性ルーティング

- ユーザーとリソースの場所に基づいて地理的近接性ルールを作成して、トラフィックをルーティングする方式  
-AWSリソースを使用している場合は、リソースを作成したAWSリージョンを場所とする。  
-AWS以外のリソースを使用している場合は、リソースの緯度と経度で位置を場所とする。
- 必要に応じてバイアスを設定し、地理的なリソースの配信範囲を調整することができる。
- トラフィックフローを利用する必要がある。

# ルーティングポリシーの選択

様々なルーティング方式を選択して設定することが可能

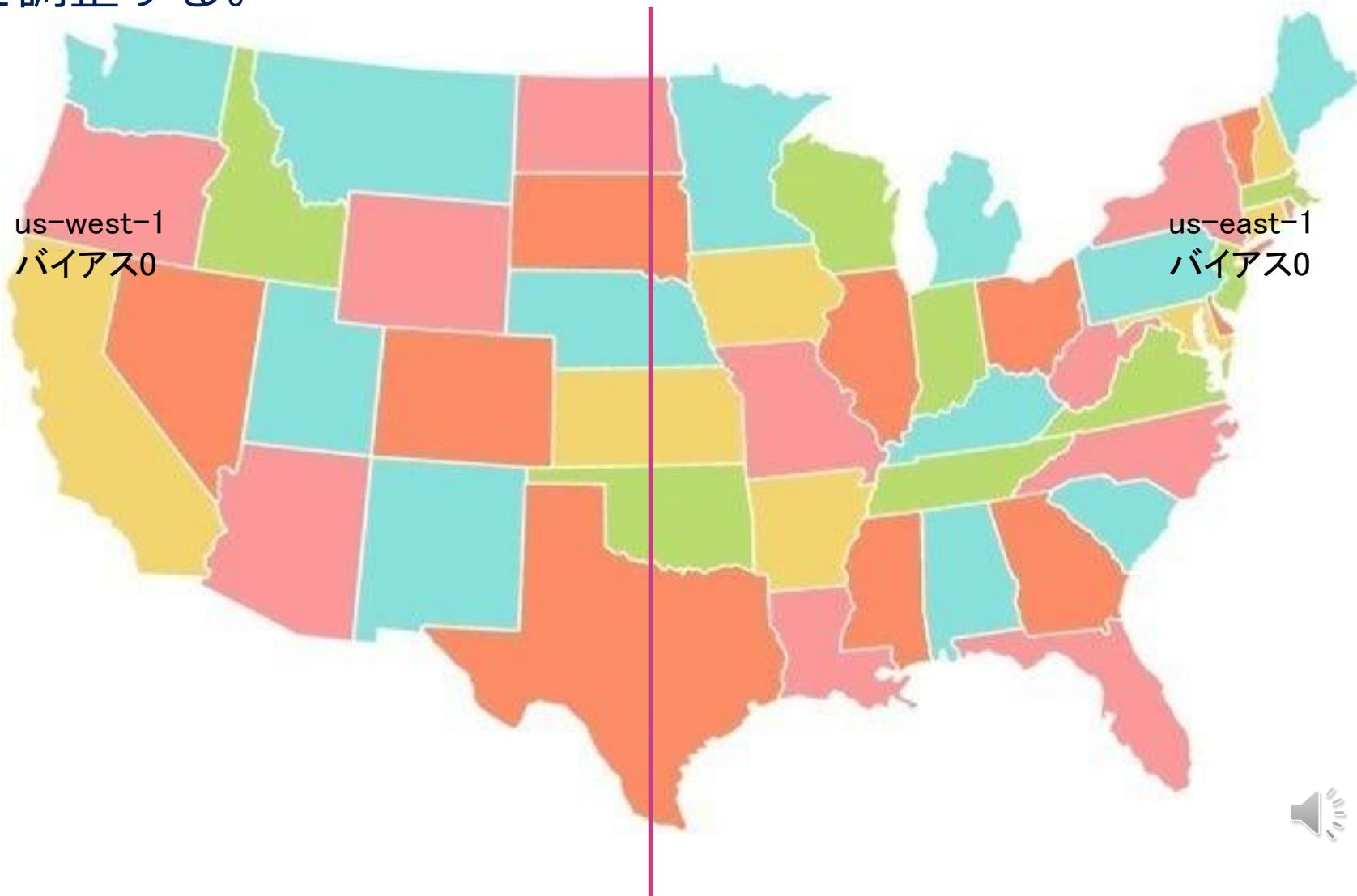
## IP ベースの ルーティングポリシー

- トラフィックの送信元の IP アドレスがわかっている場合に、IP アドレスによるユーザーの位置に基づいてトラフィックをルーティングする。
- ユーザーの IP アドレスからエンドポイントにマッピングすることで、きめ細かなルーティング制御が可能
- 顧客ごとに特有な情報に基づいてルーティングを最適化
- ユースケース
  - ✓ 特定の ISP から特定のエンドポイントにエンドユーザーをルーティングしたい場合
  - ✓ 位置情報ルーティングなど、既存の Route 53 ルーティングタイプにオーバーライドを追加



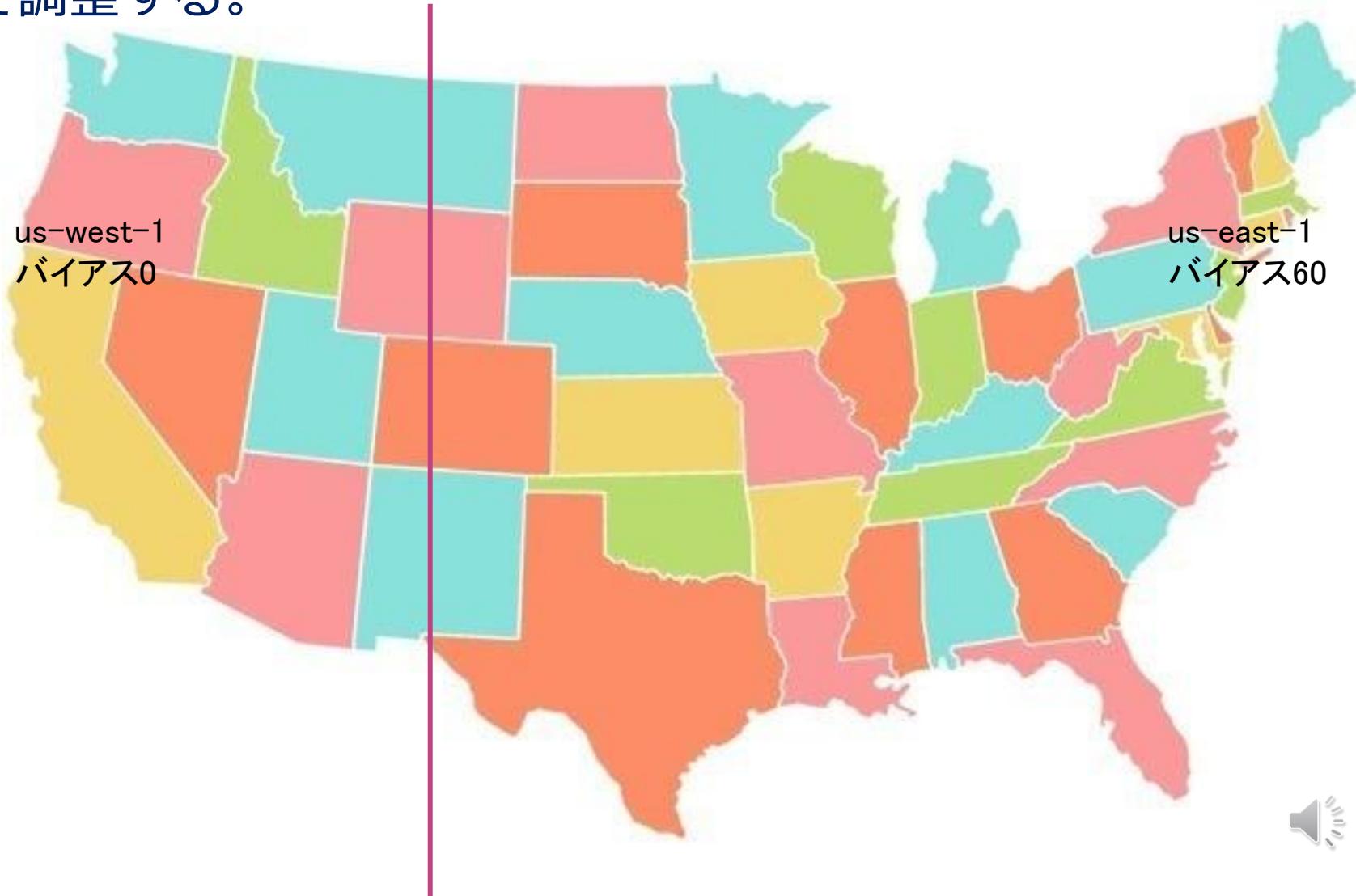
# 地理的近接性ルーティング

複数リージョンにあるリソースのルーティング範囲をバイアスで調整する。



# 地理的近接性ルーティング

複数リージョンにあるリソースのルーティング範囲をバイアスで調整する。



# [Q]フェールオーバー構成

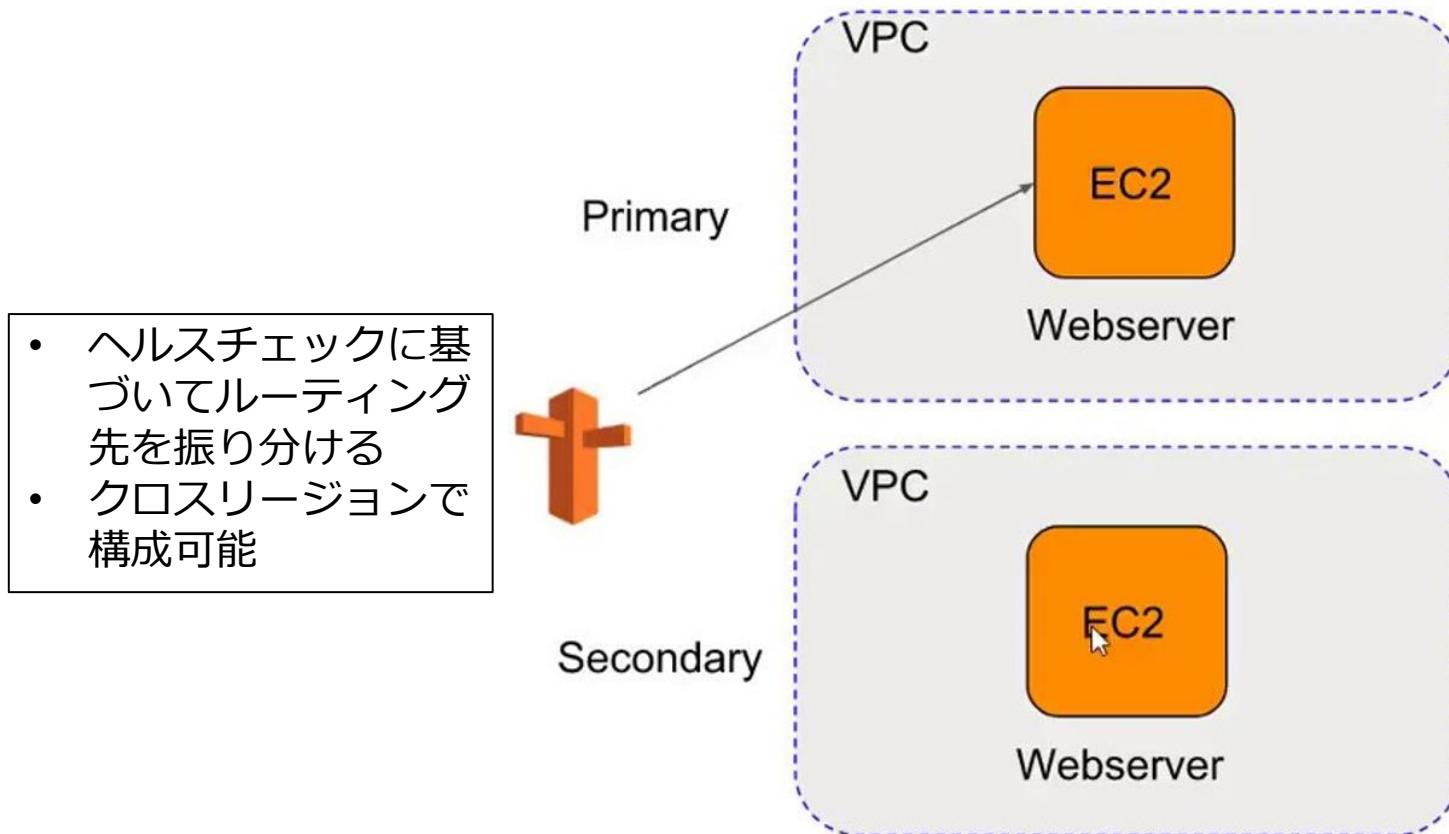
ある企業では2つのEC2インスタンスを利用してアプリケーションを構築しています。あなたはソリューションアーキテクトとして、EC2インスタンスに設定したALBに対してフェールオーバーを実行できるようにRoute53を設定することにしました。その際には、セカンダリALBを指すようにDNSエイリアスレコードを更新する必要があります。

フェイルオーバープロセスを自動化するために必要なRoute53の設定はどれでしょうか？

- 1) ELBヘルスチェックタイプを選択して、Route53を構成する。
- 2) EC2ヘルスチェックタイプを選択して、Route53を構成する。
- 3) ALBエンドポイントを指すCNAMEレコードをAmazon Route53に作成する
- 4) Amazon Route53ヘルスチェックを有効にして、ルーティングポリシーを設定する。

# フェールオーバー構成

フェールオーバー構成はRoute53のヘルスチェック機能を利用したプライマリーとセカンダリーの冗長構成のこと



# [新Q]フェールオーバー構成

ある企業はAWSにホストされた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利用ユーザーに重要なデータを処理しているため、非常に高い可用性を担保することが必要です。そのため、複数リージョンを利用した冗長構成にする必要があります。

プライマリのアプリケーションは東京リージョンにホストされています。加えて、ディザスター・リカバリー用のセカンダリーアプリケーションをソウルリージョンにホストします。東京リージョンで障害が発生した場合に、セカンダリーに切り替えられる必要があります。

この要件を満たすために、ソリューションアーキテクトはどうすればよいでしょうか。

- 1) Amazon Route53のレイテンシールーティングを利用してフェールオーバーを設定する。ソウルリージョンをプライマリーに、東京リージョンをセカンダリーに設定する。
- 2) Amazon Route53のレイテンシールーティングを利用してフェールオーバーを設定する。東京リージョンをプライマリーに、ソウルリージョンをセカンダリーに設定する。
- 3) Amazon Route53のフェールオーバールーティングポリシーを構成する。東京リージョンをプライマリーに、ソウルリージョンをセカンダリーに設定する。
- 4) Amazon Route53のフェールオーバーリーティングポリシーを構成する。ソウルリージョンをプライマリーに、東京リージョンをセカンダリーに設定する。

# フェールオーバー構成

フェールオーバー構成はRoute53のヘルスチェック機能を利用して正常なリソースを利用する構成のこと

## フェールオーバー (アクティブ/パッシブ)

- Route 53 はプライマリリソースをアクティブなリソースとしてルーティングする。障害が発生した場合、Route 53 はセカンダリーのリソースをルーティングする。
- フェールオーバーポリシーを使用して設定する。

## フェールオーバー (アクティブ/アクティブ)

- Route 53 は複数のリソースをアクティブとしてルーティングする。障害が発生した場合、Route 53 は正常なリソースにフェイルバックする。
- フェールオーバー以外のルーティングポリシーを使用して設定する。

# [Q]Route53による地域制限

大手メディアはニュース配信アプリケーションをAWS上に構築しています。ユーザーはグローバルに存在しており、グローバルにコンテンツを配信します。アプリケーションは、ALBの背後にあるプライベートサブネットに設置されたEC2インスタンスのフリートを使用しています。中国からの情報制限があり、中国からのアクセスをロックする必要があります。

次のオプションのうち、地域制限を実施できるようにするのはどれですか？（2つ選択してください）

- 1) Route53の位置情報ルーティングポリシーを使用して、コンテンツ配信を、配信権を持っている場所に限定する。
- 2) Route53の地理的近接性ルーティングポリシーを使用して、コンテンツ配信を、配信権を持っている場所に限定する。
- 3) Route53の地域制限を有効化して、特定の地域への配信制限を設定する。
- 4) CloudFrontの地域制限を有効化して、特定の地域への配信制限を設定する。
- 5) CloudFrontの配信ポリシーにより、特定の地域への配信制限を設定する。

# Route53による地域制限

位置情報ルーティングを利用して、コンテンツの配布を配信権限がある場所だけに制限することが可能

## 位置情報ルーティングによる地域制限

- 地域を指定して配信先としての制限設定し、コンテンツを権利がある場所のみに制限することが可能
- 地域に応じてコンテンツを変更するなど、コンテンツ配布のローカライズを実施することが可能
- 特定の地域からのエンドポイントを利用してローカルでのパフォーマンスを向上させる。

# [Q] トラフィックフロー

ある企業では2つのEC2インスタンスを利用して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、Route53を利用したルーティング設定を行っています。設計方針を整理したところ、組織構造やアプリユーザーが多数かつ複雑であることもあって、複雑なルーティングポリシーを設定することが必要となりました。

Route53を利用した複雑なルーティング設定を効率的に実施する方法を選定してください。

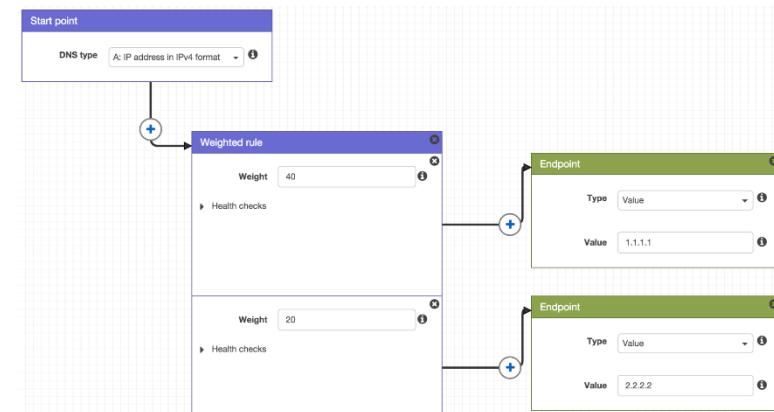
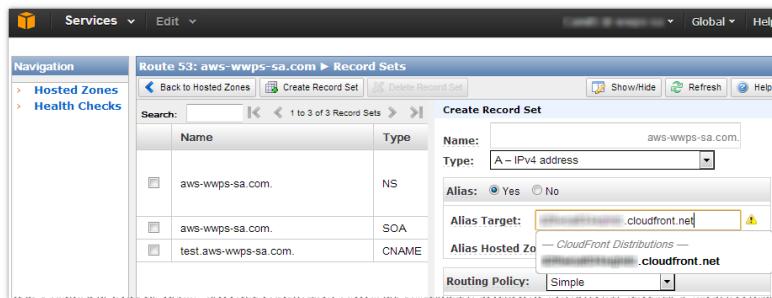
- 1) ALIASレコードを駆使して、フローを作成することでルーティングポリシーを設定する。
- 2) ALIASレコードをトラフィックフローでフロー化することでルーティングポリシーを設定する。
- 3) トラフィックフローを用いて、順序を設定することでルーティングポリシーを設定する。
- 4) JSON/YAMLファイルにルーティングを設定することでルーティングポリシーを設定する。

# トラフィックフロー

従来はALIASレコードを駆使して、複雑なルーティングポリシーを作成していたが、トラフィックフローによる視覚的なフローでの複雑なポリシー設定が可能となった

ルートレコードセット画面で  
ルーティングポリシーを設定

トラフィックフローで  
ルーティングポリシーを設定



## [Q]TTL

ある企業では2つのEC2インスタンスにELBとRoute53が設定された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example.comというドメインを利用して公開されています。最近になって災害復旧計画を整備したため、あなたはソリューションアーキテクトとして、DNSルーティングにより冗長構成とするように構成を見直しています。そのために、Route53の既存のホストゾーンに対して新しいドメインに再設定しました。しかしながら、1時間たっても新しいドメインへのルーティングが実行されません。

この問題の最も可能性が高い要因はどれでしょうか？

- 1) TTLが有効期限となっている。
- 2) CNAMEレコードが正しく構成されていない。
- 3) ヘルスチェックエラーが発生している。
- 4) ドメインが取得されたばかりで、反映されていない。

# TTL

再帰的な DNSリゾルバでレコードに関する情報をキャッシュして保持しておく時間(秒単位)を設定できる。

- DNSリゾルバーはリゾルバは自分の知っているDNSサーバへ問い合わせを行い、IPアドレスの割り出し(名前解決)を行う機能。つまりドメイン名の対応付けを確認してくれる。
- 再帰的な DNSリゾルバはドメインに変更がないか再度問い合わせること。
- その情報をキャッシュに保持することで、毎回リゾルバが名前解決しなくともドメインの情報を把握することが可能となる。
- 再帰的な DNSリゾルバで Route 53 に対して実行する必要がある呼び出しの数を減らすことが可能

# [Q]オンプレミス環境への適用

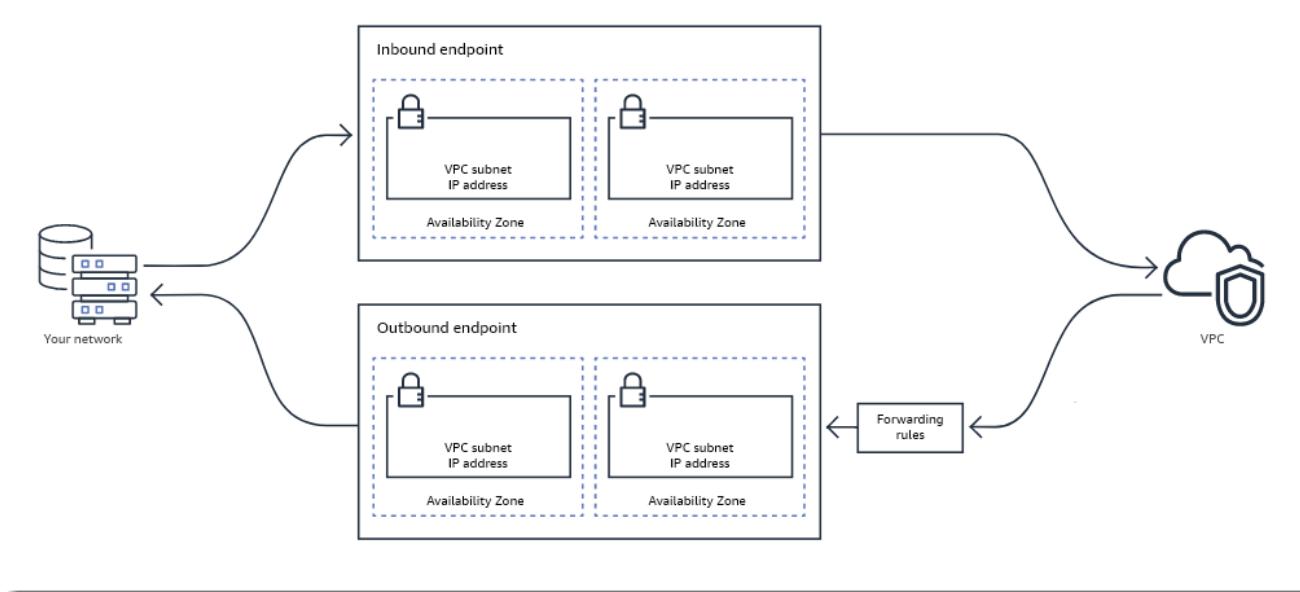
ある企業では2つのEC2インスタンスにELBとRoute53が設定されたアプリケーションを運用しています。このアプリケーションはexample.comというドメインを利用して公開されています。あなたはソリューションアーキテクトとして、Route53を利用してオンプレミス環境にも適用しようとしています。オンプレミスネットワーク内のリソースのDNSクエリをAWS VPCから解決することが必要です。

この要件を満たす、設定は次のうちどれですか？（2つ選択してください）

- 1) Route 53 リゾルバーでインバウンドエンドポイントを作成して、オンプレミスネットワーク上のDNSリゾルバーがDNSクエリをRoute 53リゾルバーに転送できるようにする。
- 2) Route 53 リゾルバーでアウトバウンドエンドポイントを作成して、Route 53 リゾルバーがオンプレミスネットワーク上のリゾルバーにクエリを転送できるようにする。
- 3) Route 53 リゾルバーでインバウンドエンドポイントを作成して、Route 53 リゾルバーがオンプレミスネットワーク上のリゾルバーにクエリを転送できるようにする。
- 4) Route 53 リゾルバーでアウトバウンドエンドポイントを作成して、オンプレミスネットワーク上のDNSリゾルバーがDNSクエリをRoute 53リゾルバーに転送できるようにする。
- 5) Route 53 リゾルバーからVPCエンドポイントを利用して、Route 53 リゾルバーとオンプレミスネットワーク上のリゾルバーがお互いに連携できるようにする。

# オンプレミス環境への適用

Route53リゾルバを利用して、オンプレミスからVPC内の名前解決が可能となった。これにより、オンプレミス、AWS相互の名前解決を実現することができる。



- ✓ インバウンドエンドポイントを作成し、VPCへの接続を設定
- ✓ アутバウンドエンドポイントを作成し、アウトバウンドへの通信を設定