

# Amazon Web Service

AWSで始めるクラウド入門

2024/08/03

# Infrastructure

## インフラとは

- ・ 生活を送るために必要な電気や水道などの設備を**社会インフラ**と呼ぶ
- ・ ITサービスなどを運営するために必要な設備も**インフラ**と呼ぶ
- ・ インフラは言わば**縁の下**の力持ち
- ・ この資料では**ITインフラ**のことをインフラとする
- ・ インフラには**アプリケーションサーバー**や**データベース**なども含まれる
- ・ インフラを用意する方法には**オンプレミス**と**クラウド**が主にある

# On-Premises

## オンプレミスとは

- ・ オンプレミスは**社内にサーバーを用意して提供**する手法
- ・ 初期投資は必要になるが、**運用資金はあまりかからない**
- ・ 全ての責任が自社にあり、**障害時にも自社で対応**する必要がある
- ・ 昔から使われている手法で多くの会社が導入している

# Cloud

## クラウドとは

- ・ クラウドは**社内にサーバーを用意せず**、  
クラウドサービス内にサーバーを構築して提供する手法
- ・ 初期投資がほとんど必要なく、  
サーバーを使用した時間などによる**従量課金のコストが必要**
- ・ 責任はサービス提供社と自社にあり、**障害時にも責任が分散している**
- ・ 近年メジャーになってきた手法で  
さまざまな会社が**オンプレミスから移行**している

# Example of Cloud

クラウドサービスの例



**Amazon Web Service**

Amazon

汎用・万能



**Google Cloud**

Google

AI系APIに特化



**Microsoft Azure**

Microsoft

Windows, OpenAI特化

# Example of Cloud

クラウドサービスの例



**Amazon Web Service**

Amazon



**Google Cloud**

Google



**Microsoft Azure**

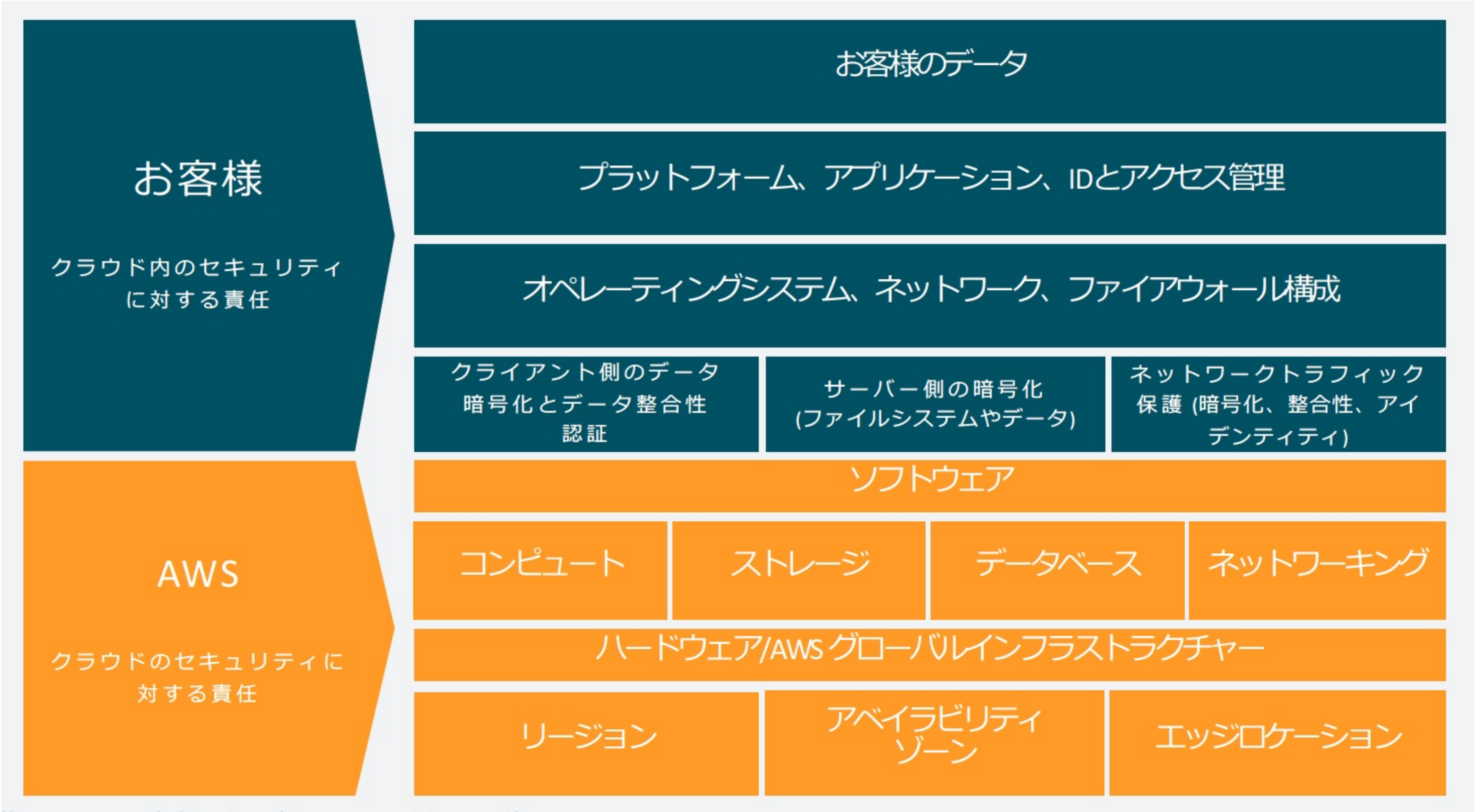
Microsoft

この資料では **Amazon Web Service (AWS)** を使用して解説します



# Shared Responsibility Model

## 責任共有モデル



# Shared Responsibility Model

## 責任共有モデル

- ・ サービスによって異なるが、基本的に**物理的な障害はAWSに責任がある**
- ・ **内部のアプリやAWSアカウントの管理などは利用者に責任がある**



# Region

## リージョン

- ・リージョンは**物理的なデータセンターの場所**を指す
- ・世界各地にリージョンがあり、**今もなお増え続けている**
- ・日本国内には**AWS, GCP, Azureともに2リージョン**存在する  
(AWS, GCPはともに東京, 大阪リージョン、  
Azureは東日本, 西日本リージョンが存在する)
- ・アカウントではデフォルトで**国内外17リージョンが有効**になっている
- ・**ユーザーから近い場所や利用したいサービスを考慮して選定**する必要がある

# Region

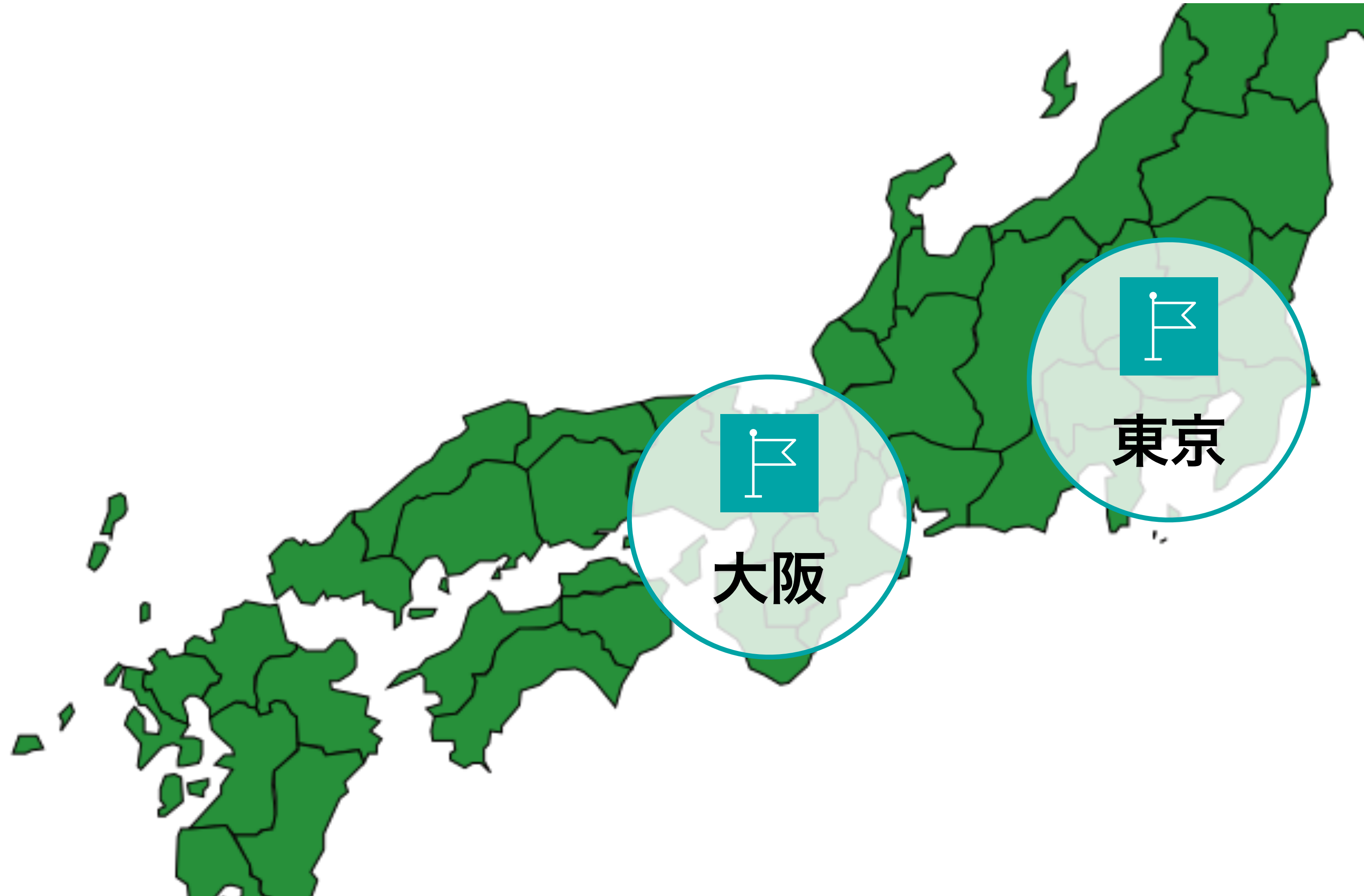
## リージョン

- ・ リージョンコードは **[エリア名]-[エリア内位置]-[番号]** で構成されている
  - ▶ 東京リージョン: **ap-northeast-1**
  - ▶ 大阪リージョン: **ap-northeast-3**

※ ap は Asia Pacific の略

# Region

リージョン



# Region

リージョン

**AWSのリージョン変更は画面内右上のこの部分から可能**



# Tokyo Region vs Osaka Region

## 東京リージョン vs 大阪リージョン

- ・ **西日本**で使う用途では**大阪リージョン**を選ぶべきか？
  - ▶ 日本では基本的に**東京リージョン**を使うべき
  - ▶ 東京リージョンの方が古くからあり、使用できるサービスが**豊富**
  - ▶ 日本のネットワークは基本的に東京に集中しているため、東京リージョンの方が速い場合が多い
- ・ 銀行などの基幹サービスはサービスの完全なダウンを防ぐために両リージョンに構成して**冗長運用**することがある



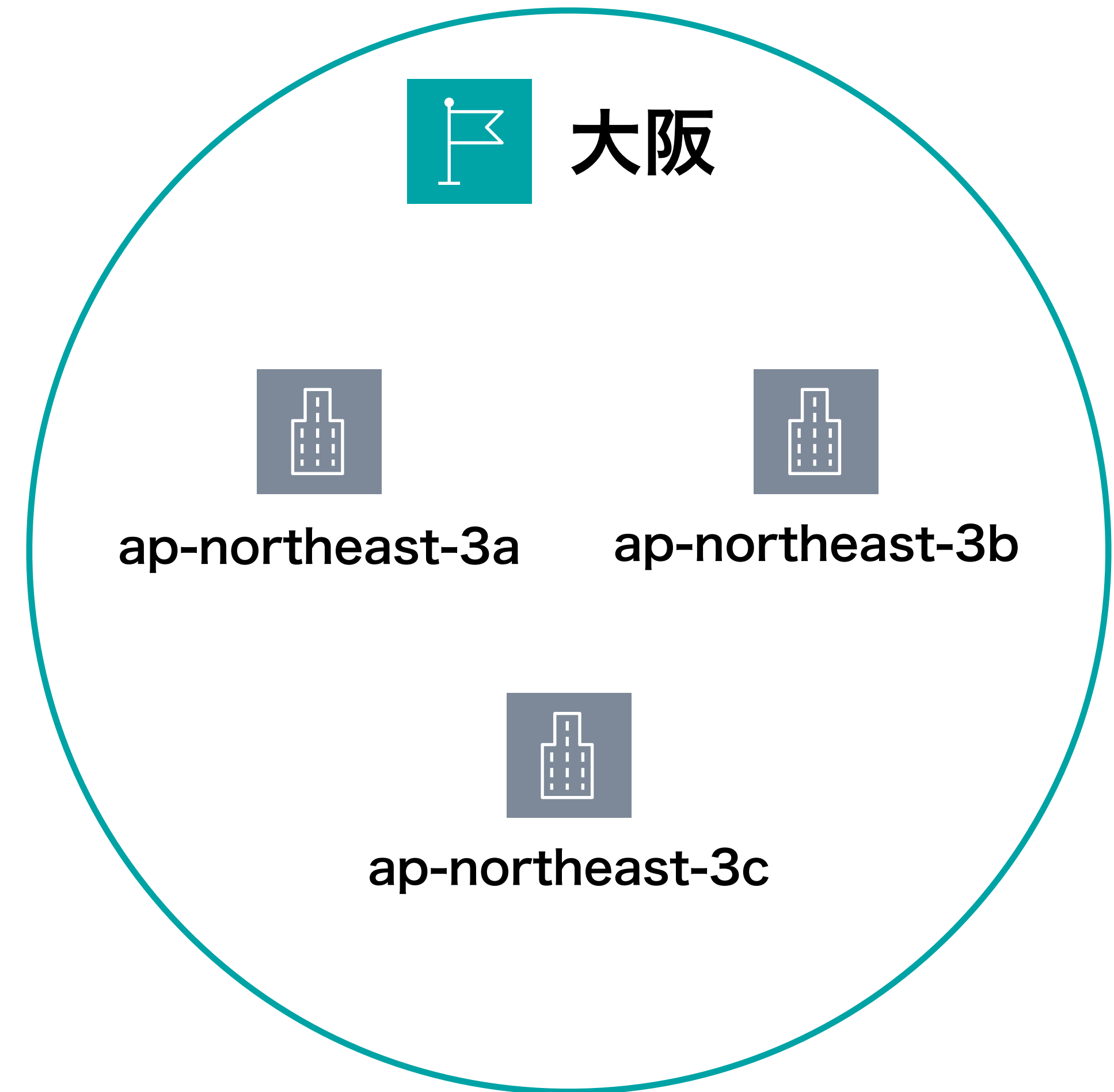
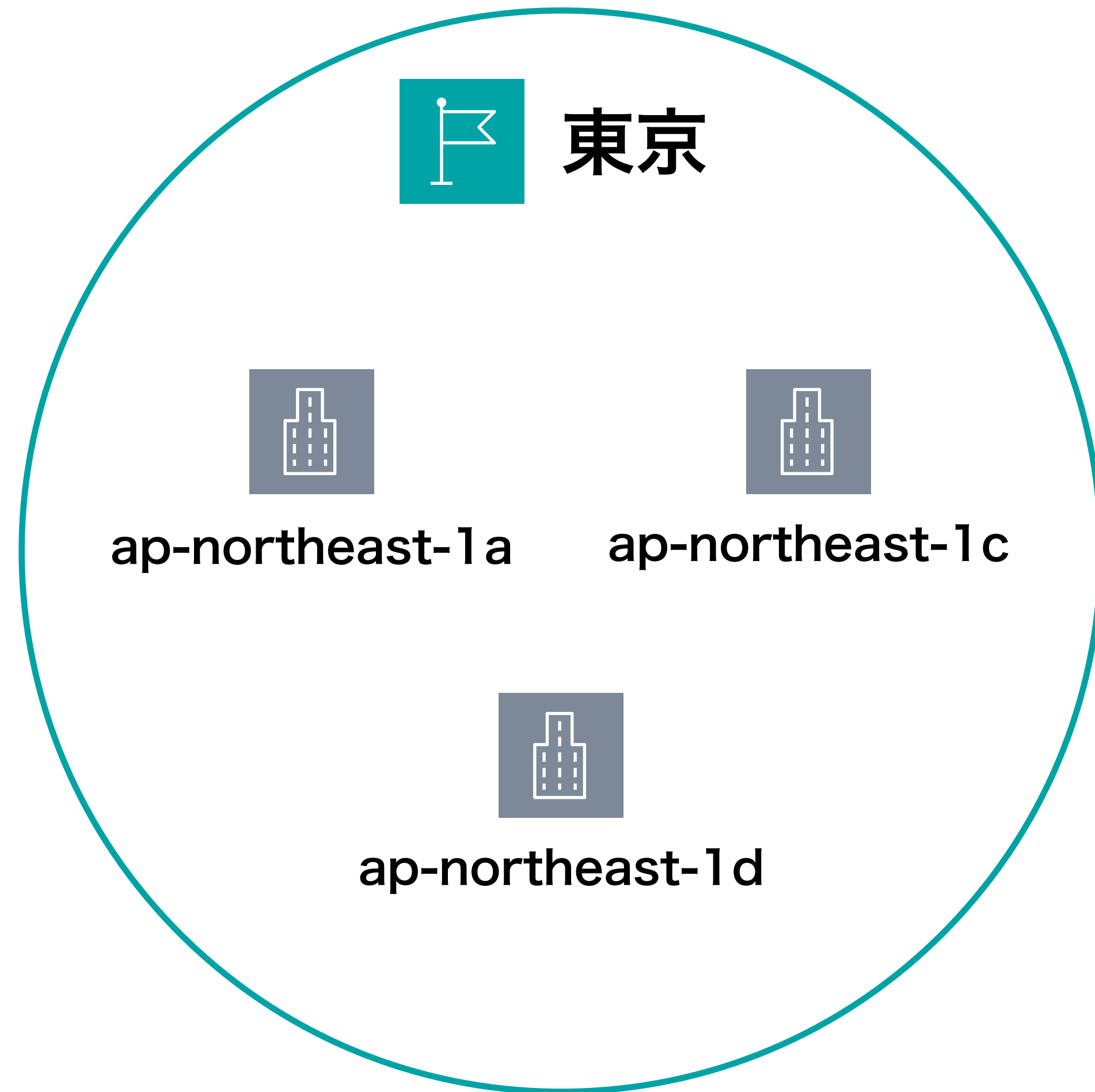
# Availability Zone

## アベイラビリティゾーン (AZ)

- ・ 1つのリージョン内に複数のAZが存在する
- ・ 自然災害・電力・ネットワーク面で独立したデータセンター
- ・ 仮想的ではなく物理的な距離があり、  
それぞれのAZ間の距離は100km以下かつ  
ネットワーク遅延が2ms以下という条件で構成されている
- ・ 1AZで障害が発生しても、他のAZで対応が可能
- ・ 東京・大阪リージョンともに3つのAZが存在する

# Availability Zone

アベイラビリティゾーン (AZ)



# Services

## クラウドが提供するサービスの種類

- AWSやGCP、Azure全てのクラウドにおいて、**さまざまなサービスが提供**
- AWSにおいてサービスは2024年4月時点で**321サービス存在**
- この資料では有名サービスを抜粋して紹介

# Networking

## ネットワーキング

### Virtual Private Cloud (VPC)

- ・クラウドに**ネットワークを構築**
- ・サブネットやルートテーブルもあり、**詳細な設計が可能**
- ・デフォルトを使用することもできるが、構築するものに合わせた設計をすることで、**クラウドリソースを保護**したりできる



# Networking

## ネットワーキング

### Amazon Route 53

- AWSが提供する**DNSサーバー**
- 1パブリックホストゾーンあたり  
**月額固定 \$0.50** で管理可能
- 世界4ヶ所に設置されたDNSサーバーに登録と、  
**かなり冗長化**された構成になっている



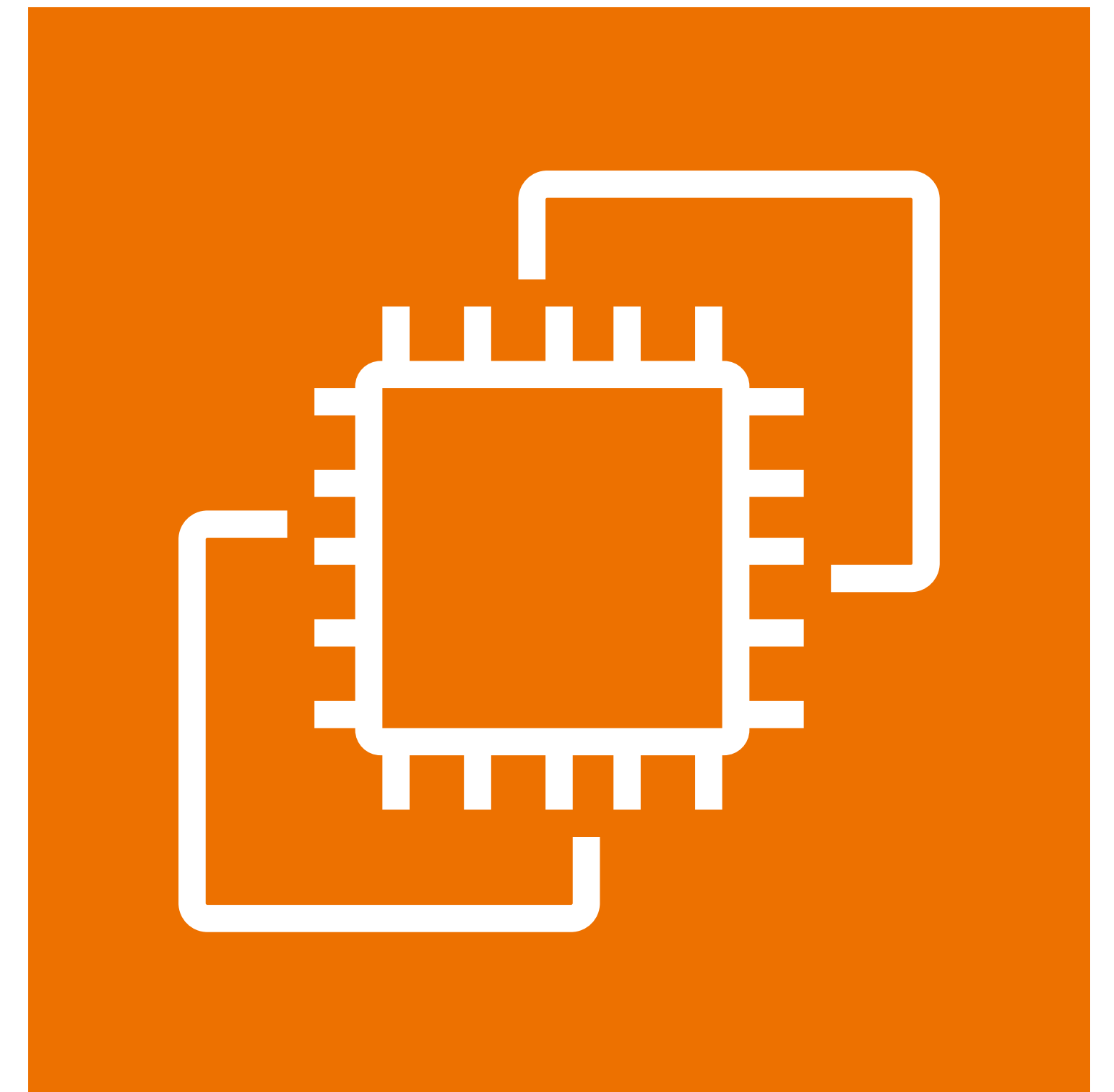


# Computing

## コンピューティング

### Elastic Compute Cloud (EC2)

- 基本的に何でもできる**万能な仮想サーバー**
- 仮想サーバーの単位を**インスタンス**という
- インスタンスの使用は**1秒単位で課金される**  
(課金はインスタンスタイプ単価×時間)  
(最小課金時間は60秒)
- 何でもできるため、**利用者の責任範囲が広い**

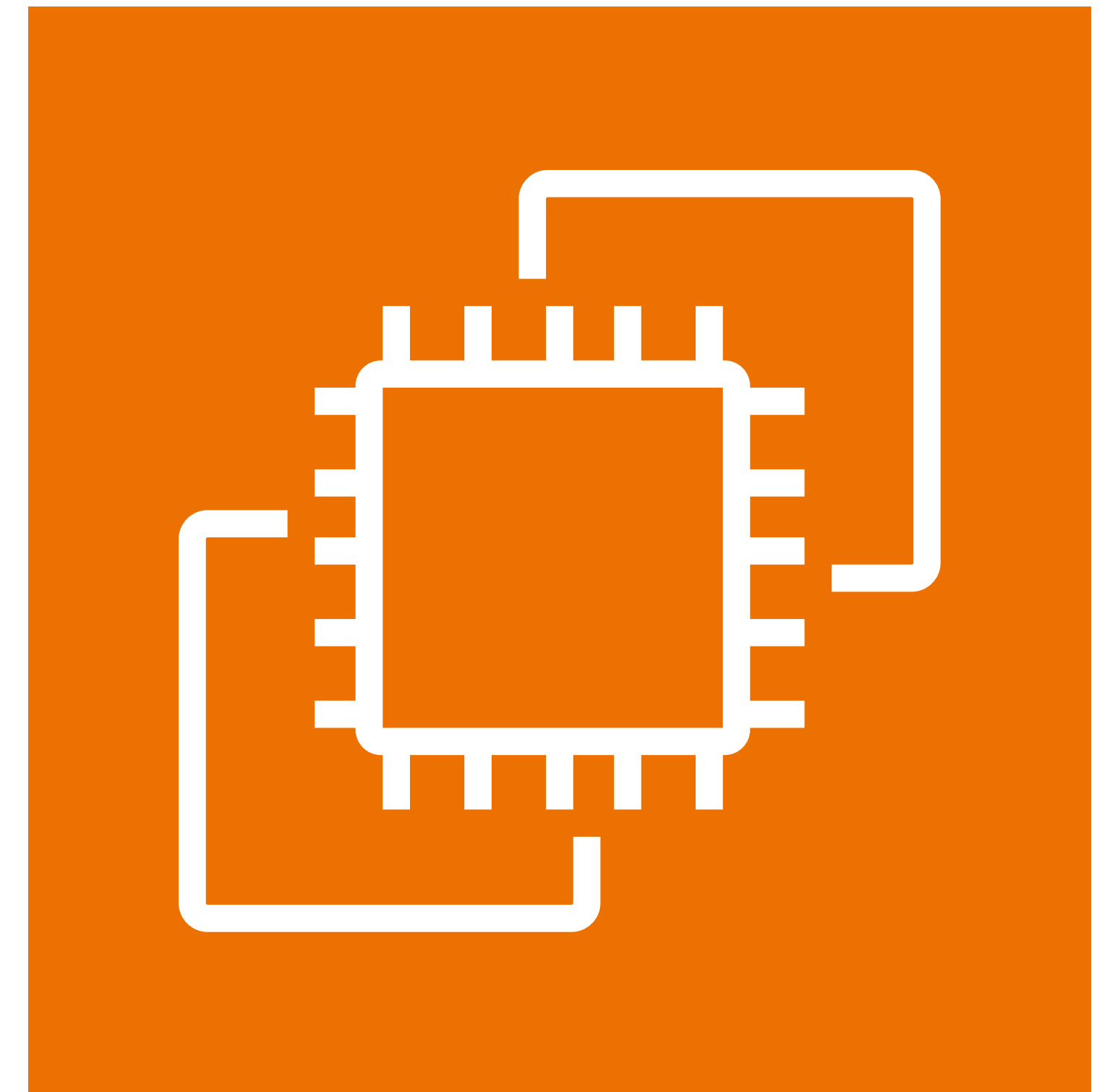


# Computing

## コンピューティング

### Elastic Compute Cloud (EC2)

- ・ 起動時にOS（マシンイメージ）を選択して起動するため、**OSのインストールは必要ない**
- ・ さまざまな性能の**インスタンスタイプ**が用意
- ・ ストレージは **Elastic Block Store** というEC2に紐づけるストレージとして課金される（課金は種類単価×容量×時間）



# Computing

## コンピューティング

### AWS Lambda

- ・ **サーバーを完全に気にせず、実行が可能**
- ・ AWS内で代表格の**サーバーレスサービス**
- ・ **基本0円から始められるサービスで、**  
小規模サービスを安価に運用できる
- ・ ランタイムは毎回リセットされるため、  
ライブラリのインストールなどで**実行時間が長い**



# Computing

コンピューティング

## AWS Lambda

- API Gatewayと組み合わせることで、**Lambdaを使った格安のAPI**が構築できる
- **PythonやRuby, Java**などといった言語をランタイムとして使用できる
- 実行開始から**1ms単位で料金が計算**される  
(メモリ種類単価×時間で計算)

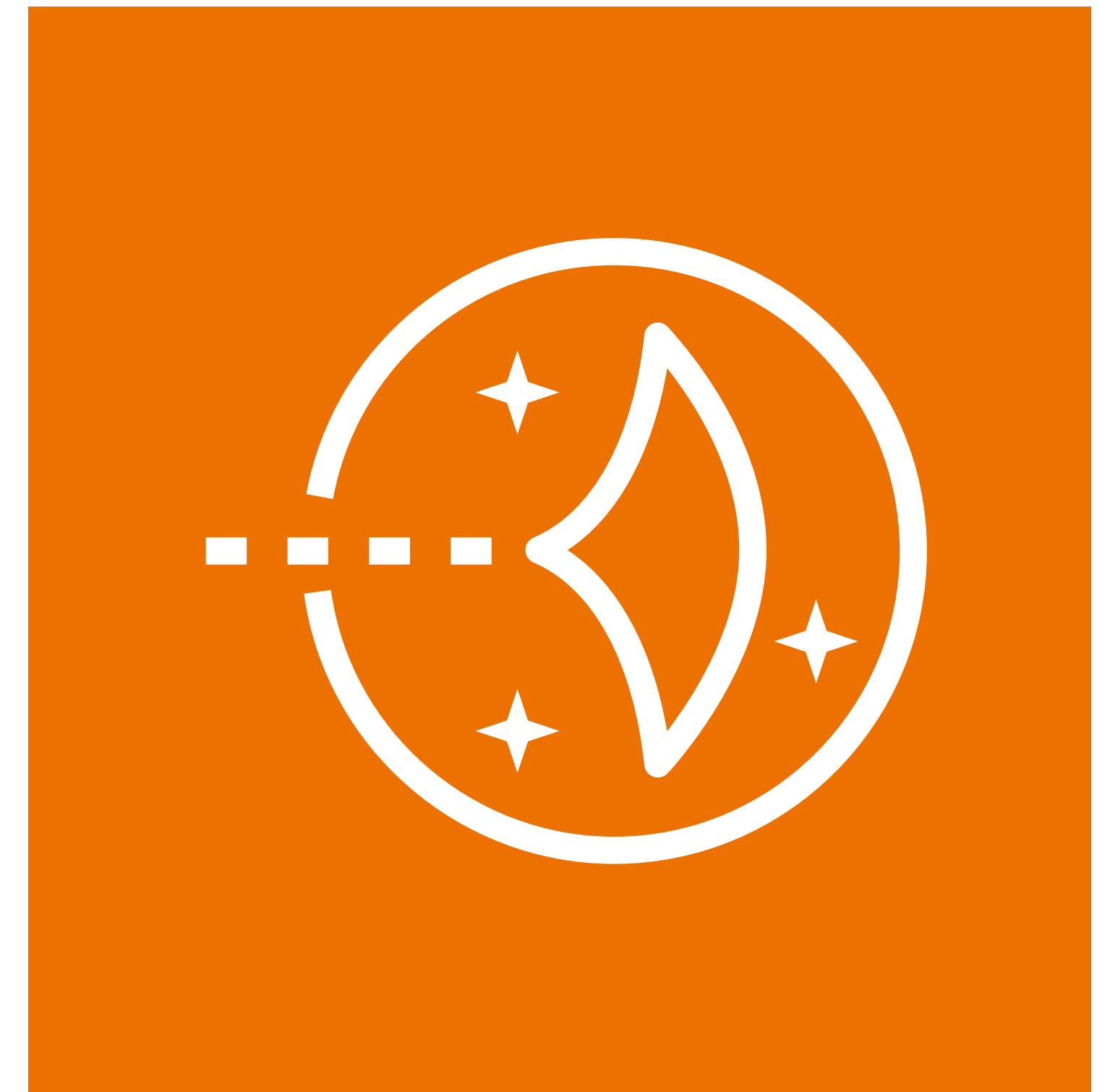


# Computing

## コンピューティング

### Amazon Lightsail

- CPU、メモリ、ストレージ、データ転送が  
予めセットになったインスタンス
- 月額固定料金で**価格が予想しやすい**
- Wordpressなどを起動時にインストールでき、  
初心者にかなり優しいサービス
- 小規模なものでは**Lightsailのみで作成できる**





# Containers

コンテナ

## Elastic Container Service (ECS)

- **Dockerイメージから直接起動**できる
- ECSの管理単位には以下のものがある
  - ▶ **タスク**: コンテナのAWSでの名称
  - ▶ **サービス**: タスクを管理する単位
  - ▶ **クラスター**: サービスを管理する単位



# Containers

コンテナ

## Elastic Container Service (ECS)

- ・ サービス内には複数のタスクを起動し、**負荷を分散**させられる（ロードバランサー必要）
- ・ 負荷を条件に**自動的にタスクを増やす**ことも可能
- ・ 使用するインスタンスは  
**EC2とFargate**から選択できる



# Containers

コンテナ

## Elastic Container Service (ECS)

- ECSは**Amazon独自の**  
コンテナオーケストレーションサービス
  - ▶ **AWS独自で、GCP, Azureでは利用できない**
- **小規模**なアプリケーションに向いている
- Dockerイメージは  
**Elastic Container Registry (ECR)** で管理



# Containers

コンテナ

## Elastic Container Registry (ECR)

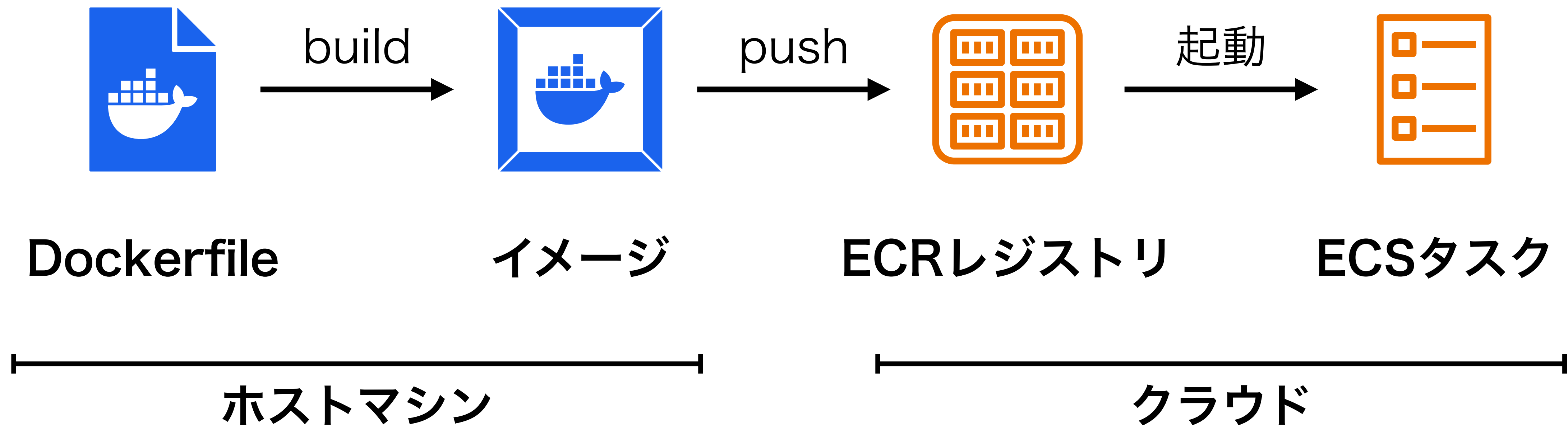
- ・ ECSで使用するDockerイメージを保存するレジストリ
- ・ レジストリには**予めビルドしたイメージ**をプッシュする
- ・ インスタンスで使用されている**アーキテクチャ**を**考えてビルド**する必要がある



# Containers

コンテナ

## Elastic Container Registry (ECR)





# Containers

## コンテナ

### Elastic Kubernetes Service (EKS)

- Googleが開発した**Kubernetes**という  
コンテナオーケストレーションサービスを  
起動できるサービス
  - ▶ **AWS, GCP, Azure**で利用できる
- **大規模**なアプリケーションに向いている



# Containers

## コンテナ

### AWS Fargate

- EC2は詳細な設定を行わなければならないが、**Fargateはコンテナ専用カスタム**されており、少ない設定で利用できる
  - ▶ サーバーの**詳細な設定を気にしなくて良い**
- 利便性が高いため、以前まで料金が高額だった
  - ▶ 現在は安価になってきている



# Front-End Web & Mobile

ウェブやモバイルのフロントエンド

## AWS Amplify

- TypeScriptに特化した自動デプロイなどの機能を搭載した開発プラットフォーム
- Next.js など SSR Webアプリケーションやネイティブモバイルアプリケーションに特化している



# Storage

## ストレージ

### Simple Storage Service (S3)

- ・ AWSを代表するサービスの一つ
- ・ さまざまなタイプがあり、**コストを最適化**できる
- ・ Standardは**リージョン内全てのAZにデータが保存**され、99.999999999%のデータ耐久性
- ・ **1AZで使用しているかのように使用できる**
- ・ **静的Webサイトをホスト**できる



# Database

データベース

## Relational Database Service (RDS)

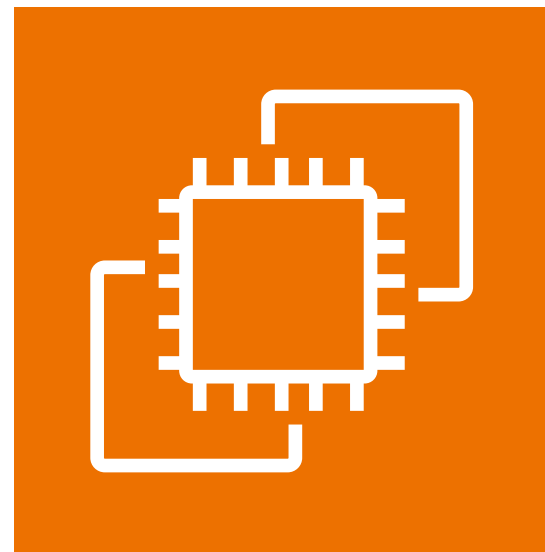
- ・ データベースサーバーを構築するサービス
- ・ MySQLやPostgreSQLなどの**SQL**を構築できる
- ・ バックアップやマルチAZなども  
**AWS**が行ってくれる



# Database

## データベース

### EC2でもDBサーバー構築は可能だが...



**EC2**

- ・ コンソールから  
インスタンスを起動
- ・ SSHでサーバーへ接続
- ・ DBのインストール
- ・ バックアップ設定
- ・ サーバー冗長化

etc ...



**RDS**

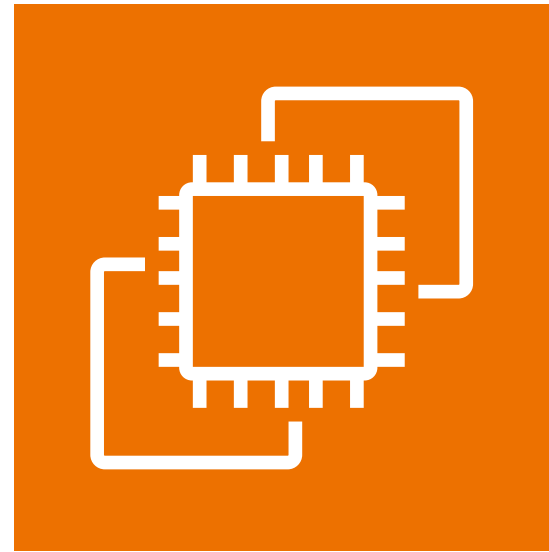
- ・ コンソールから  
インスタンスを起動

**以上！**

# Database

## データベース

### EC2でもDBサーバー構築は可能だが...



#### EC2

- ・ コンソールから  
インスタンスを起動
- ・ SSHでサーバーへ接続
- ・ DBのインストール
- ・ バックアップ設定
- ・ サーバー冗長化

etc ...



#### RDS

- ・ コンソールから  
インスタンスを起動

以上！

**EC2で大抵のものは作れるが、  
何でもEC2で作れば良いわけではない！**

**まずは専用のサービスがないか調べてみる**



# Security

## セキュリティ

### AWS Certificate Manager

- ・ **SSL（通信の暗号化）に使用する証明書を発行できるサービス**
- ・ サブドメインを含めた証明書の発行もできる



# User

ユーザー

## Identity and Access Management (IAM)

- ・ アカウント内の**ユーザー**を作成したりできる
- ・ 許可ポリシーを細かく設定でき、  
**IAMごとにアクセスキーを発行**できる
- ・ 細かく設定することで、**セキュリティを強化**
- ・ アクセスキー流出時にも**被害を最小限に**
  - ▶ 許可を与えすぎないこと



# Monitoring

## モニタリング

### Amazon CloudWatch

- AWSで使用しているリソースの**ログ**や  
設定した**アラートの状態**、  
インスタンスの**メトリクス**をまとめて確認できる  
**モニタリングサービス**
- アラートをトリガーに  
**Lambdaイベントを送信**したりもできる



# Cost Management

## コスト管理

### AWS Cost Explorer

- ・ どのサービスにどれくらいのコストがかかっているのかを取得するツール
- ・ 予算なども設定でき、**閾値を超えた際に通知**
- ・ 普段から定期的にチェックする必要がある



# Cost Estimation

## コスト試算

- ・ AWSはさまざまなリソースに対して課金が発生
- ・ 予めコスト試算をして、どれくらいの予算が必要かを炙り出す必要がある

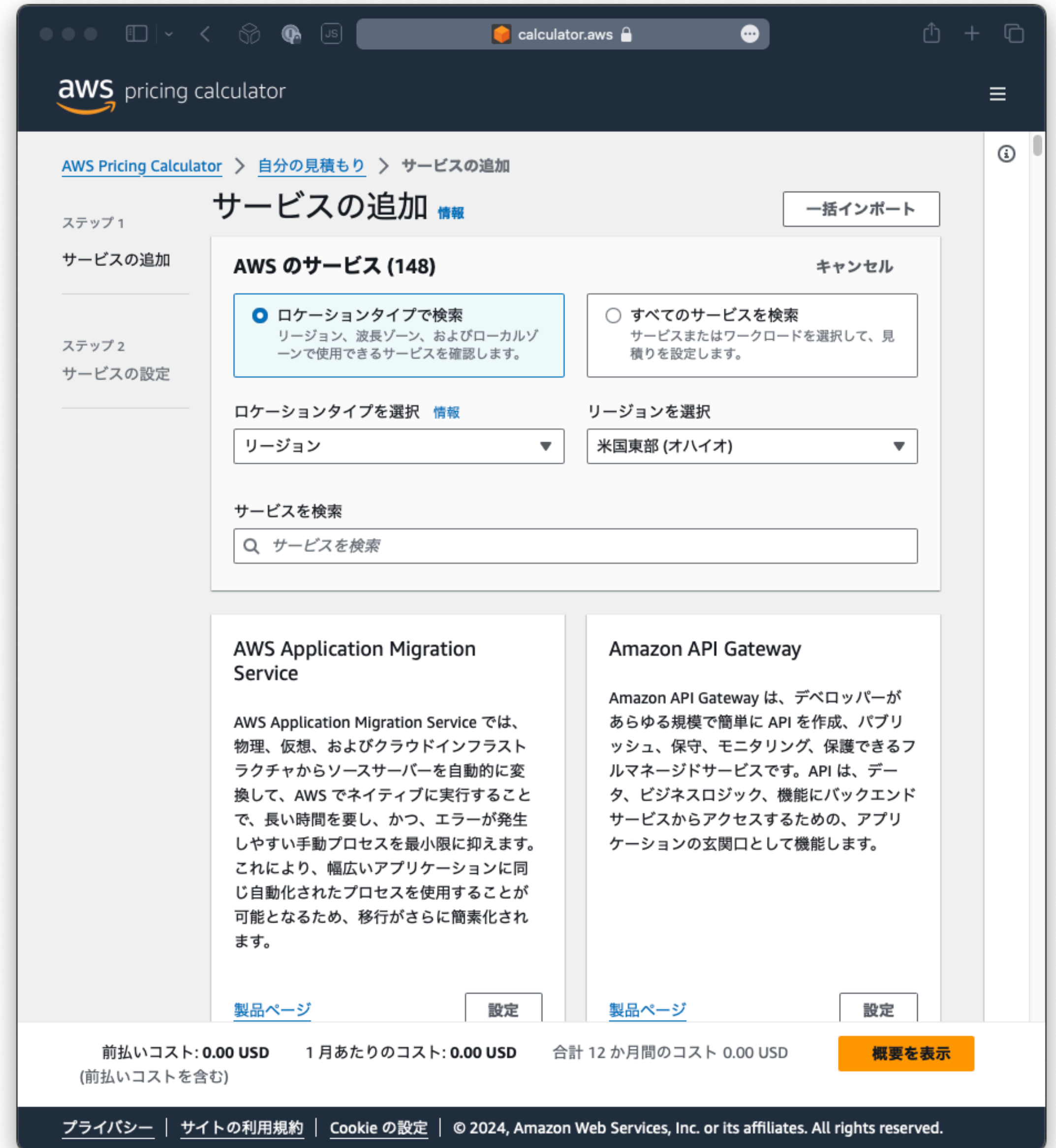
# Cost Estimation

## コスト試算

## AWS Pricing Calculator

- AWS公式のコスト試算ツール
- サービスが全て網羅されており、  
詳細なコスト試算が可能

URL: <https://calculator.aws/>



AWS Pricing Calculator



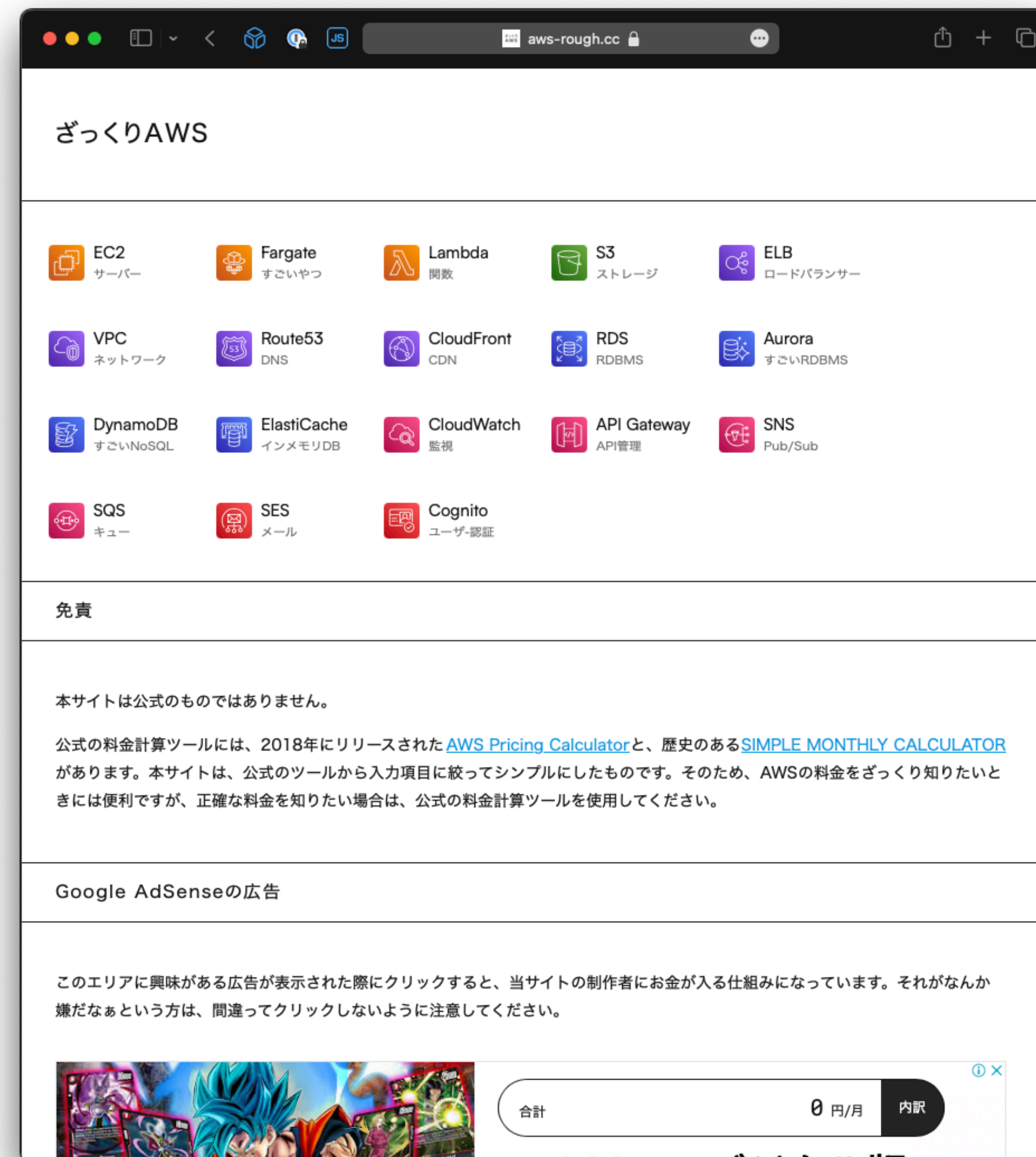
# Cost Estimation

## コスト試算

## ざっくりAWS

- ・ 日本円でざっくりした見積もりができるサービス
- ・ 大きくコストがかかるサービスのみ計算できる

URL: <https://aws-rough.cc/>



ざっくりAWS