



[初級]コンテナ 座学&ハンズオン

Amazon Web Services Japan G.K
2024.1.31

研修は14:00より開始します
それまで事前準備をしつつお待ちください
Teams上の出席簿に忘れずに○を記載ください

事前準備

- Edge、Firefox、もしくは Chrome のインストール
- 配られた URL、アカウント、ユーザ名、パスワードでAWS コンソールにログインできること



事前準備

下記 4 ファイルを Teams よりダウンロードして下さい。

- PDF ファイル（3種類）
 - AWS-Container-Hands-on.pdf / ハンズオン資料
 - AWSハンズオンアカウントの使い方 Event Engine 版.pdf / アカウント利用手順
 - AWS-Container-Hands-on-Zagaku.pdf / 投影資料(参考資料)
- Zip ファイル（1種類）
 - Dockerfile.zip
 - Dockerfile

!!注意!! 正常にスクリプトが動作しなくなる可能性があるため、上書き保存はしないでください

事前準備

時間がある方は、Teams に添付されている「AWSハンズオンアカウントの使い方 Event Engine 版.pdf」を参照して、コンソールへのログインを進めてください。

なお、開始後、コンソールの使い方については簡単に説明します
そこでログインいただく事で問題ございませんのでご安心ください。

自己紹介

守田凜々佳 (Morita Ririka)

■ Role

Solutions Architect

■ 好きな AWS サービス

Amazon QuickSight



AWSハンズオンアカウントの使い方



コンテナが、 どのような課題に役に立つのか



環境は変化し続ける

ビジネスシーンにおいて
市場の変化への迅速な対応が
要求される

そのために、安全かつ迅速に
プロダクトや新機能を市場へ
投入したい

変化に対応するために



アプリケーションへ
フォーカス



要件に応じた
インフラストラクチャの
管理



迅速なスケーリング



セキュリティの確保

コンテナを選択する理由

リスクの低減



様々な環境において
均一なセキュリティを維持

運用の効率性



「差別化に繋がらない重労働」を排除

スピード



一貫性のある環境で開発を加速

俊敏性



自動化により、テストや反復作業の
スピードと容易さを向上

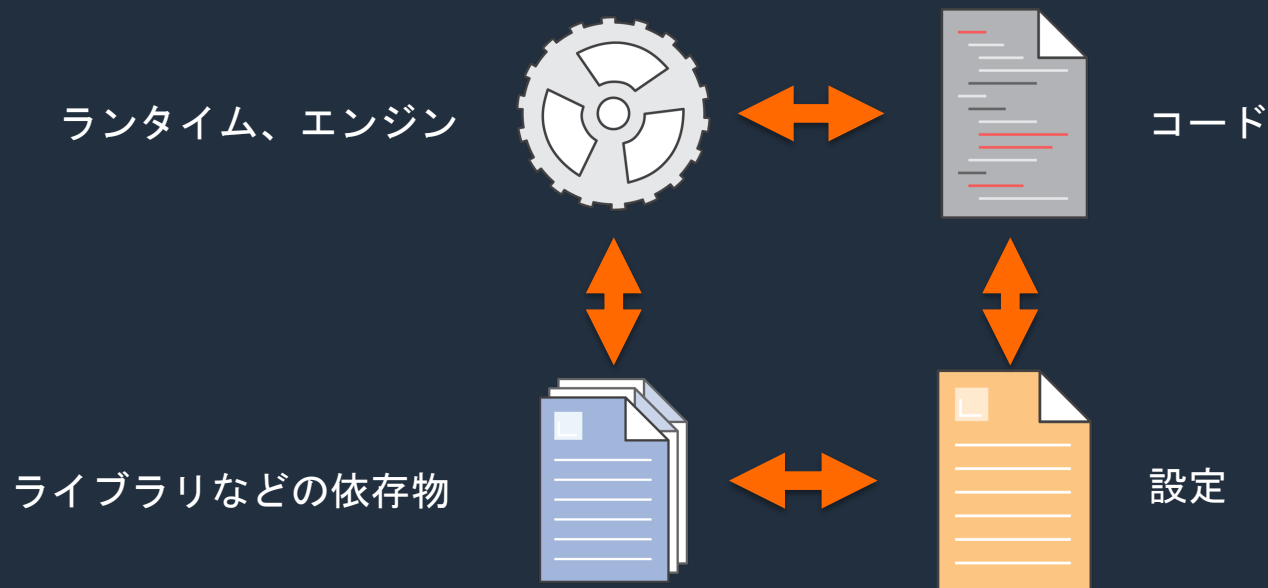
よくある課題

- 環境間の差分でバージョンアップ作業に時間がかかる
- 需要の増加に対応できない

よくある課題

- 環境間の差分でバージョンアップ作業に時間がかかる
- 需要の増加に対応できない

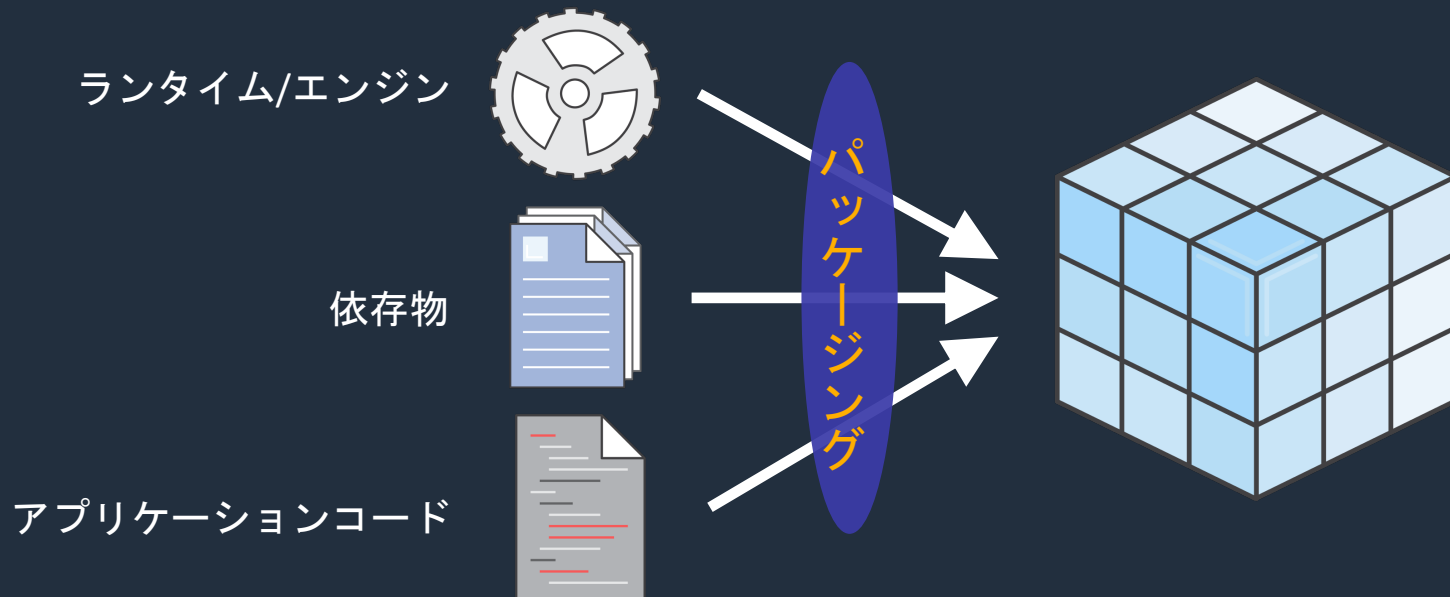
アプリケーションを構成するコンポーネント



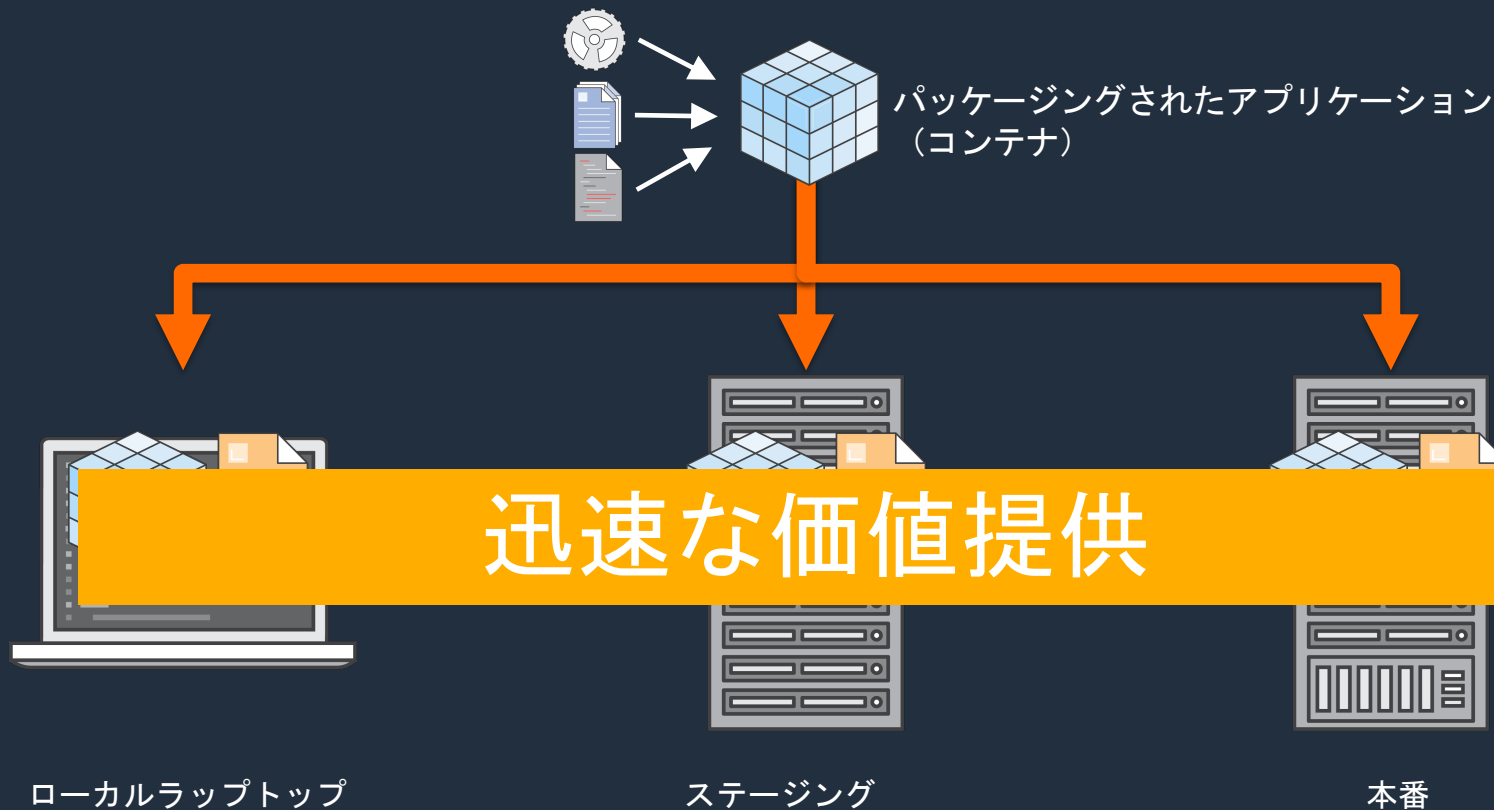
ローカルでは動いたけど、本番で動かない？



「コンテナ」 という解決策



「コンテナ」という解決策

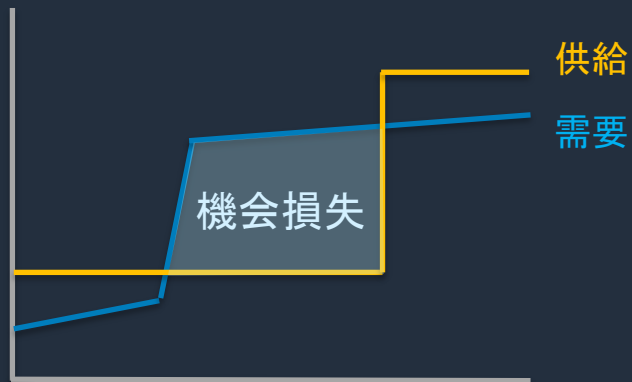


よくある課題

- 環境間の差分でバージョンアップ作業に時間がかかる
- 需要の増加に対応できない

需要の対応が難しい

- 予測不可能な需要の急増
- 機会損失で、お客様の満足度低下



コンテナの迅速性

- 仮想マシンを新たに立ち上げる時と比べて、コンテナの起動は高速
- 需要に応じた柔軟で高速なスケール調整が可能



コストを最適化
需要に対応しやすい

コンテナ化

リソースの供給が高速

Dockerとは

◆ アプリケーションをすばやく構築、テスト、デプロイする為のプラットフォーム

- ✓ Docker 社 (旧dotCloud社) が開発、2013年にオープンソース化
- ✓ 現在は OCI (Open Container Initiative) で管理・標準化

◆ Docker の革新

- ✓ OS仮想化ではなく、アプリケーションをパッケージングし、ツールのエコシステムを通じて、スムーズな開発者エクスペリエンスを提供する分野にある

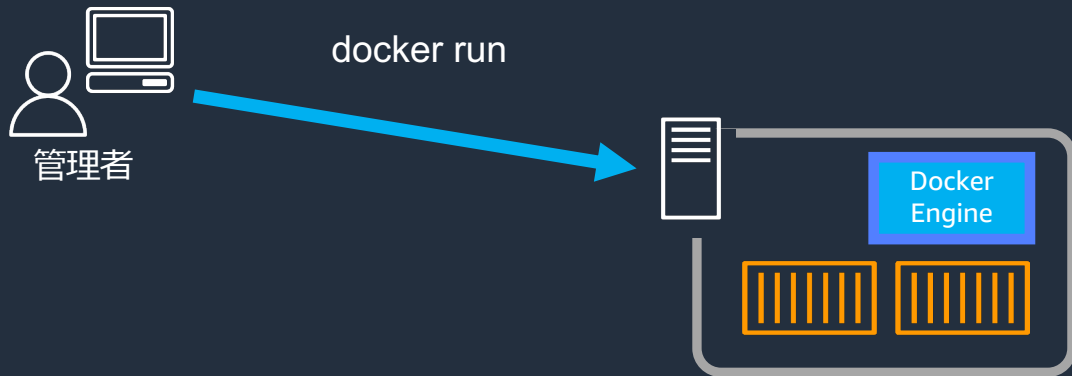
◆ Docker のメリット

- ✓ ソフトウェア出荷の数と速度を向上
- ✓ オペレーションの標準化
- ✓ シームレスに移動
- ✓ コスト削減



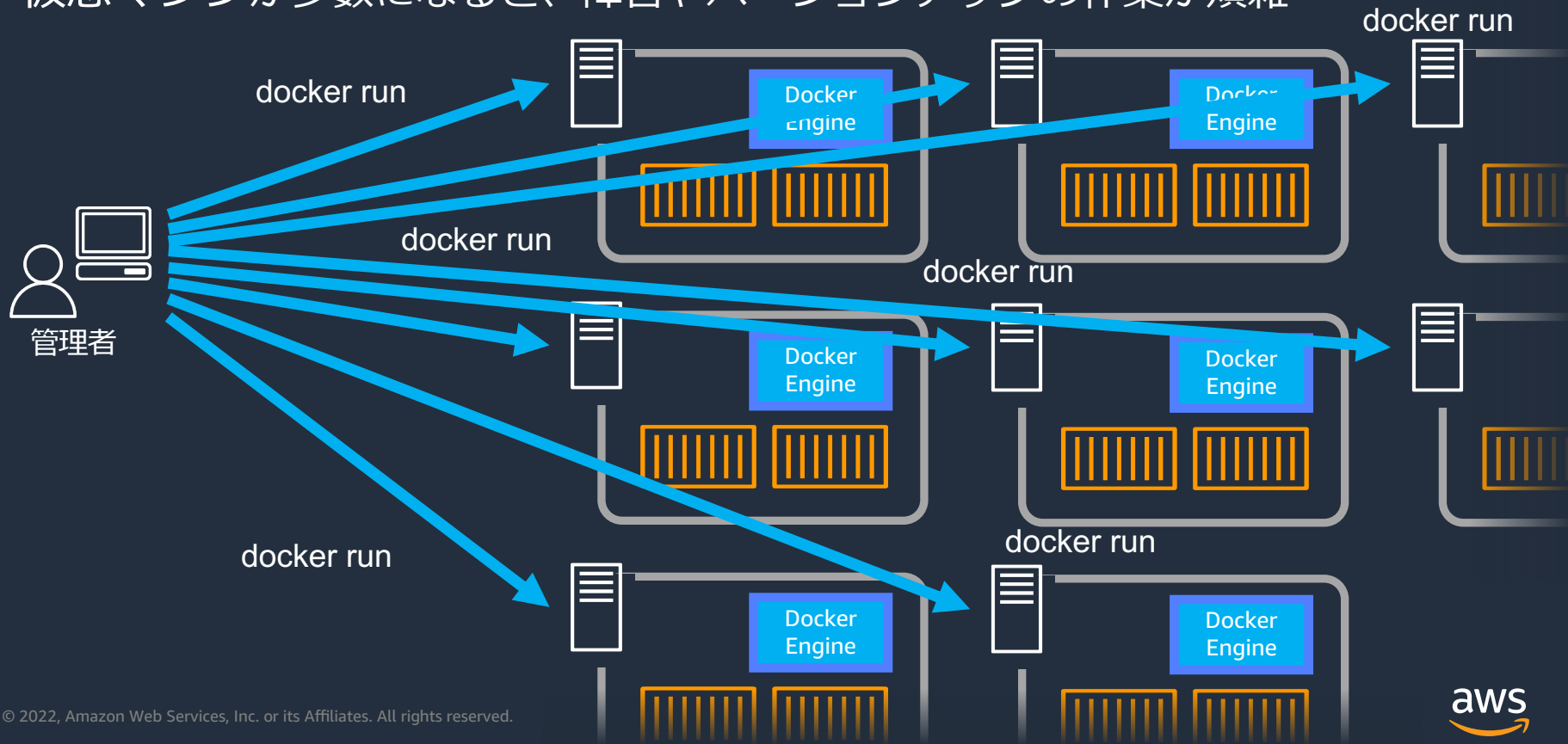
仮想マシンで手動でコンテナを動かす

単体の仮想マシン上で docker run コマンドを実行する



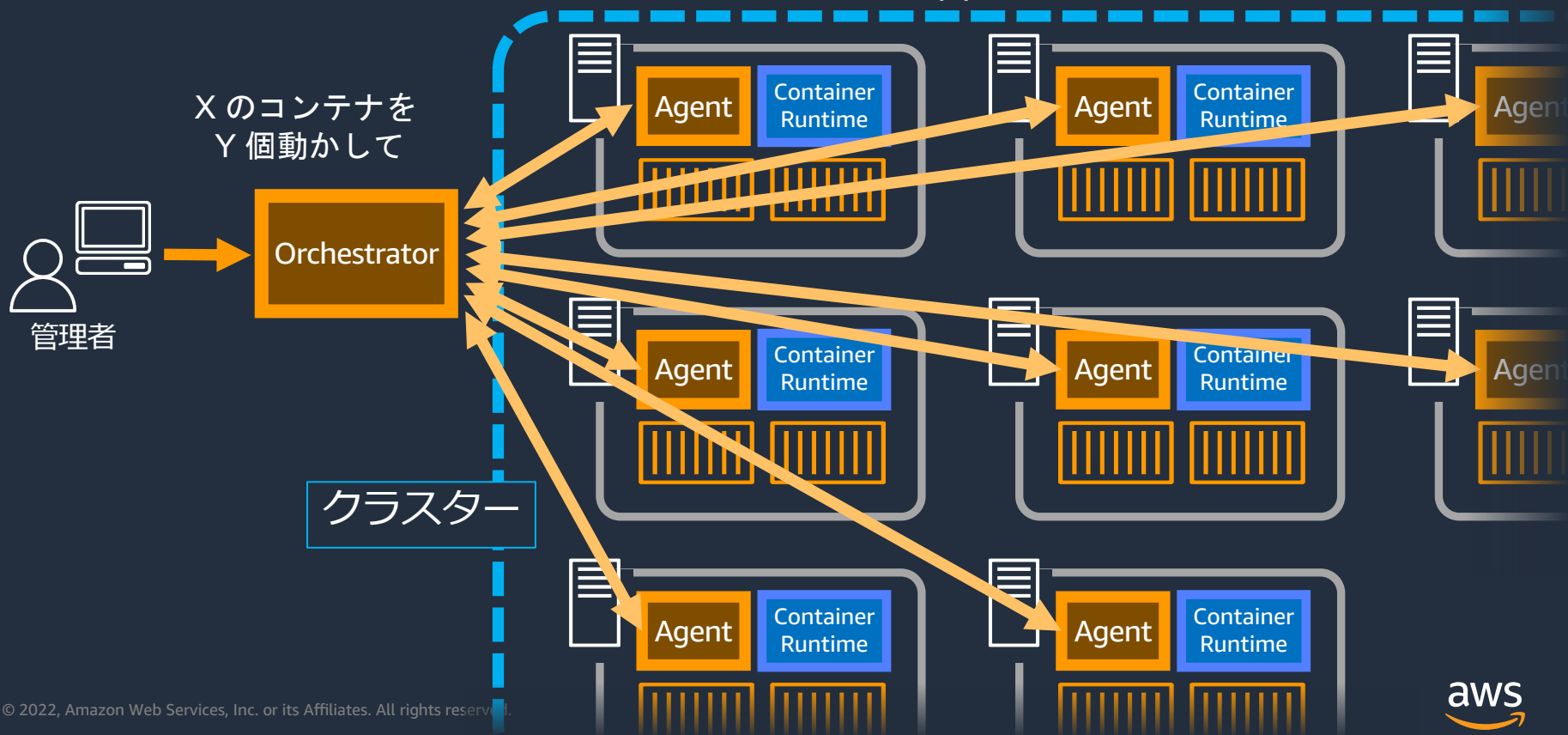
手動で管理する課題

仮想マシンが多数になると、障害やバージョンアップの作業が煩雑



コンテナオーケストレーターの目的

コンテナオーケストレーションによるクラスター管理



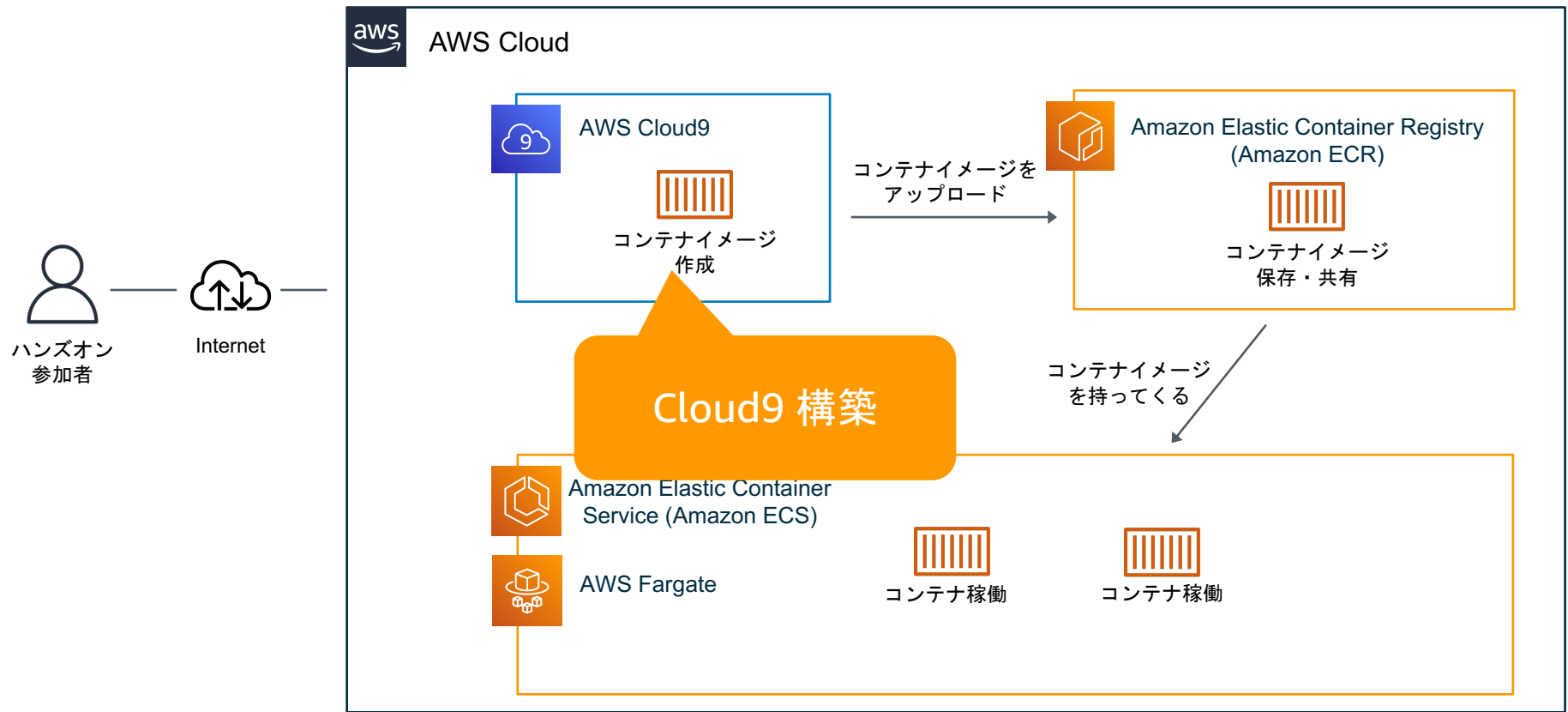
Amazon Elastic Container Service (Amazon ECS)



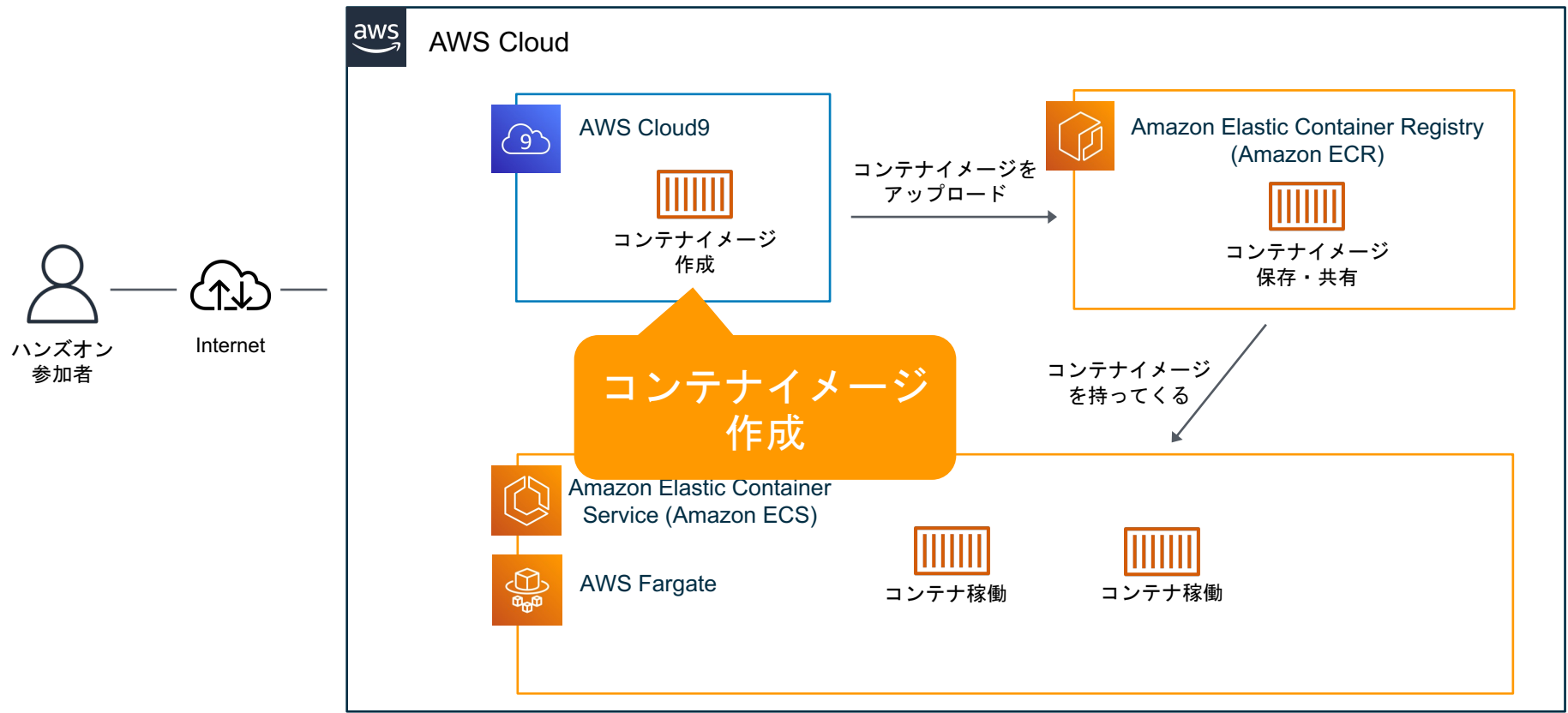
パワフル & シンプル

- クラウドでコンテナを本番環境利用するための
オーケストレーター
- 他の AWS サービスと高度に連携しコンテナを大規模に実行
- フルマネージドなコントロールプレーン
- 多様なワークロードをサポートする「タスク」「サービス」
というシンプルなリソース表現

[初級編] ハンズオンで実施する内容



ハンズオンで実施する内容



コンテナイメージを作成する流れ



AWS Cloud



AWS Cloud9



1. Dockerfile 作成
どのような
コンテナイメージにするのか
設定を入れる



2. docker build コマンドを実行



3. コンテナイメージが
作成される

Dockerfile について

```
FROM ubuntu:18.04
```

```
# Install dependencies
```

```
RUN apt-get update && ¥  
apt-get -y install apache2
```

```
# Install apache and write hello world message
```

```
RUN echo 'Hello World!' > /var/www/html/index.html
```

```
# Configure apache
```

```
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && ¥  
echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && ¥  
echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && ¥  
echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && ¥  
chmod 755 /root/run_apache.sh
```

```
EXPOSE 80
```

```
CMD /root/run_apache.sh
```

・コンテナイメージを作り上げるための命令群

・上から順番に、書かれている命令が実行される

Dockerfile について

FROM ubuntu:18.04

基とするベースイメージを指定
Docker Hub で公開されている
Ubuntu イメージを指定

```
# Install dependencies
RUN apt-get update && \
    apt-get -y install apache2
```

```
# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html
```

```
# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
    echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
    echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh
```

Web サーバーの動作に必要な設定

- ・ Apache HTTP Server のインストール
- ・ index.html ファイルの作成
- ・ 起動のためのシェルスクリプト作成

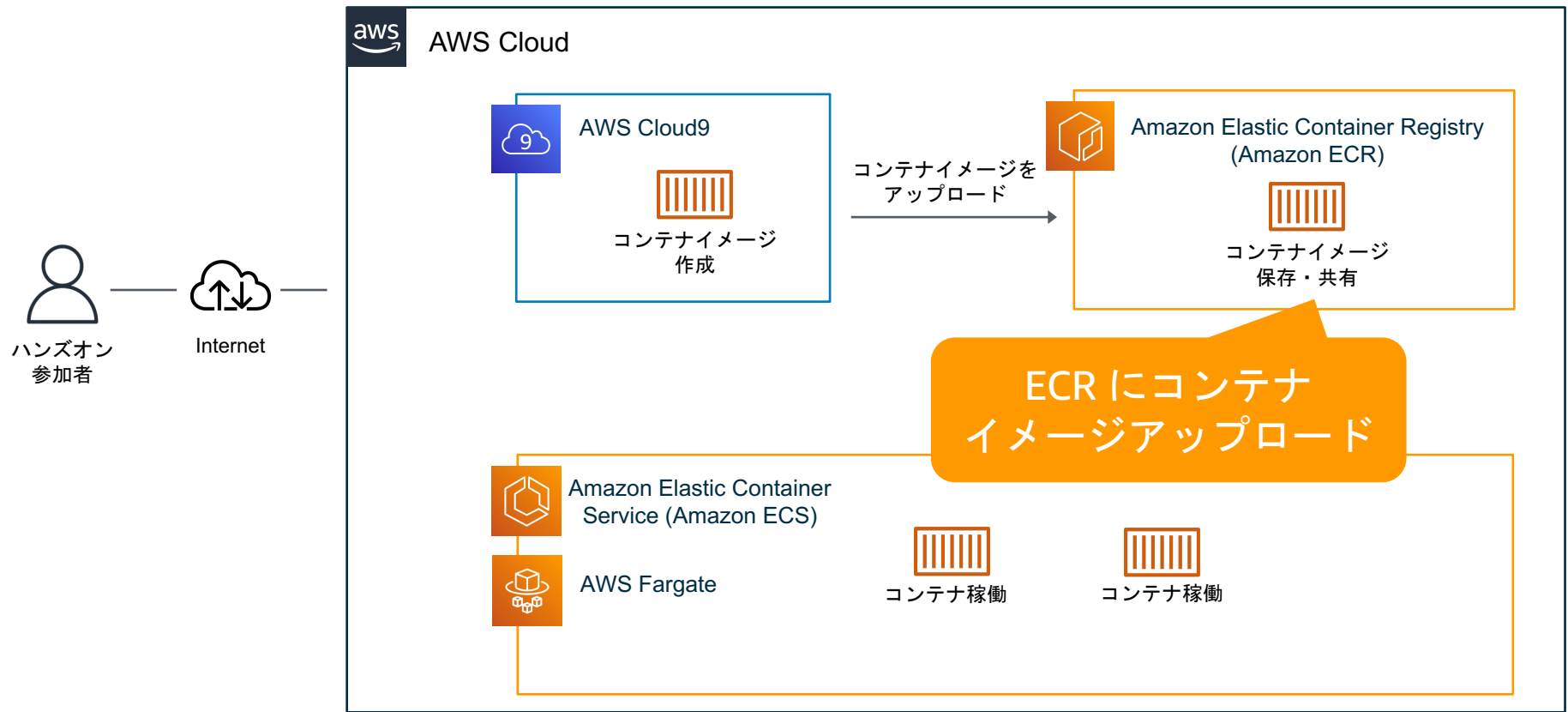
EXPOSE 80

CMD /root/run_apache.sh

コンテナ実行時に動かす
コマンドを指定
Webサーバーを起動するシェルスクリプトを指定

コンテナが利用するネットワー
クポートを指定
ポート 80、HTTP 通信を行う

ハンズオンで実施する内容



Amazon ECRとは

コンテナイメージを保存・管理するマネージドサービス

Docker CLI コマンドを利用することが可能

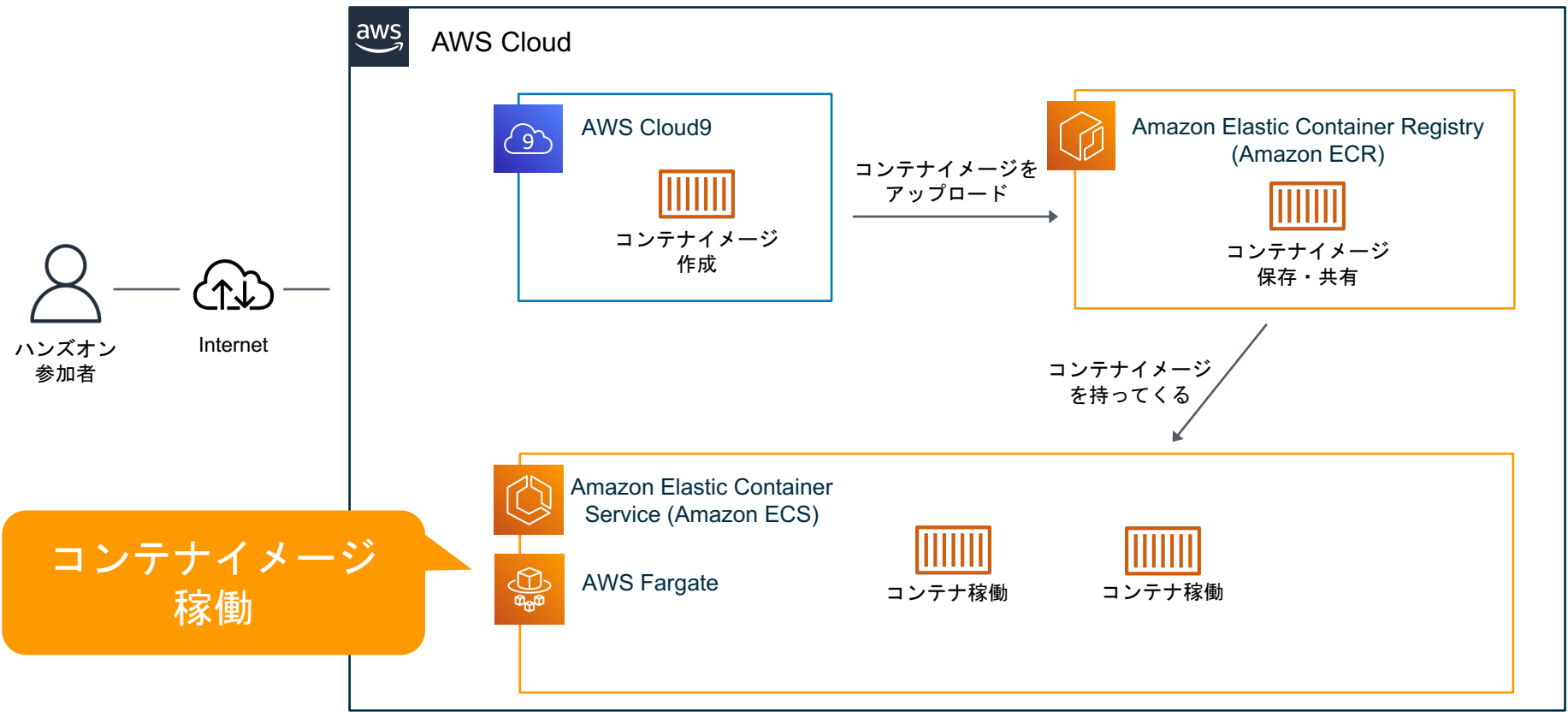
適切なアクセス権を制御

サポートする形式

- Docker イメージ(v1,v2)
- Open Container Initiative (OCI) イメージ
- OCI 互換 artifacts



ハンズオンで実施する内容



扱うサービスの概要



コンテナオーケストレーター

- ・ 何のコンテナをどう動かすか命令を出す

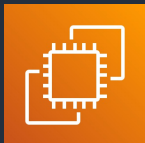
コンテナの実行環境

- ・ コンテナそのものが稼働する環境

Amazon ECS におけるコンテナ実行環境

ECS on EC2

(コンテナを仮想サーバー上で動作)



アプリケーションコンテナ

ホストのスケーリング

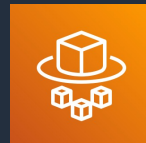
コンテナエージェント設定

ホスト OS / ライブラリ設定

お客様が管理するレイヤー

ECS on Fargate

(コンテナをサーバーレスで動作)



アプリケーションコンテナ

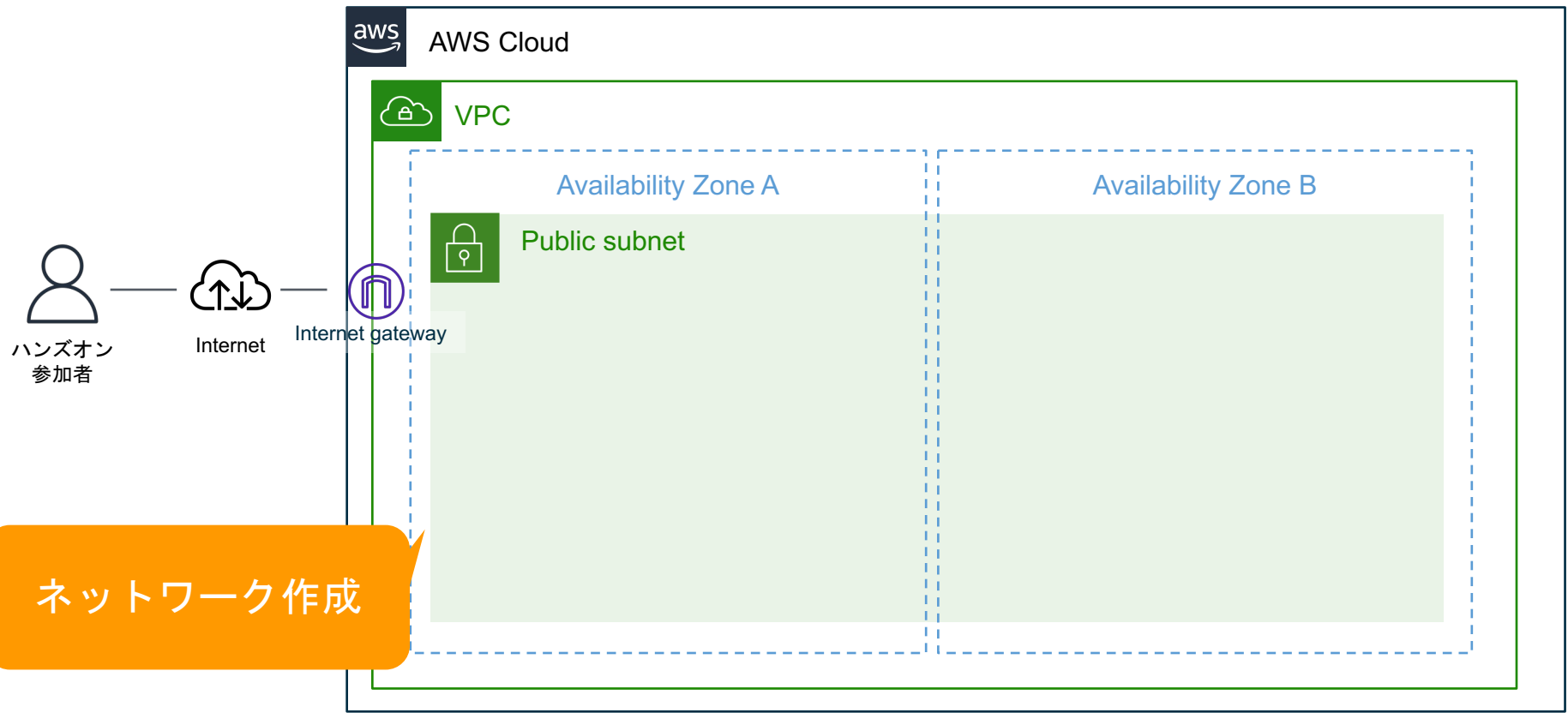
ホストのスケーリング

コンテナエージェント設定

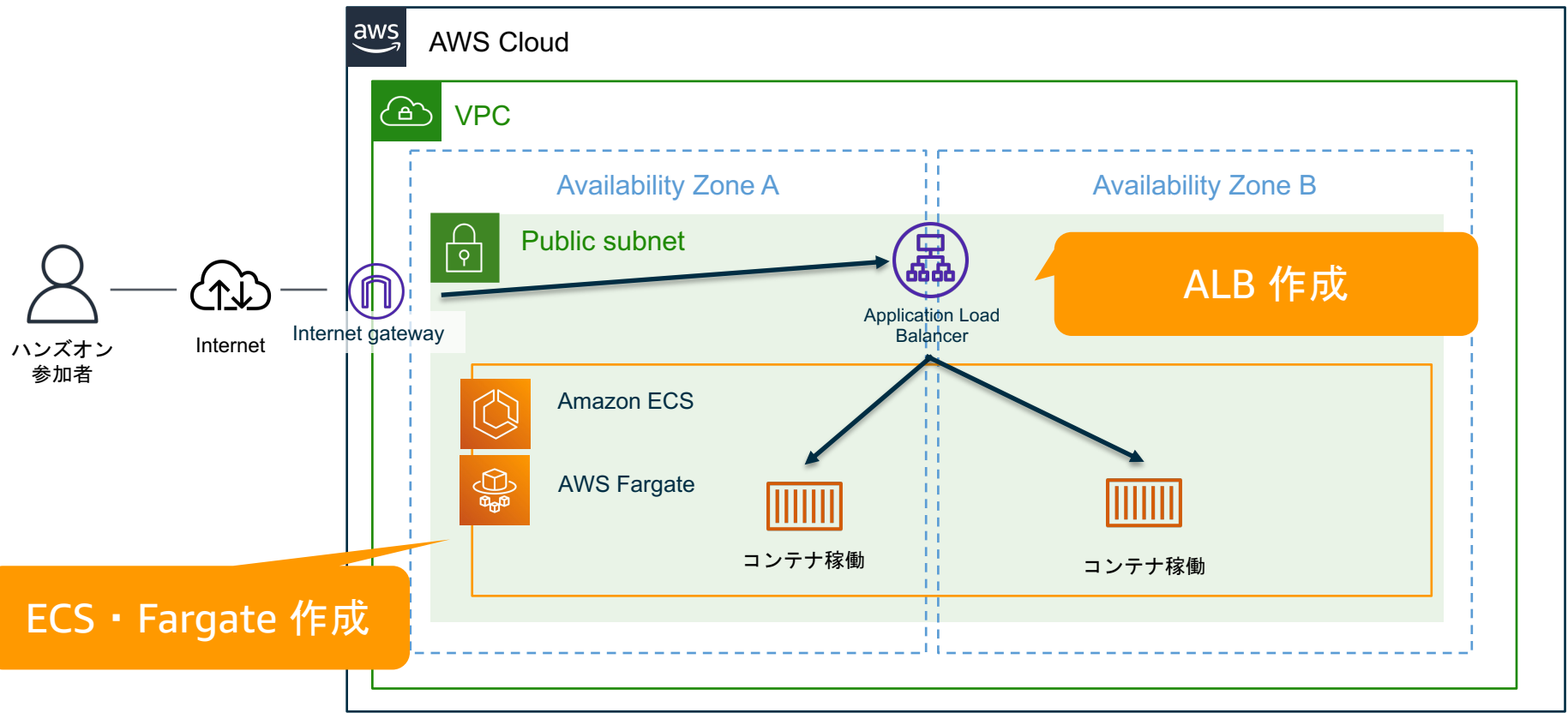
ホスト OS / ライブラリ設定

AWSが提供するレイヤー

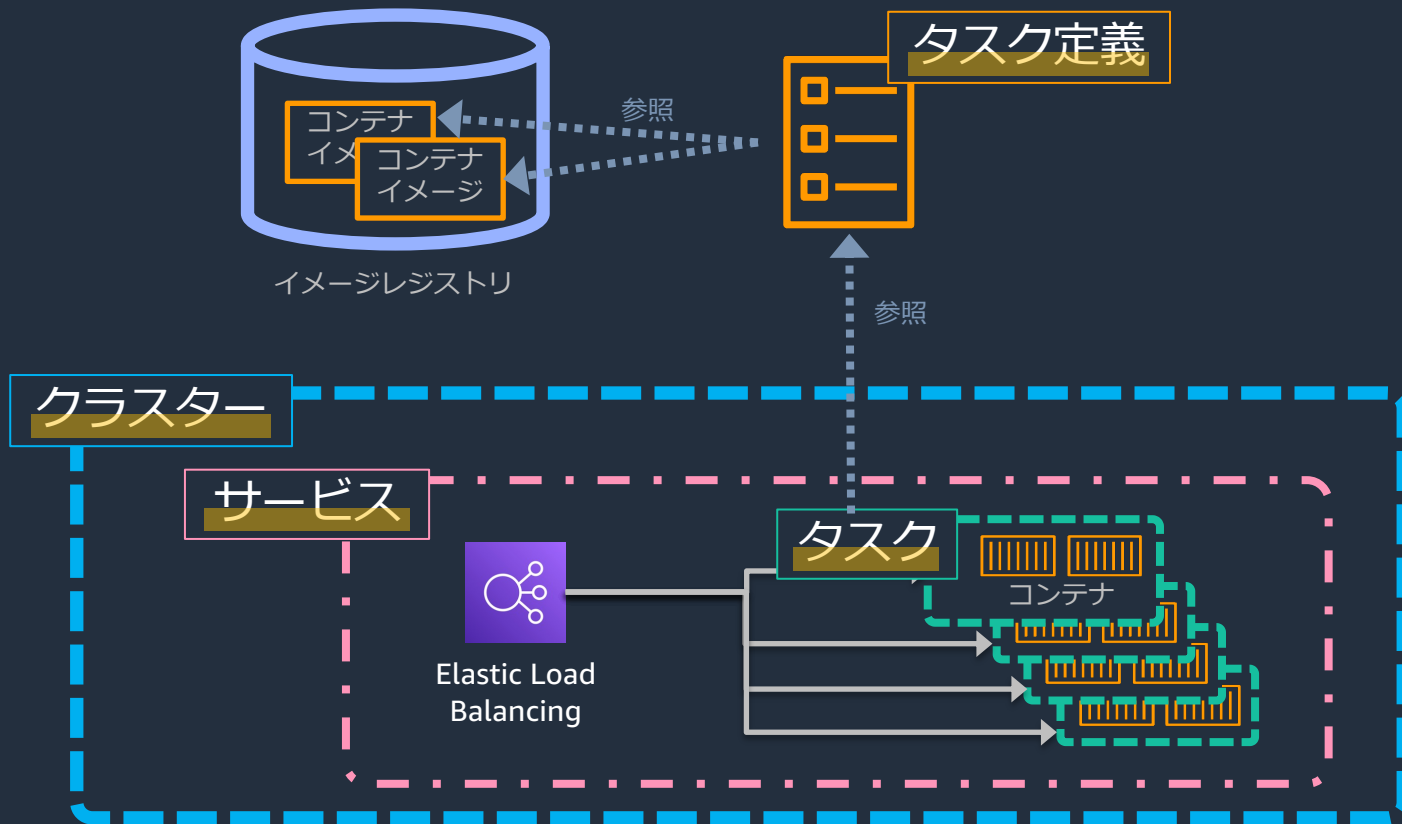
ハンズオンで実施する内容



ハンズオンで実施する内容



Amazon ECS の主要な 4 つの構成要素



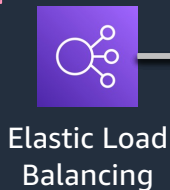
Amazon ECS の主要な 4 つの構成要素

1. 【クラスター】

- コンテナを動かすための論理的なグループ

クラスター

サービス



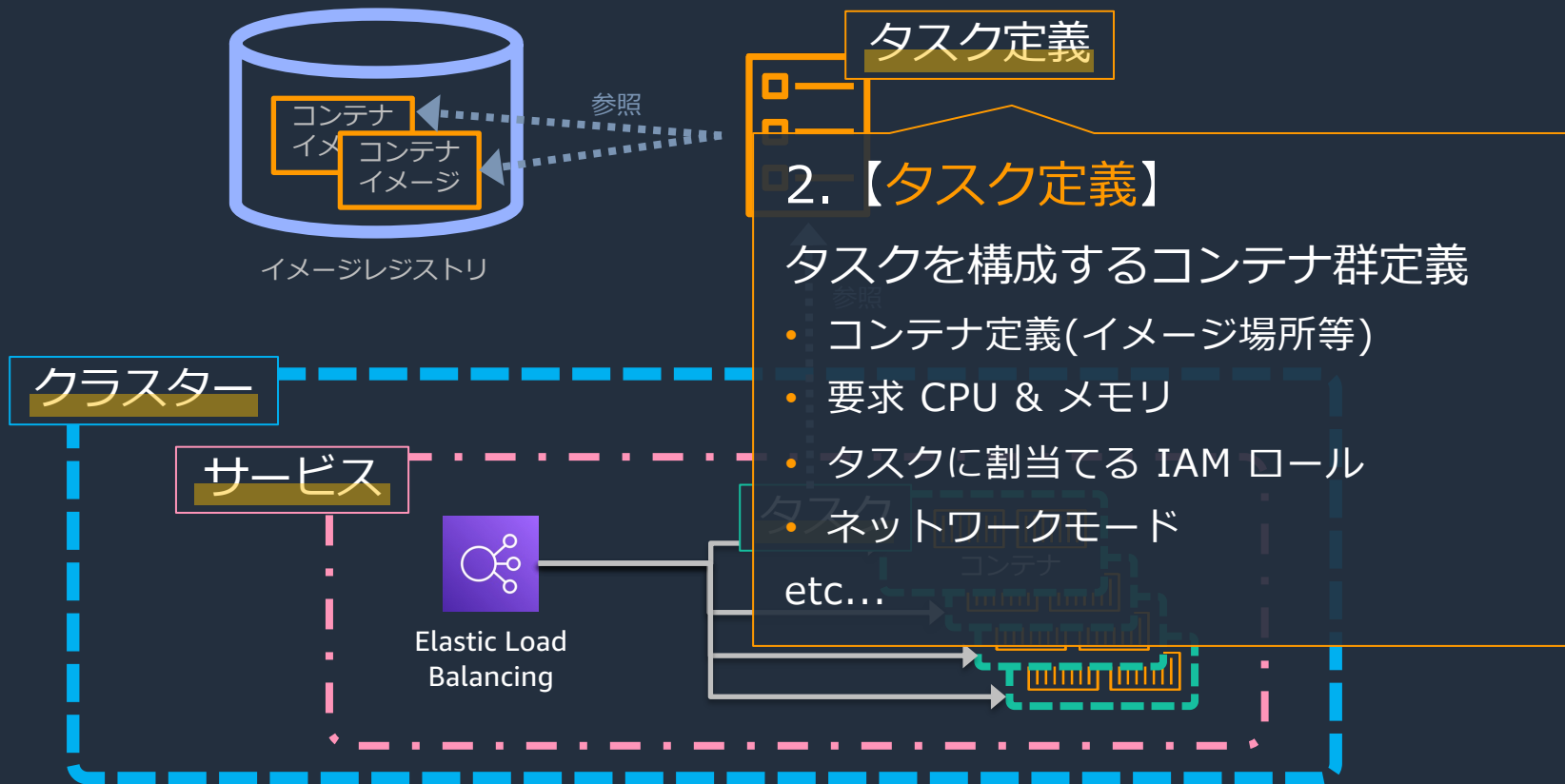
タスク



タスク定義

参照

Amazon ECS の主要な 4 つの構成要素



本シリーズのアジェンダ

5) コンテナオーケストレーションの ECS を作成する 🖐

5-1) VPCの作成

5-2) ECS クラスターの作成

5-3) タスク定義の作成

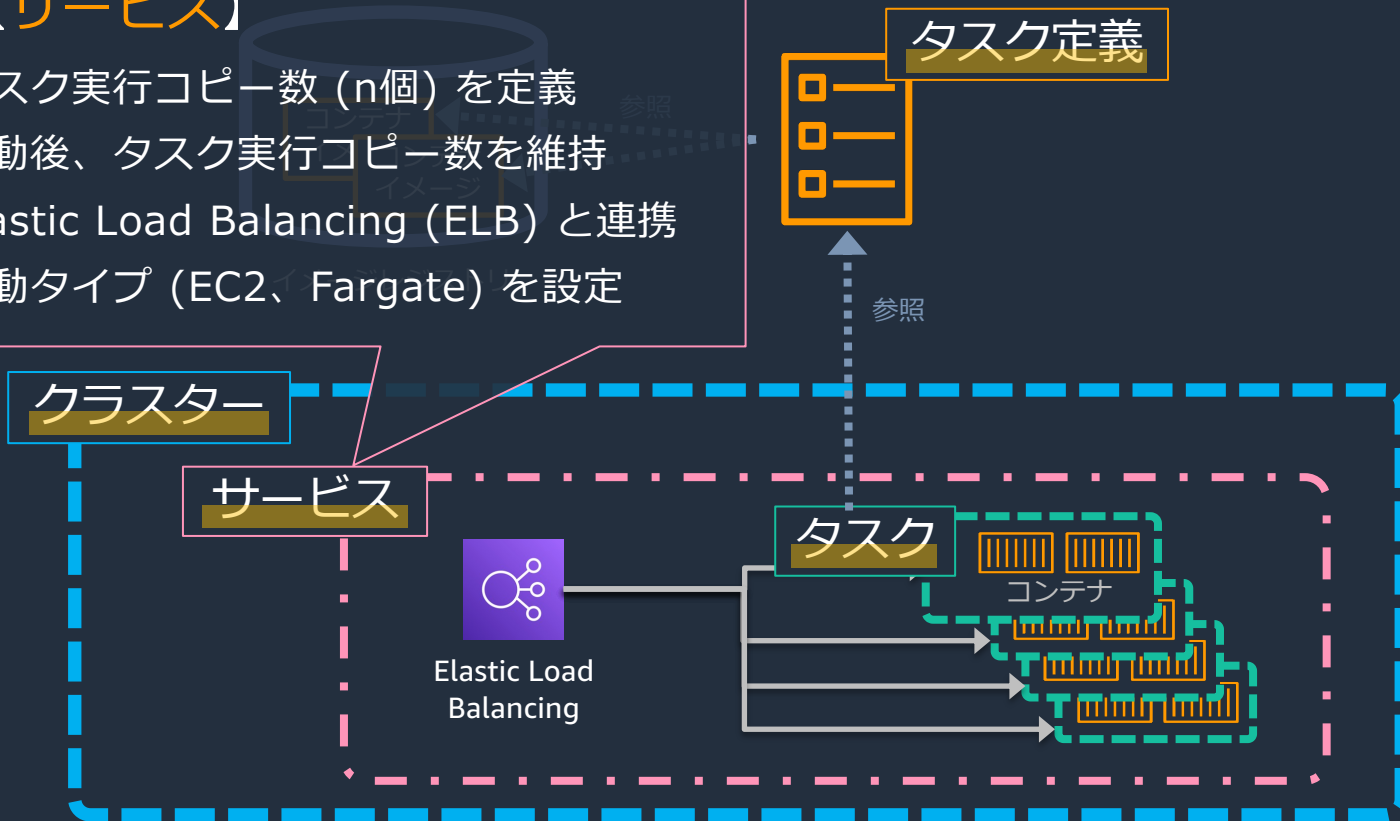
5-4) サービスの作成

5-5) ブラウザからアクセス

Amazon ECS の主要な 4 つの構成要素

3. 【サービス】

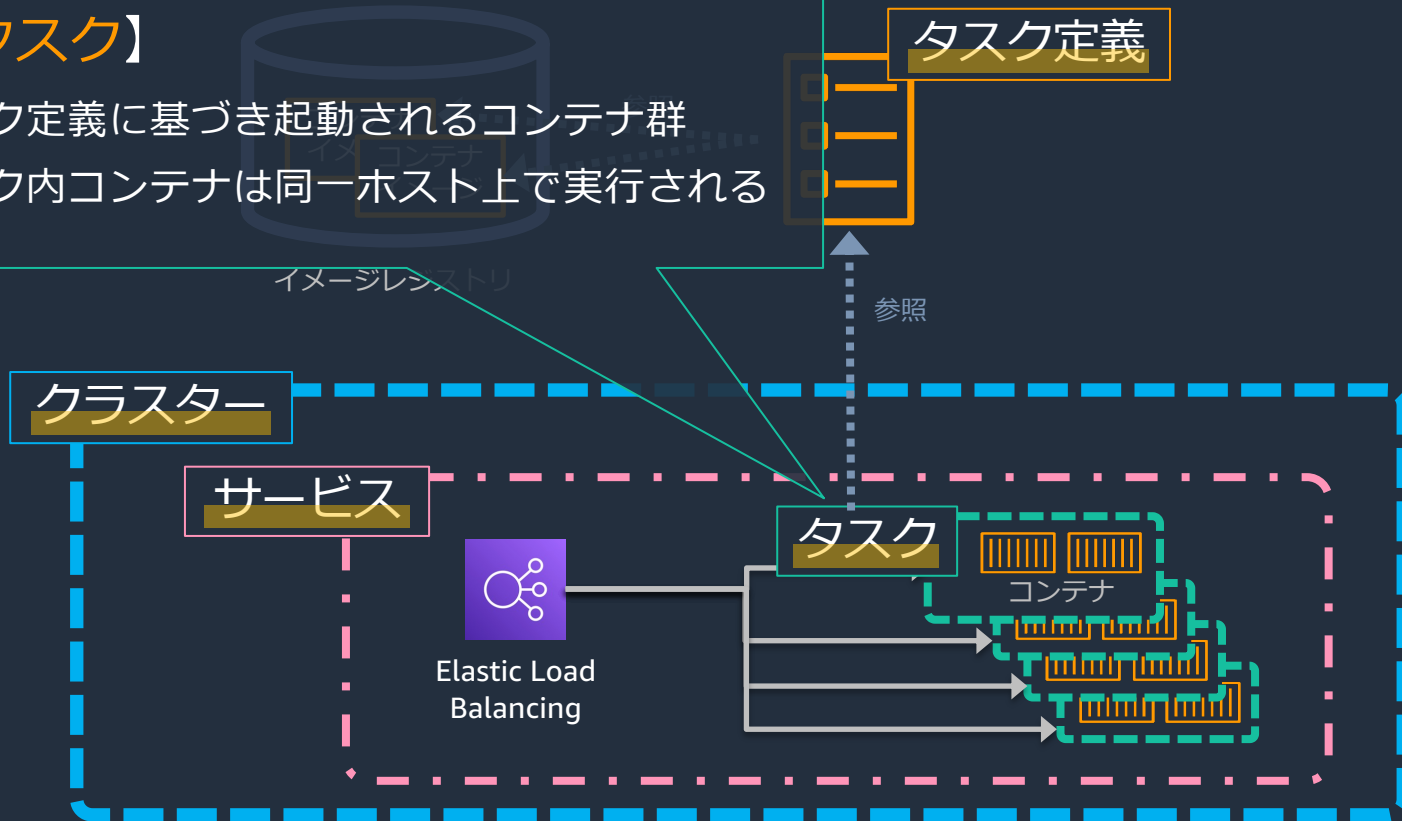
- タスク実行コピー数 (n個) を定義
- 起動後、タスク実行コピー数を維持
- Elastic Load Balancing (ELB) と連携
- 起動タイプ (EC2、Fargate) を設定



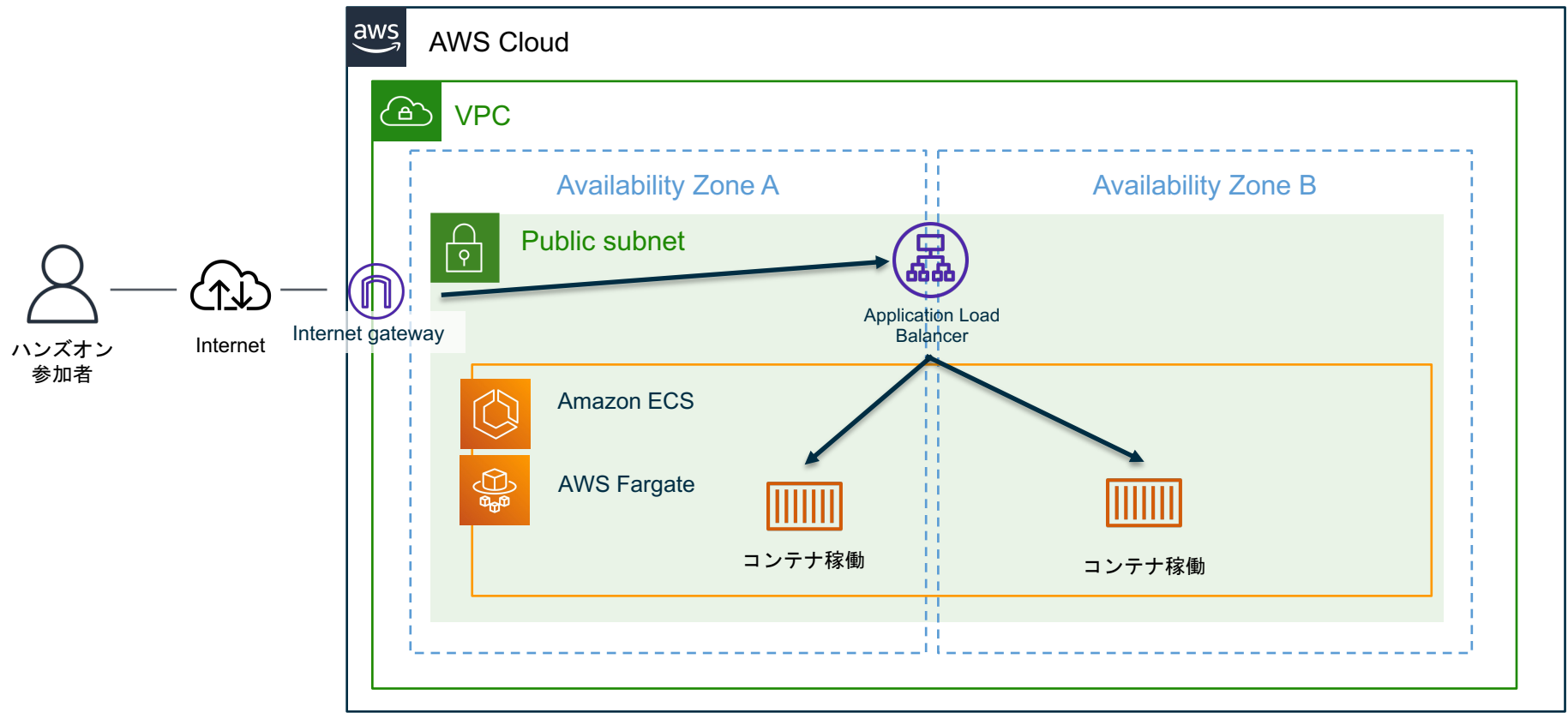
Amazon ECS の主要な 4 つの構成要素

4. 【タスク】

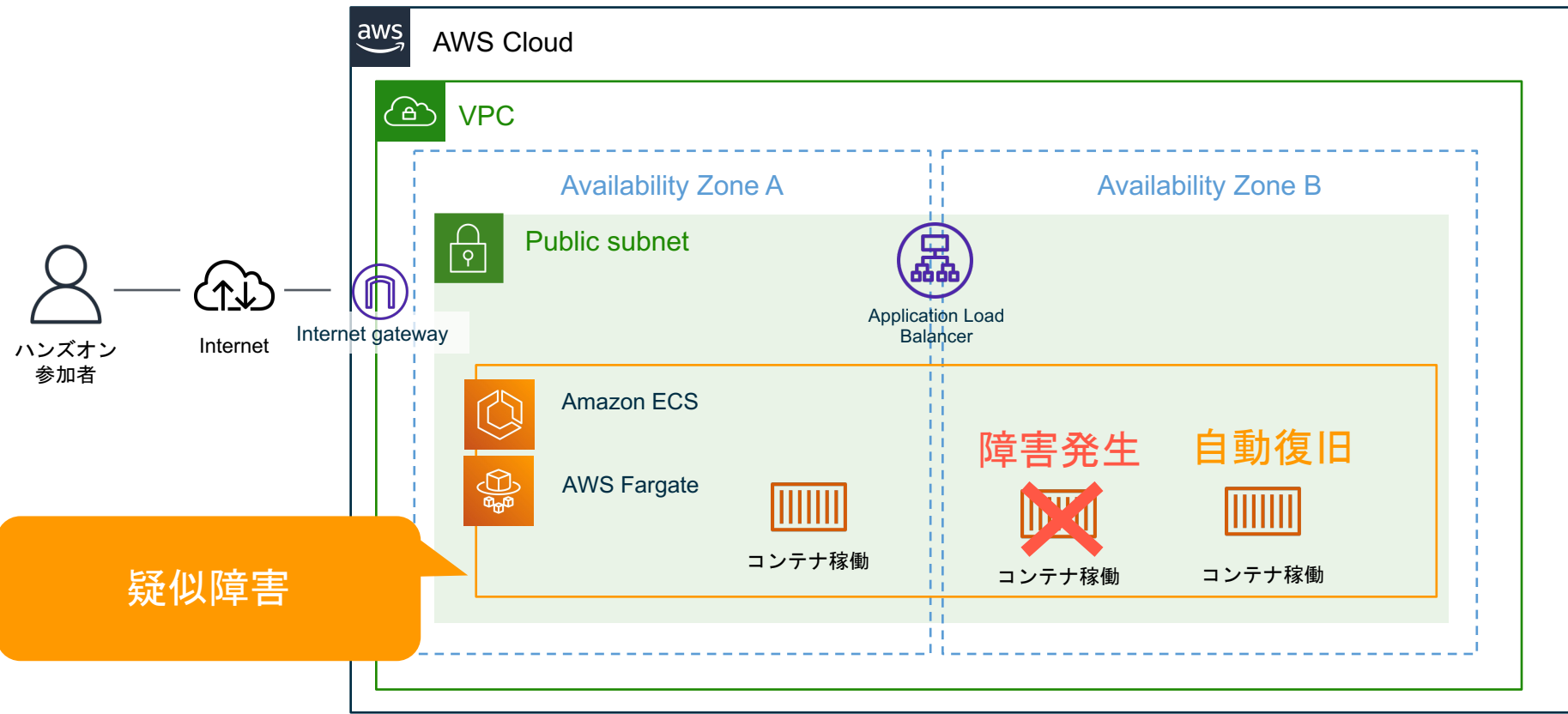
- タスク定義に基づき起動されるコンテナ群
- タスク内コンテナは同一ホスト上で実行される



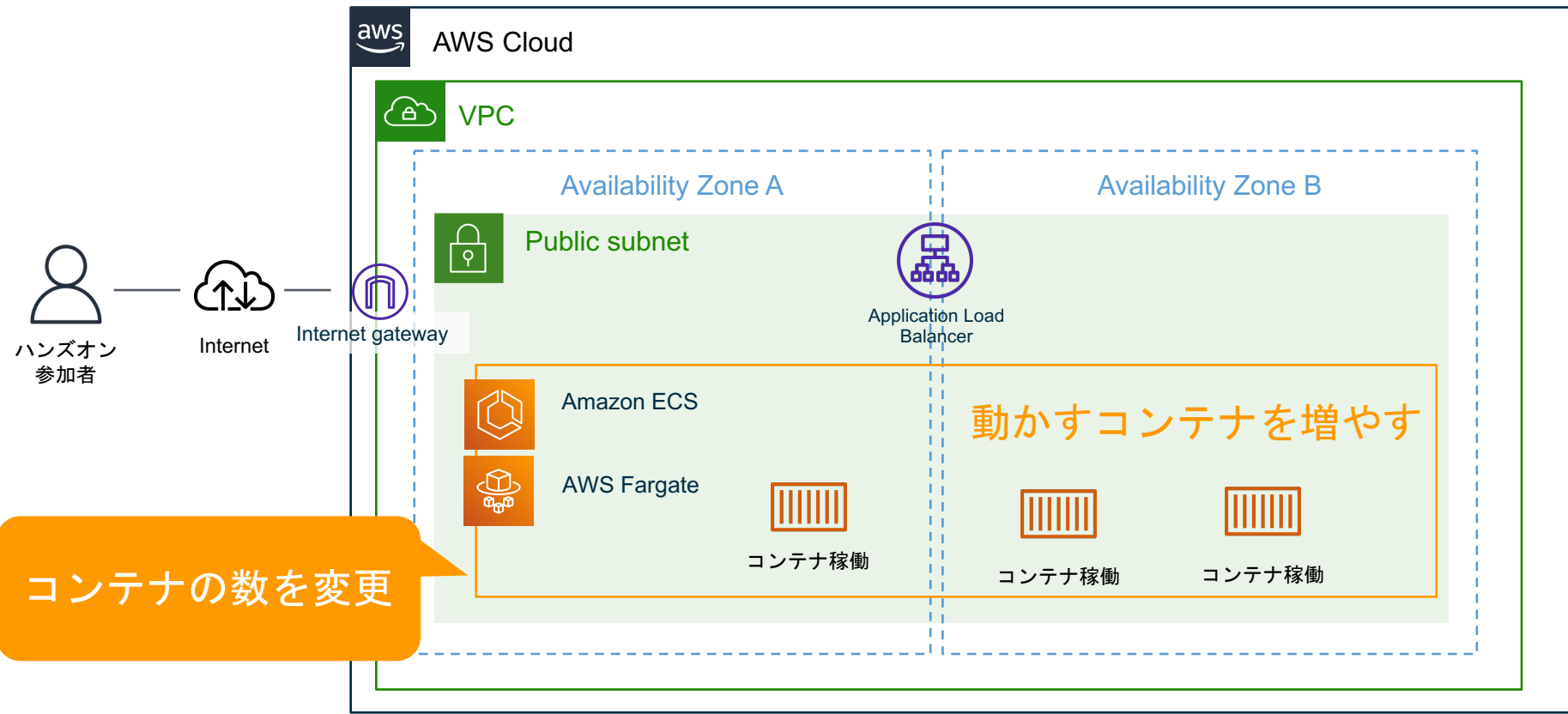
ハンズオンで実施する内容



ハンズオンで実施する内容



ハンズオンで実施する内容



アンケートへのご協力をお願いいたします！

皆さまからのフィードバックを元により良いプログラムを提供したいと考えております

◆ Hands-on に関するアンケート

- ・セッションの満足度
- ・講師の満足度
- ・今後のクラウド利用について
- ・現在の興味範囲
- ・フリーテキスト

(本セッションに関してのご意見や今後実施して欲しいコンテンツなどお気軽にお聞かせください)

受講後、Teams スレッド(基幹技術研修)にアンケートのURLを掲載致します。

https://amazonmr.au1.qualtrics.com/jfe/form/SV_aWOGsVjUUvcKuhg?date=2024年01月31日&title=AWS Hands-on for Beginners

本日の内容や、AWSについて
ご質問などありましたら、お気軽にご連絡ください！



ykkaras@amazon.co.jp

or

aws-jp-sony-sales@amazon.com



**今後もHands-onや勉強会等も企画しておりますので、
ぜひ楽しんでご受講ください 😊**