

## 練習問題

フロントサーバを  
ECSにデプロイ



## 背景

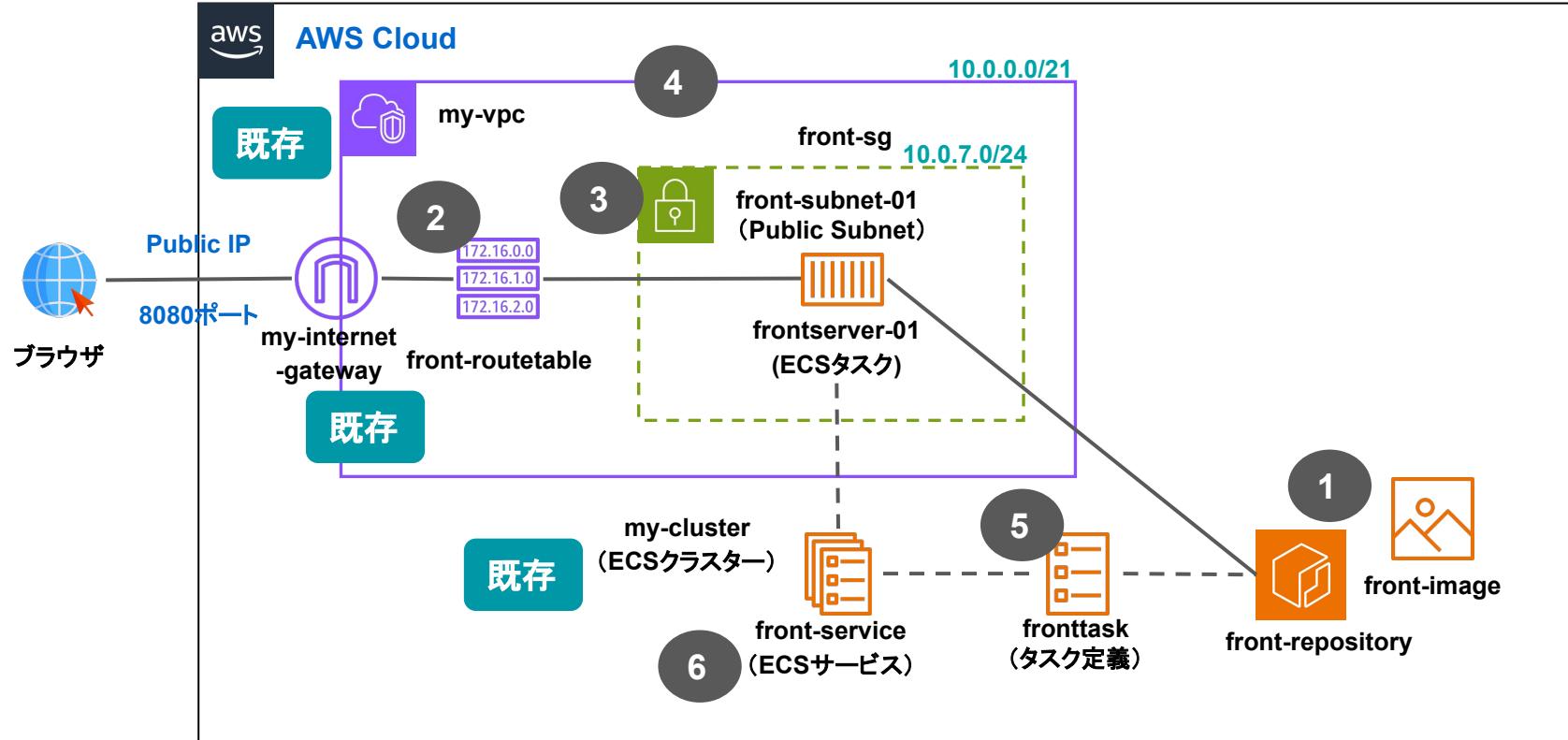
前回までに、APIサーバのECSタスクからRDSのデータベースに接続する方法を扱いました。

最後に、**フロントエンド**の画面部分についても、ECS/Fargate上で起動させ、フロント画面からAPIサーバを呼び出せるようにしたいと思います。

今回も練習問題として、**フロントサーバの ECS起動**を、実際に試してみましょう。

次のページから、課題の要件を記載しています。

# 構成図



# 要件(1/3)

1

## ECRリポジトリにフロントのイメージをプッシュ

- **front-repository**というECRリポジトリを作成する
- 以下のフォルダにある資材と Dockerfile をベースに、ローカルにて **front-image** を作成し、**front-repository** に Push する
  - [https://github.com/CloudTechOrg/course-docker/tree/main/14\\_Frontend](https://github.com/CloudTechOrg/course-docker/tree/main/14_Frontend)
- 必要に応じて、ローカルから使用する IAM ユーザ (ポリシー) を変更する必要がある
- config.js にある **baseURL** を、**API サーバの DNS 名に変更** する必要がある

```
const apiConfig = {  
  baseURL: 'http://<DNS名>'  
};
```

# 要件(2/3)

2

## ルートテーブルの作成

- フロントサーバに割り当てるルートテーブルとして、**front-routetable**を作成する
- front-routetable**には、すでに作成済みの **my-internet-gateway**を関連付けする

3

## サブネットの作成

- フロントサーバに配置するためのサブネットとして、**front-subnet-01**を作成する
- ルートテーブルとして、**front-routetable**を関連付けする
- CIDRブロックは、**10.0.7.0/24**とする

4

## セキュリティグループの作成

- フロントサーバのセキュリティグループとして、**front-sg**を作成する
- インバウンドルールとして、**すべての HTTP 通信を許可** する

# 要件(3/3)

5

## タスク定義の作成

- 起動タイプは **AWS Fargate**、オペレーティングシステムは **Linux/X86\_64**を指定する
- タスク定義ファミリ名は **front-task** とする
- コンテナの設定は下記とする(用意されたコンテナイメージを使用する)
  - イメージURL:**front-repositoryのURI**
  - コンテナポート: **80**
  - プロトコル: **TCP**
  - アプリケーションプロトコル: **HTTP**

6

## タスク定義の作成

- 既存のmy-clusterに、サービスとして **front-service**を指定する
- タスク定義として、**front-task**を指定する

## 動作確認

1. <http://<ECSタスクのパブリックIP>>にて、Webアプリが機動すること
2. **API Test**ボタンが正常に動作すること

API Response: API接続テストが成功しました

API Test

Database Test

3. **Database Test**ボタンが正常に動作すること

DB Response: データベース接続テストが成功しました (Reservationsの件数 : 1)

API Test

Database Test

# 模範解答について

まずは、ご自身の手で進めてみましょう！

どうしてもわからない場合、下記に模範解答を用意しているので参考にしてみてください

[https://github.com/CloudTechOrg/course-docker/blob/main/14\\_Frontend/ModelAnswers.md](https://github.com/CloudTechOrg/course-docker/blob/main/14_Frontend/ModelAnswers.md)

# リソースの削除

練習問題が終わったら、不要な料金が発生しないように、作成したAWSリソースを削除しておきましょう。

削除手順は下記にまとめてあります。

[https://github.com/CloudTechOrg/course-docker/blob/main/14\\_Frontend/DeleteManual.md](https://github.com/CloudTechOrg/course-docker/blob/main/14_Frontend/DeleteManual.md)