DevOps Interview – RoR App Deployment on AWS (Fargate) - Documentation

This project deploys a Ruby on Rails application on AWS using Fargate (ECS), RDS, and S3. Infrastructure is provisioned via Terraform.

Prerequisites

- AWS Account

- IAM user with admin or necessary permissions

- AWS CLI configured (`aws configure`)

- Terraform installed

- Docker installed (for building and pushing image to ECR)

- GitHub account (for forking this repo)

**Step-by-Step Deployment.**

**1. Clone & Navigate**

git clone https://github.com/YOUR\_USERNAME/DevOps-Interview-ROR-App.git

cd DevOps-Interview-ROR-App

**2. Build & Push Docker Image.**

**Authenticate Docker to AWS ECR**

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin [account\_id].dkr.ecr.us-east-1.amazonaws.com

**Build and push**

docker build -t rails-app .

docker tag rails-app:latest [account\_id].dkr.ecr.us-east-1.amazonaws.com/rails-app

docker push [account\_id].dkr.ecr.us-east-1.amazonaws.com/rails-app

Note: Account ID will be present on AWS Account.

**3. Terraform Setup.**

**Commands for the Terraform .**

terraform init

terraform plan

terraform validate

terraform apply

**This will provision:**

* VPC (public/private subnets)
* Internet/NAT Gateway
* ECS Cluster and Service (Fargate)
* RDS PostgreSQL
* S3 Bucket
* IAM Roles
* Application Load Balancer (ALB)

**4. ECS Task Definition & Service.**

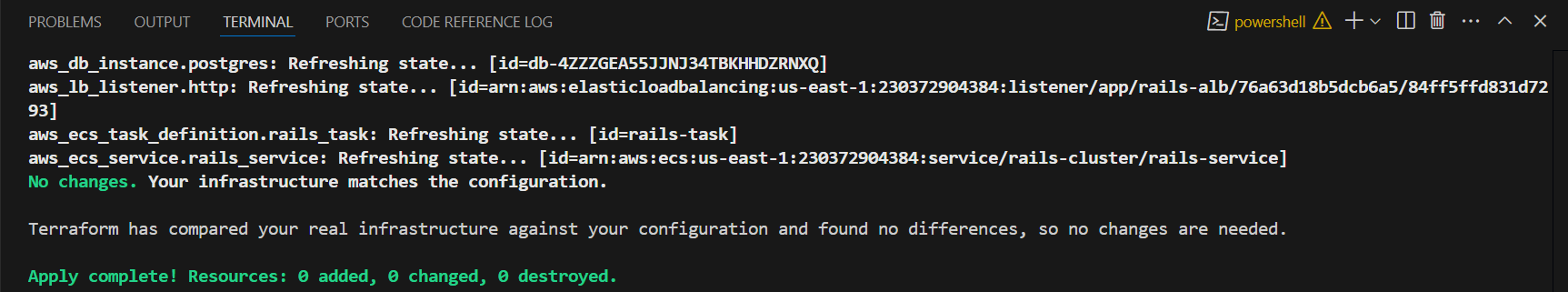
* Update ECS Task Definition with ECR image URI
* Ensure proper executionRoleArn and containerPort
* Run ECS service to deploy the container

## Access the App

* Get the DNS of the Application Load Balancer from the AWS console or Terraform output:

terraform output alb\_dns\_name

* Open it in your browser:  
  http://<alb\_dns\_name>



The architecture diagram I have attached below for your reference.

