

Automated CI/CD Pipeline on AWS Using Docker, Terraform & ECS

TechTrivandrum illustrative project guide

This document accompanies an educational DevOps architecture example. It is a conceptual project and does not claim a real deployment, tested application, or production-ready infrastructure.

Project goal

Show how a GitHub code commit can trigger an automated AWS CI/CD workflow, build a Docker image, publish it to Amazon ECR, and deploy a containerized application to Amazon ECS Fargate behind an Application Load Balancer.

Illustrative flow

Developer pushes code to GitHub. AWS CodePipeline orchestrates stages. AWS CodeBuild runs checks and builds the image. Amazon ECR stores the image. Amazon ECS Fargate deploys the service. CloudWatch provides monitoring and alarms. Terraform conceptually provisions the AWS infrastructure.

Infrastructure scope

The architecture concept includes a VPC, public subnets for the ALB, private subnets for ECS tasks, IAM roles, ECR, ECS service, task definitions, security groups, Secrets Manager or SSM Parameter Store, and CloudWatch observability.

Security notes

For any real implementation, use least-privilege IAM, private application subnets, HTTPS on the ALB, secure secret storage, image scanning, log retention, and operational alarms. Review every policy, route, and security group before deployment.

Deployment caution

The included sample snippets are educational excerpts only. They are intentionally incomplete and must not be used directly as a production deployment package.