Introduction

Create continuous integration and continuous deployment (CI/CD) pipelines and Amazon ECS clusters for deploying microservices using AWS CDK.

An application developer pushes code to a CodeCommit repository, triggering a pipeline.

CodeBuild then builds and pushes a Docker image to an Amazon ECR repository.

Subsequently, CodePipeline deploys the new image to an existing Fargate service in a non-production Amazon ECS cluster, with Amazon ECS pulling the image from the ECR repository into the Fargate service.

Testing is conducted using a non-production URL.

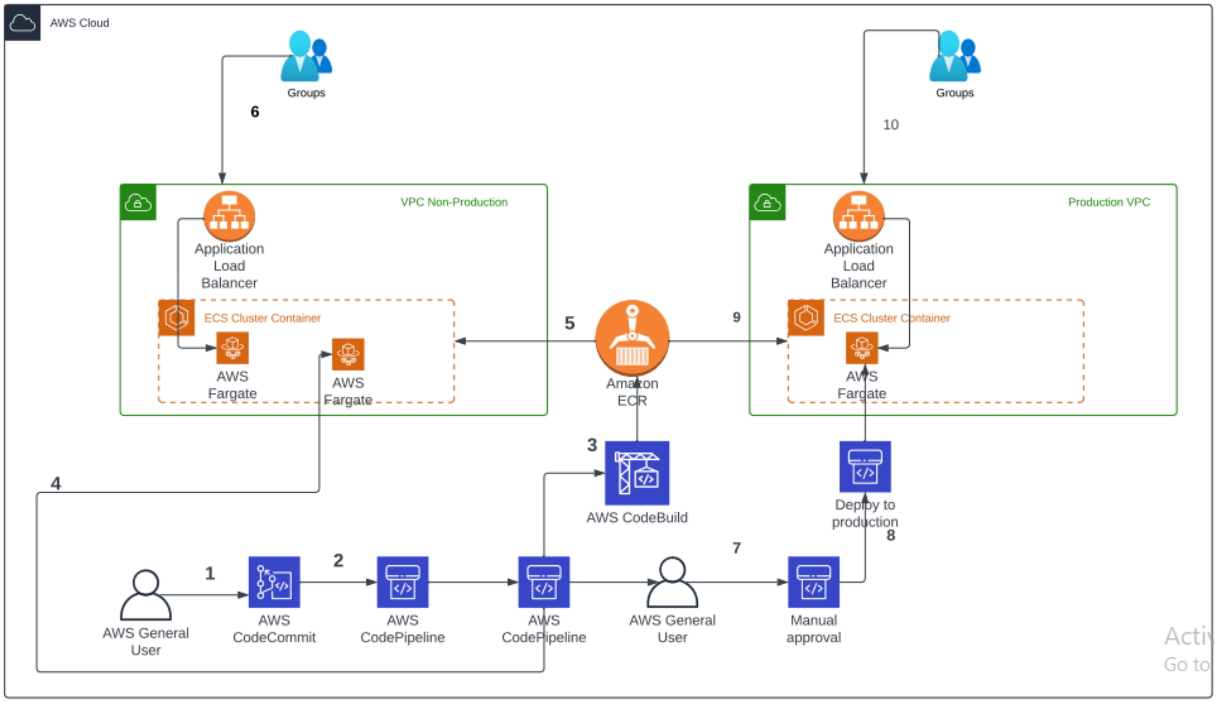
Once testing is completed, the release manager approves the production deployment. CodePipeline deploys the new image to an existing Fargate service in a production Amazon ECS cluster

Amazon ECS pulls the image from the Amazon ECR repository into the production Fargate service.

Production users access your feature by using a production URL.

Technology stack

- AWS CDK
- CodeBuild
- CodeCommit
- CodePipeline
- Amazon ECR
- Amazon ECS
- Amazon VPC



In conclusion, this involves creating CI/CD pipelines and Amazon ECS clusters for deploying microservices using AWS CDK to increase efficiency of the software delivery process.

The process includes pushing code to a CodeCommit repository, triggering a pipeline that builds and pushes a Docker image to an Amazon ECR repository. Subsequently, CodePipeline deploys the new image to non-production and production Fargate services in Amazon ECS clusters. Testing is done using a non-production URL, and once approved, the new image is deployed to the production environment.