Project 3: CI/CD Pipeline with Docker, GitHub Actions & AWS EC2

Introduction

This project implements a CI/CD pipeline for a Node.js application using Docker, GitHub Actions, and AWS EC2. The objective was to automate the build, test, and deployment process, enabling faster delivery and deployment of applications in a cloud environment.

Abstract

The CI/CD workflow is triggered by commits to the main branch in GitHub. GitHub Actions builds a Docker image from the application, pushes it to DockerHub, and deploys the container on an AWS EC2 instance using SSH. The Node.js app runs on port 3000 and is accessible publicly. The pipeline ensures seamless integration and delivery from source code to running container.

Tools Used

Node.js, Docker, DockerHub, GitHub Actions, AWS EC2, SSH

Steps Involved

1. Created a simple Node.js application (server.js, package.json).
2. Wrote a Dockerfile to containerize the application.
3. Configured GitHub Actions workflow (`deploy.yml`) to:
- Build Docker image from source code.
- Push Docker image to DockerHub.
- SSH into AWS EC2 instance and pull the latest image.
- Run the container on port 3000.
4. Verified deployment by accessing app at `http://:3000`.
5. Added local testing steps using `docker build` and `docker run`.

Conclusion

This project validated the power of GitHub Actions in automating CI/CD pipelines integrated with Docker and AWS EC2. It ensured smooth and reliable software delivery while reducing manual intervention in the deployment cycle.