CI/CD Pipeline with GitHub Actions & Docker

Introduction

Continuous Integration and Continuous Deployment (CI/CD) are vital for streamlining software delivery. This project sets up a basic CI/CD pipeline using **GitHub Actions** and **Docker**, which builds, tests, and deploys a simple Python Flask app in response to code changes.

Abstract

This project automates container image creation and deployment using GitHub Actions triggered by commits to the main branch. A lightweight Python Flask application is containerized with Docker, then built and pushed to Docker Hub. The pipeline ensures faster iterations, fewer manual errors, and seamless deployment.

Tools Used

- **GitHub Actions** – for CI/CD automation
- **Docker & DockerHub** – for image building and hosting
- **Python Flask** – web application
- **Local** – deployment environment

Steps Involved in Building the Project

Developed the Flask App (`app.py`)

- Simple web service
- Created `requirements.txt` specifying Flask dependency.

Dockerized the Application

- Using Dockerfile

Created GitHub Actions Workflow

- Workflow: `.github/workflows/docker-ci-cd.yml`

- On every push to `main`:
  - Checks out code
  - Logs into DockerHub
  - Builds and pushes image

Configured GitHub Secrets

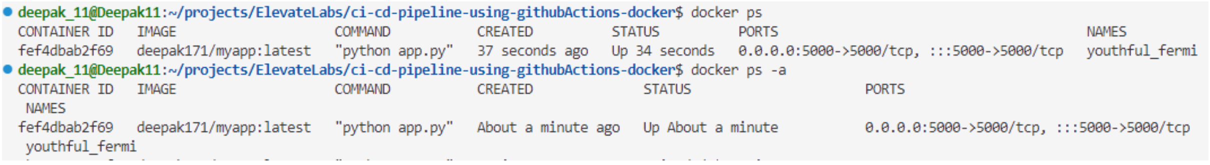
- `DOCKER_USERNAME` and `DOCKER_PASSWORD` for DockerHub authentication.

Pushed Code to GitHub

- Triggered GitHub Action on push event.

Pulled & Ran Docker Image

- docker pull deepak171/myapp
- From local machine



```
● deepak_11@Deepak11:~/projects/ElevateLabs/ci-cd-pipeline-using-githubActions-docker$ docker ps
  CONTAINER ID   IMAGE                  COMMAND           CREATED          STATUS          PORTS                                       NAMES
  fef4dbab2f69   deepak171/myapp:latest   "python app.py"   37 seconds ago   Up 34 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   youthful_fermi
● deepak_11@Deepak11:~/projects/ElevateLabs/ci-cd-pipeline-using-githubActions-docker$ docker ps -a
  CONTAINER ID   IMAGE                  COMMAND           CREATED              STATUS                 PORTS
   NAMES
  fef4dbab2f69   deepak171/myapp:latest   "python app.py"   About a minute ago   Up About a minute      0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
   youthful_fermi
```

- Docker image link →

https://hub.docker.com/repository/docker/deepak171/myapp/tags/latest/sha256-80e469389c8f217121d40f8f5cb205c1fcd3c732ca1c5441c3e787f82e1d2ae5

Conclusion

This project demonstrates an effective CI/CD setup using GitHub Actions and Docker. It streamlines the build and deployment process, enabling quick delivery of containerized applications with minimal manual overhead.