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 CI/CD pipeline using **GitHub Actions** and **Docker**, which builds, tests, and deploys a Python django application.

Abstract

This project automates container image creation and deployment using GitHub Actions triggered by commits to the main branch. A lightweight Python django 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 django App

- configured views.py and urls.py
- configured the application to access
- 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

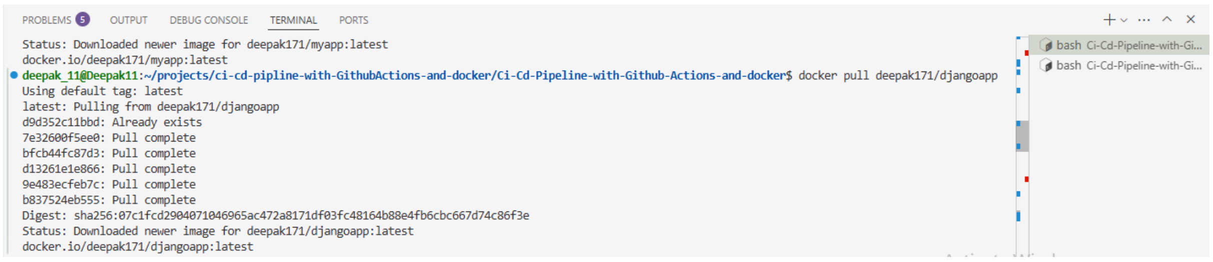
- `DOCKER_USERNAME` and `DOCKER_PASSWORD` for DockerHub authentication.

Pushed Code to GitHub

- Triggered GitHub Action on push event.

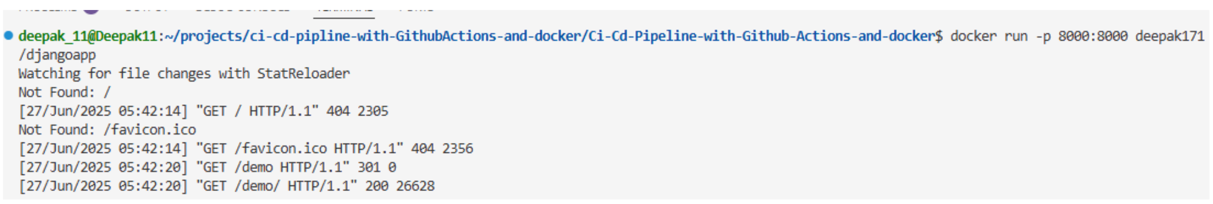
Pulled & Ran Docker Image

- docker pull deepak171/django
- From local machine


PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash Ci-Cd-Pipeline-with-Gi...
bash Ci-Cd-Pipeline-with-Gi...

```
Status: Downloaded newer image for deepak171/myapp:latest
docker.io/deepak171/myapp:latest
deepak_11@Deepak11:~/projects/ci-cd-pipline-with-GithubActions-and-docker/Ci-Cd-Pipeline-with-Github-Actions-and-docker$ docker pull deepak171/djangoapp
Using default tag: latest
latest: Pulling from deepak171/djangoapp
d9d352c11bbd: Already exists
7e32600f5ee0: Pull complete
bfcb44fc87d3: Pull complete
d13261e1e866: Pull complete
9e483ecfeb7c: Pull complete
b837524eb555: Pull complete
Digest: sha256:07c1fcd2904071046965ac472a8171df03fc48164b88e4fb6cbc667d74c86f3e
Status: Downloaded newer image for deepak171/djangoapp:latest
docker.io/deepak171/djangoapp:latest
```

- docker run



```
deepak_11@Deepak11:~/projects/ci-cd-pipline-with-GithubActions-and-docker/Ci-Cd-Pipeline-with-Github-Actions-and-docker$ docker run -p 8000:8000 deepak171
/djangoapp
Watching for file changes with StatReloader
Not Found: /
[27/Jun/2025 05:42:14] "GET / HTTP/1.1" 404 2305
Not Found: /favicon.ico
[27/Jun/2025 05:42:14] "GET /favicon.ico HTTP/1.1" 404 2356
[27/Jun/2025 05:42:20] "GET /demo HTTP/1.1" 301 0
[27/Jun/2025 05:42:20] "GET /demo/ HTTP/1.1" 200 26628
```

- Docker image link →

https://hub.docker.com/repository/docker/deepak171/djangoapp/tags/latest/sha256-07c1fcd2904071046965ac472a8171df03fc48164b88e4fb6cbc667d74c86f3e

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.