

CI/CD Pipeline with GitHub Actions & Docker

Introduction

This project demonstrates the implementation of a complete CI/CD pipeline using GitHub Actions and Docker. The pipeline automates the process of building a Docker image, running tests, and deploying the application locally without relying on cloud infrastructure.

Abstract

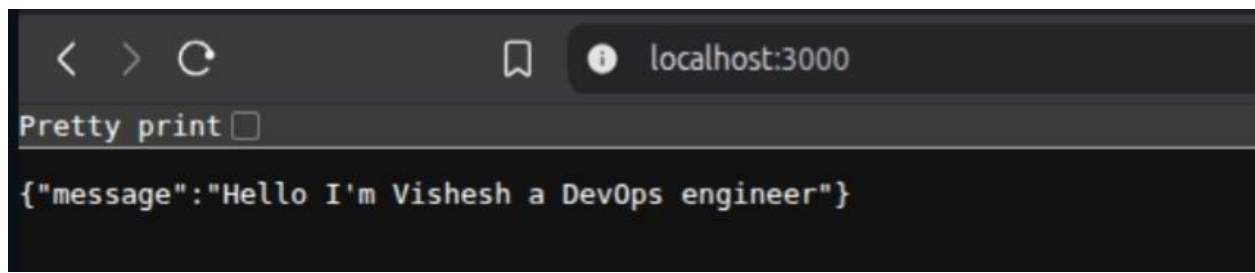
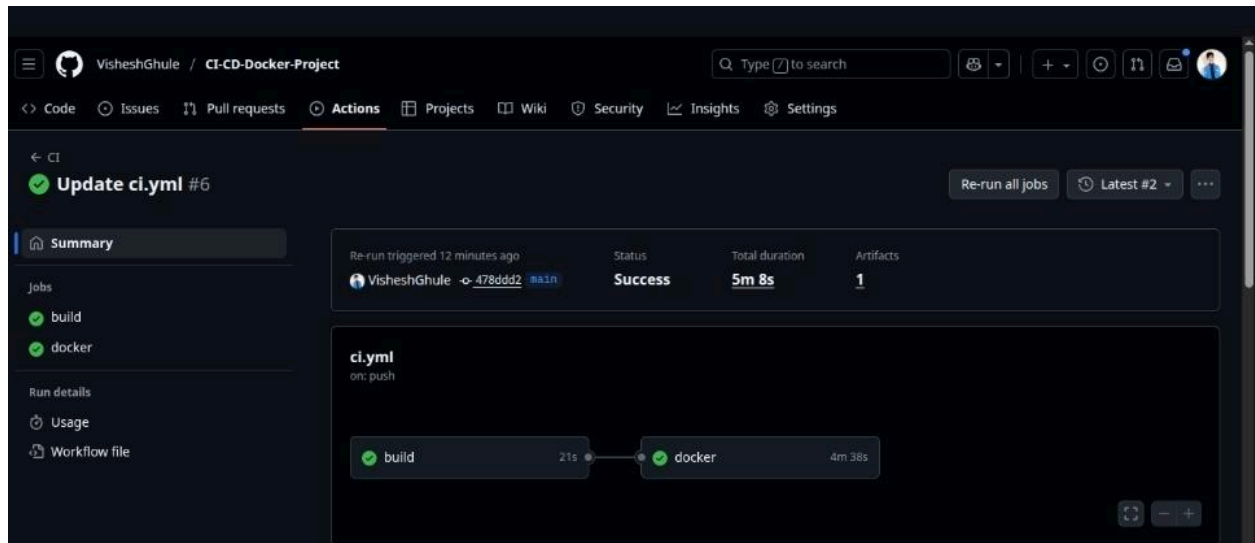
The objective of this project is to implement a full CI/CD workflow without using any cloud services. Using GitHub Actions, Docker, and Docker Hub, the project automates the building of Docker images, running tests, and deploying the application locally on local VM. This provides hands-on experience in continuous integration and deployment practices.

Tools Used

- *GitHub Actions* – for setting up the CI/CD workflow
- *Docker* – for containerizing the application
- *Docker Hub* – to store/share Docker images
- *Local VM* – to deploy and run the Docker container locally

Steps Involved in Building the Project

1. Created a Dockerfile to containerize the application and a docker-compose.yml for service configuration.
2. Configured GitHub Actions workflows to:
 - Run automated tests
 - Build the Docker image
 - Push the image to Docker Hub
3. Set up Local VM to pull the image from Docker Hub and run it locally.
4. Verified successful deployment through logs, outputs, and screenshots of the running application.



Conclusion

The project successfully established a working CI/CD pipeline with GitHub Actions and Docker. It automated code testing, image creation, and deployment, which are crucial DevOps practices. This pipeline can be extended to larger applications and integrated with cloud environments for more scalable and reliable deployments.

[Github Repo link](#)