

Project Title:

- CI/CD Pipeline for a Python Flask App Using GitHub Actions and Docker (No Cloud Needed)

Objective:

To build a complete Continuous Integration and Continuous Deployment (CI/CD) pipeline that:

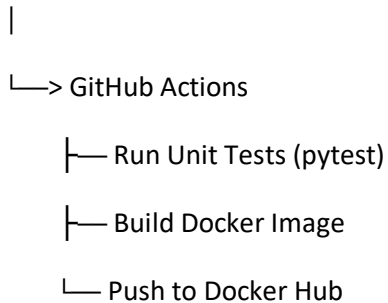
- Automatically builds and tests a Python Flask application.
- Builds and pushes a Docker image to Docker Hub.
- Deploys the application locally using Minikube or a local virtual machine.
- Demonstrates DevOps automation without relying on cloud platforms.

Tools & Technologies Used:

- Python 3.10 – Flask for the web app
- Pytest – Unit testing
- Docker – Containerization
- Docker Hub – Docker image registry
- GitHub Actions – CI/CD pipeline orchestration
- Minikube – Local Kubernetes cluster for deployment

Project Architecture:

GitHub Repo (Main Branch)



Minikube / Local VM (pulls image from Docker Hub)

└─ Deploys & exposes service at <http://localhost:<port>>

CI/CD Workflow Steps:

- Source Control: Code is hosted on GitHub (main branch).

Trigger: Workflow is triggered on every push to main.

Test Stage:

- Docker image is built for the app.
- Unit tests are executed using pytest inside the container.

Build & Push Stage:

- Docker Hub login is performed using GitHub Secrets.

- The application image is tagged and pushed to Docker Hub (ravindranadratagore/cicd:latest).

Deploy Stage (Manual/Local):

- Using Minikube, the image is pulled and deployed.
- Kubernetes exposes the service on a NodePort for browser access.

Docker Image:

Repository: Docker Hub – ravindranadhtagore/cicd

Success Indicators:

- GitHub Actions Workflow: Successfully completed with green check marks.
- Docker Hub Push: Verified image pushed and publicly accessible.
- Minikube Deployment: App runs and is reachable via minikube service.

Screenshots:

- GitHub Actions CI/CD run success
- Docker Hub image
- Local browser view of the deployed app

Conclusion:

This project demonstrates an end-to-end DevOps workflow using open-source tools without relying on any cloud platform. It automates code testing, Docker image creation, and deployment, serving as a scalable pattern for local CI/CD environments and beginner DevOps learning paths.