PROJECT_REPORT.md

Project: CI-CD-Python — End-to-End CI/CD Pipeline

1. Objective

Build a simple Python Flask application, containerize it, implement CI with GitHub Actions to run tests and build/push Docker images, and deploy locally to Minikube for validation.

2. Tools & Technologies

- Python 3.11 (Flask)
- Docker & Docker Hub
- GitHub & GitHub Actions
- Minikube & kubectl
- (Optional) Kubernetes manifests for reproducible deployment

3. Architecture

1. Developer pushes code to GitHub
2. GitHub Actions runs tests and builds Docker image
3. Built image is pushed to Docker Hub (latest + commit SHA)
4. The local environment (Minikube) pulls the Docker image and runs the app

4. Implementation Steps (detailed)

1. Application

2. `app.py` is a simple Flask app serving `/` on port 5000

3. `requirements.txt` lists dependencies (Flask)

4. Containerization

5. `Dockerfile` uses `python:3.11-slim` base

6. Copies requirements, installs, copies code, and sets `CMD ["python", "app.py"]`

7. Local Multi-container (docker-compose)

8. `docker-compose.yml` builds and maps port 5000

9. CI (GitHub Actions)

10. Workflow triggers: `push` and `pull_request` on `main`

11. Jobs:

    - `test`: uses `actions/setup-python@v5` to install and run `python -m unittest discover`
    - `build-and-push`: logs into Docker Hub using `docker/login-action@v3` and uses `docker/build-push-action@v6` to build and push image tags

12. Deployment

13. Use Minikube to run the container locally

14. Use `kubectl create deployment` or `kubectl apply -f k8s/` with YAML manifests

5. Testing Plan

- Unit tests: add `tests/` directory and run with `python -m unittest`
- Integration/smoke: run container locally and `curl` the endpoint
- Deployment verification: after deploy, validate via `kubectl get pods`, `kubectl logs`, and access service URL
- Canary/staged release (future): use two tags and traffic routing with Istio for canary testing

6. Rollback & Recovery

- If deployment fails, rollback by re-deploying previous tag:

```
kubectl set image deployment/python-app python-app=yourdockerhub/ci-cd-python:<previous-tag>
```

- Or scale down the new deployment and scale previous back to desired replica count

7. Deliverables Checklist

-

8. Appendix — Useful Commands

```
# Local dev
python -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt
python app.py

# Docker
docker compose up --build
docker build -t yourdockerhub/ci-cd-python:latest .
docker push yourdockerhub/ci-cd-python:latest

# Minikube deploy
```

```
minikube start
kubectl create deployment python-app --image=yourdockerhub/ci-cd-python:latest
kubectl expose deployment python-app --type=NodePort --port=5000
minikube service python-app --url

# GitHub Actions
# Check the Actions tab in your repo for workflow run logs
```

Prepared based on the repository and earlier project work.