CI/CD Pipeline with GitHub Actions & Docker

Abstract:

This project demonstrates the implementation of a Continuous Integration and Continuous Deployment (CI/CD) pipeline using GitHub Actions and Docker. The pipeline automates building, testing, and deployment of a sample Flask web application, ensuring faster and more reliable delivery.

Introduction:

In modern software engineering, automation is crucial for efficiency and reliability. CI/CD pipelines help streamline the development process by integrating automated builds, testing, and deployments. This project highlights how a developer can leverage GitHub Actions with Docker to set up a simple yet effective CI/CD workflow.

Tools Used:

- **GitHub Actions:** For CI/CD workflow automation.
- **Docker:** To containerize and deploy the Flask application.
- **Docker Hub:** For storing and distributing Docker images.
- **Flask:** Lightweight web framework used for the demo application.

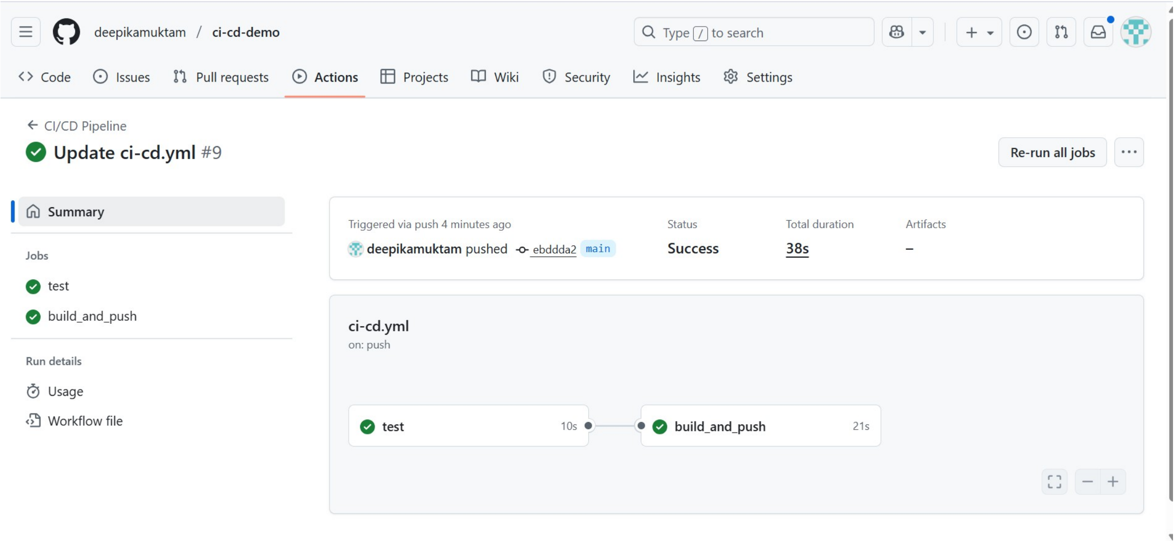
Steps Involved:

1. Developed a basic Flask application.
2. Wrote a Dockerfile to containerize the app.
3. Configured GitHub Actions workflow for building and testing the app.
4. Pushed the Docker image to Docker Hub.
5. Pulled and deployed the Docker image locally using Docker run.
6. Verified the application running at *localhost:8080*.

Conclusion:

The project successfully demonstrated a working CI/CD pipeline using GitHub Actions and Docker. It automated the build and deployment process, reducing manual effort and ensuring consistency. This approach can be extended to larger projects with cloud deployments, enhancing productivity and reliability in software development.

Screenshots:



Update ci-cd.yml #9

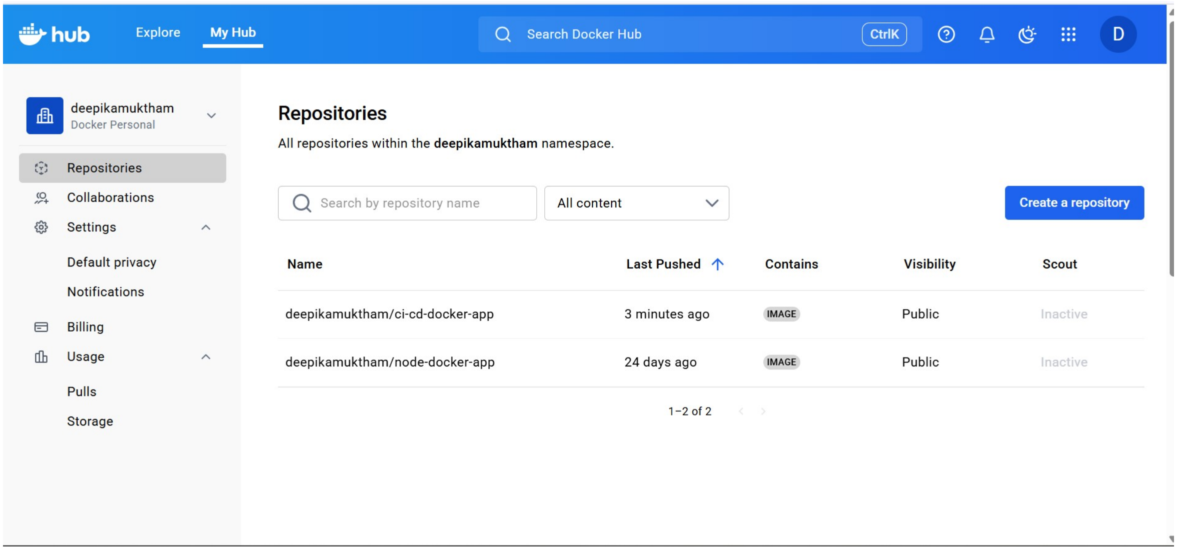
Triggered via push 4 minutes ago
deepikamuktam pushed ebddda2 main

Status
Success

Total duration
38s

Artifacts
–

ci-cd.yml
on: push



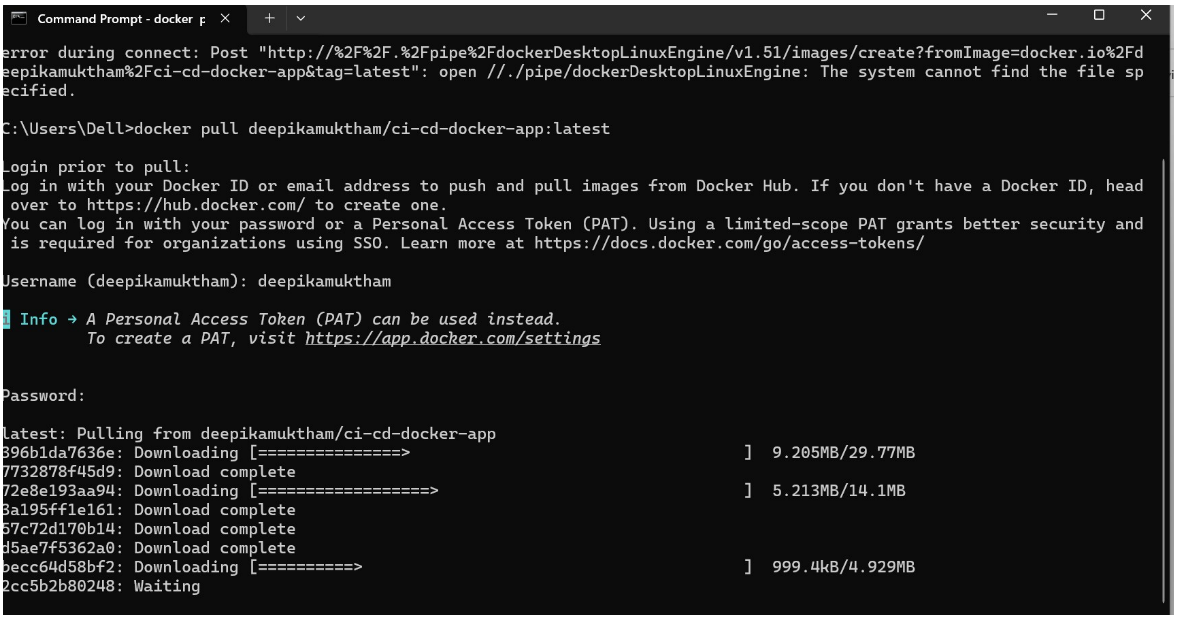
Repositories

All repositories within the **deepikamuktham** namespace.

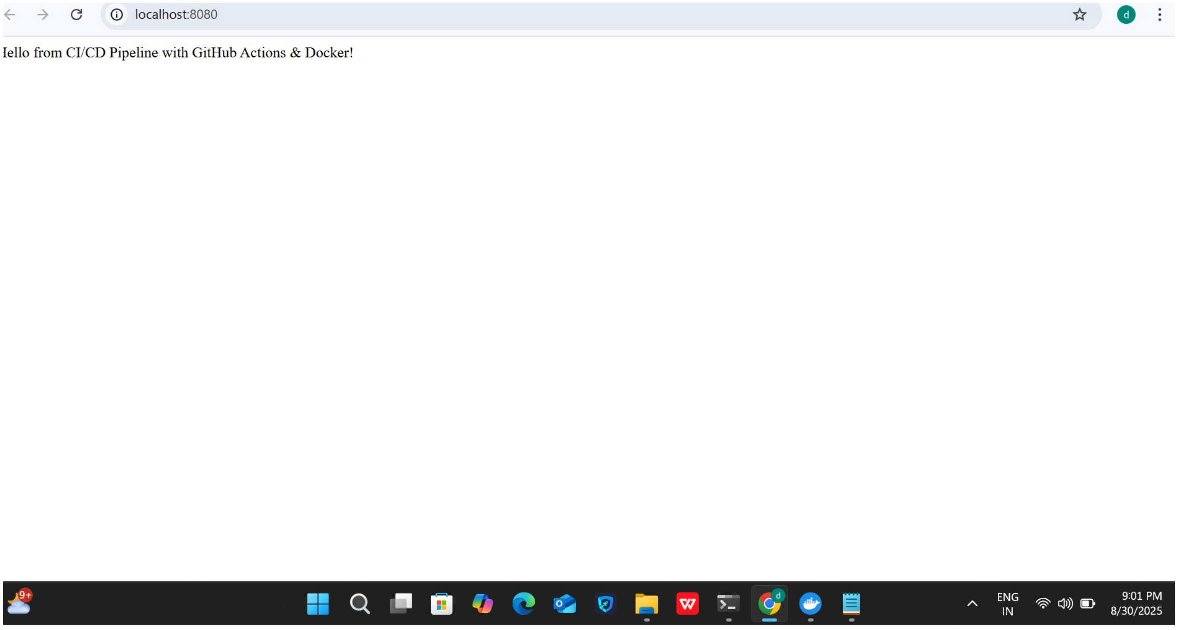
Search by repository name · All content

Name	Last Pushed	Contains	Visibility	Scout
deepikamuktham/ci-cd-docker-app	3 minutes ago	IMAGE	Public	Inactive
deepikamuktham/node-docker-app	24 days ago	IMAGE	Public	Inactive

1–2 of 2



```
error during connect: Post "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.51/images/create?fromImage=docker.io%2Fd
eepikamuktham%2Fci-cd-docker-app&tag=latest": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file sp
ecified.

C:\Users\Dell>docker pull deepikamuktham/ci-cd-docker-app:latest

Login prior to pull:
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head
 over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and
 is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username (deepikamuktham): deepikamuktham

i Info → A Personal Access Token (PAT) can be used instead.
         To create a PAT, visit https://app.docker.com/settings


Password:

latest: Pulling from deepikamuktham/ci-cd-docker-app
396b1da7636e: Downloading [===============>                                   ]  9.205MB/29.77MB
7732878f45d9: Download complete
72e8e193aa94: Downloading [==================>                                ]  5.213MB/14.1MB
3a195ff1e161: Download complete
57c72d170b14: Download complete
d5ae7f5362a0: Download complete
becc64d58bf2: Downloading [==========>                                        ]  999.4kB/4.929MB
2cc5b2b80248: Waiting
```



Iello from CI/CD Pipeline with GitHub Actions & Docker!