CI/CD Pipeline with GitHub Actions & Docker (No Cloud Needed)

Introduction

Continuous Integration and Continuous Deployment (CI/CD) pipelines automate the process of building, testing, and deploying software, ensuring faster and more reliable delivery. This project implements a CI/CD pipeline using GitHub Actions and Docker to build, test, and deploy a Java application locally.

Objective

To set up a fully automated pipeline that builds a Docker image of a Java application, runs automated tests, pushes the image to Docker Hub, and deploys the container in a local environment (Minikube or local VM), without relying on cloud services.

Tools Used

- **GitHub Actions:** For creating automated workflows to build, test, and push Docker images.
- **Docker:** To containerize the Java application using Dockerfile and docker-compose.yml.
- **Docker Hub:** Free container registry to store and share Docker images.

- **Minikube / Local VM:** To run and test the deployed containerized app locally.
- **Java & Maven:** For building and testing the Java application.

Steps Involved in Building the Project

1. **Write Dockerfile:** Created a Dockerfile to containerize the Java Maven application.
2. **Create docker-compose.yml:** Defined multi-container setup or dependencies if needed.
3. **Setup GitHub Actions Workflow:**
    - Configured YAML workflows to trigger on push/pull events.
    - Steps include checkout, build with Maven, run tests, build Docker image, and push to Docker Hub.

4. **Push Docker Image:** Successfully pushed the built Docker image to Docker Hub repository.

5. **Deploy Locally:** Pulled the Docker image on Minikube or local VM and ran the container to verify deployment.

6. **Testing:** Verified application functionality after deployment.

7. **Capture Results:** Collected workflow run logs, Docker Hub image URL, and screenshots of the running app.

Deliverables

- GitHub repository containing the full source code and CI/CD workflows.

- Docker image hosted on Docker Hub (link included).

- Workflow run logs showing successful builds and tests.

- Screenshots demonstrating the deployed app running locally.

Conclusion

The project demonstrates a practical and cloud-free CI/CD pipeline that automates build, test, and deployment cycles. Using GitHub Actions and Docker, it ensures reliable, repeatable delivery of containerized Java applications in local environments, streamlining the development process.