

CI/CD Pipeline Using GitHub Actions

Introduction

This report outlines the implementation of a CI/CD pipeline using GitHub Actions for a Java-based application. The goal of this project is to automate the software development lifecycle, including compilation, testing, code quality checks, containerization, and deployment.

Abstract

In this project, a fully automated CI/CD pipeline is developed using GitHub Actions and a self-hosted runner (AWS EC2). The pipeline follows a sequence of steps: Maven compile, unit testing, code quality analysis with SonarQube, Docker image creation, image push to Docker Hub, and container deployment. The project demonstrates how modern DevOps tools can streamline and secure the software delivery process.

Tools Used

- GitHub Actions
- Maven
- SonarQube
- Docker
- Docker Hub
- AWS EC2 (Self-hosted Runner)
- Java 17 (Temurin)

Steps Involved in Building the Project

1. **Code Checkout**: The pipeline begins with fetching the latest code from the GitHub repository.
2. **Maven Compile**: The application is compiled using `mvn compile`.
3. **Maven Test**: Unit tests are executed using `mvn test` to ensure code correctness.
4. **SonarQube Analysis**: Static code analysis is performed using SonarQube to evaluate code quality.
5. **Maven Package**: The application is packaged into a JAR file using `mvn package`.
6. **Upload Artifact**: The generated JAR file is uploaded as a GitHub artifact.
7. **Docker Image Build**: A Docker image is created using the JAR file.
8. **Push to Docker Hub**: The image is pushed to Docker Hub under the tag `sajjuquadri/bankapp:latest`.
9. **Deployment**: The image is pulled and deployed as a Docker container on the self-hosted runner.

Conclusion

This project successfully demonstrates a robust and automated CI/CD pipeline using GitHub Actions. By integrating Maven, SonarQube, and Docker in a seamless workflow, it provides a production-grade solution for modern software delivery. This approach improves software quality, reduces manual effort, and ensures faster deployment cycles.

The screenshot displays a GitHub Actions workflow run for the 'Update cicd.yml' action. The workflow is titled 'Update cicd.yml #16' and is part of a 'CICD Pipeline'. It was triggered by a push to the 'main' branch by user 'sajjuquadri' 5 minutes ago. The status is 'Success', with a total duration of 5m 56s and 2 artifacts generated.

The workflow steps are listed on the left:

- compile
- test
- build_sonar_scan
- build_docker_image_and_push
- deploy_container

The main section shows a progress bar for the 'cicd.yml' workflow, with steps: compile (10%), test (20%), build_sonar_scan (30%), build_docker_image_and_push (40%), and deploy_container (50%). Below this, there are sections for 'build_docker_image_and_push summary' and 'Docker Build summary'.

The browser tabs at the top include: 'Update cicd.yml - sajjuquadri', 'Instance details | EC2 | AWS', 'SonarQube', 'Unzip command not found', and 'Docker Hub'. The Windows taskbar at the bottom shows the time as 22:06 on 25-06-2025.