

# Project Report

---

## CI/CD Pipeline with GitHub Actions & Docker

### Abstract

This project demonstrates the implementation of a lightweight CI/CD pipeline using GitHub Actions and Docker. The pipeline automates testing, containerization, and deployment of a sample application. A Docker image is built and pushed to Docker Hub, and the application is deployed locally using Docker Compose. This setup illustrates the complete cycle of continuous integration and delivery without relying on cloud infrastructure.

### Introduction

Continuous Integration and Continuous Deployment (CI/CD) have become essential practices in modern software development. They ensure that code is regularly tested, packaged, and deployed in an automated manner. This project implements a basic CI/CD pipeline using GitHub Actions and Docker Hub to show how automation reduces manual work and provides consistent deployments.

### Tools Used

- GitHub Actions - for setting up the CI/CD workflow
- Docker - for containerization
- Docker Hub - to store and share container images
- Docker Compose - for local deployment and testing
- Node.js - for a minimal sample app and test scripts

### Steps Involved in Building the Project

1. Project Setup: Created a GitHub repository and added necessary files (Dockerfile, docker-compose.yml, package.json, app.js, and workflow YAML).
2. Dockerfile and App Creation: Wrote a minimal Node.js app (app.js) and a Dockerfile for containerization.
3. Docker Compose Configuration: Defined docker-compose.yml to run the container locally on port 3000.
4. GitHub Actions Workflow: Configured workflow to checkout, set up Node.js, run tests, login to Docker Hub, prepare and push image.
5. Deployment and Testing: Ran docker compose up, tested the app at <http://localhost:3000>, and captured screenshots.

## Conclusion

This project successfully demonstrates a simple but effective CI/CD pipeline. By automating the build, test, and deployment processes, the project reflects the core principles of modern DevOps practices. The pipeline can be extended for more complex applications or cloud deployment. Even in this minimal form, it highlights the importance and practicality of CI/CD workflows in software engineering.