

CI/CD Pipeline with GitHub Actions & Docker – Project Report

Name: KANNAIAH LOKESH

GitHub Repository: <https://github.com/Lokesh-Soft-Dev/devops-ci-cd-simple-app>

Docker Hub Repository: <https://hub.docker.com/repository/docker/lokesh111996/simple-app/>

1. Overview of the Project

- The objective of this project is to design and implement a complete CI/CD pipeline without cloud services.
- The pipeline automatically tests the application, builds a Docker image, and pushes it to Docker Hub.
- The image is then deployed locally to validate the delivery stage. This demonstrates core DevOps automation skills used in real-world software delivery environments.

2. Tools & Technologies Used

- GitHub Actions – CI/CD automation
- Docker – Containerization
- Docker Hub – Image registry
- Node.js – Application runtime
- Git – Version control
- PowerShell/Git Bash – Terminal operations

3. Implementation Steps

Step 1: Developed a Node.js application with "/", "/health", and "/metrics" endpoints.

Step 2: Wrote a Dockerfile using node:18-alpine base image, installed dependencies, exposed port 3000.

Step 3: Configured GitHub Actions workflow (.github/workflows/ci-cd.yml) to:

- Install dependencies
- Run tests
- Build Docker image using buildx
- Login to Docker Hub using secrets
- Tag and push image (latest + commit SHA)

Step 4: Pulled and deployed the Docker image locally using:

- `docker pull lokesh111996/simple-app:latest`
- `docker run -d -p 5000:3000 lokesh111996/simple-app:latest`

4. Results

- CI pipeline successfully executed in GitHub Actions.
- Docker image pushed to Docker Hub with both latest and SHA tags.
- Local deployment validated using browser and curl:
 - ✓ `curl http://localhost:5000/`
 - ✓ `curl http://localhost:5000/health`

5. Screenshots Included (to be attached by student)

- GitHub Actions run results
- Docker build and push logs
- Docker Hub tags page
- Local application running in browser
- docker ps output

6. Conclusion

The project successfully demonstrates automation of build, test, and deployment stages using GitHub Actions and Docker. It can be extended further to Kubernetes (Minikube), monitoring, or auto-deployment to production-like environments.

* * *