DevOps Observability Project – Report

Introduction

This project demonstrates a modern DevOps workflow using a CI/CD pipeline and a containerized observability system with Docker Compose. The goal is to automate deployment, monitor service health, and visualize application performance using open-source tools.

Abstract

A Node.js sample application is containerized and deployed via a GitHub Actions-driven CI/CD pipeline, which builds and publishes images to Docker Hub. The application and supporting observability stack (Prometheus, Grafana, Jaeger) are orchestrated locally using Docker Compose. Prometheus scrapes metrics from the app, Grafana provides live dashboards, and Jaeger enables distributed tracing. The project delivers a reproducible template for real-world system monitoring and deployment automation.

Tools Used

Docker, Docker Compose: Container and service orchestration

Prometheus: Metrics collection

Grafana: Dashboard visualization

Jaeger: Distributed tracing

GitHub Actions: CI/CD pipeline automation

Git: Version control

Node.js: Application runtime

Steps Involved

Developed a basic Node.js app and provided a Dockerfile for containerization.

Created .github/workflows/ci-cd-pipeline.yml to automatically build and push images to Docker Hub on each git push.

Designed a docker-compose.yml that launches the app, Prometheus, Grafana, and Jaeger.

Configured Prometheus (prometheus.yml) to scrape the app's /metrics endpoint.

Validated setup:

Accessed Prometheus at localhost:9090

Explored metrics and dashboards in Grafana (localhost:3001)

Inspected traces in Jaeger UI (localhost:16686)

Exported dashboard JSON and captured screenshots for documentation

Committed all code, configs, and documentation to https://github.com/Vivek8951/Devops-Project.

Conclusion

This project showcases automated CI/CD and complete observability for a production-like environment using popular DevOps tools. It demonstrates best practices in automation (with GitHub Actions), containerization (with Docker Compose), and monitoring (with Prometheus, Grafana, and Jaeger). The working setup is ideal for interview discussions and as a foundation for more advanced DevOps implementations.

Attachments in repository:

README.md (setup & usage)

Screenshots of Grafana, Jaeger, Prometheus

Exported Grafana dashboard JSON