Project Report:

Dockerized Node.js App Deployment using CI/CD with GitHub Actions & Minikube

1. Introduction

In today's DevOps-driven software lifecycle, automating the deployment pipeline is crucial for faster, more reliable delivery. This project showcases a complete **CI/CD workflow** using **GitHub Actions**, **Docker**, and **Minikube** to automate the testing, building, and deployment of a **Node.js Express application** to a remote Kubernetes environment.

2. Abstract

This project focuses on deploying a containerized Node.js application using a streamlined CI/CD pipeline. It leverages GitHub Actions for workflow automation, Docker for containerization, and Minikube for Kubernetes-based orchestration. The application is automatically tested, built, and deployed to a remote server each time changes are pushed to the repository. The pipeline ensures minimal manual intervention, promotes faster delivery, and provides a robust deployment environment using modern DevOps tools and practices.

3. Tools Used

Node.js + Express	Backend application framework
Docker	Containerization of the Node app
Docker Hub	Hosting Docker images
GitHub Actions	CI/CD workflow automation
Minikube	Local Kubernetes cluster for deployment
GitHub Secrets	Secure storage for sensitive credentials
SSH (Remote Login)	Secure remote server access

4. Steps Involved in Building the Project

4.1. Application Development

- Built a basic **Node.js Express** application with required routes and logic.

4.2. Dockerization

- Build a Dockerfile to containerize the application.
- Test Docker image locally before pushing to Docker Hub.

4.3. CI/CD Pipeline with GitHub Actions

- Configure .github/workflows/main.yml to automate the pipeline.
- Pipeline stages include:
  - Checking out source code
  - Setting up Node.js (v18)
  - Installing dependencies and running tests
  - Logging into DockerHub using **GitHub Secrets**
  - Building and pushing Docker image to DockerHub
  - Secure SSH login to remote server and starts Minikube
  - Cloning the GitHub repository inside the remote server
  - Deploy application to Minikube using Kubernetes manifests
  - Expose the application via a Kubernetes service

---

5. Conclusion

This project successfully demonstrates the automation of application delivery using a robust DevOps toolchain with GitHub Actions managing CI/CD, Docker enabling portability, and Minikube offering Kubernetes-based orchestration, the pipeline ensures seamless deployment from code commit to live application. The project reflects modern deployment standards and provides a scalable model for future enhancements and production-grade deployments.

Live Output: Click-Here
GitHub Repository: Click-Here

SUBMITTED BY:

K. PRANAY SESA SAI