Wellnest EHR

Presented by-

Madhura Aher(23070122507)

Mayank Verma(2370122508)

Suyash Jagtap(23070122514)

Introduction: WellNest EHR System

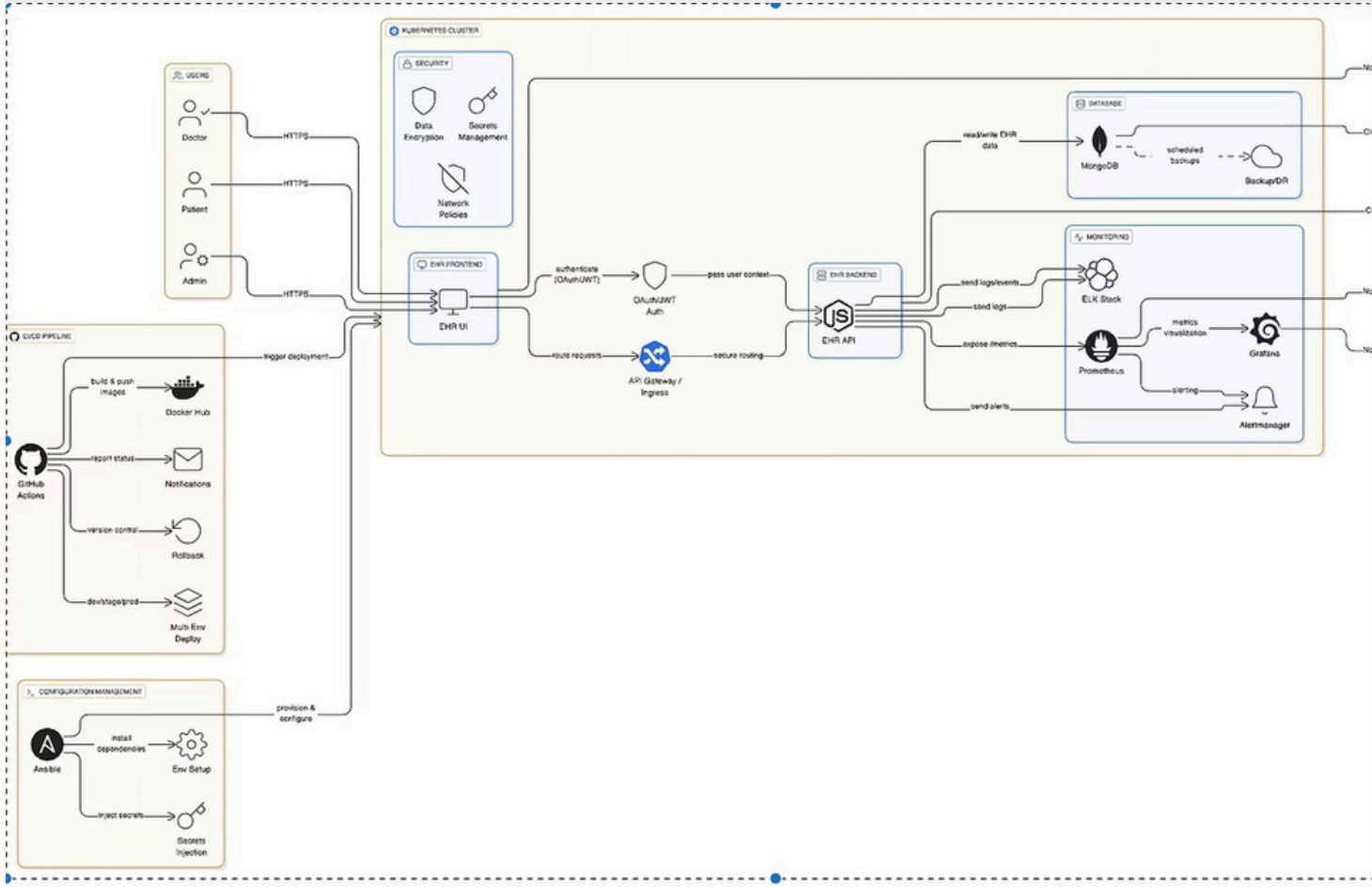
TheWellNest ElectronicHealth Record (EHR) system isaweb-basedhealthcaremanagement application designed to streamline patient appointments and health records. It provides features for:

- Booking appointments with doctors across departments
- Managing patient information securely
- Offering an intuitive frontend (HTML, CSS, JS) and a backend (Node.js + Express + MongoDB)
- Ensuring data persistence through MongoDB

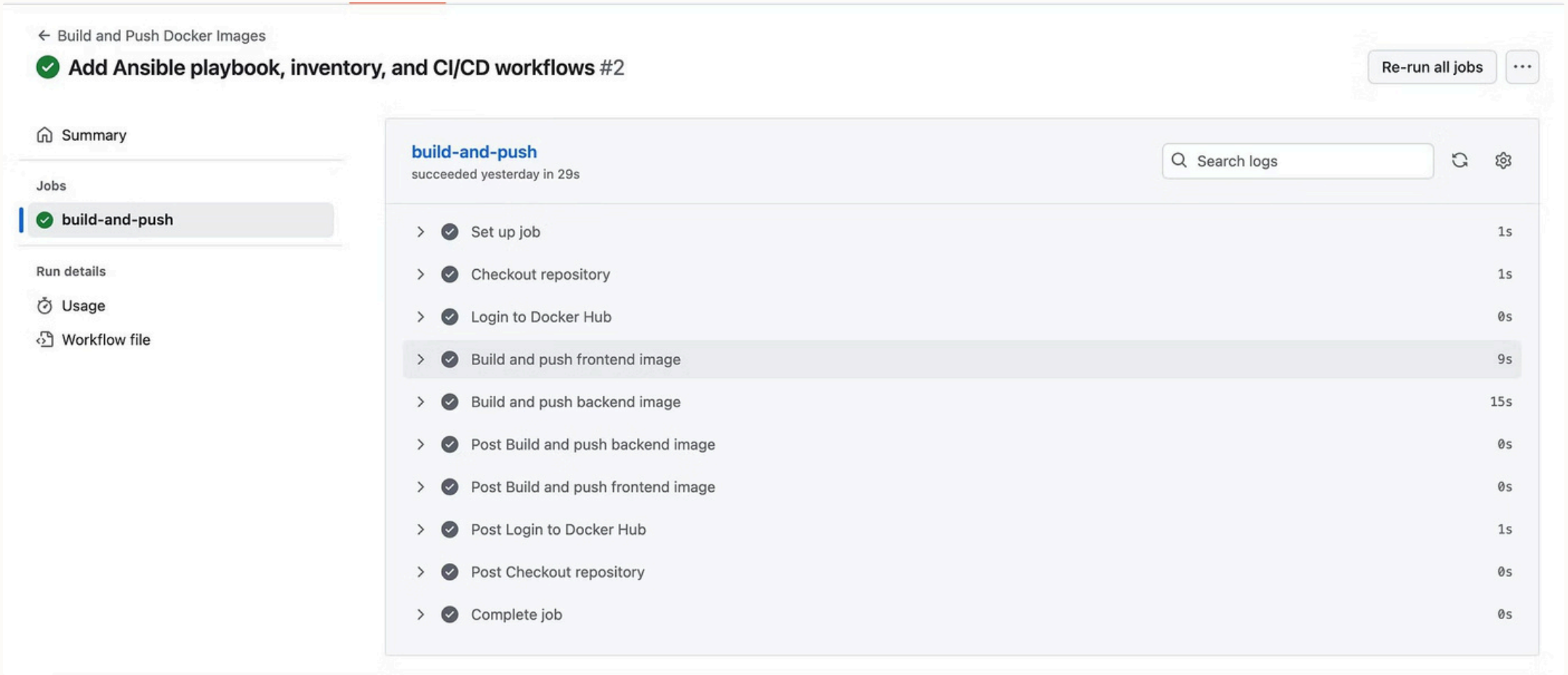
This project serves as the base for demonstrating the complete DevOps lifecycle 4 from containerization to continuous deployment, monitoring, and automation.

Architecture Overview

- Frontend: NGINX + HTML/CSS/JS
- Backend: Node.js + Express + MongoDB
- Containerized using Docker (frontend & backend)
- Managed with Kubernetes (Deployments, Services)
- Monitored via Prometheus & Grafana
- CI/CD automated with GitHub Actions



CI/CD Pipeline Flow



Challenges

- **Kubernetes Networking & MongoDB Connectivity**
- Backend couldn9t connect to MongoDB due to DNS resolution issues.
- Fixed by correcting service names & ensuring both pods were in the same namespace.
- **Prometheus Target Discovery Errors**
- Backend metrics target showed as *DOWN* in Prometheus.
- Solved by updating scrape target to `ehr-backend-service.default.svc.cluster.local:5000/metrics`.
- **CI/CD Authentication Failures**
- Docker image push failed due to invalid credentials in GitHub Actions.
- Resolved using **GitHub Secrets** for secure Docker Hub authentication

Learnings:

- **End-to-End DevOps Lifecycle**
- Implemented CI/CD pipeline with automated build, push, deploy, and monitor stages.
- **Hands-on CI/CD & Monitoring**
- Integrated Prometheus & Grafana for real-time health and performance metrics.
- Learned proactive monitoring and debugging through metrics analysis.
- **Kubernetes Orchestration Mastery**
- Designed deployments, services, and ConfigMaps.
- Understood rolling updates, rollbacks, and scaling concepts.
- **Collaborative Problem Solving**
- Faced real-world DevOps issues requiring team effort and research.
- Gained confidence in cloud-native architecture and production workflows.

Learnings

1. **End-to-End DevOps Lifecycle** Implemented CI/CD pipeline with automated build, push, deploy, and monitor stages.
2. **Hands-on CI/CD & Monitoring:** Integrated Prometheus & Grafana for real-time health and performance metrics. Learned proactive monitoring and debugging through metrics analysis.
3. **Kubernetes Orchestration Mastery:** Designed deployments, services, and ConfigMaps.Understood rolling updates, rollbacks, and scaling concepts.
4. **Collaborative Problem Solving:** Faced real-world DevOps issues requiring team effort and research.Gained confidence in cloud-native architecture and production workflows.