## Project Overview

This repository implements a scalable CI/CD pipeline for a full-stack web and mobile application consisting of:

- Frontend: React
- Backend: Django / Node.js
- Mobile App: React Native
- Authentication: Firebase
- Deployment: DigitalOcean Kubernetes Cluster
- CI/CD Tool: GitHub Actions
- Monitoring: Prometheus + Grafana
- Notifications: Slack

## Architecture Overview

Environment Segregation:

- main  Triggers production deployment
- develop  Triggers staging deployment

## Containerization

- Each component (frontend, backend, mobile-app) has its own Dockerfile.
- Services managed using Kubernetes YAML manifests.

## CI/CD Workflow (GitHub Actions)

On PR Merge to main:

- Lint & run unit tests
- Build Docker images and push to DockerHub
- Deploy to Kubernetes using kubectl
- Send Slack notification for build status and deployment

## Kubernetes Structure

k8s/

backend.yaml

frontend.yaml

ingress.yaml

Each service is exposed via Ingress and uses a LoadBalancer on DigitalOcean.

## Secrets Management

- GitHub Secrets:
- DOCKER_USERNAME, DOCKER_PASSWORD
- SLACK_WEBHOOK
- KUBE_CONFIG

## Monitoring & Alerts

- Prometheus monitors: CPU, Memory, Uptime, Latency
- Grafana dashboards
- Slack Alerts for CPU > 70%, Pod crash, Deployment status

## Rollbacks

- GitHub Actions can redeploy previous image
- Kubernetes Deployment with revision history and rollback

## Advanced Features

- Blue-Green Deployment Strategy
- Redis Caching Layer
- Canary rollout with Progressive Rollout

## Testing

```
cd frontend && npm test
cd backend && pytest
```

## Quick Start

```
docker build -t frontend ./frontend
kubectl apply -f k8s/
```

## License