

Requirement Analysis Document

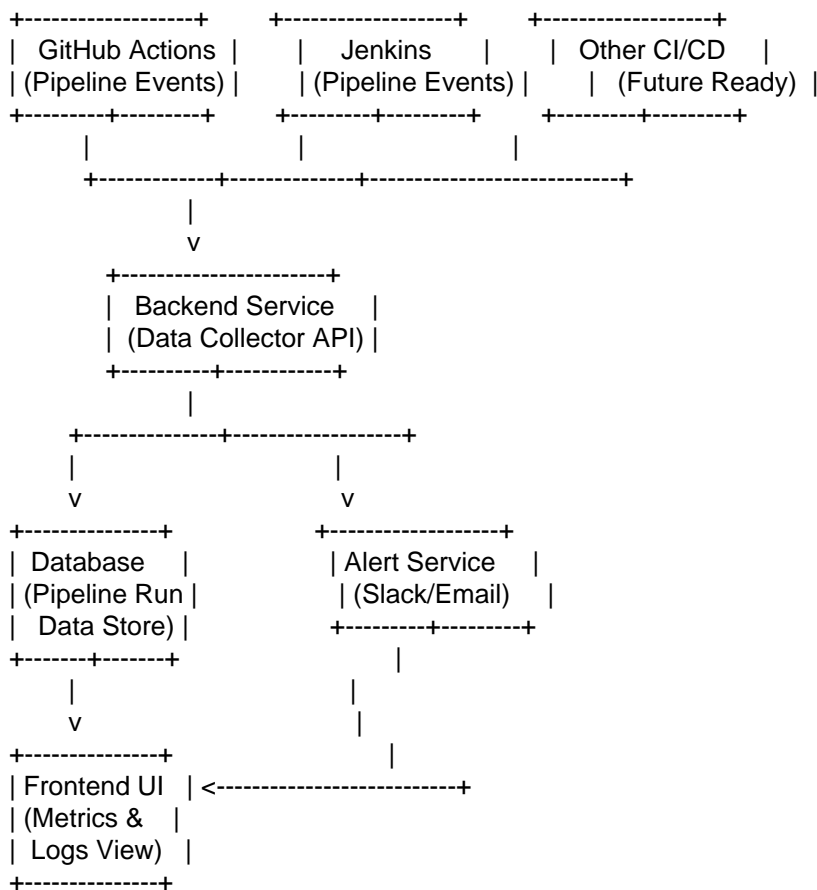
Project: CI/CD Pipeline Health Dashboard

Purpose: To provide real-time visibility into pipeline executions from CI/CD systems like GitHub Actions or Jenkins.

■ Key Features

- Data Collection – Fetch pipeline execution data (status, build time, logs) from CI/CD tools (GitHub Actions, Jenkins).
- Real-Time Metrics – Success/Failure rate of pipelines, Average build time, Last build status (success/failure/in-progress).
- Alerts & Notifications – Send alerts (Slack, Email) on pipeline failures or abnormal conditions.
- Frontend Dashboard – Visualize pipeline metrics using charts/graphs and display logs/status of latest builds in real-time.

■ Demo Flow Diagram



■ Tech Choices

Backend: Node.js/Express (API service), WebSockets (Socket.io) for real-time updates.

Database: PostgreSQL / MongoDB – Store pipeline run data.

Frontend: React.js with Tailwind CSS, Chart.js / Recharts for visualization.

Integrations: GitHub REST API / Jenkins API for pipeline status, Slack API / SMTP for alerts.
Containerization & Deployment: Docker & Docker Compose, Kubernetes (optional, future).

■ APIs / Tools Required

- GitHub API – /repos/{owner}/{repo}/actions/runs, job details/logs.
- Jenkins API – /job/{jobName}/lastBuild/api/json (build details), /consoleText (logs).
- Slack API – POST <https://slack.com/api/chat.postMessage>.
- Email – SMTP or SendGrid API for notifications.
- Frontend Libraries – Chart.js/Recharts, Axios for API calls.

■ Outcome

The dashboard will provide real-time visibility into CI/CD pipelines, help track success/failure rates, monitor build times, and alert teams in case of failures — ensuring faster issue detection and better DevOps efficiency.