Inventory Manager Application with CI/CD Pipeline: Project Report

# AVIJIT PRATAP SINGH

GITHUB-[Avijit-Github](https://github.com/Aviijeet12)

# 1. Project Overview

The "Inventory Manager" is a Java Spring Boot-based web application integrated with a fully automated DevOps CI/CD pipeline. It demonstrates how modern software systems are built, tested, deployed, and monitored using tools like Git, Jenkins, Docker, Prometheus, and Grafana.

# 2. Objective

To build a functional inventory management REST API and establish a CI/CD pipeline that enables automated building, testing, containerization, deployment, and monitoring of the application.

# 3. Tools and Technologies

Tool/Technology | Purpose

---------------------|----------------------------------------

Java + Spring Boot | Backend REST API development

Maven | Dependency & build management

Git + GitHub | Version control

Jenkins | CI/CD pipeline automation

Docker | Containerization

Prometheus | Monitoring data collection

Grafana | Visual dashboards and analytics

Ubuntu VM | Hosting and running services

# 4. Project Setup & Implementation

# 4.1 Application Code

- A Spring Boot application exposing REST endpoints (e.g., `/inventory`).

- Code organized into controller, service, and model packages.

- `pom.xml` configured with Spring Boot and JUnit dependencies.

# 4.2 CI/CD Pipeline (Jenkins)

- Jenkins pipeline defined via `Jenkinsfile`.

- Stages include:

- Build: `mvn clean package`

- Docker Build: `docker build -t inventory-manager .`

- Docker Run: `docker run -d -p 8080:8080 inventory-manager`

# 4.3 Docker Setup

- Dockerfile created to package the Spring Boot JAR into a Docker image.

- Docker commands used to build and run the container locally.

# 4.4 Prometheus + Grafana Setup

- Prometheus installed to collect metrics.

- Grafana configured to use Prometheus as a data source.

- Dashboards created to visualize system performance.

# 5. Project Usage

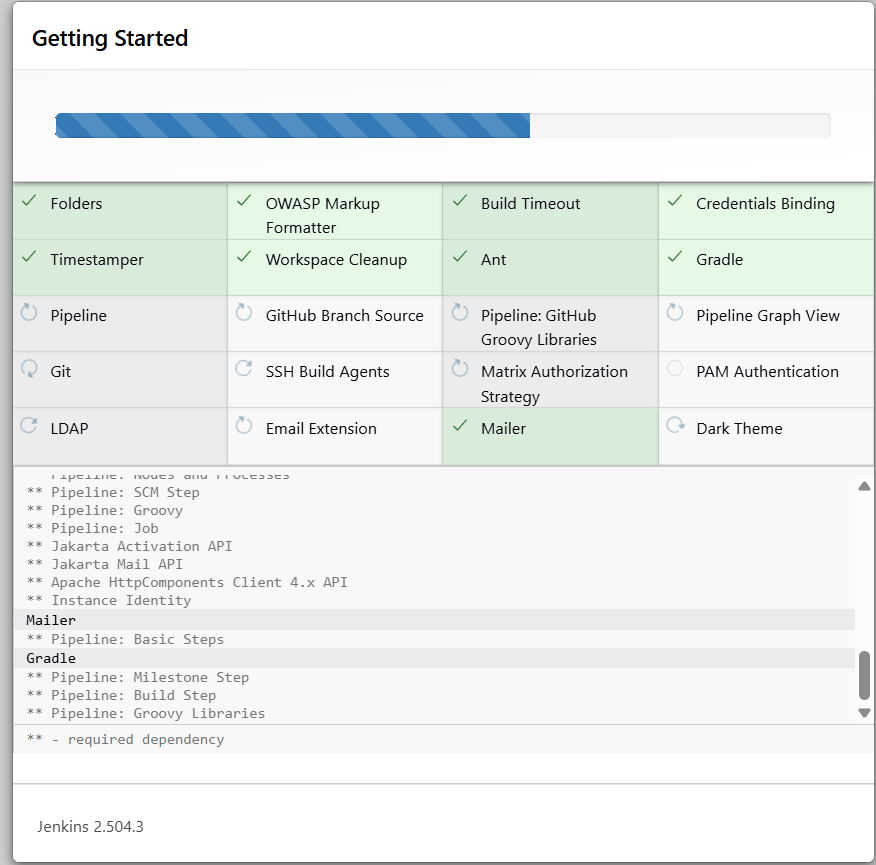
- Run the project locally or deploy on cloud.

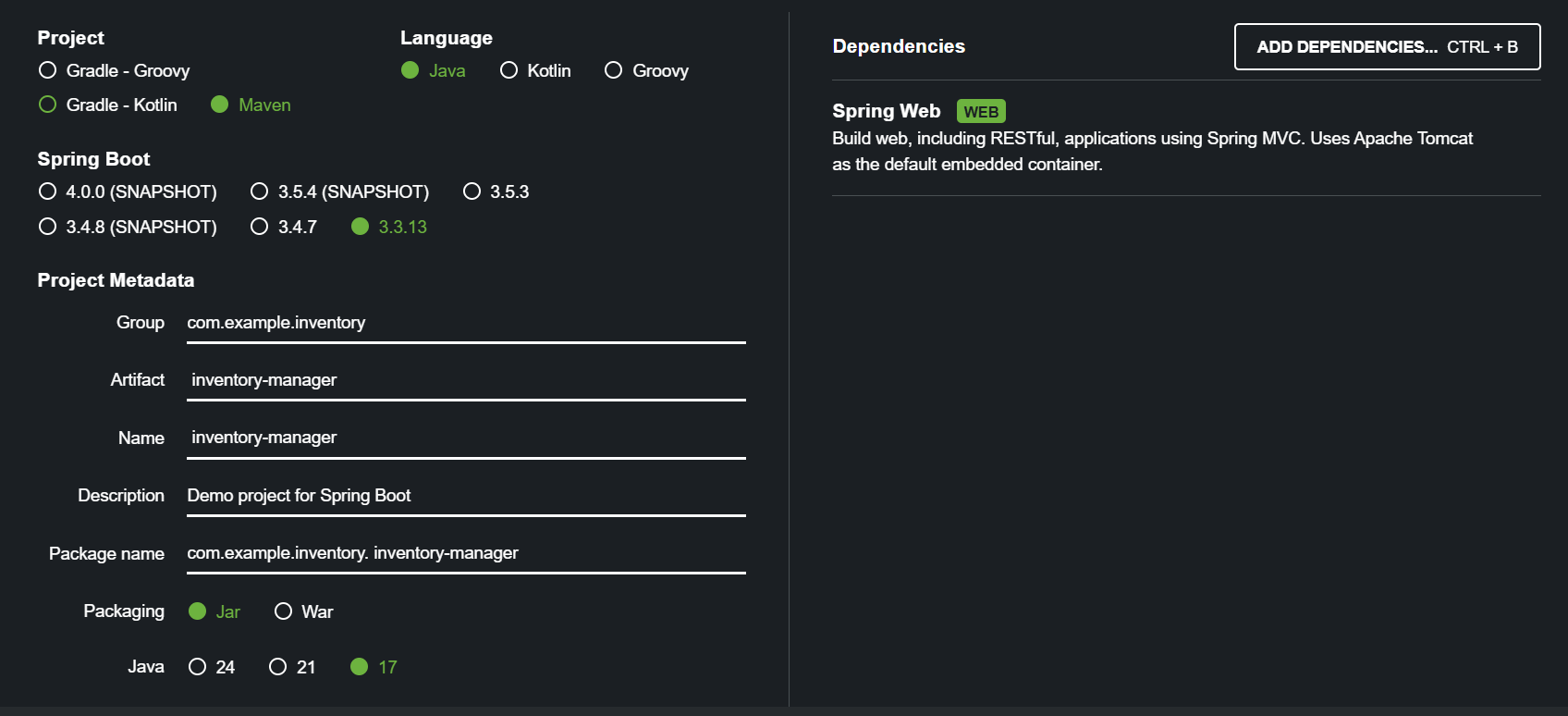
- Access endpoint: `http://localhost:8080/inventory`

- Monitor system via Grafana at: `http://localhost:3000`

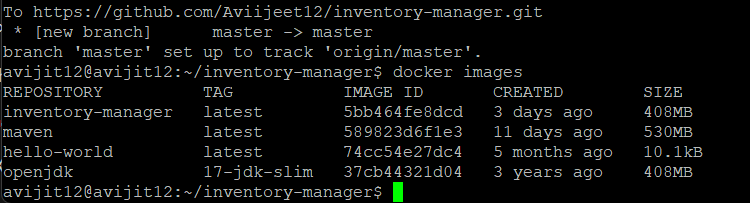
# 6. Screenshots

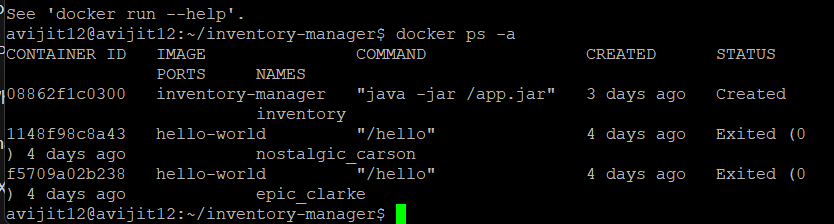
- Jenkins Pipeline Job Execution



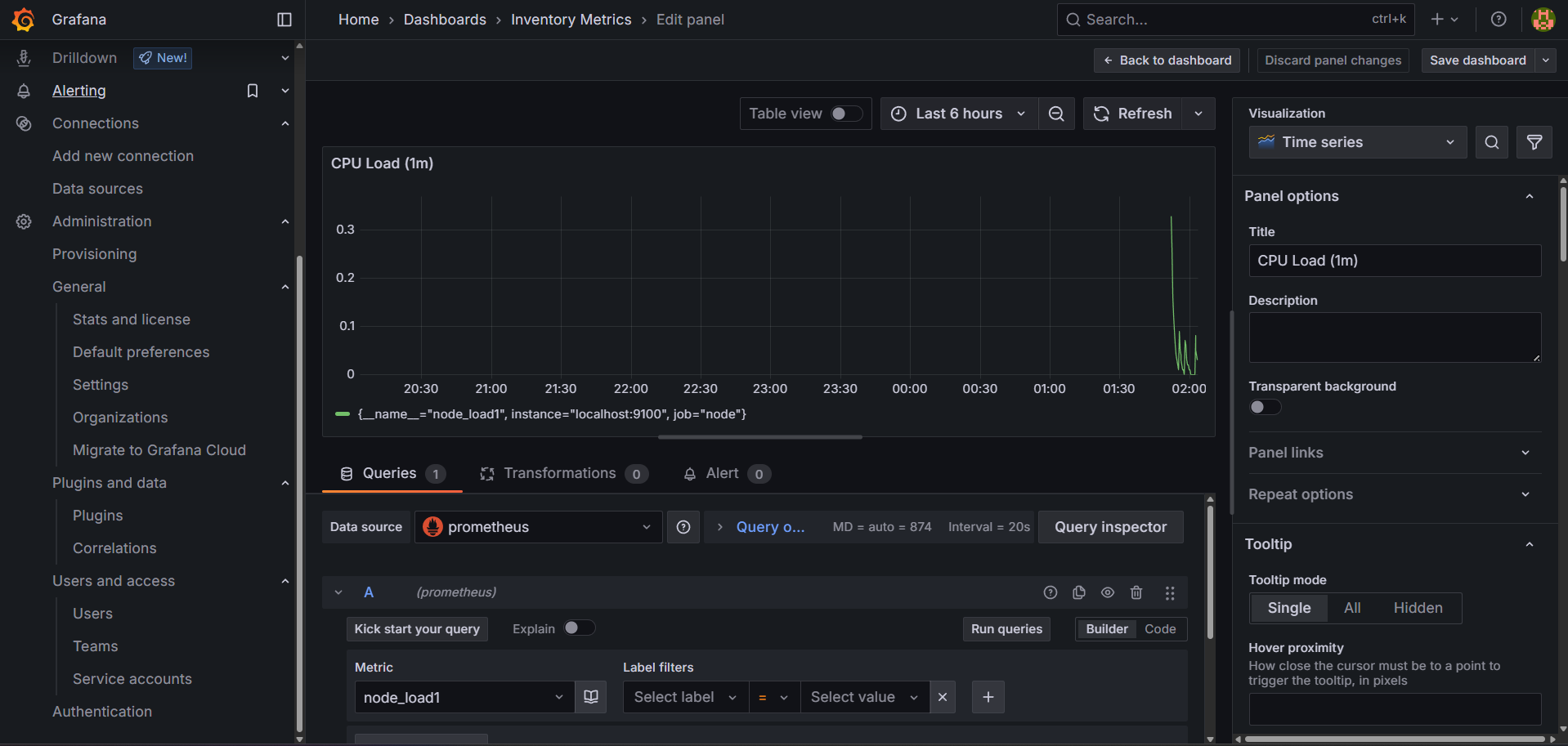


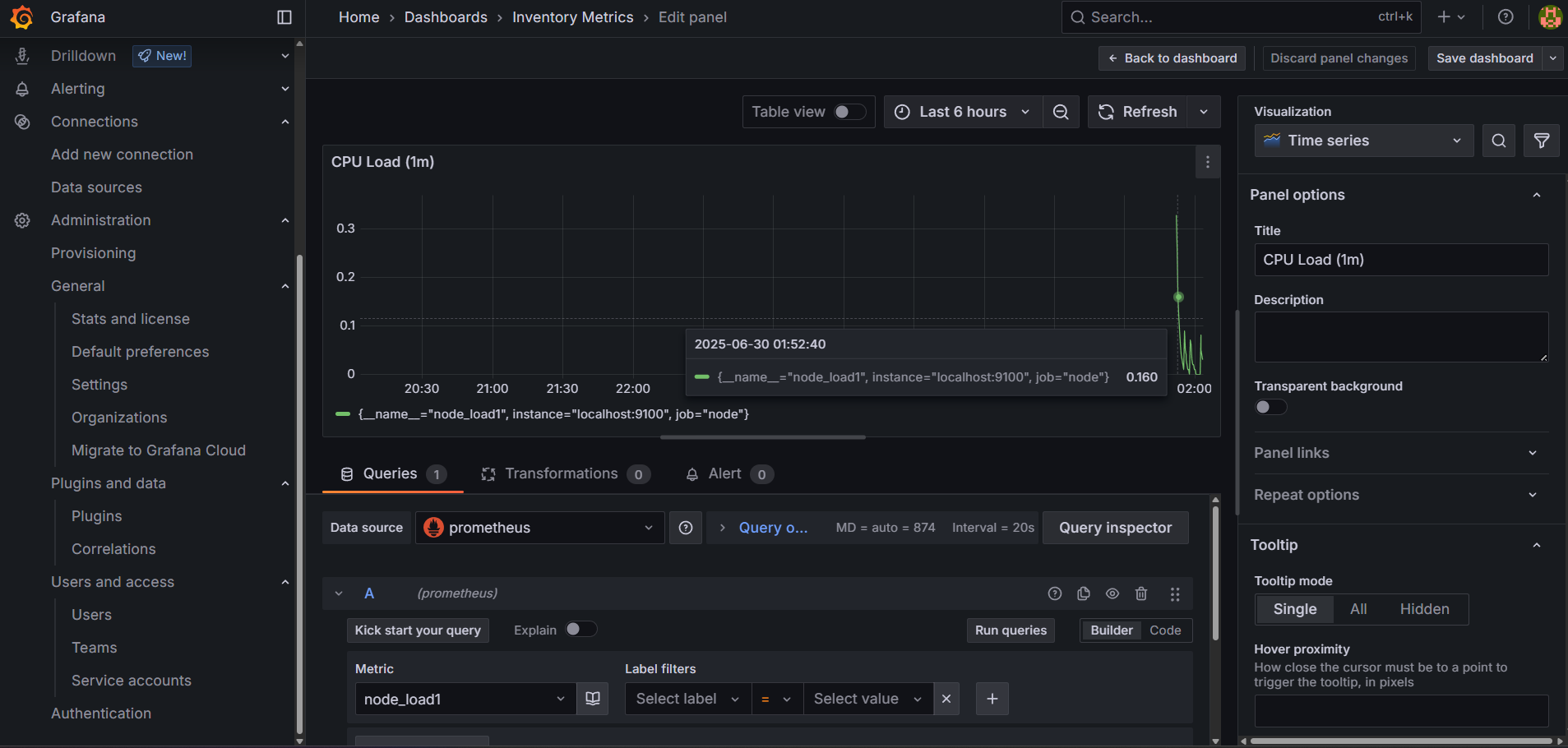
- Docker Image Creation





- Grafana Dashboard with Prometheus Metrics





# 7. Benefits and Learning Outcomes

- Full understanding of how to build, test, deploy and monitor software.

- Hands-on experience with production-grade DevOps tools.

- Ability to troubleshoot and automate workflows.

# 8. Future Enhancements

- Add a frontend using React or Angular.

- Add a database like MySQL/PostgreSQL.

- Enable GitHub Webhooks to auto-trigger Jenkins builds.

# 9. Author

Name: Avijit Pratap Singh

GitHub: [Avijit-Github](https://github.com/Aviijeet12)

Project Repository: [Inventory-Manager](https://github.com/Aviijeet12/inventory-manager)