

GRAFANA DASHBOARD FOR SYSTEM HEALTH MONITORING

Micro Project Report

Submitted by

LALITKUMAR S (22ITR055)

Course Code & Name : 22ITF02-DEVOPS

Programme & Branch : B.TECH - IT

Department : Information Technology

Kongu Engineering College, Perundurai

May 2025

BONAFIDE CERTIFICATE

Certified that this micro project documentation of “**GRAFANA DASHBOARD FOR SYSTEM HEALTH MONITORING**” is the bonafide work of “**LALITKUMAR S(22ITR055)**” who carried out the project under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of my other thesis or dissertation on the basis of which a degree or awarded was conferred on an earlier occasion on this or any other candidate.

Submitted on _____.

SIGNATURE

Mr. D . VIJAY ANAND,
MCA.,ME.,

SIGNATURE

Dr.S.ANANDAMURUGAN,
M.E.,Ph.D.,

SUPERVISOR

Assistant Professor
Department of IT,
Kongu Engineering College,
Perundurai

HEAD OF THE DEPARTMENT

Professor & HOD,
Department of IT,
Kongu Engineering College,
Perundurai

INTEGRATING PROMETHEUS WITH GRAFANA FOR MONITORING

AIM

The aim of this project is to develop a reliable and real-time monitoring system by integrating Prometheus with Grafana. It focuses on collecting system metrics such as CPU, memory, and disk usage from an Ubuntu virtual machine using Node Exporter. The collected data is visualized using Grafana dashboards to provide meaningful insights. The setup ensures continuous monitoring by using Systemd to run Prometheus and Node Exporter in the background. Nano is used as a simple text editor to modify these configuration files easily.

ABSTRACT

Monitoring system performance is vital for ensuring high availability and addressing issues before they impact operations. This project showcases the integration of Prometheus and Grafana to create a robust monitoring solution. Prometheus is employed to collect and store time-series data from an Ubuntu virtual machine, with Node Exporter offering in-depth system-level metrics. Grafana then connects to Prometheus, transforming this raw data into visually engaging and interactive dashboards. Systemd is utilized to manage the services, ensuring they run consistently in the background. Configuration is handled via YAML files, providing a clear and organized structure for setup. Nano is used for quick, efficient editing of these configuration files. The entire monitoring stack is deployed on an Ubuntu VM, offering portability and ease of installation. This solution enables real-time monitoring and visualization of system health, enhancing the ability to track, analyze, and troubleshoot system performance effectively.

SCOPE AND OBJECTIVES

SCOPE:

- Focuses on setting up a real-time system monitoring solution using Prometheus, Node Exporter, and Grafana on an Ubuntu virtual machine.
- Walks through the entire process of gathering system metrics, configuring Prometheus, and visualizing data with Grafana dashboards.
- Demonstrates how to monitor essential system metrics like CPU usage, memory usage, and disk space in real time.
- Enhances understanding of how system monitoring tools can be integrated to provide better performance insights and visibility.
- Highlights the importance of data visualization, continuous monitoring, and an easy setup process using lightweight tools and simple configurations.

OBJECTIVES :

- To install and configure Prometheus for gathering system-level metrics like CPU usage, memory consumption, and disk space.
- To integrate Node Exporter with Prometheus for exposing system metrics from the Ubuntu virtual machine.
- To visualize the collected metrics in Grafana using intuitive, user-friendly charts and graphs.
- To ensure Prometheus and Node Exporter run continuously in the background by managing them with Systemd.
- To manage Prometheus configuration using YAML files for clarity and reusability.
- To edit configuration files efficiently using the Nano text editor.

HARDWARE AND SOFTWARE REQUIREMENT

Hardware Requirements:

- Internet connection to install tools and send alerts
- Minimum 4 GB RAM (8 GB recommended for better performance)
- At least 1 GB of free storage space
- A screen size of 13" or larger with at least 1366×768 resolution

Software Requirements:

- **Ubuntu Virtual Machine** – To host and run Prometheus, Node Exporter, and Grafana.
- **Prometheus** – For collecting and storing system metrics.
- **Grafana** – For visualizing the metrics collected by Prometheus.
- **Node Exporter** – To expose system-level metrics (CPU, memory, disk usage).
- **Systemd** – For managing Prometheus and Node Exporter as background services.
- **Nano (or any text editor)** – For editing configuration files (YAML format).
- **Web Browser (Chrome, Firefox, or Edge)** – To access and interact with the Grafana dashboard interface.

PROJECT DESIGN

- The project sets up a monitoring environment on an Ubuntu virtual machine (VM) using Prometheus and Grafana.
- Prometheus is configured to scrape metrics exposed by Node Exporter installed on the same VM.
- Configuration files are written in YAML format for structured and readable setup.
- Grafana is connected to Prometheus as a data source and used to create customizable dashboards.
- Nano or any terminal-based text editor is used to modify configuration files directly within the VM.
- The final outcome is a fully functional, real-time system monitoring solution that can be extended or adapted for larger infrastructure.

ADVANTAGES

- Real-time monitoring of CPU, memory, and disk performance.
- Visual dashboards in Grafana.
- Option to add alerts for critical metrics.
- Open-source and free tools.
- Runs efficiently in the background.
- Easy setup on local or cloud systems.
- Scalable and customizable.

PROPOSED SYSTEM

- Uses **Prometheus** to collect and store time-series system metrics from an Ubuntu-based virtual machine.
- Implements **Node Exporter** to expose system-level metrics like CPU, memory, and disk usage to Prometheus.
- Uses **YAML configuration files** to define Prometheus scraping targets and settings, enabling easy and repeatable setup.
- Utilizes **Systemd** to run Prometheus and Node Exporter as background services, ensuring continuous and reliable monitoring.
- Integrates **Grafana** as a visualization tool to create user-friendly and customizable dashboards.
- Supports **real-time monitoring** and scalable design, allowing the system to be extended to cloud or multi-node environments.

IMPLEMENTATION OF THE PROJECT

The project began by setting up an Ubuntu virtual machine as the host environment. Prometheus was installed and configured using YAML files to collect metrics from Node Exporter, which was also set up on the same system. Node Exporter provided critical system metrics such as CPU, memory, and disk usage. To ensure that Prometheus and Node Exporter ran automatically in the background, Systemd service files were created. Grafana was then installed and connected to Prometheus as a data source. Custom dashboards were created in Grafana to visualize the collected metrics in real-time. The entire setup was tested and fine-tuned using lightweight tools like Nano for editing configuration files. The end result is a secure, efficient, and fully functional monitoring system designed for real-time tracking of system health.

OUTPUT

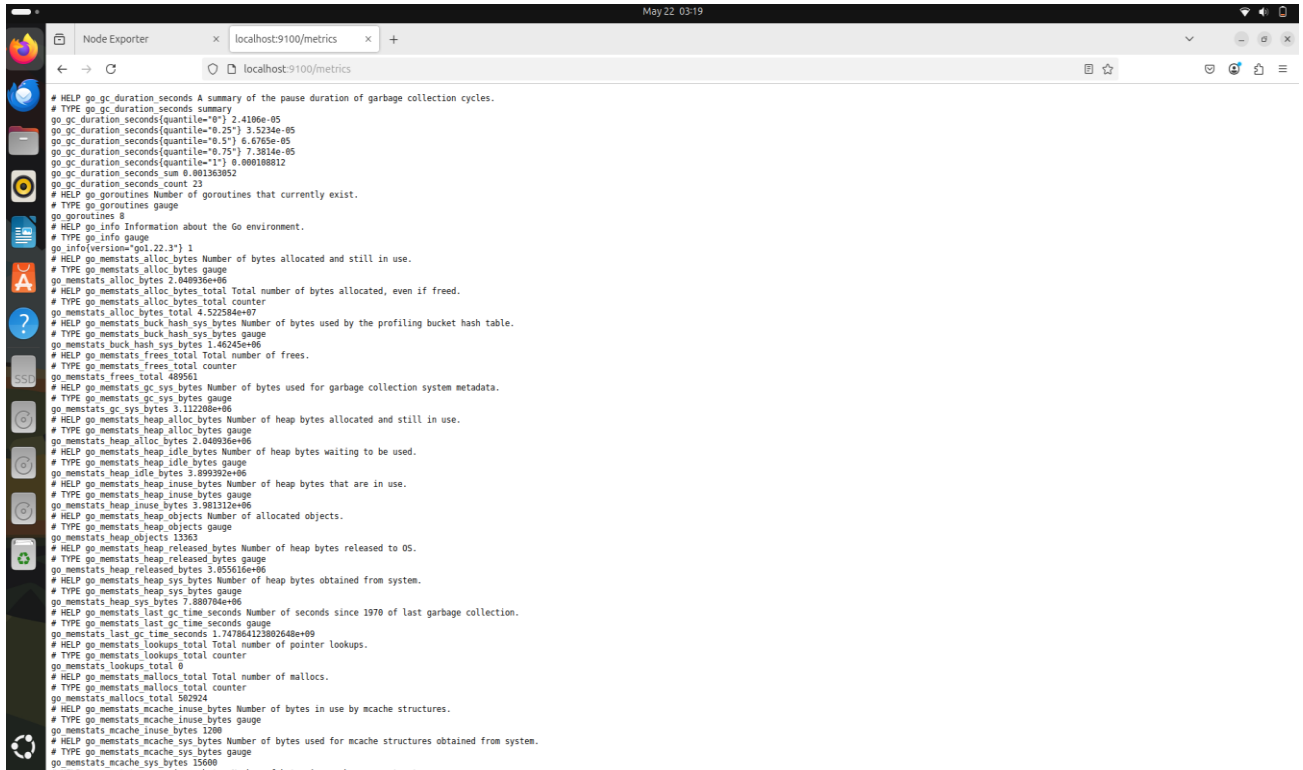


Figure:1

As shown in the Figure 1 the metrics endpoint in Node Exporter provides real-time system and runtime stats in Prometheus format. It includes metrics like memory usage, garbage collection time, and runtime. These metrics are scrapped by Prometheus for monitoring, alerting, and visualization. They help track system health, diagnose performance issues, and optimize resource usage.

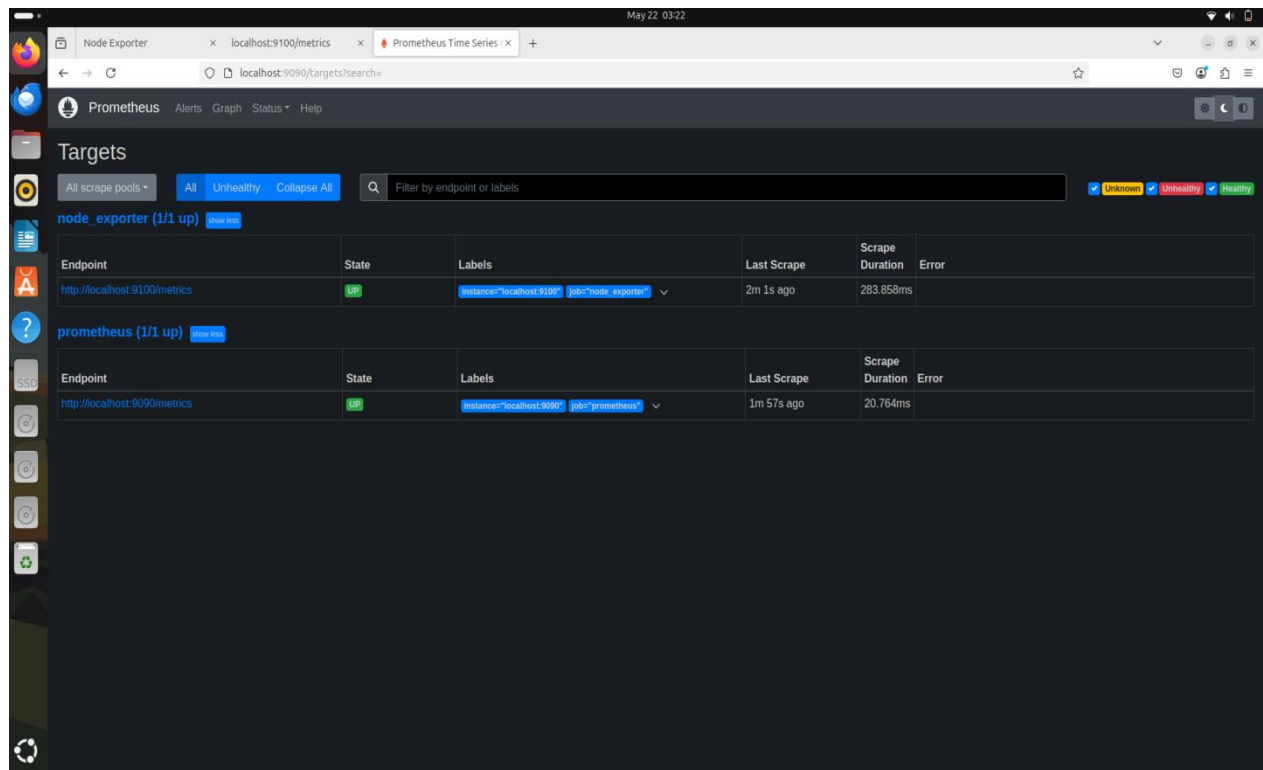


FIGURE : 2

As shown in the Figure 2 this Prometheus Targets page shows that both Node Exporter and Prometheus itself are being monitored. Each target exposes metrics via an HTTP endpoint (/metrics) that Prometheus scrapes at regular intervals. The State "UP" confirms both endpoints are healthy and being successfully scraped. Scrape Duration tells how long it took to collect metrics, useful for performance tuning.

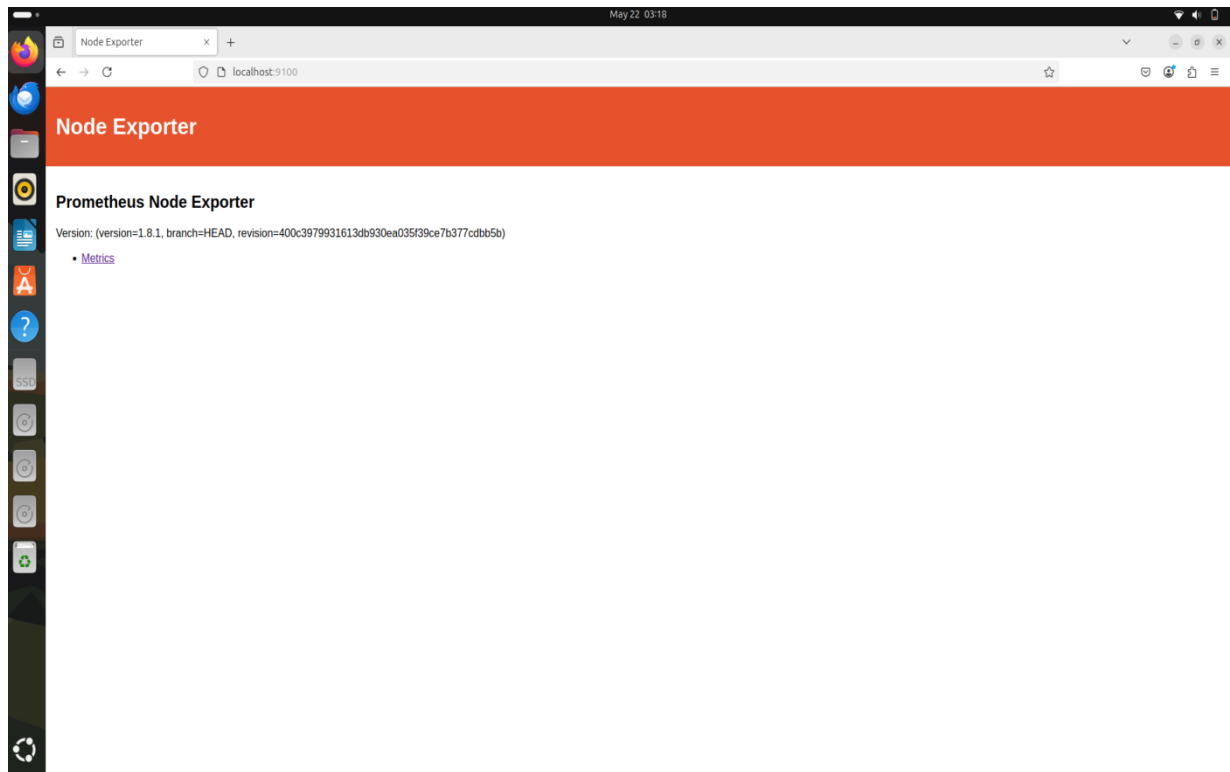


Figure : 3

As shown in the Figure 3 this is the Node Exporter web interface running on localhost:9100. It provides system-level metrics (CPU, memory, disk, etc.) for Prometheus to scrape. The version is 1.7.0, and clicking on "Metrics" shows all available system metrics. This confirms Node Exporter is correctly installed and serving data.

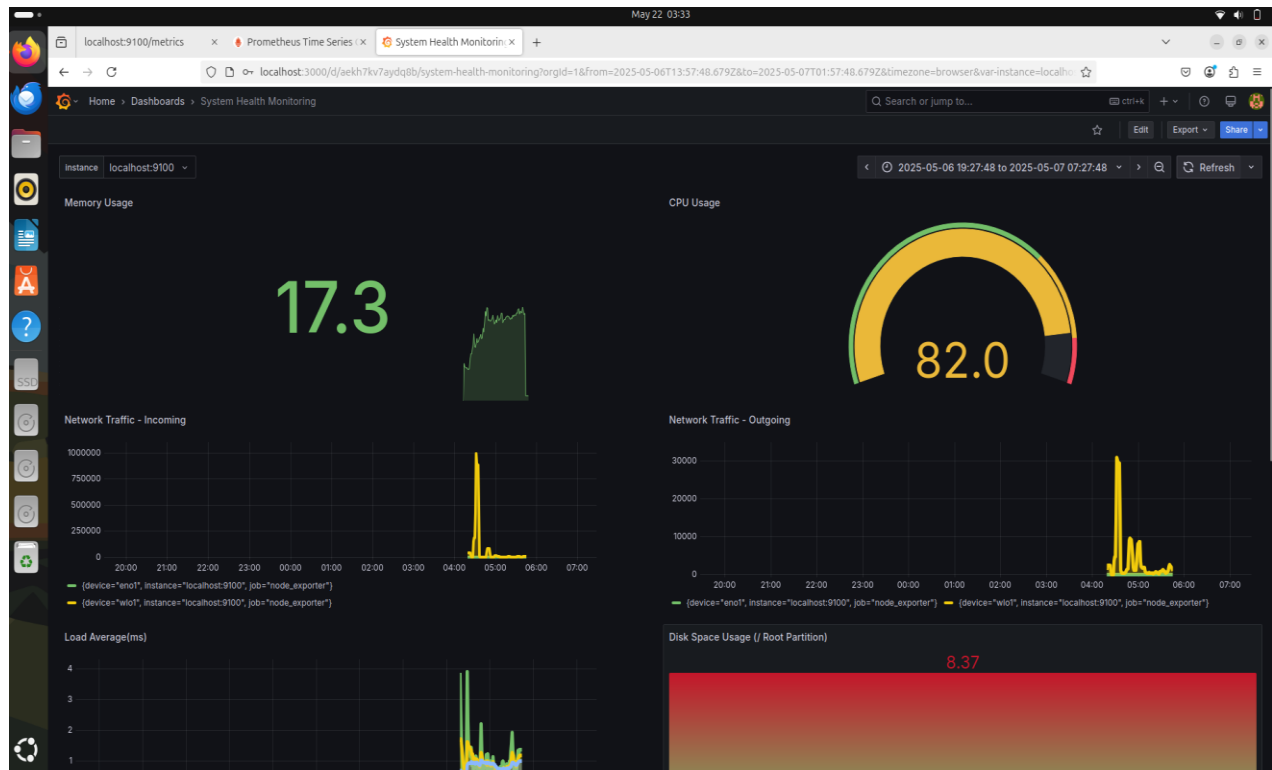


Figure : 4

As shown in the Figure 4 is Grafana that provides a web-based UI to monitor system metrics such as CPU, memory, disk, and network usage. It connects to Prometheus as the data source to query real- time metrics collected by Node Exporter. The dashboard can be customized, refreshed at intervals, and filtered by instance, job, or node to monitor specific targets.

CONCLUSION & FUTURE WORK

Integrating Prometheus and Grafana creates a robust solution for monitoring and analyzing system performance. Prometheus collects time-series data, while Grafana provides intuitive, interactive dashboards for visualizing these metrics. Using Node Exporter, detailed system-level data is captured on an Ubuntu virtual machine, and systemd ensures these services run continuously in the background. The setup is efficiently managed through YAML configuration files and edited using Nano. This solution enables real-time monitoring, simplifies troubleshooting, and enhances overall system performance and uptime. In future work, implementing alerting management in Grafana will further improve reliability by automatically notifying administrators of critical issues, enabling faster response and better system resilience.