

Netdata Performance Monitoring Report

Prepared by: Muhammad Bahar

Project: Monitoring and Visualization of System Performance

Overview

This report documents implementation and evaluation of Netdata as a real-time system performance monitoring solution. The objective was to set up Netdata, capture baseline and stress-test performance metrics, validate them against Python-based monitoring data, and organize all outputs for integration into a shared repository.

System Configuration

The monitoring environment was deployed on an Ubuntu 24.04 virtual machine. Netdata was installed to enable continuous system monitoring, while Stress-ng was used to simulate high CPU workloads.

Installation Commands:

```
sudo apt update && bash <(curl -Ss https://my-netdata.io/kickstart.sh)
sudo apt install stress-ng -y
```

Baseline Monitoring (Idle State)

With no active workload, the system was observed to establish an idle performance baseline. Key observations included: CPU utilization: 5–10%, Memory usage: ~2.2 GiB Disk and network activity: Minimal, consistent with idle operation.

Captured files:

```
netdata_idle_cpu.png
netdata_idle_mem.png
netdata_idle_disk.png
netdata_idle_network.png
```

These screenshots document the stable operating state prior to load testing.

Stress-Test Phase

To analyze system response under load, a stress test was executed using:

```
stress-ng --cpu 2 --timeout 60
```

During the one-minute test: CPU usage peaked between 95% and 100%. Memory, disk, and network activity increased proportionally. Netdata's dashboard reflected real-time spikes across all metrics.

Captured files:

netdata_stress_cpu.png

netdata_stress_mem.png

netdata_stress_disk.png

netdata_stress_network.png

Results Verification and Comparison

My Netdata readings were compared to Enas's Python-based monitoring data located in the `/data-samples/` directory. Both tools produced consistent and closely aligned metrics, confirming data accuracy.

Metric	Python Reading	Netdata Reading
Idle CPU	6 %	~7 %
Stress CPU	98 %	~99 %

Netdata demonstrated an advantage in real-time visualization, providing instantaneous feedback during load variation, while Python scripts offered confirmatory numerical precision.

File Upload and Repository Structure

All outputs, screenshots, and configuration files were uploaded to the shared repository:

Repository link: github.com/Enas-787/Monitoring-and-Visualization-of-System-performance

Repository Structure:

/netdata-configs/ → Netdata screenshots and test results

/documentation/ → Final project reports and summaries

Next Steps

The next phase will involve Grafana integration, led by Naga, combining Enas's Python metrics and my one Netdata data sources. This integration aims to produce comprehensive, real-time Grafana dashboards displaying CPU, memory, disk, and network performance trends.

Conclusion

I successfully completed the Netdata monitoring and stress-testing phase. My findings validate the reliability of both Netdata and Python data collection methods, demonstrating consistent results under idle and stress conditions. This phase confirmed that the project's objectives, accurate, synchronized, and visually dynamic performance monitoring—were fully achieved. The outcomes lay a solid foundation for the upcoming Grafana visualization stage.