Simple System Monitoring with Netdata

B. Adhitya (Roll Number: 24P35A1203)
Ch. Rishitha Chowdary (Roll Number: 24P35A1204)
I. Shanthi (Roll Number: 24P35A0550)
S. Neelima (Roll Number: 24P35A0551)
Aditya College of Engineering and Technology

Abstract

For businesses and developers, exceptional system performance along with reliability are quintessential requisites. These essentials have to be fulfilled for an organization to sustain itself within this rapidly changing digital ecosystem. Our project “Simple System Monitoring with Netdata“ seeks to leverage the power of Netdata, a robust open source monitoring solution, by providing real time system monitoring with extreme flexibility and responsiveness.

The main focus of the project is to guide users on how to install, configure and fully utilize Netdata in monitoring servers. This includes all important metrics such as CPU usage over time, memory consumption, Disk I/O latencies and throughput counters, network data traffic swapped off onboard interfaces as well as overall system health metrics. System health tracking of netdata provides comprehensive dashboards wherein these factors can be analyzed visually for better understanding.

I implemented this on an Amazon EC2 instance running Linux. For my use case, I employed netdata agents which I installed using official one line installers so setup was very smooth easily post-integration. After that step is done I linked the server with netdata cloud which enabled centralized monitoring from anywhere via secure websites. The increase in access monitors add additional benefits especially when multi-user colloborations necessitating distance are involved.

Also tested were the commands top and stress --cpu 4 to simulate

TABLE OF CONTENTS

1.Introduction

1.1 The Importance and Necessity of System Monitoring Today, cloud computing, distributed systems, and microservices have made performing routine system maintenance to validate the health, availability, and performance of IT infrastructure a necessity. Monitoring has evolved from a manual process that occurs every so often to automated and smart monitoring that provides constant insight into any layer of the technology stack with granularity designed in such a way that alerts are generated without adding significant overhead. Proactive monitoring reduces maintenance costs, prevents downtime exposure, detects performance problems and degradation before it has impacted end-users, and provides managers with critical insights to administration, management, outsourcing, and resource allocation decisions.

1.2 What is Netdata? Netdata is an free, open-source monitoring agent that is designed to collect a large number of different metrics in real-time. Netdata has become highly sought after for their low overhead, easy installation of the agent, and the self-discovery aspects of the agent. Unlike monitoring solutions that impose time or latency requirements on the data reporting the collected metrics, Netdata real-time monitoring boasts the capacity to log thousands of metrics every second collecting the metrics and visualizing them immediately on a rich, interactive web dashboard. These metrics and that granularity make Netdata an exceptional tool remaining active or when confronting live performance issues in the production environment.

1.3 Goals The primary goal of the project is to deploy and evaluate Netdata as a simple and powerful system monitoring solution. The project seeks to: 1. Successfully deploy the Netdata agent on a cloud server (Amazon EC2). 2. Explore and evaluate the real time visualization capabilities of the Netdata dashboard. 3. Configure agent and utilized Netdata Cloud for centralized management. 4. Illustrate the utility of the tool by triggering system load and consequently observing the metrics.

2. Project Methodology and System Architecture

2.1 Platform Choice: Amazon Web Services (AWS) The hosting platform chosen for our project is Amazon Web Services (AWS) (EC2). We choose Amazon EC2 (Elastic Compute Cloud) for its proven scalability, reliability, and popularity in industry. When using Amazon EC2, an Amazon Linux EC2 instance was ideal since it is widely used, stable, and a secure and representative environment for a production server.

2.2 Monitoring Tool: Netdata Netdata was chosen for its flexibility in: Real-Time Data: The ability to collect and present metrics with a granularity of once-a-second for a level of insight that no other tool can currently provide Ease of Use: A one line installation followed by zero- configuration Comprehensive: Automatic discovery of services and applications being monitored along with the ability to, by default, collect over 2000 metrics with no configuration or manual entry Low Footprint: Where Netdata is highly optimized to run on any system with no impact to the normal or primary usage.

2.3 Architecture Overview The architectural approach is straight forward: A Netdata agent is installed on the specified target Amazon EC2 instance and runs as a daemon that continuously collects metrics, then that aggregated data is presented using a web server which is also run as, and by, Netdata and available on port 19999. The Netdata agent (or "node") is subsequently connected to the Netdata Cloud platform for centralized monitoring.

3. Implementation and Setup

3.1 Step 1: Environment Preparation (Pre-Requisites)

Before installation, in the process of preparing the environment I Started an Amazon Linux EC2 instance on AWS Opened the instance's Security Group to allow inbound traffic on TCP port 19999 (for the Netdata dashboard) and TCP port 22 (for ssh access). Connected to the instance using SSH to get command line access.

3.2 Step 2: Install Netdata

The kickstart.sh script provided with Netdata installs the software with no issues. I ran the following commands in the EC2 instance terminal:It is recommended/determined to use bash.

```
# Download the official installer script

wget https://my-netdata.io/kickstart.sh

# Make the script executable

chmod +x kickstart.sh
```

Run the installer as a root user

sudo ./kickstart.sh

The script automatically detected which operating system I had deployed (Amazon Linux), installed any dependencies for Netdata, compiled the Netdata program, and auto-configured it to start when the system started.

3.3 Step 3: How to Access the Monitoring Dashboard

After completing the installation, the Netdata monitoring dashboard was made instantly available by navigating to the following URL in the web browser.

http://<your_server_public_ip>:19999

The dashboard immediately displayed hundreds of interactive charts to monitor every aspect of my server in real-time.

4. Configuration and Cloud Integration

4.1 Local Configuration and Customization While Netdata is pretty cool right off the bat, there are tons of configurations that you can make. Everything is controlled inotus pursuant to data collection plugin exclusions, health alert configuration, history and retention configuration, dashboard security via user authentication, and more, in the netdata.conf file, which can usually be found at /etc/netdata/netdata.conf.

4.2 Central proprietary cloud Management If you want centralized cloud based monitoring of multiple servers or to track monitoring results without exposing the port directly from multiple servers, then Netdata Cloud is the way to go. The whole integration process was simple uruhit. This_visit by configuring an account at netdatas cloud official website (app.netdata.cloud). From the cloud interface I created a command to connect Louli to my local agent (our EC2 instance) at the transit and execute the command on it to securely link it to my cloud account. Now, I can view all server metrics from the same unified interface anywhereand access it from anywhere on the planet.

5. System Testing and Visualization

5.1 Simulating System Load In order to test the promise of real time monitoring from Netdata, we intentionally loaded the server's resources with our stress utility. We used the stress utility to generate a large CPU load: Generated bash # this command forks 4 worker processes performing CPU load. sudo yum install stress -y # install the utility if it is not present stress --cpu 4 Use the code with caution. Bash While this was happening we monitored the Netdata dashboard. The CPU Usage and Load Average metrics showed an immediate and significant spike measuring exactly what we were doing on the server.

This command spawns 4 worker processes that perform CPU-intensive calculations.

sudo yum install stress -y # Install the utility if not present

stress --cpu 4

5.2 Reviewing Real-Time Visualizations The Netdata dashboard has all this information and more in visualized chart's: CPU Usage: an aggregate of CPU time spent on user processes, on system processes, I/O wait, etc. Memory: an aggregate of used memory, free memory, memory used for cache and memory used as buffers. Disk I/O: throughput for reads and writes onto disk. Network: inbound and outbound throughput for any and all interfaces on the server.

5.3 Review the Dashboard Interface The dashboard is also logically partitioned into tabs, including Home, Nodes (in Netdata Cloud), Metrics, Alerts, and Anomalies, so you can move between high-level overview navigational menus and detailed specific metrics.

6. Key Use Cases and Benefits

6. Major Use Cases and Value The deployment of Netdata can provide real value across a number of channels: Real-Time Server Performance Monitoring: System administrators are able to view important metrics for Linux servers, databases, and web servers, in real-time, giving them the ability to diagnose and fix issues at the time of the event. Proactive Alerts and Anomaly Detection: The health monitoring and alarms engine can quickly detect serious events such as high CPU usage, low disk space, or critical network failures and health alerts to take action before a complete failure happens. Security Auditing: The ability to monitor running processes, network connections, and system calls can help spot problems with unauthorized access, malware activity, and other anomalous activity that can

indicate a possible security breach. Improved Forecasting: Identifying and analyzing historical trends in resource utilization enables organizations to make data-driven decisions on future infrastructure requirements, like when it is time to replace a server or when they need to add additional capacity.

7. Conclusion and Future Scope

7.1 Final Thoughts This project has successfully shown the use of Netdata as a simple, effective, and highly detailed system monitoring solution. For developers, system administrators, and DevOps engineers, the tool's installation simplicity, "no-configuration" defaults, hyper-granular real-time cloud-integrated dashboards and automatic cloud integration are remarkable advantages. We have demonstrated its capability to give actionable insights on the performance and health of a cloud server which were the primary goals of this project.

7.2 Potential Improvements Although this project covers essential concepts, there is room for improvement in the features and functionalities provided by Netdata. Some future improvements entail: Advanced Custom Alerting: Setting custom health alerts with finely graded thresholds tailored to our application's workload.Third Party Integrations: Connect Netdata's alerts to notification systems such as Slack, PagerDuty or Telegram for easier incident response workflows.Custom Dashboards: Creating custom dashboards that display consolidated metrics from multiple services (a webserver alongside a database and the OS) for comprehensive application oversight.Metric Correlation: Deploying the Netdata Cloud capabilities to correlate anomalous metrics across servers for enhanced insight into interdependencies.