Network Monitoring Systems

Evaluation & Selection of Network Monitoring Software

Selena Colomo
Mentor: Arjun Shetty

SLAC National Accelerator Laboratory, Menlo Park, CA, USA

GitHub Repository

Overview

Servers across SLAC are used to to store, process and deliver network data. The goal of this project was to evaluate different network monitoring platforms–specifically LibreNMS, Grafana + Prometheus, and Zabbix–to determine the best fit for our environment. Ultimately, delivering a production-ready monitoring platform that provided real-time insight into switches, routers, and network devices with a dashboard and alerting system.

Research & Discovery Phase

Environment Setup

- Installed Homebrew on macOS to manage CUI utility tools dnf on Rocky Linux for package installation
- Deployed Podman on Rocky Linux as a container engineer to host monitoring services, along with Docker Desktop on macOS
- Utilized Bash shell scripting for system configuration, systemd service management, and network interface tuning

Network & Server Configuration

- Connected air-gapped Rocky Linux target server directly to the macOS via physical Ethernet cable
- Manually configured static IPv4 addresses and subnet masks on both interface to establish communication

```
[selenaco@XQVJ3YFKWN ~ % ping -c 4 192.168.99.2
PING 192.168.99.2 (192.168.99.2): 56 data bytes
64 bytes from 192.168.99.2: icmp_seq=0 ttl=64 time=5.084 ms
64 bytes from 192.168.99.2: icmp_seq=1 ttl=64 time=3.447 ms
64 bytes from 192.168.99.2: icmp_seq=2 ttl=64 time=4.370 ms
64 bytes from 192.168.99.2: icmp_seq=3 ttl=64 time=1.918 ms

--- 192.168.99.2 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.918/3.705/5.084/1.184 ms
```

Rocky Linux SNMP

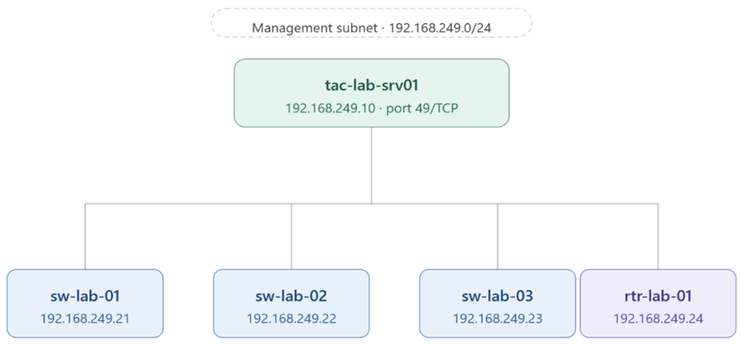
- Installed and enabled net-snmp and net-snmp-utils packages to allow software monitoring tools to poll host system metrics
- Configured a passive snmpd.conf listening profile with access restricted only to the local management subnet for secure polling

```
sudo mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.bak
sudo bash -c 'cat <<EOF > /etc/snmp/snmpd.conf
agentAddress udp:161
rocommunity public 192.168.99.0/24
EOF'
```

System Overview

Physical Hardware & Network Architecture

- Monitoring Server: A dedicated Rocky Linux host serving as the central collection node
- Network TAP: A hardware device inserted into the physical link between the network switch and target traffic flow



Grafana + Prometheus

Grafana and Prometheus are tools that work together to collect real-time metrics and visualize them on interactive dashboards. This combination was chosen as the optimal solution for our network monitoring setup due to its passive, continuous data collection, and its ability to capture sudden bursts of network traffic via the Network TAP. In addition, it has simple data visualization configuration through the use of PromQL that provides clean, easy to read, and interactive data visualizations.

Prometheus (Backend)

- Focuses on data collection by scraping metrics from applications, servers, or exporters (Node Exporter)
- Stores metrics in time-series database and uses PromQL to evaluate alerting

Grafana (Frontend)

- Queries data from prometheus to display the data
- Makes use of interactive visualizations to build dashboards, displaying time-series data

Network TAP (Test Access Point)

The TAP passively copies the raw network traffic passing through the monitored link and sends it via an Ethernet cable into interface eno3 on the Rocky Linux server.

- Configured eno3 to operate in promiscuous mode so the network interface card (NIC) accepts all captured frames

```
sudo ip link set eno3 up
sudo ip link set eno3 promisc on
```

Monitoring Platforms

	LibreNMS	Zabbix	Grafana + Prometheus
Primary Focus	Auto-Discovery & SNMP Topology	Enterprise Host & Alert Management	Time-series collection and data visualizations
Data Gathering	Active SNMP Polling	SNMP / Agent Pull	Scraping time-series from target HTTP (:9100)
Polling Interval	1-5 Minutes	30s - 1 Minute	5 - 15 Seconds
TAP Suitability	Low (Misses micro bursts)	Medium	High

Dashboard Configuration

- Used PromQL queries to produce visualizations of metrics being recorded via Network TAP
  - **Incoming Traffic (RX Bandwidth)-** Confirms mirrored frames from the TAP are actively entering the eno3 interface

```
rate(node_network_receive_bytes_total{device="eno3"}[5m]) * 8
```

  - **Outgoing Traffic (TX Bandwidth)-** Confirms passive monitoring mode

```
rate(node_network_transmit_bytes_total{device="eno3"}[5m]) * 8
```

  - **Live Bandwidth-** Shows real-time ingestion speed of incoming network traffic

```
rate(node_network_receive_bytes_total{device="eno3"}[1m]) * 8
```

  - **Total Data Volume-** Tracks total volume of raw packet data ingested by eno3 over time

```
increase(node_network_receive_bytes_total{device="eno3"}[$__range])
```

  - **Physical Errors-** Monitors physical integrity such as faulty cables or bad TAP signaling

```
rate(node_network_receive_errs_total{device="eno3"}[5m])
```

  - **Dropped Packets-** Tracks buffer overflows

```
rate(node_network_receive_drop_total{device="eno3"}[5m])
```

ACKNOWLEDGEMENTS

I would like to express my gratitude to Arjun Shetty for being a supportive menore throughout this entire project and guiding me through every challenge along the way. I would like to thank Zach Domke for being a great supervisor and providing all the necessary guidance, resources, and support along the way.