

Centralized Observability for Network and Systems

Thong D. Huynh

StudentID: 012882651

Table of Contents

A: Summary	3
B: Review of Other Work	4
B1: Works Supporting Implementation	5
C: Changes to Project Environment	6
D: Methodology	7
E: Project Goals and Objectives	8
F: Project Timeline	9
G: Unanticipated Scope Creep	10
H: Conclusion	10
H1: Success of Project	11
I: Appendices	12
References	16



Post-Implementation Report

A: Summary

The company, Streetrack, operated a Layer-3 collapsed core network that included Cisco Layer-3 switches, a perimeter firewall, and several Linux servers. The infrastructure relied on OSPF routing throughout the network segments for connectivity. Before the implementation of this project, the IT team was managing each device's logs and metrics manually and individually. This approach resulted in long response times to issues regarding performance and system health. Troubleshooting and visibility into the infrastructure needed to be optimized.

To address these challenges, a centralized observability platform was implemented to correlate logs and metrics from devices throughout the company's infrastructure. A monitoring server was deployed to gather telemetry data and visualized in a centralized manner to quickly identify system health and network changes. The logs and metrics were forwarded to the monitoring server using a standardized telemetry agent, while network devices and the firewall were integrated using industry-standard monitoring and logging protocols. To streamline the deployment, Ansible automation was used to create a repeatable process where onboarding future nodes would be consistent and reliable as the company expands.



As a result of the implementation, metrics and logs of all systems and devices were consolidated into a centralized platform with unified dashboards for accurate and real-time monitoring. The observability platform was then validated with simulated load testing and failover scenarios to confirm components were properly configured and showing expected information. Overall, the project improved the company's infrastructure visibility, reduced the reliance on manual monitoring, and allowed the IT team to quickly identify issues and troubleshoot more effectively.

B: Review of Other Work

Work 1: Huntress, an industry cybersecurity company, released an article on centralized logging in modern environments. They stated that centralized logging equals less chaos, faster detection, and stronger cybersecurity. (Danielson, 2025) The core idea is to enable easier searching, monitoring, and analysis. By consolidating the infrastructure's logs and metrics, manual efforts during incidents are reduced, and visibility is greatly improved.

Work 2: A Skedler published article explained that Grafana Alloy is an advanced OpenTelemetry-based collector that unifies the capture of metrics, logs, and traces from various sources. (Quesada, 2024) Grafana Alloy reduces the need for multiple agents and supports both push and pull methods, which simplifies complex configurations for monitored systems.



Work 3: Splunk published an article about Syslog, a standard for centralized logging.

The article explained that Syslog is a standardized protocol for centralizing system and event logs from a wide range of devices and applications. (Siddiqui, 2024) They further mentioned Syslog's layered architecture and its kernel message capabilities. Due to its support for network devices, Syslog can gather telemetry data for potential security breaches as well as system health.

B1: Works Supporting Implementation

Work 1: The Huntress article supported the choice of implementing a centralized log management because it directly addresses the problem of manually reviewing telemetry from each individual device on the network. The implementation enabled the correlation of events across servers, network devices, and the firewall, while minimizing manual workload. The company was also looking for a solution to their slow response times when issues arise, so the centralized monitoring approach improved visibility and allowed the IT team to solve problems effectively and efficiently.

Work 2: Since the company aimed to centralize its observability across the infrastructure, the Skedler article supported the choice of implementing Grafana Alloy because it allowed a simple unified agent to collect telemetry data instead of multiple agents and services. This approach kept the Ansible automation playbooks lean and reliable for onboarding new systems into the monitoring stack.



Work 3: The Splunk article concerning Syslog directly supported the implementation of using this protocol because Cisco network devices use Syslog messages natively. Furthermore, Cisco switches do not support host-based agents like Grafana Alloy, so integrating Syslog into the monitoring stack is a valid choice for retrieving telemetry data from the network devices.

C: Changes to Project Environment

Previously, the company's monitoring approach relied on manual review of logs and metrics from individual devices. The process provided minimal visibility across the infrastructure. Consequently, when issues came about, the IT team had to track down the problem, which was time-consuming and resulted in delayed remediation.

After implementing the new observability platform, device monitoring was correlated into a single visibility system. The IT team was able to identify issues quickly due to the optimized dashboards showing real-time system health and connectivity.

Overall, the project enhanced the company's strategy and aligned with its culture because it allowed the IT team to quickly confirm issues and promptly address them, so business operations remain consistent and reliable. The implementation also supported quick troubleshooting because the platform allowed the correlation of telemetry data across the entire infrastructure.



D: Methodology

This project utilized the SDLC standard life cycle as its methodology to implement the centralized monitoring solution. With this structured approach, we ensured that each phase was easily repeatable and validated.

The planning phase consisted of reviewing the network architecture, devices, and confirming reachability throughout the Layer-3 routed links. Between the firewall, switches, and servers, we ensured that all devices were producing telemetry data before moving forward.

Next was the design phase, where we defined the tools to implement the project. This included identifying how the metrics and logs would be collected and visualized. Using the automation approach, Ansible playbooks and roles were built to install the necessary services on their proper devices. Furthermore, configuration templates were designed to support consistent deployment for easily scalable implementation.

Afterwards, the implementation phase commenced with deploying the services and configurations across the infrastructure. Relying on Ansible automation, this phase ensured changes were reliable and repeatable. The monitoring server had all the required services installed and configured to collect and visualize the data. The firewall, network devices, and servers were also installed and configured with their respective requirements, such as Grafana Alloy, Syslog, and SNMP.



For the testing phase, failover scenarios were simulated to observe the behavior of the monitoring stack to confirm that telemetry data was properly ingested, displayed correctly, and responded as expected. This ensured response times were optimized and issues were quickly identified.

Finally, the maintenance phase included ongoing monitoring of the observability platform to make sure services were running consistently. This phase also supported long-term operations by allowing new systems to be onboarded into the monitoring stack using the standardized automation approach.

E: Project Goals and Objectives

Based on the company's goals and objectives, we can confirm whether the project implementation effectively accomplishes its mission. Before the implementation of the project, one of the company's goals was to improve its visibility of the infrastructure. To do so, the objective was to centralize metrics and logs from the firewall, servers, and network devices. Another objective was to provide a unified view of systems and network health. Both objectives were satisfied after the implementation of the project because test results show that metrics and logs from all devices were present on the monitoring server, and dashboards displayed a unified view of the infrastructure.



F: Project Timeline

The original project timeline was scheduled to be implemented for March 2026. After being approved, the project later moved up on the schedule and was completed between October 18, 2025, and November 29, 2025. Each milestone was successfully completed within the adjusted timeline. The reason why the project timeline was met is that each milestone was clearly identified, and by leveraging Ansible automation, deployments and configurations were streamlined, resulting in a successful process.

Milestone	Duration	Start Date	End Date
Review network topology, confirm OSPF routing, and validate reachability across servers, switches, and firewall	5 Days	10/18/2025	10/22/2025
Define monitoring architecture and develop Ansible playbooks and configuration templates	7 Days	10/23/2025	10/29/2025
Deploy Prometheus, Grafana, and Loki on monitoring server and Alloy on nodes	7 Days	10/30/2025	11/05/2025
Configure metrics, logs, SNMP, and syslog ingestion and Grafana dashboards	7 Days	11/06/2025	11/12/2025
Perform simulated failover testing and validate telemetry ingestion and accurate dashboard displays	7 Days	11/13/2025	11/19/2025
Finalize documentation, review deliverables and test results	10 Days	11/20/2025	11/29/2025



G: Unanticipated Scope Creep

During the design and implementation phases of the project, an unanticipated scope challenge occurred with the configurations of the network switches. Although automation was extensively used throughout the deployment of the project, the configurations of the Layer-3 Cisco switches were not automated as originally expected. The team was required to manually configure and enable telemetry features such as SNMP and Syslog on the switches.

While this limitation created additional effort during the implementation phase, it did not significantly affect the overall project timeline or hinder the monitoring solution from functioning as needed. The required configurations were documented and applied consistently across all switches. The scope creep was managed by limiting the amount of manual work on network switches only, while metrics and logs on the receiving end of the switches were still entirely automated. This ensured that the objectives of the project were still met and optimized.

H: Conclusion

After implementing this project to address the company's goal to improve infrastructure visibility and its incident response, it can be concluded that both goals were met. Previously, system metrics and logs were manually reviewed, causing low visibility and slow response times.



Utilizing Prometheus, Grafana, Loki, Alloy, SNMP, and Syslog, all metrics and logs were successfully aggregated into the monitoring server, providing a centralized view of system health and performance. Additionally, dashboards were optimized to display real-time data to support faster response times when incidents occur.

Overall, the project delivered a reliable and scalable monitoring solution that met all previous concerns and goals regarding the company's observability capabilities.

H1: Success of Project

The success criteria for this project were to achieve a centralized observability platform that collected and correlated metrics and logs from infrastructure devices and improve incident response times.

Based on the testing phase, both criteria were met. Traffic load tests and failover scenarios showed dashboards reacted as expected. Telemetry from all infrastructure devices was successfully ingested into the monitoring solution with unified dashboards for quick confirmation of system health. Additionally, failover tests showed downed interface counters reflected in real time, which supports faster response times as incidents occur.

According to the test results, the project successfully fulfilled the company's objectives by improving infrastructure visibility and incident response.



I: Appendices

Appendix A: Telemetry Design

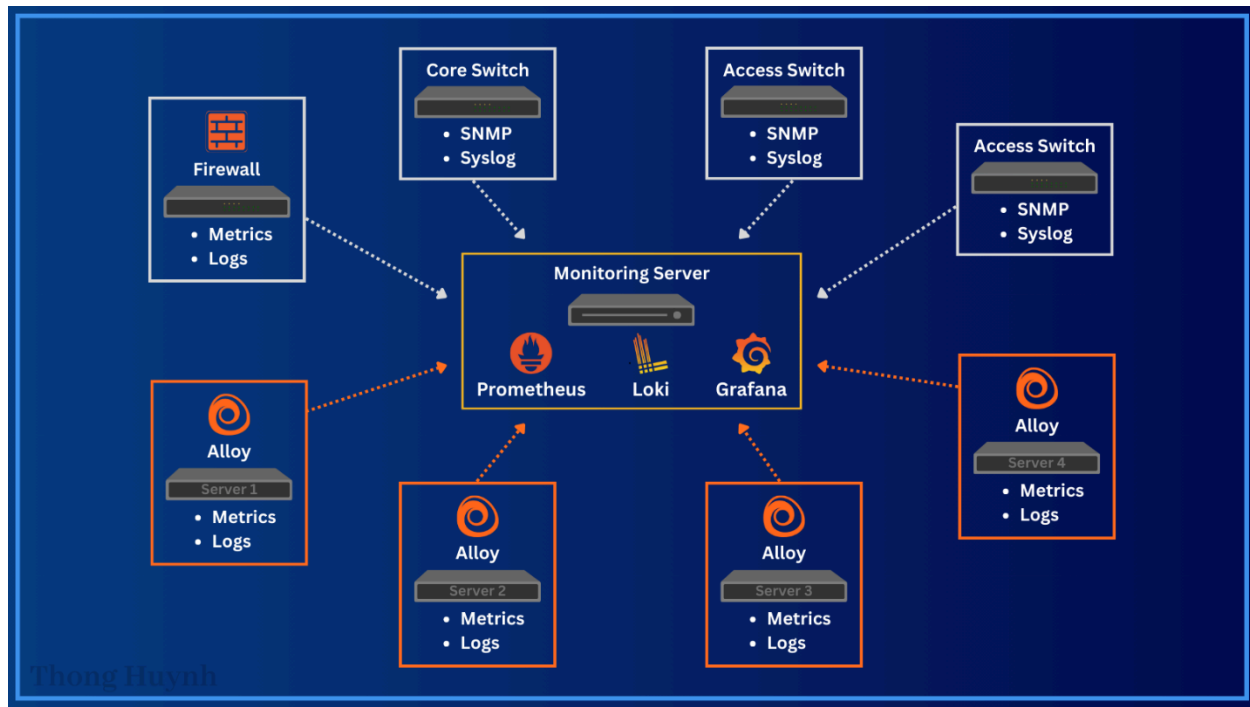


Figure A1: Telemetry Flow and Observability Architecture

Figure A1 depicts the flow of logs and metrics from each device to the dedicated monitoring server. Utilizing Grafana Alloy as a unified telemetry agent, both logs and metrics from servers can be gathered, pushed, and pulled by Loki and Prometheus. Additionally, the core and access switches leverage SNMP and Syslog to send their respective telemetry. The firewall uses its native plugin and logging to send its data as well. Overall, metrics and logs from all devices are aggregated into the centralized monitoring server to support the company's goal of providing a holistic view of the entire infrastructure.



Appendix B: Centralized Observability

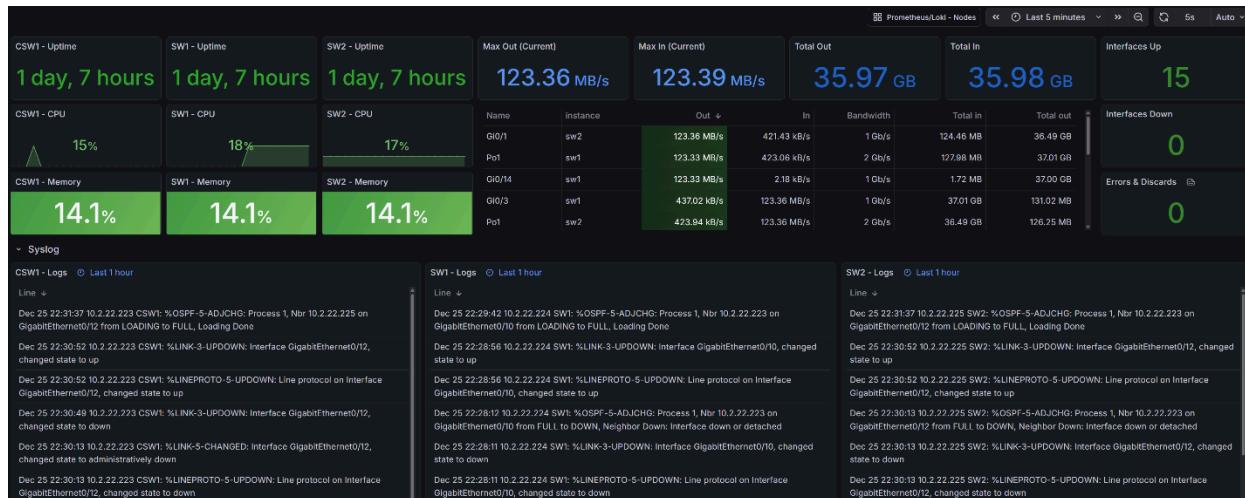


Figure B1: Network Device Metrics and Logs Dashboard

Figure B1 shows a functioning custom dashboard of the Layer-3 collapsed core network switches. It shows all three switches with their metrics such as uptime, memory, CPU utilization, and interface throughput. The lower half of the dashboard provides real-time Syslog messages from each switch regarding logs such as OSPF neighbor updates, logins, and interface status. The consolidated and centralized view demonstrates improved infrastructure visibility of system health and operational status.



Appendix C: Network Failure and Recovery Validation

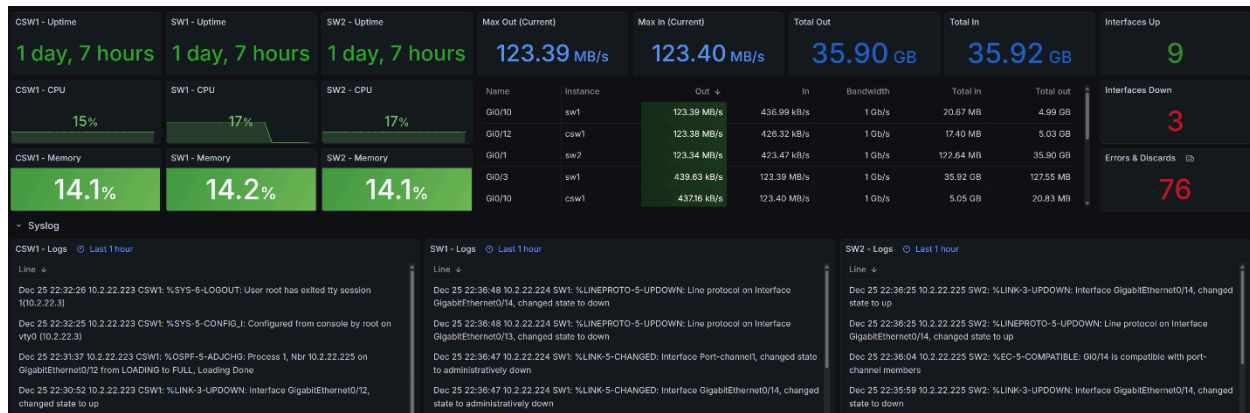


Figure C1: Simulated Network Interface Failure

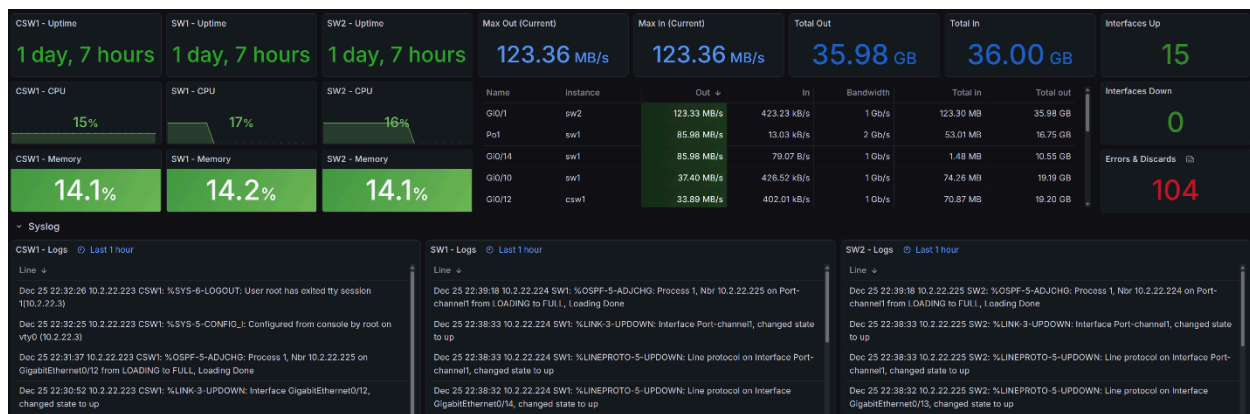


Figure C2: Network Recovery and Traffic Restoration

This appendix covers network failure detection and recovery. To confirm detection, simulated interface failures were implemented to observe the dashboards. Figure C1 shows a reflected number of downed interfaces along with errors while traffic is rerouted through the core switch. Logs from Syslog panels confirmed that a port channel and interfaces are in a downed state. Figure C2 showed that traffic stayed consistent when the interfaces were back up. This failover test demonstrated the functionality of the observability platform and confirmed real-time monitoring effectiveness.



Appendix D: Host and Security Telemetry

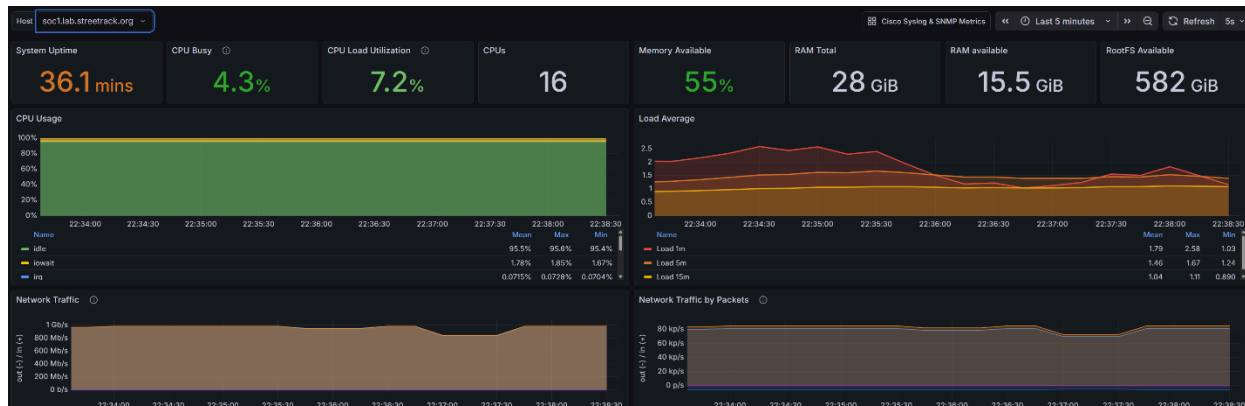


Figure D1: SOC Server System Metrics



Figure D2: Firewall Security and Authentication Logs

Appendix D demonstrates host-level monitoring and security-related telemetry. Figure D1 displays real-time performance metrics from the SOC server, such as CPU, memory, load, and uptime. Figure D2 validates firewall logs regarding SSH failed and successful login events. This confirms metrics and logs are correlated into the platform from the firewall and servers. With all logs and metrics aggregated, both operational monitoring and security awareness are supported by the implementation of the project.



References

Danielson, L. (2025, October 3). *Centralized logging explained: Your guide to modern cybersecurity log management*. Huntress.

<https://www.huntress.com/cybersecurity-101/topic/centralized-logging>

Quesada, P. (2024, April 24). *What is Grafana Alloy? The OpenTelemetry advanced collector*. Skedler.

<https://www.skedler.com/blog/grafana-alloy-an-opentelemetry-advanced-collector/>

Siddiqui, L. (2024, May 28). *What is syslog?* Splunk.

https://www.splunk.com/en_us/blog/learn/syslog.html

