

## Key Takeaways

This lab exposed me to the deployment of Prometheus and Grafana as containerized services such that I was able to visually inspect system metrics in real-time. Using Docker Compose and Docker, I was able to orchestrate the two services with a single command where Prometheus scrapes and stores metrics, and Grafana to render them as an easily accessed dashboard interface.

I also created a custom Prometheus configuration file to define scrape targets and intervals to ensure the correct metrics were being telemetered at periodic intervals. To further streamline lab configuration and allow Prometheus to collect real system level metrics, I had to execute Node Exporter as a service in the docker compose file. Although not originally specified in the instructions, this had to be done in order to enable system metrics such as CPU utilization, available memory, disk I/O, and network traffic to be checked. This was an adding depth in the monitoring stack.

To manage these services efficiently, I configured a docker-compose.yml file. This enabled me to specify and execute Prometheus, Grafana, and Node Exporter as separate services with each of them being assigned their own ports and config. Executing them detached helped ensure long-running with minimal disruption.

After that, I checked that the Prometheus data scraping was working as expected, then proceeded to add it as a data source within Grafana. I went on to create panels and graph some key system measurements using Prometheus queries, switching panel types as well as configuration based on varying data types.