Lesson 02 Demo 04

Writing PromQL Queries to Extract Specific Metrics from a Sample Dataset

Objective: To write PromQL queries for extracting and analyzing specific metrics from a Node Exporter dataset using the Prometheus UI

Tools required: Linux operating system, Docker, and Docker Compose

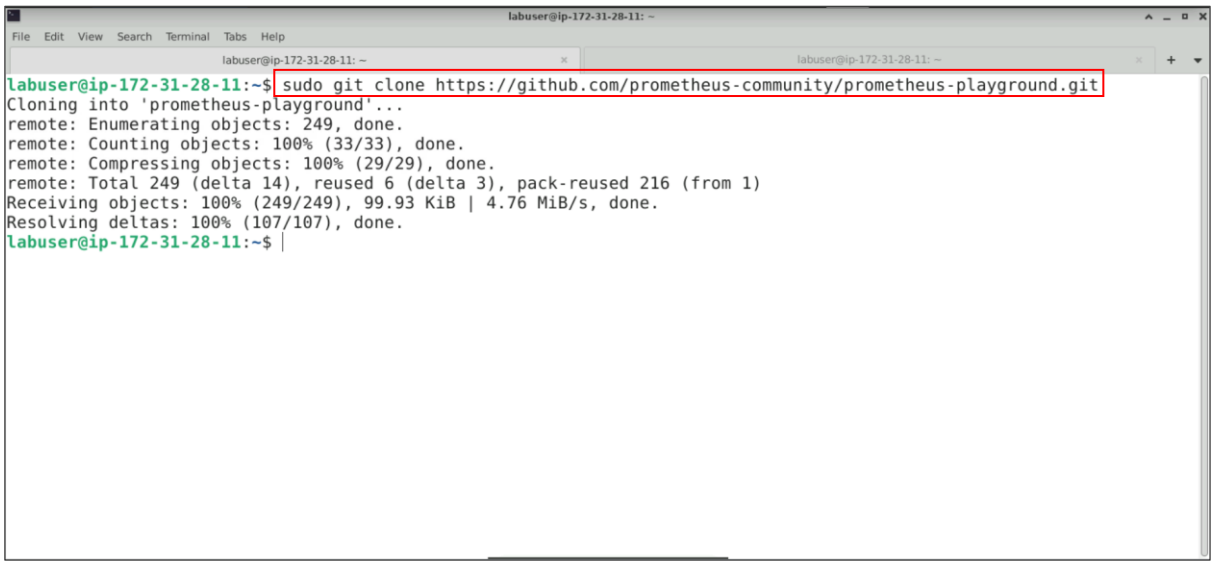
Prerequisites: A basic understanding of the Prometheus UI and PromQL queries

Steps to be followed:

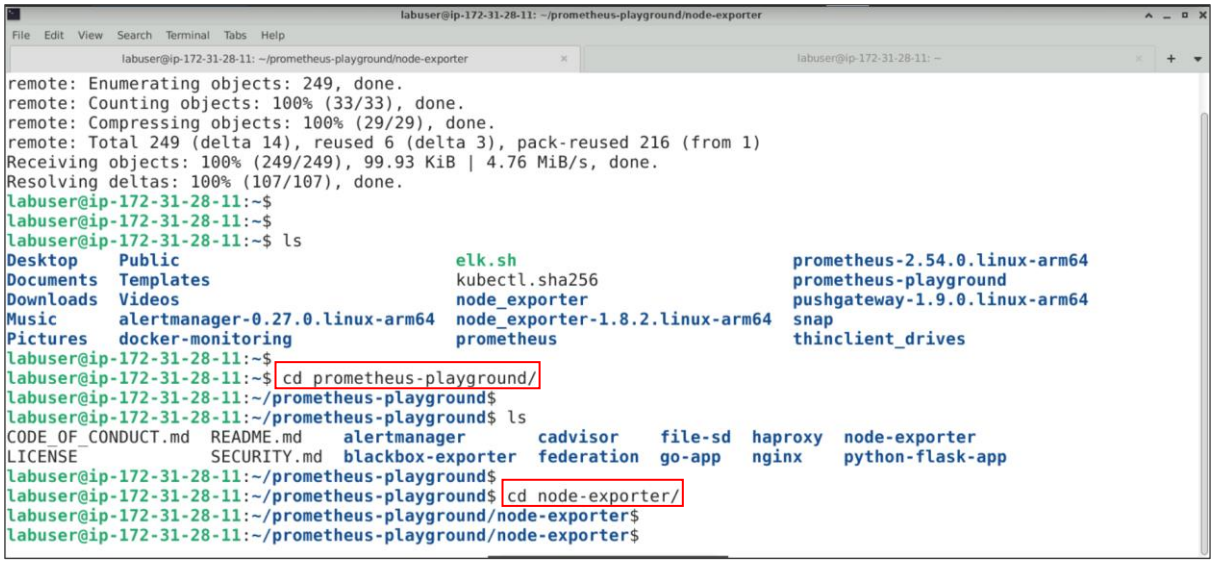
1. Set up Prometheus and Node Exporter using Docker
2. Use the Prometheus UI to query Node Exporter metrics

Step 1: Set up Prometheus and Node Exporter using Docker

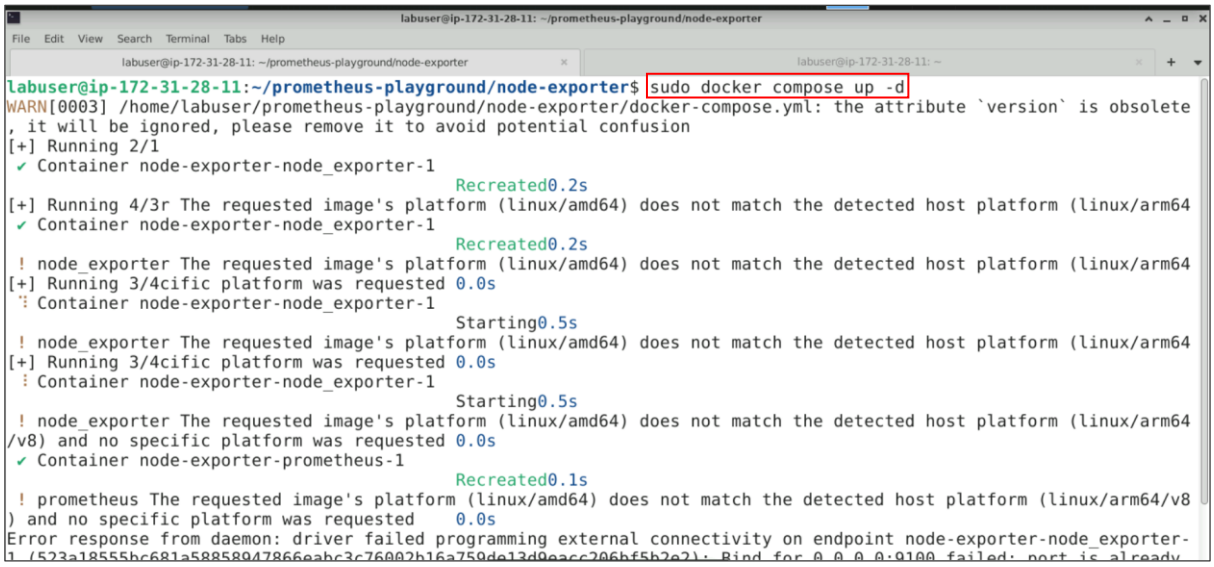
1.1 Open Terminal and clone the **prometheus-playground** repository from GitHub using the following command:
sudo git clone https://github.com/prometheus-community/prometheus-playground.git



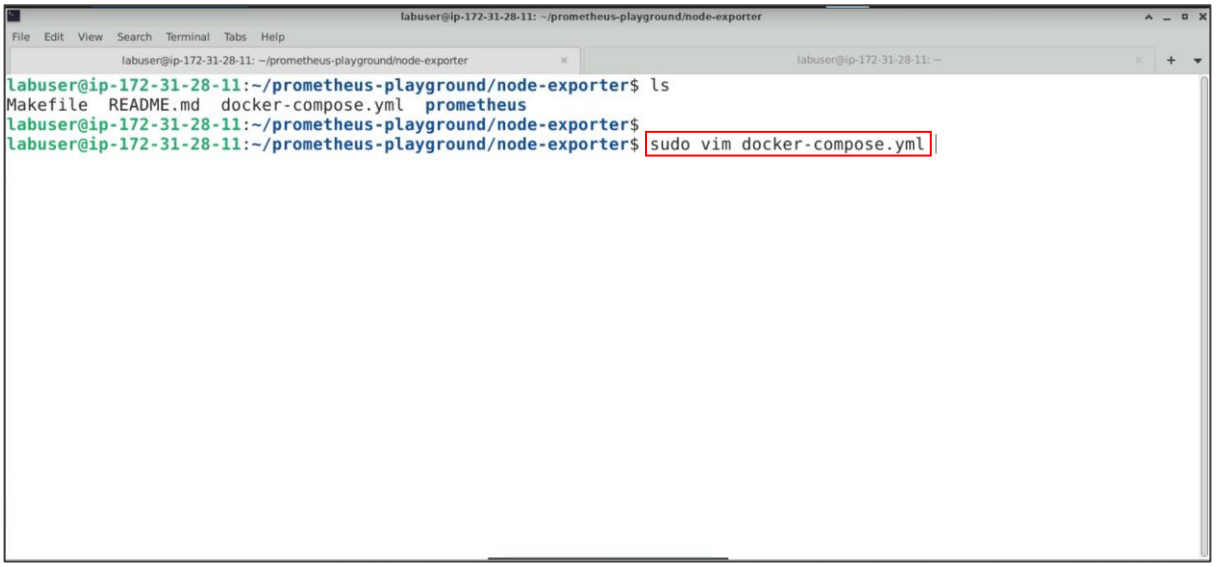
1.2 Change the current directory to the node-exporter folder within the cloned repository using the following commands:
cd prometheus-playground/
cd node-exporter/



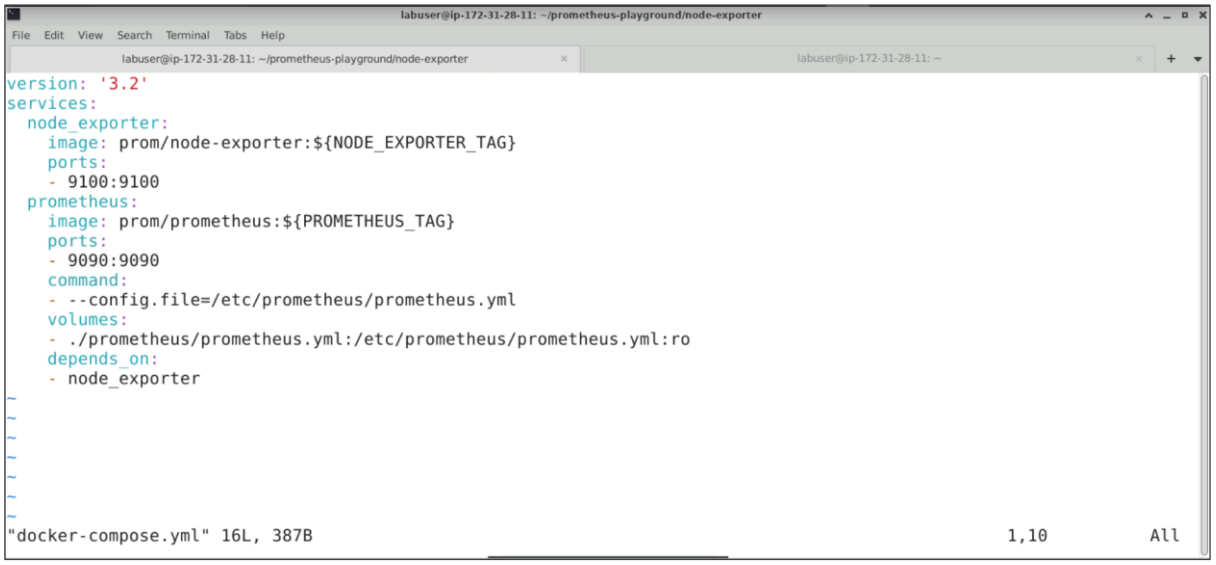
1.3 Start Docker containers in detached mode using the following command:
sudo docker compose up -d



1.4 Open the **docker-compose.yml** file for editing using the following command:
sudo vim docker-compose.yml

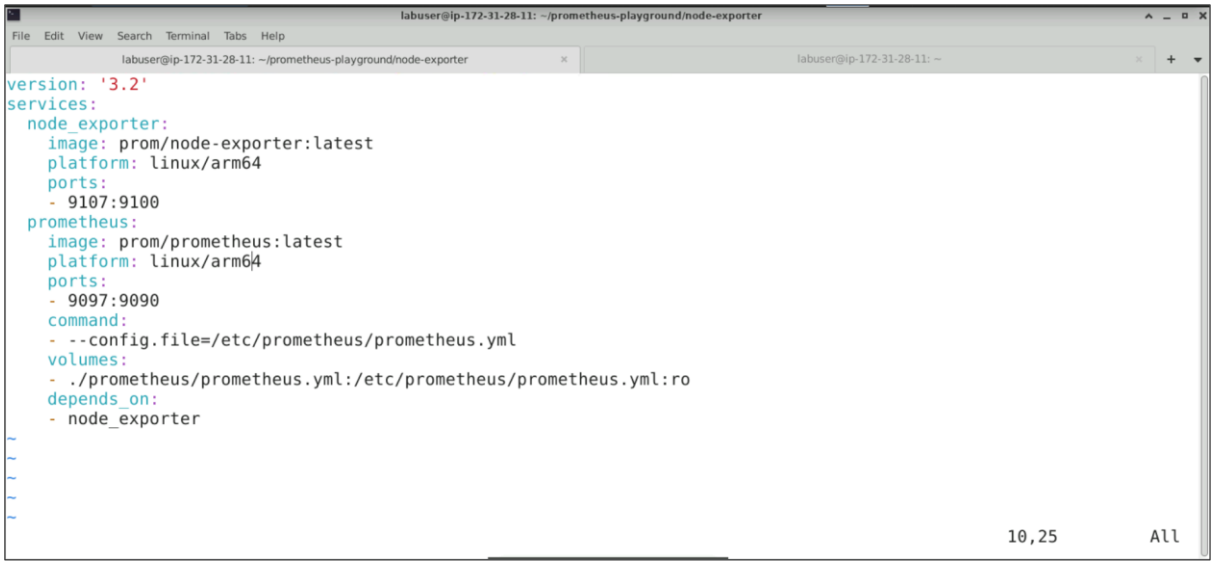


The configuration file should look like this:



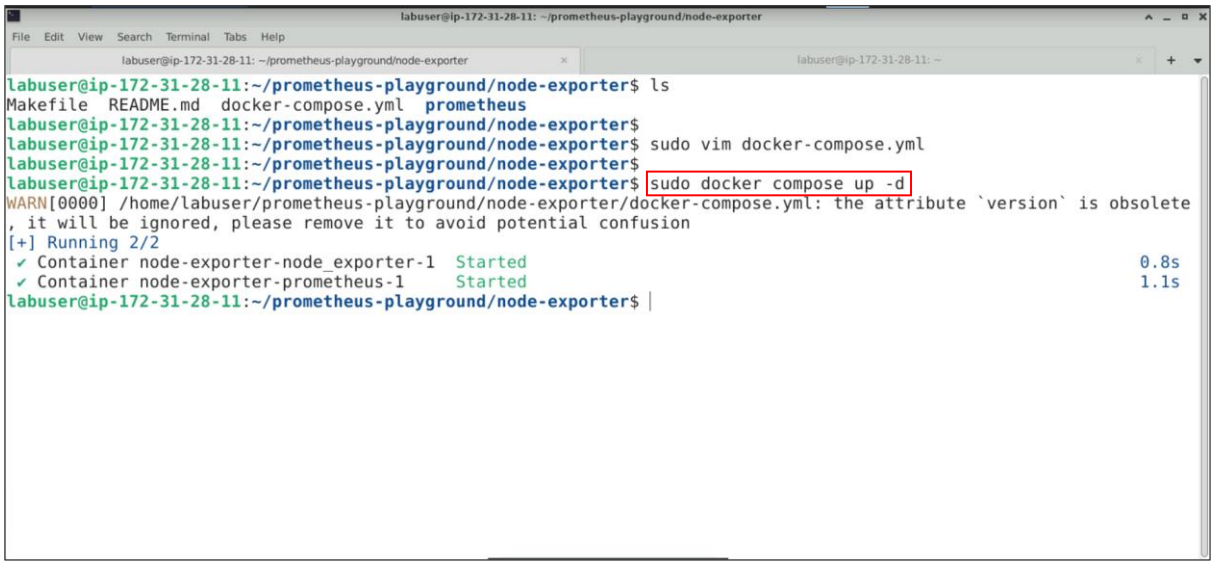
1.5 Copy and paste the following configuration into the file, then save and exit:

```
version: '3.2'
services:
 node_exporter:
  image: prom/node-exporter:latest
  platform: linux/arm64
  ports:
   - "9107:9100"
 prometheus:
  image: prom/prometheus:latest
  platform: linux/arm64
  ports:
   - "9097:9090"
  command:
   - --config.file=/etc/prometheus/prometheus.yml
  volumes:
   - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml:ro
  depends_on:
   - node_exporter
```

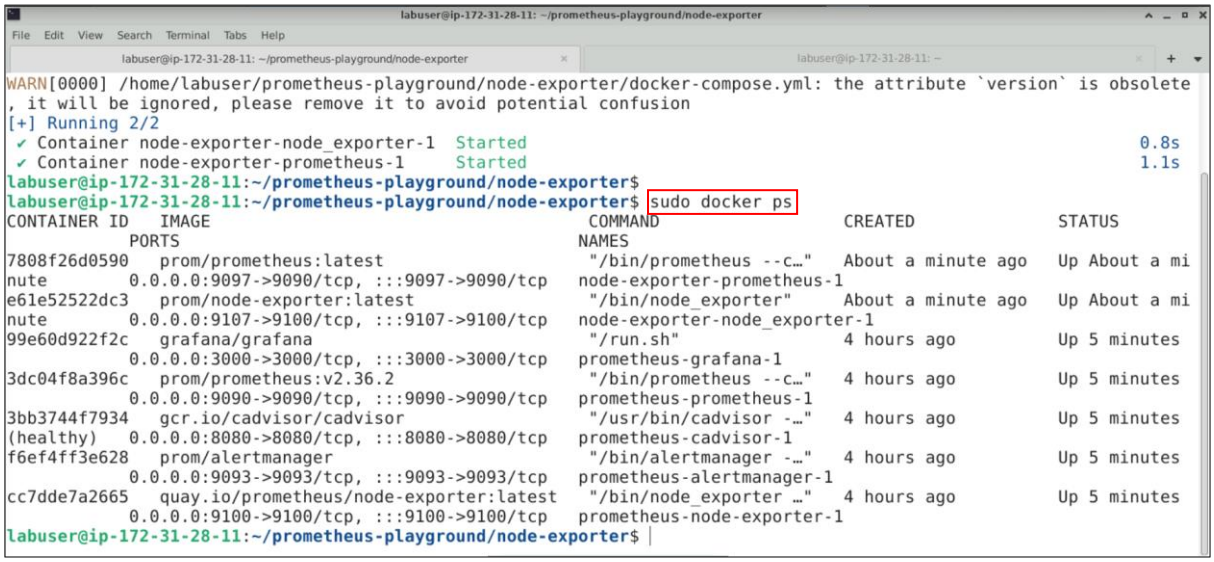


Note: The port number has been changed to avoid frequent port conflicts.

1.6 Start Docker containers using the following command:
sudo docker compose up -d

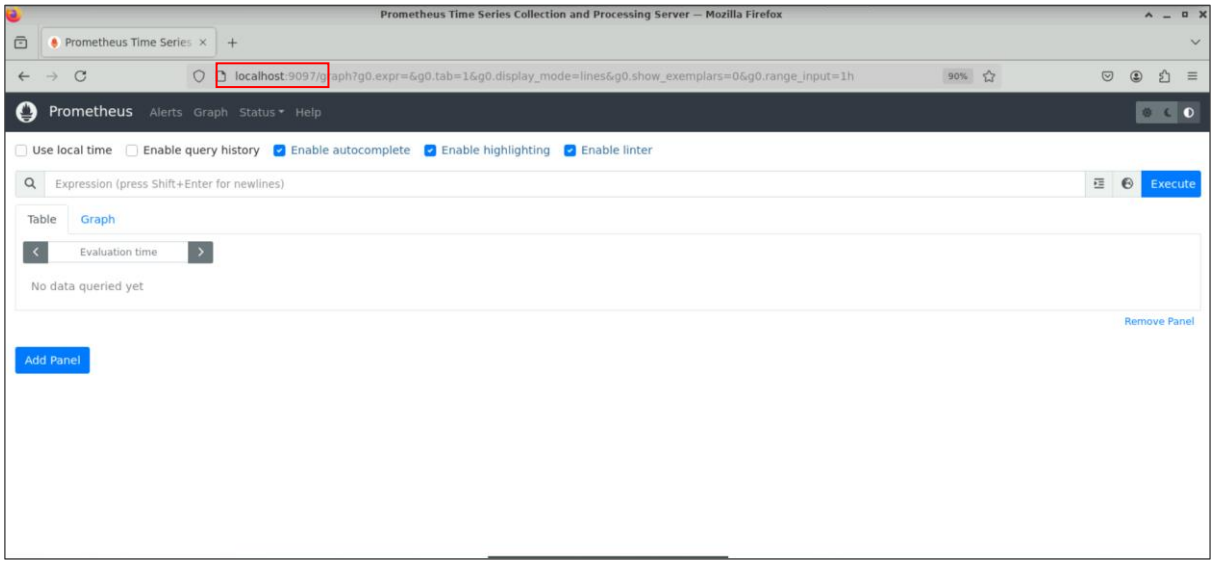


1.7 List running Docker containers using the following command:
sudo docker ps

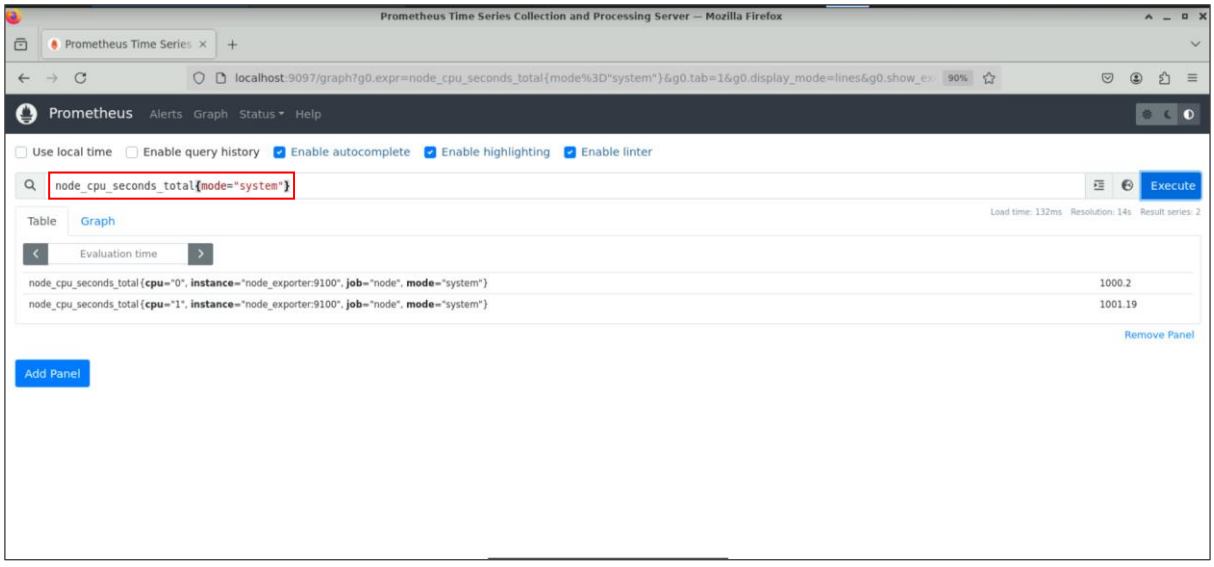


Step 2: Use the Prometheus UI to query Node Exporter metrics

2.1 Open the browser and access the Prometheus UI using the following URL:
http://localhost:9097



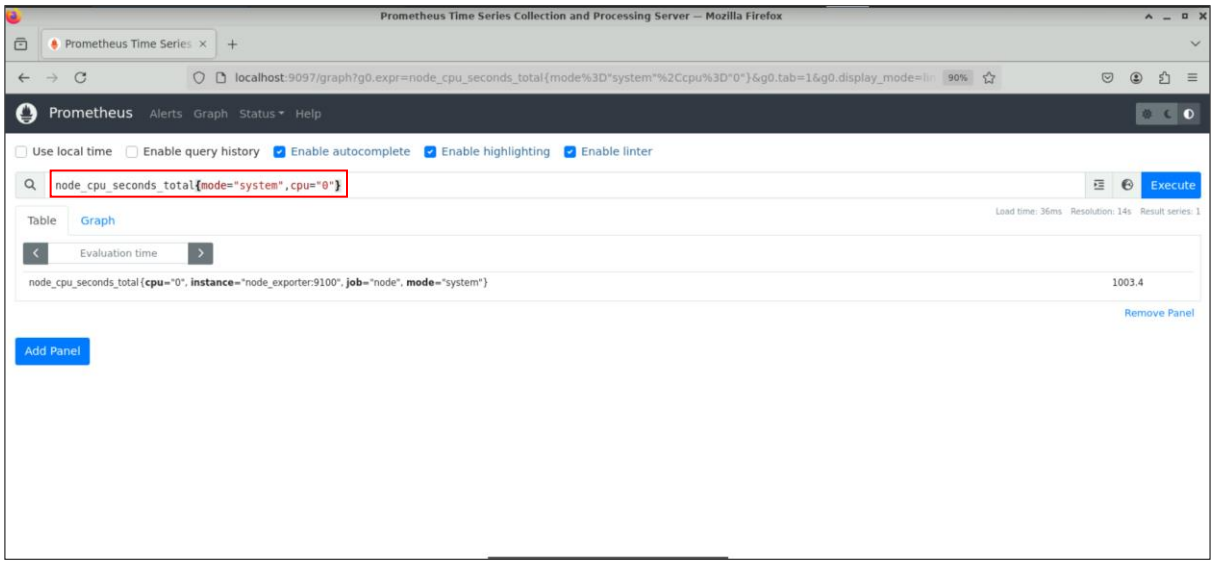
2.2 Filter metrics using the following query:
node_cpu_seconds_total{mode="system"}



Note: This example retrieves the total CPU time spent in system mode across all CPUs.

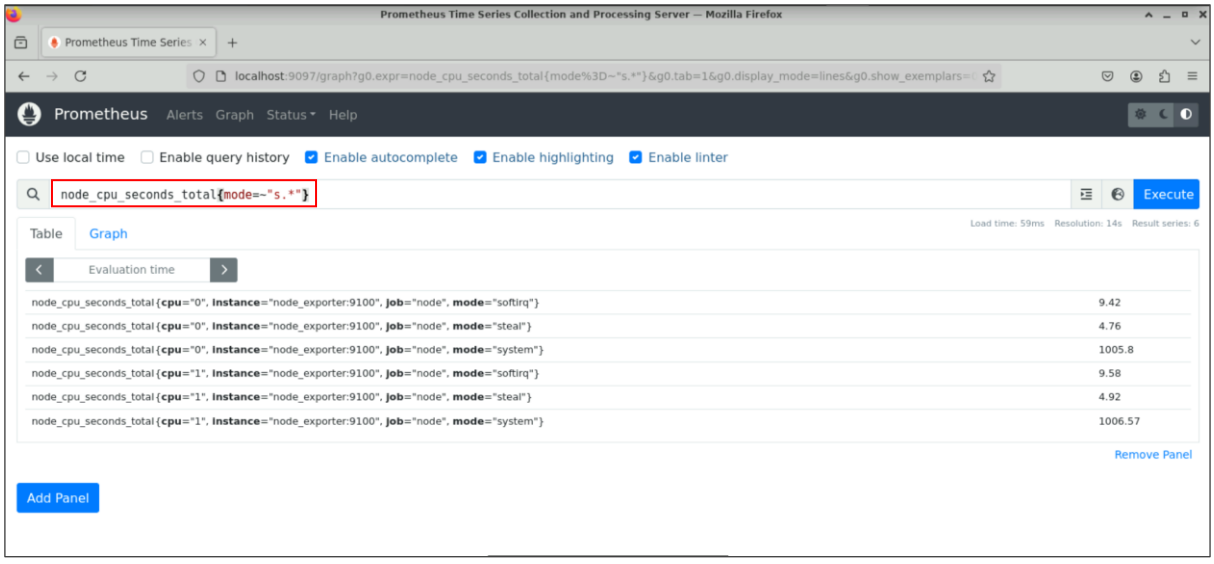
Note: Specific metrics can be selected based on their name and labels using PromQL. The basic syntax for selecting a metric is:
<metric_name>{<label_name>=<label_value>, ...}

2.3 Apply multiple label filters using the following query:
node_cpu_seconds_total{mode="system",cpu="0"}



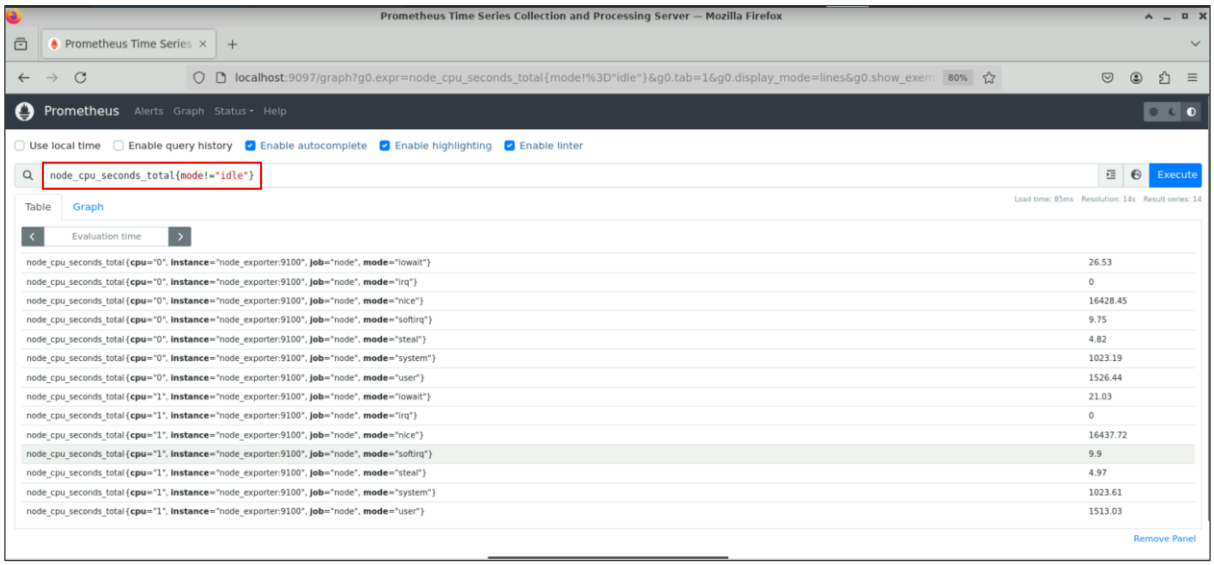
Note: This example retrieves the total CPU time spent in system mode specifically for CPU 0.

2.4 Match label values with regular expressions using the following query:
node_cpu_seconds_total{mode=~"s.*"}



Note: This example retrieves the total CPU time for all modes that start with the letter **s**.

2.5 Nullify a label filter using the following query:
node_cpu_seconds_total{mode!="idle"}



Note: This example retrieves the total CPU time for all modes except idle.

By following these steps you have successfully configured Node Exporter and Prometheus using Docker and written PromQL queries to extract metrics from the Node Exporter sample dataset.