Lesson 02 Demo 02

Writing Basic Queries in PromQL

Objective: To query and analyze monitoring data using Prometheus Query Language (PromQL) for effective monitoring of system performance and health

Tools required: Linux operating system

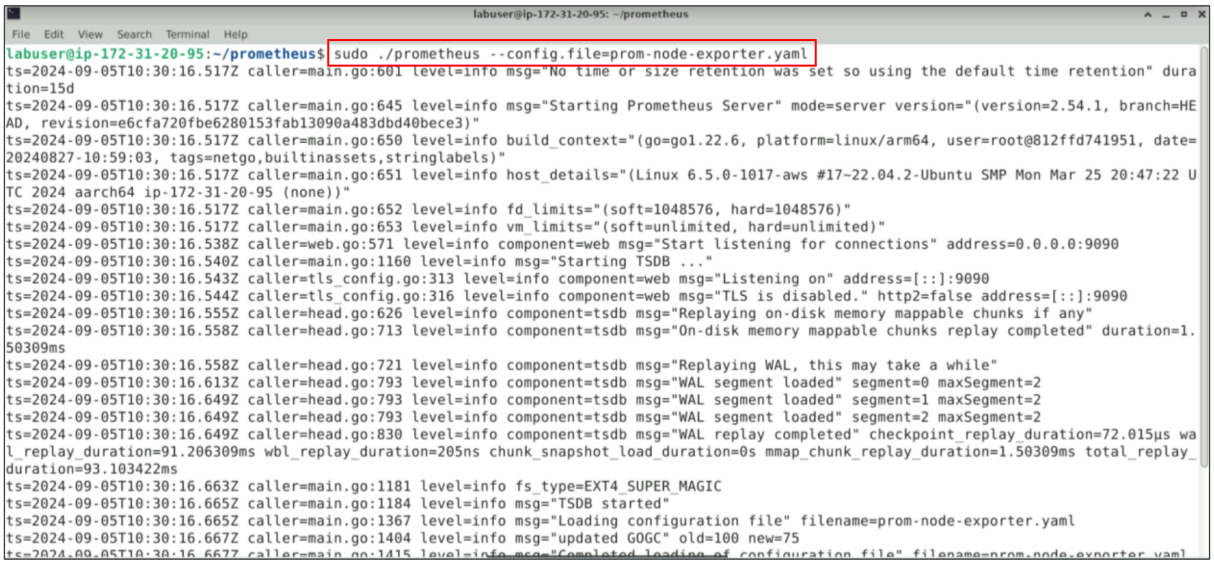
Prerequisites: Refer to Demo 01 of Lesson 02 for configuring Node Exporter

Steps to be followed:

1. Query to retrieve a single metric
2. Filter by label
3. Aggregate data with the sum() function
4. Query data using an arithmetic operation
5. Calculate a metric using the rate() function

Step 1: Query to retrieve a single metric

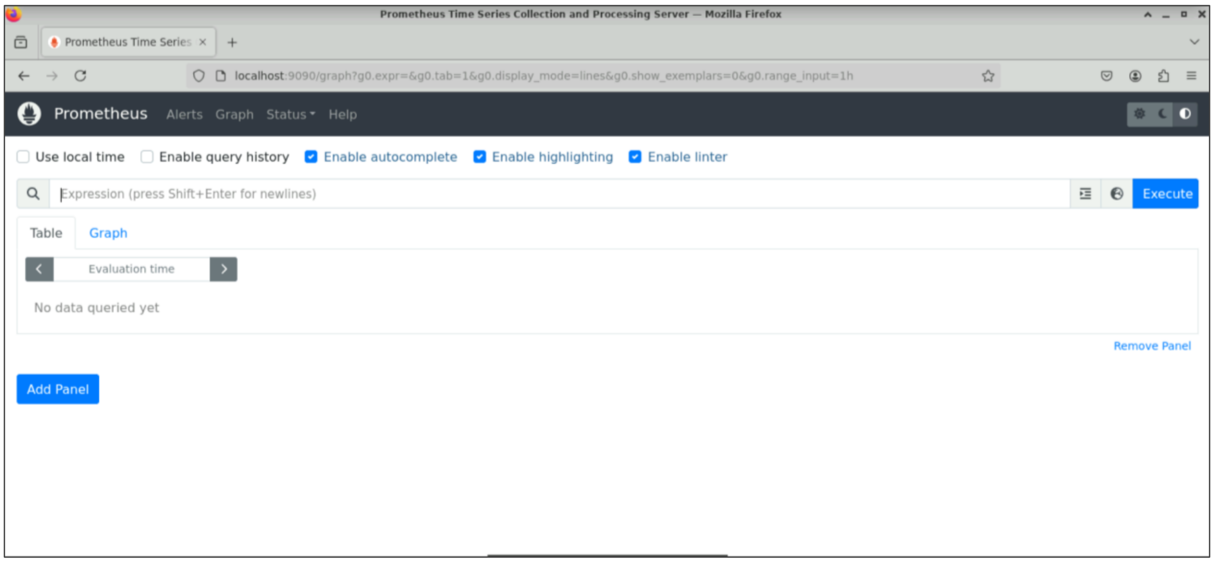
1.1 Navigate to the terminal and run the following command to start the Prometheus server:
sudo ./prometheus --config.file=prom-node-exporter.yaml



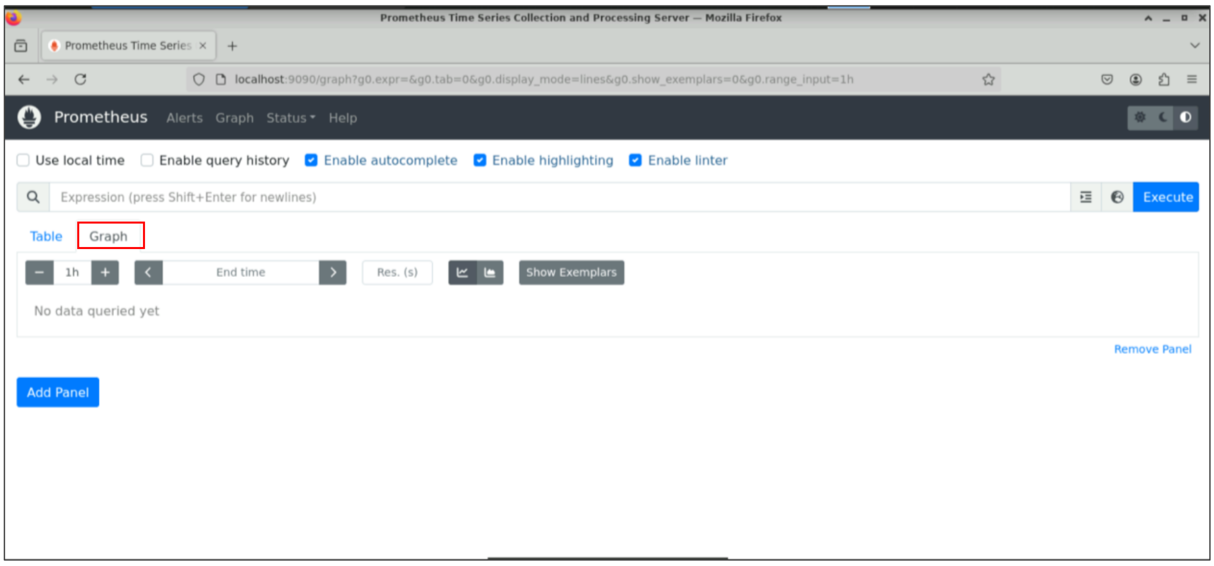
labuser@ip-172-31-20-95: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-20-95:~/prometheus$ sudo ./prometheus --config.file=prom-node-exporter.yaml
ts=2024-09-05T10:30:16.517Z caller=main.go:601 level=info msg="No time or size retention was set so using the default time retention" dura
tion=15d
ts=2024-09-05T10:30:16.517Z caller=main.go:645 level=info msg="Starting Prometheus Server" mode=server version="(version=2.54.1, branch=HE
AD, revision=e6cfa720fbe6280153fab13090a483dbd40bece3)"
ts=2024-09-05T10:30:16.517Z caller=main.go:650 level=info build_context="(go=go1.22.6, platform=linux/arm64, user=root@812ffd741951, date=
20240827-10:59:03, tags=netgo,builtinassets,stringlabels)"
ts=2024-09-05T10:30:16.517Z caller=main.go:651 level=info host_details="(Linux 6.5.0-1017-aws #17~22.04.2-Ubuntu SMP Mon Mar 25 20:47:22 U
TC 2024 aarch64 ip-172-31-20-95 (none))"
ts=2024-09-05T10:30:16.517Z caller=main.go:652 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2024-09-05T10:30:16.517Z caller=main.go:653 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-09-05T10:30:16.538Z caller=web.go:571 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2024-09-05T10:30:16.540Z caller=main.go:1160 level=info msg="Starting TSDB ..."
ts=2024-09-05T10:30:16.543Z caller=tls_config.go:313 level=info component=web msg="Listening on" address=[::]:9090
ts=2024-09-05T10:30:16.544Z caller=tls_config.go:316 level=info component=web msg="TLS is disabled." http2=false address=[::]:9090
ts=2024-09-05T10:30:16.555Z caller=head.go:626 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2024-09-05T10:30:16.558Z caller=head.go:713 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=1.
50309ms
ts=2024-09-05T10:30:16.558Z caller=head.go:721 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-09-05T10:30:16.613Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=2
ts=2024-09-05T10:30:16.649Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=1 maxSegment=2
ts=2024-09-05T10:30:16.649Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=2 maxSegment=2
ts=2024-09-05T10:30:16.649Z caller=head.go:830 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=72.015µs wa
l_replay_duration=91.206309ms wbl_replay_duration=205ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=1.50309ms total_replay_
duration=93.103422ms
ts=2024-09-05T10:30:16.663Z caller=main.go:1181 level=info fs_type=EXT4_SUPER_MAGIC
ts=2024-09-05T10:30:16.665Z caller=main.go:1184 level=info msg="TSDB started"
ts=2024-09-05T10:30:16.665Z caller=main.go:1367 level=info msg="Loading configuration file" filename=prom-node-exporter.yaml
ts=2024-09-05T10:30:16.667Z caller=main.go:1404 level=info msg="updated GOGC" old=100 new=75
ts=2024-09-05T10:30:16.667Z caller=main.go:1415 level=info msg="Completed loading of configuration file" filename=prom-node-exporter.yaml
```

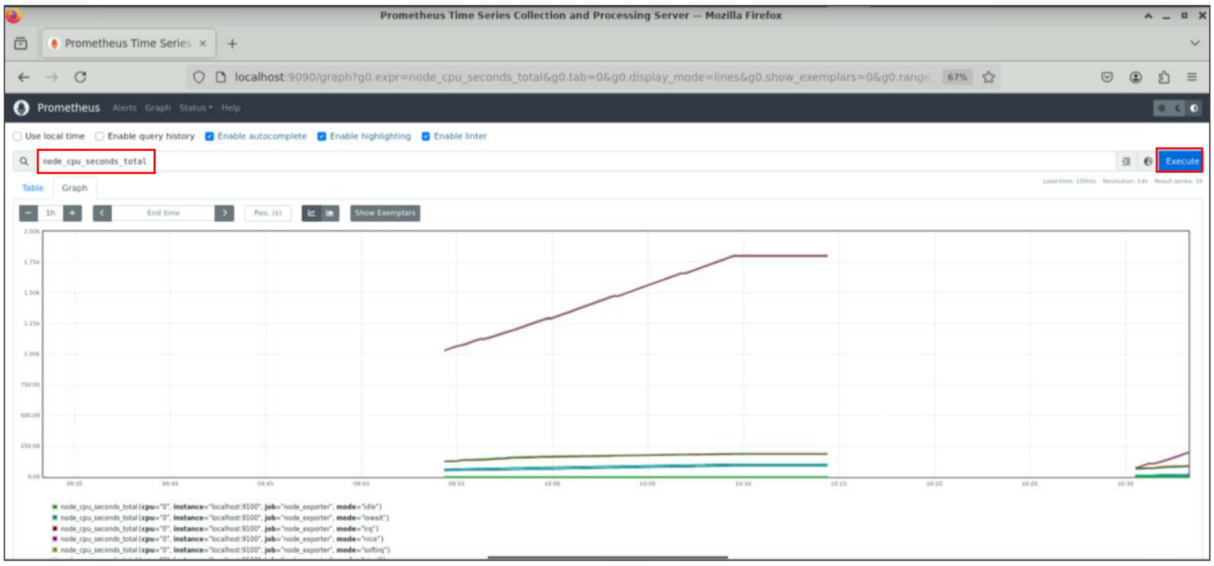
1.2 Navigate to the browser and enter the URL **http://localhost:9090/** or **http://<public-ip>:9090/** to access the Prometheus console


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series
localhost:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
Expression (press Shift+Enter for newlines)
Execute
Table
Graph
Evaluation time
No data queried yet
Remove Panel
Add Panel

1.3 Navigate to the **Graph** section



1.4 Enter the following query in the expression browser to retrieve a single metric, then click on **Execute**:
node_cpu_seconds_total

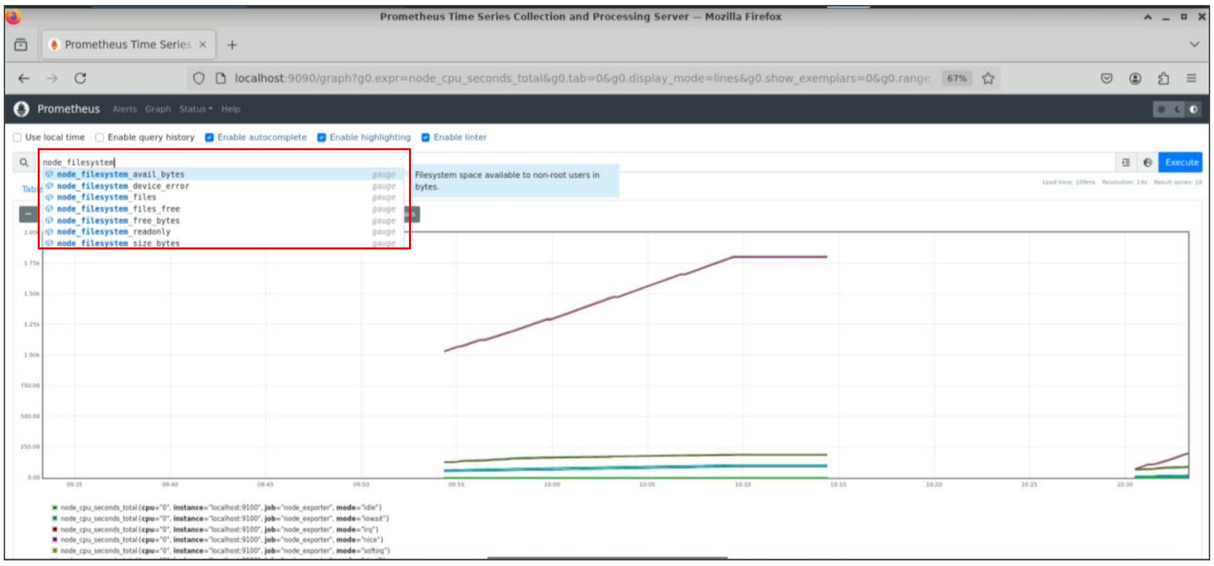


```
node_cpu_seconds_total
```

```
node_cpu_seconds_total{cpu="0", instance="localhost:9100", job="node_exporter", mode="idle"}
node_cpu_seconds_total{cpu="0", instance="localhost:9100", job="node_exporter", mode="iowait"}
node_cpu_seconds_total{cpu="0", instance="localhost:9100", job="node_exporter", mode="irq"}
node_cpu_seconds_total{cpu="0", instance="localhost:9100", job="node_exporter", mode="nice"}
node_cpu_seconds_total{cpu="0", instance="localhost:9100", job="node_exporter", mode="softirq"}
```

Step 2: Filter by label

2.1 Type **node_filesystem** in the expression browser

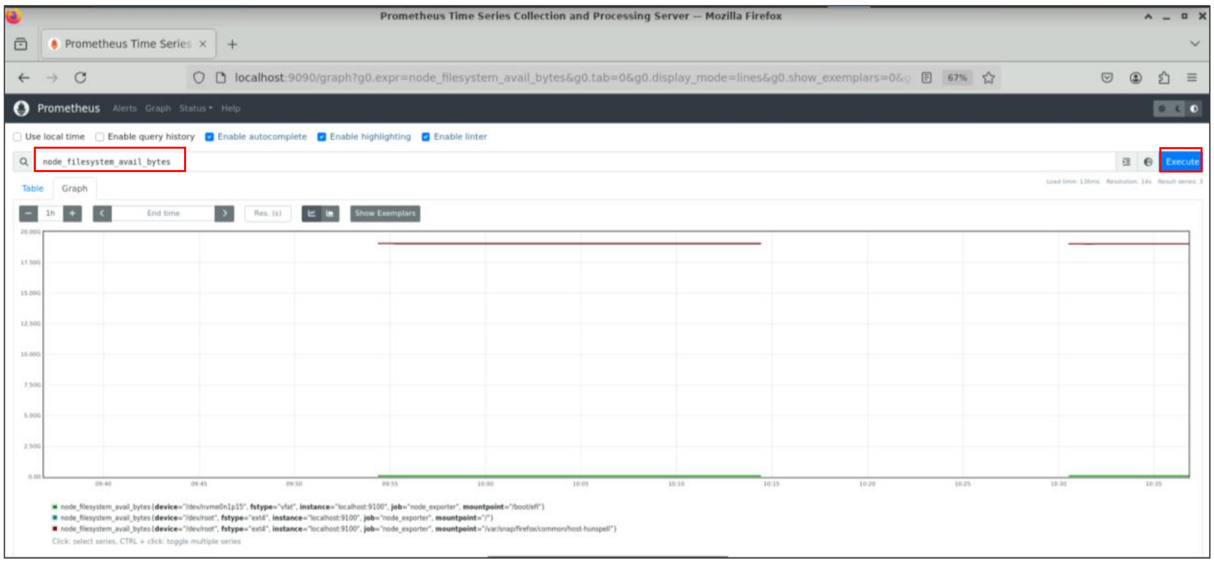


```
node_filesystem
node_filesystem_avail_bytes        gauge
node_filesystem_device_error       gauge
node_filesystem_files              gauge
node_filesystem_files_free         gauge
node_filesystem_free_bytes         gauge
node_filesystem_readonly           gauge
node_filesystem_size_bytes         gauge
```

Filesystem space available to non-root users in bytes.

It will display a popup list.

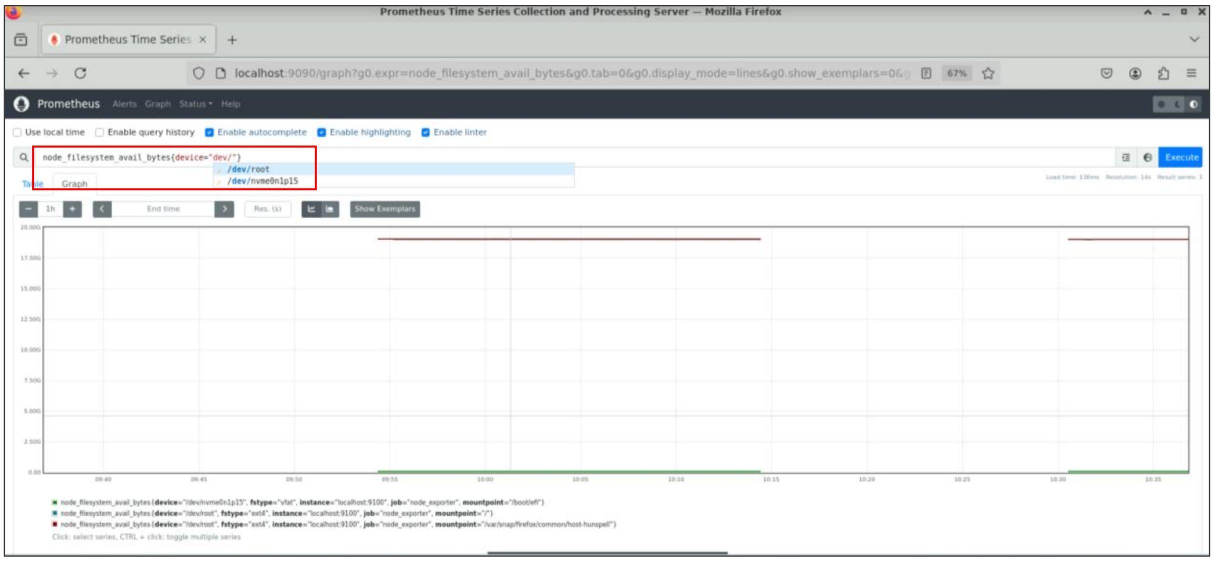
2.2 Select **node_filesystem_avail_bytes** and click **Execute**



```
node_filesystem_avail_bytes
```

```
node_filesystem_avail_bytes{device="/dev/nvme0n1p15", fstype="vfat", instance="localhost:9100", job="node_exporter", mountpoint="/boot/efi"}
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/"}
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/var/snap/firefox/common/host-hunspell"}
Click: select series, CTRL + click: toggle multiple series
```

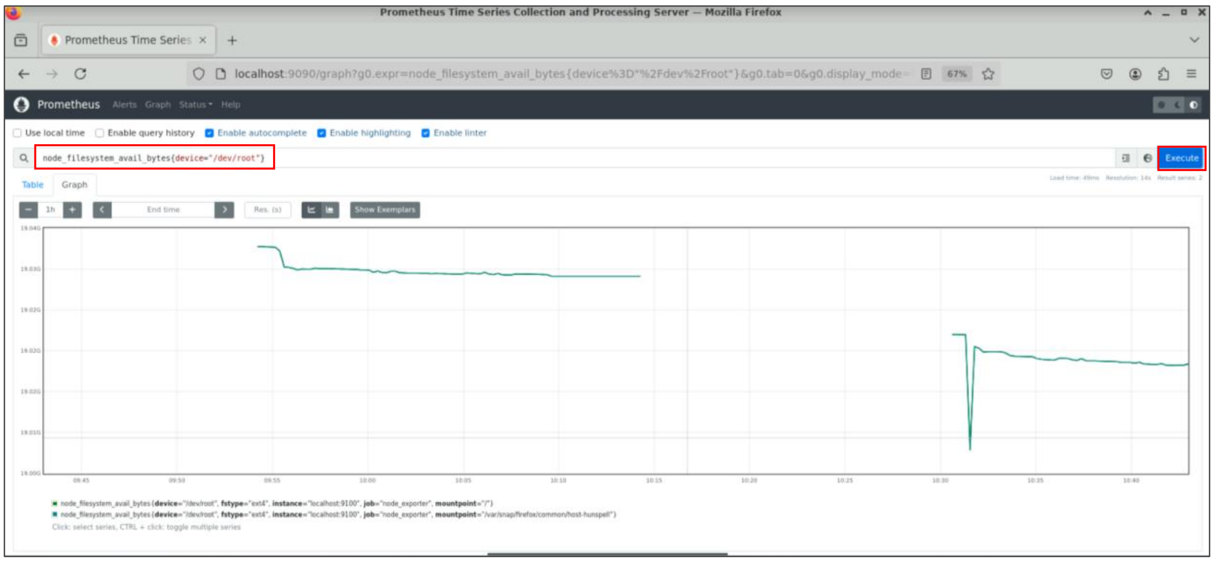
2.3 Filter by labels using the following query:
node_filesystem_avail_bytes{device="/dev/"}



```
node_filesystem_avail_bytes{device="dev/"}
/dev/root
/dev/nvme0n1p15
```

```
node_filesystem_avail_bytes{device="/dev/nvme0n1p15", fstype="vfat", instance="localhost:9100", job="node_exporter", mountpoint="/boot/efi"}
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/"}
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/var/snap/firefox/common/host-hunspell"}
Click: select series, CTRL + click: toggle multiple series
```

2.4 Select **/dev/root** and click on **Execute** to run the query



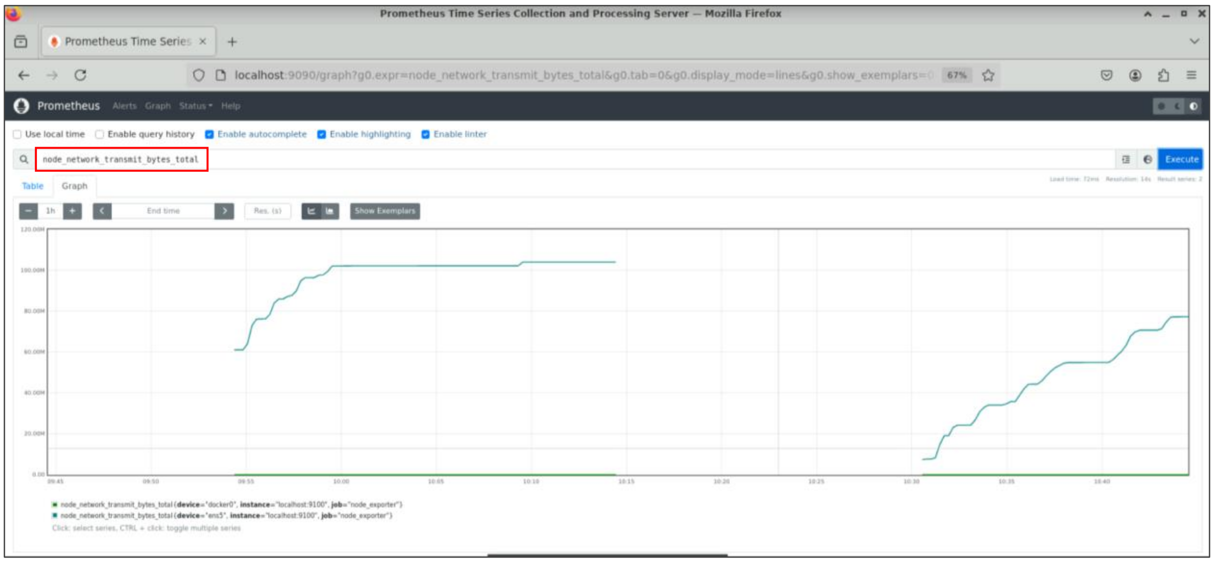
```
localhost:9090/graph?g0.expr=node_filesystem_avail_bytes{device%3D"%2Fdev%2Froot"}&g0.tab=0&g0.display_mode=
```

```
node_filesystem_avail_bytes{device="/dev/root"}
```

```
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/"}
node_filesystem_avail_bytes{device="/dev/root", fstype="ext4", instance="localhost:9100", job="node_exporter", mountpoint="/var/snap/firefox/common/host-hunspell"}
Click: select series, CTRL + click: toggle multiple series
```

Step 3: Aggregate data with the sum() function

3.1 In the expression browser, enter the following query and execute it:
node_network_transmit_bytes_total

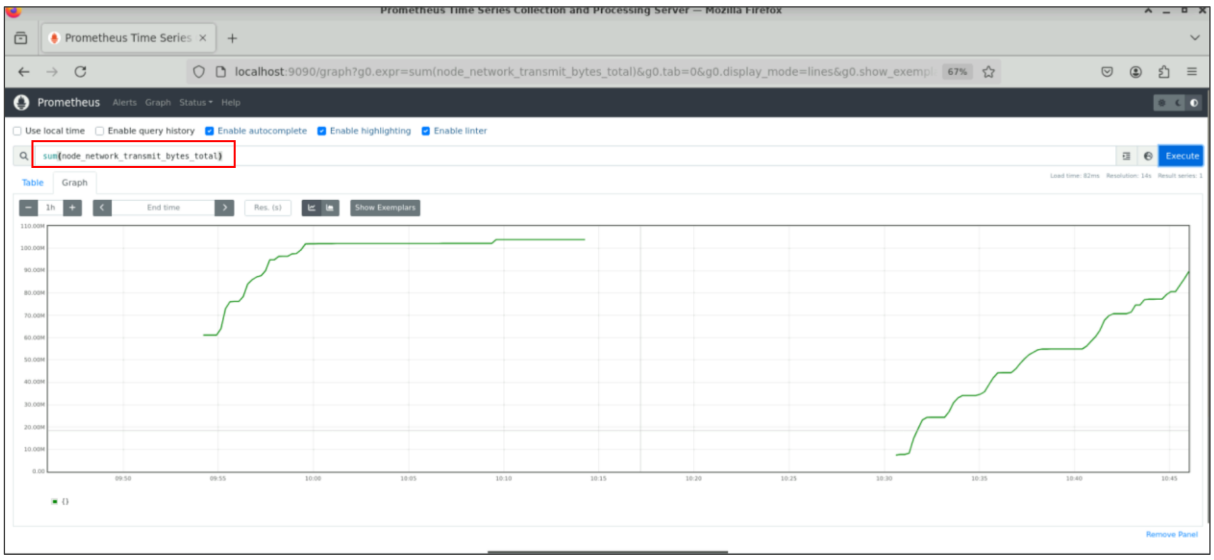


```
localhost:9090/graph?g0.expr=node_network_transmit_bytes_total&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0
```

```
node_network_transmit_bytes_total
```

```
node_network_transmit_bytes_total{device="docker0", instance="localhost:9100", job="node_exporter"}
node_network_transmit_bytes_total{device="ens5", instance="localhost:9100", job="node_exporter"}
Click: select series, CTRL + click: toggle multiple series
```

3.2 Use the following query to sum the total number of bytes transmitted over the network interface and view the graph:
sum(node_network_transmit_bytes_total)


Prometheus Time Series Collection and Processing Server — Mozilla Firefox

localhost:9090/graph?g0.expr=sum(node_network_transmit_bytes_total)&g0.tab=0&g0.display_mode=lines&g0.show_exempl

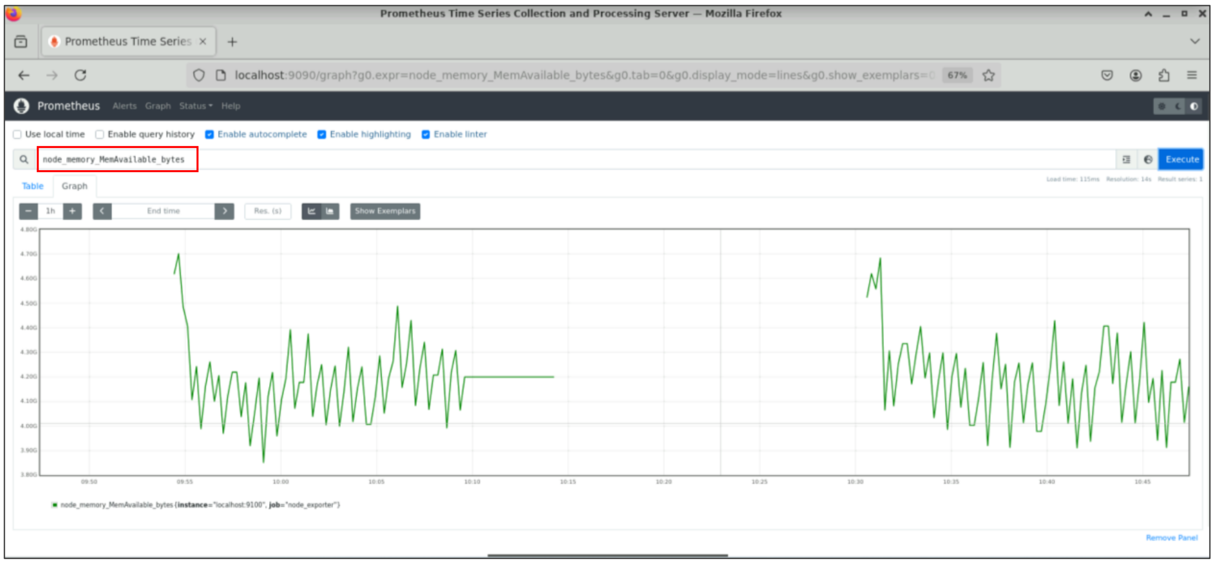
☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

sum(node_network_transmit_bytes_total)

🟩 {}

Step 4: Query data using an arithmetic operation

4.1 Enter the following query to display the amount of available memory on a node in bytes:
node_memory_MemAvailable_bytes


Prometheus Time Series Collection and Processing Server — Mozilla Firefox

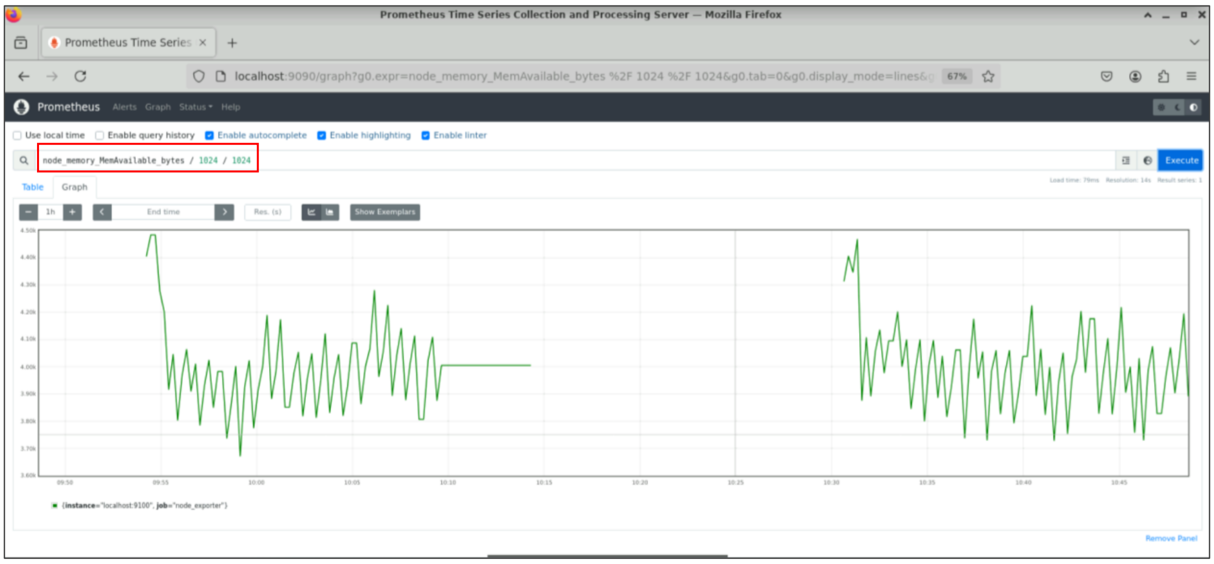
localhost:9090/graph?g0.expr=node_memory_MemAvailable_bytes&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0

☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

node_memory_MemAvailable_bytes

🟩 node_memory_MemAvailable_bytes{instance="localhost:9100", job="node_exporter"}

4.2 Divide the query executed in the previous step by **1024** twice to display it in megabytes using the following query:
node_memory_MemAvailable_bytes / 1024 / 1024


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series
localhost:9090/graph?g0.expr=node_memory_MemAvailable_bytes %2F 1024 %2F 1024&g0.tab=0&g0.display_mode=lines&g
67%
Prometheus
Alerts
Graph
Status
Help
Table
Graph
1h
End time
Res. (s)
Show Exemplars
Execute
Remove Panel

☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

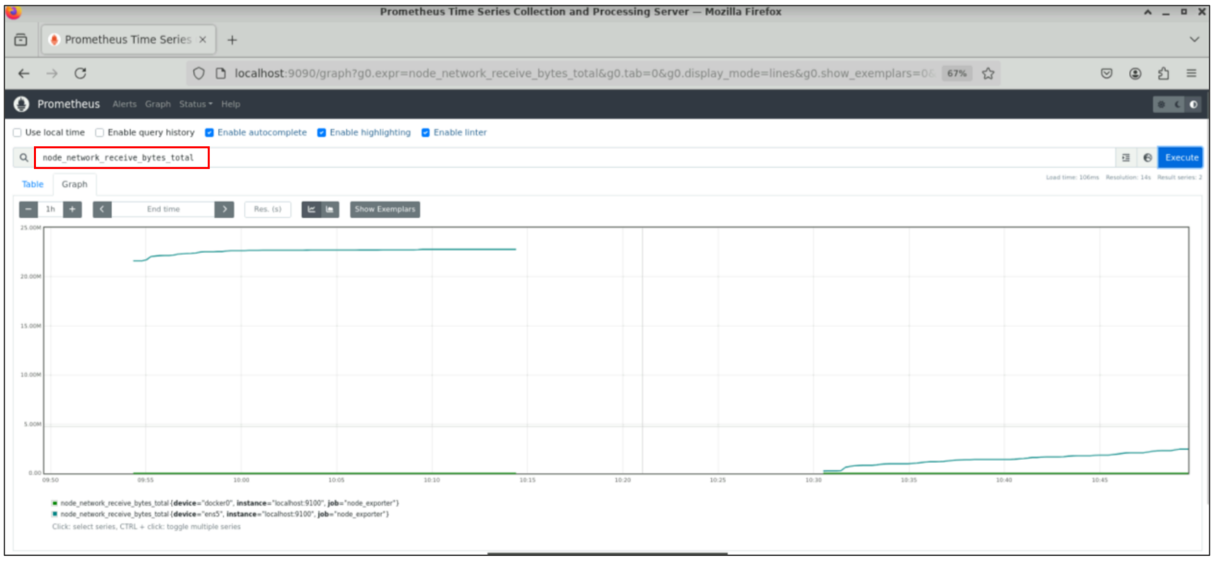
```
node_memory_MemAvailable_bytes / 1024 / 1024
```

Load time: 79ms Resolution: 14s Result series: 1

🟩 {**instance**="localhost:9100", **job**="node_exporter"}

Step 5: Calculate a metric using the rate() function

5.1 Execute the following query in the expression browser to represent the total number of bytes received over the network interface since the system started:
node_network_receive_bytes_total


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series
localhost:9090/graph?g0.expr=node_network_receive_bytes_total&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0&
67%
Prometheus
Alerts
Graph
Status
Help
Table
Graph
1h
End time
Res. (s)
Show Exemplars
Execute

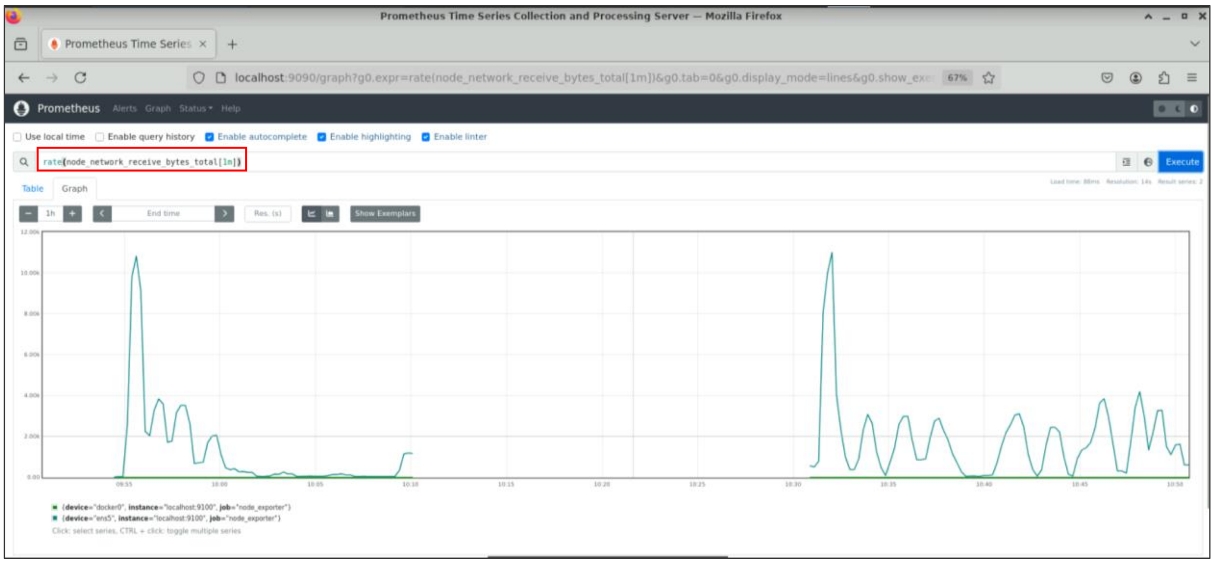
☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
node_network_receive_bytes_total
```

Load time: 106ms Resolution: 14s Result series: 2

🟩 node_network_receive_bytes_total{**device**="docker0", **instance**="localhost:9100", **job**="node_exporter"}
🟩 node_network_receive_bytes_total{**device**="ens5", **instance**="localhost:9100", **job**="node_exporter"}
Click: select series, CTRL + click: toggle multiple series

5.2 Enter the following query to calculate the average per-second rate of bytes received over the network interface in the last minute:
rate(node_network_receive_bytes_total[1m])



```
rate(node_network_receive_bytes_total[1m])
```

```
{device="docker0", instance="localhost:9100", job="node_exporter"}
{device="ens5", instance="localhost:9100", job="node_exporter"}
Click: select series, CTRL + click: toggle multiple series
```

By following these steps, you have successfully queried and analyzed monitoring data using Prometheus Query Language to effectively monitor system performance and health.