**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

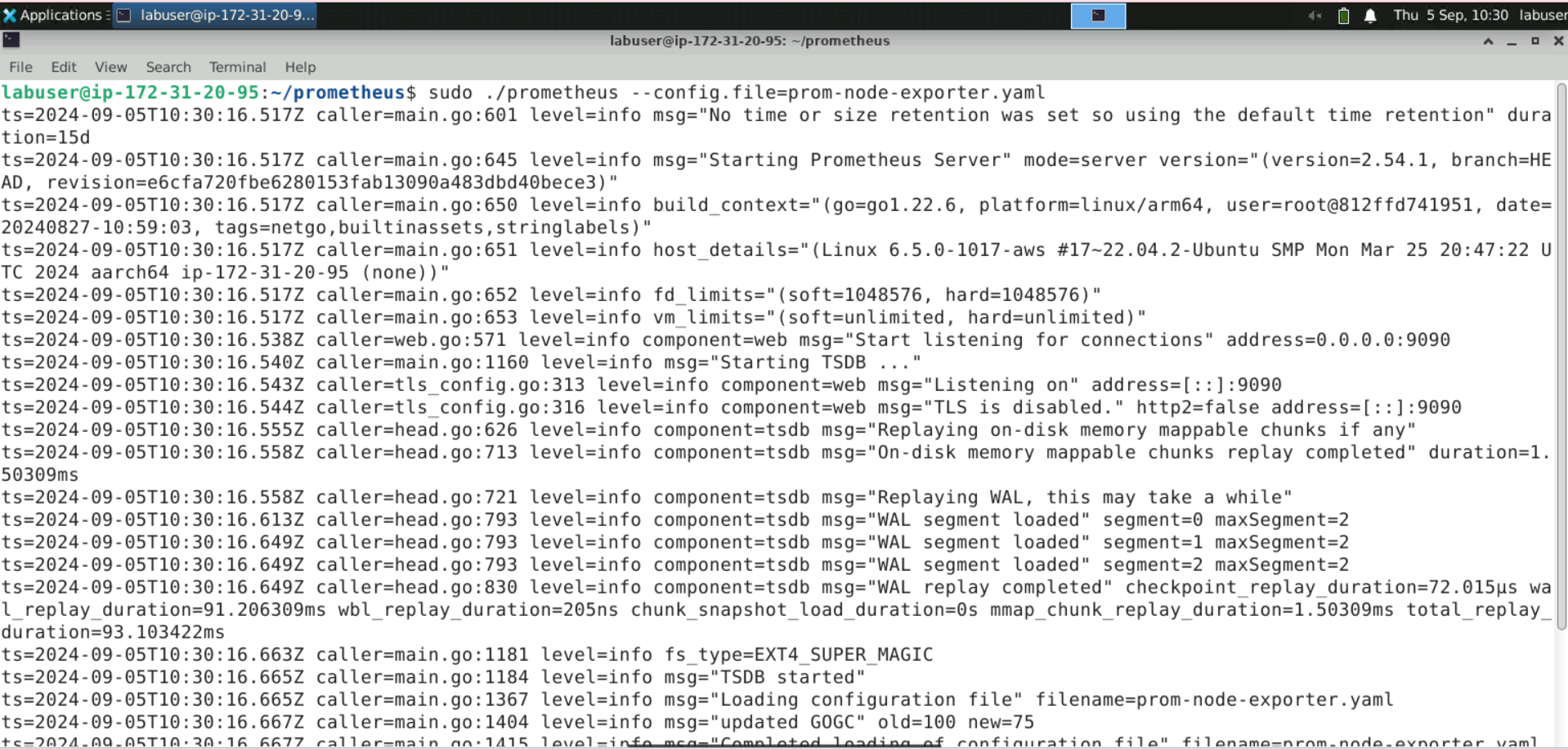
Steps to be followed:

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. Navigate to the terminal and run the following command to start the Prometheus server:

**sudo ./prometheus --config.file=prom-node-exporter.yaml**

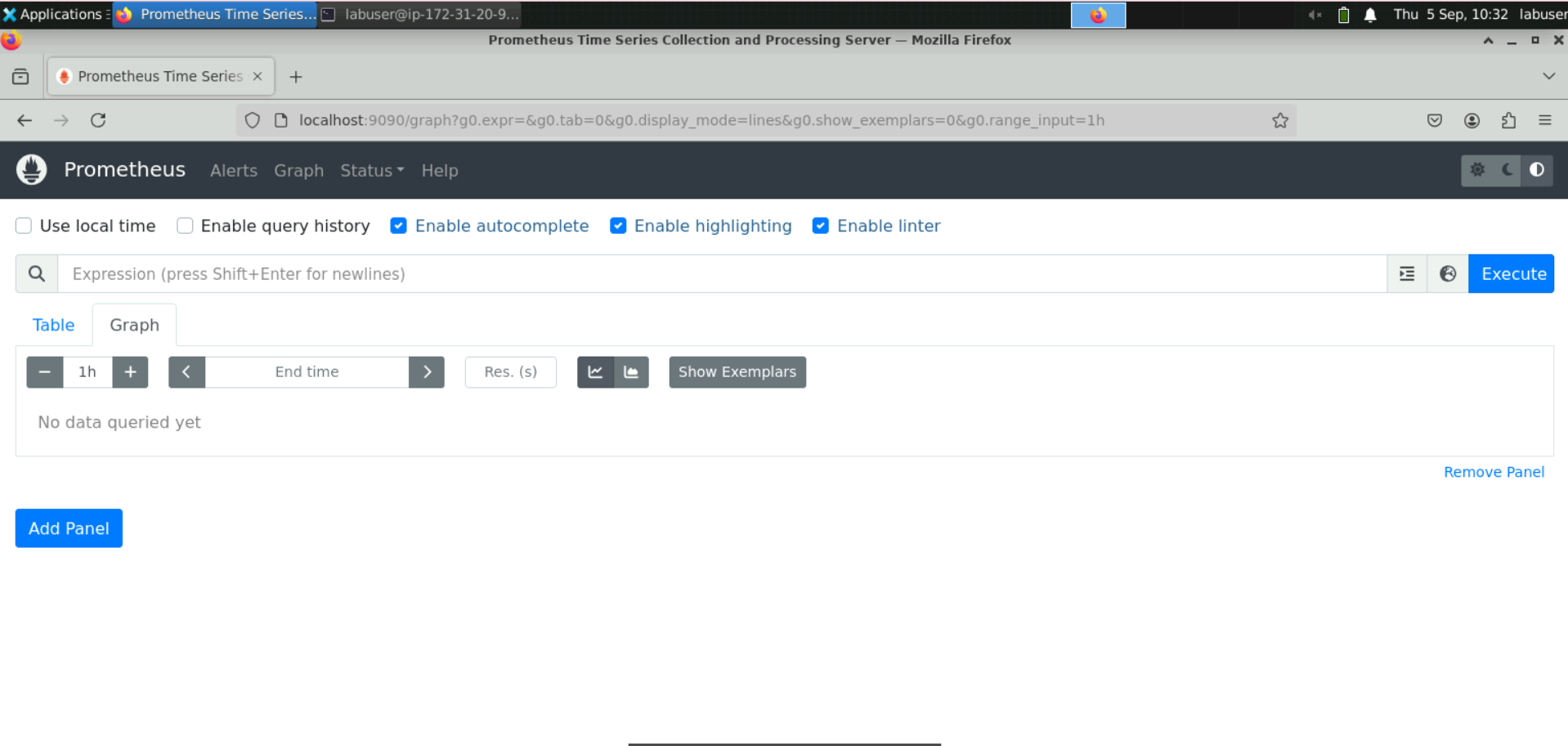


1. Navigate to the browser and enter the URL **http://localhost:9090/** or **http://<public-ip>:9090/** to access the Prometheus console

A screenshot of a computer

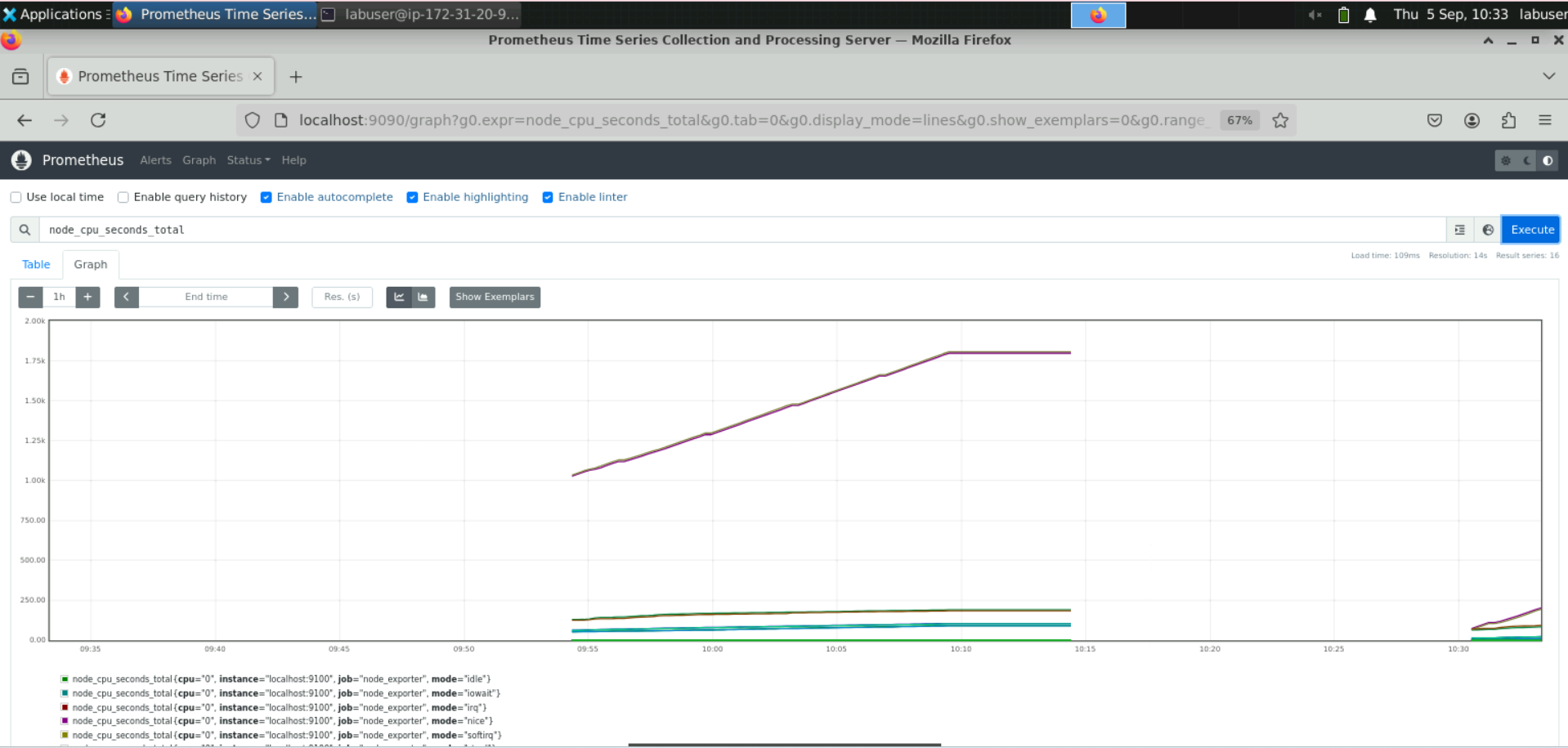
Description automatically generated

1. Navigate to the **Graph** section

****

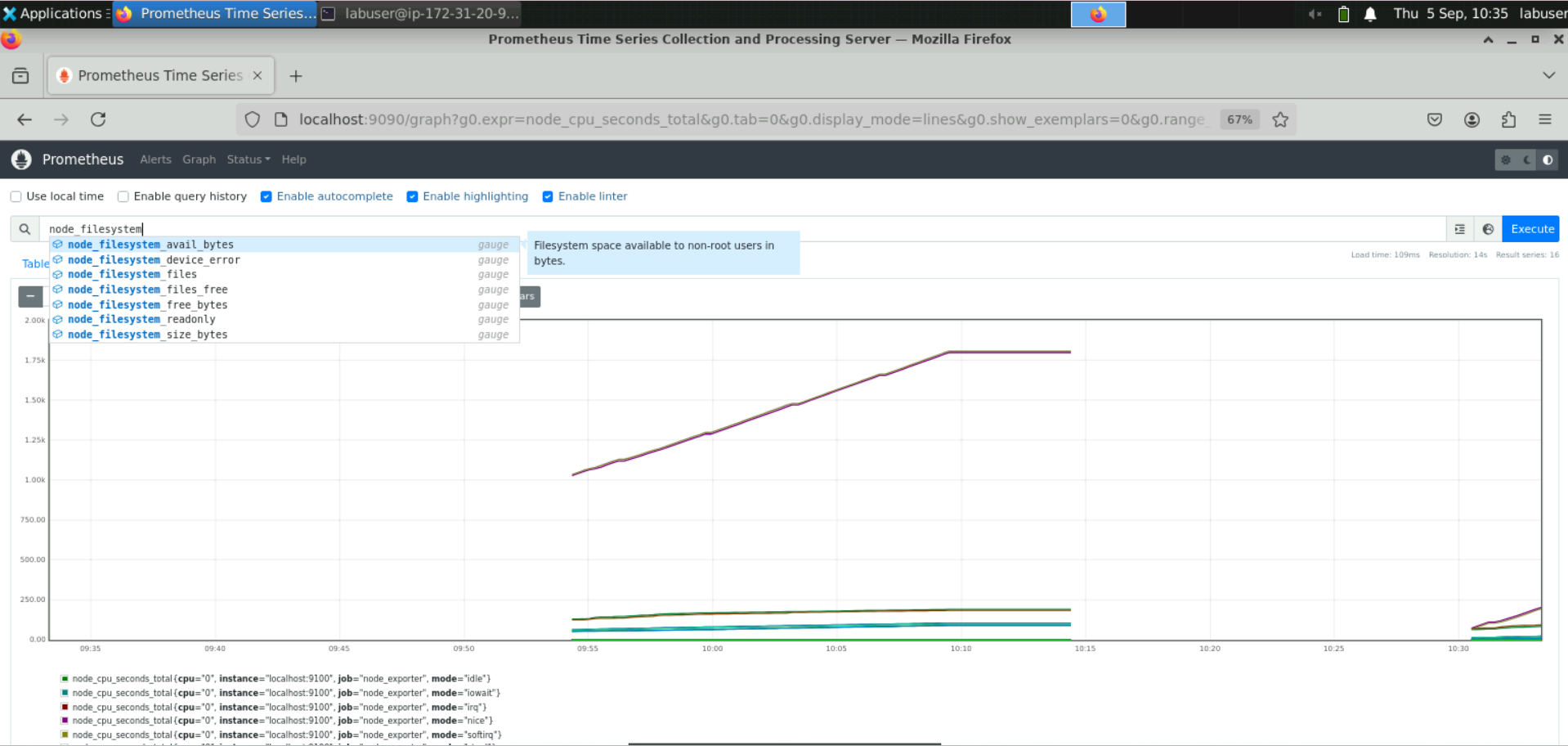
1. Enter the following query in the expression browser to retrieve a single metric, then click on **Execute**:

**node\_cpu\_seconds\_total**



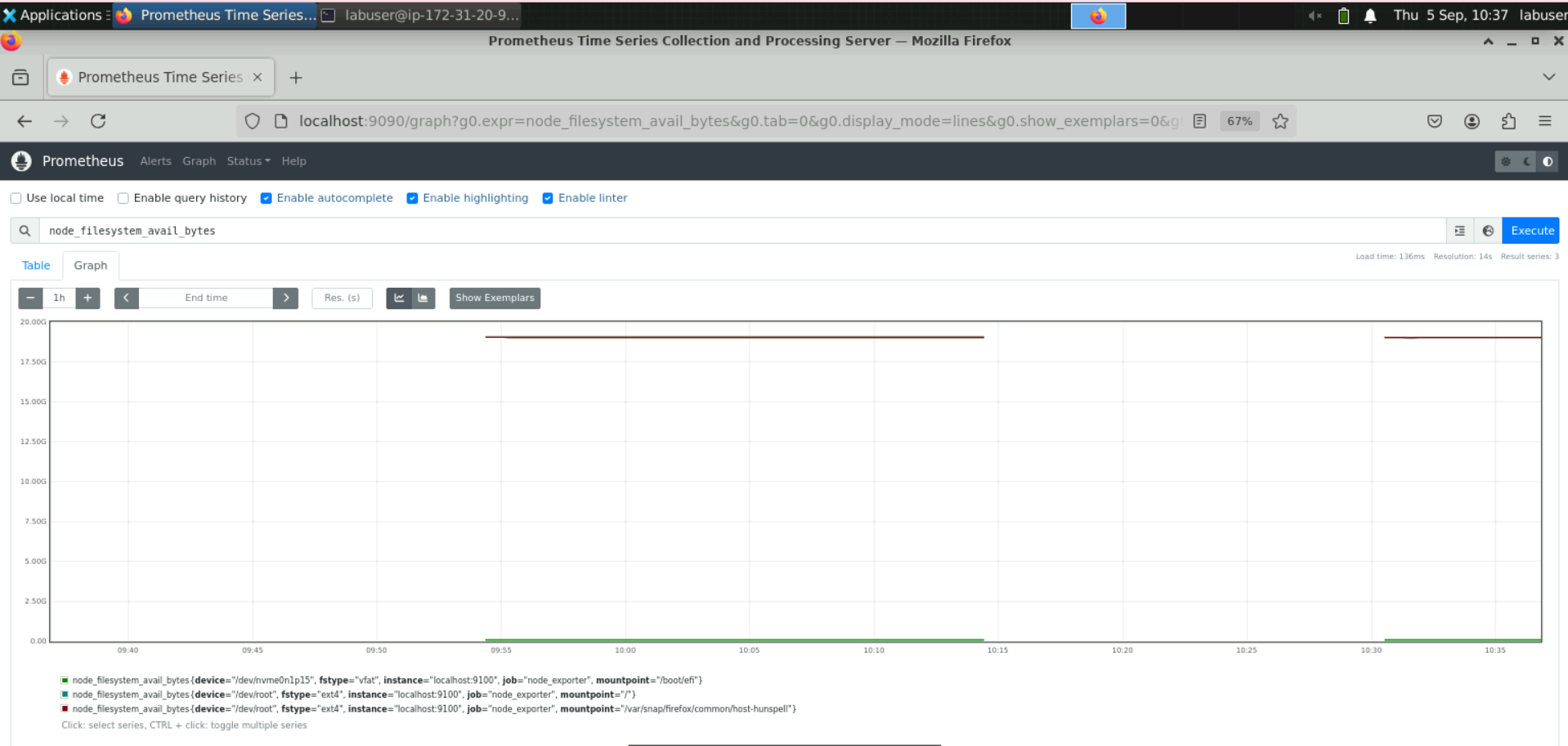
**Step 2: Filter by label**

1. Type **node\_filesystem** in the expression browser



It will display a popup list.

1. Select **node\_filesystem\_avail\_bytes** and click **Execute**



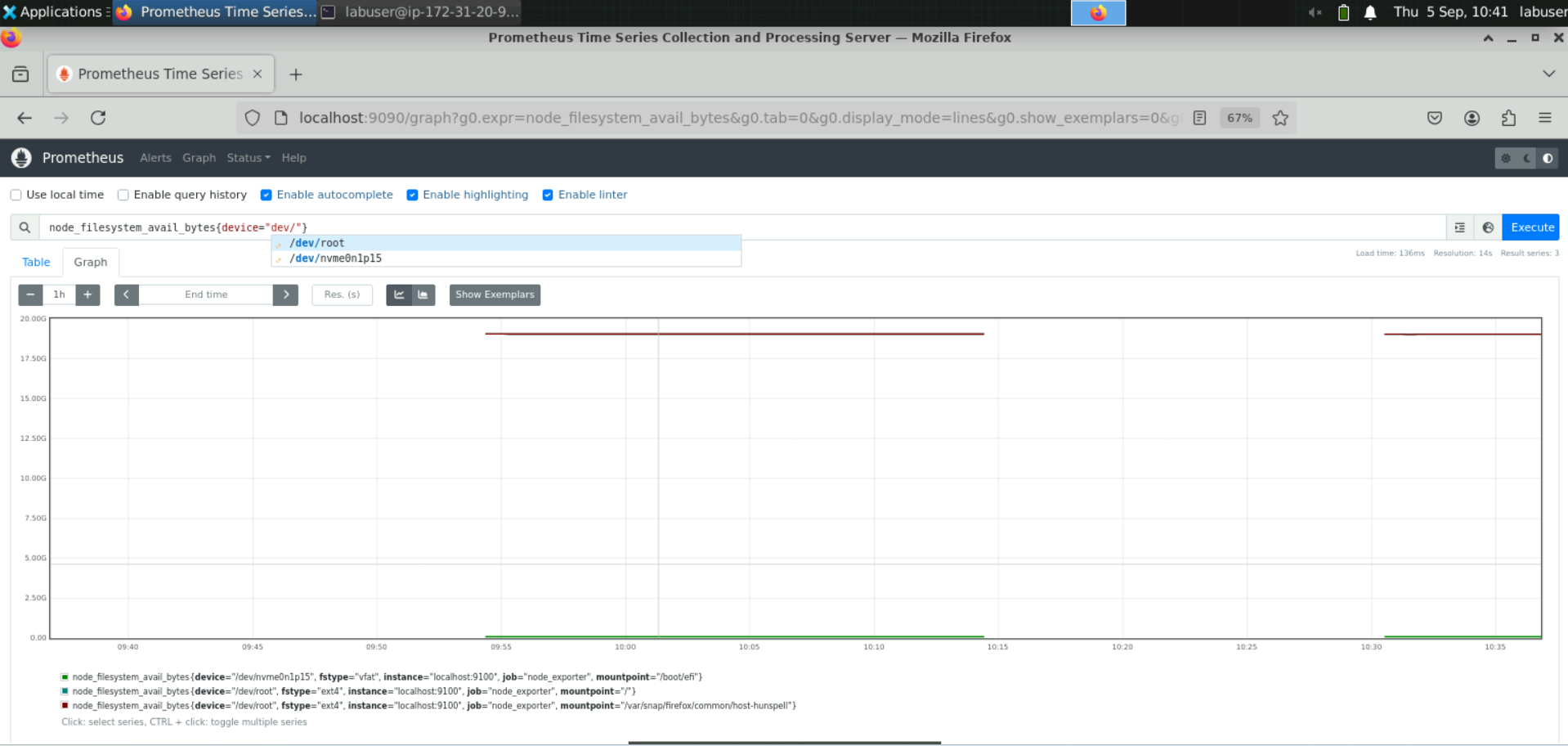
1. Filter by labels using the following query:  
    **node\_filesystem\_avail\_bytes{device=”/dev/”}**

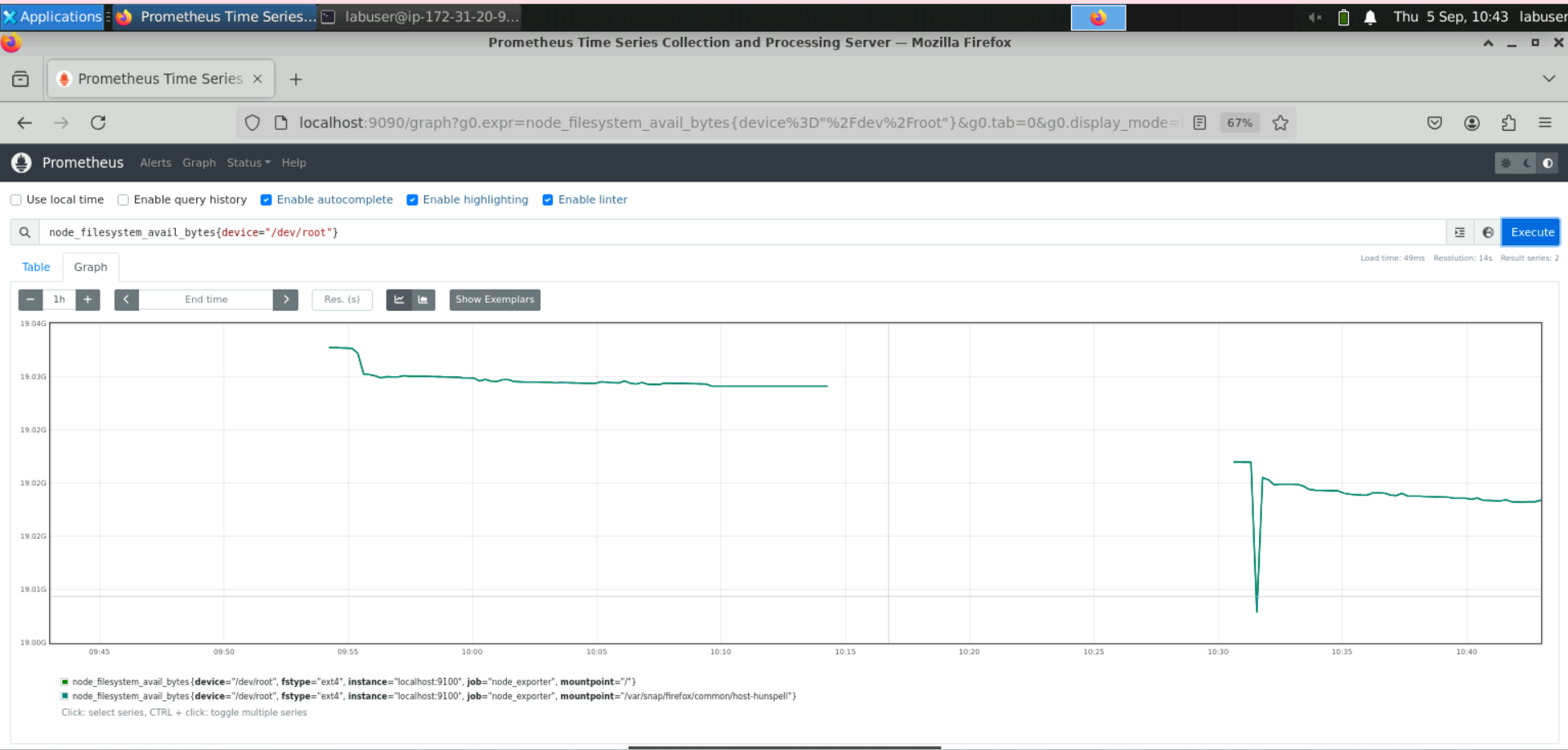
**Explanation:**

* **node\_filesystem\_avail\_bytes: This metric represents the number of available bytes on the filesystem. It's commonly used to monitor disk space.**
* **{device="/dev/"}: This filter selects only those metrics where the device label is exactly equal to "/dev/".**

### Practical Meaning:

This query retrieves the available disk space in bytes for filesystems associated with devices mounted under "/dev/". It's helpful for tracking disk space availability on specific devices.

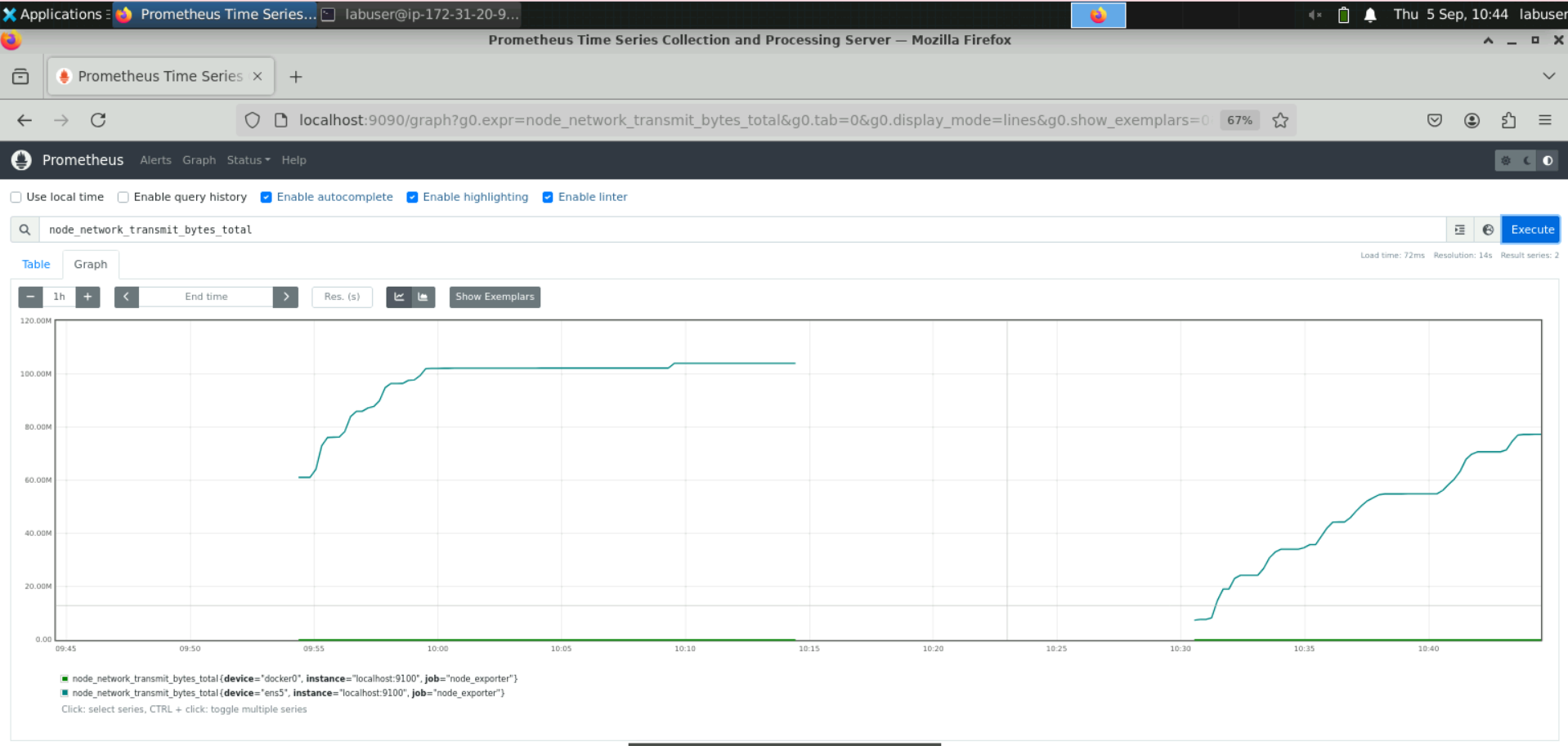
****

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

**Step 3: Aggregate data with the sum() function**

1. In the expression browser, enter the following query and execute it:  
   **node\_network\_transmit\_bytes\_total**

**node\_network\_transmit\_bytes\_total**: This metric represents the cumulative number of bytes transmitted (sent) over all network interfaces since the system started. This query fetches the total amount of data sent by the system across its network interfaces.



1. Use the following query to sum the total number of bytes transmitted over the network interface and view the graph:

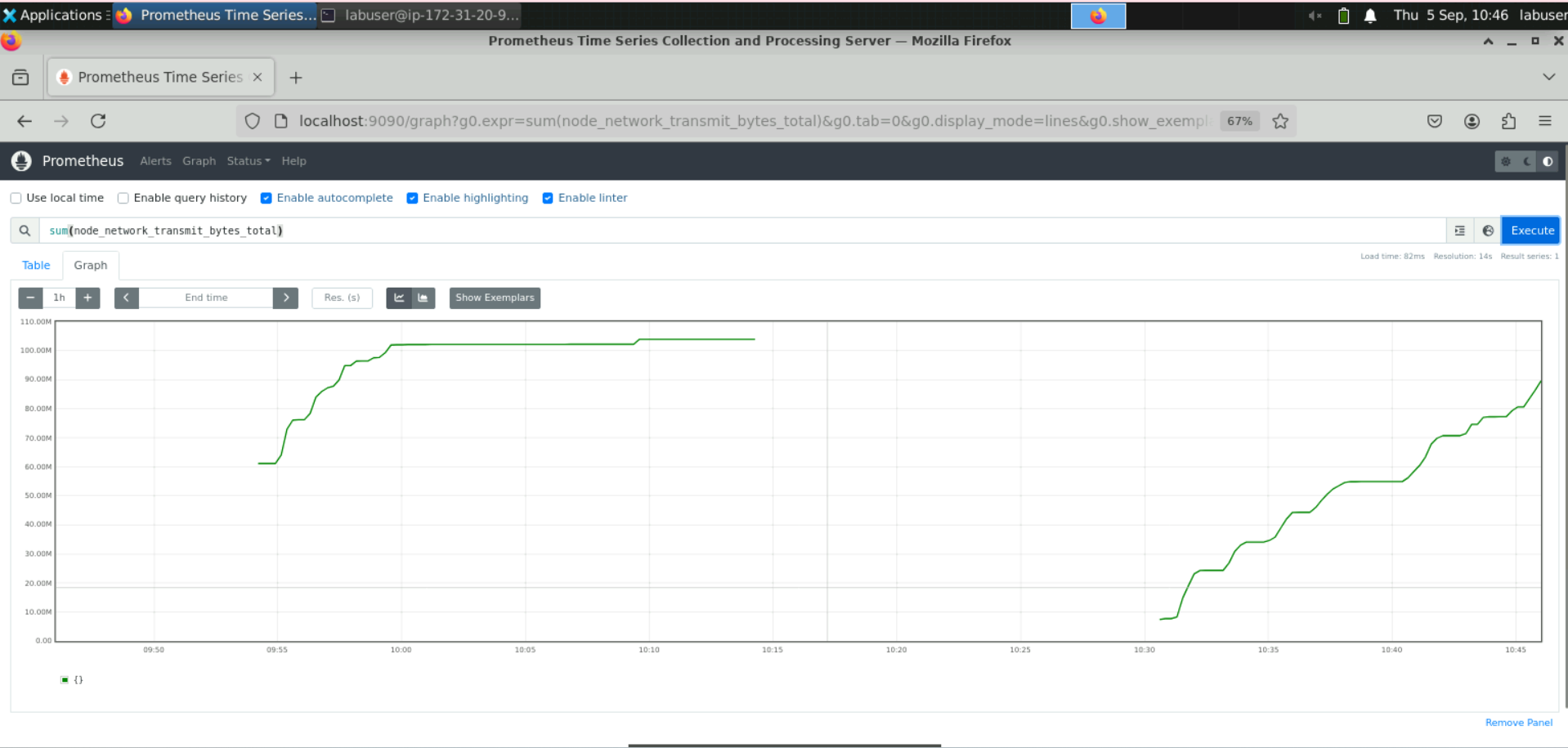
**sum(node\_network\_transmit\_bytes\_total)**

** node\_network\_transmit\_bytes\_total: This metric represents the total number of bytes transmitted over all network interfaces since the system started.**

** sum(...): Aggregates the total transmitted bytes across all instances and network interfaces.**

**Practical Meaning:**

**This query calculates the cumulative sum of all bytes transmitted across all network interfaces and nodes being monitored by Prometheus.**



**Step 4: Query data using an arithmetic operation**

* 1. Enter the following query to display the amount of available memory on a node   
     in bytes:

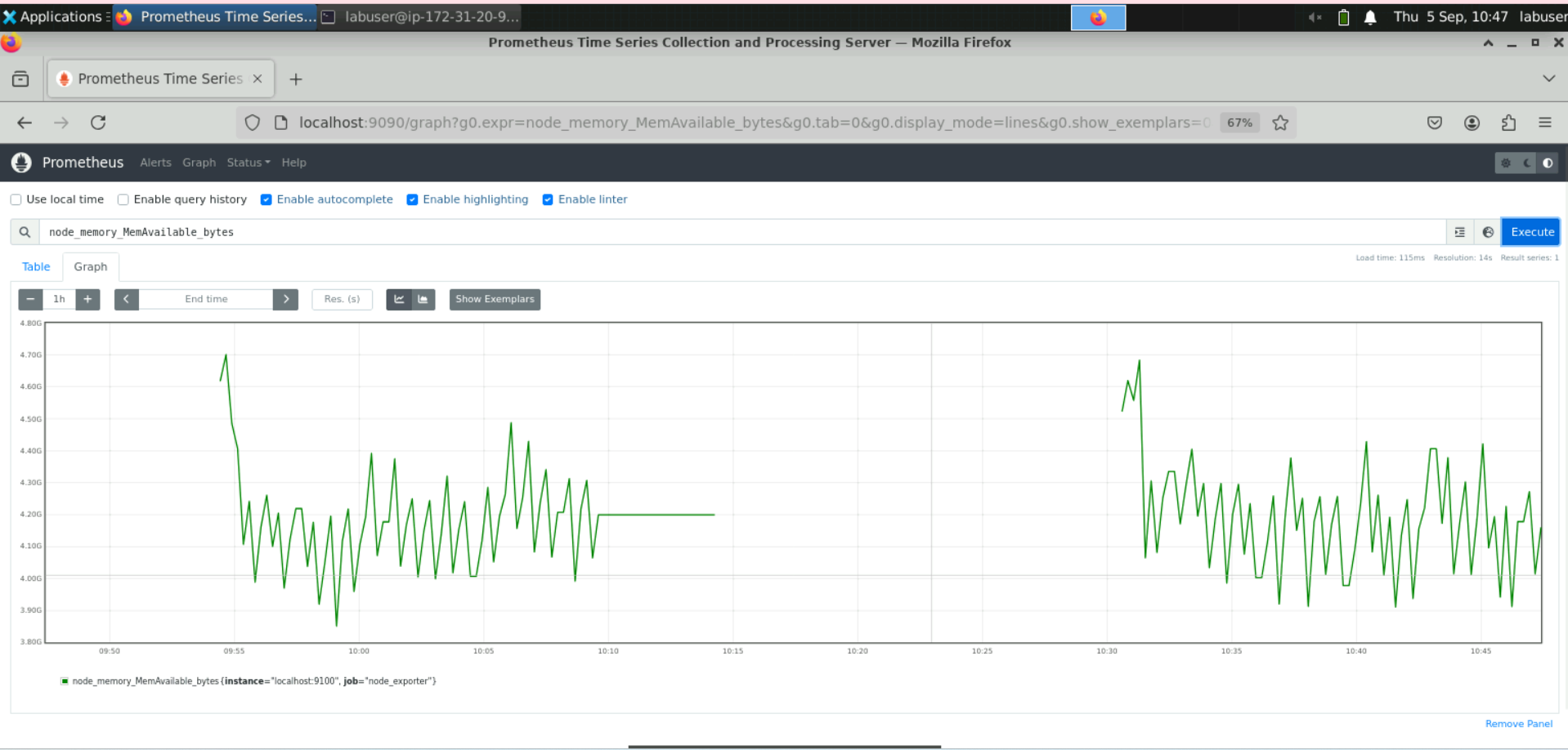
**node\_memory\_MemAvailable\_bytes**

**(node\_memory\_MemAvailable\_bytes**: This metric represents the amount of memory (in bytes) that is available for new processes to allocate without swapping. It includes free memory, reclaimable memory, and buffers/caches that can be repurposed if needed.

**Practical Meaning:**

This query provides the current amount of available system memory, which is useful for monitoring the memory health of your system or infrastructure.

)



* 1. Divide the query executed in the previous step by **1024** twice to display it in megabytes using the following query:

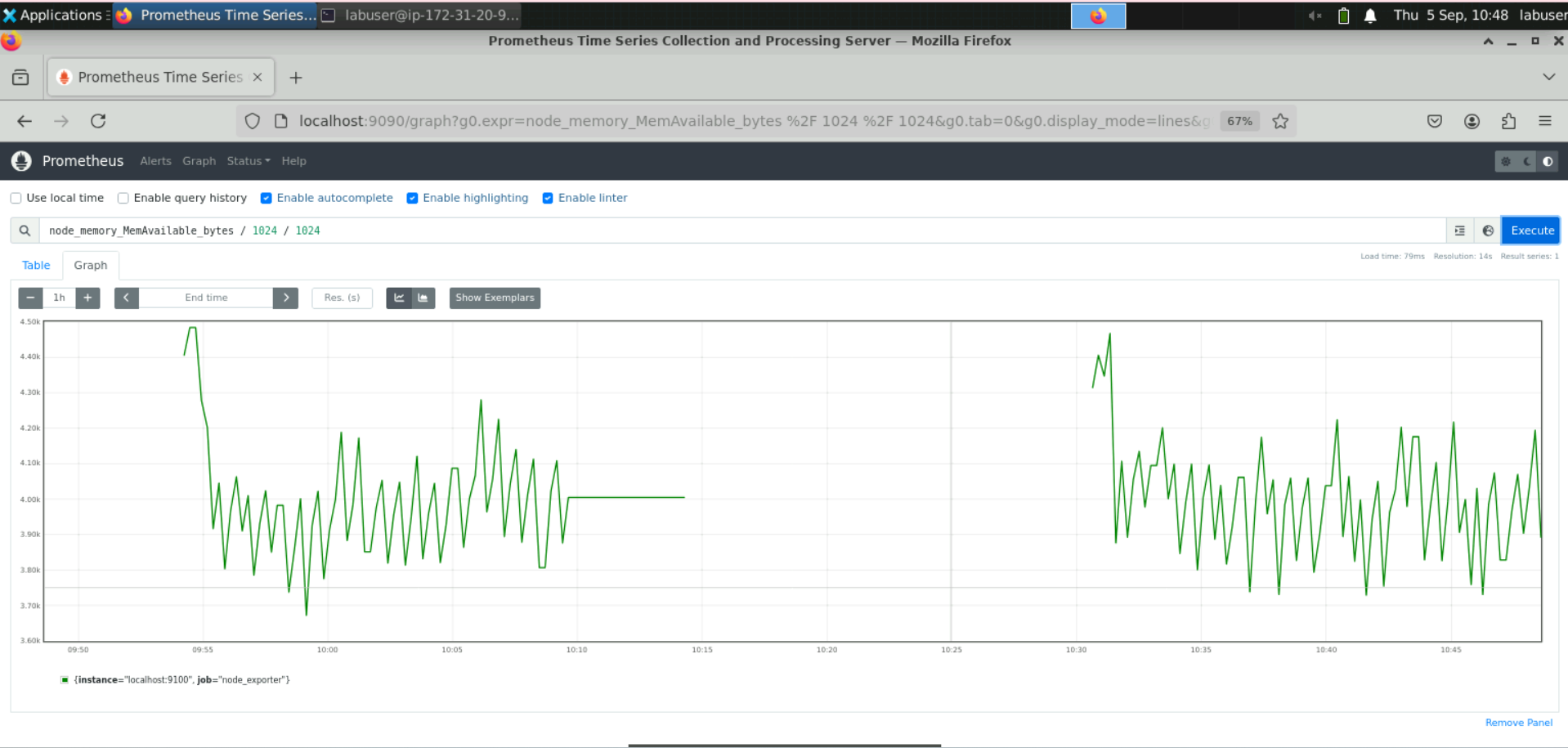
**node\_memory\_MemAvailable\_bytes / 1024 / 1024**

**Explanation:**

* **node\_memory\_MemAvailable\_bytes: This metric gives the amount of memory available for new allocations in bytes.**
* **/ 1024 / 1024: Divides the value by 1024 twice to convert bytes into megabytes (MB).**

**Practical Meaning:**

**This query converts the available memory from bytes to megabytes, making it easier to read and understand, especially when analyzing memory metrics for monitoring or reporting.**



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

* 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:

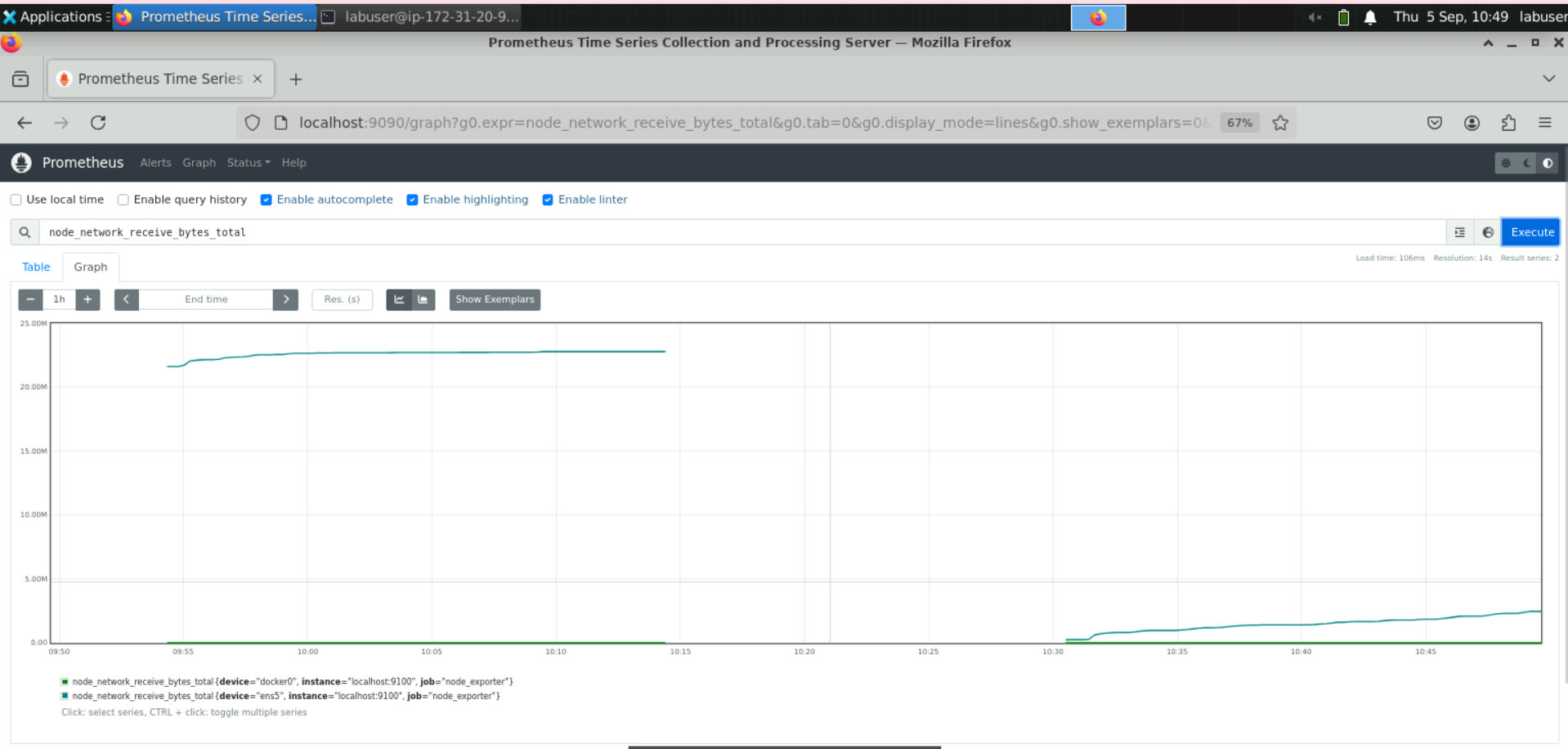
**node\_network\_receive\_bytes\_total**

**Explanation:**

* **node\_network\_receive\_bytes\_total**: This metric represents the total number of bytes received on all network interfaces since the system started. It is a cumulative counter that increments over time.

**Practical Meaning:**

This query retrieves the cumulative total of incoming network traffic (in bytes) on all network interfaces for the nodes being monitored.



* 1. Enter the following query to calculate the average per-second rate of bytes received over the network interface in the last minute:

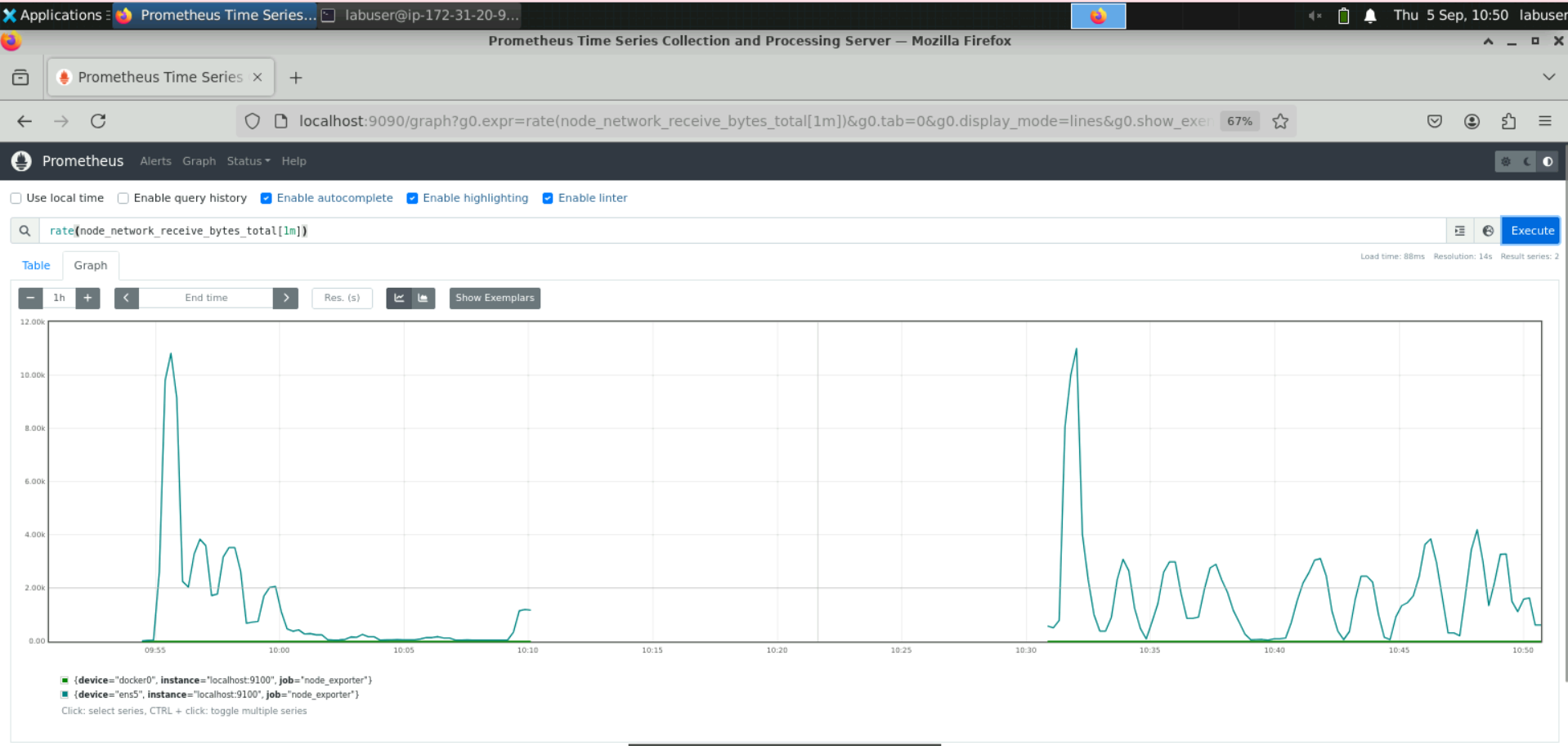
**rate(node\_network\_receive\_bytes\_total[1m])**

**Explanation:**

* **node\_network\_receive\_bytes\_total**: This metric represents the total number of bytes received on all network interfaces since the system started (a cumulative counter).
* **rate(...)**: Calculates the per-second rate of change (bytes per second) for the specified metric over the given time range.
* **[1m]**: Specifies a time window of 1 minute to calculate the rate of incoming traffic.

**Practical Meaning:**

This query calculates the average incoming network traffic rate in bytes per second over the last 1 minute for each network interface.



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