**Lesson-End Project**

**Running Prometheus as a Docker Container for Monitoring**

**Project agenda**: To demonstrate the setup and running of Prometheus as a Docker container for monitoring, enabling efficient tracking and management of system metrics

**Description:** You are a developer in a software company responsible for configuring and deploying Prometheus within a Docker container for real-time monitoring of system metrics. You begin by setting up the Docker environment, pulling the Prometheus image, and ensuring it is correctly configured for your infrastructure. After running the container, you verify that Prometheus efficiently collects and displays metrics.

**Tools required:** Docker and Prometheus

**Prerequisites:** You must have Docker installed in the lab to proceed. Refer Demo 02 for instructions on installing and starting Prometheus.

**Expected deliverables:** A fully configured Prometheus Docker container setup to scrape and visualize system metrics, accessible through a web interface for real-time monitoring and infrastructure management

Steps to be followed:

1. Pull and set up Docker environment
2. Create and edit Prometheus configuration
3. Start the Prometheus container
4. Access the Prometheus web interface

**Step 1: Pull and set up Docker environment**

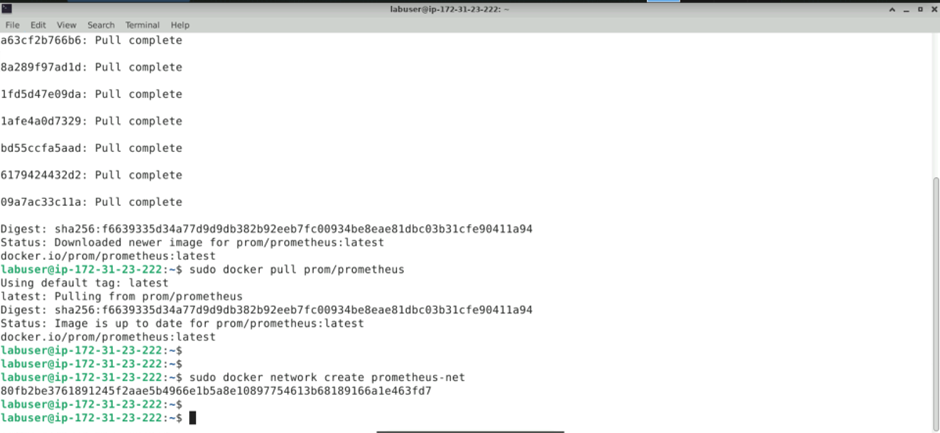
1. Run the following command on the terminal to pull the Prometheus Docker image from the official Docker Hub repository:

**sudo docker pull prom/prometheus**



1. Execute the following command on the terminal to create a Docker network:

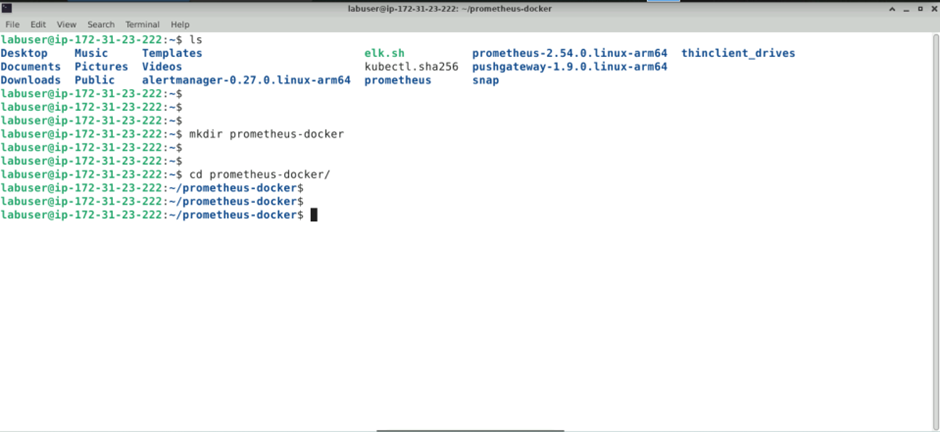
**sudo docker network create prometheus-net**



1. Run the following commands to create a new directory named **prometheus-docker**, store the Prometheus configuration file in it, and then change to the newly created directory as follows:

**mkdir prometheus-docker**

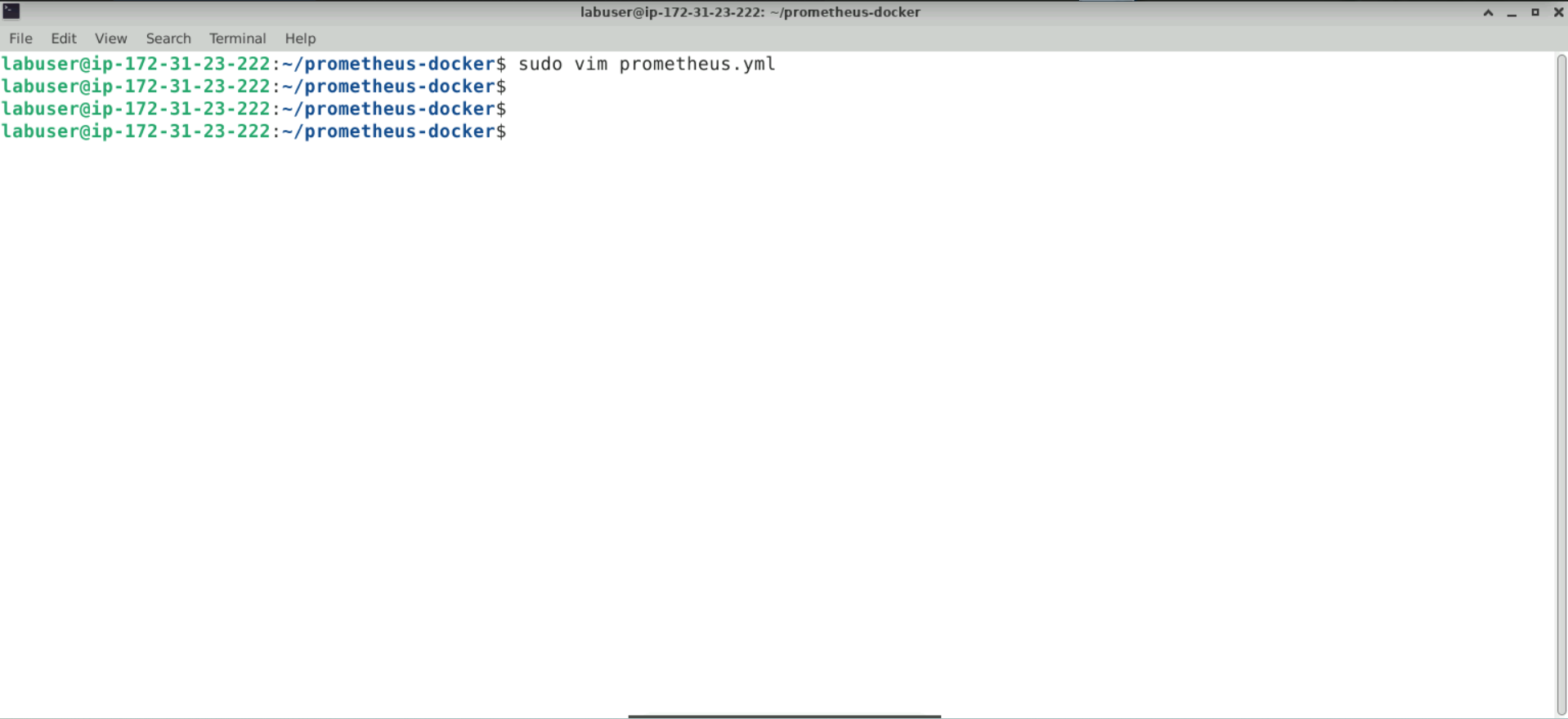
**cd prometheus-docker/**



**Step 2: Create and edit Prometheus configuration**

* 1. Run the command given below to create and edit a file named **prometheus.yml** using **vim** or a preferred text editor within the newly created directory:

**sudo vim prometheus.yml**



The editor appears as follows:



* 1. Press **I** or **i** to switch to **INSERT** mode, then copy and paste the following configuration into the file:

**global:**

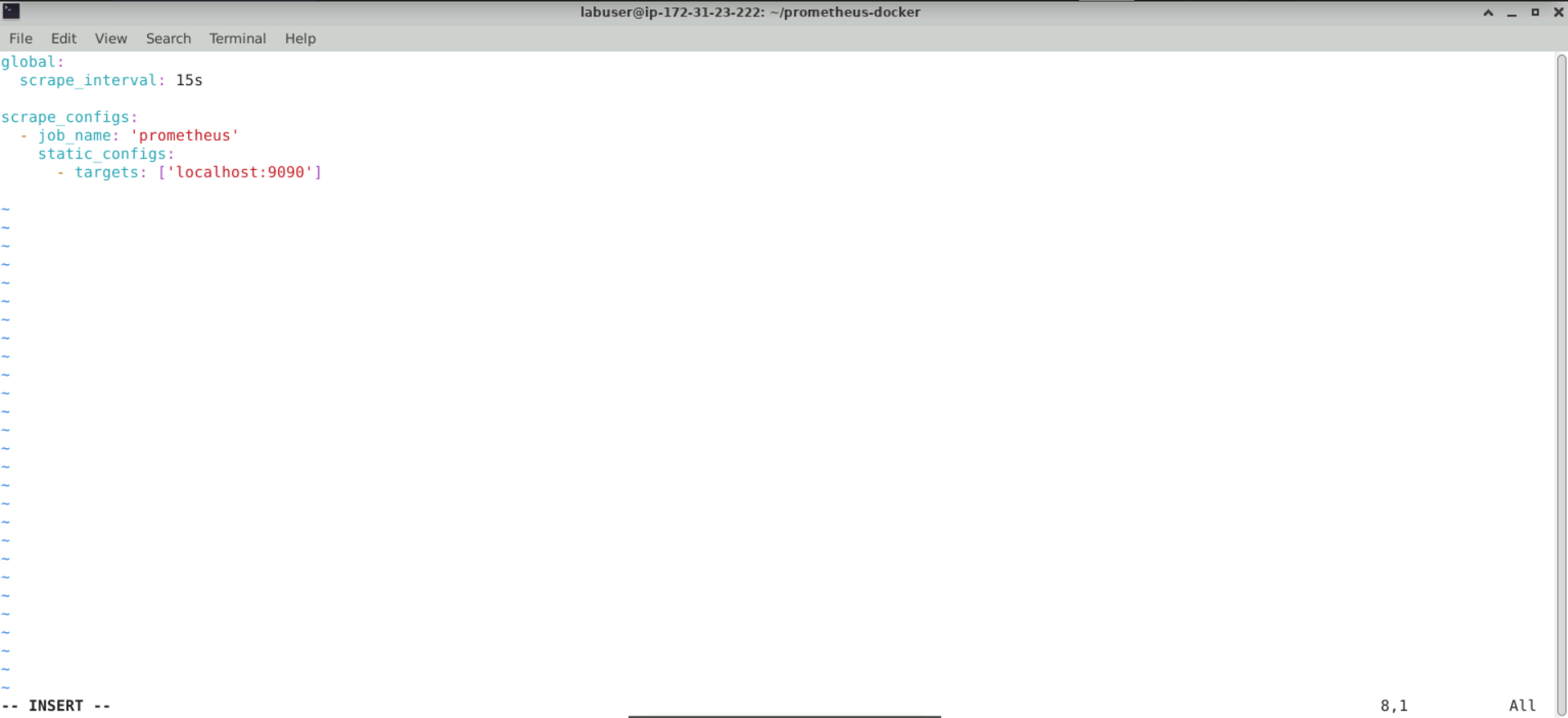
**scrape\_interval: 15s**

**scrape\_configs:**

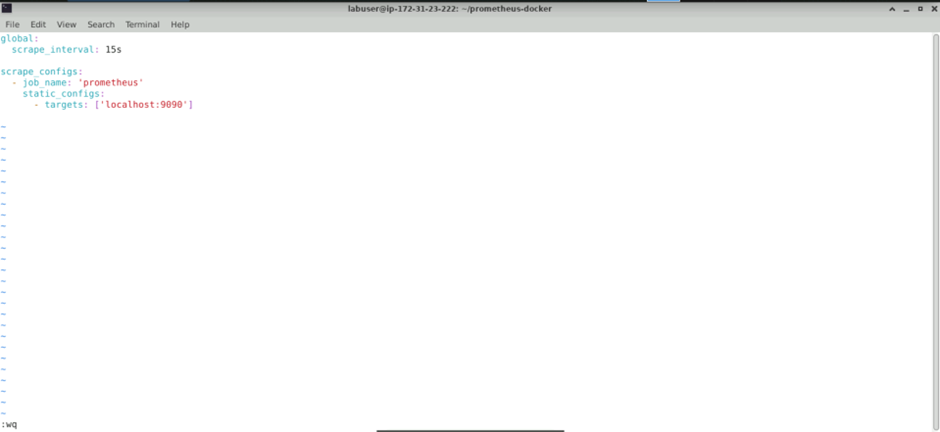
**- job\_name: 'prometheus'**

**static\_configs:**

**- targets: ['localhost:9090']**



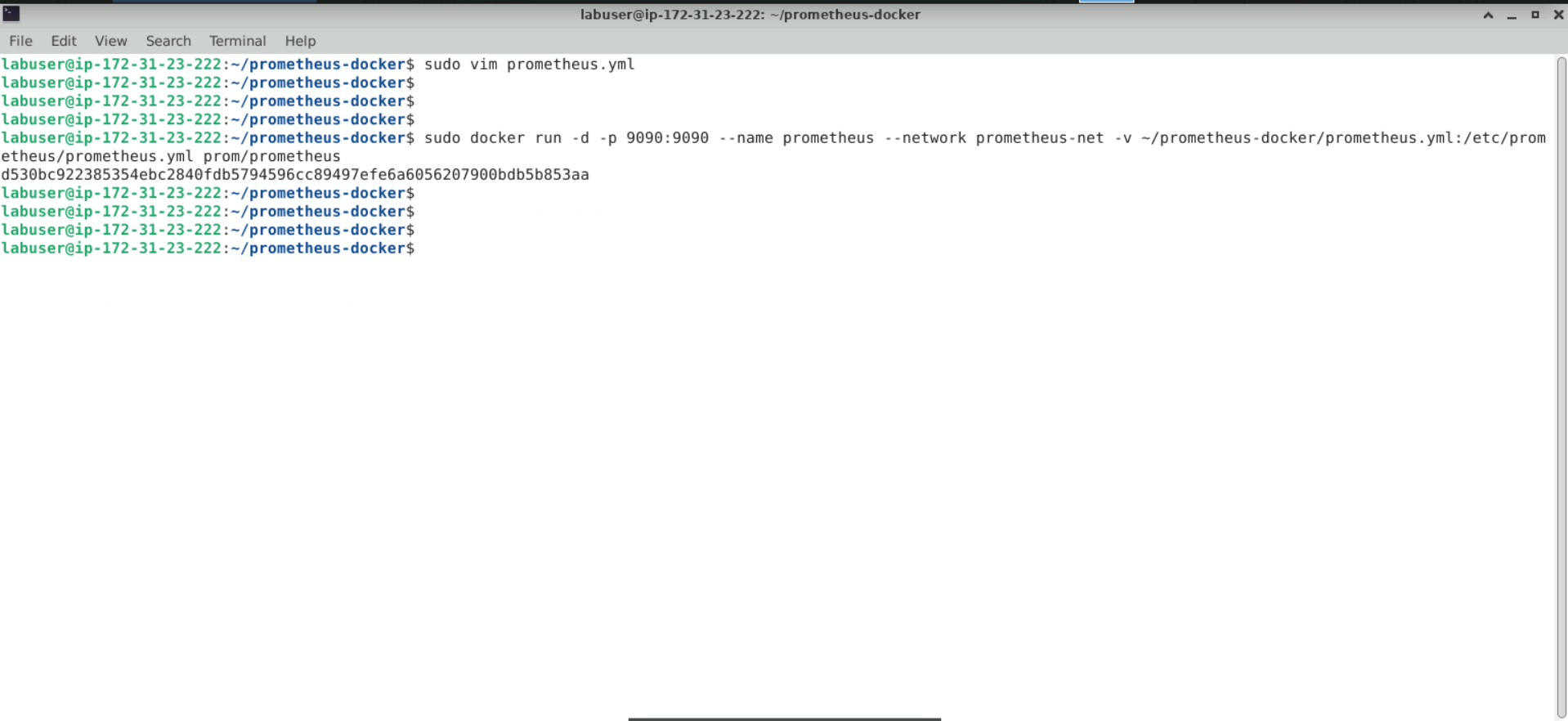
* 1. Press **Esc** to exit INSERT mode, then type **:wq** to save and exit the file



**Step 3: Start the Prometheus container**

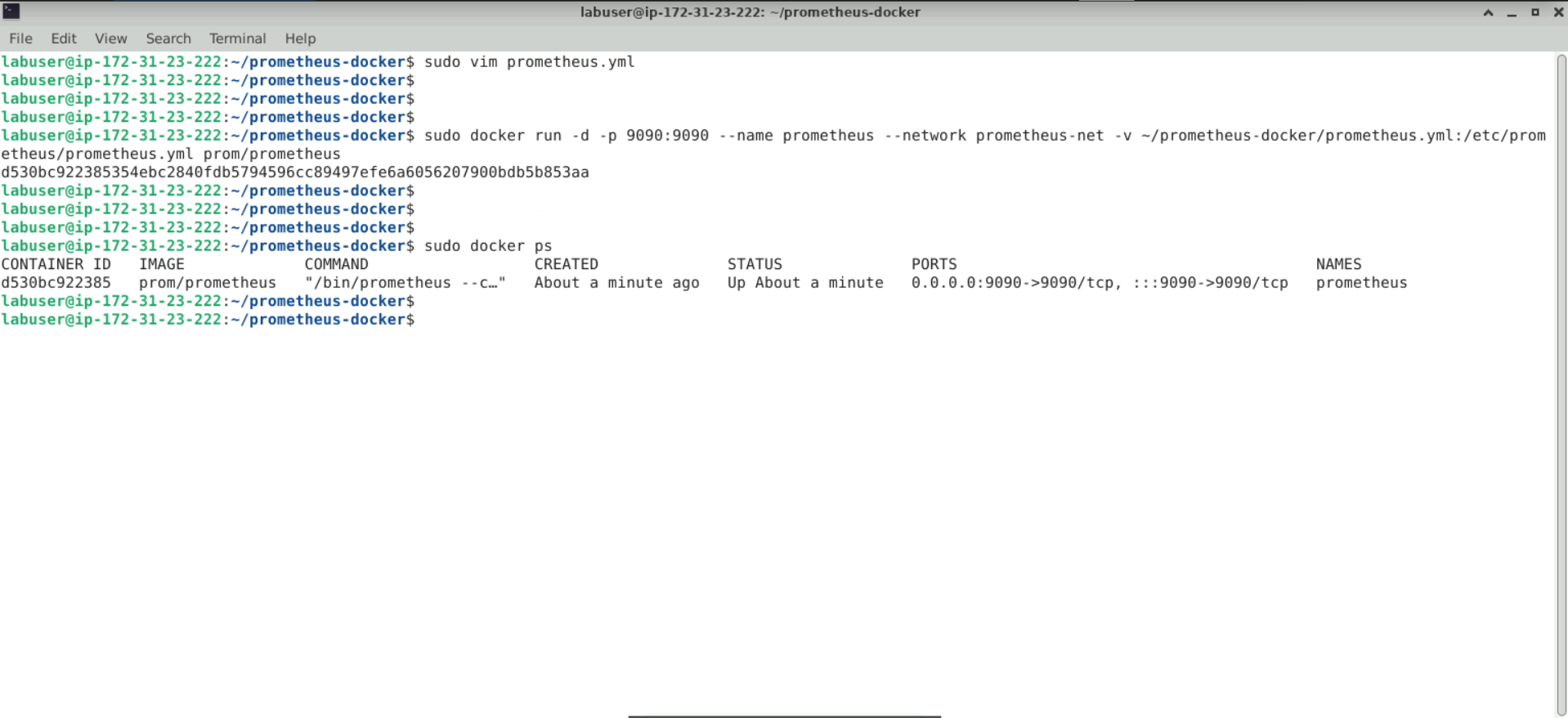
* 1. Run the following command to start a Prometheus container using the Docker image pulled in **Step 1**:

**sudo docker run -d -p 9090:9090 --name prometheus --network prometheus-net -v ~/prometheus-docker/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus**

****

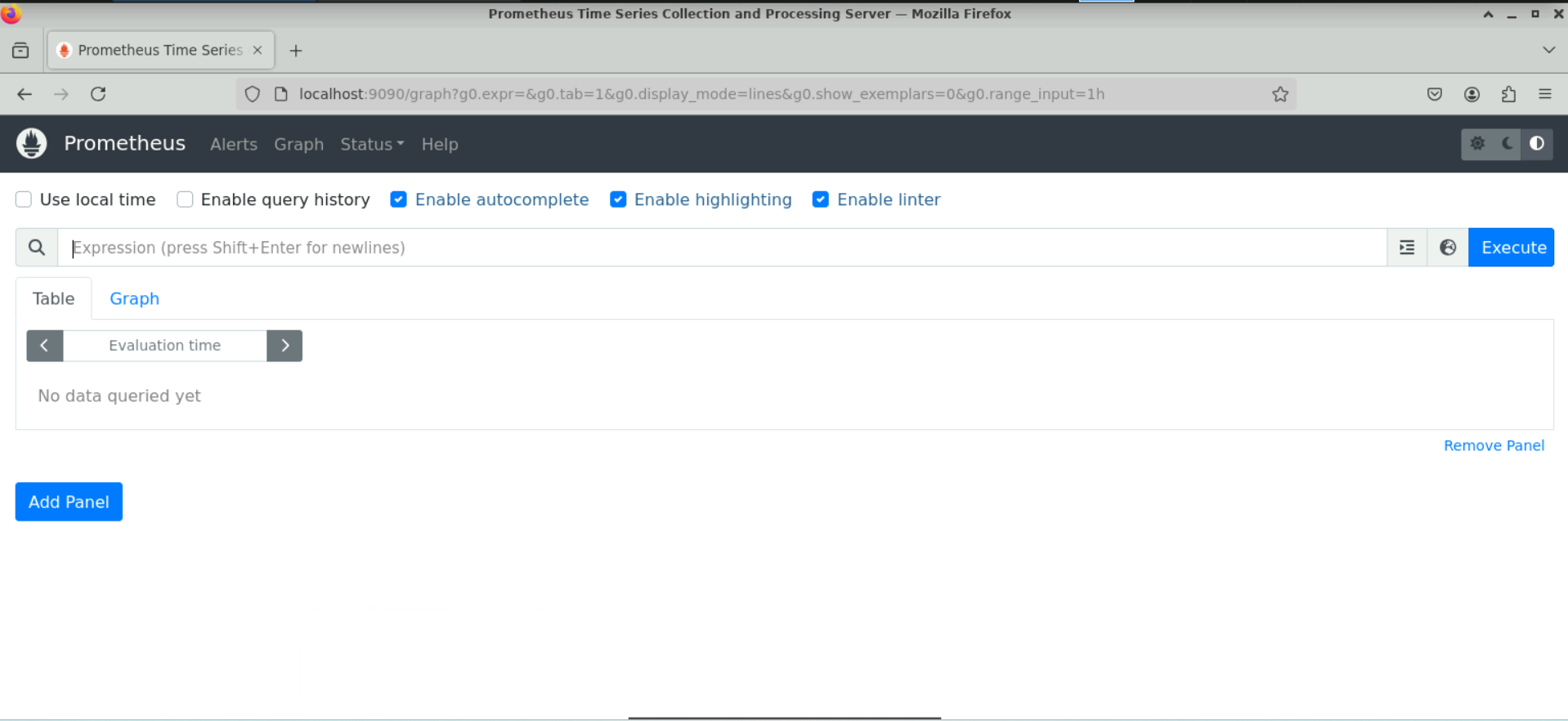
**Note:** If the error “**The container name ‘/prometheus’ is already in use by another container**” occurs, run the command: **docker rm -f prometheus** and then restart the Prometheus container.

* 1. Execute the following command to list all running Docker containers:   
     **sudo docker ps**



**Step 4: Access the Prometheus web interface**

1. Enter the URL **http://localhost:9090** in the address bar of your preferred web browser to access the Prometheus UI as shown below:

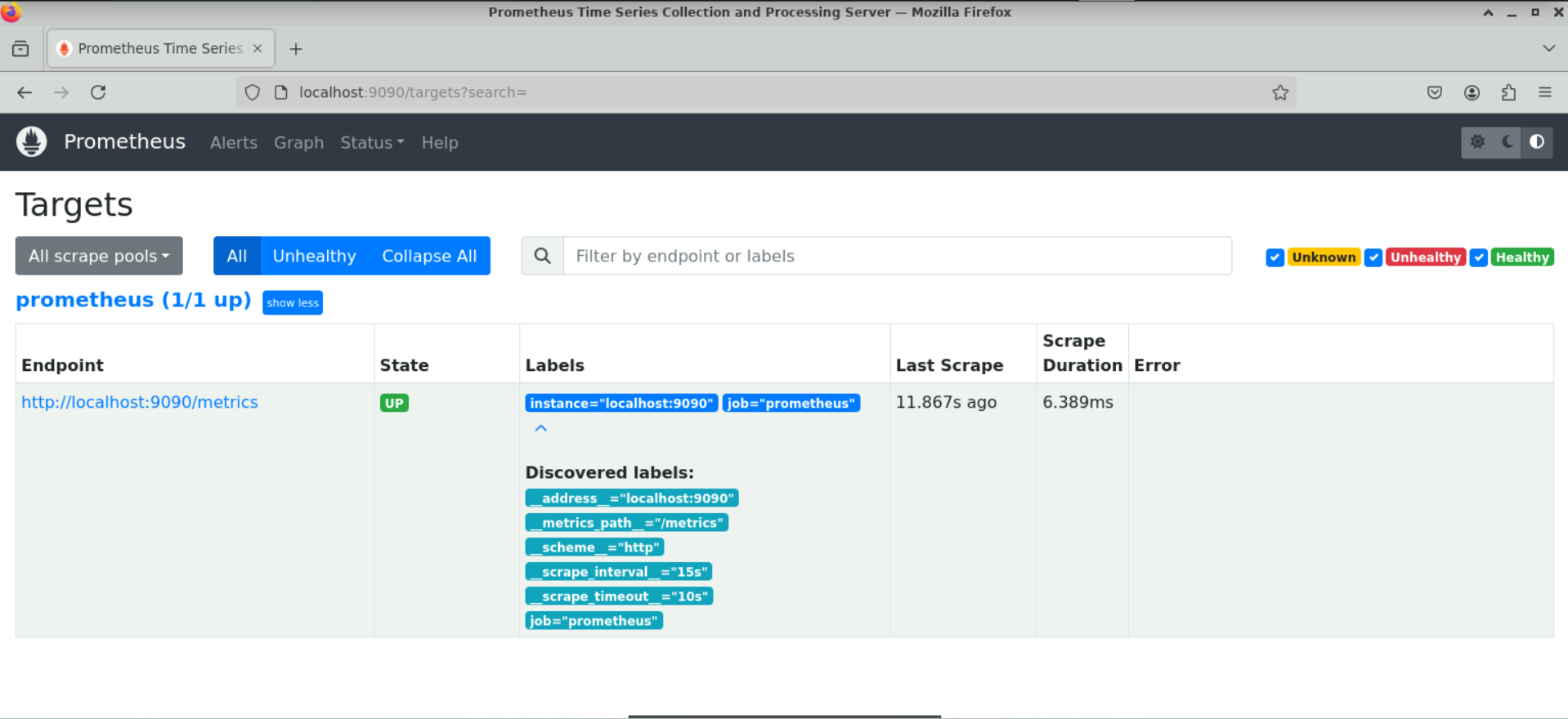


**Note:** This URL will connect to the Prometheus web interface running in the Docker container.

1. Navigate to the **Targets** section, which shows the health status and last scrape duration of the monitored endpoints

A screenshot of a computer

Description automatically generated



1. Navigate back to the dashboard and click on the **Graph** tab

A screenshot of a computer

Description automatically generated

1. Enter the following command in the Expression section:

**rate(prometheus\_http\_requests\_total[1m])**

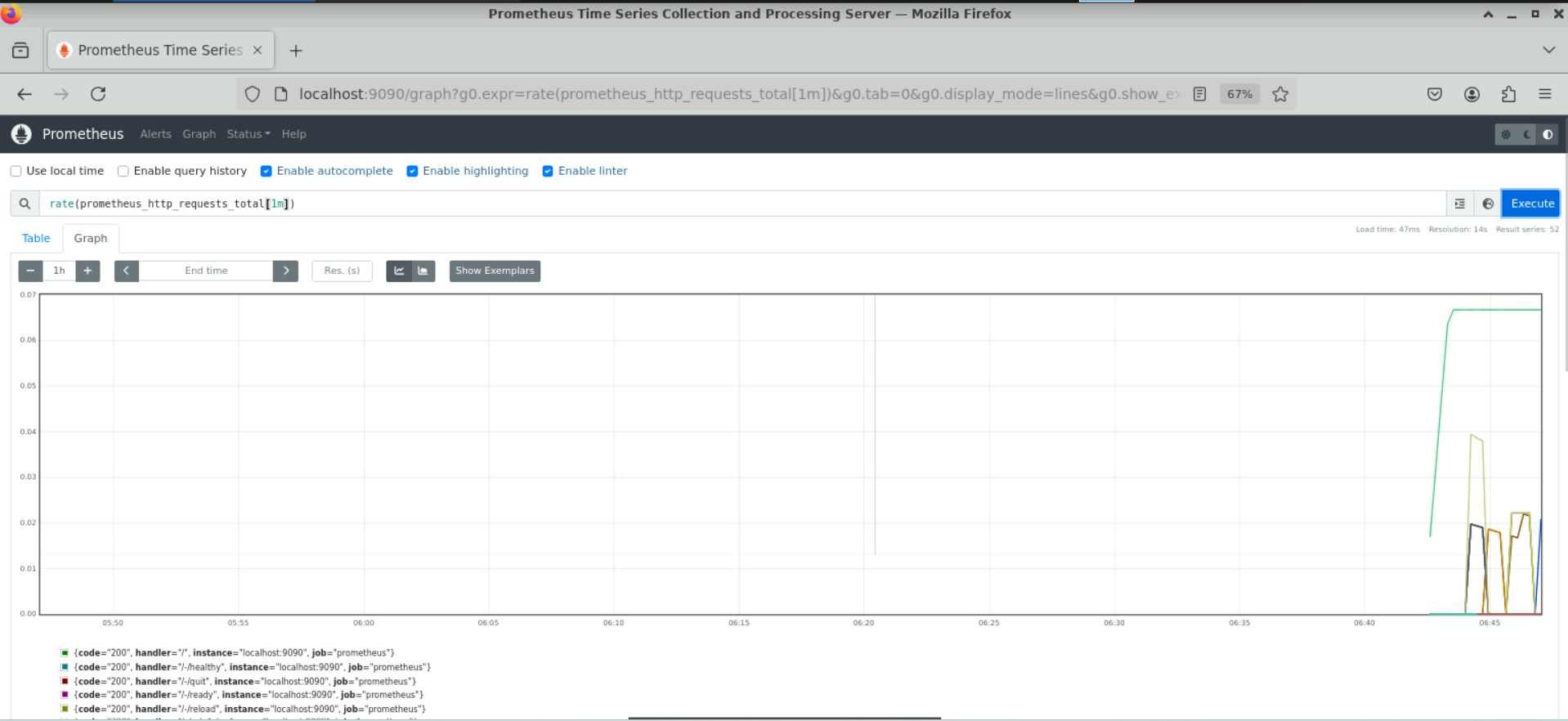
A screenshot of a computer

Description automatically generated

1. Click on the **Execute** button to visualize the result

A screenshot of a computer

Description automatically generated



By following these steps, you have successfully set up and run Prometheus as a Docker container. This enables real-time monitoring of system metrics for efficient infrastructure tracking and management.