Chapter 7 Lab: Setting up Prometheus and Grafana

Overview

Prometheus and Grafana are two powerful open source tools widely used in the monitoring and visualization of metrics in cloud native environments. They serve complementary roles: Prometheus is primarily used for metrics collection and alerting, while Grafana is leveraged for visualizing the data collected by Prometheus. Let's explore both tools, including how they work together to provide a comprehensive monitoring solution.

Prerequisites

Ensure you have an Ubuntu 24.04 host. You can follow the previous lab to provision a virtual machine for yourself.

Exercise 7.1: Installing Prometheus and Grafana

Set up the Prometheus server on an Ubuntu machine.

1. Download the latest Prometheus release, by executing the following command.

```
$ cd /tmp
$ curl -LO https://github.com/prometheus/prometheus/releases/download/v3.7.0/prometheus-3.7.0.linux-amd64.tar.gz
```

2. Extract the contents by executing the following command.

```
$ tar xvf prometheus-3.7.0.linux-amd64.tar.gz
prometheus-3.7.0.linux-amd64/
prometheus-3.7.0.linux-amd64/NOTICE
prometheus-3.7.0.linux-amd64/LICENSE
prometheus-3.7.0.linux-amd64/prometheus.yml
prometheus-3.7.0.linux-amd64/promtool
prometheus-3.7.0.linux-amd64/prometheus
```

3. Move the extracted binaries to the `bin` folder by executing the below commands.

```
$ sudo mv prometheus-3.7.0.linux-amd64/prometheus /usr/local/bin/
$ sudo mv prometheus-3.7.0.linux-amd64/promtool /usr/local/bin/
```

4. Create a directory for the Prometheus configuration file and make use of the default configuration.

```
$ sudo mkdir /etc/prometheus
$ sudo mv prometheus-3.7.0.linux-amd64/prometheus.yml
/etc/prometheus/prometheus.yml
```

5. Create a user for Prometheus. It's always a good practice to run services using a dedicated user.

```
$ sudo useradd --no-create-home --shell /bin/false prometheus
```

6. Set ownership, make Prometheus own its files.

```
$ sudo chown prometheus:prometheus /usr/local/bin/prometheus
$ sudo chown prometheus:prometheus /usr/local/bin/promtool
$ sudo chown -R prometheus:prometheus /etc/prometheus
```

7. To allow Prometheus to run as a service, create a systemd service file:

```
cat > /etc/systemd/system/prometheus.service << EOF

[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
```

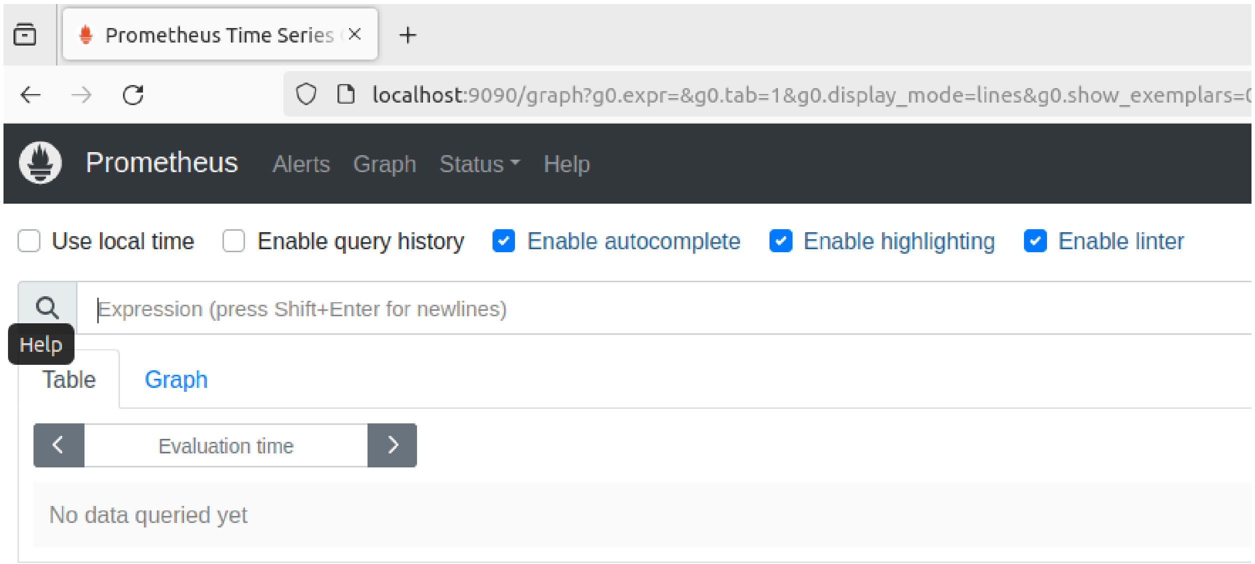
```
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /etc/prometheus/ \
    --web.listen-address=0.0.0.0:9090

[Install]
WantedBy=multi-user.target
EOF
```

8. Enable and start the Prometheus service:

```
$ sudo systemctl daemon-reload
$ sudo systemctl start prometheus
$ sudo systemctl enable prometheus
```

```
Created symlink
/etc/systemd/system/multi-user.target.wants/prometheus.service →
/etc/systemd/system/prometheus.service.
```

9. Verify if you are able to access the service by opening the browser and executing http://localhost:9090. If you are using a cloud-based VM, then you can also use your public IP to access the application. Use the command **`curl ifconfig.io`** on the command line to retrieve your public IP.



10. Let us install Grafana, add the apt Grafana repository and the GPG key to verify the packages.

```
$ sudo apt-get install -y apt-transport-https
software-properties-common wget
$ sudo mkdir -p /etc/apt/keyrings/
$ wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo
tee /etc/apt/keyrings/grafana.gpg > /dev/null

$ echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg]
https://apt.grafana.com stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
```

11. Install Grafana by executing the following command:

```
$ sudo apt-get -y update
$ sudo apt-get install -y grafana

additional packages will be installed:
  musl
The following NEW packages will be installed:
  grafana musl
0 upgraded, 2 newly installed, 0 to remove and 6 not upgraded.
Need to get 188 MB of archives.
After this operation, 694 MB of additional disk space will be used.
Get:1 https://apt.grafana.com stable/main amd64 grafana amd64 12.2.1
[187 MB]
```

```
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu noble/universe
amd64 musl amd64 1.2.4-2 [416 kB]
Fetched 188 MB in 5s (40.4 MB/s)
Selecting previously unselected package musl:amd64.
(Reading database ... 72269 files and directories currently
installed.)
Preparing to unpack .../musl_1.2.4-2_amd64.deb ...
Unpacking musl:amd64 (1.2.4-2) ...
Selecting previously unselected package grafana.
Preparing to unpack .../grafana_12.2.1_amd64.deb ...
Unpacking grafana (12.2.1) ...
Setting up musl:amd64 (1.2.4-2) ...
Setting up grafana (12.2.1) ...
info: Selecting UID from range 100 to 999 ...

info: Adding system user `grafana' (UID 110) ...
info: Adding new user `grafana' (UID 110) with group `grafana' ...
info: Not creating home directory `/usr/share/grafana'.
### NOT starting on installation, please execute the following
statements to configure grafana to start automatically using systemd
 sudo /bin/systemctl daemon-reload
 sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
 sudo /bin/systemctl start grafana-server
Processing triggers for man-db (2.12.0-4build2) …
```

12. Start and enable Grafana by executing the below command:

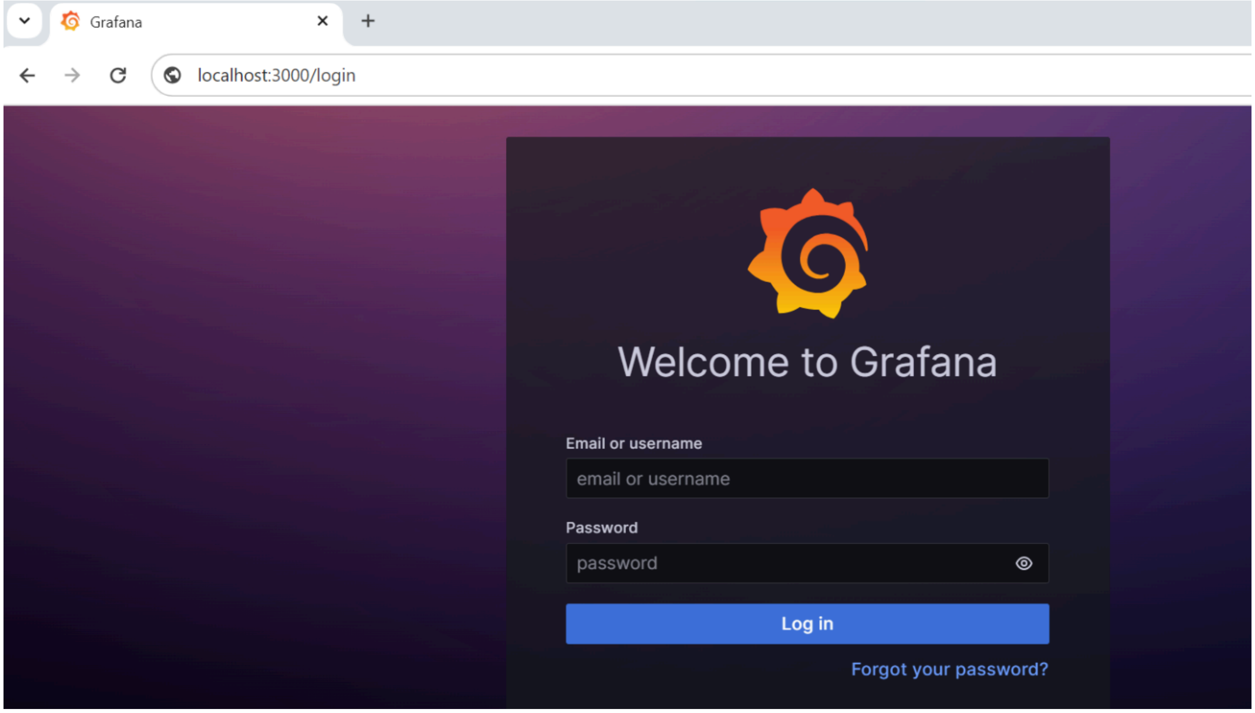
```
$ sudo systemctl start grafana-server
$ sudo systemctl enable grafana-server
```

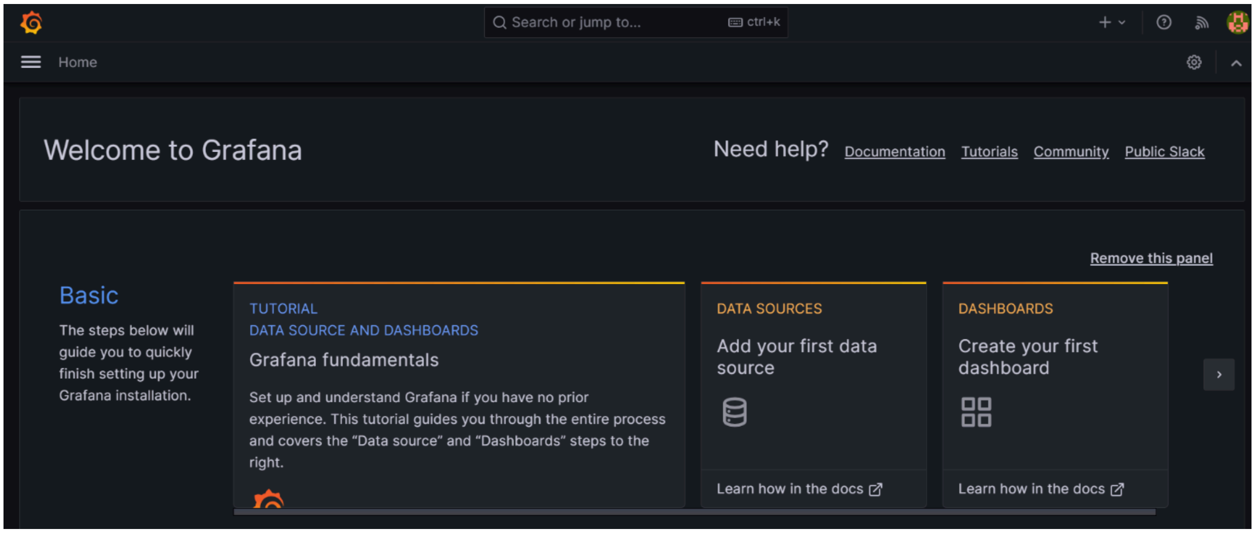
```
Synchronizing state of grafana-server.service with SysV service script
with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink
/etc/systemd/system/multi-user.target.wants/grafana-server.service →
/lib/systemd/system/grafana-server.service.
```

13. Configure Grafana to use Prometheus:

- Access Grafana web interface. Open your web browser and go to **`http://localhost:3000`**. The default login is **`admin`** for both username and password.

If you are using a cloud-based VM, then you can also use your public IP to access the application. Use the command **`curl`** [`ifconfig.io`] on the command line to retrieve your public IP.

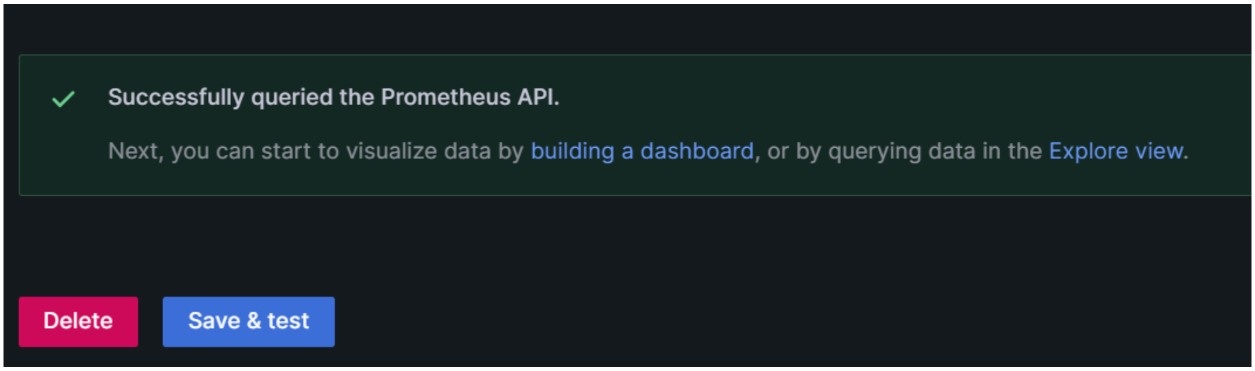




The steps below will guide you to quickly finish setting up your Grafana installation.

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

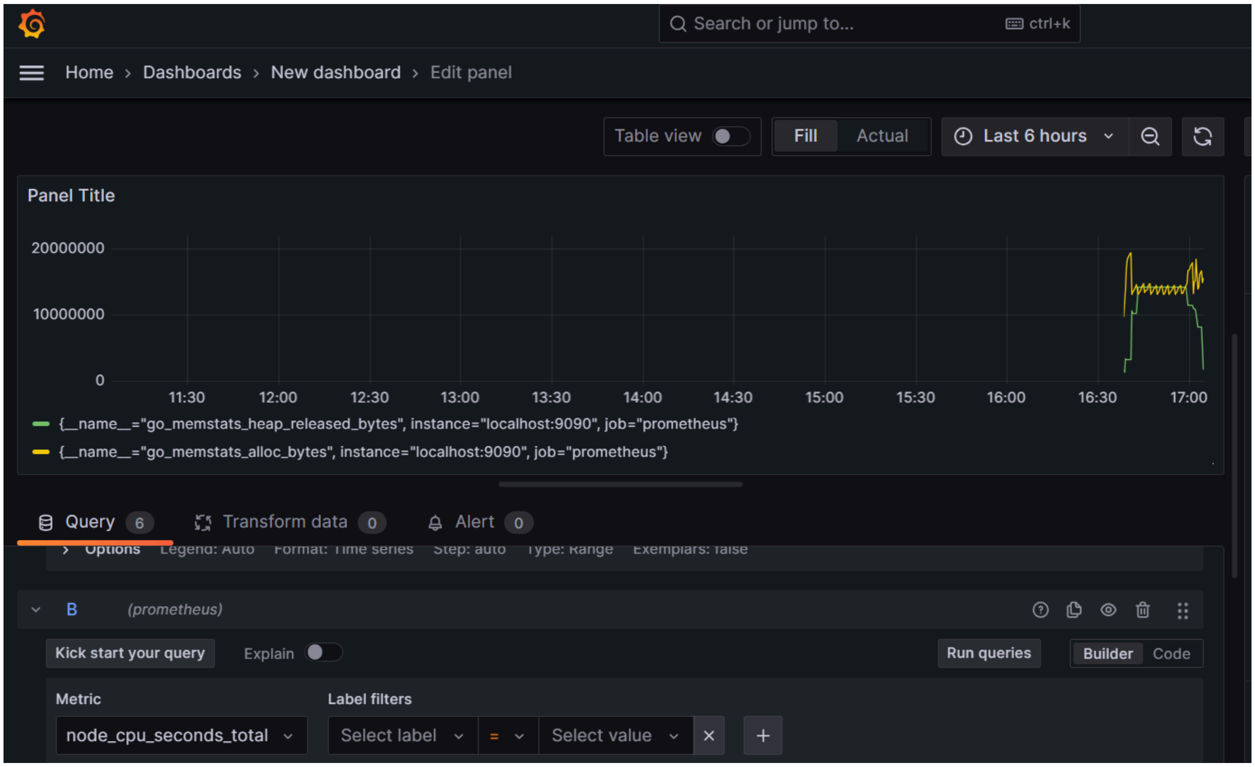
- Add Prometheus as a Data Source. Go to "Connections" > "Data Sources" > "Add Data Source". Select "Prometheus" as the type. For the URL, enter **`http://localhost:9090 and click "Save & Test`**.



✓ Successfully queried the Prometheus API.

Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.

- Create a Dashboard. Now, you can create a new dashboard and add panels to visualize metrics from Prometheus.
- In the panel configuration view, you'll see a query editor. Since you've set Prometheus as the data source, it should be selected by default.
- In the Metrics field, you can write your Prometheus query. For example, if you want to monitor the CPU usage of a system, you might use a query like **`node_cpu_seconds_total`**. This metric and query depend on the data exposed to Prometheus. For a simple start, you can use **`up{job="prometheus"}`** which will show you the up/down status of Prometheus targets.
- After entering your query, you should see the graph update in real time below the query editor. Adjust the time range and graph settings as needed. You can set the title of the panel under the Panel title section by clicking on the title name at the top.
- Save the Panel
- Once you are satisfied with the configuration, click the Apply button in the upper right corner to add the panel to your dashboard.
- You will be taken back to the dashboard view where you can see your new panel. Click the Save dashboard icon at the top, give your dashboard a name, and optionally a folder and tags, then click Save.
- To further customize your dashboard, you can add more panels by clicking the + button in the dashboard view and selecting Add Panel. Repeat the steps above to configure additional panels as needed, each with its own Prometheus query for different metrics you want to monitor.



🟩 {__name__="go_memstats_heap_released_bytes", instance="localhost:9090", job="prometheus"}

🟨 {__name__="go_memstats_alloc_bytes", instance="localhost:9090", job="prometheus"}

› Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

Metric: node_cpu_seconds_total