Monitoring with Prometheus and Grafana

What is Prometheus?

Prometheus is an open-source system monitoring application that collects server metrics at regular intervals. It can monitor itself as well as external clients.

Architecture

Client

- Each client must run a metrics collation tool known as an ***Exporter*** to collect and expose its internal statistics.
- Prometheus cannot monitor an external node that is not collecting data in this manner.
- Although various statistic collectors can be used on the client nodes, Prometheus recommends its own ***Node Exporter*** tool.
- Prometheus Node Exporter collects a large number of hardware and kernel metrics, including CPU and memory use.
- A full list of the available metrics collected can be found on the Node Exporter GitHub page.
- Node Exporter is closely integrated with Prometheus and shares the same data format.
- No conversion or pre-processing work is required to use Node Exporter and Prometheus together.
- Node Exporter should be installed on every client to monitor. Node Exporter uses port `9100`, while Prometheus uses port `9090`.

Prometheus server

- **HTTP Polling:**
    - Prometheus server uses the HTTP protocol to poll the client servers for the required data.
- **TSDB:**
    - It maintains a time-series database of the results, polling each client at a predefined interval.
    - Prometheus stores the discrete snapshots along with the timestamps indicating when the statistics were retrieved.
    - Prometheus retains the metrics it collects, allowing it to build an overview of the long-term performance of the client.

Some features

- It is open source and independent of any company
- Each Prometheus server is autonomous and does not require distributed storage or a central server
- It is highly reliable and easy to install

- Different clients can be polled at different intervals
- It uses a flexible multi-dimensional data model with support for time-series data and key-value pairs
- It supports the **PromQL query language** for retrieving and analyzing data
- It includes the **AlertManger component**. AlertManager receives alerts from clients and pushes them out to different subscribers
- Prometheus can feed its metrics into visualization tools such as **Grafana**
- It supports Docker containerization and Kubernetes

What is Grafana?

- Grafana is a visualization application available in both open-source and enterprise editions.
- Grafana does not collect any metrics from the clients and it does not store any data.
- Instead, the **Grafana displays the metrics collected by Prometheus** or another data source in an intuitive and visually-appealing format.
- Grafana presents the data using a dashboard.
- A dashboard is a template to define the values to display and how to display them.
- Dashboards are customizable and you can create your own.

Steps to install and configure

1. Download and install Prometheus on the monitoring system
2. Configure Prometheus to run as a service
3. Install Node Exporter on all clients
4. Configure Prometheus to monitor the clients
5. Install and deploy the Grafana server
6. Integrate Grafana and Prometheus
7. Import a Dashboard for the Node Exporter Statistics

Download and install Prometheus on the monitoring server

Ubuntu 22.04 LTS

```
sudo apt update
```

```
wget https://github.com/prometheus/prometheus/releases/download/v2.37.6/prometheus-2.37.6.linux-amd64.tar.gz
```

```
or
https://github.com/prometheus/prometheus/releases/download/v2.53.1/prometheus-2.53.1.linux-amd64.tar.gz
```

```
tar xvfz prometheus-*.tar.gz
```

Create two new directories for Prometheus to use.

- The **/etc/prometheus** directory stores the Prometheus configuration files.
- The **/var/lib/prometheus** directory holds application data.

```
sudo mkdir /etc/prometheus /var/lib/prometheus
```

Move the **prometheus** and **promtool** directories to the **/usr/local/bin/** directory. This makes Prometheus accessible to all admins.

```
cd prometheus-2.37.6.linux-amd64
or
cd prometheus-2.53.1.linux-amd64/

sudo mv prometheus promtool /usr/local/bin/
```

Move the **prometheus.yml** YAML configuration file to the **/etc/prometheus** directory.

```
sudo mv prometheus.yml /etc/prometheus/prometheus.yml
```

Move **consoles** and **console_libraries** directories to the **etc/prometheus** directory.

```
sudo mv consoles/ console_libraries/ /etc/prometheus/
```

Verify if Prometheus is installed.

```
prometheus --version

prometheus, version 2.37.6 (branch: HEAD, revision:
8ade24a23af6be0f35414d6e8ce09598446c29a2)
  build user:       root@5f96027a7c3e
  build date:       20230220-09:36:40
  go version:       go1.19.6
  platform:         linux/amd64
```

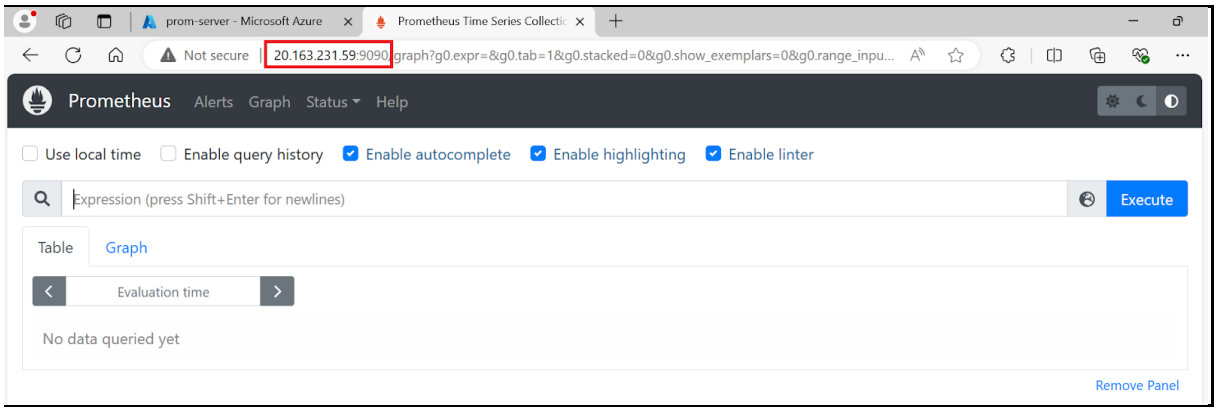
Configure Prometheus as a Service

```
sudo useradd -rs /bin/false prometheus

sudo chown -R prometheus: /etc/prometheus /var/lib/prometheus

sudo nano /etc/systemd/system/prometheus.service

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

[Service]
User=prometheus
```

```
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries \
    --web.listen-address=0.0.0.0:9090 \
    --web.enable-lifecycle \
    --log.level=info

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

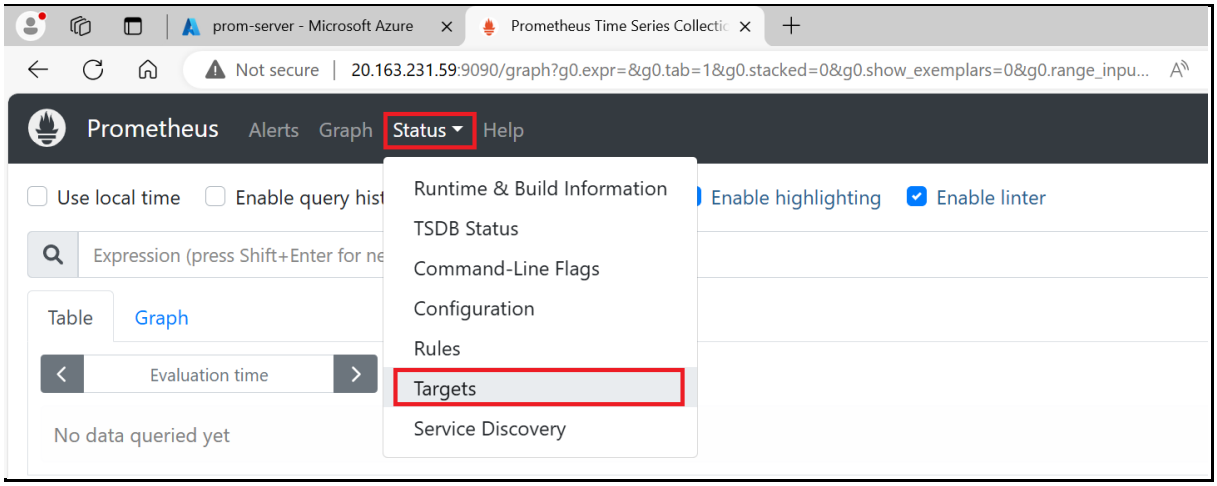
```
sudo systemctl daemon-reload

sudo systemctl enable prometheus

sudo systemctl start prometheus
sudo systemctl status prometheus
```

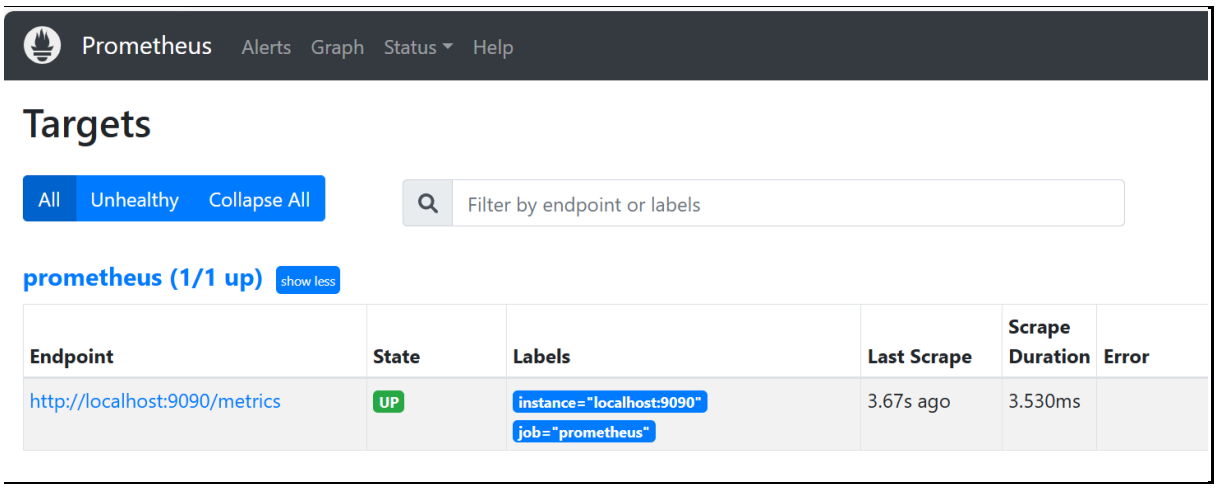
Access the Prometheus web interface and dashboard at **http://server_ip_addr:9090**



The default **prometheus.yml** file contains a directive to scrape the local host.

Click **Status and Targets** to list all the targets. Prometheus should display the local Prometheus service as the only target.





Install and Configure Node Exporter on the Client

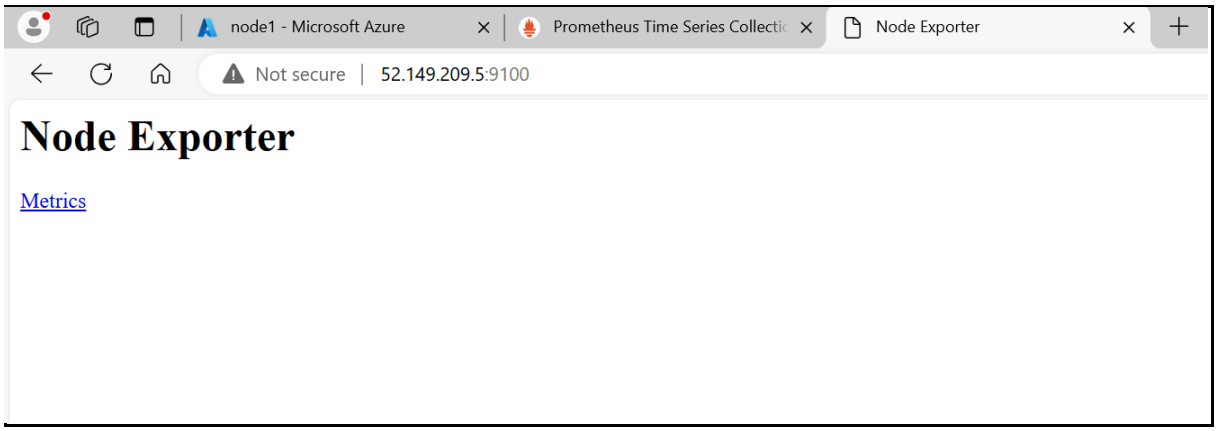
```
sudo apt update

wgethttps://github.com/prometheus/node_exporter/releases/download/v1.5.0/node_exporter-1.5.0.linux-amd64.tar.gz

or

https://github.com/prometheus/node_exporter/releases/download/v1.8.2/node_exporter-1.8.2.linux-amd64.tar.gz

tar xvfz node_exporter-*.tar.gz

sudo mv node_exporter-1.5.0.linux-amd64/node_exporter /usr/local/bin

sudo mv node_exporter-1.8.2.linux-amd64/node_exporter /usr/local/bin

node_exporter
```

```
Come out by pressing Control+C
```

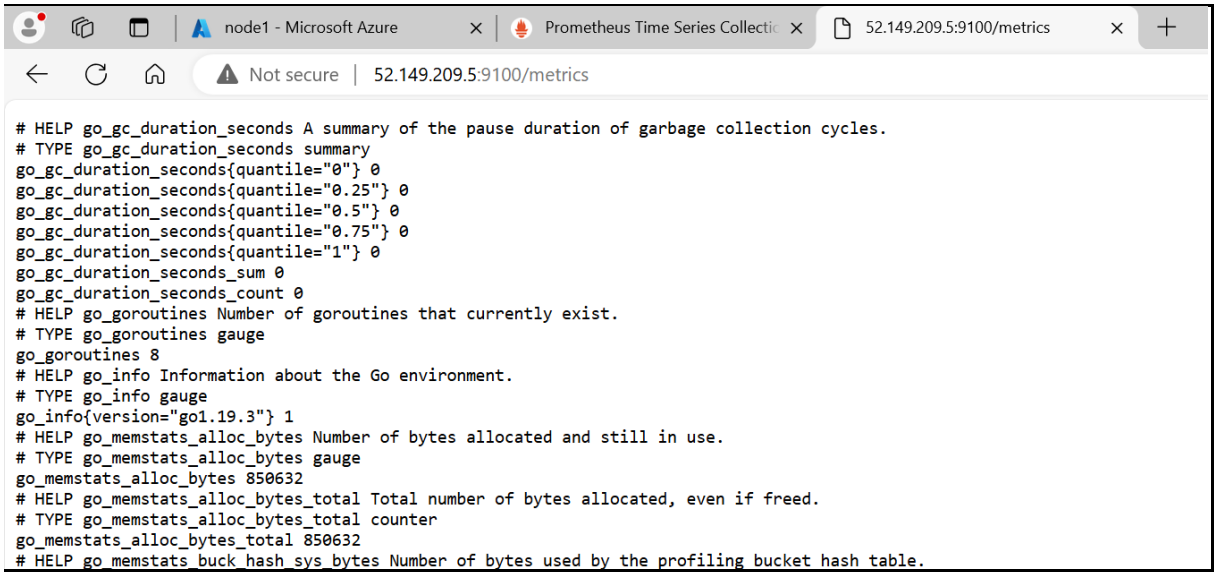
Configure as a service

```
sudo useradd -rs /bin/false node_exporter

sudo nano /etc/systemd/system/node_exporter.service

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

[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target

sudo systemctl enable node_exporter

sudo systemctl daemon-reload
sudo systemctl start node_exporter
sudo systemctl status node_exporter
```

Use a web browser to visit port 9100 on the client node, for example, http://ip_addr:9100



A page titled **Node Exporter** is displayed along with a link reading Metrics. Click the **Metrics link** and confirm the statistics are being collected.



Configure Prometheus to Monitor Client Nodes

Perform on the Prometheus Server

The client nodes are now ready for monitoring.
To add clients to `prometheus.yml`, follow the steps below:
On the monitoring server running Prometheus, open `prometheus.yml` for editing.

```
sudo nano /etc/prometheus/prometheus.yml
```

Locate the section **scrape_configs**, which contains a list of jobs.

It currently lists a single job named prometheus.

This job monitors the local **Prometheus task** on port 9090.

Beneath the prometheus job, add a second job having the **job_name of remote_collector**.

```
...
- job_name: "remote_collector"
  scrape_interval: 10s
  static_configs:
    - targets: ["remote_addr:9100"]
```

The following is in my case:

```
- job_name: "remote_collector"
  scrape_interval: 10s
  static_configs:
    - targets: ["10.0.0.5:9100"]
```

To immediately refresh Prometheus, restart the prometheus service

```
sudo systemctl restart prometheus
```

Using a web browser, revisit the Prometheus web portal at port 9090 on the monitoring server. Select Status > Targets. A second link for the **remote_collector job** is displayed, leading to port 9100 on the client. Click the link to review the statistics.

Install and Deploy the Grafana Server

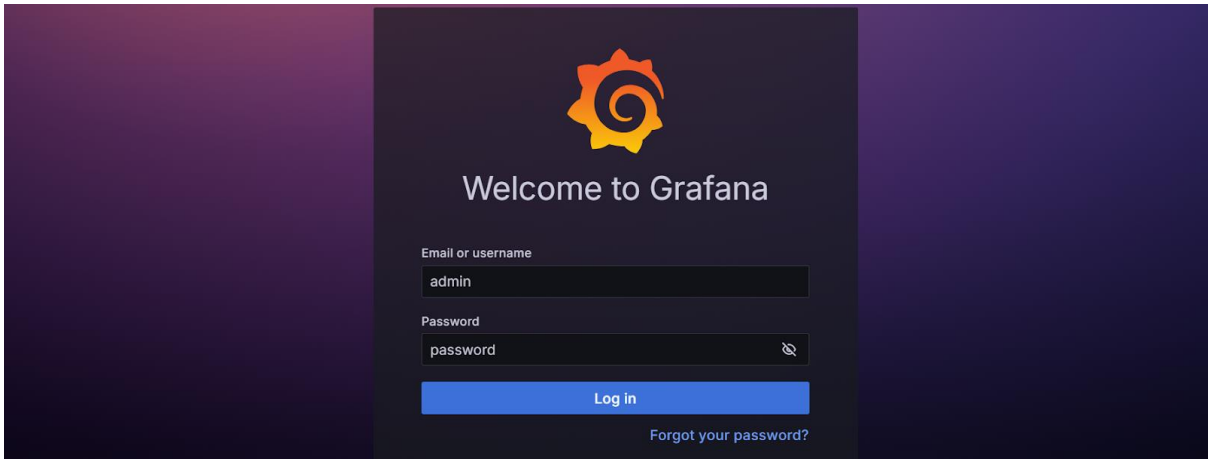
Prometheus is now collecting statistics from the clients listed in the scrape_configs section of its configuration file. However, the information can only be viewed as a raw data dump. The statistics are difficult to read and not too useful. Grafana provides an interface for viewing the statistics collected by Prometheus.

```
sudo apt-get install -y apt-transport-https software-properties-common

sudo wget -q -O /usr/share/keyrings/grafana.key https://apt.grafana.com/gpg.key

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

sudo apt-get update

sudo apt-get install grafana -y

sudo systemctl daemon-reload

sudo systemctl enable grafana-server.service
sudo systemctl start grafana-server

sudo systemctl status grafana-server
```

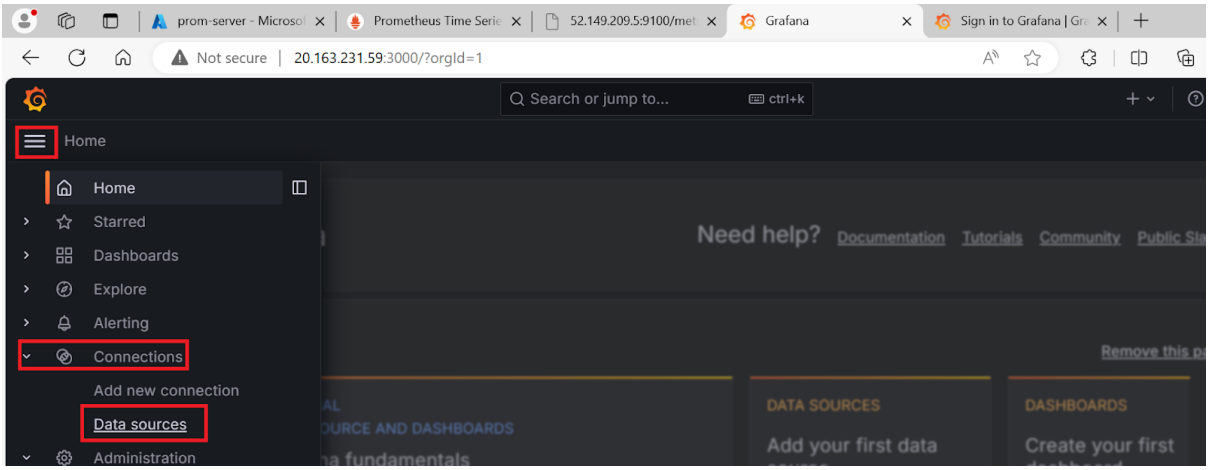
Integrate Grafana and Prometheus

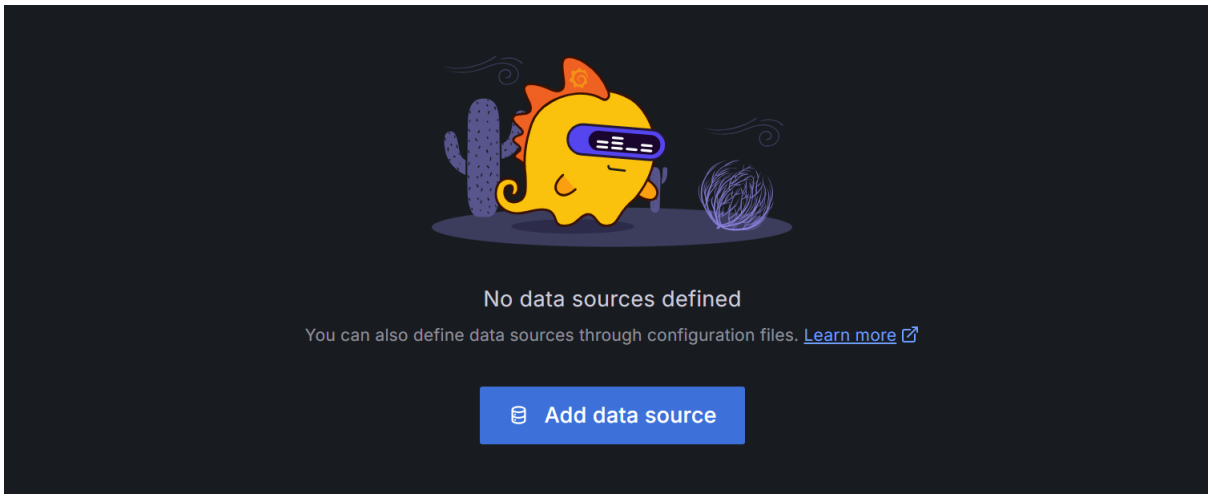
Access Grafana http://prometheus_ip:3000

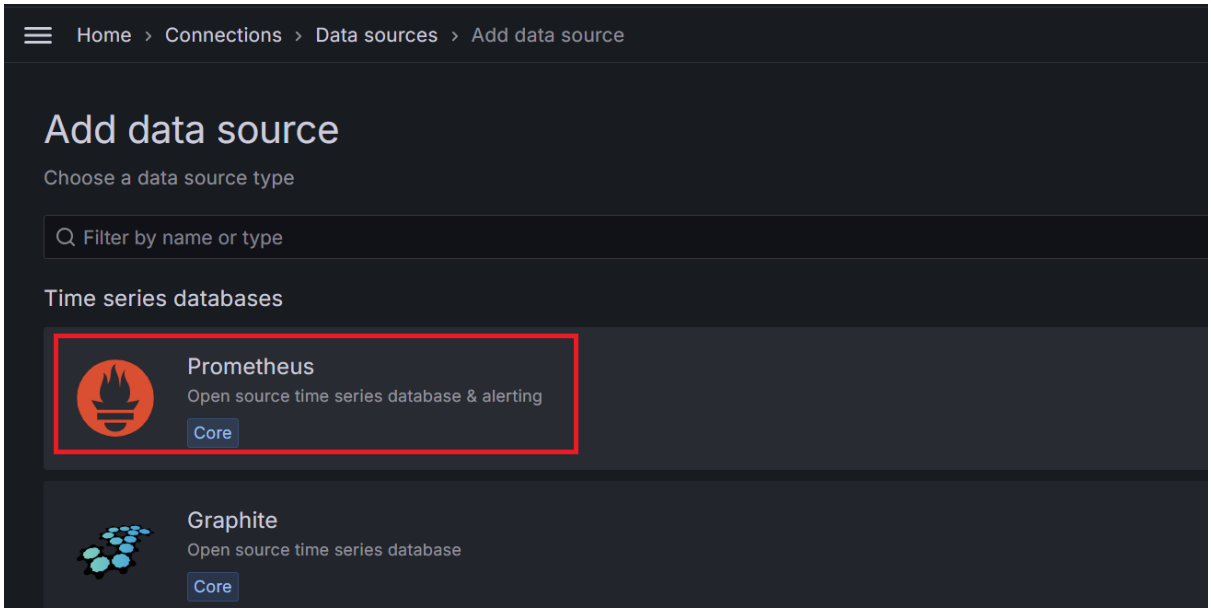
Use the user name **admin** and the default password **admin**. Change the password to a more secure value when prompted to do so.



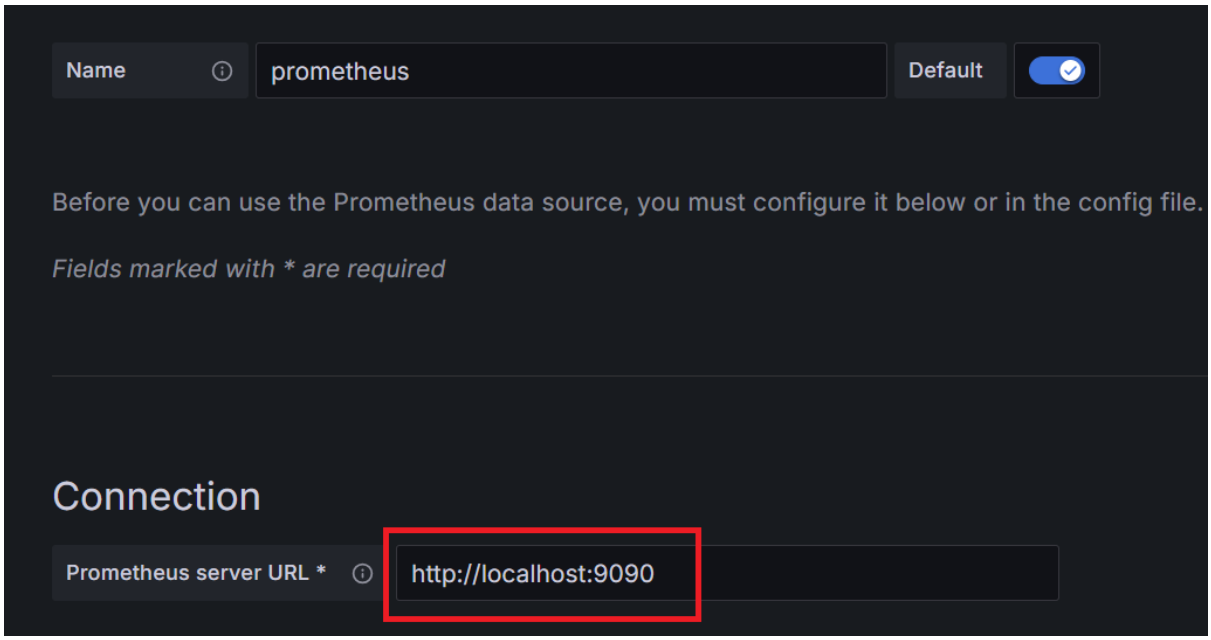
To add **Prometheus as a data source**

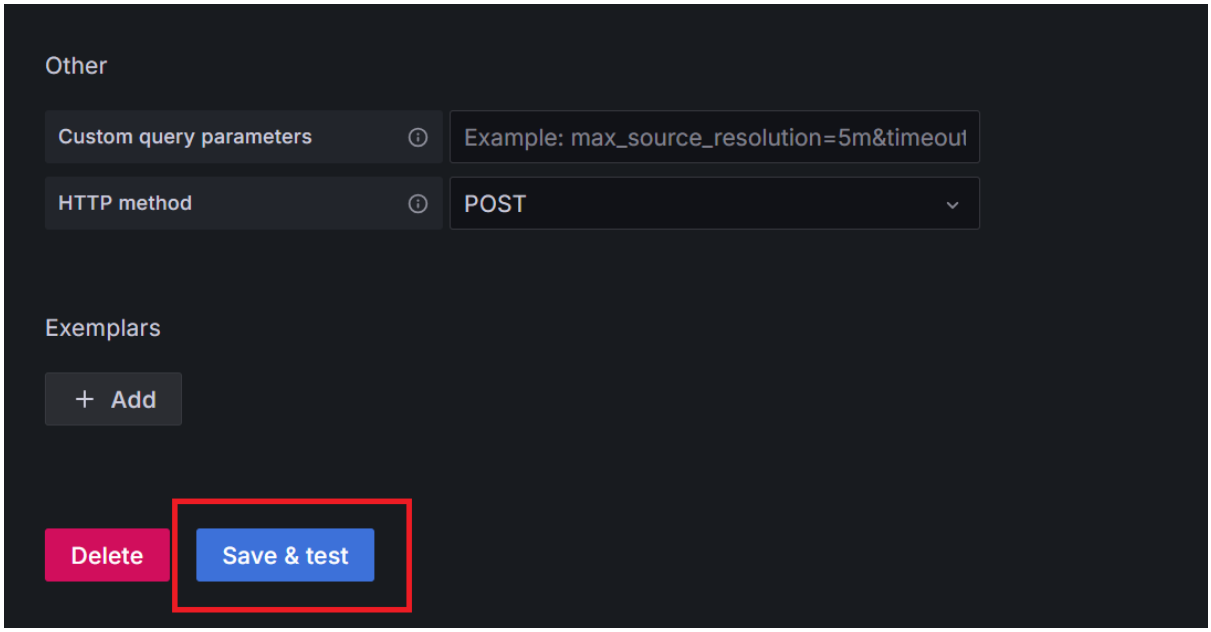


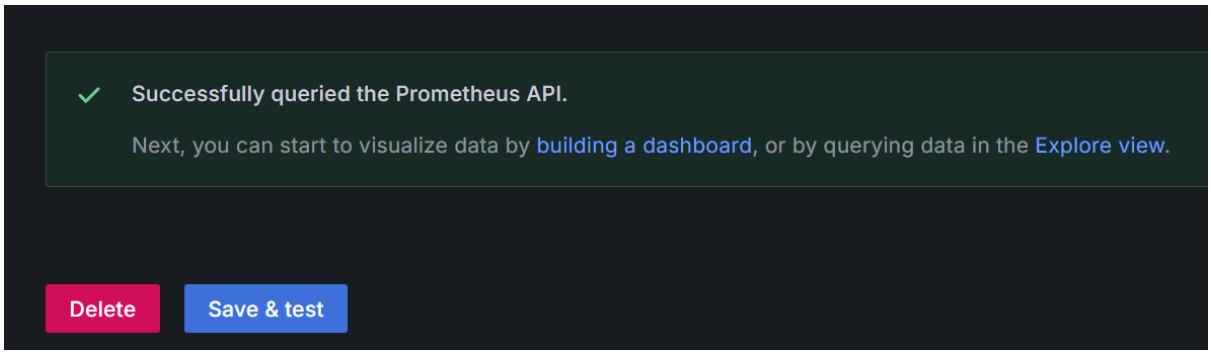




For a local Prometheus source, set the URL to http://localhost:9090.







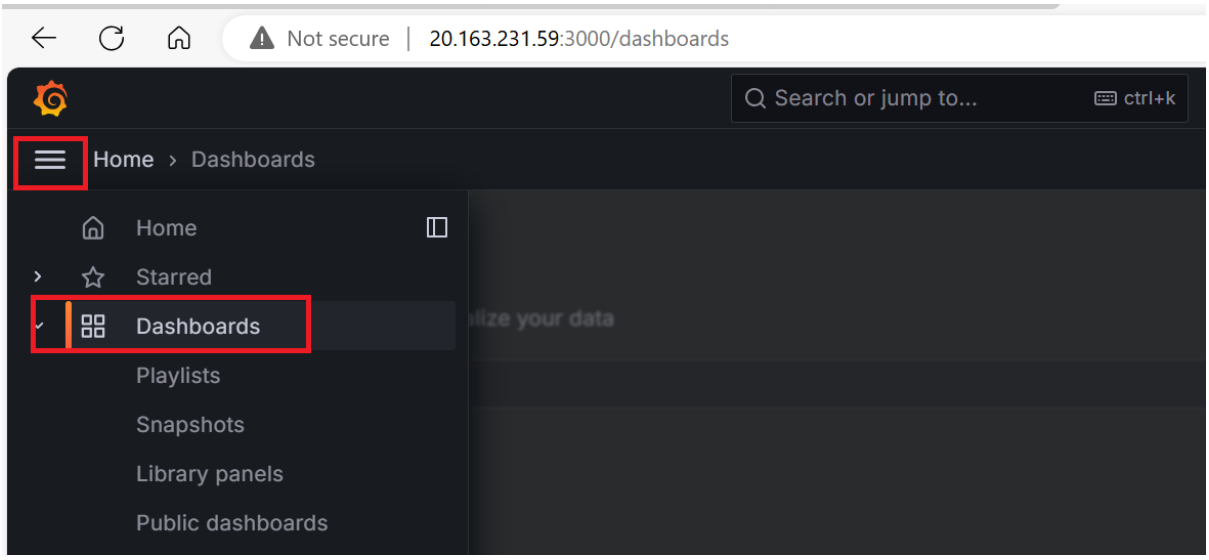
Import a Grafana Dashboard

A dashboard displays statistics for the client node using a more effective and standardized layout.

It is certainly possible to create a custom dashboard. However, **Prometheus has already created** a dashboard to support the **Node Exporter statistics**.

Import the Node Exporter dashboard:

Visit the **Grafana Dashboard Library**. Enter **Node exporter** as the search term.

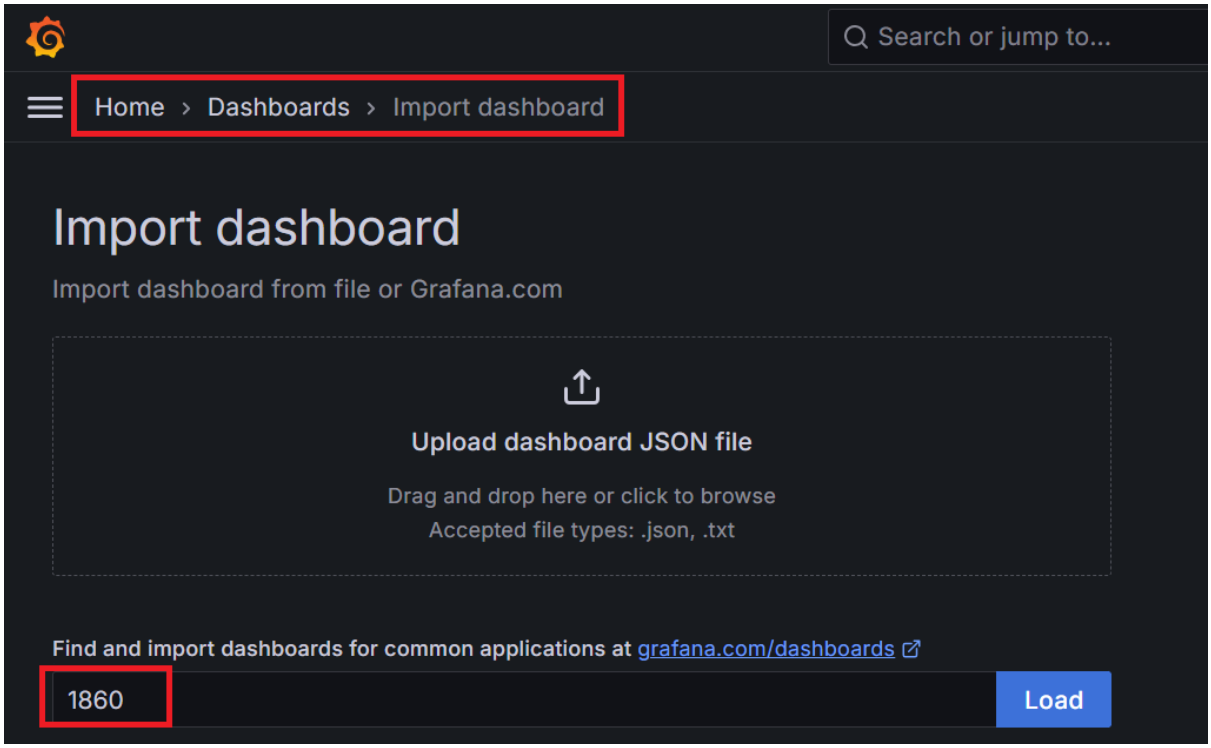


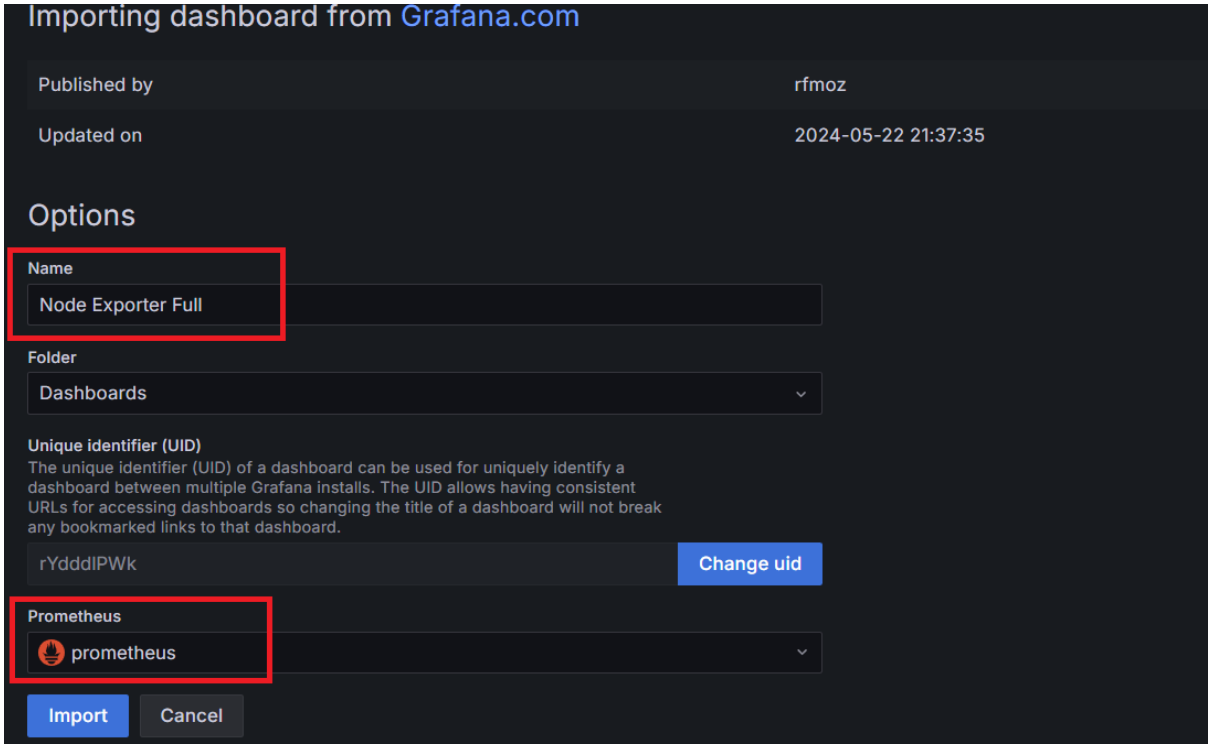
If you did not find, follow these steps:

Grafana has a template of this dashboard which you can find at:
https://grafana.com/grafana/dashboards/1860-node-exporter-full/

ID is 1860

In your Grafana click **Import Dashboard**





Select Prometheus and click **Import**.

The **Node Exporter Full dashboard** takes effect immediately.

It displays the performance metrics and state of the client node, including the Memory, RAM, and CPU details. Several drop-down menus at the top of the screen allow users to select the host to observe and the time period to highlight.