

# **Monitoring EC2 Instance using Prometheus and Grafana**

## **How Grafana and Prometheus work together**

Grafana and Prometheus are probably the most prominent tools in the application monitoring and analytics space. Prometheus is an open-source monitoring and alerting platform that collects and stores metrics as time-series data. Grafana is an open-source analytics and interactive visualization web application. It allows you to ingest data from a huge number of data sources, query this data, and display it on beautiful customizable charts for easy analysis. It is also possible to, set alerts so you can quickly and easily be notified of abnormal behavior and lots more.



## **Agenda:**

1. Install Prometheus and configure Prometheus to monitor itself
2. Install Node Exporter on the other EC2 Instances
3. Configure Prometheus for the EC2 Instance
4. Install Grafana

## **Requirement:**

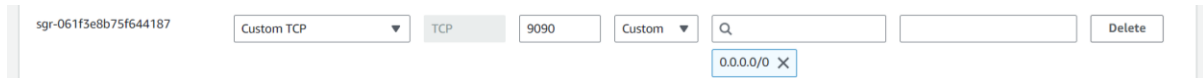
Launching 3 Amazon linux instances

**Pawar Divya**

## Solution :

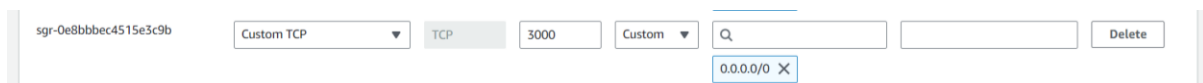
- **Step 1: Security groups configures on EC2 Instances**

**Prometheus security group port number 9090 enabled**



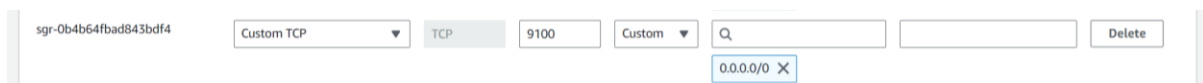
The screenshot shows the AWS Security Groups console for a security group with ID sgr-061f3e8b75f644187. The configuration is set to Custom TCP, with the port number 9090. The protocol is TCP, and the source is set to Custom. The destination is set to 0.0.0.0/0. A search bar and a Delete button are also visible.

**Grafana security group port number 3000 enabled**



The screenshot shows the AWS Security Groups console for a security group with ID sgr-0e8bbbec4515e3c9b. The configuration is set to Custom TCP, with the port number 3000. The protocol is TCP, and the source is set to Custom. The destination is set to 0.0.0.0/0. A search bar and a Delete button are also visible.

**Prometheus node exporter security group port number 9100 enabled**



The screenshot shows the AWS Security Groups console for a security group with ID sgr-0b4b64fbad843bdf4. The configuration is set to Custom TCP, with the port number 9100. The protocol is TCP, and the source is set to Custom. The destination is set to 0.0.0.0/0. A search bar and a Delete button are also visible.

## 1. Installing Prometheus

Launch one instance.

```
$ sudo yum install git
```

```
$ git clone https://github.com/Phanindra-Sangers/prometheus-monitoring.git
```

```
remote: Total 89 (delta 0), reused 0 (delta 0), pack-reused 89
Receiving objects: 100% (89/89), 19.03 KiB | 2.11 MiB/s, done.
Resolving deltas: 100% (36/36), done.
[ec2-user@ip-172-31-25-132 ~]$ ls
prometheus-monitoring  prometheus-monitoring.git  prometheus-monitoring.git.1
[ec2-user@ip-172-31-25-132 ~]$
```

```
$ cd prometheus-monitoring
```

```
$ ./install-prometheus.sh
```

```
[ec2-user@ip-172-31-25-132 ~]$ cd prometheus-monitoring/
[ec2-user@ip-172-31-25-132 prometheus-monitoring]$ ./install-prometheus.sh
--2024-05-21 05:37:39-- https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/9563ce00-2fee-11eb-8365-d1748ff14824?X-Amz-Credential=releaseassetproduction%2F20240521%2Fus-east-1%2Ffs3%2Faws4_request&X-Amz-Date=20240521T053739Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=6838921&response-content-disposition=prometheus-2.23.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
```

This script will install everything and configured it.

- Creates a new user and add new directories

```
sudo useradd --no-create-home prometheus
sudo mkdir /etc/prometheus
sudo mkdir /var/lib/prometheus
```

Downloads the Prometheus, extract it and put it in /usr/local/bin folder and finally delete the software

```
wget
https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz
```

```
sudo cp -r prometheus-2.23.0.linux-amd64/console_libraries /etc/prometheus
```

```
tar -xvf prometheus-2.23.0.linux-amd64.tar.gz
sudo cp prometheus-2.23.0.linux-amd64/prometheus /usr/local/bin
sudo cp prometheus-2.23.0.linux-amd64/promtool /usr/local/bin
sudo cp prometheus-2.23.0.linux-amd64/promtool /usr/local/bin/
```

```
rm -rf prometheus-2.23.0.linux-amd64.tar.gz prometheus-2.19.0.linux-amd64
```

## 2. Now we will configure Prometheus to monitor itself using yaml file.

Create a prometheus.yml file at /etc/prometheus/prometheus.yml with the below content.

```
$ cd /etc/prometheus
```

- copy the following content in yml file.

```
global:
  scrape_interval: 15s
  external_labels:
    monitor: 'prometheus'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
~
~
~
~
```

Now we want to run the Prometheus as a Service so that in case of server restart service will come automatically

- Create a file at

```
$ cd /etc/systemd/system/
$ vi prometheus.service
```

- copy the below content in the service file

```
[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=/var/lib/prometheus/ \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries

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

- Save and exit.

Change the ownership of all folders and files which we have created to the user which we have created in the first step

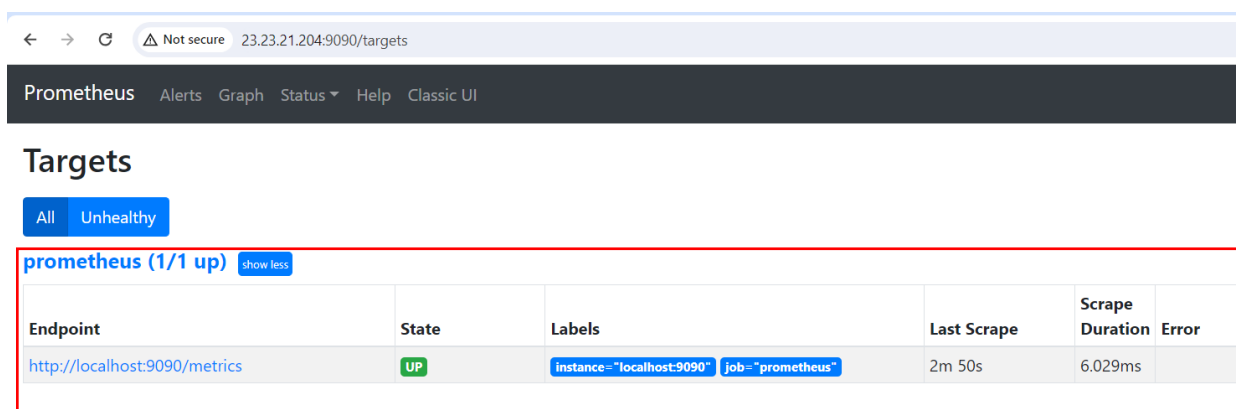
```
sudo chown prometheus:prometheus /etc/prometheus
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
sudo chown -R prometheus:prometheus /var/lib/prometheus
```

Now we will configure the service and start it

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

now you can access the prometheus dashboard

<http://<prometheus-public-ip>:9090>



The screenshot shows the Prometheus web interface. The 'Targets' section is active, displaying a table with one target. The target is 'prometheus (1/1 up)' and is in a 'UP' state. The table includes columns for Endpoint, State, Labels, Last Scrape, Scrape Duration, and Error.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	instance="localhost:9090" job="prometheus"	2m 50s	6.029ms	

## ➤ Step 2 : Install Node Exporter

Launch one more instance i.e Node exporter

```
$ sudo yum install -y git
$ git clone https://github.com/Phanindra-Sangers/prometheus-monitoring.git
```

# 1. Now to monitor your servers you need to install the node exporter on all your target machine which is like a monitoring agent on all the servers.

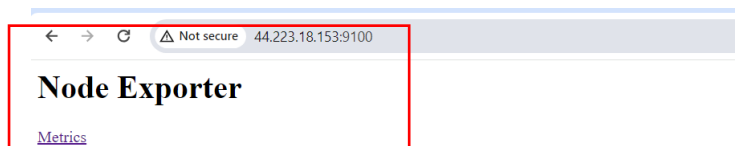
```
cd prometheus-monitoring/  
./install-node-exporter.sh
```

This script will do the below steps:

```
sudo useradd --no-create-home node_exporter  
wget  
https://github.com/prometheus/node_exporter/releases/download/v1.0.1/node_exporter-1.0.1.linux-amd64.tar.gz  
tar xzf node_exporter-1.0.1.linux-amd64.tar.gz  
sudo cp node_exporter-1.0.1.linux-amd64/node_exporter /usr/local/bin/node_exporter  
rm -rf node_exporter-1.0.1.linux-amd64.tar.gz node_exporter-1.0.1.linux-amd64  
sudo cp node-exporter.service /etc/systemd/system/node-exporter.service  
sudo systemctl daemon-reload  
sudo systemctl enable node-exporter  
sudo systemctl start node-exporter  
sudo systemctl status node-exporter
```

Access the Node Exporter using

**http:<node-exporter-public-ip>:9100**



### ➤ Step 3: Configure Prometheus for the Nodes

Now we will configure the Prometheus for our EC2 instance where we have installed the node-exporter.

Login to the Prometheus server and edit the file file at location in Prometheus instance

```
/etc/prometheus/prometheus.yml
```

```
global:
  scrape_interval: 15s
  external_labels:
    monitor: 'prometheus'

scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['34.229.126.62:9100']
```

Restart the Prometheus service

```
sudo systemctl restart prometheus
sudo systemctl status prometheus
```

Now you can open the Prometheus using below url and can see the new targets added

```
http://<prometheus-public-ip>:9090/targets
```

The screenshot shows the Prometheus web interface at the URL 54.92.218.94:9090/targets. The page title is 'Prometheus' with navigation links for Alerts, Graph, Status, Help, and Classic UI. Under the 'Targets' section, there are two tabs: 'All' (selected) and 'Unhealthy'. A table lists the targets, with one entry highlighted by a red border:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://34.229.126.62:9100/metrics	UP	instance="34.229.126.62:9100" job="node_exporter"	2m 49s	17.562ms	

## ➤ Step 4: Install Grafana

Once Prometheus is installed successfully then we can install the Grafana and configure Prometheus as a datasource.

Grafana is an opensource tool which is used to provide the visualization of your metrics.

1. Steps to Install:
2. Run the below file for apt-get pkg manager

```
[ec2-user@ip-172-31-27-226 ~]$ wget https://dl.grafana.com/enterprise/release/grafana-enterprise-11.0.0.linux-amd64.tar.gz
--2024-05-21 12:08:55-- https://dl.grafana.com/enterprise/release/grafana-enterprise-11.0.0.linux-amd64.tar.gz
Resolving dl.grafana.com (dl.grafana.com)... 146.75.30.217, 2a04:4e42:79::729
Connecting to dl.grafana.com (dl.grafana.com)|146.75.30.217|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127811601 (122M) [application/x-tar]
Saving to: 'grafana-enterprise-11.0.0.linux-amd64.tar.gz.1'

grafana-enterprise-11.0.0.linux-amd64.ta 100%[=====>] 121.89M 128MB/s in 0.9s

2024-05-21 12:08:56 (128 MB/s) - 'grafana-enterprise-11.0.0.linux-amd64.tar.gz.1' saved [127811601/127811601]

[ec2-user@ip-172-31-27-226 ~]$ ls
grafana-enterprise-11.0.0.linux-amd64.tar.gz  grafana-enterprise-11.0.0.linux-amd64.tar.gz.1  grafana-v11.0.0
[ec2-user@ip-172-31-27-226 ~]$
```

3. Untar the downloaded file.

```
[ec2-user@ip-172-31-27-226 ~]$ tar -zxvf grafana-enterprise-11.0.0.linux-amd64.tar.gz
```



## 4. Start the service and check the status

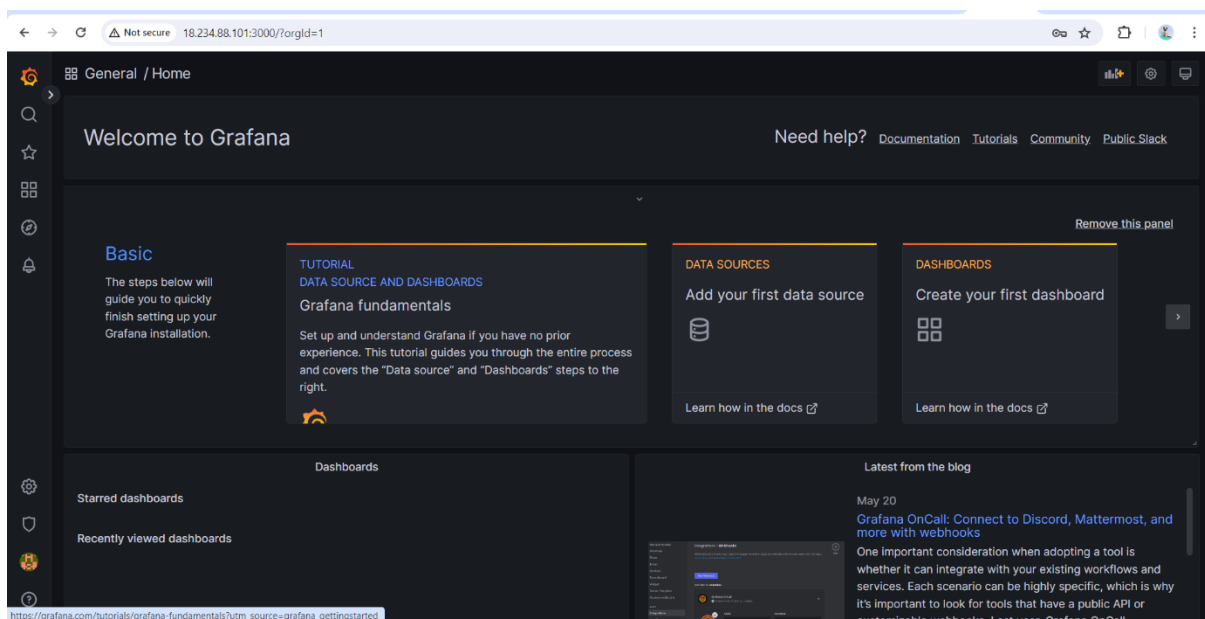
```
[ec2-user@ip-172-31-27-226 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-27-226 ~]$ sudo systemctl start grafana-server
[ec2-user@ip-172-31-27-226 ~]$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-s
Executing: /usr/lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /usr/lib/systemd/
[ec2-user@ip-172-31-27-226 ~]$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-05-21 12:46:51 UTC; 19s ago
     Docs: http://docs.grafana.org
   Main PID: 27723 (grafana)
    Tasks: 12 (limit: 1114)
   Memory: 149.3M
```

## 5. Now open it on the browser using below url:

**\*Make sure that port 3000 is open for this instance.**

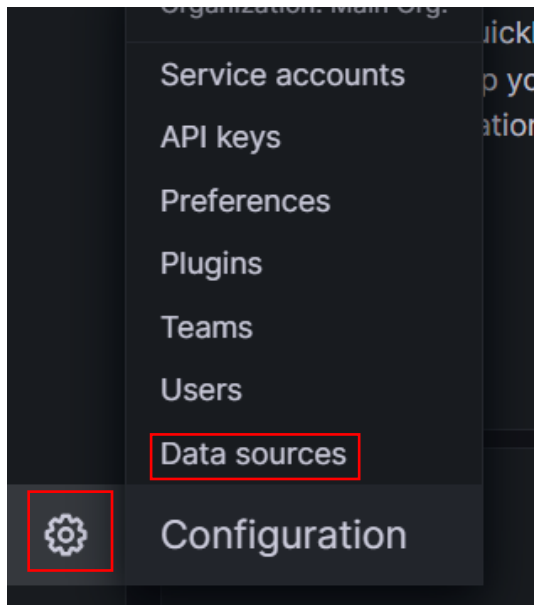
`http://<grafana-public-ip>:3000`

Login with username: `admin` and password `admin`

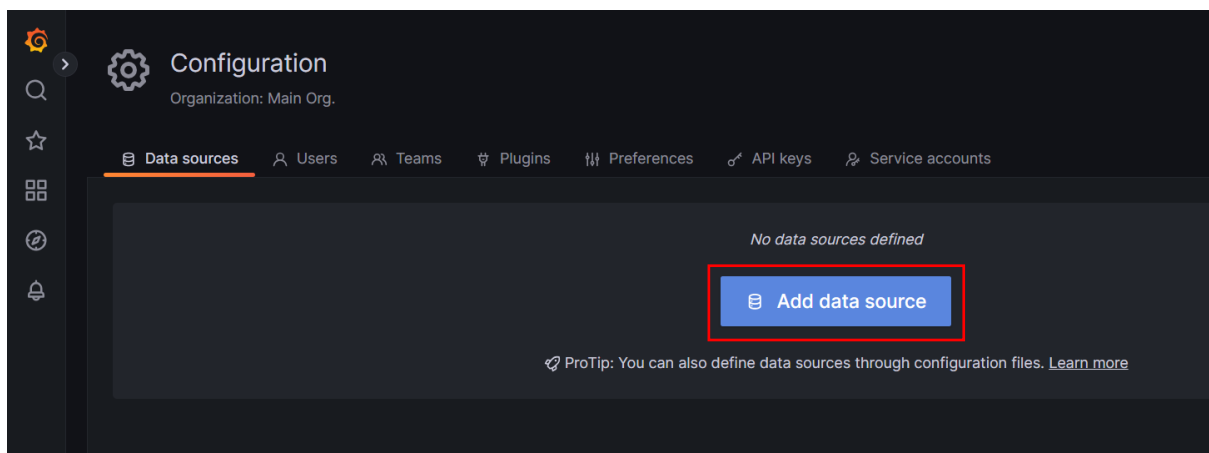


## Add Prometheus DataSource

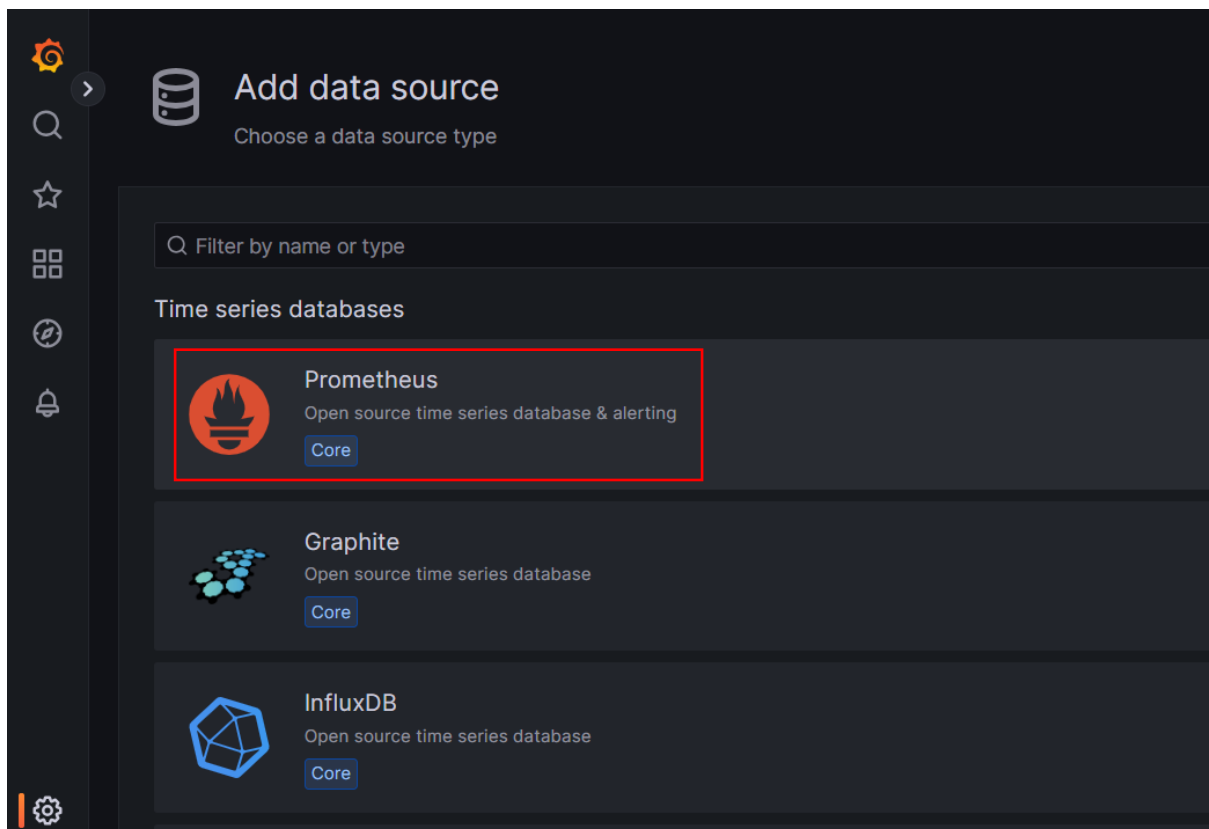
Click on **setting** > **Data sources**



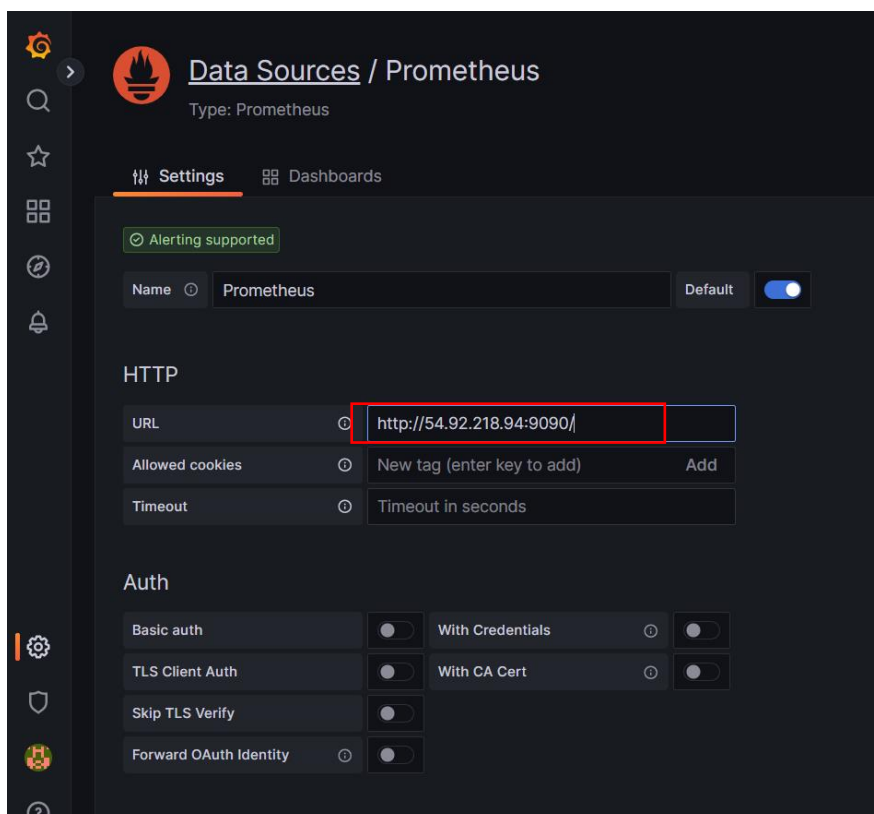
Click on **Add data source**



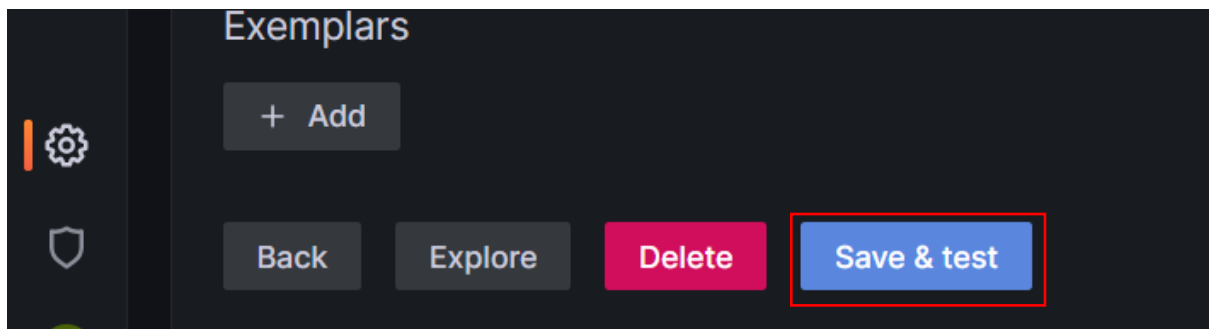
## Click on Prometheus



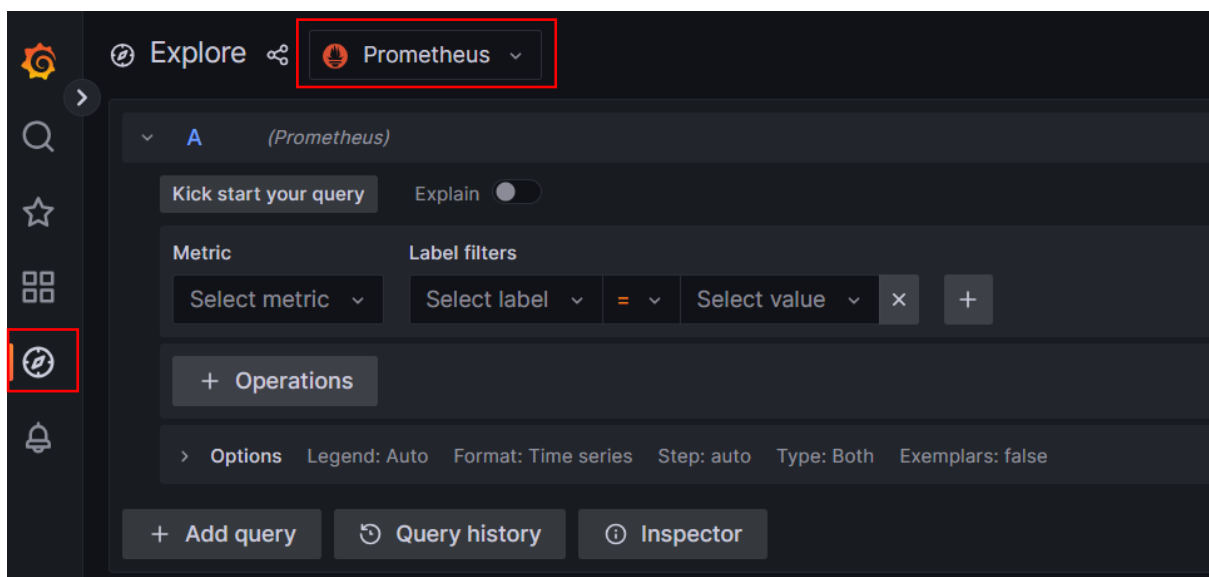
## Mention your Prometheus IP



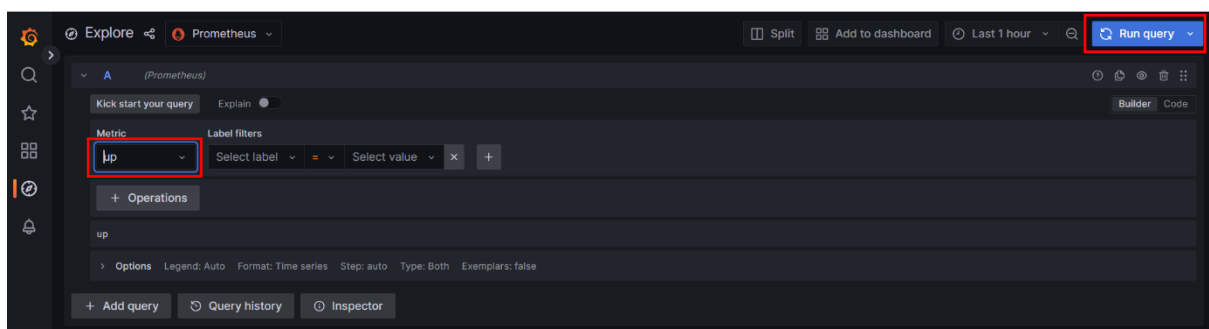
Scroll down click **save and test**.



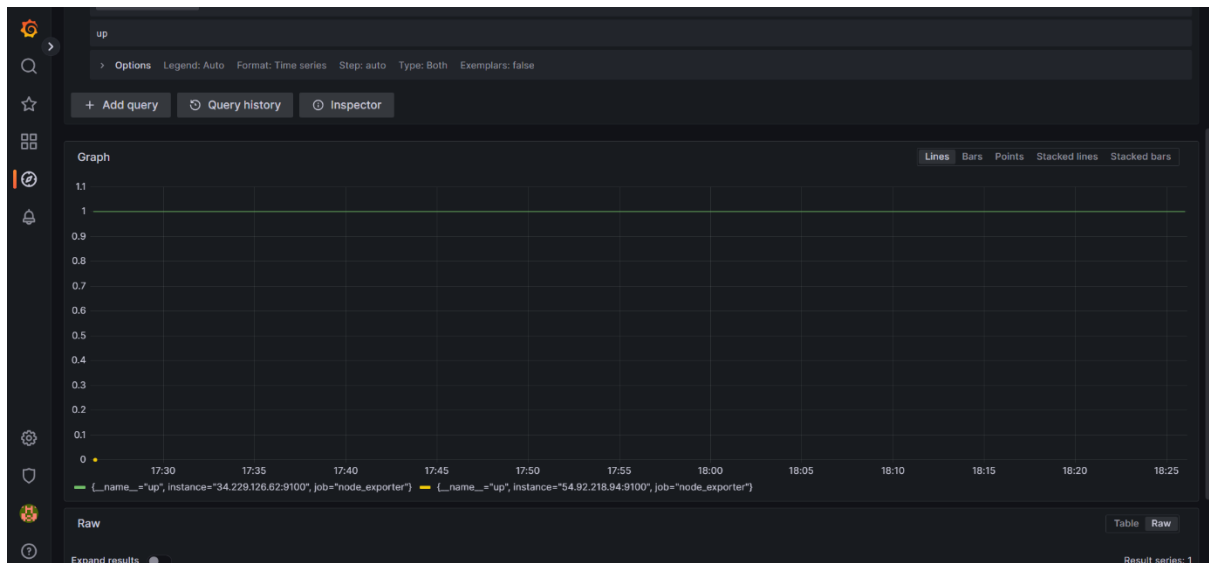
Click on **Explore** > select **Prometheus** as a **datasource**



Now click on **metrics** > Select **Up** > **Run query**



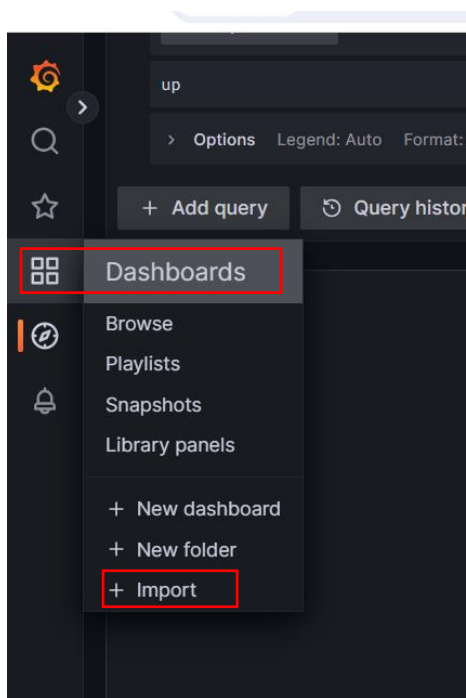
You will get dashboards



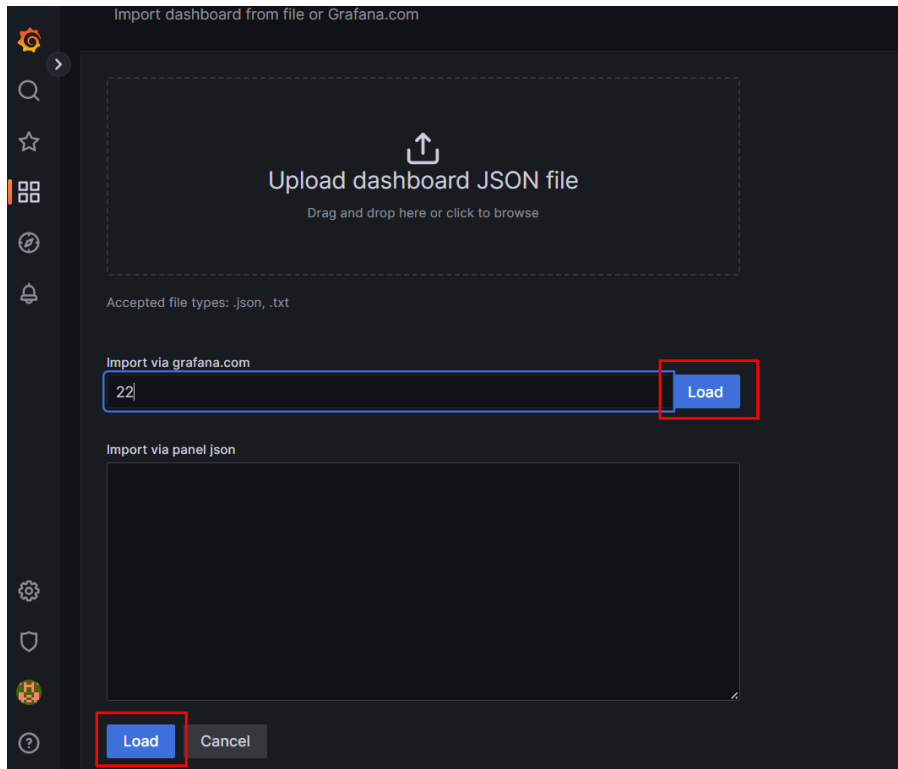
Grafana provides lot of dashboards which we can directly import in our Grafana instance and use it.

Import the dashboard

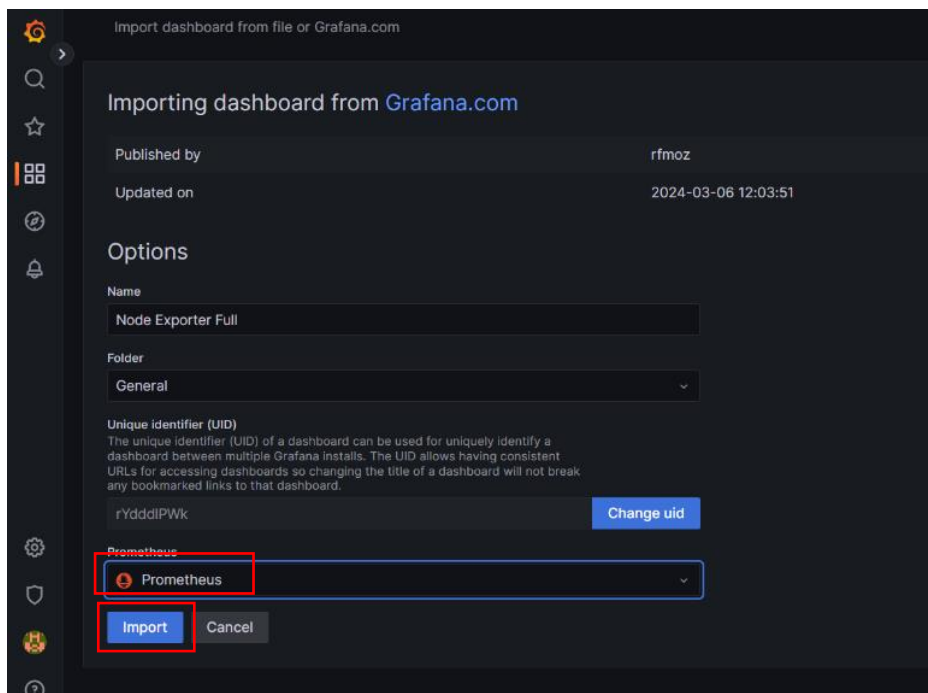
Click on **Dashboards** > click on **Import**

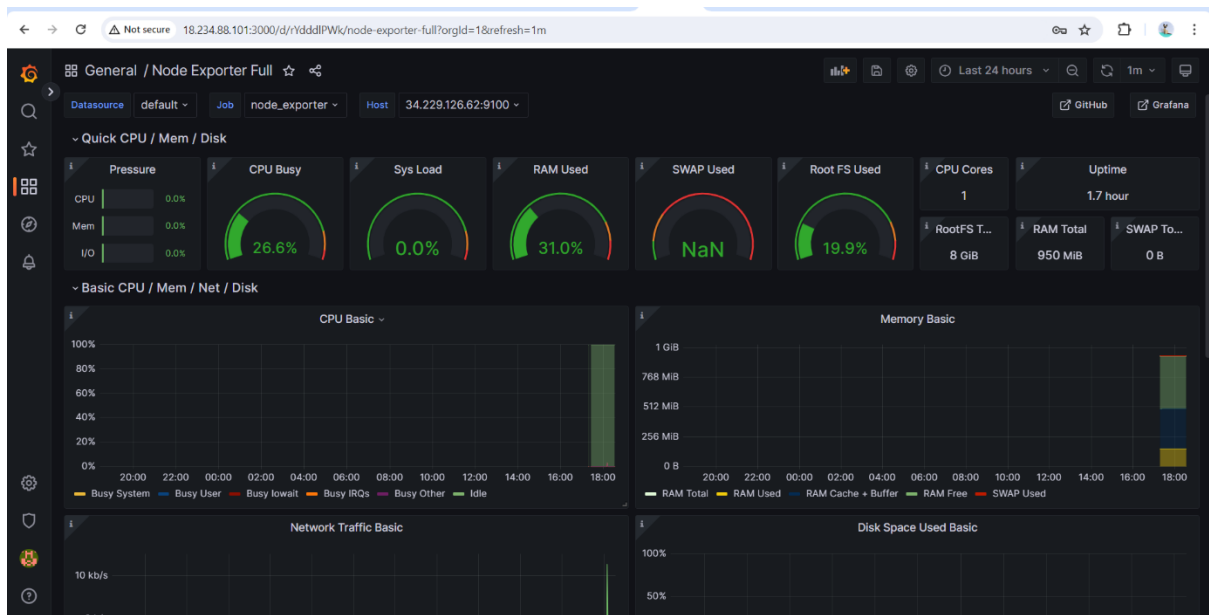


## Click on **Load**



Select the node to configure [**Prometheus**] > select **Import**





**Conclusion :**  
**We have come to know how to install Prometheus and grafana to monitor Ec2 instances**