Setup of Prometheus, Node Exporter, and Grafana on AWS EC2 Ubuntu

What is Prometheus?

Prometheus monitoring solution is a free and open-source solution for monitoring metrics, events, and alerts. It collects and records metrics from servers, containers, and applications.

In addition to providing a flexible query language (PromQL), and powerful visualization tools, it also provides an alerting mechanism that sends notifications when needed.

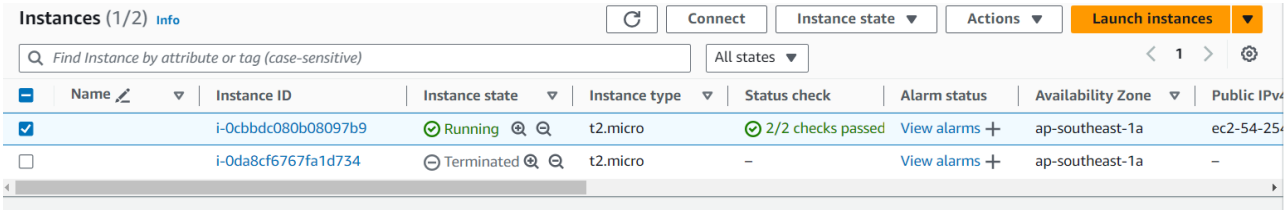
Prerequisites:

Before diving into the setup process, make sure you have the following prerequisites in place:

1. AWS EC2 Instance: Create an EC2 instance running Ubuntu.
2. SSH Access: Ensure you have SSH access to your EC2 instance.
3. Basic Linux Knowledge: Familiarity with basic Linux commands will be beneficial.

Step 1 : Launch an instance: Connect to Public IP to server.

In security Group – Edit Inbound Rule with TCP -9090 TCP – 9100 TCP – 3000



Step 2: Create a Dedicated Prometheus User

To enhance security and streamline administration, it's best to create a dedicated user for Prometheus.

This user will serve as a system account for the service. Run the following command:

sudo useradd \

- system \

- no-create-home \

- shell /bin/false Prometheus

- sudo mkdir /etc/prometheus

- sudo mkdir /var/lib/Prometheus

Step 3 : Download and Extract Prometheus

Let's fetch the latest version of Prometheus from the official download page using wget:

- wget https://github.com/prometheus/prometheus/releases/download/v2.45.0/prometheus-2.45.0.linux-amd64.tar.gz

Now, extract the Prometheus files:

- tar -xvf prometheus-2.45.0.linux-amd64.tar.gz

```
ubuntu@ip-172-31-28-28:~$ sudo groupadd --system prometheus
ubuntu@ip-172-31-28-28:~$ sudo useradd -s /sbin/nologin --system -g prometheus p
rometheus
ubuntu@ip-172-31-28-28:~$ sudo mkdir /etc/prometheus
ubuntu@ip-172-31-28-28:~$ sudo mkdir /var/lib/prometheus
ubuntu@ip-172-31-28-28:~$ wget https://github.com/prometheus/prometheus/releases
/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz
--2024-06-15 10:06:26--  https://github.com/prometheus/prometheus/releases/downl
oad/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-
2e65be/6838921/cdffb95e-2db9-48c1-8ee8-69bd587a304a?X-Amz-Algorithm=AWS4-HMAC-SH
A256&X-Amz-Credential=releaseassetproduction%2F20240615%2Fus-east-1%2Fs3%2Faws4_
request&X-Amz-Date=20240615T100626Z&X-Amz-Expires=300&X-Amz-Signature=4a997d776b
47740663f6391a1a54b0fdf02b5c5d3b2f7110433593a1ebefa3bb&X-Amz-SignedHeaders=host&
actor_id=0&key_id=0&repo_id=6838921&response-content-disposition=attachment%3B%2
0filename%3Dprometheus-2.43.0.linux-amd64.tar.gz&response-content-type=applicati
on%2Foctet-stream [following]
--2024-06-15 10:06:26--  https://objects.githubusercontent.com/github-production
-release-asset-2e65be/6838921/cdffb95e-2db9-48c1-8ee8-69bd587a304a?X-Amz-Algorit
hm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240615%2Fus-east
-1%2Fs3%2Faws4_request&X-Amz-Date=20240615T100626Z&X-Amz-Expires=300&X-Amz-Signa
ture=4a997d776b47740663f6391a1a54b0fdf02b5c5d3b2f7110433593a1ebefa3bb&X-Amz-Sign
edHeaders=host&actor_id=0&key_id=0&repo_id=6838921&response-content-disposition=
attachment%3B%20filename%3Dprometheus-2.43.0.linux-amd64.tar.gz&response-content
-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.1
99.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.
199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 91091544 (87M) [application/octet-stream]
Saving to: 'prometheus-2.43.0.linux-amd64.tar.gz'

prometheus-2.43.0.l 100%[===================>]  86.87M   135MB/s    in 0.6s

2024-06-15 10:06:28 (135 MB/s) - 'prometheus-2.43.0.linux-amd64.tar.gz' saved [9
1091544/91091544]
```

```
ubuntu@ip-172-31-28-28:~$ tar vxf prometheus*.tar.gz
prometheus-2.43.0.linux-amd64/
prometheus-2.43.0.linux-amd64/LICENSE
prometheus-2.43.0.linux-amd64/consoles/
prometheus-2.43.0.linux-amd64/consoles/prometheus.html
prometheus-2.43.0.linux-amd64/consoles/node-disk.html
prometheus-2.43.0.linux-amd64/consoles/node-overview.html
prometheus-2.43.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.43.0.linux-amd64/consoles/index.html.example
prometheus-2.43.0.linux-amd64/consoles/node-cpu.html
prometheus-2.43.0.linux-amd64/consoles/node.html
prometheus-2.43.0.linux-amd64/prometheus
prometheus-2.43.0.linux-amd64/promtool
prometheus-2.43.0.linux-amd64/NOTICE
prometheus-2.43.0.linux-amd64/console_libraries/
prometheus-2.43.0.linux-amd64/console_libraries/prom.lib
prometheus-2.43.0.linux-amd64/console_libraries/menu.lib
prometheus-2.43.0.linux-amd64/prometheus.yml
```

tep 3: Create Directories for Prometheus

Create the necessary directories for Prometheus and its configuration files:

Navigate to the Prometheus directory:

- cd proSmetheus-2.32.1.linux-amd64

Move Prometheus and promtool binaries to /usr/local/bin/:

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

Optionally, move console libraries to the Prometheus configuration directory:

- sudo mv consoles/ console_libraries/ /etc/prometheus/

Lastly, move the example Prometheus configuration file:

- sudo mv prometheus.yml /etc/prometheus/prometheus.yml

Step 5: Set Correct Ownership

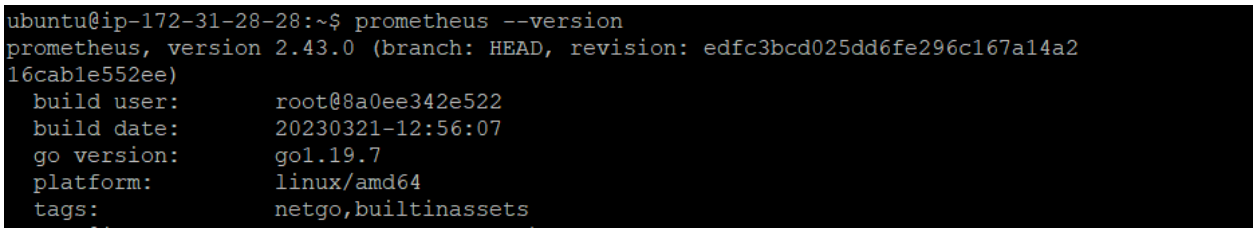
Ensure the correct ownership for /etc/prometheus/ and /data/ directories:

- sudo chown -R prometheus:prometheus /etc/prometheus/ /data/

Step 6: Verify Prometheus Installation

Check if Prometheus is installed correctly by running:

- prometheus –version



Step 7: Configure Prometheus as a systemd Service

To manage Prometheus using systemd, create a systemd unit configuration file:

- sudo vim /etc/systemd/system/prometheus.service

Add the following content to the file:

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

StartLimitIntervalSec=500

StartLimitBurst=5

[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=/data \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console_libraries \

--web.listen-address=0.0.0.0:9090 \

--web.enable-lifecycle

[Install]

WantedBy=multi-user.target[Install]

WantedBy=multi-user.target

Update the rule with command: sudo ufw allow 9090/tcp

```
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo mv consoles /etc/prometheus
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo mv console_libraries /etc/prometheus
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo mv prometheus.yml /etc/prometheus
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo chown prometheus:prometheus /etc/prometheus
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/consoles
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /var/lib/prometheus
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo vi  /etc/prometheus/prometheus.yml
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo vi /etc/systemd/system/prometheus.service
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo systemctl daemon-reload
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo systemctl start prometheus
```

Step 8: Enable and Start Prometheus

Enable Prometheus to start on boot:

- sudo systemctl enable Prometheus

Start Prometheus:

- sudo systemctl start Prometheus

Check the status of Prometheus:

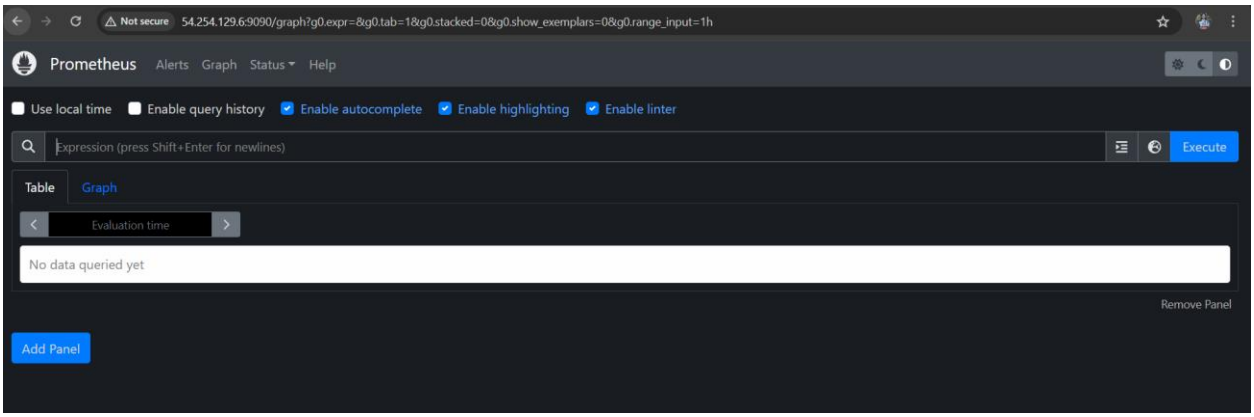
- sudo systemctl status Prometheus

```
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo systemctl status prometheus
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-06-15 10:11:52 UTC; 15s ago
   Main PID: 1632 (prometheus)
      Tasks: 6 (limit: 1130)
     Memory: 17.0M (peak: 17.1M)
        CPU: 73ms
     CGroup: /system.slice/prometheus.service
             └─1632 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.templates=/etc/prometheus/consoles --web.c
```

```
ubuntu@ip-172-31-28-28:~/prometheus-2.43.0.linux-amd64$ sudo ufw allow 9090/tcp
Rules updated
Rules updated (v6)
```

Step 9: Access Prometheus Web Interface

Access the Prometheus web interface by opening your browser and navigating to your EC2 instance's public IP address on port **9090** (e.g., **http://your-instance-ip:9090).**



Step 10: Install and Configure Node Exporter

Next, we will set up Node Exporter to collect system metrics from your EC2 instance. Node Exporter exposes these metrics in Prometheus format.

Step 11: Create a System User for Node Exporter

Create a system user for Node Exporter with the following command:

```
sudo useradd \
--system \
--no-create-home \
--shell /bin/false node_exporter
```

Step 12: Download and Install Node Exporter

Download Node Exporter:

wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz

Extract Node Exporter:

tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz

Ankita Lunawat

Move the Node Exporter binary to /usr/local/bin/:

- sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/

Step 13: Create a systemd Service for Node Exporter

Create a systemd unit configuration file for Node Exporter:

- sudo vim /etc/systemd/system/node_exporter.service

Add the following content to the file:

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

StartLimitIntervalSec=500

StartLimitBurst=5

[Service]

User=node_exporter

Group=node_exporter

Type=simple

Restart=on-failure

RestartSec=5s

ExecStart=/usr/local/bin/node_exporter \

--collector.logind

[Install]

WantedBy=multi-user.target

Step 14: Enable and Start Node Exporter

Enable Node Exporter to start on boot:

- sudo systemctl enable node_exporter

Start Node Exporter:

- sudo systemctl start node_exporter

```
ubuntu@ip-172-31-28-28:~$ sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/
ubuntu@ip-172-31-28-28:~$ sudo vim /etc/systemd/system/node_exporter.service
ubuntu@ip-172-31-28-28:~$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
ubuntu@ip-172-31-28-28:~$ sudo systemctl start node_exporter
```

Check the status of Node Exporter:

- sudo systemctl status node_exporter

```
ubuntu@ip-172-31-28-28:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
     Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-06-16 06:20:17 UTC; 14s ago
   Main PID: 1848 (node_exporter)
      Tasks: 3 (limit: 1130)
     Memory: 2.1M (peak: 2.3M)
        CPU: 9ms
     CGroup: /system.slice/node_exporter.service
             └─1848 /usr/local/bin/node_exporter --collector.logind
```

Step 15: Add Node Exporter as a Target in Prometheus

sudo vim /etc/prometheus/prometheus.yml

Add the following job configuration for Node Exporter:

- job_name: 'node_export'

  static_configs:

    - targets: ["localhost:9100"]

```
ubuntu@ip-172-31-28-28:~$ sudo vi /etc/prometheus/prometheus.yml
ubuntu@ip-172-31-28-28:~$ cat  vi /etc/prometheus/prometheus.yml
cat: vi: No such file or directory
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9100"]
```

Step 16: Reload Prometheus Configuration

Before reloading the configuration, validate it:

promtool check config /etc/prometheus/prometheus.yml

```
ubuntu@ip-172-31-28-28:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
 SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax
```

Step 17: Install Grafana for Visualization

Grafana is a powerful visualization tool that works seamlessly with Prometheus. Install Grafana by following these steps:

Ensure dependencies are installed:

- sudo apt-get install -y apt-transport-https software-properties-commo

```
Lifecycle API is not enablsudo apt-get install -y apt-transport-https software-properties-commonps software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 77 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (9329 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 71839 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Add the Grafana GPG key:

- wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

Add the Grafana repository for stable releases:

```
ubuntu@ip-172-31-28-28:~$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-28-28:~$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/oss/deb stable main
ubuntu@ip-172-31-28-28:~$
```

- echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

Update and install Grafana:

- sudo apt-get update

```
ubuntu@ip-172-31-28-28:~$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/oss/deb stable main
ubuntu@ip-172-31-28-28:~$ sudo apt-get update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://packages.grafana.com/oss/deb stable InRelease [7661 B]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://packages.grafana.com/oss/deb stable/main amd64 Packages [250 kB]
Fetched 383 kB in 1s (421 kB/s)
Reading package lists... Done
W: https://packages.grafana.com/oss/deb/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPR
```

sudo apt-get -y install Grafana

```
ubuntu@ip-172-31-28-28:~$ sudo apt-get -y install grafana
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  musl
The following NEW packages will be installed:
  grafana musl
0 upgraded, 2 newly installed, 0 to remove and 77 not upgraded.
Need to get 115 MB of archives.
After this operation, 428 MB of additional disk space will be used.
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 musl amd64 1.2.4-2 [416 kB]
Get:2 https://packages.grafana.com/oss/deb stable/main amd64 grafana amd64 11.0.0 [115 MB]
Fetched 115 MB in 10s (11.8 MB/s)
Selecting previously unselected package musl:amd64.
(Reading database ... 71843 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_11.0.0_amd64.deb ...
Unpacking grafana (11.0.0) ...
Setting up musl:amd64 (1.2.4-2) ...
Setting up grafana (11.0.0) ...
info: Selecting UID from range 100 to 999 ...

info: Adding system user `grafana' (UID 111) ...
info: Adding new user `grafana' (UID 111) 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) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Step 18: Enable and Start Grafana

Enable Grafana to start on boot:

- sudo systemctl enable grafana-server

Start Grafana:

- sudo systemctl start grafana-server

Check the status of Grafana:

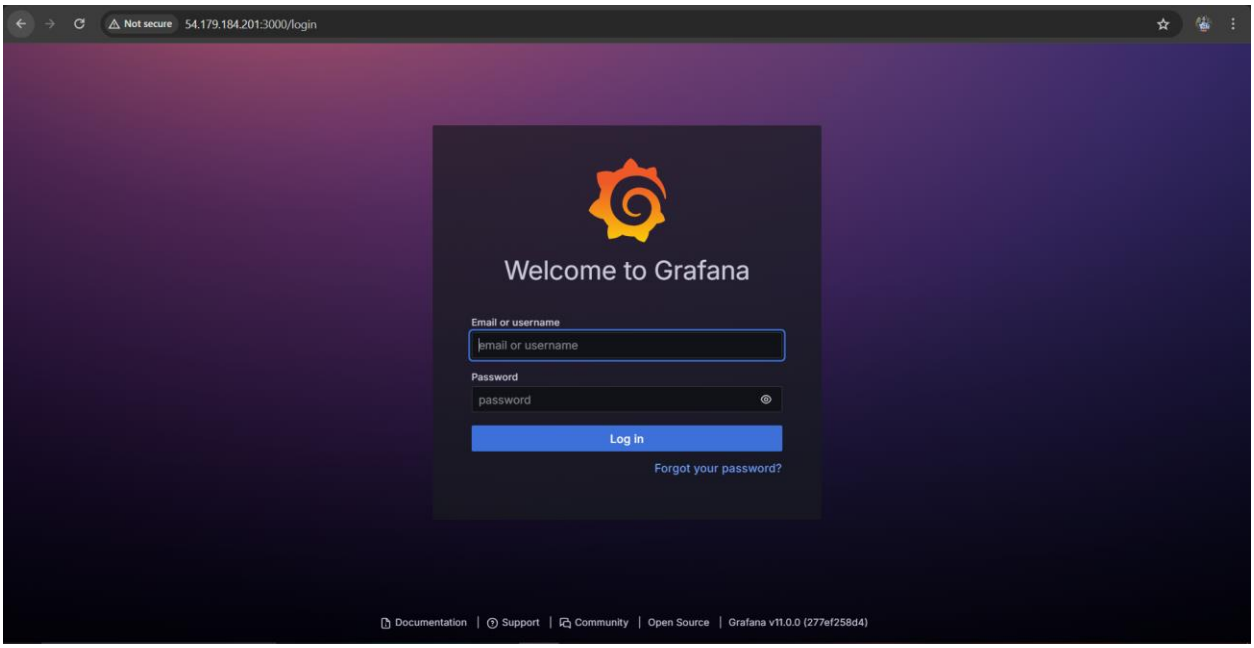
- sudo systemctl status grafana-server

```
ubuntu@ip-172-31-28-28:~$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
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/system/grafana-server.service.
ubuntu@ip-172-31-28-28:~$ sudo systemctl start grafana-server
```

```
ubuntu@ip-172-31-28-28:~$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
     Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-06-16 06:47:04 UTC; 16s ago
       Docs: http://docs.grafana.org
   Main PID: 2983 (grafana)
      Tasks: 10 (limit: 1130)
     Memory: 113.8M (peak: 114.1M)
        CPU: 2.258s
     CGroup: /system.slice/grafana-server.service
             └─2983 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/gra
```

Step 19: Access Grafana Web Interface

Access the Grafana web interface by opening your browser and navigating to your EC2 instance's public IP address on port 3000 (e.g., http://your-instance-ip:3000). Log in using the default credentials (username: admin, password: admin).



Step 20: Customize Grafana Settings

After your initial login to Grafana, you will have the opportunity to change your password. Let's set the new password to "**********"

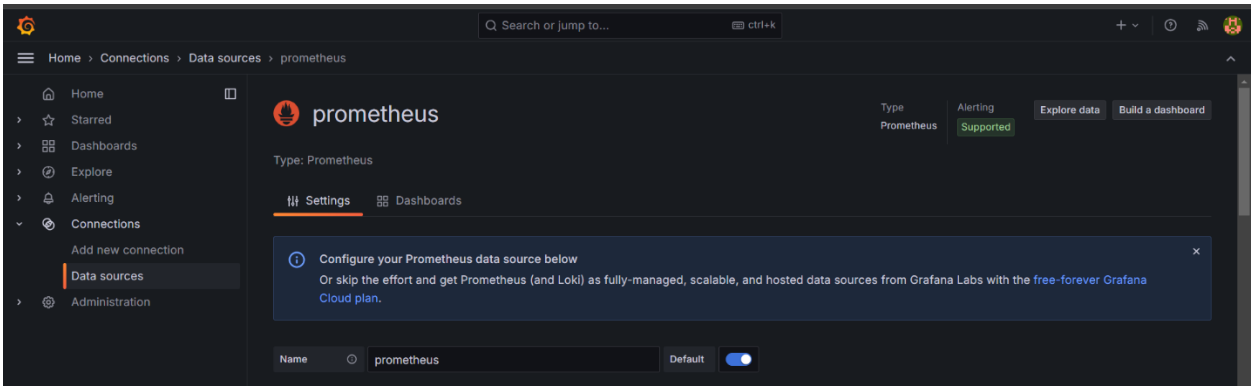
Step 21: Add Prometheus Data Source.

To visualize metrics, you need to add a data source in Grafana. Follow these steps:

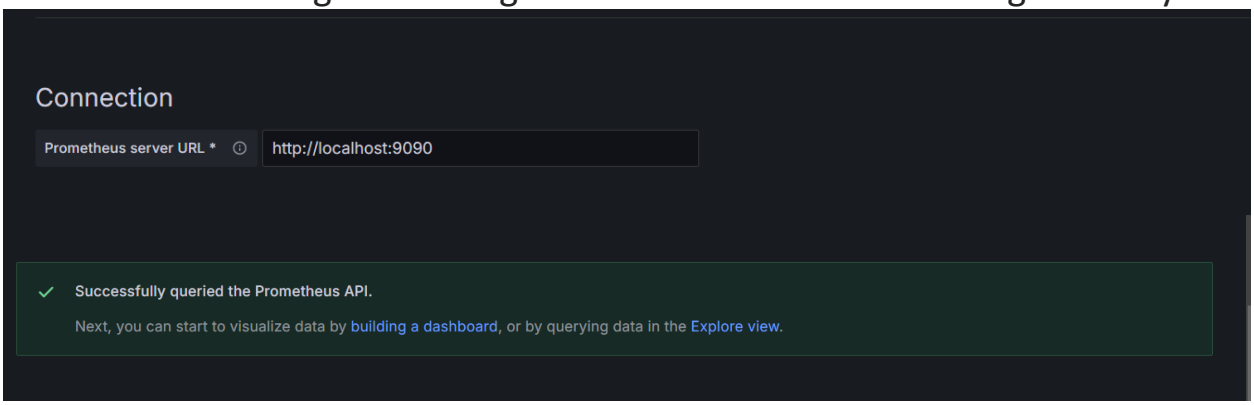
Click on the gear icon (⚙) in the left sidebar to access the Configuration menu.

Select "Data Sources" and click "Add data source."

Choose “Prometheus” as the data source type.



In the URL field, enter http://localhost:9090 and click "Save & Test." You should see a message indicating that the data source is working correctly.



Step 23: Create a Datasources Configuration File

Create a new datasources.yaml file for provisioning the data source as code:

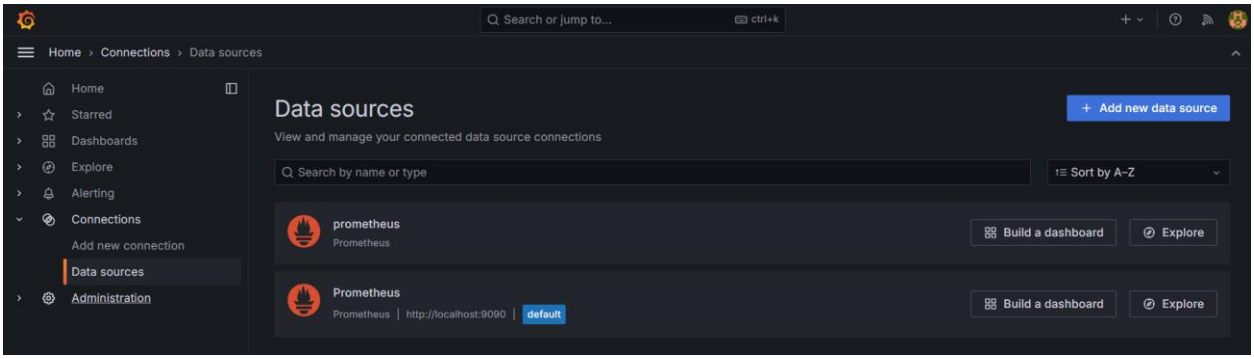
- sudo vim /etc/grafana/provisioning/datasources/datasources.yaml

Add the following content to the datasources.yaml file:

```
apiVersion: 1
datasources:
  - name: Prometheus
    type: prometheus
    url: http://localhost:9090
    isDefault: true
```

Optionally, you can set this data source as the default one.

```
ubuntu@ip-172-31-28-28:~$ sudo vim /etc/grafana/provisioning/datasources/datasources.yaml
ubuntu@ip-172-31-28-28:~$ sudo systemctl restart grafana-server
ubuntu@ip-172-31-28-28:~$ cat /etc/grafana/provisioning/datasources/datasources.yaml
apiVersion: 1
datasources:
  - name: Prometheus
    type: prometheus
    url: http://localhost:9090
    isDefault: true
```



Step 25: Create a Simple Graph in Grafana

Let's create a simple graph to visualize Prometheus metrics:

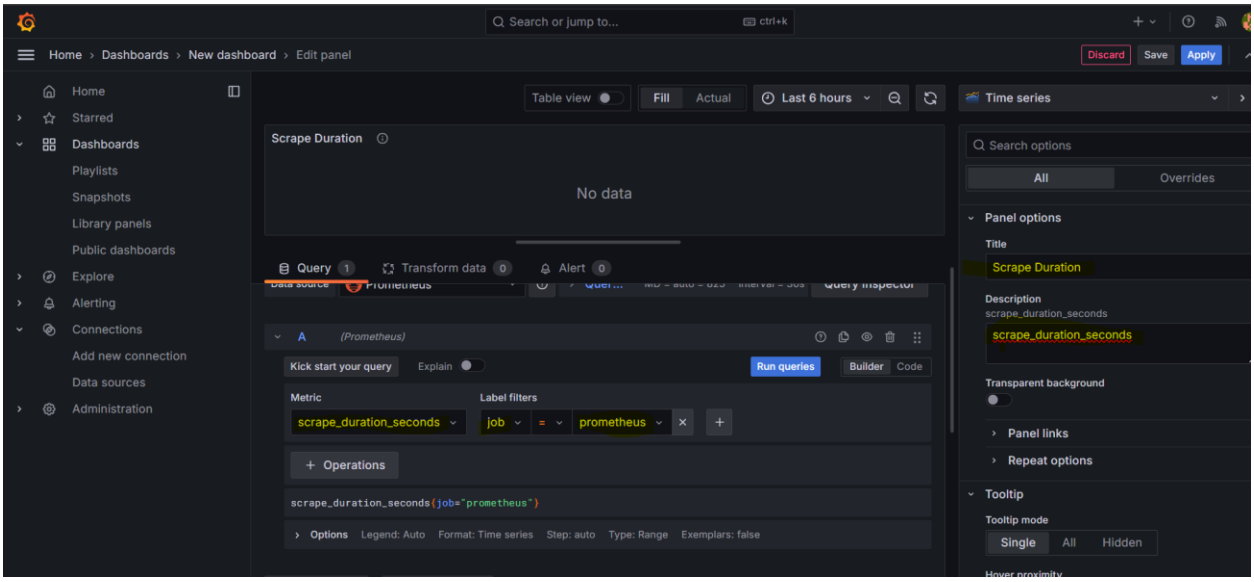
Go to Grafana and click "Create Dashboard," then add a new panel.

Give the panel a title, such as "Scrape Duration," and paste the scrape_duration_seconds metric.

You can adjust the time interval to 1 hour for a more detailed view.

Configure the legend to use the "job" label and set the unit to "seconds."

Click "Apply" and save the dashboard as "Prometheus."



Step 26: Import Node Exporter Dashboard

Since you already have Node Exporter running, you can import an open-source dashboard to visualize CPU, Memory, Network, and other metrics.

Follow these steps:

Visit the Grafana website's dashboard section at https://grafana.com/grafana/dashboards/.

Search for "node exporter" to find available dashboards.

Copy the dashboard ID (e.g., **1860**) to your clipboard.

- In Grafana:

  Click “Create” in the left sidebar to create a new dashboard.

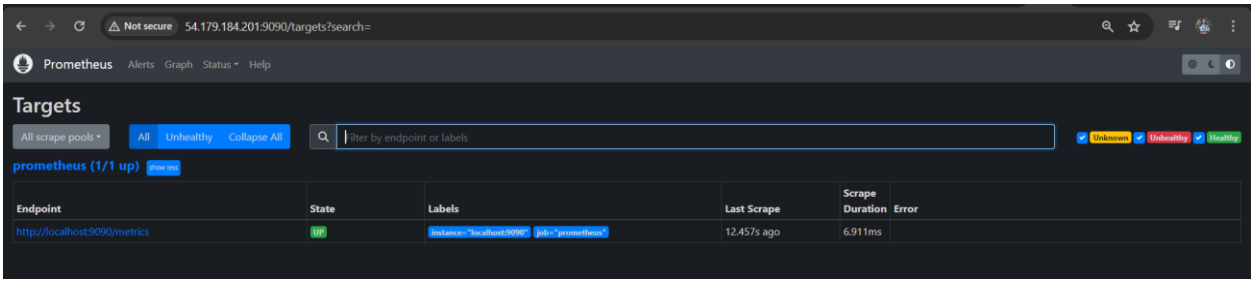
  Click “Import” and paste the dashboard ID you copied earlier.

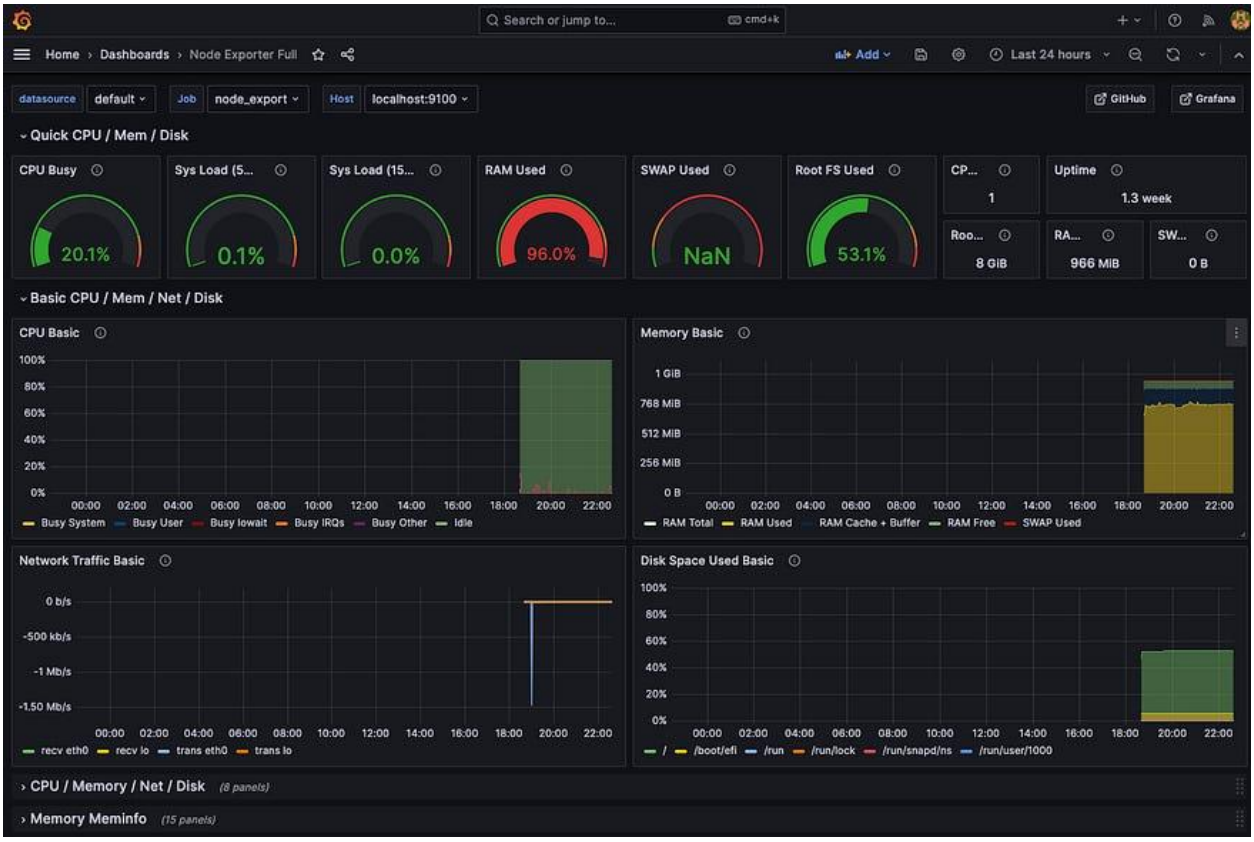
  Load the dashboard and select the Prometheus data source.

  Click “Import.”

Step 27: Explore Node Exporter Metrics

You now have access to a wide range of metrics from Node Exporter on your Grafana dashboard. These metrics include CPU, Memory, Network, and more. You can explore and customize the dashboard further to suit your monitoring needs.





You've successfully set up Prometheus, Node Exporter, and Grafana on your AWS EC2 Ubuntu instance.

This monitoring stack in place, you can visualize and analyze real-time metrics, ensuring the smooth operation of your AWS resources.
Feel free to explore Grafana's visualization options and customize dashboards to suit your specific monitoring needs.

Ankita Lunawat