Gautham Srinivasan

IT Analyst - DTCC

18 Nov 2024

Setting Up Prometheus, Node Exporters, and Grafana on AWS EC2 Instances

Project Overview

This guide provides step-by-step instructions to set up and connect Prometheus, Node Exporters, and Grafana on AWS EC2 instances for monitoring system metrics and visualizations.

Prerequisites

- AWS account with permissions to manage EC2 instances and security groups.
- Configured AWS key pair for SSH access.
- Security group allowing **All TCP traffic** with an open **CIDR block**.
- PuTTY (Windows) or terminal (macOS/Linux) for SSH.
- Familiarity with Linux commands.
- Links to download Prometheus, Node Exporter, and Grafana commands.
- Amazon Linux 2023 AMI and 30GB volume size for instances.
- Instances within the same VPC for communication.

Procedures

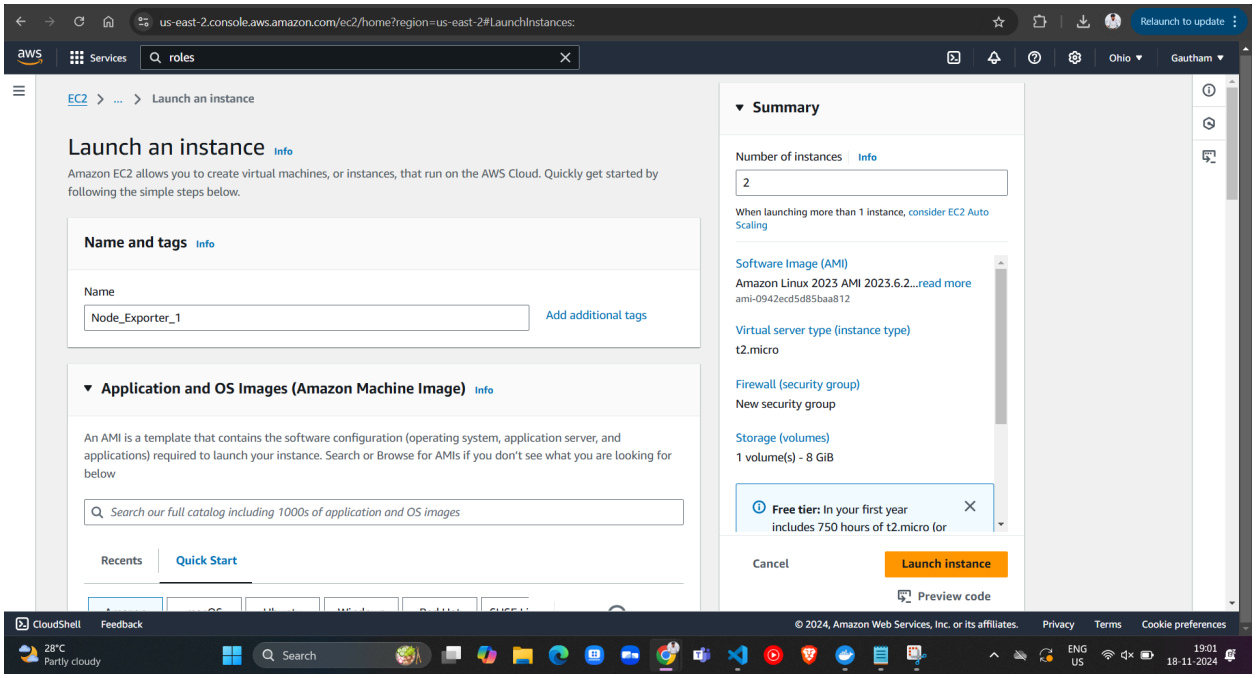
Step 1. Create 3 EC2 Instances for Prometheus and Node Exporters

Navigate to the EC2 Services page and click **Launch Instances**.

For Node Exporters:

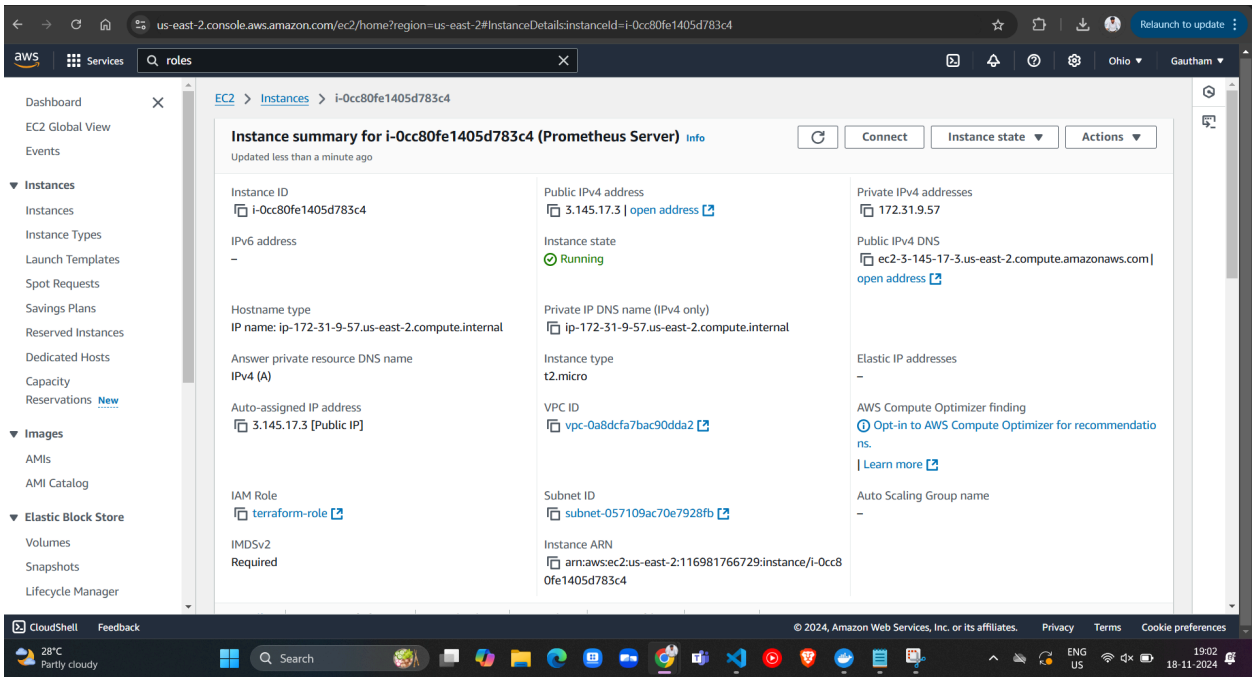
- Set the **instance count** to 2.
- Provide a name and select a suitable AMI.

- Choose a **key pair** and select the **security group** with **All TCP traffic** and an open **CIDR block**.
- Increase the **volume size** by 30GB and click **Launch Instances**.



For Prometheus Server:

- Repeat the same steps.
- Provide a unique name, select the same key pair and security group, and increase the volume by 30GB.



Step 2. Install Prometheus on One EC2 Instance and Node Exporters on the Other Two

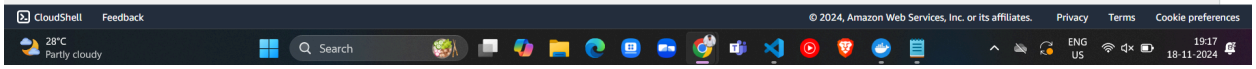
Install Prometheus

- **Connect to the Prometheus server** via PuTTY using its public IP.
- Run the following commands:
  - Switch to the root user:
    ```
    sudo su -
    ```
  - Download Prometheus:
    ```
    sudo wget <prometheus_tar_file_link>
    ```
  - Extract the downloaded tar file:
    ```
    tar -xvzf <tar_file_name>
    ```
  - Move binaries to `/usr/local/bin`:
    ```
    mv prometheus promtool /usr/local/bin/
    ```
  - Start Prometheus:
    ```
    ./prometheus --config.file=prometheus.yml
    ```

```
[ec2-user@ip-172-31-5-0 ~]$ sudo su -
[root@ip-172-31-5-0 ~]# sudo wget https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz
--2024-11-18 13:43:04--  https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/555f261a-3131-44a2-86fa-f3baac617a7e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241118T134304Z&X-Amz-Expires=300&X-Amz-Signature=3d46ba748249fca78615d71e618ef0cf460f6f3c9d9608e3f63ca5a76ca1b5bc&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.53.3.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-11-18 13:43:04--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/555f261a-3131-44a2-86fa-f3baac617a7e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241118T134304Z&X-Amz-Expires=300&X-Amz-Signature=3d46ba748249fca78615d71e618ef0cf460f6f3c9d9608e3f63ca5a76ca1b5bc&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.53.3.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 104207826 (99M) [application/octet-stream]
Saving to: 'prometheus-2.53.3.linux-amd64.tar.gz'

prometheus-2.53.3.linux-amd64.tar.gz     100%[=========================================================================>]  99.38M  85.4MB/s    in 1.2s
```

i-0c80ddc231d17fb1e (Prometheus)

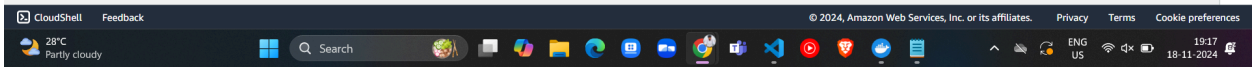
PublicIPs: 18.117.78.3 PrivateIPs: 172.31.5.0




```
prometheus-2.53.3.linux-amd64  prometheus-2.53.3.linux-amd64.tar.gz
[root@ip-172-31-5-0 ~]# cd prometheus-2.53.3.linux-amd64
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# mv prometheus promtool /usr/local/bin/
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: No such file or directory
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# cd ..
[root@ip-172-31-5-0 ~]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: No such file or directory
[root@ip-172-31-5-0 ~]# cd prometheus-2.53.3.linux-amd64/
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: No such file or directory
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: No such file or directory
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: No such file or directory
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# promethus
-bash: promethus: command not found
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# prometheus
ts=2024-11-18T13:46:24.080Z caller=main.go:589 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2024-11-18T13:46:24.080Z caller=main.go:633 level=info msg="Starting Prometheus Server" mode=server version="(version=2.53.3, branch=HEAD, revision=1491d29fb1e8f8acbab29fd54fd4ce9be2cbd7bc)"
ts=2024-11-18T13:46:24.080Z caller=main.go:638 level=info build_context="(go=go1.22.8, platform=linux/amd64, user=root@c6939e39a10c, date=20241105-12:18:07, tags=netgo,builtinassets,stringlabels)"
ts=2024-11-18T13:46:24.080Z caller=main.go:639 level=info host_details="(Linux 6.1.115-126.197.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Nov  5 17:36:57 UTC 2024 x86_64 ip-172-31-5-0.us-east-2.compute.internal (none))"
```

i-0c80ddc231d17fb1e (Prometheus)

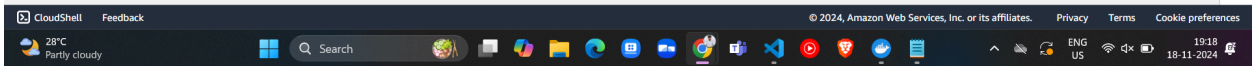
PublicIPs: 18.117.78.3 PrivateIPs: 172.31.5.0




```
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# prometheus --config.file=prometheus.yml
ts=2024-11-18T13:48:17.727Z caller=main.go:589 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2024-11-18T13:48:17.728Z caller=main.go:633 level=info msg="Starting Prometheus Server" mode=server version="(version=2.53.3, branch=HEAD, revision=1491d29fb1e8f8acbab29fd54fd4ce9be2cbd7bc)"
ts=2024-11-18T13:48:17.728Z caller=main.go:638 level=info build_context="(go=go1.22.8, platform=linux/amd64, user=root@c6939e39a10c, date=20241105-12:18:07, tags=netgo,builtinassets,stringlabels)"
ts=2024-11-18T13:48:17.728Z caller=main.go:639 level=info host_details="(Linux 6.1.115-126.197.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Nov  5 17:36:57 UTC 2024 x86_64 ip-172-31-5-0.us-east-2.compute.internal (none))"
ts=2024-11-18T13:48:17.728Z caller=main.go:640 level=info fd_limits="(soft=65535, hard=65535)"
ts=2024-11-18T13:48:17.728Z caller=main.go:641 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-11-18T13:48:17.730Z caller=web.go:568 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2024-11-18T13:48:17.730Z caller=main.go:1148 level=info msg="Starting TSDB ..."
ts=2024-11-18T13:48:17.738Z caller=head.go:626 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2024-11-18T13:48:17.738Z caller=head.go:713 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=1.653µs
ts=2024-11-18T13:48:17.738Z caller=head.go:721 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-11-18T13:48:17.739Z caller=tls_config.go:313 level=info component=web msg="Listening on" address=[::]:9090
ts=2024-11-18T13:48:17.739Z caller=tls_config.go:316 level=info component=web msg="TLS is disabled." http2=false address=[::]:9090
ts=2024-11-18T13:48:17.739Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=1 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=2 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=head.go:830 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=29.586µs wal_replay_duration=3.197215ms wbl_replay_duration=601ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=1.653µs total_replay_duration=3.324953ms
ts=2024-11-18T13:48:17.743Z caller=main.go:1169 level=info fs_type=XFS_SUPER_MAGIC
ts=2024-11-18T13:48:17.743Z caller=main.go:1172 level=info msg="TSDB started"
ts=2024-11-18T13:48:17.743Z caller=main.go:1354 level=info msg="Loading configuration file" filename=prometheus.yml
ts=2024-11-18T13:48:17.746Z caller=main.go:1391 level=info msg="updated GOGC" old=100 new=75
ts=2024-11-18T13:48:17.746Z caller=main.go:1402 level=info msg="Completed loading of configuration file" filename=prometheus.yml totalDuration=2.745167ms db_storage=1.313µs remote_storage=1.197µs web_handler=725ns query_engine=962ns scrape=2.388942ms scrape_sd=17.943µs notify=20.539µs notify_sd=14.073µs rules=1.375µs traci
```

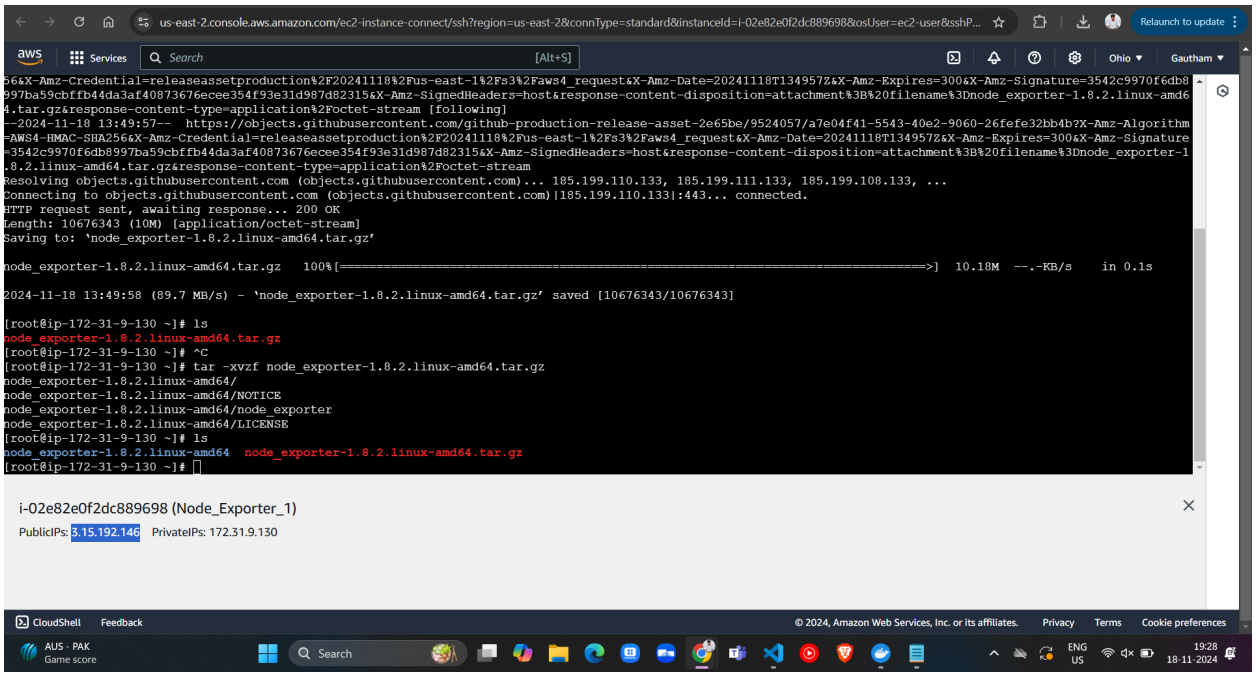
i-0c80ddc231d17fb1e (Prometheus)

PublicIPs: 18.117.78.3 PrivateIPs: 172.31.5.0



Install Node Exporter

- **Connect to each Node Exporter instance**.
- Run the following commands:
    - Download Node Exporter:
      `sudo wget <node_exporter_link>`
    - Extract the tar file:
      `tar -xvzf <tar_file_name>`
    - No need to move the Node Exporter folder to `/usr/local/bin`.
    - Run using command : ./node_exporter &



Step 3. Create 3 EC2 Instances for Prometheus and Node Exporters

Navigate to the `prometheus.yml` file on the Prometheus server.

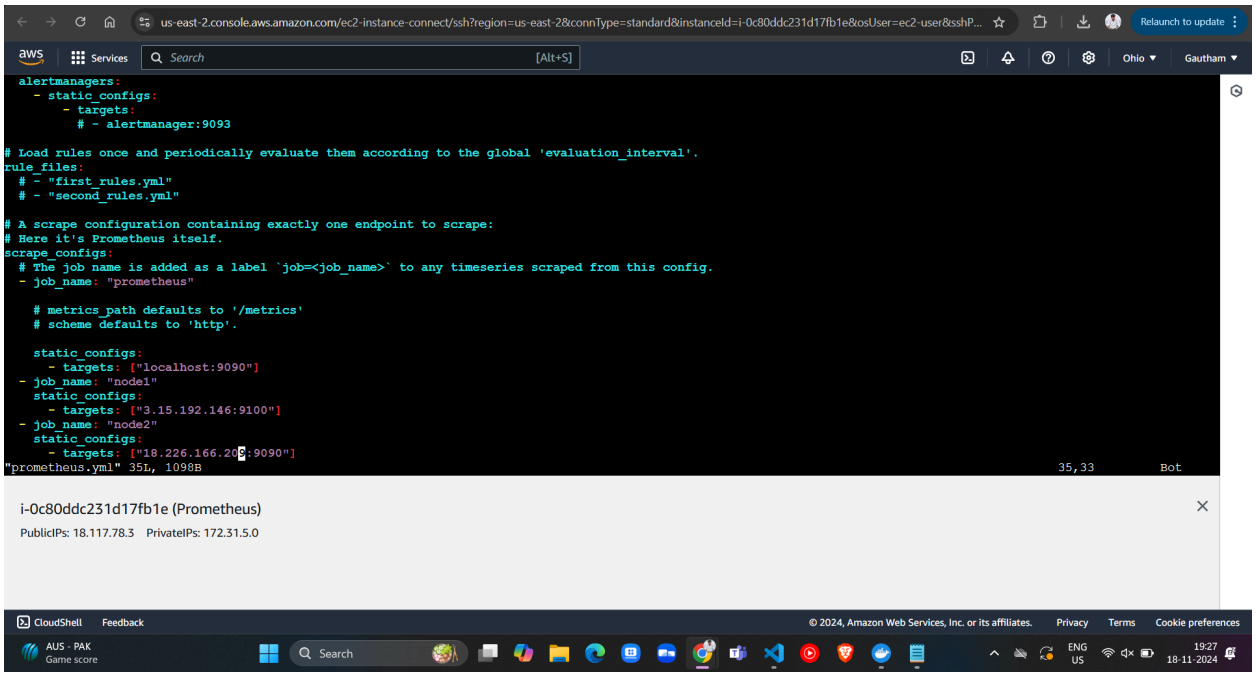
- Add the public IP addresses of the two Node Exporter instances under the `targets` section.
- Example configuration:

```
scrape_configs:

  - job_name: 'node_exporters'

    static_configs:
```

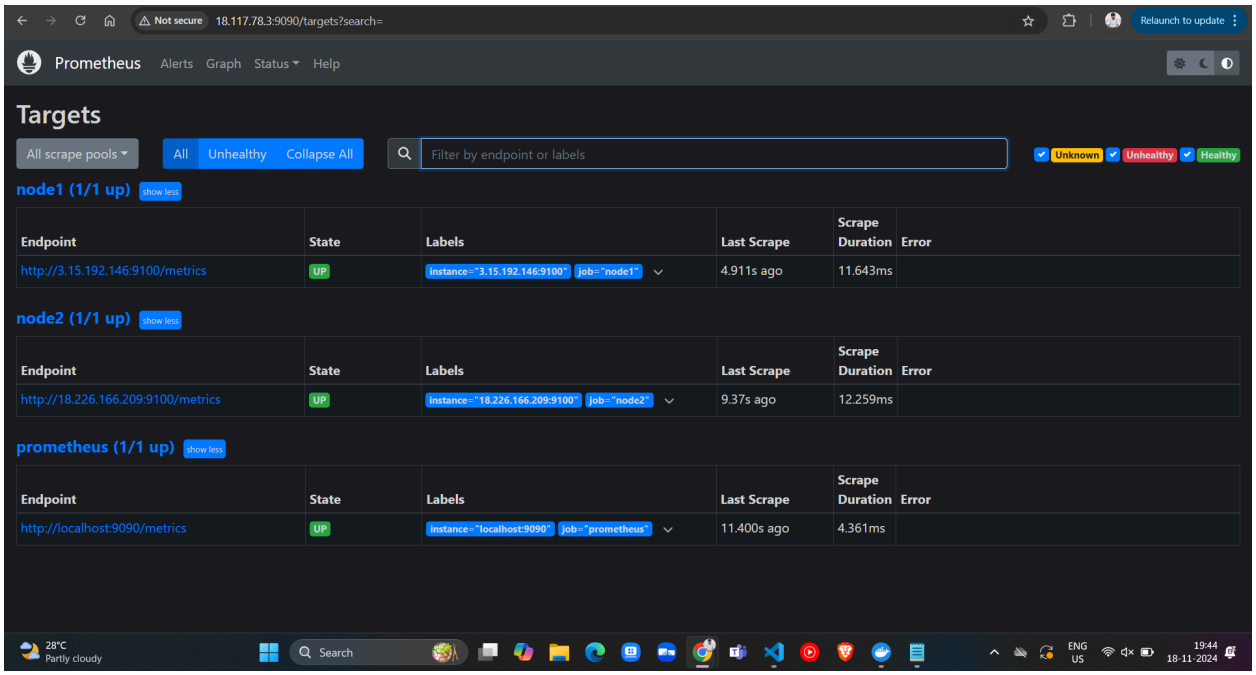
```
    - targets:
        - '<node_exporter_1_public_ip>:9100'
        - '<node_exporter_2_public_ip>:9100'
```

- Restart Prometheus:

```
pkill prometheus

./prometheus --config.file=prometheus.yml
```



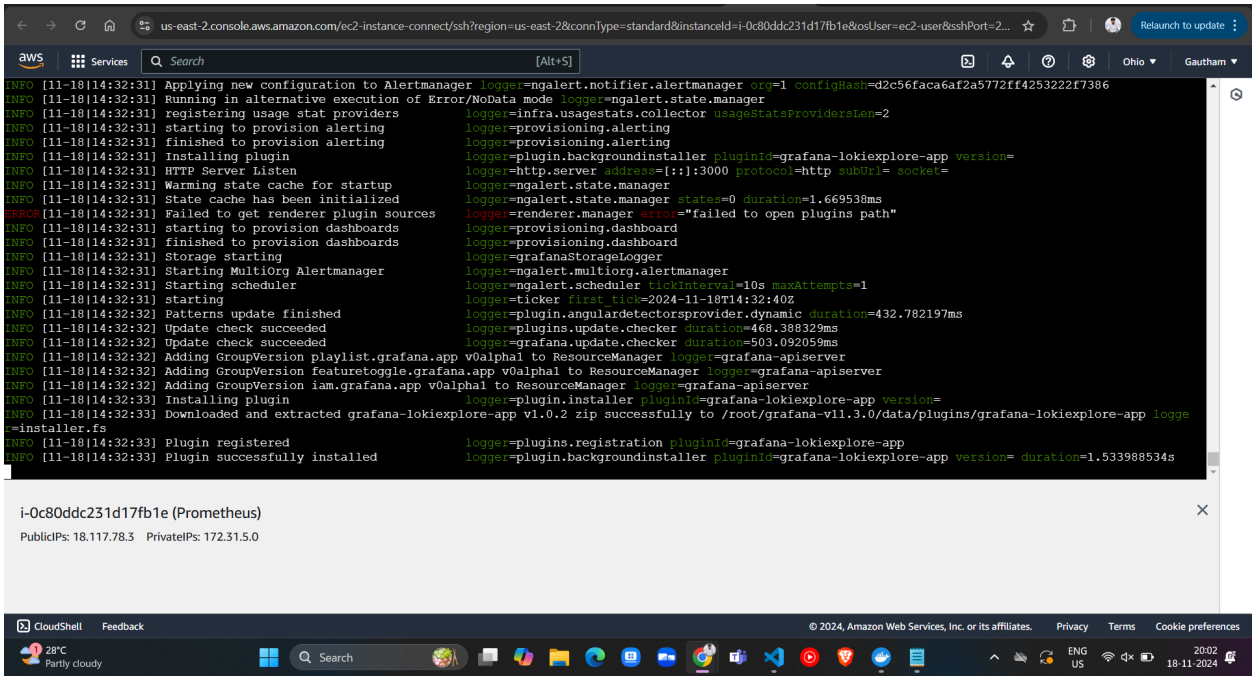
Access the Prometheus dashboard using the public IP of the Prometheus server



Step 4. Install Grafana on the Prometheus Server and Connect Prometheus as a Data Source

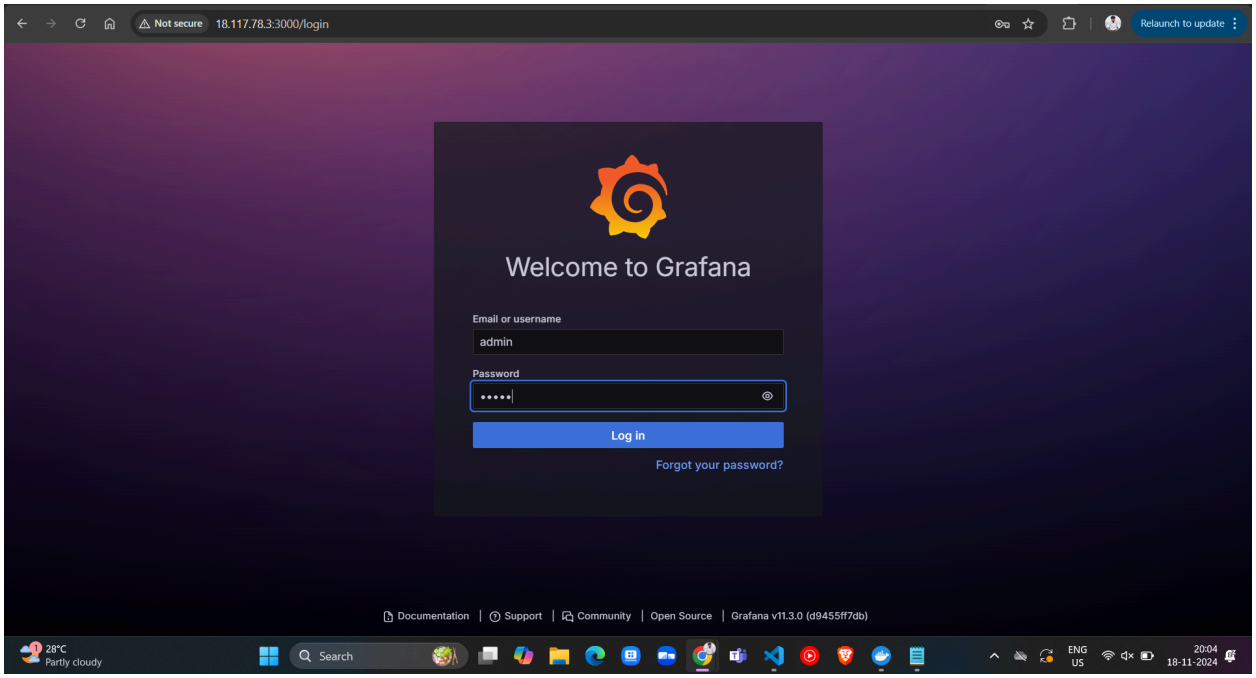
Install Grafana:

- Go to the Grafana downloads page and select **OSS edition**.
- Follow the Linux installation commands listed on the page.



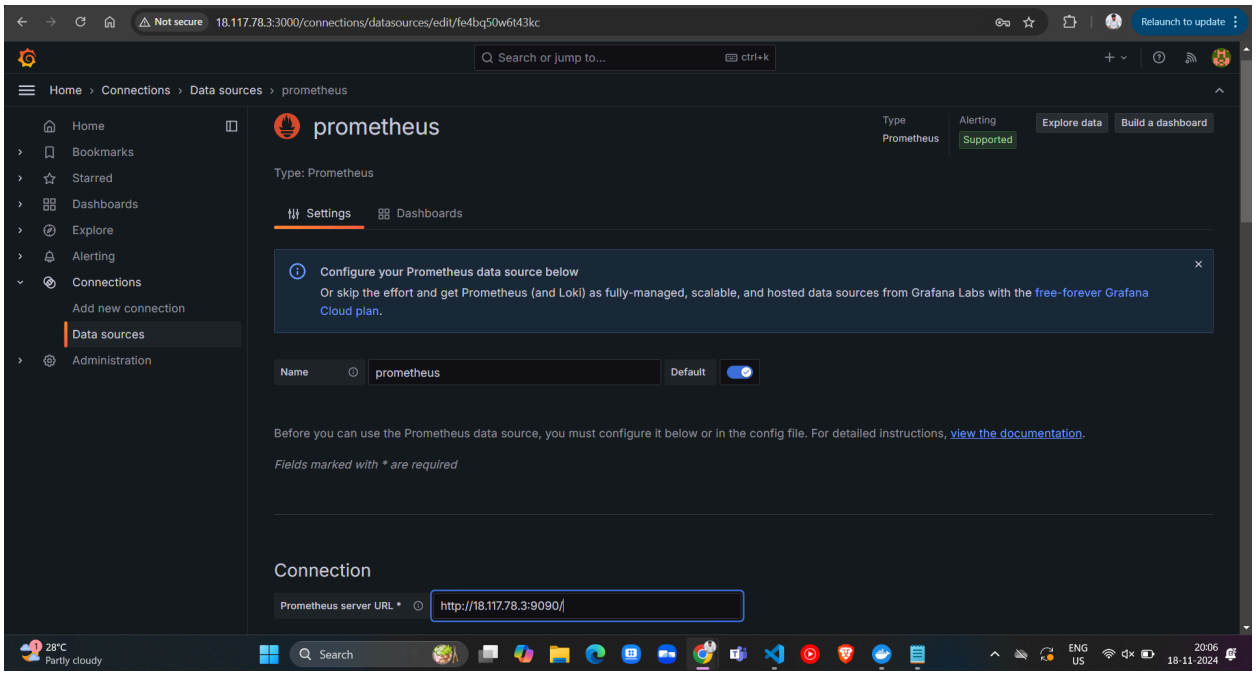
Run Grafana:

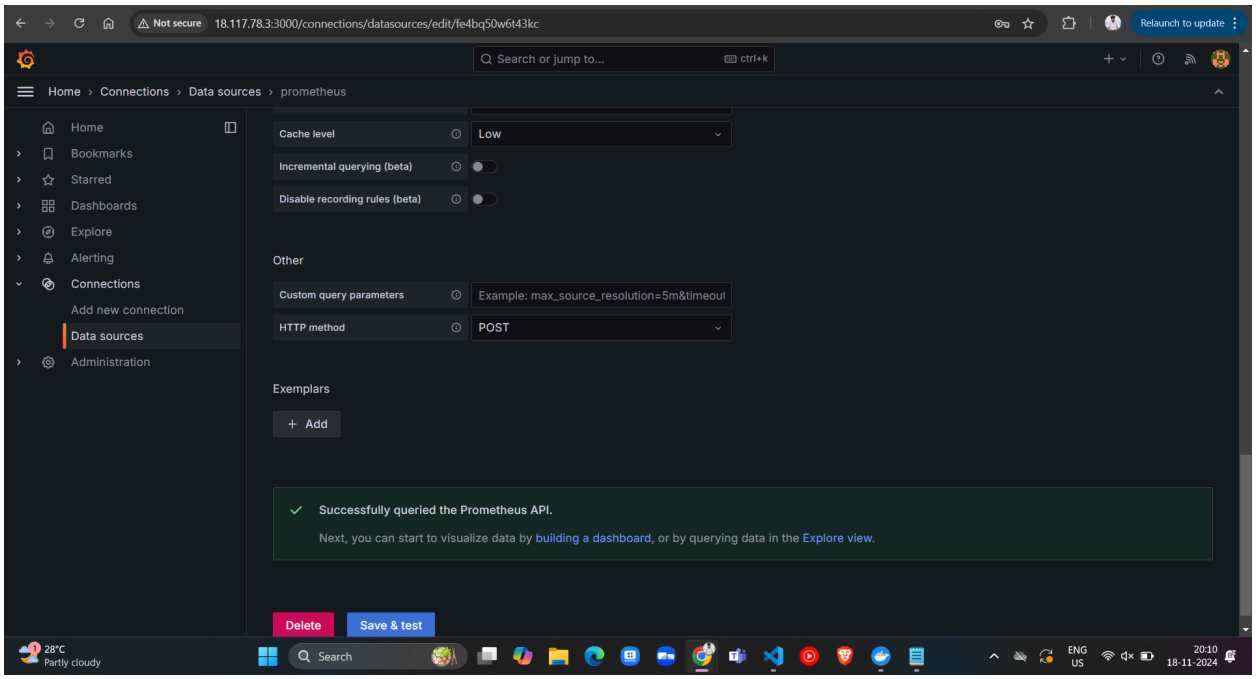
- Start Grafana and log in using the default credentials (`admin/admin`).
- Change the password when prompted.



Add Prometheus as a Data Source:

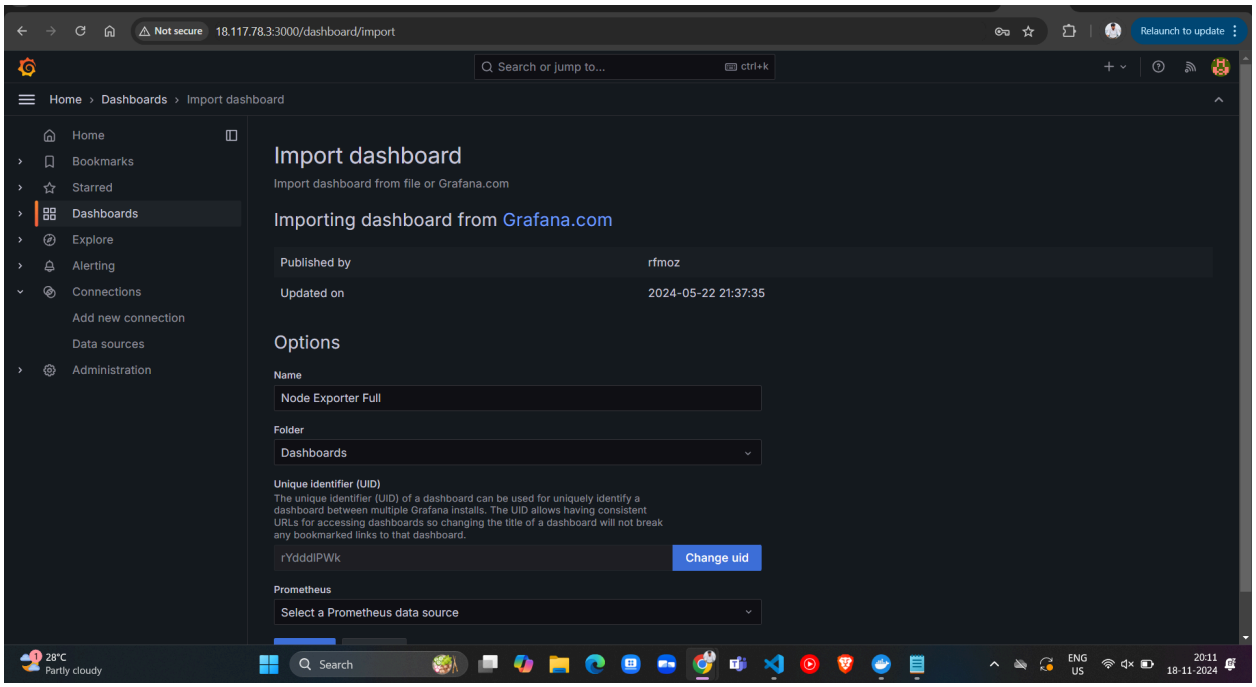
- In the Grafana dashboard, click **Add Your First Data Source**.
- Choose **Prometheus** as the data source.
- Use the URL: `http://localhost:9090` (since both Grafana and Prometheus are on the same server).
- Click **Save and Test**.





Import Node Exporter Dashboard:

- Add a Node Exporter dashboard to Grafana to visualize metrics



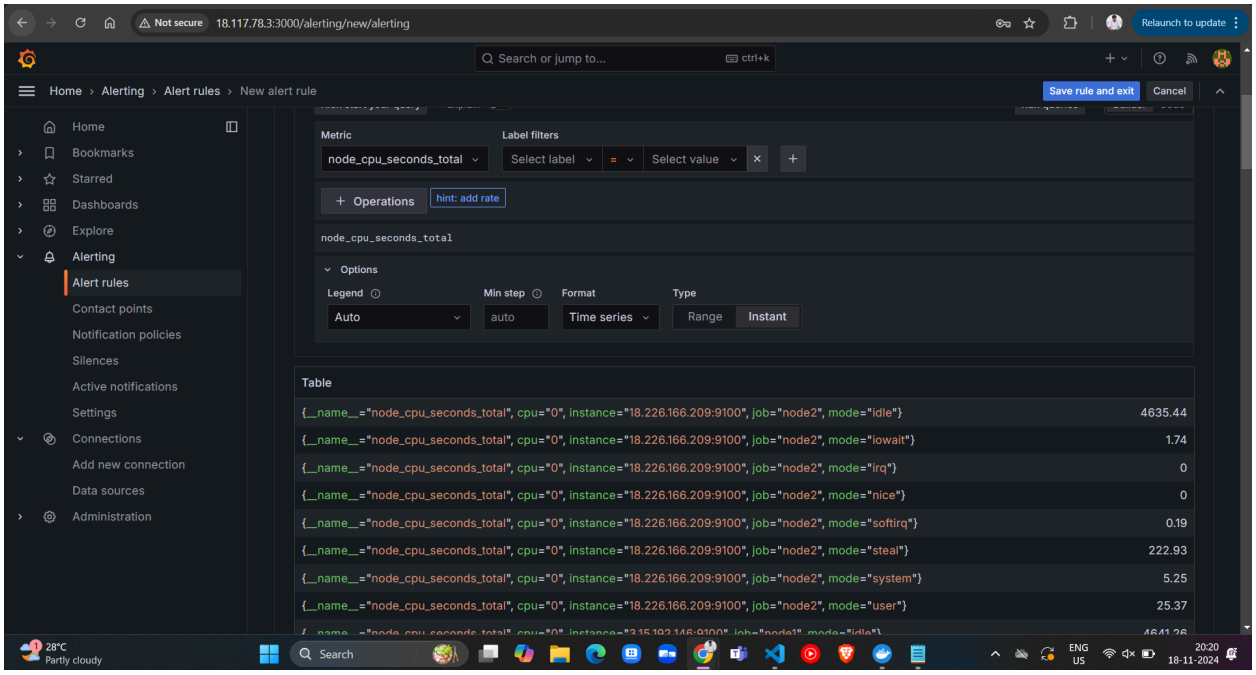
Step 5. Check Visualization

View the Grafana dashboard to ensure that metrics from Node Exporters are being displayed.

Step 6. Create Alert Rules in Grafana

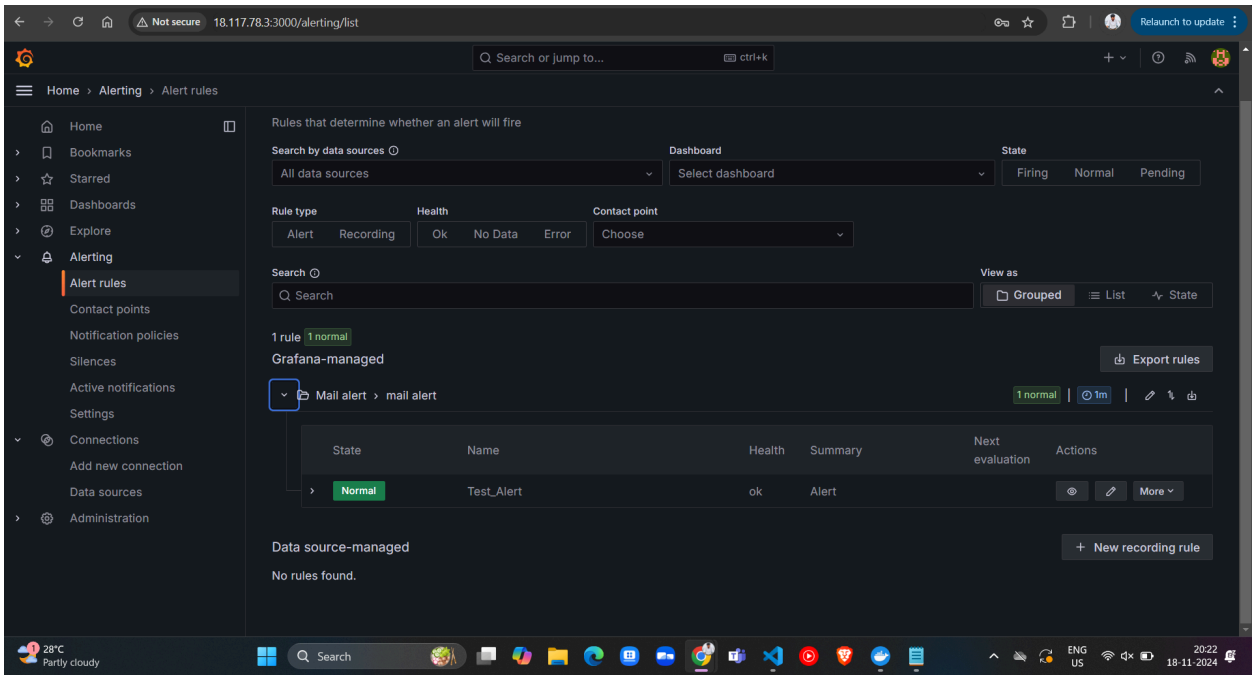
Set Up Alerts:

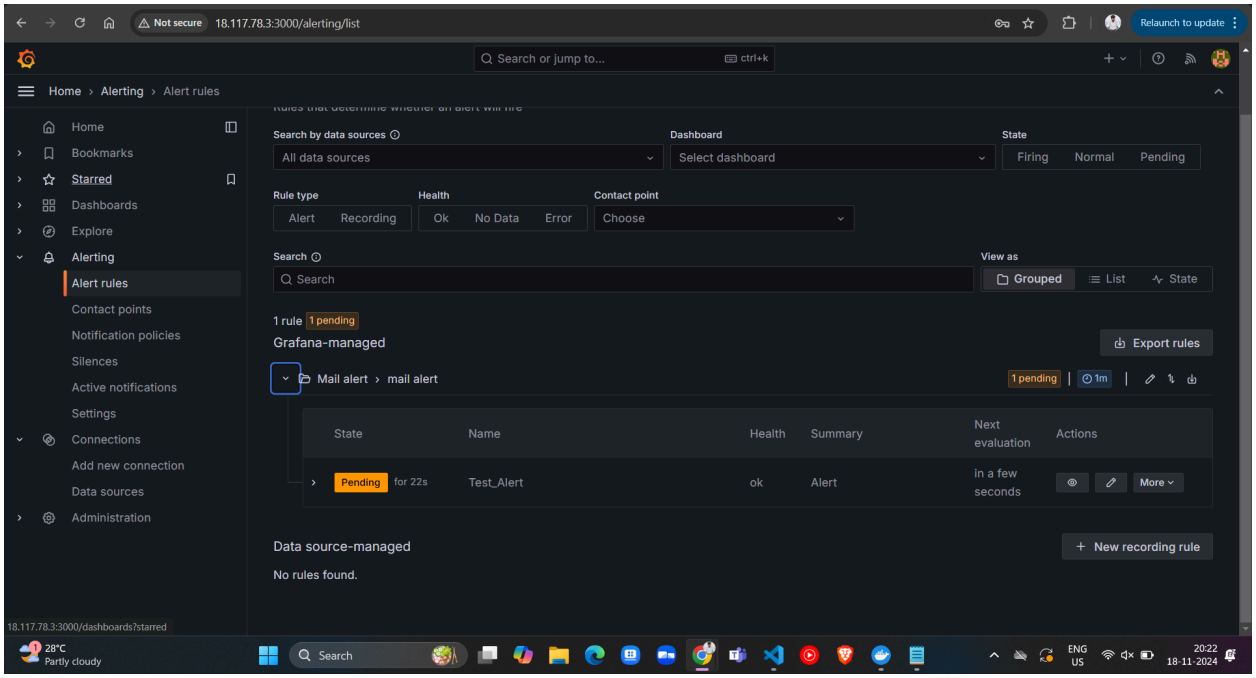
- Navigate to **Alert Rules** in Grafana.
- Provide a name for the alert and choose a suitable metric.
- Configure the **threshold condition** for the alert.
- Add a **notification message** and set Grafana default email as the notification channel.

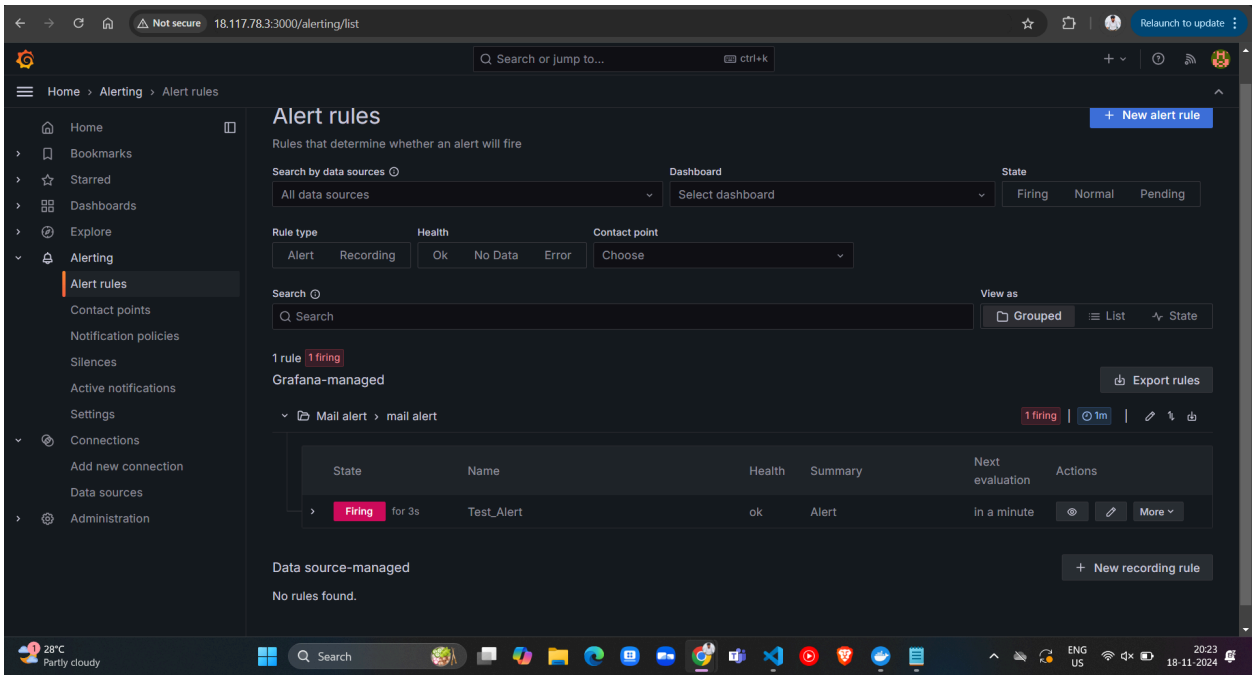


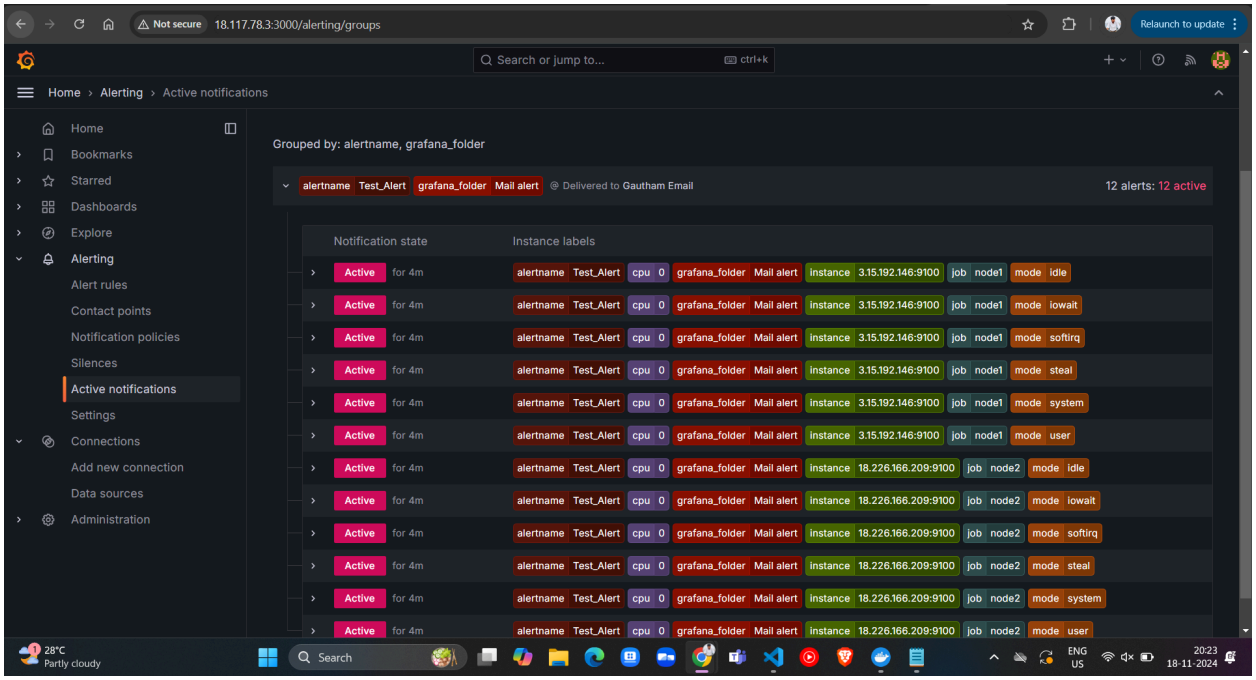
Monitor Alerts:

- Wait for the alert state to transition from **active → pending → firing**.


Not secure
18.117.78.3:3000/alerting/list
Relaunch to update
Search or jump to...
ctrl+k
Home › Alerting › Alert rules
Home
Bookmarks
Starred
Dashboards
Explore
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Settings
Connections
Add new connection
Data sources
Administration
Rules that determine whether an alert will fire
Search by data sources
All data sources
Dashboard
Select dashboard
State
Firing
Normal
Pending
Rule type
Alert
Recording
Health
Ok
No Data
Error
Contact point
Choose
Search
Search
View as
Grouped
List
State
1 rule
1 normal
Grafana-managed
Export rules
Mail alert › mail alert
1 normal
1m
State
Name
Health
Summary
Next evaluation
Actions
Normal
Test_Alert
ok
Alert
More
Data source-managed
New recording rule
No rules found.
28°C
Partly cloudy
Search
ENG
US
20:22
18-11-2024

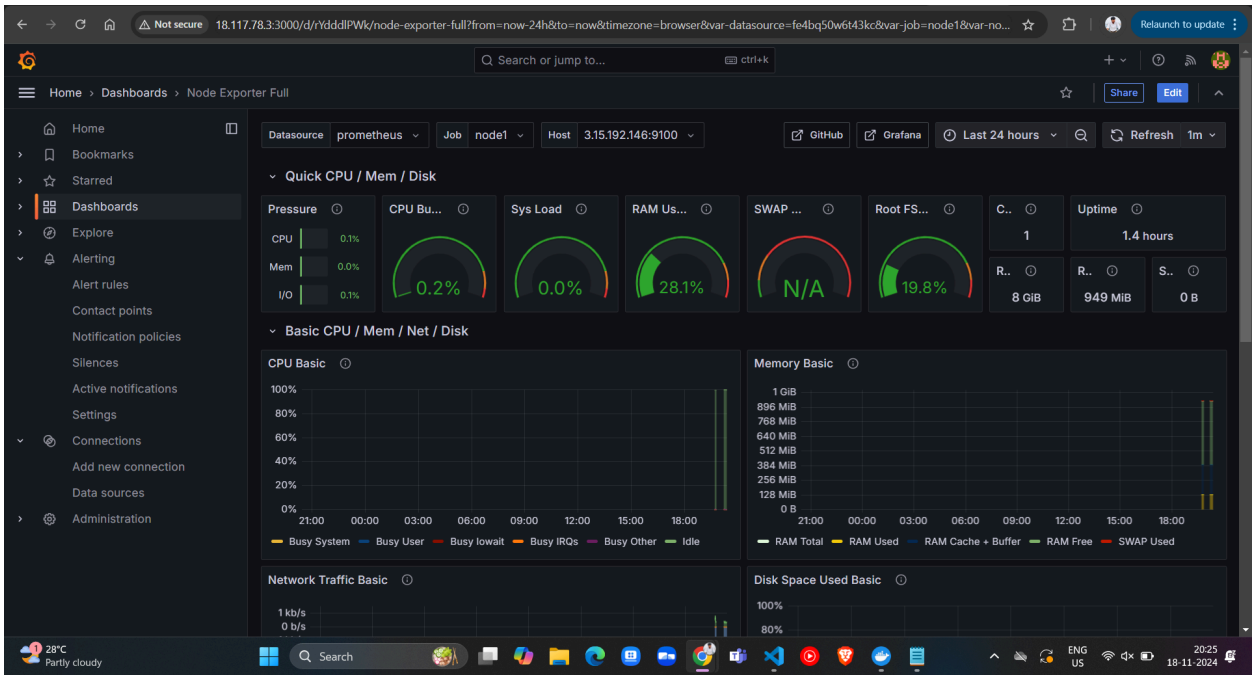

Not secure
18.117.78.3:3000/alerting/list
Relaunch to update
Search or jump to...
ctrl+k
Home › Alerting › Alert rules
Home
Bookmarks
Starred
Dashboards
Explore
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Settings
Connections
Add new connection
Data sources
Administration
Search by data sources
All data sources
Dashboard
Select dashboard
State
Firing
Normal
Pending
Rule type
Alert
Recording
Health
Ok
No Data
Error
Contact point
Choose
Search
Search
View as
Grouped
List
State
1 rule
1 pending
Grafana-managed
Export rules
Mail alert › mail alert
1 pending
1m
State
Name
Health
Summary
Next evaluation
Actions
Pending
for 22s
Test_Alert
ok
Alert
in a few seconds
More
Data source-managed
New recording rule
No rules found.
18.117.78.3:3000/dashboards?starred
28°C
Partly cloudy
Search
ENG
US
20:22
18-11-2024





Step 7. Verify the Output

- Confirm that Prometheus, Node Exporters, and Grafana are working as expected.
- Ensure the alert rules trigger appropriately and dashboards display accurate metrics.



Conclusion

This documentation outlines the complete process for setting up Prometheus, Node Exporters, and Grafana on AWS EC2 instances. Following these steps enables efficient monitoring and visualization of system metrics with alerts configured for critical thresholds.