

# 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

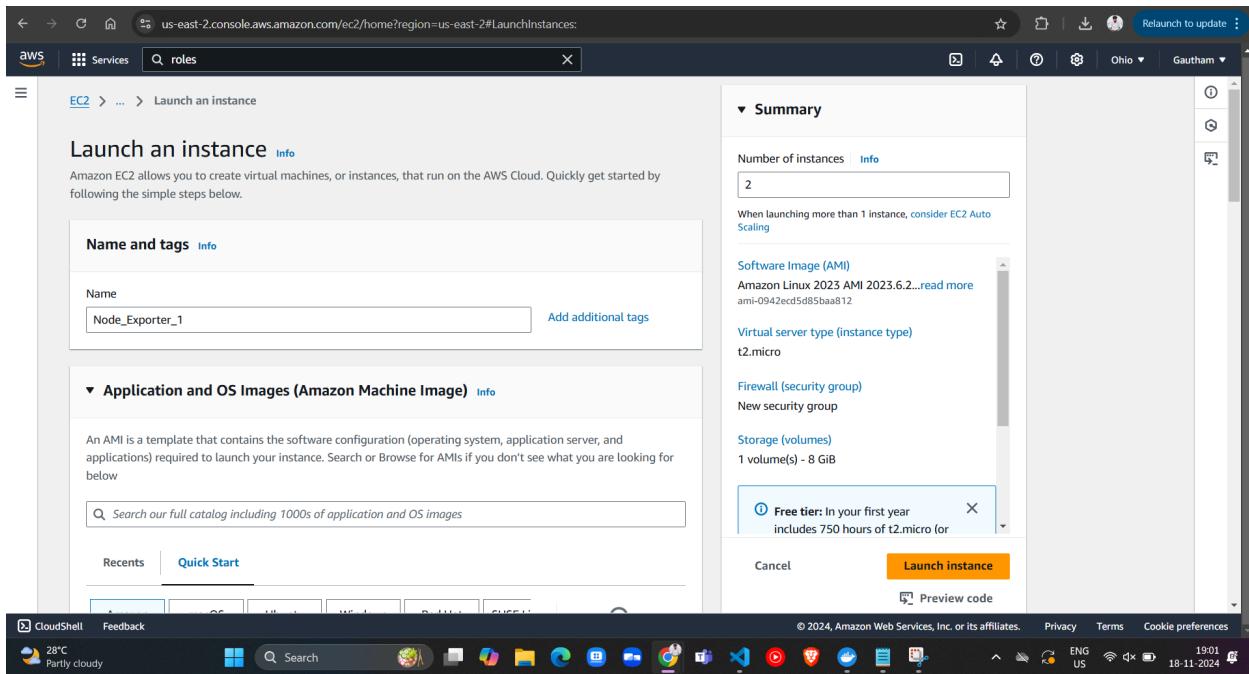
### Step 1. Create 3 EC2 Instances for Prometheus and Node Exporters

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.

The screenshot shows the AWS EC2 Instances details page for an instance named i-0cc80fe1405d783c4. The instance is a t2.micro type running on a subnet ID of subnet-057109ac70e7928fb. It has a public IP of 3.145.17.3 and a private IP of 172.31.9.57. The instance is currently running. The instance role is terraform-role, and it requires IMDSv2.

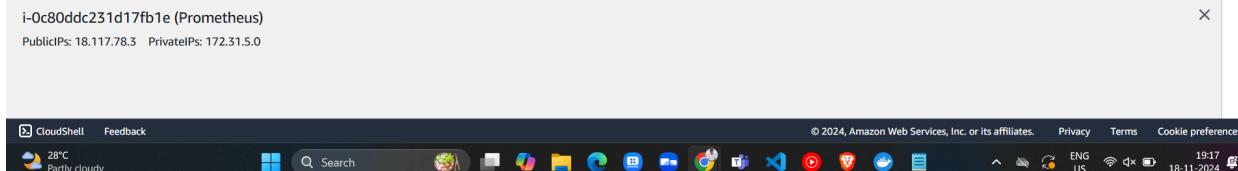
## Step 2. Install Prometheus on One EC2 Instance and Node Exporters on the Other Two

### Install Prometheus

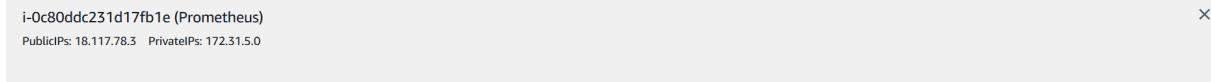
- **Connect to the Prometheus server** via PuTTY using its public IP.
- Run the following commands:
  - Switch to the root user:  
`sudo su -`
  - Download Prometheus:  
`sudo wget <prometheus_tar_file_link>`
  - Extract the downloaded tar file:  
`tar -xvzf <tar_file_name>`
  - Move binaries to `/usr/local/bin`:  
`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=releaseassetprodution%2F20241118%2Fus-east-1%2F53%2Faws4_request&X-Amz-Date=20241118T134304Z&X-Amz-Expires=300K&X-Amz-Signature=3d46ba748249fc78615d7le618ef0c460f63c9d608e3f63ca5a763cab5bc4X-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=releaseassetprodution%2F20241118%2Fus-east-1%2F53%2Faws4_request&X-Amz-Date=20241118T134304Z&X-Amz-Expires=300K&X-Amz-Signature=3d46ba748249fc78615d7le618ef0c460f63c9d608e3f63ca5a763cab5bc4X-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'

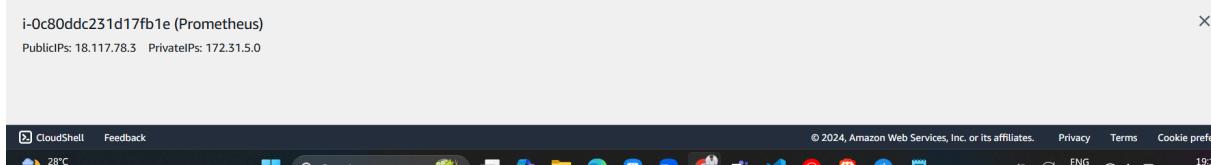
prometheus-2.53.3.linux-amd64.tar.gz    100%[=====] 99.38M  85.4MB/s   in 1.2s
```



```
[root@ip-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]# ./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]# ./prometheus --config.file=prometheus.yml
-bash: ./prometheus: 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=1491d29fb1e918acbab29fd54fd4ce9be2cb7bc)"
ts=2024-11-18T13:46:24.080Z caller=main.go:638 level=info build_context="(go=gol1.22.8, 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))"
```

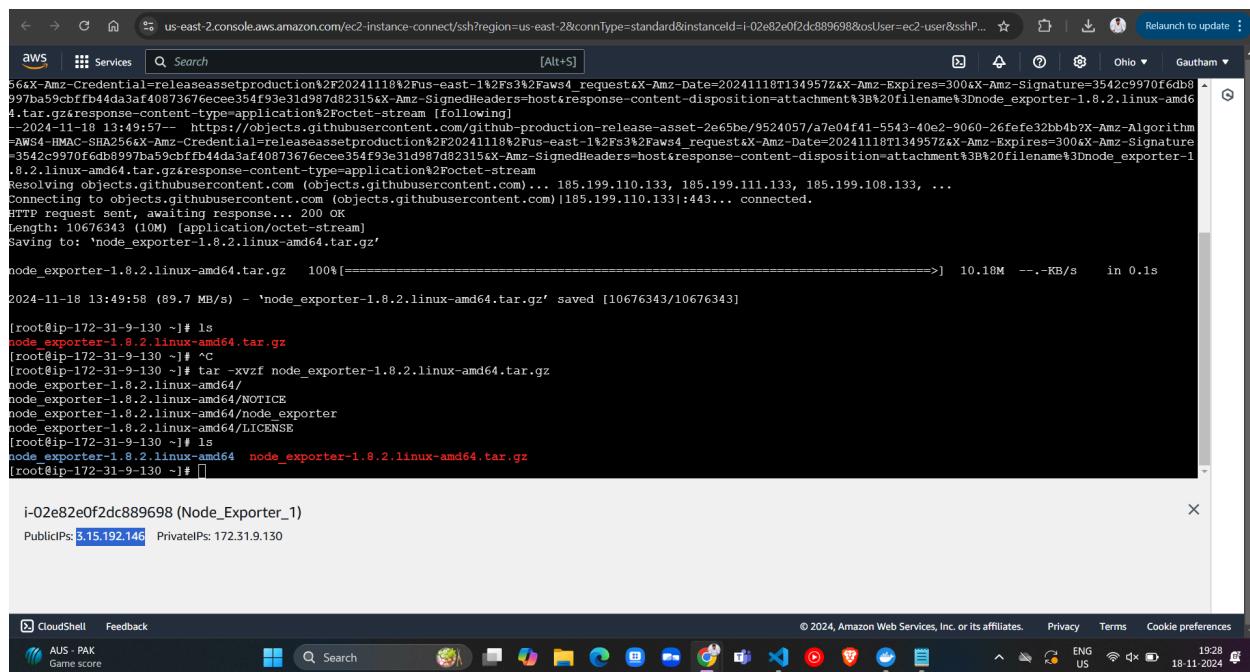


```
[root@ip-172-31-5-0 prometheus-2.53.3.linux-amd64]# prometheus --config.file=prometheus.yml
ts=2024-11-18T13:48:17.727Z caller=main.go:589 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2024-11-18T13:48:17.728Z caller=main.go:633 level=info msg="Starting Prometheus Server" mode=server version="(version=2.53.3, branch=HEAD, revision=1491d29fb1e918acbab29fd54fd4ce9be2cb7bc)"
ts=2024-11-18T13:48:17.728Z caller=main.go:638 level=info build_context="(go=gol1.22.8, 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=2024-11-18T13:48:17.738Z caller=main.go:640 level=info fd_limits="(soft=65535, hard=65535)"
ts=2024-11-18T13:48:17.738Z 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=0.0.0:9090
ts=2024-11-18T13:48:17.730Z caller=main.go:1142 level=info msg="Starting TSDB ..."
ts=2024-11-18T13:48:17.738Z caller=head.go:626 level=info component=tsd msg="Replaying on-disk memory mappable chunks if any"
ts=2024-11-18T13:48:17.738Z caller=component.tsd.go:713 level=info component=tsd 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=tsd msg="Replaying WAL, this may take a while"
ts=2024-11-18T13:48:17.739Z caller=tis.config.go:313 level=info component=web msg="Listening on" address=[::]:9090
ts=2024-11-18T13:48:17.739Z caller=tis.config.go:316 level=info component=web msg="TLS is disabled." http=false address=[::]:9090
ts=2024-11-18T13:48:17.741Z caller=component.tsd msg="WAL segment loaded" segment=0 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=head.go:793 level=info component=tso msg="WAL segment loaded" segment=1 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=component.tsd msg="WAL segment loaded" segment=2 maxSegment=2
ts=2024-11-18T13:48:17.741Z caller=head.go:830 level=info component=tso msg="WAL replay completed" checkpoint_replay_duration=29.586μs wal_replay_duration=3.197215ms wbl_replay_duration=60ms chunk_snapshot_load_duration=0μs 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_stora
ge=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
```



## 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

The screenshot shows the Prometheus Targets page with three entries:

- node1 (1/1 up)**: [\[show less\]](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://3.15.192.146:9100/metrics	UP	instance="3.15.192.146:9100" job="node1"	4.911s ago	11.643ms	

- node2 (1/1 up)**: [\[show less\]](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://18.226.166.209:9100/metrics	UP	instance="18.226.166.209:9100" job="node2"	9.37s ago	12.259ms	

- prometheus (1/1 up)**: [\[show less\]](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	11.400s ago	4.361ms	

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

### Install Grafana:

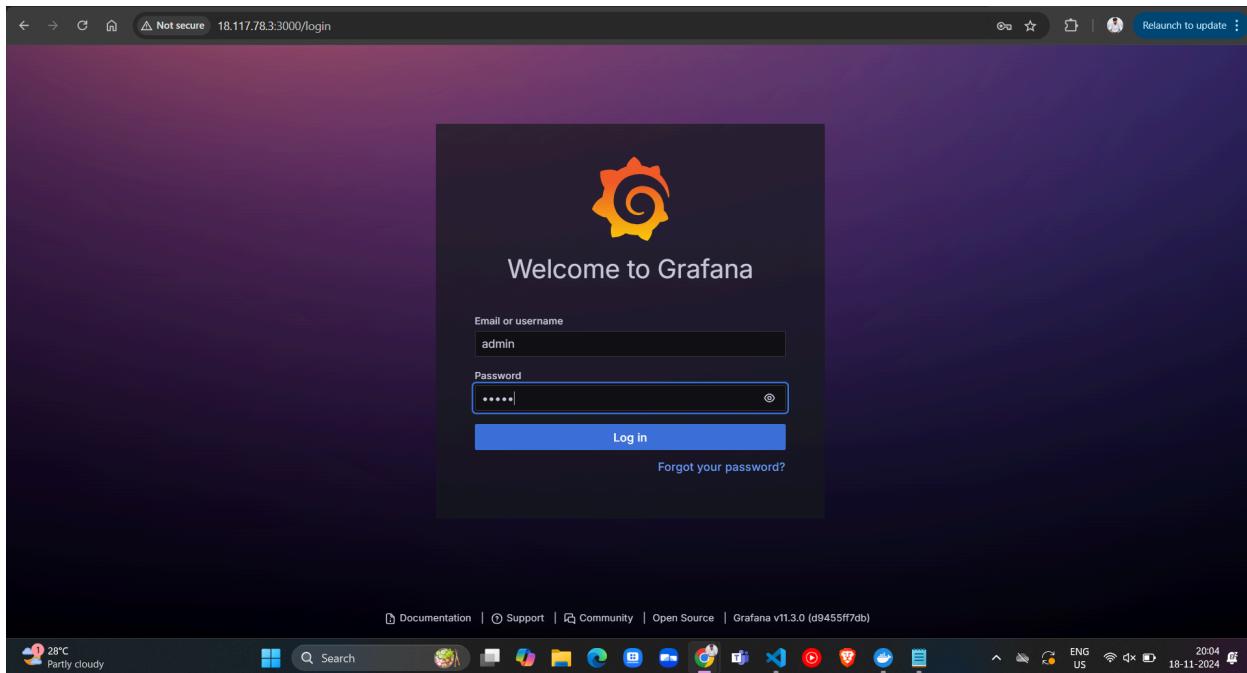
- Go to the Grafana downloads page and select **OSS edition**.
- Follow the Linux installation commands listed on the page.

```
[11-18 14:32:31] Applying new configuration to Alertmanager logger=ngalert.alertmanager.org=1 configHash=d2c56facaa6f2a5772ff425322ff7386
INFO [11-18 14:32:31] Running in alternative execution of Error/NoData mode logger=ngalert.state.manager
INFO [11-18 14:32:31] registering usage stat providers logger=infra.usagestats.collector.usageStatsProviderLen=2
INFO [11-18 14:32:31] starting to provision alerting logger=provisioning.alerting
INFO [11-18 14:32:31] finished to provision alerting logger=provisioning.alerting
INFO [11-18 14:32:31] Installing plugin logger=plugin.backgroundinstaller.pluginID=grafana-lokiexplore-app version=
INFO [11-18 14:32:31] HTTP Server Listen logger=http.server.address=[::]:3000 protocol=http subURI=
INFO [11-18 14:32:31] Warning state cache for startup logger=ngalert.state.manager
INFO [11-18 14:32:31] State cache has been initialized logger=ngalert.state.manager states=0 duration=1.669530ms
ERROR [11-18 14:32:31] Failed to get renderer plugin sources logger=renderer.manager error="failed to open plugins path"
INFO [11-18 14:32:31] starting to provision dashboards logger=provisioning.dashboard
INFO [11-18 14:32:31] finished to provision dashboards logger=provisioning.dashboard
INFO [11-18 14:32:31] Storage starting logger=grafanaStorageLogger
INFO [11-18 14:32:31] Starting MultiOrg Alertmanager logger=ngalert.multiorg.alertmanager
INFO [11-18 14:32:31] Starting scheduler logger=ngalert.scheduler.tickInterval=10s maxAttempts=1
INFO [11-18 14:32:31] starting logger=ticker first tick=2024-11-18T14:32:40Z
INFO [11-18 14:32:32] Patterns update finished logger=plugin.angulardetectorsprovider.dynamic.duration=432.782197ms
INFO [11-18 14:32:32] Update check succeeded logger=plugins.update.checker.duration=468.388329ms
INFO [11-18 14:32:32] Update check succeeded logger=grafana.update.checker.duration=503.092059ms
INFO [11-18 14:32:32] Adding GroupVersion playlist.grafana.app v0Alphal to ResourceManager logger=grafana-apiserver
INFO [11-18 14:32:32] Adding GroupVersion featuretoggle.grafana.app v0Alphal to ResourceManager logger=grafana-apiserver
INFO [11-18 14:32:32] Adding GroupVersion iam.grafana.app v0Alphal to ResourceManager logger=grafana-apiserver
INFO [11-18 14:32:33] Installing plugin logger=plugin.installer.pluginID=grafana-lokiexplore-app version=
INFO [11-18 14:32:33] Downloaded and extracted grafana-lokiexplore-app v1.0.2 zip successfully to /root/grafana-v11.3.0/data/plugins/grafana-lokiexplore-app logger=installer.fs
INFO [11-18 14:32:33] Plugin registered logger=plugins.registration.pluginID=grafana-lokiexplore-app
INFO [11-18 14:32:33] Plugin successfully installed logger=plugin.backgroundinstaller.pluginID=grafana-lokiexplore-app version= duration=1.533988534s
```

i-0c80ddc231d17fb1e (Prometheus)  
PublicIPs: 18.117.78.3 PrivateIPs: 172.31.5.0

## 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**.

The screenshot shows the Grafana interface for configuring a Prometheus data source. The URL is 18.117.78.3:3000/connections/datasources/edit/fe4bq50w6t43kc. The left sidebar shows 'Data sources' selected under 'Connections'. The main panel title is 'prometheus' (Type: Prometheus). A message box says 'Configure your Prometheus data source below' or skip to the Cloud plan. The 'Name' field is set to 'prometheus' with 'Default' checked. Below it, a note says 'Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#)'. The 'Connection' section shows the 'Prometheus server URL' as 'http://18.117.78.3:9090/'. The system tray at the bottom indicates a weather of 28°C and partly cloudy.

This screenshot shows the advanced configuration options for the same Prometheus data source. It includes fields for 'Cache level' (set to 'Low'), 'Incremental querying (beta)', and 'Disable recording rules (beta)'. Under 'Other', there are fields for 'Custom query parameters' (example: max\_source\_resolution=5m&timeout) and 'HTTP method' (set to 'POST'). The 'Exemplars' section has a '+ Add' button. A success message box states 'Successfully queried the Prometheus API.' and 'Next, you can start to visualize data by [building a dashboard](#), or by querying data in the Explore view.' The bottom navigation bar includes 'Delete' and 'Save & test' buttons. The system tray at the bottom shows the same weather information.

## Import Node Exporter Dashboard:

- Add a Node Exporter dashboard to Grafana to visualize metrics

The screenshot shows the 'Import dashboard' screen in Grafana. The left sidebar is titled 'Dashboards' and includes options like Home, Bookmarks, Starred, Explore, Alerting, Connections, Data sources, and Administration. The main area has a title 'Import dashboard' and sub-sections for 'Importing dashboard from Grafana.com' (published by rfmoz, updated on 2024-05-22 21:37:35) and 'Options'. In the 'Options' section, the 'Name' field is set to 'Node Exporter Full', the 'Folder' is 'Dashboards', and the 'Unique identifier (UID)' is 'rYdddIPWk' with a 'Change uid' button. A dropdown for 'Prometheus' is set to 'Select a Prometheus data source'. The bottom of the window shows a toolbar with various icons and system status information.

## 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.

The screenshot shows the Grafana web interface at the URL `18.117.78.3:3000/alerting/new/alerting`. The left sidebar is open, showing the navigation path: Home > Alerting > Alert rules > New alert rule. The 'Alert rules' option is highlighted. The main panel is titled 'New alert rule' and contains a configuration form. The 'Metric' dropdown is set to 'node\_cpu\_seconds\_total'. Below it, there's a 'Label filters' section with a 'Select label' dropdown and a 'Select value' dropdown. A 'hint: add rate' button is also present. The 'Options' section includes a 'Legend' dropdown set to 'Auto', a 'Min step' dropdown set to 'auto', a 'Format' dropdown set to 'Time series', and a 'Type' dropdown set to 'Instant'. At the bottom of the page, there's a table showing a list of metrics with their values:

Metric	Value
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="idle"}	4635.44
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="iowait"}	1.74
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="irq"}	0
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="nice"}	0
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="softirq"}	0.19
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="steal"}	222.93
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="system"}	5.25
{__name__="node_cpu_seconds_total",cpu="0",instance="18.226.166.209:9100",job="node2",mode="user"}	25.37
f_name="node_cpu_seconds_total",cpu="0",instance="3.15.192.146:9100",job="node1",mode="idle"	4641.26

## Monitor Alerts:

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

Not secure 18.117.78.3:3000/alerting/list

Search or jump to... ctrl+k

Home > Alerting > Alert rules

Rules that determine whether an alert will fire

Search by data sources Dashboard State

All data sources Select dashboard Firing Normal Pending

Rule type Health Contact point

Alert Recording Ok No Data Error Choose

View as Grouped List State

1 rule 1 normal Grafana-managed

Mail alert > mail alert

State	Name	Health	Summary	Next evaluation	Actions
Normal	Test_Alert	ok	Alert	in a few seconds	More

+ New recording rule

28°C Partly cloudy 2022 ENG US 18-11-2024

Not secure 18.117.78.3:3000/alerting/list

Search or jump to... ctrl+k

Home > Alerting > Alert rules

Rules that determine whether an alert will fire

Search by data sources Dashboard State

All data sources Select dashboard Firing Normal Pending

Rule type Health Contact point

Alert Recording Ok No Data Error Choose

View as Grouped List State

1 rule 1 pending Grafana-managed

Mail alert > mail alert

State	Name	Health	Summary	Next evaluation	Actions
Pending	Test_Alert	ok	Alert	in a few seconds	More

+ New recording rule

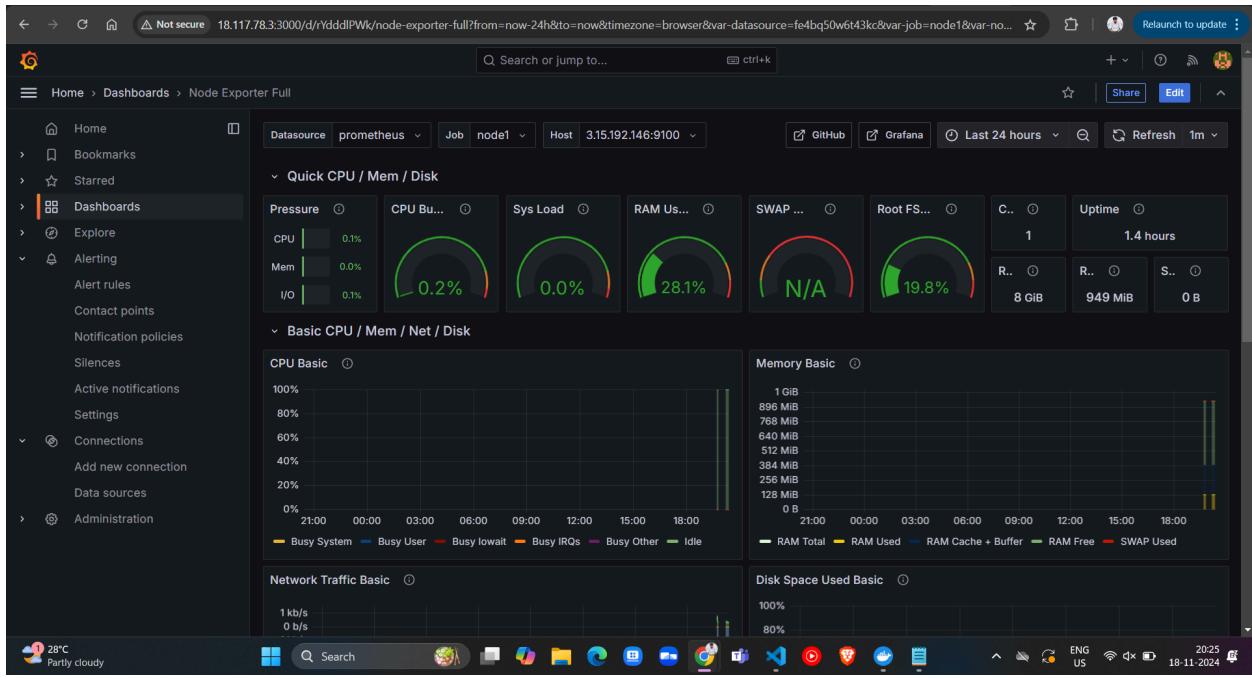
18.117.78.3:3000/dashboards?starred 28°C Partly cloudy 2022 ENG US 18-11-2024

The screenshot shows the Grafana Alerting interface at the URL [18.117.78.3:3000/alerting/list](http://18.117.78.3:3000/alerting/list). The left sidebar is collapsed, and the main area is titled "Alert rules". A search bar at the top says "Search or jump to...". Below it are filters for "Search by data sources" (set to "All data sources"), "Dashboard" (set to "Select dashboard"), and "State" (set to "Firing, Normal, Pending"). A table header row includes "Rule type" (Alert), "Health" (Ok, No Data, Error), and "Contact point" (Choose). A search bar below the table header is set to "Q. Search". The table displays one rule: "1 rule 1 firing" under the heading "Grafana-managed". The rule details show "Mail alert > mail alert" with a status of "Firing for 3s", "Name: Test\_Alert", "Health: ok", "Summary: Alert", and "Next evaluation: in a minute". The "Actions" column has a "More" button. Below this, a section for "Data source-managed" shows "No rules found." A "New recording rule" button is available. The bottom of the screen shows the Windows taskbar with various icons.

The screenshot shows the Grafana Alerting interface at the URL [18.117.78.3:3000/alerting/groups](http://18.117.78.3:3000/alerting/groups). The left sidebar is collapsed, and the main area is titled "Active notifications". A search bar at the top says "Search or jump to...". Below it is a filter bar with "alertname: Test\_Alert", "grafana\_folder: Mail alert", and "Delivered to: Gautham Email". The text "12 alerts: 12 active" is displayed. A table header row includes "Notification state" (Active) and "Instance labels". The table lists 12 rows, each representing an active alert instance. Each row contains the alert name, state (Active for 4m), and various labels: "alertname: Test\_Alert", "cpu: 0", "grafana\_folder: Mail alert", "instance: 3.15.192.146:9100", "job: node1", and "mode: idle". Other modes listed include iowait, softirq, steal, system, user, and so on. The bottom of the screen shows the Windows taskbar with various icons.

## 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.