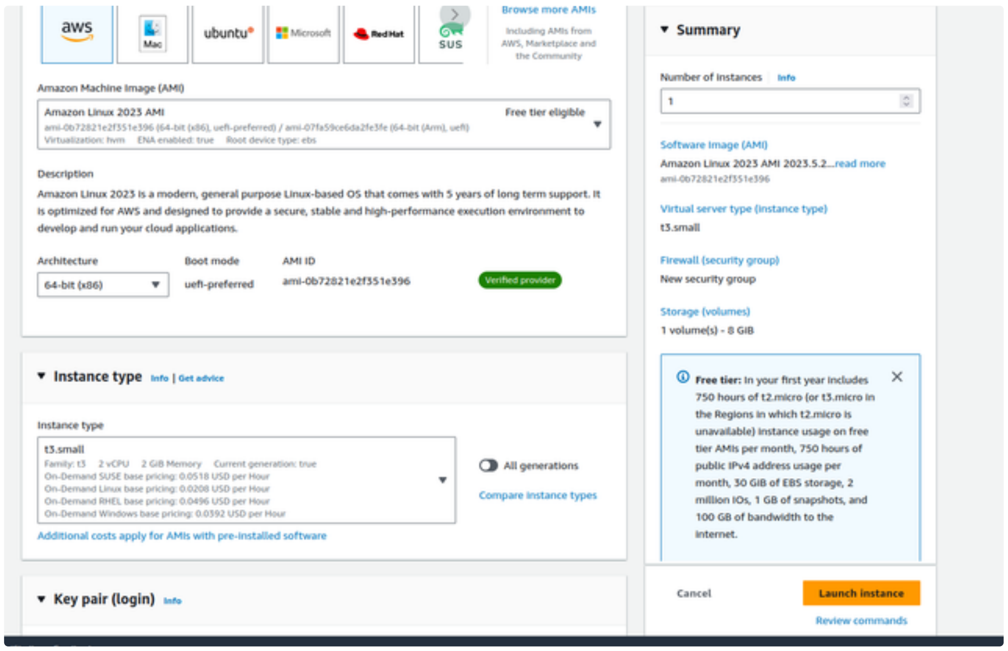
Setting up Prometheus, Node-Exporter with Grafana on EC2 for Metric checking

Setting up EC2

- Go to the AWS Management Console and launch one EC2 instance (ideally need one for Prometheus, one for Grafana, and one or more for Node Exporter).
- Choose an appropriate instance type (e.g., t2.micro for small setups, might need more RAM as did struggle or add swap memory).



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0b72821e2f351e396 (64-bit (x86), uefi-preferred) / ami-07fa59ce6da2fe3fe (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture: 64-bit (x86)
Boot mode: uefi-preferred
AMI ID: ami-0b72821e2f351e396
Verified provider

▼ Instance type Info | Get advice

Instance type

t3.small
Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0518 USD per Hour
On-Demand Linux base pricing: 0.0208 USD per Hour
On-Demand RHEL base pricing: 0.0496 USD per Hour
On-Demand Windows base pricing: 0.0392 USD per Hour

All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

▼ Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-0b72821e2f351e396

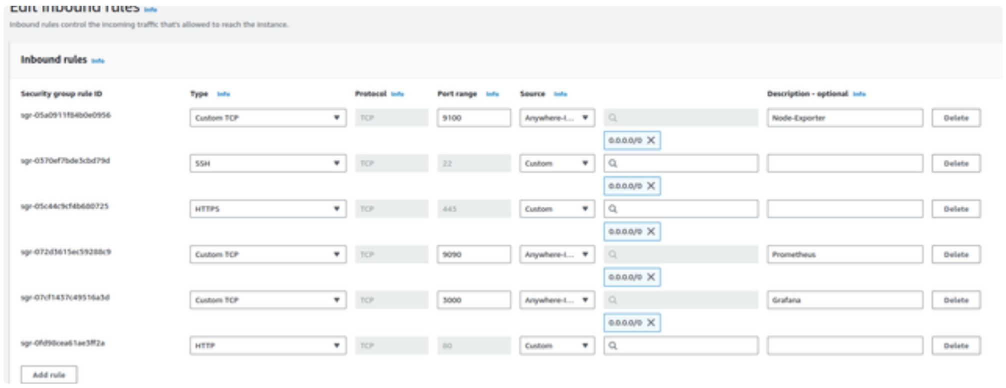
Virtual server type (instance type)
t3.small

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel
Launch instance
Review commands

- Use Amazon Linux 2 AMI or any preferred Linux distribution.
- Ensure security groups allow access on ports 9090 (Prometheus), 3000 (Grafana), and 9100 (Node Exporter) with the CIDR block 0.0.0.0/0.
- 

Edit inbound rules Info

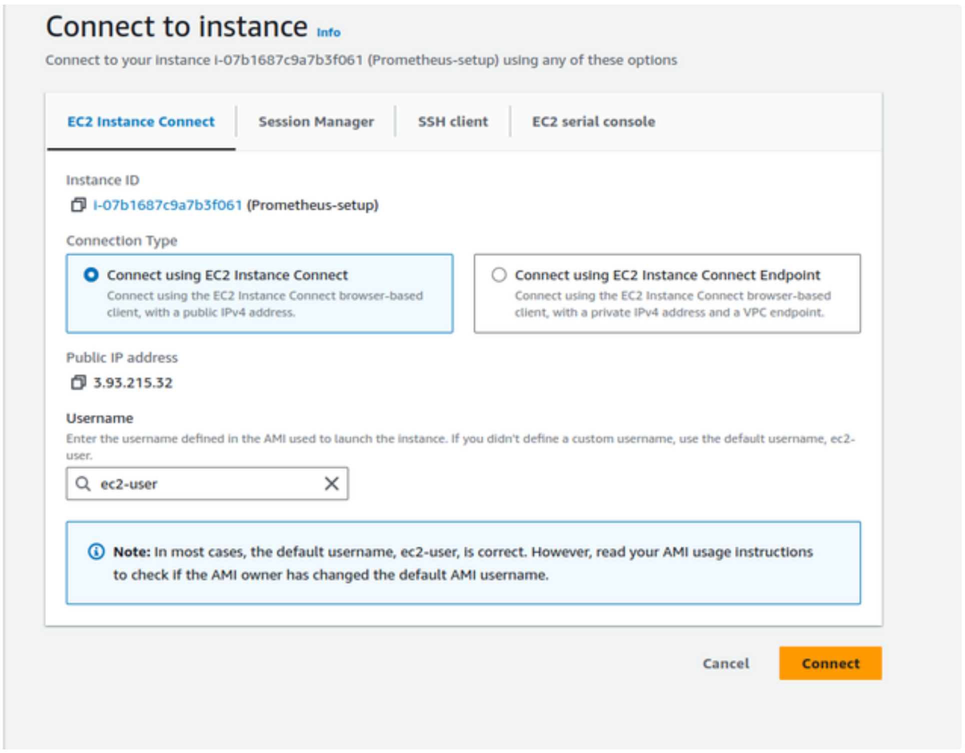
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-05a0911f84b0e0956	Custom TCP	TCP	9100	Anywhere-I...	0.0.0.0/0	Node-Exporter
sgr-0370ef7bde3cbd79d	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-05c44c9cf4b680725	HTTPS	TCP	443	Custom	0.0.0.0/0	
sgr-072d3615ec59288c9	Custom TCP	TCP	9090	Anywhere-I...	0.0.0.0/0	Prometheus
sgr-07cf1437c49516a3d	Custom TCP	TCP	3000	Anywhere-I...	0.0.0.0/0	Grafana
sgr-0fd90cea61ae3ff2a	HTTP	TCP	80	Custom		

Add rule

- SSH into Instance:
  - SSH into instance using the key pair you specified during the setup or use instance connect.
  -



Connect to instance Info

Connect to your instance i-07b1687c9a7b3f061 (Prometheus-setup) using any of these options

Instance ID

i-07b1687c9a7b3f061 (Prometheus-setup)

Connection Type

[✓ Connect using EC2 Instance Connect] Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint: Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

3.93.215.32

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

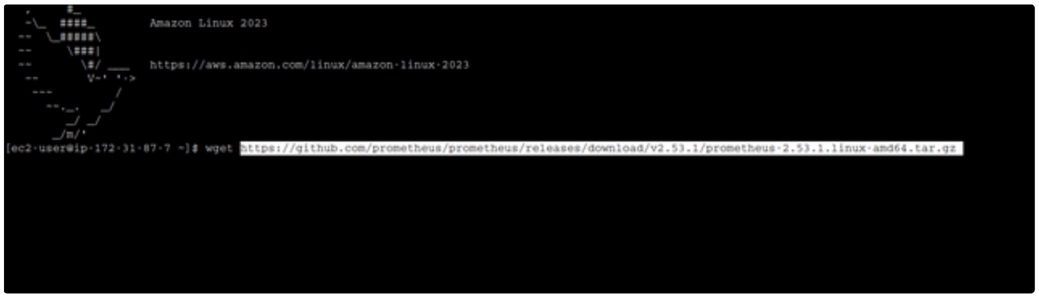
Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

-

Install and Set up Prometheus:

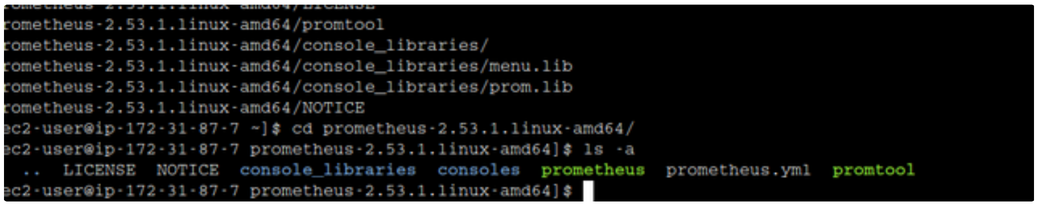
if on Amazon Linux then wget already installed. Please download the latest version , at the time of writing the latest is 2.53.1

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



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-87-7 ~]$ wget https://github.com/prometheus/prometheus/releases/download/v2.53.1/prometheus-2.53.1.linux-amd64.tar.gz
```

```
cd prometheus-2.53.1.linux-amd64
```



```
rometheus-2.53.1.linux-amd64/promtool
rometheus-2.53.1.linux-amd64/console_libraries/
rometheus-2.53.1.linux-amd64/console_libraries/menu.lib
rometheus-2.53.1.linux-amd64/console_libraries/prom.lib
rometheus-2.53.1.linux-amd64/NOTICE
ec2-user@ip-172-31-87-7 ~]$ cd prometheus-2.53.1.linux-amd64/
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ ls -a
 ..  LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$
```

Let’s start by creating a new user other than the root user. And making the Prometheus directory under /etc/ and /var/lib/ directory.

```
#This command is to create user prometheus without the home directory
sudo useradd --no-create-home --shell /bin/false prometheus
#Making prometheus directory under etc
sudo mkdir /etc/prometheus
#Making prometheus directory under lib
sudo mkdir /var/lib/prometheus
```

The following commands will be used to install the Prometheus toolkit in our system *(command is explained using a comment)*:-

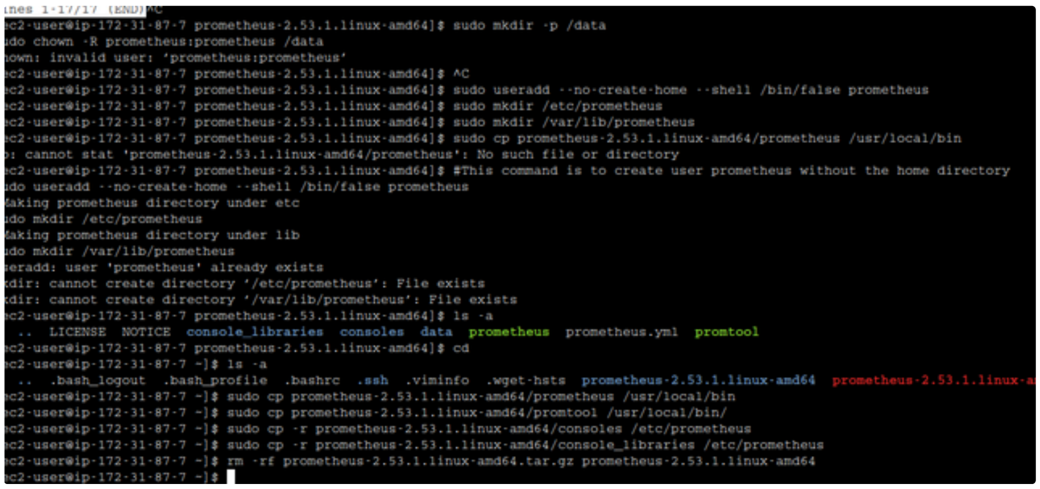
```
#Copy the Prometheus binary to /usr/local/bin
sudo cp prometheus-2.53.1.linux-amd64/prometheus /usr/local/bin
```

```
#Copy promtool to /usr/local/bin/
sudo cp prometheus-2.53.1.linux-amd64/promtool /usr/local/bin/
```

```
# Copy the consoles directory to /etc/prometheus
sudo cp -r prometheus-2.53.1.linux-amd64/consoles /etc/prometheus
```

```
#Copy the 'console_libraries' directory to /etc/prometheus
sudo cp -r prometheus-2.53.1.linux-amd64/console_libraries /etc/prometheus
```

```
#Remove the Prometheus archive and extracted directory
rm -rf prometheus-2.53.1.linux-amd64.tar.gz prometheus-2.53.1.linux-amd64
```



```
ines 1-17/17 (END)^C
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ sudo mkdir -p /data
udo chown -R prometheus:prometheus /data
hown: invalid user: ‘prometheus:prometheus’
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ ^C
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ sudo useradd --no-create-home --shell /bin/false prometheus
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ sudo mkdir /etc/prometheus
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ sudo mkdir /var/lib/prometheus
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ sudo cp prometheus-2.53.1.linux-amd64/prometheus /usr/local/bin
p: cannot stat 'prometheus-2.53.1.linux-amd64/prometheus': No such file or directory
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ #This command is to create user prometheus without the home directory
udo useradd --no-create-home --shell /bin/false prometheus
Making prometheus directory under etc
udo mkdir /etc/prometheus
Making prometheus directory under lib
udo mkdir /var/lib/prometheus
seradd: user 'prometheus' already exists
kdir: cannot create directory ‘/etc/prometheus’: File exists
kdir: cannot create directory ‘/var/lib/prometheus’: File exists
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ ls -a
 ..  LICENSE  NOTICE  console_libraries  consoles  data  prometheus  prometheus.yml  promtool
ec2-user@ip-172-31-87-7 prometheus-2.53.1.linux-amd64]$ cd
ec2-user@ip-172-31-87-7 ~]$ ls -a
 ..  .bash_logout  .bash_profile  .bashrc  .ssh  .viminfo  .wget-hsts  prometheus-2.53.1.linux-amd64  prometheus-2.53.1.linux-a
ec2-user@ip-172-31-87-7 ~]$ sudo cp prometheus-2.53.1.linux-amd64/prometheus /usr/local/bin
ec2-user@ip-172-31-87-7 ~]$ sudo cp prometheus-2.53.1.linux-amd64/promtool /usr/local/bin/
ec2-user@ip-172-31-87-7 ~]$ sudo cp -r prometheus-2.53.1.linux-amd64/consoles /etc/prometheus
ec2-user@ip-172-31-87-7 ~]$ sudo cp -r prometheus-2.53.1.linux-amd64/console_libraries /etc/prometheus
ec2-user@ip-172-31-87-7 ~]$ rm -rf prometheus-2.53.1.linux-amd64.tar.gz prometheus-2.53.1.linux-amd64
ec2-user@ip-172-31-87-7 ~]$
```

Now we need to add a **prometheus.yml** file that will have the configuration file for the prometheus.

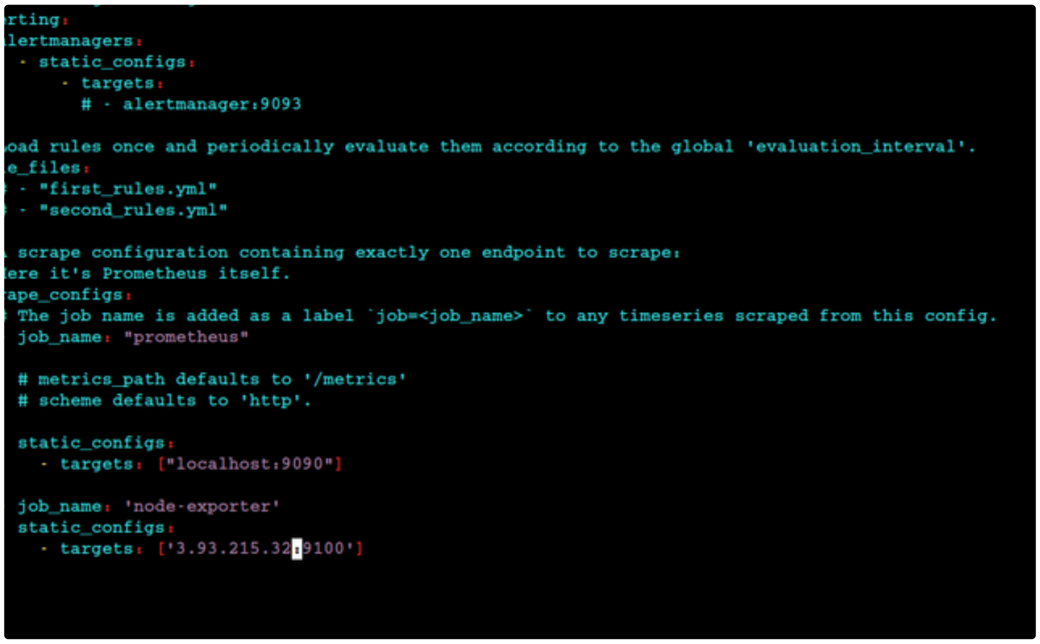
We are using –no-create-home user because we are going to use the ***prometheus*** user for a specific purpose, i.e., for using it as a service. We do not want it to interact with the user for any kind of login or personal use. So it is better to use this than using the root user for everything.

Configure Prometheus:

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

```
Edit the prometheus.yml file to include the Node Exporter instances:

yaml

global:
  scrape_interval: 15s

```

```
scrape_configs:
  - job_name: 'node-exporter'
    static_configs:
      - targets: ['<NODE_EXPORTER_INSTANCE_IP>:9100']
```

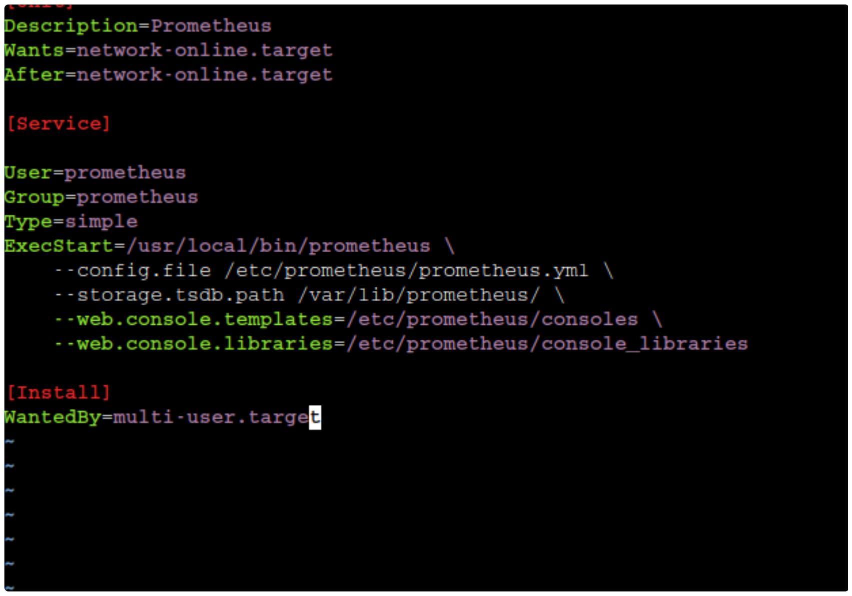


```
rting:
lertmanagers:
  - static_configs:
      - targets:
        # - alertmanager:9093

oad rules once and periodically evaluate them according to the global 'evaluation_interval'.
e_files:
  - "first_rules.yml"
  - "second_rules.yml"

 scrape configuration containing exactly one endpoint to scrape:
ere it's Prometheus itself.
ape_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:9090"]

 job_name: 'node-exporter'
 static_configs:
   - targets: ['3.93.215.32:9100']
```

Let’s see what this configuration is :

- Global Configuration(global)
- scrape_interval: This sets the interval at which Prometheus scrapes targets for metrics. In this case, it is set to 15 seconds (15s).
- external_labels: These are labels that are applied to all scraped metrics. In this case, there is one label named monitor with the value ‘prometheus’.
- Scrape Configurations (scrape_configs):
- – job_name: ‘prometheus’: This is the name of the job. In this case, it is named ‘prometheus’.
- static_configs: Defines a list of targets to scrape metrics from.
- – targets: [‘localhost:9090’]: This specifies that Prometheus should scrape metrics from the target, where presumably the Prometheus server itself is running.

Now that our configuration of Prometheus is done, we will have to add a service file that will define how our Prometheus service will be managed by systemd.



```
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
~
~
~
~
~
~
~
```

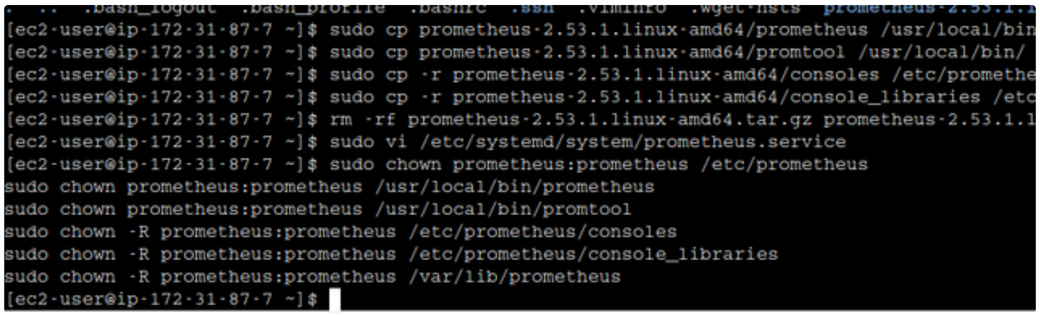
Let’s understand what we are doing here:

1. Unit Section ([Unit]):
    - Description: is a human-readable description of the service, indicating that this unit is for Prometheus.
    - Wants: Indicates a dependency on network-online.target, meaning that this service should not start until network services are available.
    - After: Specifies that this service should start after network-online.target.
2. Service Section ([Service]):
    - User=prometheus: Specifies the user under which the Prometheus service should run. This is likely the user you created earlier for Prometheus.
    - Group=prometheus: Specifies the group under which the Prometheus service should run.
    - Type=simple: Indicates that the service type is a simple process that does not fork.
    - ExecStart: Specifies the command to start the Prometheus service. It includes the path to the prometheus binary and various command-line options:
        - –config.file: Specifies the path to the Prometheus configuration file.
        - –storage.tsdb.path: Specifies the path to the directory where Prometheus should store its time-series database.
        - –web.console.templates and –web.console.libraries: Specify the paths for the web console templates and libraries.
3. Install Section ([Install]):

- WantedBy=multi-user.target: Specifies that this service is wanted by the multi-user.target, meaning it will be started as part of the multi-user system startup.

Finally, we will change the file permission we just created:-

```
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
```

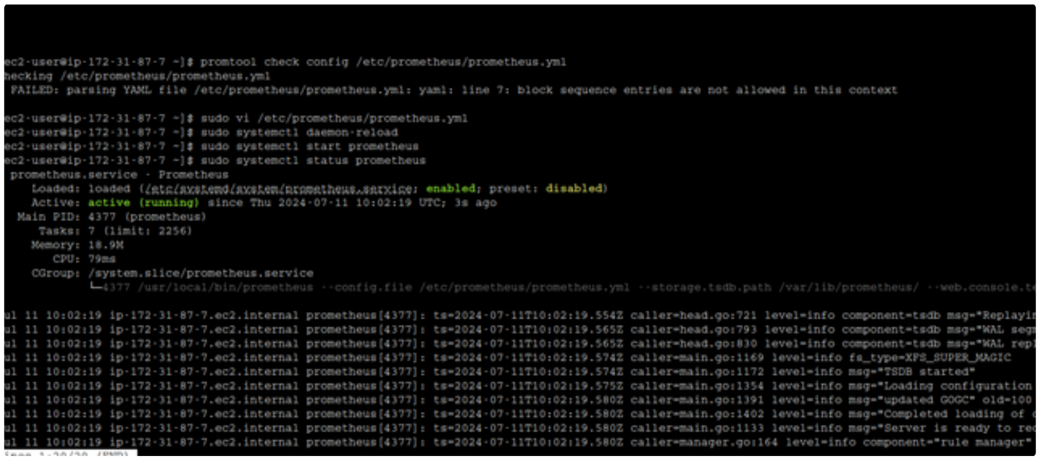


```
.  ..  .bash_logout  .bash_profile  .bashrc  .ssh  .viminfo  .wget-hsts  prometheus-2.53.1.l
[ec2-user@ip-172-31-87-7 ~]$ sudo cp prometheus-2.53.1.linux-amd64/prometheus /usr/local/bin
[ec2-user@ip-172-31-87-7 ~]$ sudo cp prometheus-2.53.1.linux-amd64/promtool /usr/local/bin/
[ec2-user@ip-172-31-87-7 ~]$ sudo cp -r prometheus-2.53.1.linux-amd64/consoles /etc/promethe
[ec2-user@ip-172-31-87-7 ~]$ sudo cp -r prometheus-2.53.1.linux-amd64/console_libraries /etc
[ec2-user@ip-172-31-87-7 ~]$ rm -rf prometheus-2.53.1.linux-amd64.tar.gz prometheus-2.53.1.l
[ec2-user@ip-172-31-87-7 ~]$ sudo vi /etc/systemd/system/prometheus.service
[ec2-user@ip-172-31-87-7 ~]$ 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
[ec2-user@ip-172-31-87-7 ~]$
```

Next, is to reload system configuration, so that it could take the new configuration that we added. Then we will enable the Prometheus service to start at the boot. Lastly, we will start the service.

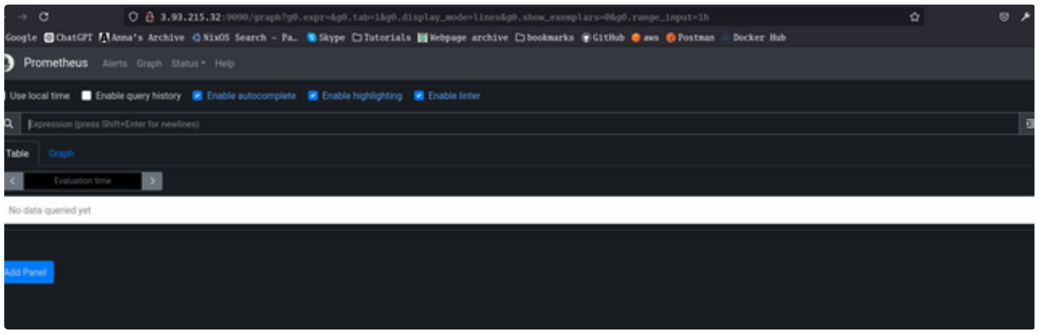
To check the status of the service is running or not, use

```
sudo systemctl status prometheus
```



```
ec2-user@ip-172-31-87-7 ~]$ promtool check config /etc/prometheus/prometheus.yml
hecking /etc/prometheus/prometheus.yml
 FAILED: parsing YAML file /etc/prometheus/prometheus.yml: yaml: line 7: block sequence entries are not allowed in this context

ec2-user@ip-172-31-87-7 ~]$ sudo vi /etc/prometheus/prometheus.yml
ec2-user@ip-172-31-87-7 ~]$ sudo systemctl daemon-reload
ec2-user@ip-172-31-87-7 ~]$ sudo systemctl start prometheus
ec2-user@ip-172-31-87-7 ~]$ sudo systemctl status prometheus
 prometheus.service - Prometheus
    Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: disabled)
    Active: active (running) since Thu 2024-07-11 10:02:19 UTC; 3s ago
  Main PID: 4377 (prometheus)
     Tasks: 7 (limit: 2256)
    Memory: 18.9M
       CPU: 79ms
    CGroup: /system.slice/prometheus.service
            └─4377 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.te

ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.554Z caller=head.go:721 level=info component=tsdb msg="Replayin
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.565Z caller=head.go:793 level=info component=tsdb msg="WAL segm
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.565Z caller=head.go:830 level=info component=tsdb msg="WAL repl
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.574Z caller=main.go:1169 level=info fs_type=XFS_SUPER_MAGIC
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.574Z caller=main.go:1172 level=info msg="TSDB started"
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.575Z caller=main.go:1354 level=info msg="Loading configuration
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.580Z caller=main.go:1391 level=info msg="updated GOGC" old=100
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.580Z caller=main.go:1402 level=info msg="Completed loading of c
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.580Z caller=main.go:1133 level=info msg="Server is ready to rec
ul 11 10:02:19 ip-172-31-87-7.ec2.internal prometheus[4377]: ts=2024-07-11T10:02:19.580Z caller=manager.go:164 level=info component="rule manager"
```

Just go to the new tab and enter <your-public-ip>:9090.



```
3.93.215.32:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h
```

Use local time [ ] Enable query history [✓ Enable autocomplete] [✓ Enable highlighting] [✓ Enable linter]

No data queried yet

Installing and Setting up Node-Exporter on EC2

Setting up Node-Exporter is similar to setting up Prometheus. If you have not yet read the blog on setting up Prometheus, we would highly recommend you go check it out.

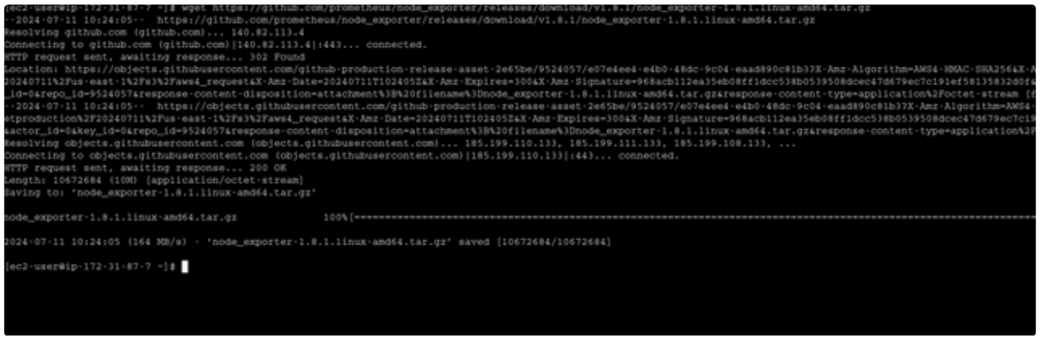
Follow these steps to set up Node-Exporter on EC2:

1. For this, we will first create a user:

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

2. Then we have to install node-exporter and go to Downloads for the latest node-exporter link.

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.3.1.linux-amd64.tar.gz

```



```
[ec2-user@ip-172-31-87-7 ~]$ wget https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz
--2024-07-11 10:24:05--  https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/e07e4ee4-e4b0-48dc-9c04-eaad890c81b3?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-A
20240711%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240711T102405Z&X-Amz-Expires=300&X-Amz-Signature=968acb112ea35eb08ff1dcc538b0539508dcec47d679ec7c191ef58135832d0f&
_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.8.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [f
--2024-07-11 10:24:05--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/e07e4ee4-e4b0-48dc-9c04-eaad890c81b3?X-Amz-Algorithm=AWS4-
etproduction%2F20240711%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240711T102405Z&X-Amz-Expires=300&X-Amz-Signature=968acb112ea35eb08ff1dcc538b0539508dcec47d679ec7c19
&actor_id=0&key_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.8.1.linux-amd64.tar.gz&response-content-type=application%2F
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10672684 (10M) [application/octet-stream]
Saving to: 'node_exporter-1.8.1.linux-amd64.tar.gz'

node_exporter-1.8.1.linux-amd64.tar.gz            100%[===============================================================================================

2024-07-11 10:24:05 (164 MB/s) - 'node_exporter-1.8.1.linux-amd64.tar.gz' saved [10672684/10672684]

[ec2-user@ip-172-31-87-7 ~]$ █
```

3. We will then untar the downloaded file and copy the node_exporter binary to the /usr/local/bin directory.

```
tar xzf node_exporter-1.8.1.linux-amd64.tar.gz
sudo mkdir /usr/local/bin/node_exporter
sudo cp node_exporter-1.8.1.linux-amd64/node_exporter /usr/local/bin/node_exporter
rm -rf node_exporter-1.8.1.linux-amd64.tar.gz node_exporter-1.8.1.linux-amd64
```

4. Now that we have installed it, we will configure the service.

```
sudo vi /etc/systemd/system/node-exporter.service
```

5. We will now add configuration to the service file.

```
[Unit]
Description=Prometheus Node Exporter Service
After=network.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

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

6. Now we have to configure systems.

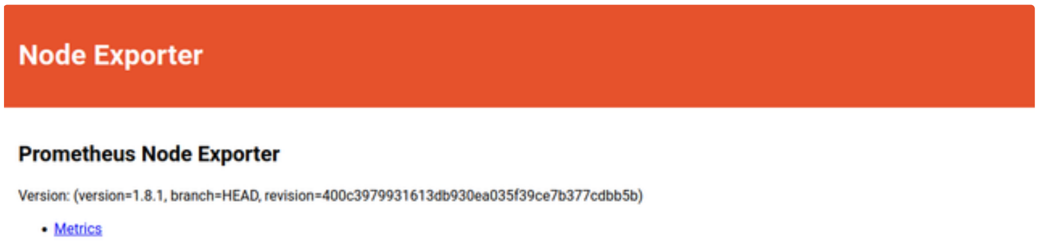
```
#This will reload the daemon to add the newly made configuration
sudo systemctl daemon-reload
#This will enable node exporter
sudo systemctl enable node-exporter
#This starts the node-exporter
sudo systemctl start node-exporter
```

```
#This will check the status
sudo systemctl status node-exporter
```

Might have to, if errors

```
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
sudo chmod 755 /usr/local/bin/node_exporter
```

This should start the node_exporter server, which you can see by using <your-public-ip>:9100 and it would look something like this:



Node Exporter

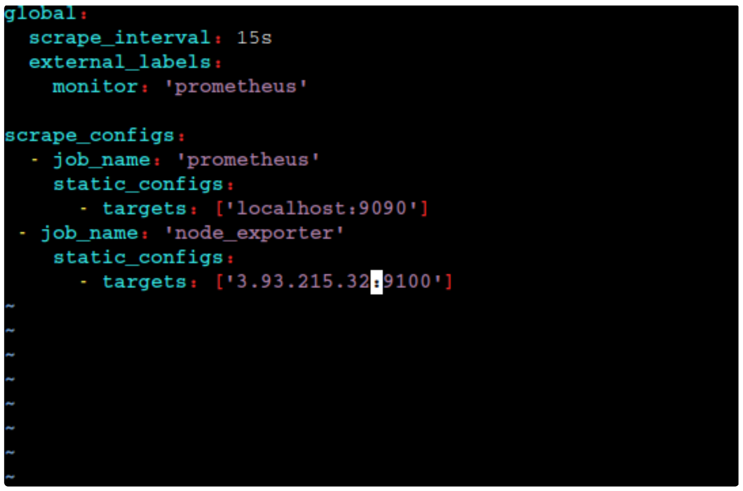
Prometheus Node Exporter

Version: (version=1.8.1, branch=HEAD, revision=400c3979931613db930ea035f39ce7b377cdbb5b)

- Metrics

Once the server is running, we now have to make a few adjustments in our prometheus.yml file to fetch the node-exporter server and to do that, add these configurations in the prometheus.yml file.adding your ip address

`sudo vi /etc/prometheus/prometheus.yml`



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

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
 - job_name: 'node_exporter'
    static_configs:
      - targets: ['3.93.215.32:9100']
~
~
~
~
~
~
~
~
```

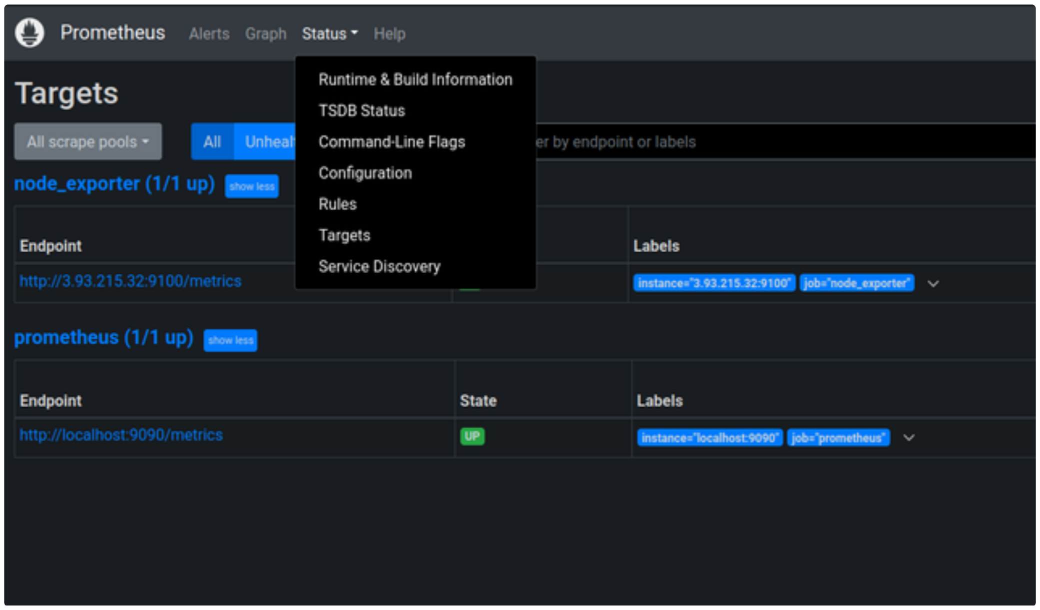
After updating the configuration, restart your daemon using

```
sudo systemctl daemon-reload
```

then

```
sudo systemctl restart prometheus.
```

Then go to the Prometheus dashboard and navigate to Target under the status drop-down (top-left).



Targets

node_exporter (1/1 up)

Endpoint	Labels
http://3.93.215.32:9100/metrics	instance="3.93.215.32:9100" job="node_exporter"

prometheus (1/1 up)

Endpoint	State	Labels
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"

Install and Setup Grafana:

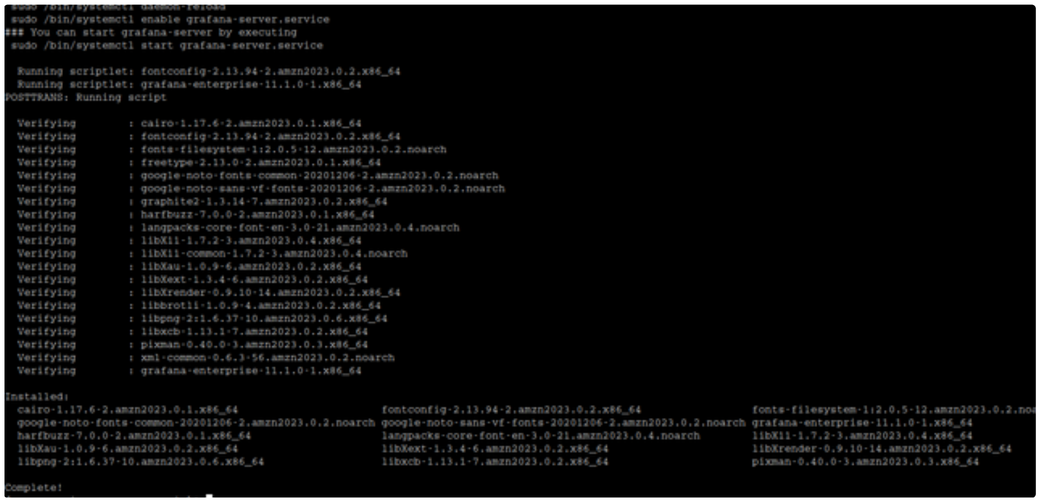
Installing Grafana on EC2 is comparatively easier. Let's look at the steps on how to easily set up Grafana on EC2 (Amazon Linux):

First, we need to update the packages.

```
sudo yum update -y
```

Then, we are importing standalone binaries, (not recommended as the pacake will not automatically update, the recommended way is adding repositories).

```
sudo yum install -y https://dl.grafana.com/enterprise/release/grafana-enterprise-11.1.0-1.x86_64.rpm
```

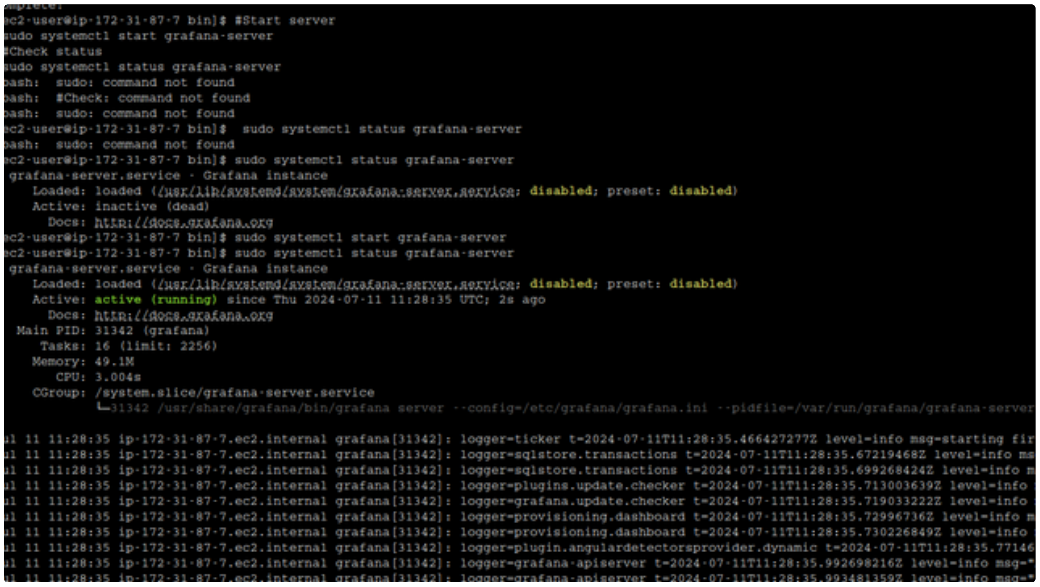


```
 sudo /bin/systemctl daemon-reload
 sudo /bin/systemctl enable grafana-server.service
### You can start grafana-server by executing
 sudo /bin/systemctl start grafana-server.service

  Running scriptlet: fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  Running scriptlet: grafana-enterprise-11.1.0-1.x86_64
POSTTRANS: Running script

  Verifying        : cairo-1.17.6-2.amzn2023.0.1.x86_64
  Verifying        : fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  Verifying        : fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
  Verifying        : freetype-2.13.0-2.amzn2023.0.1.x86_64
  Verifying        : google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
  Verifying        : google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  Verifying        : graphite2-1.3.14-7.amzn2023.0.2.x86_64
  Verifying        : harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  Verifying        : langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
  Verifying        : libX11-1.7.2-3.amzn2023.0.4.x86_64
  Verifying        : libX11-common-1.7.2-3.amzn2023.0.4.noarch
  Verifying        : libXau-1.0.9-6.amzn2023.0.2.x86_64
  Verifying        : libXext-1.3.4-6.amzn2023.0.2.x86_64
  Verifying        : libXrender-0.9.10-14.amzn2023.0.2.x86_64
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  Verifying        : libpng-2:1.6.37-10.amzn2023.0.6.x86_64
  Verifying        : libxcb-1.13.1-7.amzn2023.0.2.x86_64
  Verifying        : pixman-0.40.0-3.amzn2023.0.3.x86_64
  Verifying        : xml-common-0.6.3-56.amzn2023.0.2.noarch
  Verifying        : grafana-enterprise-11.1.0-1.x86_64

Installed:
  cairo-1.17.6-2.amzn2023.0.1.x86_64                   fontconfig-2.13.94-2.amzn2023.0.2.x86_64                   fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noa
  google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch grafana-enterprise-11.1.0-1.x86_64
  harfbuzz-7.0.0-2.amzn2023.0.1.x86_64                 langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch         libX11-1.7.2-3.amzn2023.0.4.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64                   libXext-1.3.4-6.amzn2023.0.2.x86_64                       libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64               libxcb-1.13.1-7.amzn2023.0.2.x86_64                       pixman-0.40.0-3.amzn2023.0.3.x86_64

Complete!
```

Now that we have installed Grafana in our instance, it’s time to start the Grafana server.

Use the command to start the grafana-server.

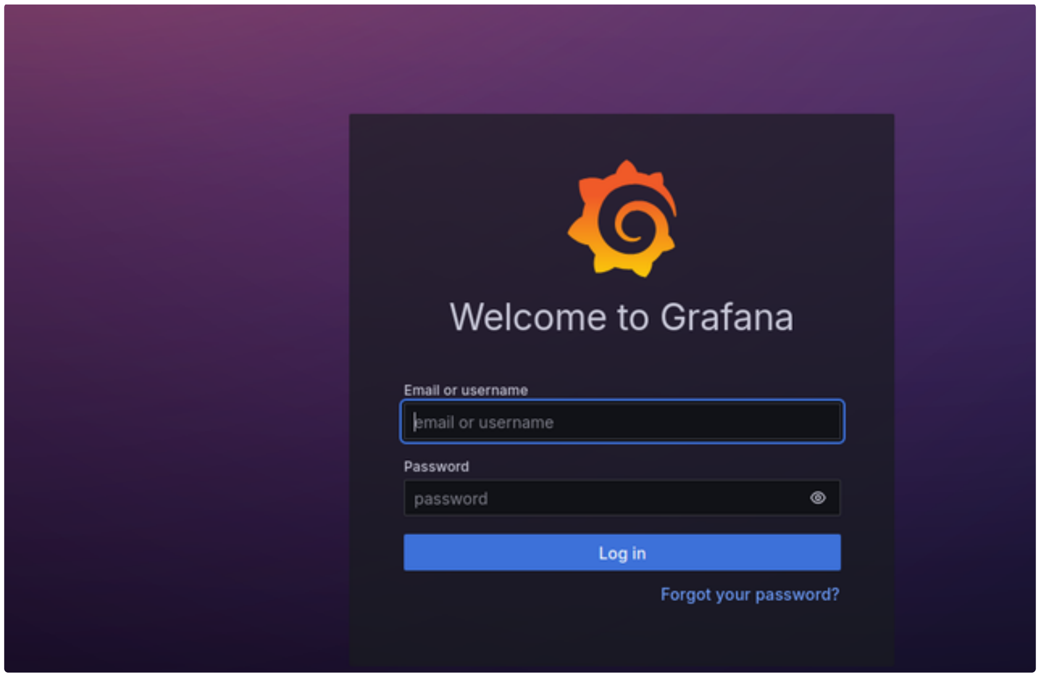
```
#Start server
 sudo systemctl start grafana-server
 #Check status
 sudo systemctl status grafana-server
```



```
ec2-user@ip-172-31-87-7 bin]$ #Start server
sudo systemctl start grafana-server
#Check status
sudo systemctl status grafana-server
bash:  sudo: command not found
bash:  #Check: command not found
bash:  sudo: command not found
ec2-user@ip-172-31-87-7 bin]$  sudo systemctl status grafana-server
bash:  sudo: command not found
ec2-user@ip-172-31-87-7 bin]$ sudo systemctl status grafana-server
 grafana-server.service - Grafana instance
    Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; preset: disabled)
    Active: inactive (dead)
      Docs: http://docs.grafana.org
ec2-user@ip-172-31-87-7 bin]$ sudo systemctl start grafana-server
ec2-user@ip-172-31-87-7 bin]$ sudo systemctl status grafana-server
 grafana-server.service - Grafana instance
    Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; preset: disabled)
    Active: active (running) since Thu 2024-07-11 11:28:35 UTC; 2s ago
      Docs: http://docs.grafana.org
  Main PID: 31342 (grafana)
     Tasks: 16 (limit: 2256)
    Memory: 49.1M
       CPU: 3.004s
    CGroup: /system.slice/grafana-server.service
            └─31342 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/var/run/grafana/grafana-server

ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=ticker t=2024-07-11T11:28:35.466427277Z level=info msg=starting fir
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=sqlstore.transactions t=2024-07-11T11:28:35.67219468Z level=info ms
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=sqlstore.transactions t=2024-07-11T11:28:35.699268424Z level=info m
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=plugins.update.checker t=2024-07-11T11:28:35.713003639Z level=info
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=grafana.update.checker t=2024-07-11T11:28:35.719033222Z level=info
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=provisioning.dashboard t=2024-07-11T11:28:35.72996736Z level=info m
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=provisioning.dashboard t=2024-07-11T11:28:35.730226849Z level=info
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=plugin.angulardetectorsprovider.dynamic t=2024-07-11T11:28:35.77146
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=grafana-apiserver t=2024-07-11T11:28:35.992698216Z level=info msg="
ul 11 11:28:35 ip-172-31-87-7.ec2.internal grafana[31342]: logger=grafana-apiserver t=2024-07-11T11:28:35.993481359Z level=info msg="
```

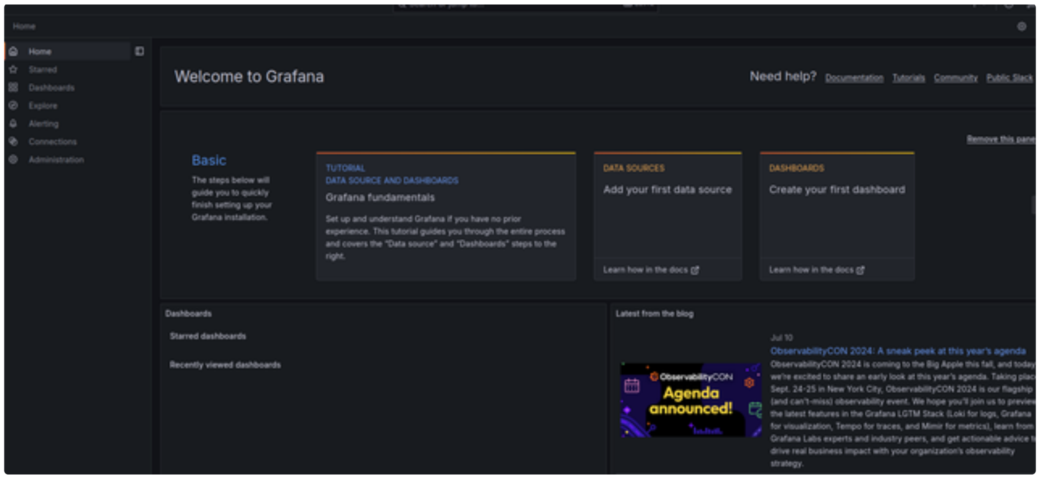
It’s time to open the Grafana server!

To open the Grafana Server, use your IP or Public IPv4 DNS with a 3000 port.



```

```

To log in to Grafana, use *admin* as your username/email and *admin* as your password. It will ask you to create a new password, create a new password, and submit.



The steps below will guide you to quickly finish setting up your Grafana installation.

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

Jul 10
ObservabilityCON 2024: A sneak peek at this year's agenda
ObservabilityCON 2024 is coming to the Big Apple this fall, and today we're excited to share an early look at this year's agenda. Taking plac Sept. 24-25 in New York City, ObservabilityCON 2024 is our flagship (and can't-miss) observability event. We hope you'll join us to previe the latest features in the Grafana LGTM Stack (Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics), learn from Grafana Labs experts and industry peers, and get actionable advice t drive real business impact with your organization's observability strategy.

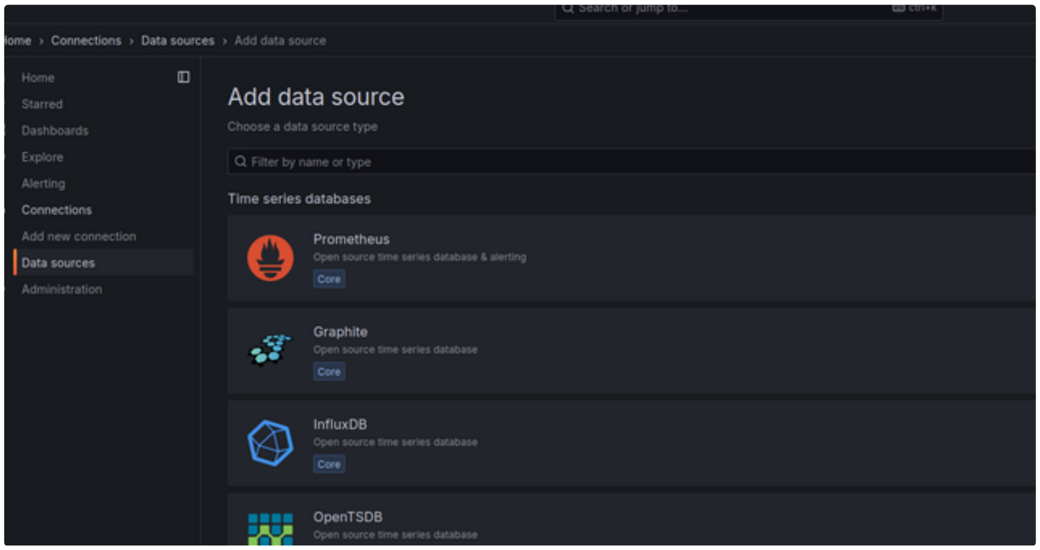
Using Grafana to monitor

We need to visualize the resources that we have created. We will be using Grafana and the Grafana Dashboard to do so. Let's start with how we can monitor using Grafana.

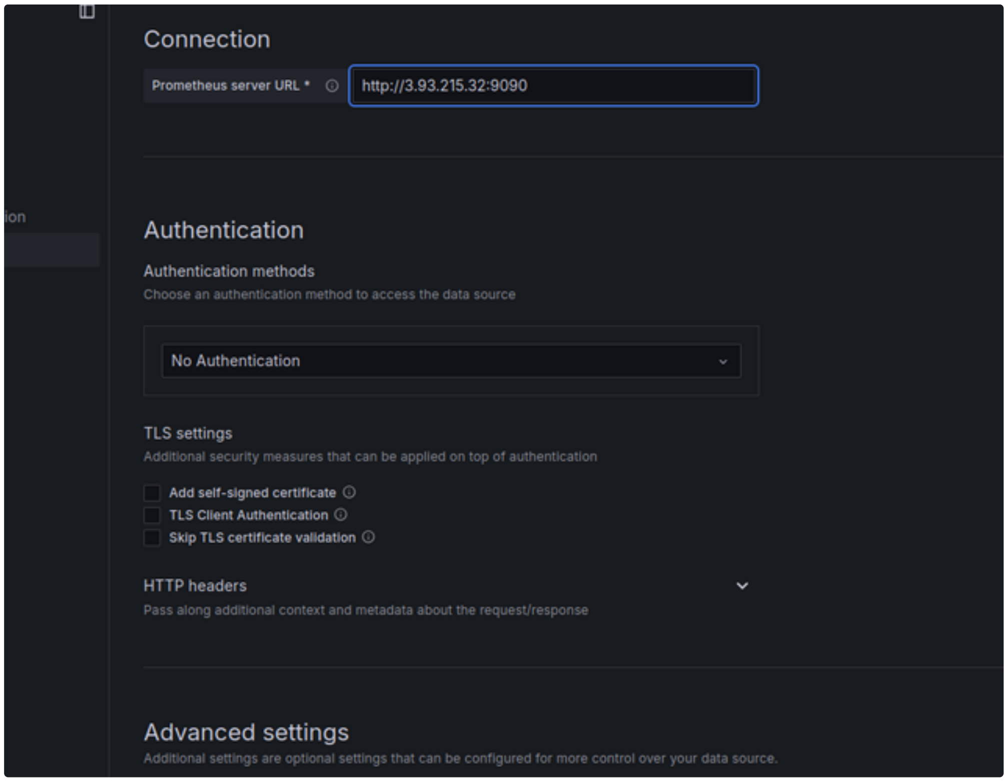
Creating Data Sources

Once you open the page, you will see Data Sources, go to Data Sources.

1. Select Prometheus as the Time Series Databases.



1. In the connection section, we will enter the Prometheus server URL. Since we are working on the same server, we will use http://localhost:9090. After this, go down and click Save and Test.



Connection

Prometheus server URL *: http://3.93.215.32:9090

Authentication

Authentication methods

Choose an authentication method to access the data source

No Authentication

TLS settings

Additional security measures that can be applied on top of authentication

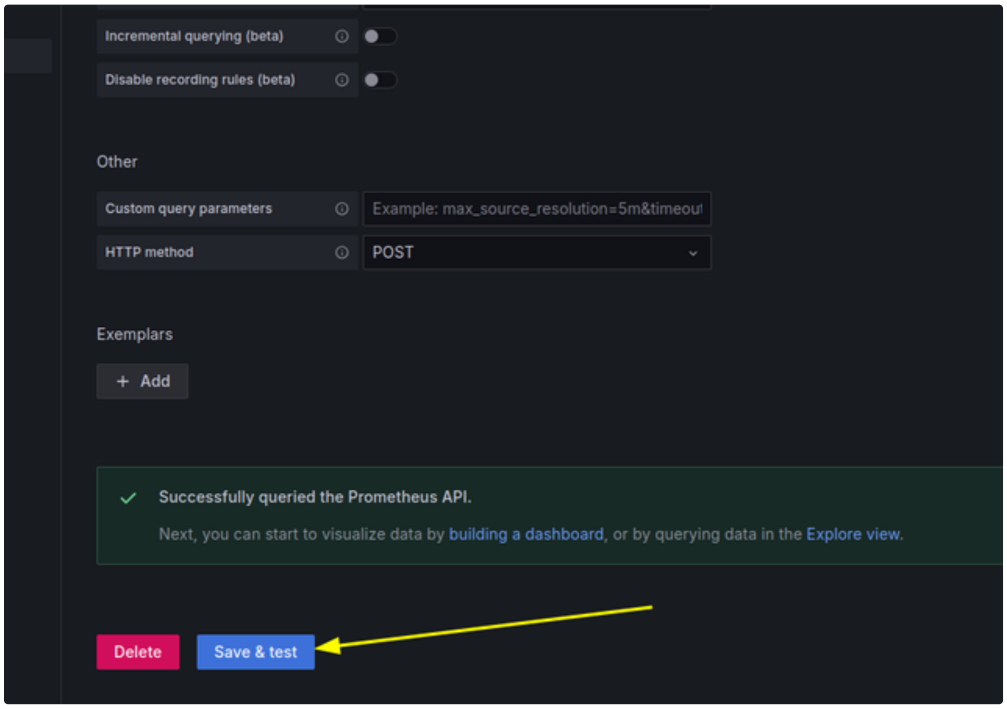
- Add self-signed certificate
- TLS Client Authentication
- Skip TLS certificate validation

HTTP headers

Pass along additional context and metadata about the request/response

Advanced settings

Additional settings are optional settings that can be configured for more control over your data source.



Incremental querying (beta)

Disable recording rules (beta)

Other

Custom query parameters: Example: max_source_resolution=5m&timeout

HTTP method: POST

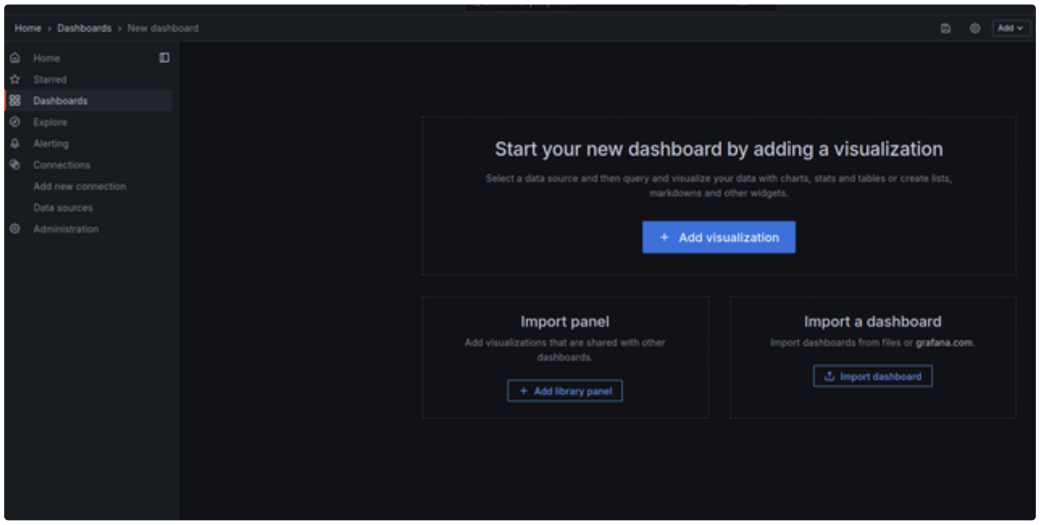
Exemplars

Successfully queried the Prometheus API.

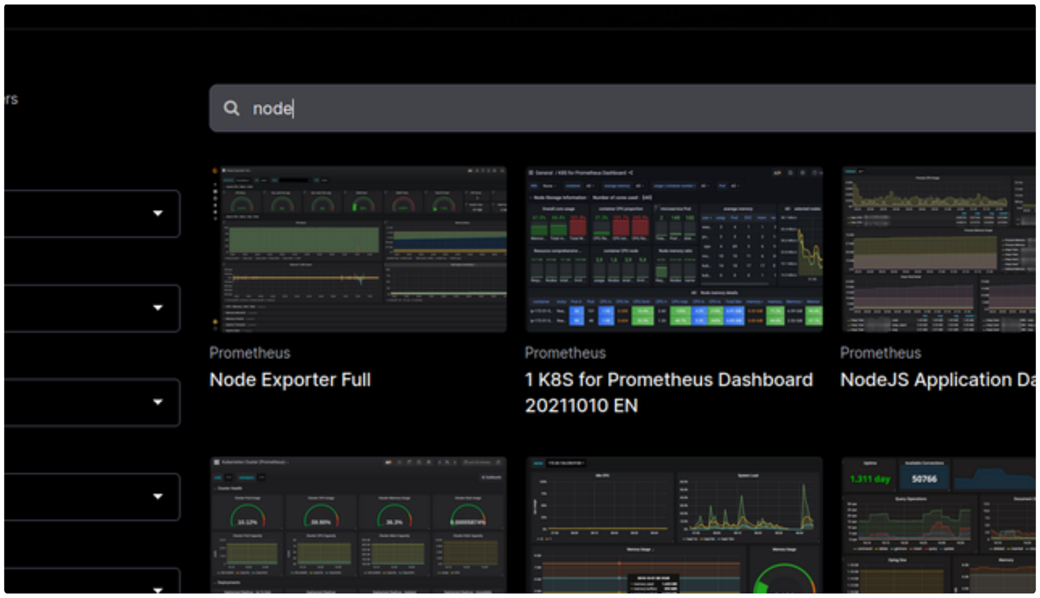
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.

Creating Dashboard

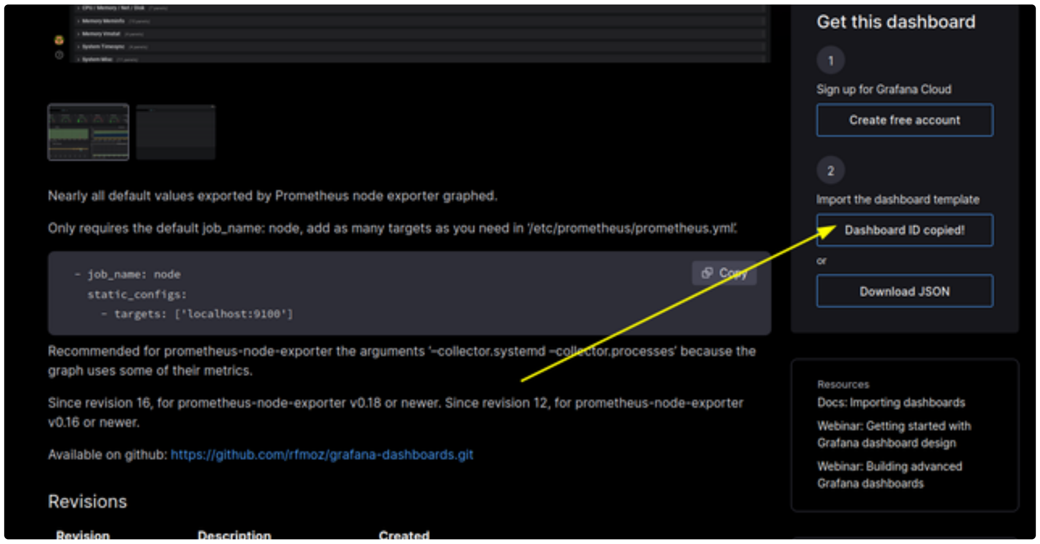
Navigate back to home and go to Dashboards.


Home › Dashboards › New dashboard
Add
Home
Starred
Dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Start your new dashboard by adding a visualization
Select a data source and then query and visualize your data with charts, stats and tables or create lists, markdowns and other widgets.
+ Add visualization
Import panel
Add visualizations that are shared with other dashboards.
+ Add library panel
Import a dashboard
Import dashboards from files or grafana.com.
Import dashboard

Now click on import dashboard; we will be importing the dashboard for our node_exporter from Grafana dashboards | Grafana Labs .Search for Node Exporter Full and open it.



Copy the dashboard ID.



Nearly all default values exported by Prometheus node exporter graphed.

Only requires the default job_name: node, add as many targets as you need in '/etc/prometheus/prometheus.yml'.

```
- job_name: node
  static_configs:
    - targets: ['localhost:9100']
```

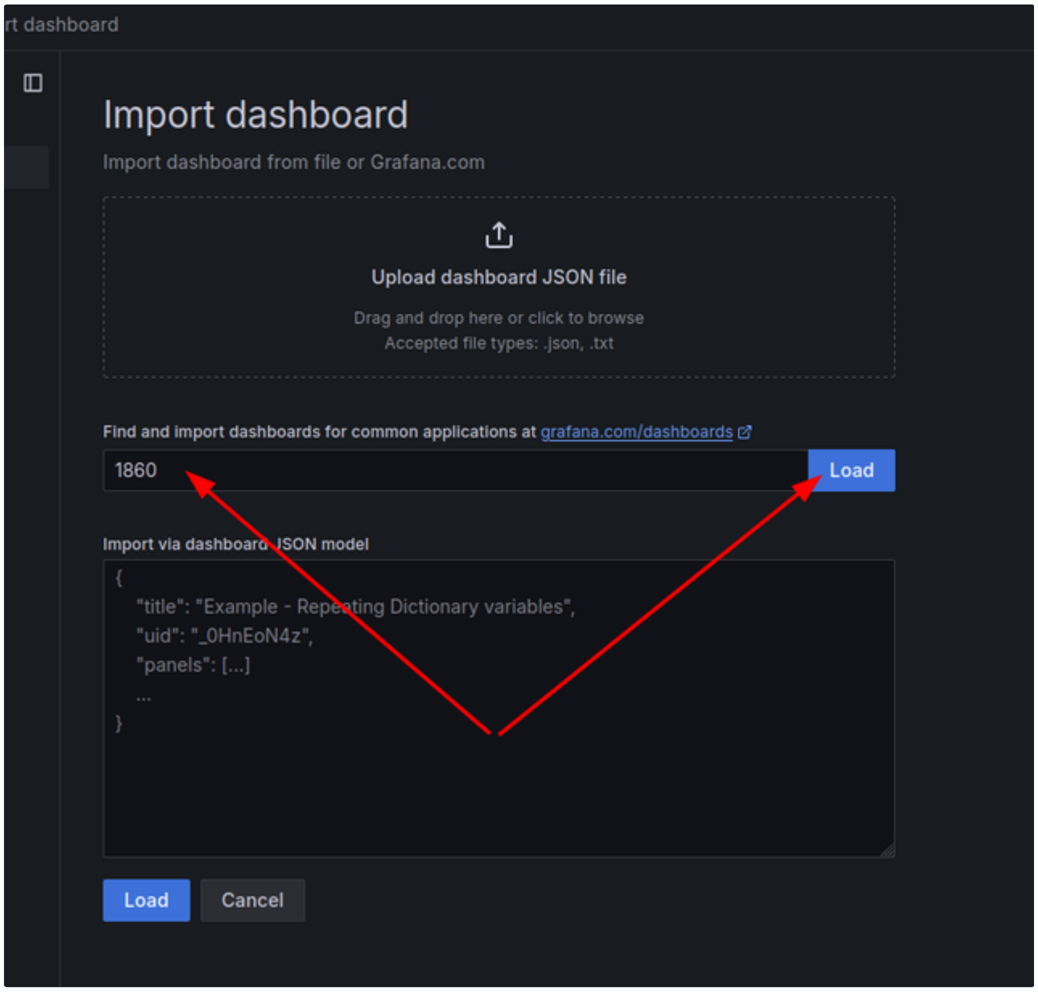
Recommended for prometheus-node-exporter the arguments '--collector.systemd --collector.processes' because the graph uses some of their metrics.

Since revision 16, for prometheus-node-exporter v0.18 or newer. Since revision 12, for prometheus-node-exporter v0.16 or newer.

Available on github: https://github.com/rfmoz/grafana-dashboards.git

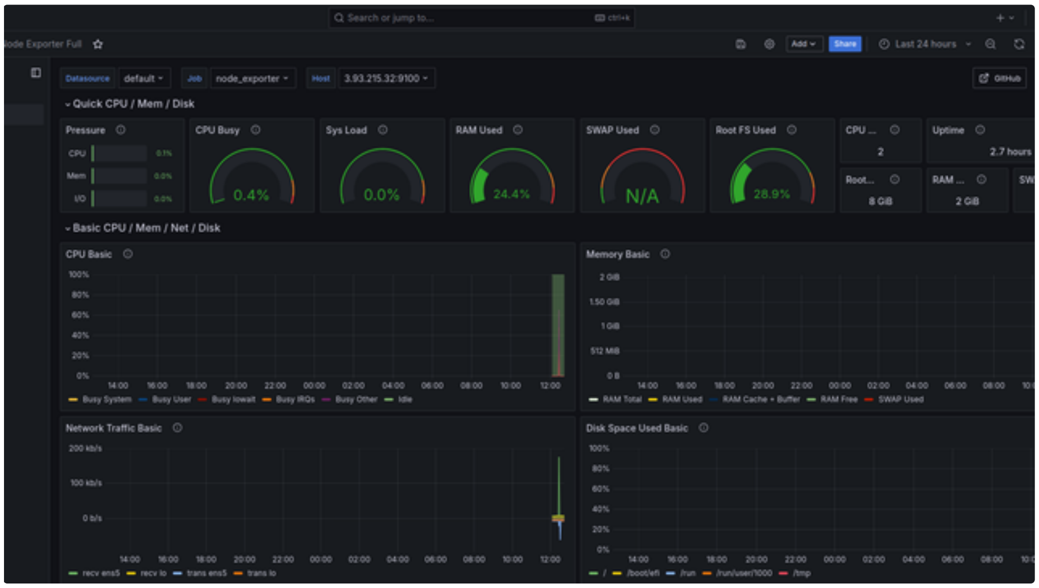
Copy
Revisions
Revision
Description
Created
Get this dashboard
1
Sign up for Grafana Cloud
Create free account
2
Import the dashboard template
Dashboard ID copied!
or
Download JSON
Resources
Docs: Importing dashboards
Webinar: Getting started with Grafana dashboard design
Webinar: Building advanced Grafana dashboards

We will be pasting this Dashboard ID to the Import Dashboard section. Next, select the **Prometheus data source** that we just created. In Grafana, we automatically load a Dashboard and show us the metrics.



rt dashboard
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Find and import dashboards for common applications at grafana.com/dashboards
1860
Load
Import via dashboard JSON model
Load
Cancel

```
{
   "title": "Example - Repeating Dictionary variables",
   "uid": "_0HnEoN4z",
   "panels": [...]
   ...
}
```



We have now set up a monitoring stack with Prometheus, Grafana, and Node Exporter on AWS EC2 instance. This setup allows you to monitor and visualize metrics from your EC2 instances effectively.

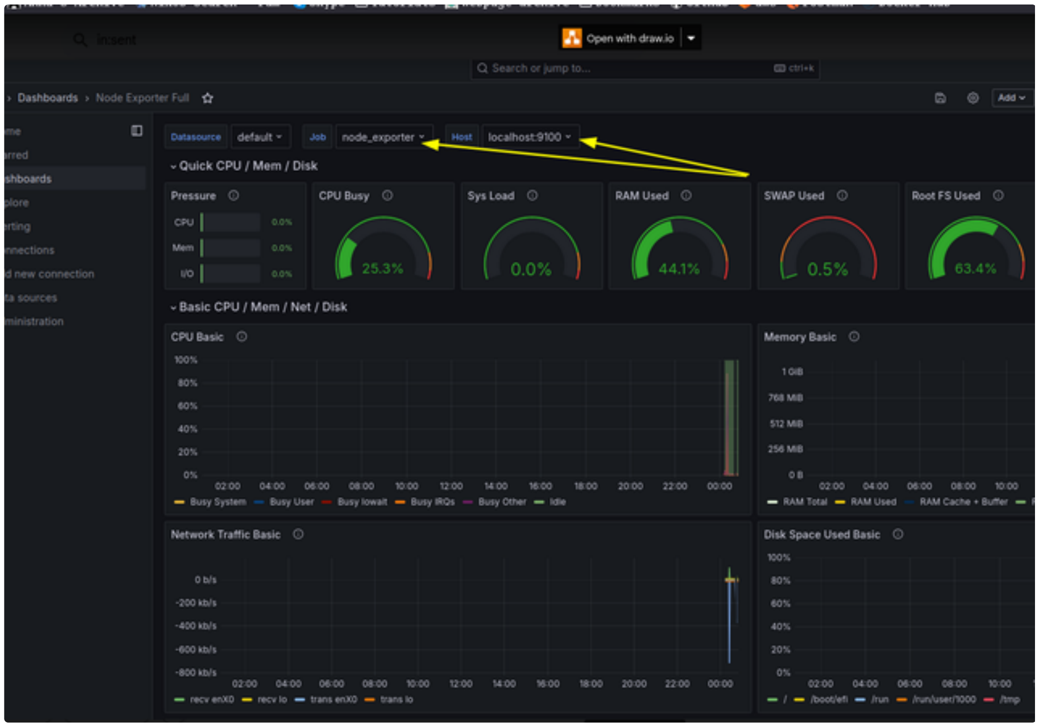
if another IP address to be monitored then it need to added to prometheus.yml file

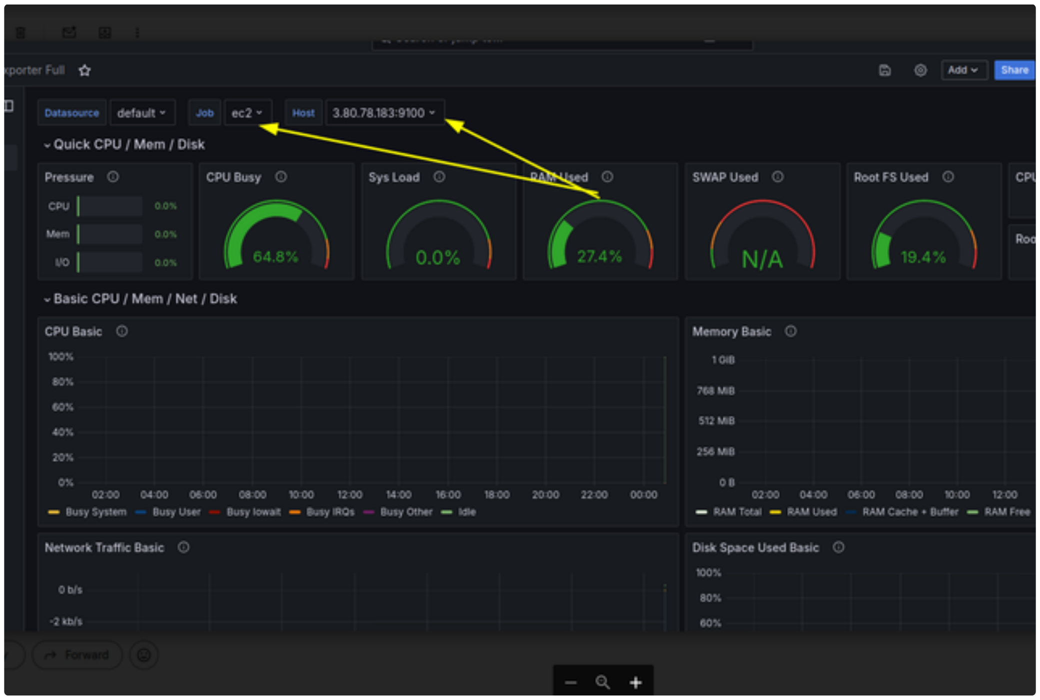
at

```
sudo vi /etc/prometheus/prometheus.yml
```

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

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

  - job_name: 'node_exporter'
    static_configs:
      static_configs:
      - targets:
        - '3.93.215.32:9100'  # Existing node_exporter instance
        - '192.168.1.100:9100'  # New node_exporter instance (replace with your IP address)


```

Need to make sure node-exporter is also installed and running in the target instnace.


Open with draw.io
Search or jump to...
Dashboards › Node Exporter Full
Datasource default
Job node_exporter
Host localhost:9100
Quick CPU / Mem / Disk
Pressure
CPU Busy 25.3%
Sys Load 0.0%
RAM Used 44.1%
SWAP Used 0.5%
Root FS Used 63.4%
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic
Network Traffic Basic
Disk Space Used Basic


xporter Full
Add
Share
Datasource default
Job ec2
Host 3.80.78.183:9100
Quick CPU / Mem / Disk
Pressure
CPU Busy 64.8%
Sys Load 0.0%
RAM Used 27.4%
SWAP Used N/A
Root FS Used 19.4%
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic
Network Traffic Basic
Disk Space Used Basic