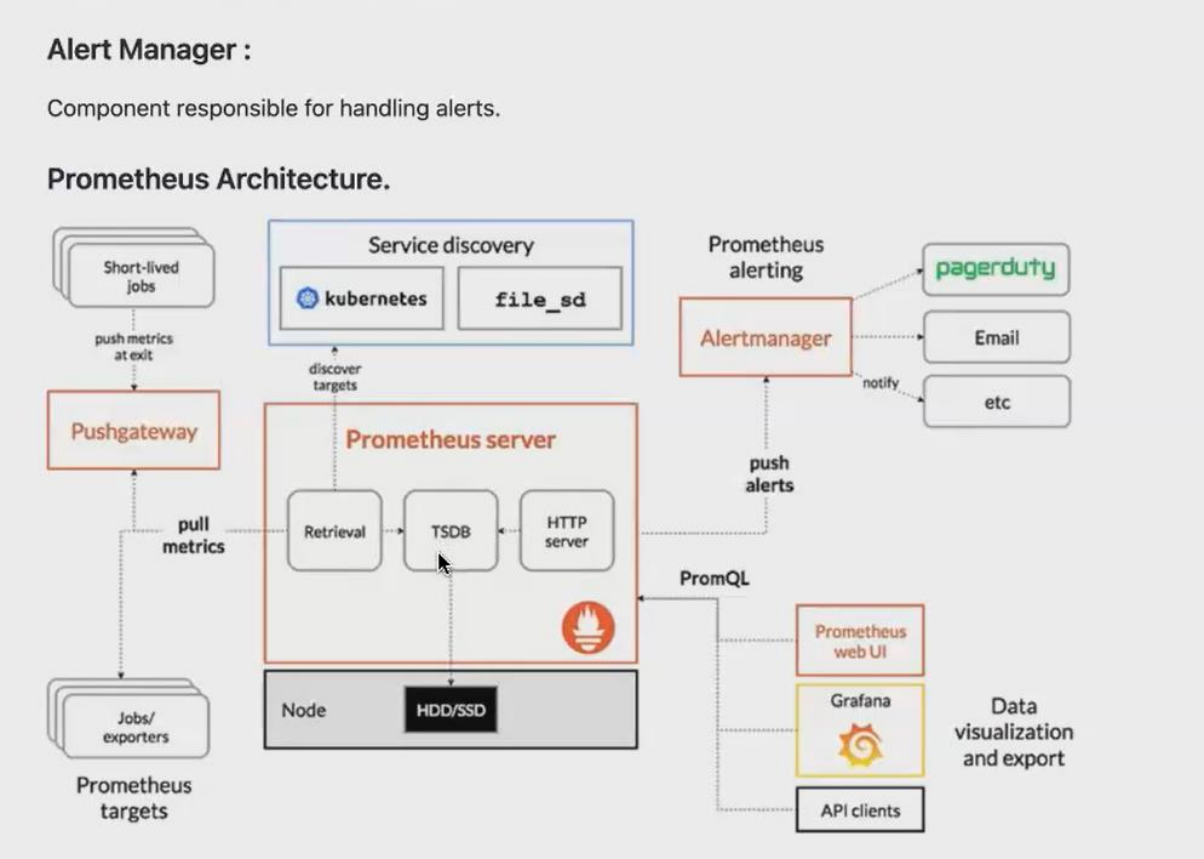
Day 2 - Monitoring Tool -Prometheus & Grafana

- Prometheus is a monitoring tool.
- So, Prometheus and Grafana are 2 different tools. But we'll be integrating both the tools.
- So, using Prometheus, you do all the monitoring part, you get the reports. We get all the information from our infrastructure and the application.
- we use Grafana just to visualize the reports that we got from the Prometheus.
- So, you have a lot of information. You will get lot of information from the infrastructure or from your application. You get a lot of information generates it keeps generating lot of reports. So just by looking into the report, getting some insights like getting some proper data will be tedious. We use a tool called Grafana, which is a data visualizing tool. So Grafana is a visualizing tool that we integrated with Prometheus.
- your data can be visualized so that it will be easier for the person who's looking into the data.
- So, it will be easier when something is given to you in the form of pictures rather than putting them in a 1 long document.
- If some data is given to you in the form of images, you can easily understand. That is what we are doing it here. Instead of giving the report in the form of document, we are giving it in the form of visuals. So, we'll be integrating a tool called Grafana, so that this Grafana will help you to visualize your data.
- using the data scraping mechanism, your prometheus will collect the metrics from your infrastructure.
- Prometheus uses a full based model, which means it periodically pulls the metric from the configured machine or the configured endpoints. So if I have a Ec2, and I want to monitor that Ec2.So what I can do I can configure my Prometheus in such a way that it monitors, it collects the data periodically from my Ec2.
- here your Prometheus, they have a database. If you see here, here you have this Prometheus server inside your Prometheus server. You have something called as Tsdb, just nothing but Time series database where your Prometheus stores all your collected data, your scraped data. Everything will be stored in the time series.

- here the data will be properly organized in your Tsdb. That is your time series database. Your data will be organized with the name of metric and label.
- so that if you want to query this database, okay, if I want to fetch any particular data, says, Suppose I am like. I am retrieving some logs. Your Prometheus is collecting some logs. I don't want all the logs that is being generated. I only want the error logs. So what we do, we all the logs will be there will be stored in my time series database.
- So your Prometheus will use a language called is prompt Ql to query the database.



Document:

https://docs.google.com/document/d/1_GDbffYmO7_45fkbd8Kv5msckANjftaM0JpbweAvQF4/edit

https://docs.google.com/document/d/1ELfM7oR-p4N5J3qLxm7fEZS4RsTdYLnqSrKrjRz-NS4/edit#heading=h.llwfmcv7eivz

[installation doc]

https://docs.google.com/document/d/1Jc3g0VxRbTtotgmVMM3ct9v2--MrU1ypPqYE6JnlTK4/edit#heading=h.ic9fb6wlxlki

https://github.com/yasminjeelani/Prometheus_Grafana

1. **Launch an ec2 [ubuntu] and connect it.**
2. **Clone the above github repo. everything in this repo**

```
ubuntu@ip-172-31-6-100:~$ git clone https://github.com/yasminjeelani/Prometheus_Grafana.git
Cloning into 'Prometheus_Grafana'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 0), reused 20 (delta 0), pack-reused 0
Receiving objects: 100% (20/20), done.
```

```
ubuntu@ip-172-31-6-100:~$ cd Prometheus_Grafana/
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ ls
README.md  install-grafana.sh       install-prometheus.sh       node-exporter.service  prometheus.yml      prometheus_relabeeling.yml
docker     install-node-exporter.sh  node-exporter-init.dservice  prometheus.service     prometheus_ec2.yml  prometheus_serviceDiscovery.yml
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$
```

Install-prometheus file: **[its an UI] open port no: 9090**

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

wget  https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz
tar -xvf prometheus-2.23.0.linux-amd64.tar.gz
sudo cp prometheus-2.23.0.linux-amd64/prometheus /usr/local/bin
sudo cp prometheus-2.23.0.linux-amd64/promtool /usr/local/bin
sudo cp -r prometheus-2.23.0.linux-amd64/consoles /etc/prometheus/
sudo cp -r prometheus-2.23.0.linux-amd64/console_libraries /etc/prometheus
sudo cp prometheus-2.23.0.linux-amd64/promtool /usr/local/bin/

rm -rf prometheus-2.23.0.linux-amd64.tar.gz prometheus-2.19.0.linux-amd64
sudo cp prometheus.yml /etc/prometheus/
sudo cp prometheus.service /etc/systemd/system/prometheus.service

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

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

```
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ vi install-prometheus.sh
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ ./install-prometheus.sh
--2024-07-29 11:33:12--  https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz
```

```
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-07-29 11:33:22 UTC; 29ms ago
   Main PID: 2193 (prometheus)
      Tasks: 4 (limit: 1120)
     Memory: 3.8M
        CPU: 6ms
```

After install Prometheus, /etc/Prometheus folder created.getinto this folder.

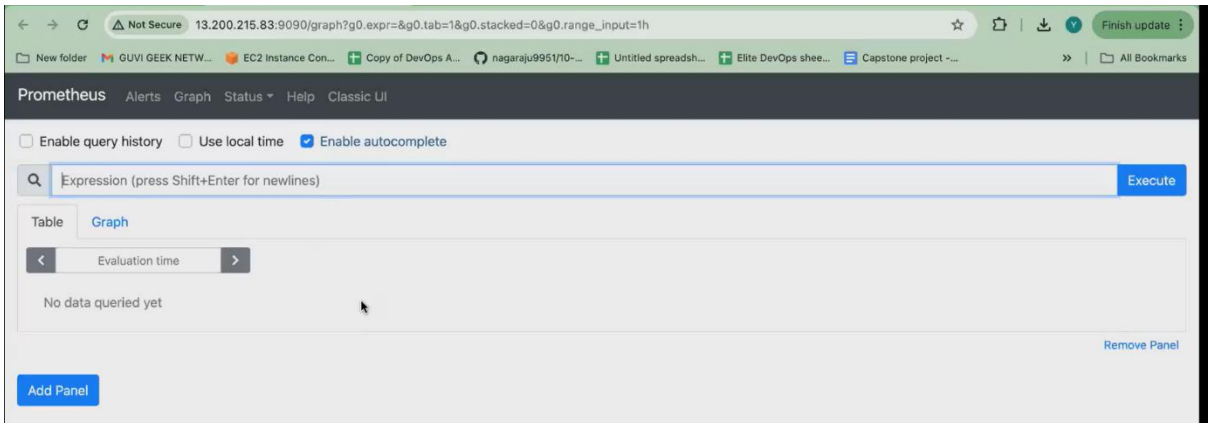
```
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ cd
ubuntu@ip-172-31-6-100:~$ cd /etc/prometheus
ubuntu@ip-172-31-6-100:/etc/prometheus$ ls
console_libraries  consoles  prometheus.yml
ubuntu@ip-172-31-6-100:/etc/prometheus$ vi prometheus.yml
```

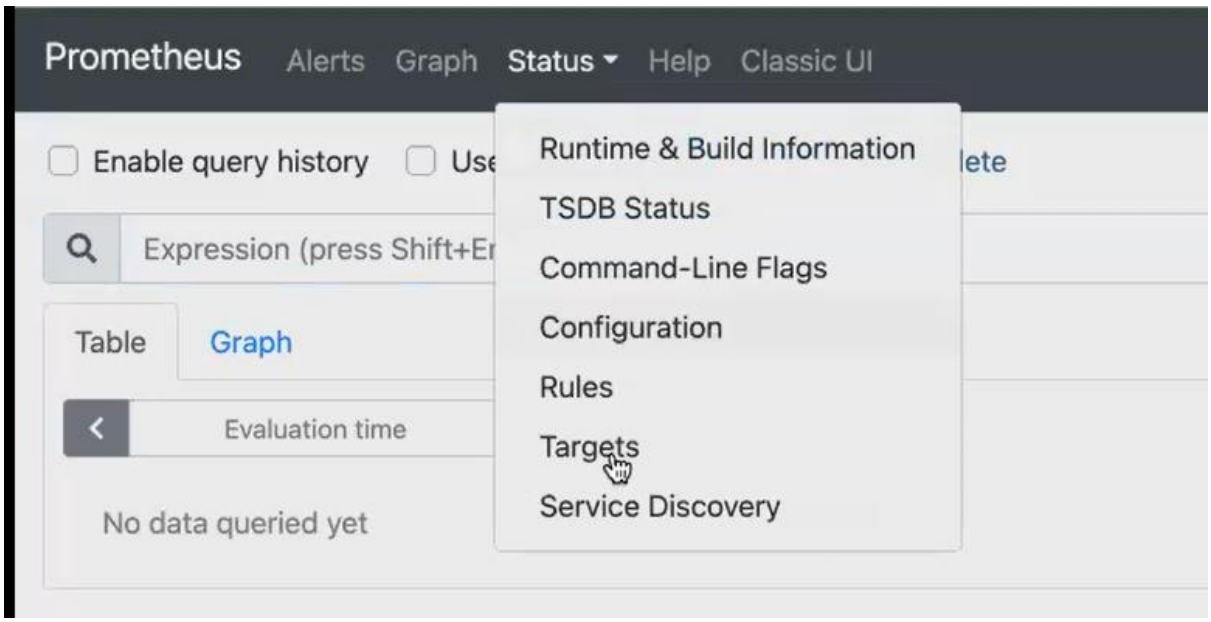
Prometheus.yml [configuring file]

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

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

Open the port no: 9090 [Prometheus port no:].copy ip address of the ec2 and paste it in browser. This is the Prometheus page.





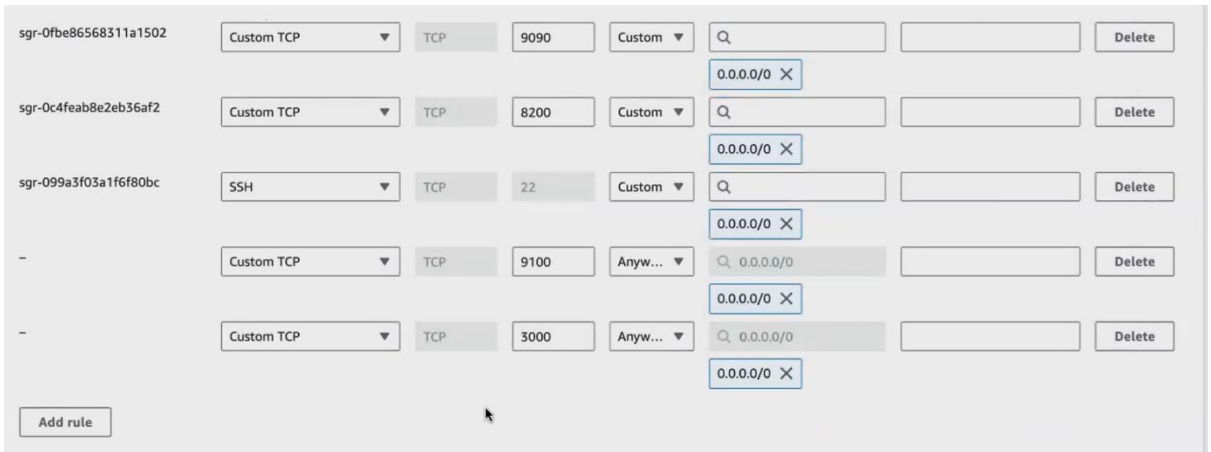
Install node exporter: **[its an UI]. open port no: 9100**

- **This node exporter will collect your data from the Ec. 2. So what is your Cp usage? What is your network usage? What is the disk? Usage, all the data your node exporter will collect and include it to your Prometheus.**

```
ubuntu@ip-172-31-6-100:/etc/prometheus$ vi prometheus.yml
ubuntu@ip-172-31-6-100:/etc/prometheus$ ./install-n
-bash: ./install-n: No such file or directory
ubuntu@ip-172-31-6-100:/etc/prometheus$ cd
ubuntu@ip-172-31-6-100:~$ cd Prometheus_Grafana/
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ ./install-node-exporter.sh
--2024-07-29 11:38:18--  https://github.com/prometheus/node_exporter/releases/download/v1.0.1/node_exporter-1.0.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/2ae54580-afed-11ea-8b8a-89172cffc39d?X-Amz-
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/node-exporter.service → /etc/systemd/system/node-exporter.service.
● node-exporter.service - Prometheus Node Exporter Service
     Loaded: loaded (/etc/systemd/system/node-exporter.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-07-29 11:38:21 UTC; 34ms ago
   Main PID: 2308 (node_exporter)
      Tasks: 3 (limit: 1120)
     Memory: 1.7M
        CPU: 6ms
     CGroup: /system.slice/node-exporter.service
             └─2308 /usr/local/bin/node_exporter
```

open port no: 9100

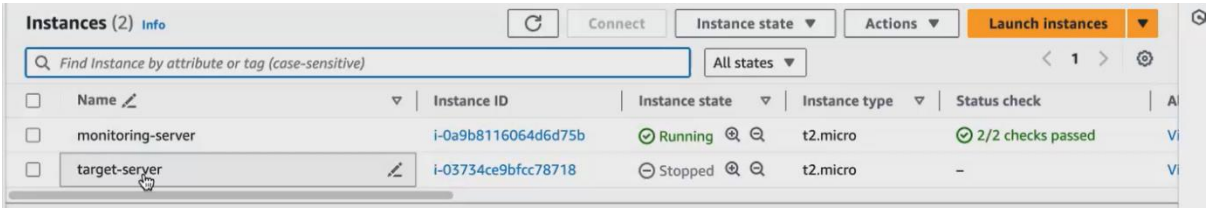


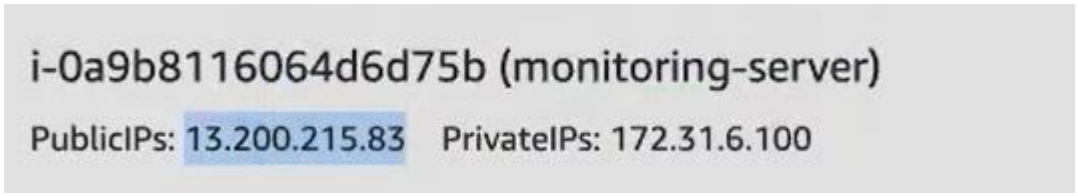
Install Grafana: **[its an UI] open port no:3000**

```
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ ./install-grafana.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
adduser is already the newest version (3.118ubuntu5).
adduser set to manually installed.
libfontconfig1 is already the newest version (2.13.1-4.2ubuntu5).
libfontconfig1 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
--2024-07-29 11:39:58--  https://dl.grafana.com/oss/release/grafana_7.3.4_amd64.deb
Resolving dl.grafana.com (dl.grafana.com)... 151.101.38.217, 2a04:4e42:8e::729
Connecting to dl.grafana.com (dl.grafana.com)|151.101.38.217|:443... connected.
```

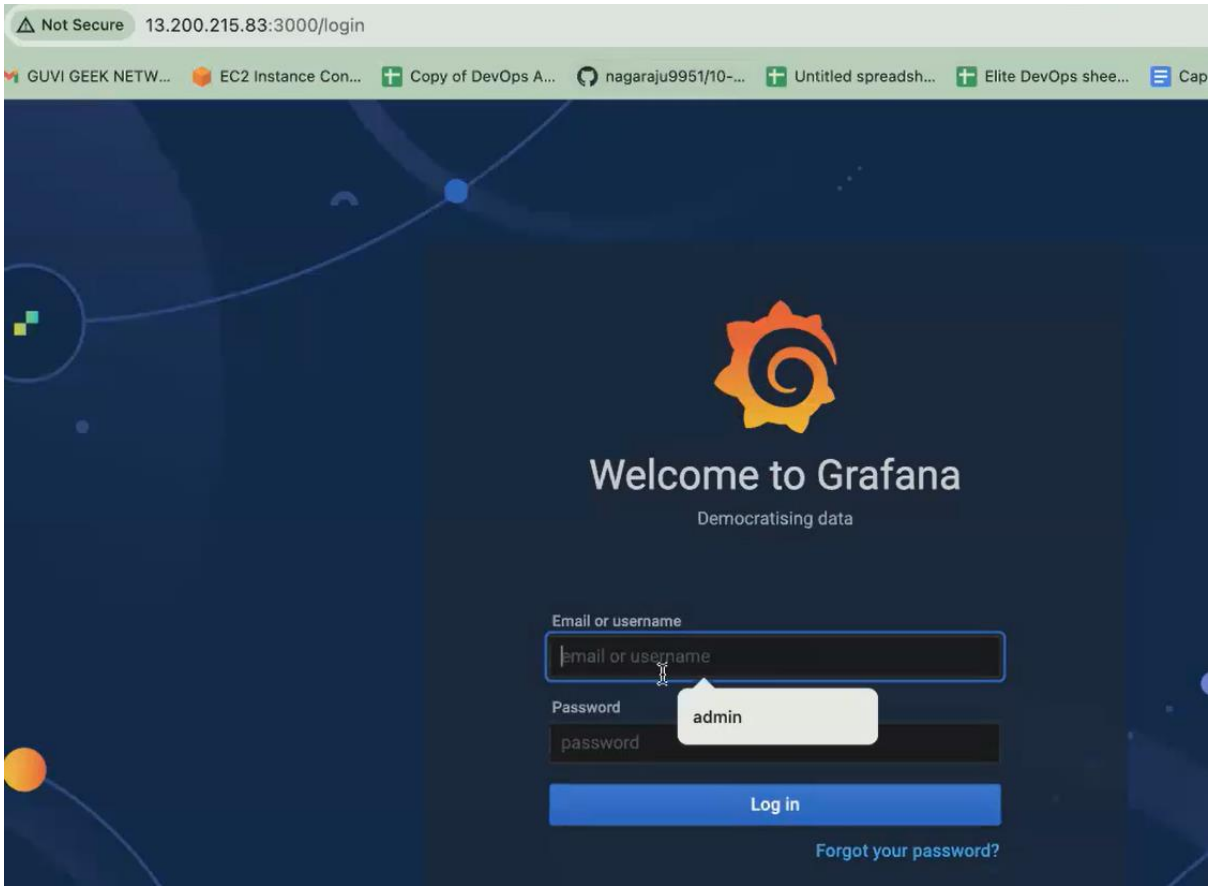
Monitoring -server: [install Prometheus and Grafana]

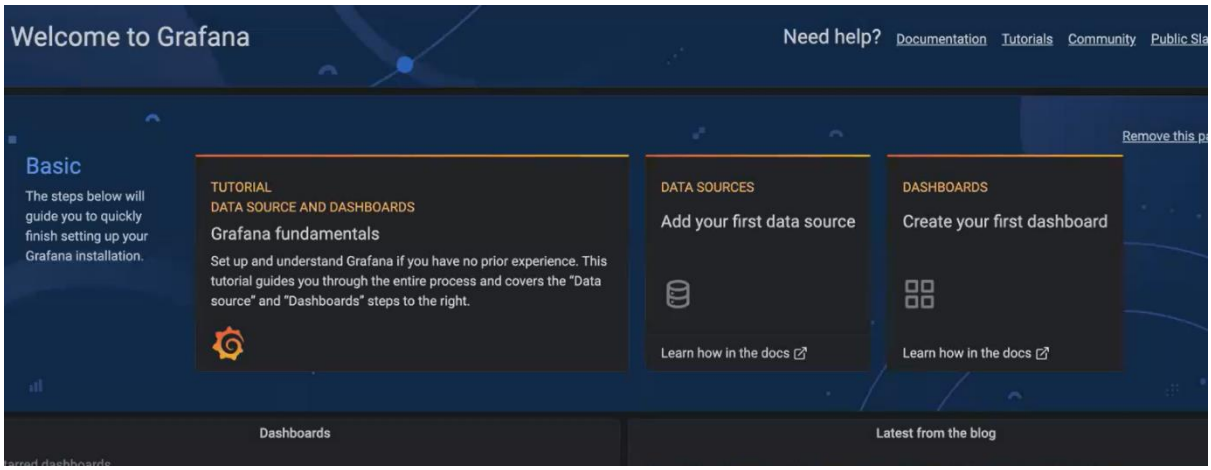
Target -server: [install node exporter]





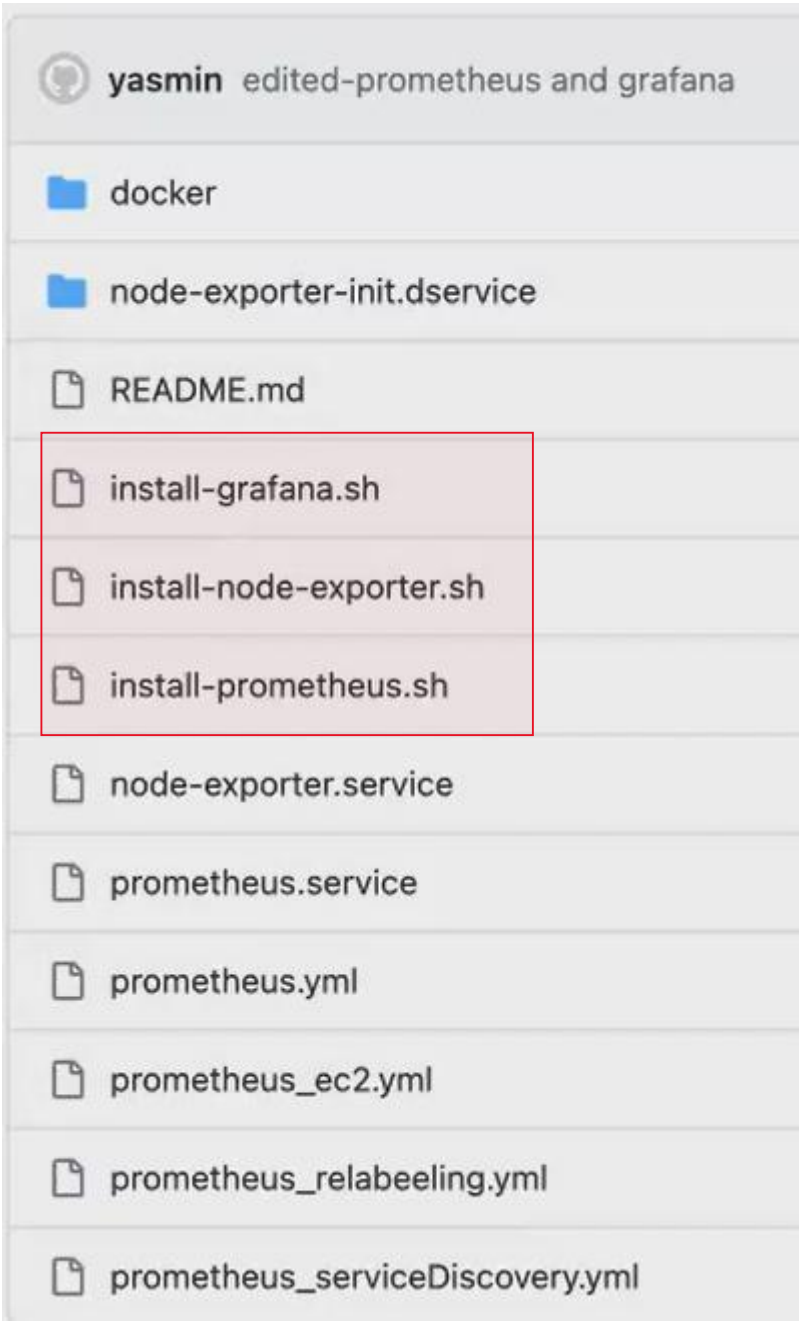
Copy IP address and open. This is login page .initially give username :admin password:admin.later on change it.





Copy monitoring server ip.This is node exporter page. Click metrics shows logs.

All installation script given in github



In target server install node exporter.

This monitoring server have to monitor your target machine. Clone the repo in target server.

```
ubuntu@ip-172-31-11-233:~$ git clone https://github.com/yasminjeelani/Prometheus_Grafana.git
Cloning into 'Prometheus_Grafana'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 0), reused 20 (delta 0), pack-reused 0
Receiving objects: 100% (20/20), done.
ubuntu@ip-172-31-11-233:~$
```

Once cloned. get into the Prometheus-Grafana folder. Execute the ./install-node.sh file

```
ubuntu@ip-172-31-11-233:~$ cd Prometheus_Grafana/
ubuntu@ip-172-31-11-233:~/Prometheus_Grafana$ ls
README.md  install-grafana.sh       install-prometheus.sh       node-exporter.service  prometheus.yml      prometheus_relabeeling.yml
docker     install-node-exporter.sh  node-exporter-init.dservice  prometheus.service     prometheus_ec2.yml  prometheus_serviceDiscovery.yml
ubuntu@ip-172-31-11-233:~/Prometheus_Grafana$ ./install-node-exporter.sh
--2024-07-29 11:47:56--  https://github.com/prometheus/node_exporter/releases/download/v1.0.1/node_exporter-1.0.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

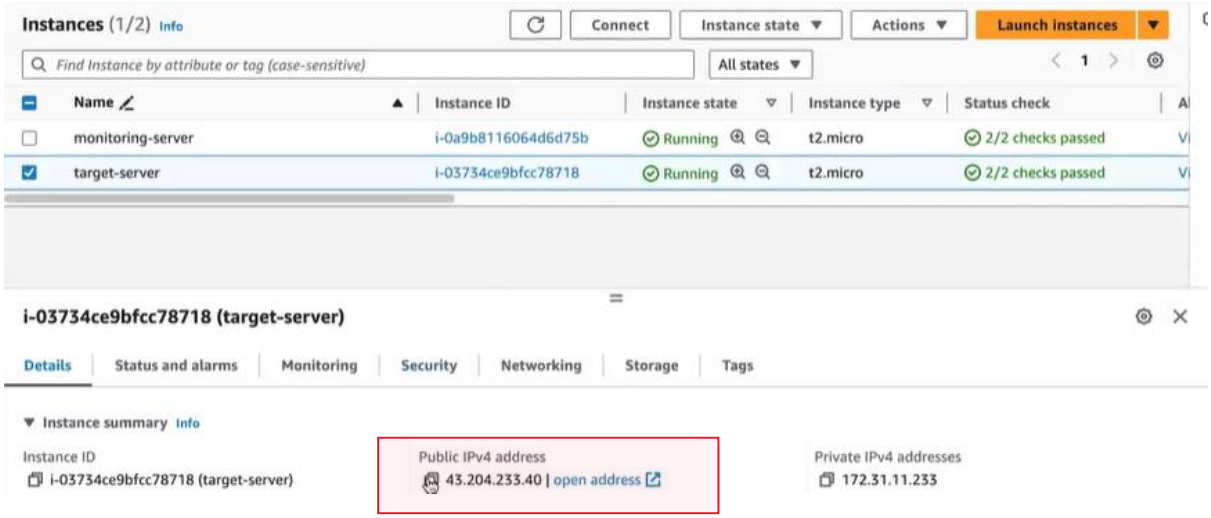
Get into monitoring machine. Edit the Prometheus.yml file , give sudo permission .then only edit this

```
ubuntu@ip-172-31-6-100:~/Prometheus_Grafana$ cd
ubuntu@ip-172-31-6-100:~$ cd /etc/prometheus
ubuntu@ip-172-31-6-100:/etc/prometheus$ ls
console_libraries  consoles  prometheus.yml
```

- Local host is node exporter. **So, who's collecting the data from the target machine. Your node exporter. So, Node exporter runs on the port number 9100. your local machine, the machine where I have installed my Prometheus.**
- Target -server ip address 43.204.233.40: 9100[node exporter port no:]

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

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9100', '43.204.233.40:9100']
```

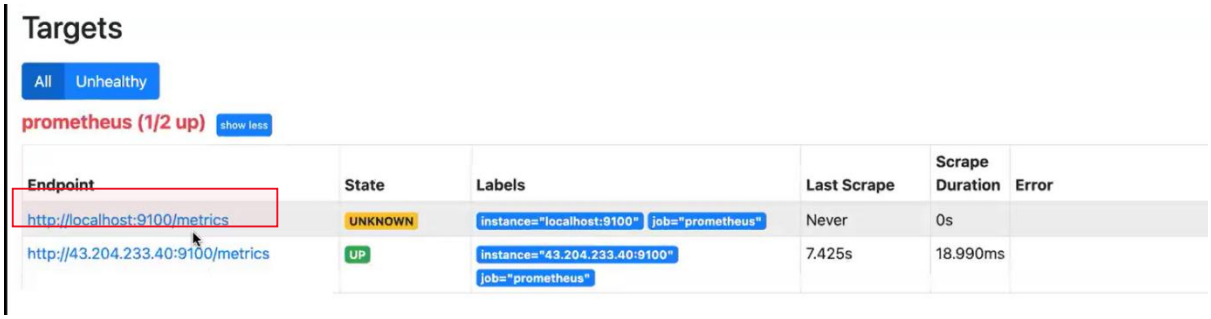



```
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo systemctl restart prometheus
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo systemctl status prometheus
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-07-29 11:53:06 UTC; 6s ago
   Main PID: 2586 (prometheus)
      Tasks: 6 (limit: 1120)
     Memory: 21.2M
        CPU: 99ms
     CGroup: /system.slice/prometheus.service
             └─2586 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --st
```

To restart your Prometheus, so give sudo systemctl **restart, because we are changing the configuration file. So for you, for you to apply the configuration. You have to restart it** now it's running. Now, my Prometheus is running.

```
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo systemctl restart prometheus
ubuntu@ip-172-31-6-100:/etc/prometheus$
```

```
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo vi prometheus.yml
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo systemctl restart prometheus
ubuntu@ip-172-31-6-100:/etc/prometheus$ sudo systemctl status prometheus
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-07-29 11:53:06 UTC; 6s ago
   Main PID: 2586 (prometheus)
      Tasks: 6 (limit: 1120)
     Memory: 21.2M
        CPU: 99ms
     CGroup: /system.slice/prometheus.service
             └─2586 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --s
```



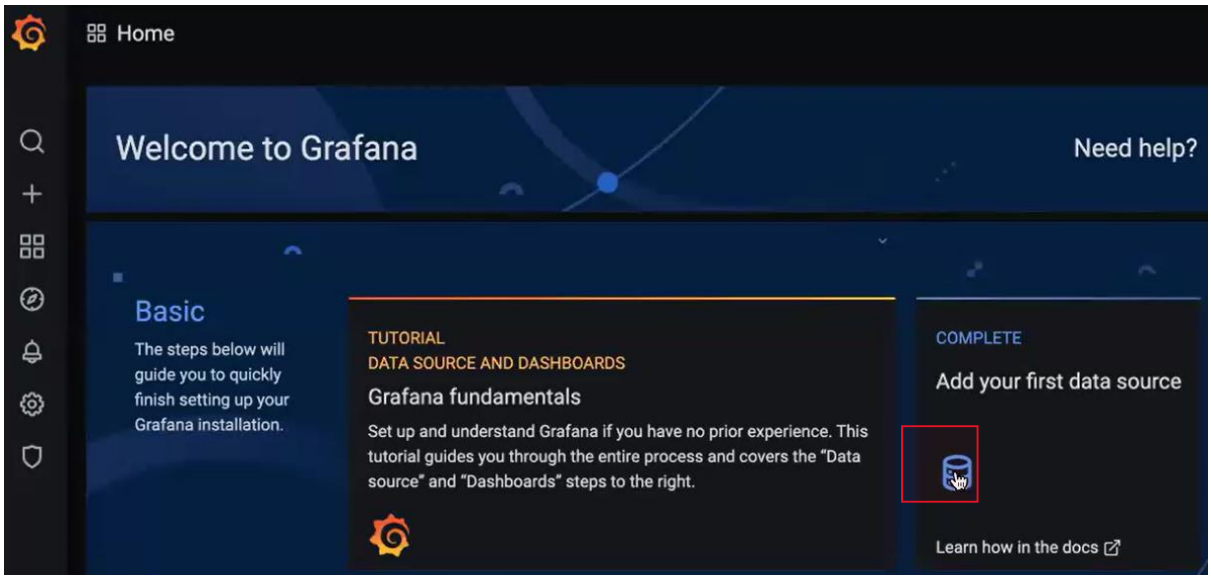
```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
```

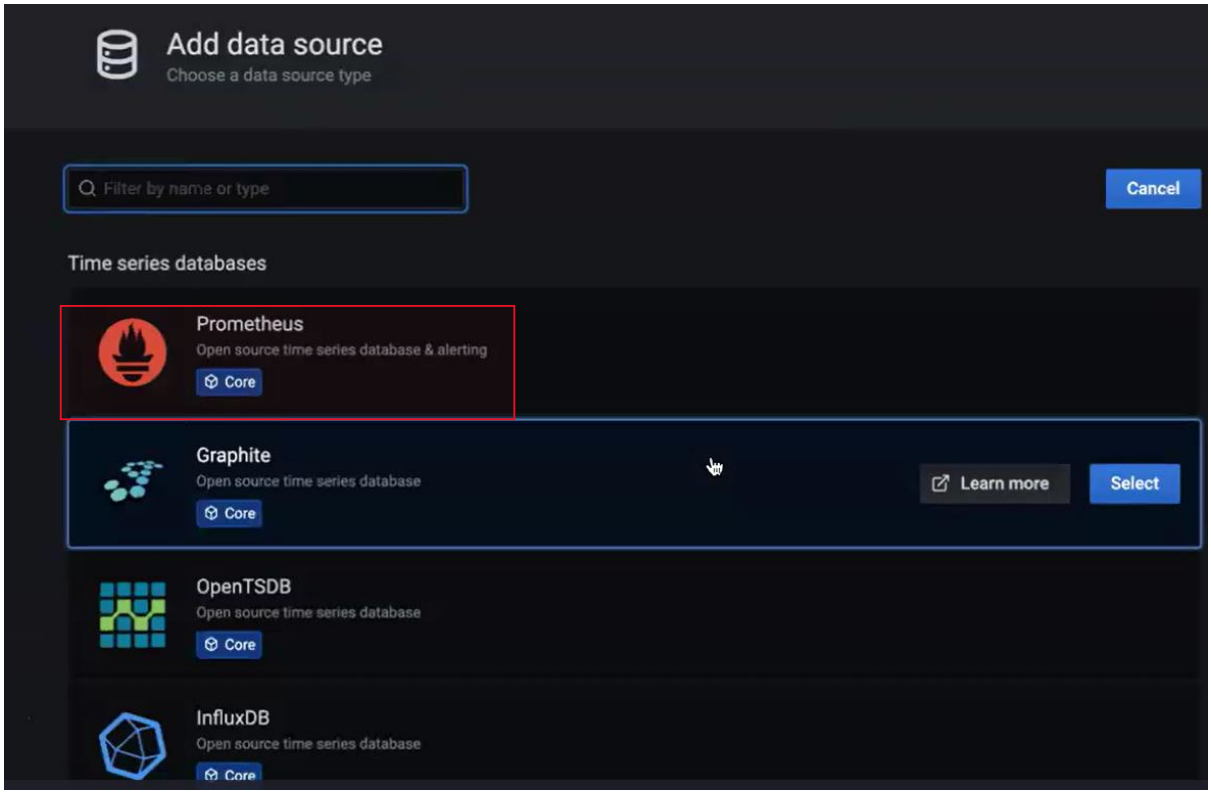
integrating this Prometheus with Grafana:

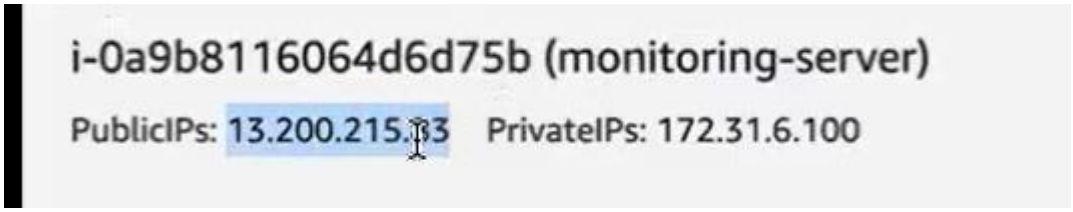
Prometheus [monitoring tool (i.e) monitoring the target machine)

Grafana [visualization tool]

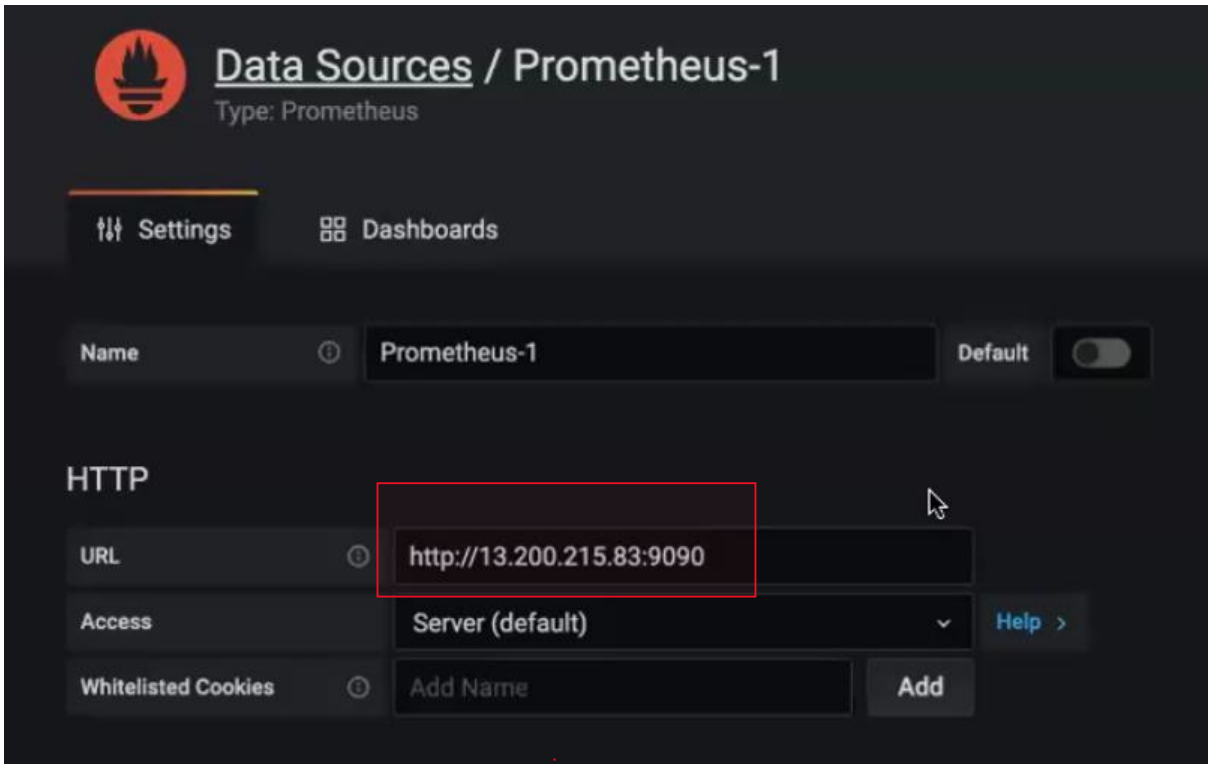
- you can connect your Grafana with any data source.
- your data source can be a excel sheet or can be a Csv data source.
- And more mainly, we also use Grafana for data science projects also.
- To process the data to visualize the data we use Grafana.
- The Grafana is mainly a data visualization tool. The data source that we are using now is Prometheus. So Prometheus will get all the details from the target machine, so it will collect all the details from the target machine.
- Data base and data source is same.

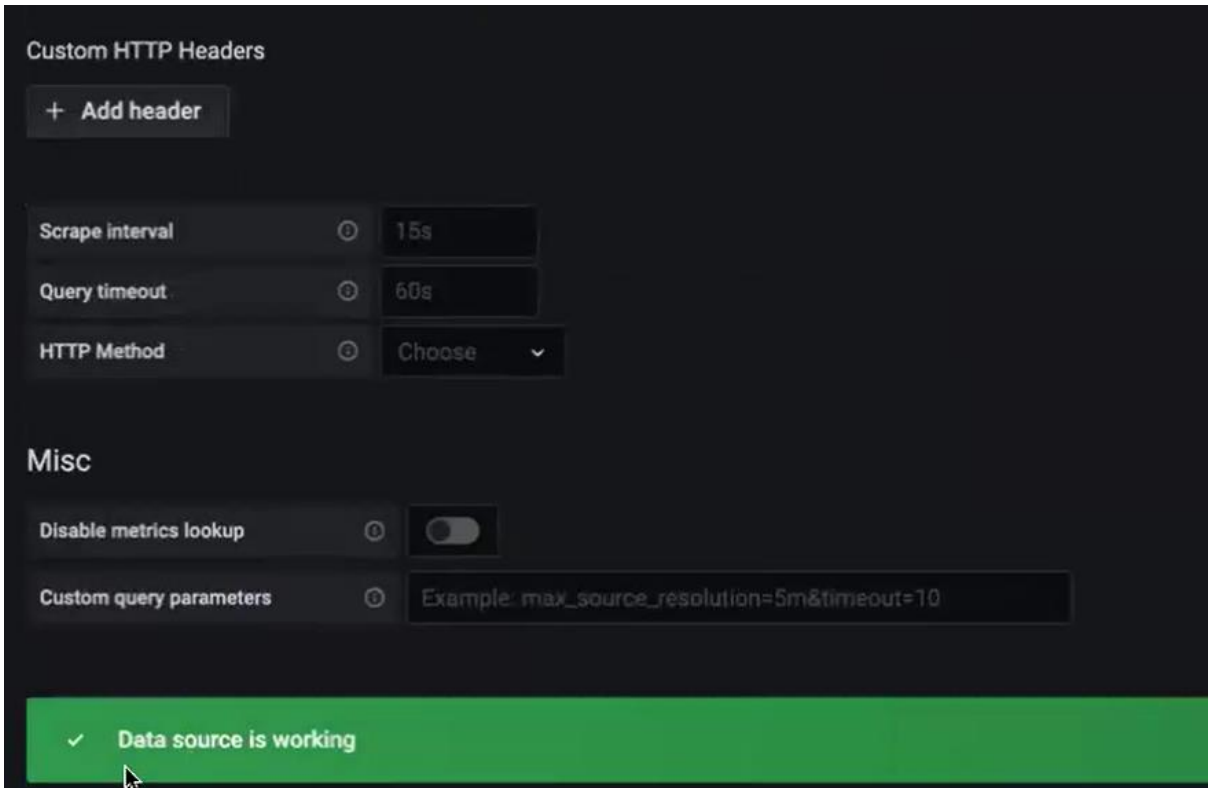




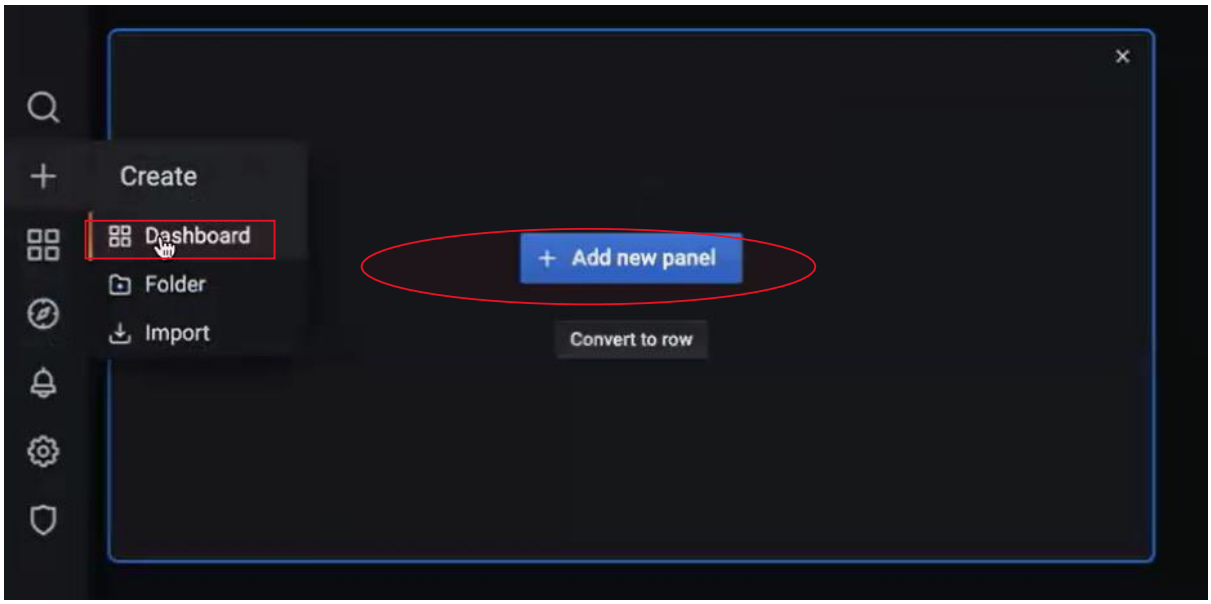


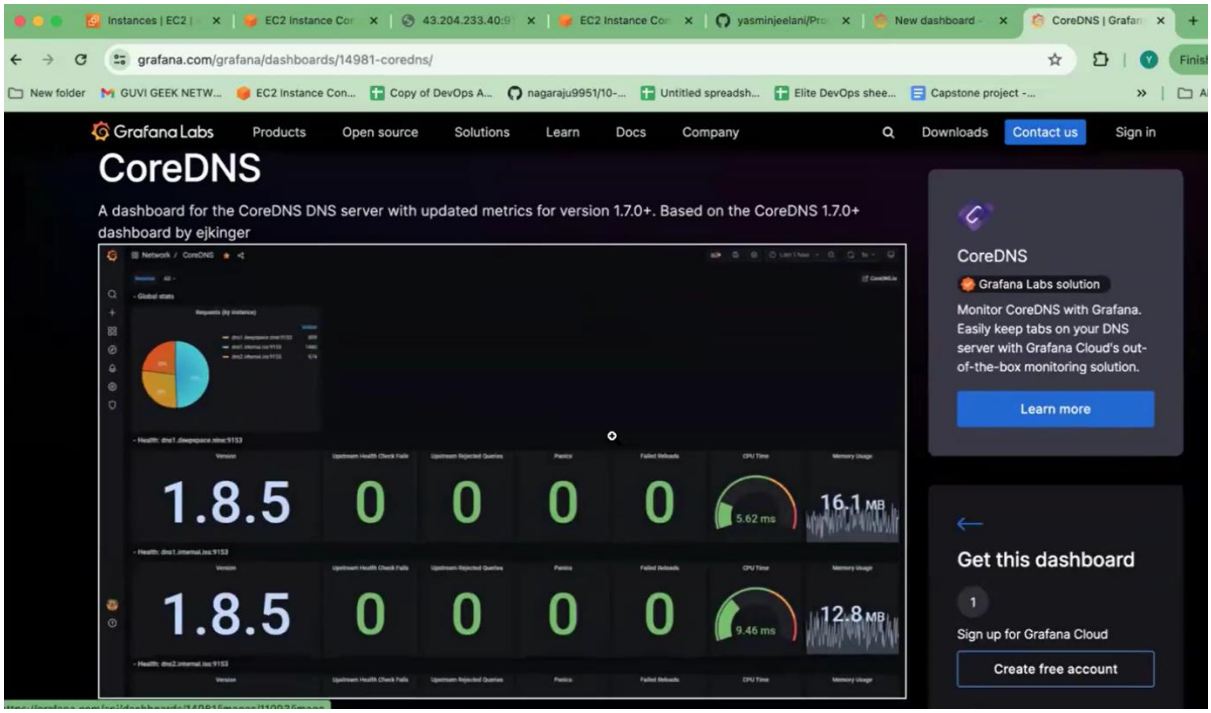
Copy the ip address of the monitoring machine. Because in this machine only install Grafana and Prometheus.


Data Sources / Prometheus-1
Type: Prometheus
Settings
Dashboards
Name
Prometheus-1
Default
HTTP
URL
http://13.200.215.83:9090
Access
Server (default)
Help
Whitelisted Cookies
Add Name
Add

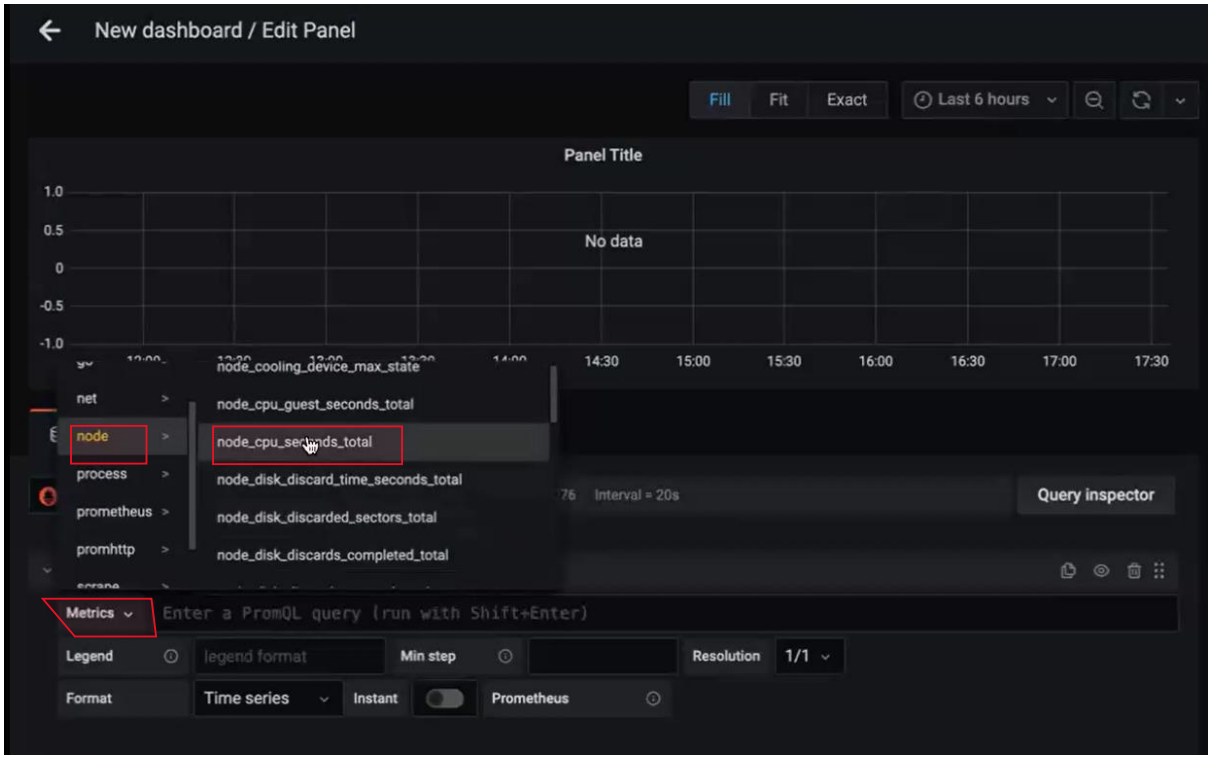

Custom HTTP Headers
+ Add header
Scrape interval
15s
Query timeout
60s
HTTP Method
Choose
Misc
Disable metrics lookup
Custom query parameters
Example: max_source_resolution=5m&timeout=10
Data source is working

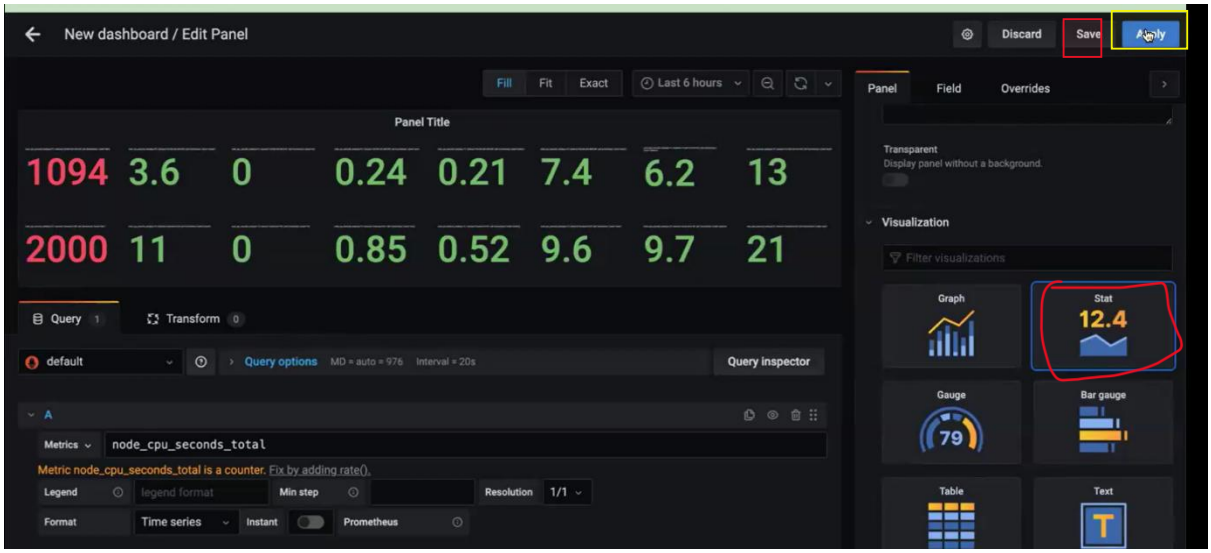
Create a dashboard for CPU utilization:

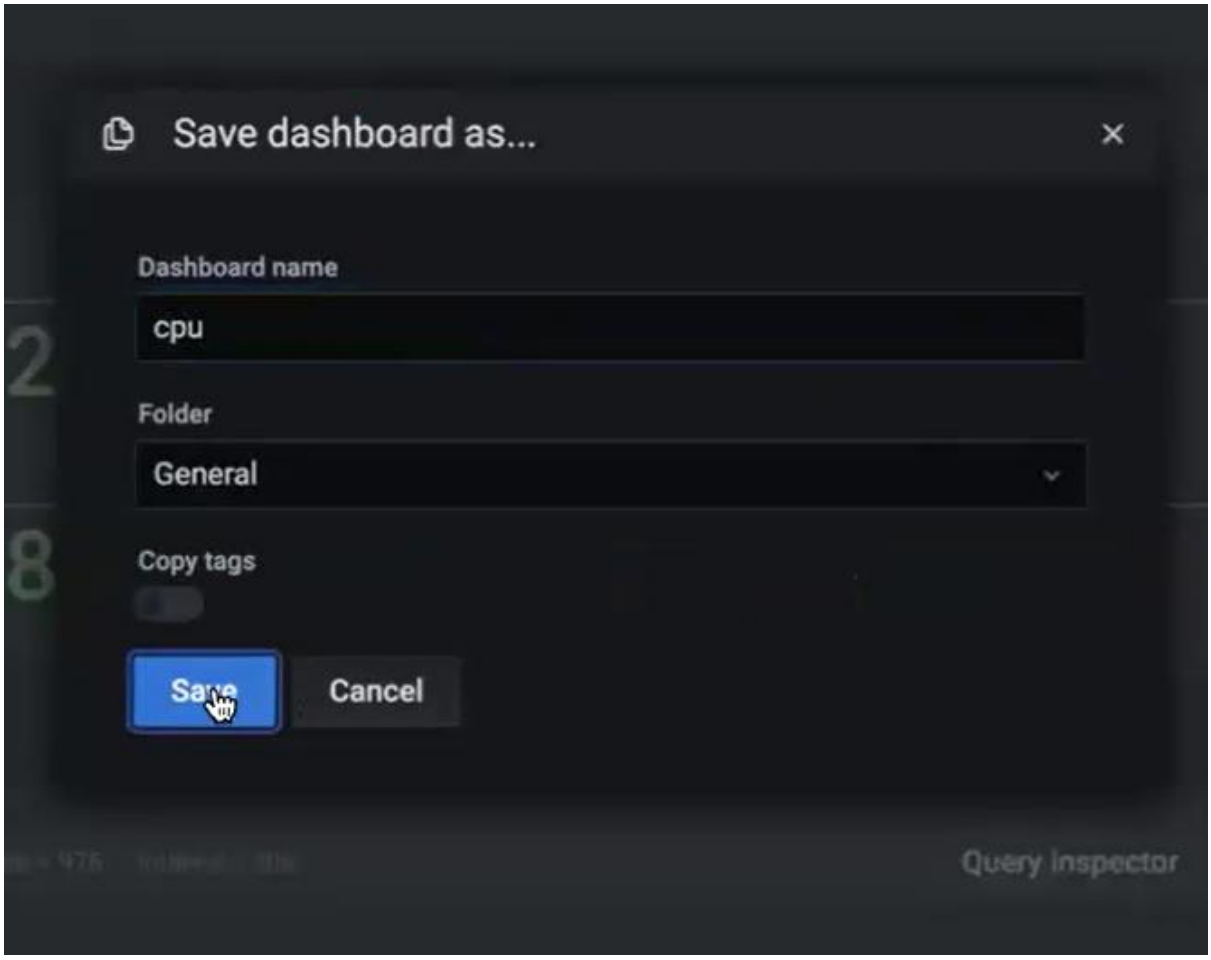


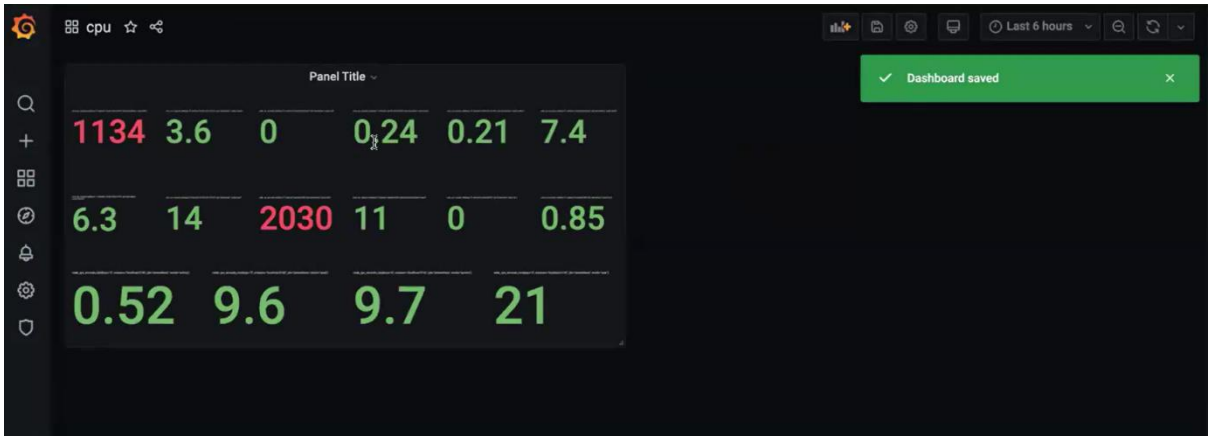


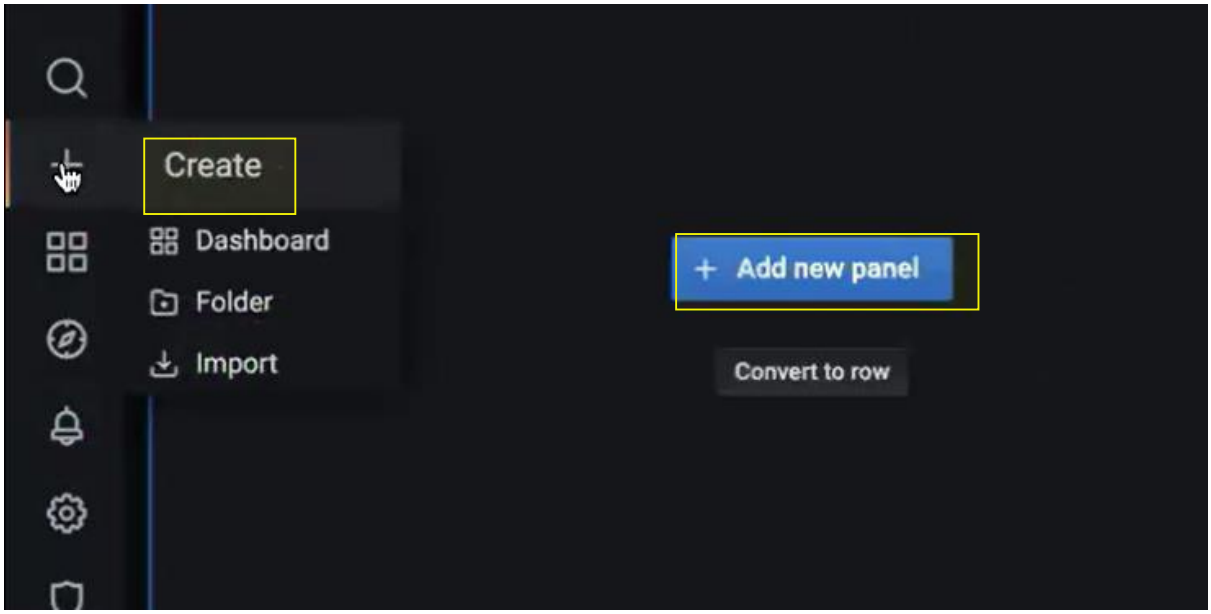
Node means system.Create metrics for CPU utilization → apply → select start

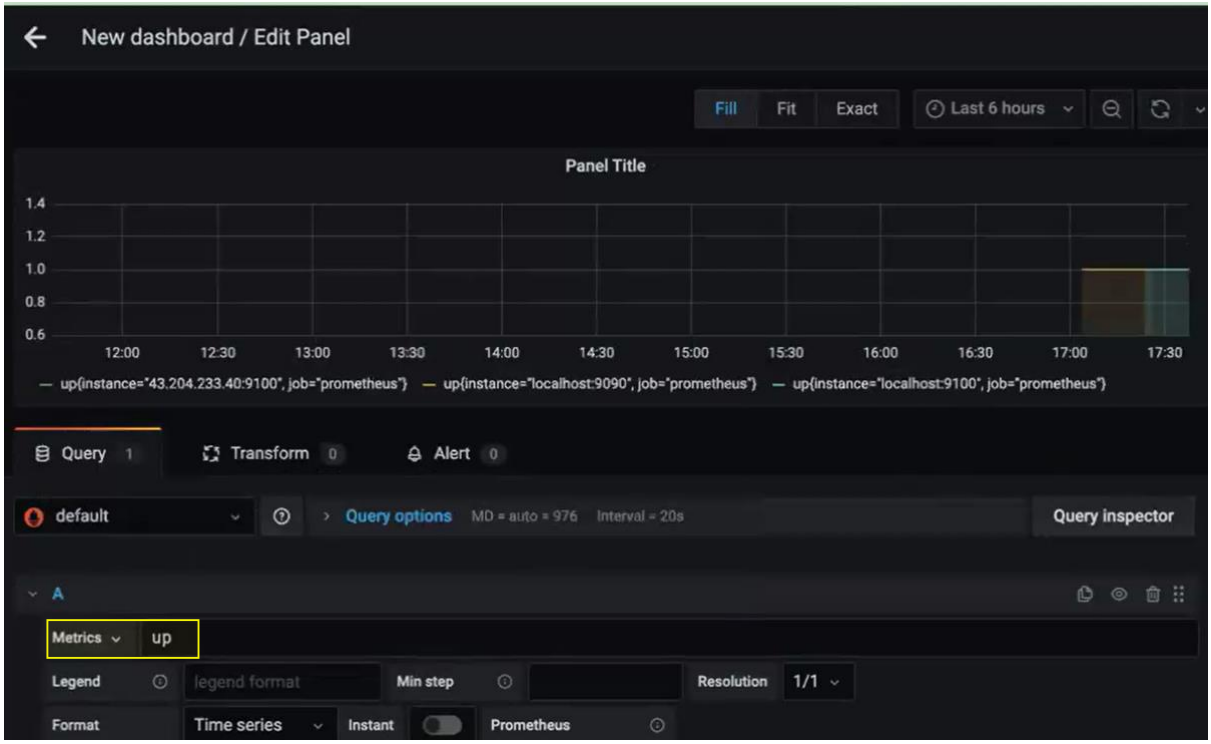


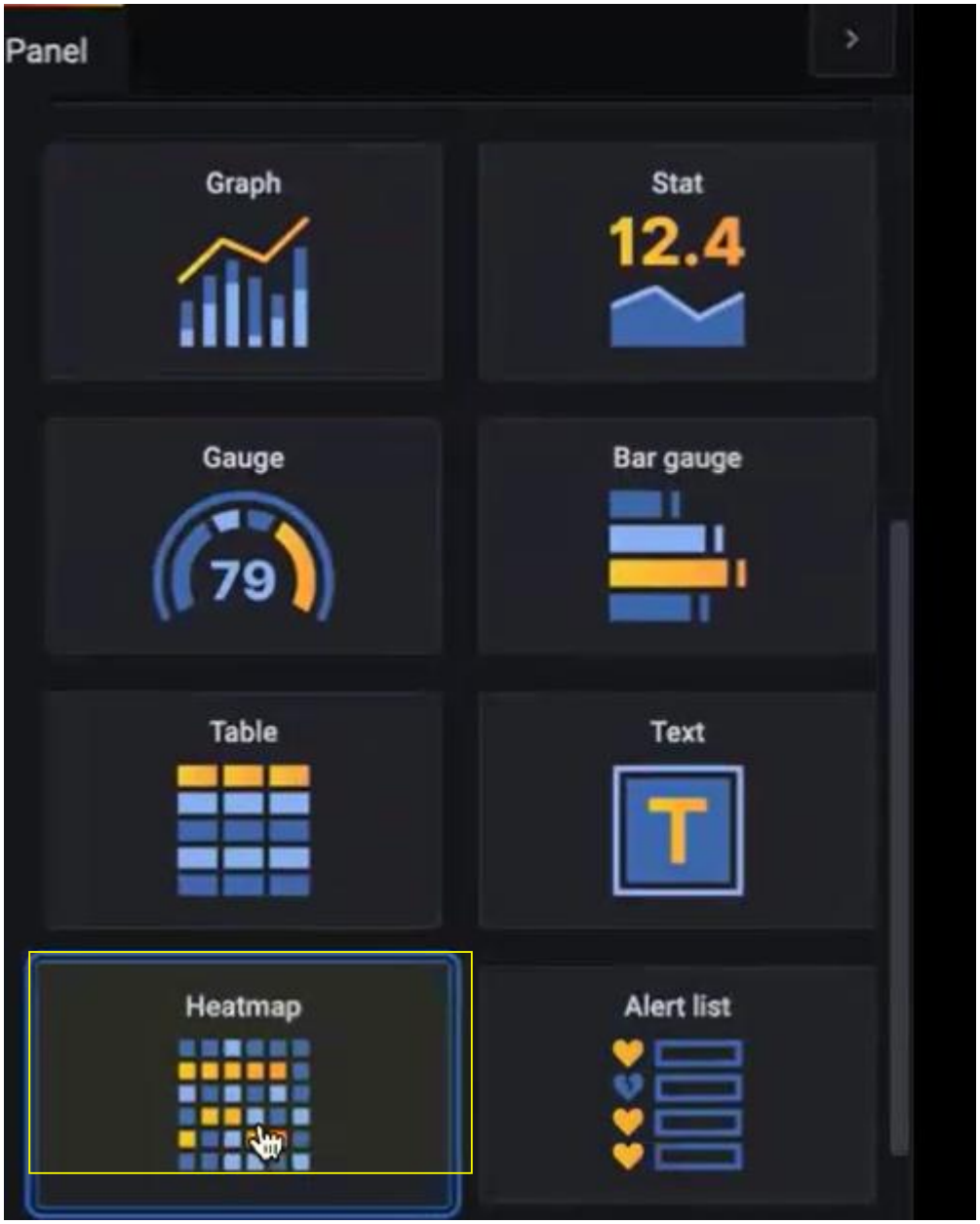


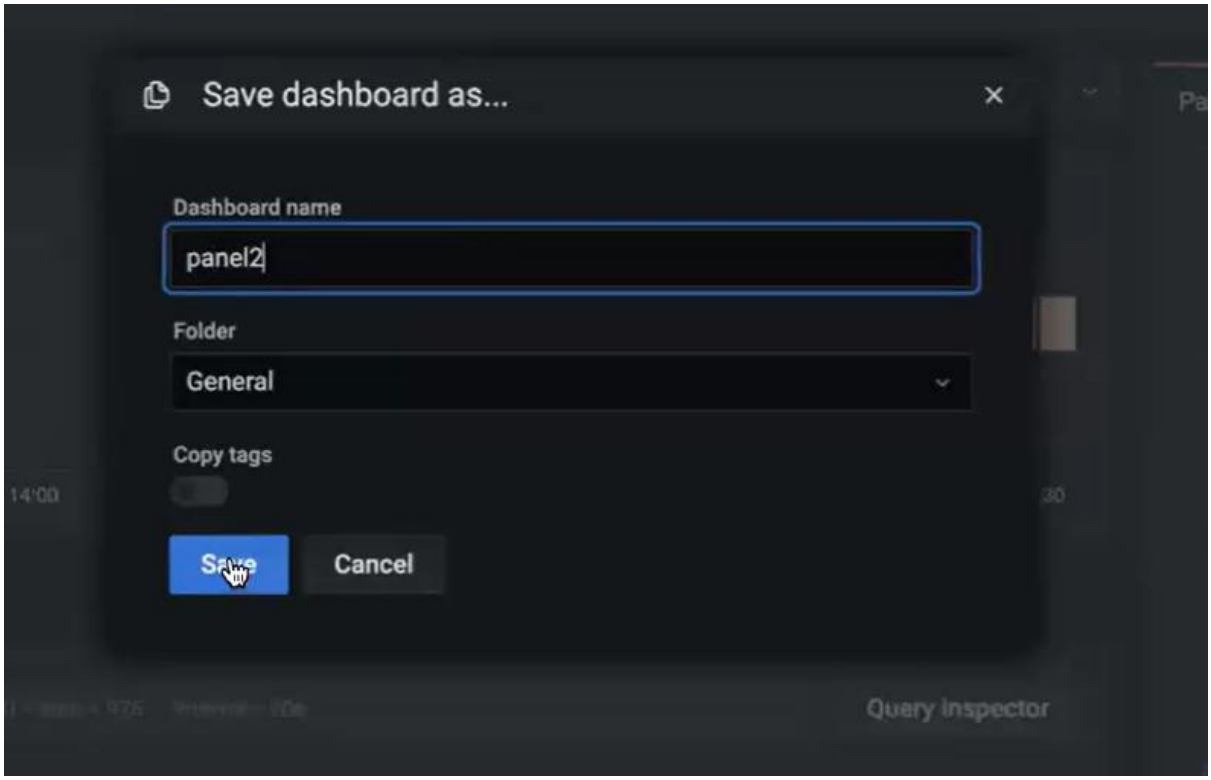

Save dashboard as...
Dashboard name
cpu
Folder
General
Copy tags
Save
Cancel
Query Inspector


cpu
Last 6 hours
Dashboard saved
Panel Title
1134 3.6 0 0.24 0.21 7.4
6.3 14 2030 11 0 0.85
0.52 9.6 9.7 21

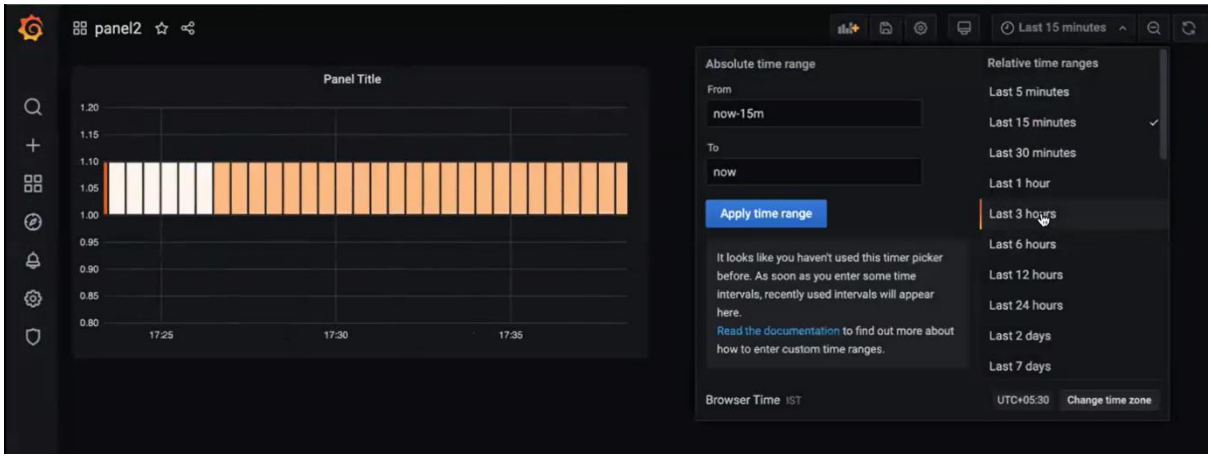

Create
Dashboard
Folder
Import
+ Add new panel
Convert to row


New dashboard / Edit Panel
Fill
Fit
Exact
Last 6 hours
Panel Title
1.4
1.2
1.0
0.8
0.6
12:00
12:30
13:00
13:30
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
up{instance="43.204.233.40:9100", job="prometheus"}
up{instance="localhost:9090", job="prometheus"}
up{instance="localhost:9100", job="prometheus"}
Query 1
Transform 0
Alert 0
default
Query options
MD = auto = 976
Interval = 20s
Query inspector
A
Metrics
up
Legend
legend format
Min step
Resolution
1/1
Format
Time series
Instant
Prometheus


Panel
Graph
Stat
12.4
Gauge
79
Bar gauge
Table
Text
T
Heatmap
Alert list


Save dashboard as...
Dashboard name
panel2
Folder
General
Copy tags
Save
Cancel
Query Inspector


panel2
Panel Title
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
0.80
12:00
13:00
14:00
15:00
16:00
17:00

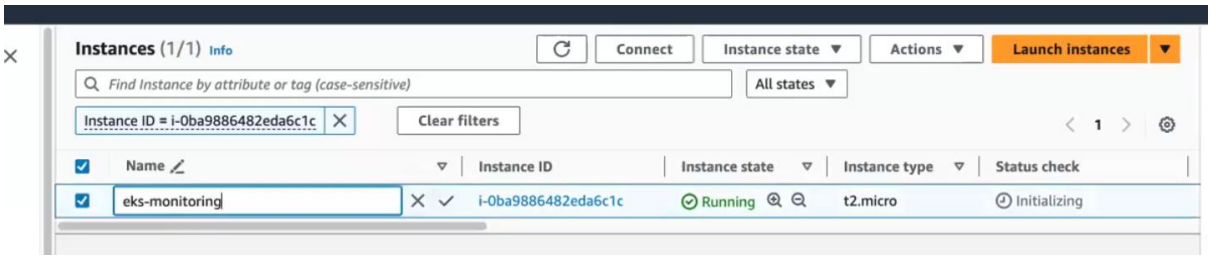


- Prometheus is mainly used to monitor your Kubernetes cluster only.
- You should have node exporter running. Without the node exporter you cannot collect the data.
- So in your target machine you will only have node exporter installed.
- But in your monitoring machine you will have all the 3 tools installed, and you should have configured the Security group also.

we will see how to monitor, the Kubernetes cluster: [go through the below doc]

launch an instance and connect.

create Kubernetes cluster




```
ubuntu@ip-172-31-35-101:~$ sudo apt update
```

```
ubuntu@ip-172-31-35-101:~$ sudo apt install awscli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
ubuntu@ip-172-31-35-101:~$ aws configure
AWS Access Key ID [None]: AKIASGIII4IGBNATXP47
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
```

```
ubuntu@ip-172-31-35-101:~$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | ta
r xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin
eksctl version
0.187.0
```

```
ubuntu@ip-172-31-35-101:~$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl

chmod +x ./kubectl
mv ./kubectl /usr/local/bin
kubectl version --short --client
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
```

```
ubuntu@ip-172-31-35-101:~$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0 57.4M    0  117k    0     0  74207      0  0:13:32  0:00:01  0:13:31 74166
```

```
ubuntu@ip-172-31-35-101:~$ chmod +x ./kubectl
ubuntu@ip-172-31-35-101:~$ mv ./kubectl /usr/local/bin
mv: cannot move './kubectl' to '/usr/local/bin/kubectl': Permission denied
ubuntu@ip-172-31-35-101:~$ sudo mv ./kubectl /usr/local/bin
```

```
ubuntu@ip-172-31-35-101:~$ kubectl version --short --client
Client Version: v1.19.6-eks-49a6c0
```

```
ubuntu@ip-172-31-35-101:~$ eksctl create cluster --name guvi \
   --region ap-south-1 \
--node-type t2.small
```

https://github.com/yasminjeelani/Prometheus_Grafana

https://docs.google.com/document/d/1_GDbffYmO7_45fkbd8Kv5msckANjftaM0JpbweAvQF4/edit

https://docs.google.com/document/d/1ELfM7oR-p4N5J3qLxm7fEZS4RsTdYLnqSrKrjRz-NS4/edit#heading=h.llwfmcv7eivz

https://docs.google.com/document/d/1Jc3g0VxRbTtotgmVMM3ct9v2--MrU1ypPqYE6JnlTK4/edit#heading=h.ic9fb6wlxlki [**we will see how to monitor, the Kubernetes cluster**]