

**Document Information and History**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Section** | **Date** | **Author** | **Reviewer** | **Review Date** | **Approver** | **Approved Date** | **Description** |
| A | 14-May- 2024 | Balaji M | Parameswara Rao |  | Manjusha Ravindran |  | System logs. |
| B | 14-May- 2024 | Balaji M | Parameswara Rao |  | Manjusha Ravindran |  | Docker logs |
| C | 14-May- 2024 | Balaji M | Parameswara Rao |  | Manjusha Ravindran |  | Jenkins logs |
|  |  |  |  |  |  |  |  |

**Document Control Information**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document ID** | **Document Owner** | **Document Approver** | **Document Custodian** | **Location** | **Retention** | **Disposition** |
| Prometheous-Grafana | Balaji m | Manjusha Ravindran |  | Devops Sharepoint |  |  |

# **Followed Below document**

# **NOTE**: - https://devops4solutions.com/monitoring-using-prometheus-and-grafana-on-aws-ec2/ **Section: A System logs Monitoring** **Requirements Monit0r the servers in main server: -**

* Prometheous
* Grafana
* Node\_exporter

# **Requirements for target servers: -**

# System metrics monitoring -Node\_exporter

# Docker metrics monitoring **-**Node\_exporter

* Jenkins metrics monitoring - Node\_exporter and Install Prometheus plugin in Jenkins dashboard

# **Introduction: -**

Prometheous:

Prometheus is an open-source technology designed to provide monitoring and alerting functionality for cloud-native environments, including Kubernetes. It can collect and store metrics as time-series data, recording information with a timestamp.

Prometheus query language (PromQL) to filter, aggregate, ingest, and query millions of unique time series metrics from your self-managed Kubernetes clusters. Automatically scale as your ingestion and query needs grow and maintain consistent response times for large container deployments.

**Step1: - Installation of the Prometheus**

### **A) Create a new user and add new directories for Prometheus**

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

**B) Download the Prometheus, extract it and put it in /usr/local/bin folder**

**Wget**[**https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz**](https://github.com/prometheus/prometheus/releases/download/v2.23.0/prometheus-2.23.0.linux-amd64.tar.gz) **(or)** Head to <https://prometheus.io/download/> and download the latest binary for the Prometheus

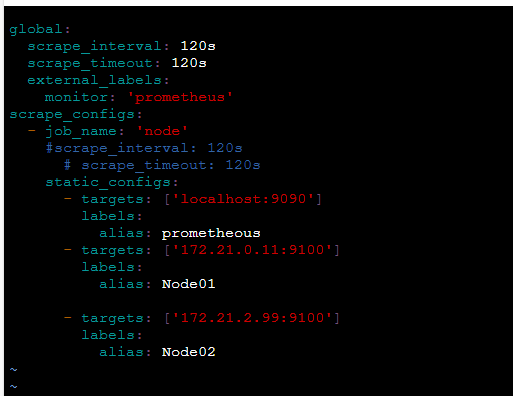
tar -xvf prometheus-2.23.0.linux-amd64.tar.gz  
 sudo cp prometheus-2.23.0.linux-amd64/prometheus /usr/local/bin  
 sudo cp prometheus-2.23.0.linux-amd64/promtool /usr/local/bin  
 sudo cp -r prometheus-2.23.0.linux-amd64/consoles /etc/prometheus/

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/

**C)** **configure Prometheus to monitor itself using yaml file**

Create a **prometheus.yml** file at **/etc/prometheus/prometheus.yml** with the below content



#### **Step2:- Security Groups Configuration**

**Ensure ports are enabled:**

Port **9090**— Prometheus Server

Port **9100**— Prometheus Node Exporter

Port  **3002**— Grafana

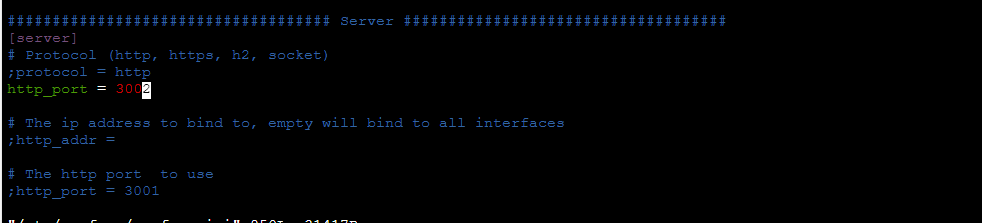
Port **9323**=--docker

Port **8080**--Jenkins(poc account)

To change the default Grafana GUI port number, you need to modify the configuration file. Here's how you can do it:

1. Locate the Grafana configuration file. The default location is **/etc/grafana/grafana.ini** on Linux

**sudo vi /etc/grafana/grafana.ini**

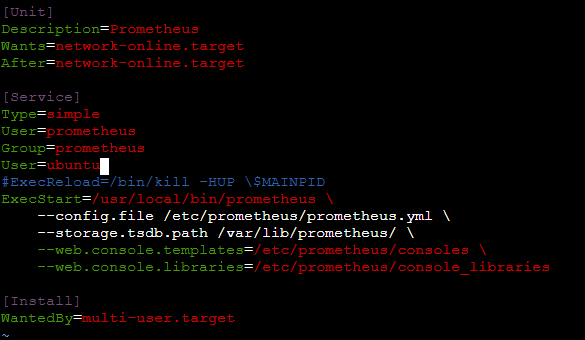


2. Restart the service after changing the ini file  
   
 **sudo systemctl restart grafana-server**

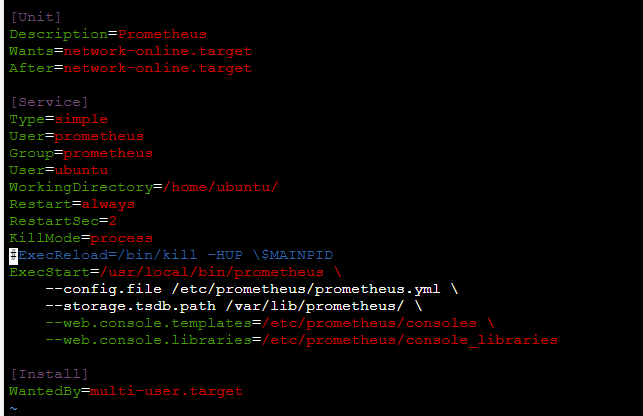
**Step3:- Prometheus as a Service file to server restart service automatically**

Now we want to run the Prometheus as a Service so that in case of server restart service will come automatically**.**

Let’s create a file **/etc/systemd/system/prometheus.service** with the below content:

  
  
**NOTE:- Getting error like failed the prometheous server when stop and start or screen once locked**

**So, we added below script**



**Step3: -** **Change the ownerships**

Change the ownership of all folders and files which we have created to the user which we have created in the first step

sudo chown prometheus:prometheus /etc/prometheus  
 sudo chown prometheus:prometheus /usr/local/bin/prometheus  
 sudo chown prometheus:prometheus /usr/local/bin/promtool

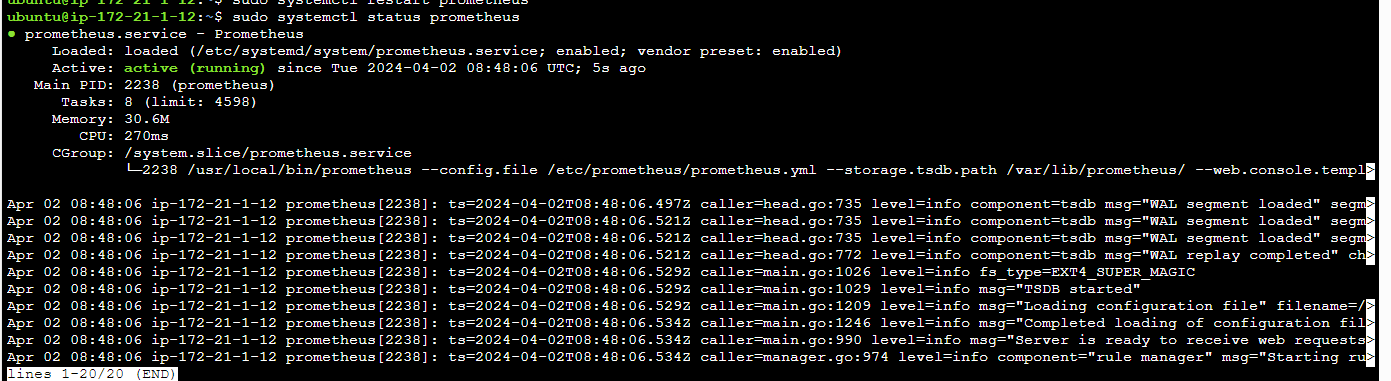
sudo chown -R prometheus:prometheus /etc/prometheus/console\_libraries  
 sudo chown -R prometheus:prometheus /var/lib/prometheus

**Step4:-** **configure the service and start and run continiously**

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

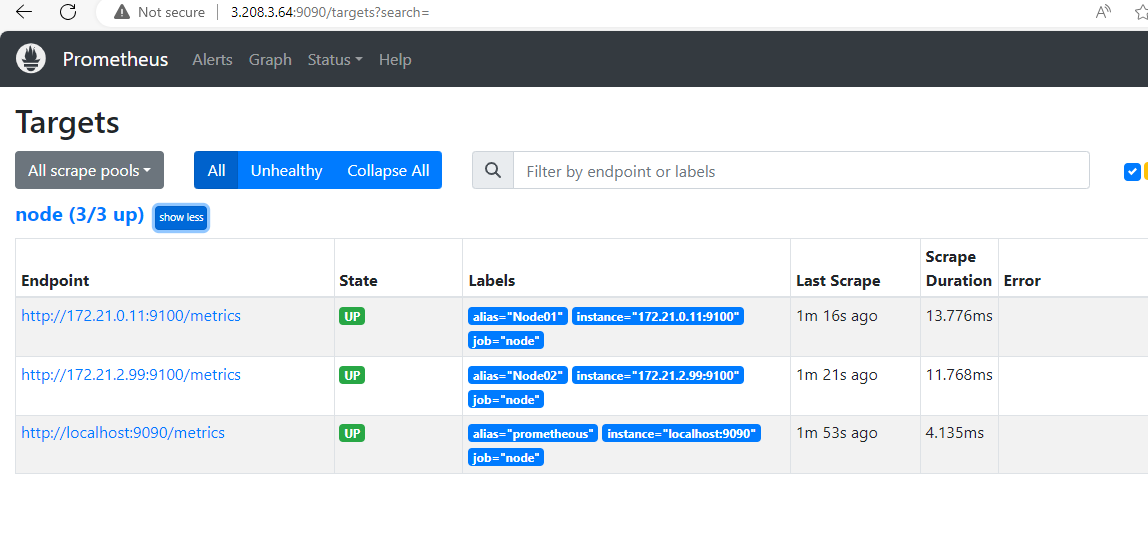
**Step5:- Checking the service and status of prometheous**

sudo systemctl status prometheus

  
**Now open it on the browser using below url:**

Type in browser **<ipaddress>:9090** to get the prometheous dashboard

Its show the prometheous configured successfully if **state=up**



**Node Exporter: -**

The node exporter is an open-source technology which enables you to measure various machine resources such as memory, disk and CPU utilization.

The Node Exporter is an agent that gathers system metrics and exposes them in a format which can be ingested by Prometheus. The Node Exporter is a project that is maintained through the Prometheus project.

**“To monitor your servers, you need to install the node exporter on all your target machines, which is like a monitoring agent on all the servers.”**

### **Step1: - Install Node Exporter**

**A) Create a node exporter user**

sudo useradd -rs /bin/false node\_exporter

B) Download **the Node Exporter, extract it and put it in /usr/local/bin folder**

Head to <https://prometheus.io/download/> and download the latest binary for the node exporter

(or)

wget https://github.com/prometheus/node\_exporter/releases/download/v0.18.1/node\_exporter-0.18.1.linux-amd64.tar.gz

tar xvzf node\_exporter-0.18.1.linux-amd64.tar.gz

sudo useradd -rs /bin/false node\_exporter

Sudo cp node\_exporter-0.18.1.linux-amd64/node\_exporter /usr/local/bin

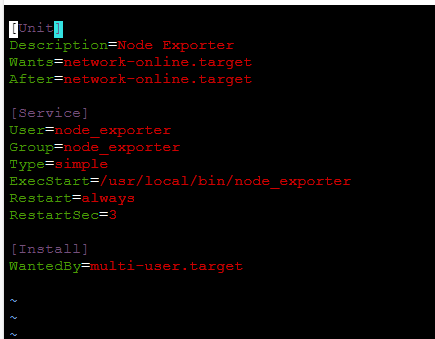
**Step2:- set the correct permissions to binary file**

Sudo chown node\_exporter:node\_exporter /usr/local/bin/node\_exporter

**Step3:- create a Node\_exporter new service file**

**Navigate to /etc/systemd/system**

cd /etc/systemd/system

sudo vi node\_exporter.service  
 

**Step4:-** **configure the service and start and run continiously**

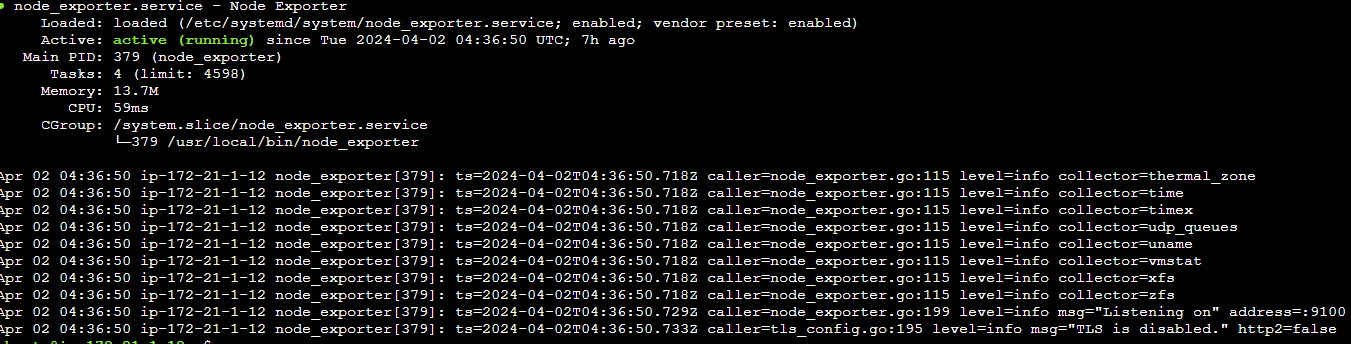
sudo systemctl daemon-reload

sudo systemctl start node\_exporter

sudo systemctl enable node\_exporter

**Step5:- Checking status of Node\_exporter and verifying**

sudo systemctl status node\_exporter.service

  
**Verify that your node exporter is correctly up and running with a simple curl command**

curl [**http://localhost:9100/metrics**](http://localhost:9100/metrics)

**Grafana: -**

Grafana is an open-source analytics and interactive visualization web application.

Grafana is a powerful tool for DevOps teams, helping to monitor, visualize, and understand the vast amount of data generated by their systems and applications. Here's what you need to know: Central Monitoring: Grafana consolidates data from various sources like Prometheus, Loki, and more into customizable dashboards.  
 Grafana is a tool used to analyze and visualize data. However, this data would have to be stored somewhere in order for Grafana to access and display it. These databases are what we refer to as data sources, and a Grafana datasource is simply any database from which it can pull data.

**Step1:- Install Grafana**

**A)Create a Grafana user**

sudo apt-get install -y adduser libfontconfig1

B)**Download the Grafana, extract**

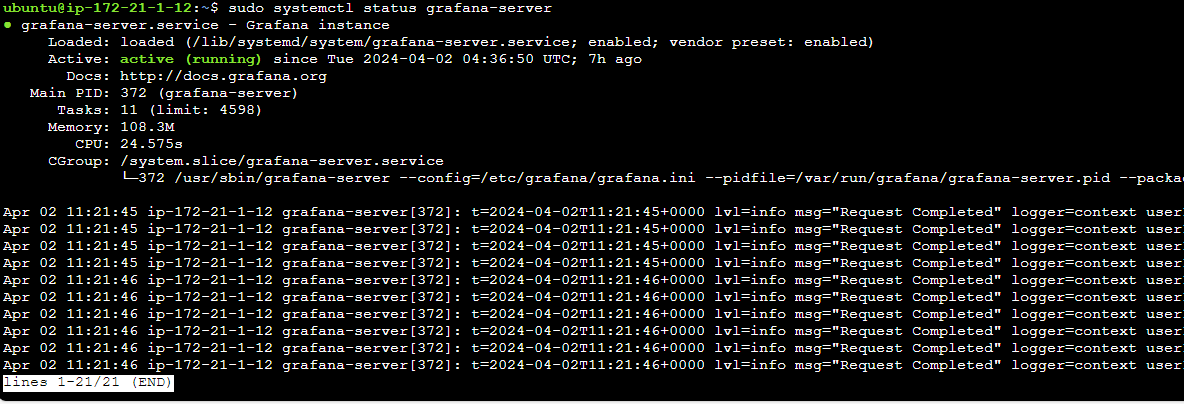
To install Grafana, head over to <https://grafana.com/grafana/download> and download the latest binaries available for you

wget [**https://dl.grafana.com/oss/release/grafana\_7.3.4\_amd64.deb**](https://dl.grafana.com/oss/release/grafana_7.3.4_amd64.deb)

sudo dpkg -i grafana\_7.3.4\_amd64.deb  
  
**Step2:-** **configure the service and start and run continiously**  
 sudo systemctl daemon-reload  
 sudo systemctl start grafana-server  
 sudo systemctl enable grafana-server.service

**Step3:- Checking status of Grafana-server**

sudo systemctl status grafana-server  
**NOTE: while checking the status we get error like failed means please refer the below link**  
**which was not installed properly packages**  
https://community.grafana.com/t/unable-to-install-grafana-from-apt-repository-on-debian-bookworm/119040

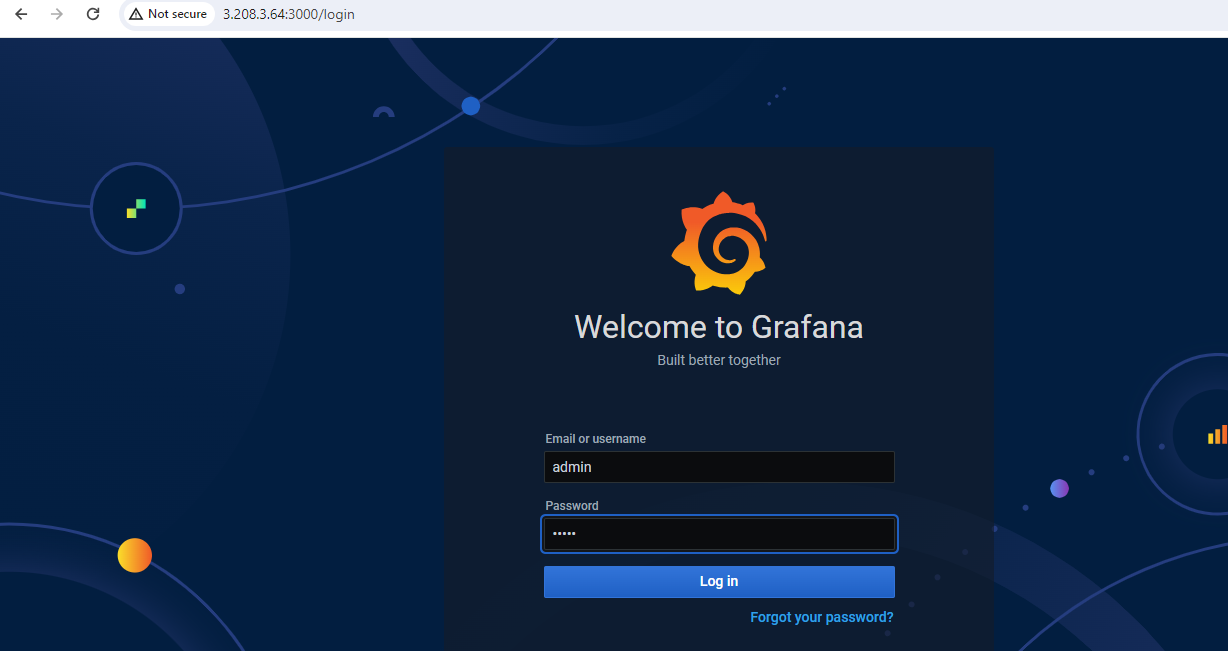
  
**step6: - Grafana Dashboard Login**

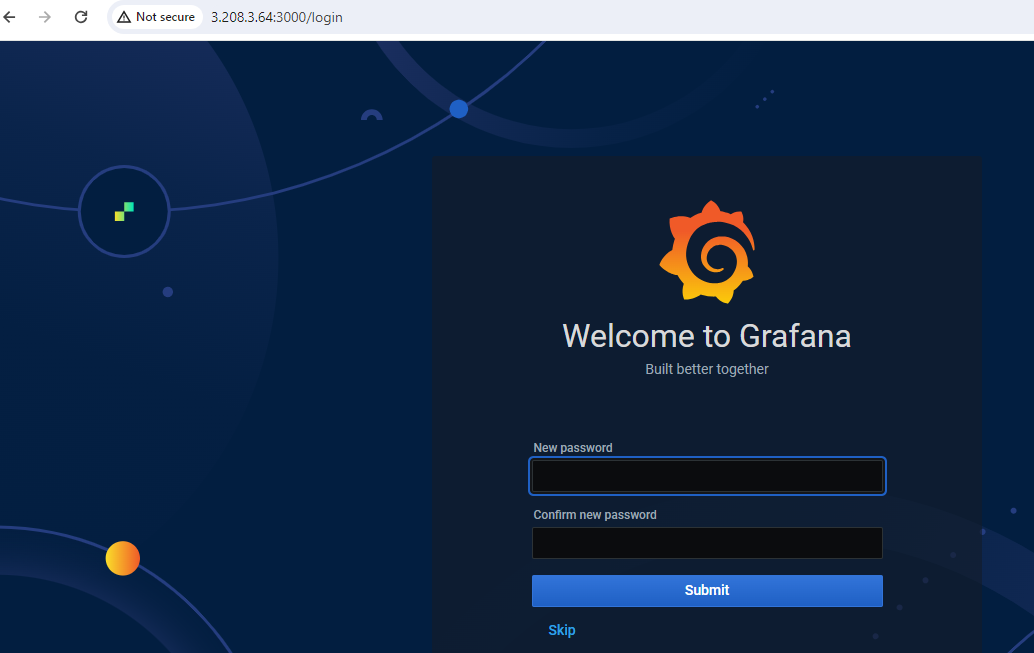
**Now open it on the browser using below url:**

[**http://publicip:3000**](http://youip:3000/)

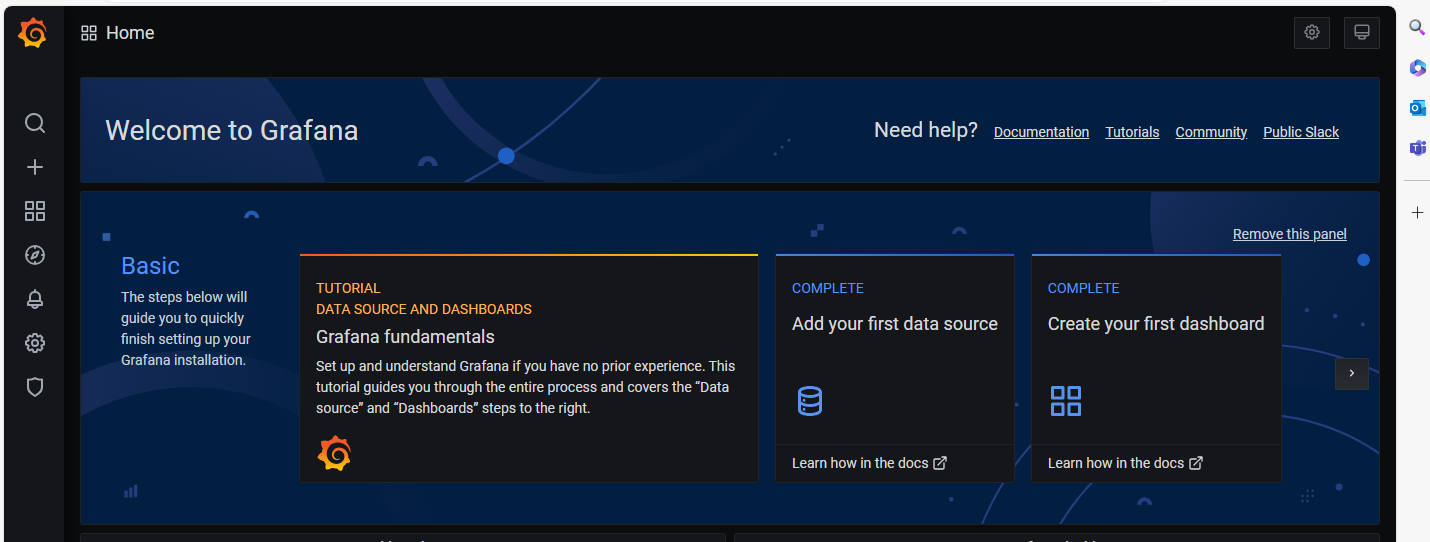
Login with default credentials username: **admin** and password **admin**

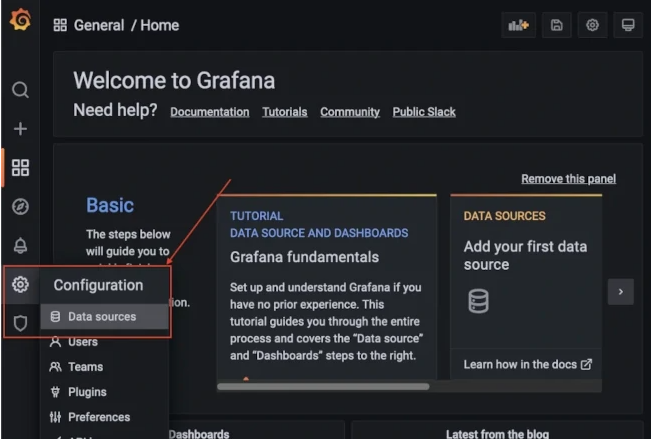
**NOTE: -** Make sure that port **3000** is open for this instance.

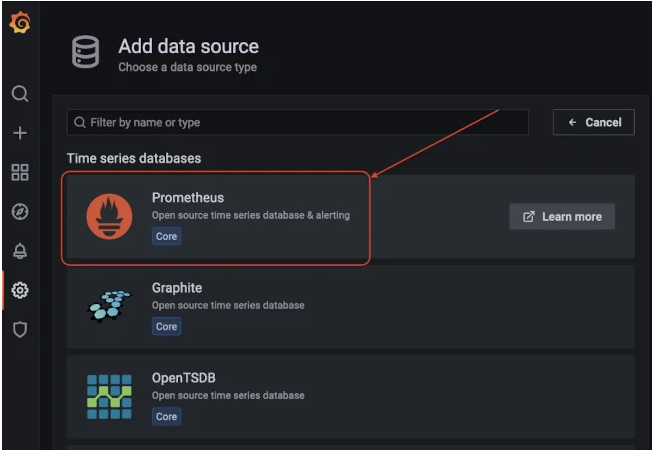
  
**After Login it will ask for Password changes**

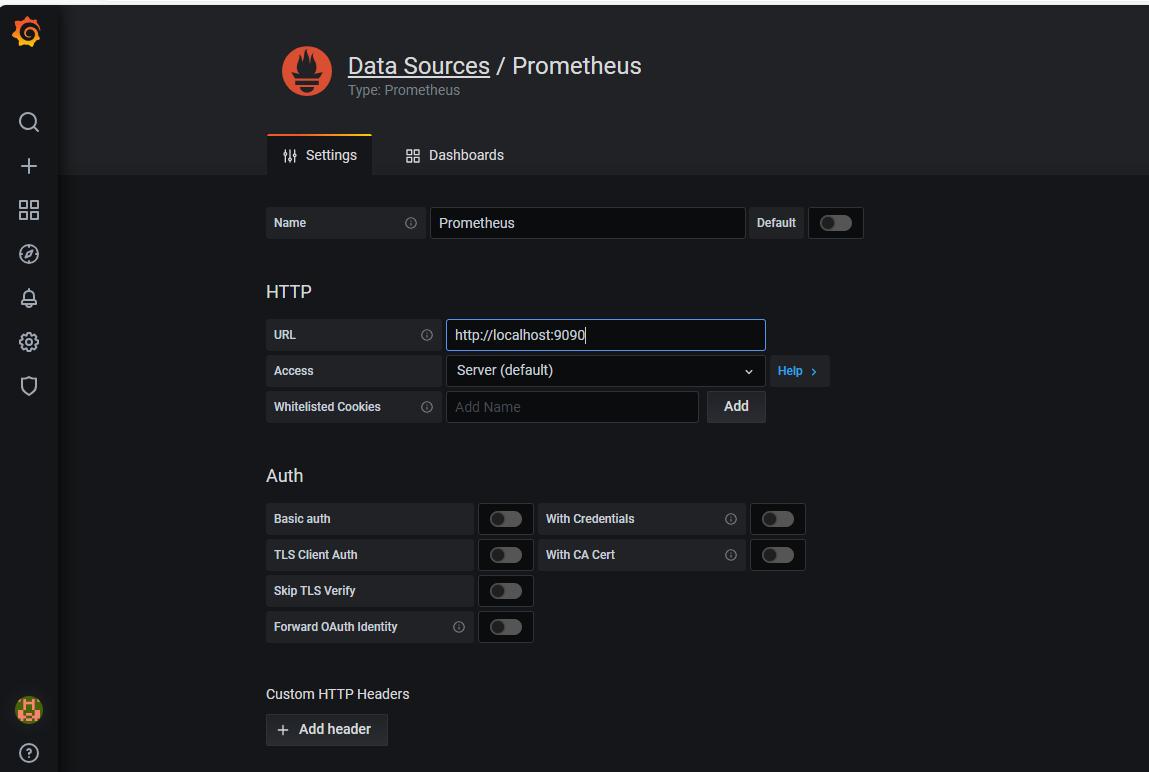


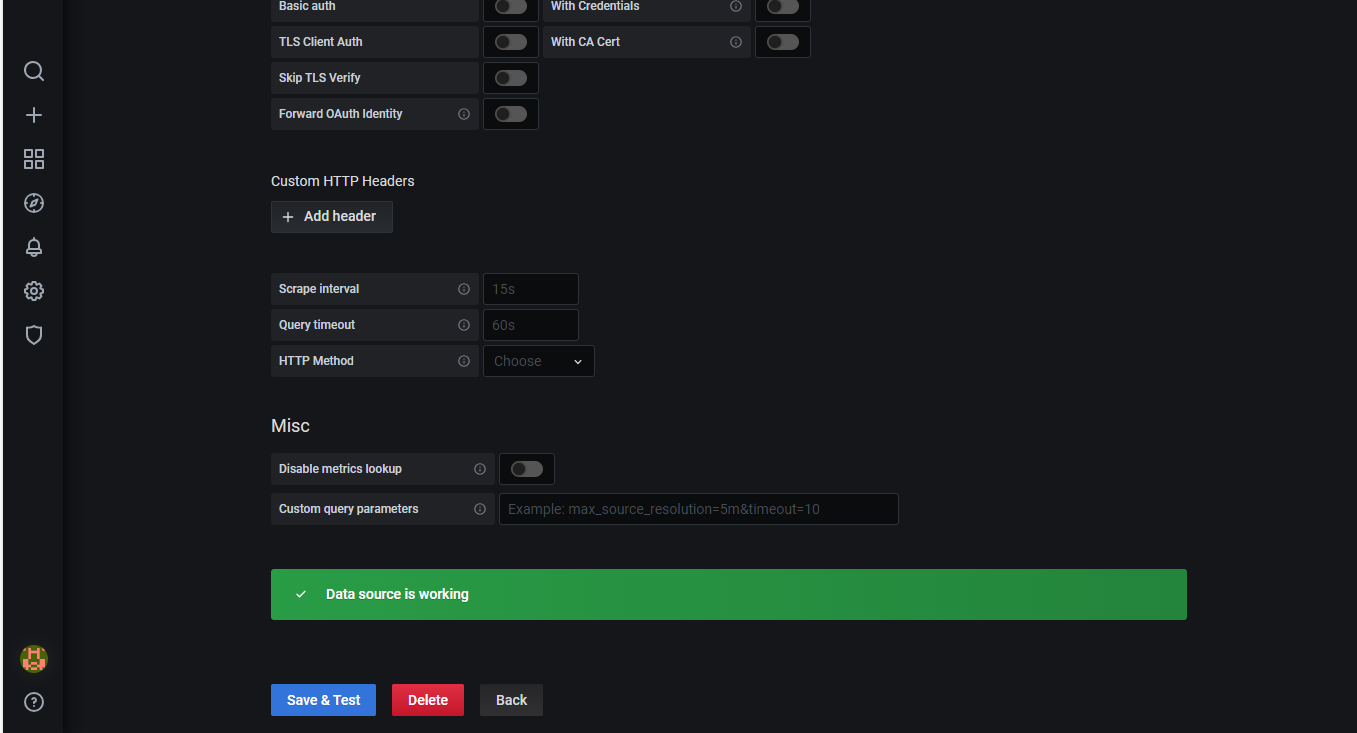
**A) Add Prometheus DataSource**

  
**Click on Setting ->datasources**

  
**Select the *Prometheus* as preferred data source -**

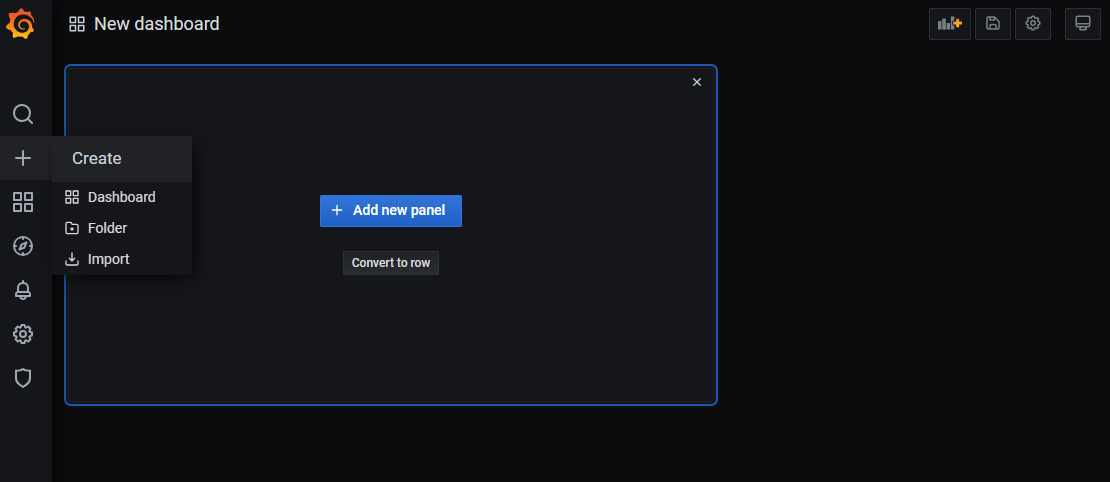
  
Enter the **hostname or IP address** of the prometheus server

  
Click on **SAVE & TEST**



B) **Importing the dashboard**

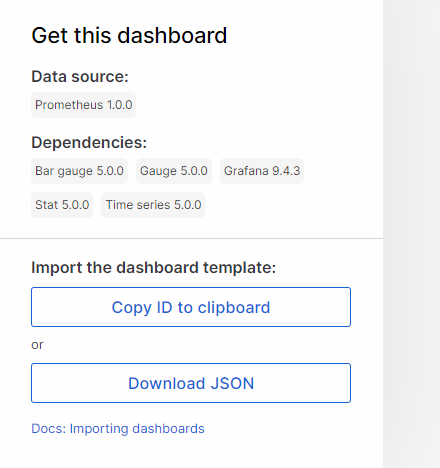
* **CLICK on add(+) symbol to import or create dashboards**



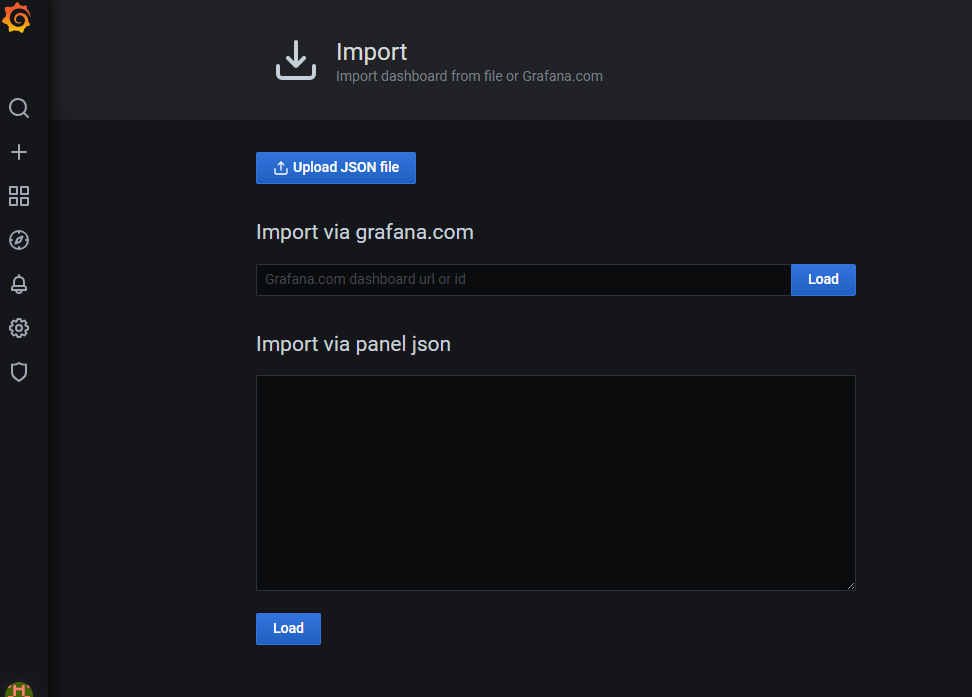
* **Import Grafana Dashboard from Grafana Labs**

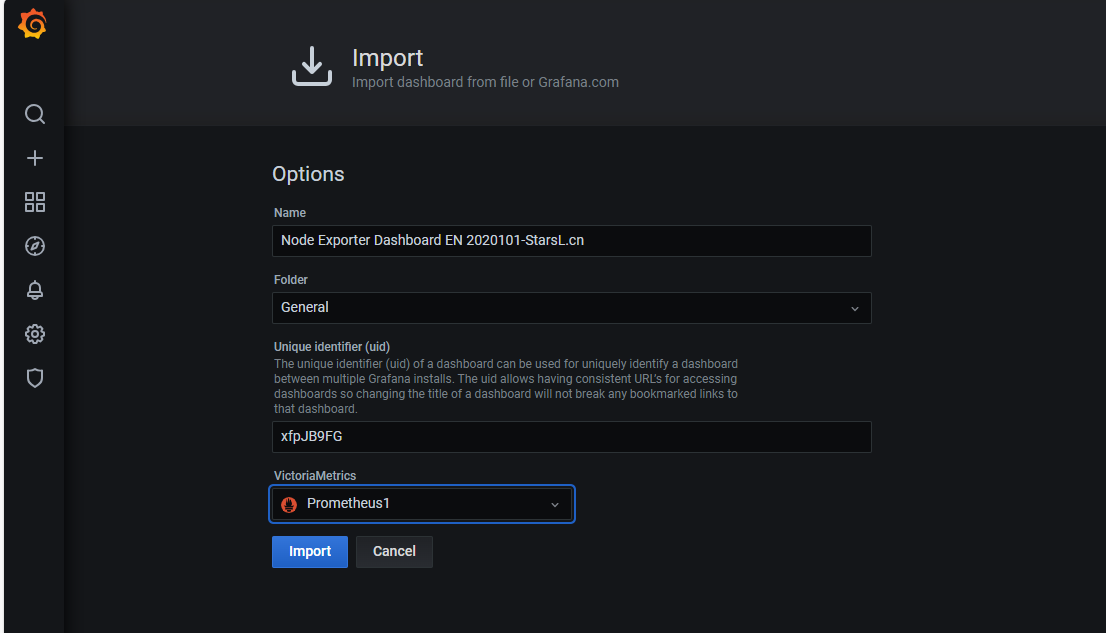
Now after settings the data source we can import pre-existing opensource dashboard from [Grafana Labs](https://grafana.com/grafana/dashboards/) using the Dashboard ID.

Goto [Grafana Dashboard](https://grafana.com/grafana/dashboards/) search some sample dashboard and download the json file

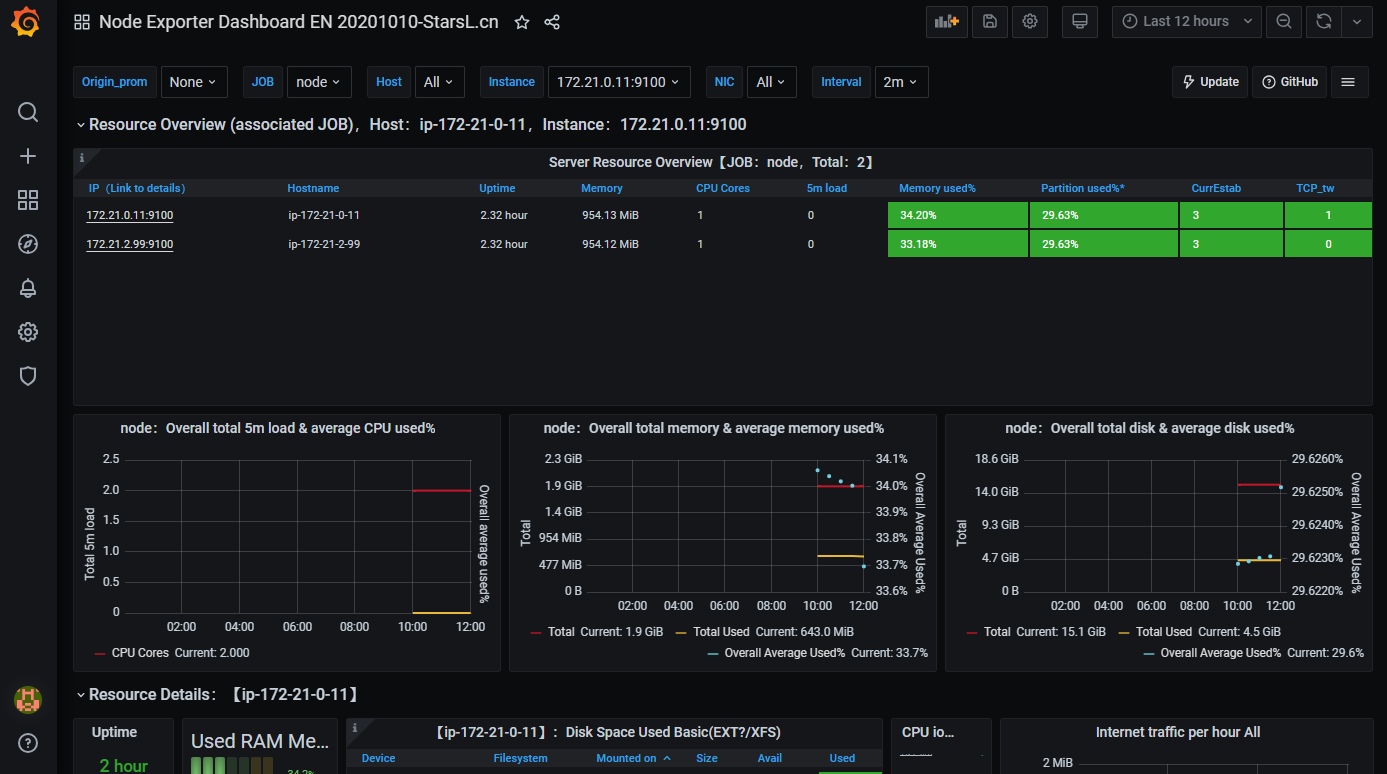


Import json file click on **upload json file**

  
Select the downloaded **Json** file and **DataSource**

  
**Click on import**

After import the dashboard it will automatically show cases the graphs



**Section :B Docker Metrics Monitoring: -**

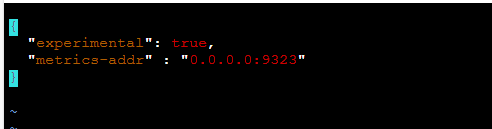
**Integrating Docker with Prometheus and Grafana**

**Step1:-Install Node\_exporter**

To Install the Node\_exporter follow the Above steps

**Step2:- enable daemon metrics**

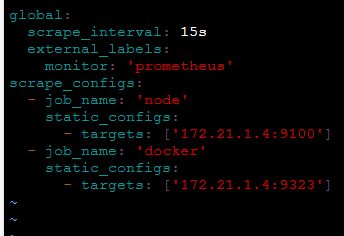
* Specify the metrics-address in the daemon.json configuration file. This daemon expects the file to be located at one of the following locations by default. If the file doesn’t exist, create it. **/etc/docker/daemon.json**
* Add the following configuration



Docker now exposes Prometheus-compatible metrics on port 9323.

**Step3: configure Prometheus to monitor itself using yaml file. Create a prometheus.yml file at /etc/prometheus/prometheus.yml with the below content**

NOTE: -We can create a special job to handle any kind of metrics.



**NOTE:** After updating the Prometheus YAML configuration file, it's necessary to restart the Prometheus service for the changes to take effect.

sudo systemctl restart prometheus

### 

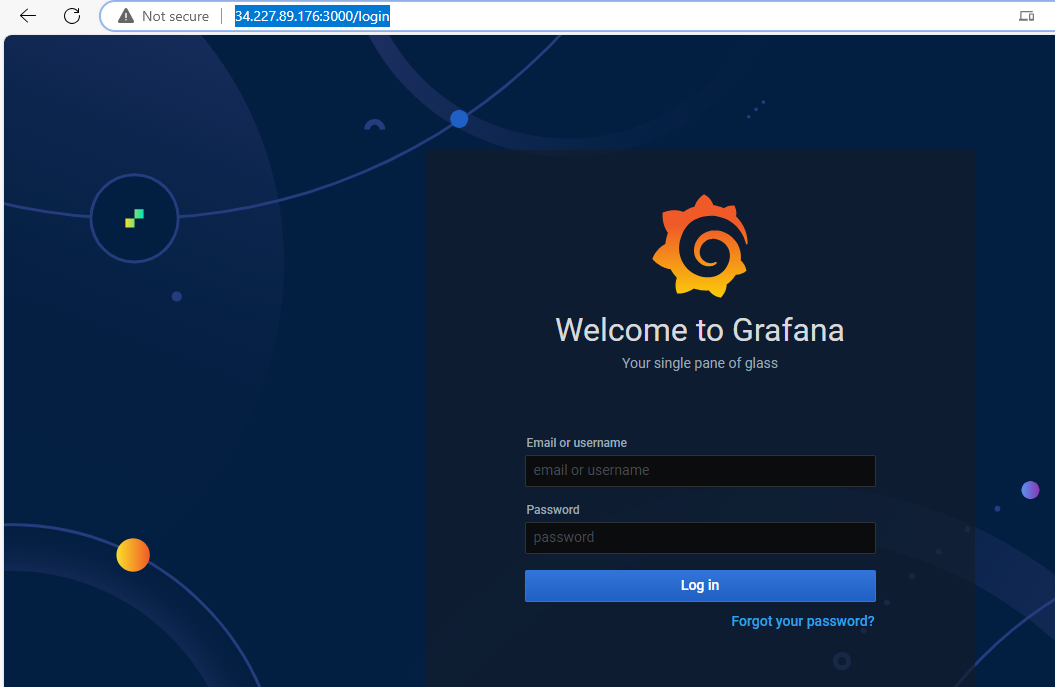
### **Step 4: Grafana Dashboard for Docker Metric**

### **Add Prometheus as a Data Source**

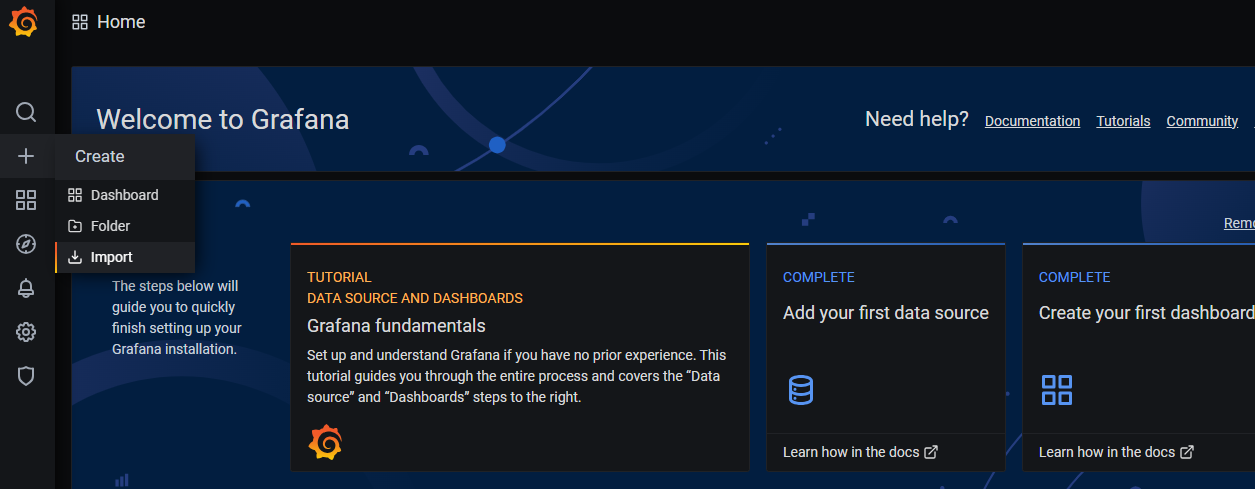
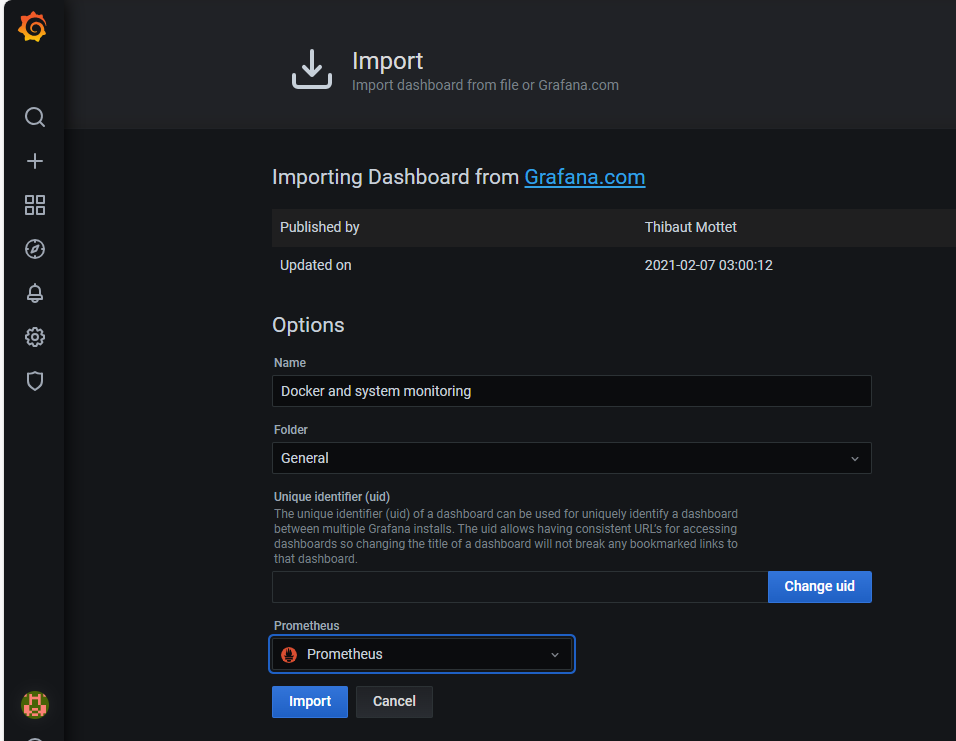
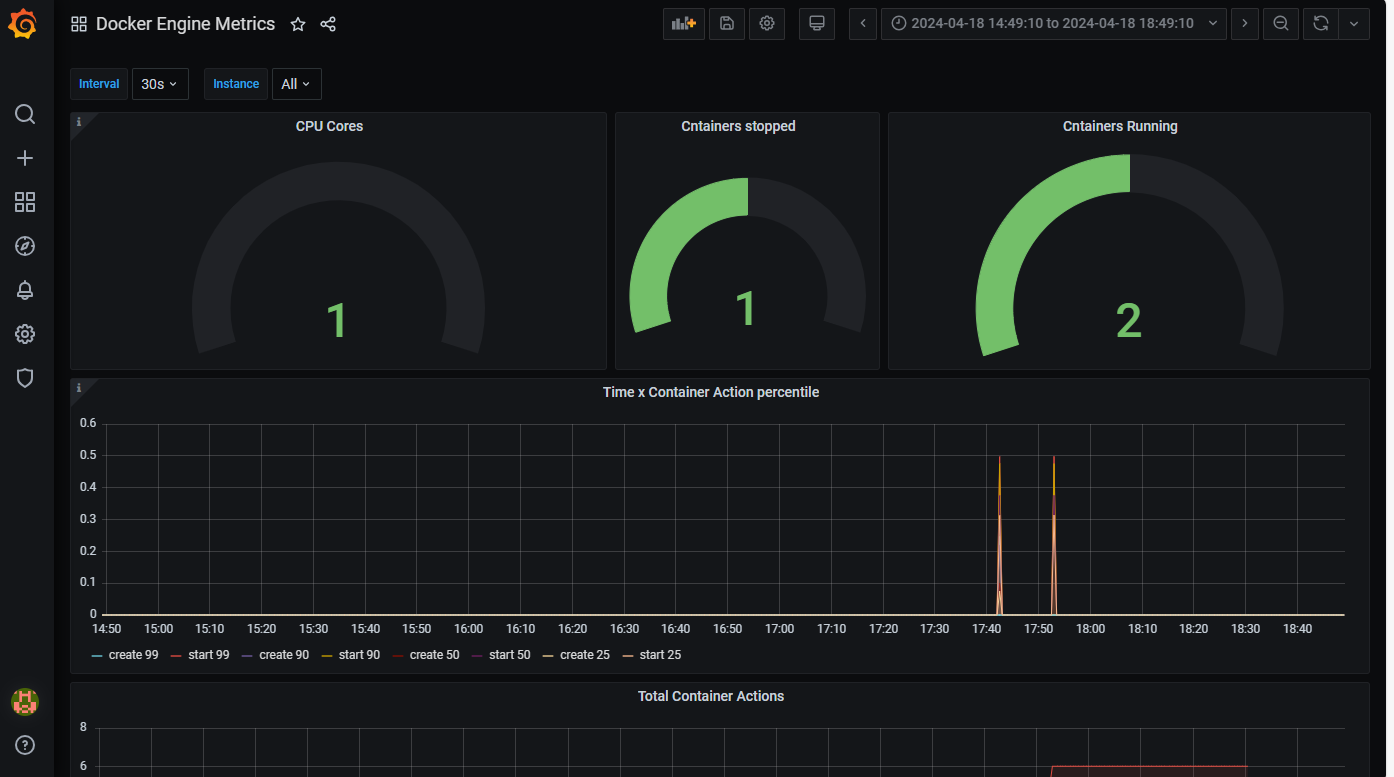
We need to add Prometheus as the data source in Grafana. Go to **Connections > Data Sources** and click the button “**Add new data source**”. Then we need to enter the IP address of the Prometheus server and it port (default 9090) in the URL

* **Login to Grafana Dashboard**

**Username:-admin**  
 **Password:-admin**



* **Import Dashboards**  
   **Docker Dashboard**: Imported ID 1229

**Section :c Jenkins Metrics Monitoring:**

**Integrating Jenkins with Prometheus and Grafana**

### **Step 1: Install Node Exporter**

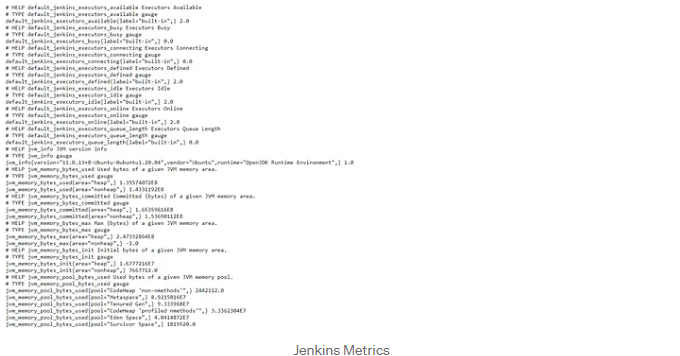
Follow the steps outlined previously to install Node Exporter. Ensure that Node Exporter is running on the Jenkins server to collect system metrics.

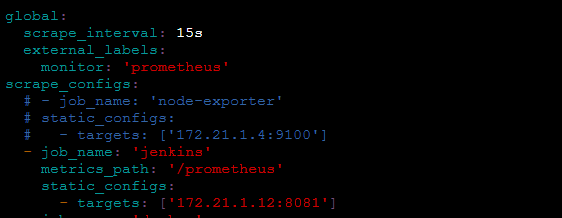
**Step 2:-Install Prometheus plugin in Jenkins**

* We need to install the plugin of **Prometheus** in Jenkins so that Prometheus can gather all the metrics of Jenkins
* In Jenkins click on manage plugin and search for **Prometheus metrics plugin** and click on install

  
 Prometheus Metrics Plugin

* The default path for Jenkins metrics is **<Public-IP:8080/prometheus>**

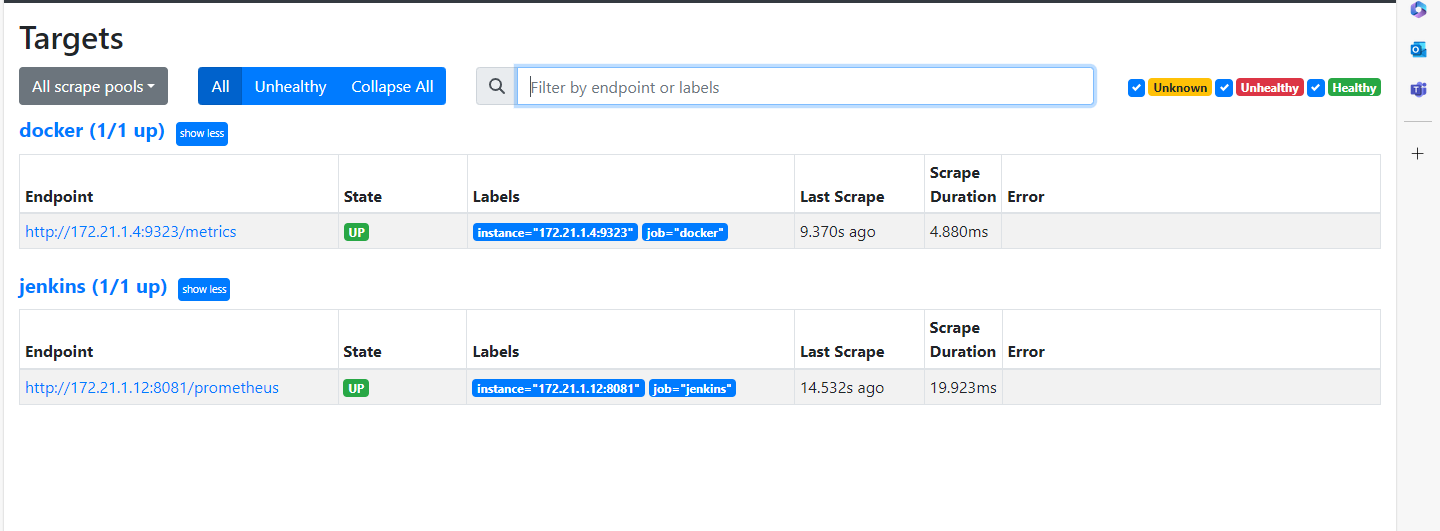


**Step 2:-** In order for Prometheus to gather the metrics we need to define below code in **prometheus.yml** under the **scrape\_configs** 

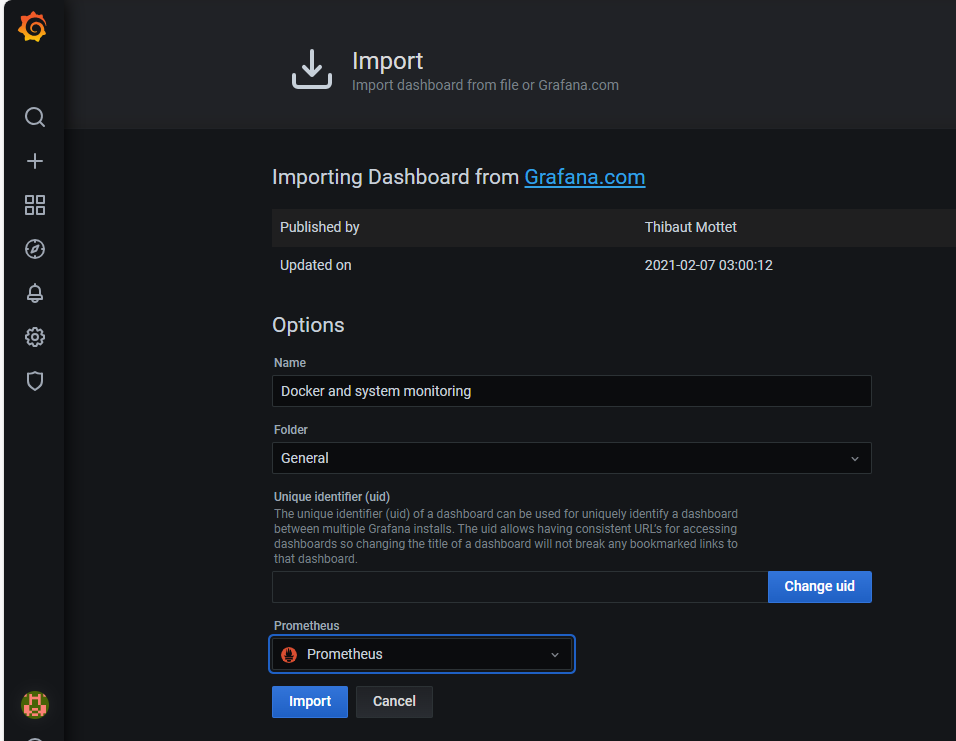
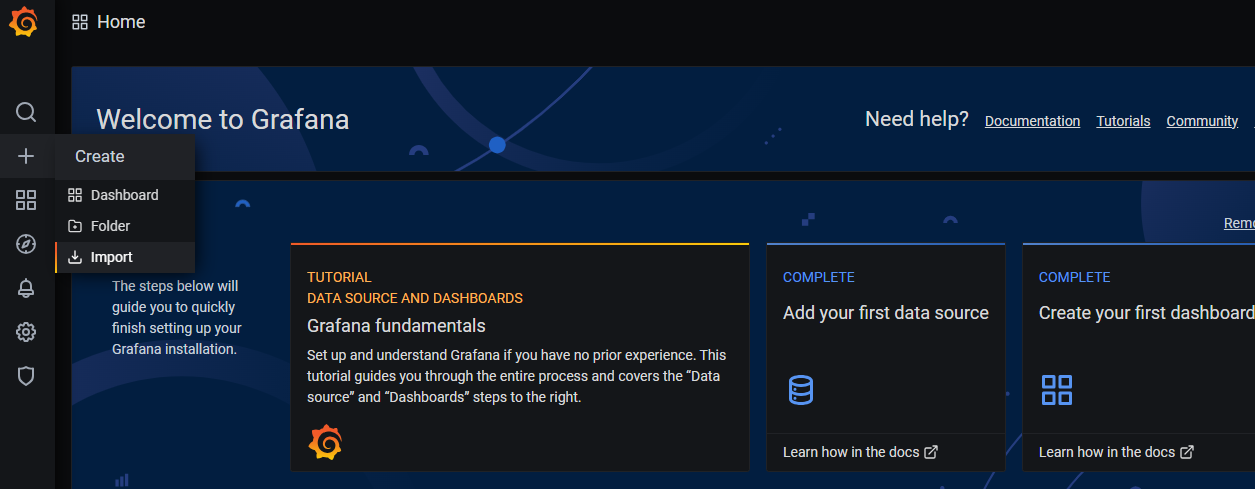
* ONCE YML FILE CHANGES MADE RUN RESTART COMMAND  
   Sudo systemctl restart prometheous

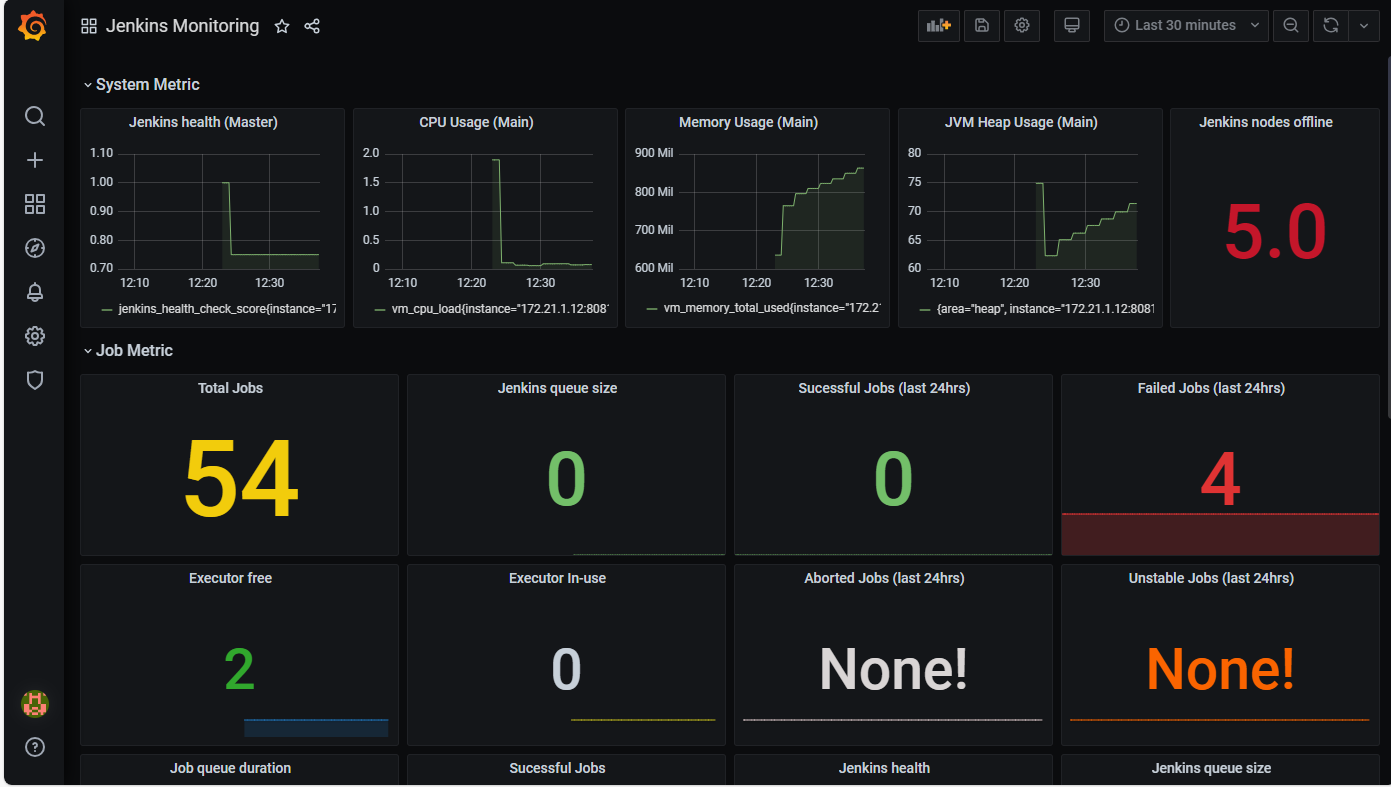
**Step 2:- Verify Jenkins Port Availability**

To ensure that the Jenkins port is up and running, you can check the Prometheus dashboard. Look for metrics related to Jenkins or the specific port Jenkins is running on to confirm its availability and status.



**Step 3:-Create a Jenkins Dashboard**

* A) **Import Dashboard**  **Select Data Source**
* Ensure that you select Prometheus as the data source while importing the Jenkins dashboard, consistent with the previously configured Prometheus setup.



### 