**Monitoring Tool Project**

|  |  |
| --- | --- |
| **Author:** | **Signature/ Date:** |
| Rabindra Nath Moharana |  |
| **Approver:** | **Signature/ Date:** |
|  |  |

List of reviewers:

|  |  |  |
| --- | --- | --- |
| **Name of Reviewer** | **Role / Function** | **Date of Review** |
|  |  |  |

Table of Contents:

Brief about the tools………………………………………………………………………03

Prerequisites ………………………………………………………………………………….04

Process details……………………………………………………………………………04-21

Conclusion………………………………………………………………………………………21

**What is Prometheus ?**

* Prometheus is a open source Linux Server Monitoring tool mainly used for metrics monitoring, event monitoring, alert management, etc.
* Prometheus has changed the way of monitoring systems and that is why it has become the Top-Level project of Cloud Native Computing Foundation (CNCF).
* Prometheus uses a powerful query language i.e. “PromQL”.
* In Prometheus tabs are on and handles hundreds of services and microservices.
* Prometheus use multiple modes used for graphing and dashboarding support.
* As above we can see an architecture of Prometheus monitoring tool.
* We made a basic design to understand it easily for you people.

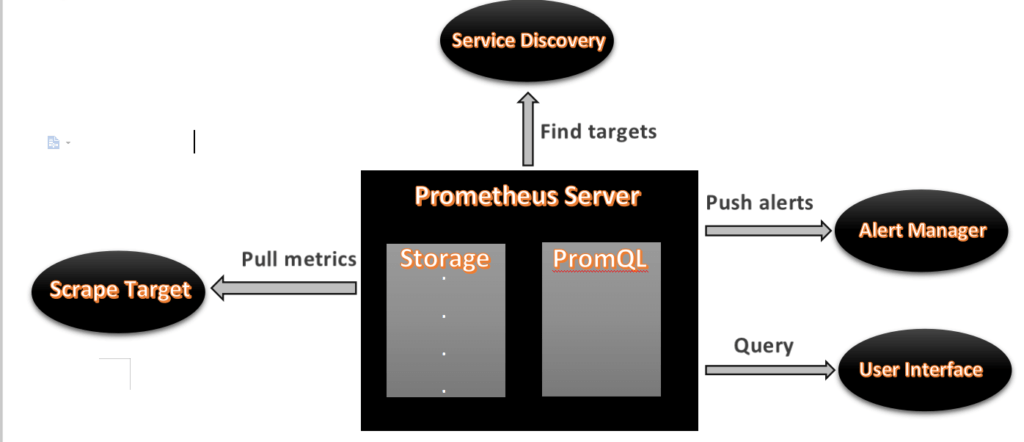
**What is Grafana ?**

* Grafana is a free and open source visualization tool mostly used with Prometheus to which monitor metrics.
* Grafana provides various dashboards, charts, graphs, alerts for the particular data source.
* Grafana allows us to query, visualize, explore metrics and set alerts for the data source which can be a system, server, nodes, cluster, etc.
* We can also create our own dynamic dashboard for visualization and monitoring.
* We can save the dashboard and can even share with our team members which is one of the main advantage of Grafana.

**What is Node Exporter ?**

* Node exporter is one of the Prometheus exporters which is used to expose servers or system OS metrics.
* With the help of Node exporter we can expose various resources of the system like RAM, CPU utilization, Memory Utilization, disk space.
* Node exporter runs as a system service which gathers the metrics of your system and that gathered metrics is displayed with the help of Grafana visualization tool.

## ****Prometheus Architecture :-****



**Prerequisites :-**

* Ubuntu with 20.04 Version
* Root user account with **sudo** privilege.
* Prometheus system user and group.
* Sufficient storage on your system and good internet connectivity.
* Ports Required- **9090 (Prometheus), 3000 (Grafana), 9100 (Node Exporter)**

We will update the system repository index by using the following command.

* **sudo apt update –y**

**switch to root user**

* **sudo su –**

**Export the release of Prometheus  
- export RELEASE="2.2.1"**

## ****#1. Creating Prometheus System Users and Directory****

We will have to create a Prometheus user named Prometheus and a Prometheus directory named as Prometheus.

Using below commands we can create a user and directory.

* **sudo useradd --no-create-home --shell /bin/false Prometheus**
* **sudo useradd --no-create-home --shell /bin/false node\_exporter**
* **sudo mkdir /etc/Prometheus**
* **sudo mkdir /var/lib/Prometheus**

## ****#2. Update Prometheus user****

As user groups and directories are created successfully which store the Prometheus data and files.

Now we will have to update the group and user ownership on the newly created directories.

By using the below command we update the ownership.

* sudo chown prometheus:prometheus /etc/Prometheus
* sudo chown prometheus:prometheus /var/lib/Prometheus

## ****#3. Download Prometheus Binary File****

Now we will download the latest version of Prometheus. We can copy the download link as per our Operating System

Using below command we can download Prometheus, here we are downloading Prometheus 2.26 version, you use above link to download specific version.

Navigate to **/opt** directory

* cd /opt/

Download the Prometheus setup using wget

* Wget <https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz>

Now we have successfully downloaded the Prometheus file and now we will extract that file.

## ****#4.Install Prometheus and Grafana on Ubuntu 20.04 LTS****

We can use sha256sum command line to generate a checksum of the Prometheus downloaded file.

We will also extract the downloaded file using the **tar** command.

* sha256sum prometheus-2.26.0.linux-amd64.tar.gz

**Output:**

f1f2eeabbf7822572dce67565dc96ffaa2dd1897dd1d844562552b11123f151a prometheus-2.26.0.linux-amd64.tar.gz

We have verify that the output from above command with checksum matches the sha256sum checksum which is on official Prometheus download page.

It will ensure that our downloaded file is not a corrupted file.

Now we will extract the Prometheus setup file using the following commands.

* tar -xvf prometheus-2.26.0.linux-amd64.tar.gz

navigate to prometheus extracted folder

- cd prometheus-2.26.0.linux-amd64

To check list of setup files

- ls

## ****#5. Copy Prometheus Binary files****

Now we have two libraries in our directory i.e. Prometheus and promtool. We will have to copy that both libraries to our /usr/local/bin directory.

By following below commands we will perform the copy operation.

* sudo cp /opt/prometheus-2.26.0.linux-amd64/prometheus /usr/local/bin/
* sudo cp /opt/prometheus-2.26.0.linux-amd64/promtool /usr/local/bin/

## #6. Update Prometheus****user ownership on Binaries****

Now we will update the user and group ownership on the binaries of Prometheus.

Using following commands we will update the user and group ownership.

* sudo chown prometheus:prometheus /usr/local/bin/Prometheus
* sudo chown prometheus:prometheus /usr/local/bin/promtool

## ****#7. Copy Prometheus Console Libraries****

We will need to copy the console and console\_libraries directories to /etc/Prometheus/.

Use below commands to copy console and console\_libraries.

* sudo cp -r /opt/prometheus-2.26.0.linux-amd64/consoles /etc/Prometheus
* sudo cp -r /opt/prometheus-2.26.0.linux-amd64/console\_libraries /etc/Prometheus
* sudo cp -r /opt/prometheus-2.26.0.linux-amd64/prometheus.yml /etc/Prometheus

## ****#8. Update Prometheus ownership on Directories****

Now we will update the user and group ownership on the directories  to Prometheus user using -R.

By executing this commands ownership is set on is ensured. Execute the following commands.

* sudo chown -R prometheus:prometheus /etc/prometheus/consoles
* sudo chown -R prometheus:prometheus /etc/prometheus/console\_libraries
* sudo chown -R prometheus:prometheus /etc/prometheus/prometheus.yml

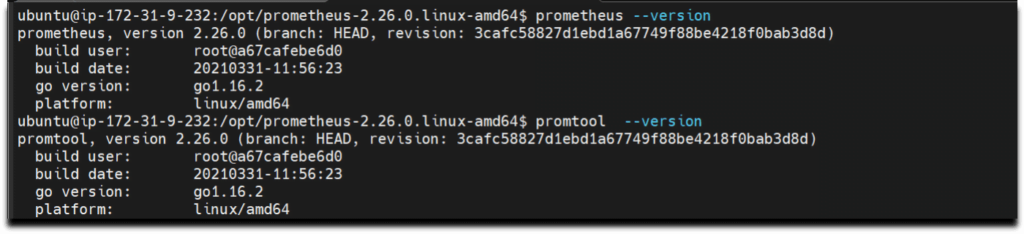
## ****#9. Check Prometheus Version****

Now the Prometheus is successfully installed on our system. We will check the version of Prometheus and promtool to configure it.

Follow the commands to verify prometheus version.

* prometheus –version
* promtool –version

**Output:**



## ****#10. Prometheus configuration file****

We have already copied **/opt/prometheus-2.26.0.linux-amd64/prometheus.yml** file **/etc/prometheus** directory, verify if it present and should look like below and modify it as per your requirement.

* cat /etc/prometheus/prometheus.yml

# my global config

global:

scrape\_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.

evaluation\_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.

# scrape\_timeout is set to the global default (10s).

# Alertmanager configuration

alerting:

alertmanagers:

- static\_configs:

- targets:

# - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation\_interval'.

rule\_files:

# - "first\_rules.yml"

# - "second\_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:

# Here it's Prometheus itself.

scrape\_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']

## #11. Creating Prometheus****Systemd file****

To run Prometheus as service we have to setting up prometheus, We will provide a path for both configuration file and data directory. We will start it with the Prometheus user using the following command.

**sudo -u prometheus /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**

**--web.listen-address=0.0.0.0:9090**

**Restart=always**

**RestartSec=10s**

Now we will create a system service file in **/etc/systemd/system** location.

* sudo nano /etc/systemd/system/prometheus.service

After creating file successfully, copy the below files and it to the newly created file. /etc/systemd/system/prometheus.service

[Unit]

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

After adding the program save the file with **Ctrl+O** and exit with **Ctrl+X**.

To use the newly created service we will have to reload the daemon services, Use the below command to reload daemon services.

* sudo systemctl daemon-reload

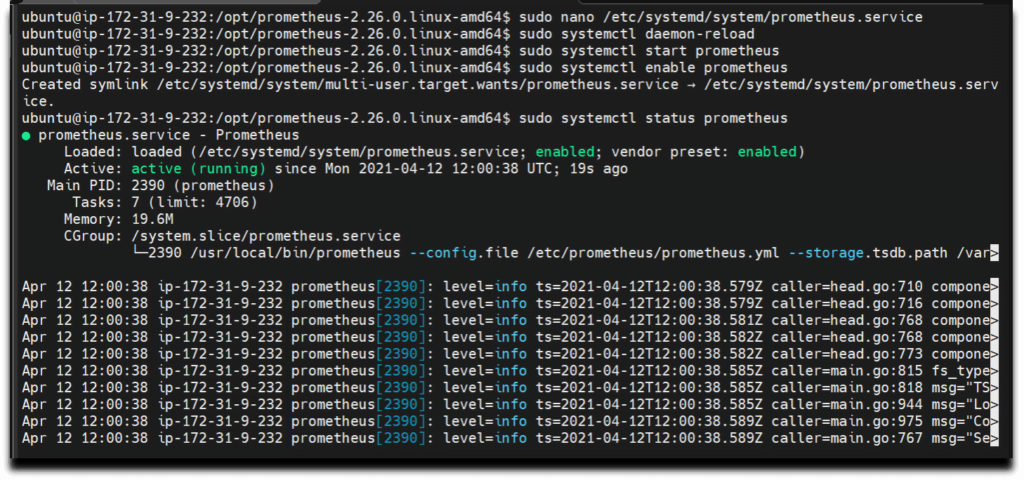
start and enable prometheus service using below commands

* sudo systemctl start Prometheus
* **sudo systemctl enable Prometheus**

We will check the Prometheus status weather it is running or not

* sudo systemctl status Prometheus

**Output :-**



Prometheus installation and configuration is set up, We can see status Active:  active(running)

## ****#12. Accessing Prometheus****

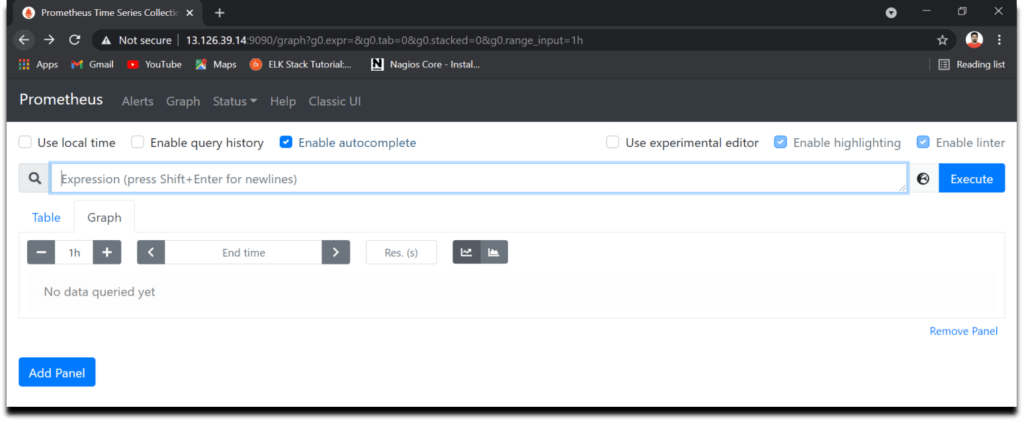
Now as Prometheus installation and configuration is set up and it is ready to use we can access  its services via web interface.Also check weather port 9090 is UP in firewall.

Use below command to enable prometheus service in firewall

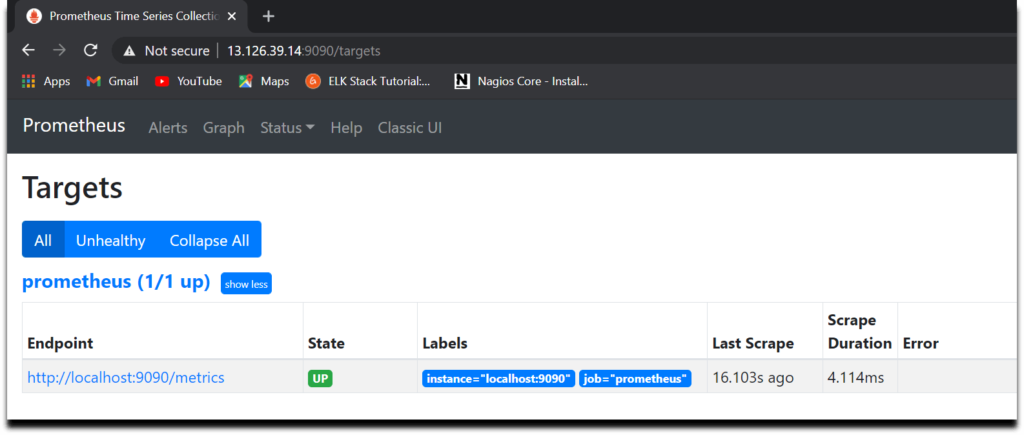
* sudo ufw allow 9090/tcp

Now Prometheus service is ready to run and we can access it from any web browser.

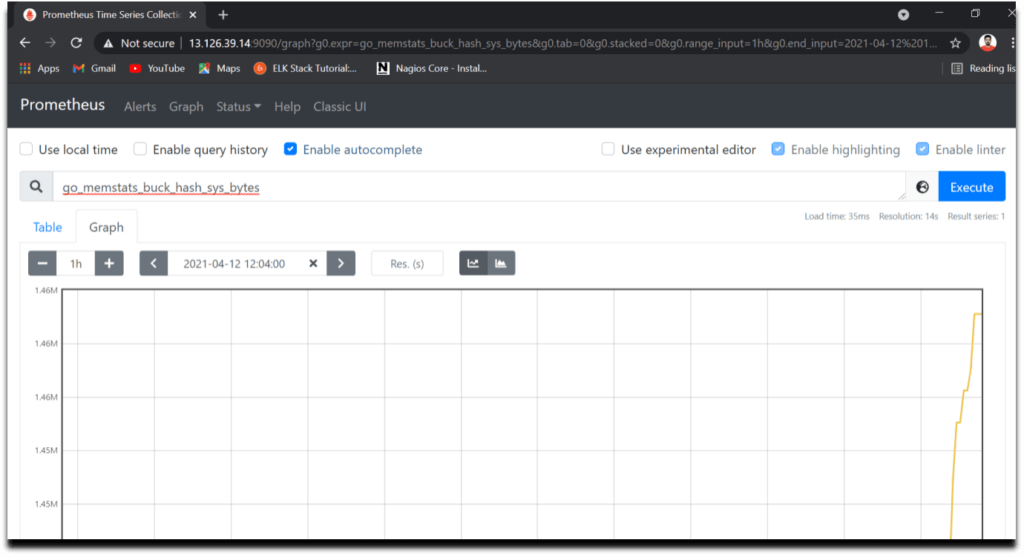
* <http://server-IP-or-Hostname:9090>.

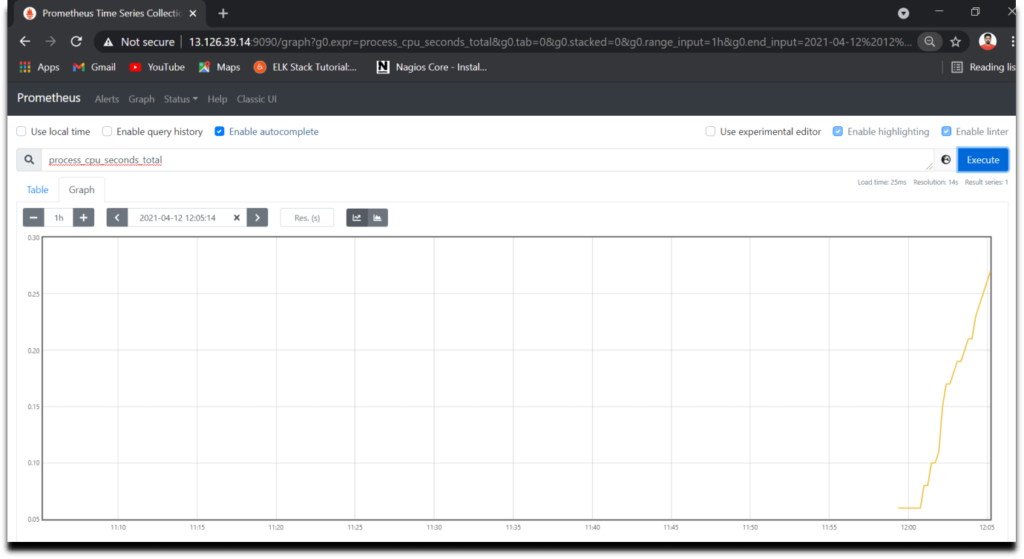


As we can see the Prometheus dashboards, we can also check the target.As we can observe Current state is UP and we can also see the last scrape.



Here are some snapshots of Prometheus dashboards and graphs.





## ****#13.****Install Grafana on Ubuntu 20.04 LTS using APT

Now lets Install Grafana for wonderful dashboards and data visualization for monitoring systems, servers, services, etc

We can install grafana on Ubuntu either by downloading .deb package from [Grafana Download page](https://grafana.com/docs/grafana/latest/installation/debian/) or using APT which is more easier.

Add the Grafana GPG key in Ubuntu using wget

* wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add –

Now add the Grafana repository in Ubuntu using APT

- echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

update the system packages to take effect

- sudo apt-get update

Now lets Install Grafana using below command

* sudo apt-get install grafana

Now start the Grafana service using below command

* sudo systemctl start grafana-server

Now finally enable the Grafana service which will automatically start the Grafana on boot

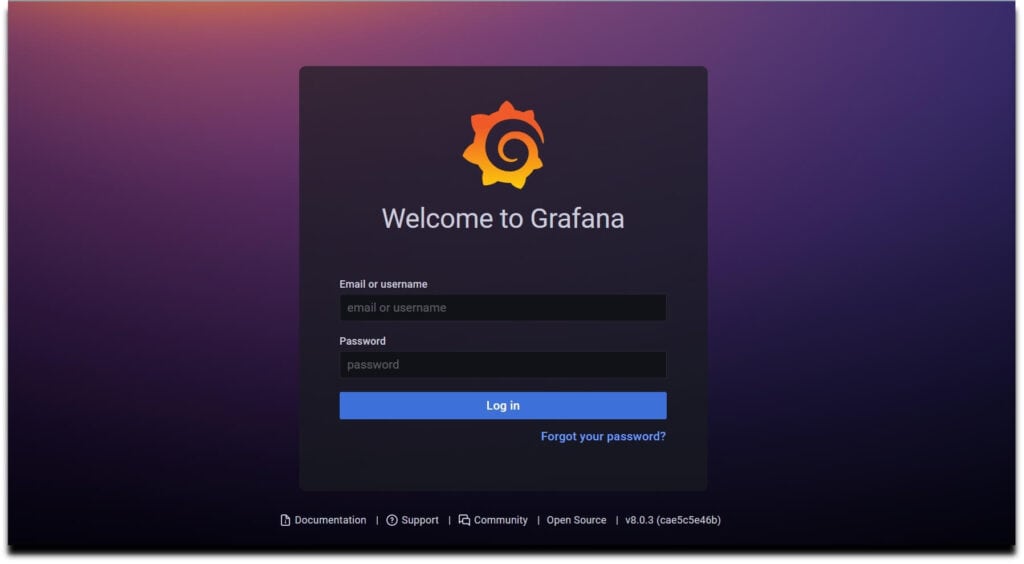
* sudo systemctl enable grafana-server.service

To access Grafana Dashboard open your favorite browser, type server IP or Name followed by grafana default port 3000.

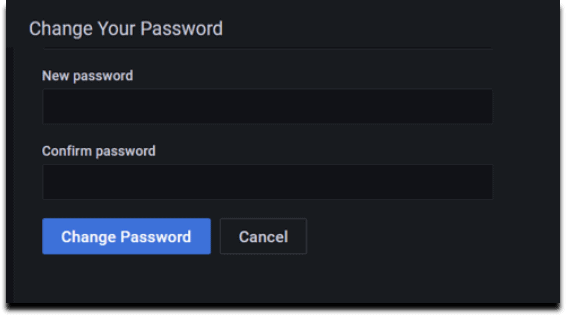
* <http://your_ip:3000>

Here you can see Login page of Grafana now you will have to login with below Grafana default UserName and Password.

* Username – admin
* Password – admin

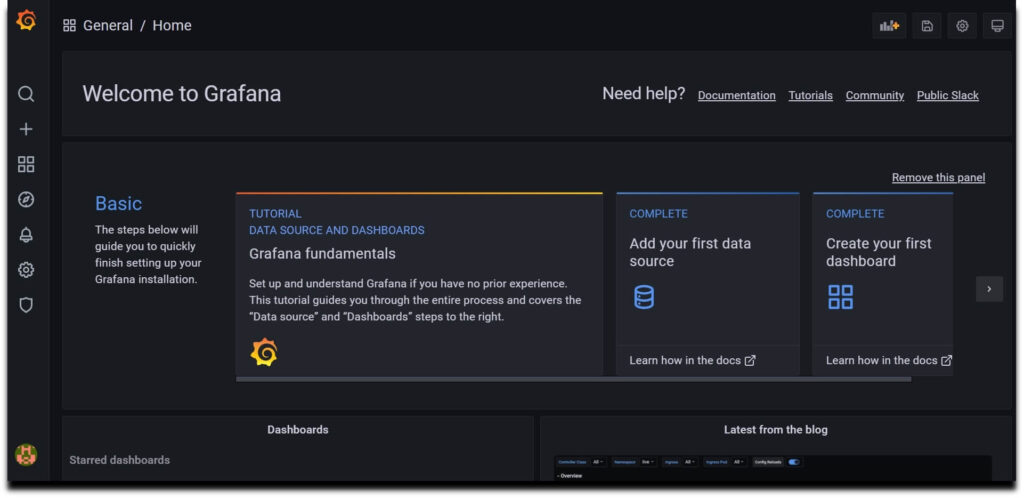


It is always a good practice to change your login credentials.



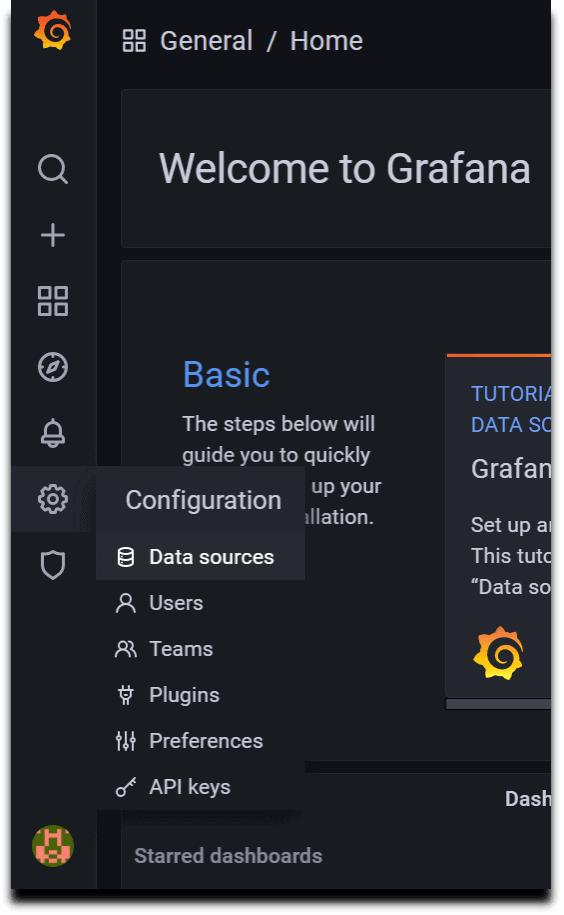
Provide your New Password and click on **Change Password**

Now here you can see Home Dashboard page of Grafana

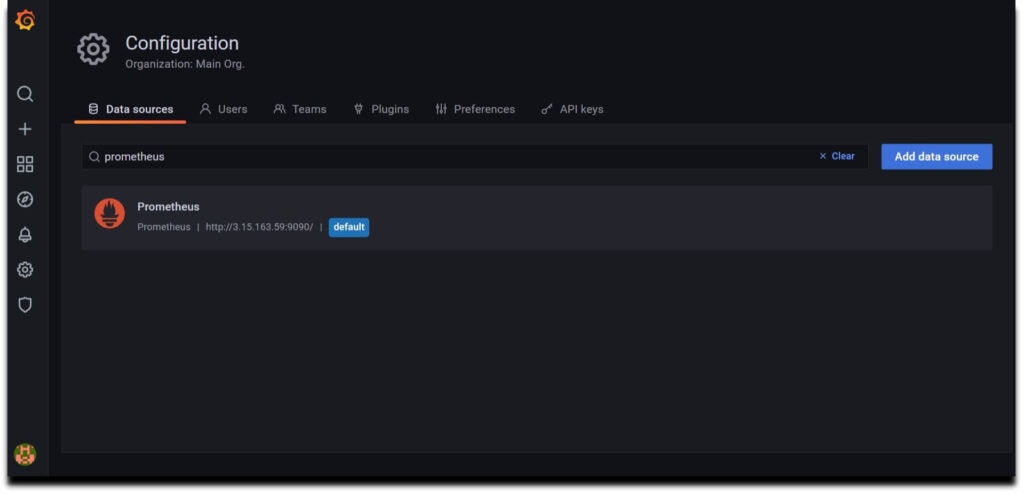


## #14. Configure Prometheus as Grafana DataSource

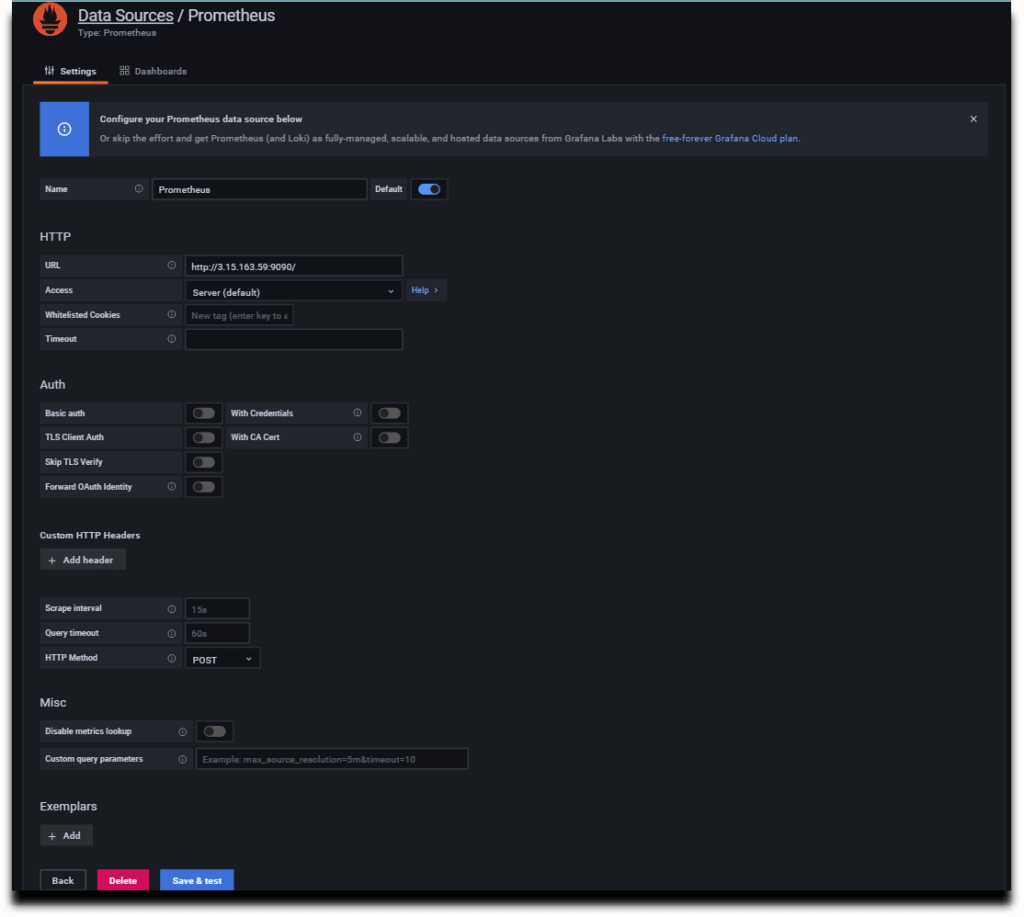
Once you logged into Grafana Now first Navigate to **Settings Icon ->> Configuration ->> data sources**



Now lets click on **Add Data sources**and select **Prometheus**

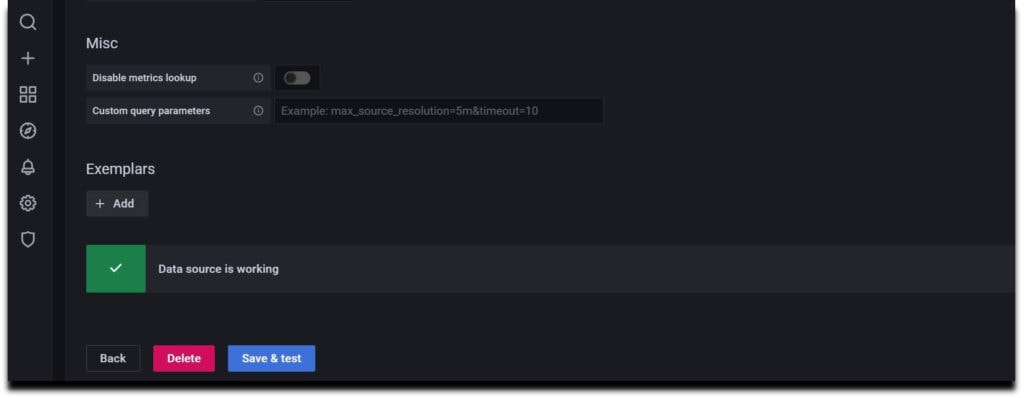


Now configure Prometheus data source by providing Prometheus URL



As per your requirement you can do other changes or you can also keep remaining configuration as default.

Now click on **Save & test**so it will prompt a message**Data Source is working**.

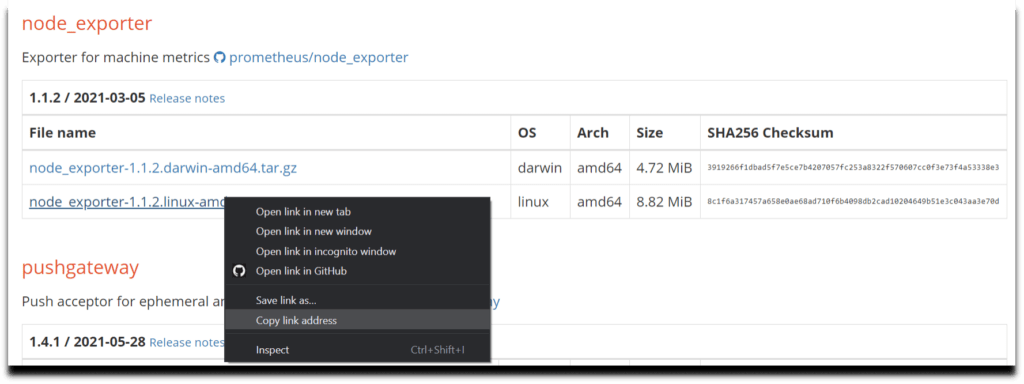


## #15. Install Node Exporter on Ubuntu 20.04 LTS

Node Exporter collects the metrics of your system such as Memory usage, CPU usage, RAM, disk space, etc.

To install Node Exporter first navigate to Prometheus official page, Scroll down and you will get **node\_exporter** section and then select Linux OS for amd64.

Now right click on node exporter and copy link address



Now lets run the copied URL with **wget** command

wget <https://github.com/prometheus/node_exporter/releases/download/v1.2.0/node_exporter-1.2.0.linux-amd64.tar.gz>

Unzip the downloaded the file using below command

* sudo tar xvzf node\_exporter-1.2.0.linux-amd64.tar.gz

Now do **ls**and your can see node\_exporter binary file.

Go to that file and move this file to your /usr/local/bin directory using below command

* cd node\_exporter-1.2.0.linux-amd64
* sudo cp node\_exporter /usr/local/bin

As early at the time of Prometheus installation we have created node\_exporter user and also updated the ownership permission so we will not repeat it again

## #16. Creating Node Exporter Systemd service

Now lets create a node\_exporter service in /lib/systemd/system directory named node\_exporter.service using below commands

* cd /lib/systemd/system
* sudo nano node\_exporter.service

Paste the below content in your service file

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

[Service]

Type=simple

User=node\_exporter

Group=node\_exporter

ExecStart=/usr/local/bin/node\_exporter \

— collector.mountstats \

— collector.logind \

— collector.processes \

— collector.ntp \

— collector.systemd \

— collector.tcpstat \

— collector.wifi

Restart=always

RestartSec=10s

[Install]

WantedBy=multi-user.target

Now lets start and enable the node\_exporter service using below commands

- sudo systemctl daemon-reload

* sudo systemctl enable node\_exporter
* sudo systemctl start node\_exporter
* sudo systemctl status node\_exporter

We have covered How to Install Prometheus and Grafana on Ubuntu 20.04 LTS with Node Exporter.

## #17. Configure the Node Exporter as a Prometheus target

Now to scrape the node\_exporter lets instruct the Prometheus by making a minor change in prometheus.yml file

So go to etc/prometheus and open prometheus.yml

* cd /etc/Prometheus
* sudo nano prometheus.yml

Now in **static\_configs**in your configuration file replace the target line with the below one

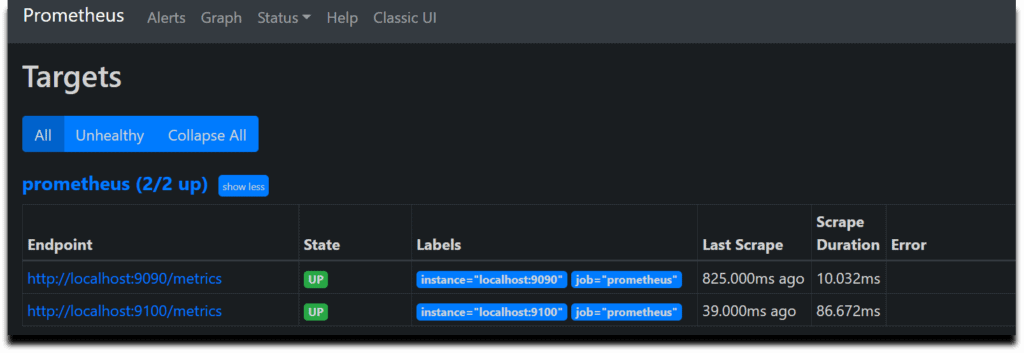
- targets: [‘localhost:9090’, ‘localhost:9100’]

Now restart the Prometheus Service

* sudo systemctl restart Prometheus

Hit the URL in your web browser to check weather our target is successfully scraped by Prometheus or not

* <https://localhost:9100/targets>

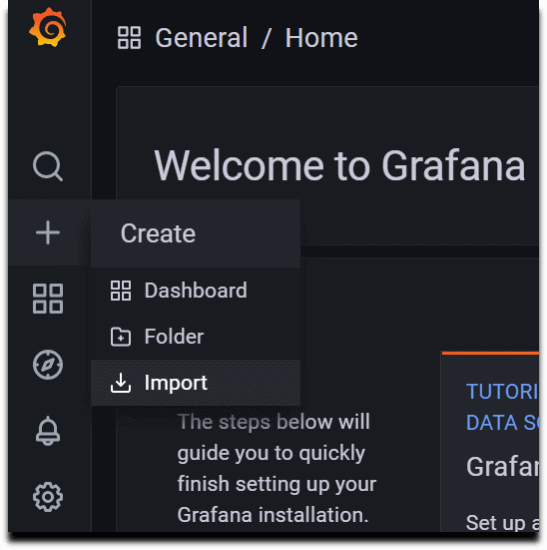


We have covered How to Install Prometheus and Grafana on Ubuntu 20.04 LTS with Node Exporter.

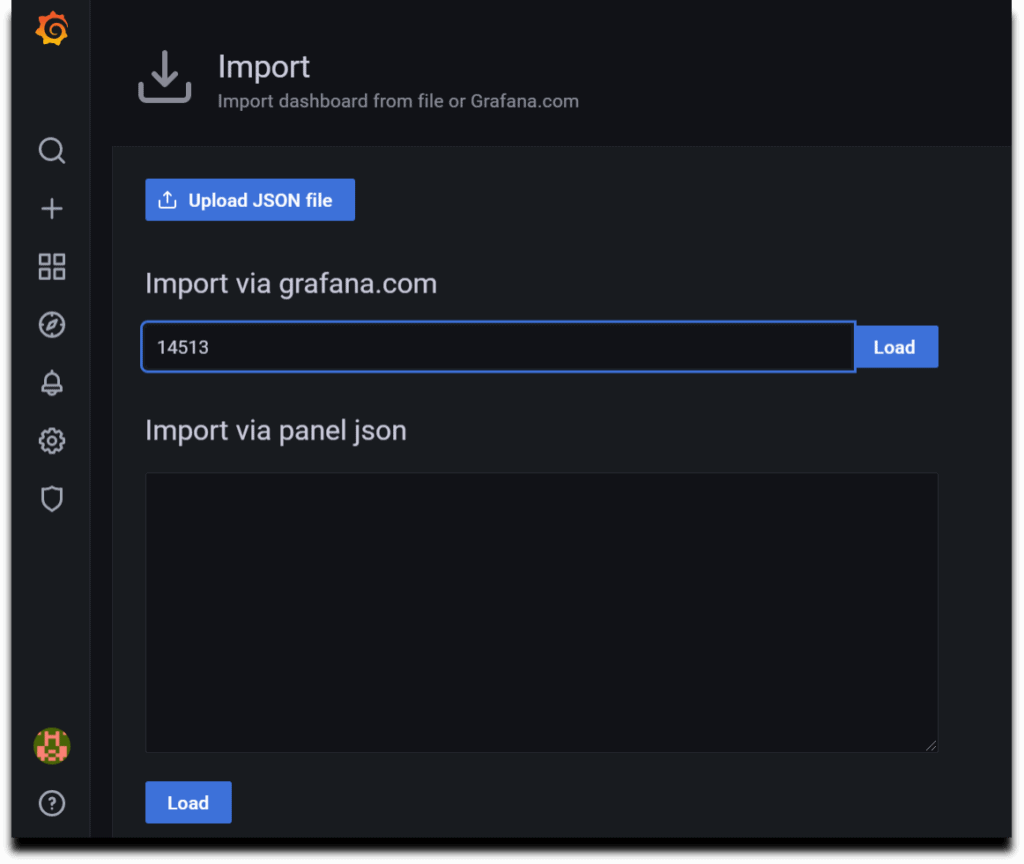
## #18. Creating Grafana Dashboard to Monitor Linux Server

Now lets build a dashboard in Grafana so then it will able to reflect the metrics of the Linux system.

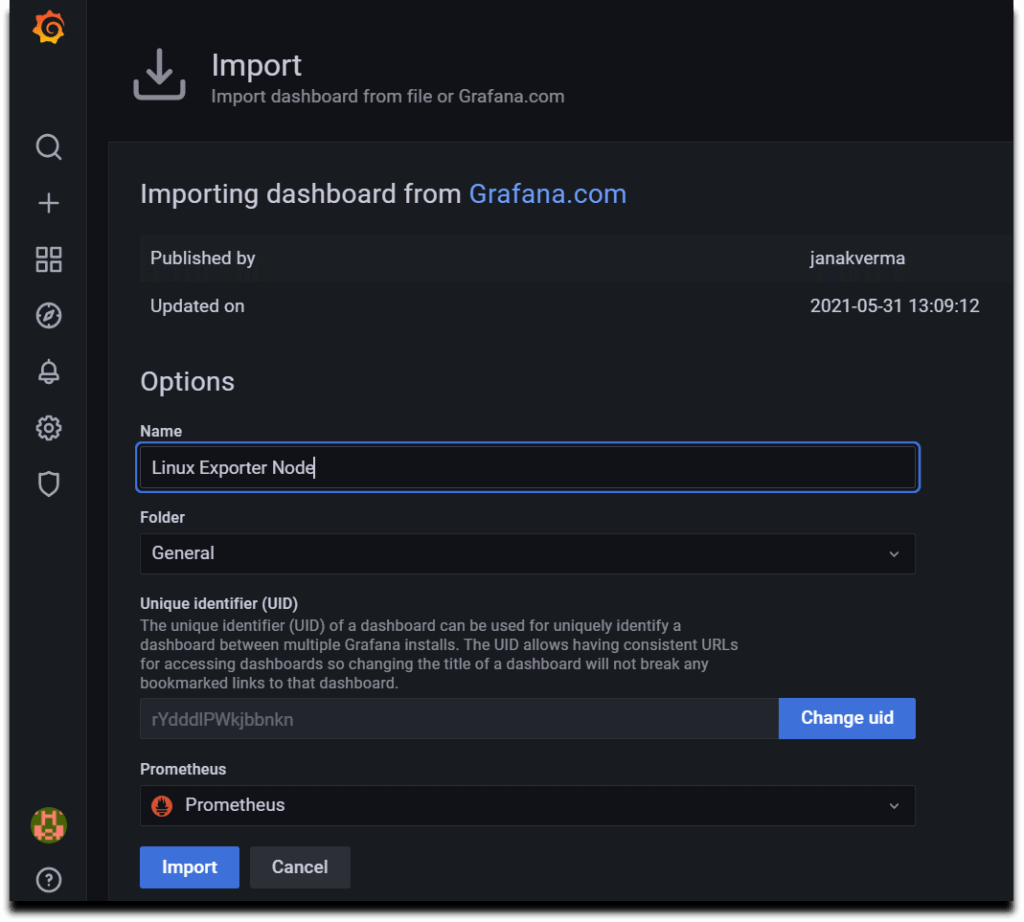
So we will use 14513 to import Grafana.com, Lets come to Grafana Home page and you can see a **“+”** icon. Click on that and select **“Import”**



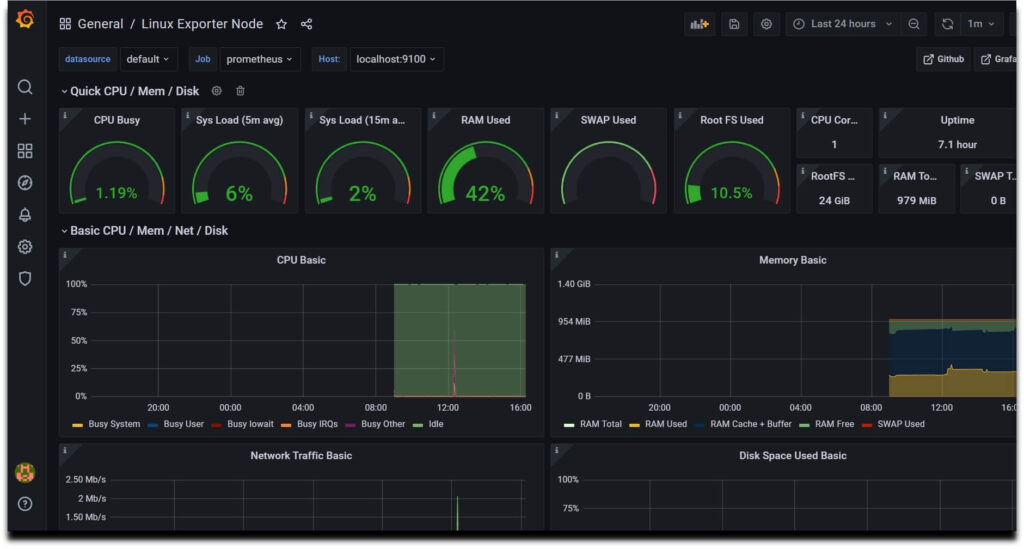
Now provide the Grafana.com Dashboard ID which is **14513**and click on **Load**



Now provide the name and select the Prometheus Datasource and click on **Import**.



There you are done with the setup. Now your Dashboard is running up!.



**Conclusion:**

In this Project we learned how to monitor a Linux server.