Installing Grafana:

```
sudo vim /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/oss/deb stable main
curl https://packages.grafana.com/gpg.key | sudo apt-key add -

sudo apt-get update
sudo apt-get install grafana

sudo service grafana-server start
sudo update-rc.d grafana-server defaults
```

Creating Users:

```
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

sudo chown prometheus:prometheus /etc/prometheus
sudo chown prometheus:prometheus /var/lib/prometheus
```

Placing Prometheus Files

https://prometheus.io/download/

```
wget https://github.com/prometheus/prometheus/releases/download/v2.7.1/prometheus-2.7.1.linux-amd64.tar.gz

tar xvf prometheus-2.7.1.linux-amd64.tar.gz

sudo cp prometheus-2.7.1.linux-amd64/prometheus /usr/local/bin/
sudo cp prometheus-2.7.1.linux-amd64/promtool /usr/local/bin/

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

sudo cp -r prometheus-2.7.1.linux-amd64/consoles /etc/prometheus
sudo cp -r prometheus-2.7.1.linux-amd64/console_libraries /etc/prometheus

sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries

rm -rf prometheus-2.7.1.linux-amd64.tar.gz prometheus-2.7.1.linux-amd64
```

Editing YAML

```
sudo vim /etc/prometheus/prometheus.yml
```

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
```

```
sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml

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
```

Kill this process by ctrl+c

Prometheus Service

```
sudo vim /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
```

Starting Prometheus:

```
sudo systemctl daemon-reload

sudo systemctl start prometheus

sudo systemctl status prometheus

sudo systemctl enable prometheus
```

Getting Node-Exporter

https://prometheus.io/download/

```
wget https://github.com/prometheus/node_exporter/releases/download/v0.17.0/node_exporter-0.17.0.linux-amd64.tar.gz

tar xvf node_exporter-0.17.0.linux-amd64.tar.gz

sudo cp node_exporter-0.17.0.linux-amd64/node_exporter /usr/local/bin
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter

rm -rf node_exporter-0.17.0.linux-amd64.tar.gz node_exporter-0.17.0.linux-amd64
```

Initializing Node-Exporter:

```
sudo vim /etc/systemd/system/node_exporter.service
```

```
[Unit]
```

```
Description=Node Exporter
Wants=network-online.target
After=network-online.target

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

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

Starting Node Exporter

```
sudo systemctl daemon-reload

sudo systemctl start node_exporter

sudo systemctl status node_exporter

sudo systemctl enable node_exporter
```

Adding Node-Exporter to Prometheus

```
sudo vim /etc/prometheus/prometheus.yml
```

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'node_exporter'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9100']
```

Restart Services:

```
sudo systemctl restart prometheus
```

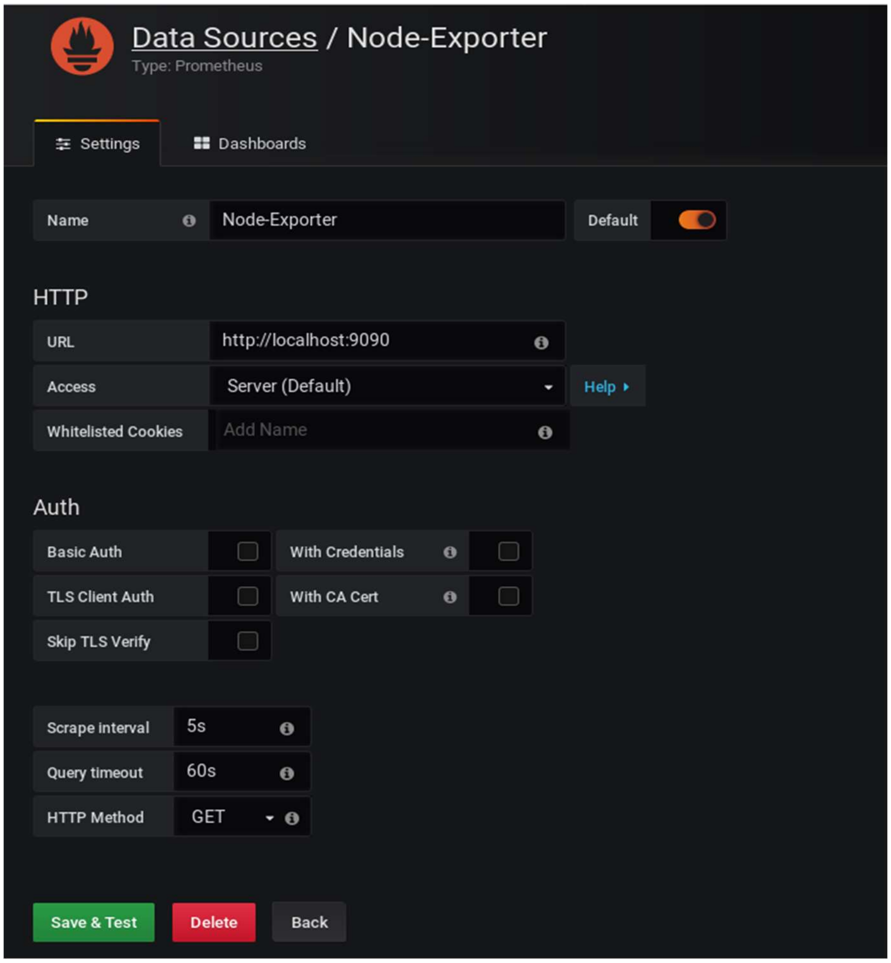
```
sudo systemctl status prometheus
```

Accessing Stuff:

Prometheus: http://localhost:9090
Grafana: http://localhost:3000
U: admin p: admin

Adding Datasource in Grafana:

Data source > Prometheus



Data Sources / Node-Exporter
Type: Prometheus

Name	Node-Exporter	Default	●

HTTP

URL	http://localhost:9090
Access	Server (Default)
Whitelisted Cookies	Add Name

Auth

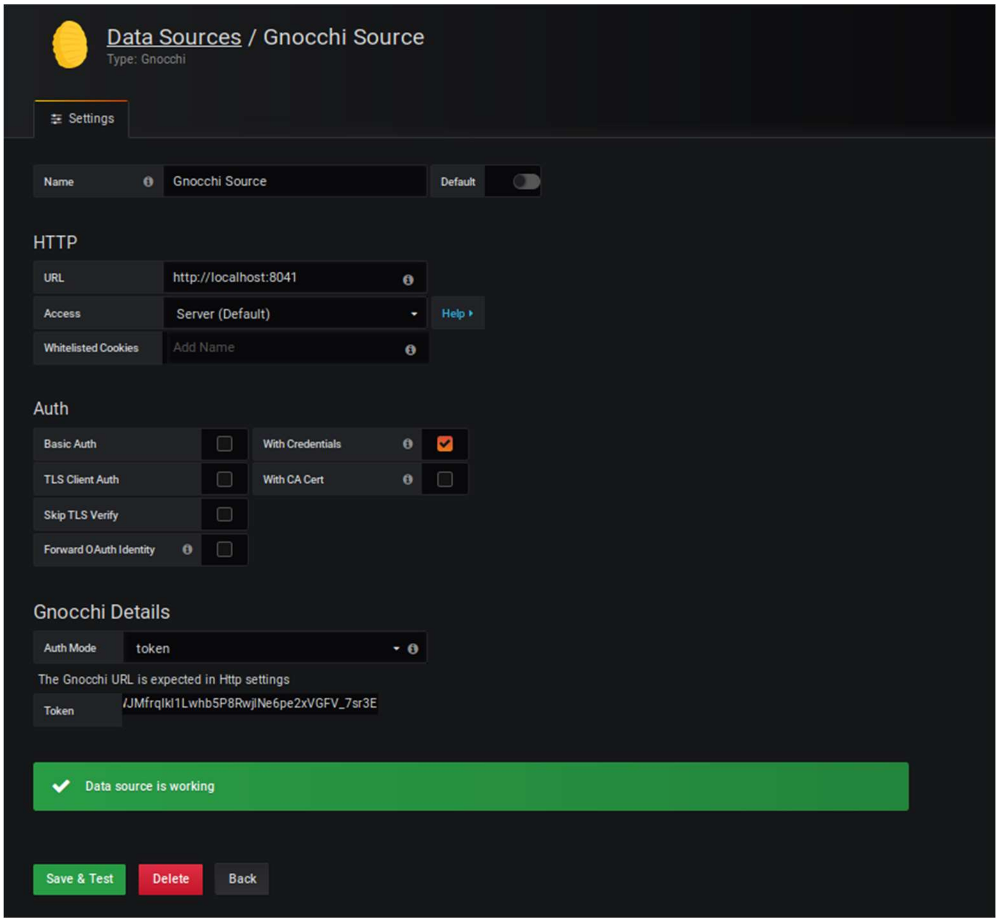
Basic Auth	□	With Credentials	□
TLS Client Auth	□	With CA Cert	□
Skip TLS Verify	□		

Scrape interval	5s
Query timeout	60s
HTTP Method	GET

Gnnochi:

```
sudo grafana-cli plugins install gnocchixyz-gnocchi-datasource
```

sudo service grafana-server
source /opt/stack/devstack/admin-openrc.sh
openstack token issue



Data Sources / Gnocchi Source
Type: Gnocchi

Name: Gnocchi Source
Default

HTTP

URL: http://localhost:8041
Access: Server (Default)

Whitelisted Cookies: Add Name

Auth

Basic Auth ☐
With Credentials ☑
TLS Client Auth ☐
With CA Cert ☐
Skip TLS Verify ☐
Forward OAuth Identity ☐

Gnocchi Details

Auth Mode: token
The Gnocchi URL is expected in Http settings
Token: /JMfrqlkl1Lwhb5P8RwjlNe6pe2xVGFV_7sr3E

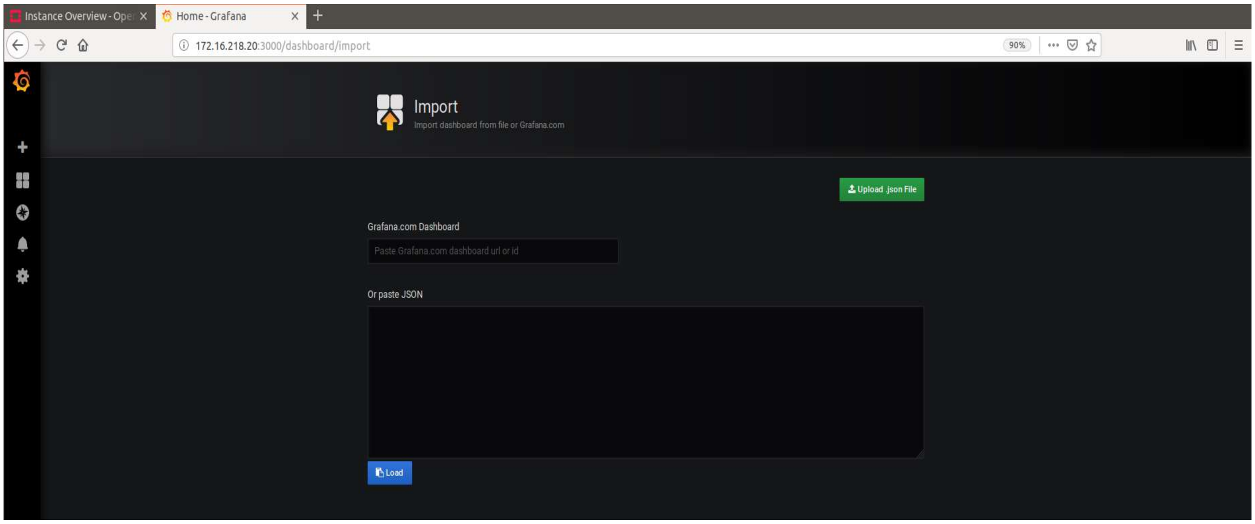
✓ Data source is working

Adding Dashboard:

To import the dashboard for collecting server metrics using Node Exporter, follow the below steps:

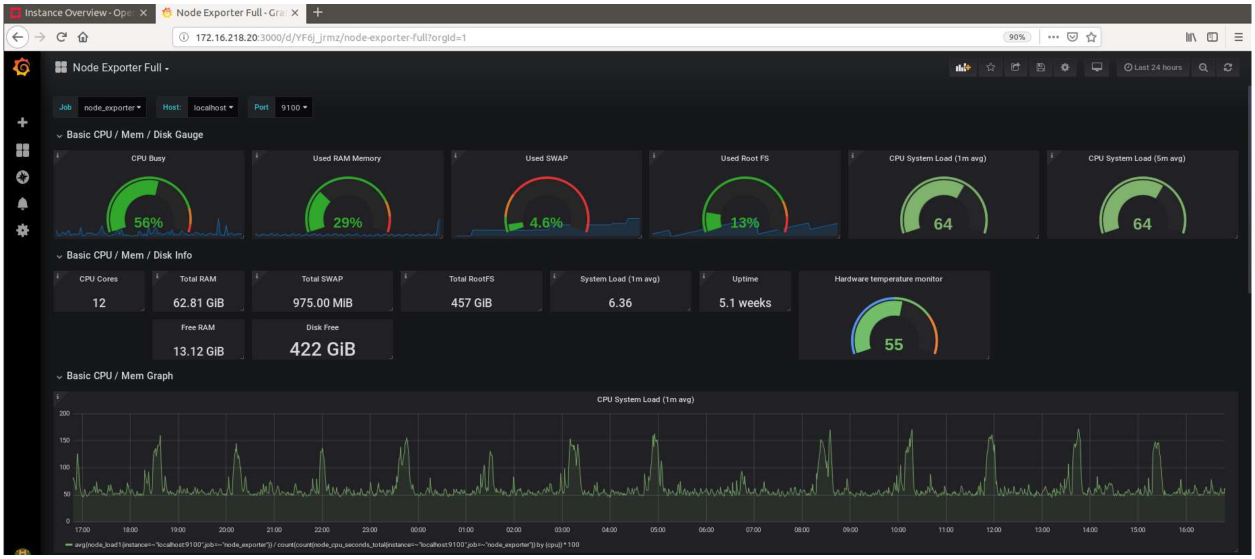
On the left end of the screen, hovering on the “+” sign, select “Import”. In the below screen, paste the contents of node_exporter.json of the git repository in the text box.

Once done, click on “Upload .json file” to create the new dashboard.



Instance Overview - Ope
Home - Grafana
172.16.218.20:3000/dashboard/import
90%
Import
Import dashboard from file or Grafana.com
Upload .json File
Grafana.com Dashboard
Paste Grafana.com dashboard url or id
Or paste JSON
Load

The dashboard after importing looks like the one below:



Job: node_exporter
Host: localhost
Port: 9100

CPU Cores: 12
Total RAM: 62.81 GiB
Free RAM: 13.12 GiB
Total SWAP: 975.00 MiB
Disk Free: 422 GiB
Total RootFS: 457 GiB
System Load (1m avg): 6.36
Uptime: 5.1 weeks

```
avg(node_load1{instance=~"localhost:9100",job=~"node_exporter"}) / count(count(node_cpu_seconds_total{instance=~"localhost:9100",job=~"node_exporter"}) by (cpu)) * 100
```