

Prometheus, Node Exporter, and Grafana Setup Documentation

Prometheus is an open-source system monitoring and alerting toolkit originally built at SoundCloud. It is now a standalone open source project . Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

Features

1. a multi-dimensional data model with time series data identified by metric name and key/value pairs
2. PromQL, a flexible query language to leverage this dimensionality
3. no reliance on distributed storage; single server nodes are autonomous
4. time series collection happens via a pull model over HTTP
5. pushing time series is supported via an intermediary gateway
6. targets are discovered via service discovery or static configuration
7. multiple modes of graphing and dashboarding support

Commands:

1. Prometheus Setup

Installing Prometheus

```
```bash
```

```
wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz
```

```
tar -xvf prometheus-2.47.1.linux-amd64.tar.gz
```

```
sudo mkdir -p /data /etc/prometheus
```

```
cd prometheus-2.47.1.linux-amd64/
```

```
sudo mv prometheus promtool /usr/local/bin/
```

```
sudo mv consoles/ console_libraries/ /etc/prometheus/
```

```
sudo mv prometheus.yml /etc/prometheus/prometheus.yml
```

```
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
```

```
```
```

Configuring Prometheus Service

Create a systemd service file:

```
```bash
```

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

```
```
```

Add the following content:

```
```ini
```

```
[Unit]
```

```
Description=Prometheus
```

```
Wants=network-online.target
```

```
After=network-online.target
```

```
StartLimitIntervalSec=500
```

```
StartLimitBurst=5
```

```
[Service]
```

User=prometheus

Group=prometheus

Type=simple

Restart=on-failure

RestartSec=5s

ExecStart=/usr/local/bin/prometheus \

--config.file=/etc/prometheus/prometheus.yml \

--storage.tsdb.path=/data \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console\_libraries \

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

--web.enable-lifecycle

[Install]

WantedBy=multi-user.target

...

### Starting Prometheus

```bash

sudo systemctl enable prometheus

sudo systemctl start prometheus

sudo systemctl status prometheus

journalctl -u prometheus -f --no-pager

...

Targets

All scrape pools | All | Unhealthy | Collapse All | Filter by endpoint or labels | Unknown | Unhealthy | Healthy

jenkins (0/1 up) [show less](#)

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|----------------------------------|-------|---|-------------|-----------------|---|
| http://localhost:8080/prometheus | DOWN | instance="localhost:8080" job="jenkins" | 14.59s ago | 12.727ms | server returned HTTP status 403 Forbidden |

node_export (1/1 up) [show less](#)

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|-------------------------------|-------|---|-------------|-----------------|-------|
| http://localhost:9100/metrics | UP | instance="localhost:9100" job="node_export" | 15.653s ago | 43.581ms | |

prometheus (1/1 up) [show less](#)

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|-------------------------------|-------|--|-------------|-----------------|-------|
| http://localhost:9090/metrics | UP | instance="localhost:9090" job="prometheus" | 6.985s ago | 50.190ms | |

2. Node Exporter Setup

Installing Node Exporter

```

```bash

wget
https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz

tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz

sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/

rm -rf node_exporter*

...

```

### ### Configuring Node Exporter Service

Create a systemd service file:

```
```bash
```

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

```
```
```

Add the following content:

```
```ini
```

```
[Unit]
```

```
Description=Node Exporter
```

```
Wants=network-online.target
```

```
After=network-online.target
```

```
StartLimitIntervalSec=500
```

```
StartLimitBurst=5
```

```
[Service]
```

```
User=node_exporter
```

```
Group=node_exporter
```

```
Type=simple
```

```
Restart=on-failure
```

```
RestartSec=5s
```

```
ExecStart=/usr/local/bin/node_exporter --collector.logind
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
```
```

```
Starting Node Exporter
```

```
```bash
```

```
sudo systemctl enable node_exporter
```

```
sudo systemctl start node_exporter
sudo systemctl status node_exporter
journalctl -u node_exporter -f --no-pager
...
```

3. Prometheus Configuration File

Modify Prometheus configuration:

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

Add the following content:

```
```yaml
scrape_configs:
  - job_name: node_export
    static_configs:
      - targets: ["localhost:9100"]

  - job_name: 'jenkins'
    metrics_path: '/prometheus'
    static_configs:
      - targets: ['<jenkins-ip>:8080']
...
```
```

Reload Prometheus configuration:

```
```bash
promtool check config /etc/prometheus/prometheus.yml
curl -X POST http://localhost:9090/-/reload
...
```
```

```
saran@FingerGripPC: ~
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"]

 - job_name: node_export
 static_configs:
 - targets: ["localhost:9100"]

 - job_name: 'jenkins'
 metrics_path: '/prometheus'
 static_configs:
 - targets: ['localhost:8080']

~
~
~
"/etc/prometheus/prometheus.yml" 38L, 1130B
38,35 All
```

## ## 4. Grafana Setup

### ### Installing Grafana

```
```bash
```

```
sudo apt-get install -y apt-transport-https software-properties-common
```

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

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

```
sudo apt-get update
```

```
sudo apt-get -y install grafana
```

```
```
```

### ### Starting Grafana

```
```bash
```

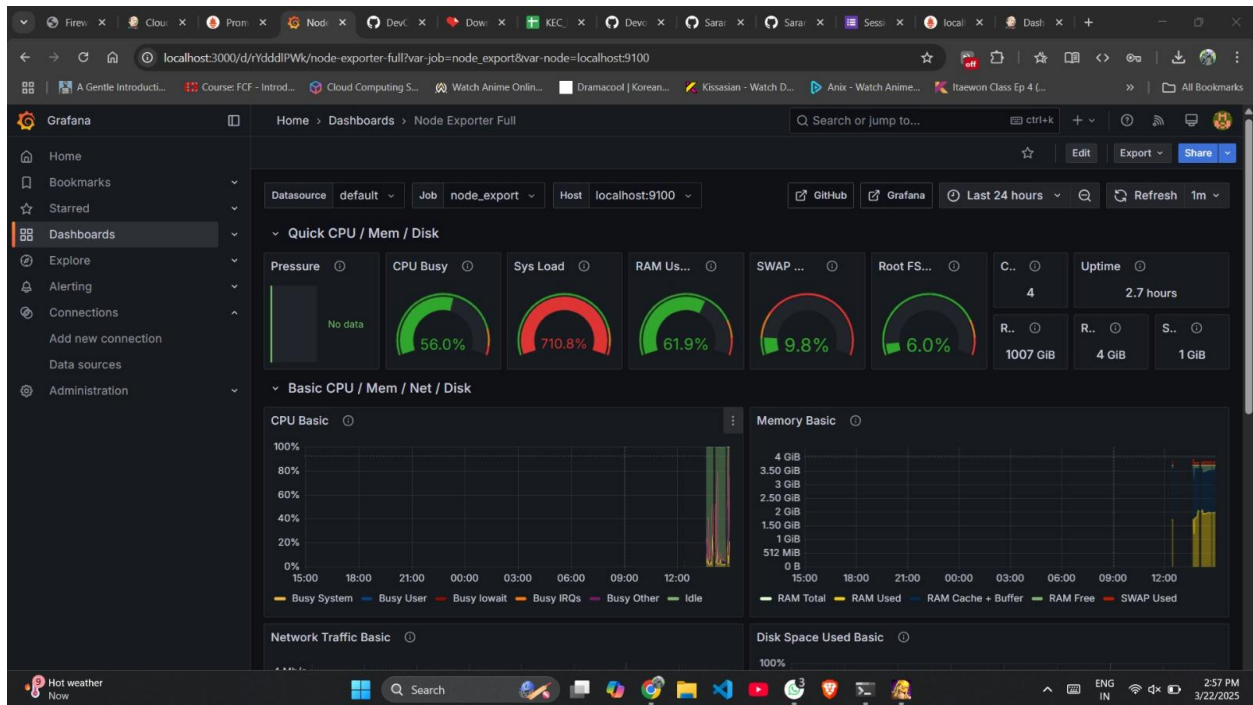
```
sudo systemctl enable grafana-server
```

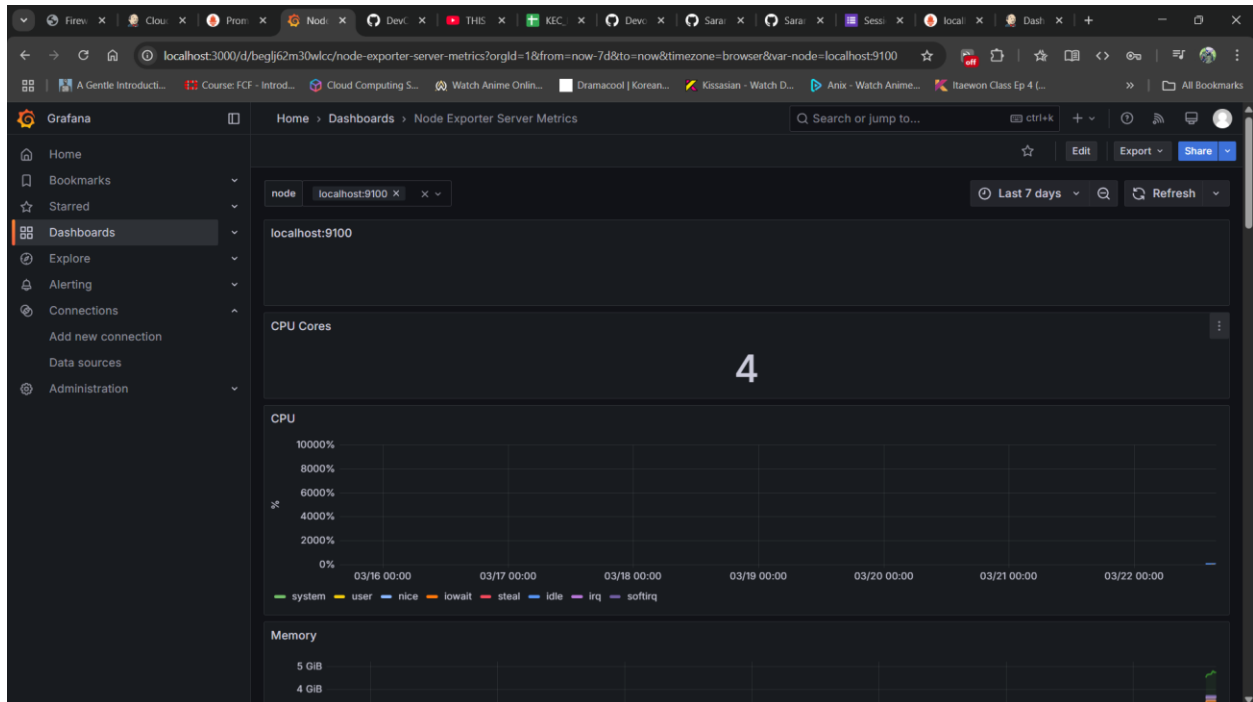
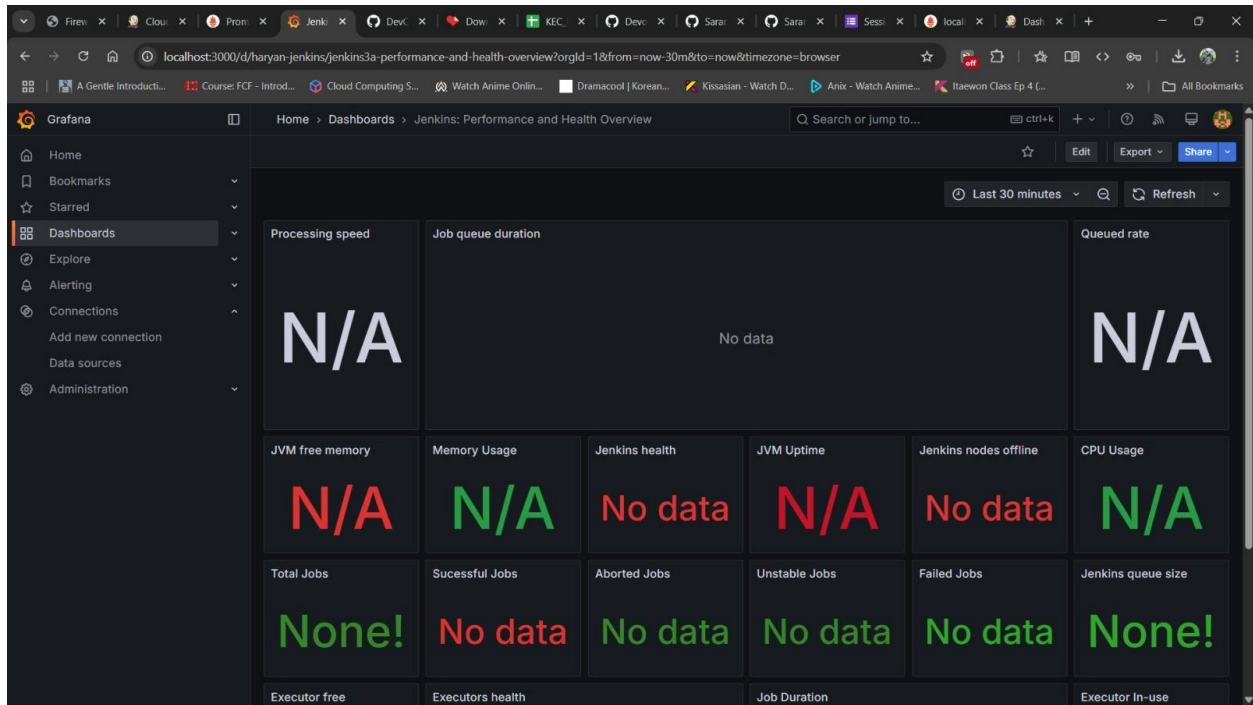
```
sudo systemctl start grafana-server
```

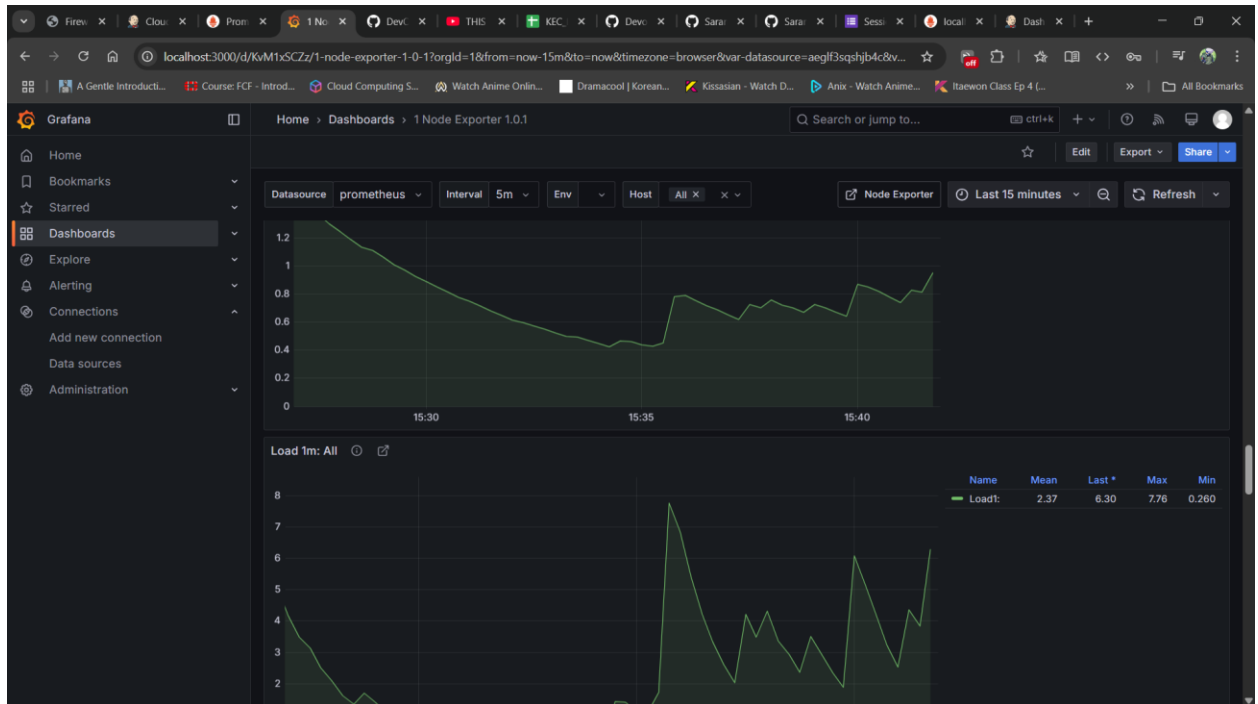
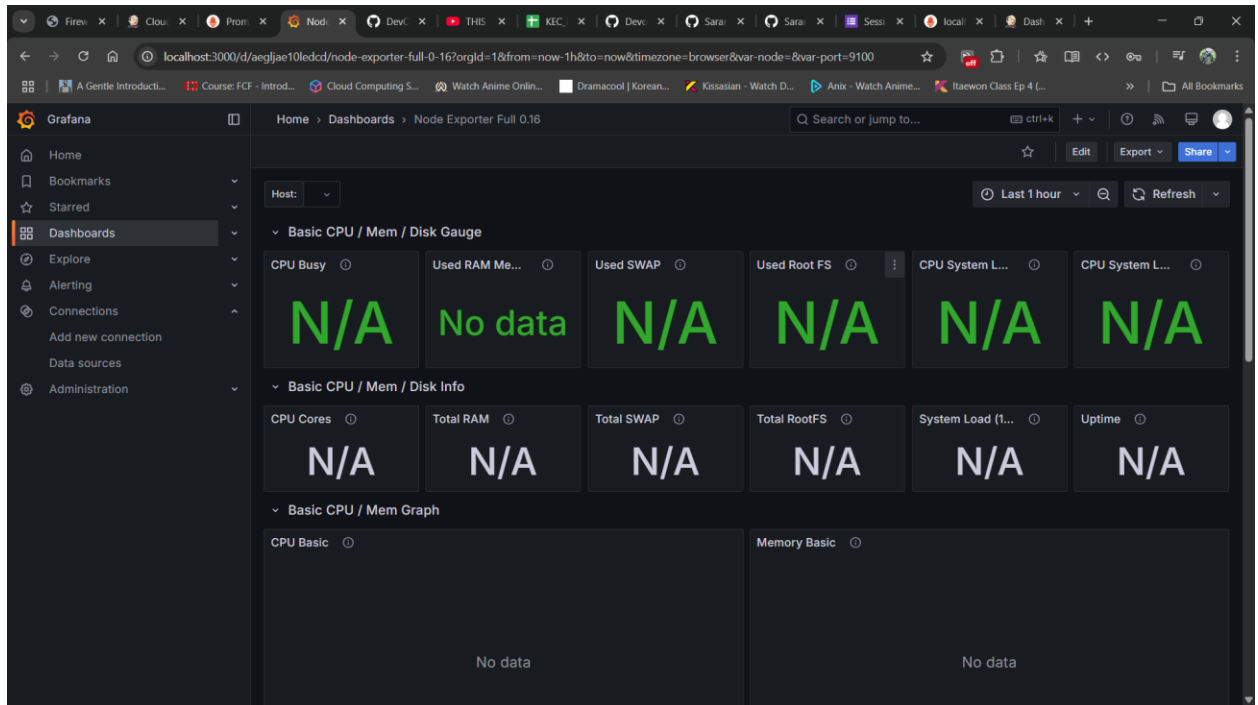
```
sudo systemctl status grafana-server
```

```
...
```

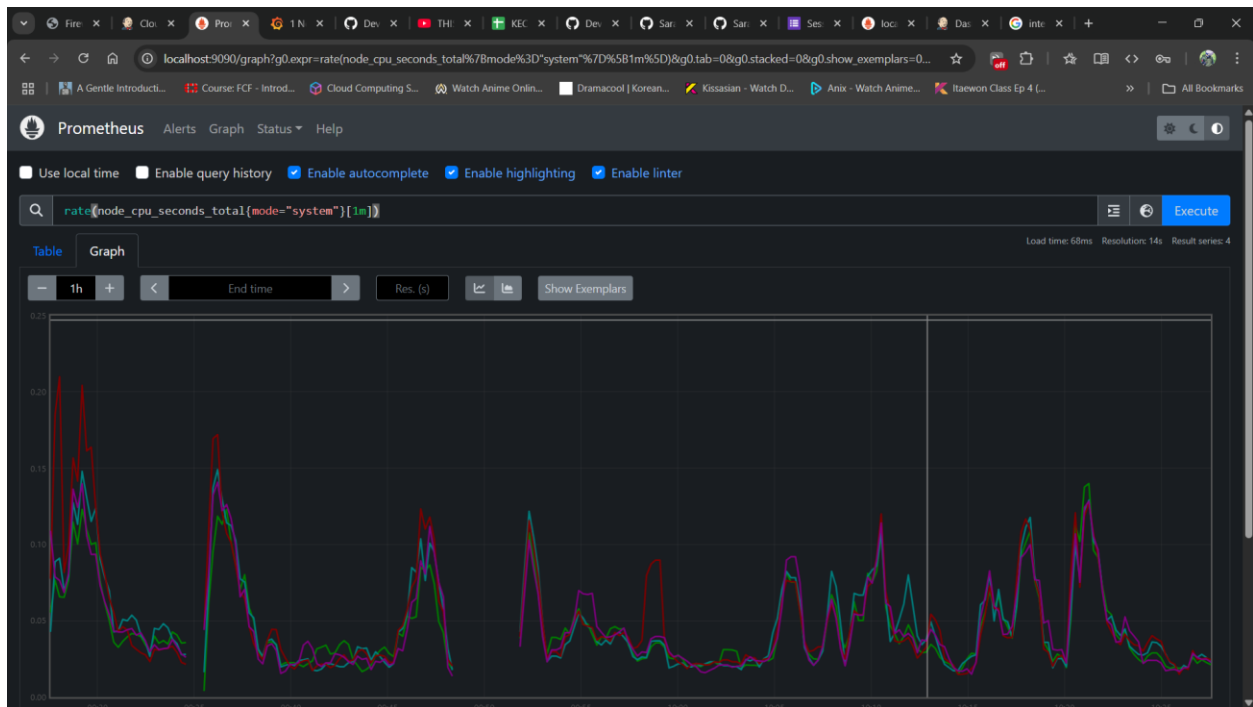
Dashboard:







`rate(node_cpu_seconds_total{mode="system"}[1m])`



`rate(node_network_receive_bytes_total[1m])`



node_load15

