DevOps

Day – 6

Assignment

Name:Debehaa J

Roll No:22CSR037

Prometheus

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.

Prometheus Architecture

Prometheus Server – Collects and stores metrics.

Pushgateway – Receives metrics from short-lived jobs.

Exporters – Agents that expose metrics (e.g., Node Exporter for system stats).

Alertmanager – Handles alerts based on defined rules

Grafana (Optional) – For visualization.

Common Prometheus Commands

sh CopyEdit prometheus --
config.file=prometheus.yml curl
http://localhost:9090/metrics promtool
check config prometheus.yml promtool
query instant up

Common Prometheus Use Cases

- Monitoring Kubernetes clusters
- Tracking system health (CPU, RAM, disk, network)
- Alerting on performance issues
- Logging API response times
- Monitoring microservices

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

PROMETHEUS INSTALLATION:

```
sudo useradd \
   --system \
   --no-create-home \
       --shell /bin/false prometheus
   wget
```

https://github.com/prometheus/prometheus/releases/download/v2.47.1/promethe us-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/

[12:00 PM, 3/22/2025] +91 90928 13114: cd rm -rf

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

-version sudo vim

/etc/systemd/system/prometheus.service [12:09 PM,

3/22/2025] +91 90928 13114: [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 sudo

systemctl enable prometheus sudo

systemctl start prometheus sudo

systemctl status prometheus journalctl -

u prometheus -f --no-pager sudo useradd

\

--system \

--no-create-home \

--shell /bin/false node_exporter

wget
https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_ex
porter-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* node_exporter --version sudo

vim /etc/systemd/system/node_exporter.service

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 sudo systemctl

enable node_exporter sudo systemctl start

node_exporter sudo systemctl status

node_exporter journalctl -u node_exporter

-f --no-pager

- job_name: 'jenkins' metrics_path: '/prometheus'

static_configs:

- targets: ['<jenkins-ip>:8080promtool check config

/etc/prometheus/prometheus.yml curl -X POST

http://localhost:9090/-/reload 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

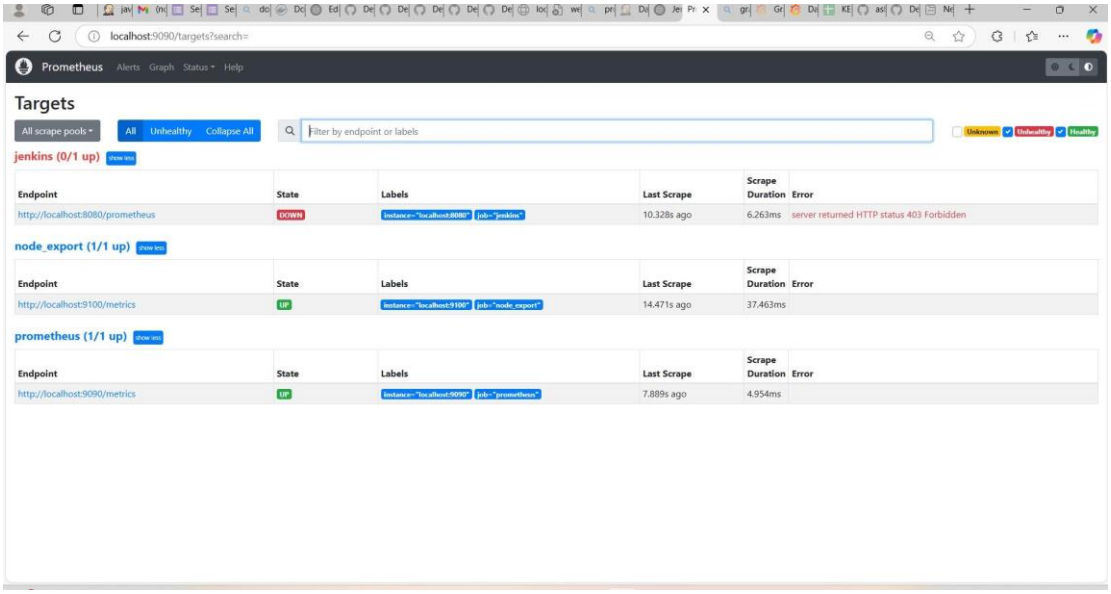
sudo apt-get update sudo apt-get -y

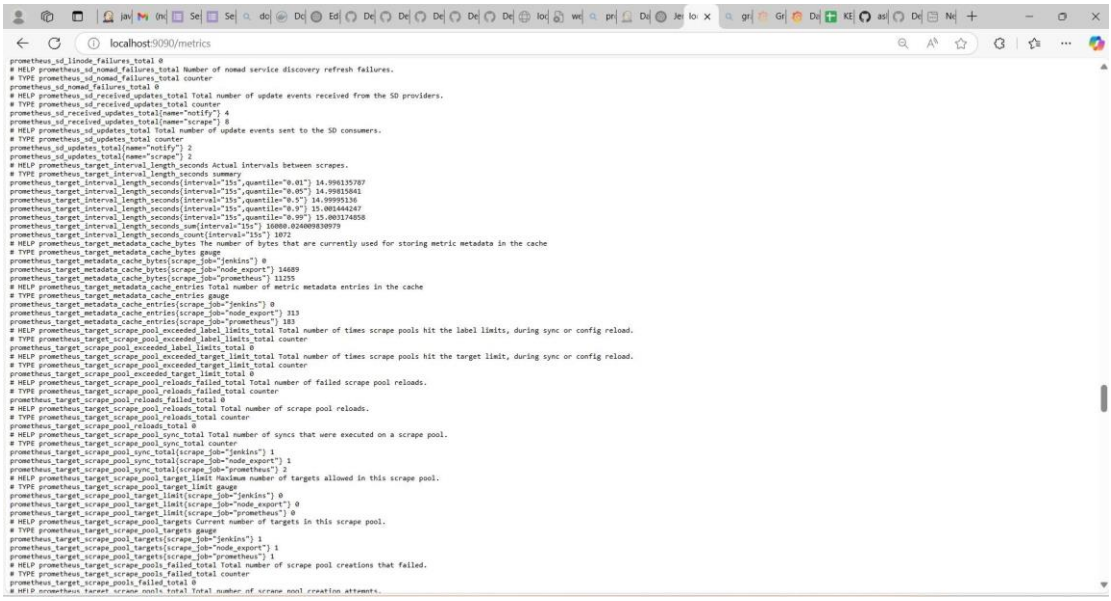
install grafana sudo systemctl enable

grafana-server sudo systemctl start

grafana-server sudo systemctl status

grafana-server





QUERY:

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

node_cpu_seconds_total: This metric represents the total CPU time spent in different modes (user, system, idle, etc.). mode="system": Filters only CPU time spent in **system/kernel mode**.

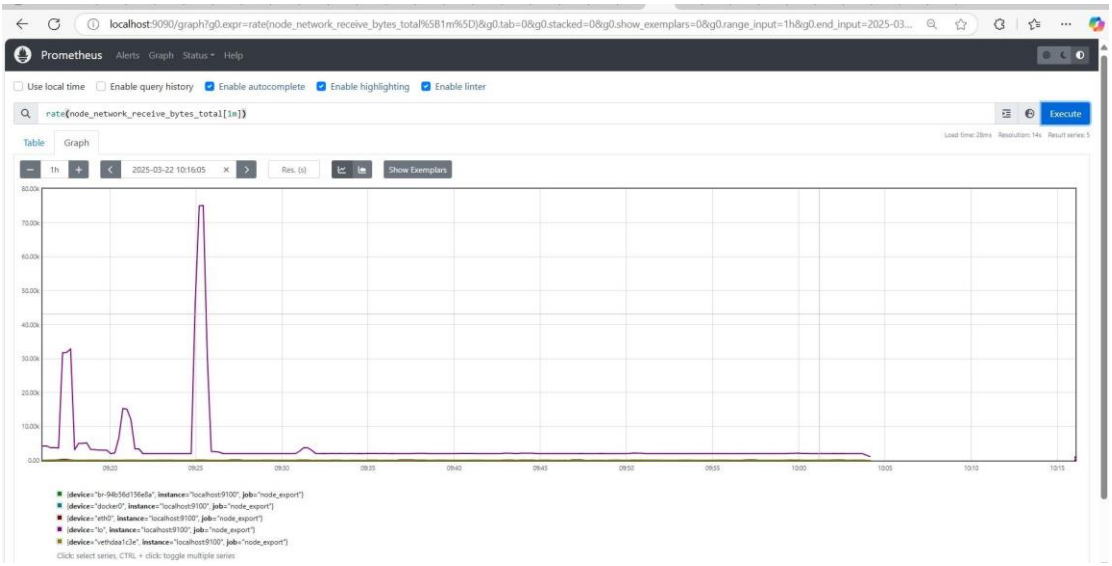
rate(...[1m]): Calculates the **per-second increase** of this metric over the last **1 minute**.

What it does:

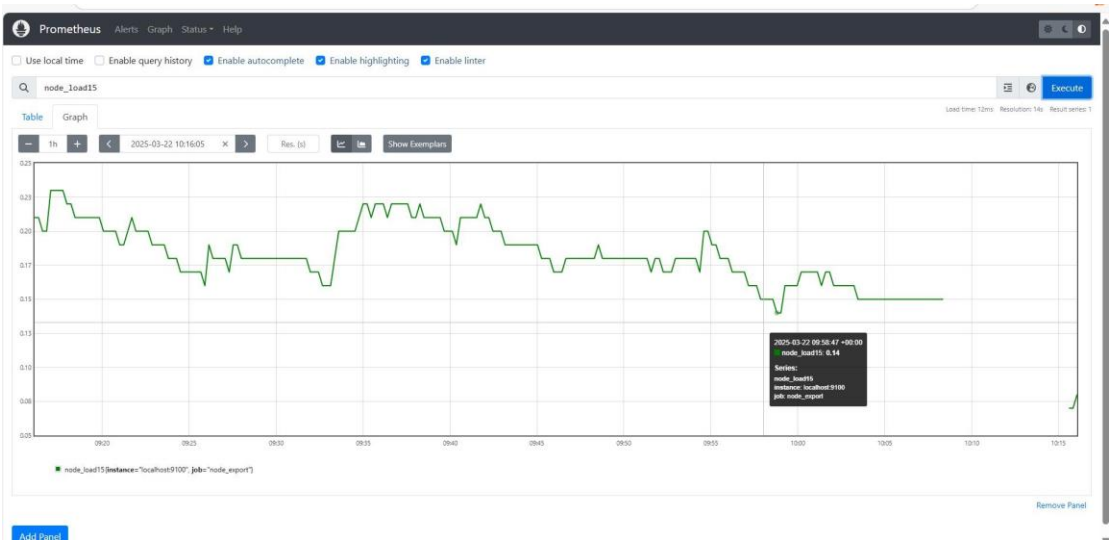
This query shows the **CPU usage in system mode per second** over the past 1 minute.

Useful for detecting high system resource consumption by kernel processes.

rate(node_network_receive_bytes_total[1m])



node_load15

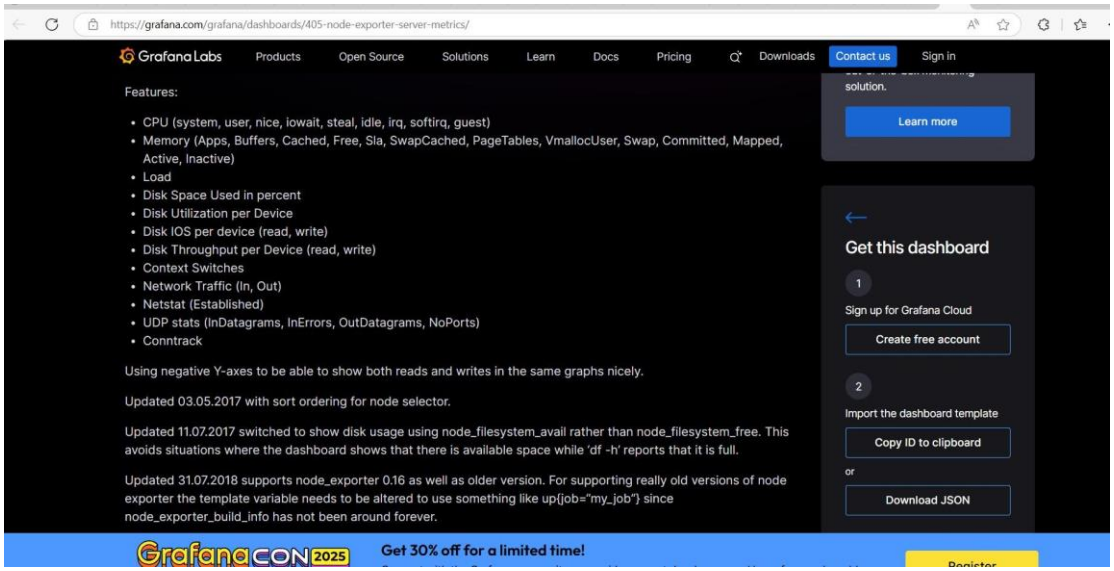


GRAFANA:

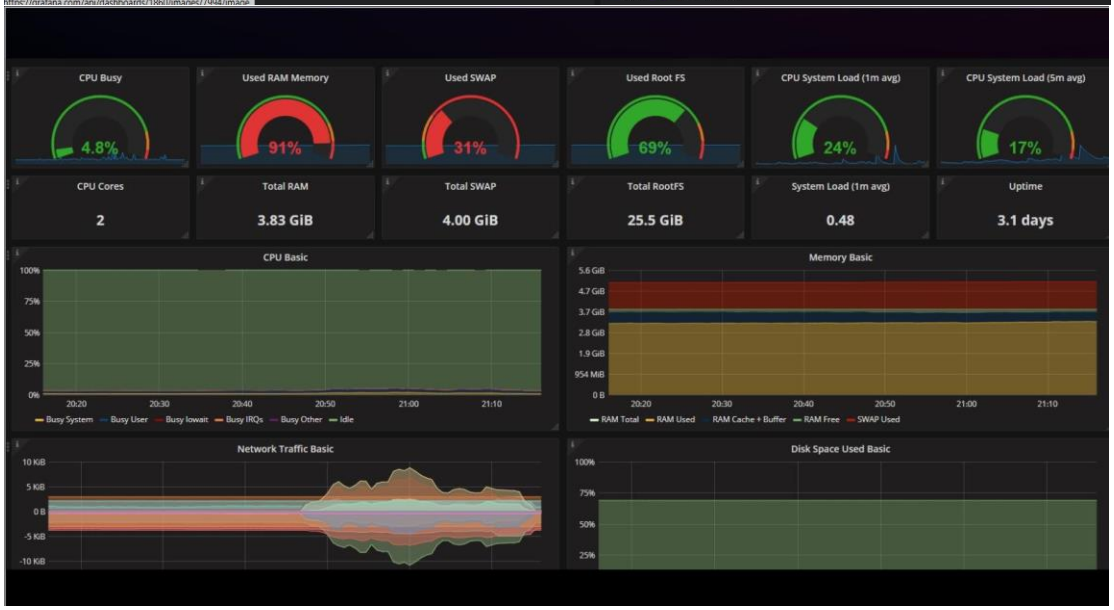
Grafana is an **open-source analytics and visualization platform** used for **monitoring and observability**. It allows users to create **interactive dashboards** from multiple data sources like **Prometheus, InfluxDB, Elasticsearch, MySQL, and more**.

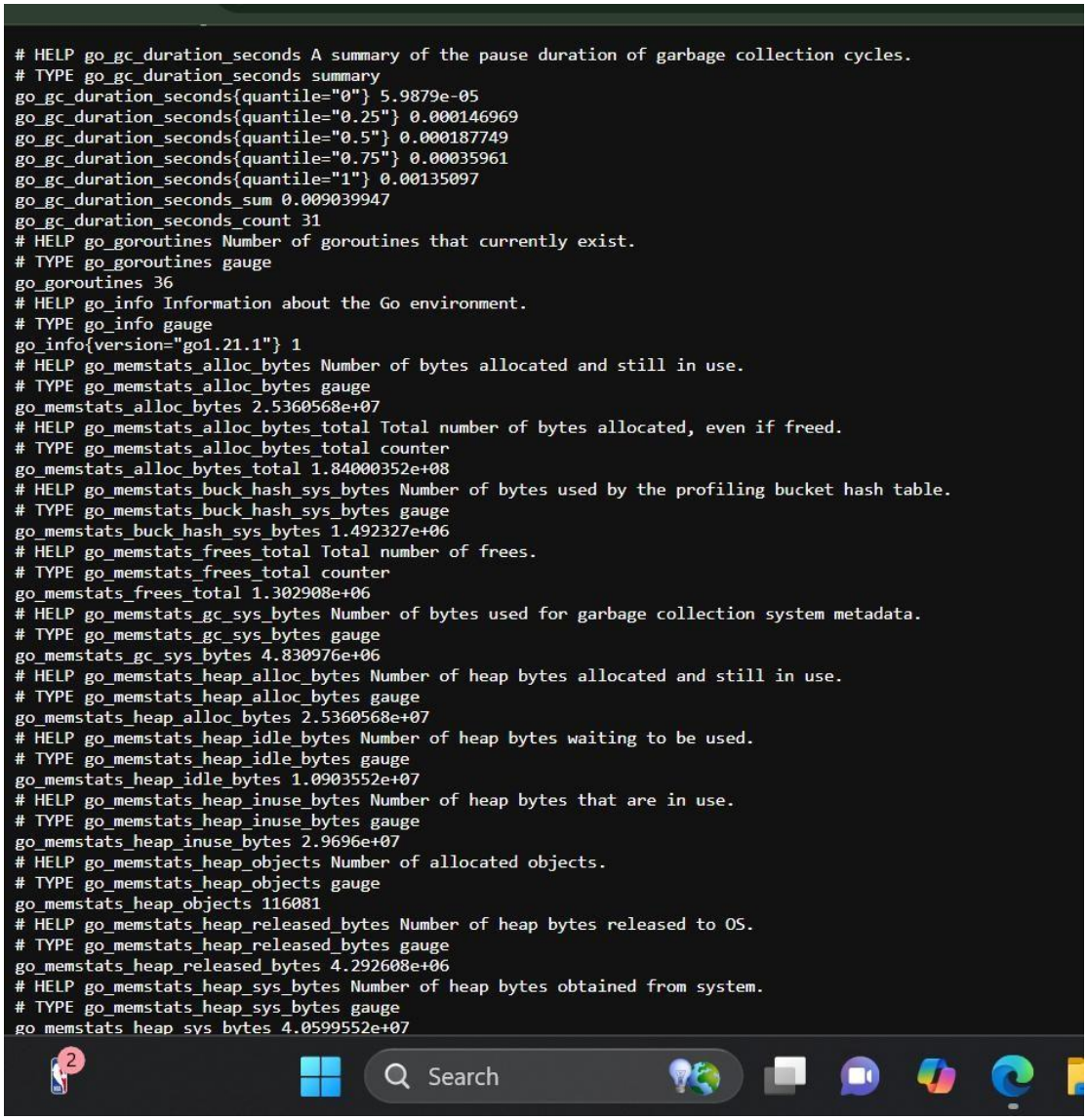
GRAFANA INSTALLATION:

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 sudo systemctl enable
grafana-server sudo systemctl start
grafana-server sudo systemctl status
grafana-server

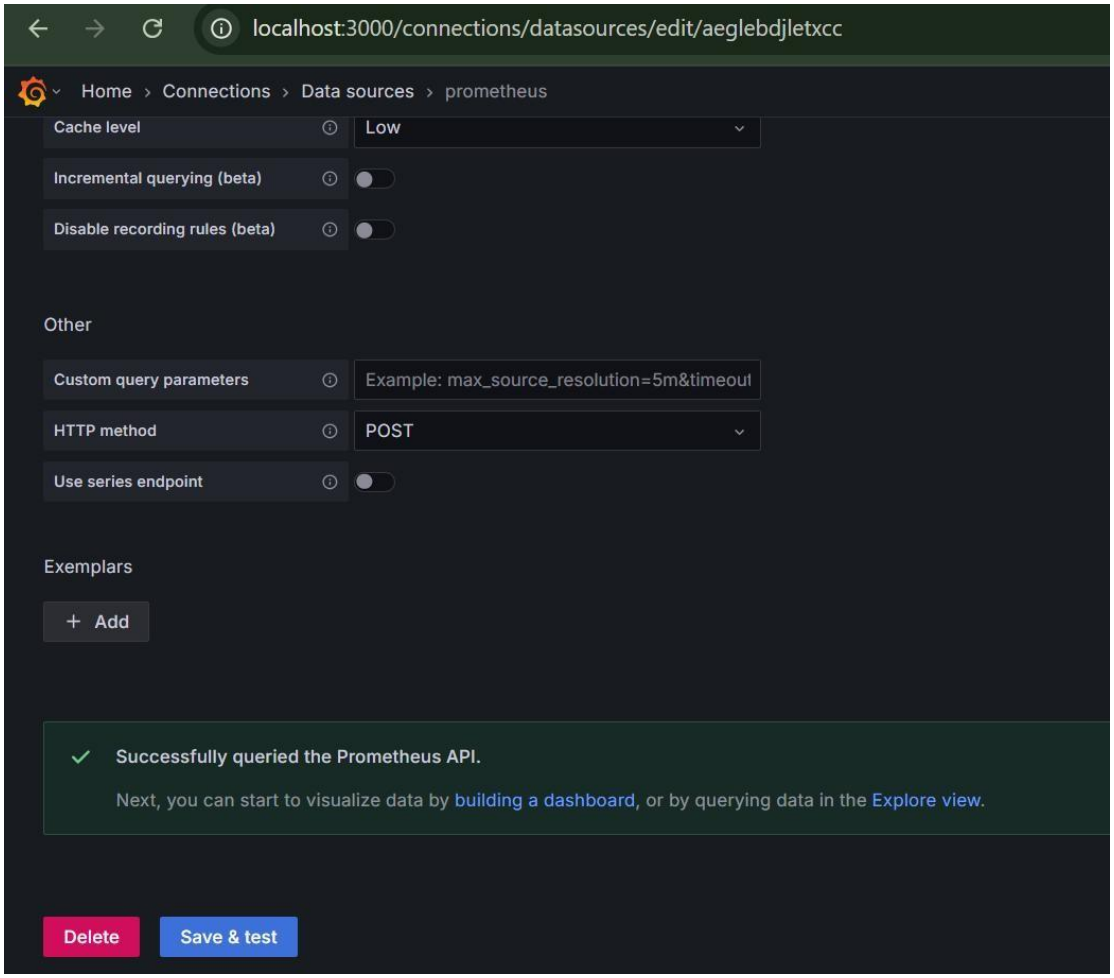

https://grafana.com/grafana/dashboards/405-node-exporter-server-metrics/
Grafana Labs
Products
Open Source
Solutions
Learn
Docs
Pricing
Downloads
Contact us
Sign in
Features:
CPU (system, user, nice, iowait, steal, idle, irq, softirq, guest)
Memory (Apps, Buffers, Cached, Free, Sla, SwapCached, PageTables, VmallocUser, Swap, Committed, Mapped, Active, Inactive)
Load
Disk Space Used in percent
Disk Utilization per Device
Disk IOS per device (read, write)
Disk Throughput per Device (read, write)
Context Switches
Network Traffic (In, Out)
Netstat (Established)
UDP stats (InDatagrams, InErrors, OutDatagrams, NoPorts)
Conntrack
Using negative Y-axes to be able to show both reads and writes in the same graphs nicely.
Updated 03.05.2017 with sort ordering for node selector.
Updated 11.07.2017 switched to show disk usage using node_filesystem_avail rather than node_filesystem_free. This avoids situations where the dashboard shows that there is available space while 'df -h' reports that it is full.
Updated 31.07.2018 supports node_exporter 0.16 as well as older version. For supporting really old versions of node exporter the template variable needs to be altered to use something like up{job="my_job"} since node_exporter_build_info has not been around forever.
solution.
Learn more
Get this dashboard
1
Sign up for Grafana Cloud
Create free account
2
Import the dashboard template
Copy ID to clipboard
or
Download JSON
GrafanaCON 2025
Get 30% off for a limited time!

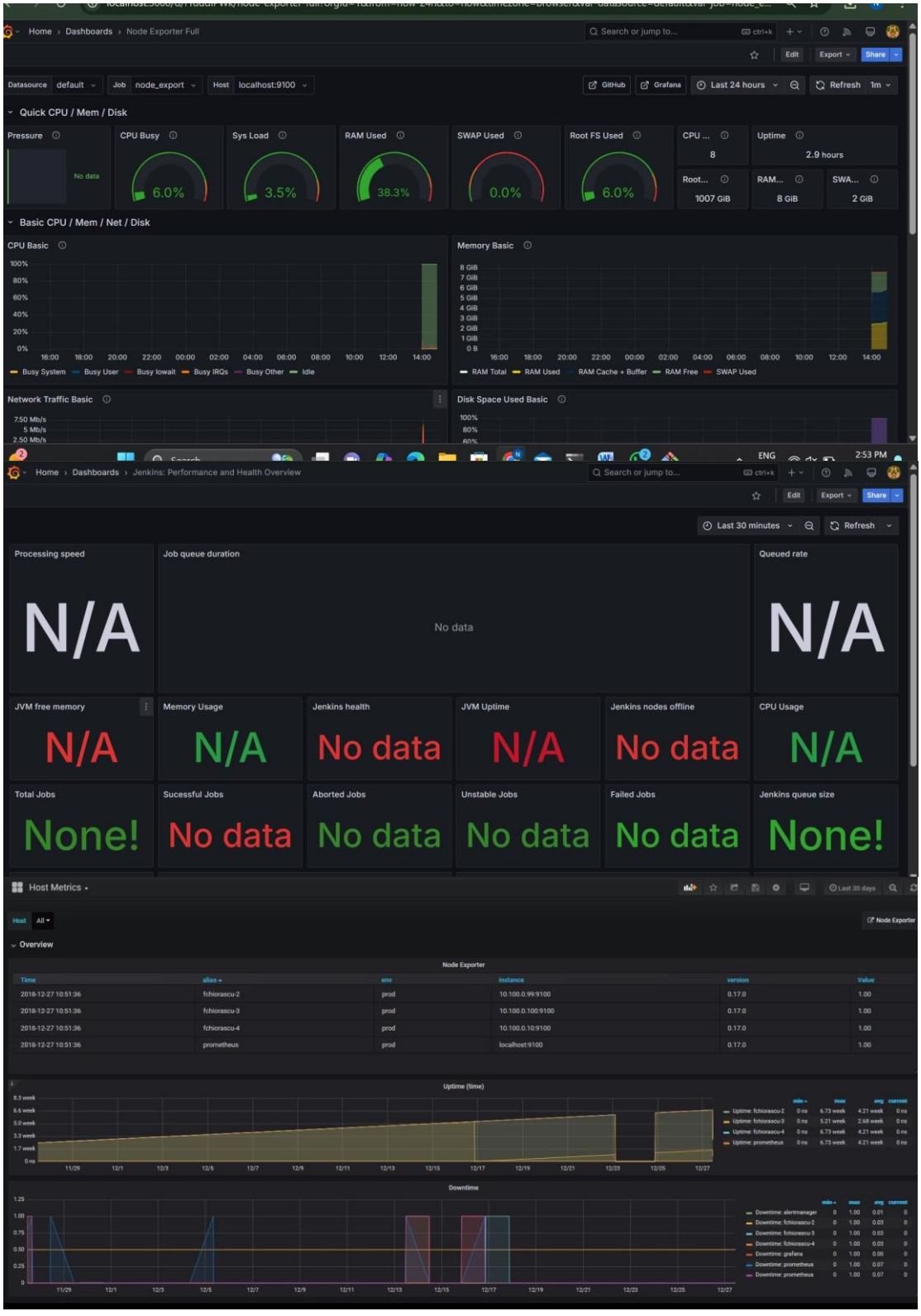

Node Exporter Full
Quick CPU / Mem / Disk
CPU Busy
Sys Load (5m avg)
Sys Load (15m avg)
RAM Used
SWAP Used
Root FS Used
CPU Cores
Uptime
RootFS Total
RAM Total
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic
Network Traffic Basic
Disk Space Used Basic


CPU Busy
Used RAM Memory
Used SWAP
Used Root FS
CPU System Load (1m avg)
CPU System Load (5m avg)
CPU Cores
Total RAM
Total SWAP
Total RootFS
System Load (1m avg)
Uptime
CPU Basic
Memory Basic
Network Traffic Basic
Disk Space Used Basic



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"} 5.9879e-05
go_gc_duration_seconds{quantile="0.25"} 0.000146969
go_gc_duration_seconds{quantile="0.5"} 0.000187749
go_gc_duration_seconds{quantile="0.75"} 0.00035961
go_gc_duration_seconds{quantile="1"} 0.00135097
go_gc_duration_seconds_sum 0.009039947
go_gc_duration_seconds_count 31
HELP go_goroutines Number of goroutines that currently exist.
TYPE go_goroutines gauge
go_goroutines 36
HELP go_info Information about the Go environment.
TYPE go_info gauge
go_info{version="go1.21.1"} 1
HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.5360568e+07
HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.84000352e+08
HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.492327e+06
HELP go_memstats_frees_total Total number of frees.
TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.302908e+06
HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 4.830976e+06
HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.5360568e+07
HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 1.0903552e+07
HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 2.9696e+07
HELP go_memstats_heap_objects Number of allocated objects.
TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 116081
HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 4.292608e+06
HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 4.0599552e+07
2
Search


localhost:3000/connections/datasources/edit/aeglebdjletxcc
Home › Connections › Data sources › prometheus
Cache level
Low
Incremental querying (beta)
Disable recording rules (beta)
Other
Custom query parameters
Example: max_source_resolution=5m&timeout
HTTP method
POST
Use series endpoint
Exemplars
+ Add
Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete
Save & test


Home › Dashboards › Node Exporter Full
Search or jump to...
Edit
Export
Share
Datasource default
Job node_export
Host localhost:9100
GitHub
Grafana
Last 24 hours
Refresh 1m
Quick CPU / Mem / Disk
Pressure
No data
CPU Busy
6.0%
Sys Load
3.5%
RAM Used
38.3%
SWAP Used
0.0%
Root FS Used
6.0%
CPU ...
8
Uptime
2.9 hours
Root...
1007 GiB
RAM...
8 GiB
SWA...
2 GiB
Basic CPU / Mem / Net / Disk
CPU Basic
Busy System
Busy User
Busy Iowait
Busy IRQs
Busy Other
Idle
Memory Basic
RAM Total
RAM Used
RAM Cache + Buffer
RAM Free
SWAP Used
Network Traffic Basic
Disk Space Used Basic
Home › Dashboards › Jenkins: Performance and Health Overview
Search or jump to...
Edit
Export
Share
Last 30 minutes
Refresh
Processing speed
N/A
Job queue duration
No data
Queued rate
N/A
JVM free memory
N/A
Memory Usage
N/A
Jenkins health
No data
JVM Uptime
N/A
Jenkins nodes offline
No data
CPU Usage
N/A
Total Jobs
None!
Sucessful Jobs
No data
Aborted Jobs
No data
Unstable Jobs
No data
Failed Jobs
No data
Jenkins queue size
None!
Host Metrics
Last 30 days
Host
All
Node Exporter
Overview
Node Exporter
Time
alias
env
instance
version
Value
2018-12-27 10:51:36
fchiorascu-2
prod
10.100.0.99:9100
0.17.0
1.00
2018-12-27 10:51:36
fchiorascu-3
prod
10.100.0.100:9100
0.17.0
1.00
2018-12-27 10:51:36
fchiorascu-4
prod
10.100.0.10:9100
0.17.0
1.00
2018-12-27 10:51:36
prometheus
prod
localhost:9100
0.17.0
1.00
Uptime (time)
Downtime