

RASWANTHSABARISH – 22CSR160

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

Prometheus Installation:

Username Creation:

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

Commands:

```
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/

cd

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

prometheus --version

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

```
ragu_ubuntu@Kavin: ~/prometheus$ + 
prometheus-2.47.1.linux-amd64/consoles/prometheus.html
prometheus-2.47.1.linux-amd64/consoles/prometheus-overview.html
prometheus-2.47.1.linux-amd64/consoles/node-cpu.html
prometheus-2.47.1.linux-amd64/consoles/index.html.example
prometheus-2.47.1.linux-amd64/consoles/node.html
prometheus-2.47.1.linux-amd64/consoles/node-disk.html
prometheus-2.47.1.linux-amd64/consoles/node-overview.html
prometheus-2.47.1.linux-amd64/promtool
^[[A
gzip: stdin: unexpected end of file
tar: Unexpected EOF in archive
tar: Unexpected EOF in archive
tar: Error is not recoverable: exiting now
^[[Amv: cannot stat 'prometheus': No such file or directory
mv: cannot stat 'console_libraries': No such file or directory
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ wget https://github.com/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 /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.yaml /etc/prometheus/prometheus.yaml
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
--2025-03-22 06:31:46-- https://github.com/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/2f9b7b37-63a0-428b-adb5-0294482fd743?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250322T063045ZGX-Amz-Expires=300&X-Amz-Signature=ed575488b409680d63dd5daa828b8b4c87cbfe8c835d5f52ce1c05389b4809d96X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.47.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-03-22 06:31:46-- https://objects.githubusercontent.com/github-production-release-asset%2F20250322T063045ZGX-Amz-Algorithm=AWS4-HMAC-SHA256X-Amz-Credential=releaseassetproduction%2F20250322T063045ZGX-Amz-Expires=300&X-Amz-Signature=ed575488b409680d63dd5daa828b8b4c87cbfe8c835d5f52ce1c05389b4809d96X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.47.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.

```

```
ragu_ubuntu@Kavin: ~/prometheus$ + 
rm -rf prometheus-2.47.1.linux-amd64.tar.gz
prometheus --version
sudo vim /etc/systemd/system/prometheus.service
prometheus, version 2.47.1 (branch: HEAD, revision: c4d1a8beff37cc004f1dc4ab9d2e73193f51aaeb)
  build user:          root@4829330363be
  build date:         20231004-10:31:16
  go version:        go1.21.1
  platform:          linux/amd64
  tags:              netgo,builtinassets,stringlabels
ragu_ubuntu@Kavin: $ ls
deploy.yaml docker-compose.yaml ns.yaml pod-ns.yaml pod.yaml prometheus-2.47.1.linux-amd64 rs.yaml
ragu_ubuntu@Kavin: $ cd
.aws/           .docker/           .landscape/           prometheus-2.47.1.linux-amd64/
.azure/         .jenkins/         .local/             .minikube/
.cache/         .kube/            .minikube/
ragu_ubuntu@Kavin: $ cd prometheus-2.47.1.linux-amd64/
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ ld
ld: no input files
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ ls
LICENSE NOTICE prometheus-2.47.1.linux-amd64 prometheus-2.47.1.linux-amd64.tar.gz
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ rm
LICENSE          prometheus-2.47.1.linux-amd64/
NOTICE          prometheus-2.47.1.linux-amd64.tar.gz
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ rm prometheus-2.47.1.linux-amd64
prometheus-2.47.1.linux-amd64/          prometheus-2.47.1.linux-amd64.tar.gz
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ rm prometheus-2.47.1.linux-amd64.tar.gz
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ ls
LICENSE NOTICE prometheus-2.47.1.linux-amd64
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ cd prometheus-2.47.1.linux-amd64/
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ prometheus --version
prometheus, version 2.47.1 (branch: HEAD, revision: c4d1a8beff37cc004f1dc4ab9d2e73193f51aaeb)
  build user:          root@4829330363be
  build date:         20231004-10:31:16
  go version:        go1.21.1
  platform:          linux/amd64
  tags:              netgo,builtinassets,stringlabels
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ sudo vim /etc/systemd/system/prometheus.service
ragu_ubuntu@Kavin:~/prometheus-2.47.1.linux-amd64$ 
```

Prometheus.service:

[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 \

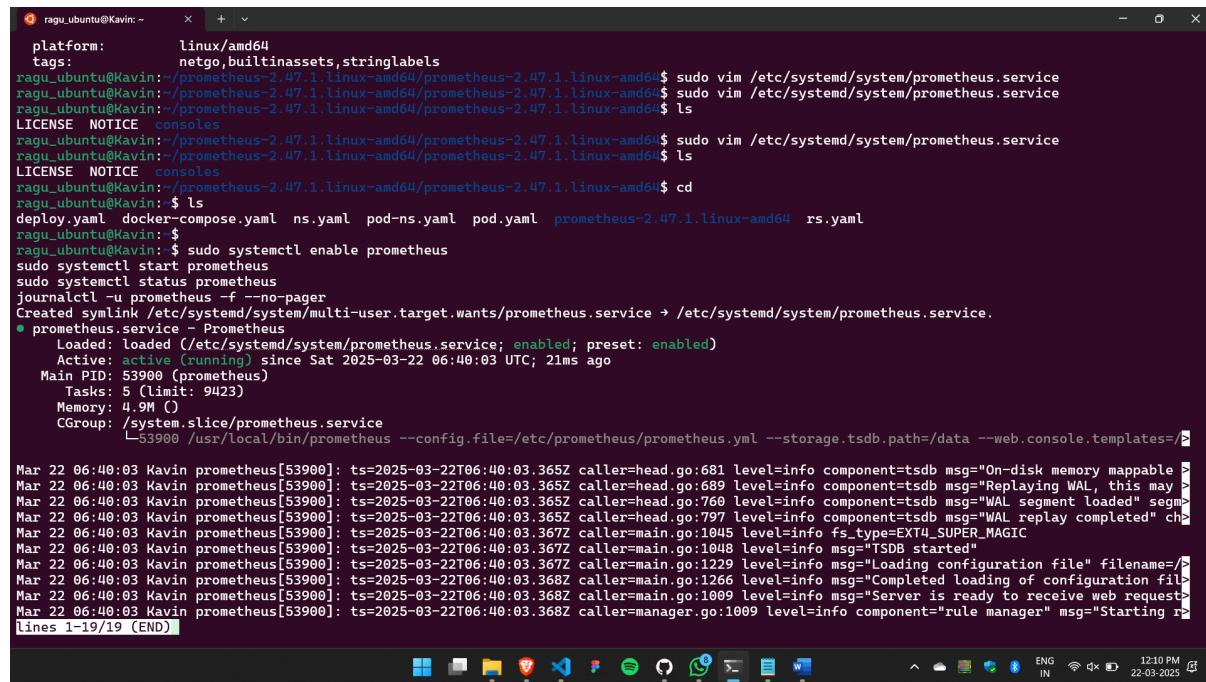
--web.console.libraries=/etc/prometheus/console_libraries \

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

--web.enable-lifecycle

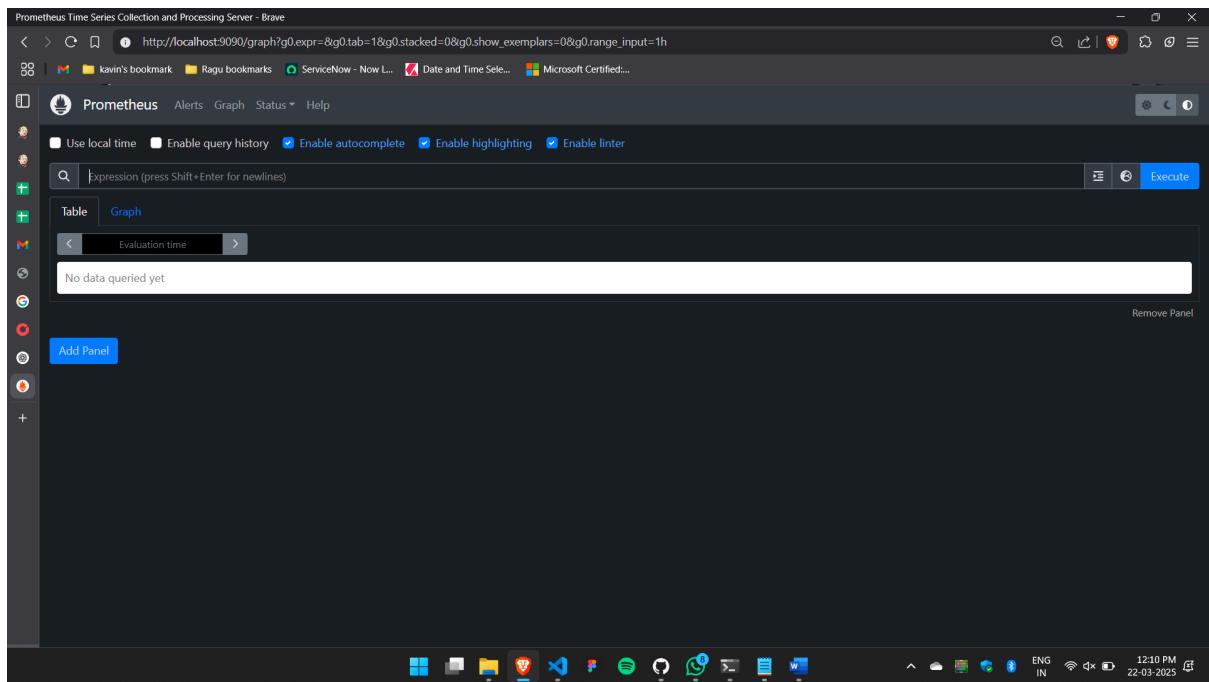
[Install]

WantedBy=multi-user.target



The screenshot shows a terminal window on a Linux system. The user is navigating through the Prometheus configuration files and starting the service. The terminal output includes:

- Setting environment variables: `platform: linux/amd64`, `tags: netgo,builtinassets,stringlabels`.
- Editing the Prometheus service file with `sudo vim /etc/systemd/system/prometheus.service`.
- Listing files with `ls`.
- Noticing license files: `LICENSE NOTICE consoles` and `LICENSE NOTICE prometheus`.
- Changing directory with `cd`.
- Deploying with `deploy.yaml docker-compose.yaml ns.yaml pod.yaml prometheus-2.47.1.linux-amd64 rs.yaml`.
- Enabling the service with `sudo systemctl enable prometheus`.
- Starting the service with `sudo systemctl start prometheus`.
- Checking status with `sudo systemctl status prometheus`.
- Using journalctl with `journalctl -u prometheus -f --no-pager`.
- Creating a symlink: `Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.`
- Checking the service status: `● prometheus.service - Prometheus`.
 - Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
 - Active: active (running) since Sat 2025-03-22 06:40:03 UTC; 21ms ago
 - Main PID: 53900 (prometheus)
 - Tasks: 5 (limit: 9423)
 - Memory: 4.9M
 - CGroup: /system.slice/prometheus.service
- Logs from the service:
 - On-disk memory mappable segments loaded.
 - Replaying WAL, this may take a while.
 - WAL segment loaded.
 - WAL replay completed.
 - TSDB started.
 - Loading configuration file.
 - Completed loading of configuration file.
 - Server is ready to receive web requests.
 - Starting rule manager.



Node Exporter:

Commands for installation:

```
sudo mv \
node_exporter-1.6.1.linux-amd64/node_exporter \
/usr/local/bin/
rm -rf node_exporter*
```

Node exporter file:

[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]

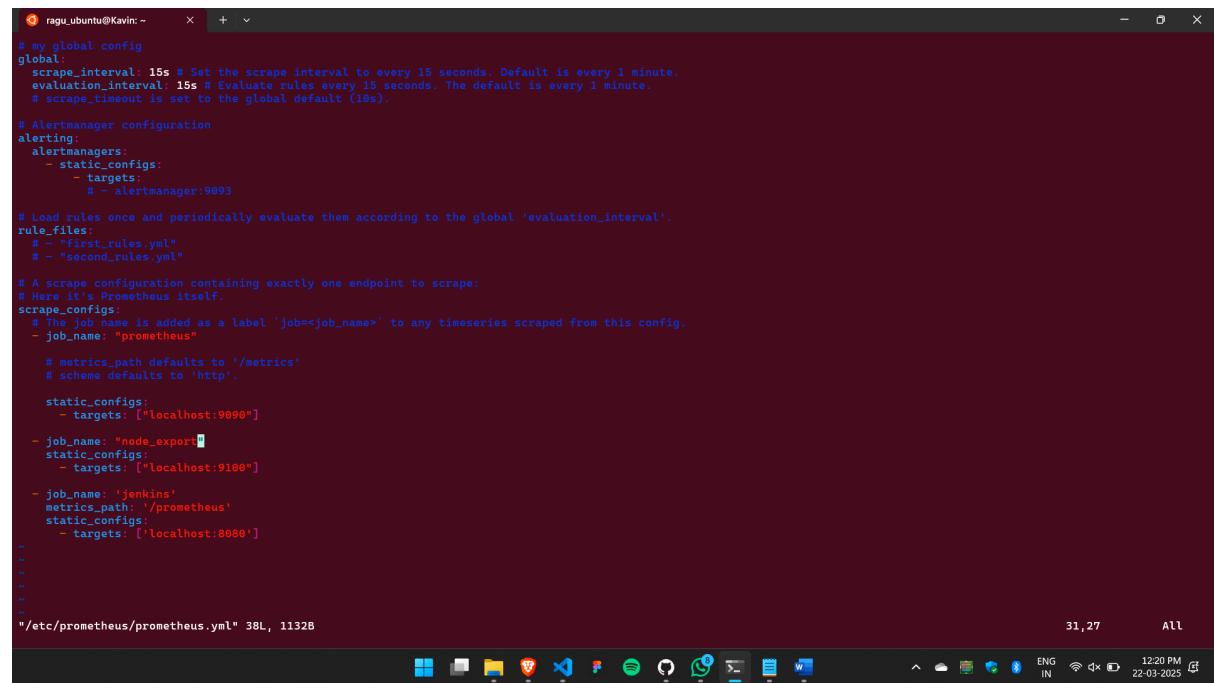
WantedBy=multi-user.target

Service check : sudo systemctl enable node_exporter

sudo systemctl start node_exporter

sudo systemctl status node_exporter

journalctl -u node_exporter -f --no-pager



```
# 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-exporter"
    static_configs:
      - targets: ["localhost:9100"]

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

# ...
# ...
# ...
# ...

"/etc/prometheus/prometheus.yml" 38L, 1132B
```

Reload Prometheus: curl -X POST <http://localhost:9090/-/reload>

The screenshot shows the Prometheus Targets page. It lists three scrape pools: Jenkins (0/1 up), node_export (1/1 up), and prometheus (1/1 up). Each pool has a table with columns: Endpoint, State, Labels, Last Scrape, Scrape Duration, and Error.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8080/prometheus	DOWN	instance="localhost:8080" job="jenkins"	32.331s ago	6.208ms	server returned HTTP status 403 Forbidden

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_export"	24.854s ago	38.140ms	

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	20.766s ago	5.218ms	

Grafana:

```
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
```

Grafana UI:

The screenshot shows the Grafana home page. On the left is a sidebar with navigation links: Home, Bookmarks, Starred, Dashboards, Explore, Alerting, Connections, and Administration. The main content area has a title "Welcome to Grafana". Below it, there's a "Basic" section with a brief introduction and links to "TUTORIAL", "DATA SOURCE AND DASHBOARDS", and "Grafana fundamentals". To the right are three panels: "DATA SOURCES" (with a link to "Add your first data source"), "DASHBOARDS" (with a link to "Create your first dashboard"), and "Latest from the blog" (a card for "Mar 21" about "Grafana Labs at KubeCon: Prometheus, OpenTelemetry, AI, and more"). At the bottom of the main content area, there are sections for "Dashboards" (Starred dashboards and Recently viewed dashboards) and "Latest from the blog". The status bar at the bottom shows system icons and the date/time.

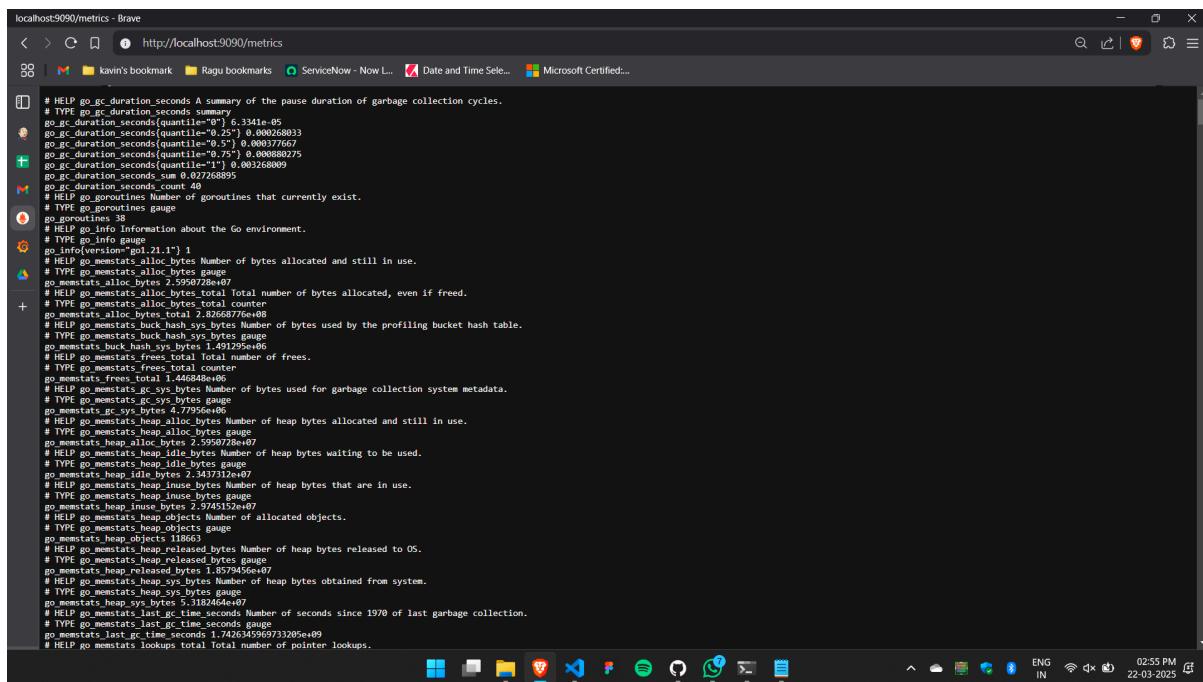
Status in promethues:

The screenshot shows the Prometheus Targets page. The top navigation bar includes links for Prometheus, Alerts, Graph, Status, and Help. The main content is titled "Targets" and displays three groups of targets:

- jenkins (0/1 up)**:
- Endpoint: http://localhost:8080/prometheus, State: DOWN, Labels: instance="localhost:8080", job="jenkins". Last Scrape: -872.000ms ago, Scrape Duration: 1.723ms, Error: server returned HTTP status 403 Forbidden.
- node_export (1/1 up)**:
- Endpoint: http://localhost:9100/metrics, State: UP, Labels: instance="localhost:9100", job="node_export". Last Scrape: 9.665s ago, Scrape Duration: 22.271ms.
- prometheus (1/1 up)**:
- Endpoint: http://localhost:9090/metrics, State: UP, Labels: instance="localhost:9090", job="prometheus". Last Scrape: 5.576s ago, Scrape Duration: 3.332ms.

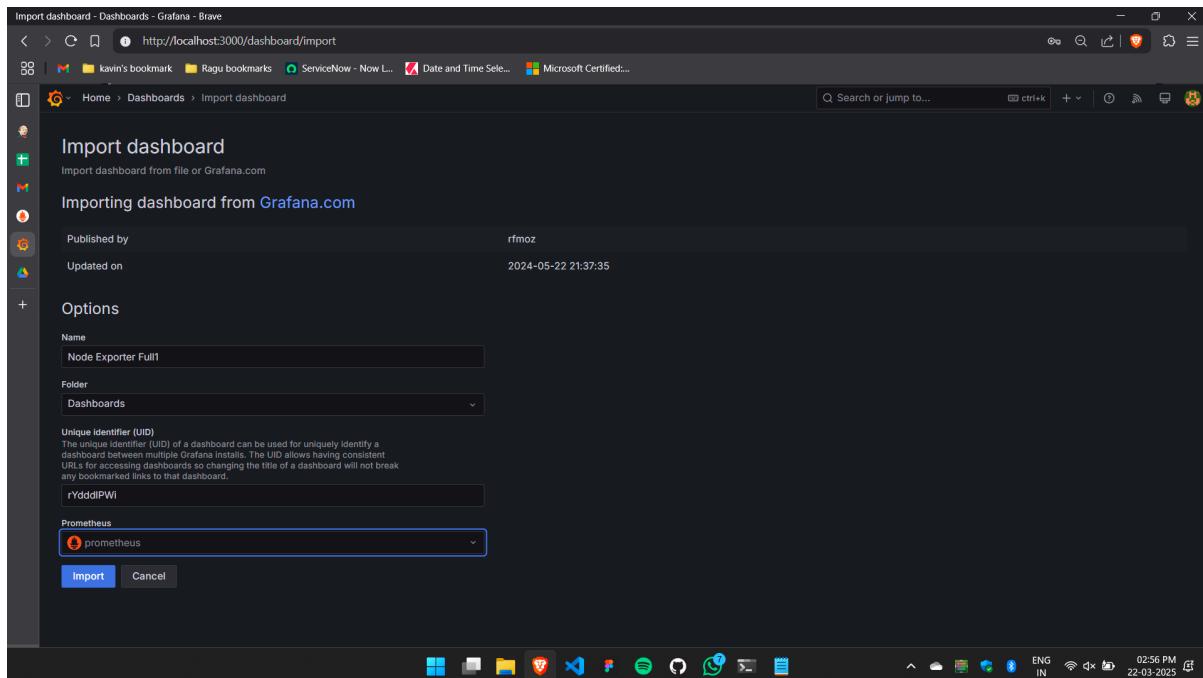
The status bar at the bottom shows system icons and the date/time.

Metrics in Prometheus:

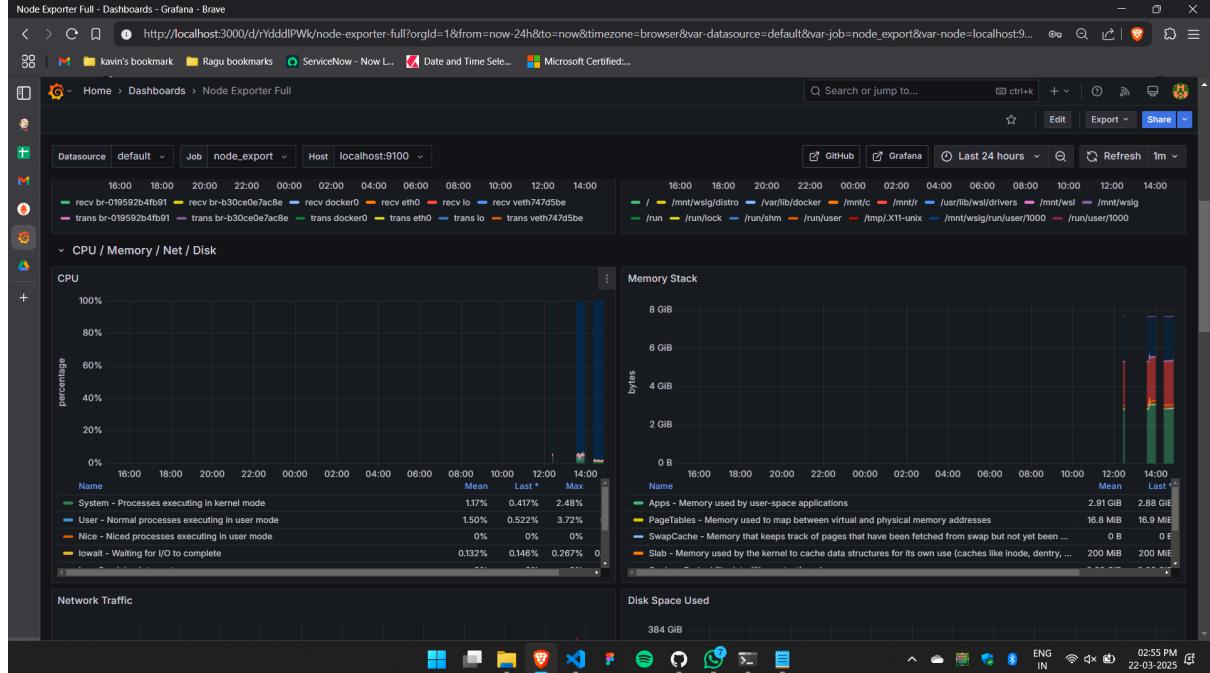
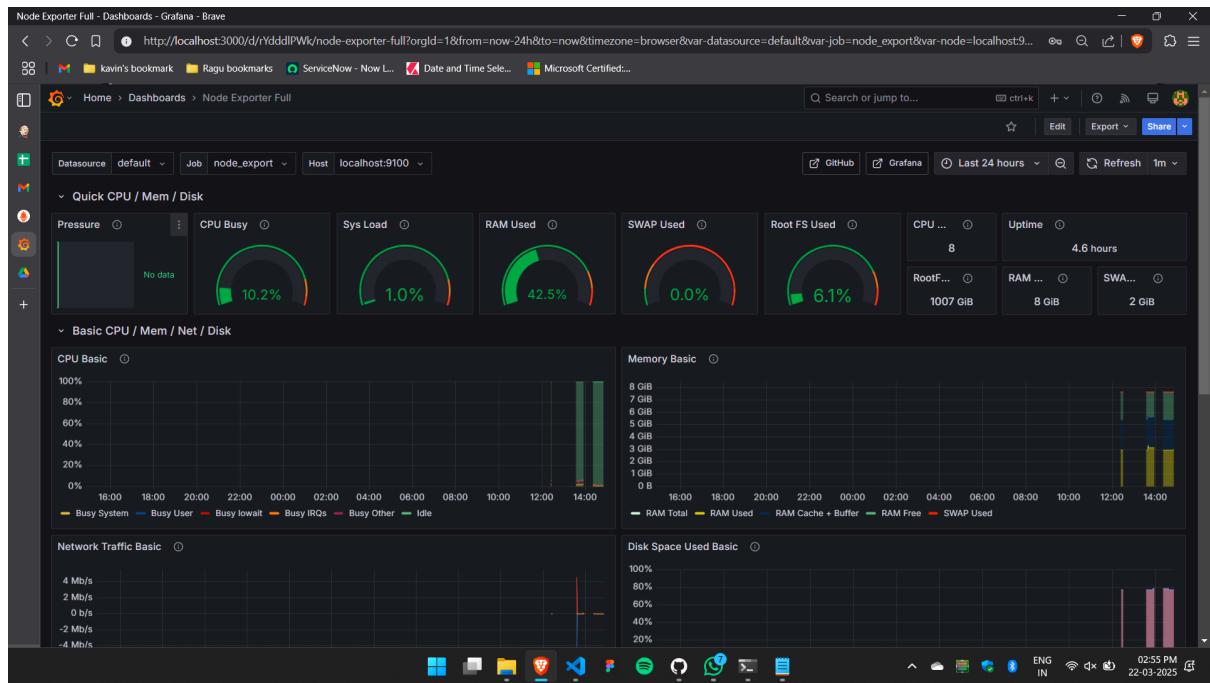


```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 6.3341e-05
go_gc_duration_seconds{quantile="0.25"} 0.000268033
go_gc_duration_seconds{quantile="0.5"} 0.00077667
go_gc_duration_seconds{quantile="0.75"} 0.000880275
go_gc_duration_seconds{quantile="1"} 0.00368809
go_gc_duration_seconds_sum 0.02726889
go_gc_duration_seconds_sum 0.02726889
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 38
# HELP go_info Information about the Go environment.
# TYPE go_info info
go_infriverion="go1.21.1"
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.82658776e+08
# 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 2.82658776e+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 1.491295e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.4469480496
# 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 1.7770000000000002e-07
# HELP go_memstats_heap_all_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_all_bytes gauge
go_memstats_heap_all_bytes 2.5950728e+07
# HELP go_memstats_heap_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_bytes gauge
go_memstats_heap_bytes 2.3437312e+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 1.8574556e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 11863
# 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 1.8574556e+07
# 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 1.824264e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.742634569733205e+09
# HELP go_memstats_locks_total Total number of pointer lookups.
```

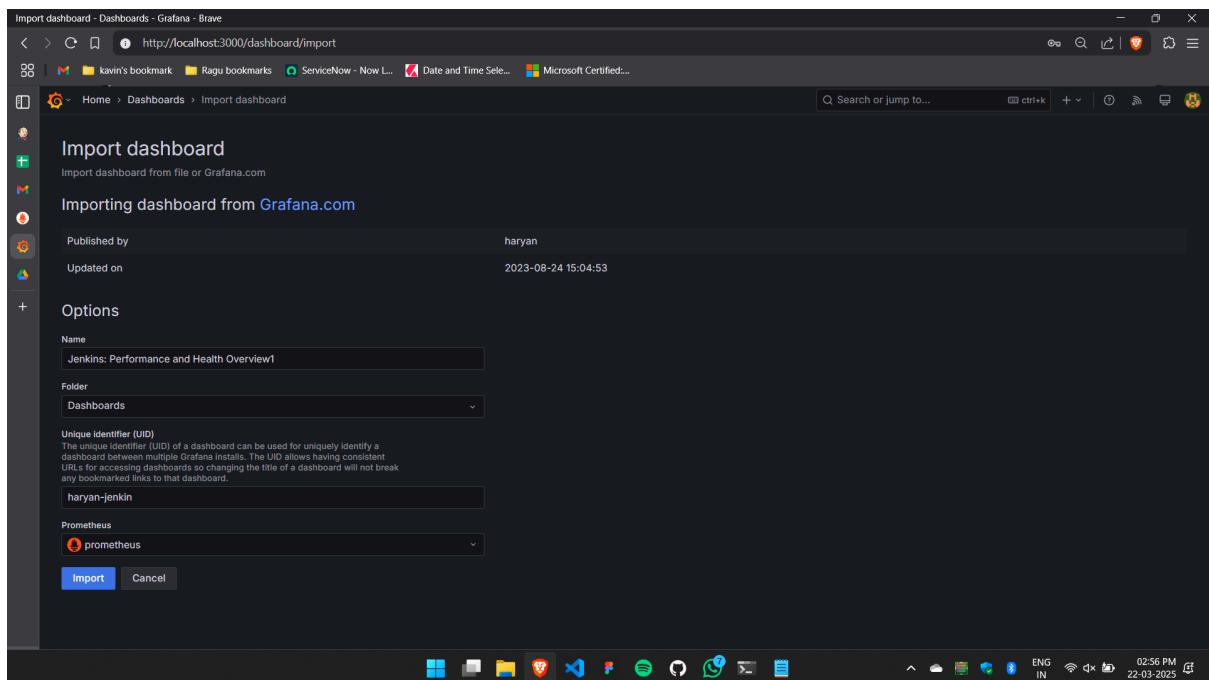
Node Exporter:



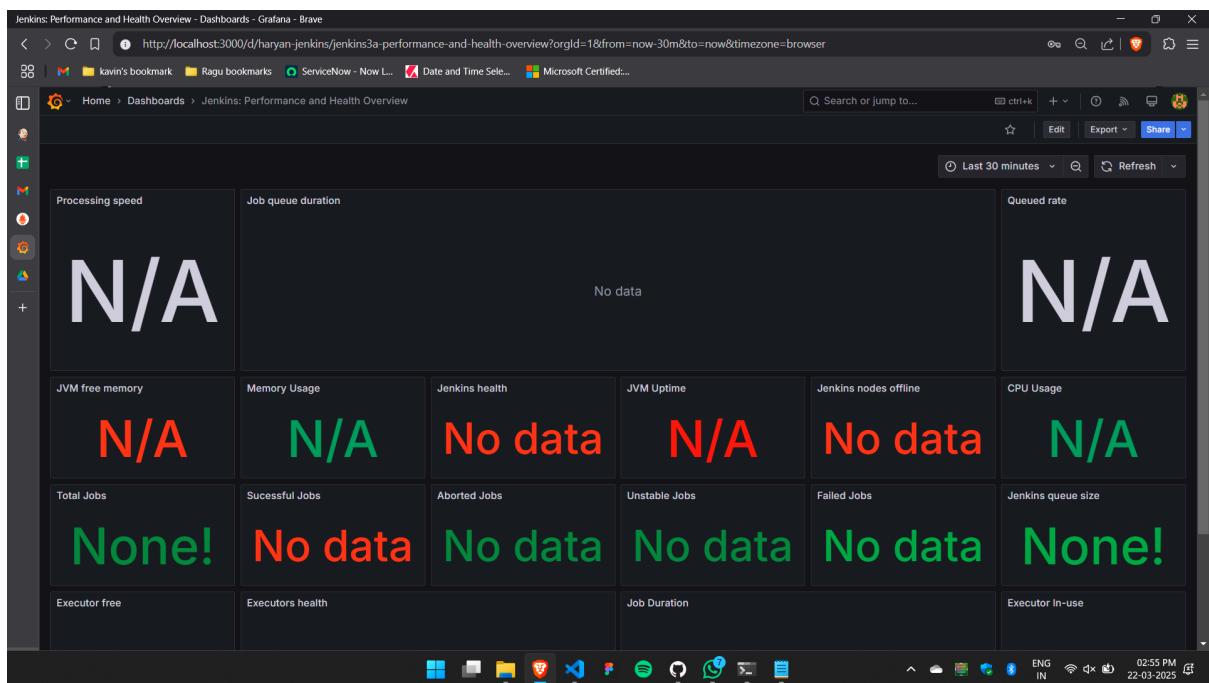
Dashboard:



Jenkins Overview:

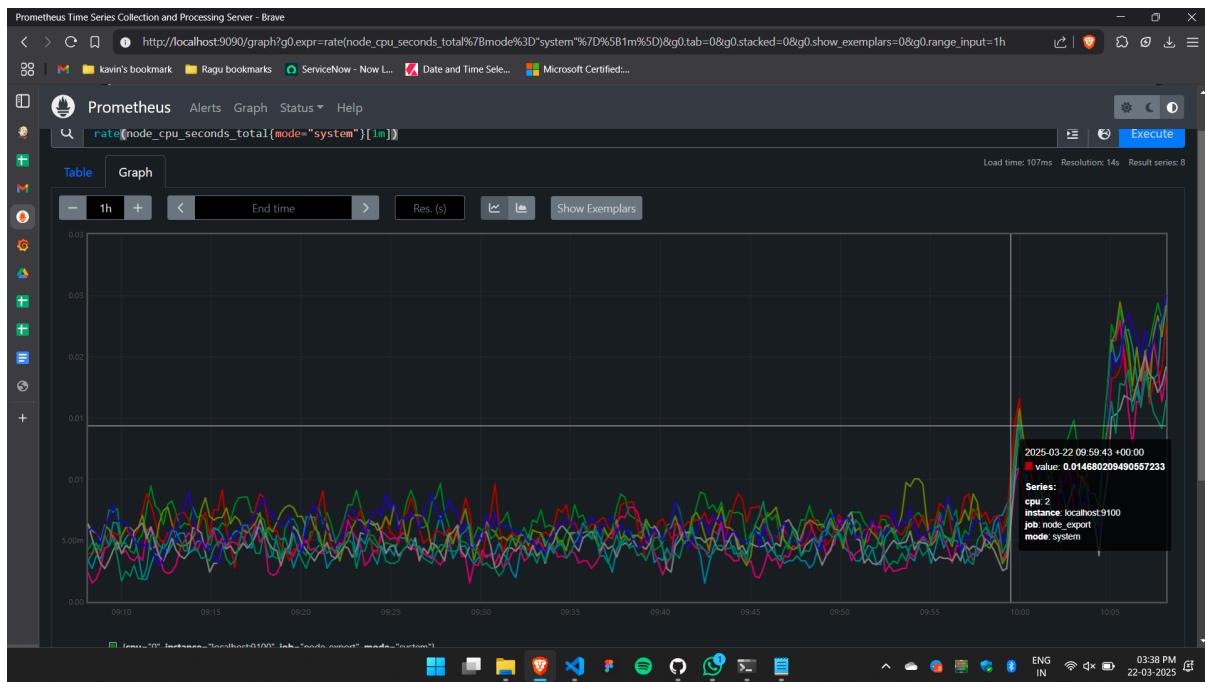


Dashboard:

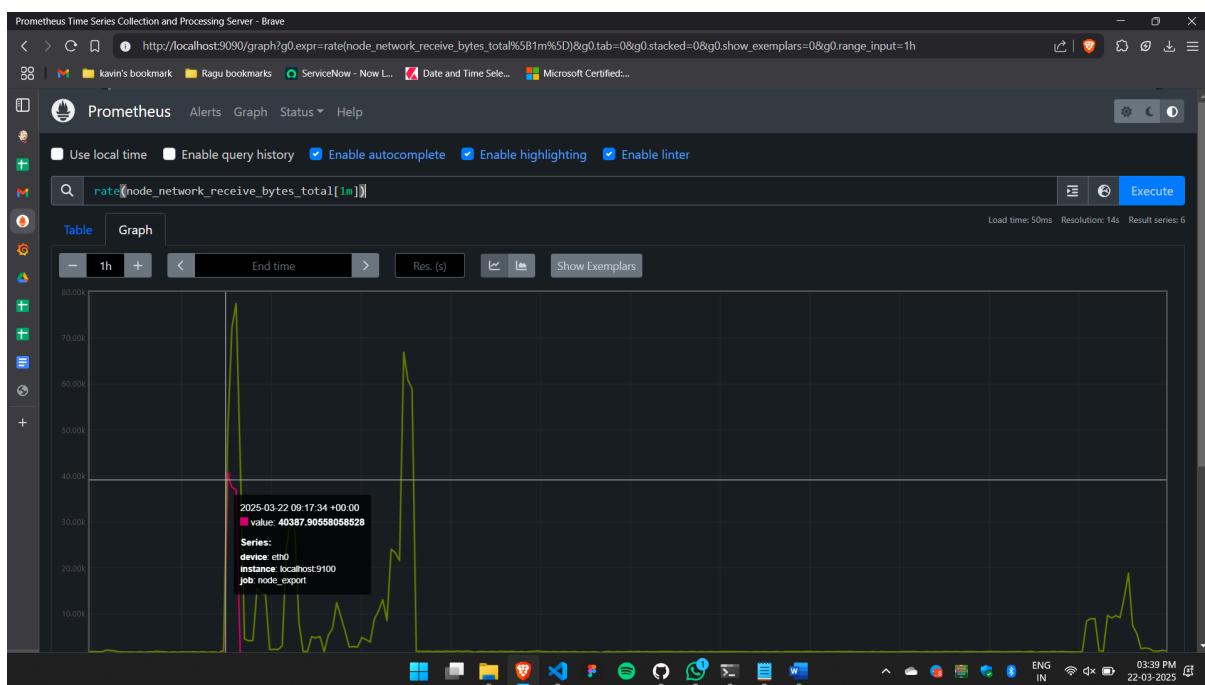


Prometheus analysis:

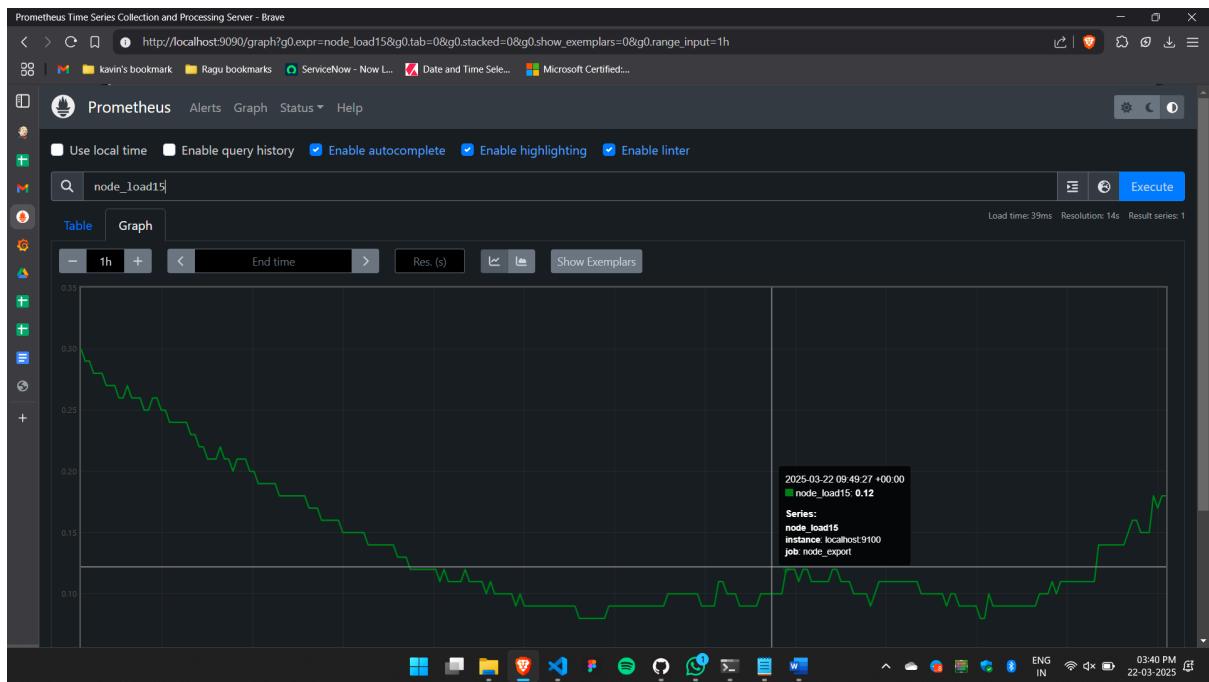
```
rate(node_cpu_seconds_total{mode="system"}[1m])
```



rate(node_network_receive_bytes_total[1m])

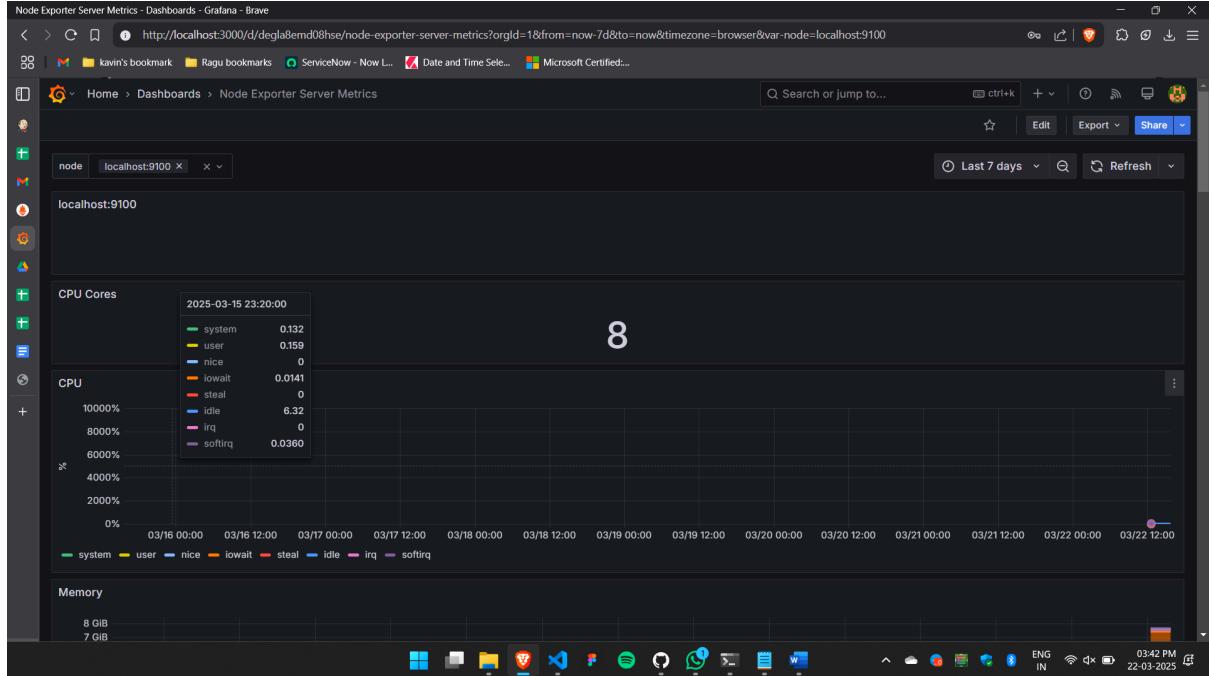


node_load15



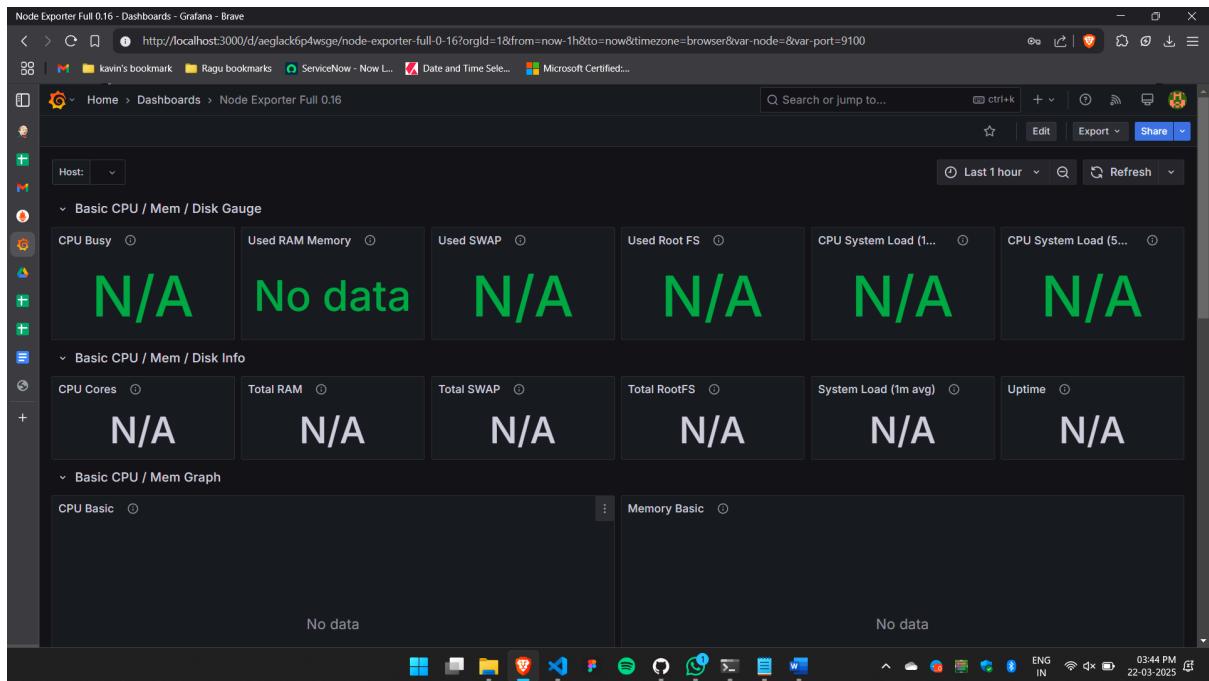
Dashboard: (405)

Node Exporter Service metrics,



Dashboard (5174):

Node Exporter Full 0.16,



Dashboard (9096):

1 Node Exporter 1.0.1

