

DAY 6 | 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.

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
```

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 \

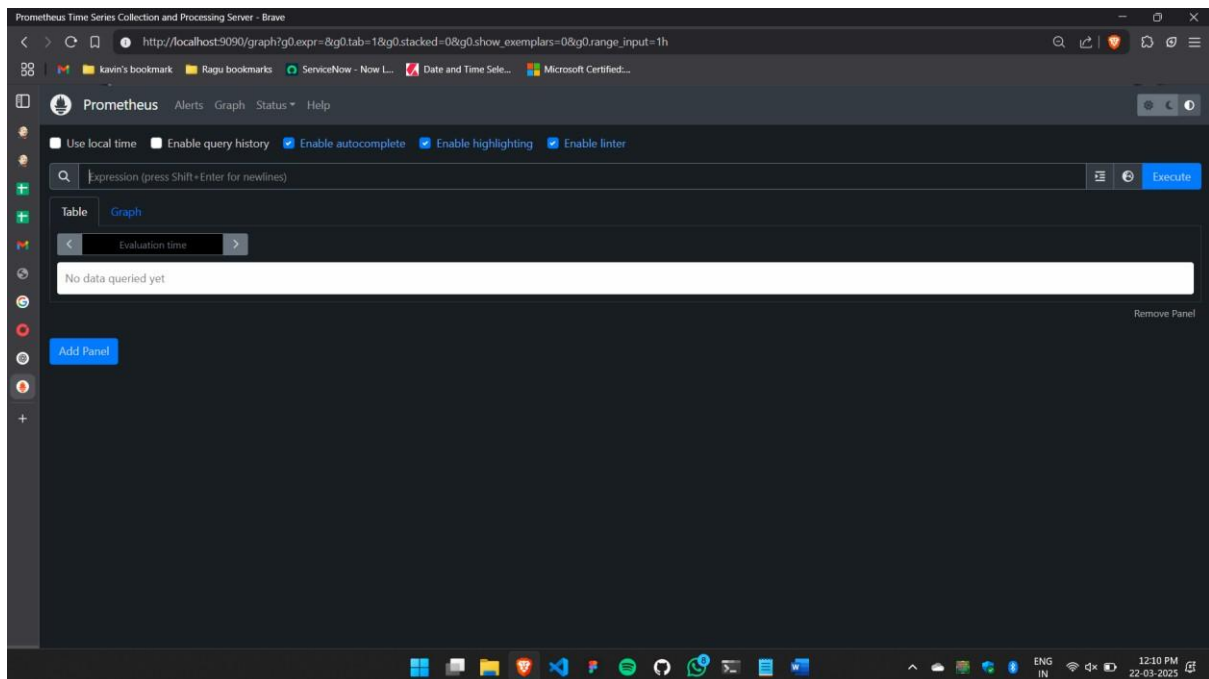
--web.console.libraries=/etc/prometheus/console_libraries \

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

--web.enable-lifecycle

[Install]

WantedBy=multi-user.target



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]

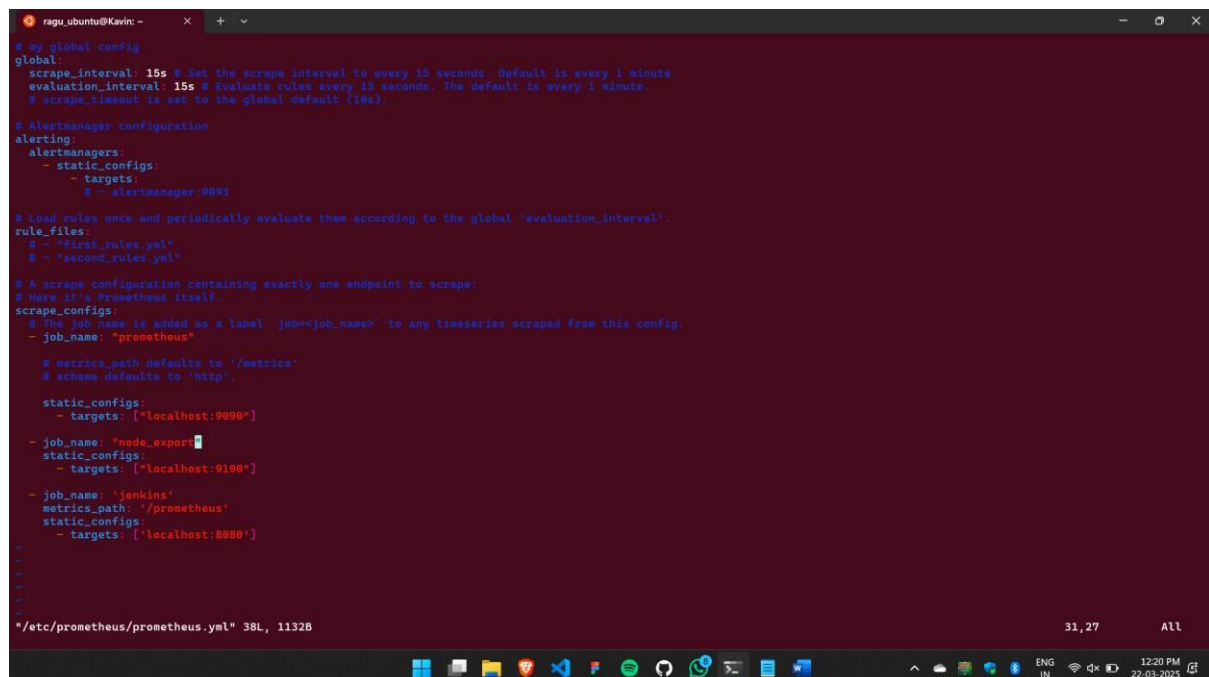
WantedBy=multi-user.target

Service check : sudo systemctl enable node_exporter

sudo systemctl start node_exporter sudo systemctl

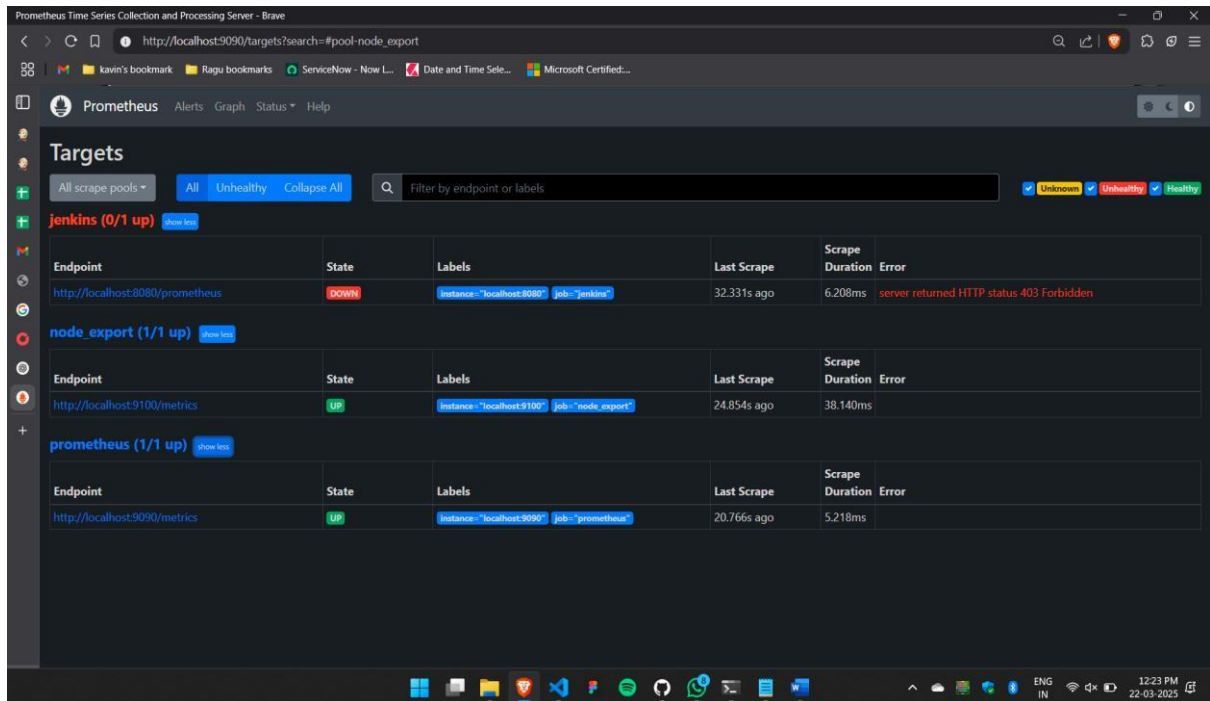
status node_exporter journalctl -u node_exporter -f -

-no-pager

A screenshot of a terminal window with a dark background. The terminal shows the content of the Prometheus configuration file located at /etc/prometheus/prometheus.yml. The configuration includes global settings like scrape_interval (15s) and evaluation_interval (15s), alerting configurations, rule_files, and scrape_configs. The scrape_configs section lists three jobs: 'prometheus' (scraping Prometheus itself), 'node_export' (scraping node metrics), and 'jenkins' (scraping Jenkins metrics). The terminal window title is 'ragu_ubuntu@Kavin: ~'. The bottom of the terminal shows the file path and line numbers: '/etc/prometheus/prometheus.yml' 38L, 1132B. The system tray at the bottom indicates the time is 12:28 PM on 22-03-2024.

```
ragu_ubuntu@Kavin: ~  
# 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, 1132B
```

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



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
```

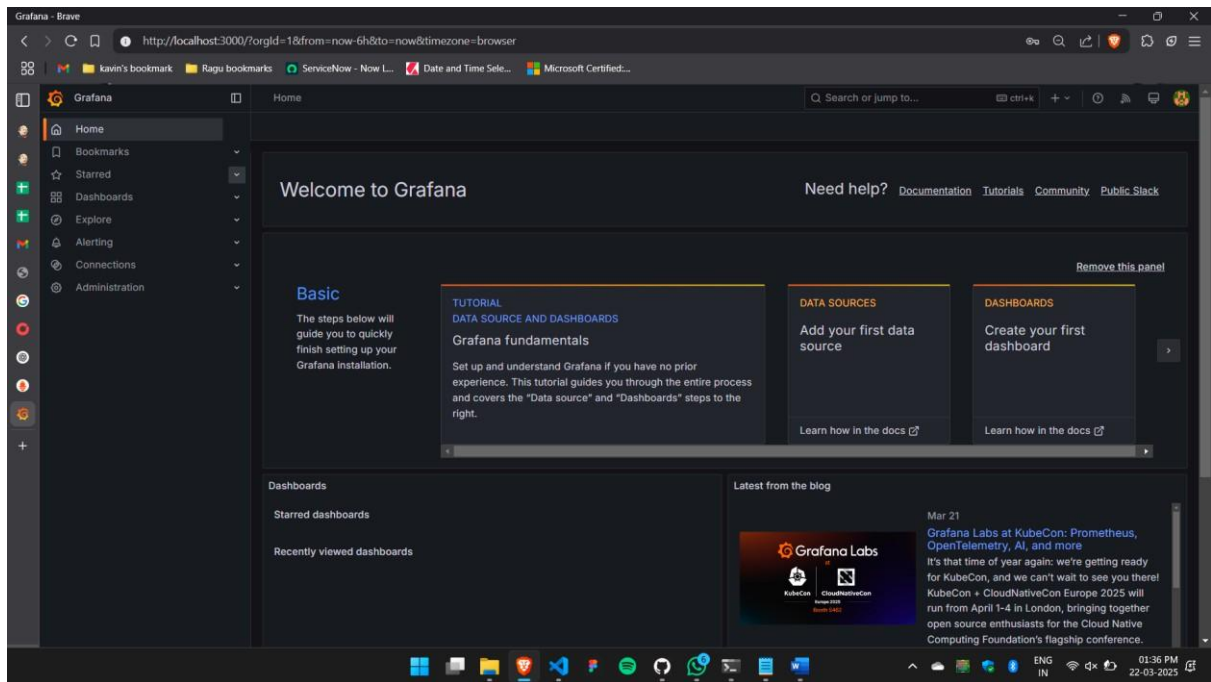
```
sudo apt-get update sudo apt-get -y
```

```
install grafana sudo systemctl enable
```

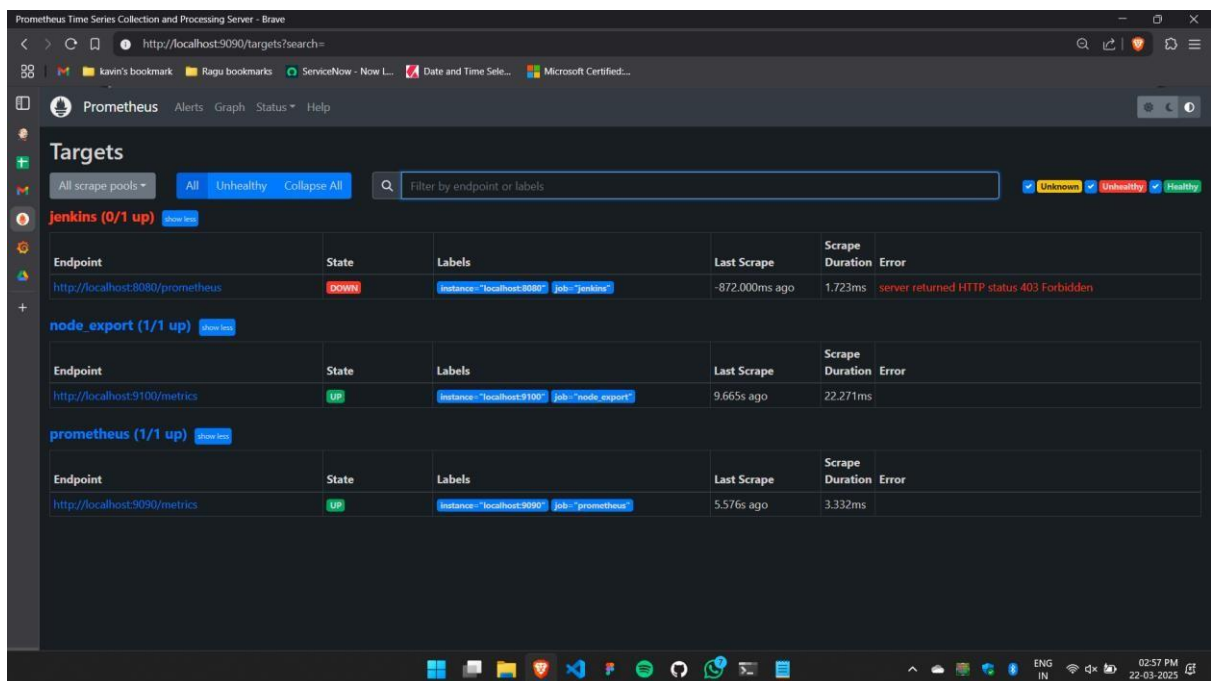
```
grafana-server sudo systemctl start
```

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

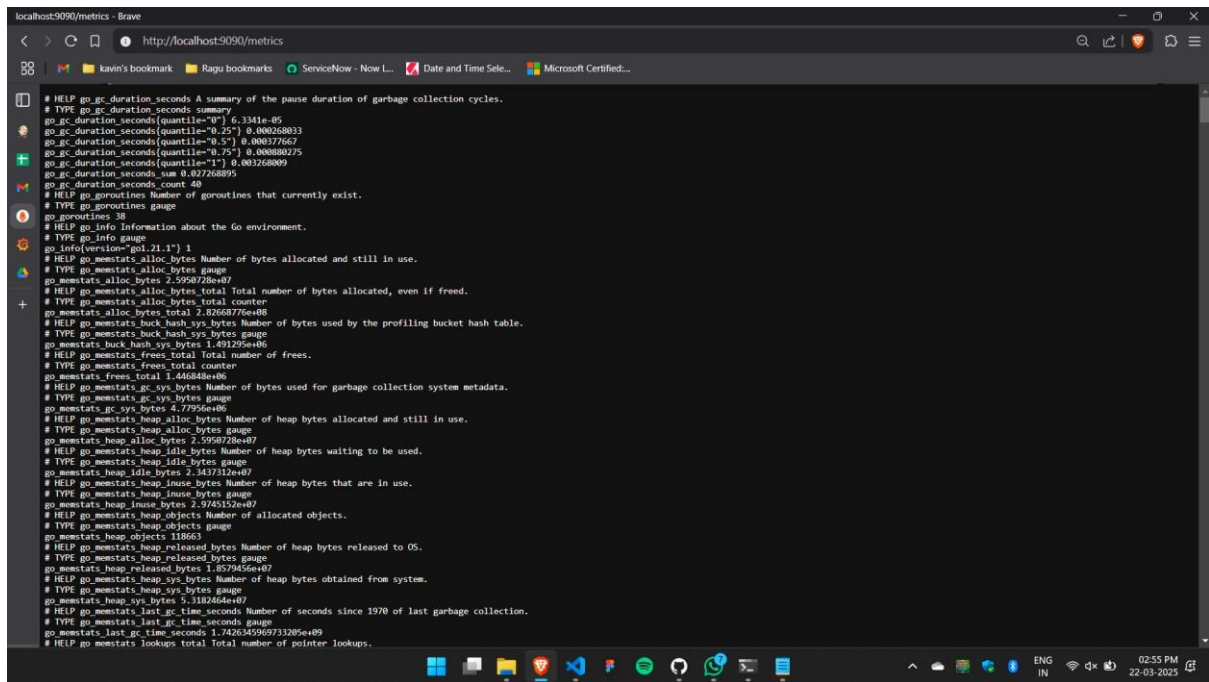
```
grafana-server Grafana UI:
```



Status in promethues:



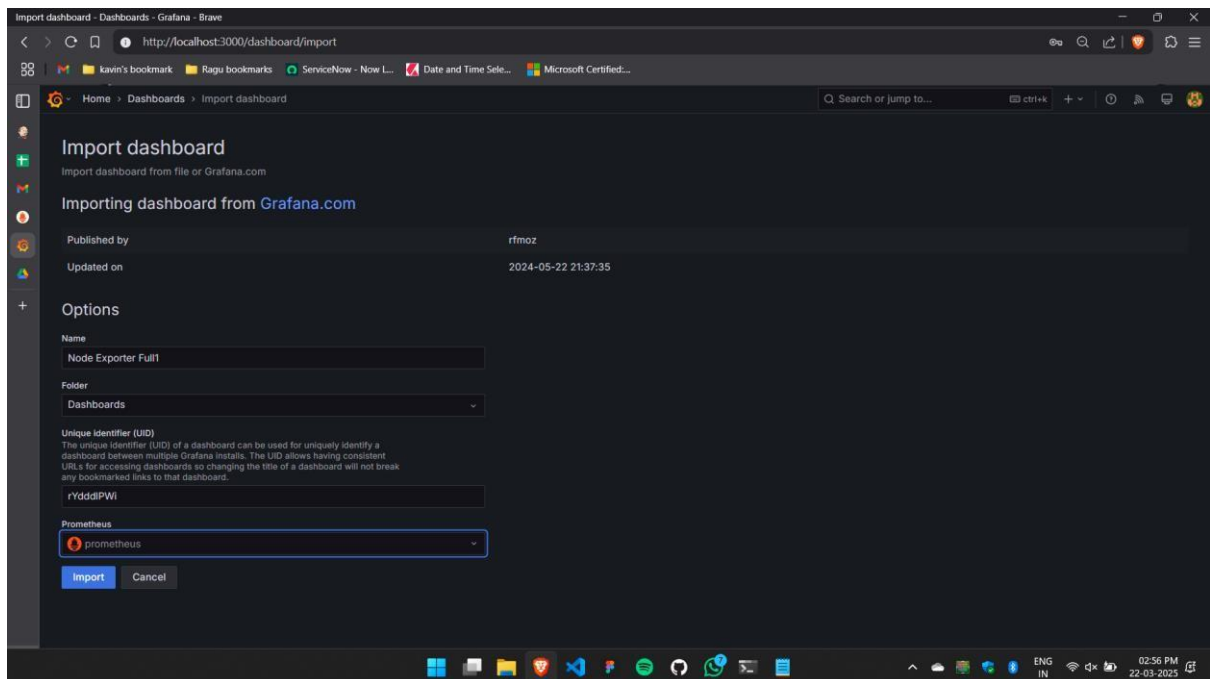
Metrics in Prometheus:



The screenshot shows a web browser window with the URL `http://localhost:9090/metrics`. The page displays a list of Prometheus metrics for a Go application. The metrics include:

- `go_gc_duration_seconds`: A summary of the pause duration of garbage collection cycles. It includes quantiles (0, 0.25, 0.5, 0.75, 1) and a sum.
- `go_goroutines`: The number of goroutines that currently exist.
- `go_info`: Information about the Go environment.
- `go_memstats_alloc_bytes`: The number of bytes allocated and still in use.
- `go_memstats_alloc_bytes_total`: The total number of bytes allocated, even if freed.
- `go_memstats_buck_hash_sys_bytes`: The number of bytes used by the profiling bucket hash table.
- `go_memstats_buck_hash_sys_bytes`: The number of bytes used by the profiling bucket hash table.
- `go_memstats_frees_total`: The total number of frees.
- `go_memstats_gc_sys_bytes`: The number of bytes used for garbage collection system metadata.
- `go_memstats_heap_alloc_bytes`: The number of heap bytes allocated and still in use.
- `go_memstats_heap_alloc_bytes`: The number of heap bytes allocated and still in use.
- `go_memstats_heap_idle_bytes`: The number of heap bytes waiting to be used.
- `go_memstats_heap_inuse_bytes`: The number of heap bytes that are in use.
- `go_memstats_heap_inuse_bytes`: The number of heap bytes that are in use.
- `go_memstats_heap_objects`: The number of allocated objects.
- `go_memstats_heap_released_bytes`: The number of heap bytes released to OS.
- `go_memstats_heap_sys_bytes`: The number of heap bytes obtained from system.
- `go_memstats_last_gc_time_seconds`: The number of seconds since 1970 of last garbage collection.
- `go_memstats_lookups_total`: The total number of pointer lookups.

Node Exporter:

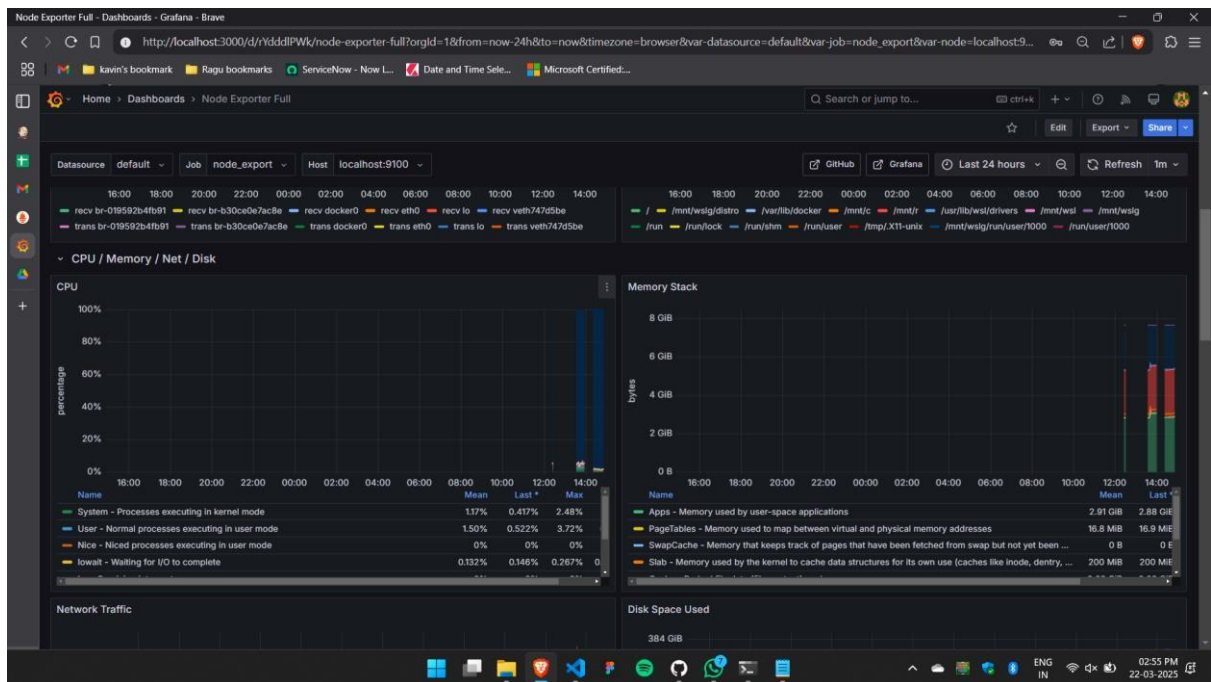
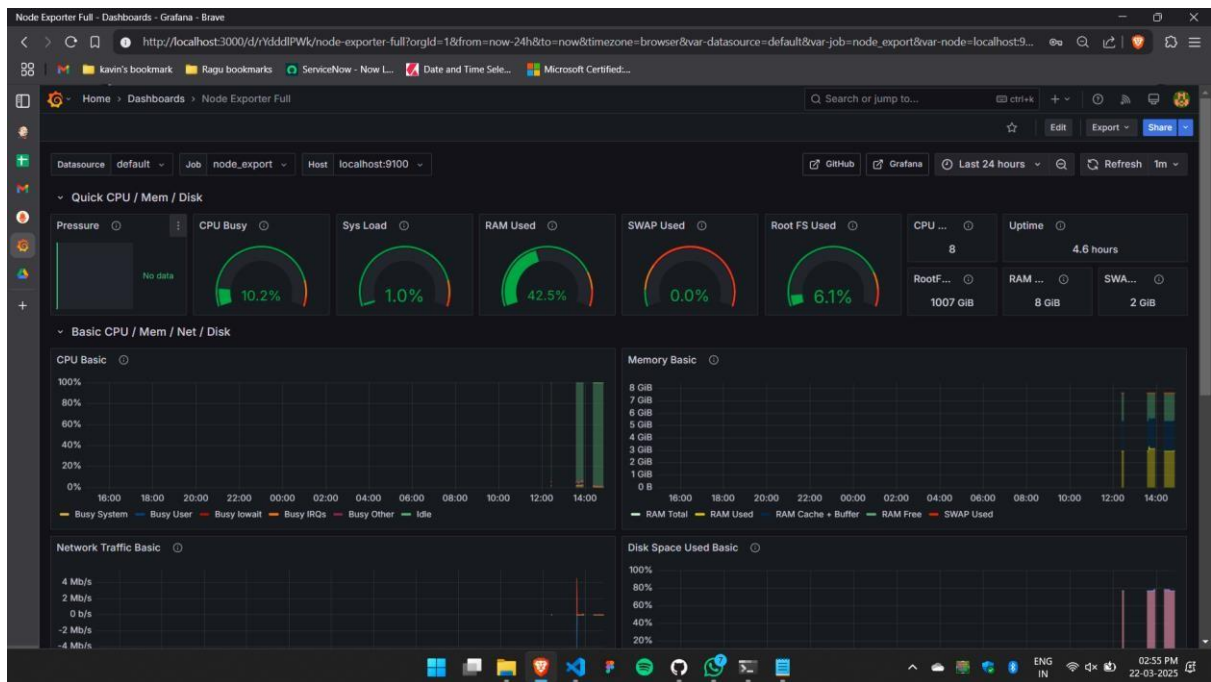


The screenshot shows the 'Import dashboard' dialog in Grafana. The dialog is titled 'Import dashboard' and has a subtitle 'Import dashboard from file or Grafana.com'. The 'Importing dashboard from Grafana.com' section is active. The 'Published by' field shows 'rfmoz' and the 'Updated on' field shows '2024-05-22 21:37:35'. The 'Options' section is expanded, showing the following fields:

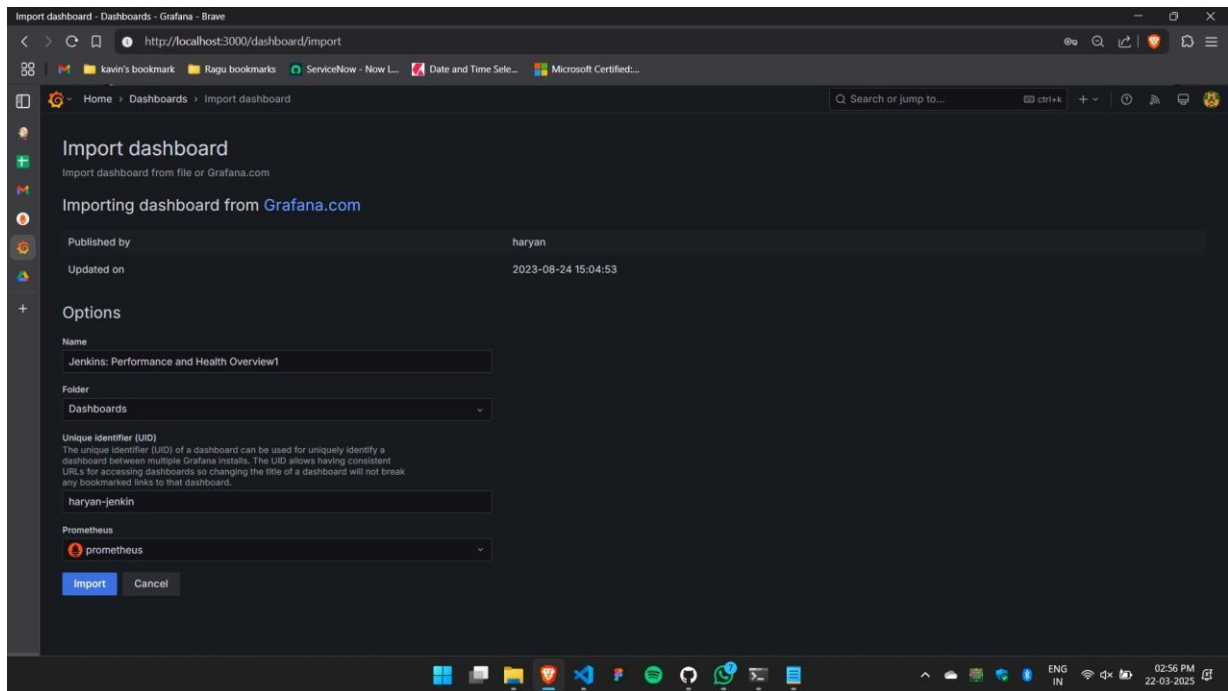
- Name**: Node Exporter Full
- Folder**: Dashboards
- Unique Identifier (UID)**: rYdddIPWl
- Prometheus**: prometheus

The 'Import' button is highlighted in blue.

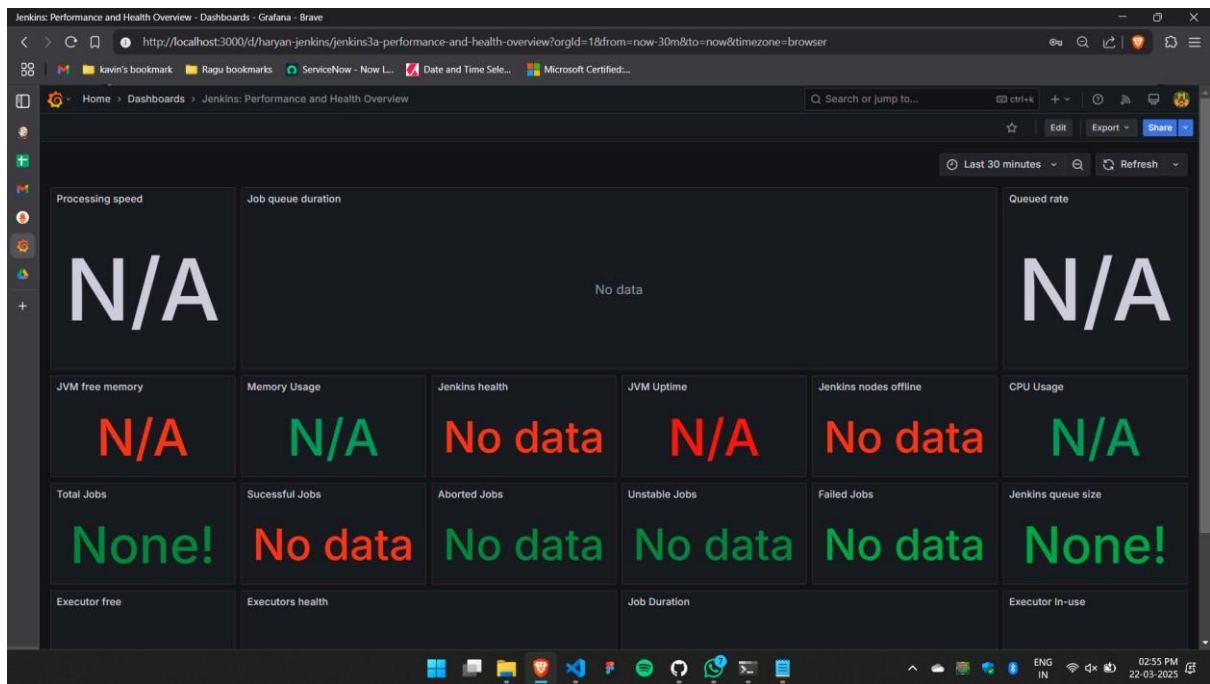
Dashboard:



Jenkins Overview:

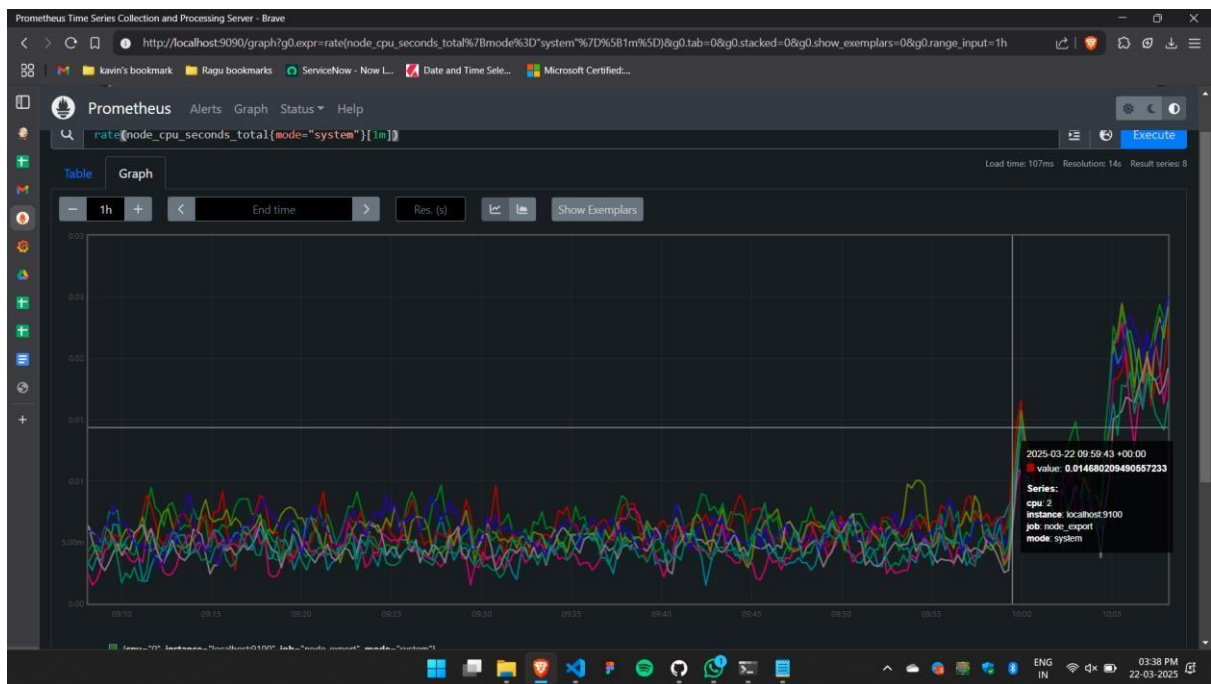


Dashboard:

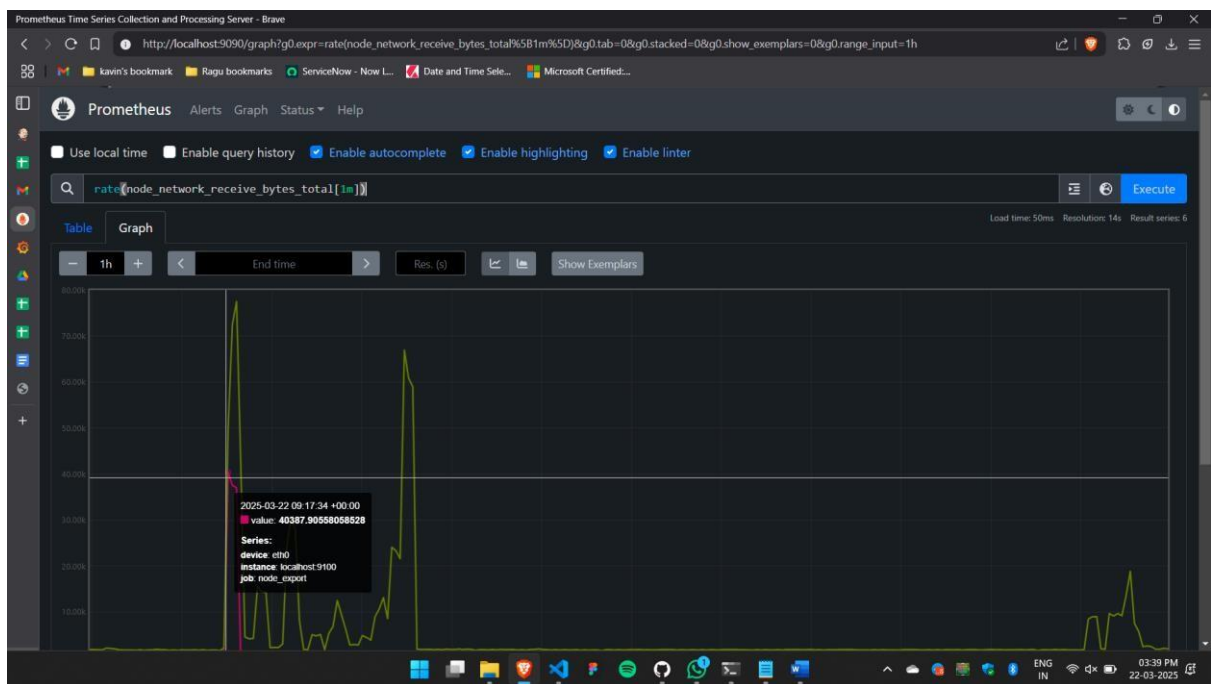


Prometheus analysis:

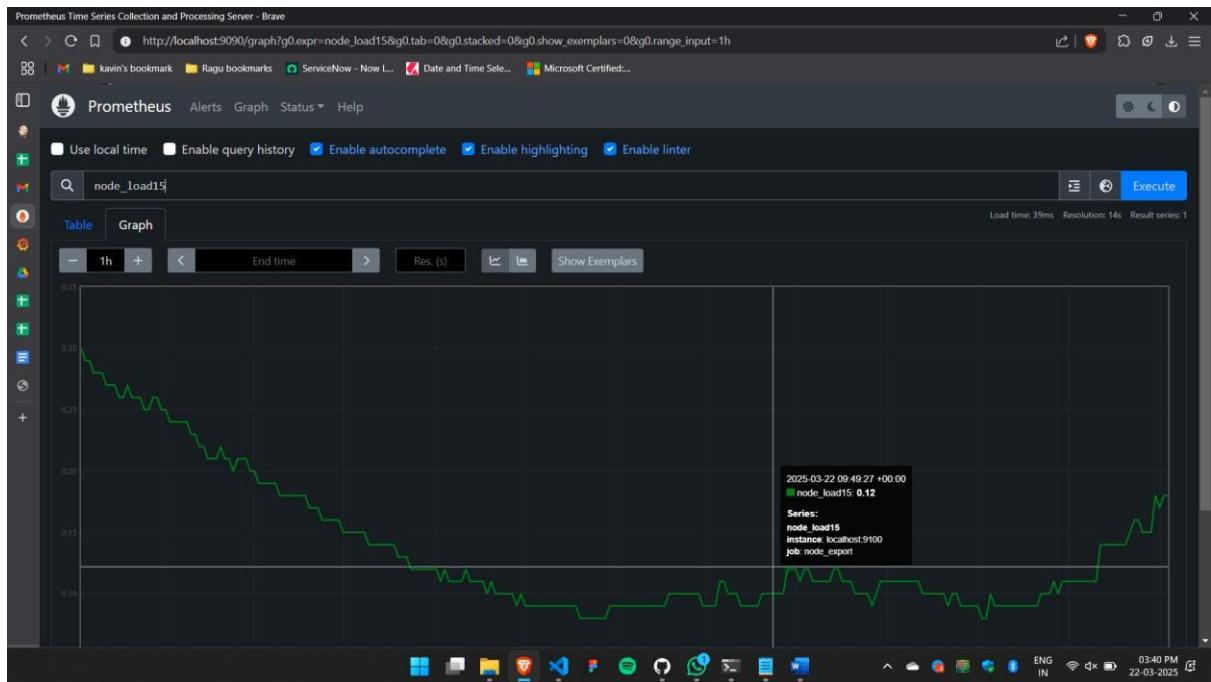
```
rate(node_cpu_seconds_total{mode="system"}[1m])
```



`rate(node_network_receive_bytes_total[1m])`

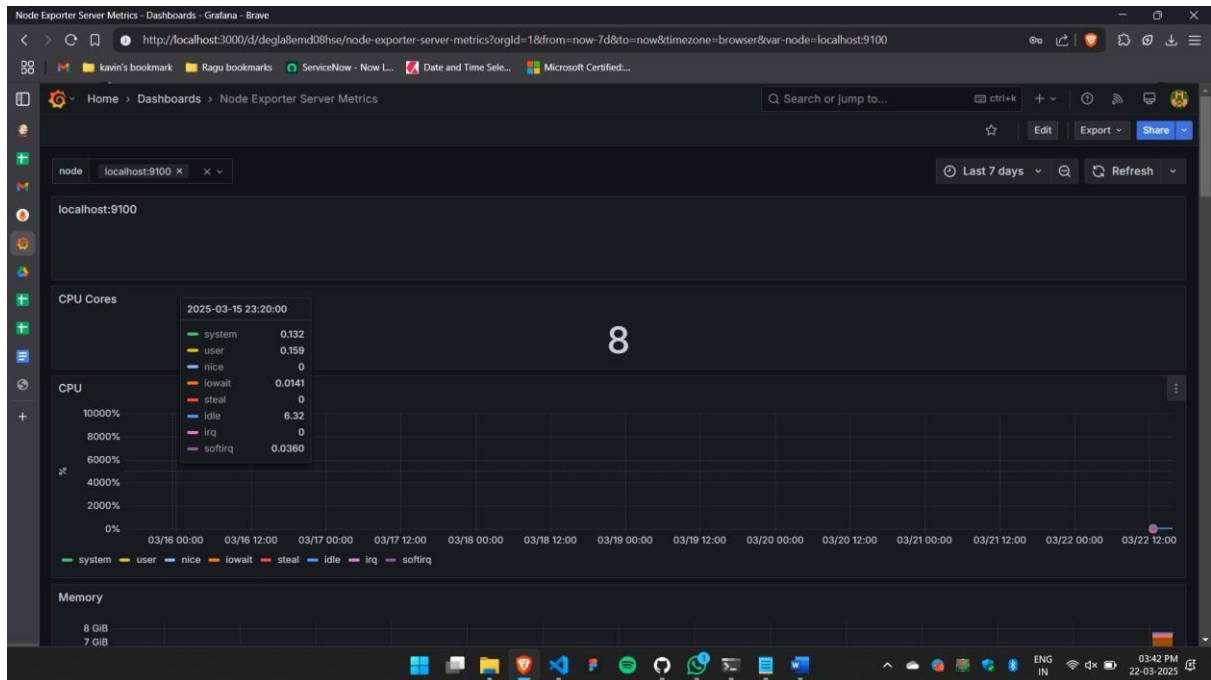


`node_load15`



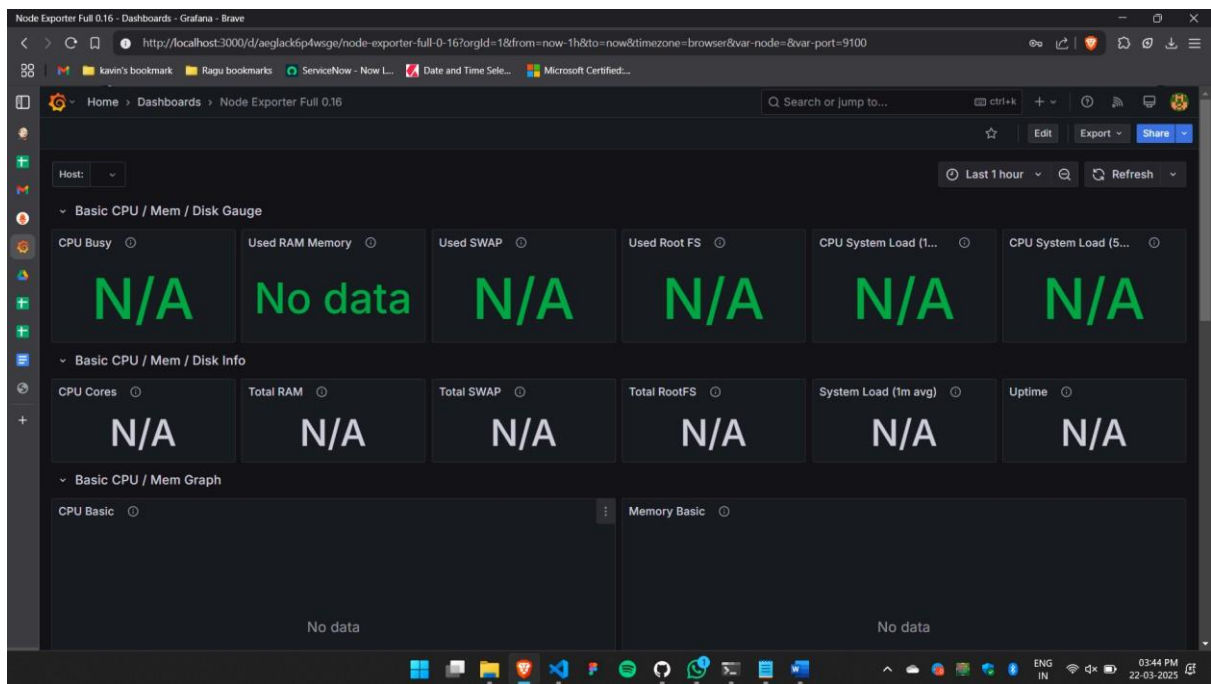
Dashboard: (405)

Node Exporter Service metrics,



Dashboard (5174):

Node Exporter Full 0.16,



Dashboard (9096):

1 Node Exporter 1.0.1

