🚀 Prometheus Setup Guide with Node Exporter and Jenkins Monitoring

This guide will help you properly configure **Prometheus**, fix your YAML errors, and ensure correct scraping of **Node Exporter** and **Jenkins metrics**.

✅ 1. `prometheus.yml` Configuration

```
# 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"
    static_configs:
      - targets: ["localhost:9090"]


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


  - job_name: "jenkins"
```

```
    metrics_path: "/prometheus"

    static_configs:

      - targets: ["<jenkins-ip>:8080"]
```

✅ 2. Validate Configuration

After updating `prometheus.yml`, run:

```
promtool check config /etc/prometheus/prometheus.yml
```

If the output is:

```
Checking /etc/prometheus/prometheus.yml
  SUCCESS: 0 rule files found
```

You're good to go! ✅

✅ 3. Restart Prometheus

```
sudo systemctl restart prometheus
sudo systemctl status prometheus
```

If it's running fine, you should see:

```
Active: active (running) since ...
```

✅ 4. Access Prometheus Dashboard

Go to http://localhost:9090

- Click **"Status"** → **"Targets"**
- Ensure `prometheus`, `node_exporter`, and `jenkins` show **UP**. ✅

✅ 5. Check Node Exporter Metrics

Test if **Node Exporter** is running:

```
curl http://localhost:9100/metrics | head -n 10
```

You should see CPU, memory, and system stats.

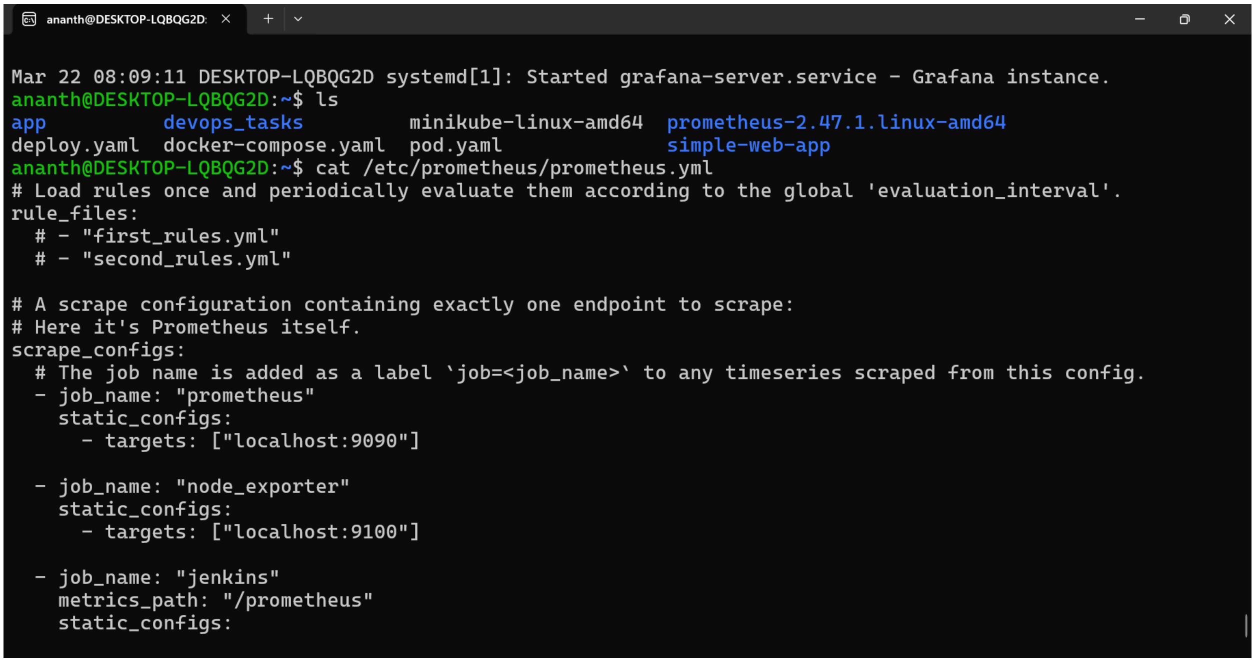
✅ 6. Check Jenkins Metrics

If **Jenkins Metrics Plugin** is installed:

```
curl http://<JENKINS-IP>:8080/prometheus
```

You should see **Jenkins performance metrics**.

🎉 **Prometheus is now correctly monitoring your system!** 🚀
Let me know if you need any more help. 😊

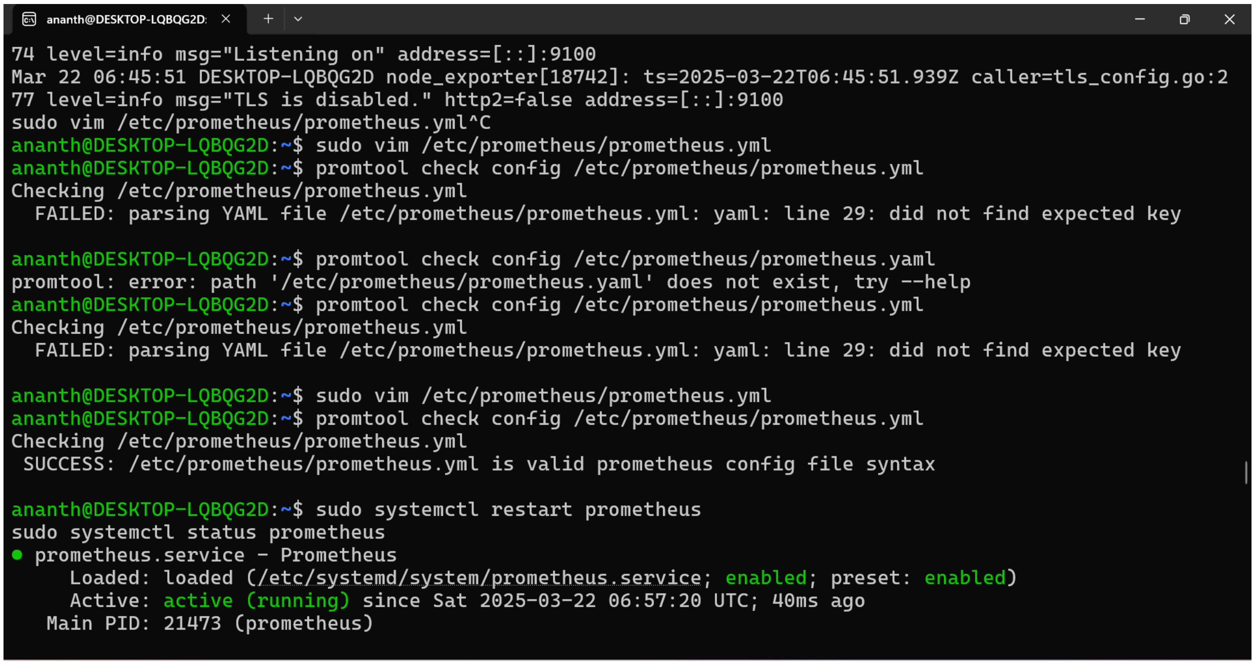


```
Mar 22 08:09:11 DESKTOP-LQBQG2D systemd[1]: Started grafana-server.service - Grafana instance.
ananth@DESKTOP-LQBQG2D:~$ ls
app          devops_tasks         minikube-linux-amd64  prometheus-2.47.1.linux-amd64
deploy.yaml  docker-compose.yaml  pod.yaml              simple-web-app
ananth@DESKTOP-LQBQG2D:~$ cat /etc/prometheus/prometheus.yml
# 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"
    static_configs:
      - targets: ["localhost:9090"]

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

  - job_name: "jenkins"
    metrics_path: "/prometheus"
    static_configs:
```



```
74 level=info msg="Listening on" address=[::]:9100
Mar 22 06:45:51 DESKTOP-LQBQG2D node_exporter[18742]: ts=2025-03-22T06:45:51.939Z caller=tls_config.go:2
77 level=info msg="TLS is disabled." http2=false address=[::]:9100
sudo vim /etc/prometheus/prometheus.yml^C
ananth@DESKTOP-LQBQG2D:~$ sudo vim /etc/prometheus/prometheus.yml
ananth@DESKTOP-LQBQG2D:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
  FAILED: parsing YAML file /etc/prometheus/prometheus.yml: yaml: line 29: did not find expected key

ananth@DESKTOP-LQBQG2D:~$ promtool check config /etc/prometheus/prometheus.yaml
promtool: error: path '/etc/prometheus/prometheus.yaml' does not exist, try --help
ananth@DESKTOP-LQBQG2D:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
  FAILED: parsing YAML file /etc/prometheus/prometheus.yml: yaml: line 29: did not find expected key

ananth@DESKTOP-LQBQG2D:~$ sudo vim /etc/prometheus/prometheus.yml
ananth@DESKTOP-LQBQG2D:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
 SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax

ananth@DESKTOP-LQBQG2D:~$ sudo systemctl restart prometheus
sudo systemctl status prometheus
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-03-22 06:57:20 UTC; 40ms ago
   Main PID: 21473 (prometheus)
```

🔍 What is Node Exporter?

Node Exporter is a tool used by **Prometheus** to collect system metrics like:
✅ **CPU usage**
✅ **Memory usage**
✅ **Disk stats**

✅ **Network activity**
✅ **Load average**
✅ **Logged-in users**

These metrics help monitor system health in **Prometheus dashboards**.

📜 Breakdown of `node_exporter.service`

ini
CopyEdit

```ini
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target
StartLimitIntervalSec=500
StartLimitBurst=5
```

🔹 What This Does?

- **`Description=Node Exporter`** → Describes the service.
- **`Wants=network-online.target`** → Ensures the network is online before starting.
- **`After=network-online.target`** → Waits for the network to be ready.
- **`StartLimitIntervalSec=500`** & **`StartLimitBurst=5`** → Prevents rapid restart loops.

---

ini
CopyEdit

```ini
[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
```

🔹 What This Does?

- **`User=node_exporter`** → Runs as the `node_exporter` user (security best practice).
- **`Group=node_exporter`** → Uses the `node_exporter` group.
- **`Type=simple`** → Runs as a simple process.
- **`Restart=on-failure`** → If it crashes, restart automatically.
- **`RestartSec=5s`** → Wait **5 seconds** before restarting.

---

ini

CopyEdit

```ini
ExecStart=/usr/local/bin/node_exporter \
    --web.listen-address=":9100" \
    --collector.cpu \
    --collector.meminfo \
    --collector.diskstats \
    --collector.filesystem \
    --collector.netdev \
    --collector.loadavg \
    --collector.logind
```

🔹 What This Does?

- **`ExecStart`** → Defines how Node Exporter runs.
- **`--web.listen-address=":9100"`** → Exposes metrics on port **9100**.
- **`--collector.cpu`** → Collects **CPU usage**.
- **`--collector.meminfo`** → Collects **memory stats**.
- **`--collector.diskstats`** → Tracks **disk usage**.
- **`--collector.filesystem`** → Monitors **file system usage**.

- `--collector.netdev` → Captures **network activity**.
- `--collector.loadavg` → Records **system load average**.
- `--collector.logind` → Monitors **logged-in users**.

---

ini
CopyEdit

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

🔹 What This Does?

- **Ensures Node Exporter starts on boot** with **multi-user mode** (default for servers).

🔎 How to Use It?

1️⃣ Start Node Exporter

sh
CopyEdit

```
sudo systemctl start node_exporter
```

2️⃣ Enable on Boot

sh
CopyEdit

```
sudo systemctl enable node_exporter
```

3️⃣ Check Status

sh
CopyEdit

```
sudo systemctl status node_exporter
```

Expected output:

arduino
CopyEdit

```
Active: active (running)
```

4️⃣ Verify Metrics

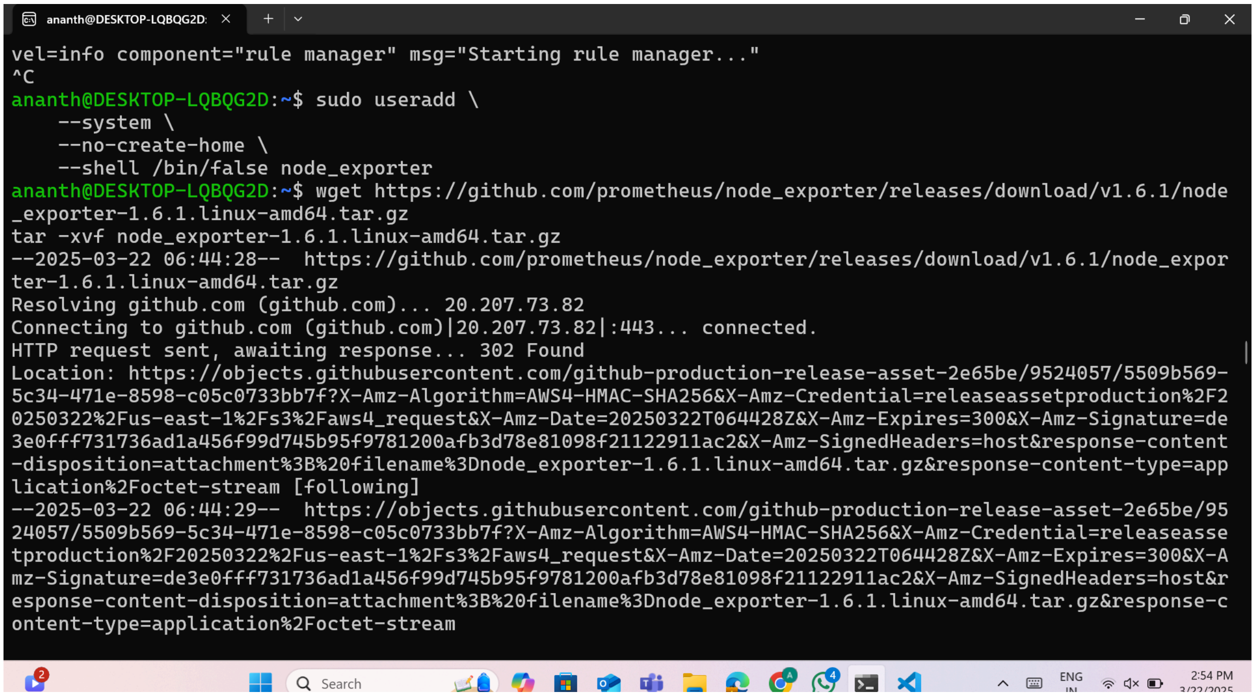
Open a browser and visit:
🔗 **http://localhost:9100/metrics**

If successful, you'll see system metrics!

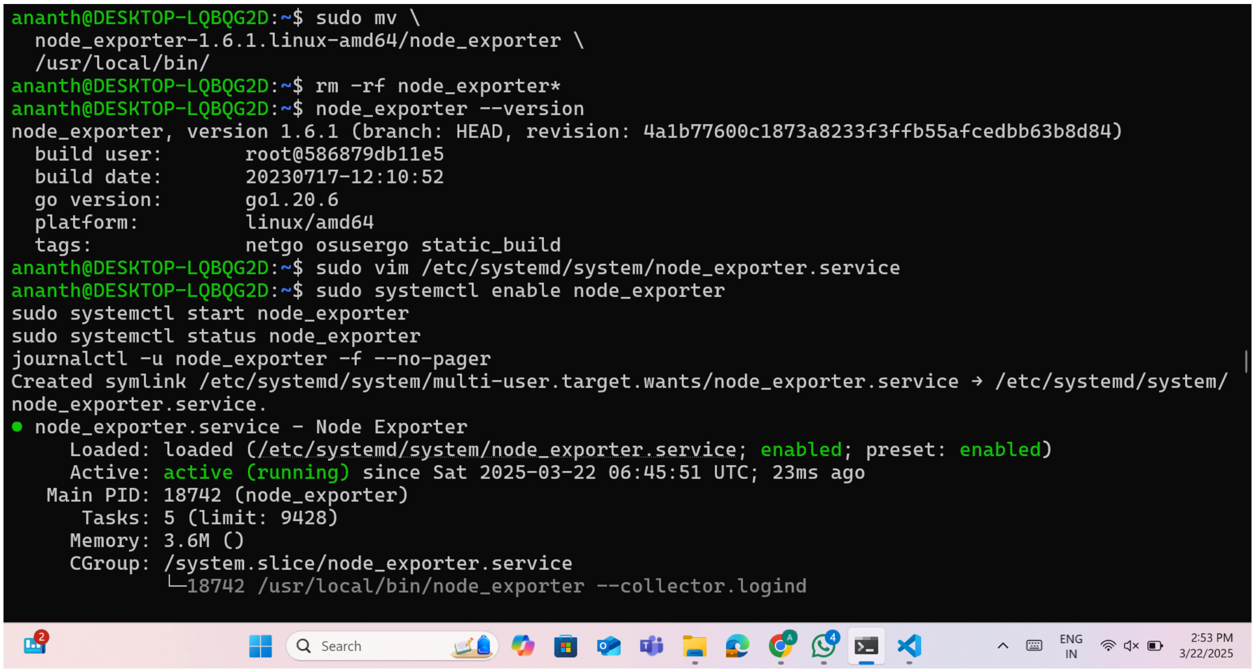
🎯 Summary

Component	What It Does?
Node Exporter	Collects system metrics for Prometheus
Systemd Service	Runs Node Exporter as a background process
Port 9100	Exposes metrics to Prometheus
Collectors	CPU, Memory, Disk, Network, Load, Logged-in users

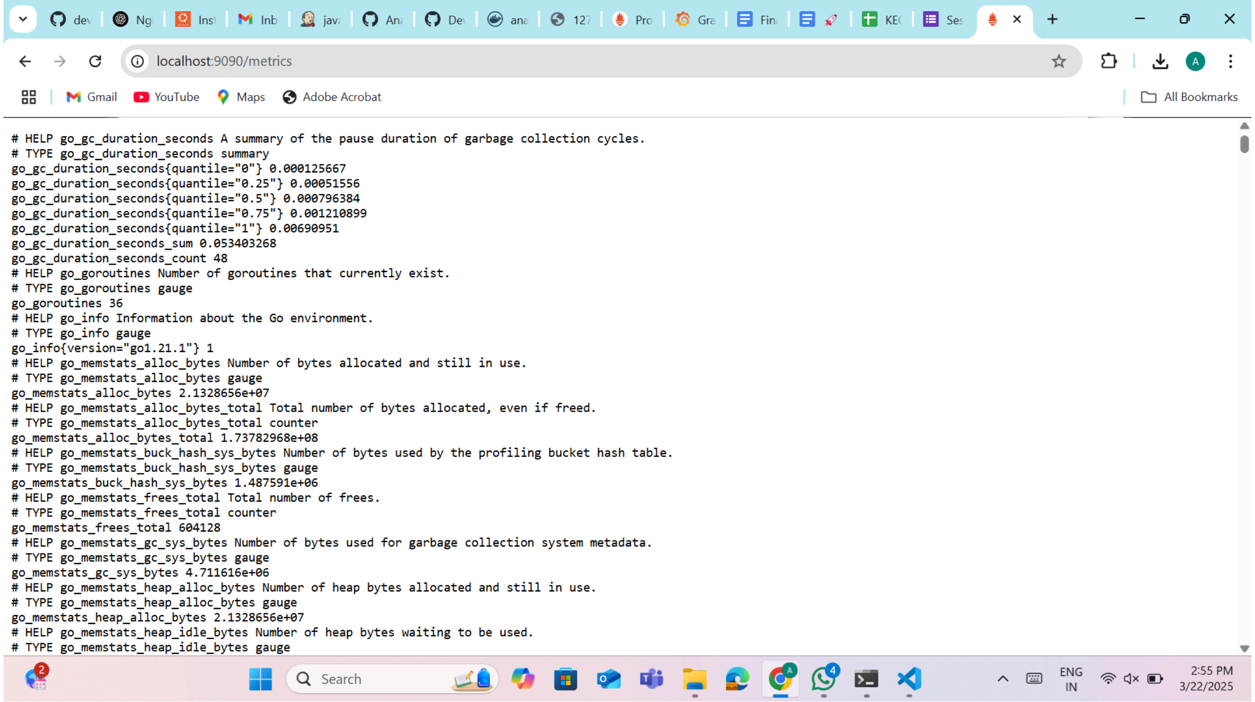
✅ **Now, your system is sending real-time metrics to Prometheus!** 🚀
Let me know if you have any questions! 😊



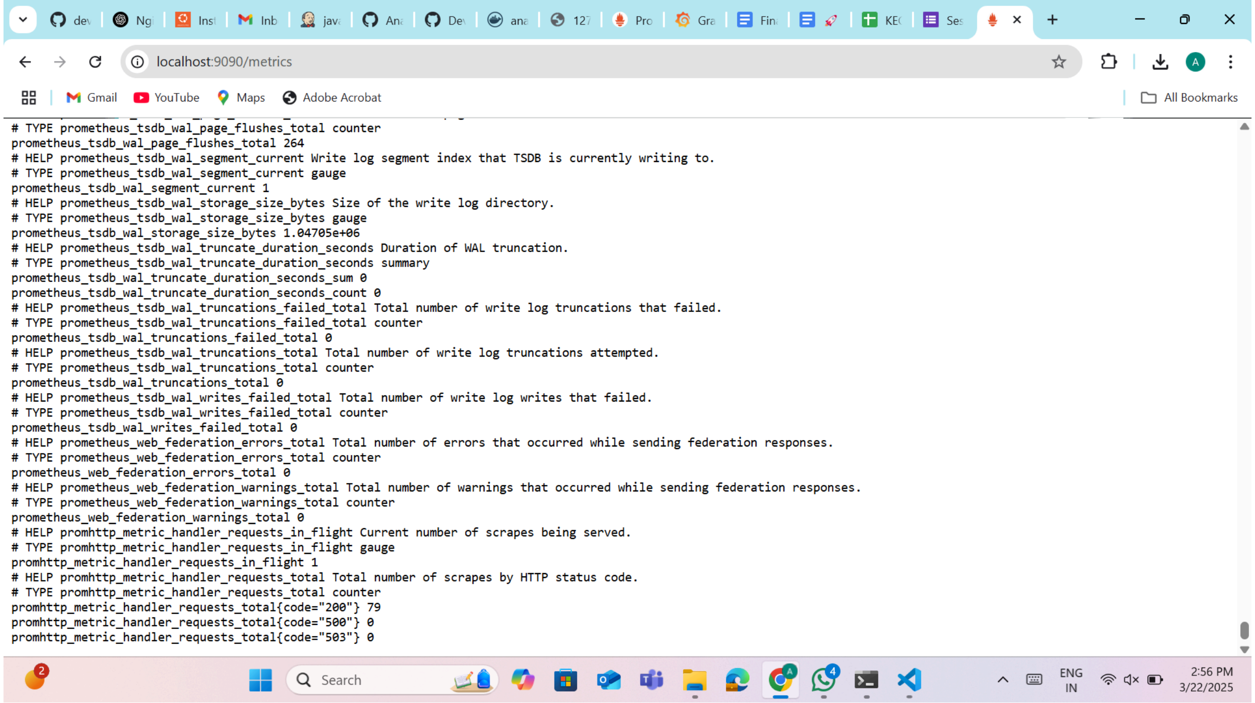
```
vel=info component="rule manager" msg="Starting rule manager..."
^C
ananth@DESKTOP-LQBQG2D:~$ sudo useradd \
    --system \
    --no-create-home \
    --shell /bin/false node_exporter
ananth@DESKTOP-LQBQG2D:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node
_exporter-1.6.1.linux-amd64.tar.gz
tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
--2025-03-22 06:44:28--  https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_expor
ter-1.6.1.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/9524057/5509b569-
5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F2
0250322%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250322T064428Z&X-Amz-Expires=300&X-Amz-Signature=de
3e0fff731736ad1a456f99d745b95f9781200afb3d78e81098f21122911ac2&X-Amz-SignedHeaders=host&response-content
-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-content-type=app
lication%2Foctet-stream [following]
--2025-03-22 06:44:29--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/95
24057/5509b569-5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releasease
tproduction%2F20250322%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250322T064428Z&X-Amz-Expires=300&X-A
mz-Signature=de3e0fff731736ad1a456f99d745b95f9781200afb3d78e81098f21122911ac2&X-Amz-SignedHeaders=host&r
esponse-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-c
ontent-type=application%2Foctet-stream
```



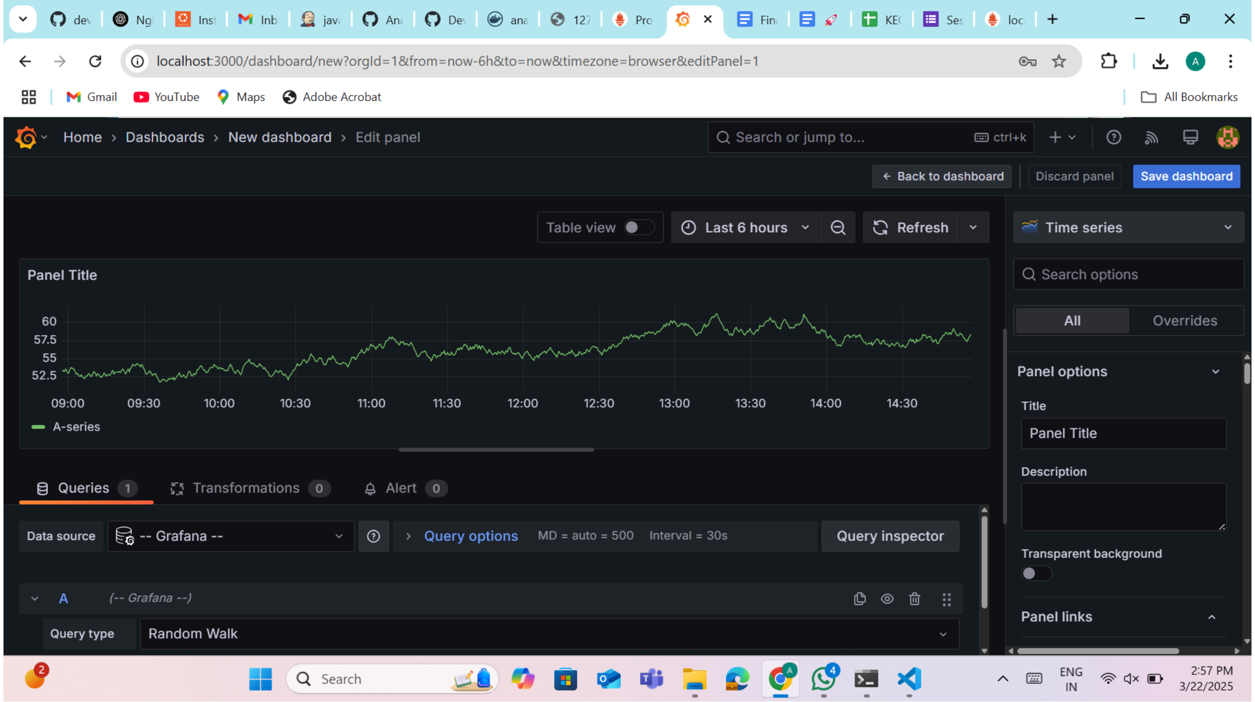
```
ananth@DESKTOP-LQBQG2D:~$ sudo mv \
  node_exporter-1.6.1.linux-amd64/node_exporter \
  /usr/local/bin/
ananth@DESKTOP-LQBQG2D:~$ rm -rf node_exporter*
ananth@DESKTOP-LQBQG2D:~$ node_exporter --version
node_exporter, version 1.6.1 (branch: HEAD, revision: 4a1b77600c1873a8233f3ffb55afcedbb63b8d84)
  build user:       root@586879db11e5
  build date:       20230717-12:10:52
  go version:       go1.20.6
  platform:         linux/amd64
  tags:             netgo osusergo static_build
ananth@DESKTOP-LQBQG2D:~$ sudo vim /etc/systemd/system/node_exporter.service
ananth@DESKTOP-LQBQG2D:~$ sudo systemctl enable node_exporter
sudo systemctl start node_exporter
sudo systemctl status node_exporter
journalctl -u node_exporter -f --no-pager
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/
node_exporter.service.
● node_exporter.service - Node Exporter
     Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-03-22 06:45:51 UTC; 23ms ago
   Main PID: 18742 (node_exporter)
      Tasks: 5 (limit: 9428)
     Memory: 3.6M ()
     CGroup: /system.slice/node_exporter.service
             └─18742 /usr/local/bin/node_exporter --collector.logind
```



```
# 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"} 0.000125667
go_gc_duration_seconds{quantile="0.25"} 0.00051556
go_gc_duration_seconds{quantile="0.5"} 0.000796384
go_gc_duration_seconds{quantile="0.75"} 0.001210899
go_gc_duration_seconds{quantile="1"} 0.00690951
go_gc_duration_seconds_sum 0.053403268
go_gc_duration_seconds_count 48
# 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 2.1328656e+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 1.73782968e+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 1.487591e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 604128
# 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 4.711616e+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 2.1328656e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
```



```
# TYPE prometheus_tsdb_wal_page_flushes_total counter
prometheus_tsdb_wal_page_flushes_total 264
# HELP prometheus_tsdb_wal_segment_current Write log segment index that TSDB is currently writing to.
# TYPE prometheus_tsdb_wal_segment_current gauge
prometheus_tsdb_wal_segment_current 1
# HELP prometheus_tsdb_wal_storage_size_bytes Size of the write log directory.
# TYPE prometheus_tsdb_wal_storage_size_bytes gauge
prometheus_tsdb_wal_storage_size_bytes 1.04705e+06
# HELP prometheus_tsdb_wal_truncate_duration_seconds Duration of WAL truncation.
# TYPE prometheus_tsdb_wal_truncate_duration_seconds summary
prometheus_tsdb_wal_truncate_duration_seconds_sum 0
prometheus_tsdb_wal_truncate_duration_seconds_count 0
# HELP prometheus_tsdb_wal_truncations_failed_total Total number of write log truncations that failed.
# TYPE prometheus_tsdb_wal_truncations_failed_total counter
prometheus_tsdb_wal_truncations_failed_total 0
# HELP prometheus_tsdb_wal_truncations_total Total number of write log truncations attempted.
# TYPE prometheus_tsdb_wal_truncations_total counter
prometheus_tsdb_wal_truncations_total 0
# HELP prometheus_tsdb_wal_writes_failed_total Total number of write log writes that failed.
# TYPE prometheus_tsdb_wal_writes_failed_total counter
prometheus_tsdb_wal_writes_failed_total 0
# HELP prometheus_web_federation_errors_total Total number of errors that occurred while sending federation responses.
# TYPE prometheus_web_federation_errors_total counter
prometheus_web_federation_errors_total 0
# HELP prometheus_web_federation_warnings_total Total number of warnings that occurred while sending federation responses.
# TYPE prometheus_web_federation_warnings_total counter
prometheus_web_federation_warnings_total 0
# HELP promhttp_metric_handler_requests_in_flight Current number of scrapes being served.
# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_in_flight 1
# HELP promhttp_metric_handler_requests_total Total number of scrapes by HTTP status code.
# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="200"} 79
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
```


localhost:3000/dashboard/new?orgId=1&from=now-6h&to=now&timezone=browser&editPanel=1

Home › Dashboards › New dashboard › Edit panel

Last 6 hours

Time series

Panel Title

Data source: -- Grafana --

Query options MD = auto = 500 Interval = 30s

A (-- Grafana --)

Query type: Random Walk

Panel options

Title: Panel Title

Description:

Transparent background

Panel links