

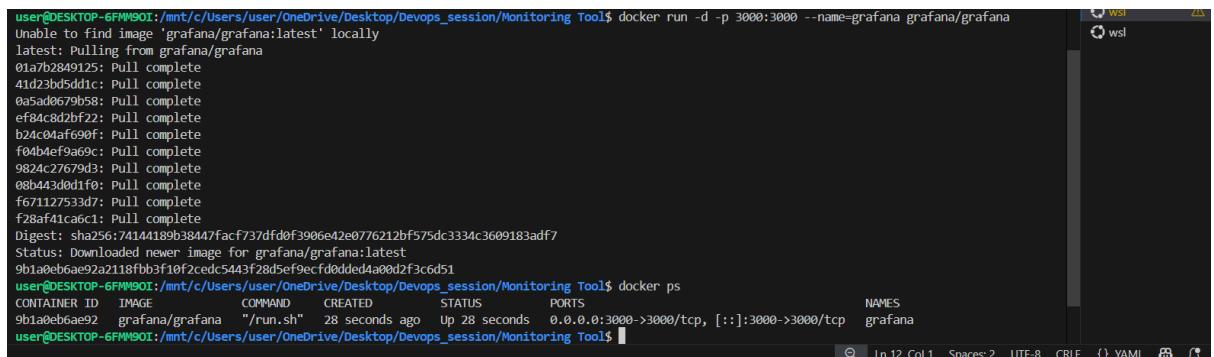
11-10-2025 (Day -4)

Day-4 :- Grafana & Prometheus

Note :- All Docker commands you are using — like running Prometheus, Grafana, and Node Exporter — require Docker Desktop s/w to be running in the background.

- Starts a **Grafana container** in detached mode (-d) and exposes port **3000**.

```
docker run -d -p 3000:3000 --name=grafana grafana/Grafana
```



```
user@DESKTOP-6FMM90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker run -d -p 3000:3000 --name=grafana grafana/grafana
Unable to find image 'grafana/grafana:latest' locally
latest: Pulling from grafana/grafana
01a7b2849125: Pull complete
41d23bd5d1c1: Pull complete
0a5ad679508: Pull complete
ef84c8d2bf22: Pull complete
b24c0af690f: Pull complete
f04b4ef9a69c: Pull complete
9824c27679d3: Pull complete
08b443d6df0: Pull complete
f671127533d7: Pull complete
f28af41caec1: Pull complete
Digest: sha256:74144189b38447facf737dfdf3906e42e0776212bf575dc3334c3609183adf7
Status: Downloaded newer image for grafana/grafana:latest
9b1a0eb6ae92a2118fb3f10f2cedc5443f28d5ef9ecfd0ded4a0d2f3cd51
user@DESKTOP-6FMM90I:/mnt/c/Users/user/oneDrive/Desktop/Devops_session/Monitoring Tool$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9b1a0eb6ae92 grafana/grafana "/run.sh" 28 seconds ago Up 28 seconds 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp grafana
user@DESKTOP-6FMM90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$
```

- In terminal, your Grafana is exposed at **port 3000**.

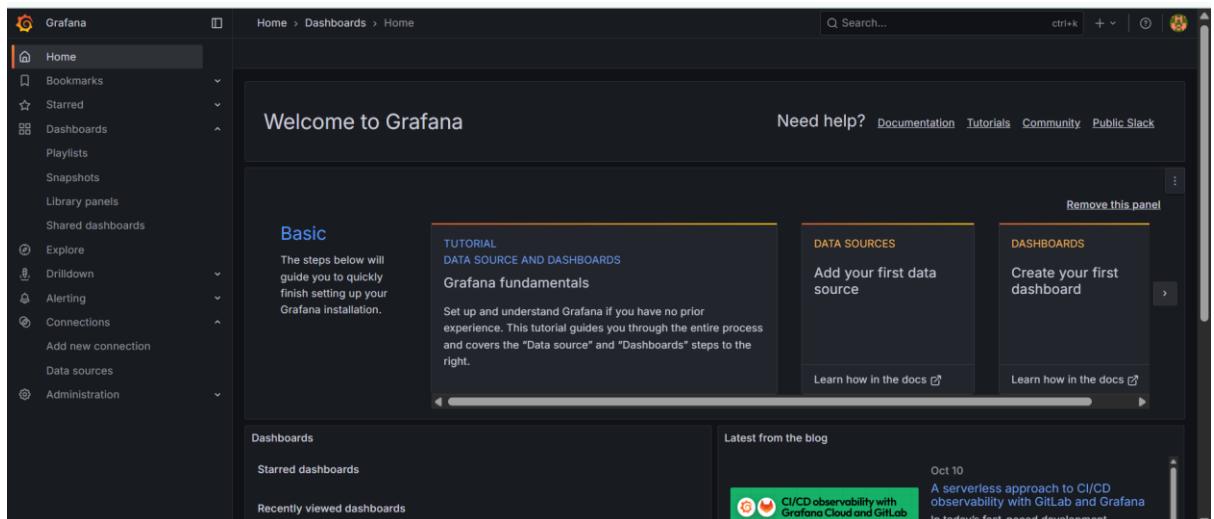
Open Chrome and go to:

<http://localhost:3000>

Login to Grafana

- Default Username:** admin
- Default Password:** admin

On first login, Grafana will prompt you to **change the password** for security. Set a new password of your choice.  then display



- Starts a **Prometheus container** in detached mode and exposes port **9090**.

```
docker run -d -p 9090:9090 --name=prometheus prom/Prometheus
```

```
user@DESKTOP-6FMM90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker run -d -p 9090:9090 --name=prometheus prom/prometheus
Unable to find image 'prom/prometheus:latest' locally
latest: Pulling from prom/prometheus
d1ad9311398e: Pull complete
4314b14247f8: Pull complete
e9fa37e588a8: Pull complete
15e2cd5823b3: Pull complete
7dc70dd519ad: Pull complete
92fd369f57f0: Pull complete
03def9af9150: Pull complete
0357ac67262f: Pull complete
Digest: sha256:76947e7ef22f8a698fc638f706685909be425dbe09bd7a2cd7aca849f79b5f64
Status: Downloaded newer image for prom/prometheus:latest
1afeda9b9a7118734112886b94028645245aedb2cc3b92776f92fc6b1ccc1f5e
user@DESKTOP-6FMM90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1afeda9b9a71 prom/prometheus "/bin/prometheus --c..." 55 seconds ago Up 54 seconds 0.0.0.0:9090->9090/tcp, [::]:9090->9090/tcp prometheus
9b1ae96ae92 grafana/grafana "/run.sh" 6 minutes ago Up 6 minutes 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp grafana
user@DESKTOP-6FMM90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$
```

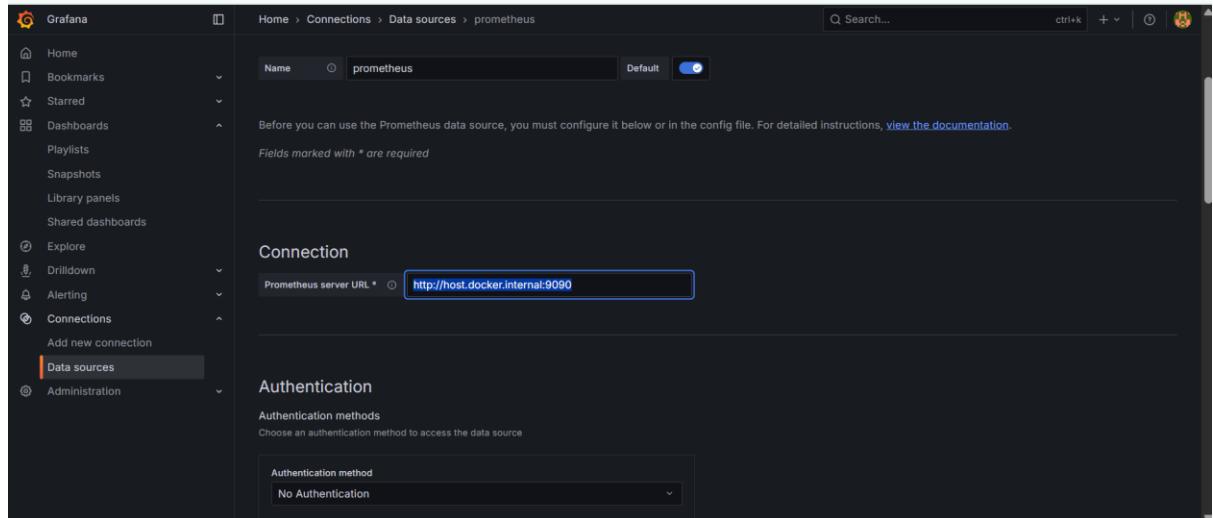
- In terminal, your prometheus is exposed at **port 9090**.

Open Chrome and go to:

<http://localhost:9090>

- Grafana reach Prometheus across container network. We are connecting **Grafana → Prometheus** so Grafana can **read and visualize metrics** that Prometheus collects (for example, CPU, memory, Docker stats, etc).

- 1) In Grafana Dashboard - On the **left sidebar**, click the **Connection**
- 2) then Click “**Data sources**” under Connections.
- 3) Click the “Add data source” button.
- 4) Scroll and select “Prometheus” from the list.
- 5) Paste the <http://host.docker.internal:9090> ip in Prometheus server URL

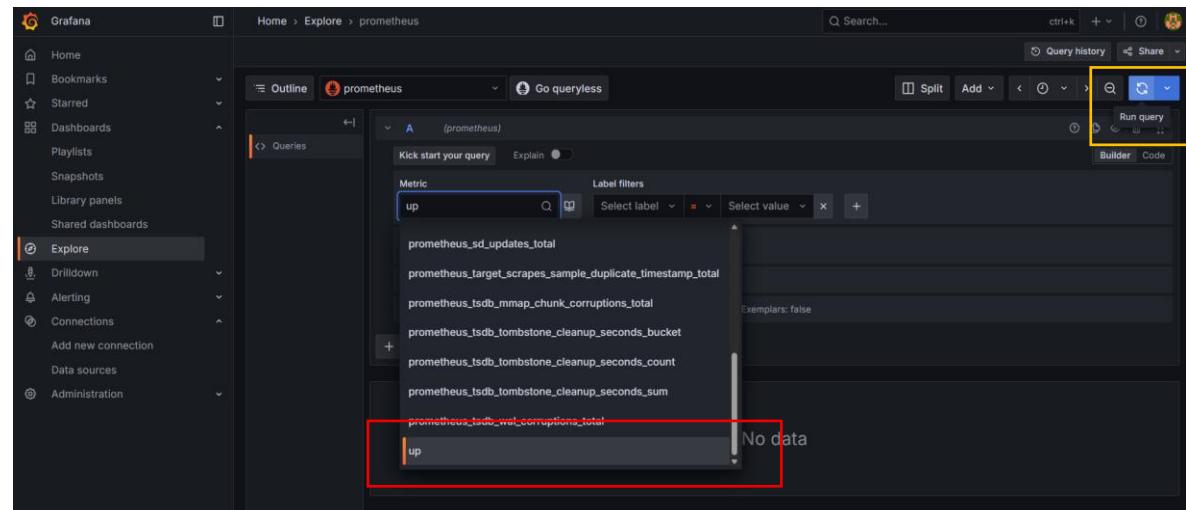


- 6) Click the “Save & Test” button at the bottom.
- 7) Grafana successfully connected to Prometheus
- 8) Verify :- On the left sidebar → Explore . Choose your Prometheus data source from the top-left dropdown.

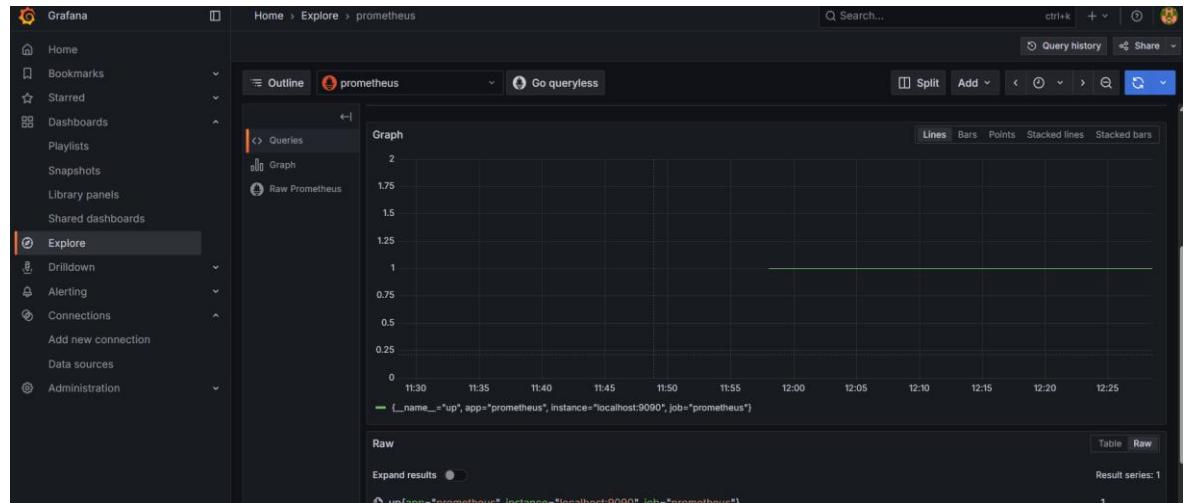
Try a simple query, e.g.:

up

Then click Run query. (we can run many different PromQL queries)



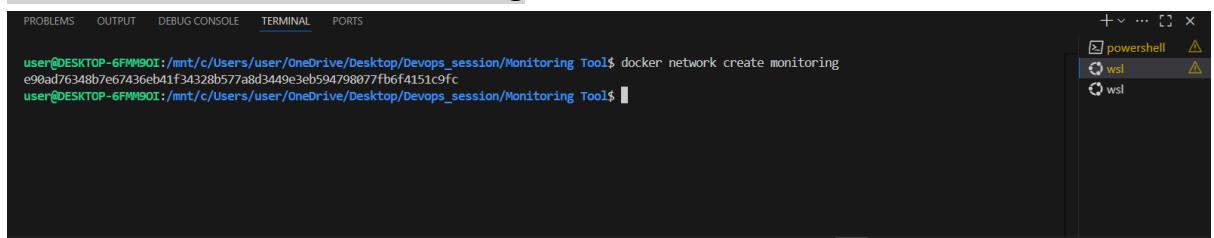
- 9) When U run query, display following :-



Prometheus + Node Exporter + Grafana monitoring stack using Docker.

- Creates a custom Docker network called monitor-net. This allows Grafana, Prometheus, and Node Exporter to communicate **inside the same network**.

docker network create monitoring



- Create **prometheus.yml** and Save this file in a folder (example: Monitoring tool/prometheus.yml)

```
global:
  scrape_interval: 5s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "node-exporter"
    static_configs:
      - targets: ["node-exporter:9100"]
```

8. **Run Node Exporter Container** :- node-exporter exposes hardware and OS metrics (CPU, memory, disk, network, etc.) to Prometheus.

```
docker run -d \
  --name node-exporter \
  --network monitoring \
  -p 9100:9100 \
  prom/node-exporter
```

The screenshot shows a terminal window with the following content:

```
Monitoring Tool > ! prometheus.yml
3
4   scrape_configs:
5     - job_name: "prometheus"
6       static_configs:
7         - targets: ["localhost:9090"]
8
9     - job_name: "node-exporter"
10    static_configs:
11      - targets: ["node-exporter:9100"]
12

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1afeda9b9a71 prom/prometheus "/bin/prometheus --c..." 7 minutes ago Up 7 minutes 0.0.0.0:9090->9090/tcp, [::]:9090->9090/tcp prometheus
9b1a0e0bae92 grafana/grafana "/run.sh" 12 minutes ago Up 12 minutes 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp grafana
user@DESKTOP-6FMM9OI:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker run -d \
--name node-exporter \
--network monitoring \
-p 9100:9100 \
  prom/node-exporter
784c6abf91b5f9e162be6cd28b791aftb3b14272bcbb8dd24c0645b27722d5a60
user@DESKTOP-6FMM9OI:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$
```

The terminal shows the configuration file content, the running containers (Prometheus and Grafana), and the command to run the node-exporter container.

9. **Run Prometheus Container** : Run this command in the same folder where your prometheus.yml exists.

```
docker rm -f prometheus
docker run -d \
  -p 9090:9090 \
  --name prometheus \
  --network monitoring \
  -v "$(pwd)/prometheus.yml:/etc/prometheus/prometheus.yml" \
  prom/prometheus
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
user@DESKTOP-6FMW90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker run -d --name node-exporter --network monitoring -p 9100:9100 prom/node-exporter
e3a4b0996a839258bb9da68bdd983122ff0a5f75935441bf8f81284befff46a8
user@DESKTOP-6FMW90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker rm -f prometheus
docker run -d \
-p 9090:9090 \
-p 9090:9090 \
-p 9090:9090 \
--name prometheus \
--network monitoring \
-v "${pwd}/prometheus.yml:/etc/prometheus/prometheus.yml" \
prom/prometheus
4187a88de4034429179713ce92fbc6cf516eda45f8e17eab6aeaf23f3160de53
user@DESKTOP-6FMW90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4187a88de403 prom/prometheus "/bin/prometheus --c..." 5 seconds ago Up 4 seconds 0.0.0.0:9090->9090/tcp, [::]:9090->9090/tcp prome
eus
e3a4b0996a83 prom/node-exporter "bin/node_exporter" About a minute ago Up About a minute 0.0.0.0:9100->9100/tcp, [::]:9100->9100/tcp node-
exporter
9b1a0eb6ae92 grafana/grafana "/run.sh" About an hour ago Up About an hour 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp grafa
na
user@DESKTOP-6FMW90I:/mnt/c/Users/user/OneDrive/Desktop/Devops_session/Monitoring Tool$ 

```

10. Check *Targets* tab → both Prometheus and Node Exporter should show **UP** <http://localhost:9090/targets>

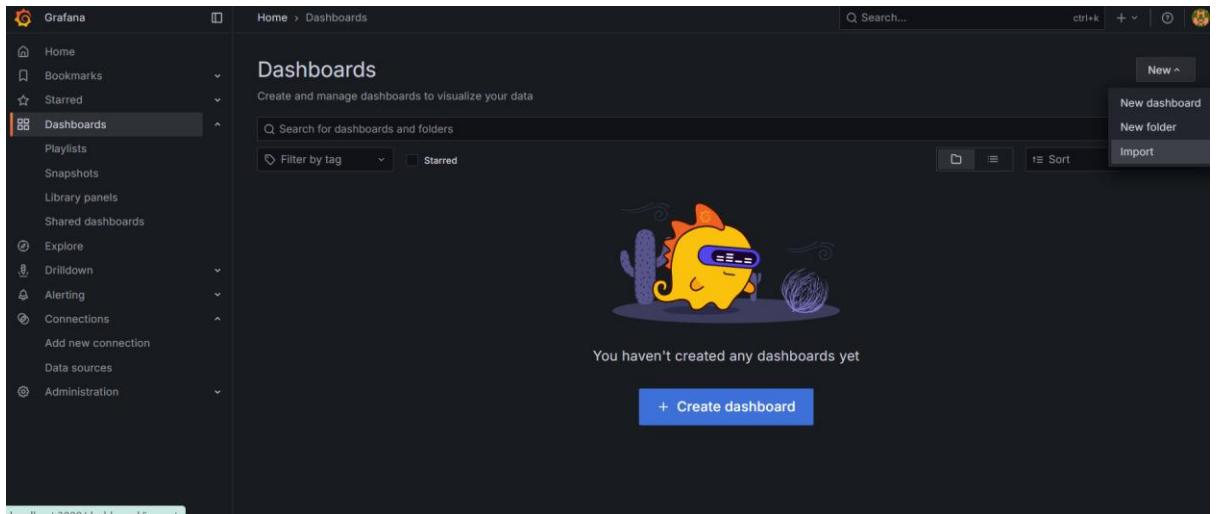
node-exporter		Last scrape		State	
Endpoint	Labels	3.52s ago	28ms	1 / 1 up	UP
http://node-exporter:9100/metrics	instance="node-exporter:9100" job="node-exporter"				

prometheus		Last scrape		State	
Endpoint	Labels	2.613s ago	8ms	1 / 1 up	UP
http://localhost:9090/metrics	instance="localhost:9090" job="prometheus"				

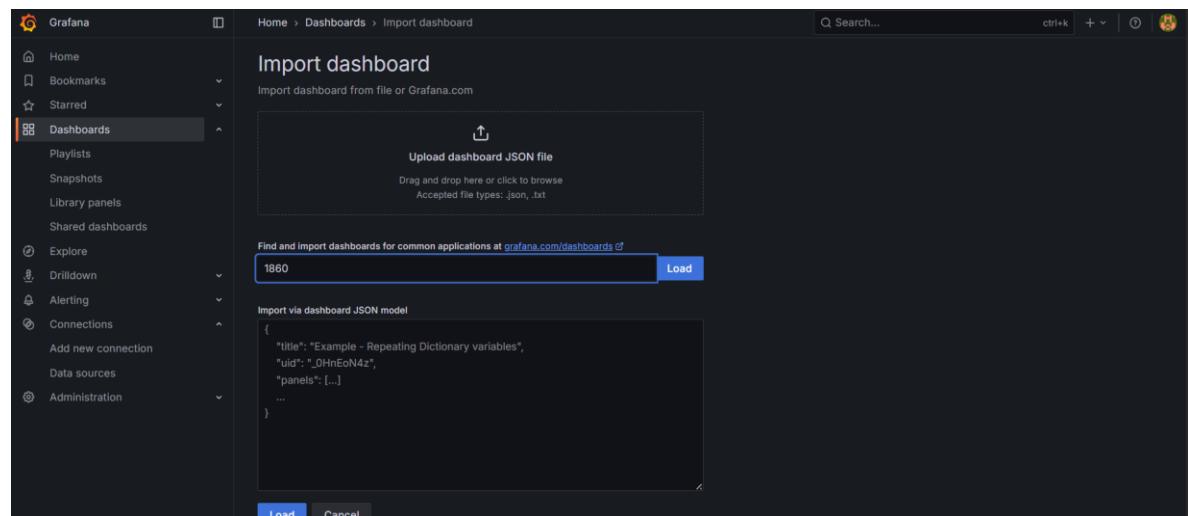
11. Add Prometheus as Grafana Data Source:

Open Grafana → <http://localhost:3000>

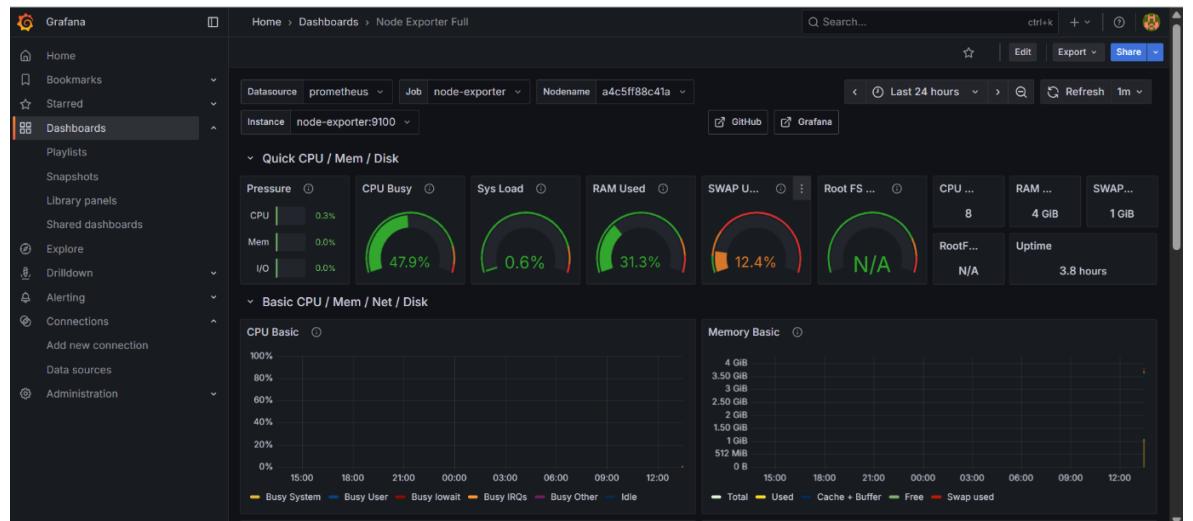
- 1) In Grafana → Left Menu → Dashboards → Import



- 2) Enter dashboard ID: 1860
- 3) Click Load
- 4) Select Prometheus as the data source
- 5) Click Import



- 6) After import Display Dashboard Like This:-



12. This imports a prebuilt dashboard showing CPU, memory, disk, and network metrics collected from Node Exporter.