Prometheus & Grafana

**Monitor System Metrics with Prometheus + Node Exporter + Grafana :**

**🔹 Step 1: Create a Virtual Machine (VM)**

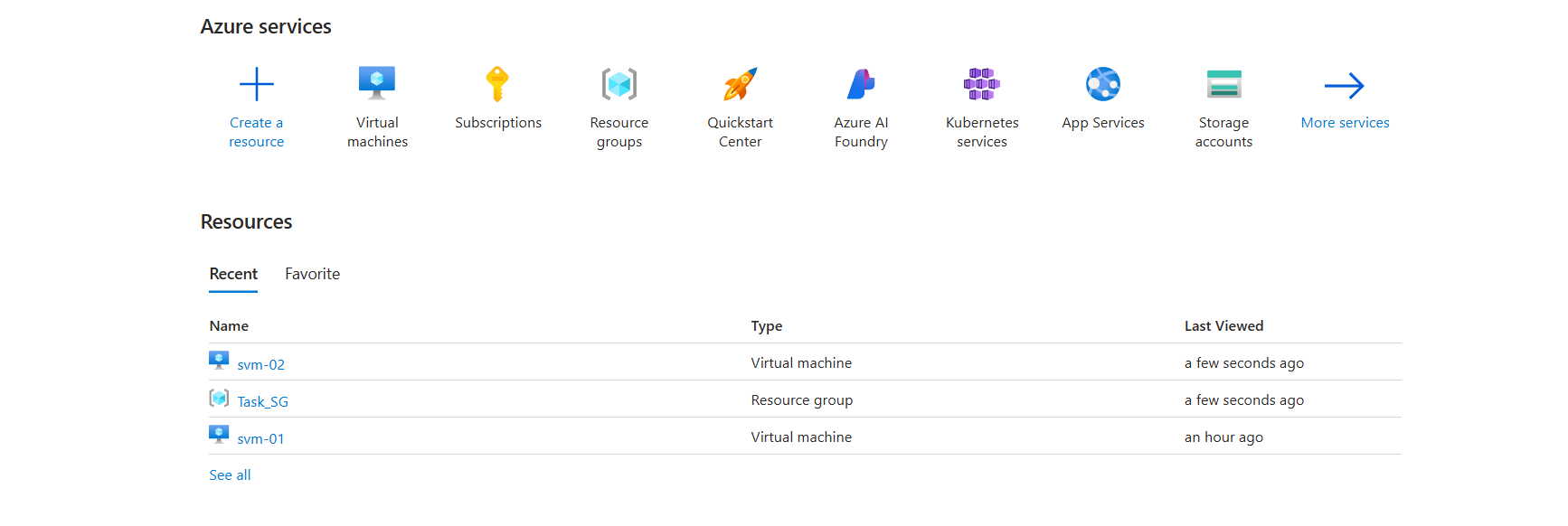
1. **Login** to Azure Portal.
2. **Go to** → **Virtual Machines** → **+ Create VM**
3. **Basics:**
   * Resource group: Select or create new
   * VM name: monitoring-vm
   * Image: Ubuntu 20.04 LTS
   * Size: B1s or above
   * Authentication: Use password or SSH key
4. **Disks**: Default (Standard SSD)
5. **Networking**:
   * Allow **SSH (port 22)**, **HTTP (80)**, and **Custom ports** (9090, 9100, 3000)
6. **Create** → Wait till it's ready.
7. Copy the **public IP** of the VM for SSH.

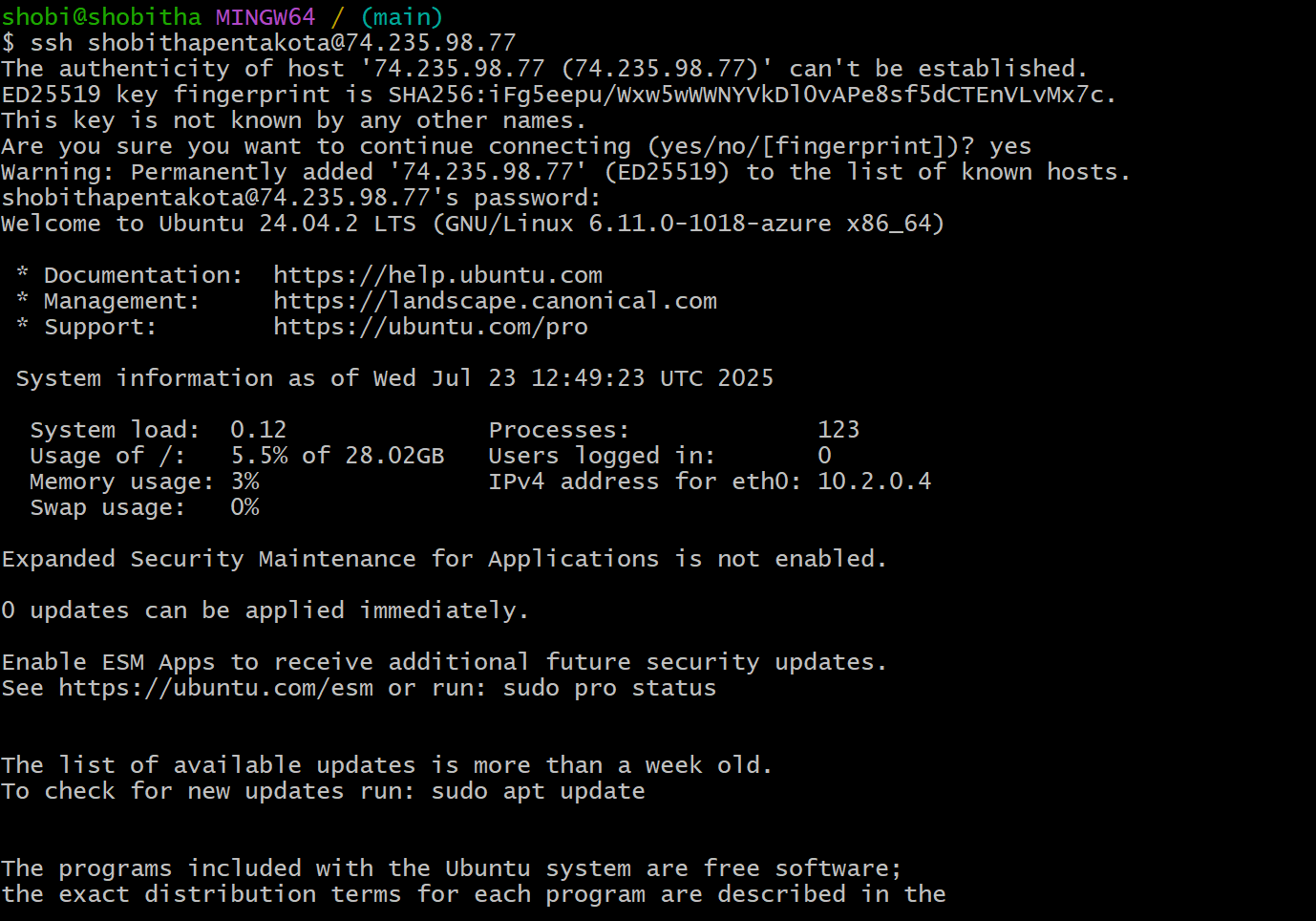
**🔹 Step 2: SSH into the VM**





* Svm-01

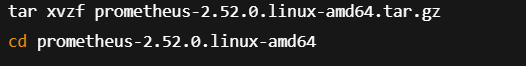


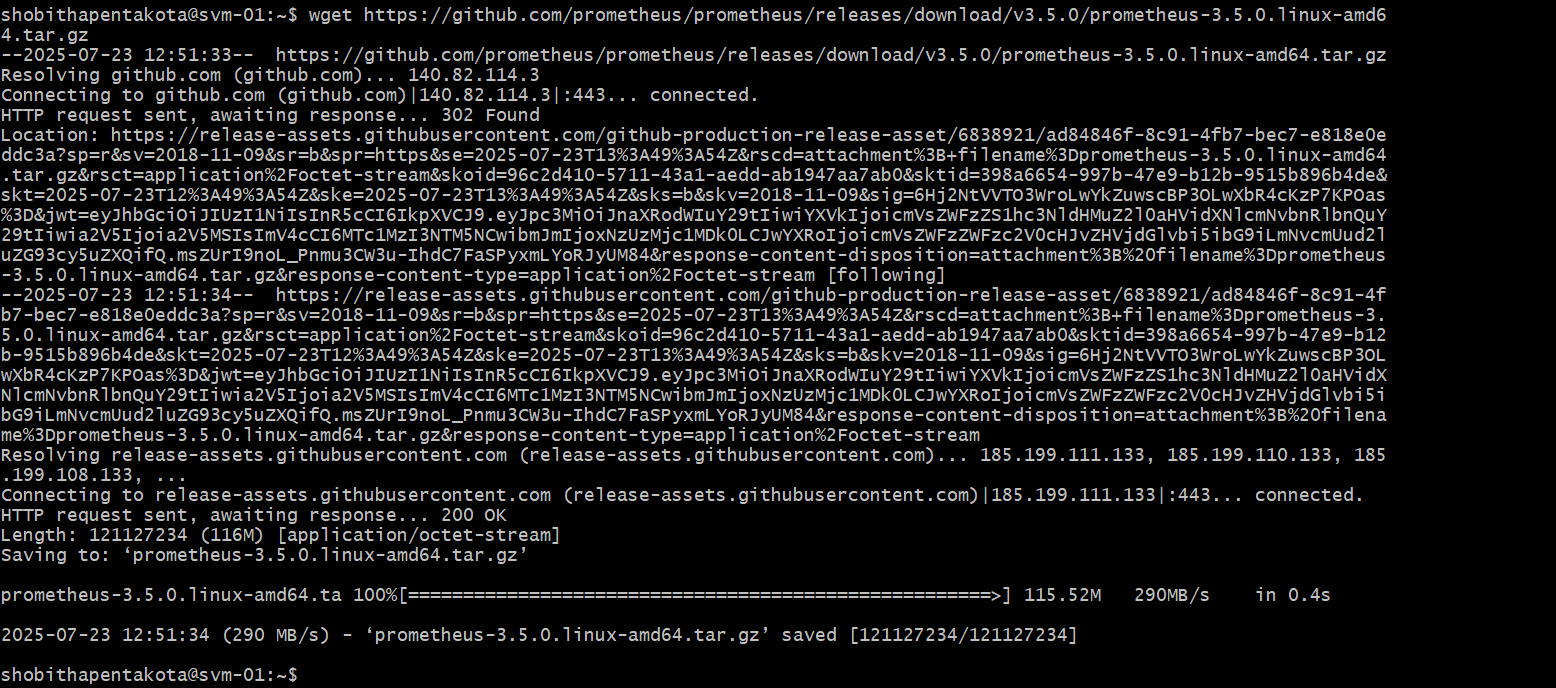


**🔹 Step 3: Install Prometheus**

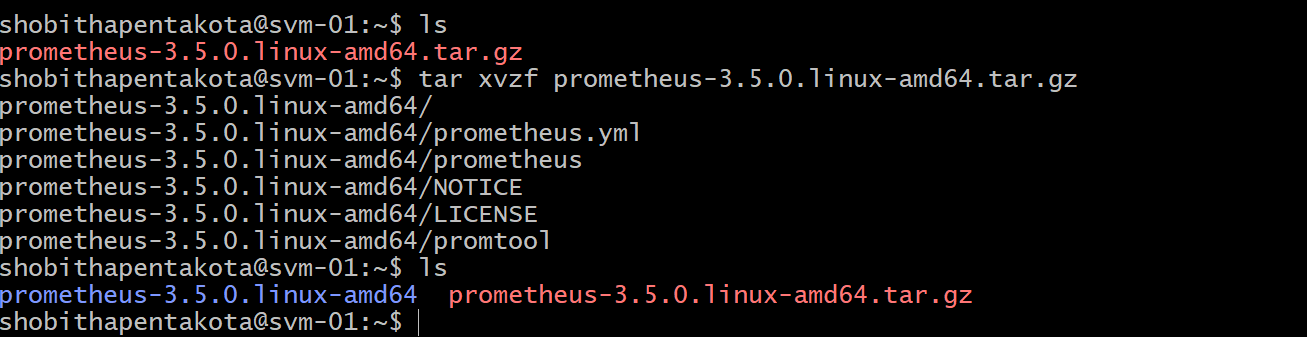
1. Go to Prometheus downloads
2. On the VM terminal:

wget <https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz>

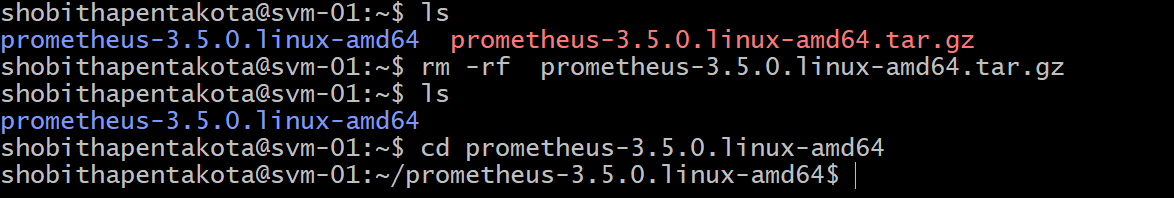


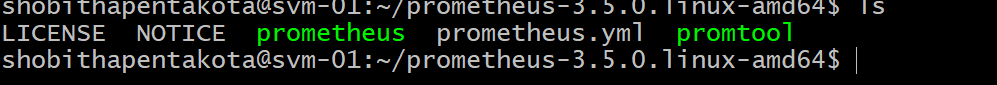


tar xvzf prometheus-2.52.0.linux-amd64.tar.gz



cd prometheus-2.52.0.linux-amd64



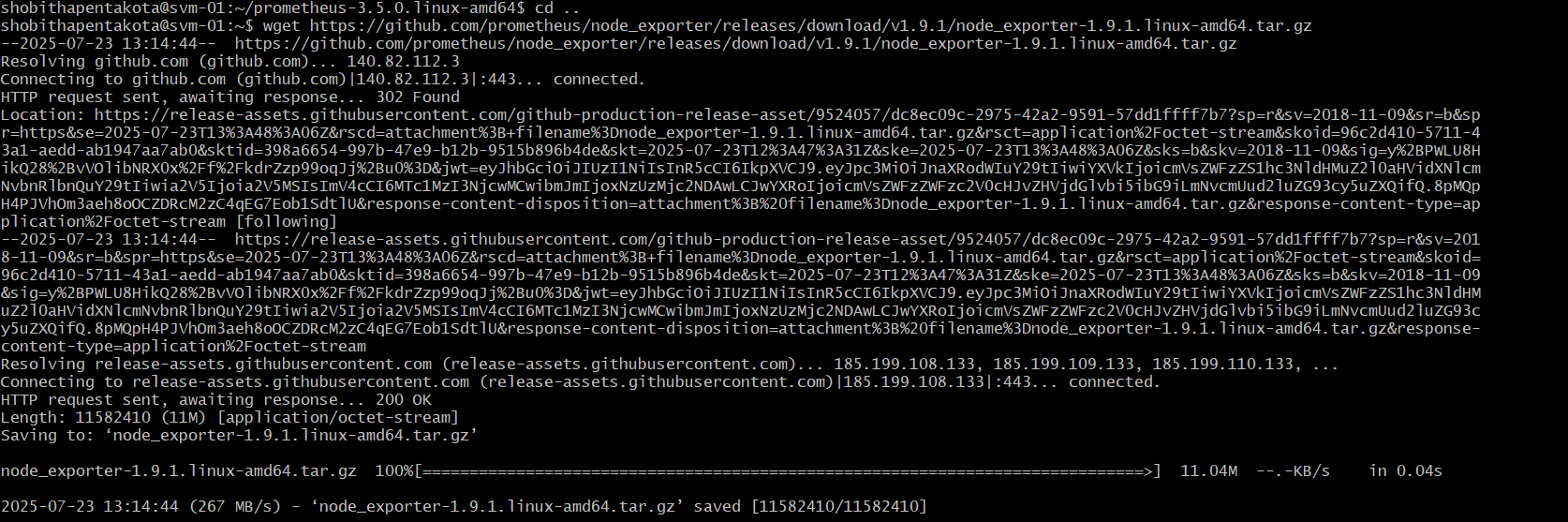


**🔹 Step 4: Install Node Exporter**

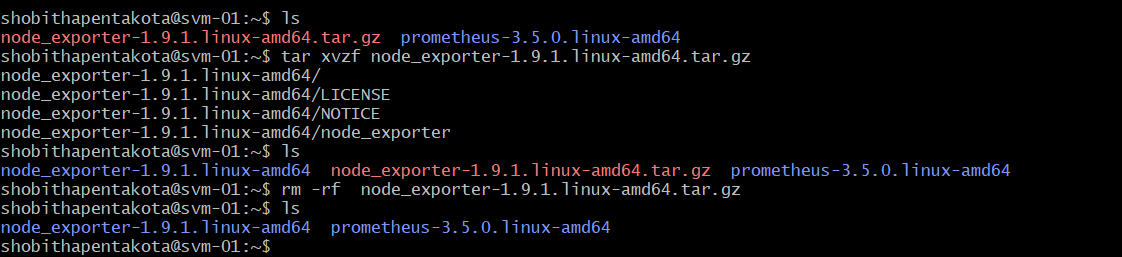
1. Go to Node Exporter download



wget <https://github.com/prometheus/node_exporter/releases/download/v1.8.0/node_exporter-1.8.0.linux-amd64.tar.gz>



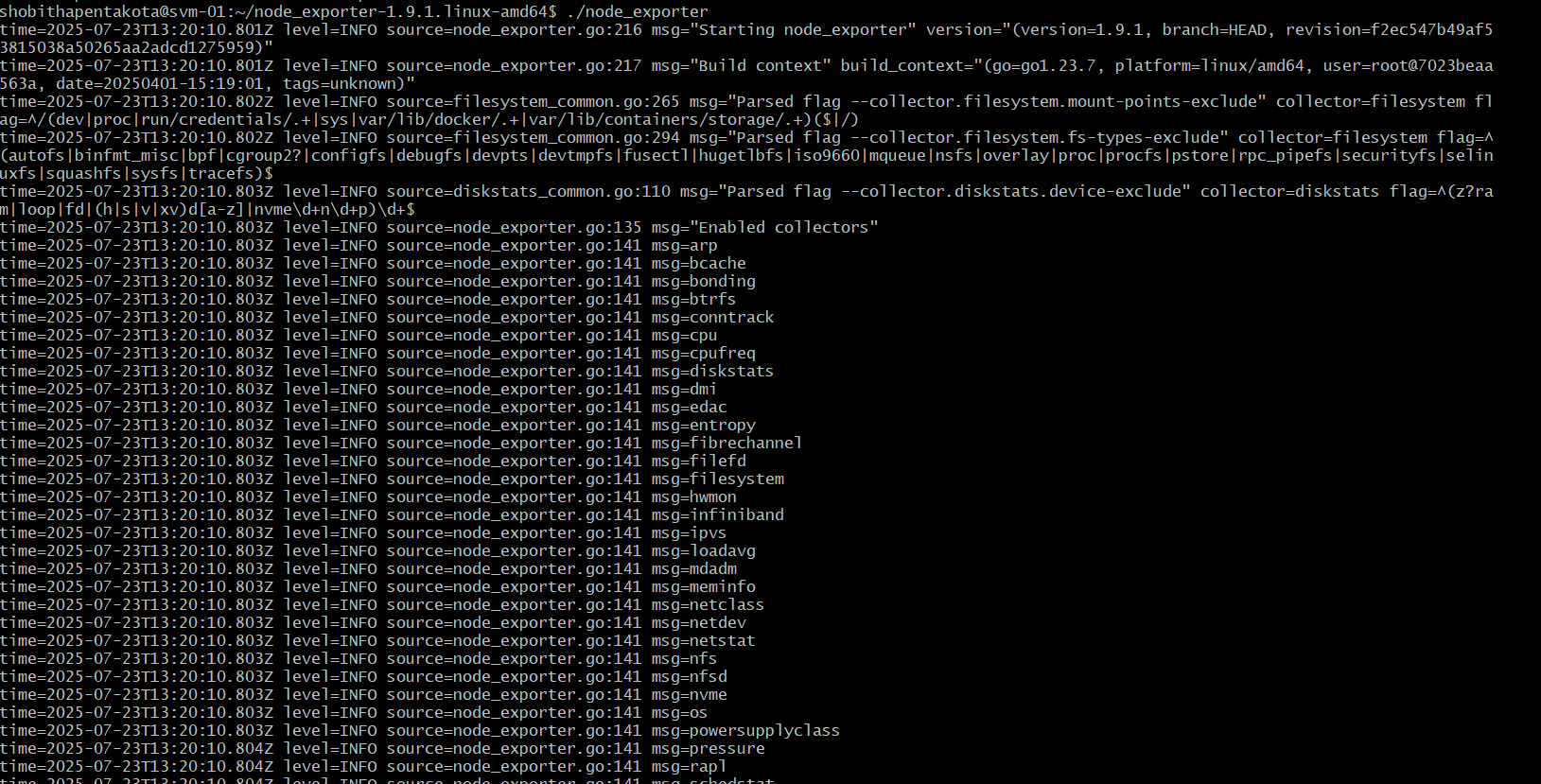
tar xvzf node\_exporter-1.8.0.linux-amd64.tar.gz

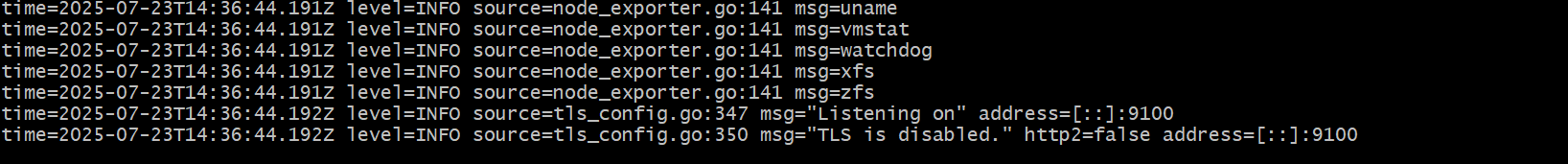


cd node\_exporter-1.8.0.linux-amd64



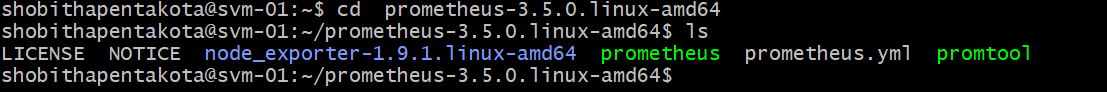
Run node exporter: ./node\_exporter



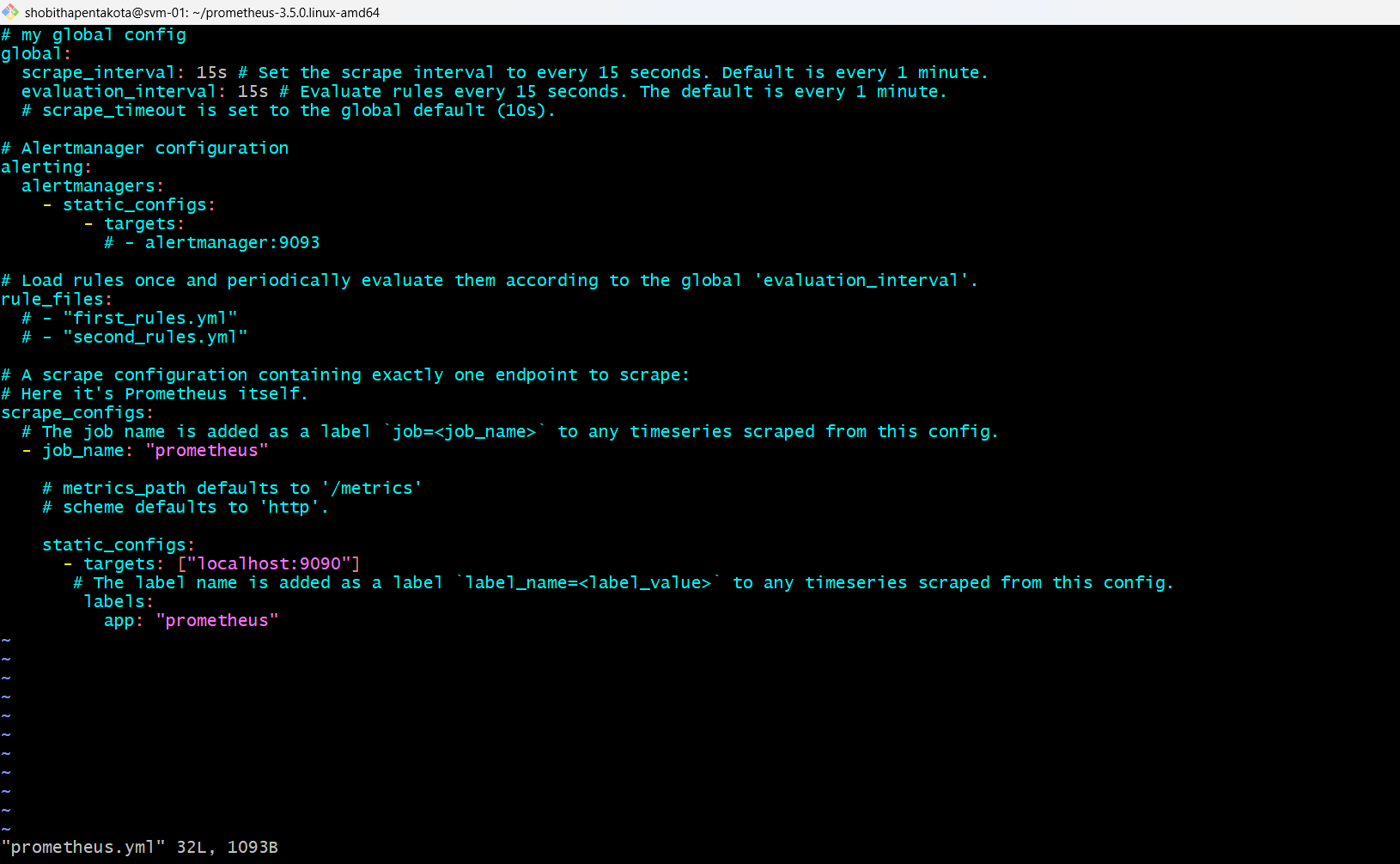


**🔹 Step 5: Configure Prometheus to Scrape Node Exporter**

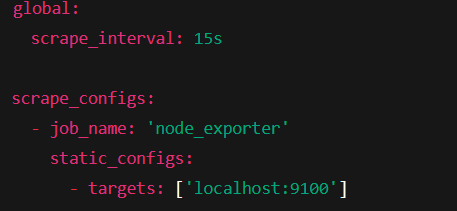
1. Go back to the Prometheus directory:

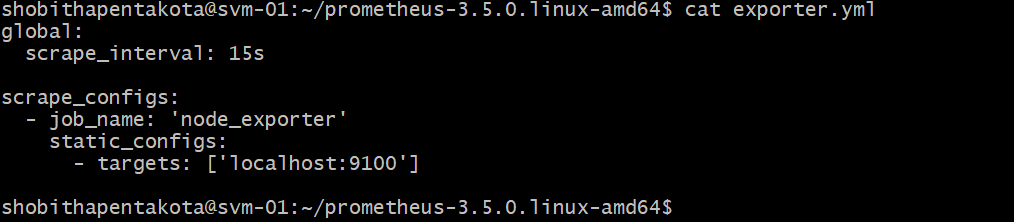


Promethus.yml

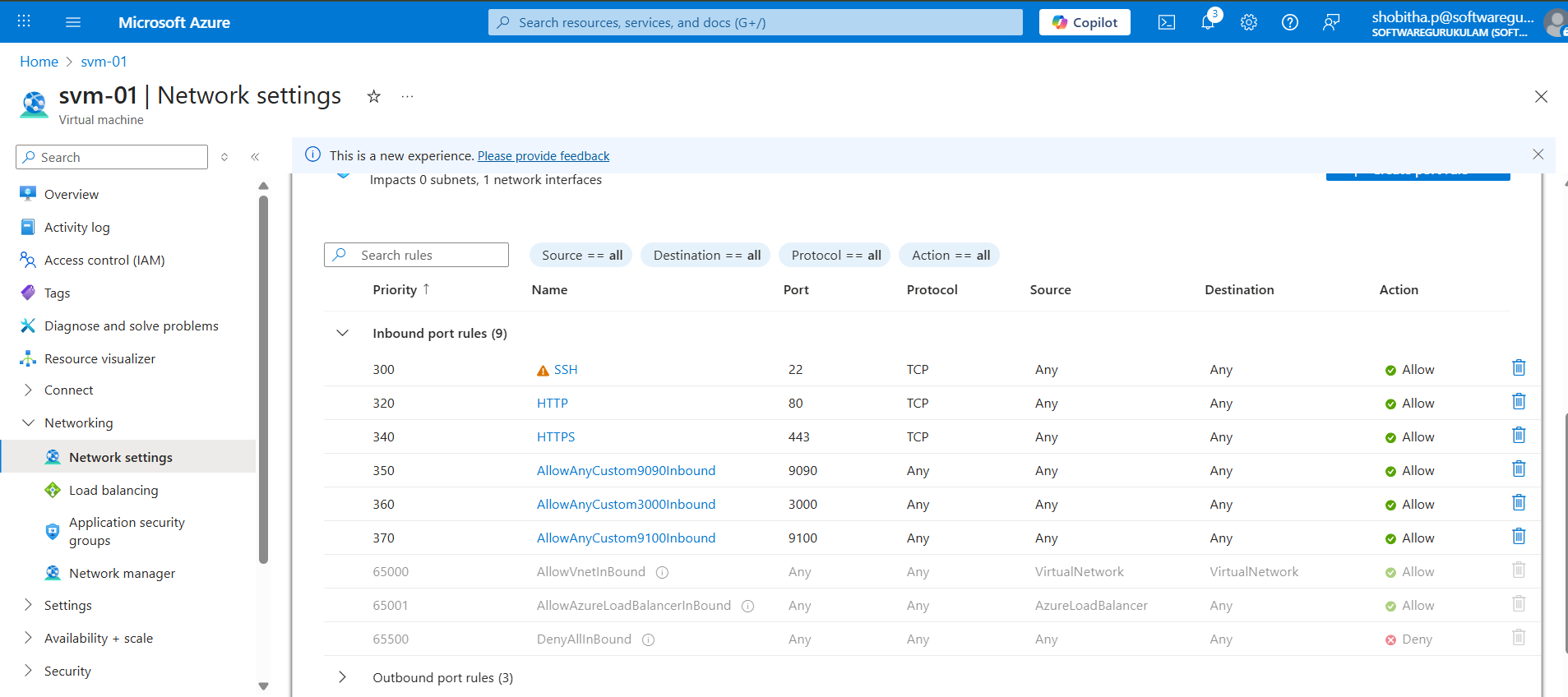


Update prometheus.yml like this: To exporter.yml



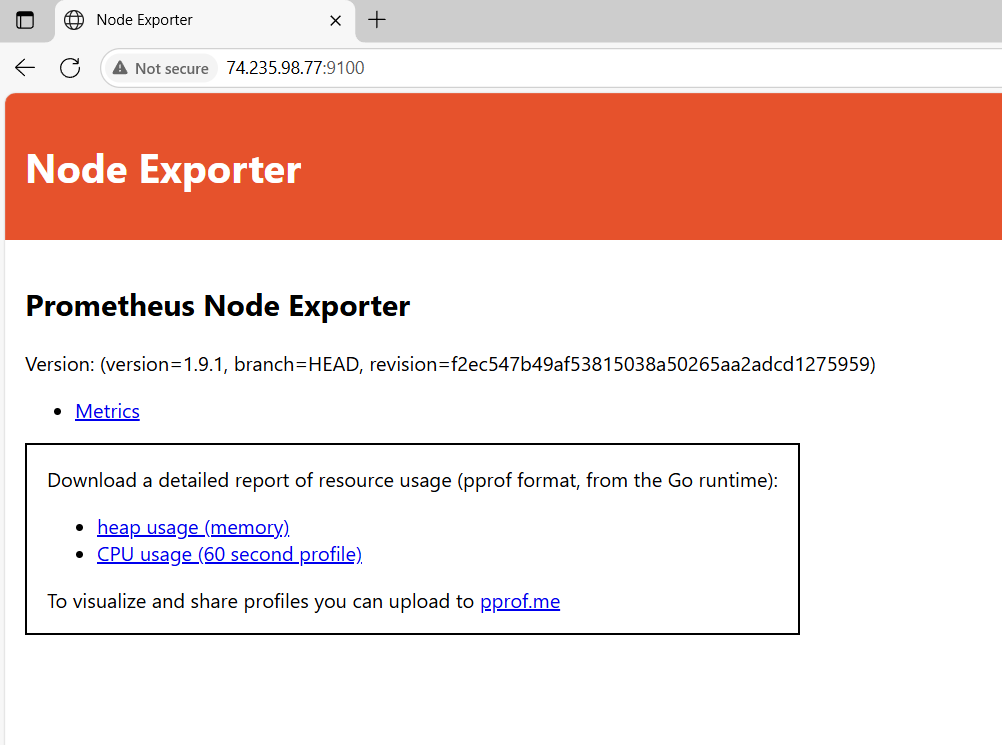


Create port rules:



**Go to node\_exporter directory**: run ./node\_exporter

Then open browser: ip address :9100

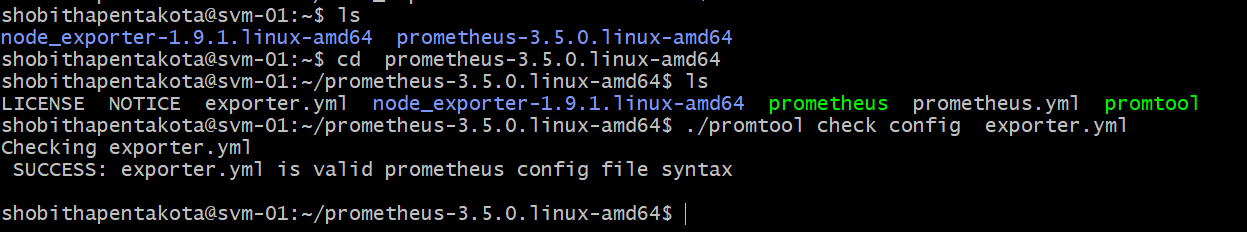




🔹 Step 6: Verify Prometheus Config:

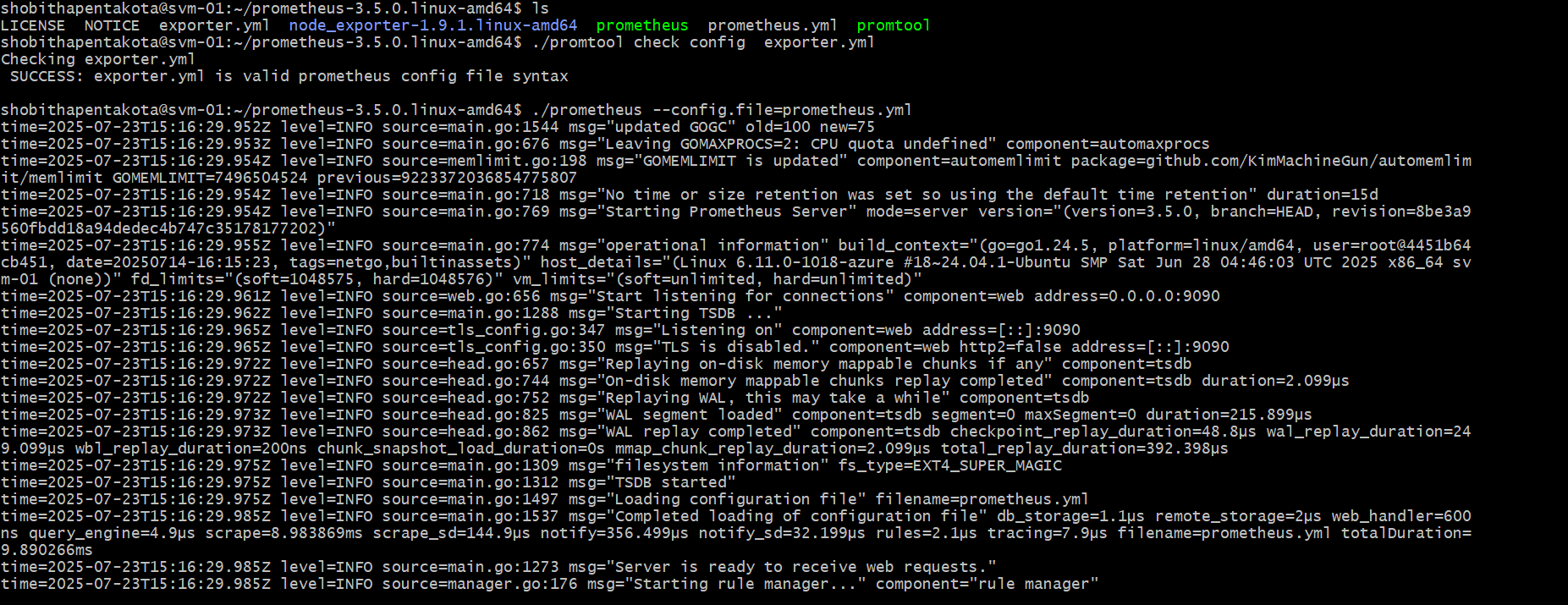
./promtool check config prometheus.yml

Go to Prometheus directory:



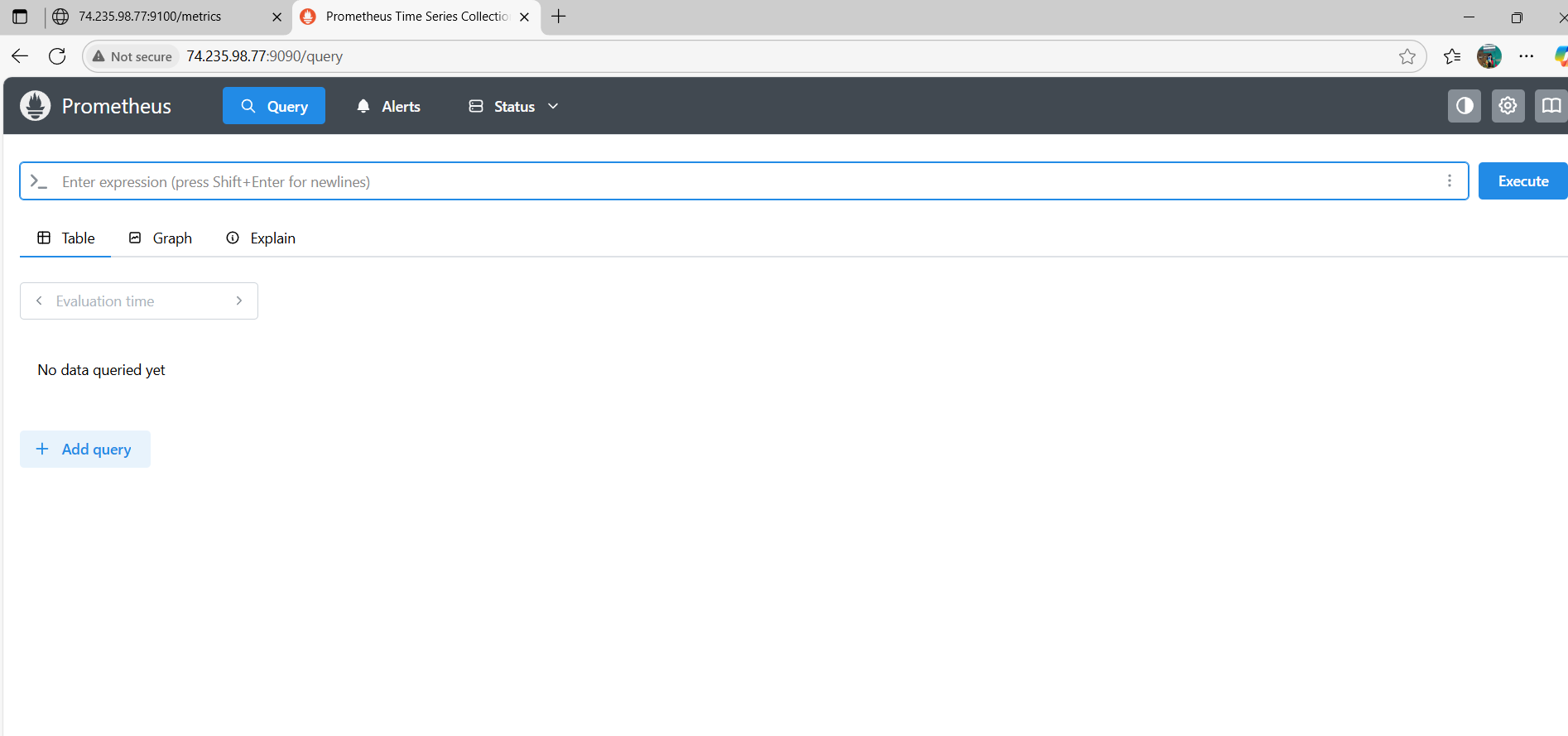
🔹 Step 7: Run Prometheus:

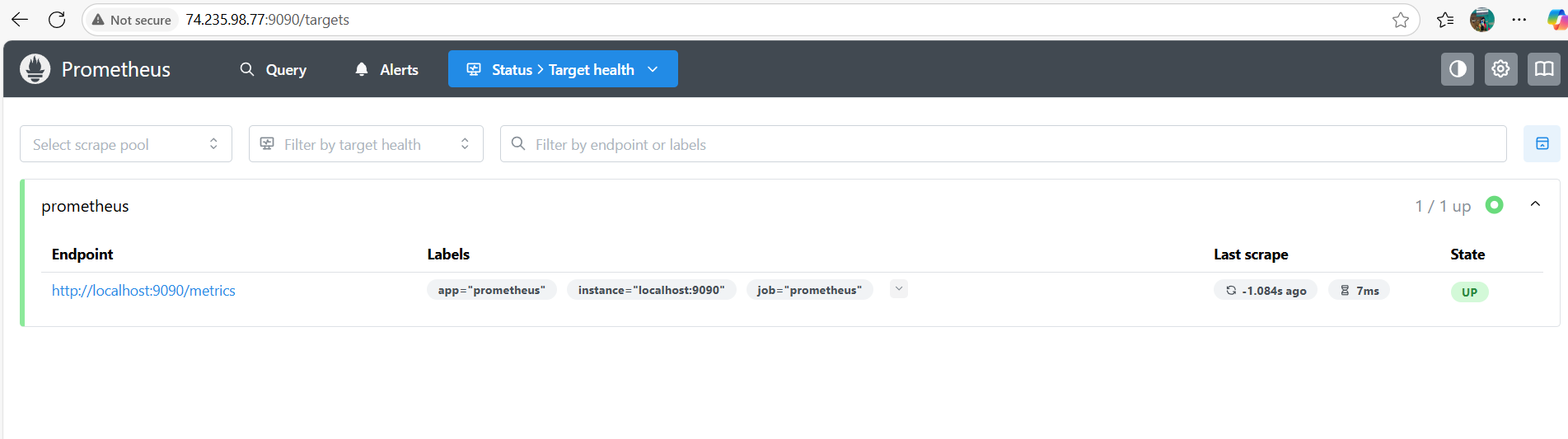
./prometheus --config.file=exporter.yml



Access it from:

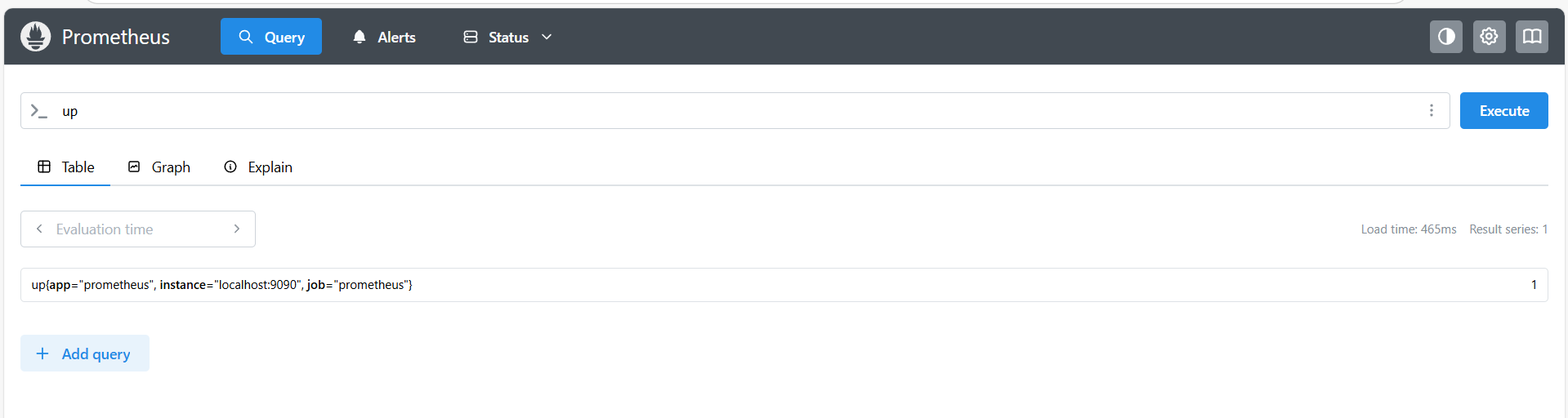
http://<your-vm-public-ip>:9090





We have query:

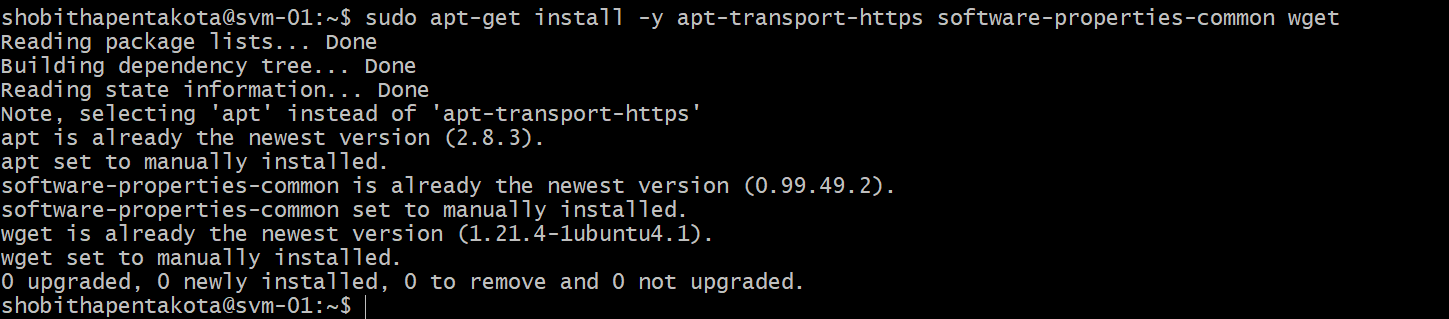
Up🡪execute



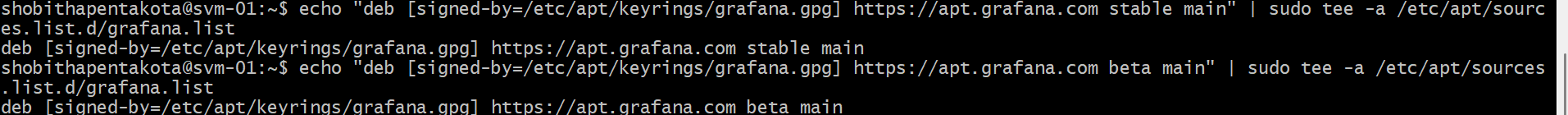
**🔹 Step 8: Install Grafana:**

**Download and install Grafana:**

sudo apt-get install -y software-properties-common

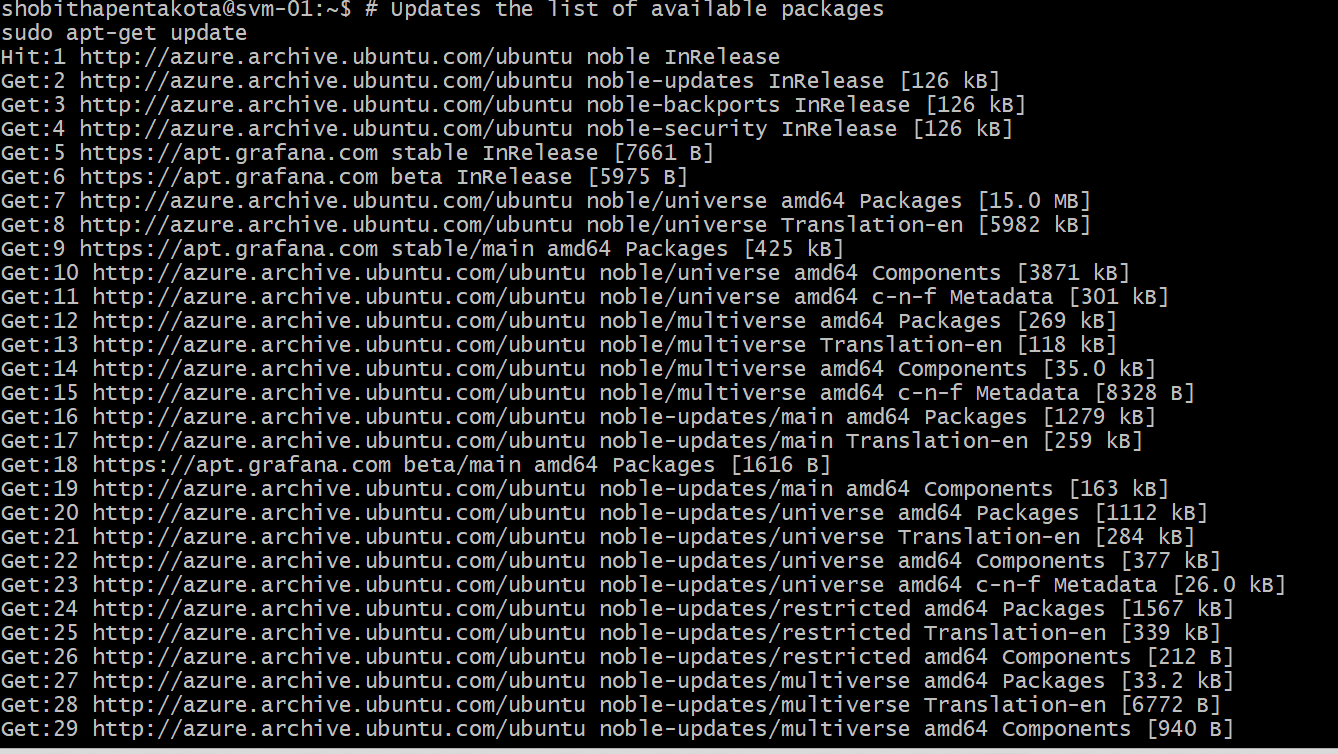


sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"



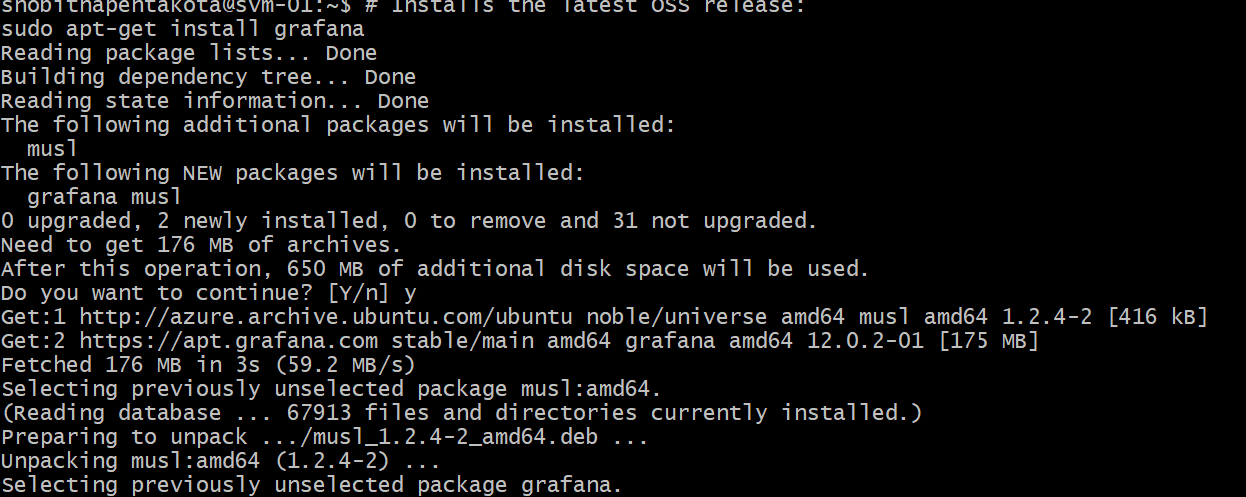
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

sudo apt-get update





sudo apt-get install grafana -y

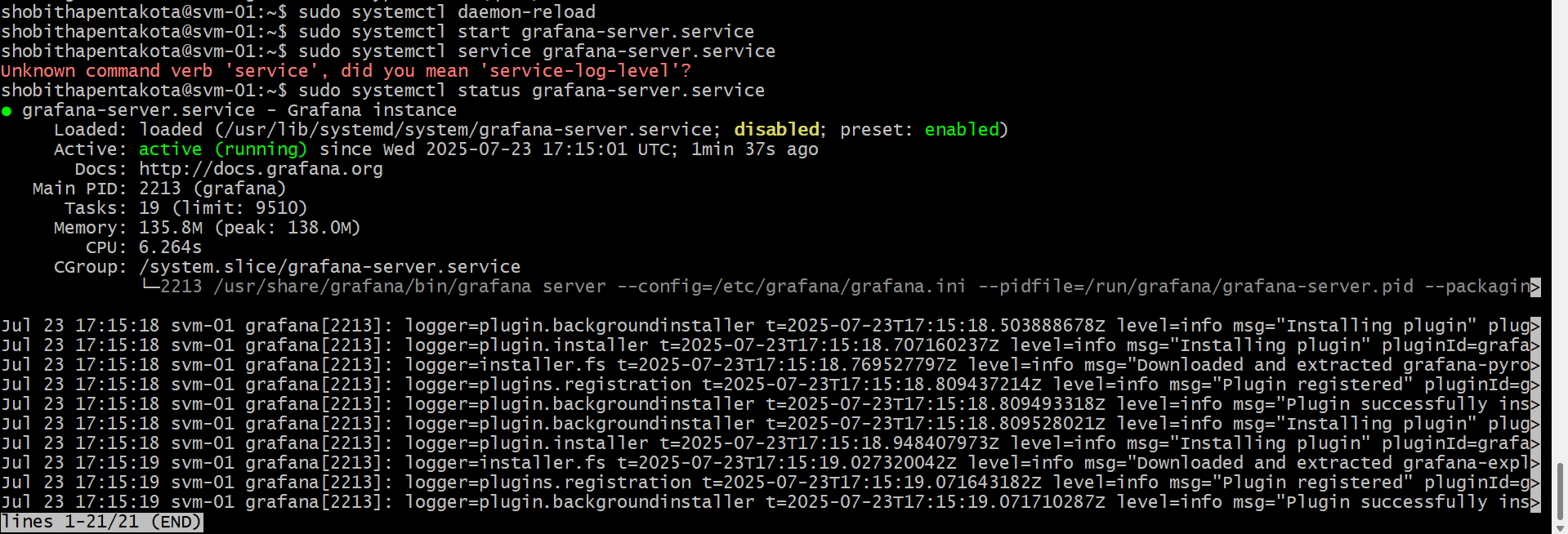


**Start and enable the service:**

sudo systemctl daemon-reload

sudo systemctl start grafana-server.service

sudo systemctl status grafana-server.service

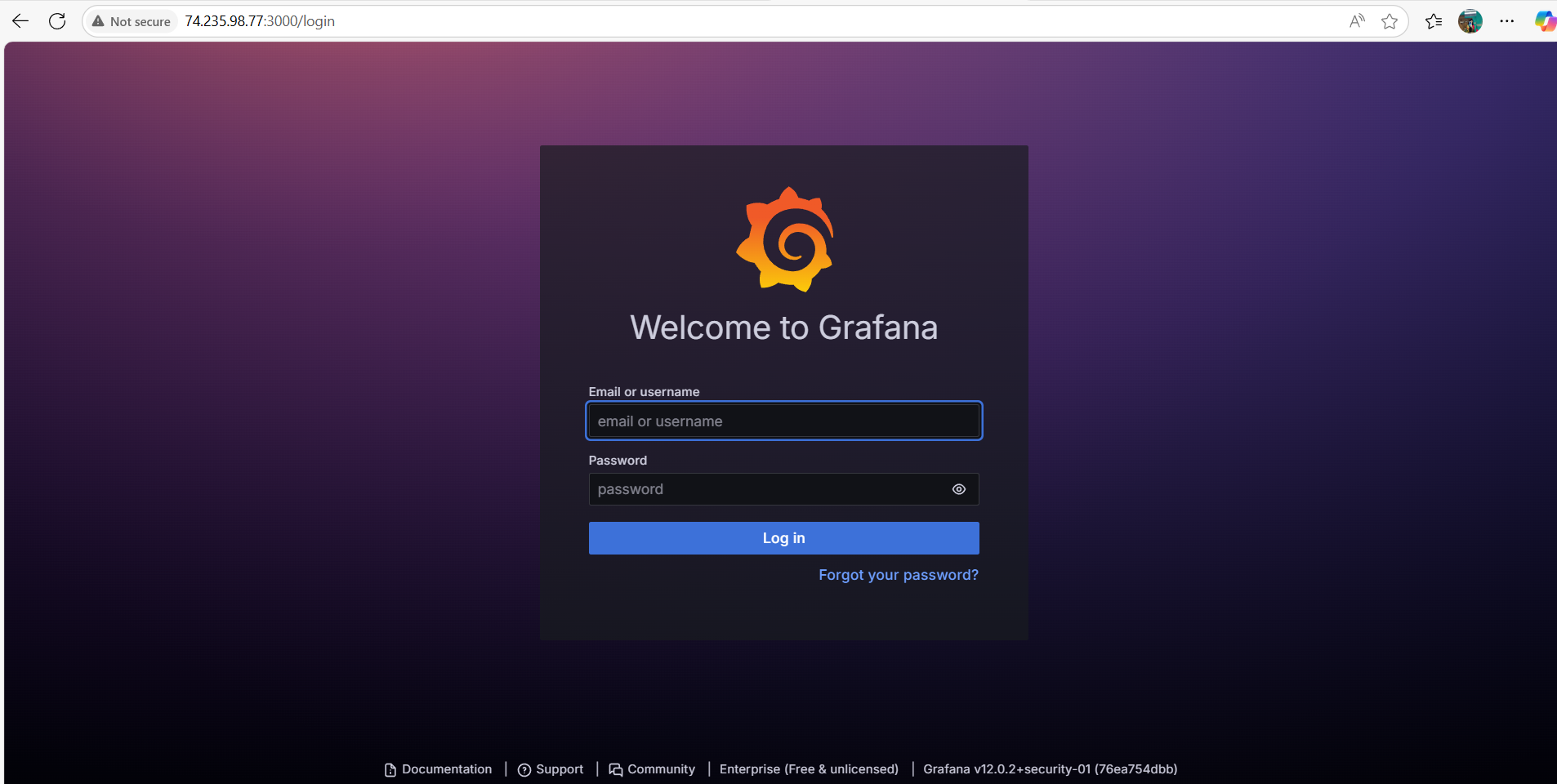


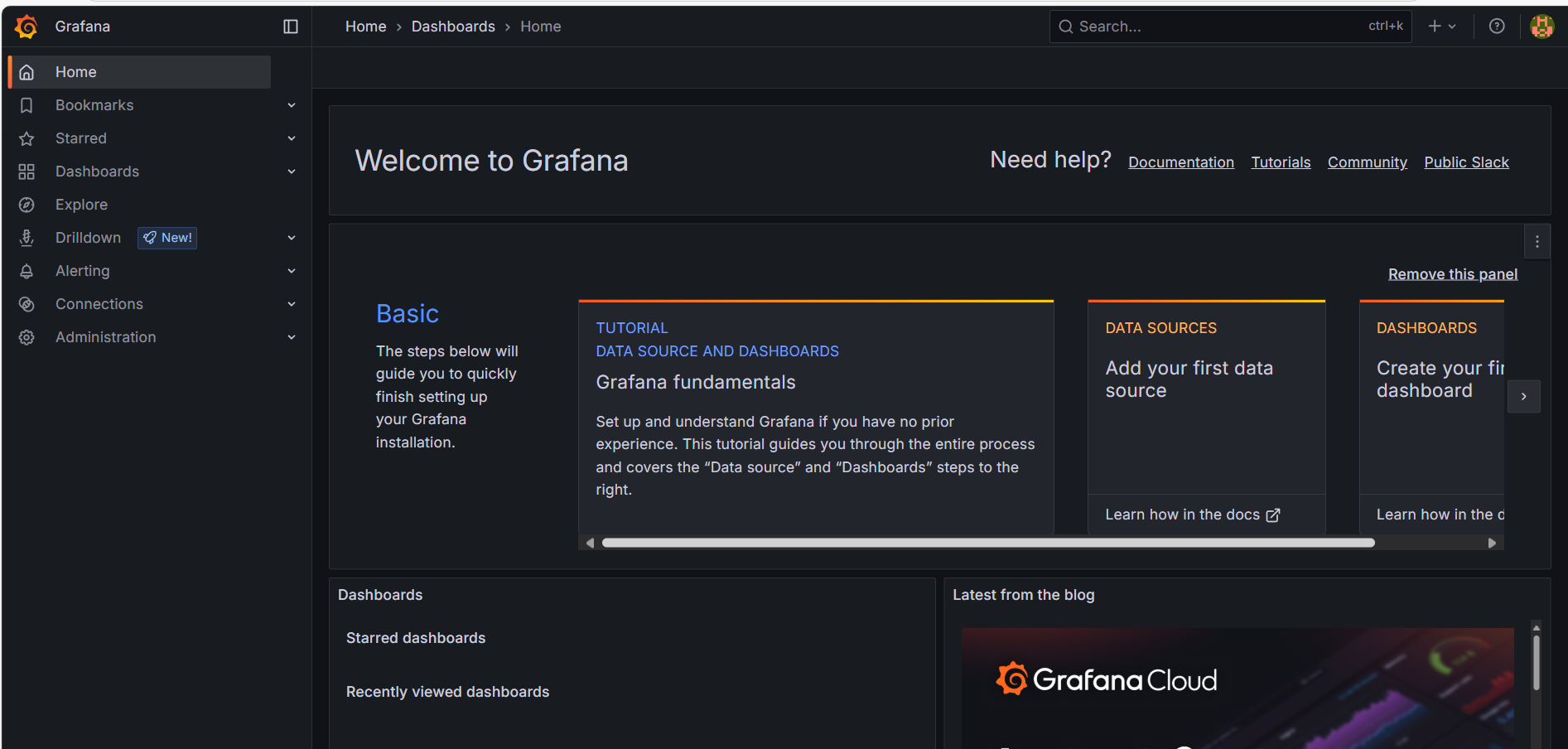
**Access Grafana from:**

http://<your-vm-ip>:3000

Login:

* **Username**: admin
* **Password**: admin





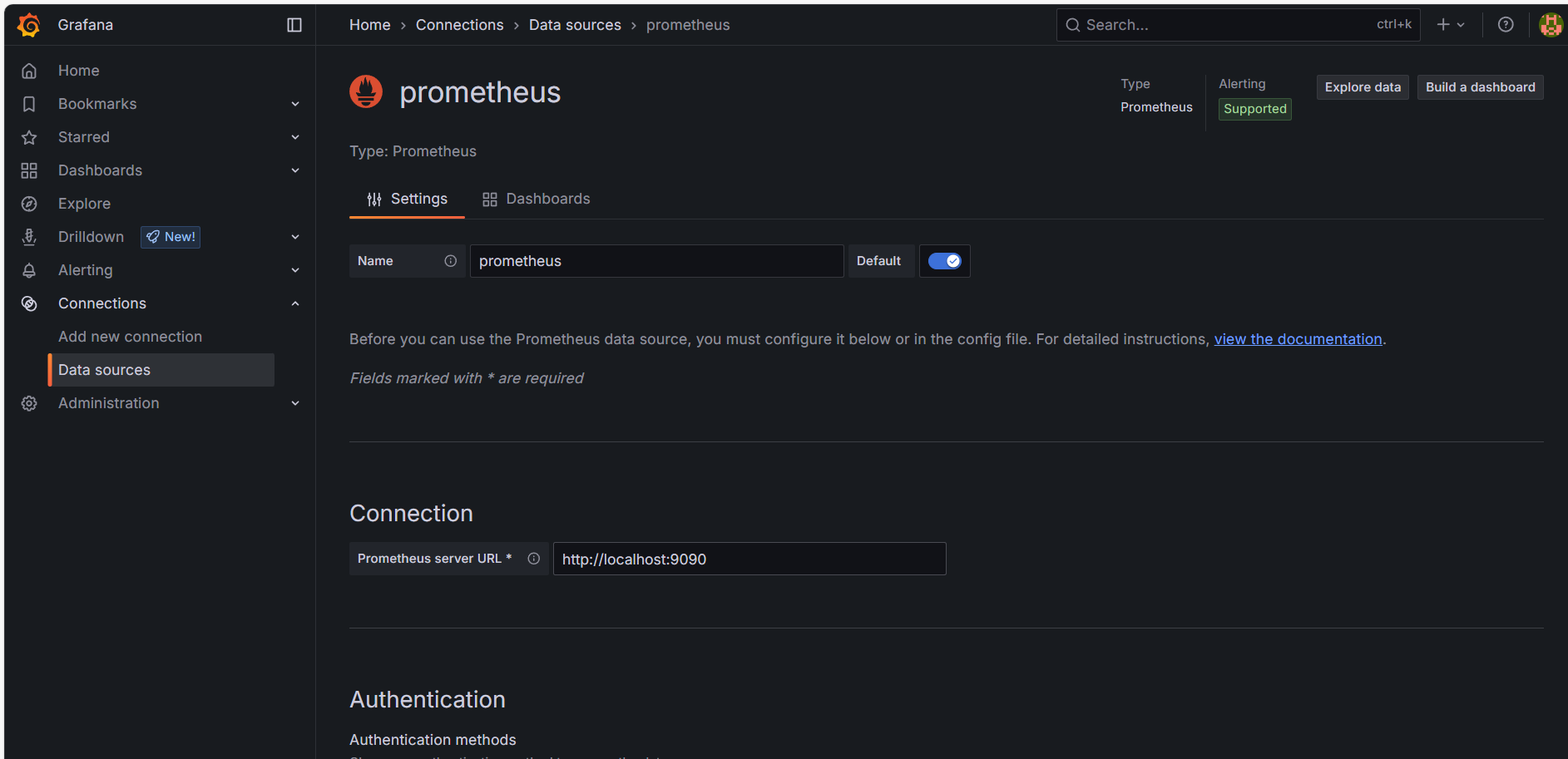
**🔹 Step 9: Connect Prometheus as Data Source in Grafana**

1. In Grafana → Go to Connections → Data Sources

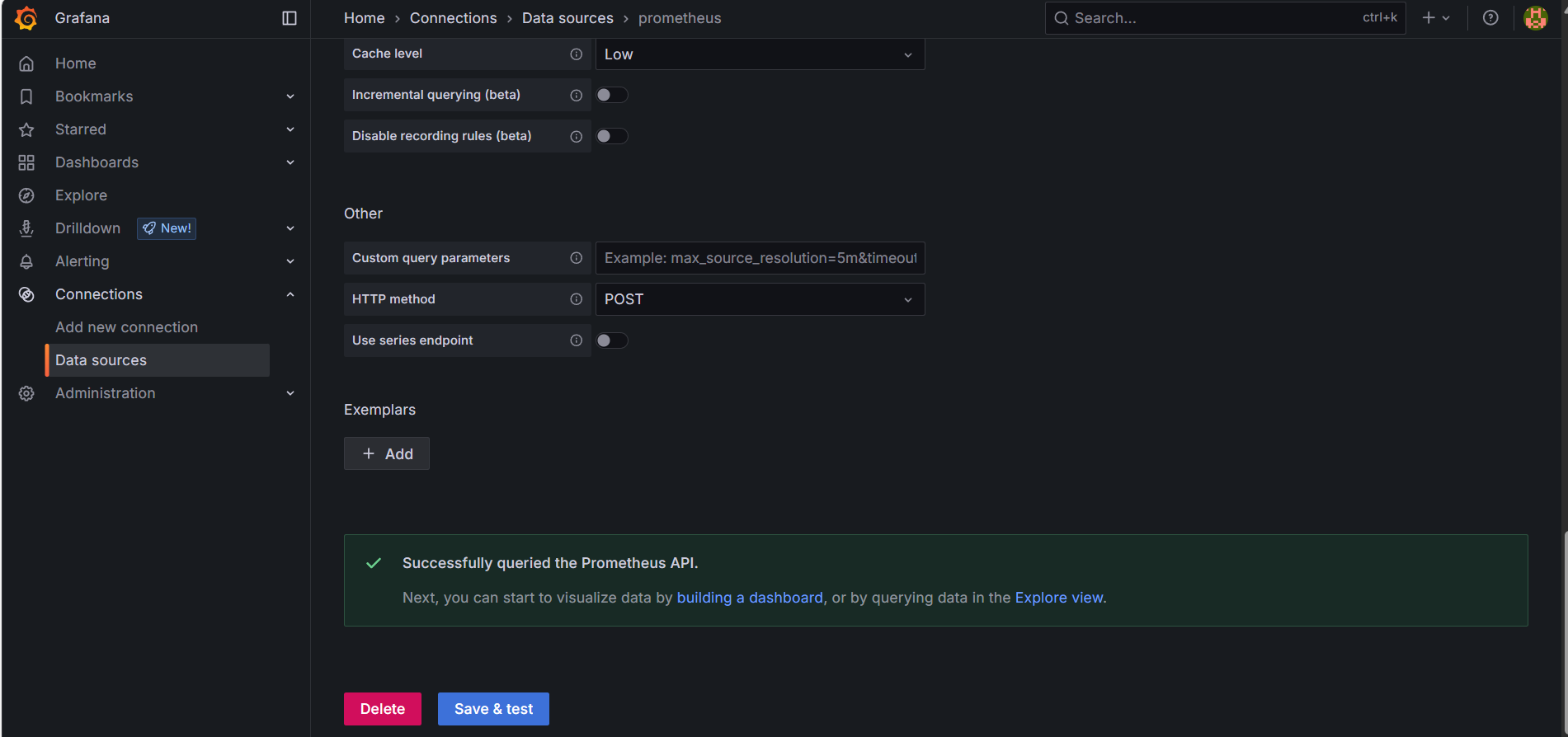


1. Choose **Prometheus**
2. Set URL:

<http://localhost:9090>

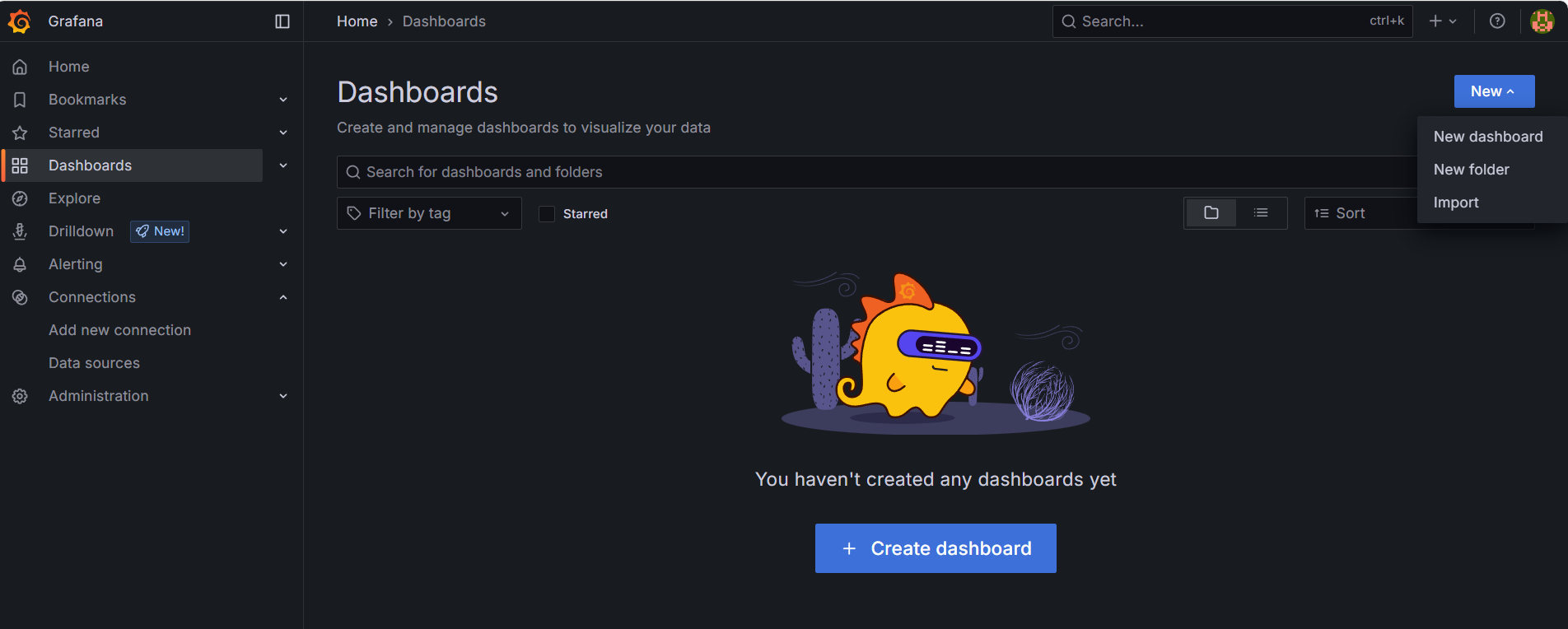


1. Save and Test – should be successful ✅

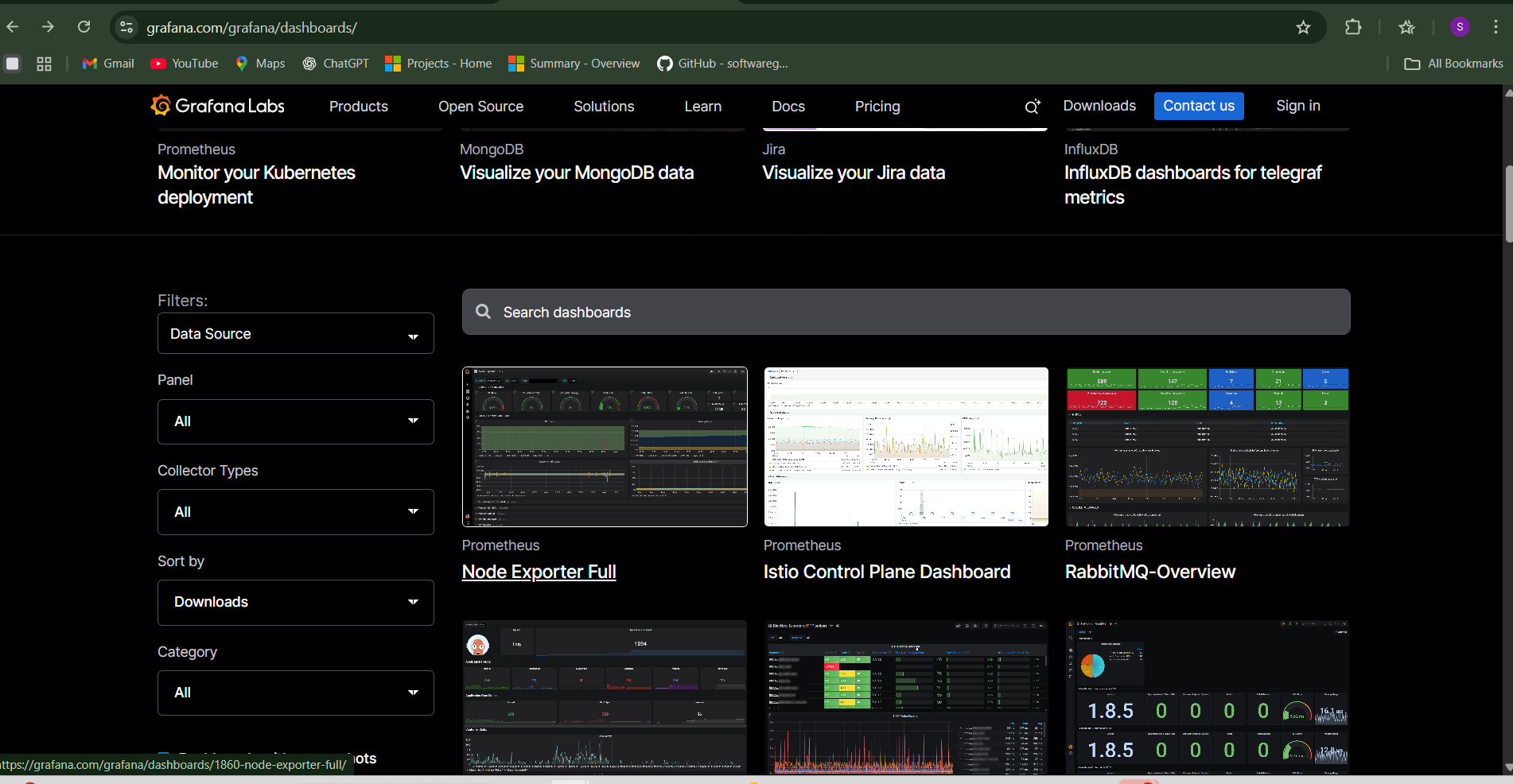


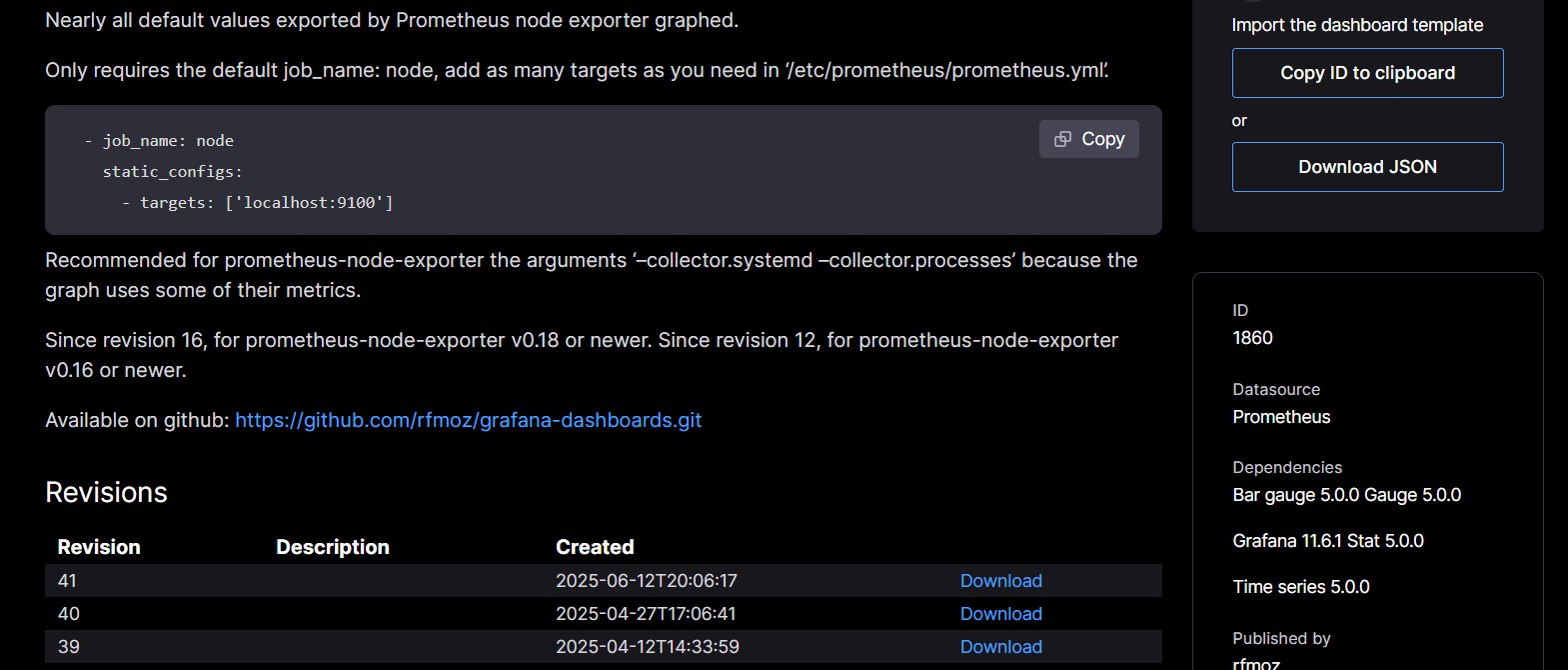
**🔹 Step 10: Import Dashboard**

1. Go to Grafana Dashboards

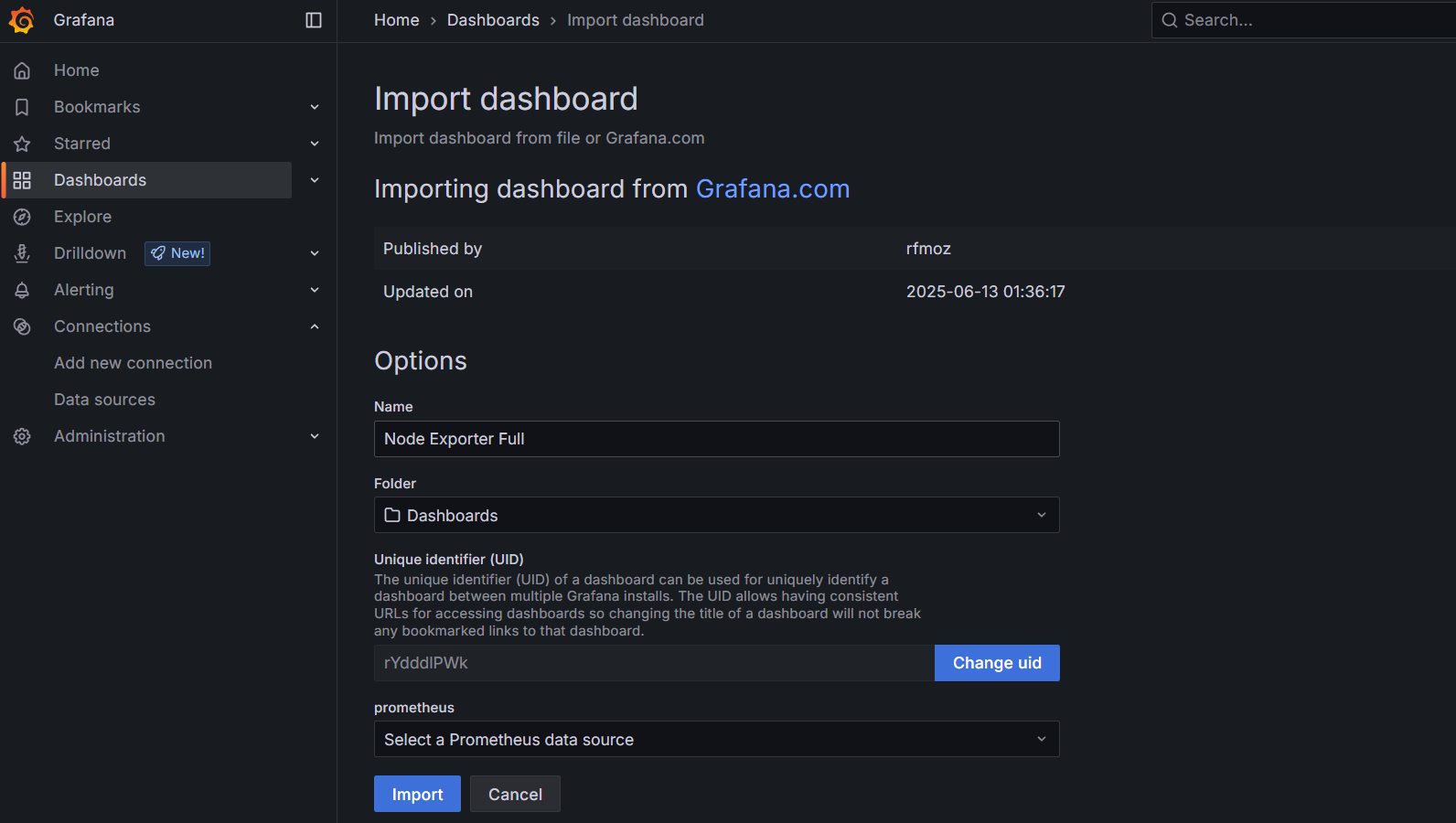


Go to Grafana labs 🡪open Grafana dashboards🡪id



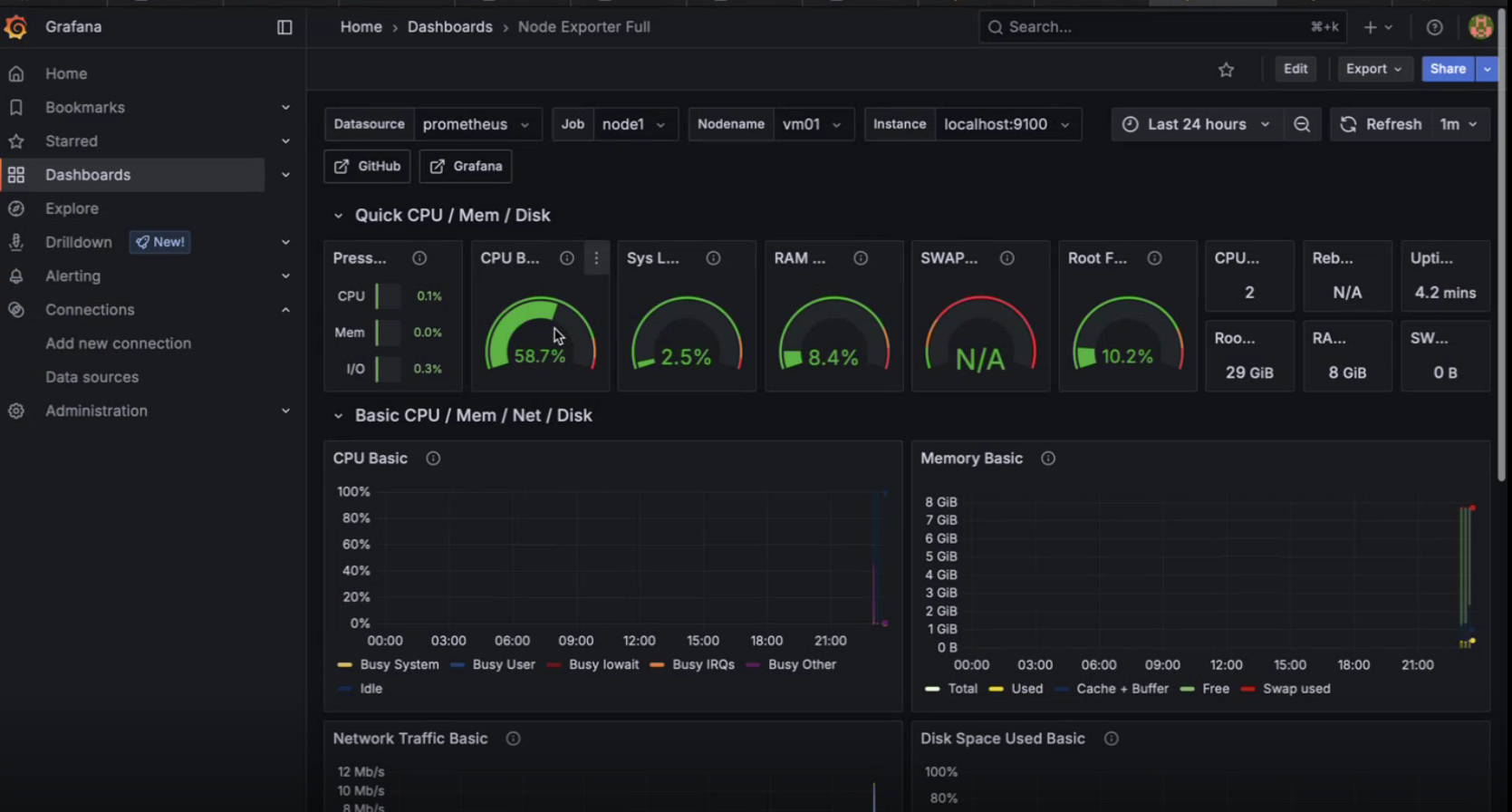


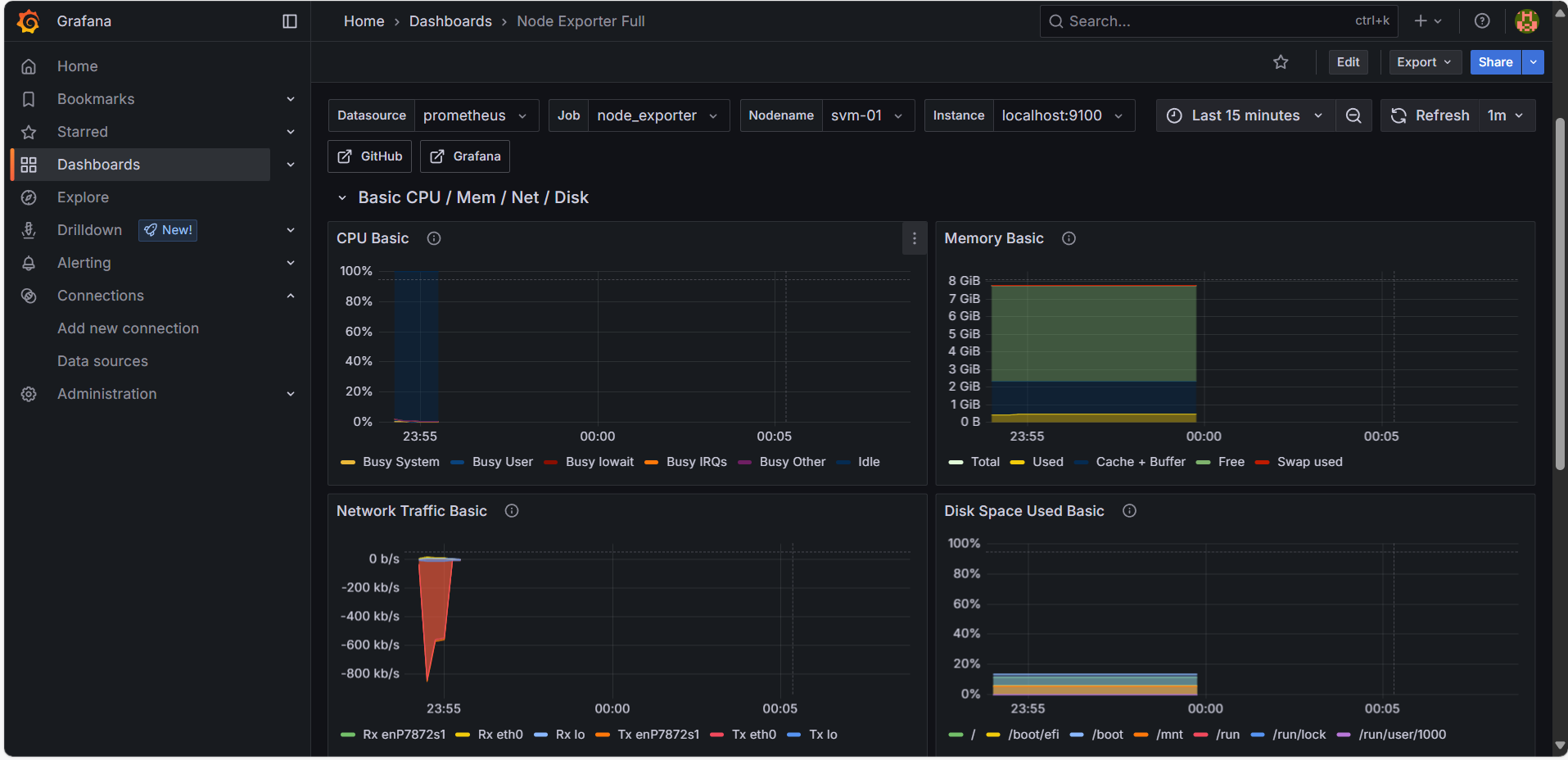
1. Copy a dashboard **ID** (e.g., 1860 for Node Exporter Full)
2. In Grafana → **Import Dashboard**
   * Paste ID



* + Choose Prometheus as data source
  + Click **Import**

Final output:





**📊 Prometheus:**

🔍 What is Prometheus?

Prometheus is an open-source monitoring and alerting toolkit built for recording real-time metrics in a time-series database, using a pull model over HTTP.

| **Feature** | **Description** |
| --- | --- |
| **Time-series DB** | Stores metrics with timestamps (e.g., CPU usage every 15s) |
| **Pull model** | Prometheus **pulls** data from configured targets like node\_exporter |
| **PromQL** | Powerful **query language** for filtering and analyzing metrics |
| **Service Discovery** | Automatically finds instances (in Kubernetes, EC2, etc.) |
| **Alertmanager** | Sends alerts via email, Slack, etc., based on rules |
| **Exporters** | Applications like **node\_exporter** expose OS/hardware metrics |

**🧠 How Prometheus Works:**

1. **Targets** (like node\_exporter) expose metrics on /metrics endpoint.
2. Prometheus **scrapes** those endpoints every 15s or so.
3. The data is stored in **Prometheus TSDB**.
4. You can query this data using **PromQL**.
5. **Grafana** can visualize it beautifully.

**🔧 Exporters in Prometheus**

* **node\_exporter** → OS metrics (CPU, memory, disk)
* **blackbox\_exporter** → Probes HTTP/TCP endpoints
* **mysql\_exporter**, **nginx\_exporter**, etc. → Application-specific exporters

**✅ What is Node Exporter?**

**Node Exporter** is a **Prometheus exporter** for **hardware and OS-level metrics** from *Linux* or *Unix* systems.

It exposes **metrics about the system’s health**, like:

* CPU usage
* Memory usage
* Disk I/O
* Filesystem stats
* Network traffic
* System load
* Temperature, entropy, etc.

These metrics are **scraped by Prometheus** and **visualized in Grafana**.

**📈 What is Grafana?**

**Grafana** is an open-source **dashboarding tool** used to **visualize data** from various sources like:

* Prometheus ✅
* MySQL
* Elasticsearch
* InfluxDB
* Loki (for logs)

| **Feature** | **Description** |
| --- | --- |
| **Visual Dashboards** | Charts, graphs, gauges, tables |
| **Data source plugin** | Easily connect Prometheus, Elasticsearch, etc. |
| **Alerts & Notifications** | Set thresholds and get notified |
| **Templating** | Create reusable and dynamic dashboards |
| **User Management** | Supports login, roles, organizations |

**How Grafana Works:**

1. Grafana connects to **Prometheus** as a **data source**.
2. You import or create dashboards that use **PromQL queries**.
3. Dashboards show graphs based on scraped data.
4. You can set **alerts** (e.g., CPU > 90%) and send notifications.