**How to Install Prometheus, Node-Exporter & Grafana on Ubuntu**

**Prometheus Installation on Ubuntu (Main Server)**

**Prometheus:** Prometheus is an open-source monitoring tool that monitors and alerts users about systems & applications. It exposes a web server on a designated port (default is 9090).

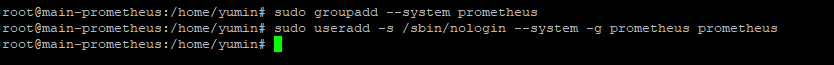
**Step – 1: Update System Packages**

sudo apt-get update

**Step – 2: Create System User for Prometheus**

sudo groupadd –-system prometheus

sudo useradd –s /sbin/noligin –-system –g prometheus prometheus



This will create a system user & group named “Prometheus” for Prometheus with limited limited privileges, reducing the risk of unauthorized access.

**Step – 3: Create Directories for Prometheus**

sudo mkdir /etc/prometheus

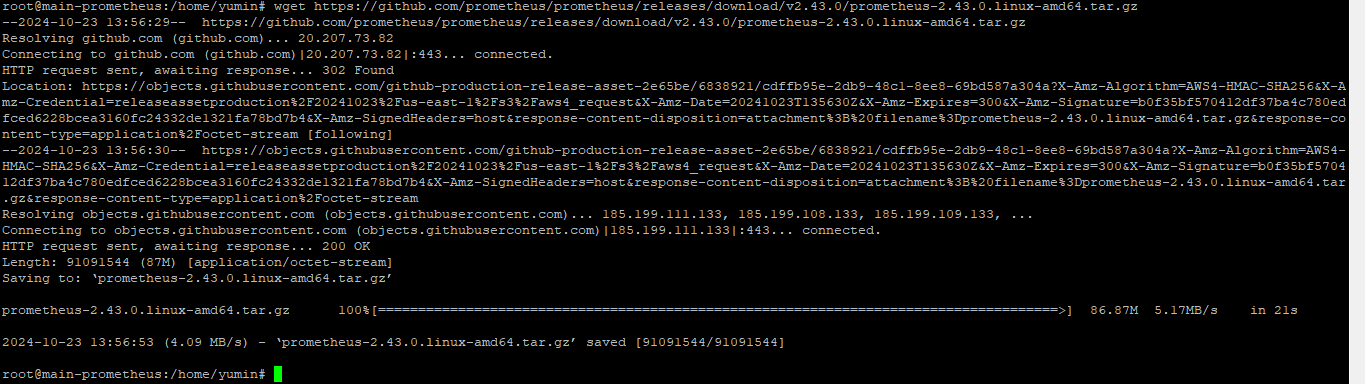
sudo mkdir /var/lib/prometheus

To store configuration files and libraries for Prometheus, we need to create a directories. The directories located in /etc & /var/lib directories.

**Step – 4 : Download Prometheus & Extract files**

wget <https://github.com/prometheus/prometheus/releases/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz>

tar –xvzf prometheus\*.tar.gz



**Step – 5: Navigate to Prometheus Directory**

cd prometheus-2.43.0.linux-amd64 (or) cd Prometheus\*

**Configuring Prometheus on Ubuntu (Main Server)**

**Step – 6: Move the Binary files & Change the Ownership**

sudo mv prometheus /usr/local/bin

sudo mv promtool /usr/local/bin

sudo chown prometheus:prometheus /usr/local/bin/prometheus

sudo chown prometheus:prometheus /usr/local/bin/promtool

Moving binary files (Prometheus & promtool) and change the ownership of the files to the prometheus user & group.

**Step – 7:** **Move the Configuration files & Change the Ownership**

sudo mv consoles /etc/prometheus

sudo mv console\_libraries /etc/prometheus

sudo mv prometheus.yml /etc/prometheus

sudo chown prometheus:prometheus /etc/prometheus

sudo chown -R prometheus:prometheus /etc/prometheus/consoles

sudo chown -R prometheus:prometheus /etc/prometheus/console\_libraries

sudo chown -R prometheus:prometheus /var/lib/prometheus

The prometheus.yml file is the main Prometheus configuration file. It includes settings for targets to be monitored, data scraping frequency etc.

sudo nano /etc/prometheus/prometheus.yml

**Step – 8: Create Prometheus Systemd Service**

Create and open a system service file for prometheus “prometheus.service” file with the nano/vim/vi text editor

nano /etc/systemd/system/prometheus.service

Copy & Paste the below mentioned lines, This configuration file defines a systemd service for running Prometheus, a monitoring and alerting, ensuring it starts after the network is online and runs with specified user permissions and configurations.

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

Group=prometheus

Type=simple

ExecStart=/usr/local/bin/prometheus \

--config.file /etc/prometheus/prometheus.yml \

--storage.tsdb.path /var/lib/prometheus/ \

--web.console.templates=/etc/prometheus/consoles \

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

[Install]

WantedBy=multi-user.target

Save & Exit

**Step – 9: Reload Systemd, enable & start Prometheus**

systemctl daemon-reload -> refreshes the systemd manager configuration, ensuring it recognizes any changes made to service files.

systemctl enable prometheus -> enable the Prometheus service to start on boot.

systemctl start prometheus -> start the service immediately.

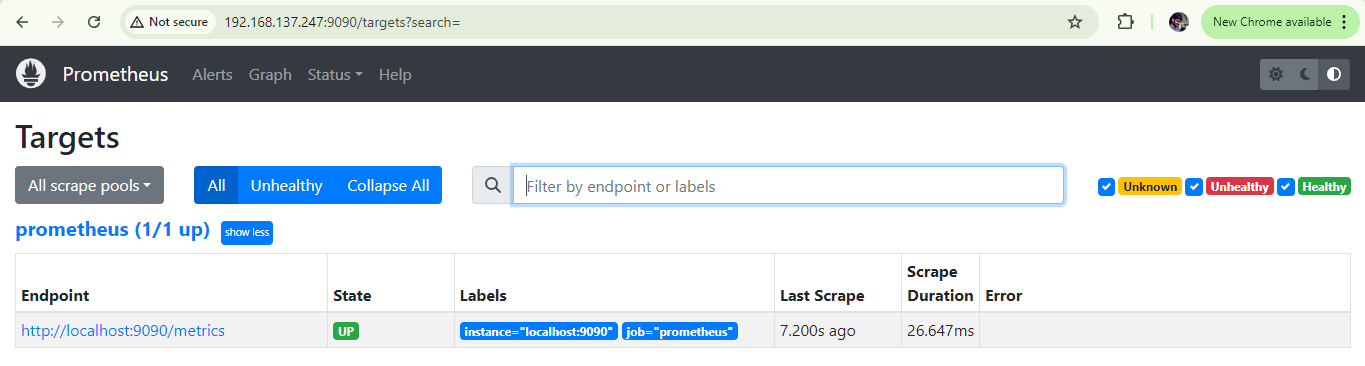
systemctl status Prometheus -> check its current status of the service.

**Step – 10: Access Prometheus Web Interface**

hostname –I (or) ip a (or) ifconfig -> These commands will display the network interfaces and their corresponding IP addresses.



<http://192.168.137.247:9090>



If in web interface <http://192.168.137.247:9090> shows error please check the firewall.

sudo ufw status

sudo ufw allow 9090/tcp

**Node Exporter Installation on Ubuntu (Main Server)**

Node Exporter is a tool widely used in the monitoring and alerting ecosystem, particularly within the Prometheus monitoring system. It is designed to collect and export a wide range of hardware and OS metrics from Unix/Linux servers, allowing for detailed monitoring of system resources such as CPU, memory, disk utilization, network statistics, and many more.

Node Exporter runs as an agent on each server you wish to monitor. It exposes a web server on a designated port (default is 9100).

**Step – 1: Download Node Exporter & Extract the Files**

wget <https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz>

tar –xvzf node\_exporter-1.7.0.linux-amd64.tar.gz

**Step – 2: Moves Files**

sudo mv node\_exporter-1.7.0.linux-amd64 /etc/node\_exporter

**Step – 3: Create Service**

sudo nano /etc/systemd/system/node\_exporter.service

Copy & Paste the below mentioned lines, file allows Node Exporter to run as a service, ensuring it is available for Prometheus to scrape metrics.

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

[Service]

ExecStart=/etc/node\_exporter/node\_exporter

Restart=always

[Install]

WantedBy=multi-user.target

**Step – 4: Run Node Exporter**

sudo systemctl daemon-reload

sudo systemctl enable node\_exporter

sudo systemctl start node\_exporter

sudo systemctl restart node\_exporter

**Step – 5: To integrate Node Exporter with Prometheus**

**Open Prometheus configuration file**

sudo nano /etc/prometheus/prometheus.yml

Paste the below mentioned lines

- job\_name: "Local Server"

static\_configs:

# - targets: ["<IP\_ADDRESS>:9100"]

- targets: ["localhost:9100"] -> Server IP(192.168.137.247)

Save & Exit

These lines tell Prometheus to collect metrics from Node Exporter running on the local machine at port 9100, which is where Node Exporter exposes its metrics.

**Node Exporter Installation on Ubuntu (Client Server)**

Follow the same steps mentioned earlier to install Node Exporter.

**Open Prometheus configuration file**

sudo nano /etc/prometheus/prometheus.yml

Paste the below mentioned lines

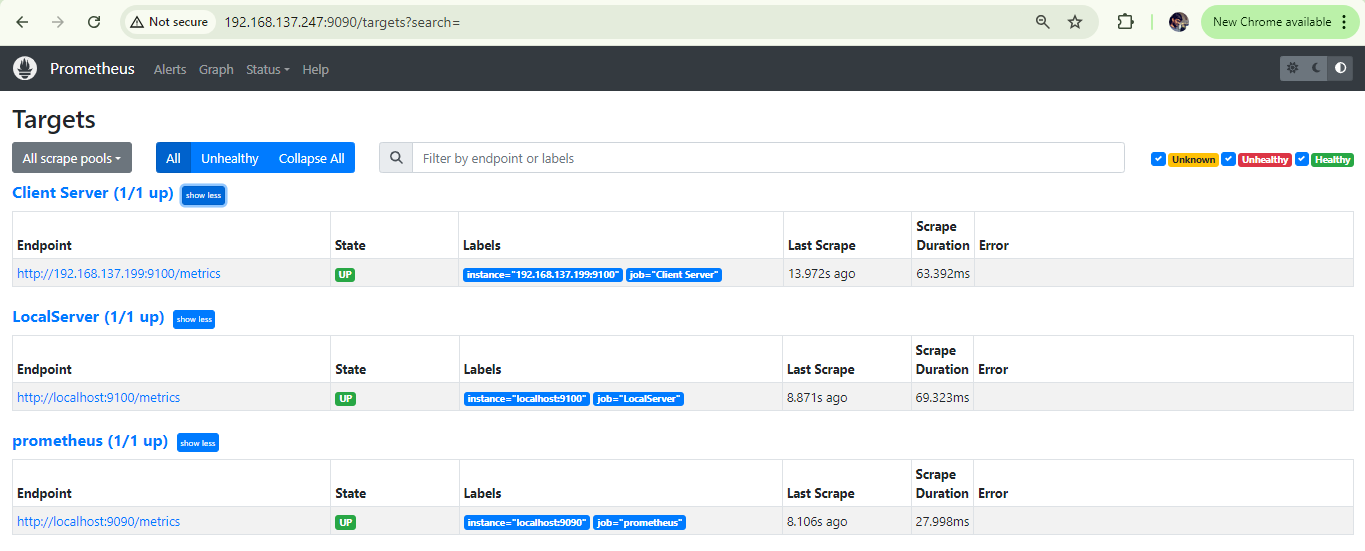
- job\_name: "Local Server"

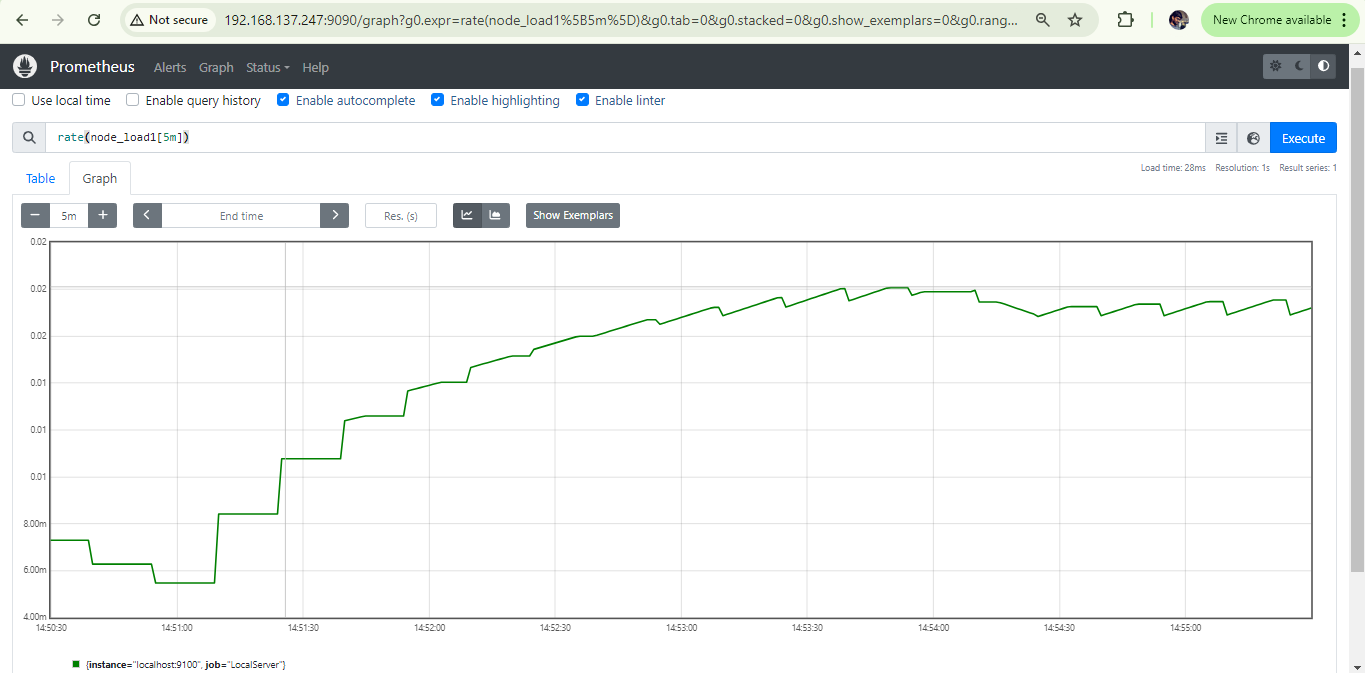
static\_configs:

# - targets: ["<IP\_ADDRESS>:9100"]

- targets: ["localhost:9100"] -> Server IP(192.168.137.199)

Save & Exit





**Grafana Installation on Ubuntu (Main Server)**

Grafana is a free & open Source tool that helps to visualize and monitor data from different sources. It lets you create custom dashboards to display metrics and logs in an easy-to-understand way. You can use it with various data sources like Prometheus, InfluxDB & MySQL. It exposes a web server on a designated port (default is 3000).

Using Grafana with Prometheus enhances data visualization, allowing for customizable dashboards and easy analysis.

**Step – 1: Installation**

sudo apt-get install -y adduser libfontconfig1 musl

wget https://dl.grafana.com/oss/release/grafana\_10.4.5\_amd64.deb

sudo dpkg -i grafana\_10.4.5\_amd64.deb

**Step – 2: Run Grafana**

sudo systemctl daemon-reload

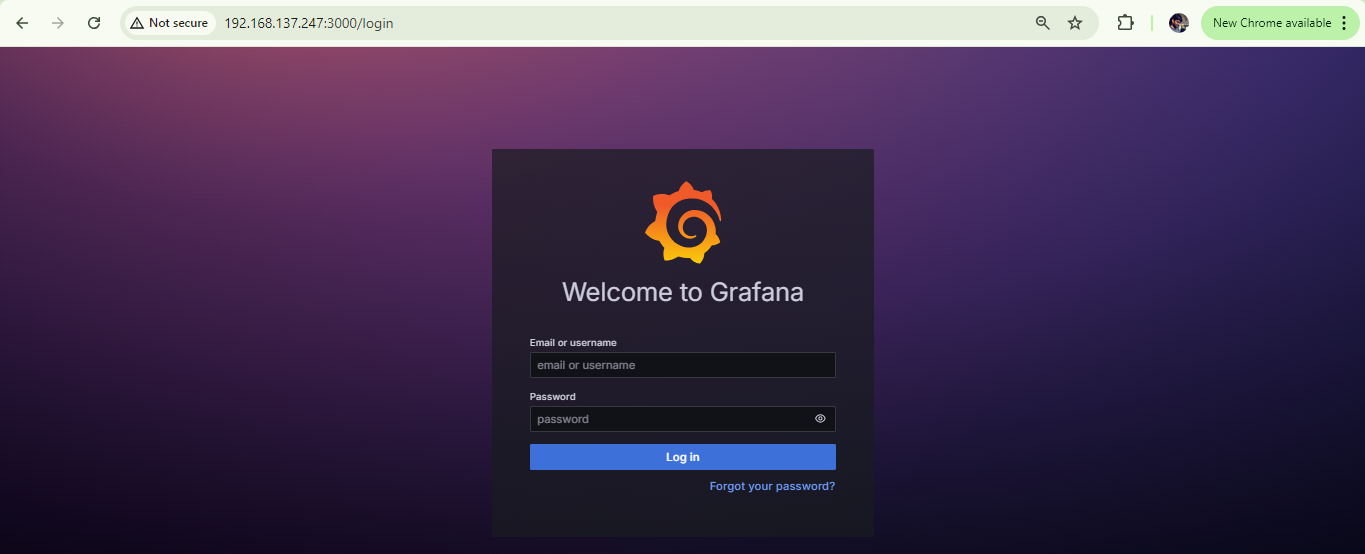
sudo systemctl enable grafana-server

sudo systemctl start grafana-server

sudo systemctl restart grafana-server

**Step – 3:** **Access Grafana Web Interface**

<http://192.168.137.247:3000/login>



To connect Prometheus as a data source in Grafana, follow these steps:

### Prerequisites:

* **Prometheus** is already installed and running.
* **Grafana** is installed and running.

### Steps to Connect Prometheus as a Data Source:

1. **Log in to Grafana**:

* Open your web browser and go to the Grafana URL (usually something like http://localhost:3000 or the IP address where Grafana is hosted).
* Log in with your credentials. The default login is usually admin/admin unless changed during setup.

1. **Navigate to Data Sources**:

* In the left-hand side menu, click on the **gear icon** (⚙️) labeled **Configuration**.
* Then, click on **Data Sources**.

1. **Add Prometheus as a Data Source**:

* Click the **Add data source** button at the top.
* In the list of available data sources, select **Prometheus**.

1. **Configure Prometheus Data Source**:

* You'll be taken to the Prometheus data source configuration page.
* In the **URL** field, enter the Prometheus server URL.

1. **Test the Connection**:

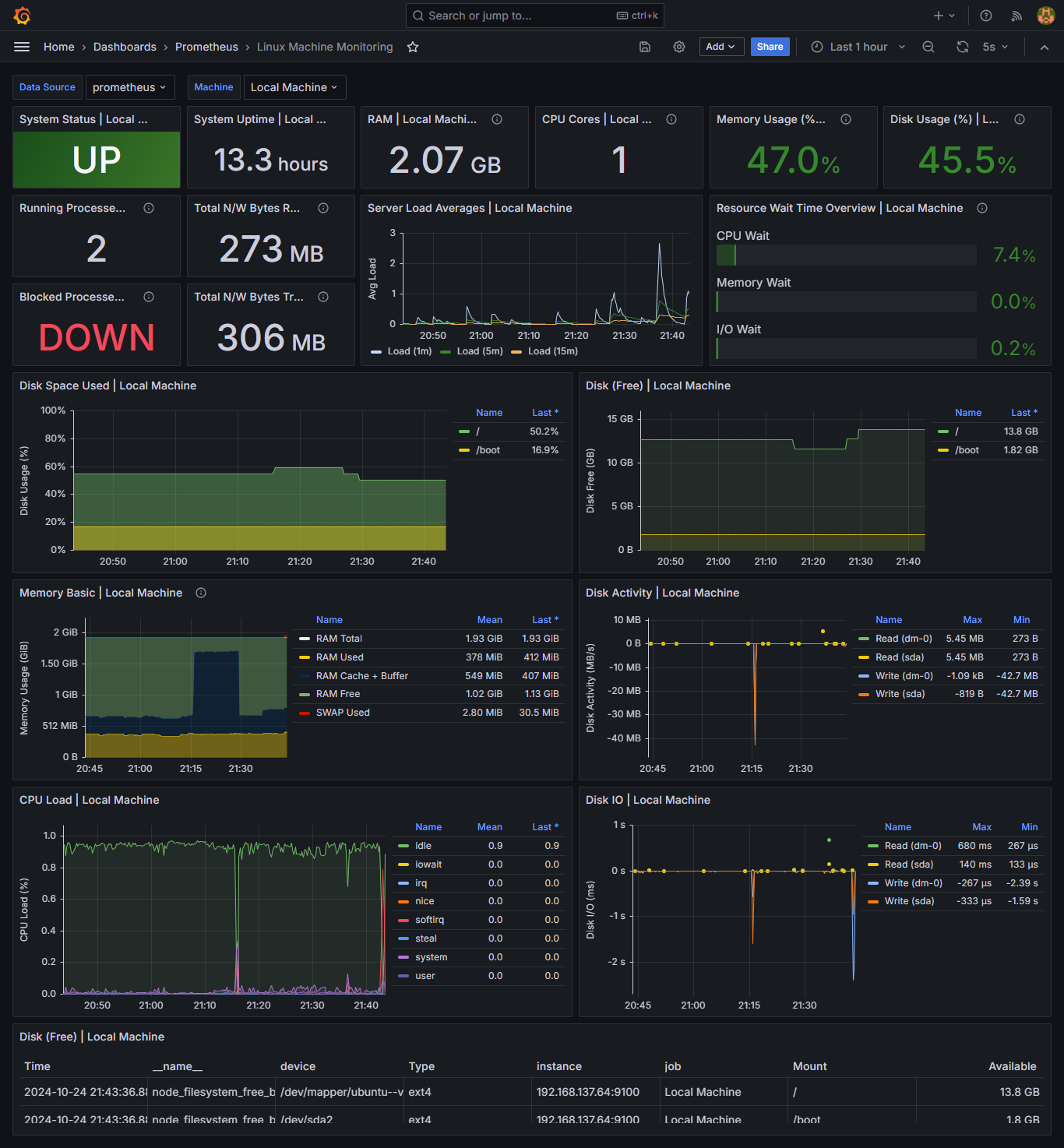
* After configuring the URL, click the **Save & Test** button at the bottom of the page.
* If successful, you’ll see a confirmation message that the data source is working.

You can download the Grafana dashboard export file from my GitHub. Once you've downloaded the file, you can import it into your own Grafana instance to use the same dashboard.

### How to Import the Dashboard:

1. Go to my **GitHub repository** and download the dashboard export file.
2. In Grafana, click on the **"+" (Create)** icon in the left-hand menu.
3. Select **Import**.
4. Upload the downloaded JSON file from GitHub.
5. Click **Load** and choose the Prometheus data source to link with the dashboard.

**Linux Machine Monitoring Dashboard**

****

### 1. **System Status (Local Machine)**:

* **What it shows:** This panel displays whether the monitored Linux system is up and running.
* **Explanation:** If the system is running, it shows "UP" in green, indicating the server is operational. If the system goes down, this status will change, alerting administrators of potential issues.

### 2. **Uptime (Local Machine)**:

* **What it shows:** The uptime of the system in hours.
* **Explanation:** This tells how long the system has been running continuously since its last reboot. A higher uptime usually indicates stable performance.

### 3. **RAM (Local Machine)**:

* **What it shows:** The amount of RAM currently in use on the system.
* **Explanation:** This panel shows the total RAM usage out of the available physical memory on the system. Monitoring this helps detect if the system is running out of memory or if it's under heavy load.

### 4. **CPU Cores (Local Machine)**:

* **What it shows:** The number of CPU cores available on the local machine.
* **Explanation:** Knowing the number of cores is essential for performance monitoring, especially to understand load distribution across multiple cores.

### 5. **Memory Usage (%)**:

* **What it shows:** The percentage of total memory being used.
* **Explanation:** This gives an at-a-glance view of memory consumption. If the percentage is too high, it could indicate memory pressure, and if it reaches near 100%, it could lead to performance degradation.

### 6. **Disk Usage (%) (Local Machine)**:

* **What it shows:** The percentage of disk space currently used on the local machine.
* **Explanation:** This panel tracks how much disk space is being consumed. If usage nears 100%, it could lead to system issues like inability to write new data.

### 7. **Running Processes**:

* **What it shows:** The number of active running processes.
* **Explanation:** Tracks the count of currently running processes. A sudden increase in processes can indicate an overload, a resource-intensive task, or a misbehaving application.

### 8. **Blocked Processes**:

* **What it shows:** Displays if any processes are blocked.
* **Explanation:** Blocked processes indicate a problem where certain tasks are waiting for resources or have encountered an error. This helps detect underlying system or application issues.

### 9. **Total N/W Bytes Received**:

* **What it shows:** The total amount of network traffic (bytes) received by the local machine.
* **Explanation:** Tracks incoming network data. High network traffic could indicate the server is handling a lot of requests or data, which could lead to network bottlenecks if not managed.

### 10. **Total N/W Bytes Transmitted**:

* **What it shows:** The total amount of network traffic (bytes) sent by the local machine.
* **Explanation:** Tracks outgoing network data. Monitoring this helps understand the network load being placed on the server and could indicate high outgoing traffic (e.g., data backups, uploads, or serving files).

### 11. **Server Load Averages (Local Machine)**:

* **What it shows:** The system's load average over 1, 5, and 15-minute intervals.
* **Explanation:** The load average gives a sense of how busy the system is. A value higher than the number of CPU cores indicates that the system is overloaded and processes are queuing for CPU time.

### 12. **Resource Wait Time Overview**:

* **What it shows:** The amount of time the CPU, memory, and I/O resources spend waiting to be used.
* **Explanation:** This panel helps detect if certain resources (like CPU or memory) are being underutilized or overburdened. High wait times can indicate performance bottlenecks.

### 13. **Disk Space Used (Local Machine)**:

* **What it shows:** A time-series graph of the disk usage percentage.
* **Explanation:** Tracks disk usage over time to detect gradual disk consumption. If the trend is rising consistently, the system could run out of disk space soon.

### 14. **Disk (Free) (Local Machine)**:

* **What it shows:** The amount of free disk space remaining on the local machine.
* **Explanation:** This panel shows how much free space is left, helping in tracking when additional storage may be needed.

### 15. **Memory Basic (Local Machine)**:

* **What it shows:** Displays memory usage, including RAM and swap usage.
* **Explanation:** It breaks down memory consumption into categories like RAM total, RAM used, RAM cached, and swap used. This is helpful for understanding how memory is allocated and where the pressure is (e.g., if swap is being used, it may indicate memory exhaustion).

### 16. **Disk Activity (Local Machine)**:

* **What it shows:** This panel shows the read/write activity on disk devices in MB/s.
* **Explanation:** Helps track the level of disk I/O happening on the system. High disk activity can impact performance and may indicate disk bottlenecks.

### 17. **CPU Load (Local Machine)**:

* **What it shows:** Displays the current CPU load, broken down by different types of CPU usage (idle, system, user, etc.).
* **Explanation:** This panel helps track CPU usage in real time, providing an overview of how much of the CPU is in use and what is consuming it. Spikes in user or system time may indicate heavy processing loads.

### 18. **Disk I/O (Local Machine)**:

* **What it shows:** A time-series graph of disk input/output latency in milliseconds.
* **Explanation:** This tracks the speed of reading/writing to disk. High disk I/O latency can slow down applications, so monitoring this is crucial for performance.

### 19. **Disk (Free) (Local Machine) Table**:

* **What it shows:** Lists specific disk partitions and their free space in GB.
* **Explanation:** This table gives a breakdown of free space across specific partitions, showing exactly which drives or partitions are nearing capacity.