**---------------------------------Premetheus &Grafana----------------------------------**

In IT and DevOps, to **"monitor"** means to continuously **observe**, **track**, and **analyze** the performance, health, and behavior of systems, applications, or infrastructure.

What is an moniter tool ?

Monitoring tools are used to track and visualize the health, performance, and availability of systems like servers, applications, databases, and network infrastructure. These tools help detect issues early and ensure everything is working as expected.

### ✅ ****Types of Monitoring Tools****

Monitoring tools are generally categorized based on **what they monitor** and **how they work**:

| Type | Description | Examples |
| --- | --- | --- |
| **Infrastructure Monitoring** | Monitors servers, containers, networks, CPU, memory, disk, etc. | **Prometheus**, Zabbix, Nagios, Datadog |
| **Application Performance Monitoring (APM)** | Monitors application behavior, transactions, and user experience. | New Relic, AppDynamics, Dynatrace |
| **Log Monitoring** | Collects and analyzes logs for troubleshooting and auditing. | ELK Stack (Elasticsearch, Logstash, Kibana), Graylog |
| **Network Monitoring** | Monitors network traffic, bandwidth, connectivity, and devices. | SolarWinds, PRTG, Nagios |
| **Cloud Monitoring** | Monitors cloud services like AWS, Azure, GCP. | CloudWatch (AWS), Azure Monitor, Datadog |
| **Synthetic Monitoring** | Simulates user interactions to test availability and performance. | Pingdom, Uptrends |
| **Security Monitoring** | Focuses on detecting security threats and anomalies. | Splunk, Wazuh, OSSEC |

### 🔍 ****Prometheus and Grafana: Which Type?****

| Tool | Type | Purpose |
| --- | --- | --- |
| **Prometheus** | Infrastructure Monitoring | Time-series database used to **collect and store metrics** (CPU, memory, etc.) |
| **Grafana** | Visualization / Dashboard Tool | Used to **visualize** metrics from Prometheus (and other data sources) |

They’re often used **together**:

* **Prometheus** pulls metrics from services and stores them.
* **Grafana** queries Prometheus and displays those metrics on dashboards.
* Prometheus to stores the data as time series database.

Installation of Premetheus & Grafana:

----------------------------------------------

cd /opt && wget https://github.com/prometheus/prometheus/releases/download/v2.53.4/prometheus-2.53.4.linux-amd64.tar.gz

tar xvzf [prometheus-2.53.4.linux-amd64.tar.gz](https://github.com/prometheus/prometheus/releases/download/v2.53.4/prometheus-2.53.4.linux-amd64.tar.gz)

mv prometheus-2.53.4.linux-amd64 prometheus && cd prometheus

./prometheus (or)

vim /etc/systemd/system/prometheus.service

[Unit]

Description=Prometheus Server

Documentation=https://prometheus.io/docs/introduction/overview/

After=network-online.target

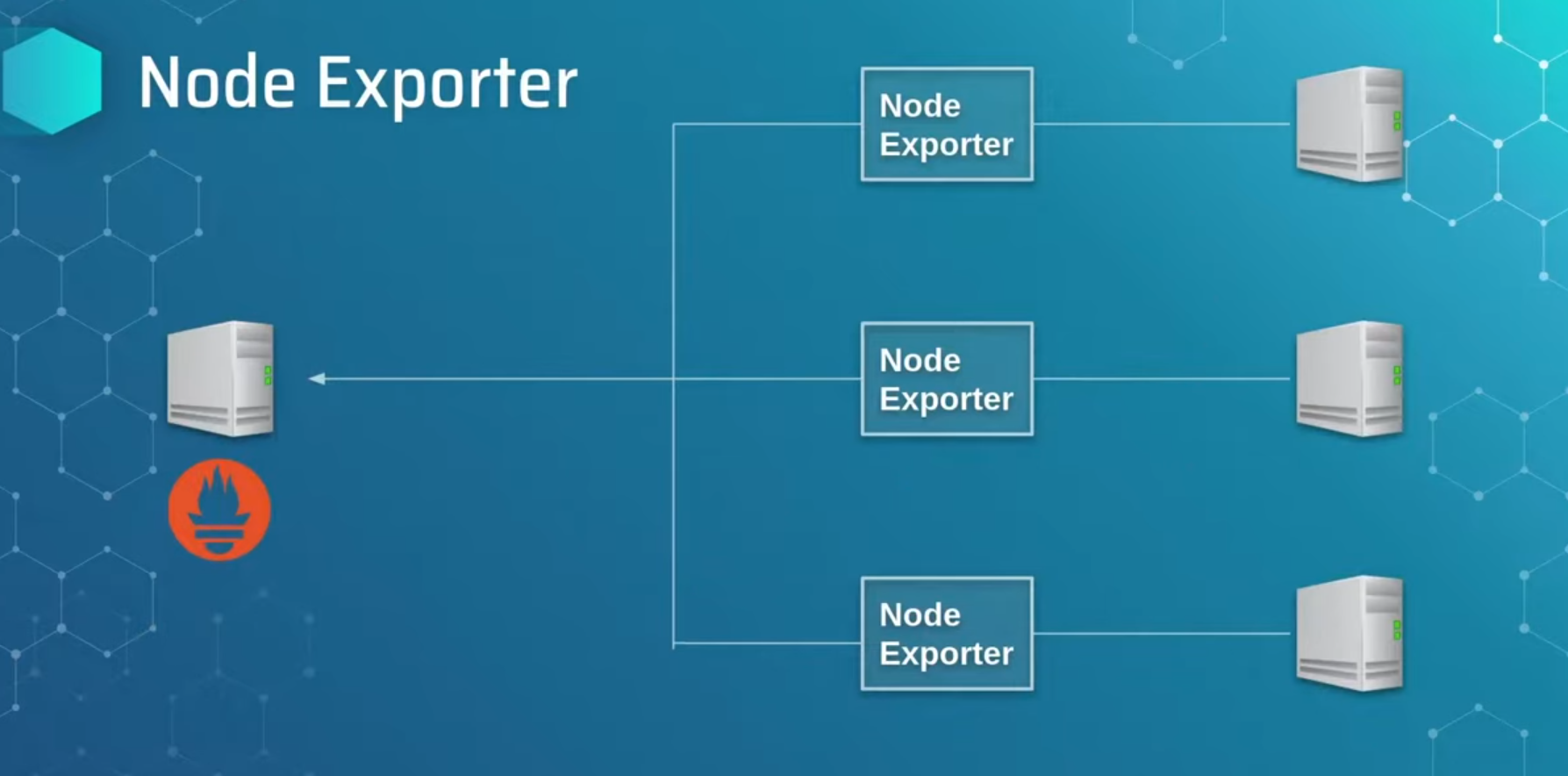
[Service]

Restart=on-failure

ExecStart=/opt/prometheus/prometheus --config.file=/opt/prometheus/prometheus.yml

[Install]

WantedBy=multi-user.target



goto browser <ip\_addr>:9090

fig 1.0

to install a node exporter in client side

cd /opt/ && wget <https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz> && tar -xvzf [node\_exporter-1.9.1.linux-amd64.tar.gz](https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz) &&

mv [node\_exporter-1.9.1.linux-amd64.tar.gz](https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz) nod\_exporter

cd node\_exporter && ./nodeexporter (or)

vim /etc/systemd/system/node\_exporter.service

[Unit]

Description= Node-exporter

After=network-online.target

[Service]

Restart=on-failure

ExecStart=/opt/node\_exporter/node\_exporter

[Insatll]

WantedBy=multi-user.target

systemctl daemon-reload

systemctl start node\_exporter

goto webrowser <ip\_server>:9100

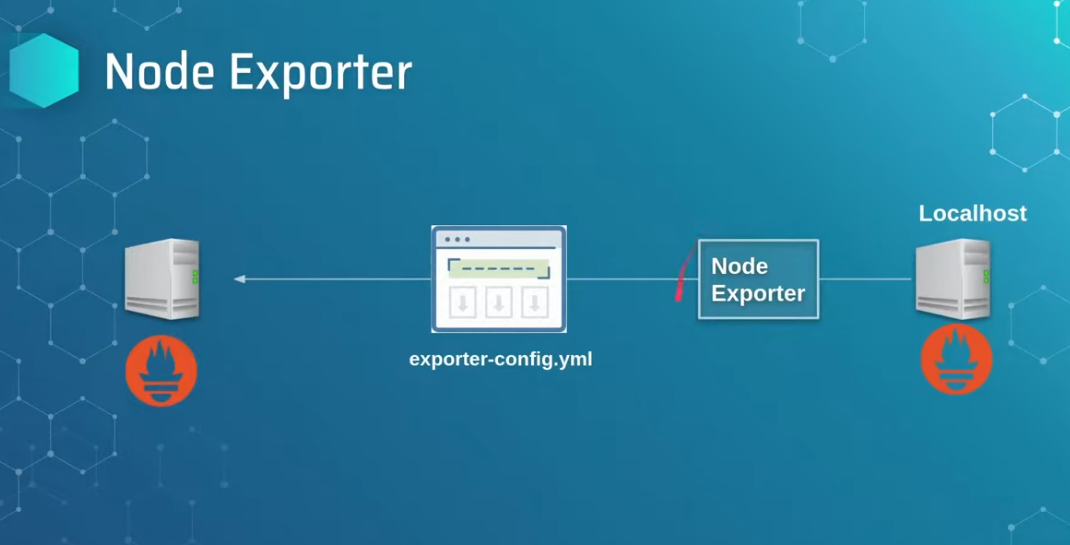


fig 1.2

if want to connect client node\_exporter and prometheus server via yaml script.

Exporter.yaml

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: node

static\_configs:

- targets: ['localhost:9100','100.0.0.3:9100']

to paste in prometheus server

vim /opt/premetheus/export.yaml

./promtool check config <export.yaml>

success the yml is ok

Instaling grafana:

--------------------

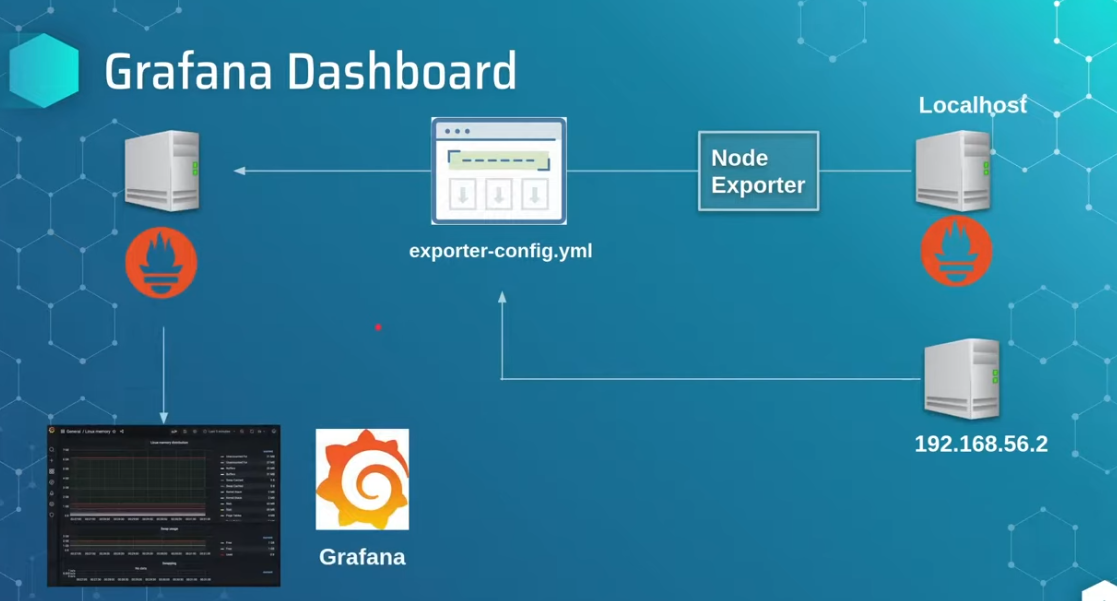
to install grafana for visualization of metrics

sudo apt-get install -y adduser libfontconfig1 musl

wget https://dl.grafana.com/oss/release/grafana\_11.6.0\_amd64.deb

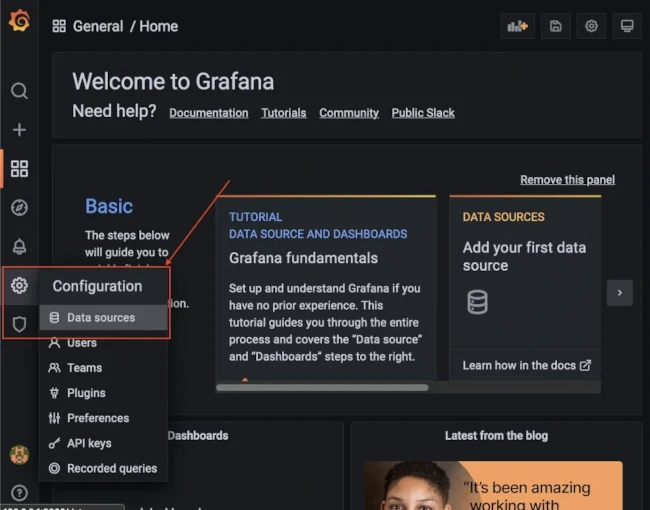
sudo dpkg -i grafana\_11.6.0\_amd64.deb

sudo systemctl start garafana.server

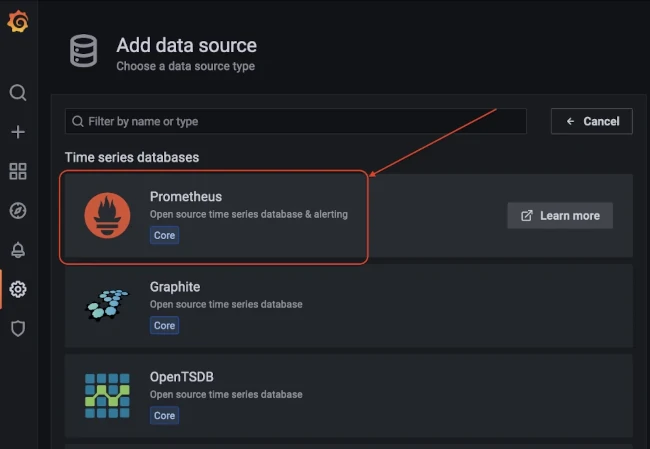
fig 1.3

go to <ip\_server>:3000

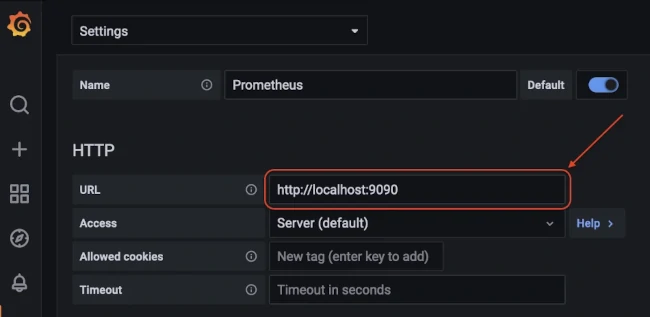
goto configuration ---→data\_sources --→add -→premetheus

fig 1.4

adding premetheus:

fig 1.5

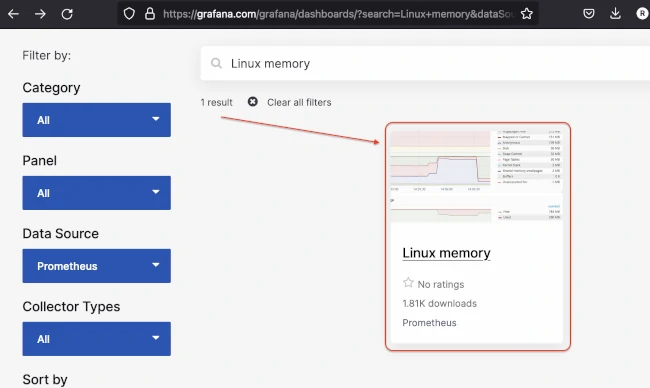
adding url

fig 1.6

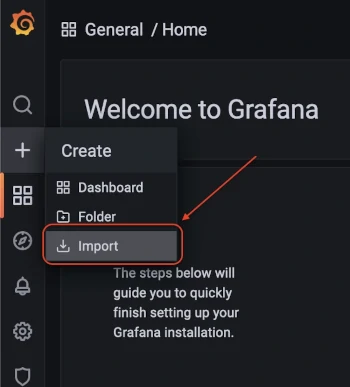
and click on save&test it prompts success.

#### Import Grafana Dashboard from Grafana Labs

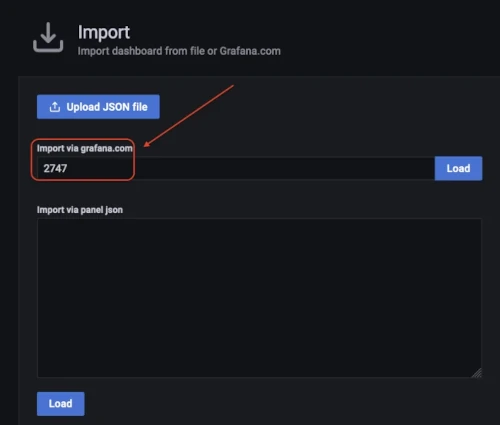
Now after settings the data source we can import pre-existing opensource dashboard from [Grafana Labs](https://grafana.com/grafana/dashboards/) using the Dashboard ID.

Fig 1.7

copy the dashboard\_id and import in grafana server.

Fig 1.8

to load the id

fig 1.9

Click load and then select the data source as **Prometheus**

Finally Click on Import

You should be able to see the memory dashboard.