**Detailed Setup Guide for Prometheus and Grafana on Ubuntu**

**1. Update and Install Dependencies**

sudo apt update && sudo apt upgrade -y

sudo apt install -y wget curl gnupg2 software-properties-common

**2. Install Prometheus**

**2.1 Create a Prometheus User**

sudo useradd --no-create-home --shell /bin/false prometheus

**2.2 Create Necessary Directories**

sudo mkdir /etc/prometheus

sudo mkdir /var/lib/prometheus

**2.3 Download and Extract Prometheus**

cd /tmp  
wget https://github.com/prometheus/prometheus/releases/download/v3.1.0/prometheus-3.1.0.linux-amd64.tar.gz

tar -xvf prometheus-3.1.0.linux-amd64.tar.gz

cd prometheus-3.1.0.linux-amd64

**2.4 Move Binaries and Set Permissions**

sudo mv prometheus /usr/local/bin/

sudo mv promtool /usr/local/bin/

sudo mv consoles /etc/prometheus

sudo mv console\_libraries /etc/prometheus

sudo mv prometheus.yml /etc/prometheus

sudo chown -R prometheus:prometheus /etc/prometheus

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

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

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

**2.5 Create Prometheus Systemd Service**

sudo nano /etc/systemd/system/prometheus.service

**Add the following content:**

[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/

[Install]

WantedBy=multi-user.target

**2.6 Start Prometheus**

sudo systemctl daemon-reload

sudo systemctl start prometheus

sudo systemctl enable prometheus

**3. Install Grafana**

**3.1 Add Grafana APT Repository**

sudo apt install -y software-properties-common

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

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

sudo apt update

**3.2 Install Grafana**

sudo apt install -y grafana

**3.3 Start Grafana**

sudo systemctl start grafana-server

sudo systemctl enable grafana-server

**4. Basic Usage Examples**

**4.1 Prometheus**

* **Access Prometheus:** <http://localhost:9090>
* **Test Query:**  
  up  
  This will display the status of monitored targets.

**4.2 Grafana**

* **Access Grafana:** <http://localhost:3000>
* **Default Login:**
  + Username: admin
  + Password: admin
* **Add Prometheus Data Source:**
  1. Navigate to *Configuration > Data Sources*.
  2. Choose *Prometheus*.
  3. Set URL to http://localhost:9090 and save.
* **Create a Dashboard:**
  1. From your home page, go to *Create > Dashboard*.
  2. Add a new panel and use the query up to visualize Prometheus target status.

**5. Monitoring Capabilities and Visualizations**

**5.1 What Prometheus Can Monitor**

* **System Metrics:** CPU, memory, disk, and network usage
* **Application Metrics:** Web server stats, database performance
* **Docker/Kubernetes:** Container performance, pod health
* **Custom Metrics:** Custom business or service metrics via exporters

**5.2 Grafana Visualizations**

* **Graph Panels:** Line, bar, and area charts for trends over time
* **Gauge Panels:** Real-time single-value metrics (e.g., CPU usage %)
* **Heatmaps:** Visualizing high-density data (e.g., request latencies)
* **Dashboards:** Customizable multi-panel views for comprehensive monitoring

**6. Directory Overview**

* **Prometheus Config:** /etc/prometheus/prometheus.yml
* **Prometheus Data:** /var/lib/prometheus
* **Grafana Config:** /etc/grafana
* **Grafana Data:** /var/lib/grafana

**7. Firewall Configuration (Optional)**

sudo ufw allow 9090/tcp Prometheus

sudo ufw allow 3000/tcp Grafana

sudo ufw reload

**Setup Complete!**

Prometheus and Grafana are now running and ready for monitoring and visualization.

**Quering Prometheus from Grafana**

Get the current CPU usage rate on a system, excluding idle mode

**sum(rate(node\_cpu\_seconds\_total{mode!="idle"}[5m])) by (instance)**

Check the total memory usage

**node\_memory\_MemTotal\_bytes - node\_memory\_MemAvailable\_bytes**

Determine disk space usage by subtracting free from total space

**node\_filesystem\_size\_bytes - node\_filesystem\_free\_bytes**

Display which instances are currently up and running

up

Measure network traffic received over a specific network interface

**rate(node\_network\_receive\_bytes\_total[5m])**

Query the rate of HTTP requests over the last 5 minutes

**rate(http\_requests\_total[5m])**

----

Here is a detailed breakdown of the considerations mentioned:

Ensure that your Prometheus setup collects metrics for these queries:\*\*

- When you set up Prometheus, it scrapes data from various endpoints. The specific metrics available depend on the targets specified in your Prometheus configuration.

- Your configuration file (‘prometheus.yml’) should list all exporters and targets that provide the data you want to monitor. Each scraper target needs to correctly expose the ‘/metrics’ endpoint.

- Verify in the Prometheus user interface (typically accessible at `http://localhost:9090`) that your targets are monitored and their metrics are collected successfully.

If you are using custom exporters, verify that the metric names correspond to those exporters:

- Custom exporters may not use the default metric names that standard exporters or node/system-level exporters use.

- For example, if you're using a specialized application exporter, ensure that the metrics it provides align with what you expect. Check the documentation provided with the exporter for precise metric names.

- In your queries, make sure you're referencing the exact names of the metrics as defined by those exporters. This might involve inspecting a sample of data provided by the ‘/metrics’ endpoint.

Adjust based on your specific monitoring environment and needs:

- You may need to tailor queries to suit the particular architecture or requirements of your environment. This may involve changing aggregation functions, modifying time ranges, or adapting label filters.

- Evaluate the performance characteristics or resource limitations of your environment and adjust your monitoring frequencies and alerting thresholds accordingly.

- Extend your setup to include alerts and dashboards that address specific operational or business requirements, ensuring they are actionable and relevant to your usage context.

By carefully configuring your Prometheus setup to mirror the requirements and structure of your infrastructure and monitoring objectives, you can ensure robust and relevant metrics collection.

----

Query crafting

Step 1: Access the Prometheus UI

Open a web browser and navigate to the Prometheus web interface, typically available at http://localhost:9090

Step 2: Use the 'Graph' Tab

In the Prometheus web UI, click on the "Graph" tab to access the query interface. This tab allows you to type queries and visualize results graphically or as console output.

Step 3: Craft Your Query

In the query box, type your Prometheus Query Language (PromQL) query.

For example, to see the CPU usage rate:

sum(rate(node\_cpu\_seconds\_total{mode!="idle"}[5m])) by (instance)

Make sure your query reflects the metrics available from your configured targets.

Use the 'Autocomplete' feature, when available in the interface, to help guide your query creation.

Step 4: Run the Query

After entering your query, click the “Execute” button.

This will process the query and update the graph or output below based on the data your Prometheus server has collected.

Step 5: Iterate and Modify

Based on the results, refine or adjust your query.

PromQL supports functions, aggregations, and various selectors that can help tailor the results to your needs.

Step 6: Save and Reuse

For repeated use, document or store your queries in a handy location.

Consider integrating them with a dashboard solution like Grafana for enhanced visualization and analysis.

Additional Consideration

If querying specific metrics provided by non-standard exporters, review the metric name and label conventions those exporters use.