Prometheus Installation & Setup (Linux)

1. Create Prometheus User

sudo useradd --no-create-home --shell /bin/false prometheus
sudo mkdir -p /etc/prometheus /var/lib/prometheus

2. Set Permissions

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

3. Download Prometheus Package

Visit the Prometheus official downloads page: https://prometheus.io/download/

cd /tmp
wget
https://github.com/prometheus/prometheus/releases/download/v3.5.0/prometheus-3.5.0.linux-amd64.tar.gz

4. Extract Package

tar -xvf prometheus-3.5.0.linux-amd64.tar.gz

5. Move Binaries & Config

sudo mv prometheus promtool /usr/local/bin/
sudo mv prometheus.yml /etc/prometheus/

6. Adjust Permissions

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

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

7. Add Prometheus Targets

sudo nano /etc/prometheus/prometheus.yml

Add your targets inside the scrape_configs section.

8. Create Systemd Service

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

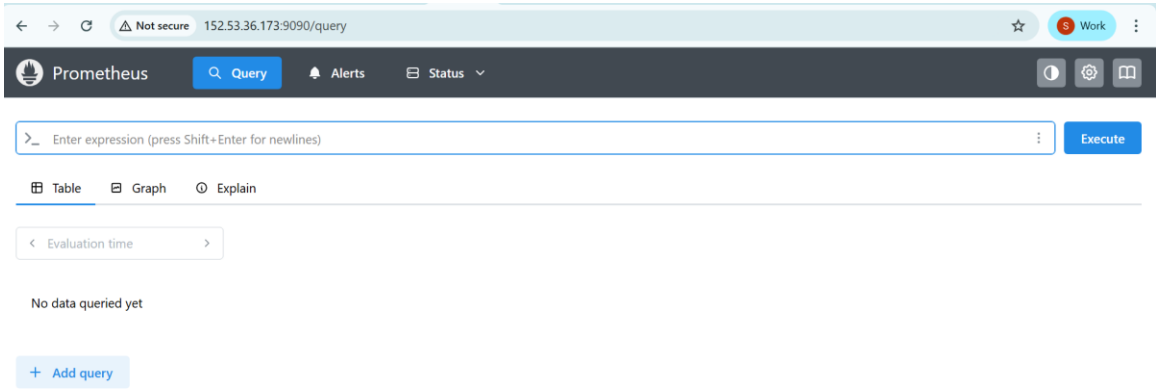
Paste the following content:

```
[Unit]
Description=Prometheus Monitoring System
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  /
web.listen.address="0.0.0.0:9090"

[Install]
WantedBy=multi-user.target
```

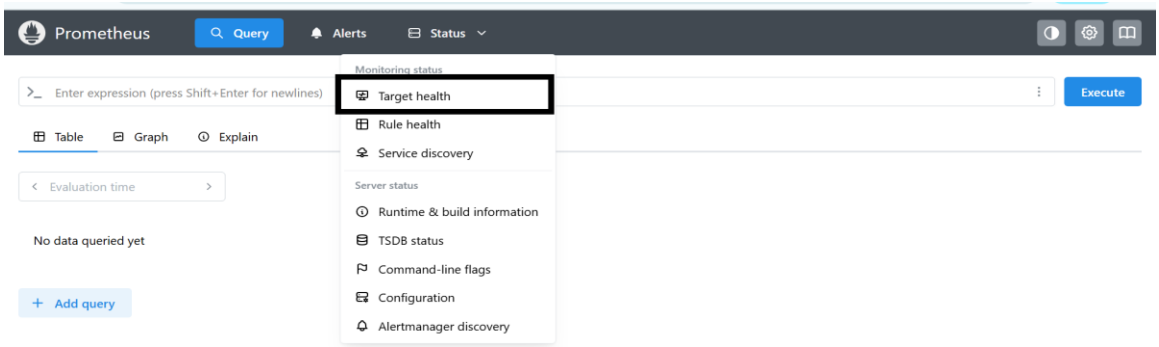
9. Reload Daemon & Start Service

sudo systemctl daemon-reload
sudo systemctl enable --now prometheus

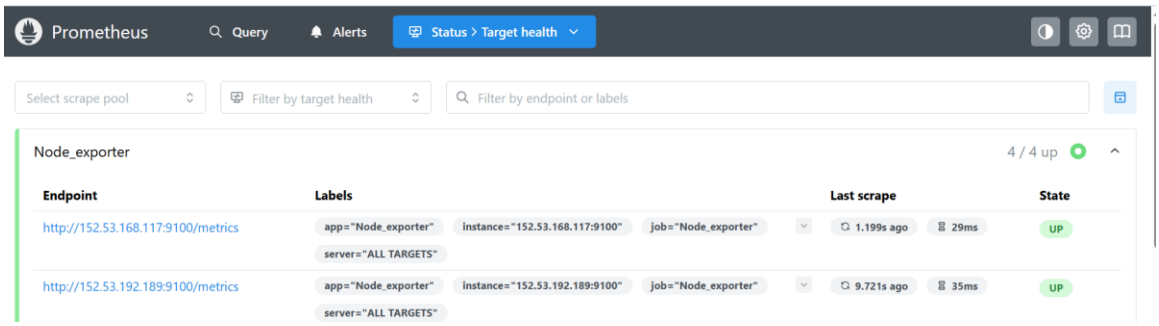
Prometheus will now be running and accessible at: http://<server-ip>:9090



Click on the Target Health Link inside status:



You will see all available targets:



Node Exporter Setup (Linux)

1. Create Prometheus User

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

2. Create Directory and change permission

sudo mkdir -p /etc/node_exporter
sudo chown -R node_exporter:node_exporter /etc/node_exporter

3. Download the node exporter

Goto Official Prometheus Website: https://prometheus.io/download/
cd /tmp
wget
https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz

4. Extract the package

tar -xvf node_exporter-1.9.1.linux-amd64.tar.gz

5. Move node exporter to the binary folder and change permission

sudo mv node_exporter /usr/local/bin/
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter

6. Create the service

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

[Unit]
Description=Node Exporter for Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter --web.listen-address="0.0.0.0:9100"

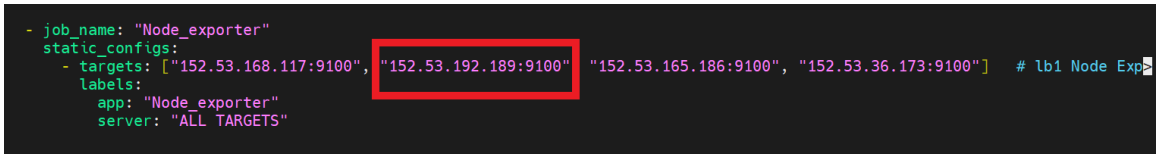
[Install]
WantedBy=multi-user.target

8. Start the service

sudo systemctl daemon-reload
sudo systemctl enable --now node_exporter
sudo systemctl status node_exporter

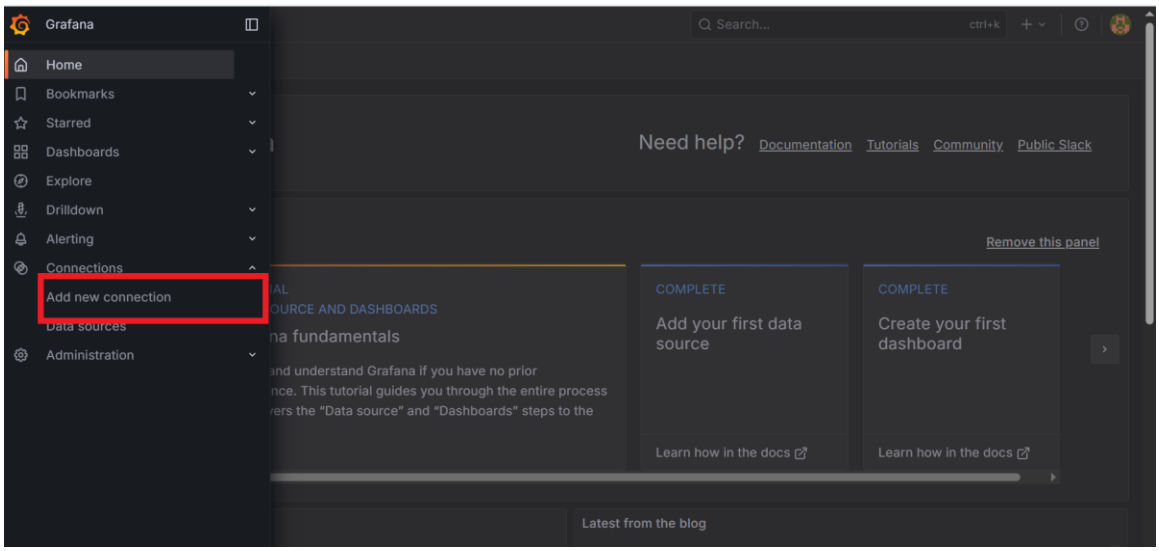
9. Add Node Exporter in prometheus.yml

sudo nano /etc/prometheus/prometheus.yml

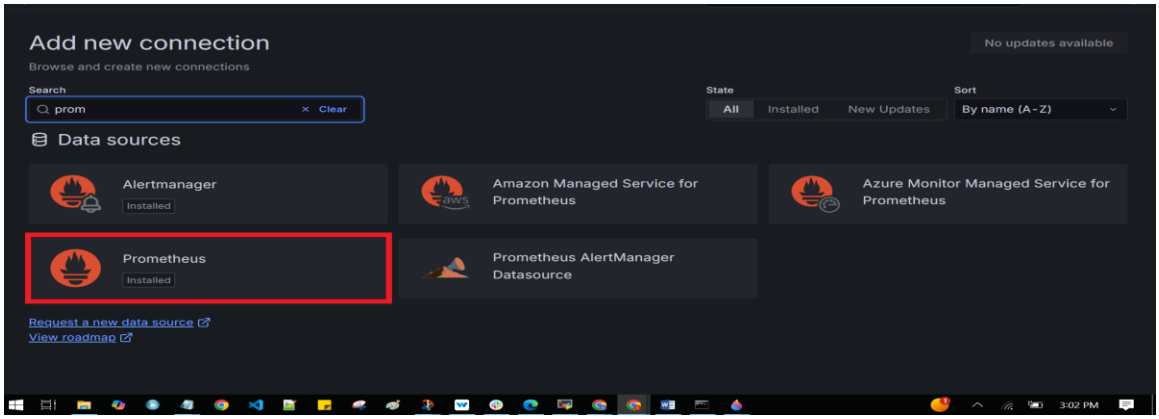


10. Create Dashboard

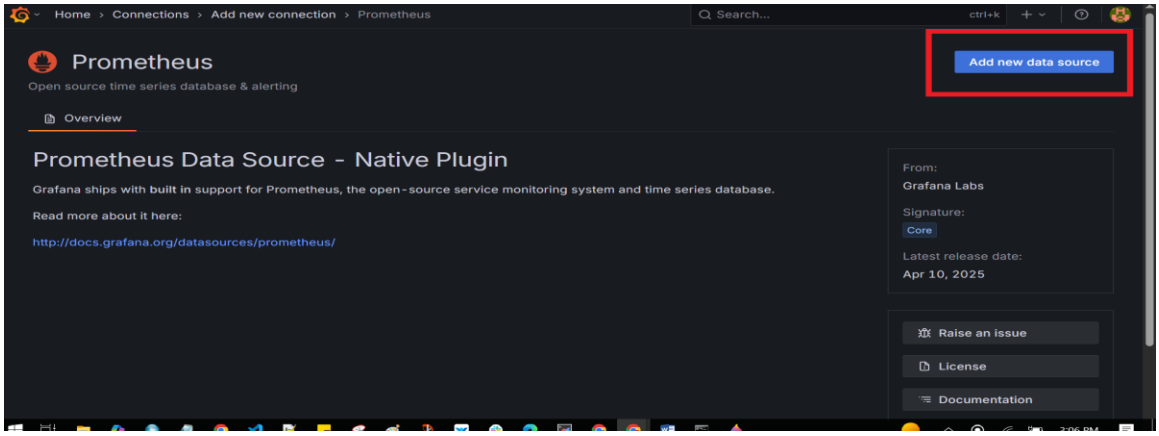
Go to Add new connection tap inside the Connections Downbar:



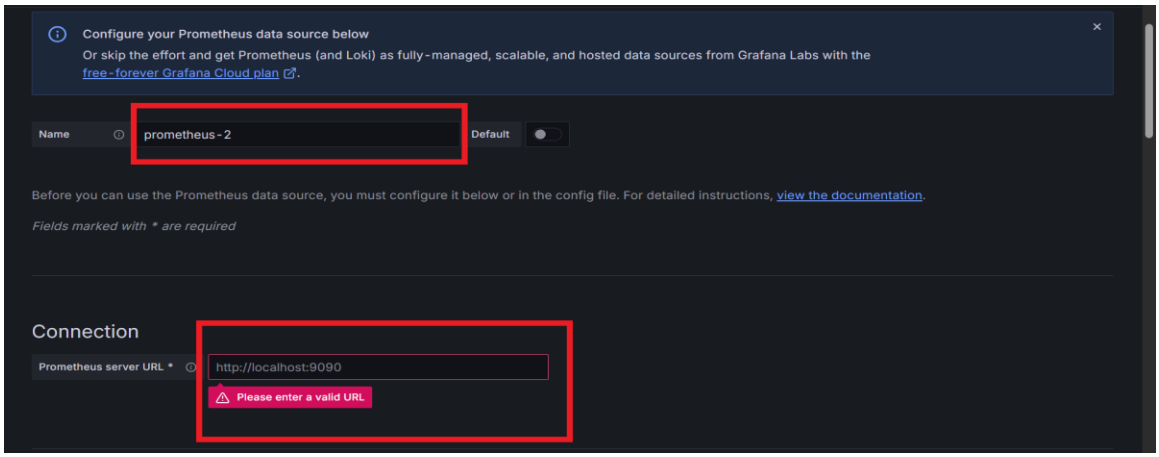
Select Prometheus:



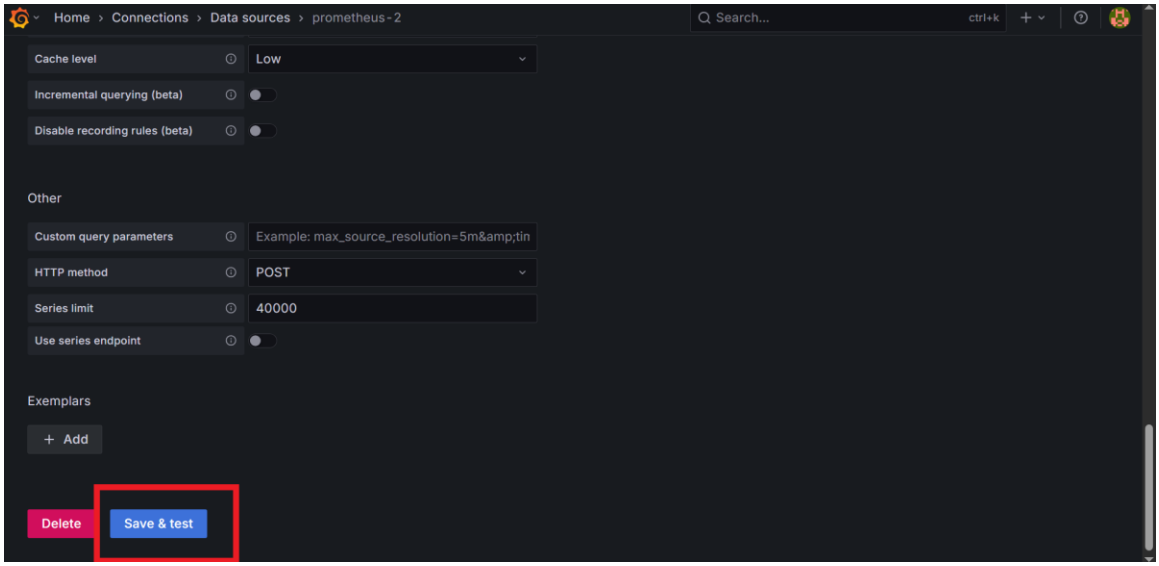
Select Add new data source:



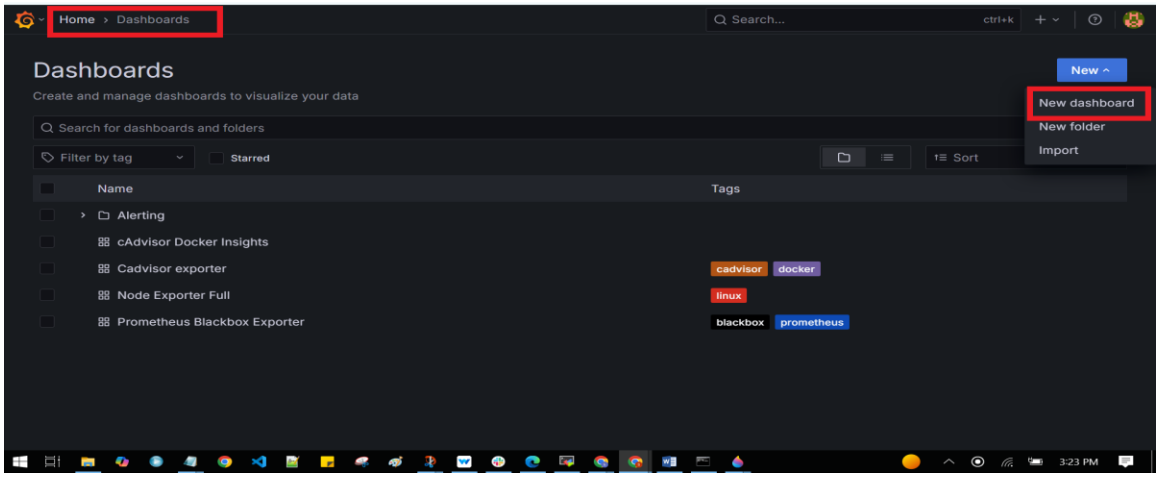
Add Name and the URL of the Prometheus:



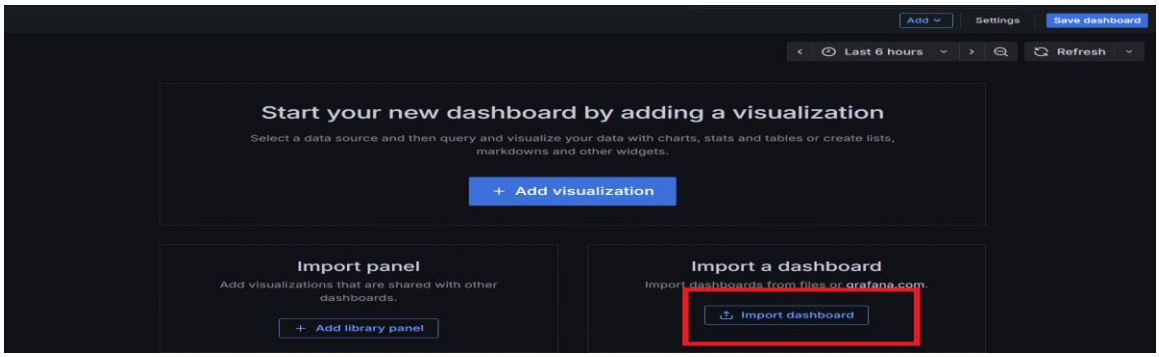
Save and Test the datasource:



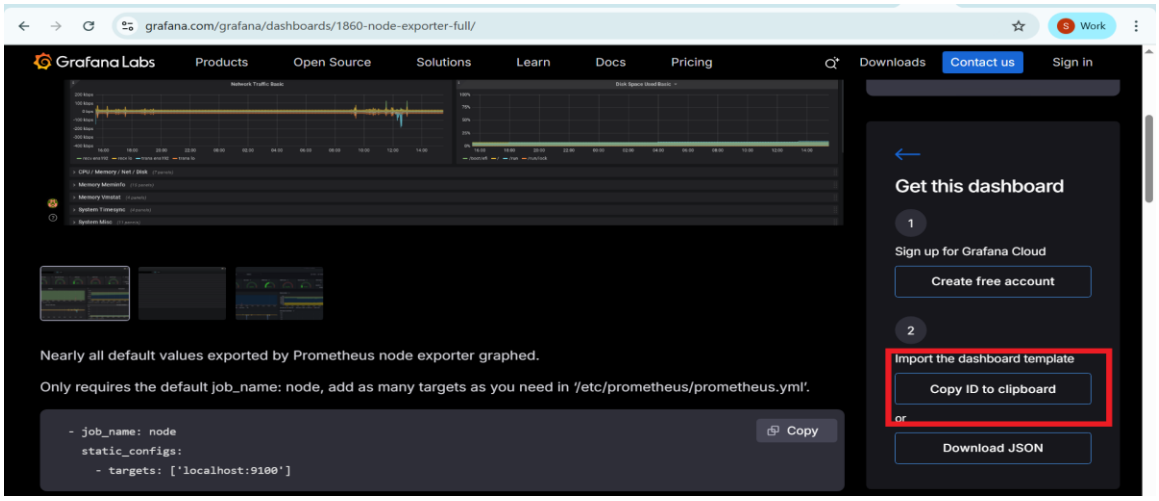
Inside the Dashboard Folder select New Dashboard:



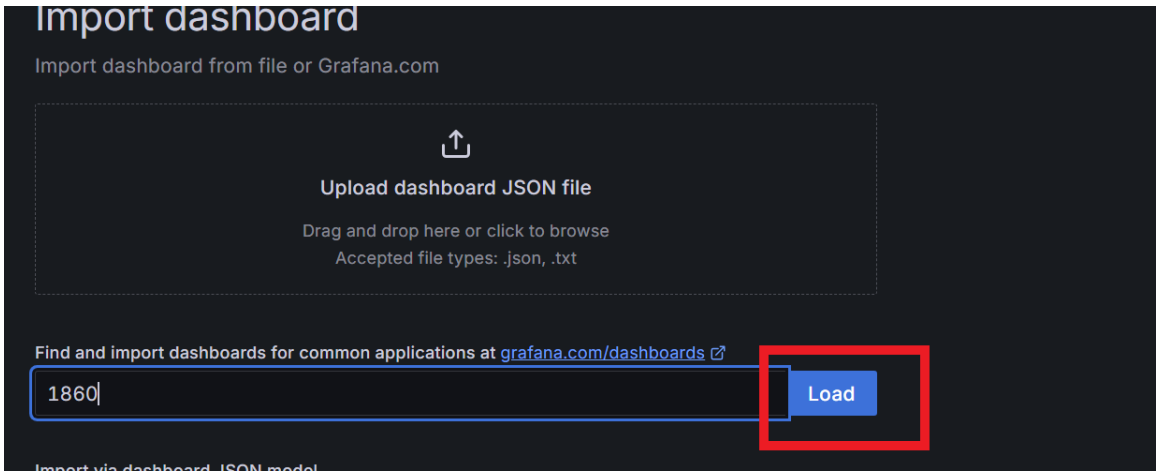
Select Import Dashbord:



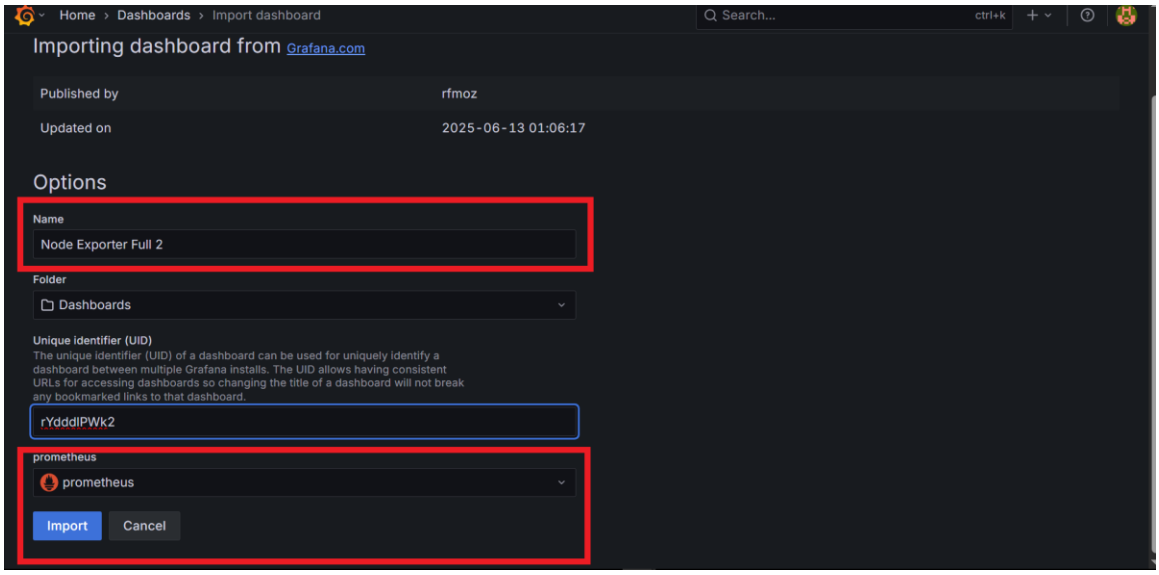
Copy Dashboard ID:



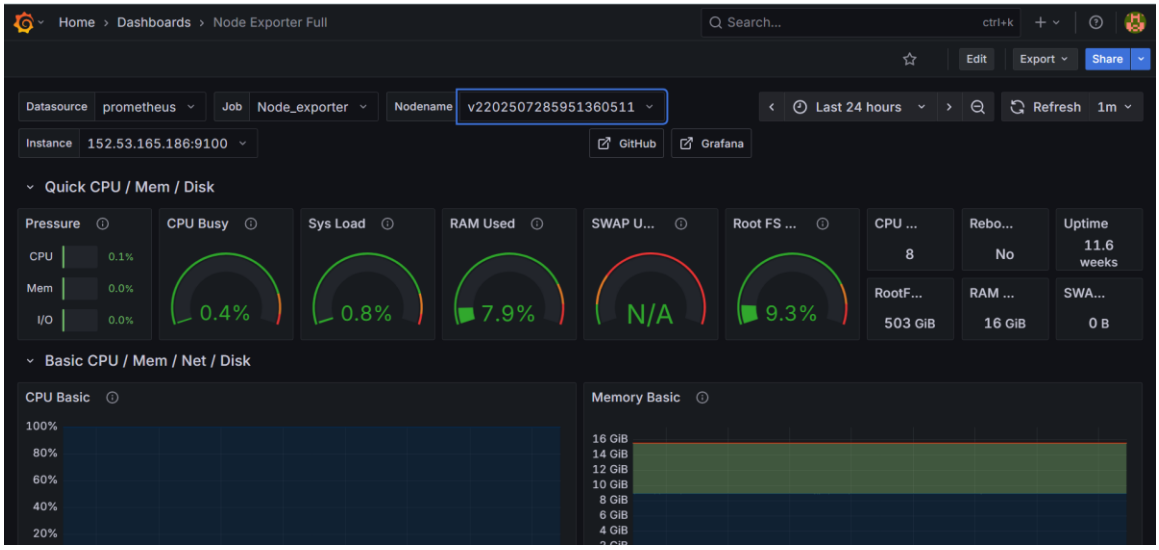
Add the ID and Press the Load button:



Add the Dashboard and the Prometheus Exporter:



Check the dashboard:



BlackBox Exporter Setup (Linux)

1. Create BlackBox User

sudo useradd --no-create-home blackbox

2. Create Directory

sudo mkdir /etc/blackbox

3. Download the node exporter

Goto Official Prometheus Website: https://prometheus.io/download/
cd /tmp
wget
https://github.com/prometheus/blackbox_exporter/releases/download/v0.27.0/blackbox_exporter-0.27.0.linux-amd64.tar.gz

5. Extract the package

tar -xvf blackbox_exporter-0.27.0.linux-amd64.tar.gz

6. Move node exporter to the binary folder and change permission

sudo chown blackbox:blackbox /usr/local/bin/blackbox_exporter

sudo chown -R blackbox:blackbox /etc/blackbox/*

7. Create the service

sudo vim /etc/systemd/system/blackbox.service

[Unit]

Description=Blackbox

Wants=network-online.target

After=network-online.target

[Service]

User=blackbox

Group=blackbox

Type=simple

ExecStart=/usr/local/bin/blackbox_exporter --config.file=/etc/blackbox/blackbox.yml --web.listen-address="0.0.0.0:9115"

[Install]

WantedBy=multi-user.target

8. Start the service

sudo systemctl daemon-reloads

sudo systemctl enable blackbox

sudo systemctl start blackbox

sudo systemctl status blackbox

9. Add Modulus setting in blackbox

sudo vim /etc/blackbox/blackbox.yml

http_prometheus:

  prober: http

  timeout: 5s

  http:

   method: GET

   valid_http_versions: ["HTTP/1.1", "HTTP/2"]

```
    fail_if_ssl: false
    fail_if_not_ssl: false
```

10. Edit Prometheus Setting

```
sudo nano /etc/prometheus/prometheus.yml
```

```
- job_name: "blackbox"
  metrics_path: /probe
  params:
    module: [http_prometheus]
  static_configs:
    - targets:
      - https://reflys.com
      - https://api.reflys.com
      - https://operator.reflys.com
      - https://cp.reflys.com
      labels:
        app: "uptime"
        probe_type: "https"

  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target

    - source_labels: [__param_target]
```

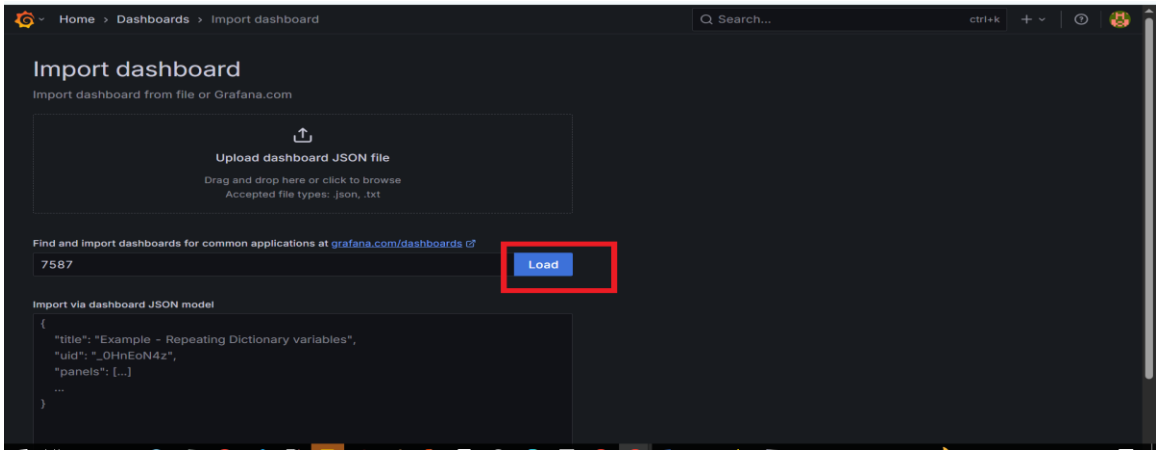
target_label: instance

- target_label: __address__

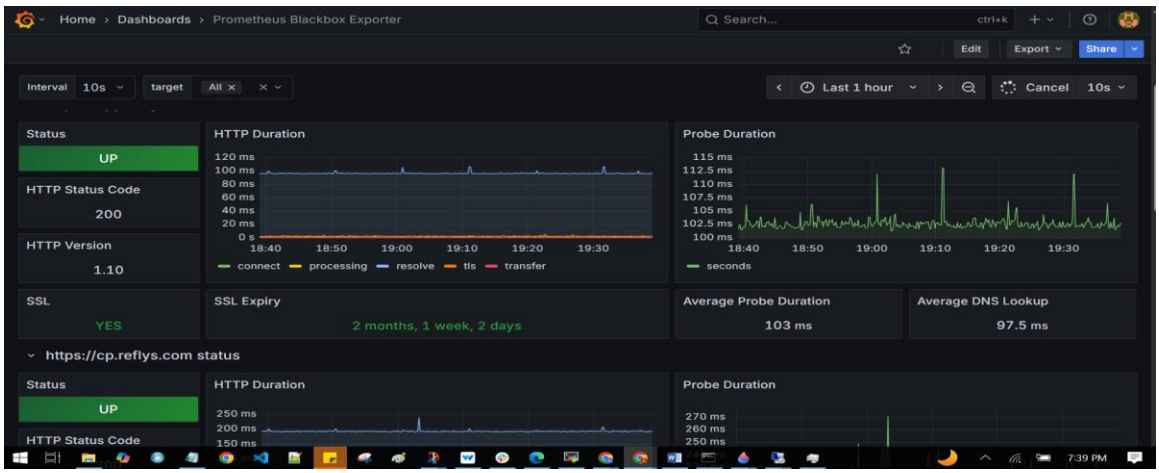
replacement: 127.0.0.1:9115

11. Create Dashboard

Copy the Dashboard Id from Grafana and paste it here and then click load:



Check the Dashboard:



Cadvisor Intallation Guide (Docker)

1. Run Docker Command:

sudo docker run \

--volume=/:/rootfs:ro \

--volume=/var/run:/var/run:rw \

--volume=/sys:/sys:ro \

--volume=/var/lib/docker/:/var/lib/docker:ro \

--publish=8085:8080 \

--restart=unless-stopped \

--detach=true \

--name=cadvisor \

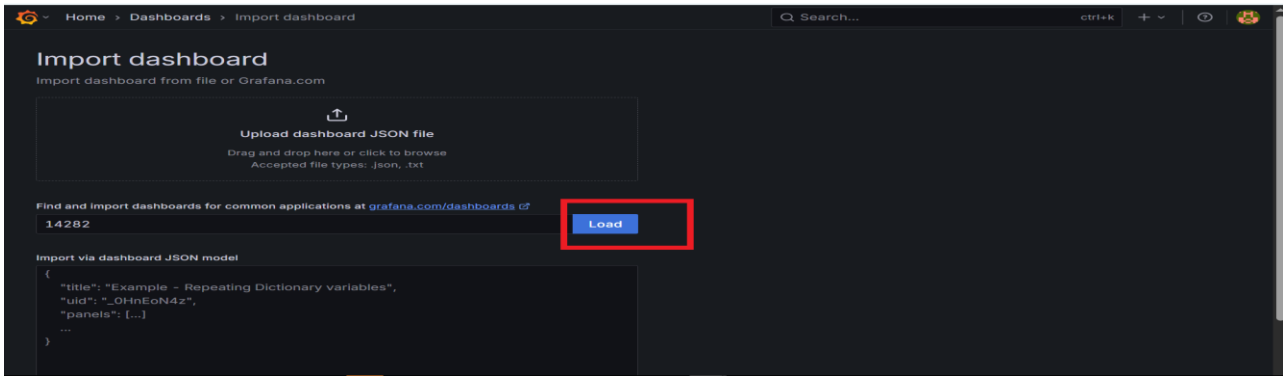
gcr.io/cadvisor/cadvisor:latest

2. Add Prometheus Configuration

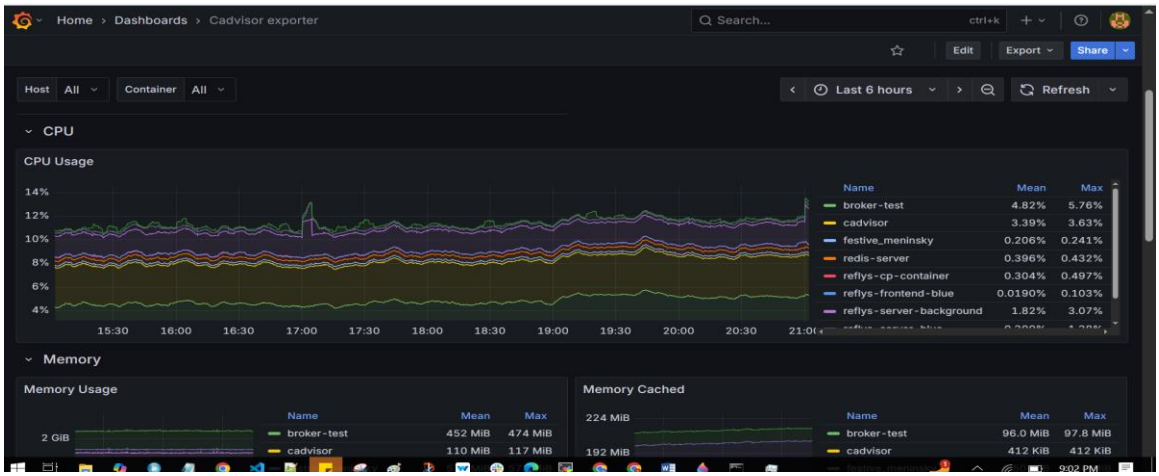
```
- job_name: "cadvisor"
  static_configs:
    - targets: ['152.53.36.173:8085','152.53.165.186:8085', '152.53.192.189:8085']
```

3. Add Docker Desktop:

ADD Dashboard ID:

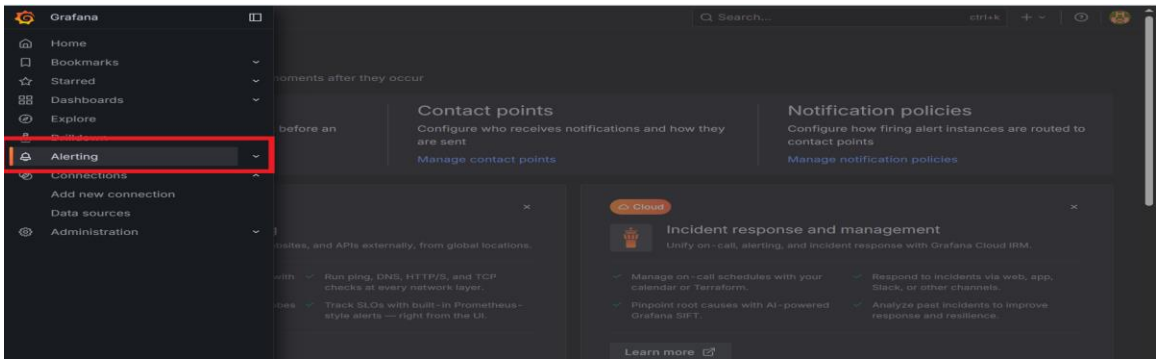


View the dashboard:

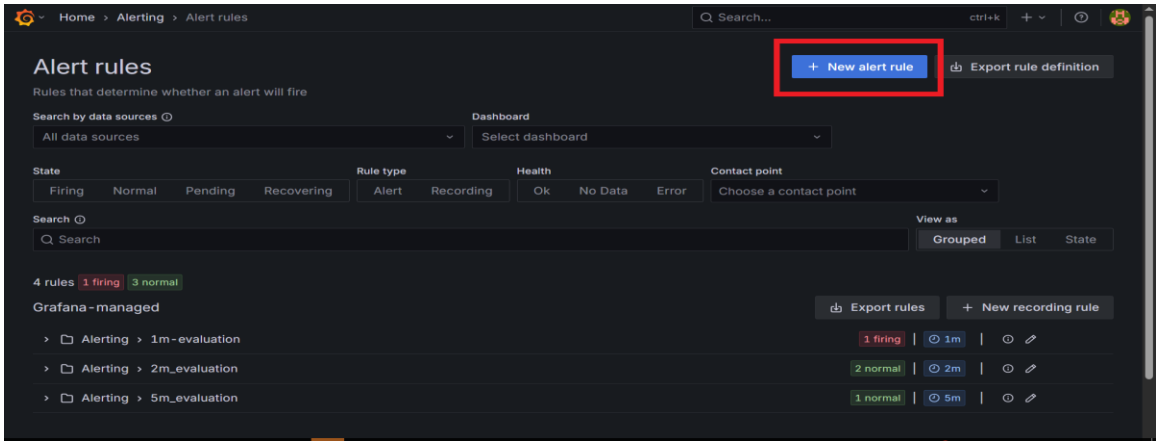


Grafana Alerting:

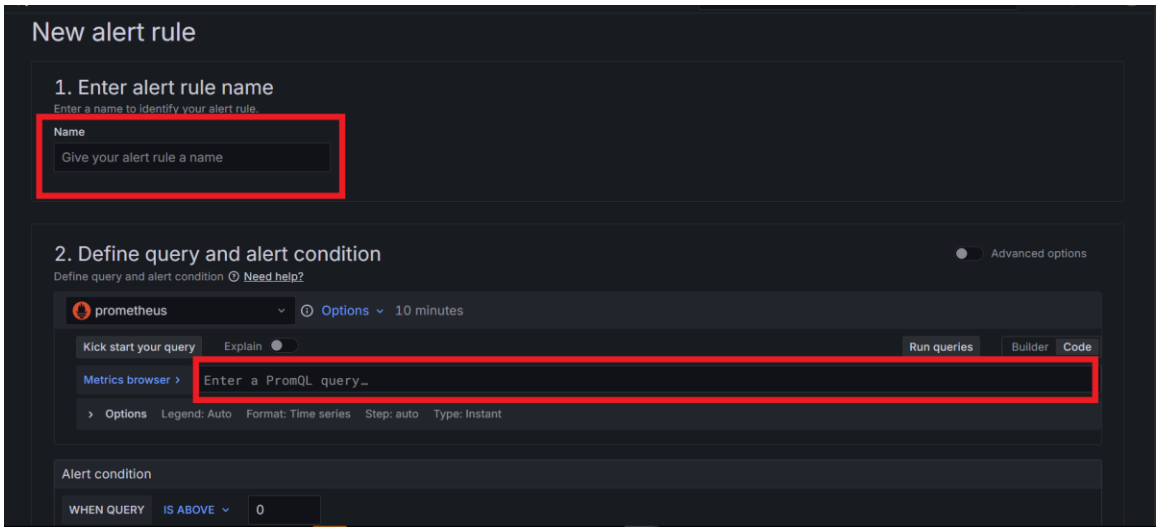
Open the Alerting tab:



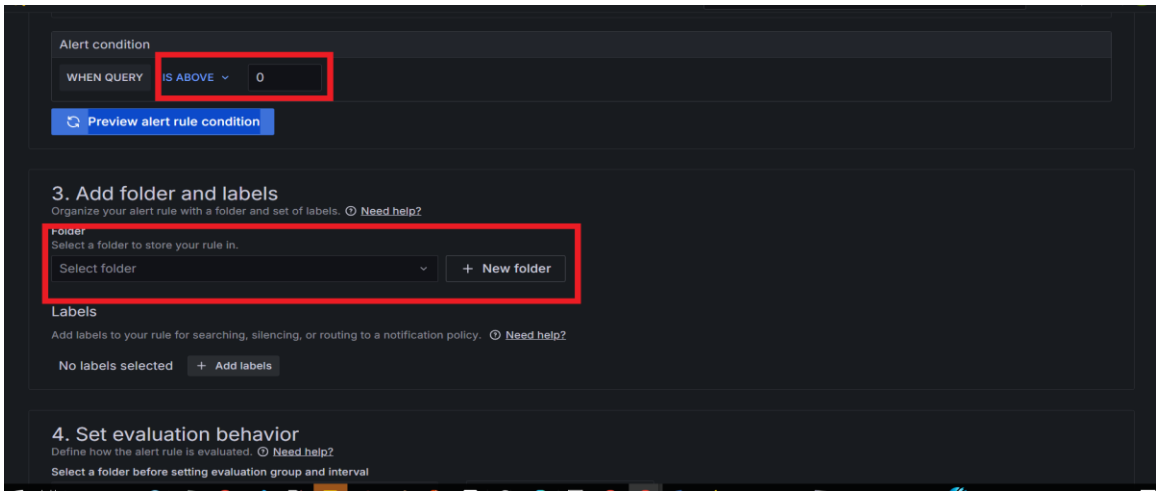
Select the New alert rule:



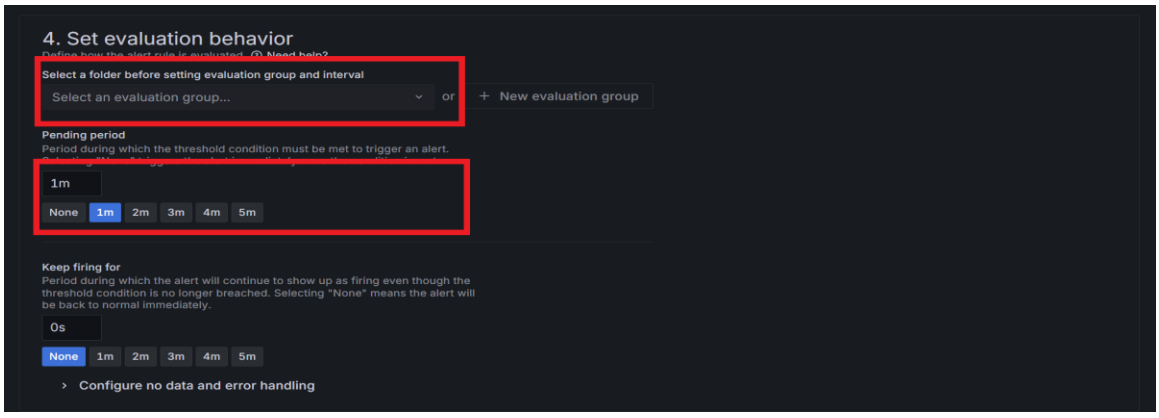
Add alert rule and add PROMQL query:



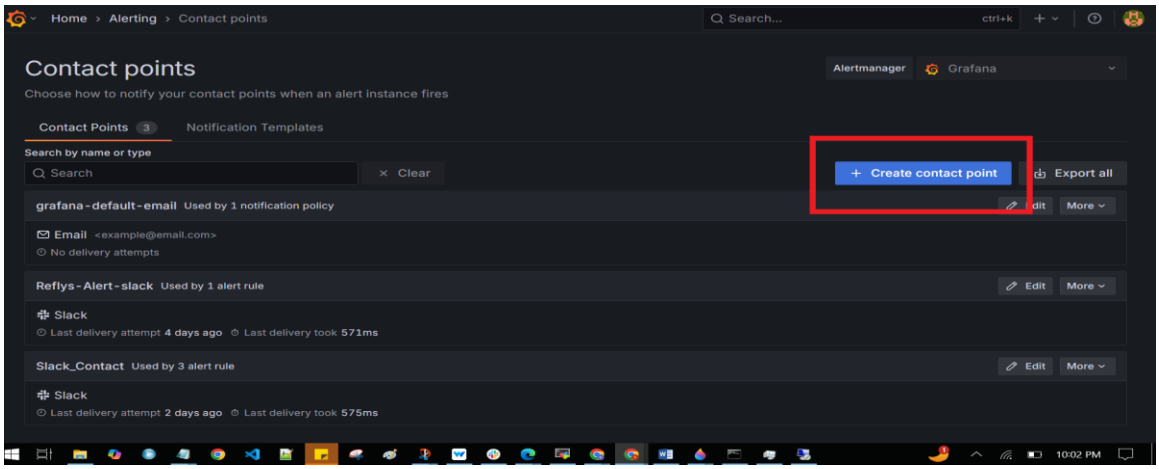
Add the query condition and select folder:



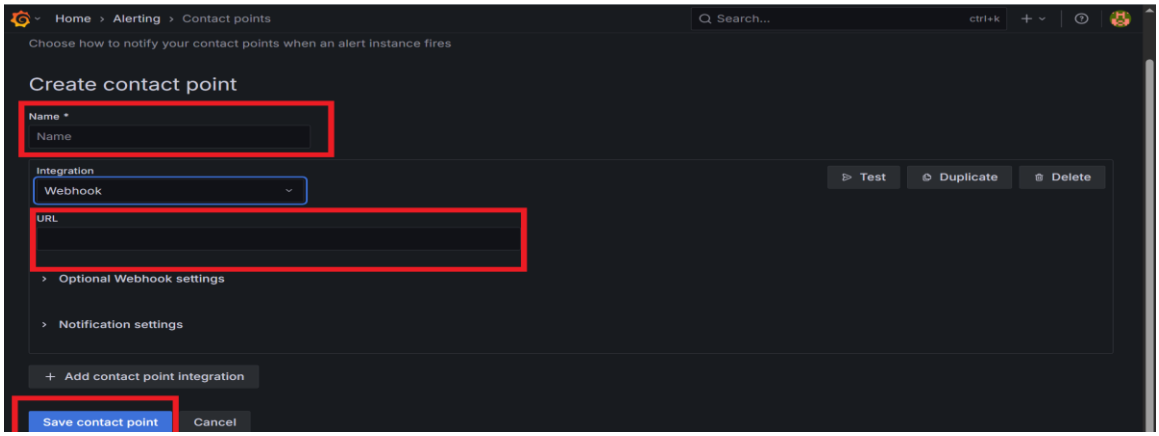
Select Evaluation Period And Pending Period in alert:



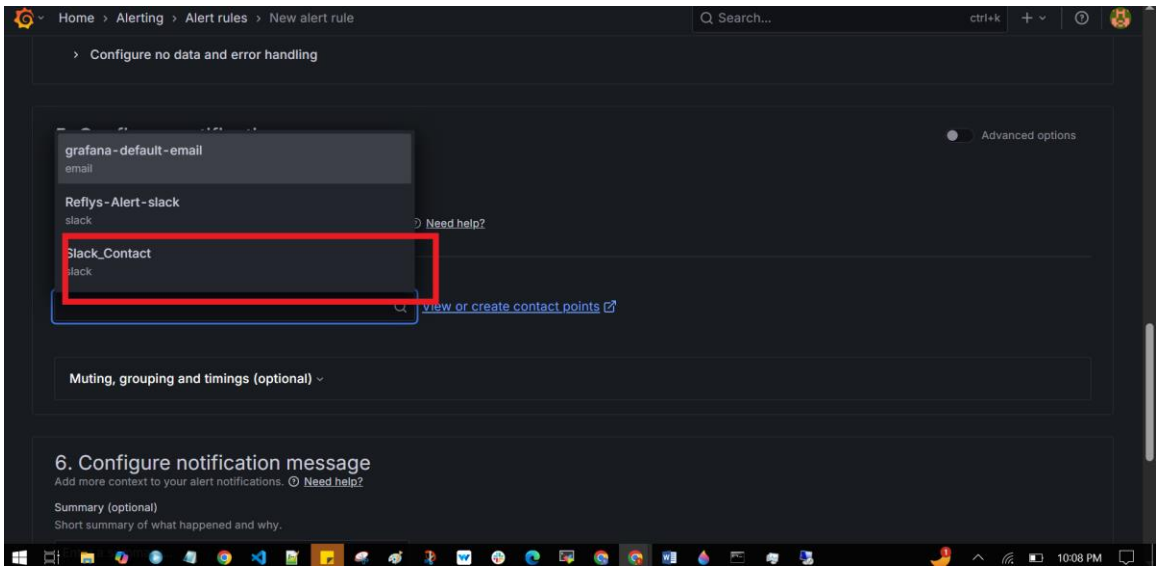
Create New Contact Point:



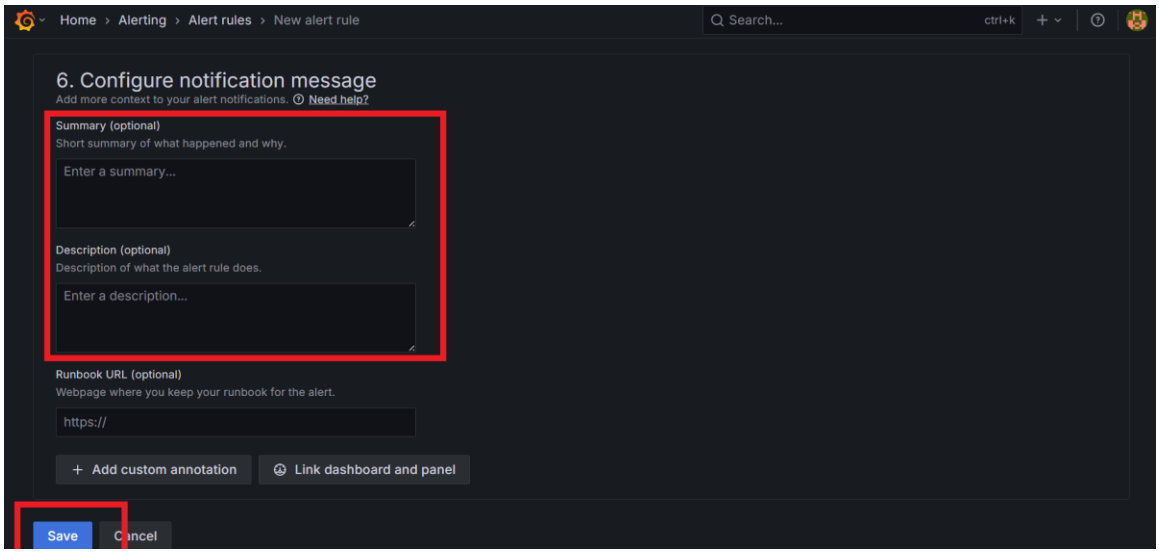
Select Name and slack Webhook Url:



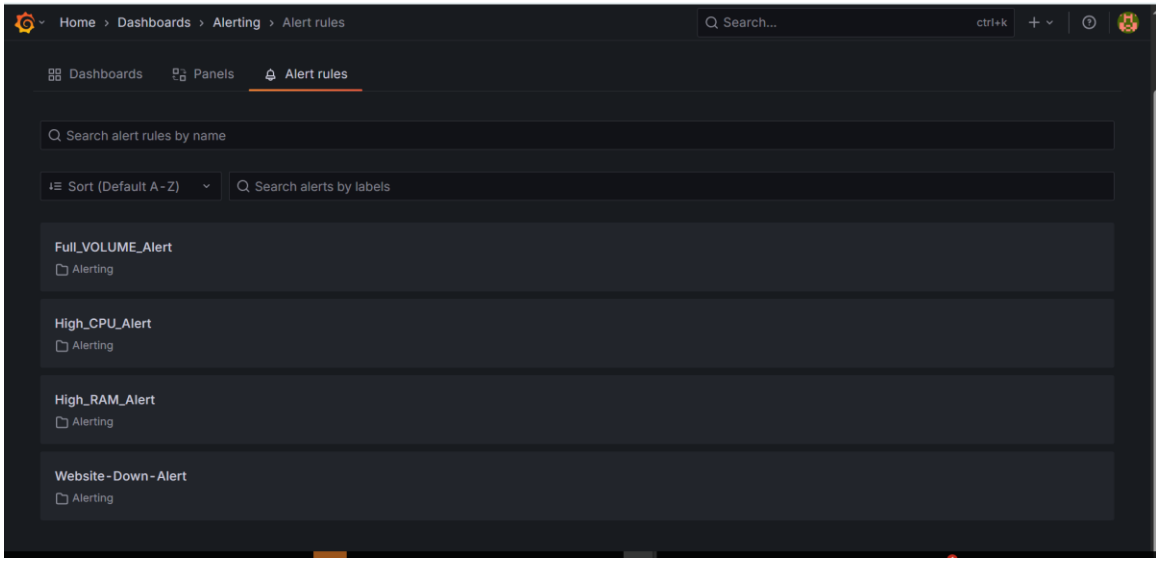
Select the contact point:



Add summary or description and save alert:

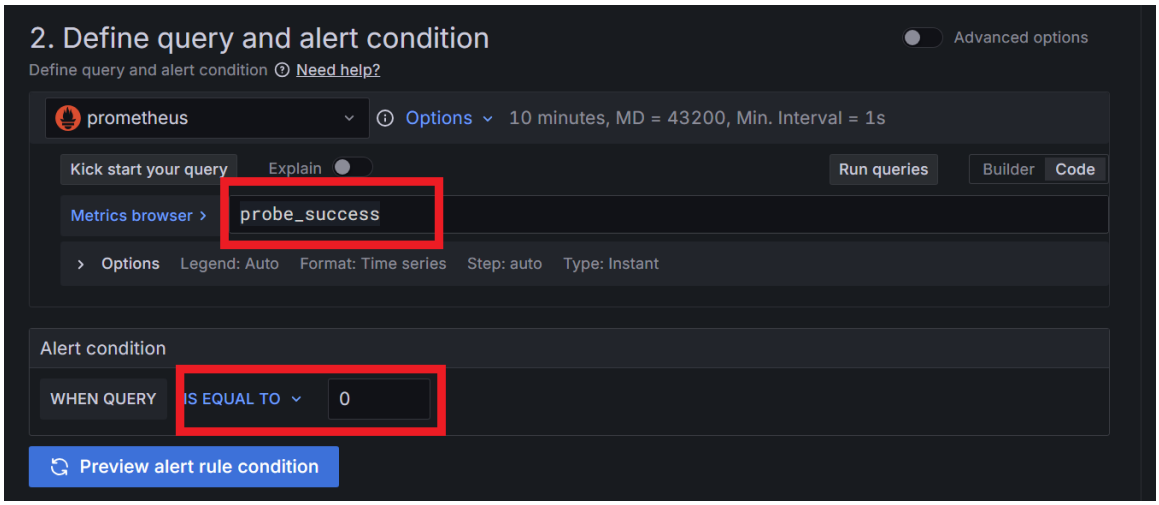


Check all alerts:



Grafana Website Down Alerts Setup:

Use this query and condition in alert:



Press the Preview button:

Add Template:

{{ range .Alerts }}

Alert: {{ .Labels.alertname }}

Status: {{ .Status }}

Website: {{ .Labels.instance }}

{{- if eq .Status "firing" }}

Severity: {{ .Labels.severity }}

{{- else }}

Resolved at: {{ .EndsAt }}

{{- if .Labels.resolution_reason }}

Reason: {{ .Labels.resolution_reason }}

{{- else }}

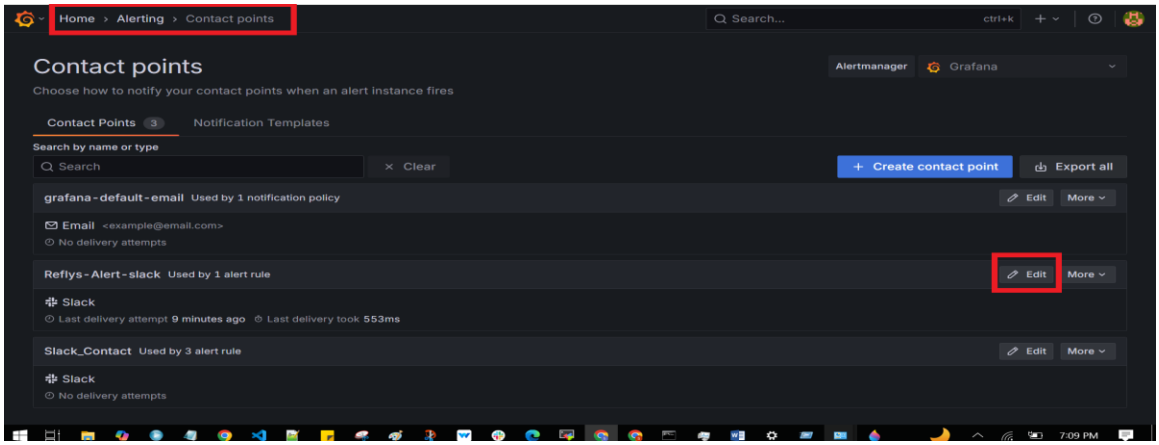
Reason: Alert condition is no longer true.

{{- end }}

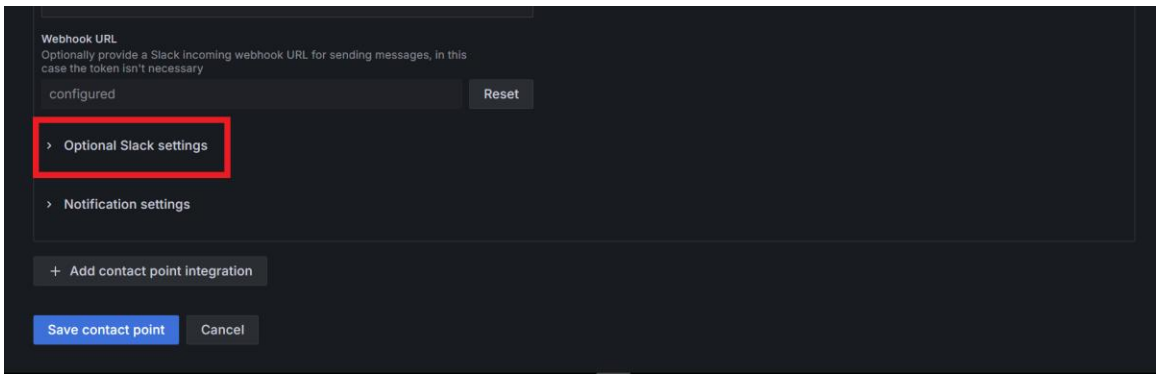
{{- end }}

{{ end }}

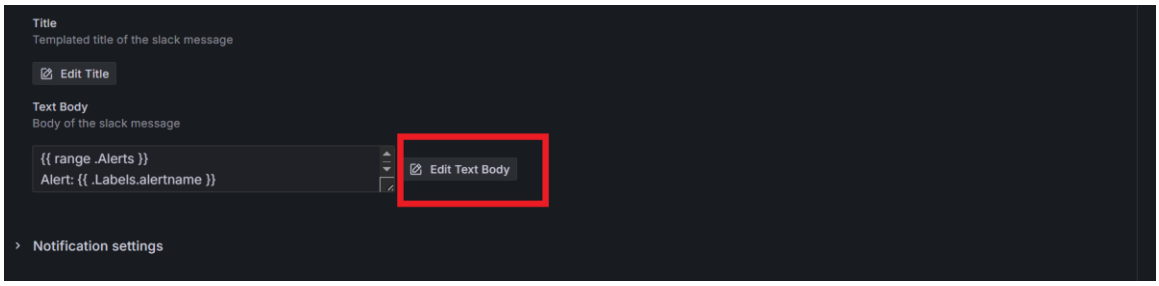
Inside the Contact points(Edit Contact):



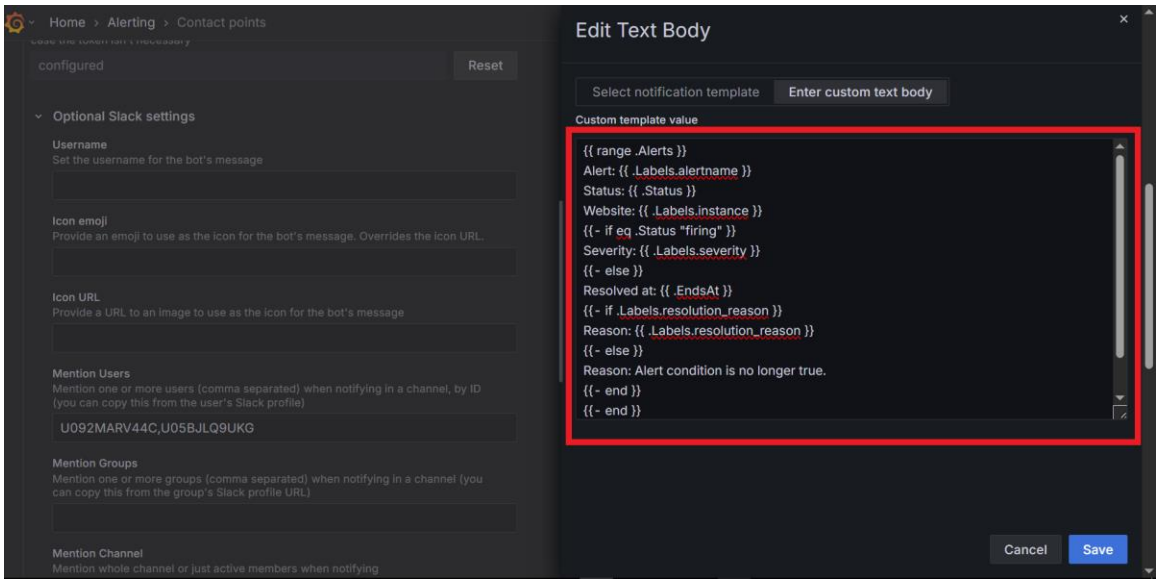
Edit the Optional Slack Settings:



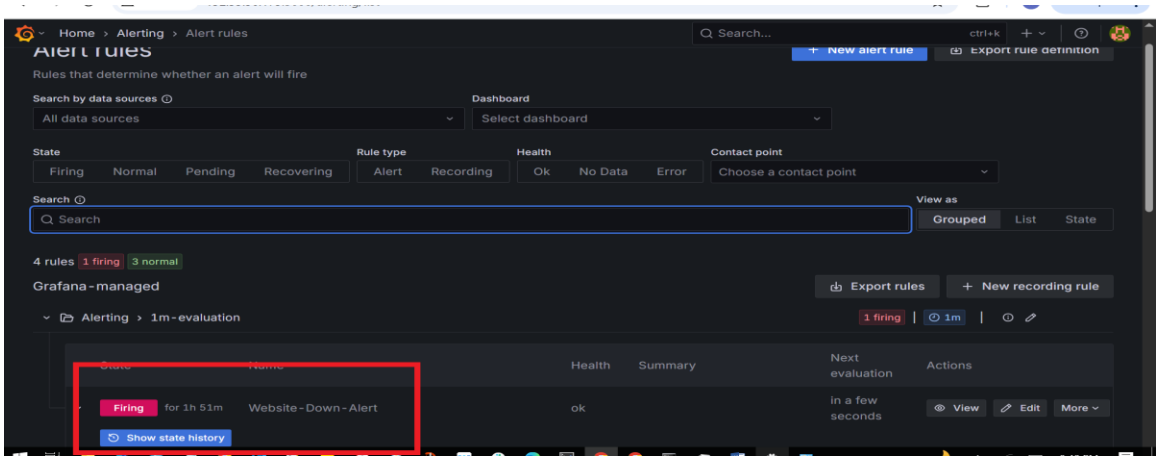
Press the Edit Text Body Button:



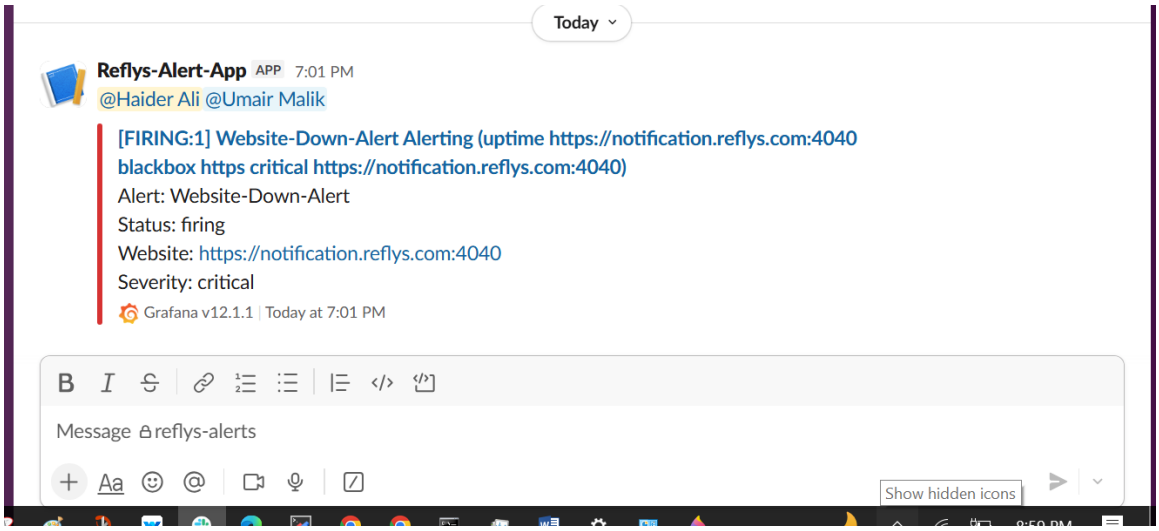
Add the New Syntax in this box:



Check Alert State:

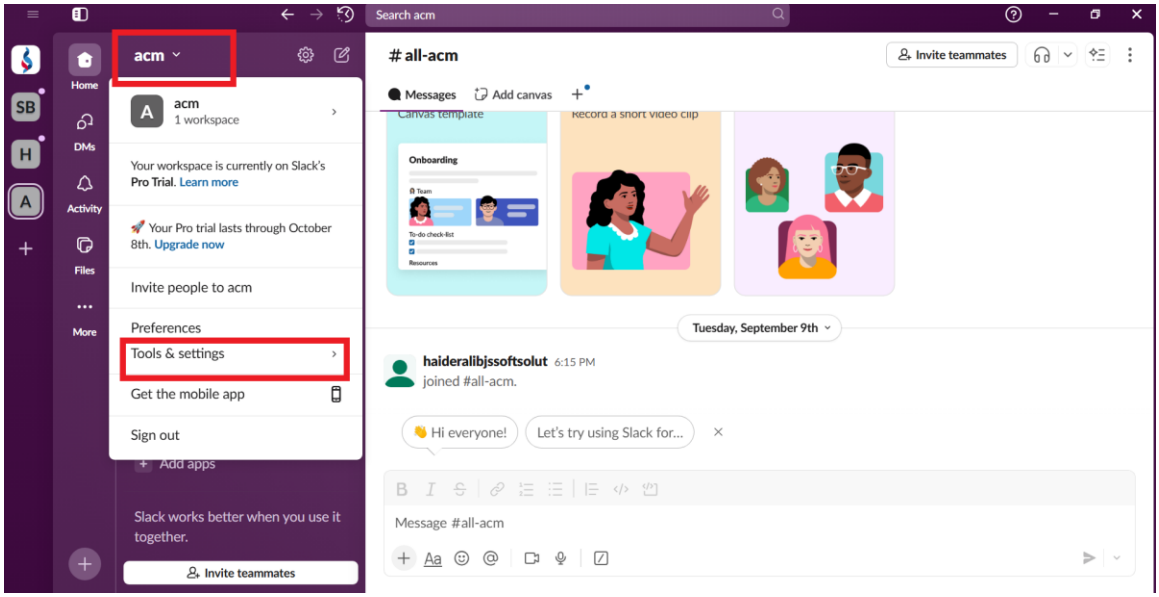


Check Slack Message:

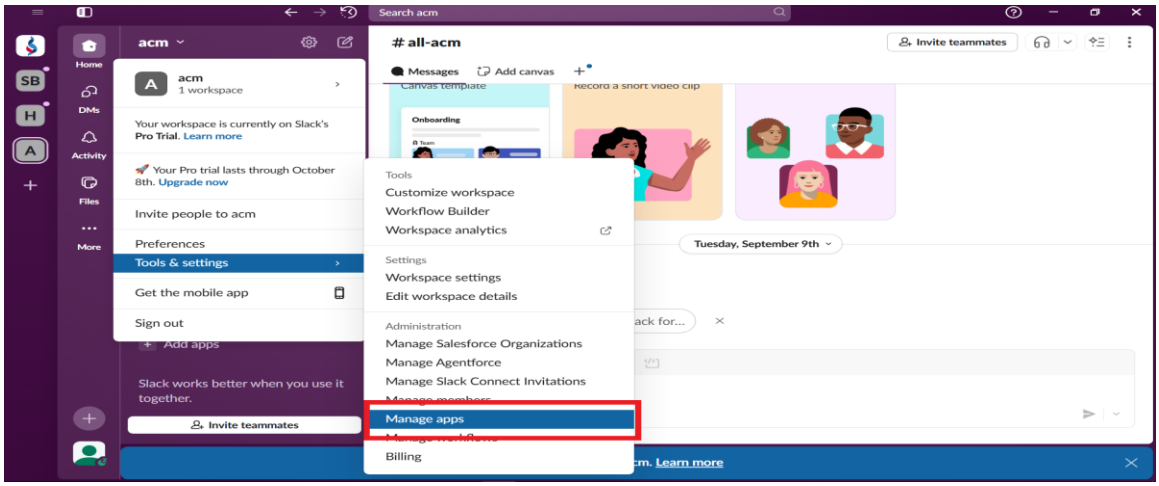


Slack Webhook Creation:

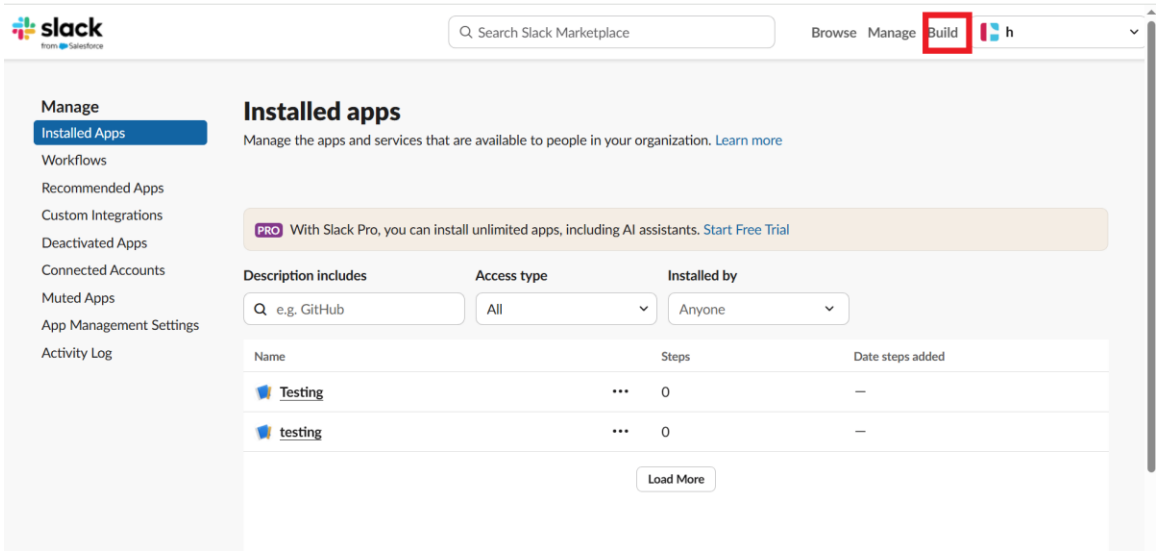
First Select the Workspace Name and then press on **Tools & settings** button:



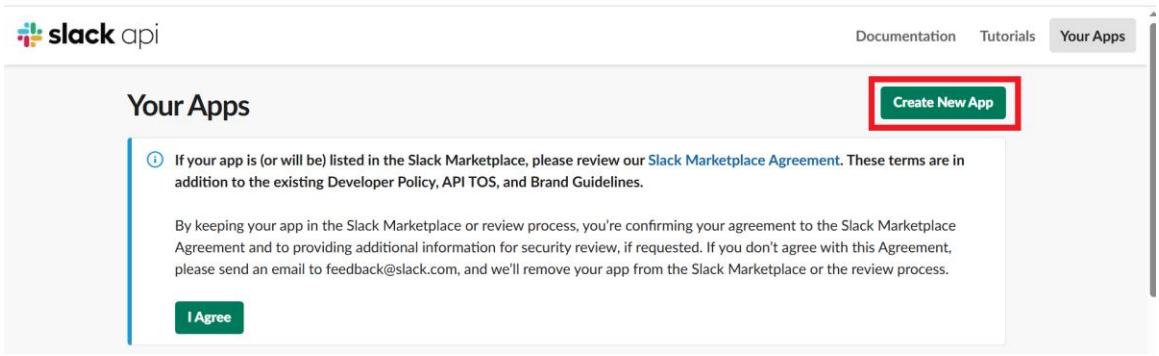
Press the Manage Apps Button:



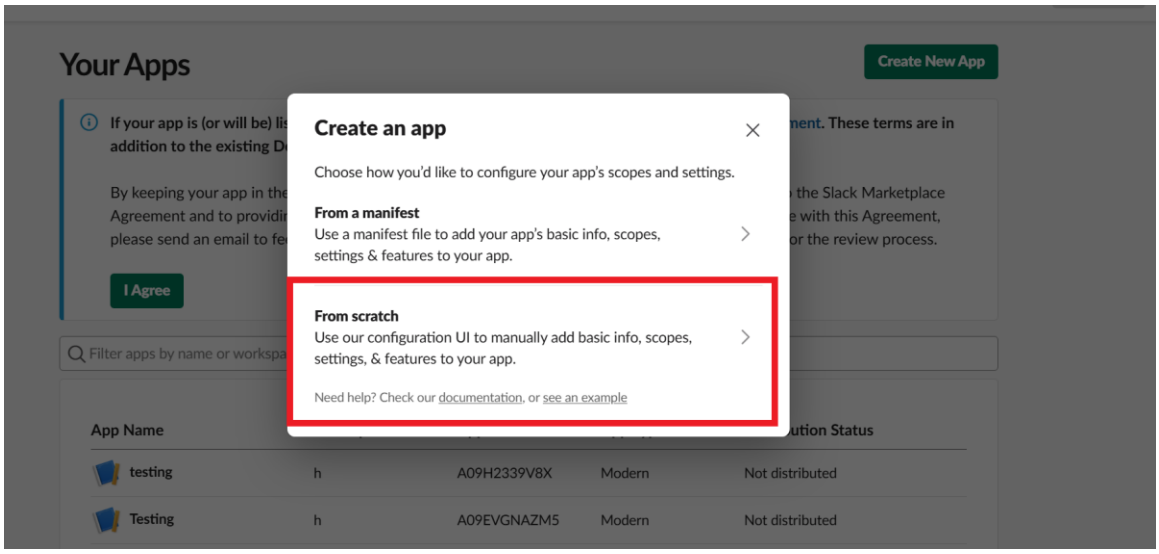
Click on Build Button:



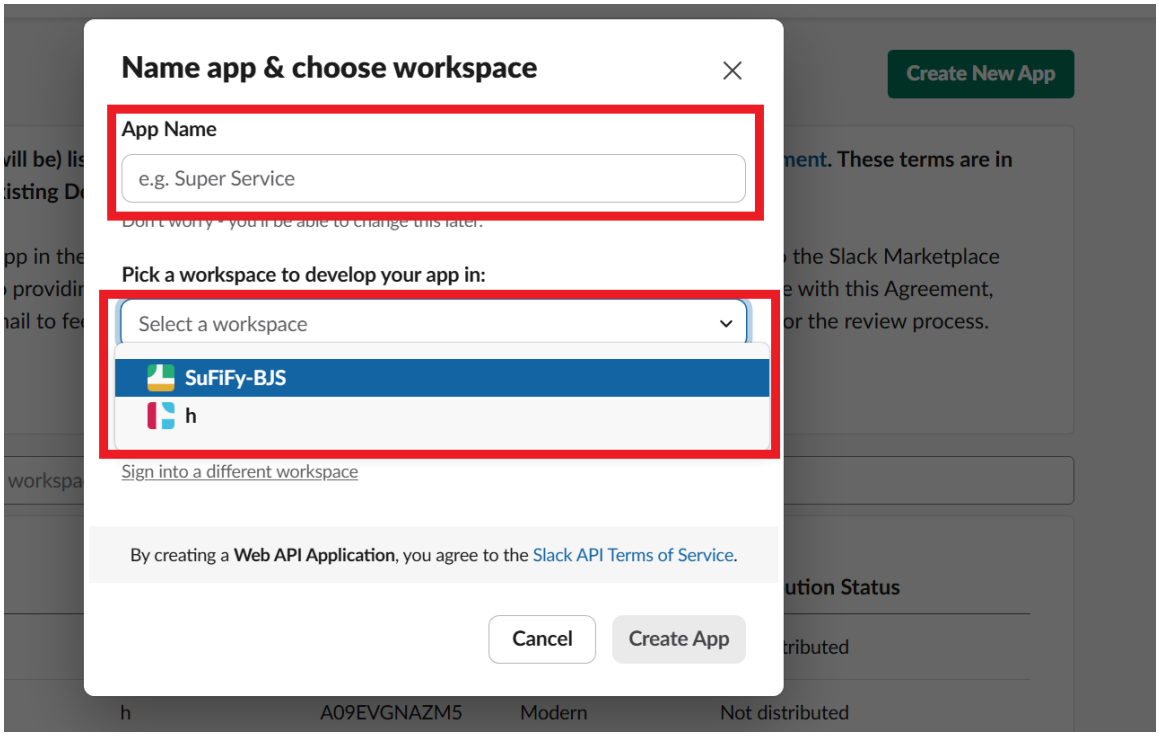
Create New App:



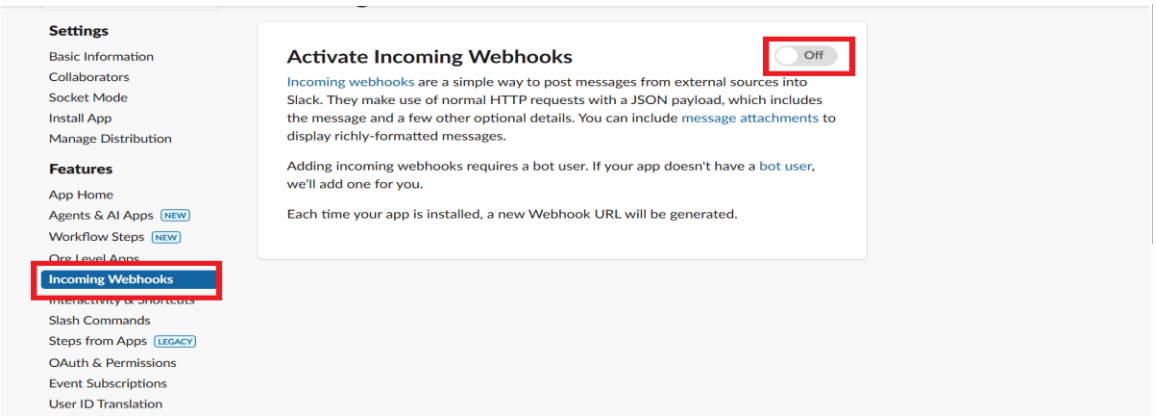
Add From Scratch:



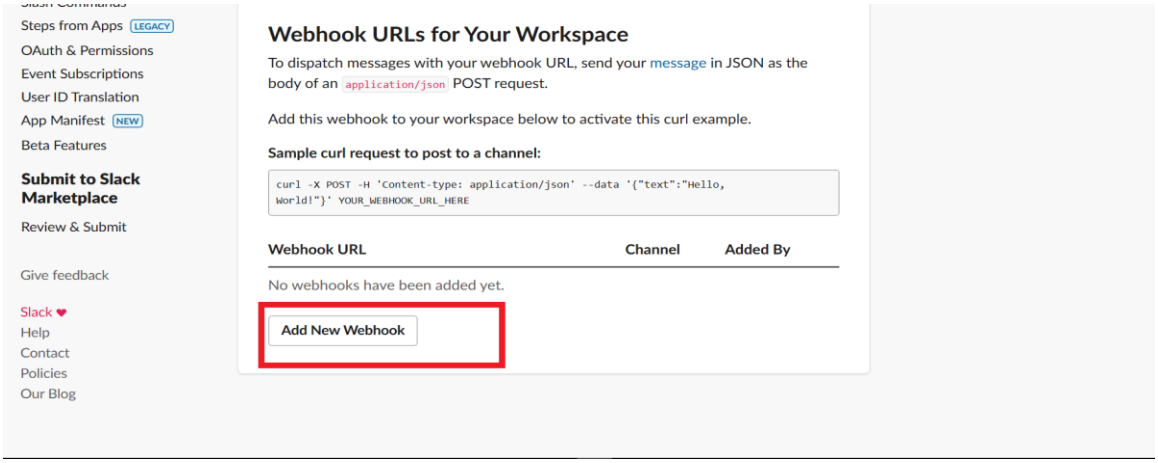
Add Name and pick up the workspace:



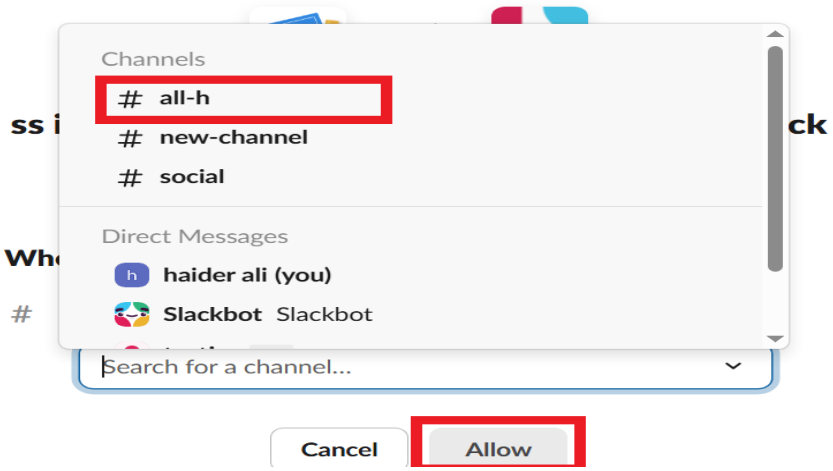
Activate the webhook:



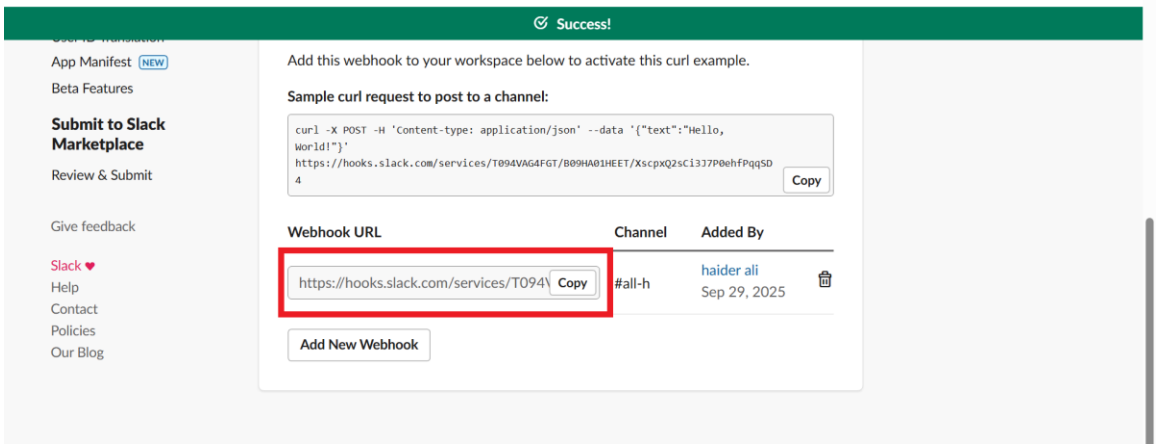
Add new webhook:



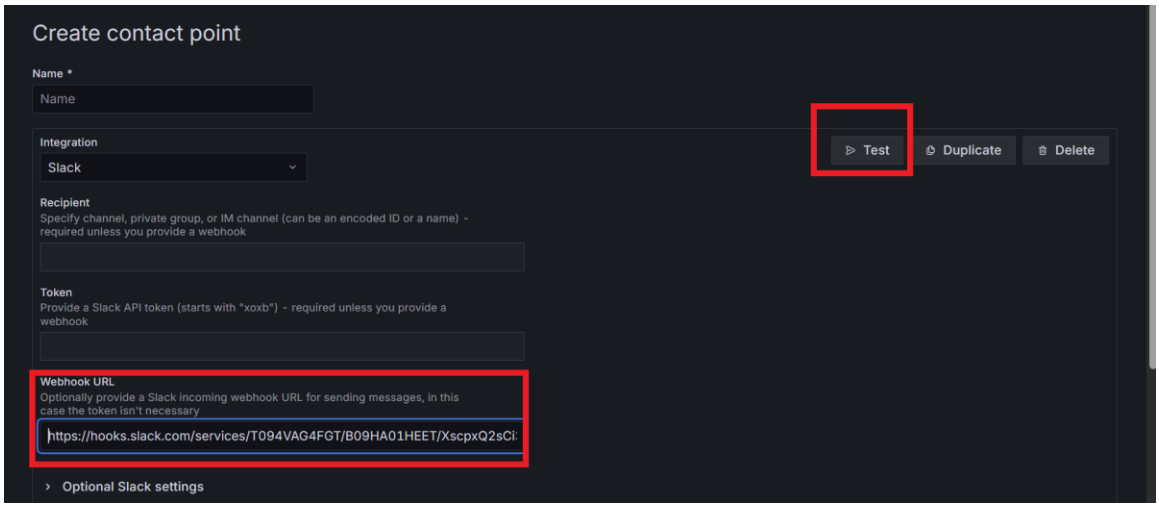
Select Channel and press allow button:



Copy the Webhook:



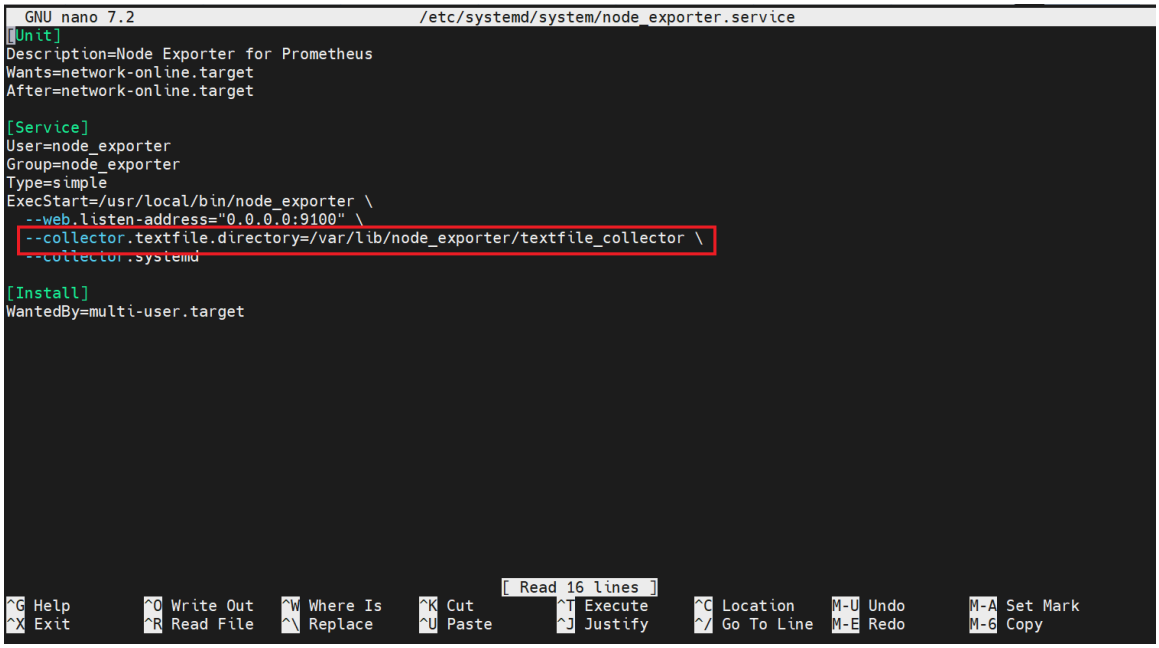
Paste the webhook url and press Test:



Node Exporter Custom Metric Creation:

1. Edit the node_exporter service

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



2. Create folder and fix permissions

sudo mkdir -p /var/lib/node_exporter/textfile_collector

sudo chown node_exporter:node_exporter /var/lib/node_exporter/textfile_collector

3. Create script

sudo nano /usr/local/bin/check-reboot-required.sh

#!/bin/bash

OUTPUT="/var/lib/node_exporter/textfile_collector/reboot_required.prom"

if [-f /run/reboot-required]; then

  echo "node_reboot_required 1" > "$OUTPUT"

else

echo "node_reboot_required 0" > "$OUTPUT"

fi

4. Adjust permission

sudo chmod +x /usr/local/bin/check-reboot-required.sh

5. Add crontab

sudo crontab -e

*/5 * * * * /usr/local/bin/check-reboot-required.sh

6. Check output

sudo /usr/local/bin/check-reboot-required.sh

cat /var/lib/node_exporter/textfile_collector/reboot_required.prom

Test Custom Metric: