

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

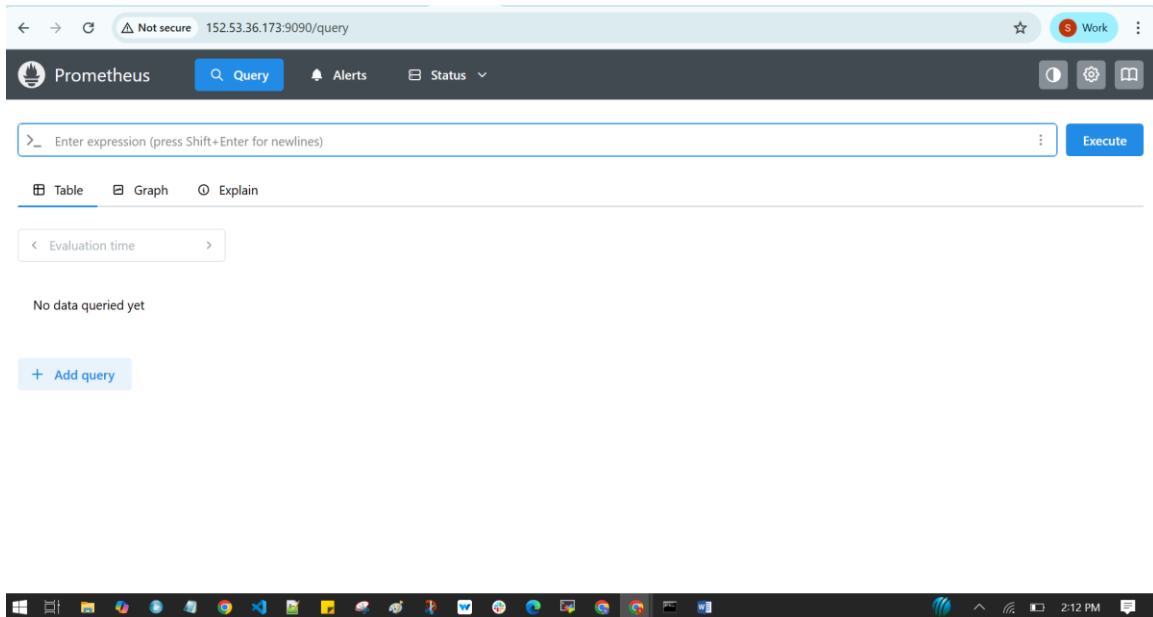
[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml /
--storage.tsdb.path=/var/lib/prometheus/ /
web.console.templates=/etc/prometheus/consoles /
web.console.libraries=/etc/prometheus/console_libraries /
web.listen.address="0.0.0.0:9090"

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

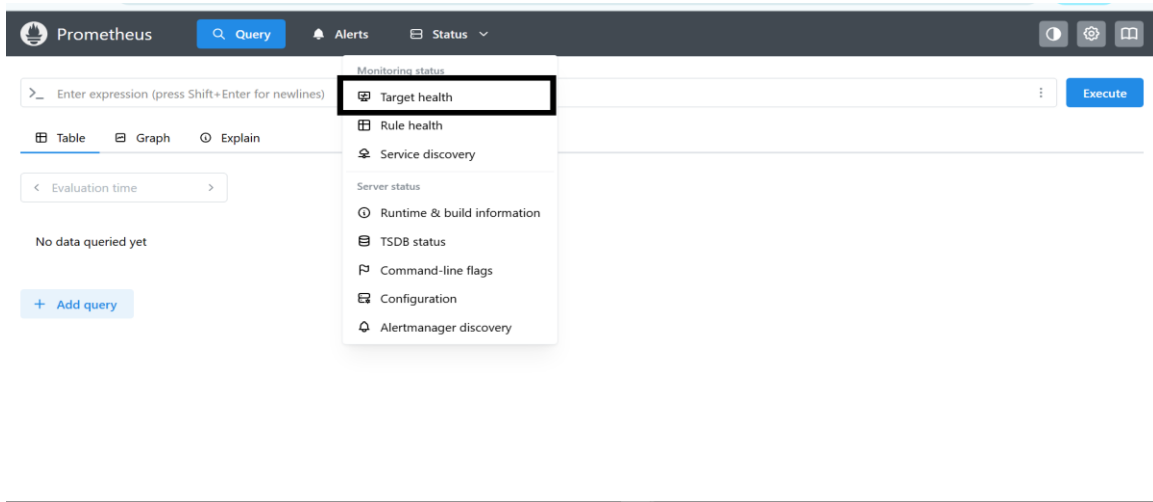
9. Reload Daemon & Start Service

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

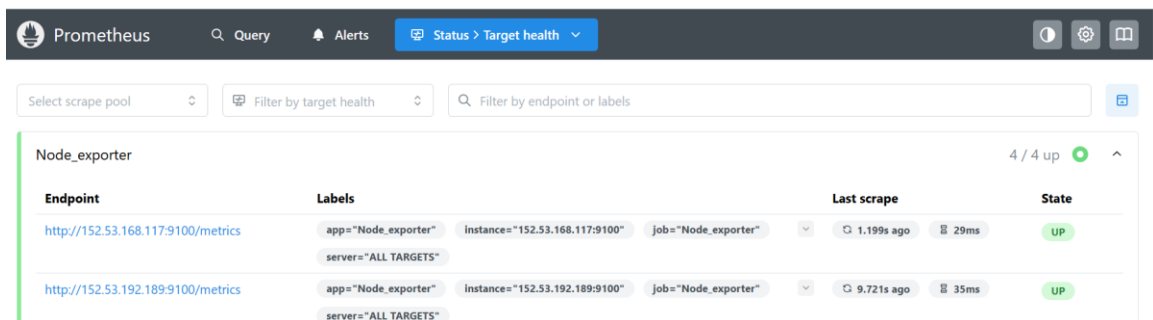
Prometheus will now be running and accessible at: <http://<server-ip>:9090>



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

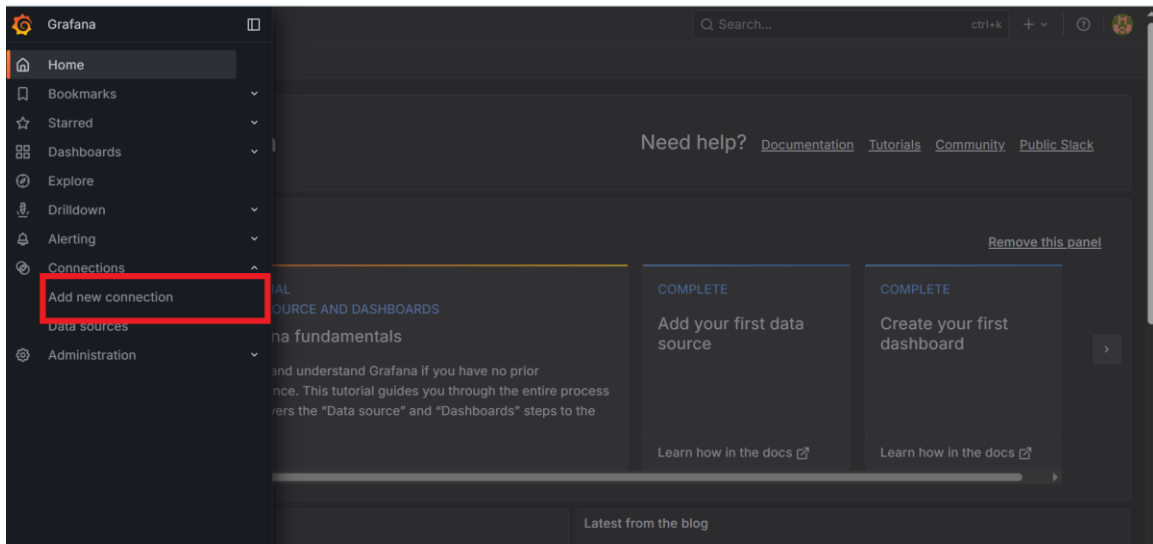
9. Add Node Exporter in prometheus.yml

sudo nano /etc/prometheus/prometheus.yml

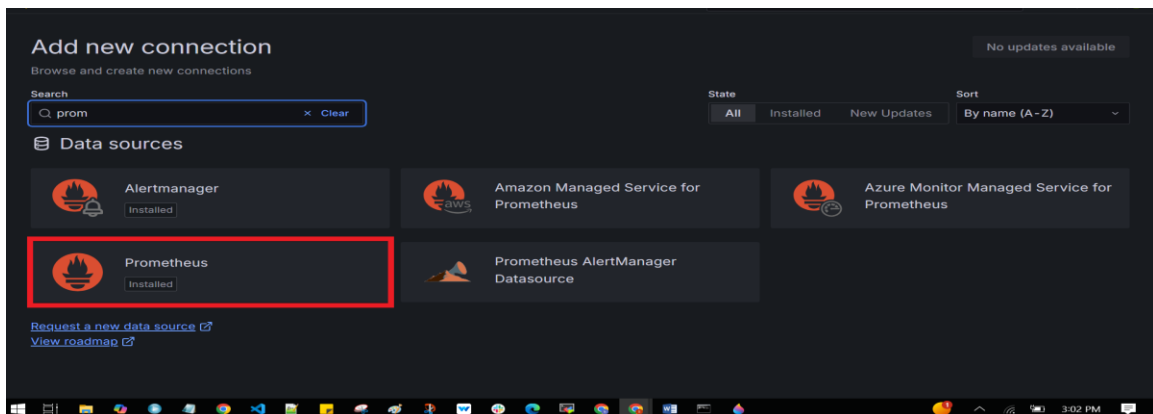
```
- job_name: "Node_exporter"
  static_configs:
    - targets: ["152.53.168.117:9100", "152.53.192.189:9100", "152.53.165.186:9100", "152.53.36.173:9100"] # lb1 Node Exp
```

10. Create Dashboard

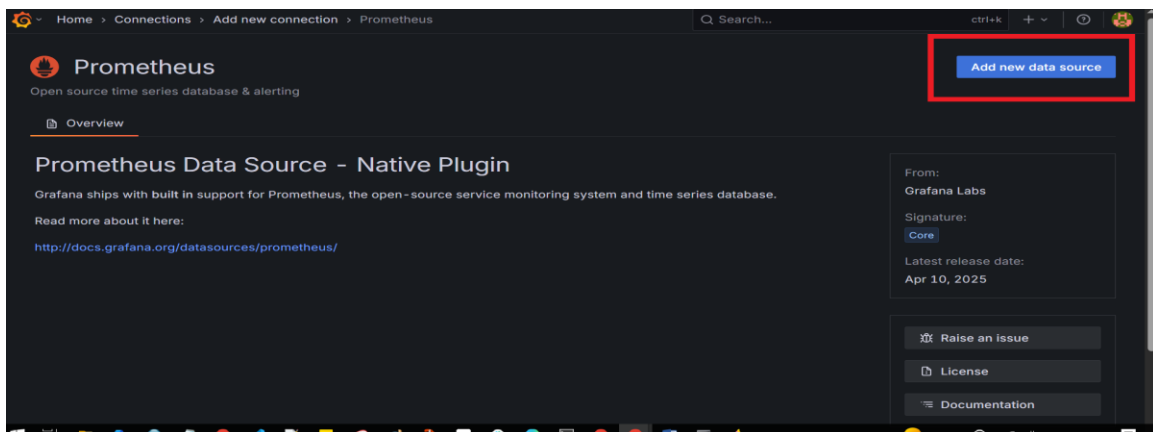
Go to Add new connection tap inside the Connections Downbar:



Select Prometheus:



Select Add new data source:



Add Name and the URL of the Prometheus:

Configure your Prometheus data source below
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-forever Grafana Cloud plan](#).

Name Default ☐

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with * are required

Connection

Prometheus server URL
Please enter a valid URL

Save and Test the datasource:

Home > Connections > Data sources > prometheus-2

Cache level

Incremental querying (beta) ☐

Disable recording rules (beta) ☐

Other

Custom query parameters

HTTP method

Series limit

Use series endpoint ☐

Exemplars

+ Add

Delete Save & test

Inside the Dashboard Folder select New Dashboard:

Home > Dashboards

Dashboards

Create and manage dashboards to visualize your data

Search for dashboards and folders

Filter by tag Starred

Sort

New ~

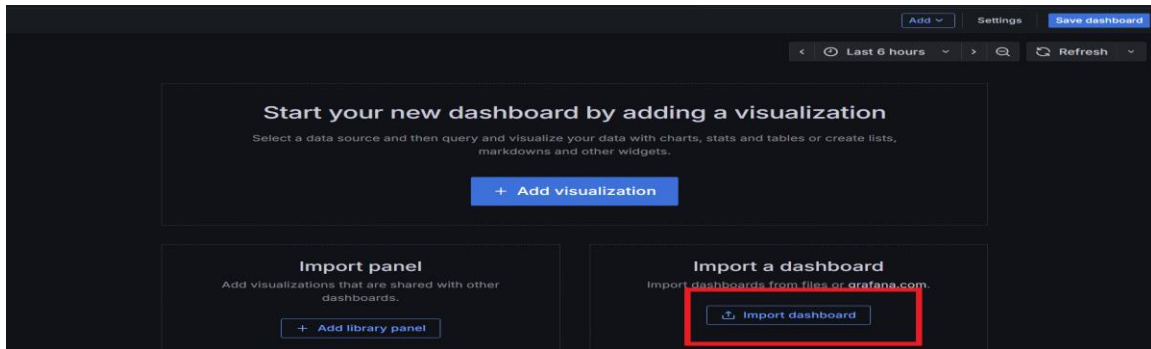
New dashboard

New folder

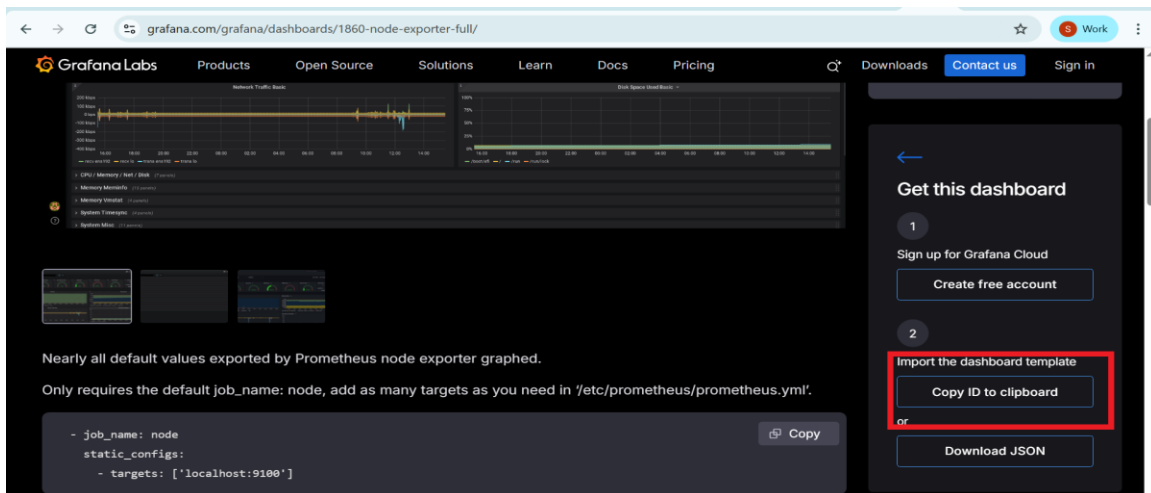
Import

Name	Tags
Alerting	
cAdvisor Docker Insights	
Cadvisor exporter	cadvisor docker
Node Exporter Full	linux
Prometheus Blackbox Exporter	blackbox prometheus

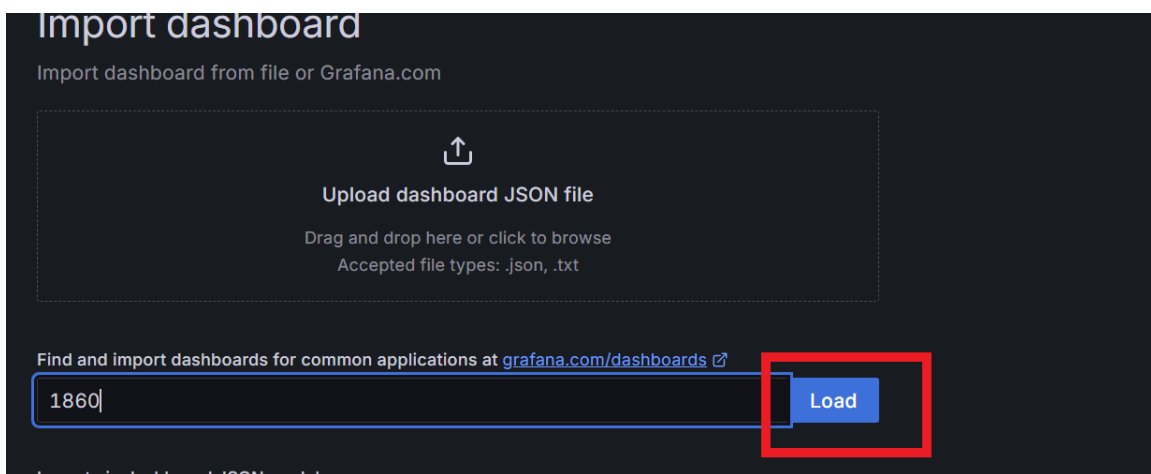
Select Import Dashboard:



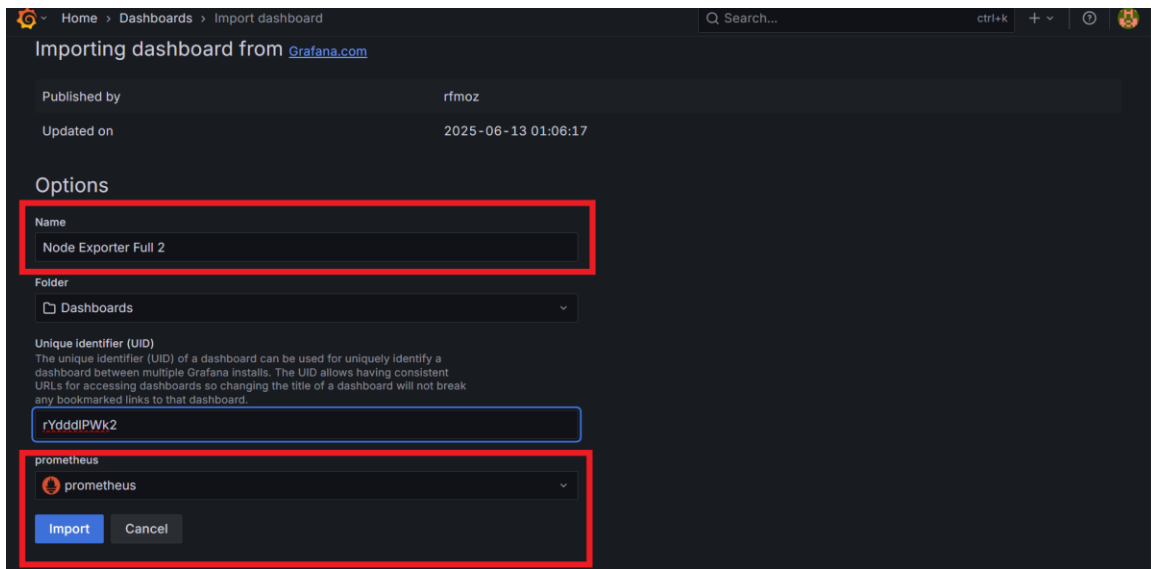
Copy Dashboard ID:



Add the ID and Press the Load button:



Add the Dashboard and the Prometheus Exporter:



The image shows the Grafana 'Import dashboard' interface. The breadcrumb navigation at the top reads 'Home > Dashboards > Import dashboard'. The page title is 'Importing dashboard from [Grafana.com](#)'. Below this, it shows 'Published by rfm0z' and 'Updated on 2025-06-13 01:06:17'. The 'Options' section contains a 'Name' field with the value 'Node Exporter Full 2', a 'Folder' dropdown set to 'Dashboards', and a 'Unique Identifier (UID)' field with the value 'rYdddlPWk2'. At the bottom, there is a 'prometheus' dropdown menu with a 'prometheus' option selected, and two buttons: 'Import' and 'Cancel'. Red rectangular boxes highlight the 'Name' field, the 'UID' field, and the 'prometheus' dropdown menu.

Home > Dashboards > Import dashboard

Importing dashboard from [Grafana.com](#)

Published by rfm0z

Updated on 2025-06-13 01:06:17

Options

Name
Node Exporter Full 2

Folder
Dashboards

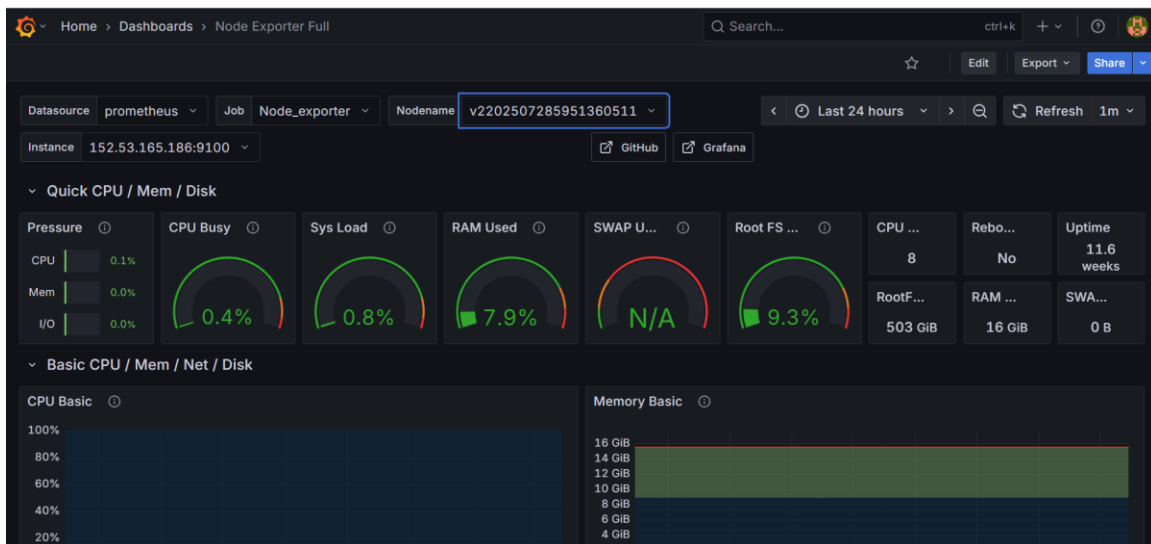
Unique Identifier (UID)
The unique identifier (UID) of a dashboard can be used to uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

rYdddlPWk2

prometheus
prometheus

Import Cancel

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]

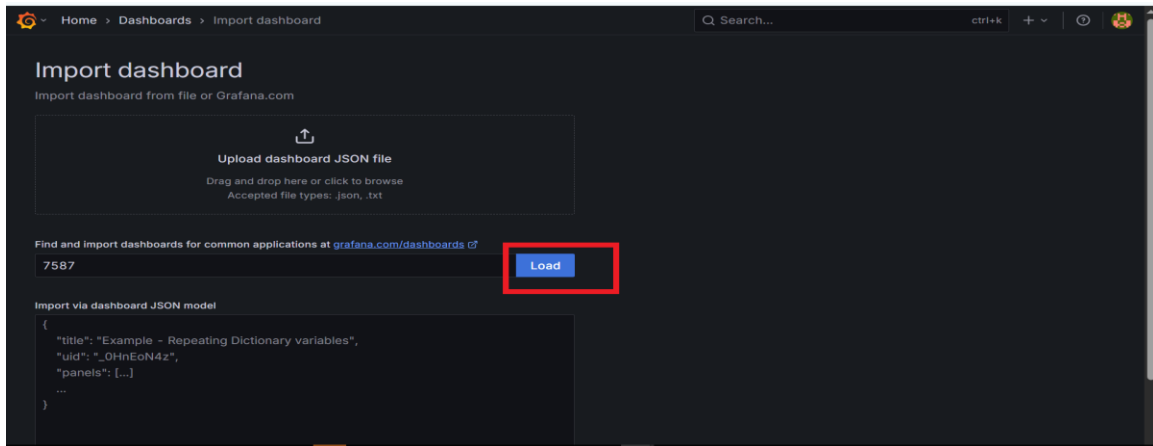
target_label: instance

- target_label: __address__

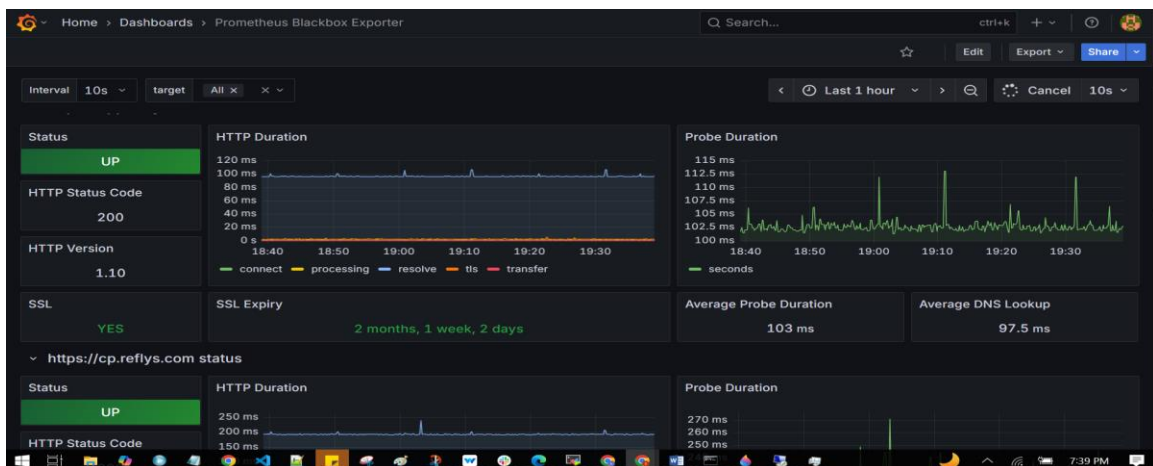
replacement: 127.0.0.1:9115

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
```

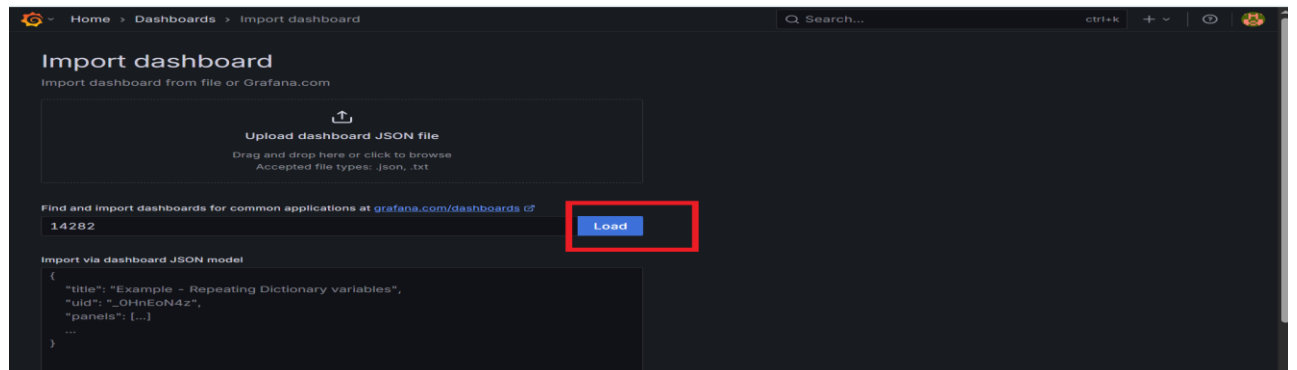
2. Add Prometheus Configuration

```
target_label: 'site_name'

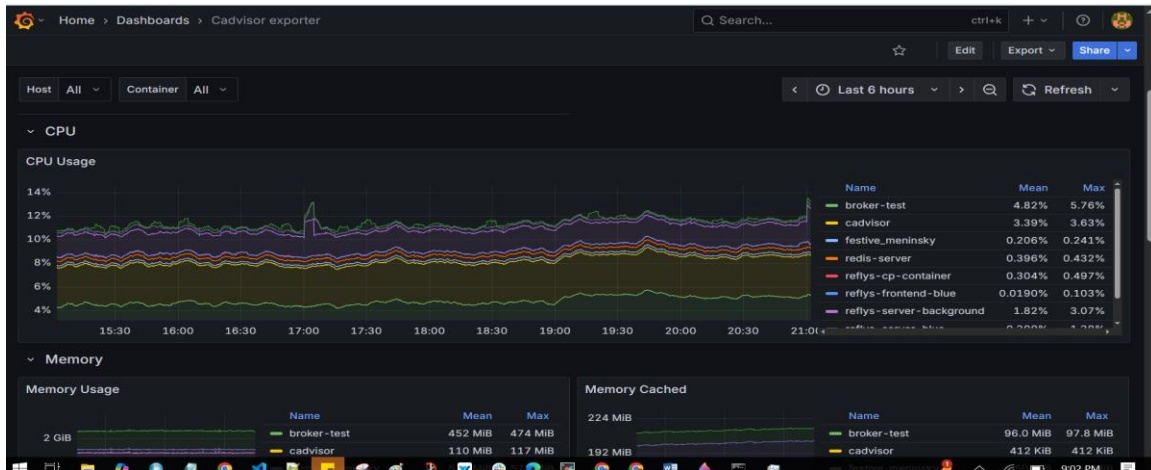
- job_name: "cadvisor"
  static_configs:
    - targets: ['152.53.36.173:8085', '152.53.165.186:8085', '152.53.192.189:8085']
```

3. Add Docker Desktop:

ADD Dashboard ID:

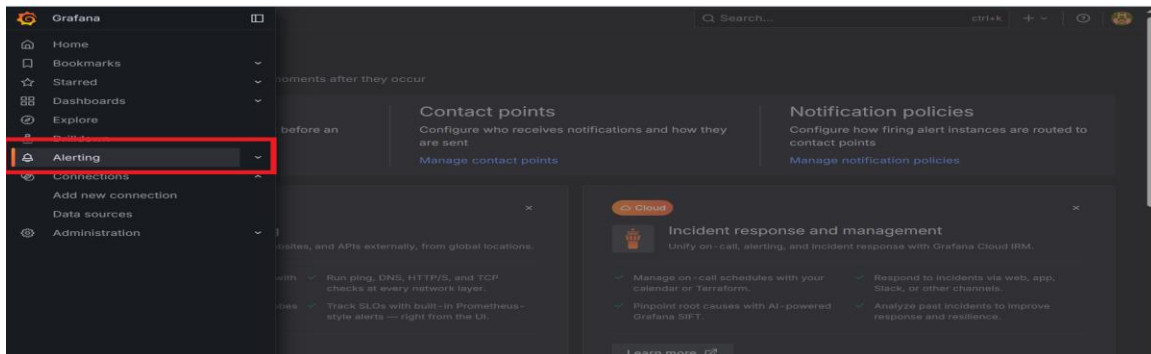


View the dashboard:

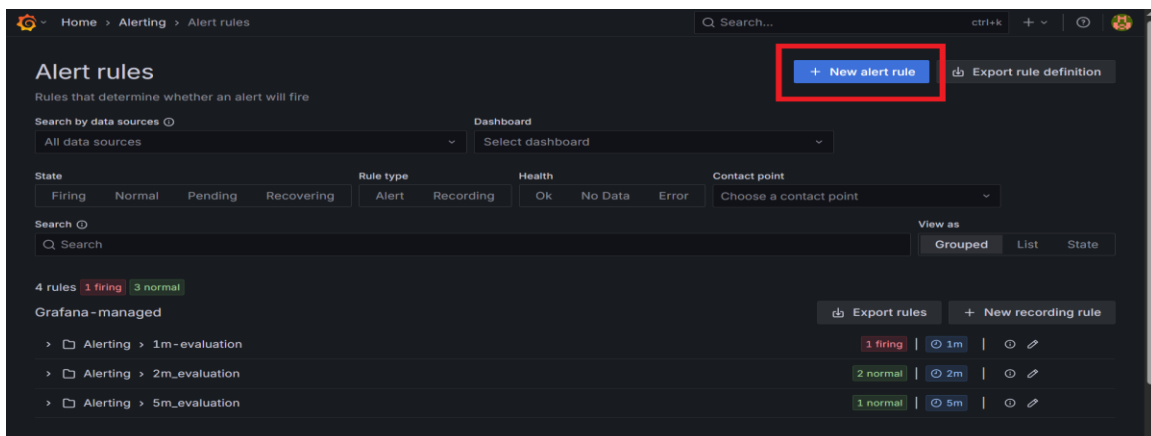


Grafana Alerting:

Open the Alerting tab:



Select the New alert rule:



Add alert rule and add PROMQL query:

New alert rule

1. Enter alert rule name
Enter a name to identify your alert rule.

Name
Give your alert rule a name

2. Define query and alert condition [Need help?](#)

prometheus Options 10 minutes

Kick start your query Explain Run queries Builder Code

Metrics browser > Enter a PromQL query...

Options Legend: Auto Format: Time series Step: auto Type: Instant

Alert condition

WHEN QUERY IS ABOVE 0

Add the query condition and select folder:

Alert condition

WHEN QUERY IS ABOVE 0

Preview alert rule condition

3. Add folder and labels
Organize your alert rule with a folder and set of labels. [Need help?](#)

Folder
Select a folder to store your rule in.

Select folder + New folder

Labels
Add labels to your rule for searching, silencing, or routing to a notification policy. [Need help?](#)

No labels selected + Add labels

4. Set evaluation behavior
Define how the alert rule is evaluated. [Need help?](#)

Select a folder before setting evaluation group and interval

Select Evaluation Period And Pending Period in alert:

4. Set evaluation behavior
Define how the alert rule is evaluated. [Need help?](#)

Select a folder before setting evaluation group and interval

Select an evaluation group... or + New evaluation group

Pending period
Period during which the threshold condition must be met to trigger an alert.

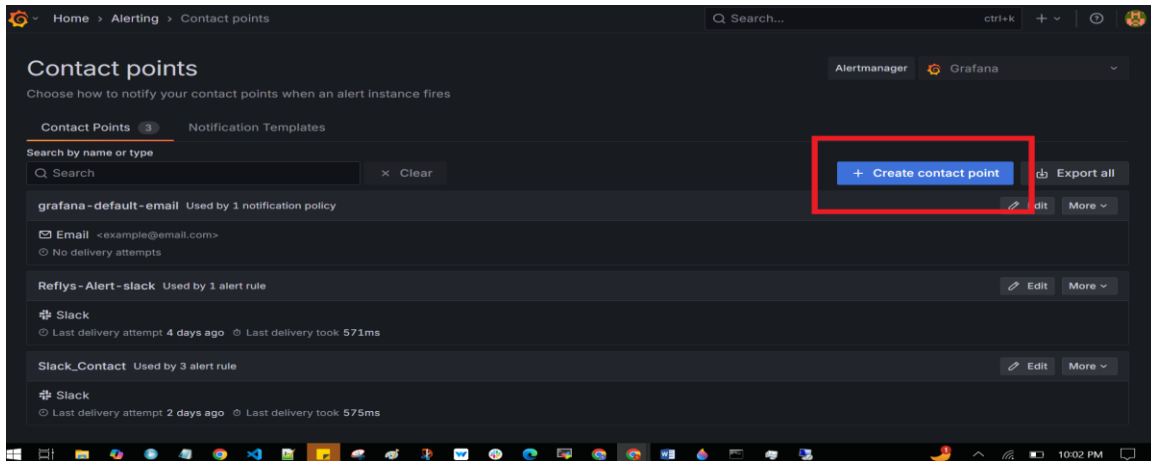
1m
None 1m 2m 3m 4m 5m

Keep firing for
Period during which the alert will continue to show up as firing even though the threshold condition is no longer breached. Selecting "None" means the alert will be back to normal immediately.

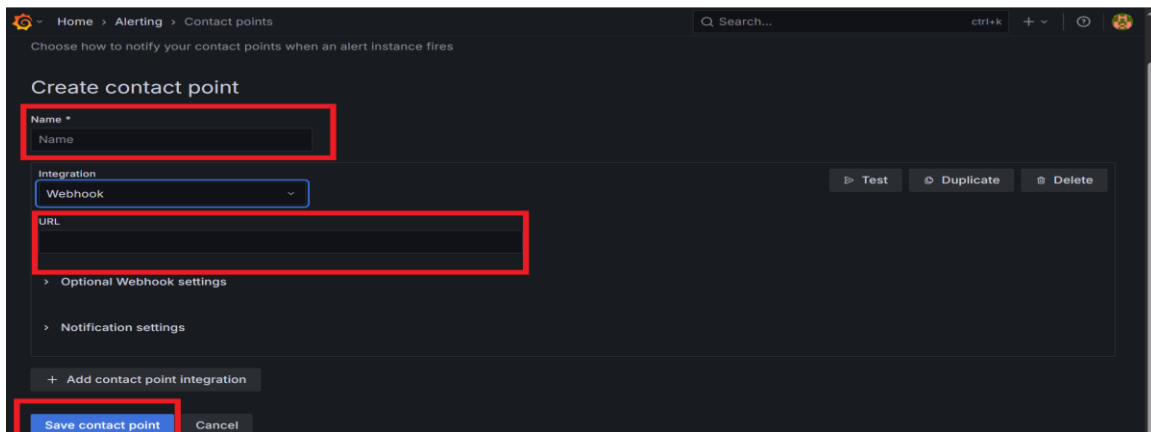
0s
None 1m 2m 3m 4m 5m

> Configure no data and error handling

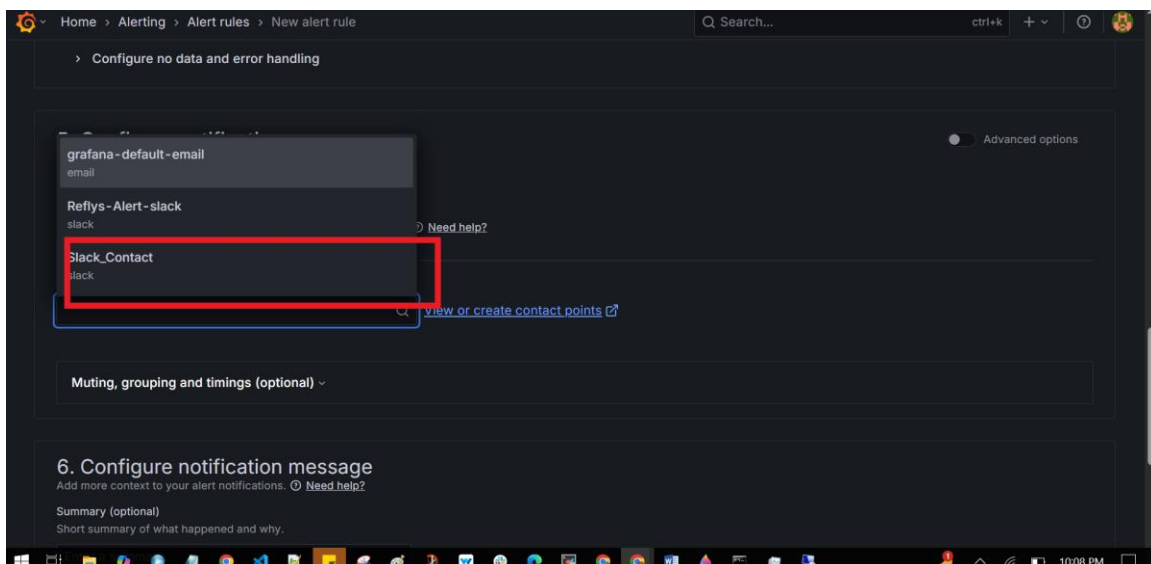
Create New Contact Point:



Select Name and slack Webhook Url:



Select the contact point:



Add summary or description and save alert:

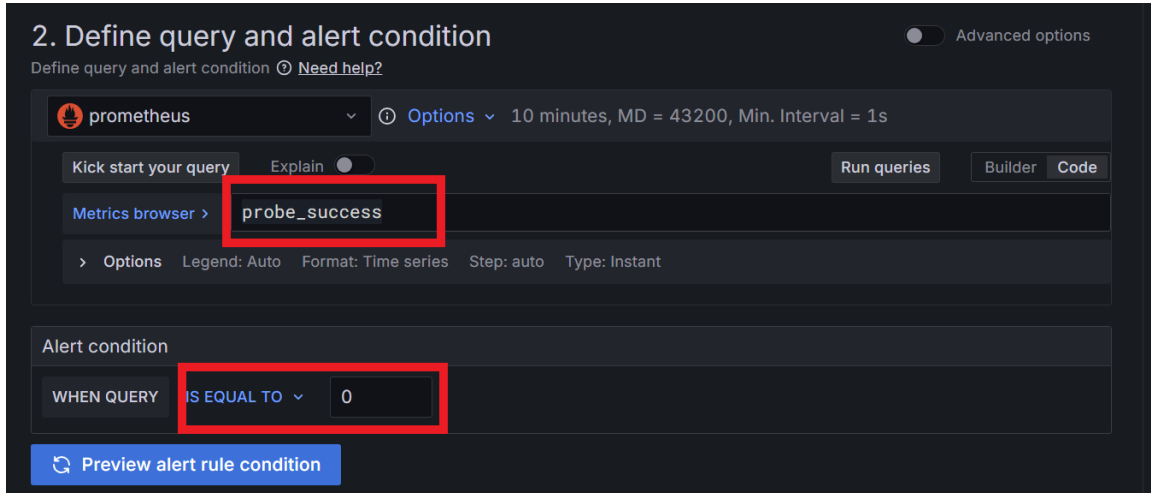
The screenshot shows the '6. Configure notification message' form in Grafana. The form is titled '6. Configure notification message' and includes a subtitle 'Add more context to your alert notifications. [Need help?](#)'. The form contains three main sections: 'Summary (optional)' with a text area labeled 'Enter a summary...', 'Description (optional)' with a text area labeled 'Enter a description...', and 'Runbook URL (optional)' with a text area labeled 'Enter a runbook URL...'. Below these sections are two buttons: '+ Add custom annotation' and 'Link dashboard and panel'. At the bottom of the form, there are two buttons: 'Save' and 'Cancel'. A red box highlights the 'Summary (optional)' and 'Description (optional)' sections, and another red box highlights the 'Save' button.

Check all alerts:

The screenshot shows the 'Alert rules' page in Grafana. The page has a dark theme and a sidebar on the left with navigation links for 'Dashboards', 'Panels', and 'Alert rules'. The 'Alert rules' link is highlighted. The main content area has a search bar labeled 'Search alert rules by name' and a sort dropdown menu labeled 'Sort (Default A-Z)'. Below the search bar, there is a list of alert rules. The list contains four entries: 'Full_VOLUME_Alert', 'High_CPU_Alert', 'High_RAM_Alert', and 'Website - Down - Alert'. Each entry has a small icon to its left and the label 'Alerting' below it.

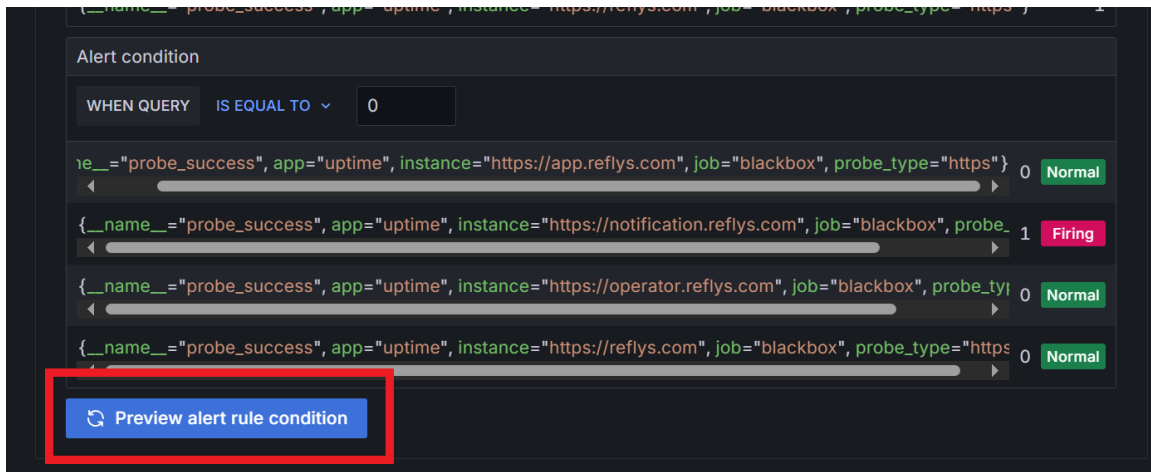
Grafana Website Down Alerts Setup:

Use this query and condition in alert:



The screenshot shows the '2. Define query and alert condition' step in Grafana. The data source is set to 'prometheus'. The query is 'probe_success', which is highlighted with a red box. The alert condition is set to 'WHEN QUERY IS EQUAL TO 0', with '0' also highlighted by a red box. A 'Preview alert rule condition' button is visible at the bottom.

Press the Preview button:



The screenshot shows the 'Alert condition' preview page. It displays a table of alert results for the 'probe_success' query. The table has columns for the alert name, app, instance, job, probe type, and status. The status column shows 'Normal' for most alerts and 'Firing' for one. The 'Preview alert rule condition' button is highlighted with a red box.

Alert name	app	instance	job	probe type	Status
probe_success	uptime	https://app.reflys.com	blackbox	https	Normal
probe_success	uptime	https://notification.reflys.com	blackbox	https	Firing
probe_success	uptime	https://operator.reflys.com	blackbox	https	Normal
probe_success	uptime	https://reflys.com	blackbox	https	Normal

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 }}
```

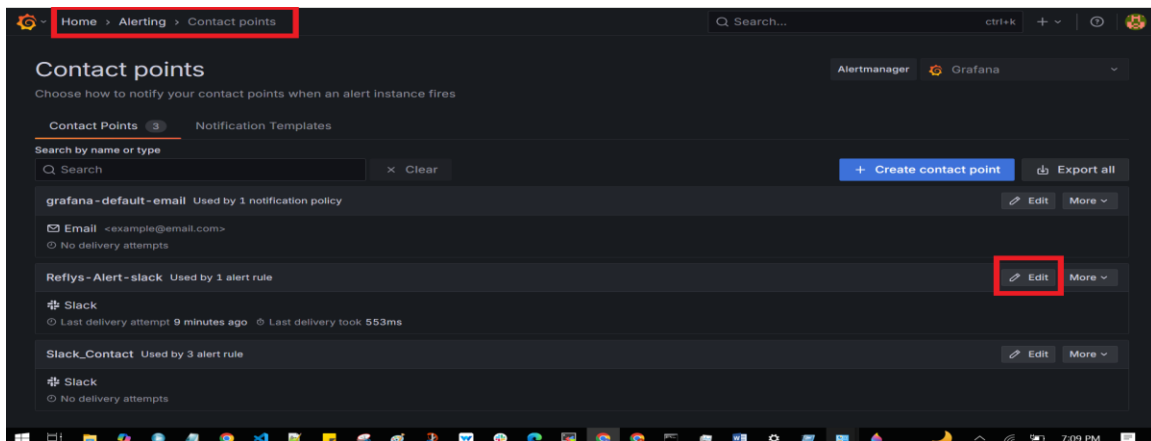
```
Reason: Alert condition is no longer true.
```

```
{{- end }}
```

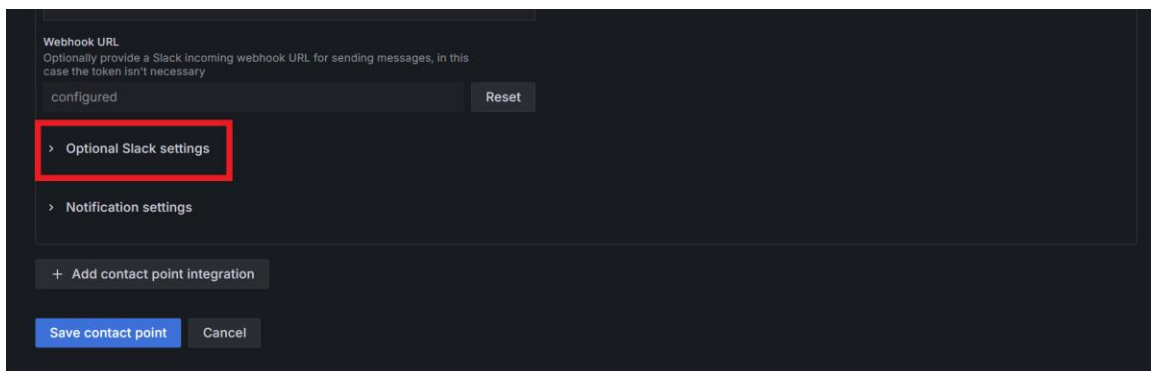
```
{{- end }}
```

```
{{ end }}
```

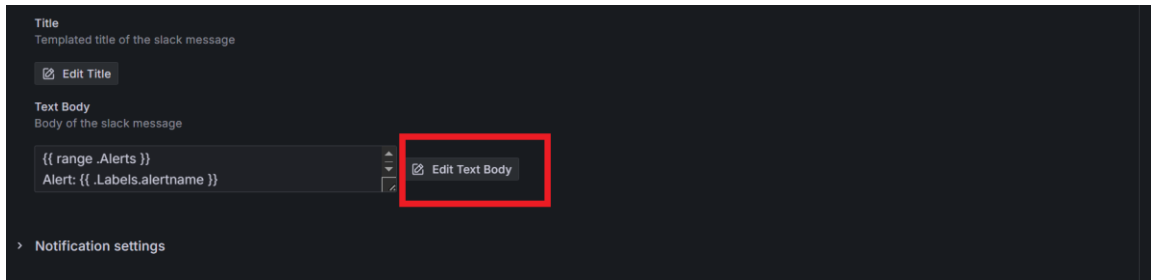
Inside the Contact points(Edit Contact):



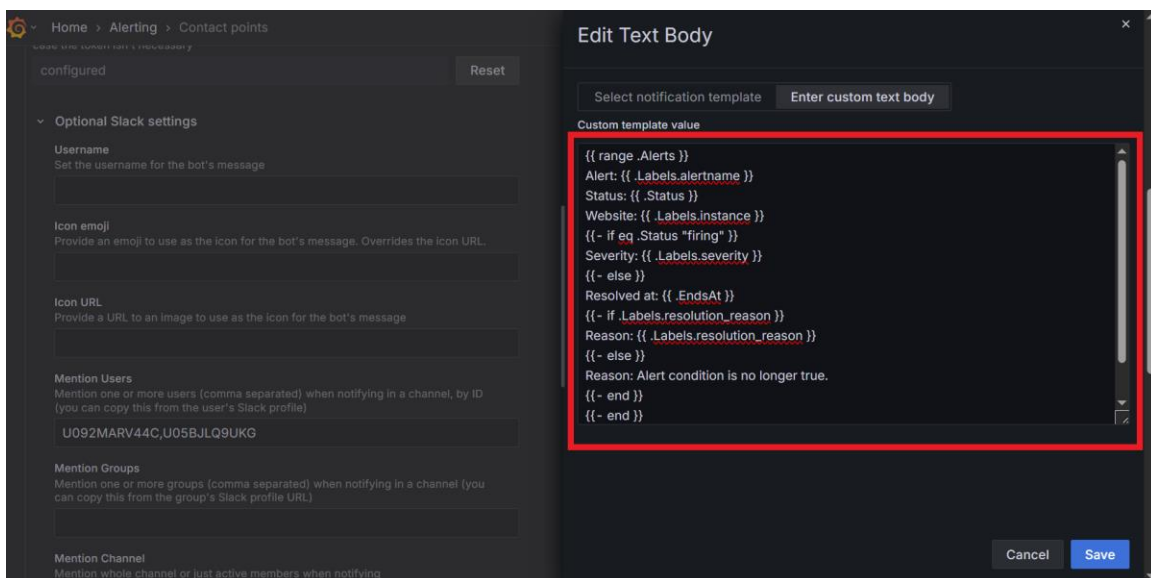
Edit the Optional Slack Settings:



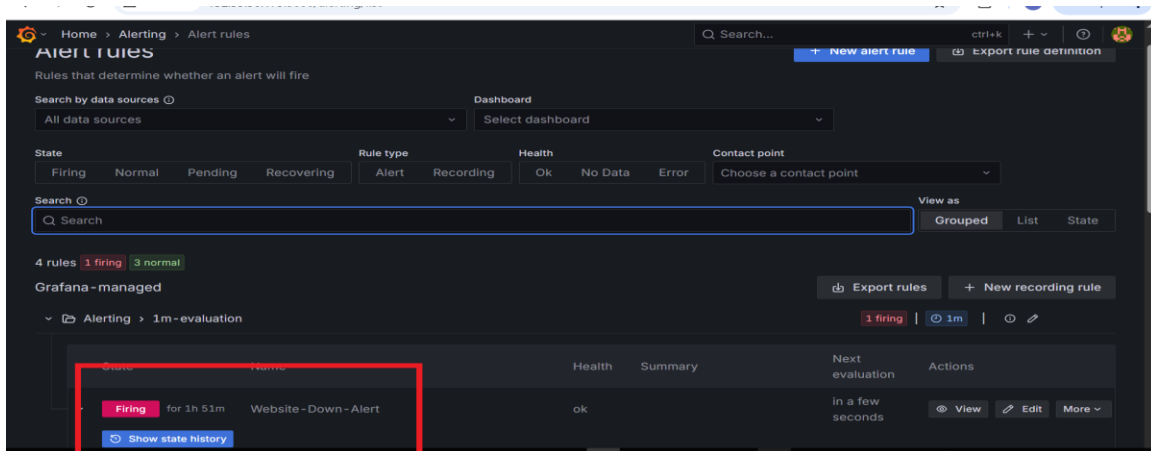
Press the Edit Text Body Button:



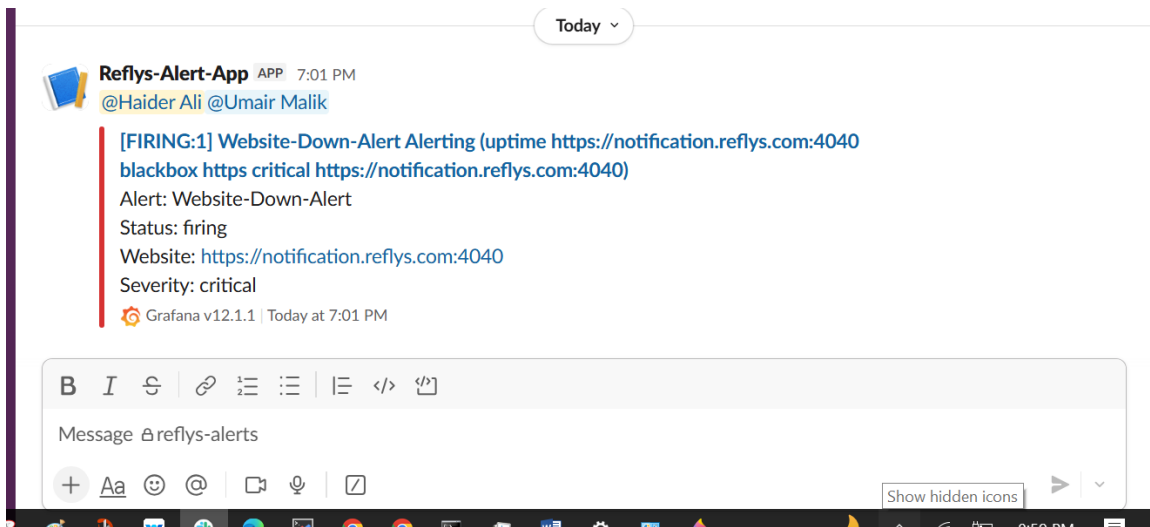
Add the New Syntax in this box:



Check Alert State:

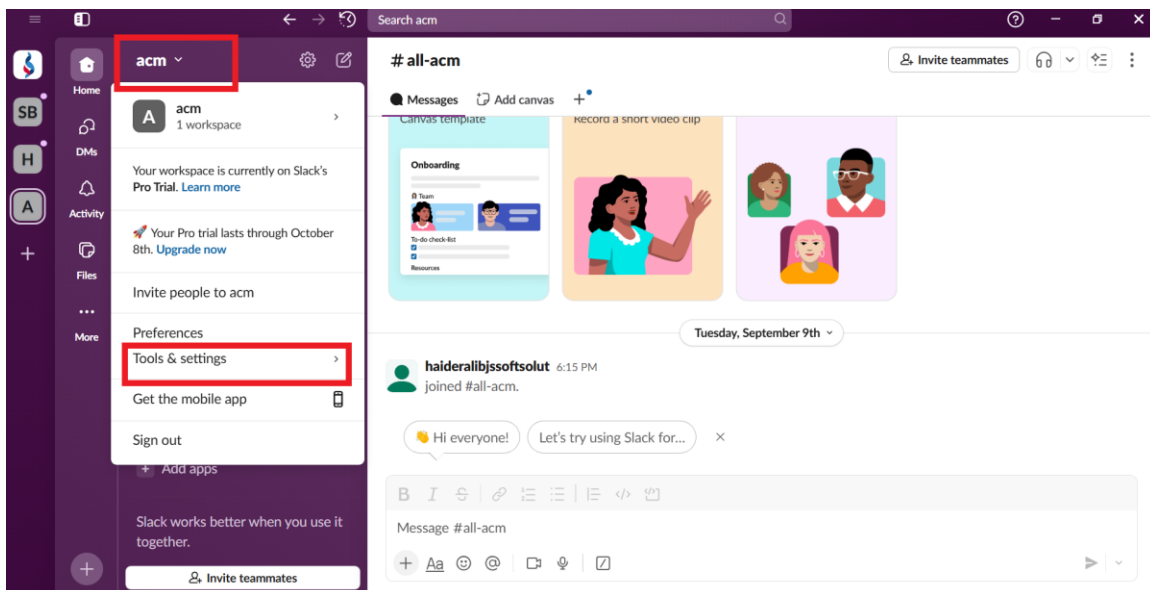


Check Slack Message:

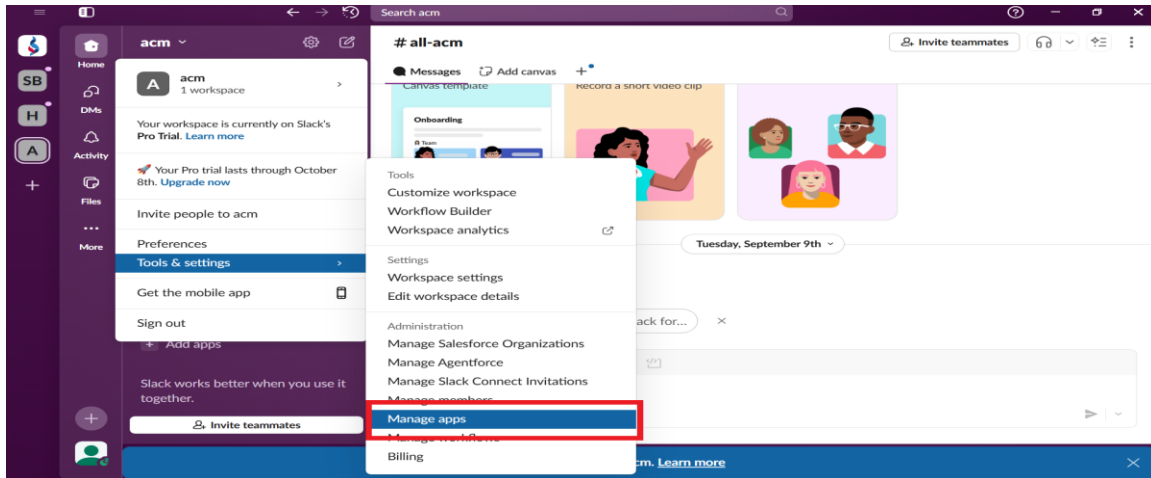


Slack Webhook Creation:

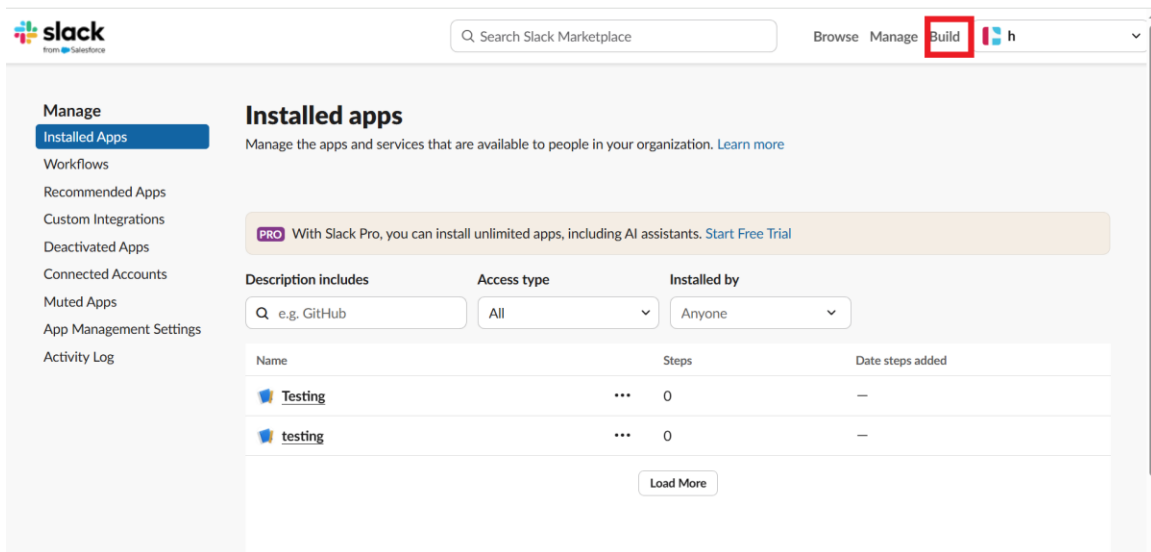
First Select the Workspace Name and then press on **Tools & settings** button:



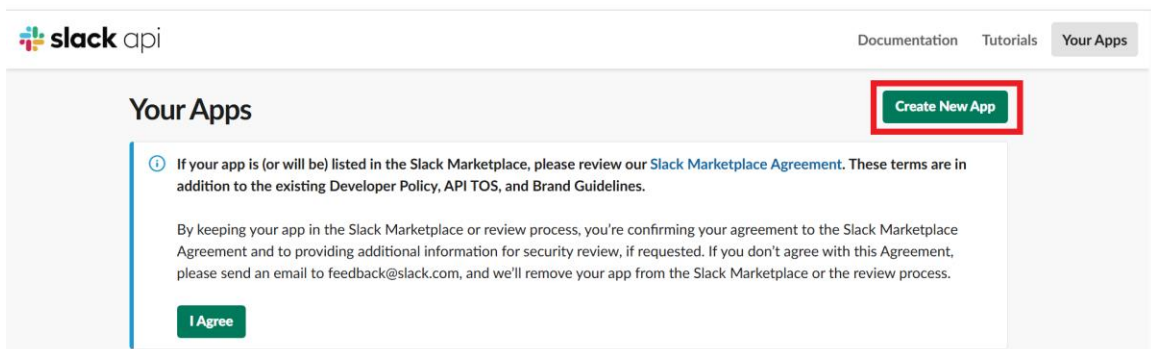
Press the Manage Apps Button:



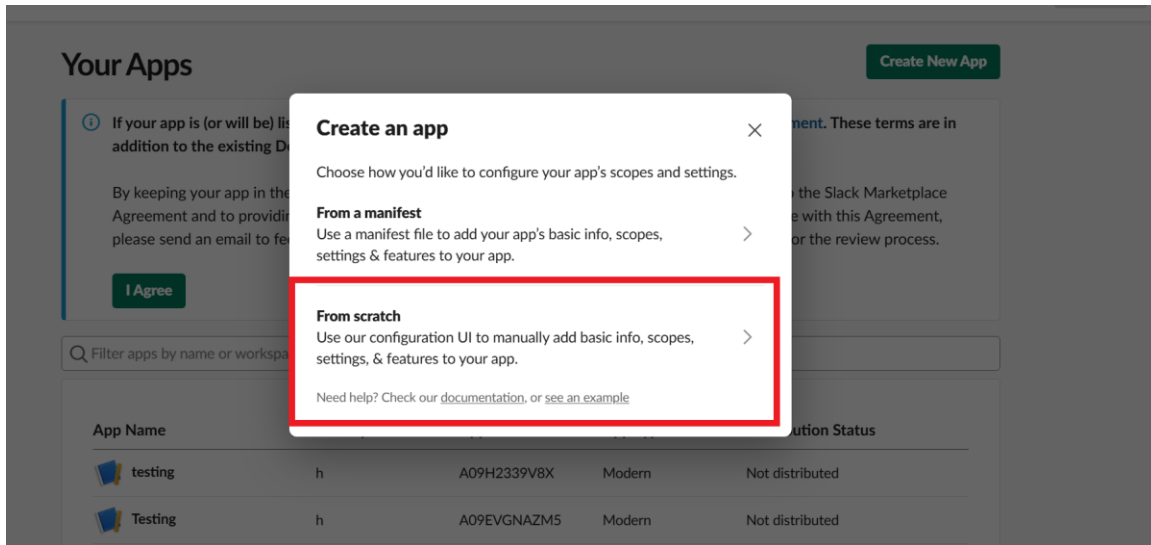
Click on Build Button:



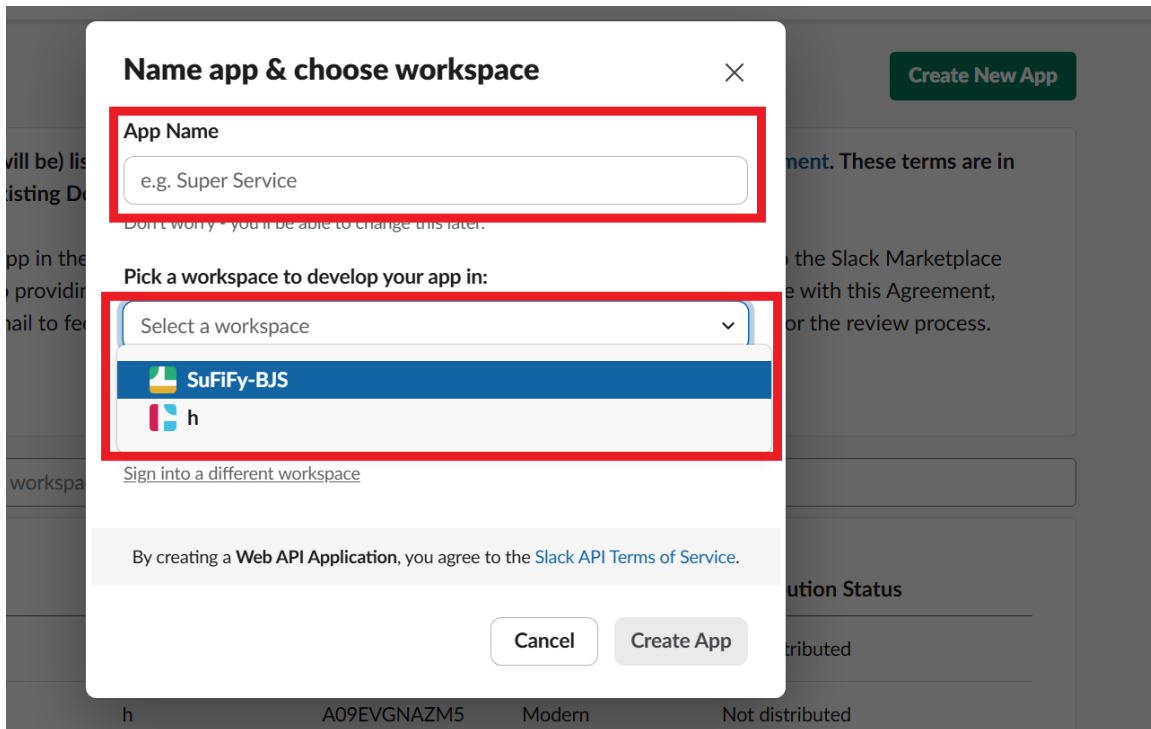
Create New App:



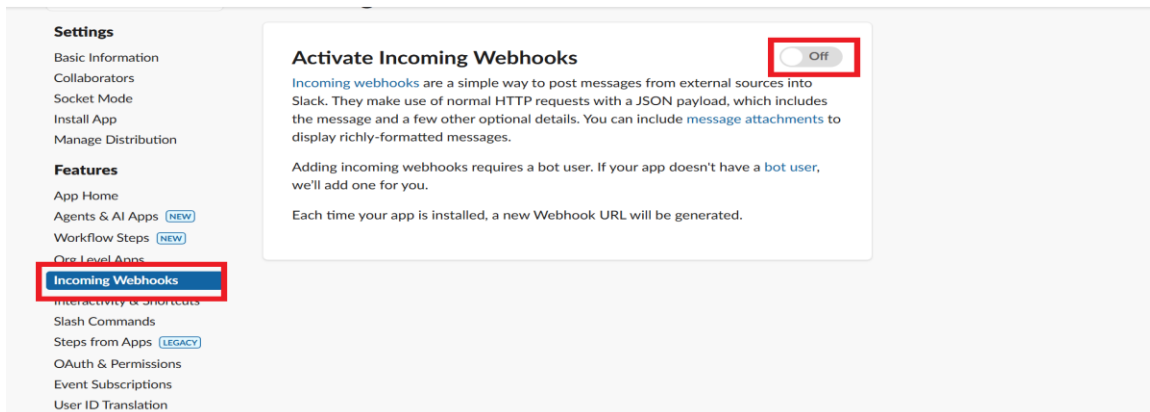
Add From Scratch:



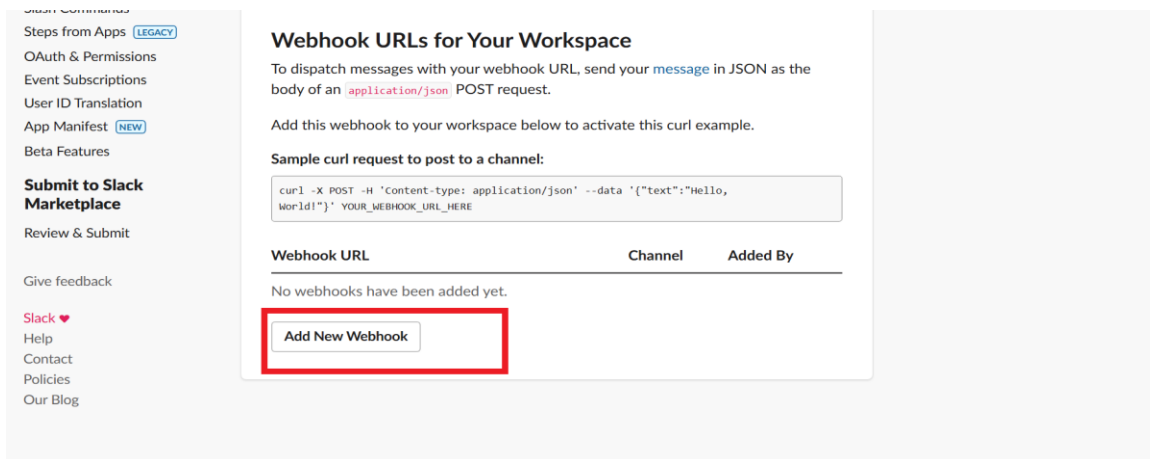
Add Name and pick up the workspace:



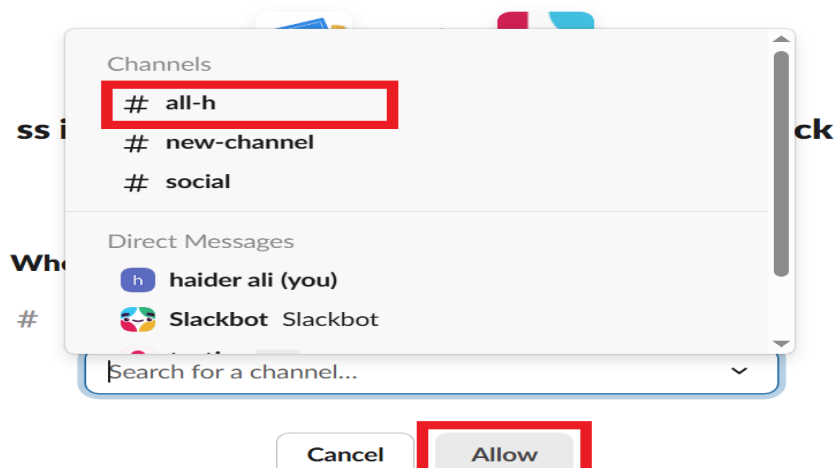
Activate the webhook:



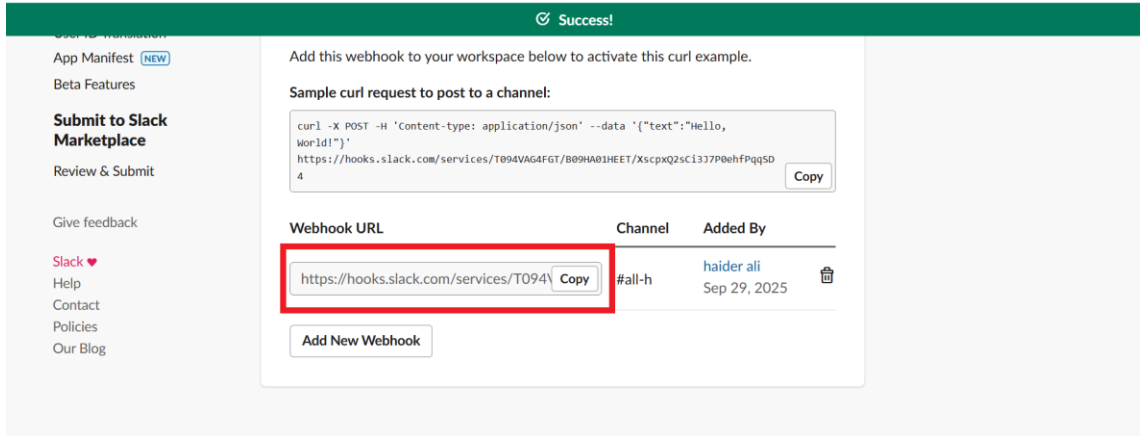
Add new webhook:



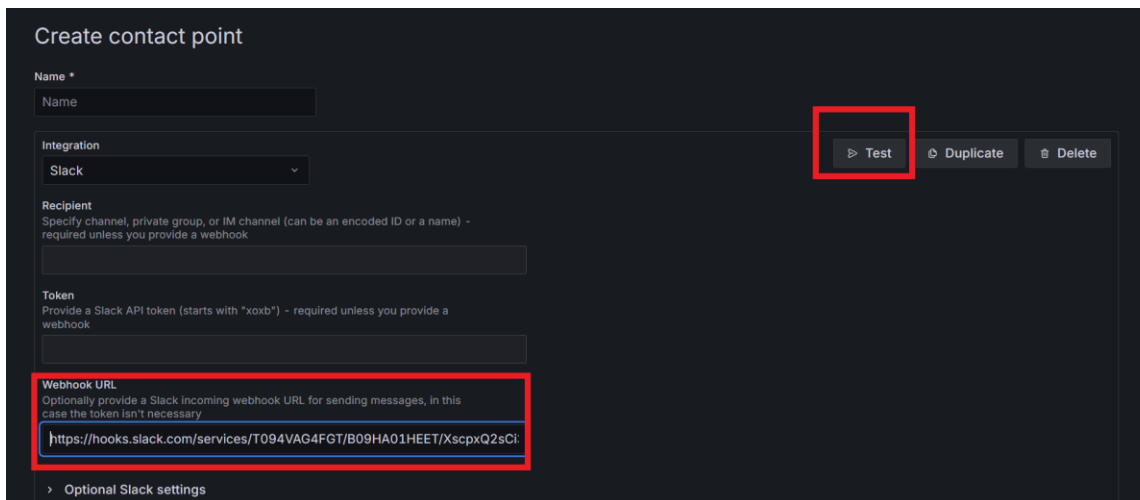
Select Channel and press allow button:



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
```

```
GNU nano 7.2 /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" \
--collector.textfile.directory=/var/lib/node_exporter/textfile_collector \
--collector.systemd

[Install]
WantedBy=multi-user.target

Read 16 lines
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^G Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line M-E Redo     M-C Copy
```

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:

