

## Tugas:

### 1. Install VM dan Konfigurasi Nginx

- Install 1 VM, kemudian konfigurasi Nginx.
- Buat file `index.html` sederhana dengan isi:  
`<h1>Hello, ini website pertama saya!</h1>`
- **Pengujian:** Akses `http://IP_VM` dan pastikan halaman yang dibuat muncul.

### 2. Monitoring Memory Usage

Buat monitoring memory usage dari VM di atas dengan menggunakan **Node Exporter** (untuk pengambilan metrik), **Prometheus** (sebagai data source), dan **Grafana** (untuk visualisasi).

### 3. Alert Memory Usage

- Buat alert ketika memory usage melewati threshold yang ditentukan.
- Kirim notifikasi alert tersebut ke **Discord**.

### 4. Monitoring Webserver dengan Uptime Kuma

- Konfigurasi monitoring webserver menggunakan **Uptime Kuma**.
- Jika `http://IP_VM` mengalami **down**, kirim notifikasi alert ke **Discord**.

#### 1. install nginx

```
sudo apt update
sudo apt install nginx -y
```

#### 2. Cek nginx

```
ubuntu@ip-172-31-0-203:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-12-21 22:00:12 UTC; 51s ago
```

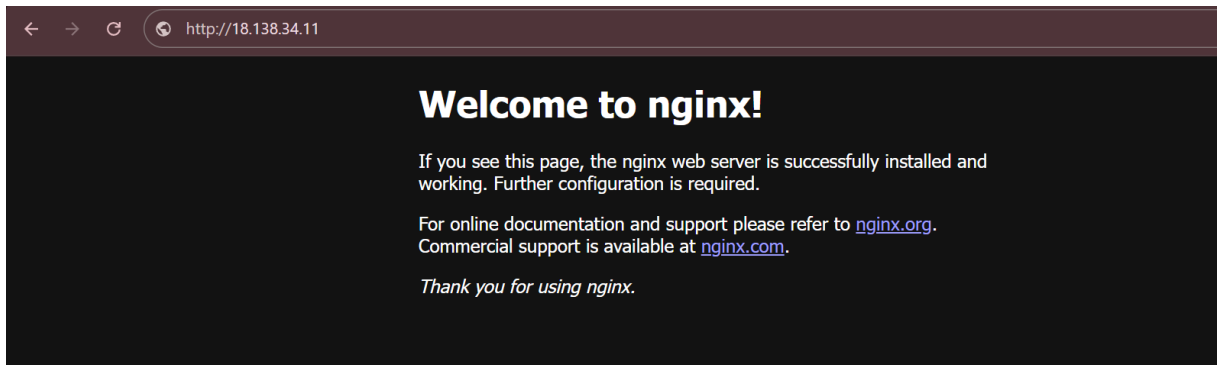
Allow port 80 di security group

**Inbound rules** [Info](#)

Inbound rule 1 Delete

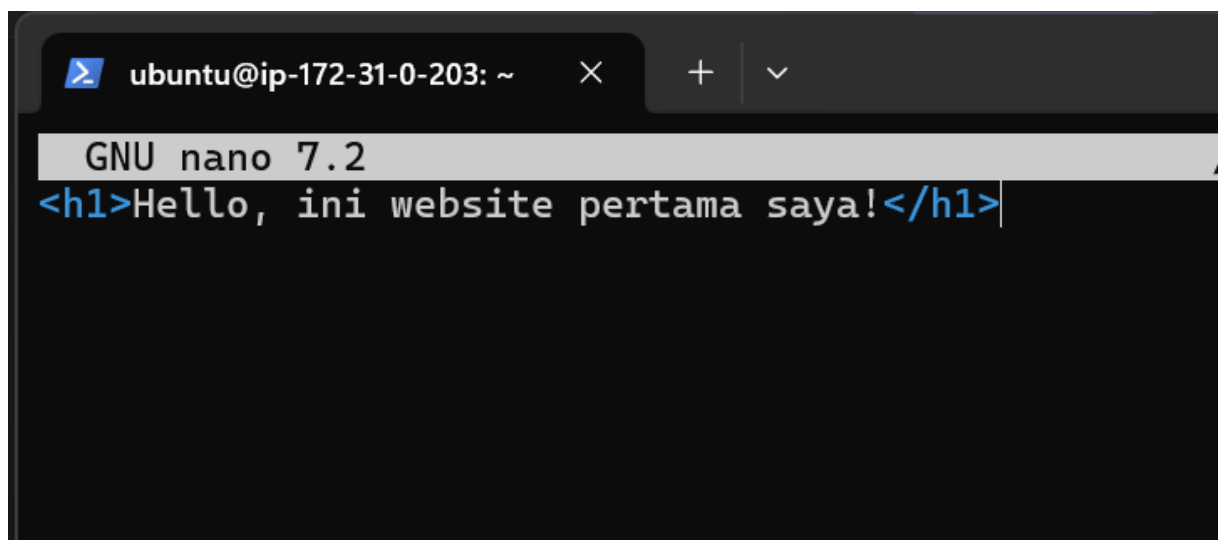
Security group rule ID sgr-0f1195d040d659e0e	Type <a href="#">Info</a> HTTP	Protocol <a href="#">Info</a> TCP
Port range <a href="#">Info</a> 80	Source type <a href="#">Info</a> Custom	Source <a href="#">Info</a> 0.0.0.0/0

Cek di browser:



3. Buat index.html

```
ubuntu@ip-172-31-0-203:~$ sudo nano /var/www/html/index.html
```



Dan test lagi :



**Hello, ini website pertama saya!**

## STEP 2

1. Install node exporter

```
cd /tmp
```

```
wget
```

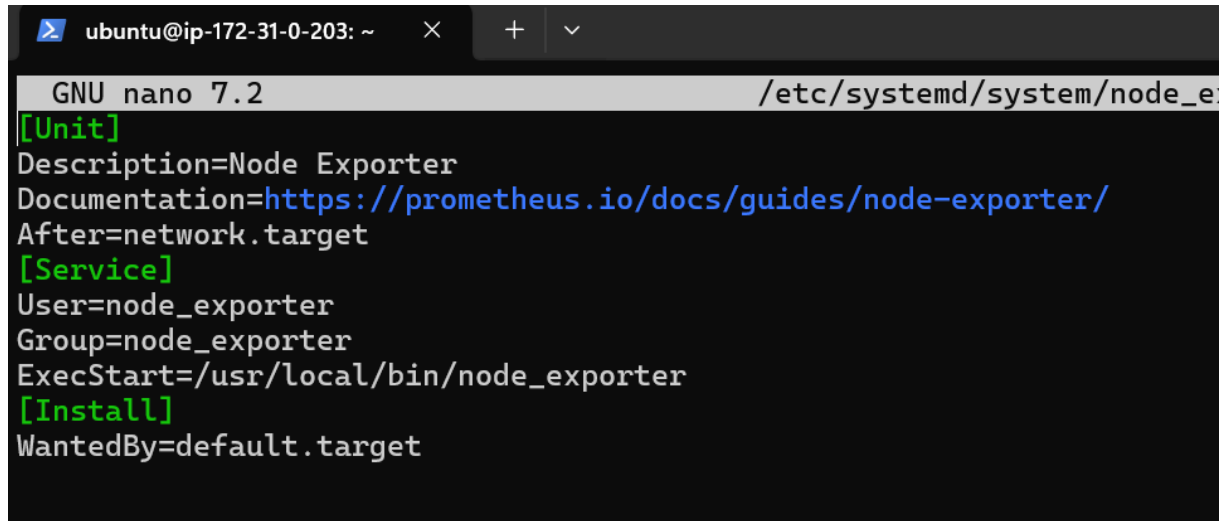
```
https://github.com/prometheus/node_exporter/releases/download/v1.8.1/no  
de_exporter-1.8.1.linux-amd64.tar.gz
```

```
tar xvf node_exporter-*.tar.gz
```

```
sudo mv node_exporter-*/node_exporter /usr/local/bin/
```

## 2. Buat systemd service

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



```
ubuntu@ip-172-31-0-203: ~  
GNU nano 7.2 /etc/systemd/system/node_e:  
[Unit]  
Description=Node Exporter  
Documentation=https://prometheus.io/docs/guides/node-exporter/  
After=network.target  
[Service]  
User=node_exporter  
Group=node_exporter  
ExecStart=/usr/local/bin/node_exporter  
[Install]  
WantedBy=default.target
```

## 3. Enable dan start

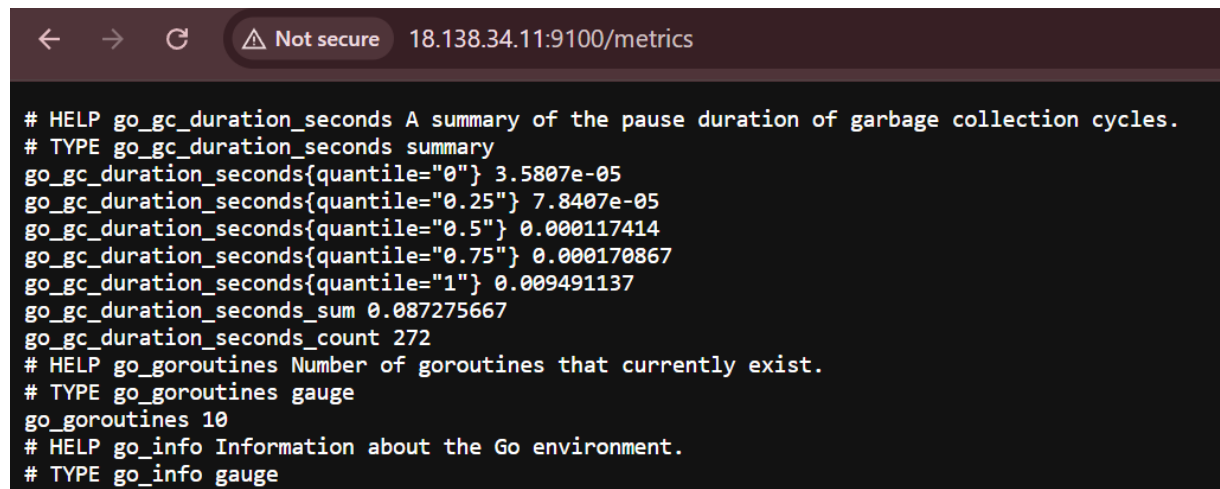
```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable node_exporter
```

```
sudo systemctl start node_exporter
```

## 4. Testing



```
< > ↻ ⚠ Not secure 18.138.34.11:9100/metrics  
  
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.  
# TYPE go_gc_duration_seconds summary  
go_gc_duration_seconds{quantile="0"} 3.5807e-05  
go_gc_duration_seconds{quantile="0.25"} 7.8407e-05  
go_gc_duration_seconds{quantile="0.5"} 0.000117414  
go_gc_duration_seconds{quantile="0.75"} 0.000170867  
go_gc_duration_seconds{quantile="1"} 0.000491137  
go_gc_duration_seconds_sum 0.007275667  
go_gc_duration_seconds_count 272  
# HELP go_goroutines Number of goroutines that currently exist.  
# TYPE go_goroutines gauge  
go_goroutines 10  
# HELP go_info Information about the Go environment.  
# TYPE go_info gauge  
go_info{version="1.16.6"} 1
```

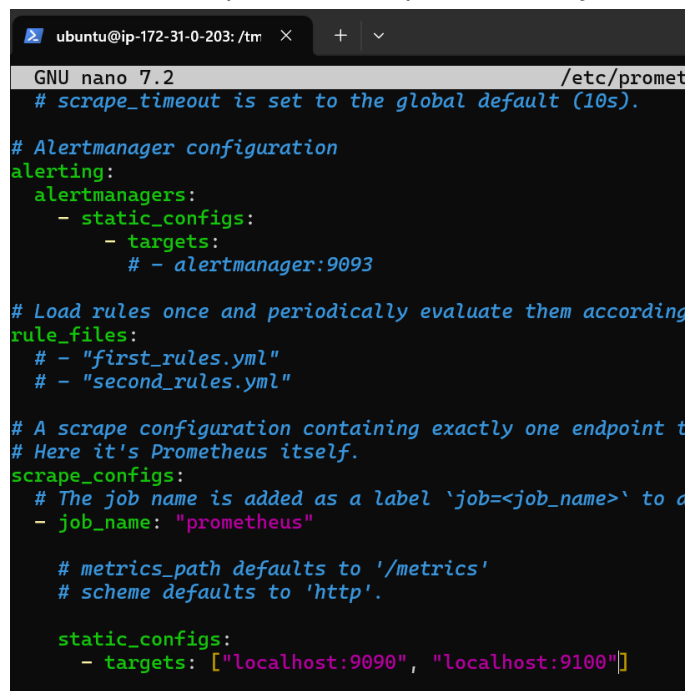
### STEP 3

#### 1. Download Promentheus

```
cd /tmp
wget
https://github.com/prometheus/prometheus/releases/download/v2.53.0/
prometheus-2.53.0.linux-amd64.tar.gz
tar xvf prometheus-*.tar.gz
sudo mv prometheus-* /etc/Prometheus
```

#### 2. Konfigurasi Promentheus

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

A screenshot of a terminal window with a dark background. The title bar shows 'ubuntu@ip-172-31-0-203: /tmp'. The terminal shows the GNU nano 7.2 editor editing the file /etc/prometheus/prometheus.yml. The content of the file is as follows:

```
# scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the
rule_files:
# - "first_rules.yml"
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to all metrics
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090", "localhost:9100"]
```

#### 3. Systemd promentheus

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

copy ini:

[Unit]

Description=Prometheus

After=network.target

[Service]

ExecStart=/etc/prometheus/prometheus \

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

[Install]

WantedBy=multi-user.target

```
ubuntu@ip-172-31-0-203: /tm × + ▾
GNU nano 7.2 /et
[Unit]
Description=Prometheus
After=network.target

[Service]
ExecStart=/etc/prometheus/prometheus \
--config.file=/etc/prometheus/prometheus.yml

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

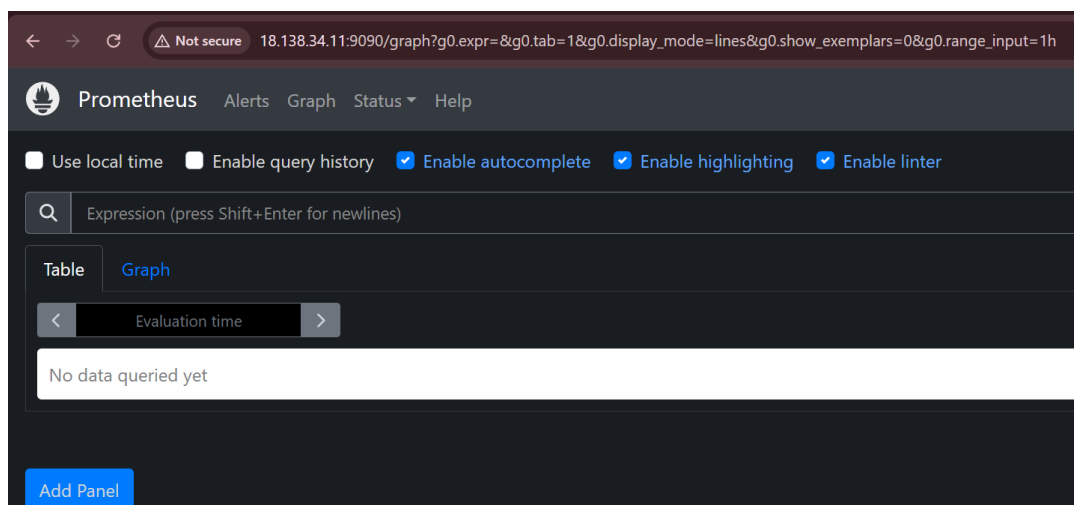
4. Start

```
sudo systemctl daemon-reload
sudo systemctl enable prometheus
sudo systemctl start Prometheus
```

5. Test  
Jangan lupa allow security group dulu:

▼ Inbound rules

Filter rules					< 1 >	
Name	Security group rule ID	Port range	Protocol			
-	sgr-0f1195d040d659e0e	80	TCP			
-	sgr-035c1bdcb6fb150a4	22	TCP			
-	sgr-05e84002f6817ea17	9100	TCP			
-	sgr-04493c49fd932a968	9090	TCP			



## STEP 4

### 1. Install Grafana

```
sudo apt install -y apt-transport-https 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  
sudo apt install grafana -y
```

### 2. Start

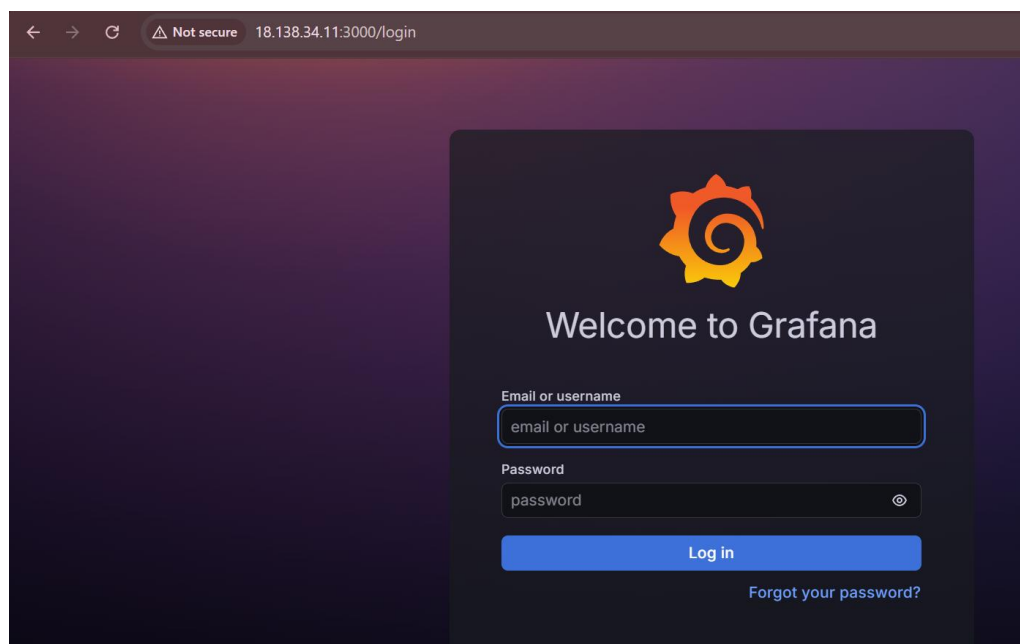
```
sudo systemctl enable grafana-server  
  
sudo systemctl start grafana-server
```

### 3. Akses

Allow security group 3000

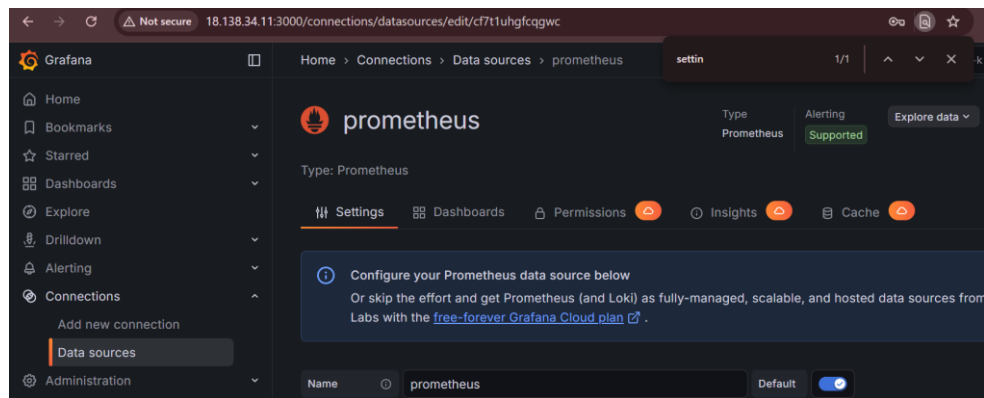
Login : admin

Password : admin

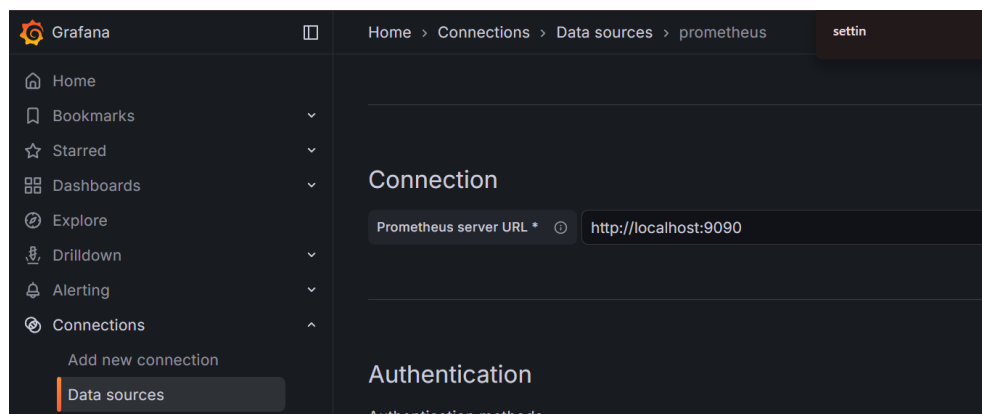


### 4. Tambah data source

Setting – data source – Prometheus

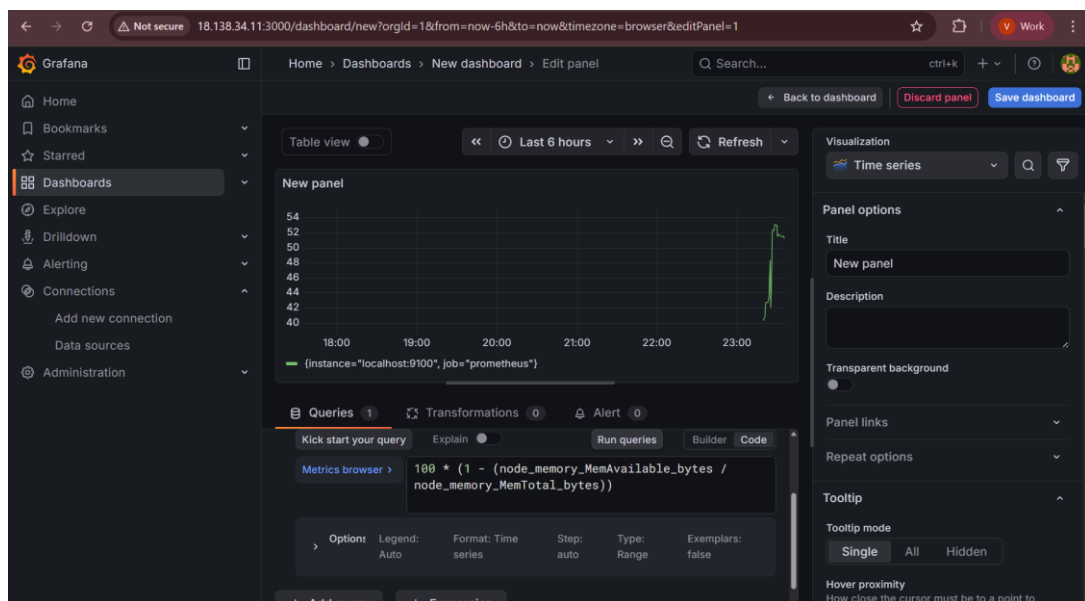


localhost:9090



## 5. Dashboard memory usage

$100 * (1 - (\text{node\_memory\_MemAvailable\_bytes} / \text{node\_memory\_MemTotal\_bytes}))$



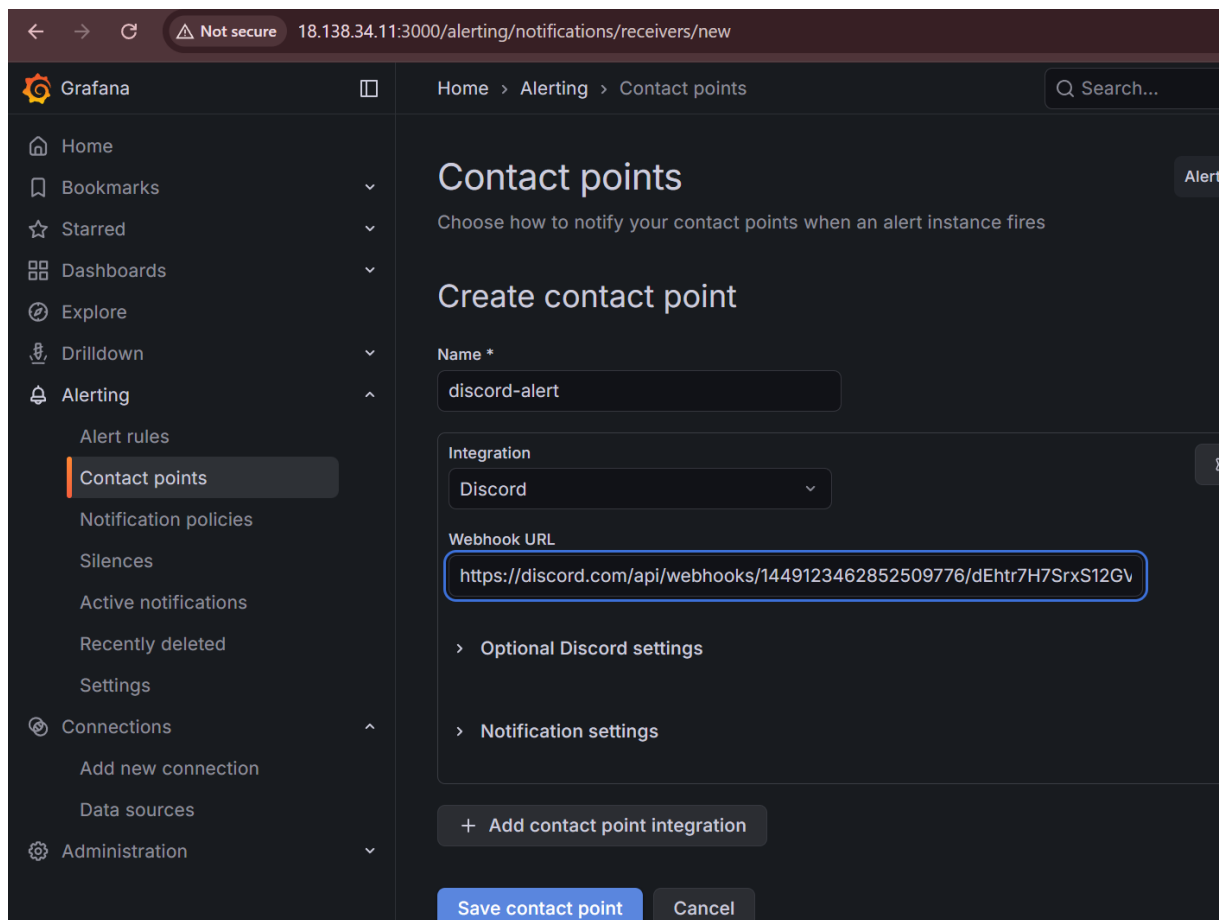
STEP 5

## 1. Alert memory – discord

Buat Discord Webhook

[https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7\\_24\\_OFJrUMceKu](https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7_24_OFJrUMceKu)

## 2. Buat Contact Point Discord di Grafana



The screenshot shows the Grafana web interface for configuring a new contact point. The browser address bar indicates the URL: 18.138.34.11:3000/alerting/notifications/receivers/new. The left sidebar shows the 'Alerting' menu with 'Contact points' selected. The main panel is titled 'Contact points' and 'Create contact point'. The form includes a 'Name' field with the value 'discord-alert', an 'Integration' dropdown set to 'Discord', and a 'Webhook URL' field containing the Discord webhook URL. Below these fields are expandable sections for 'Optional Discord settings' and 'Notification settings'. At the bottom, there is a '+ Add contact point integration' button and 'Save contact point' and 'Cancel' buttons.

← → ↻ Not secure 18.138.34.11:3000/alerting/notifications/receivers/new

Grafana

Home > Alerting > Contact points

Search...

### Contact points

Choose how to notify your contact points when an alert instance fires

#### Create contact point

Name \*

discord-alert

Integration

Discord

Webhook URL

[https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7\\_24\\_OFJrUMceKu](https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7_24_OFJrUMceKu)

> Optional Discord settings

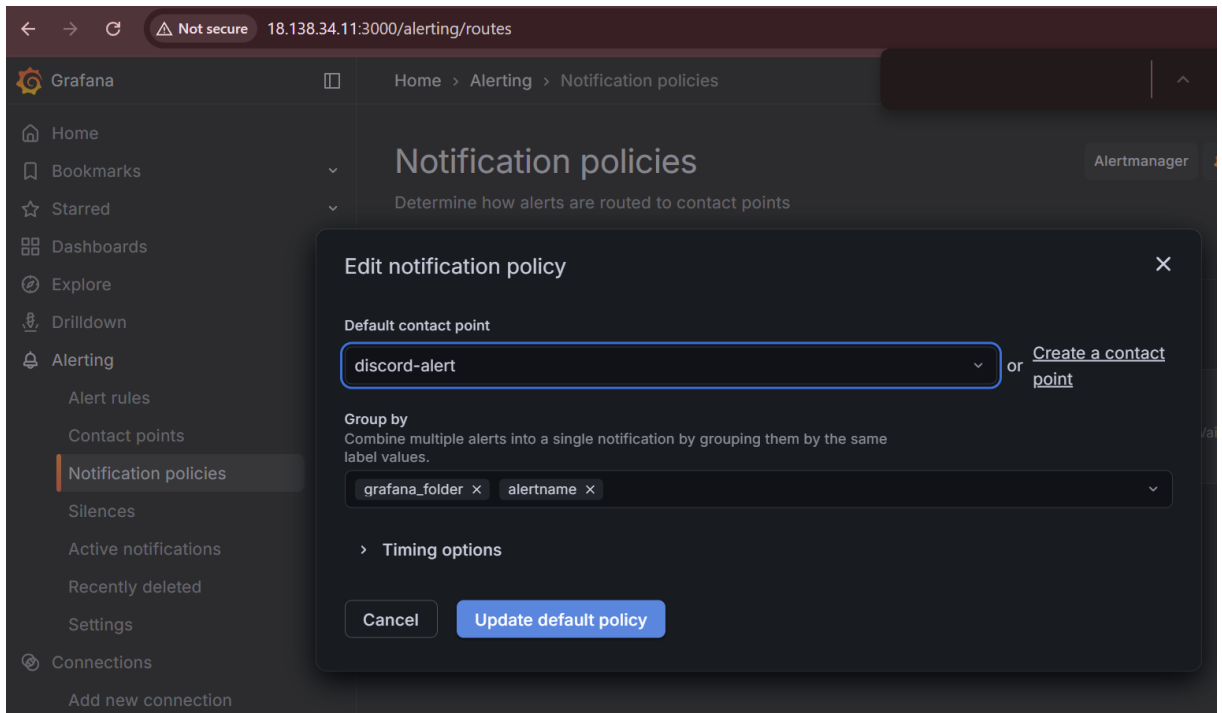
> Notification settings

+ Add contact point integration

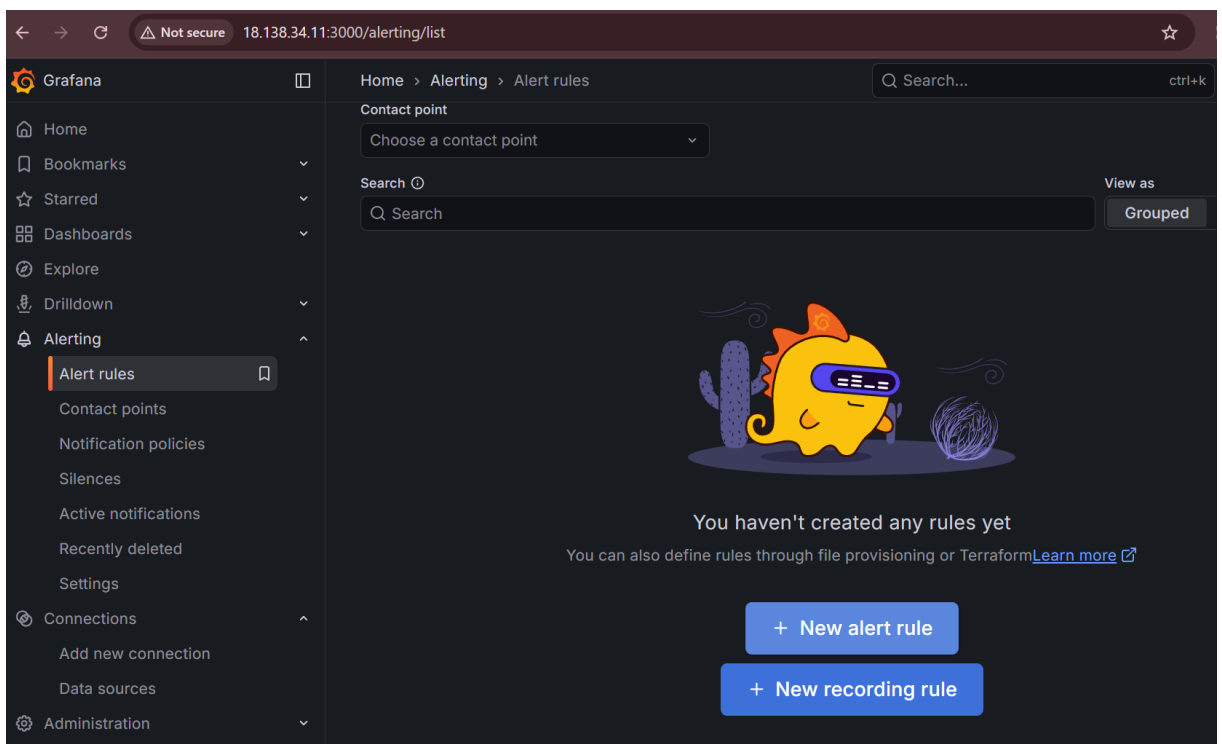
Save contact point Cancel

## 2. Buat Notification Policy





### 3. Buat Alert Rule Memory Usage



$100 * (1 - (\text{node\_memory\_MemAvailable\_bytes} / \text{node\_memory\_MemTotal\_bytes}))$

The screenshot shows the Grafana Alerting interface for creating a new alert rule. The left sidebar contains navigation links: Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting (selected), Alert rules (selected), Contact points, Notification policies, Silences, Active notifications, Recently deleted, Settings, Connections, Add new connection, Data sources, and Administration. The main area is titled '2. Define query and alert condition' with a sub-header 'Define query and alert condition'. It features a 'prometheus' data source selector, a '10m' time range, and a 'Run queries' button. The query editor shows the expression: `100 * (1 - (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes))`. Below the query, the 'Options' section includes 'Legend' (Auto), 'Min step' (auto), 'Format' (Time series), and 'Type' (Range, Instant). A 'Table' section displays a single row with labels `{instance="localhost:9100", job="prometheus"}` and a value of 52.7. The 'Alert condition' section is set to 'WHEN QUERY IS ABOVE 0'. A 'Preview alert rule condition' button is at the bottom.

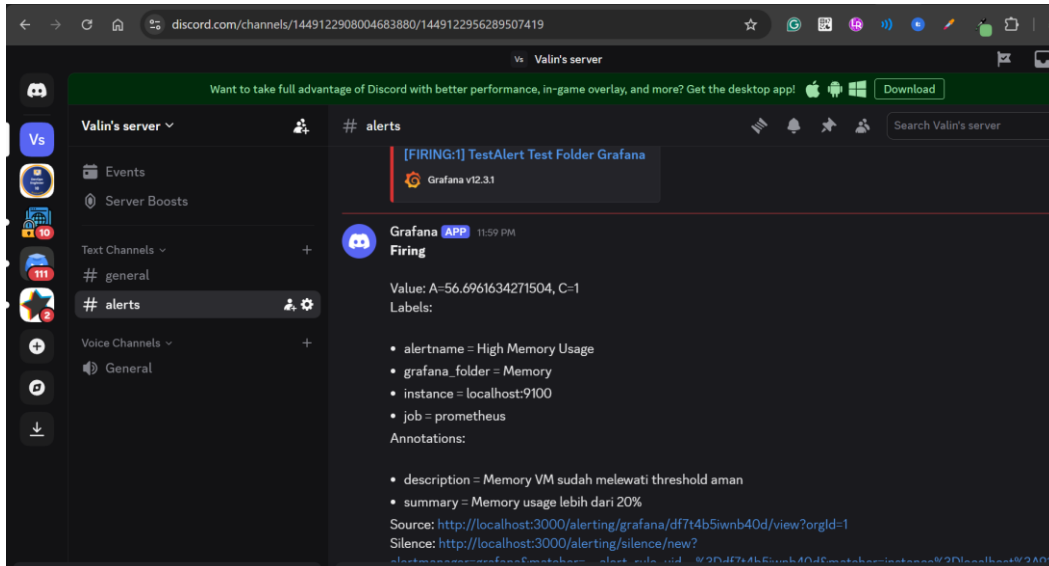
This close-up view shows the 'Alert condition' configuration. It includes a 'Table' section with the label `{instance="localhost:9100", job="prometheus"}`. The 'Alert condition' section is set to 'WHEN QUERY IS ABOVE' with a value of '20' entered in a text box. Below this, the same label `{instance="localhost:9100", job="prometheus"}` is shown. A 'Preview alert rule condition' button is located at the bottom.

The screenshot shows the '6. Configure notification message' step. It includes a 'Summary (optional)' section with a text area containing 'Memory usage lebih dari 20%'. Below this is a 'Description (optional)' section with a text area containing 'Memory VM sudah melewati threshold aman'. At the bottom, there is a 'Runbook URL (optional)' section with a text area.

#### 4. Test Alert

`sudo apt install stress -y`

`stress --vm 1 --vm-bytes 300M --timeout 60s`



#### STEP 6

Monitoring webserver dengan uptime kuma

##### 1. install docker

`sudo apt update`

`sudo apt upgrade -y`

`sudo apt install docker.io -y`

##### 2. Start dan enable docker

`sudo systemctl start docker`

`sudo systemctl enable docker`

Cek : `systemctl status docker`

##### 3. install uptime kuma dengan docker

`sudo docker run -d \`

`--restart=always \`

`-p 3001:3001 \`

-v uptime-kuma:/app/data \

--name uptime-kuma \

louislam/uptime-kuma

```
ubuntu@ip-172-31-0-203:/tmp$ sudo docker run -d --restart=always -p 3001:3001 -v uptime-kuma:/app/data --name uptime-kuma louislam/uptime-kuma
Unable to find image 'louislam/uptime-kuma:latest' locally
latest: Pulling from louislam/uptime-kuma
b338562f40a7: Pull complete
874bf4d93720: Pull complete
b16337721583: Pull complete
```

Pastikan container running : docker ps

3. allow port 3001 di security group

Inbound rule 3 Delete

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>
-	Custom TCP	TCP
Port range <a href="#">Info</a>	Source type <a href="#">Info</a>	Source <a href="#">Info</a>
3001	Anywhere-IPv4	0.0.0.0/0
		<span>0.0.0.0/0 X</span>

5. buka uptime kuma dengan port 3001 dan add new monitor

18.138.34.11:3001/add

## Add New Monitor

[+ Add New Monitor](#)

Select

Menu Status Active Tags

No Monitors, please [add one](#)

### General

Monitor Type: HTTP(s)

Friendly Name: Test Service Availability

URL: http://18.138.34.11

Heartbeat Interval (Check every 20 seconds): 20

Retries: 0

Maximum retries before the service is marked as down and a notification is sent

[Save](#)

### Notifications

Not available, please setup.

[Setup Notification](#)

### Proxy

Not available, please setup.

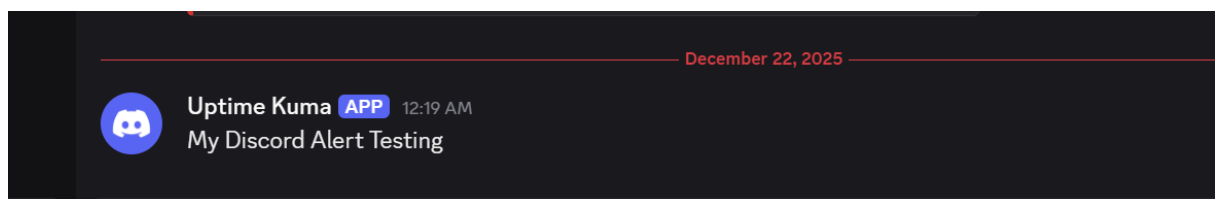
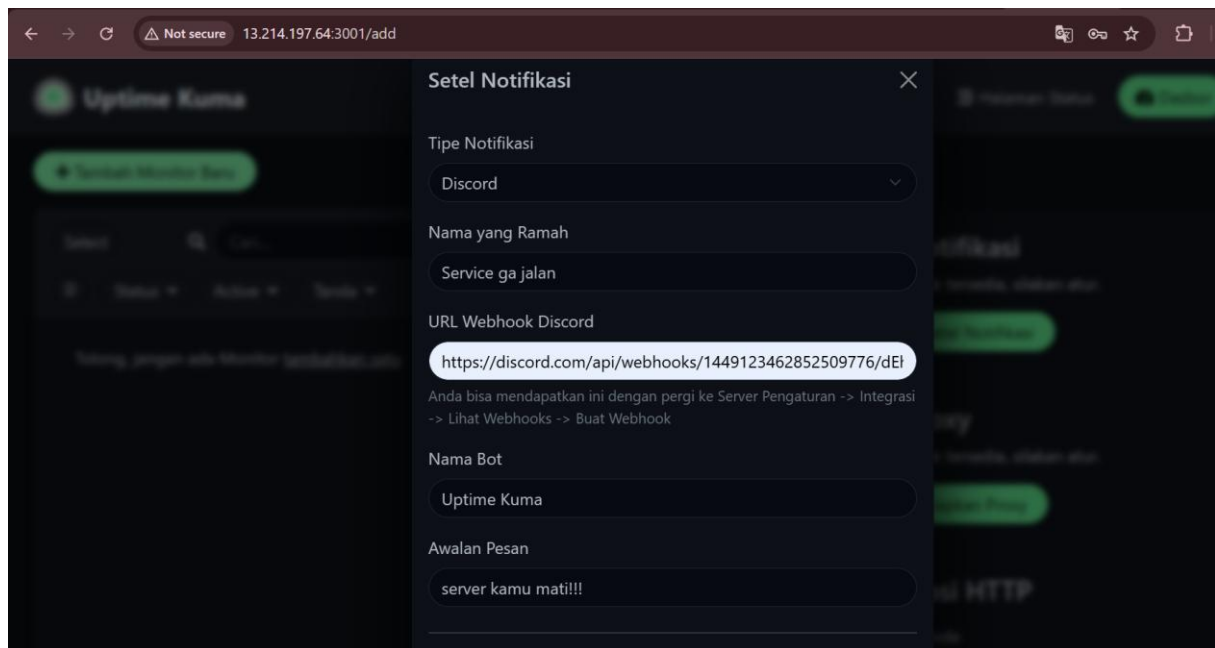
[Setup Proxy](#)

### HTTP Options

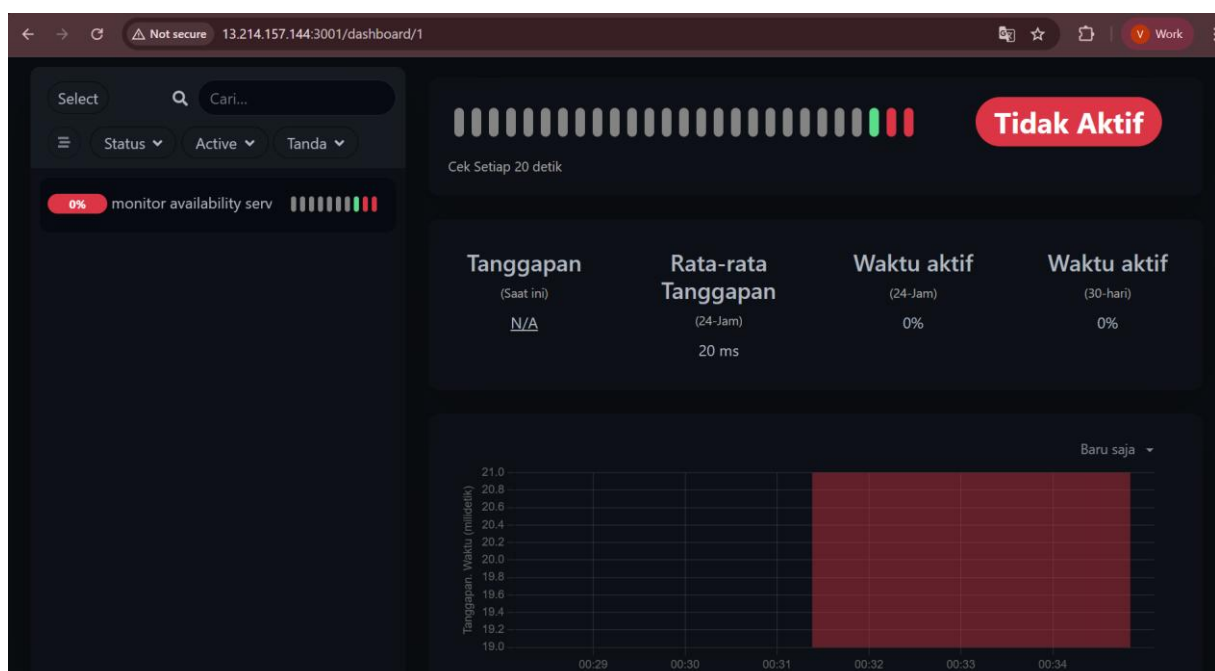
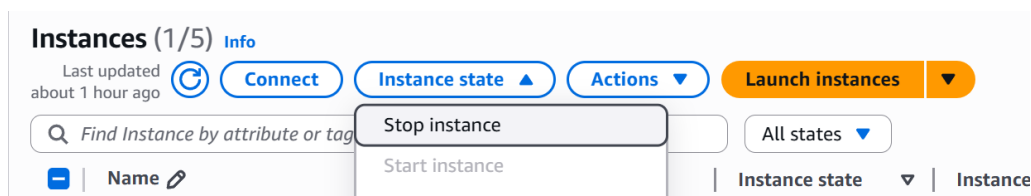
Method: GET

Body Encoding: JSON

Dan setup notification



Kita tes dengan stop instance :





Uptime Kuma APP 12:34 AM

server kamu mati!!!

✖ Your service monitor availability service went down. ✖

**Service Name**

monitor availability service

**Service URL**

<http://13.214.197.64/>

**Time (Europe/Berlin)**

2025-12-22 00:34:07

**Error**

timeout of 8000ms exceeded

Today at 12:34 AM