

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

- Konfigurasikan 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

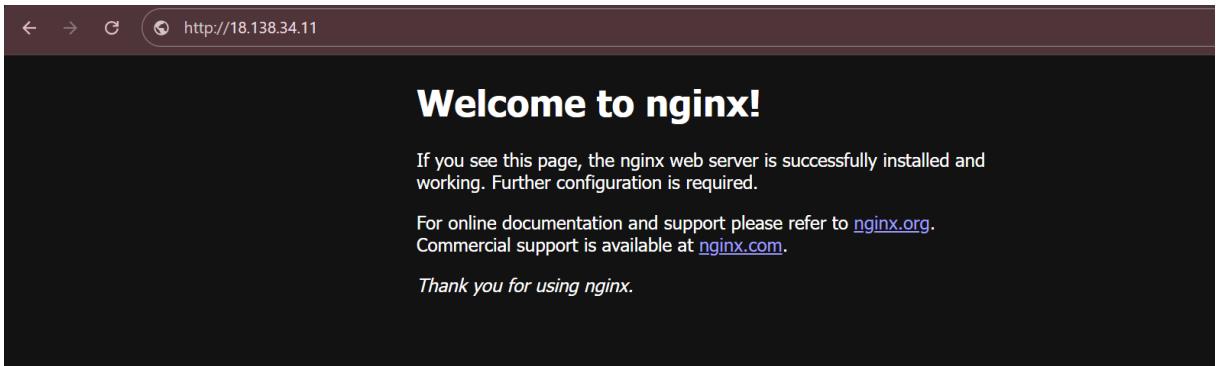
```
ubuntu@ip-172-31-0-203:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Sun 2025-12-21 22:00:12 UTC; 51s ago
```

Allow port 80 di security group

The screenshot shows the 'Inbound rules' section of a security group configuration. There is one existing rule named 'Inbound rule 1'. The rule details are as follows:

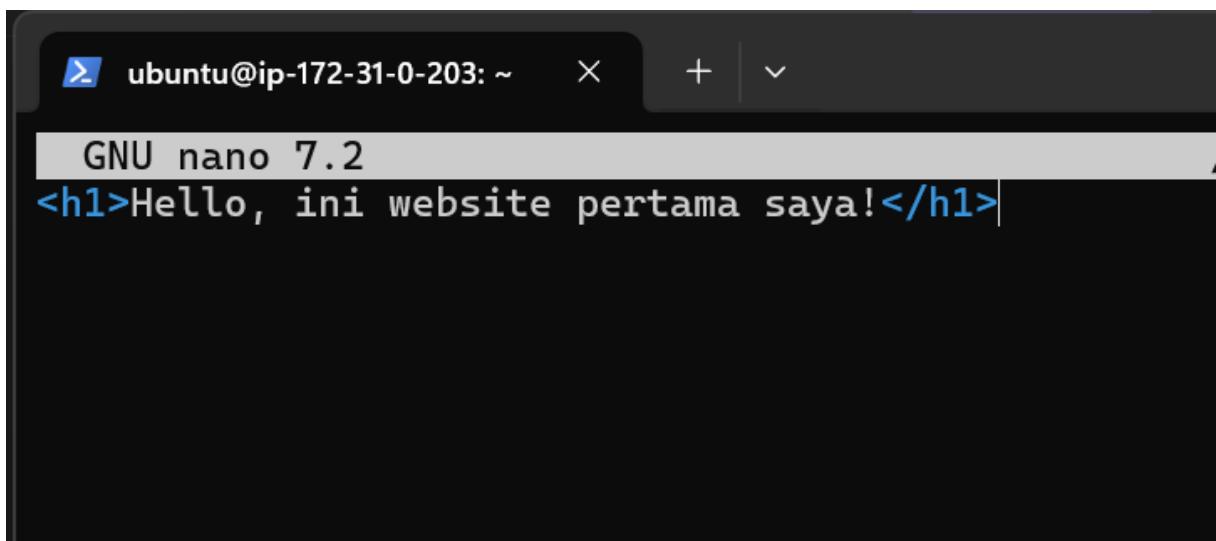
- Security group rule ID:** sgr-0f1195d040d659e0e
- Type:** HTTP
- Protocol:** TCP
- Port range:** 80
- Source type:** Custom
- Source:** 0.0.0.0/0

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
```

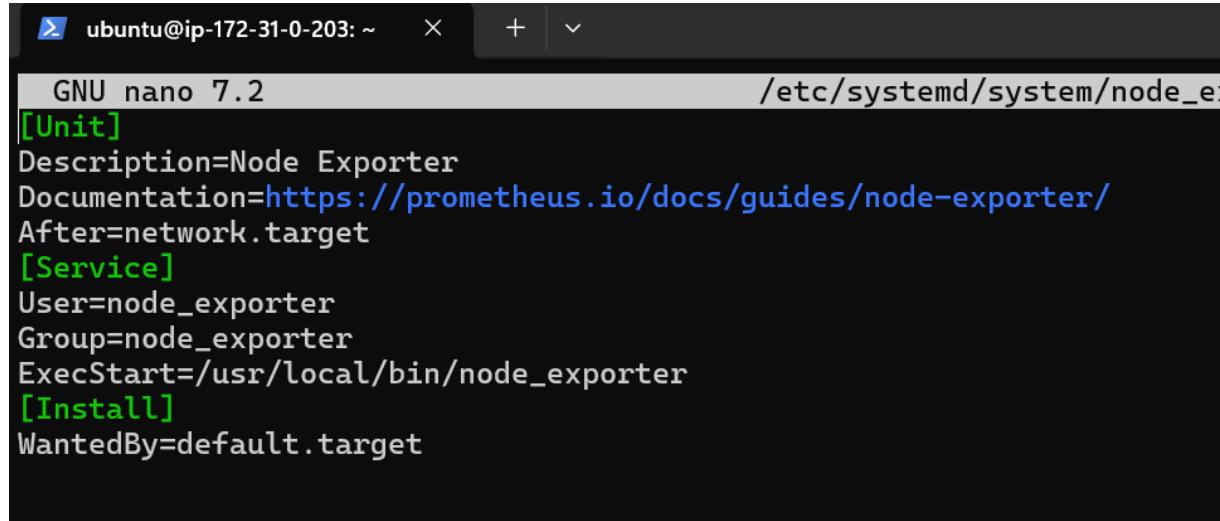
```
https://github.com/prometheus/node_exporter/releases/download/v1.8.1/no  
de_exporter-1.8.1.linux-amd64.tar.gz
```

```
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
```

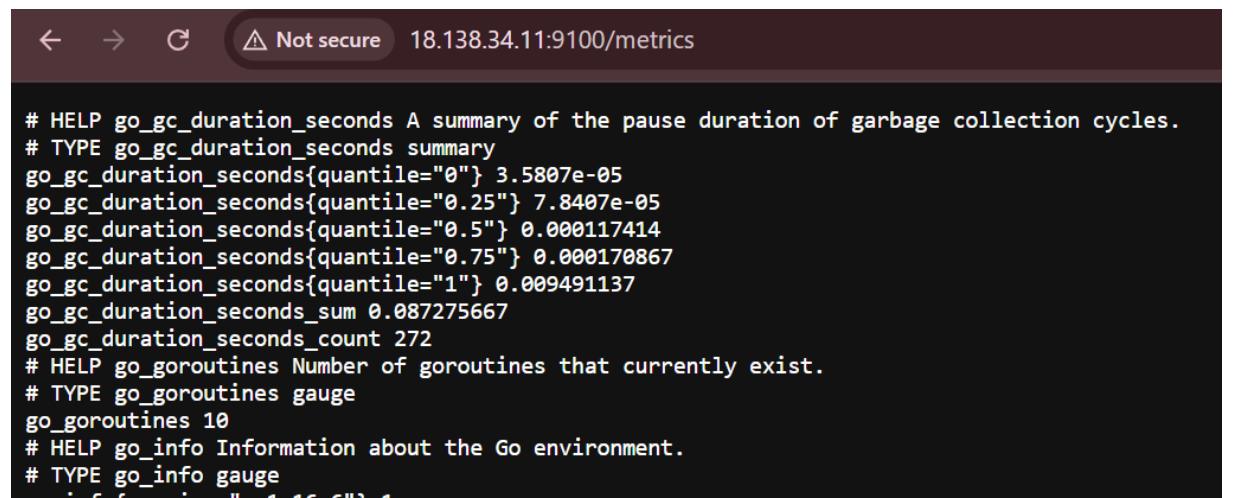


```
GNU nano 7.2                               /etc/systemd/system/node_e
[Unit]
Description=Node Exporter
Documentation=https://prometheus.io/docs/guides/node-exporter/
After=network.target
[Service]
User=node_exporter
Group=node_exporter
ExecStart=/usr/local/bin/node_exporter
[Install]
WantedBy=default.target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

3. Enable dan start

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable node_exporter
sudo systemctl start node_exporter
```

4. Testing



```
# 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"} 0.009491137
go_gc_duration_seconds_sum 0.087275667
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_golang_version="1.16.6"
```

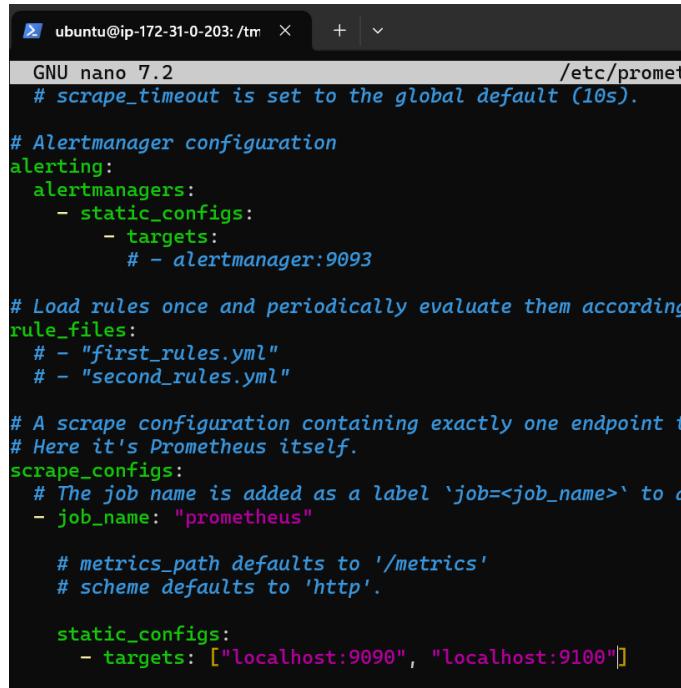
STEP 3

1. Download Prometheus

```
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 Prometheus

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



```
ubuntu@ip-172-31-0-203:/tmp ~ + | v  
GNU nano 7.2 /etc/prometheus.yml  
# 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 schedule.  
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 the metrics.  
  - job_name: "prometheus"  
  
    # metrics_path defaults to '/metrics'  
    # scheme defaults to 'http'.  
  
    static_configs:  
      - targets: ["localhost:9090", "localhost:9100"]
```

3. Systemd prometheus

```
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: /tmp %
GNU nano 7.2 /etc/systemd/system/multi-user.target.wants/prometheus.service

[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

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

Prometheus

Not secure 18.138.34.11:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h

Alerts Graph Status Help

Use local time Enable query history Enable autocomplete Enable highlighting Enable linter

Expression (press Shift+Enter for newlines)

Table Graph

Evaluation time < >

No data queried yet

Add Panel

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

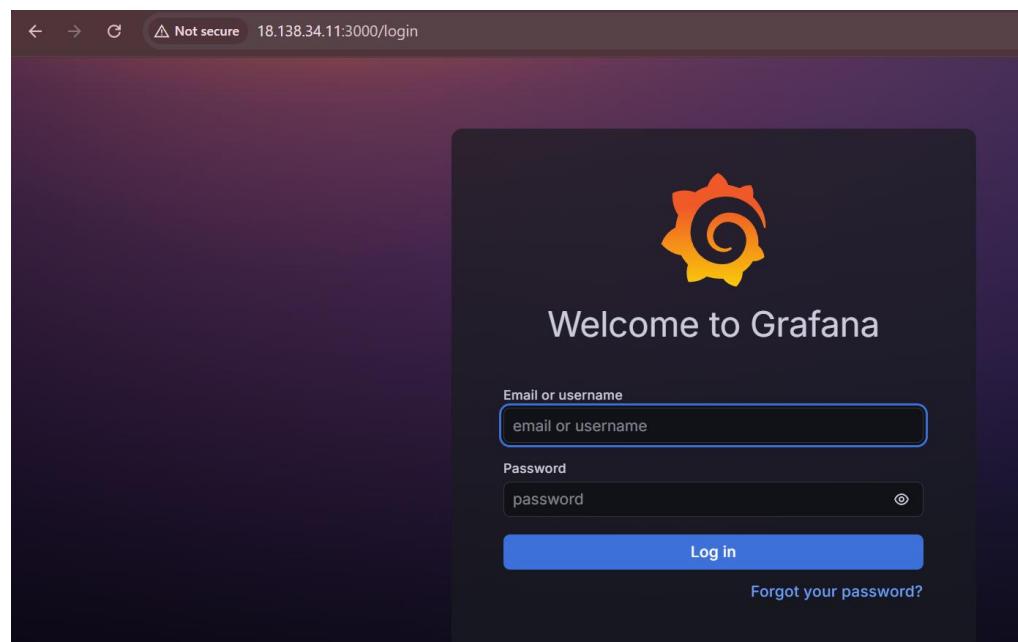
```
sudo systemctl enable grafana-server  
sudo systemctl start grafana-server
```

3. Akses

Allow security group 3000

Login : admin

Password : admin



4. Tambah data source

Setting – data source – Prometheus

The screenshot shows the Grafana interface for managing data sources. A connection to a 'prometheus' data source has been configured. The 'Type' is set to 'Prometheus' and the status is 'Supported'. The configuration panel displays instructions to configure the Prometheus data source and mentions skipping effort for managed services. The data source is named 'prometheus' and is marked as the 'Default' source.

localhost:9090

This screenshot shows the 'Connection' configuration step within the Grafana data source setup. The 'Prometheus server URL' field is populated with 'http://localhost:9090'. The 'Authentication' section is visible below it.

5. Dashboard memory usage

$100 * (1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes}))$

The screenshot shows the Grafana Metrics browser interface. A single query is defined: $100 * (1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes}))$. The visualization is set to 'Time series'. The results show a sharp increase in the metric value starting around 23:00, indicating a significant drop in available memory.

STEP 5

1. Alert memory – discord

Buat Discord Webhook

https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7_24_OFJrUMceKu

2. Buat Contact Point Discord di Grafana

The screenshot shows the Grafana web interface with a dark theme. The left sidebar has a navigation menu with items like Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Alert rules, Contact points (which is selected and highlighted in orange), Notification policies, Silences, Active notifications, Recently deleted, Settings, Connections, Add new connection, Data sources, and Administration. The main content area is titled "Contact points" and contains instructions: "Choose how to notify your contact points when an alert instance fires". Below this is a section titled "Create contact point" with fields for "Name *" (set to "discord-alert"), "Integration" (set to "Discord"), and "Webhook URL" (set to "https://discord.com/api/webhooks/1449123462852509776/dEhtr7H7SrxS12GVeVtufNBRXp2W4UesxMutxgTRSBsvgDQMDemTy7_24_OFJrUMceKu"). There are also links for "Optional Discord settings" and "Notification settings". At the bottom are "Save contact point" and "Cancel" buttons.

2. Buat Notification Policy

Not secure 18.138.34.11:3000/alerting/routes

Grafana Home > Alerting > Notification policies

Notification policies

Determine how alerts are routed to contact points

Alertmanager

Edit notification policy

Default contact point

discord-alert or [Create a contact point](#)

Group by

Combine multiple alerts into a single notification by grouping them by the same label values.

grafana_folder x alertname x

Timing options

Cancel Update default policy

Home Bookmarks Starred Dashboards Explore Drilldown Alerting Alert rules Contact points **Notification policies** Silences Active notifications Recently deleted Settings Connections Add new connection

3. Buat Alert Rule Memory Usage

Not secure 18.138.34.11:3000/alerting/list

Grafana Home > Alerting > Alert rules

Search... ctrl+k

Contact point

Choose a contact point

Search ⓘ View as Grouped

You haven't created any rules yet

You can also define rules through file provisioning or Terraform [Learn more](#)

+ New alert rule + New recording rule

Home Bookmarks Starred Dashboards Explore Drilldown Alerting **Alert rules** Contact points Notification policies Silences Active notifications Recently deleted Settings Connections Add new connection Data sources Administration

$100 * (1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes}))$

The screenshot shows the Grafana Alerting interface at the URL 18.138.34.11:3000/alerting/new/alerting. The left sidebar is open, showing the 'Alert rules' section selected. The main area is titled '2. Define query and alert condition'. A Prometheus query is defined: `100 * (1 - (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes))`. The 'Options' section shows 'Legend: Auto', 'Min step: auto', 'Format: Time series', and 'Type: Range'. Below the query is a table with one row: `{instance="localhost:9100", job="prometheus"}` with a value of 52.77. The 'Alert condition' section shows 'WHEN QUERY IS ABOVE 0'. Another table row is shown: `{instance="localhost:9100", job="prometheus"}` with a value of 1. A blue button at the bottom right says 'Preview alert rule condition'.

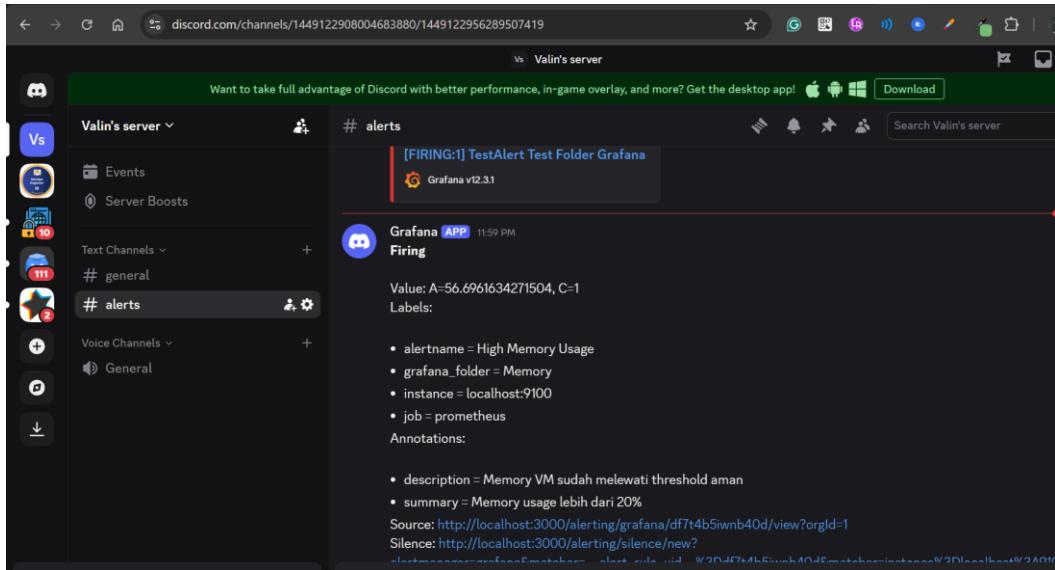
This is a zoomed-in view of the alert rule configuration. It shows the 'Table' section with the query `{instance="localhost:9100", job="prometheus"}`. The 'Alert condition' section shows 'WHEN QUERY IS ABOVE 20'. A blue box highlights the threshold value '20'. Below it is another table row: `{instance="localhost:9100", job="prometheus"}`. At the bottom is a blue button labeled 'Preview alert rule condition'.

The screenshot shows the '6. Configure notification message' step. It includes a summary input field containing 'Memory usage lebih dari 20%' and a description input field containing 'Memory VM sudah melewati threshold aman'. Both fields have blue borders. At the bottom, there is a 'Runbook URL (optional)' input field with the placeholder 'Webpage where you keep your runbook for the alert.'.

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
874bfb4d93720: Pull complete
b16337721583: Pull complete
```

Pastikan container running : docker ps

3. allow port 3001 di security group

Inbound rule 3

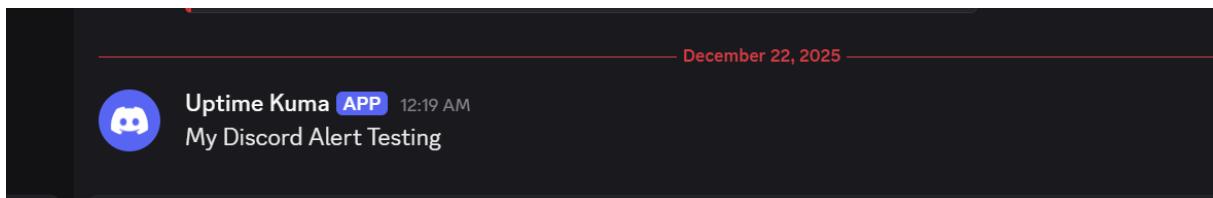
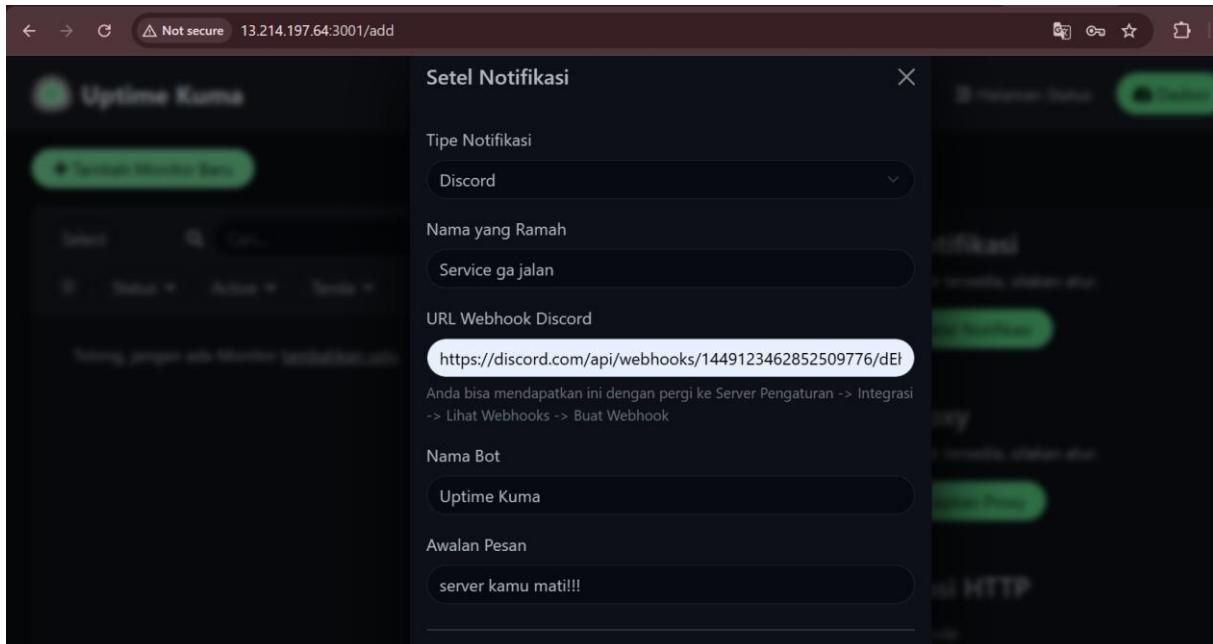
Security group rule ID	Type Info	Protocol Info
-	Custom TCP	TCP
Port range Info	Source type Info	Source Info
3001	Anywhere-IPv4	0.0.0.0/0 X

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

The screenshot shows the Uptime Kuma web interface with the following details:

- Header:** Not secure, 18.138.34.11:3001/add
- Left Sidebar:** Shows a list of monitors with filters for Select, Status (Active), and Tags.
- Center:**
 - Add New Monitor:** A green button.
 - General:** Monitor Type: HTTP(s), Friendly Name: Test Service Availability, URL: http://18.138.34.11, Heartbeat Interval: 20 seconds, Retries: 0.
 - Notifications:** Not available, please setup. Setup Notification button.
 - Proxy:** Not available, please setup. Setup Proxy button.
 - HTTP Options:** Method: GET, Body Encoding: JSON.
- Bottom:** Save button.

Dan setup notification



Kita tes dengan stop instance :

The screenshot shows the 'Instances (1/5)' page. The 'Actions' dropdown menu is open, with 'Stop instance' highlighted. Other options include 'Start instance' and 'Launch instances'. There is also a search bar for finding instances by attribute or tag.

