

Homework Sesi 12 Bootcamp DevOps Engineer DigitalSkola

1. Buat 1 Monitoring CPU & Memory Usage Menggunakan tools Grafana untuk visualisasinya, Prometheus & node exporter untuk mengambil metriknya.
2. Create alert cpu/memory usage dan notification dikirim ke discord.

Pada homework sesi 11 node exporter dan prometheus telah diinstall. Maka dari itu, sesi ini akan diinstall grafana terlebih dahulu.

Instalasi Grafana (Visualisasi Data)

Tujuan Grafana: Grafana digunakan untuk memvisualisasikan data metrik dari berbagai sumber data, termasuk Prometheus.

Langkah-langkah Instalasi Grafana:

1. Menambahkan Repository Grafana

```
ubuntu_admin@server2:~$ sudo apt-get install -y apt-transport-https
ubuntu_admin@server2:~$ sudo apt-get install -y software-properties-common
ubuntu_admin@server2:~$ sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
ubuntu_admin@server2:~$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee /etc/apt/sources.list.d/grafana.list
```

Perintah-perintah ini menambahkan repository Grafana ke sistem Debian/Ubuntu.

- **sudo apt-get install -y apt-transport-https, sudo apt-get install -y software-properties-common:** Menginstal paket pendukung.
- **sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -:** Mengunduh dan menambahkan kunci GPG Grafana untuk verifikasi paket.
- **echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee /etc/apt/sources.list.d/grafana.list:** Menambahkan repository Grafana ke daftar sumber paket apt.

2. Memperbarui Daftar Paket dan Menginstal Grafana

```
ubuntu_admin@server2:~$ sudo apt-get update
ubuntu_admin@server2:~$ sudo apt-get install grafana
```

- **sudo apt-get update:** Memperbarui daftar paket dari repository yang dikonfigurasi.
- **sudo apt-get install grafana:** Menginstal paket grafana.

3. Memulai dan Mengaktifkan Grafana

```
ubuntu_admin@server2:~$ sudo systemctl start grafana-server
```

```
ubuntu_admin@server2:~$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /
lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.se
rvice → /lib/systemd/system/grafana-server.service.

ubuntu_admin@server2:~$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; ve
   Active: active (running) since Thu 2025-03-06 09:34:52 UTC; 18s ago
```

- **sudo systemctl start grafana-server:** Memulai layanan Grafana Server.
- **sudo systemctl enable grafana-server:** Mengaktifkan autostart Grafana Server.
- **sudo systemctl status grafana-server:** Memeriksa status Grafana Server.

4. Mengakses Web UI Grafana

Buka browser dan akses:

http://<IP_SERVER>:3000

5. Konfigurasi Grafana Data Source (Prometheus)

- Arahkan ke **Connections** → **Data Sources**.
- Klik **Add new data source**.
- Cari dan pilih **Prometheus**.
- Beri nama yang diinginkan, contoh **prometheus-bootcamp-devops-engineer**.
- Di **Settings**, atur **URL** Prometheus menjadi **http://<IP_SERVER>:9090**. Namun, karena Prometheus berada di server yang sama dapat memakai **http://localhost:9090**.
- Klik **Save & Test**. Pastikan muncul pesan "Data source is working".

Grafana dengan data source prometheus dengan nama **prometheus-bootcamp-devops-engineer** telah dikonfigurasi. Selanjutnya membuat dashboard monitoring CPU dan Memory Usage.

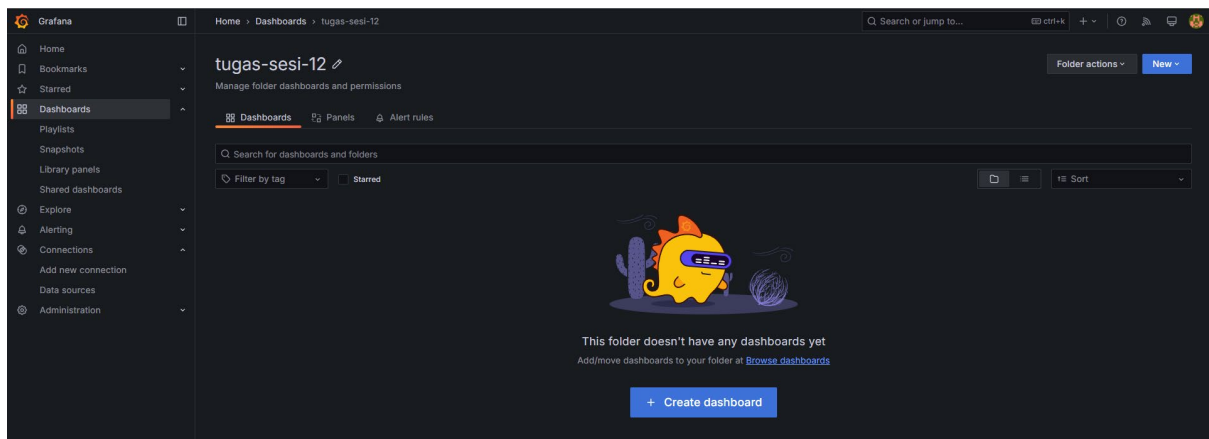
Membuat Dashboard Monitoring CPU & Memory Usage di Grafana

Tujuan: Membuat dashboard di Grafana yang menampilkan grafik penggunaan CPU dan memori server secara real-time.

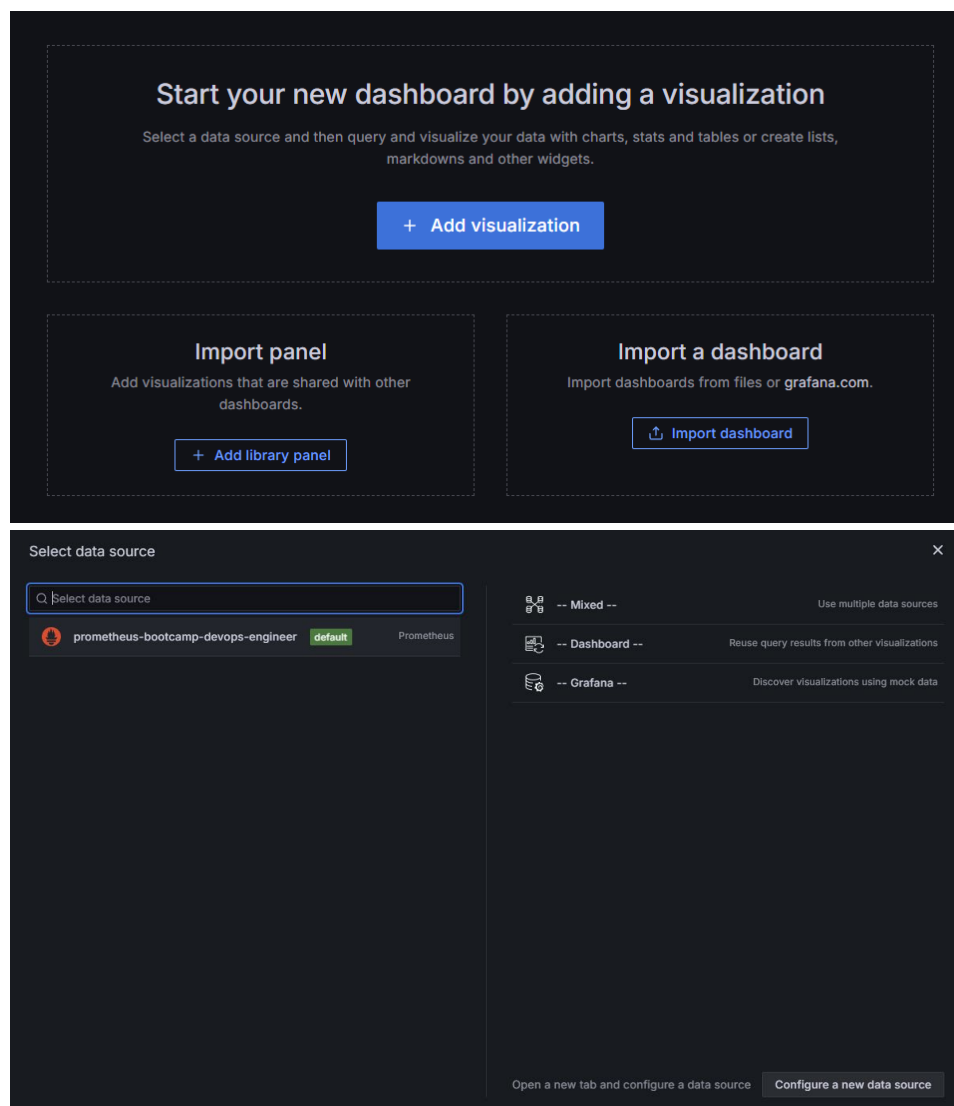
Langkah-langkah:

1. **Akses Grafana Web UI:** Buka browser web dan kunjungi **http://<IP_SERVER >:3000**. Login dengan username dan password Grafana.
2. **Buat Dashboard Baru:**
 - Pada sidebar kiri, klik Pilih "**Dashboard**".
 - Buat folder dengan nama tugas-sesi-12 (opsional)

- Terlihat dashboard kosong baru. Pilih "**Create dashboard**" untuk membuat dashboard baru.

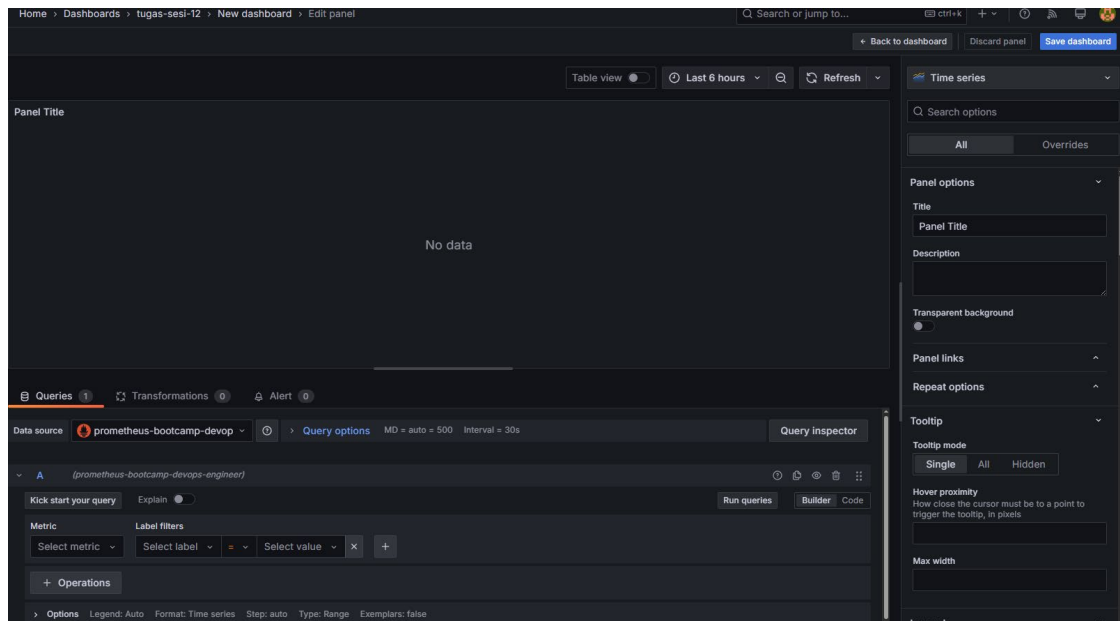


- **Add visualization** > pilih prometheus yang telah dimasukkan sebelumnya sebagai data source

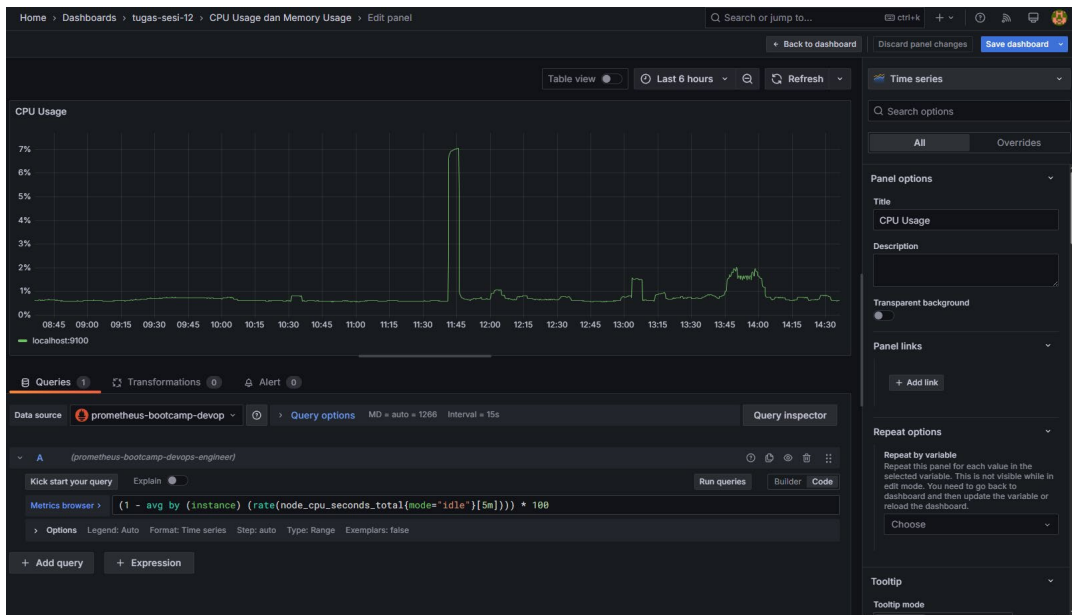


3. Konfigurasi Panel Grafik CPU Usage:

- Di panel query yang baru terbuka, pastikan Data source yang dipilih adalah **prometheus-bootcamp-devops-engineer**.



- Pada kolom "**Query**" (**A**), pilih "**Code**" (di sebelah "**Builder**") masukkan query PromQL berikut untuk menampilkan persentase penggunaan CPU:
 $(1 - \text{avg by (instance) (rate(node_cpu_seconds_total\{mode='idle'\}[5m])) * 100$
Penjelasan Query:
 - **node_cpu_seconds_total{mode="idle"}**: Metrik dari Node Exporter yang mencatat total detik CPU dalam mode idle.
 - **rate(...)**: Menghitung laju perubahan (per detik) dari metrik CPU idle dalam rentang waktu 5 menit terakhir ([5m]).
 - **avg by (instance) (...)**: Menghitung rata-rata laju perubahan CPU idle per instance (server).
 - **1 - (...)**: Mengurangkan rata-rata laju idle dari 1, untuk mendapatkan proporsi CPU yang tidak idle (yaitu, digunakan).
 - **100**: Mengalikan hasilnya dengan 100 untuk mendapatkan persentase.
- Pada sidebar kanan, pastikan "**Time series**" dipilih.
- Pada tab "**Panel option**", di bagian "**Title**", beri judul panel ini "**CPU Usage**".
- Di bagian "**Unit**", pilih "**Percent (%)**". Ini akan memastikan sumbu Y pada grafik menampilkan persentase.



- Panel untuk visualisasi CPU Usage telah dibuat. Pilih **"Back to dashboard"** untuk melanjutkan pembuatan Memory Usage

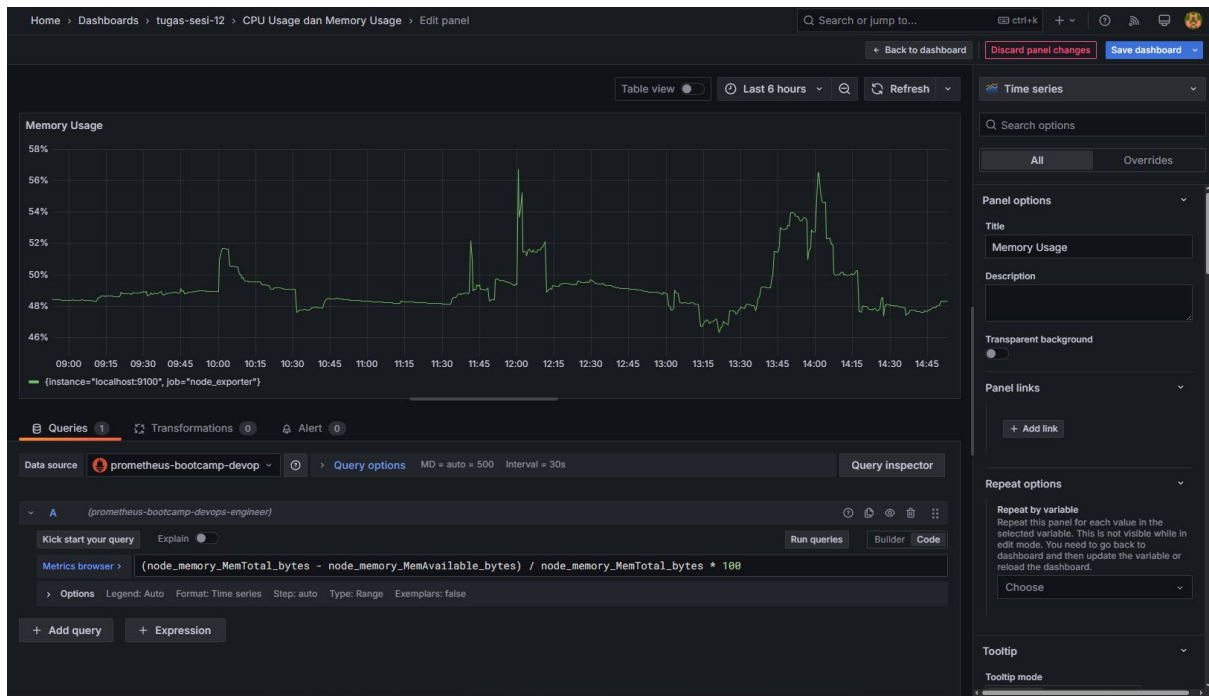
4. Konfigurasi Panel Grafik Memory Usage:

- Pilih dropdown **"Add"** → **"Visualization"**
- Pastikan Data source yang dipilih adalah **prometheus-bootcamp-devops-engineer**.
- Pada kolom **"Query" (A)**, masukkan query PromQL berikut untuk menampilkan persentase penggunaan memori:

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

Penjelasan Query:

- **node_memory_MemTotal_bytes**: Mengambil nilai metrik total memori.
 - **node_memory_MemAvailable_bytes**: Mengambil nilai metrik memori yang tersedia.
 - **-**: Operator pengurangan. **node_memory_MemTotal_bytes - node_memory_MemAvailable_bytes** menghitung perkiraan memori yang sedang digunakan (used memory).
 - **/**: Operator pembagian. Hasil pengurangan (used memory) dibagi dengan **node_memory_MemTotal_bytes** (total memory). Ini menghasilkan rasio *used memory* terhadap *total memory*.
 - **100**: Hasil rasio dikalikan 100 untuk mendapatkan persentase.
- Pada sidebar kanan, pastikan **"Time series"** dipilih.
 - Pada tab **"Panel option"**, di bagian **"Title"**, beri judul panel ini **"Memory Usage"**.
 - Di bagian **"Unit"**, pilih **"Percent (%)"**. Ini akan memastikan sumbu Y pada grafik menampilkan persentase.



- Panel untuk visualisasi CPU Usage telah dibuat. Pilih **“Save dashboard”** > isi title **“CPU Usage dan Memory Usage”**
- Dashboard telah selesai dibuat.



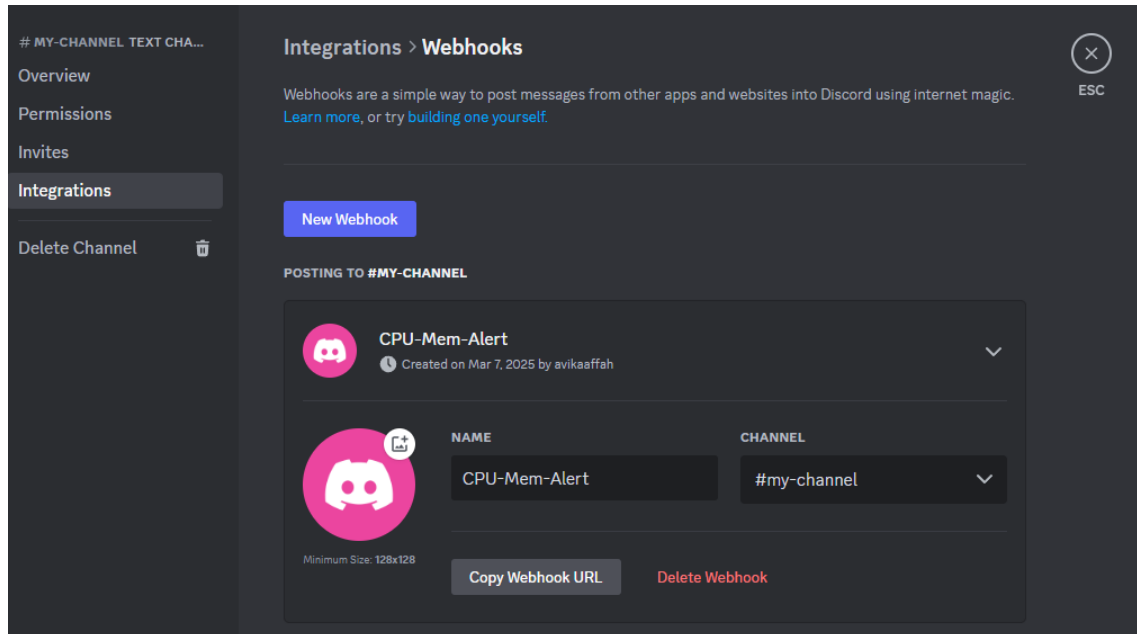
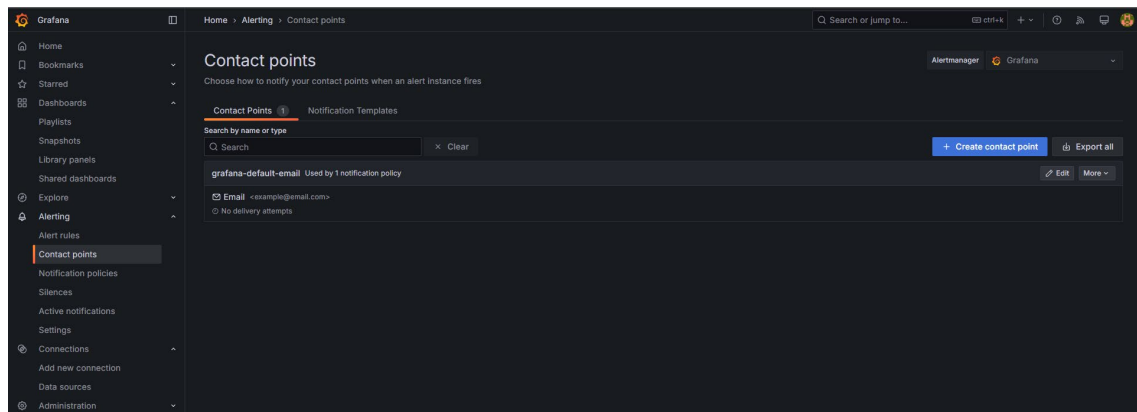
Membuat Alert CPU & Memory Usage di Grafana dan Notifikasi ke Discord

Tujuan: Membuat aturan alert di Grafana yang akan memicu alert jika penggunaan CPU atau memori melebihi batas yang ditentukan dan mengirimkan notifikasinya ke discord.

Langkah-langkah Instalasi Grafana:

1. Menambahkan Contact Point ke discord

- Pada sidebar kiri pilih **Alerting** → **Contact points**
- Pilih **+ Create contact point**
- Untuk menambahkan channel discord sebagai contact point, perlu webhook. Buka channel discord yang diinginkan untuk menerima alert. Pilih **Setting** → **Integrations** → **Webhooks** → **New Webhook** → **Copy Webhook URL**.



- Kembali ke grafana. Isi nama contact point. Lalu pilih Discord pada Integration → paste Webhook URL → Save contact point.

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

Create contact point

Name *

alert-cpu-mem-usage

Integration

Discord

Webhook URL

https://discord.com/api/webhooks/1347487508581318727/CWQNrXnAGZ54T3

Optional Discord settings

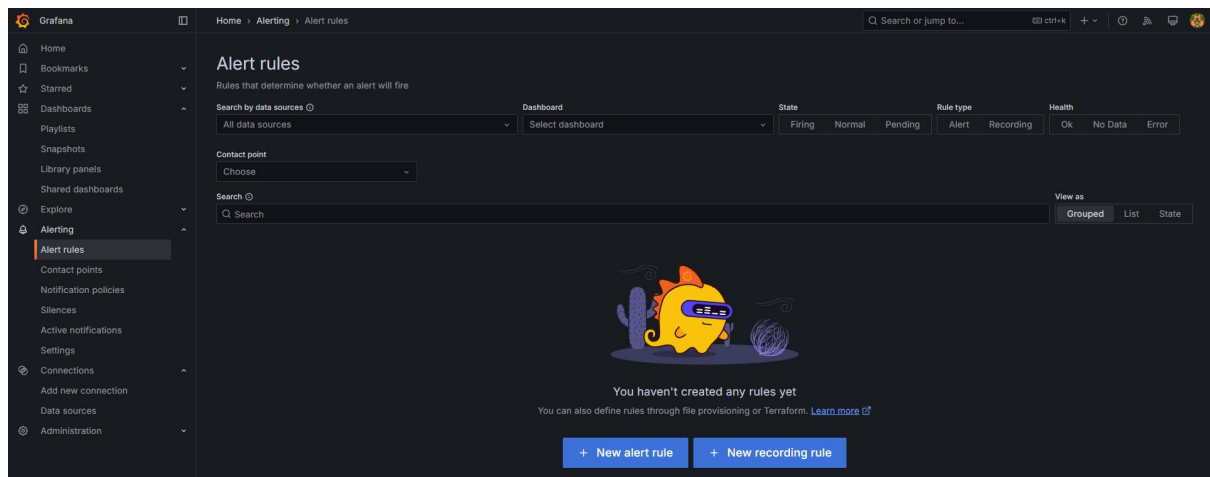
Notification settings

☐ Disable resolved message
Disable the resolve message [OK] that is sent when alerting state returns to false

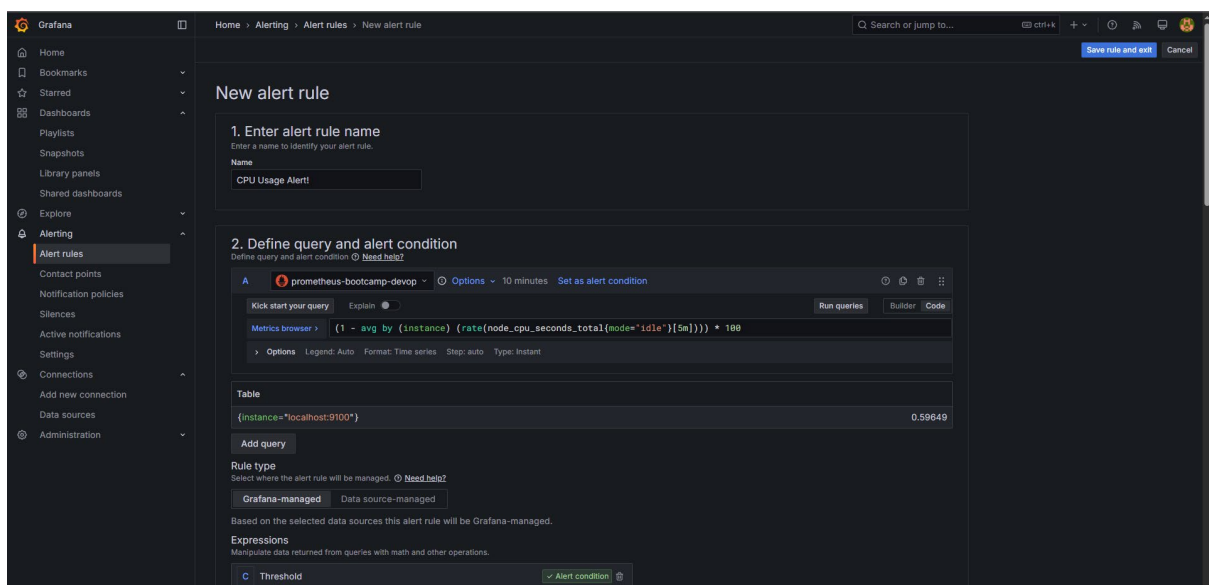
+ Add contact point integration

Save contact point Cancel

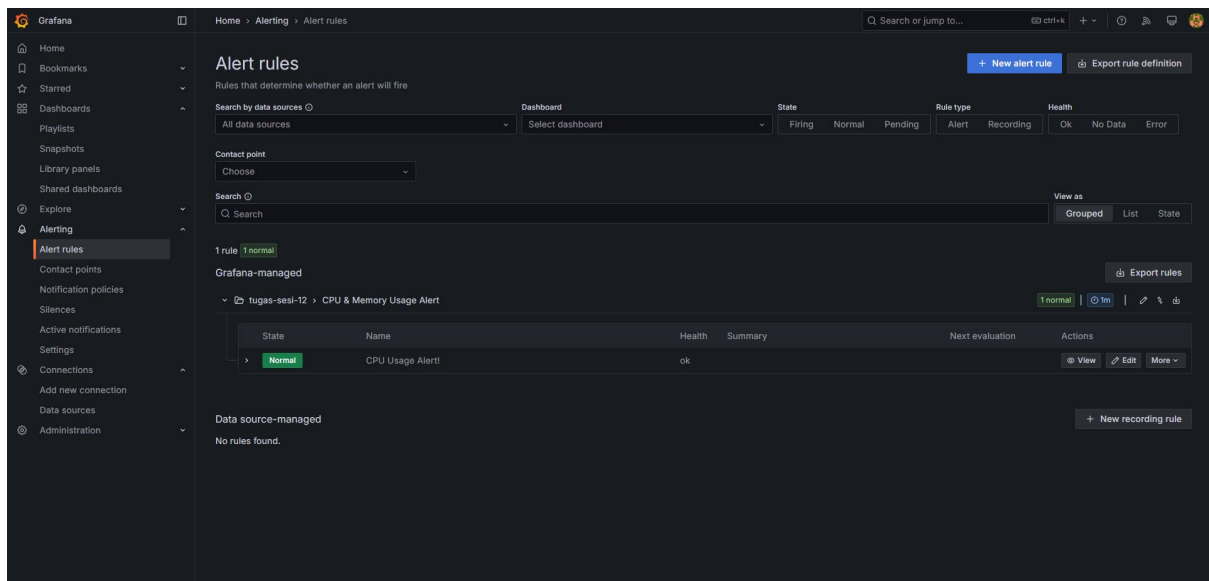
2. Membuat CPU Usage Alert Rule di Grafana



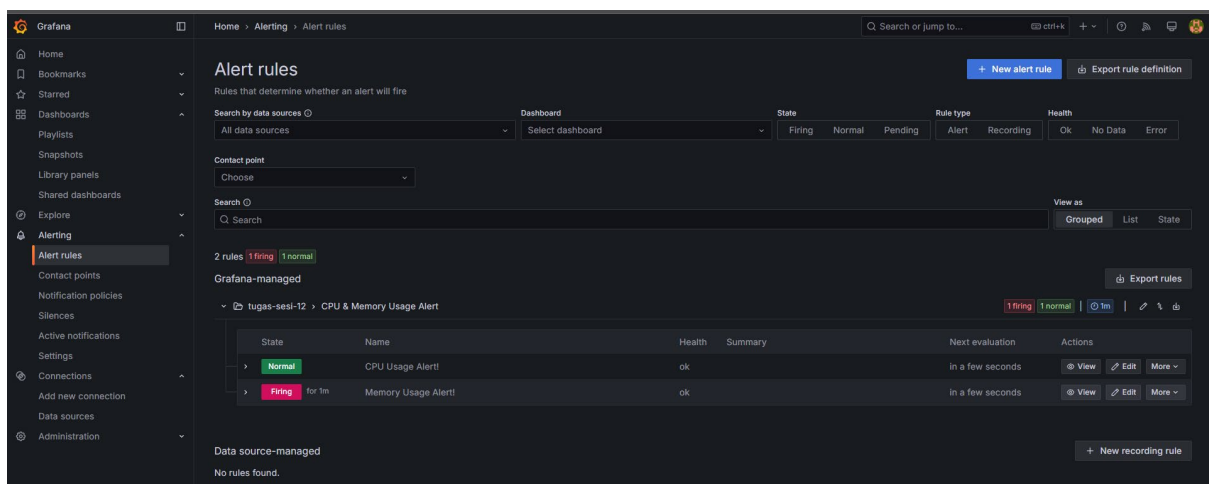
- Pada sidebar kiri pilih **Alerting** → **Alert rules** → **+ New alert rule**
- Isi nama alert menjadi **CPU Usage Alert!**
- Define query and alert condition dengan mengisi query untuk CPU usage
- Isi threshold above 40
- Pilih Add folder and labels dengan folder tugas-sesi-12
- Buat baru evaluation group CPU & memory usege alert
- Configure notifications dengan memilih Contact point discord yang telah dibuat sebelumnya
- Save rule and exit



3. Membuat Memory Usage Alert Rule di Grafana



- Pilih **+ New alert rule**
- Isi nama alert menjadi **Memory Usage Alert!**
- Define query and alert condition dengan mengisi query untuk memory usage
- Isi threshold above 40
- Pilih Add folder and labels dengan folder tugas-sesi-12
- Pilih evaluation group CPU & memory usege alert
- Configure notifications dengan memilih Contact point discord yang telah dibuat sebelumnya
- Save rule and exit



4. Verifikasi Notifikasi di Discord

- Karena Memory Usage adalah 50 yang artinya di atas threshold (40) maka memory usage akan alerting dan firing notifikasi ke discord

- CPU usage tidak alert karena di bawah nilai threshold

