

LAPORAN
MONITORING DOCKER CONTAINER DENGAN
PROMETHEUS DAN GRAFANA



Disusun Oleh:

2 D4 Teknik Informatika A (Kelompok 7)

Muhammad Qois Haidar (3122600001)

Raihan Eka Pramudya (3122600011)

Muhammad Arief S. W. (3122600015)

PROGRAM STUDI D4 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2024

I. Abstraksi

Proyek ini bertujuan untuk membangun sistem monitoring container Docker menggunakan Prometheus, cAdvisor, dan Grafana. Sistem ini akan memantau performa dan kesehatan container Docker, serta mengirimkan notifikasi melalui bot Telegram jika terjadi kondisi abnormal. Implementasi sistem ini diharapkan dapat meningkatkan efisiensi dalam pengelolaan dan pemeliharaan container Docker.

II. Pendahuluan

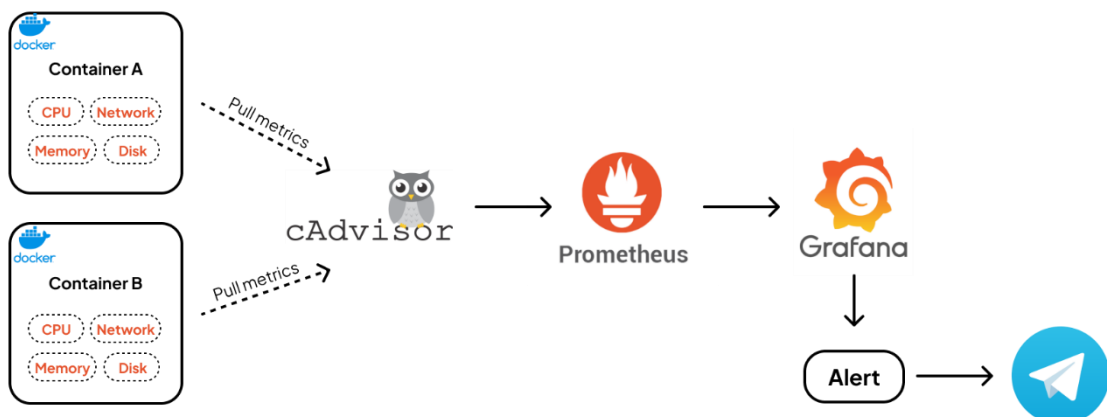
Container Docker adalah alat yang populer untuk mengemas aplikasi dengan dependensi mereka, memastikan bahwa aplikasi berjalan dengan konsisten di berbagai lingkungan. Untuk memastikan kinerja dan kesehatan container tetap optimal, diperlukan sistem monitoring yang baik. Proyek ini menggunakan cAdvisor untuk pengumpulan metrik dari container, Prometheus untuk menyimpan metrik, dan Grafana untuk visualisasi data. Selain itu, sistem ini juga diintegrasikan dengan bot Telegram untuk mengirimkan notifikasi alert.

III. Ruang Lingkup

Proyek ini mencakup:

- Instalasi dan konfigurasi Prometheus, cAdvisor, dan Grafana.
- Pengaturan bot Telegram untuk menerima notifikasi alert.
- Monitoring metrik performa utama seperti CPU, memori, dan jaringan untuk container Docker.
- Visualisasi metrik tersebut menggunakan Grafana.
- Pengaturan alert di Grafana yang dikirimkan melalui bot Telegram jika terjadi kondisi kritis pada container.

IV. Desain Sistem



- cAdvisor: Mengumpulkan metrik dari container Docker.
- Prometheus: Bertugas untuk mengumpulkan dan menyimpan metrik dari cAdvisor.
- Grafana: Menyediakan antarmuka visual untuk memantau metrik yang dikumpulkan oleh Prometheus.
- Telegram Bot: Mengirimkan notifikasi alert dari Grafana kepada tim/user melalui Telegram

V. Tim dan Tugas

- **Muhammad Qois Haidar**
 - Melakukan pengujian sistem untuk memastikan semua komponen berfungsi dengan baik.
- **Raihan Eka Pramudya**
 - Bertanggung jawab atas instalasi dan konfigurasi Prometheus, cAdvisor, dan Grafana.
 - Mengawasi keseluruhan proyek dan memastikan tujuan tercapai.
 - Menangani integrasi dengan bot Telegram.
- **Muhammad Arief S. W.**
 - Memastikan infrastruktur server berjalan dengan aman dan optimal.

VI. Tahapan Pelaksanaan

1. **Perencanaan:** Mendefinisikan tujuan dan ruang lingkup proyek, serta pembagian tugas.
2. **Instalasi dan Konfigurasi:**
 - Instalasi Docker, Prometheus, cAdvisor, dan Grafana.
 - Konfigurasi Prometheus untuk mengumpulkan data dari cAdvisor.
 - Konfigurasi Grafana untuk visualisasi data.
3. **Integrasi Bot Telegram:**
 - Membuat bot Telegram dan mendapatkan token API.
 - Konfigurasi Alertmanager untuk mengirim alert ke Telegram.
4. **Pengaturan Alert:** Membuat aturan alert di Grafana.
5. **Pengujian Sistem:** Melakukan pengujian untuk memastikan semua komponen berfungsi dengan baik dan alert dikirim dengan benar.
6. **Implementasi dan Peluncuran:** Mendemonstrasikan sistem monitoring yang telah berjalan.

VII. Implementasi

1. Instalasi Docker:

- Menghapus Versi Lama (Jika sebelumnya pernah memasang)

```
raihan@sysadmin-3122600011:~$ for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker.io' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker-doc' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker-compose' is not installed, so not removed
```

- Install Dengan Repositori APT

```
raihan@sysadmin-3122600011:~$ sudo apt-get update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 https://download.docker.com/linux/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
raihan@sysadmin-3122600011:~$ sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311).
curl is already the newest version (7.88.1-10+deb12u5).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
raihan@sysadmin-3122600011:~$ sudo install -m 0755 -d /etc/apt/keyrings
raihan@sysadmin-3122600011:~$ sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
raihan@sysadmin-3122600011:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc
raihan@sysadmin-3122600011:~$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Install Paket Docker

```
raihan@sysadmin-3122600011:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man iptables liberror-perl libip6tc2 libslirp0 patch pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn firewallld ed diffutils-doc
The following NEW packages will be installed:
```

- Cek Versi Docker

```
raihan@sysadmin-3122600011:~$ docker --version
Docker version 26.1.1, build 4cf5afa
```

2. Instalasi Prometheus, Grafana, cAdvisor:

- Membuat Direktori Baru Untuk Monitoring

```
root@debian:/home/raihan# mkdir Monitoring
root@debian:/home/raihan# ls
Desktop Documents Downloads Monitoring
```

- Membuat File docker-compose.yml dan prometheus.yml

```
root@debian:/home/raihan/Monitoring# vi docker-compose.yml
root@debian:/home/raihan/Monitoring# vi prometheus.yml
```

- Isi dari docker-compose.yml untuk instalasi Prometheus, Grafana, dan cAdvisor.

```
version: '3.2'

services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    ports:
      - 9090:9090
    command:
      - --config.file=/home/raihan/Monitoring/prometheus.yml
    volumes:
      - ./prometheus.yml:/home/raihan/Monitoring/prometheus.yml:ro
    depends_on:
      - cAdvisor

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana-storage:/var/lib/grafana

  cAdvisor:
    image: gcr.io/cadvisor/cadvisor:latest
    container_name: cAdvisor
    ports:
      - 8080:8080
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker:/var/lib/docker:ro
    depends_on:
      - redis

  redis:
    image: redis:latest
    container_name: redis
    ports:
      - 6379:6379

volumes:
  grafana-storage:
```

- Prometheus

- image: Menggunakan image prom/prometheus:latest, yang merupakan versi terbaru dari Prometheus.
- container_name: Menetapkan nama container sebagai prometheus.
- ports: Memetakan port 9090 di host ke port 9090 di container, yang merupakan port default Prometheus untuk akses web.
- command: Menjalankan Prometheus dengan file konfigurasi yang ditentukan (--config.file=/etc/prometheus/prometheus.yml).
- volumes: Mengikat file konfigurasi prometheus.yml dari direktori lokal ke dalam container di jalur /etc/prometheus/prometheus.yml dengan akses read-only (:ro).
- depends_on: Menentukan bahwa layanan prometheus bergantung pada layanan cAdvisor, sehingga cAdvisor harus dijalankan terlebih dahulu.

- Grafana
 - image: Menggunakan image grafana/grafana:latest, yang merupakan versi terbaru dari Grafana.
 - container_name: Menetapkan nama container sebagai grafana.
 - ports: Memetakan port 3000 di host ke port 3000 di container, yang merupakan port default Grafana untuk akses web.
 - volumes: Menyimpan data Grafana di volume Docker bernama grafana-storage, yang di-mount ke dalam container di jalur /var/lib/grafana.

- cAdvisor
 - image: Menggunakan image gcr.io/cadvisor/cadvisor:latest, yang merupakan versi terbaru dari cAdvisor.
 - container_name: Menetapkan nama container sebagai cadvisor.
 - ports: Memetakan port 8080 di host ke port 8080 di container, yang merupakan port default cAdvisor untuk akses web.
 - volumes:
 - (/:/rootfs:ro) Mounting root filesystem dari host ke dalam container pada direktori /rootfs dengan akses read-only. Ini bertujuan supaya cAdvisor melihat keseluruhan filesystem host.
 - (/var/run:/var/run:rw) Mounting direktori /var/run di host ke dalam container dengan akses read-write. Bertujuan supaya cAdvisor dapat mengakses informasi runtime dari Docker.
 - (/sys:/sys:ro) Mounting direktori /sys di host ke dalam container dengan akses read-only. Bertujuan untuk memberikan cAdvisor akses ke filesystem sys untuk melihat statistik kernel.
 - (/var/lib/docker:/var/lib/docker:ro) Mounting direktori /var/lib/docker di host ke dalam container dengan akses read-only. Bertujuan memberikan akses cAdvisor ke data Docker yang penting, seperti metadata container.

- Isi dari prometheus.yml untuk mengumpulkan metrik.

```
global:
  scrape_interval: 15s

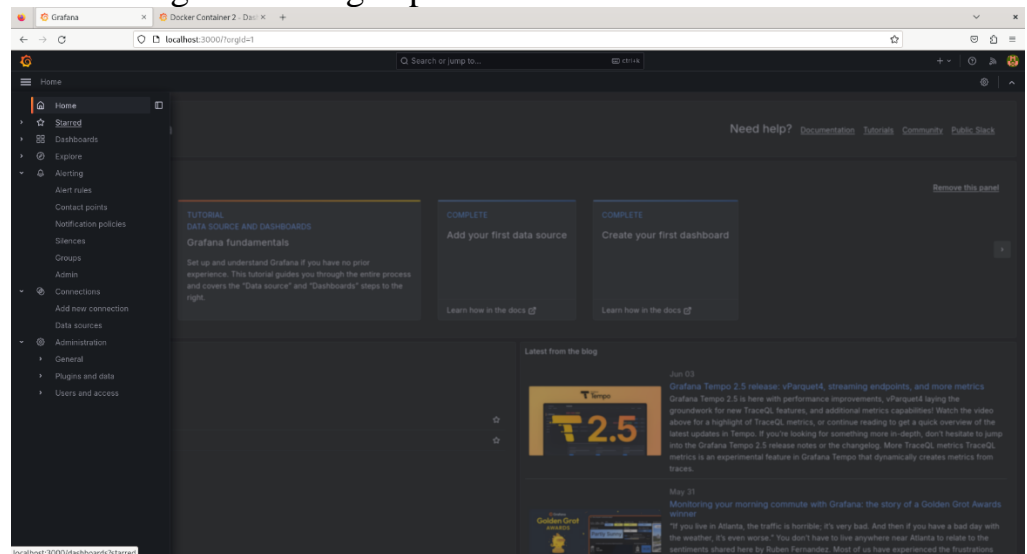
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']
```

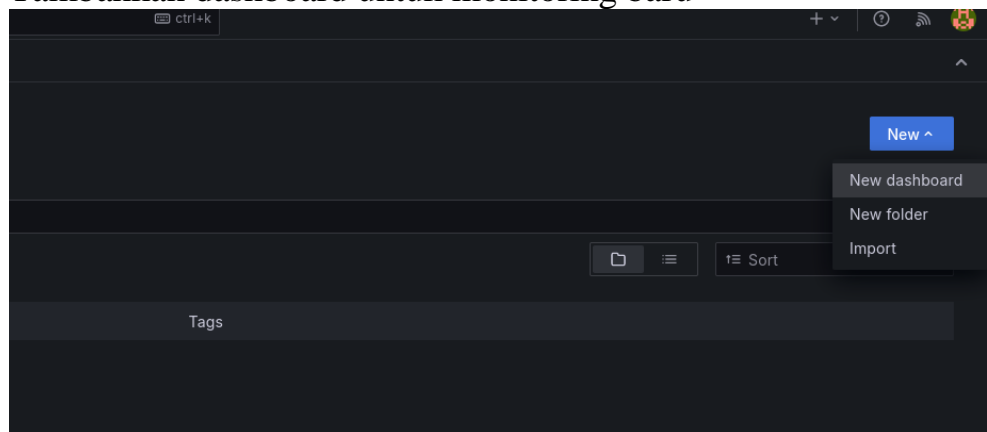
- Global Configuration: Menetapkan interval scraping default untuk semua target.
- Job 'prometheus': Konfigurasi untuk melakukan scraping data dari instance Prometheus itu sendiri.
- Job 'cadvisor': Konfigurasi untuk melakukan scraping data dari instance cAdvisor.

3. Konfigurasi Grafana Dashboard:

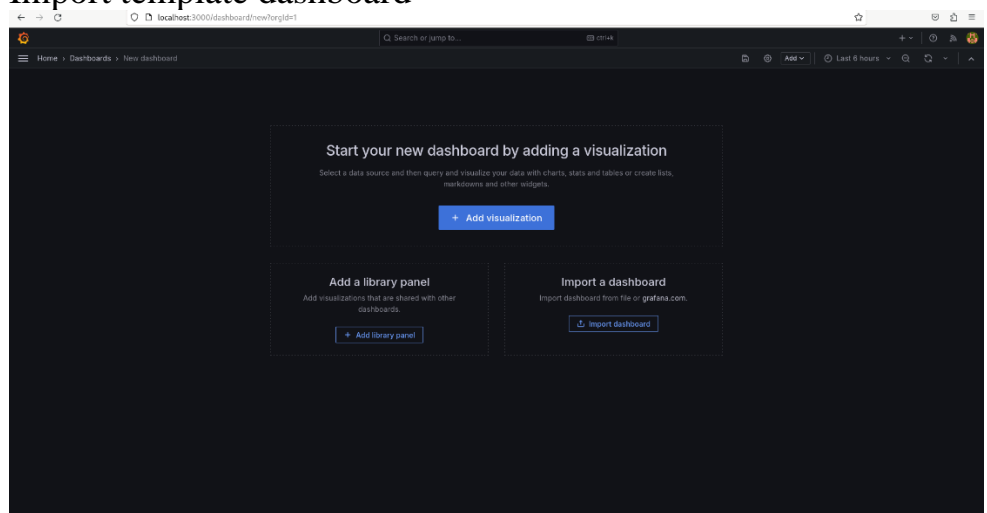
- Buka web grafana dengan port 3000



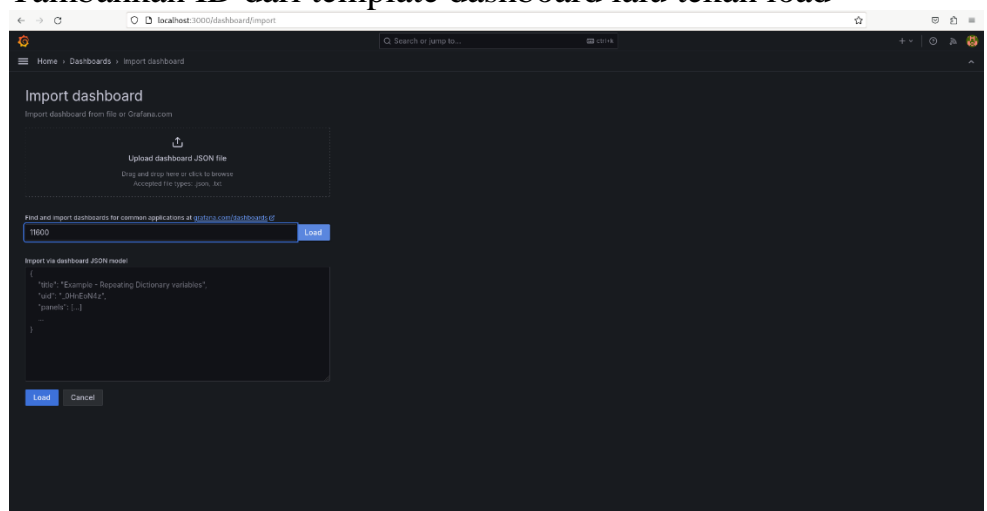
- Tambahkan dashboard untuk monitoring baru



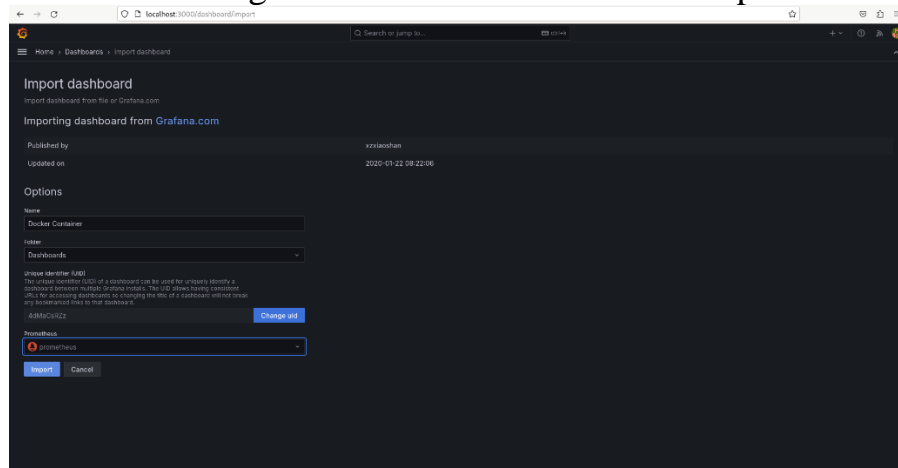
- Import template dashboard



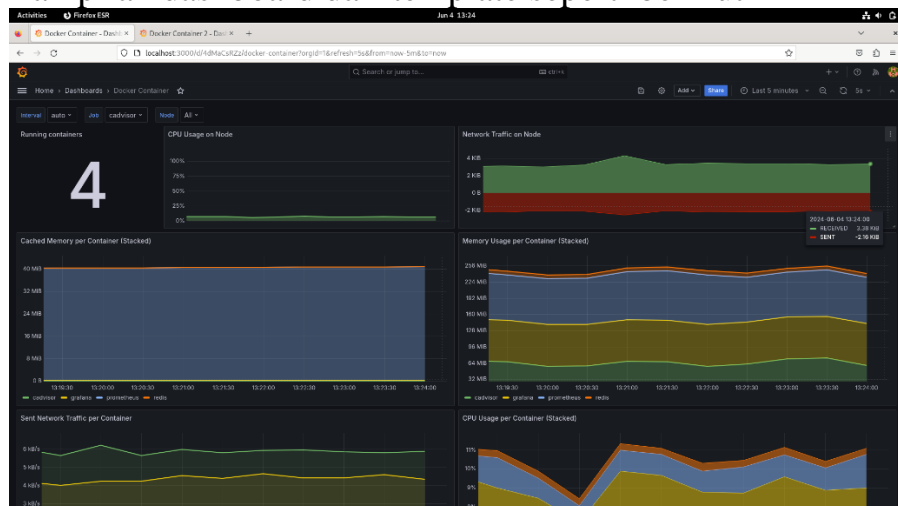
- Tambahkan ID dari template dashboard lalu tekan load



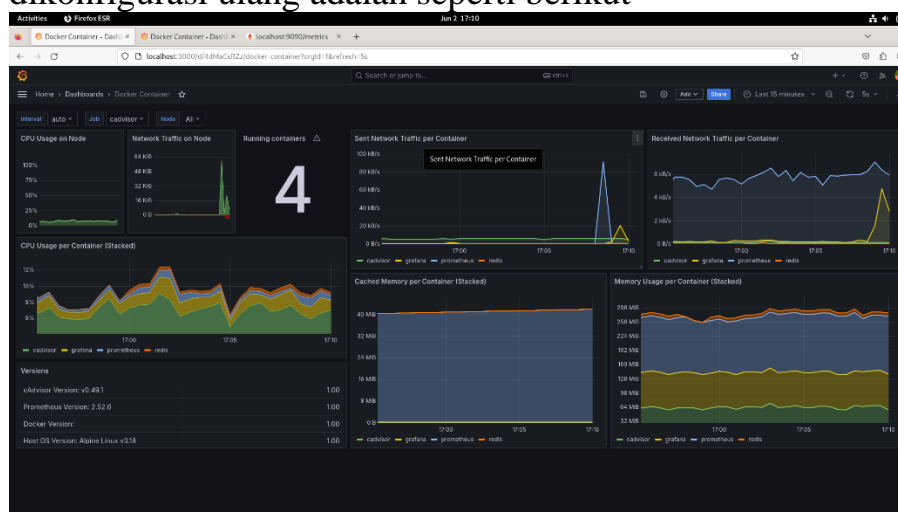
- Sesuaikan konfigurasi dashboard dan ambil dari prometheus



- Tampilan dashboard dari template seperti berikut

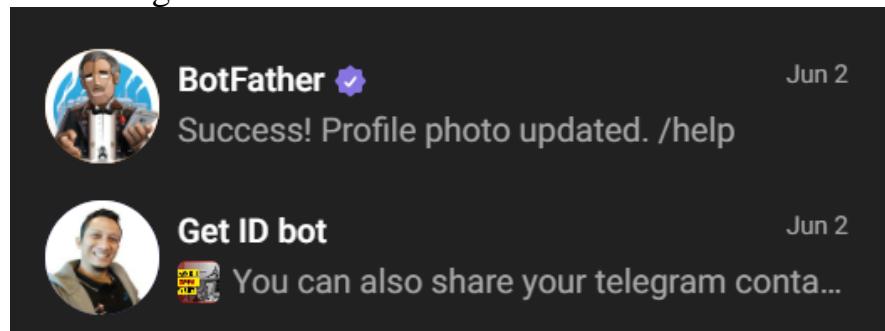


- Sesuaikan panel dan konfigurasi lain sehingga dapat menampilkan metrik yang diinginkan atau yang ingin dimonitoring, sehingga tampilan dashboard setelah dikonfigurasi ulang adalah seperti berikut

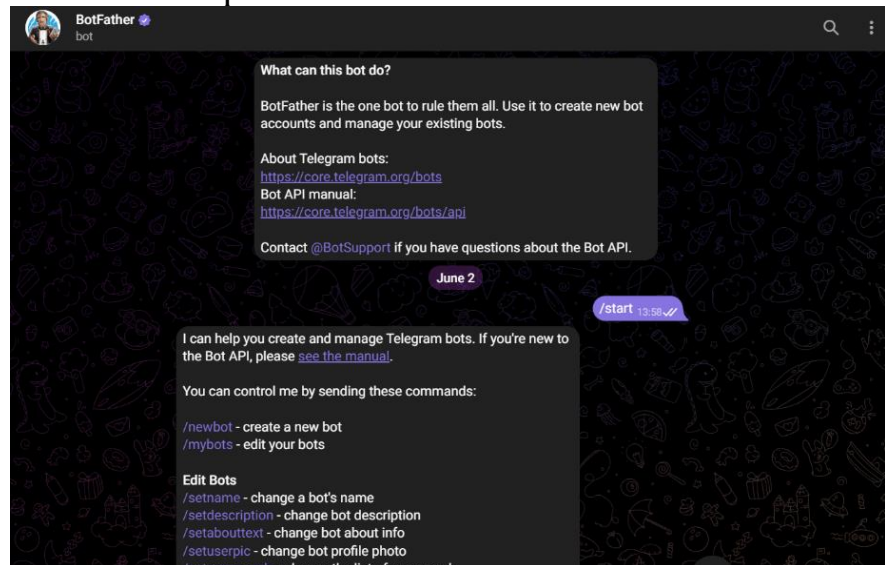


4. Konfigurasi Alert:

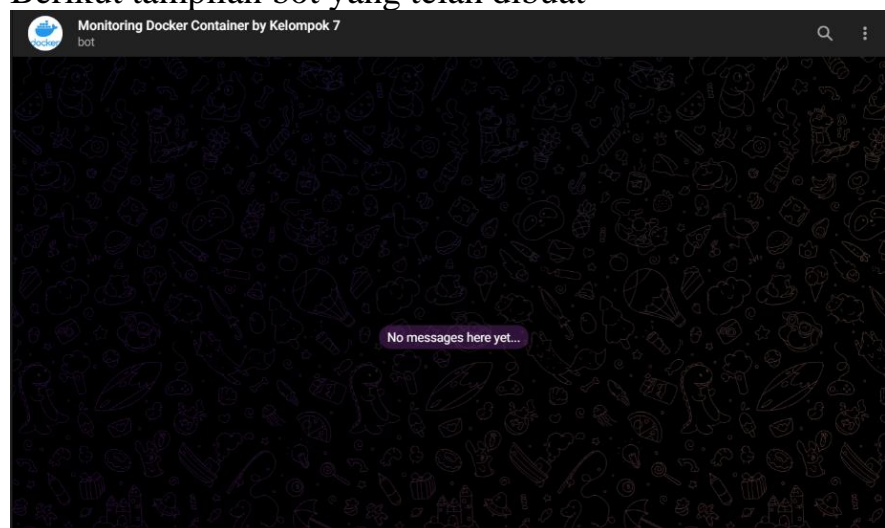
- Membuat bot Telegram:
 - Buka telegram dan search BotFather dan Get ID bot



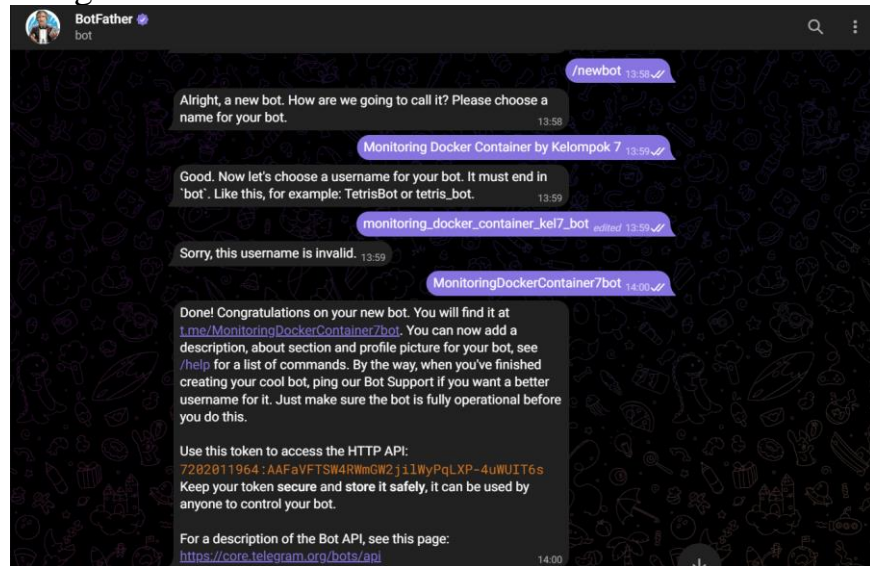
- Buat bot alert pada BotFather



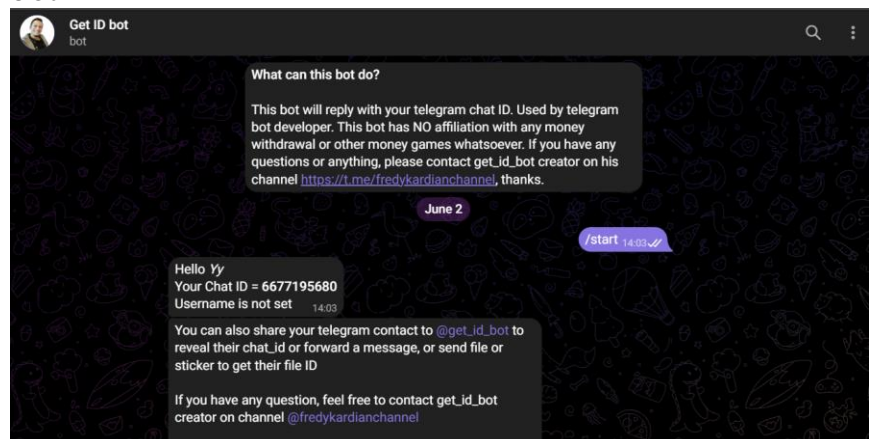
- Berikut tampilan bot yang telah dibuat



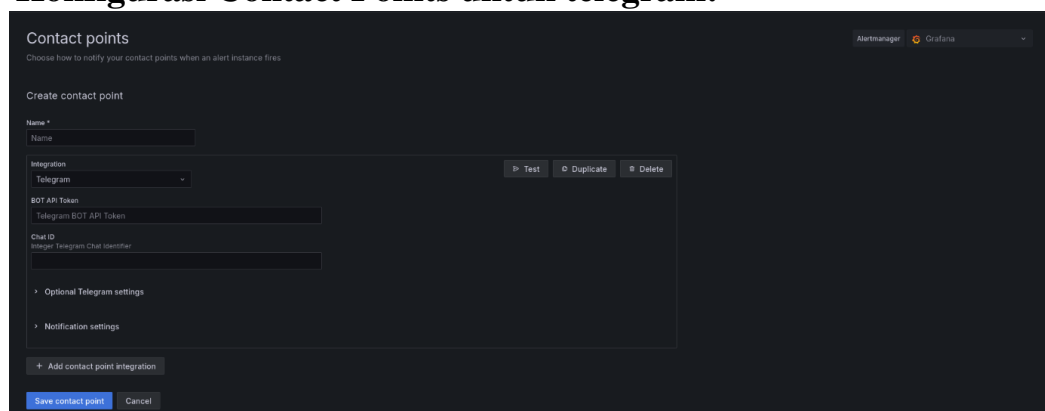
- Ikuti alurnya sehingga kita mendapatkan token API untuk mengirim data ke bot



- Dapatkan ID dari telegram kita menggunakan bot Get ID bot



5. Konfigurasi Contact Points untuk telegram:



6. Konfigurasi Notification Policy untuk alert telegram:

The screenshot shows the 'Edit notification policy' dialog. The 'Default contact point' is set to 'grafana-telegram'. The 'Group by' section shows 'grafana_folder' and 'alertname'. The 'Timing options' section includes 'Group wait' (30s), 'Group interval' (1m), and 'Repeat interval' (1h). The 'Update default policy' button is highlighted.

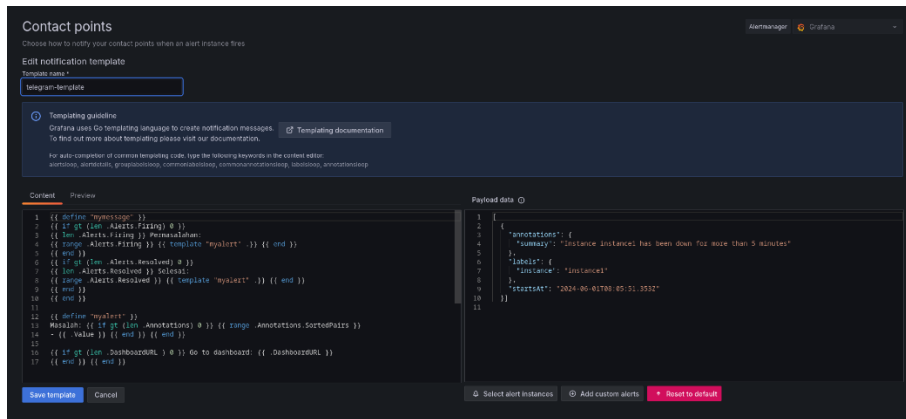
7. Konfigurasi alert rules untuk container yang mati:

The screenshot shows the 'Edit rule' dialog for a 'Container Alert' rule. The 'Define query and alert condition' section shows a query: `count(rate(container_last_seen{name=~".*"})[30s])`. The 'Rule type' is set to 'Grafana-managed'. The 'Expressions' section shows a 'Threshold' condition.

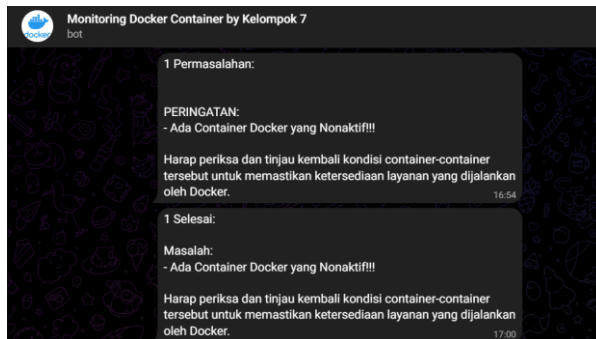
8. Konfigurasi alert rules untuk memory pada tiap container:

The screenshot shows the 'Edit rule' dialog for a 'Memory Alert' rule. The 'Define query and alert condition' section shows a query: `sum(container_memory_usage{name=~".*"} by (name)) / 1024000`. The 'Rule type' is set to 'Grafana-managed'. The 'Expressions' section shows a 'Threshold' condition.

9. Konfigurasi template message untuk susunan teks telegram:



10. Percobaan ketika ada container yang mati:



VIII. Sistem Testing

1. Uji Fungsional:

- Verifikasi bahwa Prometheus mengumpulkan metrik dari cAdvisor.
- Pastikan Grafana menampilkan metrik yang dikumpulkan dengan benar.
- Uji pengiriman alert melalui bot Telegram dengan memicu kondisi (misalnya, mematikan salah satu container docker).

2. Uji Integrasi:

- Pastikan integrasi antara cAdvisor, Prometheus, Grafana, dan bot Telegram berjalan dengan baik.
- Verifikasi pengiriman alert ke Telegram dan pastikan pesan alert sesuai dengan aturan yang ditentukan.

3. Uji Performa:

- Uji kinerja sistem monitoring dengan beberapa container untuk memastikan tidak ada bottleneck.
- Monitor penggunaan sumber daya (CPU, memori) oleh Prometheus, cAdvisor, dan Grafana.