Laporan Akhir

Membangun Sistem Monitoring untuk RAM dan CPU di Docker dengan Grafana dan Prometheus

Diajukan untuk Memenuhi Tugas Mata Kuliah

”Workshop Administrasi Jaringan”

Dosen: Ferry Astika

Oleh:

Daffa Azhar Putra Utama (3122600014)

Aulia Ilham Nur Alfian (3122600024)

Fasihul Ilmi (3122600027)

2 D4 Teknik Informatika A

Semester 4

D4 Teknik Informatika

Departemen Teknik Informatika dan Komputer

Politeknik Elektronika Negeri Surabaya

DAFTAR ISI

A. ABSTRAKSI

Sistem monitoring container Docker adalah komponen penting dalam manajemen infrastruktur modern, memastikan kinerja yang optimal dan deteksi dini terhadap masalah. Implementasi sistem monitoring ini menggunakan Prometheus dan Grafana bertujuan untuk mengumpulkan, menyimpan, dan memvisualisasikan metrik kinerja container Docker secara real-time. Prometheus, sebagai sistem monitoring dan alerting yang andal, mengumpulkan metrik dari Node Exporter yang memonitor host mesin tempat container berjalan. Data yang dikumpulkan ini kemudian disimpan dalam database time-series dan di-query untuk analisis mendalam. Grafana berperan sebagai platform visualisasi yang kuat, memungkinkan pembuatan dashboard interaktif yang memudahkan pengguna dalam memantau performa container dan host. Implementasi ini memberikan visibilitas yang lebih baik terhadap kesehatan dan kinerja container Docker, memfasilitasi pemeliharaan proaktif dan pengelolaan infrastruktur yang lebih efisien.

B. PENDAHULUAN

Sistem monitoring container Docker adalah komponen penting dalam manajemen infrastruktur modern, memastikan kinerja yang optimal dan deteksi dini terhadap masalah. Dengan meningkatnya adopsi teknologi container untuk pengembangan dan penyebaran aplikasi, kebutuhan akan solusi monitoring yang efektif dan efisien menjadi semakin krusial. Sistem monitoring tidak hanya membantu dalam memantau kinerja aplikasi yang berjalan dalam container, tetapi juga memberikan wawasan mendalam tentang kesehatan infrastruktur yang mendasarinya.

Implementasi sistem monitoring menggunakan Prometheus dan Grafana bertujuan untuk mengumpulkan, menyimpan, dan memvisualisasikan metrik kinerja container Docker secara real-time. Prometheus, sebagai sistem

monitoring dan alerting yang andal, mengumpulkan metrik dari Node Exporter yang memonitor host mesin tempat container berjalan. Data yang dikumpulkan ini kemudian disimpan dalam database time-series dan di-query untuk analisis mendalam, memungkinkan identifikasi masalah secara cepat dan akurat.

Grafana berperan sebagai platform visualisasi yang kuat, memungkinkan pembuatan dashboard interaktif yang memudahkan pengguna dalam memantau performa container dan host. Melalui dashboard ini, pengguna dapat memvisualisasikan data metrik dalam berbagai bentuk grafis, memberikan visibilitas yang lebih baik terhadap kesehatan dan kinerja container Docker. Hal ini memfasilitasi pemeliharaan proaktif dan pengelolaan infrastruktur yang lebih efisien, sehingga membantu organisasi dalam menjaga keberlangsungan operasi dan meningkatkan respons terhadap insiden.

Implementasi ini memberikan berbagai manfaat, termasuk deteksi dini terhadap potensi masalah, pengurangan downtime, dan peningkatan kinerja keseluruhan dari aplikasi yang berjalan dalam container Docker. Dengan demikian, penggunaan Prometheus dan Grafana sebagai solusi monitoring tidak hanya meningkatkan efisiensi operasional tetapi juga mendukung strategi manajemen infrastruktur yang lebih tangguh dan adaptif terhadap perubahan kebutuhan bisnis.

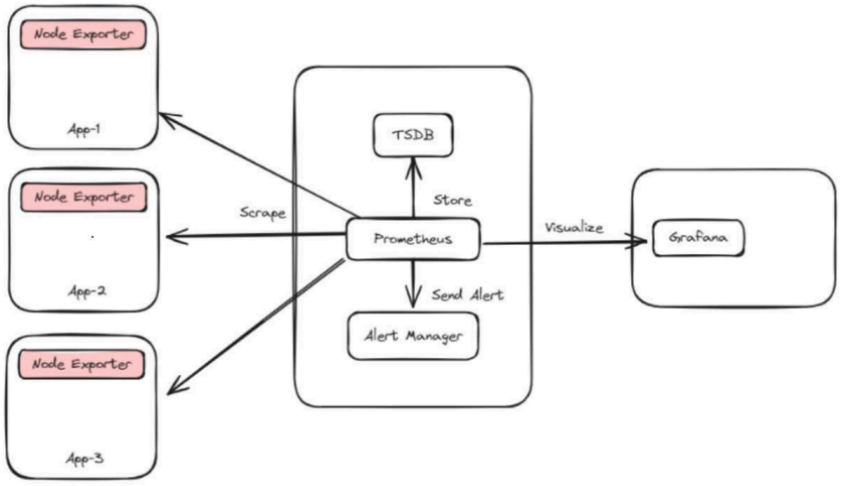
C. RUANG LINGKUP

Ruang lingkup laporan ini mengarah pada pembangunan sistem monitoring untuk penyimpanan akses acak (RAM) dan unit pemrosesan utama (CPU) di docker menggunakan Grafana dan Prometheus, yaitu sebagai berikut:

1. Penelitian ini dibatasi pada lingkungan docker yang digunakan dalam infrastruktur Teknologi Informasi.
2. Penelitian menggunakan metode implementasi dan integrasi alat monitoring Prometheus untuk pengumpulan data dan Grafana untuk visualisasi data.

3. Fokus utama adalah pada pemantauan kinerja RAM dan CPU secara real-time guna mendeteksi masalah sejak dini dan mengelola sumber daya secara efisien.

D. DESAIN SISTEM



1. Docker Containers
    - Aplikasi yang berjalan di dalam container Docker.
2. Prometheus
    - Prometheus berfungsi sebagai alat pengumpul data (data collection). Prometheus diinstal pada mesin virtual atau server terpisah. Prometheus dikonfigurasi untuk mengumpulkan metrik dari mesin virtual melalui port 9090.
3. Node Exporter
    - Node Exporter adalah agen yang berjalan pada mesin virtual Debian. Ia mengumpulkan data metrik seperti penggunaan CPU dan RAM dan mengirimkannya ke Prometheus melalui port 9100.
4. Grafana

    Grafana adalah alat visualisasi data yang mengambil data dari Prometheus.

Dengan Grafana, administrator jaringan dapat membuat dashboard yang menampilkan metrik penggunaan RAM dan CPU secara real-time.

5. Alert Manager

   Alert Manager bertanggung jawab untuk mengelola dan mengirimkan notifikasi jika terjadi kondisi abnormal pada metrik yang dipantau.

   Dalam desain ini, Grafana dapat diintegrasikan dengan Alert Manager untuk mengirimkan pesan kepada administrator melalui platform komunikasi seperti Telegram.

Alur Kerja Sistem

Pengumpulan Data oleh Node Exporter

Node Exporter mengumpulkan data metrik dari host mesin yang menjalankan container Docker, termasuk informasi tentang penggunaan RAM dan CPU. Data metrik ini kemudian dikirimkan ke Prometheus melalui port 9100.

Pengumpulan dan Penyimpanan Data oleh Prometheus

Prometheus mengumpulkan data metrik dari Node Exporter pada interval waktu yang telah ditentukan dan menyimpannya dalam database internal.

Visualisasi Data oleh Grafana

Grafana mengambil data metrik yang telah dikumpulkan oleh Prometheus. Administrator dapat membuat berbagai dashboard dan grafik untuk memvisualisasikan data penggunaan RAM dan CPU secara real-time.

Pengelolaan Peringatan oleh Alert Manager

Grafana atau Prometheus dapat diatur untuk mengirimkan alert ke Alert Manager jika metrik tertentu mencapai ambang batas yang telah ditentukan.

E. TIM & TUGAS

1. Daffa Azhar Putra Utama (3122600014)
2. Aulia Ilham Nur Alfian (3122600024)
3. Fasihul Ilmi (3122600027)

Eksekutor	Task
Daffa, Ilham	Setup prometheus
Daffa, Ilham	Setup Node Exporter
Daffa, Ilham	Integrasi Node Exporter ke Prometheus
Daffa, Ilmi	Setup Grafana
Daffa, Ilmi	Menambahkan Dashboard Resource Monitoring di Grafana
Daffa, Ilmi	Setup Alerting Contact Point dan bot Telegram
Daffa, Ilmi	Memilih Query Data yang akan di set ke Alerting Rules
Daffa, Ilmi	Setup Alerting Alert Rules
Ilham	Membuat Laporan Hasil

F. TAHAPAN PELAKSANAAN

Tanggal	Keterangan
27 Mei 2024	Menjalankan 3 service di Docker Container
29 Mei 2024	Menjalankan Prometheus dan Grafana di Docker

30 Mei 2024	Instalasi & Setup Grafana dan Layanannya
1 Juni 2024	Mengambil data dari Node Exporter dan memvisualisasikannya di Grafana
2 Juni 2024	Integrasi Alert Manager ke Email Pembuatan Laporan & Finalisasi Laporan

G. IMPLEMENTASI

1. Mulai Node Exporter
2. Verifikasi Node Exporter:
   Akses http://localhost:9100/metrics di peramban web Anda untuk melihat metrik Node Exporter.
3. Buat file konfigurasi Prometheus (prometheus.yml)
4. Buat file Docker Compose untuk Prometheus
5. Mulai Prometheus
6. Verifikasi Prometheus: |
   Akses http://localhost:9090
7. Buat file Docker Compose untuk Grafana
8. Mulai Grafana
9. Akses Grafana:
   Buka http://localhost:3000 di peramban web Anda. Masuk dengan nama pengguna dan kata sandi default (admin/admin).
10. Tambahkan Sumber Data Prometheus di Grafana:
    Pergi ke Configuration > Data Sources.
    Klik Add data source.
    Pilih Prometheus. Masukkan http://prometheus:9090 sebagai URL.
    Klik Save & Test.
11. Impor Dashboard:
    i. Pergi ke Create > Import.
    ii. Masukkan ID dashboard dari situs dashboard Grafana (misalnya, 1860 untuk Node Exporter Full).

iii. Pilih sumber data Prometheus.

iv. Klik Import.

12. Modifikasi prometheus.yml untuk menambahkan Alertmanager
13. Buat Aturan Peringatan:

    Untuk saat down, high cpu usage, high memory usage.
14. Restart Prometheus
15. Periksa Status Layanan
16. Akses Dashboard Grafana

H. SISTEM TESTING

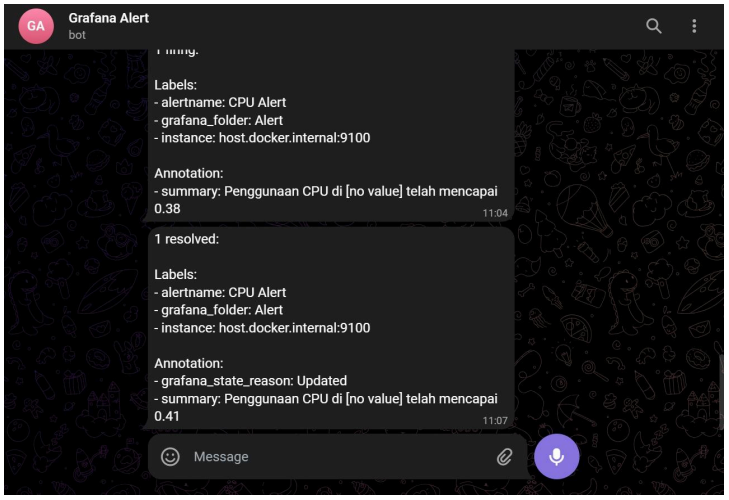
Bab ini akan menguraikan prosedur pengujian yang diterapkan pada aplikasi Grafana dan Prometheus yang telah dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa sistem monitoring berfungsi dengan baik dan sesuai harapan. Berikut adalah detail dari proses pengujian yang dilakukan:

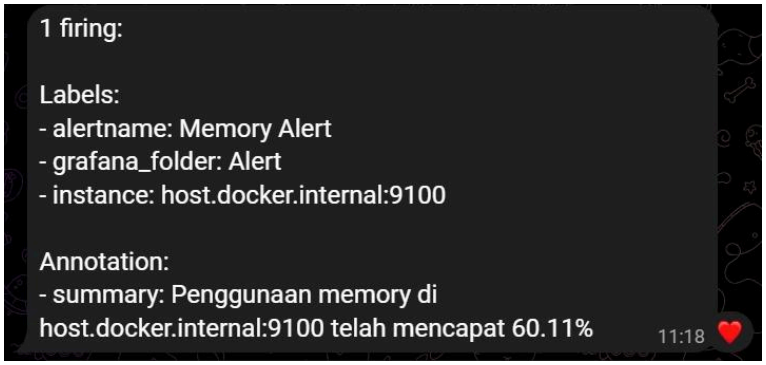
Proses Menambahkan Ambang Batas untuk Eksekusi Bot Telegram
Tujuan:Memastikan bot Telegram menerima notifikasi ketika ambang batas tertentu tercapai.
Langkah-langkah:
1. Konfigurasikan ambang batas pada Prometheus untuk parameter tertentu, seperti penggunaan CPU atau memori dan waktu henti.
2. Buat aturan alert pada Prometheus yang mendefinisikan kondisi saat ambang batas tercapai.
3. Integrasikan aturan alert ini dengan bot Telegram menggunakan webhook atau metode lain yang didukung oleh Telegram.
4. Uji dengan memanipulasi data sehingga ambang batas tercapai dan pastikan bot Telegram menerima notifikasi.

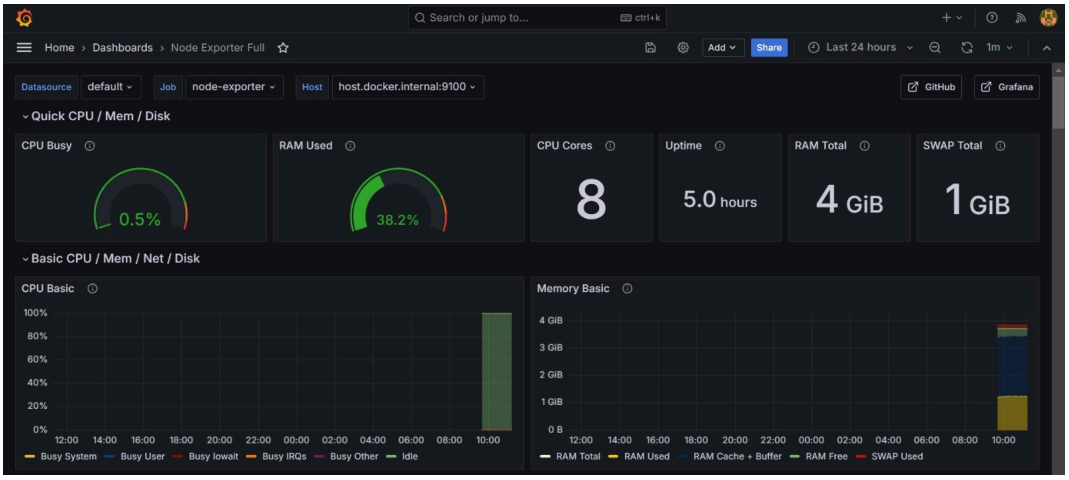
Aturan Peringatan Penggunaan Memori Ketika Ambang Batas Di Atas 60%
Tujuan: Memastikan bahwa peringatan akan aktif ketika penggunaan memori mencapai atau melebihi 60% dari ambang batas yang ditentukan.

Langkah-langkah:

1. Tentukan ambang batas penggunaan memori sebesar 60% dalam konfigurasi Prometheus.
2. Buat aturan alert pada Prometheus yang menyatakan bahwa jika penggunaan memori melebihi 60%, peringatan harus aktif.
3. Verifikasi aturan alert dengan memantau sistem hingga penggunaan memori melebihi 60%.
4. Periksa bahwa peringatan aktif terjadi sesuai dengan kondisi yang telah ditentukan.
5. Pastikan notifikasi diterima melalui sistem yang telah diintegrasikan, seperti Grafana atau bot Telegram.




1 firing:
Labels:
- alertname: Memory Alert
- grafana_folder: Alert
- instance: host.docker.internal:9100
Annotation:
- summary: Penggunaan memory di host.docker.internal:9100 telah mencapat 60.11%
11:18


Search or jump to...
ctrl+k
Home › Dashboards › Node Exporter Full
Add
Share
Last 24 hours
1m
Datasource
default
Job
node-exporter
Host
host.docker.internal:9100
GitHub
Grafana
Quick CPU / Mem / Disk
CPU Busy
0.5%
RAM Used
38.2%
CPU Cores
8
Uptime
5.0 hours
RAM Total
4 GiB
SWAP Total
1 GiB
Basic CPU / Mem / Net / Disk
CPU Basic
100%
80%
60%
40%
20%
0%
12:00 14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00
Busy System
Busy User
Busy Iowait
Busy IRQs
Busy Other
Idle
Memory Basic
4 GiB
3 GiB
2 GiB
1 GiB
0 B
12:00 14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00
RAM Total
RAM Used
RAM Cache + Buffer
RAM Free
SWAP Used