

Modul Praktikum 10

Logging Menggunakan Loki pada Grafana Berbasis Docker

1. MAKSUD

Maksud dari praktikum ini adalah untuk memberikan pengalaman langsung dalam penerapan sistem logging menggunakan Loki , yang merupakan komponen dari ekosistem Grafana , dan diimplementasikan dalam lingkungan container menggunakan Docker . Melalui modul ini, peserta akan memahami bagaimana cara mengumpulkan, menyimpan, serta memvisualisasikan log dari aplikasi atau container lain melalui antarmuka Grafana.

2. TUJUAN

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- a) Memahami konsep dasar logging dan peran Loki dalam manajemen log.
- b) Menginstal dan menjalankan Grafana beserta Loki menggunakan Docker.
- c) Menghubungkan log dari container ke Loki.
- d) Melihat dan menganalisis log melalui antarmuka Grafana.
- e) Menerapkan visualisasi log secara real-time.

3. DASAR TEORI

Apa itu Loki?

Loki adalah sistem logging open source yang dikembangkan oleh Grafana Labs . Loki dirancang untuk menyimpan dan mencari log tanpa menyimpan isi log tersebut dalam bentuk teks mentah secara lengkap. Loki menyimpan metadata seperti label (label-based indexing), sehingga efisien dalam hal penyimpanan dan performa.

Fitur Utama Loki:

- a) Ringkas dan hemat ruang penyimpanan
- b) Terintegrasi dengan Grafana
- c) Mendukung scraping log dari Docker container
- d) Mendukung agregasi log berdasarkan label

Komponen Ekosistem Loki:

- a) Promtail : Agen yang bertugas membaca log dari sistem dan mengirimnya ke Loki.
- b) Loki : Server yang menerima, memproses, dan menyimpan log.

c) Grafana : Antarmuka visualisasi untuk menampilkan log dan metrik.

Apa itu Docker?

Docker adalah platform open source untuk mengembangkan, menjalankan, dan mendistribusikan aplikasi dalam container. Container adalah lingkungan terisolasi yang memiliki semua dependensi yang dibutuhkan aplikasi.

4. PROSEDUR PRAKTIK

Langkah 1: Instalasi Docker & Docker Compose

Pastikan Docker dan Docker Compose sudah terinstal di sistem Anda:

```
sudo apt update
sudo apt install docker.io docker-compose -y
```

Langkah 2: Buat File docker-compose.yml

Buat file baru bernama **docker-compose.yml** dengan konten berikut:

```
version: '3'

services:
  loki:
    image: grafana/loki:latest
    ports:
      - "3100:3100"
    command: -config.expand-env
    environment:
      - HOST=host.docker.internal
    volumes:
      - ./loki-config.yaml:/etc/loki/local-config.yaml
    networks:
      - loki-net

  promtail:
    image: grafana/promtail:latest
    volumes:
      - ./loki-config.yaml:/etc/promtail/config.yaml
      - /var/log:/var/log
    command: -config.expand-env
    depends_on:
      - loki
    networks:
      - loki-net

  grafana:
    image: grafana/grafana:latest
    ports:
      - "3000:3000"
    environment:
      - GF_SECURITY_ADMIN_PASSWORD=admin
    depends_on:
```

```
- loki
networks:
  - loki-net

networks:
  loki-net:
```

Langkah 3: Buat File Konfigurasi Loki

Buat file **loki-config.yaml** sebagai berikut:

```
auth_enabled: false

server:
  http_listen_port: 3100

ingester:
  lifecycler:
    address: 127.0.0.1
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
    heartbeat_timeout: 1m

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  filesystem:
    chunks_directory: /tmp/loki/chunks
    rules_directory: /tmp/loki/rules

compactor:
  working_directory: /tmp/loki/boltdb-shipper-compactor
  retention_enabled: true
  retention_period: 1d

limits_config:
  enforce_metric_name: false
  reject_old_samples: true
  reject_old_samples_max_age: 168h

chunk_store_config:
  max_look_back_period: 0

table_manager:
  retention_deletes_enabled: true
```

```
retention_period: 1d

tracing:
  enable: false
```

Langkah 4: Jalankan Layanan dengan Docker Compose

`docker-compose up -d`

Langkah 5: Akses Grafana

Buka browser dan akses:

`http://localhost:3000`

Login dengan:

- a) Username: **admin**
- b) Password: **admin**

Langkah 6: Tambahkan Loki sebagai Data Source

1. Masuk ke Grafana → Configuration → Data Sources → Add data source
2. Cari dan pilih Loki
3. Isi URL: **http://loki:3100**
4. Klik Save & Test

Langkah 7: Lihat Log di Grafana

1. Klik menu Explore
2. Pilih Loki sebagai data source
3. Gunakan label seperti **{job="varlog"}** untuk melihat log

5. STUDI KASUS DAN PENYELESAIAN

Studi Kasus:

Sebuah sistem mikroservices dijalankan dalam container Docker. Tim DevOps ingin mengelola log dari setiap layanan agar dapat diakses secara terpusat dan mudah dianalisis.

Penyelesaian:

Tim menggunakan Loki + Promtail + Grafana (LG) stack untuk:

- a) Mengumpulkan log dari container mikroservices

- b) Menyimpan log secara terstruktur dengan label seperti nama layanan (**{service="auth-service"}**)
- c) Visualisasi log dan alert jika ada error tertentu

Dengan LG Stack:

- a) Semua log terpusat
- b) Mudah dicari dan difilter
- c) Integrasi langsung dengan alert Grafana

6. TUGAS

1. Lakukan instalasi Loki, Promtail, dan Grafana menggunakan Docker Compose seperti pada langkah praktikum.
2. Ganti konfigurasi Promtail agar hanya mengambil log dari satu service tertentu (misalnya Nginx).
3. Buat dashboard di Grafana yang menampilkan jumlah error log per jam.
4. Lakukan simulasi error log dengan membuat file log buatan, dan pastikan log tersebut tampil di Grafana.
5. Tuliskan hasil observasi dan kesimpulan dari praktikum ini.