

Logging Data Backup Simulation with Docker

1. Prerequisites

- Install Docker and Docker Compose:
 - Install Docker
 - Install Docker Compose

2. Create a Docker Compose File

Set up a docker-compose.yaml file to simulate the environment.

```
version: "3.9"

services:
  loki:
    image: grafana/loki:latest
    ports:
      - "3100:3100"
    command: -config.file=/etc/loki/local-config.yaml
    volumes:
      - ./loki-config.yaml:/etc/loki/local-config.yaml:ro
    depends_on:
      - minio

  minio:
    image: minio/minio:latest
    ports:
      - "9000:9000"
    environment:
```

```
MINIO_ROOT_USER: enterprise-logs
MINIO_ROOT_PASSWORD: supersecret
command: server /data
volumes:
  - minio-data:/data

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

volumes:
  minio-data:
```

3. Create the Loki Configuration

Create a loki-config.yaml file in the same directory as your docker-compose.yaml:

```
auth_enabled: false

server:
  http_listen_port: 3100

ingester:
  wal:
    enabled: false
  chunk_idle_period: 5m
  max_chunk_age: 1h
  chunk_target_size: 1048576
  lifecycler:
    ring:
      kvstore:
        store: inmemory
```

```
replication_factor: 1
```

```
schema_config:
```

```
  configs:
```

```
    - from: 2022-01-01  
      store: boltdb-shipper  
      object_store: s3  
      schema: v12  
      index:  
        prefix: loki_index_  
        period: 24h
```

```
storage_config:
```

```
  boltdb_shipper:
```

```
    active_index_directory: /data/loki/boltdb-shipper-active  
    shared_store: s3  
    cache_location: /data/loki/boltdb-shipper-cache
```

```
  aws:
```

```
    s3: http://minio:9000  
    bucketnames: chunks  
    access_key_id: enterprise-logs  
    secret_access_key: supersecret  
    s3forcepathstyle: true
```

```
limits_config:
```

```
  retention_period: 744h
```

4. Start the Environment

Run the following command in the directory containing your files:

```
docker-compose up -d
```

This will spin up:

- MinIO on <http://localhost:9000> (Access Key: enterprise-logs, Secret Key: supersecret)
- Loki on <http://localhost:3100>
- Grafana on <http://localhost:3000> (User: admin, Password: admin)

5. Add Buckets to MinIO

Using AWS

Configure AWS:

```
aws configure
```

Or

```
aws configure set aws_access_key_id enterprise-logs --profile minio  
aws configure set aws_secret_access_key supersecret --profile minio
```

Create buckets:

```
aws --endpoint-url http://localhost:9000 s3 mb s3://chunks --region us-east-1 --profile minio  
aws --endpoint-url http://localhost:9000 s3 mb s3://rules --region us-east-1 --profile minio
```

Show bucket list:

```
aws --endpoint-url http://localhost:9000 s3 ls
```

The output should like this:

```
2024-11-15 18:52:53 chunks  
2024-11-15 18:54:49 rules
```

6. Configure Grafana to Use Loki

1. Open Grafana: <http://localhost:3000>.
2. Log in with the default credentials (admin / admin).
3. Add Loki as a data source:
 - Go to Configuration > Data Sources > Add data source.
 - Select Loki.
 - Set the URL to <http://loki:3100>.
 - Click Save & Test.

7. Send Test Logs

To simulate log ingestion:

- Install and run Promtail or send HTTP POST requests to Loki's `/loki/api/v1/push`.

Example Promtail Docker Compose service:

```
promtail:
  image: grafana/promtail:latest
  command: -config.file=/etc/promtail/config.yml
  volumes:
    - ./promtail-config.yaml:/etc/promtail/config.yml:ro
  depends_on:
    - loki
```

Promtail config.yaml:

```
server:
  http_listen_port: 9080
clients:
  - url: http://loki:3100/loki/api/v1/push
scrape_configs:
  - job_name: system
    static_configs:
      - targets:
          - localhost
    labels:
      job: system
      host: localhost
      __path__: /var/log/*.log
```

To run only the Promtail service, you can use the following command:

```
docker-compose up -d promtail
```

If you want to stop and remove the container in one command, you can do:

```
docker-compose down promtail
```

8. Check Logs in Grafana

1. Open Grafana.
2. Go to Explore.
3. Choose Loki as the data source and query logs using the built-in query editor.

Cleanup

To stop and remove the environment:

```
docker-compose down -v
```

Revision #5

Created 15 November 2024 07:18:00 by Ahmad

Updated 15 November 2024 10:57:15 by Ahmad