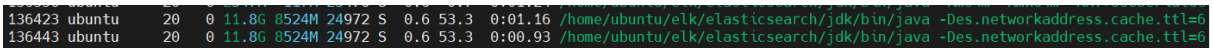
Elasticsearch, Logstash 설정

트러블 슈팅



- htop 명령어로 메모리 사용량을 확인했을 때 ES의 JVM 힙이 50프로(8GB)를 차지하고 있음
- 현재 제공받은 EC2가 16GB메모리이므로 메모리 제약상 절반으로 줄여야 함

```
# ES 힙을 4GB로 줄이기
# jvm.options 공식가이드

## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## which should be named with .options suffix, and the min and
## max should be set to the same value. For example, to set the
## heap to 4 GB, create a new file in the jvm.options.d
## directory containing these lines: ## ## -Xms4g ## -Xmx4g

cd ~ elk/elasticsearch/config/jvm.options.d
sudo nano heap.options

-Xms4g
-Xmx4g
```

엘라스틱서치 초기 구성

```
# 기본 패키지 설치
sudo apt update
sudo apt upgrade -y
sudo apt install -y wget curl tree openjdk-11-jdk
```

```
java -version

# elk 작업 디렉터리 생성
mkdir elk
cd elk

# 엘라스틱서치 8.13.4(basic) 버전 다운로드
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.1
3.4-linux-x86_64.tar.gz
tar -xzvf elasticsearch-8.13.4-linux-x86_64.tar.gz
mv elasticsearch-8.13.4 elasticsearch

# 엘라스틱서치 초기설정 수정
cd ~/elk/elasticsearch/config
sudo nano elasticsearch.yml

# elasticsearch.yml에서 하위 내용 수정
cluster.name: elasticsearch-cluster
node.name: node-1
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
#실제로는 기업 내에서 elasticsearch에 접속할 IP로 바꿔야 함
network.host: 127.0.0.1
http.port: 9200

cluster.initial_master_nodes: ['node-1']

# systemd 등록
sudo nano /etc/systemd/system/elasticsearch.service

# elasticsearch.service
[Unit]
Description=Elasticsearch
After=network.target

[Service]
Type=simple
User=ubuntu
```

```
ExecStart=/home/ubuntu/elk/elasticsearch/bin/elasticsearch
Restart=always
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target

# 엘라스틱서치 실행
+ 실행 하기전 로그 경로랑 권한 추가
sudo mkdir -p /var/lib/elasticsearch /var/log/elasticsearch
sudo chown -R ubuntu:ubuntu /var/lib/elasticsearch /var/log/elasticsearch

sudo systemctl daemon-reload
sudo systemctl enable elasticsearch
sudo systemctl start elasticsearch
sudo systemctl status elasticsearch

# 포트 리스닝
ss -lntp | grep 9200

# 헬스 체크
curl -s http://localhost:9200
```

MySQL 8.0.43 ver. (main DB)

```
# MySQL 8.0.43 설치
sudo apt update
sudo apt install -y mysql-server

# 실행 및 자동시작 확인
sudo systemctl enable mysql
sudo systemctl start mysql
sudo systemctl status mysql

# 초기 비밀번호 설정
sudo mysql_secure_installation
n → y → y → y → y
```

```
(초기에 강한 비밀번호 정책을 넣으려면 1번 설정을 y로 바꾸기)

# mysql root 로그인 후 설정
CREATE DATABASE {DB_NAME} CHARACTER SET utf8mb4 COLLATE utf8
mb4_general_ci;

# 서비스 유저 생성
[fastapi], [logstash]
CREATE USER '{DB_USER}'@'%' IDENTIFIED BY '{DB_PASSWORD}';
GRANT ALL PRIVILEGES ON {DB_NAME}.* TO '{DB_USER}'@'%';
FLUSH PRIVILEGES;
```

로그스태시 초기 구성

```
# 로그스태시 8.13.4 버전 다운로드
cd elk/
wget https://artifacts.elastic.co/downloads/logstash/logstash-8.13.4-linux-x
86_64.tar.gz
tar -xzvf logstash-8.13.4-linux-x86_64.tar.gz
mv logstash-8.13.4 logstash

# mysql jdbc 드라이버 패치
cd ~/elk/logstash
mkdir -p vendor/jar
cd vendor/jar

wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-
j-8.4.0.tar.gz
tar -xzf mysql-connector-j-8.4.0.tar.gz
cp mysql-connector-j-8.4.0/mysql-connector-j-8.4.0.jar .

# 로그스태시 초기설정 수정
cd ~/elk/logstash/config
sudo nano logstash.conf

# 로그스태시 keystore (환경변수 저장) 만들기
```

```
cd ~/elk/logstash
sudo bin/logstash-keystore create
y

[로그]
Created Logstash keystore at /home/ubuntu/elk/logstash/config/logstash.k
eystore

# 환경변수 추가하기
cd ~/elk/logstash

sudo bin/logstash-keystore add db_name
sudo bin/logstash-keystore add mysql_user
sudo bin/logstash-keystore add mysql_password
sudo bin/logstash-keystore add es_host

# 등록된 키 목록 확인하기
cd ~/elk/logstash
sudo bin/logstash-keystore list

# logstash 권한 설정
sudo chown -R ubuntu:ubuntu /home/ubuntu/elk/logstash
# keystore는 보안상 제약 필요
sudo chmod 600 /home/ubuntu/elk/logstash/config/logstash.keystore


# systemd 등록
[Unit]
Description=Logstash Data Pipeline Service
After=network.target

[Service]
Type=simple
User=ubuntu
Group=ubuntu
WorkingDirectory=/home/ubuntu/elk/logstash
ExecStart=/home/ubuntu/elk/logstash/bin/logstash -f /home/ubuntu/elk/log
stash/config/logstash.conf
```

```
Restart=always
LimitNOFILE=65536

StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target

# logstash 실행
sudo systemctl daemon-reload
sudo systemctl enable logstash
sudo systemctl start logstash
sudo systemctl status logstash

# 로그 모니터링
journalctl -u logstash -f
```

logstash - mysql jdbc 테스트 명령어

```
sudo ./bin/logstash -e '
input {
  jdbc {
    jdbc_driver_library ⇒ "/home/ubuntu/elk/logstash/vendor/jar/mysql-conn
ector-j-8.4.0.jar"
    jdbc_driver_class   ⇒ "com.mysql.cj.jdbc.Driver"
    jdbc_connection_string ⇒ "jdbc:mysql://127.0.0.1:3306/${db_name}?use
SSL=false&allowPublicKeyRetrieval=true&serverTimezone=Asia/Seoul&cha
racterEncoding=utf8"
    jdbc_user     ⇒ "${mysql_user}"
    jdbc_password ⇒ "${mysql_password}"
    statement ⇒ "SELECT 1 AS ping"
  }
}
output {
  stdout { codec ⇒ rubydebug }
```

```
}
'
```

결과

```
[2025-10-27T02:12:09,479][INFO ][logstash.agent           ] Successfully star
ted Logstash API endpoint {:port⇒9600}
[2025-10-27T02:12:10,592][INFO ][logstash.inputs.jdbc     ][main][2754f021
90eabdfab2e95dbb4a8b78012f3f2050b725fc223ce0224190b942ce] (0.02
7390s) SELECT 1 AS ping
{
        "ping" ⇒ 1,
   "@timestamp" ⇒ 2025-10-27T02:12:10.661Z,
     "@version" ⇒ "1"
}
[2025-10-27T02:12:11,153][INFO ][logstash.javapipeline    ][main] Pipeline t
erminated {"pipeline.id"⇒"main"}
[2025-10-27T02:12:11,590][INFO ][logstash.pipelinesregistry] Removed pip
eline from registry successfully {:pipeline_id⇒:main}
[2025-10-27T02:12:11,674][INFO ][logstash.runner          ] Logstash shut do
wn.
```

logstash 파이프라인 구축 → logstash.conf

```
##############################
# MySQL → Logstash → Elasticsearch
# 대상: annotation (semantic search 최소 필드)
##############################
input {
  jdbc {
    jdbc_driver_library ⇒ "/home/ubuntu/elk/logstash/vendor/jar/mysql-conn
ector-j-8.4.0.jar"
    jdbc_driver_class   ⇒ "com.mysql.cj.jdbc.Driver"

    # keystore에서 MySQL 정보 불러오기
    jdbc_connection_string ⇒ "jdbc:mysql://127.0.0.1:3306/${db_name}?use
SSL=false&allowPublicKeyRetrieval=true&serverTimezone=Asia/Seoul&cha
```

```
racterEncoding=utf8"
    jdbc_user => "${mysql_user}"
    jdbc_password => "${mysql_password}"

    # 5분마다 증분 수행
    schedule => "*/5 * * * *"

    statement => "
      SELECT
        an.id         AS annotation_id,
        a.id          AS asset_id,
        lc.name       AS class_name,
        an.created_at AS created_at
      FROM annotation an
      JOIN asset a            ON a.id = an.asset_id
      LEFT JOIN label_class lc ON lc.id = an.label_class_id
      WHERE an.created_at > :sql_last_value
      ORDER BY an.created_at ASC
    "

    use_column_value => true
    tracking_column => "created_at"
    tracking_column_type => "timestamp"
    last_run_metadata_path => "/home/ubuntu/elk/logstash/.last_run_annotati
on.yml"
    clean_run => false
  }
}

filter {
  # class_name이 null일 때 lowercase 에러 방지 + 소문자 통일
  ruby {
    code => "
      v = event.get('class_name')
      v = '' if v.nil?
      event.set('class_name', v.to_s.downcase)
    "
  }
```

```
  mutate {
    add_field ⇒ { "search_text" ⇒ "%{class_name}" }
    add_field ⇒ { "embedding_missing" ⇒ true }
    remove_field ⇒ ["@version"]
  }
}

output {
  # 데이터가 있을 때 → Elasticsearch 업데이트
  if [annotation_id] {
    elasticsearch {
      hosts ⇒ ["${es_host:http://localhost:9200}"]
      index ⇒ "annotations_semantic"
      action ⇒ "update"
      document_id ⇒ "%{annotation_id}"
      doc_as_upsert ⇒ true
    }
    stdout { codec ⇒ rubydebug }
  }
  # 데이터가 없을 때 → 로그 출력
  else {
    stdout {
      codec ⇒ line {
        format ⇒ "이번 주기에는 새로운 annotation 데이터가 없습니다."
      }
    }
  }
}
```

Logstash → ES 파이프라인 연동 성공

```
# Logstash 로그
Oct 30 10:37:49 ip-172-26-0-7 logstash[141584]: [2025-10-30T10:37:49,49
5][WARN ][logstash.outputs.elasticsearch][main] Restored connection to E
S instance {:url⇒"http://localhost:9200/"}
```

```
# Elasticsearch 로그
Oct 30 10:37:49 ip-172-26-0-7 elasticsearch[136981]: [2025-10-30T10:37:4
9,984][INFO ][o.e.c.m.MetadataIndexTemplateService] [node-1] adding ind
ex template [ecs-logstash] for index patterns [ecs-logstash-*]
```

ES 매핑

```
PUT annotations_semantic
{
  "settings": {
    "index": {
      "refresh_interval": "1s",
      "number_of_shards": 1,
      "number_of_replicas": 0
    }
  },
  "mappings": {
    "properties": {
      "annotation_id":  { "type": "keyword" },
      "asset_id":       { "type": "keyword" },
      "class_name":     { "type": "text", "analyzer": "english" },
      "search_text":    { "type": "text", "analyzer": "english" },
      "embedding_vector": {
        "type": "dense_vector",
        "dims": 768,
        "index": true,
        "similarity": "cosine"
      },
      "embedding_missing": { "type": "boolean" },
      "created_at":     { "type": "date" }
    }
  }
}
```

하이브리드 검색 예시 (BM25 + kNN)

```
POST annotations_semantic/_search
{
  "size": 50,
  "knn": {
    "field": "embedding_vector",
    "query_vector": [/* embed("animal") */],
    "k": 200,
    "num_candidates": 400
  },
  "query": { "match": { "search_text": "animal" } },
  "_source": ["annotation_id","asset_id","class_name"]
}
```

→ 응답의 `asset_id` 리스트로 DB에서 썸네일/geometry를 한 번에 조회해서 그리면 됨.