

**A406 : 코스모스(Coursemores)**

삼성청년SW아카데미 서울캠퍼스 8기

자율프로젝트 (2023.04.10 ~ 2023.05.19)

**포팅 매뉴얼**

담당 컨설턴트 : 오형남

이상민(팀장), 이지원, 우상빈, 임성준, 윤지영, 전태영

**[목차]**

1. 프로젝트 기술 스택 --------------------------------------------------2
2. 빌드 및 배포 설정 --------------------------------------------------4
3. DB 계정 및 프로퍼티 정의 --------------------------------------------6
4. 소셜 로그인 KAKAO ------------------------------------------------9
5. S3 SERVER ------------------------------------------------------11
6. 구글 플레이스토어 --------------------------------------------------12
7. 프로젝트 기술 스택

1) 이슈관리 : Jira

2) 형상관리 : GitLab

3) 커뮤니케이션 : Notion, MatterMost

4) 개발 환경

(1). OS : Windows 10

(2). IDE

A. IntelliJ IDEA 2022.2.2.

B. Visual Studio Code 1.75

(3). DataBase : MySQL, Redis

(4). Server : AWS EC2

A. Ubuntu 20.04.4 LTS

B. Nginx 1.18.0

C. Docker 20.10.12

5) 상세 사용

(1) Back-end

A. Java(Open JDK zulu 11.0.18)

B. Spring Boot 2.7.7

C. Gradle 7.6

D. Lombok 1.18.24, JPA Hibernate

(2) Front-end

A. Flutter 3.10.1

(3) AWS S3  
(4) Infra

A. Elasticsearch:7.9.1

B. logstash:8.7.1

C. Kibana:7.9.1

D. Apm-server:7.9.2

E. Jenkins 2.387.3

1. 빌드 및 배포 설정

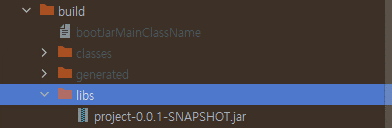
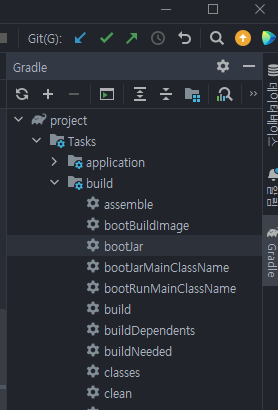
1) Front-end

Flutter 설치 후 프로젝트 SDK를 설정

‘flutter build appbundle’

입력 시 aab 파일 생성

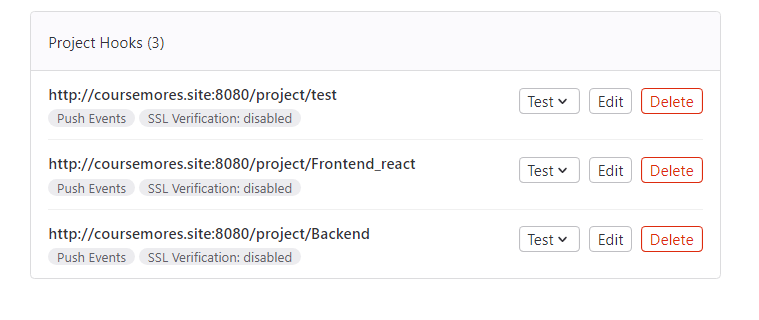
2) Back-end



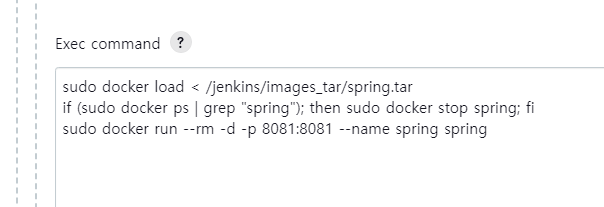
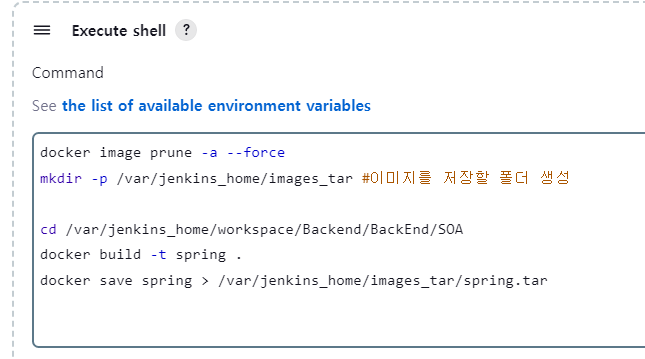
Gradle 탭에서 bootJar를 실행하면 build/libs에 .jar 실행파일이 생성된다.

3) Jenkins

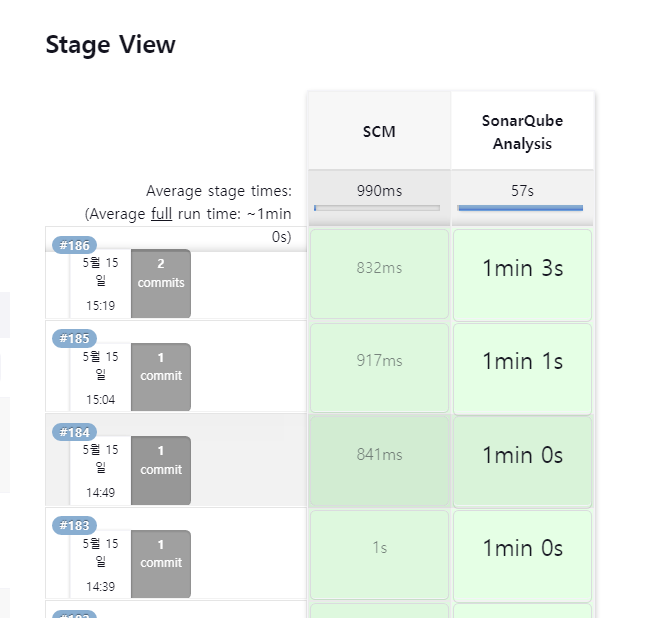


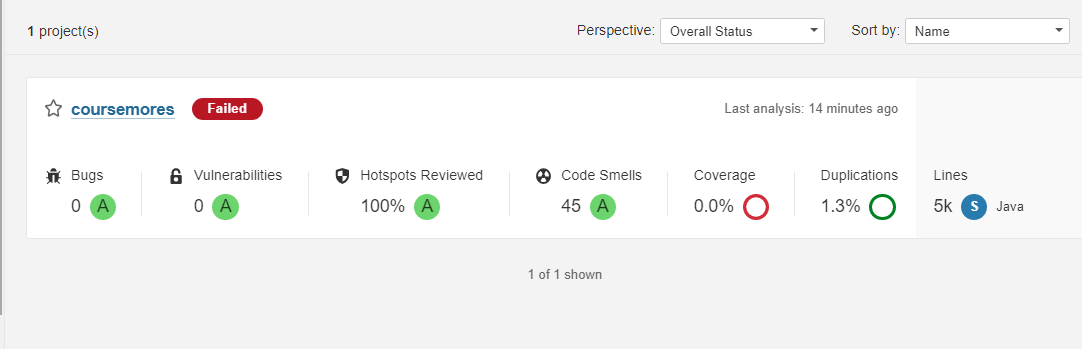


Jenkins webhook을 이용해 CI/CD를 구축하고 GitLab 저장소 main 브랜치에 push event가 일어나면 Backend, Front\_react item에서 각각 빌드파일이 만들어지고 빌드 후 조치로 **Send build artifacts over SSH의** Exec command의 다음과 같은 명령어를 실행함으로써 Docker container가 실행하게 된다.



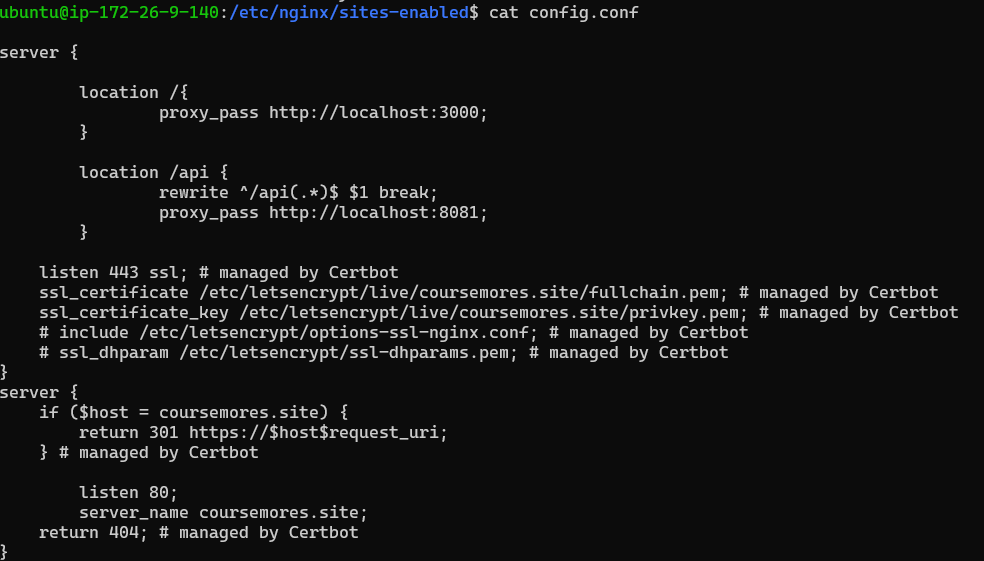
Test item은 Sonarqube를 이용하기 위한 파이프라인을 구축한 것이며



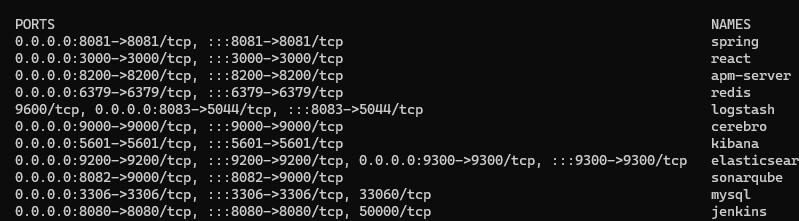


위와 같이 Webhook이 발생하면 Sonarqube에 프로젝트 파일이 전송되어 코드 분석이 이루어진다.

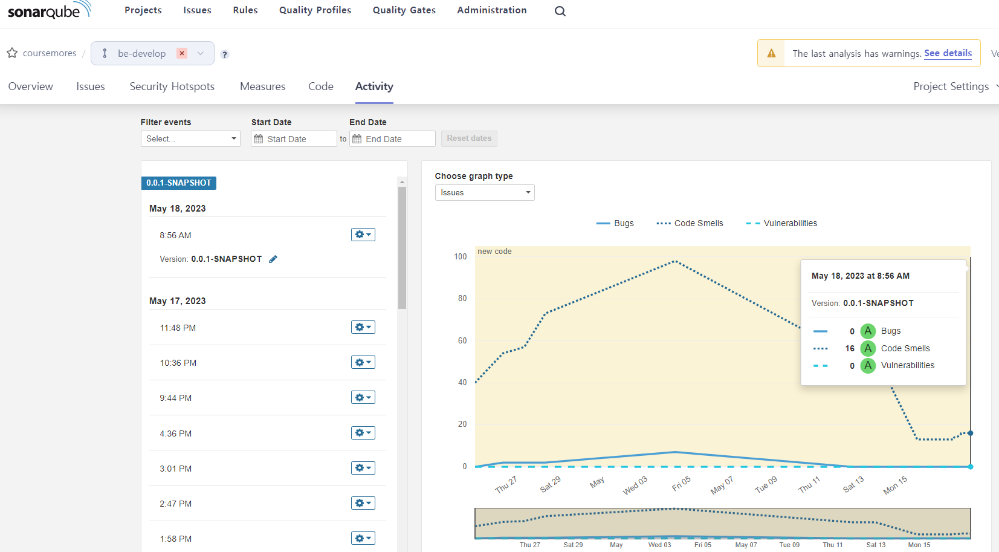
4) NginX



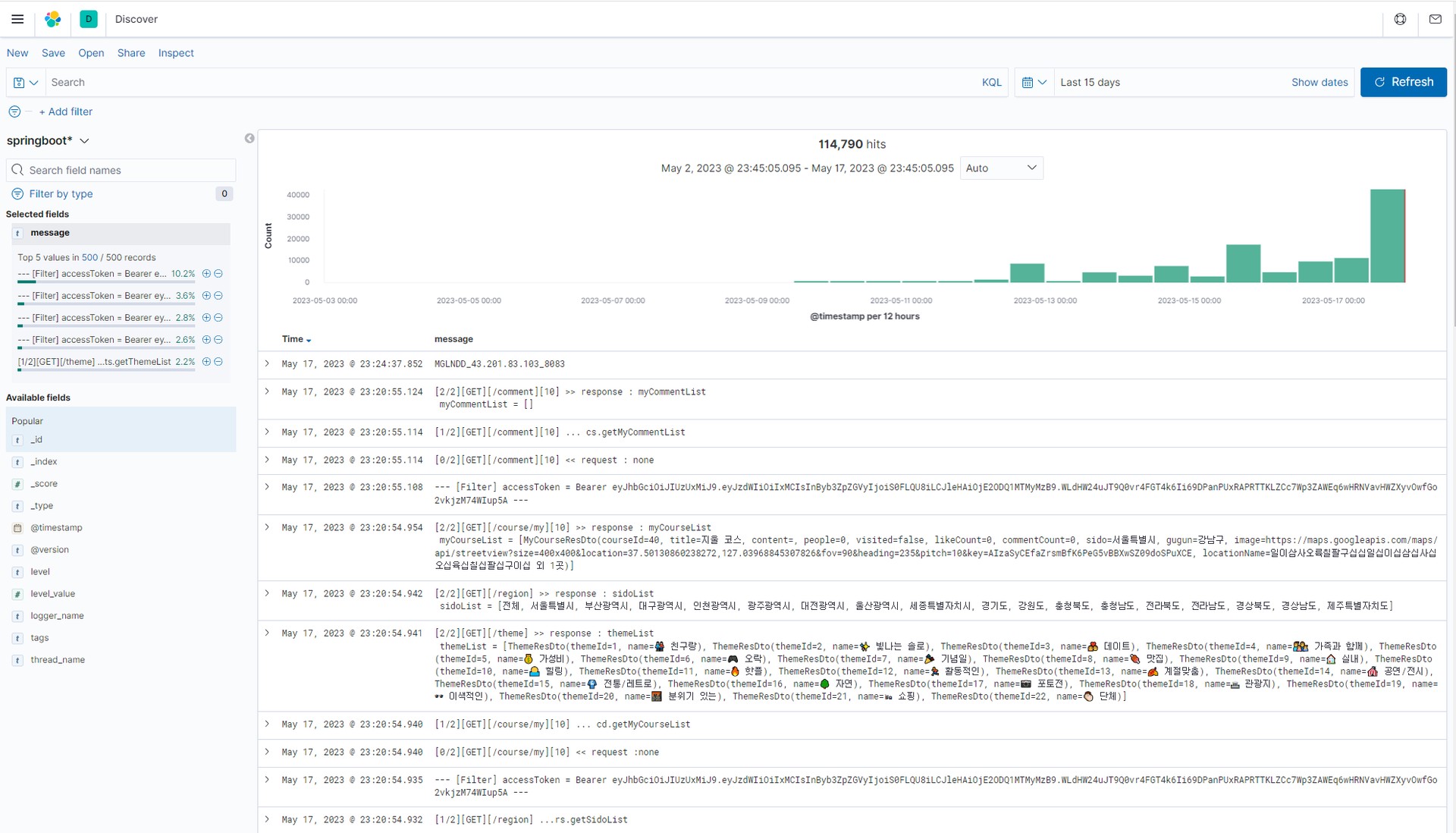
SSL 인증서 적용 및 프록시 설정

5) Docker 

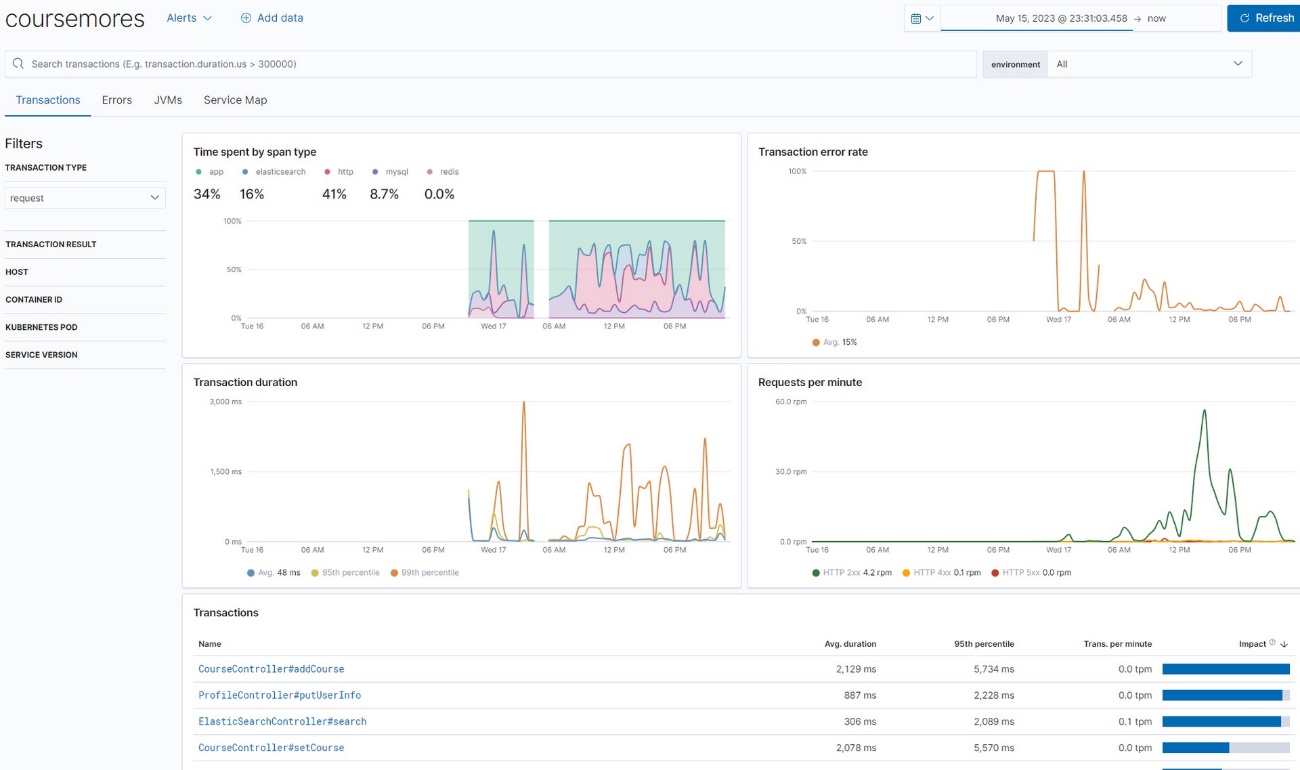
6) sonarqube



7) ELK



8) Elastic APM

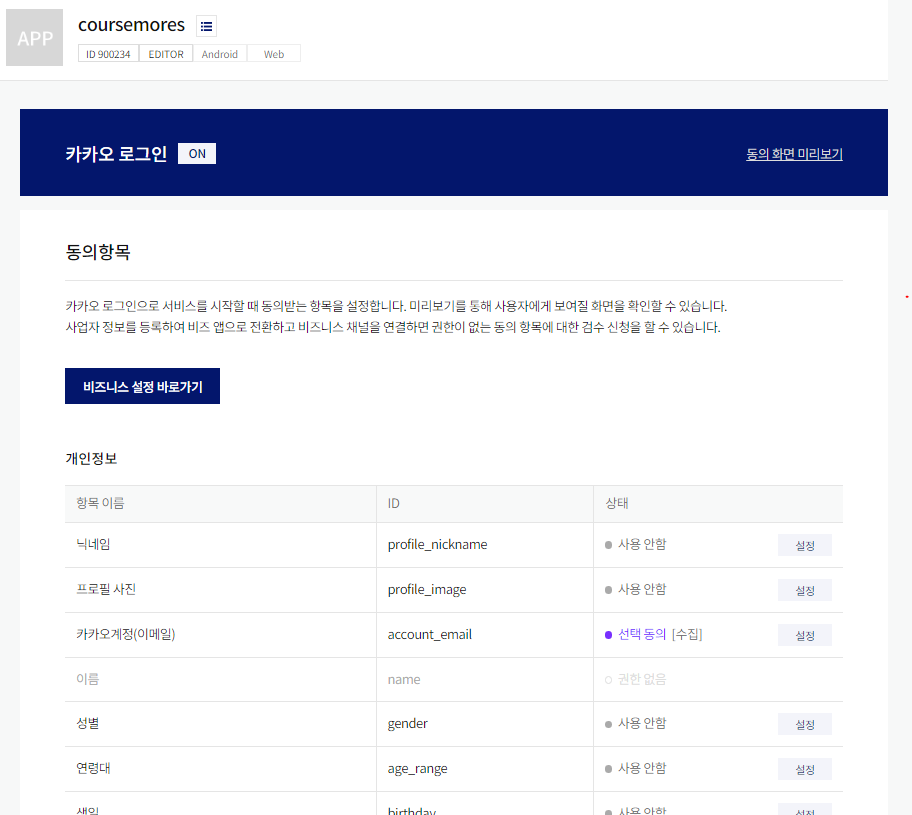


3. DB 계정 및 프로퍼티 정의

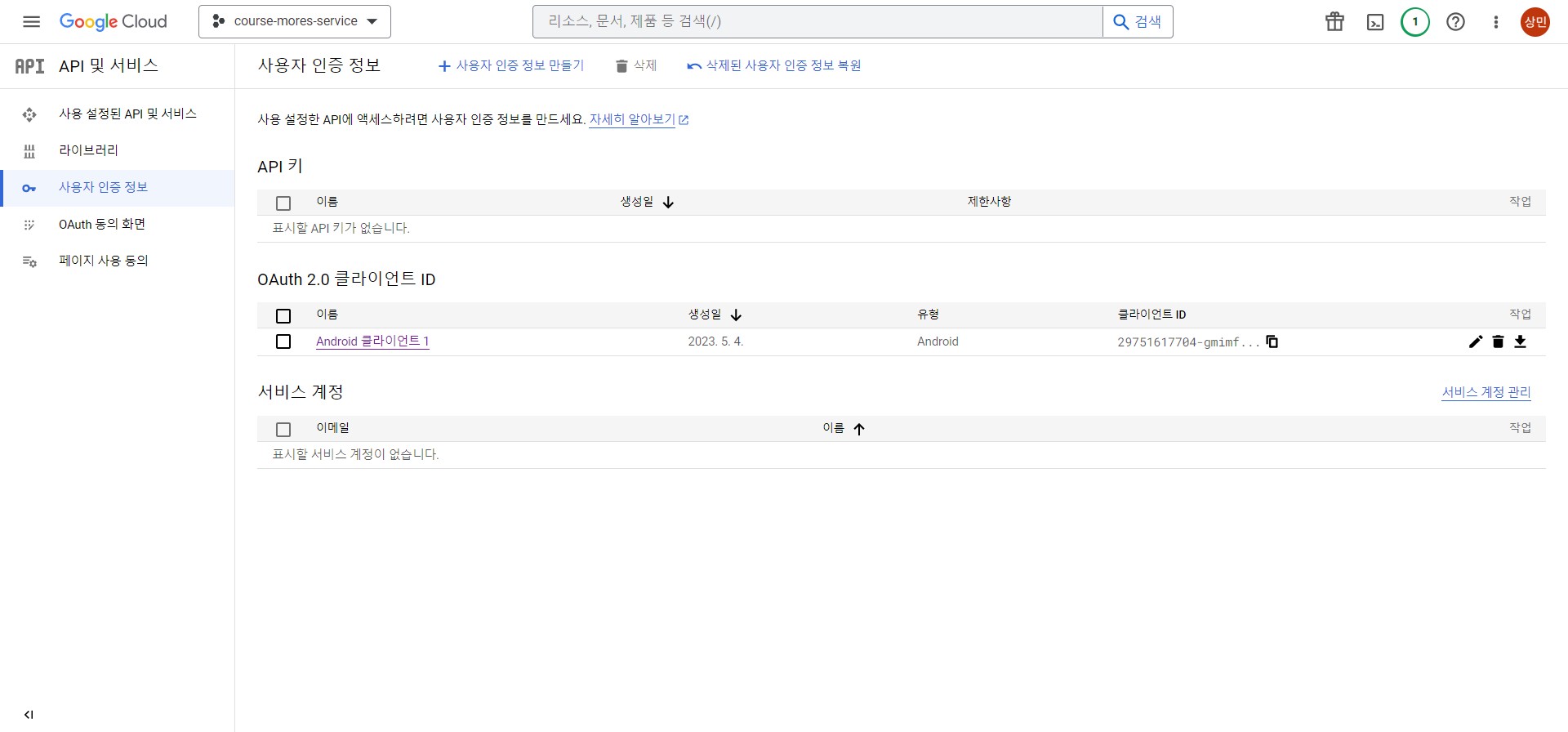
# server  
server.port=8081  
  
# MySql DB  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://coursemores.site:3306/coursemores?serverTimezone=Asia/Seoul&characterEncoding=UTF-8  
spring.datasource.username=root  
spring.datasource.password=moham1234  
  
# Redis  
redis.host=coursemores.site  
redis.port=6379  
  
# JPA  
spring.jpa.hibernate.ddl-auto=update  
#spring.jpa.properties.hibernate.format\_sql=true  
#spring.jpa.properties.hibernate.highlight\_sql=true  
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.properties.hibernate.default\_batch\_fetch\_size=100  
spring.jpa.open-in-view=false  
  
#logging.level.org.hibernate.SQL=debug  
#logging.level.org.hibernate.type=true  
  
# logger  
logging.level.com.moham.coursemores.api=debug  
logging.level.com.moham.coursemores.common.filter=debug  
  
  
# jwt  
jwt.secret = c3ByaW5nLWJvb3Qtc2VjdXJpdHktand0LXR1dG9yaWFsLWppd29vbi1zcHJpbmctYm9vdC1zZWN1cml0eS1qd3QtdHV0b3JpYWwK  
  
# image  
spring.servlet.multipart.maxFileSize=100MB  
spring.servlet.multipart.maxRequestSize=100MB  
  
# S3  
cloud.aws.credentials.access-key=AKIA4RXWK672QPN6M2GL  
cloud.aws.credentials.secret-key=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  
cloud.aws.stack.auto-=false  
cloud.aws.s3.bucket=coursemores  
cloud.aws.region.static=us-east-1  
  
# kakao login  
oauth.kakao.url.auth=https://kauth.kakao.com  
oauth.kakao.url.api=https://kapi.kakao.com  
oauth.kakao.client-id=ca761c26a8b276f2877dd33a41d3dfc6  
  
#elasticsearch  
spring.main.allow-bean-definition-overriding=true  
  
# 변수  
token.authorities.key=provider  
# 30분  
token.access.expire=180000000  
# 1일  
token.refresh.expire=604800000

4. 소셜 로그인 KAKAO & GOOGLE

[개인정보 동의항목 - 카카오]

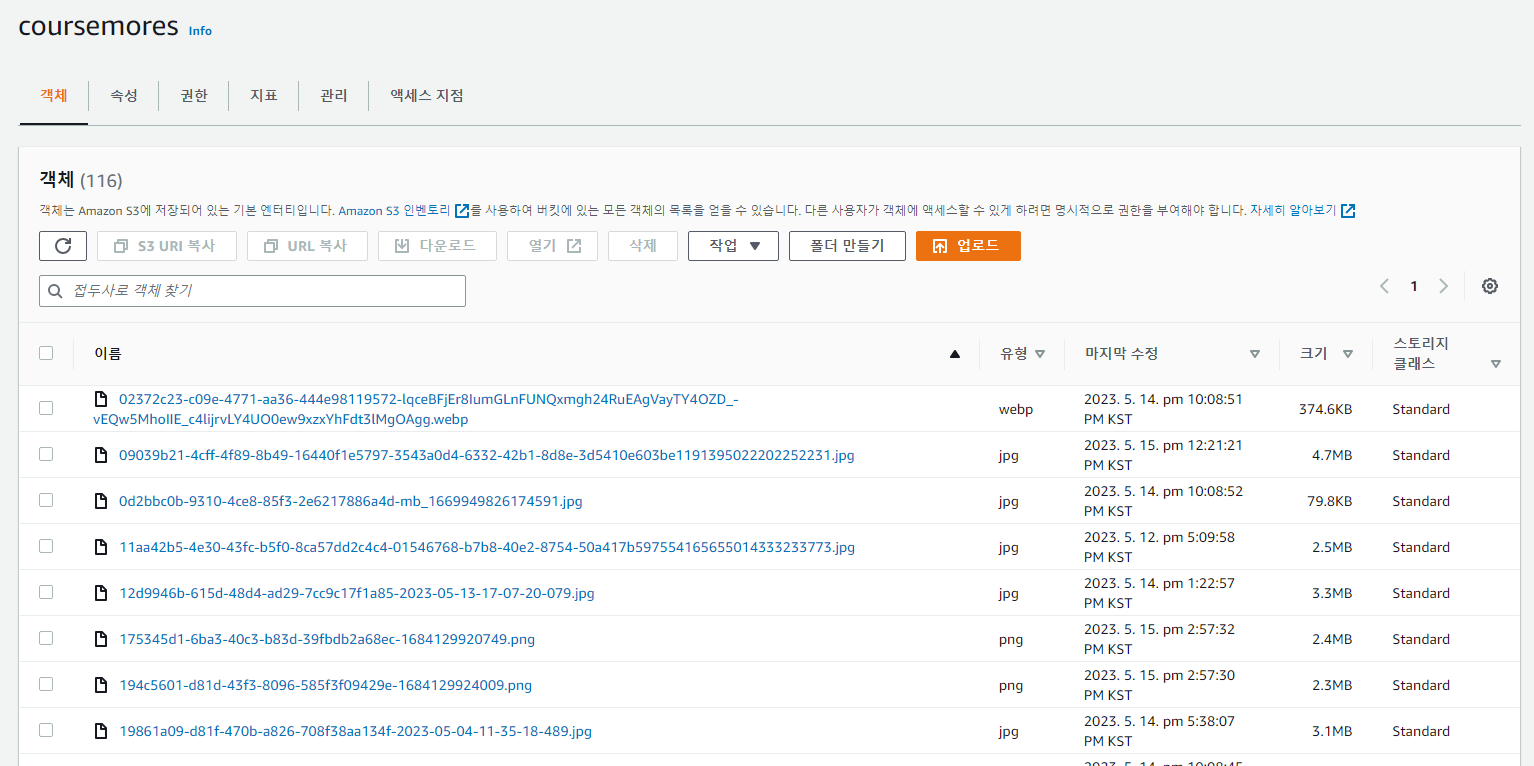


[구글 로그인 ]



5. S3 SERVER

[버킷 사용 정보]



6. 구글 플레이스토어

