



Seller B

D105 : SellerB

삼성SW청년아카데미 구미캠퍼스 7기

공동프로젝트 (7주 : 2022/07/04 ~ 2022/08/19)

포팅 매뉴얼

담당 컨설턴트 : 서성수

신혜연(팀장), 배혜연, 이기덕, 이주형, 임채현

<<목차>>

1. 프로젝트 기술 스택.....	2
2. 빌드 상세 내용.....	3
3. 배포 특이사항.....	5
4. DB 계정.....	7
5. 프로퍼티 정의.....	7
6. 외부 서비스.....	9

1. 프로젝트 기술 스택

가. 이슈 관리 : Jira

나. 형상 관리 : Gitlab, Sourcetree

다. 커뮤니케이션 : Mattermost, Notion, Discord, Webex

라. 개발 환경

1) OS : Windows 10

2) IDE

가) IntelliJ IDE 2022.1.3

나) Visual Studio Code

다) Android Studio 7.2.1

3) Database : MariaDB 10.8.3

4) Server : AWS EC2 (MobaXterm)

가) Ubuntu 20.04 LTS

나) Docker version 20.10.17

마. 상세 사용

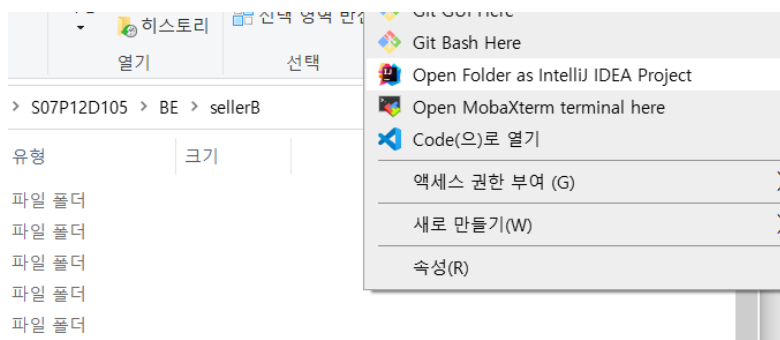
1) Backend

- 가) Java (Azul Zulu version 11.0.16)
 - 나) Spring Boot Gradle 7.5
 - 다) Lombok 1.18.24, springdoc-openapi-ui 1.6.6, firebase 9.0.0, spring-cloud-starter-aws 2.3.1
- 2) Frontend
- 가) HTML5, CSS3, JavaScript(ES6)
 - 나) React
 - React 18.2.0
 - 다) 라이브러리
 - @ckeditor/ckeditor5-build-classic : ^35.0.1
 - @ckeditor/ckeditor5-react : ^5.0.2
 - react-js-pagination : ^3.0.3
 - react-router-dom : ^6.3.0
 - react-toastify : ^8.2.0
- 3) Android
- 가) Android Studio 7.2.1
 - 나) Gradle

2. 빌드 상세 내용

가. BE/sellerB 빌드

1) Git clone 후 intellij로 프로젝트 실행



- 2) Secret.yml 파일을 WS07P12D105WBE\sellerB\src\main\resources에 이동
- 3) ./gradlew clean build 실행

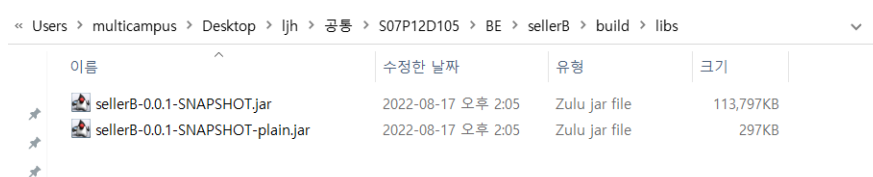
```

MINGW64:/c:/Users/multicampus/Desktop/ljh/공통/S07P12D105/BE/sellerB
multicampus@DESKTOP-KVCQHCD MINGW64 ~/Desktop/ljh/공통/S07P12D105/BE/sellerB (C7b3deded...)
$ ./gradlew clean build

> Task :compileJava
C:\Users\multicampus\Desktop\ljh\공통\S07P12D105\BE\sellerB\src\main\java\background\sellerB\dto\RequestOpenviduBody.java:13: warning: @Builder.Default requires @Builder or @SuperBuilder on the class for it to mean anything.
    @Builder.Default
    ^

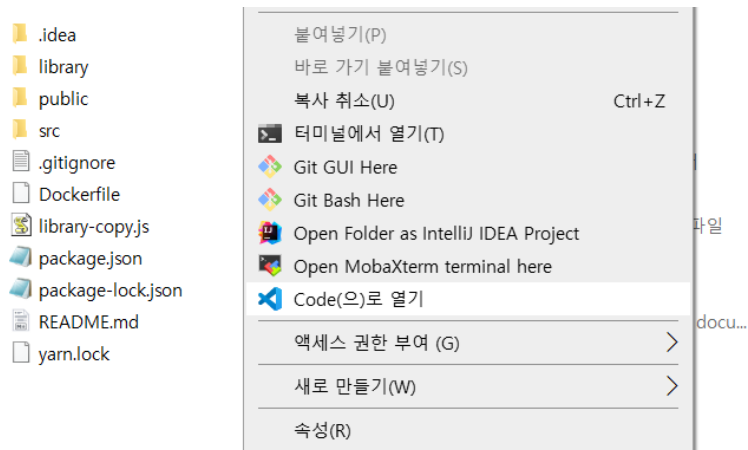
```

- 4) WS07P12D105WBE\sellerB\build\libs에 jar파일 생성



나. FE/sellerb_fe

- 1) Git clone 후 vscode로 프로젝트 실행



- 2) cmd에 npm install
- 3) cmd에 npm run build

다. AOS

- 1) Git clone
- 2) Mac/Linux : ./gradlew tasks, Windows : gradlew.bat tasks
- 3) Max/Linux : ./gradlew installDebug, Windows : gradlew.bat installDebug

3. 배포 특이사항

가. Jenkins shell script

1) backend

The first screenshot shows a Jenkins 'Execute shell' script block with the following commands:

```
cd ./BE/sellerB
./gradlew clean build
docker build -t sellerb_backend .
```

The second screenshot shows a Jenkins 'Execute shell' script block with the following commands:

```
docker ps -q --filter "name=sellerb-backend" | grep -q . && docker stop sellerb-backend && docker rm -f sellerb-backend | true
docker run -p 5000:5000 -d --name=sellerb-backend sellerb_backend -v /etc/localtime:/etc/localtime:ro -e TZ=Asia/Seoul
```

2) frontend

The screenshot shows a Jenkins 'Execute shell' script block with the following commands:

```
cd ./FE/sellerb_fe
docker build -t sellerb_frontend .
docker ps -a -q --filter name=sellerb-frontend | grep -q . && docker stop sellerb-frontend && docker rm sellerb-frontend
docker run -d -p 3000:3000 --name sellerb-frontend sellerb_frontend
```

나. Gitlab Webhook

- 1) Gitlab User Settings에 Access Token을 생성하고 젠킨스에 입력, 젠킨스에서 Gitlab push되면 빌드를 하도록 체크하고 Webhook URL을 기억해놓고 고급을 눌러 secret token을 생성한 후 Gitlab의 Settings의 Webhooks로 진입해 Webhook을 생성하고 기억해둔 URL과 secret token을 넣어주면 된다.

다. EC2 세팅

- 1) 도커 설치

```

sudo apt-get update

sudo apt-get install

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux /ubuntu bionic stable"

sudo apt-get update

sudo apt-get install docker-ce

```

2) Nginx

```

sudo apt-get install nginx

nginx -v

sudo apt-get install letsencrypt

sudo systemctl stop nginx

sudo letsencrypt certonly --standalone -d [도메인 주소]

cd /etc/nginx/sites-available

vi config

➔ 아래 프로퍼티에서 nginx 세팅 값 입력

sudo ln -s /etc/nginx/sites-available/config /etc/nginx/sites-enabled/config

sudo nginx -t

sudo systemctl restart nginx

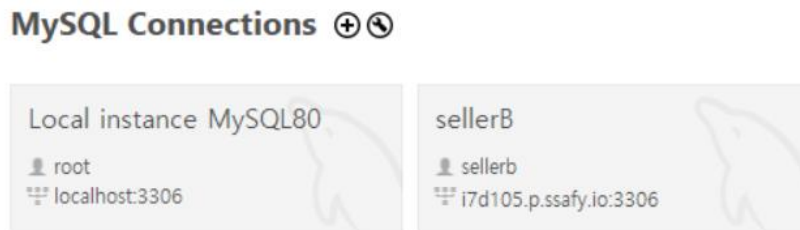
```

라. 최종 도커 컨테이너 상태

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
2ff0f51559f6	sellerb_frontend	"docker-entrypoint.s..."	sellerb_frontend	7 hours ago	Up 7 hours	0.0.0.0:3000→3000/tcp, :::3000→3000/tcp
902f6d6d383f	sellerb_backend	"java -jar /sellerB..."	sellerb_backend	7 hours ago	Up 7 hours	0.0.0.0:5000→5000/tcp, :::5000→5000/tcp
6516be889ba9	jenkins/jenkins:lts	"/usr/bin/tini -- /u..."	sellerb-backend	2 weeks ago	Up 2 weeks	0.0.0.0:8080→8080/tcp, :::8080→8080/tcp
50000/tcp	jenkins	"docker-entrypoint.s..."	jenkins	2 weeks ago	Up 2 weeks	0.0.0.0:3306→3306/tcp, :::3306→3306/tcp
b0f8f98394c9	mysql	"docker-entrypoint.s..."	mysql	2 weeks ago	Up 2 weeks	
8154da725e10	openvidu/openvidu-call:2.22.0	"docker-entrypoint.s..."	openvidu_app_1	2 weeks ago	Up 2 weeks	
4e5212b88743	kurento/kurento-media-server:6.16.0	"/entrypoint.sh"	openvidu_kms_1	2 weeks ago	Up 2 weeks (healthy)	
8de6997cb2e9	openvidu/openvidu-coturn:2.22.0	"docker-entrypoint.s..."	openvidu_coturn_1	2 weeks ago	Up 2 weeks	0.0.0.0:3478→3478/tcp, 0.0.0.0:3478→3478/udp
udp, :::3478→3478/tcp, :::3478→3478/udp, 5349/tcp, 454afedc2b1f	openvidu/openvidu-proxy:2.22.0	"/docker-entrypoint..."	openvidu_nginx_1	2 weeks ago	Up 2 weeks	
1eb062c5e3c8	openvidu/openvidu-server:2.22.0	"/usr/local/bin/entr..."	openvidu_server_1	2 weeks ago	Up 2 weeks	

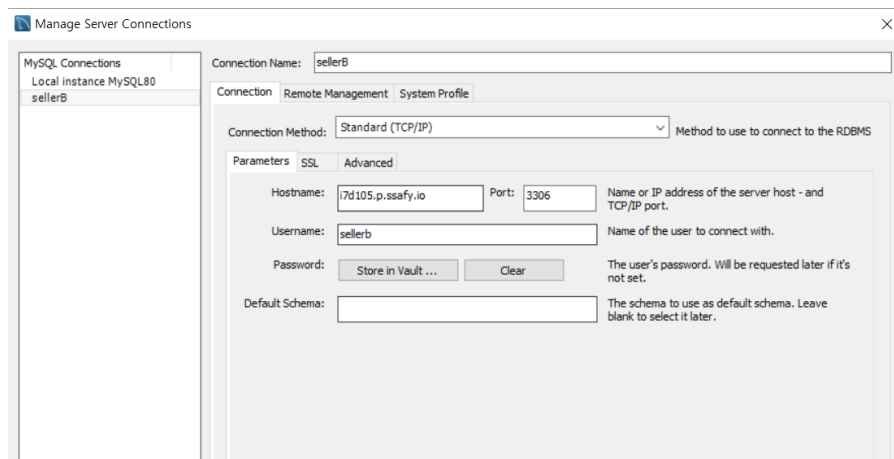
4. DB 계정

가. Mysql Workbench 추가하기



- Mysql Workbench를 열어서 '+' 누르기

나. EC2 계정 정보 넣기



5. 프로퍼티 정의

가. Nginx 세팅

1) 세팅값 변경

```

server {
    location /{
        proxy_pass http://localhost:3000;
    }

    location /api {
        proxy_pass http://localhost:5000/api;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i7d105.p.ssafy.io/fullchain.pem; # managed by
Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7d105.p.ssafy.io/privkey.pem; # managed by
Certbot
    # include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    # ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = i7d105.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name i7d105.p.ssafy.io;
    return 404; # managed by Certbot
}

```

- 모두 WS07P12D105WBEWsellerBWsrcWmainWresources에 위치

나. aws.yml

```

cloud:
  aws:
    # AWS Account Credentials (AWS 접근 키)
    credentials:
      accessKey: access키값
      secretKey: secret키값

  spring:
    datasource:
      driver-class-name: org.mariadb.jdbc.Driver
      url: jdbc:mariadb://도메인:포트번호/database이름?serverTimezone=Asia/Seoul
      username:
      password:
    servlet:
      multipart:
        max-file-size: 5MB
        max-request-size: 5MB

  jwt:
    secret: jwt secret 값

```

다. firebase admin sdk.json

- 1) Firebase Console에서 설정 > 서비스 계정을 엽니다.

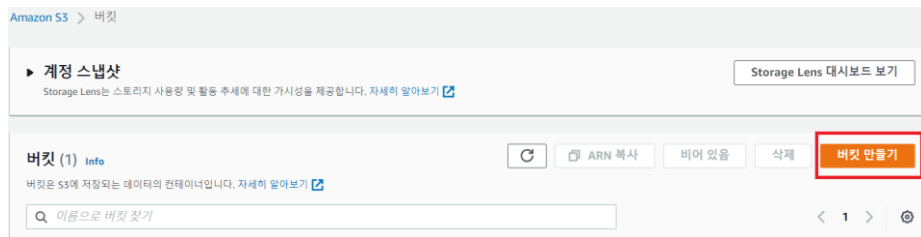
- 2) 새 비공개 키 생성을 클릭한 다음 키 생성을 클릭하여 확인합니다.
- 3) 키가 들어 있는 JSON파일을 연습니다.

6. 외부 서비스

가. AWS S3

S3의 버킷 안에 사진이나 파일 등을 저장하고 주어지는 객체 url을 통해 파일을 이용할 수 있는 클라우드 서비스입니다. Bucket에 대한 권한과 액세스 키가 필요합니다.

1) 버킷 만들기



2) 기본 버킷 설정

일반 구성

버킷 이름

버킷 이름은 전역에서 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. [버킷 이름 지정 규칙 참조](#)

AWS 리전

아시아 태평양(서울) ap-northeast-2

기존 버킷에서 설정 복사 - 선택 사항
다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

3) 버킷 액세스 조건

이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(엑세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. [자세히 알아보기](#)

☐ 모든 퍼블릭 액세스 차단

이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

☐ 새 ACL(엑세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 새로 추가된 버킷 또는 객체에 적용되는 퍼블릭 액세스 권한을 차단하며, 기존 버킷 및 객체에 대한 새 퍼블릭 액세스 ACL 생성을 금지합니다. 이 설정은 ACL을 사용하여 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 권한을 변경하지 않습니다.

☐ 임의의 ACL(엑세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 모든 ACL을 무시합니다.

☐ 새 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지점 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.

☐ 임의의 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 정책을 사용하는 버킷 또는 액세스 지점에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

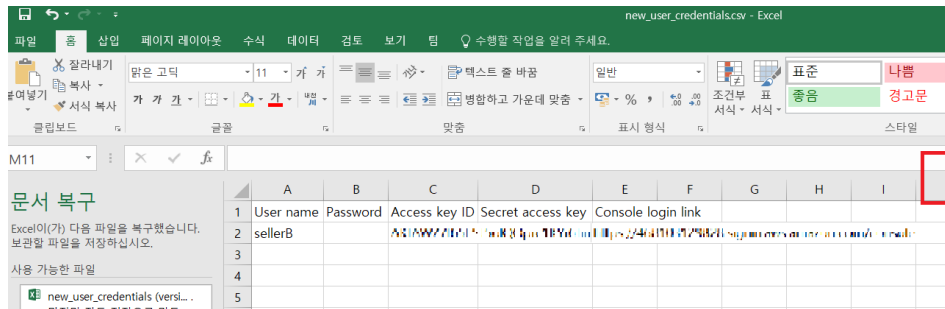


모든 퍼블릭 액세스 차단을 비활성화하면 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있습니다.

정적 웹 사이트 호스팅과 같은 구체적으로 확인된 사용 사례에서 퍼블릭 액세스가 필요한 경우가 아니면 모든 퍼블릭 액세스 차단을 활성화하는 것이 좋습니다.

☐ 현재 설정으로 인해 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있음을 알고 있습니다.

4) 버킷 액세스 키 발급



5) 버킷 정책 추가

버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체에는

```
{
  "Version": "2012-10-17",
  "Id": "Policy1660137646961",
  "Statement": [
    {
      "Sid": "Stmt1660137637300",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::sellerb/*"
    }
  ]
}
```