****

**D105 : SellerB**

삼성SW청년아카데미 구미캠퍼스 7기

공통프로젝트 (7주 : 2022/07/04 ~ 2022/08/19)

**포팅 매뉴얼**

담당 컨설턴트 : 서성수

신혜연(팀장), 배혜연, 이기덕, 이주형, 임채현

**<<목차>>**

[1. 프로젝트 기술 스택 2](#_Toc111670606)

[2. 빌드 상세 내용 3](#_Toc111670607)

[3. 배포 특이사항 4](#_Toc111670608)

[4. DB 계정 5](#_Toc111670609)

[5. 프로퍼티 정의 6](#_Toc111670610)

[6. 외부 서비스 6](#_Toc111670611)

1. 프로젝트 기술 스택
   1. 이슈 관리 : Jira
   2. 형상 관리 : Gitlab, Sourcetree
   3. 커뮤니케이션 : Mattermost, Notion, Discord, Webex
   4. 개발 환경
      1. OS : Windows 10
      2. IDE
         1. Intellij IDE 2022.1.3
         2. Visual Studio Code
         3. Android Studio 7.2.1
      3. Database : MariaDB 10.8.3
      4. Server : AWS EC2 (MobaXterm)
         1. Ubuntu 20.04 LTS
         2. Docker version 20.10.17
   5. 상세 사용
      1. Backend
         1. Java (Azul Zulu version 11.0.16)
         2. Spring Boot Gradle 7.5
         3. Lombok 1.18.24, springdoc-openapi-ui 1.6.6, firebase 9.0.0, spring-cloud-starter-aws 2.3.1
      2. Frontend
         1. HTML5, CSS3, JavaScript(ES6)
         2. React

React 18.2.0

* + - 1. 라이브러리

@ckeditor/ckeditor5-build-classic : ^35.0.1

@ckeditor/ckeditor5-react : ^5.0.2

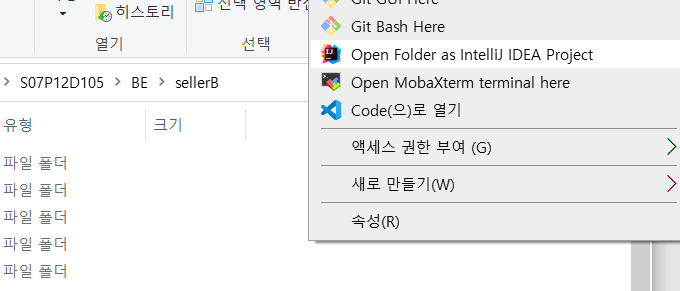
react-js-pagination :^3.0.3

react-router-dom : ^6.3.0

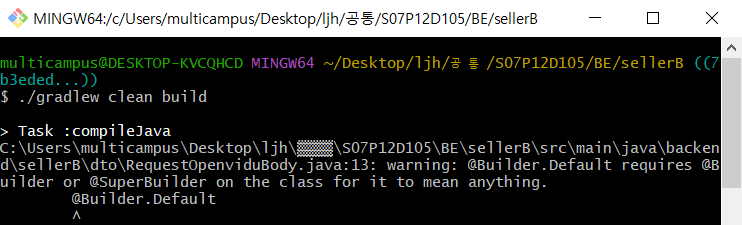
react-toastify : ^8.2.0

* + 1. Android
       1. Android Studio 7.2.1
       2. Gradle

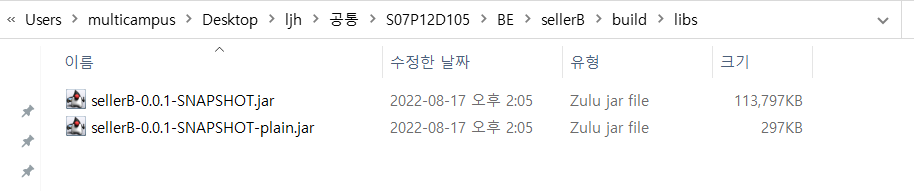
1. 빌드 상세 내용
   1. BE/sellerB 빌드
      1. Git clone 후 intellij로 프로젝트 실행



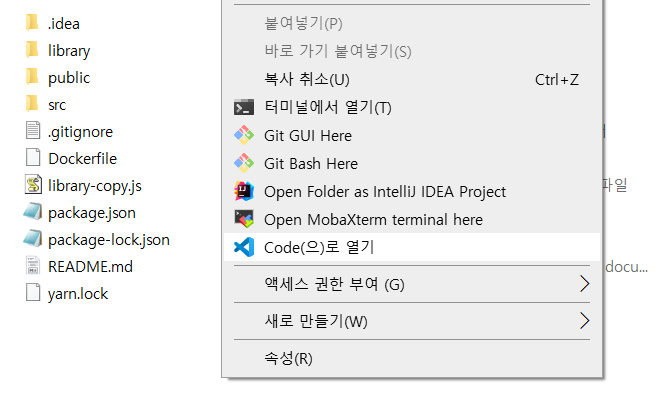
* + 1. Secret.yml 파일을 \S07P12D105\BE\seller\src\main\resources에 이동
    2. ./gradlew clean build 실행



* + 1. \S07P12D105\BE\sellerB\build\libs에 jar파일 생성



* 1. FE/sellerb\_fe
     1. Git clone 후 vscode로 프로젝트 실행



* + 1. cmd에 npm install
    2. cmd에 npm run build
  1. AOS
     1. Git clone
     2. Mac/Linux : ./gradlew tasks, Windows : gradlew.bat tasks
     3. Max/Linux : ./gradlew installDebug, Windows : gradlew.bat installDebug

1. 배포 특이사항
   1. Jenkins shell script
      1. backend



* + 1. frontend



* 1. 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.
  2. EC2 세팅
     1. 도커 설치

sudo apt update

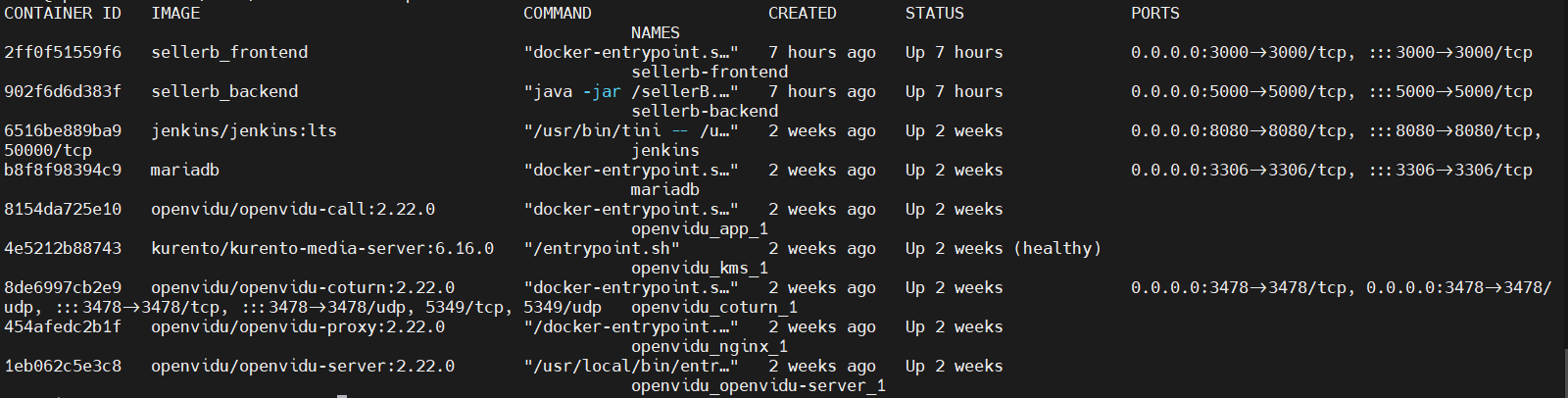
sudo apt install

curl –fsSl <https://download.docker.com/linux/ubuntu/gpg> | sudo apt-key add –

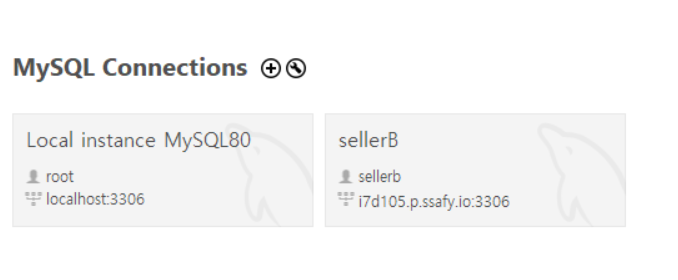
sudo add-apt-repository "deb [arch=amd64] <https://download.docker.com>/linux /ubuntu bionic stable"

sudo apt update

* 1. 최종 도커 컨테이너 상태

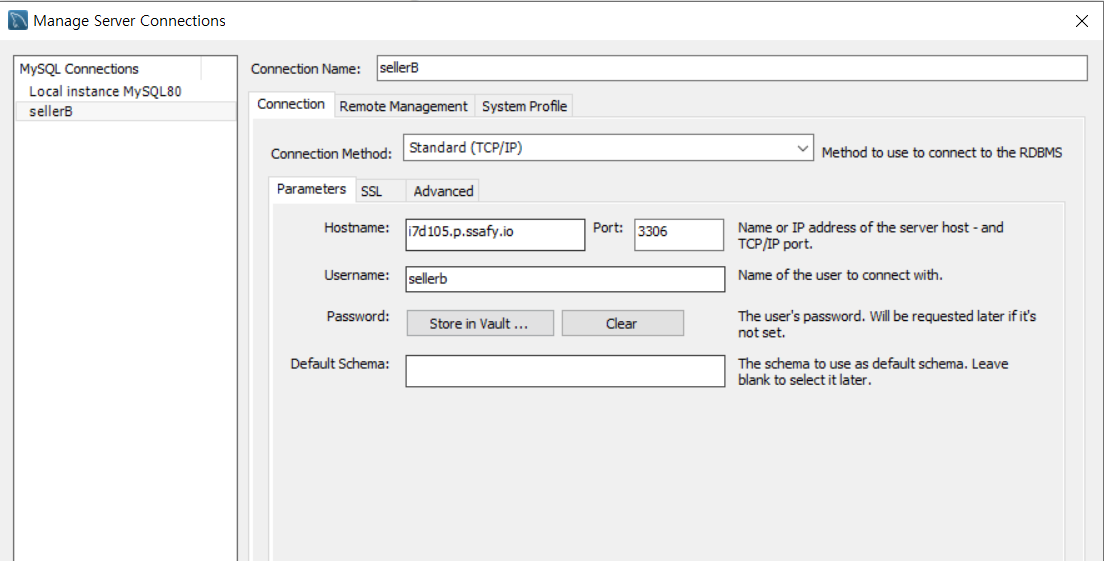


1. DB 계정
   1. Mysql Workbench 추가하기



- Mysql Workbench를 열어서 ‘+’ 누르기

* 1. EC2 계정 정보 넣기



1. 프로퍼티 정의
   1. 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

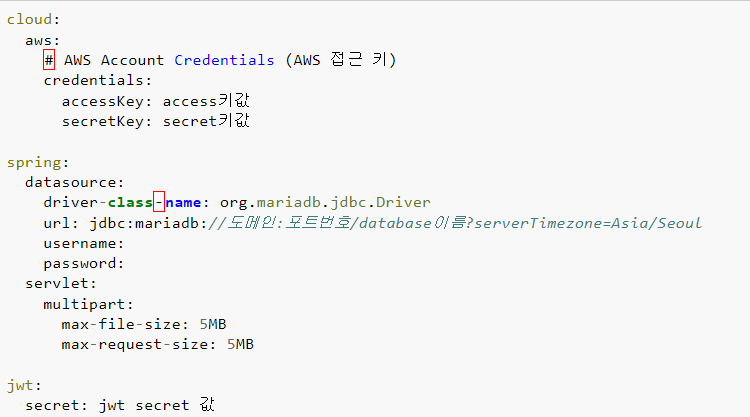
listen 80;

server\_name i7d105.p.ssafy.io;

return 404; # managed by Certbot

}

* 모두 \S07P12D105\BE\sellerB\src\main\resources에 위치
  1. aws.yml

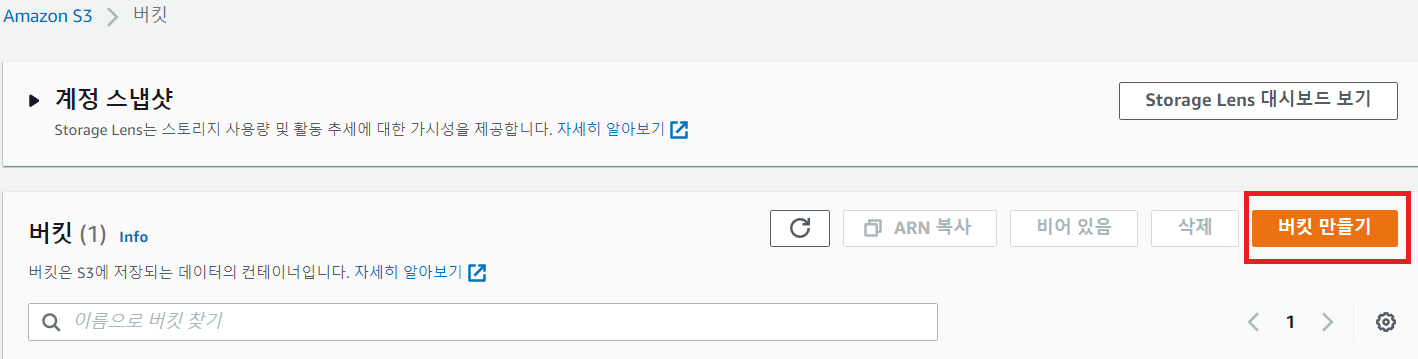


* 1. firebase admin sdk json
     1. Firebase Console에서 설정 > 서비스 계정을 엽니다.
     2. 새 비공개 키 생성을 클릭한 다음 키 생성을 클릭하여 확인합니다.
     3. 키가 들어 있는 JSON파일을 얻습니다.

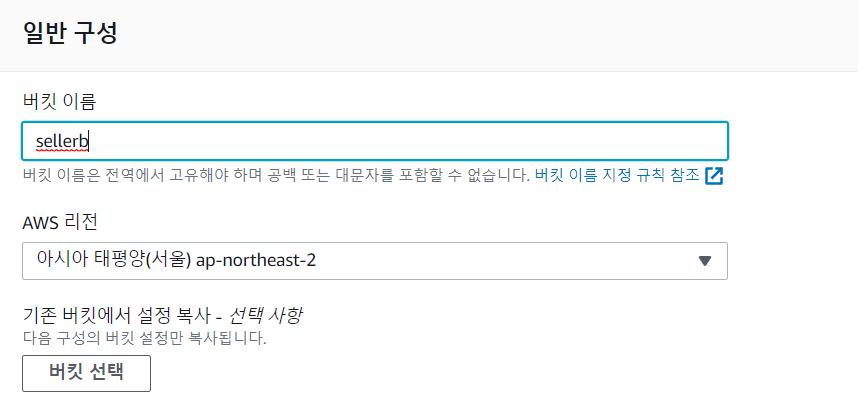
1. 외부 서비스
   1. AWS S3

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



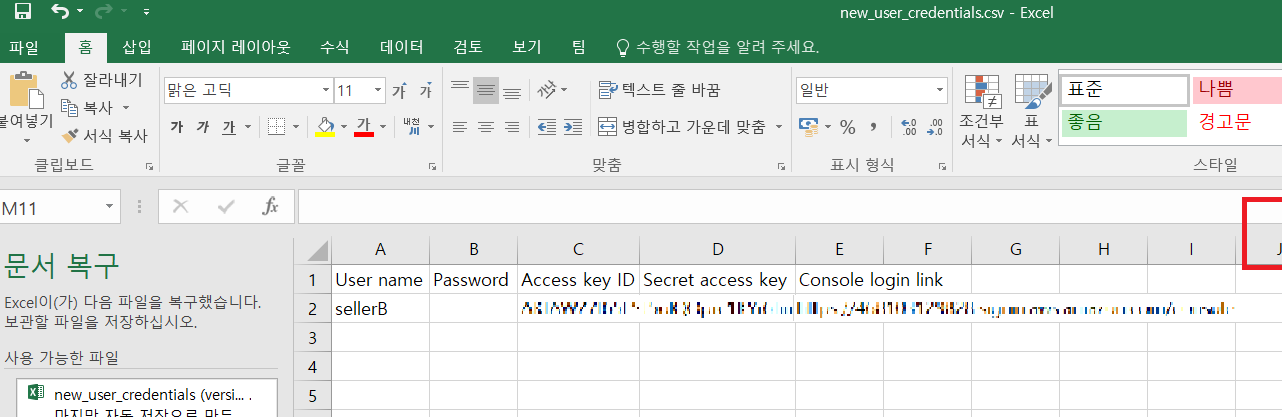
* + 1. 기본 버킷 설정



* + 1. 버킷 액세스 조건



* + 1. 버킷 액세스 키 발급



* + 1. 버킷 정책 추가

