

포팅 메뉴얼



A501 : 알다시피

삼성 청년 SW 아카데미 서울캠퍼스 7기

특화프로젝트 6주, 2022.08.29 ~ 2022.10.07

<<목차>>

1. 기술 스택
2. 배포 특이사항
3. DB 계정
4. 프로퍼티 정의

1. 기술 스택

- 이슈 관리 : JIRA
- 형상 관리 : GitLab
- 협업 도구 : Mattermost, Notion
- 개발 환경
 - IDE
 - IntelliJ IDEA Community Edition 2022.1.3
 - Visual Studio Code 1.69.1
 - Database : MySQL 8.0
 - Server : AWS EC2
 - Backend
 - JDK 8
 - Spring Framework
 - Django Rest Framework
 - Frontend
 - React
 - Typescript
 - Redux
 - Scss
 - 배포

- Nginx
- Docker

2. 빌드, 배포 특이사항

1. 배포 서버 설정

- SSL

```
#letsencrypt 설치
$ sudo apt-get update
$ sudo apt-get install letsencrypt -y
#인증서 발급받기
$ sudo letsencrypt certonly --standalone -d [j7a501.p.ssafy.io]
```

- Nginx default 설정

```
listen 443 ssl;
listen [::]:443 ssl;

server_name www.j7a501.p.ssafy.io;

ssl_certificate /etc/letsencrypt/live/j7a501.p.ssafy.io/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/j7a501.p.ssafy.io/privkey.pem;

root /home/ubuntu/deploy/build;
index index.html;
```

- Docker, Redis 설치

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
sudo docker pull redis
```

- Redis 실행 및 Redis-cli 실행

```
sudo docker run --name aldacipe_redis -d -p 6379:6379 aldacipe_redis
sudo docker exec -it aldacipe_redis redis-cli
```

2. React 배포

```
#localhost -> 배포서버
sed -i 's/http:\/\/localhost:8080/https:\/\/j7a501.p.ssafy.io/g' ./frontend/src/api/config/http-config.ts
# frontend로 들어가서
cd frontend
#빌드를 한 후에
npm install
npm run build
#ssh로 전송
ssh로 jenkins컨테이너에서 build폴더를 배포서버로 전송한다.
```

3. Spring 배포

```
# 기존 실행 중이던 서버 종료
sudo kill -9 `ps -ef|grep SNAPSHOT.jar|awk '{print $2}'` > /dev/null 2>&1 &

# 로컬에서 사용하던 db설정을 배포 기준으로 변경 후 빌드
sed -i 's/spring.datasource.username=a501/spring.datasource.username=ssafy/g' /home/ubuntu/backend/MainServer/src/main/resources/appli
sed -i 's/spring.datasource.password=local_aldacipe/spring.datasource.password=59E9AB522A739D2A9A21/g' /home/ubuntu/backend/MainServ
#clean 후 build
gradle clean
gradle -x test
#ssh로 전송
build 후에 Jenkins 컨테이너 내의 build/libs/jar파일을 배포 서버로 전송
```

```
# 서버 실행
nohup java -jar /home/ubuntu/deploy/MainServer-0.0.1-SNAPSHOT.jar > /dev/null 2>&1 &
```

4. Django 배포

- Django 프로젝트 폴더 내부에 Dockerfile 생성

```
FROM python:3.8

WORKDIR /usr/src/aldacipe

COPY . .

RUN apt-get update

RUN pip install --upgrade pip
RUN pip install -r requirements.txt

EXPOSE 8000
CMD ["python3", "-u", "manage.py", "runserver", "0:8000", "--noreload"]
```

- 생성한 경로에서 Docker Image로 빌드 진행

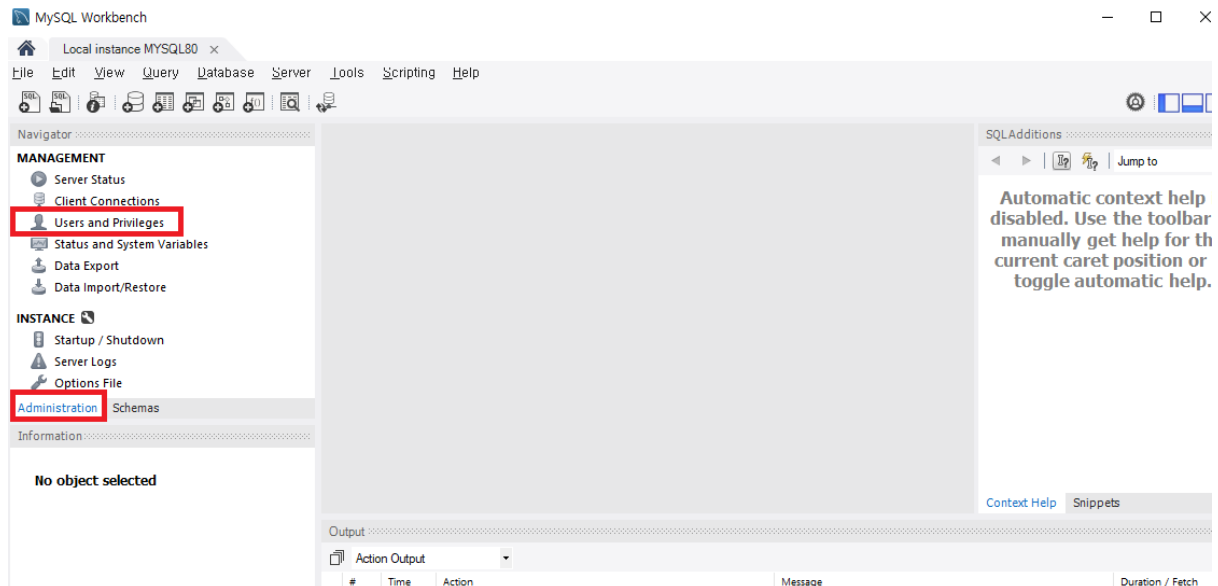
```
sudo docker build . -t aldacipe_recommendation --network=host
```

- 생성된 Docker image를 실행해서 Django 서버 실행

```
sudo docker run --net=host --name aldacipe_recommendation -d -p 8000:8000 aldacipe_recommendation
```

3. DB 계정

- MySQL 계정 추가



- MySQL 계정 이름, 비밀번호 설정 (Login Name : a501 / Password : deploy_aldacipe)

Administration - Users and Privileges

Local instance MySQL80
Users and Privileges

User Accounts

User	From Host
a603	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
newuser	%
root	localhost
ssafy	%

Details for account newuser@%

Login Account Limits Administrative Roles Schema Privileges

Login Name: a501

Authentication Type: Standard

Limit to Hosts Matching: %

Password:

Confirm Password:

Add Account Delete Refresh

Revert Apply

- Roles 설정에서 DBA 체크 후 Apply

Administration - Users and Privileges

Local instance MySQL80
Users and Privileges

User Accounts

User	From Host
a603	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
newuser	%
root	localhost
ssafy	%

Details for account newuser@%

Login Account Limits Administrative Roles Schema Privileges

Role

Role	Description
<input checked="" type="checkbox"/> DBA	grants the rights to perform all tasks
<input type="checkbox"/> MaintenanceAdmin	grants rights needed to maintain service
<input type="checkbox"/> ProcessAdmin	rights needed to assess, monitor, and manage
<input type="checkbox"/> UserAdmin	grants rights to create users logins and
<input type="checkbox"/> SecurityAdmin	rights to manage logins and grant and
<input type="checkbox"/> MonitorAdmin	minimum set of rights needed to monitor
<input type="checkbox"/> DBManager	grants full rights on all databases
<input type="checkbox"/> DBDesigner	rights to create and reverse engineer a
<input type="checkbox"/> ReplicationAdmin	rights needed to setup and manage repl
<input type="checkbox"/> BackupAdmin	minimal rights needed to backup any d

Global Privileges

<input type="checkbox"/> ALTER
<input type="checkbox"/> ALTER ROUTINE
<input type="checkbox"/> CREATE
<input type="checkbox"/> CREATE ROUTINE
<input type="checkbox"/> CREATE TABLESPACE
<input type="checkbox"/> CREATE TEMPORARY TABLES
<input type="checkbox"/> CREATE USER
<input type="checkbox"/> CREATE VIEW
<input type="checkbox"/> DELETE
<input type="checkbox"/> DROP
<input type="checkbox"/> EVENT
<input type="checkbox"/> EXECUTE
<input type="checkbox"/> FILE

Revoke All Privileges

Add Account Delete Refresh

Revert Apply

4. 프로퍼티 정의

- /etc/nginx/sites-available/default

```
server {
    #if ($host = j7a501.p.ssafy.io) {
    #    return 301 https://$host$request_uri;
    #}

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name www.j7a501.p.ssafy.io;
    return 301 https://www.j7a501.p.ssafy.io$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
```

```

server_name www.j7a501.p.ssafy.io;

ssl_certificate /etc/letsencrypt/live/j7a501.p.ssafy.io/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/j7a501.p.ssafy.io/privkey.pem;

root /home/ubuntu/deploy/build;
index index.html;

location / {
    try_files $uri $uri/ /index.html;
}

location /api {
    proxy_pass http://localhost:8080;
    proxy_redirect off;
    charset utf-8;

    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-proto $scheme;
    proxy_set_header    X-NginX-Proxy true;
}
}

```

Jenkins 접속

```

http://j7a501.p.ssafy.io:9090
id : a501
pw : j7a501.p.ssafy.io

front item : a501_front
back item : a501_back

```