D106: 酒酒클럽

삼성청년SW아카데미 구미캠퍼스 7기

특화프로젝트 (빅데이터 분산)

포팅매뉴얼

담당 컨설턴트: 강시몬

정종일(팀장), 김민정, 박지은, 오도석, 최연지

<<목차>>

1. 프로젝트 기술 스택

1) 이슈관리: Jira
2) 형상관리: Gitlab
3) 커뮤니케이션: Mattermost, Notion, Webex
4) 개발 환경
    a) OS: Windows 10
    b) IDE
        i) IntelliJ IDEA 2022.1.4
        ii) Visual Studio Code 1.69.0
    c) Database: mysql Ver 8.0.30
    d) Server: AWS EC2(MobaXterm 22.1)
        i) Ubuntu 20.04.2 LTS
        ii) Jenkins 2.346.2
    e) Storage: AWS S3
5) 상세내용
    a) BackEnd
        i) Java 1.8.0
        ii) Spring Boot Gradle 7.5
        iii) Lombok 1.18.24, Swagger2
        iv) jjwt 0.9.1
    b) FrontEnd
        i) Vue: 2.6.14
        ii) Vuex: 3.6.2
        iii) vue-router: 3.5.1
    c) Big Data Dispersion

i) Python 3.10.5

ii) hadoop-3.2.3

iii) sqoop-1.4.7

iv) spark-3.2.2

2. 빌드 상세내용

1) EC2환경변수

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/ubuntu/hadoop
export PDSH_RCMD_TYPE=ssh
#Hadoop Related Options export HADOOP_HOME=/home/hdoop/hadoop-3.2.1

export HADOOP_CLASSPATH=/home/ubuntu/sqoop/lib/
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

export SQOOP_HOME=/home/ubuntu/sqoop
export SQOOP_CONF_DIR=$SQOOP_HOME/conf

export SPARK_HOME=/home/ubuntu/spark-3.2.2

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin:$SQOOP_HOME/bin:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

3. 배포 특이사항

- Blue Green 무중단 자동배포

1) Gitlab Event -> Jenkins Webhook

Execute shell

```
cd /var/jenkins_home/workspace/ssafyD106
chmod -R 777 BE
cd BE/Drink
./gradlew clean build

chmod 400 /var/jenkins_home/J7D106T.pem
scp -v -o StrictHostKeyChecking=no -i /var/jenkins_home/J7D106T.pem /var/jenkins_home/workspace/ssafyD106/BE/Drink/build/libs/Drink-0

cd /var/jenkins_home/workspace/ssafyD106
chmod -R 777 FE
cd FE/joojooclub
rm -rf dist
npm install
npm run build
tar -zcvf front.tar.gz dist
scp -v -o StrictHostKeyChecking=no -i /var/jenkins_home/J7D106T.pem /var/jenkins_home/workspace/ssafyD106/FE/joojooclub/front.tar.gz
```

빌드 후 조치

```
Source files -> /var/jenkins_home/workspace/ssafyD106/BE/Drink/build/libs/Drink-0.0.1-SNAPSHOT.jar
Remove prefix -> /var/jenkins_home/workspace/ssafyD106/BE/Drink/build/libs
Remote directory -> /home/ubuntu/app-server
Exec command -> /home/ubuntu/app-server/deploy.sh


Source files -> /var/jenkins_home/workspace/ssafyD106/FE/joojooclub/front.tar.gz
Remove prefix -> /var/jenkins_home/workspace/ssafyD106/FE/joojooclub
Remote directory -> /home/ubuntu/app-server2
Exec command ->
cd /home/ubuntu/app-server2
tar -zxvf front.tar.gz
```

2) Deploy.sh 실행으로 jar build및 run , 컨테이너 변경으로 port Switch

Dockerfile

```
FROM openjdk:8
ARG IDLE_PROFILE
ARG JAR_FILE=Drink-0.0.1-SNAPSHOT.jar
RUN echo $JAR_FILE
ENV ENV_IDLE_PROFILE=$IDLE_PROFILE
COPY ${JAR_FILE} app.jar
RUN echo $ENV_IDLE_PROFILE
ENTRYPOINT ["java","-Dspring.profiles.active=${ENV_IDLE_PROFILE}","-jar","/app.jar"]
```

Deploy.sh

```
#!/bin/bash
echo "> 현재 구동중인 profile 확인"
CURRENT_PROFILE=$(curl -s http://j7d106.p.ssafy.io/api/utils/profile)
echo "> $CURRENT_PROFILE"

if [ $CURRENT_PROFILE == production-set1 ]
then
  IDLE_PROFILE=production-set2
  IDLE_PORT=8889
elif [ $CURRENT_PROFILE == production-set2 ]
then
  IDLE_PROFILE=production-set1
  IDLE_PORT=8888
else
  echo "> 일치하는 Profile이 없습니다. Profile: $CURRENT_PROFILE"
  echo "> set1을 할당합니다. IDLE_PROFILE: set1"
  IDLE_PROFILE=production-set1
  IDLE_PORT=8888
fi

IMAGE_NAME=app_server
TAG_ID=$(docker images | sort -r -k2 -h | grep "${IMAGE_NAME}" | awk 'BEGIN{tag = 1} NR==1{tag += $2} END{print tag}')

echo "> 도커 build 실행 : docker build --build-arg IDLE_PROFILE=${IDLE_PROFILE} -t ${IMAGE_NAME}:${TAG_ID} ."
docker build --build-arg IDLE_PROFILE=${IDLE_PROFILE} -t ${IMAGE_NAME}:${TAG_ID} /home/ubuntu/app-server

echo "> $IDLE_PROFILE 배포"
echo "> 도커 run 실행 :  sudo docker run --name $IDLE_PROFILE -d --rm -p $IDLE_PORT:${IDLE_PORT} ${IMAGE_NAME}:${TAG_ID}"
docker run --name $IDLE_PROFILE -d --rm -p $IDLE_PORT:${IDLE_PORT} ${IMAGE_NAME}:${TAG_ID}

echo "> $IDLE_PROFILE 10초 후 Health check 시작"
echo "> curl -s http://j7d106.p.ssafy.io:$IDLE_PORT/api/actuator/health "
sleep 10

for retry_count in {1..10}
do
  response=$(curl -s http://j7d106.p.ssafy.io:$IDLE_PORT/api/actuator/health)
  up_count=$(echo $response | grep 'UP' | wc -l)

  if [ $up_count -ge 1 ]
  then
    echo "> Health check 성공"
    break
  else
    echo "> Health check의 응답을 알 수 없거나 혹은 status가 UP이 아닙니다."
    echo "> Health check: ${response}"
  fi

  if [ $retry_count -eq 10 ]
  then
    echo "> Health check 실패. "
    echo "> Nginx에 연결하지 않고 배포를 종료합니다."
    exit 1
  fi

  echo "> Health check 연결 실패. 재시도..."
  sleep 10
done

echo "> 스위칭을 시도합니다..."
sleep 5

/home/ubuntu/app-server/switch.sh
```

Switch.sh

```
#!/bin/bash
echo "> 현재 구동중인 Port 확인"
CURRENT_PROFILE=$(curl -s http://j7d106.p.ssafy.io/api/utils/profile)

if [ $CURRENT_PROFILE == production-set1 ]
then
  CURRENT_PORT=8888
  IDLE_PORT=8889
elif [ $CURRENT_PROFILE == production-set2 ]
then
  CURRENT_PORT=8889
  IDLE_PORT=8888
else
  echo "> 일치하는 Profile이 없습니다. Profile:$CURRENT_PROFILE"
  echo "> 8888을 할당합니다."
  IDLE_PORT=8888
fi

echo "> 현재 구동중인 Port: $CURRENT_PORT"
echo "> 전환할 Port : $IDLE_PORT"
echo "> Port 전환"
echo "set \$service_url http://j7d106.p.ssafy.io:${IDLE_PORT};" | sudo tee /etc/nginx/conf.d/service-url.inc

echo "> ${CURRENT_PROFILE} 컨테이너 삭제"
sudo docker stop $CURRENT_PROFILE

echo "> Nginx Reload"

sudo service nginx reload
```

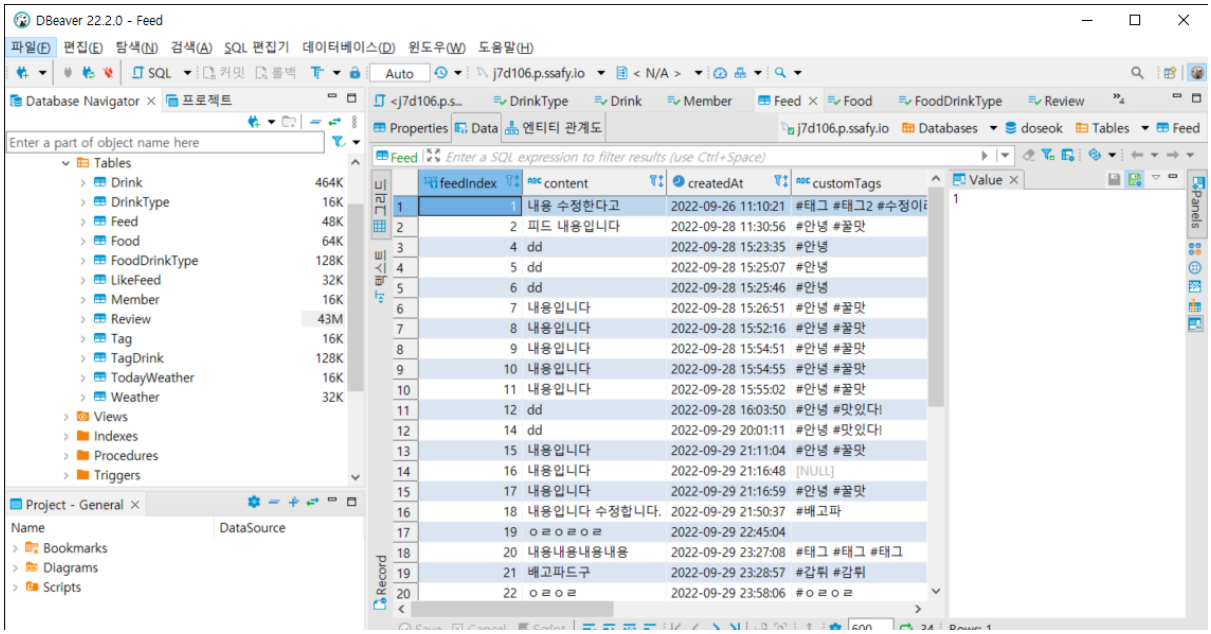
4. DB 계정

1) MySQL

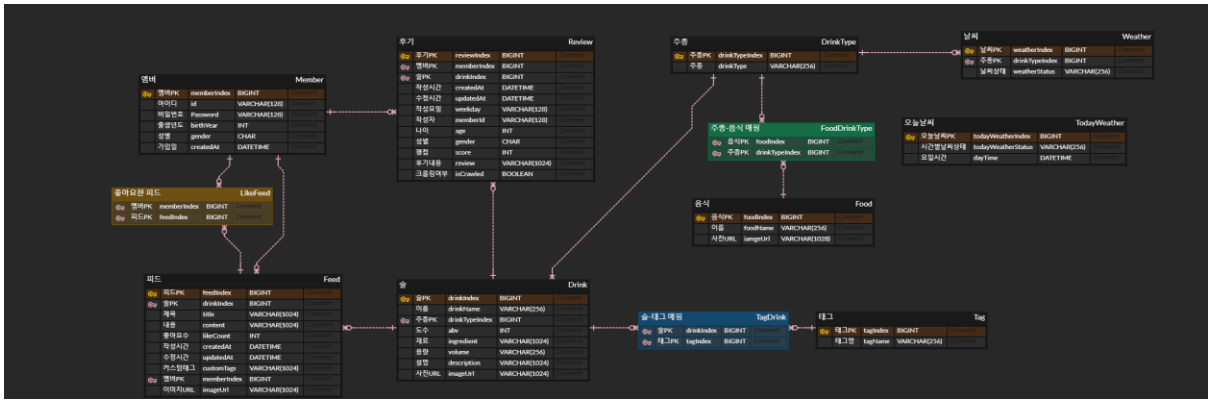
Id : hive

Pwd : nh+7&(tUa7p[NYcAae{Sno&LM698=1

Tables



ERD


Member
Review
DrinkType
Weather
LikeFeed
FoodDrinkType
TodayWeather
Food
Feed
Drink
TagDrink
Tag

5. 프로퍼티 정의

1) Nginx default

```
server {
  listen 80 default_server;
  listen [::]:80 default_server;
  root   /home/ubuntu/app-server2/dist;
  index  index.html index.htm;
  server_name j7d106.p.ssafy.io;
  include /etc/nginx/conf.d/service-url.inc;    ->    proxy_pass가 되는 url이 계속해서 바뀌는데 inc파일에 계속해서 다르게 저장한다.

  resolver 127.0.0.53 valid=5s;
  set $elb " j7d106.p.ssafy.io ";

  location / {
    client_max_body_size 0;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Proto https;
    try_files $uri /index.html;
  }
   location /api {
        client_max_body_size 0;
        proxy_pass $service_url;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  }


listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/j7d106.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/j7d106.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
}
```

2) Hadoop

Core-site.xml

```
<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/ubuntu/tmpdata</value>

</property>

<property>

<name>fs.default.name</name>

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>
```

Hdfs-site.xml

```
<configuration>
<property>

<name>dfs.data.dir</name>

<value>/home/ubuntu/dfsdata/namenode</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>/home/ubuntu/dfsdata/datanode</value>

</property>

<property>

<name>dfs.replication</name> //자신이 가지고 있는 data를 몇개나 복제할 것인가

<value>1</value>

</property>
</configuration>
```

Mapred-site.xml

```
<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>
```

Yarn-site.xml

```
<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name> <value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>0.0.0.0</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>0.0.0.0:8032</value>

</property>

<property>

<name>yarn.web-proxy.address</name>

<value>0.0.0.0:8089</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name> <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_

</property>

</configuration>
```