

## E101 : 업서사전

삼성청년SW아카데미 부울경캠퍼스 7기

특화프로젝트 [7주 2022.07.22 ~ 2022.10.07]

## 포팅 매뉴얼

담당 컨설턴트 : 이태희

박정현[팀장], 민경대, 박제학, 손지민, 전현우, 최년우

## === 목차 ===

1. 기술스택	*****	3
2. 빌드 상세내용	*****	4
3. 배포 특이사항	*****	5
4. DB 계정	*****	6
5. 프로퍼티 정의	*****	7
6. 외부 서비스	*****	10

---

업서사전 서비스를 이용하여 기부도 하고 직접 그린 그림으로 만들어진 업서를 수집해 보세요.

그림을 못 그려도 괜찮아요! 스케치만 하시면 AI가 자동으로 사진처럼 만들어 줍니다.

화풍을 선택하여 사진을 그림으로 그린 것처럼 변경하여 저장 할 수 있습니다.

직접 만든 그림으로 기부가 필요한 재단에 업서를 만들어서 기부할 수 있습니다.

# 1. 프로젝트 기술 스택

가. 이슈관리 : Jira

나. 형상관리 : Gitlab

다. 커뮤니케이션 : Mattermost, Notion

라. 개발 환경

A. OS : Windows 10

B. IDE

i. IntelliJ 2022.1.3 (Ultimate Edition)

ii. Visual Studio Code 1.70.0v

C. DataBase : MySQL Workbench 8.0CE

D. Server : AWS EC2 (MobaXterm)

i. Ubuntu 20.04 LTS

ii. Docker 20.10.17v (server/client 동일)

iii. jenkins

E. 상세 사용

i. BackEnd

1) Java (OpenJDK Zulu 8.33.0.1-win64)

2) Spring Boot Gradle 6.7

ii. FrontEnd

1) HTML5, CSS, JavaScript(ES6)

2) vuev 3.2.38

3) vue-router v4.1.5, vuex v4.0.2, axios v0.27.2, html2canvas v1.4.1

iii. AWS S3

1) JupyterHup

2) Python 3.9.7

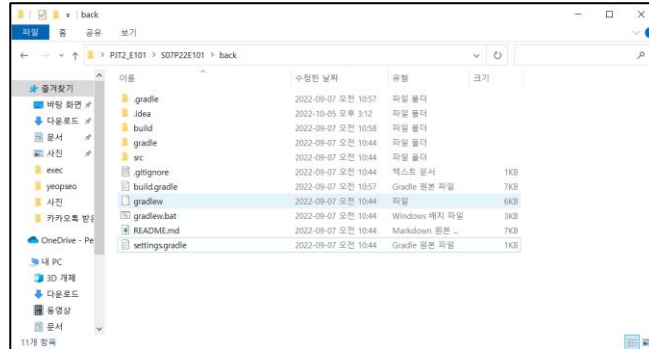
iv. AWS S3

## 2. 빌드 상세내용

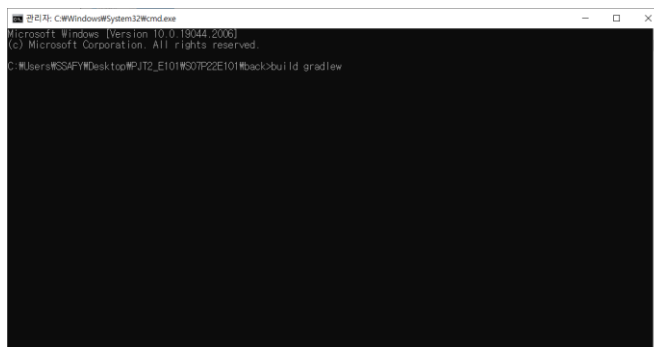
### 2-1. 백엔드 빌드 방법

가. Git clone <https://lab.ssafy.com/s07-ai-image-sub2/S07P22E101.git>

나. 클론 한 폴더에서 S07P22E101 > back 폴더에 들어갑니다.



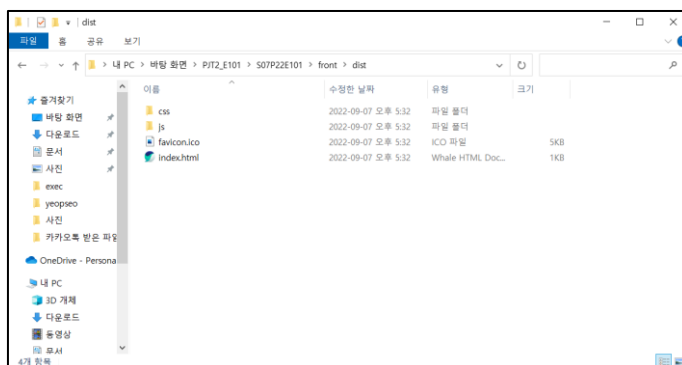
다. 주소창에서 cmd를 쳐서 명령 프롬프트창을 열어 [gradlew build] 명령어로 빌드합니다.



### 2-2. 프론트엔드 빌드 방법

가. frontend 폴더에서 [npm run build] 명령어를 사용하여 프론트엔드를 빌드합니다..

나. build 폴더에서 build된 파일들을 확인합니다.



### 3. 배포 특이사항

- Jenkins 로 자동 배포

#### 1. 프로젝트 빌드

**Build Steps**

≡ Execute shell ?

Command

See [the list of available environment variables](#)

```
cd back
chmod +x gradlew
./gradlew clean build
```

고급...

Add build step ▾

#### 2. 빌드 후 sh 파일 실행 > 실행되고있는 서버 kill > 빌드 파일 백그라운드에서 실행

빌드 후 조치

≡ Send build artifacts over SSH ?

SSH Publishers

SSH Server Name ?

aws

고급...

Transfers

Transfer Set

Source files ?

back/build/libs/\*.jar

Remove prefix ?

back/build/libs

Remote directory ?

/target

Exec command ?

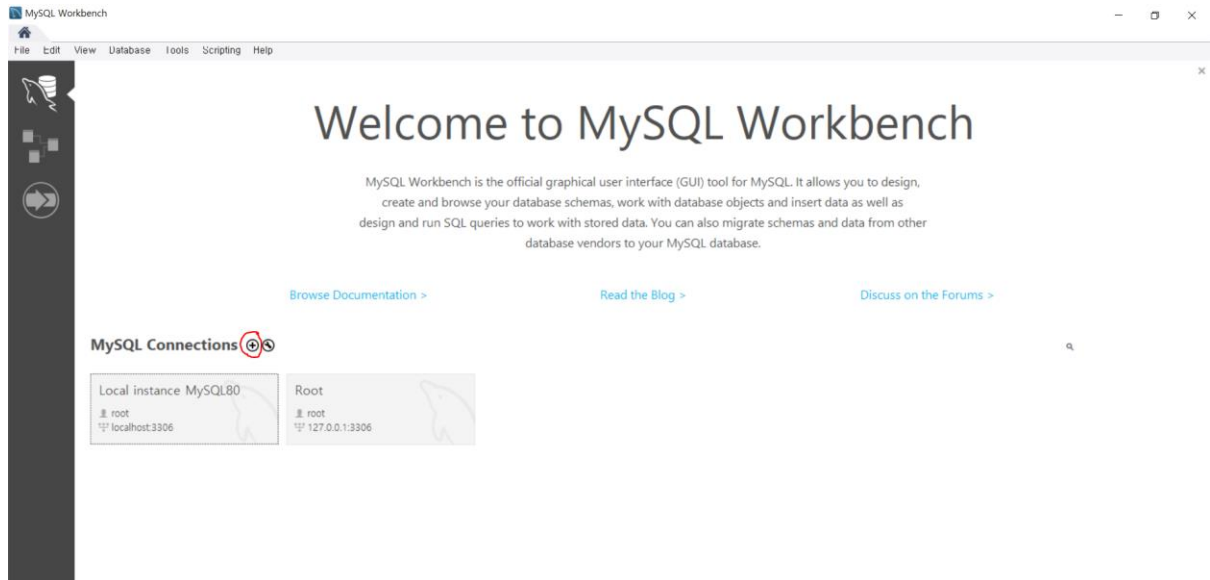
sudo sh /home/ubuntu/A/bootrun.sh

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

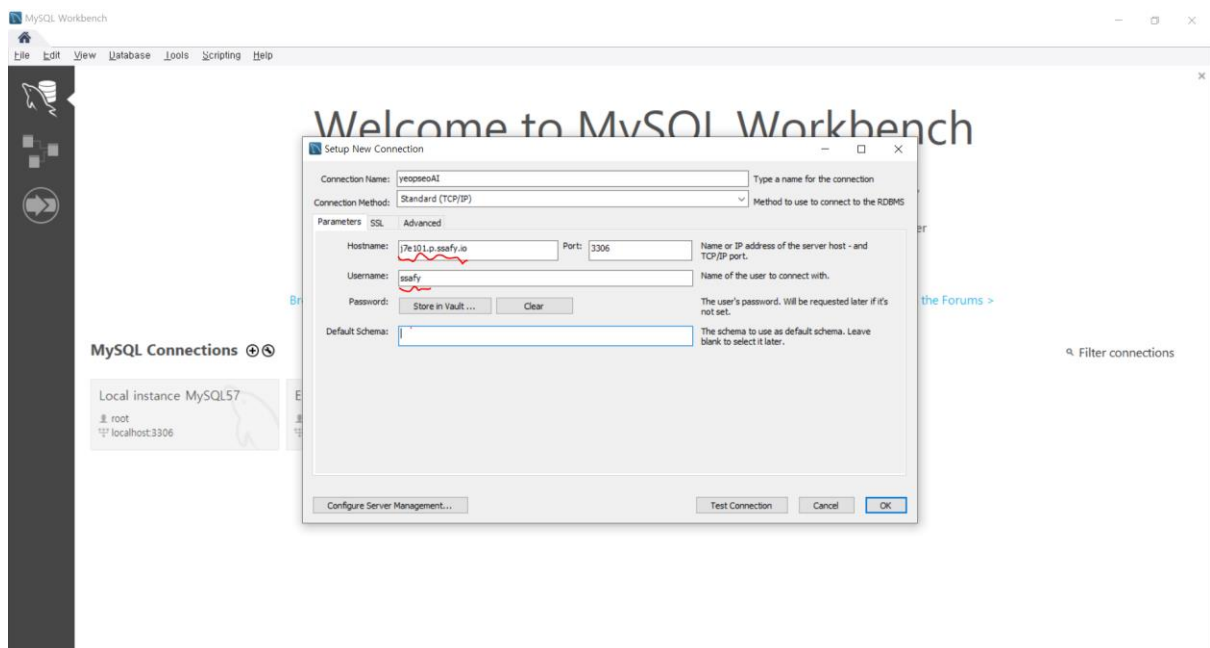
고급...

## 4. DB 계정

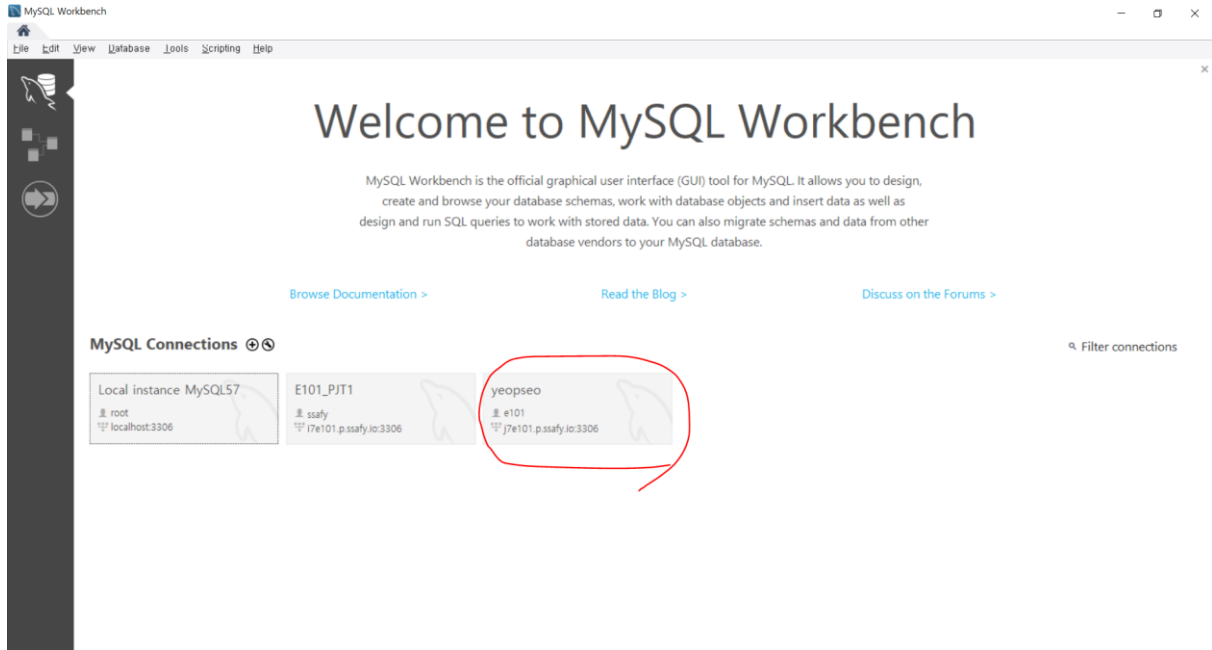
가. MySQL workbench 계정 등록



>> MySQL workbench 에 새로운 계정 등록을 위해 + 버튼 클릭



>> EC2 계정 정보를 사용하여 연동된 DB를 불러옴



>> 저장된 connection 클릭하여 DB 접속

## 5. 프로퍼티 정의

가. Certbot SSL 인증서 발급

A. Certbot 설치

```
$ sudo snap install --classic certbot
```

B. SSL 인증서 발급

```
$ sudo certbot --nginx
```

나. Nginx 세팅

A. 서버의 패키지 목록 업데이트

```
$ sudo apt update
```

B. Nginx 설치

```
$ sudo apt install nginx
```

## C. Nginx 설정 파일 수정

```
user ubuntu;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;
```

```
    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml
```

```
    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;

    # upload size change
    client_max_body_size 10M;

    # server timeout setting
    proxy_read_timeout 380;
    proxy_connect_timeout 380;
    proxy_send_timeout 380;
}
```



## 다. AWS EC2 세팅

### A. MySQL 설치

#### i. 세팅을 위해 최신 상태 업데이트

```
$ sudo apt-get install
```

#### ii. MySQL 설치

```
$ sudo apt-get install mysql-server (진행 중에 Y)
```

#### iii. MySQL 접속

```
$ sudo mysql
```

#### iv. MySQL에서 계정 생성 후 권한 설정

```
> use mysql  
> CREATE USER '<ID>'@'%' IDENTIFIED BY '<PASSWORD>';  
> GRANT ALL PRIVILEGES ON *.* TO '<ID>'@'%' WITH GRANT OPTION;  
> FLUSH PRIVILEGES;  
> EXIT
```

#### v. 외부 접속 허용

```
$ cd /etc/mysql/mysql.conf.d (폴더 이동)  
$ sudo vi mysqld.cnf (파일 수정)
```

#### vi. 수정할 내용

```
bind-address = 0.0.0.0
```

#### vii. 포트 열고 재시작

```
$ sudo ufw allow 3306 (포트 열기)  
$ sudo systemctl restart mysql.service (재시작)
```

## B. Docker 설치

### i. 세팅을 위해 최신 상태 업데이트

```
$ sudo apt-get update
```

### ii. Docker 설치

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)"
-o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

## 6. 외부 서비스

### 가. AWS S3

#### A. 버킷 생성

버킷 (2) <a href="#">Info</a>		<a href="#">↺</a>	<a href="#">ARN 복사</a>	<a href="#">비어 있음</a>	<a href="#">삭제</a>	<a href="#">버킷 만들기</a>
버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기 <a href="#">🔗</a>						
<input type="text" value="이름으로 버킷 찾기"/>						
<div>&lt; 1 &gt; <a href="#">🔍</a></div>						
	이름	AWS 리전	액세스	생성 날짜		
<input type="radio"/>	bucket-e101-yeopseo	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2022. 9. 6. pm 2:08:16 PM KST		

#### B. 버킷 액세스 설정

### 퍼블릭 액세스 차단(버킷 설정)

퍼블릭 액세스는 ACL(엑세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 모든 S3 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 [모든 퍼블릭 액세스 차단]을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지정에만 적용됩니다. AWS에서는 [모든 퍼블릭 액세스 차단]을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 버킷 또는 내부 객체에 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 지정 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

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

☐ 임의의 퍼블릭 버킷 또는 액세스 지정 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 지정에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

나. AI 서버 로컬에서 실행

- A. Git clone <https://lab.ssafy.com/s07-ai-image-sub2/S07P22E101.git>
- B. VSCODE로 클론 한 폴더에서 S07P22E101 > ai 폴더를 열어줍니다.
- C. 터미널을 열어서 ai/yeopseo 폴더로 들어가 [pip install -r requirements.txt] 명령어를 사용하여 사용할 모듈을 설치 해 줍니다.
- D. 터미널에서 ai/yeopseo/yeopseo 폴더로 들어가서  
git clone <https://github.com/NVlabs/SPADE.git> 을 해줍니다.
- E. 터미널에서 다시 SPADE 폴더로 들어가서 아래와 같은 작업들을 실행합니다.

This code requires PyTorch 1.0 and python 3+. Please install dependencies by

```
pip install -r requirements.txt
```

This code also requires the Synchronized-BatchNorm-PyTorch rep.

```
cd models/networks/  
git clone https://github.com/vacancy/Synchronized-BatchNorm-PyTorch  
cp -rf Synchronized-BatchNorm-PyTorch/sync_batchnorm .  
cd ../../
```

```
cd checkpoints  
tar xvf checkpoints.tar.gz  
cd ../
```

- F. ai/yeopseo 에서 [python manage.py makemigrations], [python manage.py migrate]를 차례로 실행합니다.
- G. [python manage.py runserver]를 실행해 ai서버를 실행합니다.