



SWING 포팅 매뉴얼

특화 프로젝트 A405

이연수, 고명진, 성다연, 유다윗, 이상화, 인수경

서비스 명 : SWING(Study With Image N Game)!

목차

1. 기술 스택 및 버전
 2. CI/CD 설정 및 빌드
 3. Database 설정
-

1. 기술 스택 및 버전

Front-End

- npm: 8.19.3
- node.js: 18.13.0
- react: 18.2.0
- react-router-dom: 6.9.0
- recoil: 0.7.7
- styled-components: 5.3.9
- emotion: 11.10.6
- react-bootstrap-icons: 1.10.2
- axios: 1.3.4
- recoil-persist: 4.2.0

Back-end

- Jenkins 2.387.1
- Docker 23.0.1, build a5ee5b1
- NginX 1.18.0
- Java 8(zulu 1.8.0_192)
- Spring Boot 2.7.9
- Gradle 7.6.1
- Python 3.9.2
- Django 3.2.18

AI

- Keras 2.10.0
- Scikit-learn 1.2.2
- Sentence-transformers 2.2.2
- Tensorflow 2.10.0
- Torch 1.13.1

Port

- React 3000
 - Jenkins 9090
 - Spring Boot 8080
 - Django 8000
 - MySQL 3306
-

2. CI/CD 설정 및 빌드

ec2 ubuntu 환경

```
sudo apt-get update
sudo apt-get upgrade
```

1) MySQL 설치

```
// docker를 이용하여 MySQL 이미지 당겨옴
docker pull mysql

// 3306 포트로 MySQL 도커라이즈
docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=asdf1234 --name mysql mysql --character-set-server=utf8mb4 --collation-server=utf8mb4

// MySQL bash 접속
docker exec -it mysql bash

// root 유저 사용
mysql -u root -p

// mysql 데이터베이스 접속
use mysql;

// 유저 생성
create user 'swing'@'%' identified by 'asdf1234';

// 권한 부여
grant all privileges on *.* to 'swing'@'%';

exit
```

2) nginx 설치

```
sudo apt install nginx
```

3) python 3.9 설치

```
sudo apt-get install python3.9
```

4) Docker 설치

```
apt-get remove docker docker-engine docker.io containerd runc

apt-get update

apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update

apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

5) Dockerize Jenkins

```
sudo chmod 666 /var/run/docker.sock
docker run -u 0 -d -p 9090:8080 -p 50000:50000 -v /var/jenkins:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock --name jenkins
```

6) Jenkins 기본 설정하기

1. j8a405.p.ssafy.io:9090 접속 후 초기 비밀번호 확인을 위해 우분투에서 var/jenkins_home/secrets/initialAdminPassword 접근
2. admin 계정으로 swing/asdf1234 계정 생성
3. Jenkins URL로 http://j8a405.p.ssafy.io:9090 설정
4. Jenkins 플러그인으로 GitLab, Generic Webhook Trigger, GitLab API, GitLab Authentication, Docker, Docker API, Docker Pipeline, Docker C

7) Jenkins 안에 Docker 설치

```
docker exec -it jenkins /bin/bash

apt-get remove docker docker-engine docker.io containerd runc

apt-get update

apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update

apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

8) Jenkins에 item으로 front 추가하기

1. 소스 코드 관리에서 Git 선택
2. Repository URL에 `https://lab.ssafy.com/s08-ai-image-sub2/S08P22A405.git` 입력
3. Build Steps에서 Execute Shell 선택 후 아래와 같이 입력

```
cd front

docker build -t front .

docker run -d -p 3000:3000 --name front front

4. 저장 후 빌드
```

9) Jenkins에 item으로 back-springboot 추가하기

- 2번까지는 front와 같음
3. Build Steps에서 Execute Shell 선택 후 아래와 같이 입력

```
cd back-springboot

chmod +x gradlew
./gradlew clean build

docker build -t back-springboot .

docker run -d -p 8080:8080 --name back-springboot back-springboot

4. 저장 후 빌드
```

10) Jenkins에 item으로 back-django 추가하기

- 2번까지는 front와 같음
3. Build Steps에서 Execute Shell 선택 후 아래와 같이 입력

```
cd back-django

docker build -t back-django .

docker run -d -p 8000:8000 --name back-django back-django

4. 저장 후 빌드
```

3. Database 설정

S3 설정

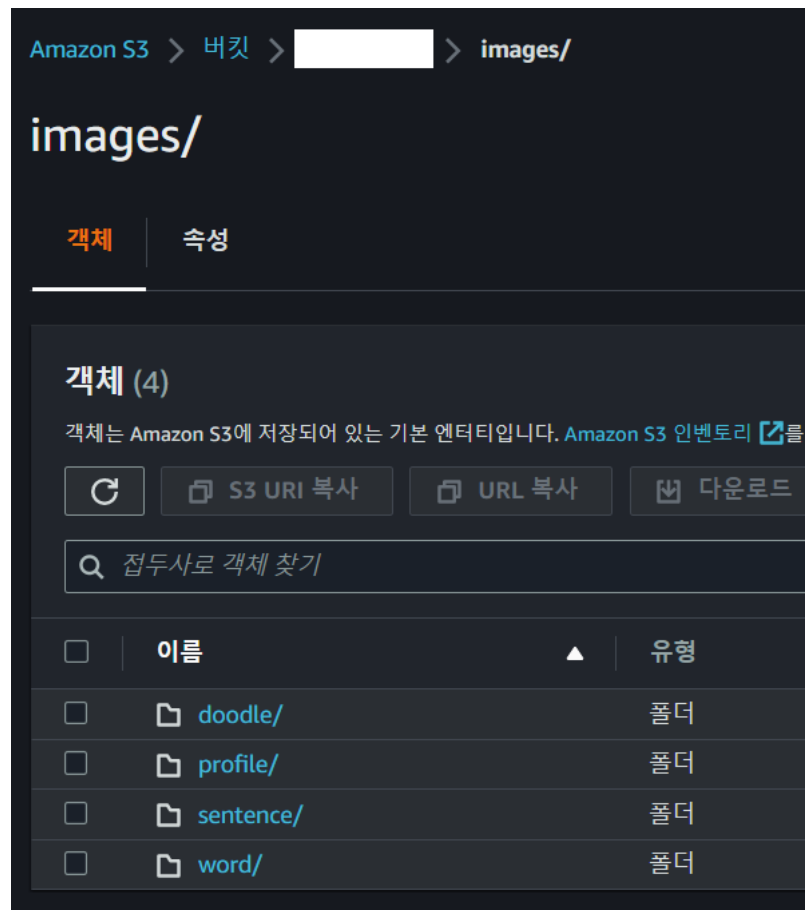
SWING의 이미지들(프로필 사진, 게임 이미지 등)은 S3 파일 서버에 저장해야 하기 때문에 Spring Boot의 `application.yml` 파일에서 크루덴셜과 버킷 이름을 입력해주셔야 합니다.

```
cloud:
  aws:
    credentials:
      access-key: 
      secret-key: 
    s3:
      bucket : 
      region:
        static: ap-northeast-2
      stack:
        auto: false
```

각각의 이미지들은 아래와 같은 경로에 목적에 맞게 위치하게 됩니다.

1. 프로필 이미지(`[버킷 이름]/images/profile`)
2. Sentency, 오답노트 문장(`[버킷 이름]/images/sentence`)
3. Hi-Five, 오답노트 단어(`[버킷 이름]/images/word`)
4. Speedoodle(`[버킷 이름]/images/doodle`)

S3 파일 서버의 profile, word, sentence 폴더에 이미지 덤프 파일을 업로드하셔야 합니다.
(이미지들은 dump 폴더안의 S3 폴더에 있습니다.)



MySQL 설정

S3에 이미지들을 업로드 후 MySQL에 이미지 경로를 삽입할 때 `[버킷 이름]`으로 변경된 경로를 실제 버킷 이름으로 변경해주어야 합니다.

```
--
-- Dumping data for table `sentence`
--

LOCK TABLES `sentence` WRITE;
/*!40000 ALTER TABLE `sentence` DISABLE KEYS */;
INSERT INTO `sentence` VALUES (1, 'https://[버킷 이름].s3.ap-northeast-2.amazonaws.com/images/sentence/000000479679.jpg');
/*!40000 ALTER TABLE `sentence` ENABLE KEYS */;
UNLOCK TABLES;
```