



# 포팅 메뉴얼

## CU;LETTER 포팅 매뉴얼

| SSAFY 6기 특화 프로젝트 서울 2반 1팀 CU;LETTER

김도현 강민수 김경협 김은송 김정연 정유환



### 목차

- 1 프로젝트 기술 스택
- 2 환경 설정 및 프로퍼티 파일
- 3 빌드 및 배포 방법
- 4 외부 서비스



### 프로젝트 기술 스택

1. 이슈 관리 : Jira
2. 형상 관리 : Gitlab
3. 커뮤니케이션 : Mattermost, Notion, Discord
4. 개발 환경
  - OS : Windows 10
  - IDE : IntelliJ 2021.3.1 / Visual Studio Code 1.63.2
  - Server : AWS EC2
    - Ubuntu 20.04 LTS
    - Jenkins 2.319.2
    - Docker 20.10.7
    - Nginx 1.18.0 (Ubuntu)
  - Database : MySQL 5.7
  - File Storage : Amazon S3
  - Frontend
    - HTML5, CSS3, Javascript(ES6)
    - React
    - Next.js
    - Material-UI
    - Node.js 16.13.1
  - Backend
    - Java 1.8.0
    - Spring Boot 2.6.4
    - JPA(Hibernate)

- Gradle 7.4.1
- AI
  - Python 3.6.15
  - Flask
  - Tensorflow 1.15
  - Tensorflow model server 1.15

## 환경 설정 및 프로퍼티 파일

### Frontend

- .env.production

```
frontend
└─ .env.production
```

```
NEXT_PUBLIC_SERVER_BASE_URL=https://www.culetter.site/api
```

### Backend

- application.properties / application.yml

```
backend
└─ src
   └─ main
       └─ resources
           ├── application.properties
           └─ application.yml
```

- DB 설정
- Email 설정
- Amazon S3 설정 (application.yml)
  - 서비스 접속을 위한 비공개 키
  - 파일이 저장되는 경로

- CORS 설정

```
backend
└─ src
   └─ main
       └─ java
           └─ com
               └─ culetter
                   └─ config
                       └─ SecurityConfig.java
```

- host에서 오는 요청에 대해 CORS 허용

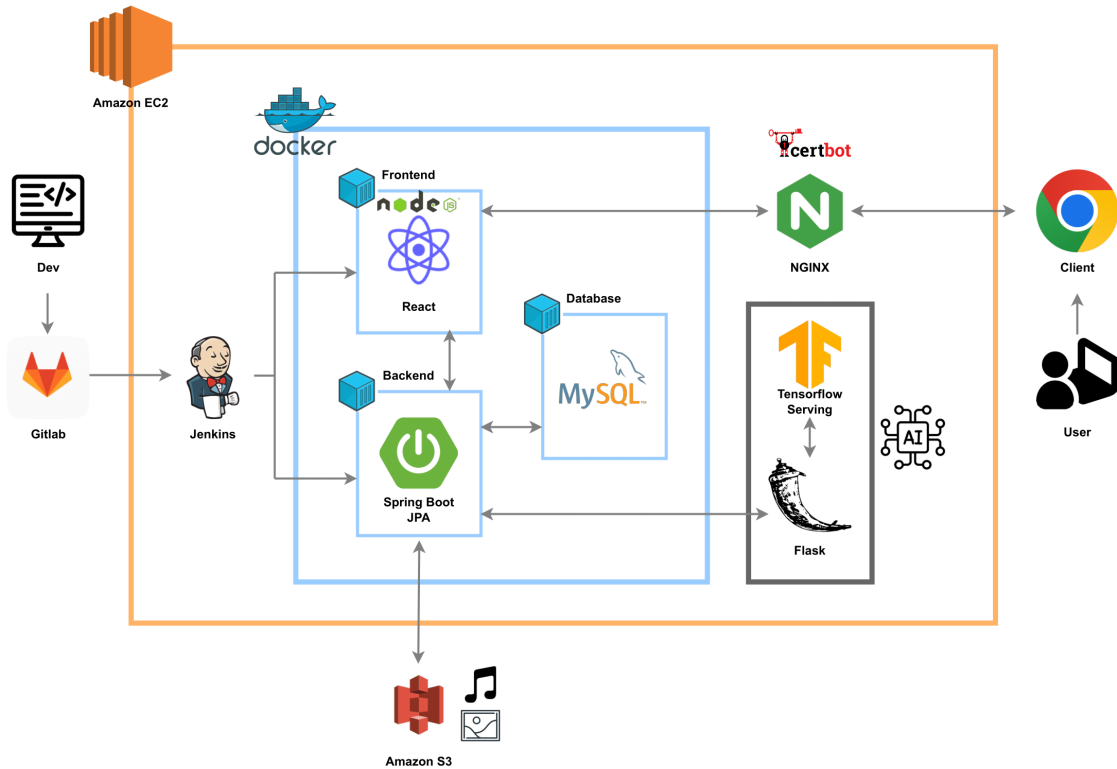
```
@Bean
CorsConfigurationSource corsConfigurationSource() {
    CorsConfiguration configuration = new CorsConfiguration();

    configuration.addAllowedOrigin("http://localhost:3000");
    configuration.addAllowedOrigin("https://j6a201.p.ssafy.io");
    configuration.addAllowedOrigin("https://www.culetter.site");
}
```

## 📢 빌드 및 배포 방법

본 빌드 및 배포 과정은 Ubuntu(Linux)를 기반으로 작성되었습니다. Docker를 활용하여 빌드된 파일을 Docker image로 만들고 container로 실행시키는 과정으로 빌드와 배포를 진행합니다.

## 0. 서버 아키텍처



## 1. Docker 설치

- **Ubuntu(Linux) and etc.**
  - Ubuntu : <https://docs.docker.com/engine/install/ubuntu/>

```
# Set up the repository
$ sudo apt-get update
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install Docker Engine
$ sudo apt-get install -y docker.io
$ sudo usermod -a -G docker $USER
```

- etc : <https://docs.docker.com/engine/install/>

## 2. Git Clone

```
$ git clone https://lab.ssfy.com/s06-webmobile1-sub2/S06P12A501.git
```

## 3. 빌드 및 배포 (Dockerize)

### • Frontend

frontend 디렉토리로 이동 후 docker image 생성, docker container 실행

```
$ cd ./frontend
$ docker build -t frontend-image .
$ docker run -d --rm --name frontend-container --net=host frontend-image
```

### • Backend

backend 디렉토리로 이동 후 docker image 생성, docker container 실행

```
$ cd ./backend
$ ./gradlew clean build
$ docker build -t backend-image .
$ docker run -d --rm --name backend-container --net=host backend-image
```

## 4. AI 모델 플러그인(Tensorflow)

### • Tensorflow

```
$ sudo pip install tensorflow==1.15
```

### • Tensorflow\_hub

```
$ sudo pip install "tensorflow>=1.15,<2.0"
$ sudo pip install --upgrade tensorflow-hub
```

### • Mecab(형태소 분석기)

```
# python3-dev
$ sudo apt install python3-dev
# python3-dev 설치 후에도 안될 경우 다음 커맨드로 libpython3.x-dev 설치
$ sudo apt-get install libpython3.6-dev

# python-mecab-ko 1.0.12:
# https://github.com/jonghwanhyeon/python-mecab-ko
# 위에서 zip 파일 받은 후, ec2에 올려놓고 그 폴더에 들어가서 다음 커맨드 실행

$ sudo python3 setup.py install
```

### • sentencepiece(구글 BPE Tokenizer)

```
$ sudo pip install sentencepiece==0.1.91
```

### • scikit-learn

```
$ sudo pip install scikit-learn==1.0.2
```

- **matplotlib**

```
$ sudo pip install matplotlib==3.5.1
```

- **numpy**

```
$ sudo pip install numpy
```

- **six**

```
# testresources 패키지 먼저 설치 후 six 패키지 설치
$ sudo apt install python3-testresources
$ sudo pip install six==1.16.0
```

## 5. AI 모델 서버 배포(Tensorflow model server, Flask)

- **Tensorflow model server**

- **Tensorflow model server 설치**

```
$ wget 'http://storage.googleapis.com/tensorflow-serving-apt/pool/tensorflow-model-server-1.15.0/t/tensorflow-model-server/te
$ dpkg -i tensorflow-model-server_1.15.0_all.deb
$ sudo pip install tensorflow-serving-api==1.15
```

- **TF 서버 실행**

```
tensorflow_model_server --rest_api_port=8501 --model_name=emotion_model --model_base_path=/home/ubuntu/ai_dev/NIA_model/emoti
```

- **Flask**

- **Flask 설치 및 실행**

```
# Flask, Flask-RESTX 설치
$ pip install Flask
$ pip install flask-restx

# app.py 파일이 존재하는 디렉토리로 이동
$ nohup python3 app.py &
```

- **서버 설정**

```
# app.py
...
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
# host='0.0.0.0' - 외부에서 접근 가능한 서버로 설정
# port=5000 - port 번호를 5000번으로 설정
```

## 외부 서비스

### 감정 분류 모델

| <https://aihub.or.kr/aidata/7978>

- 개요 : 대화체 감정 분류 모델 데이터 및 모델 제공

- 감정 분류 모델 활용
  - 감정 분류 모델 사용: ALBERT 모델을 한국어 대화체로 학습시킨 ML 모델 사용
  - 모델 고도화: 감정 대화체 관련 데이터 활용

## Amazon S3

<https://zzang9ha.tistory.com/358>

<https://artiicy.tistory.com/16>

- 개요 : 객체를 저장할 수 있는 저장소를 제공
- 버킷 생성 및 활용
  - S3에서 버킷 만들기
  - 버킷 이름 입력 / 리전 선택
  - 퍼블릭 액세스 권한 설정
  - 퍼블릭 정책 생성 및 적용
  - 액세스 키 생성
    - IAM > 사용자 > 사용자 추가 > 기존 정책 직접 연결 > AmazonS3FullAccess 체크 > 사용자 만들기 > 액세스 키 기록

## 카카오톡 메시지 API

<https://developers.kakao.com/docs/latest/ko/message/common>

- 개요 : 메시지 API는 사용자가 카카오톡 친구에게 카카오톡 메시지를 보내는 기능을 제공
- 카카오톡 메시지 활용
  - 카카오톡 공유하기 기능 : 편지 전송 완료 후 카카오톡으로 편지 공유