포팅 매뉴얼

# **목차**

1. **프로젝트 기술 스택**
2. **Frontend**
3. **Backend**
   1. **Spring**
   2. **Django**
4. **AWS EC2**
   1. **SSH 설정**
   2. **Docker**
   3. **Jenkins**
5. **Jenkins**
6. **Docker**
   1. **Frontend**
   2. **Django**
   3. **Backend**
7. **Nginx**
8. **외부 서비스 정보**
   1. **카카오 공유**
   2. **Youtube API**
   3. **STEAM API**
   4. **AWS S3**
9. **프로젝트 기술 스택**

|  |  |  |  |
| --- | --- | --- | --- |
| 형상관리 | GitLab | | . |
| 이슈관리 | Jira | | . |
| UX/UI | Figma | | . |
| 프로젝트 일정 관리 | Notion | | . |
| 빌드 및 배포 관리 | AWS EC2 | Jenkins | 2.387.1 |
| Docker | 23.0.1 |
| Docker compose | 1.25.0 |
| Nginx | 1.18.0 |
| Certbot | . |
| IDE | IntelliJ | | 2021.2.4 |
| VSCode | | 1.75.1 |
| Database | MySQL | | 8.0.32 |
| MongoDB | | 6.0.5 |
| Redis | | 7.0.10 |
| 프론트엔드 | Node.js | | 18.12.1 |
| yarn | | 1.22.19 |
| VSCode IDE EXTENSION | ESLint | 2.4.0 |
| Prettier | 0.10.4 |
| React | | 17.0.2 |
| React-router-dom | | 6.8.2 |
| recoil | | 0.7.7 |
| 백엔드 | Spring | Springboot | 2.7.7 |
| Lombok | . |
| Spring security | . |
| JPA | . |
| Gradle | 7.6 |
| Jdk | Zulu-11 |
| Swagger | 3.0.0 |
| Django | Django | 3.2.13 |
| python | 3.11 |
| pandas | 1.5.3 |

1. **Frontend**

### **패키지 설치 및 실행**

### **빌드 및 배포**

Dockerfile을 통해 진행

1. **Backend**

## **Spring**

### **환경변수 설정**

`Edit Configurations` 상단 + 를 눌러 다음의 환경변수를 추가

AWS\_ACCESS\_KEY

AWS\_SECRET\_KEY

JWT\_SECRET\_KEY

STEAM\_API\_KEY

### **빌드 및 배포**

Dockerfile을 통해 진행

## **Django**

### **가상환경 설정 및 실행**



### **빌드 및 배포**

Dockerfile을 통해 진행

1. **AWS EC2**

## **SSH**

### **방화벽 설정**



## **Docker**

### Apt가 HTTPS를 통해 repository를 이용하는 것을 허용할 수 있도록 해주는 패키지들 설치



### Docker 공식 GPG key 추가 및 repository 등록 및 설치



## **Jenkins**

### Docker에 Jenkins 설치 및 구동



1. **Jenkins**

## **플러그인 설치**

### 다음의 플러그인 설치

GitLab

Publish Over SSH

## **자동 빌드 및 배포 설정**

### **Jenkins React execute shell**



### **Jenkins Springboot execute shell**



### **Jenkins Django execute shell**



## **환경변수 등록**

### Jenkins 관리 -> 시스템 설정 -> Global properties -> Environment variables

AWS\_ACCESS\_KEY / AWS\_SECRET\_KEY / JWT\_SECRET\_KEY

STEAM\_API\_KEY / REACT\_APP\_KAKAO\_JAVASCRIPT\_KEY 등록

1. **Docker**

## **Frontend**

### **React dockerfile**



### **Nginx.conf**



### **Dockercompose.yml**



## **Django**

### **Django dockerfile**



## **Backend**

### **Springboot dockerfile**



### **Dockercompose.yml**



1. **Docker**

## **/etc/nginx nginx.conf**





## **외부 서비스 정보**

### **카카오 인증**

카카오 개발자 사이트인 <https://developers.kakao.com/> 사이트를 이용합니다

사용할 애플리케이션을 식별이 가능하도록 서비스명으로 등록합니다. 애플리케이션을 만들면 앱키가 발급됩니다. 그 중에서JavaScript 키를 복사합니다.

애플리케이션에서 Web 플랫폼을 사용하도록 등록합니다. 이 때 우리 서비스의 도메인인 [https://gamemakase.com](https://gamemakase.com/) 을 등록합니다. (필요하다면 localhost 서버도 등록합니다.)

.env 파일에 복사해둔 JavaScript키를 작성해둔 후, React에서 process.env.를 통해 앱키를 불러와 Kakao.init 함수를 실행 합니다.

### **유튜브 API**

유튜브 개발자 사이트를 이용하여 API키를 발급받습니다. (<https://console.cloud.google.com/apis/library/youtube.googleapis.com?hl=ko&project=still-resource-367503>)

사용자 인증 정보 메뉴에서 우측 상단의 API KEY를 통해 키를 복사하여 사용 할 수 있습니다.

### **STEAM API**

다음의 공식 문서를 참고하여 진행합니다. (<https://developer.valvesoftware.com/wiki/Steam_Web_API#GetPlayerSummaries_.28v0001.29>)

아래 링크를 통해 API키를 발급받습니다. (<https://steamcommunity.com/dev/apikey>)  
스팀 계정이 있어야 하며, 해당 계정에 게임이 있어야 발급이 가능합니다.

### **AWS S3**

서비스 내에서 사용되는 이미지 등의 자료를 공유하기 위해 AWS S3 클라우드  
스토리지를 적용했습니다.