

# 프로젝트 매뉴얼

## 목차

- 개요
  - 프로젝트 개요
  - 프로젝트 사용 도구
  - 개발 환경
  - 외부 서비스
- 빌드와 배포
  - 환경 변수
  - 빌드하기
  - 배포하기

## 개요

### 프로젝트 개요

요즘 대형 게임사를 제외한 거의 모든 게임이 스팀에서 구매와 플레이가 가능합니다. 스팀 판매 페이지에서는 대부분 같은 제작사의 게임이나 단순 유사장르의 게임을 추천합니다.

저희 서비스인 CrabPick은 이러한 문제점을 개선하여 이용자에게 알맞은 게임을 추천해 주는 것을 목표로 하고 있습니다.

스팀은 방대한 이용자를 기반으로 게임 별 많은 유저들의 게임 평가 데이터를 보유하고 있습니다. 이를 바탕으로 CrabPick은 게임 평가 데이터, 즉 리뷰를 수집하여 리뷰 빅데이터를 구축합니다. 그리고 그 빅데이터를 분산처리로 분석하여 게임별 유저들의 보유, 좋은 리뷰 평가 등으로 가중치를 계산하여 알맞은 게임을 제공합니다.

또한 CrabPick은 스팀 로그인 연동 기능을 제공하며, 스팀 로그인을 진행한 유저라면 그 유저의 게임 보유 정보, 게임 별 플레이 시간 등을 수집하여 맞춤 추천 서비스를 제공합니다.

### 프로젝트 사용 도구

- 이슈 관리 : JIRA
- 형상 관리 : Gitlab
- 커뮤니케이션 : Mattermost, Discord, Notion
- 디자인 : Balsamiq, Mui
- UCC : Movavi

- CI/CD : Jenkins

## 개발 환경

- VS Code : 1.74.3
- IntelliJ : 2022.3.1 (Ultimate Edition)
- JVM : azul-11
- SpringBoot: 2.7.9
- Node.js : 18.14.0 (npm 9.3.1)
- SERVER : AWS EC2 Ubuntu 20.04.5 LTS
- Distributed SERVER : AWS EC2 Ubuntu 20.04.5 LTS
- DB : Mysql 8.0.32 for Linux on x86\_64 (MySQL Community Server - GPL)
- Distributed Computing : apache hadoop 3.3.1

## 외부 서비스

SSL : certbot, letsencrypt

## 빌드와 배포

### 환경 변수

#### port

- spring-back : 8080
- node.js-back : 4000
- react-front : 80 / 443

### 빌드하기

#### dokerfile

```

FROM node:16.15.0 as build-stage
WORKDIR /var/jenkins_home/workspace/deploytest/testproject_react
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
FROM nginx:stable-alpine as production-stage
# RUN apk update && apk add certbot

COPY --from=build-stage /var/jenkins_home/workspace/deploytest/testproject_react/build /usr/share/nginx/html
COPY --from=build-stage /var/jenkins_home/workspace/deploytest/testproject_react/deploy_conf/nginx.conf /etc/nginx/conf.d/default.conf
# RUN mkdir /etc/nginx/ssl
COPY ./fullchain2.pem /etc/nginx/ssl/
COPY ./privkey2.pem /etc/nginx/ssl/
EXPOSE 80
EXPOSE 443
CMD ["nginx", "-g","daemon off;"]

```

```

FROM node:18.14.0 as build-stage
WORKDIR /var/jenkins_home/workspace/deploytest/login_server
COPY package*.json ./
RUN npm install
COPY . .

EXPOSE 4000

ENTRYPOINT npm run start

```

```

FROM node:18.14.0 as build-stage
WORKDIR /var/jenkins_home/workspace/deploytest/login_server
COPY package*.json ./
RUN npm install
COPY . .

EXPOSE 4000

ENTRYPOINT npm run start

```

## 배포하기

### NginX

```

upstream backend{
    ip_hash;
    server 172.26.2.240:8080;
}

upstream login{
    ip_hash;
    server 172.26.2.240:4000;
}

server {
    listen      80;
    listen     [::]:80;
    server_name j8e107.p.ssafy.io;

    location / {
        return 301 https://$server_name$request_uri;
    }
}

```

```

}
server {

    listen 443 ssl;
    server_name j8e107.p.ssafy.io;

    ssl_certificate /etc/nginx/ssl/fullchain2.pem;
    ssl_certificate_key /etc/nginx/ssl/privkey2.pem;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
        try_files $uri /index.html;
    }

    location /api {
        proxy_pass http://backend;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /auth {
        proxy_pass http://login;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}

```

## Jenkins

```

docker build -t backimg ./Backend
if (docker ps | grep "backimg"); then docker stop backimg; fi
if (docker ps -a | grep "backimg"); then docker rm backimg; fi
docker run -it -d -p 8080:8080 --name backimg backimg
echo "Run steam-back run"
docker ps -a

docker build -t frontimg ./frontend
if (docker ps | grep "frontimg"); then docker stop frontimg; fi
if (docker ps -a | grep "frontimg"); then docker rm frontimg; fi
docker run -it -d --rm -p 80:80 -p 443:443 --name frontimg frontimg
echo "Run steam-front run"

docker build -t loginimg ./node-passport-steam
if (docker ps | grep "loginimg"); then docker stop loginimg; fi
if (docker ps -a | grep "loginimg"); then docker rm loginimg; fi
docker run -it -d -p 4000:4000 --name loginimg loginimg
docker ps -a
echo "Run steam-login run"

```