

포팅 매뉴얼

- 포팅 매뉴얼
 - 테스트 환경
 - Requirements
 - 빌드 과정
 - 프로젝트 설정파일 수정
 - .env 파일
 - (Optional) 외부 MariaDB 서버를 사용하는 경우
 - SSL 설정
 - 빌드 및 서버 실행
 - (Optional) SSL 인증서 발급

테스트 환경

- AWS EC2
- Ubuntu 20.04 LTS

Requirements

- Docker >= 20.10.0

빌드 과정

프로젝트 설정파일 수정

.env 파일

빌드 및 운영에 관여하는 모든 시크릿 정보가 담긴 파일

- .env.template를 .env로 복사한 후 내용을 작성
 - **SERVICE_HOST**: 서비스의 도메인 주소 (ie. j7a704.p.ssafy.io)
 - **MYSQL_###**: MaridDB 관련 변수
 - **OAuth2_###**: 소셜로그인을 위한 ID, SECRET
 - Google
 - Kakao
 - Naver
 - **FRONTEND_REDIRECT_URI**: 소셜로그인 완료 후 리다이렉트 할 주소 (ie. https://j7a704.p.ssafy.io/auth/redirect)
 - **JWT_###_ENCODE_KEY**: Base64URL로 인코드 한 HMAC-SHA KEY
 - **BARCODE_KEY**: [식품의약품안전처](#)에서 발급받은 API KEY

(Optional) 외부 MariaDB 서버를 사용하는 경우

- .env 수정
 - **MYSQL_###** 를 외부 MariaDB 서버에 맞게 수정
- docker-compose.yml 수정

- **db**: 블록 전체 주석처리
- **backend**: 블록의 **depends_on**: 블록 주석처리

```
# 예시
# ...
# db:
#   image: mariadb:latest
#   expose:
#     - ${MYSQL_PORT}
#   volumes:
#     - ./db/conf.d:/etc/mysql/conf.d
#     - ./db/data:/var/lib/mysql
#     - ./db/initdb.d:/docker-entrypoint-initdb.d
#   env_file:
#     - .env
#   healthcheck:
#     test: "/usr/bin/mysql --user=root --password=${MYSQL_ROOT_PASSWORD} --
execute \"SHOW DATABASES;\""
#     interval: 2s
#     timeout: 20s
#     retries: 10
backend:
  build: ./backend
  image: easylose/backend
  expose:
    - 8080
  command: java -jar ./app.jar
  #   depends_on:
  #     db:
  #       condition: service_healthy
  restart: always
  env_file:
    - .env
# ...
```

SSL 설정

1. 기존 인증서가 존재하는 경우
 - docker-compose.yml 수정
 - **nginx**: 블록의 **volumes**: 중 /etc/letsencrypt 를 기존 인증서의 경로로 지정
 - nginx/nginx.conf.template 수정
 - **ssl_certificate** 및 **ssl_certificate_key**를 알맞게 설정
2. 기존 인증서가 없는 경우 (신규 발급)
 - nginx/nginx.conf.template 수정
 - **listen 443 ssl;** 가 있는 server 전체를 주석처리
 - 서버 실행 후 (Optional) **SSL 인증서 발급**를 진행

빌드 및 서버 실행

```
$ cd {PROJECT_ROOT}
$ docker compose up -d
```

(Optional) SSL 인증서 발급

- nginx/docker-compose.yml 수정
 - `command:`의 `-email {email 주소}` 와 `-d {도메인 주소}` 를 수정
- Certbot 컨테이너를 실행하여 인증서를 발급받음

```
$ cd nginx/certbot
$ docker compose up
```

- 이후 주석처리 했던 `server`의 주석을 해제하고 `ssl_certificate` 및 `ssl_certificate_key`를 발급받은 인증서 파일에 맞게 수정
- nginx 컨테이너를 다시 빌드 후 재시작

```
$ docker compose up -d --build nginx
```