

빌드 및 배포 문서

1. 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정값, 버전(IDE 버전 포함) 기재

종류

웹서버 - Nginx

WAS - Tomcat

버전

Backend

Java - OpenJDK 11.0.14.1

spring-boot - 2.6.3

spring 내장 tomcat - 2.6.3

gradle - 7.3.3

Frontend

node - 14.18.1

npm - 6.14.15

nginx - 1.21.6(Ubuntu)

IDE

intelliJ IDEA - 2021.3.1

Visual Studio Code - 1.64.2

2. 빌드 시 사용되는 환경변수 등의 주요 내용 상세 기재

Backend

backend/ButtonUp/src/main/resources/ `application.yml` 파일 추가

```
spring:
  datasource:
    url: jdbc:mariadb://i6b205.p.ssafy.io:5000/buttonup
    username: ssafy
    password: ssafy
    driver-class-name: org.mariadb.jdbc.Driver
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  jpa:
    generate-ddl: false
    open-in-view: false
    hibernate:
      ddl-auto: validate
    properties:
      hibernate:
        format_sql: true
  logging.level:
    org.hibernate.SQL: debug
```

AWS에서 빌드 과정

1. Git 설치 및 연동(/home/ubuntu 경로에서)

```
apt-get install git
git clone https://lab.ssafy.com/s06-webmobile2-sub2/S06P12B205.git
# 로그인
```

2. build

```
cd front 폴더로
npm install
npm run build
```

Frontend

frontend/src/config/ `index.js` 파일 추가

```
const API_BASE_URL = "http://localhost:8080";

export { API_BASE_URL };
```

AWS에서 빌드 과정

1. Java 설치

```
apt-get install openjdk-11-jdk
```

2. 빌드()

```
sudo chmod 777 ./gradlew # 폴더나 파일에 권한부여(777 - 모든 권한)  
sudo ./gradlew build      # build
```

3. 배포 시 특이사항 기재

Frontend

방법 - AWS에서 nodejs로 빌드 후, Docker에서 NGINX로 실행

1. image 다운

```
docker pull nginx
```

2. nginx의 conf파일 필요! - `/home/ubuntu/nginx.conf` 생성

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /usr/share/nginx/html; # 실행시킬 html이 있는 위치  
  
    index index.html index.htm;  
  
    server_name _;  
    location / {  
        root /usr/share/nginx/html # html 폴더 위치  
        try_files $uri $uri/ /index.html; # vue의 히스토리 모드 해결  
    }  
}
```

3. container 실행 - docker 내의 html 폴더와 default.conf 파일에 volumn 설정

```
docker run --name nginx -d -p 80:80
-v /home/ubuntu/nginx.conf:/etc/nginx/conf.d/default.conf
-v /home/ubuntu/S06P12B205/frontend/dist:/usr/share/nginx/html nginx
```

Backend

방법 - AWS에서 Gradle로 빌드 후, Docker에서 Spring 내장 tomcat로 실행

1. Backend 프로젝트 내에 Dockerfile 파일 생성(src 폴더나 gradlew 있는 곳)

Dockerfile

```
FROM openjdk:11-jdk
EXPOSE 8080
ARG JAR_FILE=build/libs/ButtonUp-0.0.1-SNAPSHOT.jar # jar파일 경로
COPY ${JAR_FILE} app.jar # docker 내의 app.jar로 복사
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

2. Docker image 생성(빌드)

```
# Dockerfile 있는 곳에서
docker build . -t back:0.1 # -t : 태그
```

3. Docker Container 실행

```
docker run --name back -d -p 8080:8080 back:0.1
```

4. DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

DB명

buttonup

관리자 계정

사용자 : buttonup

암호 : danchu

해당 db에 대한 권한만 가진 사용자 계정

사용자 : ssafy

암호 : ssafy

프로퍼티

--character-set-server=utf8mb4

--collation-server=utf8mb4_unicode_ci