

Gitlab 소스 클론 이후 빌드 및 배포 매뉴얼

1. 사용한 JVM, 웹 서버, WAS 제품 등의 종류와 설정값, 버전 (IDE버전 포함) 기재

1-1. 종류

- 웹 서버 : Nginx

2. 빌드 시 사용되는 환경 변수 등의 내용 상세 기재

- 프론트 프로젝트의 nextjs-container.sh 컨테이너 실행 시 환경변수 주입

```
sudo docker rmi -f jweon/alfs_fe || true
sudo docker stop nextjs-container || true
sudo docker rm -f nextjs-container || true
sudo docker run -d -p 3000:3000 --name nextjs-container \
    -e NEXTAUTH_URL=[서비스 url] \
    -e NEXTAUTH_SECRET=[로그인 할 때 쓰는 next.js 인증키] \
    jweon/alfs_fe
```

- 백엔드 프로젝트의 spring-container.sh 컨테이너 실행 시 환경변수 주입

```
#!/bin/bash

sudo docker rmi -f jweon/alfs_be || true
sudo docker stop spring-container || true
sudo docker rm -f spring-container || true

sudo docker run -d -p 8080:8080 --name spring-container\
    -e cloud.aws.credentials.accessKey=[AWS S3의 accessKey]\
    -e cloud.aws.credentials.secretKey=[AWS S3의 secretKey]\
    -e cloud.aws.s3.bucket=[AWS S3에서 사용할 bucket 이름]\
    -e cloud.aws.region.static=[AWS 계정의 region 버전명]\
    -e cloud.aws.stack.auto=false\
    -e spring.datasource.hikari.username=[mysql DB의 user name]\
    -e spring.datasource.hikari.password=[mysql DB의 password]\
    -e X_OCR_SECRET=[Naver Clovar OCR의 Secret Key] \
    -e OCR_INVOKE_URL=[Naver Clovar OCR의 Invoke URL] \
    -e spring.redis.pw=[redis DB의 password] \
    jweon/alfs_be
```

2. DB 접속 정보 등 프로젝트에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

- 백엔드 프로젝트의 application.properties

```

#it will be set build date by gradle. if this value is @build.date@, front-end
is development mode
build.date=@build.date@
server.port=8080
server.address=0.0.0.0
server.servlet.contextPath=/api
# Charset of HTTP requests and responses. Added to the "Content-Type" header if
not set explicitly.
server.servlet.encoding.charset=UTF-8
# Enable http encoding support.
server.servlet.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and responses.
server.servlet.encoding.force=true
# PathVariable Charset
server.tomcat.uri-encoding=UTF-8

# database
spring.datasource.url=jdbc:mysql://k9c204.p.ssafy.io:3306/alfs?serverTime-
zone=Asia/Seoul&characterEncoding=UTF-8
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# jpa
spring.jpa.hibernate.ddl-auto=update

# aws management
spring.profiles.include=ignore

# S3
spring.servlet.multipart.max-file-size= 10MB
spring.servlet.multipart.max-request-size= 10MB

#logging
logging.level.com.world.alfs=debug

# file
#upload.directory=c:\\alfs\\img
upload.directory=/var/lib/alfs/img

# spring batch
spring.batch.job.enabled=false
spring.jpa.properties.hibernate.enable_lazy_load_no_trans=true

# redis
spring.redis.host=k9c204.p.ssafy.io
spring.redis.port=6379

```

- 백엔드 프로젝트의 application-ignore.properties 설정

```
# S3
cloud.aws.credentials.accessKey=[AWS S3의 accessKey]
cloud.aws.credentials.secretKey=[AWS S3의 secretKey]
cloud.aws.s3.bucket=my-alfs-aws-bucket
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false

# Databse
spring.datasource.hikari.username=[mysql DB 의 user name]
spring.datasource.hikari.password=[mysql DB 의 password]

# Redis
spring.redis.pw=[redis DB 의 password]

# Naver Clova OCR
X_OCR_SECRET=[Naver Clovar OCR 의 Secret Key]
OCR_INVOKE_URL=[Naver Clovar OCR 의 Invoke URL]
```

- 백엔드 프로젝트의 build.gradle의 설정

```

plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.17'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
}

group = 'com.world'
version = '0.0.1-SNAPSHOT'

java {
    sourceCompatibility = '11'
}

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    runtimeOnly 'com.mysql:mysql-connector-j'
    runtimeOnly 'org.mariadb.jdbc:mariadb-java-client:2.7.4'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'
    testImplementation 'org.junit.jupiter:junit-jupiter-api:5.6.0'
    testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine:5.6.0'
    implementation group: 'org.json', name: 'json', version: '20160810'
    implementation 'org.springframework.boot:spring-boot-starter-batch'
    testImplementation 'org.springframework.batch:spring-batch-test'
    testCompileOnly 'junit:junit:4.12'
    implementation 'com.h2database:h2'

    // JWT
    implementation 'org.springframework.boot:spring-boot-starter-security'
    // testImplementation 'org.springframework.security:spring-security-test'
    // implementation group: 'io.jsonwebtoken', name: 'jjwt-api', version: '0.11.5'
    // runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-impl', version: '0.11.5'
    // runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-jackson', version: '0.11.5'

    //Redis
    implementation 'org.springframework.boot:spring-boot-starter-data-re-
dis'

```