

포팅 매뉴얼

소유자	석영
태그	

Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서

1)

- **JVM**

- OpenJDK 17

- **WAS**

- Spring Boot 내장 Tomcat
- Spring Boot 버전: 3.3.4
- 내장 Tomcat 버전: 10.1.30
- 주요 설정
 - 포트: 8080
 - Context Path: /
 - build.gradle

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.3.4'  
    id 'io.spring.dependency-management' version '1.1.6'  
}  
  
group = 'com.ssafy'  
version = '0.0.1-SNAPSHOT'  
  
java {  
    toolchain {  
        languageVersion = JavaLanguageVersion.of(17)  
    }  
}  
  
configurations {  
    compileOnly {  
        extendsFrom annotationProcessor  
    }  
}  
  
repositories {
```

```

        mavenCentral()
    }

    dependencies {
        implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
        implementation 'org.springframework.boot:spring-boot-starter-validation'
        implementation 'org.springframework.boot:spring-boot-starter-web'
        implementation 'org.flywaydb:flyway-core'
        implementation 'org.flywaydb:flyway-mysql'
        implementation 'org.springframework.boot:spring-boot-starter-security'
        implementation 'io.jsonwebtoken:jjwt:0.9.1'
        implementation 'javax.xml.bind:jaxb-api:2.3.0'
        implementation 'org.springframework.boot:spring-boot-starter-webflux'
        implementation 'com.fasterxml.jackson.core:jackson-databind'
        testImplementation 'org.springframework.security:spring-security-test'
        compileOnly 'org.projectlombok:lombok'
        runtimeOnly 'org.mariadb.jdbc:mariadb-java-client'
        annotationProcessor 'org.projectlombok:lombok'
        testImplementation 'org.springframework.boot:spring-boot-starter-test'
        testRuntimeOnly 'org.junit.platform:junit-platform-launcher'
    }

    tasks.named('test') {
        useJUnitPlatform()
    }
}

```

2) 빌드 시 사용되는 환경 변수 등의 내용 상세 기재

Spring Boot

```

LOCAL_NAME=root;
LOCAL_PASSWORD=ssafy;
DEV_NAME=admin;
DEV_PASSWORD=ssafyssafy;
REFRESH_KEY=refresh-local;
SECRET_KEY=secret-local;
MEAL_KEY=98ad2adb065f4197a21606b1775b6305

```

fastAPI

```

API_URL:"https://v4u297oqwl.apigw.ntruss.com/custom/v1/35355/f7f1daa58efa2f4c
SECRET_KEY:"VmJuRHRiZlhlWF1TU2RMa25MY01NdVRmaW1yUlRUR3M="
CRAWLING_URL: "https://m.onestore.co.kr/mobilepoc/search/searchMain.omp"

```

3) 배포 시 특이사항 기재

ECR 사용 정보

- 리포지토리 이름: `padlock`
- 리포지토리 URI: `396608802848.dkr.ecr.ap-northeast-2.amazonaws.com/padlock`
- Docker 이미지 배포 흐름:
 1. Jenkins 서버에서 Docker 이미지를 빌드
 2. AWS ECR에 이미지 push
 3. EC2 서버(Spring Boot 애플리케이션 실행 환경)에서 이미지를 pull하여 컨테이너로 실행

IAM 역할 및 EC2 설정

- Jenkins용 EC2 인스턴스:
 - Jenkins 서버를 호스팅하는 EC2에 IAM 역할이 부여되어 있음.
 - IAM 역할 이름: `jenkins`
 - IAM 정책: `AmazonEC2ContainerRegistryFullAccess`
- IAM 정책 설명:
 - 이 정책을 통해 Jenkins EC2는 다음 작업을 수행할 수 있음:
 - ECR에서 Docker 이미지를 푸시 및 가져오기
 - 리포지토리 목록 확인 및 관리
 - IAM 역할이 부여된 EC2 인스턴스에서는 `aws ecr get-login-password` 를 통해 인증을 자동으로 수행할 수 있음.
 - IAM 역할이 부여된 EC2 인스턴스에서는 별도의 AWS 자격 증명(access key/secret key)을 제공하지 않아도 `aws ecr get-login-password` 명령어를 통해 인증을 수행할 수 있음.

```
aws ecr get-login-password --region ap-northeast-2 | docker login --
```

4) DB 접속 정보 등 프로젝트에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

프로퍼티 파일 목록:

- application.yml (공통 설정)
- application-local.yml (로컬 환경 설정)
- application-dev.yml (개발 및 운영 환경 설정)

```
# application.yml

spring:
  application:
    name: padlock
  jpa:
    properties:
      hibernate:
        show_sql: true
        format_sql: true
      jdbc:
        time_zone: Asia/Seoul
  sql:
    init:
      mode: always

  servlet:
    multipart:
      max-file-size: 100MB
      max-request-size: 100MB

jwt:
  secret-key: ${SECRET_KEY}
  refresh-key: ${REFRESH_KEY}

app:
  fastapi:
    url: http://222.107.238.44:8001
```

```
# application-local.yml

spring:
  config:
    activate:
      on-profile: local
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://localhost:3306/padlock?serverTimezone=Asia/Seoul&char
    username: ${LOCAL_NAME}
    password: ${LOCAL_PASSWORD}
  jpa:
    hibernate:
      ddl-auto: validate
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect
  flyway:
```

```
enabled: true
baseline-on-migrate: true
baseline-version: 1
fail-on-missing-locations: true
```

```
# application-dev.yml

spring:
  config:
    activate:
      on-profile: dev
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://padlock.chs6oo82883l.ap-northeast-2.rds.amazonaws.com
    username: ${DEV_NAME}
    password: ${DEV_PASSWORD}
  jpa:
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect
  flyway:
    enabled: false
```

프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서

1) 나이스 교육정보 개방 포털 급식 식단 정보 API

- 사용 목적: 학교 별 급식 데이터 조회시 사용

2) CLOVA API

- 사용 목적: 글씨 OCR 변환 시 사용

시연 시나리오

▼ 핸드폰

로그인 된 상태로 대기

1. 학생 앱 비콘 출석 → 출석상태 바뀌는거 보여주기 (미출석 → 지각)
2. 부모 학생 출석 변경 확인
3. 부모 학생 gps확인
4. 부모 상담신청 보여주기
5. 학생 건의하기 보여주기

▼ 태블릿(앱제어, OCR)

0. 최신화 해놓기
1. 교사 앱목록 추가 보여주기
2. 학생 앱 바뀐거 확인 후 앱 동작해보기(안되는거, 되는거 둘 다 보여주기)
3. 학생 ocr사진찍기
4. 필기변환하기
5. 저장된 필기 보여주기
6. 교사 시간표 설정하는거 보여주기