

산출물

사용한 JVM 웹 서버 WAS 제품과 설정 값, 버전(IDE버전 포함)

JVM	Tomcat	Spring Tool Suite
<ul style="list-style-type: none">zulu8.68.0.19-ca-jdk8.0.362-win_x64	<ul style="list-style-type: none">spring-boot-starter-tomcat-2.7.9tomcat 9.0.71	<ul style="list-style-type: none">sts-4.17.1.RELEASE

빌드 시 사용되는 환경 변수 등의 주요 내용 상세

BackEnd	FrontEnd
<ul style="list-style-type: none">Java Open JDK 1.8.0_362Spring Boot 2.7.9Gradle 8.0.2MySQL 8.0.31Hadoop-3.3.4Hive-3.1.2	<ul style="list-style-type: none">Npm 9.2.0NodeJs 16.19.0
	OS Server
	<ul style="list-style-type: none">Ubuntu 20.04

배포 시 특이사항

- Blue/Green 무중단 배포 전략 – Jenkins, Docker, Nginx, Spring Boot, React

```
Spring Boot profile 기본 설정
- Blue profile: 9001 port
- Green profile: 9002 port
```

- GitLab에 새로운 버전이 배포되면 Webhook을 통해 Jenkins에 트리거를 보내고, Jenkins는 최신 버전의 Jar 파일을 빌드한다.
 - 프론트 배포 브랜치 – develop/FE
 - 백엔드 배포 브랜치 – develop/BE
- Jenkins에서 빌드한 Jar 파일을 EC2 서버로 전송한 후 배포 쉘 스크립트를 실행한다.
- 도커 이미지를 빌드한 후 현재 가동 중인 profile을 확인해서 blue가 가동 중이면 green에, green이 가동 중이면 blue에 도커 컨테이너를 띄운다.
- 배포된 애플리케이션이 구동되었는지 10초 주기로 Health Check를 한다.
- Health Check가 통과되면 Nginx의 reverse proxy 방향을 배포된 포트로 스위칭한다.
- 이전에 배포되어 있던 구 버전의 도커 컨테이너를 kill 한다.

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

- DB 접속 정보
 - Hostname: j8a606.p.ssafy.io
 - Port: 3306
 - Username: kangwedance
 - Password: aLo8aFitTJ7Yole
- Jenkins Admin User
 - 계정명: ssafya606
 - 암호: silSACxJZYMFjdK

application.yml

```
server:
  servlet:
    encoding:
      force: 'true'
      charset: UTF-8
      enabled: 'true'
    context-path: /dev
  logging:
    level:
      root: info

spring:
  profiles:
    active: local
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    username: kangwedance
    url: jdbc:mysql://j8a606.p.ssafy.io:3306/kangwedance?allowMultiQueries=true&useUnicode=yes&characterEncoding=UTF-8&serverTimezone=
    password: aLo8aFitTJ7YoIe

mybatis:
  mapper-locations: classpath:mappers/*.xml
  type-aliases-package: com.ssafy.kang.*.model

cloud:
  aws:
    credentials:
      secret-key: IUbrwTIFA3bLB/f1KA1s2sNE2brE14M6nASf1Ltw
      access-key: AKIARYAZWIARPVAYS4F7
    stack:
      auto: 'false'
    s3:
      bucket: kangwedance
    region:
      static: ap-northeast-2
      auto: 'false'

---

spring:
  config:
    activate:
      on-profile: local
server:
  port: 8888

---

spring:
  config:
    activate:
      on-profile: set1
server:
  port: 9001

---

spring:
  config:
    activate:
      on-profile: set2
server:
  port: 9002
---

spring:
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher

hive:
  connectionURL: jdbc:hive2://52.55.191.26:10006/default
  username: APP
  password: mine
```

build.gradle

```

plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.9'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
    id 'project-report'
}

group = 'com.ssafy.kang'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '1.8'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

test {
    useJUnitPlatform()
}

dependencies {
    implementation("org.springframework.boot:spring-boot-starter-web")
    implementation 'org.springframework.boot:spring-boot-starter-actuator'
    implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.3.0'
    implementation group: 'org.springframework.cloud', name: 'spring-cloud-starter-aws', version: '2.0.1.RELEASE'
    implementation group: 'javax.xml.bind', name: 'jaxb-api', version: '2.3.1'
    implementation group: 'org.springframework.cloud', name: 'spring-cloud-aws-context', version: '1.2.1.RELEASE'
    implementation group: 'org.springframework.cloud', name: 'spring-cloud-aws-autoconfigure', version: '1.2.1.RELEASE'
    implementation group: 'com.google.code.gson', name: 'gson', version: '2.7'
    implementation group: 'io.springfox', name: 'springfox-swagger2', version: '2.9.2'
    implementation group: 'io.springfox', name: 'springfox-swagger-ui', version: '2.9.2'
    implementation group: 'io.jsonwebtoken', name: 'jjwt-api', version: '0.10.5'
    implementation group: 'org.apache.commons', name: 'commons-dbcp2', version: '2.9.0'
    implementation ('org.apache.hive:hive-jdbc:3.1.2') {
        [new Tuple('org.eclipse.jetty.orbit', 'javax.servlet')
        , new Tuple('org.eclipse.jetty.aggregate', 'jetty-all')
        , new Tuple('org.json', 'json')
        , new Tuple('org.slf4j', 'slf4j-log4j12')
        , new Tuple('org.apache.logging.log4j', 'log4j-slf4j-impl')
        , new Tuple('org.eclipse.jetty', 'jetty-runner')]
        ].each {
            exclude group: "${it.get(0)}", module: "${it.get(1)}"
        }
    }

    runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-impl', version: '0.10.5'
    runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-jackson', version: '0.10.5'
    compileOnly 'org.projectlombok:lombok'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    runtimeOnly 'com.mysql:mysql-connector-j'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}

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

```

package.json

```

{
  "name": "dancingroo",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@nivo/bullet": "^0.80.0",
    "@nivo/bump": "^0.80.0",
    "@nivo/core": "^0.80.0",
    "@nivo/line": "^0.80.0",
    "@nivo/radar": "^0.80.0",
    "@reduxjs/toolkit": "^1.9.3",
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^1.3.4",
    "dotenv": "^16.0.3",
    "html2canvas": "^1.4.1",
    "konva": "^8.4.2",

```

```

    "moment": "^2.29.4",
    "react": "^18.2.0",
    "react-calendar": "^4.0.0",
    "react-cookie": "^4.1.1",
    "react-dom": "^18.2.0",
    "react-hover-video-player": "^10.0.0",
    "react-icons": "^4.8.0",
    "react-konva": "^18.2.5",
    "react-redux": "^8.0.5",
    "react-router-dom": "^6.8.2",
    "react-scripts": "5.0.1",
    "react-webcam": "^7.0.1",
    "styled-components": "^5.3.8",
    "sweetalert2": "^11.7.3",
    "use-image": "^1.1.0",
    "use-sound": "^4.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}

```

외부 서비스 정보

```

# 카카오
REACT_APP_API_KEY_KAKAO = eb3617506bc76ea56b625f99f56e42dd
REACT_APP_REDIRECT_URI_SITE = https://kangwedance.site/users/oauth2-
REACT_APP_LOGOUT_REDIRECT_URI_SITE = https://kangwedance.site/

# 네이버
REACT_APP_API_KEY_NAVER = GBp5LyAN_0Wl9eQyjVmb
REACT_APP_STATE_TOKEN = oauth-naver-chanchanchan

```