빌드 및 배포 문서

태그

목차

1. 개발 환경

- node.js : 18.16.1
- vscode : 1.82.3
- react : 18.2
- jdk : 17.0.8
- springboot : 3.1.3
- intellij : 2023.1.4
- docker : 24.0.6
- mysql : 8.0.33
- nginx : 1.18.0 (Ubuntu)
- jenkins : 2.422

2. 설정파일

springboot

- build.gradle

```
plugins {
  id 'java'
  id 'org.springframework.boot' version '3.1.5'
  id 'io.spring.dependency-management' version '1.1.3'
  id 'org.asciidoctor.jvm.convert' version '3.3.2'
}

group = 'com.boyworld'
```

```
version = '0.0.1-SNAPSHOT'

java {
  sourceCompatibility = '17'
}

configurations {
  compileOnly {
    extendsFrom annotationProcessor
  }
  asciidoctorExt
}

repositories {
  mavenCentral()
}

ext {
  set('snippetsDir', file("build/generated-snippets"))
}

dependencies {
  // spring boot
  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 group: 'org.springframework.boot', name: 'spring-boot-starter-actuator', version: '3.1.5'
  testImplementation 'org.springframework.boot:spring-boot-starter-test'
  developmentOnly 'org.springframework.boot:spring-boot-devtools'

  // redis
  implementation 'org.springframework.boot:spring-boot-starter-data-redis'

  // security
  implementation 'org.springframework.boot:spring-boot-starter-security'
  testImplementation 'org.springframework.security:spring-security-test'

  // jwt
  implementation 'io.jsonwebtoken:jjwt-api:0.11.5'
  implementation 'io.jsonwebtoken:jjwt-impl:0.11.5'
  implementation 'io.jsonwebtoken:jjwt-jackson:0.11.5'

  // lombok
  compileOnly 'org.projectlombok:lombok'
  annotationProcessor 'org.projectlombok:lombok'
  testCompileOnly 'org.projectlombok:lombok'
  testAnnotationProcessor 'org.projectlombok:lombok'

  // database
  runtimeOnly 'com.mysql:mysql-connector-j'
  runtimeOnly 'com.h2database:h2:1.4.200'

  // Querydsl 추가
  implementation 'com.querydsl:querydsl-jpa:5.0.0:jakarta'
  annotationProcessor "com.querydsl:querydsl-apt:${dependencyManagement.importedProperties['querydsl.version']}:jakarta"
  annotationProcessor "jakarta.annotation:jakarta.annotation-api"
  annotationProcessor "jakarta.persistence:jakarta.persistence-api"

  // rest docs
  asciidoctorExt 'org.springframework.restdocs:spring-restdocs-asciidoctor'
  testImplementation 'org.springframework.restdocs:spring-restdocs-mockmvc'

  // Amazon S3
  implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'

  // Google FCM
  implementation 'com.google.firebase:firebase-admin:9.2.0'

  // iamport payment
  implementation 'com.github.iamport:iamport-rest-client-java:0.2.21'

  //email
  implementation 'org.springframework.boot:spring-boot-starter-mail'
  implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
}

// iamport payment
allprojects {
  repositories {
    maven { url 'https://jitpack.io' }
  }
}

tasks.named('test') {
```

```
  outputs.dir snippetsDir
  useJUnitPlatform()
}

asciidoctor {
  inputs.dir snippetsDir
  configurations 'asciidoctorExt'

  sources {
    include("**/index.adoc")
  }
  // 특정 .adoc에 다른 adoc 파일을 가져와서(include) 사용하고 싶을 경우
  // 경로를 baseDir로 맞춰주는 설정
  baseDirFollowsSourceFile()
  dependsOn test
}

asciidoctor.doFirst {
  delete file('src/main/resources/static/docs')
}

// asccidoctor 작업 이후 생성된 HTML 파일을 static/docs 로 copy
tasks.register('copyDocument', Copy) {
  dependsOn asciidoctor
  from file("build/docs/asciidoc")
  into file("src/main/resources/static/docs")
}

build {
  dependsOn copyDocument
}

bootJar {
  dependsOn asciidoctor
  from("${asciidoctor.outputDir}/html5") {
    into 'static/docs'
  }
}

// Querydsl 설정부
def generated = 'src/main/generated'

// querydsl QClass 파일 생성 위치를 지정
tasks.withType(JavaCompile) {
  options.getGeneratedSourceOutputDirectory().set(file(generated))
}

// java source set 에 querydsl QClass 위치 추가
sourceSets {
  main.java.srcDirs += [ generated ]
}

// gradle clean 시에 QClass 디렉토리 삭제
clean {
  delete file(generated)
}
```

application.yml

```
spring:
  profiles:
    include: dev, env  #dev
  #    include: server, env  #server

  servlet:
    multipart:
      max-file-size: 100MB
      max-request-size: 100MB
  main:
    allow-bean-definition-overriding: true
logging:
  file:
    path: logs
jwt:
  secret: ${JWT_SECRET}
server:
  servlet:
    encoding:
      charset: UTF-8
```

```
      force: true
    context-path: /api
```

application-dev.yml, application-server.yml

```
spring:
  # database setting
  datasource:
    url: ${MYSQL_URL}
    username: ${MYSQL_USER}
    password: ${MYSQL_PASSWORD}
    driver-class-name: com.mysql.cj.jdbc.Driver
  # jpa setting
  jpa:
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        default_batch_fetch_size: 100
        format_sql: true
        show_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
        # reserved words error config (add backticks)
        auto_quote_keyword: false
        globally_quoted_identifiers: false
    open-in-view: true
    defer-datasource-initialization: false
  data:
    redis:
      host: ${REDIS_HOST}
      port: ${REDIS_PORT}
      pw: ${REDIS_PASSWORD}
  mail:
    host: smtp.gmail.com
    port: 587
    username: ${EMAIL_ID}
    password: ${EMAIL_PASSWORD}
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true
            required: true
            timeout: 5000
      auth-code-expiration-millis: 1800000  # 30 * 60 * 1000 == 30분
  sql:
    init:
      mode: always
      encoding: UTF-8
# log setting
logging:
  level:
    org.hibernate.sql: debug
    com.boyworld.carrot: debug

server:
  port: 8001

file:
  dir: /var/images

cloud:
  aws:
    s3:
      bucket: carrottruck
    credentials:
      access-key: ${ACCESS_KEY}
      secret-key: ${SECRET_KEY}
    region:
      static: ap-northeast-2
      auto: false
    stack:
      auto: false

iamport:
  key: ${PORT_ONE_KEY}
  secret: ${PORT_ONE_SECRET}
```

application-env.yml

```
##db
MYSQL_USER: <secret>
MYSQL_PASSWORD: <secret>
MYSQL_URL: jdbc:mysql://<도메인 url>/donworry?serverTimezone=UTC&useUniCode=yes&characterEncoding=UTF-8

##db
MYSQL_USER: <secret>
MYSQL_PASSWORD: <secret>
MYSQL_URL: jdbc:mysql://localhost:3306/carrottruck?serverTimezone=UTC&useUniCode=yes&characterEncoding=UTF-8

##jwt
JWT_SECRET: <secret>

##email
EMAIL_ID: carrot.truck.noreply@gmail.com
EMAIL_PASSWORD: <secret>

##naver-maps
NAVER_CLIENT_ID: <secret>
NAVER_CLIENT_SECRET: <secret>
NAVER_GEOCODE_URL: https://naveropenapi.apigw.ntruss.com/map-geocode/v2/geocode
NAVER_REVERSE_GEOCODE_URL: https://naveropenapi.apigw.ntruss.com/map-reversegeocode/v2/gc

##store-info
DATA_GO_SERVICE_KEY: <secret>
STORE_DONG_URL: <secret>
STORE_RADIUS_URL: <secret>
DONG_CODE_URL: <secret>

## redis
REDIS_HOST: <secret>
REDIS_PORT: <secret>
REDIS_PASSWORD: <secret>

## Amazon S3
ACCESS_KEY: <secret>
SECRET_KEY: <secret>

## Port One API
PORT_ONE_KEY: <secret>
PORT_ONE_SECRET: <secret>

---

##dev
spring:
  config:
    activate:
      on-profile: dev

##db
MYSQL_USER: <secret>
MYSQL_PASSWORD: <secret>
MYSQL_URL: jdbc:mysql://localhost:3306/carrottruck?serverTimezone=UTC&useUniCode=yes&characterEncoding=UTF-8

---

##server
spring:
  config:
    activate:
      on-profile: server

##db
MYSQL_USER: <secret>
MYSQL_PASSWORD: <secret>
MYSQL_URL: jdbc:mysql://<secret>/carrottruck?serverTimezone=UTC&useUniCode=yes&characterEncoding=UTF-8
```

front .env.development

```
REACT_APP_HOST=localhost
REACT_APP_API_URL=http://localhost:8001/api
REACT_APP_CLIENT_ID=<secret>
REACT_APP_CLIENT_SECRET=<secret>
REACT_APP_PAYMENT_CODE=<secret>
```

```
# npm run start
```

front .env.production

```
REACT_APP_HOST=carrottruck.com
REACT_APP_API_URL=<secret>
REACT_APP_CLIENT_ID=<secret>
REACT_APP_CLIENT_SECRET=<secret>
REACT_APP_PAYMENT_CODE=<secret>

#npm run build
```

3. 배포관련

3-1 사용자 및 Spring

nginx default conf파일

```
grade to it.


3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***
Last login: Fri Nov 17 00:02:23 2023 from 118.235.16.211
ubuntu@ip-172-26-2-210:~$ vi /etc/nginx/sites-available/default
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #try_files $uri $uri/ =404;
                #proxy_pass $service_url;
                proxy_pass http://172.22.1.3:3000;

                }
#       location /vendor {
#               proxy_pass http://172.24.1.3:3001;
#       }

        location /api {
                proxy_pass http://172.27.1.3:8001;
        }

        location ~ ^/(actuator|docs) {
                proxy_pass http://172.27.1.3:8001;
        }
        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
        #       include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
        #       fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}


    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/k9c211.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/k9c211.p.ssafy.io/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    if ($host = k9c211.p.ssafy.io) {
```

```
        return 308 https://$host$request_uri;
    } # managed by Certbot


        listen 80 ;
        listen [::]:80 ;
    server_name k9c211.p.ssafy.io;
    return 404; # managed by Certbot


}
```

react.dockerfile

```
FROM node:18

RUN npm install -g serve

RUN mkdir ./build

COPY ./build ./build

ENTRYPOINT ["serve", "-s", "build"]
```

react.docker-compose

```
version: "3.3"

services:
  react:
    build:
      context: .
      dockerfile: react.dockerfile
    container_name: "react"

    expose:
      - "3000"

    networks:
      default_bridge:
        ipv4_address: 172.22.1.3

networks:
  default_bridge:
    ipam:
      driver: default
      config:
        - subnet: 172.22.1.0/24
```

springboot.dockerfile

```
FROM openjdk:17-jdk-slim

ARG JAR_FILE=build/*.jar

COPY ${JAR_FILE} app.jar

EXPOSE 8001

ENTRYPOINT [ "java", "-jar", "/app.jar" ]
```

springboot.docker-compose

```
version: "3.3"

services:

  springboot:
    container_name: "springboot"

    build:
      context: "."
```

```
      dockerfile: "springboot.dockerfile"

    volumes:
      - "./images:/var/images"
      - "./logs:/var/log"

    expose:
      - "8001"

    networks:
      default_bridge:
        ipv4_address: 172.27.1.3


networks:
  default_bridge:
    ipam:
      driver: default
      config:
        - subnet: 172.27.1.0/24
```

mysql.docker-compose

```
version: "3.3"

services:

  mysql:
    container_name: "mysql"

    image: "mysql:8.0.33"

    restart: "always"

    ports:
      - "3306:3306"

    environment:
            TZ: Asia/Seoul

    volumes:
      - "/home/ubuntu/carrottruck/default/mysql_home:/var/lib/mysql"

    command:
      - "--character-set-server=utf8mb4"
      - "--collation-server=utf8mb4_unicode_ci"
      - "--skip-character-set-client-handshake"
      - "--sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION"
    env_file:
      - "./env/mysql.env"
```

redis.docker-compose

```
version: "3.3"

services:
  redis:
    container_name: "redis"
    image: "redis"
    ports:
      - "6379:6379"
```

jenkins CI/CD Pipeline Script

```
node {
    stage('Initial'){
        sh "pwd"
        sh "rm -r ./**"
    }
    stage('Pull') {
        cleanWs()
        checkout scmGit(branches: [[name: '*/develop']], extensions: [], userRemoteConfigs: [[credentialsId: 'gitlab_carrottruck', url
    }

    stage('Change setting file'){
```

```
        sh "pwd"
        sh "cp /var/jenkins_home/application.yml /var/jenkins_home/workspace/CarrotTruck/Backend/src/main/resources/"
        sh "cp /var/jenkins_home/application-env.yml /var/jenkins_home/workspace/CarrotTruck/Backend/src/main/resources/"
        sh "cp /var/jenkins_home/application-server.yml /var/jenkins_home/workspace/CarrotTruck/Backend/src/main/resources/"
        sh "cp /var/jenkins_home/carrottruck-2f9cf-firebase-adminsdk-bi488-258c9179a3.json /var/jenkins_home/workspace/CarrotTruck/Bac
        sh "cp /var/jenkins_home/.env.production /var/jenkins_home/workspace/CarrotTruck/frontend/client"
        // sh "cp /var/jenkins_home/.env.production /var/jenkins_home/workspace/CarrotTruck/frontend/vendor"
    }

    stage('Build'){
        sh "pwd"
        dir('frontend/client') {
            nodejs('NodeJs 18.16.1') {
                sh "pwd"
                sh "npm install"
                sh "CI= npm run build"
            }
        }

        dir('Backend') {
            // some block
            sh "pwd"
            sh "chmod +x gradlew"
            withGradle {
                // some block
                sh './gradlew clean'
                sh './gradlew bootJar'
            }
        }
    }
    stage('ssh client') {
        dir('frontend/client') {
        sshPublisher(publishers: [sshPublisherDesc(configName: 'c211ec2',
        transfers: [sshTransfer(cleanRemote: false, excludes: '', execCommand: '''
        cd /home/ubuntu/carrottruck/fe
        sudo docker-compose down
        sudo docker image rm fe_react
        sudo docker-compose up -d
        ''',
        execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false,
        patternSeparator: '[, ]+', remoteDirectory: 'fe/build', remoteDirectorySDF: false,
        removePrefix: 'build', sourceFiles: 'build/**')],
        usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
        }
    }


    stage('ssh server') {
        dir('Backend') {
        sshPublisher(publishers: [sshPublisherDesc(configName: 'c211ec2',
        transfers: [sshTransfer(cleanRemote: false, excludes: '', execCommand: '''
        cd /home/ubuntu/carrottruck/be
        sudo docker-compose down
        sudo docker image rm be_springboot
        sudo docker-compose up -d
        ''',
        execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false,
        patternSeparator: '[, ]+', remoteDirectory: 'be/build', remoteDirectorySDF: false,
        removePrefix: 'build/libs', sourceFiles: 'build/libs/*.jar'
        )],
        usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
        }
    }
}
```

3-2 사업자

jenkins CI/CD Pipeline Script

```
node {
    stage('Initial'){
        sh "pwd"
        sh "rm -r ./**"
    }
    stage('Pull') {
        cleanWs()
        checkout scmGit(branches: [[name: '*/develop']], extensions: [], userRemoteConfigs: [[credentialsId: 'gitlab_carrottruck', url
    }

    stage('Change setting file'){
```

```
        sh "pwd"
        sh "cp /var/jenkins_home/.env.production /var/jenkins_home/workspace/CarrotTruck/frontend/vendor"
    }

    stage('Build'){
        dir('frontend/vendor') {
            nodejs('NodeJs 18.16.1') {
                sh "pwd"
                sh "npm install"
                sh "CI= npm run build"

            }
        }

    }

    stage('ssh back vendor') {
        dir('frontend/vendor') {
        sshPublisher(publishers: [sshPublisherDesc(configName: 'c211ec2',
        transfers: [sshTransfer(cleanRemote: false, excludes: '', execCommand: '''
        cd /home/ubuntu/carrottruck/fe_vendor
        sudo docker-compose down
        sudo docker image rm fe_vendor_react
        sudo docker-compose up -d
        ''',
        execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false,
        patternSeparator: '[, ]+', remoteDirectory: 'fe_vendor/build', remoteDirectorySDF: false,
        removePrefix: 'build', sourceFiles: 'build/**')],
        usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
        }
    }

}
```

nginx default conf파일

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;
```

```
        server_name _;
```

react.dockerfile

```
FROM node:18

RUN npm install -g serve

RUN mkdir ./build

COPY ./build ./build

ENTRYPOINT ["serve", "-s", "build"]
```

react.docker-compose

```
version: "3.3"

services:
  react:
    build:
      context: .
      dockerfile: react.dockerfile
    container_name: "react"

    expose:
      - "3000"

    networks:
      default_bridge:
        ipv4_address: 172.24.1.3

networks:
  default_bridge:
    ipam:
      driver: default
      config:
        - subnet: 172.24.1.0/24
```