자율 프로젝트 fakediary 포팅매뉴얼

빌드시 사용되는 환경변수 등의 주요 상세내용 기재

포팅매뉴얼 정리
현재 작성완료된것 : 개발환경, 사용기술, 환경 버전(자바 11, SpringBoot 2.7.10 Gradle

1.Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서 제출

백엔드

개발환경

Backend

Java : openjdk 11.0.18 2023-01-17 LTS

SpringBoot : 2.7.5
Intellij : IntelliJ IDEA 2023.1 (Ultimate Edition)

DB

MySQL : 8.0.32

Server

Server : Ubuntu 20.04 LTS

빌드

build.gradle

```
buildscript {
  ext {
    queryDslVersion = "5.0.0"
  }
}


plugins {
  id 'java'
  id 'org.springframework.boot' version '2.7.10'
  id 'io.spring.dependency-management' version '1.0.15.RELEASE'
  id "com.ewerk.gradle.plugins.querydsl" version "1.0.10"
}

jar{
  enabled = false
}

group = 'com.a101'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
  compileOnly {
    extendsFrom annotationProcessor
  }
}

repositories {
  mavenCentral()
  google()
  maven {
    url "https://firebase.google.com/maven/public"
  }
```

```
}

dependencies {
  implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
  //implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'
  //implementation 'org.springframework.boot:spring-boot-starter-security'
  implementation 'org.springframework.boot:spring-boot-starter-validation'
  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'
  annotationProcessor 'org.projectlombok:lombok'
  testImplementation 'org.springframework.boot:spring-boot-starter-test'
  //testImplementation 'org.springframework.security:spring-security-test'

  // https://mvnrepository.com/artifact/org.javassist/javassist
  implementation group: 'org.javassist', name: 'javassist', version: '3.29.0-GA'

  // Swagger
  implementation 'io.springfox:springfox-boot-starter:3.0.0'
  implementation 'io.springfox:springfox-swagger-ui:3.0.0'

  //  S3
  // https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-aws
  implementation group: 'org.springframework.cloud', name: 'spring-cloud-starter-aws', version: '2.2.6.RELEASE'
  // https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-aws-context
  implementation group: 'org.springframework.cloud', name: 'spring-cloud-aws-context', version: '2.2.6.RELEASE'
  // https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-aws-autoconfigure
  implementation group: 'org.springframework.cloud', name: 'spring-cloud-aws-autoconfigure', version: '2.2.6.RELEASE'

  //  WebClient
  implementation 'org.springframework.boot:spring-boot-starter-webflux'

  //  WebClient dependency 추가
  implementation 'org.springframework.boot:spring-boot-starter-webflux'

  // https://mvnrepository.com/artifact/org.json/json
  implementation group: 'org.json', name: 'json', version: '20230227'

  // https://mvnrepository.com/artifact/commons-io/commons-io
  implementation 'commons-io:commons-io:2.11.0'

  // Spring Batch
  implementation 'org.springframework.boot:spring-boot-starter-batch'

  // https://mvnrepository.com/artifact/org.springframework/spring-mock
  testImplementation 'org.springframework:spring-mock:2.0.8'

  //querydsl
  implementation "com.querydsl:querydsl-jpa:${queryDslVersion}"
  implementation "com.querydsl:querydsl-apt:${queryDslVersion}"

  // https://mvnrepository.com/artifact/com.google.firebase/firebase-admin
  implementation 'com.google.firebase:firebase-admin:9.1.1'
  implementation 'com.google.firebase:firebase-core:20.0.0'
  implementation 'com.google.firebase:firebase-messaging:23.0.0'

  // https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java
  implementation group: 'org.seleniumhq.selenium', name: 'selenium-java', version: '4.9.1'

  //  Python 코드 실행 위해 추가
  implementation 'org.apache.commons:commons-exec:1.3'
}

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

// querydsl 사용할 경로 지정합니다. 현재 지정한 부분은 .gitignore에 포함되므로 git에 올라가지 않습니다.
def querydslDir = "$buildDir/generated/'querydsl'"

// JPA 사용여부 및 사용 경로 설정
querydsl {
  jpa = true
  querydslSourcesDir = querydslDir
}

// build시 사용할 sourceSet 추가 설정
sourceSets {
  main.java.srcDir querydslDir
}


// querydsl 컴파일 시 사용할 옵션 설정
compileQuerydsl {
  options.annotationProcessorPath = configurations.querydsl
}
```

```
// querydsl이 compileClassPath를 상속하도록 설정
configurations {
  compileOnly {
    extendsFrom annotationProcessor
  }
  querydsl.extendsFrom compileClasspath
}
```

환경 변수(application.yml) : $이 포함되어 적혀있는 부분 수정하여 적용

```
spring:
  datasource:
    url: jdbc:mysql://${MYSQL_ENDPOINT}:3306/${DBNAME}?serverTimezone=Asia/Seoul
    username: ${USERNAME}
    password: ${PASSWORD}
    driver-class-name: com.mysql.cj.jdbc.Driver

  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    open-in-view: false
    show-sql: false # sql문 디버깅 필요할시 true
    generate-ddl: true
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
      open-in-view: false
  servlet:
    multipart:
      max-request-size: -1
      max-file-size: -1

  # spring batch
  batch:
    jdbc:
      initialize-schema: always # schema가 생성될 수 있도록

notification:
  mattermost:
    webhook-url: https://meeting.ssafy.com/hooks/tjy5k57fptbhffzpiekcrigfyh
    location: Springboot-Backend-local

cloud:
  aws:
    credentials:
      access-key: ${AWS_ACCESS_KEY}
      secret-key: ${AWS_SECRET_KEY}
    stack:
      auto: false
    s3:
      bucket: fakediary
      url: https://fakediary.s3.ap-northeast-2.amazonaws.com/
    region:
      static: ap-northeast-2

fake-diary:   # 접두사
  deep-art:
    base-url: https://api.deeparteffects.com/v1/noauth
    api-key: ${DEEP_ART_API_KEY}
    access-key: ${DEEP_ART_ACCESS_KEY}
    secret-key: ${DEEP_ART_SECRET_KEY}
  chat-gpt:
    api-key-3-5: ${CHAT_GPT_3.5_API_KEY}
    api-key-4-0: ${CHAT_GPT_4_API_KEY}
    model3-5: gpt-3.5-turbo
    model4-0: gpt-4-0314
    base-url: https://api.openai.com/v1/chat/completions
    max-tokens3-5: 3000
    max-tokens4-0: 7000
    n: 1
    temperature: 0.5
  stable-diffusion:
    max-memory-size: 134217728
    base-url: https://stablediffusionURL입력 (GPU서버에서 구동중)
  papago:
    base-url: https://openapi.naver.com/v1/papago/n2mt
```

```
  sound-raw:
    python: python3
    crawler: C:\\Users\\SSAFY\\Desktop\\proj\\macro\\crawler.py
    base-url: https://soundraw.io/edit_music

project:
  properties:
    firebase-create-scoped: "https://www.googleapis.com/auth/firebase.messaging"
    firebase-multicast-message-size: 500
```

Server

nginx, docker, jenkins 설치

배포과정

Developer → jenkins → nginx 무중단 배포

Developer

gitLab에 develop branch에 Push || Merge

1. Webhook 연결된 gitLab `develop브랜치 Push || Merge`
2. Jenkins 감지

Jenkins

1. `Git Clone` EC2 서버에서 develop branch를 git clone
2. `Make Yml File` clone한 프로젝트에 yml파일 생성하여 주입 yml파일 상단에 서버 포트를 정하는 코드 추가

```
sh 'touch application.yml'
                    sh '''
                    ACTIVE_ENVIRONMENT=$(grep "proxy_pass http://.*;" /etc/nginx/sites-available/default | awk -F'[/:;]' '{print
 $4}')

                    # Set the spring boot port based on the active environment
                    # 일기자동생성에서 포트번호감지를 위해 8080이아닌 8081, 8082로 설정
                    # blue면 green만들어야하니 8082로, 반대는 8081
                    if [ "$ACTIVE_ENVIRONMENT" = "blue" ]; then
                        SPRING_BOOT_PORT=8082
                    else
                        SPRING_BOOT_PORT=8081
                    fi

                    echo "
                    server:
                        port: $SPRING_BOOT_PORT
                        ~~~이후 진행
```

3. `Build Gradlew` 'gradlew clean build' 빌드파일 생성
4. `Docker Build` 'docker build -t ${id/projectName:Tag} .' 도커빌드 진행
5. `Docker Run` 현재 nginx의 proxy_pass가 blue인지, green인지 감지하여 감지되지 않은 색상의 도커 컨테이너 로그 출력, stop, remove 이후 run
6. `Run 20second Check` 20초를 기다린 후 방금 구동한 컨테이너가 정삭 작동중인것을 확인
7. `Update NGINX Configuration` Nginx proxy_pass를 방금 구동한 컨테이너 색상으로 변경후 'nginx -s reload' (약 0.1초소요)
8. `Docker Origin Image Remove` 사용중이지 않은 이전의 도커이미지 삭제

Stage View

	Git Clone	Make Yml File	Build Gradlew	Docker Build	Docker Run	Run 20second Check	Update NGINX Configuration	Docker Origin Image Remove
Average stage times: (Average full run time: ~54s)	441ms	964ms	20s	5s	3s	22s	314ms	312ms
#468 5월 18일 14:59 1 commit	389ms	994ms	20s	5s	1s	22s	317ms	319ms
#467 5월 18일 14:56 1 commit	442ms	973ms	20s	4s	3s	22s	311ms	312ms
#466 5월 18일 14:52 1 commit	427ms	959ms	19s	5s	3s	22s	313ms	310ms

Nginx

Nginx Configuration

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        client_max_body_size 0;
        proxy_read_timeout 600s;

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
        ##

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
```

```
        # Gzip Settings
        ##

        gzip on;

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/jav

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}


#mail {
#       # See sample authentication script at:
#       # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen     localhost:110;
#               protocol   pop3;
#               proxy      on;
#       }
#
#       server {
#               listen     localhost:143;
#               protocol   imap;
#               proxy      on;
#       }
#}
```

/etc/nginx/sites-available/default

```
upstream blue {
    server localhost:8081;
}

upstream green {
    server localhost:8082;
}

server {
    listen 80;
    server_name k8a101.p.ssafy.io;

    location / {
        proxy_pass http://blue; #상황에 맞게 젠킨스가 blue, green 변경
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/k8a101.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/k8a101.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 ($host = k8a101.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    return 404; # managed by Certbot
}
```

Ufw Status

```
To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
3306/tcp                   ALLOW       Anywhere
80                         ALLOW       Anywhere
443                        ALLOW       Anywhere
8888                       ALLOW       Anywhere
8888/tcp                   ALLOW       Anywhere
8081                       ALLOW       Anywhere
8082                       ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
3306/tcp (v6)              ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
8888 (v6)                  ALLOW       Anywhere (v6)
8888/tcp (v6)              ALLOW       Anywhere (v6)
8081 (v6)                  ALLOW       Anywhere (v6)
8082 (v6)                  ALLOW       Anywhere (v6)
```

22 : ssh

3306 : Database

80 : http

443 : https

8081 : Backend blue server

8082 : Backend green server