moti

포팅 매뉴얼

A509

고명진 / 유다윗 / 이상화 / 이연수 / 정석진 / 한원석

Written by 한원석

목차

기술 스택 및 버전

기술 스택

- **Version**

  Spring boot : v2.7.7,

  Java : openjdk v1.8.0_192,

  Apache Tomcat : v9.0.70,

  Vue : v2.7.14,

  MySQL : 8.0

  Node.js : v18.14.0(LTS),

  npm : 8.9.13,

Back-end 설정 및 배포

1. Spring boot 빌드 및 배포

- cd {프로젝트root} // 프로젝트 root 디렉토리로 이동
- ./gradlew clean build // gradle 빌드
- docker build –t moti_back . // docker 빌드
- docker container run -d --name moti_back -p 8080:8080 moti_back // docker 컨테이너 실행

2. Redis 배포

- docker pull redis // redis 설치
- docker container run -d --name redis -p 6379:6379 redis --requirepass asdf1234 // docker 컨테이너 실행

Front-end 설정 및 배포

1. Vue 배포

- npm i // module 설치
- npm run build // npm 빌드
- docker build –t moti_front . // docker 빌드
- docker container run -d --name moti -p 9000:9000 moti // docker 컨테이너 실행

CI/CD 설정

1. 동작원리

- Gitlab push 또는 merge시 gitlab hook 발생. SCM 요청에 의해 빌드 및 배포 실행

2. Jenkins pipeline

- **moti_back (back-end)**

```
-   pipeline {
-       agent any
-       triggers {
-           pollSCM('*/3 * * * *')
-       }
-       environment {
-           imagename = "moti_back"
-           dockerImage = ''
-       }
-       stages {
-           // git 에서 repository clone
-           stage('Prepare') {
-             steps {
-               echo 'Clonning Repository'
-               git url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12A509.git',
-                 branch: 'dev-back',
-                 credentialsId: '398bc7a7-b1e9-420c-87fa-3fe3d3676b64'
-               }
-               post {
-                success {
-                  echo 'Successfully Cloned Repository'
-                }
-                failure {
-                  error 'This pipeline stops here...'
-                }
```

```
-             }
-           }
-           // gradle build
-           stage('Bulid Gradle') {
-             steps {
-               echo 'Bulid Gradle'
-               sh 'chmod +x gradlew'
-               sh  './gradlew clean build'
-             }
-             post {
-               failure {
-                 error 'This pipeline stops here...'
-               }
-             }
-           }
-
-           // docker build
-           stage('Bulid Docker') {
-             steps {
-               echo 'Bulid Docker'
-               script {
-                   dockerImage = docker.build imagename
-               }
-             }
-             post {
-               failure {
-                 error 'This pipeline stops here...'
-               }
-             }
-           }
-           stage('Dockerise') {
-             steps {
-               echo 'Start Docker Container'
-               script {
-                   sh "docker rm -f moti_back"
-                   sh "docker rm -f redis"
-                   sh "docker container run -d --name moti_back -p 8080:8080 moti_back"
-                   sh "docker container run -d --name redis -p 6379:6379  redis --
    requirepass 'asdf1234'"
-               }
-             }
-             post {
-               failure {
-                 error 'This pipeline stops here...'
-               }
-             }
-           }
-       }
-   }
```

- **moti_front (front-end)**

```
pipeline {
    agent any

    triggers {
        pollSCM('*/3 * * * *')
    }
    environment {
        imagename = "moti"
        dockerImage = ''
    }
    stages {
        // git 에서 repository clone
        stage('Prepare') {
          steps {
            echo 'Clonning Repository'
            git url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12A509.git',
              branch: 'dev-front',
              credentialsId: '398bc7a7-b1e9-420c-87fa-3fe3d3676b64'
            }
            post {
             success {
               echo 'Successfully Cloned Repository'
             }
             failure {
               error 'This pipeline stops here...'
             }
          }
        }
        // npm build
        stage('Bulid npm') {
          steps {
            echo 'build'
            dir('front-end') {
                sh "npm i"
                sh "npm run build"
            }
          }
          post {
            failure {
              error 'This pipeline stops here...'
            }
          }
        }
        // docker build
```

```
-           stage('Bulid Docker') {
-             steps {
-               echo 'Bulid Docker'
-               dir('front-end') {
-                   script {
-                       dockerImage = docker.build imagename
-                   }
-               }
-
-             }
-             post {
-               failure {
-                 error 'This pipeline stops here...'
-               }
-             }
-           }
-           stage('Dockerise') {
-             steps {
-               echo 'Start Docker Container'
-               script {
-                   sh "docker rm -f moti"
-                   sh "docker container run -d --name moti -p 9000:9000 moti"
-               }
-             }
-             post {
-               failure {
-                 error 'This pipeline stops here...'
-               }
-             }
-           }
-       }
-   }
```

Nginx 설정

1. 도메인

- moti.today

2. ssl 발급

- certbot

3. default.conf

```
server {
    server_name moti.today;
    location / {
        proxy_pass http://localhost:9000;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $http_host;
    }
    location /api {
        proxy_pass http://localhost:8080$request_uri;
    }
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/moti.today/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/moti.today/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 ($host = moti.today) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
    listen 80;
    server_name moti.today;
    location / {
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
}
```