



## 개발 환경 및 프로젝트 기술 스택

### 이슈 관리

- Jira

### 형상 관리

- Gitlab

### UI/UX

- Figma

### Communication

- Mattermost
- Notion

### DB

- MySQL 8.0.32

### OS

- Windows 10
- Ubuntu 20.04 LTS

### IDE

- Visual Studio Code 1.76.2
- IntelliJ Ultimate 2022.3.1

### 기타 편의 툴

- Postman
- Sonarqube(ssafy) 8.9.9

### Front-End

- Node 18.12.1
- React 18.2.0
- Tailwind 3.2.7
- Chartjs 4.2.1

### Back-End

- Java-11.0.18
- Spring Boot 2.7.10
- Gradle 7.6.1
- JPA : latest

### IOT

- Raspberry Pi 4.0.0
- Dynamo DB
- Amazon Cogmito
- Amazon Iot Core
- Teachable Machine

### Server

- Amazon EC2
  - Docker 23.0.1
  - R3
- Jenkins
  - Jenkins 2.375.1
  - openjdk 11.0.17+8



## 백엔드 빌드 방법

intellij 터미널 실행

```
git clone https://lab.ssafy.com/s08-final/S08P31B202.git
```

```
cd BackEnd
```

ctl + shift + F11 -> 빌드 시작 후 자동 실행



## 프론트엔드 빌드 방법

---

vs code 터미널 실행

```
git clone https://lab.ssafy.com/s08-final/S08P31B202.git
```

```
cd FrontEnd
```

```
npm install --force
```

```
npm start
```



**Jenkins**

## Jenkins Pipeline

---

### Backend

---

```

pipeline {
    agent any

    stages {

        stage('Checkout and Clone') {
            steps {
git branch: 'develop', credentialsId: 'todaktodak', url: 'https://lab.ssafy.com/s08-
final/S08P31B202'
            }

            stage('Clone success'){
                steps {
                    echo 'Clone success'
                    sh 'ls -al'
                    sh 'cp /var/jenkins_home/todaktodak-6846e-firebase-adminsdk-v719d-
f41e0aa4e9.json /var/jenkins_home/workspace/test/BackEnd/src/main/resources/'

                }

            }

            stage('Spring Build') {
                steps {
                    dir('BackEnd') {
                        sh 'chmod +x gradlew'
                        sh './gradlew clean build -x test'
                        sh 'ls -al ./build'
                    }
                }
            }

            stage('Spring Build success'){
                steps {
                    echo 'Spring build success'
                    sh 'ls -al'
                }
            }

            // stage('Docker Build') {
            //     steps {
            //         sh 'docker build --build-arg JAR_FILE=build/libs/backend-0.0.1-
            //         SNAPSHOT.jar -t todaktodak/back ./BackEnd'
            //         sh 'docker ps'

            //     }
            // }

            stage('Throw Spring from Jenkins2EC2') {

```

```

    steps {
        script {
            sshagent(['d4667187-f34b-469b-bee7-0e2d1de481ef']) {
                sh "ls -al"
                sh "scp -i /root/.ssh/id_rsa BackEnd/build/libs/BackEnd-0.0.1-
                SNAPSHOT.jar ubuntu@172.26.1.175:/home/ubuntu"
                // sh "scp -i /root/.ssh/id_rsa DevOps/jenkinsfile
                ubuntu@172.26.1.175:/home/ubuntu"
            }
        }
    }
}

stage('Spring Docker Build & Run') {
    steps {
        script {
            sshagent(['d4667187-f34b-469b-bee7-0e2d1de481ef']) {

                sh '''
                    if docker inspect eomtiger/spring:todak-1.0 >/dev/null
2>&1; then

                        docker rmi eomtiger/spring:todak-1.0
                        echo "image deleted"
                    else
                        echo "image not found"
                    fi
                    '''

                sh '''
                    ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
build . -t eomtiger/spring:todak-1.0"
                    '''
                sh '''
                    ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
stop spring"
                    '''
                sh '''
                    ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
rm spring"
                    '''

                sh '''
                    ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
run -d -p 8080:8080 \
                    -e "ACTIVE=${ACTIVE}" \
                    -e "JWT_SECRET=${JWT_SECRET}" \
                    -e "KAKAO_CLIENT_SECRET_KEY=${KAKAO_CLIENT_SECRET_KEY}" \
                    -e "KAKAO_RESTAPI_KEY=${KAKAO_RESTAPI_KEY}" \
                    -e "PROD_NAME=${PROD_NAME}" \
                    -e "PROD_PASSWORD=${PROD_PASSWORD}" \

```

```
-e "S3_ACCESS_KEY=${S3_ACCESS_KEY}" \
-e "S3_BABY_IMAGE=${S3_BABY_IMAGE}" \
-e "S3_DEFAULT_IMAGE=${S3_DEFAULT_IMAGE}" \
-e "S3_SECRET_KEY=${S3_SECRET_KEY}" \
-e "BUCKET_ADDRESS=${BUCKET_ADDRESS}" \
-e "S3_USER_IMAGE=${S3_USER_IMAGE}" \
-e "KEY_STORE_PASSWORD=${KEY_STORE_PASSWORD}" \
-e "KEY_ALIAS=${KEY_ALIAS}" \
-e "FCM_KEY_LOCATION=${FCM_KEY_LOCATION}" \
-e "FCM_PROJECT_ID=${FCM_PROJECT_ID}" \
-e "KAKAO_REDIRECT_URI=${KAKAO_REDIRECT_URI}" \
--name spring eomtiger/spring:todak-1.0"
...

```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

## Frontend

---

```

pipeline {
    agent any
    tools{
        nodejs 'node16'
    }

    stages {
        stage('Checkout and Clone') {
            steps {
                git branch: 'develop', credentialsId: 'todaktodak', url:
'https://lab.ssafy.com/s08-final/S08P31B202'}
            }

            stage('Clone success'){
                steps {
                    echo 'Clone success'
                    sh 'ls -al'
                }
            }

            stage('React Build') {
                steps {
                    dir('FrontEnd') {
                        sh 'npm i --save --legacy-peer-deps'
                        sh 'npm run build'
                    }
                }
            }

            stage('React Build success'){
                steps {
                    echo 'React build success'
                    sh 'ls -al'
                }
            }

            // stage('Docker Build') {
            //     steps {
            //         sh 'docker build --build-arg JAR_FILE=build/libs/backend-0.0.1-
            //         SNAPSHOT.jar -t todaktodak/back ./BackEnd'
            //         sh 'docker ps'
            //     }
            // }

            stage('Throw React from Jenkins2EC2') {
                steps {
                    script {
                        sshagent(['d4667187-f34b-469b-bee7-0e2d1de481ef']) {
                            sh "ls -al"
                            sh "scp -r /root/.ssh/id_rsa FrontEnd/build"
                        }
                    }
                }
            }
        }
    }
}

```

```
ubuntu@172.26.1.175:/home/ubuntu/react"
```

```
    }
  }
}
stage('React Docker Build & Run') {
  steps {
    script {
      sshagent(['d4667187-f34b-469b-bee7-0e2d1de481ef']) {

        sh '''
          if docker inspect eomtiger/react:today-1.0 >/dev/null
2>&1; then
            docker rmi eomtiger/react:today-1.0
            echo "image deleted"
          else
            echo "image not found"
          fi
          ...
        dir('react'){
          sh '''
            ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175
"docker build . -t eomtiger/spring:today-1.0"
            ...
          }
          sh '''
          ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
stop react"
          ...
          sh '''
          ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
rm react"
          ...

          sh '''
          ssh -o "StrictHostKeyChecking=no" ubuntu@172.26.1.175 "docker
run -d -p 3000:3000 \
--name react eomtiger/react:today-1.0"
          ...
        }
      }
    }
  }
}
```