



# 서버 포스팅 방법

각 서버에 맞도록 파일을 Home 디렉토리에 배포한다.

Spring 2.7.16

Hadoop 3.3.6

nodejs 18.17.1

→ 하둡 인스턴스가 많을수록 빨라짐

▼ 각 홈에 풀어 놓기.

EC2-1.zip

env.test.properties

## EC2 - 1 서버

필요 소프트웨어 Docker, Docker-compose, openSSH

셋팅 방법

Docker-compose.yml, nginx-init.yml, nginx.conf, nginx-init.conf를 수정한다.

nginx-init.yml을 이용하여 Docker-compose를 실행한다.

일반 docker-compose.yml을 이용하여 Docker-compose를 실행한다.

openSSH 공개키와 개인키를 백업 받는다.

젠킨스에 접속한다.

젠킨스에서 NodeJS플러그인을 설치하고 18.17.1을 셋팅해둔다.

각 서버마다 아래는 PipeLine이다.

▼ EC2-1

프론트

```
p
pipeline {
  agent { node { label 'EC2-1' } }
  tools {
    nodejs 'nodejs18.17.1'
  }
  stages{
    stage('git pull') {
      steps{
        sh "echo 'Ready'"
        git branch: 'develop', credentialsId: 'dkdkdkal123-naver', url: 'https://lab.ssafy.com/s09-bigdata-c
      }
    }

    stage('Front Build') {
      steps{
        sh "node -v"
        sh "cd ~ && pwd"
        sh "cd fe && npm install"
        sh "cd fe && npm run build"
      }
    }

    stage('Front Old Version Remove') {
      steps{
        sh "rm -rf /var/www/web/test"
```

```

    }
  }

  stage('Front New Version Publish') {
    steps{
      sh "cp -r ./fe/build /var/www/web/test"
    }
  }
}

```

## 백엔드

```

pipeline {
  agent { node { label 'EC2-1' } }
  stages{
    stage('git pull') {
      steps{
        sh "echo 'Ready'"
        git branch: 'develop', credentialsId: 'dkdkdkal123-naver', url: 'https://lab.ssafy.com/s09-bigdata-c
      }
    }

    stage('Inject Properties'){
      steps {
        sh "ls -al"

        sh "cp /env.test.properties ./BackEnd/src/main/resources/properties/env.properties"
      }
    }
    stage('BackEnd Build') {
      steps{
        sh "cd BackEnd && chmod +x gradlew && ./gradlew build"
      }
    }

    stage('BackEnd Image Build') {
      steps{
        sh 'if [ $(docker ps -aqf "name=testwascontainer") ]; then docker stop testwascontainer; fi'
        sh "cd BackEnd && docker build -t testwasimage --build-arg JAR_FILE=build/libs/BackEnd-0.0.1-SNAPSHOT
      }
    }
    stage('BackEnd Container start') {
      steps{
        sh "docker run -it -d --rm -p 8181:8080 --name testwascontainer testwasimage"
      }
    }
  }
}

```

## ▼ EC2-2

### 프론트

Docker로 대체되었습니다.

## 백엔드

```

pipeline {
  agent {
    label 'EC2-2'
  }
  stages {
    stage('git pull') {

      steps {
        git branch: 'develop', credentialsId: 'dkdkdkal123-naver', url: 'https://lab.ssafy.com/s09-bigdata-c
        sh "ls"
      }
    }
    stage('Inject Properties'){
      steps {
        sh "pwd"
        sh "cd ~ && ls"
        sh "cat ~/env.test.properties"
        sh "mkdir -p ~/jenkins/workspace/HLSTest/Hls/src/main/resources/properties"
      }
    }
  }
}

```

```

        sh "cp ~/env.test.properties ~/jenkins/workspace/HLS/Hls/src/main/resources/properties/env.properties"
    }
}
stage('HLS Gradle Build') {
    steps {
        sh "ls"
        sh "cd Hls && chmod 777 gradlew && ./gradlew build"
        sh "id"
    }
}
stage('Docker HLS Run jar') {
    steps {
        sh "if [ \$(jps -V | grep test.jar | awk '{print \$1}')] ]; then jps -V | grep test.jar | awk '{print \$1}'"
        sh "cp ./Hls/build/libs/BackEnd-0.0.1-SNAPSHOT.jar ./test.jar"
        sh "JENKINS_NODE_COOKIE=dontKillMe ./test.sh"
    }
}
}
}
}

```

## ▼ Cluster1

### 하둡

```

pipeline {
    agent {
        label 'oracle-hadoop-1'
    }
    stages{
        stage('git pull') {
            steps{
                git branch: 'master', credentialsId: 'dkdkdkal123-naver-hadoop', url: 'https://lab.ssafy.com/dkdkdkal123-naver-hadoop'
            }
        }
        stage('build gradle') {
            steps{
                sh 'chmod 777 gradlew && ./gradlew build'
            }
        }
    }
}

```

### 스프링

```

pipeline {
    agent {
        label 'oracle-hadoop-1'
    }
    stages{
        stage('git pull') {
            steps{
                git branch: 'develop', credentialsId: 'dkdkdkal123-naver', url: 'https://lab.ssafy.com/s09-bigdata-c'
            }
        }
        stage('Inject Properties') {
            steps{
                sh "mkdir -p Split/src/main/resources/properties"
                sh "cp ~/env.properties Split/src/main/resources/properties/env.properties"
            }
        }
        stage('build gradle') {
            steps{
                sh 'cd Split && chmod 777 gradlew && ./gradlew build'
            }
        }
        stage('AISpringServer Run jar') {
            steps {
                sh "if [ \$(jps -V | grep test.jar | awk '{print \$1}')] ]; then jps -V | grep test.jar | awk '{print \$1}'"
                sh "cp Split/build/libs/Split-0.0.1-SNAPSHOT.jar ./test.jar"
                sh "JENKINS_NODE_COOKIE=dontKillMe sh ./test.sh"
            }
        }
    }
}

```

▼

## 셋팅 방법

openSSH 공개키와 개인키를 백업 받는다.

python3 HDFSSocketServer.py. → Git AI폴더 참조.

`sudo ufw allow from IP` 를 이용하여 서로를 허용한다.

