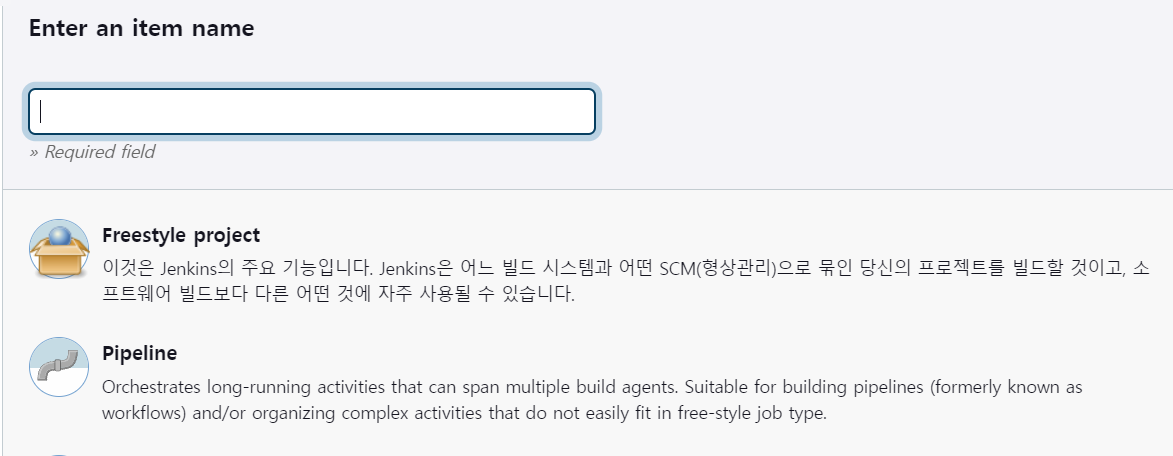
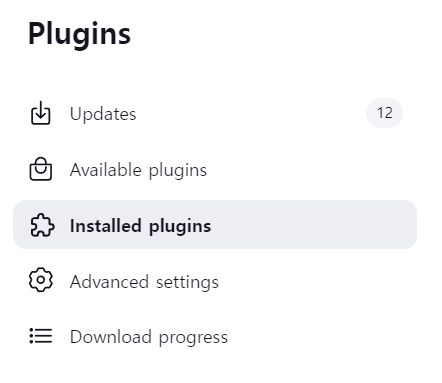
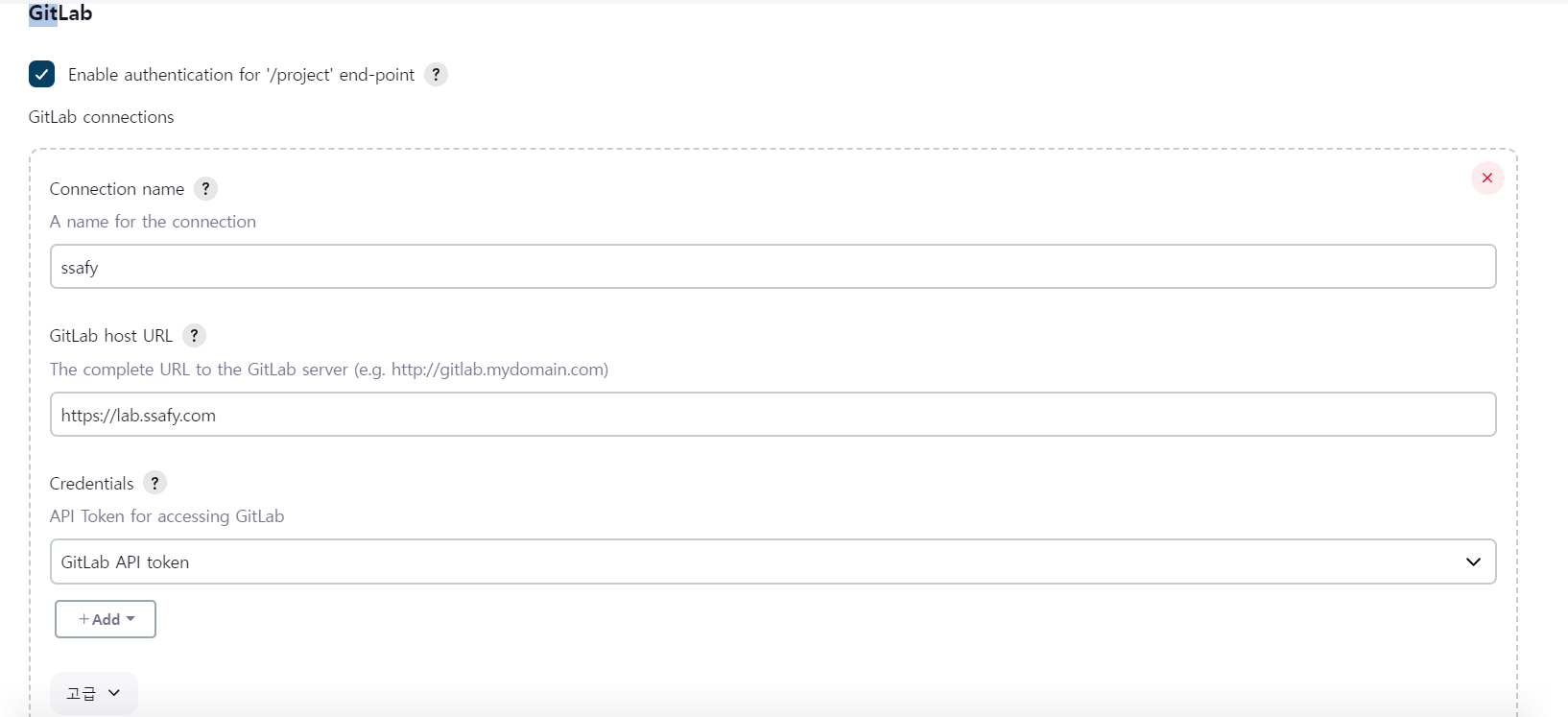
1. 새로운 pipeline 아이템 생성

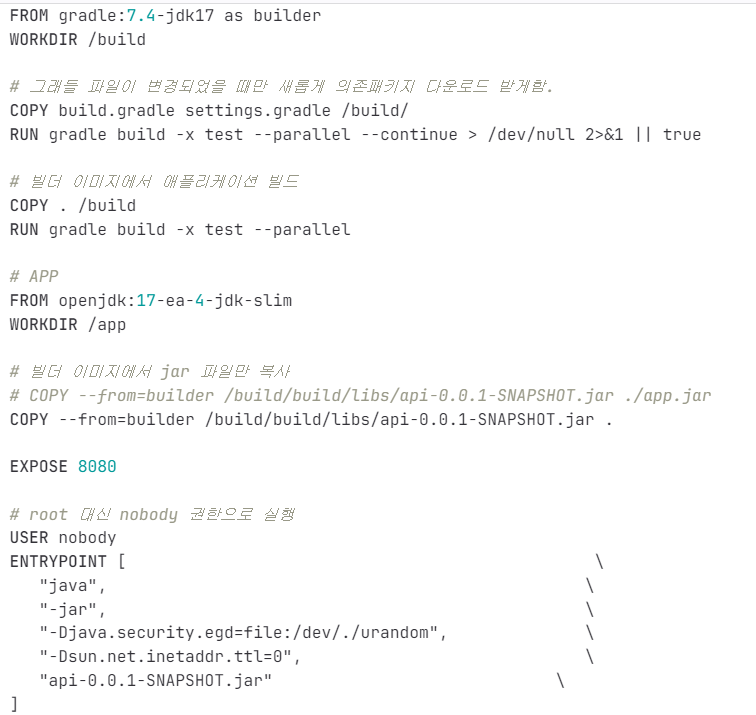


1. Gitlab, GitLab Authentication, GitLab API 플러그인 설치



1. Gitlab 설정 (Test connection success 시 성공)

(Credentials 본인 gitlab -user settings – Access token 생성)  


1. 백엔드 폴더 내 Dockerfile 생성   
   
2. 생성한 Item -> 구성 -> Pipeline 작성

pipeline

{

agent any

stages{

stage('connection') //깃랩 연결

{

steps

{

git branch: 'master', //브랜치

credentialsId: 'kofoos',

url: 'https://lab.ssafy.com/s10-webmobile2-sub2/S10P12A309.git'

}

}

stage('build') // 빌드 테스트

{

steps

{ sh 'cd /var/jenkins\_home/workspace/kofoos/server/src/main/resources'

dir('resources'){

sh ‘설정 파일들 생성’

}

sh 'cd ..'

sh 'cd ..'

sh 'cd ..'

dir('server'){

sh 'chmod +x ./gradlew'

sh './gradlew clean build'

}

}

}

stage('deploy'){ // 배포

steps{

sh 'docker stop spring && docker rm spring && docker rmi backend'

dir('server'){

sh 'docker build -t backend ./'

sh 'docker run --restart=on-failure -p 8080:8080 -d --name spring backend'

}

}

}

}

post { //메타모스트 연결

success {

script {

def Author\_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()

def Author\_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()

mattermostSend (color: 'good',

message: "빌드 성공: ${env.JOB\_NAME} #${env.BUILD\_NUMBER} by ${Author\_ID}(${Author\_Name})\n(<${env.BUILD\_URL}|Details>)",

endpoint: 'https://meeting.ssafy.com/hooks/o5dykndwbj8s9r5rz5uk1odjmc',

channel: 'jenkins-kofoos'

)

}

}

failure {

script {

def Author\_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()

def Author\_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()

mattermostSend (color: 'danger',

message: "배포 실패: ${env.JOB\_NAME} #${env.BUILD\_NUMBER} by ${Author\_ID}(${Author\_Name})\n(<${env.BUILD\_URL}|Details>)",

endpoint: 'https://meeting.ssafy.com/hooks/o5dykndwbj8s9r5rz5uk1odjmc',

channel: 'jenkins-kofoos'

)

}

}

}

}