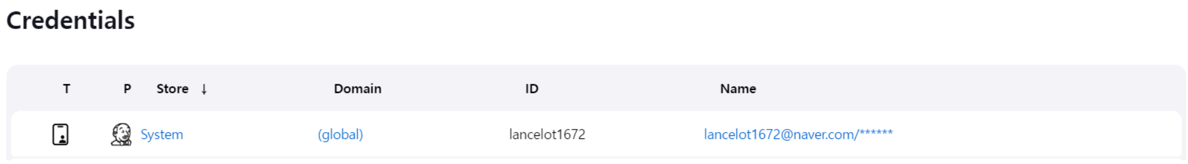
Jenkins 배포 Script

젠킨스 Credentials



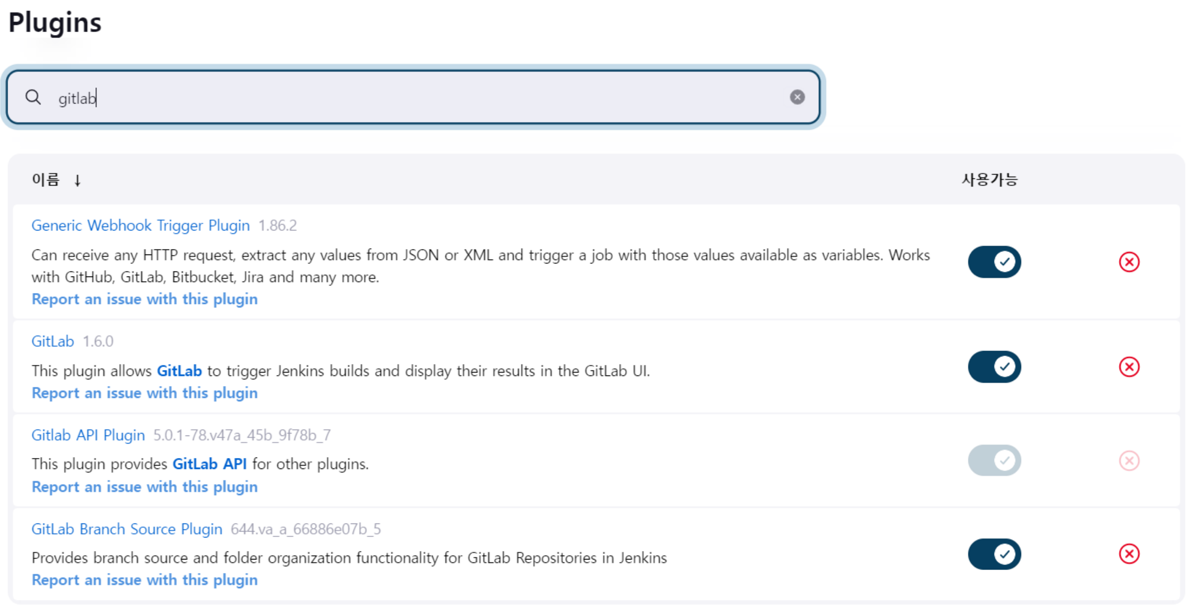
Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	lancelot1672	lancelot1672@naver.com/******

젠킨스 플러그인

- 원활한 배포를 위하여 아래 해당하는 플러그인을 설치해야한다.

GitLab

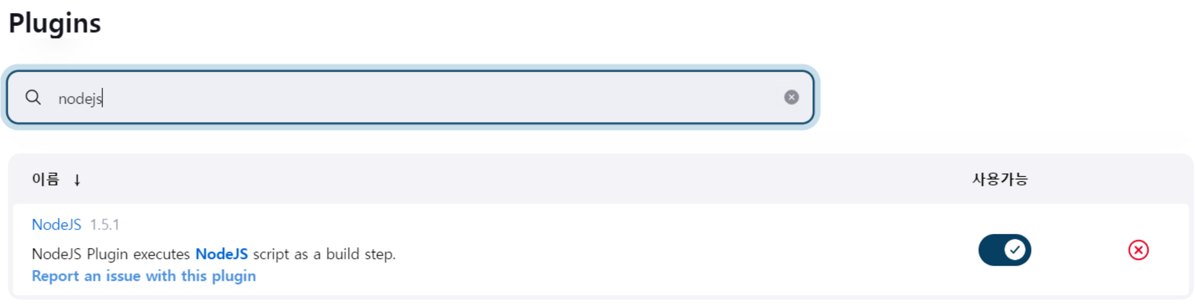


Plugins

gitlab

이름 ↓	사용가능	
Generic Webhook Trigger Plugin 1.86.2 Can receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. Works with GitHub, GitLab, Bitbucket, Jira and many more. Report an issue with this plugin	✓	⊗
GitLab 1.6.0 This plugin allows **GitLab** to trigger Jenkins builds and display their results in the GitLab UI. Report an issue with this plugin	✓	⊗
Gitlab API Plugin 5.0.1-78.v47a_45b_9f78b_7 This plugin provides **GitLab API** for other plugins. Report an issue with this plugin	✓	⊗
GitLab Branch Source Plugin 644.va_a_66886e07b_5 Provides branch source and folder organization functionality for GitLab Repositories in Jenkins Report an issue with this plugin	✓	⊗

NodeJS(NPM)



Plugins

nodejs

이름 ↓	사용가능	
NodeJS 1.5.1 NodeJS Plugin executes **NodeJS** script as a build step. Report an issue with this plugin	✓	⊗

젠킨스 스크립트

BackEnd 프로젝트

▼ PJT1 - Bussiness

```
node {
    stage('clone') {
        git branch: 'develop',
            credentialsId : 'lancelot1672',
            url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12B310.git'
    }
    stage('properties'){
        sh 'ls -al'
        sh 'cp /var/jenkins_home/application/pjt1/env.properties /var/jenkins_home/workspace/backend-bussiness/BackEnd/pjt1/src/ma
        sh 'cp /var/jenkins_home/application/pjt1/application.yml /var/jenkins_home/workspace/backend-bussiness/BackEnd/pjt1/src/m
        sh 'cp /var/jenkins_home/application/pjt1/keystore.p12 /var/jenkins_home/workspace/backend-bussiness/BackEnd/pjt1/src/mai
    }
    dir('BackEnd/pjt1'){
        stage('Build') {
      // gralew이 있어야됨. git clone해서 project를 가져옴.
        sh  './gradlew clean build -x test'


        sh 'ls -al ./build'
        }
    }
    stage('Docker rm') {
        try{
            sh 'docker stop pjt1-container'
            sh 'docker rm pjt1-container'
        }catch(e){
            sh 'echo "docker rm Fail!!"'
        }
    }
    dir('BackEnd/pjt1'){
        stage('Dockerizing'){
            sh 'echo "Image Bulid Start"'
            sh 'docker build -t moonrise-pjt1:1.0 .'
        }
    }
    stage('Deploy') {
        sh 'docker run --name pjt1-container --network test-network -d -p 9001:9001 -v /etc/localtime:/etc/localtime:ro -e TZ=Asia
    }
}
```

▼ PJT2 - Auth

```
node {
    stage('clone') {
        git branch: 'develop',
            credentialsId : 'lancelot1672',
            url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12B310.git'
    }
    stage('properties'){
        sh 'cp /var/jenkins_home/application/pjt2/application.yml /var/jenkins_home/workspace/backend-auth/BackEnd/pjt2/src/main/r

        sh 'cp /var/jenkins_home/application/pjt2/env.properties /var/jenkins_home/workspace/backend-auth/BackEnd/pjt2/src/main/re
        sh 'cp /var/jenkins_home/application/pjt2/keystore.p12 /var/jenkins_home/workspace/backend-auth/BackEnd/pjt2/src/main/reso
    }
    dir('BackEnd/pjt2'){
        stage('Build') {
        sh 'chmod +x ./gradlew'
      // gralew이 있어야됨. git clone해서 project를 가져옴.
        sh  './gradlew clean build -x test'


        sh 'ls -al ./build'
        }
    }
    stage('Docker rm') {
        try{
            sh 'docker stop pjt2-container'
            sh 'docker rm pjt2-container'
        }catch(e){
            sh 'echo "docker rm Fail!!"'
        }
    }
    dir('BackEnd/pjt2'){
        stage('Dockerizing'){
            sh 'echo "Image Bulid Start"'
            sh 'docker build -t moonrise-pjt2:1.0 .'
        }
    }
    stage('Deploy') {
        sh 'docker run --name pjt2-container --network test-network -d -p 9002:9002 -v /etc/localtime:/etc/localtime:ro -e TZ=Asia

    }
}
```

▼ PJT3 - Chat

```
node {
    stage('clone') {
        git branch: 'develop',
            credentialsId : 'lancelot1672',
            url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12B310.git'
    }
    stage('properties'){
        sh 'cp /var/jenkins_home/application/pjt3/application.yml /var/jenkins_home/workspace/backend-chat/BackEnd/pjt3/src/main/r

        sh 'cp /var/jenkins_home/application/pjt3/keystore.p12 /var/jenkins_home/workspace/backend-chat/BackEnd/pjt3/src/main/reso
        }
    dir('BackEnd/pjt3'){
        stage('Build') {
        sh 'chmod +x ./gradlew'
      // gralew이 있어야됨. git clone해서 project를 가져옴.
        sh  './gradlew clean build -x test'


        sh 'ls -al ./build'
        }
    }
    stage('Docker rm') {
        try{
            sh 'docker stop pjt3-container'
            sh 'docker rm pjt3-container'
        }catch(e){
            sh 'echo "docker rm Fail!!"'
        }
    }
    dir('BackEnd/pjt3'){
        stage('Dockerizing'){
            sh 'echo "Image Bulid Start"'
            sh 'docker build -t moonrise-pjt3:1.0 .'
        }
    }
    stage('Deploy') {
        sh 'docker run --name pjt3-container --network test-network -d -p 9003:9003 -v /etc/localtime:/etc/localtime:ro -e TZ=Asia
    }
}
```

FrontEnd 프로젝트

▼ React Script

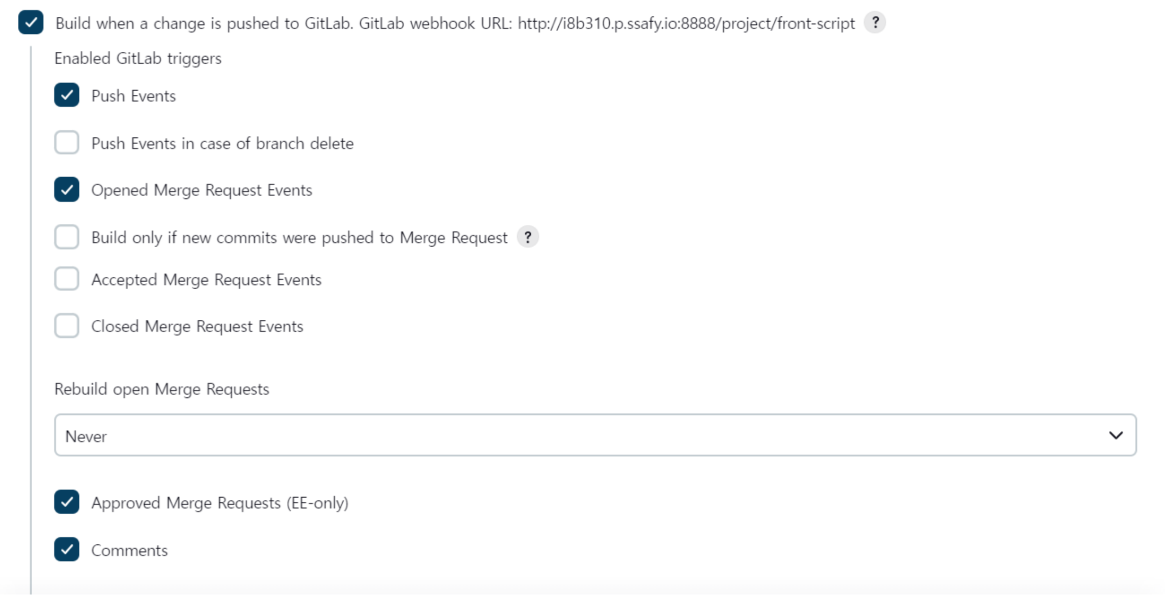
```
node {
    stage('npm setting') {
        env.NODEJS_HOME = "${tool 'nodejs'}"
        // on linux / mac
        env.PATH="${env.NODEJS_HOME}/bin:${env.PATH}"

        sh 'npm --version'
    }
    stage('clone') {
        git branch: 'develop-front',
            credentialsId : 'lancelot1672',
            url: 'https://lab.ssafy.com/s08-webmobile2-sub2/S08P12B310.git'
    }
    stage('env setting'){
        sh 'cp /var/jenkins_home/env/release/.env /var/jenkins_home/workspace/front-release/frontend'
    }
    dir('frontend'){
        stage('npm build'){
            sh 'npm install --force'
            sh 'npm run build'
        }
    }

    stage('build move'){
        sh 'docker cp /var/jenkins_home/workspace/front-release/frontend/build nginx-container2:/usr/share/nginx/html'
    }
}
```

Jenkins Web Hook(웹 훅) 등록

- 특정 branch에 push/merge 할 경우 Jenkins가 이를 감지 및 자동 빌드

▼ WebHook 등록 과정

1. Build when a change is pushed to GitLab 체크



☑ Build when a change is pushed to GitLab. GitLab webhook URL: http://i8b310.p.ssafy.io:8888/project/front-script

Enabled GitLab triggers

☑ Push Events

□ Push Events in case of branch delete

☑ Opened Merge Request Events

□ Build only if new commits were pushed to Merge Request

□ Accepted Merge Request Events

□ Closed Merge Request Events

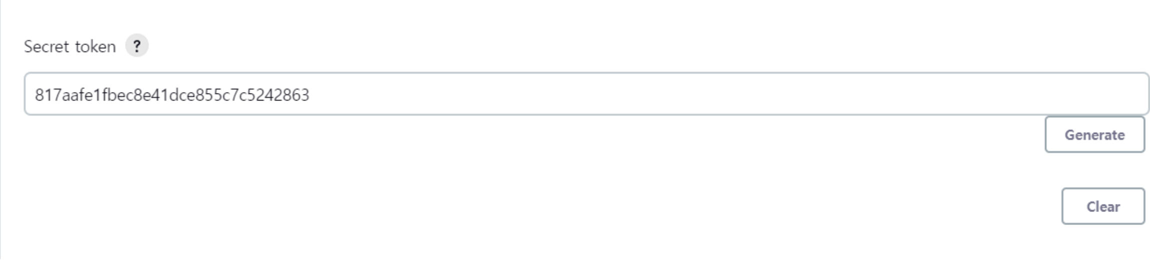
Rebuild open Merge Requests

Never

☑ Approved Merge Requests (EE-only)

☑ Comments

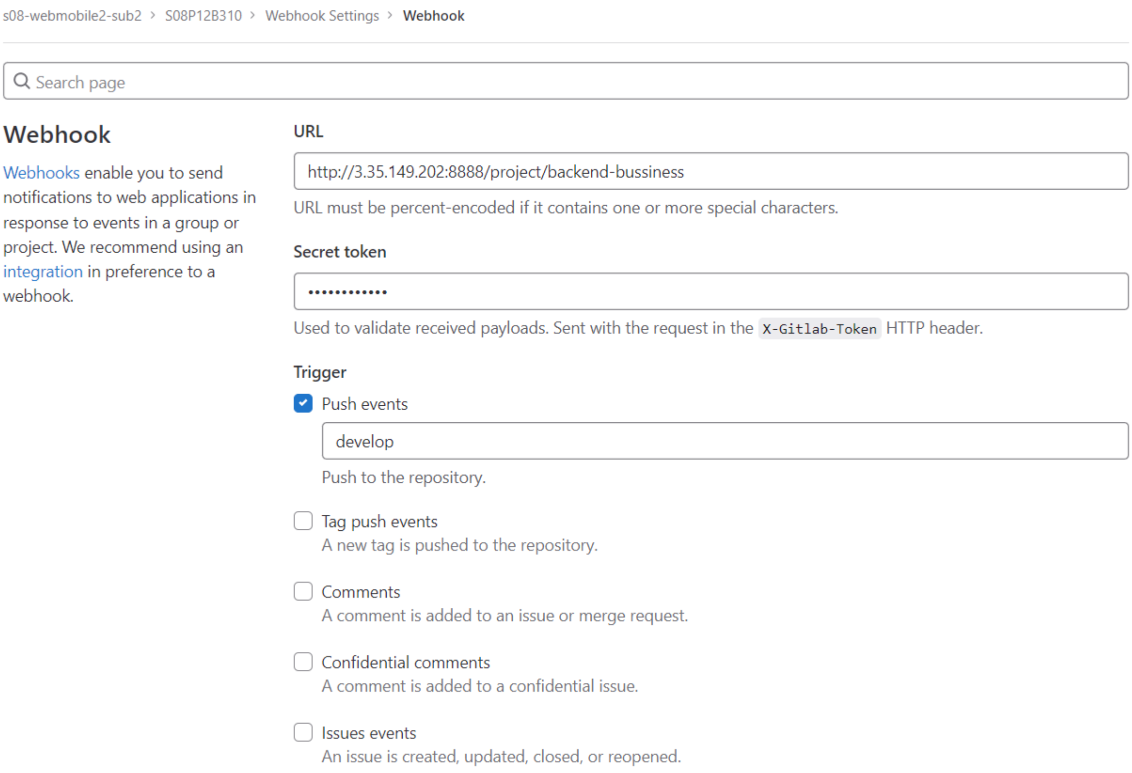
2. Secret Ket 발급



Secret token

817aafe1fbec8e41dce855c7c5242863

3. Gitlab WebHook 등록



s08-webmobile2-sub2 › S08P12B310 › Webhook Settings › **Webhook**

Webhook

Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an integration in preference to a webhook.

URL

http://3.35.149.202:8888/project/backend-bussiness

URL must be percent-encoded if it contains one or more special characters.

Secret token

••••••••••••

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.

Trigger

☑ Push events

develop

Push to the repository.

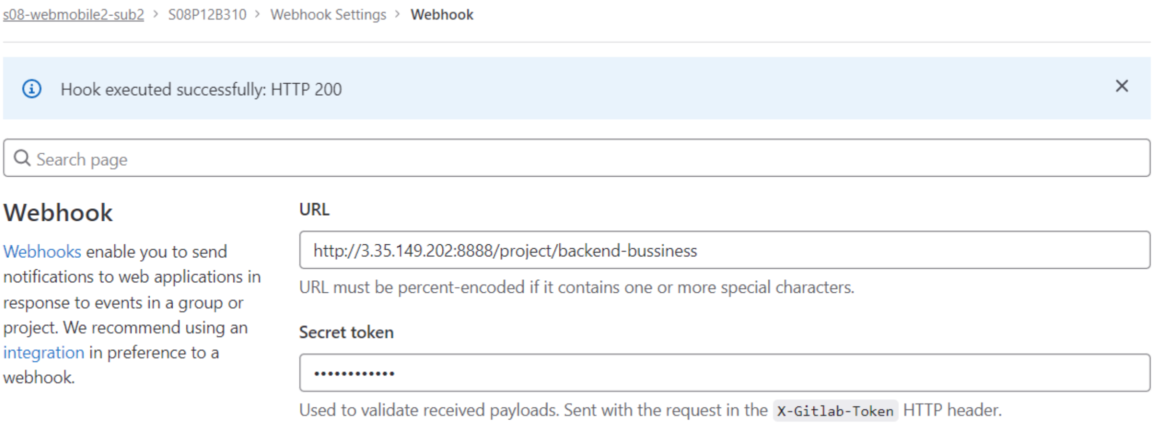
□ Tag push events
A new tag is pushed to the repository.

□ Comments
A comment is added to an issue or merge request.

□ Confidential comments
A comment is added to a confidential issue.

□ Issues events
An issue is created, updated, closed, or reopened.

4. Test



s08-webmobile2-sub2 › S08P12B310 › Webhook Settings › **Webhook**

Hook executed successfully: HTTP 200

Webhook

Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an integration in preference to a webhook.

URL

http://3.35.149.202:8888/project/backend-bussiness

URL must be percent-encoded if it contains one or more special characters.

Secret token

••••••••••••

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.