

# Jenkins

- jenkins는 java로 개발되어 있기 때문에 jdk 설치가 필수임

## ▼ 자바 설치 확인

```
$ java -version
```

자바가 설치되어 있지 않을 경우

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install openjdk-11-jdk
```

## ▼ 젠킨스 설치

키를 다운 받아야 젠킨스 설치 가능

```
$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
```

jenkins.list에 주소 추가

```
$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

설치

```
$ sudo apt-get update
$ sudo apt-get install jenkins
```

## ▼ CI 설정

설명 입력

**General**   소스 코드 관리   빌드 유발   빌드 환경   Build   빌드 후 조치

설명

develop 서버 CI 설정입니다

[Plain text] [미리보기](#)

## 오래된 빌드 삭제

☒ 오래된 빌드 삭제

Strategy

Log Rotation

빌드 이력 유지 기간(일)

공백일 경우, [보관할 최대갯수] 만큼 기록됩니다.

보관할 최대갯수

5

if not empty, only up to this number of build records are kept

[고급...](#)

## 소스코드관리

### Git으로 설정

git lab 해당 repository에 가서 deploy token 발급

Repository URL ⇒ 토큰이름:키@저장소주소

### 소스 코드 관리

☐ None  
☒ Git

Repositories

Repository URL

https://gitlab+deploy-token-2566:yaPV6x9BLxLGbG71pfoE@lab.ssafy.com/s04-final/DaHaeng

Credentials

- none - Add

Name

Refspec

Add Repository

### 빌드할 브랜치 입력

Branches to build

Branch Specifier (blank for 'any')

\*/develop

Add Branch

### 빌드유발 선택

### 빌드 유발

☐ 빌드를 원격으로 유발 (예: 스크립트 사용)  
☐ Build after other projects are built  
☐ Build periodically  
☒ Build when a change is pushed to GitLab. GitLab webhook URL: https://jenkins.da-haeng.com/project/kichulove

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

그 밑에 있는 고급 버튼 누르기

Rebuild open Merge Requests

Never

☒ Approved Merge Requests (EE-only)

☒ Comments

Comment (regex) for triggering a build

Jenkins please retry a build

고급...

확장된 탭에서 secret token 밑에 있는 Generate 버튼 클릭

☒ Enable [ci-skip]

☒ Ignore WIP Merge Requests

Labels that forces builds if they are added (comma-separated)

☒ Set build description to build cause (eg. Merge request or Git Push )

☐ Build on successful pipeline events

Pending build name for pipeline

☐ Cancel pending merge request builds on update

Allowed branches

☒ Allow all branches to trigger this job

☐ Filter branches by name

☐ Filter branches by regex

☐ Filter merge request by label

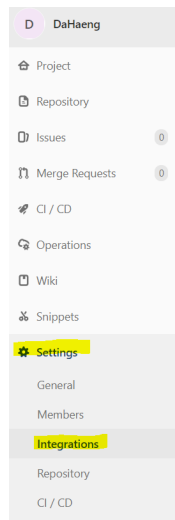
Secret token

Generate

발급된 token으로 git lab repo에 webhook 추가하기

push event 밑에 브랜치 이름 넣기

add webhook 버튼 클릭



s04-final > DaHaeng > Integrations Settings

## Integrations

Webhooks can be used for binding events when something is happening within the project.

### URL

https://jenkins.da-haeng.com/project/kichulove

### Secret Token

cc9d4d065c438728d20a2da480b1dfd3

Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.

### Trigger

#### ☒ Push events

This URL will be triggered by a push to the repository

develop

#### ☐ Tag push events

This URL will be triggered when a new tag is pushed to the repository

#### ☐ Comments

This URL will be triggered when someone adds a comment

#### ☐ Confidential Comments

This URL will be triggered when someone adds a comment on a confidential issue

#### ☐ Issues events

This URL will be triggered when an issue is created/updated/merged

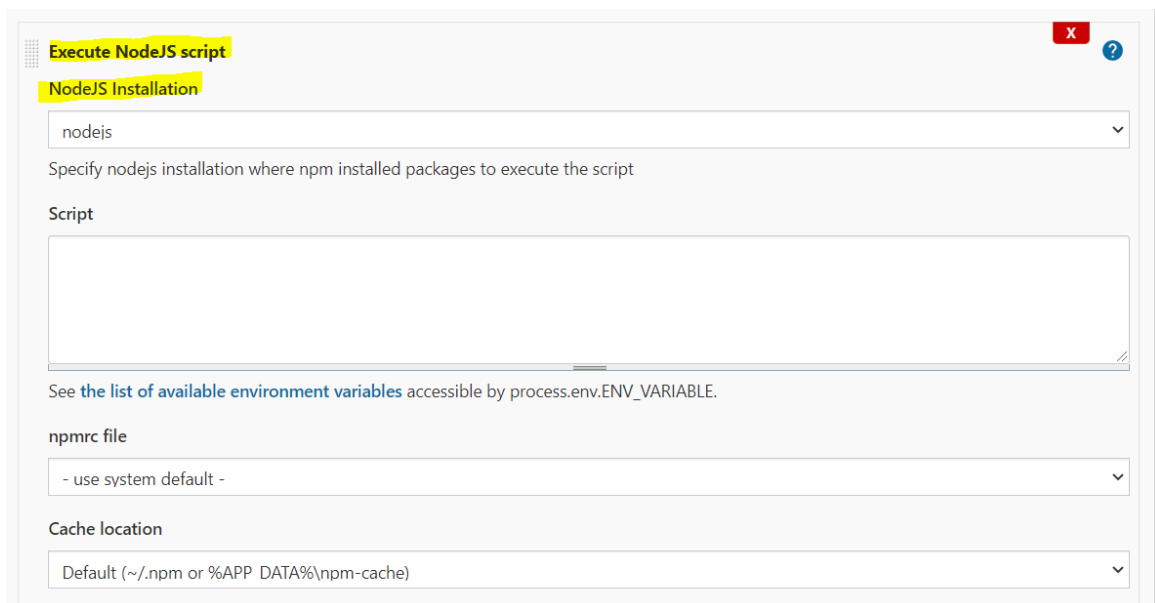
#### ☐ Confidential Issues events

This URL will be triggered when a confidential issue is created/updated/merged

## 빌드 1

Execute NodeJS script 추가

내가 만든 nodejs 추가



## 빌드 2

Execute shell 추가

command 추가



여기서 test 할 때 오류 발생

⇒ context를 load하지 못함

⇒ application.yaml이 없어서 db에 접근을 못함

해결책

```
cd ..
mkdir ../back/src/test/resources
cp /home/ubuntu/da-haeng/backend/application.yaml ../back/src/test/resources/application.yaml
```

코드를 추가하여 빌드 과정에서 test를 정상적으로 진행할 수 있도록 함

빌드3

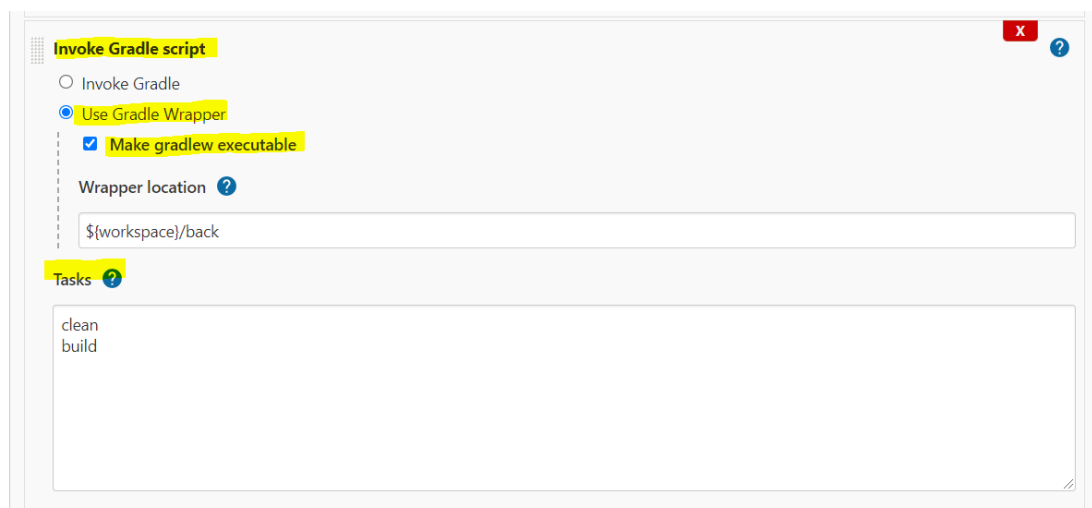
Invoke gradle script 추가

use gradle wrapper 선택

make gradlew executable 선택

wrapper location 추가

Task 추가



쪽 내려서 Build file 추가

**Build File** ?

Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)

☐ Force GRADLE\_USER\_HOME to use workspace ?

jenkins 관리 > 시스템 설정

Dashboard

- 새로운 Item
- 사람
- 빌드 기록
- Jenkins 관리**
- My Views
- Lockable Resources
- New View

빌드 대기 목록


빌드 대기 항목이 없습니다.


빌드 실행 상태


- 1 대기 중
- 2 대기 중


## Jenkins 관리


### System Configuration

**시스템 설정**  
환경 변수 및 경로 정보등을 설정합니다.


**Global Tool Configuration**  
Configure tools, their locations and automatic installers.


**플러그인 관리**  
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.


**노드 관리**  
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.


**Managed files**  
e.g. settings.xml for maven, central managed scripts, custom files, ...


### Security

**Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.

**Manage Credentials**  
Configure credentials

**Configure Credential Providers**  
Configure the credential providers and types

**Manage Users**  
Create/delete/modify users that can log in to this Jenkins

**In-process Script Approval**  
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

쪽 내려서 Publish over SSH 밑에 있는 추가 버튼 클릭

**Publish over SSH**

Jenkins SSH Key ?

Passphrase ?

[Change Password](#)

Path to key ?

Key ?

☐ Disable exec ?

적당히 잘 입력하고 Auth fail 나오면 됨 (key를 등록 안 한 상태니까)

SSH Server

Name ?

kichul love

Hostname ?

13.125.175.136

Username ?

ubuntu

Remote Directory ?

/home/ubuntu/da-haeng

고급...

Test Configuration

✖ Failed to connect or change directory

jenkins.plugins.publish\_over.BapPublisherException: Failed to connect and initialize SSH connection. Message: [Failed to connect session for config [kichul love ]. Message [Auth fail]]

빌드 후 조치

send build artifacts over SSH 추가

설정에서 만들었던 server로 추가

빌드 후 조치

Send build artifacts over SSH ✖ ?

SSH Publishers

SSH Server

Name ?

kichul love

고급...

Transfers

credentials 클릭

username과 key 추가



☒ **Credentials** ?

Username ?

ubuntu

Passphrase / Password ?

Path to key ?

Key ?

```
/PgGzHkKpw36VQ1Mqy/E0MB0UfXvE1asbYvfJ9Im3lsxVNYIKIKnz9vuQ/JAHC5
nQEUNiBc4G8KcyWuSBMfnJeDpNyZ9XiO81s7e0GCN78Bf6dXPlqZ4Z39IW3uAodl
/dKyfmUCqYARJfUMUuGapBdiTlkXyBjZC4PnX4aUOXzOxuazgmAuvAgBjQUYNJ5n
14AufnxC39wCbnPvVM4HcYRQw1S1vZJ/26M8bgCIWxfc5Jj9wv9eS97WDXs5Insi
QqYllt+BZxiQsfo/0Fc+wUTs2aeR81i9JWUjv66KYmaSfazE2pvWVw==
-----END RSA PRIVATE KEY-----
```

**Test Configuration**

jenkins server에 있는 {workspace}/front/build 밑에 있는 모든 파일을

앞에 front 지우고 ⇒ {remote directory}/build

frontend 붙여서 ⇒ {remote directory}/frontend/build

이 폴더로 다 옮김

**Transfers**

Transfer Set

Source files ?

front/build/\*\*/\*

Remove prefix ?

front

Remote directory ?

frontend

Exec command ?

All of the transfer fields (except for Exec timeout) support substitution of **Jenkins environment variables**

**고급...**

transfer set 하나 더 추가해서

백엔드 빌드된 jar 파일을 이동

그 다음 shell script 실행

Transfer Set

Source files ?

back/build/libs/\*

Remove prefix ?

back/build/libs

Remote directory ?

backend

Exec command ?

```
#!/bin/sh
cd ./backend
sudo ls -td1 .jar | tail -n+2 | xargs rm -f
cd ~/da-haeng
sudo mv ./backend/dahaeng-*.jar ./backend/dahaeng-0.0.${BUILD_NUMBER}.jar
sudo docker-compose stop spring
sudo docker build -t dahaeng:latest .
sudo docker-compose up -d spring
```

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

고급...