

포팅 메뉴얼

1. gitlab 소스 클론 이후 빌드 및 배포할 수 있는 작업 문서

1. 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정값, 버전

Sts → 3.9.13

Spring boot → 2.4.8

Java → 1.8

Mysql → 8.0.26 for Linux on x86_64

Android Studio 4.2.2

```
compileSdkVersion 30
buildToolsVersion "30.0.0"
minSdkVersion 24
targetSdkVersion 30

애플레이터 API - 28
Target -> Android 9.0
```

Vue.js → 2.6.14

nginx → 1.21.1

2. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재

프론트엔드

```
sudo docker build . -t frontend-ayj

docker run -d -p 80:80 -it --name frontend-ayj frontend-ayj
```

백엔드

```
sudo yum install maven

mvn -N io.takari:maven:wrapper

./mvnw clean install

docker build -t backend-ayj .

docker run -d -p 8080:8080 --name backend-ayj backend-ayj
```

Mysql

```
docker run -d --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=ssafy mysql --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci
```

3. 배포 시 특이사항 기재

젠킨스 적용

도커를 이용하기 때문에 사전에 Docker, Docker-compose 설치

플러그인 목록

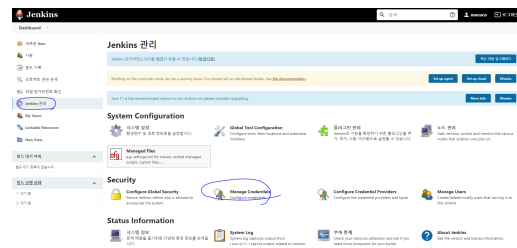
GitLab, NodeJS

젠킨스 내부에서 도커 명령어 사용을 위해 볼륨을 적용하여 젠킨스 컨테이너 실행


```
docker run -d --name my_jenkins -p 9080:8080 -v /home/jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker -v /usr/local/bin/docker-compose:/usr/local/bin/docker-compose -u root jenkins/jenkins
```

깃 계정 설정

Manage Credentials → global





Add Credentials



Jenkins

Dashboard ▾ ▸ Credentials ▸ System ▸ Global credentials (unrestricted) ▸

 Back to credential domains

 Add Credentials

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

ayj8655

☐ Treat username as secret

Password

.....

ID


ayj-test

Description

OK


Username = git 아이디
 Password = git 비밀번호
 ID = 젠킨스에 저장될 키 이름


새로운 Item 생성
 Freestyle project 적용





Jenkins


Dashboard ▾ ▸ Credentials


 새로운 Item


 사람


 빌드 기록


 프로젝트 연관 관계

 파일 핑거프린트 확인

 Jenkins 관리

 My Views

 Lockable Resources

 New View

소스 코드 관리에서 깃 URL 입력 및 Credentials 키 적용

빌드 환경

플러그인에서 Nodejs 설치 및 Global Tool Configuration에 NodeJS 추가

Build

전체 로직

프론트 폴더 이동 및 npm 설치, 빌드

도커 이미지 생성

백엔드 폴더 이동 및 메이븐 빌드

도커 이미지 생성

이전에 실행중이던 도커 컨테이너 정지, 삭제 후 다시 실행

만약 실행중이던 도커가 없다면 stop과 rm을 주석 처리 후 1회 실행

빌드 유발

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: `http://15b203.p.ssafy.io:9080/project/test`
- ☒ 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
- Comment (regex) for triggering a build
 - Jenkins please retry a build
- ☐ GitHub hook trigger for GITScm polling

고급...

Build when a change is pushed to GitLab 선택

빌드 유발 - 고급

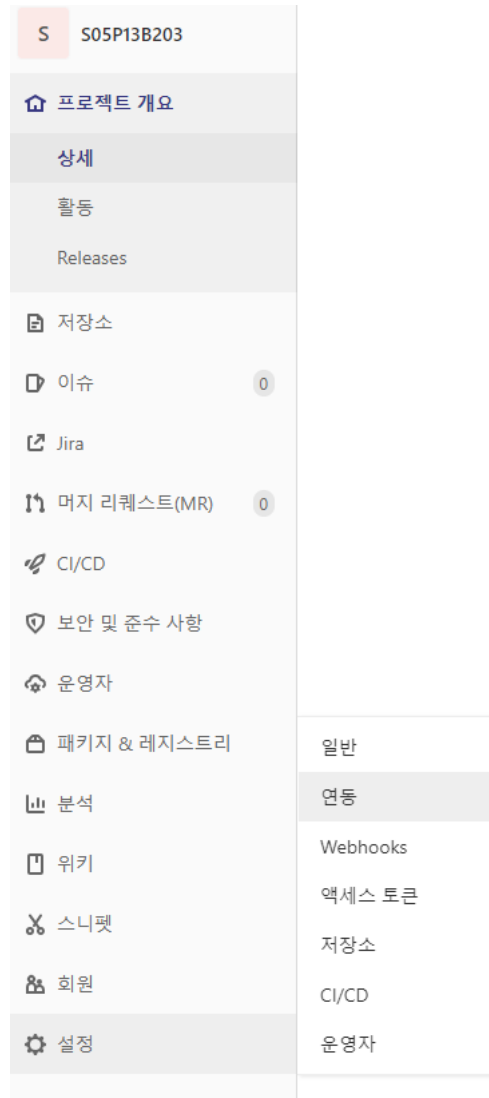
빌드 유발 - 고급

- ☒ 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
 - 682bd1c931c4012c4d28b9f5f049396
- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM
- 빌드 환경
 - ☐ Delete workspace before build starts

저장 Apply

시크릿 토큰과 GitLab webhook URL 저장 후 깃랩으로 이동

GitLab



🔍 Search settings

📘 Webhooks have moved. They can now be found under the Settings menu.

[Go to Webhooks](#)

연동

Webhooks

Webhooks를 사용하면 그룹 또는 프로젝트의 이벤트에 대한 응답으로 웹 애플리케이션에 알림을 보낼 수 있습니다. Webhook보다 통합을 사용하는 것이 좋습니다.

URL

URL must be percent-encoded if necessary.

Secret token

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

Trigger

☒ **Push events**

URL is triggered by a push to the repository

☐ **Tag push events**

URL is triggered when a new tag is pushed to the repository

☐ **Comments**

URL is triggered when someone adds a comment

☐ **Confidential comments**

URL is triggered when someone adds a comment on a confidential issue

☐ **Issues events**

URL is triggered when an issue is created, updated, closed, or reopened

☐ **Confidential issues events**

URL is triggered when a confidential issue is created, updated, closed, or reopened

☐ **Merge request events**

URL is triggered when a merge request is created, updated, or merged

GitLab webhook URL, 시크릿토큰, 푸시 이벤트에 사용할 브랜치 설정

webhook 테스트

☐ **Feature flag events**
URL is triggered when a feature flag is f

☐ **Releases events**
URL is triggered when a release is creat

SSL verification
☒ **Enable SSL verification**

Push events
Tag push events
Issues events
Confidential issues events
Note events
Confidential note events
Merge requests events
Job events
Pipeline events

Project Hooks (2)

http://i5b203.p.ssafy.io:9080/project/mococo_pipeline/	Test ▼	편집	삭제
Push Events SSL Verification: 사용			
http://i5b203.p.ssafy.io:9080/project/ayj	Test ▼	편집	삭제
Push Events SSL Verification: 사용			

HTTP 200이 될 경우 성공

i Hook executed successfully: HTTP 200 ✕

🔍 Search settings

Webhooks

Webhooks를 사용하면 그룹 또는 프로젝트의 이벤트에 대한 응답으로 웹 애플리케이션에 알림을 보낼 수 있습니다. Webhook보다 **통합**을 사용하는 것이 좋습니다.

URL

http://example.com/trigger-ci.json

URL must be percent-encoded if necessary.

Secret token

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

Trigger

☒ **Push events**

Branch name or wildcard pattern to trigger on (leave blank for all)

URL is triggered by a push to the repository

Execute shell Command

```
cd /var/jenkins_home/workspace/ayj/frontend
npm install
npm run build

docker build . -t frontend-ayj

cd /var/jenkins_home/workspace/ayj/backend
npm install maven
#권한추가
chmod +x mvnw
./mvnw clean install

docker build -t backend-ayj .

docker stop frontend-ayj
docker stop backend-ayj

docker rm frontend-ayj
docker rm backend-ayj

docker run -d -p 80:80 -it --name frontend-ayj frontend-ayj
docker run -d -p 8080:8080 --name backend-ayj backend-ayj
```

4. 데이터베이스 접속 정보 등 프로젝트에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

application.properties