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

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

   ec2 version: Ubuntu 20.04 LTS

   nginx - 1.18.0 (Ubuntu)

   gradle - 4.10.2

   Spring boot - 2.1.7

   Java - 1.8

   azul-1.8 version 1.8.0_312

   Mysql - 8.0.25 for Linux on x86_64

   redis - 6.2.6

   react - 17.0.2

   node.js(컴파일 서버): 14.18.1

   node.js(yjs): 14.18.1

2. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재

   환경 변수 값들은 코드에 다 들어있습니다.

   프론트엔드

   ```
   cd /var/jenkins_home/workspace/codeback/front-end
   npm install --global yarn
   yarn
   yarn build
   ```

   백엔드 - spring boot

   ```
   chmod 777 /var/jenkins_home/workspace/codeback/back_springboot/gradlew
   cd /var/jenkins_home/workspace/codeback/back_springboot
   ./gradlew build
   docker build --build-arg DEPENDENCY=build/dependency -t back-end .

   docker run -d -p 8080:8080 --name back-end back-end
   ```

   백엔드 - nodejs (컴파일 서버)

```
cd /var/jenkins_home/workspace/codeback/back_nodejs

docker build . -t back-node-compile

docker stop back-node-compile
docker rm back-node-compile

docker run -d -p 8081:8081 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-node-compile ba
```

백엔드 - nodejs (공동문서편집 서버)

```
cd /var/jenkins_home/workspace/codeback/back_yjs

docker build . -t back-yjs

docker stop back-yjs
docker rm back-yjs

docker run -d -p 8082:8082 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-yjs back-yjs
```

Mysql

```
docker run -d --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=ssafy mysql —character-set-server=utf8mb4 --collation-server=utf8mb
```

Redis

```
docker run --name dingrr -p 6378:6379 --network redis-net -d redis redis-server --appendonly yes

docker run -it --network redis-net --rm redis:alpine redis-cli -h dingrr

# redis-cli로 들어가기
docker exec -it dingrr redis-cli

# redis-cli 안에서 위의 명령어 입력
set slave true
config set slave-read-only no
config set stop-writes-on-bgsave-error no
```

3. 배포 시 특이사항 기재

젠킨스 적용

도커를 이용하기 때문에 사전에 Docker 설치

플러그인 목록

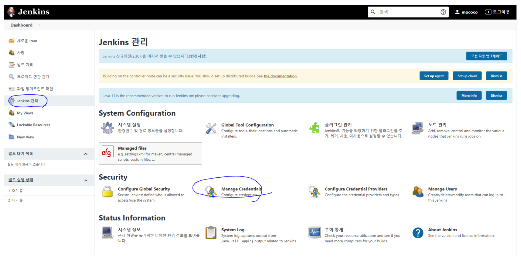
GitLab, NodeJS

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



빌드 대기 항목이 없습니다.

1 대기 중

2 대기 중

Jenkins 관리

Jenkins 신규버전(2.307)을 여기서 받을 수 있습니다.(변경사항).

Building on the controller node can be a security issue. You should set up distributed builds. See the documentation.

Java 11 is the recommended version to run Jenkins on; please consider upgrading.

System Configuration

시스템 설정 — 환경변수 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

Status Information

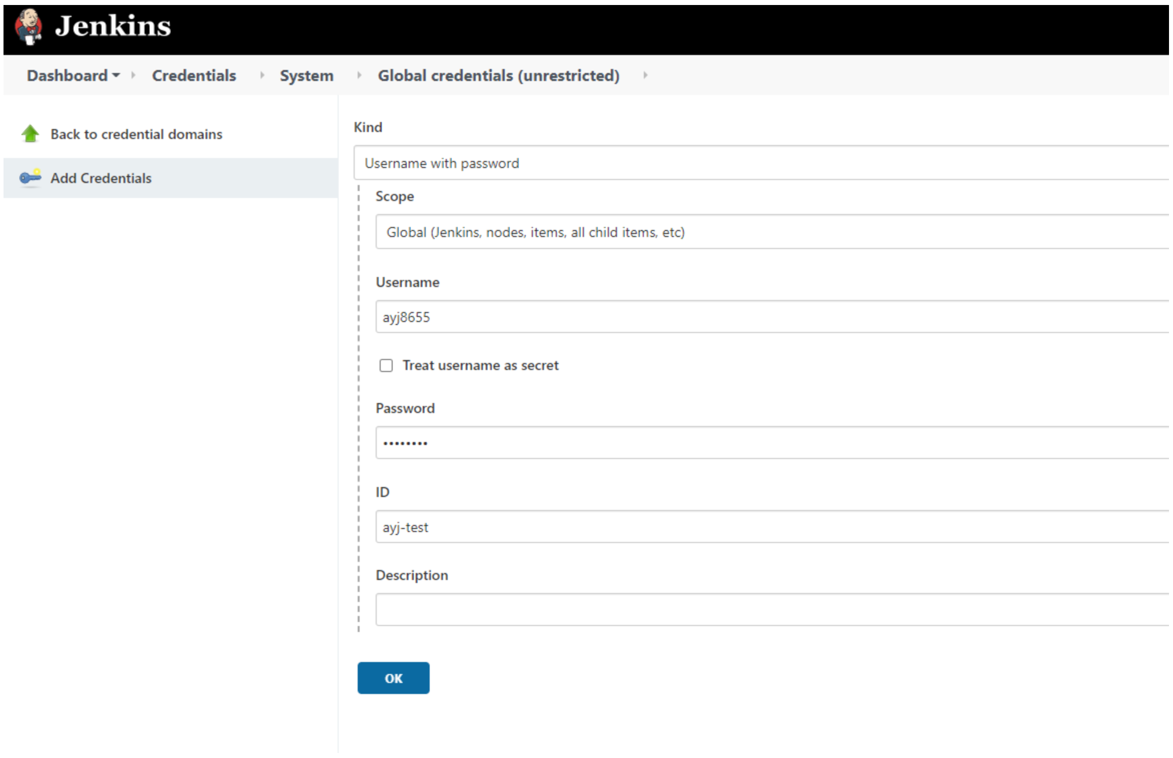
시스템 정보 — 문제 해결을 돕기위한 다양한 환경 정보를 보여줍니다.

System Log — System log captures output from java.util.logging output related to Jenkins.

부하 통계 — Check your resource utilization and see if you need more computers for your builds.

About Jenkins — See the version and license information.

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:

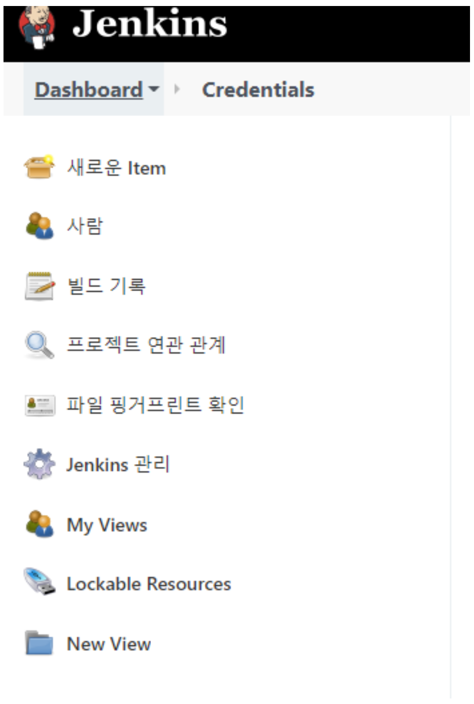
Username = git 아이디

Password = git 비밀번호

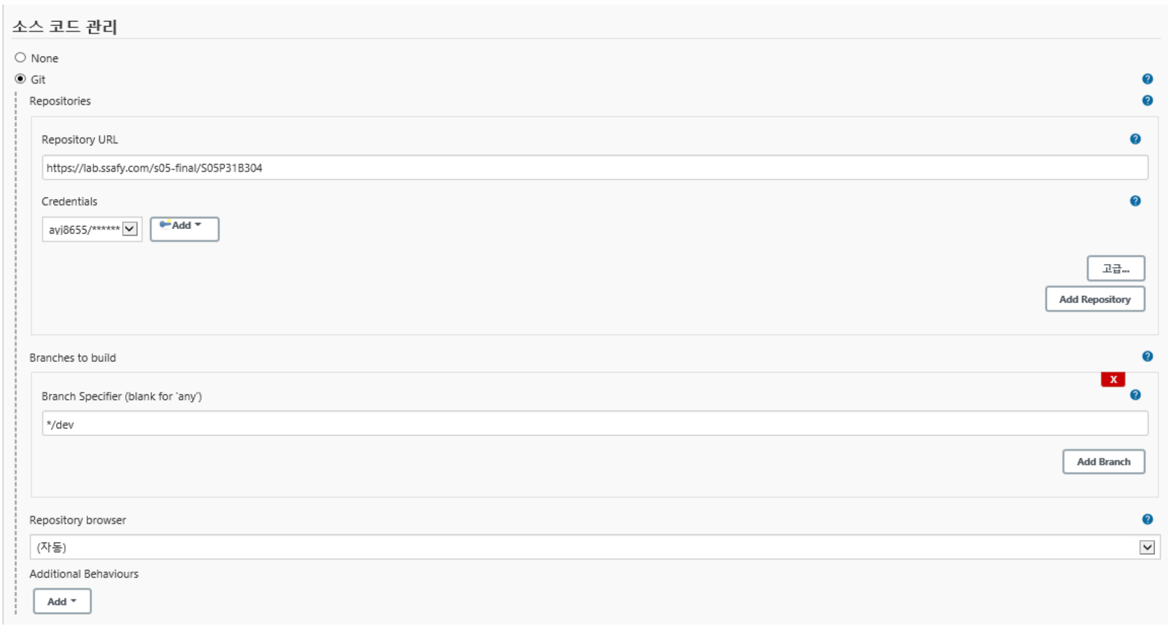
ID = 젠킨스에 저장될 키 이름

새로운 Item 생성

Freestyle project 적용



소스 코드 관리에서 깃 URL 입력 및 Credentials 키 적용


소스 코드 관리

None [✓ Git]

Repositories

Repository URL: https://lab.ssafy.com/s05-final/S05P31B304

Credentials: ayj8655/******

Branches to build

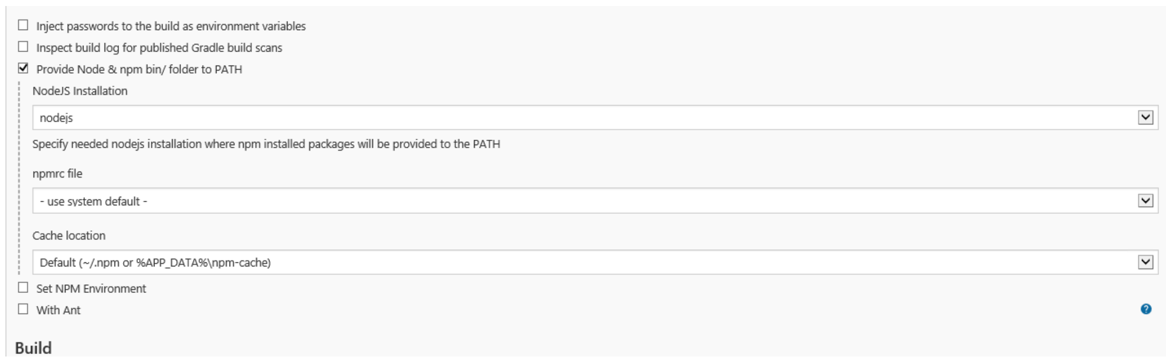
Branch Specifier (blank for 'any'): */dev

Repository browser: (자동)

Additional Behaviours

빌드 환경

플러그인에서 Nodejs 설치 및 Global Tool Configuration에 NodeJS 추가



☐ Inject passwords to the build as environment variables
☐ Inspect build log for published Gradle build scans
☑ Provide Node & npm bin/ folder to PATH

NodeJS Installation
nodejs
Specify needed nodejs installation where npm installed packages will be provided to the PATH

npmrc file
- use system default -

Cache location
Default (~/.npm or %APP_DATA%\npm-cache)

☐ Set NPM Environment
☐ With Ant

Build

Build

전체 로직

프론트 폴더 이동 및 yarn 설치 후 빌드

docker 외부의 nginx와 연결

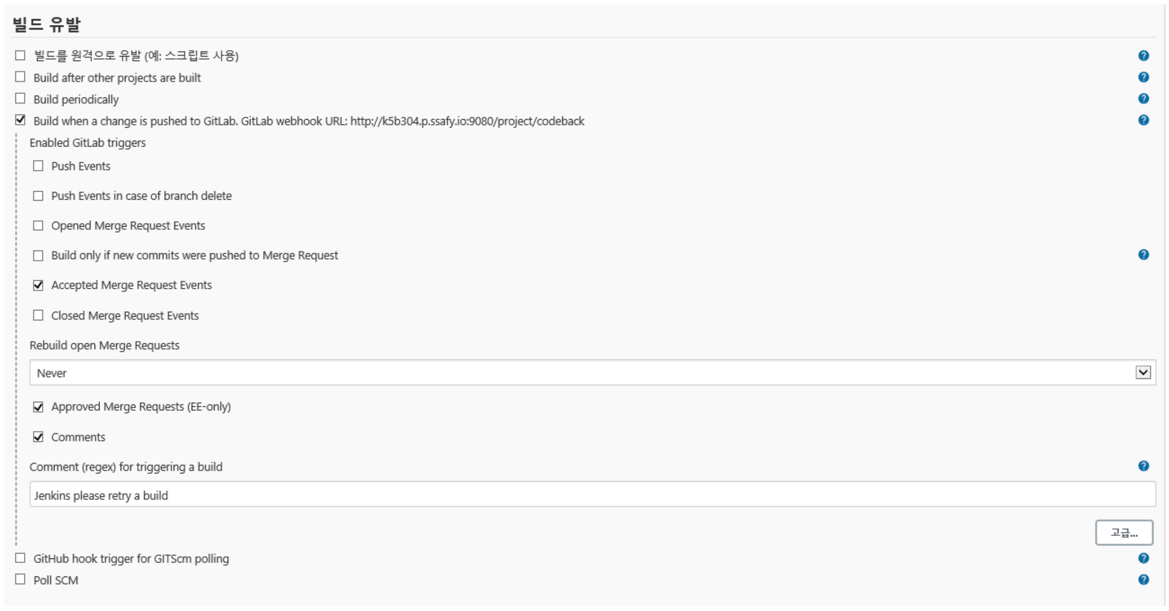
백엔드 폴더 이동 및 gradle 빌드

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://k5b304.p.ssafy.io:9080/project/codeback

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

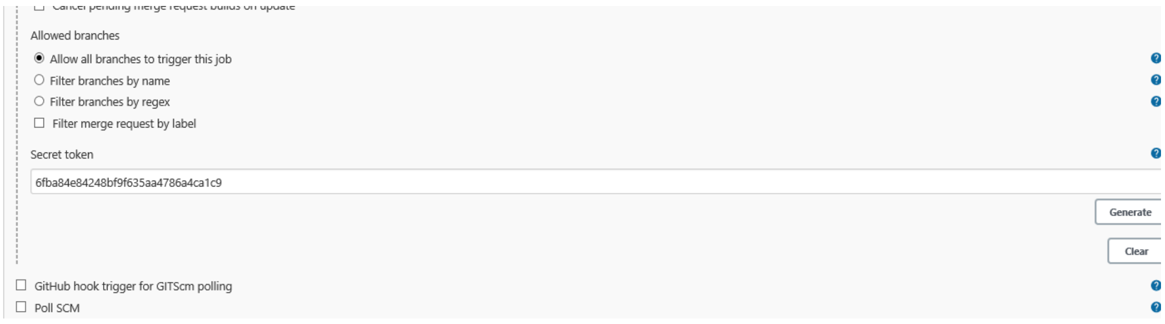
고급...

☐ GitHub hook trigger for GITScm polling
☐ Poll SCM

Build when a change is pushed to GitLab 선택

고급 선택 후 아래 처럼 입력

빌드 유발 - 고급


☐ 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

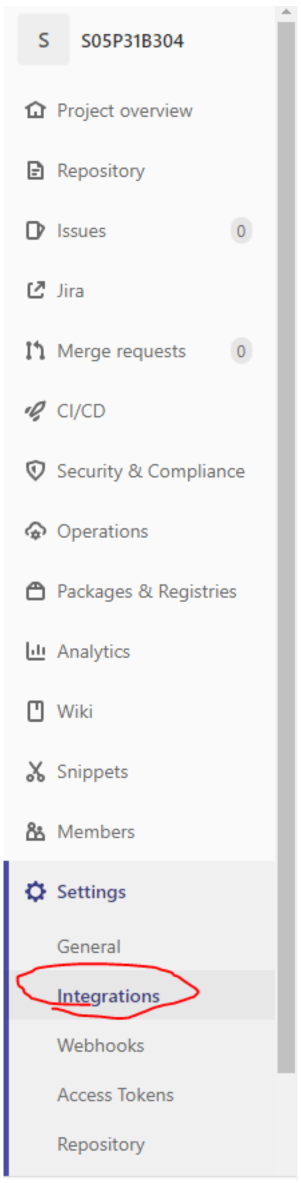
6fba84e84248bf9f635aa4786a4ca1c9

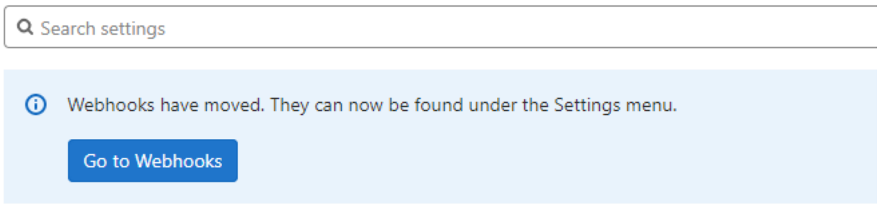
☐ GitHub hook trigger for GITScm polling

☐ Poll SCM

시크릿 토큰과 GitLab webhook URL 저장 후 깃랩으로 이동

GitLab

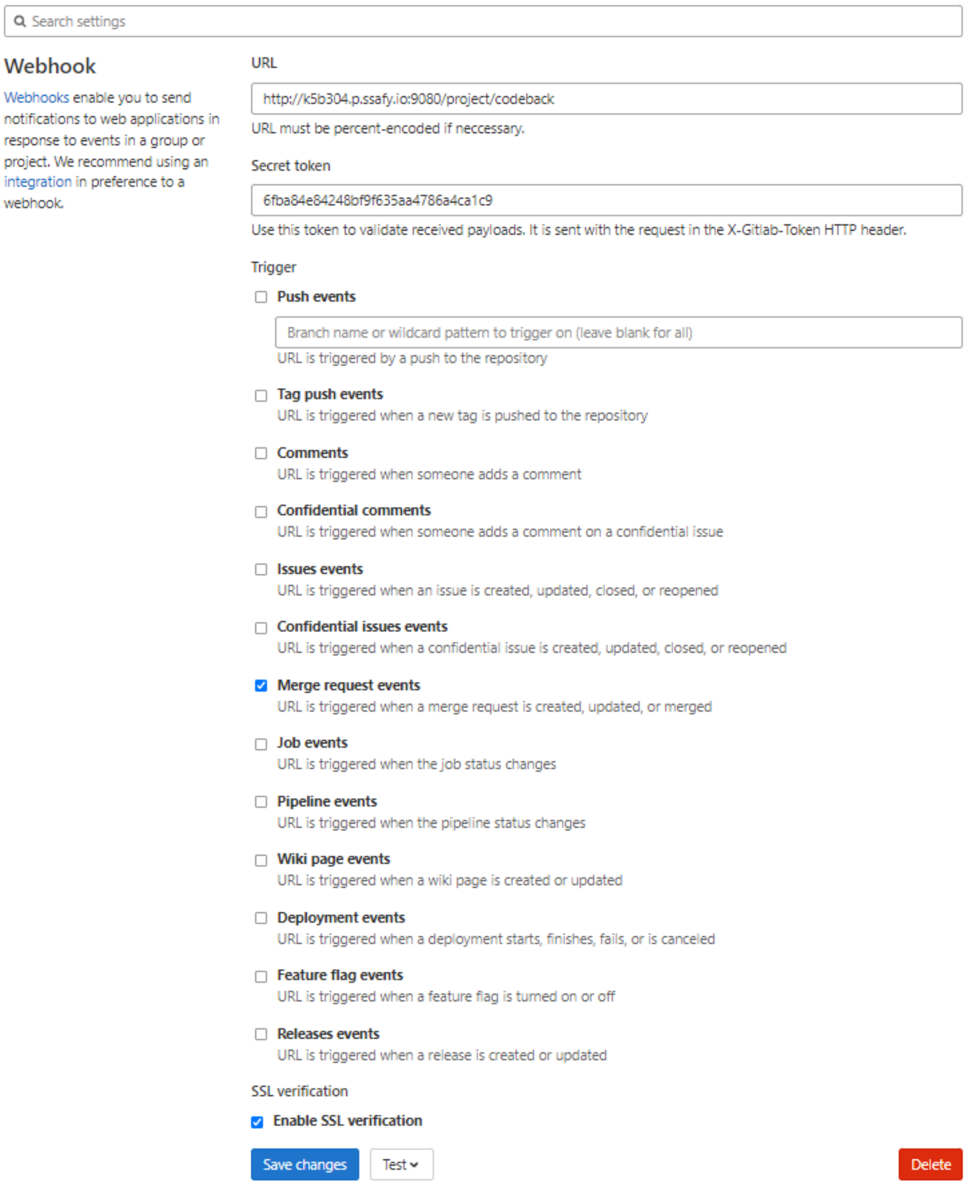



Search settings

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

연동

GitLab webhook URL, 시크릿토큰, 푸시 이벤트에 사용할 브랜치 설정


Search settings

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://k5b304.p.ssafy.io:9080/project/codeback

URL must be percent-encoded if neccessary.

Secret token

6fba84e84248bf9f635aa4786a4ca1c9

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

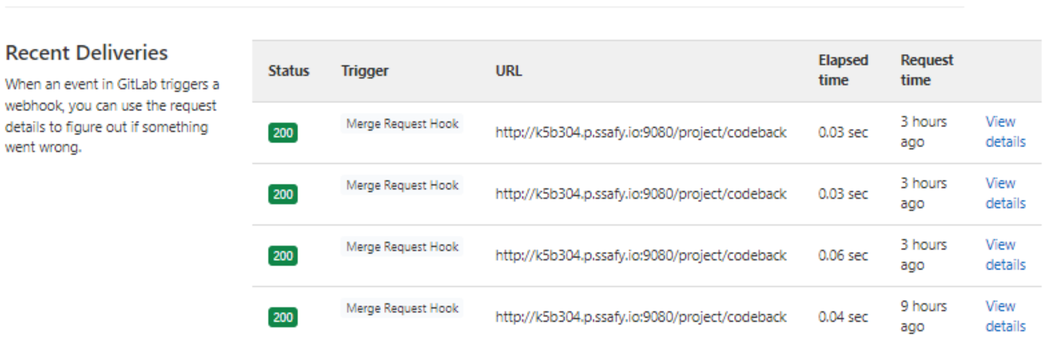
Trigger

- [ ] Push events
  Branch name or wildcard pattern to trigger on (leave blank for all)
  URL is triggered by a push to the repository
- [ ] 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
- [x] Merge request events
  URL is triggered when a merge request is created, updated, or merged
- [ ] Job events
  URL is triggered when the job status changes
- [ ] Pipeline events
  URL is triggered when the pipeline status changes
- [ ] Wiki page events
  URL is triggered when a wiki page is created or updated
- [ ] Deployment events
  URL is triggered when a deployment starts, finishes, fails, or is canceled
- [ ] Feature flag events
  URL is triggered when a feature flag is turned on or off
- [ ] Releases events
  URL is triggered when a release is created or updated

SSL verification

- [x] Enable SSL verification

webhook 테스트

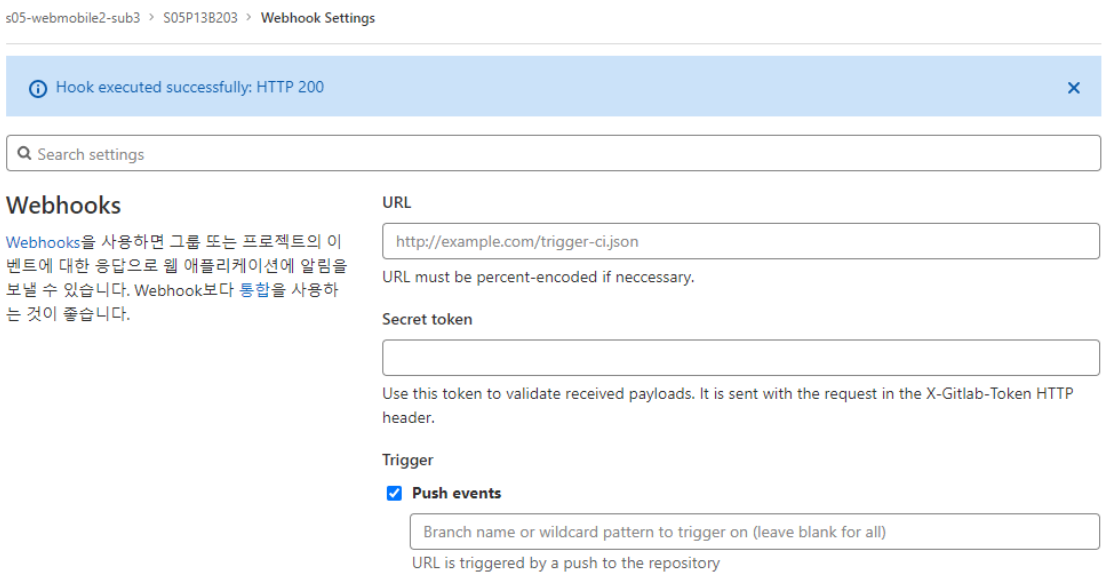


Recent Deliveries

When an event in GitLab triggers a webhook, you can use the request details to figure out if something went wrong.

Status	Trigger	URL	Elapsed time	Request time	
200	Merge Request Hook	http://k5b304.p.ssafy.io:9080/project/codeback	0.03 sec	3 hours ago	View details
200	Merge Request Hook	http://k5b304.p.ssafy.io:9080/project/codeback	0.03 sec	3 hours ago	View details
200	Merge Request Hook	http://k5b304.p.ssafy.io:9080/project/codeback	0.06 sec	3 hours ago	View details
200	Merge Request Hook	http://k5b304.p.ssafy.io:9080/project/codeback	0.04 sec	9 hours ago	View details

HTTP 200이 뜰 경우 성공



s05-webmobile2-sub3 › S05P13B203 › **Webhook Settings**

Hook executed successfully: HTTP 200

Search settings

Webhooks

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

URL

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

URL must be percent-encoded if neccessary.

Secret token

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

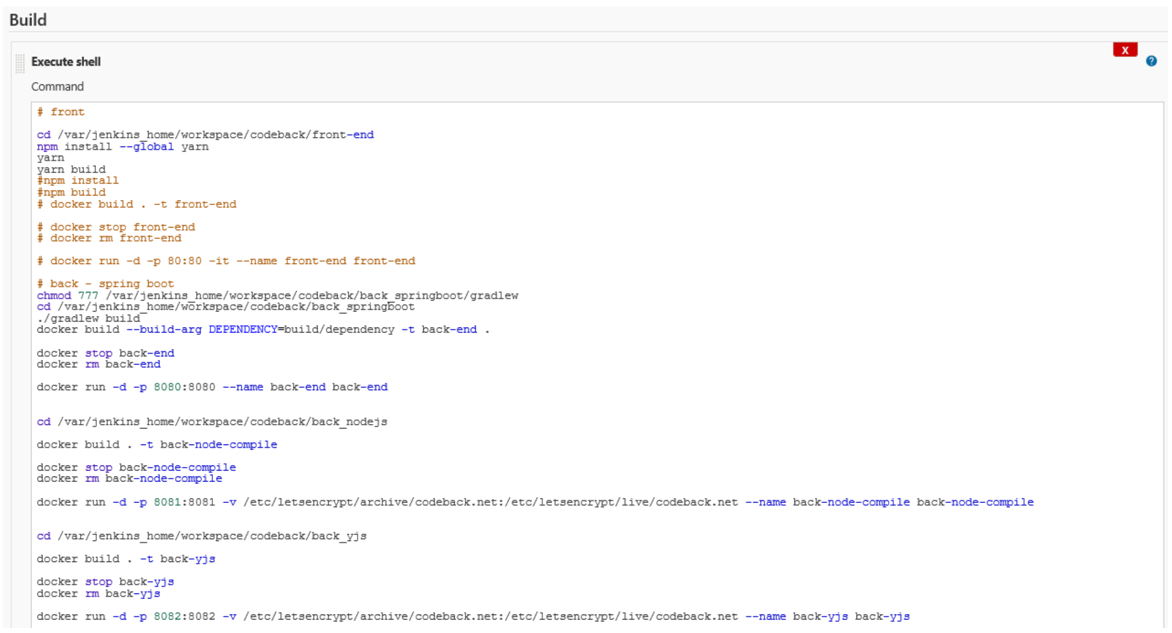
Trigger

[✓] **Push events**

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

URL is triggered by a push to the repository

Jenkins build script



Build

Execute shell

Command

```
# front

cd /var/jenkins_home/workspace/codeback/front-end
npm install --global yarn
yarn
yarn build
#npm install
#npm build
# docker build . -t front-end

# docker stop front-end
# docker rm front-end

# docker run -d -p 80:80 -it --name front-end front-end

# back - spring boot
chmod 777 /var/jenkins_home/workspace/codeback/back_springboot/gradlew
cd /var/jenkins_home/workspace/codeback/back_springboot
./gradlew build
docker build --build-arg DEPENDENCY=build/dependency -t back-end .

docker stop back-end
docker rm back-end

docker run -d -p 8080:8080 --name back-end back-end


cd /var/jenkins_home/workspace/codeback/back_nodejs

docker build . -t back-node-compile

docker stop back-node-compile
docker rm back-node-compile

docker run -d -p 8081:8081 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-node-compile back-node-compile


cd /var/jenkins_home/workspace/codeback/back_yjs

docker build . -t back-yjs

docker stop back-yjs
docker rm back-yjs

docker run -d -p 8082:8082 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-yjs back-yjs
```

Execute shell Command

```
# front

cd /var/jenkins_home/workspace/codeback/front-end
npm install --global yarn
yarn
yarn build
#npm install
#npm build
# docker build . -t front-end

# docker stop front-end
# docker rm front-end

# docker run -d -p 80:80 -it --name front-end front-end

# back - spring boot
chmod 777 /var/jenkins_home/workspace/codeback/back_springboot/gradlew
cd /var/jenkins_home/workspace/codeback/back_springboot
./gradlew build
docker build --build-arg DEPENDENCY=build/dependency -t back-end .

docker stop back-end
docker rm back-end

docker run -d -p 8080:8080 --name back-end back-end


cd /var/jenkins_home/workspace/codeback/back_nodejs

docker build . -t back-node-compile

docker stop back-node-compile
docker rm back-node-compile

docker run -d -p 8081:8081 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-node-compile ba


cd /var/jenkins_home/workspace/codeback/back_yjs

docker build . -t back-yjs

docker stop back-yjs
docker rm back-yjs

docker run -d -p 8082:8082 -v /etc/letsencrypt/archive/codeback.net:/etc/letsencrypt/live/codeback.net --name back-yjs back-yjs
```

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

MYSQL

ID: root

Password: ssafy

backend spring boot

application.properties

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://k5b304.p.ssafy.io:3306/codeback?useUniCode=yes&characterEncoding=UTF-8&serverTimezone=Asia/Seou
spring.datasource.username=root
spring.datasource.password=ssafy


# jpa
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.use_sql_comments=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update

# jwt
tokenSecret= c2lsdmVybmluZS10ZWNoLXNwcmluZy1ib290LWp3dC10dXRvcmlhbC1zZWNyZXQtc2lsdmVybmluZS10ZWNoLXNwcmluZy1ib290LWp3dC10dXRvcmlhb
```

```
tokenExpirationMsec = 5000000
refreshTokenExpirationMsec = 500000000

# COOKIE
accessTokenCookieName = accessToken
refreshTokenCookieName = refreshToken
emailCookieName = emailCookie

#jwt
jwt.header = Authorization
jwt.secret = c2lsdmVybmluZS10ZWNoLXNwcmluZy1ib290LWp3dC10dXRvcmlhbC1zZWNyZXQtc2lsdmVybmluZS10ZWNoLXNwcmluZy1ib290LWp3dC10dXRvcmlhb
jwt.token-validity-in-seconds = 86400

# SMTP
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=leeyongjig7679@gmail.com
spring.mail.password=xobjkfiiwohjiwle
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
signUpCookieName = signUpCookie

#redis
spring.redis.database=0
spring.redis.host=k5b304.p.ssafy.io
spring.redis.port=6378
spring.redis.timeout=60000


#https
server.ssl.key-store=classpath:keystore.p12
server.ssl.key-store-password=123123

server.port=8080
```

front end (front-end 폴더 하위에 존재)

.env.development.local

```
REACT_APP_API_DOMAIN_URL=https://codeback.net:8080
REACT_APP_OPENVIDU_URL=https://codeback.net:8443
REACT_APP_OPENVIDU_KEY=CODEBACK
REACT_APP_COMPILE_SOCKET_URL=https://codeback.net:8081
REACT_APP_EDITOR_SOCKET_URL=wss://codeback.net:8082
```

.env.production

```
REACT_APP_API_DOMAIN_URL=https://codeback.net:8080
REACT_APP_OPENVIDU_URL=https://codeback.net:8443
REACT_APP_OPENVIDU_KEY=CODEBACK
REACT_APP_COMPILE_SOCKET_URL=https://codeback.net:8081
REACT_APP_EDITOR_SOCKET_URL=wss://codeback.net:8082
```

backend nodejs (back_nodejs폴더 하위에 존재)
process.env 파일

```


JDOODLE_ID = 272acc2fecb4661ea47e443f8ea3d8a2
JDOODLE_SECRET = e5904b4f5f20ea37e968153ca239b982bb00fb73c5100b8fa92ed810013b7fef

SECRET_PATH = /etc/letsencrypt/live/codeback.net
```

backend yjs (back_yjs폴더 하위에 존재)
process.env 파일

```
SECRET_PATH = /etc/letsencrypt/live/codeback.net
```