포팅 메뉴얼

목차

---

1. 개발환경

1-1 프론트엔드

- VsCode : 1.86.1
- Node.js : 20.11.0
- Vue : 3.3.11

1-2 백엔드

- IntelliJ : 2023.3.2
- Spring Boot : 3.2.2
- JDK : OpenJDK17
- MySQL : 8.0.22
- Redis : 7.2.4

1-3 서버 및 인프라

- Server : Ubuntu 20.04.6 LTS
- OpenVidu : 2.29.0
- Jenkins : 2.426.2

2. 설정파일 및 환경 변수 정보

Spring

application.yml

```
spring:
  #메일 인증을 위한 설정
  mail:
    host: ENC(viy9Ztw5f5J212yEvuLOfI/AtepmAaGK) #gmail 서비스
    port: 587
    username: ENC(80aS6n6xT6g/f/ZnpAfGDE8jHTm+Qyb3ZAHefBG4fes
    password: ENC(W9n+wP/FM0oGHZWPqEflaYNIaInbEquF1GCxU38uwlY
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true
            required: true
          connectiontimeout: 5000
          timeout: 5000
          writetimeout: 5000
    auth-code-expiration-millis: 1800000  # 30 * 60 * 1000 ==

  #레디스 설정 정보
  redis:
    host: ENC(5lc3y/8+rbT2qZUwQAJmB2uMQ/2njthp6miNwxuZBHY=) #
    port: 6379 #레디스 컨테이너 포트

  #하이버네이트 설정
  h2:
    console:
      enabled: true

  #Mysql 설정
  datasource:
    url: ENC(hf8hLOPLxNmFLgEX05LWevL2cHUP3jtN8T5Hs20HCbepzBc1
    driver-class-name: com.mysql.cj.jdbc.Driver
    username: ENC(fVvNH1SNPMpW9Q29/p13/g==)
    password: ENC(EeJHR6Zf+2w8xuN45J2hRw==)

  #jpa 설정
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true

  #jwt 설정
  jwt:
    secret: ENC(x+0ILox+LZ+MIU6Xqo6ziYi6kL0+9aqJBhr+nZgQVjAZg

#swagger 로그인 엔드포인트 생성
springdoc:
  show-login-endpoint: true

#openvidu 설정
OPENVIDU_URL: https://i10d205.p.ssafy.io:8443
OPENVIDU_SECRET: ENC(VTaBF034COcbXIMho1LVUK+Q51Iy7Swy)



server:
  ssl:
    enabled: false

  #서버 앤드포인트 설정
  servlet:
    context-path: /api/v1

  port: 8080

#jpa 쿼리 로그 설정
logging:
  level:
    org.hibernate.SQL: debug
    org.hibernate.type: trace
```

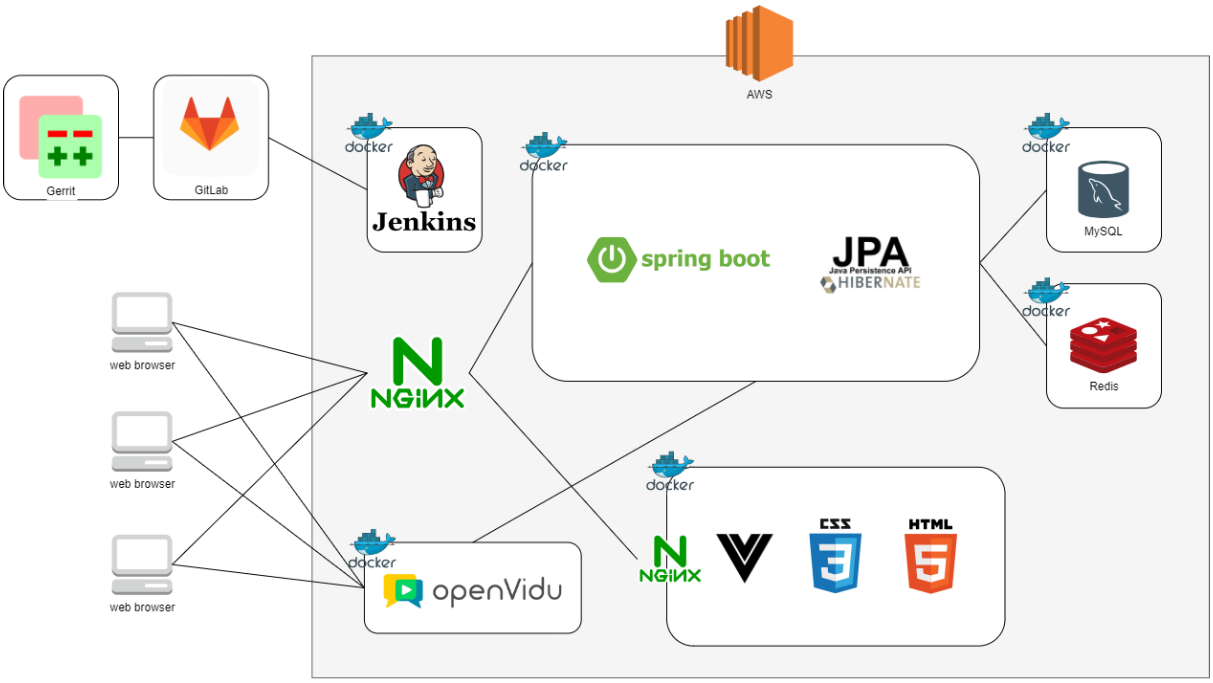
Vue

.env

```
// 서버 REST-API 접속 주소
VITE_VUE_API_URL=https://i10d205.p.ssafy.io/api/v1
// 서버 웹소켓 접속 주소
VITE_WSS_API_URL=wss://i10d205.p.ssafy.io/api/v1/connect
```

3. AWS 서버 설정

3-1 프로젝트 구조



3-2 Docker 설치

1. **apt 업데이트**

   `apt-get update`

2. **도커 설치**

   `apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin`

3. **도커 설치 확인**

   `docker run hello-world`

4. **도커 버전 확인**

   `docker -v`

5. **도커 데몬 실행**

   `systemctl start docker`

3-3 Jenkins 설치 및 설정

1. **jenkins container 생성 및 구동**

```
cd /home/ubuntu && mkdir jenkins-data
```

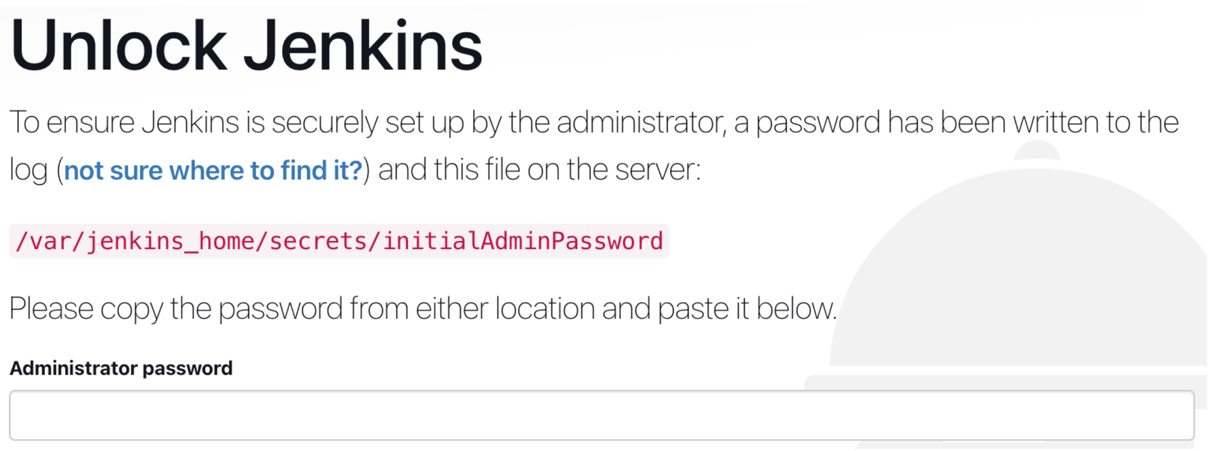
```
sudo ufw allow *8080*/tcp
sudo ufw reload
sudo ufw status

sudo docker run -d -p 8080:8080 -v /home/ubuntu/jenkins-dat
a:/var/jenkins_home --name jenkins jenkins/jenkins:lts

sudo docker logs jenkins

sudo docker stop jenkins
sudo docker ps -a
```

2. **환경 변수 설정**

```
cd /home/ubuntu/jenkins-data

mkdir update-center-rootCAs

wget https://cdn.jsdelivr.net/gh/lework/jenkins-update-center

sudo sed -i 's#https://updates.jenkins.io/update-center.json#

sudo docker restart jenkins
```

3. **젠킨스 접속**

   http://i10d205.p.ssafy.io:8080/



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

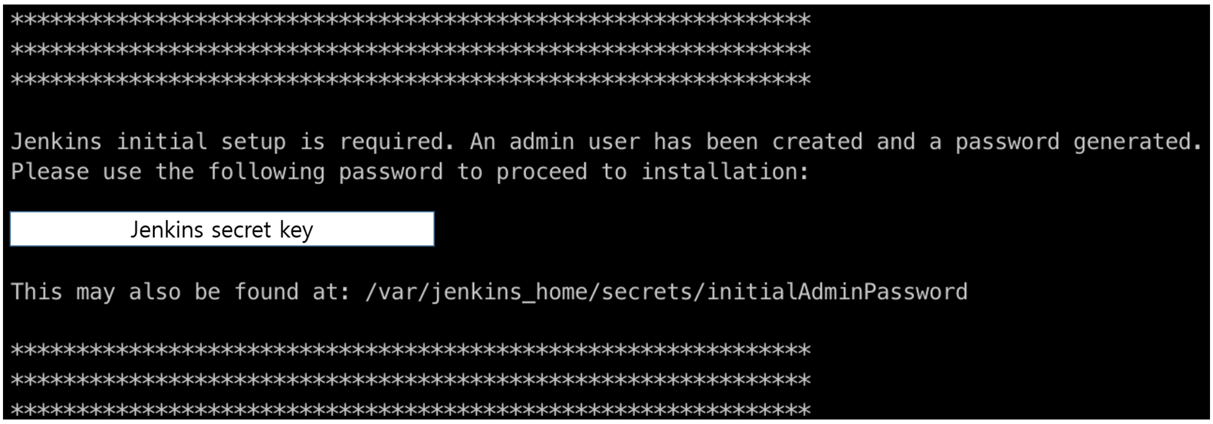
`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

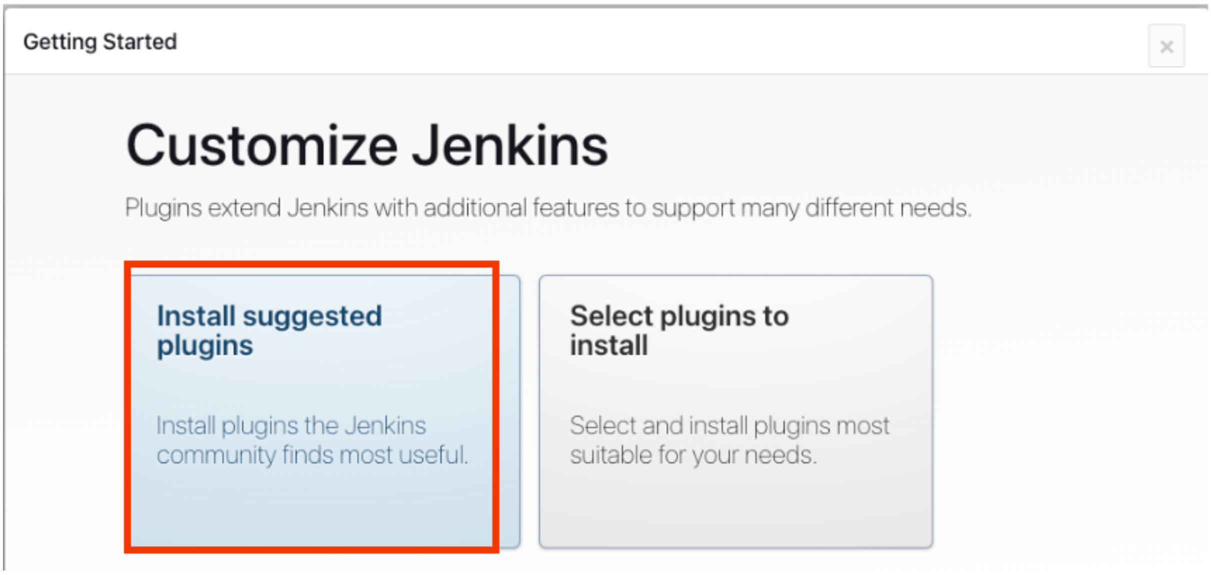
4. 젠킨스 접속 키 확인

```
sudo docker logs jenkins
```



```
*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

Jenkins secret key

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

5. 필수 플러그인 설치


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

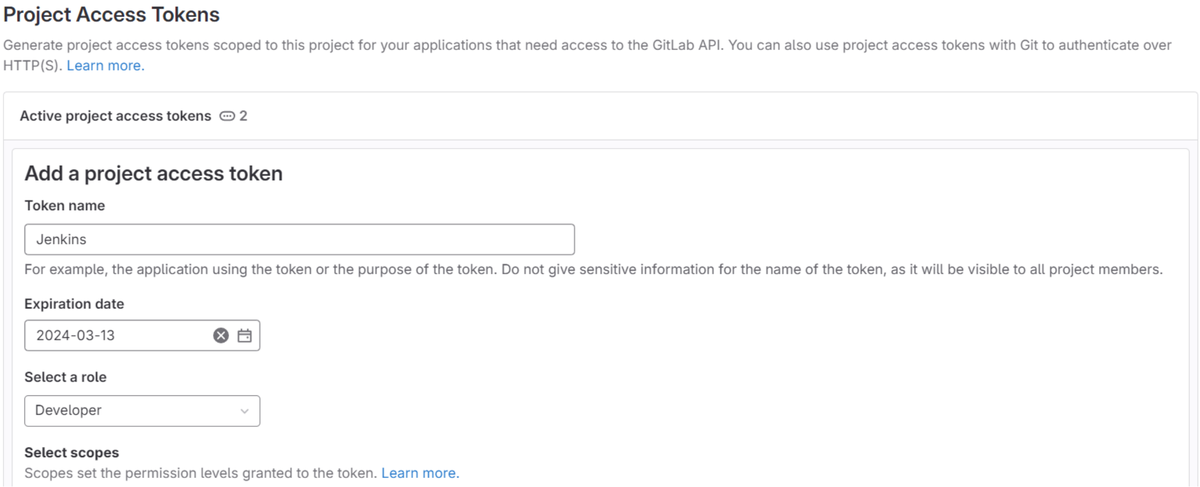
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

6. GitLab 연동 설정

깃랩 api 토큰 생성


Project Access Tokens

Generate project access tokens scoped to this project for your applications that need access to the GitLab API. You can also use project access tokens with Git to authenticate over HTTP(S). Learn more.

Active project access tokens 2

Add a project access token

Token name

Jenkins

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

Expiration date

2024-03-13

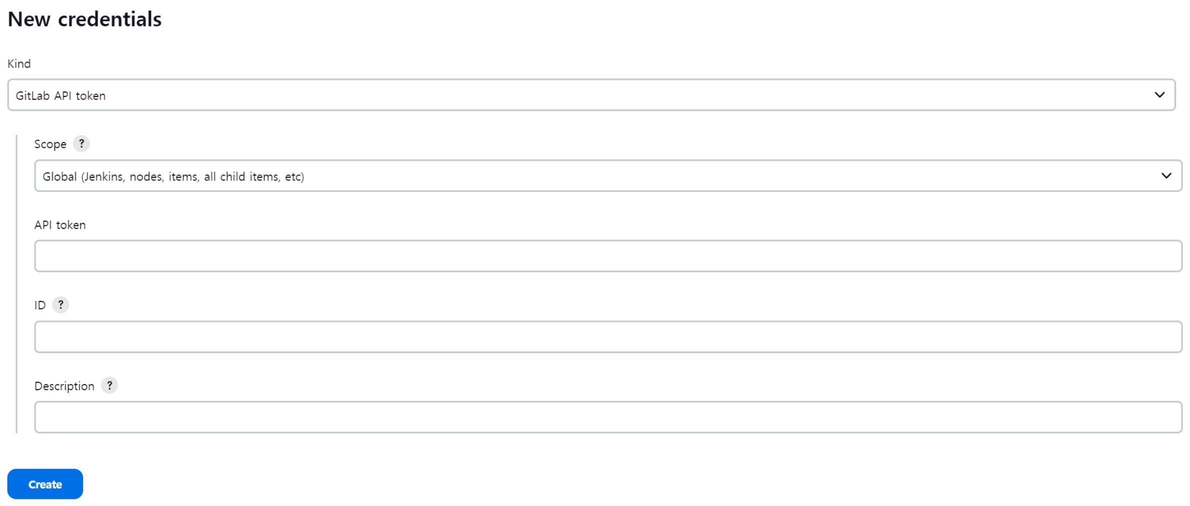
Select a role

Developer

Select scopes

Scopes set the permission levels granted to the token. Learn more.

Jenkins credential 추가



New credentials

Kind

GitLab API token

Scope ?

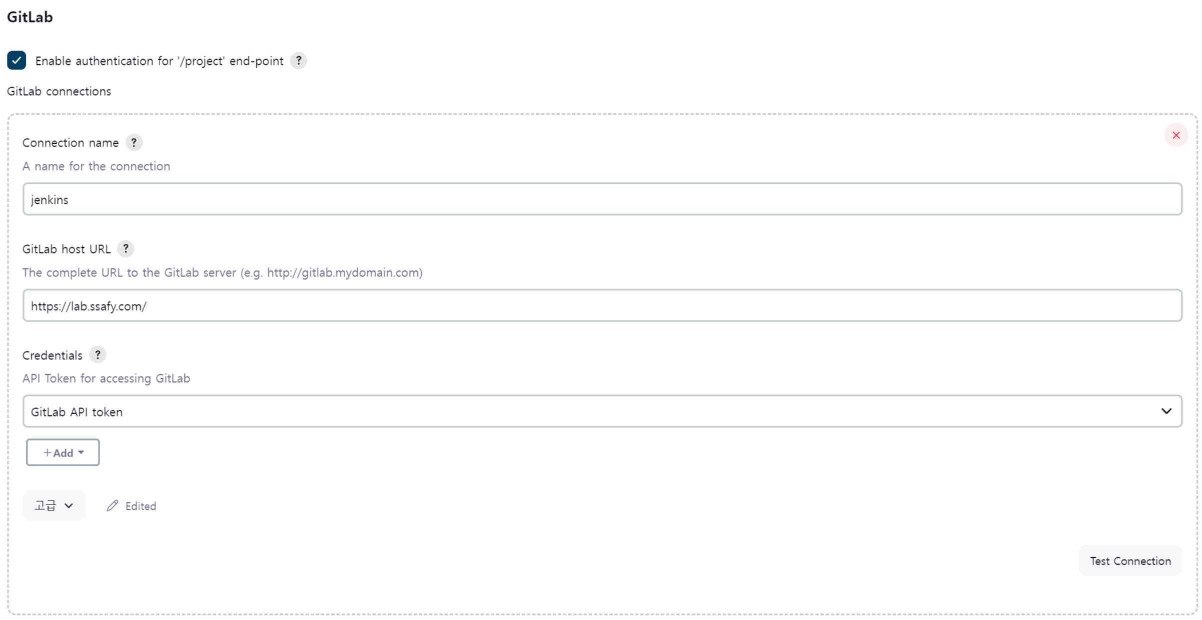
Global (Jenkins, nodes, items, all child items, etc)

API token

ID ?

Description ?

system 설정 깃랩 설정



GitLab

☑ Enable authentication for '/project' end-point ?

GitLab connections

Connection name ?

A name for the connection

jenkins

GitLab host URL ?

The complete URL to the GitLab server (e.g. http://gitlab.mydomain.com)

https://lab.ssafy.com/

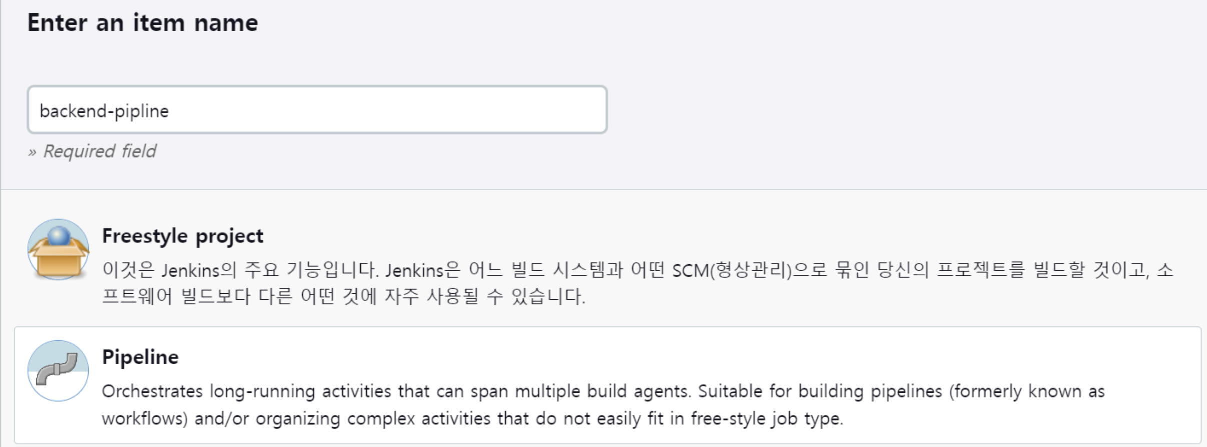
Credentials ?

API Token for accessing GitLab

GitLab API token

7. 백앤드 CI/CD 파이프라인

새로운 Item을 Pipeline으로 생성



Enter an item name

backend-pipline

» Required field

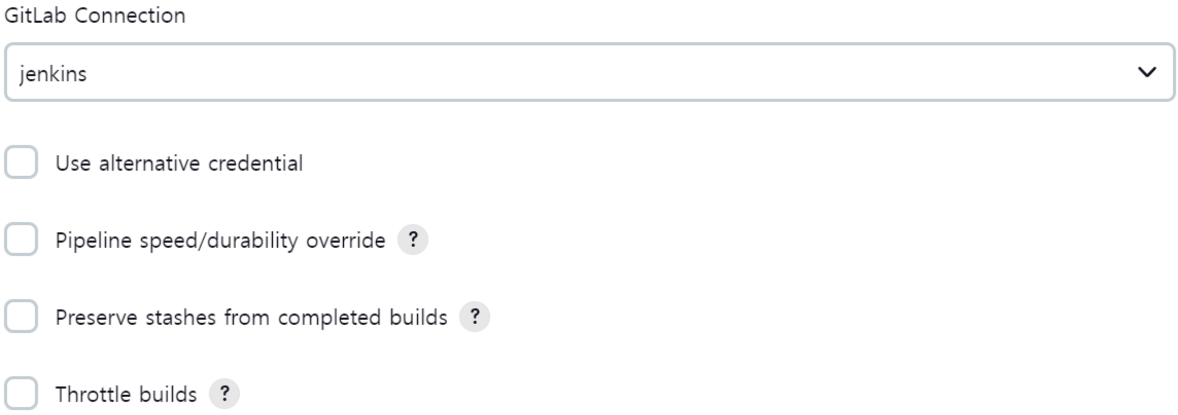
Freestyle project

이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

미리 만들어둔 Gitlab 연동 할당



GitLab Connection

jenkins

□ Use alternative credential

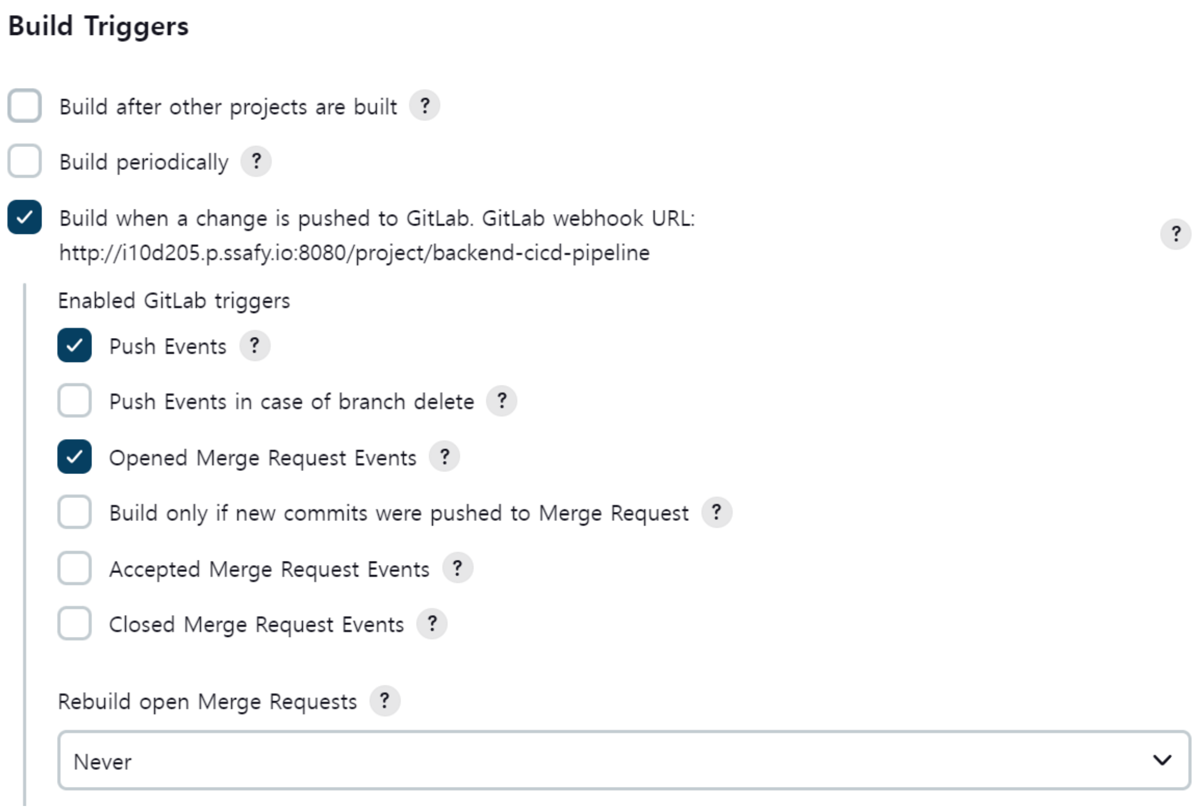
□ Pipeline speed/durability override ?

□ Preserve stashes from completed builds ?

□ Throttle builds ?

Build Triggers 설정

- 백엔드, 프론트엔드 브런치에 푸쉬 이벤트 발생 시 빌드 유발하도록 설정



Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☑ Build when a change is pushed to GitLab. GitLab webhook URL: http://i10d205.p.ssafy.io:8080/project/backend-cicd-pipeline ?

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 ?

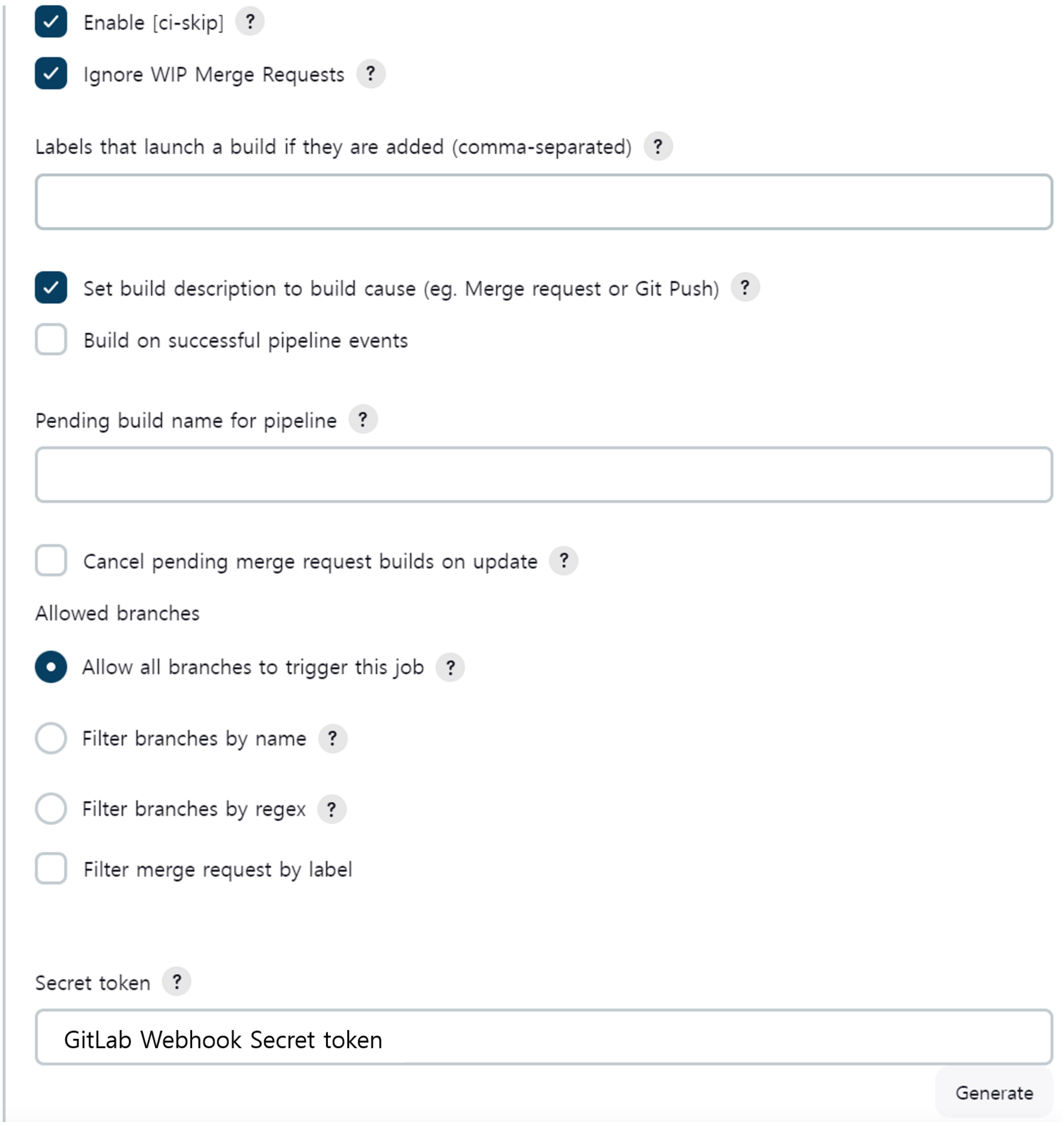
☐ Accepted Merge Request Events ?

☐ Closed Merge Request Events ?

Rebuild open Merge Requests ?

Never

GitLab 웹훅 설정



☑ Enable [ci-skip] ?

☑ Ignore WIP Merge Requests ?

Labels that launch a build 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 ?

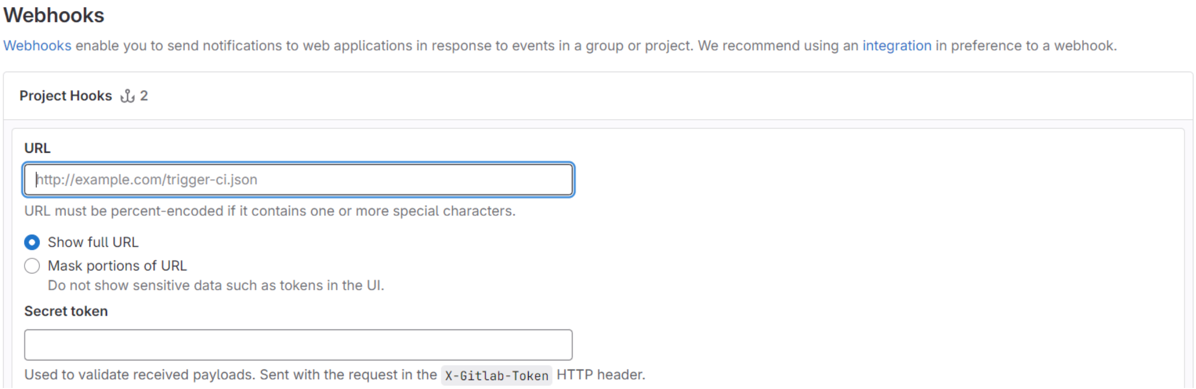
○ Filter branches by regex ?

□ Filter merge request by label

Secret token ?

GitLab Webhook Secret token

GitLab 웹훅 추가



Webhooks

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.

Project Hooks ⚓ 2

URL

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

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

● Show full URL

○ Mask portions of URL
Do not show sensitive data such as tokens in the UI.

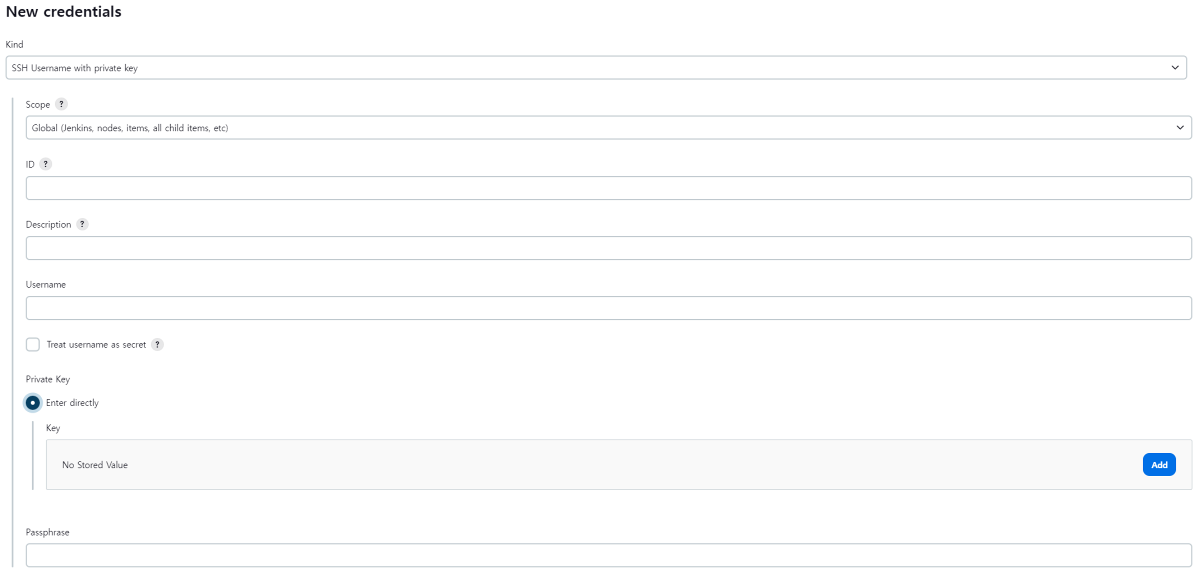
Secret token

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

8. **SSH 설정**

Jenkins Pipeline 에서 AWS 서버에 빌드 파일 전송 및 실행을 위한 SSH 설정

- Jenkins Plugin 에서 SSH Agent Plugin 설치
- Credentials 추가



New credentials

Kind

SSH Username with private key

Scope ?

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

ID ?

Description ?

Username

□ Treat username as secret ?

Private Key

● Enter directly

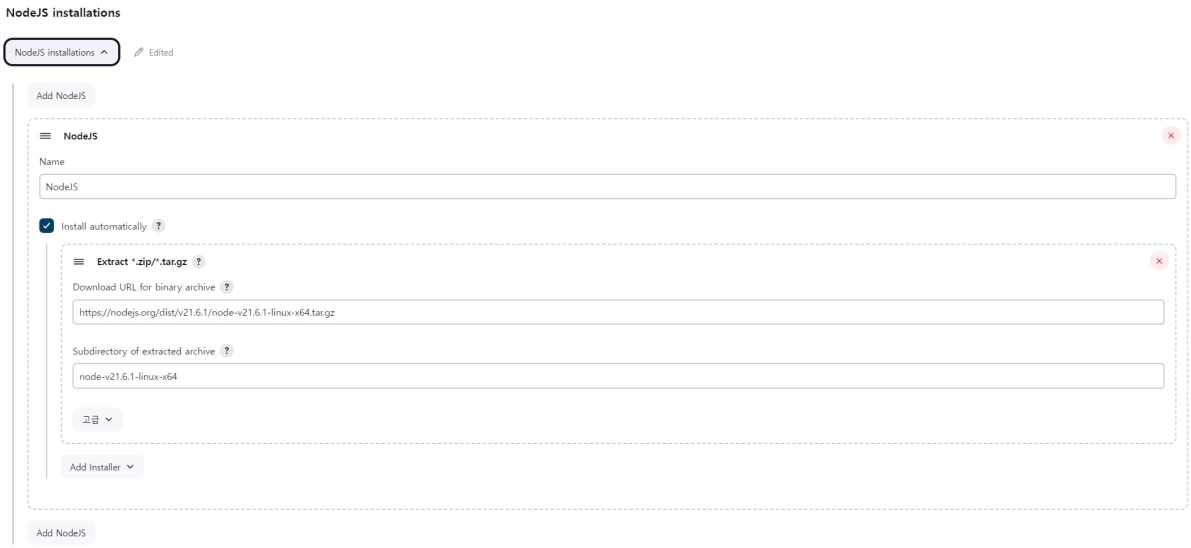
Key

No Stored Value

Passphrase

- key 에 SSH pem 파일 정보 복사 후 붙여넣기

9. **NodeJs 설정**



NodeJS installations

NodeJS

Name: NodeJS

☑ Install automatically ?

Extract *.zip/*.tar.gz ?

Download URL for binary archive ?: https://nodejs.org/dist/v21.6.1/node-v21.6.1-linux-x64.tar.gz

Subdirectory of extracted archive ?: node-v21.6.1-linux-x64

10. 백엔드 파이프라인 스크립트

```
pipeline {
    agent any
    tools {
        gradle 'gradle'
    }
    stages {
        stage('clone') {
            steps {
                git branch: 'develop-BE', credentialsId : 'je
            }
        }
        stage('BE-Build') {
            steps {
                dir("./BE") {
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deploy') {
           steps {
               dir('BE/build/libs') {
                   sh "chmod 777 ./*"
                   sshagent(credentials: ['pem']) {
                        sh 'ssh -o StrictHostKeyChecking=no u
                                                #빌드파일을 서버
                        sh 'scp BE-0.0.1-SNAPSHOT.jar ubuntu@
                                                #서버에서 도커
                        sh 'scp ../../Dockerfile ubuntu@i10d2
                                                #컨테이너 생성을
                        sh 'ssh ubuntu@i10d205.p.ssafy.io "sh
                   }
               }
           }
       }
    }
}
```

11. **백엔드 DockerFile**

```
FROM openjdk:17-alpine
EXPOSE 5000
ARG JAR_FILE=BE-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

12. **백엔드 run.sh**

```
echo "start build image"
echo "=======================================================
sudo docker build -t be/app . #이미지 빌드
echo "=======================================================
echo "complete built"
echo "=======================================================
echo "is any same name container running?" #같은 이름으로 실행중인
sudo docker ps -a --filter "name = be-app" | grep -q . && sud
echo "=======================================================
echo "docker run"
```

```
sleep 5
sudo docker run -p 5000:8080 -d --name=be-app be/app #컨테이너
echo "=======================================================
echo "rmi process running" #같은 이름의 이미지가 있다면 삭제
sudo docker rmi -f $(sudo docker images -f "dangling=true" -q
```

13. 프론트엔드 파이프라인 스크립트

```
pipeline {
   agent any

   stages {
      stage('Clone') {
          steps {
              git branch: 'develop-FE', credentialsId : 'jenk
          }
      }
      stage('Build'){
          steps {
              dir('FE'){
                nodejs(nodeJSInstallationName: 'NodeJS'){
                  sh 'node --version'
                  sh 'npm install'
                  sh 'npm run build'
                }
              }
          }
      }
       stage('tar') {
            steps {
                dir('FE'){
                    sh 'tar -cvf dist.tar dist' #빌드 파일 압축
                }
            }
       }
      stage('ssh') {
        steps {
            dir('FE'){
                sshagent(credentials: ['pem']) {
                    sh 'ls'
                    sh 'ssh -o StrictHostKeyChecking=no ubunt
                                        #빌드 파일 서버 전송
                    sh 'scp dist.tar ubuntu@i10d205.p.ssafy.i
                }
            }
        }
      }
      stage('unpack'){
          steps {
              sshagent(credentials: ['pem']) {
                                        #빌드 파일 압축 해제
                    sh 'ssh ubuntu@i10d205.p.ssafy.io "cd /ho
                }
          }
      }
      stage('run.sh'){
          steps {
              sshagent(credentials: ['pem']) {
                                        #컨테이너 생성을 위한 쉘
                    sh 'ssh ubuntu@i10d205.p.ssafy.io "cd /ho
                }
          }
      }
   }
```

14. **프론트엔드 DockerFile**

```
# nginx 이미지를 사용합니다. 뒤에 tag가 없으면 latest 를 사용합니다.
FROM nginx:latest

# root 에 app 폴더를 생성
RUN mkdir /app

# work dir 고정
```

```
WORKDIR /app

# work dir 에 dist 폴더 생성 /app/dist
RUN mkdir ./dist

# host pc의 현재경로의 dist 폴더를 workdir 의 dist 폴더로 복사
ADD ./dist ./dist

# nginx 의 default.conf 를 삭제
RUN rm /etc/nginx/conf.d/default.conf

# host pc 의 default.conf 를 아래 경로에 복사
COPY ./default.conf /etc/nginx/conf.d

# 8082 포트 오픈
EXPOSE 8082
```

15. **프론트엔드 run.sh**

```
echo "start build image"
echo "=======================================================
sudo docker build -t fe/app .
echo "=======================================================
echo "complete built"
echo "=======================================================
echo "is any same name container running?"
sudo docker ps -a --filter "name = fe-app" | grep -q . && sud
echo "=======================================================
echo "docker run"
sleep 5
sudo docker run -p 8082:8082 -d --name=fe-app fe/app
echo "=======================================================
echo "rmi process running"
sudo docker rmi -f $(sudo docker images -f "dangling=true" -q
```

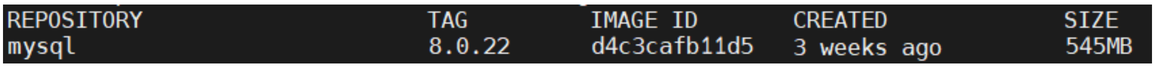
3-4 MySQL 컨테이너 생성

1. **MySQL Docker Image 다운로드**

   `docker pull mysql:8.0.22`

2. **다운로드 된 Docker Image 확인**

   `docker images -a`



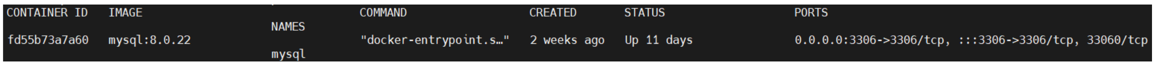
```
REPOSITORY                     TAG         IMAGE ID       CREATED          SIZE
mysql                          8.0.22      d4c3cafb11d5   3 weeks ago      545MB
```

3. **MySQL Docker 컨테이너 생성 & 실행**

   `docker run --name mysql -e MYSQL_ROOT_PASSWORD=[패스워드] -d -p 3306:3306 mysql:8.0.22`

4. **Docker 컨테이너 리스트 확인**

   `docker ps -a`



```
CONTAINER ID   IMAGE                        COMMAND                  CREATED       STATUS                     PORTS
                                NAMES
fd55b73a7a60   mysql:8.0.22                 "docker-entrypoint.s…"   2 weeks ago   Up 11 days                 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
                                mysql
```

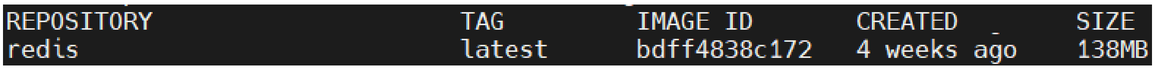
3-5 Redis 컨테이너 생성

1. **Redis Docker Image 다운로드**

   `docker pull --platform linux/amd64 redis`

2. **다운로드 된 Docker Image 확인**

   `docker images -a`



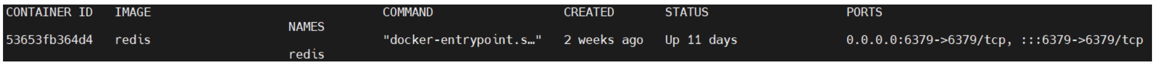
```
REPOSITORY                     TAG         IMAGE ID       CREATED          SIZE
redis                          latest      bdff4838c172   4 weeks ago      138MB
```

3. **Redis Docker 컨테이너 생성 & 실행**

   `docker run --name redis -p 6379:6379 --network redis-network -it -d redis`

4. **Docker 컨테이너 리스트 확인**

   `docker ps -a`



```
CONTAINER ID   IMAGE                              COMMAND                  CREATED        STATUS          PORTS
                                 NAMES
53653fb364d4   redis                              "docker-entrypoint.s…"   2 weeks ago    Up 11 days      0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
                                 redis
```

3-6 NginX 설치 및 설정

1. Nginx 설치

```
sudo apt update
sudo apt install nginx
```

2. Nginx 상태 체크

```
systemctl status nginx
```

3. HTTPS 적용

- Certbot 설치

```
sudo apt install certbot python3-certbot-nginx
```

- 인증서 발급

```
sudo certbot --nginx -d 도메인 이름 -d www.도메인 이름
```

- 옵션 선택 2번

```
Please choose whether or not to redirect HTTP traffic to HTTP
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver con
2: Redirect - Make all requests redirect to secure HTTPS acce
```

```
new sites, or if you're confident your site works on HTTPS. Y
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' t
```

- Nginx 설정 파일 작성

```
cd /etc/nginx/sites-available
sudo vim deploy-test.conf




server {

        location / {
                proxy_pass http://localhost:8082;
        }

        location /api/v1 {
                proxy_pass http://localhost:5000;
        }

        listen 443 ssl;
        ssl_certificate /etc/letsencrypt/live/<도메인>/fullchai
        ssl_certificate_key /etc/letsencrypt/live/<도메인>/priv
}

server {

        if ($host = <도메인>) {
                return 301 https://$host$request_uri;
        }

        listen 80;
        server_name <도메인>;
        return 404;
}
```

3-7 OpenVidu 설치 및 설정

1. Docker-compose 설치

- Docker-compose 실행 파일 다운로드

```
$ curl -L "https://github.com/docker/compose/releases/downloa
```

- 파일 실행 권한 설정

```
$ chmod +x /usr/bin/docker-compose
-rwxr-xr-x 1 root root  /usr/bin/docker-compose
$ ls -al /usr/bin/docker-compose
```

2. OpenVidu 서버 설치 (OpenVidu CE 버전 On premises 설치)

- 루트 권한

```
sudo su
```

- 폴더 이동

```
cd /opt
```

- 설치

```
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/insta
```

- 환경 설정

```
nano .env
```

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference

# NOTE: This file doesn't need to quote assignment values, li
# All values are stored as-is, even if they contain spaces, s

# Domain name. If you do not have one, the public IP of the m
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP={도메인 주소}

# OpenVidu SECRET used for apps to connect to OpenVidu server
OPENVIDU_SECRET={비밀번호}

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for
#                Users will see an ERROR when connected to we
# - owncert:     Valid certificate purchased in a Internet se
#                Please put the certificates files inside fol
#                with names certificate.key and certificate.c
# - letsencrypt: Generate a new certificate using letsencrypt
#                required contact email for Let's Encrypt in
#                variable.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a va
LETSENCRYPT_EMAIL={이메일}

# Proxy configuration
# If you want to change the ports on which openvidu listens,

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the f
# if you have chosen to deploy with the option CERTIFICATE_TY
HTTP_PORT=8081
```

```
# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to thi
HTTPS_PORT=8443
```

3. OpenVidu 서버 실행

```
./openvidu start
```

3-8 OpenVidu 와 Nginx 충돌 문제 해결

- 기존 Nginx 삭제

```
apt-get remove --purge nginx nginx-full nginx-common
```

- OpenVidu 삭제 후 재설치
- OpenVidu .env 파일 설정 (포트설정 x)

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference

# NOTE: This file doesn't need to quote assignment values, li
# All values are stored as-is, even if they contain spaces, s

# Domain name. If you do not have one, the public IP of the m
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP={도메인 주소}

# OpenVidu SECRET used for apps to connect to OpenVidu server
OPENVIDU_SECRET={비밀번호}

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for
#                Users will see an ERROR when connected to we
# - owncert:     Valid certificate purchased in a Internet se
#                  Please put the certificates files inside fol
#                  with names certificate.key and certificate.c
# - letsencrypt: Generate a new certificate using letsencrypt
#                  required contact email for Let's Encrypt in
#                  variable.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a va
LETSENCRYPT_EMAIL={이메일}

# Proxy configuration
# If you want to change the ports on which openvidu listens,

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the f
# if you have chosen to deploy with the option CERTIFICATE_TY
#HTTP_PORT=8081

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to thi
#HTTPS_PORT=8443
```

- OpenVidu 서버에 SSL 적용 확인 후 포트 번호 주석 해제
- AWS 서버 Nginx 재설치 및 설정

4. MySQL dump

```
-- MySQL dump 10.13  Distrib 8.0.34, for Win64 (x86_64)
--
-- Host: i10d205.p.ssafy.io    Database: yut_db
-- ------------------------------------------------------
-- Server version   8.0.22

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESUL
```

```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECK
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FO
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALU
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `blue_team_member`
--

DROP TABLE IF EXISTS `blue_team_member`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `blue_team_member` (
  `id` int NOT NULL AUTO_INCREMENT,
  `user_index` int NOT NULL,
  `game_code` varchar(255) DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK37tga58jub77ie1f29oowdiig` (`game_code`),
  KEY `FK1mics7rg7mjuirqnrwyjmcuq6` (`user_id`),
  CONSTRAINT `FK1mics7rg7mjuirqnrwyjmcuq6` FOREIGN KEY (`user
  CONSTRAINT `FK37tga58jub77ie1f29oowdiig` FOREIGN KEY (`game
) ENGINE=InnoDB AUTO_INCREMENT=382 DEFAULT CHARSET=utf8mb4 CO
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `blue_team_member`
--

LOCK TABLES `blue_team_member` WRITE;
/*!40000 ALTER TABLE `blue_team_member` DISABLE KEYS */;
INSERT INTO `blue_team_member` VALUES (1,3,'96ee57',26),(2,4,
/*!40000 ALTER TABLE `blue_team_member` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `blue_team_unit`
--

DROP TABLE IF EXISTS `blue_team_unit`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `blue_team_unit` (
  `id` int NOT NULL AUTO_INCREMENT,
  `contactor` varchar(255) DEFAULT NULL,
  `is_gole` bit(1) NOT NULL,
  `place` varchar(255) DEFAULT NULL,
  `scal` varchar(255) DEFAULT NULL,
  `stuff` varchar(255) DEFAULT NULL,
  `time` varchar(255) DEFAULT NULL,
  `unit_index` int NOT NULL,
  `game_code` varchar(255) DEFAULT NULL,
  `unit_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK6v652mro3fpen8eru3wph18b3` (`game_code`),
  KEY `FKdp5cajw4wgikn4bvuqg1ilhc6` (`unit_id`),
  CONSTRAINT `FK6v652mro3fpen8eru3wph18b3` FOREIGN KEY (`game_
  CONSTRAINT `FKdp5cajw4wgikn4bvuqg1ilhc6` FOREIGN KEY (`unit_
) ENGINE=InnoDB AUTO_INCREMENT=626 DEFAULT CHARSET=utf8mb4 CO
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `blue_team_unit`
--

LOCK TABLES `blue_team_unit` WRITE;
/*!40000 ALTER TABLE `blue_team_unit` DISABLE KEYS */;
INSERT INTO `blue_team_unit` VALUES (1,'ê²½ë¹ë³',_binary '\0'
/*!40000 ALTER TABLE `blue_team_unit` ENABLE KEYS */;
UNLOCK TABLES;

--
```

```
-- Table structure for table `boards`
--

DROP TABLE IF EXISTS `boards`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `boards` (
  `id` int NOT NULL AUTO_INCREMENT,
  `content` varchar(255) DEFAULT NULL,
  `created_date` datetime(6) NOT NULL,
  `subject` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLL
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `boards`
--

LOCK TABLES `boards` WRITE;
/*!40000 ALTER TABLE `boards` DISABLE KEYS */;
INSERT INTO `boards` VALUES (1,'0.8.1 ë² í íì¤í¸ ì¤ì\n ë´ì© :
/*!40000 ALTER TABLE `boards` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `friends`
--

DROP TABLE IF EXISTS `friends`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `friends` (
  `id` int NOT NULL AUTO_INCREMENT,
  `from_user_id` int NOT NULL,
  `to_user_id` int NOT NULL,
  `we_are_friend` bit(1) NOT NULL,
  PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=29 DEFAULT CHARSET=utf8mb4 COL
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `friends`
--

LOCK TABLES `friends` WRITE;
/*!40000 ALTER TABLE `friends` DISABLE KEYS */;
INSERT INTO `friends` VALUES (1,26,1,_binary ''),(2,1,26,_bin
/*!40000 ALTER TABLE `friends` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `game`
--

DROP TABLE IF EXISTS `game`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `game` (
  `game_code` varchar(255) NOT NULL,
  `blue_spy_hint` varchar(255) DEFAULT NULL,
  `blue_spy_id` int NOT NULL,
  `blue_team_reasoning_result` bit(1) NOT NULL,
  `game_speed` int NOT NULL,
  `game_theme` varchar(255) DEFAULT NULL,
  `mission_region` varbinary(255) DEFAULT NULL,
  `red_spy_hint` varchar(255) DEFAULT NULL,
  `red_spy_id` int NOT NULL,
  `red_team_reasoning_result` bit(1) NOT NULL,
  `winner` int NOT NULL,
  PRIMARY KEY (`game_code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `game`
```

```
--

LOCK TABLES `game` WRITE;
/*!40000 ALTER TABLE `game` DISABLE KEYS */;
INSERT INTO `game` VALUES ('010b35','ë°ì ì ì¶ìì ê²½ë³µêµìì ê¶
/*!40000 ALTER TABLE `game` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `missions`
--

DROP TABLE IF EXISTS `missions`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `missions` (
  `mission_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`mission_id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLL
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `missions`
--

LOCK TABLES `missions` WRITE;
/*!40000 ALTER TABLE `missions` DISABLE KEYS */;
INSERT INTO `missions` VALUES (1,'íë¦¬ì¡ê¸°'),(2,'ì¼ìµë¨¹ê¸°'
/*!40000 ALTER TABLE `missions` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `red_team_member`
--

DROP TABLE IF EXISTS `red_team_member`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
```

```
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `red_team_member` (
  `id` int NOT NULL AUTO_INCREMENT,
  `user_index` int NOT NULL,
  `game_code` varchar(255) DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKnq85ccqy3dnl83vxlc7twss8w` (`game_code`),
  KEY `FK6dpeo3g5fohtbm3u4bng51cs3` (`user_id`),
  CONSTRAINT `FK6dpeo3g5fohtbm3u4bng51cs3` FOREIGN KEY (`user_
  CONSTRAINT `FKnq85ccqy3dnl83vxlc7twss8w` FOREIGN KEY (`game_
) ENGINE=InnoDB AUTO_INCREMENT=382 DEFAULT CHARSET=utf8mb4 CO
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `red_team_member`
--

LOCK TABLES `red_team_member` WRITE;
/*!40000 ALTER TABLE `red_team_member` DISABLE KEYS */;
INSERT INTO `red_team_member` VALUES (1,0,'96ee57',28),(2,1,'
/*!40000 ALTER TABLE `red_team_member` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `red_team_unit`
--

DROP TABLE IF EXISTS `red_team_unit`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `red_team_unit` (
  `id` int NOT NULL AUTO_INCREMENT,
  `contactor` varchar(255) DEFAULT NULL,
  `is_gole` bit(1) NOT NULL,
  `place` varchar(255) DEFAULT NULL,
  `scal` varchar(255) DEFAULT NULL,
  `stuff` varchar(255) DEFAULT NULL,
  `time` varchar(255) DEFAULT NULL,
  `unit_index` int NOT NULL,
  `game_code` varchar(255) DEFAULT NULL,
  `unit_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKg36jabl5sam974ejme3cl2qfx` (`game_code`),
  KEY `FKoj91mi4ir22udjvxeer6a5bud` (`unit_id`),
  CONSTRAINT `FKg36jabl5sam974ejme3cl2qfx` FOREIGN KEY (`game_
  CONSTRAINT `FKoj91mi4ir22udjvxeer6a5bud` FOREIGN KEY (`unit_
) ENGINE=InnoDB AUTO_INCREMENT=636 DEFAULT CHARSET=utf8mb4 CO
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `red_team_unit`
--

LOCK TABLES `red_team_unit` WRITE;
/*!40000 ALTER TABLE `red_team_unit` DISABLE KEYS */;
INSERT INTO `red_team_unit` VALUES (1,'ê²½ë¹ë³',_binary '\0',
/*!40000 ALTER TABLE `red_team_unit` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `rooms`
--

DROP TABLE IF EXISTS `rooms`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `rooms` (
  `room_id` int NOT NULL AUTO_INCREMENT,
  `room_code` varchar(255) NOT NULL,
  `owner_id` int NOT NULL,
  `title` varchar(255) NOT NULL,
  `theme` varchar(255) NOT NULL,
  `game_speed` int NOT NULL,
  `is_public` tinyint(1) NOT NULL DEFAULT '1',
  `password` varchar(255) DEFAULT NULL,
  `start_at` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `end_at` datetime DEFAULT NULL,
  PRIMARY KEY (`room_id`),
  KEY `fk_room_owner` (`owner_id`),
  CONSTRAINT `fk_room_owner` FOREIGN KEY (`owner_id`) REFEREN
) ENGINE=InnoDB AUTO_INCREMENT=1102 DEFAULT CHARSET=utf8mb4 C
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `rooms`
--

LOCK TABLES `rooms` WRITE;
/*!40000 ALTER TABLE `rooms` DISABLE KEYS */;
INSERT INTO `rooms` VALUES (1076,'d08cef',38,'1','ì¤ë ',0,1,N
/*!40000 ALTER TABLE `rooms` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `units`
--

DROP TABLE IF EXISTS `units`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `units` (
  `unit_id` int NOT NULL AUTO_INCREMENT,
  `age` int NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `skill` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`unit_id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLL
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `units`
--
```

```
LOCK TABLES `units` WRITE;
/*!40000 ALTER TABLE `units` DISABLE KEYS */;
INSERT INTO `units` VALUES (1,67,'ëì','ë°ì  ì í ë¶ê°'),(2,32,
/*!40000 ALTER TABLE `units` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `user_game_history`
--

DROP TABLE IF EXISTS `user_game_history`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user_game_history` (
  `id` int NOT NULL AUTO_INCREMENT,
  `game_code` varchar(255) DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKiied6g14r41eom0rkao0s09od` (`game_code`),
  KEY `FK3u9wdxypw2rctuvkqnjuv5jcc` (`user_id`),
  CONSTRAINT `FK3u9wdxypw2rctuvkqnjuv5jcc` FOREIGN KEY (`user
  CONSTRAINT `FKiied6g14r41eom0rkao0s09od` FOREIGN KEY (`game
) ENGINE=InnoDB AUTO_INCREMENT=763 DEFAULT CHARSET=utf8mb4 CO
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user_game_history`
--

LOCK TABLES `user_game_history` WRITE;
/*!40000 ALTER TABLE `user_game_history` DISABLE KEYS */;
INSERT INTO `user_game_history` VALUES (1,'96ee57',28),(2,'96
/*!40000 ALTER TABLE `user_game_history` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `users`
--
```

```
DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `users` (
  `birth_date` date NOT NULL,
  `gender` varchar(1) NOT NULL,
  `id` int NOT NULL AUTO_INCREMENT,
  `is_delete` bit(1) NOT NULL,
  `created_date` datetime(6) NOT NULL,
  `nickname` varchar(10) NOT NULL,
  `email` varchar(30) NOT NULL,
  `role` varchar(30) NOT NULL,
  `password` varchar(60) NOT NULL,
  `profile_img_url` varchar(255) DEFAULT NULL,
  `refresh_token` varchar(255) DEFAULT NULL,
  `team_id` varchar(255) DEFAULT NULL,
  `member` varchar(255) DEFAULT NULL,
  `user_list_game_code` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_2ty1xmrrgtn89xt7kyxx6ta7h` (`nickname`),
  UNIQUE KEY `UK_6dotkott2kjsp8vw4d0m25fb7` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=44 DEFAULT CHARSET=utf8mb4 COL
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `users`
--

LOCK TABLES `users` WRITE;
/*!40000 ALTER TABLE `users` DISABLE KEYS */;
INSERT INTO `users` VALUES ('1996-03-09','ë¨',1,_binary '\0',
/*!40000 ALTER TABLE `users` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
```

```
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT *
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION *
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2024-02-16  7:07:10
```