포팅 메뉴얼

목차

1. 개발 환경

1-1. 프론트엔드 (Android)

- Android Studio (Iguana 2023.2.1 RC 2)
- Kotlin (1.9.0)
- JDK (OpenJDK 17)

1-2. 백엔드 (Spring Boot)

- IntelliJ (2023.3.4)
- JDK (OpenJDK 21)
- Spring Boot (3.2.3)
- MySQL (8.0.36)
- Redis (7.2.4)

1-3. AI

- Python (3.9.5)
- Uvicorn (0.11.3)
- Fast API

1-4. 서버 및 인프라

- Server (Ubuntu 20.04.6 LTS)
- Docker (25.0.4)
- Jenkins (2.430-jdk21)

2. 설정 파일 및 환경 변수 설정

2-1. 프론트엔드 (local.properties)

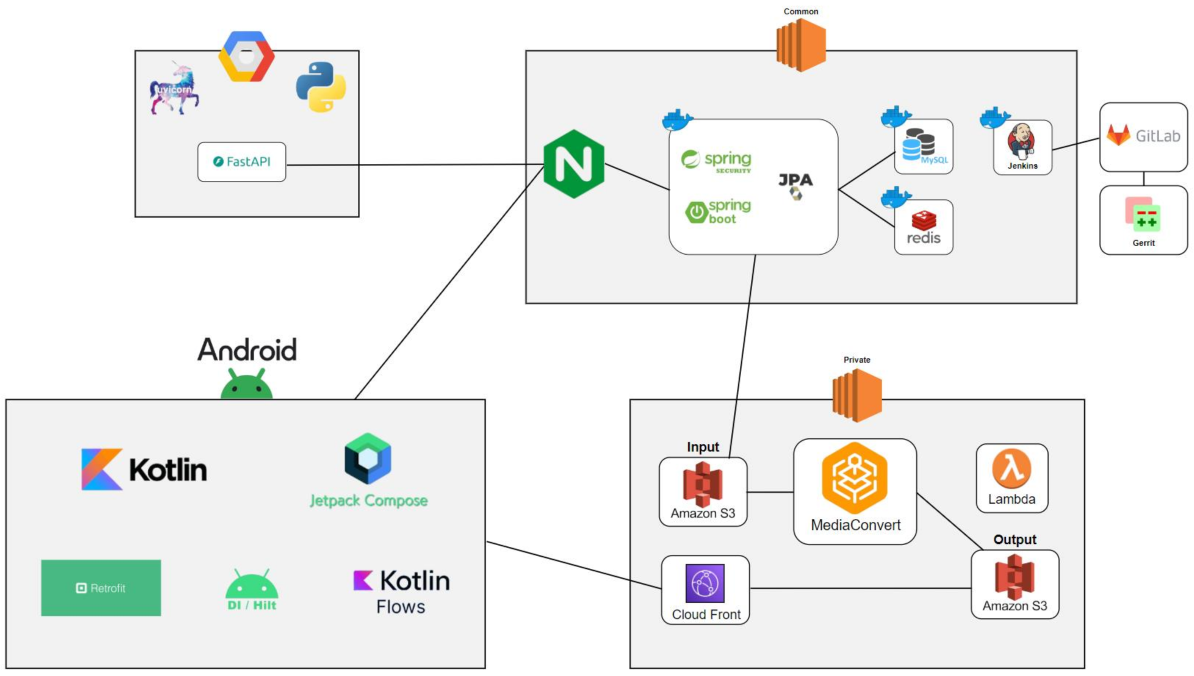
```
sdk.dir=C₩:₩₩Users₩₩SSAFY₩₩AppData₩₩Local₩₩Android₩₩Sdk
BASE_URL=https://j10d106.p.ssafy.io/ NAVER_LOGIN_CLIENT_ID=wblqEHfv0zDw2JAnjDvA
NAVER_LOGIN_CLIENT_SECRET=9_SwHqygYg NAVER_LOGIN_CLIENT_NAME=Vodle
SECRET_KEY=eqw=doasdijh213ljcpoqwr12 NAVER_MAPS_CLIENT_ID=fltrw9kto3
S3_URL=https://d3ouud0p1bofwo.cloudfront.net/
```

2-2. 백엔드 (application.yml)

```yaml
spring:
  #multipart
  servlet:
    multipart:
      max-file-size: 5MB
      max-request-size: 5MB
  # database
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url:
ENC(c2XaD9tTX+MwEYL2ZQtN6mgwrbYjjN72fa0VDYB9fywemrhWIzF7RRmTpMX/tyYzA+dfWo0OG
TLj6U0nrw3CZ8SG1E5c0aHf0SiXEpN4R/UYKR1x2XhFOxFrjve7N7TI4McICETq80YNIKjkJiSO5YSh2JApJ
cp6)
    username: ENC(M4jaZqWjVcaGxh6IMa2tUA==)
    password: ENC(iuI5JAKh61WCtgI+ycuhlw==)

    # hikari
    hikari:
      maximum-pool-size: 10
      connection-timeout: 5000
      connection-init-sql: SELECT 1
      validation-timeout: 2000
      minimum-idle: 10
      idle-timeout: 600000
      max-lifetime: 1800000

  # redis
  data:
    redis:
      host: ENC(LpNh+/vjAiMaJql6GVj5IjKfieYyfNUfj57BbylKYSA=)
      port: ENC(SgCoO9dYakGvtqFFCrgmjw==)

  # jpa
  jpa:
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        jdbc:
          time_zone: Asia/Seoul
        format_sql: true
        show_sql: true;
    open-in-view: false

  # jwt
  jwt:
    secret:
ENC(0Z0P50zspDrX8AuIBj2ts8bRuLpiidkxlqYPUu3Lo1Qme1upmXU9A46aehgAJcv1BMBX3UR0djyqIr
6s2fzZCw==)

  # hmac
  hmac:
    secret: ENC(7udXj2JZIot0lHv4nA5CJZeiv00aI8rJjTTX/PitNTpGBZP2IGPgzA==)

  # AWS S3
  cloud:
    aws:
      s3:
        bucket: ENC(w13REsuS0o8tjstlxera7jPMXKssk0CK)
        credentials:
          access-key: ENC(86hjlbi9+cK8pvTJv6/k9QpmcevZ+5cKDXWZ8jNZzQw=)
          secret-key:
ENC(ZUi0c0/TD5GFhvt/OAWI6w7/lqXHMhC759SVi5hXEZzL5Q3Z948MgpDLIz53uokb7K24e6v6RmU
=)
      region:
        static: ap-northeast-2

      # EC2 의 Spring Cloud 자동 구성 해제
      stack:
        auto: false

      # CloudFront
      cloud-front:
ENC(YOztYcBSgrn8DUGg3dWmxIAvG6ppWAIQdttQBPwDac7AK+i8Fd/x225kvqnFqi8m)

  # log
  logging.level:
    org.hibernate.SQL: debug

server:
  url: https://j10d106.p.ssafy.io/api

  port: 8081

  ssl:
    enabled: false

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

ai:
  api:
    url: ENC(3XawsTH65cyA4SxkV6wewT1hIbQEfGiOqv10RpWasME=)

naver:
  clientId: ENC(hL2L/8xOeTiXXqB2ixs7apjCtatiJGoI)
  clientSecret:
ENC(RAniocJzj1B7buu30mwJK1odtVdJm3vev8kml+whZc0n2yvVaIuEPHI9hb79BfHGGkKfEPp875U=)
```

3. EC2 서버 구축

3-1. 전체 프로젝트 구조



3-2. Docker

3-2-1. 도커 설치 전 사전 작업

```
sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

3-2-2. 도커 사전 작업 후 설치

```
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io
```

3-2-3. 도커 설치 후 그룹 추가

```
sudo su

usermod -aG docker {USER이름}
```

3-3. Jenkins

3-3-1. 젠킨스 이미지 불러오기

```
sudo docker pull jenkins/jenkins:2.430-jdk21
```

3-3-2. 젠킨스 실행 전 사전 작업

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

sudo chmod -R 777 /home/ubuntu/jenkins-data
```

3-3-3. 젠킨스 실행

```
sudo docker run -d -p 8080:8080 -v /home/ubuntu/jenkins-data:/var/jenkins_home --name jenkins
jenkins/jenkins:2.430-jdk21
```

3-3-4. 젠킨스 접속

```
http://j10d106.p.ssafy.io:8080
```

3-3-5. 젠킨스 초기 비밀번호

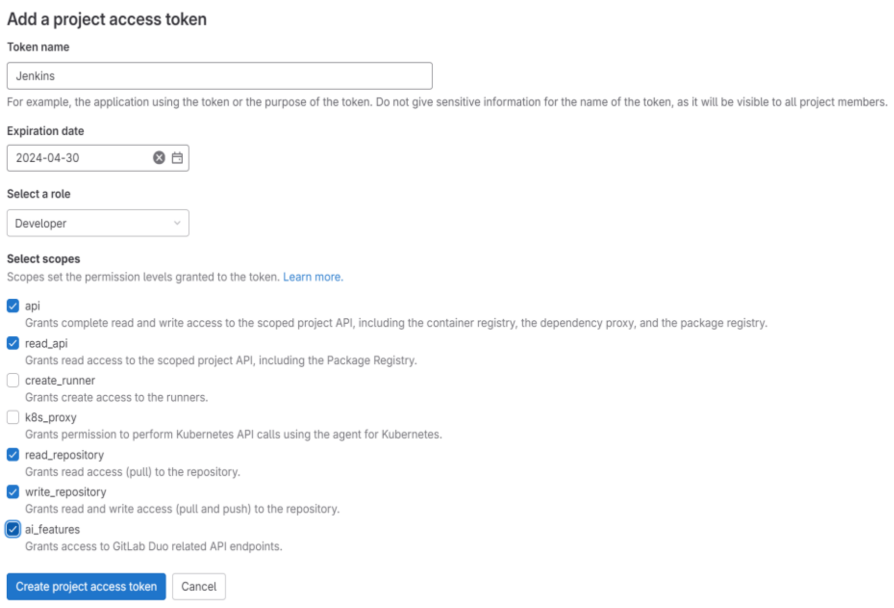
```
docker exec -it jenkins bash

cd /var/jenkins_home/secrets/

cat initialAdminPassword
```

3-3-6. 젠킨스 플러그인 설치 목록

```
GitLab Plugin

: GitLab 웹 훅을 걸기 위해 필요


SSH Agent Plugin

: 외부 SSH 접속을 위해 필요
```

3-3-7. 깃랩 연동 설정

[1] 깃랩의 Access Token 생성



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

2024-04-30

Select a role

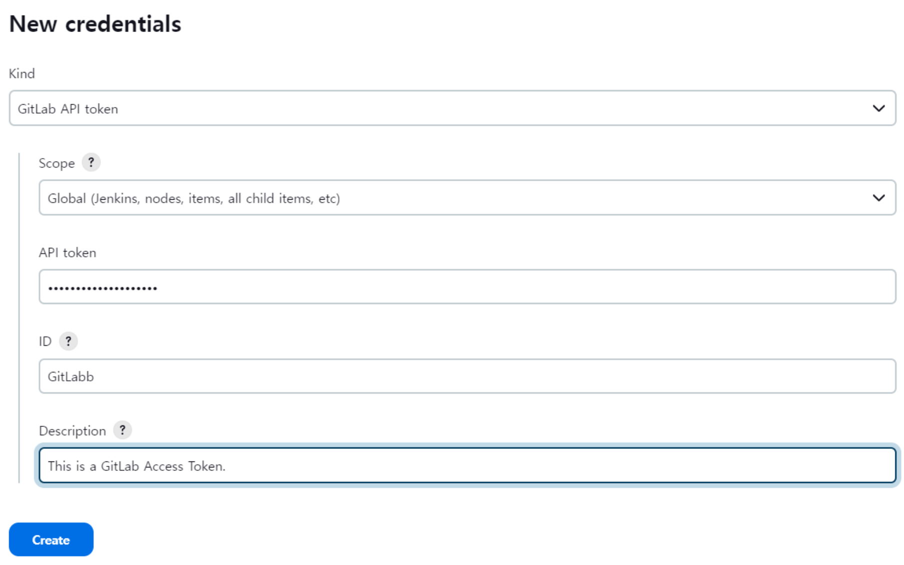
Developer

Select scopes

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

- [x] api
  Grants complete read and write access to the scoped project API, including the container registry, the dependency proxy, and the package registry.
- [x] read_api
  Grants read access to the scoped project API, including the Package Registry.
- [ ] create_runner
  Grants create access to the runners.
- [ ] k8s_proxy
  Grants permission to perform Kubernetes API calls using the agent for Kubernetes.
- [x] read_repository
  Grants read access (pull) to the repository.
- [x] write_repository
  Grants read and write access (pull and push) to the repository.
- [x] ai_features
  Grants access to GitLab Duo related API endpoints.

[2] [1]에서 받은 토큰 정보를 젠킨스 credentials 추가


New credentials

Kind
GitLab API token

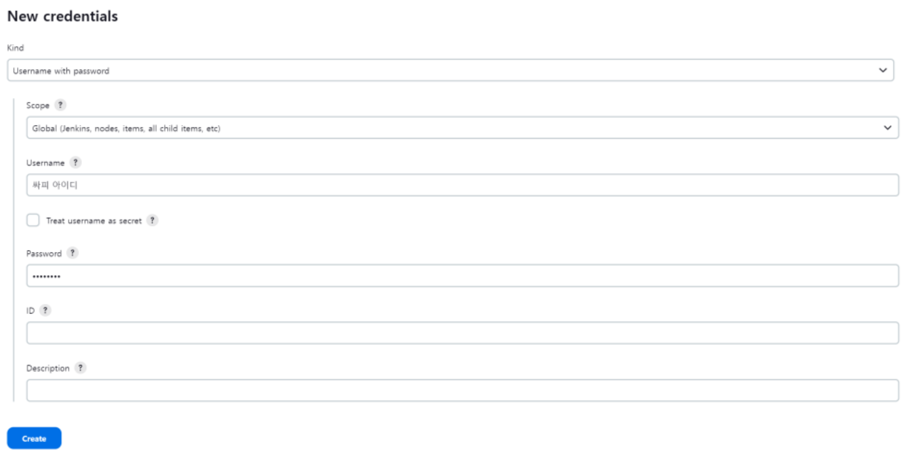
Scope
Global (Jenkins, nodes, items, all child items, etc)

API token
••••••••••••••••••••

ID
GitLabb

Description
This is a GitLab Access Token.

[3] GitLab 접속을 위한 아이디, 비밀번호 입력


New credentials

Kind
Username with password

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

Username
싸피 아이디

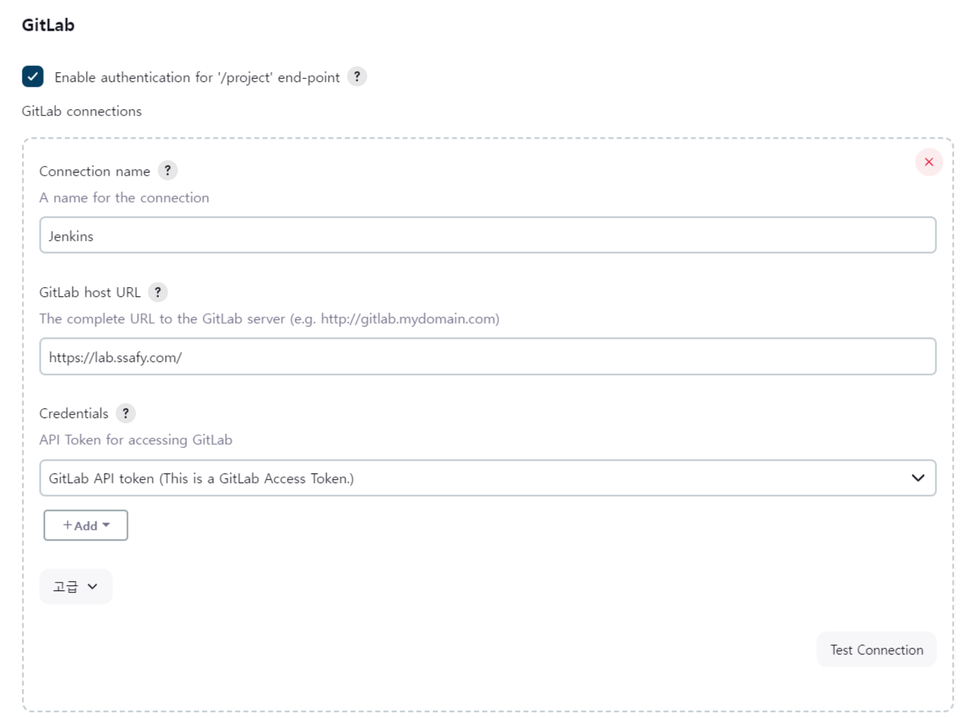
☐ Treat username as secret

Password
•••••••••

ID

Description

[4] 젠킨스 관리에서 System 설정



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

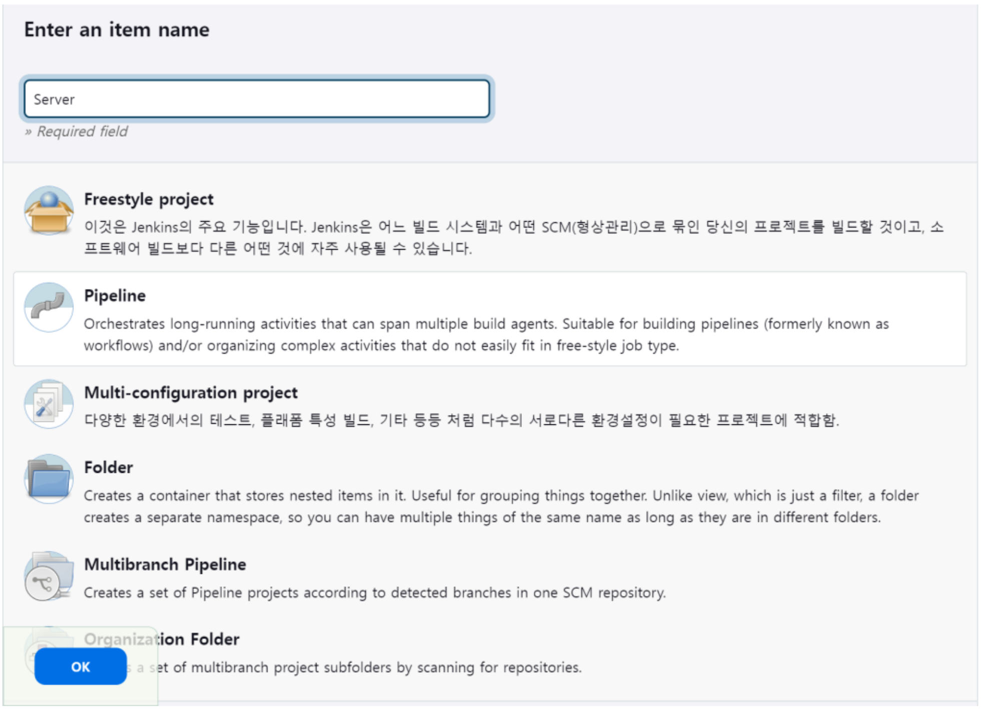
GitLab connections

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 (This is a GitLab Access Token.)

[5] 파이프 라인 추가


Enter an item name

Server

» 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.

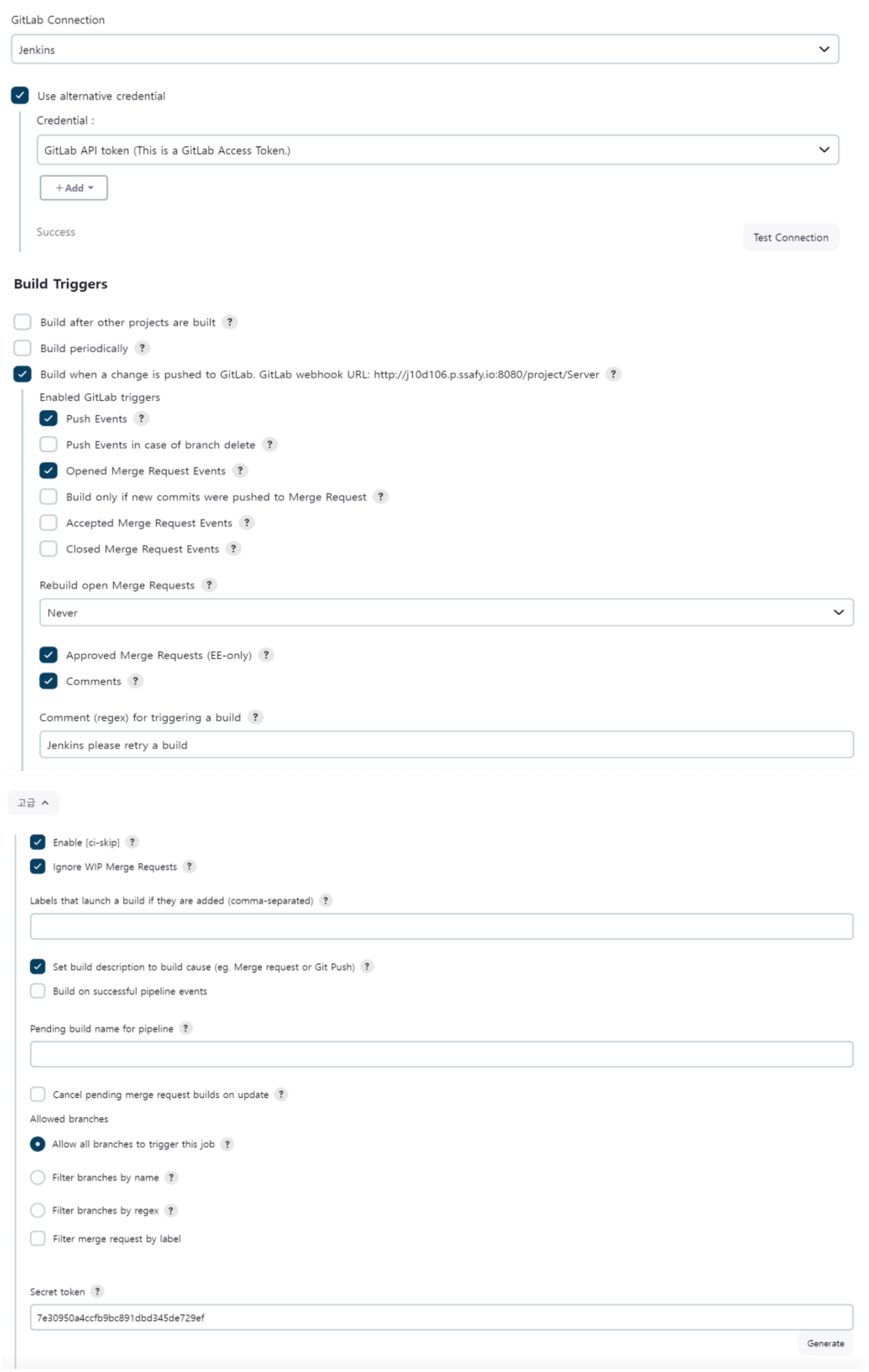
Multi-configuration project
다양한 환경에서의 테스트, 플래폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

[6] 파이프 라인 설정



GitLab Connection

Jenkins

[✓] Use alternative credential

Credential :

GitLab API token (This is a GitLab Access Token.)

Build Triggers

[ ] Build after other projects are built

[ ] Build periodically

[✓] Build when a change is pushed to GitLab. GitLab webhook URL: http://j10d106.p.ssafy.io:8080/project/Server

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

[✓] 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

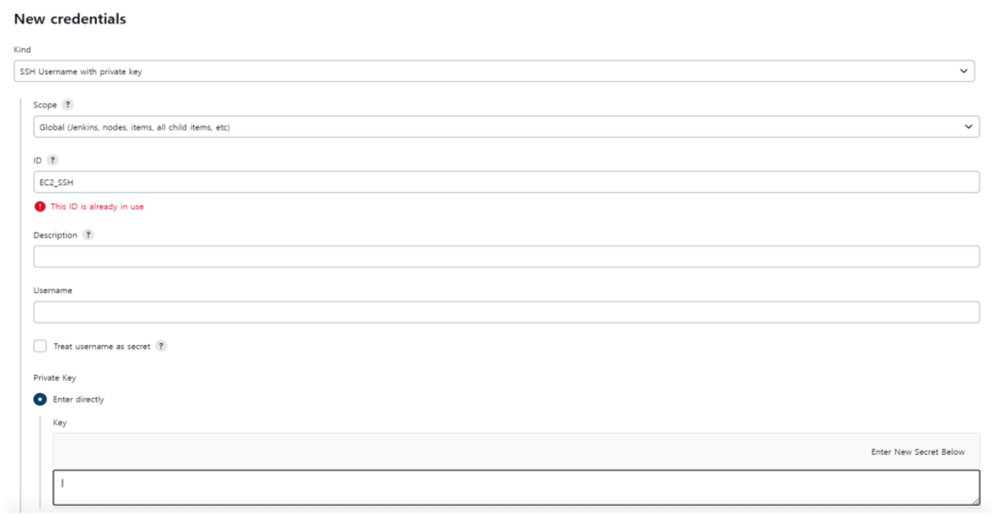
Filter branches by regex

[ ] Filter merge request by label

Secret token

7e30950a4ccfb9bc891dbd345de729ef

(비밀키는 따로 저장 필요 --> 깃허브 추가)



New credentials

Kind: SSH Username with private key

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

ID: EC2_SSH

This ID is already in use

Description:

Username:

Treat username as secret

Private Key

● Enter directly

Key

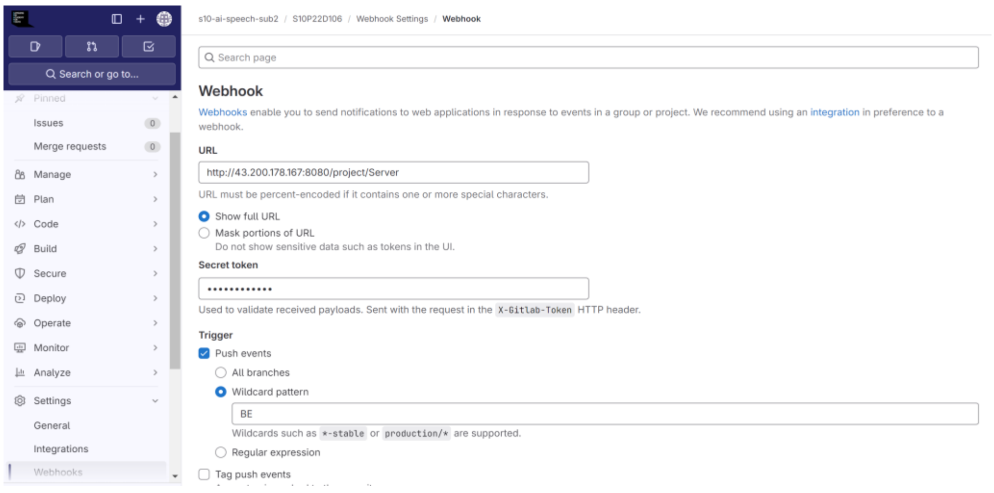
Enter New Secret Below

(.pem 내용 그대로 key에 복사/붙여넣기)

[7] WAS 서버에 대한 스크립트 작성

```
pipeline {
    agent any
    environment {
        JASYPT_KEY = 'vodlessafy'
    }
    stages {
        stage('Clone') {
            steps {
                git branch: 'BE', credentialsId: 'GitLab User',url: 'https://lab.ssafy.com/s10-ai-speech-sub2/S10P22D106.git'
            }
        }
        stage('Build') {
            steps {
                dir("./BE_repo") {
                    sh "chmod +x ./gradlew"
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deploy') {
           steps {
               dir('BE_repo/build/libs') {
                   sh "chmod 777 ./*"
                   sshagent(credentials: ['EC2_SSH']) {
                        sh 'ssh -o StrictHostKeyChecking=no ubuntu@ip-172-26-2-121'
                        sh 'scp BE_repo-0.0.1-SNAPSHOT.jar ubuntu@ip-172-26-2-121:/home/ubuntu'
                        sh 'scp ../../Dockerfile ubuntu@ip-172-26-2-121:/home/ubuntu'
                        sh 'ssh ubuntu@ip-172-26-2-121 "sh run.sh"'
                   }
               }
           }
       }
    }
}
```

[8] GitLab에 웹훅 걸기



s10-ai-speech-sub2 / S10P22D106 / Webhook Settings / **Webhook**

Search page

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://43.200.178.167:8080/project/Server

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.

Trigger

- [✓ Push events]
  - All branches
  - [✓ Wildcard pattern]
    BE
    Wildcards such as `*-stable` or `production/*` are supported.
  - Regular expression
- Tag push events

[9] /home/ubuntu 위치에 run.sh 작성하기

```
GitLab Plugin

: GitLab 웹 훅을 걸기 위해 필요


SSH Agent Plugin
```

3-4. MySQL

3-4-1. 이미지 가져오기

```
docker pull mysql:8.0.36
```

3-4-2. 실행하기

```
sudo docker run -p 2222:3306 --name mysql -e MYSQL_ROOT_PASSWORD=WssafyW -d
mysql:8.0.36
```

3-4-3. CLI 접속하기

```
docker exec -it mysql mysql -uroot -p
```

3-5. Redis

3-5-1. 이미지 가져오기

```
sudo docker pull redis:7.2.4
```

3-5-2. 해당 이미지 실행하기

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

3-5-3. 레디스 접속하기

```
sudo docker exec -it redis bash
```

3-5-4. CLI 접속하기

```
redis-cli
```

3-6. NGINX

3-6-1. NGINX 설치

```
sudo apt update

sudo apt install nginx
```

3-6-2. NGINX 상태 체크하기

```
systemctl status nginx
```

3-6-1. NGINX 적용시키기

```
1. Certbot 설치

sudo apt install certbot python3-certbot-nginx

2. 인증서 발급

sudo certbot --nginx -d {도메인 이름}

2-1. 옵션 선택 (2번)
```