⚒️

포팅 매뉴얼 - E204(움직여! ZOO)

1. [개발 환경]

- **Frontend**

Node Js	`21.6.0`
Recoil	`0.7.7`
tensorflow/face-detection	`1.0.2`
tensorflow/hand-pose-detection	`2.0.1`
Vscode	`1.73.1`
Tailwind	`v3`
Axios	`1.6.7`

- **Backend**

Spring Boot	`3.2.2`
Spring Security	`6.2.1`
ORM	`JPA(Hibernate)`
JDK	`OpenJDK17`
MySQL	`8.3.0`
Mysql Workbench	`8.0 CE`

Redis	`7.2.4`
Intellj	`2023.3.2`

- **Server**

Ec2	`Ubuntu 20.04 LTS`
Nginx	`1.18.0`
Jenkins	`2.426.3`
openvidu-server	`2.29.0`
docker	`25.0.0`
Terminus	

- **Grahpic**

Blender	`4.0.2`

1. Docker

2.1 docker 설치

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl software-properties-common
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 focal
sudo apt update
sudo apt install docker-ce
docker --version
```

2.2 sudo없이 docker 명령어 사용하기

```
sudo usermod -aG docker ${USER}
id -nG
```

3. Nginx 환경설정

3.1 Nginx, letsencrypt, cerbot 설치

```
sudo apt install nginx -y
sudo systemctl status nginx
sudo apt-get install letsencrypt
sudo apt-get install cerbot python3-cerbot-nginx
sudo certbot --nginx

[your domain]
[2번 선택(redirect)]
```

3.2 Nginx conf 수정

```
server {
        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html;

        server_name i10e204.p.ssafy.io;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ /index.html;
        }
        location /api {
                proxy_pass http://i10e204.p.ssafy.io:5000;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
        }
        location /oauth2{
                proxy_pass http://i10e204.p.ssafy.io:5000;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                }
        location /login{
                proxy_pass http://i10e204.p.ssafy.io:5000;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
        }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i10e204.p.ssafy.io/fullchain.pem; # managed by
    ssl_certificate_key /etc/letsencrypt/live/i10e204.p.ssafy.io/privkey.pem; # managed
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
```

```
}
server {
    if ($host = i10e204.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen 80 default_server;
        listen [::]:80 default_server;

        server_name i10e204.p.ssafy.io;
    return 404; # managed by Certbot


}
```

3.3 nginx 명령어

```
# niginx 시작
sudo service nginx start

# nginx 중지
sudo service nginx stop

# nginx 재시작
sudo service nginx restart
```

4. Docker Container 설치

4.1 mysql 설치 및 세팅

```
docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=[your password]-d -p 3306:3306
docker exec -it mysql bash
mysql -u root -p
[enter your password]
create database movezoo
exit
```

4.2 redis 설치

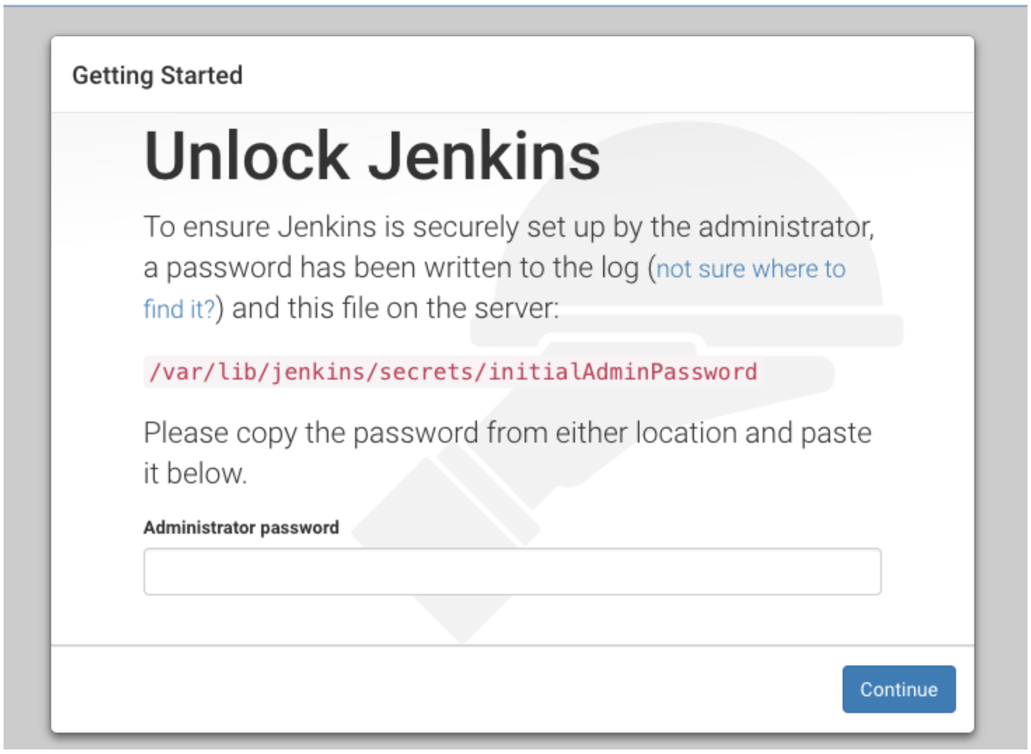
```
docker run -p 6379:6379 --name redis -d redis:latest --requirepass [your password]
```

4.3 jenkins 설치

```
cd /home/ubuntu/ && mkdir jenkins-data
sudo ufw allow 8080 /tcp
sudo ufw reload
sudo ufw status
docker run -d -v jenkins_home:/var/jenkins_home -p 8080:8080 -p 50000:50000 --restart=on
sudo docker logs jenkins
```

```
# 로그에 출력되는 비밀번호 확인
.
.
.
please use the following password to proceed to installtion;
[your password]
.
.
.
```

4.3.1 초기 패스워드 입력



Getting Started

Unlock Jenkins

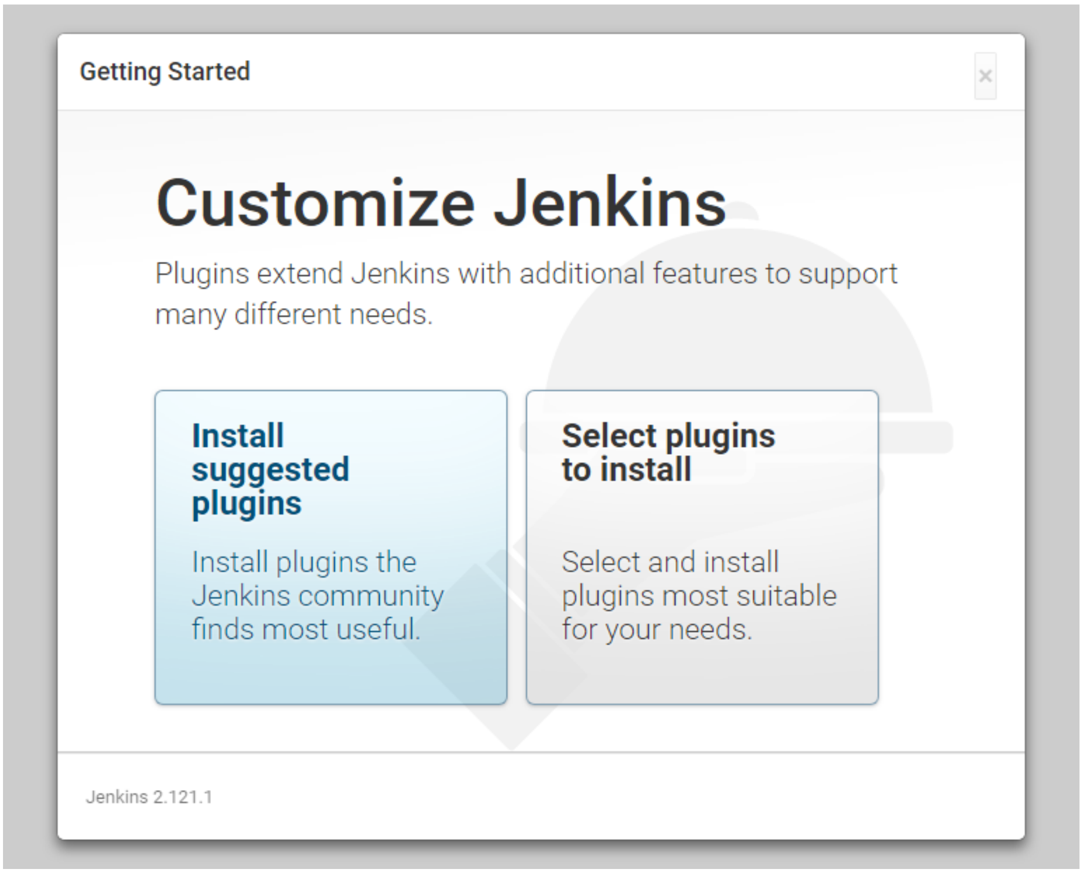
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

4.3.2 플러그인 설치



Getting Started

Customize Jenkins

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

Install suggested plugins

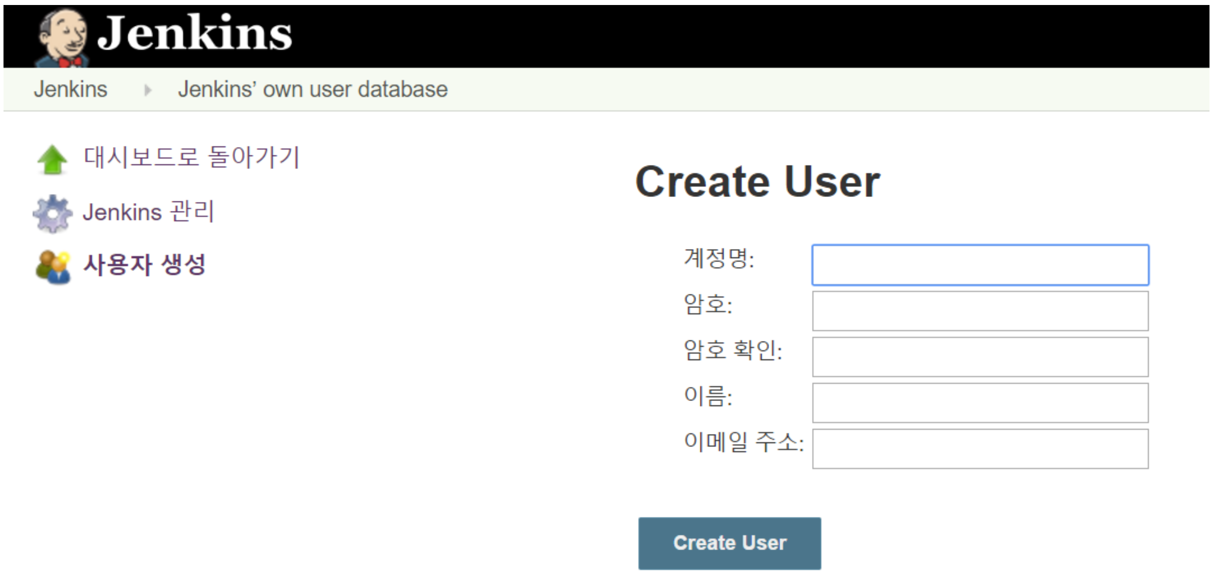
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.121.1

4.3.3 admin 계정 생성



Jenkins ▸ Jenkins' own user database

대시보드로 돌아가기
Jenkins 관리
사용자 생성

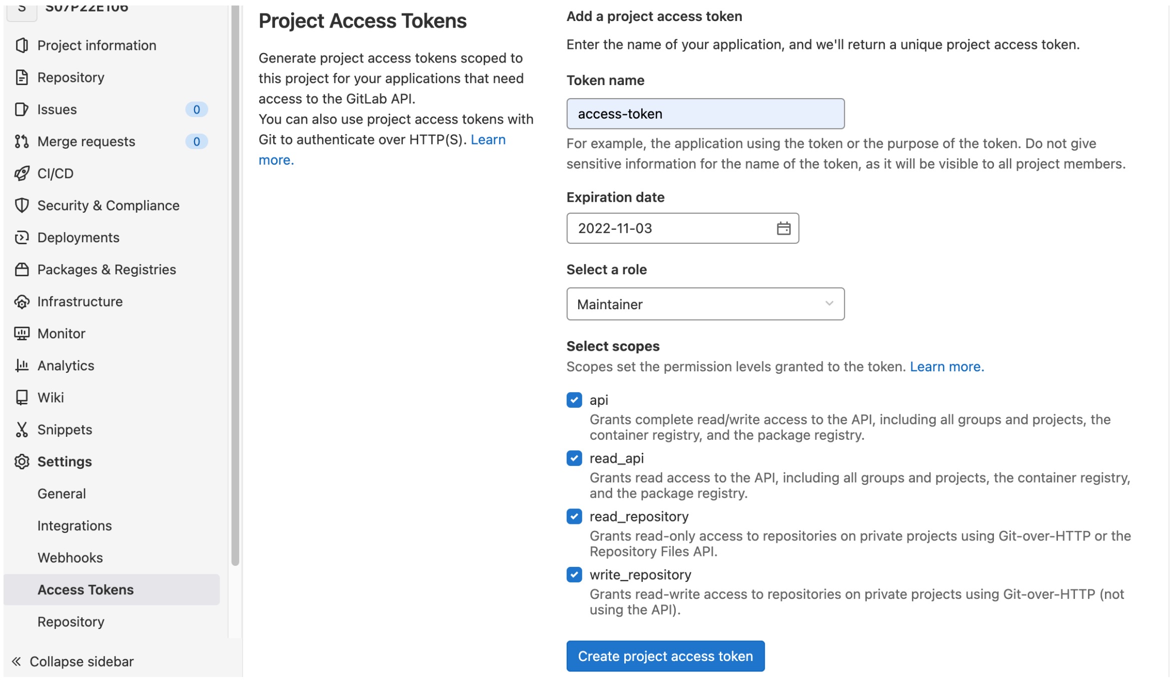
Create User

계정명:	
암호:	
암호 확인:	
이름:	
이메일 주소:	

Create User

4.3.4 jenkins - git lab 연동

- Gitlab의 Settings → Access Tokens에서 토큰 생성



S07P22E106

- Project information
- Repository
- Issues 0
- Merge requests 0
- CI/CD
- Security & Compliance
- Deployments
- Packages & Registries
- Infrastructure
- Monitor
- Analytics
- Wiki
- Snippets
- Settings
  - General
  - Integrations
  - Webhooks
  - Access Tokens
  - Repository
- Collapse sidebar

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.

Add a project access token

Enter the name of your application, and we'll return a unique project access token.

Token name

access-token

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

2022-11-03

Select a role

Maintainer

Select scopes

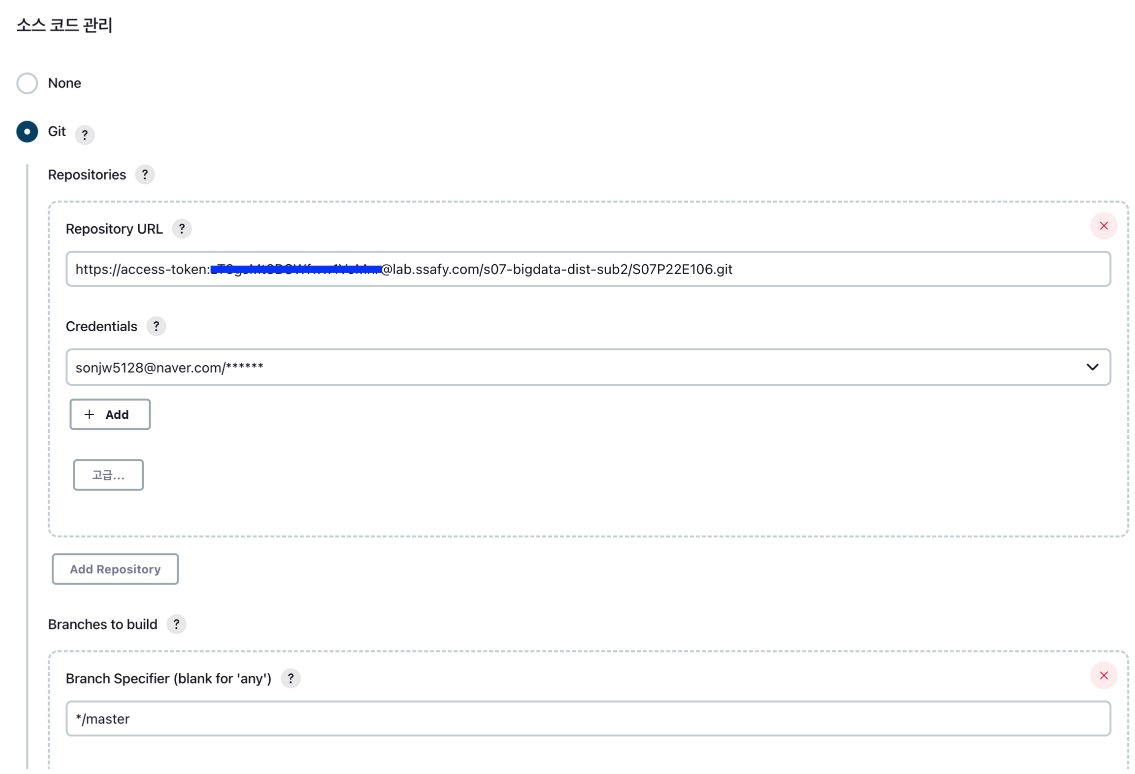
Scopes set the permission levels granted to the token. Learn more.

- ✓ api
  Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- ✓ read_api
  Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ✓ read_repository
  Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ✓ write_repository
  Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Create project access token

4.3.5 프로젝트 URL입력

- 프로젝트 선택 → 구성 → 소스코드 관리 → Git



소스 코드 관리

None [✓ Git]

Repositories

Repository URL

https://access-token:[redacted]@lab.ssafy.com/s07-bigdata-dist-sub2/S07P22E106.git

Credentials

sonjw5128@naver.com/******

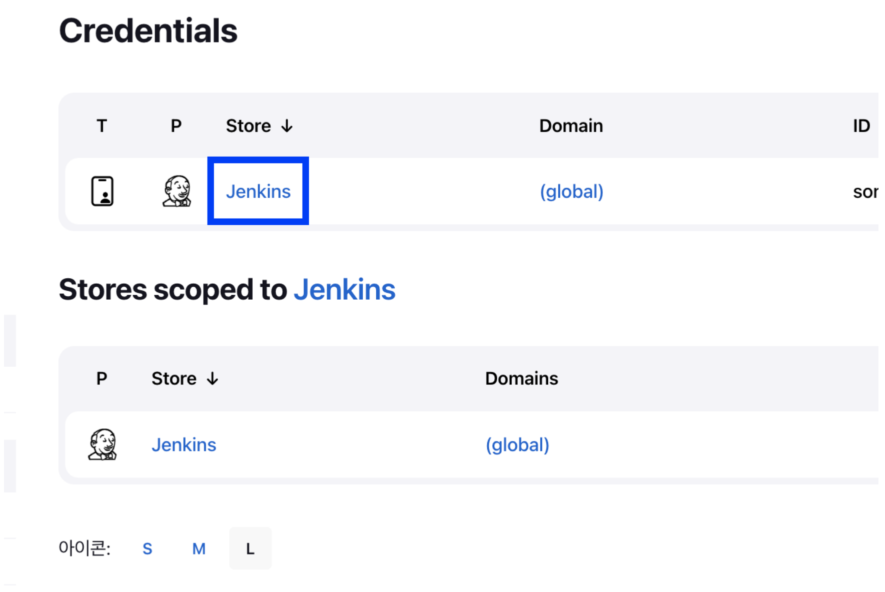
Branches to build

Branch Specifier (blank for 'any')

*/master

4.3.6 Credentials 생성

- Credentials는 Add를 클릭해서 username with password로 깃랩 아이디와 비밀번호를 입력해서 credential 생성

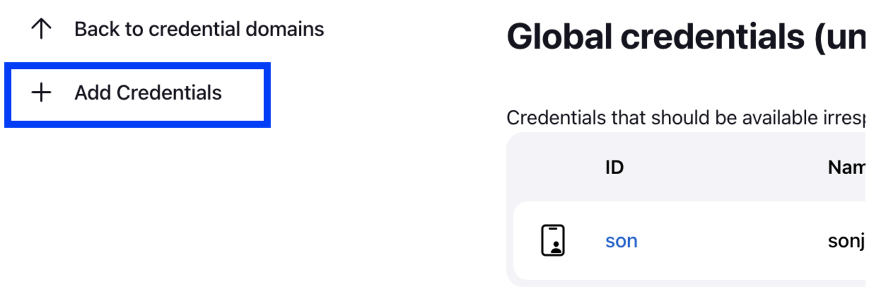


Credentials

T	P	Store ↓	Domain	ID
		Jenkins	(global)	sor

Stores scoped to Jenkins

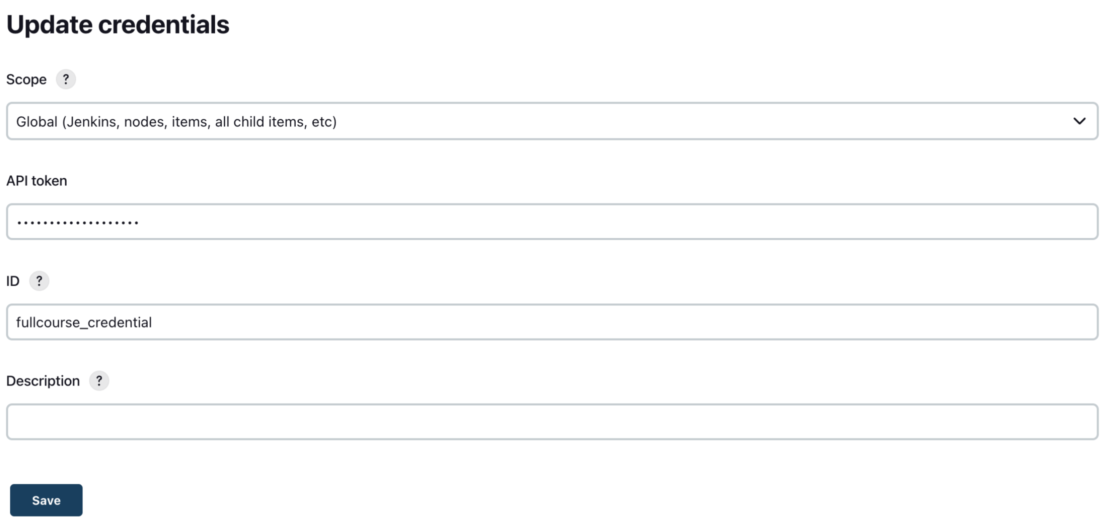
P	Store ↓	Domains
	Jenkins	(global)



Global credentials (un

Credentials that should be available irres

ID	Nam
son	sonj



Update credentials

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

API token: ····················

ID: fullcourse_credential

Description:

4.4 openvidu 설치

```
# 관리자 권한
$ sudo su

# openvidu가 설치되는 경로
$ cd /opt
```

```
# openvidu on promises 설치
$ curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | b

$ exit

$ cd openvidu
```

4.4.1 openvidu docker-compose.yml 수정

```
vi /opt/openvidu/docker-compose.yml
```

```
version: '3.1'

services:

    openvidu-server:
        image: openvidu/openvidu-server:2.29.0
        restart: on-failure
        network_mode: host
        entrypoint: ['/usr/local/bin/entrypoint.sh']
        volumes:
            - ./coturn:/run/secrets/coturn
            - /var/run/docker.sock:/var/run/docker.sock
            - ${OPENVIDU_RECORDING_PATH}:${OPENVIDU_RECORDING_PATH}
            - ${OPENVIDU_RECORDING_CUSTOM_LAYOUT}:${OPENVIDU_RECORDING_CUSTOM_LAYOUT}
            - ${OPENVIDU_CDR_PATH}:${OPENVIDU_CDR_PATH}
        env_file:
            - .env
        environment:
            - SERVER_SSL_ENABLED=false
            - SERVER_PORT=5443
            - KMS_URIS=["ws://localhost:8888/kurento"]
            - COTURN_IP=${COTURN_IP:-auto-ipv4
                        - COTURN_PORT=${COTURN_PORT:-3478}
        logging:
            options:
                max-size: "${DOCKER_LOGS_MAX_SIZE:-100M}"

    kms:
        image: ${KMS_IMAGE:-kurento/kurento-media-server:7.0.1}
        restart: always
        network_mode: host
        ulimits:
          core: -1
        volumes:
            - /opt/openvidu/kms-crashes:/opt/openvidu/kms-crashes
            - ${OPENVIDU_RECORDING_PATH}:${OPENVIDU_RECORDING_PATH}
            - /opt/openvidu/kurento-logs:/opt/openvidu/kurento-logs
        environment:
            - KMS_MIN_PORT=40000
            - KMS_MAX_PORT=57000
            - GST_DEBUG=${KMS_DOCKER_ENV_GST_DEBUG:-}
            - KURENTO_LOG_FILE_SIZE=${KMS_DOCKER_ENV_KURENTO_LOG_FILE_SIZE:-100}
            - KURENTO_LOGS_PATH=/opt/openvidu/kurento-logs
        logging:
            options:
                max-size: "${DOCKER_LOGS_MAX_SIZE:-100M}"

    coturn:
        image: openvidu/openvidu-coturn:2.29.0
        restart: on-failure
        ports:
            - "${COTURN_PORT:-3478}:${COTURN_PORT:-3478}/tcp"
            - "${COTURN_PORT:-3478}:${COTURN_PORT:-3478}/udp"
        env_file:
            - .env
        volumes:
            - ./coturn:/run/secrets/coturn
        command:
            - --log-file=stdout
            - --listening-port=${COTURN_PORT:-3478}
                        - --fingerprint
            - --min-port=${COTURN_MIN_PORT:-57001}
            - --max-port=${COTURN_MAX_PORT:-65535}
            - --realm=openvidu
            - --verbose
            - --use-auth-secret
            - --static-auth-secret=$${COTURN_SHARED_SECRET_KEY}
        logging:
            options:
                max-size: "${DOCKER_LOGS_MAX_SIZE:-100M}"

    nginx:
        image: openvidu/openvidu-proxy:2.29.0
        restart: always
        network_mode: host
        volumes:
            -  /etc/letsencrypt:/etc/letsencrypt
            - ./owncert:/owncert
            - ./custom-nginx-vhosts:/etc/nginx/vhost.d/
            - ./custom-nginx-locations:/custom-nginx-locations
            - ${OPENVIDU_RECORDING_CUSTOM_LAYOUT}:/opt/openvidu/custom-layout
        environment:
            - DOMAIN_OR_PUBLIC_IP=${DOMAIN_OR_PUBLIC_IP}
            - CERTIFICATE_TYPE=${CERTIFICATE_TYPE}
            - LETSENCRYPT_EMAIL=${LETSENCRYPT_EMAIL}
            - PROXY_HTTP_PORT=${HTTP_PORT:-}
            - PROXY_HTTPS_PORT=${HTTPS_PORT:-}
            - PROXY_HTTPS_PROTOCOLS=${HTTPS_PROTOCOLS:-}
            - PROXY_HTTPS_CIPHERS=${HTTPS_CIPHERS:-}
            - PROXY_HTTPS_HSTS=${HTTPS_HSTS:-}
            - ALLOWED_ACCESS_TO_DASHBOARD=${ALLOWED_ACCESS_TO_DASHBOARD:-}
            - ALLOWED_ACCESS_TO_RESTAPI=${ALLOWED_ACCESS_TO_RESTAPI:-}
            - PROXY_MODE=CE
            - WITH_APP=true
            - SUPPORT_DEPRECATED_API=${SUPPORT_DEPRECATED_API:-false}
            - REDIRECT_WWW=${REDIRECT_WWW:-false}
            - WORKER_CONNECTIONS=${WORKER_CONNECTIONS:-10240}
                        - PUBLIC_IP=${PROXY_PUBLIC_IP:-auto-ipv4}
        logging:
            options:
                max-size: "${DOCKER_LOGS_MAX_SIZE:-100M}"
```

4.3.2 openvidu 설정 변경

```
vi opt/openvidu/.env
```

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

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP=[your domain]

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to Ope
OPENVIDU_SECRET=MY_SECRET

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for production use.
#                Users will see an ERROR when connected to web page.
# - owncert:     Valid certificate purchased in a Internet services company.
#                Please put the certificates files inside folder ./owncert
#                with names certificate.key and certificate.cert
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                variable.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL=[your email]

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

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

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

--- 이하 변경사항없음
```

4.3.3 중복될 수 있는 인증서 삭제

```
# openvidu에도 https를 적용시키기 위해 certbot으로 발급받았다. 중복되어 적용이 안될 수 있어 삭제
cd /opt/openvidu
sudo rm -rf certificates
```

4.3.4 openvidu 실행

```
cd /opt/openvidu
./openvidu start

# 종료할 때는 같은 경로에서 stop
./openvidu stop
```

5. 환경변수와 docker file

5.1 스프링 부트 - application.properties

```
server.port=5000
server.ssl.enabled=false

OPENVIDU_URL=[your domain]
OPENVIDU_SECRET=MY_SECRET

## MySQL
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
## DB Source URL
spring.datasource.url=[your mysql domain]/movezoo

## DB username
spring.datasource.username=[your mysql username]
## DB password
spring.datasource.password=[your mysql password]

#spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=none

spring.jpa.properties.hibernate.format_sql=true
spring.jpa.defer-datasource-initialization =true

#google mail
spring.mail.host = smtp.gmail.com
spring.mail.port=587
spring.mail.username=[your google id]
spring.mail.password=[your google smpt password]
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.timeout=5000
spring.mail.properties.mail.smtp.starttls.enable=true

## Redis
spring.data.redis.host=[your domain]
spring.data.redis.port=6379
spring.data.redis.password=[your redis password]
```

```
## SocialLogin
spring.security.oauth2.client.registration.google.client-id=[your google client id]
spring.security.oauth2.client.registration.google.client-secret=[your google secret key]
spring.security.oauth2.client.registration.google.redirect-uri=[your redirect url]
spring.security.oauth2.client.registration.google.scope=profile, email
```

5.1.1 스프링 부트 - Dockerfile

```
FROM openjdk:17
ARG JAR_FILE=build/libs/[your project name]-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} movezoo-0.0.1-SNAPSHOT.jar
ENTRYPOINT ["java","-jar","/[your project name]-0.0.1-SNAPSHOT.jar"]
```

5.1.2 스프링 부트 - docker-compose.yml

```
version: "3"
services:
  container_name: backend
  build:
    context: ./
    dockerfile: Dockerfile
  volumes:
    - ./src/main/resources:/src/main/resources
  ports:
    - "5000:5000"
  restart: alwaysㅌ
```

6. 외부서비스

- 구글 메일 : `https://cloud.google.com/appengine/docs/standard/go111/mail?hl=ko`
- 구글 로그인 : `https://cloud.google.com/identity-platform/docs/web/google?hl=ko`