Kidkidk 포팅 매뉴얼

> 제목 1 (#) : 노란색 배경
> 제목 2 (##) : 초록색 배경
> 제목 3 (###) : 보라색 배경

1. 개발환경

1.1. Frontend

- Node JS 20.11.0 (LTS)
- React 18.2.0
    - Recoil 0.7.7
- Axios 1.6.7

1.2. Backend

- Java
    - Azul Zulu 17.0.9+8 (LTS)
    - Spring Boot 3.2.2
        - Spring Data JPA 3.2.2
        - Spring Security 3.2.2
        - Spring-Kafka 3.1.1
        - JUnit 5.10.1
        - Lombok 1.18.30
    - Gradle 8.5

1.3. Server

- Ubuntu 20.04.6 LTS
- Nginx 1.18.0
- Docker 25.0.0
- Docker Compose (plugin) 2.24.1
- Jenkins 2.426.3

1.4. Database

- MySQL 8.0.36

1.5. Event Streaming Platform

- Kafka
    - Kafka (wurstmeister) 2.8.1
    - Zookeeper (wurstmeister) 3.4.13
    - Kafka-ui 0.7.1

1.6. UI/UX

- Figma

1.7. IDE

- Visual Studio Code 1.85.1
- IntelliJ IDEA 2023.2

1.8. 형상 / 이슈관리

- Gitlab 16.7.3
- Gerrit 3.8.1
- Jira

1.9. 기타 Tool

- Postman 10.23.1

2. EC2 세팅

2.1 EC2 Port

내용	Port	Port (in Docker)
SSH	22	-
HTTP (HTTPS로 redirect)	80	-
HTTPS	443	-
Frontend (Nginx + React)	3000, 3001	80
MySQL	7777	3306
Backend (API)	8080, 8081	8080
Backend (Notification)	8180, 8181	8080
kafka-ui	8083	8080
Gerrit	8989	-
Jenkins	9000	8080

2.2. 방화벽(UFW) 설정

```
# 1. 해당 포트 개방
# 22 TCP
# 80 TCP
# 443 TCP
# 3000, 3031 TCP
# 7777 TCP
# 8080, 8081 TCP
# 8180, 8181 TCP
# 8083 TCP
# 8989 TCP
# 9000 TCP
# 예시
sudo ufw allow 22/TCP

# 2. UFW 활성화 및 상태 확인
sudo ufw enable
sudo ufw status verbose

# 3. Nginx reverse proxy 설정 후 불필요한 포트 닫기
sudo ufw deny 3000, 3031/TCP  # Frontend
sudo ufw deny 7777/TCP        # MySQL
sudo ufw deny 8080, 8081/TCP  # Backend API
sudo ufw deny 8180, 8181/TCP  # Backend Notification
sudo ufw deny 8083/TCP        # kafka-ui
sudo ufw deny 9000/TCP        # Jenkins
```

2.3. Nginx 설치

```
# 1. Nginx 설치
sudo apt-get install nginx
nginx -v

# 2. Let's Encrypt 설치 및 SSL 발급
sudo apt-get install letsencrpyt
sudo systemctl stop nginx
sudo letsencrypt certonly --standalone -d {도메인명}

# 3. Nginx 설정파일 생성
cd /etc/nginx/conf.d
vim application.conf
############

server {
        include /etc/nginx/conf.d/service-url.inc;
        include /etc/nginx/conf.d/api-url.inc;

        server_name i10b305.p.ssafy.io;

        location / {
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host $http_host;
                proxy_set_header X-Forwarded-Proto $scheme;

                proxy_pass $service_url;
        }

        location /api {
                proxy_pass $api_url;
        }


        listen 443 ssl;
        ssl_certificate /etc/letsencrypt/live/i10b305.p.ssafy.io/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/i10b305.p.ssafy.io/privkey.pem;

}

server {
        if ($host = i10b305.p.ssafy.io){
                return 301 https://$host$request_uri;
        }

        listen 80;
        server_name i10b305.p.ssafy.io;
        return 404;

}


upstream service {
        server 127.0.0.1:3000;
        server 127.0.0.1:3001;
}

server {
        listen 80;
        server_name www.silvstone.xyz;
        return 301 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name www.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/silvstone.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/silvstone.xyz/privkey.pem;

        location / {
                proxy_pass http://service;
        }
}


upstream api {
        server 127.0.0.1:8080;
        server 127.0.0.1:8081;
}

server {
        listen 80;
        server_name api.silvstone.xyz;
        return 308 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name api.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/silvstone.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/silvstone.xyz/privkey.pem;

        location / {
                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;

                proxy_pass http://api;
        }
}


upstream jenkins {
       server 127.0.0.1:9000;
}

server {
        listen 80;
        server_name jenkins.silvstone.xyz;
        return 308 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name jenkins.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/silvstone.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/silvstone.xyz/privkey.pem;

        location / {
                proxy_pass http://jenkins;
        }
}


upstream gerrit {
        server 127.0.0.1:8989;
}

server {
        listen 80;
        server_name gerrit.silvstone.xyz;
        return 308 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name gerrit.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/silvstone.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/silvstone.xyz/privkey.pem;

        location / {
                proxy_pass https://gerrit;
        }
```

```
}


upstream notification {
        server 127.0.0.1:8180;
        server 127.0.0.1:8181;
}

server {
        listen 80;
        server_name notification.silvstone.xyz;
        return 301 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name notification.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/notification.silvstone.xyz/fullchain
        ssl_certificate_key /etc/letsencrypt/live/notification.silvstone.xyz/privk

        location / {
                proxy_set_header Connection '';
                proxy_http_version 1.1;

                proxy_set_header Cache-Control 'no-cache';
                proxy_set_header X-Accel-Buffering 'no';
                proxy_buffering off;
                chunked_transfer_encoding on;
                proxy_read_timeout 86400s;

                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host $http_host;

                proxy_pass http://notification;
        }
}

upstream kafka-ui {
        server 127.0.0.1:8083;
}

server {
        listen 80;
        server_name kafka-ui.silvstone.xyz;
        return 308 https://$host$request_uri;
}

server {
        listen 443 ssl;
        server_name kafka-ui.silvstone.xyz;
        ssl_certificate /etc/letsencrypt/live/notification.silvstone.xyz/fullchain
        ssl_certificate_key /etc/letsencrypt/live/notification.silvstone.xyz/privk

        location / {
                proxy_pass http://kafka-ui;
        }
}
############

sudo nginx -t

sudo systemctl restart nginx
```

2.4. Docker 설치

```
### install-docker.sh

# 1. Uninstall all conflicting packages
for pkg in docker.io docker-doc docker-compose docker-compose-v2 \
        podman-docker containerd runc; do sudo apt-get remove $pkg; done

# 2. Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o \
        /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# 3. Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
    https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

# 4. Install
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin \
    docker-compose-plugin
```

2.5. MySQL, Jenkins 설치 (Docker)

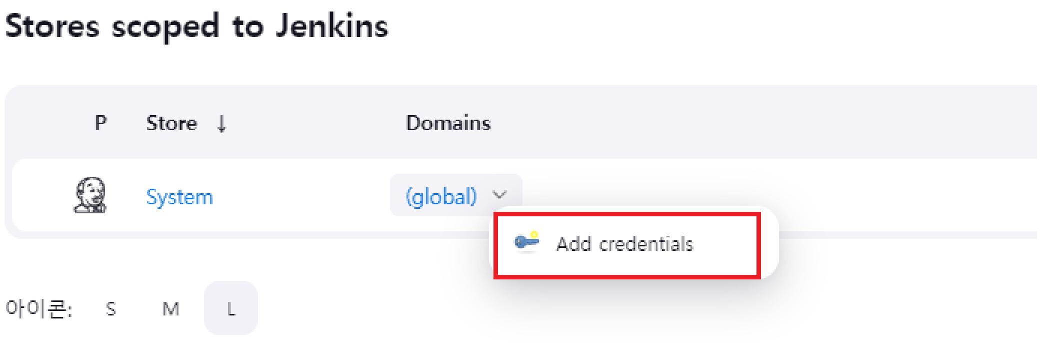
```
############

### docker-compose.init.yaml
# docker compose file for MySQL, Jenkins

services:
    jenkins:
        container_name: jenkins
        image: jenkins/jenkins:lts-jdk17
        restart: always
        ports:
          - 9000:8080
        volumes:
          - /home/ubuntu/vm1/jenkins/jenkins_home:/var/jenkins_home
          - /var/run/docker.sock:/var/run/docker.sock
        networks:
          - my-network

    mysql:
        container_name: mysql
        image: mysql:8.0.35
        restart: always
        ports:
          - 7777:3306
        environment:
          MYSQL_ROOT_PASSWORD: 1234
          TZ: Asia/Seoul
        volumes:
          - /home/ubuntu/vm1/mysql:/var/lib/mysql
        networks:
          - my-network


networks:
    my-network:
        name: my-network

############

docker compose -f docker-compose.init.yaml up -d
```

3. CI/CD 구축

3.1. Jenkins 설정

3.1.1. GitLab Credentials 설정

1. 좌측 메뉴 “Jenkins 관리” 클릭
2. Security → Credentials 클릭
3. “Store : System” → “(global)” → “Add vredentials” 클릭



Stores scoped to Jenkins

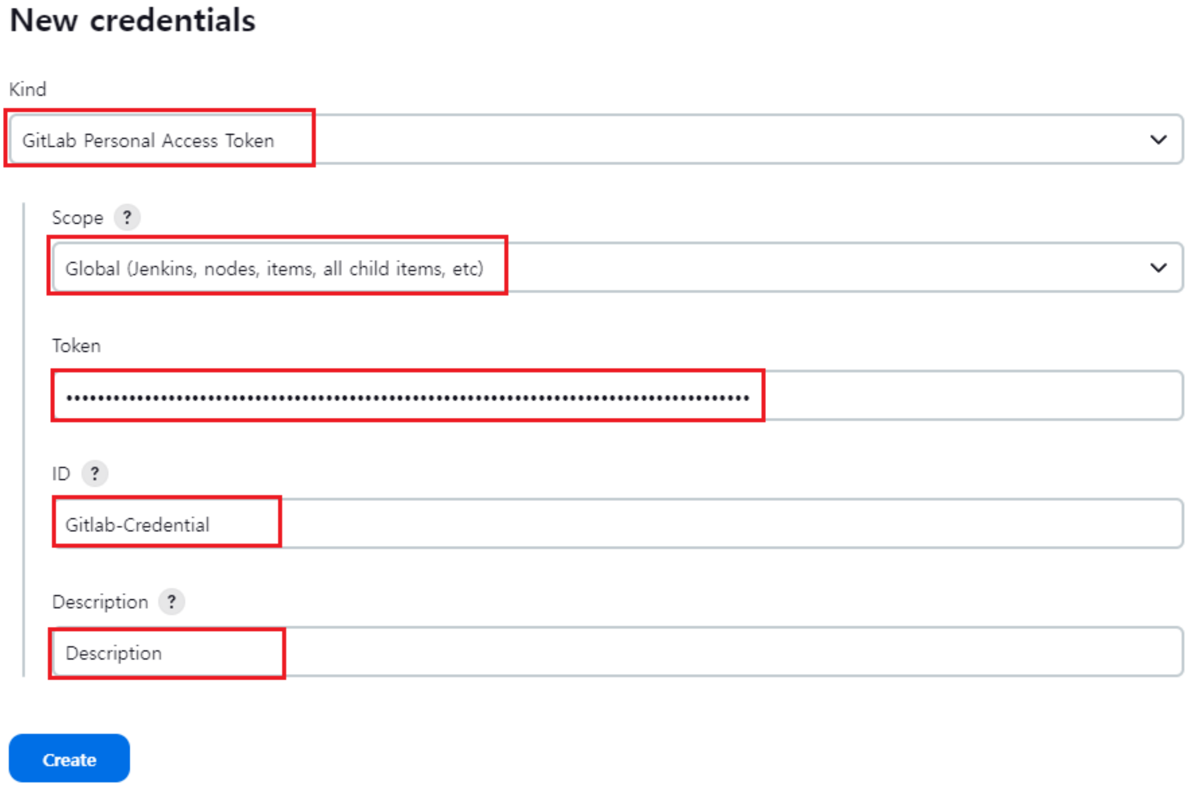
P	Store ↓	Domains
	System	(global)

Add credentials

아이콘: S M L

4. “Kind”에 “GitLab Personal Access Token” 입력 → “Scope”에 “Global” 입력 → “Token”에 Gitlab Personal Acess Token 입력 → “ID”에 임의의 아이디 입력 → 생성

   *** Personal Access Token은 Gitlab > User Setings > Access Tokens 에서 생성



New credentials

Kind: GitLab Personal Access Token

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

Token: ••

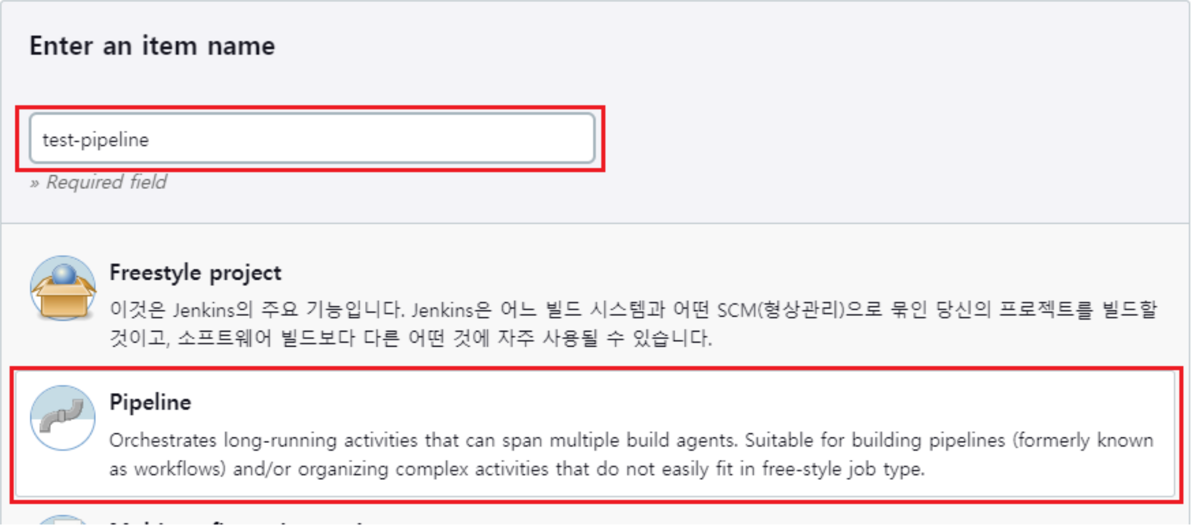
ID: Gitlab-Credential

Description: Description

Create

3.1.2. Jenkins Item 생성

1. 좌측 메뉴 “새로운 Item” 클릭
2. “Enter an item name”에 임의의 Item 이름 입력 → “Pipeline” 선택 → 생성



Enter an item name

test-pipeline

» Required field

Freestyle project

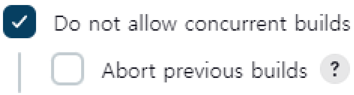
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.

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.

3. “General” → “Do not allow concurrent builds” 클릭

   (빌드가 진행중이면 동시에 빌드를 진행하지 않게 한다)



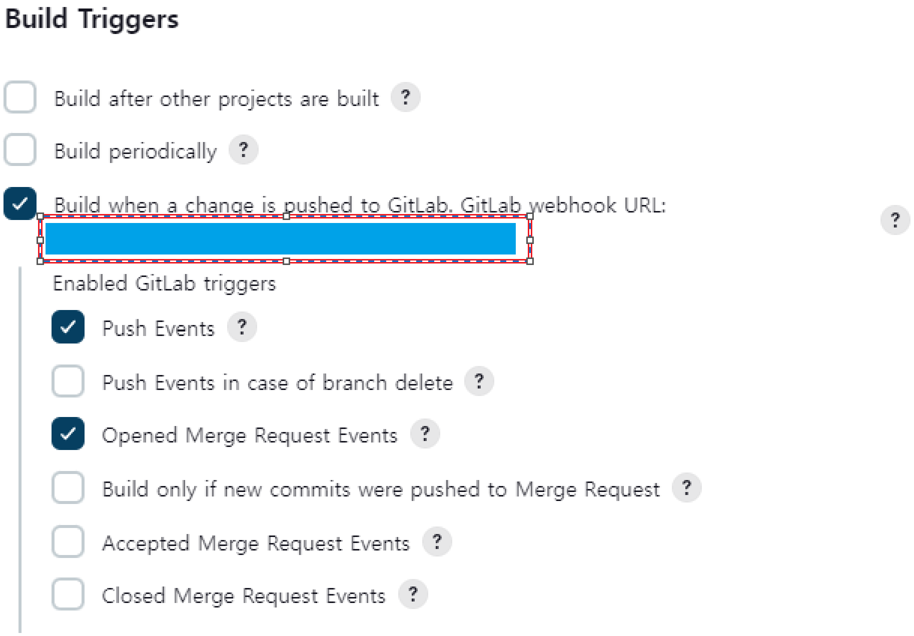
☑ Do not allow concurrent builds

□ Abort previous builds ?

4. “Build Triggers” → “Build when a change is pushed to GitLab” 클릭

   (Webhook 설정 : GitLab 특정 브랜치 push 시 자동 빌드 + 배포 설정)

   (해당 URL 복사 → Webhook 설정 시 사용)



Build Triggers

□ Build after other projects are built ?

□ Build periodically ?

☑ Build when a change is pushed to GitLab. GitLab webhook URL: [redacted] ?

Enabled GitLab triggers

☑ Push Events ?

□ Push Events in case of branch delete ?

☑ Opened Merge Request Events ?

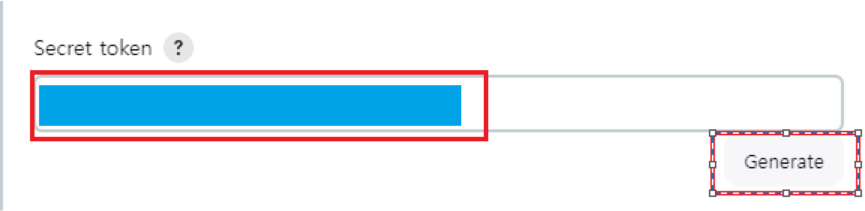
□ Build only if new commits were pushed to Merge Request ?

□ Accepted Merge Request Events ?

□ Closed Merge Request Events ?

5. “Build when a change is pushed to GitLab” 하위의 “고급…” 클릭

6. “Secret token”의 “Generate” 클릭 후 생성된 토큰 값 복사 (Webhook 설정 시 사용)

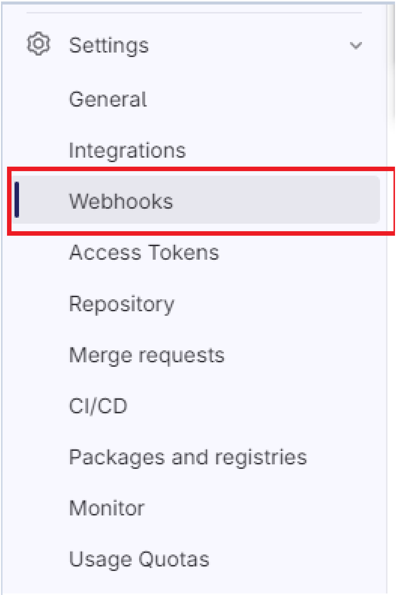


Secret token ?

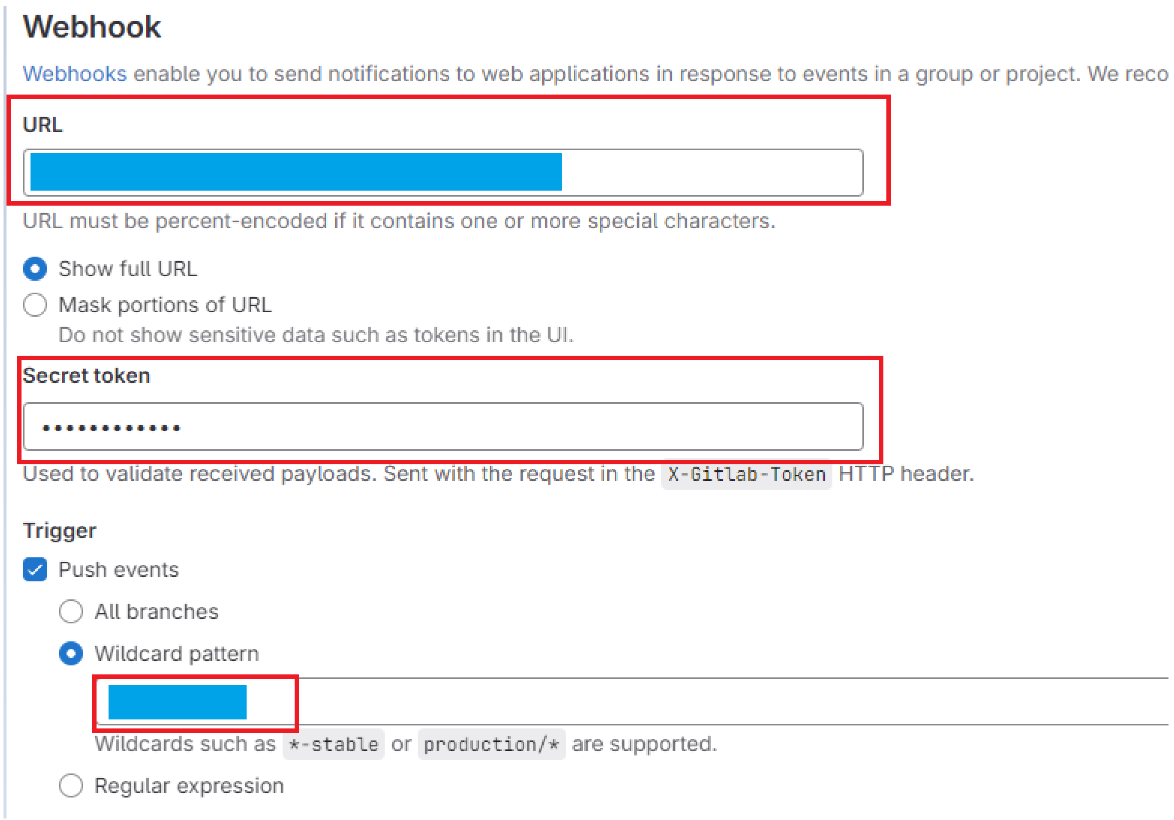
[redacted]

3.1.3. Gitlab Webhook 설정

1. 프로젝트 GitLab → “Settings” → “Webhooks” 클릭



2. “URL”에 사전에 복사해놓은 Jenkins URL 입력 → “Secret token”에 서전에 복사해놓은 Secret token 입력 → “Push events” 클릭 후 Webhook을 적용할 브랜치 입력



Webhook

Webhooks enable you to send notifications to web applications in response to events in a group or project. We reco

URL

[redacted]

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

[redacted]

Wildcards such as `*-stable` or `production/*` are supported.

○ Regular expression

3.1.4. Pipeline Script 작성

```
### Pipeline Script 예시
pipeline {
    agent any

    stages {
        stage('Pull') {
            steps {
                checkout scmGit(branches: [[name: '*/release-be']], extensions: []

            }
            post {
                success {
                    sh 'echo "Successfully Cloned Repository"'
                }
                failure {
                    sh 'echo "Fail Cloned Repository"'
                }
            }
        }

        stage('Build'){
            steps{
                dir('backend'){
                    sh '''
                        chmod +x gradlew
                        ./gradlew clean build
                    '''
                }
            }

            post {
                success {
                    sh 'echo "Build Success"'
                }
                failure {
                    sh 'echo "Build Fail"'
                }
            }
        }

        stage('Build Docker Image'){
            steps{
                dir('backend'){
                    sh '''
                        sh build-image.sh
                    '''
                }
            }

            post {
                success {
                    sh 'echo "Build Image Success"'
                }
                failure {
                    sh 'echo "Build Image Fail"'
                }
            }
        }

        stage('Deploy'){
            steps{
                dir('backend/cicd'){
                    sh '''
                        sh deploy.sh
                    '''
                }
            }

            post {
                success {
                    script {
                        def Author_ID = sh(script: "git show -s --pretty=%an", ret
                        def Author_Name = sh(script: "git show -s --pretty=%ae", r
                        mattermostSend (color: 'good',
                        message: "빌드 성공: ${env.JOB_NAME} #${env.BUILD_NUMBER} by
                        endpoint: 'https://meeting.ssafy.com/hooks/fddiaf73ebrwpfy
                        )
                    }
                }
                failure {
                    script {
                        def Author_ID = sh(script: "git show -s --pretty=%an", ret
                        def Author_Name = sh(script: "git show -s --pretty=%ae", r
                        mattermostSend (color: 'danger',
                        message: "빌드 실패: ${env.JOB_NAME} #${env.BUILD_NUMBER} by
                        endpoint: 'https://meeting.ssafy.com/hooks/fddiaf73ebrwpfy
                        )
                    }
                }
            }
        }
    }
}
```

3.2. DooD (Docker out of Docker) 설정

> Docker 컨테이너 내부에서 컨테이너 외부에 존재하는 Docker host의 기능을 빌려서 사용하기 위한 설정, Jenkins 컨테이너 내부에서 아래의 명령어를 실행합니다.

```
# 호스트의 docker 그룹 ID(998)로 지정된 그룹명을 docker로 변경
groupmod -g 998 docker

# 컨테이너 내부 사용자가 호스트의 파일에 접근 가능하도록 호스트 사용자 아이디(1000)로 변경
usermod -u 1000 jenkins

apt-get update
apt-get -y install lsb-release apt-transport-https ca-certificates \
    curl gnupg2 software-properties-common
apt-get -y upgrade apt-transport-https

# Docker 다운로드 링크 추가
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg
    > /tmp/dkey
apt-key add /tmp/dkey

# Docker Repository 등록
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os

# Docker 설치
apt-get update
apt-get -y install docker-ce docker-ce-cli containerd.io
```

3.3 빌드 및 배포 과정

> Option 1. 상기 Webhook 설정한 Branch로 Push

> Option 2. Jenkins 홈 화면 → Jenkins Item 클릭 → “지금 빌드” 클릭

4. Kafka 클러스터 설정 (Single-Broker)

4.1. Kafka 클러스터 설치 (Docker)

```
############

### docker-compose.kafka.yaml
# docker compose file for kafka cluster

version: '3'
services:
  zookeeper:
    image: wurstmeister/zookeeper
    container_name: zookeeper
    ports:
      - "2181:2181"
    volumes:
      - /home/ubuntu/vm1/zookeeper:/home/src

  kafka:
    image: wurstmeister/kafka
    container_name: kafka
    ports:
      - "9092:9092"
    environment:
      KAFKA_ADVERTISED_LISTENERS: INSIDE://:29092,OUTSIDE://13.124.225.184:9092
      KAFKA_LISTENERS: INSIDE://:29092,OUTSIDE://0.0.0.0:9092
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INSIDE:PLAINTEXT,OUTSIDE:PLAINTEXT
      KAFKA_INTER_BROKER_LISTENER_NAME: INSIDE
      KAFKA_CREATE_TOPICS: "test:1:1"
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock\
      - /home/ubuntu/vm1/kafka:/home/src
    depends_on:
      - zookeeper

    kafka-ui:
    image: provectuslabs/kafka-ui
    container_name: kafka-ui
    ports:
      - "8083:8080"
    restart: always
    environment:
      - KAFKA_CLUSTERS_0_NAME=local
      - KAFKA_CLUSTERS_0_BOOTSTRAPSERVERS=kafka:29092
      - KAFKA_CLUSTERS_0_ZOOKEEPER=zookeeper:2181

############

docker compose -f docker-compose.kafka.yaml up -d
```

5. 외부 서비스

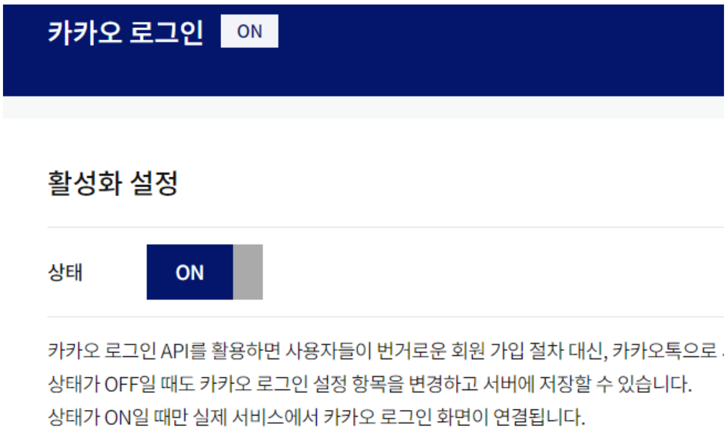
5.1. 소셜 로그인 - 카카오

> ### 카카오 로그인 REST API

https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api

5.1.1 애플리케이션 생성

1. Kakao developer 에서 애플리케이션 추가
2. 카카오 로그인 활성화



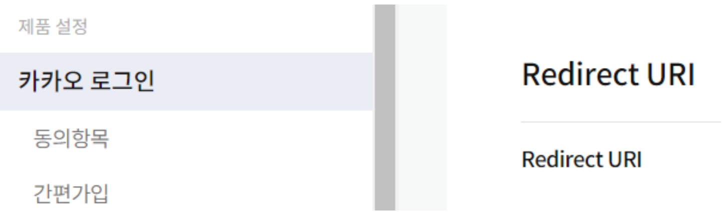
상태 ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로

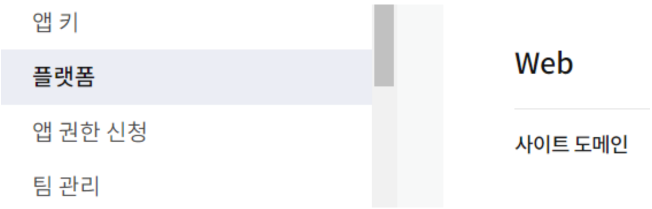
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.

상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

3. Redirect URI 등록

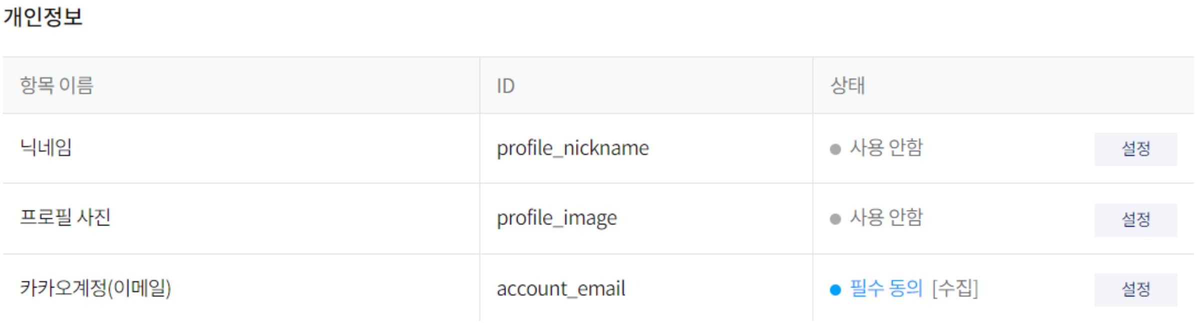


1. 플랫폼 - Web - 사이트 도메인 등록



5.1.2 동의항목

- **카카오계정(이메일) - 필수 동의**



개인정보

항목 이름	ID	상태	
닉네임	profile_nickname	● 사용 안함	설정
프로필 사진	profile_image	● 사용 안함	설정
카카오계정(이메일)	account_email	● 필수 동의 [수집]	설정

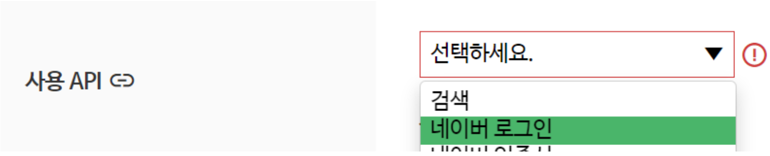
5.2. 소셜로그인 - 네이버

네이버 로그인 api

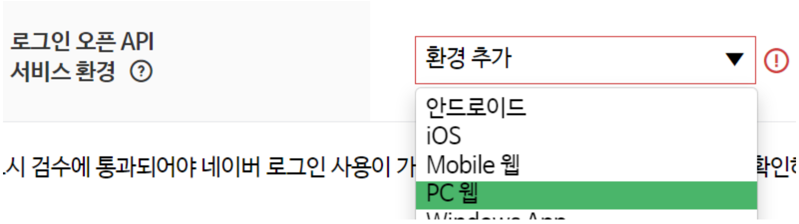
https://developers.naver.com/docs/login/devguide/devguide.md

5.2.1 애플리케이션 생성

1. Naver developers 에서 애플리케이션 추가
2. 사용 API - 네이버로그인 선택



3. 로그인 오픈 API 서비스 환경 - PC 웹

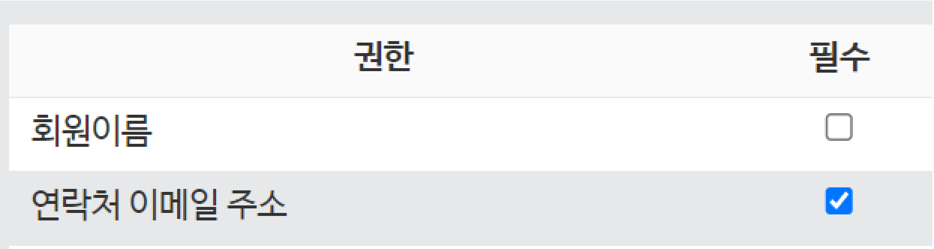


.시 검수에 통과되어야 네이버 로그인 사용이 가 … 확인

4. 서비스 URL 추가
5. 로그인 Callback URL 추가

5.2.1 동의항목

- 연락처 이메일 주소 (필수)



권한	필수
회원이름	□
연락처 이메일 주소	☑

6. DB 덤프 파일

```
-- MySQL dump 10.13  Distrib 8.0.34, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: kdkd
-- ------------------------------------------------------
-- Server version	8.0.35

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!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_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `child`
--

DROP TABLE IF EXISTS `child`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `child` (
  `coin` int DEFAULT NULL,
  `fund_money` int DEFAULT NULL,
  `child_id` bigint NOT NULL,
  PRIMARY KEY (`child_id`),
  CONSTRAINT `FKan809c5ywbh5hj4eqpuhadrcy` FOREIGN KEY (`child_id`) REFERENCES `pr
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `child`
--

LOCK TABLES `child` WRITE;
/*!40000 ALTER TABLE `child` DISABLE KEYS */;
/*!40000 ALTER TABLE `child` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `deposit`
--

DROP TABLE IF EXISTS `deposit`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `deposit` (
  `amount` int DEFAULT NULL,
  `money` int DEFAULT NULL,
  `type` bit(1) DEFAULT NULL,
  `child_id` bigint DEFAULT NULL,
  `data_log` datetime(6) DEFAULT NULL,
  `deposit_id` bigint NOT NULL AUTO_INCREMENT,
  `detail` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`deposit_id`),
  KEY `FKemq7ymh390w81ar5ih77odkvb` (`child_id`),
  CONSTRAINT `FKemq7ymh390w81ar5ih77odkvb` FOREIGN KEY (`child_id`) REFERENCES `ch
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `deposit`
--

LOCK TABLES `deposit` WRITE;
/*!40000 ALTER TABLE `deposit` DISABLE KEYS */;
/*!40000 ALTER TABLE `deposit` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `education`
--

DROP TABLE IF EXISTS `education`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `education` (
  `education_id` bigint NOT NULL AUTO_INCREMENT,
  `category` enum('SAVING','FUND','PROPERTY','TAX','SUPPLYANDDEMAND','PRODUCTIONAN
  `content` longtext,
  PRIMARY KEY (`education_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `education`
--

LOCK TABLES `education` WRITE;
/*!40000 ALTER TABLE `education` DISABLE KEYS */;
/*!40000 ALTER TABLE `education` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `fund`
--

DROP TABLE IF EXISTS `fund`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `fund` (
  `yield` int DEFAULT NULL,
  `fund_id` bigint NOT NULL,
  `content` varchar(255) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`fund_id`),
  CONSTRAINT `FKn8vp2sun3s2u33fci2bc03lh5` FOREIGN KEY (`fund_id`) REFERENCES `chi
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `fund`
--

LOCK TABLES `fund` WRITE;
/*!40000 ALTER TABLE `fund` DISABLE KEYS */;
/*!40000 ALTER TABLE `fund` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `fund_history`
--

DROP TABLE IF EXISTS `fund_history`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `fund_history` (
  `pnl` int DEFAULT NULL,
  `seed_money` int DEFAULT NULL,
  `yield` int DEFAULT NULL,
  `child_id` bigint DEFAULT NULL,
  `data_log` datetime(6) DEFAULT NULL,
  `fund_history_id` bigint NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`fund_history_id`),
  KEY `FK5qi3ggm3eoii2pwkwirs4flqx` (`child_id`),
  CONSTRAINT `FK5qi3ggm3eoii2pwkwirs4flqx` FOREIGN KEY (`child_id`) REFERENCES `ch
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `fund_history`
--

LOCK TABLES `fund_history` WRITE;
/*!40000 ALTER TABLE `fund_history` DISABLE KEYS */;
/*!40000 ALTER TABLE `fund_history` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `fund_news`
--

DROP TABLE IF EXISTS `fund_news`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `fund_news` (
  `child_id` bigint DEFAULT NULL,
  `data_log` datetime(6) DEFAULT NULL,
  `fund_news_id` bigint NOT NULL AUTO_INCREMENT,
  `content` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`fund_news_id`),
  KEY `FK3r8kharr0pgia4uoom3y0fkay` (`child_id`),
  CONSTRAINT `FK3r8kharr0pgia4uoom3y0fkay` FOREIGN KEY (`child_id`) REFERENCES `ch
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `fund_news`
--

LOCK TABLES `fund_news` WRITE;
/*!40000 ALTER TABLE `fund_news` DISABLE KEYS */;
/*!40000 ALTER TABLE `fund_news` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `fund_reservation`
--

DROP TABLE IF EXISTS `fund_reservation`;
```

```
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `fund_reservation` (
  `state` bit(1) DEFAULT NULL,
  `yield` int DEFAULT NULL,
  `fund_reservation_id` bigint NOT NULL,
  `content` varchar(255) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`fund_reservation_id`),
  CONSTRAINT `FKm529p0574xjuixf5n3ls56f8k` FOREIGN KEY (`fund_reservation_id`) REF
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `fund_reservation`
--

LOCK TABLES `fund_reservation` WRITE;
/*!40000 ALTER TABLE `fund_reservation` DISABLE KEYS */;
/*!40000 ALTER TABLE `fund_reservation` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `fund_status`
--

DROP TABLE IF EXISTS `fund_status`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `fund_status` (
  `amount` int DEFAULT NULL,
  `answer` bit(1) DEFAULT NULL,
  `submit` bit(1) DEFAULT NULL,
  `fund_status_id` bigint NOT NULL,
  PRIMARY KEY (`fund_status_id`),
  CONSTRAINT `FKrutshwi1fx356arjdyl2mwbdc` FOREIGN KEY (`fund_status_id`) REFERENC
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `fund_status`
--

LOCK TABLES `fund_status` WRITE;
/*!40000 ALTER TABLE `fund_status` DISABLE KEYS */;
/*!40000 ALTER TABLE `fund_status` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `job`
--
```

```
DROP TABLE IF EXISTS `job`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `job` (
  `done_count` int DEFAULT NULL,
  `task_amount` int DEFAULT NULL,
  `wage` int DEFAULT NULL,
  `job_id` bigint NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `task` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`job_id`),
  CONSTRAINT `FKfa5l8t6yojmhla9gexxjusp1w` FOREIGN KEY (`job_id`) REFERENCES `chil
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `job`
--

LOCK TABLES `job` WRITE;
/*!40000 ALTER TABLE `job` DISABLE KEYS */;
/*!40000 ALTER TABLE `job` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `job_reservation`
--

DROP TABLE IF EXISTS `job_reservation`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `job_reservation` (
  `state` bit(1) DEFAULT NULL,
  `task_amount` int DEFAULT NULL,
  `wage` int DEFAULT NULL,
  `job_reservation_id` bigint NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `task` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`job_reservation_id`),
  CONSTRAINT `FKir7y2gnpu0p4c8n9wifbtixn4` FOREIGN KEY (`job_reservation_id`) REFE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `job_reservation`
--

LOCK TABLES `job_reservation` WRITE;
/*!40000 ALTER TABLE `job_reservation` DISABLE KEYS */;
/*!40000 ALTER TABLE `job_reservation` ENABLE KEYS */;
```

```
UNLOCK TABLES;

--
-- Table structure for table `parent`
--

DROP TABLE IF EXISTS `parent`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `parent` (
  `parent_id` bigint NOT NULL,
  PRIMARY KEY (`parent_id`),
  CONSTRAINT `FK3inlojyp6ixcimtn040bun4gn` FOREIGN KEY (`parent_id`) REFERENCES `p
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `parent`
--

LOCK TABLES `parent` WRITE;
/*!40000 ALTER TABLE `parent` DISABLE KEYS */;
/*!40000 ALTER TABLE `parent` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `profile`
--

DROP TABLE IF EXISTS `profile`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `profile` (
  `pin` int DEFAULT NULL,
  `type` bit(1) DEFAULT NULL,
  `profile_id` bigint NOT NULL,
  `user_id` bigint DEFAULT NULL,
  `nickname` varchar(255) DEFAULT NULL,
  `profile_image` tinytext,
  PRIMARY KEY (`profile_id`),
  KEY `FKawh070wpue34wqvytjqr4hj5e` (`user_id`),
  CONSTRAINT `FKawh070wpue34wqvytjqr4hj5e` FOREIGN KEY (`user_id`) REFERENCES `use
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `profile`
--

LOCK TABLES `profile` WRITE;
/*!40000 ALTER TABLE `profile` DISABLE KEYS */;
```

```
/*!40000 ALTER TABLE `profile` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `profile_seq`
--

DROP TABLE IF EXISTS `profile_seq`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `profile_seq` (
  `next_val` bigint DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `profile_seq`
--

LOCK TABLES `profile_seq` WRITE;
/*!40000 ALTER TABLE `profile_seq` DISABLE KEYS */;
INSERT INTO `profile_seq` VALUES (1);
/*!40000 ALTER TABLE `profile_seq` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `quiz`
--

DROP TABLE IF EXISTS `quiz`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `quiz` (
  `answer` bit(1) DEFAULT NULL,
  `quiz_id` bigint NOT NULL AUTO_INCREMENT,
  `category` enum('SAVING','FUND','PROPERTY','TAX','SUPPLYANDDEMAND','PRODUCTIONAN
  `question` tinytext,
  PRIMARY KEY (`quiz_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `quiz`
--

LOCK TABLES `quiz` WRITE;
/*!40000 ALTER TABLE `quiz` DISABLE KEYS */;
/*!40000 ALTER TABLE `quiz` ENABLE KEYS */;
UNLOCK TABLES;

--
```

```
-- Table structure for table `roi`
--

DROP TABLE IF EXISTS `roi`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `roi` (
  `count` int DEFAULT NULL,
  `success` int DEFAULT NULL,
  `roi_id` bigint NOT NULL,
  PRIMARY KEY (`roi_id`),
  CONSTRAINT `FKkgclbvw07g3574rcc7vggwwef` FOREIGN KEY (`roi_id`) REFERENCES `chil
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `roi`
--

LOCK TABLES `roi` WRITE;
/*!40000 ALTER TABLE `roi` DISABLE KEYS */;
/*!40000 ALTER TABLE `roi` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `saving`
--

DROP TABLE IF EXISTS `saving`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `saving` (
  `count` int DEFAULT NULL,
  `payment` int DEFAULT NULL,
  `rate` int DEFAULT NULL,
  `saving_id` bigint NOT NULL,
  `start_date` datetime(6) DEFAULT NULL,
  PRIMARY KEY (`saving_id`),
  CONSTRAINT `FK5t5ny7xeq6uwy9lqams9uqie4` FOREIGN KEY (`saving_id`) REFERENCES `c
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `saving`
--

LOCK TABLES `saving` WRITE;
/*!40000 ALTER TABLE `saving` DISABLE KEYS */;
/*!40000 ALTER TABLE `saving` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `saving_history`
--

DROP TABLE IF EXISTS `saving_history`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `saving_history` (
  `amount` int DEFAULT NULL,
  `type` bit(1) DEFAULT NULL,
  `child_id` bigint DEFAULT NULL,
  `data_log` datetime(6) DEFAULT NULL,
  `saving_history_id` bigint NOT NULL AUTO_INCREMENT,
  `detail` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`saving_history_id`),
  KEY `FK3horse8h431je6bvjgj10177m` (`child_id`),
  CONSTRAINT `FK3horse8h431je6bvjgj10177m` FOREIGN KEY (`child_id`) REFERENCES `ch
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `saving_history`
--

LOCK TABLES `saving_history` WRITE;
/*!40000 ALTER TABLE `saving_history` DISABLE KEYS */;
/*!40000 ALTER TABLE `saving_history` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `user`
--

DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user` (
  `user_id` bigint NOT NULL AUTO_INCREMENT,
  `access_token` varchar(2000) DEFAULT NULL,
  `email` varchar(2000) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user`
--

LOCK TABLES `user` WRITE;
/*!40000 ALTER TABLE `user` DISABLE KEYS */;
/*!40000 ALTER TABLE `user` 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  3:53:16
```