프로젝트에서 사용하는 외부 서비스

소셜 인증

- 카카오 로그인
  - 필수 동의항목 : 카카오계정(이메일)
  - 선택 동의항목 : 닉네임, 프로필 사진

코드 컴파일

- Docker
  - java

```
FROM openjdk:8
WORKDIR /app
ARG JAR_FILE=build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","./app.jar"]
```

- Docker Compose
  - jenkins

```
version: '3'

services:
    jenkins:
        image: jenkins/jenkins:lts
        container_name: jenkins
        volumes:
            - /var/run/docker.sock:/var/run/docker.sock
            - /jenkins:/var/jenkins_home
        ports:
            - "{포트번호}:{포트번호}"
        user: root
```

  - mysql

```
version: '3'
services:
  mysql:
    image: mysql:8.0
    container_name: mysql
    ports:
      - {포트번호}:{포트번호}
    environment:
      MYSQL_ROOT_PASSWORD: {비밀번호}
      LC_ALL: C.UTF-8
    command:
      - --lower_case_table_names=1
      - --character-set-server=utf8mb4
      - --collation-server=utf8mb4_unicode_ci
    volumes:
      - ./db/mysql/data:/var/lib/mysql
```

- Jenkins
  - Webhook을 이용하여 git lab 변화 감지
  - 파이프라인

```
pipeline {
    agent any

    stages {
        stage('Clone') {
            steps {
                git branch: 'develop', credentialsId: 'leedeok', url: 'https://lab.ssafy.com/s09-webmobile3-sub2/S09P12B102.g
            }
        }
        stage('Build') {
            steps {
                sh '''
                    cd BE/ssuk
                    echo " 8080" >> src/main/resources/port.yml
                    sed -i 's/\${ENCODE_KEY}/'${jasypt 암호화 키}'/g' src/main/resources/application.yml
                    echo "build start"
                    chmod +x ./gradlew
                    ./gradlew clean bootJar
                    sed -i 's/'${jasypt 암호화 키}'/\${ENCODE_KEY}/g' src/main/resources/application.yml
                '''
                echo "after build"
            }
        }
        stage('docker build'){
            steps{
                script{
                    echo "build start"
                    sh '''
                        cd BE/ssuk/
                        docker build -f ./Dockerfile -t ssuk_dev .
                        echo "Docker build start"
                    '''
                }
            }
        }

        stage('docker run'){
            steps{
                sh '''
                    docker ps -f name=ssuk_dev -q | xargs --no-run-if-empty docker container stop
                    docker container ls -a -f name=ssuk_dev -q | xargs -r docker container rm
                    docker images --no-trunc --all --quiet --filter="dangling=true" | xargs --no-run-if-empty docker rmi
                    docker run -d --name ssuk_dev --net host -p 8080:8080 ssuk_dev:latest
                '''
            }
        }
    }
}
```

- Amazon S3
  - 버킷 정책

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::{버킷 이름}/{파일 위치}"
        }
    ]
}
```