

프로젝트에서 사용하는 외부 서비스

구글 클라우드 버킷

- 서비스내의 카드 이미지 제공 및 형상 관리를 위한 클라우드 데이터베이스
- <https://cloud.google.com/storage?hl=ko>

코드 컴파일

Jenkins

- Web Hook을 이용한 GitLab 변화 감지
- 파이프라인

```
pipeline {
  agent any

  stages {
    stage('Checkout') {
      steps {
        // GitLab 레포지토리를 체크아웃
        git branch: 'develop', url: 'https://lab.ssafy.com/s09-fintech-finance-sub2/S09P22B208.git', credentialsId: 'dd109e40-

        echo 'checkout!!'
      }
    }

    stage('Stop and Remove Old Container') {
      steps {
        // 이전 컨테이너 정리
        script {
          sh 'docker stop bank || true'
          sh 'docker rm bank || true'
          sh 'docker rmi bank || true'
          sh 'docker stop back || true'
          sh 'docker rm back || true'
          sh 'docker rmi back || true'
          sh 'docker stop front || true'
          sh 'docker rm front || true'
          sh 'docker rmi front || true'
        }
      }
    }

    stage('Build spring boot - bank') {
      steps {

        sh 'cd bank && chmod +x gradlew && ./gradlew build'

        echo 'bank build spring boot!!'
      }
    }

    stage('Build spring boot - back') {
      steps {

        sh 'cd back && chmod +x gradlew && ./gradlew build'

        echo 'back build spring boot!!'
      }
    }

    stage('Build front - front') {
      steps {

        sh'''
        cd front
        docker build -t front .
      '''
      }
    }
  }
}
```

```

    }
}

stage('Build Docker Image BackEnd') {
    steps {
        // Docker 이미지 빌드
        sh '''
            cd bank
            pwd
            docker build -t bank .
        '''
        sh '''
            cd back
            pwd
            docker build -t back .
        '''
        echo 'Docker build!!'
    }
}

stage("Docker run"){
    steps{
        sh '''
            pwd
            ls
            docker run -d --name bank --net host -p 8081:8081 bank
            docker run -d --name back --net host -p 8080:8080 back
            docker run -d --name front --net host -p 3000:3000 front
        '''
    }
}
}
}

```

Doker

- 스프링부트, 리액트 자동 빌드를 위한 docker 파일
- Redis, MySQL, Ngnix의 균일한 자동세팅을 위한 docker-compose 파일
- **docker(Spring boot)**

```

FROM openjdk:17
WORKDIR /app
COPY ./build/libs/cocopay-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]

```

- **docker(React)**

```

FROM node:14
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]

```

- **docker-compose**

```

version: '3'

services:

```

```

jenkins:
  image: jenkins/jenkins:jdk17
  container_name: jenkins
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - /jenkins:/var/jenkins_home
  ports:
    - "9090:8080"
  user: root

mysql:
  image: mysql:8.0.34-debian
  container_name: mysql
  command: --lower_case_table_names=1
  environment:
    MYSQL_DATABASE: bank
    MYSQL_ROOT_PASSWORD: palddak
    MYSQL_USER: ssafy
    MYSQL_PASSWORD: ssafy
    TZ: Asia/Seoul

  cap_add:
    - SYS_NICE

  ports:
    - "3306:3306"
  volumes:
    - ./mysql/conf.d:/etc/mysql/conf.d

nginx:
  image: nginx
  container_name: nginx
  ports:
    - 80:80
    - 443:443
  volumes:
    - /etc/letsencrypt:/etc/letsencrypt
    - ./nginx/conf.d:/etc/nginx/conf.d

redis:
  image : redis:6.2.6
  container_name: redis
  hostname: ssafy
  command: redis-server --requirepass ssafy --port 6379

  ports:
    - "6379:6379"

```

문자인증

- Naver SENS API를 활용한 문자 인증 서비스
- back/src/main/java/com/cocopay/utills/sens/Naver_Sens_V2 중

```

String hostNameUrl = "https://sens.apigw.ntruss.com";
// 요청 URL
String requestUrl= "/sms/v2/services/";
// 요청 URL Type
String requestUrlType = "/messages";
// 개인 인증키
String accessKey = "YZ8AHwLfjR1PsF23Y2fK";
// 2차 인증을 위해 서비스마다 할당되는 service secret
String secretKey = "q6ghKhSnUAG9o0yKe0yvVo51tHGxjhEJLBQg2l7j";
// 프로젝트에 할당된 SMS 서비스 ID
String serviceId = "ncp:sms:kr:269707162992:coco";

```

Firestore

- FCM 푸시알림, 외부의 앱 접속을 위한 Dynamic Link

- back/resource/google-service.json
- back/resource/ssuk-ssuk-push-server-firebase-adminsdk-cue54-4b189a9f26.json