

버전 1.0

날짜 2022.11.21



기모치

프 인 우
티 개 기
콘

기모치 포팅 메뉴얼

PORTING MANUAL

SSAFY 7 기 서울 A205

김동욱 김다운 이성민 이승훈 전민재 최태순

기모치 포팅 메뉴얼

개발 환경

이름	버전
Server	AWS EC2 Ubuntu 22.04 LTS
JDK	11.0.17
MySQL	8.0.31-0ubuntu0.20.04.1
Jenkins	2.361.2
Nginx	1.18.0
IntelliJ IDEA	17.0.4.1+7-b469.62 amd64
Android Studio	Android SDK 30
Visual Studio Code	1.70.0
Node	16.18.0

초기 환경 설정

REACT NATIVE CLI 설치

([HTTPS://REACTNATIVE.DEV/DOCS/ENVIRONMENT-SETUP](https://reactnative.dev/docs/environment-setup))

초코 설치 (<https://chocolatey.org/>)

```
choco install -y node.js-lts openjdk11
```

ANDROID_HOME 환경변수 편집

```
%LOCALAPPDATA%\Android\Sdk
```

platform-tools

%LOCALAPPDATA%\Android\Sdk\platform-tools

react native cli 설치

npm i -g react-native

npx react-native init MyApp --template [react-native-template-typescript@6.8.9](https://github.com/react-native-community/react-native-template-typescript)

(<https://github.com/react-native-community/react-native-template-typescript>)

Android Studio 에서 Virtual Devices 설정하기 SDK 30

#앱 실행하기

npm run android # 안드로이드 실행 명령어

#앱 실행 후

cmd + R => 리로딩

cmd + D => 디버그 메뉴

젠킨스 설치

apt 업데이트

sudo apt-get update

자바 설치 (젠킨스 최신버전에서는 자바 11 이상 필요)

sudo apt-get install openjdk-11-jdk

젠킨스 저장소 키 다운로드

sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

sources.list 에 추가

sudo echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee
/etc/apt/sources.list.d/jenkins.list

키 등록

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys FCEF32E745F2C3D5
```

apt 재업데이트

```
sudo apt-get update
```

젠킨스 설치

```
sudo apt-get install Jenkins
```

젠킨스 포트 변경

```
sudo vi /lib/systemd/system/jenkins.service
```

- Environment="JENKINS_PORT=원하는 포트번호 입력"

젠킨스 재시작

```
sudo service jenkins restart
```

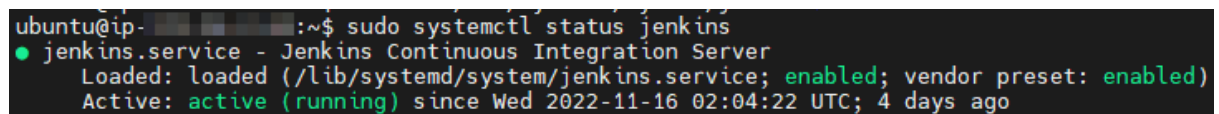
- Warning: The unit file, source configuration file or drop-ins of jenkins.service changed on disk. Run 'systemctl daemon-reload' to reload units.

해당 오류 발생 시,

```
sudo systemctl daemon-reload
```

젠킨스 상태 확인

```
sudo systemctl status Jenkins
```



```
ubuntu@ip-10-10-10-10:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-11-16 02:04:22 UTC; 4 days ago
```

젠킨스 초기 비밀번호 확인

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

젠킨스 초기 페이지 접속

도메인:포트번호 로 초기화면 접속 및 세팅 (깃랩 웹훅)

https://github.com/hjs101/CICD_manual (참고 자료)

NGINX + CERTBOT 설치 및 환경 설정

<https://velog.io/@smat91/AWS-EC2-Ubuntu%EC%97%90%EC%84%9C-Nginx%EC%9D%98-HTTPS-%EC%84%B8%ED%8C%85> (참고 자료)

Nginx 설정

```
vi /etc/nginx/sites-available/default
```

```
location /api/ {
    proxy_pass http://k7a205.p.ssafy.io:8080;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Host $server_name;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_buffer_size      128k;
    proxy_buffers           4 256k;
    proxy_busy_buffers_size 256k;
}
```

MYSQL 설치

MySQL 설치

```
sudo apt-get install mysql-server
```

MySQL 접속

```
sudo /usr/bin/mysql -u root -p
```

bind-address 설정

bind-address = 0.0.0.0 으로 변경

계정 생성 및 권한 설정

```
create user '계정명'@'% 'identified by '비밀번호';
```

```
grant all privileges on *.* to '계정명'@'% ';
```

```
flush privileges;
```

배포

젠킨스 EXECUTESHELL

```
cd /var/lib/jenkins/workspace/gimochi-be/backend # 프로젝트 경로로 이동
sudo chmod +x gradlew # gradlew 권한 설정
sudo ./gradlew clean build # 기존 빌드 클린 후 재빌드
cd /var/lib/jenkins/workspace/gimochi-be/backend/build/libs # jar 파일 위치로 이동
sudo sh /home/ubuntu/start_server.sh # 배포 스크립트 실행
```

START_SERVER.SH (배포 스크립트)

```
#!/bin/bash

# 변수 설정
REPOSITORY=/var/lib/jenkins/workspace
PROJECT_NAME=gimochi-be
JAR_NAME=ssafy-web-project-1.0-SNAPSHOT.jar

# 기존 배포 중인 서비스 중단
CURRENT_PID=$(pgrep -f $JAR_NAME)
echo "> 현재 구동 중인 애플리케이션 pid 확인: $CURRENT_PID"
if [ -z "$CURRENT_PID" ]; then
    echo "> 현재 구동 중인 애플리케이션 없음"
else
    echo "> 현재 구동 중인 애플리케이션 종료"
    sudo kill -15 $CURRENT_PID
    sleep 5
fi

echo "> jar 디렉토리로 이동"
cd $REPOSITORY/$PROJECT_NAME/backend/build/libs
```

echo "> 프로젝트 배포"

sudo nohup java -jar \$JAR_NAME 2>&1 &

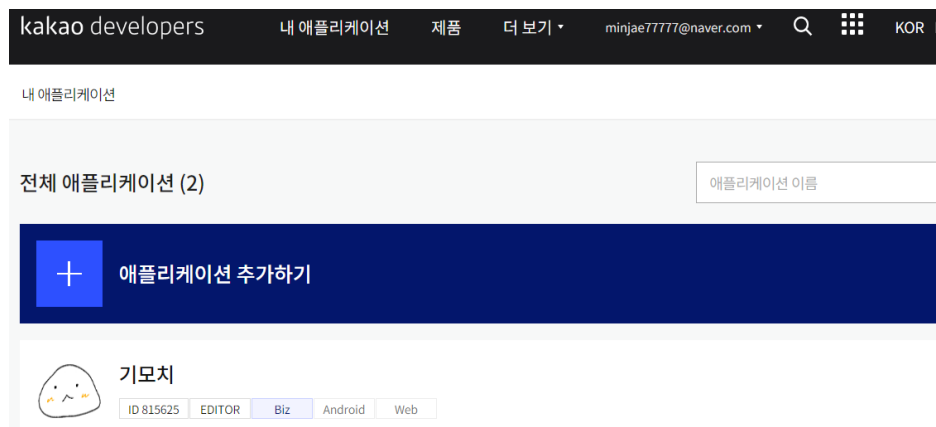
외부 서비스

KAKAO DEVELOPERS

(<https://developers.kakao.com/docs>)

카카오 소셜 로그인

1. <https://developers.kakao.com/> 에 접속해서 애플리케이션을 추가한다 (내 애플리케이션 > 애플리케이션 추가하기 > 앱 이름 입력 > 사업자명 입력 > 저장)



2. 내 애플리케이션에서 앱 키를 확인하고 기록한다 (좌측 Nav 바에서 앱 설정 > 요약 정보 > 앱 키 > REST API 키 제품 설정 > 카카오 로그인 > 보안 > Client Secret 의 코드 발급)

앱 키

네이티브 앱 키

REST API 키

JavaScript 키

Admin 키

3. 도메인을 등록(로그인 사용 시 Redirect URI)한다

Redirect URI

삭제 수정

Redirect URI	http://localhost:8081/kakao/oauth http://localhost:3000/kakao/oauth http://10.0.2.2:8081/kakao/oauth https://k7a205.p.ssafy.io/kakao/oauth http://localhost:8080/kakao/oauth https://k7a205.p.ssafy.io/api/kakao/oauth
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

↑

4. 로그인 동의 항목 구성을 설정한다. (제품 설정 > 카카오 로그인 > 동의항목에서 위치럼 설정)

☰

내 애플리케이션 > 제품 설정 > 카카오 로그인 > 동의항목

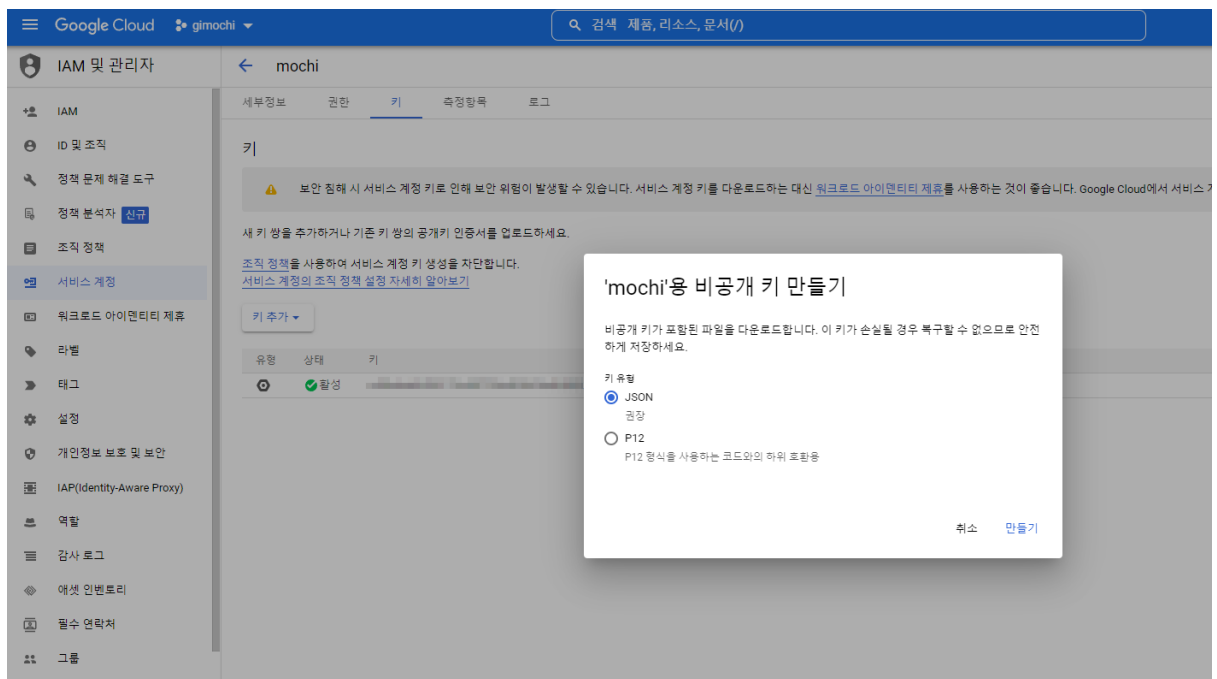
닉네임	profile_nickname	● 필수 동의	설정
프로필 사진	profile_image	● 필수 동의	설정
카카오계정(이메일)	account_email	● 선택 동의 [수집]	설정
성별	gender	● 사용 안함	설정
연령대	age_range	● 사용 안함	설정
생일	birthday	● 선택 동의 [수집]	설정
출생 연도	birthyear	○ 권한 없음	
카카오계정(전화번호)	phone_number	○ 권한 없음	
CI(연계정보)	account_ci	○ 권한 없음	
카카오 서비스 내 친구목록(프로필사진, 닉네임, 즐겨찾기 포함)	friends	● 이용 중 동의	설정
카카오톡 채널 추가 상태 및 내역	plusfriends	○ 권한 없음	

↑

GOOGLE VISION API

<https://cloud.google.com/vision/docs/detect-labels-image-client-libraries?hl=ko> (공식 문서)

1. 구글 클라우드 가입 및 결제수단 등록
2. 프로젝트 생성
3. Vision API 사용 설정
4. 서비스 계정 만들기 (IAM)
5. 서비스 계정 키 만들기



6. 환경 변수 설정

JSON 파일 다운로드 후 JSON 파일을 해당하는 OS의 환경변수로 설정

- 윈도우

시스템 환경 변수 편집 -> 환경 변수 -> 사용자 변수 새로 만들기 -> 변수 이름 : GOOGLE_APPLICATION_CREDENTIALS / 변수 값 : JSON 파일 경로

- 리눅스

빌드 및 배포 시에 sudo 명령어 사용하지 않을 경우)

export GOOGLE_APPLICATION_CREDENTIALS="JSON 파일 경로" (1 회성)

빌드 및 배포 시에 sudo 명령어 사용하는 경우)

sudo visudo 명령 실행 ->

Defaults

secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin" 해당 구문 맨 앞에 # 붙여서 주석 처리 ->

Defaults env_keep="GOOGLE_APPLICATION_CREDENTIALS" 밑에 해당 구문
추가

7. build.gradle 추가

implementation platform('com.google.cloud:libraries-bom:26.1.3')

implementation 'com.google.cloud:google-cloud-vision'

<https://cloud.google.com/vision/docs/ocr?hl=ko> (이미지 텍스트 인식 예시 문서)

AWS S3

<https://antdev.tistory.com/93> (참고 자료)

1. AWS 회원가입 및 결제수단 등록

2. S3 버킷 생성

3. IAM 계정 생성 및 정책 생성 & 키 발급

4. build.gradle 추가

implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'

5. application.properties 추가

cloud.aws.s3.bucket=버킷명

cloud.aws.credentials.access-key=엑세스키

cloud.aws.credentials.secret-key=비밀키

cloud.aws.region.static=지역

cloud.aws.region.auto=false

cloud.aws.stack.auto=false

6. config 파일 추가

@Configuration

public class S3Config {

@Value("\${cloud.aws.credentials.access-key}")

private String accessKey;

@Value("\${cloud.aws.credentials.secret-key}")

private String secretKey;

```
@Value("${cloud.aws.region.static}")
private String region;

@Bean
public AmazonS3Client amazonS3Client() {
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey,
secretKey);
    return (AmazonS3Client) AmazonS3ClientBuilder.standard()
        .withRegion(region)
        .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
        .build();
}
}
```
