

WhatSub?

A105 : WhatSub

삼성SW청년아카데미 서울캠퍼스 7기

특화 프로젝트 : 2022/08/22 ~ 2022/10/07

포팅 매뉴얼

담당 컨설턴트 : 유상진

윤혜진(팀장), 박현영, 엄윤규, 이도연, 이동환, 이현빈

<목차>

1. 기술 스택
 2. .gitignore 파일
 3. 빌드 상세내용
 4. 배포 특이사항
 5. DB 계정
 - 6 외부 서비스
-

서브웨이는 다양한 선택지를 통해 고객 개개인의 취향에 맞는 서비스를 제공해주지만, 서브웨이 주문 올령증이라는 말이 있을 정도로 젊은 사람들도 흔하게 어려움을 겪고는 합니다. 저희는 서브웨이 본사의 입장에서, 기존의 고객들은 더 다양한 조합의 샌드위치를 접할 수 있도록 하고, 새로운 고객들에게는 진입장벽을 낮춰 결과적으로는 매출을 증대시킬 수 있는 방법을 고안해 보았습니다. 소극적인 성격의 경우 기존의 비대면 주문 서비스의 편의성을 강화하고, 입맛에 맞는 조합을 추천해주는 서비스를, 식단을 관리하는 사람들의 경우 영양 정보를 기반으로 추천해주는 서비스를, 그리고 IT 서비스 활용을 어려워하는 사람들을 위해 간편 주문 서비스를 제공합니다.

1. 기술 스택

가. 이슈 관리 : Jira

나. 형상 관리 : Gitlab

다. 커뮤니케이션 : MatterMost, Notion, Webex

라. 개발 환경

1) OS : WINDOW 10

2) IDE

가) IntelliJ 22.2

나) Visual Studio Code 1.70.0

3) DataBase

가) MySQL 8.0.30

4) Server

가) AWS EC2 (Ubuntu 20.04 LTS)

나) AWS S3

마. 상세내용

1) Frontend

가) Node.js v16.16.0

나) Vue.js 2.6.14

2) Backend

가) Java 17 (jdk-17.0.4.1)

나) Spring Boot 2.7.3

다) Gradle 7.5

라) Python 3.9.10

마) Django 3.2.12

3) 기타

가) Git 2.25.1

나) Docker 20.10.18

2. .gitignore 파일

가. Frontend (Vue) : .env 파일

1) 저장 위치

[project root directory]/vue/vue.env

2) 파일 내용

```
VUE_APP_KAKAO_API_KEY="c6ff3f44296c7bc3532eb58e1697554c"  
VUE_APP_REDIRECTURI="https://j7a105.p.ssafy.io/signup"  
VUE_APP_JAVASCRIPT_KEY="444b740ed28463503b0e9ad14932d76a"  
VUE_APP_INTEGRITY="sha384-  
PFHeU/4gvSH8kpvhrigAPfZGBDPs372JceJq3jAXce11bVA6rMvGWzvP4fMQuBGL"
```

나. Backend (Spring) : account.properties 파일

1) 저장 위치

[project root directory]/spring/src/main/resources/account.properties

2) 파일 내용

```
#mysql setting
mysql.url=jdbc:mysql://j7a105.p.ssafy.io:3306/whatsub?serverTimezone=Asia/Seoul
mysql.username=whatsub
mysql.password=whatsub1229

#S3 setting
s3.region=ap-northeast-2
s3.access-key=AKIAS5SMONZEMP3BTSFZ
s3.secret-key=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
s3.bucket=whatsub
s3.bucket-url=https://whatsub.s3.ap-northeast-2.amazonaws.com
#https://s3.ap-northeast-2.amazonaws.com/whatsub

#kakao login setting
kakao.client-id=c6ff3f44296c7bc3532eb58e1697554c
kakao.jwt.secretKey=whatsub1229
```

다. Backend (Django) : my_settings.py 파일

1) 저장 위치

[project root directory]/django/my_settings.py

2) 파일 내용

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'whatsub', # 연동할 MySQL 의 데이터베이스 이름
        'USER': 'whatsub', # DB 접속 계정명
        'PASSWORD': 'whatsub1229', # 해당 DB 접속 계정 비밀번호
        'HOST': 'j7a105.p.ssafy.io', # 실제 DB 주소
        'PORT': '3306', # 포트번호
    }
}
SECRET_KEY = 'django-insecure-
397)!dwtb#c=_+73i&mjg&)ehp4z=t8+1vrfym2!1(b60eh1$z'
```

라. command 창에서 파일 작성하는 방법

1) 텍스트 에디터(vim) 열기 (sudo는 경우에 따라 빼도 됨)

```
sudo vi .env
```

2) i를 누른 후, 입력할 내용 붙여넣기(ctrl+v)

3) esc를 누른 후, :wq! 입력하고 enter

```
VUE_APP_KAKAO_API_KEY="c6ff3f44296c7bc3532eb58e1697554c"  
VUE_APP_REDIRECTURI="https://j7a105.p.ssafy.io/signup"  
VUE_APP_JAVASCRIPT_KEY="444b740ed28463503b0e9ad14932d76a"  
VUE_APP_INTEGRITY="sha384-PFHeU/4gvSH8kpvhrigAPfZGBDPs372JceJq3jAXce11bVA6rMvGW  
zvP4fMQuBGL"  
  
~  
:wq!
```

위의 세 파일들은 빌드할 때 꼭 넣어줘야 정상적으로 빌드할 수 있다.

3. 빌드 상세내용

가. Frontend 빌드하기

- 1) [project root directory]/vue 위치로 이동
- 2) node modules 설치

```
npm i
```

- 3) 빌드

```
npm build
```

나. Backend (Spring) 빌드 및 실행

- 1) [project root directory]/spring 위치로 이동
- 2) 빌드

```
./gradlew clean build
```

- 3) 생성된 jar 파일 실행

```
java -jar build/libs/*.jar
```

다. Backend (Django) 실행

```
python manage.py runserver 0.0.0.0:8000
```

4. 배포 특이사항

가. https 적용을 위한 SSL 인증서 발급

- 1) Certbot 설치 (snap은 패키지 관리 도구, 우분투에 기본으로 설치되어 있음)

```
sudo snap install certbot --classic
```

2) standalone 방식으로 인증서 발급

```
sudo certbot certonly --standalone
```

3) email 입력 → Y → Y → 도메인 이름 입력 (j7a105.p.ssafy.io) 하면 완료

나. 편리한 배포를 위한 Docker 설치

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

다. Dockerfile을 이용한 배포 (필요한 Dockerfile은 git repository에 저장되어 있음)

1) ssh를 통해 EC2 서버에 접속 후, git repository를 clone

(Dockerfile 사용을 위해 /Jenkins/workspace 디렉터리에 clone 진행)

```
sudo git clone https://lab.ssafy.com/s07-bigdata-recom-
sub2/S07P22A105.git /jenkins/workspace/whatsub
```

2) 해당 디렉터리로 이동 후, Vue, Spring, Django 도커 이미지를 각각 빌드 후 실행

가) repository 안으로 이동하기

```
cd /jenkins/workspace/whatsub/
```

나) vue + nginx 이미지 빌드 및 실행

- 빌드

```
docker build -t whatsub/vue vue
```

- 실행

```
sudo docker run -it -d -p 80:80 -p 443:443 -v
/etc/letsencrypt:/etc/letsencrypt/ -v
/jenkins/workspace/whatsub/django/staticfiles:/home/ubuntu/staticfiles
--name vue whatsub/vue
```

다) spring 이미지 빌드 및 실행

- 빌드

```
docker build -t whatsub/spring spring
```

- 실행

```
sudo docker run -it -d --rm -p 8081:8081 --name spring whatsub/spring
```

라) Django 이미지 빌드 및 실행

- 빌드

```
docker build -t whatsub/django django
```

- 실행

```
sudo docker run -it -d --rm -p 8000:8000 --name django whatsub/django
```

5. DB 계정

가. 접속 정보

1) IP주소/Host명 : j7a105.p.ssafy.io

2) root 계정 password : whatsub1229

3) user 계정

가) id : whatsub

나) password : whatsub1229

나. ssh에 MySQL 설치 및 설정

1) MySQL 설치 및 초기 설정

```
sudo apt-get update # 패키지 매니저 업데이트  
sudo apt-get install mysql-server # mysql 설치  
sudo ufw allow mysql # 외부 접속 기능 설정 (포트 3306 오픈)  
sudo systemctl start mysql # mysql 실행  
sudo systemctl enable mysql # mysql 서버 재시작 시 mysql 자동 재시작
```

2) MySQL 접속 및 root 비밀번호 변경

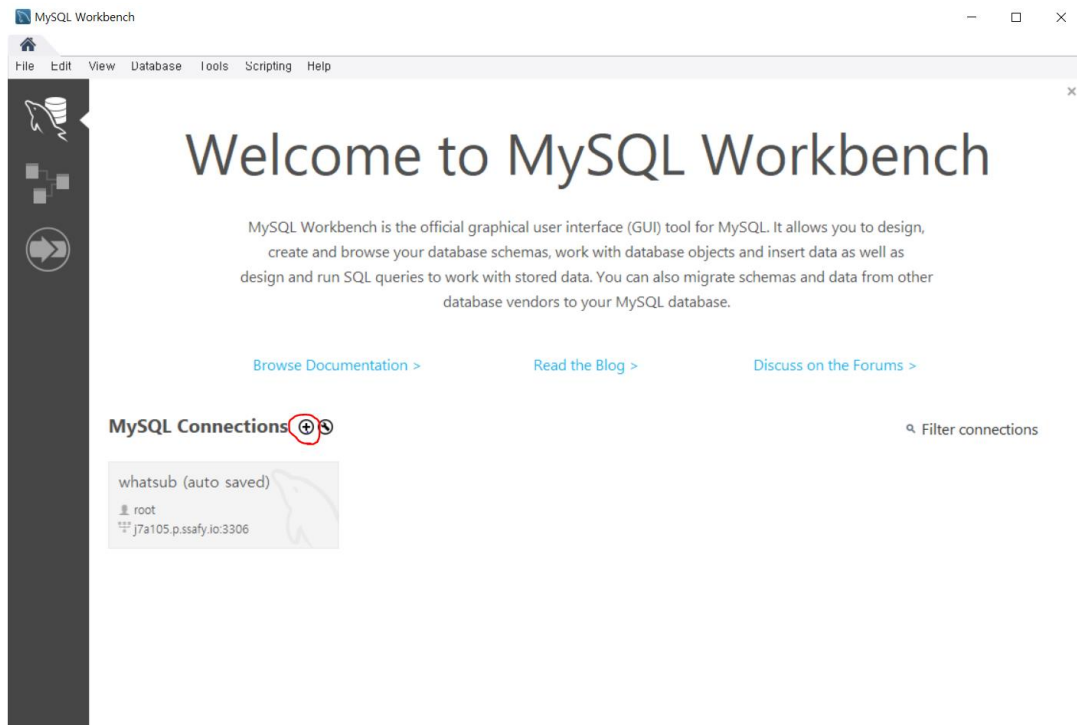
```
sudo /usr/bin/mysql -u root -p  
  
mysql> alter user 'root'@'localhost' identified with  
mysql_native_password by 'whatsub1229';
```

3) user 계정 생성 및 권한 부여

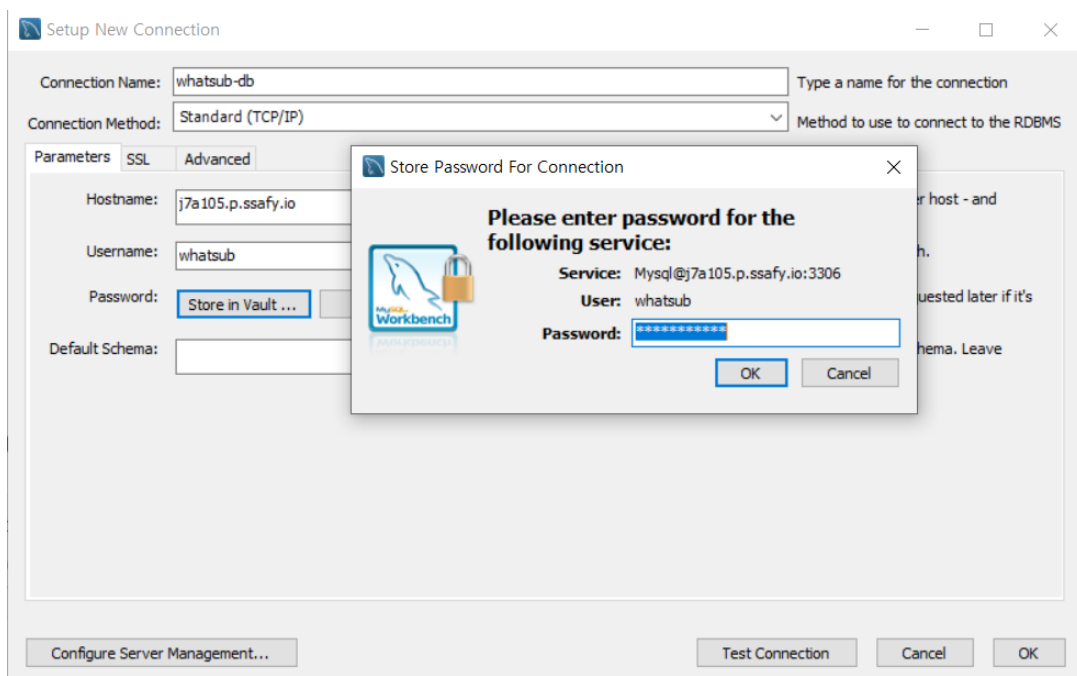
```
mysql> create user 'whatsub'@'%' identified by 'whatsub1229';  
mysql> grant all privileges on *.* to 'whatsub'@'%';
```

다. MySQL Workbench에서 접속 (로컬 컴퓨터에서)

1) MySQL Connections 옆의 + 버튼 클릭



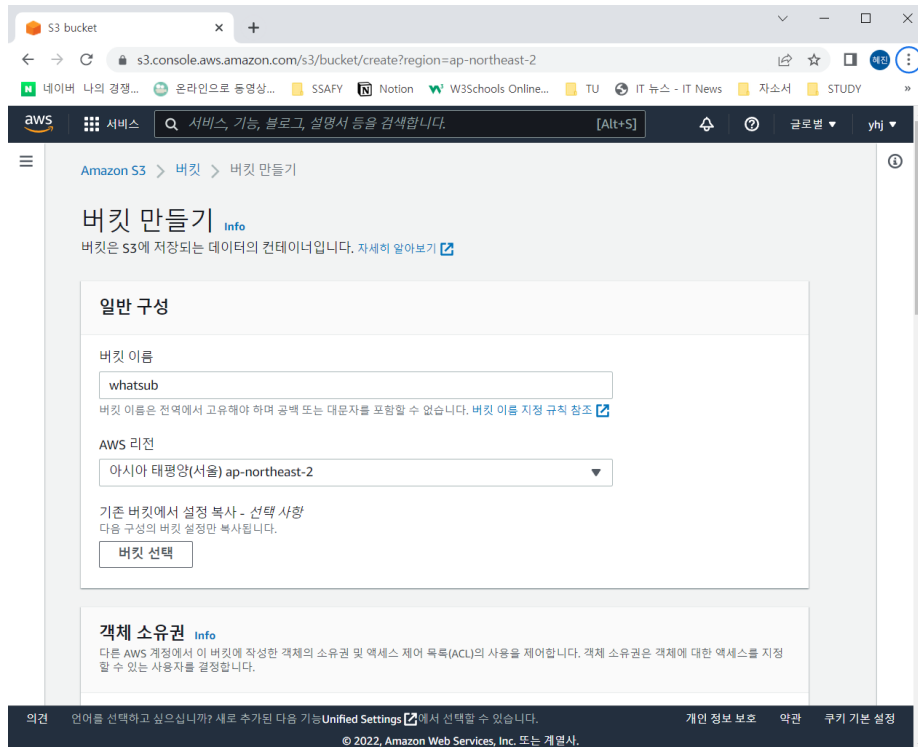
2) 가)의 접속 정보대로 입력



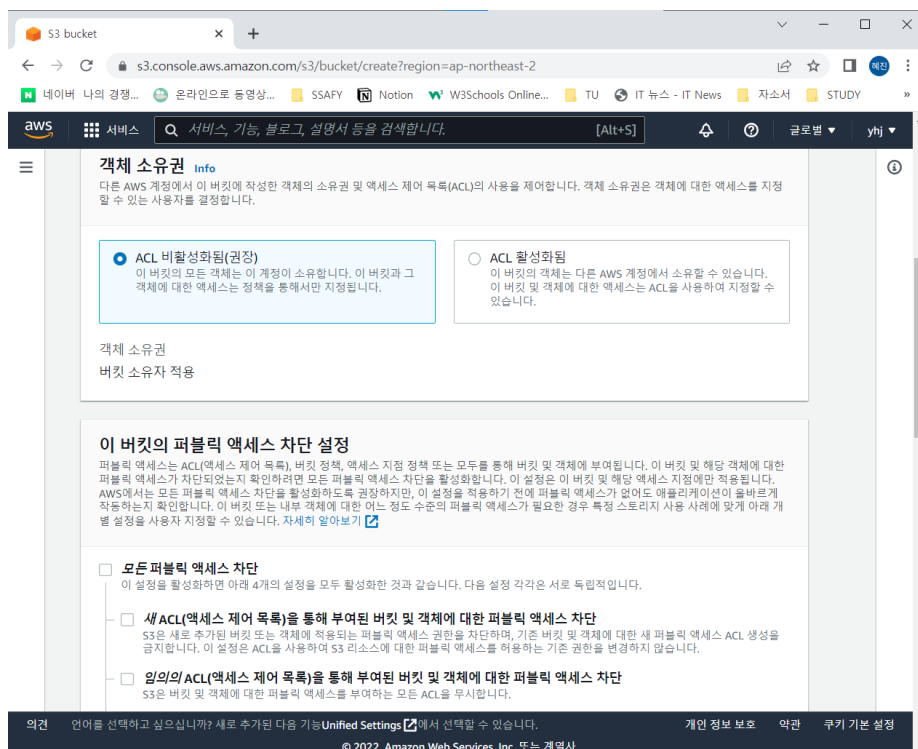
6. 외부 서비스

가. 이미지 저장을 위한 AWS S3 서버 구축

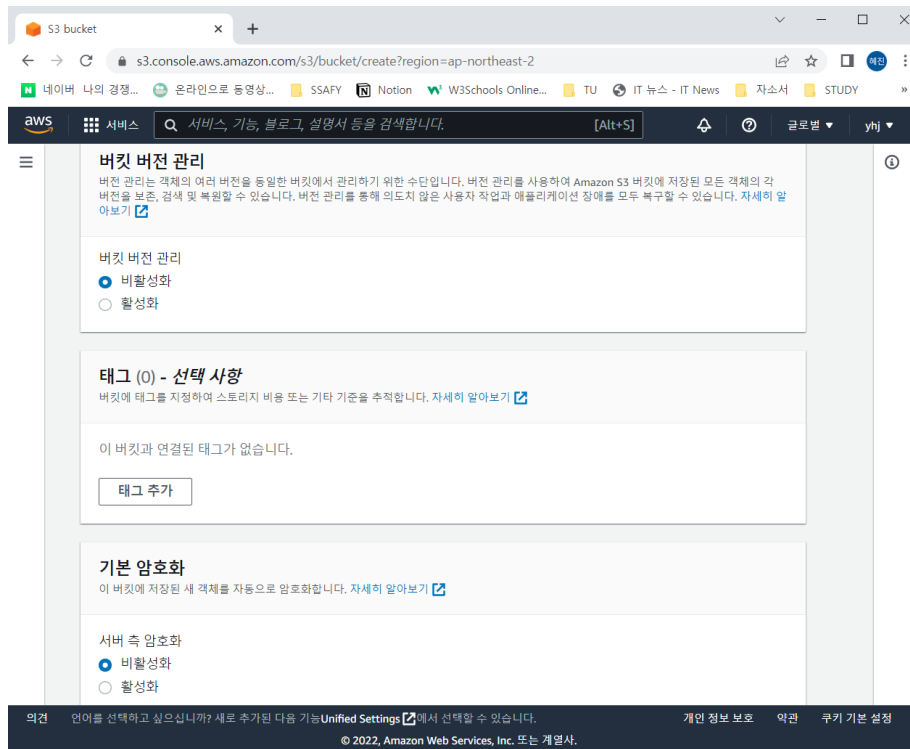
1) 프로젝트 이름으로 버킷 생성 및 지역 설정



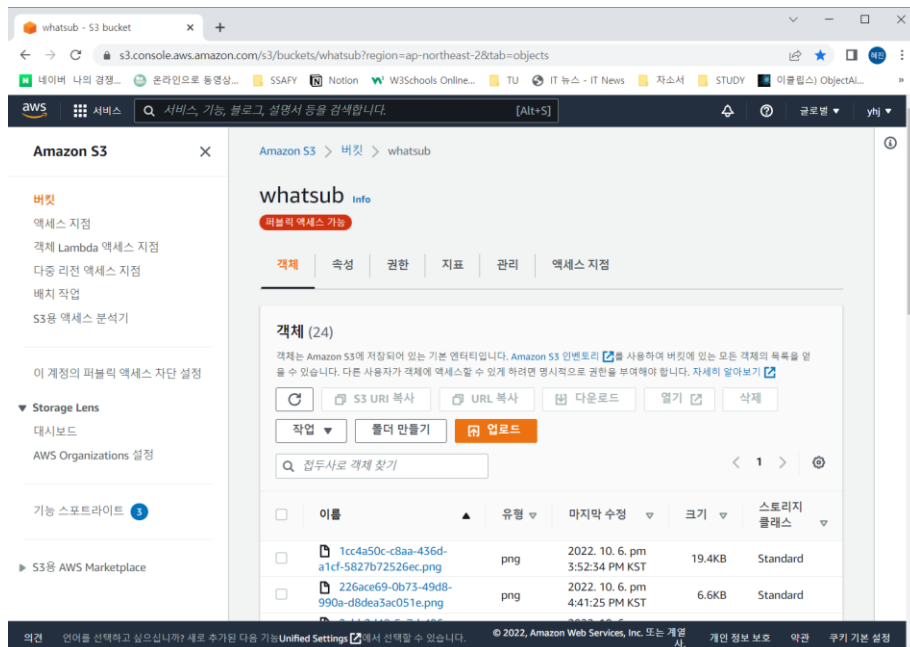
2) 퍼블릭 액세스 차단 해제 (이미지 서버 접근 가능 설정)



3) 나머지 부분은 default로 설정



4) 생성 완료 후 해당 버킷 클릭



5) 버킷 권한 정책 설정

가) 권한 -> 버킷 정책 -> 편집 -> 정책 생성기

나) Actions에서 GetObject, DeleteObject, PutObject 선택 (나머지는 사진과 같이 설정)

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy: S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect: ☒ Allow ☐ Deny

Principal:

Use a comma to separate multiple values.

AWS Service: Amazon S3 ☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions: 3 Action(s) Selected ☐ All Actions ("*")

Amazon Resource Name (ARN): arn:aws:s3:::whatsub

ARN should follow the following format: arn:aws:s3:::s(BucketName)/s(KeyName). Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

6) 생성된 JSON 복사

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will **not** be reflected in the policy generator tool.

```
{
  "Id": "Policy1665066820907",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1665066820907",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::whatsub",
      "Principal": "*"
    }
  ]
}
```

Close

7) 버킷 정책에 붙여넣고 저장

Amazon S3

버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체에는 적용되지 않습니다. 자세히 알아보기

정책 예제 정책 생성기

버킷 ARN: arn:aws:s3:::whatsub

정책

```
1 {
2   "Version": "2012-10-17",
3   "Id": "Policy1663919492907",
4   "Statement": [
5     {
6       "Sid": "Stmt1663919481564",
7       "Effect": "Allow",
8       "Principal": "*",
9       "Action": [
10        "s3:DeleteObject",
11        "s3:GetObject",
12        "s3:PutObject"
13      ],
14      "Resource": "arn:aws:s3:::whatsub"
15    }
16  ]
17 }
```

문 편집

문 선택

정책에서 기존 문을 선택하거나 새 문을 추가합니다.

+ 새 문 추가

© 2022, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

나) 카카오 로그인과 지도 사용을 위한 카카오 API 설정

1) 카카오 애플리케이션 생성

2) 활성화 설정 ON

내 애플리케이션 > 제품 설정 > 카카오 로그인

블릿

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

연결 끊기

사용자 프로퍼티

보안

고급

메시지

APP

whatsub

ID 802785 OWNER Web

카카오 로그인 ON

동의 화면 미리보기

활성화 설정

상태 ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다. 상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다. 상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

3) Redirect URL 추가 (카카오 로그인)

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

연결 끊기

사용자 프로퍼티

Redirect URI

삭제 수정

Redirect URI	http://localhost:8081/api/v1/auth/login http://localhost:8080/signup https://j7a105.p.ssafy.io/signup
--------------	---

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

4) 동의 항목 선택

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

연결 끊기

사용자 프로퍼티

보안

고급

메시지

카카오톡 채널

개인정보

항목 이름	ID	상태
닉네임	profile_nickname	필수 동의 <div>설정</div>
프로필 사진	profile_image	필수 동의 <div>설정</div>
카카오계정(이메일)	account_email	선택 동의 <div>설정</div>
성별	gender	선택 동의 <div>설정</div>
연령대	age_range	사용 안함 <div>설정</div>
생일	birthday	사용 안함 <div>설정</div>
출생 연도	birthyear	권한 없음