



# 포팅메뉴얼

```
# [B102] FoodReco

FoodReco 프로젝트는 Docker, Docker Compose, 그리고 Jenkins Pipeline을 활용한
CI/CD 자동화 환경을 구성하고 있습니다.

Gitlab의 Webhook 설정으로 인해 develop 브랜치에 Merge 이벤트 발생 시,
Jenkins Pipeline을 통해 자동 빌드와 배포가 이루어집니다.

프론트엔드 부분은 npm 환경에서 빌드하고 배포합니다.
백엔드는 Gradle을 사용하여 빌드하며, Docker Compose를 통해 컨테이너를 관리하고 배포합니다.

## Version

### Frontend

| Type          | Version |
| ----- | ----- |
| Node.js      | 16.18.50 |
| React.js     | 18.2.21 |
| Axios        | 1.5.0    |

### Backend

| Type          | Version |
| ----- | ----- |
| Java          | openjdk : 11.0.18 |
| Spring Boot   | 2.7.15  |
| Gradle        | 8.2.1   |
| Spring Security | 5.7.10  |
| JPA           | -       |
| Hibernate     | 7.0.5   |
| IntelliJ Ultimate | 2023.1.3 |
| JWT           | 4.2.1   |
| smtp          | 2.7.0   |
| S3            | 2.2.1   |
| lombok        | 1.18.14 |

### Database

| Type          | Version |
| ----- | ----- |
| MySQL         | 8.0.33  |
| MySQL Workbench | 8.0 CE  |
| Redis         | 7.2.1   |

## ufw allow status

| Port | Content |
| ---- | ---- |
| 22    | SSH     |
| 80    | HTTP    |
| 443   | HTTPS   |
| 3306  | MySQL   |
```

## 1. EC2 Setting

### 1-1. Docker

- ec2에 docker 설치

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

echo \
```

```
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keysrings/docker.gpg] https://download.docker.com/linux/ubuntu \
"${. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- docker engine과 그에 따른 plugin설치

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo apt install docker-compose

#정상 설치 되었는지 확인
sudo docker -v
sudo docker compose version
```

## 1-2. nginx

- 서버의 패키지 목록 업데이트
  1. sudo apt update
  2. sudo apt upgrade
  3. sudo apt autoremove
- Nginx 설치
  1. sudo apt install nginx
- 실행
  1. sudo service start nginx
  2. sudo service status nginx
- 환경 설정
  1. sudo vi /etc/nginx/sites-available/nginx.conf
  2. 아래 내용을 붙여 넣음

```
server {
    listen 80; # 80포트로 받을 때

    server_name j9b102.p.ssafy.io; # 도메인주소

    return 308 https://$host$request_uri;
}

server {
    listen 443 ssl default_server;
    server_name j9b102.p.ssafy.io;

    # ssl 인증서 적용하기
    ssl_certificate /etc/letsencrypt/live/j9b102.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j9b102.p.ssafy.io/privkey.pem;

    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    client_max_body_size 75M;

    location /api/crew/sse {

        proxy_pass http://localhost:8080;

        charset utf-8;
        proxy_set_header Host $http_host;
```

```

    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Connection '';
    proxy_http_version 1.1;
    proxy_buffering off;
    proxy_set_header X-Accel-Buffering no;

    add_header 'Content-Type' 'text/event-stream'; # SSE 스트림에만 적용
    add_header 'Cache-Control' 'no-cache';
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
    add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';
    add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
    proxy_read_timeout 36000s;

}

location / {
    # /로 들어오는 경우
    proxy_pass http://localhost:3000;
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Expose-Headers' 'Authorization, Authorizationrefresh';

    proxy_buffer_size 128k;
    proxy_buffers 4 256k;
    proxy_busy_buffers_size 256k;
}

location /api {
    # 일반 API 호출의 경우
    proxy_pass http://localhost:8080;
    #add_header 'Cross-Origin-Embedder-Policy' 'require-corp';
    #add_header 'Cross-Origin-Opener-Policy' 'same-origin';
    charset utf-8;

    proxy_http_version 1.1;
    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_set_header X-NginX-Proxy true;
    proxy_buffer_size 128k;
    proxy_buffers 4 256k;
    proxy_busy_buffers_size 256k;
}

location /fastapi {
    proxy_pass http://localhost:7777;

    charset utf-8;
    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;
}
}

```

- 실제 사용되는 site-enabled에 nginx.conf 연결

1. `sudo ln -s /etc/nginx/sites-available/nginx.conf /etc/nginx/sites-enabled`

## 1-3. SSL 인증서

### 1. let's Encrypt 설치

```
$ sudo apt-get install letsencrypt
```

### 2. Cerbot 설치

```
$ sudo apt-get install certbot python3-certbot-nginx
```

### 3. certbot 동작

```
$ sudo certbot --nginx
```

- 이메일 입력, 약관동의, 이메일발송동의, 도메인 입력
- 여기서 3 이상 실패 시 인증서 발급이 일주일동안 막힘, 꼭꼭 주의해서 입력해서 한번에 성공해야 함!!

## 1-4. Jenkins 설치

- docker-compose를 사용하여 jenkins container를 실행하기

### 1. jenkins container를 실행시킬 docker-compose 파일을 만듦

```
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:
      - "9090:8080"
    user: root
  mysql:
    image: mysql:8.0.33
    container_name: mysql
    restart: always
    ports:
      - "3306:3306"
    volumes:
      - ./volume:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=dlrnehdtjd
      - MYSQL_DATABASE=specialdb
      - MYSQL_USER=cho
      - MYSQL_PASSWORD=dlrnehdtjd
    networks:
      - network
  redis:
    image: redis:latest
    command: redis-server --requirepass wjdtjdcn --port 6379
    container_name: redis
    labels:
      - "name=redis"
      - "mode=standalone"
    ports:
      - 6379:6379
networks:
  network:
```

### 2. 정상적으로 container가 실행되고 있는지 확인

```
sudo docker ps -a
```

```
bc3a359676a4 jenkins/jenkins:lts "/usr/bin/tini -- /u..." 3 weeks ago Up 6 days 50000/t
cp, 0.0.0.0:9090->8080/tcp, :::9090->8080/tcp jenkins
```

## 2. Jenkins 세팅

- Jenkins 접속

```
http://9b102.p.ssafy.io:9090/
```

## 2-1. 초기 계정 설정

### 1. 초기 접속화면 Unlock Jenkins

- `cat /var/lib/jenkins/secrets/initialAdminPassword`
- 여기서 비밀번호를 확인하고 입력

### 2. install suggested plugins 선택

### 3. Create First Admin User

- a. 계정명 : 2H4D
- b. 암호 : wjdtjdcn
- c. 이름 : 2H4D

### 4. Jenkins Url

- a. default사용해도 됨,

## 2-2. plugin 설치

Jenkins 관리 → Plugin in

### 추가로 설치 해야 할 plugin 목록

- GitLab
- Generic Webhook Trigger
- GitLab API
- GitLab Authentication
- Loading plugin extention
- NodeJs (자동 배포로 프론트엔트 빌드 시 필요)

## 2-3. Credential 설정

Jenkins 관리 → Security → Credentials → Stores scoped to Jenkins → (global)

→ + Add Credential 선택

Scope ?

Global (Jenkins, nodes, items, all child items, etc) ▼

API token

🔒 Concealed Change Password

ID ?

f563bb03-9b5a-4caa-afc8-3808231ba01e

Description ?

Save

- gitlab에서 먼저 access Token을 발급 받고 토큰을 넣어준다.
  - 토큰 발급 받기
- gitlab 로그인 → 사람 누름 → Edit profile → access token
- 이름, 만료날짜, 권한 범위 등을 설정

s09P22B102

Project information

Repository

Issues 0

Jira

Merge requests 0

CI/CD

Security and Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

**Settings**

General

Integrations

Webhooks

**Access Tokens**

Repository

Merge requests

s09-bigdata-recom-sub2 > S09P22B102 > Access Tokens

Q Search page

**Project Access Tokens**

Generate project access tokens scoped to this project for your applications that need access to the GitLab API. You can also use project access tokens with Git to authenticate over HTTP(S). [Learn more.](#)

**Add a project access token**

Enter the name of your application, and we'll return a unique project access token.

**Token name**

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

**Expiration date**

2023-11-04 🗓

**Select a role**

Guest ▼

**Select scopes**

Scopes set the permission levels granted to the token. [Learn more.](#)

☐ api  
Grants complete read and write access to the scoped project API, including the Package Registry.

☐ read\_api  
Grants read access to the scoped project API, including the Package Registry.

☐ read\_repository  
Grants read access (pull) to the repository.

☐ write\_repository  
Grants read and write access (pull and push) to the repository.

Create project access token

## 2-4. jenkins GitLab Connection 등록

Jenkins관리 → System → GitLab 이동

- 원하는 connection 이름 설정
- Gitlab 주소 입력
- 앞서 만든 Credential 연결

## GitLab

☒ Enable authentication for '/project' end-point ?

GitLab connections

Connection name ?

A name for the connection

Backend

GitLab host URL ?

The complete URL to the GitLab server (e.g. http://gitlab.mydomain.com)

https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B102/backend

Credentials ?

API Token for accessing GitLab

- none -

Add

API Token for GitLab access required

고급

고급

API-Level ?

API Level for accessing GitLab

autodetect

☐ Ignore SSL Certificate Errors ?

Connection timeout (in seconds) ?

The time to wait for establishing the connection

10

Read timeout (in seconds) ?

The time to wait while receiving the response

10

Test Connection

Connection name ?

저장

Apply

## 2-5. Jenkins pipeline 생성

- + 새로운 Item → 이름 입력, Pipeline 선택 → OK

- 구성 → build Trigger 이동
- build를 유발할 Tirgger 옵션을 선택하여 적용
- 고급을 눌러 webhook 설정을 위한 Secret Token을 발급

포팅메뉴얼



고급 ^

☒ Enable [ci-skip] ?

☒ Ignore WIP Merge Requests ?

Labels that launch a build if they are added (comma-separated) ?

☒ Set build description to build cause (eg. Merge request or Git Push) ?

☐ Build on successful pipeline events

Pending build name for pipeline ?

☐ Cancel pending merge request builds on update ?

Allowed branches

☒ Allow all branches to trigger this job ?

☐ Filter branches by name ?

☐ Filter branches by regex ?

Secret token ?

54719285f76f693a8db57c5c46ebdf1c

Generate

→ generate를 눌러 Secret token 생성

## 2-6. Gitlab webhook 설정

- jenkins 작업물의 변화를 감지하여 build, run 하기 위해서는 webHook 필수!
- gitlab project → settings → webhooks
- url : **http://j9b102.p.ssafy.io:9090/project/backend-pipeline**
- secret token: jenkins System에서 받아온 token 입력
- merge를 할때마다 요청

- SSL verification

- Test : 200이여야 함!

### 3. Build & Distribute

### 3-1. Backend - Spring Boot

```
FROM openjdk:11-jdk
ARG JAR_FILE=build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "-Duser.timezone=Asia/Seoul", "/app.jar"]
```

## 포팅메뉴얼

```

pipeline {
    agent any

    stages {
        stage('Springboot build') {
            steps {
                dir('backend'){
                    sh '''
                        echo 'springboot build'
                        chmod +x gradlew
                        ./gradlew clean build
                    '''
                }
            }
        }
        stage('Dockerimage build') {
            steps {
                dir('backend'){
                    sh '''
                        echo 'Dockerimage build'
                        docker build -t docker-springboot:0.0.1 .
                    '''
                }
            }
        }
        stage('Deploy') {
            steps {
                dir('backend'){

                    sh '''
                        echo 'Deploy'
                        docker stop springboot
                        docker rm springboot
                        docker run -d -p 127.0.0.1:8080:8080 --name springboot docker-springboot:0.0.1
                        docker images -f "dangling=true" -q | xargs -r docker rmi
                    '''
                }
            }
        }
    }
}

```

## Jenkins 설정

## Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B102.git

Credentials ?

jjhjjh1159@gmail.com/\*\*\*\*\*

Add

고급

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/develop

Add Branch

Repository browser ?

(자동)

Additional Behaviours

**Polling ignores commits in certain paths** ?

Included Regions ?

Excluded Regions ?

```
frontend/**
fastapi/**
```

Add ▾

Script Path ?

```
backend/Jenkinsfile
```

☒ Lightweight checkout ?

[Pipeline Syntax](#)

저장

Apply

- jenkinsfile과 dockerfile이 있는 경로 지정
- 지금 빌드 실행하여 확인

## Stage View

			Declarative: Checkout SCM	Springboot build	Dockerimage build	Deploy
Average stage times: (Average <u>full</u> run time: ~34s)			456ms	19s	5s	3s
#318	10월 05 일 15:53	No Changes	533ms	16s	5s	4s
#317	10월 05 일 15:32	1 commit	403ms	17s	5s	3s

## 3-2. Frontend - React

### Dockerfile

```
FROM node:18.17.1

RUN npm install -g serve

RUN mkdir ./build
ADD ./build ./build

ENTRYPOINT ["serve", "-s", "build"]
```

## Jenkinsfile

```
pipeline {
    agent any

    tools {
        nodejs "NodeJS" // 여기서 "Node18"은 위에서 설정한 Node.js의 이름입니다.
    }

    stages {

        stage('React build') {
            steps {
                dir('frontend') {
                    echo 'React build'
                    sh 'npm install' // --save 옵션은 더 이상 필요하지 않습니다.
                    sh 'CI=false npm run build'
                }
            }
        }

        stage('Dockerimage build') {
            steps {
                dir('frontend') {
                    sh '''
                        echo 'Dockerimage build for React'
                        docker build -t docker-react:0.0.1 .
                    '''
                }
            }
        }

        stage('Deploy') {
            steps {
                sh '''
                    echo 'Deploy React'
                    docker stop react
                    docker rm react
                    docker run -d -p 127.0.0.1:3000:3000 --name react docker-react:0.0.1
                    docker images -f "dangling=true" -q | xargs -r docker rmi
                '''
            }
        }
    }
}
```

## Jenkins 설정

## Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B102.git

Credentials ?

jjhjjh1159@gmail.com/\*\*\*\*\*

Add

고급

Excluded Regions ?

backend/\*\*  
fastapi/\*\*

Add

Script Path ?

frontend/Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

저장

Apply

- jenkins 관리 → Tools → NodeJs 설정
- node 버전과 이름을 입력

## NodeJS installations

NodeJS installations ^ Edited

Add NodeJS

NodeJS  
Name

NodeJS

☒ Install automatically ?

Install from nodejs.org

Version

NodeJS 18.17.1

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☒ Force 32bit architecture

Global npm packages to install  
Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

Global npm packages refresh hours  
Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

Add Installer

Add NodeJS

Save Apply

- 지금 빌드 실행하여 확인

## Stage View

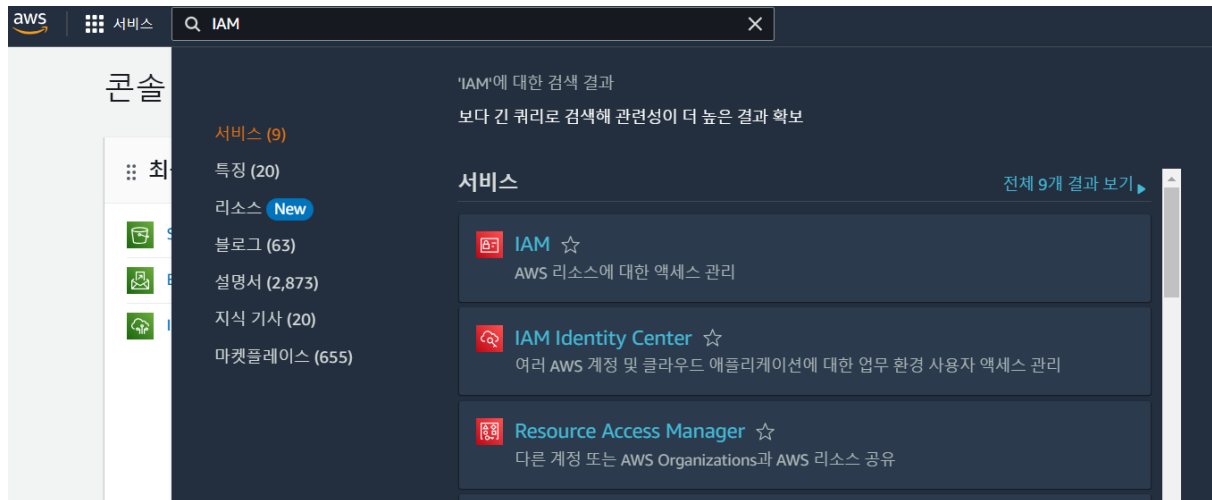
		Declarative: Checkout SCM	Declarative: Tool Install	React build	Dockerimage build	Deploy
Average stage times: (Average <u>full</u> run time: ~1min 4s)		505ms	90ms	56s	2s	912ms
#274	10월 05 일 16:01 1 commit	416ms	96ms	55s	2s	913ms
#273	10월 05 일 15:32 1 commit	591ms	119ms	53s	2s	919ms

## 4. 외부 서비스 이용



## 4-1. S3

- IAM



- S3에 접근하는 IAM 사용자 생성



- S3 접근 권한 부여 - AmazonS3FullAccess 추가

## 권한 설정

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 직무별로 사용자의 권한을 관리하려면 그룹을 사용하는 것이 좋습니다. [자세히 알아보기](#)

### 권한 옵션

☐ 그룹에 사용자 추가

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 그룹을 사용하여 직무별로 사용자 권한을 관리하는 것이 좋습니다.

☐ 권한 복사

기존 사용자의 모든 그룹 멤버십, 연결된 관리형 정책 및 인라인 정책을 복사합니다.

☒ 직접 정책 연결

관리형 정책을 사용자에게 직접 연결합니다. 사용자에게 연결하는 대신, 정책을 그룹에 연결한 후 사용자를 적절한 그룹에 추가하는 것이 좋습니다.

### 권한 정책 (1130)

새 사용자에게 연결할 정책을 하나 이상 선택합니다.



정책 생성

필터링 기준 유형

Q S3

모든 유형

12 개 일치

< 1 > ⚙

<input type="checkbox"/>	정책 이름	유형	연결된 엔터티
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	AWS 관리형	0
<input type="checkbox"/>	AmazonS3FullAccess	AWS 관리형	1

- 생성한 사용자 설정
- 액세스 키 만들기

## Food\_Reco 정보

삭제

### 요약

ARN arn:aws:iam::140654384673:user/Food_Reco	콘솔 액세스 MFA 없이 활성화됨	액세스 키 1 AKIASBP5HSYQUZHWPNC - Active 🟢 사용됨 19시간 전, 14일 기준.
생성됨 September 21, 2023, 12:46 (UTC+09:00)	마지막 콘솔 로그인 🟢 6일 전	액세스 키 2 액세스 키 만들기

- 'AWS 외부에서 실행되는 애플리케이션' 선택
- 액세스 키, 비밀 액세스 키 생성 완료 (csv 파일 다운로드)

🔑 액세스 키 생성됨  
지금 이 아니면 비밀 액세스 키를 보거나 다운로드할 수 없습니다. 나중에 복구할 수 없습니다. 하지만 언제든지 새 액세스 키를 생성할 수 있습니다.

IAM > 사용자 > Food\_Reco > 액세스 키 만들기

1단계  
액세스 키 모범 사례 및 대안

2단계 - 선택 사항  
설명 태그 설정

3단계  
액세스 키 검색

### 액세스 키 검색 정보

액세스 키

AKIASBP5HSYQQCEASRW6

비밀 액세스 키

\*\*\*\*\* 표시

### 액세스 키 모범 사례

- 액세스 키를 일반 텍스트, 코드 리포지토리 또는 코드로 저장해서는 안 됩니다.
- 더 이상 필요 없는 경우 액세스 키를 비활성화하거나 삭제합니다.
- 최소 권한을 활성화합니다.
- 액세스 키를 정기적으로 교체합니다.

액세스 키 관리에 대한 자세한 내용은 [AWS 액세스 키 관리 모범 사례](#)를 참조하세요.

[.csv 파일 다운로드](#) [완료](#)

## • S3 서비스 접속

aws 서비스 🔍 S3

Identity and Access Management (IAM)

🔍 IAM 검색

대시보드

▼ 액세스 관리

사용자 그룹

**사용자**

역할

정책

자격 증명 공급자

서비스 (9)

특징 (26)

리소스 **New**

블로그 (148)

설명서 (2,246)

지식 기사 (20)

튜토리얼 (7)

마켓플레이스 (1,369)

'S3'에 대한 검색 결과  
보다 긴 쿼리로 검색해 관련성이 더 높은 결과 확보

### 서비스

전체 9개 결과 보기 ▶

S3 ☆

클라우드의 확장 가능한 스토리지

S3 Glacier ☆

클라우드의 아카이브 스토리지

AWS Snow Family ☆

대규모 데이터 전송

## • 버킷 생성 및 설정

**버킷 (1) 정보**

버킷은 S3에 저장되는 데이터의 컨테이너입니다. [자세히 알아보기](#)

🔄 ARN 복사 비어 있음 삭제 **버킷 만들기**

🔍 이름으로 버킷 찾기

< 1 > ⚙️

	이름 ▲	AWS 리전 ▼	액세스 ▼	생성 날짜 ▼
○	food-reco	아시아 태평양(서울) ap-northeast-2	퍼블릭	2023. 9. 21. pm 2:18:41 PM KST

## 일반 구성

버킷 이름

asdasdasd

버킷 이름은 글로벌 네임스페이스 내에서 고유해야 하며 버킷 이름 지정 규칙을 따라야 합니다. [버킷 이름 지정 규칙 보기](#)

AWS 리전

아시아 태평양(서울) ap-northeast-2

기존 버킷에서 설정 복사 - 선택 사항

다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

## 객체 소유권 정보

다른 AWS 계정에서 이 버킷에 작성한 객체의 소유권 및 액세스 제어 목록(ACL)의 사용을 제어합니다. 객체 소유권은 객체에 대한 액세스를 지정할 수 있는 사용자를 결정합니다.

### ☐ ACL 비활성화됨(권장)

이 버킷의 모든 객체는 이 계정이 소유합니다. 이 버킷과 그 객체에 대한 액세스는 정책을 통해서만 지정됩니다.

### ☒ ACL 활성화됨

이 버킷의 객체는 다른 AWS 계정에서 소유할 수 있습니다. 이 버킷 및 객체에 대한 액세스는 ACL을 사용하여 지정할 수 있습니다.

## 이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. [자세히 알아보기](#)

### ☐ 모든 퍼블릭 액세스 차단

이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

#### ☐ 새 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 새로 추가된 버킷 또는 객체에 적용되는 퍼블릭 액세스 권한을 차단하며, 기존 버킷 및 객체에 대한 새 퍼블릭 액세스 ACL 생성을 금지합니다. 이 설정은 ACL을 사용하여 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 권한을 변경하지 않습니다.

#### ☐ 임의의 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 모든 ACL을 무시합니다.

#### ☐ 새 퍼블릭 버킷 또는 액세스 지정 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지정 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.

#### ☐ 임의의 퍼블릭 버킷 또는 액세스 지정 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 정책을 사용하는 버킷 또는 액세스 지점에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

## 버킷 버전 관리

버전 관리는 객체의 여러 버전을 동일한 버킷에서 관리하기 위한 수단입니다. 버전 관리를 사용하여 Amazon S3 버킷에 저장된 모든 객체의 각 버전을 보존, 검색 및 복원할 수 있습니다. 버전 관리를 통해 의도치 않은 사용자 작업과 애플리케이션 장애를 모두 복구할 수 있습니다. [자세히 알아보기](#)

### 버킷 버전 관리

- ☒ 비활성화
- ☐ 활성화

## 기본 암호화 정보

서버 측 암호화는 이 버킷에 저장된 새 객체에 자동으로 적용됩니다.

### 암호화 유형 정보

- ☒ Amazon S3 관리형 키(SSE-S3)를 사용한 서버 측 암호화
- ☐ AWS Key Management Service 키를 사용한 서버 측 암호화(SSE-KMS)
- ☐ AWS Key Management Service 키를 사용한 이중 계층 서버 측 암호화(DSSE-KMS)  
두 개의 개별 암호화 계층으로 객체를 보호합니다. 요금에 대한 자세한 내용은 [Amazon S3 요금 페이지](#)의 스토리지 탭에서 DSSE-KMS 요금을 참조하세요.

### 버킷 키

SSE-KMS용 S3 버킷 키를 사용하면 AWS KMS에 대한 호출을 줄여 암호화 비용이 절감됩니다. DSSE-KMS에는 S3 버킷 키가 지원되지 않습니다. [자세히 알아보기](#)

- ☐ 비활성화
- ☒ 활성화

## ▼ 고급 설정

### 객체 잠금

WORM(Write-Once-Read-Many) 모델을 사용하여 객체를 저장하면 고정된 시간 동안 또는 무기한으로 객체가 삭제되거나 덮어쓰이지 않도록 할 수 있습니다. [자세히 알아보기](#)

- ☒ 비활성화
- ☐ 활성화  
이 버킷의 객체를 영구적으로 잠글 수 있습니다. 버킷 생성 후 이 버킷의 객체가 삭제되거나 덮어쓰기 되지 않도록 버킷 세부 정보에 추가 객체 잠금 구성이 필요합니다.

## 생성된 버킷 설정

- 정책 설정

## 버킷 정책

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


```
{
  "Version": "2012-10-17",
  "Id": "Policy1695273592316",
  "Statement": [
    {
      "Sid": "Stmnt1695273591323",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::food-reco/*"
    }
  ]
}
```


### • ACL 설정

#### ACL(액세스 제어 목록)






다른 AWS 계정에 기본 읽기/쓰기 권한을 부여합니다. [자세히 알아보기](#)

편집

 중복 피부여자에 대한 통합 액세스 권한 부여가 콘솔에 표시됩니다.  
전체 ACL 목록을 보려면 Amazon S3 REST API, AWS CLI 또는 AWS SDK를 사용하세요.

 AWS는 모든 사용자 피부여자에게 액세스 권한을 부여하지 않도록 권장합니다.  
누구나 이 버킷의 객체에 액세스할 수 있습니다.

[자세히 알아보기](#)

피부여자	객체	버킷 ACL
버킷 소유자(AWS 계정) 정식 ID:  f6fa0ec7f41b46d65ff438f0ecbd584b52c11d7956953bcb81b7f6fb8c954584	나열, 쓰기	읽기, 쓰기
모든 사람(퍼블릭 액세스) 그룹:  http://acs.amazonaws.com/groups/global/AllUsers	 나열	 읽기
인증된 사용자 그룹(AWS 계정이 있는 모든 사용자) 그룹:  http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-

이제 발급받은 access-key, secret-key, bucket정보를 Spring에서 이용하여 S3에 접근할 수 있다.

## 2. 공공데이터

### 날씨 오픈 API

XML
JSON
**기상청\_단기예보 ((구)\_동네예보) 조회서비스**
활용신청

초단기실황, 초단기예보, 단기((구)동네)예보, 예보버전 정보를 조회하는 서비스입니다. 초단기실황정보는 예보 구역에 대한 대표 AWS 관측값을, 초단기예보는 예보시점부터 6시간까지의 예보를, 단기예보는 예보기간을 글짜까지 확장 및 예보단위를 상세화(3시간→1시간)하여 시공간적으로 세분화한 예보를 제공합니다.

133
17
관심

## 활용 신청 및 서비스키 발급

요청주소 [http://apis.data.go.kr/1360000/VilageFcstInfoService\\_2.0/getUltraSrtNcst](http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst)

### 요청변수(Request Parameter)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	ServiceKey	4	필수	-	공공데이터포털에서 받은 인증키
페이지 번호	pageNo	4	필수	1	페이지번호
한 페이지 결과 수	numOfRows	4	필수	1000	한 페이지 결과 수
응답자료형식	dataType	4	옵션	XML	요청자료형식(XML/JSON) Default: XML
발표일자	base_date	8	필수	20210628	'21년 6월 28일 발표
발표시각	base_time	4	필수	0600	06시 발표(정시단위)
예보지점 X 좌표	nx	2	필수	55	예보지점의 X 좌표값
예보지점 Y 좌표	ny	2	필수	127	예보지점의 Y 좌표값

### 출력결과(Response Element)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	2	필수	00	결과코드
결과메시지	resultMsg	50	필수	OK	결과메시지
한 페이지 결과 수	numOfRows	4	필수	10	한 페이지 결과 수
페이지 번호	pageNo	4	필수	1	페이지번호
전체 결과 수	totalCount	4	필수	3	전체 결과 수
데이터 타입	dataType	4	필수	XML	응답자료형식 (XML/JSON)
발표일자	baseDate	8	필수	20210628	'21년 6월 28일 발표
발표시각	baseTime	6	필수	0600	06시 발표(매 정시)
예보지점 X 좌표	nx	2	필수	55	입력한 예보지점 X 좌표
예보지점 Y 좌표	ny	2	필수	127	입력한 예보지점 Y 좌표
자료구분코드	category	3	필수	RN1	자료구분코드
실황 값	obsrValue	2	필수	0	RN1, T1H, UUU, VVV, WSD 실수로 제공

## Response 예시

```

<?xml version="1.0" encoding="UTF-8"?>
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_SERVICE</resultMsg>
  </header>
  <body>
    <dataType>XML</dataType>
    <items>
      <item>
        <baseDate>20231005</baseDate>
        <baseTime>0630</baseTime>
        <category>LGT</category>
        <fcstDate>20231005</fcstDate>
        <fcstTime>0700</fcstTime>
        <fcstValue>0</fcstValue>
        <nx>55</nx>
        <ny>127</ny>
      </item>
      <item>
        <baseDate>20231005</baseDate>
        <baseTime>0630</baseTime>
        <category>LGT</category>
        <fcstDate>20231005</fcstDate>
        <fcstTime>0800</fcstTime>
        <fcstValue>0</fcstValue>
        <nx>55</nx>
        <ny>127</ny>
      </item>
      <item>
        <baseDate>20231005</baseDate>
        <baseTime>0630</baseTime>
        <category>LGT</category>
        <fcstDate>20231005</fcstDate>
        <fcstTime>0900</fcstTime>
        <fcstValue>0</fcstValue>
        <nx>55</nx>
        <ny>127</ny>
      </item>
      <item>
        <baseDate>20231005</baseDate>
        <baseTime>0630</baseTime>
        <category>LGT</category>
        <fcstDate>20231005</fcstDate>
        <fcstTime>1000</fcstTime>
        <fcstValue>0</fcstValue>
      </item>
    </items>
  </body>
</response>

```

## Google Fitness

구글 클라우드 콘솔에서 프로젝트 생성 및 구글 클라우드 플랫폼에서 OAuth 동의 화면 구성

Google Cloud

My Project 63628

리소스, 문서, 제품 등 검색(/)

검색

API 및 서비스

OAuth 동의 화면

알아보기

사용 설정된 API 및 서비스

라이브러리

사용자 인증 정보

OAuth 동의 화면

페이지 사용 동의

FoodReco

확인 상태

인증 필요

하나 이상의 민감한 범위를 사용하고 있기 때문에 앱 등록 시 Google의 확인이 필요합니다. 앱을 준비하여 확인을 위해 제출하세요. 자세히 알아보기

확인 준비

게시 상태

프로덕션 단계

테스트로 돌아가기

사용자 유형

외부

내부로 설정

프로덕션 준비

확인

이 필요한 방식으로 앱을 구성했습니다. 확인을 완료하려면 다음을 제공해야 합니다.

- 앱 개인정보처리방침의 공식 링크
- 범위 내에서 가져오는 Google 사용자 데이터를 어떻게 사용할 계획인지 보여주는 YouTube 동영상
- 민감하거나 제한된 사용자 데이터에 액세스해야 하는 이유를 Google에 알리는 서면 설명서
- Google Search Console에서 확인된 모든 도메인

Google OAuth 동의 화면

OAuth 동의 화면이란 무엇인가요?

OAuth 동의 범위란 무엇인가요?

민감한 API 범위란 무엇인가요?

제한된 API 범위란 무엇인가요?

애드 로 프로그래서

사용자 인증정보 생성

리디렉션 URI를 잘 설정해야 합니다.



Google Cloud | My Project 63628 | 리소스, 문서, 제품 등 검색(/) | 검색

**API 및 서비스**

- 사용 설정된 API 및 서비스
- 라이브러리
- 사용자 인증 정보**
- OAuth 동의 화면
- 페이지 사용 동의

**사용자 인증 정보** + 사용자 인증 정보 만들기 삭제 삭제된 사용자 인증 정보 복원

사용 설정한 API에 액세스하려면 사용자 인증 정보를 만드세요. [자세히 알아보기](#)

▲ 나와 내 사용자가 보호받으려면 동의 화면과 애플리케이션이 Google의 확인을 거쳐야 합니다. [자세히 알아보기](#) 동의 화면 구성

**API 키**

이름	생성일 ↓	제한사항	작업
표시할 API 키가 없습니다.			

**OAuth 2.0 클라이언트 ID**

이름	생성일 ↓	유형	클라이언트 ID	작업
▲ <a href="#">클라이언트 1</a>	2023. 9. 19.	웹 애플리케이션	195561660115-6gse...	<a href="#">편집</a> <a href="#">삭제</a> <a href="#">다운로드</a>

**서비스 계정** [서비스 계정 관리](#)

이메일	이름 ↑	작업
표시할 서비스 계정이 없습니다.		

구글 플레이그라운드에서 테스트 해볼 수 있습니다.

응답 예시

```
{
  "bucket": [
    {
      "startTimeMillis": "1696416286449",
      "endTimeMillis": "1696502686449",
      "dataset": [
        {
          "dataSourceId": "derived:com.google.step_count.delta:com.google.android.gms:aggregated",
          "point": [
            {
              "startTimeNanos": "1696418489163491328",
              "originDataSourceId": "derived:com.google.step_count.delta:com.google.ios.fit:appleinc.:watch:3084e4fa:top_level",
              "endTimeNanos": "1696430572520426752",
              "value": [
                {
                  "mapVal": [],
                  "intVal": 909
                }
              ]
            }
          ],
          "dataTypeName": "com.google.step_count.delta"
        }
      ]
    },
    {
      "startTimeMillis": "1696502686449",
      "endTimeMillis": "1696502686450",
      "dataset": [
        {
          "dataSourceId": "derived:com.google.step_count.delta:com.google.android.gms:aggregated",
          "point": [
            {
              "startTimeNanos": "1696502686450",
              "originDataSourceId": "derived:com.google.step_count.delta:com.google.ios.fit:appleinc.:watch:3084e4fa:top_level",
              "endTimeNanos": "1696502686450",
              "value": [
                {
                  "mapVal": [],
                  "intVal": 909
                }
              ]
            }
          ],
          "dataTypeName": "com.google.step_count.delta"
        }
      ]
    }
  ]
}
```