

WATERBELL

포팅 매뉴얼 & 외부 서비스 정보

삼성청년SW아카데미 대전캠퍼스 9기

공통 프로젝트 (6주, 2023.07.10 ~ 2023.08.18)

김동현, 김예진, 이효경, 조준희, 황윤영, 황종인

목차

목차

1. 프로젝트 기술 스택

[Frontend](#)

[Backend](#)

[IOT](#)

[기타](#)

2. 개발 환경 Setting

[2.1 Frontend](#)

[2.2 IoT](#)

3. DB Connection

[3.1 DBeaver에 추가하기](#)

4. EC2 세팅

[4.1 EC2 접속하기](#)

[4.2 방화벽 설정하기](#)

`sudo ufw allow {포트 번호}`

[4.3 DB 세팅](#)

[4.4 Redis 세팅](#)

[4.5 Docker 설치](#)

[4.6 Jenkins 설치하기](#)

5. Jenkins 세팅

[5.1 초기 계정 설정](#)

[5.2 plugin 설치](#)

추가로 설치 해야 할 plugin 목록

[5.3 Credential 설정](#)

[5.4 jenkins GitLab Connection 등록](#)

[5.5 Jenkins pipeline 생성](#)

[5.6 Gitlab webhook 설정](#)

6. AWS IoT Core

[6.1 IoT Device 등록](#)

[6.2 MQTT Client 연결\(Python\)](#)

[6.2 MQTT Client 연결\(Spring Boot\)](#)

7. Build & Distribute

[7.1 Backend - Spring Boot](#)

	Dockerfile
	Jenkinsfile
	jenkins 설정
7.2	frontend - vue
	Dockerfile
	Jenkinsfile
	jenkins setting
8.	Nginx setting
8.1	설치 및 실행
8.2	인증서 설치
9.	외부 서비스
9.1	Naver 로그인
	Springboot에 적용
9.2	Naver 문자
	네이버 sens 서비스
	API URL
	NAVER Cloud Platform 인증키 및 Signature 생성
	SENS 서비스 신청하고 서비스 ID 발급받기
	메시지
	메시지 발송
	요청 URL
	Path Variables
	Headers
	API Header
	요청 Body
	응답 Body
	응답 Status
9.3	AWS S3
	버킷 만들기
	IAM 사용자 권한 추가
	spring boot에 추가하기
9.4	공공 데이터 - 단기 예보 데이터
9.5	한강홍수통제소 오픈API- 단기 예보 데이터
9.6	카카오맵 API

1. 프로젝트 기술 스택

Frontend

- 기술 스택 및 버전
 - Node.js 16.20.1
 - Vue 3.2.36, Vue-CLI 4.5.13, Vuex 4.0.0, Vuex-persistedstate 4.1.0
 - TypeScript
- 사용 툴
 - Visual Studio Code
 - 필수 설치 Extension : Vetur, vue3 snippets, ESLint, Prettier
 - Vue.js.Devtools

Backend

- 기술 스택 및 버전
 - Spring boot 2.7.14 , MySQL 8.0.33 , Unit4, OAuth2, SpringSecurity, JDK 11.0.15,

- Nginx 1.18.0 , Jenkins 2.346.2 , AWS EC2, AWS S3,
- 사용 툴
 - IntelliJ 2022.2 , Mobaxterm V23.2, MySql workbench 8.0.33 , DBeaver 23.1.2, Postman v10.17.1

IOT

- 기술 스택 및 버전
 - Arduino(C/C++), RaspberryPi(Python 3.9)
 - AWS IoT Core, Mosquitto 3.1.1 / 2.0.16
- 사용 툴
 - Arduino IDE 2.1.1
 - Autodesk Fusion360

기타

- 형상 관리 : gitlab
- 이슈 관리 : Jira
- 커뮤니케이션 : Mattermost, notion, Webex

2. 개발 환경 Setting

2.1 Frontend

1. Node.js 설치 → NVM 설치 → 버전 변경
 - Node.js 설치

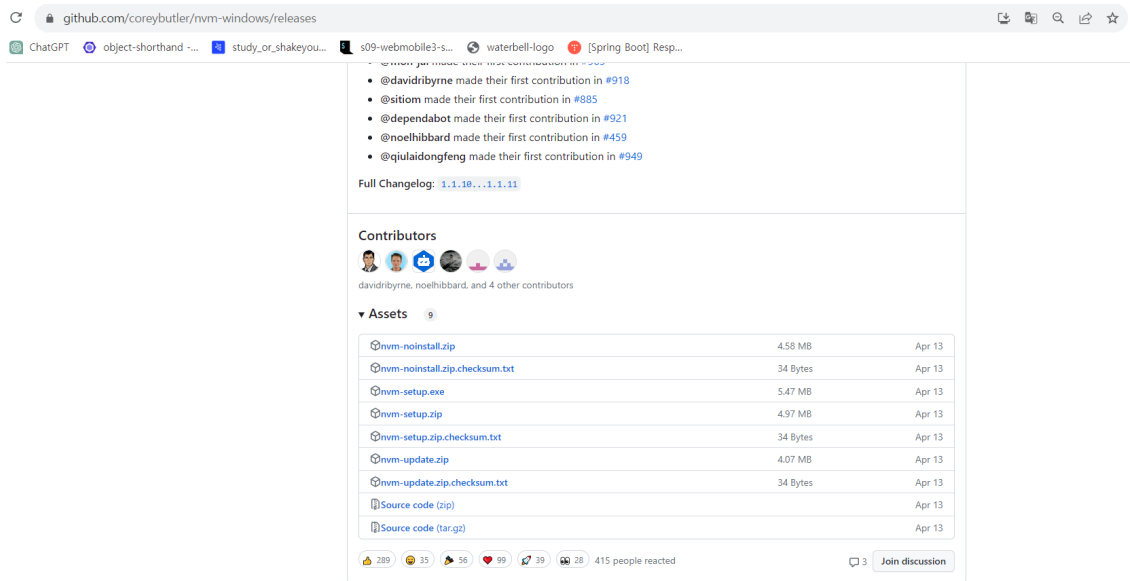
Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

다운로드 - Windows (x64)

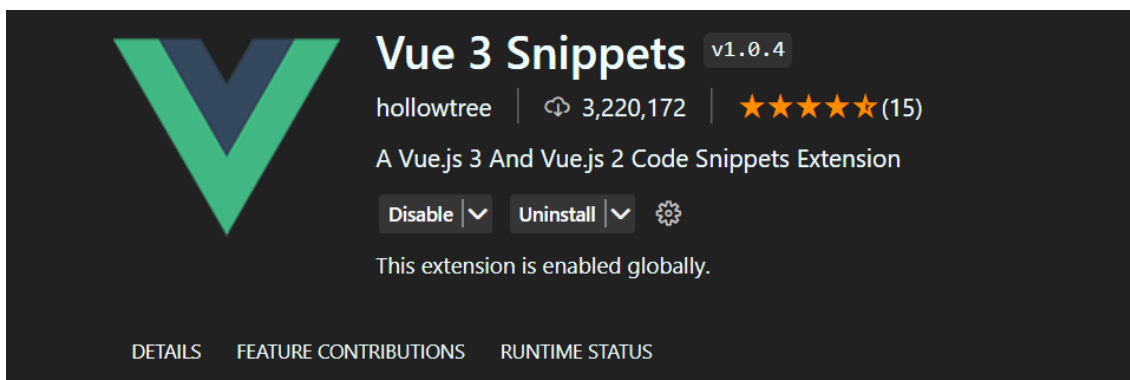
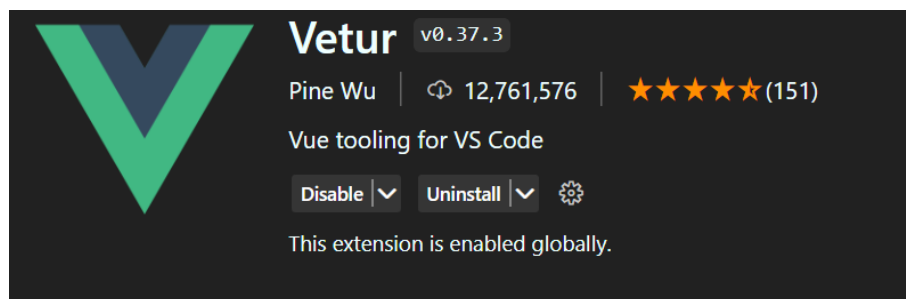
<div style="border: 2px solid red; padding: 10px; background-color: #008000; color: white;"> 18.17.1 LTS 안정적, 신뢰도 높음 </div> <p>다른 운영 체제 변경사항 API 문서</p>	<div style="padding: 10px; background-color: #008000; color: white;"> 20.5.1 현재 버전 최신 기능 </div> <p>다른 운영 체제 변경사항 API 문서</p>
---	---

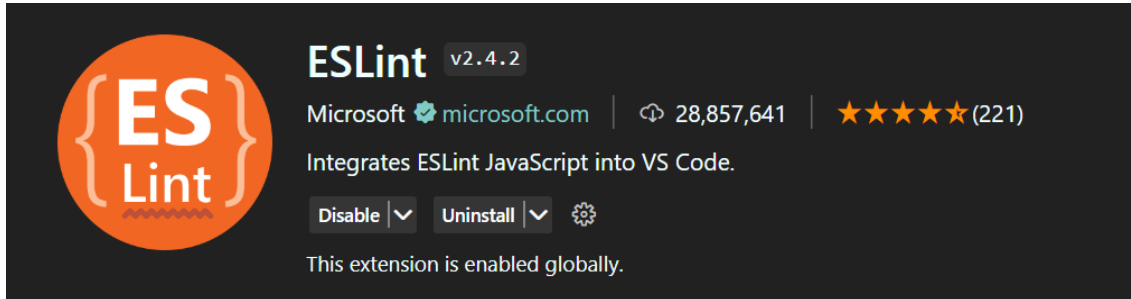
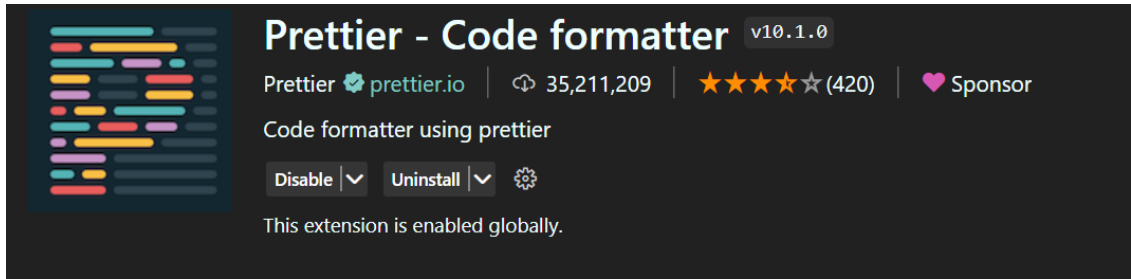
LTS 일정은 [여기서 확인하세요.](#)

- NVM 설치
 - **nvm-setup.exe** 파일로 설치

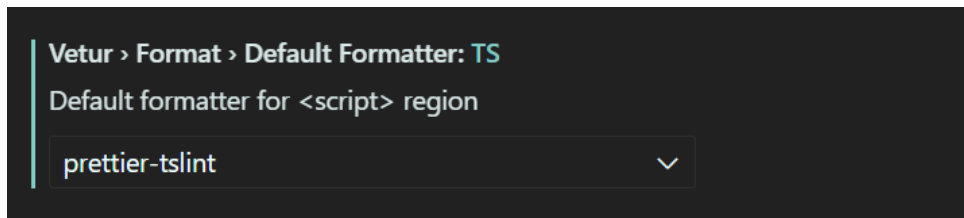
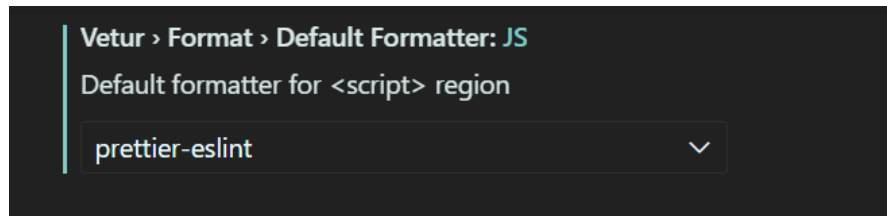


- 설치중 문구에서 멈출 경우 원인 및 대처법
 - nvm의 설치 경로 설정이 잘못되어 프로그램이 설치할 위치를 찾지 못하고 있어 명령어를 통해 수동으로 설정해주어야 함 (PC 사용자 이름에 특수기호나 공백이 있는 경우에도 발생)
 - `nvm root "C:\Users\{해당 PC username}\AppData\Roaming\nvm"`
 - `nvm install 16` : node v16 설치
 - Vue-cli 설치 : `npm install -g @vue/cli@4.5.13` / 에러 발생 시, `sudo npm install -g @vue/cli`
 - VS Code 확장 프로그램 설치
 - Vetur, vue3 snippets, ESLint, Prettier



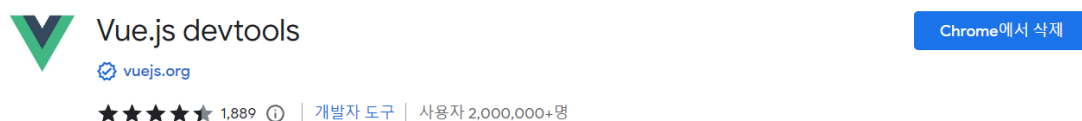


- Prettier 설정 : [settings](Ctrl+,) → Prettier 검색 → 아래 설정 변경



- 그 외 Prettier와 ESLint 설정은 각각 프로젝트 파일 내에 `.prettierrc` 와 `.eslintrc` 파일대로 설정되므로 설정 건드리지 않기. 건드릴 시 충돌로 에러 발생할 수 있음
- 다른 설정 문제로 에러 발생 시, `.prettierrc` 와 `.eslintrc` 파일 내용 참고하여 Settings 에서 Prettier과 ESLint 설정 모두 변경
- TypeScript 설치 : `npm install -g typescript`
- Vue.js.Devtools 설치 : Chrome 웹스토어 - Vue.js.Devtools 설치

홈 > 확장 프로그램 > Vue.js devtools



2.2 IoT

1. Window Mosquitto 설정

- 다운로드

- 다운로드 링크 : <https://mosquitto.org/download/>

Windows

- [mosquitto-2.0.16-install-windows-x64.exe](#) (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- [mosquitto-2.0.16-install-windows-x32.exe](#) (32-bit build, Windows Vista and up, built with Visual Studio Community 2019)

Older installers can be found at <https://mosquitto.org/files/binary/>.

See also README-windows.md after installing.

- Window 권한 설정

- 다른 ip주소의 접근을 허용하기 위해 권한 설정 필요
 - mosquitto.conf file 경로

```
cd "C:\Program Files\mosquitto"
```

- mosquitto.conf 파일 수정

```
allow_anonymous true
listener 1883 0.0.0.0
```

- 주요 명령어

- mosquitto 실행 (localonly)

```
cd "C:\Program Files\mosquitto"
mosquitto.exe -v
```

- mosquitto 실행 (config file 적용)

```
cd "C:\Program Files\mosquitto"
mosquitto -c mosquitto.conf -v
```

- mosquitto publish

```
mosquitto_pub -t "topic name" -m "message name"
```

- mosquitto subscribe

```
mosquitto_sub -t "topic name"
```

2. 라즈베리파이 Mosquitto 설정

- 설치

1. mosquitto 인증키 다운로드

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
```

2. mosquitto 저장소 패키지 등록

```
cd /etc/apt/source.list.d/  
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

3. MQTT 브로커 설치

```
sudo apt-get update  
sudo apt-cache search mosquitto  
sudo apt-get install mosquitto mosquitto-clients
```

- 라즈베리파이 ip 접속허가

- Config file 경로

```
cd /etc/mosquitto/  
sudo nano mosquitto.conf
```

- Config file 수정

```
# Place your local configuration in /etc/mosquitto/conf.d/  
#  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example  
  
pid_file /run/mosquitto/mosquitto.pid  
  
persistence true  
persistence_location /var/lib/mosquitto/  
  
log_dest file /var/log/mosquitto/mosquitto.log  
  
include_dir /etc/mosquitto/conf.d
```

아래에 두 줄 추가

```
bind_address 0.0.0.0  
allow_anonymous true
```

재실행

```
sudo systemctl restart mosquitto
```

- 주요 명령어

- 실행

```
sudo /etc/init.d/mosquitto start
```

3. 라즈베리파이 VNC연결

- vnc 설정

```
sudo raspi-config
```

Interface → VNC → enable

- 해상도 설정

```
vncserver -geometry 1280x1024
```

- vnc 에서 나온 ip로 접속

4. Arduino board 추가

- 경로
File → Preferences → Additional boards manager URLs → OK
- ESP 8266

http://arduino.esp8266.com/stable/package_esp8266com_index.json

- ESP 32

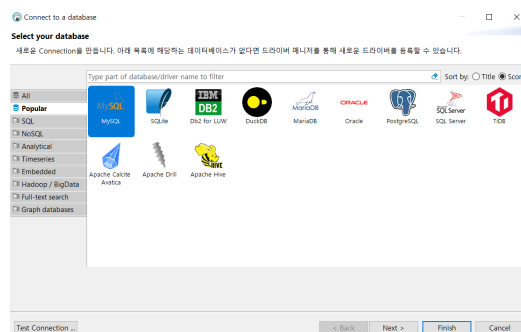
https://dl.espressif.com/dl/package_esp32_index.json

- Tools → Board → Board Manager → ESP8266/ESP32 검색 후 설치

3. DB Connection

3.1 DBeaver에 추가하기

1. 추가 버튼에서 Mysql 선택 → Next



2. EC2 계정 정보 넣기 → Ok

Connection "Waterbell-operation" configuration

MySQL connection settings

MySQL

Connection settings

Initialization

Shell Commands

Client identification

Transactions

General

Metadata

Errors and timeouts

> 데이터 편집기

> SQL 편집기

Main Driver properties SSH Proxy SSL

Server

Connect by: ☐ Host ☒ URL

URL: jdbc:mysql://i9b101.p.ssafy.io:3306/

Server Host: localhost Port: 3306

Database:

Authentication (Database Native)

Username: WaterBell

Password: ☒ Save password locally

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL Server 8.0

[You can use variables in connection parameters.](#)

Driver name: MySQL

Driver Settings Driver license

Test Connection ...

OK Cancel

- URL: jdbc:mysql://i9b101.p.ssafy.io:3306/
- Username : WaterBell
- Password: thinkerbell
- root 계정이 아닌 WaterBell 계정으로 진행

4. EC2 세팅


4.1 EC2 접속하기

- MobaXterm 활용
- SSH 선택
- Remote host : i9b101.p.ssafy.io
- User private Key : 발급받은 key 입력

i9B101T.pem

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings



Remote host * i9b101.p.ssafy.io ☐ Specify username  Port 22


Advanced SSH settings Terminal settings Network settings Bookmark settings

☒ X11-Forwarding ☒ Compression Remote environment: Interactive shell

Execute command: ☐ Do not exit after command ends

SSH-browser type: SFTP protocol ☐ Follow SSH path (experimental)

☒ Use private key C:\Users\SSAFY\Downloads\I9B101  

 Expert SSH settings

Execute macro at session start: <none>

OK Cancel

4.2 방화벽 설정하기

`sudo ufw allow {포트 번호}`

```
sudo ufw allow ssh
sudo ufw allow http
sudo ufw allow https
```

4.3 DB 세팅

1. Mysql 설치

먼저 서버 업데이트 : `sudo apt-get update`

설치 : `sudo apt install mysql-server`

방화벽 포트 허용 : `sudo ufw allow mysql`

2. 상태 확인

```
sudo systemctl status mysql
```

3. root 계정 접속

설치하면서 별도로 비밀번호를 설정하지 않은 경우, 비밀번호 입력 창에서 엔터를 입력

```
sudo mysql -u root -p
```

4. 데이터베이스 생성

```
create database WaterBell;
```

`show databases;` 생성된 DB확인

5. User 생성

```
use mysql;  
create user 'WaterBell'@'%' identified by 'thinkerbell';
```

mysql에서는 %가 와일드카드, 어느 호스트에서든지 접속이 가능하다

6. 생성한 유저에게 권한 부여하기

```
grant all on WaterBell.* to 'WaterBell'@'%';
```

권한 확인하기

```
show grants for 'WaterBell'@'%';
```

참고: mysql에서 밖으로 나가기: `exit`

7. 외부 접속 허용하기

```
sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

sudo로 접근하지 않으면 수정 불가

bind-address를 0.0.0.0으로 수정하기

```
#bind-address            = 127.0.0.1  
bind-address             = 0.0.0.0  
mysqlx-bind-address      = 127.0.0.1  
#
```

8. Mysql 재실행

```
sudo service mysql restart
```

서버 재 시작 시 mysql 자동 활성화 : `sudo systemctl enable mysql`

9. 루트 계정 비밀번호 설정

```
mysql -u root -p
```

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';
```

4.4 Redis 세팅

1. Redis 설치

먼저 서버 업데이트 : `sudo apt-get update`

설치 : `sudo apt-get install redis-server`

방화벽 포트 허용 : `sudo ufw allow redis`

2. Redis conf 설정 변경

```
sudo vi /etc/redis/redis.conf
```

bind 127.0.0.1 ::1 → bind 0.0.0.0 ::1으로 수정

```
bind 0.0.0.0 ::1
```

비밀번호 설정

```
requirepass password
```

외부접속 허용

```
protected-mode no
```

포트 설정(redis 기본포트 6379)

```
port 6379
```

3. Redis 재시작

```
sudo systemctl restart redis-server.service
```

4. redis 접속

```
redis-cli -h {hostname} -p {port} -a {password}
```

4.5 Docker 설치

- 공식 문서를 먼저 확인

Install Docker Engine on Ubuntu

Jumpstart your client-side server applications with Docker Engine on Ubuntu. This guide details prerequisites and multiple methods to install.

 <https://docs.docker.com/engine/install/ubuntu/>



- 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=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
"$(. /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
```

4.6 Jenkins 설치하기

- docker-compose를 사용하여 jenkins container를 실행하기

1. jenkins container를 실행시킬 docker-compose 파일을 만들

```
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
```

- jenkins 기본 포트는 8080인데 9090포트 사용하도록 지정해줌
- 현재 버전은 jdk11을 사용할 경우 - 현재 서버 버전

2. 정상적으로 container가 실행되고 있는지 확인

```
sudo docker ps -a
```

```
bcaa359676a4 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
ubuntu@ip-172-26-12-252:~$
```

3. jenkins container에 접속하여 Docker를 설치

a. jenkins 컨테이너 내부에 접속

```
ubuntu@ip-172-26-12-252:~$ sudo docker exec -it jenkins /bin/bash
root@bcaa359676a4:/#
```

b. docker 설치

```
sudo docker exec -it jenkins bin/bash

## root~~ 나오면
## jenkins 컨테이너 내부에 접속 완료

##docker 설치
apt-get update
apt-get install ca-certificates curl gnupg lsb-release
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
apt-get update

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

4. Jenkins 접속

http://i9b101.p.ssafy.io:9090/

5. Jenkins 세팅

5.1 초기 계정 설정

1. 초기 접속화면 Unlock Jenkins

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

2. install suggested plusins 선택

3. Create First Admin User

- a. 계정명 : WaterBell
- b. 암호 : thinkerbell
- c. 이름 : Waterbell

4. Jenkins Url

- a. default사용해도 됨,

5.2 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)

5.3 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

User Settings

- Profile
- Account
- Applications
- Chat
- Access Tokens**
- Emails
- Password
- Notifications
- SSH Keys
- GPG Keys
- Preferences
- Comment Templates
- Active Sessions
- Authentication Log

User Settings · Access Tokens

Search page

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Add a personal access token

Enter the name of your application, and we'll return a unique personal 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

2023-09-15

Select scopes

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

- ☐ **api**
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **read_api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **read_user**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- ☐ **read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ☐ **write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Create personal access token

5.4 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

WaterBell

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

https://lab.ssafy.com/

Credentials ?
API Token for accessing GitLab

GitLab API token

Add +

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

Success

Test Connection

5.5 Jenkins pipeline 생성

- + 새로운 Item → 이름 입력, Pipeline 선택 → ok
- 프론트와 백엔드 2개를 만들어야 함

상세 내용 입력

All +

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
✓	☀	WaterBell-frontend-pipeline	6 hr 2 min #233	5 days 17 hr #78	1 min 51 sec ▶
✓	☀	WaterBell-Pipeline	6 hr 5 min #404	3 days 6 hr #294	1 min 2 sec ▶

아이콘: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

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

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☒ Build when a change is pushed to GitLab. GitLab webhook URL: <http://9b101.p.ssafy.io:9090/waterbell/project/WaterBell-Pipeline> ?

Enabled GitLab triggers

☒ Push Events ?

☐ Push Events in case of branch delete ?

☒ Opened Merge Request Events ?

☐ Build only if new commits were pushed to Merge Request ?

☐ Accepted Merge Request Events ?

☐ Closed Merge Request Events ?

Rebuild open Merge Requests ?

Never

☒ Approved Merge Requests (EE-only) ?

☒ Comments ?

Comment (regex) for triggering a build ?

Jenkins please retry a build

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 ?

☐ Filter merge request by label

Secret token ?

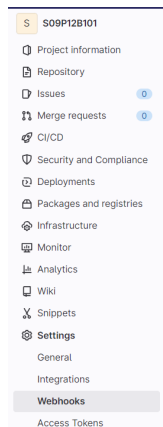
2b9d270a9c20fa96ed67bb524bbbc763

Generate

→ generate를 눌러 Secret token생성

5.6 Gitlab webhook 설정

- jenkins 작업물의 변화를 감지하여 build, run 하기 위해서는 webHook 필수!
- gitlab project → settings → webhooks



- url : <http://9b101.p.ssafy.io:9090/project/WaterBell-frontend-pipeline>
- secret token: jenkins System에서 받은 token 입력
- merge를 할때마다 요청

URL

http://i9b101.p.ssafy.io:9090/project/WaterBell-frontend-pipeline

URL must be percent-encoded if it contains one or more special characters.

☒ Show full URL

☐ Mask portions of URL

Do not show sensitive data such as tokens in the UI.

Secret token

.....

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.

Trigger

☐ Push events

☐ Tag push events

A new tag is pushed to the repository.

☐ Comments

A comment is added to an issue or merge request.

☐ Confidential comments

A comment is added to a confidential issue.

☐ Issues events

An issue is created, updated, closed, or reopened.

☐ Confidential issues events

A confidential issue is created, updated, closed, or reopened.

☒ Merge request events

A merge request is created, updated, or merged.

☐ Job events

A job's status changes.

- ssl verification


SSL verification

☒ Enable SSL verification

Save changes

Test ▾

- Test : 200이어야 함!

 Hook executed successfully: HTTP 200

Save changes

Test ▾

6. AWS IoT Core

6.1 IoT Device 등록

1. 디바이스 연결 (라즈베리파이)

링크 : <https://ap-southeast-2.console.aws.amazon.com/iot/home?region=ap-southeast-2#/connectdevice/>

1) device 등록

사물 속성

☒ 새 사물 생성

☐ 기존 사물 선택

사물 이름

문자, 숫자, 하이픈, 콜론 또는 밑줄만 포함하는 고유한 이름을 입력합니다. 사물 이름에는 공백을 포함할 수 없습니다.

2) device 환경 설정

→ Linux / Python 설정

플랫폼 및 SDK

디바이스에 사용할 플랫폼 OS 및 AWS IoT 디바이스 SDK를 선택합니다.

디바이스 플랫폼 운영 체제

AWS에 연결할 디바이스에 설치된 운영 체제입니다.

☒ Linux/macOS
Linux 버전: any
macOS 버전: 10.13+

☐ Windows
버전 10

AWS IoT 디바이스 SDK

디바이스가 지원하는 언어로 된 디바이스 SDK를 선택합니다.

☐ Node.js
버전 10+
Node.js 및 npm 설치 필요

☒ Python
버전 3.6+
Python 및 Git 설치 필요

☐ Java
버전 8
Java JDK, Maven 및 Git 설치 필요

3) device에 인증키 다운

연결 키트

인증서

rrr.cert.pem

메시지 전송 및 수신 스크립트

start.sh

프라이빗 키

rrr.private.key

정책

rrr-Policy

[정책 보기](#)

AWS IoT 디바이스 SDK

Python

다운로드

디바이스의 브라우저에서 이를 실행하는 경우 연결 키트를 다운로드하면 브라우저의 다운로드 폴더에 다운로드됩니다.

디바이스의 브라우저에서 이를 실행하지 않는 경우 1단계에서 디바이스를 준비할 때 테스트한 방법을 사용하여 브라우저의 다운로드 폴더에서 디바이스로 연결 키트를 전송해야 합니다.

 [연결 키트 다운로드](#)

4) device 압축 풀기 및 실행

```
unzip connect_device_package.zip
chmod +x start.sh
./start.sh
```

6.2 MQTT Client 연결(Python)

```
THING_NAME = 'rasp'
ENDPOINT = "<end point>"
CERTPATH = "<경로>/<thingname>.cert.pem" # cert파일 경로
KEYPATH = "<경로>/<thingname>.private.key" # key 파일 경로
CAROOTPATH = "<경로>/root-CA.crt" # RootCaPem 파일 경로
```

- Endpoint
 - AWS IoT > 설정

설정 정보

디바이스 데이터 엔드포인트 정보



디바이스는 계정의 디바이스 데이터 엔드포인트를 사용하여 AWS에 연결할 수 있습니다.

각 사물에는 이 엔드포인트에서 사용할 수 있는 REST API가 있습니다. MQTT 클라이언트와 [AWS IoT 디바이스 SDK](#) 도 이 엔드포인트를 사용합니다.

엔드포인트



보안 정책 선택 [정보](#)

TLS 버전 및 지원되는 암호 제품군과 같은 TLS 설정을 사용자 지정하려면 보안 정책을 선택합니다.

IoTSecurityPolicy_TLS13_1_2_2022_10



[보안 정책 비교](#)

- client 생성

```
IoTCore = mqtt.Client()
IoTCore.on_connect = on_connect
IoTCore.tls_set(CAROOTPATH, certfile= CERTPATH, keyfile=KEYPATH, tls_version=ssl.PROTOCOL_TLSv1_2, ciphers=None)
IoTCore.connect(ENDPOINT, 8883, 60)
```

6.2 MQTT Client 연결(Spring Boot)

1. dependencies(gradle)

```
//aws iot connect
implementation 'com.amazonaws:aws-iot-device-sdk-java:1.3.9'
implementation 'com.amazonaws:aws-iot-device-sdk-java-samples:1.3.9'
```

2. config

```
String clientEndpoint = "<Endpoint>";
String clientId = "<Thingname>";
String awsAccessKeyId = "";
String awsSecretAccessKey = "";
```

```
new AWSIotMqttClient(clientEndpoint, clientId, awsAccessKeyId, awsSecretAccessKey, null)
```

3. AWS 자격증명

- IAM → 액세스 키 만들기

엑세스 키 (2)

엑세스 키를 사용하여 AWS CLI, AWS Tools for PowerShell, AWS SDK 또는 직접 AWS API 호출을 통해 AWS에 프로그래밍 방식 호출을 전송합니다. 한 번에 최대 두 개의 엑세스 키(활성 또는 비활성)를 가질 수 있습니다. [자세히 알아보기](#)

작업 ▼

엑세스 키 만들기

	엑세스 키 ID	생성 날짜	마지막으로 사용한 엑세스 키	마지막으로 사용된 리전	마지막으로 사용한 서비스	상태
<input type="radio"/>		3일 전	3일 전	us-east-2	iot	✔ Active
<input type="radio"/>		어제	12분 전	us-east-2	iot	✔ Active

7. Build & Distribute

7.1 Backend - Spring Boot

Dockerfile

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

Jenkinsfile

```
pipeline {
    agent any

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

                    sh '''
                        echo 'Deploy'
                        docker stop WaterBell
                        docker rm WaterBell
                        docker run -d -p 8080:8080 --name WaterBell docker-springboot:0.0.1
                    '''
                }
            }
        }
    }
}
```

jenkins 설정

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://lab.ssafy.com/s09-webmobile3-sub2/S09P12B101.git

Credentials ?

gyrud0482@naver.com/*****

Add

고급

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(자동)

Additional Behaviours

Add

Script Path ?

backend/fcc/Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

- jenkinsfile과 dockerfile이 있는 경로 지정
- 지금 빌드 실행하여 확인

Pipeline WaterBell-Pipeline

Stage View

		Declarative: Checkout SCM	Springboot build	Dockerimage build	Deploy
Average stage times: (Average full run time: ~35s)		514ms	24s	3s	1s
#4006	No Changes	357ms	19s	3s	1s
#4007	No Changes	429ms	24s	2s	1s

7.2 frontend - vue

Dockerfile

```
FROM node:16.20.1
WORKDIR /app
COPY package*.json ./
RUN npm install
RUN npm install -g serve
COPY . .
RUN NODE_ENV=production npm run build
ENTRYPOINT ["serve", "-s", "dist"]
```

Jenkinsfile

```
pipeline {
    agent any

    tools {nodejs "node"}

    stages {
        stage('prepare') {
            steps {
                dir('frontend/waterbell'){
                    sh 'npm install'
                }
            }
        }
        stage('build') {
            steps {
                dir('frontend/waterbell'){
                    sh 'npm run build'

                    sh '''
                    echo 'Dockerimage build'
                    docker build -t docker-vue:0.0.1 .
                    '''
                }
            }
        }
        stage('Deploy') {
            steps {
                dir('frontend/waterbell'){
                    sh '''
                    echo 'Deploy'
                    docker stop WaterBellFront
                    docker rm WaterBellFront
                    docker run -d -p 3000:3000 --name WaterBellFront docker-vue:0.0.1
                    '''
                }
            }
        }
    }
}
```


jenkins setting

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://lab.ssafy.com/s09-webmobile3-sub2/S09P12B101.git

Credentials ?

gyrud0482@naver.com/*****

Add

고급

Add Repository

Branches to build

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(자동)

Additional Behaviours

Add

Script Path ?

backend/fcc/Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

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

NodeJS
Name
node

☒ Install automatically ?

Install from nodejs.org

Version
NodeJS 16.20.1

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

- 지금 빌드 실행하여 확인

Pipeline WaterBell-frontend-pipeline

Stage View

			Declarative: Checkout SCM	Declarative: Tool Install	prepare	build	Deploy
Average stage times: (Average full run time: ~1min 18s)			554ms	89ms	7s	1min 7s	1s
#238	8월 16일 22:59	No Changes	352ms	77ms	12s	26s	983ms
#237	8월 16일 22:58	No Changes	422ms	85ms	5s	29s	1s

8. Nginx setting

- 엔진엑스(Nginx)는 동시접속 처리에 특화된 웹 서버이다.
- 트래픽이 많은 웹 사이트를 호스팅 하는데 주로 사용되는 가볍고 Apache보다 동작이 단순하며 강력하다.

8.1 설치 및 실행

1. 서버의 패키지 목록 업데이트

- a. `sudo apt update`
- b. `sudo apt upgrade`
- c. `sudo apt autoremove`

2. Nginx 설치

- a. `sudo apt install nginx`

3. 실행

- a. `sudo service start nginx`
- b. `sudo service status nginx`

4. 환경 설정

- a. `sudo vi /etc/nginx/sites-available/nginx.conf`
- b. 아래 내용을 붙여 넣음

```
location / {
    root /jenkins/workspace/WaterBell-frontend-pipeline/frontend/waterbell/dist;
    try_files $uri $uri/ /index.html;
}

location /api/ {
    proxy_pass http://backend/;

    charset utf-8;

    proxy_http_version 1.1;
    proxy_set_header    Connection "upgrade";
    proxy_set_header    Upgrade $http_upgrade;
    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    X-Forwarded-Proto $scheme;
    proxy_set_header    X-NginX-Proxy true;
}
}
```

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

8.2 인증서 설치

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!!

9. 외부 서비스

9.1 Naver 로그인

- 어플리케이션 이름 : WaterBell
- 사용 API : 이름, 이메일주소

애플리케이션 이름	WaterBell																														
카테고리	기타																														
	선택하세요																														
	내이버 로그인																														
	제공 정보 선택(이름과 식별자는 기본 정보로 제공)																														
	필수 항목은 개인정보보호법 제3조 제1항, 제16조 제1항 등에 따라 서비스 제공을 위해 필요한 최소한의 개인정보만을 선택해야 합니다.																														
사용 API	<table><thead><tr><th>권한</th><th>필수</th><th>추가</th></tr></thead><tbody><tr><td>회원인증</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>연락처 이메일 주소</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>별명</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>프로필 사진</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>성명</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>성별</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>연령대</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>출생연도</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>중간인증번호</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>	권한	필수	추가	회원인증	<input checked="" type="checkbox"/>	<input type="checkbox"/>	연락처 이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>	별명	<input type="checkbox"/>	<input type="checkbox"/>	프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>	성명	<input type="checkbox"/>	<input type="checkbox"/>	성별	<input type="checkbox"/>	<input type="checkbox"/>	연령대	<input type="checkbox"/>	<input type="checkbox"/>	출생연도	<input type="checkbox"/>	<input type="checkbox"/>	중간인증번호	<input type="checkbox"/>	<input type="checkbox"/>
권한	필수	추가																													
회원인증	<input checked="" type="checkbox"/>	<input type="checkbox"/>																													
연락처 이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>																													
별명	<input type="checkbox"/>	<input type="checkbox"/>																													
프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>																													
성명	<input type="checkbox"/>	<input type="checkbox"/>																													
성별	<input type="checkbox"/>	<input type="checkbox"/>																													
연령대	<input type="checkbox"/>	<input type="checkbox"/>																													
출생연도	<input type="checkbox"/>	<input type="checkbox"/>																													
중간인증번호	<input type="checkbox"/>	<input type="checkbox"/>																													

- 서비스 url : <https://i9b101.p.ssafy.io/>
- Callback Url
 - 운영 : <https://i9b101.p.ssafy.io/auth/naver> 필수!!!
 - 개발 : <http://localhost:3000/auth/naver> , <http://i9b101.p.ssafy.io:3000/auth/naver>

로그인 오픈 API
서비스 환경 ⓘ

PC 웹

서비스 URL

서비스 URL에서: (O) http://naver.com (X) http://www.naver.com
 서비스 URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때까지 네이버 로그인 사용이 일시적으로 제한됩니다.
 불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다.
 서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 **네이버 로그인** 배지가 노출됩니다.

네이버 로그인
Callback URL (최대 5개)

http://localhost:3000/auth/naver	-
https://i9b101.p.ssafy.io/auth/naver	-
http://i9b101.p.ssafy.io:3000/auth/naver	+

텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
 Callback URL은 네이버 로그인 후 이동할 페이지 URL입니다. Callback URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때까지 네이버 로그인 사용이 일시적으로 제한됩니다.
 입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.

로그 이미지 ⓘ

- client ID, Secret

애플리케이션 정보

Client ID	<input type="text" value="dlwg6T0yWa9t8y2yMsHJ"/>
Client Secret	<div style="border: 1px solid #ccc; padding: 2px 5px; display: flex; align-items: center;"> • • • • • </div> <div style="text-align: center; margin-top: 5px;"> <input type="button" value="보기"/> </div>

Springboot에 적용

```

spring.security.oauth2.client.registration.naver.client-id={lient-id}
spring.security.oauth2.client.registration.naver.client-secret={client-secret}
  
```

9.2 Naver 문자

네이버 sens 서비스

Simple & Easy Notification Service Update

SMS, PUSH, 알림톡 등 메시지 알림 기능을 구현하는 서비스

[이용 신청하기](#)[요금 계산하기](#)[특징](#)[상세 기능](#)[요금](#)[사용 가이드](#)

알림 및 메시지 기능을 구현하는 쉬운 방법

웹/모바일, iOS / Android 상관없이 다양한 유형의 알림 기능을 적용할 수 있는 서비스입니다.

효율적인 프로젝트 관리

프로젝트를 생성하고 메타 정보를 입력하면 메시지 및 알림 전송 기능을 즉시 사용할 수 있습니다. 각 단계에 필요한 정보를 입력한 후에는 메시지

다양한 발송 옵션

예약 발송 기능을 통해 메시지 발송 스케줄을 관리할 수 있습니다. 또한 알림톡/친구톡에 대한 SMS Fail over 기능을 지원하여, 카카오톡이 설치

다양한 연동 방법

네이버 클라우드 플랫폼에서 제공하는 서비스와의 연동은 물론 APNS(iOS), GCM/FCM(Android), SMS, LMS 외에도 카카오톡 비즈 메시지(알림톡,

API URL

`https://sens.apigw.ntruss.com/sms/v2`

NAVER Cloud Platform 인증키 및 Signature 생성

ncloud 회원가입(<https://www.ncloud.com/>)

인증키 관리 : 네이버 클라우드 플랫폼 메인 페이지 > 마이페이지 > 인증키 관리

> 신규 API 인증키 생성

NAVER
CLOUD PLATFORM

소개 서비스 솔루션 마켓플레이스 요금 고객지원 파트너 가이드센터 마이페이지

문의하기

로그인

이용관리 계정관리 결제관리 나의 문의내역 알림 관리 솔루션 이용 현황

계정 관리

회원정보 변경 비밀번호 변경 파트너 관리 보안 설정 SNS 연동 계정 변경 **인증키 관리** 회원 탈퇴

네이버 클라우드 플랫폼은 제공하는 서비스를 안전하게 이용하도록 회원별 API 인증키를 발급하고 있습니다. API 인증키는 API를 호출한 사용자가 권한을 가진 사용자인지 식별하는 도구입니다.

API 인증키 이용안내

- API 인증키를 도용되었다고 의심될 때는 기존 API 인증키를 삭제하고 새 API 인증키를 생성하세요.
- 새 API 인증키를 생성한 다음에는 반드시 사용하고 있는 서비스에 변경된 API 인증키를 적용해야 합니다.
- API 인증키를 사용하지 않을 때는 '사용 중지'로 설정할 수 있으며, 삭제는 '사용 중지' 상태에서에만 가능합니다.

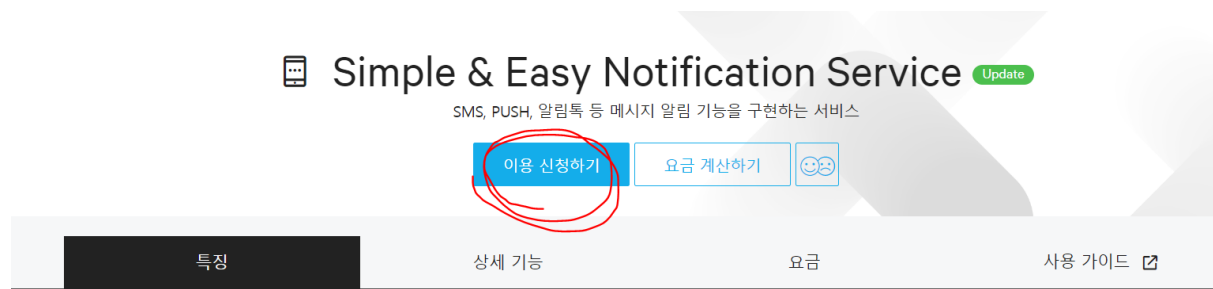
API 인증키 관리

[신규 API 인증키 생성](#)

Access Key ID	Secret Key	생성일자	상태	관리
---------------	------------	------	----	----

SENS 서비스 신청하고 서비스 ID 발급받기

서비스 이용 신청하기



콘솔 > SENS 메뉴에서 프로젝트 생성 후, 우측 열쇠 아이콘을 클릭

Project



메시지

메시지 발송

SMS / LMS / MMS 메시지를 발송

요청 URL

POST <https://sens.apigw.ntruss.com/sms/v2/services/{serviceId}/messages>

Content-Type: application/json; charset=utf-8
x-ncp-apigw-timestamp: {Timestamp}
x-ncp-iam-access-key: {Sub Account Access Key}
x-ncp-apigw-signature-v2: {API Gateway Signature}

Path Variables

항목	Mandatory	Type	설명	비고
serviceId	Mandatory	String	서비스 아이디	프로젝트 등록 시 발급받은 서비스 아이디

Headers

API Header

항목	Mandatory	설명
Content-Type	Mandatory	요청 Body Content Type을 application/json으로 지정 (POST)
x-ncp-apigw-timestamp	Mandatory	- 1970년 1월 1일 00:00:00 협정 세계시(UTC)부터의 경과 시간을 밀리초 (Millisecond)로 나타냄- API Gateway 서버와 시간 차가 5분 이상 나는 경우 유효하지 않은 요청으로 간주
x-ncp-iam-access-key	Mandatory	포털 또는 Sub Account에서 발급받은 Access Key ID
x-ncp-apigw-signature-v2	Mandatory	- 위 예제의 Body를 Access Key Id와 맵핑되는 SecretKey로 암호화한 서명- HMAC 암호화 알고리즘은 HmacSHA256 사용

요청 Body

```
{
  "type": "(SMS | LMS | MMS)",
  "contentType": "(COMM | AD)",
  "countryCode": "string",
  "from": "string",
  "subject": "string",
  "content": "string",
  "messages": [
    {
      "to": "string",
      "subject": "string",
      "content": "string"
    }
  ],
  "files": [
    {
      "fileId": "string"
    }
  ],
  "reserveTime": "yyyy-MM-dd HH:mm",
  "reserveTimeZone": "string"
}
```

항목	Mandatory	Type	설명	비고
type	Mandatory	String	SMS Type	SMS, LMS, MMS (소문자 가능)
contentType	Optional	String	메시지 Type	- COMM: 일반메시지- AD: 광고메시지- default: COMM
countryCode	Optional	String	국가 번호	- SENS에서 제공하는 국가로의 발송만 가능- default: 82- 국제 SMS 발송 국가 목록
from	Mandatory	String	발신번호	사전 등록된 발신번호만 사용 가능
subject	Optional	String	기본 메시지 제목	LMS, MMS에서만 사용 가능- LMS, MMS: 최대 40byte
content	Mandatory	String	기본 메시지 내용	- SMS: 최대 80byte- LMS, MMS: 최대 2000byte
messages	Mandatory	Object	메시지 정보	- 아래 항목 참조 (messages.XXX)- 최대 100개
messages.to	Mandatory	String	수신번호	불임표 (-)를 제외한 숫자만 입력 가능
messages.subject	Optional	String	개별 메시지 제목	LMS, MMS에서만 사용 가능- LMS, MMS: 최대 40byte
messages.content	Optional	String	개별 메시지 내용	- SMS: 최대 80byte- LMS, MMS: 최대 2000byte
files.fileId	Optional	String	파일 아이디	MMS에서만 사용 가능 파일 업로드 참조
reserveTime	Optional	String	예약 일시	메시지 발송 예약 일시 (yyyy-MM-dd HH:mm)
reserveTimeZone	Optional	String	예약 일시 타임존	- 예약 일시 타임존 (기본: Asia/Seoul)- 지원 타임존 목록 - TZ database name 값 사용

응답 Body


```
{
  "requestId": "string",
  "requestTime": "string",
  "statusCode": "string",
  "statusName": "string"
}
```

항목	Mandatory	Type	설명	비고
requestId	Mandatory	String	요청 아이디	
requestTime	Mandatory	DateTime	요청 시간	yyyy-MM-dd'T'HH:mm:ss.SSS
statusCode	Mandatory	String	요청 상태 코드	- 202: 성공- 그 외: 실패- HTTP Status 규격을 따름
statusName	Mandatory	String	요청 상태명	- success: 성공- fail: 실패

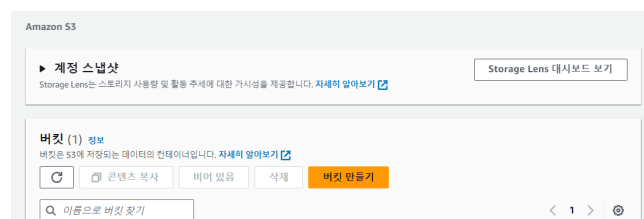
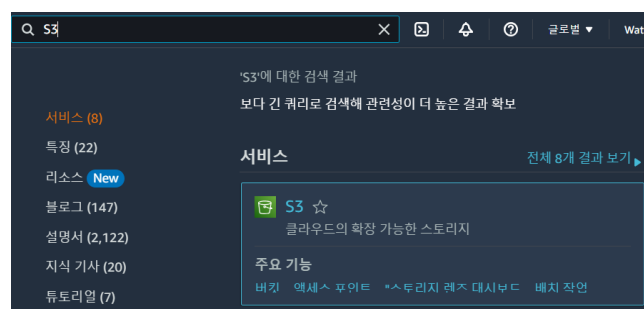
응답 Status

HTTP Status	Desc
202	Accept (요청 완료)
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
429	Too Many Requests
500	Internal Server Error

9.3 AWS S3

버킷 만들기

- 회원 가입 → S3 검색 후 선택 → 버킷 만들기

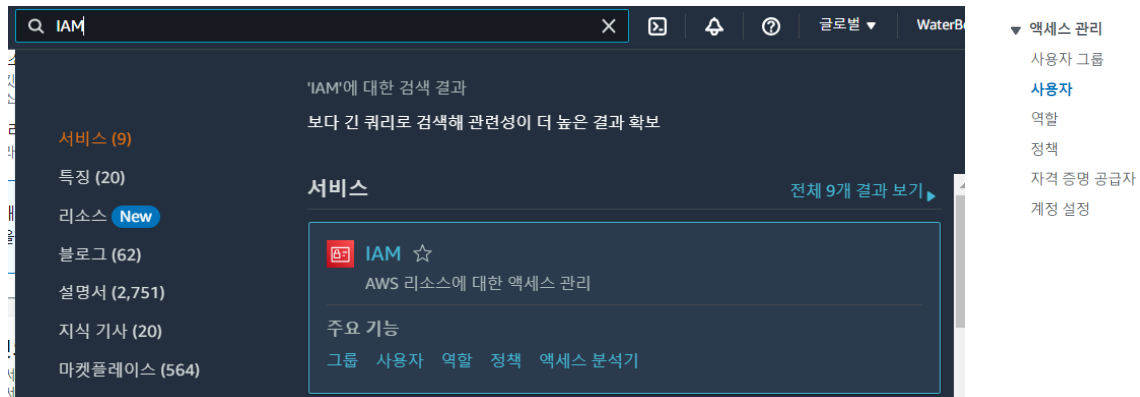


- 버킷 이름

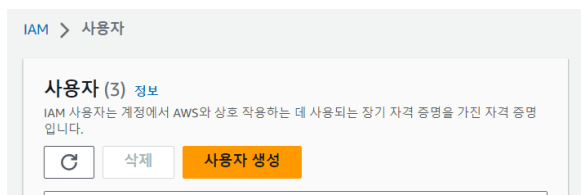
- AWS 리전 : ap-northeast-2
- 객체 소유권 : ACL 활성화 됨
- 모든 퍼블릭 액세스 차단 해제 → 퍼블릭 상태가 될 수 있음을 알고 있습니다. 체크
- 버전관리 : 비활성화

IAM 사용자 권한 추가

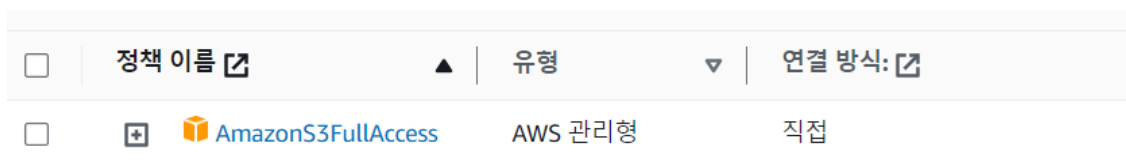
- IAM 검색 후 서비스 선택 → 왼쪽 액세스 관리 → 사용자 선택



- 사용자 생성 클릭



- 사용자 이름 입력
- 권한 옵션 : 직접 정책 연결
- 권한 정책



- 생성 후, 액세스키 비밀엑세스 키 저장 필수!!!!!!

spring boot에 추가하기

- spring boot의 application.properties에 입력

```
cloud.aws.credentials.accessKey={엑세스키}
cloud.aws.credentials.secretKey={비밀엑세스키}
cloud.aws.s3.bucket=waterbell-origin-s3
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false
```

- build.gradle에 추가

```
implementation "com.amazonaws:aws-java-sdk-s3:1.12.281"
```

9.4 공공 데이터 - 단기 예보 데이터

- 공공데이터 포털 이동

공공데이터 포털

국가에서 보유하고 있는 다양한 데이터를 『공공데이터의 제공 및 이용 활성화에 관한 법률(제11956호)』에 따라 개방하여 국민들이 보다 쉽고 용이하게 공유·활용할 수 있도록 공공데이터(Dataset)와 Open API로 제공하는 사이트입니다.

<https://www.data.go.kr/>



- 기상청 단기예보 조회 서비스 활용 신청

과학기술

기상청

활용신청

[승인] 기상청_단기예보 ((구)_동네예보) 조회서비스

계정 개발

신청일 2023-08-04

만료예정일 2025-08-04

- 단기 예보 데이터 조회 url

http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtFcst

- 요청 메시지에 인증키 필요

항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터
serviceKey	인증키	100	1	인증키 (URL Encode)
numOfRows	한 페이지 결과 수	4	1	10
pageNo	페이지 번호	4	1	1
dataType	응답자료형식	4	0	XML
base_date	발표일자(필수)	8	1	20210628
base_time	발표시각(필수)	4	1	0630
nx	예보지점 X 좌표(필수)	2	1	55
ny	예보지점 Y 좌표(필수)	2	1	127

- 날씨 예보 시 사용한 인증키
- 강수량 예보 시 사용한 인증키

서비스정보

데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0
API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다. 포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	k5EKhcVcFEXIn2mnb2nd7hZtlbvquM7M0AbKZeLZmT58550frSYtSymQCTNjpqQNDu0S7bfuYN0Q18CYRQzg%3D%3D
일반 인증키 (Decoding)	k5EKhcVcFEXIn2mnb2nd7hZtlbvquM7M0AbKZeLZmT58550frSYtSymQCTNjpqQNDu0S7bfuYN0Q18CYRQzg==

서비스정보

데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0
API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다. 포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	v1eq8lhYcwTLBo4wgQV%2BhZ5AEd%2BjklDa8XsUmabyU0gdDxPWP0A%2FVTOXGT5KktfCS5yTQKpaVLtekYDVrjI0Q%3D%3D
일반 인증키 (Decoding)	v1eq8lhYcwTLBo4wgQV+hZ5AEd+jklDa8XsUmabyU0gdDxPWP0A/VTOXGT5KktfCS5yTQKpaVLtekYDVrjI0Q==

9.5 한강홍수통제소 오픈API- 단기 예보 데이터

- 한강홍수통제소 홈페이지 이동 > 정보공개 > 오픈 API

한강홍수통제소 홈페이지를 방문해 주셔서 고맙습니다 : Welcom to HRFco(Han river flood control office)
홍수 통제현황, 수문자료, 홍수예경보, 간행물, 한강, 임진강, 안성천 기상특성 등 안내
<https://www.hrfco.go.kr/main.do>

- 인증키 발급 요청(아래 메일 제외하고 요청하고, 인증키 메일 회신 안 올 경우 전화로 요청하기)

▼ 인증키발급

인증키발급 안내

- 한강홍수통제소 오픈플랫폼의 새로운 수문자료 서비스를 개발하기 위해서는 하나의 인증키 값을 필요로 합니다.
- 인증키 값을 받기 위해서는 사용하고자 하는 사이트 URL(IP)을 입력하신 후 인증키를 발급 받으시기 바랍니다.
- 인증키 발급요청시, 등록된 신청자의 이메일 계정으로 메일이 발송됩니다.
- 메일에서 [인증키 사용] 버튼을 클릭하여 인증키를 승인 후 사용 할 수 있습니다.
- 1분당 1000건 이상 API를 호출 하고, 하루 3회 이상 발생시 인증키가 차단되며, 차단 정보를 이메일로 발송 됩니다.
- 1년 이상 미 사용 인증키는 휴면 상태로 전환 되며, 인증키 관리에서 휴면 상태 해제 인증 절차를 거쳐야 재 사용 할 수 있습니다.

*다음메일/한메일/Gmail을 사용하실 경우 정상적인 메일 발송이 되지 않을 수 있습니다.

• 데이터 조회 URL

`http://api.hrfco.go.kr/{{ServiceKey}}/waterlevel/info.json`

◦ 데이터 조회 결과 예시

`http://api.hrfco.go.kr/52832662-D130-4239-9C5F-730AD3BE6BC6/waterlevel/info.json`

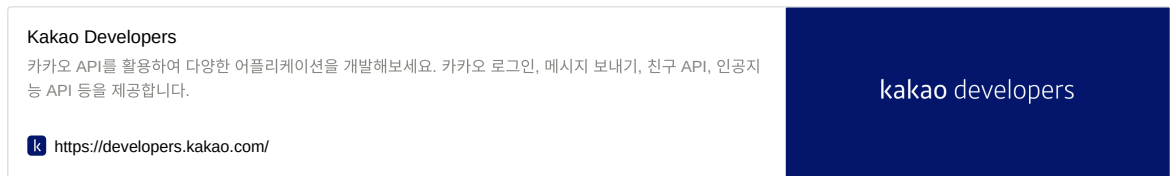
• 수위 자료 요청 포맷

변수명	필수여부	사용예	설명
ServiceKey	필수	OpenAPI 인증키	한강홍수통제소에서 발급한 OpenAPI 인증키
HydroType	필수	waterlevel	수문자료 종류 : 수위
DataType	필수	list	실시간 자료
TimeType	필수	10M	10분 단위 자료
		1H	1시간 단위 자료
		1D	1일 단위 자료
Wlobscd	선택	관측소 코드	수위 관측소 코드, 명시하지 않을 경우 전체 관측소의 최종 시간 자료 제공
Sdt	선택	yyyyMMddHHmm	검색 시작시간 및 검색 종료시간(10분 단위), 최대1개월
		yyyyMMddHH	검색 시작시간 및 검색 종료시간(1시간 단위), 최대1년
Edt		yyyyMMdd	검색 시작시간 및 검색 종료시간(1일 단위), 최대1년
DocumentType	필수	.xml	xml 문서 포맷
		.json	json 문서 포맷

API Key: CB761FAB-95F7-42CB-BA81-741A0D283E98

9.6 카카오맵 API

- Kakao Developers 홈페이지 이동



- 로그인 후 내 어플리케이션 추가 - 인증키 발급



- 사이트 도메인

Web

사이트 도메인	https://i9b101.p.ssafy.io https://i9b101.p.ssafy.io:3000 http://localhost:3000 http://localhost:8081 http://localhost:8080
---------	---

- 인증키

앱 키

네이티브 앱 키	bdbefd1021cda26e42c035a2f752a0e7
REST API 키	c55df60a1bdade8bc2cc020e2bc50e54
JavaScript 키	374cff0d8903c1ed2bf1e9533bb0feab
Admin 키	499503c83cf972170df4ba760f8ad102