포팅 매뉴얼

목차

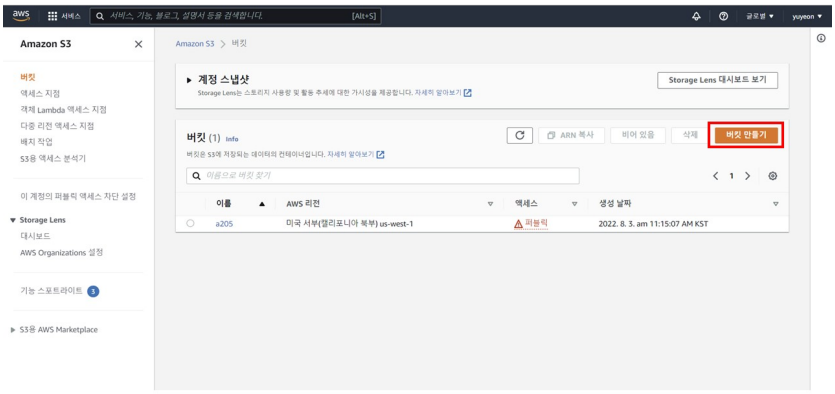
1. 기술 스택 및 사용 툴

- **Frontend**
  - React v5.0.1
  - node.js v16.14.0
  - npm v8.7.0
  - redux-toolkit v1.8.3
  - styled-components v5.3.5
- **Backend**
  - Spring Boot 2.7.1
  - Spring Data JPA 2.7.1
  - Spring Security 5.7.2
  - Spring Cloud 2.2.6
- **CI/CD & Database**
  - AWS ec2 - Ubuntu 20.04.4 LTS
  - Docker 20.10.12
  - Jenkins 2.346.3
  - nginx/1.18.0 (Ubuntu)
  - MySQL 8.0.30-0ubuntu0.20.04.2
  - certbot 0.40.0
- **Web RTC**
  - openVidu 2.22.0
- **Tools**
  - Intellij 2022.2
  - VS Code
  - Google Chrome 104.0.5112.81

2. 환경 설정

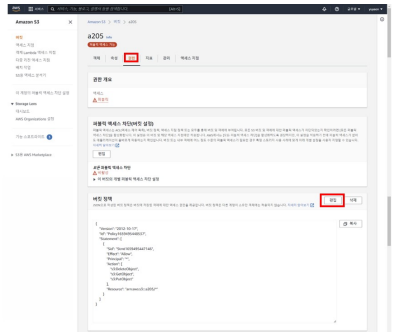
AWS S3 Bucket

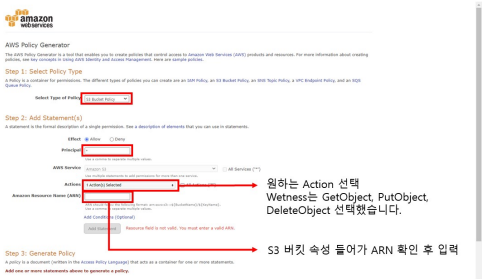
1. AWS 계정 생성
2. AWS S3 검색해서 S3 Management Console 접속 → 버킷 만들기



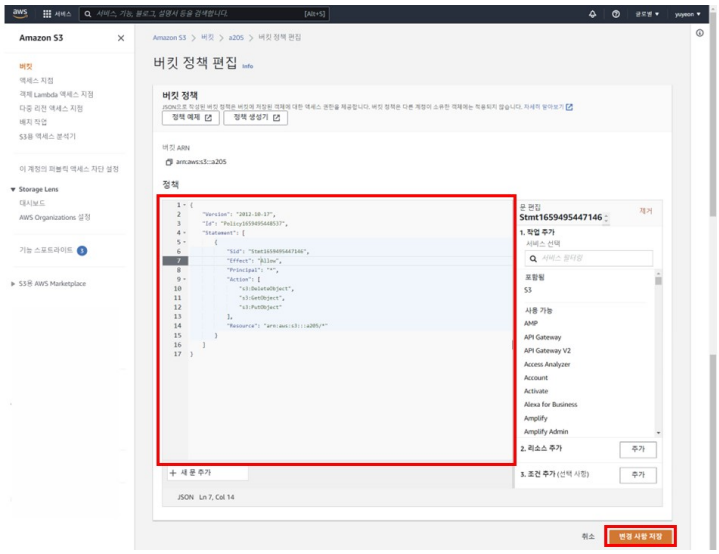
3. S3 버킷 설정 후 생성

4. 생성한 버킷 클릭 → 권한 설정으로 들어가 버킷 정책 생성

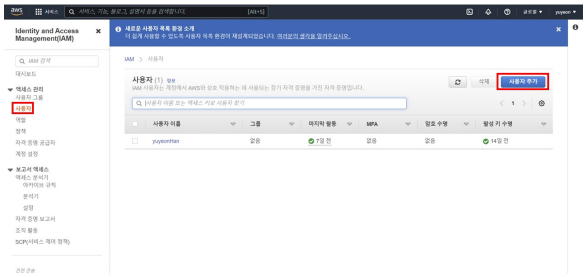


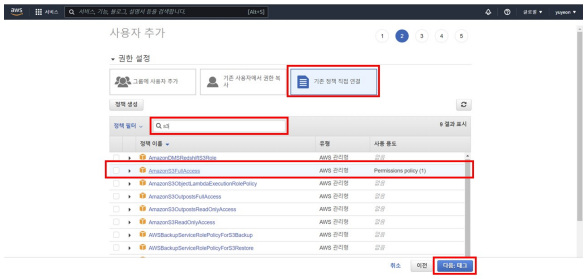


5. 생성된 JSON을 정책에 입력

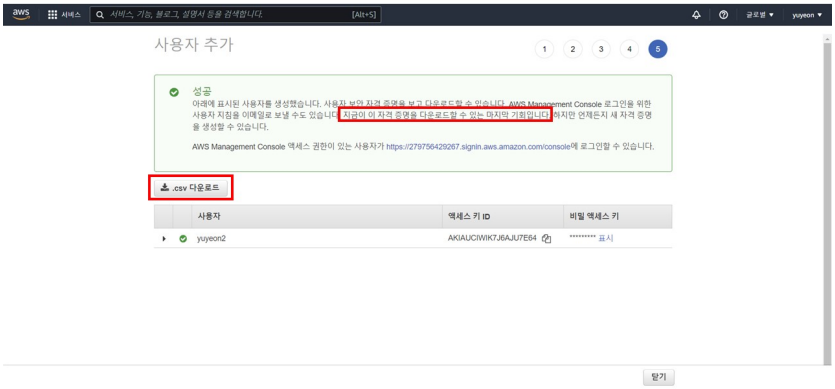


6. IAM 검색해서 IAM 콘솔로 들어가, 사용자 추가



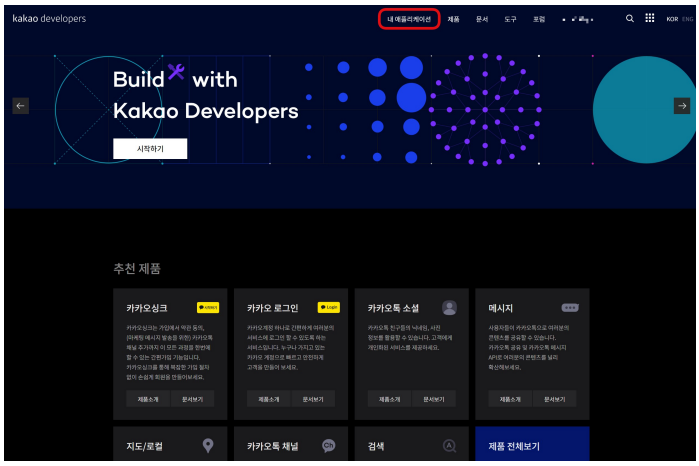


7. 생성한 직후, csv 다운로드 클릭해 AccessKey, SecretKey를 저장

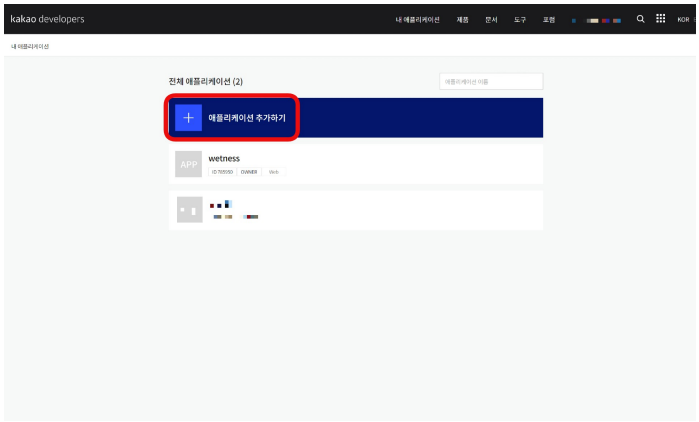


Oauth

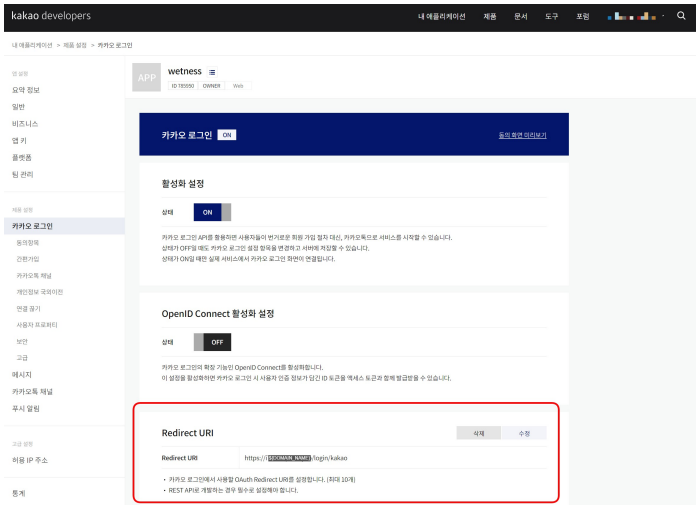
1. 카카오에서 내 애플리케이션 등록하기
    - https://developers.kakao.com/ 접속 후 로그인
    - “내 애플리케이션”으로 이동



- 애플리케이션 추가하기 및 이름 입력



- 해당 프로젝트에 REST API 설정



- 요약 정보의 REST API키와 REDIRECT URI를 하기된 서버 인스턴스 환경 변수에 저장

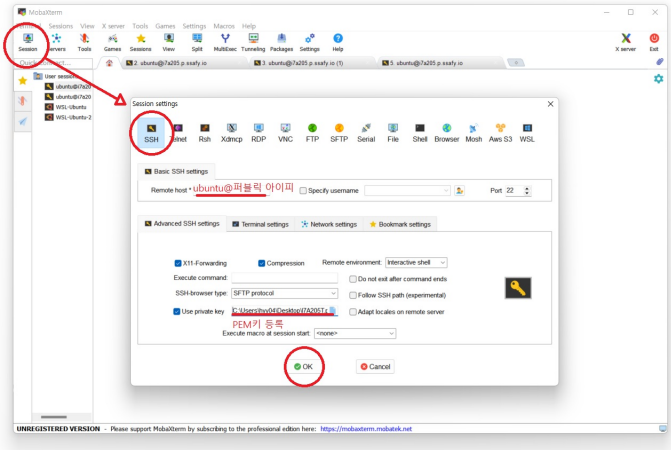
4. EC2 서버 세팅

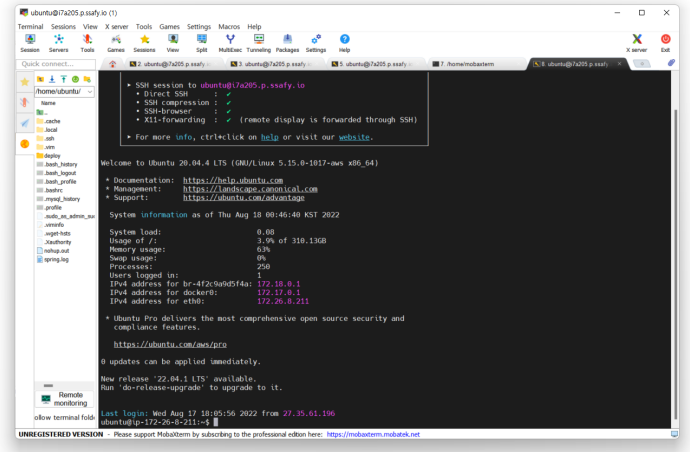
EC2 서버 접속

- Window Terminal 이용 시
  1. Windows PowerShell 사용
  2. 사용할 ,pem 키가 저장된 디렉토리로 이동
  3. SSH 클라이언트를 사용하여 .pem키에 저장된 서버인스턴스에 접근

```
PS C:\Users\SSAFY> cd        .pem키가 저장된 디렉토리
PS C:\[illegible]> ssh -i I7A205T.pem ubuntu@i7a205.p.ssafy.io
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Aug 18 04:17:26 KST 2022

  System load:                      0.3
  Usage of /:                       3.8% of 310.13GB
  Memory usage:                     64%
  Swap usage:                       0%
  Processes:                        226
  Users logged in:                  1
  IPv4 address for br-4f2c9a9d5f4a: 172.18.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            172.26.8.211

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

0 updates can be applied immediately.

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Aug 18 00:46:40 2022 from 106.255.139.202
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1017-aws x86_64)
```

- MobaXterm 이용 시





Java

- 패키지 업데이트 후 자바 설치

```
sudo apt-get update
sudo apt-get install openjdk-8-jdk
```

- 자바 환경변수 등록

```
nano /etc/profile
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
```

- 설정한 환경변수 적용

```
source /etc/profile
```

Mysql

- mysql 설치

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

- mysql 서버 실행

```
sudo systemctl start mysql
```

- 루트 계정으로 덤프 파일 실행

```
source /app/swim/workspace/wetness/backend/dump_wetness.sql
```

- wetness 계정 생성, CRUD 권한 부여

```
create user 'wetness'@'localhost' identified by '${password}'
grant select, insert, update, delete on wetness.* to 'wetness'@'localhost'
```

Docker

- 공식 문서에 따른 설치

```
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo 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/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

OpenVidu

1. 관리자 권한 획득(오픈비두 프로그램 설치 디렉토리 관련)

```
sudo su
```

2. 오픈비두가 권장하는 디렉토리 이동, 오픈비두 설치 및 설치된 경로로 이동

```
cd /opt
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | bash
cd openvidu
```

- 정상적인 설치 이후 폴더 구조

```
root@ip-172-26-8-211:/opt/openvidu# ls
cdr           coturn         custom-nginx-locations  docker-compose.override.yml  kms-crashes   openvidu  recordings
certificates  custom-layout  custom-nginx-vhosts     docker-compose.yml           kurento-logs  owncert
```

3. 사용자 서버 상황에 따라 openvidu config 파일 수정 필요

```
nano .env
```

- 서버 배포 상황에 따라 맞춤형 설정 필요

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP= 배포할 서버 도메인

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu
OPENVIDU_SECRET= 오픈비두 서버 접속 시 사용할 암호

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for production use.
#                Users will see an ERROR when connected to web page.
# - owncert:     Valid certificate purchased in a Internet services company.
#                Please put the certificates files inside folder ./owncert
#                with names certificate.key and certificate.cert
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                variable.
CERTIFICATE_TYPE= 오픈비두 서버에서 P2P 연결 시 사용할 인증타입 : letsencrypt 권장

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL= letsencrypt 선택 시 해당 서비스에서 인증 관련 알림을 보낼 이메일

# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the first boot
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
HTTP_PORT= 오픈비두의 NGINX가 돌아갈 http 포트번호

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
HTTPS_PORT= 오픈비두 실제 서버가 돌아갈 https 포트번호
```

- 본 프로젝트에서는 HTTP_PORT=81, HTTPS_PORT = 8443로 설정

4. 오픈비두 실행

```
./openvidu start
```

- 오픈비두 파일에 설치된 도커 이미지 파일이 컨테이너에 빌드 되며 실행됨

Nginx

```
sudo apt-get install nginx
```

Cerbot

- 서버에서 HTTPS 프로토콜을 사용하기 위한 SSL 인증서 발급 프로그램
- 본 프로젝트에서 사용된 우분투 20.04.4 버전으로 18.04 버전까지 패키지에서 제공하던 ppa:certbot/certbot가 지원하지 않음
  - 서버 /etc/apt/sources.list 파일에 접근해 수동으로 관련 레포지토리 추가
  - `deb http://cz.archive.ubuntu.com/ubuntu focal main universe`

```
deb http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal universe
deb http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal multiverse
# deb-src http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal multiverse
deb http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-updates multiverse
# deb-src http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-backports main
# deb-src http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu/ focal-backports main

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu focal partner
# deb-src http://archive.canonical.com/ubuntu focal partner

deb http://security.ubuntu.com/ubuntu focal-security main restricted
# deb-src http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
# deb-src http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
# deb-src http://security.ubuntu.com/ubuntu focal-security multiverse
deb http://kr.archive.ubuntu.com/ubuntu focal main universe
```

```
sudo apt-get update
sudo apt-get install python-certbot-nginx
certbot certonly --nginx -d ${DOMAIN_NAME}
```

- 아래와 같은 내용이 나오면 인증서 발급이 완료된 것 (유효기간 90일)

root@ip-172-26-8-211:/etc/apt# certbot certonly --nginx -d ${DOMAIN_NAME}
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for ${DOMAIN_NAME}
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/${DOMAIN_NAME}/fullchain.pem -해당 경로에 fullchain.pem 저장됨
  Your key file has been saved at:
  /etc/letsencrypt/live/${DOMAIN_NAME}/privkey.pem -해당 경로에 privkey.pem 저장됨
  Your cert will expire on 2022-11-13. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

- NGINX에 SSL 인증서 관련 설정

```
sudo nano /etc/nginx/sites-available/default
```

```
server {

        listen 443 default_server;
        listen [::]:443 default_server;

        # SSL configuration
        ssl on;
        ssl_certificate /etc/letsencrypt/live/${DOMAIN_NAME}/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/${DOMAIN_NAME}/privkey.pem;


        server_name _;

        # Frontend
        location / {
                root /app/swim/workspace/wetness/frontend/build;
                index index.html index.htm;
                try_files $uri $uri/ /index.html;

        }

        # Backend
        location /api {
                proxy_pass https://localhost:5000;
        }

}
```

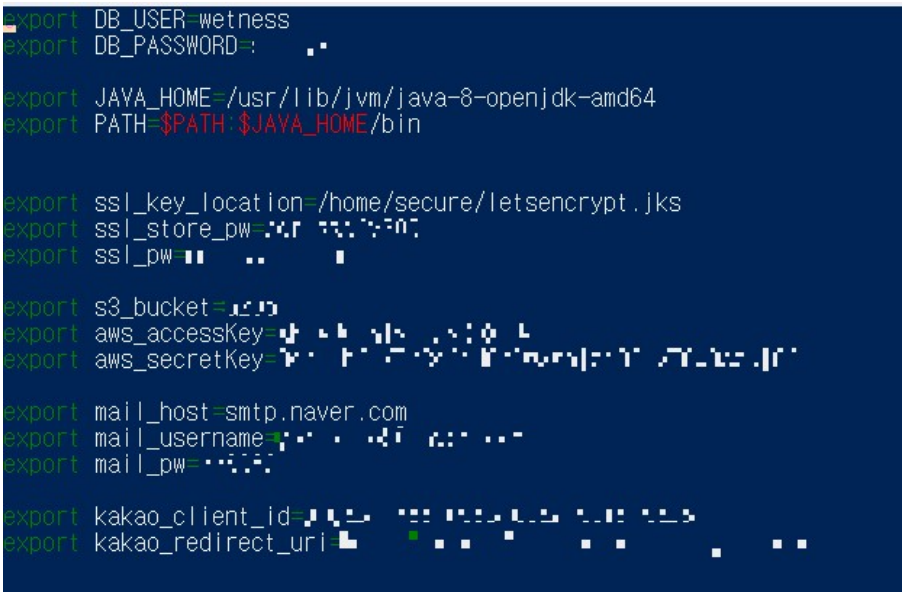
Environment Variable

- 환경 변수 리스트
    1. DB_USER
    2. DB_PASSWORD
    3. JAVA_HOME
    4. PATH
    5. ssl_key_location
    6. ssl_store_pw
    7. ssl_pw
    8. s3_bucket
    9. aws_accessKey
    10. aws_secretKey
    11. mail_host
    12. mail_username
    13. mail_pw

1. ec2 접속 후 command 창에 명령어 입력

```
cd ~/
sudo vim .bash_profile
```

2. bash_profile에 환경변수 입력



3. 편집 완료 후 command 창에 명령어 입력

```
source .bash_profile
```

5. 빌드 및 실행

Frontend 빌드

1. Frontend 폴더가 있는 곳으로 이동

```
cd frontend
```

2. frontend에 필요한 패키지 설치

```
npm install
```

3. 파일을 빌드

```
npm run build
```

Backend 빌드

1. Build.gradle이 있는 폴더로 이동

```
cd backend
```

2. gradle에 권한을 부여

```
chmod +x gradlew
```

3. 기존의 build를 clean하고, 다시 build

```
./gradlew clean build
```

실행 명령어

젠킨스 통해 프로젝트 빌드된 후, ec2 command에 명령어 입력

```
sudo service nginx restart
java -jar /home/ubuntu/deploy/backend-0.0.1-SNAPSHOT.jar

# 백그라운드로 실행하고 싶으면 jar 실행 위해 다음 입력
nohup java -jar /home/ubuntu/deploy/backend-0.0.1-SNAPSHOT.jar &
```