



D208: 나의 단짝친구

삼성청년SW아카데미 구미캠퍼스 7기
공통프로젝트 (6주: 2022/07/05 ~ 2022/08/19)

포팅매뉴얼

담당 컨설턴트: 유상진
허건녕(팀장), 김민정, 박신혜, 오도석, 최연지

<<목 차>>

1. 프로젝트 소개	3
2. 프로젝트 기술 스택	4
a. 이슈관리	
b. 형상관리	
c. 커뮤니케이션	
d. 개발 환경	
e. 상세내용	
3. 빌드 상세내용	5
a. Web 빌드	
b. IoT 빌드	
c. Chatting Server 빌드	
4. 배포 특이사항	7
a. Jenkins를 사용한 CI/CD	
5. DB 계정	10
a. MySQL	
6. 프로퍼티 정의	12
a. Nginx default	
b. EC2 설정	
7. 외부 서비스	15
a. Firebase Storage	
8. 출처 표기	16

1. 프로젝트 소개

"나의 단짝친구" 프로젝트는 SSAFY에서 시작한 프로젝트로, 장기 입원으로 지친 소아 환우들을 위한 IoT 기기 개발을 목표로 시작되었다. 소아병동에 입원한 아이들의 평균 나이는 만 3.6세로, 이 시기에 경험한 것들이 성장에 큰 영향을 미친다. 따라서 소아 병동에서 오래 입원한 아이들에게 언제든지 찾아가 대화할 수 있는 친구와 놀 거리를 제공함으로써 정서적 안정을 주고자 하였다.

프로젝트의 핵심은 세 가지로 나눌 수 있는데, 먼저 아이가 직접 캐릭터를 선택할 수 있는 맞춤형이 있다. 다음은 소통으로, 웹과 IoT 기기를 통해 부모님 혹은 보호자가 아이에게 이야기를 전달하거나 채팅을 통해 대화할 수 있다. 마지막 키워드는 즐거움이다. 아이가 캐릭터와 상호작용을 하고 그 외에도 제공되는 다양한 활동을 통해 지친 병원 생활에 활력소가 될 수 있기를 기대한다.

2. 프로젝트 기술 스택

- 1) 이슈관리: Jira
- 2) 형상관리: Gitlab
- 3) 커뮤니케이션: Mattermost, Notion, Webex
- 4) 개발 환경
 - a) OS: Windows 10
 - b) IDE
 - i) IntelliJ IDEA 2022.1.4
 - ii) Visual Studio Code 1.69.0
 - c) Database: MySQL Workbench 8.0 CE
 - d) Server: AWS EC2(MobaXterm 22.1)
 - i) Ubuntu 20.04.4 LTS
 - ii) Jenkins 2.346.2
 - e) Storage: Firebase Storage
- 5) 상세내용
 - a) BackEnd
 - i) Java 1.8.0
 - ii) Spring Boot Gradle 7.5
 - iii) Lombok 1.18.24, Swagger2
 - iv) jjwt 0.9.1
 - b) FrontEnd
 - i) React 18.0
 - ii) Node.js 16.16
 - iii) Three.js r143
 - c) IoT
 - i) Python 3.10.5
 - ii) OpenCV 4.6.0.66 (latest)
 - iii) PyQt5 5.15.7 (latest)
 - iv) Arduino 1.8.19
 - v) Naver Clova Voice & Speech Recognition

3. 빌드 상세내용

1) Web 빌드

- a) git merge
- b) jenkins auto build
- c) url(<https://i7d208.p.ssafy.io/>) 접속

2) IoT 빌드

a) Node.js

```
apt list | grep nodejs // 노드 버전확인
sudo curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash // 저장소를 최신버전으로 변경(16버전)
sudo apt-get install nodejs
```

b) React

```
npm install @react-three/fiber // react three js
npm install @react-three/drei // 마우스 추적
npm install "@react-three/cannon" // 물리효과

npm install 'recoil'
npm install 'styled-components'
npm install 'axios'

echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p // 리액트 오류 해결코드
```

c) express

```
npm install express
npm install cors
```

d) 얼굴인식

```
python -m pip install opencv-python
python -m pip install opencv-contrib-python
python -m pip install dlib
python -m pip install face_recognition
```

e) TTS

```
pip install playsound==1.2.2
```

f) 사진 촬영

```
npm install opencv-python  
npm install mediapipe  
npm install playsound  
npm install numpy  
npm install cvzone
```

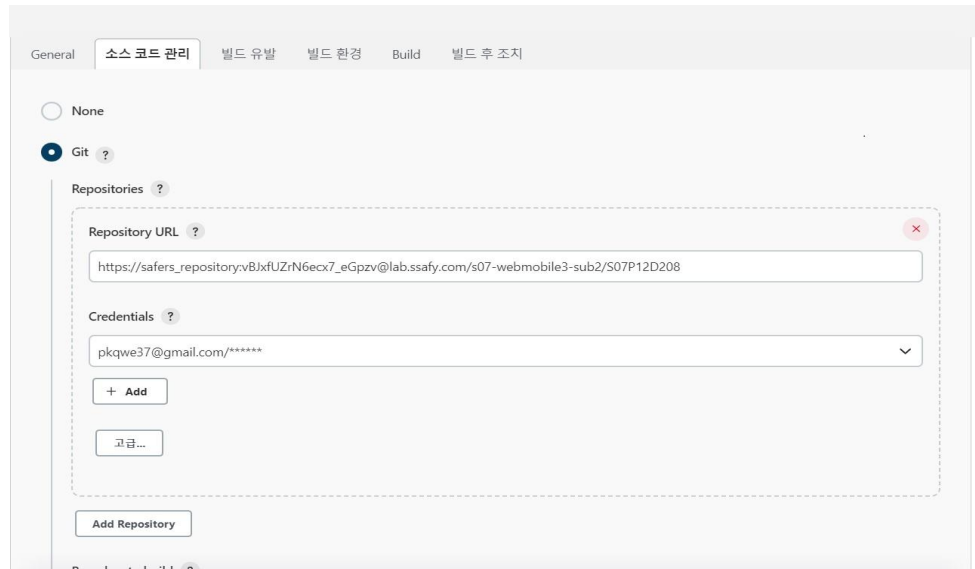
3) Chatting Server 빌드

- a) git merge
- b) jenkins auto build
- c) <https://i7d208.p.ssafy.io:4000> run
- d) request get <https://i7d208.p.ssafy.io/socket.io>
=> socket connect

4. 배포 특이사항

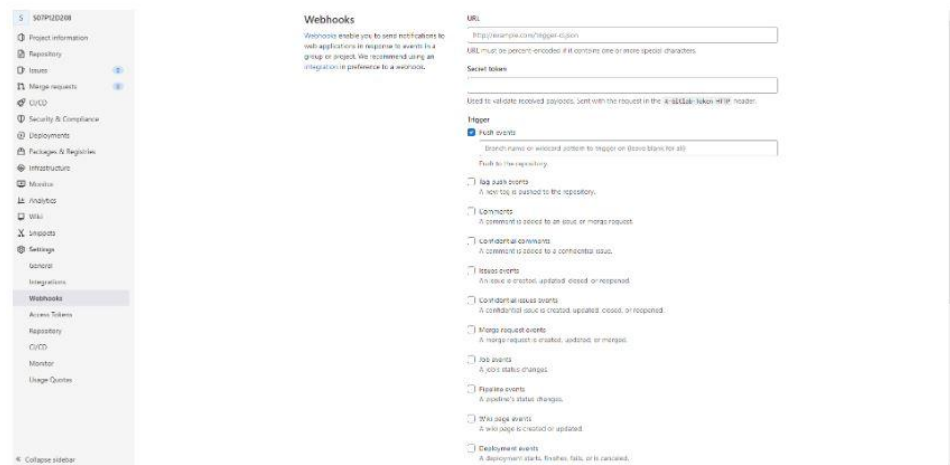
1) Jenkins를 사용한 CI/CD

a) Jenkins Git Connect



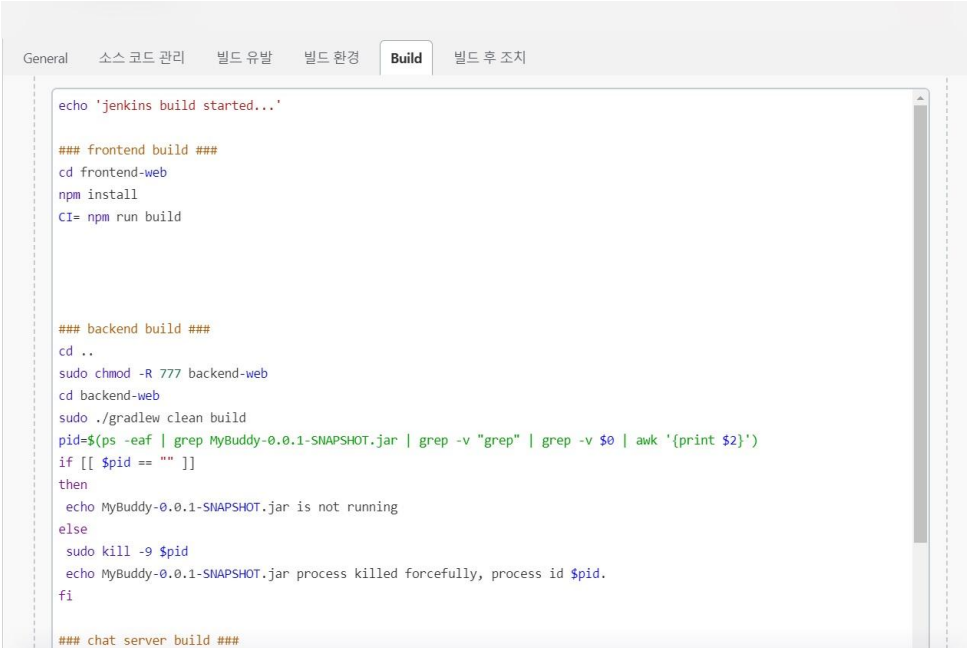
Jenkins와 Gitlab 연결

깃랩연결은 ssh연결과 accessToken연결 두가지가 있다 나는 accessToken으로 한다. jenkins와 gitlab을 연결해서 webhook을 걸기 위해서 git의 accesstoken과 jenkins의 시크릿토큰, 웹훅 url이 필요하다. jenkins프로젝트 생성, Build Triggers에 Build when a change is pushed to GitLab. GitLab webhook URL을 체크하면 gitlab 웹훅 url이 나온다. 아래에 고급 버튼을 누르고 generate 누르면 시크릿 토큰이 나오는데 이 두가지를 넣어준다.



그리고 발급받은 accessToken은 jenkins 소스 코드 관리의 Repository URL에 넣는다. https://safers_repository:accessToken@gitlab repository url 연결이 완료되면 Build의 Execute shell을 작성하자

b) Build execute shell command



The screenshot shows the Jenkins 'Build' tab with a shell script for building a project. The script is color-coded and includes comments for frontend, backend, and chat server builds. It uses 'cd' to change directories, 'npm install' for frontend, and 'gradlew clean build' for backend. It also includes logic to check if a process is running and kill it if necessary.

```
General 소스 코드 관리 빌드 유발 빌드 환경 Build 빌드 후 조치

echo 'jenkins build started...'

### frontend build ###
cd frontend-web
npm install
CI= npm run build

### backend build ###
cd ..
sudo chmod -R 777 backend-web
cd backend-web
sudo ./gradlew clean build
pid=$(ps -eaf | grep MyBuddy-0.0.1-SNAPSHOT.jar | grep -v "grep" | grep -v $0 | awk '{print $2}')
if [[ $pid == "" ]]
then
echo MyBuddy-0.0.1-SNAPSHOT.jar is not running
else
sudo kill -9 $pid
echo MyBuddy-0.0.1-SNAPSHOT.jar process killed forcefully, process id $pid.
fi

### chat server build ###
```

```
echo 'jenkins build started...'

### frontend build ###
cd frontend-web
npm install
CI= npm run build

### backend build ###
cd ..
sudo chmod -R 777 backend-web
cd backend-web
sudo ./gradlew clean build
pid=$(ps -eaf | grep MyBuddy-0.0.1-SNAPSHOT.jar | grep -v "grep" | grep -v $0 | awk '{print $2}')
if [[ $pid == "" ]]
then
echo MyBuddy-0.0.1-SNAPSHOT.jar is not running
else
sudo kill -9 $pid
echo MyBuddy-0.0.1-SNAPSHOT.jar process killed forcefully, process id $pid.
fi

### chat server build ###
cd ..
cd temp
cd ServerForChat
cd chatserver
ls
echo ls
pid=$(ps -eaf | grep app.js | grep -v "grep" | grep -v $0 | awk '{print $2}')
if [[ $pid == "" ]]
then
echo app.js is not running
else
sudo kill -9 $pid
echo app.js process killed forcefully, process id $pid.
fi
```


c) Post Build Task

General 소스 코드 관리 빌드 유발 빌드 환경 Build **빌드 후 조치**

Log text ×

BUILD SUCCESS

Operation

-- AND --

추가

Script ?

```
sudo nohup java -jar /var/lib/jenkins/workspace/ssafyD208/backend-web/build/libs/MyBuddy-0.0.1-SNAPSHOT.jar > jenkinsMybuddy.log 2>&1 &
```

```
sudo nohup node /var/lib/jenkins/workspace/ssafyD208/temp/ServerForChat/chatserver/app.js > nodeserver.log 2>&1 &
```

☐ Run script only if all previous steps were successful ?

☒ Escalate script execution status to job status ?

빌드 후 조치

Post build task

Log text : BUILD SUCCESS

Script :

```
sudo nohup java -jar /var/lib/jenkins/workspace/ssafyD208/backend-web/build/libs/MyBuddy-0.0.1-SNAPSHOT.jar > jenkinsMybuddy.log 2>&1 &
```

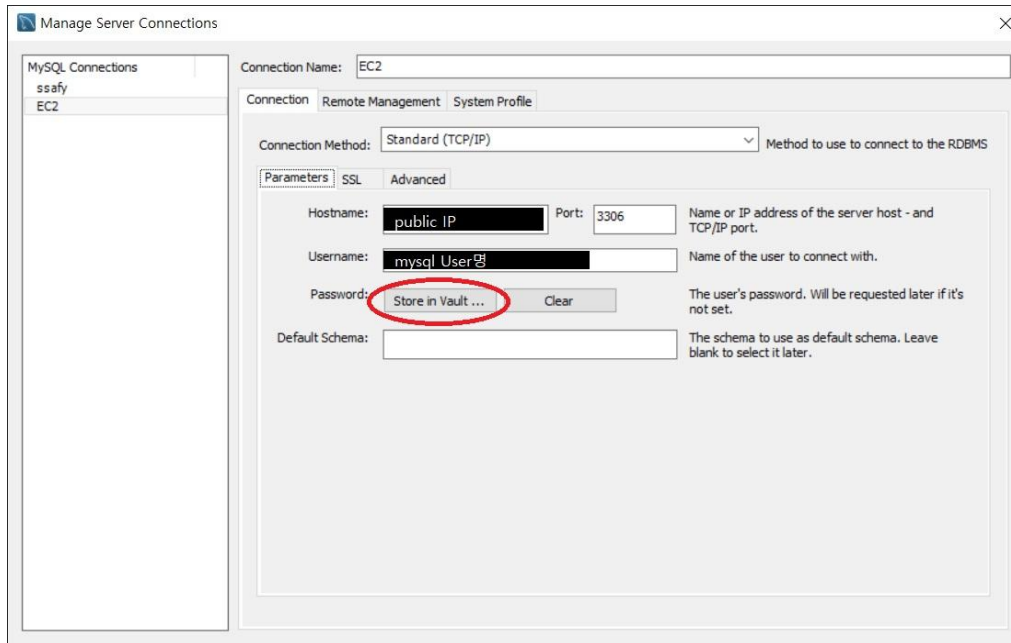
```
sudo nohup node /var/lib/jenkins/workspace/ssafyD208/temp/ServerForChat/chatserver/app.js > nodeserver.log 2>&1 &
```

Escalate script execution status to job status 체크

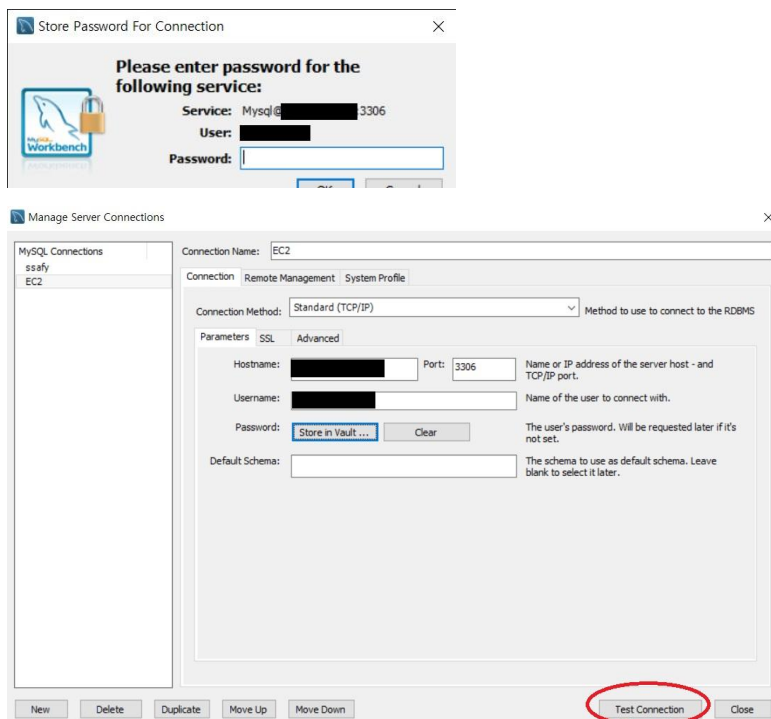
5. DB 계정

1) MySQL

Hostname에 public IP, Username에 Mysql User명 을넣고
Password check



Password 입력후 Test Connection -> Success 확인



MySQL Workbench



Successfully made the MySQL connection

Information related to this connection:

Host: [REDACTED]

Port: 3306

User: [REDACTED]

SSL: [REDACTED]

A successful MySQL connection was made with the parameters defined for this connection.

6. 프로퍼티 정의

1) Nginx default

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/lib/jenkins/workspace/ssafyD208/frontend-web/build;
    index index.html index.htm;
    server_name i7d208.p.ssafy.io;

    location / {
        # root /var/lib/jenkins/workspace/ssafyD208/frontend-web/build;
        # index index.html index.htm;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-Proto https;
        try_files $uri /index.html;
    }
    location /mybuddy{
        proxy_pass http://i7d208.p.ssafy.io:9999;
        proxy_redirect off;
        charset utf-8;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-NginX-Proxy true;
    }
    location /socket.io{
        proxy_pass http://i7d208.p.ssafy.io:4000;
        proxy_redirect off;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-NginX-Proxy true;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i7d208.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7d208.p.ssafy.io/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

2) EC2 설정

a) Mysql 설치

Mysql설치

```
sudo apt-get update
sudo apt-get install mysql-server (설치 중간중간에 password를 물어보는데 빈칸으로 두고 ok를 누르면 ubuntu 서버 비밀번호와 동기화 된다)
sudo ufw allow mysql (외부 접속을 허용해주자)
sudo service mysql start
sudo systemctl enable mysql ( ubuntu 서버가 재시작 되더라도 mysql이 자동 시작되도록 등록)
sudo mysql -u root -p (root로 접속, 비밀번호는 동일)
mysql>
use mysql
ALTER USER 'root'@'localhost' IDENTIFIED BY '바꿀비밀번호'; or ALTER user 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
flush privileges; (변경사항 저장)
create user '사용자 계정명'@'%' identified by '비밀번호'; (%는 사용자 계정을 허용하는 접속 ip이다)
SELECT User, Host, authentication_string FROM mysql.user; (계정이 만들어졌나 확인가능)
GRANT ALL PRIVILEGES ON \*. \% to '만든계정이름'@'%'; (모든권한을 모든db에 허용, 정말 조심해야함 !! 꼭 비밀번호는 어렵게 !!!)
exit
sudo su
cd /etc/mysql/mysql.conf.d
vi mysqld.cnf
bind-address를 찾아서 0.0.0.0으로 변경. (mysql의 외부접속 허용)
```

```
# Instead of skip-networking the default
# localhost which is more compatible on
bind-address = 0.0.0.0
```

```
service mysql restart
```

b) jenkins 설치 , port 변경

jenkins설치

1. apt-get update
2. sudo apt-get install openjdk-8-jdk (자신의 프로젝트에 맞는 버전으로 사용)
3. wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add - (repository키 추가)
4. sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list' (sources.list에 추가)
5. sudo apt-get update && apt-get install jenkins (jenkins를 apt-get 으로 설치)
6. sudo ufw allow 8080 (혹시 방화벽있으면 열기, 방화벽 확인 : ufw status)
7. sudo systemctl start jenkins (jenkins 시작)
8. sudo cat /var/lib/jenkins/secrets/initialAdminPassword (http://publicIP:8080으로 최초 접속시 필요한 Administrator password)

• 젠킨스 포트변경?

1. sudo vi /etc/default/jenkins

```
# UMASK=027
# port for HTTP connector (default 8080; disable with -1)
HTTP_PORT=9090
```

```
i (insert mode)
HTTP_PORT = 바꿀포트번호

ESC -> :wq! (저장후 나감)
sudo ufw allow 9090
```

2. sudo vi /etc/sysconfig/jenkins

c) https 설정을 위한 certbot

https 적용을 위한 Certbot 설치하기

참고 : <https://velog.io/@kimmjieun/nginx%E9%97%90-SSLHTTPS-%EC%A0%81%EC%9A%A9%ED%95%98%EA%B8%B0>

```
# sudo su #/var/www/html 안에서
# apt-get update
# apt-get install software-properties-common
# add-apt-repository universe
# add-apt-repository ppa:certbot/certbot
# apt-get update

# apt install certbot
# apt install python-certbot-nginx -> nginx와 사용할때

# certbot --nginx
```

디버깅을 위한 로그, 긴급한 사항 혹은 보안 사항 등을 전달할 이메일을 입력해달라는 내용, 이메일을 입력한 후 엔터 Enter email

약관을 읽고 SSL 을 사용하기 위한 서버 등록을 승낙해야 한다는 이야기. Y 입력한 후 엔터 Y

위에서 입력했던 이메일을 재단에도 공유해서 재단의 뉴스, 캠페인, 디지털 자유운동 지원 방법 등을 받을 것인지에 대한 내용. Y

어떤 도메인에 HTTPS를 적용시킬지 물어본다. 그냥 엔터를 누르면 목록에 쓴 모든 도메인에 HTTPS를 적용시켜주고, `1 3` 이런식으로 숫자를 찍어서

sudo certbot --nginx -d 도메인 (인증서 설치)

이메일 등록과, 약관에 동의하시면 기존 http 연결을 어떻게 설정할 것인가에 대한 질문이 나타나게 됩니다. 아래 목록을 확인하시고 원하시는 설정

* 1을 입력한다면 http 연결을 https로 리다이렉트 하지 않습니다.

* 2를 입력한다면 https 연결을 https 로 리다이렉트 시킵니다.

2

vi /etc/crontab (자동갱신)

0 0 * * * root certbot -q renew --nginx >> /var/log/letsencrypt.renewal.log 2>&1 #추가

service cron restart

수동 리다이렉트

참고 : <https://velog.io/@newon-seoul/Certbot%E9%9D%84-%ED%99%9C%E9%A9%ED%95%9C-HTTPS-%EC%A0%81%EC%9A%A9%EA%B3%BC-%EB%A6%AC%EB%B8%A>

7. 외부 서비스

1) Firebase Storage

a) Bucket 주소

 <gs://my-buddy-359c8.appspot.com>

b) Storage 정보



c) Configuration

```
# firebase
config = {
  "apiKey": "AIzaSyBkdppI7NQyWJr5uzIkwaSDvsBVQWrBfXc",
  "authDomain": "my-buddy-359c8.firebaseio.com",
  "projectId": "my-buddy-359c8",
  "storageBucket": "my-buddy-359c8.appspot.com",
  "messagingSenderId": "377294439460",
  "appId": "1:377294439460:web:df58512675aa6f0d0e04b1",
  "measurementId": "G-C4P4WZEC0B1",
  "databaseURL": "",
  "serviceAccount": "c:/Users/multicampus/Desktop/MyGit/PJT/New/S07P12D208/temp/SERVER/python/face_recognition/serviceAccountKey.json"
}
```

8. 출처 표기

- flaticon

- chat.png

- https://www.flaticon.com/free-icon/chat_8256607?related_id=8256607&origin=search

- send.png

- https://www.flaticon.com/free-icon/send_786205?term=send&page=1&position=2&page=1&position=2&related_id=786205&origin=search