1. **환경 변수**
2. **nodeJS 환경변수**

PORT=백엔드 포트번호

MONGO\_URL=MongoDB URL

REDIS\_URL=Redis URL

EMAIL\_USER=Email 주소

EMAIL\_PASS=Email 비밀번호

* **Nodejs 버전 : 14.21.3**

1. **빌드하기**
2. **Back-end**

cd /var/jenkins\_home/workspace/3bti/back-end

cp /var/jenkins\_home/workspace/.env ./

npm install

docker stop back-end

docker rm back-end

docker rmi backend

docker build -t backend

docker run -d -p 5000:5000 -v /home/ubuntu/user:/server/user -e TZ=Asia/Seoul --name back-end --network my-network backend

1. **Front-end**

cd /var/jenkins\_home/workspace/3bti/front-end

npm install

CI=false npm run build

docker stop front-end

docker rm front-end

docker rmi frontend

docker build -t frontend .

docker run -d -p 3000:80 -v /home/ubuntu/user:/app/build/user -e TZ=Asia/Seoul --name front-end --network my-network frontend

1. **Hard-ware**

Lxpanelctl restart

Sudo python /sketchbook/wifi.py

와이파이 SSID 및 Password 입력

이후 자동실행

1. **Nginx 설정**

server {

listen 8888 ssl;

server\_name i9c102.p.ssafy.io:8080;

ssl\_certificate /etc/letsencrypt/live/i9c102.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i9c102.p.ssafy.io/privkey.pem;

location / {

proxy\_pass http://localhost:8080;

}

}

server {

root /app/bulid;

location / {

proxy\_pass http://localhost:3000;

}

location /api/ {

proxy\_pass http://localhost:5000;

}

listen [::]:443 ssl ipv6only=on;

listen 443 ssl;

ssl\_certificate /etc/letsencrypt/live/i9c102.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i9c102.p.ssafy.io/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

}

server {

if ($host = i9c102.p.ssafy.io) {

return 301 https://$host$request\_uri;

}

listen 80;

server\_name i9c102.p.ssafy.io;

root /app/build;

return 404;

}

1. **DB 설정**
2. **회원 정보**

**{**

email: { type: String, required: true, trim: true, unique: 1, },

password: { type: String, maxlength: 20 minlength: 8 trim: true, required: true, },

role: { type: String, default: "user", },

auth: { type: String, default: "", },

token: { type: String, },

tokenExp: { type: Number, },

isConnected: { type: Boolean, default: "false", },

}

1. **Redis**

* **0번 DB**

Key : OTP / value : email : OTP기기에서 연결되면 value: true로 수정

* **1번 DB**

Key : email / value : 명령어 : 현재 상태전달하고 그에 따라 프로그램 시작

* **2번 DB**

Key : email / value : 사람의 관절 좌표

* **3번 DB**

Key : email / value : 캐릭터의 관절 좌표

* **4번 DB**

Key : email / value : 캐릭터의 이동 값