1_E201_포팅메뉴얼

1. 개발 환경

프론트엔드

- React

백엔드

- SpringBoot
- SpringSecurity
- JWT
- MyBatis

데이터베이스

- MySQL
- Redis cache

인프라

- AWS EC2
- Docker
- Nginx
- Jenkins

기타 툴

- GitLab
- Jira
- Notion
- Figma
- ErdCloud
- CloudCraft

2. EC2 세팅

2.1 Docker 설치

```
sudo apt-get update
sudo apt-get install -y docker docker-io

# test
docker ps
```

2.2 Nginx 설치

```
docker pull nginx
docker run --name nginx -d -p 80:80  -p 443:443 nginx
```

2.3 SSL 적용

```
# 컨테이너 내부로 진입
docker exec -it nginx /bin/bash

# Let's Encrypt 설치
apt-get update
apt-get install vim
apt-get install certbot
apt-get install python3-certbot-nginx

# 도메인 설정
cd etc/nginx/conf.d
vim domain-name.conf

server {
    server_name <도메인1> <도메인2> ...
}

# 컨테이너 제시작
docker restart nginx

# domain-name.conf 파일 내부

server {
    server_name j12e201.p.ssafy.io;



    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/j12e201.p.ssafy.io/fullchain.pem; # mana
    ssl_certificate_key /etc/letsencrypt/live/j12e201.p.ssafy.io/privkey.pem; # m
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    location / {
        proxy_pass http://3.36.67.192:5080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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;
    }


    location /api/ {
        proxy_pass http://3.36.67.192:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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_buffering off;
        proxy_cache off;
        proxy_read_timeout 3600s;
        chunked_transfer_encoding off;
    }

    location /multi/ {
        proxy_pass http://3.36.67.192:19092/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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 $scheme;
        proxy_read_timeout 3600;
        proxy_send_timeout 3600;
        tcp_nodelay on;
    }
}
server {
    if ($host = j12e201.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        server_name j12e201.p.ssafy.io;
    listen 80;
    return 404; # managed by Certbot
}
```

2.4 EC2 Port

Port 번호	내용
80	HTTPS로 redirect용
443	HTTPS
8000	apigateway
8761	eureka
8086	meeting-service
8089	socket-service
8084	fairytale-service
5080	frontend
18080	user-service
8083	friend-service
3478	openvidu-coturn
6380	redis
19090	jenkins
3306	mysql

3. CI/CD 구축

3.1 Jenkins 도커 이미지 + 컨테이너 생성

```
docker run -d \
  --name jenkins \
  -p 19090:19090 \
  -p 50000:50000 \
  -v jenkins_home:/var/jenkins_home \
  -v /var/run/docker.sock:/var/run/docker.sock \
  jenkins/jenkins:lts

# host docker를 사용위해 docker 설치
sudo docker exec -it jenkins bash
apt-get update
apt-get install docker docker.io
```

3.2 jekins 설정

- 아래 링크 참고

https://docs.gitlab.com/ee/integration/jenkins.html - jenkins gitlab 연동

[CI/CD] Gitlab과 Jenkins로 CI/CD 구축하기 - Jenkins 설정법

4. Openvidu 설정

- 아래 링크 참고

https://docs.openvidu.io/en/stable/deployment/ce/on-premises/ - deployment → openvidu(CE) →On premises