

배포

host IP: ubuntu@i4c108.p.ssafy.io

| 배포는 **컴파일** → **빌드** → **배포** 의 과정을 거쳐야한다.

baseURL: 'http://i4c108.p.ssafy.io/api/'

사전 테스트

1. Front

1. npm install
2. npm run serve → 서버 확인

2. Back

1. sudo apt-get install openjdk-8-jdk
2. gradle이 있는 디렉토리에서 `./gradlew test`
3. 권한이 없으면 `sudo chmod 777 ./gradlew`
4. `./gradlew build`

이렇게 빌드가 성공하면 build 디렉토리가 생성되고, build/libs 디렉토리 아래에 jar 파일이 생성됩니다.

5. 서버 실행

SpringBoot는 톰캣 서버가 내장되어있으므로, 빌드된 파일(jar)만 실행해주면 서버가 가동됩니다.

```
[ec2-user@ip-172-31-40-7 libs]$ java -jar demo-0.0.1-SNAPSHOT.jar
```

Nginx

클라우드에서 실행을 확인 한 후 Nginx를 통해서 항상 서버가 운영될 수 있게 해야합니다.

1. install

```
sudo apt install nginx
```

2. 프론트

```
npm run build
```

1) Nginx 수정

```
sudo vim /etc/nginx/sites-enabled/default
#웹서버 루트 기존 설정은 주석처리, front를 build한 위치로 변경
-----
#root /var/www/html;
root /home/ubuntu/폴더/frontend/dist;
...
location / {
    try_files $uri $uri/ /index.html;
}
-----
# 설정 변경 후 syntax 검사 필수
sudo nginx -t
# 설정 변경 후 Nginx 재시작
sudo service nginx restart
```

```
sudo vim /etc/nginx/sites-enabled/default
-----
server {
    listen 80;
    ...
    ...
    location / {

    }
    ### backend reverse proxy 설정 추가 ###
    location /api {
        proxy_pass http://localhost:8080;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        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_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Port $server_port;
}
### 여기까지 ###
...
}

-----

# 설정 변경 후 syntax 검사 필수
sudo nginx -t
# 설정 변경 후 Nginx 재시작
sudo service nginx restart

```

```

gradle이면 -> build/libs/ xxxx.jar 실행
java -jar xxxx.jar

```

백엔드 HA구성

Reverse proxy 설정 및 upstream 추가, proxy_pass 수정

```

sudo vim /etc/nginx/sites-enabled/default
-----
# backend upstream 설정
upstream backend {
    server localhost:8080;
    server localhost:8081;
}
server {
    listen 80;
    ...
    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
    location /api {
        # 이 부분 upstream으로 변경
        proxy_pass http://backend;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        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_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Port $server_port;
}
}

```

```

# 설정 변경 후 syntax 검사 필수
sudo nginx -t
# 설정 변경 후 Nginx 재시작
sudo service nginx restart

```

터미널에서 backend 프로그램 실행 후 빠져나오면 프로세스가 종료됨 ⇒ nohup, &

```

# 첫번째 서버 실행
nohup java -jar demo-0.0.1-SNAPSHOT.jar \
--server.servlet.context-path=/api \    //back 서버 요청 기본 주소
--server.address=127.0.0.1 \
--server.port=8080 &
# 두번째 서버 실행
nohup java -jar demo-0.0.1-SNAPSHOT.jar \
--server.servlet.context-path=/api \
--server.address=127.0.0.1 \
--server.port=8081 &
# process 확인
ps -ef | grep java
Linux ps (프로세스 확인하기)

```

수정 후 재 빌드 방법

```

kill -9 `pgrep java`
ps와 grep으로 pid를 조회하여 프로세스 종료

# 기존 git repo 삭제
rm -rf ~/.project`
# git clone
git clone
frontend와 backend의 build 폴더를 삭제하고, git pull 수행 후 다시 build 해도 상관없다.
# front
cd frontend
npm install
npm run build
#backend
cd backend -> demo-0.0.1-SNAPSHOT.jar 위치( build/libs/)

```

```
nohup java -jar demo-0.0.1-SNAPSHOT.jar \  
--server.servlet.context-path=/api \  
--server.address=127.0.0.1 \  
--server.port=8080 &  
# 두번째 서버 실행  
nohup java -jar demo-0.0.1-SNAPSHOT.jar \  
--server.servlet.context-path=/api \  
--server.address=127.0.0.1 \  
--server.port=8081 &  
ps -ef | grep java
```
