

**포팅 매뉴얼**

**A704**

강신욱 / 김민경 / 박진우 / 최재희 / 한원석 / 함형준

**목차**

1. 기술 스택 및 버전

2. Back-end 설정 및 빌드

3. Front-end 설정 및 빌드

**기술 스택 및 버전**



* **Version**

Spring boot : v2.7.7,

Java : openjdk v11.0.18,

Apache Tomcat : v9.0.70,

React.js : v18.2.0,

MySQL : v8.0CE

Node.js : v18.12.1(LTS),

Yarn : v1.22.19

npm : 8.9.13,

**Back-end 설정 및 배포**

1. **Spring boot 빌드 및 배포**

* cd {프로젝트root} // 프로젝트 root 디렉토리로 이동
* ./gradlew clean build // gradle 빌드
* docker build –t pnut\_back . // docker 빌드
* docker container run -d --name pnut\_back -p 9090:9090 pnut\_back // docker컨테이너 실행

1. **Redis 배포**

* docker pull redis // redis 설치
* docker container run -d --name redis -p 6379:6379 redis // docker 컨테이너 실행

1. **Django 빌드 및 배포**

* Python manage.py migrate
* Docker build –t pnut\_back\_django
* docker container run -d --name pnut\_back\_django -p 8000:8000 pnut\_back\_django

**Front-end 설정 및 배포**

1. **React 배포**

* yarn i // module 설치
* yarn build // yarn 빌드
* dist 폴더의 index.html로 접속

1. **Tailwindcss**

* 수정시
* **npx tailwindcss -i ./src/index.css -o ./src/output.css --watch**

**Nginx 설정**

1. **도메인**

* pnut,site

1. **ssl 발급**

* certbot

1. **default.conf** }
2. server {
3. #server\_name pnut.site;
4. location / {
5. root /home/ubuntu/jenkins\_docker/jenkins\_home/workspace/pnut-front/front/p-nut/dist;
6. index index.html index.htm;
7. try\_files $uri $uri/ /index.html
8. proxy\_set\_header X-Real-IP $remote\_addr;
9. proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;
10. proxy\_http\_version 1.1;
11. proxy\_set\_header Upgrade $http\_upgrade;
12. proxy\_set\_header Connection "upgrade";
13. proxy\_set\_header Host $http\_host;
14. }
15. location /api {
16. proxy\_pass http://localhost:9090$request\_uri;
17. }
18. location /foods {
19. proxy\_hide\_header Access-Control-Allow-Origin;
20. add\_header 'Access-Control-Allow-Origin' '\*';
21. add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
22. add\_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
23. add\_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
24. proxy\_pass http://localhost:8000;
25. }
26. listen 443 ssl; # managed by Certbot
27. ssl\_certificate /etc/letsencrypt/live/pnut.site/fullchain.pem; # managed by Certbot
28. ssl\_certificate\_key /etc/letsencrypt/live/pnut.site/privkey.pem; # managed by Certbot
29. include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
30. ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
31. }
32. server {
33. if ($host = pnut.site) {
34. return 301 https://$host$request\_uri;
35. } # managed by Certbot
36. listen 80;
37. server\_name pnut.site;
38. location / {
39. proxy\_http\_version 1.1;
40. proxy\_set\_header Upgrade $http\_upgrade;
41. proxy\_set\_header Connection "upgrade";
42. proxy\_pass http://localhost:9090;
43. proxy\_set\_header X-Real-IP $remote\_addr;
44. proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;
45. proxy\_set\_header Host $http\_host;
46. }
47. }