빌드 및 배포 과정 문서

SSAFY 11기 C108 킥보드브레이커

▷ 개발환경

|  |  |  |
| --- | --- | --- |
|  | frontend | Backend |
| Language | TypeScript | JAVA 21 |
| Framework | React, Zustand | SpringBoot, JPA |
| Library | Tailwind | JWT, JPA |
| DB |  | PostgreSQL, Redis |

▷ 버전정보

|  |  |
| --- | --- |
| VS Code | 1.95.3 |
| IntellJ | 2024.1.4 |
| JVM | OpenJDK 21 |
| Node.js | 20 |
| SERVER – AWS EC2 | 20.04.6 LTS |
| PostgreSQL | 17 |
| Redis | 7.4.1 |

▷ 빌드 환경변수 정보

* Frontend

|  |
| --- |
| VITE\_KAKAOMAP\_API\_KEY=59dfafcaacec223b4ef29b0a497643f8  VITE\_URL=https://k11c108.p.ssafy.io/api/v1  VITE\_API\_BASE\_URL=https://k11c108.p.ssafy.io  VITE\_API\_URL=https://k11c108.p.ssafy.io/api/v1 |

* Backend

|  |
| --- |
| spring:  datasource:  url: jdbc:postgresql://k11c108.p.ssafy.io:5432/kb  username: postgres  password: 8xlazlrqhem  cloud:  aws:  credentials:  access-key: AKIATJHQDZGV5UBLFDFF  secret-key: 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  jwt:  secret-key: 8xlazlrqhem8xlazlrqhem8xlazlrqhem8xlazlrqhem8xlazlrqhem8xlazlrqhem  data:  redis:  host: redis  port: 6379  password:  server:  port: 8081 |

▷ Frontend Dockerfile

|  |
| --- |
| # Use Node.js base image  FROM node:20 AS build  # Set working directory inside container  WORKDIR /app  # Copy package files and install dependencies  COPY package\*.json ./  COPY .env .env  RUN export $(cat .env | xargs)  RUN npm install  # Copy all frontend files  COPY . .  # Build React app for production  RUN npm run build  FROM nginx  COPY ./letsencrypt /etc/letsencrypt/  COPY ./nginx/nginx.conf /etc/nginx/nginx.conf  #Copy  COPY --from=build /app/dist /usr/share/nginx/html  #Expose port 3000  EXPOSE 3000  # ngnix  CMD ["nginx", "-g", "daemon off;"] |

▷ Backend Dockerfile

|  |
| --- |
| # Start with an OpenJDK base image  FROM openjdk:21-jdk-slim  # Set working directory  WORKDIR /app  # Copy the built Spring Boot application JAR file to the container  COPY kickboard/build/libs/kickboard-0.0.1-SNAPSHOT.jar app.jar  # Expose the port that the app will run on (e.g., 8080)  EXPOSE 8080  # Run the JAR file  ENTRYPOINT ["java", "-jar", "app.jar"] |

▷ Nginx

|  |
| --- |
| user nginx;  worker\_processes auto;  error\_log /var/log/nginx/error.log notice;  pid /var/run/nginx.pid;  events {  worker\_connections 1024;  }  http {  include /etc/nginx/mime.types;  default\_type application/octet-stream;  log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log main;  sendfile on;  #tcp\_nopush on;  keepalive\_timeout 65;  #gzip on;  include /etc/nginx/conf.d/\*.conf;  upstream backend{  server my\_network:8081;  }  server {  listen 80;  server\_name k11c108.p.ssafy.io;  return 301 https://$host$request\_uri;  # return 404; # managed by Certbot  }  server {  listen 443 ssl;  server\_name k11c108.p.ssafy.io;  location / {  root /usr/share/nginx/html;  index index.html;  try\_files $uri $uri/ /index.html;  # 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 $scheme;  }  location /api/ {  add\_header Access-Control-Allow-Origin \*;  add\_header Access-Control-Allow-Methods 'GET, POST, PUT, DELETE, OPTIONS';  add\_header Access-Control-Allow-Headers 'Origin, Content-Type, X-Requested-With, Authorization';  add\_header Access-Control-Allow-Credentials 'true';  if ($request\_method = 'OPTIONS'){  add\_header Access-Control-Max-Age 1728000;  add\_header Content-Type 'text/plain charset=UTF-8';  add\_header Content-Length 0;  return 204;  }  # proxy\_pass http://localhost:8081;  proxy\_pass http://backend;  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;  }  ssl\_certificate /etc/letsencrypt/live/k11c108.p.ssafy.io/fullchain.pem; # managed by Certbot  ssl\_certificate\_key /etc/letsencrypt/live/k11c108.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  }  } |

▷ 특이사항

* kakaoMap이 뜨지 않음