

프론트 배포 정리 (젠킨스,도커)

 Created By	
 Stakeholders	
 Status	
 Type	
 Created	@2022년 8월 17일 오후 10:02
 Last Edited Time	@2022년 8월 17일 오후 10:08
 Last Edited By	

배포는 백엔드 배포정리와 동일.

DockerFile

```
FROM node:14.19.2 as builder

WORKDIR /app

COPY package*.json ./

RUN npm i

COPY ./ ./

ARG KAKAO_ID
ARG NAVER_ID
ARG GOOGLE_ID
ARG AWS_SECRET_KEY
ARG AWS_ACCESS_KEY
ARG KAKAO_MAP

RUN REACT_APP_KAKAO_MAP=${KAKAO_MAP} REACT_APP_AWS_ACCESS_KEY=${AWS_ACCESS_KEY} REACT_APP_AWS_SECRET_KEY=${AWS_SECRET_KEY} REACT_APP_GOOGLE_CLIENT_ID=${GOOGLE_ID} REACT_APP_NAVER_CLIENT_ID=${NAVER_ID} REACT_APP_KAKAO_CLIENT_ID=${KAKAO_ID} npm run build

FROM nginx

EXPOSE 3000

COPY ./nginx/default.conf /etc/nginx/conf.d/default.conf

COPY --from=builder /app/build /usr/share/nginx/html

CMD ["nginx","-g","daemon off;"]
```

DockFile.e2e

```
FROM node:14.15.4-buster
FROM cypress/base

RUN apt-get update && \
  apt-get install --no-install-recommends -y \
  libgtk2.0-0 \
  libgtk-3-0 \
  libnotify-dev \
  libgconf-2-4 \
  libgdm-dev \
  libnss3 \
  libxss1 \
  libasound2 \
  libxtst6 \
  xauth \
  xvfb

WORKDIR /app

COPY ./package*.json ./

RUN npm i

ARG CYPRESS_RECORD_KEY

ENV CYPRESS_RECORD_KEY ${CYPRESS_RECORD_KEY}

COPY ./ ./
```

JenkinsFile

```
pipeline {
  agent any
  stages {
    // stage("unit test"){
    //   steps{
    //     script{
    //       try {
    //         sh 'docker rmi front-unit'
    //       } catch (e) {
    //         sh 'echo "fail remove test-images"'
    //       }
    //     }
    //     sh 'docker build -t front-unit -f ./frontend/Dockerfile.unit ./frontend'
    //     // sh 'docker run front-test npm run test'
    //   }
    //   post {
    //     success{
    //       sh 'echo unit 테스트 성공!!'
    //     }
    //     failure{
    //       sh 'echo unit 테스트 실패'
    //     }
    //   }
    // }
    // stage("e2e test"){
    //   steps{
    //     script{
    //       try {
    //         sh 'docker stop e2e'
    //         sh 'docker rm e2e'
    //         sh 'docker rmi front-e2e'
    //       } catch (e) {
    //         sh 'echo "fail remove e2e-test-images"'
    //       }
    //       try{
    //         sh 'docker build -t front-e2e -f ./frontend/dockerfile.e2e ./frontend --build-arg CYPRESS_RECORD_KEY=${CYPRESS_RECORD_KEY}'
    //         sh 'docker run -d --name e2e front-e2e /app/node_modules/.bin/cypress run --record --key ${CYPRESS_RECORD_KEY}'
    //       }catch(e){
    //         sh 'echo "테스트 실패"'
    //       }
    //     }
    //   }
    //   post {
    //     success{
    //       sh 'echo e2e 테스트 실패 성공'
    //     }
    //     failure{
    //       sh 'echo e2e 테스트 실패'
    //     }
    //   }
    // }
    stage("build"){
      steps {
        script{
          try {
            sh 'docker stop frontend'
            sh 'docker rm frontend'
            sh 'docker rmi wlsgh97/nahonsan-frontend'
```

```
} catch (e) {  
    sh 'echo "fail to stop and remove container"'  
}  
  
}  
  
sh 'docker build -t wlsgh97/nahonsan-frontend ./frontend --build-arg KAKAO_MAP=${KAKAO_MAP} --build-arg AWS_ACCESS_KEY=${AWS_ACCESS_KEY} --build-arg AWS_SECRET_KEY=${AWS_SECRET_KEY} --build-arg KAKAO_ID=${KAKAO_ID} --build-arg NAVER_ID=${NAVER_ID}'  
sh 'echo "$DOCKER_HUB_PASSWORD' | docker login -u "$DOCKER_HUB_ID" --password-stdin'  
echo 'build image...'  
  
sh "docker push wlsgh97/nahonsan-frontend"  
sh "docker run -d -p 3000:3000 --name frontend wlsgh97/nahonsan-frontend"  
  
}  
}  
}
```