Infra

기본

sudo apt-get update

Docker apt-key추가

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Docker Repository 추가

add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

Docker 설치

sudo apt-get install docker-ce docker-ce-cli containerd.io

Docker AmazoneCorreto:11(JDK) Image 다운로드

sudo docker pull amazoncorretto:11

Docker Mysql Image 다운로드

sudo docker pull mysql

Docker Mysql Container 실행

sudo docker run --name mysql -e MYSQL_ROOT_PASSWORD=Gy8zn152Yh -p3306:3306 -d mysql

Docker Nginx Image 다운로드

sudo docker search nginx
sudo docker pull nginx

Docker Nginx 실행하기

sudo docker run -d -p 80:80 -p 443:443 --restart always --name nginx -v ~/nginx/static:/usr/share/nginx/html -v ~/nginx/config:/etc/nginx nginx

Docker nginx Shell 실행

Docker exec nginx /bin/bash

Docker nginx Shell 나오기

exit

Docker Reverse Proxy 셋팅 (Cors 에러도 해결)

apt-get update

apt-get install vim

vim ~/nginx/config/nginx.conf

//vim 진입

//"#include /etc/nginx/conf.d/*.conf;"

▼ 하단에 추가

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root   /usr/share/nginx/html;
    index index.html index.htm;
    server_name front;
    location / {
            try_files $uri $uri/ /index.html;
    }

    location /api {
            proxy_pass http://i9a105.p.ssafy.io:8080/api;
            proxy_redirect off;
            charset utf-8;

            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_Nginx-proxy true;
    }
}
```

Docker nginx CertBot(Https ssl)

Docker exec -it nginx /bin/bash

//nginx 콘솔로 진입되어야함.

apt-get update

apt-get install software-properties-common certbot python3-certbot-nginx

certbot —nginx -d DomainNameAddress

exit

//nginx 콘손을 빠져나와야함.

Docker Jenkins 이미지 다운로드

Docker pull jenkins/jenkins

Docker Jenkins 실행하기 - ()안은 옵션

Docker run -d —name jenkins -restart always -p 8181:8181 -v /var/jenkins_home:/var/jenkins_home -v

/var/run/docker.sock:/var/run/docker.sock -v ~/nginx/static:/var/jenkins_home/front -e TZ=Asia/seoul jenkins/jenkins (-u root)
//웹브라우저에서 “http://ip:8181”로 접속!

//private key를 입력하라는 창에서 콘솔에서 입력시 키값이 보임.

docker logs jenkins

Docker Jenkins GitLab 계정 등록하기

DashBoard > Jenkins 관리 > Credentials > (Global)클릭 > AddCredentials

KIND : User And Password 선택

Scope : Global

Username : Gitlab 아이디 입력

Password : GitlabPassword OR Token입력. (Token은 아래 펼쳐보기 참조.)

▼ Token펼쳐보기

GitLab 로그인 후 해당 Repository로 이동.

Setting > Access Tokens

Token Name : 원하는 이름 입력(해당 이름으로 토큰이 관리됨)

Expiration Date : 만료일

Select a Role : 권한(Dev 이상 선택)

모든 체크 박스 체크후 생성 클릭.

주의 토큰은 다시 볼 수 없습니다.

상단 Your new project access token 복사후 잃어버리지 않도록 주의

ID : Jenkins에서 관리될 이름 작성

Docker Jenkins Gitlab WebHook하기.

DashBoard > 새로운 Items

Enter an item name : jenkins에서 관리될 Item name

아래 택1 이후 OK클릭

FreeStyle : Command를 통하여 CI/CD

PipeLine : Script + Command를 통한 CI/CD

Folder : 모름

MultoBranch PipeLine : Branch별 Command를 통한 CI/CD 가능

&&&기초를 위하여 FreeStyle로 작성&&&

General > 설명

소스 코드 관리

None > Git으로 변경

Repository URL : 자동 배포할 Git 주소입력

Credentials : Jenkins GitLab 계정 등록하기에서 만든 Jenkins에서 관리될 이름 클릭

Branches to build : */master

Build when a change is pushed to GitLab. GitLab webhook URL : Check ** << 하단 URL주소 필수 복사 필요. Ex) ~~~~.p.ssafy.io:8181/project/~~

Push Events : Check
Opened Merge Request Events : Check
Approved Merge Requests (EE-only) : Check

고급(확장)

맨 하단 Secret token도 복사

▼ Execute Shell 안에 입력.

```
cd backend
ls
pwd
chmod 667 gradlew
./gradlew build
if [ $(docker ps -aqf "name=back") ]; then
docker stop back
fi

echo "spring:
datasource:
driver-class-name: com.mysql.cj.jdbc.Driver
username: root
password: Gy8zn152Yh
```

url: jdbc:mysql://i9a105.p.ssafy.io:3306/S09P11A105?
serverTimezone=UTC&useUnicode=true&characterEncoding=utf8
hikari:
maximum-pool-size: 2
jpa:
properties:
hibernate:
show_sql: true
format_sql: true
hibernate:
ddl-auto: update
database-platform: org.hibernate.dialect.MySQL8Dialect

mail:
host: smtp.naver.com
port: 587
username: cafet123@naver.com
password: cafet!23
properties:
mail:
smtp:
auth: true
starttls:
enable: true
required: true
startssl:
enable: true
required: true
sender: cafet123@naver.com
jwt:
key: asdfasdfasdfafsdadsfdaffasdsadffasdfasdfasdfafsdadsfdaffasdsadffasdfasdfasdfafsdadsfdaffasdsadff
accessTokenExpirationTime:
int: 3000000
refreshTokenExpirationTime:
int: 60
whiteList:
/swagger/**,**
/swagger-resources/,
/health,
/v3/api-docs/**,**
/swagger-ui/,
/api/boardfile/**,**
/api/user/logout,
/api/user/login/,
/api/user/new/**,**
/api/board/,
/api/shop/**,**
/api/location/,
/api/menu/**,**
/api/mail/

opendata:
key:
G3uj7g7AEmUsSHLZV4LlslK73nTPC7mo/t7LfzMO5Q8g0fJjdmml+iyF12xdZUmYXFy+CT083HHACNQgny4WVw

mail:
regist:
ExpirationTime:
int: 5

CheckExpirationTime:
int: 5

Kakao api

kakao:
map:
key: KakaoAK 3f4683b5cba1076dea94d245d77fde0f

AWS Account Credentials

cloud:
aws:
credentials:
accessKey: AKIAYXKKVNOPXKJX3NE4
secretKey: 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
s3:
bucket: picturepractice
bucket-url: https://picturepractice.s3.ap-northeast-2.amazonaws.com/
region:
static: ap-northeast-2
stack:
auto:
false

" > /var/jenkins_home/workspace/web/backend/src/main/resources/application.yml

docker build -t backend --build-arg JAR_FILE=build/libs/PetManBE-0.0.1-SNAPSHOT.jar -f Dockerfile .
docker run -it -d --rm -p 8080:8080 -e TZ=Asia/Seoul --name back backend
cd ./../frontend
touch .env;
echo "REACT_APP_KAKAOMAP_KEY = " >> ./.env
npm install
npm run build

창 닫지 말고 Gitlab과 같이 해야함.

DockerFile 생성

▼ BackEnd DokcerFile생성. 이하 내용

FROM amazoncorretto:11
LABEL authors="송진현"
VOLUME /tmp
ARG *JAR_FILE*
COPY ${*JAR_FILE*} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]

Docker Jenkins DOD install

//host에서

sudo docker exec -it jenkins bash

▼ //jenkins continer 안에서 실행

apt-get update
apt-get install ca-certificates curl gnupg lsb-release
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \ $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update
apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
exit

gitlab WebHook

원하는 GitRepository 이동

좌측 하단 Settings > Webhook > URL에
Docker Jenkins Gitlab WebHook 표시된 URL 붙여넣기.

Secrect Token도 붙여 넣기.

Push Event : check

Tag push events : Check

Merger Request Events : Check

Enable SSL Verification : Check