

# 2024.07.09

## 오전

안되면 mvn clean

mvn package

sudo chown -R hadoop:hadoop ./demo

rm -rf .git

포트 확인

sudo netstat -antp | grep 8080

프로세스 확인

sudo ps -ef | grep [pid]

프로세스 죽이기

sudo kill -9 [pid]

docker pull openjdk:17

docker file 만들기

spring boot 프로젝트 폴더에서

/home/hadoop/workspace/demo

vim Dockerfile

FROM openjdk:17

EXPOSE 80

COPY ./target/de31-0.0.2-SNAPSHOT.jar /opt/de31-0.0.2-SNAPSHOT.jar

WORKDIR /opt

ENTRYPOINT ["java", "-jar", "de31-0.0.2-SNAPSHOT.jar"]

**docker build -t myimg:0.1 .**

tag를 붙여달라는거 .은 현재폴더..

demo docker run -d --name myboot -p 80:8080 myimg:0.1

http://ip:80/api/v1/post-api/domain

확인해보기..

run을 돌렸으니 서버가 고정이 되어버렸다 entrypoint가 되었으므로...

docker exec -it myboot /bin/bash

**docker run -d --name myboot -p 80:8080 --restart always myimg:0.1**

mvnw clean은 maven이 설치가 안되어도 돌아가게 한다.. maven wrapper

:set nonu

FROM openjdk:17

EXPOSE 80

COPY ./demo /tmp

WORKDIR /tmp

RUN ./mvnw clean package

RUN mv target/demo-0.0.2-SNAPSHOT.jar /opt/demo-0.0.2-SNAPSHOT.jar

WORKDIR /opt

ENTRYPOINT ["java", "-jar", "demo-0.0.2-SNAPSHOT.jar"]

docker build -t myimg:0.5 .

./demo가 있어 ㅇ하므로 workspace에서하기..

demo docker run -d --name myboot -p 80:8080 myimg:0.5

http://ip:80/api/v1/post-api/domain

docker exec -it myboot /bin/bash

들어가서 확인해보자 어디에 들어가있는지..

멀티스테이지 빌드

FROM openjdk:17 AS int-build

EXPOSE 80

COPY ./demo /tmp

WORKDIR /tmp

RUN ./mvnw clean package

FROM openjdk:17

COPY --from=int-build /tmp/target/demo-0.0.2-SNAPSHOT.jar /opt/demo-0.0.2-SNAPSHOT.jar

WORKDIR /opt

ENTRYPOINT ["java", "-jar", "demo-0.0.2-SNAPSHOT.jar"]

docker build -t myimg:0.6 .

docker images

경량화된 도커 자바 image로 용량을 줄이기

일단 docker pull **gcr.io/distroless/java:11**

Docker file **gcr.io/distroless/java:11** 바꾸고

`docker build -t myimg:0.7 .`

`docker run -d --name myboot7 -p 80:8080 myimg:0.7`

view logs로 화긴하기..

포트 안겹치게 조심히 지우기...

ifconfig치면 ve머시기 뜨는데 이게 docker container 네트워크 장비이다.virtual

`docker stop myboot` 멈추는순간 바로 가상네트워크장비가 내려간다.

192는 c

172는 b private

`docker run -itd ubuntu:22.04`

이렇게 하면 이름이 랜덤으로 정해진다..

로드밸런스구축 nginx

부하를 분산 o 늘 해준다...

`mkdir loadbal`만들어서

`sudo apt upgrade`하기

80번포트는 nginx꺼

`sudo apt install nginx -y`

`sudo service nginx start`

`sudo service nginx status`

`docker build -t phpserver:1.0 .`

`mkdir lb01 lb02 lb03`

\*\*

`docker run -itd -p 5001:80`

`-h nginx-lb01`

`-v ./lb01:/var/log/apache2`

`-e SERVER_PORT=5001`

`--name=nginx-lb01`

`phpserver:1.0`

`docker run -itd -p 5002:80`

`-h nginx-lb02`

`-v ./lb02:/var/log/apache2`

`-e SERVER_PORT=5002`

```
--name=nginx-lb02
```

```
phpserver:1.0
```

```
docker run -itd -p 5003:80
```

```
-h nginx-lb03
```

```
-v ./lb03:/var/log/apache2
```

```
-e SERVER_PORT=5003
```

```
--name=nginx-lb03
```

```
phpserver:1.0
```

파일복사

\*\*

```
docker cp index.php3 nginx-lb01:/var/www/html/index.php
```

```
docker cp index.php3 nginx-lb02:/var/www/html/index.php
```

```
docker cp index.php3 nginx-lb03:/var/www/html/index.php
```

\*\*

```
sudo vim /etc/nginx/nginx.conf
```

그냥 40dd 치면 지워짐,.vim에서

\*\*

```
events {
```

```
    worker_connections 1024;
```

```
}
```

```
http {
```

```
    upstream backend-lb {
```

```
        server 127.0.0.1:5001;
```

```
        server 127.0.0.1:5002;
```

```
        server 127.0.0.1:5003;
```

```
    }
```

```
    server {
```

```
        listen 80 default_server;
```

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

```
# 연결 프락시 정보
```

```
location / {
```

```
    proxy_pass    http://backend-lb;
```

```
}
```

```
}
```

```
}
```

```
**
```

```
sudo service nginx restart
```

```
// status
```

localhost에 nginx설치되어있더 80 port

curl localhost는 뭐지

config파일만 고침...

라인드로빈형식이다... 1->2->3->1

nginx는 지시만하고있다.

http://ip/ 확인해봐도된다..

```
server 127.0.0.1:5001 weight=6;
```

```
14 server 127.0.0.1:5002 weight=2;
```

```
15 server 127.0.0.1:5003 weight=2;
```

재가동

least\_conn; 옵션변경 놓고있는 애들을 굴려라!!

/home/hadoop/workspace/loadbal/lb01

여기에 acces,log파일을 보면 tail -f로봐라..

```
**# Method -> Put
```

- Web API를 사용해서 DB 같은 저장소에 리소스 값을 업데이트하는데 목적\*\*  
1번 2번 3번의 차이는?

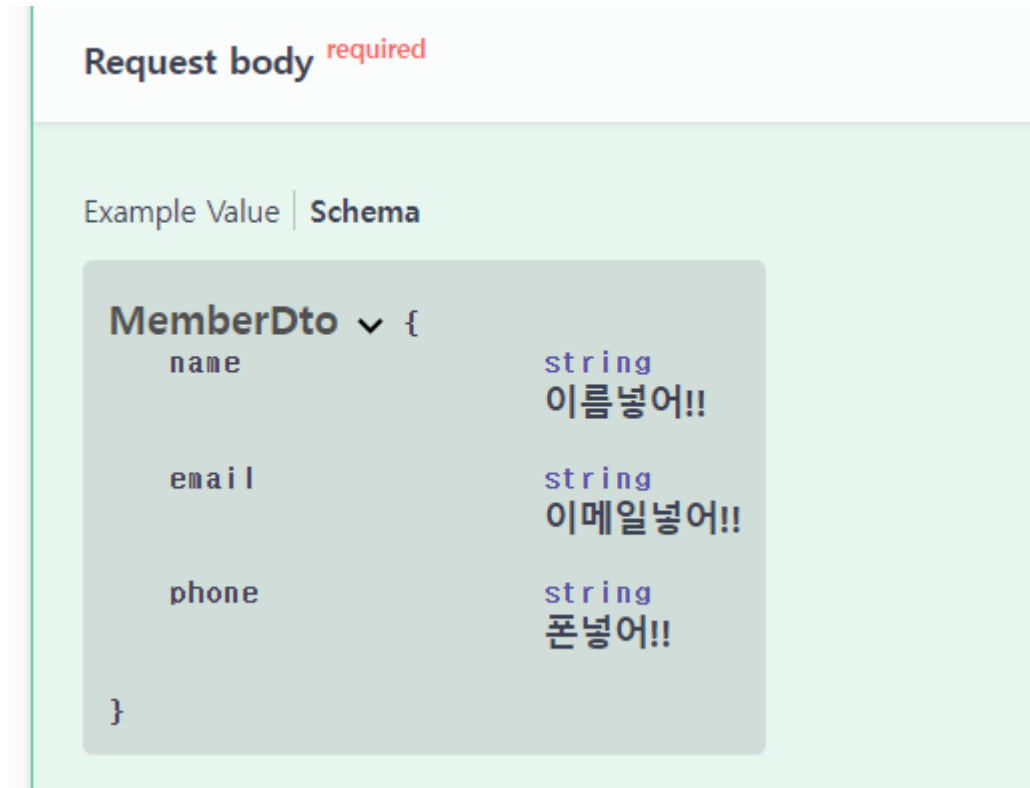
3번은 json 응답코드가 다르다..

1번 200

3번은 202다

스웨거..

컨트롤러에 hidden 적어놓으면 다양한게 안보인다..



# docker -> mysql

\*\*

```
docker run -d --name mydb2 -v /home/hadoop/mydb:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=123 -e MYSQL_DATABASE=springboot -p 13306:3306
mariadb:10.5
```

\*\*

내 ip 비번 123 13306 포트