Jenkins, Docker + Web + DNS - 실습가이드

최초 작성일 : 2024/02/16

최종 제출일 : 2024/02/19

김민경

내용

1. 개념정리

1. Jenkins

1) 정의

• 모든 언어의 조합과 소스 코드 레포지토리에 대한 **지속적인 통합(Continuous integration, CI)**과 **지속적 배포(continuous delivery, CD)** 환경을 구축하기 위한 도구

• 빌드, 테스트, 배포 프로세스를 자동화하여 소프트웨어 품질과 개발 생산성을 높일 수 있음

2) 장점

• 프로젝트 표준 컴파일 환경에서 컴파일 오류 검출

• 자동화 테스트 수행

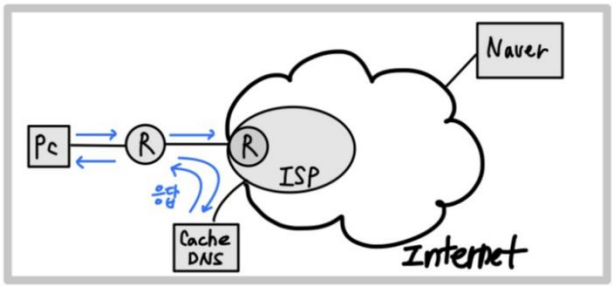
• 정적 코드 분석에 의한 코딩 규약 순수여부 체크

• 프로파일링 툴을 이용한 소스 변경에 따른 성능 변화 감지

• 결합 테스트 환경에 대한 배포 작업

2. DNS

1) 동작과정



www. naver . com 입력
Host Name Domain

① 본인 PC를 뒤짐 (DNS Cache)

② if 본인 PC에 naver.com에 해당하는 주소 X시
host file 뒤짐

③ if host file에도 naver.com에 해당하는 주소 X시
DNS에게 물어봄
하지만 대부분 공유기는 갖추고 있어 공유기가 대신 응답을 하기도 함
↳ DNS 포워딩 기능
(자기가 DNS인 것처럼 응답을 대신 함)

④ if 공유기가 응답 X하면
ISP의 DNS에게 물어봄

⑤ ISP의 DNS Cache가 naver.com의 IP를 PC에게 전달

⑥ 그러면 이제 PC의 DNS Cache에
naver.com의 IP주소가 저장됨
∴ 그 다음부터 PC에서 naver.com 입력시
ISP의 DNS에게 요청하는 것이 X라
DNS Cache에 저장된 IP 주소로 바로 연결
(But 유효기간이 있어서 유효기간 지나면 Cache 사라짐)
ex) 유효기간 : 5분
5분 전까지는 Cache에 naver.com의 IP 주소가 저장되어 있어
naver.com 입력시 바로 해당 IP로 연결되지만,
5분 후에는 Cache에서 사라져 다시 ISP의 Cache DNS에게 ask

3. Docker Network

1) 정의

• 컨테이너 생성 시 **NET namespace**라는 기술을 통해 구현된 가상화 기법을 사용하여 각자 독립된 네트워크 공간을 할당 받음

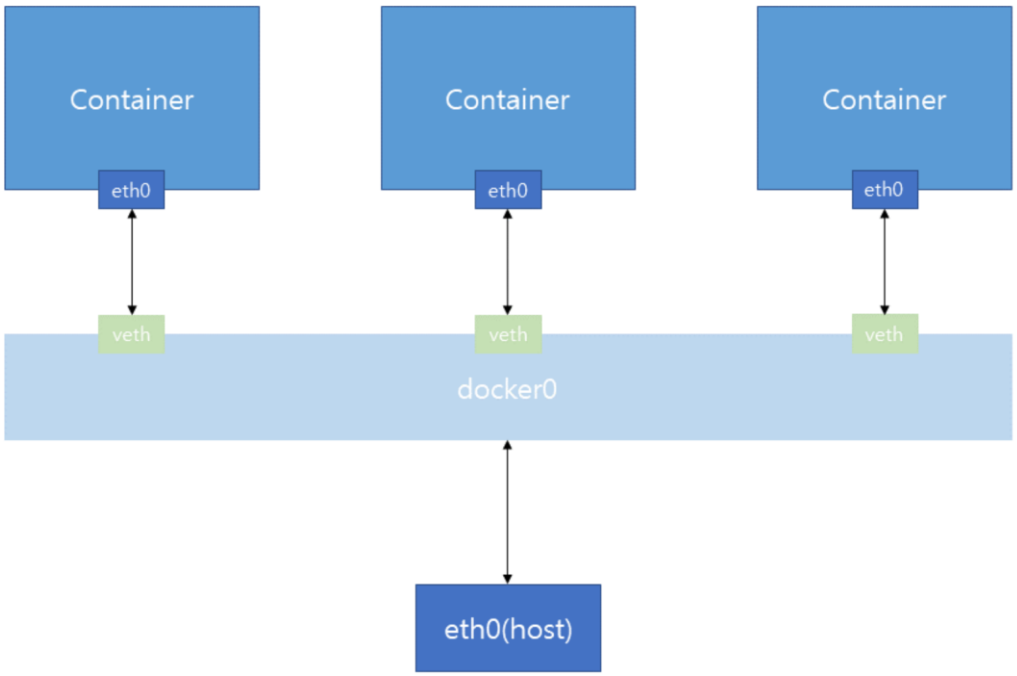
참조) Namespace

• 프로세스 실행 시 시스템의 리소스를 분리해서 실행할 수 있도록 도와주는 기능

참조) Net Namespace

• 네트환경을 분리하면 Namespace에 속한 프로세스들에 새로운 IP를 부여하거나 네트워크 인터페이스를 추가하는 것이 가능하며 네트워크 Namespace별로 개별적인 인터페이스와 ip 구성을 가짐

2) 구조



• 컨테이너 생성 시 컨테이너는 host와 통신하기 위해 네트워크 인터페이스인 eth0를 할당받음

• 동시에 host도 네트워크 인터페이스인 veth(virtual ethernet)를 할당받음

• veth 인터페이스는 docker 0와 바인딩되고, docker 0은 host의 eth0 인터페이스와 연결되어 외부로 들어온 요청을 처리함

Ⅱ. 실습 1번(Jenkins)

1. 실습 과정

1) 기존에 설치된 모든 컨테이너와 이미지 정지 및 삭제

sudo docker stop $(sudo docker ps) //실행 중인 모든 컨테이너 정지

sudo docker rm -f $(sudo docker ps -a) //모든 컨테이너 강제 삭제

sudo docker rm- -f $(sudo docker ps -a) //모든 이미지 강제 삭제

2) 도커 기능을 사용하여 Jenkins 검색

```
root@ubuntu:~# docker search jenkins --filter is-official=true
NAME      DESCRIPTION                                     STARS     OFFICIAL
jenkins   DEPRECATED; use "jenkins/jenkins:lts" instead   5688      [OK]
root@ubuntu:~#
```

3) Jenkins를 사용하여 설치

docker pull Jenkins/Jenkins //registry로 jenkins/jenkins 이미지 다운로드 하기

```
root@ubuntu:~# sudo docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
1b13d4e1a46e: Pull complete
0f000f100427: Pull complete
7f1ba56be45d: Pull complete
fc4f46d69995: Pull complete
8c78259e0f06: Pull complete
efb662c90d75: Pull complete
788955f514e8: Pull complete
18ac78cb58f1: Pull complete
d17d3c5a675f: Pull complete
f81b6e9fa7dd: Pull complete
57f4339a7c3e: Pull complete
a16fa6f385a3: Pull complete
Digest: sha256:75c908c34bf45fcf50cae779155262fa6a2d85dfd7c14b5d6b5e664521c766bf
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
root@ubuntu:~#
```

docker images -a //다운받은 모든 이미지 확인하기

```
root@ubuntu:~# docker images -a
REPOSITORY        TAG       IMAGE ID       CREATED      SIZE
jenkins/jenkins   latest    db75aa81ca2b   5 days ago   475MB
root@ubuntu:~#
```

4) Jenkins 포트로 접속하여 웹 서비스 열기

• Jenkins의 전용 포트 : 8080

docker run -d --name jenkins -p 81:8080 jenkins/jenkins

//외부에서 접근할 포트로 81포트를 사용함. 이 포트는 사용하지 x는 포트를 사용하는 것이 좋음

```
root@ubuntu:~# docker run -d --name jenkins -p 81:8080 jenkins/jenkins
841506418c874a8e4442656f992a14f45b66e94092de6ff85e72d0b99b629b36
```

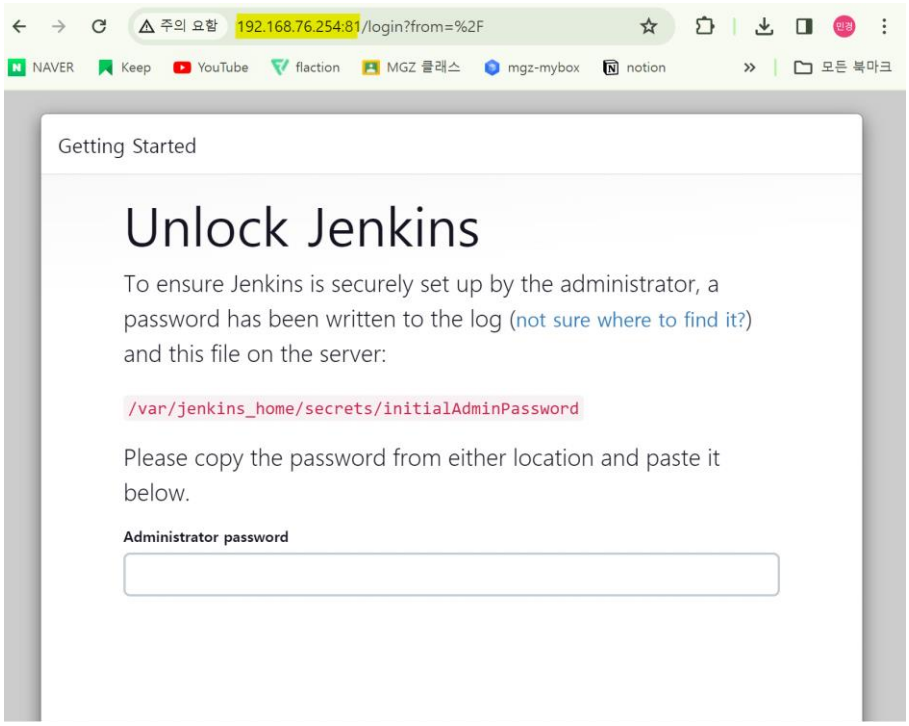
docker ps -a //확인하기

```
root@ubuntu:~# docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
841506418c87   jenkins/jenkins   "/usr/bin/tini -- /u…"   3 minutes ago   Up 3 minutes   50000/tcp, 0.0.0.0:81->8080/tcp, :::81->8080/tcp   jenkins
```

5) Real pc에서 접근 확인하기

• docker의 ip는 가상 ip임

⇒ 실제 pc에서 확인이 불가능하기 때문에 U-S의 ip로 접근하기



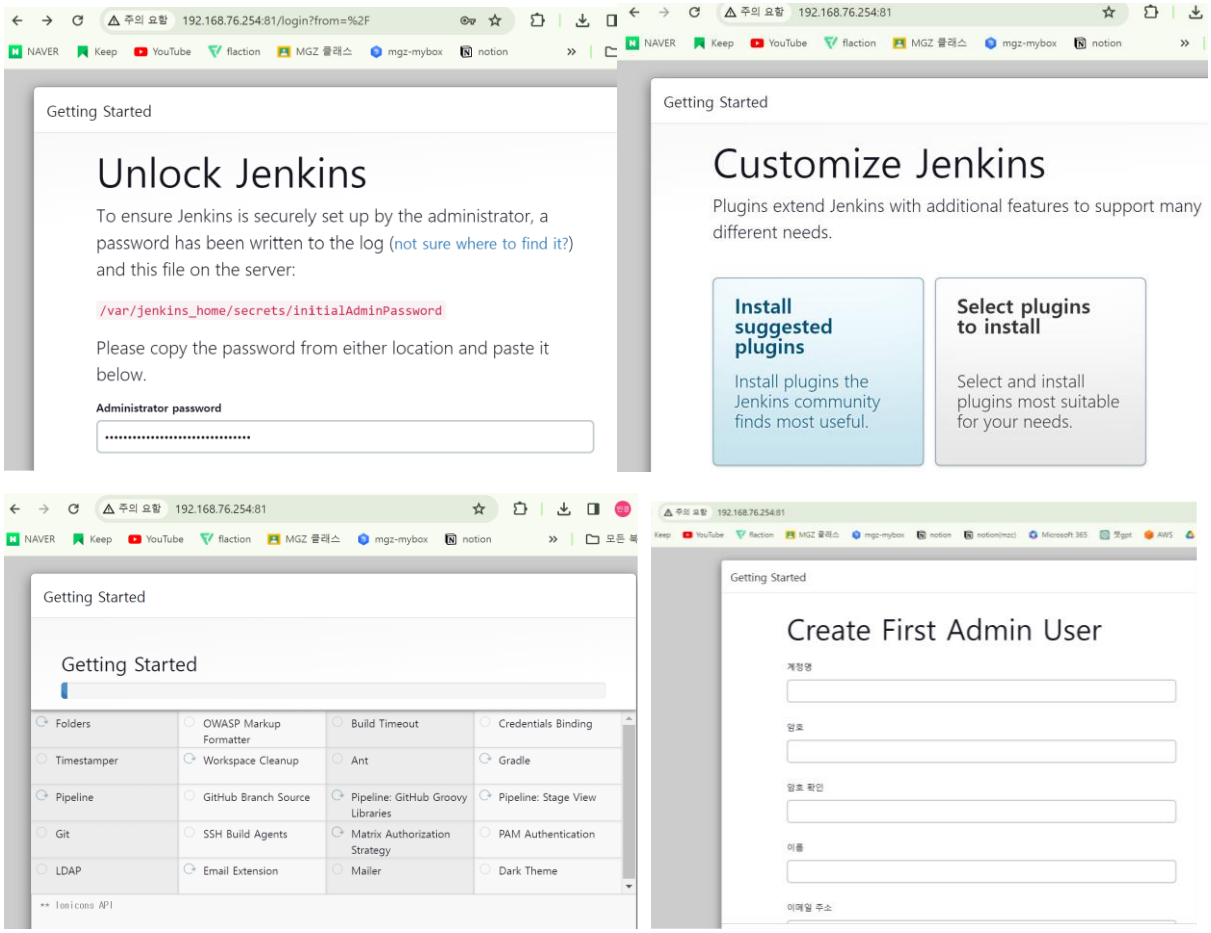
6) Jenkins의 초기 패스워드 찾아서 로그인하기

6-1) 초기 패스워드 찾기

docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword

```
root@ubuntu:~# docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
21eac241e099461193ff975a7a0c9dab
```

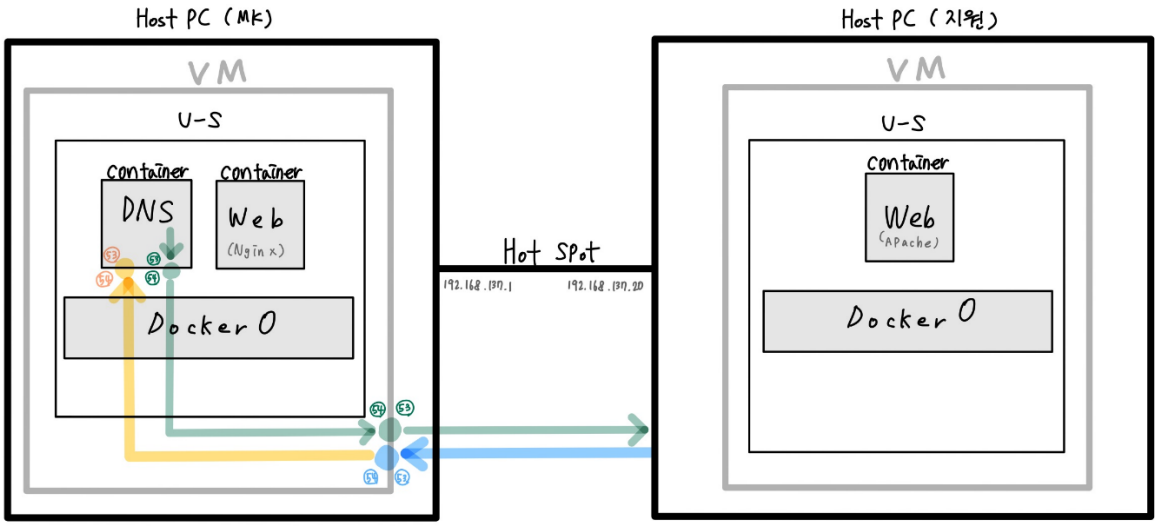
6-2) 로그인하기



Ⅲ. 실습 2번

1. 실습 설명

1) DNS 요청과정



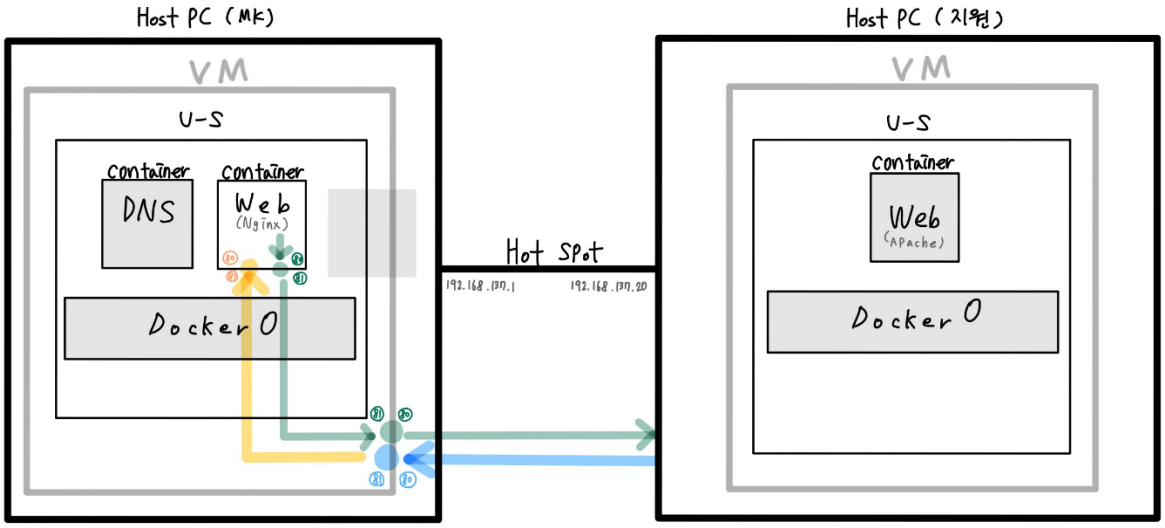
① 지원 host pc에서 'wonkima.com' 입력 (민경pc의 DNS서버에게 ip를 물어봄)

② VM에서 포트포워딩 되며 VM에 접근 (192.168.137.1:53 → 10.0.2.15:54)

③ DNS에서 포트포워딩되며 DNS에 접근 (VM의 54번 포트 → 53번 포트)

④ 같은 방식으로 DNS & VM에서 포트포워딩 되면서 지원 host pc로 ip 전달

2) HTTP 요청과정



① 지원 host pc에서 민경 host pc에게 받은 ip 주소 '197.168.137.1'를 입력

② VM에서 포트포워딩 되며 VM에 접근 (192.168.137.1:80 → 10.0.2.15:81)

③ Web에서 포트포워딩되며 DNS에 접근 (VM의 81번 포트 → 80번 포트)

④ 같은 방식으로 DNS & VM에서 포트포워딩 되면서 지원 host pc로 ip 전달

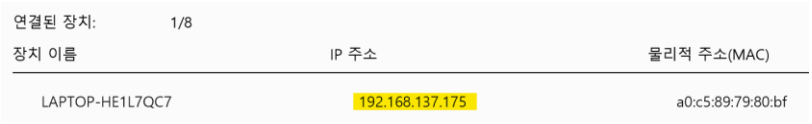
2. 실습 과정

1) 환경 구성

- 민경 pc와 지원 pc에 컨테이너 설치하기
- 민경 pc의 핫스팟에 지원 pc를 연결하기

-민경 pc의 ip: 192.168.137.1

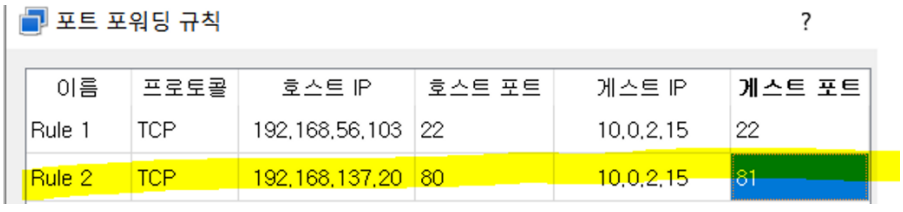
-지원 pc의 ip: 192.168.137.175



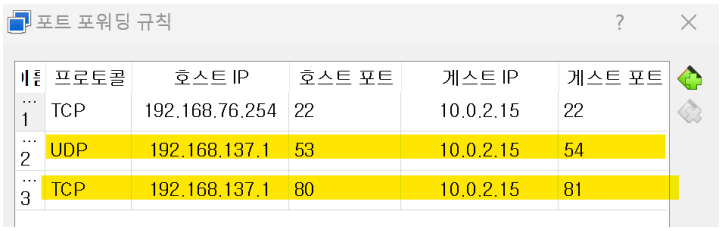
2) Docker container에 Nginx를 설치하여 웹 브라우저로 접근하기

2-1) VM 포트포워딩 설정하기

- 지원pc) Web 접근에 대한 포트포워딩 설정하기



- 민경pc) Web & DNS 접근에 대한 포트포워딩 설정하기



- DNS 패킷은 UDP 기반으로 이루어짐

2-2) [민경pc] 컨테이너 실행

docker run --name nginx -d -p 81:80 nginx

```
root@ubuntu:~# docker run --name nginx -d -p 81:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e1caac4eb9d2: Pull complete
88f6f236f401: Pull complete
c3ea3344e711: Pull complete
cc1bb4345a3a: Pull complete
da8fa4352481: Pull complete
c7f80e9cdab2: Pull complete
18a869624cb6: Pull complete
Digest: sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
Status: Downloaded newer image for nginx:latest
907135e1e6b9b652b9110315ad71fc0d0038ddae75b7fa48f8c30f9d46802195
root@ubuntu:~#
```

```
root@ubuntu:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
907135e1e6b9   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:81->80/tcp, :::81->80/tcp   nginx
```

docker exec -it nginx /bin/bash // nginx 컨테이너 안으로 접근

apt-get update

apt-get install nginx

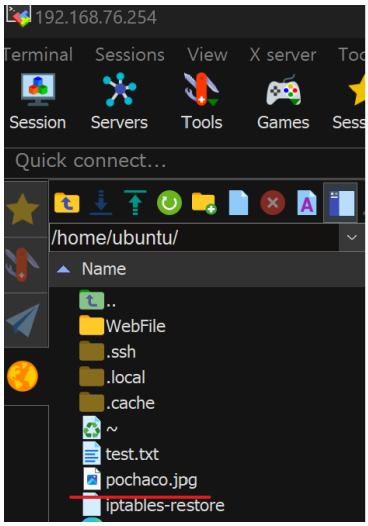
apt-get install vim // 파일 편집을 위해 설치

service nginx status // nginx 서버가 잘 작동하는 지 확인

```
root@907135e1e6b9:/# service nginx status
nginx is running.
```

2-3) nginx 웹에 이미지 넣기

- 삽입하고자 하는 사진을 모바엑스텀으로 옮기기(Real PC → 민경 U-S)



- 삽입하고자 하는 사진을 복사하기 (민경 U-S → nginx 컨테이너)

  - ctrl+p+q 로 컨테이너 잠깐 나가서 진행하기

    docker cp /home/ubuntu/pochaco.jpg nginx:/usr/share/nginx/html/

```
root@ubuntu:~# docker cp /home/ubuntu/pochaco.jpg nginx:/usr/share/nginx/html/
Successfully copied 24.1kB to nginx:/usr/share/nginx/html/
```

  - 복사 정상적으로 되었는지 확인

```
root@907135e1e6b9:/usr/share/nginx/html# ls
50x.html  index.html  pochaco.jpg
```

- index.html 파일 수정하기

  - **vim index.html**

  - **cat index.html**

```
root@907135e1e6b9:/usr/share/nginx/html# cat index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to MinKyeong Server!</h1>
<h2> This is Nginx</h2>
<img src="pochaco.jpg" width="400" height="400">
</body>
</html>
root@907135e1e6b9:/usr/share/nginx/html#
```

  - **service nginx restart** // nginx 재시작하기

2-4) [민경pc] 웹 접근하기

- but 웹 접근이 안됐었음

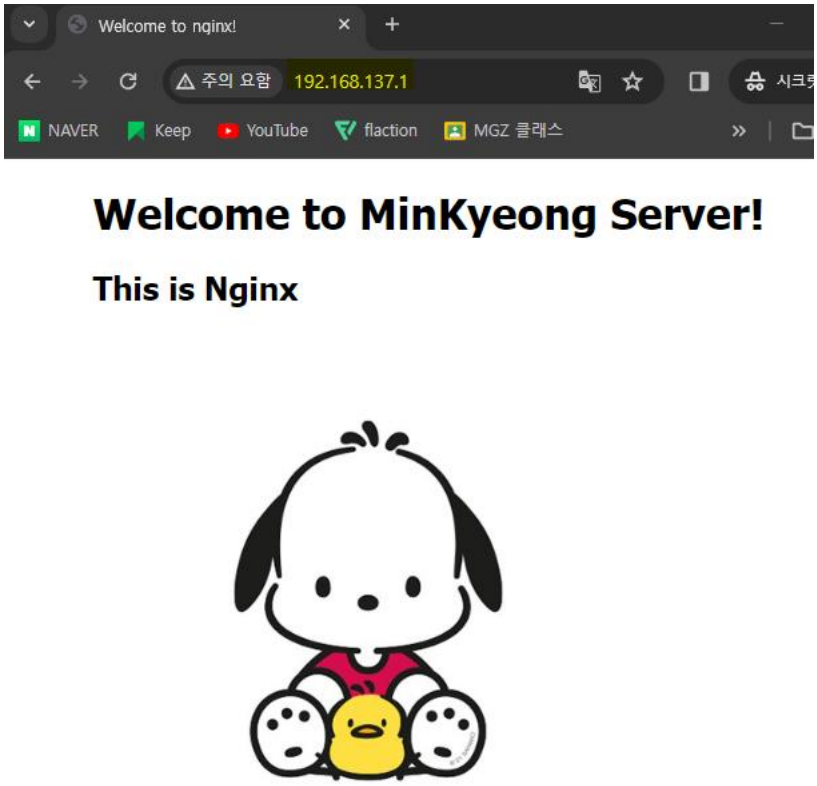
  ⇒ pochaco.jpg에 권한이 부여되지 않은 상태라 안되었던 것임

```
root@61a9e03e81b7:/usr/share/nginx/html# ls -al
total 60
drwxr-xr-x 1 root root  4096 Feb 19 01:32 .
drwxr-xr-x 1 root root  4096 Feb 14 19:53 ..
-rw-r--r-- 1 root root 12288 Feb 19 01:32 .index.html.swp
-rw-r--r-- 1 root root   497 Feb 14 16:03 50x.html
-rw-r--r-- 1 root root   344 Feb 19 01:23 index.html
-rw-rw-r-- 1 1000 1000 22219 Feb 18 18:33 pochaco.jpg
```

- 권한 부여하기

```
root@61a9e03e81b7:/usr/share/nginx/html# chmod 777 pochaco.jpg
```

- 결과 확인하기(웹 접근하기)



2-5) 지원 pc에서 접근하기

3) Docker container에 DNS를 설치하여 도메인 이름으로 웹 브라우저 접근하기

- 포트포워딩 설정하기

... 2	UDP	192.168.137.1	53	10.0.2.15	54

3-1) DNS라는 컨테이너 생성 및 포트포워딩

```
root@ubuntu:~# docker run -it --name dns -p 54:53/udp ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
01007420e9b0: Pull complete
Digest: sha256:f9d633ff6640178c2d0525017174a688e2c1aef28f0a0130b26bd5554491f0da
Status: Downloaded newer image for ubuntu:latest
root@e06c08a91584:/#
```

apt-get update

apt-get install bind9 -y // DNS 설정을 위해 bind9 설치

apt-get install nano -y //파일 설정 변경을 위해 설치

apt-get install dnsutils //DNS 관련 작업을 수행할 때 사용하는 패키지

https://yoo11052.tistory.com/208

https://jiheon95.tistory.com/55

젠킨스

https://narup.tistory.com/179

https://velog.io/@rnqhstlr2297/Jenkins-%EA%B0%9C%EB%85%90%EA%B3%BC-%EC%82%AC%EC%9A%A9

수민님 가이드