

- Ch_2 _ PXE 서버 구축
- Ch_3 _ 도커 개념정리 및 실습 구축
- PXE 설치 서버의 개념과 구현

➤ PXE 설치 서버의 개념도



- PXE 설치를 위해서는 PXE 설치 서버와 컴퓨터가 모두 같은 네트워크 안에 있어야 한다

```
root@server:~/바탕화면# apt -y install isc-dhcp-server tftpd-hpa inetutils-inetd vsftpd pxelinux
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
```

- ➔ isc-dhcp-server tftpd-hpa inetutils-inetd vsftpd pxelinux 5 개의 패키지 설치

```
ubuntu-20.04-live-server-amd64.iso 93%[+++++
ubuntu-20.04-live-server-amd64.iso 95%[+++++
ubuntu-20.04-live-server-amd64.iso 99%[+++++
ubuntu-20.04-live-server-amd64.iso 100%[+++++
2022-03-30 19:05:41 (233 KB/s) - 'ubuntu-20.04-live-server-amd64.iso' 저장 함 [
ubuntu@Server:/srv/ftp/pub$
ubuntu@Server:/srv/ftp/pub$ ls
ubuntu-20.04-live-server-amd64.iso
ubuntu@Server:/srv/ftp/pub$
ubuntu@Server:/srv/ftp/pub$
ubuntu@Server:/srv/ftp/pub$
```

- ➔ 실습을 위해 iso 파일 설치

```
root@Server:/srv/ftp/pub#
root@Server:/srv/ftp/pub#
root@Server:/srv/ftp/pub# mount ubuntu-20.04-live-server-amd64.iso /mnt
mount: /mnt: WARNING: device write-protected, mounted read-only.
root@Server:/srv/ftp/pub#
root@Server:/srv/ftp/pub# cp /mnt/casper/vm* /srv/tftp/
root@Server:/srv/ftp/pub# cp /mnt/casper/initrd* /srv/tftp/
root@Server:/srv/ftp/pub# cp /usr/lib/syslinux/modules/bios/ldlinux.c32 /srv/tftp/
> ^C
root@Server:/srv/ftp/pub# cp /usr/lib/syslinux/modules/bios/ldlinux.c32 /srv/tftp/
root@Server:/srv/ftp/pub#
```

- chmod -R 777 /srv/ftp/pub/ 디렉터리 권한 설정

- mount ubuntu... /mnt 설치한 iso 파일 마운트

- /mnt/casper/vm* /mnt/casper/initrd* /usr/lib/syslinux... 파일 /srv/tftp 디렉터리에 복사 (cp)
- 이후 iso 파일 언마운트

```
root@Server:/srv/ftp/pub# cd /srv/tftp/
root@Server:/srv/tftp#
root@Server:/srv/tftp# wget http://dw.hanbit.co.kr/ubuntu/20.04/pxelinux.0
--2022-03-31 18:13:22-- http://dw.hanbit.co.kr/ubuntu/20.04/pxelinux.0
dw.hanbit.co.kr (dw.hanbit.co.kr) 해석 중... 218.38.58.196
다음으로 연결 중: dw.hanbit.co.kr (dw.hanbit.co.kr)[218.38.58.196]:80... 연결했습니다.
HTTP 요청을 보냈습니다. 응답 기다리는 중... 200 OK
길이: 42446 (41K) [text/plain]
저장 위치: 'pxelinux.0'

pxelinux.0          100%[=====] 41.45K  --.-KB/s   / 0.02s
2022-03-31 18:13:22 (1.64 MB/s) - 'pxelinux.0' 저장함 [42446/42446]
root@Server:/srv/tftp#
```

- /srv/tftp/ 디렉터리에서 파일 다운로드

```
115
116 subnet 192.168.111.0 netmask 255.255.255.0
117         option routers 192.168.111.2;
118         option subnet-mask 255.255.255.0;
119         range dynamic-bootp 192.168.111.120 192.168.111.199;
120         option domain-name-servers 192.168.111.2;
121         allow booting;
122         allow bootp;
123         next-server 192.168.111.100;
124         option bootfile-name "pxelinux.0";
125
```

- /etc/dhcp/dhcpd.conf 파일 내용 추가

```
default
/srv/tftp/pxelinux.cfg

1 DEFAULT Ubuntu_Auto_Install
2
3 LABEL Ubuntu_Auto_Install
4     kernel vmlinuz
5     initrd initrd
6     append ip=dhcp url=ftp://192.168.111.100/pub/ubuntu-20.04-live-server-am64.iso
```

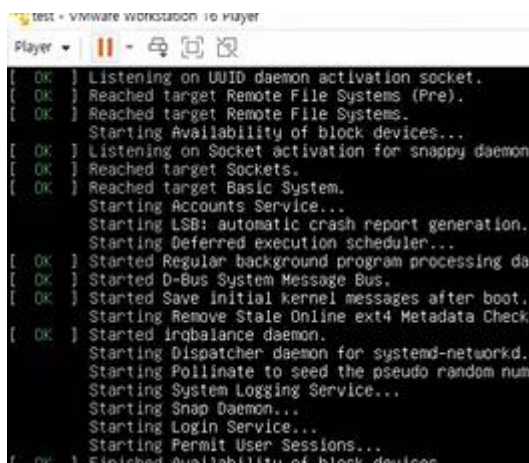
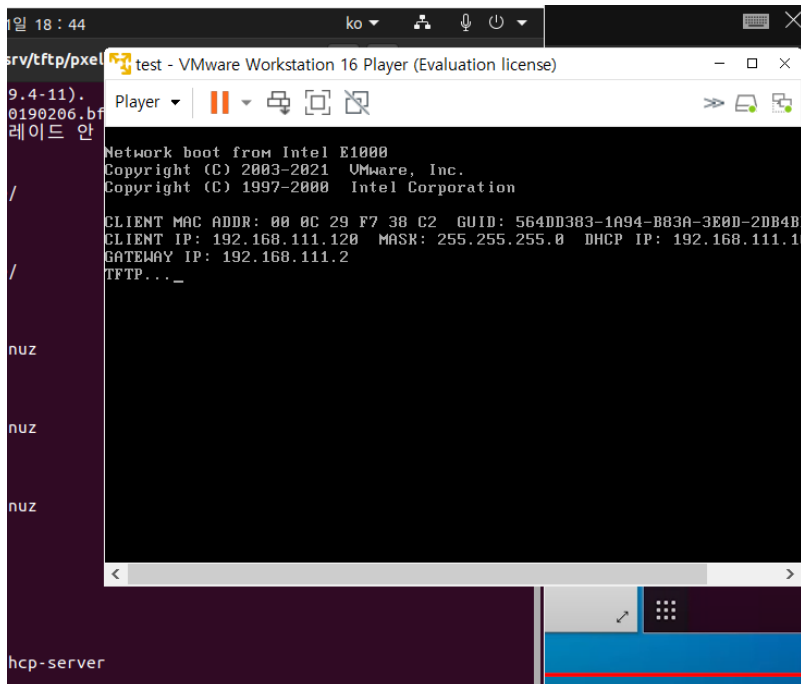
- /srv/tftp/pxelinux.cfg 파일 내용 추가

```
24 # Allow anonymous FTP? (Disabled by default).
25 anonymous_enable=YES
26 #
27 # Uncomment this to allow local users to log i
```

- /etc/vsftpd.conf 파일 no > yes 수정

```
root@Server:/srv/tftp/pxelinux.cfg# systemctl restart isc-dhcp-server
root@Server:/srv/tftp/pxelinux.cfg# systemctl restart vsftpd
root@Server:/srv/tftp/pxelinux.cfg# systemctl restart tftpd-hpa
root@Server:/srv/tftp/pxelinux.cfg#
root@Server:/srv/tftp/pxelinux.cfg# systemctl enable isc-dhcp-server
Synchronizing state of isc-dhcp-server.service with SysV service script with /lib/systemd/systemd-sysv-i
n stall.
Executing: /lib/systemd/systemd-sysv-install enable isc-dhcp-server
root@Server:/srv/tftp/pxelinux.cfg# systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
root@Server:/srv/tftp/pxelinux.cfg# systemctl enable tftpd-hpa
tftpd-hpa.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable tftpd-hpa
root@Server:/srv/tftp/pxelinux.cfg#
```

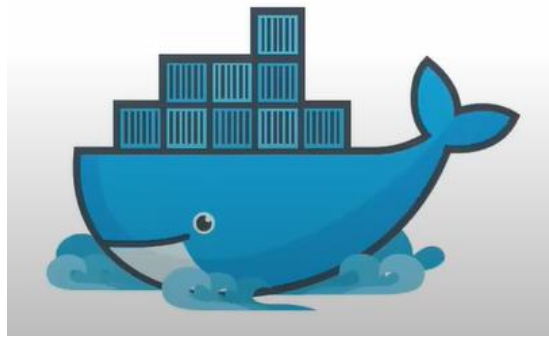
➔ 3 개의 시스템 재시작(적용) 이후 방화벽 비활성화



➔ test 용 장치 생성 후 부팅 (PXE 정상작동 확인)

● 도커의 개념

- 도커 Docker 란 작은 운영체제를 포함한 가상화 기술을 의미
- 가상 컴퓨터는 완전한 OS 를 포함한 독립된 컴퓨터로 간주되며 상당히 무거움
- 도커는 가상 컴퓨터와 거의 비슷한 기능을 갖고 있으면서도 훨씬 가볍게 생성하고 운영
- 도커에서 사용되는 작은 컴퓨터를 컨테이너 Container 라고 부른다



➤ 가상 컴퓨터와 컨테이너의 차이점



```
root@server:~/바탕화면# apt -y install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
패키지 목록을 읽는 중입니다... 완료
```

➔ 도커에 필요한 패키지 5 개 설치

- 설치 중 패키지 curl 의존성 오류 발생
- focal-updates 저장소를 사용 / 사용 후 update
- 문제해결. 정상적으로 설치 완료

```
root@Server1:~/바탕화면# curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| apt-key add -
OK
root@Server1:~/바탕화면#
```

➔ curl 명령어를 사용하여 해당 장소 다운로드

```
ubuntu@Server1:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
기존: 1 http://mirror.kakao.com/ubuntu focal InRelease
받기: 2 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
받기: 3 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [15.5 kB]
기존: 4 http://archive.ubuntu.com/ubuntu focal InRelease
받기: 5 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
받기: 6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]
받기: 7 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [391 kB]
받기: 8 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
내려받기 856 k바이트, 소요시간 3초 (253 k바이트/초)
```

➔ 도커 저장소 추가(다운로드)

```
ubuntu@Server1:~$ sudo apt -y install docker-ce docker-ce-cli containerd.io
패 키 지 목 록 을 읽 는 중 입 니 다 ... 완 료
의 존 성 트 리 를 만 드 는 중 입 니 다
부 태 정 보 를 읽 는 중 있 습 니 다 ... 완 료
```

➔ 도커 설치 완료

```
ubuntu@Server1:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a83e95a41d40c38
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

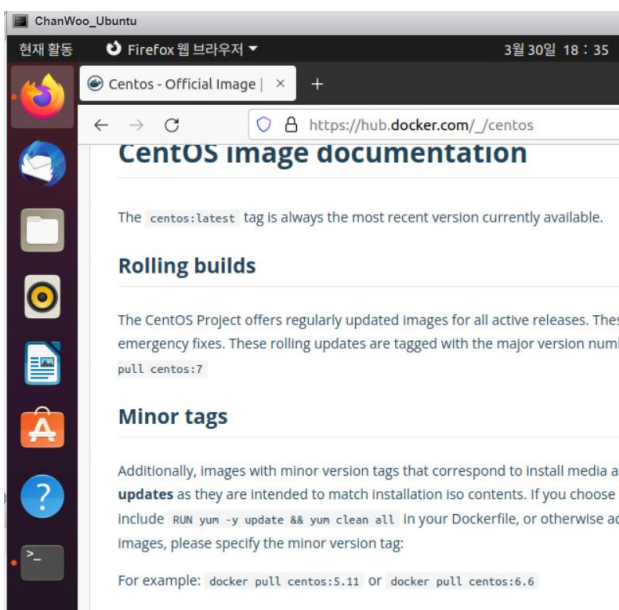
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

ubuntu@Server1:~$
```

➔ docker run hello-world 명령어로 도커 정상 작동 확인



➔ 웹 브라우저에서 hub.docker.com 접속 / centos 검색하여 설치 명령어 확인 가능

```
ubuntu@Server1:~$ sudo docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Extracting 1.671MB/83.52MB
█
```

➔ docker pull 명령어로 다운로드

```
ubuntu@Server1:~$ sudo docker run -it centos bash
[root@6572fb20af8c /]#
[root@6572fb20af8c /]# cat /etc/system-release
CentOS Linux release 8.4.2105
[root@6572fb20af8c /]# █
```

➔ 도커로 centos bash 사용 확인

```
ubuntu@Server1:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5   6 months ago   13.3kB
centos        latest    5d0da3dc9764   6 months ago   231MB
ubuntu@Server1:~$ █
```

➔ docker images 명령어로 현재까지 설치된 컨테이너 확인 가능

- 삭제는 docker rm '컨테이너 id' //컨테이너 id 는 docker container ls -a (컨테이너 자세히 보기) 명령으로 확인 가능

```
root@server:~/바탕화면# docker rm 187b1c32562e
187b1c32562e
root@server:~/바탕화면# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
centos        latest    300e315adb2f   6 weeks ago    209MB
hello-world   latest    bf756fb1ae65   12 months ago   13.3kB
root@server:~/바탕화면# █
```