

Docker 실습 과제

2020039009 차현아

1. 통신을 위해 /etc/hosts/ 파일 수정

```
chahyeona@docker-ubuntu: ~  
127.0.0.1    localhost  
10.100.0.105 docker-ubuntu.example.com    docker-ubuntu  
10.100.0.106 docker-centos.example.com    docker-centos
```

2. ping을 통해 가상머신과 외부 네트워크 연결 확인

```
chahyeona@docker-ubuntu:~$ ping -c 3 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 바이트 (8.8.8.8에서): icmp_seq=1 ttl=117 시간=44.0 ms  
64 바이트 (8.8.8.8에서): icmp_seq=2 ttl=117 시간=42.8 ms  
64 바이트 (8.8.8.8에서): icmp_seq=3 ttl=117 시간=43.5 ms  
  
--- 8.8.8.8 ping 통계 ---  
3 패킷이 전송되었습니다, 3 수신되었습니다, 0% 패킷 손실, 시간 2004ms  
rtt 최소/평균/최대/표준편차 = 42.837/43.450/44.011/0.480 ms  
chahyeona@docker-ubuntu:~$
```

3. text 로그인 모드로 전환 세팅

```
chahyeona@docker-ubuntu:~$ su - root  
암호:  
root@docker-ubuntu:~# systemctl set-default multi-user.target  
Created symlink /etc/systemd/system/default.target → /lib/systemd/system/multi-user.target.
```

4. ssh 데몬 설치

```
root@docker-ubuntu:~# apt-get install -y openssh-server curl vim tree  
패키지 목록을 읽는 중입니다... 완료  
의존성 트리를 만드는 중입니다  
상태 정보를 읽는 중입니다... 완료
```

5. ssh 데몬 설치 확인

```
root@docker-ubuntu:~# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Sat 2022-06-04 14:05:19 KST; 52s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 3931 (sshd)
     Tasks: 1 (limit: 4632)
    Memory: 1.0M
    CGroup: /system.slice/ssh.service
            └─3931 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

6월 04 14:05:19 docker-ubuntu systemd[1]: Starting OpenBSD Secure Shell server:
6월 04 14:05:19 docker-ubuntu sshd[3931]: Server listening on 0.0.0.0 port 22.
6월 04 14:05:19 docker-ubuntu sshd[3931]: Server listening on :: port 22.
6월 04 14:05:19 docker-ubuntu systemd[1]: Started OpenBSD Secure Shell server.
lines 1-15/15 (END)
```

```
chahyeona@docker-ubuntu:~$ exit
로그아웃
Connection to localhost closed.
root@docker-ubuntu:~# exit
로그아웃
```

```
root@docker-ubuntu:~# ssh chahyeona@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:IfQTcGVFtg5rLXuWHCJeXegt6H2v1rcxIe9kjRah2tA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yse
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
chahyeona@localhost:~$
```

6. xshell을 사용하여 가상머신과 연결(포트 포워딩 과정 포함)

일반

이름(N): chahyeona

프로토콜(P): SSH

호스트(H): 127.0.0.1

포트 번호(O): 105

설명(D):

사용자 이름(U): chahyeona

암호(P):

방법(M): ☒ Password ☐ Public Key ☐ Keyboard Interactive ☐ GSSAPI ☐ PKCS11 ☐ CAPI

설정(S)...

위로(U)

아래로(D)

이름	프로토콜	호스트 IP	호스트 포트	게스트 IP	게스트 포트
docker1	TCP	127.0.0.1	105	10.100.0.105	22
docker2	TCP	127.0.0.1	106	10.100.0.106	22

7. xshell을 사용하여 우분투 연결 접속 완료

```
Xshell 7 (Build 0100)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 127.0.0.1:105...
Connection established.
To escape to local shell, press 'Ctrl+Alt+'.

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-44-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

185 updates can be applied immediately.
132 of these updates are standard security updates.
추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요 .

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Jun  4 14:07:00 2022 from 127.0.0.1
/usr/bin/xauth: file /home/chahyeona/.Xauthority does not exist
chahyeona@docker-ubuntu:~$
```

```
Xshell 7 (Build 0100)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 127.0.0.1:105...
Connection established.
To escape to local shell, press 'Ctrl+Alt+'.

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-44-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

185 updates can be applied immediately.
132 of these updates are standard security updates.
추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요 .

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Jun  4 14:07:00 2022 from 127.0.0.1
/usr/bin/xauth: file /home/chahyeona/.Xauthority does not exist
chahyeona@docker-ubuntu:~$ su -
root@docker-ubuntu:~# exit
로그아웃
chahyeona@docker-ubuntu:~$
```

<우분투와 마찬가지로 Centos와 xshell 연결하기>

8. centos도 동일하게 진행(/etc/hostname, /etc/hosts)

```
root@docker-centos:~  
File Edit View Search Terminal Help  
docker-centos.example.com  
~  
~
```

```
root@docker-centos:~  
File Edit View Search Terminal Help  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
10.100.0.105 docker-ubuntu.example.com docker-ubuntu  
10.100.0.106 docker-centos.example.com docker-centos  
~  
~
```

9. ping을 통해 가상머신과 외부 네트워크 연결 확인

```
[root@docker-centos ~]# vi /etc/hosts  
[root@docker-centos ~]# ping 8.8.8.8 -c 3  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=102 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=46.9 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=47.9 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2004ms  
rtt min/avg/max/mdev = 46.924/65.923/102.884/26.138 ms  
[root@docker-centos ~]#
```

10. 텍스트 로그인 모드 전환 세팅

```
[root@docker-centos ~]# systemctl set-default multi-user.target  
Removed symlink /etc/systemd/system/default.target.  
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.  
[root@docker-centos ~]#
```

11. sshd 동작 확인

```
[root@docker-centos ~]# systemctl status sshd  
● sshd.service - OpenSSH server daemon  
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2022-06-04 15:05:28 KST; 14min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Main PID: 1198 (sshd)  
    CGroup: /system.slice/ssh.service  
            └─1198 /usr/sbin/sshd -D  
  
Jun 04 15:05:27 docker-centos.example.com systemd[1]: Starting OpenSSH server...  
Jun 04 15:05:28 docker-centos.example.com sshd[1198]: Server listening on 0.0.0.0 port 22.  
Jun 04 15:05:28 docker-centos.example.com sshd[1198]: Server listening on :: port 22.  
Jun 04 15:05:28 docker-centos.example.com systemd[1]: Started OpenSSH server daemon.  
Hint: Some lines were ellipsized, use -l to show in full.
```

12. tree설치

```
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : tree-1.6.0-10.el7.x86_64 1/1  
  Verifying   : tree-1.6.0-10.el7.x86_64 1/1  
  
Installed:  
  tree.x86_64 0:1.6.0-10.el7  
  
Complete!  
[root@docker-centos ~]#
```

13. xshell에서 centos에 접속하기 위해 세팅

<p>사용자 이름(U):</p> <p>암호(P):</p> <p>방법(M):</p>	<input type="text" value="chahyeona"/> <div>● ● ● ● ● ● ● ●</div> <div> <input checked="" type="checkbox"/> Password <input type="checkbox"/> Public Key <input type="checkbox"/> Keyboard Interactive <input type="checkbox"/> GSSAPI <input type="checkbox"/> PKCS11 <input type="checkbox"/> CAPI </div>	<p>설정(S)...</p> <p>위로(U)</p> <p>아래로(D)</p>
---	---	--

<p>이름(N):</p> <p>프로토콜(P):</p> <p>호스트(H):</p> <p>포트 번호(O):</p> <p>설명(D):</p>	<input type="text" value="docker-centos"/> <input type="text" value="SSH"/> <input type="text" value="127.0.0.1"/> <input type="text" value="106"/> <div></div>	
---	---	---

14. 연결 성공 확인

```
1 docker-centos x +
Xshell 7 (Build 0109)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 127.0.0.1:106...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Last login: Sat Jun 4 15:06:18 2022
/usr/bin/xauth: file /home/chaheyona/.Xauthority does not exist
[chaheyona@docker-centos ~]$
```

```
1 docker-centos x +
[chahyeona@docker-centos ~]$ whoami
chahyeona
[chahyeona@docker-centos ~]$ hostname
docker-centos.example.com
[chahyeona@docker-centos ~]$
```

<우분투에서 도커 설치하기>

15. 우분투에서 도커를 설치하기 위해 필요한 요구 프로그램 설치

```
chahyeona@docker-ubuntu:~$ sudo apt-get install \
> ca-certificates \
> curl \
> gnupg \
> lsb-release
```

패키지 목록을 읽는 중입니다 ... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다 ... 완료
패키지 lsb-release는 이미 최신 버전입니다 (11.1.0ubuntu1)

16. 도커 인증서 저장

```
chahyeona@docker-ubuntu:~$ sudo mkdir -p /etc/apt/keyrings
chahyeona@docker-ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```


17. url 등록

```
chahyeona@docker-ubuntu:~$ echo \  
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/do  
cker.gpg] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.l  
ist > /dev/null
```

18. 도커엔진 설치

```
N. See apt-secure(8) manpage for repository creation and user configuration details.  
chahyeona@docker-ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io d  
ocker-compose-plugin
```

ssh://chahyeona@127.0.0.1 SSH2 xterm 87x27 27,21 2 세션 CAP NUM

19. 도커 실행 확인

```
chahyeona@docker-ubuntu:~$ sudo docker version  
Client: Docker Engine - Community  
Version: 20.10.16  
API version: 1.41  
Go version: go1.17.10  
Git commit: aa7e414  
Built: Thu May 12 09:17:23 2022  
OS/Arch: linux/amd64  
Context: default  
Experimental: true  
  
Server: Docker Engine - Community  
Engine:  
Version: 20.10.16  
API version: 1.41 (minimum version 1.12)  
Go version: go1.17.10  
Git commit: f756502  
Built: Thu May 12 09:15:28 2022  
OS/Arch: linux/amd64  
Experimental: false  
containerd:  
Version: 1.6.4  
GitCommit: 212e8b6fa2f44b9c21b2798135fc6fb7c53efc16
```

<centos에서 도커 설치하기>

20. centos에서 도커를 설치하기 위해 필요한 요구 프로그램 설치

```
nothing to do  
[root@docker-centos ~]# yum-config-manager \  
> --add-repo \  
> https://download.docker.com/linux/centos/docker-ce.repo
```

21. 도커엔진 설치

```
repo saved to /etc/yum.repos.d/docker-ce.repo  
[root@docker-centos ~]# yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y
```

22. 도커 동작 확인

```
[root@docker-centos ~]# systemctl start docker
[root@docker-centos ~]# docker version
Client: Docker Engine - Community
Version:      20.10.16
API version:  1.41
Go version:   go1.17.10
Git commit:   aa7e414
Built:        Thu May 12 09:19:45 2022
OS/Arch:      linux/amd64
```

23. 우분투에서 내 계정에 도커 관리 권한 부여(centos와 명령어 동일)

```
Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Sat Jun  4 15:32:40 2022 from 10.100.0.2
chahyeona@docker-ubuntu:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
chahyeona@docker-ubuntu:~$
```

24. centos에서 내 계정에 도커 관리 권한 부여

```
[root@docker-centos ~]# usermod -a -G docker chahyeona
[root@docker-centos ~]# su - chahyeona
Last login: Sat Jun  4 15:33:32 KST 2022 from 10.100.0.2 on pts/0
[chahyeona@docker-centos ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[chahyeona@docker-centos ~]$
```

-- 여기까지 3.2 실습 동영상을 수행하기 위한 세팅 완료 --

<3.2 도커 컨테이너 살펴보기 :실습편>

1. 도커데몬이 동작중인지 확인(루트계정과 유저 계정에서 둘다 확인), 세션을 복제해서 둘다 열어놓은 상태

```
chahyeona@docker-ubuntu:~$ docker version
Client: Docker Engine - Community
Version: 20.10.16
API version: 1.41
Go version: go1.17.10
Git commit: aa7e414
Built: Thu May 12 09:17:23 2022
OS/Arch: linux/amd64
Context: default
Experimental: true

Server: Docker Engine - Community
Engine:
```

```
root@docker-ubuntu:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; v
   Active: active (running) since Sat 2022-06-04 22:18:14 KST; 3m
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 721 (dockerd)
      Tasks: 8
     Memory: 106.4M
    CGroup: /system.slice/docker.service
            └─721 /usr/bin/dockerd -H fd:// --containerd=/run/cont

6월 04 22:18:13 docker-ubuntu.example.com dockerd[721]: time="2022
```

2. 써치 명령으로 내가 원하는 컨테이너가 도커 허브에 존재하는지 찾기

```
chahyeona@docker-ubuntu:~$ docker search nginx
NAME                                DESCRIPTION                                STARS     OFFICIAL   AUTOMATED
userxy2015/nginx                   nginx                                      15
merpso/nginx-plus-k8s-ingress      NGINX+ kubernetes-ingress                2
jhuiting/nginx                     nginx                                      1
ludwringlicien/nginx-php7.1        nginx-php7.1                              1         [OK]
snehagupta83/nginx                 Test setup                                 0
alama1hoda/nginx                   nginx                                      0
hamropatrorrepo/nginx-reverse-proxy nginx with config for a static js app      0
bharath2012/nginx                  nginx rate limit examples                 0
afzal/nginx_rate_limit              nginx docker image with ubuntu            0
silvade/nginx-static               nginx server                              0
sushanth53/nginx_server             nginx server                              0
ssvreddy/nginx                     nginx                                      0
agualbbus/nginx-php-proxy          Already set listen port of nginx to $PORT o... 0
bguruprasad/nginx-print-hostname   nginx-print-hostname                     0
sitek/nginx-vsftpd                 nginx-vsftpd                              0         [OK]
sanghaikshay/nginx_https_webserver nginx_https_webserver                     0
jenorish/nginx                     nginx                                      0
venkateshs2006/nginx-php-fpm-laravel nginx                                      0
covenant/nginx                     nginx                                      0
theburi/nginx                       tde test                                 0         [OK]
starlkj/nginx                       nginx                                      0
sunlitweb/nginx                     Image for static app without root user    0
motork/nginx-unprivileged-static-app nginx                                      0
chahyeona@docker-ubuntu:~$
```

3. 컨테이너 이미지 다운로드

```
chahyeona@docker-ubuntu:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
42c077c10790: Pull complete
62c70f376f6a: Pull complete
915cc9bd79c2: Pull complete
75a963e94de0: Pull complete
7b1fab684d70: Pull complete
db24d06d5af4: Pull complete
Digest: sha256:2bcabc23b45489fb0885d69a06ba1d648aeda973fae7bb981ba
bb884165e514
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
chahyeona@docker-ubuntu:~$
```

4. 루트계정으로 overlay2 폴더에서 이미지 레이어가 잘 들어온걸 확인 가능
- nginx는 6개의 레이어로 구성된것을 알 수 있음 + 사용자에서 확인도 가능

```
root@docker-ubuntu:/var/lib/docker/overlay2# ls -l
합계 28
drwx--x--- 3 root root 4096 6월 4 22:29 079ffcc73622d58d31fa9dec8d8d12b899082c0
7f15fad58a706a469c065fe16
drwx--x--- 4 root root 4096 6월 4 22:29 568e0c2893cd246b18474eb6870e59396e38efd
2aae34613007b03adeacaf0c1
drwx--x--- 4 root root 4096 6월 4 22:29 af425693c56f9796ae21fd09e0b2beb8c376272
9895e1cd9acbd0c04165cc223
drwx--x--- 4 root root 4096 6월 4 22:29 afbcfcb01043bbe5b2952c0df0228971a35e00c
db7e518a6c09b7f7915b20591
drwx--x--- 4 root root 4096 6월 4 22:29 d6c3c8c2caf35d0513298288562c5a3cec07708
e66c2ab68c3b29dcfe7945f55
drwx--x--- 4 root root 4096 6월 4 22:29 f633f3590767b7f4b33d3eb23fdbea346789d54
3a75bd97cb1f31ae652cecea6
drwx----- 2 root root 4096 6월 4 22:29 l
root@docker-ubuntu:/var/lib/docker/overlay2#
```

```
chahyeona@docker-ubuntu:~$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 0e901e68141f 7 days ago 142MB
chahyeona@docker-ubuntu:~$
```

5. 컨테이너 실행하고 확인해보기
- nginx 컨테이너가 실행, 고유한 컨테이너 아이디를 알 수 있음

```
chahyeona@docker-ubuntu:~$ docker run --name web -d -p 80:80 nginx
28bb764802b6b4fc4e251a255cfa0c1bac3131a8e6970401c23d0e7fe8977e70
chahyeona@docker-ubuntu:~$
```

6. 현재 동작중인 도커 컨테이너 출력

```
chahyeona@docker-ubuntu:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED NAMES
STATUS PORTS
28bb764802b6 nginx "/docker-entrypoint..." About a minute ago
o Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp web
chahyeona@docker-ubuntu:~$
```

7. 웹 페이지 출력을 확인
- 다른 컴퓨터와 완전히 분리된 환경에서 컨테이너 모양을 가지고 어플리케이션 동작을 하게 해줌

```
chahyeona@docker-ubuntu:~$ curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: chahyeona@docker-ubuntu:~$ docker image ls
</style>
REPOSITORY TAG IMAGE ID CREATED SIZE
</head>
ngi|chahyeona@docker-ubuntu:~$ docker image ls
<body>
cha|REPOSITORY TAG IMAGE ID CREATED SIZE
<h1>Welcome to ,nginx latest 0e901e68141f 7 days ago 142MB
<p>If you see t|chahyeona@docker-ubuntu:~$
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
```


8. web 컨테이너 사용 중지

```
chahyeona@docker-ubuntu:~$ docker stop web
web
chahyeona@docker-ubuntu:~$ curl localhost:80
curl: (7) Failed to connect to localhost port 80: 연결이 거부됨
chahyeona@docker-ubuntu:~$
```

9. 컨테이너 삭제

```
chahyeona@docker-ubuntu:~$ docker rm web
web
chahyeona@docker-ubuntu:~$
```

10. 컨테이너 이미지 삭제

```
chahyeona@docker-ubuntu:~$ docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:2bcabc23b45489fb0885d69a06ba1d648aeda973fae7bb981bafbb884165e514
Deleted: sha256:0e901e68141fd02f237cf63eb842529f8a9500636a9419e3cf4fb986b8fe3d5d
Deleted: sha256:1e877fbb1acf761377390ab38bbad050a1d5296f1b4f51878c2695d4ecdb98c62
Deleted: sha256:834e54d50f731515065370d1c15f0ed47d2f7b6a7b0452646db80f14ace9b8de
Deleted: sha256:d28ca7ee17ff94497071d5c075b4099a4f2c950a3471fc49bdf9876227970b24
Deleted: sha256:096f97ba95539883af393732efac02acdd0e2ae587a5479d97065b64b4eded8c
Deleted: sha256:de7e3b2a7430261fde88313fbf784a63c2229ce369b9116053786845c39058d5
Deleted: sha256:ad6562704f3759fb50f0d3de5f80a38f65a85e709b77fd24491253990f30b6be
chahyeona@docker-ubuntu:~$
```

11. 루트 계정에서 컨테이너 이미지 삭제 확인

```
root@docker-ubuntu:/var/lib/docker/overlay2# ls -l
합계 4
drwx----- 2 root root 4096 6월 4 22:43 l
root@docker-ubuntu:/var/lib/docker/overlay2#
```