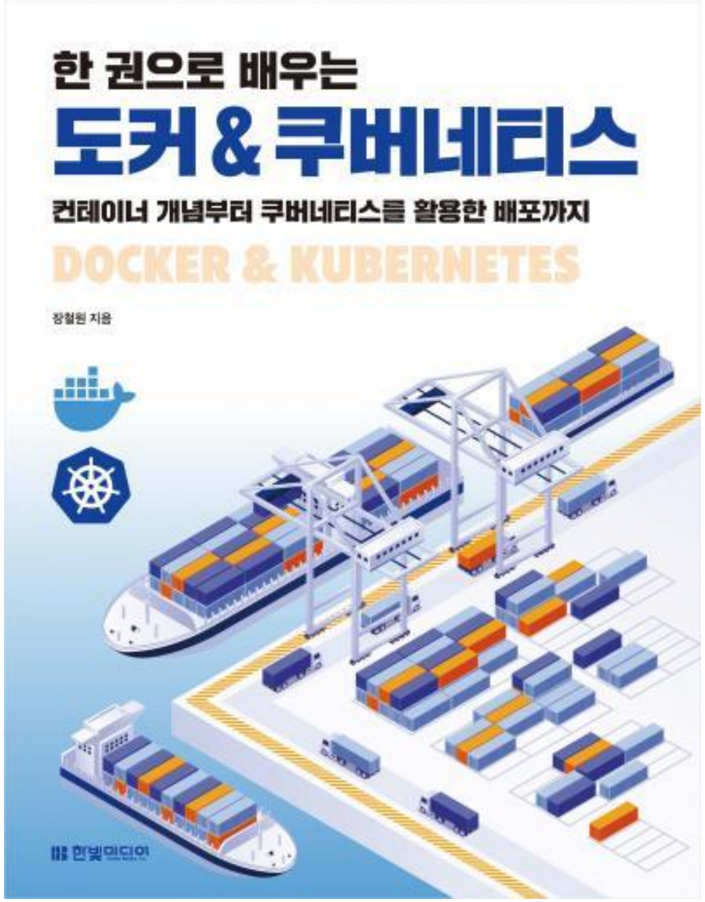
Basics of docker

Autumn 2024

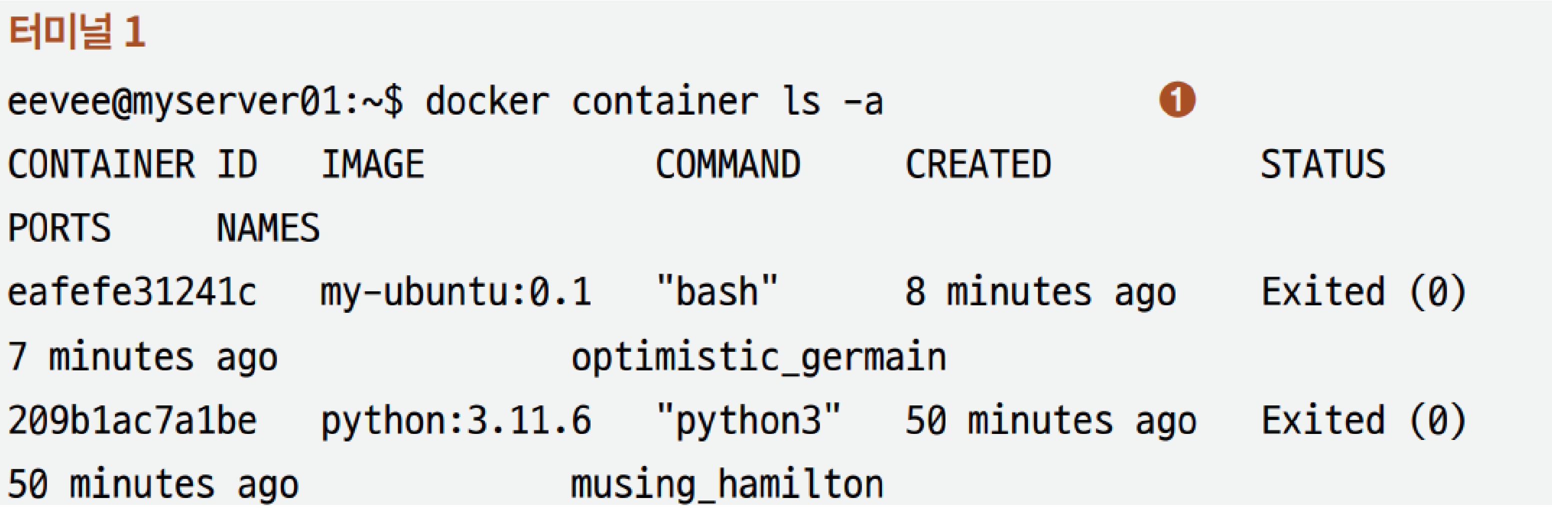
AI융합학과
Seongbok Baik
sbbaik@dju.ac.kr

00 Textbook



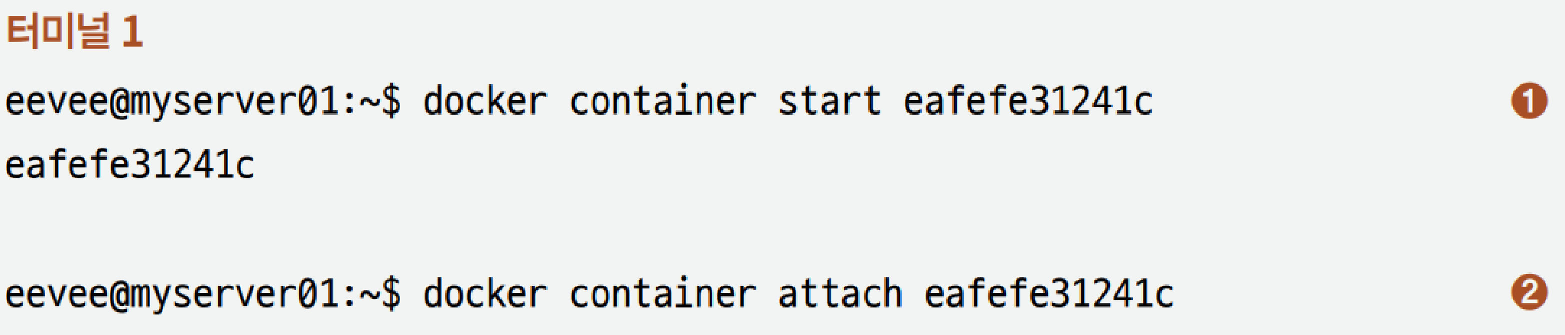
장철원 소프트웨어공학자

01 도커 컨테이너 내부에서 네트워크 구성 확인-1



터미널 1

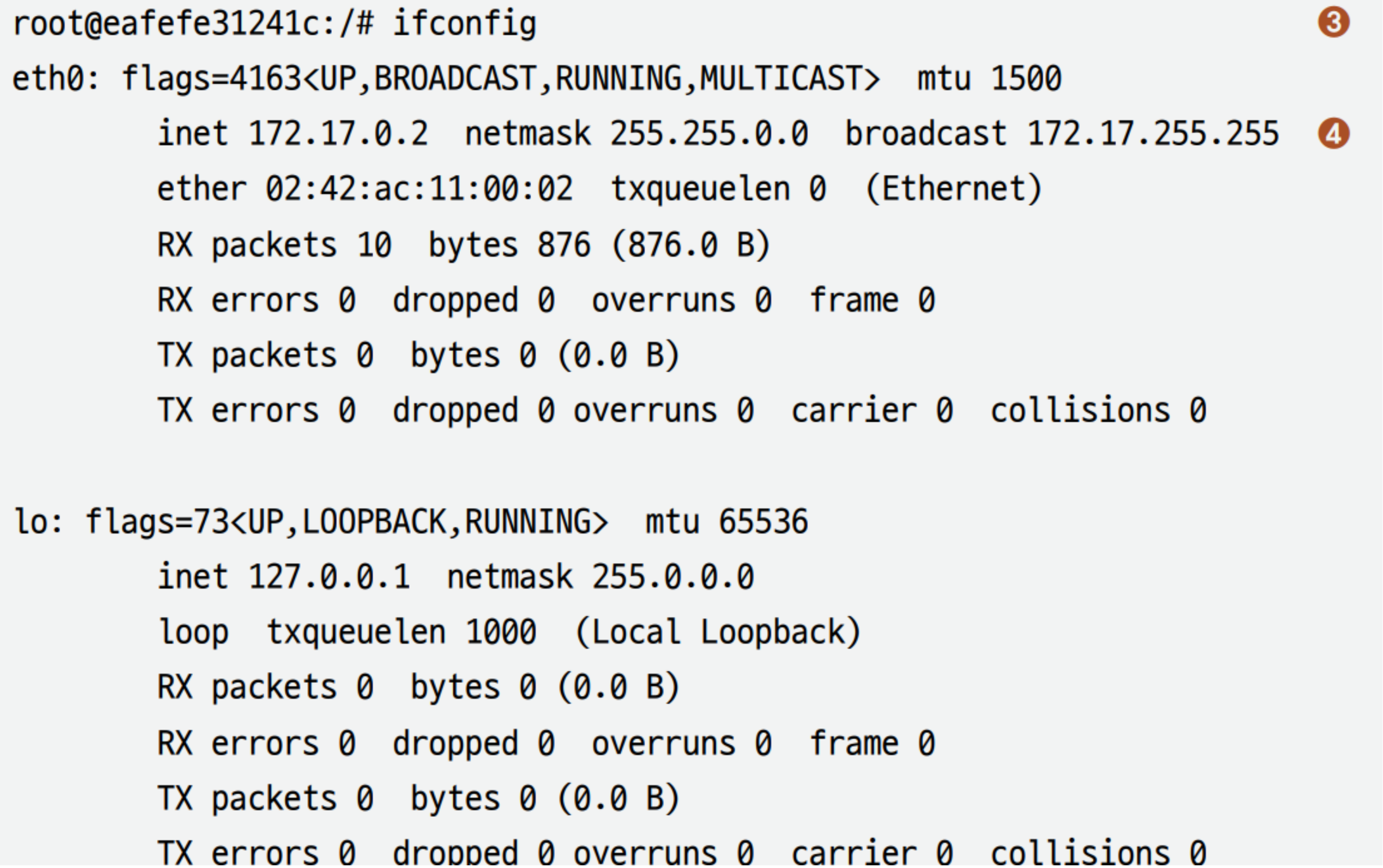
```
eevee@myserver01:~$ docker container ls -a                 ❶
CONTAINER ID   IMAGE           COMMAND     CREATED          STATUS
PORTS     NAMES
eafefe31241c   my-ubuntu:0.1   "bash"      8 minutes ago    Exited (0)
7 minutes ago              optimistic_germain
209b1ac7a1be   python:3.11.6   "python3"   50 minutes ago   Exited (0)
50 minutes ago             musing_hamilton
```



터미널 1

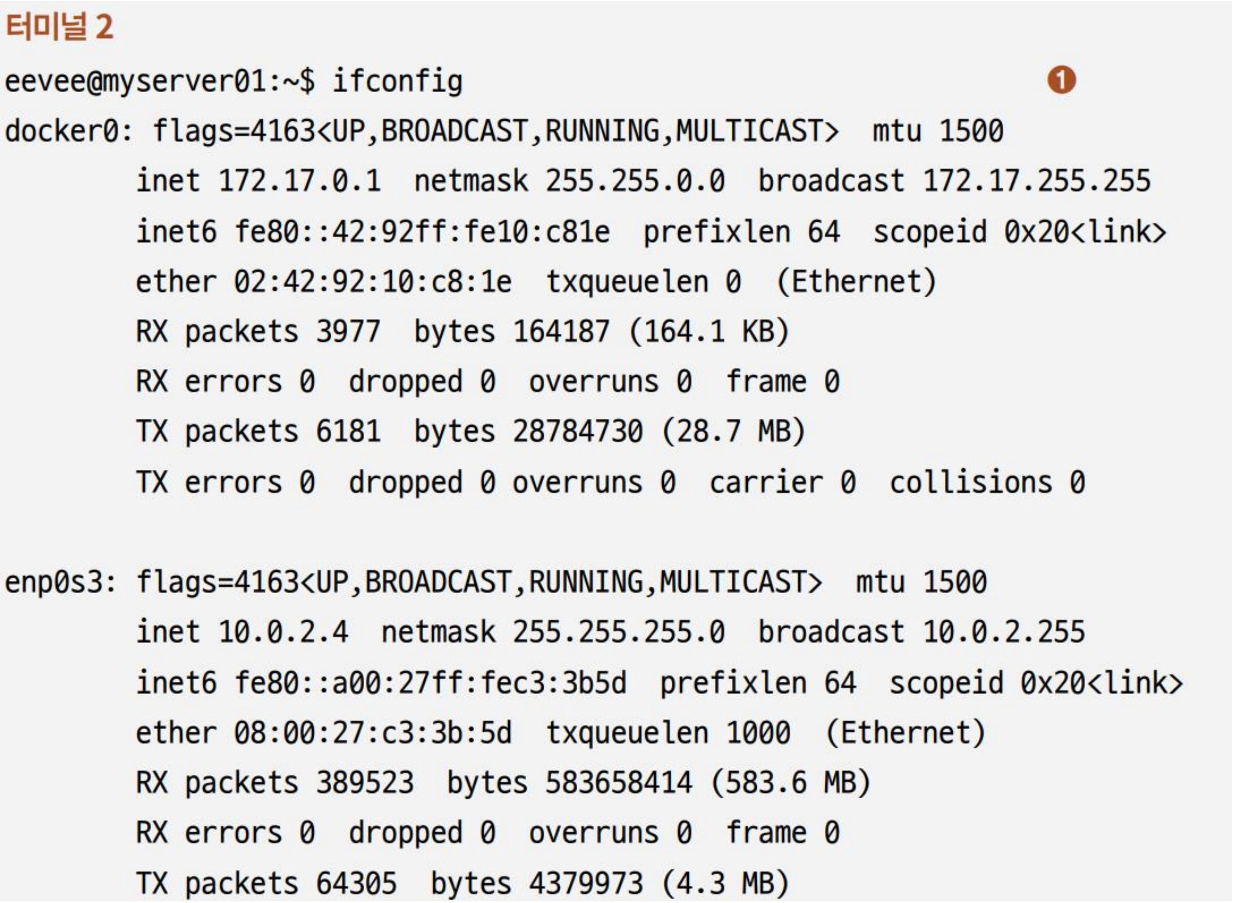
```
eevee@myserver01:~$ docker container start eafefe31241c                  ❶
eafefe31241c

eevee@myserver01:~$ docker container attach eafefe31241c                 ❷
```

02 도커 컨테이너 내부에서 네트워크 구성 확인-2



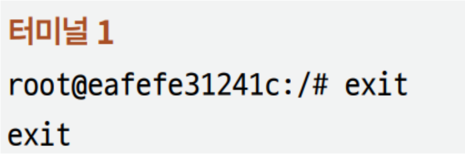
```
root@eafefe31241c:/# ifconfig                                                    ❸
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255  ❹
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 10  bytes 876 (876.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

03 도커 컨테이너 외부에서 네트워크 구성 확인



터미널 2

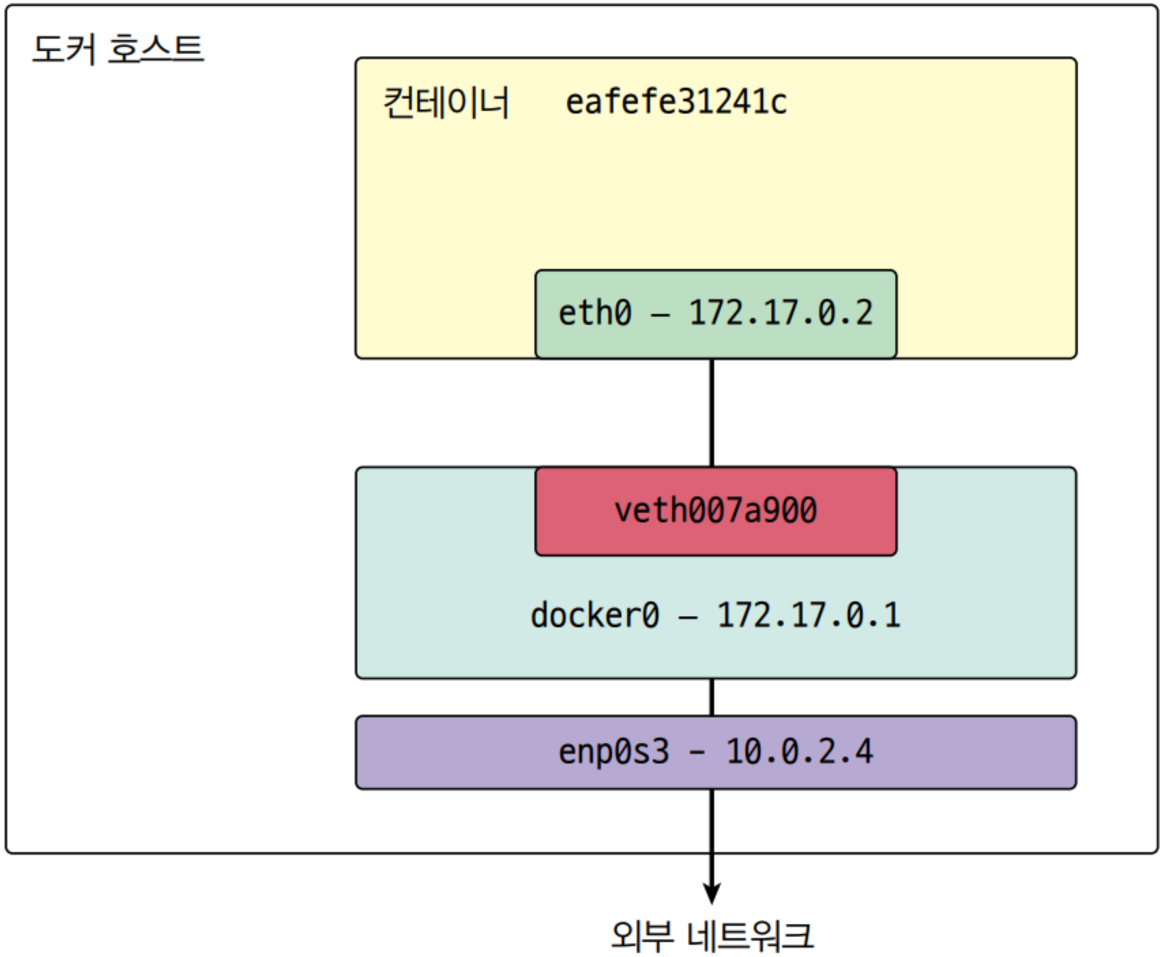
```
eevee@myserver01:~$ ifconfig                                         ❶
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:92ff:fe10:c81e  prefixlen 64  scopeid 0x20<link>
        ether 02:42:92:10:c8:1e  txqueuelen 0  (Ethernet)
        RX packets 3977  bytes 164187 (164.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6181  bytes 28784730 (28.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fec3:3b5d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c3:3b:5d  txqueuelen 1000  (Ethernet)
        RX packets 389523  bytes 583658414 (583.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64305  bytes 4379973 (4.3 MB)
```



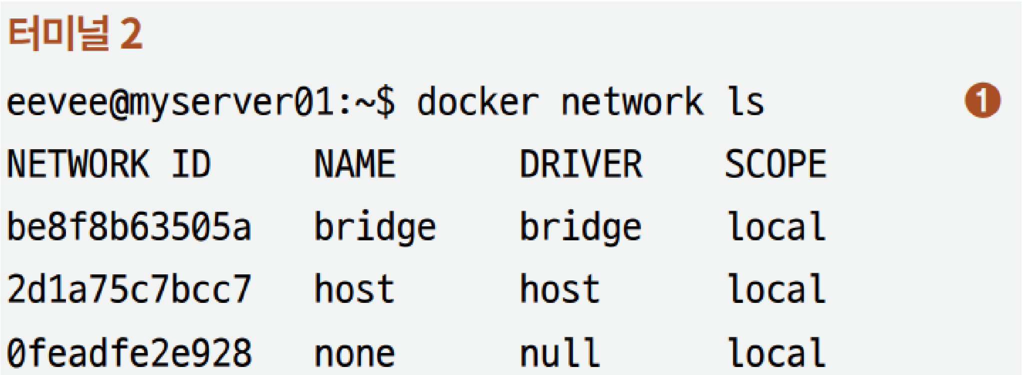
터미널 1

```
root@eafefe31241c:/# exit
exit
```

04 도커 컨테이너 네트워크 연결 구조



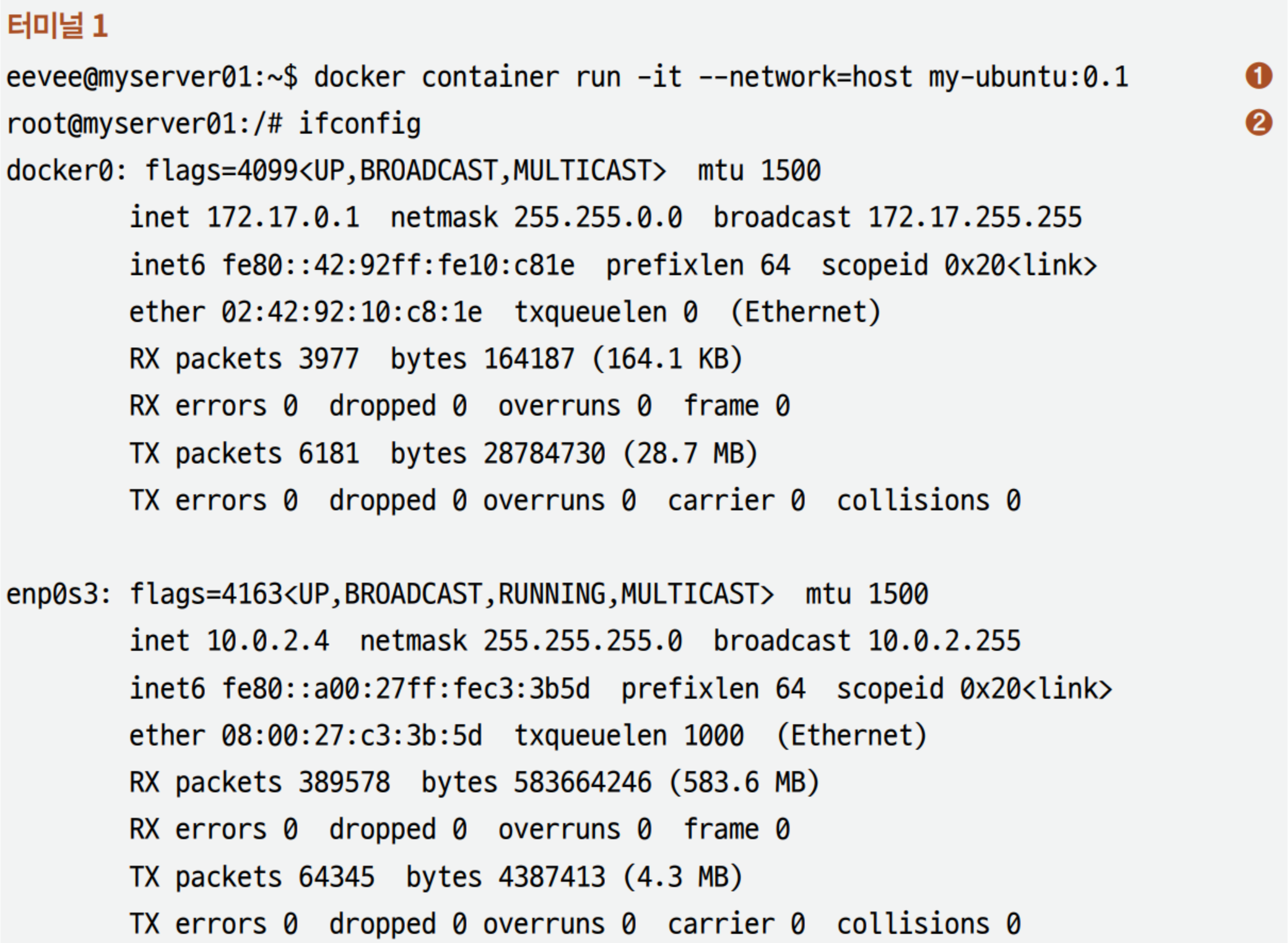
05 도커 네트워크 확인


터미널 2

```
eevee@myserver01:~$ docker network ls          ❶
NETWORK ID     NAME      DRIVER    SCOPE
be8f8b63505a   bridge    bridge    local
2d1a75c7bcc7   host      host      local
0feadfe2e928   none      null      local
```

- bridge 드라이버: 컨테이너 생성시 제공되는 기본 드라이버. 각 컨테이너는 각자의 네트워크 인터페이스 보유. 도커 호스트의 docker0와 바인딩됨.
- host 드라이버: 컨테이너 자체적으로 네트워크 인터페이스를 가지지 않고 호스트 컴퓨터의 네트워크 인터페이스를 공유.
- none 드라이버: 컨테이너가 네트워크 인터페이스 갖지 않음. 외부 통신 불가능

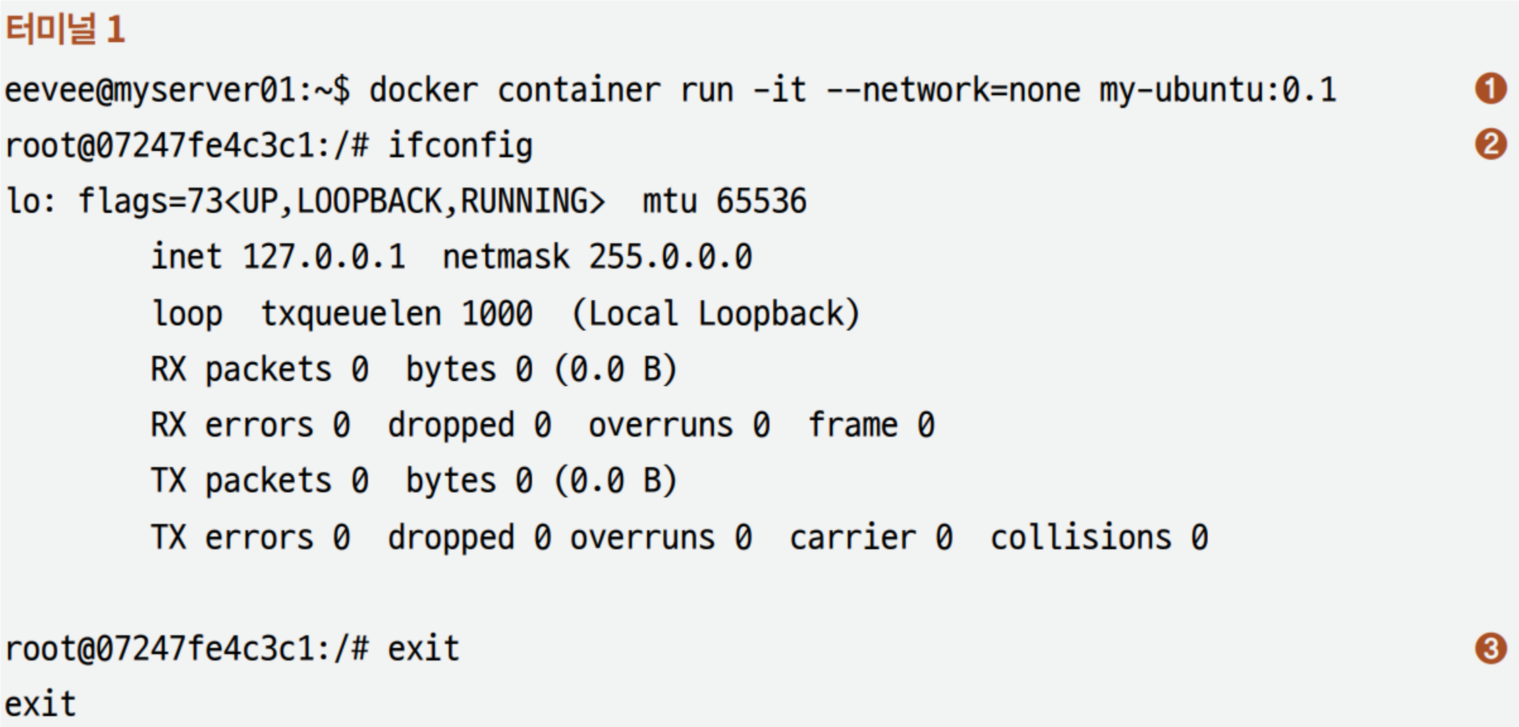
06 호스트 드라이버를 갖는 컨테이너 생성



터미널 1

```
eevee@myserver01:~$ docker container run -it --network=host my-ubuntu:0.1        ❶
root@myserver01:/# ifconfig                                                       ❷
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:92ff:fe10:c81e  prefixlen 64  scopeid 0x20<link>
        ether 02:42:92:10:c8:1e  txqueuelen 0  (Ethernet)
        RX packets 3977  bytes 164187 (164.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6181  bytes 28784730 (28.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fec3:3b5d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c3:3b:5d  txqueuelen 1000  (Ethernet)
        RX packets 389578  bytes 583664246 (583.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64345  bytes 4387413 (4.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

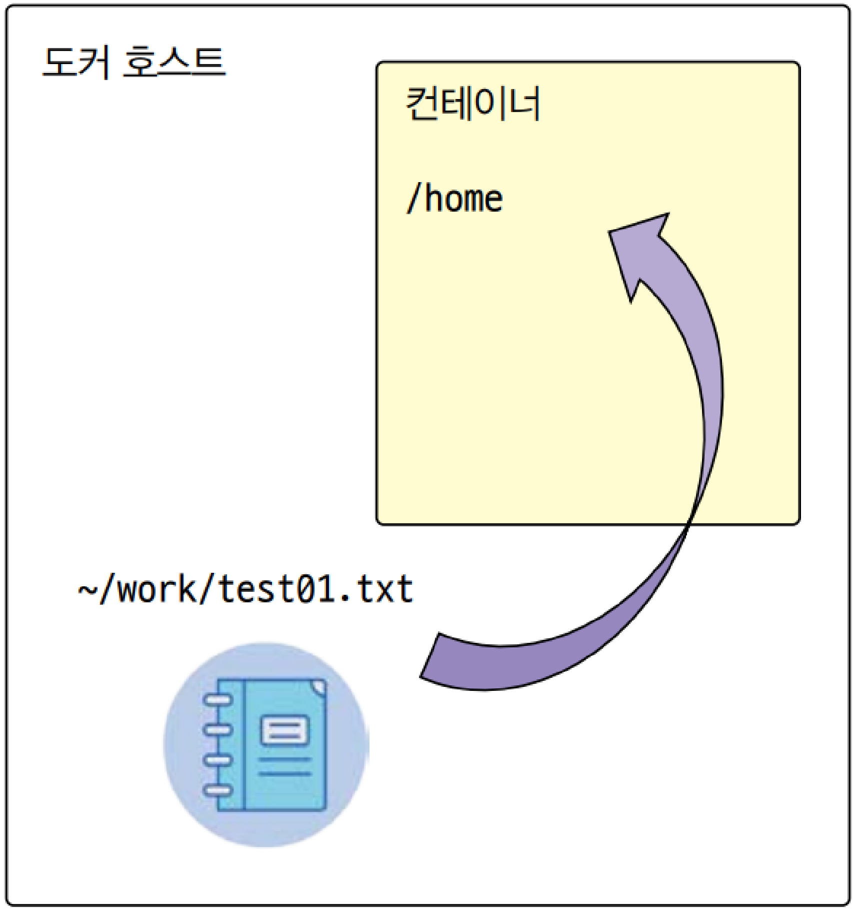
07 none 드라이버를 갖는 컨테이너 생성

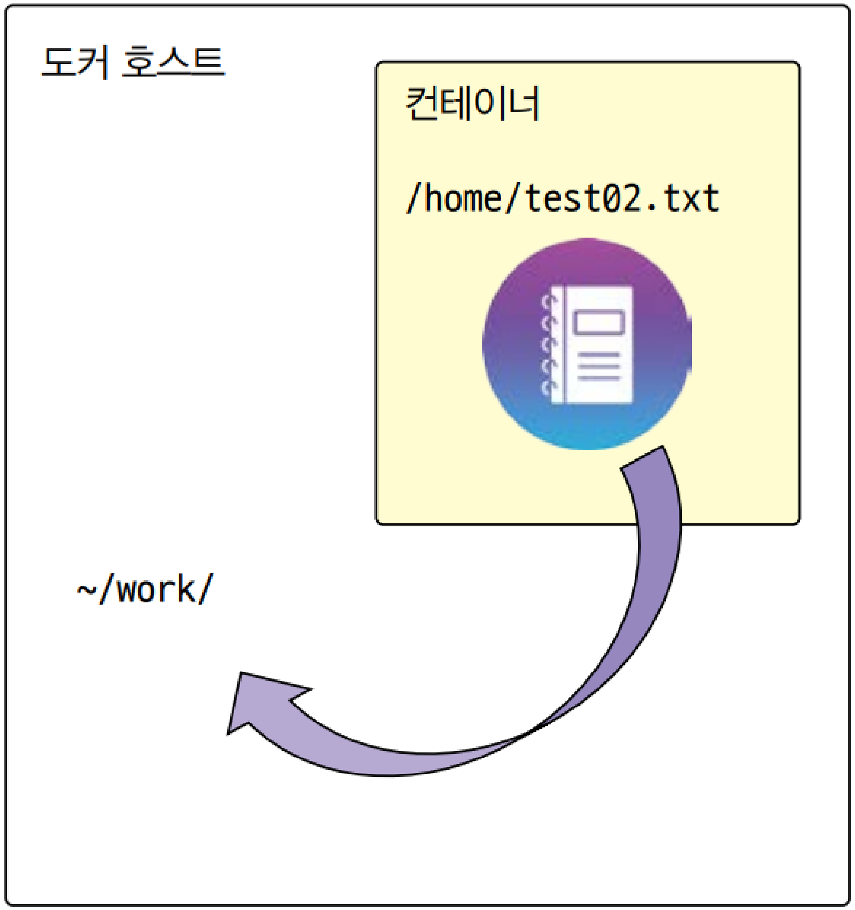


터미널 1

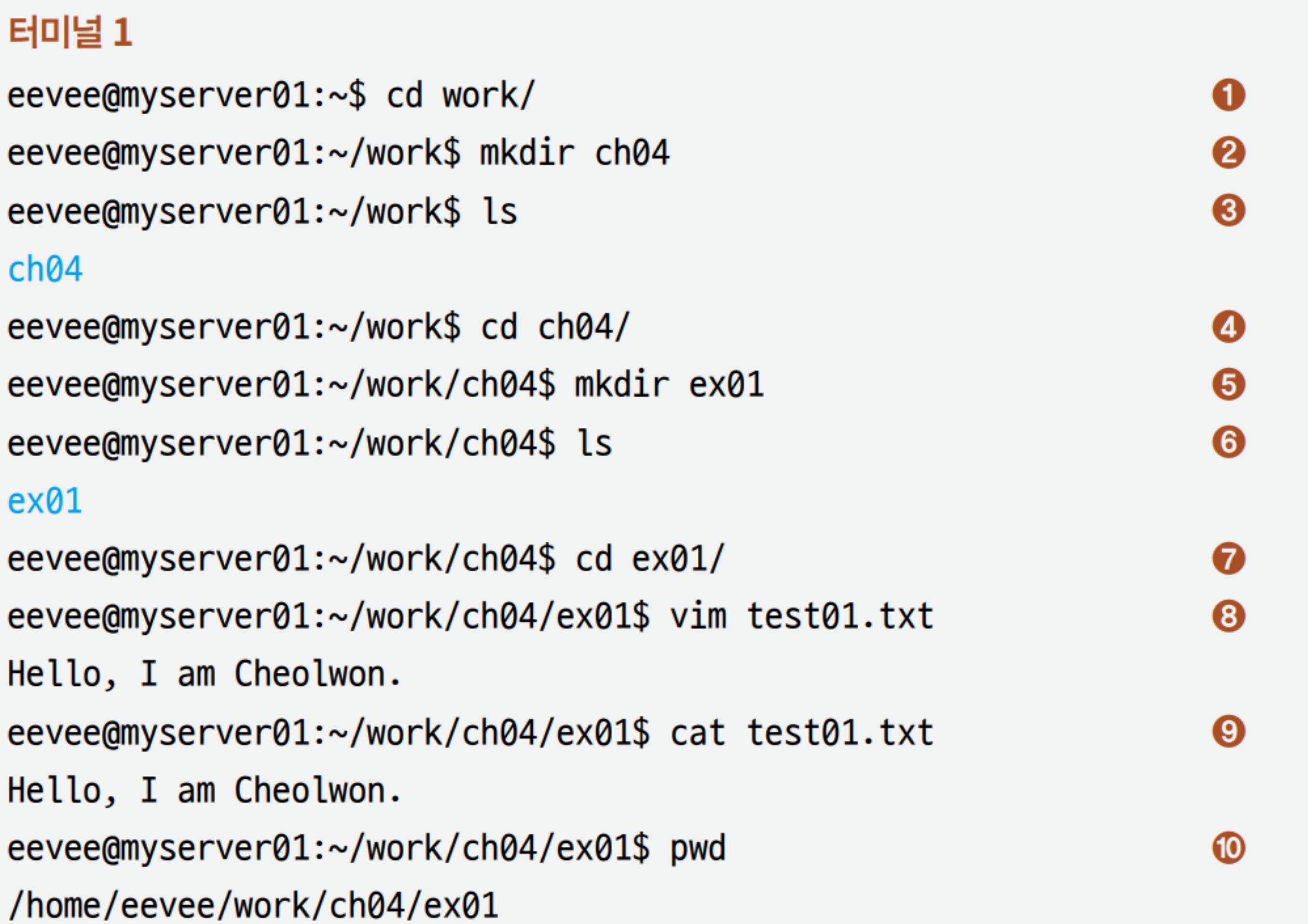
```
eevee@myserver01:~$ docker container run -it --network=none my-ubuntu:0.1       ❶
root@07247fe4c3c1:/# ifconfig                                                   ❷
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@07247fe4c3c1:/# exit                                                       ❸
exit
```

08 호스트에서 컨테이너로 파일 전송





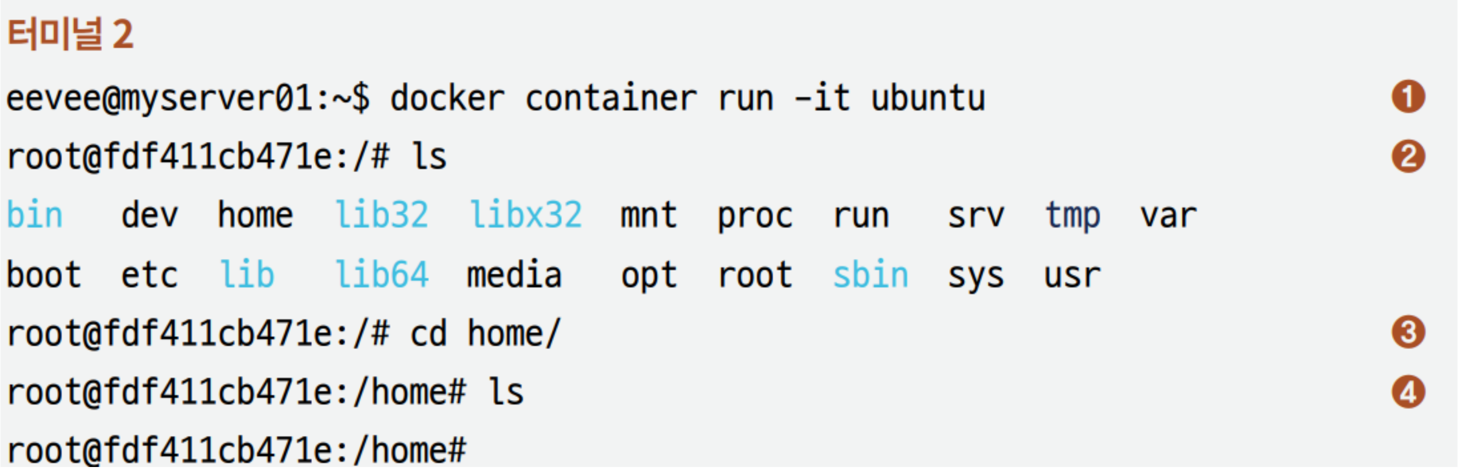
09 호스트 컴퓨터에 파일 준비



터미널 1

```
eevee@myserver01:~$ cd work/                                    ❶
eevee@myserver01:~/work$ mkdir ch04                             ❷
eevee@myserver01:~/work$ ls                                     ❸
ch04
eevee@myserver01:~/work$ cd ch04/                               ❹
eevee@myserver01:~/work/ch04$ mkdir ex01                        ❺
eevee@myserver01:~/work/ch04$ ls                                ❻
ex01
eevee@myserver01:~/work/ch04$ cd ex01/                          ❼
eevee@myserver01:~/work/ch04/ex01$ vim test01.txt               ❽
Hello, I am Cheolwon.
eevee@myserver01:~/work/ch04/ex01$ cat test01.txt               ❾
Hello, I am Cheolwon.
eevee@myserver01:~/work/ch04/ex01$ pwd                          ❿
/home/eevee/work/ch04/ex01
```

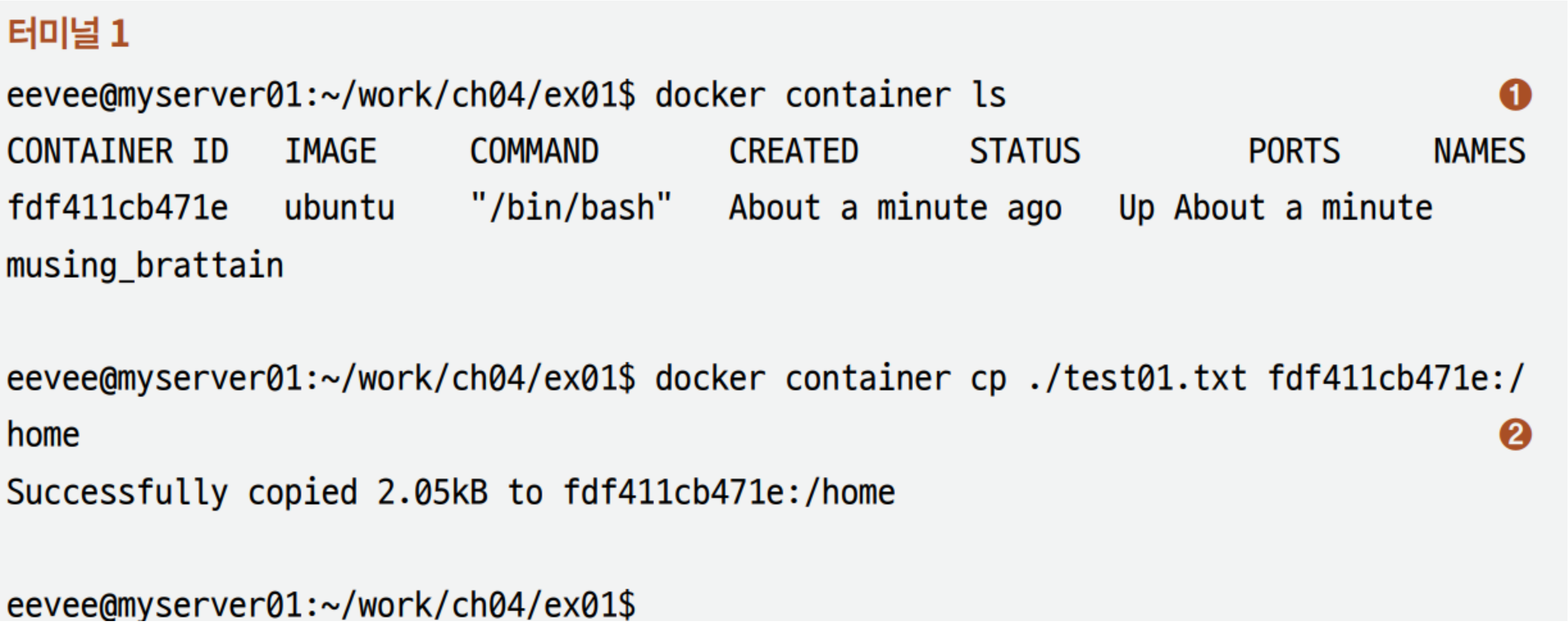
10 컨테이너에서 디렉토리 확인



터미널 2

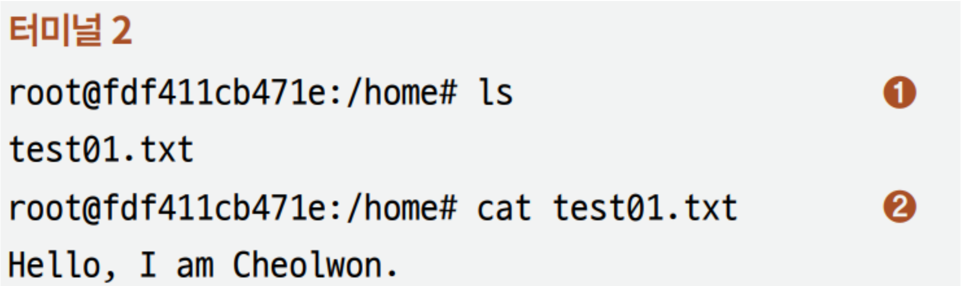
```
eevee@myserver01:~$ docker container run -it ubuntu                       ❶
root@fdf411cb471e:/# ls                                                   ❷
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
root@fdf411cb471e:/# cd home/                                             ❸
root@fdf411cb471e:/home# ls                                               ❹
root@fdf411cb471e:/home#
```

11 호스트에서 컨테이너로 파일 복사



터미널 1

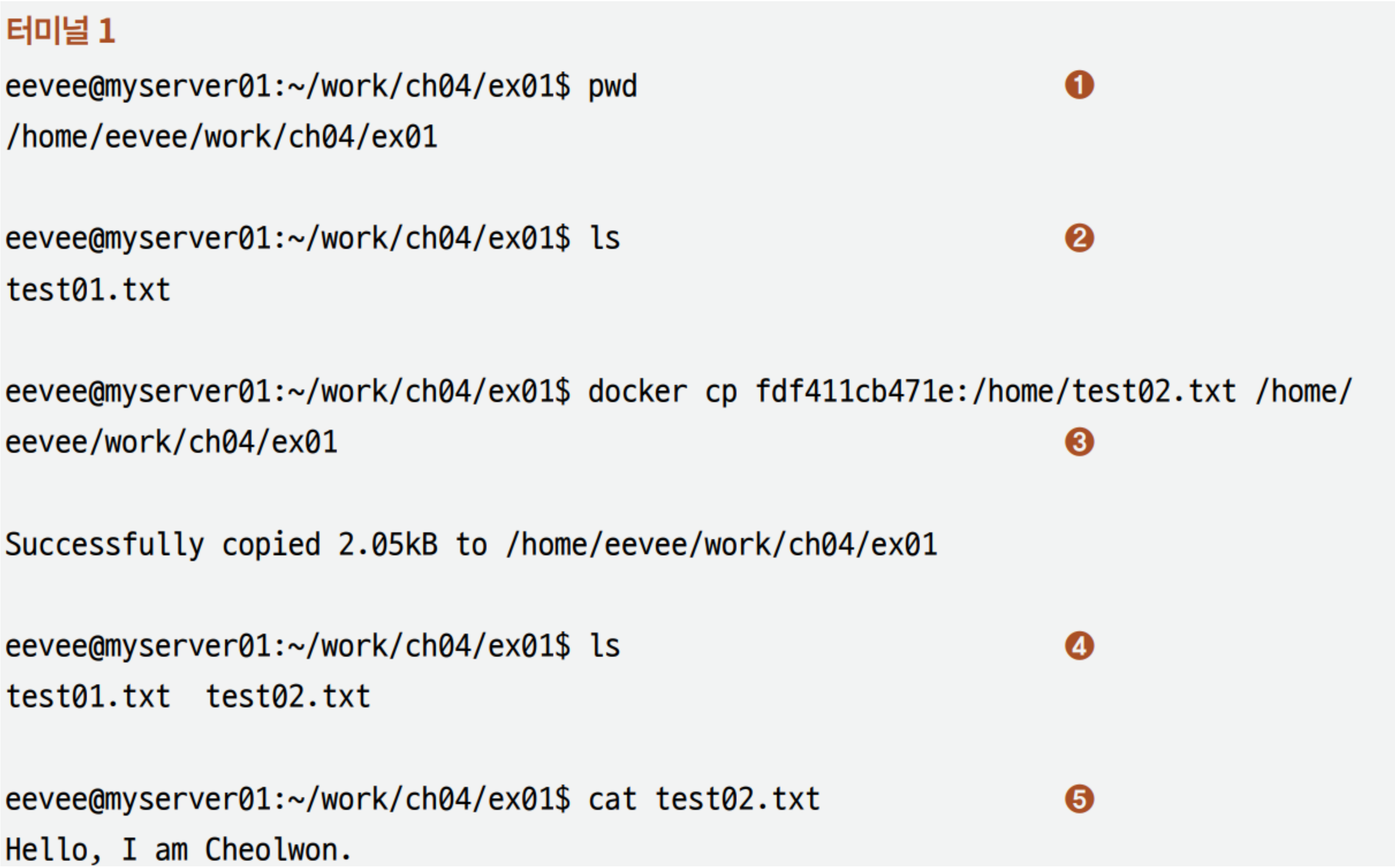
```
eevee@myserver01:~/work/ch04/ex01$ docker container ls                         ❶
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS            PORTS     NAMES
fdf411cb471e   ubuntu    "/bin/bash"   About a minute ago   Up About a minute
musing_brattain

eevee@myserver01:~/work/ch04/ex01$ docker container cp ./test01.txt fdf411cb471e:/
home                                                                           ❷
Successfully copied 2.05kB to fdf411cb471e:/home

eevee@myserver01:~/work/ch04/ex01$
```



터미널 2

```
root@fdf411cb471e:/home# ls                    ❶
test01.txt
root@fdf411cb471e:/home# cat test01.txt        ❷
Hello, I am Cheolwon.
```

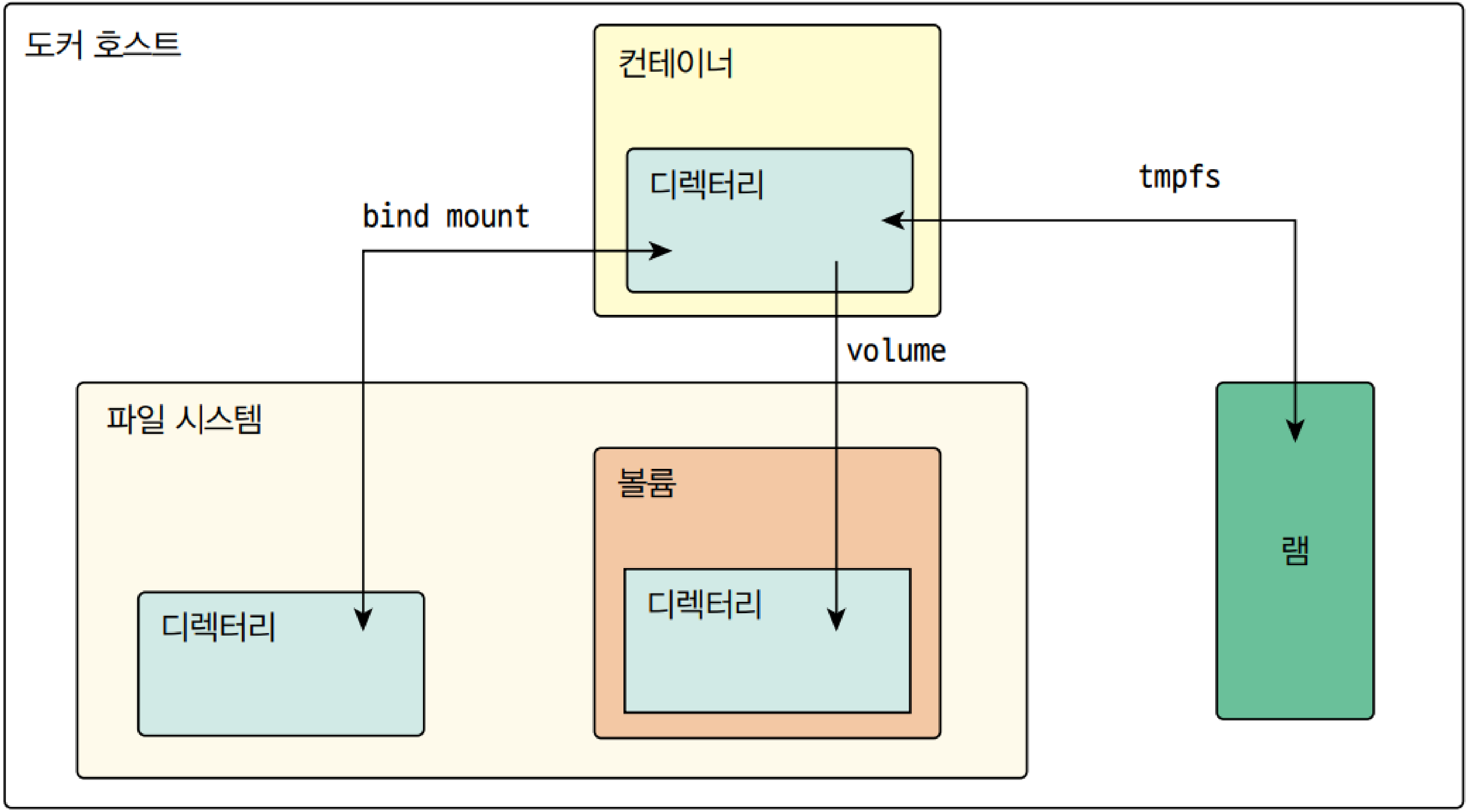
12 컨테이너에서 호스트로 파일 복사



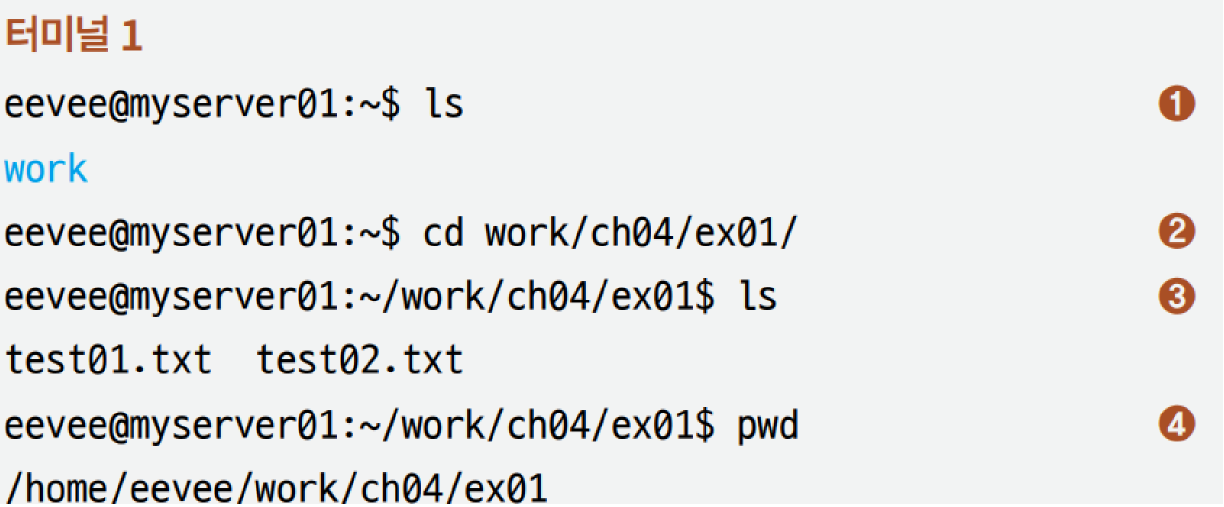
터미널 1

```
eevee@myserver01:~/work/ch04/ex01$ pwd                                    ❶
/home/eevee/work/ch04/ex01

eevee@myserver01:~/work/ch04/ex01$ ls                                     ❷
test01.txt

eevee@myserver01:~/work/ch04/ex01$ docker cp fdf411cb471e:/home/test02.txt /home/
eevee/work/ch04/ex01                                                      ❸

Successfully copied 2.05kB to /home/eevee/work/ch04/ex01

eevee@myserver01:~/work/ch04/ex01$ ls                                     ❹
test01.txt  test02.txt

eevee@myserver01:~/work/ch04/ex01$ cat test02.txt                         ❺
Hello, I am Cheolwon.
```

13 도커 스토리지



14 도커 스토리지 - bind mount

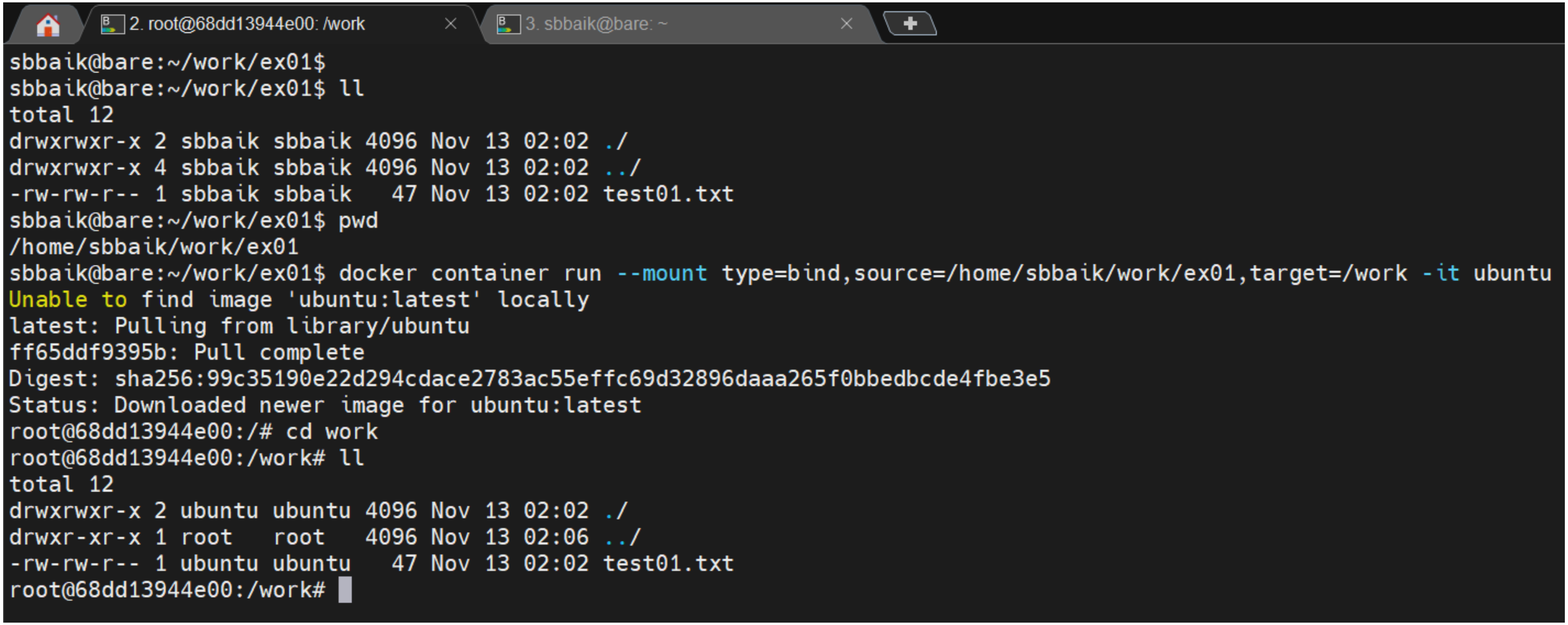


터미널 1

```
eevee@myserver01:~$ ls                                ❶
work
eevee@myserver01:~$ cd work/ch04/ex01/                ❷
eevee@myserver01:~/work/ch04/ex01$ ls                 ❸
test01.txt  test02.txt
eevee@myserver01:~/work/ch04/ex01$ pwd                ❹
/home/eevee/work/ch04/ex01
```

15 도커 스토리지 - bind mount

[터미널1]

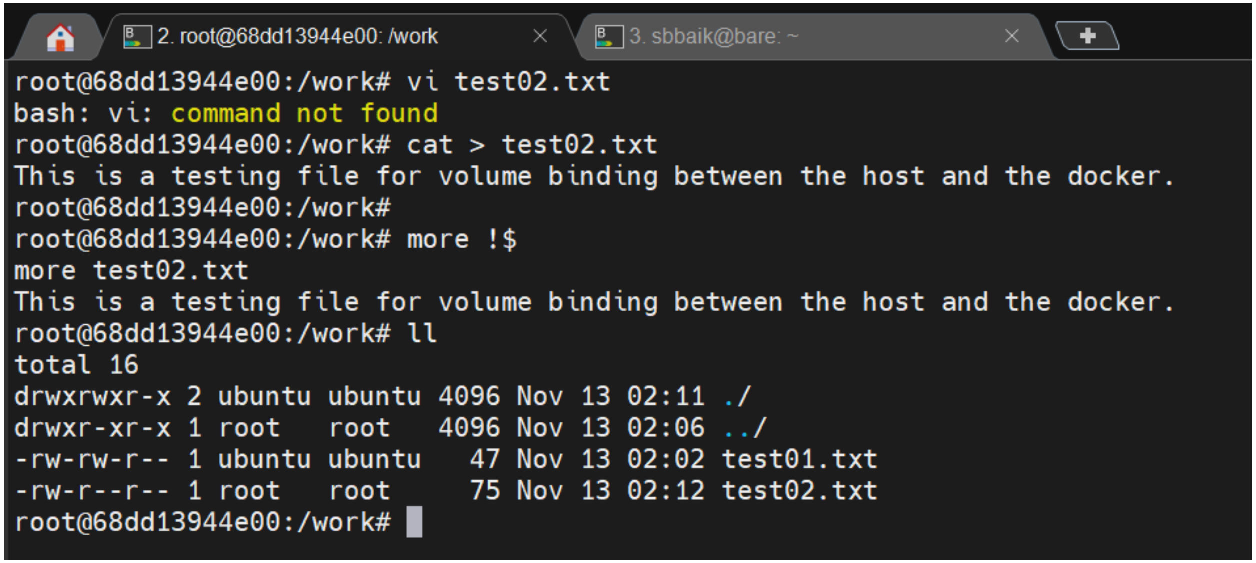


```
sbbaik@bare:~/work/ex01$
sbbaik@bare:~/work/ex01$ ll
total 12
drwxrwxr-x 2 sbbaik sbbaik 4096 Nov 13 02:02 ./
drwxrwxr-x 4 sbbaik sbbaik 4096 Nov 13 02:02 ../
-rw-rw-r-- 1 sbbaik sbbaik   47 Nov 13 02:02 test01.txt
sbbaik@bare:~/work/ex01$ pwd
/home/sbbaik/work/ex01
sbbaik@bare:~/work/ex01$ docker container run --mount type=bind,source=/home/sbbaik/work/ex01,target=/work -it ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
ff65ddf9395b: Pull complete
Digest: sha256:99c35190e22d294cdace2783ac55effc69d32896daaa265f0bbedbcde4fbe3e5
Status: Downloaded newer image for ubuntu:latest
root@68dd13944e00:/# cd work
root@68dd13944e00:/work# ll
total 12
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 13 02:02 ./
drwxr-xr-x 1 root   root   4096 Nov 13 02:06 ../
-rw-rw-r-- 1 ubuntu ubuntu   47 Nov 13 02:02 test01.txt
root@68dd13944e00:/work# 
```

$$\text{docker container run --mount} \\ \underbrace{\text{type=bind,source=/home/sbbaik/work/ex01,target=/work}}_{\text{공백 없이 붙여 쓸 것}}\ \text{-it ubuntu}$$

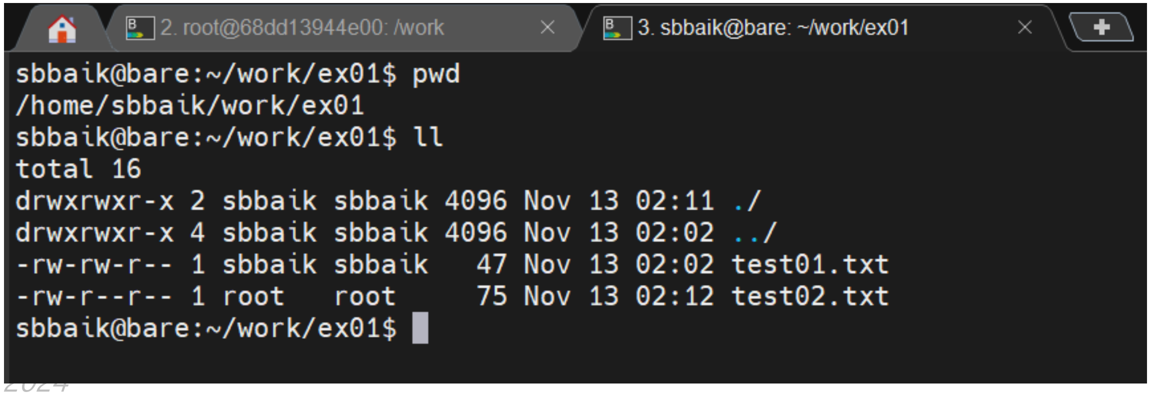
16 디렉토리 공유 확인

[터미널1]



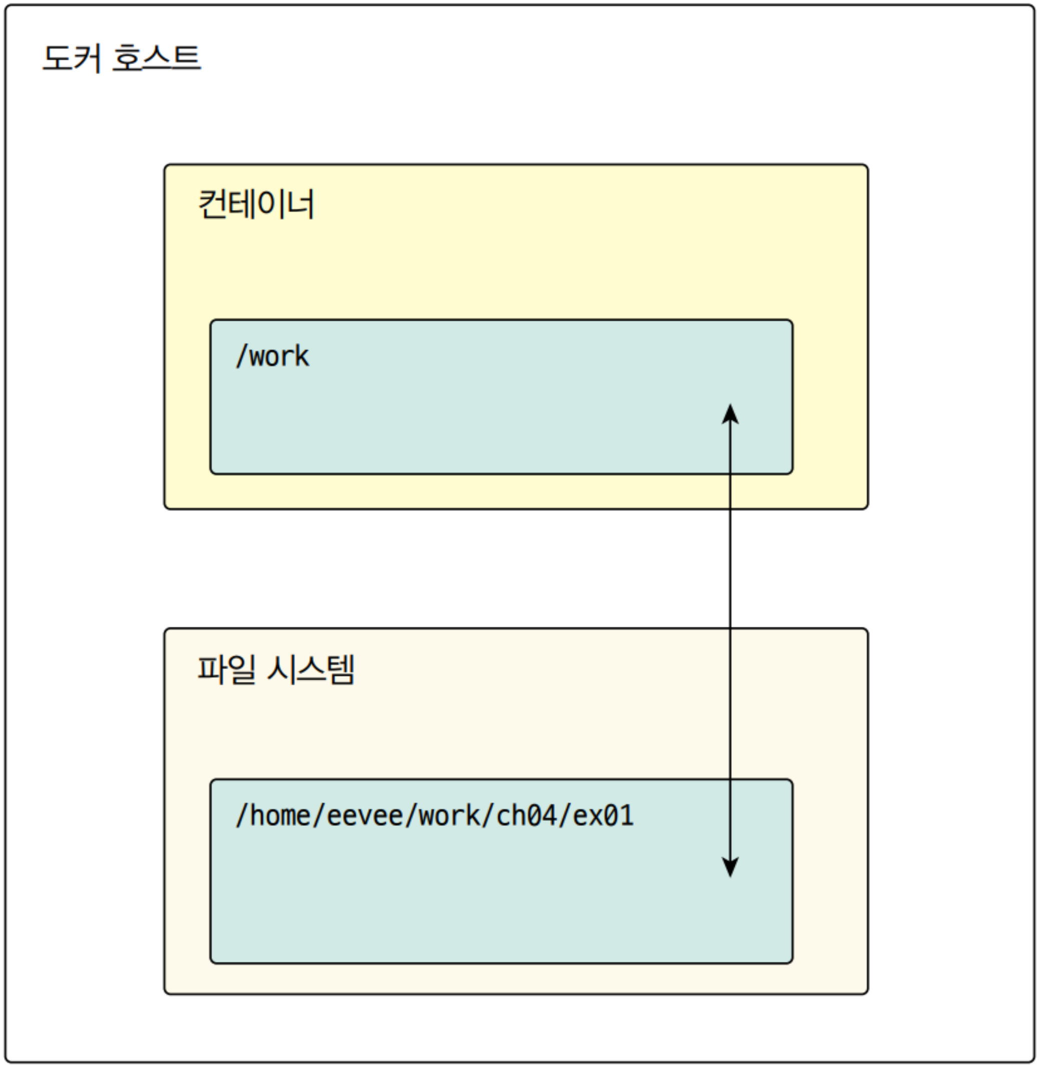
```
root@68dd13944e00:/work# vi test02.txt
bash: vi: command not found
root@68dd13944e00:/work# cat > test02.txt
This is a testing file for volume binding between the host and the docker.
root@68dd13944e00:/work#
root@68dd13944e00:/work# more !$
more test02.txt
This is a testing file for volume binding between the host and the docker.
root@68dd13944e00:/work# ll
total 16
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 13 02:11 ./
drwxr-xr-x 1 root   root   4096 Nov 13 02:06 ../
-rw-rw-r-- 1 ubuntu ubuntu   47 Nov 13 02:02 test01.txt
-rw-r--r-- 1 root   root     75 Nov 13 02:12 test02.txt
root@68dd13944e00:/work# 
```

[터미널2]



```
sbbaik@bare:~/work/ex01$ pwd
/home/sbbaik/work/ex01
sbbaik@bare:~/work/ex01$ ll
total 16
drwxrwxr-x 2 sbbaik sbbaik 4096 Nov 13 02:11 ./
drwxrwxr-x 4 sbbaik sbbaik 4096 Nov 13 02:02 ../
-rw-rw-r-- 1 sbbaik sbbaik   47 Nov 13 02:02 test01.txt
-rw-r--r-- 1 root   root     75 Nov 13 02:12 test02.txt
sbbaik@bare:~/work/ex01$ 
```

17 bind mount 개념



서로에게, 자신에게 친절 합시다 - 허준이

DAEJEON UNIVERSITY
SINCE 1980
대전대학교