Basics of docker

Autumn 2024

AI융합학과
Seongbok Baik
sbbaik@dju.ac.kr

00 Textbook

장철원 소프트웨어공학자

01 도커 컨테이너 내부 접속

```
eevee@myserver01:~$ docker container run -it ubuntu          ❶
root@d76df685ca90:/# ls                                      ❷
bin   etc   lib32   media  proc  sbin  tmp  boot  home  lib64
mnt    root  srv   usr dev   lib   libx32  opt    run   sys   var
```

(1) 옵션 -it: 가상 터미널을 통해 키보드 입력 사용

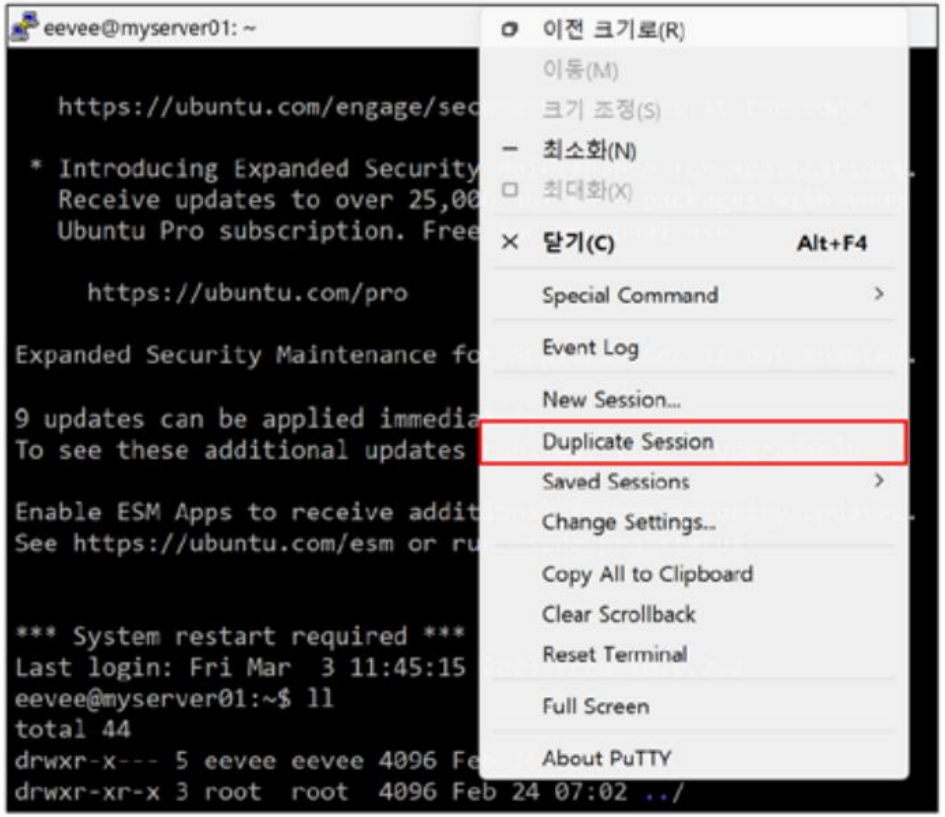
- i: interactive, 표준 입력(STDIN) 사용
- t: tty, 가상 터미널

(2) 사용자 이름과 컴퓨터 이름 변경

- 사용자 이름: root
- 컴퓨터 이름: 컨테이너 ID

02 여러 터미널 화면으로 작업하기

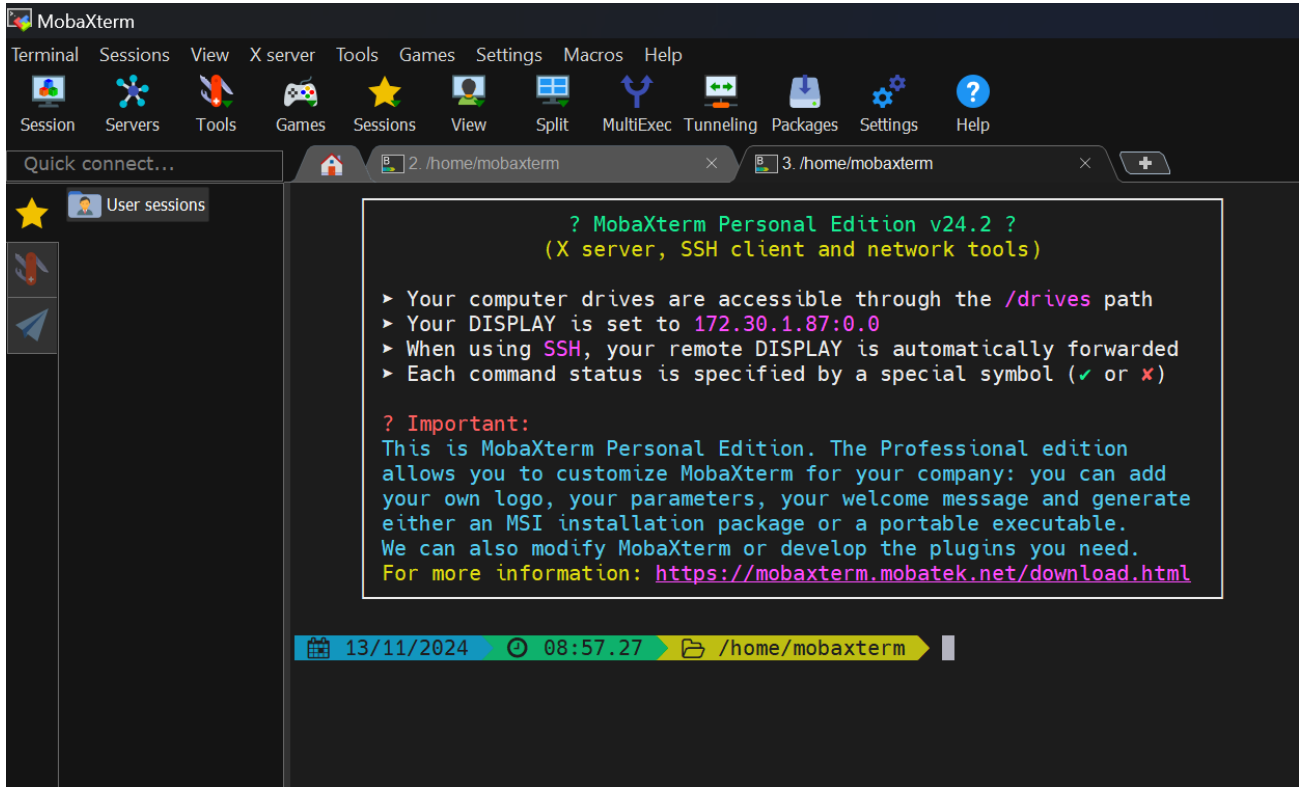
- putty에서 터미널 화면 추가



- 터미널 1 : 컨테이너 내부 접속
- 터미널 2 : 컨테이너 외부, 즉 호스트 컴퓨터 접속

03 여러 터미널 화면으로 작업하기

- MobaXterm에서 터미널 화면 추가

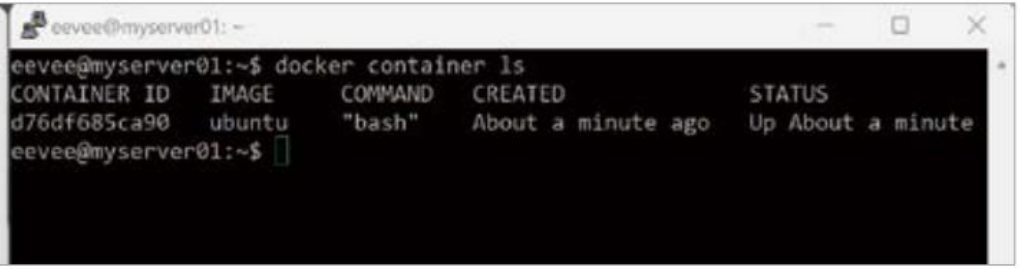


- 터미널 1 : 컨테이너 내부 접속
- 터미널 2 : 컨테이너 외부, 즉 호스트 컴퓨터 접속

04 여러 터미널 화면 생성 확인

터미널 1

터미널 2



05 실행 중인 컨테이너 상태 확인

터미널 2

```
eevee@myserver01:~$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED              STATUS              PORTS
NAMES
d76df685ca90   ubuntu    "bash"    About a minute ago   Up                  About a
minute              hungry_gauss
```

06 컨테이너 종료 명령

(첫번째 방법)

터미널 1에서 'exit' 명령 실행

(두번째 방법) 안전한 방법

터미널 2

```
eevee@myserver01:~$ docker container stop d76df685ca90
d78c2cbe3001
```

(세번째 방법) 강제 종료

터미널 2에서 docker container kill

07 컨테이너 재 접속 방법

터미널 1

```
eevee@myserver01:~$ docker container start d76df685ca90                    ❶
d76df685ca90

eevee@myserver01:~$ docker container ls                                    ❷
CONTAINER ID   IMAGE     COMMAND   CREATED          STATUS          PORTS     NAMES
d76df685ca90   ubuntu    "bash"    12 minutes ago   Up
5 seconds              hungry_gauss

eevee@myserver01:~$ docker container attach d76df685ca90                   ❸
root@d76df685ca90:/# exit                                                  ❹
exit
eevee@myserver01:~$
```

(3) attach 명령으로 컨테이너 내부 진입

08 컨테이너 삭제

```
eevee@myserver01:~$ docker container ls -a                    ❶
CONTAINER ID   IMAGE           COMMAND       CREATED          STATUS
PORTS          NAMES
d76df685ca90   ubuntu          "bash"        15 minutes ago   Exited (0)
About a minute ago             hungry_gauss
209b1ac7a1be   python:3.11.6   "python3"     18 minutes ago   Exited (0)
18 minutes ago                 musing_hamilton
19e30e8d5a98   ubuntu          "/bin/bash"   19 minutes ago   Exited (0)
```

```
eevee@myserver01:~$ docker container rm d76df685ca90          ❶
d76df685ca90
```

(1) docker container rm 명령으로 컨테이너 삭제

09 컨테이너 삭제 확인

```
eevee@myserver01:~$ docker container ls -a                    ❶
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS
PORTS     NAMES
209b1ac7a1be   python:3.11.6   "python3"     20 minutes ago   Exited (0)
20 minutes ago             musing_hamilton
19e30e8d5a98   ubuntu          "/bin/bash"   20 minutes ago   Exited (0)
20 minutes ago             epic_goldwasser
b68b3e20dc68   hello-world     "/hello"      2 hours ago      Exited (0)
2 hours ago                nervous_shannon
```

10 여러 컨테이너 삭제 동시 삭제

```
eevee@myserver01:~$ docker container rm 19e30e8d5a98 b68b3e20dc68        ❶
19e30e8d5a98
b68b3e20dc68
eevee@myserver01:~$ docker container ls -a                               ❷
CONTAINER ID   IMAGE           COMMAND     CREATED          STATUS
PORTS     NAMES
209b1ac7a1be   python:3.11.6   "python3"   21 minutes ago   Exited (0)
21 minutes ago             musing_hamilton
```

11 도커 이미지 삭제

```
eevee@myserver01:~$ docker image ls                    ❶
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
python        3.11.6    3f7984adbac4   2 weeks ago     1.01GB
ubuntu        latest    e4c58958181a   3 weeks ago     77.8MB
hello-world   latest    9c7a54a9a43c   6 months ago    13.3kB
```

```
eevee@myserver01:~$ docker image rm 9c7a54a9a43c                    ❶
Untagged: hello-world:latest
Untagged: hello-world@sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c
5af48e8d
Deleted: sha256:9c7a54a9a43cca047013b82af109fe963fde787f63f9e016fdc3384500c2823d
Deleted: sha256:01bb4fce3eb1b56b05adf99504dafd31907a5aadac736e36b27595c8b92f07f1
```

12 도커 이미지 변경 - 나만의 도커 이미지 만들기

터미널 1

```
eevee@myserver01:~$ docker image ls                                    ❶
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
python       3.11.6    3f7984adbac4   2 weeks ago   1.01GB
ubuntu       latest    e4c58958181a   3 weeks ago   77.8MB
eevee@myserver01:~$ docker container run -it ubuntu                    ❷
root@67f64db65b10:/#                                                   ❸
```

기존에 다운로드해서 사용 중이던 ubuntu 도커 이미지에 "네트워크 도구"를 추가 설치한 나만의 도커 이미지 생성

13 네트워크 도구 부재 확인

터미널 1

```
root@67f64db65b10:/# ifconfig                                    ❶
bash: ifconfig: command not found

root@67f64db65b10:/# apt update && apt install net-tools         ❷
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
……
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
```

컨테이너에 "네트워크 도구"가 설치되어 있지 않다는 것을 확인하고 추가 설치

14 네트워크 도구 설치 확인

터미널 1

```
root@67f64db65b10:/# ifconfig                                          ❶
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255  ❷
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 6174  bytes 28783616 (28.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3977  bytes 219865 (219.8 KB)
        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
```

ifconfig 커맨드로 "네트워크 도구" 설치 확인

15 나만의 도커 이미지 생성

터미널 2

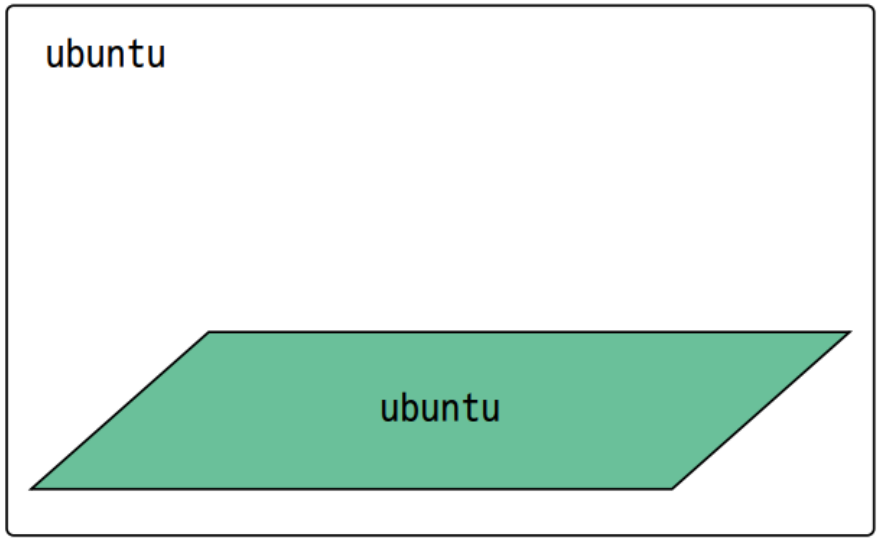
```
eevee@myserver01:~$ docker container ls                                    ❶
CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS         PORTS     NAMES
67f64db65b10   ubuntu    "bash"    5 minutes ago   Up         5 minutes
upbeat_noyce
```

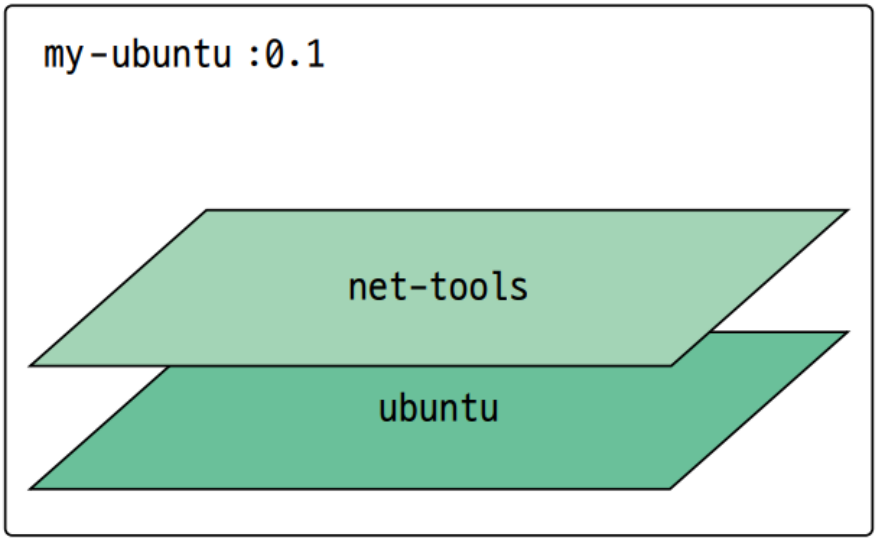
터미널 2

```
eevee@myserver01:~$ docker container commit 67f64db65b10 my-ubuntu:0.1     ❶
sha256:8252e4215d037d3ece56d21232d41306557cad6d1dfedd275fb8ec824b8cbf96
```

"commit" 명령어로 나만의 도커이미지 "my-ubuntu:0.1" 생성

16 도커 이미지 변경 개념





17 기존 도커 컨테이서 제거

터미널 1

```
root@67f64db65b10:/# exit                                          ❶
exit

eevee@myserver01:~$ docker container ls -a                         ❷
CONTAINER ID   IMAGE           COMMAND     CREATED          STATUS
PORTS     NAMES
67f64db65b10   ubuntu          "bash"      7 minutes ago    Exited (0)
31 seconds ago             upbeat_noyce
209b1ac7a1be   python:3.11.6   "python3"   41 minutes ago   Exited (0)
41 minutes ago             musing_hamilton

eevee@myserver01:~$ docker container rm 67f64db65b10               ❸
67f64db65b10
```

18 신규 생성 도커 이미지 확인

```
eevee@myserver01:~$ docker image ls                                    4
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
my-ubuntu    0.1       8252e4215d03   About a minute ago   125MB
python       3.11.6    3f7984adbac4   2 weeks ago          1.01GB
ubuntu       latest    e4c58958181a   3 weeks ago          77.8MB
```

19 신규 생성 도커 컨테이너 실행 및 확인

```
eevee@myserver01:~$ docker container run -it my-ubuntu:0.1      ❺

root@eafefe31241c:/# ifconfig                                   ❻
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 9  bytes 806 (806.0 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

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

20 도커 이미지 명령어

명령어	설명
docker image build	Dockerfile로부터 이미지 빌드합니다.
docker image history	이미지 히스토리를 확인합니다.
docker image import	파일 시스템 이미지 생성을 위한 타볼tarball 콘텐츠를 임포트합니다.
docker image inspect	이미지 정보를 표시합니다.
docker image load	타볼로 묶인 이미지를 로드합니다.
docker image ls	이미지 목록을 확인합니다.
docker image prune	사용하지 않는 이미지를 삭제합니다.
docker image pull	레지스트리로부터 이미지를 다운로드합니다.
docker image push	레지스트리로 이미지를 업로드합니다.
docker image rm	하나 이상의 이미지를 삭제합니다.
docker image save	이미지를 타볼로 저장합니다.
docker image tag	이미지 태그를 생성합니다.

21 도커 컨테이너 명령어-1

명령어	설명
docker container attach	실행 중인 컨테이너의 표준 입출력 스트림에 붙습니다(attach).
docker container commit	변경된 컨테이너에 대한 새로운 이미지를 생성합니다.
docker container cp	컨테이너와 로컬 파일 시스템 간 파일/폴더를 복사합니다.
docker container create	새로운 컨테이너를 생성합니다.
docker container diff	컨테이너 파일 시스템의 변경 내용을 검사합니다.
docker container exec	실행 중인 컨테이너에 명령어를 실행합니다.
docker container export	컨테이너 파일 시스템을 타볼로 추출합니다.
docker container inspect	하나 이상의 컨테이너의 자세한 정보를 표시합니다.
docker container kill	하나 이상의 실행 중인 컨테이너를 kill합니다.
docker container logs	컨테이너 로그를 불러옵니다.
docker container ls	컨테이너 목록을 확인합니다.

22 도커 컨테이너 명령어-2

docker container pause	하나 이상의 컨테이너 내부의 모든 프로세스를 정지합니다.
docker container port	특정 컨테이너의 매핑된 포트 리스트를 확인합니다.
docker container prune	멈춰 있는(stopped) 모든 컨테이너를 삭제합니다.
docker container rename	컨테이너 이름을 다시 짓습니다.
docker container restart	하나 이상의 컨테이너를 재실행합니다.
docker container rm	하나 이상의 컨테이너를 삭제합니다.
docker container run	이미지로부터 컨테이너를 생성하고 실행합니다.
docker container start	멈춰 있는 하나 이상의 컨테이너를 실행합니다.
docker container stats	컨테이너 리소스 사용 통계를 표시합니다.
docker container stop	하나 이상의 실행 중인 컨테이너를 정지합니다.
docker container top	컨테이너의 실행 중인 프로세스를 표시합니다.
docker container unpause	컨테이너 내부의 멈춰 있는 프로세스를 재실행합니다.
docker container update	하나 이상의 컨테이너 설정을 업데이트합니다.
docker container wait	컨테이너가 종료될 때까지 기다린 후 exit code를 표시합니다.

서로에게, 자신에게 친절합시다 - 허준이

DAEJEON UNIVERSITY
SINCE 1980
대전대학교