

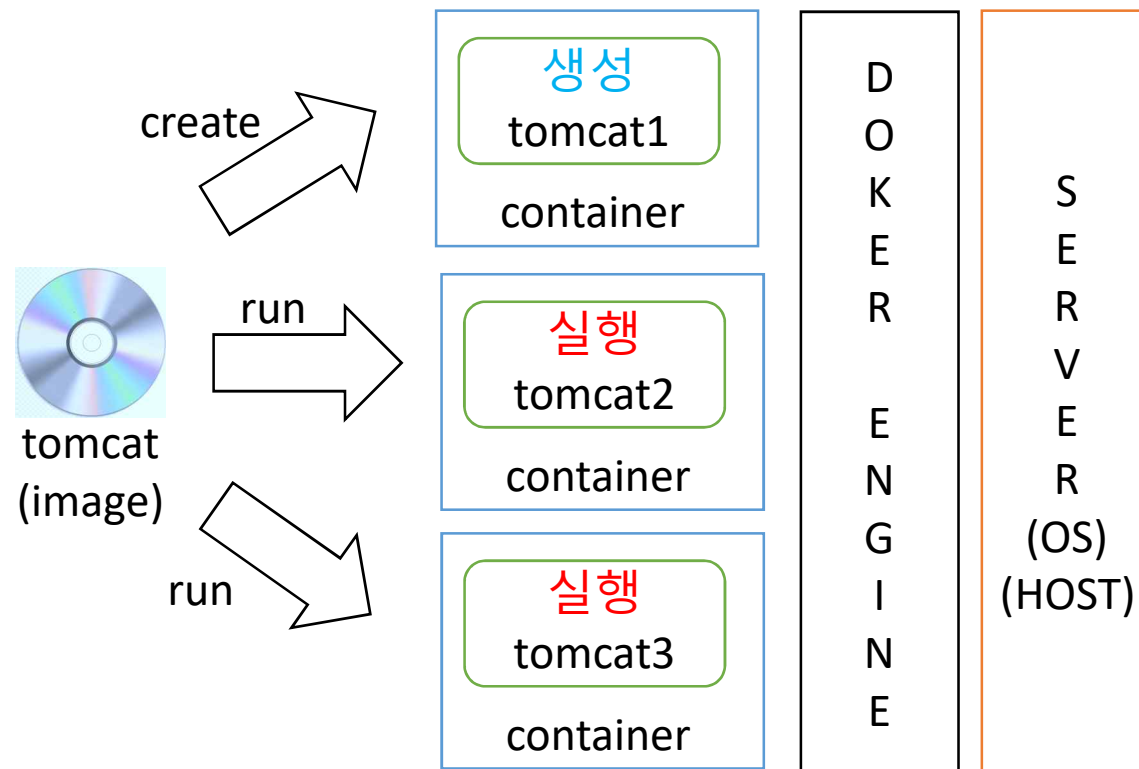


Docker Container

▶ Docker Container

✓ Container

image를 기반으로 생성되는 한 개의 독립적인 프로세서
생성된 container는 격리되어 실행되기 때문에 서로 영향을 주지 않음



Docker container 명령어

▶ Docker Container

✓ Container 명령어

docker container 명령어 [옵션] [이미지명 || container식별자]

✓ Container 실행, 중단, 삭제 명령어

명령어	내용
create	이미지를 기반으로 container를 생성
run	이미지를 기반으로 container를 생성하고 실행
start	생성, 중단된 container를 실행
stop	실행중인 container를 중단
rm	생성된 container 삭제
restart	중단된 container를 재실행
pause / unpause	실행중인 컨테이너를 일시정지 / 재실행

▶ Docker Container

✓ Container 조작 명령어

명령어	내용
attach	가동 중인 container에 연결 * 백그라운드 실행 container가 쉘이 동작하지 않으면 연결 불가
exec	가동 중인 container에 새로운 프로세스 생성 후 연결
rename	container 생성 이름을 변경
cp	host와 container 사이에 파일을 복사

✓ Container 조회 명령어

명령어	내용
ls	생성된 container 조회
stats	container의 가동상태 확인
top	container의 프로세스 정보를 확인
diff	기본 생성된 container와 차이점 조회
port	container가 사용하는 port 확인

**Container 실행,
중단, 삭제 명령어**

▶ Docker 명령어

✓ container실행 명령어

👉 **docker container run [옵션] 이미지명 || 이미지 ID**

container를 실행하는 명령어, pull 하지 않은 이미지는 자동으로
다운로드 후 실행

기본실행 옵션

이름	내용
--name	container이름설정
--interactive,-i	container에 표준 입력 열기
--tty,-t	tty(단말기 디바이스)사용
--detach,-d	container를 백그라운드 실행
--rm	명령실행 완료 후 container 자동 삭제

▶ Docker명령어

✓ 기본 docker run 실습

- 기본 실행하기 *ubuntu, tomcat pull 받아서 실습

명령어 : docker container run ubuntu
docker container ls -a

```
C:\Users\bsyoo>docker container run ubuntu

C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
35f480424351	ubuntu	"/bin/bash"	41 seconds ago	Exited (0) 40 seconds ago		jovial_easley

```
C:\Users\bsyoo>
```

ubuntu를 실행하면 단순히 실행 후 가동을 종료 됨. * STATUS Exited

- ubuntu container이용해서 문구 출력하기

명령어 : docker container run ubuntu echo hello docker
docker container ls -a

```
C:\Users\bsyoo>docker container run ubuntu echo hello docker
hello docker

C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
81ebe8498233	ubuntu	"echo hello docker"	4 seconds ago	Exited (0) 3 seconds ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	17 seconds ago	Exited (0) 17 seconds ago		nervous_payne

▶ Docker명령어

✓ 기본 docker run 실습

- 기본 실행하기 tomcat 실행하기

명령어 : docker container run tomcat
tomcat이 실행되고 tomcat 로그가 출력됨.

```
C:\Users\bsyoo>docker container run tomcat
11-Aug-2023 08:39:18.441 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name: Apache Tomcat/10.1.11
11-Aug-2023 08:39:18.444 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built: Jul 6 2023 13:45:39 UTC
11-Aug-2023 08:39:18.444 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version number: 10.1.11.0
11-Aug-2023 08:39:18.445 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name: Linux
11-Aug-2023 08:39:18.445 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version: 5.10.16.3-microsoft-standard-WSL2
11-Aug-2023 08:39:18.445 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture: amd64
11-Aug-2023 08:39:18.445 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home: /opt/java/openjdk
11-Aug-2023 08:39:18.446 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version: 17.0.8+7
11-Aug-2023 08:39:18.446 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor: Eclipse Adoptium
11-Aug-2023 08:39:18.446 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE: /usr/local/tomcat
11-Aug-2023 08:39:18.447 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME: /usr/local/tomcat
11-Aug-2023 08:39:18.459 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties
11-Aug-2023 08:39:18.460 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
11-Aug-2023 08:39:18.460 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djdk.tls.ephemeralDHKeySize=2048
```

run 했을때 container별 설정에 따라 실행이 다름, 새 cmd를 열어 실행 확인, * container에 네트워크설정을 하지 않아 접속은 되지 않음

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad2fde869450	tomcat	"/entrypoint.sh cata..."	4 minutes ago	Up 4 minutes	8080/tcp	ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	6 minutes ago	Exited (0) 6 minutes ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	6 minutes ago	Exited (0) 6 minutes ago		nervous_payne

▶ Docker명령어

✓ docker run 옵션

명령어 : docker container run --name testubuntu -it ubuntu
실행 컨테이너의 이름을 설정하고 prompt를 연결하여 명령어 입력창 출력
-- name속성값은 중복되면 안됨.

```
C:\Users\bsyoo>docker container run --name testubuntu -it ubuntu
root@274629d116fc:/# |
```

새 커멘트창을 열어 실행 중인 도커확인
실행 컨테이너이름이 설정되고 status가 up중 임

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
274629d116fc	ubuntu	"/bin/bash"	6 minutes ago	Up 6 minutes		<u>testubuntu</u>
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) 8 minutes ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 55 minutes ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

▶ Docker명령어

✓ docker run 옵션

-it 옵션으로 운영체제에 접속하여 명령어를 입력할 수 있는 커멘드창이 실행됨.

* container는 기본적으로 리눅스가 기본 운영체제로 bash shell이름.
기본 명령어 실행확인 pwd, ls, cat /etc/os-release

```
root@274629d116fc:/# pwd
/
root@274629d116fc:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@274629d116fc:/# whoami
root
root@274629d116fc:/# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.2 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

▶ Docker명령어

✓ docker run 옵션

container 콘솔창 종료하기 * name설정값으로 propmt를 호출가능
실행 종료 : exit

```
root@274629d116fc:/# exit
exit
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
274629d116fc	ubuntu	"/bin/bash"	About an hour ago	Exited (0) 9 seconds ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) About an hour ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 2 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

container콘솔창 종료하기

propmt만 종료 : ctrl+p+q

* container 재실행 후 attach로 접속 후 종료하기

```
C:\Users\bsyoo>docker container restart testubuntu
testubuntu
```

```
C:\Users\bsyoo>docker container attach testubuntu
root@274629d116fc:/# read escape sequence
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
274629d116fc	ubuntu	"/bin/bash"	About an hour ago	Up 11 minutes		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) About an hour ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 2 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

▶ Docker명령어

✓ docker run 옵션

--detach, -d 옵션

명령어 : docker container run --name testtomcat -d tomcat
container백그라운드에서 실행시키기

```
C:\Users\bsyoo>docker container run --name testtomcat -d tomcat
01b04a05fe052642ee1602fe89c36f5cbf36619e9f2af7499eef8f11c0d3f3c2

C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
01b04a05fe05	tomcat	"/entrypoint.sh cata..."	8 seconds ago	Up 8 seconds	8080/tcp	testtomcat

--rm 옵션

명령어 : docker container stop testtomcat

실행중이 container가 종료하면 자동으로 container를 삭제하는 옵션
rm옵션이 없는 container를 중단하면 STATUS가 Exited로 설정되고 삭제
되지 않음

```
C:\Users\bsyoo>docker container stop testtomcat
testtomcat

C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
01b04a05fe05	tomcat	"/entrypoint.sh cata..."	6 minutes ago	Exited (143) 4 seconds ago		testtomcat

▶ Docker명령어

✓ docker run 옵션

--rm 옵션

명령어 : docker container run --name testtomcat -d --rm tomcat
rm 옵션을 설정하고 container 실행 중단하기
중단되는 순간 container를 삭제

```
C:\Users\bsyoo>docker container rm testtomcat
testtomcat
```

```
C:\Users\bsyoo>docker container run --name testtomcat -d --rm tomcat
5f24ed1cdbafccba7940b33eeeab63657ea4ce900283d2bd7dc69afd7ad61558
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5f24ed1cdbaf	tomcat	"/entrypoint.sh cata..."	13 seconds ago	Up 13 seconds	8080/tcp	testtomcat
422348bd374a	ubuntu	"/bin/bash"	14 minutes ago	Exited (0) 14 minutes ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) 3 hours ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 4 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

```
C:\Users\bsyoo>docker container stop testtomcat
testtomcat
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
422348bd374a	ubuntu	"/bin/bash"	14 minutes ago	Exited (0) 14 minutes ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) 3 hours ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 4 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

▶ Docker 명령어

✓ container 실행, 재실행 명령어

👉 `docker container start [옵션] container이름||아이디`

실행중인 컨테이너를 중단하는 명령어

STATUS를 Created,Exited -> Up으로 변경해 줌

명령어 : `docker container start stoptomcat`

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	11 minutes ago	Exited (143) 5 seconds ago		stoptomcat
422348bd374a	ubuntu	"/bin/bash"	3 hours ago	Exited (0) 3 hours ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	2 days ago	Exited (143) 7 hours ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	2 days ago	Exited (0) 7 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	2 days ago	Exited (0) 2 days ago		nervous_payne

```
C:\Users\bsyoo>docker container start stoptomcat
stoptomcat

C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	11 minutes ago	Up 6 seconds	8080/tcp	stoptomcat

▶ Docker 명령어

✓ container 중단 명령어

👉 **docker container stop container이름||아이디**

실행중인 컨테이너를 중단하는 명령어

STATUS를 Up -> Exited로 변경해 줌

명령어 : docker container run -d --name stoptomcat tomcat
docker container stop stoptomcat

```
C:\Users\bsyoo>docker container run -d --name stoptomcat tomcat
3f80c6cc84bcc2f38bdc43070bb8640ac28c489ec6dc405d812ced7a7a7ede08
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	8 seconds ago	Up 7 seconds	8080/tcp	stoptomcat

```
C:\Users\bsyoo>docker container stop stoptomcat
stoptomcat
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	About a minute ago	Exited (143) 5 seconds ago		stoptomcat

▶ Docker 명령어

✓ container 실행, 재실행 명령어

👉 `docker container restart container이름||아이디`

실행중인 컨테이너를 중단하는 명령어

STATUS를 Created,Exited -> Up으로 변경해 줌

명령어 : `docker container restart stoptomcat`

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	2 hours ago	Exited (143) 2 hours ago		stoptomcat
422348bd374a	ubuntu	"/bin/bash"	6 hours ago	Exited (0) 2 hours ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	3 days ago	Exited (143) 9 hours ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	3 days ago	Exited (0) 10 hours ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	3 days ago	Exited (0) 3 days ago		nervous_payne

```
C:\Users\bsyoo>docker container restart stoptomcat
stoptomcat
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f80c6cc84bc	tomcat	"/entrypoint.sh cata..."	2 hours ago	Up 7 seconds	8080/tcp	stoptomcat

▶ Docker 명령어

✓ container 삭제 명령어

👉 `docker container rm container이름||아이디`

중단시킨 후 컨테이너를 삭제

명령어 : `docker container stop stoptomcat`
`docker container rm stoptomcat`

```
C:\Users\bsyoo>docker container stop stoptomcat
stoptomcat

C:\Users\bsyoo>docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
3f80c6cc84bc   tomcat    "/entrypoint.sh cata..." 3 hours ago    Exited (143) 5 seconds ago           stoptomcat
422348bd374a   ubuntu    "/bin/bash"               6 hours ago    Exited (0) 2 hours ago               testubuntu
ad2fde869450   tomcat    "/entrypoint.sh cata..." 3 days ago     Exited (143) 9 hours ago             ecstatic_merkle
81ebe8498233   ubuntu    "echo hello docker"        3 days ago     Exited (0) 10 hours ago              exciting_leakey
0b212e108965   ubuntu    "/bin/bash"               3 days ago     Exited (0) 3 days ago               nervous_payne

C:\Users\bsyoo>docker container rm stoptomcat
stoptomcat

C:\Users\bsyoo>docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
422348bd374a   ubuntu    "/bin/bash"               6 hours ago    Exited (0) 2 hours ago               testubuntu
ad2fde869450   tomcat    "/entrypoint.sh cata..." 3 days ago     Exited (143) 9 hours ago             ecstatic_merkle
81ebe8498233   ubuntu    "echo hello docker"        3 days ago     Exited (0) 10 hours ago              exciting_leakey
0b212e108965   ubuntu    "/bin/bash"               3 days ago     Exited (0) 3 days ago               nervous_payne
```

▶ Docker 명령어

✓ container 삭제 명령어

👉 **docker container rm container이름||아이디**

실행중인 컨테이너 삭제

명령어 : `docker container run -d --name testtomcat tomcat`

`docker container rm testtomcat`

실행중인 container는 삭제를 하지 못함. 중단상태 container만 삭제가능

```
C:\Users\bsyoo>docker container run -d --name testtomcat tomcat
8a4fd5886d3ce702a21ab017734fdabc09a4e254669e2d2c0257b8966a7729e3
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8a4fd5886d3c	tomcat	"/entrypoint.sh cata..."	8 seconds ago	Up 7 seconds	8080/tcp	testtomcat
422348bd374a	ubuntu	"/bin/bash"	44 hours ago	Exited (0) 41 hours ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	4 days ago	Exited (143) 47 hours ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	4 days ago	Exited (0) 2 days ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	4 days ago	Exited (0) 4 days ago		nervous_payne

```
C:\Users\bsyoo>docker container rm testtomcat
```

```
Error response from daemon: You cannot remove a running container 8a4fd5886d3ce702a21ab017734fdabc09a4e254669e2d2c0257b8966a7729e3. Stop the container before attempting removal or force remove
```

▶ Docker 명령어

✓ container 삭제 명령어

👉 `docker container rm container이름||아이디`

실행중인 컨테이너는 삭제하지 못함. 중단 후 삭제해야함.

명령어 : `docker container stop testtomcat`
`docker container rm testtomcat`

```
C:\Users\bsyoo>docker container stop testtomcat
testtomcat
```

```
C:\Users\bsyoo>docker container rm testtomcat
testtomcat
```

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
422348bd374a	ubuntu	"/bin/bash"	44 hours ago	Exited (0) 41 hours ago		testubuntu
ad2fde869450	tomcat	"/entrypoint.sh cata..."	4 days ago	Exited (143) 2 days ago		ecstatic_merkle
81ebe8498233	ubuntu	"echo hello docker"	4 days ago	Exited (0) 2 days ago		exciting_leakey
0b212e108965	ubuntu	"/bin/bash"	4 days ago	Exited (0) 4 days ago		nervous_payne

▶ Docker 명령어

✓ container 일괄 삭제 명령어

☞ **docker container rm \$(docker ps -aq) 리눅스**
FOR /f "tokens=* " %i IN ('docker ps -aq')
DO docker container rm %i 윈도우

명령어 : **FOR /f "tokens=*" %i IN ('docker ps -aq') DO docker container rm %i**

```
C:\Users\bsyoo>docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
422348bd374a   ubuntu   "/bin/bash"             46 hours ago   Exited (0)    42 hours ago   testubuntu
ad2fde869450   tomcat   "/entrypoint.sh cata..." 4 days ago     Exited (143)  2 days ago     ecstatic_merkle
81ebe8498233   ubuntu   "echo hello docker"      4 days ago     Exited (0)    2 days ago     exciting_leakey
0b212e108965   ubuntu   "/bin/bash"             4 days ago     Exited (0)    4 days ago     nervous_payne

C:\Users\bsyoo>FOR /f "tokens=*" %i IN ('docker ps -aq') DO docker container rm %i

C:\Users\bsyoo>docker container rm 422348bd374a
422348bd374a

C:\Users\bsyoo>docker container rm ad2fde869450
ad2fde869450

C:\Users\bsyoo>docker container rm 81ebe8498233
81ebe8498233

C:\Users\bsyoo>docker container rm 0b212e108965
0b212e108965

C:\Users\bsyoo>docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
```

Container 조회 명령어

▶ Docker 명령어

✓ container조회 명령어

👉 `docker container ls [옵션] container이름 || 아이디`

👉 `docker ps [옵션] container이름 || 아이디`

현재 실행, 중단된 container를 조회하는 명령어

이름	내용
<code>--all, -a</code>	모든 container를 조회
<code>--filter, -f</code>	지정된 key값으로 조회
<code>--quiet, -q</code>	container아이디만 조회
<code>--size, -s</code>	container 파일크기 조회
<code>--format</code>	원하는 template으로 조회(json)
<code>--no-trunc</code>	상세내용까지 조회
<code>--latest, -l</code>	최신container 1개 조회
<code>--last, -n</code>	최신 n개 container 조회

▶ Docker명령어

✓ docker container ls 실습

다수의 컨테이너 실행하기 * background로 container실행

명령어 : docker container run -d --name mytomcat1 tomcat
docker container run -d --name mytomcat2 tomcat
docker container run -d --name mytomcat3 tomcat
docker container run -d --name mytomcat4 tomcat

```
C:\Users\bsyoo>docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\bsyoo>docker container run -d --name mytomcat1 tomcat
bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f
C:\Users\bsyoo>docker container run -d --name mytomcat2 tomcat
5baf9fae9357f9ee5e0e3b302b7d28a13c9e3543d8b730dec5b74bbd8c3ee383
C:\Users\bsyoo>docker container run -d --name mytomcat3 tomcat
aea8ca8bfe084f573780c5f2238a2f1e87ac8c537eecca376c67a290ba3b22f4
C:\Users\bsyoo>docker container run -d --name mytomcat4 tomcat
4910bf9428043e7c37bc67387172a8d5cf7b582ab9448b5a32105998a314b0e6
```


▶ Docker명령어

✓ docker container ls 실습

전체 컨테이너 조회하기

명령어 : docker container ls -a

```
C:\Users\bsyoo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4910bf942804	tomcat	"/entrypoint.sh cata..."	About a minute ago	Up About a minute	8080/tcp	mytomcat4
aea8ca8bfe08	tomcat	"/entrypoint.sh cata..."	About a minute ago	Up About a minute	8080/tcp	mytomcat3
5baf9fae9357	tomcat	"/entrypoint.sh cata..."	About a minute ago	Up About a minute	8080/tcp	mytomcat2
bbc9653b5db5	tomcat	"/entrypoint.sh cata..."	About a minute ago	Up About a minute	8080/tcp	mytomcat1

명령어 : docker ps -a

```
C:\Users\bsyoo>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4910bf942804	tomcat	"/entrypoint.sh cata..."	2 minutes ago	Up 2 minutes	8080/tcp	mytomcat4
aea8ca8bfe08	tomcat	"/entrypoint.sh cata..."	2 minutes ago	Up 2 minutes	8080/tcp	mytomcat3
5baf9fae9357	tomcat	"/entrypoint.sh cata..."	2 minutes ago	Up 2 minutes	8080/tcp	mytomcat2
bbc9653b5db5	tomcat	"/entrypoint.sh cata..."	2 minutes ago	Up 2 minutes	8080/tcp	mytomcat1

명령어 : docker ps || docker ls * 실행중인 컨테이너만 조회하기

```
C:\Users\bsyoo>docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bbc9653b5db5	tomcat	"/entrypoint.sh cata..."	12 days ago	Up 29 seconds	8080/tcp	mytomcat1


```
C:\Users\bsyoo>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bbc9653b5db5	tomcat	"/entrypoint.sh cata..."	12 days ago	Up 31 seconds	8080/tcp	mytomcat1

▶ Docker명령어

✓ docker container ls 실습

전체 컨테이너 아이디만 조회하기
명령어 : docker container ls -aq
docker ps -aq

```
C:\Users\bsyoo>docker container ls -aq
4910bf942804
aea8ca8bfe08
5baf9fae9357
bbc9653b5db5
```

```
C:\Users\bsyoo>docker ps -aq
4910bf942804
aea8ca8bfe08
5baf9fae9357
bbc9653b5db5
```

▶ Docker명령어

✓ docker container ls 실습

최근생성된 container 2개 상세조회

명령어 : docker container ls -n=2 --no-trunc
docker ps -n=2 --no-trunc

```
C:\Users\bsyoo>docker container ls -n=2 --no-trunc
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4910bf9428043e7c37bc67387172a8d5cf7b582ab9448b5a32105998a314b0e6	tomcat	"/entrypoint.sh catalina.sh run"	15 minutes ago	Up 15 minutes	8080/tcp	mytomcat4
aea8ca8bfe084f573780c5f2238a2f1e87ac8c537ecca376c67a290ba3b22f4	tomcat	"/entrypoint.sh catalina.sh run"	15 minutes ago	Up 15 minutes	8080/tcp	mytomcat3


```
C:\Users\bsyoo>docker ps -n=2 --no-trunc
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4910bf9428043e7c37bc67387172a8d5cf7b582ab9448b5a32105998a314b0e6	tomcat	"/entrypoint.sh catalina.sh run"	16 minutes ago	Up 16 minutes	8080/tcp	mytomcat4
aea8ca8bfe084f573780c5f2238a2f1e87ac8c537ecca376c67a290ba3b22f4	tomcat	"/entrypoint.sh catalina.sh run"	16 minutes ago	Up 16 minutes	8080/tcp	mytomcat3

최근생성된 container 1개 json방식으로 조회

명령어 : docker container ls --latest --format "{{json .}}"
docker ps --latest --format "{{json .}}"

```
C:\Users\bsyoo>docker container ls --latest --format "{{json .}}"
```

```
{
  "Command": "\"/entrypoint.sh cata...\"",
  "CreatedAt": "2023-08-16 12:19:21 +0900 KST",
  "ID": "4910bf942804",
  "Image": "tomcat",
  "Labels": "org.op
encontainers.image.ref.name=ubuntu,org.opencontainers.image.version=22.04",
  "LocalVolumes": "0",
  "Mounts": "",
  "Names": "mytomcat4",
  "Networks": "bridge",
  "Ports": "8080/tcp",
  "RunningFor": "18 minutes ago",
  "Size": "37.4kB (virtual 429MB)",
  "State": "running",
  "Status": "Up 18 minutes"
}
```



```
C:\Users\bsyoo>docker ps --latest --format "{{json .}}"
```

```
{
  "Command": "\"/entrypoint.sh cata...\"",
  "CreatedAt": "2023-08-16 12:19:21 +0900 KST",
  "ID": "4910bf942804",
  "Image": "tomcat",
  "Labels": "org.op
encontainers.image.ref.name=ubuntu,org.opencontainers.image.version=22.04",
  "LocalVolumes": "0",
  "Mounts": "",
  "Names": "mytomcat4",
  "Networks": "bridge",
  "Ports": "8080/tcp",
  "RunningFor": "19 minutes ago",
  "Size": "37.4kB (virtual 429MB)",
  "State": "running",
  "Status": "Up 19 minutes"
}
```

▶ Docker 명령어

✓ container 상세정보 확인 명령어

👉 **docker container inspect container이름 || ID**
container 실행환경에 대한 상세한 정보를 조회
네트워크, 저장hash값, image 등 조회

명령어 : docker container inspect mytomcat1
전체정보를 json방식으로 조회

```
C:\Users\bsyoo>docker container inspect mytomcat1
[
  {
    "Id": "bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f",
    "Created": "2023-08-16T03:19:10.1236583Z",
    "Path": "/entrypoint.sh",
    "Args": [
      "catalina.sh",
      "run"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 1998,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-08-16T03:19:10.5418297Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:e46faef801b7e857475feb00ce71821fada4bc7847324439096cc67bea48ef6",
    "ResolvConfPath": "/var/lib/docker/containers/bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f/hostname",
    "HostsPath": "/var/lib/docker/containers/bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f/hosts",
    "LogPath": "/var/lib/docker/containers/bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f/bbc9653b5db5a9fab3f33606b271d2eeaed9dfaf196d39c13d18ec0e1171220f-json.log"
  }
]
```

▶ Docker명령어

✓ docker container stats 실습

실시간으로 container 사용현황 확인, 깜빡이면서 지속적으로 현황갱신

* ctrl+c로 빠져나올 수 있음

명령어 : docker container stats

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
4910bf942804	mytomcat4	0.21%	119.6MiB / 11.62GiB	1.01%	1.43kB / 0B	0B / 0B	40
aea8ca8bfe08	mytomcat3	0.21%	112.7MiB / 11.62GiB	0.95%	1.43kB / 0B	0B / 0B	40
5baf9fae9357	mytomcat2	0.20%	116.9MiB / 11.62GiB	0.98%	1.43kB / 0B	0B / 0B	40
bbc9653b5db5	mytomcat1	0.20%	118.7MiB / 11.62GiB	1.00%	1.61kB / 0B	0B / 0B	40

json방식으로 1회만 출력하기

명령어 : docker container stats --no-stream --format "{{json .}}"

```
C:\Users\bsyoo>docker container stats --no-stream --format "{{json .}}"
{"BlockIO":"0B / 0B","CPUPerc":"0.16%","Container":"4910bf942804","ID":"4910bf942804","MemPerc":"1.01%","MemUsage":"119.6MiB / 11.62GiB","Name":"mytomcat4","NetIO":"1.43kB / 0B","PIDs":"40"}
{"BlockIO":"0B / 0B","CPUPerc":"0.14%","Container":"aea8ca8bfe08","ID":"aea8ca8bfe08","MemPerc":"0.96%","MemUsage":"114.7MiB / 11.62GiB","Name":"mytomcat3","NetIO":"1.43kB / 0B","PIDs":"40"}
{"BlockIO":"0B / 0B","CPUPerc":"0.18%","Container":"5baf9fae9357","ID":"5baf9fae9357","MemPerc":"1.00%","MemUsage":"118.9MiB / 11.62GiB","Name":"mytomcat2","NetIO":"1.43kB / 0B","PIDs":"40"}
{"BlockIO":"0B / 0B","CPUPerc":"0.16%","Container":"bbc9653b5db5","ID":"bbc9653b5db5","MemPerc":"1.01%","MemUsage":"120.7MiB / 11.62GiB","Name":"mytomcat1","NetIO":"1.61kB / 0B","PIDs":"40"}
```

Container 조작 명령어

▶ Docker 명령어

✓ container 조작 명령어

👉 **docker container attach container이름||아이디**

실행 중인 container 보조운영체제(리눅스)에 표준입출력으로 접근하여 실행 중인 application의 설정 확인 및 변경

* 리눅스명령어로 확인

ubuntu컨테이너 백그라운드 실행 후 실습

명령어 : docker container run --name testubuntu -itd ubuntu

docker ps -al

docker container attach testubuntu

```
C:\Users\bsyoo>docker container run --name testubuntu -itd ubuntu
278c67711f7a95b80f51d138e56456f8dc48cc9f6d92c8371d3225b268943bff
```

```
C:\Users\bsyoo>docker ps -al
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
278c67711f7a	ubuntu	"/bin/bash"	8 seconds ago	Up 7 seconds		testubuntu

```
C:\Users\bsyoo>docker container attach testubuntu
```

```
root@278c67711f7a:/# |
```


▶ Docker명령어

✓ docker container attach 실습

ubuntu 각종 정보 확인 및 패키지 도구 설치하기

명령어 : docker container attach testubuntu

linux shell 모드에서 명령어를 입력하여 확인하기.

pwd 현재경로확인, whoami 접속한 사용자확인, ls 현재경로 파일조회

env 환경변수확인

```
C:\Users\bsyoo>docker container attach testubuntu
root@278c67711f7a:/# pwd
/
root@278c67711f7a:/# whoami
root
root@278c67711f7a:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@278c67711f7a:/# env
HOSTNAME=278c67711f7a
PWD=/
HOME=/root
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
TERM=xterm
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
_=/usr/bin/env
root@278c67711f7a:/#
```


▶ Docker명령어

✓ docker container attach 실습

tomcat은 입출력이 리눅스셸이 아니기 때문에 커서만 출력됨
tomcat container는 기본 log출력이 실행되기때문에 log 출력과 연결됨
* exec로 리눅스셸 프로세스로 접근이 가능하다.

명령어 : docker container run --name testtomcat -itd tomcat
docker container attach testtomcat

```
C:\Users\bsyoo>docker container run --name testtomcat -itd tomcat
40aefb8c0350829b32955cb7e0360e36249ccc0eb0b15ae2bf6e35bc893f229a

C:\Users\bsyoo>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
40aefb8c0350   tomcat    "/entrypoint.sh cata..." 6 seconds ago  Up 4 seconds  8080/tcp       testtomcat

C:\Users\bsyoo>docker container attach testtomcat
|
```

다른터미널에서 container를 종료하면 log가 출력되고 종료됨.

```
C:\Users\bsyoo>docker container attach testtomcat
29-Aug-2023 22:50:25.908 INFO [Thread-1] org.apache.coyote.AbstractProtocol.pause Pausing ProtocolHandle
r ["http-nio-8080"]
29-Aug-2023 22:50:25.915 INFO [Thread-1] org.apache.catalina.core.StandardService.stopInternal Stopping
service [Catalina]
29-Aug-2023 22:50:25.917 INFO [Thread-1] org.apache.coyote.AbstractProtocol.stop Stopping ProtocolHandle
r ["http-nio-8080"]
29-Aug-2023 22:50:25.921 INFO [Thread-1] org.apache.coyote.AbstractProtocol.destroy Destroying ProtocolH
andler ["http-nio-8080"]

C:\Users\bsyoo>

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bsyoo>docker container stop testtomcat
testtomcat

C:\Users\bsyoo>
```

▶ Docker 명령어

✓ container 조작 명령어

👉 **docker container exec 옵션 container식별자 명령**

docker container 내에 새로운 프로세스로 명령을 실행

셸열기, echo, touch, 환경변수설정 등 실행가능

이름	내용
--detach, d	명령을 백그라운드에서 실행
--interactive, -i	컨테이너의 표준입력을 열기
--tty, -t	tty단말기를 사용
--user, -u	명령실행할 계정을 선택

▶ Docker명령어

✓ docker container exec 실습

ubuntu echo명령 실행 *echo 인자값을 프롬프트에 출력

명령어 : docker container run --name -itd testubuntu

docker container exec testubuntu /bin/echo hello ubuntu

```
C:\Users\bsyoo>docker container run --name testubuntu -itd ubuntu  
b537f9f29b7610ed0ca37625bc6ad3a114174a0660a42234604ac16dc0f543e2
```

```
C:\Users\bsyoo>docker container exec testubuntu /bin/echo hello ubuntu  
hello ubuntu
```

ubuntu bash를 실행하여 container에서 linux명령어 실행하기

명령어 : docker container exec -it testubuntu /bin/bash

```
C:\Users\bsyoo>docker container exec -it testubuntu /bin/bash  
root@b537f9f29b76:/# pwd  
/  
root@b537f9f29b76:/# ls  
bin    dev    home   lib32  libx32  mnt    proc   run    srv    tmp    var  
boot   etc    lib    lib64  media   opt    root   sbin   sys    usr
```

Container 네트워크 옵션

▶ Docker 명령어

✓ container 네트워크연결 옵션

👉 docker container run network 옵션 이미지

도커 container 실행시 네트워크에 대한 설정을 하는 명령

* 기본값이 있어 필요한 내용만 설정, **-p만 옵션 사용하여 연결**

이름	내용
--add-host=[호스트명:IP]	컨테이너의 /etc/hosts에 호스트명과 IP주소를 설정
--dns=[IP]	컨테이너용 DNS서버의 IP주소 설정
--expose	지정한 범위의 포트번호를 할당
--mac-address=[mac주소]	컨테이너의 MAC주소 설정
--net=[bridge none container: 이름,ID 호스트 NETWORK]	네트워크지정
--publish,-p[host:container]	호스트와 컨테이너 포트매핑
--publish-all,-p	호스트의 임의의 포트를 컨테이너에 연결

▶ Docker명령어

✓ docker container 기본네트워크정보 확인

docontainer 내부 네트워크는 기본적으로 호스트와 게스트 네트워크를 bridge방식으로 연결이 하고, docker가 자동으로 ip, mac주소, 게이트 웨이 등을 설정 함.

- testubuntu container 네트워크 정보 확인하기

명령어 : docker container inspect testubuntu || docker inspect testubuntu

* Networks항목에서 실행된 컨테이너의 네트워크정보 확인 가능

```
C:\Users\bsyoo>docker inspect testubuntu
```

▶ Docker명령어

✓ docker container 기본네트워크정보 확인

- testubuntu container 네트워크 정보

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "eeb7bb543301d2fba0815844cd2aeb28ab38659693690bade20da7870731143c",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {},
  "SandboxKey": "/var/run/docker/netns/eeb7bb543301",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "6fd20afc37066e40d1e6b4322950a42e65b9c5d82b7d6bb8b7c5e2965197b218",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:02",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "0071ed5ec3dc270a746429eda747c817b718a77ead770f2f2047af7db621c2f9",
      "EndpointID": "6fd20afc37066e40d1e6b4322950a42e65b9c5d82b7d6bb8b7c5e2965197b218",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:02",
      "DriverOpts": null
    }
  }
}
```

▶ Docker명령어

✓ docker container 기본네트워크정보 확인

- 톰캣실행 네트워크확인, 톰캣이 사용하는 포트를 확인할 수 있음

명령어 : docker container run --name testtomcat -itd tomcat

docker container inspect testtomcat

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "2e616cf38551e9de1cc1c0388dba0f0723820680f4b5766b476f7cca3497ed6c",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "8080/tcp": null
  },
  "SandboxKey": "/var/run/docker/netns/2e616cf38551",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "177b557edef9d9ecdb77ca5a84b39f74b76d771931688e9b976a9fe1a151f25f",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:02",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "1ff52afbc1a55c76d4645317423a925e8e73a68a0f188b3aaa57fae2c0164641",
      "EndpointID": "177b557edef9d9ecdb77ca5a84b39f74b76d771931688e9b976a9fe1a151f25f",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:02",
      "DriverOpts": null
    }
  }
}
```


▶ Docker명령어

✓ docker container 네트워크설정 변경

- 톰캣실행 네트워크설정 변경하기 -> 호스트에서 docker에 접속하기 브리지방식으로 연결되어있어 -p옵션을 이용해서 호스트와 게스트에 포트 포워딩만 해주면 호스트에서 container서비스에접속할 수 있음

명령어 : docker container run --name nettomcat -d -p 9999:8080 tomcat
docker container ls

```
C:\Users\bsyoo>docker container run --name nettomcat -d -p 9999:8080 tomcat  
ffa8c8c4921e661e05bdc2ca206409bc890f049779366882256f8e0f6565da17
```

```
C:\Users\bsyoo>docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ffa8c8c4921e	tomcat	"/entrypoint.sh cata..."	32 seconds ago	Up 31 seconds	0.0.0.0:9999->8080/tcp	nettomcat

▶ Docker명령어

✓ docker container 네트워크설정 변경

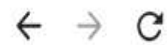
- 상세정보확인, 포트연결정보 확인

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "bf45968e81c3576580dd6a7bb0dd2ef879696bbcbff60ad1c0e849ac41d1d9fa",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "8080/tcp": [
      {
        "HostIp": "0.0.0.0",
        "HostPort": "9999"
      }
    ]
  },
  "SandboxKey": "/var/run/docker/netns/bf45968e81c3",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "16eac6d944666b2d00631a22e44c3e38af7164dc5925a52a9964543f8a323aa7",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.3",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:03",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "c18228a90ad2b608980b84fd81cc1e1a69fb19960f2abbedecce045cbd62032a",
      "EndpointID": "16eac6d944666b2d00631a22e44c3e38af7164dc5925a52a9964543f8a323aa7",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.3",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:03",
      "DriverOpts": null
    }
  }
}
```

▶ Docker명령어

✓ docker container 네트워크설정 변경

- 호스트에서 톰켓에 접속해 보기 * index페이지 없기때문에 404



localhost:9999



HTTP 상태 404 – 찾을 수 없음

타입 상태 보고

설명 Origin 서버가 대상 리소스를 위한 현재의 representation을 찾지 못했거나, 그것이 존재하는지를 밝히려 하지 않습니다.

Apache Tomcat/10.1.11

Container 추가설정하기

▶ Docker 명령어

✓ container 추가설정하기

👉 docker container run 추가설정옵션 이미지

container실행시 환경변수등록, 작업폴더 설정, 사용자설정 등 추가환경을 설정할 수 있다.

이름	내용
--env, -e	서브 리눅스에 환경변수를 등록 key=value로 설정, 여러개설정 가능
--env-file	지정파일에 있는 내용으로 환경변수 설정
--workdir	container 작업폴더 지정
--user, -u	사용자명, UID지정 사용자명은 서브 리눅스에 생성된 사용자명 사용해야함.

▶ Docker명령어

✓ docker container 추가정보옵션

- ubuntu컨테이너에 환경변수 설정하기

명령어 : docker container run --name extraubuntu -itd W
-e test=admin -e test2=1234 ubuntu

docker container exec extraubuntu env

제공 이미지에 환경변수설정이 필요할때 사용 * mysql 등

```
C:\Users\bsyoo>docker container exec extraubuntu env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=80171946b4f0
test=admin
test2=1234
HOME=/root
```

▶ Docker명령어

✓ docker container 추가설정옵션

- ubuntu컨테이너에 기본작업폴더 설정하기

명령어 : `docker container run -itd --name extraubuntu -w /test ubuntu`

`docker container exec -it extraubuntu /bin/bash`

리눅스내에서 `pwd` 실행

기본폴더가 `/test`로 지정됨.

- 기본실행 작업폴더

```
C:\Users\bsyoo>docker container exec -it extraubuntu /bin/bash
root@80171946b4f0:/# pwd
/
```

- 옵션적용 후 작업폴더

```
C:\Users\bsyoo>docker container run -itd --name extraubuntu -w /test ubuntu
72472a14a219e92a39835388181cd219a6552228e08674374d2cf11181dad858
C:\Users\bsyoo>docker container exec -it extraubuntu /bin/bash
root@72472a14a219:/test# pwd
/test
root@72472a14a219:/test# |
```

▶ Docker 명령어

✓ host, container 파일 복사하기

☞ `docker container cp` 복사될대상파일 복사할 위치

호스트 파일을 container에 복사하거나 container파일을 호스트에 저장하는 명령어

파일 작성 방법

호스트파일 : 호스트파일경로 예) `./test.txt || ./`

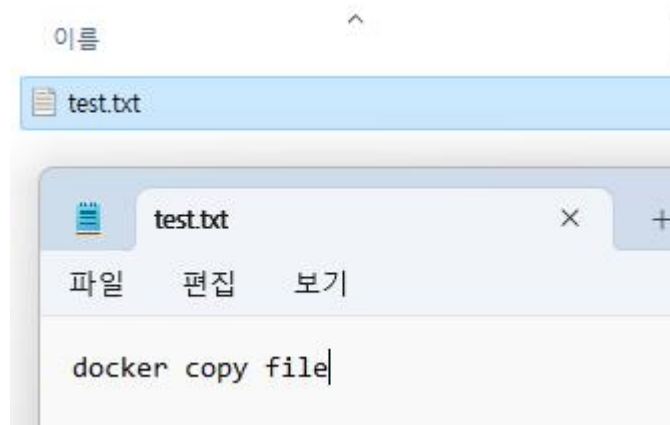
container파일 : container이름||아이디:파일경로

예) `extraubuntu:/test/test.txt || extraubuntu:/test/`

▶ Docker명령어

✓ docker container cp 실습

- 호스트파일 container에 복사하기
- 사용자계정/내문서/dockertest폴더를 생성 후 test.txt파일 생성
내용 docker copy file로 작성



```
C:\Users\bsyoo\dockertest>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 524B-8A7E

C:\Users\bsyoo\dockertest 디렉터리

2023-09-06 오후 05:33 <DIR> .
2023-09-06 오후 05:33 <DIR> ..
2023-09-06 오후 05:33          16 test.txt
                  1개 파일          16 바이트
                  2개 디렉터리 21,188,427,776 바이트 남음
```

▶ Docker명령어

✓ docker container cp 실습

- 호스트파일 컨테이너에 복사 명령어

명령어 : `docker container run -itd --name extraubuntu -w /test ubuntu`

`docker container cp ./test.txt extraubuntu:/test/`

`docker container exec extraubuntu ls /test`

`docker container exec extraubuntu cat /test/test.txt`

```
C:\Users\bsyoo\dockertest>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
72472a14a219   ubuntu   "/bin/bash"             23 minutes ago Up 23 minutes                               extraubuntu
ffa8c8c4921e   tomcat   "/entrypoint.sh cata..." 10 hours ago   Up 10 hours   0.0.0.0:9999->8080/tcp              nettomcat

C:\Users\bsyoo\dockertest>docker container cp ./test.txt extraubuntu:/test/

C:\Users\bsyoo\dockertest>docker container exec extraubuntu ls /test
test.txt

C:\Users\bsyoo\dockertest>docker container exec extraubuntu cat /test/test.txt
docker copy file
C:\Users\bsyoo\dockertest>
```

▶ Docker명령어

✓ docker container cp 실습

- container파일 host에 복사하기
- 폴더이동 : cmd에서 폴더 dockertest로 이동
- tomcat설정 파일 conf파일 복사하기
- tomcat 재실행하기

명령어 : docker container retart nettomcat * 실행 중이면 안해도 됨.

```
docker container cp nettomcat:/usr/local/tomcat/conf/
```

```
₩ ./tomcatconf
```

```
dir * 복사정보 확인
```

```
cd tomcatconf
```

```
dir * 폴더내 파일들 확인
```

▶ Docker명령어

✓ docker container cp 실습

```
C:\Users\bsyoo\dockertest>docker container cp nettomcat:/usr/local/tomcat/conf/ ./tomcatconf
```

```
C:\Users\bsyoo\dockertest>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 524B-8A7E
```

C:\Users\bsyoo\dockertest 디렉터리

```
2023-09-08 오전 07:38 <DIR> .
2023-09-06 오후 05:33 <DIR> ..
2023-09-06 오후 05:33          16 test.txt
2023-09-06 오전 07:49 <DIR>      tomcatconf
                1개 파일          16 바이트
                3개 디렉터리 20,275,519,488 바이트 남음
```

```
C:\Users\bsyoo\dockertest>cd tomcatconf
```

```
C:\Users\bsyoo\dockertest\tomcatconf>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 524B-8A7E
```

C:\Users\bsyoo\dockertest\tomcatconf 디렉터리

```
2023-09-06 오전 07:49 <DIR> .
2023-09-08 오전 07:38 <DIR> ..
2023-09-06 오전 07:49 <DIR> Catalina
2023-07-06 오후 10:45      12,926 catalina.policy
2023-07-06 오후 10:45      7,693 catalina.properties
2023-07-06 오후 10:45      1,411 context.xml
2023-07-06 오후 10:45      1,149 jaspic-providers.xml
2023-07-06 오후 10:45      2,313 jaspic-providers.xsd
2023-07-06 오후 10:45      4,144 logging.properties
2023-07-06 오후 10:45      6,978 server.xml
2023-07-06 오후 10:45      2,756 tomcat-users.xml
2023-07-06 오후 10:45      2,558 tomcat-users.xsd
2023-07-06 오후 10:45     172,391 web.xml
                10개 파일          214,319 바이트
                3개 디렉터리 20,254,031,872 바이트 남음
```

▶ Docker 명령어

✓ host, container 변경사항 확인하기

👉 **docker container diff container식별자**

기본이미지 설정내용에서 변경된 내용을 출력해주는 명령어

이름	내용
A	파일추가 표시
D	파일삭제표시
C	파일 수정표시

- extraubuntu컨테이너 실행 후 변경내용 확인하기

명령어 : docker container diff extraubuntu

```
C:\Users\bsyoo\dockertest\tomcatconf>docker container diff extraubuntu
C /root
A /root/.bash_history
A /test
A /test/test.txt
```

Container volume설정하기

▶ Docker Container

✓ docker volume

docker container는 한 개의 프로세스로 container에 저장된 데이터를 실행 중에만 유지되고 container를 삭제하면 그 설정 데이터는 삭제되기때문에 필요한 데이터를 host컴퓨터와 연결하여 유지할 수 있음 (백업)
container실행 초기 부터 host에 이미 설정한 내용을 불러와 사용할 수 있음

✓ volume선언하기

1. container 실행시 option을 이용해서 내부 volume이용하기
 - docker run -v host경로:container 경로 이미지명 * 자동으로 volume이 생성됨
2. volume생성 후 이용하기
 - docker volume create volume이름 * 경로가 자동으로 설정됨.

* volume은 host데이터를 우선하기 때문에 생성된 폴더에 연결할때 container의 설정데이터가 사라질 수 있으니 주의해야 함.

▶ Docker명령어

✓ docker volume create volume이름

- testvol생성하기 * wsl이 사용하는 내부 하드 위치에 저장됨.

* 저장위치 : C:\Users\사용자계정\AppData\Local\Docker\wsl\data

명령어 : docker volume create testvol

docker volume ls * 생성된 volume조회

docker volume inspect testvol * volume상세정보 확인

```
C:\Users\bsyoo>docker volume create testvol
testvol

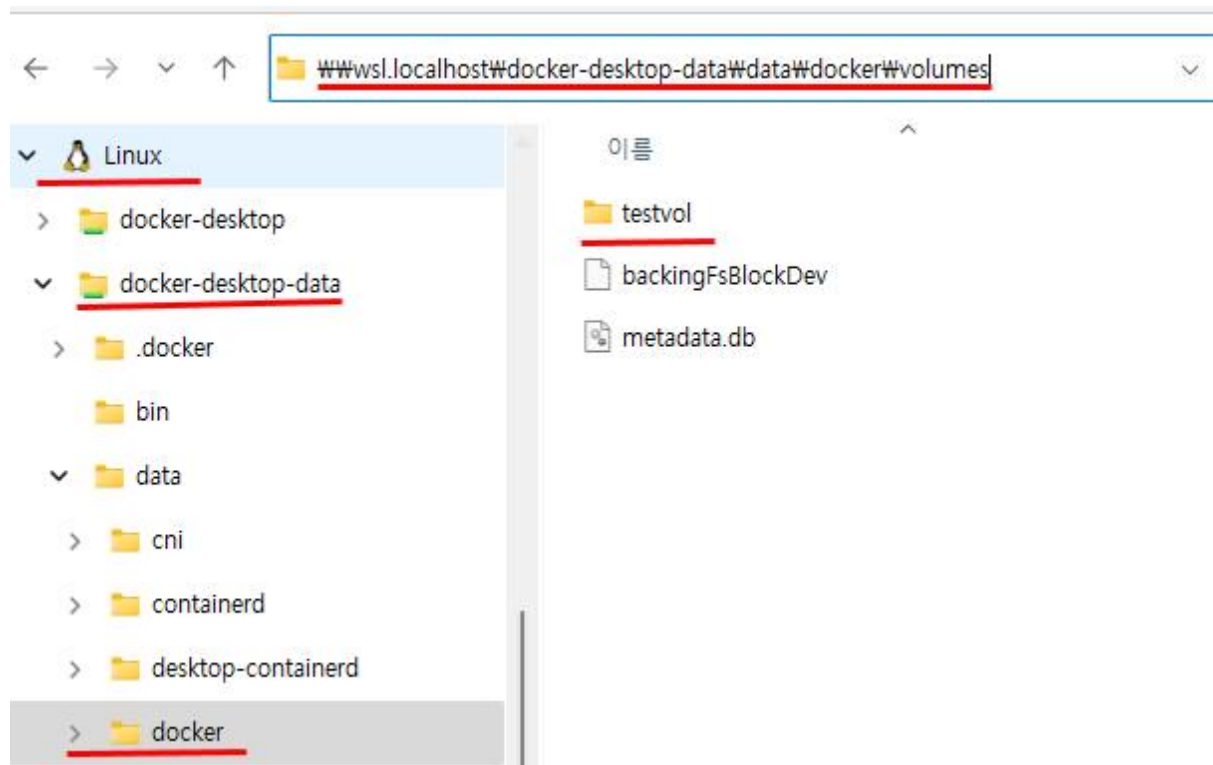
C:\Users\bsyoo>docker volume ls
DRIVER      VOLUME NAME
local       testvol

C:\Users\bsyoo>docker volume inspect testvol
[
  {
    "CreatedAt": "2023-09-10T23:31:29Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/testvol/_data",
    "Name": "testvol",
    "Options": {},
    "Scope": "local"
  }
]
```


▶ Docker명령어

✓ **docker volume create volume이름**

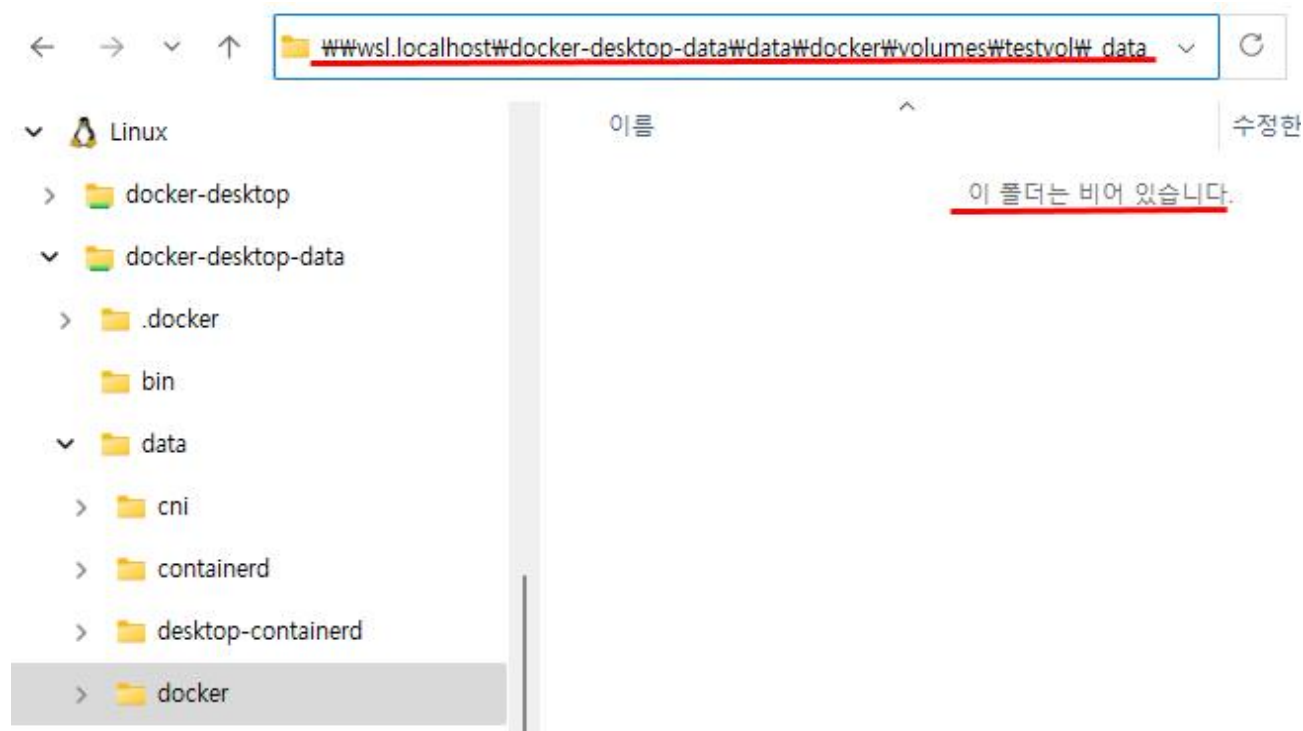
- 탐색기에서 volume폴더확인하기
- testvol폴더에 _data에 volume연결한 데이터 확인



▶ Docker명령어

✓ docker volume create volume이름

- 생성된 volume폴더 testvol는 container연결전까지 내용이 비어있음



▶ Docker명령어

✓ docker container run -v 옵션활용하기

- 도커 실행시 생성된 volume연결하기
- tomcat기본폴더를 생성한 testvolvolume과 연결하기

명령어 : docker run --name voltomcat -d -p 9090:8080 ₩
-v testvol : /usr/local/tomcat tomcat

```
C:\Users\bsyoo\dockertest\webtomcat>docker run --name voltomcat -d -p 9090:8080 -v testvol:/usr/local/tomcat tomcat  
a3cf8c3003bf34616f8d145845844db3e1d9018b56e8c0bd610a84102fb129df
```

```
C:\Users\bsyoo\dockertest\webtomcat>docker container ls
```

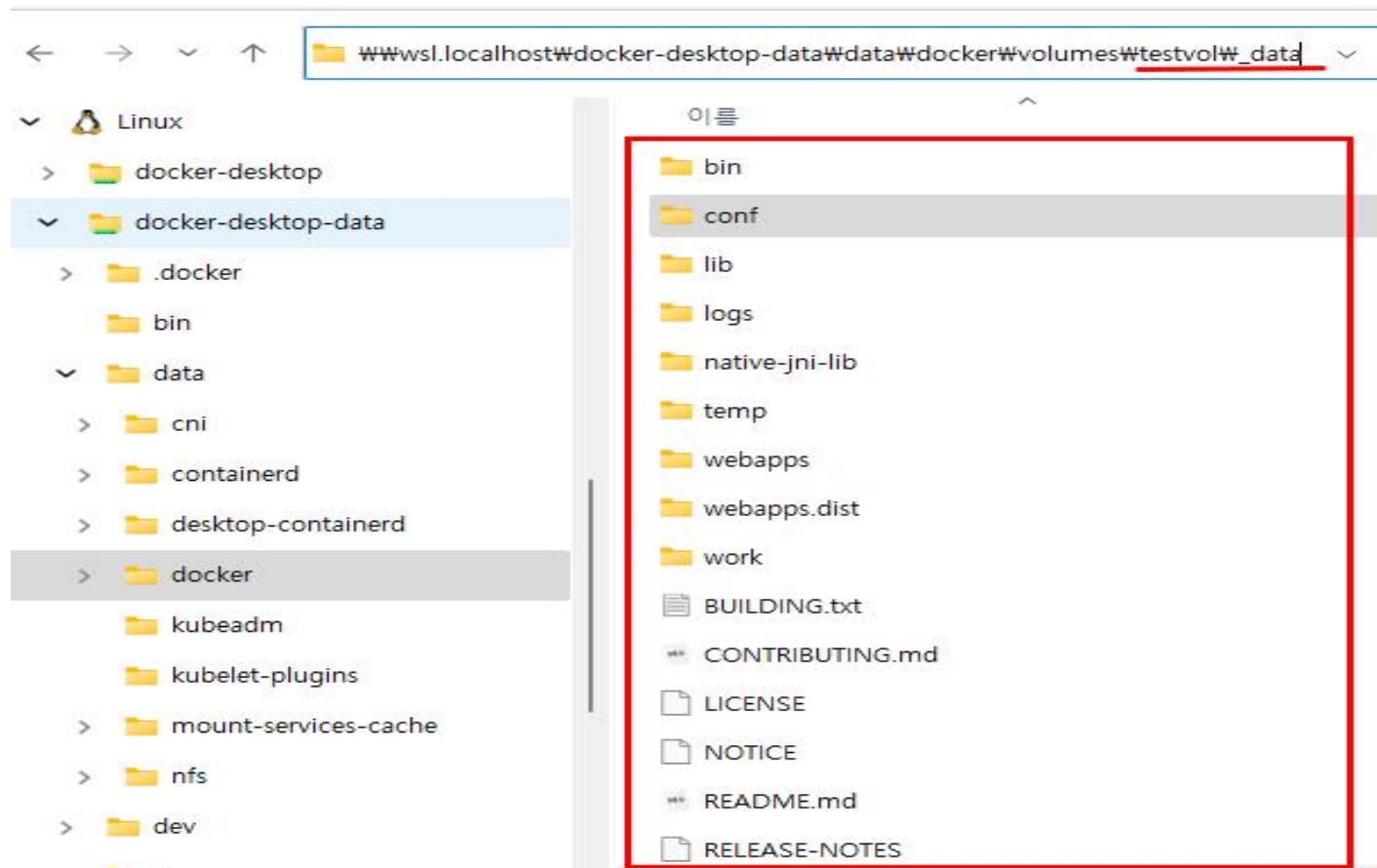
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a3cf8c3003bf	tomcat	"/entrypoint.sh cata..."	6 seconds ago	Up 5 seconds	0.0.0.0:9090->8080/tcp	voltomcat

```
C:\Users\bsyoo\dockertest\webtomcat>
```

▶ Docker명령어

✓ docker volume create volume이름

- volume연결 후 컨테이너의 데이터가 host에 복사되어 저장됨.
- 이 파일은 host와 container가 공유함.



▶ Docker명령어

✓ docker volume create volume이름

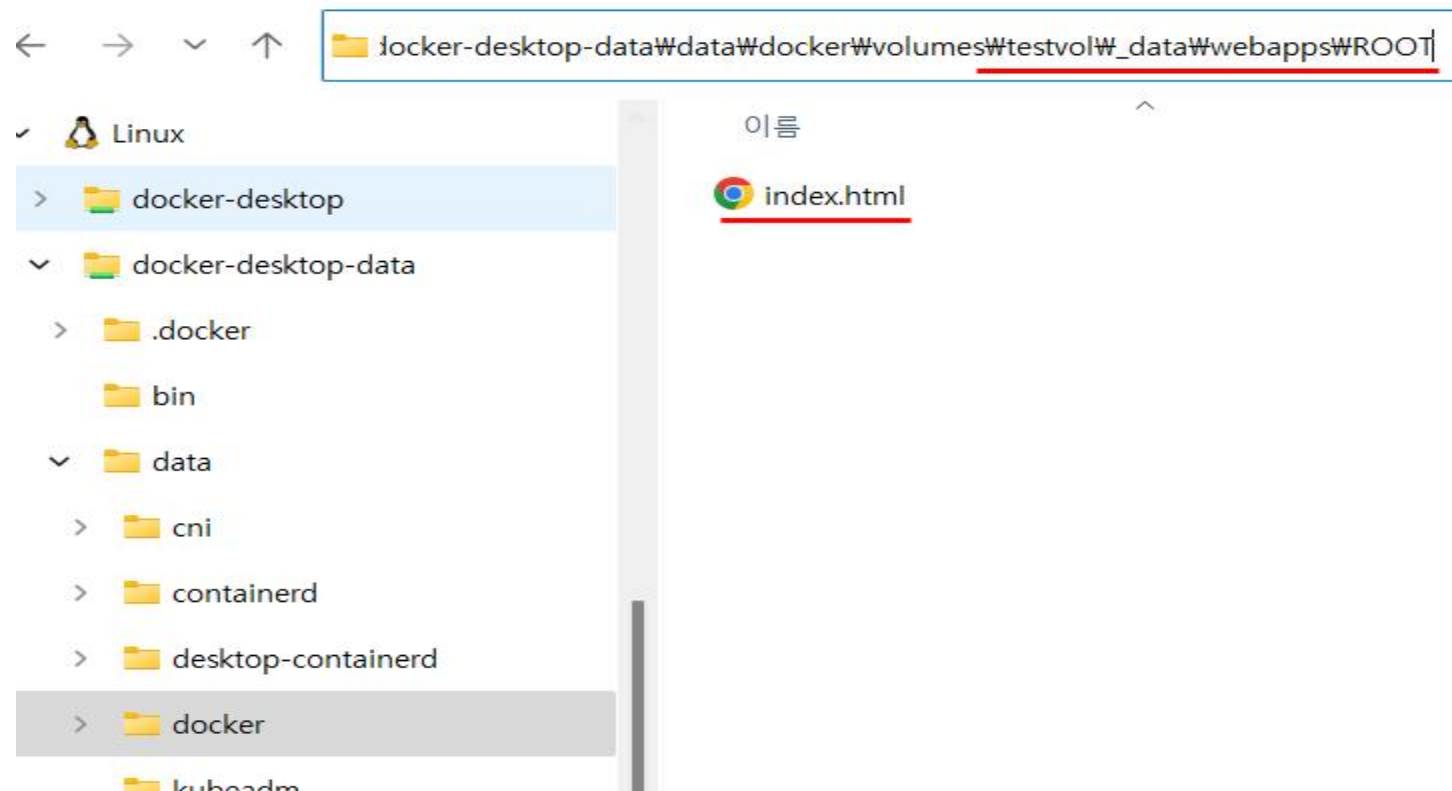
- volumefolder는 container와 공유하고 있어 내용을 변경하면 container에 변경내용이 반영이 됨.
- 브라우저에서 localhost:9090접속하기
 - >webapps폴더에 데이터가 없어서 404에러가 발생함



▶ Docker명령어

✓ docker volume create volume이름

- host의 volume폴더의 webapps에 ROOT폴더를 생성하고 index.html파일 추가



▶ Docker명령어

✓ docker volume create volume이름

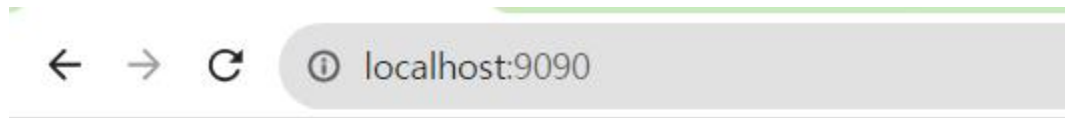
- index.html파일 내용

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>시작화면설정</title>
</head>
<body>
  <h2>기본 시작화면을 설정할 수 있다.</h2>
</body>
</html>
```

▶ Docker명령어

✓ **docker volume create volume이름**

- 폴더와 파일 추가 후 browser에서 접속



기본 시작화면을 설정할 수 있다.

▶ Docker명령어

✓ **docker volume create volume이름**

- 실행중인 컨테이너에 접속하여 파일을 수정하면 host와 공유하기 때문에 같이 수정됨.
- container(리눅스)에서 수정하기 위해 추가 패키지설치

명령어 : `docker container exec -it voltomcat /bin/bash`

`cd webapps/ROOT` *경로 이동

`vi index.html` * 문서편집기가 없어 실행 하지 못함.

`apt-get update` * ubuntu 패키지관리자 업데이트

`apt-get install -y vim` * 문서편집기 vim패키지 설치

▶ Docker명령어

✓ docker volume create volume이름

```
C:\Users\bsyoo>docker container exec -it voltomcat /bin/bash
root@94428913aaf5:/usr/local/tomcat# cd webapps
root@94428913aaf5:/usr/local/tomcat/webapps# ls
ROOT
root@94428913aaf5:/usr/local/tomcat/webapps# cd ROOT
root@94428913aaf5:/usr/local/tomcat/webapps/ROOT# ls
index.html
root@94428913aaf5:/usr/local/tomcat/webapps/ROOT# vi index.html
bash: vi: command not found
root@94428913aaf5:/usr/local/tomcat/webapps/ROOT# apt-get update
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://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [993 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
```

```
root@94428913aaf5:/usr/local/tomcat/webapps/ROOT# apt-get install -y vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libmpdec3 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libreadline8
  libsodium23 libsqlite3-0 media-types readline-common vim-common vim-runtime xxd
Suggested packages:
  gpm readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
  libgpm2 libmpdec3 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libreadline8
  libsodium23 libsqlite3-0 media-types readline-common vim vim-common vim-runtime xxd
0 upgraded, 14 newly installed, 0 to remove and 7 not upgraded.
Need to get 14.4 MB of archives.
After this operation, 60.7 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libmpdec3 amd64 2.5.1-2build2 [86.8 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpython3.10-minimal amd64 3.10.12-1~22.04.2 [811 kB]
```

▶ Docker명령어

✓ docker volume create volume이름

- 패키지설치 후 indxe.html에 접속하여 수정

명령어 : vim index.html

- * vim 문서편집기 수정 i를 누르고 수정할 위치에 가서 수정 후
esc누르고, : 누른 후 wq enter(수정내용저장 후 나가기)

```
root@94428913aaf5:/usr/local/tomcat/webapps/ROOT# vim index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>시작화면설정</title>
</head>
<body>
  <h2>기본 시작화면을 설정할 수 있다.</h2>
  <h3>서버에서 수정한 내용</h3>
</body>
</html>
:wq|
```

▶ Docker명령어

✓ docker volume create volume이름

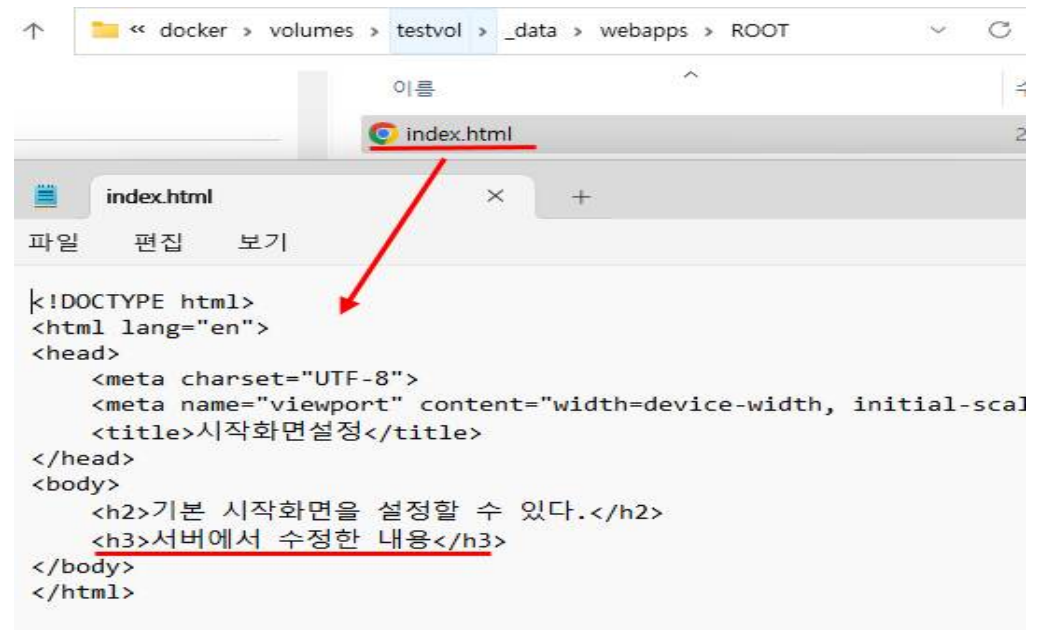
- 수정 후 브라우저에서 접속하기

← → ↻ ⓘ localhost:9090

기본 시작화면을 설정할 수 있다.

서버에서 수정한 내용

- host volume폴더에서
변경내용확인



▶ Docker명령어

✓ 생성된 volume 다른 container와 공유하기

- voltomcat에서 사용하는 volume을 공유하기때문에 설정한 내용이 동일하게 적용됨

```
C:\Users\bsyoo>docker run --name voltomcat2 -d -p 9091:8080 -v testvol:/usr/local/tomcat/ tomcat  
a5ecc6e82f14d3d91b696e1de8807f213cfc78c75884f9e391e4044f811fff3a
```

```
C:\Users\bsyoo>
```

시작화면설정

localhost:9091

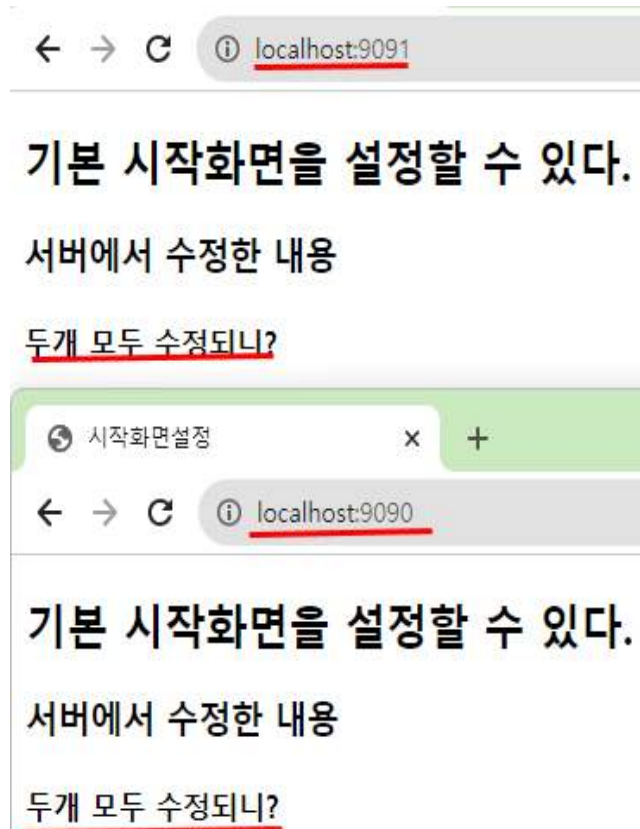
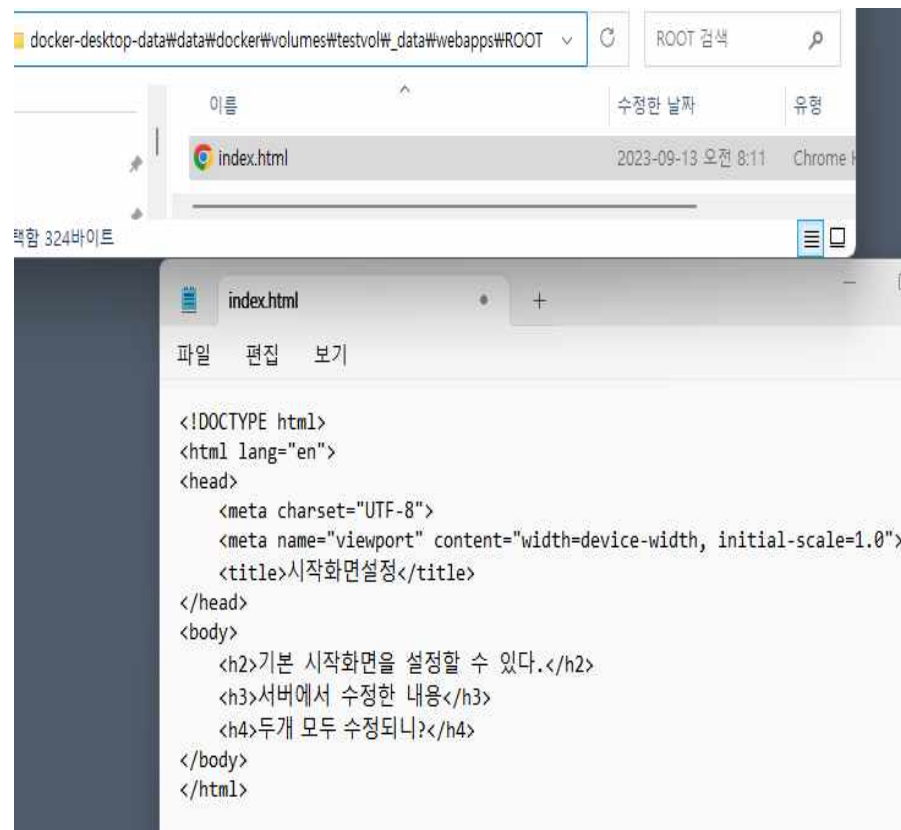
기본 시작화면을 설정할 수 있다.

서버에서 수정한 내용

▶ Docker명령어

✓ 생성된 volume 다른 container와 공유하기

- host에서 index를 수정하면 두 container에 모두 반영이 된다.



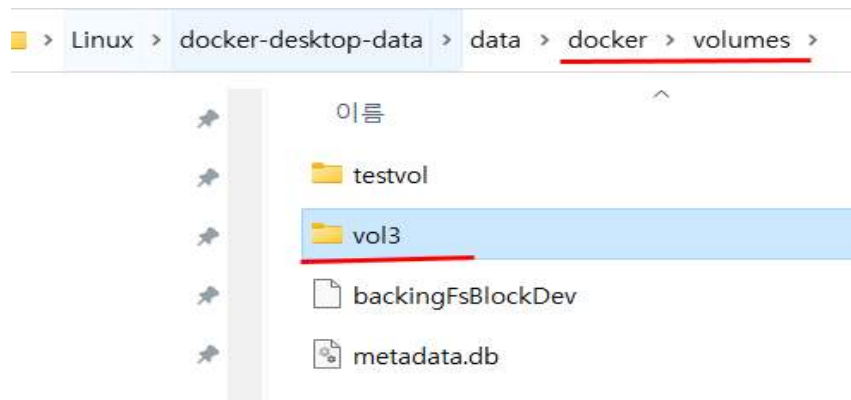
▶ Docker명령어

✓ docker volume폴더 지정해서 생성하기

- container생성시 -v host폴더/container폴더 원하는 폴더로 지정
- 절대경로를 설정하면 지정할 폴더로, 상대경로로 설정하면 volume폴더로 설정됨.

명령어 : `docker run--name voltomcat3 -d -v ./vol3 : ₩
/usr/local/tomcat/webapps -p 9092:8080 tomcat`

```
C:\Users\bsyoo>docker run --name voltomcat3 -d -v ./vol3:/usr/local/tomcat/webapps -p 9092:8080 tomcat  
247900642db33665a85f61ae3fe6a78146ea9066458da4ef7744e05189070108
```



▶ Docker명령어

✓ **docker volume create volume이름**

- 절대경로로 설정하면 원하는 지정한 폴더에 volume폴더가 생성됨.

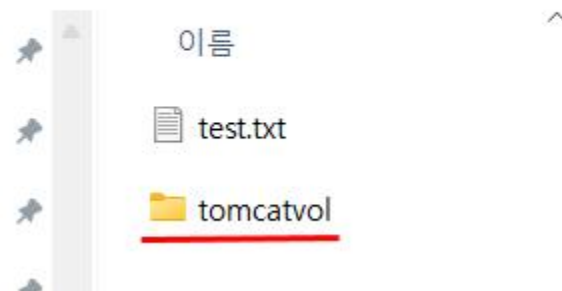
명령어 : `docker run --name voltomcat4 -d ₩`

`-v c:/Users/bsyoo/tomcatvol : /usr/local/tomcat/webapps`

`-p 9093:8080 tomcat`

```
C:\Users\bsyoo>docker run --name voltomcat4 -d -v c:/Users/bsyoo/dockertest/tomcatvol:/usr/local/tomcat/webapps -p 9093:8080 tomcat
2201b031b4fdc1ba115fbae76bd183362c4404315ec2632bb56992930bc7554b
```

C:\Users\bsyoo\dockertest



Container 변환하기

▶ Docker Container

✓ docker container 이미지로 저장

실행중인 container의 내용을 이미지로 따로 반환하는 기능

실행중인 container의 설정, 추가 파일 등을 포함한 이미지를 생성할때 사용

* volume으로 연결된 내용은 반영이 안됨.

명령어 : docker container commit container식별자 이미지명:태그명

✓ docker container commit 활용하기

- tomcat이미지를 실행하고 실행 container에 vim패키지설치
- 패키지가 설치된 container를 commit을 이용해서 image로 생성하기
- 생성 후 생성된 이미지로 실행
- 패키지가 있는지 확인

▶ Docker명령어

✓ docker container commit 활용하기

명령어 : `docker run --name testcom -d -p 9090:8080 tomcat`

`docker container exec -it testcom /bin/bash`

(리눅스) `apt-get update`

(리눅스) `apt-get install -y vim`

```
C:\Users\bsyoo>docker run --name testcom -d -p 9090:8080 tomcat
633af4e260b54628edd0a4c5e5d5957cd894685d490e7fb9d1185add669a8222
```

```
C:\Users\bsyoo>docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
633af4e260b5	tomcat	"/entrypoint.sh cata..."	4 seconds ago	Up 2 seconds	0.0.0.0:9090->8080/tcp	testcom

```
C:\Users\bsyoo\dockertest>docker container exec -it testcom /bin/bash
```

```
root@633af4e260b5:/usr/local/tomcat# apt-get update
```

```
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
```

```
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
```

```
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [966 kB]
```

```
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
```

```
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
```

▶ Docker명령어

✓ docker container commit 활용하기

```
root@633af4e260b5:/usr/local/tomcat# apt-get install -y vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libmpdec3 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libreadline8 libsodium23 libsqlite3-0
  media-types readline-common vim-common vim-runtime xxd
Suggested packages:
```

- 패키지를 추가한 container를 commit으로 image만들기

명령어 : docker container commit testcom bsyoo/tomcat:1.0

```
C:\Users\bsyoo\dockertest>docker container commit testcom bsyoo/tomcat:1.0
sha256:232d685989e460e28b19e5f7fd23c5de8c9e547426a7828fbc9ff42caf0951dc
```

```
C:\Users\bsyoo\dockertest>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<u>bsyoo/tomcat</u>	<u>1.0</u>	<u>232d685989e4</u>	<u>6 seconds ago</u>	<u>532MB</u>

▶ Docker명령어

✓ docker container commit 활용하기

- 생성된 이미지로 container실행하고 패키지설치 확인하기

명령어 : `docker run --name bstomcat -d -p 9091:8080 bsyoo/tomcat:1.0`

`docker container exec -it bstomcat /bin/bash`

(리눅스) `dpkg -l | grep vim` * vim편집기 설치 확인

```
C:\Users\bsyoo>docker run --name bstomcat -d -p 9091:8080 bsyoo/tomcat:1.0
b8a686eacd3f00a7d53e696ac54c49b71b87d9ed06e7e12aa460f7d3f297b980
```

```
C:\Users\bsyoo>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b8a686eacd3f	bsyoo/tomcat:1.0	"/entrypoint.sh cata..."	7 seconds ago	Up 6 seconds	0.0.0.0:9091->8080/tcp	bstomcat

```
C:\Users\bsyoo>docker container exec -it bstomcat /bin/bash
```

```
root@b8a686eacd3f:/usr/local/tomcat# dpkg -l | grep vim
```

ii	vim	2:8.2.3995-1ubuntu2.11	amd64	Vi IMproved - enhanced vi editor
ii	vim-common	2:8.2.3995-1ubuntu2.11	all	Vi IMproved - Common files
ii	vim-runtime	2:8.2.3995-1ubuntu2.11	all	Vi IMproved - Runtime files