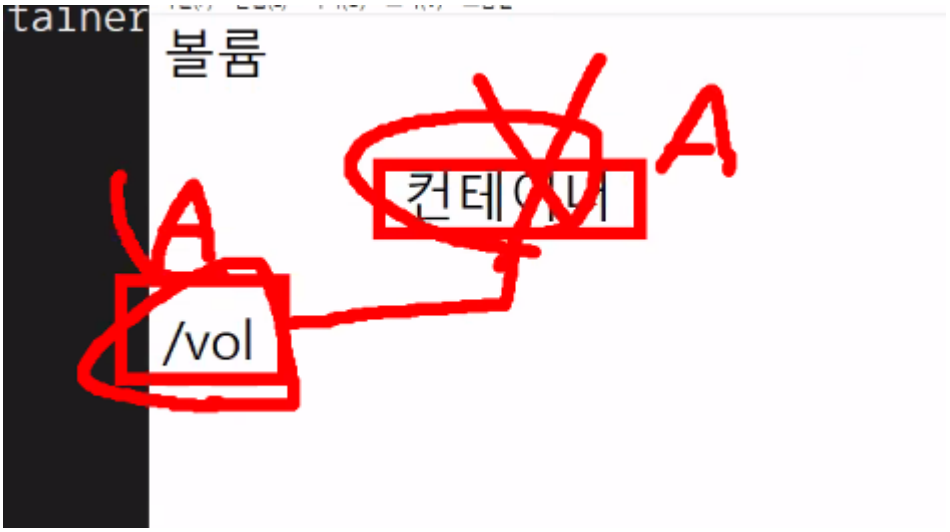
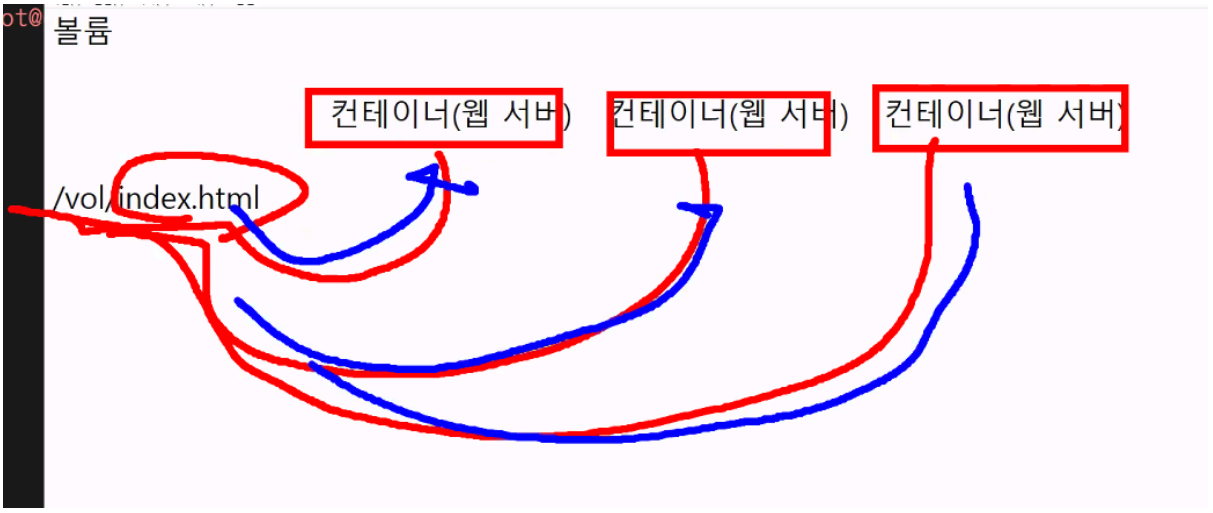
day3
docker volume, network



- 컨테이너의 하드디스크
- 컨테이너를 지워도 볼륨에 유지된다.
- 다시 컨테이너를 만들면 볼륨에서 백업가능



- 작업한 파일들을 볼륨에다 넣어놓으면 여러 컨테이너에 적용이 가능하다.
- 여러 웹 서버를 돌릴때 유용하다.

볼륨의 종류

- volume 방식

  - volume dir에 연결해서 사용하는 방식

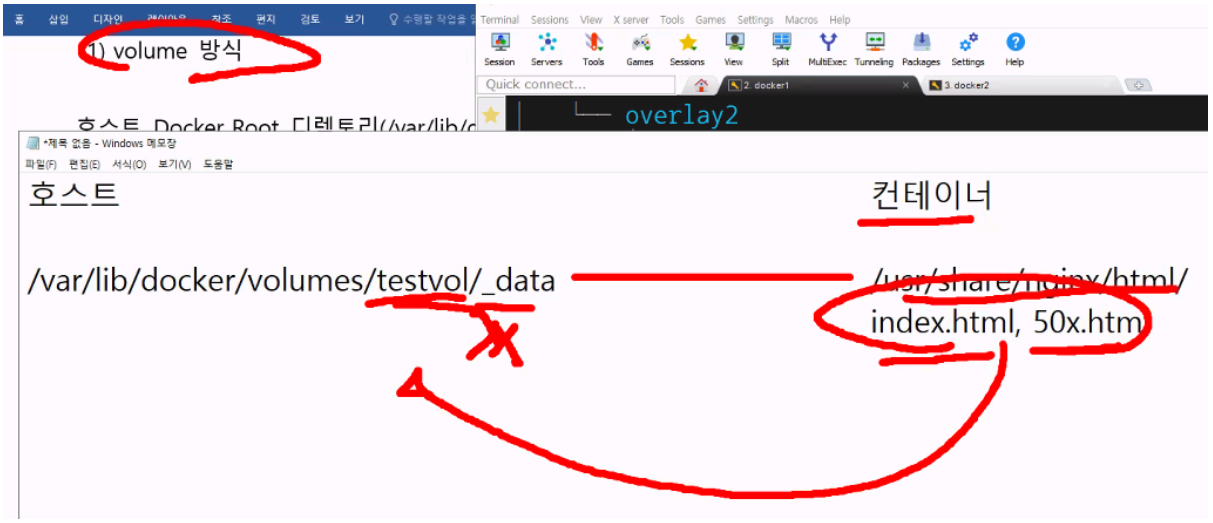
- bind mount 방식
    - 아무데나 연결해서 만드는 방식

volume

```
# docker container run -d --name myweb -v testvol:/usr/share/nginx/html -p 8080:80 nginx
```

```
# docker container run -d --name myweb -v /usr/share/nginx/html -p 8080:80 nginx
```

자동으로 생성된 볼륨은 이름이 해시값으로 길게 나오기 때문에 권장하지 않는다.



- 볼륨 연결을 하면 호스트에서 컨테이너에 있는 중요파일을 관리한다.
- 호스트에서 고치면 컨테이너도 고쳐진다. (동기화)

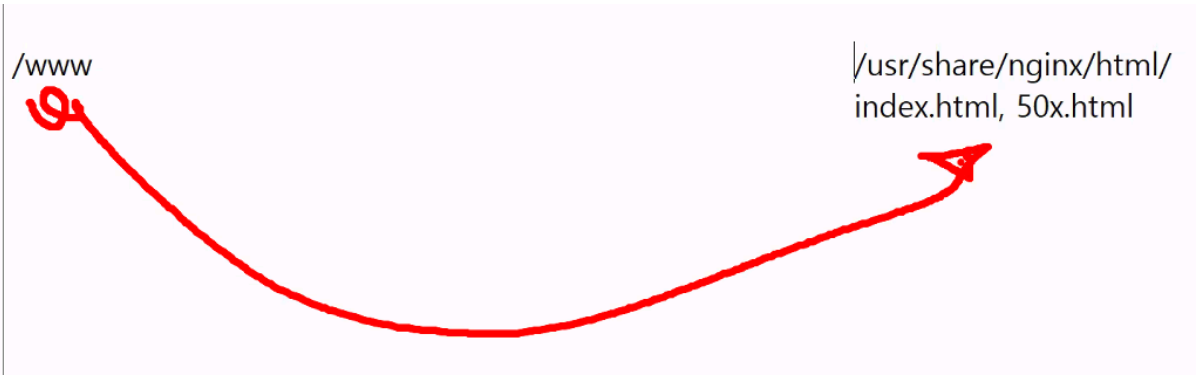
bind-mount

2) bind mounts 방식

호스트의 특정 디렉토리를 마운트하여 컨테이너 데이터를 저장하는 방식이다.
다음 예제는 '/www' 디렉토리와 컨테이너의 '/usr/share/nginx/html' 디렉토리를 bind mounts 방식으로 구성한 것이다. 이때, 볼륨으로 지정한 '/www' 디렉토리는 자동으로 생성된다.

```
# docker container run -d --name myweb -v /www:/usr/share/nginx/html -p 8080:80 nginx
```

컨테이너 '/usr/share/nginx/html' 디렉토리는 무조건 '/www' 디렉토리에 동기화된다.



- 컨테이너에서 가져오지않고 무조건 리눅스에서 들고온다.
- 따라서 파일들이 없어질수도 있다.

Mysql 설치 (docker1):
docker container run -d --name mydb -v mysqlvol:/var/lib/mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=password mysql:5.7

8장 docker network

3) 명령어 사용 예시

- 컨테이너 포트 매핑

```
# docker container run -d -p 8080:80 nginx
```

- 컨테이너 DNS 서버 지정

```
# docker container run -d --dns 8.8.8.8 nginx
```

- 컨테이너 MAC 주소 지정

```
# docker container run -it --mac-address="00:00:00:11:11:11" centos
```

- 컨테이너 '/etc/hosts' 파일 설정

```
# docker container run -it --add-host docker1:192.168.2.10 centos
```

- 컨테이너 호스트명 파일 설정

```
# docker container run -it --hostname webserver centos
```

bridge
- 여러개의 컨테이너를 같은 네트워크 사용할때

host
- 컨테이너가 이 네트워크장치를 그대로 쓸때

none
- loopback만 존재