

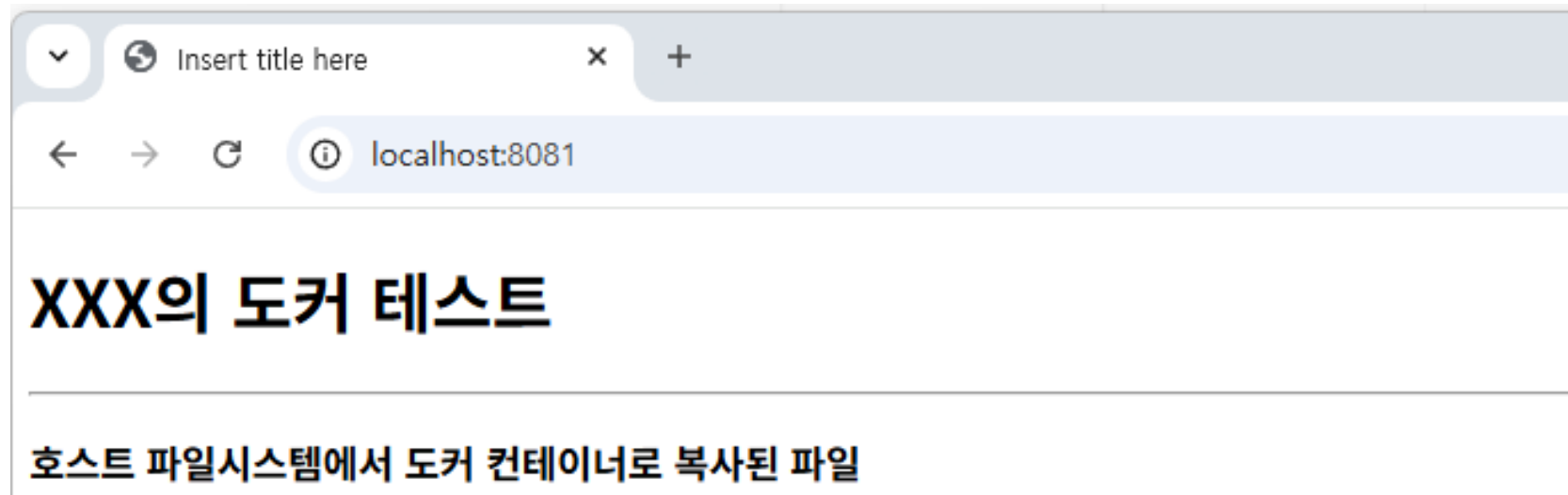
그림 6-2-1 호스트와 컨테이너 간에 파일 복사

경로의 예

항목	값
문서 폴더(윈도우)	C:\Users\사용자명\Documents\파일명
문서 폴더(macOS)	/Users/사용자명/Documents/파일명
리눅스의 홈 디렉터리	/home/사용자명/파일명



컨테이너에 문서 띄우기.txt 참고



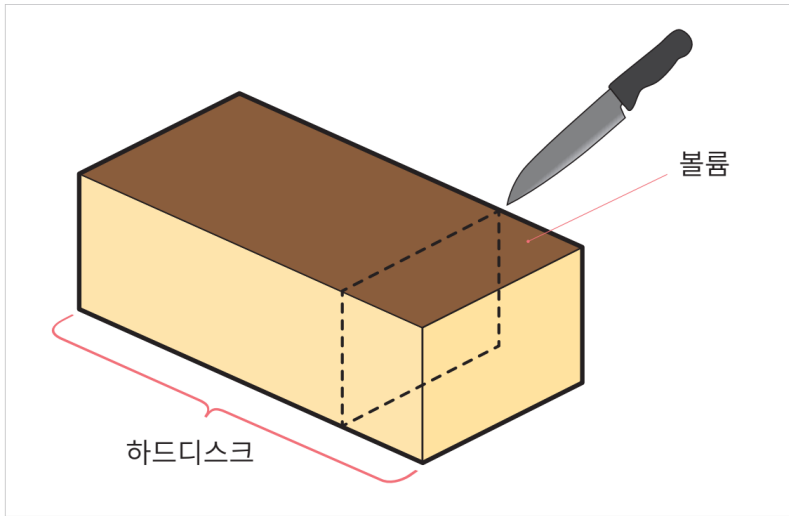


그림 6-3-1 볼륨

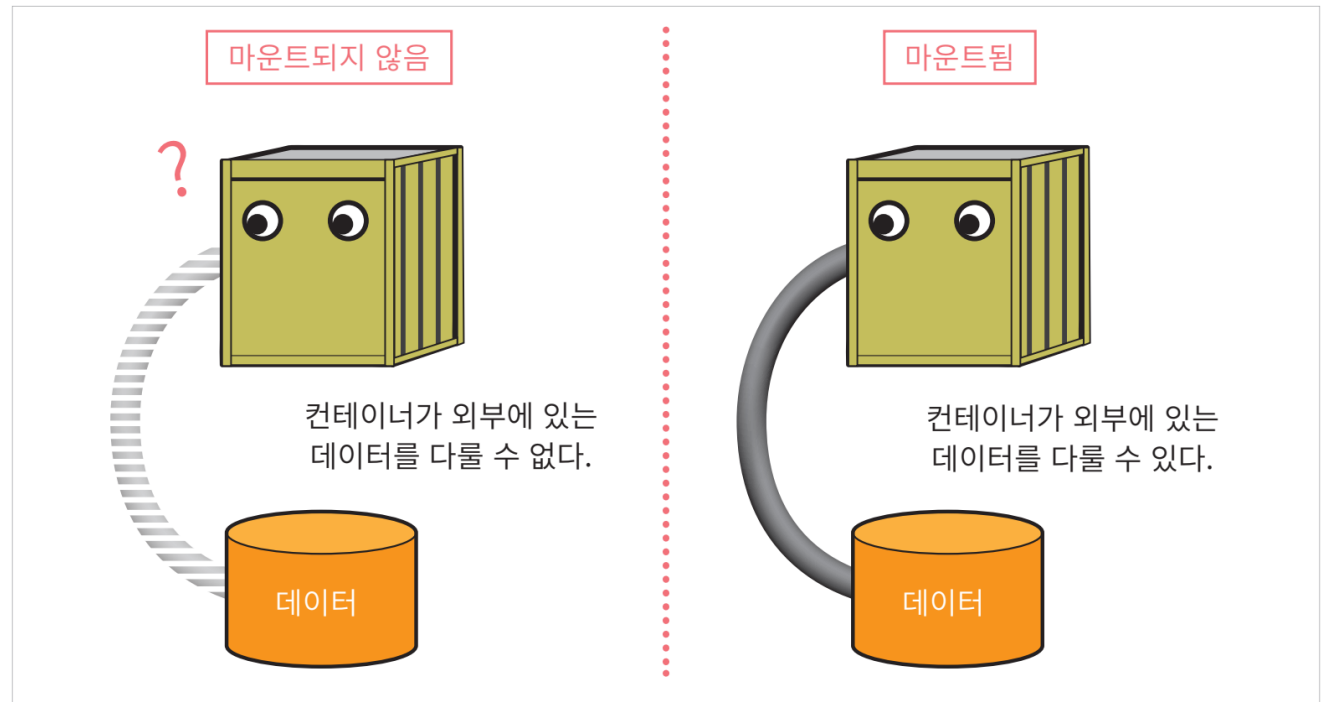


그림 6-3-2 마운트를 하면 이렇게 된다.

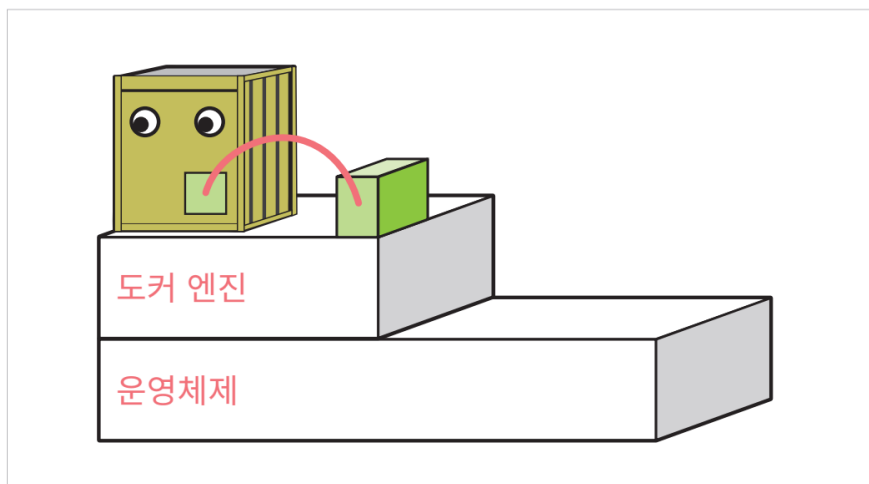


그림 6-3-3 볼륨 마운트

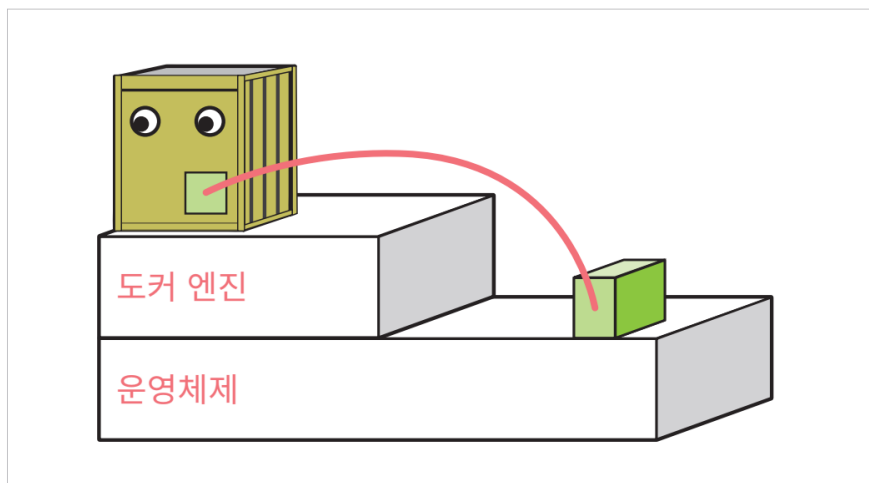
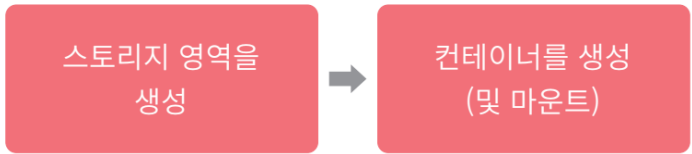


그림 6-3-4 바인드 마운트

볼륨 마운트와 바인드 마운트의 차이

항목	볼륨 마운트	바인드 마운트
스토리지 영역	볼륨	디렉터리 또는 파일
물리적 위치	도커 엔진의 관리 영역	어디든지 가능
마운트 절차	볼륨을 생성한 후 마운트	기존 파일 또는 폴더를 마운트
내용 편집	도커 컨테이너를 통해서	일반적인 파일과 같이
백업	절차가 복잡함	일반적인 파일과 같이



주요 하위 커맨드

커맨드	내용	생략형	주요 옵션
create	볼륨을 생성	X	거의 사용하지 않음
inspect	볼륨의 상세 정보를 출력	X	거의 사용하지 않음
ls	볼륨의 목록을 출력	X	거의 사용하지 않음
prune	현재 마운트되지 않은 볼륨을 모두 삭제	X	거의 사용하지 않음
rm	지정한 볼륨을 삭제	X	거의 사용하지 않음

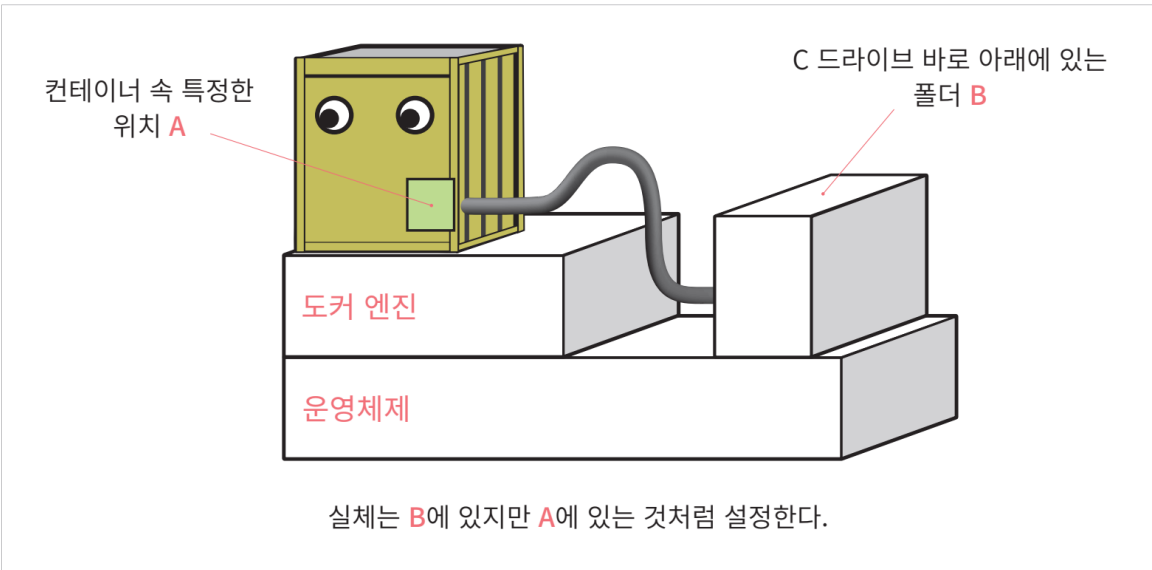


그림 6-3-5 마운트는 단축 아이콘과 같다(바인드 마운트)



항목	값
컨테이너 이름	apa00ex20
이미지 이름	httpd
포트 번호 설정	8090

항목	값
컨테이너 마운트 경로 (마운트 대상)	/usr/local/apache2/htdocs
실제 폴더 / 디렉터리 이름 (마운트 원본)	apa_folder
실제 마운트 원본 경로(윈도우)	C:\Users\사용자명\Documents\apa_folder
실제 마운트 원본 경로(macOS)	/Users/사용자명/Documents/apa_folder
실제 마운트 원본 경로(리눅스)	/home/사용자명/apa_folder

자신의 운영체제에 맞는 명령어를 선택



그림 6-3-6 폴더만 있을 때의 아파치 초기 화면

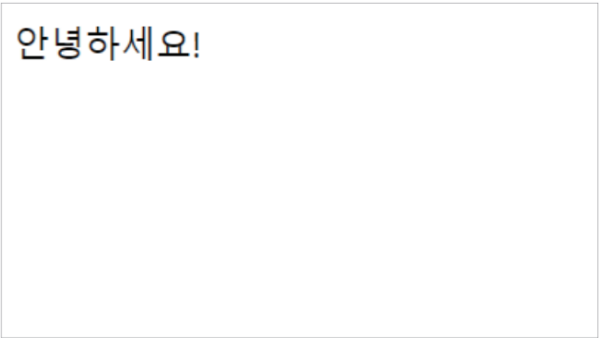
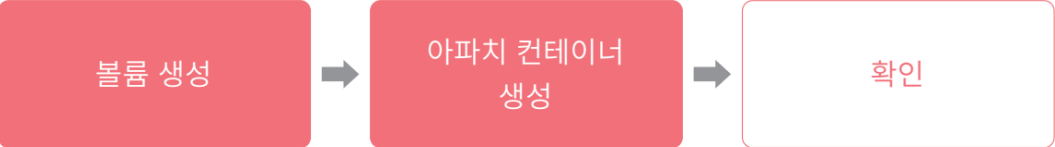


그림 6-3-7 index.html 파일의 내용이 출력된다.



항목	값
컨테이너 이름	apa000ex21
이미지 이름	httpd
볼륨 이름	apa000vol1(마지막은 소문자 L 뒤에 숫자 1이다)
포트 번호 설정	8091

마운트 원본과 마운트 대상 정보

항목	값
컨테이너 마운트 경로	/usr/local/apache2/htdocs
마운트 원본 볼륨	apa000vol1

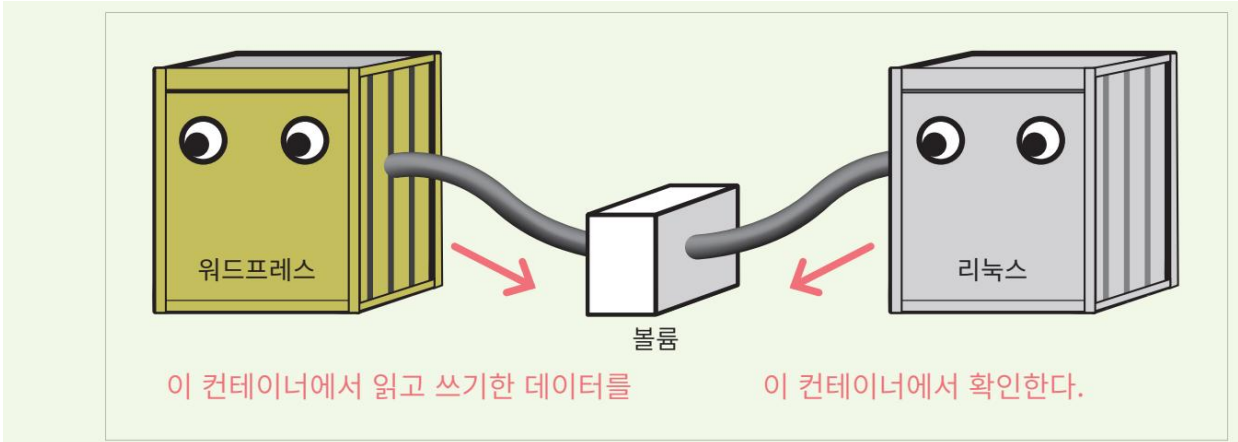


그림 6-3-8 별도의 컨테이너에서 볼륨의 내용 확인하기

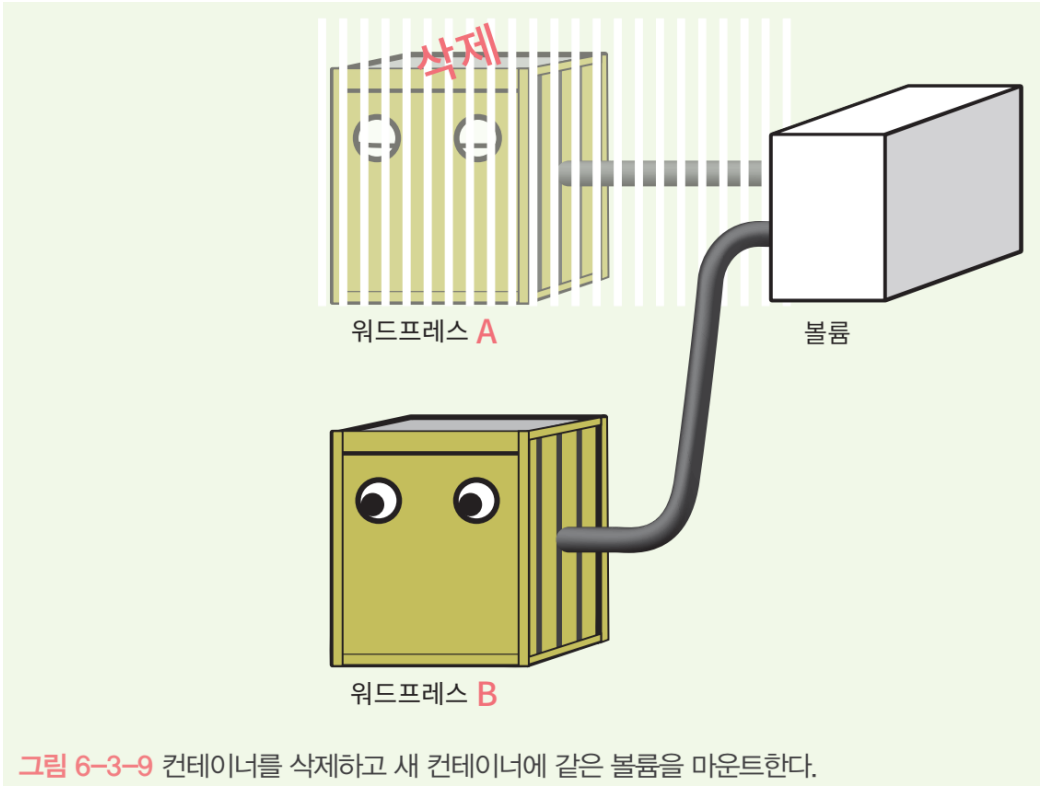


그림 6-3-9 컨테이너를 삭제하고 새 컨테이너에 같은 볼륨을 마운트한다.

주 컨테이너의 종료를 확인



리눅스 운영체제 컨테이너를
만들고 tar 명령어를 사용해
백업을 수행

항목	값
볼륨 이름	apa000vol1
백업 저장 폴더 이름	C:\Users\사용자명\Documents
busybox 컨테이너의 백업 원본 디렉터리 이름	/source
busybox 컨테이너의 백업 저장 디렉터리 이름	/target
백업파일명	backup_apa

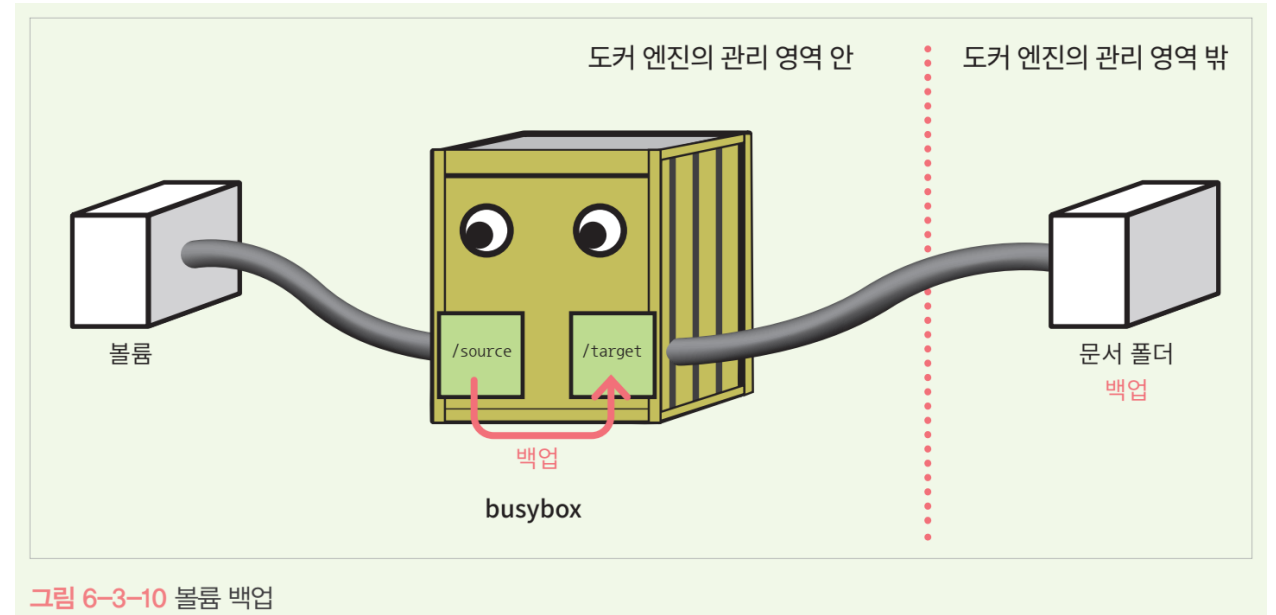


그림 6-3-10 볼륨 백업

```
docker run --rm -v apa000vol1:/source -v C:\Users\mit-305\docker:/target busybox tar czvf /target/backup_apa.tar.gz -C /source .
```

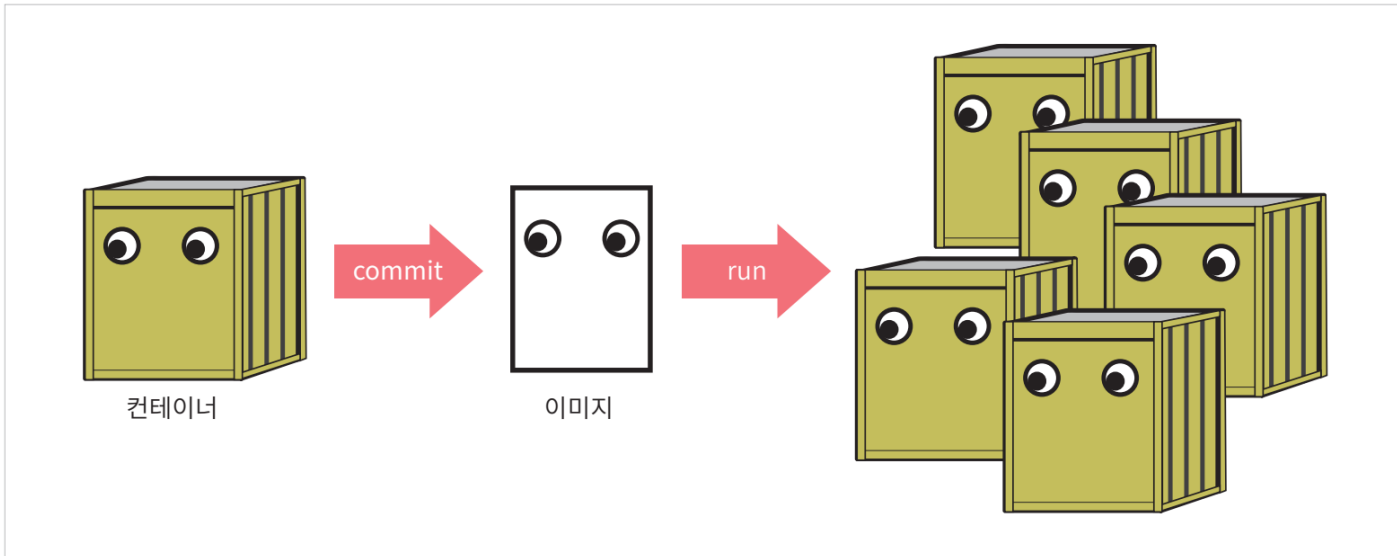



그림 6-4-1 똑같은 컨테이너를 대량으로 만들기

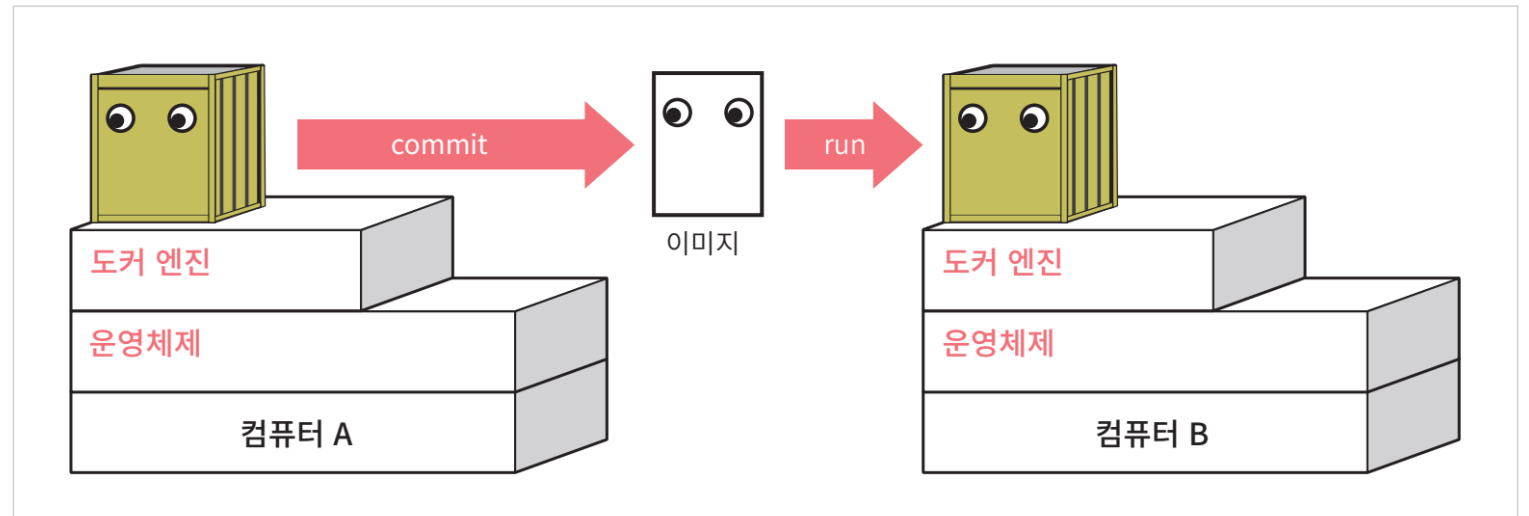


그림 6-4-2 컨테이너를 이동할 수도 있다.

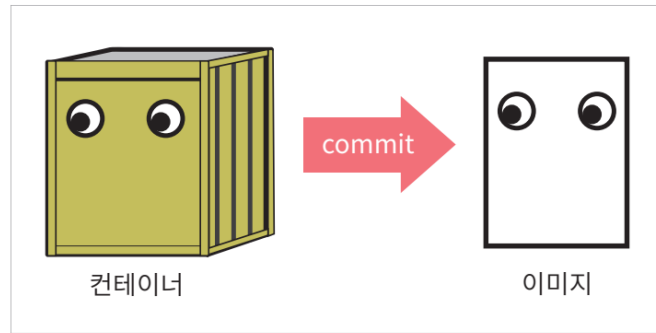


그림 6-4-3 컨테이너를 이미지로 변환

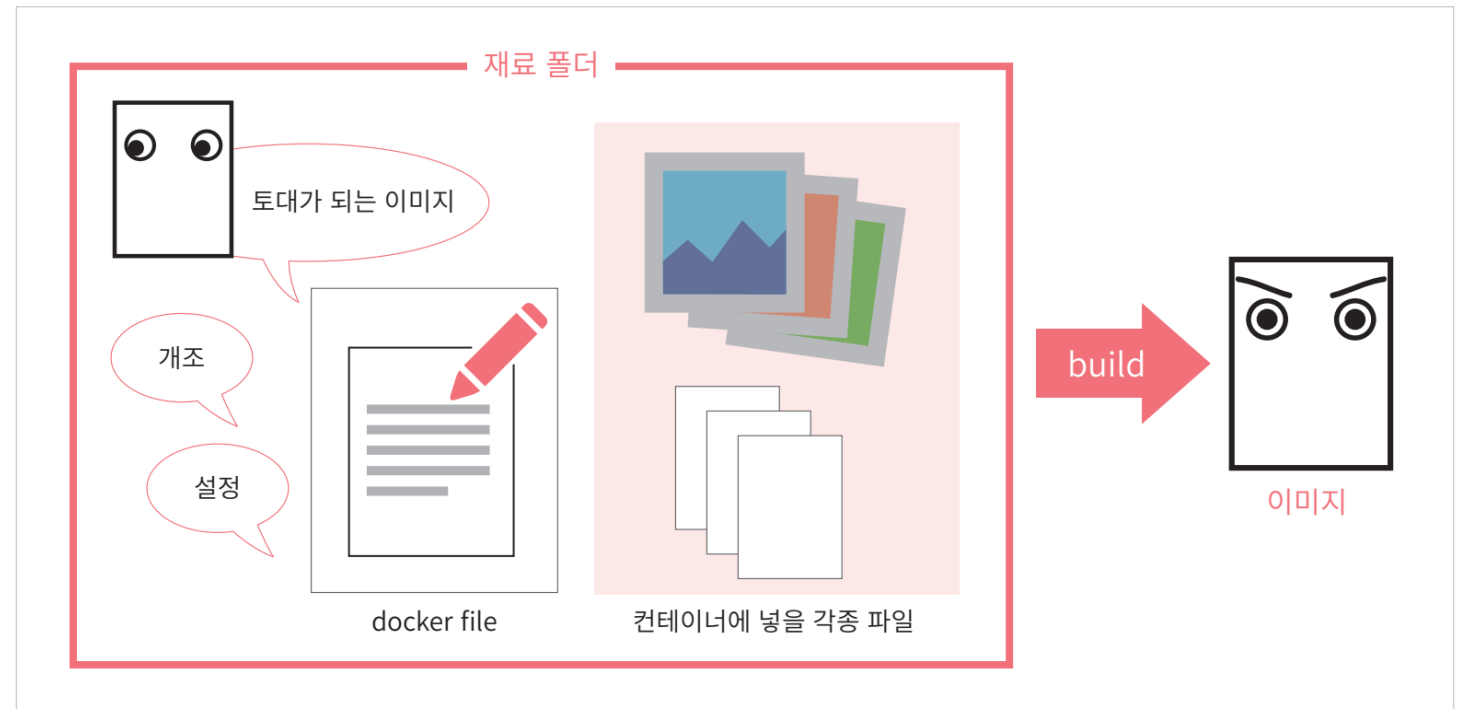
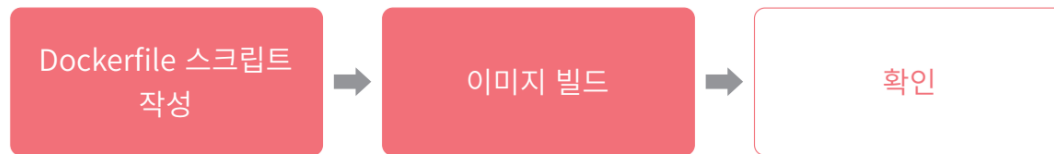
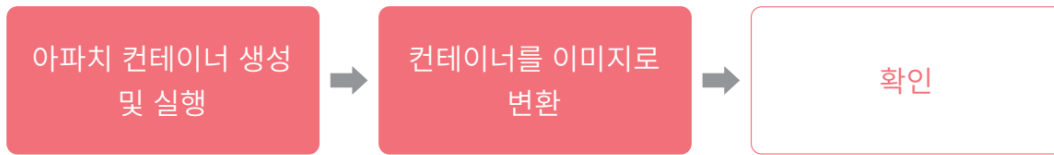


그림 6-4-4 Dockerfile 스크립트로 이미지 만들기

주요 Dockerfile 인스트럭션

인스트럭션	내용
FROM	토대가 되는 이미지를 지정
ADD	이미지에 파일이나 폴더를 추가
COPY	이미지에 파일이나 폴더를 추가
RUN	이미지를 빌드할 때 실행할 명령어를 지정
CMD	컨테이너를 실행할 때 실행할 명령어를 지정
ENTRYPOINT	컨테이너를 실행할 때 실행할 명령어를 강제 지정
ONBUILD	이 이미지를 기반으로 다른 이미지를 빌드할 때 실행할 명령어를 지정
EXPOSE	이미지가 통신에 사용할 포트를 명시적으로 지정
VOLUME	퍼시스턴시 데이터를 저장할 경로를 명시적으로 지정
ENV	환경변수를 정의
WORKDIR	RUN, CMD, ENTRYPOINT, ADD, COPY에 정의된 명령어를 실행하는 작업 디렉터리를 지정
SHELL	빌드 시 사용할 셸을 변경
LABEL	이름이나 버전, 저작자 정보를 설정
USER	RUN, CMD, ENTRYPOINT에 정의된 명령어를 실행하는 사용자 또는 그룹을 지정
ARG	docker build 커맨드를 사용할 때 입력받을 수 있는 인자를 선언
STOPSIGNAL	docker stop 커맨드를 사용할 때 컨테이너 안에서 실행 중인 프로그램에 전달되는 시그널을 변경
HEALTHCHECK	컨테이너 헬스체크 방법을 커스터마이징



도커파일 작성법

항목	값
컨테이너 이름	apa000ex22
새로 만들 이미지 이름	ex22_original1 (끝부분은 소문자 L과 숫자 1이다)

항목	값
기본 이미지 이름	httpd
새로 만들 이미지 이름	ex22_original2
재료 폴더의 경로(윈도우)	C:\Users\사용자명\Documents\apa_folder
재료 폴더의 경로(macOS)	/Users/사용자명/apa_folder
재료 폴더의 경로(리눅스)	/home/사용자명/apa_folder

자신이 사용하는 운영체제에 맞춰 명령어를 작성할 것

Dockerfile 스크립트의 내용

인스트럭션	값
FROM	httpd
COPY	index.html /usr/local/apache2/htdocs

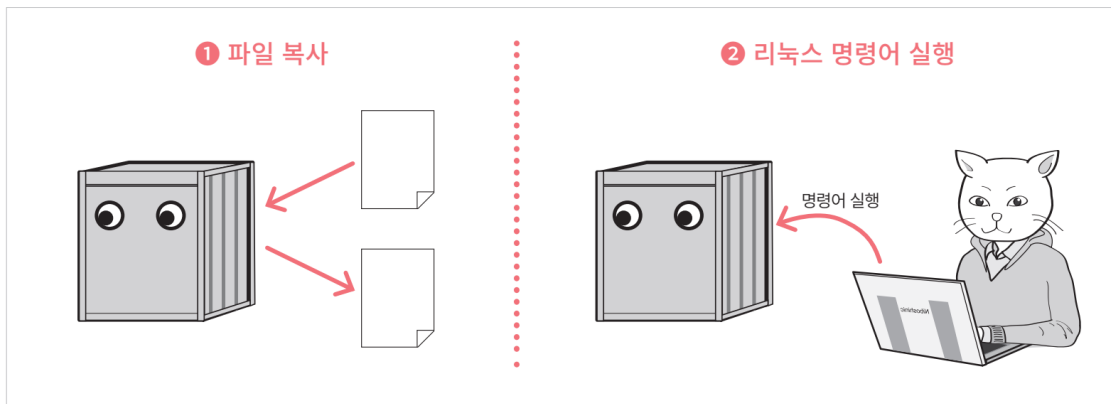


그림 6-5-1 컨테이너를 개조하는 두 가지 방법

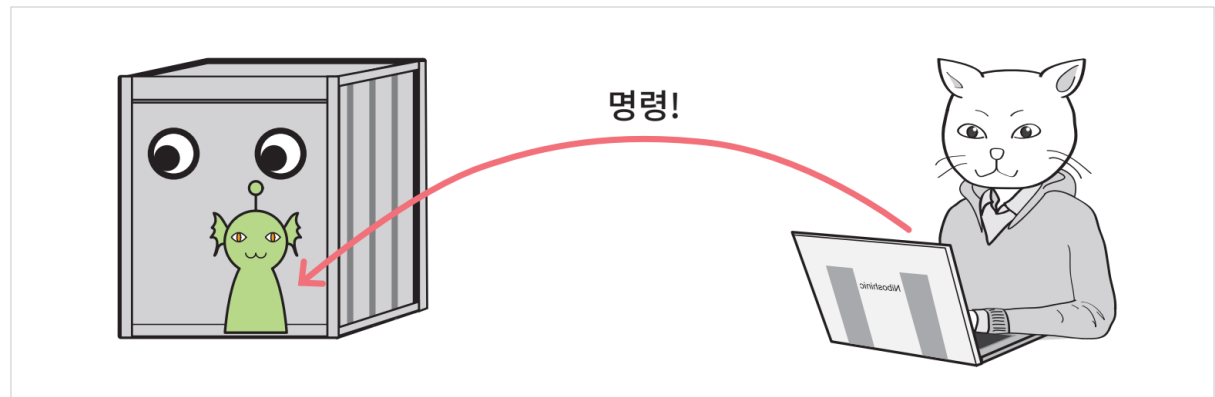


그림 6-5-2 컨테이너에 명령을 전달하는 셀

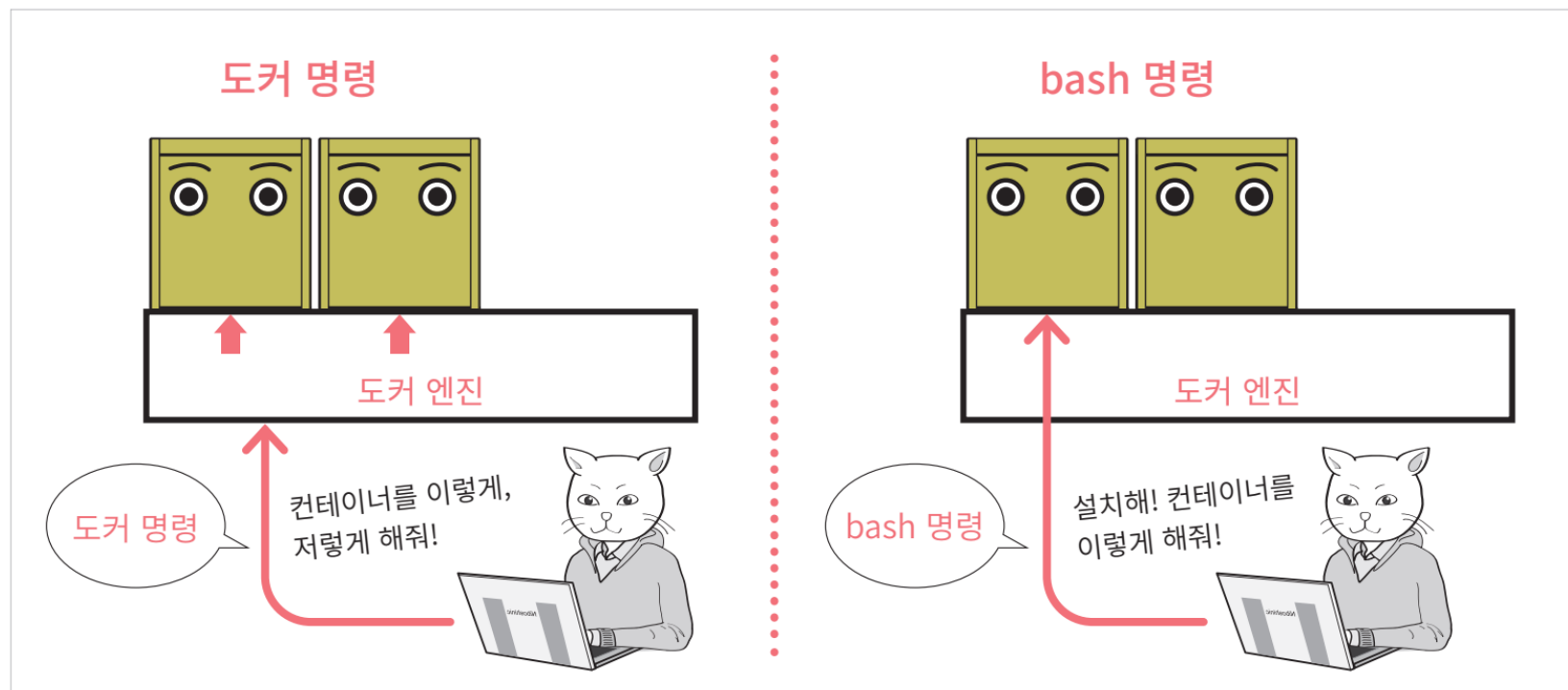
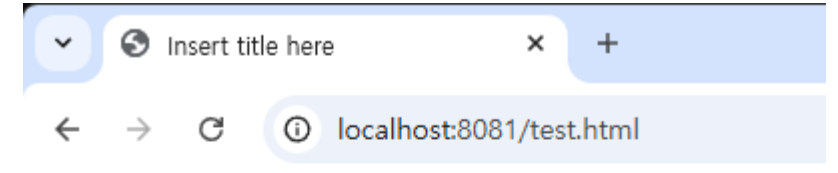


그림 6-5-3 도커 명령과 bash 명령의 차이

```
C:\Users\mit-305\docker>docker exec -it myhttpd1 /bin/bash
root@508b807cc7a6:/usr/local/apache2# cd htdocs
root@508b807cc7a6:/usr/local/apache2/htdocs# cat index.html
root@508b807cc7a6:/usr/local/apache2/htdocs# cp index.html test.html
root@508b807cc7a6:/usr/local/apache2/htdocs# sed -i "11,12d" test.html
root@508b807cc7a6:/usr/local/apache2/htdocs# cat test.html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
  </head>
  <body>
    <h1>XXX의 도커 테스트</h1>
    <hr>
    <h3>호스트 파일시스템에서 도커 컨테이너로 복사된 파일</h3>
root@508b807cc7a6:/usr/local/apache2/htdocs# cat >> test.html
  <h4> test.html </h4>
</body>
</html>
root@508b807cc7a6:/usr/local/apache2/htdocs# cat test.html
```

마지막 행을 입력한 후에 엔터키를 누르고 ctrl+d 를 입력



XXX의 도커 테스트

호스트 파일시스템에서 도커 컨테이너로 복사된 파일

test.html

```
<h4> test.html </h4>
</body>
</html>
```

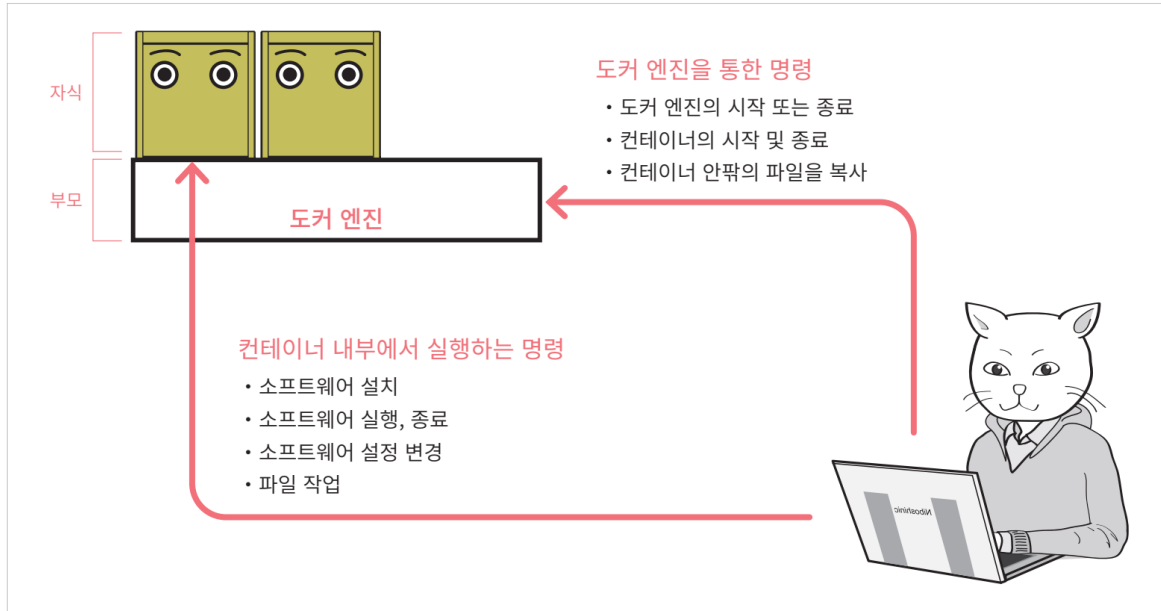


그림 6-5-4 도커 엔진을 통한 명령과 컨테이너 내부에서 실행하는 명령의 차이

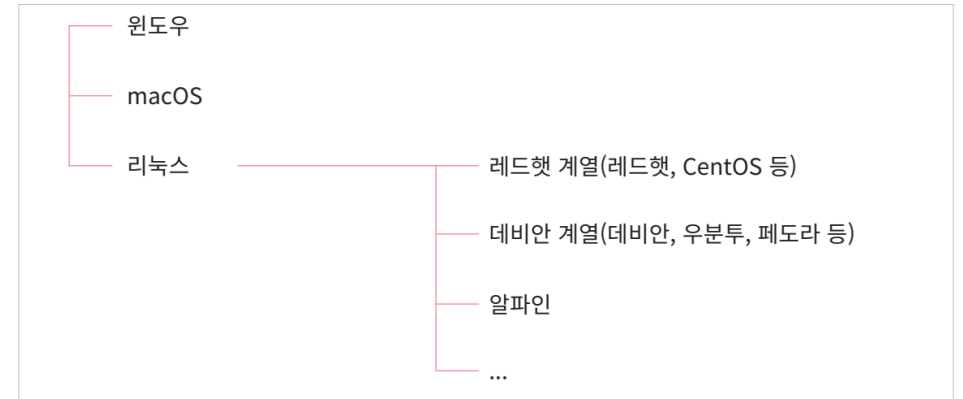


그림 6-5-6 언어(스타일)의 차이

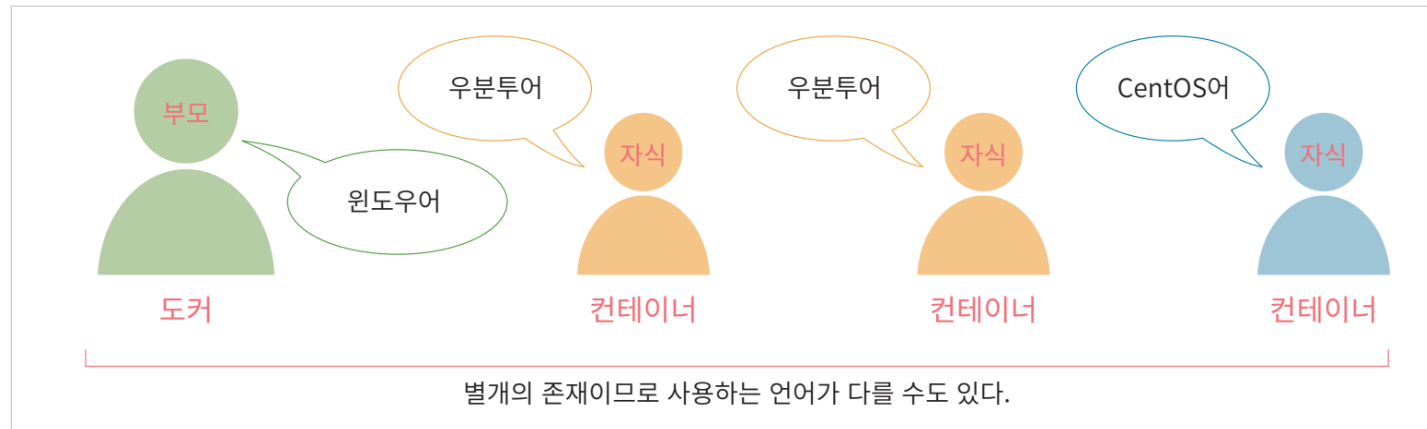


그림 6-5-5 도커와 컨테이너는 별개의 언어를 사용한다.



그림 6-5-7 운영체제에 따라 컨테이너 내부의 명령 스타일이 달라진다.

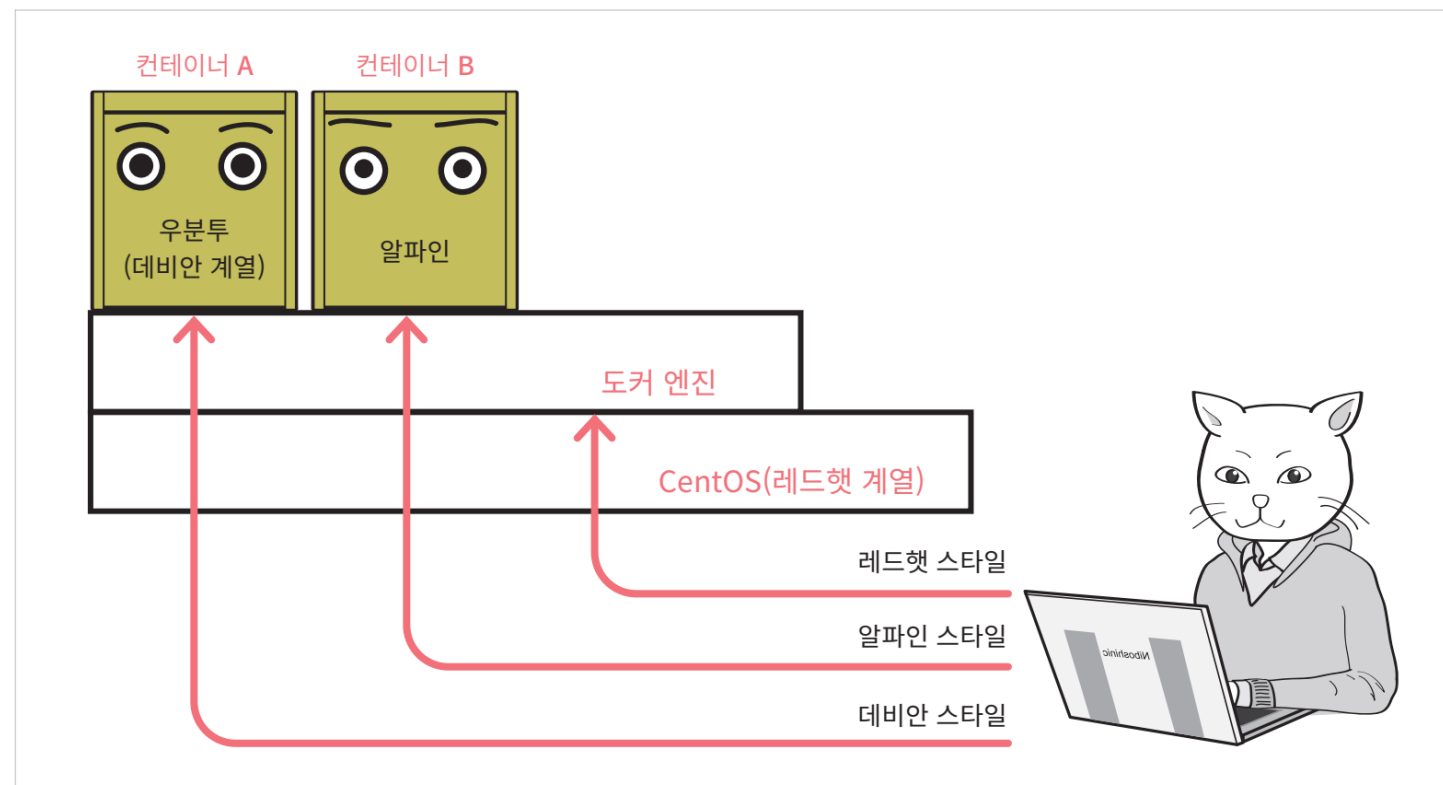


그림 6-5-8 호스트 컴퓨터의 운영체제마저 다른 스타일인 경우

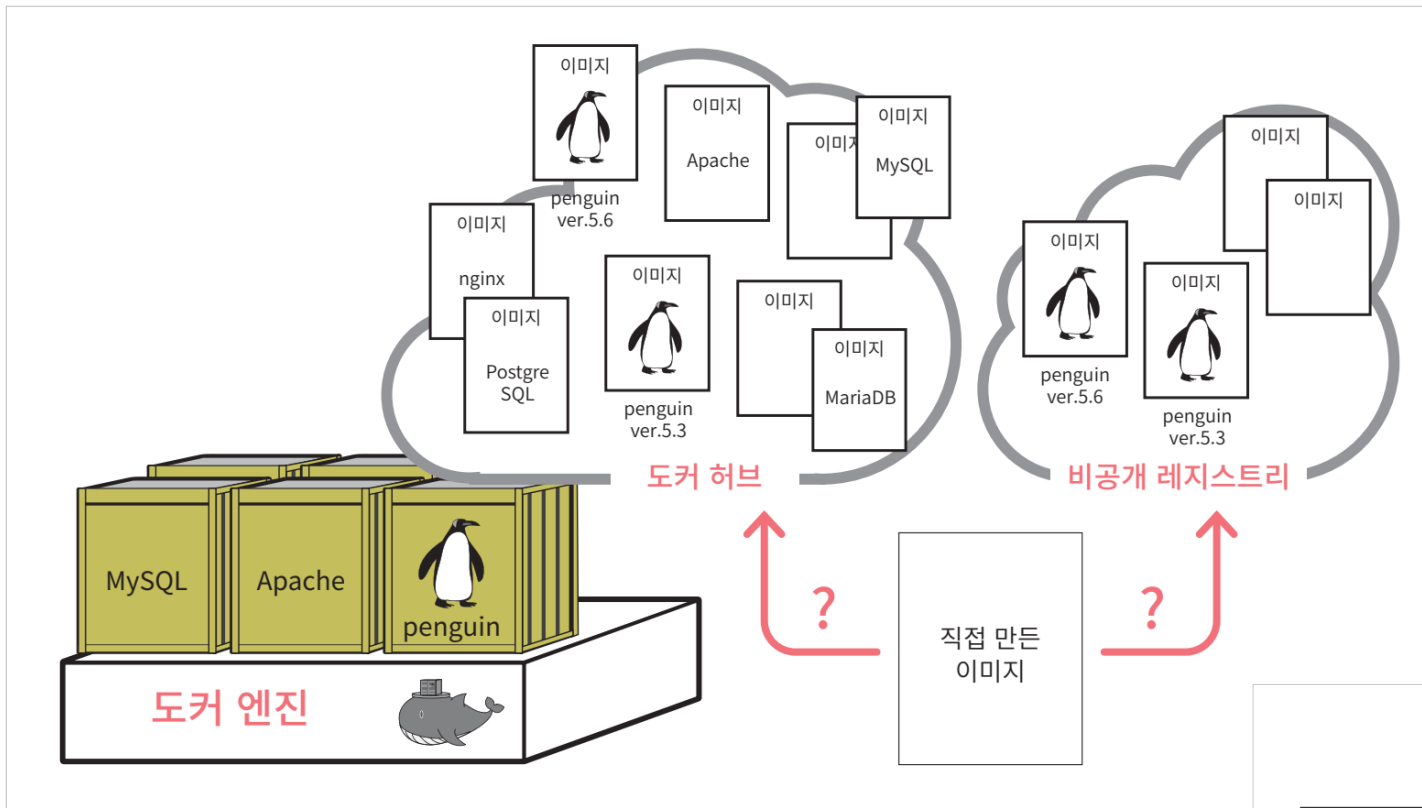


그림 6-6-1 직접 만든 이미지는 어디에 올려야 할까?

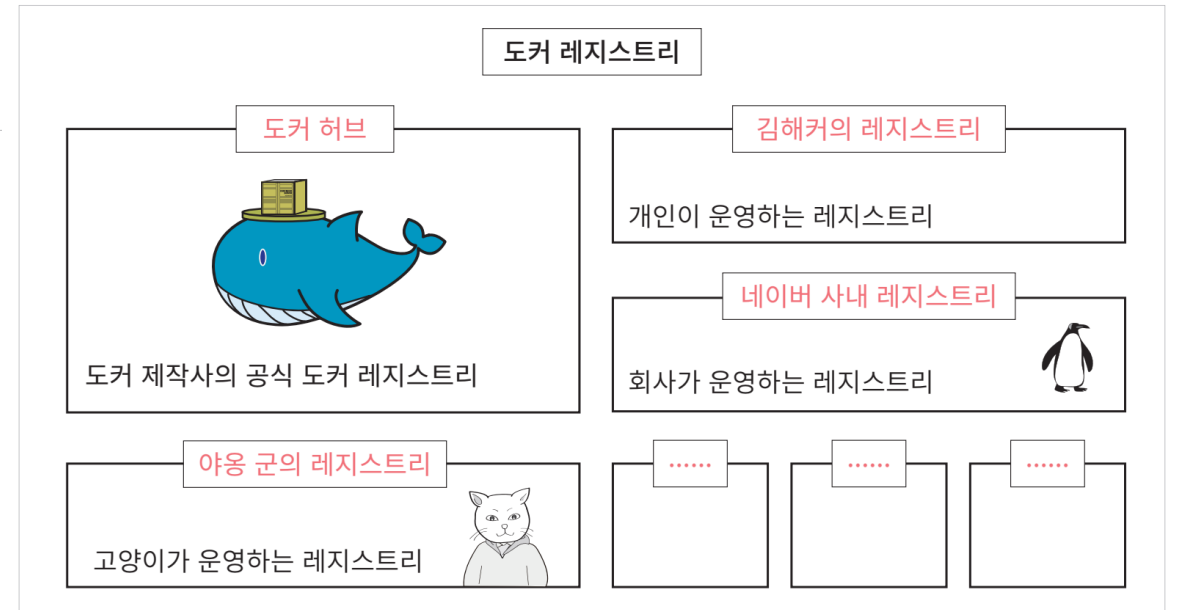


그림 6-6-2 도커 레지스트리

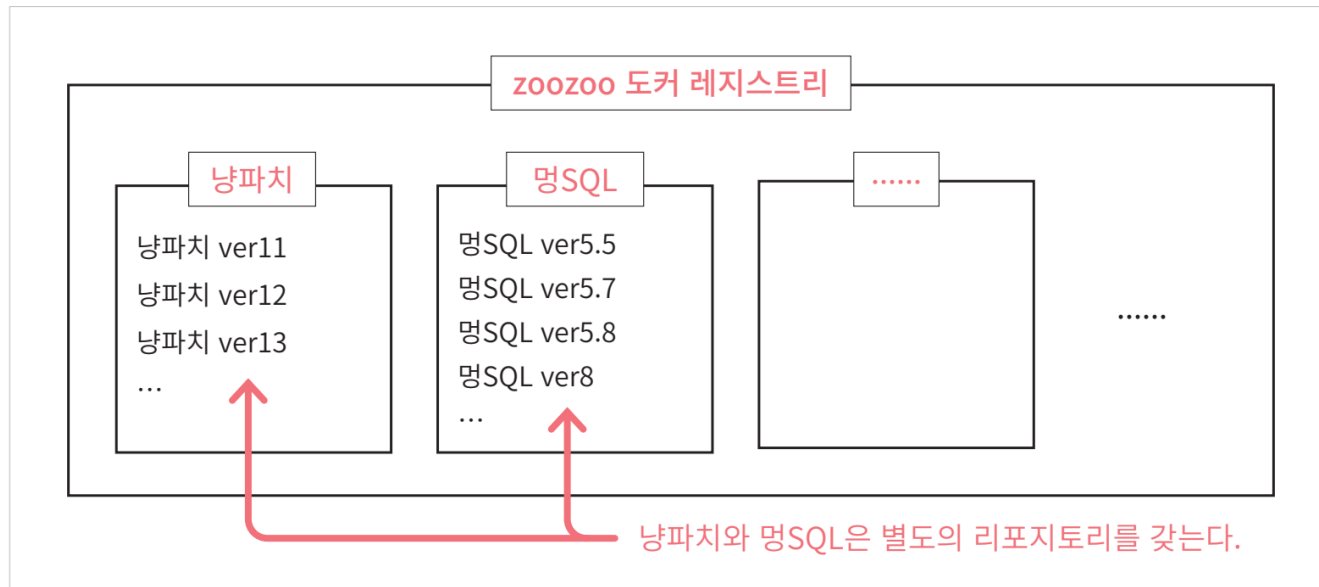


그림 6-6-3 각각의 리포지토리

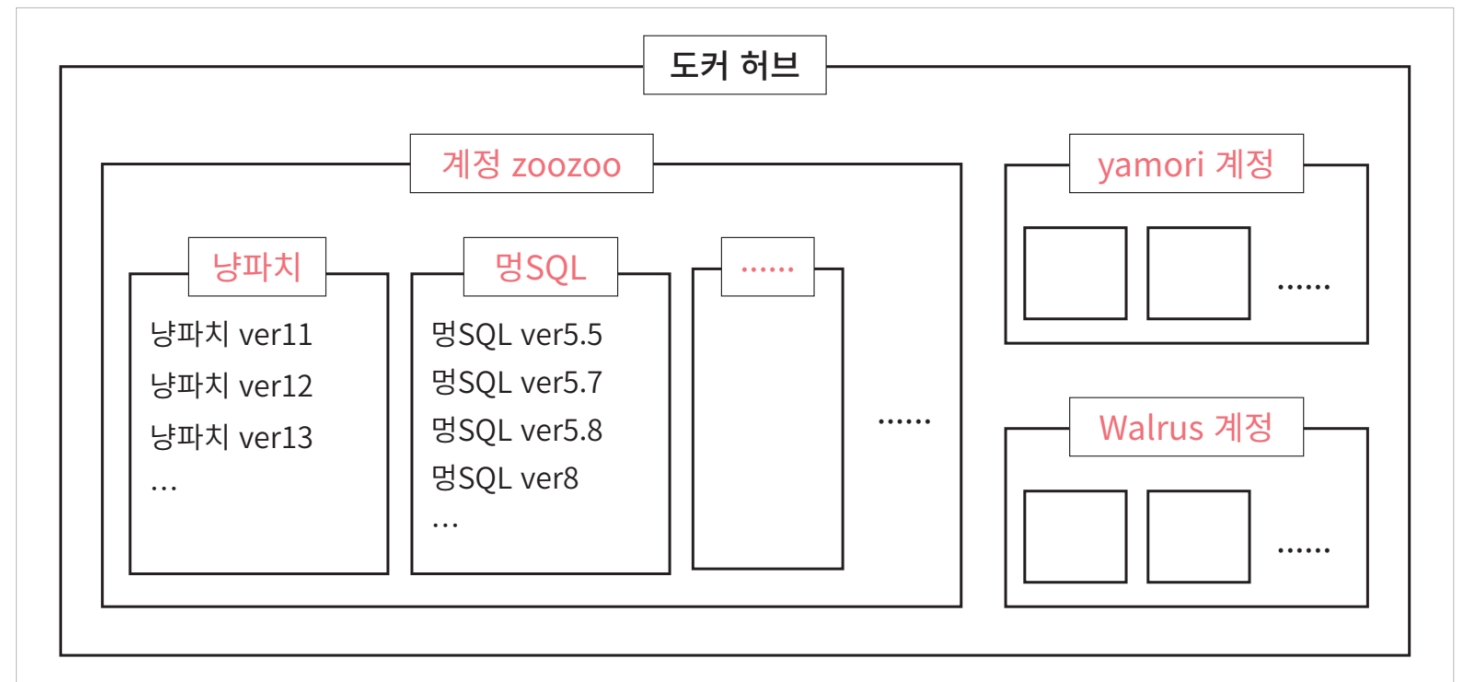


그림 6-6-4 도커 허브는 레지스트리가 여럿 모인 형태다.



그림 6-6-5 태그를 붙이는 방법

비공개 레지스트리의 태그 명명 예

비공개 레지스트리의 태그	레지스트리_주소/리포지토리_이름:버전
자신의 PC에 만든 레지스트리, 리포지토리 이름은 nyapacchi, ver13	localhost:5000/nyapacchi:13
레지스트리 도메인은 zoozoo.coomm, 리포지토리 이름은 nyapacchi, ver13	zoozoo.coomm/nyapacchi:13

도커 허브의 태그 명명 예

도커 허브의 태그	도커_허브_ID/리포지토리_이름:버전
도커 허브 ID는 zoozoousagi, 리포지토리 이름은 nyapacchi, ver13	zoozoousagi/nyapacchi:13

```
C:\Users\mit-305\docker>docker commit myhttpd1 teacherlee/myhttpd1:1.0
```

```
C:\Users\mit-305\docker>docker image ls
```

```
C:\Users\mit-305\docker>docker push teacherlee/myhttpd1:1.0
```

teacherlee
Docker Personal

Repositories

Settings

Default privacy

Notifications

Billing

Usage

Pulls

Repositories

All repositories within the teacherlee namespace.

Search by repository name

All content

Name	Last Pushed	Contains	Visibility
teacherlee/myhttpd1	6 minutes ago	IMAGE	Public

1-1 of 1

teacherlee
Docker Personal

Repositories

Settings

Default privacy

Notifications

Billing

Usage

Pulls

Storage

[Repositories](#) / [myhttpd1](#) / [Settings](#)

Using 0 of 1 private rep

teacherlee/myhttpd1

Last pushed 6 minutes ago

[Add a description](#)

[Add a category](#)

Docker commands

To push a new tag to this repository:

```
docker push teacherlee/myhttpd1:tagname
```

General Tags Image Management **BETA** Collaborators Webhooks Settings

Image security insight settings

Features and controls that help you uncover, understand, and fix issues with your container images in Docker Scout

Docker Scout image analysis **RECOMMENDED**

Delete repository

Deleting a repository will **destroy** all images stored in it.

Delete repository

Delete repository

Warning This deletes the repository, all the images it contains, and its build settings. This cannot be undone.

To confirm deletion, type the name of your repository **myhttpd1**

myhttpd1

Cancel Delete repository forever