

Docker Almacenamiento

Docker Almacenamiento

Volúmenes Docker

- 1.Crea un volumen docker que se llame miweb .
- 2.Crea un contenedor desde la imagen php:7.4-apache donde montes en el directorio /var/www/html (que sabemos que es el DocumentRoot del servidor que nos ofrece esa imagen) el volumen docker que has creado.
- 3.Utiliza el comando docker cp para copiar un fichero index.html en el directorio /var/www/html .
- 4.Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html .
- 5.Borra el contenedor.
- 6.Crea un nuevo contenedor y monta el mismo volumen como en el ejercicio anterior.
- 7.Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html . ¿Seguía existiendo ese fichero?

Bind Mount

- 1.Crea un directorio en tu host y dentro crea un fichero index.html .
- 2.Crea un contenedor desde la imagen php:7.4-apache donde montes en el directorio /var/www/html el directorio que has creado por medio de bind mount .
- 3.Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html .
- 4.Modifica el contenido del fichero index.html en tu host y comprueba que al refrescar la página ofrecida por el contenedor, el contenido ha cambiado.
- 5.Borra el contenedor
- 6.Crea un nuevo contenedor y monta el mismo directorio como en el ejercicio anterior.
- 7.Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html . ¿Se sigue viendo el mismo contenido?

Volúmenes Docker

1.Crea un volumen docker que se llame miweb .

```
docker volume create miweb
```

```
daw@daw-docker:~$ docker volume ls
DRIVER      VOLUME NAME
local       061c985b9317685288f3c75e4c6dd38fbd900e27d0e692c8bd88b0878a420927
local       miweb
daw@daw-docker:~$
```

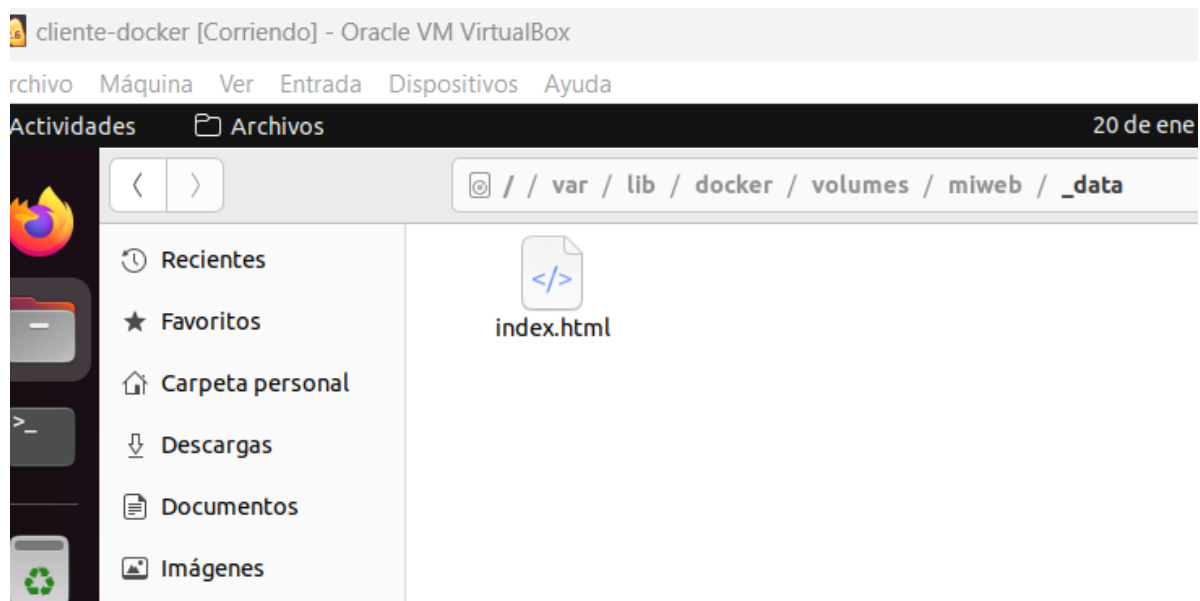
2.Crea un contenedor desde la imagen php:7.4-apache donde montes en el directorio /var/www/html (que sabemos que es el DocumentRoot del servidor que nos ofrece esa imagen) el volumen docker que has creado.

```
docker run -d --name myApache --mount type=volume,src=miweb,dst=/var/www/html -p 8080:80 php:7.4-apache
```

```
daw@daw-docker:~$ docker run -d --name myApache --mount type=volume,src=miweb,dst=/var/www/html -p 8080:80 php:7.4-apache
f00356e7f890f6f9fc1813e4563e176d08d00fdf2fa493467066883fb03ad3a9
daw@daw-docker:~$
```

3.Utiliza el comando docker cp para copiar un fichero index.html en el directorio /var/www/html .

```
docker cp index.html myApache:/var/www/html
```



4.Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html .



5. Borra el contenedor.

Borramos el contenedor:

```
docker rm -f myApache
```

```
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
f00356e7f890   php:7.4-apache "docker-php-entrypoi..." 10 minutes ago Exited
(0) About a minute ago
myApache
daw@daw-docker:~$ docker rm -f myApache
myApache
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
daw@daw-docker:~$
```

6. Crea un nuevo contenedor y monta el mismo volumen como en el ejercicio anterior.

Volvemos a crear el mismo contenedor que en el ejercicio 2. Por lo que usamos el mismo comando:

```
docker run -d --name myApache --mount type=volume,src=miweb,dst=/var/www/html -p 8080:80 php:7.4-apache
```

```
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
3c0ede1dc114   php:7.4-apache "docker-php-entrypoi..." About a minute ago Up About
a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp myApache
daw@daw-docker:~$
```

7. Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html . ¿Seguía existiendo ese fichero?

Al acceder al servidor observamos que el fichero index.html sigue activo, ya que el archivo está alojado en nuestro volumen local `miweb`.



Bind Mount

1. Crea un directorio en tu host y dentro crea un fichero index.html .

Creemos la carpeta myApp y dentro un archivo index.html.

```
daw@daw-docker:~$ mkdir myApp
daw@daw-docker:~$ cd myApp/
daw@daw-docker:~/myApp$ sudo nano index.html
daw@daw-docker:~/myApp$ ls
index.html
daw@daw-docker:~/myApp$
```

2. Crea un contenedor desde la imagen php:7.4-apache donde montes en el directorio /var/www/html el directorio que has creado por medio de bind mount .

```
docker run -d --name myAppBind -v /home/daw/myApp:/var/www/html -p 8080:80 php:7.4-apache
```

```
daw@daw-docker:~/myApp$ docker run -d --name myAppBind -v /home/daw/myApp:/var/www/html -p 8080:80 php:7.4-apache
970b82823daeba8693c63a0e82e284f4869f22a1a70b6b8ba68ae1f8ca7bae96
daw@daw-docker:~/myApp$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
970b82823dae	php:7.4-apache	"docker-php-entrypoi..."	48 seconds ago	Up 46 seconds	0.0.0.0:8080->80/tcp, :::8080->80/tcp

```
myAppBind
daw@daw-docker:~/myApp$
```

3. Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html .

Accedemos a la dirección ip del servidor y visualizamos el archivo.



4. Modifica el contenido del fichero index.html en tu host y comprueba que al refrescar la página ofrecida por el contenedor, el contenido ha cambiado.

Modificamos el index.html desde la carpeta local:



Al acceder al servidor, vemos que el contenido ha cambiado:



5. Borra el contenedor

Eliminamos el contenedor:

```
docker rm -f myAppBind
```

```
myAppBind
daw@daw-docker:~/myApp$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
daw@daw-docker:~/myApp$
```

6. Crea un nuevo contenedor y monta el mismo directorio como en el ejercicio anterior.

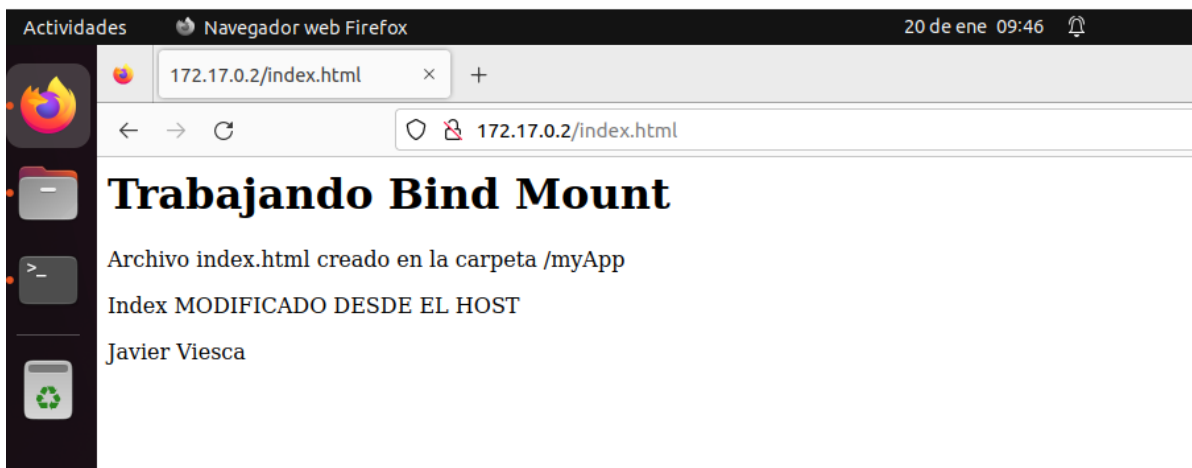
Volvemos a crear el mismo contenedor que en el ejercicio anterior:

```
docker run -d --name myAppBind -v /home/daw/myApp:/var/www/html -p 8080:80 php:7.4-apache
```

```
daw@daw-docker:~/myApp$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS
7da5b62bf3ca  php:7.4-apache  "docker-php-entrypoi..."  40 seconds ago  Up 39 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp
myAppBind
daw@daw-docker:~/myApp$
```

7. Accede al contenedor desde el navegador para ver la información ofrecida por el fichero index.html. ¿Se sigue viendo el mismo contenido?

Al acceder al servidor vemos que se sigue viendo el mismo contenido, el servidor sigue leyendo los datos de nuestro host:



Trabajando Bind Mount

Archivo index.html creado en la carpeta /myApp

Index MODIFICADO DESDE EL HOST

Javier Viesca