

## Tarea 3 - Almacenamiento en Docker - IESGN

Realizado por Pablo R.

Vamos a trabajar con volúmenes docker:

- 1. Crear un volumen docker que se llame miweb.

```
docker volume create miweb
```

```
root@daw-docker:/home/daw# docker volume create miweb
miweb
root@daw-docker:/home/daw#
```

Creamos el volumen mi web.

- 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 -p 80:80 --name apache-php2 -v miweb:/var/www/html php:7.4-apache
```

```
root@daw-docker:/home/daw# docker run -d -p 80:80 --name apache-php2 -v miweb:/var/www/html php:7.4-apache
980643bbe9a994269878f29d5e8dce1ccf55e07c5de47a3a14990451537fd556
root@daw-docker:/home/daw#
```

Creamos el php asociado al volumen miweb.

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

```
nano /home/daw/index.html
docker cp /home/daw/index.html apache-php2:/var/www/html
```

```
root@daw-docker: /home/daw
GNU nano 6.2 index.html
<!DOCTYPE html>
<html>
<body>
<h1>Mi web Pablo</h1>
</body>
</html>
```

```
root@daw-docker: /home/daw
root@daw-docker:/home/daw# docker cp /home/daw/index.html apache-php2:/var/www/html
root@daw-docker:/home/daw#
```

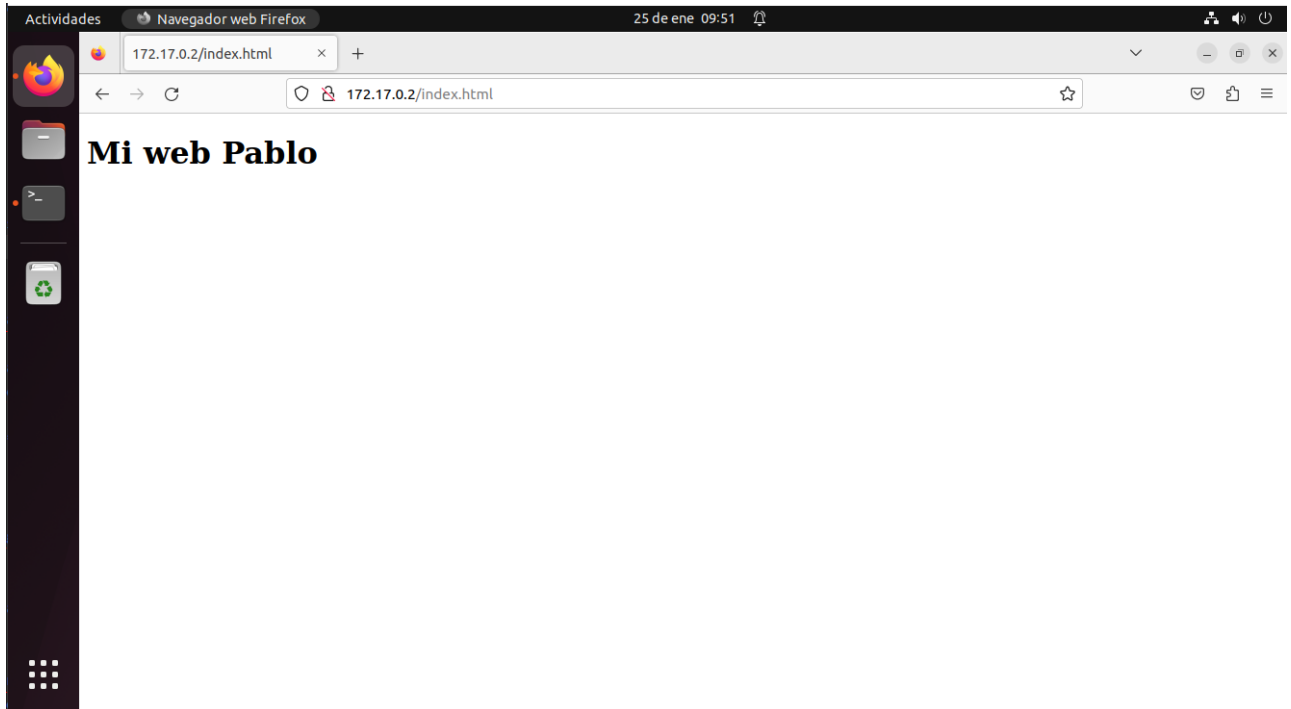
Creamos el archivo index.html en nuestro equipos con nano y copiamos dicho fichero a la máquina docker.

Copiamos el archivo index.html al contenedor.

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

```
docker start apache-php2
```

```
root@daw-docker:/home/daw# docker start apache-php2
apache-php2
root@daw-docker:/home/daw#
```



Iniciamos el contenedor y accedemos por el navegador.

- 5. Borra el contenedor

```
docker stop <id_contenedor>
docker rm <id_contenedor>
```

```
root@daw-docker:/home/daw# docker stop 980643bbe9a9
980643bbe9a9
root@daw-docker:/home/daw# docker rm 980643bbe9a9
980643bbe9a9
root@daw-docker:/home/daw#
```

Paramos el contenedor y eliminamos el contenedor.

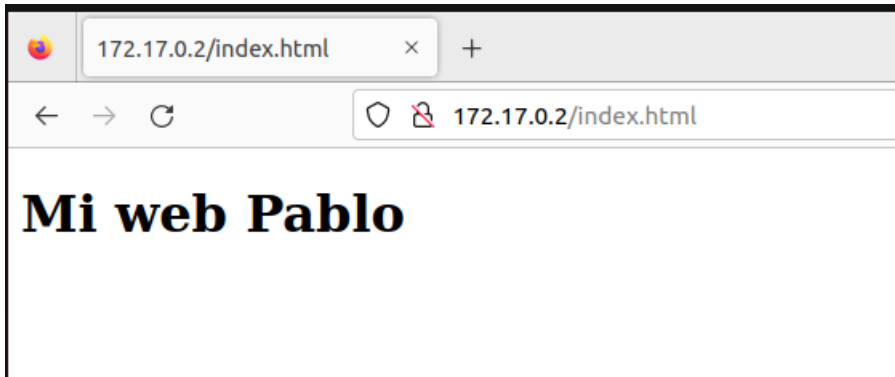
- 6. Crea un nuevo contenedor y monta el mismo volumen como en el ejercicio anterior.

```
docker run -d -p 80:80 --name apache-php3 -v miweb:/var/www/html php:7.4-apache
```

```
root@daw-docker:/home/daw# docker run -d -p 80:80 --name apache-php3 -v miweb:/var/www/html php:7.4-apache
28feb6e3e09ed57fa06d71882803d621ba27e988c488a768db29685067acd78
root@daw-docker:/home/daw#
```

Creamos un nuevo contenedor y montamos el mismo volumen del ejercicio anterior.

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



Sigue existiendo pues reutilizamos el volumen anterior.

Vamos a trabajar con bind mount:

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

```
mkdir /home/daw/miweb2
nano /home/daw/miweb2/index.html
```

```
root@daw-docker: /home/daw/miweb2

root@daw-docker:/home/daw# mkdir miweb2
root@daw-docker:/home/daw# cd miweb2/
root@daw-docker:/home/daw/miweb2# nano index.html
root@daw-docker:/home/daw/miweb2#
```

Creamos el directorio y el fichero index.

- 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 -p 80:80 --name apache-php4 -v /home/daw/miweb2:/var/www/html
php:7.4-apache
```

Creamos el contenedor con bind mount.

```
root@daw-docker:/home/daw/miweb2# docker run -d -p 80:80 --name apache-php4 -v /home/daw/miweb2:/var/www/html
php:7.4-apache
fcba98bdb9377a6cae897a3107236f7e79e19ab67585e0577574b2e76ac1a5bc
root@daw-docker:/home/daw/miweb2#
```

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



Accedemos a la web y observamos que tenemos la web funcionando.

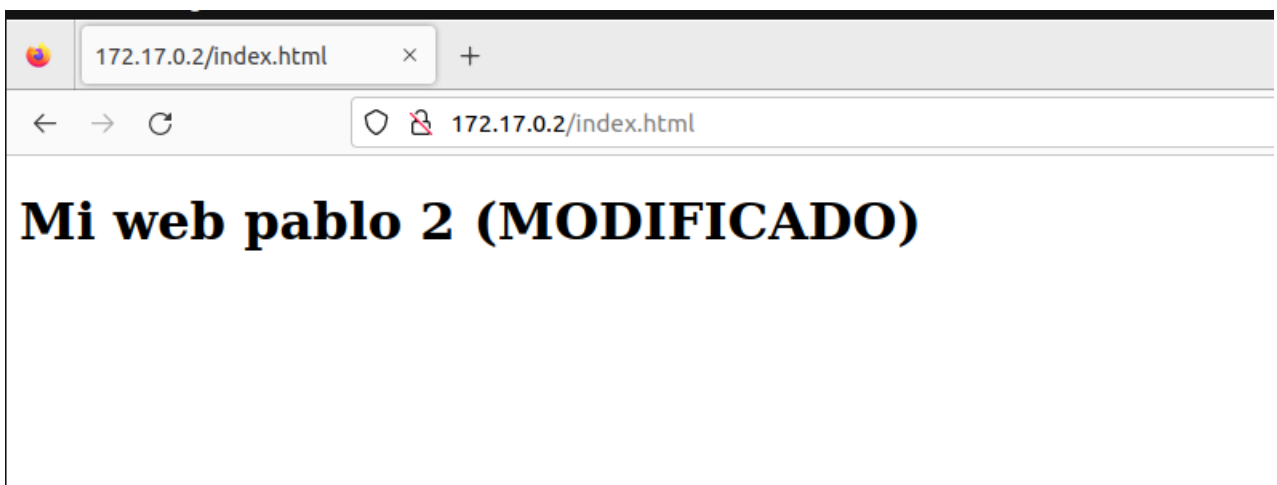
- 4. Modifica el contenido del fichero index.html en tu host y comprueba que al refrescar la página ofrecida por el contenedor, el contenido ha cambiado.

```
nano /home/daw/miweb2/index.html
```

```
root@daw-docker: /home/daw/miweb2
GNU nano 6.2 /home/daw/miweb2/index.html
<!DOCTYPE html>
<html>
<body>
<h1>Mi web pablo 2 (MODIFICADO)</h1>
</body>
</html>

root@daw-docker:/home/daw/miweb2# nano /home/daw/miweb2/index.html
root@daw-docker:/home/daw/miweb2#
```

Modificamos el archivo en el host.



Observamos el cambio.

- 5. Borra el contenedor

```
docker stop <id_contenedor>
docker rm <id_contenedor>
```

```
root@daw-docker:/home/daw/miweb2# docker stop fcba98bdb937
fcba98bdb937
root@daw-docker:/home/daw/miweb2# docker rm fcba98bdb937
fcba98bdb937
root@daw-docker:/home/daw/miweb2#
```

Borramos el contenedor.

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

```
docker run -d -p 80:80 --name apache-php5 -v /home/daw/miweb2:/var/www/html
php:7.4-apache
```

```
root@daw-docker:/home/daw/miweb2# docker run -d -p 80:80 --name apache-php5 -v /home/daw/miweb2:/var/www/html php:7.4-apache
296d8581bd04252c24c7b8dcd71000fad6f68210868fd70c7a0e71b3837e0d4
root@daw-docker:/home/daw/miweb2#
```

Creamos el contenedor de nuevo.

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



Observamos que el contenido se sigue visualizando.