

# Docker

1. Crea un volumen docker que se llame `miweb`.

```
docker volume create miweb
```

```
root@docker:/home/docker# docker volume create miweb
miweb
root@docker:/home/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 apache1 -v miweb:/var/www/html php:7.4-apache
```

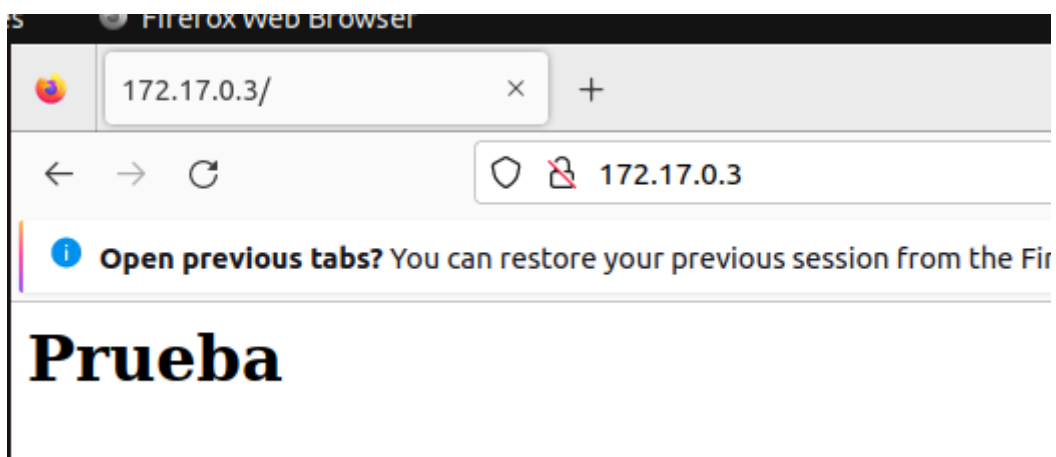
```
root@docker:/home/docker# docker run -d --name apache1 -v miweb:/var/www/html php:7.4-apache
e2ce9629f3fff93339e3f03d60a44c49f04cf4e41cafa3ec740ab9fab524b4ff
root@docker:/home/docker#
```

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

```
docker cp index.html apache1:/var/www/html
```

```
root@docker:/home/docker# docker cp index.html apache1:/var/www/html
Successfully copied 2.048kB to apache1:/var/www/html
root@docker:/home/docker#
```

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



5. Borra el contenedor

```
docker stop idcontenedor && docker rm idcontenedor
```

```
76643ba91d4b php:7.4-apache docker-php-entrypoint... 5 hours ago Up 5 m
root@docker:/home/docker# docker stop e2ce9629f3ff && docker rm e2ce9629f3ff
e2ce9629f3ff
e2ce9629f3ff
root@docker:/home/docker#
```

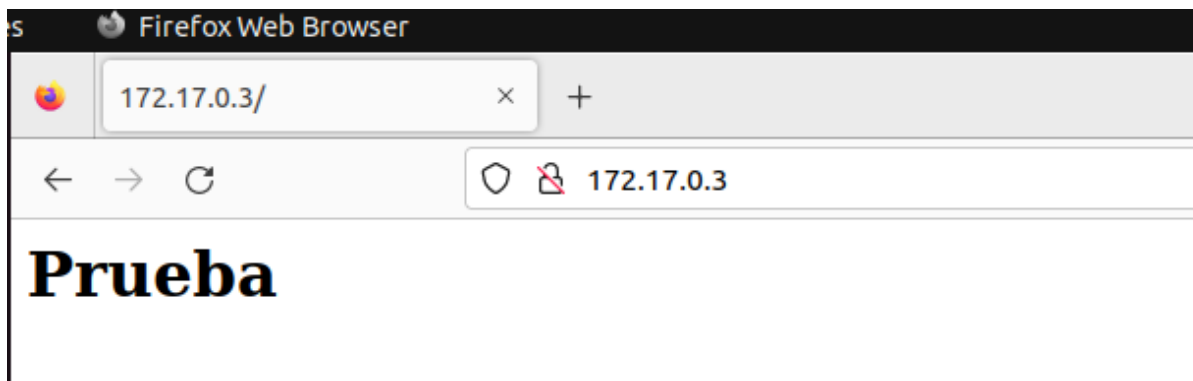
6. Crea un nuevo contenedor y monta el mismo volumen como en el ejercicio anterior.

```
docker run -d --name apache1 -v miweb:/var/www/html php:7.4-apache
```

```
root@docker:/home/docker# docker run -d --name apache2 -v miweb:/var/www/html php:7.4-apache
021ff25b3ab81dd4c804fed726a51774392ff8a4f19a510eb68c077eed33f8d7
root@docker:/home/docker#
```

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

Si, al usar el mismo volumen.



## Ejercicios de Bind Mount

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

```
mkdir contenedor
cd contenedor/
nano index.html
cat index.html
```

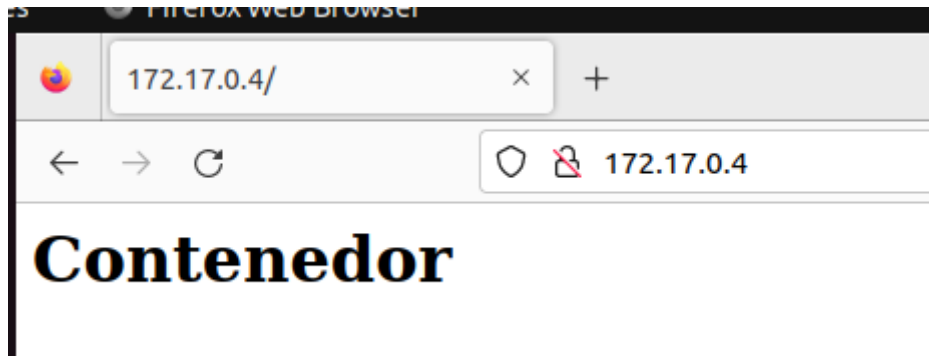
```
root@docker:/home/docker# mkdir contenedor
root@docker:/home/docker# cd contenedor/
root@docker:/home/docker/contenedor# nano index.html
root@docker:/home/docker/contenedor# cat index.html
<h1>Contenedor</h1>
root@docker:/home/docker/contenedor#
```

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 apache3 -v /home/docker/contenedor:/var/www/html php:7.4-apache
```

```
see 'docker run --help'.
root@docker:/home/docker/contenedor# docker run -d --name apache3 -v /home/docker/contenedor:/var/www/html php:7.4-apache
e76122b55423ecb1ffc844050fa890edea67d45eacdd22094daff43fdd9b4cbe
root@docker:/home/docker/contenedor#
```

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.

```
root@docker:/home/docker/contenedor# nano index.html
root@docker:/home/docker/contenedor# cat index.html
<h1>Contenedor 1</h1>
root@docker:/home/docker/contenedor#
```



5. Borra el contenedor

```
docker stop idcontenedor && docker rm idcontenedor
```

```
root@docker:/home/docker/contenedor# docker stop e76122b55423 && docker rm e76122b55423
e76122b55423
e76122b55423
root@docker:/home/docker/contenedor#
```

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

```
docker run -d --name apache4 -v /home/docker/contenedor:/var/www/html php:7.4-apache
```

```
e76122b55423
root@docker:/home/docker/contenedor# docker run -d --name apache4 -v /home/docker/contenedor:/var/www/html php:7.4-apache
1c769790f2813ffc6c4f7f0d905e09917910c36f8e5e6843599ac43a9915e5d7
root@docker:/home/docker/contenedor#
```

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

Si, porque es el mismo volumen.

