

Resolución del ejercicio 1 - trabajo con imágenes

Realizado por Neri Sánchez

Servidor web

En primer lugar conseguimos la imagen requerida: `php:7.4-apache` :

```
docker pull php:7.4-apache
```

```
nerisg@cliente:~$ sudo docker pull php:7.4-apache
[sudo] contraseña para nerisg:
7.4-apache: Pulling from library/php
ae13dd578326: Pull complete
f15d475049bf: Pull complete
886e5161983f: Pull complete
aa7666573a25: Pull complete
59357a0f9863: Pull complete
dc3ffb8c774e: Pull complete
513e9383f6d4: Pull complete
be02a7ef9c8a: Pull complete
c40cdae8b7d7: Pull complete
966bfd894459: Pull complete
66283f2c0bc8: Pull complete
bacb62113170: Pull complete
f4d522605a58: Pull complete
Digest: sha256:ba2d78d2d1d16e66718d98374556ad5840f0b2e4be3903d77441447b2b482f0b
Status: Downloaded newer image for php:7.4-apache
docker.io/library/php:7.4-apache
```

Una vez lista crearemos el contenedor que correrá esta imagen:

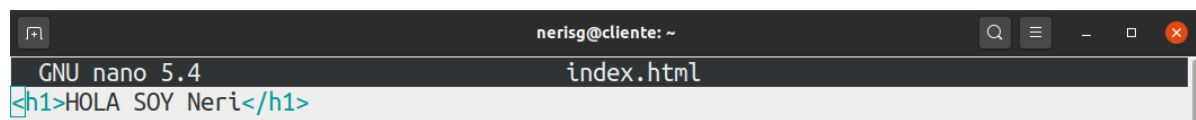
```
sudo docker run -d --name web -p 8000:80 php:7.4-apache
```

```
nerisg@cliente:~$ sudo docker run -d --name web -p 8000:80 php:7.4-apache
7c2157dfa1ca93b9afb604472ed35def8afaea142247a0cb682231e0a30d038a
```

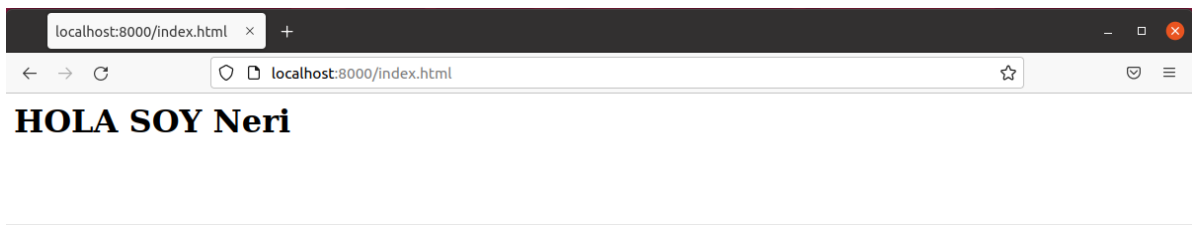
Accedemos dinámicamente a Apache para colocar el archivo `index.html` y editarlo (una vez iniciamos la consola nos coloca en la carp `/var/www/html`)

```
docker exec -it web bash
apt-get update
apt-get install nano
nano index.html
```

Insertamos la línea de código en el `index.html` y guardamos



Pantallazo que desde el navegador muestre el fichero `index.html` .



A continuación creamos el script PHP que mostrará el mes actual.



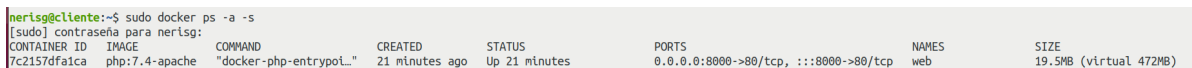
Pantallazo que desde un navegador muestre la salida del script `mes.php`.



Después de añadir los archivos comprobamos el tamaño del contenedor:

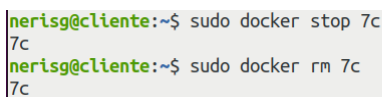
```
sudo docker ps -a -s
```

Pantallazo donde se vea el tamaño del contenedor `web` después de crear los dos ficheros.



Por último borramos el contenedor

```
docker stop 7c
docker rm 7c
```



Servidor de base de datos

Descargamos una imagen de `mariadb`

```
sudo docker pull mariadb
```

```

nerisg@cliente:~$ sudo docker pull mariadb
[sudo] contraseña para nerisg:
Using default tag: latest
latest: Pulling from library/mariadb
4d32b49e2995: Pull complete
03231033a668: Pull complete
5bf8bcd9a929: Pull complete
6c260950e90f: Pull complete
35baff4c5363: Pull complete
16cdd7e19187: Pull complete
813bfa8c4c6e: Pull complete
e5fe98b3cc0e: Pull complete
98f9c19436dd: Pull complete
9eac0168aeb9: Pull complete
5ec17c09c8a0: Pull complete
Digest: sha256:f2eddab27776f16767ea3473143abbebcf40e7d122874dc90763439b90b7d7ce
Status: Downloaded newer image for mariadb:latest
docker.io/library/mariadb:latest

```

A continuación creamos la base de datos:

```

docker run --name bbdd -e MARIADB_ROOT_PASSWORD=root -e MARIADB_DATABASE=prueba -e
MARIADB_USER=invitado -e MARIADB_PASSWORD=invitado -p 3336:3306 -d mariadb

```

```

nerisg@cliente:~$ sudo docker run --name bbdd -e MARIADB_ROOT_PASSWORD=root -e MARIADB_DATABASE=prueba -e MARIADB_USER=invitado -e MARIADB_PASSWORD=invitado -p 3336:3306 -d mariadb
638e4ca8be83d21baf57151ab64a9aa738507ac79e14a4a0ca2513fbb3402f

```

Para hacer las pruebas es necesario crear el contenedor de phpMyAdmin añadiendo el 'flag' `--link` :

```

sudo docker pull phpmyadmin
sudo docker run --name myadmin -d -e PMA_ARBITRARY=1 --link bbdd:mariadb -p 8080:80
phpmyadmin

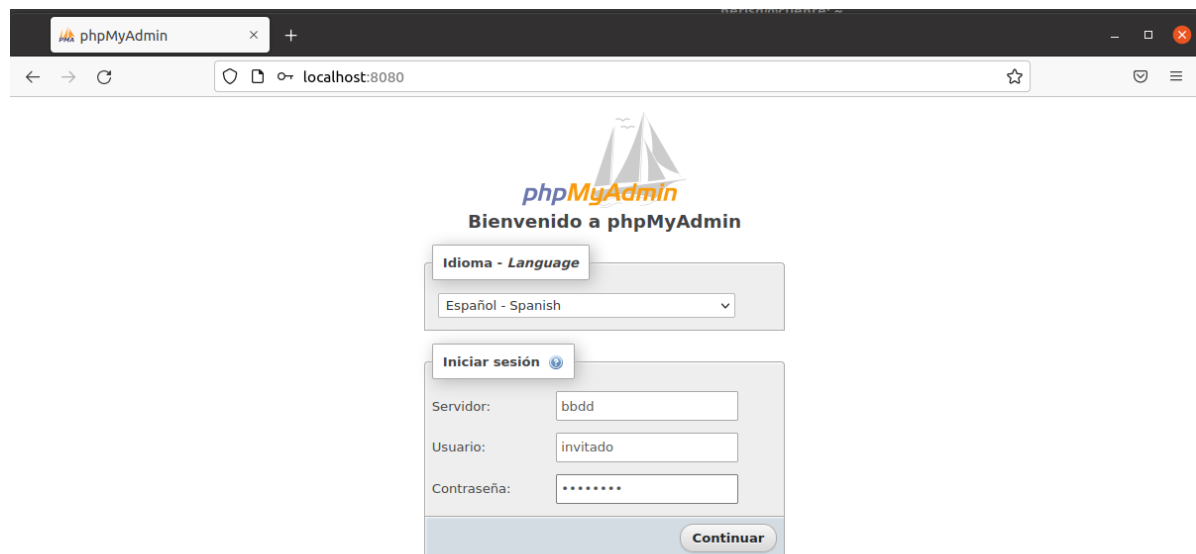
```

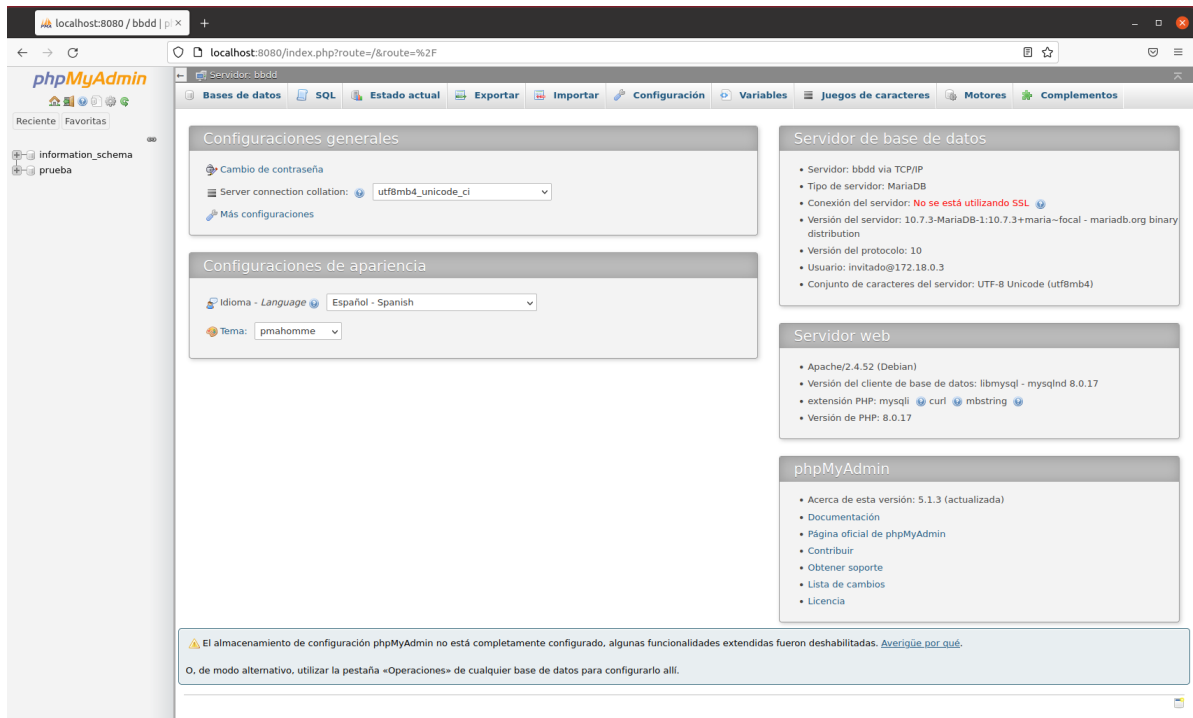
```

nerisg@cliente:~$ sudo docker run --name myadmin -d -e PMA_ARBITRARY=1 --link bbdd:mariadb -p 8080:80 phpmyadmin
0206621ab07615b1f3fc20a9c3ed141342e22789c7a3d12d26cc7fb4f92b8a2f

```

Pantallazo de la consola de la Base de Datos donde se pueda observar que hemos podido conectarnos al servidor de base de datos con el usuario creado y que se ha creado la base de datos prueba (`show databases`).





Intentamos borrar la imagen mariadb:

```
sudo docker rmi mariadb
```

Pantallazo donde se comprueba que no se puede borrar la imagen **mariadb** mientras el contenedor **bbdd** está creado.

```
nerisg@cliente:~$ sudo docker rmi mariadb
Error response from daemon: conflict: unable to remove repository reference "mariadb" (must force) - container e1801ad98e0c is using its referenced image 6e0162b44a5f
```

Pantallazo donde se vean las imágenes que tienes en tu registro local.

```
sudo docker images
```

```
nerisg@cliente:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
phpmyadmin    latest   7b7afabfa66e   4 days ago    508MB
mariadb       latest   6e0162b44a5f   4 days ago    414MB
php           7.4-apache 91d896afaf11   6 days ago    452MB
nginx         latest   f6987c8d6ed5   3 months ago  141MB
httpd         2.4      dabbfbfe0c57b  3 months ago  144MB
ubuntu        latest   ba6acccedd29   5 months ago  72.8MB
hello-world   latest   feb5d9fea6a5   6 months ago  13.3kB
```

Pantallazo donde se vea cómo se eliminan los contenedores utilizados.

Listamos los contenedores

```
sudo docker ps -a -s
```

Borramos los usados durante la tarea

```
sudo docker rm e1
sudo docker rm 02
```

```
nerlsg@cliente:~$ sudo docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES          SIZE
0206621ab076   phpmyadmin    "/docker-entrypoint..." 23 minutes ago Exited (0) 5 minutes ago           myadmin        220kB (virtual 508MB)
e1801ad98e0c   mariadb       "docker-entrypoint.s..." 25 minutes ago Exited (0) 5 minutes ago           bbdd           0B (virtual 414MB)
3e702cc239c3   httpd:2.4     "httpd-foreground"       2 months ago   Exited (0) 2 months ago           my-apache-app  101B (virtual 144MB)
nerlsg@cliente:~$ sudo docker rm 02
02
nerlsg@cliente:~$ sudo docker rm e1
e1
nerlsg@cliente:~$ sudo docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES          SIZE
3e702cc239c3   httpd:2.4     "httpd-foreground"       2 months ago   Exited (0) 2 months ago           my-apache-app  101B (virtual 144MB)
```