

Resolucion Ejercicio 1

Laura A. Álvarez Cubillas

Resolución Ejercicio 1

Servidor web

- Arrancar el contenedor `php:7.4-apache`
 - usamos `-d` para que quede arrancado en background
 - `--name` para dar nombre al contenedor
 - `-p` para establecer la conexión entre puertos

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

```
laura @ Artico ~  
└─ $ docker run -d --name web -p 8000:80 php:7.4-apache  
Unable to find image 'php:7.4-apache' locally  
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
```

Como no tenía la imagen se la ha descargado a la vez que crea el contenedor.

Ahora ya se puede ver la imagen descargada en el equipo con el comando:

```
docker images
```

```
laura @ Artico ~  
└─ $ docker images  
REPOSITORY          TAG             IMAGE ID        CREATED        SIZE  
php                  7.4-apache     91d896afaf11   8 days ago    452MB
```

Comprobamos que el contenedor esta arrancado:

```
docker ps -a
```

```
laura @ Artico ~
└─ $ ▶ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED
325740a6bff5   php:7.4-apache "docker-php-entrypoi..." 4 minutes ago

STATUS        PORTS
Up 4 minutes  0.0.0.0:8000->80/tcp, :::8000->80/tcp
NAMES
web
```

Para poder trabajar en el contenedor abrimos una terminal en él con el comando `exec` :

```
docker exec -it web bash
```

y ahora estamos dentro del contenedor

```
laura @ Artico ~
└─ $ ▶ docker exec -it web bash
root@325740a6bff5:/var/www/html#
```

- ✓ Creamos el fichero `index.html` , para ello primero actualizo el contenedor e instalo el editor `nano` para editar los archivos que creo con el comando `touch` , la secuencia es:

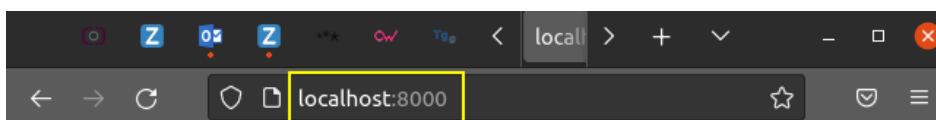
```
apt-get update
apt-get install nano
touch index.html
touch mes.php
nano index.html
```

```
root@325740a6bff5:/var/www/html# touch index.html
root@325740a6bff5:/var/www/html# touch mes.php
root@325740a6bff5:/var/www/html# ls -la
total 12
drwxrwxrwx 1 www-data www-data 4096 Mar 26 17:08 .
drwxr-xr-x 1 root     root     4096 Mar 17 18:44 ..
-rw-r--r-- 1 root     root      0 Mar 26 17:08 index.html
-rw-r--r-- 1 root     root      0 Mar 26 17:08 mes.php
root@325740a6bff5:/var/www/html#
```

```
root@325740a6bff5:/var/www/html# nano index.html
```

```
GNU nano 5.4
<h1>HOLA, SOY LAURA ALVAREZ CUBILLAS</h1>
index.html *
```

- ✓ La salida del navegador del archivo `index.html` en el puerto establecido



The screenshot shows a web browser window with the address bar set to `localhost:8000`. The page content is `<h1>HOLA, SOY LAURA ALVAREZ CUBILLAS</h1>`.

**HOLA, SOY LAURA ALVAREZ
CUBILLAS**

- ✓ Edición y visualización del archivo `mes.php`, en este caso se escribió la ruta al script en el navegador

```
root@325740a6bff5:/var/www/html# nano mes.php
```

```
GNU nano 5.4 mes.php
<?php
echo 'Estamos en el mes: ';
```

```
echo date("F");
?>
```

- ✓ La salida del navegador del script `mes.php`

```
localhost:8000/mes.php
```

Estamos en el mes: March

- ✓ Para ver el tamaño del contenedor después de crear los ficheros usamos el comando siguiente, donde el flag `-s` nos indica el tamaño de los contenedores que están activos, en este caso lo que hemos creado dentro del contenedor son 19.5 MB.

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

```
laura @ Artico ~
└─ $ docker ps -a -s
```

CONTAINER ID	IMAGE	COMMAND	CREATED
325740a6bff5	php:7.4-apache	"docker-php-entrypoi..."	About an hour ago

STATUS	PORTS	NAMES	SIZE
Up About an hour	0.0.0.0:8000->80/tcp, :::8000->80/tcp	web	19.5MB (virtual 472MB)

Servidor de base de datos

Se crea el contenedor con una instancia de **mariadb** siguiendo las instrucciones de docker hub para definir las variables de entorno solicitadas y que se cree una base de datos llamada `prueba` al iniciar la imagen.

```
docker run -d --name bbdd -e MARIADB_ROOT_PASSWORD=root -e
MARIADB_DATABASE=prueba -e MARIADB_USER=invitado -e
MARIADB_PASSWORD=invitado mariadb:latest
```

```
laura @ Artico ~
└─ $ docker run -d --name bbdd -e MARIADB_ROOT_PASSWORD=root -e MARIADB_DATABASE=prueba
-e MARIADB_USER=invitado -e MARIADB_PASSWORD=invitado mariadb:latest
4b07ed4bae41a1b398ae6cb62cbf63861917c152cb0161710557602b64fc38e7
```

Abrimos una terminal interactiva del contenedor que esta corriendo para conectarnos a la base de datos `bbdd`

```
docker exec -it bdd bash
```

y una vez en ella conectamos con la consola de la base de datos con el usuario invitado que creamos, para ello seguimos las instrucciones de docker hub

```
mysql -uinvitado -pinvitado
```

```
laura @ Artico ~
└─ $ docker exec -it bdd bash
root@4b07ed4bae41:/# mysql -uinvitado -pinvitado
```

Desde la consola de base de datos con el comando `show databases;` vemos el listado de bases de datos, entre ellas se encuentra la base `prueba` creada al iniciar la imagen.

```
show databases;
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.7.3-MariaDB-1:10.7.3+maria-focal mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| prueba |
+-----+
2 rows in set (0.001 sec)
```

y con el comando `status;` podemos ver el usuario invitado con el que estamos conectados a la consola.

```
status;
```

```
MariaDB [(none)]> status;
-----
mysql Ver 15.1 Distrib 10.7.3-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2

Connection id:          4
Current database:
Current user:           invitado@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MariaDB
Server version:         10.7.3-MariaDB-1:10.7.3+maria-focal mariadb.org binary distribution
Protocol version:       10
Connection:             Localhost via UNIX socket
Server character set:   utf8mb4
Db character set:       utf8mb4
Client character set:   latin1
Conn. character set:    latin1
UNIX socket:            /run/mysqld/mysqld.sock
Uptime:                 8 min 19 sec

Threads: 1  Questions: 9  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.018
-----
```

Para saber que comandos usar en la consola hice uso de `--help`

Para salir de la consola de la base de datos se usa `exit;`

```
exit;
```

```
MariaDB [(none)]> exit;
Bye
root@4b07ed4bae41:/#
```

Borrado del contenedor

Al estar el contenedor bbdd corriendo no nos permite ni borrarlo, a no ser que se fuerze el borrado, ni borrar la imagen en la que está basado.

Con `docker ps` se ven los contenedores que hay corriendo y con los siguientes comandos se intentan borrar el contenedor y la imagen y vemos en pantalla que no podemos hacerlo y la salida del error al intentar hacerlo por estar el contenedor corriendo.

```
docker ps
docker rm bbdd
docker rmi mariadb
```

```
laura @ Artico ~
└─ $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
4b07ed4bae41   mariadb:latest "docker-entrypoint.s..." 14 minutes ago Up 14 minutes 3306/tcp
bbdd
laura @ Artico ~
└─ $ docker rm bbdd
Error response from daemon: You cannot remove a running container 4b07ed4bae41a1b398ae6cb62cbf63861917c152cb0161710557602b64fc38e7. Stop the container before attempting removal or force remove
laura @ Artico ~
└─ $ docker rmi mariadb
Error response from daemon: conflict: unable to remove repository reference "mariadb" (must force) - container 4b07ed4bae41 is using its referenced image 665218ee5fdf
```

Para borrar el contenedor primero se para y luego se borra, también se podría borrar con el flag `-f` y forzar el borrado `docker rm -f bbdd`

```
docker stop bbdd
docker rm bbdd
```

```
laura @ Artico ~
└─ $ docker stop bbdd
bbdd
laura @ Artico ~
└─ $ docker rm bbdd
bbdd
laura @ Artico ~
└─ $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
laura @ Artico ~
└─ $
```

Las imágenes que tengo en local actualmente son:

```
laura@Artico:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
laura2alv2cub/web_nginx	1.0	45f6554cb13e	3 days ago	163MB
mariadb	latest	665218ee5fdf	5 days ago	414MB
adminer	latest	1ce5b489aaa5	7 days ago	90.5MB
nginx	1.20.2	8f34c303855f	8 days ago	141MB
php	7.4-apache	91d896afaf11	2 weeks ago	452MB

Al este el último apartado realizado de la tarea, se pueden ver todas las imágenes usadas en todos los ejercicios.