



CICLO FORMATIVO DE GRADO SUPERIOR - TÉCNICO EN ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN REDES

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS

Docker

Nombre y apellidos:
-Wuke Zhang

PRÁCTICA 3. Ejecución avanzada de contenedores. Persistencia.

Ahora que nos hacemos una idea de cómo manejar contenedores, vamos a indagar en la configuración adicional.

1. Redireccionar puertos. (1 pts)

Al desplegar un contenedor basado en un servicio, este se encuentra configurado para utilizar sus puertos por defecto. Sin embargo, estos puertos pueden estar ya ocupados por otros contenedores (porque alojas varias páginas web o recursos por ejemplo) o el propio servidor. Para solucionarlo, podemos emparejar un puerto del ordenador . Ejecuta el siguiente comando, luego abre el navegador y entra en *localhost:8080*.

```
root@iso:/home/ubuntu/Desktop# docker run -d -p 8080:80 httpd:2.4
```

```
/mnt/c/Users/34640/Desktop (11.208s)
docker run -d -p 8080:80 httpd:2.4
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
a1a1b409f475: Download complete
c29f5b76f736: Download complete
830a84f99cc8: Download complete
80350326cd93: Download complete
4f4fb700ef54: Download complete
35b1ecb71608: Download complete
Digest: sha256:3195404327ecd95b2fa0a5d4eac1f2206bb12996fb2561393f91254759e422b9
Status: Downloaded newer image for httpd:2.4
d24667b4d6b3b8ee4e809681dbaea1ebded2662ea1f157f988b45e4e6150b2ac
```

2. Renombrar contenedores. (0.75 puntos)

A medida que el número de contenedores aumenta, empieza a hacer necesario mantener un orden y registro de que contenedor realiza cada tarea. Para ello podemos usar la opción `--name` para asignar un nombre concreto al contenedor.

```
root@iso:/home/ubuntu/Desktop# docker run -d --name docker_web -p 80:80 httpd:2.4
```

```
/mnt/c/Users/34640/Desktop (11.208s)
docker run -d -p 8080:80 httpd:2.4
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
a1a1b409f475: Download complete
c29f5b76f736: Download complete
830a84f99cc8: Download complete
80350326cd93: Download complete
4f4fb700ef54: Download complete
35b1ecb71608: Download complete
Digest: sha256:3195404327ecd95b2fa0a5d4eac1f2206bb12996fb2561393f91254759e422b9
Status: Downloaded newer image for httpd:2.4
d24667b4d6b3b8ee4e809681dbaea1ebded2662ea1f157f988b45e4e6150b2ac
```

3. Editar archivos en el contenedor. (1.5 pts)

De la misma forma que los puertos están configurados por defecto, toda la estructura de directorios se mantiene. Puedes consultar la documentación de la cada imagen en <https://hub.docker.com/>. Particularmente, para httpd se utiliza el directorio /usr/local/apache2/. Prueba a modificar la página web con la siguiente instrucción.

```
root@iso:/home/ubuntu/Desktop# docker exec -it docker_web bash
```

```
root@d00974c31cd7:/usr/local/apache2# cd htdocs/
root@d00974c31cd7:/usr/local/apache2/htdocs# ls
index.html
root@d00974c31cd7:/usr/local/apache2/htdocs# echo "<h1>Prueba Docker</h1>" > index.html
root@d00974c31cd7:/usr/local/apache2/htdocs# exit
```

```
/mnt/c/Users/34640/Desktop (53.197s)
docker exec -it docker_web bash
root@85b72eaba575:/usr/local/apache2# cd htdocs/
root@85b72eaba575:/usr/local/apache2/htdocs# ls
index.html
root@85b72eaba575:/usr/local/apache2/htdocs# echo "<h1>Prueba Docker</h1>" > index.html
root@85b72eaba575:/usr/local/apache2/htdocs# exit
exit
```

```
/mnt/c/Users/34640/Desktop (0.098s)
docker exec -it docker_web cat /usr/local/apache2/htdocs/index.html
<h1>Prueba Docker</h1>
```

4. Gestionar variables de entorno. (1 pts)

Otra de las opciones específicas que nos permite habilitar los contenedores son las variables de ambiente. Estás permiten cargar una serie de variables predefinidas en la imagen con valores que permitan personalizar la ejecución del contenedor.

Un ejemplo básico sería usar la variable de usuario en Ubuntu como en el siguiente comando (opción -e --env).

```
root@iso:/home/ubuntu/Desktop# docker run -it --name prueba -e USUARIO=prueba ubuntu bash
root@bdebffa68967:/# echo $USUARIO
prueba
```

```
/mnt/c/Users/34640/Desktop (0.098s)
docker exec -it docker_web cat /usr/local/apache2/htdocs/index.html
<h1>Prueba Docker</h1>
```

Warpify subshell Ctrl I

```
/mnt/c/Users/34640/Desktop
docker run -it --name prueba -e Usuario=prueba ubuntu bash
root@bc1ec6b82eb1:/# echo $usuario
prueba
root@bc1ec6b82eb1:/# echo $USUARIO
prueba
root@bc1ec6b82eb1:/# echo $Usuario
prueba
root@bc1ec6b82eb1:/#
```

5. Visualizar y utilizar variables de entorno. (1.5 pts)

Ahora bien, por defecto muchas imágenes contienen variables de entorno ya predeterminadas a usar en la configuración. Para ello, hay que revisar la configuración de la imagen en cuestión en [docker.hub](https://hub.docker.com/). Vamos a probar el uso de variables de ambiente para configurar un contenedor de mariadb. Si visitamos https://hub.docker.com/_/mariadb, comprobaremos como nos requiere utilizar la variable `MYSQL_ROOT_PASSWORD` para configurar la contraseña de la base de datos. Ejecutamos la instrucción siguiente para crear el contenedor, incluyendo la variable en cuestión junto con su valor:

```
root@iso:/home/ubuntu/Desktop# docker run -d --name mariadb -e MYSQL_ROOT_PASSWORD=docker1234 mariadb
```

Comprobamos que se ha creado correctamente con el siguiente comando.

```
root@iso:/home/ubuntu/Desktop# docker exec mariadb env
```

```

/mnt/c/Users/34640/Desktop (12.469s)
docker run -d --name mariadb -e MYSQL_ROOT_PASSWORD=docker1234 mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
5c3a22df929b: Download complete
901fe9394c00: Download complete
597f7afe50fe: Download complete
bdec990c29c: Download complete
e1dede558384: Download complete
5db80086e4da: Download complete
43eb19e1b102: Download complete
Digest: sha256:310d29fbb58169dcddb384b0ff138edb081e2773d6e2eceb976b3668089f2f84
Status: Downloaded newer image for mariadb:latest
ae45589e8eeec743771f79a86e1a93893e3a615ff2600422275e98f37aa00299
d

/mnt/c/Users/34640/Desktop (0.107s)
docker exec mariadb env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=ae45589e8eee
MYSQL_ROOT_PASSWORD=docker1234
GOSU_VERSION=1.17
LANG=C.UTF-8
MARIADB_VERSION=1:11.7.2+maria~ubu2404
HOME=/root

```

Por último, comprobamos que podemos acceder a la base de datos.

```
root@iso:/home/ubuntu/Desktop# docker exec -it mariadb bash
```

```

root@3e7267cc905b:/# mariadb -u root -p"$MYSQL_ROOT_PASSWORD"
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 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)]>

```

(Recuerda que, para salir de una sesión en la terminal, hay que ejecutar el comando *exit*).

```

/mnt/c/Users/34640/Desktop
docker exec -it mariadb bash

root@ae45589e8eee:/# mariadb -u root -p"$MYSQL_ROOT_PASSWORD"
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.7.2-MariaDB-ubu2404 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)]>

```

6. Crear persistencia en contenedores. Volúmenes de docker. (2.5 pts)

Como hemos visto, los contenedores son por definición no persistentes ya que se crean y arrancan a partir de imágenes de aplicaciones y utilidades varias. Para

agregarle persistencia necesitamos “conectar” el contenedor a algun medio de almacenamiento.

El primero de los métodos de almacenamiento son los **volúmenes**. Los volúmenes son espacios de almacenamiento que gestiona docker dentro del directorio /var/lib/docker/volumes. Sus ventajas son:

- Restringe el acceso solamente a docker y usuarios del mismo nivel de permisos.
- Permite compartir datos entre contenedores si usan el mismo volumen
- Permite crear backups y ports para otros contenedores.

Vamos ha probar a hacer un volumen. Ejecuta el siguiente comando:

```
docker volume create miweb
```

Comprobamos en la lista de volúmenes:

```
Se7207CC903D - Mariadb - docker-entrypoint.S... About an hour ago Up 7
root@iso:/home/ubuntu/Desktop# docker volume list
DRIVER      VOLUME NAME
local       4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4
local       83f824061320d01d86da5147b5baf39c5a4b06bc820191dc7391e2eca71abe12
local       1028c281a884ec516ae655631de9d6ca6d4cfd1dadf223a53ea9d5be4fe0d523
local       d26f11f8b9185d5228ec367b29cfa5dae4a7bfc824304f21d94234eb930445c8
local       ed0c9bba43acd849c2ff0f612d9767be6362760bd1dfcffe09d42b1495a47cef
local       f58bb17f4b3272a60c24db4015b76553a7985b3fd4ac2304fcaa3c0579650ad9
local       miweb
```

```
/mnt/c/Users/34640/Desktop
docker exec -it mariadb bash
root@ae45589e8eee:/# mariadb -u root -p"$MYSQL_ROOT_PASSWORD"
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.7.2-MariaDB-ubu2404 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)]>
```

Para vincular el contenedor al volumen, usamos la siguiente versión del comando docker run:

```
root@iso:/home/ubuntu/Desktop# docker run -d --name web
--mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs
-p 8080:80 httpd:2.4
```

Modificamos el index.html del contenedor y lo comprobamos:

```
root@iso:/home/ubuntu/Desktop# docker exec web bash
-c 'echo "<h1>Ejemplo de persistencia</h1>" > /usr/local/apache2/htdocs/index.html'

root@iso:/home/ubuntu/Desktop# curl http://localhost:8080
<h1>Ejemplo de persistencia</h1>
```

Eliminamos y volvemos a crear el contenedor, en este caso usaremos la opción -v, alternativa al --mount.

```
root@iso:/home/ubuntu/Desktop# docker rm -f web

root@iso:/home/ubuntu/Desktop# docker run -d --name web -v miweb:/usr/local/apache2/htdocs -p 8080:80 httpd:2.4

root@iso:/home/ubuntu/Desktop# curl http://localhost:8080
<h1>Ejemplo de persistencia</h1>
```

The screenshot shows a terminal window with the following content:

```
/mnt/c/Users/34648/Desktop (0.422s)
docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
docker: Error response from daemon: Conflict. The container name "/web" is already in use by container "e69882574a201ecd948155257fe0faea8784e24b118bcb49e7e66decb28c680". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

Below the terminal output, there is a notification box titled "Remove the existing container named web." with the following text:

The error message indicates that a container named "web" is already present, which is causing the conflict. To proceed with removing the existing container, we need to stop the running container (if it's running) and then remove it.

A green checkmark icon is next to the command: `docker rm -f web`

Below the notification, there is a text box stating: "The existing container named "web" has been successfully removed. You can now proceed to run your Docker command again to start a new container with the name "web". If you want to run it on a different port such as 8081, you can use:

```
1 docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
```

Below the text box, there is a terminal prompt: `bash`

At the bottom of the screenshot, the terminal output shows the successful execution of the command:

```
/mnt/c/Users/34648/Desktop (0.422s)
docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
9c09aa48cfe6d1190b67dba0f826fb8325029ee1971a703c350fbc3d4c04fd7
```


Por último, otros comandos interesantes para manejo de volúmenes son:

- docker volumen inspect

```
root@iso:/home/ubuntu/Desktop# docker volume inspect 4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4
[
  {
    "CreatedAt": "2024-01-09T18:42:04Z",
    "Driver": "local",
    "Labels": {
      "com.docker.volume.anonymous": ""
    },
    "Mountpoint": "/var/lib/docker/volumes/4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4/_data",
    "Name": "4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4",
    "Options": null,
    "Scope": "local"
  }
]
```

```
/mnt/c/Users/34640/Desktop (0.086s)
docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
docker: Error response from daemon: Conflict. The container name "/web" is already in use by container "e69882574a201ecd948155257fe60faea8784e24b118bcb49e7e66decb28c680". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

Remove the existing container named web.

The error message indicates that a container named "web" is already present, which is causing the conflict. To proceed with removing the existing container, we need to stop the running container (if it's running) and then remove it.

```
✓ docker rm -f web
```

The existing container named "web" has been successfully removed. You can now proceed to run your Docker command again to start a new container with the name "web". If you want to run it on a different port such as 8081, you can use:

```
1 docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
```

```
bash
```

```
/mnt/c/Users/34640/Desktop (0.422s)
docker run -d --name web --mount type=volume,src=miweb,dst=/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
9c09aa48cefe6d1190b67dba0f826fb8325029ee1971a703c350fbc3d4c04fd7
```

- docker volumen prune (elimina los volúmenes sin usar):

```
root@iso:/home/ubuntu/Desktop# docker volume ls
DRIVER      VOLUME NAME
local       4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4
local       83f824061320d01d86da5147b5baf39c5a4b06bc820191dc7391e2eca71abe12
local       1028c281a884ec516ae655631de9d6ca6d4cfd1dadf223a53ea9d5be4fe0d523
local       d26f11f8b9185d5228ec367b29cfa5dae4a7bfc824304f21d94234eb930445c8
local       ed0c9bba43acd849c2ff0f612d9767be6362760bd1dfcffe09d42b1495a47cef
local       f58bb17f4b3272a60c24db4015b76553a7985b3fd4ac2304fcaa3c0579650ad9
local       miweb
root@iso:/home/ubuntu/Desktop# docker volume prune
WARNING! This will remove anonymous local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
f58bb17f4b3272a60c24db4015b76553a7985b3fd4ac2304fcaa3c0579650ad9
4d8f7ea0d2b067f47f37cdf68c84fc930326dab76c54fa2aa86e3944d118bef4
83f824061320d01d86da5147b5baf39c5a4b06bc820191dc7391e2eca71abe12
1028c281a884ec516ae655631de9d6ca6d4cfd1dadf223a53ea9d5be4fe0d523

Total reclaimed space: 0B
root@iso:/home/ubuntu/Desktop# docker volume ls
DRIVER      VOLUME NAME
local       d26f11f8b9185d5228ec367b29cfa5dae4a7bfc824304f21d94234eb930445c8
local       ed0c9bba43acd849c2ff0f612d9767be6362760bd1dfcffe09d42b1495a47cef
local       miweb
```



```
/mnt/c/Users/34640/Desktop (5.594s)
```

```
docker volume prune
```

```
WARNING! This will remove anonymous local volumes not used by at least one container.
```

```
Are you sure you want to continue? [y/N] y
```

```
Deleted Volumes:
```

```
ec92e7474083abb17d4114526b16f252c18042398feb07778da6404ab800bed3
```

```
157940b741facb0800323713e0449feef84851ba586ea14ace2b46ea7dcdb53f
```

```
Total reclaimed space: 209.1MB
```

- docker volumen rm

7. Crear persistencia en contenedores. Vincular directorios. (1,75 pts)

La otra alternativa a los volúmenes para generar persistencia en un contenedor consiste en vincular los espacios de directorios.

Para realizar este paso, primero tenemos que crear el directorio a compartir:

```
root@iso:/home/ubuntu/Desktop# mkdir web
root@iso:/home/ubuntu/Desktop# cd web
root@iso:/home/ubuntu/Desktop/web# pwd
/home/ubuntu/Desktop/web
```

Una de las ventajas de este sistema radica en que podemos preparar el directorio directamente para producción, ubicando los archivos de configuración directamente en el directorio incluso antes de crear el contenedor.

```
root@iso:/home/ubuntu/Desktop/web# echo "<h1>Otro ejemplo de persistencia</h1>" > index.html
```

Por último, utilizamos el siguiente comando para crear el contenedor, prestando atención a que las rutas de los directorios están correctas.

```
root@iso:/home/ubuntu/Desktop/web# docker run -d --name web
-v /home/ubuntu/Desktop/web:/usr/local/apache2/htdocs -p 8080:80 httpd:2.4
```

```
/mnt/c/Users/34640/Desktop (0.423s)
docker run -d --name web -v /mnt/c/Users/34640/Desktop:/usr/local/apache2/htdocs -p 8081:80 httpd:2.4
081669a16a6ea7ad03658ea2ffe2b9d0ec64f2ec0c1870d0fad7db8db38b9440

/mnt/c/Users/34640/Desktop (0.097s)
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
081669a16a6e	httpd:2.4	"httpd-foreground"	9 seconds ago	Up 8 seconds	0.0.0.0:8081->80/tcp	web
ae45589e8eee	mariadb	"docker-entrypoint.s..."	About an hour ago	Up About an hour	3306/tcp	mariadb
85b72eaba575	httpd:2.4	"httpd-foreground"	2 hours ago	Up 2 hours	0.0.0.0:80->80/tcp	docker_web
d24667b4d6b3	httpd:2.4	"httpd-foreground"	2 hours ago	Up 2 hours	0.0.0.0:8080->80/tcp	laughing_goldstine