

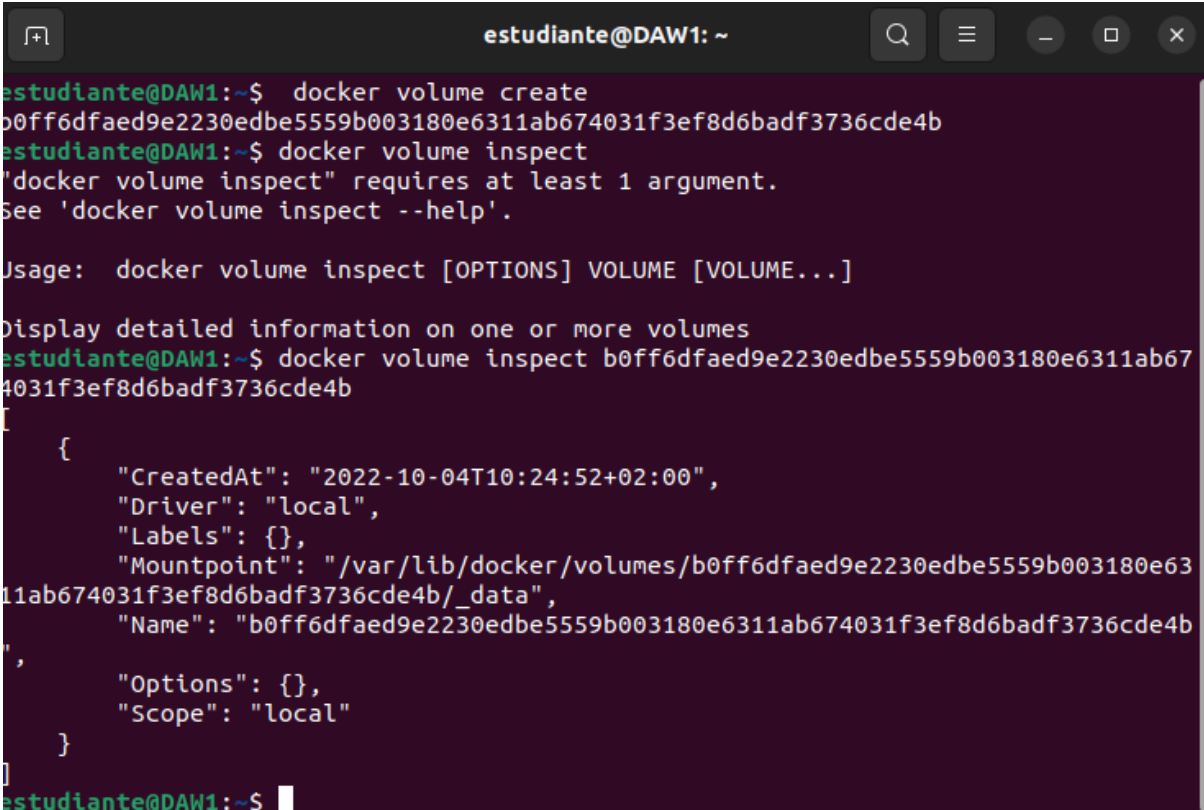
1. ¿Para qué sirven los volúmenes?

Un volumen de contenedor permite conservar los datos, aunque se elimine el Docker container. Los volúmenes también permiten un intercambio práctico de datos entre el host y el container. Crear un volumen de Docker es una buena solución para poder: Transferir datos a un contenedor de Docker

2. ¿Qué diferencias hay entre los volúmenes y los Bind Mount?

Los volúmenes son creados y administrados por docker, a diferencia de bind mount están aislados de nuestra pc y son administrados por docker. Este es un sistema de archivos temporales, estos no se quedan guardados ni en el disco ni en el host de docker, solo se pueden usar mientras el contenedor exista

3. Crea un volumen sin nombre, ¿como lo ha llamado docker?. Muestra la información detallada del volumen y explica qué significa cada linea. Haz lo mismo con un volumen creado por el sistema (lístalos primero).



```
estudiante@DAW1: ~  
estudiante@DAW1:~$ docker volume create  
b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b  
estudiante@DAW1:~$ docker volume inspect  
'docker volume inspect' requires at least 1 argument.  
See 'docker volume inspect --help'.  
  
Usage: docker volume inspect [OPTIONS] VOLUME [VOLUME...]  
  
Display detailed information on one or more volumes  
estudiante@DAW1:~$ docker volume inspect b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b  
[  
  {  
    "CreatedAt": "2022-10-04T10:24:52+02:00",  
    "Driver": "local",  
    "Labels": {},  
    "Mountpoint": "/var/lib/docker/volumes/b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b/_data",  
    "Name": "b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b",  
    "Options": {},  
    "Scope": "local"  
  }  
]  
estudiante@DAW1:~$
```

Nombre: b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b
información:

1. Fecha de creación del volumen
2. Tipo del driver
3. Etiquetas asociadas
4. Punto de montaje
5. Nombre del volumen
6. Opciones del driver
7. Ámbito del volumen

Volumen predeterminado del sistema:

```
]
estudiante@DAW1:~$ docker volume ls
DRIVER      VOLUME NAME
local       724fcf9105a1a0e5b0586845c0010e5aebc9099f552a34f64a9e472b39221e22
local       b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b
estudiante@DAW1:~$ docker volume inspect 724fcf9105a1a0e5b0586845c0010e5aebc9099f552a34f64a9e472b39221e22
[
  {
    "CreatedAt": "2022-09-27T10:29:53+02:00",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/724fcf9105a1a0e5b0586845c0010e5aebc9099f552a34f64a9e472b39221e22/_data",
    "Name": "724fcf9105a1a0e5b0586845c0010e5aebc9099f552a34f64a9e472b39221e22",
    "Options": null,
    "Scope": "local"
  }
]
estudiante@DAW1:~$
```

4. Elimina todos los volúmenes que hayas creado.

```
]
estudiante@DAW1:~$ docker volume rm b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b
b0ff6dfaed9e2230edbe5559b003180e6311ab674031f3ef8d6badf3736cde4b
estudiante@DAW1:~$ docker volume ls
DRIVER      VOLUME NAME
local       724fcf9105a1a0e5b0586845c0010e5aebc9099f552a34f64a9e472b39221e22
estudiante@DAW1:~$
```

5. Arranca un Bind Mount usando la carpeta “web” del usuario como directorio raíz del servidor apache (Haz lo mismo con un volumen). Después obtén información del volumen y el bind mount y explica lo que se te muestra.

```
estudiante@DAW1: ~/web
estudiante@DAW1:~/web$ docker inspect vol1
[
  {
    "CreatedAt": "2022-10-04T10:41:11+02:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/vol1/_data",
    "Name": "vol1",
    "Options": {},
    "Scope": "local"
  }
]
estudiante@DAW1:~/web$ docker inspect apache
[
  {
    "Id": "c77fec38e34d234f546e8d96ee7094c739023f5b6553ae26d11816901ca63baa"
  },
  {
    "Created": "2022-10-04T08:37:27.43945741Z",
    "Path": "httpd-foreground",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2022-10-04T08:37:28.145834842Z",
```

6. Arranca la versión más reciente del contenedor de ubuntu y comprueba que está “up”. Después páralo, comprueba que está parado. Por último, elimina el contenedor de ubuntu.

```
estudiante@DAW1:~$ docker run -t -i ubuntu /bin/bash
root@bb8c9ba6457b:/# exit
exit
estudiante@DAW1:~$
```

```
estudiante@DAW1:~$ docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|---------------|--------------|--------------------------|---------------|-------------|
| bb8c9ba6457b | ubuntu | "/bin/bash" | 6 seconds ago | Up 5 second |
| s | | elegant_wescoff | | |
| cec2595a83af | ubuntu | "bash" | 2 minutes ago | Exited (0) |
| 2 minutes ago | | upbeat_mendel | | |
| 51e311220cec | busybox | "sh" | 7 days ago | Exited (0) |
| 7 days ago | | sad_haslett | | |
| b9854f095c87 | python | "python3" | 7 days ago | Exited (0) |
| 7 days ago | | festive_boyd | | |
| a0e2aaae79e3 | mysql:5.7.22 | "docker-entrypoint.s..." | 8 days ago | Exited (0) |
| 7 days ago | | coches | | |
| 2817b09367fb | hello-world | "/hello" | 3 months ago | Exited (0) |
| 3 months ago | | wizardly_snyder | | |

```
estudiante@DAW1:~$
```

7. Ejecuta el contenedor de apache (busca en Dockerhub) poniéndole nombre “web” ¿qué IP le ha asignado?. Compruébalo.

```
estudiante@DAW1:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
Digest: sha256:71e882df50adc606c57e46e5deb3c933288e2c7775472a639326d9e4e40a47c2
Status: Image is up to date for httpd:latest
docker.io/library/httpd:latest
estudiante@DAW1:~$ docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|--------------|--------------------------|--------------------|------------|----------------|-------|
| 1fcddc695567 | httpd | "httpd-foreground" | About a minute ago | Exited (0) | 34 seconds ago | web |
| bb8c9ba6457b | ubuntu | "/bin/bash" | 13 minutes ago | Exited (0) | 12 minutes ago | elega |
| nt_wescoff | | | | | | |
| cec2595a83af | ubuntu | "bash" | 15 minutes ago | Exited (0) | 15 minutes ago | upbea |
| t_mendel | | | | | | |
| a0e2aaae79e3 | mysql:5.7.22 | "docker-entrypoint.s..." | 8 days ago | Exited (0) | 7 days ago | coche |
| s | | | | | | |
| 2817b09367fb | hello-world | "/hello" | 3 months ago | Exited (0) | 3 months ago | wizar |
| dly_snyder | | | | | | |

```
estudiante@DAW1:~$
```

8. Arranca un contenedor del servicio Tomcat versión jdk11. , llamándolo “Tomcat”, redirigiendolo al puerto 9999 (tomcat usa el puerto 8080). Comprueba que está funcionando.

```
docker run -p 9999:8080 tomcat:jdk11 --name Tomcat
docker ps -a
```

9. Para todos los contenedores que estén funcionando y bórralos.

```
docker container stop $(docker container list -q)
docker container rm -r $(docker container list -q)
```

10. Descarga la imagen mariadb (base de datos) y crea un volumen llamado DATA donde vayamos a guardar datos de mariadb. Comprueba que está creado.

```
docker pull mariadb
docker run mariadb
docker volume create --name DATA
docker ps -a
docker volume -ls
```

11. Arranca un contenedor con el servicio mariadb funcionando... llamado "db1" con redirección de puerto 3336:3306 y haz el montaje en el volumen DATA y destino /var/lib/mysql (carpeta del servidor)... decirle una variable de entorno -e MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=test mariadb

```
docker run -p 3336:3306 mariadb --name db1 -v DATA -e
MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=test mariadb
```

12. Comprueba que el servidor está funcionando

```
docker ps
```

13. Busca en el repositorio de dockerhub y descarga la imagen mysql con una versión no actualizada y obtén información de la misma (explícala) . Muestra las imágenes descargadas hasta ese momento.

```
docker pull mysql:5.7.39
docker inspect ID
docker ps -a
```

14. Borra dos imágenes a la vez en caso de que existan.

```
docker rm mysql db1
```