Actividad 03-10-2022

1. ¿Para qué sirven los volúmenes?

- Sirven para almacenar los datos del contenedor en un sistema de ficheros de docker, al que solo él tendrá acceso

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

- 1. Bind Mount esta gestionado por el usuario mientras que el volumen es gestionado por Docker

- 2. Los volumenes comparte datos entre contenedores, y el Bind Mount entre host y contenedores

3. Crea un volúmen 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).

- Lo ha llamado con un id alfanumérico aleatorio
- La lineas de informacion dan, respectivamente, los siguientes datos:

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

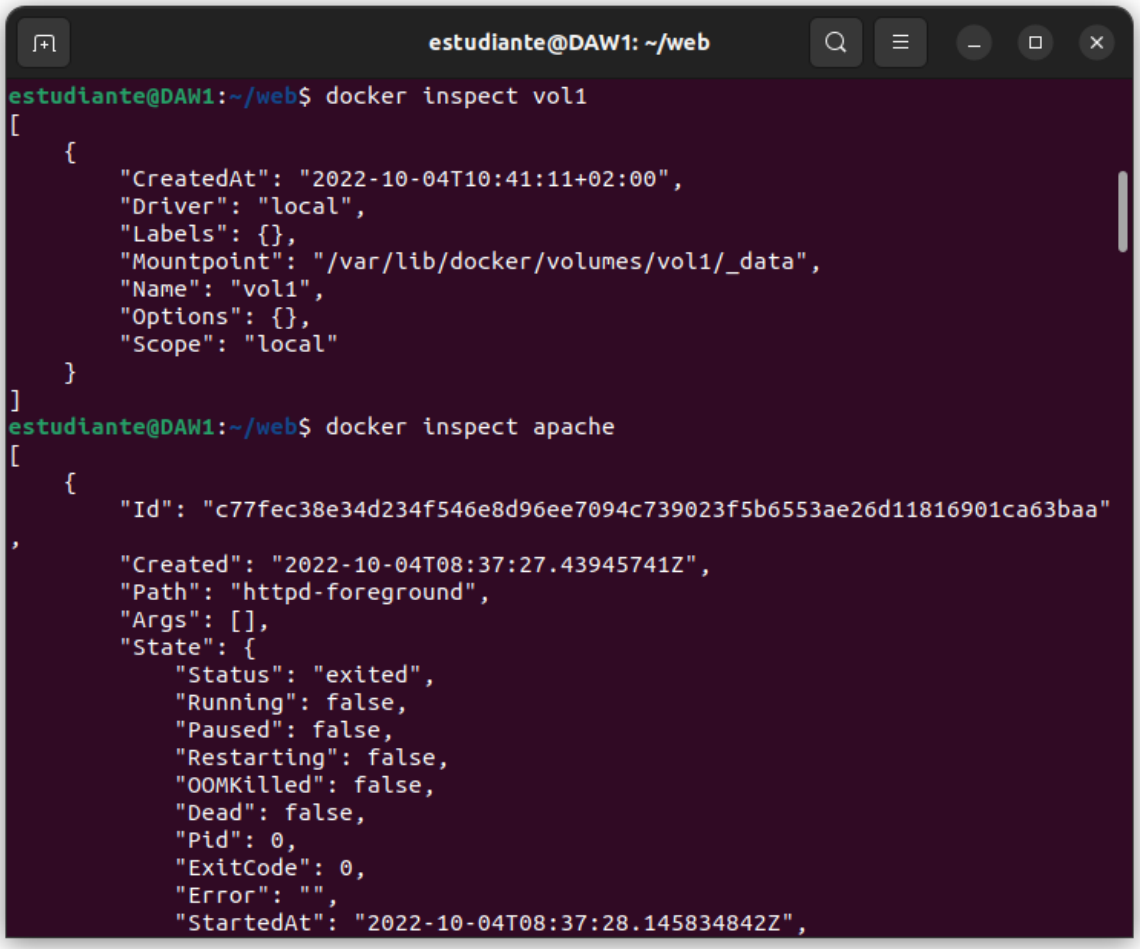
- A continuación, veremos un volumen predeterminado del sistema:

```
estudiante@DAW1:~$ docker volume inspect c298d958136bdb5d39307cf77ecd761f5d16529
03415fa45c224dcdf48e6631c
[
    {
        "CreatedAt": "2022-09-27T10:30:56+02:00",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/c298d958136bdb5d39307cf77ecd761f5
d1652903415fa45c224dcdf48e6631c/_data",
        "Name": "c298d958136bdb5d39307cf77ecd761f5d1652903415fa45c224dcdf48e6631
c",
        "Options": null,
        "Scope": "local"
    }
]
estudiante@DAW1:~$
```

4. Elimina todos los volúmenes que hayas creado.

```
estudiante@DAW1:~$ docker volume rm 4a367f9e92fc72b7d7a65cf41c4987f57d3a2086a352
9bc3deb933e5ae112d31
4a367f9e92fc72b7d7a65cf41c4987f57d3a2086a3529bc3deb933e5ae112d31
estudiante@DAW1:~$ docker volume ls
DRIVER    VOLUME NAME
local     0ec376be9b3605fb9d3eae6f16371c8d4de063b63b0467c19a2b18956275aa06
local     b5a706c6a670dd9705c78e6b73aadd080f668bf6700189365e3acab8c03e74f1
local     c298d958136bdb5d39307cf77ecd761f5d1652903415fa45c224dcdf48e6631c
estudiante@DAW1:~$
```

5. Arranca un Bind Mount usando la carpeta “web” del usuario como directorio raiz 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.



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.

docker run ubuntu

docker ps -a

docker stop ubuntu

docker ps -a

docker rm -r ubuntu

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

De los contenedores que maneja Apache hemos elegido HTTPD para este ejercicio

docker pull httpd –name web

Como podremos comprobar en la terminal, se le asigna una ID alfanumerica aleatoria

docker ps -a

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 volúmen 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