Actividad 03-10-2022

**1. ¿Para qué sirven los volúmenes?**

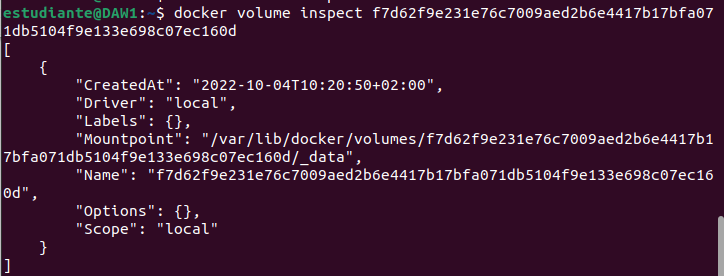
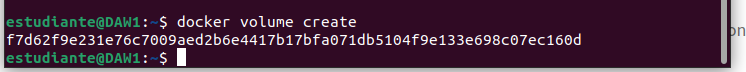
Los volúmenes de Docker son utilizados para persistir los datos que manejan los contenedores. Esto nos permite crear contenedores e indicarles donde tienen que persistir nuestros datos.

**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 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).**

Le asignó un nombre por defecto el sistema

CreatedAt: Fecha de creación

Driver: Es donde esta montado el volumen

Labels: No contiene etiqueta

Mountpoint: Punto de montaje del docker

Name: El nombre que el sistema le asignó por defecto

Options: Opciones adicionales del docker

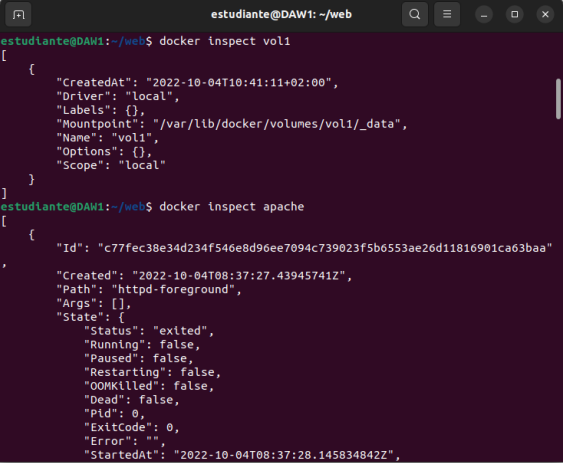
Scope: Alcance del volumen



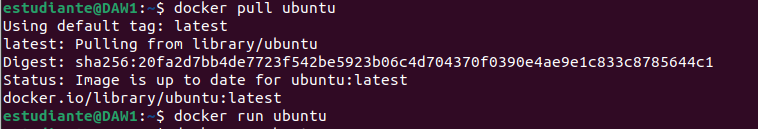
**4. Elimina todos los volúmenes que hayas creado.**

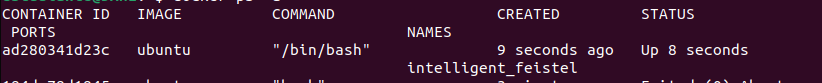


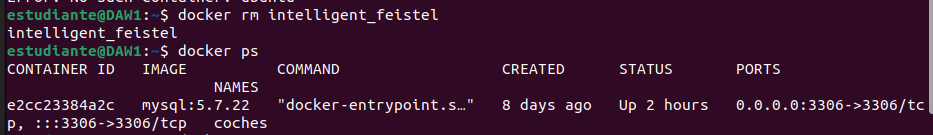
**5. Arranca un Bind Mount usando la carpeta “web” del usuario como directorio raiz del servidor apache (Haz lo mismo con un volumen). Después obtén información del volumen y el bind mount y explica lo que se te muestra.**

****

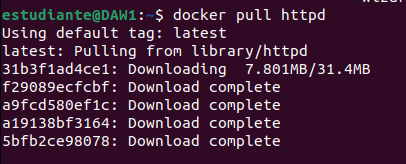
**6. Arranca la versión más reciente del contenedor de ubuntu y comprueba que está “up”. Después páralo, comprueba que está parado. Por último, elimina el contenedor de ubuntu.**

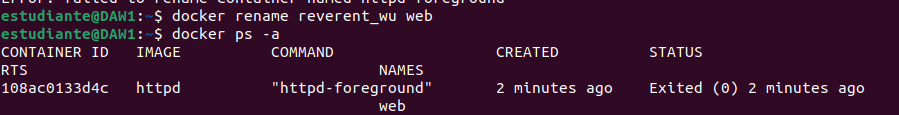






**7. Ejecuta el contenedor de apache (busca en Dockerhub) poniéndole nombre “web” ¿qué IP le ha asignado?. Compruébalo.**



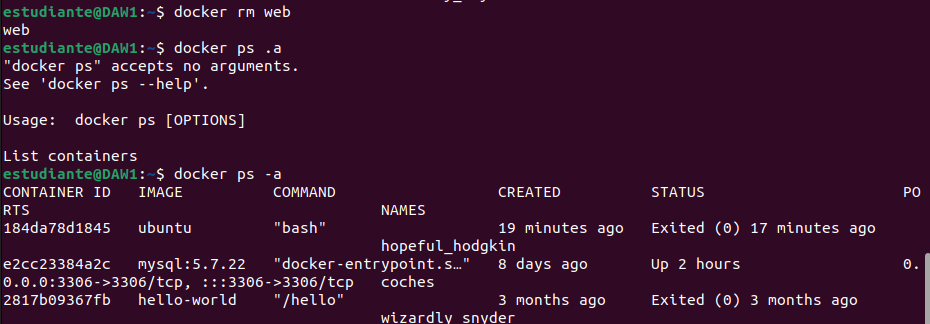


**8. Arranca un contenedor del servicio Tomcat versión jdk11. , llamándolo “Tomcat”, redirigiendolo al puerto 9999 (tomcat usa el puerto 8080). Comprueba que está funcionando.**

docker run -p 9999:8080 tomcat:jdk11 --name Tomcat

docker ps -a

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



**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