

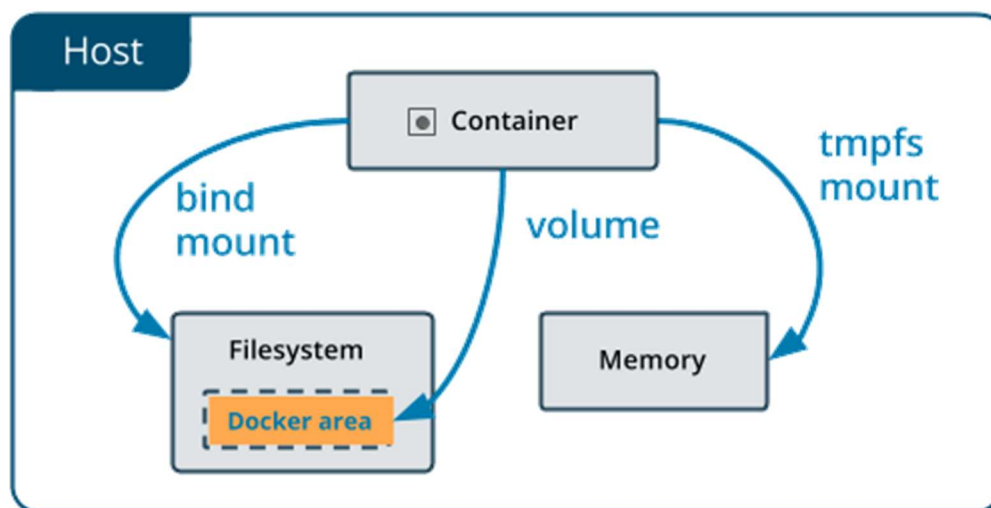
### 1. ¿Para qué sirven los volúmenes?

Los volúmenes se usan para compartir datos entre contenedores; para hacer copias de seguridad, ya sea para que sean usadas posteriormente por otros contenedores o para mover dichos volúmenes a otros hosts; para almacenar datos de un contenedor; y cuando se requiere más rendimiento usando Docker Desktop.

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

Mientras que los bind mounts dependen de un directorio o archivo de la máquina host, los volúmenes son manejados íntegra por Docker.

La siguiente imagen muestra esta relación.



Además, los bind mounts tienen una serie de características propias, como que los cambios ejecutados desde el contenedor sobre los archivos del bind mount reemplazan, por defecto, los archivos originales del host.

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

```
sandra@sandra-UX410UAK:~$ sudo docker volume create
6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5
sandra@sandra-UX410UAK:~$ ^C
sandra@sandra-UX410UAK:~$ sudo docker volume inspect 6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5
[
  {
    "CreatedAt": "2022-10-04T10:25:02+02:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/snap/docker/common/var-lib-docker/volumes/6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5/_data",
    "Name": "6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5",
    "Options": {},
    "Scope": "local"
  }
]
```

Línea 1 → fecha de creación del volumen

Línea 2 → driver del volumen

Línea 3 → labels, que no tiene

Línea 4 → ubicación del volumen

Línea 5 → nombre del volumen

Línea 6 → options del volumen, que no tiene

Línea 7 → scope, que es local

Volúmenes listados:

```
sandra@sandra-UX410UAK:~$ sudo docker volume ls
DRIVER      VOLUME NAME
local       6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5
local       8a13d867f089d3ed44a04656a439863377d3938d70e4b39c9e59892093e3bf93
local       13caed894ae14ed5f14a3e4b4da1d7923e78f84c59e4aaff0dcf32c16c5eabea
local       74ba4d5be6bba68fa5a646fcb4c882423015507e0e2eba71996728c80f943d8d
local       488ebbbded6a74f9c6ec09efef963685720aea6bf3ae7b98258054f7b2d3bc05b
local       7115e05bb44e661c64c744cf3233fbc3ff0ea5decea8235ebd4a9d1ffb9c96cd
local       40453d319dcac8ffe087633f1c53abed9e913fc4890ef492afe9c551eebd24ec
local       cc3e537dfd6125a12538b954adb8e3648308cafcfbf1deb591390cf7d8a7d058a
sandra@sandra-UX410UAK:~$
```

Información detallada del volumen:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker volume inspect 7115e05bb44e661c64c744cf3233fbc3ff0ea5decea8235ebd4a9d1ffbbc96cd  
[  
  {  
    "CreatedAt": "2022-09-27T09:25:20+02:00",  
    "Driver": "local",  
    "Labels": null,  
    "Mountpoint": "/var/snap/docker/common/var-lib-docker/volumes/7115e05bb44e661c64c744cf3233fbc3ff0ea5decea8235ebd4a9d1ffbbc96cd/_data",  
    "Name": "7115e05bb44e661c64c744cf3233fbc3ff0ea5decea8235ebd4a9d1ffbbc96cd",  
    "Options": null,  
    "Scope": "local"  
  }  
]  
sandra@sandra-UX410UAK:~$
```

4. Elimina todos los volúmenes que hayas creado.

```
sandra@sandra-UX410UAK:~$ sudo docker volume rm 6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5  
6f18c87d48304d1d35fcdef32a2da66206c97d85a5b29d974abdeead307178c5
```

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.

**BIND MOUNT:** `docker run --name apache -p 80:80 --mount type=bind,src=/home/sandra/web,dst=/usr/local/apache2/htdocs httpd`

Da error:

```
sandra@sandra-UX410UAK:~$ sudo docker run --name apache -p 80:80 --mount type=bind,src=/home/sandra/web,dst=/usr/local/apache2/htdocs httpd  
[sudo] password for sandra:  
docker: Error response from daemon: invalid mount config for type "bind": bind source path does not exist: /home/sandra/web.  
See 'docker run --help'.
```

VOLUMEN: `docker run --name apache -p 80:80 --mount type=volume,dst=/home/sandra/web,dst=/usr/local/apache2/htdocs httpd`

`docker inspect apache`

```
"Mounts": [
  {
    "Type": "volume",
    "Name": "b9af68529583672dcbceacc91beadccedfc40dc3ebf281d800f4f61d2bb71f6f",
    "Source": "/var/snap/docker/common/var-lib-docker/volumes/b9af68529583672dcbceacc91beadccedfc40dc3ebf281d800f4f61d2bb71f6f/_data",
    "Destination": "/usr/local/apache2/htdocs",
    "Driver": "local",
    "Mode": "z",
    "RW": true,
    "Propagation": ""
  }
],
```

Explico lo que se ve en el bind mount a partir de un ejemplo, ya que me ha dado error al arrancarlo:

**BIND MOUNT EN USO**

```
"Mounts": [
  {
    "Type": "bind",
    "Source": "/home/pekechis/web",
    "Destination": "/usr/local/apache2/htdocs",
    "Mode": "",
    "RW": true,
    "Propagation": "rprivate"
  }
],
```

Type: es el tipo, en este caso un bind.

Source: la fuente

Destination: el destino

Mode: modo, que no tiene

Rw: como es true, está montado en modo Read-Write.

Propagation: propagación, de tipo "rprivate".

**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.**

Arranco:

```
sandra@sandra-UX410UAK: ~
sandra@sandra-UX410UAK:~$ sudo docker start libreriaJacaranda
libreriaJacaranda
sandra@sandra-UX410UAK:~$
```



Compruebo que está up:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker ps  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS  
PORTS          NAMES  
3e8238d7e2b0   mysql:5.7.22   "docker-entrypoint.s..." 12 days ago   Up 34 seconds  
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp   libreriaJacaranda  
sandra@sandra-UX410UAK:~$
```

Paro y compruebo que está parado:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker stop libreriaJacaranda  
libreriaJacaranda  
sandra@sandra-UX410UAK:~$ sudo docker ps  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES  
sandra@sandra-UX410UAK:~$
```

Para eliminarlo:

`docker rm mysql`

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

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker run -d --name web -p 3306:3306 httpd:2.4.54  
Unable to find image 'httpd:2.4.54' locally  
2.4.54: Pulling from library/httpd  
bd159e379b3b: Pull complete  
36d838c2f6d6: Pull complete  
b55eda22bb18: Pull complete  
f6e6bfa28393: Pull complete  
a1b49b7ecb8a: Pull complete  
Digest: sha256:4400fb49c9d7d218d3c8109ef721e0ec1f3897028a3004b098af587d565f4ae5  
Status: Downloaded newer image for httpd:2.4.54  
ed13d1c89b140d14b18f44c32d80bc1dddfc33026ae98c4b6f03c7c6f771eff7  
sandra@sandra-UX410UAK:~$
```

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker ps -l  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS  
PORTS          NAMES  
ed13d1c89b14   httpd:2.4.54   "httpd-foreground"      36 seconds ago Up 34 seconds  
80/tcp, 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp   web  
sandra@sandra-UX410UAK:~$
```

IP: 0.0.0.0

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

No lo he hecho en la Terminal para no tener conflictos con el Tomcat que ya tengo.

El comando sería:

`docker run -d --name Tomcat -p 999:999 httpd:10.1.0-jdk11-temurin-jammy`

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

Para pararlos: `docker stop $(docker ps -a -q)`

Para borrarlos: `docker rm $(docker ps -a -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.

Creación:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker pull mariadb  
Using default tag: latest  
latest: Pulling from library/mariadb  
cf92e523b49e: Pull complete  
11a7b642a1b0: Pull complete  
d05db1f7ddc9: Pull complete  
043662c3afa1: Pull complete  
de48eea20795: Pull complete  
1a40b9e7476d: Pull complete  
d053ff7fa7cc: Pull complete  
f4459f17c9a8: Pull complete  
05ae67b7d96a: Pull complete  
9bd55ebdb8b3: Pull complete  
baf1cda74ce3: Pull complete  
Digest: sha256:59ef1139afa1ec26f98e316a8dbef657daf9f64f84e9378b190d1d7557ad2feb  
Status: Downloaded newer image for mariadb:latest  
docker.io/library/mariadb:latest  
sandra@sandra-UX410UAK:~$
```

Creación del volumen:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker volume create dataMariadb  
dataMariadb
```

Comprobación:

```
sandra@sandra-UX410UAK:~$ sudo docker volume ls
DRIVER      VOLUME NAME
local       8a13d867f089d3ed44a04656a439863377d3938d70e4b39c9e59892093e3bf93
local       13caed894ae14ed5f14a3e4b4da1d7923e78f84c59e4aaff0dcf32c16c5eabea
local       74ba4d5be6bba68fa5a646fcb4c882423015507e0e2eba71996728c80f943d8d
local       488ebbded6a74f9c6ec09efef963685720aea6bf3ae7b98258054f7b2d3bc05b
local       7115e05bb44e661c64c744cf3233fbc3ff0ea5decea8235ebd4a9d1ffbbc96cd
local       40453d319dcac8ffe087633f1c53abed9e913fc4890ef492afefc551eebd24ec
local       b9af68529583672dcbceacc91beadccedfc40dc3ebf281d800f4f61d2bb71f6f
local       cc3e537dfd6125a12538b954adb8e3648308cafcbf1deb591390cf7d8a7d058a
local       dataMariadb
sandra@sandra-UX410UAK:~$
```

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 -it -d --name db1:/var/lib/mysql -p 3306:3306 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.

```
sandra@sandra-UX410UAK:~$ sudo docker pull mysql:5.7
5.7: Pulling from library/mysql
0056409b8e89: Pull complete
219bd535343d: Pull complete
f220ee65eb90: Pull complete
7bbb395b2290: Pull complete
645e487e5f0a: Pull complete
266b96d99da2: Pull complete
5b1965015916: Pull complete
5f5976bb40e4: Pull complete
d7dde9165879: Pull complete
604d9c61b759: Pull complete
cdd75db38492: Pull complete
Digest: sha256:a85b8313feb7298ae240c4beb33a1b4d2e3a3867d3195bab9ed9346d332217c7
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
sandra@sandra-UX410UAK:~$
```



```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker inspect mysql:5.7  
[  
  {  
    "Id": "sha256:eb175b0743cc4475f4440b3d9dacbe7e3f29dc455a60999ce260366709bb1d00",  
    "RepoTags": [  
      "mysql:5.7"  
    ],  
    "RepoDigests": [  
      "mysql@sha256:a85b8313feb7298ae240c4beb33a1b4d2e3a3867d3195bab9ed9346d332217c7  
",  
    ],  
    "Parent": "",  
    "Comment": "",  
    "Created": "2022-10-07T21:08:18.344035624Z",  
    "Container": "792afe12e49c9a55524938c3417e8ea1c103778a3151f0a83271a4c7d954104c",  
    "ContainerConfig": {  
      "Hostname": "792afe12e49c",  
      "Domainname": "",  
      "User": "",  
      "AttachStdin": false,  
      "AttachStdout": false,  
      "AttachStderr": false,  
      "ExposedPorts": {  
        "3306/tcp": {},  
        "33060/tcp": {}  
      },  
      "Tty": false,  
      "OpenStdin": false,  
      "StdinOnce": false,  
      "Env": [  
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",  
        "GOSU_VERSION=1.14",  
        "MYSQL_MAJOR=5.7",  
        "MYSQL_VERSION=5.7.39-1.el7",  
        "MYSQL_SHELL_VERSION=8.0.30-1.el7"  
      ],  
      "Cmd": [  
        "/bin/sh",  
        "-c",  
        "#(nop) ",  
        "CMD [\"mysql\\d\"]"  
      ],  
      "Image": "sha256:ec90e6c654bb89a8d6a5559fcbb612a21f9452701629778bf14f712a472c6  
1ca",  
      "Volumes": {  
        "/var/lib/mysql": {}  
      },  
      "WorkingDir": "",  
      "Entrypoint": [  
        "docker-entrypoint.sh"  
      ]  
    }  
  ]  
]
```



Ver las imágenes descargadas:

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ sudo docker images  
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE  
mysql                5.7             eb175b0743cc   40 hours ago   433MB  
mariadb              latest          14f1097913ec   3 days ago     384MB  
httpd                2.4.54         d16a51d08814   4 days ago     145MB  
redis                latest          9da089657551   2 weeks ago    117MB  
busybox              latest          2bd29714875d   3 weeks ago    1.24MB  
httpd                latest          f2789344c573   3 weeks ago    145MB  
ubuntu               latest          2dc39ba059dc   5 weeks ago    77.8MB  
alpine               latest          9c6f07244728   2 months ago   5.54MB  
mysql                8.0             5b4c624c7fe1   8 months ago   519MB  
mysql                5.7.22          6bb891430fb6   4 years ago    372MB  
store/oracle/database-enterprise 12.2.0.1       12a359cd0528   5 years ago    3.44GB  
sandra@sandra-UX410UAK:~$
```

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

`docker rmi mysql:8.0.22 mysql:5.7`