

Chuletario persistencia en docker

Román Millán Díaz

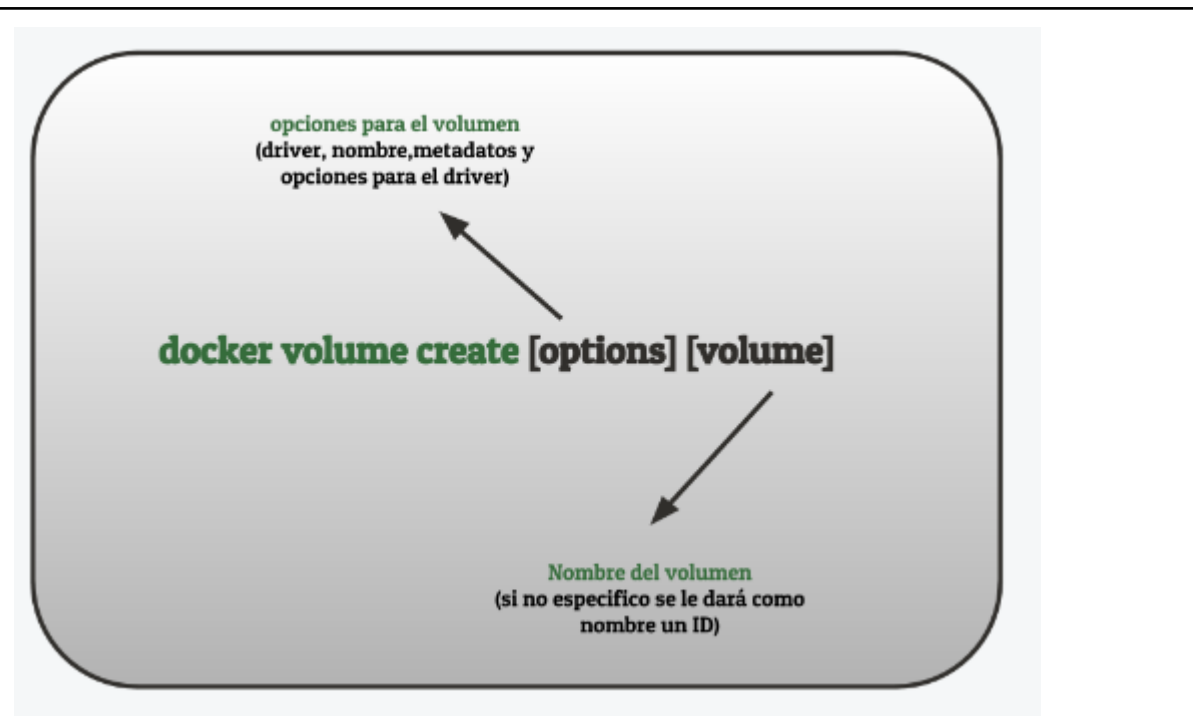
Los VOLÚMENES docker.

Los BIND MOUNT.

Existen otras formas como los tmpfs mounts para Linux y los named pipes para Windows

volúmenes	usando volúmenes estamos haciendo que los datos de los contenedores que nosotros decidamos se almacenen en una parte del sistema de ficheros que es gestionada por docker y a la que, debido a sus permisos, sólo docker tendrá acceso.
<p>Esa "ZONA RESERVADA" de docker cambia de un sistema operativo a otro y también puede cambiar dependiendo de la forma de instalación, pero de manera general podemos decir que es:</p> <ul style="list-style-type: none">• <code>/var/lib/docker/volumes</code> en las distribuciones de Linux si lo hemos instalado desde paquetes estándar.• <code>/var/snap/docker/common/var-lib-docker/volumes</code> en Linux si hemos instalado docker mediante snap (no lo recomiendo).• <code>C:\ProgramData\docker\volumes</code> en las instalaciones de Windows.• <code>/var/lib/docker/volumes</code> también en Mac aunque se requiere que haya una conexión previa a la máquina virtual que se crea.	
Bind Mounts	usando bind mount lo que estamos haciendo es "mapear" una parte de mi sistema de ficheros, de la que yo normalmente tengo el control, con una parte del sistema de ficheros del contenedor.

CREACIÓN DE VOLÚMENES



Ejemplos:

Creación de un volumen llamado datos (driver local sin opciones)

> docker volume create data

Creación de un volumen data especificando el driver local

> docker volume create -d local data

Creación de un volumen llamando web añadiendo varios metadatos

> docker volume create --label servicio=http --label server=apache Web

ELIMINACIÓN DE VOLÚMENES

Ejemplos:

Borrar un volumen por nombre

> docker volume rm nombre_volumen

Borrar un volumen por ID

> docker volume rm
a5175dc955cfcf7f118f72dd37291592a69915f82a49f62f83666ddc81f67441

Borrar dos volúmenes de una sola vez

> docker volume rm nombre_volumen1 nombre_volumen2

Forzar el borrado de un volumen -f o --force

```
> docker volume rm -f nombre_volumen
```

```
# Borrar todos los volúmenes que no tengan contenedores asociados
```

```
> docker volume prune
```

```
# Borrar todos los volúmenes que no tengan contenedores asociados sin pedir confirmación (-f o --force)
```

```
> docker volume prune -f
```

```
# Borrar todos los volúmenes sin usar que contengan cierto valor de etiqueta (--filter)
```

```
> docker volume prune --filter label=valor
```

NOTA: NO SE PUEDEN ELIMINAR VOLÚMENES EN USO POR CONTENEDORES, salvo que usemos el flag -f o --force y no es algo recomendado.

OBTENCIÓN DE INFORMACIÓN DE LOS VOLÚMENES

- Usando **docker volume ls** que nos proporciona una lista de los volúmenes creados y algo de información adicional.
- Usando **docker volume inspect** que nos dará una información mucho más detallada de el volumen que hayamos elegido.

```
# Información detallada de un volumen por nombre
```

```
> docker volume inspect nombre_volumen
```

```
# Información detallada de un volumen por ID
```

```
> docker volume inspect
```

```
a5175dc955cfcf7f118f72dd37291592a69915f82a49f62f83666ddc81f67441
```

Asociando almacenamiento a los contenedores

- El flag **--volume** o **-v**. Este flag lo utilizaremos para establecer bind mounts.
- El flag **--mount**. Este flag nos servirá para establecer bind mounts y para usar volúmenes previamente definidos (entre otras cosas).

BIND MOUNT (flag -v): La carpeta web del usuario será el directorio raíz del servidor apache. Se crea si no existe

```
> docker run --name apache -v  
/home/usuario/web:/usr/local/apache2/htdocs -p 80:80 httpd
```

BIND MOUNT (flag --mount): La carpeta web del usuario será el directorio raíz del servidor apache. Se crea si no existe

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

VOLUME (flag --mount). Mapear el volumen previamente creado y que se llama Data en la carpeta raíz del servidor apache

```
> docker run --name apache -p 80:80 --mount  
type=volume,src=Data,dst=/usr/local/apache2/htdocs httpd
```

VOLUME (flag --mount). Igual que el anterior pero al no poner nombre de volumen se crea uno automáticamente (con un ID como nombre)

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

Diferencias entre VOLUMEN y BIND MOUNT:

VOLUMEN EN USO

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

BIND MOUNT EN USO

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

