

Practica 6 – Docker Volumes

MARKEL ORALLO NOGUEIRA

Crea un volumen (demovol) e inspecciona su metadata. Intenta interpretar la Información.

```
morallo@morallo-VirtualBox:/var/lib/docker$ docker volume create demovol
demovol
morallo@morallo-VirtualBox:/var/lib/docker$ docker volume inspect demovol
[
  {
    "CreatedAt": "2021-09-28T00:19:56+02:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/demovol/_data",
    "Name": "demovol",
    "Options": {},
    "Scope": "local"
  }
]
```

Creamos el volumen “demovol” e inspeccionamos los metadatos mediante el comando docker volumen inspect. Nos da la siguiente información:

- CreatedAt: timestamp de creación del volumen.
- Driver: Que driver utiliza el volumen. En este caso usa el driver “local” que es el driver por defecto.
- Labels: Etiquetas que tiene el volumen. En este caso ninguna ya que no hemos especificado ningún etiquetado de dicho volumen.
- Mountpoint: Directorio local del host de docker donde se encuentra el volumen.
- Name: Nombre del volumen
- Scope: “local”. Indica que el volumen solo tiene que ser creado en el host local.

Pon en marcha un contenedor que monte este volumen y mira el contenido del contenedor en el directorio.

Ej. `$ docker container run -it -v demovol:/demo centos:7 bash`

`#ls`

```
morallo@morallo-VirtualBox:/var/lib/docker$ docker run -it -v demovol:/demo centos:7 /bin/bash
Unable to find image 'centos:7' locally
7: Pulling from library/centos
2d473b07cdd5: Pull complete
Digest: sha256:9d4bcbbb213dfd745b58be38b13b996ebb5ac315fe75711bd618426a630e0987
Status: Downloaded newer image for centos:7
[root@9c6fce44a585 /]# ls
anaconda-post.log  demo  etc  lib  media  opt  root  sbin  sys  usr
bin                dev  home  lib64  mnt  proc  run  srv  tmp  var
[root@9c6fce44a585 /]#
```

Encuentra el path donde está el directorio montado e interpreta la información que proporciona:

```
[root@9c6fce44a585 /]# cd demo/
[root@9c6fce44a585 demo]# ls
[root@9c6fce44a585 demo]# pwd
/demo
[root@9c6fce44a585 demo]#
```

El directorio donde encontramos nuestro volumen es /demo (tal y como hemos indicado al ejecutar el comando docker run)

Escribe un fichero en el volumen que hemos montado

Ej. echo 'dummy file' > /demo/mydata.dat

```
[root@9c6fce44a585 demo]# pwd
/demo
[root@9c6fce44a585 demo]# echo "dummy file" > mydata.dat
[root@9c6fce44a585 demo]# ls
mydata.dat
[root@9c6fce44a585 demo]# cat mydata.dat
dummy file
[root@9c6fce44a585 demo]#
```

Sal del contenedor y lista el contenido del directorio donde hemos montado el volume (Mountpoint). Encuentras el mydata. dat?

```
morallo@morallo-VirtualBox:/var/lib/docker$ sudo ls /var/lib/docker/volumes/demovol/_data
mydata.dat
morallo@morallo-VirtualBox:/var/lib/docker$
```

Como podemos mydata.dat esta en el mountpoint en el que hemos creado el volumen.

Borra el contenedor

```
morallo@morallo-VirtualBox:/var/lib/docker$ docker rm 9c6fce44a585
9c6fce44a585
```

Pon en marcha otro contenedor y monta el mismo volumen y únete a su shell. Comprueba que el fichero anterior sigue estando en el nuevo contenedor.

```
morallo@morallo-VirtualBox:/var/lib/docker$ docker run -it -v demovol:/demo centos:7 /bin/bash
[root@93e248522680 /]# ls
anaconda-post.log demo etc lib media opt root sbin sys usr
bin dev home lib64 mnt proc run srv tmp var
[root@93e248522680 /]# cd demo/
[root@93e248522680 demo]# ls
mydata.dat
[root@93e248522680 demo]# cat mydata.dat
dummy file
[root@93e248522680 demo]#
```

Como podemos ver el fichero existe en el nuevo contenedor. Esto se debe a que los datos persisten en el host de docker al haber almacenado la información en un volumen.

Intenta borrar el volumen. Te deja? Por qué?

```
morallo@morallo-VirtualBox:/var/lib/docker$ docker volume remove demovol
Error response from daemon: remove demovol: volume is in use - [93e248522680dd60a69f1b620283245a5a364f34332d56758fc99e9cf722abf7]
morallo@morallo-VirtualBox:/var/lib/docker$
```

No deja borrar el volumen porque el ultimo contenedor que hemos creado tiene el volumen en uso. Hasta que el contenedor que esta utilizando el volumen no se elimine no nos dejara borrar el volumen.

Para usuarios de OS X:

Docker corre en una maquina virtual a la que podemos acceder a través del comando:

```
stty -echo -icanon && nc -U ~/Library/Containers/com.docker.docker/Data/debug-shell.sock && stty sane
```

Se nos abrirá una terminal dentro de la maquina virtual de docker. Alli encontraremos el directorio `/var/lib/docker` donde se guarda nuestro volumen (Se han seguido los mismos pasos que en Ubuntu para crear el volumen, por no poner dos veces lo mismo omito las capturas).

```
✓ markel in com.docker.docker/Data/ > stty -echo -icanon && nc -U ~/Library/Containers/com.docker.docker/Data/debug-shell.sock && stty sane
[/ # uname -a
Linux docker-desktop 5.10.47-linuxkit #1 SMP Sat Jul 3 21:51:47 UTC 2021 x86_64
Linux
/ #
```

Nuestro dummy file:

```
/ # cd /var/lib/docker/
/var/lib/docker # cd volumes/demovol/_data/
/var/lib/docker/volumes/demovol/_data # ls
mydata.dat
/var/lib/docker/volumes/demovol/_data # cat mydata.dat
dummy file
/var/lib/docker/volumes/demovol/_data #
```