Persistiendo datos

Por defecto ya hemos indicado que un contenedor está aislado de todo. Hemos visto como podemos conectar el contenedor a un puerto de red para poder acceder a él. Eso incluye al sistema de archivos que contiene. De tal manera que si se elimina el contenedor, se eliminan también sus archivos.

Si queremos almacenar datos (una web, una base de datos, etc.) dentro de un contenedor necesitamos una manera de almacenarlos sin perderlos.

Docker ofrece tres maneras:

- A través de **volúmenes**, que son objetos de Docker como las imágenes y los contenedores.
- Montando un **directorio** de la máquina anfitrión dentro del contenedor.
- Almacenándolo en la **memoria del sistema** (aunque también se perderían al reiniciar el servidor).

Lo normal es usar volúmenes, pero habrá ocasiones en que es preferible montar directamente un directorio de nuestro espacio de trabajo. Por ejemplo, para guardar los datos de una base de datos usaremos volúmenes, pero para guardar el código de una aplicación o de una página web montaremos el directorio.

La razón para esto último es que tanto nuestro entorno de desarrollo como el contenedor tengan acceso a los archivos del código fuente. Los volúmenes, al contrario que los directorios montados, no deben accederse desde la máquina anfitrión.

Crear un volumen

Como en la siguiente sección necesitaremos crear una base de datos para instalar un blog con *WordPress* vamos a crear un volumen donde guardar la información:

```
$ docker volume create wordpress-db
wordpress-db
```

Listar volúmenes

Con `docker volume ls` podemos visualizar todos los volúmenes disponibles.

```
$ docker volume ls
DRIVER              VOLUME NAME
local               wordpress-db
```

Visualizar volúmenes

Los volúmenes se crean en un directorio del sistema y no es recomendable acceder a él, no al menos mientras haya un contenedor usándolo. En cualquier caso, si queremos ver los metadatos de un volumen podemos usar `docker volume inspect`

```
$ docker volume inspect wordpress-db
[
    {
        "CreatedAt": "yyyy-mm-ddThh:ii:ss+Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/wordpress-db/_data",
        "Name": "wordpress-db",
        "Options": {},
        "Scope": "local"
    }
]
```

Borrar volúmenes

Como todos los objetos de *Docker*, los volúmenes también pueden ser borrados, pero solo si no están en uso. Mucha precaución al borrar los volúmenes, porque perderíamos todos los datos que contenga.

Para borrar un contenedor usaremos `docker volume rm` y el nombre del volumen.

```
$ docker volume rm wordpress-db
```

Para saber más

Se puede profundizar mucho más en el tema de volúmenes en Docker, pero para el propósito de nuestro curso es suficiente con lo visto hasta aquí. Si quieres profundizar más en el tema, así como aprender cómo trata Docker el "Networking" puedes consultar este documento.

Redes Y volúmenes

En este cheatsheet tienes los principales comandos de volúmenes y networking:

Cheatsheet

Y aquí tienes esta práctica para afianzar conocimientos:

Caso práctico 02 - Balanceo de carga con HAProxy