

xisten plugins en Docker para permitir a los contenedores integrarse con sistemas de almacenamiento externo como AWS EBS que persisten los datos del contenedor más allá de la vida del contenedor en sí mismo. También es posible crear tu propio plugin para la gestión de contenedores de datos en Docker.

Un plugin para volúmenes hace uso de los argumentos `-v` y `--volume-driver=DRIVER_NAME` cuando hacemos uso del comando 'docker run'. Estos plugins requieren un nombre para el volumen que puede ser pasado con el argumento `-v` y con el siguiente formato `-v VOLUME_NAME:/misdatos`. Este nombre permite asociar el volumen con el volumen persistente gestionado por el plugin y así ser capaz de re-utilizarlo cuando sea conveniente. El nombre del volumen no puede empezar por '/'. Uno de los drivers existentes es Flocker:

```
$ docker run -ti -v VOLUME_NAME:/misdatos --volume-driver=flocker busybox sh
```

## Directivas para crear tu propio Driver

Para ello debemos de pasar un campo `VolumeDriver` cuando llamemos a la ruta de la Docker API (`/containers/create`) durante la creación de un contenedor. Este campo permite especificar el nombre de driver, es de tipo string, y su valor por defecto es `local`.

Cualquier driver para volúmenes en Docker tiene que implementar los siguientes puntos de entrada:

- **/VolumeDriver.Create:** Como comentamos previamente la creación de un volumen necesita el nombre del volumen.
  - Argumento de entrada: 'volume\_name'; Argumento de salida: 'error'
- **/VolumeDriver.Remove:** Docker necesita el nombre del volumen para borrarlo. - Argumento de entrada: 'volume\_name'; Argumento de salida: 'error'
- **/VolumeDriver.Path:** Docker necesita saber el directorio donde está el volumen en el host.
  - Argumento de entrada: 'volume\_name'; Argumento de salida: 'Mountpoint', 'error'
- **/VolumeDriver.Unmount:** Esta operación se usa para indicar que Docker no va a usar el volumen. Esta operación es llamada cuando usamos `docker stop`. - Argumento de entrada: 'volume\_name'; Argumento de salida: 'error'

Ejercicios para casa:

1. Crear un contenedor mysql y pasar como volumen un directorio de vuestro host con la directiva `-v`.
2. Crear un contenedor que hace uso de otro contenedor con la directiva `volume-from`.
3. Crear un Flocker cluster de 2 nodes en vuestra cuenta de AWS. Después intenta crear contenedores de datos y moverlos de un host a otro usando la cli de Flocker.