Docker Básico

Arturo Silvelo

Try New Roads

Volúmenes

¿Qué son los volúmenes en Docker?

Los **volúmenes** en Docker son un mecanismo para almacenar y persistir datos.

- **Persistir datos**: Los datos no se pierden al detener o eliminar un contenedor.
- **Compartir datos entre contenedores**: Permiten que varios contenedores accedan al mismo conjunto de datos.

Independientes del contenedor: Los volúmenes existen de forma separada, por lo que los datos permanecen incluso si el contenedor se elimina.

Ejemplo práctico: Persistencia de datos en MongoDB

- Si ejecutamos un contenedor de MongoDB sin un volumen, **todos los datos se perderán** si el contenedor se elimina.
- Al usar un volumen, los datos se almacenan de forma persistente en el sistema del host.

```
docker run -d -p 27017:27017 --name mongodb -v mongodb_data:/data/db mongo
```

- En este ejemplo, el volumen `mongodb_data` almacena los datos persistentes de MongoDB.
- Incluso si el contenedor se elimina, los datos quedan guardados en el volumen y pueden reutilizarse en un nuevo contenedor.

Tipos de Volúmenes en Docker

Volúmenes gestionados por Docker (Named Volumes)

- Son volúmenes creados y gestionados automáticamente por Docker, sin necesidad de configurarlos manualmente.
- Se almacenan en una ubicación predeterminada dentro del sistema de archivos del host (generalmente en `/var/lib/docker/volumes`).
- Docker asegura la persistencia de los datos, incluso si el contenedor se detiene o elimina.

- **Ventaja principal:** Los datos persisten independientemente del ciclo de vida del contenedor.
- **Uso común:** Almacenar bases de datos y otros datos que deben mantenerse a través de múltiples ciclos de vida de contenedores.

```
docker volume create volumen_nombre
docker run -d -v volumen_nombre:/data/db mongo
```

Montajes de directorios del host (Bind Mounts)

- Los **bind mounts** permiten usar directorios del host directamente dentro del contenedor.
- El contenedor puede **leer y escribir** en estos directorios, permitiendo acceso directo a los archivos del host.

- **Ventajas:**
  - Permite compartir archivos entre el host y el contenedor, útil para logs, configuraciones, bases de datos, etc.
  - Cambios realizados en el host se reflejan inmediatamente en el contenedor y viceversa.

- **Desventajas:**
  - Riesgo de corrupción de datos si el contenedor y el host no están bien sincronizados.
  - Dependencia de la estructura del sistema de archivos del host.

- **Uso común:** En entornos de desarrollo, donde se necesitan cambios inmediatos entre el código del host y el contenedor.

```
docker run -d -v ${PWD}/index.html:/usr/share/nginx/html/index.html nginx
```

Volúmenes temporales (Anonymous Volumes)

- Volúmenes creados automáticamente por Docker sin un nombre explícito.
- Estos volúmenes no tienen un nombre y **persisten** después de que el contenedor se elimina, aunque no se pueden acceder fácilmente sin un nombre.

- **Ventajas:** Útiles para almacenar datos temporales generados por un contenedor, como archivos de logs o datos temporales.
- **Desventajas:** No se pueden gestionar fácilmente y pueden acumularse si no se eliminan explícitamente.

- **Uso común:** Datos que solo deben existir durante la vida del contenedor, como archivos temporales generados durante su ejecución.

```
docker run -d -v /data/db mongo
```