

¿Qué son los Volúmenes en Docker?

Los volúmenes son una característica esencial de Docker que desempeña un papel crítico en la administración de datos en nuestros contenedores.

Imagina un contenedor de Docker como una caja autónoma que contiene una aplicación y todo lo que necesita para funcionar correctamente. Esto incluye el código de la aplicación, sus dependencias y su entorno. Pero, aquí está el truco: cuando destruimos o detenemos un contenedor, todos los datos almacenados en ese contenedor se pierden. ¿Y si necesitamos que nuestros datos persistan más allá de la vida útil de ese contenedor? Bueno, aquí es donde los volúmenes entran en escena.

Ventajas de los Volúmenes

Ventaja 1: Persistencia de Datos

Los volúmenes en Docker nos permiten lograr la persistencia de datos. Esto significa que, incluso si destruimos o detenemos un contenedor, los datos que se almacenaron en ese contenedor no se pierden.

Ventaja 2: Aislamiento de Datos

Segundo, los volúmenes proporcionan un aislamiento efectivo de datos. Cada contenedor puede tener su propio espacio de almacenamiento aislado, lo que significa que los datos de un contenedor no se mezclan ni interfieren con los datos de otro contenedor en la misma máquina host. Esto es crucial para evitar conflictos y garantizar la seguridad de los datos.

Ventaja 3: Escalabilidad

Visualización Gráfica: (Muestra varios contenedores compartiendo datos a través de volúmenes)

Los volúmenes permiten la escalabilidad de nuestras aplicaciones. Con volúmenes, podemos compartir datos entre múltiples contenedores de manera eficiente. Esto es especialmente útil cuando estamos diseñando aplicaciones distribuidas o microservicios, ya que garantiza que todos los contenedores tengan acceso a los mismos datos actualizados.

Ejemplo práctico 1

Crear contenedor sin volumen

Eliminar contenedor

Iniciar otro contenedor y comprobar que no hay datos.

```
docker volume create python_data
docker volume ls
docker run -it -v python_data:/data python:3.10 bash
# touch test.txt
docker run -it -v python_data:/data python:3.11 bash
#ls /data
```

Ejemplo práctico 2

Utilizar volúmenes para almacenar archivos de configuración compartidos entre múltiples contenedores

```
docker volume create nginx_8080_conf
docker run -v nginx_8080_conf:/etc/nginx/conf.d nginx
docker run -it -v nginx_8080_conf:/data alpine sh
#ls /data
#vim default (poner puerto 8080)
docker run -v nginx_8080_conf:/etc/nginx/conf.d nginx
```

Ejemplo practico 3

Para mysql generar un dockerfile con un my.cnf personalizado.

Despues poner los logs dentro de un volumen

Tambien generar un volumen para los datos

Ejemplo practico 4

Lo mismo que mysql pero con MongoDB

Curiosidades y pro tips de volúmenes

docker volume prune – Eliminación de Volúmenes no Utilizados

docker run -v nombre-del-volumen:/ruta-en-el-contenedor:ro imagen-del-contenedor
– volumen de solo lectura

docker run -d --tmpfs /ruta-en-el-contenedor imagen-del-contenedor –
volumetemporal