

Cuarto laboratorio

1. Trabajar con volúmenes en Docker

Un volumen en Docker es un directorio especial asignado a un contenedor, diseñado para persistir datos. Los volúmenes se almacenan en el host dentro del directorio `/var/lib/docker/volumes`. Estos se utilizan para garantizar que los datos persistan independientemente de la vida útil del contenedor.

```
docker volume
```

Uso:

```
Usage:  docker volume COMMAND
```

```
Manage volumes
```

```
Commands:
```

<code>create</code>	Create a volume
<code>inspect</code>	Display detailed information on one or more volumes
<code>ls</code>	List volumes
<code>prune</code>	Remove all unused local volumes
<code>rm</code>	Remove one or more volumes

Ejemplo de listar volúmenes existentes:

```
docker volume ls
```

Salida:

DRIVER	VOLUME NAME
local	0bf977f8de99b90fa1f2d7250415700cad4a6aaeff80c1e0c7bbd934aecaac19

Crear y usar volúmenes al iniciar contenedores

Opcion `--mount`:

```
docker run -d \  
  --name devtest \  
  --mount source=myvol2,target=/app \  
  nginx:latest
```

Opcion -v:

```
docker run -d \  
  --name devtest \  
  -v myvol2:/app \  
  nginx:latest
```

2. Laboratorio: Redes Docker y persistencia con MySQL y WordPress

Este laboratorio conecta un contenedor de WordPress con un contenedor de MySQL mientras se asegura la persistencia de los datos de la base de datos.

Parar y eliminar contenedores existentes

```
docker stop $(docker ps -a -q)  
docker rm -f $(docker ps -a -q)
```

Crear directorio en el host para persistir los datos de MySQL

```
mkdir /bd
```

Lanzar contenedores**1. Contenedor MySQL:**

```
docker run -dti --name servidor_mysql -e MYSQL_ROOT_PASSWORD=000000 -v  
/bd:/var/lib/mysql mysql:5.7
```

2. Contenedor WordPress:

```
docker run -dti --name servidor_wp -p 8000:80 --link servidor_mysql:mysql  
wordpress:5.6.2-php7.3
```

Acceso a WordPress: <http://192.168.33.10:8000/>

Verificar persistencia tras reinicio

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)

# Reiniciar contenedores

docker run -dti --name servidor_mysql -e MYSQL_ROOT_PASSWORD=000000 -v
/bd:/var/lib/mysql mysql:5.7
docker run -dti --name servidor_wp -p 8000:80 --link servidor_mysql:mysql
wordpress:5.6.2-php7.3
```

3. Formas de trabajar con volúmenes

Primera forma: Usar volúmenes predeterminados

Ejemplo:

```
docker run -dtiP --name centos6-lamp -v /var/www/html docker.io/nickistre/centos-
lamp
docker inspect centos6-lamp
```

Salida (fragmento relevante):

```
"Mounts": [
  {
    "Type": "volume",
    "Name": "f4204ec823b82eae69eb3488cfb143a6215fecfe15ed4ee2c524d80697631288"
  }
]
```

Verificar contenido del volumen:

```
docker volume inspect
f4204ec823b82eae69eb3488cfb143a6215fecfe15ed4ee2c524d80697631288
```

Segunda forma: Montar un directorio del host

1. Crear y preparar el directorio:

```
mkdir /web
cd /web
echo "prueba de volumen" > index.html
```

2. Lanzar contenedores:

```
# Volumen con permisos de escritura

docker run -dtiP --name centos6-prueba-web -v /web:/var/www/html
docker.io/nickistre/centos-lamp

# Volumen solo lectura

docker run -dtiP --name centos6-prueba-web2 -v /web:/var/www/html:ro
docker.io/nickistre/centos-lamp
```

3. Verificar:

```
docker ps -a
```

Tercera forma: Crear volúmenes nombrados

1. Crear un volumen:

```
docker volume create --name webapp
```

2. Inspeccionar volumen:

```
docker volume inspect webapp
```

Salida (fragmento relevante):

```
"Mountpoint": "/var/lib/docker/volumes/webapp/_data",
```

3. Crear un archivo en el volumen:

```
echo "prueba de volumen" > /var/lib/docker/volumes/webapp/_data/index.html
```

4. Lanzar contenedores:

```
mkdir /web

docker run -dtiP --name centos6-pruebacreacion3 -v webapp:/var/www/html -v /web:/datos docker.io/nickistre/centos-lamp
```

5. Verificar configuración del contenedor:

```
docker inspect centos6-pruebacreacion3 |grep -A 5 -i mounts
```

6. Acceder al contenedor y listar contenido:

```
docker exec -ti centos6-pruebacreacion3 /bin/bash
ls /datos
```

Eliminar contenedores en estado "Exited"

```
docker rm $(docker ps -a -f status=exited -q)
```

Uso de volúmenes con la opción --mount

Iniciar un contenedor con un volumen:

```
docker run -d \
  --name devtest \
  --mount source=myvol2,target=/app \
  nginx:latest
```

Verificar:

```
docker volume ls
docker volume inspect myvol2
```

Salida (fragmento):

```
"Mountpoint": "/var/lib/docker/volumes/myvol2/_data",
```

Dentro del contenedor:

```
docker exec -ti devtest bash
root@d74215cfb8ba:/# df -h
```

Usar un volumen solo lectura:

Opcion `--mount`:

```
docker run -d \
  --name=nginxtest \
  --mount source=nginx-vol,destination=/usr/share/nginx/html,readonly \
  nginx:latest
```

Opcion `-v`:

```
docker run -d \
  --name=nginxtest \
  -v nginx-vol:/usr/share/nginx/html:ro \
  nginx:latest
```