

Práctica 1. Bind Mount

Un *bind mount* vincula una carpeta del host con una del contenedor.

Ejemplo 1: Compartir una carpeta del host con un contenedor Ubuntu

Objetivo:

Tener una carpeta local `C:\docker_pruebas` (o `/home/usuario/docker_pruebas` en Linux) que se vea dentro del contenedor.

Comando:

```
docker run -it --name contenedor1 -v "C:\docker_pruebas:/data" ubuntu bash
```

Explicación:

- `-it` → modo interactivo con terminal.
- `--name contenedor1` → nombre del contenedor.
- `-v "C:\docker_pruebas:/data"` → monta tu carpeta local en `/data` dentro del contenedor.
- `ubuntu bash` → abre una consola Bash en Ubuntu.

Dentro del contenedor

```
echo "Hola desde Docker" > /data/hola.txt  
cd /data  
ls  
cat hola.txt  
exit
```

A continuación modifica el archivo desde el host y comprueba que se ven los cambios en docker

Ejemplo 2

Objetivo

1. Montar un archivo `nginx.conf` del host dentro del contenedor Nginx.
2. Arrancar Nginx usando esa configuración personalizada.
3. Verificar que los cambios en el host afectan inmediatamente al contenedor.

1. Crear carpeta y archivo de configuración en Windows

Crea un archivo `nginx.conf` básico:

```
@  
events {}  
  
http {  
    server {  
        listen 8080;  
        location / {  
            return 200 'Hola desde Nginx en Docker!';  
        }  
    }  
}  
"@ > C:\docker_nginx_test\nginx.conf
```

Esto crea un servidor Nginx que responde en el puerto 8080 con un mensaje simple.

2. Ejecutar contenedor Nginx con Bind Mount

```
docker run -d --name mi_nginx `   
-v C:\docker_nginx_test\nginx.conf:/etc/nginx/nginx.conf:ro `   
-p 8080:8080 `   
nginx:latest
```

Explicación

Parte	Significado
<code>-d</code>	Contenedor en segundo plano (detached)
<code>--name mi_nginx</code>	Nombre legible para el contenedor
<code>-v C:\docker_nginx_test\nginx.conf:/etc/nginx/nginx.conf:ro</code>	Monta el archivo de configuración desde Windows dentro del contenedor en modo solo lectura (<code>:ro</code>)
<code>-p 8080:8080</code>	Publica el puerto 8080 del contenedor en tu máquina
<code>nginx:latest</code>	Imagen oficial de Nginx

Nota: Es importante usar `:ro` para que el contenedor no pueda sobrescribir tu archivo host.

3. Verificar que Nginx funciona

En tu navegador o usando PowerShell:

```
curl http://localhost:8080
```

Deberías ver:

```
Hola desde Nginx en Docker!
```

Esto confirma que **Nginx está usando el `nginx.conf` que montaste desde Windows.**

4. Probar cambios en tiempo real

Edita el archivo `nginx.conf` en Windows, por ejemplo:

```
notepad C:\docker_nginx_test\nginx.conf
```

Cambia el mensaje:

```
return 200 'Mensaje actualizado desde Windows!';
```

Luego recarga Nginx dentro del contenedor:

```
docker exec mi_nginx nginx -s reload
```

Ahora, si haces otra solicitud:

```
curl http://localhost:8080
```

Verás el mensaje actualizado:

```
Mensaje actualizado desde Windows!
```
