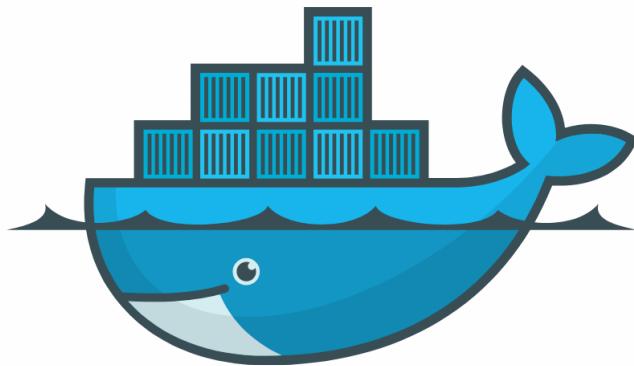




CONTENEDORES: CREACIÓN IMAGEN

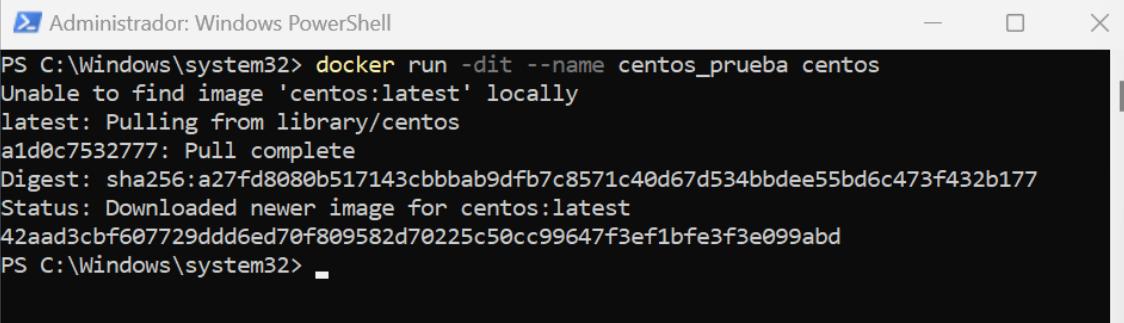


Ivana Sánchez Pérez

Parte 1: Comprobar que el archivo se pierde al eliminar el contenedor

1- Iniciar un contenedor CentOS

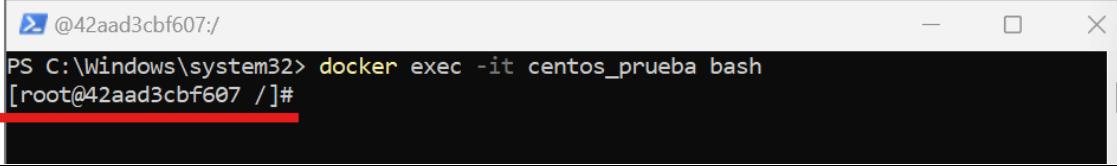
Ejecutamos un contenedor basado en CentOS con el comando **docker run -dit --name centos_prueba centos**



```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker run -dit --name centos_prueba centos
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
42aad3cbf607729ddd6ed70f809582d70225c50cc99647f3ef1bfe3f3e099abd
PS C:\Windows\system32>
```

2- Acceder al contenedor

Con el comando docker **exec -it centos_prueba bash**



```
@42aad3cbf607:/
PS C:\Windows\system32> docker exec -it centos_prueba bash
[root@42aad3cbf607 /]#
```

3- Crear el archivo dentro del contenedor

Dentro del contenedor, creamos el directorio y el archivo en ***/opt/actividades/ejercicio.txt***. Para ello utilizaremos los siguientes comandos:

mkdir -p /opt/actividades → para crear el directorio actividades

echo "Esto es un archivo de prueba para ver si persiste una imagen después de borrarla" > /opt/actividades/ejercicio.txt → para crear el archivo de texto

Finalmente haremos un cat para comprobar que se ha creado todo correctamente.

```
[root@42aad3cbf607 /]#  
[root@42aad3cbf607 /]# mkdir -p /opt/actividades  
[root@42aad3cbf607 /]# echo "Esto es un archivo de prueba para ver si persiste una imagen después de borrarla" > /opt/actividades/ejercicio.txt  
[root@42aad3cbf607 /]# cat /opt/actividades/ejercicio.txt  
Esto es un archivo de prueba para ver si persiste una imagen después de borrarla  
[root@42aad3cbf607 /]#
```

4- Salir y eliminar el contenedor

Salimos con exit, paramos con stop y eliminamos con rm. Es decir,

- **Exit**
- **docker stop centos_prueba**
- **docker rm centos_prueba**

The screenshot shows a Windows PowerShell window titled 'Administrador: Windows PowerShell'. The command history is as follows:

```
[root@42aad3cbf607 /]# exit  
exit  
PS C:\Windows\system32> docker stop centos_prueba  
centos_prueba  
PS C:\Windows\system32> docker rm centos_prueba  
Error response from daemon: No such container: centos  
Error response from daemon: No such container: prueba  
PS C:\Windows\system32> docker rm centos_prueba  
centos_prueba  
PS C:\Windows\system32>
```

5- Iniciar un nuevo contenedor desde la imagen base

Volvemos a ejecutar el comando **docker run -dit --name centos_prueba centos**

The screenshot shows a Windows PowerShell window titled 'Administrador: Windows PowerShell'. The command history is as follows:

```
PS C:\Windows\system32> docker run -dit --name centos_prueba centos  
a632685d125990dd319bd340f1edf9c1c5efdc109b1959f2478e8d8c5d01b531  
PS C:\Windows\system32>
```

Verificamos si el directorio sigue existiendo con:

Docker exec -it centos_prueba bash y luego un **ls /opt/**

Observaremos que el directorio no está, ya que los contenedores son efímeros.

```
PS C:\Windows\system32> docker exec -it centos_prueba bash
[root@a632685d1259 /]# ls /opt/
[root@a632685d1259 /]#
```

Parte 2: Persistir el archivo mediante una imagen

1- Iniciar un nuevo contenedor y crear el archivo

Hacemos lo mismo que antes para crear un nuevo contenedor:

docker run -dit --name centos_persistente centos

docker exec -it centos_persistente bash

```
PS C:\Windows\system32> docker run -dit --name centos_persistente centos
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbb9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
0561b5e4a757696131b15377adb9790892b1b81dd08d8e9af355bf0d6fcfb511
PS C:\Windows\system32> docker exec -it centos_persistente bash
[root@0561b5e4a757 /]#
```

Creamos dentro del contenedor de nuevo el directorio actividades y el archivo ejercicio.txt con su texto, tal y como hicimos antes.

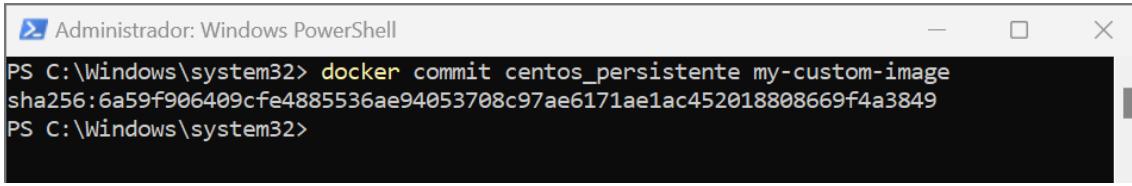
```
[root@0561b5e4a757 /]#
[root@0561b5e4a757 /]#
[root@0561b5e4a757 /]#
[root@0561b5e4a757 /]# mkdir -p /opt/actividades
[root@0561b5e4a757 /]# echo "Esto es un archivo de prueba para confirmar que esta vez persiste una imagen después de borrarla" > /opt/actividades/ejercicio.txt
[root@0561b5e4a757 /]# cat /opt/actividades/ejercicio.txt
Esto es un archivo de prueba para confirmar que esta vez persiste una imagen después de borrarla
[root@0561b5e4a757 /]#
```

2- Crear una imagen a partir del contenedor

Ejecutamos el siguiente comando para guardar el estado del contenedor en una imagen llamada my-custom-image:

- Docker commit → crea una imagen a partir de un contenedor
- Centos-persistente → nombre del contenedor a capturar.
- My-custom-image → nombre de la nueva imagen

Docker commit centos_persistente my-custom-image



```
PS C:\Windows\system32> docker commit centos_persistente my-custom-image
sha256:6a59f906409cfe4885536ae94053708c97ae6171ae1ac452018808669f4a3849
PS C:\Windows\system32>
```

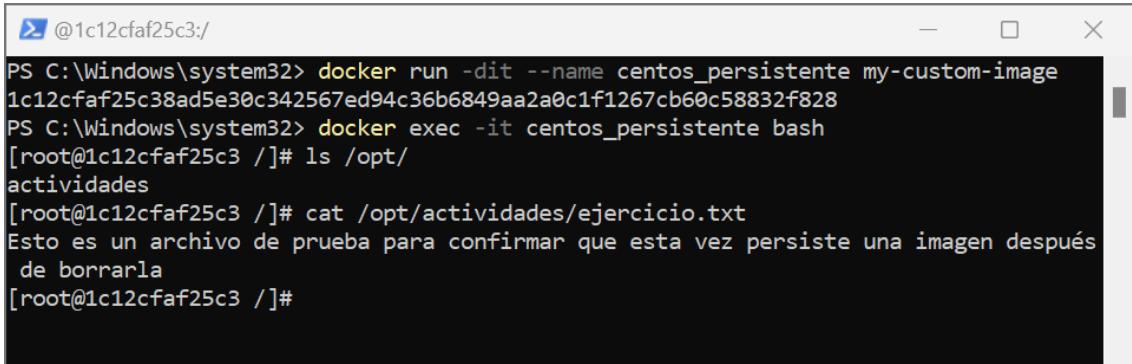
3- Iniciar un nuevo contenedor desde la imagen personalizada

Con el comando **docker run -dit --name centos_persistente my-custom-image** iniciamos el nuevo contenedor

4- Verificar la persistencia del directorio y archivo

Ahora, comprobamos tanto el directorio como el archivo .txt siguen existiendo o no con los comandos:

- **Docker exec -it centos_persistente ls /opt/actividades**
- **Docker exec -it centos_persistente bash**
- **ls /opt/**
- **cat /opt/actividades/ejercicio.txt**



```
PS C:\Windows\system32> docker run -dit --name centos_persistente my-custom-image
1c12cfaf25c38ad5e30c342567ed94c36b6849aa2a0c1f1267cb60c58832f828
PS C:\Windows\system32> docker exec -it centos_persistente bash
[root@1c12cfaf25c3 /]# ls /opt/
actividades
[root@1c12cfaf25c3 /]# cat /opt/actividades/ejercicio.txt
Esto es un archivo de prueba para confirmar que esta vez persiste una imagen después
de borrarla
[root@1c12cfaf25c3 /]#
```

Comprobamos que ambos siguen existiendo, debido a que lo hemos incluido en la imagen.

