

Pistas para entender, y lidiar con, la persistencia en Docker for Windows

Artículo

[Jose Tomas Salvador](#) · Nov 27, 2019



Lectura de 5 min

Pistas para entender, y lidiar con, la persistencia en Docker for Windows

¡Hola a todos!

Como seguramente ya sabréis, ya está disponible en Docker Hub una versión totalmente gratuita de [InterSystems IRIS for Health \(I4H\)](#). Se trata de la versión Community y, como digo, es totalmente gratuita y funcional. Ya se ha hablado de ella en algunos artículos y posts,... así que hoy no va de eso. Aquí vamos a ver el "misterioso caso de la desaparición o, mejor, ausencia de mis datos persistentes cuando arranco el contenedor con la opción *durable* " (no he encontrado la fuente de letra terrorífica para darle más suspense... el editor de posts da para lo que da 😊).

Bien, pues si seguimos las instrucciones de instalación que encontramos en Docker Hub para una utilización básica de la imagen, veremos que hay un apartado titulado " *Start an InterSystems IRIS container with persistent instance-specific storage* ", donde se nos indica el comando de Docker que tenemos que ejecutar para que nuestro contenedor mantenga las bases de datos, directorios y archivos relevantes del sistema (y por tanto toda la configuración de seguridad, etc..), en nuestro host, permitiéndonos igualmente tener una ubicación para otras bases de datos que quisiésemos crear y mantener aún cuando decidiéramos destruir el contenedor.

En concreto, el comando es:

```
docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773  
--volume /home/user1:/durable [redacted]  
--env ISC_DATA_DIRECTORY=/durable/iris store/intersystems/iris-community:2019.4.0.379.0
```

Si ejecutamos este comando en nuestra consola de *Powershell* , veremos que el contenedor arranca sin problemas. Podemos entrar en el portal desde nuestro navegador (utilizando el puerto mapeado) con <http://localhost:9092/csp/sys/UtilHome.csp> y nos dará acceso a nuestro I4H tras indicar una nueva password (por defecto la password es SYS).

Pero... ¿dónde están mis datos persistentes?... podéis pensar: " *Ah... claro! Le he puesto /home/user1... claro, como esta gente sólo piensa en Linux o Mac y no se acuerda de que el 80% de los sistemas están en Windows puesss... (perdón a los maClinuxeros ... dejad que nos quejemos un poco los que todavía nos gusta Windows 😊*

) ... Claro, lo que habrá pasado es que ha dado error y lo ha arrancado por defecto sin mantener un volumen durable ".

¡Error! Se ha arrancado bien, lo ha hecho bien, y tenéis un volumen " *durable* ", sólo que no está donde puede que creyeráis.

El caso es que Docker for Windows se ejecuta realmente junto con una máquina virtual **mobyLinux** en **Hyper-V** . Y es ahí, en el host real (aunque sea virtual..vaya lío, eh?), y no en Windows, dónde realmente tendrás que buscar tu directorio: */home/user1* ; al que está mapeado el directorio */durable* del contenedor. En */home/user1* tendrás la copia de tus ficheros de sistema y bases de datos que quieras mantener incluso cuando el contenedor se destruye.

Y diréis: "Ya... pero vaya faena... no tengo forma de 'jugar' con esos datos porque no puedo entrar en la máquina virtual que trae Docker"... bueno, sí y no. Igual no puedes entrar con una shell directa, pero sí que podemos acceder al contenido del disco que utiliza la máquina *mobyLinux*. Para ello, cuando tengas arrancado tu contenedor I4H, abre otra consola PowerShell y ejecuta lo siguiente:

```
docker run --net=host --ipc=host --uts=host --pid=host -it \
--security-opt=seccomp=unconfined --privileged --rm -v /:/host alpine /bin/sh
```

Con esto descargaréis una imagen de Linux Alpine (si no lo tenéis ya), arrancaréis un contenedor ¡ligado a tu máquina *mobyLinux*! y entraréis en la shell de Alpine. Una vez allí, al ejecutar un simple `ls -l`, veréis que hay un directorio `/host`, y ¡¡et voilà!! , dentro de él habrá un `./home/user1` donde tendréis todo vuestro contenido que queréis mantener.

Por supuesto existe la posibilidad de ubicar el `/durable` en un directorio de vuestro Windows. Si ejecutamos lo mismo pero cambiando el path (para que no haya conflicto de nombre y puertos debemos parar antes el otro contenedor con `docker stop my-iris`):

```
docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773 \
--volume c:/Temp/Data:/durable \
--env ISC_DATA_DIRECTORY=/durable/iris store/intersystems/iris-community:2019.4.0.379.0
```

Comprobaremos que, si existe en nuestro Windows el directorio `c:/Temp/Data`, se habrán copiado archivos y BDs de sistema de I4H en ese directorio y, por tanto, serán accesibles desde Windows. ¡Precaución! El problema que nos podemos encontrar aquí es con los permisos de los directorios y ficheros. Es un problema común cuando se habla de persistencia y contenedores Linux en Windows, no sólo de I4H, y viene derivado del hecho de que la tecnología de contenedores viene básicamente del mundo Linux, de ahí que se necesite la máquina virtual *mobyLinux*, y de que no hay una equivalencia exacta entre el sistema de permisos de Windows y el de Linux...

De hecho, si volvéis a entrar en la shell de Alpine, veréis que también tenéis acceso a vuestro `host` Windows desde *mobyLinux*. Ejecutad lo siguiente: `cd /host/host_mnt/c/Temp/Data/iris` y estaréis, desde Alpine->*MobyLinux*, viendo el contenido de `c:/Temp/Data/iris`. Si ahí ejecutáis un `ls -l`, veréis que todos los directorios pertenecen a **root** y están en el grupo **root**. Si ahora ejecutáis en la shell de Alpine: `cd /host/home/user1/iris` y después `ls -l`, veréis que aparecen usuarios y grupos (no definidos) con **uid 52773** (`irisuser`) y **gid 51773** (`irisowner`) respectivamente. Este tema de los permisos puede darnos conflictos cuando usamos imágenes con SO base Linux con Docker for Windows... ya digo, no es exclusivo de I4H y tiene que ver con la no equivalencia entre usuarios/permisos en Linux y en Windows. Conviene tenerlo en cuenta para evitar quebraderos de cabeza.

IMPORTANTE: Docker resetea el directorio `/home` de la máquina virtual *mobyLinux* cada vez que se reinicia, por lo que si rearrancamos el ordenador o simplemente paramos e iniciamos Docker, veremos que todo el contenido de nuestro `/home/user1` ha desaparecido. Para preservarlo, puedes almacenarlo en el mismo sitio en que él almacena los volúmenes: `/var/lib/docker/volumes`; que, si entramos con nuestra "shell mágica", estaría en `/host/var/lib/docker/volumes`.

Así, podríamos arrancar el contenedor de este modo:

```
docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773 --volume \
/var/lib/docker/volumes/user1:/durable --env ISC_DATA_DIRECTORY=/durable/iris store/intersystems/iris \
community:2019.4.0.379.0
```

y así nos aseguraríamos de que nuestra información persistente sigue ahí, no sólo aunque destruyamos nuestro contenedor, sino incluso tras reiniciar Docker for Windows (ojo, si reinstalamos Docker ya sí que perderíamos todo).

Bueno, espero que os haya servido.

¡¡Disfrutad!!

Pistas para entender, y lidiar con, la persistencia en Docker for Windows

Published on InterSystems Developer Community (<https://community.intersystems.com>)

[#Consejos y trucos](#) [#Contenedorización](#) [#Docker](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

10 1 0 0 133

Log in or sign up to continue
Añade la respuesta

URL de fuente: <https://es.community.intersystems.com/post/pistas-para-entender-y-lidiar-con-la-persistencia-en-docker-windows>