Article
[Jose-Tomas Salvador](#) · Nov 27, 2019 5m read

Hints to understand, and deal with, persistence in Docker for Windows

¡Hi everybody!

As you likely are aware, the new version of [InterSystems IRIS for Health](#) (I4H) it's already available in Docker Hub. It's the Community version and is free and fully functional. There have been comments about it in other articles and posts,... so today I won't add anything about features. Here I want to explore *"the mistery about the disappearance, or better, absence of our persistent data when we run a container with the durable option"* (I didn't find a terrifying font to emphasize the thriller... post editor is not terrific for styling 🙂).

Well, so if we follow setup instructions that we find in Docker Hub for a basic use of an image, we'll see a section named "*Start an InterSystems IRIS container with persistent instance-specific storage*", where you can see the docker command to execute so that our container keeps databases, folders and files relevant for the system (and so security settings, etc...), in our host, allowing us also to have a location for other new databases that we would like to create and maintain even in the cases where we decide to destroy the container.

Specifically, the command is:

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

If we execute this command in our *Powershell* console, we'll see that the container starts without problems. We can then login into the portal using our browser (using the mapped port) with *http://localhost:9092/csp/sys/UtilHome.csp* and that will give us access to our I4H after setting a new password (by default the password is SYS).

But.. ¿ where is my persistent data?... you could think: *"Ah... of course! They used /home/user1... I see, as these people only thinks about Linux or Mac these days and don't remember that 80% of systems run on Windows theeennn.... "* (sorry to "maCLinuxers"... let the people that still appreciate Windows complain a bit 🙂) ...let's go on with internal thinking... " *Of course, this is what has happened... there has been a hidden error and it's started by default without creating a volume durable*".

¡Error! It started right, it did it right, and you have a volume "durable", only perhaps it's not where you would expect.

The point is that Docker for Windows really runs on a **mobyLinux** virtual machine in **Hyper-V**. And it's there, in the real host (even it's virtual... what a mess, eh?), and not in windows, where you will actually have to find your physical folder: */home/user1*; to which */durable* is mapped from within the container. In */home/user1* you'll have all your system files and all the databases you want to keep in the case in which the container is destroyed.

And you'd say: "*Sure... what a bummer!... I have no way to 'play' with those data because I can't log into the docker virtual machine*"... well, yes and no. Perhaps you cannot run a direct shell to that VM, but we can access to the virtual hard disk content that mobyLinux is using. To do that open a PowerShell console and execute:

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

With this you would download a Linux Alpine image (if you don't have one yet), run a container (linked to your mobyLinux virtual machine!) and will log into the Alpine's shell. Once there, if you execute a basic ls -l , you'll see that there is a folder *host*, ¡¡et voilá!!, within it there'll be a *../home/user1* folder with all the persistent content you wanted to maintain.

Of course, you can also locate */durable* in a folder in your Windows system. If we run the same command but changing the path (to avoid name and port conflicts, first you should stop the previous container with docker stop my-iris):

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

After that we'll realize that, if folder c:/Temp/Data already existed in Windows, there we'll have system DBs and files of I4H and, then, they'll be accesible from Windows explorer. ¡Caution! We could bump into a problem with folder and files' permissions. It's a wide common problem when we talk about *persistence vs linux containers in Windows*, it's not related to I4H, and derives from the fact that container technology comes from Linux world, that's why a mobyLinux VM is required, and also from the fact that there is no straightforward match between Windows and Linux regarding users, groups, permissions...

In fact, if you log in again with the shell of your Alpine container, you'll see that you also have access to your Windows host from mobyLinux. Execute the following: cd /host/hostmnt/c/Temp/Data/iris and you would be, from *Alpine->MobyLinux*, looking at the content of *c:/Temp/Data/iris*. Once there, if you execute ls -l, you'll see that all the folders belong to **root** user and **root** group. If then you execute: cd /host/home/user1/iris and then ls -l, you'll see that there are users and groups (undefined) with **uid 52773** (irisuser) and **gid 51773** (irisowner) respectively. These issue about users, groups and permissions, can lead to conflicts when we use images where the base OS is Linux in a Docker for Windows environment... as I said, it's not releated to I4H itself, it has to be with mismatch between user/group/permissiins between Linux and Windows. Better take it into account to avoid headaches.
IMPORTANT: Docker resets the /home folder in its mobyLinux virtual machine each time that it restarts, so everytime we restart Docker or our Windows desktop, we'll find out that all the content in /home/user1, even the folder itself, has disappeared. To preserve your persistent content, you can store it in the exact same place that Docker uses to store the volumes: /var/lib/docker/volumes; which, if we access with our "magic" Alpine shell, it would be: /host/var/lib/docker/volumes.
So, we could run the container in this way:

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

then we would make sure that our persistent information it'll be safely there, not only if we decide to destroy the container, but also after restarting Docker for Windows (be aware, you would lost everything if you reinstall Docker)

Well, I hope you find this useful. *Enjoy!!*

#Containerization #Docker #Tips & Tricks #InterSystems IRIS #InterSystems IRIS for Health

Source URL:https://community.intersystems.com/post/hints-understand-and-deal-persistence-docker-windows