Artículo
Jose-Tomas Salvador · 3 feb, 2020 Lectura de 7 min

Docker para Windows y la plataforma de datos InterSystems IRIS

¡Hola Comunidad!

Ahora que la plataforma de datos InterSystems IRIS Data Platform Community Edition está disponible en Docker Hub, parece el momento ideal de probar InterSystems IRIS en un contenedor. Desde hace un tiempo, la *Community Edition* ha estado disponible en la nube, por ejemplo en AWS, pero podría ser interesante probarla localmente también. Los usuarios de Windows estamos acostumbrados a ver caras de desconfianza y escuchar que "nuestros resultados podrían variar" cada vez que mencionamos usar Docker para Windows. A veces incluso nos dicen que realmente deberíamos considerar ejecutar Docker dentro de máquinas virtuales con Ubuntu. Pfff...

Docker y Windows, especialmente al usar contenedores basados en Linux (el único tipo soportado por InterSystems IRIS) puede que no sean una combinación ideal. Para más detalles sobre esto, vea: Linux Containers on Windows. Sin embargo, con algo de paciencia, ganas de aventura y unos pasos adicionales, es perfectamente posible ejecutar InterSystems IRIS en un contenedor Docker en Windows. Esta configuración, aunque no está soportada para casos de producción, funciona suficientemente bien como para probar varias funcionalidades de InterSystems IRIS. Por ejemplo, he creado namespaces y bases de datos, he cargado datos, creado producciones y hasta ejecutado algunas consultas SPARK.

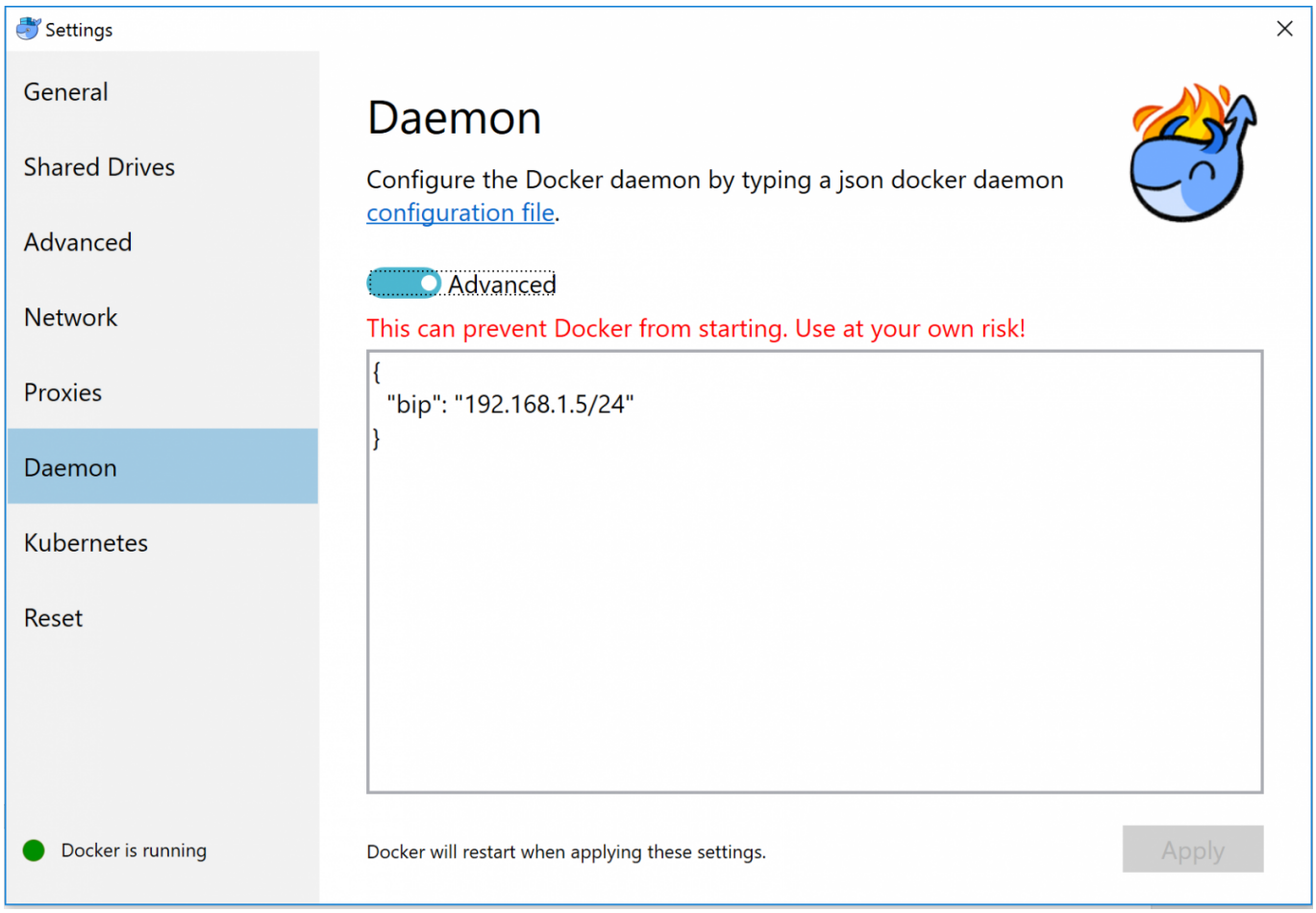
A continuación, enumeraré los pasos que seguí para lanzar la plataforma de datos **InterSystems IRIS Community Edition**. Mostraré tanto un enfoque por línea de comandos de Docker como el enfoque más práctico de componer un archivo. *Advertencia: no soy un experto en Docker y estaremos hablando de Docker para Windows, así que tengan en cuenta que "nuestros resultados podrían variar" ;-)*

Configuración

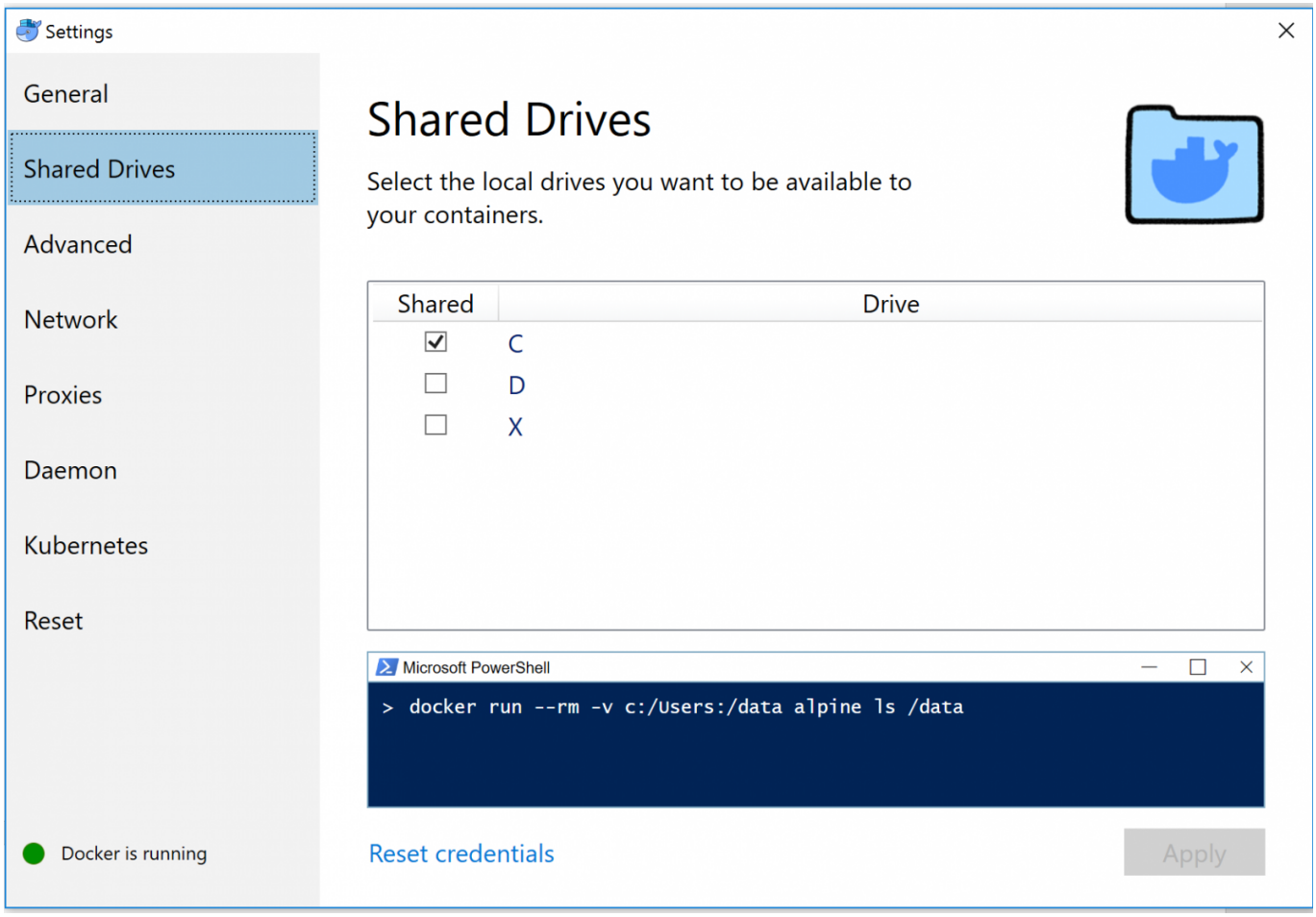
1. Windows 10 Pro
2. Docker Desktop Community Edition, versión 0.0.3 (31259) Canal: estable, disponible aquí

Algunos ajustes de Docker

1. Cambié el " bip " en mis ajustes de Daemon para evitar los conflictos de red



2. Configuré mi unidad C:\como unidad compartida. Esto permitirá que el contenedor acceda a los archivos de C:\



Apunte sobre *Name Volumes* y Montajes Bind

Los contenedores son efímeros. Los iniciamos. Los detenemos. Los eliminamos. Sin embargo, a menudo quisiéramos que los datos generados y usados por el contenedor persistan. Docker ofrece dos enfoques para la persistencia de datos de un contenedor: volúmenes y montajes bind. Los contenedores de InterSystems IRIS incluyen la funcionalidad *%SYS perdurable* (*Durable %SYS*) para datos generados y usados por la plataforma de datos. En Docker para Windows, el *%SYS perdurable* debe usar un volumen Docker. Al usar Docker en otros sistemas operativos, por ejemplo Docker para Mac o Docker para Ubuntu, un %SYS perdurable puede usar tanto un montaje bind como un volumen.

Los volúmenes son totalmente gestionados por Docker. A diferencia de los montajes bind, no se corresponden directamente con cualquier archivo o directorio en el sistema de archivos del host. Debemos usar comandos de Docker para crear, eliminar e inspeccionarlos. El hecho de que no podamos ajustar o usar productos y herramientas avanzadas para gestionar los datos en volúmenes Docker es una gran desventaja de usar Docker en Windows. Claramente no es un entorno de producción.

Creación de un N*amed Volume*

Mi próximo paso fue crear un named volume desde la línea de comandos de Windows:

```
docker volume create durable
```

Para ver los detalles de este volumen, use el comando docker volume inspect. Cuando quiera eliminarlo, usaré el comando docker volume rm.

Puesta en funcionamiento del contenedor

Finalmente, estaba listo para lanzar el contenedor. Aquí está el comando de ejecución:

docker run –it –d –p 51773:51773 –p 52773:52773 –-name iris –-volume durable:/durable –-env ISCDATADIRECTORY=/durable/irissys store/intersystems/iris:2019.1.0.510.0-community

Algo más de información sobre las opciones que pasé al comando run:

Opción	**Descripción**
it	Básicamente, significa ejecutar en modo interactivo y brindar acceso por terminal
d	Significa ejecutar en modo *detached*, separado del proceso principal de la consola Windows
p	Mapea los puertos dentro del contenedor a puertos exteriores del contenedor en el host. Usaremos estos puertos desde Windows para acceder a InterSystems IRIS. Tenga en cuenta que si tiene IRIS instalado localmente, es muy probable que esté usando los puertos 51773 y 52773 y obtendrá conflictos de puertos si intenta mapear estos puertos
volume	Mapea el *named volume* creado antes, a un directorio dentro del contenedor
env	Configura las variables de entorno. En este caso, estamos configurando la ubicación del *%SYS perdurable* para que sea nuestro directorio duradero dentro del contenedor.

Para verificar que todo estuviera bien con el contenedor, usé el siguiente comando:

docker container ls

Esto listó todos los contenedores en ejecución en mi sistema. La primera vez que ejecuté el comando, listó el estado de mi contenedor iris como "starting" (iniciando). Cuando volví a ejecutarlo luego de pasado algún tiempo, listó el estado del contenedor como "healthy" (saludable).

Acceso a InterSystems IRIS

Una vez que mi contenedor se estuvo ejecutando sin problema, el siguiente paso fue acceder a la plataforma de datos. Para lograrlo, primero lancé el Portal de gestión usando la URL usual: http://localhost:52773/csp/sys/UtilHome.csp

La página me solicitó un usuario/contraseña. Indiqué SuperUser/SYS y luego me llevó a una página de cambio de contraseña. Después tuve acceso completo al Portal de gestión.

A continuación probé usar el acceso por línea de comandos. Para hacerlo, primero ejecuté una shell *bash* dentro del contenedor con el siguiente comando en la línea de comandos de Windows:

docker exec –it iris bash

Esto abrió una shell bash. Desde ahí, abrí una sesión del terminal IRIS de la siguiente forma:

iris session iris

Inicié sesión usando el usuario/contraseña definido arriba.

Detener el Contenedor

Finalmente, para detener el contenedor, añadí lo siguiente en la línea de comandos de Windows:

docker container stop iris

Pasar a Compose

Usar el comando run de Docker con todas esas opciones a escribir es bastante tedioso. Por suerte, Docker admite colocar las opciones en un archivo *compose* y luego iniciar un contenedor usando comandos de docker-compose para leer las opciones desde el archivo e iniciar el contenedor adecuadamente.

Para usar este método, primero hay que crear un directorio de Windows. Yo creé un directorio llamado iriscommunity. Ten en cuenta que el nombre del directorio se convierte en el nombre del proyecto Docker. Dentro de iriscommunity creé otro directorio llamado local. Mi archivo compose creará un montaje bind usando este directorio y podré usarlo para pasar datos y código entre mi sistema de archivos Windows y mi contenedor. Luego coloqué mi archivo compose, llamado docker-compose.yml dentro de iriscommunity. Por lo tanto, la estructura del directorio se ve así

iriscommunity

local

docker-compose.yml

Estos son los contenidos de mi docker-compose.yml. Espero que los comentarios sean suficientes para explicar lo que hace.

```
version: '3.2'

services:
  iris:
    image: store/intersystems/iris:2019.1.0.510.0-community

    container_name: iris-community

    ports:
    # 51773 is the superserver default port
    - "51773:51773"
    # 52773 is the webserver/management portal port
    - "52773:52773"

    volumes:
    # Maps directory in containr, named durable, to named volume defined below
    - durable:/durable
    # Mounts a local directory for passing in files and test scripts
    - ./local:/Samples/local

    environment:
    # Uses ISC_DATA_DIRECTORY to place the durable %SYS in the named volume
    - ISC_DATA_DIRECTORY=/durable/irissys

    # Creates the named docker volume
volumes:
  durable:
```

Iniciar con docker-compose

Para iniciar el contenedor, abrí una línea de comandos de Windows en mi directorio: *iriscommunity* y añadí lo siguiente:

docker-compose up

Esta vez no definí la opción detach (separar), por lo que veo un flujo de salida desde docker. Tenga en cuenta que el volumen nombrado creado en este caso será iriscommunitydurable, por lo que no entrará en conflicto con el volumen anterior.

Para detener el contenedor, uso el siguiente comando, siempre desde el directorio: *iriscommunity* :

docker-compose down

#Contenedorización #Principiante #InterSystems IRIS

URL de fuente:https://es.community.intersystems.com/post/docker-para-windows-y-la-plataforma-de-datos-intersystems-iris