Artículo

[Ricardo Paiva](#) · 20 ene, 2020 · Lectura de 9 min

Puesta en funcionamiento de IRIS usando Docker

¡Hola Comunidad!

Este breve documento describe los pasos para iniciar IRIS community edition usando Docker en MAC. Para quienes sean nuevos con Docker, primero se explica cómo instalar Docker y ejecutar algunos comandos básicos necesarios para poner en funcionamiento IRIS community edition. Los usuarios con más experiencia en Docker pueden saltarse esa parte.

1. Obtener Docker Desktop para MAC. Éste se puede descargar desde Docker Hub, pero revise primero todos los requerimientos antes de descargarlo para estar seguro de que funcionará correctamente. Puede encontrar las instrucciones y el enlace de descarga aquí: <https://docs.docker.com/docker-for-mac/install/> Las instalaciones de Docker en Windows requieren Docker Toolbox, ya que Docker no puede ejecutarse de forma nativa en Windows. Docker Toolbox crea una máquina virtual sobre la que los usuarios de Windows pueden ejecutar Docker. Los usuarios de Windows deben seguir las instrucciones del sitio web de Docker: <https://hub.docker.com/editions/community/docker-ce-desktop-windows>
2. Pruebe con Docker. El primer comando lógico es determinar qué imágenes de docker se encuentran en el repositorio local. En el terminal de Mac, escriba: `docker images`. En esta etapa no encontrará nada, pero podrá ver los siguientes encabezados: Repository, Tag, Image ID, Created y Size. Ahora, para ejecutar un contenedor de prueba, escriba lo siguiente en la línea de comandos: `docker run busybox:1.24 echo "hello world"` Como la imagen de busybox no se encontró en el repositorio de docker, se "extrajo" (pull) del hub docker, y el comando echo se ejecutó en el contenedor, para imprimir "hola mundo" en la línea de comandos. Ahora pruebe a ejecutar `docker images` otra vez. Ahora debería ver la imagen de busybox en el repositorio local. Al usar Docker, es importante especificar tanto el nombre de la imagen como la etiqueta, separándolas con dos puntos. Esto permite a docker saber qué versión de la imagen ejecutar (p. ej. busybox:1.24). Cuando hay un contenedor local, no se volverá a descargar.
3. Pruebe a ejecutar otro comando en la imagen de Docker. Escriba `docker run busybox:1.24 ls /`. Debería ver un listado de los directorios y archivos en la raíz del contenedor.
4. Al ejecutar un contenedor, hay dos opciones importantes. La marca `-i` inicia un contenedor interactivo, y la marca `-t` crea un pseudo TTY que adjunta stdin y stdout. Escriba esto: `docker run -i -t busybox:1.24`. Ahora podrá ejecutar comandos en el intérprete del contenedor, incluyendo añadir archivos. Si escribe exit, saldrá

del contenedor y lo apagará. Perderá cualquier archivo creado mientras estuviera en el contenedor.

5. Los contenedores normalmente se ejecutan en segundo plano (de forma no interactiva). La marca -d, que significa "detach" (separar) ejecuta el contenedor en modo "separado". Escriba este comando: `docker run -d busybox:1.24 sleep 1000`. Docker devuelve inmediatamente el id del contenedor, y el contenedor se está ejecutando. Para verificarlo, escriba `docker ps`. Esto mostrará qué contenedores se están ejecutando, lo que en nuestro caso es este único contenedor. Para ver los contenedores que se están ejecutando y que se han ejecutado anteriormente, escriba `docker ps -a`. Para eliminar un contenedor cuando sale, escriba `docker run -rm busybox:1.23 sleep 1`. El contenedor se ejecutará pero no aparecerá en el historial que se muestra al usar `docker ps -a`.

6. Docker genera "nombres" extraños de contenedores al ejecutar contenedores. Para especificar el nombre que queremos que Docker genere, debemos usar la opción `--name`. Escriba `docker run --name micontenedor busybox:1.24`. Ahora, al usar `docker ps -a`, podremos ver que el nombre del contenedor es `micontenedor`.

7. Docker ofrece una útil función para obtener detalles sobre información de bajo nivel acerca de contenedores como IP Address y Ports. Escriba `docker inspect IDdeContenedor` (sustituya IDdeContenedor con uno de los ID de contenedores de `docker ps -a`).

8. A menudo, los contenedores ejecutan procesos que "escuchan" en un puerto específico. Para comunicarse con el puerto del contenedor desde el host, debe mapear el puerto del host al contenedor. El formato del comando es `-p puertohost: puertocontenedor`. Si se ejecuta un contenedor de docker localmente, la dirección IP del contenedor es "localhost". En Windows puede ser 127.0.0.1. Un comando útil para resolver errores es consultar el archivo de registro del contenedor. Puede hacer esto fácilmente con el comando `docker logs containerId` (sustituya ID_{contenedor} con los ID de los contenedores de `docker ps -a`).

9. Los contenedores están hechos de capas. Se comienza con la capa base y se agregan capas adicionales. Docker ofrece el comando `history` para listar las capas de un contenedor. Escriba `docker history busybox:1.24`

10. Docker puede hacer mucho más, incluyendo construir imágenes más complejas con el comando "commit" o un Dockerfile (consulte <https://docs.docker.com/>). Esta breve introducción a Docker ofrece lo básico para poner en funcionamiento el contenedor de IRIS community.

1. Puede encontrar las instrucciones para usar el contenedor Docker para IRIS Community Edition en:
https://hub.docker.com/_/intersystems-iris-data-platform/plans/222f869e-567c-4928-b572-eb6a29706fbd?tab=instructions
2. Debemos explicar un comando adicional de Docker para configurar el archivo de contraseñas. Los contenedores de Docker no pueden acceder directamente al sistema de archivos del host. El contenedor debe montar un directorio que se pone a su disposición desde el host. Inicialmente, Docker para Mac pone a disposición los siguientes directorios para enlazar/montar en contenedores de Docker: /Users, /Volumes, /private, /tmp. Para almacenar contraseñas fuera del contenedor (recuerde que todos los archivos escritos en el contenedor desaparecerán al salir), necesitamos crear un archivo en /Users/suNombreDeUsuario/external/password.txt. Montar un volumen usa la marca -v y luego el nombre de host, seguido por dos puntos y luego el directorio del contenedor.
3. Ahora ejecute el contenedor con el siguiente comando: `docker run --name irisdb -v /external:/external -d store/intersystems/iris:2019.1.0.511.0-community --password-file /external/password.txt`. Después de ejecutar este comando y verificar el estado con el comando `docker ps -a`, debería ver lo siguiente tras un par de minutos:

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
d74f2b7a48d5	store/intersystems/iris:2019.1.0.511.0-community	"/iris-main --passwo..."	
2 minutes ago	Up 2 minutes (healthy)	irisdb	

4. Conéctese al contenedor recién iniciado mediante el comando: `docker exec -it irisdb iris session iris`. Esta conexión nos permite iniciar una sesión de Iris. El nombre de usuario es SuperUser y la contraseña es SYS. Ahora IRIS le solicitará cambiar la contraseña, como se muestra a continuación:

Node: d74f2b7a48d5, Instance: IRIS

Username: SuperUser

Password: ***

Password change required

Please enter a new password

Password: *****

Please retype your new password

Password: *****

USER>

Ahora podemos escribir comandos directamente en la línea de comandos del terminal IRIS. Como por ejemplo:

```
USER>set firstname = "Peter"
```

```
USER>write firstname
```

```
Peter
```

```
USER>
```

Podemos escribir comandos interactivos, ¿pero cómo iniciamos el portal de administración?

Normalmente, para hacer esto, abriríamos un navegador y escribiríamos la dirección:

<http://localhost:52773/csp/sys/UtilHome.csp>, pero cuando lo hacemos, se muestra un mensaje de error y el sitio del host local rechaza la conexión. ¿Recuerda la sección sobre mapeo de puertos y cómo los contenedores están aislados de la red externa sin ellos? Debemos mapear dos puertos, el puerto predeterminado de superserver de IRIS en 51773, y el puerto de administración en 52773.

En el caso anterior, primero detenga el proceso de línea de comandos escribiendo Halt. Luego, detenga el contenedor con: **docker stop [containerid]**

5. Volvamos a poner este contenedor en funcionamiento con mapeo de puertos: **docker run -p 51773:51773 -p 52773:52773 -v /external:/external -d store/intersystems/iris:2019.1.0.511.0-community --password-file /external/password.txt**

Si ahora abrimos una ventana del navegador con la dirección <http://localhost:52773/csp/sys/UtilHome.csp>, deberíamos acceder al portal de administración.

6. A pesar de que logramos usar el archivo de contraseñas en el sistema host, no pudimos mantener el cambio de contraseña que hicimos antes. ¿Por qué no? Debemos indicar a IRIS el directorio donde

guardar sus ajustes de configuración persistentes. Es necesario agregar una marca `--env`
`ISCDATADIRECTORY=/external/iconfig`.

Así, el comando completo es: `docker run -p 51773:51773 -p 52773:52773 -v /external:/external --env ISCDATADIRECTORY=/external/iconfig -d store/intersystems/iris:2019.1.0.511.0-community --password-file /external/password.txt`

7. En este punto, no debería tener que cambiar la contraseña del portal de administración que definió inicialmente.

8. La cadena del comando es difícil de recordar, y otro método es almacenarlo en un archivo que puede utilizar una utilidad de docker relacionada, llamado Docker-compose. El archivo es de tipo yml o yaml, y debe nombrarse docker-compose. El archivo docker-compose ofrece varias formas de configurar cualquier cantidad de contenedores y sus dependencias. Sin embargo, para nuestro objetivo alcanza con un archivo más simple. Aquí están los contenidos (debe nombrar al archivo `docker-compose.yml`):

```
version: '3'
```

```
services:
```

```
  iris:
```

```
    image: store/intersystems/iris:2019.1.0.511.0-community
```

```
    ports:
```

```
      - "51773:51773"
```

```
      - "52773:52773"
```

```
    volumes:
```

```
      - /external:/external
```

```
    environment:
```

```
      - ISCDATADIRECTORY=/external/iconfig
```

```
    entrypoint: /iris-main --password-file /external/password.txt
```

Tenga en cuenta el parecido con la línea de comandos con unos pocos añadidos. Primero, la versión del archivo docker-compose es la "3". Tenemos un servicio llamado "iris" u otro nombre a elección. La imagen, los puertos, los volúmenes y el entorno son autoexplicativos. La última línea, "entrypoint", es el programa que se ejecuta (incluyendo las opciones, en este caso la ubicación del archivo de contraseñas).

Si ahora escribimos `docker-compose up -d`. El contenedor se gira hacia arriba igual que antes. Para detener el contenedor, escriba `docker-compose down`.

Puede encontrar más información sobre las opciones de Docker-compose

en: <https://docs.docker.com/compose/>

[#Principiante](#) [#InterSystems IRIS](#)

URL de fuente: <https://es.community.intersystems.com/post/puesta-en-funcionamiento-de-iris-usando-docker>