InterSystems IRIS Basics: Run a Container

Version 2023.1
2024-07-11

Table of Contents

InterSystems IRIS Basics: Run a Container

Want to start up an InterSystems IRIS container? This document provides the basic procedure, including the following steps:

1. Select a Platform

2. Install Docker

3. Download the InterSystems IRIS Image

4. Add the License Key to the External Storage Location

5. Run a Container from the InterSystems IRIS Image

6. Use Durable %SYS to Store Instance-specific Data

For a quick, online hands-on exercise covering these steps and more, try Deploying and Customizing InterSystems IRIS Containers. For an overview of InterSystems IRIS containers and detailed information about using InterSystems IRIS container images and other images provided by InterSystems, see Running InterSystems Products in Containers (to which this document contains numerous links) and the other resources described in Learn More About InterSystems IRIS in Containers at the end of this document.

Note: Some of the commands included here may require root privileges.

1 Step 1: Select a Platform

The instructions in this hands-on were created for, and are most easily used in, a Linux environment. If you do not have a Linux system available, InterSystems recommends that you create an account on a public cloud provider such as Google Cloud Platform (GCP), Amazon Web Services (AWS), or Microsoft Azure and provision a CentOS VM with Docker installed.

Note: If need be, the instructions can be adapted to Windows or macOS. On Windows, you can execute the instructions in a Command Prompt or Windows PowerShell window, opened with **Run as administrator**; do not use PowerShell ISE. For more information about Docker for Windows, see Using InterSystems IRIS Containers with Docker for Windows on InterSystems Developer Community and the Docker documentation.

2 Step 2: Install the Container Runtime Engine

Install the container runtime engine(s) on which you intend to deploy InterSystems containers, such as Docker or Podman, on your servers. The specific instructions and procedures in this document are intended to be used with the Docker Engine.

Important: Container images from InterSystems comply with the Open Container Initiative (OCI) specification and are therefore supported on any OCI-compliant runtime engine on Linux-based operating systems, both on premises and in public clouds.

3 Step 3: Download the InterSystems IRIS Image

To make the InterSystems IRIS image from InterSystems available for use in this hands-on, you must download the image to the system you are working on. The following alternatives describe the InterSystems IRIS images that are or may be available to you.

- You can use an InterSystems IRIS Community Edition image from the InterSystems Container Registry (ICR), which includes repositories for all images available from InterSystems and is described in Using the InterSystems Container Registry. You can also download the Community Edition image from Docker Hub.

  InterSystems IRIS Community Edition comes with a free built-in 13-month license (and some functionality restrictions); if you use Community Edition in this hands-on, you won't have to provide a license key as described in the next step (Add the License Key to the External Storage Location). For more information, see Deploy InterSystems IRIS Community Edition on Your Own System in *Deploy and Explore InterSystems IRIS*.

  Note: Another option is to provision a cloud node hosting a running InterSystems IRIS Community Edition container on GCP, AWS, or Azure; for more information, see Deploy InterSystems IRIS Community Edition on a Cloud Node.

- If you are an InterSystems customer, you can use a released InterSystems IRIS image from the InterSystems Container Registry (ICR). Using the InterSystems Container Registry explains how to browse the InterSystems IRIS images available in the ICR web portal and use your WRC credentials to authenticate to the registry so you can download one.

- Your organization may have a private image registry that includes one or more InterSystems IRIS images. If so, obtain the location of the registry and the repository and tag for the image you need, as well as the credentials needed for access.

When you have identified the registry you want to download from and the credentials you need (if any), see Downloading the InterSystems IRIS Image for instructions for downloading the InterSystems IRIS image.

For simplicity, these instructions assume you are working with the image intersystems/iris:2022.3.0.205.0.

4 Step 4: Add the License Key to the External Storage Location

Like any InterSystems IRIS instance, an instance running in a container requires a license key (typically called iris.key).

The InterSystems IRIS Community Edition image available from the ICR or Docker Hub (described in the previous section) comes with a free built-in temporary license. Generally, however, license keys are not and cannot be included in an InterSystems IRIS container image, but instead must be copied into a container after it is started to be activated for the InterSystems IRIS instance running there. The **iris-main** program provides an option for this, but it requires you to place the license key in a storage location to be mounted as an external volume; instructions for using it are provided in the next section. To learn more about license keys for InterSystems IRIS containers, see License Keys for InterSystems IRIS Containers.

Copy your InterSystems IRIS license key file, iris.key, to the external storage location.

5 Step 5: Run a Container from the InterSystems IRIS Image

Once you have made the InterSystems IRIS image available on your local machine and have identified the external storage location and placed your license key on it, you are ready to use the **docker run** command to create and start a container. The **docker run** command actually combines three separate commands, as follows:

- **docker pull** — Downloads an image if it is not already present locally.

- **docker create** — Creates a container from the image.

- **docker start** — Starts the container.

Each of these commands is useful separately, for various purposes in different contexts. For more information, see Docker run reference in the Docker documentation.

A sample **docker run** command follows; all of its options are explained in the accompanying text. Note that options to the **docker run** command appear on the command line before the image specification, while options to the InterSystems **iris-main** program come after. (In this case, the **pull** command that is part of **docker run** is not needed, as you have already downloaded the **iris** image you want to use.)

```
docker run --name iris
  --detach
  --publish 52773:52773
  --volume /nethome/pmartinez/iris_external:/external
    intersystems/iris:2022.3.0.205.0
  --key /external/iris.key
```

- **--name** *container_name*

  Specifies the name of the container, which you can use to refer to the container in other Docker commands, for example **docker stop** *container_name* when you want to stop the container.

- **--detach**

  Runs the container in the background (and displays the container's unique ID).

- **--publish** *host_port:container_port*

  Publishes a port within the container to a port on the host so that entities outside the container (on the host or on other machines) can contact a program in the container. For example, an InterSystems IRIS instance's Management Portal is reached through the instance's web server port (52773 by default). If this port inside the container is published to a port on the host, the instance's Management Portal can be loaded into a browser using the host's port.

  --volume *external_storage_path:internal_volume*

  Mounts an external storage location accessible by the container as a storage volume inside the container. For information about which storage locations can be mounted in this way and Docker configuration that may be required, see Volumes in the Docker documentation.

  Important: The host file system location you mount and specify for this purpose must be writable by user 51773. (You will most likely need root privileges to effect this.)

  InterSystems does not support mounting NFS locations as external volumes in InterSystems IRIS containers.

- *repository/image***:***tag*

Specifies the image to be pulled and used to create a container (see Download the InterSystems IRIS Image). Use the **docker images** command to list available images and make sure you are specifying the right one.

- **--key** *license_key_path*

  An **iris-main** option that identifies the InterSystems IRIS license key to be installed in the instance in the container; this location must be on a mounted volume (see Add the License Key to the External Storage Location). When the container is running, **iris-main** continuously monitors the staged license key for changes; if any change is detected, it is copied to the current /mgr/ directory and activated.

Use the preceding sample and explanations to construct your own **docker run** command and execute it on the command line. When the command has completed, use the **docker ps** command to see your container in the list with a status of **Up**.

```
$ docker run --name iris --detach --publish 52773:52773
    --volume /nethome/pmartinez/iris_external:/external
    intersystems/iris:2022.3.0.205.0
    --key /external/iris.key
426d4a511d6746d89ec2a24cf93b29aa546ea696b479a52210d37da4c6d04883
$ docker ps
CONTAINER ID  IMAGE                                COMMAND               CREATED        STATUS
426d4a511d67  intersystems/iris:2022.3.0.205.0  "/iris-main --key ..."  5 seconds ago  Up 3 seconds
    PORTS                    NAMES
    0.0.0.0:52773->52773/tcp  iris
```

Note: The **--key** option is not needed with the InterSystems IRIS Community Edition image (see Download the Inter-Systems IRIS Image), which comes with a free built-in license.

If the image is not yet present locally but is in your organization's repository, Docker pulls (downloads) the image before creating and starting the container.

As shown in the example, after creating the container, Docker outputs the *UUID long identifier*; the first 12 characters make up the *UUID short identifier*, which is used to identify the container in other output, for example from the **docker ps** command.

6 Step 6: Use Durable %SYS to Store Instance-specific Data

Because a containerized application is isolated from the host environment, it does not write persistent data; whatever it writes inside the container is lost when the container is removed and replaced by a new container. Therefore, an important aspect of containerized application deployment is arranging for data to be stored outside of the container and made available to other and future containers.

The durable %SYS features enables persistent storage of instance-specific data — such as user definitions, audit records, and the log, journal, and WIJ files — when InterSystems IRIS is run in a container, allowing a single instance to run sequentially in multiple containers over time. For example, if you run an InterSystems IRIS container using durable %SYS, you can upgrade the instance by stopping the original container and running a new one that uses the instance-specific data created by the old one.

To try running the container you ran in the previous section with durable %SYS, make a single change to your **docker run** command line — specifying the durable %SYS directory to be created on your mounted external volume using the

ISC_DATA_DIRECTORY environment variable. Since we call that volume /external when running the container, let's call the durable %SYS directory to be created on it durable and modify the command line in the previous section as follows:

```
$ docker run --name iris-durable --detach --publish 53773:52773
    --volume /nethome/pmartinez/iris_external:/external
    --env ISC_DATA_DIRECTORY=/external/durable
    intersystems/iris:2022.3.0.205.0
    --key /external/iris.key
696b479a52210d37da4c6d04883426d4a511d6746d89ec2a24cf93b29aa546ea
$ docker ps
CONTAINER ID  IMAGE                            COMMAND              CREATED       STATUS
696b479a5221  intersystems/iris:2022.3.0.205.0 "/iris-main --key ..." 5 seconds ago  Up 3 seconds
    PORTS                    NAMES
    0.0.0.0:52773->52773/tcp  iris-durable
```

Important: The command line above changes the published host port of 52773 in the previous command line to 53773 and the name of the container to iris-durable. You can use port 52773 and the name iris as you did before, however, if you first stop and remove the previous container with the following commands:

```
$ docker stop iris
iris
$ docker rm iris
iris
```

To examine the instance-specific data of the InterSystems IRIS instance in the container, navigate to the durable %SYS directory on the mounted external volume on the host file system, for example:

```
$ cd /nethome/pmartinez/iris_external/durable
$ ls -l
total 30
drwxrwxr-x  7 irisowner irisowner  4096 Jan 19 10:03 csp
drwxr-xr-x  3 irisowner irisowner  4096 Jan 19 10:03 dist
drwxrwxr-x  5 irisowner irisowner  4096 Jan 19 10:03 httpd
-rw-rw-r--  1 irisowner irisowner 12713 Jan 19 10:04 iris.cpf
-rwxr-xr-x  1 irisowner irisowner  9527 Jan 19 10:04 _LastGood_.cpf
drwx------  1 irisowner irisowner  4096 Jan 30 18:27 mgr
```

For detailed information about running a container with durable %SYS, see Durable %SYS for Persistent Instance Data; for information about upgrading, see Upgrading InterSystems IRIS Containers.

7 Learn More About InterSystems IRIS in Containers

Use the documentation and resources below to continue exploring what containers and InterSystems IRIS have to offer.

• Docker Containers and InterSystems IRIS (video)

• Deploying and Customizing InterSystems IRIS Containers (above video and hands-on exercise)

• Running InterSystems Products in Containers (full documentation)

• Articles on InterSystems Developer Community:

  – What is a Container?

  – What is a Container Image?

  – Using InterSystems IRIS Containers with Docker for Windows

• Docker Documentation

• InterSystems Cloud Manager Guide — Use InterSystems Cloud Manager (ICM) to easily and intuitively provision infrastructure and deploy containers on it in a variety of ways. ICM brings the benefits of Infrastructure as Code (IaC),nd containerized deployment to InterSystems IRIS without requiring major investments in new technology, training,

configuration, and management. This guide contains documentation on both ICM and using InterSystems IRIS with Docker containers.