InterSystems API Manager Version 1.5 Guide

2024-05-06

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

InterSystems API Manager Version 1.5 Guide
InterSystems IRIS Data Platform 2024-05-06

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

1

Introduction to the InterSystems API Manager (IAM) Version 1.5

InterSystems API Manager (IAM) is a component of InterSystems IRIS® data platform that allows you to take advantage of microservices and APIs that are either exposed or consumed by your InterSystems IRIS applications. Acting as an API gateway between your InterSystems IRIS servers and applications, it gives you the ability to more effectively monitor and control the traffic of calls between your server-side APIs and your client-side applications. IAM enables you to monitor, control, and govern HTTP-based API traffic.

The more distributed your environment the more critical it becomes to properly govern and monitor your API traffic. IAM enables you to easily route all your traffic through a centralized gateway and forward API requests to appropriate target nodes.

The InterSystems API Manager is powered by Kong Enterprise Version 1.5.0.9, an industry-leading API manager from Kong, Inc. The InterSystems Worldwide Response Center (WRC) provides support for IAM including the Kong Enterprise features. The main source of information about IAM is provided by the Kong Enterprise 1.5 documentation. InterSystems provides information about installing and upgrading IAM and some details about the IAM environment.

For information on using the previous version of IAM, see InterSystems API Manager.

1.1 Benefits of Using IAM

Using IAM allows you to do the following:

- Monitor your HTTP-based API traffic and understand who is using your APIs; what are your most popular APIs and which could require a rework.
- Control who is using your APIs and restrict usage in various ways. From simple access restrictions to throttling API traffic and fine-tuning request payloads, you have fine-grained control and can react quickly.
- Protect your APIs with central security mechanisms like OAuth2.0 or Key Token Authentication.
- Onboard third-party developers and provide them with a superb developer experience right from the start by providing a dedicated Developer Portal for their needs.
- Scale your API demands and deliver low-latency responses.
- Provide a uniform API for underlying services that have different APIs.

1.2 Features of IAM

IAM has the following features:

- Kong Gateway — the open source foundation of Kong Enterprise. It is a lightweight, fast, and flexible cloud-native API gateway. It provides a central way to access REST APIs.
- Administrative API — provides a REST API to administer and configure services, routes, plugins, and consumers.
- Management Portal — allows you to administer and control the Kong Gateway. You can define security groups, policies and monitor performance.
- Developer Portal — allows you to provide access to developers, generate API documentation for REST interfaces through their specifications, manage different versions of REST APIs, and secure developer access.
- Plugins — provide advanced functionality such as authentication and rate limits. The InterSystems API Manager provides all of the Kong plugins provided by Kong Enterprise. You can purchase additional third-party plugins in the Kong marketplace, known as Kong Hub. InterSystems does not provide support for third-party plugins.
- Vitals — provide metrics about the health and performance of IAM and of the usage of the underlying services.

IAM provided with InterSystems IRIS must be used only for APIs that are either provided by InterSystems IRIS or used by InterSystems IRIS. If you want to extend the use of IAM to cover APIs that are independent of InterSystems IRIS, contact InterSystems sales for more information.

Note: The following features of Kong Enterprise are not included in the InterSystems API Manager: Kong Brain, Kong Immunity, and Kong Studio.

1.3 How to Learn About IAM

The main source of information about IAM is provided by the Kong Enterprise 1.5 documentation. The following topics provide installation instructions and upgrade instructions, The InterSystems documentation web site also provides a first look guide. The InterSystems online learning site has online classes, videos and documents that cover topics on IAM, such as building FHIR applications and best practices. On the online learning site, search for API Manager for these resources.

2

Installing IAM 1.5

This topic tells you how to install the InterSystems API Manager (IAM) from the installation tar file. IAM is provided in container format. You need software that supports the Open Container Initiative (OCI) to be able to install and run the IAM container. This topic uses Docker as an example of software that supports OCI. You can download the installation tar file from the InterSystems Worldwide Response Center (WRC) download page: https://wrc.intersystems.com/wrc/coDistGen.csp. To show only the IAM kits, type `IAM` in the Name column. IAM is distributed as a compressed tarball archive. Once you uncompress it and extract the files, you will find the following contents in the distribution kit:

- IAM Docker image, iam-image.tar — do not extract the files from this archive.
- scripts directory with:
  - docker-compose.yml script to start and stop IAM
  - unix and win directories with UNIX and Windows scripts to setup and test IAM — these optional scripts provide an easy way to start and test IAM. The startup scripts set up the environment variables in the current shell used by IAM. If you do not use the scripts, you need to define these variables some other way. The variables are described in the next section.
- readme.txt file with brief instructions for starting IAM. This topic is based on the readme but provides some additional information.
- EULA files with terms and conditions

2.1 Setting Up IAM

To set up IAM, follow these steps:

1. Ensure that your system has the required prerequisites:
   a. Install Docker if your system does not already have it. You must have the Docker engine version 17.04 or later. See Running InterSystems Products in Containers for a brief introduction to containers and Docker.
   b. Install the docker-compose command line interface version 1.12.0 or later if your system does not already have it.
   c. On UNIX systems, install the curl utility. The UNIX script files to start and test IAM use curl. The Windows script file uses PowerShell to invoke web requests and does not require curl.

d. Ensure that you have a running instance of InterSystems IRIS, InterSystems IRIS for Health, or HealthShare Health Connect that support IAM. InterSystems IRIS and InterSystems IRIS for Health first supported IAM in version 2019.1.1, and HealthShare Health Connect first supported IAM in version 2020.1.

e. Ensure that your InterSystems IRIS license file specifies “API Management enabled”. If it doesn't, contact InterSystems to obtain a license file that enables IAM.

f. Download the distribution archive (tarball file) with the IAM software and setup script files from the InterSystems WRC download site. You can download distribution archive from the following WRC download page: https://wrc.intersystems.com/wrc/coDistGen.csp. To show only the IAM kits, type IAM in the Name column.

g. Extract the files from the distribution archive.

2. Enable the IAM user and web application on the InterSystems IRIS instance. The purpose of the IAM user is to allow the setup script to get a copy of the IAM license from the instance of InterSystems IRIS; the IAM user has very limited privileges and is only used to access the IAM license information. In the Management Portal for the instance of InterSystems IRIS, InterSystems IRIS for Health, or HealthShare Health Connect:

   a. Select **System Administration** > **Security** > **Applications** > **Web Applications** and select the `/api/iam` web application.

   b. Select the **Enable Application** check box.

   c. Select **Save**.

   d. Select **System Administration** > **Security** > **Users** and select the IAM user.

   e. Enter a new password and select the **User enabled** check box.

   f. Select **Save**.

3. Execute the following command (in the directory where you extracted the IAM archive) to create the container and load the IAM image:

   ```
   docker load -i iam_image.tar
   ```

4. If you want to secure the IAM administrative API, you must add an authentication plug-in to IAM. You should set a password for the IAM Super Admin account by defining the environment variable KONG_PASSWORD in the container before running the setup script. You can either define it in the shell or in the docker-compose file. This step must be performed before you execute `docker-compose` for the first time. If this password is defined in the container, IAM creates a user, `kong_admin`, and specifies the value as the password for an account that can be used to log in to IAM Manager or to make Admin API requests when RBAC is enabled. For information on how to enable RBAC, see the corresponding section in Kong documentation.

5. Run the setup script and start IAM:

   a. Run the IAM setup script. If you are running the script on UNIX, you must ensure that the script is run in the same process, not a subprocess by using the bash source command or the dot command. In a Windows PowerShell, enter:

      ```
      .\scripts\win\iam-setup.ps1
      ```

      In a UNIX bash shell, enter:

      ```
      source ./scripts/unix/iam-setup.sh
      ```

      Or in another UNIX shell, enter the equivalent dot command:

      ```
      . ./scripts/unix/iam-setup.sh
      ```

b. Enter the full image repository, name and tag for your IAM docker image. For example, it could be:

   `intersystems/iam:1.5.0.9-4`

c. Enter the IP address for your InterSystems IRIS instance. If your instance is on your local machine, please use your externally visible local IP address, not "localhost". If the instance is running in a container, use the IP address of the host environment, not the IP address of the container. To avoid any DNS issues, use the numeric form of the IP address.

d. Enter the web server port for your InterSystems IRIS instance.

e. Enter the password for the IAM user for your InterSystems IRIS instance.

f. Re-enter the password.

g. If you want IAM to request the license from InterSystems IRIS using HTTPS instead of HTTP, provide the full path to your CA Certificate file; otherwise, type the Enter key.

h. Confirm your entries.

This script sets the two environment variables required by the docker-compose file

2.2 Environment Variables

The following environment variables are used by the docker-compose file and by IAM. These are set by the startup script. If you do not use the startup script, you must define these variables.

- ISC_IAM_IMAGE — contains the respository, name, and tag of the IAM docker image. The docker-compose file uses this variable to access the docker image. The value has the format:

  repository`/`*name*`:`*tag*

- ISC_IRIS_URL — contains the URL to access the InterSystems IRIS instance to get the IAM license. The docker-compose file defines this environment variable. The value has the format:

  `http://IAM:`*password*`@`*ip-address*`:`*port-number*`/api/iam/license`

- ISC_CA_CERT — optionally contains the contents of the CA certificate file for the server running InterSystems IRIS. If local policy requires that HTTPS be used for communication, then this environment variable must contain the contents of the server's CA certificate.

These environment variables are defined in the shell and allow the docker-compose file to access the IAM container and the InterSystems IRIS image. If you are not in the shell where you executed the setup script, these environment variables are not defined. You can either re-run the script or define them in another way.

By default IAM listens on the following ports:

- `:8000` on which IAM listens for incoming HTTP traffic from your clients, and forwards it to your upstream services.
- `:8443` on which IAM listens for incoming HTTPS traffic. This port has a similar behavior as the `:8000` port, except that it expects HTTPS traffic only. This port can be disabled via the configuration file.
- `:8003` on which IAM listens for the Developer Portal GUI traffic — if the Developer Portal is enabled.
- `:8004` on which IAM listens for the Developer Portal `/files` traffic — if the Developer Portal is enabled.
- `:8001` on which the Administration API listens.
- `:8444` on which the Administration API listens for HTTPS traffic.

- `:8002` on which IAM listens.
- `:8445` on which IAM listens for HTTPS traffic.

2.3 Start and Test API Manager

To start IAM, navigate to the /scripts directory with the docker-compose.yml file and execute the following command to start IAM:

```
docker-compose up -d
```

You can access the user interface at `http://localhost:8002/`.

To test the IAM setup, navigate to the directory with the scripts for the operating system you are running (`scripts/win` or `scripts/unix`), and run the `iam-test` script. This script sets up a route and a service in IAM and allows you to check connectivity with your InterSystems IRIS instance.

2.4 Stop API Manager

To stop the IAM container navigate to the directory with the docker-compose.yml file and execute the following command:

```
docker-compose down
```

Note that you need to be in the same shell as the one that you ran the setup scripts or you need to define the `ISC_IAM_IMAGE` and `ISC_IRIS_URL` environment variables.

2.5 Restart API Manager

To restart the IAM container, navigate to the /scripts directory with the docker-compose.yml file and execute the following command to start IAM:

```
docker-compose up -d
```

Note that you need to be in the same shell as the one that you ran the setup scripts or you need to define the `ISC_IAM_IMAGE` and `ISC_IRIS_URL` environment variables.

3

Upgrading IAM

This topic describes how to upgrade the InterSystems API Manager (IAM) from version 0.34–1 to version 1.5.

3.1 Migration Steps from 0.34 to 1.5

Due to iterative changes between IAM version 0.34-1 and IAM version 1.5, it is necessary to complete four total migrations through three intermediary releases. The steps for each migration between versions are identical, and can be found in the Kong documentation. You may perform each step manually, or create docker-compose files to assist you (as described in Running a Migration Using Docker Compose).

To upgrade from version 0.34-1 to version 1.5, you must perform the following migrations in order:

Order	Start Version	End Version	Migration Instructions
1	*0.34-1*	0.35	https://docs.konghq.com/enterprise/0.35-x/deployment/migrations/
2	0.35	0.36-x	https://docs.konghq.com/enterprise/0.36-x/deployment/migrations/
3	0.36-x	1.3.x	https://docs.konghq.com/enterprise/1.3-x/deployment/migrations/
4	1.3.x	*1.5*	https://docs.konghq.com/enterprise/1.5.x/deployment/migrations/

To upgrade IAM from 0.34 to 1.5, you can download the installation tar files from the InterSystems Worldwide Response Center (WRC) download page: https://wrc.intersystems.com/wrc/coDistGen.csp. To show only the IAM kits, type `IAM` in the Name column. You need to download the compressed tarball archives for the 1.5.0.9 version and the three "for upgrade only versions," 0.35–5, 0.36.6, and 1.3.0.2. Once you uncompress and extract the files for each distribution kit, load the IAM image into your local repository by executing the following command in the directory where you extracted the IAM archive:

Docker Compose

```
docker load -i iam_image.tar
```

Podman Compose

```
podman load -i iam_image.tar
```

Then perform the upgrade following the procedures in this topic. See "Installing IAM" for a list of the contents in each distribution kit.

To see what features have been added in these releases see the Kong Enterprise changelog from release 1.5.0.9 to release 0.35.

3.2 Running a Migration Using Docker Compose

Upgrading from IAM version 0.34-1-x to version 1.5.0.3-x requires several incremental migrations, as described above. One way to run these migrations is by creating a series of docker-compose files to perform each step. Using docker-compose files to run the migrations can streamline the process.

The following two sections describe how to create the docker-compose files, and then how to run them.

3.2.1 Creating the Docker Compose Files

As each IAM cluster is unique, you must create the docker-compose files based on the configuration of your IAM cluster. You should create all these files before you start the migration. Be sure to locate all the docker-compose files in the same directory as the main docker-compose.yml file used to start and stop IAM; this ensures the files share the same network.

As an example, this section describes the process for a single-node IAM cluster:

1. Begin by creating a copy of the main docker-compose.yml file. Name the copy step1.yml.
2. Next, edit step1.yml to perform the first step of the migration: create a new node of the target version (for the first migration, 0.35) that points to the same datastore, and run `kong migrations up`. This may look like:

```
version: '3.2'
services:
  iam-migrations:
    image: intersystems/iam:0.35-5-1
    command: kong migrations up
    depends_on:
      - db
[...]
  iam-1:
    image: intersystems/iam:0.34-1-1
[...]
```

3. Then, create another new docker-compose file and name it step2.yml. When run, this file should provision a node of the target version to replace the older node. In the example below, the old "iam-1" node is decommissioned, and a new "iam-1" node is provisioned with the newer version of IAM.

```
version: '3.2'
services:
  iam-1:
    image: intersystems/iam:0.35-5-1
    depends_on:
      - db
[...]
```

   For multi-node clusters, you will need to create additional docker-compose files to reprovision each node in your cluster. As this is a single-node example, we can move on to the last step of the migration.

4. Finally, create the next docker-compose file, step3.yml. This file should finish the current migration when run (in the case of this example, 0.34-1 => 0.35).

```
version: '3.2'
services:
  iam-migrations:
    image: intersystems/iam:0.35-5-1
    command: kong migrations finish
    depends_on:
      - db
[...]
  iam-1:
    image: intersystems/iam:0.34-1-1
[...]
```

5. Now, repeat this entire process for the next migration, starting by creating a copy of step3.yml and naming it step4.yml. Continue until you have created a set of docker-compose files for each incremental migration.

When you are done, you should have four sets of docker-compose files (one for each incremental migration). In this example of a single-node cluster, you should have files from step1.yml to step12.yml.

3.2.2 Running the Docker Compose Files

Once all the files are created, you can run them in sequence to follow the instructions in Migration Steps. The command to run the docker-compose file is:

```
docker-compose -f step#.yml up -d
```

where `step#.yml` is the name of the file to run.