



InterSystems API Manager Version 3.2 Guide

2024-05-06

InterSystems API Manager Version 3.2 Guide
InterSystems IRIS Data Platform 2024-05-06
Copyright © 2024 InterSystems Corporation
All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, and TrakCare are registered trademarks of InterSystems Corporation. HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, and InterSystems TotalView™ For Asset Management are trademarks of InterSystems Corporation. TrakCare is a registered trademark in Australia and the European Union.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

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 3.2	1
1.1 Benefits of Using IAM	1
1.2 Learning About IAM	2
2 Installing IAM Version 3.2	3
2.1 New in IAM 3.2	3
2.2 Steps Required to Install IAM	3
2.3 Ensure Your Linux System Has the Required Prerequisites	3
2.4 Download and Extract the IAM Installation Files	4
2.5 Enable Your InterSystems IRIS Instance to Use IAM	4
2.6 Set Up IAM	5
2.7 Start IAM	6
2.8 Test IAM	6
2.9 Stop IAM	7
2.10 Environment Variables	7
2.11 Ports	7
2.12 Troubleshooting	8
2.12.1 Get the IAM Logs	8
2.12.2 Error Message "FATAL: sorry, too many clients already" Appears in the Logs	8
2.12.3 "error calculating" Appears in the IAM Portal	8
2.12.4 Cannot Get IAM License	9
2.12.5 IAM Management Portal or Developer Portal Is Empty	9
3 Upgrading to IAM Version 3.2	11
3.1 New in IAM 3.2	11
3.2 Upgrade Steps	11
3.3 Perform the Upgrade	11
3.3.1 Stop IAM 3.0	11
3.3.2 Load the IAM 3.2 Image and Set Up Environment Variables for IAM 3.2	12
3.3.3 Start IAM 3.2	13

1

Introduction to the InterSystems API Manager (IAM) Version 3.2

InterSystems API Manager (IAM) 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 Gateway Version 3.2, an industry-leading API manager from Kong, Inc. While the [InterSystems Worldwide Response Center \(WRC\)](#) provides support for IAM, the main source of information about IAM is the Kong Gateway 3.2.x documentation, which can be found at <https://docs.konghq.com/gateway/3.2.x/>. InterSystems provides information about installing and upgrading IAM, along with some details about the IAM environment.

For information on other versions of IAM, see [InterSystems API Manager \(IAM\)](#).

Note: In this article, InterSystems IRIS® refers to any InterSystems product based on InterSystems IRIS that supports IAM. This includes InterSystems IRIS for Health and HealthShare® Health Connect.

1.1 Benefits of Using IAM

IAM allows you to do the following:

- Monitor your HTTP-based API traffic to find out:
 - Which developers or applications are using your APIs
 - What APIs are being called and which ones are most popular
 - How often your APIs are being called
 - Which APIs could benefit from a rework
 - If there are any patterns or trends
- 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.
- Blue-green deployments.
- Canary releases.

Note: IAM provided with InterSystems IRIS must be used only for APIs that are either provided by InterSystems IRIS or used by InterSystems IRIS.

1.2 Learning About IAM

The IAM Version 3.2 documentation provided by InterSystems includes:

- [Installing IAM](#)
- [Upgrading IAM](#)

For documentation of previous IAM releases, see [InterSystems API Manager \(IAM\)](#).

See the Kong Gateway 3.2.x documentation at <https://docs.konghq.com/gateway/3.2.x/> for details on using IAM.

For additional information about IAM, see the InterSystems Developer Community [article](#) and the InterSystems [online learning site](#). The 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` to find these resources.

2

Installing IAM Version 3.2

This article tells you how to install, start, stop, and test the InterSystems API Manager (IAM). IAM must be installed on a Linux system.

If you are upgrading your existing version of IAM, see [Upgrading to IAM Version 3.2](#).

Since IAM is provided in container format, you need software that supports the [Open Container Initiative \(OCI\)](#) to install and run the IAM container. This article uses Docker as an example of software that supports OCI.

Note: In this article, InterSystems IRIS® refers to any InterSystems product based on InterSystems IRIS that supports IAM. This includes InterSystems IRIS for Health and HealthShare® Health Connect.

2.1 New in IAM 3.2

- Kong Gateway 3.2 contains many new features, deprecations, and fixes that may affect existing applications. See the changelog at <https://docs.konghq.com/gateway/changelog/#3210> for details.

2.2 Steps Required to Install IAM

1. Ensure the Linux system where you will be installing IAM has the required prerequisites
2. Download and extract the IAM installation files from the tar file
3. Enable the InterSystems IRIS instance that IAM will connect with, to use IAM
4. Set up IAM

2.3 Ensure Your Linux System Has the Required Prerequisites

1. You need to install the following tools if you do not already have them:

- a. Docker or Podman. See [Running InterSystems Products in Containers](#) for a brief introduction to containers and Docker.
 - b. Docker Compose or Podman Compose command line interface tool.
 - c. Curl. The script files that set up and test IAM use curl.
2. Ensure that you have a running instance of InterSystems IRIS, InterSystems IRIS for Health, or HealthShare Health Connect that supports IAM:
 - InterSystems IRIS and InterSystems IRIS for Health first supported IAM in version 2019.1.1.
 - HealthShare Health Connect first supported IAM in version 2020.1.

2.4 Download and Extract the IAM Installation Files

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 tar file. Choose the 64 bit kit if you are installing on x86-64 architecture, or choose the ARM 64 kit if you are installing on ARM64 architecture. Once you uncompress it and extract the files, you will find the following contents in the distribution kit:

- `iam-image.tar` — This is the IAM Docker image. Do not extract the files from the archive at this time.
- `scripts` directory containing:
 - `docker-compose.yml` script to start and stop IAM
 - `iam-setup.sh` script to provide an easy way to set up the [environment variables](#) required by IAM
 - `iam-test.sh` script to provide an easy way to test connectivity to your InterSystems IRIS instance
 - `iam-upgrade-db` directory which contains the tools and scripts required to upgrade to IAM 3.2
- `readme.txt` file containing brief instructions for starting IAM.
- `EULA.pdf` file containing terms and conditions

2.5 Enable Your InterSystems IRIS Instance to Use IAM

1. Ensure your InterSystems IRIS license specifies “API Management”. If it doesn't, contact InterSystems to obtain a license that enables IAM.
 - a. In the Management Portal, select **System Administration** > **Licensing** > **License Key**.
 - b. Look in the “Authorization Key” section. If “-API Management” is listed, this license will allow the instance to use IAM.
2. Enable the IAM user. 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 that it will connect with. The IAM user has very limited privileges and is only used to access the IAM license information.
 - a. In the Management Portal, select **System Administration** > **Security** > **Users** and select the `IAM` user

- b. Click the **Password** radio button
 - c. Enter and confirm a password for the IAM user
 - d. Select the **User enabled** check box
 - e. Select **Save**
3. Enable the IAM web application.
 - a. In the Management Portal select **System Administration > Security > Applications > Web Applications** and select the `/api/iam` web application
 - b. Select the **Enable Application** check box
 - c. Select **Save**

2.6 Set Up IAM

1. Load the IAM image into your local repository by executing the following command in the directory where you extracted the IAM archive:

Docker

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

Podman

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

2. Make a note of the value of `Loaded image` from the output of the `docker load` command. It is required in step 3b.
3. Run the IAM setup script and enter the requested information at the prompts.

- a. In a UNIX bash shell, enter:

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

- b. At the first prompt, enter the container image name. The container image name can be found in the output of the `docker load` command, as the value of `Loaded image`. For example, it could be:

```
intersystems/iam:3.2.1.0
```

- c. Enter the IP address for your InterSystems IRIS instance. If your instance is on your local machine, use your externally visible local IP address, not `localhost` or `127.0.0.1`. 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 on 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, press **Enter**.

- h. With certain InterSystems IRIS configurations, the instance is not accessible by using the instance server name. In these cases, your InterSystems IRIS instance is only accessible via its CSPConfigName URL prefix (see [Changing the InterSystems IRIS Server Name in the URL](#)) and you need to provide the prefix with a trailing slash (/) now. If this does not apply, press **Enter**.
- i. To continue and use the information you entered, press **y** then press **enter**.

This script sets the [environment variables](#) required by the docker-compose.yml file.

2.7 Start IAM

The docker-compose.yml file provides a convenient way to start and stop IAM in the docker container. You must be in the same shell as the one in which you ran the setup script or you will need to define the *ISC_IAM_IMAGE* and *ISC_IRIS_URL* [environment variables](#). To start IAM, navigate to the scripts directory containing the docker-compose.yml file and execute the following command:

Docker Compose v2

```
docker compose up -d
```

Docker Compose v1

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

Podman Compose

```
podman-compose up -d
```

You can now access the IAM management portal at <http://localhost:8002/> and the developer portal at 127.0.0.1:8003.

Note: The docker-compose.yml file defines the URLs that are used to access the IAM management portal and the developer portal. To avoid Cross-Origin Resource Sharing (CORS) errors when accessing the IAM management and developer portals, the URLs that you use must match the URLs defined in the docker-compose.yml file in the *KONG_ADMIN_GUI_URL* and *KONG_PORTAL_GUI_HOST* environment variables. The default values of these are http://localhost:8002 for the management portal and 127.0.0.1:8003 for the developer portal. If you will be using different URLs to access these portals, you must edit the docker-compose file before you start IAM. For details on how Kong Enterprise handles CORS and other DNS issues, see [DNS Considerations for Kong Gateway](#).

2.8 Test IAM

To test the IAM setup, use the same shell as the one in which you ran the setup script or you will need to define the *ISC_IAM_IMAGE* and *ISC_IRIS_URL* [environment variables](#). Navigate to the scripts directory with the docker-compose.yml file and execute the following command:

```
./iam-test.sh
```

This script sets up a route and a service in IAM. To test if requests to IAM are successfully forwarded to your InterSystems IRIS instance, run the curl command provided by the test script.

2.9 Stop IAM

You must be in the same shell as the one in which you ran the setup script or you will need to define the `ISC_IAM_IMAGE` and `ISC_IRIS_URL` environment variables.

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

Docker Compose v2

```
docker compose down
```

Docker Compose v1

```
docker-compose down
```

Podman Compose

```
podman-compose down
```

2.10 Environment Variables

The following environment variables are used by the `docker-compose.yml` file. They are set by the setup script. If you do not use the setup script, you must define these variables:

- `ISC_IAM_IMAGE` — set to the container image name of your IAM docker image. The value has the format:
repository/name:tag
- `ISC_IRIS_URL` — set to the username, password, and URL of the InterSystems IRIS instance where IAM will get license from. The value has the format:
`IAM:password@ip-address:port-number/api/iam/license`
where *password* is the password of the IAM account on the InterSystems IRIS instance and *ip-address:port-number* are the IP address and web server port of the instance.
- `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 `docker-compose` to access the IAM container. 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.

2.11 Ports

By default, IAM listens on the following ports:

- `:8000` IAM listens for incoming HTTP traffic from your clients and consumers, and forwards it to your upstream services.

- :8443 IAM listens for incoming HTTPS traffic. This port has similar behavior to port :8000, except that it expects HTTPS traffic. This port can be disabled via the configuration file.
- :8003 IAM listens for HTTP Developer Portal GUI traffic, if the Developer Portal is enabled.
- :8004 IAM listens for Developer Portal /files traffic over HTTP, if the Developer Portal is enabled.
- :8001 Administration API. Listens for calls from the command line over HTTP.
- :8444 Administration API. Listens for calls from the command line over HTTPS.
- :8002 IAM listens for HTTP management portal GUI traffic.
- :8445 IAM listens for HTTPS management portal GUI traffic.

2.12 Troubleshooting

This topic covers some common issues you can run into when installing or running IAM.

2.12.1 Get the IAM Logs

It is a good practice to check the logs after installing or updating IAM. If you run into any problems, start your troubleshooting by viewing the logs. When using Docker or Podman and the supplied setup scripts, you can find the logs with:

Docker Compose v2

```
docker logs scripts-iam-1
docker logs scripts-db-1
```

Docker Compose v1

```
docker logs scripts_iam_1
docker logs scripts_db_1
```

Podman Compose

```
podman logs scripts_iam_1
podman logs scripts_db_1
```

2.12.2 Error Message "FATAL: sorry, too many clients already" Appears in the Logs

The number of NGINX worker processes is being computed incorrectly in the IAM container. This problem can be resolved by explicitly setting the number of worker processes. You can do this by adding the following line to the iam.environment section of the docker-compose.yml file:

```
KONG_NGINX_WORKER_PROCESSES: 15
```

The correct number of worker processes can be difficult to determine. See [Kong's documentation](#) for more information.

2.12.3 "error calculating" Appears in the IAM Portal

The number of NGINX worker processes is being computed incorrectly in the IAM container. This problem can be resolved by explicitly setting the number of worker processes. You can do this by adding the following line to the iam.environment section of the docker-compose.yml file:

KONG_NGINX_WORKER_PROCESSES: 15

The correct number of worker processes can be difficult to determine. See [Kong's documentation](#) for more information.

2.12.4 Cannot Get IAM License

There are multiple reasons why IAM cannot communicate with the InterSystems IRIS instance to get the license. This problem shows up in the following ways:

- The `iam-setup.sh` script displays one of the following:
 - The `/api/iam` web application is disabled. Please enable it before running this script again.
 - Authorization failed. Please make sure to enable the IAM user and reset its password before running this script again. This error may also mean that you entered the wrong password to this script.
 - No content. Either your InterSystems IRIS instance is unlicensed or your license key does not contain an IAM license.
 - Request failed with a 400 status code. You may be trying to use HTTP on an SSL-enabled server port.
 - Couldn't reach InterSystems IRIS at `$ip:$port`. One or both of your IP and Port are incorrect.
- In some rarer cases, IAM may not succeed in getting the license even if the script reports “Successfully got IAM license!”. In this case, the symptom would be “could not decode license JSON: No license found” appearing in the IAM log. This condition could be caused by a network setup where the script running on the host has access to the InterSystems IRIS instance but IAM running in a container does not have access to it.

One other reason that IAM cannot access the InterSystems IRIS instance, is if it is configured so that the instance is not accessible by the default URL with the server name or the `/api` interface is blocked. In this case, you must specify the `CSPConfigName` URL prefix in the startup script (see [Changing the InterSystems IRIS Server Name in the URL](#)).

2.12.5 IAM Management Portal or Developer Portal Is Empty

If you enter the URL for the IAM management portal or the developer portal and the portal does not display with content, it is possible that you have entered a URL that may seem correct but is not the URL specified by the `docker-compose.yml` file. The `docker-compose.yml` file defines the URLs that are used to access the IAM management portal and the developer portal. To avoid Cross-Origin Resource Sharing (CORS) errors when accessing the IAM management and developer portals, the URLs that you use to access them must match the URLs defined in the `docker-compose.yml` file in the `KONG_ADMIN_GUI_URL` and `KONG_PORTAL_GUI_HOST` [environment variables](#). The default values of these are `http://localhost:8002` for the management portal and `127.0.0.1:8003` for the developer portal. If you will be using different URLs to access these portals, you must edit the `docker-compose.yml` file before you start IAM. For details on how Kong Enterprise handles CORS and other DNS issues, see [DNS Considerations for Kong Gateway](#).

3

Upgrading to IAM Version 3.2

This topic describes how to upgrade the InterSystems API Manager (IAM) from version 3.0 to version 3.2.

3.1 New in IAM 3.2

- Kong Gateway 3.2 contains many new features, deprecations, and fixes that may affect existing applications. See the changelog at <https://docs.konghq.com/gateway/changelog/#3210> for details.

3.2 Upgrade Steps

To upgrade from 3.0 to 3.2:

1. 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.
2. Uncompress the file and extract the contents.
See [Installing IAM](#) for a list of the contents.
3. [Perform the upgrade](#) which consists of the following high level steps:
 - a. Stop IAM 3.0
 - b. Load the IAM 3.2 image and set up environment variables for IAM 3.2
 - c. Start IAM 3.2

3.3 Perform the Upgrade

3.3.1 Stop IAM 3.0

Before upgrading to IAM 3.2, you must ensure your current version of IAM is no longer running. To stop the IAM container you must be in a shell where you have previously run the setup script for your current version of IAM, or you will need to

define the `ISC_IAM_IMAGE` and `ISC_IRIS_URL` [environment variables](#). Navigate to the scripts directory containing the `docker-compose.yml` file for your current IAM version, and stop it, for example:

Docker Compose v2

```
cd iam3.2/scripts
docker compose down
```

Docker Compose v1

```
cd iam3.2/scripts
docker-compose down
```

Podman Compose

```
cd iam3.2scripts
podman-compose down
```

3.3.2 Load the IAM 3.2 Image and Set Up Environment Variables for IAM 3.2

1. 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
```

2. Make a note of the value of `Loaded image` from the output of the `docker load` command. It is required in step 3b.
3. Run the IAM setup script and enter the requested information at the prompts.
 - a. In a UNIX bash shell, enter:

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

- b. At the first prompt, enter the container image name. The container image name can be found in the output of the `docker load` command, as the value of `Loaded image`. For example, it could be:

```
intersystems/iam:3.2.1.0
```

- c. Enter the IP address for your InterSystems IRIS instance. If your instance is on your local machine, use your externally visible local IP address, not `localhost` or `127.0.0.1`. 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 on 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, press **Enter**.
- h. With certain InterSystems IRIS configurations, the instance is not accessible by using the instance server name. In these cases, your InterSystems IRIS instance is only accessible via its `CSPConfigName` URL prefix (see

Changing the InterSystems IRIS Server Name in the URL) and you need to provide the prefix with a trailing slash (/) now. If this does not apply, press **Enter**.

- i. To continue and use the information you entered, press **y** then press **enter**.

This script sets the `ISC_IAM_IMAGE` and `ISC_IRIS_URL` environment variables required by the `docker-compose.yml` file.

3.3.3 Start IAM 3.2

Navigate to the directory where you extracted the IAM archive. For example, `/users/iam/iam32`. Run the following commands:

Docker Compose v2

```
cd scripts
docker compose up -d
```

Docker Compose v1

```
cd scripts
docker-compose up -d
```

Podman Compose

```
cd scripts
podman-compose up -d
```

After waiting a minute, go to `http://localhost:8002/overview` to verify that everything is running.

