Deploy a System Alerting and Monitoring Image

2024-05-06

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

Deploy a System Alerting and Monitoring Image
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

Deploy a System Alerting and Monitoring Image

Important: System Alerting and Monitoring (SAM) has been deprecated; the following documentation is provided for existing users only. Customers interested in a comprehensive view of their operational platform can access the metrics API and structured logs of InterSystems products within another observability tool. Existing users who would like assistance identifying an alternative solution should contact the WRC.

System Alerting and Monitoring (*SAM*) is a *containerized* application that integrates an InterSystems IRIS® instance called the SAM Manager with multiple open-source applications (Prometheus, Grafana, Alertmanager, and Nginx) to provide a resilient and scalable monitoring platform for InterSystems products.

This page will help you choose the appropriate version of SAM for your system and start it for the first time.

1 Choose a Version of SAM

All released versions of System Alerting and Monitoring (SAM) allow you to group InterSystems IRIS instances into clusters, track their performance in real time, and create custom alerts using an intuitive web interface or a REST API. All released versions of SAM support monitoring instances of InterSystems IRIS version 2020.1 and later.

1.1 SAM 1.1 Release Notes

SAM version 1.1 provides performance improvements for the graphs in the Grafana dashboard and the underlying Prometheus queries, especially when displaying metrics over a longer period of time.

If you are currently using SAM 1.0, you can upgrade to SAM 1.1.

1.2 SAM 2.0 Release Notes

SAM version 2.0 adds several additional features. These include:

- An easy to use tool for exporting and importing SAM cluster configurations as JSON files
- Support for SSL/TLS connections between SAM and your IRIS instances
- Support for defining instances with URL prefixes, providing added flexibility for systems hosting multiple IRIS instances at the same IP address
- Custom Grafana dashboards, allowing you to toggle between multiple perspectives of instance performance

1.2.1 Upgrading from SAM 1.0 or SAM 1.1

SAM versions 1.0 and 1.1 do not include utilities for exporting data about your system configuration for import into other instances of SAM. This means that it is not possible to upgrade seamlessly from these earlier versions to version 2.0 or above; all cluster and instance configuration data from the previous version is lost, and you must define and configure your clusters and instances again in SAM 2.0.

Before you can deploy a new version of SAM 2.0 on a system that previously ran SAM 1.0 or 1.1, you must ensure that the containerization engine initializes new volumes for SAM 2.0 instead of attempting to reuse the volumes it created for the previous version. To do so, you can either delete the volumes which the previous version used, or you can specify a different project name for your SAM 2.0 deployment in the start.sh and stop.sh scripts as follows:

1. Open the start.sh file for SAM 2.0 in a text editor and locate the following command line:

   ```
   docker-compose -p sam up -d
   ```

2. In this command, replace `sam` with a different project name. For instance, if you wish to initialize SAM 2.0 as `sam2`, the command should now read `docker-compose -p sam2 up -d`.
3. Open the stop.sh file for SAM 2.0 in a text editor and locate the following line:

   ```
   docker-compose -p sam down
   ```

4. In this command, replace `sam` with the same project name you specified for SAM 2.0 in the start.sh file during step 2. Taking the same example name as before (`sam2`) the command should now read `docker-compose -p sam2 down`.

After you have made this change, running the .start.sh script generates new volumes (as well as other entities necessary for the environment) using names that begin with the project name you have specified.

2 Prepare Your Instances for Monitoring

In order for System Alerting and Monitoring (SAM) to monitor an InterSystems IRIS instance, the following must be true:

- The target instance is version 2020.1 or higher
- The /api/monitor endpoint allows unauthenticated access
- The built-in monitoring tools are enabled for the target instance
- The instance must be uniquely identifiable
- For interoperability productions: collection of metrics must be enabled

2.1 Allow Unauthenticated Access to the /api/monitor/ Endpoint

Each InterSystems IRIS instance version 2020.1 or higher contains the /api/monitor web application. In order for SAM to collect metrics and alerts from the /api/monitor endpoint, the endpoint must allow for unauthenticated access. To make sure this is the case:

1. Open the Management Portal of the InterSystems IRIS instance you would like to add to SAM.
2. Go to the **Web Applications** page (**System Administration** > **Security** > **Applications** > **Web Applications**).
3. Select /api/monitor to open the **Edit Web Application** page.
4. In the **Security Settings** section, select **Unauthenticated**.

For more information about the /api/monitor/ web application, see Monitoring InterSystems IRIS Using REST API in *Monitoring Guide*.

2.2 Enable the Built-in Monitoring Tools

InterSystems IRIS instances have built-in monitoring tools, and SAM uses these tools to collect information about the instance. Check that the following tools are enabled on the InterSystems IRIS instance:

- System Monitor, which SAM uses when determining the instance state. By default, System Monitor is enabled.
- Log Monitor, which enables SAM to see alerts from the instance. By default, Log Monitor is enabled and writes alerts to the alerts log.

  Important: SAM is only able to view instance alerts if Log Monitor writes them to the alerts log. If Log Monitor sends alerts by email instead of by writing them to the alerts log, SAM cannot view alerts for the instance.

2.3 Ensure that the Instance is Uniquely Identifiable

In order for SAM to monitor an InterSystems IRIS instance, the instance must be uniquely identifiable.

For SAM 1.0 and 1.1, an instance must be uniquely identifiable by a combination of **IP** (or domain name) and **Port**.

For SAM 2.0, an instance must be uniquely identifiable by a combination of **IP** (or domain name), **Port**, and **URL Prefix**.

Note: SAM 1.0 and 1.1 *do not* support connecting to an InterSystems IRIS instance with a URL prefix; URL prefixes are most common when multiple InterSystems IRIS instances are located on the same system.

For more information about configuring multiple InterSystems IRIS instances on the same system and URL prefixes, see the Connecting to Remote Servers topic in *System Administration Guide*.

2.4 For Interoperability Productions: Enable Collection of Metrics

Beginning with InterSystems IRIS version 2021.1, instances can also record metrics about active interoperability productions and make them available to SAM. The recording of these interoperability metrics is disabled by default, and must be enabled for each interoperability production you want to monitor.

Refer to Interoperability Metrics in Monitoring InterSystems IRIS Using REST API for detailed instructions for enabling these metrics, as well as descriptions of the metrics which are available.

3 Download and Extract the SAM image

You can obtain distribution files for System Alerting and Monitoring (SAM) in the following locations:

- The **Components** section of the WRC software distribution download page (type `SAM` in the **Name** column to show only the available SAM kits)
- The SAM GitHub Repository, which hosts the latest available version

If you obtain the distribution files as a tarball, use the following command to uncompress it while preserving permissions:

```
tar zpxvf sam-<version-number>-unix.tar.gz
```

Replace <*version-number*> with the version of the SAM tarball you have.

The contents of this image define the container configuration necessary for SAM. They include:

readme.txt	A brief text document for getting started.
docker-compose.yml	A file that defines and configures the SAM components. Docker Compose then uses this file to deploy SAM. This reduces setup work; all of the initial configuration steps described in the section which follows involve editing this file.
config/	A directory that contains settings for the SAM application components.
start.sh, stop.sh	Scripts to facilitate starting and stopping SAM.

4 Install and Configure a Containerization Engine

System Alerting and Monitoring (SAM) is a network of components which run simultaneously, each in its own container. The SAM image includes a docker-compose.yml file which provides instructions for how these containers should be configured. This file is written according to the Compose specification. To run SAM, you must first set up a container engine capable of interpreting these instructions.

Currently, SAM supports the following container engines:

- Docker Compose
- Podman Compose

This section provides instructions for configuring your chosen container engine to run SAM.

Note: Due to the way containers are implemented in Windows and macOS versions of container engines compatible with SAM, InterSystems does not currently support containerized deployments of its products on Windows or macOS at this time. (See here for details.) All instructions for installing, configuring, and using SAM assume that you are deploying SAM on a Linux system.

4.1 Docker Compose

For instructions on installing Docker Compose, see the installation guide provided in the Docker documentation. The following versions are required:

- Docker Engine version 19.03.098 or higher
- Docker Compose version 1.25 or higher

Docker Compose needs no further configuration in order to run SAM. However, after you download and extract a SAM image, you may still need to configure the docker-compose.yml file according to the needs of your system.

4.2 Podman Compose

Install Podman using the package manager for your distribution according to the instructions provided in the Podman documentation. For instructions on installing Podman Compose, see the GitHub page for the project. InterSystems recommends using Podman version 3.4.0 or higher and the most recent stable version of Podman Compose.

The following additional steps are necessary to run SAM using Podman Compose:

1. If you are running SAM on a system secured with SELinux, SAM components require elevated permissions. (You can check whether this is the case by issuing the `sestatus` command from the command line.) If this is the case, include the following line for each service in the docker-compose.yml file:

   ```
   privileged: true
   ```

   Note: The docker-compose.yml file for images of SAM version 2.0 or higher includes this line as a comment everywhere it is required; you only need to remove the # character to complete this configuration step.

2. If the file system for your Linux distribution does not feature both an /etc/timezone file and an /etc/localtime symbolic link, you must map paths to the equivalent locations on your system in the volume configuration for the `iris` container. To do so:

   a. In the `iris` section of the docker-compose.yml file, locate the following lines:

   ```
   - /etc/timezone:/etc/timezone:ro
   - /etc/localtime:/etc/localtime:ro
   ```

   b. As needed, replace the first path in one or both lines (the path which does not feature the `:ro` option) to the equivalent path on your system.

3. To allow SAM to alias IP addresses, you must ensure that Podman is using one of the following network stacks:

   - Netavark
   - Container Networking Interface (CNI) with the dnsname plug-in

   If you configure one of these network stacks before you start SAM for the first time, no further configuration is necessary. If you attempt to start SAM using CNI without installing the dnsname plug-in first, startup will fail and you must reconfigure the SAM container network manually by completing the following steps:

   a. Stop all SAM containers.

   b. Install the dnsname plug-in for CNI. The `podman-plugins` package contains dnsname. On Red Hat Enterprise Linux systems, you can install this package by issuing the following command from the command line:

   ```
   sudo yum install podman-plugins
   ```

   c. Locate the configuration file for the SAM container network. By default, this file is saved to /etc/cni/net.d/sam_default.conflist.

   d. Using a text editor, edit the SAM network configuration file so that it includes the following block:

   ```
   {
        "type": "dnsname",
        "domainName": "dns.podman",
        "capabilities": {
             "aliases": true
        }
   }
   ```

   e. Delete the misconfigured SAM containers and the `sam_default` network by issuing the following command:

   ```
   podman rm -a; podman network rm sam_default
   ```

   f. Restart SAM to rebuild the SAM containers and the `sam_default` network using the correct configuration.

> **Note:** If you decide to switch to Netavark instead of adding the dnsname plug-in to CNI, it is not necessary to edit the sam_default.conflist network configuration file as described in the preceding instructions. Instead, you must edit the Podman containers.conf file (described on this Github page) to specify the new `network_backend`. Then, delete the old sam_default.conflist network configuration file as well as the SAM containers and the `sam_default` network. When you restart SAM, the startup procedure will automatically generate a new sam_default.conflist file configured for the new network stack.

4. If you would like to run the ./start.sh and ./stop.sh scripts to start and stop SAM (instead of invoking the `podman` commands directly), edit these scripts in a text editor as follows:

   - In the start.sh script:

     a. Change the `docker-compose` command to an equivalent `podman-compose` command, such as `podman-compose -p sam up -d`

     b. On a new line *before* the `podman-compose` command, add the following command:

     ```
     podman unshare chown 51773:51773 <instanceDir>
     ```

     Where <*instanceDir*> is the directory which will contain your “durable” storage directory. This gives SAM containers the appropriate user permissions to access the directory when they run.

     > **Note:** The file system entities associated with the InterSystems IRIS container are associated with the user account **`irisowner`** (UID 51773). Therefore **`irisowner`** user must have full permissions to the locations of these entities within the file system, including the durable storage directory. For more information, refer to Ownership and Directories in Running InterSystems Products in Containers.

   - In the stop.sh script:

     a. Change the `docker-compose` command to an equivalent `podman-compose` command. Because the specific options supported by `podman-compose` may change, this guide will use the simplified command `podman-compose -p sam down`

5 Initial SAM Configuration

You may need to change the default settings specified in the docker-compose.yml file in order to run System Alerting and Monitoring (SAM) on your system. To do so, simply open the file in a text editor and make the changes described in the following sections.

5.1 Set Up Your License Key

SAM comes with a free, built-in Community Edition license with the capacity to monitor approximately 40 instances. If you are using the Community Edition license, you can skip this section.

Note: The Community Edition license for SAM limits its container to using eight cores.

To specify a different SAM license:

1. Locate the `iris` service in the contents of the docker-compose.yml file.
2. Add a new `command` directly beneath the `image` line that specifies the desired license key to use.

For example, with a key named iris.key located in the SAM /config subdirectory, add:

```
[...]
image:intersystems/SAM:1.0.0.100
command: ["--key","/config/iris.key"]
init:true
[...]
```

5.2 For Docker Version 20.10.14 and up: Disable Capability Checker

Beginning with Docker Engine version 20.10.14, Docker grants containers a set of Linux capabilities incompatible with the Linux capability checker which the InterSystems IRIS containers for SAM 1.0 and 1.1 run on startup. For SAM 1.0 and 1.1 deployments using one of these later versions of Docker, the `iris` container will fail to start unless the InterSystems IRIS Linux capability checker is disabled.

To disable the capability checker:

1. Locate the `iris` service in the contents of the docker-compose.yml file.
2. Add the following line directly beneath the `image` line:

```
command: --check-caps false
```

Note: Beginning with SAM version 2.0, the default docker-compose.yml file includes the required line (`command: --check-caps false`) by default. For SAM 2.0 and all versions after, there is no need to add it yourself.

5.3 Configure the System Alerting and Monitoring Port

By default, SAM deploys on port 8080 of the host system. On Linux machines, you can check whether the port 8080 is in use by using the **netcat** command:

```
$ nc -zv localhost 8080
Connection to localhost 8080 port [tcp/http-alt] succeeded!
```

If the connection succeeds (like in the preceding example), then an application on your host system is already using port 8080 to listen for traffic. You may also need to configure your firewall to allow traffic on this port.

If necessary, you can change the host port mapping in the `nginx` section of the docker-compose.yml file. To do so:

1. Locate the `nginx` service in the contents of the docker-compose.yml file.
2. In the `ports` section, enter the desired port on your host machine. For example, if you would like to access SAM on port 9999, edit the section to look like:

```
[...]
ports:
   - 9999:8080
[...]
```

For more information, see the “ports” section of the Docker *Compose File Reference* (https://docs.docker.com/compose/compose-file/#ports).

5.4 Change the Default Location for Storing Your Monitoring Data

Although containerized, SAM maintains a persistent copy of the data you collect about your IRIS instances by extending the durable %SYS feature of InterSystems IRIS. By default, SAM uses a Docker volume which stores data in the

/var/lib/docker/volumes/ directory. However, this may be an inadequate storage location, especially if you are monitoring many instances or if this /var/ directory is located on a small partition.

InterSystems recommends implementing a higher volume storage solution for your monitoring data by configuring the location of the volume mounted for the SAM Manager, which corresponds to the `iris` service defined in the docker-compose.yml file. To do so:

1. Locate the `iris` service in the docker-compose.yml file.
2. Locate the `volumes` section. In its default configuration, this section looks like this:

   ```
   [...]
   volumes:
     - irisdata:/dur
     - ./config:/config
   [...]
   ```

3. Change the line which reads `- irisdata:/dur` so that the /dur directory in the IRIS container maps onto the desired location on your host system, instead of the named volume irisdata. To specify a directory elsewhere in the filesystem, replace `irisdata` with the path to the desired directory.

   Docker volumes can be configured to allow for a variety of storage solutions. For more information, refer to the Docker page about volumes and the “volumes” section of the Docker *Compose File Reference* page.

6 Start SAM for the First Time

You are now ready to use System Alerting and Monitoring (SAM). To start SAM:

1. Using the `cd` command in the command line, navigate to the directory containing the SAM docker-compose.yml file, which was acquired during initial setup.
2. - If you are either using Docker Compose or you are using Podman Compose and you have edited the `start.sh` script as specified in the Podman Compose configuration section:
     a. Run the `start.sh` script which InterSystems includes with the SAM image:

        ```
        ./start.sh
        ```

   - If you are using Podman Compose but you have not edited the script files:
     a. Before starting the SAM containers, issue the following command to allow SAM containers to write to the host directory which contains the durable storage location :

        ```
        podman unshare chown 51773:51773 <instanceDir>
        ```

        where <*instanceDir*> is the file system location which contains the SAM durable storage directory.

     b. Now, issue the following command (or a similar `podman-compose` command) to start the SAM containers in detached mode, with the project name `sam`:

        ```
        podman-compose -p sam up -d
        ```

3. Optionally, you can use the `docker ps` or `podman ps` command to confirm that all the containers are running. The output should look similar to the following:

```
$ docker ps
CONTAINER ID   IMAGE                       COMMAND                  CREATED             STATUS
                    PORTS                                                  NAMES
2aaa06f06a9c   nginx:1.17.9-alpine         "nginx -g 'daemon of..." About an hour ago   Up About
an hour             80/tcp, 0.0.0.0:8080->8080/tcp                         sam_nginx_1
0e2b30fcb376   grafana/grafana:6.7.1       "/run.sh"                About an hour ago   Up About
an hour             3000/tcp                                               sam_grafana_1
d2c825f9d220   prom/alertmanager:v0.20.0   "/bin/alertmanager -..." About an hour ago   Up About
an hour             9093/tcp                                               sam_alertmanager_1
4851893bc369   prom/prometheus:v2.17.1     "/bin/prometheus --w..." About an hour ago   Up About
an hour             9090/tcp                                               sam_prometheus_1
61120be391df   intersystems/sam:1.0.0.83   "/iris-main"             About an hour ago   Up About
an hour (healthy)   2188/tcp, 1973/tcp, 1974/tcp                           sam_iris_1
```

Note: For SAM version 1.1 and versions after, the SAM image also includes an `iris-init` container, which runs an initialization service briefly at startup and then stops.

7 Next Steps: Using SAM

To learn how to monitor your system with System Alerting and Monitoring (SAM), refer to the guide which corresponds to the version you have deployed:

- *Monitor Your InterSystems IRIS Instances with SAM (Version 1.0/1.1)*
- *Monitor Your InterSystems IRIS Instances with SAM (Version 2.0)*