Monitor Your InterSystems IRIS Instances with SAM (Version 2.0)

2024-05-06

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

Monitor Your InterSystems IRIS Instances with SAM (Version 2.0)
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

Start Out with System Alerting and Monitoring

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 platform for monitoring InterSystems products.

This page and the pages which follow provide a guide for using SAM version 2.0 to monitor your system. They assume that you have already completed the initial setup.

If you are using SAM version 1.0 or 1.1, please refer to the guide for those versions instead.

Note: You can also interface with SAM using its REST API.

1.1 Start and Stop SAM

InterSystems provides two scripts that make it easy start or stop System Alerting and Monitoring.

1.1.1 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 suggested in the setup instructions:

     a. Run the `start.sh` script which InterSystems includes with the SAM image:

     ```
     ./start.sh
     ```

     This runs a `docker-compose` command to start SAM.

   - 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 to write to the host directory which you have specified as the durable storage location:

```
podman unshare chown 51773:51773 [hostDir]
```

where *[hostDir]* is the host file system location being mounted for SAM durable storage.

b. Now, issue the following command to start the SAM containers:

```
podman-compose up &
```

3. Optionally, you can use the `docker 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, the SAM image also includes an `iris-init` container, which runs an initialization service briefly at startup and then stops.

1.1.2 To Stop SAM

1. Using the `cd` command in the command line, navigate to the directory containing the SAM docker-compose.yml file.
2. • If you are using Docker Compose, run the `stop.sh` script which InterSystems includes with the SAM image:

```
./stop.sh
```

This runs a `docker-compose` command to stop SAM.

• If you are using Podman Compose, issue the following command to stop all SAM containers:

```
podman-compose down
```

3. Optionally, you can use the `docker ps` or `podman ps` command to confirm that all the containers have stopped. Use the `-a` flag to view all containers, even those that are not running:

```
docker ps -a
```

1.2 Access the SAM Web Portal

When System Alerting and Monitoring is running, you can access it from a web browser at the following address:

```
http://<sam-domain-name>:<port>/api/sam/app/index.csp
```

where <*sam-domain-name*> is the DNS name or IP address of the system SAM is running on, and <*port*> is the configured Nginx port (8080 by default). You may want to bookmark this address.

When accessing SAM, you must log in using a valid **User Name** and **Password**. Like InterSystems IRIS, SAM includes several predefined accounts with the default password SYS. Choose any of these accounts with login permissions (such as *Admin* or *SuperUser*) and log in using the default password SYS.

Note: The first time you sign in with one of the predefined accounts, SAM prompts you to enter a new password. To secure the SAM application, be sure to set a new password for all the predefined accounts. For a list of all the predefined accounts, see Predefined User Accounts in the "Users" chapter of the *Security Administration Guide*.

1.2.1 First Time: Get Started with the Welcome Page

The first time you sign in to the SAM web application, SAM displays a **Welcome** page, where you can get started by selecting one of the following actions:

- **Create Your First Cluster** — opens the **Add New Cluster** dialog
- **Open Existing Configuration** — allows you to import an existing SAM configuration

Upon subsequent logins, SAM displays the **Monitor Clusters** page, which provides a system-wide performance overview.

2

Manage Clusters, Instances, and Alert Rules

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.

To monitor InterSystems IRIS instances with System Alerting and Monitoring (SAM) version 2.0, you must first create at least one SAM *cluster* (that is, a group of instances) and then define each instance as part of a cluster. You can then set alerts for a cluster to receive a notification if a metric for any instances in the cluster cross a given threshold. This page describes how to manage clusters, instances, and alert rules using the SAM web portal; you can also manage clusters, instances, and alert rules using the SAM REST API.

Note: Clusters are sets of instances that do not intersect. In other words, an instance can only be assigned to one cluster. Attempting to add the same instance to a second cluster yields an error.

2.1 Organize Clusters for Related Instances

Grouping instances into SAM clusters allows you to monitor your system with greater precision. If your system contains instances for which you have different performance expectations, organize them into different clusters so that you can apply different sets of alert rules to them. For example, you may organize instances running on servers configured to handle high volumes of traffic into a cluster named `highTraffic` and other instances into a cluster named `lowTraffic`. You can then set an alert for when the number of SQL queries a `lowTraffic` instance receives per second exceeds a certain threshold, but set a higher threshold for alerts on instances in the `highTraffic` cluster.

2.1.1 Add or Edit a Cluster

To add a SAM cluster, do the following:

1. Navigate to the main (**Monitor Clusters**) page in the SAM web application. Selecting **System Alerting and Monitoring** from anywhere within the application returns you to the main page.
2. Select **+ New Cluster** to open the **Add New Cluster** dialog.
3. Fill in the following fields:

- **Cluster Name** — A unique identifier for the cluster. SAM ignores case when it evaluates cluster names. If you attempt to assign a name to a cluster which differs from the name of an existing cluster in case alone (for example, *transactarchive* and *transactArchive*), you will receive an error.

- **Description** — An optional description for the cluster.

4. Select **Add Cluster**. This opens the **Edit Cluster** dialog, where you can continue to define your new cluster by adding instances and setting alert rules.

   Note: You can edit or delete an existing SAM cluster by navigating to the **Monitor Clusters** page and selecting the edit icon on the card for that cluster.

   When you change the **Cluster Name** for an existing cluster, the old instance metrics retained by the Prometheus database from before the name change remain associated with the previous cluster name. It has no effect on the availability of this data to queries using instance names.

2.2 Define the Instances You Want to Monitor

System Alerting and Monitoring can collect metrics and alerts from any InterSystems IRIS instance version 2020.1 or higher. To add an instance to SAM, first prepare the instance for monitoring (as described in the initial set up instructions) and then add it to a SAM cluster.

Note: There is no specific limit to the number of instances SAM can monitor, but this number is constrained by the available memory in the SAM database. The maximum database size for the SAM Community Edition is 10GB, which is enough to support monitoring of about 40 instances with the default settings.

You can monitor the SAM Manager to ensure the SAM database does not run out of space. If you need to monitor more instances from SAM, consider using a different license.

2.2.1 Add or Edit an Instance within a Cluster

To add or edit an InterSystems IRIS instance within a SAM cluster, do the following:

1. Ensure that you have prepared the instance for monitoring by SAM.
2. Navigate to the main (**Monitor Clusters**) page in the SAM web application. Clicking **System Alerting & Monitoring** from anywhere within the application returns you to this page.
3. Select the cluster to which the instance belongs (or will belong). If there are no clusters, you must create one.
4. Click **Edit Cluster** to open the **Edit Cluster** dialog.
5. Click the **+ Add Instance** button at the top of the **Instances** table.

   Note: To edit or delete an existing instance, select it from the **Instances** table.

6. Fill in the following fields:

   - **Host Name** – The fully qualified domain name or IP address of the machine hosting the target InterSystems IRIS instance.

     InterSystems recommends using domain names whenever possible, as IP addresses may change.

Note: If the instance you are monitoring is located on the same system as SAM, you may enter `host.docker.internal` in this field.

- **Web Server Port** – The web server port of the target InterSystems IRIS instance.
- **Dashboard** — The Grafana dashboard you wish to display on the single instance page for this instance. SAM includes one pre-configured dashboard (**SAM Dashboard**); you can create others within Grafana.
- **Instance name** and **Description** – Optional text descriptors to help you identify the instance.
- **URL Prefix** — An URL prefix which identifies an instance in a system where one web server serves multiple instances. URL prefixes must include a slash at the beginning, but not at the end.

7. Click **Save** to begin monitoring the instance with SAM.

2.3 Set Rules for when to Receive Alerts

System Alerting and Monitoring automatically collects InterSystems IRIS alerts from the instances it monitors. If you want to specify additional events that generate alerts, you can do so by defining Prometheus *alert rules*.

An alert displays information about the instance that generated it; the time the alert fired; and the alert name, message, and severity. A Prometheus alert rule indicates when SAM should fire an alert by evaluating an *alert expression* written in Prometheus Query Language (PromQL) using data from your instances collected in real time.

Alert rules are defined at the cluster level, but are evaluated distinctly for each instance within the cluster. This means instances within a cluster share the same alert rules, but generate alerts individually. By sorting your instances into clusters based on your performance expectations for them and setting different alert rules for each cluster, you can calibrate your monitoring strategy more precisely.

Note: By default, alerts display in the SAM web portal, in **Alerts** tables. You can specify additional actions for System Alerting and Monitoring to perform when an alert fires, such as sending a text or email. To do so, refer to the instructions for writing and importing custom alert handlers in a later section on this page.

2.3.1 Create an Alert Rule for a Cluster

To create a new alert rule for a cluster, do the following:

1. Navigate to the main (**Monitor Clusters**) page in the SAM web application. Clicking **System Alerting & Monitoring** from anywhere within the application navigates to this page.
2. Select the cluster for which you would like to create an alert rule. If there are no clusters, you must create one.
3. Click **Edit Cluster** to open the **Edit Cluster** dialog.
4. Click the **+ Add Alert Rule** button at the top of the **Alert Rules** table.

   Note: You may select an existing alert rule from this table to edit or delete it.

5. Fill in the following fields:
   - **Name** – Any name for the alert rule. It is often useful to include the metric the rule uses in the name.
   - **Severity** – Either **Critical** or **Warning**. The severity of the alert determines the impact it will have on the instance state (described in a later section on this page)

Note: You can give multiple alert rules the same name, but different severities. If both rules fire at the same time, SAM suppresses the rule with lower severity. This behavior reduces duplicate alerts firing for the same event.

- **Expression** – An expression that defines when the alert fires, written in Prometheus Query Language.

  The alert expression syntax section contains an overview of the Prometheus Query Language syntax and several examples.

- **Message** – A text description of the alert rule, which SAM displays when the alert fires.

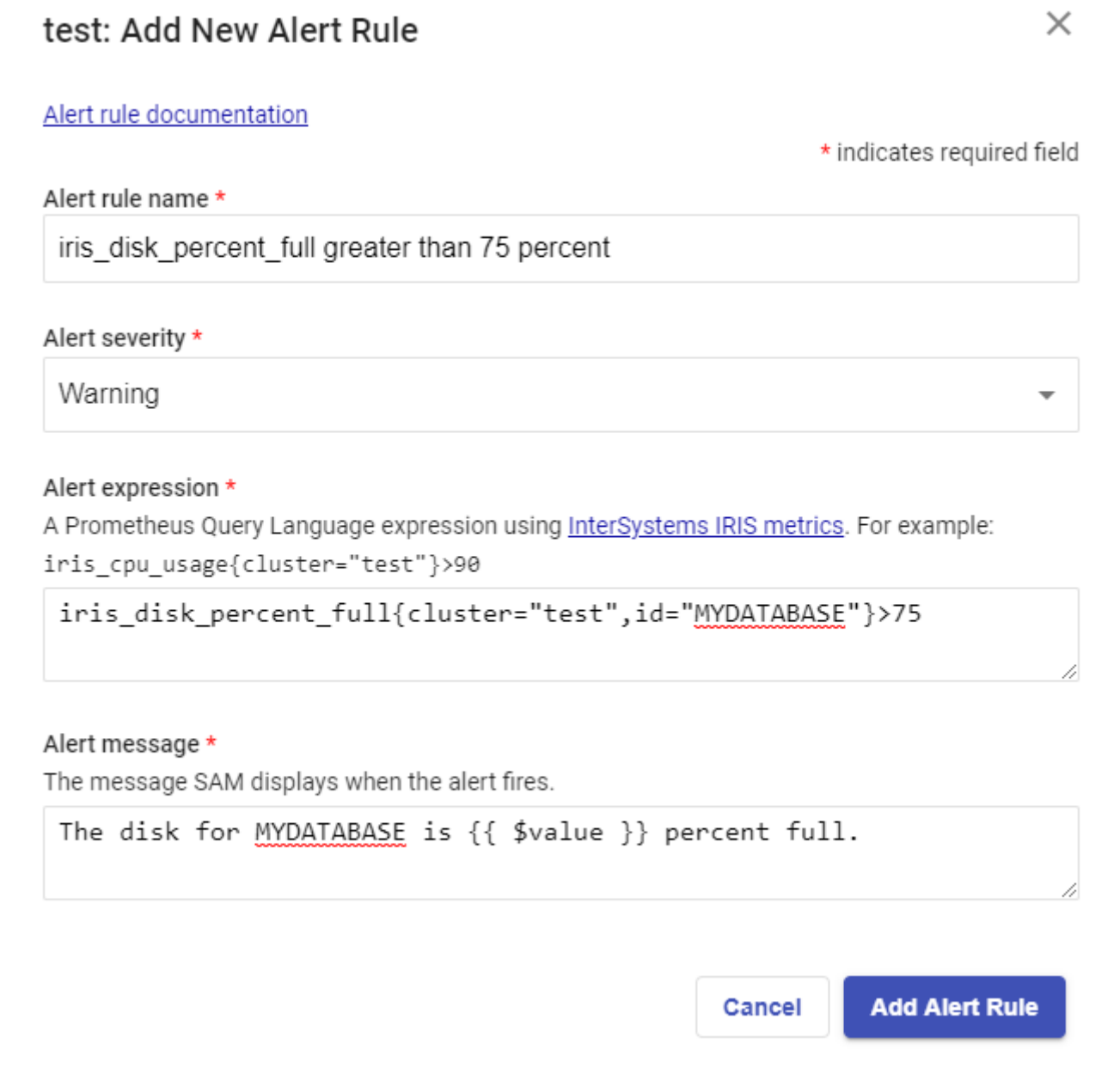
  Note: The **Alert message** supports the `$value` variable, which contains the evaluated value of an alert expression. The syntax is:

  ```
  {{ $value }}
  ```

  The `$value` variable only holds one value; as such, you should not use it for alert rules that evaluate to multiple values (such as a rule that uses the `and` operator).

6. Click **Save**. SAM validates the alert expression and then adds the alert rule to the cluster.

Below is an example of an alert rule:



2.3.2 Basic Syntax for Alert Expressions

This section provides an overview of how to write alert expressions and some examples.

Note: If you want to learn how to write advanced alert expressions, read about the full capabilities of PromQL on the "Querying Prometheus" page in the Prometheus Documentation (https://prometheus.io/docs/prometheus/latest/querying/basics/).

A simple alert expression compares a metric to a value. For example:

```
# Greater than 80 percent of InterSystems IRIS licenses are in use:
iris_license_percent_used{cluster="production"}>80

# There are less than 5 active InterSystems IRIS processes:
iris_process_count{cluster="test"}<5

# The disk storing the MYDATA database is over 75% full:
iris_disk_percent_full{cluster="test",id="MYDATA"}>75

# Same as above, but specifying directory instead of database name:
iris_disk_percent_full{cluster="production",dir="/IRIS/mgr/MYDATA"}>75
```

The basic format for an alert expression based on a single metric is:

```
metric_name{cluster="cluster_name",label(s)}>value
```

metric_name	The metric that the alert rule uses. See Metric Descriptions in the "Monitoring InterSystems IRIS Using REST API" section of *Monitoring Guide* for a table of all the default metrics you can use.
cluster_name	The cluster to which the alert rule applies. SAM applies the alert rule to all instances in the specified cluster. If any instance in the cluster triggers the rule, SAM generates an alert for that instance.
additional_labels	If a metric contains labels, you may include these after the `cluster` label. Multiple labels are separated by commas. All metrics must include the `cluster` label, described above.
operator	The following comparison operators are available: • > (greater than) or >= (greater than or equal to) • < (less than) or <= (less than or equal to) • == (equal to) or `!=` (not equal to)
value	The value can be positive or negative, and may include a decimal component.

2.3.3 Alert Examples

The examples which follow demonstrate some of the capabilities of PromQL.

2.3.3.1 Example 1: Basic Alert Rule

The simplest alert expressions directly compare a single metric to a value.

The following alert expression evaluates `iris_cpu_usage`, which measures the total percent of CPU in use on the machine running InterSystems IRIS. If the value of `iris_cpu_usage` exceeds `90` for any InterSystems IRIS instance in the **test** cluster, the alert fires.

```
iris_cpu_usage{cluster="test"}>90
```

2.3.3.2 Example 2: Arithmetic Operators

PromQL supports the following arithmetic operators, ordered by precedence:

1. `^` (exponentiation)
2. `*` (multiplication), `/` (division), `%` (modulo)
3. `+` (addition), `-` (subtraction)

Arithmetic operators are particularly useful when writing an alert expression that contains two or more metrics.

The following expression is triggered when the **USER** database in the **test** cluster is greater than 90 percent full. The expression calculates the percent by dividing the database size (`iris_db_size_mb`) by the database maximum size (`iris_db_max_size_mb`).

```
(iris_db_size_mb{cluster="test",id="USER"}/iris_db_max_size_mb{cluster="test",id="USER"})*100>90
```

2.3.3.3 Example 3: Logical OR Operator

PromQL supports logical operators for writing more complex rules. When using the `or` operator, the expression evaluates two conditions and fires if *either* is true.

One use for the `or` operator is to check whether a metric falls outside of a certain range. The following alert expression is triggered when either of the following conditions is true:

- There are greater than 20 active ECP connections in the **production** cluster.
- There is less than one active ECP connection in the **production** cluster.

```
iris_ecp_conn{cluster="production"}<1 or iris_ecp_conn{cluster="production"}>20
```

2.3.3.4 Example 4: Logical AND Operator

PromQL also supports the `and` operator. When using the `and` operator, the expression evaluates two conditions and fires if *both* are true.

The following example shows an alert rule that fires when both conditions are true:

- There are unread alerts in the **test** cluster.
- The system health state of an instance in the **test** cluster is something other than `0`.

```
iris_system_alerts_new{cluster="test"}>=1 and iris_system_monitor_health_state{cluster="test"}!=0
```

3

Inspect System Performance Using the SAM Web Portal

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.

Once you have defined your clusters, instances, and alert rules within System Alerting and Monitoring (SAM) version 2.0, you can use it to see real-time metrics and alerts for your InterSystems IRIS instances. The SAM web portal consists of multiple pages that display this information at different levels of detail.

These pages are:

- **Monitor Clusters** Page – the "home page" for the SAM web portal, which displays an overview of all clusters.
- Single Cluster Page – a more focused view, which displays only the information for instances in a single cluster.
- Single Instance Page – the narrowest and most detailed view, which displays the instance's details, alerts, and metrics dashboard.

These pages allow you to:

- View the **Alerts** Table
- Inspect SAM Metrics
- Understand Instance State

3.1 Monitor Clusters Page

The **Monitor Clusters** page displays an overview of all your clusters. The **Monitor Clusters** page is the home page for the SAM web application, and you can return to it at any time by selecting the InterSystems logo (beside the **System Alerting & Monitoring** title) at the top of any other page in the application.

The **Monitor Clusters** page displays a cluster card for each cluster in your system. This card features a circle depicting the state of all the instances for the cluster. The **Monitor Clusters** page also includes an **Alerts** table, showing the recent alerts from all monitored instances, and provides access to the configuration settings.

To see detailed information about a specific cluster, click on the cluster card or the cluster name in the **Alerts** table. To see detailed information about a specific instance, click on the **IP:Port** for the instance in the **Alerts** table or .

3.1.1 Settings Menu

The main System Alerting and Monitoring page contains a gear icon, located near the top of the screen. Click this icon to access the **Settings** drop-down menu, which contains the following options:

- **Settings** — opens the **Configuration Settings** dialog, where you can:
  - Set the number of days (between 1 and 30) for SAM to store alert and metric data in an IRIS database for long-term analysis
  - **Use HTTPS** to connect to your InterSystems IRIS instances, if you have configured SAM and your target instances for SSL/TLS.
- **Save Current Configuration** — exports data about your clusters, instances, and alerts as a JSON file. Use this file to migrate your system configuration into another SAM instance.
- **Open Existing Configuration** — opens a File Upload window, where you can import a SAM system configuration JSON file.

  CAUTION: Importing a SAM system configuration erases the current configuration, and cannot be undone.

The section "Tune and Troubleshoot SAM" describes procedures for changing other SAM settings, including how to increase the retention time for data in the Prometheus database, which can help you improve performance if you are constantly querying data more than two hours old.

3.2 Single Cluster Page

To view details about a cluster, click on the cluster card on the **Monitor Clusters** page.

The single cluster page displays an Alerts table, showing the recent alerts from all instances in that cluster. There is also an **Instances** table with details about the target instances. The **Instances** table shows the following details:

- **IP:Port** – The IP address and Port which specify where a target instance is located. You can click this to "zoom in" to the **Instance** page.
- **State** – The state of the instance, which can be **OK**, **Warning**, **Critical**, or **Unresponsive**. See "Understand Instance State" below for a description of how SAM determines instance state.
- **Name** – The name of the instance.
- **Description** – The description of the instance.

3.3 Single Instance Page

To see the page for a single instance, click on the **IP:Port** for the instance. The single instance page contains the following sections:

- A **Details** table, which contains the instance's **IP:Port**, **State**, **Name**, **Description**, and a link to the Management Portal. For details about how SAM calculates **State**, see the "Understand Instance State" section below.

- An **Alerts** table, showing the recent alerts for the current instance.
- A **Dashboard**, which shows an overview of the current Grafana Dashboard for the instance.

The page also has an **Edit Instance** button, which allows you to modify some of the instance details, and **Delete Instance** button, which allows you to remove the instance from SAM.

Note: If you edit an instance and change its network address, SAM purges all existing alerts tied to that instance. This is because SAM assumes different network address refer to different instances.

3.3.1 Grafana Dashboards

A **Dashboard** displays several graphs of metrics, providing a snapshot of recent activity on the instance.

The table below describes the information reported by the pre-configured **SAM Dashboard**:

Dashboard Graph	Metric(s) used	Description
CPU Utilization	`iris_cpu_usage`	The CPU usage of the system running the instance for the past 30 minutes.
Glorefs	`iris_glo_ref_per_sec` `iris_glo_ref_rem_per_sec`	The global references to local (blue line) and remote (orange line) databases for the past 30 minutes
Global Updates	`iris_glo_update_per_sec`	Updates to globals located on local databases per second for the past 30 minutes
IRIS Disk Percent	`iris_disk_percent_full`	Percent of used space on the storage volume for the IRISSYS database
IRIS Disk Remaining	`iris_directory_space`	Free space available on the storage volume for the IRISSYS database
Database Reads	`iris_phys_reads_per_sec`	Physical database block reads from disk per second for the past 30 minutes
IRIS Database Latency	`iris_db_latency`	Milliseconds to complete a random read from the database for the past 30 minutes
Total Pri Jnl Size	`iris_jrn_size{id="primary"}`	Current size of the primary journal file
Pri Jnl Free	`iris_jrn_free_space{id="primary"}`	Free space available on the primary journal directory's storage volume
WIJ Free	`iris_jrn_free_space{id="WIJ"}`	Free space available on the WIJ journal directory's storage volume
License Current Pct	`iris_license_percent_used`	Percent of licenses currently in use

Dashboard Graph	Metric(s) used	Description
Licenses Available	`iris_license_available`	Number of licenses currently not in use
System Alerts	`iris_system_alerts`	The number of alerts posted to the messages log since system startup

You can create and edit SAM dashboards using Grafana, an open-sourced metrics visualization tool. Select **View in Grafana** to open Grafana. When you save a new dashboard in Grafana, you can set it as the default **Dashboard** to display for an instance by selecting its name from a drop-down menu in the **New Instance** or **Edit Instance** dialog for that instance.

For more information about creating and editing dashboards, check out the Grafana documentation (https://grafana.com/docs/guides/getting_started/).

Note: If you edit the dashboard to display metrics older than two hours, you may want to increase the Prometheus database retention time.

To add or edit an instance definition using the REST API in this version of SAM, you must specify the dashboard you wish to display for the instance by setting the `dashboardId` property. The value of `dashboardId` is the `uid` which Grafana assigns the dashboard when it is saved. You can find the value of `uid` by examining the JSON model for the dashboard in Grafana.

3.4 Review the Alerts Table

Each page SAM provides for inspecting your system (the **Monitor Clusters** page, the **Cluster** page, and the **Instance** page) provides an **Alerts** table listing alerts recorded for instances within the scope of that page. By default, this table displays alerts from the last hour; to view all alerts, select **Show All**.

An **Alerts** table contains the following information:

- **Last Reported** – The most recent time the alert was reported.
- **Cluster** – The cluster containing the instance that generated the alert.
- **IP:Port** – The IP address and Port of the instance that generated the alert.
- **Severity** – The severity of the alert: either **Critical** or **Warning**.
- **Source** – The source that generated the alert: either **IRIS** or **Prometheus**.
  - An **IRIS** alert is generated by an InterSystems IRIS instance. The instance's log monitor scans the messages log and posts notifications with severity 2 or higher to the alerts log, where SAM collects them. For more information, see the *Monitoring Guide*.
  - A **Prometheus** alert is generated by SAM according to user-defined alert rules. For more information, refer to the previous section on alerts.
- **Name** – The name of the alert.
- **Message** – The message associated with the alert.

3.5 Inspect Instance Metrics

All InterSystems IRIS instances collect metrics that describe the status and operation of the instance. System Alerting and Monitoring allows you to monitor those metrics over time, and use them to configure alert rules.

For a list of all these metrics, see Metrics Description in the "Monitoring InterSystems IRIS Using REST API" section of the *Monitoring Guide* that corresponds to your version of InterSystems IRIS. The Create Application Metrics section on the same page describes how to create your own metrics.

3.6 Understand Instance State

Instance state indicates whether an InterSystems IRIS instance has fired any alerts recently. There are four possible values for instance state: **OK**, **Warning**, **Critical**, or **Unreachable**. A state of **OK** means there have been no recent alerts. When an alert fires for an instance, System Alerting and Monitoring elevates that instance's state to **Warning** or **Critical**. **Unreachable** means that, for some reason, SAM cannot access the instance.

Note: A state of **OK** does not necessarily mean there are no problems with an instance. Likewise, you may determine that no action is required for an instance with a **Critical** state. The instance state reflects the number of recent alerts, but does not provide comprehensive information about the instance.

Instance state is a combination of two factors: the InterSystems IRIS instance's System Health State (which SAM obtains from the `iris_system_state` metric), and recent Prometheus alerts generated by the instance. For information about the System Health State, see System Monitor Health State in the "Using System Monitor" chapter of the *Monitoring Guide*. For more information about Prometheus alerts, see the Receive Alerts section above.

System Alerting and Monitoring determines instance state as follows:

- The state is **Critical** if either of the following is true:
  - A Prometheus alert with severity **Critical** fired within the past 30 minutes.
  - The System Monitor Health State is `2` or `-1`.
- Otherwise, the state is **Warning** if any of the following are true:
  - A Prometheus alert with severity **Critical** fired between 30 and 60 minutes ago.
  - A Prometheus alert with severity **Warning** fired within the past 30 minutes.
  - The System Monitor Health State is `1`.
- Finally, the state is **OK** if:
  - No Prometheus alerts have fired in the past hour.
  - The System Monitor Health State is `0`.
- **Unreachable** means SAM cannot access the instance. See the section below for guidance troubleshooting an unreachable instance.

4

Tune and Troubleshoot SAM

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.

This section describes how to perform common tasks to configure the performance of System Alerting and Monitoring (SAM) version 2.0, and how to troubleshoot common problems.

4.1 Access the Management Portal for the SAM Manager

The SAM Manager is the InterSystems IRIS instance that powers the System Alerting and Monitoring application. Like other InterSystems IRIS instances, the SAM Manager provides a web application for performing maintenance and administration called the Management Portal. You can access the Management Portal for the SAM at the following address:

```
http://<sam-domain-name>:<port>/csp/sys/UtilHome.csp
```

where <*sam-domain-name*> is the DNS name or IP address of the system SAM is running on, and <*port*> is the configured Nginx port (8080 by default).

Actions you can perform using the Management Portal for SAM include:

- Adjusting Startup Settings
- Clearing the SAM Database
- Monitoring the SAM Manager
- Setting Up an SSL/TLS Configuration (as described in a later section)
- Importing Alert Handlers (as described in a later section)

Important: The SAM Manager should not be used to develop or run any application; it is strictly for use by SAM. The directions in this section describe appropriate uses and interactions with the SAM Manager.

For a general purpose InterSystems IRIS instance, install InterSystems IRIS community edition.

4.1.1 Adjust Startup Settings

The SAM Manager initially allocates memory on startup as follows:

- 2,000 MB of 8KB blocks for the database cache
- 300 MB for the routines cache

This allocation should be sufficient when monitoring a modest number (30 or fewer) of InterSystems IRIS instances. If you are monitoring a large number of instances, or find that the SAM Manager is regularly using the full amount of allocated memory, you can increase these limits.

For details on adjusting these settings, see the Allocating Memory to the Database and Routine Caches topic in the *System Administration Guide*.

4.1.2 Clear the SAM Database

System Alerting and Monitoring Community Edition has a maximum database size limit of 10 GB. If this limit is met, SAM may exhibit unexpected behavior, and it becomes necessary to clear the database.

In the **SQL** page of the SAM Manager (**System Explorer** > **SQL**), enter the following command to delete all SAM metric data:

```
DELETE FROM %SAM.PrometheusSample
```

To prevent the SAM database from filling up again, consider using a difference license, changing the storage location, or lowering the number of days that SAM stores metrics using the configuration settings dialog on the **Monitor Clusters** page.

4.1.3 Monitor the SAM Manager

It is possible to use System Alerting and Monitoring to monitor the SAM Manager, as the SAM Manager is itself an InterSystems IRIS instance. This allows you to keep track of whether the SAM database is at risk of filling up, and make sure the configured cache sizes are sufficient for SAM operations.

Adding the SAM Manager to a SAM cluster is the same as adding any other InterSystems IRIS instance, with the following difference:

For the **IP** and **Port** fields, specify the fully qualified DNS name and port (8080 by default) where SAM runs. You can see these values in the address bar of your browser when accessing SAM. For example, if the URL for SAM is:

```
http://<sam-domain-name>:<port>/api/sam/app/index.csp
```

Specify `<sam-domain-name>` in the **IP** field, and `<port>` in the **Port** field.

Note: It does not work to specify `localhost` in the **IP** field; you must enter a fully qualified DNS name.

4.2 Configure SAM to use HTTPS

Beginning with version 2.0, SAM can monitor InterSystems IRIS instances over HTTPS, which is secured using SSL/TLS. Using this protocol, you can be certain that the information SAM receives about target instances is genuine.

Setting up an HTTPS connection between SAM and the instances it monitors involves the following steps:

1. Configure an SSL/TLS port for each instance you want to monitor.
2. Set up an `ISC_SAM` SSL/TLS configuration through the Management Portal for the SAM Manager. This allows the SAM Manager to fetch InterSystems IRIS alerts over an HTTPS connection.

3. Modify the isc_prometheus.yml file to use HTTPS. This secures the collection of instance metrics from each instance by Prometheus.
4. Update each instance in the SAM web application, specifying the SSL/TLS port you configured in step 1.
5. In the SAM web application, enable the use of HTTPS. This directs the SAM Manager to use your HTTPS configuration to fetch alerts. When HTTPS is enabled, the SAM web portal also provides HTTPS links to the Management Portal on the page for each instance.

This section describes each successive step in greater detail.

4.2.1 Configure SSL/TLS for the Target Instance

To monitor an instance over SSL/TLS, you must configure the web server associated with the instance to serve the instance's web applications (including /api/monitor) over an SSL/TLS connection. For container environments, a `webgateway` container provides the web server for an instance's web applications. You must configure the web server within the `webgateway` container to use SSL/TLS. Refer to the documentation for your web server for guidance.

4.2.2 Set up the `ISC_SAM` SSL/TLS Configuration

In order for SAM to obtain alerts from your target instances over HTTPS, you must configure SAM to authenticate the instances as an SSL/TLS client. To do so:

1. Place a copy of the certificate for the Certificate Authority which can authenticate your instances in the durable storage directory for your SAM `iris` container.
2. Access the Management Portal for the SAM Manager.
3. In the Management Portal, navigate to **System Administration** > **Security** > **SSL/TLS Configurations**.
4. On the **SSL/TLS Configurations** page, select **Create New Configuration**.
5. Fill out the **New SSL/TLS Configuration** form to specify the client-side SSL/TLS configuration appropriate for your deployment. The following parameters are required:
   - The **Configuration Name** must be `ISC_SAM`
   - The configuration **Type** must be **Client**
   - **Server certificate verification** must be required
   - The **File containing trusted Certificate Authority certificate(s)** must specify the path to the certificate file which was placed in the durable storage directory in step 1.
6. To test your configuration, select **Test**. As prompted, enter the server name and the SSL/TLS port number you have specified for one of your target instances. (Select **OK** after each prompt.)

   After the SAM Manager attempts to establish an SSL/TLS connection at the address specified, a message appears at the top of the **New SSL/TLS Configuration** page informs you of the result of its attempt.
7. Select **Save**.

4.2.3 Modify the isc_prometheus.yml File

SAM collects alerts using the SAM Manager, but it uses another component (Prometheus) to collect metrics. Therefore, to collect metrics over HTTPS you must also configure Prometheus to authenticate target instances as an SSL/TLS client.

Prometheus is configured by the isc_prometheus.yml file, which is located in the /config/prometheus/ directory within the SAM image. Modify the Prometheus configuration by performing the following steps:

1. Open the isc_prometheus.yml file in a text editor.
2. Locate the `scrape_configs` section
3. Change the value of the `scheme` parameter to `https`
4. Below the line which now reads `scheme: https`, add lines so that the start of the `scrape_configs` section reads as follows:

```
scrape_configs:
  - job_name: 'SAM'
    metrics_path: '/api/monitor/metrics'
    scheme: https
    tls_config:
      ca_file: [pathToCACert]
```

   where *[pathToCACert]* is the path to a copy of the certificate file for the Certificate Authority which can authenticate your target instances.

   Note: You may wish to configure Prometheus further to meet the needs of your SAM deployment. Refer to the Prometheus documentation for a detailed description of your configuration options.

5. Save your changes

4.2.4 Update the Instance Port in the SAM Web Application

Usually, a web server uses a different port number for HTTPS connections than it does for HTTP connections. In order for SAM to connect with a target instance over HTTPS, SAM must address the instance using the SSL/TLS port.

This means that for each instance you wish to monitor over HTTPS, you must edit the instance in the SAM web application so that it specifies the new combination SSL/TLS **Web Server Port**. Ensure that the other connection information is correct as well. For further instructions, see Add or Edit an Instance within a Cluster.

4.2.5 Enable HTTPS in the SAM Configuration Settings

Once you have completed all the preceding configuration steps, simply select **Use HTTPS** in the **Settings** menu for the SAM web application and then select **Save**. This directs the SAM Manager to use the SSL/TLS connection you have configured to fetch alerts over HTTPS, and to provide HTTPS links to the Management Portal for each instance on the single instances pages of the SAM web portal.

Congratulations! Your System Alerting and Monitoring deployment is now communicating with your system over HTTPS.

4.3 Create Custom Alert Handlers

You can create custom alert handlers that specify additional actions for System Alerting and Monitoring to perform when an alert fires, such as sending a text or email. Setting up an alert handler is a two step process:

1. Write the Alert Handler
2. Import the Alert Handler

4.3.1 Write the Alert Handler

To create an alert handler, you must create a class using an ObjectScript IDE. Connect this IDE to an InterSystems IRIS instance that is not part of SAM.

Important: You *cannot* use the SAM Manager to create the alert handler, as SAM is not a development platform.

Instead, you must connect the IDE to a different InterSystems IRIS instance (such as the InterSystems IRIS Community Edition), and later import the alert handler into the SAM Manager.

After setting up the IDE, create a class with the following characteristics:

- The class extends the %SAM.AbstractAlertsHandler class.
- The class implements the **HandleAlerts()** class method. Within this method, specify the desired behavior when an alert fires.

When SAM detects a new alert (or multiple new alerts), SAM calls the **HandleAlerts()** method of all alert handlers. The **HandleAlerts()** method receives a %DynamicArray packet of alerts with the following format:

```
[
  {
    "labels":{
      "alertname":"High CPU Usage",
      "cluster":"1",
      "instance":"10.0.0.24:9092",
      "job":"SAM",
      "severity":"critical"
    },
    "annotations":{
      "description":"CPU usage exceeded the 95% threshold."
    },
    "ts": "2020-04-17 18:07:42.536"
  },
  {
    "labels":{
      "alertname":"iris_system_alert",
      "cluster":"1",
      "instance":"10.0.0.24:9092",
      "job":"SAM",
      "severity":"critical"
    },
    "annotations":{
      "description":"Previous system shutdown was abnormal, system forced down or crashed"
    },
    "ts":"2020-04-17 18:07:36.926"
  }
]
```

Note: Alerts generated by an InterSystems IRIS instance are all named `iris_system_alert`.

Below is an example of an alert handler class. This example writes a message to the messages log (or Console Log) whenever an alert fires:

```
/// An example Alert Handler class, which writes messages to the messages log.
Class User.AlertHandler Extends %SAM.AbstractAlertsHandler
{

ClassMethod HandleAlerts(packet As %DynamicArray) As %Status
{
      set iter = packet.%GetIterator()
      while iter.%GetNext(.idx, .alert) {
            set msg = alert.annotations.description
            if alert.labels.severity = "critical" {set severity = 2} else {set severity = 1}
            do ##class(%SYS.System).WriteToConsoleLog(msg, 1, severity)
         }
      q $$$OK
}

}
```

4.3.2 Import the Alert Handler into SAM

After creating the alert handler, the next step is to import it into SAM.

1. First, export the alert handler in XML format. How to do this depends on the IDE you are using.

2. Next, log in to the Management Portal for the SAM Manager from a web browser, using the following address:

   ```
   http://<sam-domain-name>:8080/csp/sys/UtilHome.csp
   ```

   where <*sam-domain-name*> is the fully qualified DNS name or IP address of the system SAM is running on.
3. Navigate to the **Classes** page (**System Explorer** > **Classes**).
4. Make sure the **SAM** namespace is selected, then click **Import**. This brings up the **Import Classes** dialog.
5. In the **Import Classes** dialog:
   - For **The import file resides on**, select **My Local Machine**.
   - For **Select the path and name of the import file**, click the **Choose File** button and select the alert handler XML file from your file system.
6. At the bottom of the dialog, click **Next**, then **Import**. A result dialog should appear to tell you the status of your import.

After the import is complete, you have successfully added the alert handler to SAM. From now on, any time SAM detects a new alert, it calls the **HandleAlerts()** method of your class.

If you ever need to update an alert handler, simply repeat the steps above with the newer version. This replaces the previous version with the new one.

4.4 Improve Performance When Querying Older Metrics

SAM includes two databases: the Prometheus database (used for short-term metrics storage) and an InterSystems IRIS database (used for longer-term storage). The Prometheus database retains data for two hours in a cache optimized for rapid querying, while the InterSystems IRIS database retains the data for long term analysis.

If you constantly run queries for data older than two hours, increasing the Prometheus retention time may increase performance. Adjust this setting by changing the "`--storage.tsdb.retention.time`" flag in the docker-compose.yml file. For more information, see "Operational aspects" in the Prometheus documentation (https://prometheus.io/docs/prometheus/latest/storage/#operational-aspects).

4.5 Troubleshoot an Unreachable Instance

There are many reasons the state of an instance could become **Unreachable**. This section provides several potential causes and solutions.

If none of these procedures resolve the **Unreachable** status, contact the InterSystems Worldwide Response Center (WRC) for further troubleshooting help.

4.5.1 The target instance is not outputting metrics

The /api/monitor application for the instance you are monitoring with SAM may not be outputting metrics. To determine this, use the web browser or the `curl` command in the command window to access the following URL:

```
http://<instance-host>:<port>/api/monitor/metrics
```

If this does not return a list of metrics, ensure that the instance is on InterSystems IRIS version 2020.1 or higher and that the /api/monitor application is configured to allow unauthenticated access, as described in the section on preparing instances for monitoring.

4.5.2 The target instance has an IP address in the 172.17.x.x range

System Alerting and Monitoring may not be able to reach an instance with an IP address in the 172.17.*x*.*x* range (for example, `172.17.123.123`). This is because Docker uses this IP range for its own networks.

You can resolve this issue by changing the Docker IP address range. To do this, specify a different range (e.g. 10.10.*x*.*x*) in the Docker daemon configuration file using the `default-address-pools` option. Refer to the Docker documentation for further help editing this file.

4.5.3 The target instance is not responding before timeout

The /api/monitor application may be outputting metrics, but failing to respond to the GET request from Prometheus before the connection timeout (ten seconds, by default). An analysis of traffic over the network can confirm that Prometheus is ending each unsuccessful attempt to connect to the instance with a TCP FIN packet.

Restarting the instance may be sufficient to render the /api/monitor application responsive again.

Alternatively, if you have defined custom application metrics for the instance, the time required to compute these metrics may be exceeding the default scraping interval for Prometheus. While it is possible to specify larger values for the scraping interval and timeout parameters in the isc_prometheus.yml file, in this case InterSystems recommends adjusting your organization's monitoring strategy so that SAM can reliably receive updates to all the metrics it monitors with the default frequency.

4.5.4 The SAM database is full

If the **SAM** database fills up, instances may show up as **Unreachable** and stop reporting metrics. To check whether this is the case:

1. Open the SAM Manager from a web browser, using the following address:

   `http://<sam-domain-name>:8080/csp/sys/UtilHome.csp`

2. Navigate to the **Databases** page (**System Operation** > **Databases**).
3. Select **Free Space View**.
4. Check the **% Free** column for the **SAM** database to see whether the value is 0.

If the database is full, you should free some space by deleting data, as described in an earlier section. Once you have done so, shut down System Alerting and Monitoring using the `stop.sh` script, and restart it using `start.sh`.

To prevent this from happening again, you can lower the number of days SAM stores data using the Configuration Settings menu.

Alternatively, you may prefer to change the location where Docker stores the persistent copy of the database to accommodate the volume of data collected. To do this:

1. Shut down SAM using the `stop.sh` script.
2. Change the location where Docker stores a persistent copy of the data from the SAM IRIS container, as described in our guide to setting up SAM.

3. Move the contents of the former storage location to the new location. The default configuration specified by the docker-compose.yml file stores monitoring data in a named volume (`irisdata`) located in the /var/lib/docker/volumes/ directory.
4. Restart SAM using the `start.sh` script.