


Article

[Luca Ravazzolo](#) · Jul 14, 2020  3m read

## System Alerting and Monitoring or SAM - A Use Case

Hi All,

With this article, I would like to show you how easily and dynamically [System Alerting and Monitoring](#) (or SAM for short) can be configured. The use case could be that of a fast and agile CI/CD provisioning pipeline where you want to run your unit-tests but also stress-tests and you would want to quickly be able to see if those tests are successful or how they are stressing the systems and your application (the InterSystems IRIS [backend SAM API is extendable](#) for your [APM](#) implementation).

SAM provides a new graphical interface that displays metrics and alerts. By default, it honors the alerts raised by the IRIS instances we monitor, and although you can go and set your sensors values to your liking, SAM allows you to quickly set up SAM rules -against the included Prometheus so that in your CI/CD pipeline you can quickly be alerted of things as they come through.

There are a few requirements you must pay attention to for running SAM. The appliance is packaged in an easy-to-use container package and assembled automatically with Docker Compose for us. You can get all the details on how to run SAM from the above link. It's easy 😊

With SAM running, we then provision four AWS clusters in four different regions (the fictitious company we work for is a global organization and the quality-assurance team is located in the US on the west coast, while production is running in London for customers in the financial square-mile district).

To quickly provision and run our IRIS clusters we leverage [InterSystems Cloud Manager](#) (ICM) that quickly and easily is driven by simple scripts to do the dirty work of carving out those cloud clusters in the various AWS regions.

When the lengthy provisioning process is finished we call the [SAM-populate.sh](#) script that, adequately configured (see [env-config.sh](#)), populates in few seconds our running SAM solution on our local workstation.

Through a single SAM page, one is able to monitor several InterSystems IRIS clusters without having to jump across several AWS management portals pages. Neat :) but more to the point, it allows us "IRISanians" 😊 to see how our data-platform instances are doing vs having only the default EC2-OS metrics. Bear in mind that SAM at present offers more than 100 InterSystems IRIS kernel [metrics](#) and combine those with InterSystems IRIS [alerts](#). The fusion of the two should be the beginning... not of a love story, but of a more accurate reading of your systems.

You can find the code of the described project & use-case at [this Github link](#).

The project highlights are:

1. The use of the [SAM REST API](#)
2. SAM simple, clean & effective UI for the view of multiple InterSystems IRIS instances clustered together
3. The use of the new [CPF merge functionality](#)
4. [ICM](#) power and programmability or was it scriptability? 😊

If you want to run it pay attention to:

- the need for InterSystems IRIS keys with sharding technology enabled
- the right InterSystems IRIS container (you'll need 2020.2 and above)
- the container registry ICM will pull from
- the container registry credentials
- the cost you'll incur in running those clusters
- and probably few other things :)

I hope you'll find the scripts useful and inspiring for

- the SAM REST API calls and
- the way ICM is leveraged non-interactively.

Be well & safe

--

[#Cloud](#) [#Dashboards](#) [#DevOps](#) [#Docker](#) [#Monitoring](#) [#REST API](#) [#SAM - System Alerting and Monitoring](#) [#System Administration](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/system-alerting-and-monitoring-or-sam-use-case>