

Article

[Murray Oldfield](#) · Nov 18, 2019 ^{8m} read

Example: Review Monitor Metrics From InterSystems IRIS Using Default REST API

The following steps show you how to display a sample list of metrics available from the `/api/monitor` service.

In the last post, I gave an overview of the service that exposes IRIS metrics in Prometheus format. The post shows how to set up and run [IRIS preview release 2019.4](#) in a container and then list the metrics.

This post assumes you have Docker installed. If not, go and do that now for your platform :)

Step 1. Download and run the IRIS preview in docker

Follow the download instructions at [Preview Distributions](#) to download the Preview Licence Key and an IRIS Docker image. For the example, I have chosen InterSystems IRIS for Health 2019.4.

Follow the instructions at [First Look InterSystems Products in Docker Containers](#). If you are familiar with containers, jump to the section titled: Download the InterSystems IRIS Docker Image.

The following terminal output illustrates the processes I used to load the docker image. The docker load command may take a couple of minutes to run;

```
$ pwd
/Users/myhome/Downloads/iris_2019.4

$ ls
InterSystems IRIS for Health (Container)_2019.4.0_Docker(Ubuntu)_12-31-2019.ISCkey  i
rishealth-2019.4.0.379.0-docker.tar

$ docker load -i irishealth-2019.4.0.379.0-docker.tar
762d8e1a6054: Loading layer [=====>] 91
.39MB/91.39MB
e45cfbc98a50: Loading layer [=====>] 15
.87kB/15.87kB
d60e01b37e74: Loading layer [=====>] 12
.29kB/12.29kB
b57c79f4a9f3: Loading layer [=====>] 3.
072kB/3.072kB
b11f1f11664d: Loading layer [=====>] 73
.73MB/73.73MB
22202f62822e: Loading layer [=====>] 2.
656GB/2.656GB
50457c8fa41f: Loading layer [=====>] 1
4.5MB/14.5MB
bc4f7221d76a: Loading layer [=====>] 2.
048kB/2.048kB
4db3eda3ff8f: Loading layer [=====>] 1.
491MB/1.491MB
Loaded image: intersystems/irishealth:2019.4.0.379.0
```

```
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED
SIZE
intersystems/irishealth 2019.4.0.379.0 975a976ad1f4     3 weeks ago
2.83GB
```

For simplicity copy the key file to a folder location you will use for persistent storage and rename to iris.key;

```
$ mkdir -p /Users/myhome/iris/20194
$ cp 'InterSystems IRIS for Health (Container)_2019.4.0_Docker(Ubuntu)_12-31-2019.ISC
key' /Users/myhome/iris/20194/iris.key

$ cd /Users/myhome/iris/20194
$ ls
iris.key
```

Start IRIS using the folder you created for persistent storage;

```
$ docker run --name iris --init --detach --publish 52773:52773 --volume `pwd`:/external
intersystems/irishealth:2019.4.0.379.0 --key /external/iris.key

$ docker ps -a
CONTAINER ID          IMAGE                                COMMAND
CREATED              STATUS                            PORTS                NAMES
009e52c121f0         intersystems/irishealth:2019.4.0.379.0  "/iris-main --key /e..."
About a minute ago   Up About a minute (healthy)         0.0.0.0:52773->52773/tcp  iris
```

Cool! You can now connect to the [System Management Portal](#) on the running container. I used login/password SuperUser/SYS; you will be prompted to change the password first time.

Navigate to Web Applications. System > Security Management > Web Applications

You will see a Web Application: /api/monitor this is the service exposing IRIS metrics.

You do not have to do anything to return metrics, it just works.

Step 2. Preview metrics

In later posts, we will scrape this endpoint with Prometheus or SAM to collect metrics at set intervals. But for now, let us see the full list of metrics returned for this instance. A simple way, for example on Linux and OSX, is to issue an HTTP GET using the curl command. For example; on my (pretty much inactive) container the list starts with;

```
$ curl localhost:52773/api/monitor/metrics
:
:
iris_cpu_usage 0
iris_csp_activity{id="127.0.0.1:52773"} 56
iris_csp_actual_connections{id="127.0.0.1:52773"} 8
iris_csp_gateway_latency{id="127.0.0.1:52773"} .588
iris_csp_in_use_connections{id="127.0.0.1:52773"} 1
```

```
iris_csp_private_connections{id="127.0.0.1:52773"} 0
iris_csp_sessions 1
iris_cache_efficiency 35.565
:
```

And so on. The list can be very long on a production system. I have dumped the full list at the end of the post.

Another useful way is to use the [Postman application](#), but there are other ways. Assuming you have installed Postman for your platform, you can issue an HTTP GET and see the metrics returned.

Summary

That is all for now. In the next post, I will start with collecting the data in Prometheus and look at a sample Grafana dashboard.

Full list from preview container

A production system will have more metrics available. As you can see from some of the labels, for example; {id="IRISLOCALDATA"} there are some metrics that are per-database or for CPU by process type {id="CSPDMN"}.

```
iris_cpu_pct{id="CSPDMN"} 0
iris_cpu_pct{id="CSPSRV"} 0
iris_cpu_pct{id="ECPWorker"} 0
iris_cpu_pct{id="GARCOL"} 0
iris_cpu_pct{id="JRNDMN"} 0
iris_cpu_pct{id="LICENSESRV"} 0
iris_cpu_pct{id="WDSLAVE"} 0
iris_cpu_pct{id="WRTDMN"} 0
iris_cpu_usage 0
iris_csp_activity{id="127.0.0.1:52773"} 57
iris_csp_actual_connections{id="127.0.0.1:52773"} 8
iris_csp_gateway_latency{id="127.0.0.1:52773"} .574
iris_csp_in_use_connections{id="127.0.0.1:52773"} 1
iris_csp_private_connections{id="127.0.0.1:52773"} 0
iris_csp_sessions 1
iris_cache_efficiency 35.850
iris_db_expansion_size_mb{id="ENSLIB"} 0
iris_db_expansion_size_mb{id="HSCUSTOM"} 0
iris_db_expansion_size_mb{id="HSLIB"} 0
iris_db_expansion_size_mb{id="HSSYS"} 0
iris_db_expansion_size_mb{id="IRISAUDIT"} 0
iris_db_expansion_size_mb{id="IRISLOCALDATA"} 0
iris_db_expansion_size_mb{id="IRISSYS"} 0
iris_db_expansion_size_mb{id="IRISTEMP"} 0
iris_db_free_space{id="ENSLIB"} .055
iris_db_free_space{id="HSCUSTOM"} 2.3
iris_db_free_space{id="HSLIB"} 113
iris_db_free_space{id="HSSYS"} 9.2
iris_db_free_space{id="IRISAUDIT"} .094
iris_db_free_space{id="IRISLOCALDATA"} .34
iris_db_free_space{id="IRISSYS"} 6.2
iris_db_free_space{id="IRISTEMP"} 20
iris_db_latency{id="ENSLIB"} 0.030
iris_db_latency{id="HSCUSTOM"} 0.146
```

```
iris_db_latency{id="HSLIB"} 0.027
iris_db_latency{id="HSSYS"} 0.018
iris_db_latency{id="IRISAUDIT"} 0.017
iris_db_latency{id="IRISSYS"} 0.020
iris_db_latency{id="IRISTEMP"} 0.021
iris_db_max_size_mb{id="ENSLIB"} 0
iris_db_max_size_mb{id="HSCUSTOM"} 0
iris_db_max_size_mb{id="HSLIB"} 0
iris_db_max_size_mb{id="HSSYS"} 0
iris_db_max_size_mb{id="IRISAUDIT"} 0
iris_db_max_size_mb{id="IRISLOCALDATA"} 0
iris_db_max_size_mb{id="IRISSYS"} 0
iris_db_max_size_mb{id="IRISTEMP"} 0
iris_db_size_mb{id="HSLIB",dir="/usr/irissys/mgr/hslib/"} 1321
iris_db_size_mb{id="HSSYS",dir="/usr/irissys/mgr/hssys/"} 21
iris_db_size_mb{id="ENSLIB",dir="/usr/irissys/mgr/enslib/"} 209
iris_db_size_mb{id="IRISSYS",dir="/usr/irissys/mgr/"} 113
iris_db_size_mb{id="HSCUSTOM",dir="/usr/irissys/mgr/HSCUSTOM/"} 11
iris_db_size_mb{id="IRISTEMP",dir="/usr/irissys/mgr/iristemp/"} 21
iris_db_size_mb{id="IRISAUDIT",dir="/usr/irissys/mgr/irisaudit/"} 1
iris_db_size_mb{id="IRISLOCALDATA",dir="/usr/irissys/mgr/irislocaldata/"} 1
iris_directory_space{id="HSLIB",dir="/usr/irissys/mgr/hslib/"} 53818
iris_directory_space{id="HSSYS",dir="/usr/irissys/mgr/hssys/"} 53818
iris_directory_space{id="ENSLIB",dir="/usr/irissys/mgr/enslib/"} 53818
iris_directory_space{id="IRISSYS",dir="/usr/irissys/mgr/"} 53818
iris_directory_space{id="HSCUSTOM",dir="/usr/irissys/mgr/HSCUSTOM/"} 53818
iris_directory_space{id="IRISTEMP",dir="/usr/irissys/mgr/iristemp/"} 53818
iris_directory_space{id="IRISAUDIT",dir="/usr/irissys/mgr/irisaudit/"} 53818
iris_disk_percent_full{id="HSLIB",dir="/usr/irissys/mgr/hslib/"} 10.03
iris_disk_percent_full{id="HSSYS",dir="/usr/irissys/mgr/hssys/"} 10.03
iris_disk_percent_full{id="ENSLIB",dir="/usr/irissys/mgr/enslib/"} 10.03
iris_disk_percent_full{id="IRISSYS",dir="/usr/irissys/mgr/"} 10.03
iris_disk_percent_full{id="HSCUSTOM",dir="/usr/irissys/mgr/HSCUSTOM/"} 10.03
iris_disk_percent_full{id="IRISTEMP",dir="/usr/irissys/mgr/iristemp/"} 10.03
iris_disk_percent_full{id="IRISAUDIT",dir="/usr/irissys/mgr/irisaudit/"} 10.03
iris_ecp_conn 0
iris_ecp_conn_max 2
iris_ecp_connections 0
iris_ecp_latency 0
iris_ecps_conn 0
iris_ecps_conn_max 1
iris_glo_a_seize_per_sec 0
iris_glo_n_seize_per_sec 0
iris_glo_ref_per_sec 7
iris_glo_ref_rem_per_sec 0
iris_glo_seize_per_sec 0
iris_glo_update_per_sec 2
iris_glo_update_rem_per_sec 0
iris_journal_size 2496
iris_journal_space 50751.18
iris_jrn_block_per_sec 0
iris_jrn_entry_per_sec 0
iris_jrn_free_space{id="WIJ",dir="default"} 50751.18
iris_jrn_free_space{id="primary",dir="/usr/irissys/mgr/journal/"} 50751.18
iris_jrn_free_space{id="secondary",dir="/usr/irissys/mgr/journal/"} 50751.18
iris_jrn_size{id="WIJ"} 100
iris_jrn_size{id="primary"} 2
iris_jrn_size{id="secondary"} 0
iris_license_available 31
```

```
iris_license_consumed 1
iris_license_percent_used 3
iris_log_reads_per_sec 5
iris_obj_a_seize_per_sec 0
iris_obj_del_per_sec 0
iris_obj_hit_per_sec 2
iris_obj_load_per_sec 0
iris_obj_miss_per_sec 0
iris_obj_new_per_sec 0
iris_obj_seize_per_sec 0
iris_page_space_per_cent_used 0
iris_phys_mem_per_cent_used 95
iris_phys_reads_per_sec 0
iris_phys_writes_per_sec 0
iris_process_count 29
iris_rtn_a_seize_per_sec 0
iris_rtn_call_local_per_sec 10
iris_rtn_call_miss_per_sec 0
iris_rtn_call_remote_per_sec 0
iris_rtn_load_per_sec 0
iris_rtn_load_rem_per_sec 0
iris_rtn_seize_per_sec 0
iris_sam_get_db_sensors_seconds .000838
iris_sam_get_jrn_sensors_seconds .001024
iris_system_alerts 0
iris_system_alerts_new 0
iris_system_state 0
iris_trans_open_count 0
iris_trans_open_secs 0
iris_trans_open_secs_max 0
iris_wd_buffer_redirty 0
iris_wd_buffer_write 0
iris_wd_cycle_time 0
iris_wd_proc_in_global 0
iris_wd_size_write 0
iris_wd_sleep 10002
iris_wd_temp_queue 42
iris_wd_temp_write 0
iris_wdwij_time 0
iris_wd_write_time 0
iris_wij_writes_per_sec 0
```

[#API](#) [#Best Practices](#) [#Dashboards](#) [#InterSystems Business Solutions and Architectures](#) [#InterSystems Data Platform Blog](#) [#Monitoring](#) [#Performance](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Source URL: <https://community.intersystems.com/post/example-review-monitor-metrics-intersystems-iris-using-default-rest-api>