
Artículo

[Alberto Fuentes](#) · 10 ene, 2020 Lectura de 8 min

Ejemplo: Mostrar Métricas de Monitorización de InterSystems IRIS utilizando la API REST

A continuación veremos cómo mostrar una lista de métricas de ejemplo disponibles a través del servicio `/api/monitor`.

En el [artículo anterior](#), echamos un vistazo al servicio que expone las métricas de IRIS en formato Prometheus. Veamos cómo configurar y ejecutar [IRIS preview release 2019.4](#) en un contenedor y listar las métricas disponibles.

Asumimos que tienes Docker instalado. Si no es así, descárgalo e instálalo para tu sistema operativo :)

Si ya te resulta familiar trabajar con IRIS y contenedores Docker, puedes saltar directamente al Paso 2. Mostrar métricas.

Paso 1. Descargar and ejecutar IRIS en un contenedor

Sigue las instrucciones de [Preview Distributions](#) para descargar la Clave de Licencia Preview y una imagen Docker IRIS. Por ejemplo, yo he escogido InterSystems IRIS for Health 2019.4.

Sigue las instrucciones de [First Look InterSystems Products in Docker Containers](#).

Este es el proceso que hemos seguido para cargar la imagen Docker. El comando `docker load` tardará algunos instantes en completarse.

```
$ 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 [=====>] 14
.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
```

Por simplicidad, copia la licencia al directorio que utilizaremos como almacenamiento persistente y renómbrala a iris.key;

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

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

Ejecuta IRIS utilizando el directorio que acabamos de crear para almacenamiento persistente;

```
$ 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
```

Genial! Ya puedes conectarte al [Portal de Gestión](#) del contenedor que tenemos en ejecución. Entraremos con login/password SuperUser/SYS; a continuación, el sistema te pedirá que cambies la contraseña por defecto.

Dirígete a la configuración de las Aplicaciones Web. System > Security Management > Web Applications

Verás una aplicación web llamada /api/monitor, éste es el servicio que expone las métricas de IRIS.

No tienes que hacer nada adicional para que el servicio retorne las métrica, simplemente funcional :)

Paso 2. Mostrar métricas

En artículos posteriores, emplearemos este endpoint desde Prometheus o SAM para recoger las métricas a intervalos específicos. Pero por ahora, vamos a mostrar la lista completa de métricas disponibles para la instancia. Una forma simple de hacerlo en Linux y OSX es lanzando un HTTP GET a través del comando curl;

```
$ 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
:
:
Y la lista continúa...
```

Otra forma muy útil de hacerlo es utilizar la aplicación [Postman](#). Si tienes instalado Postman, puedes enviar un HTTP GET y examinar las métricas devueltas.

Resumen

Esto es todo por ahora. En el próximo artículo, comenzaremos a recuperar los datos desde Prometheus y veremos un ejemplo de un cuadro de mando de Grafana.

Lista completa de métricas para el contenedor de prueba

Un sistema de producción tendrá muchas más métricas disponibles. Como podéis ver por algunas de las etiquetas, por ejemplo {id="IRISLOCALDATA"} hay métricas que son definidas por base de datos o por CPU por el tipo de proceso{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](#) [#Monitorización](#) [#Paneles de control](#) [#Rendimiento](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de
fuente: <https://es.community.intersystems.com/post/ejemplo-mostrar-m%C3%A9tricas-de-monitorizaci%C3%B3n-de-intersystems-iris-utilizando-la-api-rest>