記事
Toshihiko Minamoto · 2020年5月10日 10m read

例: デフォルトREST APIを使用したInterSystems IRISの監視メトリックのレビュー

次の手順で、/api/monitor サービスから利用可能なメトリックのサンプル一覧を表示することができます。

前回の投稿では、IRISのメトリックをPrometheus形式で公開するサービスの概要を説明しました。
この投稿では、コンテナにIRISプレビューリリース2019.4
をセットアップして実行し、メトリックを一覧表示する方法をお伝えします。

この投稿は、Dockerがインストールされた環境があることを前提としています。
そうでない場合は、今すぐお使いのプラットフォームにインストールしてください :)

ステップ 1. dockerでIRISプレビューをダウンロードして実行する

「プレビューの配布
」のダウンロード手順に従い、プレビューライセンスキーとIRISのDockerイメージをダウンロードします。
この例では、InterSystems IRIS for Health 2019.4を選択しています。

「機能紹介：Dockerコンテナ内のInterSystems製品について」の指示に従ってください。 すでにコンテナに精通している場合は、「InterSystems IRISのDockerイメージをダウンロードする」というタイトルのセクションに進んでください。

次のターミナル出力は、私がDockerイメージの読み込みに使用している手順を示しています。 docker loadコマンドの実行には数分かかる場合があります。

```
$ 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
```

作業しやすくするため、永続ストレージに使用するフォルダーの場所にキーファイルをコピーし、名前を iris.key に変更します。

```
$ 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
```

次のように、永続ストレージに作成したフォルダーを使用してIRISを起動します。

```
$  docker run --name iris --init --detach --publish 52773:52773 --volume `pwd`:/exter
nal 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
```

成功しました。 これでコンテナ内で実行中のシステム管理ポータルに接続できるようになりました。
今回はパスワードに「SuperUser/SYS」を使用しましたが、初回実行時はパスワードの変更が促されます。

「システム > セキュリティ管理 > Webアプリケーション」を開き、Webアプリケーションに移動します。

次のWebアプリケーションが表示されています：/api/monitorはIRISのメトリックを公開するサービスです。

メトリックスを返すために何か特別な設定を行う必要はありません。何もしなくても機能します。

ステップ 2. メトリックをプレビューする

今後の投稿では、このエンドポイントをPrometheusやSAMを使ってスクレイピングし、設定された間隔でメトリックを収集します。
しかし、現時点ではこのインスタンスに対して返されるメトリックの完全なリストを見ることにしましょう。
例えば、LinuxとOSXでは curl コマンドを使用してHTTP GETを発行するのがお手軽です。
例えば、私の（あまり活用していない）コンテナでは次のようなメトリックからリストが表示されます。

```
$ 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
:
:
???????????????????????????????????? ????????????????????????
```

Postmanアプリケーションを使用するのも便利ですが、他にも方法はあります。
ご利用中のプラットフォームにPostmanがインストールされている場合、HTTP GETを発行すれば返されるメトリックを確認できます。

概要

現時点でやるべきことは以上です。 次の投稿では、先にPrometheusでのデータ収集について説明し、その後にGrafanaダッシュボードのサンプルをご紹介します。

プレビューコンテナの完全なリスト

本番環境では、より多くのメトリックを使用できます。
いくつかのラベルからお分かりのように、`{id="IRISLOCALDATA"}` のようにデータベースごと、またはプロセスタイプ `{id="CSPDMN"}` ごとのCPUメトリックもあります。

```
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 #ダッシュボード #パフォーマンス #監視 #InterSystems IRIS #InterSystems IRIS for Health

ソースURL:
https://jp.community.intersystems.com/post/%E4%BE%8B-%E3%83%87%E3%83%95%E3%82%A9%E3%83%AB%E3%83%88rest-api%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%9Fintersystems-iris%E3%

81%AE%E7%9B%A3%E8%A6%96%E3%83%A1%E3%83%88%E3%83%AA%E3%83%83%E3%82%AF%E3%81%AE%E3%83%AC%E3%83%93%E3%83%A5%E3%83%BC