Artículo

[Alberto Fuentes](#) · 16 dic, 2019 Lectura de 7 min

Monitorización de InterSystems IRIS utilizando API REST, Prometheus y Grafana

En [InterSystems IRIS 2019.1.4](#) se incluye ya el servicio `/api/monitor`, que permite acceder a métricas de IRIS en formato Prometheus.

Esto es una gran noticia para aquellos interesados en utilizar métricas de IRIS como parte de su solución de monitorización y alertas. Este servicio (API) es un componente del nuevo System Alerting and Monitoring (SAM) que se liberará en próximas versiones de InterSystems IRIS.

Sin embargo, no es necesario esperar a que se libere System Alerting and Monitoring (SAM) para empezar a planificar y probar esta API para monitorizar tus instancias de IRIS. En próximos artículos trataremos en profundidad las métricas disponibles, su significado y también construiremos cuadros de mando de ejemplo interactivos. Pero en primer lugar, vamos a comenzar con algunas cuestiones básicas.

IRIS (y Caché) está siempre capturando decenas de métricas sobre sí mismo y la plataforma en la que está ejecutándose. Siempre han existido [diferentes maneras de capturar estas métricas para monitorizar IRIS o Caché](#). En la práctica no muchas instalaciones utilizan las soluciones incluidas en IRIS / Caché. Por ejemplo, el History Monitor ha estado disponible desde hace tiempo como base de datos histórica de rendimiento y utilización del sistema. Sin embargo, no había una manera demasiado obvia de aprovechar estas métricas en tiempo real.

Las soluciones construidas sobre la plataforma InterSystems IRIS están pasando de una sola aplicación monolítica ejecutándose en unas pocas instancias en las propias instalaciones de una organización a ser soluciones distribuidas desplegadas en cualquier parte. Para muchos casos de uso, las opciones de monitorización de IRIS existentes no se terminan de adaptar a estos nuevos paradigmas. Para solventar este problema, en lugar de reinventar la rueda, InterSystems utilizará soluciones conocidas y bien probadas del mundo open source para monitorización y alertas.

Prometheus

Prometheus es una solución open source de monitorización muy conocida y ampliamente adoptada. Dispone de una gran variedad de plugins. Está diseñada para poder trabajar en entornos de cloud pero también resulta igual de útil para aplicaciones desplegadas en nuestras propias instalaciones.

Los plugins incluyen sistemas operativos, servidores web como Apache y otras muchas aplicaciones. Prometheus se utiliza generalmente junto con un cliente front-end o de presentación, como por ejemplo Grafana, que provee una interfaz gráfica que es extremadamente adaptable a nuestras necesidades.

Grafana

Grafana es también open source. Conforme esta serie de artículos progrese, pasaremos ejemplos de plantillas de cuadros de mando para escenarios típicos. Podrás utilizar esos ejemplos como base para diseñar los cuadros de mando que te interesen.

El gran potencial de estas herramientas surge cuando combinas las métricas de IRIS en el contexto de las métricas de tu propia solución. Desde los componentes de la plataforma donde se ejecuta, el sistema operativo, IRIS y especialmente cuando además se añade instrumentación para monitorizar tus propias aplicaciones.

¿No he visto esto antes en algún sitio?

Monitorizar IRIS / Caché con Prometheus y Grafana no es algo nuevo. Se han utilizado estas aplicaciones durante años. Si buscas en la Comunidad de Desarrolladores el término "Prometheus" encontrarás otros artículos como por ejemplo [estos excelentes artículos de Mikhail Khomenko](#) que muestran cómo exponer métricas de Caché para utilizarlas con Prometheus.

La diferencia es que ahora la API /api/monitor está incluida y habilitada por defecto. No necesitas incluir código para exponer estas métricas.

Introducción rápida a Prometheus

A continuación vamos a hacer una primera orientación rápida a Prometheus y su terminología. Vamos a verlo a alto nivel y a preparar el trabajo de base que servirá para dejar la puerta abierta para que penséis cómo visualizar y consumir las métricas de IRIS o de otras fuentes.

Prometheus funciona obteniendo datos en formato de serie temporal expuestos por aplicaciones a través de una URL (por ejemplo la API de /api/monitor). Existen librerías de exportadores y clientes para muchos lenguajes, frameworks y aplicaciones open source — por ejemplo para servidores web como Apache, distintos sistemas operativos, docker, Kubernetes, bases de datos y ahora IRIS —.

Los exportadores se utilizan para instrumentalizar aplicaciones y servicios de manera que expongan (hagan accesible) las métricas en una URL. Los componentes estándar como los servidores web, bases de datos y demás se soportan por los exportadores core. Existen otros muchos exportadores open source disponibles en la comunidad Prometheus.

Terminología Prometheus

Estos son algunos términos que vamos a manejar:

- Los destinos son aquellos servicios que quieres monitorizar, como un servidor, una aplicación o un servicio como Apache, IRIS o tu propia aplicación.
- Prometheus interroga a estos destinos utilizando HTTP para obtener métricas en formato de serie temporal.
- Los datos en formato de serie temporal se exponen por las aplicaciones, como por ejemplo IRIS, o través de exportadores.
- Los exportadores se utilizan para aquellos elementos que no podemos controlar como por ejemplo métricas del kernel de Linux.
- Los datos en formato de serie temporal resultantes se almacenan localmente en el servidor de Prometheus en una base de datos**.
- La base de datos de series temporales se puede consultar utilizando un lenguaje de consulta optimizado para ello PromQL. Utilizaremos este lenguaje para crear alertas o desde una aplicación cliente como Grafana para mostrar las métricas en un cuadro de mando.

** Spoiler Alert! Por razones de seguridad, escalabilidad, alta disponibilidad y eficiencia, en la nueva solución System Alerting and Monitoring (SAM) la base de datos utilizada para almacenar las series temporales de Prometheus será InterSystems IRIS! El acceso a la base de datos de Prometheus (IRIS) será transparente y no supondrá ninguna circunstancia especial para acceder desde otras aplicaciones como Grafana.

Modelo de datos de Prometheus

Las métricas que retorna la API `/api/monitor` están en formato Prometheus. Prometheus utiliza un formato simple basado en una métrica por línea:

```
[ (time n, value n), ....]
```

Estas métricas pueden etiquetarse como parejas clave-valor. Las etiquetas son una forma muy potente de filtrar métricas y dimensiones. Por ejemplo, vamos a examinar esta métrica del servicio `/api/monitor` que retorna la cantidad de espacio libre para `journal`:

```
iris_jrn_free_space{id="WIJ",dir="/fast/wij/"} 401562.83
```

El identificador indica qué métrica es y de dónde viene:

```
iris_jrn_free_space
```

Se pueden utilizar diferentes etiquetas para decorar las métricas, y así utilizarlas a la hora de filtrar y consultar. En el ejemplo, podemos ver el `WIJ` y el directorio donde se guarda el propio `WIJ`:

```
id="WIJ",dir="/fast/wij/"
```

Y finalmente, observamos también un valor: `401562.83 (MB)`.

¿Qué métricas tenemos disponibles para IRIS?

En la [documentación](#) aparece la lista de las métricas disponibles. Aunque también puedes simplemente consultar la API en el endpoint `/api/monitor/metrics` para que te devuelva la lista de las métricas.

¿Qué debería monitorizar?

Ten en cuenta los siguientes puntos a la hora de pensar cómo debes monitorizar tus sistemas y aplicaciones.

Cuando te sea posible, utiliza métricas que afecten a los usuarios:

- A los usuarios no les importa que una de tus máquinas esté corta de CPU.
- A los usuarios sí que les importa si el servicio que utilizan es lento o está teniendo errores.
- En tus cuadros de mando principales céntrate en métricas de alto nivel que impacten directamente en los usuarios.

Evita construir una pantalla llena de gráficas en tus cuadros de mando:

- Los humanos, en general, no podemos asimilar demasiados datos a la vez.
- Simplifica y utiliza por ejemplo un cuadro de mando por servicio.

Piensa en términos de servicios, no de máquinas:

- Una vez hayas aislado un problema a un servicio, podrás profundizar y ver realmente cuál es la máquina que presenta el problema.

Referencias

Documentación y descargas: [Prometheus](#) y [Grafana](#).

[@Murray Oldfield](#) presentó una sesión sobre SAM (incluyendo Prometheus y Grafana) en el pasado InterSystems Global Summit 2019. Encontrarás más información en [InterSystems Learning Services](#) buscando el término "System Alerting and Monitoring Made Easy".

[#Administración del sistema](#) [#API](#) [#DevOps](#) [#Monitorización](#) [#Paneles de control](#) [#SAM - System Alerting and Monitoring](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de
fuente: <https://es.community.intersystems.com/post/monitorizaci%C3%B3n-de-intersystems-iris-utilizando-api-rest-prometheus-y-grafana>