Artículo Bernardo Linarez · 28 oct, 2020 Lectura de 12 min

Monitorización de InterSystems IRIS y Caché con Prometheus

Prometheus es uno de los sistemas de monitorización adaptado para recoger datos de series temporales.

Su instalación y configuración inicial son relativamente sencillos. El sistema tiene un subsistema gráfico integrado llamado <u>PromDash</u> para la visualización de datos, pero los desarrolladores recomiendan usar un producto de otro proveedor, llamado <u>Grafana</u>. Prometheus puede monitorizar muchas cosas (hardware, contenedores, distintos sistemas de gestión de base de datos), pero en este artículo me gustaría analizar la monitorización de una instancia de <u>Caché</u> (para ser exactos, será una instancia de Ensemble, pero las métricas serán de Caché). Si te interesa, sigue leyendo.

En nuestro caso extremadamente simple, Prometheus y Caché vivirán en una única máquina (Fedora Workstation 24 x86<u>6</u>4). Versión de Caché:

	_
%SYS>write \$zv	
Cache for UNIX (Red Hat Enterprise Linux for x86-64) 2016.1 (Build 656U) Fri Mar 11 2016 17:58:47 EST	

Instalación y configuración

Vamos a descargar un paquete de distribución de Prometheus desde la página oficial y lo vamos a guardar en la carpeta /opt/prometheus.

prometrieus.to/ download/		×.1.1	C Q Search		M 🖽	• •••	~	
checksums soon.								
Operating system linux Architecture and	nd64 -							
prometheus								
prometheus The Prometheus monitoring system and time ser	ies database. O prometh	eus/promethe	JS					
prometheus The Prometheus monitoring system and time ser 1.4.1 / 2016-11-28 Release notes	es database. O prometh	eus/promethe	JS					
prometheus The Prometheus monitoring system and time ser 1.4.1 / 2016-11-28 Release notes File name	es database. O prometh	eus/promether	Size	SHA256 C	hecksum			

Descomprime el archivo, modifica el archivo de configuración de plantilla según tus necesidades e inicia Prometheus. De forma predeterminada, Prometheus mostrará sus registros directamente en la consola, por eso guardaremos sus registros de actividad en un archivo de log.

Iniciar Prometheus

pwd
/opt/prometheus
prometheus-1.4.1.linux-amd64.tar.gz
tar -xzf prometheus-1.4.1.linux-amd64.tar.gz
Is
prometheus-1.4.1.linux-amd64 prometheus-1.4.1.linux-amd64.tar.gz
cd prometheus-1.4.1.linux-amd64/
Is
console <u>li</u> braries consoles LICENSE NOTICE prometheus prometheus.yml promtool



La configuración prometheus.yml se escribe en lenguaje YAML, que no se lleva bien con los símbolos de tabulación, por lo que solo deberías usar espacios. Ya mencionamos que las métricas se descargarán desde http://localhost:57772 y enviaremos solicitudes a /metrics/cache (el nombre de la aplicación es arbitrario), es decir, la dirección de destino para recopilar métricas será http://localhost:57772/metrics/cache. Se agregará una etiqueta "job=iscc_ache" a cada métrica. Una etiqueta, de forma muy aproximada, es el equivalente a WHERE en SQL. En nuestro caso no se usará, pero funcionará bien para más de un servidor. Por ejemplo, los nombres (y/o instancias) de servidores pueden guardarse en etiquetas, y luego se pueden usar las etiquetas para parametrizar solicitudes para realizar gráficas. Vamos a asegurarnos de que Prometheus está funcionando (en la salida de arriba podemos ver el puerto que está escuchando: 9090):

Collocalhost.9090/graph	V C Search	☆ 自 ♣ 1	♠ ♥ ♥ ▪ ♥ ≡
Prometheus Alerts Graph Status - Help			
Expression (press Shift+Enter for newlines)		i.	
Execute - insert metric at cursor -			
Graph Console			
Element	Value		
no data			
			Remove Graph
Add Graph			

Se abre una interfaz web, lo que significa que Prometheus está funcionando. Sin embargo, aún no ve métricas de Caché (verifiquemos haciendo clic en Status Targets):

🗲 🛞 localhost:909)/targets					⊽ C Q Sean	h	☆	1 ₽	⋒	9	* -	Ξ
Prometheus	Alerts	Graph	Status -	Help									
Targets													
isc_cache													
Endpoint				State	Labels	Last Scrape	Error						
http://localhost:5	772/metrics	/cache		DOWN	instance="localhost:57772"	7.501s ago	server returned HTTP s	tatus 40	4 Not Fo	ound			

Preparación de métricas

Nuestra tarea es dejar las métricas disponibles para Prometheus en <u>un formato adecuado</u> en <u>http://localhost:57772/metrics/cache</u>. Usaremos <u>las capacidades REST de Caché</u> debido a su simplicidad. Es necesario saber que Prometheus solo "entiende" las métricas numéricas, por lo que no exportaremos métricas de cadenas de texto. Para obtener lo último, usaremos la API de la clase <u>SYS.Stats.Dashboard</u>. Estas métricas las

usa el propio Caché para mostrar la barra de herramientas de Sistema:

	Serve	er: HP-6360B Namespac	ce: %SYS	This is a	Test System		
stem Dashboard	User:	_SYSTEM Licensed	to: ISC Learning Services	Instance:	ENS2016		
Global and Routine Statistics E	CP Statistics Disk	and Buffer Statistics	System Resource Stat	tistics	Last auto-up	date: 201	7-05-13 14:11:
SYSTEM PERFORMANCE		System Time			ERRORS AND ALERTS		
Globals/Second:	29.00	System Up Time:		1d 2h 06m	Serious Alerts:		0
Global Refs:	272,450,498	Last Backup:	Mar 26 201	7 09:58AM	Application Errors:		0
Global Sets:	42,330,666	Everen llever					
Routine Refs:	25,259,822	Database Space:		Normal	LICENSING		
Logical Requests:	205,561,317	Database Journal:		Normal	License Limit:		150
Disk Reads:	14,942	Journal Space:		Normal	Current License Use:		
Disk Writes:	99,275	Journal Entries:		46 200 946	21 Highest License Lise:		
Cache Efficiency:	2,385.38	Lock Table:		Normal	3%		
		LUCK Table.		Normal			
FCD		vvrite Daemon:		Normai	TASK MANAGER		
ECP AND SHADOWING		Transactions:			Upcoming Tasks:	Time	Status
Application Servers:	Normal	Processes:		16	Обновление статистики	15:00	Scheduled
Application Server Traffic:	0.00	CSP Sessions:		2	sanpocoe SQL	13.00	Scheduled
Data Servers:	Normal	Most Active Processe	es:		Обновление статистики	16:00	Scheduled
Data Server Traffic:	0.00	Process 25912	Commands	2.426	Обновление статистики	17:00	Scheduled
Shadow Source:	Normal	7847		2,420	SanpocoB SQL		
Shadow Server:	Warning			0	Обновление статистики запросов SQL	18:00	Scheduled
				0	Обновление статистики запросов SQL	19:00	Scheduled

Ejemplo de lo mismo en el Terminal:

%SYS>set dashboard = ##class(SYS.Stats.Dashboard).Sample()
%SYS>zwrite dashboard
dashboard= <object reference="">[2@SYS.Stats.Dashboard]</object>
+general information
oref value: 2
class name: SYS.Stats.Dashboard
reference count: 2
+ attribute values
ApplicationErrors = 0
CSPSessions = 2
CacheEfficiency = 2385.33
DatabaseSpace = "Normal"
DiskReads = 14942
DiskWrites = 99278
ECPAppServer = "OK"
ECPAppSrvRate = 0
ECPDataServer = "OK"
ECPDataSrvRate = 0
GloRets = 272452605
GloRetsPerSec = "70.00"
GloSets = 42330/92
JournalEntries = 16399816
LicenseCurrentDet = 3
Licensecurrent rot = 2
El espacio USER será nuestro "entorno controlado" (sandbox). Para empezar, creamos una aplicación/métricas
REST. Para añadir un nivel muy básico de seguridad, protegeremos nuestro inicio de sesión con una contraseña y

El espacio USER sera nuestro "entorno controlado" (sandbox). Para empezar, creamos una aplicacion/metricas REST. Para añadir un nivel muy básico de seguridad, protegeremos nuestro inicio de sesión con una contraseña y asociaremos la aplicación web con algún recurso, que llamaremos PromResource. Necesitamos desactivar el acceso público al recurso, por lo que haremos lo siguiente:

%SYS>write ##class(Security.Resources).Create("PromResource", "Resource for Metrics web page", "")

La configuración de nuestra aplicación web:

Edit: /metrics		Server: HP-6360B	Namespace: %SYS	This is a	Test System
Save Canc	el	User: _STSTEM	Licensed to: ISC Learning	services instance.	ENS2016
dit definition for we	b application /metrics:				
General	Application Roles	Matching Re	oles		
Name	/metrics Required. (e.g. /csp/appname)				
Description					
Namespace	USER - Default Appli	ication for USER: /o	csp/user 🔲 Namespace D	efault Application	
Enabled	Application CSP/ZEN	Inbound Web	Services		
Permitted Classes					
Security Settings	Resource Requi	red PromResourc	e 🗸	Group By ID	
	Allowed Authentication Metho	ods 🔲 Unauthentic	ated Password 🗆 LDA	P Login Cookie	
Session Settings	Session Timeout 900) seconds	Event Class		
	Use Cookie for Session Alv	ays - Sess	ion Cookie Path /metrics/	-	
Dispatch Class	my.Metrics				

También necesitaremos un usuario con acceso a este recurso. El usuario también debería poder leer desde nuestra base de datos (USER, en nuestro caso) y guardar datos en ella. Además de esto, el usuario necesitará permisos de escritura para la base de datos del sistema CACHESYS, ya que más tarde pasaremos al espacio %SYS en el código. Seguiremos el esquema estándar, es decir, crear un rol PromRole con estos permisos y luego crear un usuario PromUser asignado a este rol. Para la contraseña, usaremos "Secret":

%SYS>write ##class(Security.Roles).Create("PromRole","Role for PromResource","PromResource:U,%DBUSER:RW,%DBCACHESYS:R")
' %SYS>write ##class(Security.Users).Create("PromUser","PromRole","Secret") 1
Este usuario PromUser será el que usaremos para autenticación en la configuración de Prometheus. Una vez hecho, enviaremos una señal SIGNUP al proceso del servidor, para releer la configuración.

Una configuración más segura



Las métricas serán provistas por la clase de procesamiento de solicitudes my. Metrics. Aquí está la implementación:

```
Class my.Metrics Extends %CSP.REST
Parameter ISCPREFIX = "isc_cache";
Parameter DASHPREFIX = {..#ISCPREFIX_"_dashboard"};
XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/cache" Method="GET" Call="getMetrics"/>
</Routes>
}
/// Output should obey the Prometheus exposition formats. Docs:
/// https://prometheus.io/docs/instrumenting/exposition_formats/
111
/// The protocol is line-oriented. A line-feed character (\n) separates lines.
/// The last line must end with a line-feed character. Empty lines are ignored.
ClassMethod getMetrics() As %Status
ł
 set nl = $c(10)
 do ..getDashboardSample(.dashboard)
 do ..getClassProperties(dashboard.%ClassName(1), .propList, .descrList)
 for i=1:1:$ll(propList) {
  set descr = $lg(descrList,i)
  set propertyName = $lg(propList,i)
  set propertyValue = $property(dashboard, propertyName)
  // Prometheus supports time series database
  // so if we get empty (for example, backup metrics) or non-digital metrics
  // we just omit them.
  if ((propertyValue '= "") && ('$match(propertyValue, ".*[-A-Za-z ]+.*"))) {
   set metricsName = ..#DASHPREFIX_..camelCase2Underscore(propertyName)
   set metricsValue = propertyValue
   // Write description (help) for each metrics.
   // Format is that the Prometheus requires.
   // Multiline descriptions we have to join in one string.
   write "# HELP "_metricsName_" "_$replace(descr,nl," ")_nl
   write metricsName_" "_metricsValue_nl
  }
 }
 write nl
 quit $$$OK
}
ClassMethod getDashboardSample(Output dashboard)
ł
 new $namespace
 set $namespace = "%SYS"
 set dashboard = ##class(SYS.Stats.Dashboard).Sample()
}
```

ClassMethod getClassProperties(className As %String, Output propList As %List, Output

```
descrList As %List)
{
 new $namespace
 set $namespace = "%SYS"
 set propList = "", descrList = ""
 set properties = ##class(%Dictionary.ClassDefinition).%OpenId(className).Properties
 for i=1:1:properties.Count() {
  set property = properties.GetAt(i)
  set propList = propList_$lb(property.Name)
  set descrList = descrList_$lb(property.Description)
 }
}
/// Converts metrics name in camel case to underscore name with lower case
/// Sample: input = WriteDaemon, output = _write_daemon
ClassMethod camelCase2Underscore(metrics As %String) As %String
{
 set result = metrics
 set regexp = "([A-Z])"
 set matcher = ##class(%Regex.Matcher).%New(regexp, metrics)
 while (matcher.Locate()) {
  set result = matcher.ReplaceAll("_"_"$1")
 }
 // To lower case
 set result = $zcvt(result, "1")
 // _e_c_p (_c_s_p) to _ecp (_csp)
 set result = $replace(result, "_e_c_p", "_ecp")
 set result = $replace(result, "_c_s_p", "_csp")
 quit result
}
}
```

Usaremos la consola para verificar que nuestros esfuerzos no han sido en vano (agregamos la clave --silent para que curl no nos estorbe con su barra de progreso):

curluser PromUser:Secretsilent -XGET ' <u>http://localhost:57772/metrics/cache</u> ' head -20
HELP isc <u>c</u> ache <u>d</u> ashboard <u>a</u> pplication <u>e</u> rrors Number of application errors that have been logged.
isc <u>c</u> ache <u>d</u> ashboard <u>a</u> pplication <u>e</u> rrors 0
HELP isc <u>c</u> ache <u>d</u> ashboard <u>c</u> sp <u>s</u> essions Most recent number of CSP sessions.
isc <u>c</u> ache <u>d</u> ashboard <u>c</u> sp <u>s</u> essions 2
HELP isccachedashboardcacheefficiency Most recently measured cache efficiency (Global references / (physical reads + writes))
isc <u>c</u> ache <u>d</u> ashboard <u>c</u> ache <u>e</u> fficiency 2378.11
HELP isc <u>c</u> ache <u>d</u> ashboard <u>d</u> isk <u>r</u> eads Number of physical block read operations since system startup.
isc <u>c</u> ache <u>d</u> ashboard <u>d</u> isk <u>r</u> eads 15101
HELP isc <u>c</u> ache <u>d</u> ashboard <u>d</u> isk <u>w</u> rites Number of physical block write operations since system startup
isc <u>c</u> ache <u>d</u> ashboard <u>d</u> isk <u>w</u> rites 106233
HELP isc <u>c</u> ache <u>d</u> ashboard <u>e</u> cp <u>a</u> pp <u>s</u> rv <u>r</u> ate Most recently measured ECP application server traffic in bytes/second.
isc <u>c</u> ache <u>d</u> ashboard <u>e</u> cp <u>a</u> pp <u>s</u> rv <u>r</u> ate 0
HELP isc <u>c</u> ache <u>d</u> ashboard <u>e</u> cp <u>d</u> ata <u>s</u> rv <u>r</u> ate Most recently measured ECP data server traffic in bytes/second.
isc <u>c</u> ache <u>d</u> ashboard <u>e</u> cp <u>d</u> ata <u>s</u> rv <u>r</u> ate 0
HELP isc <u>c</u> ache <u>d</u> ashboardglorefs Number of Global references since system startup.
isc <u>c</u> ache <u>d</u> ashboardglo <u>r</u> efs 288545263
HELP isc <u>c</u> ache <u>d</u> ashboardglorefspersec Most recently measured number of Global references per second.
isc <u>c</u> ache <u>d</u> ashboardglo <u>r</u> efsper <u>s</u> ec 273.00
HELP isc <u>c</u> ache <u>d</u> ashboardglo <u>s</u> ets Number of Global Sets and Kills since system startup.
isc <u>c</u> ache <u>d</u> ashboardglo <u>s</u> ets 44584646
Ahora podemos verificar lo mismo en la interfaz de Prometheus:

Cocalhost:9090/targets		V C Search	☆自	+ 1	9	* -	◙≡
Prometheus Alerts Graph Status - Help							
Targets							
isc_cache							
Endpoint	State	Labels	Last Scrape	e		Error	
http://localhost:57772/metrics/cache	UP	instance="localhost:57772"	14.427s ago)			

Y aquí está la lista de nuestras métricas:

🛞 localho	st:9090/graph	
rometheu	JS Alerts Graph Status - Hel	р
Expression	(press Shift+Enter for newlines)	
Execute	- insert metric at cursor -	
	- insert metric at cursor -	*
Graph C	isc_cache_dashboard_application_errors	
	isc_cache_dashboard_cache_efficiency	
Flomont	isc_cache_dashboard_csp_sessions	
Liement	isc_cache_dashboard_disk_reads	
no data	isc_cache_dashboard_disk_writes	
	isc_cache_dashboard_ecp_app_srv_rate	
	isc_cache_dashboard_ecp_data_srv_rate	-
	isc_cache_dashboard_glo_refs	=
Add Graph	isc_cache_dashboard_glo_refs_per_sec	
Auu Graph	isc_cache_dashboard_glo_sets	
	isc_cache_dashboard_journal_entries	
	isc_cache_dashboard_license_current	
	isc_cache_dashboard_license_current_pct	
	isc_cache_dashboard_license_high	
	isc_cache_dashboard_license_nign_pct	
	isc_cache_dashboard_license_limit	
	isc_cache_dashboard_logical_reads	
	isc_cache_dashboard_processes	
	isc cache dashboard rou reis	

No nos enfocaremos en verlas en Prometheus. Puedes seleccionar la métrica que necesites y hacer clic en el

botón "Execute". Selecciona la pestaña "Graph" para ver la gráfica (muestra la eficiencia del caché):

cecut	e is	c_cache_	dashboard_	Cî 🗖													
ph	Consol	е															
	- 15m	ı	+ *	Until		₩ R	les. (s)	O stacke	d								
<u>.</u>								9 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9									
-																	
-																	
-							<u> </u>										
-																	
		43	44	45	 46	47	48		49	50	51	52	53	54	55	56	

Visualización de métricas

Con fines de visualización, instalemos <u>Grafana</u>. Para este artículo, elegí la instalación desde un tarball. Sin embargo, hay otras opciones de instalación, desde paquetes hasta un contenedor. Sigamos los siguientes pasos (después de crear la carpeta /opt/grafana y entrar a ella):

Files .deb (64bit) .rpm (64bit) .tar.gz (Linux 64bit) .zip (Windows 64bit) .brew (Mac OSX)	
.deb (64bit) .rpm (64bit) .tar.gz (Linux 64bit) .ZIP (Windows 64bit) .brew (Mac OSX)	
SHA1: 81274ebb469ef00c7a471c3a9399d8c10cdfe2df	
<pre>wget https://grafanarel.s3.amazonaws.com/builds/grafana-4.0.2-1481203731.li tar -zxvf grafana-4.0.2-1481203731.linux-x64.tar.gz cd grafana-4.0.2-1481203731</pre>	inux-x64.tar.gz
<pre>cp cont/sample.ini cont/custom.ini # make changes to conf/custom.ini then start grafana-server ./bin/grafana-server</pre>	

Dejemos la configuración sin cambios por ahora. En nuestro último paso, iniciamos Grafana en segundo plano (background mode). Guardaremos el log de Grafana a un archivo, igual que lo hicimos con Prometheus:

./bin/grafana-server > /var/log/grafana.log 2>&1 &

Por defecto, la interfaz web de Grafana está accesible a través el puerto 3000. Usuario/contraseña: admin/admin.

Monitorización de InterSystems IRIS y Caché con Prometheus Published on InterSystems Developer Community (https://community.intersystems.com)

Contract Con	▽	C Q Search	☆自♣	⋒ 9 ≉ -	
	Grafana				
	Log in Sign	ир			
	User admin Password ••••• Log in				
	Forgot your password?				

Para tener instrucciones detalladas sobre cómo hacer que Prometheus funcione con Grafana, hac clic <u>aquí</u>. En resumen, necesitamos añadir una nueva Fuente de Datos del tipo Prometheus. Selecciona tu opción para acceso directo/proxy:

🗲 🛞 localhost:30	000/datasources/new										
🧔 • 😡 Dat	ta Sources										
Add dat	Add data source										
Config	Config Dashboards										
Name	Prometheus O Default										
Туре	Prometheus										
Http setting	Http settings										
Url	http://localhost:9090 0										
Access	proxy 🗸 🕑										
Http Auth											
Basic Auth	With Credentials										
TLS Client Auth	With CA Cert 0										
Add	Cancel										

Una vez hecho esto, necesitamos añadir un tablero con los paneles necesarios. El ejemplo de prueba de un tablero está <u>disponible públicamente</u>, junto con el código de la clase de recolección de métricas. Es posible importar fácilmente un tablero a Grafana (Dashboards Import):

🛓 Import Do	shboard	×
Options		
Name	ISC Cache	×
Fedora	0 Prometheus	
🖺 Save & Open	Cancel Back	

Tras la importación, obtendremos lo siguiente:



Guarda el tablero:

🗲 🛞 loo	alhost:3000/dashboard/db/isc-cache
@ -	🗱 ISC Cache 🖌 🏫 🖻 🖪 🌣
1.0 —	Application Error Save dashboard CTRL+S
0.5	
0 —	
-0.5	
-1.0 —	17:30 18:00 18:30 19:00 19:30 20:00

Puedes seleccionar el intervalo de tiempo y el plazo de actualización en la esquina superior derecha:

Time rangeQuick ranges $2016-12-31 \cdot 17.11:53 \cdot 10.011:53 \cdot 10.011:53$	Ø -	🎆 ISC Cache 🗸	☆	6	a	۵				Zoom Out	> 🕑 Last 3 h	ours Refresh every 5s	ວ
							Time range From: now-3h To: now Refreshing every: 5s App	al and a second	Quick range Last 7 days Last 30 days Last 60 days Last 90 days Last 90 days Last 1 year Last 1 year Last 2 years Last 5 years	2S Yesterday Day before yesterday This day last week Previous week Previous month Previous year	Today Today so far This week This week so fa This month This year	2016-12-31 17:11:53 2016-12-31 20:11:53 Last 5 minutes Last 5 minutes Last 30 minutes Last 30 minutes Last 1 hour Last 1 hour Last 6 hours Last 2 hours Last 24 hours	

Ejemplos de tipos de monitorización

Probemos la monitorización de llamadas a globales:

USER>for i=1:1:1000000 {set ^prometheus(i) = i} USER>kill ^prometheus

Podemos ver que el número de referencias a globales por segundo ha aumentado, mientras que la eficiencia del caché cayó (la global ^Prometheus aún no se ha guardado en caché):



Veamos nuestro uso de licencias. Para esto, vamos a crear una página CSP primitiva, llamada PromTest.csp, en el namespace USER:

<html> <head><title>Prometheus Test Page</title></head> <body>Monitoring works fine!</body> </html>

I a visitaremos tantas veces (asumimos que la aplicación /csp/user no está protegida por contraseña):

ab -n77 http://localhost:57772/csp/user/PromTest.csp

Veremos la siguiente imagen de uso de licencias:



Conclusiones

Como podemos ver, implementar la funcionalidad de monitorización no es nada difícil. Incluso después de unos pocos pasos iniciales, podemos obtener información importante sobre el trabajo del sistema, como: uso de licencias, eficiencia del caché de globales (globals caching) y errores de aplicación. Para este tutorial usamos el tablero <u>SYS.Stats.Dashboard</u>, pero otras clases de paquetes SYS, %SYSTEM, %SYS también merecen atención. También puedes escribir tu propia clase que genere métricas personalizadas para tu propia aplicación. Por ejemplo: el número de documentos de un tipo en particular. Algunas métricas útiles, con el tiempo, se compilarán en una plantilla separada para Grafana.

Continuará

Si le interesa aprender más sobre este tema, escribiré más sobre el mismo. Estos son mis planes:

- 1. Preparar una plantilla para Grafana con métricas para el daemon de registro. Sería interesante hacer algún tipo de gráfico equivalente de la herramienta <u>Amgstat</u> al menos para alguna de sus métricas.
- 2. La protección con contraseña para las aplicaciones web es buena, pero sería bueno verificar la posibilidad de usar certificados.
- 3. Usar Prometheus, Grafana y algunos exportadores para Prometheus como contenedores Docker.
- 4. Usar servicios de descubrimiento (discovery services) para añadir automáticamente nuevas instancias de Caché a la lista de monitorización de Prometheus. Aquí es también donde querría demostrar (en la práctica) lo conveniente que son Grafana y sus plantillas. Esto es parecido a los paneles dinámicos, donde se muestran las métricas de un servidor seleccionado en particular, todo en el mismo tablero.
- 5. Administrador de Alertas de Prometheus (Prometheus Alert Manager).
- 6. Ajustes de configuración de Prometheus relacionados con la duración de los datos almacenados, así como posibles optimizaciones para sistemas con una gran cantidad de métricas y un intervalo corto de recopilación de estadísticas.
- 7. Otras ideas útiles y detalles que surgirán en el camino.

Enlaces útiles

Al preparar este artículo, visité varios sitios web útiles y vi una gran cantidad de videos:

• Prometheus project website

- Grafana project website
- Blog of one of Prometheus developers called Brian Brazil
- Tutorial on DigitalOcean
- Some videos from Robust Perception
- <u>Many videos from a conference devoted to Prometheus</u>

¡Gracias por leer hasta aquí!

#Administración del sistema #Mejores prácticas #Monitorización #Visualización #Caché #Ensemble

URL de

fuente:<u>https://es.community.intersystems.com/post/monitorizaci%C3%B3n-de-intersystems-iris-y-cach%C3%A9-con-prometheus</u>