Índice

Introducción

Cuando tenemos una aplicación desplegada en producción o cuando estamos realizando pruebas de carga o de estrés, una de las cosas que necesitamos hacer es monitorizar diferentes parámetros de la misma y del servidor de aplicaciones que la contiene. Estos parámetros pueden incluir, por ejemplo, la evolución de la memoria, el número de sesiones, etcétera.

En este artículo explicaré cómo configurar JBoss para que comunique diferentes métricas que sean leídas y analizadas por la herramienta Cacti.

SNMP y JMX

SNMP y JMX son las siglas de dos protocolos de monitorización de recursos. El primero es un protocolo genérico, y bien conocido por nuestros queridos compañeros de sistemas, y el segundo es el protocolo propio de la tecnología Java.

Puesto que SNMP es un protocolo más antiguo, que aplica a todo tipo de componentes hardware (como un router) o servicios software (por ejemplo, un servidor web) y que es bien conocido por los profesionales que tienen que administrar estas infraestructuras, tiene una gran cantidad de herramientas y gestores que permiten trabajar con él. Estas herramientas en general sólo capturan información para ser analizada, pero el protocolo también permite la modificación de ciertos parámetros. JMX es algo parecido pero aplicado a la máquina virtual y a los servicios Java.

Puesto que tanto las herramientas como los profesionales que tienen que administrar recursos están más acostumbrados a trabajar con SNMP que con JMX (que pueden llegar a ignorarlo por completo), una posible solución pasa por poder exportar la información proporcionada mediante JMX al protocolo SNMP. El presente artículo intenta explicar básicamente esto: cómo una fuente JMX (nuestro querido JBoss) puede integrarse con una herramienta de gestión SNMP (Cacti).

No entraré a explicar los detalles de los dos protocolos porque, entro otras cosas, ni soy un experto en los mismos ni es relevante para el objetivo de este tutorial. Si queréis más información sólo hay que seguir los enlaces que voy proporcionando. Aquello que vayamos necesitando lo iré explicando a lo largo del tutorial.

A modo de resumen (y extraído de la wikipedia):

> “Una red administrada a través de SNMP consiste de tres componentes claves:

- dispositivos administrados;
- agentes;
- sistemas administradores de red (NMS's).

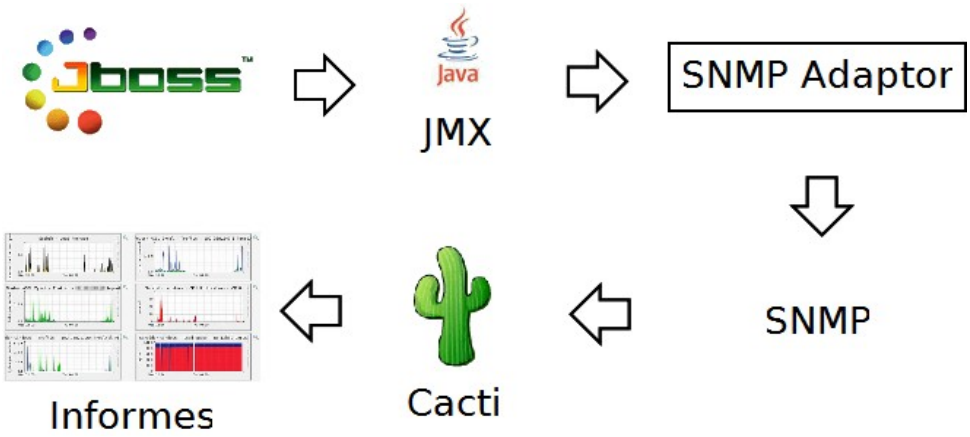
Un dispositivo administrado es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual es puesta a disposición de los NMS's usando SNMP. Los dispositivos administrados, a veces llamados elementos de red, pueden ser routers, servidores de acceso, switches, bridges, hubs, computadores o impresoras.

Un agente es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etcétera), la cual es traducida a un formato compatible con SNMP y organizada en jerarquías.

Un NMS ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS's proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS's deben existir en cualquier red administrada."

Entorno y herramientas

Partiendo de la descripción anterior de infraestructura SNMP, nuestro dispositivo administrado será JBoss, nuestro agente una aplicación desplegada en JBoss que se llama "SNMP adaptor" y nuestro NMS será Cacti.



Los lectores de este blog sin duda conocen qué es JBoss, uno de los servidores de aplicaciones JEE open source más conocidos, sin embargo, es más posible que no estén tan familiarizados con el resto de herramientas ni en tareas de monitorización. Esto último suele pasar porque es muy habitual que este tipo de tareas las realicen perfiles técnicos más orientados a sistemas (lo cual me parece correcto); un desarrollador no debería preocuparse de mirar si los *logs* del Apache tienen cosas raras o de si una máquina debería ampliar su sistema de ficheros porque en los últimos meses se ha aumentado mucho su necesidad de espacio.

El problema suele ser que los perfiles de sistemas dan un trato 'especial' a los servidores Java. Esto se debe a que la administración de estos servicios es un poco diferente al resto de cosas a las que están acostumbrados (archivos de configuración XML vs. texto plano, máquina virtual vs. ejecución nativa, JMX vs. SNMP, etc.). Consecuencia de esta realidad, que yo me he encontrado en más de un sitio, los desarrolladores hemos tenido que aprender algunas cosas de sistemas.

En este artículo trabajaré con una versión virgen de Ubuntu 9.10 y explicaré paso a paso lo siguiente:

- cómo instalar JBoss,
- cómo instalar Cacti,
- configurar JBoss para exportar métricas mediante SNMP y
- configurar Cacti para monitorizar estas métricas.

Instalar JBoss

Aquí no dedicaremos mucho tiempo (entiendo que todo el mundo tiene esta fase superada). Primero instalamos el JDK de Sun con el gestor de paquetes:

```
apt-get install sun-java6-jdk
```

A continuación descargamos el JBoss AS de http://www.jboss.org/jbossas/downloads/ (en el momento de escribir este tutorial la última versión estable, y que yo voy a usar, es la 5.1.0-GA).

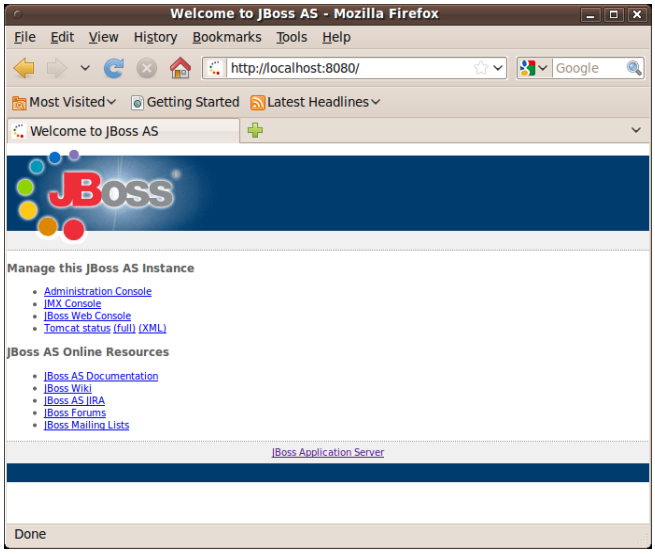
La instalación es sencilla y básicamente consiste en descomprimir el *bundle*, establecer los permisos adecuados y, opcionalmente, configurar el script de arranque en *init.d*. En mi caso instalo el servidor en /opt/jboss pero no es demasiado relevante. Me referiré al directorio raíz de JBoss como JBOSS_HOME.

Recordemos que para poner en marcha el servidor se hace mediante el script JBOSS_HOME/bin/run.sh y que si no le pasamos ningún parámetro por defecto arranca la configuración *default*.

Pongamoslo en marcha para ver si todo funciona correctamente. En el *log* debería aparecer una línea del tipo

```
15:50:56,747 INFO  [ServerImpl] JBoss (Microcontainer) [5.1.0.GA (build: SVNTag=JBoss_5_1_0_GA
date=200905221053)] Started in 1m:41s:932ms
```

y deberíamos poder acceder a la consola de administración del mismo en el puerto 8080 de la máquina:



Instalar Cacti

En un entorno productivo, si se está usando Cacti para monitorizar los sistemas (como es el caso en mi empresa), obviamente serán los chicos de sistemas los que lo tendrán montado y configurado. En nuestra prueba de concepto lo instalaremos en la misma máquina desde cero y con la inestimable ayuda de nuestro querido gestor de paquetes de Ubuntu.

```
apt-get install cacti
```

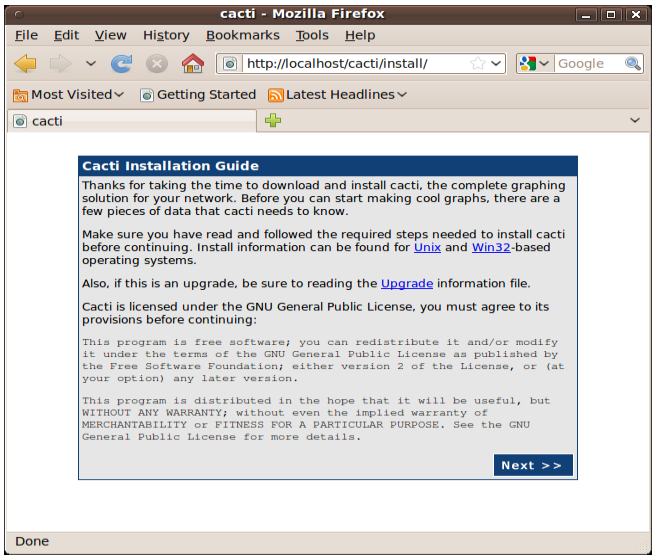
Cacti es una aplicación PHP que almacena la información que necesita en una base de datos MySQL por lo que tiene como dependencias importantes: el servidor MySQL, el Apache y, obviamente, el PHP. Si tenemos una instalación limpia, el gestor de paquetes nos pedirá datos de configuración de estos componentes (como por ejemplo el passwor de root del MySQL).

A continuación el script de instalación nos irá haciendo una serie de preguntas y como no queremos pelearnos con la configuración avanzada de Cacti, cogeremos el camino fácil:

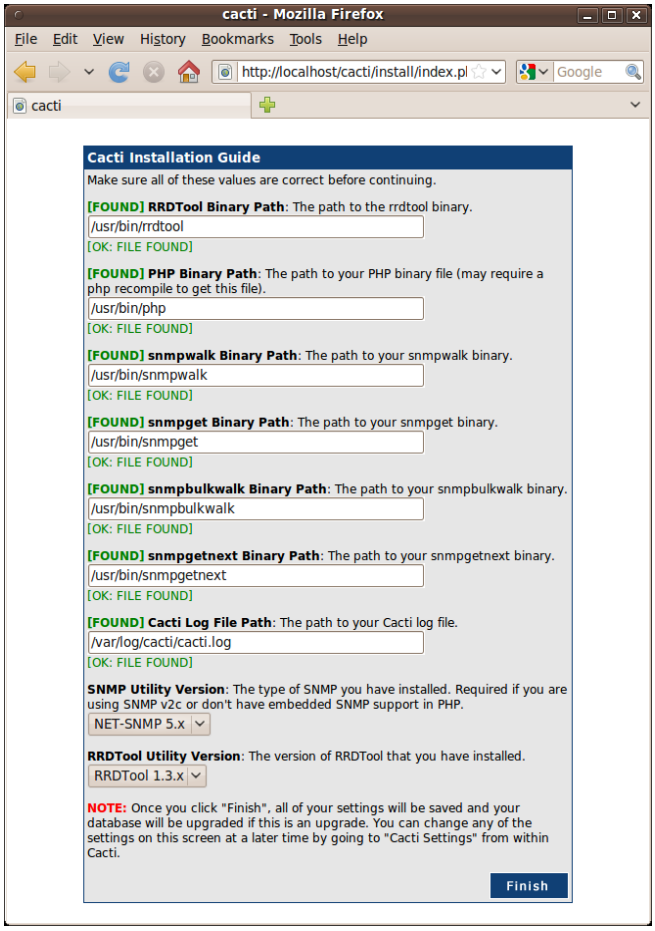
- ¿Qué servidor web quieres?
  - Apache 2.
- Configure database for cacti with dbconfig-common? (que traducido sería, ¿quieres que te configure yo automáticamente la base de datos o prefieres hacerlo tú a mano)
  - Sí.
- Dame la contraseña de root de MySQL parara que te pueda crear la base de datos.
- Dame la contraseña que quieres que utilice para el usuario de Cacti en la base de datos.
  - DATABASE_CACTI_PASSWORD (obviamente, aquí ponemos el que estimemos oportuno).

Aún no ha acabado la instalación pero el resto lo haremos directamente desde la aplicación web. Si todo ha ido bien, tendremos un Apache 2 corriendo con un flamante Cacti instalado en la ruta

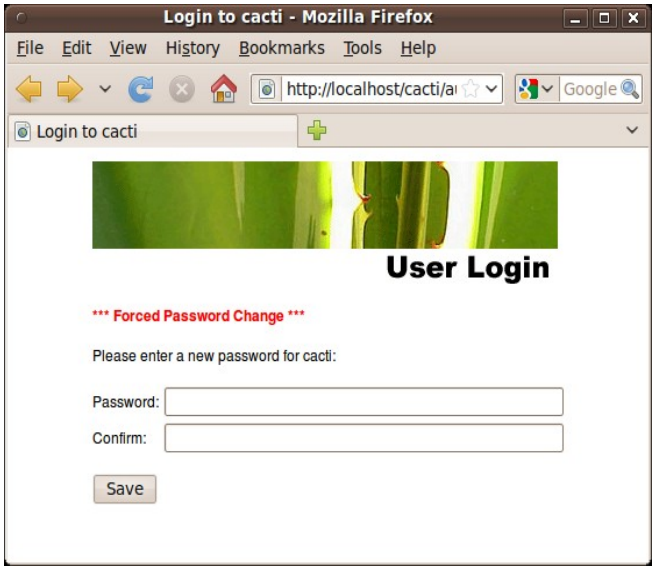
http://localhost/cacti que nos llevará al asistente de configuración:



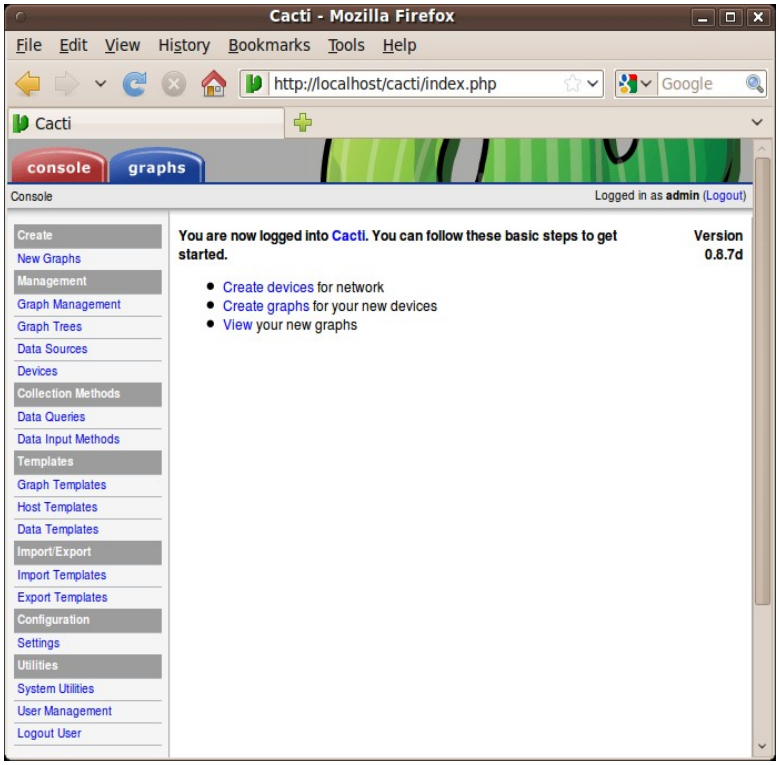
El asistente es muy sencillo y básicamente no requiere explicación: la siguiente pantalla nos pregunta el tipo de instalación (nueva) y la última nos muestra las rutas a la diferentes herramientas que necesita preguntándonos confirmación (configuración que nos parecerá estupenda, por supuesto).



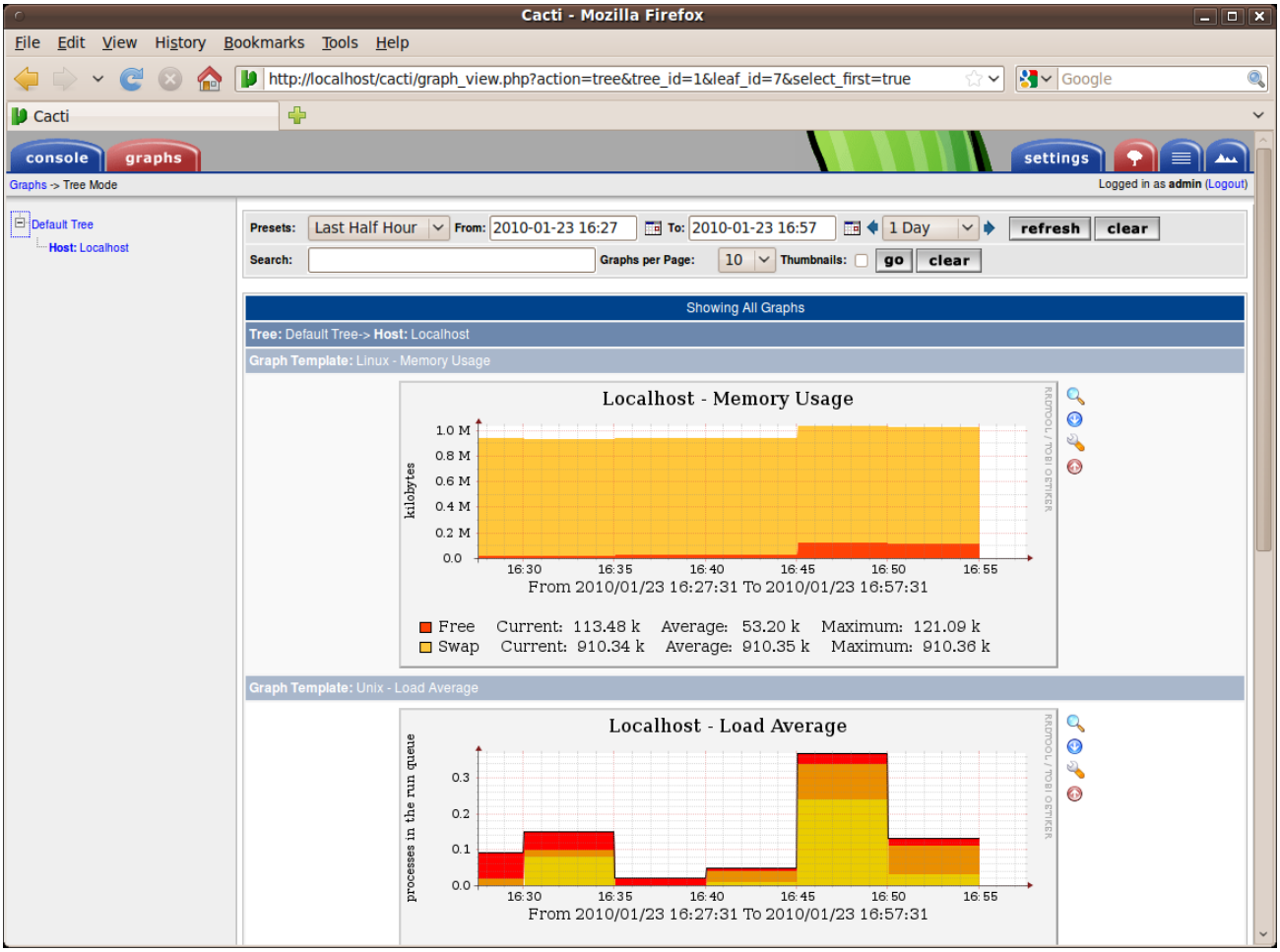
Si todo ha ido bien, al finalizar el asistente, seremos redirigidos a la pantalla de *login* de Cacti, sin embargo, en ningún punto anterior hemos sido preguntados por usuario o contraseña administrativa (las contraseñas que hemos introducido anteriormente se referían al usuario en la base de datos pero no a la aplicación propiamente dicha). Para entrar utilizaremos los valores por defecto que tanto para el usuario como para la contraseña son “admin” y seremos redirigidos a otra pantalla que nos obliga a decidir una contraseña algo más segura. Nos referiremos a esta contraseña como ADMIN_CACTI_PASSWORD.



En este punto ya hemos acabado de configurar Cacti y por fin entramos en la página principal de la aplicación.



Cacti es una herramienta muy poderosa, pero no es el propósito de este artículo explicar todo lo que puede hacer, así que sólo explicaré unas nociones. La interfaz se divide en dos secciones principales representadas por las dos pestañas del margen superior izquierdo: la consola y los gráficos. La consola permite lleva a cabo todas las tareas de configuración (añadir dispositivos SNMP, crear nuevos informes, etc.). La sección gráficos muestra los informes. Efectivamente, Cacti presenta la información como gráficos a lo largo del tiempo (por ejemplo, uso de la memoria a lo largo de la última media hora).



Una única instancia de Cacti puede monitorizar toda la infraestructura de una red. Sólo hay que añadir los agentes en cada uno de los dispositivos y luego configurar Cacti para que los escuche. Los diferentes dispositivos se ven en el árbol de la izquierda. En la configuración básica que tenemos montada sólo estamos monitorizando algunos parámetros básicos de la máquina local.

El marco superior nos permite seleccionar el marco temporal de los gráficos que estamos visualizando. Por defecto Cacti nos muestra algunos parámetros locales interesantes, el uso de memoria, de CPU, etc. Nuestro objetivo es añadir nuevos gráficos como: tamaño del heap del JBoss o número de sesiones de la aplicación X.

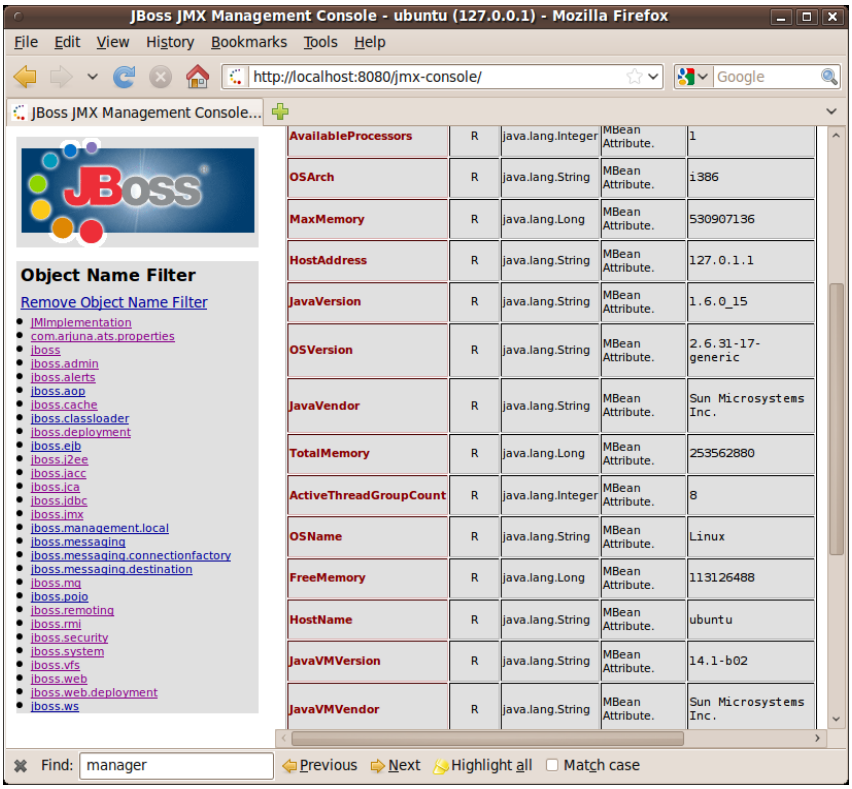
Los gráficos, además, son elementos interactivos que permiten redefinir el periodo temporal que están mostrando a golpe de ratón (zoom-in y zoom-out) o exportar los datos en formato CSV para trabajar con ellos en cualquier herramienta que nos interese (como por ejemplo OpenOffice Calc o Microsoft Excel).

Configurar JBoss para exportar datos por SMMP

En este punto ya tenemos una configuración básica de JBoss y de Cacti. El siguiente paso es hacer que JBoss sea capaz de producir métricas en el protocolo SNMP. Por suerte, va a ser bastante sencillo.

La manera de hacerlo será convertir los datos que de forma nativa ya está generando en JMX a

SNMP. Pero, ¿qué información está exportando por JMX? Muy sencillo, lo podemos ver directamente en la consola JMX de JBoss que está accesible en http://localhost:8080/jmx-console/



Para más información sobre la API JMX de JBoss, podéis ir al propio manual o echarle un ojo a este interesante tutorial que propone otras alternativas de monitorización con herramientas Java.

Ahora que sabemos qué podemos monitorizar, ¿cómo lo exportamos a SNMP? Fácil, desplegaremos una aplicación que viene con el propio JBoss: el SNMP Adaptor.

Esta aplicación viene desplegada únicamente en la configuración de servidor *all*, así que si estamos utilizando otra (como la *default* o una personalizada), tendremos que desplegarla donde corresponda. En mi caso, que estoy usando la configuración *default,* haría lo siguiente:

```
cp -fR JBOSS_HOME/server/all/deploy/snmp-adaptor.sar JBOSS_HOME/server/default/deploy
```

Desde este mismo momento ya estamos produciendo datos SNMP para algunos parámetros que vienen configurados por defecto, como por ejemplo, la cantidad de memoria libre. Para comprobar que está funcionando usaremos un cliente SNMP de línea de comandos:

```
snmpwalk -v 1 -c public localhost:1161 .1.2.3.4.1
```

a lo que JBoss debería responder con algo parecido a:

```
iso.2.3.4.1.1 = INTEGER: 60
iso.2.3.4.1.2 = Gauge32: 97540696
iso.2.3.4.1.3 = Gauge32: 530907136
iso.2.3.4.1.4 = INTEGER: 0
End of MIB
```

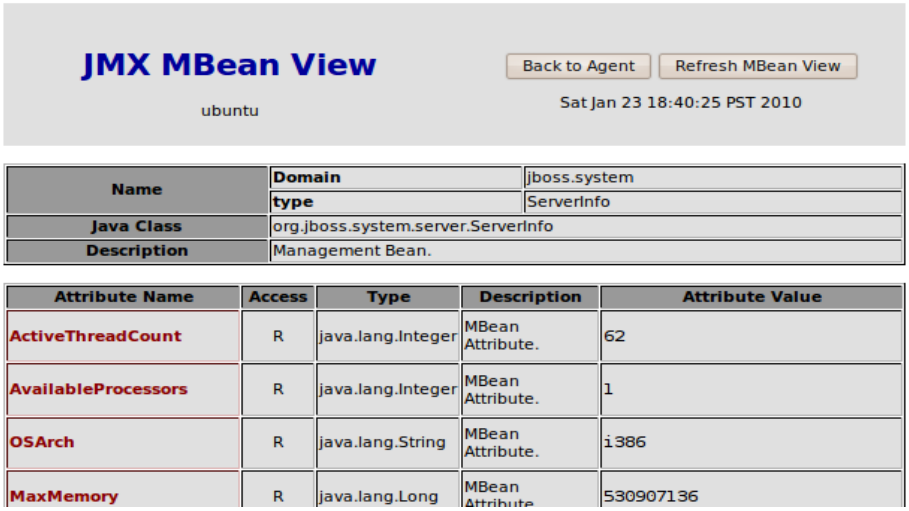
El *snmpwalk* es un cliente simple, que recibe los parámetros de dónde buscar el agente (en *localhost* en el puerto 1161) y qué información buscar, la identificada por .1.2.3.4.1. Los identificadores SNMP (*oid*'s) son un acuerdo entre el agente y los clientes SNMP y tienen estructurar jerárquica. La respuesta que nos ha dado el servidor es el conjunto de todos los parámetros que está produciendo que 'cuelgan' del identificador.1.2.3.4.1.

Cada uno de estos parámetros y su mapeo con el identificador SNMP lo podemos encontrar en archivo de configuración del SNMP Adaptor que se encuentra en el fichero XML siguiente: JBOSS_HOME/server/all/deploy/snmp-adaptor.sar/attributes.xml. Básicamente sirve para configurar un parámetro JMX con un identificador SNMP. Por ejemplo, los cuatro parámetros que hemos visto en el ejemplo se corresponden con esta entrada:

```
<attribute-mappings>

  <!-- basic system information -->
  <mbean name="jboss.system:type=ServerInfo" oid-prefix=".1.2.3.4.1">
    <attribute name="ActiveThreadCount" oid=".1"/>
    <attribute name="FreeMemory" oid=".2"/>
    <attribute name="MaxMemory" oid=".3"/>
  </mbean>
  ../..
</attribute-mappings>
```

Como vemos por un lado estamos identificando el bean JMX y los atributos del mismo que producen la información y por otro el identificador SNMP que queremos vincularle. En la siguiente captura de pantalla del cliente, podemos ver parte de este bean:



Obviamente podemos modificar libremente el fichero para quitar o añadir los bean y atributos que queramos.

Exportar información de sesión para una aplicación web por SNMP

Un ejemplo de información interesante a monitorizar es la asociada a las sesiones de las aplicaciones desplegadas en el servidor. Estos parámetros no están configurados por defecto porque el bean JMX que los exporta depende del nombre del contexto de la aplicación desplegada.

Con la configuración por defecto de JBoss tenemos desplegadas una serie de aplicaciones web, una

de ellas es la propia consola JMX. La usaremos como ejemplo. Así pues, para monitorizar una selección de parámetros relacionados con la sesión para la aplicación jmx-console añadiríamos el siguiente fragmento XML al archivo de configuración del SNMP Adaptor. En negrita he marcado el nombre del bean, que, como vemos, incluye el nombre del la aplicación (del contexto) cuyos parámetros queremos exportar. Los *oid* elegidos son arbitrarios.

```
<mbean name="jboss.web:host=localhost,path=/jmx-console,type=Manager" oid-prefix=".1.2.3.4.5">
        <!-- Number of active sessions at this moment -->
        <attribute name="activeSessions" oid=".1"/>

        <!-- Average time an expired session had been alive -->
        <attribute name="sessionAverageAliveTime" oid=".2"/>

        <!-- longest time an expired session had been alive -->
        <attribute name="sessionMaxAliveTime" oid=".3"/>

        <!-- Total number of sessions created by this manager -->
        <attribute name="sessionCounter" oid=".4"/>

        <!-- Number of sessions that expired ( doesn't include explicit invalidations ) -->
        <attribute name="expiredSessions" oid=".5"/>

        <!-- Number of sessions we rejected due to maxActive beeing reached -->
        <attribute name="rejectedSessions" oid=".6"/>

        <!-- Maximum number of active sessions so far -->
        <attribute name="maxActive" oid=".7"/>

        <!-- Number of duplicated session ids generated -->
        <attribute name="duplicates" oid=".8"/>

        <!-- Time spent doing housekeeping and expiration -->
        <attribute name="processingTime" oid=".9"/>
</mbean>
```

Al igual que antes, hacemos una petición desde la línea de comandos para ver si todo ha ido bien:

```
snmpwalk -v 1 -c public localhost:1161 .1.2.3.4.5
```

para lo que deberíamos obtener una respuesta similar a lo siguiente:

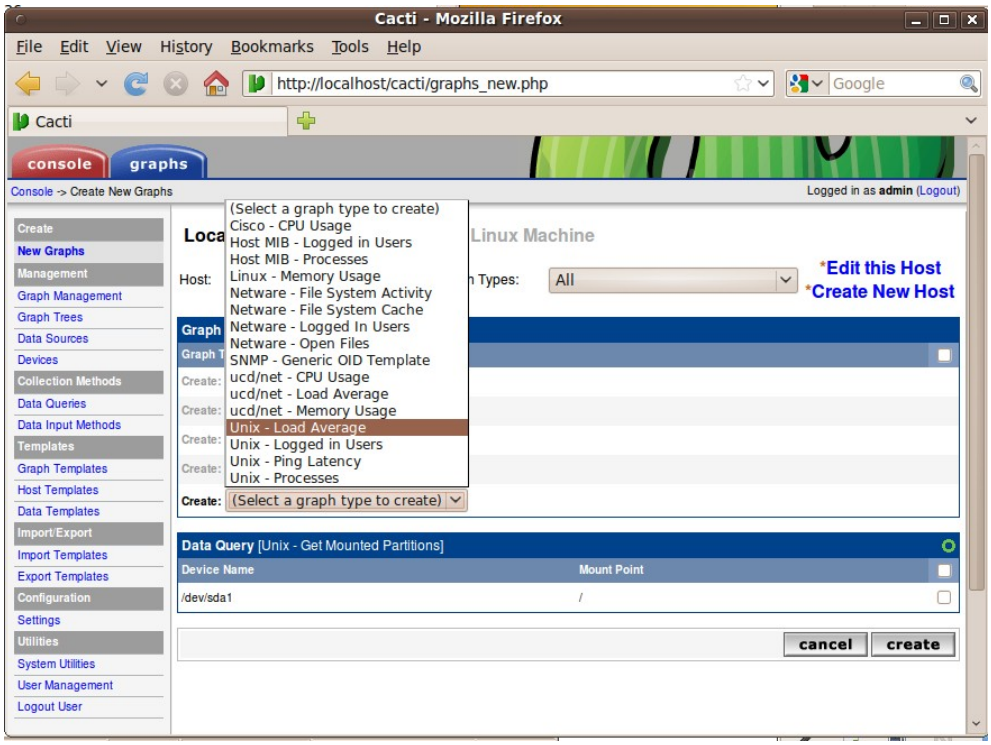
```
iso.2.3.4.5.1 = INTEGER: 1
iso.2.3.4.5.2 = INTEGER: 92
iso.2.3.4.5.3 = INTEGER: 92
iso.2.3.4.5.4 = INTEGER: 2
iso.2.3.4.5.5 = INTEGER: 1
iso.2.3.4.5.6 = INTEGER: 0
iso.2.3.4.5.7 = INTEGER: 1
iso.2.3.4.5.8 = INTEGER: 0
iso.2.3.4.5.9 = Gauge32: 0
End of MIB
```

Configurar el Cacti para monitorizar el JBoss

Cacti es capaz de monitorizar cualquier fuente SNMP. Básicamente hay que añadir los diferentes hosts y dispositivos con la consola administrativa de la interfaz web y luego definir las las gráficas que se quieren graficar sobre los diferentes servicios SNMP que tengan cada uno de esos dispositivos.

La tarea anterior puede ser más o menos tediosa (que no complicada), afortunadamente el propio Cacti viene con una serie de gráficas que permiten configurar a golpe de ratón servicios típicos. Por

ejemplo, la siguiente captura de pantalla muestra alguna de las posibilidades: Console → New Graphs → Create.

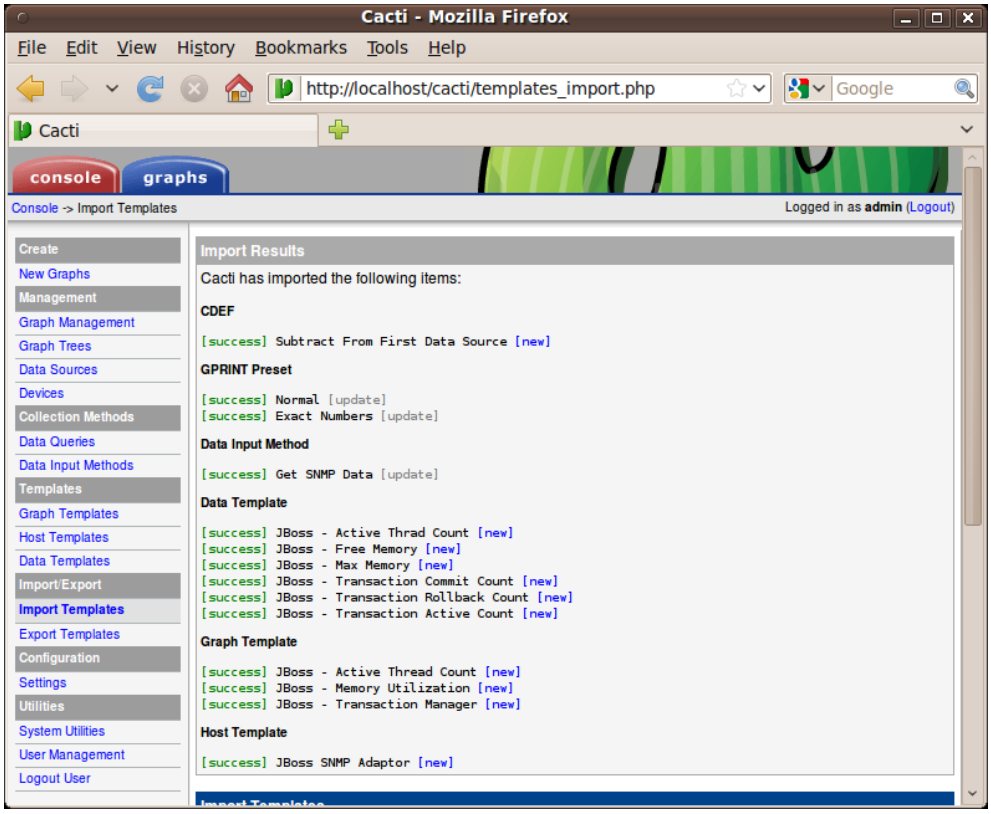


Además de los informes que vienen incluidos con esta distribución en Internet pueden encontrarse para cualquier tipo de dispositivo que se nos pueda ocurrir. La distribución de estos informes se hace mediante lo que Cacti denomina *templates* (plantillas). Estas plantillas son unos ficheros XML que llevan la configuración SNMP de los diferentes dispositivos y además la información para que Cacti pueda crear los gráficos prediseñados. Así pues, si tenemos un router de la marca *Acme* modelo *Roadrunner*, sólo tenemos que buscar un poco en la página de Cacti o googlear un poco porque seguro que alguien ha creado la plantilla.

Como podéis ver en la captura de pantalla anterior, la propia interfaz administrativa tiene dos enlaces que nos permiten importar y exportar plantillas respectivamente.

JBoss no iba a ser menos, así que existen al menos un par de plantillas proporcionadas por la comunidad. Una de ellas puede descargarse desde directamente desde desde este hilo en los foros de Cacti. El autor ha proporcionado una plantilla que grafica directamente los parámetros que vienen en la configuración por defecto del "SNMP Adaptor" de JBoss.

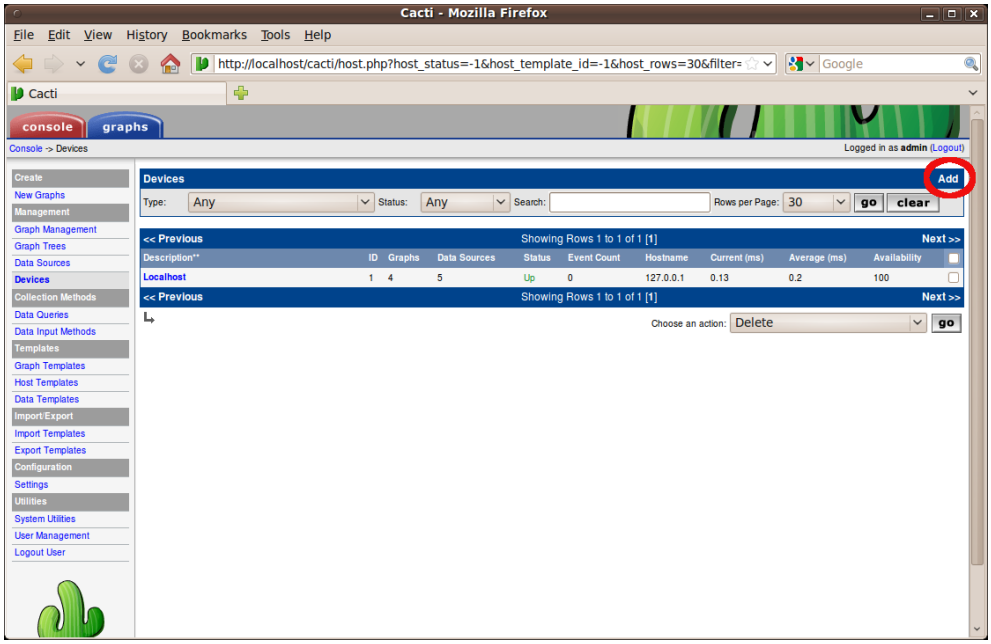
Para instalarla sólo tenemos que descargarla e importarla mediante la interfaz gráfica. Si todo va bien, Cacti nos muestra un informe con todas las entidades procesadas que ha podido importar de la plantilla tal y como se muestra en la siguiente captura.



No hace falta entrar en los detalles, pero básicamente se ha configurado un nuevo agente SNMP, diferentes parámetros producidos por el mismo y tres definiciones de gráficas construidas a partir de estos parámetros. Ahora sólo faltaría construir nuevas gráficas usando estas definiciones para poder monitorizar el JBoss. Para ello deberemos:

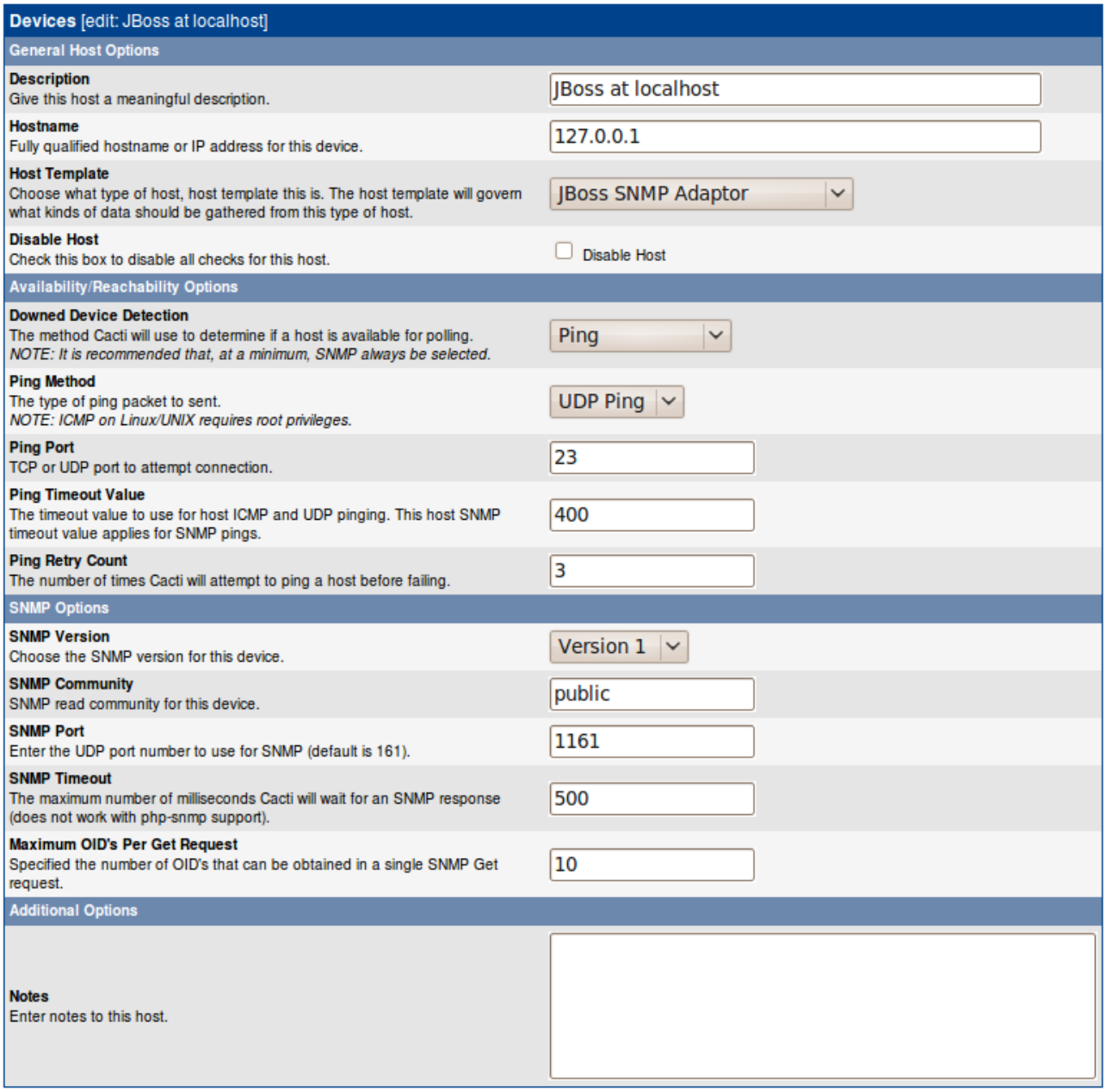
1. Construir un nuevo dispositivo administrado que se corresponda con el JBoss.
2. Configurar las gráficas asociadas.

Para construir un nuevo dispositivo administrado, hacemos click en “Devices” y en el frame superior del área central (que actúa como filtro de los dispositivos listados) hacemos click en “Add” que nos lleva una pantalla de configuración de dispositivo.

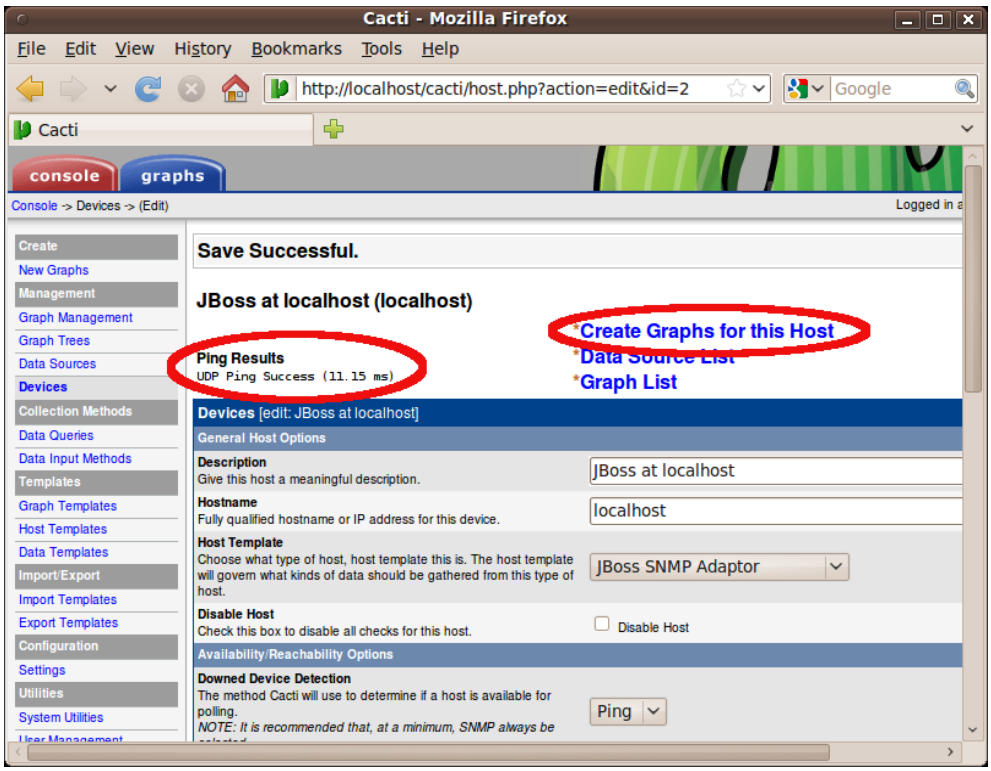


En la pantalla de configuración del dispositivo seleccionamos los siguientes parámetros:

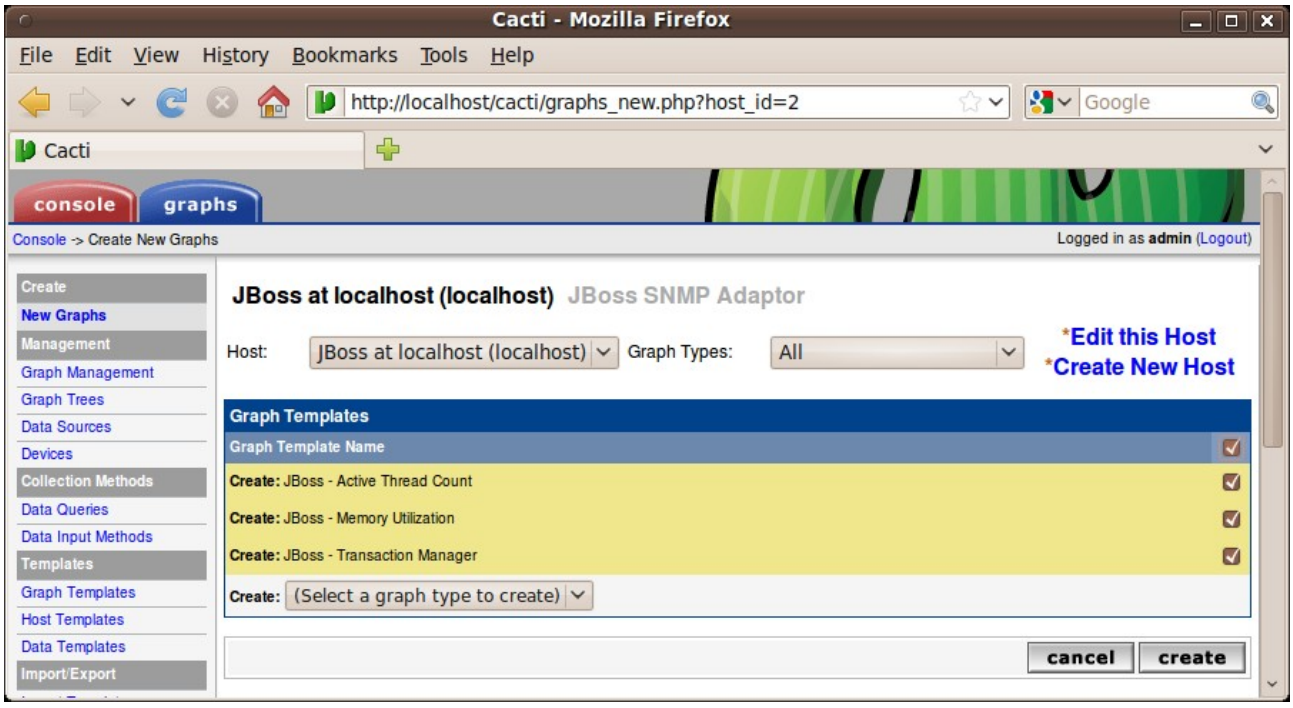
- Configuración general
  - **Description**: un nombre significativo para identificar el servidor JBoss.
  - **Hostname**: el nombre de la máquina o su IP.
  - **Host template**: aquí es donde tenemos que seleccionar el “JBoss SNMP Adaptor” que acabamos de importar.
- La siguiente sección “Availability / Reachibility options” se refiere al mecanismo que debe usar Cacti para determinar que el host sigue vivo. Cuando el mecanismo seleccionado le indica que el host no está ahí, el *poller* de Cacti (el subsistema que interroga al agente SNMP en el dispositivo) deja de recoger datos para ese dispositivo. Los parámetros por defecto basados en *ping* ya me parecen correctos para esta prueba de concepto, pero sentíos libres de adaptarlos a vuestro entorno.
- Configuración SNMP que debe coincidir con la configurada en el SNMP Adaptor
  - **SNMP version**: 1
  - **SNMP community**: public
  - **SNMP Port**: 1161 (¡ojo el puerto por defecto es 161!)
  - Y el resto de parámetros tal y como estén o prefiráis.



Si no ha habido ningún problema y Cacti ha podido comprobar que el servidor está vivo, éste nos redirige a la misma página de edición del servidor pero con un mensaje de *feedback* informando de ello y con nuevas opciones entre las que se nos propone la creación de gráficos asociados al dispositivo.



Hacemos caso de la sugerencia y seguimos el link “Create Graphs for this Host”, que nos lleva a una pantalla que nos permite seleccionar los gráficos a seleccionar de entre los configurados en la plantilla para este tipo de dispositivo. Los seleccionamos todos y pulsamos “create”.



En este momento los gráficos han empezado a procesarse, sin embargo, si vamos a la sección “graphs” no están visibles. Esto se debe a que para verlos debemos incluirlo en un árbol de gráficos

(que es lo que muestra la pantalla “Graphs”). Para ello hacemos click en “Graph Trees” en la pantalla “Console”, sección “Management” y usamos el asistente para incluir los diferentes gráficos.

No nos vamos a complicar, así que aprovechamos el “default tree” para añadir los nuevos gráficos. Paso a paso:

1 Click en “Default Tree”

2 Añadir un nodo que contendrá todos los gráficos de JBoss

2.1 Click en “Add” y configurar los siguientes parámetros:

2.1.1 Parent Item: [root]

2.1.2 Tree Item Type: Header

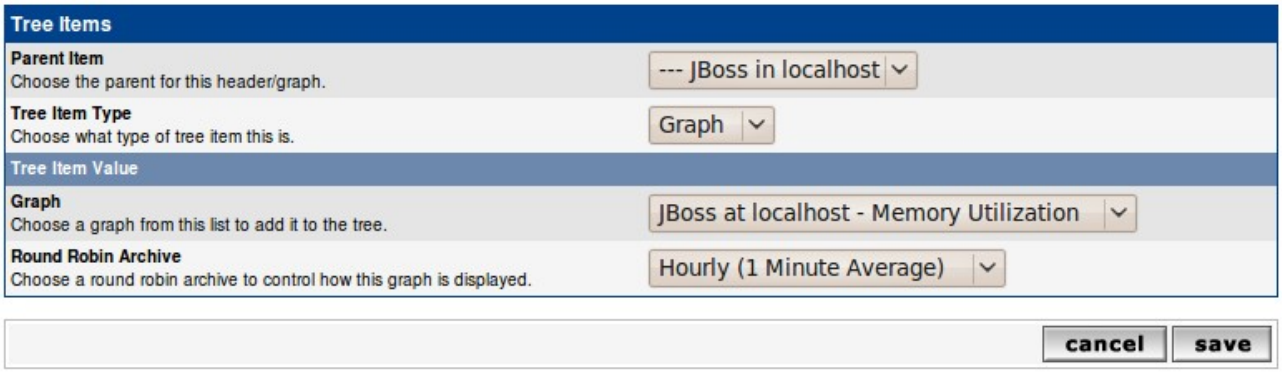
2.1.3 Title: JBoss in localhost

3 Añadir un nodo de tipo gráfico para cada uno de los gráficos:

3.1 Click en “Add” al lado del nombre del nodo y configurar los siguientes parámetros:

3.1.1 Parent Item: JBoss in localhost

3.1.2 Graph: cada uno de los diferentes gráficos de JBoss



Al terminar de añadir los gráficos, por fin podemos ir a la sección “Graphs” y empezar a disfrutar de nuestra nueva monitorización. Ahora sólo habrá que tener un poco de paciencia para que el *poller* de Cacti recoja suficientes datos para que los gráficos aporten información útil.

Monitorizar información de sesión para una aplicación web

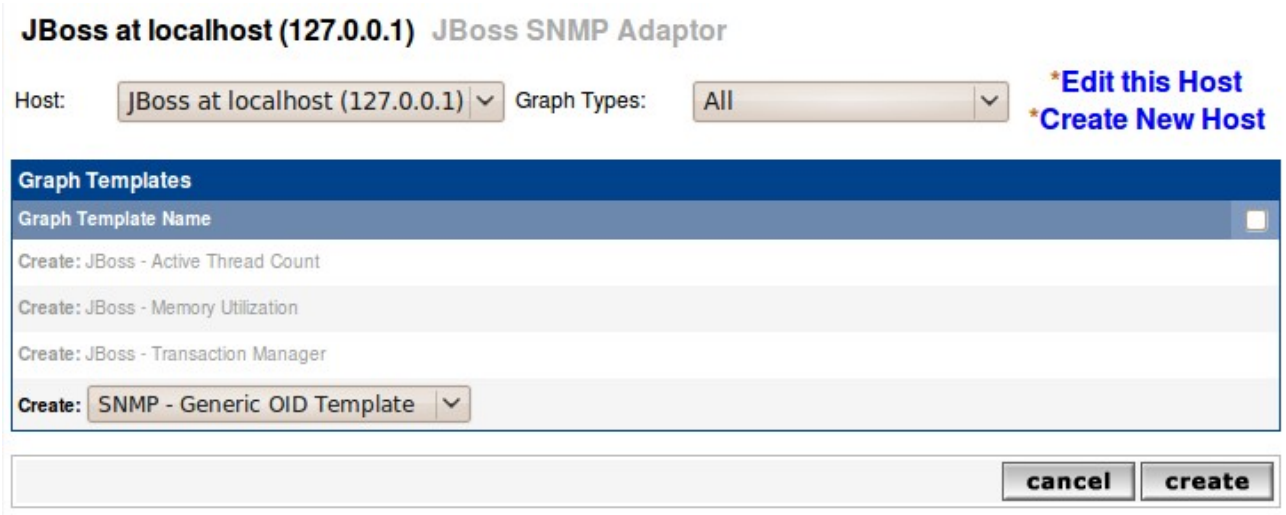
La configuración de los gráficos precedentes ha sido más o menos sencilla porque la plantilla que hemos descargado nos lo ha dado todo hecho, ¿pero qué pasa si queremos graficar otros parámetros SNMP exportados por JBoss (o por cualquier otro dispositivo)? Pues que tenemos que usar la interfaz para crear bastante fácilmente esos nuevos gráficos.

Como ejemplo vamos a construir un gráfico que se genere a partir del número de sesiones activas en cada momento. Si recordáis, en una sección precedente explicamos cómo hacer que JBoss exportara estos datos por SNMP para la aplicación jmx-console. Utilizaremos el mismo ejemplo, así que el *oid* SNMP del número de sesiones activas se corresponde con .1.2.3.4.5.1

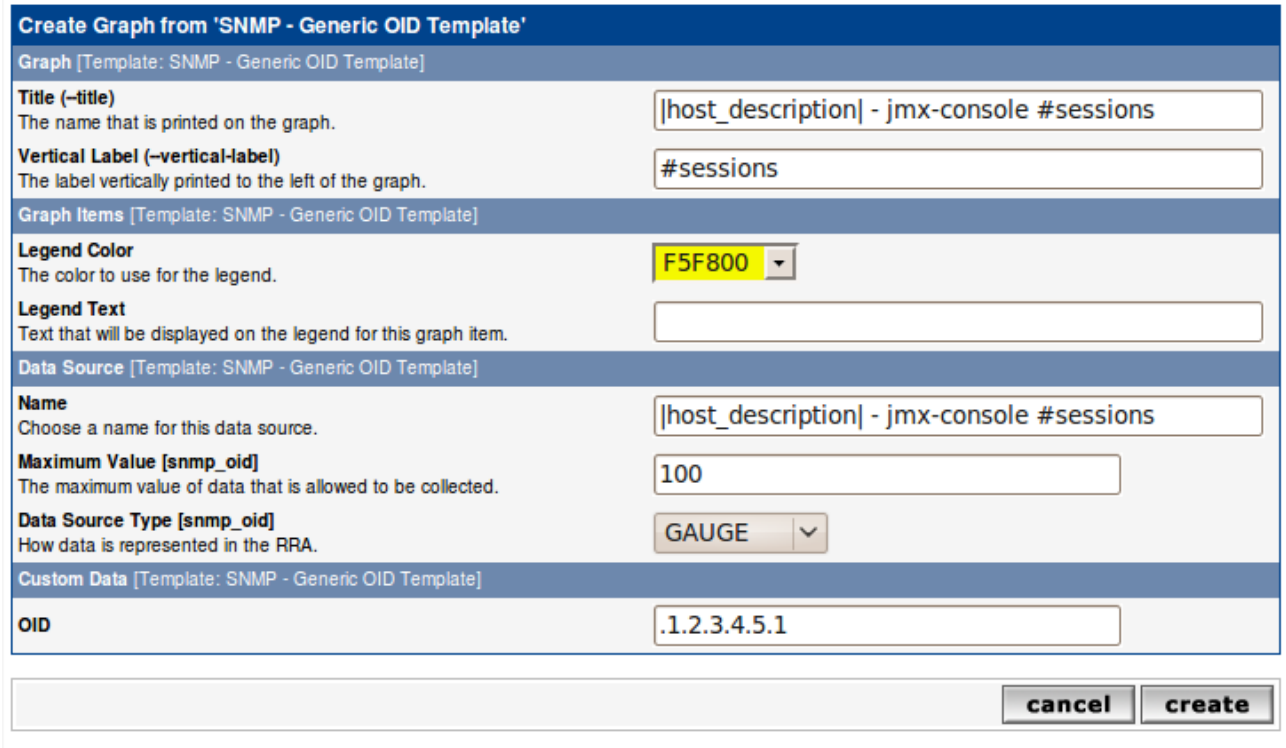
El proceso es sencillo:

1. Crear la definición del nuevo gráfico.
2. Crear un gráfico del nuevo tipo en el dispositivo JBoss.
3. Añadir el gráfico al árbol de gráficos.

Para crear el gráfico en la sección “Console”, clickaremos en el enlace “New graphs”. Nos aseguramos de tener seleccionado el dispositivo correspondiente al JBoss en el desplegable “Host” y la opción “SNMP – Generic OID Template” en el desplegable “Create”. A continuación pulsamos el botón “Create”.



A continuación introducimos los parámetros de configuración del gráfico basado en SNMP tal y como muestra la siguiente captura. Todo son parámetros descriptivos del gráfico generado excepto el *oid* del parámetro que queremos graficar (en nuestro caso el que se corresponde con el número de sesiones exportado por el JBoss).

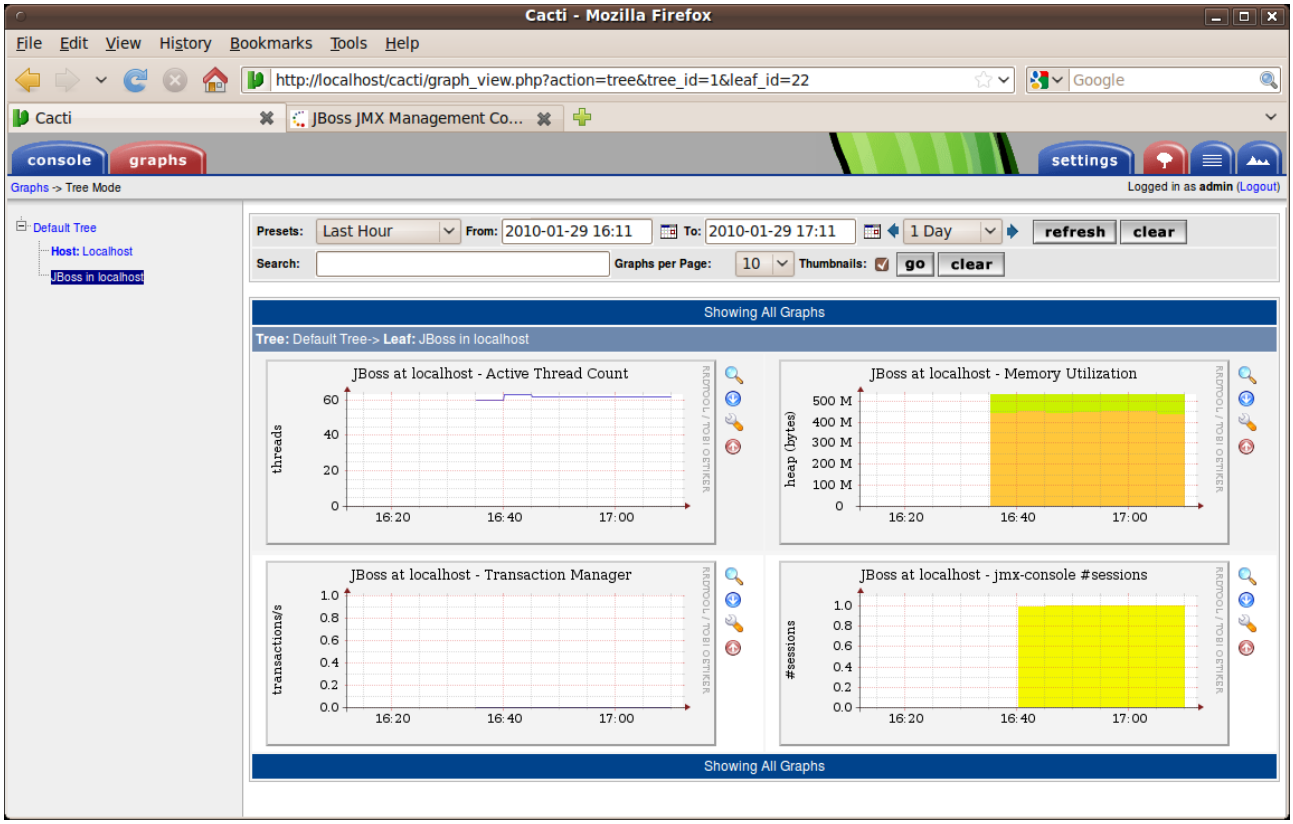


Cuando le damos al botón “create” han pasado tres cosas:

1. Se ha creado la definición del gráfico.
2. Se ha vinculado un gráfico de este tipo en el dispositivo JBoss.
3. Se ha creado un *datasource* de tipo “SNMP – Generic OID” que utiliza el *poller* para obtener los datos que generan el gráfico.

Ahora ya sólo falta añadir el gráfico a nuestro árbol de gráficos por defecto tal y como hicimos con el resto de gráficos que venían con la plantilla de JBoss.

Una vez lo hemos hecho, vemos nuestro gráfico justo al resto en la vista “graphs” tal y como muestra la siguiente captura.



¿Qué hemos visto?

A modo de resumen:

1. JMX es el protocolo de monitorización nativo de Java.
2. SNMP es un protocolo de monitorización de recursos en red.
3. Hay más herramientas SNMP que JMX y los profesionales de sistemas se sienten más cómodos con SNMP.
4. JBoss exporta de manera nativa su estado mediante JMX.
5. JBoss proporciona una aplicación que hace la traducción de JMX a SNMP (un agente SNMP) que se denomina “SNMP Adaptor”
6. Cacti es una poderosa herramienta de monitorización SNMP.
7. La comunidad ha creado plantillas para monitorizar en Cacti los aspectos básicos de JBoss (la configuración por defecto de la aplicación “SNMP Adaptor”).
8. Podemos configurar tanto JBoss como Cacti para monitorizar otros aspectos aspectos que nos interesen de nuestro servidor de aplicaciones o de nuestras aplicaciones.

Sobre el autor

Iván Párraga es ingeniero de software que puedes encontrar y contactar para cualquier cosa en:

- LinkedIn: http://es.linkedin.com/in/ivanparragagarcia
- Twitter: ivan_parraga
- e-mail: ivan.parraga.garcia@gmail.com
- blog: http://ivanator.wordpress.com/