MK Livestatus. Instalación independiente en Nagios.

Introducción

Ya hemos hablado en otros artículos sobre MK Livestatus. Hasta la aparición de MK Livedtatus la forma habitual de acceder al estado de los objetos de Nagios era o bien a través del fichero status.dat, o bien a través de una base de datos (NDO) que recibía periódicamente dichos datos mediante un módulo intermedio. Ambas opciones tenían bastantes pegas y hacían engorroso el tema de acceso a los datos de estado. MK Livestatus vino a cambiar la aproximación al problema. Crea un socket a través del cual se pueden obtener los datos en tiempo real bajo demanda. Rápido, sencillo, funcional, no consume apenas recursos y se pueden realizar la consultas con un lenguaje propio próximo al SQL. ¿Alguien da más? No. De hecho otras herramientas que hacen uso de acceso a datos como Nagvis, NagiosBP y por supuesto check_mk_multisite, rápidamente pasaron a usar MK Livestatus como el método preferido de acceso a estos.

Hasta ahora habíamos visto dos formas de disponer de MK Livestatus:

- Al instalar check_mk "full" instalará por defecto cmk_livestatus y configurará su integración con Nagios.
- Integrado en OMD. Instalando OMD cada nueva instancia que creemos integrará CMK Livestatus.

La tercera opción sería una instalación independiente exclusivamente del módulo para Nagios. Podemos necesitar instalar solamente MK Livestatus en un sistema con Nagios para poder acceder a sus datos desde una consola CMK Multisite o bien para poder acceder a los datos de Nagios desde herramientas externas mediante scripting p.e.

Instalación de CMK Livestatus.

Bajamos de la página de descargas de check_mk la última versión estable de MK Livestatus (no confundir con la versión entera de check_mk). El software viene empaquetado en un tar como "mk-livestatus-.*.tar.gz". No hay binarios disponibles. Hay que compilarlo.

Lo descomprimimos y compilamos

```
./configure
make
make install
```

Esto nos copia dos ficheros en nuestro sistema, normalmente en:

/usr/local/lib/mk-livestatus/livestatus.o (módulo de livestatus para cargar en Nagios)

/usr/local/bin/unixcat (Binario para acceder a los datos de Nagios usando el socket de Livestatus).

Editamos nuestro fichero de configuración de Nagios (nagios.cfg) y añadimos las líneas siguientes, al final normalmente:

```
broker_module=/usr/local/lib/mk-livestatus/livestatus.o   /usr/local/nagios/var/rw/live
event_broker_options=-1
```

Esto cargará un módulo para Nagios (livestatus.o) y generará un fichero de tipo socket de acceso a los datos de este en /usr/local/nagios/var/rw/live (en este caso).

El fichero socket se creará en el directorio indicado pero debemos tener cuidado de que el usuario Nagios pueda crearlo en dicha ubicación. Lo mejor es usar el directorio que usa ya Nagios para generar el fichero "nagios.cmd" que suele ser, en una instalación compilada, "/usr/local/nagios/var/rw/live". Si tenemos una instalación mediante binarios de la distro buscamos el directorio "rw" que estará normalmente en /var/nagios o similar.

Iniciamos Nagios y verificamos en el log de este que carga adecuadamente el módulo y que genera en el directorio indicado el fichero "live" (socket de acceso). Un ejemplo:

```
[1396806973] livestatus: Livestatus 1.x.x by Mathias Kettner. Socket: '/usr/local/nagios/var/rw/live'
[1396806973] Event broker module '/usr/lib/check_mk/livestatus.o' initialized successfully.
```

Probando Livestatus "local" para acceder a los datos.

En un sistema que esté ejecutándose y creado el fichero socket podemos probarlo de forma sencilla obteniendo todo el listado de hosts. Solo necesitamos el binario "unixcat" en el path, la ubicación del fichero socket que se genera al iniciar Nagios y cargar el módulo de livestatus, p.e.

```
# Obtener la lista de hosts
echo 'GET hosts' | unixcat /usr/local/nagios/var/rw/live
# O contándolos:
echo 'GET hosts' | unixcat /usr/local/nagios/var/rw/live | wc
    462  256433 2760716
```

En el próximo articulo seguiremos hablando del acceso a las distintas tablas de datos a través de MK Livestatus mediante el lenguaje específico LQL, de forma remota y el acceso mediante la API específica para Python