

NAEMON. Introducción e instalación.

Introducción.

Buenas noticias para el panorama “Nagios”. Tenemos un nuevo nacimiento :-). Fruto quizá de que Nagios es cada vez más “Enterprise” surge este nuevo “Fork” de Nagios Core que de entrada nos trae planteamientos interesantes. Si quieres saber más de quién está detrás del pequeño y porqué, tienes información interesante en <http://www.eldespistado.com/revolucion-comunidad-nagios/> enlace.

[Naemon](#) nace directamente de la versión Nagios Core 4.0.2 e incluye sus propios parches y mejoras. La promesa es hacerlo más abierto a la comunidad para que su desarrollo no esté solo en manos de una empresa e incorpore únicamente las funcionalidades que esta crea conveniente. Por supuesto OP5 tiene sus propios intereses es que esto sea así, de esta forma podrá sustituir el corazón de su producto de Nagios Core a Naemon y dotarle de funcionalidades que crea necesarias.

Las novedades más importantes de momento son:

- Se prescinde del interface GUI de Nagios Core. Se usa directamente Thruk.
- Se usa [Livestatus](#) de CMK para el acceso de Thruk a los datos de Nagios (Interesante, a fecha de hoy es un fork de Livestatus ya que CMK no ha actualizado oficialmente el broker de Livestatus para Nagios 4...).
- La rotación de LOGs se deja al sistema operativo (ya no se encarga el “Core”).
- Puedes echar un ojo a [todos los cambios](#).

Instalación.

La instalación la realizaremos usando los [repositorios](#) ubicados en [ConSol Labs](#) (ya tenemos otro involucrado interesante). Aunque está en “testing” tenemos los paquetes ya preparados para las distros más habituales. En nuestro caso probaremos inicialmente con los paquetes para RHEL6 en un CentOS 6.5. Instalamos previamente algunas utilidades varias que necesitaremos e instalamos apache si aún no lo tenemos instalado en nuestro sistema. Es importante porque si dejamos que se instale por dependencias (Thruk) puede pasar (y pasará) que se instale después de Thruk y este no encuentre el usuario y grupo de Apache para configurarse correctamente.

```
yum install updatedb vim wget yum-priorities httpd
```

Lo que no hace la instalación de Apache es esto último por lo que debemos configurar Apache para su inicio y verificamos que funcione correctamente.

```
chkconfig httpd on  
service httpd start
```

Será necesario configurar el repositorio de EPEL ya que algunas dependencias de los paquetes de Naemon no se encuentran en los repositorios de CentOS y si en estos.

```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
```

Desactivaremos SELinux mejor (al menos inicialmente para probar).

Interesa usar prioridades para los repositorios de tal forma que el de EPEL tenga baja prioridad. De esta forma se bajará preferentemente los paquetes de los repositorios oficiales y solo si no está en estos buscará en EPEL. Para ello hemos instalado el paquete yum-priorities

```
#Editamos /etc/yum.repos.d/epel.repo y añadimos en la sección de epel
add priority=10
```

Configuramos los repositorios Testing de ConSol Labs (de momento Naemon solo está en Testing).

```
rpm -Uvh "http://labs.consol.de/repo/testing/rhel6/x86_64/labs-consol-testing.rhel6.noarch.rpm"
```

Verificamos que nos ha añadido el repositorio en /etc/yum.repos.d/labs-consol-testing.repo e instalamos Naemon con sus dependencias:

```
yum install naemon
```

La instalación resuelve las dependencias y baja los paquetes necesarios de los repositorios de CentOS y EPEL.

Verificamos con “chkconfig –list” que ha instalado los demonios para Naemon y Thruk y los ha configurado para su inicio.

Arrancamos en este orden: httpd, naemon y thruk (ojo con iptables, habilitar el acceso a Apache)

Accedemos a la página de inicio: http://nuestro_host/naemon (thrukadmin / thrukadmin) y ya tenemos funcionando la instalación. Podían haberse ahorrado los tan manidos ejemplos de serie. Queda feo entrar en ver ya errores por todos los lados :-). Ahora vamos con los plugins ya que si nos lo encuentra no funcionará...

The screenshot shows the Thruk web interface. On the left is a navigation menu with sections like General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, and Problems. The main content area is divided into 'Current Network Status' and 'Service Status D'. The 'Current Network Status' section shows the last update time, update frequency, and version information. Below it are links for viewing history and notifications. The 'Service Status D' section features a table with columns for Host, Service, Status, Last Check, Duration, Attempt, and a final status column. Several services are marked as 'CRITICAL'.

Host	Service	Status	Last Check	Duration	Attempt	
hpl2605dn	PING	CRITICAL	20:13:07	0d 4h 11m 33s	3/3	CRITICAL
	Printer Status	CRITICAL	20:05:28	0d 4h 12m 21s	3/3	(No output)
linksys-srw224p	PING	CRITICAL	20:10:28	0d 4h 10m 51s	3/3	(No output)
	Port 1 Bandwidth Usage	CRITICAL	20:05:29	0d 4h 9m 21s	3/3	(No output)

Plugins.

Necesitamos instalar los plugins. Tenemos dos opciones: bajar y compilar los [Plugins originales de monitorización](#) que mantiene un grupo aparte. O bien usar los de EPEL ya compilados y listos para instalar. Optamos por esta última opción.

```
yum install nagios-plugins-all
```

Cambiamos en `/etc/naemon/resource.cfg` la ruta de nuestros plugins ya que está apuntando a un directorio Naemon que no existe de momento. Podemos optar por otras soluciones si nos apetece (copiar, enlazar,...):

```
$USER1$=/usr/lib64/nagios/plugins
```

La instalación del paquete nos ha creado también un usuario y grupo nagios que no necesitamos. Pero mejor no lo borramos y lo dejamos desactivado ya que cuando actualizamos los plugins de nuestra distro lo volverá a crear. ¿Tendremos un usuario naemon algún día? :-)

Una vez instalado reiniciamos naemon y thruk vemos ya que los plugins para el host localhost que venía por defecto están ya funcionando. OJO. Si iniciamos Thruk inmediatamente después de reiniciar Naemon puede ser que este último aún no haya creado el socket en `"/var/cache/naemon/live"` y no inicie.

localhost	Current Load	OK	OK - load average: 0.01, 0.05, 0.03	
	Current Users	OK	USERS OK - 3 users currently logged in	
	HTTP	OK WARNING	HTTP WARNING: HTTP/1.1 403 Forbidden - 5237 bytes in 0.003 second response time	
	PING	OK	PING OK - Packet loss = 0%, RTA = 0.13 ms	
	Root Partition	OK	DISK OK - free space: / 15145 MB (90% inode=96%):	
	SSH	OK OK	SSH OK - OpenSSH_5.3 (protocol 2.0)	
	Swap Usage	OK	SWAP OK - 100% free (2047 MB out of 2047 MB)	
	Total Processes	OK	PROCS OK: 114 processes with STATE = RSZDT	

Livestatus y check_mk

Como comentaba es un placer contar de inicio ya con Livestatus ya que numerosas aplicaciones lo usan para acceder a los datos de Nagios (PNP4Nagios, Nagvis, el mismo Thruk) y me parece imprescindible para que Naemon progrese. De momento solo le veo un problema. El livestatus que incluyen es un fork del original de check_mk modificado para que el broker funcione para Nagios Core 4 / Naemon. ¿Porque un fork? Básicamente porque parece que CMK no tiene aún un Livestatus para Nagios 4 y la gente de Naemon (OP5) consciente de la importancia se ha encargado del tema. ¿Funciona check_mk con el fork de Livestatus para Nagios 4 / Naemon? Pues me temo que no. Tendremos que esperar a que cmk actualice su Livestatus. Esperemos que se sume al lado "oscuro". De hecho OMD del que ya hemos hablado y visto su instalación y configuración en otros artículos... está preparando una versión con Livestatus y Nagios 4 que [podemos encontrar en los repositorios de ConSol Labs](#). Dado que CMK está detrás de la [versión comercial de OMD](#) con soporte y actualizaciones es lógico que se así pero esperemos que [OMDDistro](#) siga con nosotros y no quedé solo en la versión comercial con soporte.