

# Nagios. Introducción a los plugins estándar.

## **Introducción.**

Existen varias formas de realizar chequeos para Nagios. Desde clientes específicos que se instalan en los Sistemas Operativos a monitorizar: vimos en un [artículo anterior](#) el cliente de **check\_mk** pero están también entre otros, clientes específicos para Windows como [nsclient++](#) y para Linux como [NRPE](#), a otros métodos como obtener los datos del equipo remoto por snmp o conectándonos a este para ejecutar un programa mediante ssh. En este caso vamos a echar un vistazo a los [plugins estándar de Nagios](#).

Los plugins estándar de Nagios que están o deberían estar en cualquier instalación de Nagios no son parte del programa de Nagios y no vienen con este, son paquetes aparte mantenidos por otro grupo diferente. Si hemos realizado nuestra instalación siguiendo artículos previos, mediante [OMD Distro](#) o mediante una instalación de paquetes de [Nagios de nuestra distribución](#) los tendremos ya instalados.

Los Plugins “oficiales” usados por Nagios tienen un diseño sencillo y unas especificaciones detalladas para que “cualquiera”, a partir de dichas especificaciones, pueda programar sus propios Plugins. De forma resumida:

- El Plugin debe retornar un valor numérico como resultado de su evaluación: 0 (OK) 1 (Warning), 2 (Critical), 3 (Unknow).
- El Resultado del Plugin debe ser una cadena de texto que será la mostrada en el interface Web.
- Otra cadena de texto (separada por “|” de la principal) puede añadirse opcionalmente para proporcionar datos de rendimiento. Generalmente se usa para realizar gráficas con programas de terceros.
- Se debe seguir un estándar específico para pasarle los diferentes valores al Plugin (Host, Intervalos de “Warning” y “Critical”, Comunidad SNMP, URL, ...)
- Cumpliendo dichas especificaciones se puede usar cualquier lenguaje de programación o script para realizar los Plugins. Obviamente los lenguajes de Script o interpretados penalizan más el rendimiento del sistema que el uso de lenguajes compilados como C. De ahí la razón de que los Plugins oficiales estén hecho en C.

Siguiendo estas directrices cualquier programa, compilado o interpretado que cumpla al menos las dos primeras puede ser un plugin. Puedes encontrar toda la información necesaria en la documentación oficial de estos y más en concreto de parámetros a recibir, rangos de valores, ... en su página de [documentación de desarrollo](#).

En nuestro caso, los plugins oficiales siguen estas directivas por lo que nos encontraremos que los parámetros que aceptan siguen los estándar en todos los casos.

## Clasificando los plugins.

Para entender un poco mejor los plugins estándar podemos hacer una pequeña clasificación con ellos. Su ubicación habitual será en “/usr/lib/nagios/plugins” o similar. En OMD estarán en: “/opt/omd/versions/default/lib/nagios/plugins”.

### Plugins “locales”

Podemos denominar como locales a los plugins que necesitan ejecutarse en el propio servidor del que queremos obtener datos. ¿Qué sentido tiene entonces que estos plugins estén en el servidor de Nagios? Ninguno, salvo su utilidad para chequear aplicaciones locales en el propio servidor de Nagios. ¿Y cómo se llaman desde Nagios para su ejecución en sistemas remotos? Pues tenemos varias opciones, entre ellas usar el plugin para ejecución remota por ssh (`check_by_ssh`) o usar [NRPE](#) (Nagios Remote Plugin Executor). En cualquier caso necesitaremos instalar también los plugins de Nagios en el servidor remoto. En el caso de NRPE necesitaríamos:

- Plugin de nagios `check_nrpe`. Estará presente en el directorio de plugins de este. Se puede instalar como paquete de la distribución. En OMD está ya presente.
- Demonio NRPE en el servidor Linux/Unix remoto.
- Plugins “locales” de Nagios en el servidor remoto.
- Fichero de configuración en el servidor remoto.

En otro artículo veremos detenidamente ambas opciones `by_ssh` y NRPE. ¿Que opción es mejor? Pues como siempre... depende. `By_ssh` carga más ambos sistemas que NRPE pero nos puede dar igual. Además NRPE podemos usarlo también para chequear terceros equipos desde el servidor remoto en nombre de Nagios. Puede ser interesante para equipos en otras redes, dmz,..., aunque para hacer esto también hay otras opciones .

¿Y para chequear sistemas Windows? Disponemos de otro plugin “local”, `check_nt`, que se usará para estos sistemas. En el Windows remoto deberemos también disponer de un... servicio que responda, unos ficheros de configuración y unos plugins. [Nsclient++](#) p.e. nos aportará la parte necesaria en el servidor Windows, aunque no es el único cliente de este tipo.

Ejemplos de plugins locales: `check_users`, `check_disk`, `check_apt`, `check_mailq`, `check_procs`,...

### Plugins “de red”

Estos plugins son los que nos proporcionan el chequeo de un servicio (o puerto) en un servidor remoto usando solo la comunicación por red con este. Algunos se limitan a chequear algo tan básico como un puerto abierto, otros buscan repuestas concretas conectando al socket y otros más elaborados nos requerirán incluso usuario y contraseña y/o acciones concretas a consultar en el servicio remoto.

Ejemplos de este tipo de plugin son: `check_tcp`, `check_udp`, `check_ping`, `check_http`, `check_mysql`, `check_radius`, `check_snmp`, `check_pop`, ...

### Plugins “especiales”

Podemos englobar en plugins especiales los que necesitan un “colaborador” y/o plugins en el equipo remoto a chequear. Ya los hemos mencionado: `check_nt`, `check_nrpe`, `check_by_ssh`,...

En futuros artículos iremos desgranando el uso de algunos de los plugins mencionados así como el uso de NRPE, NSClient.

Además de los plugins estandar de Nagios existen muchos más plugins. Hablamos del tema en nuestro [siguiente artículo](#).

