Nagios. Introducción a los objetos. (II)

Continuando con el articulo previo seguimos echando un vistazo al resto de objetos de Nagios.

contact

El objeto contact se usa en Nagios para definir las propiedades del contacto que usarán otros objetos para alertar / enviar información de lo que está pasando. Lo normal es que en lugar de usar directamente un contacto se use un grupo de contactos (contactgroup), pero previamente tenemos que crear estos.

En nuestra instalación por defecto de Nagios podemos encontrar un ejemplo de definición para el objeto contact.

```
define contact{
contact_name                    root
alias                           Root
service_notification_period     24x7
host_notification_period        24x7
service_notification_options    w,u,c,r
host_notification_options       d,r
service_notification_commands   notify-service-by-email
host_notification_commands      notify-host-by-email
email                           root@localhost
}
```

Vemos que tiene las directivas habituales xxx_name, alias y que a su vez está referenciando a un montón de objetos más de tipo t**imeperiod** (24×7) que veremos en breve y de tipo **command** (notify-…). Muy importante la directiva email que será el correo final al que tendrá que mandar los avisos.

Como comentábamos, los objetos en Nagios tienen una fuerte dependencia unos de otros. En este caso, este objeto contacto está referenciando mediante directivas a otros cuatro objetos.

Podemos obtener más información en el manual online de Nagios, es importante sobre todo tener en cuenta (marcadas en rojo en este) las directivas obligatorias.

contactgroup

La forma óptima de configurar a quien alertar será mediante un grupo de contactos ya que nos permite más flexibilidad a la hora de quitar / poner integrantes (contacts). Un ejemplo de la instalación por defecto:

```
define contactgroup{
contactgroup_name admins
```

```
alias Nagios     Administrators
members          root
}
```

Sencillo. Nombre, alias y la directiva “members” donde listaremos separados por comasnuestros objetos “contact“.

timeperiod

Periodo de tiempo. Nagios nos da mucha, mucha potencia en cuanto a la gestión de tiempos y escalado de alertas. Un ejemplo de nuestra instalación:

```
define timeperiod{
timeperiod_name     24x7
alias               24 Hours A Day, 7 Days A Week
sunday              00:00-24:00
monday              00:00-24:00
tuesday             00:00-24:00
wednesday           00:00-24:00
thursday            00:00-24:00
friday              00:00-24:00
saturday            00:00-24:00
}
# Here is a slightly friendlier period during work hours
define timeperiod{
timeperiod_name     workhours
alias               Standard Work Hours
monday              09:00-17:00
tuesday             09:00-17:00
wednesday           09:00-17:00
thursday            09:00-17:00
friday              09:00-17:00
}
```

La sintaxis es aparentemente sencilla (en el ejemplo) pero ciertamente puedes complicarla todo lo que quieras o necesites: El nombre, el alias y todos los días de la semana en inglés poniendo en cada uno de ellos los rangos de horas activos en formato 24h (en su forma sencilla). Si no ponemos un día lo estaremos excluyendo. ¿Cuál es la parte complicada? No solo tenemos la posibilidad de especificar días y su intervalo de horas, podemos especificar años, un día específico de la semana, un rango de días del mes,… prácticamente lo que se nos ocurra. Para profundizar en el tema los ejemplos de la documentación en línea de Nagios al respecto son estupendos. Como ejemplo:

Estos objetos de Periodo de tiempo los podremos usar para definir en otros objetos en que intervalos aparecerán las alertas en el interface gráfico, mandará un correo a contacto o a grupo de contactos, …

El resto de objetos relativos al escalado de alertas y a dependencia de hosts y servicios son ya de un articulo avanzado y los veremos más detenidamente en un futuro.