

# Iconos en Nagios (y check\_mk).

## Introducción.

Nagios incluye la posibilidad de añadir iconos a nuestros objetos Host y/o Servicios. Los iconos nos solo valen para que nuestro interface quede más vistoso y nos sea más sencillo localizar los equipos. Existen también otros usos interesantes destinados a un fin más práctico. Por ejemplo, en los iconos de servicios podemos distinguir la importancia de estos con iconos que marquen o distingan el uso y/o criticidad de estos. A mi me gusta poner al menos en los servicios solo informativos (versión agente nsclient++/nrpe, uptime, ...) el icono de la "I" informativa en azul para que sea más visual su cometido.

Otro uso interesante donde funcionan equipos de soporte cuya primera herramienta de alertas en Nagios, es asignar a los servicios en función de su criticidad distintos iconos con letras/colores que representen el SLA o la criticidad en la respuesta (N1, N2, N3,...). De esta forma de un vistazo se puede establecer el protocolo de actuación en función de la asignación del servicio que falla.

## Donde se configuran los iconos en Nagios (directivas de objetos hosts / servicios)

Los iconos se configuran como directivas de Host y/o Servicios de Nagios. Las directivas que definen que iconos usar son las siguientes:

### # Hosts

```
icon_image    image_file
icon_image_alt alt_string
vrmf_image    image_file
statusmap_image    image_file
```

### # Servicios

```
icon_image    image_file
icon_image_alt alt_string
```

Según el manual de Nagios relativo a estas directivas:

- `icon_image` → Define el nombre del archivo de imagen GIF, PNG y JPG que se asociarán con este host y se visualizará en varios lugares en los CGIs. La imagen debe ser preferentemente de 40x40 pixeles
- `icon_image_alt` → Texto alternativo para la Imagen para el TAG alt. No necesario.
- `vrmf_image image` → Define el nombre del archivo de imagen GIF, PNG y JPG que se asociarán con este host y se visualizará exclusivamente en el CGI statuswrl. Esto realmente es una rareza que nunca usé ni vi usar a nadie aunque lo probé en su día. Ese CGI nos muestra un "status map" pero en 3D tipo Realidad Virtual. En su día el standard VRML parecía que sería algo prometedor pero se quedó en nada, al menos para este tipo de representación. Necesitas un visor especial integrador con tu Navegador y el efecto conseguido es decepcionante. No pierdas el tiempo.
- `statusmap_image` - > Define el nombre del archivo GD2 de imagen que se asociarán con este host y se visualizará exclusivamente en StatusMap. Puedes usar también archivos GIF,

PNG o JPG pero no se recomienda ya que consumirá mucha CPU al generar la vista de estado. La imagen debería ser para que se integre perfectamente de 24x24. ¿Usas las vista de Status Map? Si es así te merecerá la pena tener tus iconos al día. Con instalaciones grandes, a no ser que te coincida una arquitectura genialmente jerárquica de padres / hijos... no tiene mucha utilidad en mi opinión. Si quieres tener una representación visual de tu arquitectura... usa Nagvis. Requiere mucho más trabajo pero si te lo curras el resultado puede ser espectacular.

## **Ubicación de los ficheros de iconos.**

Donde podemos localizar los iconos dependerá de como hayas instalado Nagios. Tienes que localizar el directorio “share” donde están los recursos que usa el interface web. En una instalación compilada estarán normalmente en:

*/usr/local/nagios/share/images/logos*

En una instalación de **check\_mk** normalmente el directorio será el correspondiente a Nagios en:

*/usr/share/check\_mk/web/htdocs/images/icons/*

Si usas **check\_mk con OMD** la ruta será normalmente (relativa a tu site en OMD):

*~/local/share/check\_mk/web/htdocs/images/icons*

## **Paquetes de iconos.**

Nagios trae solo algunos iconos concretos que forman parte del GUI pero no incluye de serie iconos para nuestro servicios / hosts, debemos localizarlos en terceros y copiarlos al directorio de iconos para poder usarlos.

Sin duda el mejor paquete de iconos es el [F\\*Nagios icon pack](#) que podemos bajar de [monitoringexchange](#)

Tenemos también disponibles otros paquetes menos completos en [Nagiosexchange](#)



## Ejemplos de configuración para Nagios.

Es muy cómodo usar plantillas preconfiguradas para la definición de iconos y posteriormente asignárselas a nuestros hosts para que hereden de estas. Un ejemplo:

```
# Plantilla iconos Linux
define host{
    name                linux-icons
    icon_image          linux.png
    icon_image_alt      Linux_Server
    statusmap_image     linux.gd2
    register            0
}

# Plantilla iconos Windows
define host{
    name                windows-icons
    icon_image          windows_server.png
    icon_image_alt      Windows Server
    statusmap_image     windows_server.gd2
    register            0
}
```

Ejemplo de definición de host Windows heredando la plantilla específica con las directivas de iconos.

```
define host{
    use                generic-host,windows-icons
    host_name          SRVSQLSRV
    alias              SQL Server
    parents            Router1
    address             192.168.1.14
}
```

Una vez que le asignemos los iconos a nuestros hosts veremos que la añadimos una nota de color a nuestro Nagios cuando nos aparezcan los iconos al lado de nuestro Hosts.

VMWare-ESX1	EstadoSistema -Hardware-	OK	Compaq/HP Agent Check: overall system state OK
	PING	OK	PING OK - Packet loss = 0%, RTA = 0.82 ms
	VMware CPU Usage	OK	CHECK_ESX3 OK - cpu usage=1.45 %
	VMware IO Reads	OK	CHECK_ESX3 OK - io read latency=3 ms
	VMware IO Write	OK	CHECK_ESX3 OK - io write latency=0 ms
	VMware Mem Usage	OK	CHECK_ESX3 OK - mem usage=51.97 %
	VMware NET Usage	OK	CHECK_ESX3 OK - net usage=0.00 KB/s
	VMware Runtime Issues	OK	CHECK_ESX3 OK - No issues
	VMware Runtime Status	OK	CHECK_ESX3 OK - overall status=green
	VMware Swap Usage	OK	CHECK_ESX3 OK - swap usage=0.00 MB
VMWare-ESX2	EstadoSistema -Hardware-	OK	Compaq/HP Agent Check: overall system state OK
	PING	OK	PING OK - Packet loss = 0%, RTA = 0.25 ms
	VMware CPU Usage	OK	CHECK_ESX3 OK - cpu usage=10.20 %
	VMware IO Reads	OK	CHECK_ESX3 OK - io read latency=0 ms
	VMware IO Write	OK	CHECK_ESX3 OK - io write latency=0 ms
	VMware Mem Usage	OK	CHECK_ESX3 OK - mem usage=41.90 %
	VMware NET Usage	OK	CHECK_ESX3 OK - net usage=0.00 KB/s
	VMware Runtime Issues	OK	CHECK_ESX3 OK - No issues
	VMware Runtime Status	OK	CHECK_ESX3 OK - overall status=green
	VMware Swap Usage	OK	CHECK_ESX3 OK - swap usage=0.00 MB

## Ejemplos de configuración para Nagios + Check\_mk.

Para configurar unos templates para los iconos en check\_mk usaremos la sintaxis de este. Definimos configuraciones extra de Host para la directiva "icon\_image" y asignamos el icono concreto que queremos a los hosts con el tag asociado. De esta forma con unas pocas líneas asignamos iconos a todos nuestros hosts.

```
extra_host_conf["icon_image"] = [
    ("redhat.png", ["linux"], ALL_HOSTS ),
    ("windows.png", ["windows"], ALL_HOSTS ),
    ("vmware.png", ["esxi"], ALL_HOSTS ),
    ("hp.png", ["hp"], ALL_HOSTS ),
]
```

Para saber más de check\_mk, su sintaxis, tags,... puedes ver el artículo correspondiente .

Así de mono (lamento los tachones) quedara unos de nuestro grupos de Hosts con los iconos de cada Host en el interface de CMK. Servidores vSphere, Redhat y cabina de HP.

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	vmware1		vmware1	8	1	0	1	0
UP	vmware2		vmware2	10	0	0	0	0
UP				31	0	0	0	0
UP	eth0			1	0	0	0	0
UP	eth1			1	0	0	0	0
UP				21	0	0	0	0

## Creación de iconos.

Si no encuentras el icono siempre podrás crearlo con alguna herramienta para estos. Una vez creado el fichero base en GIF puedes convertirlo directamente al resto de formatos con utilidades Linux. La imagen GIF debería ser de 40x40 pixels.

Hay instalar el paquete netpbm, que es el que trae las herramientas de conversión a otros formatos. También las utilidades de GD (para el icono gd2).

```
# Debian
apt-get install netpbm libgd-tools
#CentOS/Redhat?
yum install netpbm-progs gd-progs
```

El script que hay usar para convertir (crear fichero run.sh p.e.)

```
path="/usr/bin"
for arg
do
  if [ -f "$arg" ]; then
    echo converting $arg
    arg="$(echo $arg | sed 's/\.gif$//')".png
    $path/giftoptopnm $arg.gif > $arg.pnm
    $path/pnmtopng -transparent rgb:ff/ff/ff $arg.pnm > $arg.png
    $path/pnmtjpeg -quality=100 -optimize -smooth=0 $arg.pnm > $arg.jpg
    $path/pngtogd2 $arg.png $arg.gd2 0 1
  fi
done
rm -f *.pnm
```

Ejecutamos: run.sh ARCHIVO.GIF

Podemos probar bajándonos el icono 40x40 del “muro” :- ) en [esta página](#) por ejemplo y probar el script. Lo ideal es que los gif que hagamos sean transparentes para que queden mucho mejor. Si te animas buscando algún software para usar y/o manipular GIFs. Todos llevamos un artista dentro :- )