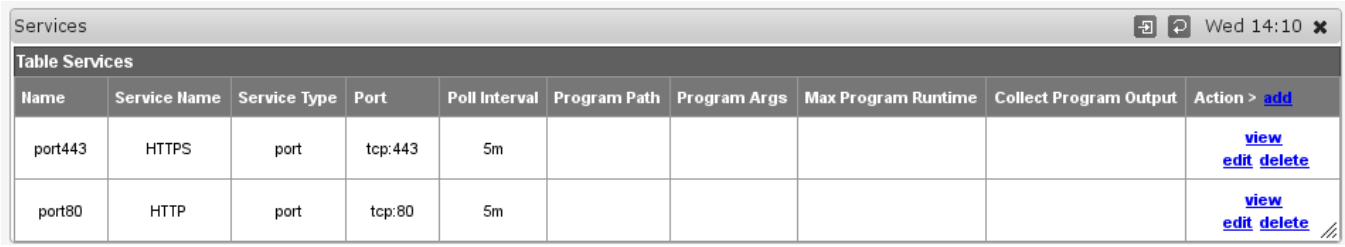
Ejemplos de Servicios Monitoreados

Web

Remoto, solo puerto

NMIS puede monitorear la accesibilidad de los puertos TCP (usando la herramienta NMAP) y un servicio web que le dirá si el servidor es accesible (pero no si está funcionando completamente). Sin embargo, este tipo de monitoreo no requiere que se ejecute ningún software en el servidor de destino.

Aquí hay un fragmento de configuración para este nivel de monitoreo, para los puertos web estándar 443 y 80, que activaría para el servidor que desea probar:



Services — Wed 14:10

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
port443	HTTPS	port	tcp:443	5m					view edit delete
port80	HTTP	port	tcp:80	5m					view edit delete

Proceso del servidor

Si SNMP está habilitado para el sistema en cuestión, si NMIS está colectando información de ese sistema y, si el sistema y su modelo admiten la MIB de recursos de host, entonces NMIS puede verificar los estados del proceso y verificar la existencia de un proceso específico. El tipo de servicio debe ser `"service"`, el nombre del proceso se debe proporcionar como Service name y debe activar el servicio para el nodo que desea probar.

Para un nodo con CentOS y Apache 2.2.x, estaríamos usando la siguiente definición de servicio, que busca procesos llamados `"httpd"`:

Services — Wed 13:57

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
HTTP_Daemon	httpd	service		5m					view edit delete

Monitoreo basado en script SAPI

NMIS también puede realizar una cantidad limitada de interacción con un servicio basado en TCP utilizando scripts SAPI. Los scripts de ejemplo para POP3 y HTTP básico se encuentran en NMIS en la carpeta `/usr/local/nmis8/conf/scripts`. El script `http` default, conecta al servidor web en cuestión e intenta descargar la URL del índice raíz `"/"`; si esta solicitud tiene éxito o devuelve una redirección HTTP, se considera que el servicio está bien.

Para habilitar este tipo de supervisión, debe definir el servicio con el nombre que coincida con el nombre del archivo de secuencia de comandos. El nombre del servicio puede ser un texto de su elección, pero el tipo de servicio debe ser `"script"`, y debe activar ese servicio para el nodo con el que desea comunicarse:

Services Wed 14:17

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
http	Basic HTTP Interaction	script	80	5m					view edit delete

Monitoreo end-to-end usando un programa personalizado

Si necesita una interacción más precisa con su servicio web que la que pueden proporcionar los scripts SAPI (por ejemplo, SSL / TLS o cookies o similares), necesitará utilizar un script personalizado. NMIS 8.5.4G contiene un script de ejemplo de ese tipo: `/usr/local/nmis8/install/scripts/webtest`, que debe moverse a un directorio destinado a binarios (por ejemplo, `/usr/local/nmis8/bino /usr/local/bin/`) si desea utilizarlo.

El script de ejemplo, descarga una página web (opcionalmente siguiendo una serie de redirecciones) usando http o https y, opcionalmente, verifica que el contenido del documento coincida con una expresión regular determinada. Debe definir este servicio con el Tipo de servicio `"program"`, proporcionar la configuración del programa adecuada para el programa y activar el servicio para el servidor que desea probar (pero tenga en cuenta que el programa personalizado siempre se ejecutará *localmente* en su servidor NMIS).

Así es como verificamos que el sitio web de Opmantek esté funcionando: esto descarga la página usando https, luego busca la frase "Opmantek Products":

Services Wed 14:27

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
opmantek-site-ssl	opmantek-site-ssl	program		5m	/usr/local/bin/webtest	-c 'opmantek.*products' https://opmantek.com/	10	false	view edit delete

DNS

Remoto, solo puerto

NMIS puede monitorear la accesibilidad de los puertos TCP y UDP (usando la herramienta NMAP), que en el caso de DNS daría solo una indicación aproximada de si el servidor DNS es accesible.

A continuación, se muestra un fragmento de configuración para este nivel de monitoreo:

Services Wed 13:13

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
port53	DNS port reachability	port	udp:53	5m					view edit delete

Remoto, solo protocolo

Para verificar el funcionamiento general de un servidor DNS remoto, puede utilizar el servicio 'dns' integrado en NMIS. Este servicio realizará una solicitud de DNS al servidor en cuestión y luego activará alertas de interrupción en función de recuperar o no un registro de DNS (y también captura el tiempo de respuesta).

Así es como se configura nuestro propio monitoreo interno para verificar nuestro propio dominio, que involucra servidores fuera de nuestro control: hemos definido nodos con el modelo establecido estáticamente en "PingOnly" para los servidores DNS externos en cuestión, y el servicio activado "opmantek-dns" para ellos, que se ve así:

Services Wed 13:17

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
opmantek-dns	opmantek.com	dns		5m					view edit delete

Tenga en cuenta que el modelo "PingOnly" por sí solo no es suficiente para deshabilitar los accesos SNMP (o WMI); también debe cambiar la opción de configuración del nodo `collect` a false.

Secuencia de comandos local y personalizado

En un sistema que está bajo su control y que ejecuta NMIS, puede ejecutar scripts arbitrarios para recopilar estados de servicio. La secuencia de comandos de ejemplo a continuación verifica que el servidor NMIS local en sí tenga un proceso de servidor DNS BIND en ejecución:

```
#!/bin/sh
# small script that tests that a local bind is up and communicating
if /sbin/pidof named >/dev/null 2>&1 && /usr/sbin/rndc status | grep -q 'up and running'; then
        exit 100
else
        exit 0
fi
```

Para usar esto, guarde el script en algún lugar donde NMIS pueda acceder (como `/usr/local/bin/bindpresent` por ejemplo), luego configure NMIS con este servicio de tipo "program" y active el servicio para el propio servidor NMIS:

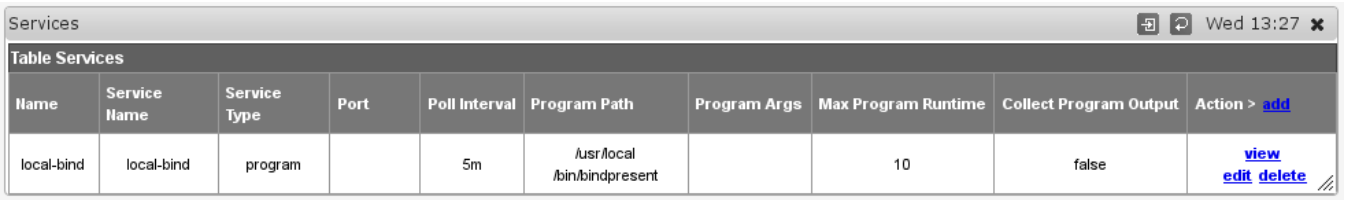

Services Wed 13:27

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
local-bind	local-bind	program		5m	/usr/local /bin/bindpresent		10	false	view edit delete

Base de datos MySQL

Remoto, solo puerto

Para verificar que su servidor de base de datos MySQL sea accesible, puede definir un servicio para verificar el puerto TCP 3306, similar a los ejemplos anteriores. Naturalmente, esa no es una prueba end-to-end.

Remoto, estado del proceso del servidor

Además de la accesibilidad del puerto, puede definir un servicio para verificar la existencia del proceso `mysqld`, si está colectando información del servidor en cuestión con SNMP:

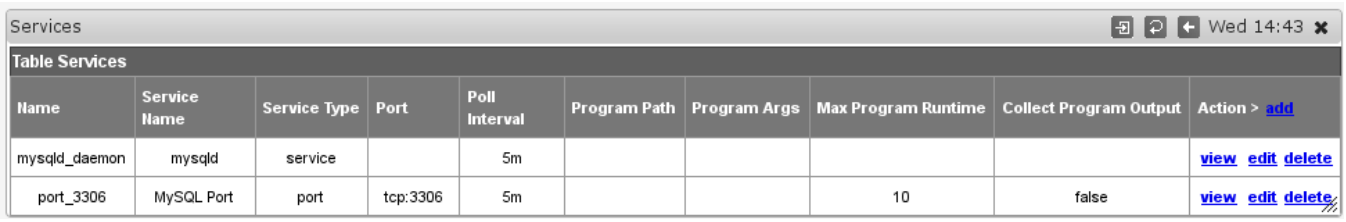

Services Wed 14:43

Table Services

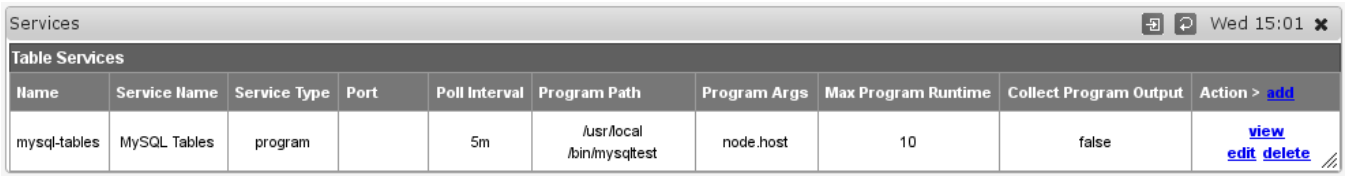
Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
mysqld_daemon	mysqld	service		5m					view edit delete
port_3306	MySQL Port	port	tcp:3306	5m			10	false	view edit delete

Secuencia de comandos remota y personalizada

La tercera y más completa configuración de monitoreo end-to-end requeriría un pequeño script personalizado que realmente se conecta al servidor y realiza una consulta en dicho servidor. Aquí hay un ejemplo de dicho script, que debería ajustarse para su entorno (o cambiarse para aceptar más argumentos de línea de comando) y guardarse como `/usr/local/bin/mysqltest`:

```
#!/bin/sh
# a small wrapper around the mysql client, which connects to a test database
# and runs show tables; if successful (and there are tables) we call it good
NODE=$1                                                 # passed in, comes from node.host
DBUSER="mytest"
DBPASSWORD="something secret"
DBNAME="testdb"
OUTPUT=`mysql -u$DBUSER -p$DBPASSWORD -h$NODE $DBNAME -e "show tables;"`
if [ $? != 0 ]; then
        exit 0                                          # service bad
elif ! echo "$OUTPUT" | grep -q "Tables_in_"; then
        exit 50;                                        # service not fully ok
else
        exit 100;                                       # service good
fi
```

Para utilizar esta prueba de servicio, definiría un servicio de Tipo de servicio `"program"`, con una ruta de programa adecuada y con los argumentos de programa configurados en `"node.host"`, que se reemplazaría por la dirección o el nombre de host del nodo en cuestión:



Estado del UPS

Scripts personalizados

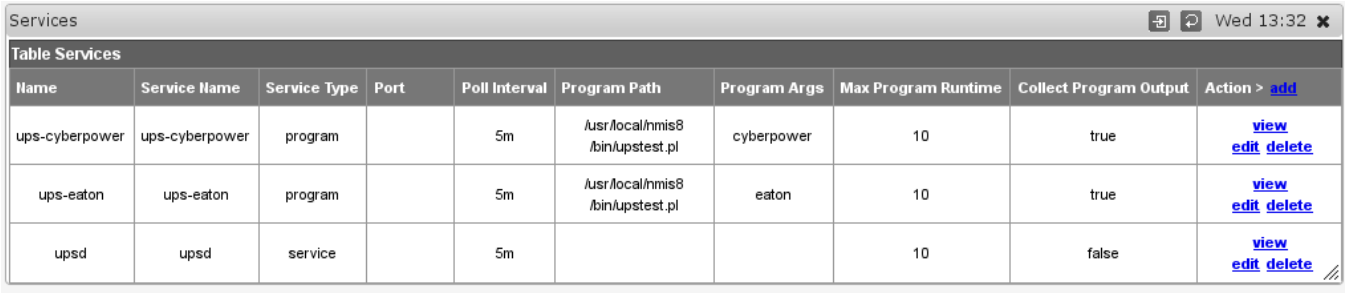
Los sistemas UPS más baratos que no tienen redes integradas o capacidades SNMP también pueden ser monitoreados por NMIS, siempre que haya algún tipo de infraestructura de administración que admita la consulta del estado del UPS. En este ejemplo, estamos verificando dos sistemas UPS que están conectados a nuestro servidor NMIS a través de cables USB, donde la suite NUT (Network UPS Tools) se encarga de la interconexión.

El `upstest.pl` siguiente script usa las herramientas NUT para consultar el UPS nombrado e informa si está funcionando y a qué nivel de carga se encuentra. (NMIS aún no grafica variables adicionales como el nivel de carga aquí a partir de la versión 8.5.4G, pero esta función se agregará pronto).

```
#!/usr/bin/perl
# a tiny wrapper around upsc to integrate with nmis
# exits with 100 if ups online, charge otherwise
# this means the service is down only when the ups is dead,
# NOT while its discharging.
# also reports battery charge as charge=NNN
use strict;
# args: name of the ups
my $upsname = $ARGV[0];
die "usage: $0 <upsname>\n" if (!@ARGV);
my @knownones = `upsc -L 2>/dev/null`;
die "unknown ups $upsname\n" if !grep (/^$upsname:/, @knownones);
my ($status,$charge);
for my $line (`upsc $upsname 2>/dev/null`)
{
                chomp $line;
                my ($varname,$value) = split(/\s*:\s*/, $line);
                if ($varname eq "ups.status")
                {
                                $status = $value;
                }
                elsif ($varname eq "battery.charge")
                {
                                $charge = $value;
                }
}
print "charge=$charge\n" if (defined $charge);
exit ($status =~ /^OL/? 100 : $charge);
```

Para nuestros sistemas UPS, primero utilizamos el monitoreo de estado del proceso basado en SNMP incorporado de NMIS, que verifica que haya al menos un proceso activo con un nombre dado (aquí 'upsd'), y luego agregamos las verificaciones de estado por UPS con los nombres de UPS pasaron al script de prueba. Esta configuración de ejemplo requiere que los UPS estén conectados al servidor NMIS mismo, pero, por supuesto, se puede acceder a NUT a través de la red.

Aquí está nuestra definición del servicio:



Services — Wed 13:32

Table Services

Name	Service Name	Service Type	Port	Poll Interval	Program Path	Program Args	Max Program Runtime	Collect Program Output	Action > add
ups-cyberpower	ups-cyberpower	program		5m	/usr/local/nmis8 /bin/upstest.pl	cyberpower	10	true	view edit delete
ups-eaton	ups-eaton	program		5m	/usr/local/nmis8 /bin/upstest.pl	eaton	10	true	view edit delete
upsd	upsd	service		5m			10	false	view edit delete