



# Nagios<sup>®</sup>

## Monitoreo de la red con Nagios – PARTE IV

### Descripción

#### PARTE IV

##### |\_8. Monitorización en Nagios

- |\_8.1. Configuración de los chequeos de Nagios al localhost
- |\_8.2. Configuración de los chequeos de Nagios a hosts Linux
- |\_8.3. Configuración de los chequeos de Nagios a los Hypervisors
- |\_8.4. Configuración de los chequeos de Nagios a dispositivos de red
- |\_8.5. Configuración de los chequeos de Nagios a hosts Windows



|\_ \_8.6. Mostrando los resultados de los chequeos

## 8. Monitorización en Nagios

En este apartado se definirán las plantillas de configuración necesarias para monitorizar nuestra red. Ellas son las siguientes:

- `localhost.cfg`: Plantilla de configuración para monitorizar los servicios del propio servidor Nagios.
- `hypervisor.cfg`: Plantilla de configuración para monitorizar los servicios de las plataformas de virtualización.
- `linux.cfg`: Plantilla de configuración para monitorizar los servicios de los hosts Linux.
- `windows.cfg`: Plantilla de configuración para monitorizar los servicios de los hosts Windows.
- `netdev.cfg`: Plantilla de configuración para monitorizar los servicios de los dispositivos de red.

También crearemos cuatro una serie de grupos de monitorización:

- `nagios-servers`: Agrupa a todos los servidores nagios de nuestra red.
- `hypervisors`: Agrupa a nuestra infraestructura de virtualización.
- `network-devices`: Agrupa a los dispositivos de red.
- `linux-servers`: Agrupa a los servidores Linux (no necesariamente localhost ni los host hypervisors ni dispositivos de red, que también pueden ser host Linux).
- `windows-servers`: Agrupa a los servidores Windows.
- `admins-stations`: Agrupa a las estaciones de trabajo de los administradores, que se encuentran en Windows.



#=====

#### NOTA:

Si vamos al fichero «/etc/nagios/objects/templates.cfg» veremos cinco plantillas predefinidas para los hosts:

- generic-host: Plantilla genérica que usan el resto de plantillas.
- linux-server: Plantilla para servidores Linux. Es la que se usa por defecto para Nagios (localhost).
- windows-server: Plantilla para servidores Windows.
- generic-printer: Plantilla para monitorizar impresoras.
- generic-switch: Plantilla para monitorizar switches.

Todas las plantillas son bastante similares, con algunas pequeñas modificaciones. De todos modos, intentaremos explicar brevemente algunas de las directivas más interesantes. Pero antes dejaremos claro que muchas de estas directivas simplemente tienen como valor un “1” o un “0”. Si está en “1” es que esta activada, si está en “0” desactivada. Las directivas que no se usan estarán desactivadas, pero algunas otras tienen valores por defecto, para conocer mejor esto lo mejor es mirar la documentación de Nagios. Así mismo, como ya comentamos, una plantilla puede tener un valor para una directiva, pero cuando otra plantilla o host usa esa plantilla puede sobrescribir la directiva con un valor distinto.

Una vez aclarado esto veamos algunas de las distintas directivas:

- name: Nombre utilizado para la directiva.
- notifications\_enabled: Si esta activada enviará notificaciones al o los contactos definidos.
- flap\_detection\_enabled: Habilita o no la detección de “flapping”. Ahora veremos brevemente qué es.
- notification\_period: Especifica el nombre del “time period” en el que se permite el envío de notificaciones.
- Register: Se usa únicamente con las plantillas y su valor debe ser “0”.

Antes de seguir avanzando, sería oportuno realizar una explicación rápida de algunas cosas.

- Las notificaciones son mensajes de alertas enviados a un contacto cuando sucede algo. Por ejemplo, si un host se apaga o falla un servicio, es posible configurar que se envíe una notificación a uno o más contactos vía email, sms, mensajería instantánea etc...
- El flapping es un estado especial que tienen los hosts y los servicios. Básicamente podemos decir que esto ocurre cuando su estado cambia muchas veces en poco tiempo. Por ejemplo, cuando lo reiniciamos varias veces en poco tiempo. Una de las ventajas de esto es que cuando sucede no se envían notificaciones. Imagina lo



molesto que es recibir muchas notificaciones en poco tiempo por un equipo que estamos reiniciando.

- Los time periods se definen en el fichero “timeperiods.cfg” y pueden ser usados para definir cuando queremos que se monitorice un equipo, o cuando queremos que se envíen notificaciones.

#=====

### En “nagios-debian9”:

Definiendo el host «linux.cfg»:

```
nano /etc/nagios/nagios.cfg
```

Debajo de la sección “OBJECT CONFIGURATION FILE(S)” añadimos la referencia a los que serán nuestros ficheros de configuración:

```
# Definitions for monitoring a Linux machine
cfg_file=/etc/nagios/objects/linux.cfg

# Definitions for monitoring the virtualization plattfomr host
cfg_file=/etc/nagios/objects/hypervisor.cfg

# Definitions for monitoring a Linux machine
cfg_file=/etc/nagios/objects/linux.cfg

# Definitions for monitoring the network devices
cfg_file=/etc/nagios/objects/netdev.cfg
```

Aprovechamos para editar el directorio de los logs de nagios, por el siguiente:

```
log_file=/var/log/nagios.log
```

Hacemos una salva del fichero «/etc/nagios/objects/localhost.cfg»:

```
cp /etc/nagios/objects/localhost.cfg /etc/nagios/objects/localhost.cfg.save
```



## 8.1. Configuración de los chequeos de Nagios al localhost

En “nagios-debian9”:

Editamos el fichero de configuración del localhost (nagios-debian9):

```
nano /etc/nagios/objects/localhost.cfg
```

Borramos todo y agregamos lo siguiente (adaptando a su red):

```
#####  
#####  
#  
# HOST DEFINITION  
#  
#####  
#####  
  
define host{  
    use                linux-server  
    host_name          nagios-debian9  
    alias Nagios  
    address            127.0.0.1  
    icon_image_alt     GNU/Linux  
    icon_image         nagios-logo.png  
    vrmf_image         nagios-logo.png  
    statusmap_image   nagios-logo.png  
}  
  
#####  
#####  
#  
# HOST GROUP DEFINITION  
#
```



```
#####  
#####
```

```
define hostgroup{  
    hostgroup_name      nagios-servers ; The name of the hostgroup  
    alias                Nagios Servers ; Long name of the group  
    members              nagios-debian9 ; Comma separated list of hosts that belong to this group  
}
```

```
#####  
#####
```

```
#  
# SERVICE DEFINITIONS FOR LOCALHOST (NAGIOS)  
#
```

```
#####  
#####
```

```
#####  
# SERVICIOS GENERICOS #  
#####
```

```
# Define a service to check the swap usage the local machine.  
# Critical if less than 10% of swap is free, warning if less than 20% is free
```

```
define service{  
    use                generic-service ; Name of service template to use  
    host_name          nagios-debian9  
    service_description Swap Usage  
    check_command      check_local_swap!20!10  
}
```

```
# Define a service to check SSH on the local machine.  
# Disable notifications for this service by default, as not all users may have SSH enabled.
```

```
define service{  
    use                generic-service ; Name of service template to use  
    host_name          nagios-debian9  
    service_description SSH  
}
```



```
        check_command      check_ssh
        notifications_enabled 0
    }

# Define a service to check the disk space of the root partition
# on the local machine. Warning if < 20% free, critical if
# < 10% free space on partition.
define service{
    use                generic-service ; Name of service template to use
    host_name          nagios-debian9
    service_description Root Partition
    check_command      check_local_disk!20%!10%!/
}

# Define a service to check the number of currently logged in
# users on the local machine. Warning if > 2 users, critical
# if > 5 users.
define service{
    use                generic-service ; Name of service template to use
    host_name          nagios-debian9
    service_description Current Users
    check_command      check_local_users!2!5
}

# Define a service to check the number of currently running procs
# on the local machine. Warning if > 250 processes, critical if
# > 400 users.
define service{
    use                generic-service ; Name of service template to use
    host_name          nagios-debian9
    service_description Total Processes
    check_command      check_local_procs!250!400!RSZDT
}

# Define a service to check the load on the local machine.
define service{
```



```
    use                generic-service ; Name of service template to use
    host_name          nagios-debian9
    service_description Current Load
    check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

#####
# SERVICIOS ESPECIFICOS #
#####

# Define a service to check HTTP on the local machine.
# Disable notifications for this service by default, as not all users may have HTTP enabled.
define service{
    use                generic-service ; Name of service template to use
    host_name          nagios-debian9
    service_description HTTP
    check_command      check_http
    notifications_enabled 0
}

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description System Description
    check_command      check_snmp!127.0.0.1!sysDescr.0!3!SHA!authPriv!AES!nagios!nagioskey!nagi
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description Location
    check_command      check_snmp!127.0.0.1!sysLocation.0!3!SHA!authPriv!AES!nagios!nagioskey!n
```





```
# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description Contact
    check_command       check_snmp!127.0.0.1!sysContact.0!3!SHA!authPriv!AES!nagios!nagioskey!nagio
}

# Define un servicio para obtener la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description Description eth0
    check_command       check_snmp!127.0.0.1!ifDescr.2!3!SHA!authPriv!AES!nagios!nagioskey!nagio
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "IP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description Link Status eth0
    check_command       check_snmp!127.0.0.1!ifOperStatus.2!3!SHA!authPriv!AES!nagios!nagioskey!nagio
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          nagios-debian9
    service_description Speed eth0
    check_command       check_snmp!127.0.0.1!ifSpeed.2!3!SHA!authPriv!AES!nagios!nagioskey!nagio
}
```



Reiniciamos el servicio de nagios:

```
systemctl restart nagios.service
```

## 8.2. Configuración de los chequeos de Nagios a hosts Linux

Editaremos el fichero “linux.cfg”, para adecuarlo a nuestros hosts Linux. Lo haremos con la posibilidad de que nrpe envíe comandos con argumentos, sin ser necesariamente el propio localhost (nagios), el host Linux a chequear. En este ejemplo, se tendrán en cuenta las siguientes consideraciones:

- El host FW1 (fw1-debian9) debe tener configurado múltiples servidores ovpnN. Si no se tiene esta estructura, entonces FW1 deberá enrutar el tráfico de la subred de gestión con el resto de la red. Ya no se tendría el principio de una gestión segura (por cifrado de la VPN) y aislada (por no ser enrutable). A continuación, se muestran las interfaces de FW1:



```
root@fw1-debian9:~# ifconfig
eth0: flags=419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet 172.16.12.2 netmask 255.255.255.252 broadcast 172.16.12.3
    ether ca:51:5e:94:05:88 txqueuelen 1000 (Ethernet)
    RX packets 114 bytes 13056 (12.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 255 bytes 26183 (25.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet 192.168.100.1 netmask 255.255.255.0 broadcast 192.168.100.255
    ether 56:fc:29:ef:67:c1 txqueuelen 1000 (Ethernet)
    RX packets 378 bytes 36176 (35.3 KiB)
    RX errors 0 dropped 9 overruns 0 frame 0
    TX packets 11 bytes 1500 (1.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet 192.168.2.1 netmask 255.255.255.0 broadcast 192.168.2.255
    ether be:c5:b7:fb:df:e0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 1372 (1.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 192.168.10.1 netmask 255.255.255.255 destination 192.168.10.2
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 192.168.30.1 netmask 255.255.255.255 destination 192.168.30.2
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Interfaz de red de cara a la red LOCAL

Interfaz de red de cara a la GESTION

Interfaz de red de cara a la red de ALMACENAMIENTO

ovpn1 para nagios

ovpn2 para nagios

dos servidores ovpn para nagios, para balancear la carga de la comunicación gestor-agente de nagios



- Nagios tendrá la siguiente configuración de red (adapte a su entorno):

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.100.3 netmask 255.255.255.0 broadcast 192.168.100.255
ether fe:67:f4:7b:7c:da txqueuelen 1000 (Ethernet)
RX packets 97716 bytes 10863414 (10.3 MiB)
RX errors 0 dropped 667 overruns 0 frame 0
TX packets 28723 bytes 17069332 (16.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1 (Local Loopback)
RX packets 1298 bytes 491535 (480.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1298 bytes 491535 (480.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 192.168.10.3 netmask 255.255.255.255 destination 192.168.10.4
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
RX packets 7637 bytes 1066921 (1.0 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8312 bytes 6343541 (6.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 192.168.20.3 netmask 255.255.255.255 destination 192.168.20.4
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Nagios monitorizará servicios de los hosts Linux sin comandos NRPE para: localhost (nagios).
- Nagios monitorizará servicios de los hosts Linux mediante comandos NRPE con argumentos y chequeos directos para: pve1-debian9, fw1-debian9, "hostmaster-pc" y el resto de host Linux y Windows que se vayan agregando con el tiempo.
- Nagios monitorizará el SNMP de los dispositivos de interconexión de red, mediante comandos NRPE con argumentos y chequeos indirectos para: fw2-vyos1.2.0-rc8, fw3-vyos1.2.0-rc8, a través de "fw1-debian9". Esto se debe a que, los dispositivos de red generalmente son dispositivos físicos que no tienen implementado un cliente OpenVPN (necesario para la comunicación de agente-gestor de la red



propuesta en este tutorial).

- También monitorizará su propia información SNMP, así como la de los hosts que esté monitorizando (no necesariamente dispositivos de red).
- La información SNMP aportará los siguientes datos:

- o sysDescr: Descripción del sistema.
- o sysLocation: Localización del servidor o dispositivo de red.
- o sysContact: Contacto del administrador.
- o ifDescr: Descripción de la interfaz de red.
- o ifOperStatus: Estado operacional de la interfaz de red.
- o ifSpeed: velocidad de la interfaz de red.

### En “nagios-debian9”:

Comenzamos a configurar la monitorización a realizar por nagios:

```
nano /etc/nagios/objects/linux.cfg
```

Borramos todo y agregamos lo siguiente (adaptando a su red):

```
#####  
#####  
#  
# HOST DEFINITION  
#  
#####  
#####  
  
define host{  
    use                linux-server  
    host_name          netprofana-debian9  
    alias              Netdata+Prometheus+Grafana  
    address            192.168.100.5  
    icon_image_alt     GNU/Linux
```



```
    icon_image          linux40.png
    vrml_image          linux40.png
    statusmap_image     linux40.gd2
}

define host{
    use                  linux-server
    host_name           ftp2-debian9
    alias               FTP DMZ
    address             192.168.20.81
    icon_image_alt      GNU/Linux
    icon_image          linux40.png
    vrml_image          linux40.png
    statusmap_image     linux40.gd2
}

#####
#####
#
# HOST GROUP DEFINITION
#
#####
#####

define hostgroup{
    hostgroup_name      linux-servers ; The name of the hostgroup
    alias               Linux Servers ; Long name of the group
    members             netprofana-debian9,ftp2-debian9 ; Comma separated list of hosts that be
}

#####
#####
#
# SERVICE DEFINITIONS FOR HOSTGROUPS
#
#####
```



```
#####  
  
# Define un servicio para chequear el uso de la swap en el host.  
# Warning: si es < 20% de swap libre  
# Critical: si es < 10% de swap libre  
define service{  
    use                generic-service,graphed-service  
    hostgroup_name     linux-servers  
    service_description Swap Usage  
    check_command      check_nrpe!check_swap!-a 20 10  
}  
  
# Define un servicio para chequear si esta habilitado o no el SSH.  
# Critical: si no esta habilitado  
define service{  
    use                generic-service  
    hostgroup_name     linux-servers  
    service_description SSH  
    check_command      check_nrpe!check_ssh!-a 127.0.0.1  
}  
  
#####  
#####  
#  
# SERVICE DEFINITIONS FOR LINUX HOSTS  
#  
#####  
#####  
  
#####  
# netprofana-debian9  
#####  
  
#####  
# SERVICIOS GENERICOS #  
#####
```



```
# Define un servicio para chequear el espacio en disco de la particion
# root del host.
# Warning: cuando queden 1024MB de espacio libre
# Critical: cuando queden 512MB de espacio libre
define service{
    use                generic-service,graphed-service
    host_name          netprofana-debian9
    service_description Root Partition
    check_command      check_nrpe!check_disk!-a 1024 512 /
}

# Define un servicio para chequear el numero de usuarios logueados en el
# sistema del host.
# Warning: si es > 2
# Critical: si es > 5
define service{
    use                generic-service,graphed-service
    host_name          netprofana-debian9
    service_description Current Users
    check_command      check_nrpe!check_users!-a 1 5
}

# Define un servicio para chequear el numero de procesos actuales
# que corren en el sistema.
# Warning: si el total es > 200
# Critical: si el total es > 400
define service{
    use                generic-service,graphed-service
    host_name          netprofana-debian9
    service_description Total Processes
    check_command      check_nrpe!check_procs!-a 500 800
}

# Define un servicio para chequear el average de la carga del sistema,
# por nucleo.
```





```
# Warning: si el average es > 5.0,4.0,3.0
# Critical: si el average es > 10.0,6.0,4.0
define service{
    use                generic-service,graphed-service
    host_name          netprofana-debian9
    service_description Current Load
    check_command      check_nrpe!check_load!-a 5.0,4.0,3.0 10.0,6.0,4.0
}

#####
# SERVICIOS ESPECIFICOS #
#####

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description System Description
    check_command      check_nrpe!check_snmp!-a 192.168.100.5 sysDescr.0 3 SHA authPriv AES
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description Location
    check_command      check_nrpe!check_snmp!-a 192.168.100.5 sysLocation.0 3 SHA authPriv AES
}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description Contact
    check_command      check_nrpe!check_snmp!-a 192.168.100.5 sysContact.0 3 SHA authPriv AES
}
```



```
# Define un servicio para obtener la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description Description eth0
    check_command       check_nrpe!check_snmp!-a 192.168.100.5 ifDescr.2 3 SHA authPriv AES n
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description Link Status eth0
    check_command       check_nrpe!check_snmp_ifOperStatus!-a 192.168.100.5 ifOperStatus.2 3
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          netprofana-debian9
    service_description Speed eth0
    check_command       check_nrpe!check_snmp!-a 192.168.100.5 ifSpeed.2 3 SHA authPriv AES n
}

#####
# ftp2-debian9
#####

#####
# SERVICIOS GENERICOS #
#####
```



```
# Define un servicio para chequear el espacio en disco de la particion
# root del host.
# Warning: cuando queden 1024MB de espacio libre
# Critical: cuando queden 512MB de espacio libre
define service{
    use                generic-service,graphed-service
    host_name          ftp2-debian9
    service_description Root Partition
    check_command      check_nrpe!check_disk!-a 1024 512 /
}

# Define un servicio para chequear el numero de usuarios logueados en el
# sistema del host.
# Warning: si es > 2
# Critical: si es > 5
define service{
    use                generic-service,graphed-service
    host_name          ftp2-debian9
    service_description Current Users
    check_command      check_nrpe!check_users!-a 1 5
}

# Define un servicio para chequear el numero de procesos actuales
# que corren en el sistema.
# Warning: si el total es > 200
# Critical: si el total es > 400
define service{
    use                generic-service,graphed-service
    host_name          ftp2-debian9
    service_description Total Processes
    check_command      check_nrpe!check_procs!-a 500 800
}

# Define un servicio para chequear el average de la carga del sistema,
# por nucleo.
```



```
# Warning: si el average es > 5.0,4.0,3.0
# Critical: si el average es > 10.0,6.0,4.0
define service{
    use                generic-service,graphed-service
    host_name          ftp2-debian9
    service_description Current Load
    check_command       check_nrpe!check_load!-a 5.0,4.0,3.0 10.0,6.0,4.0
}

#####
# SERVICIOS ESPECIFICOS #
#####

# Chequeo directo de Nagios a nas1-omv4.1.22
#

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description System Description
    check_command       check_nrpe!check_snmp!-a 192.168.20.81 sysDescr.0 3 SHA authPriv AE
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description Location
    check_command       check_nrpe!check_snmp!-a 192.168.20.81 sysLocation.0 3 SHA authPriv

}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
```



```
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description Contact
    check_command      check_nrpe!check_snmp!-a 192.168.20.81 sysContact.0 3 SHA authPriv AES
}

# Define un servicio para obtener la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description Description eth0
    check_command      check_nrpe!check_snmp!-a 192.168.20.81 ifDescr.2 3 SHA authPriv AES
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description Link Status eth0
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 192.168.20.81 ifOperStatus.2
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          ftp2-debian9
    service_description Speed eth0
    check_command      check_nrpe!check_snmp!-a 192.168.20.81 ifSpeed.2 3 SHA authPriv AES
}
```



```
}
```

Reiniciamos el servicio de nagios:

```
systemctl restart nagios.service
```

### 8.3. Configuración de los chequeos de Nagios a los Hypervisors

#### En “nagios-debian9”:

Los hypervisors de nuestra red son Proxmox VE 5.4-3, sobre S.O. Debian 9, por lo que se pueden considerar como host Linux también. Lo separamos de los demás hosts porque pudieran ser ESXi u otra plataforma de virtualización.

Comenzamos a configurar la monitorización a realizar por nagios:

```
nano /etc/nagios/objects/hypervisor.cfg
```

Borramos todo y agregamos lo siguiente (adaptando a su red):

```
#####  
#####  
#  
# HOST DEFINITION  
#  
#####  
#####  
  
define host{  
    use                linux-server  
    host_name          pve1-proxmox5.4-3  
    alias              Proxmox VE 5.4-3  
    address            192.168.100.2  
    icon_image_alt    GNU/Linux  
    icon_image         pve-logo.png  
    vrmf_image         pve-logo.png
```



```
        statusmap_image                pve-logo.png
    }

#####
#####
#
# HOST GROUP DEFINITION
#
#####
#####

define hostgroup{
    hostgroup_name        hypervisors ; The name of the hostgroup
    alias                 Hypervisors ; Long name of the group
    members               pvel-proxmox5.4-3 ; Comma separated list of hosts that belong to t
}

#####
#####
#
# SERVICE DEFINITIONS FOR HOSTGROUPS
#
#####
#####

# Define un servicio para chequear el uso de la swap en el host.
# Warning: si es < 20% de swap libre
# Critical: si es < 10% de swap libre
define service{
    use                   generic-service
    hostgroup_name       hypervisors
    service_description   Swap Usage
    check_command         check_nrpe!check_swap!-a 20 10
}

# Define un servicio para chequear si esta habilitado o no el SSH.
```



```
# Critical: si no esta habilitado
define service{
    use                generic-service
    hostgroup_name     hypervisors
    service_description SSH
    check_command      check_nrpe!check_ssh!-a 127.0.0.1
}

# Define un servicio que muestre el tiempo que lleva encendido el servidor
define service{
    use                local-service ; Name of service template to use
    hostgroup_name     hypervisors
    service_description Uptime
    check_command      check_nrpe!check_uptime
}

#####
#####
#
# SERVICE DEFINITIONS FOR HYPERVISORS HOSTS
#
#####
#####

#####
# pve1-debian9
#####

#####
# SERVICIOS GENERICOS #
#####

# Define un servicio para chequear el espacio en disco de la particion
# root del host.
# Warning: cuando queden 1024MB de espacio libre
# Critical: cuando queden 512MB de espacio libre
```





```
define service{
    use                generic-service
    host_name          pvel-proxmox5.4-3
    service_description Root Partition
    check_command      check_nrpe!check_disk!-a 1024 512 / }

# Define un servicio para chequear el numero de usuarios logueados en el
# sistema del host.
# Warning: si es > 2
# Critical: si es > 5
define service{
    use                generic-service
    host_name          pvel-proxmox5.4-3
    service_description Current Users
    check_command      check_nrpe!check_users!-a 1 5
}

# Define un servicio para chequear el numero de procesos actuales
# que corren en el sistema.
# Warning: si el total es > 200
# Critical: si el total es > 400
define service{
    use                generic-service
    host_name          pvel-proxmox5.4-3
    service_description Total Processes
    check_command      check_nrpe!check_procs!-a 500 800
}

# Define un servicio para chequear el average de la carga del sistema,
# por nucleo.
# Warning: si el average es > 5.0,4.0,3.0
# Critical: si el average es > 10.0,6.0,4.0
define service{
    use                generic-service
    host_name          pvel-proxmox5.4-3
    service_description Current Load
```



```
check_command                check_nrpe!check_load!-a 5.0,4.0,3.0 10.0,6.0,4.0
}

#####
# SERVICIOS ESPECIFICOS #
#####

# Chequeo directo de Nagios a pvel-debian9
#

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          pvel-proxmox5.4-3
    service_description System Description
    check_command      check_nrpe!check_snmp!-a 192.168.100.2 sysDescr.0 3 SHA authPriv
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          pvel-proxmox5.4-3
    service_description Location
    check_command      check_nrpe!check_snmp!-a 192.168.100.2 sysLocation.0 3 SHA authPr
}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    host_name          pvel-proxmox5.4-3
    service_description Contact
    check_command      check_nrpe!check_snmp!-a 192.168.100.2 sysContact.0 3 SHA authPri
}

# Define un servicio para obtener la descripcion de las interfaces de red
```



```
define service {
    use                generic-service ; Inherit values from a template
    host_name          pve1-proxmox5.4-3
    service_description Description eth0
    check_command      check_nrpe!check_snmp!-a 192.168.100.2 ifDescr.2 3 SHA authPriv A
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          pve1-proxmox5.4-3
    service_description Link Status eth0
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 192.168.100.2 ifOperStatus
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          pve1-proxmox5.4-3
    service_description Speed eth0
    check_command      check_nrpe!check_snmp!-a 192.168.100.2 ifSpeed.2 3 SHA authPriv
}
```

Reiniciamos el servicio de nagios:

```
systemctl restart nagios.service
```

## 8.4. Configuración de los chequeos de Nagios a dispositivos de red



## En “nagios-debian9”:

Los dispositivos de red que se tendrán en nuestra red son VNFs basadas en VyOS, herederos del antiguo sistema de networking de Vyatta. Usar este sistema no es obligado, ustedes pueden emplear cualquiera. Esto es sólo un ejemplo.

Podrán notar, que los tres hosts a ser monitoreados por nagios en esta plantilla, responden a la misma dirección IP, del host “fw1-debian9”. Esto se debe a que nagios sabe cómo alcanzar la IP de “fw1-debian9”, pero no la de los dispositivos de red, porque nagios está en una subred no enrutable (características propias de la infraestructura del ejemplo), por lo que se debe hacer uso de chequeos indirectos con nrpe, a través de “fw1-debian9”.

Comenzamos a configurar la monitorización a realizar por nagios:

```
nano /etc/nagios/objects/linux.cfg
```

Borramos todo y agregamos lo siguiente (adaptando a su red):

```
#####  
#####  
#  
# HOST DEFINITION  
#  
#####  
#####  
  
define host{  
    use                generic-switch  
    host_name          fw1-debian9  
    alias              FW1  
    address            192.168.100.1  
    icon_image_alt     Firewall  
    icon_image         fw.png  
    vrm1_image         fw.png  
    statusmap_image    fw.png  
}
```



```
define host{
    use                generic-switch
    host_name          fw2-vyos1.2.0-rc8
    alias              FW2
    address             192.168.100.1
    icon_image_alt     Firewall
    icon_image         fw.png
    vrml_image         fw.png
    statusmap_image    fw.png
}

define host{
    use                generic-switch
    host_name          fw3-vyos1.2.0-rc8
    alias              FW3
    address             192.168.100.1
    icon_image_alt     Firewall
    icon_image         fw.png
    vrml_image         fw.png
    statusmap_image    fw.png
}

#####
#####
#
# HOST GROUP DEFINITION
#
#####
#####

define hostgroup{
    hostgroup_name    network-devices ; The name of the hostgroup
    alias              Netwok Devices ; Long name of the group
    members            fw1-debian9, fw2-vyos1.2.0-rc8, fw3-vyos1.2.0-rc8 ; Comma separated
```



```
#####
#####
#
# SERVICE DEFINITIONS FOR LINUX HOSTS
#
#####
#####

#####
# fw1-debian9
#####

#####
# SERVICIOS GENERICOS #
#####

# Define un servicio para chequear el uso de la swap en el host.
# Warning: si es < 20% de swap libre
# Critical: si es < 10% de swap libre
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description Swap Usage
    check_command      check_nrpe!check_swap!-a 20 10
}

# Define un servicio para chequear si esta habilitado o no el SSH.
# Critical: si no esta habilitado
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description SSH
    check_command      check_nrpe!check_ssh!-a 127.0.0.1
}
}
```



```
# Define un servicio para chequear el espacio en disco de la particion
# root del host.
# Warning: cuando queden 1024MB de espacio libre
# Critical: cuando queden 512MB de espacio libre
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description Root Partition
    check_command      check_nrpe!check_disk!-a 1024 512 /
}

# Define un servicio para chequear el numero de usuarios logueados en el
# sistema del host.
# Warning: si es > 2
# Critical: si es > 5
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description Current Users
    check_command      check_nrpe!check_users!-a 1 5
}

# Define un servicio para chequear el numero de procesos actuales
# que corren en el sistema.
# Warning: si el total es > 200
# Critical: si el total es > 400
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description Total Processes
    check_command      check_nrpe!check_procs!-a 200 400
}

# Define un servicio para chequear el average de la carga del sistema,
# por nucleo.
# Warning: si el average es > 5.0,4.0,3.0
```



```
# Critical: si el average es > 10.0,6.0,4.0
define service{
    use                generic-service
    host_name          fw1-debian9
    service_description Current Load
    check_command      check_nrpe!check_load!-a 5.0,4.0,3.0 10.0,6.0,4.0
}

#####
# SERVICIOS ESPECIFICOS #
#####

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description System Description
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 sysDescr.0 3 SHA authPriv AE
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Location
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 sysLocation.0 3 SHA authPriv AE
}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Contact
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 sysContact.0 3 SHA authPriv AE
}
```





```
# Define un servicio para obtener la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Description eth0
    check_command       check_nrpe!check_snmp!-a 127.0.0.1 ifDescr.2 3 SHA authPriv AES
}

define service {
use generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Description eth1
    check_command       check_nrpe!check_snmp!-a 127.0.0.1 ifDescr.3 3 SHA authPriv AES
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Description eth2
    check_command       check_nrpe!check_snmp!-a 127.0.0.1 ifDescr.4 3 SHA authPriv AES
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description Link Status eth0
    check_command       check_nrpe!check_snmp_ifOperStatus!-a 127.0.0.1 ifOperStatus.2
}

define service {
    use                generic-service ; Inherit values from a template
```



```
    host_name          fw1-debian9
    service_description Link Status eth1
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 127.0.0.1 ifOperStatus.3
}

define service {
    use          generic-service ; Inherit values from a template
    host_name   fw1-debian9
    service_description Link Status eth2
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 127.0.0.1 ifOperStatus.4
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use          generic-service ; Inherit values from a template
    host_name   fw1-debian9
    service_description Speed eth0
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 ifSpeed.2 3 SHA authPriv AES
}

define service {
    use          generic-service ; Inherit values from a template
    host_name   fw1-debian9
    service_description Speed eth1
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 ifSpeed.3 3 SHA authPriv AES
}

define service {
    use          generic-service ; Inherit values from a template
    host_name   fw1-debian9
    service_description Speed eth2
    check_command      check_nrpe!check_snmp!-a 127.0.0.1 ifSpeed.4 3 SHA authPriv AES
}

# Define un servicio para obtener el consumo de ancho de banda en la interfaz de red:
#Warning: si el consumo llega a 80%
```



```
#Critical: si el consumo llega a 90%
    define service {
        use                generic-service ; Inherit values from a template
        host_name          fw1-debian9
        service_description BW eth0
        check_command      check_nrpe!check_iftraffic64.pl! -a 127.0.0.1 172.16.12.2 m -B
    }

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description BW eth1
    check_command      check_nrpe!check_iftraffic64.pl! -a 127.0.0.1 192.168.100.1 m -B
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw1-debian9
    service_description BW eth2
    check_command      check_nrpe!check_iftraffic64.pl! -a 127.0.0.1 192.168.2.1 m -B
}

#####
# fw2-vyos1.2.0-rc8
#####

# Define un servicio para chequear si esta habilitado o no el SSH.
# Critical: si no esta habilitado
define service{
    use                generic-service
    host_name          fw2-vyos1.2.0-rc8
    service_description SSH
    check_command      check_nrpe!check_ssh!-a 192.168.3.1
}

# Define un servicio para obtener la descripcion del dispositivo de red
```



```
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description System Description
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 sysDescr.0 3 MD5 authPriv
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Location
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 sysLocation.0 3 MD5 authPr
}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Contact
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 sysContact.0 3 MD5 authPri
}

# Define un servicio para obtener el estado del servicio SSH
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description SSH
    check_command       check_nrpe!check_ssh!-a 172.16.12.1
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
```



```
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth0
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth1
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth2
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth3
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth4
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
```



```
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status eth5
    check_command       check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Link Status bond0
    check_command       check_nrpe!check_snmp_ifOperStatus!-a 172.16.12.1 ifOperStatus.
}

# Define un servicio para obtener la velocidad de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth0
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.2 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth1
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.3 3 MD5 authPriv AES nagios nagioskey nagi
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth2
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.4 3 MD5 authPriv A
}

define service {
```



```
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth3
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.5 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth4
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.6 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed eth5
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.7 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Speed bond0
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifSpeed.8 3 MD5 authPriv A
}

# Define un servicio para obtener la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw2-vyos1.2.0-rc8
    service_description Description eth0
    check_command       check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.2 3 MD5 authPriv A
}

define service {
```



```
    use generic-service ; Inherit values from a template
    host_name fw2-vyos1.2.0-rc8
    service_description Description eth1
    check_command check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.3 3 MD5 authPriv A
}

define service {
    use generic-service ; Inherit values from a template
    host_name fw2-vyos1.2.0-rc8
    service_description Description eth2
    check_command check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.4 3 MD5 authPriv A
}

define service {
    use generic-service ; Inherit values from a template
    host_name fw2-vyos1.2.0-rc8
    service_description Description eth3
    check_command check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.5 3 MD5 authPriv A
}

define service {
    use generic-service ; Inherit values from a template
    host_name fw2-vyos1.2.0-rc8
    service_description Description eth4
    check_command check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.6 3 MD5 authPriv A
}

define service {
    use generic-service ; Inherit values from a template
    host_name fw2-vyos1.2.0-rc8
    service_description Description eth5
    check_command check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.7 3 MD5 authPriv A
}

define service {
    use generic-service ; Inherit values from a template
```





```
host_name          fw2-vyos1.2.0-rc8
service_description Description bond0
check_command      check_nrpe!check_snmp!-a 172.16.12.1 ifAlias.8 3 MD5 authPriv A
}

#####
# fw3-vyos1.2.0-rc8
#####

# Define un servicio para chequear si esta habilitado o no el SSH.
# Critical: si no esta habilitado
define service{
    use          generic-service
    host_name    fw3-vyos1.2.0-rc8
    service_description SSH
    check_command check_nrpe!check_ssh!-a 172.16.23.1
}

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use          generic-service ; Inherit values from a template
    host_name    fw3-vyos1.2.0-rc8
    service_description System Description
    check_command check_nrpe!check_snmp!-a 172.16.23.1 sysDescr.0 3 MD5 authPriv .
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use          generic-service ; Inherit values from a template
    host_name    fw3-vyos1.2.0-rc8
    service_description Location
    check_command check_nrpe!check_snmp!-a 172.16.23.1 sysLocation.0 3 MD5 authPr

}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
```



```
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Contact
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 sysContact.0 3 MD5 authPri
}

# Define un servicio para obtener el estado del servicio SSH
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description SSH
    check_command      check_nrpe!check_ssh!-a 172.16.23.1
}

# Define un servicio para obtener el estado operacional de las interfaces de red
#ifOperStatus da valor "1" si la interfaz esta "UP"
#ifOperStatus da valor "2" si la interfaz esta "DOWN"
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Link Status eth0
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Link Status eth1
    check_command      check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
```



```
    service_description    Link Status eth2
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Link Status eth3
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Link Status eth4
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Link Status eth5
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Link Status eth6
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Link Status bond0
}
```



```
    check_command          check_nrpe!check_snmp_ifOperStatus!-a 172.16.23.1 ifOperStatus.
}

# Define un servicio para mostrar la velocidad de cada interface de red
define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Speed eth0
    check_command          check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.2 3 MD5 authPriv A
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Speed eth1
    check_command          check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.3 3 MD5 authPriv A
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Speed eth2
    check_command          check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.4 3 MD5 authPriv A
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Speed eth3
    check_command          check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.5 3 MD5 authPriv A
}

define service {
    use                    generic-service ; Inherit values from a template
    host_name              fw3-vyos1.2.0-rc8
    service_description    Speed eth4
}
```



```
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.5 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Speed eth5
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.6 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Speed eth6
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.7 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Speed bond0
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifSpeed.8 3 MD5 authPriv A
}

# Define un servicio para mostrar la descripcion de las interfaces de red
define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth0
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.2 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth1
```



```
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.3 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth2
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.4 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth3
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.5 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth4
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.6 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description eth5
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.7 3 MD5 authPriv A
}

define service {
    use                generic-service ; Inherit values from a template
    host_name          fw3-vyos1.2.0-rc8
    service_description Description bond0
    check_command      check_nrpe!check_snmp!-a 172.16.23.1 ifAlias.8 3 MD5 authPriv A
}
```



```
}
```

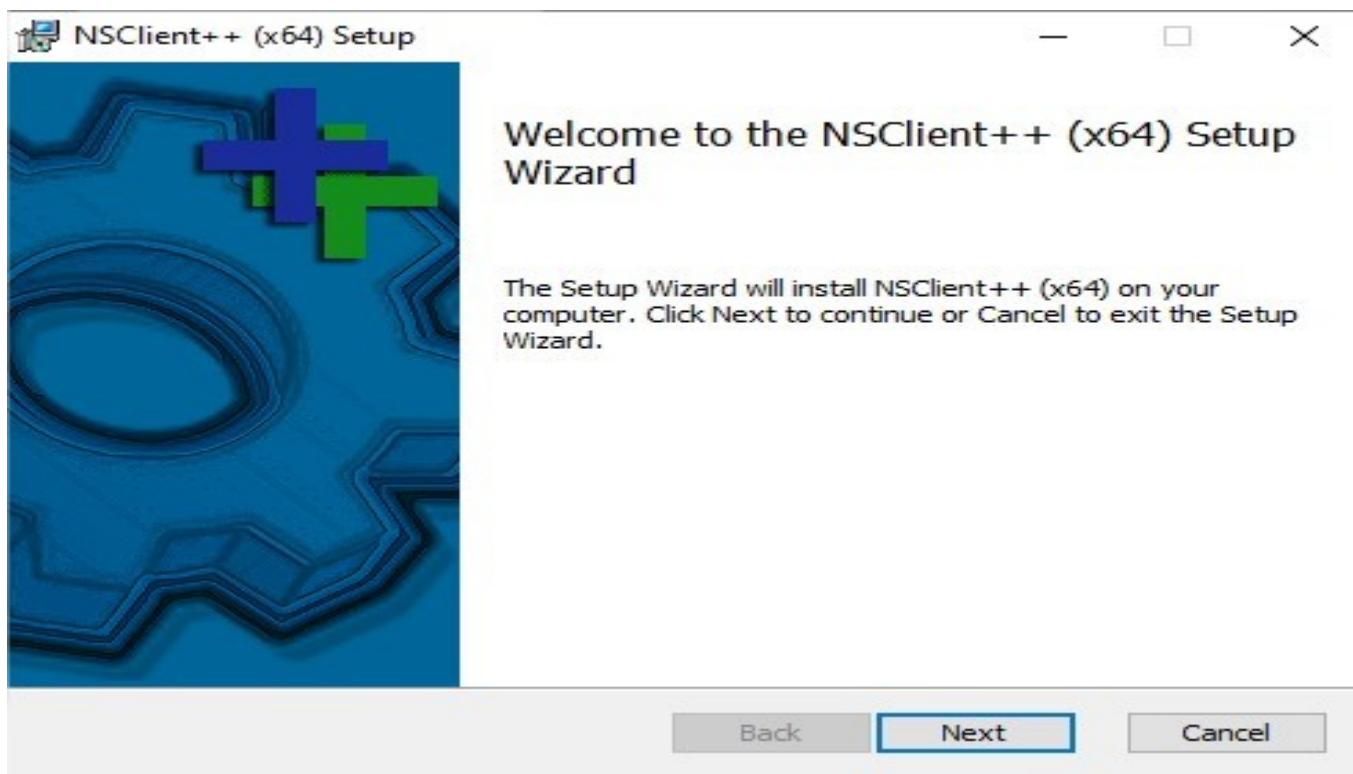
Reiniciamos el servicio de nagios:

```
systemctl restart nagios.service
```

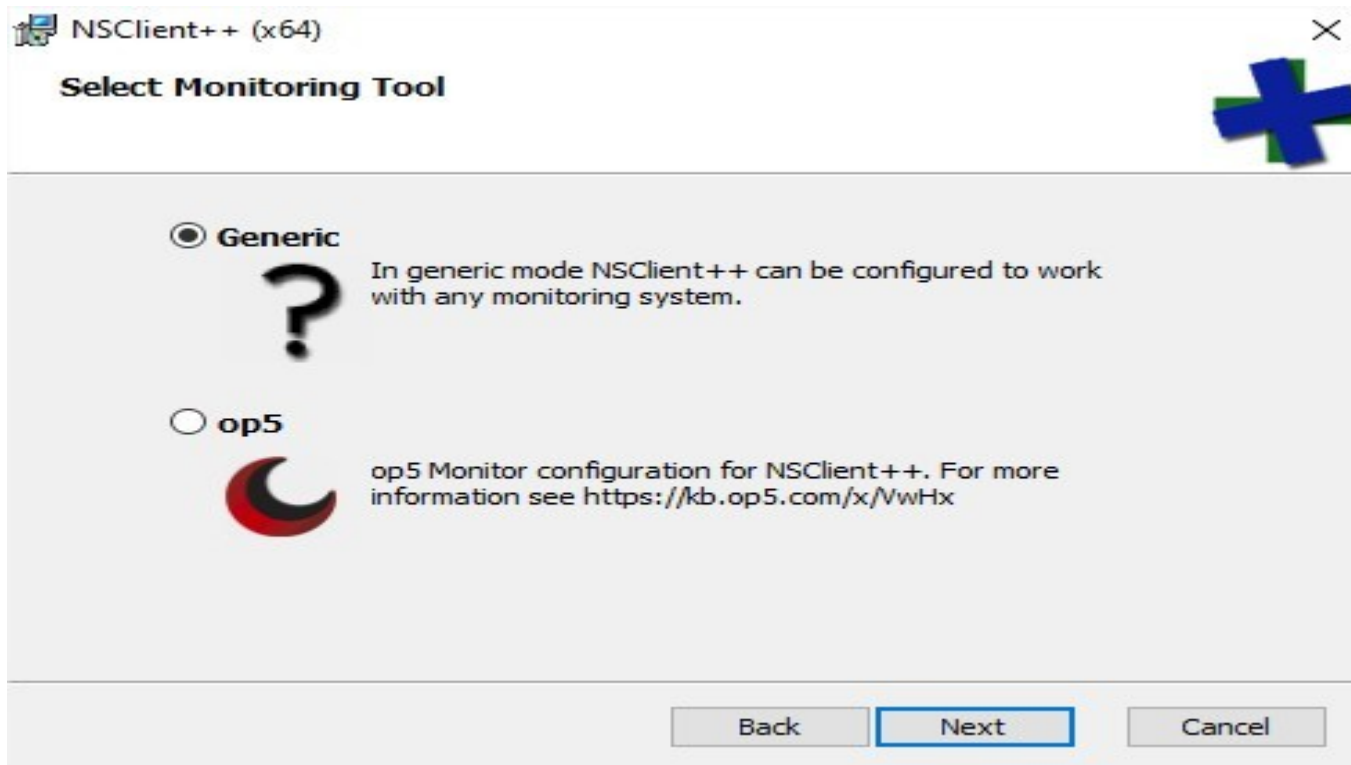
## 8.5. Configuración de los chequeos de Nagios a hosts Windows

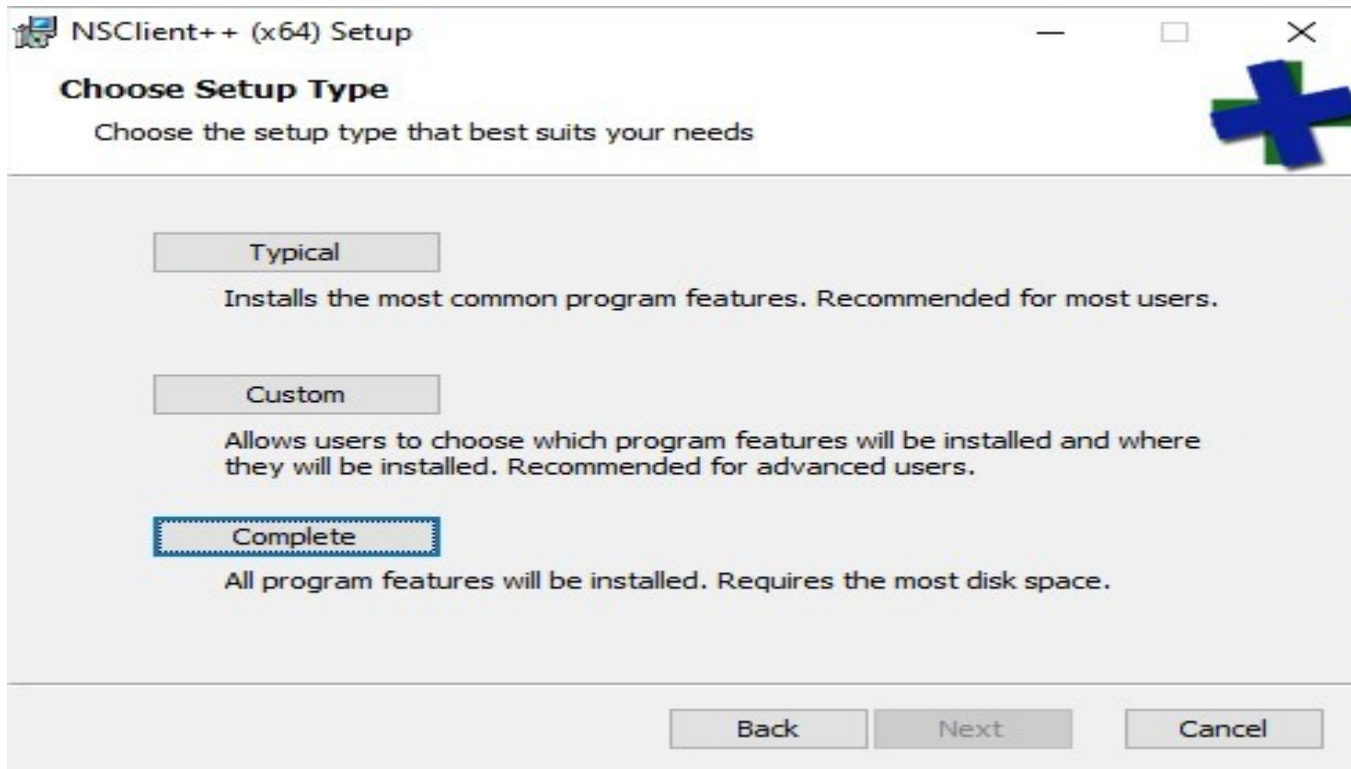
**En todos los hosts Windows:**

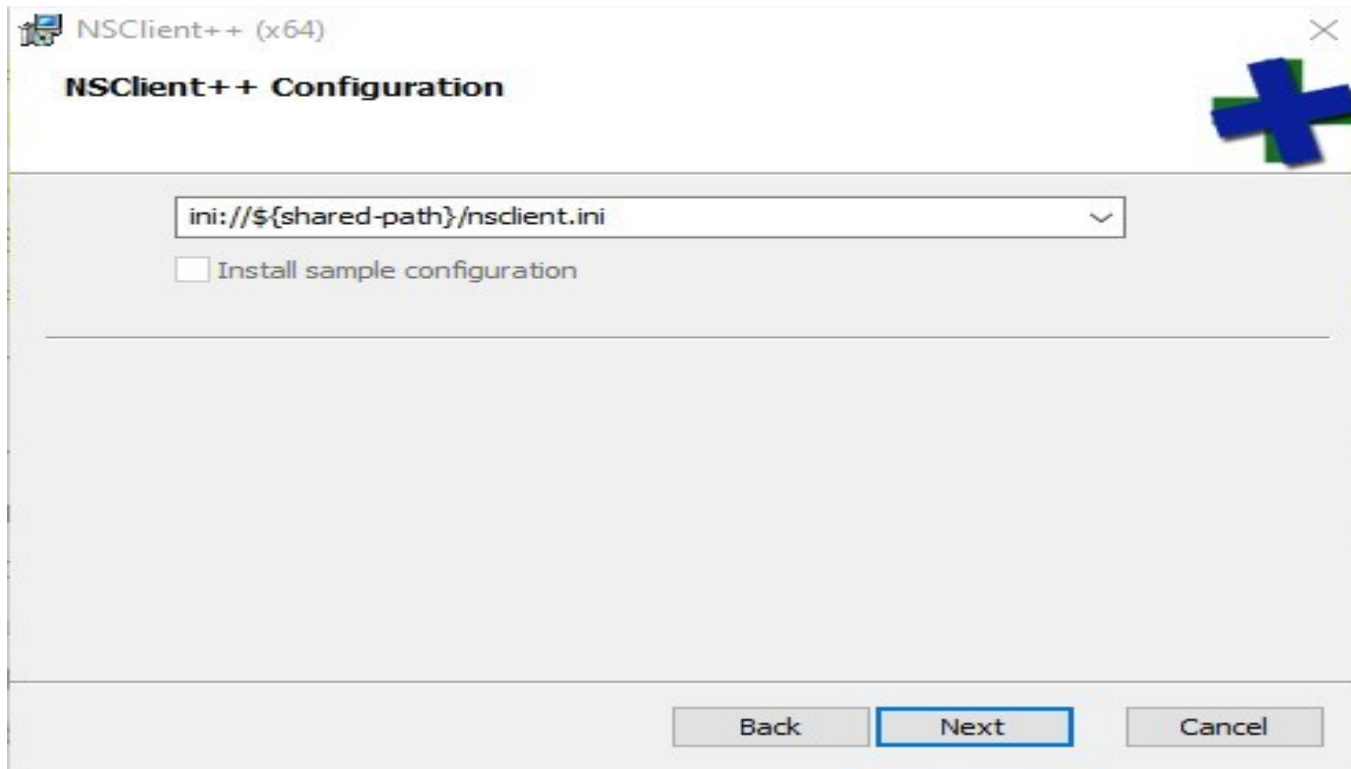
La configuración en los equipos Windows es muy sencilla. Lo primero es instalar el agente en los hosts Windows a monitorizar, en nuestro caso instalaremos “NSClient++” que es el que recomiendan en la documentación de Nagios. Sin embargo, hay más clientes que podrían servir, como “NC\_Net”. La instalación no es difícil, solo hay que seguir el asistente y aceptar la licencia.



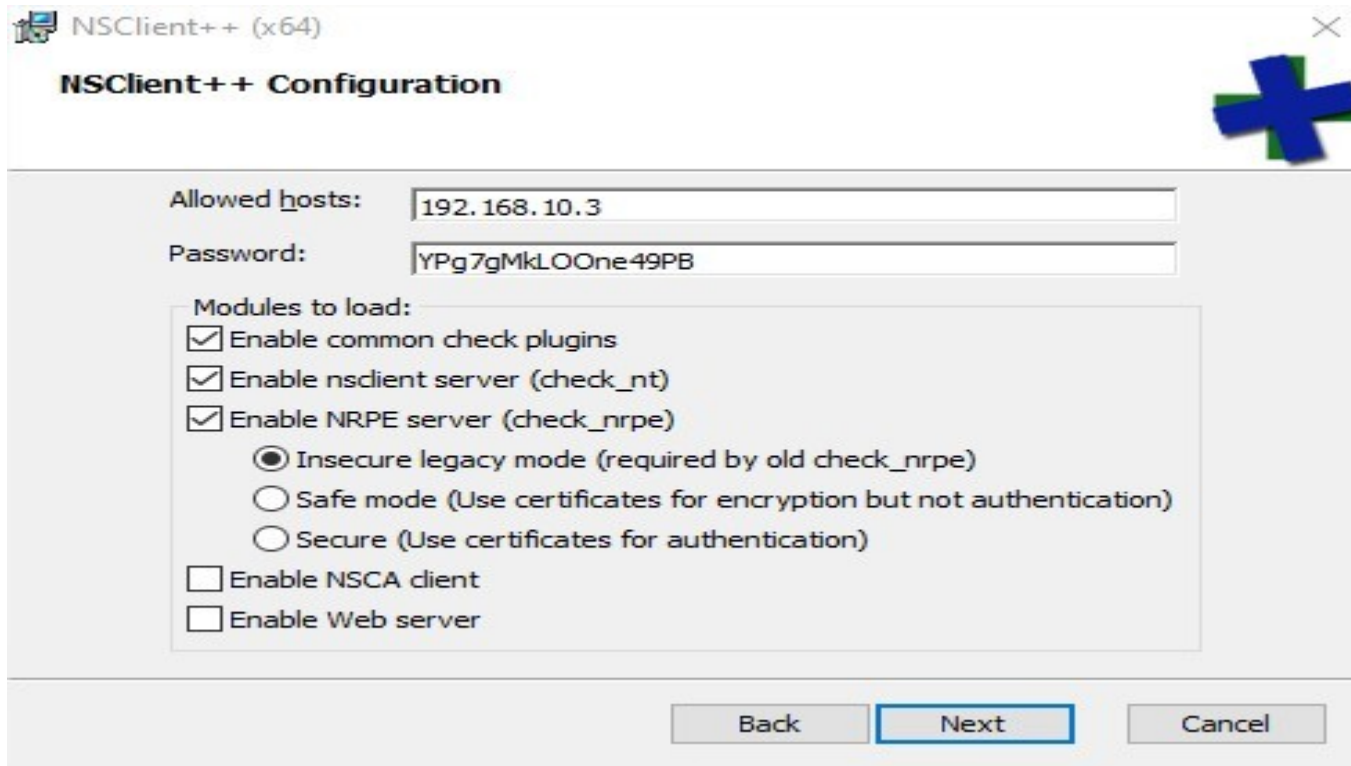








Establecemos en “Allowed hosts” la IP del servidor Nagios (en este caso en particular, la que tendrá por el ovpn1). Además, tenemos la opción de añadir una contraseña, aunque por defecto nos genera una aleatoria. Podemos utilizar esta, dejarlo en blanco o personalizarla a nuestro gusto. Es recomendable usar una contraseña ya que es una medida más de seguridad.

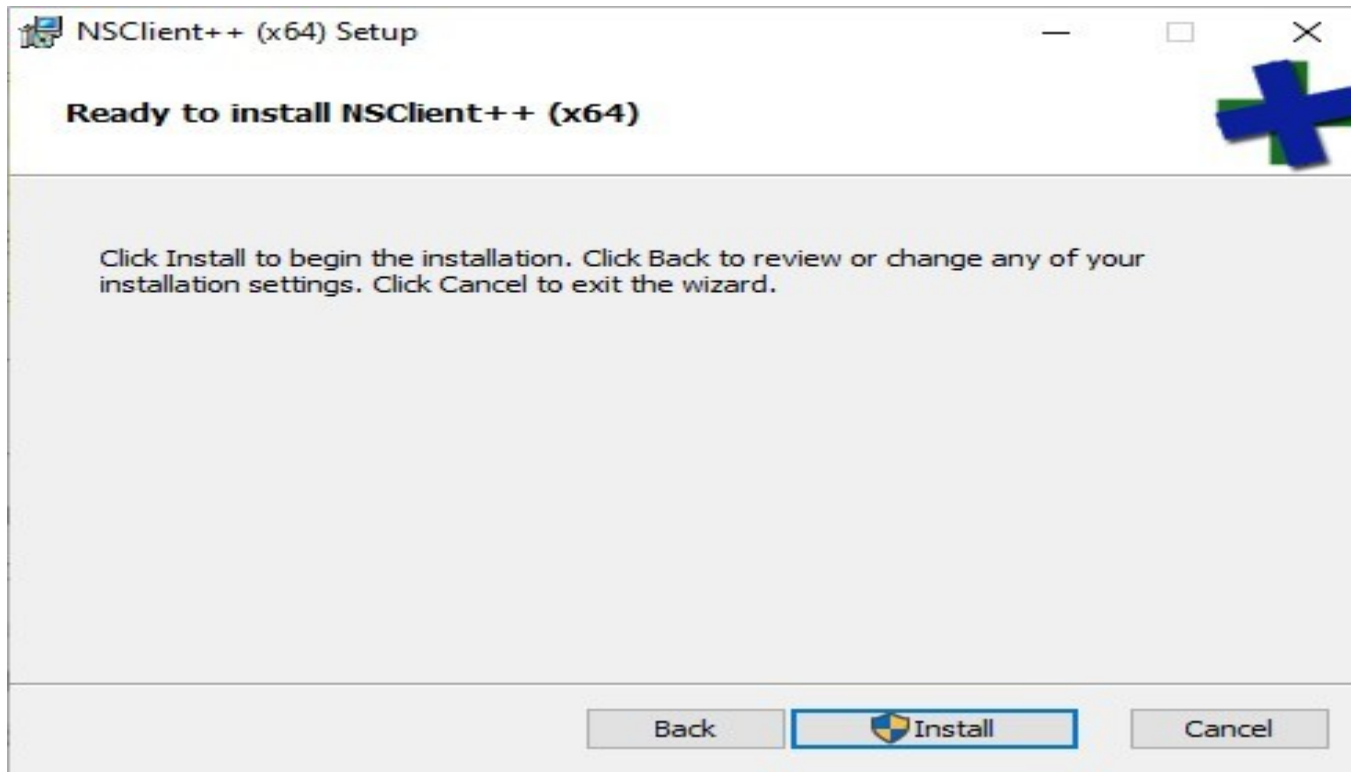


#=====

NOTA:

Copiamos la contraseña anterior “YPg7gMkLOOne49PB” (ponga la que le generó el cliente) y se la vamos a poner a cada host Windows en el que instalemos el “NSClient++”.

#=====



Hecho esto en todos los hosts Windows a monitorizar, procederemos con la configuración de los archivos de nagios. La idea es crear dos nuevos grupos (usted puede crear los que crea convenientes). Uno agrupará a las estaciones de trabajo de los administradores (generalmente en Windows) y el otro a los servidores con Windows. Se asumirá que las estaciones y servidores usan Windows 10.

### En “nagios-debian9”:

Editamos el fichero de configuración de nagios:

```
nano /etc/nagios/nagios.cfg
```



Descomentamos la línea que especifica la plantilla de configuración para servidores Windows:

```
# Definitions for monitoring a Windows machine
cfg_file=/etc/nagios/objects/windows.cfg
```

Editamos la plantilla de nagios que alberga la configuración genérica de sus objetos por defecto:

```
nano /etc/nagios/objects/templates.cfg
```

En la sección de “HOST TEMPLATES”, debajo de la definición de para host “windows-server”, agregamos lo siguiente:

```
define host {
    name                windows-stations                ; The name of this host template
    use                 generic-host                    ; Inherit default values from the generic-host t
    check_period        24x7                            ; By default, Windows servers are monitored roun
    check_interval      5                               ; Actively check the server every 5 minutes
    retry_interval      1                               ; Schedule host check retries at 1 minute interv
    max_check_attempts  10                             ; Check each server 10 times (max)
    check_command       check-host-alive               ; Default command to check if servers are "alive
    notification_period 24x7                            ; Send notification out at any time - day or nig
    notification_interv 30                             ; Resend notifications every 30 minutes
    notification_optio  d,r                             ; Only send notifications for specific host stat
    contact_groups      admins                          ; Notifications get sent to the admins by defaul
    hostgroups          admins-stations                 ; Host groups that Windows servers should be a m
    register            0                               ; DON'T REGISTER THIS - ITS JUST A TEMPLATE
}
```

Hacemos una copia del fichero original para servidores Windows:

```
cp /etc/nagios/objects/windows.cfg /etc/nagios/objects/windows.cfg.save
```

Editamos el fichero donde se definen los comandos para nagios:



```
nano /etc/nagios/objects/commands.cfg
```

En la definición del comando para “check\_nt”, comentamos el que viene por defecto y agregamos el siguiente:

```
command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s YPg7gMkL00ne49PB -v $ARG1$ $ARG2$
```

Editamos la plantilla de nagios para servidores Windows:

```
nano /etc/nagios/objects/windows.cfg
```

Borramos todo y agregamos lo siguiente (adaptando a su red):

```
#####  
#  
# HOST DEFINITIONS  
#  
#####
```

```
# Define a host for the Windows machine we'll be monitoring  
# Change the host_name, alias, and address to fit your situation
```

```
define host {
```

```
    use                windows-server                ; Inherit default values from a template  
    host_name          fileserver                    ; The name we're giving to this host  
    alias               FileServer                   ; A longer name associated with the host  
    address            192.168.20.51                 ; IP address of the host  
    icon_image         windows10-logo.png  
    icon_image_alt     Microsoft Windows 10  
    vrm1_image         windows10-logo.png  
    statusmap_image    windows10-logo.png  
}
```

```
define host {
```



```
use windows-stations ; Inherit default values from a template
host_name hostmaster-pc ; The name we're giving to this host
alias hostmaster-PC ; A longer name associated with the host
address 192.168.10.6 ; IP address of the host
icon_image windows10-logo.png
icon_image_alt Microsoft Windows 10
vrml_image windows10-logo.png
statusmap_image windows10-logo.png
}

#####
#
# HOST GROUP DEFINITIONS
#
#####

# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be a member of this group

define hostgroup {

    hostgroup_name windows-servers ; The name of the hostgroup
    alias Windows Servers ; Long name of the group
    members filesaver
}

define hostgroup {

    hostgroup_name admins-stations ; The name of the hostgroup
    alias Admins Stations ; Long name of the group
    members hostmaster-pc
}

#####
#
```





```
# HOSTGROUP SERVICE DEFINITIONS
#
#####

# Create a service for monitoring the version of NSClient++ that is installed
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description NSClient++ Version
    check_command      check_nt!CLIENTVERSION
}

# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined above

define service {

    use                local-service
    hostgroup_name     windows-servers,admins-stations
    service_description Uptime
    check_command      check_nt!UPTIME
}

# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description CPU Load
    check_command      check_nt!CPULOAD!-l 5,80,90
}


```



```
# Create a service for monitoring memory usage
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description Memory Usage
    check_command      check_nt!MEMUSE!-w 80 -c 90
}

# Create a service for monitoring C:\ disk usage
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description C:\ Drive Space
    check_command      check_nt!USEDISKSPACE!-l c -w 80 -c 90
}

define service {

    use                generic-service
    hostgroup_name     admins-stations
    service_description D:\ Drive Space
    check_command      check_nt!USEDISKSPACE!-l d -w 80 -c 90
}

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description Explorer
```



```
    check_command      check_nt!PROCSTATE!-d SHOWALL -l explorer.exe
}

define service {

    use                generic-service
    hostgroup_name     windows-servers,admins-stations
    service_description OCS Inventory NG
    check_command      check_nt!PROCSTATE!-d SHOWALL -l OcsService.exe
}

# Define un servicio para obtener la descripcion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    hostgroup_name     windows-servers,admins-stations
    service_description System Description
    check_command      check_snmpv2c!sysDescr.0!public!2c
}

# Define un servicio para obtener la localizacion del dispositivo de red
define service {
    use                generic-service ; Inherit values from a template
    hostgroup_name     windows-servers,admins-stations
    service_description Location
    check_command      check_snmpv2c!sysLocation.0!public!2c
}

# Define un servicio para obtener la informacion de contacto del administrador del dispositivo
define service {
    use                generic-service ; Inherit values from a template
    hostgroup_name     windows-servers,admins-stations
    service_description Contact
    check_command      check_snmpv2c!sysContact.0!public!2c
}
```



```
#####  
# hostmaster-pc  
#####
```

```
# Define un servicio para obtener la descripcion de las interfaces de red  
define service {  
    use                generic-service ; Inherit values from a template  
    host_name          hostmaster-pc  
    service_description Description eth0  
    check_command      check_snmpv2c!ifAlias.32!public!2c  
}
```

```
# Define un servicio para obtener el estado operacional de las interfaces de red  
#ifOperStatus da valor "1" si la interfaz esta "UP"  
#ifOperStatus da valor "2" si la interfaz esta "DOWN"  
#Con estos datos podemos crear una alerta cuando este "DOWN" la interfaz  
#Critical: si > 1 (-c :1, significa que todo lo que sea 1 o menor, no sera critico)  
define service {  
    use                generic-service ; Inherit values from a template  
    host_name          hostmaster-pc  
    service_description Link Status eth0  
    check_command      check_snmpv2c!ifOperStatus.32!public!2c  
}
```

```
# Define un servicio para obtener la velocidad de las interfaces de red  
define service {  
    use                generic-service ; Inherit values from a template  
    host_name          hostmaster-pc  
    service_description Speed eth0  
    check_command      check_snmpv2c!ifSpeed.32!public!2c  
}
```

Reiniciamos el servidor nagios para aplicar los cambios:


















```
systemctl restart nagios
```

## 8.6. Mostrando los resultados de los chequeos

Accedemos a la web de nagios y verificamos los cambios realizados. Podremos ver cómo nos saldrán los grupos que planificados:

Service Overview For All Host Groups

Admins Stations (admins-stations)				Linux Servers (linux-servers)				Windows Servers (windows-servers)			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
hostmaster-pc	UP	7 OK 1 CRITICAL	  	fw1-debian9	UP	57 OK 1 WARNING	  	fileserver	DOWN	7 CRITICAL	  
				nagios-debian9	UP	7 OK	  				
				pve1-debian9	UP	6 CRITICAL	  				

#=====

NOTA:

En este caso aparecen algunos hosts en estado “DOWN”, porque no aún no están creados dichos servidores o no se tienen configurados los plugins de nagios. Se hizo para que se vieran todos los estados posibles en Nagios.

#=====

Con las configuraciones realizadas, ahora el host “nagios-debian9”, por ejemplo, no solo muestra los servicios que tenía por defecto, sino que ahora agrega también, la información proporcionada por SNMP:



nagios-debian9		Current Load	OK	08-31-2019 20:46:27	5d 8h 22m 39s	1/4	OK - load average: 0.00, 0.02, 0.00
		Current Users	OK	08-31-2019 20:46:51	2d 2h 37m 31s	1/4	USERS OK - 1 users currently logged in
		HTTP	OK	08-31-2019 20:46:51	5d 8h 22m 39s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0,010 second response time
		Nagios Contact	OK	08-31-2019 20:45:27	0d 0h 5m 20s	1/3	SNMP OK - "ing, Franco Diaz <hostmaster@nodo.snet>"
		Nagios Description eth0	OK	08-31-2019 20:45:29	0d 0h 5m 18s	1/3	SNMP OK - "VMware VMXNET3 Ethernet Controller"
		Nagios Link Status eth0	OK	08-31-2019 20:44:44	0d 0h 6m 3s	1/3	SNMP OK - No Such Instance currently exists at this OID
		Nagios Location	OK	08-31-2019 20:45:05	0d 0h 5m 42s	1/3	SNMP OK - "pve1, NODO, 10 de Octubre, Sevillano, La Habana"
		Nagios Speed eth0	OK	08-31-2019 20:45:00	0d 0h 6m 23s+	1/3	SNMP OK - No Such Instance currently exists at this OID
		Nagios System Description	OK	08-31-2019 20:44:06	0d 0h 46m 41s	1/3	SNMP OK - "Linux nagios 4.9.0-3-amd64 #1 SMP Debian 4.9.30-2+deb9u2 (2017-06-26) x86_64"
		Root Partition	OK	08-31-2019 20:46:51	5d 8h 22m 39s	1/4	DISK OK - free space: / 926 MB (35,76% inode=62%);
		SSH	OK	08-31-2019 20:47:54	5d 8h 22m 39s	1/4	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u3 (protocol 2.0)
		Swap Usage	OK	08-31-2019 20:48:30	5d 8h 22m 39s	1/4	SWAP OK - 100% free (537 MB out of 537 MB)
		Total Processes	OK	08-31-2019 20:49:07	5d 8h 22m 39s	1/4	PROCS OK: 46 processes with STATE = RSZDT

De igual manera Check\_MK mostrará estas informaciones en su web:



### Hostgroups

2 30s

#### Admins Stations

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	hostmaster-pc		hostmaster-PC	14	0	0	1	0

#### Hypervisors

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	pve1-proxmox5.4-3		Proxmox VE 5.4-3	0	0	0	15	0

#### Linux Servers

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
DOWN	ftp2-debian9		FTP DMZ	0	0	0	14	0
DOWN	netprofana-debian9		Netdata+Prometheus+Grafana	0	0	0	14	0

#### Nagios Servers

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	nagios-debian9		Nagios	15	0	0	0	0

#### Network Devices

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	fw1-debian9		FW1	21	1	0	0	0
UP	fw2-vyos1.2.0-rc8		FW2	28	0	0	0	0
UP	fw3-vyos1.2.0-rc8		FW3	29	0	0	2	0

#### Windows Servers

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
DOWN	fileserver		FileServer	0	0	0	10	0

#=====

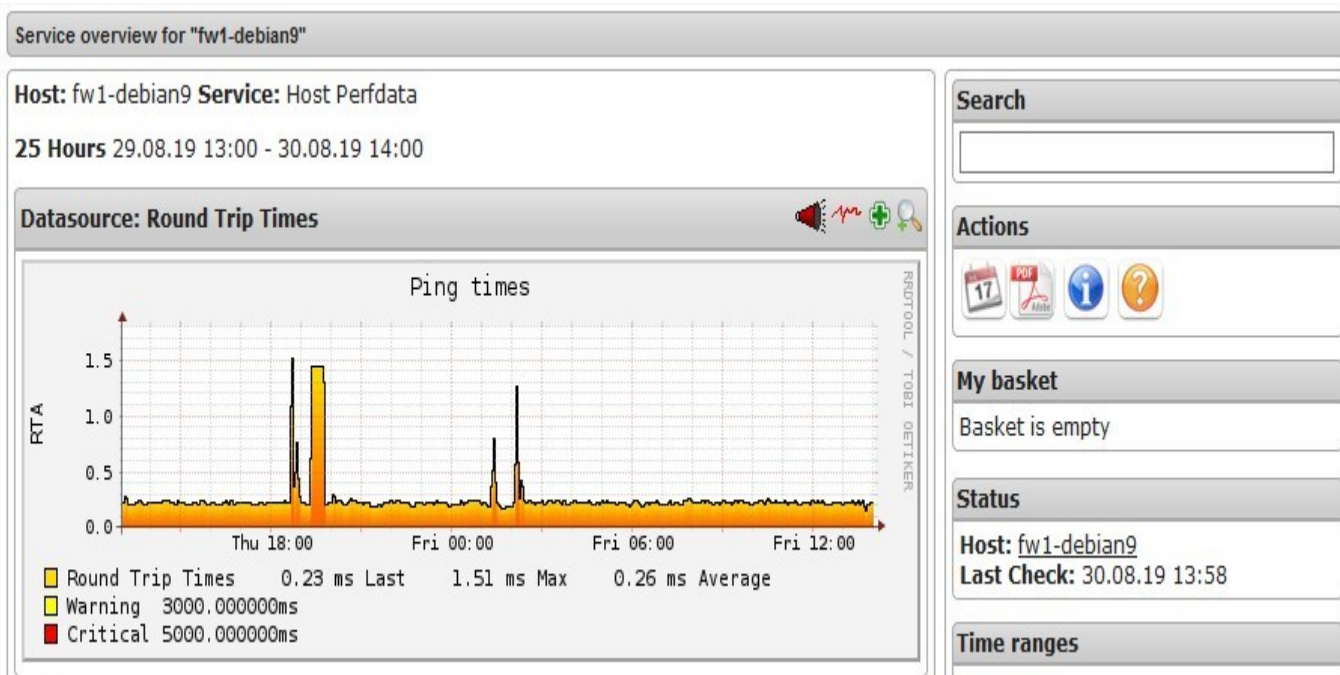
#### NOTA:

- Con los cambios hechos en las plantillas de nagios, es necesario recordar que ahora en la web de pnp4nagios podremos buscar las gráficas de cada uno de los hosts agregados a los chequeos. Como "localhost" ya no es uno de esos chequeos (es nagios-debian9), lo eliminamos de los registros de pnp4nagios:

```
rm -rf /usr/local/pnp4nagios/var/perfdata/localhost
```

Accedemos a la web de pnp4nagios con el siguiente enlace:

<http://192.168.100.3/site01/pnp4nagios/>



#=====

## Categoría

1. Como se hace
2. Grafiado
3. Monitoreo
4. Redes
5. Software Libre

Fecha de creación  
septiembre 2019





**Autor**  
franco\_sparrow