

Implementación de un Sistema de Monitoreo de Equipos de Red con Nagios

Emanuel David Cortes Antonio

22 de abril de 2019

Índice general

1. Desarrollo	5
1.1. Instalación y Configuración del sistema operativo con Nagios Core	5
1.1.1. Instalación del Sistema operativo	5
1.1.2. Ajustes previos del sistema operativo	6
1.1.3. Instalación de dependencias para compilar Nagios	7
1.1.4. Instalación de Nagios y plugins	7
1.2. Agregar Equipos a Nagios	10
1.2.1. Definición de objeto en Nagios	10
1.2.2. Definiendo un Host	10
1.2.3. Definiendo un HostGroup	11
1.2.4. Definiendo los servicios	11
1.3. Modificando la Interfaz Web	12
1.4. Alertas por Correo Electrónico	13

Capítulo 1

Desarrollo

1.1. Instalación y Configuración del sistema operativo con Nagios Core

1.1.1. Instalación del Sistema operativo

La instalación y configuración de Nagios Core puede variar según la elección del sistema operativo, para esta implementación se optó por CentOS que es un sistema Linux basado en RedHat y las principales características de su elección son que es software libre y que esta distribución está orientada a los servicios de Red.



Figura 1.1: CentOS v7

Después de descargar e instalar CentOS en el servidor es necesario configurar la interfaz de red para tener salida a internet y poder actualizar algunos repositorios y así tener una correcta instalación de Nagios Core.

```
# Para acceder a la configuración de red, es necesario abrir el archivo de  
# configuración de la interfaz de red del servidor
```

```
vim /etc/sysconfig/network-scripts/ifcfg-enp0
```

```
# En la configuración es necesario definir una ip estática
```

```
TYPE=Ethernet  
BOOTPROTO=static
```

```
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=ens3
ONBOOT=yes
IPADDR0= 192.168.80.23
NETMASK= 255.255.255.0
GATEWAY= 192.168.80.1
DNS1=172.16.16.5
DNS2=172.16.16.4

# Ejecutar los siguientes comandos para actualizar los cambios

systemctl stop NetworkManager
systemctl disable NetworkManager
systemctl restart network.service
```

1.1.2. Ajustes previos del sistema operativo

Se necesitan realizar algunos ajustes al sistema operativo que permitirán la instalación de paquetes y ejecución correcta de Nagios.

```
# Desactivar SELINUX que es un modulo con políticas de seguridad.
# editar el siguiente archivo

vim /etc/selinux/config

# Buscar la siguiente linea:

SELINUX = enforcing

# Reemplazar por:

SELINUX = disable

# Configurar puertos firewall para acceder al servidor a través de la web:

firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --permanent --add-port=443/tcp

# Aplicar cambios

firewall-cmd --reload

# Para comprobar que fue exitosa la configuración:
```

1.1. INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA OPERATIVO CON NAGIOS CORE 7

```
firewall-cmd --list-all

# Si todo fue correcto reiniciar el servidor

systemctl reboot
```

1.1.3. Instalación de dependencias para compilar Nagios

Es necesario preparar el sistema operativo con los paquetes de software que Nagios Core requiere para ser instalado, se deben introducir los siguientes comandos como usuarios root en la terminal para ser instalados:

```
#Instalacion de paquetes de dependencias
#1
yum install -y gettext wget net-snmp-utils openssl-devel glibc-common unzip
perl epel-release gcc php gd automake autoconf httpd make glibc gd-devel
net-snmp
#2
yum install perl-Net-SNMP
# Por motivos de seguridad Nagios debe tener su propio usuario y grupo, por
# ello es necesario crearlos:

useradd nagios
usermod -a -G nagios apache
id nagios
id apache
```

1.1.4. Instalación de Nagios y plugins

Lo primero que hay que hacer es descargar los archivos de instalación, para ello navegamos al sitio oficial y descargamos los paquetes de Nagios Core y Nagios plugins en su última versión:

```
# Descargar los siguientes paquetes en la ruta /home de preferencia

wget -c https://assets.nagios.com/downloads/nagioscore/releases/nagios-
4.4.3.tar.gz
wget -c https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
```

Ya con los archivos tar.gz descargados, es necesario dirigirse a la ruta donde se encuentran y será necesario descomprimirlo y compilarlo para ello se ingresarán los siguientes comandos en la terminal:

```
# Hay que dirigirse a la carpeta home donde se encuentran los archivos
```

```
cd /home
```

```
# Ahora hay que descomprimir el archivo de instalación de Nagios Core
```

```
tar xzvf nagios-4.4.3.tar.gz
```

```
# Dirigirse a la carpeta donde se descomprimió
```

```
cd nagios-4.4.3
```

```
# Ejecutamos el archivo configure y compilamos el makefile
```

```
./configure
```

```
make all
```

```
# Una vez compilado, se tendrá que instalar para ello ejecutamos:
```

```
make install
```

```
make install-init
```

```
make install-commandmode
```

```
make install-config
```

```
meke install-webconfig
```

```
# Sera necesario habilitar nagios en el sistema:
```

```
systemctl enable nagios
```

```
# También habilitar el servicio de apache
```

```
systemctl enable httpd
```

```
# Ahora sera necesario crear un usuario y una contraseña para el sistema nagios
```

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
password
```

```
re-password
```


1.1. INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA OPERATIVO CON NAGIOS CORE 9

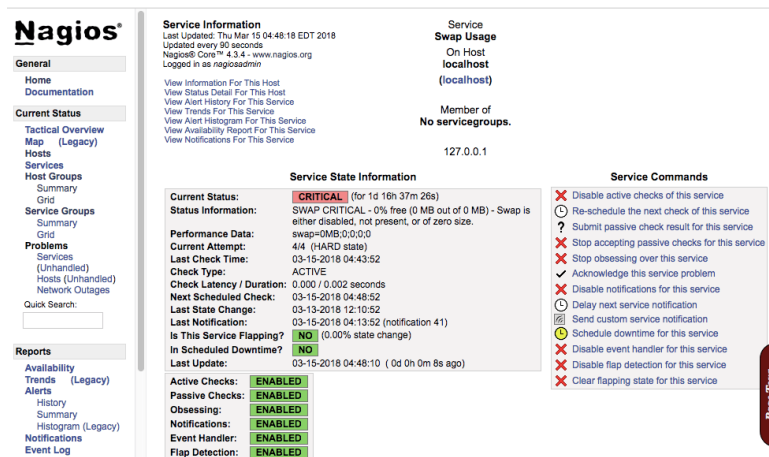
Para verificar que el las configuraciones esten correctas:

```
systemctl status httpd
```

```
systemctl status nagios
```

En el navegador web introducir la ip del servidor diagonal nagios

(xxxx.xxxx.xxxx.xxxx/nagios).



The screenshot shows the Nagios Core web interface. On the left is a sidebar with navigation links: General, Home, Documentation, Current Status, Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Summary, Grid, Service Groups, Summary, Grid, Problems, Services (Unhandled), Hosts (Unhandled), Network Outages, Quick Search, Reports, Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, and Event Log. The main content area is titled 'Service Information' and shows details for 'Swap Usage' on the 'localhost' host. The status is 'CRITICAL' (for 1d 16h 37m 26s). The service is a member of 'No servicegroups' and has a version of '127.0.0.1'. The 'Service State Information' section shows 'Current Status: CRITICAL', 'Status Information: SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.', 'Performance Data: swap=0MB:0:0:0', 'Current Attempt: 4/4 (HARD state)', 'Last Check Time: 03-15-2018 04:43:52', 'Check Type: ACTIVE', 'Check Latency / Duration: 0.000 / 0.002 seconds', 'Next Scheduled Check: 03-15-2018 04:48:52', 'Last State Change: 03-13-2018 12:10:52', 'Last Notification: 03-15-2018 04:13:52 (notification 41)', 'Is This Service Flapping? NO (0.00% state change)', 'In Scheduled Downtime? NO', and 'Last Update: 03-15-2018 04:48:10 (0d 0h 0m 8s ago)'. The 'Service Commands' section on the right lists various actions like 'Disable active checks of this service', 'Re-schedule the next check of this service', 'Submit passive check result for this service', 'Stop accepting passive checks for this service', 'Stop obsessing over this service', 'Acknowledge this service problem', 'Disable notifications for this service', 'Delay next service notification', 'Send custom service notification', 'Schedule downtime for this service', 'Disable event handler for this service', 'Disable flap detection for this service', and 'Clear flapping state for this service'. A 'Page Four' indicator is visible on the right side of the interface.

Figura 1.2: Nagios Core Página de inicio

Nagios es un sistema de monitorización con multiples servicios y por ello a la hora de configurar alguno en especifico podria haber un error el cual seria muy complejo detectar, para solucionar este problema y especificar algun servicio que se desee agregar (un servidor, switch o router) se dispone de un paquete oficial de plugins en la pagina de Nagios Core:

Anteriormente descargamos el paquete nagios-plugins-2.2.1.tar.gz dirigirse a
su ubicación para descomprimir el paquete ingresar:

```
tar xzf nagios-plugins-2.2.1.tar.gz
cd nagios-plugins-2.2.1
./configure
make && make install
```

Una vez terminado, podemos comprobar si se han instalado correctamente en
el directorio.
/usr/local/nagios/libexec/ mediante el comando:

```
ls /usr/local/nagios/libexec
```

1.2. Agregar Equipos a Nagios

1.2.1. Definición de objeto en Nagios

Los switches y routers pueden ser monitorizados por Nagios, la forma mas sencilla para determinar perdidas de paquetes es "pingueandolos" con agentes SNMP el inconveniente con esta última forma es que no todos los equipos de red lo soportan, por ello para esta implementación se optó por utilizar sólo pings.

En Nagios a las definiciones de hosts, contactos, servicios, comandos y otros se le conoce como objetos. La definición de objetos en Nagios se realiza a través de archivos con extensión ".cfg", la definición de estos archivos puede ser en cualquier parte de los directorios pero por convención y recomendación se pueden ubicar en la ruta `/usr/local/nagios/etc/objects` estos archivos ademas deben ser incluidos en la dirección `/usr/local/nagios/etc/nagios.cfg` en el cual estan las rutas de todos los objetos agregados. Sin embargo, Nagios ya trae unos cuantos archivos agregados en la ruta `/usr/local/nagios/etc/objects` para la definicion de los objetos, estos serviran como especies de plantillas para distintos servicios como servidores windows, linux, switches y routers.

1.2.2. Definiendo un Host

Lo primero será definir el host:

```
define host {  
    use generic-switch  
    host_name biblioteca  
    alias ENTERASYS  
    address 192.168.3.254  
    hostgroups Veterinaria  
    icon_image switch40.gif  
    statusmap_image switch40.gd2  
}
```

1. **use:** Con esta directiva se le indica una plantilla de la que heredar la configuración. En caso de conflicto porque una misma directiva se utilice tanto en la plantilla como en la definición del host, siempre tendrá prioridad el valor que se establece en la definición host. De momento usaremos esta plantilla y más adelante las veremos más detalladamente.

2. **host_name:** Nombre corto usado para identificar al host.
3. **alias:** Nombre o descripción usada para identificar al host.
4. **address:** Dirección IP del equipo a monitorizar.
5. **hostgroups:** Nombre del hostgroup al que pertenece.
6. **icon_image:** Agregar un icono al equipo en la página principal.
7. **statusmap_image:** Agregar un icono en el índice del host.

Aquí se pueden establecer muchas más directivas, y no todos los hosts tienen que estar definidos de la misma forma. Unos pueden tener unos valores y unas directivas, y otros otras. Incluso se puede evitar la utilización de plantillas, aunque siempre se deben incluir algunas directivas que son obligatorias, ya sean puestas explícitamente o heredadas de una plantilla.

1.2.3. Definiendo un HostGroup

En Nagios la definición de la directiva hostgroup no es obligatoria pero puede ser útil a la hora de visualizar los equipos en la interfaz web ya que se pueden agrupar o facilitar la gestión en algún servicio que será aplicado a todos los hosts de un determinado hostgroup.

```
define hostgroup {
    hostgroup_name    Veterinaria
    alias              Veterinaria
}
```

Un hostgroup solo necesita dos directivas, el nombre y el alias.

1.2.4. Definiendo los servicios

Son muchos los servicios que Nagios incluye para monitorear como se menciono anteriormente como algún servidor con sistema operativo windows o Linux, o en este caso un equipo de red.

```
define service{
    use                generic-service
    host_name           Biblioteca
    service_description PING
    check_command        check_ping!200.0,20%,!600.0,60%
    check_interval       5
    retry_interval       2
}
```

}

1. **use:** Como en el caso de la definición del host, esta directiva se emplea para utilizar una plantilla, en este caso generic-service.
2. **host_name:** Es el nombre del host o hosts a los que se le aplicará la monitorización de este servicio. Si quisiéramos especificar un hostgroup podríamos hacerlo utilizando la directiva hostgroup_name en su lugar. Si lo hiciéramos ya no sería obligatorio el uso de la directiva host_name.
3. **service_description:** Nombre descriptivo para el servicio.
4. **check_command:** El comando que usará este servicio junto con su variable, en este caso check_ping.
5. **check_interval:** Es el intervalo de tiempo que se espera para recibir una respuesta del comando, por default una unidad son 60 segundos.
6. **retry_interval:** Es el intervalo de tiempo que se espera para reintentar ejecutar el comando en caso que la respuesta del comando sea negativa.

1.3. Modificando la Interfaz Web

Ahora que ya tenemos instalado y agregados equipos a Nagios Core nos queda configurar la interfaz web en la que se podrán observar la monitorización de los servicios para ello es necesario acceder a la ruta `/usr/local/nagios/share` que es donde se encuentra la plantilla por defecto y es aquí donde se sustituye o se modifican los archivos html y css.

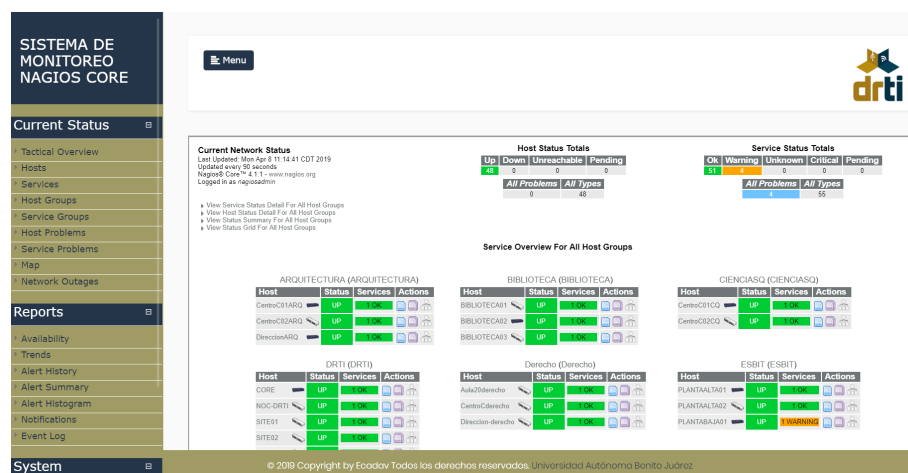


Figura 1.3: Nueva interfaz web de Nagios

Se optó por un diseño básico sin muchas modificaciones a la interfaz inicial, pero que incluyera una pequeña descripción del lugar en donde se implementó el sistema de monitoreo, además que el diseño fuera responsivo para poderse adaptar a todo tipo de tamaño de pantallas en los diferentes dispositivos con los que se cuentan.

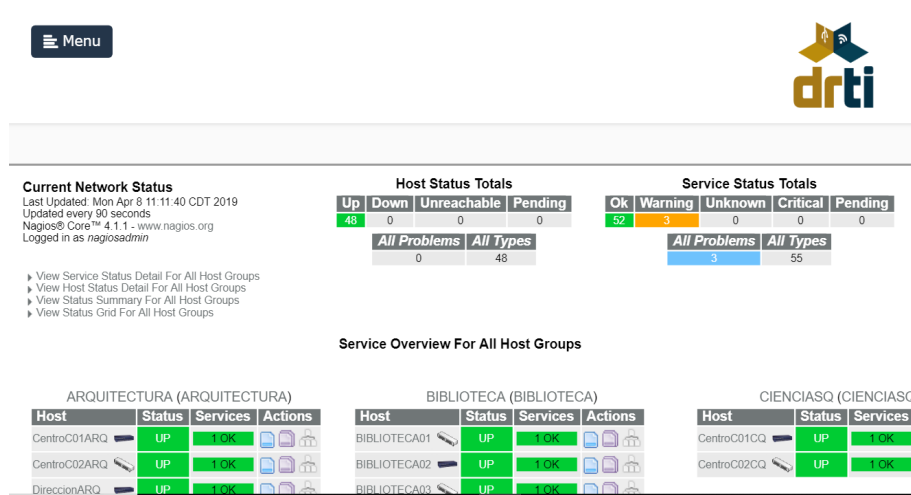


Figura 1.4: Nueva interfaz web de Nagios

1.4. Alertas por Correo Electrónico

Una de las características de Nagios es la de poder enviar notificaciones a ciertas personas cuando ocurre algo. Así, si un equipo está apagado, tiene un problema de algún tipo, un servicio no funciona etc. Además de plasmarlo en la interfaz de Nagios, podrá enviar una notificación al personal oportuno. Como ya comentamos también es posible configurar un manejador de eventos, para que se ejecute algo cuando sucede cierta cosa. Sin embargo aquí nos centraremos únicamente en las notificaciones. Estas notificaciones o alertas pueden realizarse de muchos métodos, pero aquí solo veremos como configurarlas para el correo.

Instalacion de dependencias para notificaciones por correo

Instalacion del repositorio remi y los paquetes php

```
yum install yum-utils http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

```
yum-config-manager --enable remi-php70
```

```
yum install php php-cli php-gd php-curl php-zip php-intl php-mbstring php-xml
```

Instalacion del composer de php

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

Instalacion del cliente smtp

```
wget https://github.com/boolean-world/smtp-cli/archive/master.zip
unzip -d /opt master.zip
cd /opt/smtp-cli-master
composer install
```

Crear el archivo config.json y colocar en él los datos de la cuenta de correo que enviara las notificaciones

```
"host": "smtp.gmail.com",  
"username": "the-senders-email@gmail.com",  
"password": "the-password-of-the-account",  
"secure": "tls",  
"port": 587
```

Ya tenemos configurado Nagios y el correo, ya solo quedan los contactos. Si recordáis, en `templates.cfg` también había una plantilla para los contactos. Ahora vamos a ver qué son y como usarlos para recibir las alertas vía email.

```
# Lo primero que haremos será analizar la plantilla:  
define contact{  
name generic-contact ;  
service_notification_period 24x7 ;  
host_notification_period 24x7 ;  
service_notification_options w,u,c,r,f,s ;  
host_notification_options d,u,r,f,s ;  
service_notification_commands notify-service-by-email ;  
host_notification_commands notify-host-by-email ;  
register 0 ;  
}
```

1. **service_notification_period:** Periodo de tiempo en el cual el contacto puede ser notificado
2. **host_notification_period:** Igual que el anterior pero para las notificaciones de los hosts.
3. **service_notification_options:** Tipo de notificaciones que serán enviadas para los servicios.
4. **host_notification_options:** Igual que el anterior pero con los estados de los hosts.
5. **service_notification_commands:** Comando que se ejecuta al enviar una notificación sobre un servicio.
6. **host_notification_commands:** Igual que el anterior pero para las notificaciones de los hosts.

Los contactos al igual que otros objetos también tienen su fichero particular. En este caso es `contacts.cfg` y en su interior ya viene definido un contacto llamado `nagiosadmin`. Nosotros nos quedaremos con este contacto tal como esta y únicamente editaremos el email.

```
# Para definir un contacto:
define contact{
contact_name nagiosadmin ;
use generic-contact ;
alias Nagios Admin ;
email nagios@openmailbox.org ;
}
```

Las directivas con las que viene definido ya nos deben sonar de otros objetos. Sin embargo existen algunas directivas más que no hemos visto aquí. Para obtener más información sobre estas y la definición de los contactos podéis leer la documentación.