

# Práctica Nagios

Palbo Jiménez Mateo y Mihaita Alexandru Lupoiu

Diciembre 2014

## Índice

<b>1. Introducción a Nagios</b>	<b>2</b>
1.1. Descripción . . . . .	3
1.2. Beneficios de utilizar Nagios . . . . .	4
1.3. Alternativas a Nagios . . . . .	4
<b>2. Funcionamiento de Nagios</b>	<b>5</b>
<b>3. Instalación de Nagios</b>	<b>5</b>
3.1. Instalación desde repositorio . . . . .	5
3.2. Instalación desde código fuente . . . . .	6
<b>4. Configuración</b>	<b>8</b>
4.1. Configuración de Nagios . . . . .	8
4.2. Configuración de los hosts externos . . . . .	9
<b>5. Acceso a Nagios mediante el navegador</b>	<b>10</b>
<b>6. Experiencia de instalación y configuración de Nagios</b>	<b>10</b>

## 1. Introducción a Nagios

Nagios es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado. Entre sus características principales figuran la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP...), la monitorización de los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado de los puertos...), independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL cifrados o SSH, y la posibilidad de programar plugins específicos para nuevos sistemas[2].

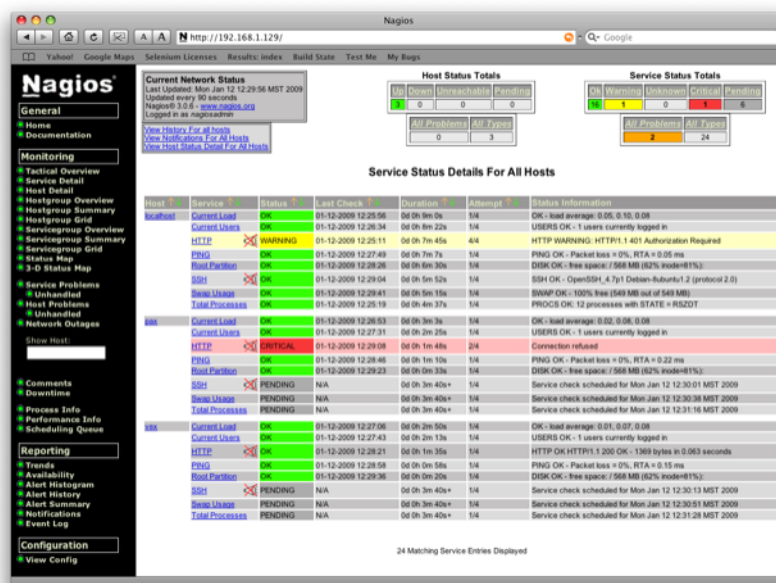


Figura 1: Interfaz de Nagios

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los responsables correspondientes mediante (entre otros medios) correo electrónico y mensajes SMS, cuando estos parámetros exceden de los márgenes definidos por el administrador de red.

Nagios[1] fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix. Está licenciado bajo la GNU General Public License Version 2 publicada por la Free Software Foundation.

## 1.1. Descripción

- Monitorización de Servicios de Red
- Monitorización de los recursos de equipos hardware (carga del procesador, uso de los discos, logs del sistema) en varios sistemas operativos, incluso Microsoft Windows con los plugins *NRPE<sub>N</sub>T* o *NSClient++*.
- Monitorización remota, a través de túneles SSL cifrados o SSH.
- Monitoreo de Host y sus recursos como CPU, Memoria, Discos, etc
- Diseño simple de plugins, que permiten a los usuarios desarrollar sus propios chequeos de servicios dependiendo de sus necesidades, usando sus herramientas preferidas (Bash, C++, Perl, Ruby, Python, PHP, C#...).
- Chequeo de servicios paralizados.
- Posibilidad de definir la jerarquía de la red, permitiendo distinguir entre host caídos y host inaccesibles.
- Capacidad de Definir Host/Servicios padres o hijos, lo que permite detectar el origen del problema en caso de no ser de la propia máquina
- Notificaciones a los contactos cuando ocurren problemas en servicios o hosts, así como cuando son resueltos (a través del correo electrónico, buscapersonas, Jabber, SMS, o cualquier método definido por el usuario junto con su correspondiente complemento).
- Posibilidad de definir manejadores de eventos que ejecuten al ocurrir un evento de un servicio o host para resoluciones de problemas proactivas.
- Log de eventos
- Interface Web para la visualización de estados de servicio, históricos, Archivo de Log, etc
- Integración con herramientas que la comunidad ha desarrollado
- Multiplataforma, aunque fue desarrollado originalmente para correr sobre Linux
- Visualización del estado de la red en tiempo real a través de interfaz web, con la posibilidad de generar informes y gráficas de comportamiento de los sistemas monitorizados, y visualización del listado de notificaciones enviadas, historial de problemas, archivos de registros....

## 1.2. Beneficios de utilizar Nagios

- Supervisión Continua de la plataforma de TI
- Esto te permite mejorar los tiempos de disponibilidad de los servicios
- Alertar al equipo de TI ante alertas preventivas (Warning) o críticas (Critical)
- Reaccionar de manera preventiva y no reactiva ante los eventos que nagios detecte.
- Aumenta la productividad de las TIC
- Generar Reportes de los eventos.
- Planificar mantención de tu hardware o servicios.
- Planificar el cambio o renovación de la Infraestructura de TIC

## 1.3. Alternativas a Nagios

Algunas de las alternativas existentes a Nagios son:

- Pandora FMS[3] es un software de código abierto que sirve para monitorizar y medir todo tipo de elementos. Monitoriza sistemas, aplicaciones o dispositivos. Permite saber el estado de cada elemento de un sistema a lo largo del tiempo. Pandora FMS está orientado a grandes entornos, y permite gestionar con y sin agentes, varios miles de sistemas, por lo que se puede emplear en grandes clusters, centros de datos y redes de todo tipo.



Figura 2: Interfaz Pandora FMS

- Zenoss (Zenoss Core)[4] es una aplicación de informática de código abierto, plataforma para la gestión de red y servidores basada en el servidor de aplicaciones Zope. Zenoss Core provee una interfaz web que permite a los administradores de sistemas monitorear disponibilidad, inventario/configuración, desempeño y eventos.

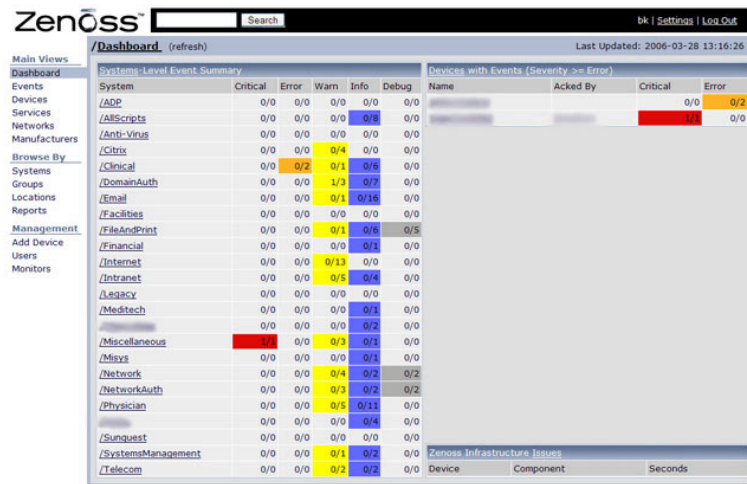


Figura 3: Interfaz Zenoss

## 2. Funcionamiento de Nagios

## 3. Instalación de Nagios

En este apartado se explica cómo instalar Nagios de dos formas, desde repositorio y también compilando su código fuente. Ambos métodos tienen como distribución objetivo CentOS/RedHat.

### 3.1. Instalación desde repositorio

Suponiendo que se tiene acceso root a la máquina, el comando a ejecutar como superusuario es el siguiente

```
yum install nagios*
```

esto se encargará de instalar Nagios, sus dependencias y los plugins. Los plugins nos permiten un mayor control de los sistemas a monitorizar así como una visualización más completa de la información.

### 3.2. Instalación desde código fuente

En primer lugar debemos crear el usuario y grupo al que pertenecerá la ejecución de Nagios. Esto se realiza con los siguientes comandos en modo superusuario

```
groupadd nagios
groupadd nagcmd
useradd nagios -g nagios
useradd nagios -g nagcmd
```

El usuario Nagios será el que tendrá en ejecución el daemon correspondiente. Además hemos creado dos grupos, **nagios** y **nagcmd**

- **nagios**: El grupo principal de Nagios.
- **nagcmd**: Este es el grupo que permite la ejecución de comandos externos a través de la interfaz web.

Ahora necesitamos instalar las dependencias, incluyendo el servidor web. De nuevo como superusuario ejecutamos lo siguiente

```
yum -y install httpd php glibc glibc-common gd gd-devel net-snmp
yum -y groupinstall "Development Tools"
```

**httpd** y **php** serán los encargados del servidor web. **glibc**, **glibc-common**, **gd** y **gd-devel** son librerías para el manejo de gráficos y **net-snmp** es el que incluye las librerías para manejar el protocolo SNMP, que nos permitirá monitorizar los ordenadores en red.

**Development Tools** instala todos los paquetes necesarios para poder compilar los códigos fuente. Incluye, como pequeño ejemplo, gcc y make.

Ahora ya estamos listos para descargar el código fuente. En el momento de escribir esta memoria la última versión disponible es la 4.0.8 y es la que se usará a lo largo de la memoria, pero los pasos deberían ser iguales o muy similares en las versiones posteriores.

En primer lugar, descargamos y descomprimos las fuentes de Nagios Core. Nagios Core es el engine que tiene toda la lógica de la aplicación, sobre él se añadirán plugins y demás para personalizarlo.

```
wget http://netcologne.dl.sourceforge.net/project/nagios/
      nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
tar -xzf nagios-4.0.8.tar.gz
```

Accedemos a la carpeta y realizamos un **./configure**

```
cd nagios-4.0.8
./configure
```

`./configure` comprueba que las dependencias del programa estén satisfechas (por lo menos las esenciales). En caso de no cumplirse algún requisito indispensable el programa saldrá con un mensaje de error indicando qué se debe instalar. Si el programa considera que se cumplen los requisitos mínimos generará un fichero **Makefile** con los comandos necesarios para compilar e instalar el programa.

Una vez el comando ha acabado ejecutamos

```
make all
make install
make install-init
make install-commandmode
make install-config
make install-webconf
```

- **all**: Compila el programa dejando los ejecutables preparados para ejecutarse desde local o para instalarse.
- **install**: Copia los ejecutables base a sus respectivas carpetas para que se puedan encontrar desde el `$PATH`.
- **install-init**: Este comando instala el fichero en `/etc/rc.d/init.d` para poder iniciarlo y pararlo de manera cómoda.
- **install-commandmode**: Instala y configura los permisos en el directorio para almacenar los ficheros externos de comandos.
- **install-config**: Este comando instala plantillas de configuración en `/usr/local/nagios/etc` para que sea más fácil su edición.
- **install-webconf**: Instala el fichero de configuración de Apache.

Ahora vamos a compilar e instalar los plugins desde código fuente, a fecha de escribir este artículo la última versión de los plugins es la 2.0.3

```
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
tar -xzf nagios-plugins-2.0.3.tar.gz
cd nagios-plugins-2.0.3
```

Una vez descargados procedemos a su configuración, compilación e instalación.

```
./configure
make
make install
```

Con esto tendríamos instalado Nagios y sus plugins desde código fuente, ahora sólo queda configurarlos. Esto se explica más adelante.

## 4. Configuración

### 4.1. Configuración de Nagios

En esta sección se explica cómo configurar Nagios paso a paso.

En primer lugar debemos crear los usuarios. En esta memoria crearemos un usuario administrador y otro que sólo pueda consultar la interfaz web.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users administrador
htpasswd /usr/local/nagios/etc/htpasswd.users usuario
```

En ambos casos se pedirá que se introduzca una contraseña, en nuestro ejemplo serán **labora2000** y **practicass** respectivamente. El parámetro -c usado en el primer comando crea el fichero si no existe, de existir lo sobrescribe.

El siguiente paso es cambiar el e-mail de contacto. Este e-mail será el que recibirá todas las notificaciones de Nagios. Para ello editamos el fichero /usr/local/nagios/etc/objects/contacts.cfg y cambiamos la siguiente línea para que contenga nuestro e-mail

```
email usuario@gmail.com; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
```

Ahora debemos decirle a Apache que cargue el módulo PHP o la web no funcionará correctamente. Para ello modificamos el fichero /etc/httpd/conf/httpd.conf y añadimos la línea

```
LoadModule php5_module modules/libphp5.so
```

con esto nos aseguramos de que funcione PHP en nuestro servidor web.

Por último configuramos los usuarios de Nagios. Para ello hay que editar el fichero /usr/local/nagios/etc/cgi.cfg

Hay que editar las líneas siguientes para incluir los usuarios que hemos creado

```
authorized_for_system_information=administrador,usuario
authorized_for_configuration_information=administrador
authorized_for_system_commands=administrador
authorized_for_all_services=administrador,usuario
authorized_for_all_hosts=administrador,usuario
authorized_for_all_service_commands=administrador
authorized_for_all_host_commands=administrador
```

Las líneas hacen lo siguiente:

- **authorized\_for\_system\_information:** Estos usuarios podrán ver los procesos e información de los sistemas en la interfaz web.



- **authorized\_for\_configuration\_information:** Estos usuarios podrán ver los comandos y la configuración de los sistemas a través de la interfaz web.
- **authorized\_for\_system\_commands:** Estos usuarios podrán apagar y reiniciar Nagios.
- **authorized\_for\_all\_services:** Estos usuarios podrán ver la información de todos los servicios monitorizados.
- **authorized\_for\_all\_hosts:** Estos usuarios podrán ver la información de todos los sistemas monitorizados.
- **authorized\_for\_all\_service\_commands:** Estos usuarios podrán ejecutar comandos sobre todos los servicios.
- **authorized\_for\_all\_host\_commands:** Estos usuarios podrán ejecutar comandos sobre todos los sistemas.

De esta manera sólo le damos poderes totales al usuario **administrador** y poderes de consulta al usuario **usuario**.

Con esto hemos acabado la configuración de Nagios, pero no es muy útil si no añadimos algunos hosts externos para monitorizar.

## 4.2. Configuración de los hosts externos

Para comenzar la configuración, crearemos el fichero `/usr/local/nagios/etc/objects/hosts.cfg` que contendrá todos los parámetros, hosts y servicios a configurar.

La plantilla básica que vamos a usar para configurar cada uno de los hosts es la siguiente

```
define host {

    host_name           PC_108
    alias               108 lab
    address             150.128.49.108
    check_command       check-host-alive
    max_check_attempts  3
    check_period        24x7
    notification_interval 180
    notification_period 24x7
    notification_options d,r,u
}
```

- **host\_name:** Un nombre corto y único para identificar el ordenador.

- **alias:** Un nombre largo o descripción para identificar el ordenador.
- **address:** La dirección IP
- **check\_command:** Comando que se ejecutará para saber si el host está disponible o no.
- **max\_check\_attempts:** El número de intentos que realizará Nagios para comprobar si un host no está disponible después de que este no le responda.
- **check\_period:** Indica el intervalo sobre el cual se puede hacer checks sobre el servidor.
- **notification\_interval:** El intervalo que tardará Nagios en decirnos que el host todavía no está disponible.
- **notification\_period:** Indica el intervalo sobre el cual se pueden enviar notificaciones a los usuarios.
- **notification\_options:** Indica cuándo enviar notificaciones. **d:** Indicar cuando el host no está disponible. **u:** Indicar cuándo el host no es accesible. **r:** Indicar cuándo se ha recuperado el servidor.

Tendremos un bloque como el anterior **para cada uno** de los hosts que pretendemos monitorizar.

El siguiente paso es el de configurar grupos, de manera que se pueda asignar la monitorización de un servicio al grupo entero en lugar de ir host por host. La plantilla es la siguiente

```
define hostgroup {

    hostgroup_name    grupolab1
    alias             Grupo laboratorio
    members           PC_108, PC_107, PC_105

}
```

- **hostgroup\_name:** Nombre por el que se identifica el grupo.
- **alias:** Pequeña descripción del grupo.
- **members:** Hosts por nombre que pertenecen al grupo.

Y por último la plantilla usada para definir los servicios que queremos monitorizar es la siguiente

```

define service{

    use                generic-service
    hostgroup_name      grupolab1
    service_description SSH
    is_volatile         0
    check_period        24x7
    max_check_attempts  4
    normal_check_interval 5
    retry_check_interval 1
    contact_groups      admins
    notification_options w,u,c,r
    notification_interval 960
    notification_period  24x7
    check_command        check_ssh

}

```

## 5. Acceso a Nagios mediante el navegador

Se debe explicar cómo se debe configurar el servidor web para acceder a Nagios a través del navegador y también la interfaz web que presenta Nagios.

## 6. Experiencia de instalación y configuración de Nagios

Se deben detallar los pasos que se han seguido para la instalación y configuración de Nagios tanto para monitorizar el sistema local como sistemas remotos.

## Referencias

- [1] Página web del proyecto de Nagios: <http://www.nagios.org>
- [2] Wikipedia Nagios: <http://es.wikipedia.org/wiki/Nagios>
- [3] Página web del proyecto de PandoraFMS: <http://pandorafms.com/>
- [4] Página web del proyecto de Zenoss: <http://www.zenoss.com/>