E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Entorno de gestión abierto para un laboratorio de redes de comunicaciones basado en software de monitorización NAGIOS y herramientas SNMP

(Open management environment for a communication networks laboratory based on Nagios monitoring software and SNMP tools)

Para acceder al Título de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Martín Pereira Diéguez

Octubre - 2015

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Martín Pereira Diéguez
Director del TFG: José Ángel Irastorza Teja

Título: "Entorno de gestión abierto para un laboratorio de redes de comunicaciones basado en software de monitorización NAGIOS y herramientas SNMP"
Title: "Open management environment for a communication networks laboratory based on Nagios monitoring software and SNMP tools "

Presentado a examen el día: 27 de octubre de 2015

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Bernardos Llorente, Pilar
Secretario (Apellidos, Nombre): García Gutiérrez, Alberto Eloy
Vocal (Apellidos, Nombre): Irastorza Teja, José Ángel

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente	Fdo.: El Secretario
Fdo.: El Vocal	Fdo.: El Director del TFG (sólo si es distinto del Secretario)
Vº Bº del Subdirector	Trabajo Fin de Grado Nº (a asignar por Secretaría)

AGRADECIMIENTOS

Me gustaría aprovechar estas líneas para agradecer a las personas que han contribuido a llegar a este momento, ya que sin su ayuda este documento no se habría hecho realidad.

A mi familia, de quien recibí el apoyo que necesitaba, en los momentos de flaqueza.

A mi pareja, por todo el ánimo y alegría que con sus palabras me ha infundido.

A mi tutor, por escucharme, aconsejarme y guiarme por el buen camino.

RESUMEN

Existen muchas razones para monitorizar los recursos de una red. Una de las principales razones es la mejora de la calidad de la misma. La utilización de una herramienta de monitorización detectará posibles fallos o incidencias y permitirá actuar en consecuencia con mayor rapidez. En ocasiones se puede tardar horas o días en obtener una notificación de un error, aunque muchos usuarios estén utilizando el servicio y estén experimentando errores. Es por ello necesario una herramienta que avise al administrador en estos casos. La gestión integrada de redes y sistemas, junto con una monitorización proactiva, reducirá significativamente los fallos de red. Nagios combinado con la obtención de datos a través SNMP, permitirá a alcanzar estas metas.

Palabras clave: Monitorización, Gestión integrada de redes y sistemas, Nagios, SNMP, Net-SNMP, Agente SNMP, SNMPTT, MG-SOFT, PNP4Nagios, NagVis.

ABSTRACT

There are many reasons to monitor network resources. A major reason is to improve the quality of it. Using a monitoring tool will detect possible failures or incidents and allow to act accordingly faster. Sometimes it can take hours or days to get error reports, although many users are using the service and are experiencing errors. It is therefore necessary a tool to alert the administrator in these cases. Integrated network and system monitoring, along with proactive network monitoring, will significantly reduce network failures. Nagios combined with SNMP data collection is used to achieve this goals.

Keywords: Monitoring, Integrated Network and Systems Management, Nagios, SNMP, Net-SNMP, SNMP Agent, SNMPTT, MG-SOFT, PNP4Nagios, NagVis.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. PREÁMBULO

1.1 Introducción a la gestión de redes y sistemas

Para empezar este documento, se exponen unas nociones previas y conceptuales de lo que es la gestión de redes y sistemas. Gestión, *es la tarea que cubre todas las precauciones y actividades que aseguren el uso eficiente y efectivo de procesos y recursos distribuidos, los cuales pueden constituir una red de comunicaciones o un sistema distribuido, dependiendo del objetivo de la gestión*. [1] Si se da prioridad a la gestión de los componentes de una red de comunicaciones hablaremos de gestión de red (*Network Management*); si por el contrario predomina la gestión de un sistema distribuido, hablaremos de gestión de sistemas (*System Management*). En la actualidad se tiende hacia un entorno de gestión donde los dos conceptos anteriores se funden en lo que se denomina la gestión integrada de redes y sistemas (*Integrated Network and Systems Management*).

La base del funcionamiento de los sistemas de gestión, reside en el intercambio de información entre los nodos gestores y nodos gestionados. Es lo que se llama paradigma gestor-agente. El gestor emite las directivas de operaciones de gestión y es quien recibe las notificaciones y respuestas. El agente tiene la función de responder a las directivas enviadas por el gestor y alertar de determinados eventos que surjan en los dispositivos o en la red.

La interacción entre el gestor y el agente reside en el uso de un protocolo de gestión, por ejemplo, SNMP (*Simple Network Management Protocol*), que no es más que un conjunto de normas o estándares que especifican el método para enviar y recibir datos entre varios equipos conectados entre sí. El otro pilar en la interacción es la base de información de gestión, la MIB (*Management Information Base*), que almacena los datos de los dispositivos o componentes de la red.

1.2 Motivación y objetivos

El presente proyecto tiene por objeto principal explicar detalladamente la herramienta de monitorización de red Nagios. Una de las más populares, debido a que es gratuita y con gran capacidad de personalización.

La monitorización integrada de redes y sistemas, es un concepto que nace con el objetivo de realizar un control sobre la red y sus dispositivos de manera que se puedan gestionar las incidencias de una forma más rápida y eficaz. En el entorno empresarial, apunta a la reducción de los costes generados por la aparición de errores en dichos equipos.

De aquí nace el concepto de monitorización proactiva, que es aquella en la que se toman medidas preventivas y consecuentes con la información que se obtiene de los dispositivos conectados a una red, para evitar posibles eventualidades que interrumpan el correcto funcionamiento de alguno de ellos.

Para poder llevar a cabo este concepto, es necesario un protocolo unificador que soporte la comunicación con la diversidad de equipos y fabricantes. Debido a dicha heterogeneidad, se realizará un acercamiento al protocolo SNMP, un estándar que soportan prácticamente todos los equipos de red para habilitar su gestión.

La posibilidad de la adaptación para cada entorno, es un aspecto fundamental de estas herramientas. La modificación de la topología de una red debe ser un obstáculo fácilmente superable.

Lo anteriormente expuesto, da lugar a una serie de objetivos más concretos que se perseguirán en este documento:

- Análisis del gestor de red Nagios
- Introducción al protocolo SNMP
- Análisis de las herramientas Net-SNMP, SNMPTT y MG-SOFT
- Instalación de Nagios y sus módulos adicionales
- Adaptación de Nagios al Laboratorio docente de Telemática de la Escuela de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria.
- Implementación de un agente SNMP para Windows
- Integración de la gestión de notificaciones SNMP en Nagios

1.3 Estructura del documento

La redacción del proyecto se ha dividido en cinco capítulos los cuales se muestran a continuación:

Capítulo 1: Preámbulo. En el presente capítulo se realiza una introducción que describe tanto las motivaciones que han propiciado la realización del proyecto, como los objetivos que persigue

Capítulo 2: Herramientas de monitorización. Se presentan las plataformas utilizadas en el escrito. Se detalla el funcionamiento de Nagios y de dos módulos adicionales. Se hace un pequeño repaso al protocolo SNMP y a la parte software del protocolo, gestores y agentes.

Capítulo 3: Instalación del software. El tercer capítulo recoge los aspectos referidos a la instalación del entorno de trabajo. En esta parte se describe los pasos dados para hacer operativa la interfaz de monitorización de sistemas.

Capítulo 4. Implementación en el laboratorio. Se recogen los procedimientos para establecer una configuración personalizada adaptada a los requerimientos del laboratorio. Además se explica cómo se implementa un subagente SNMP para Windows como un servicio del sistema para que interactúe con Nagios a través del envío de notificaciones.

Capítulo 5: Conclusiones y Líneas Futuras Finalmente se hace balance del proyecto y se reflexiona acerca de las posibles mejoras a realizar a corto y medio plazo.

2. HERRAMIENTAS DE MONITORIZACIÓN

Este capítulo contiene un análisis del gestor de red Nagios. Una visión general de las herramientas de monitorización que se han empleado para la realización del trabajo. Así como un acercamiento al protocolo SNMP y al software específico que funciona sobre él.

2.1 Nagios

Nagios, es una herramienta de monitorización de red de código abierto. Está licenciado bajo la *GNU General Public License Version 2*.

Como herramienta de monitorización, vigila que los equipos de la red funcionan como deberían. Nagios comprueba constantemente si los dispositivos funcionan correctamente. Es capaz de verificar si determinados servicios, en los diferentes equipos, están activos. Además, acepta los informes de estado de otros procesos o equipos, por ejemplo, un servidor web puede informar directamente a Nagios si no está sobrecargado.

La monitorización de sistemas en Nagios se divide en dos categorías de objetos: *hosts* o equipos y servicios. Los equipos representan un dispositivo físico o virtual en la red (servidores, routers, estaciones de trabajo, impresoras, etc.) Los servicios son funcionalidades específicas, por ejemplo, un servidor SSH (*Secure Shell*) puede definirse como un servicio monitorizado. Cada servicio se asocia con el equipo en el que está corriendo. Además, los equipos pueden asociarse en grupos de equipos (*hostgroups*) como muestra la *Figura 1*.

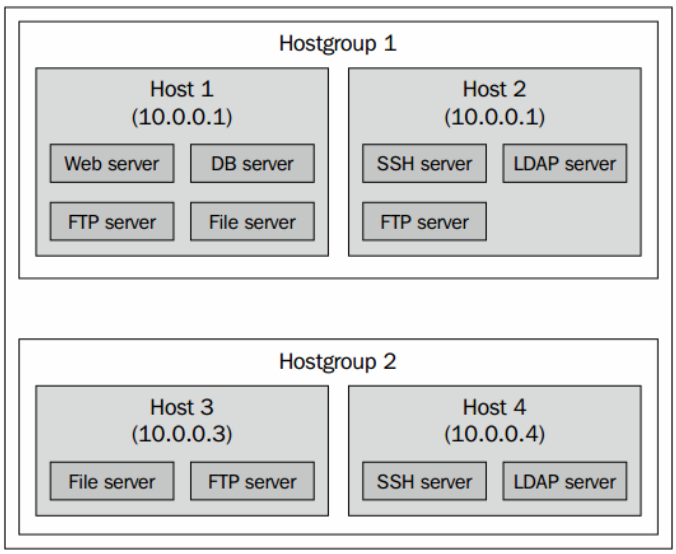


Figura 1 - Hostgroups

Un gran beneficio de las pruebas de rendimiento en Nagios es que solamente utiliza cuatro diferentes estados: “Ok”, “Warning”, “Critical”, y “Unknown”. Este sistema de comprobaciones está basado en plugins, lo que significa que si se quiere monitorizar algo que todavía no es posible, se puede escribir un pequeño código que lo consiga.

Desde el punto de vista de administrador, tener solamente tres estados, permite ignorar los valores de los mismos y determinar dónde se encuentran los límites de cada uno de los estados. Este concepto es mucho más eficiente que monitorizar gráficos y analizar tendencias. Por ejemplo, los administradores tienden a ignorar eventos como la disminución gradual del espacio en disco. A menudo se ignora este

hecho hasta que un proceso se queda sin espacio en el disco duro. Tener un límite estricto para controlar el espacio en un equipo, permite visualizar el problema si cambia de un estado de aviso a estado crítico.

Ésta es una de las ventajas de Nagios, cada comprobación realizada se convierte de valores numéricos a uno de los tres posibles estados para una identificar un problema con la mayor rapidez posible.

Nagios informa al administrador de forma unívoca del número de servicios que están en cada estado. Esto ahorra tiempo y es mucho más comprensible que visualizar una matriz de valores. Ahorra el tiempo de verificar qué funciona y qué falla. También ayuda a priorizar qué problemas necesitan ser solucionados antes y cuáles se solucionarán después.

Nagios realiza todas sus comprobaciones de estado a través de *plugins*. Éstos son componentes externos que pasan la información a Nagios sobre los servicios que deben comprobarse y sobre sus límites. Los plugins son responsables de la realización de las comprobaciones y de analizar los resultados. La salida de una comprobación (*output*), es el estado y un texto adicional describiendo la información del servicio en detalle.

Nagios incorpora un conjunto de plugins por defecto que permiten realizar comprobaciones de estado para una amplia gama de servicios.

2.1.1 Características

La principal característica de Nagios es su flexibilidad, ya que puede configurarse para monitorizar una red de muchas formas distintas. Incluso posee un mecanismo que reacciona a determinados problemas y con un sistema de notificaciones.

- **Comandos**: Son las definiciones de cómo Nagios debe realizar determinados tipos de comprobaciones.
- **Periodos**: Franjas temporales en las que una o varias operaciones deben realizarse o no.
- **Host y hostgroups**: Son los propios dispositivos a monitorizar que puede agruparse en conjuntos.
- **Servicios**: Son funcionalidades o recursos para monitorizar uno o varios hosts.
- **Contactos**: Personas a las que se debe notificar información sobre el estado de la red.
- **Notificaciones**: Éstas definen qué se le notifica a cada persona.

Una característica muy importante de Nagios es el sistema de dependencia. Este se basa en los niveles de dependencia entre los equipos de la red, es decir, qué equipo está conectado a qué otro u otros. Esto permite posteriormente reflejar la topología real de la red. Y de tal forma, Nagios no realizará una petición a un dispositivo dependiente de otro que se encuentre apagado, ya que no podrá ser fructífera si no existe otro camino. La *Figura 2* representa esta funcionalidad.

Otra característica útil es la programación de periodos planificados de mantenimiento. Durante estos periodos es posible que ciertos equipos no sean accesibles. Así se puede planificar el mantenimiento y evitar avisos e informes innecesarios durante el tiempo que dure la realización de las tareas en cuestión.

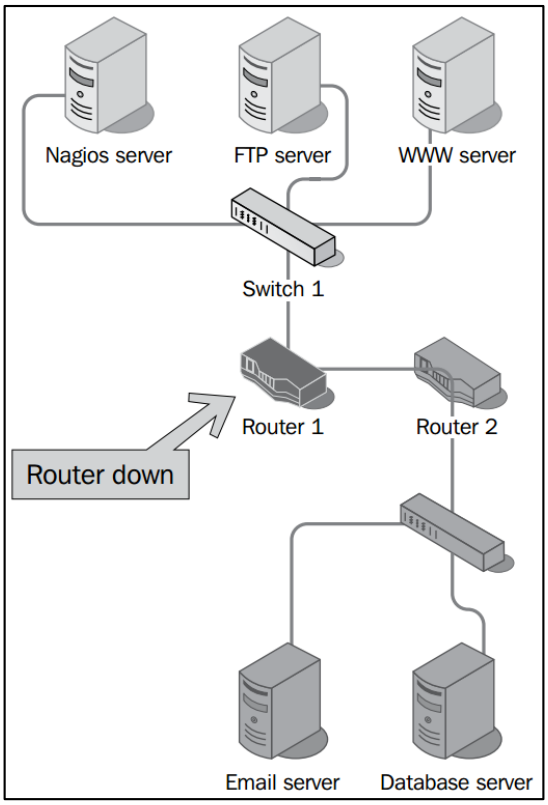


Figura 2 - Representación de la dependencia entre equipos [2]

2.1.2 Parámetros de configuración

El fichero de configuración principal se llama `nagios.cfg`, y es el archivo principal que se carga durante el arranque de Nagios. Tiene una sintaxis simple, la línea que comienza por `#` es un comentario, y todas las líneas de la forma `<parámetro>=<valor>` establecen un valor.

A continuación se muestra un fragmento del fichero de configuración de Nagios:

```
# LOG FILE
[...]
log_file=/usr/local/nagios/var/nagios.log

# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
```

El fichero de configuración necesita definir un archivo de log, y debe ser el primer parámetro que debe aparecer en dicho fichero. Los parámetros de Nagios permiten modificar su comportamiento y rendimiento. La *Tabla 8* en el *Anexo I* muestra algunos de los parámetros más usados en este fichero.

La opción `resource_file` define un fichero para almacenar variables de usuario. Este fichero puede usarse para guardar información adicional que puede ser accesible por las definiciones de los objetos. Normalmente contiene información sensible que no es accesible desde la interfaz web. Esto hace que sea posible ocultar contraseñas a usuarios de Nagios que no tengan privilegios de administrador.

Las opciones `cfg_file` y `cfg_dir` se usan para especificar los ficheros en los que se encuentran las definiciones de los objetos. La primera opción especifica un único fichero y la segunda un directorio en el que se leerán todos los ficheros con la extensión `.cfg`. Cada fichero puede contener diferentes tipos de objetos. En la siguiente sección se describirá cada tipo de las definiciones que Nagios puede utilizar.

2.1.2.1 Macros

La capacidad de utilizar en macros en Nagios es una característica clave. Ofrecen flexibilidad a la hora de definir objetos y comandos. Nagios proporciona macros personalizadas y existe además una amplia comunidad de desarrolladores que las proporciona gratuitamente en su página web.

Todas las definiciones de comandos pueden utilizar macros. Las definiciones de macros permiten parámetros de otros objetos, como hosts, servicios y contactos para hacer referencia de manera que un comando no necesita que todo se le pase como argumento. Cada invocación de una macro comienza y termina con un signo $.

Un ejemplo típico es la macro `HOSTADDRESS`, que hace referencia al campo de la dirección del objeto host. A continuación se muestra la definición de un objeto y un comando:

```
define host{
host_name           PC
address             10.0.0.2
check_command       check-host-alive
}
define command{
command_name check-host-alive
command_line $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80% -c
```

Nótese que la macro $USER1$ se utiliza para expandir el directorio de los plugins de Nagios.

Lo que en realidad produce es la ejecución de la siguiente instrucción:

```
/usr/local/nagios/libexec/check_ping -H 10.0.0.2 -w 3000.0,80% -c 5000.0,
100% -p 5
```

La *Tabla 9* en el *Anexo I* muestra algunas de las macros más utilizadas.

2.1.2.2 Hosts

Los “hosts” son objetos que describen las equipos que deben ser monitorizados, que pueden ser físicos o máquinas virtuales. La definición de un host consta de un nombre corto, un nombre descriptivo y una dirección IP o nombre de dominio. También le dice a Nagios cuándo y cómo debe monitorizarse, así como con quién debe ponerse en contacto si surge un problema.

Un ejemplo de definición de un host puede ser el siguiente:

```
define host{
       host_name               windowspc01
       hostgroups              windowsmachines
       alias                   Windows PC 01
       address                 192.168.0.5
       check_command           check-host-alive
       check_interval          5
```

```
        retry_interval          1
        max_check_attempts      5
        check_period            24x7
        contact_groups          windows-admins
        notification_interval   30
        notification_period     24x7
        notification_options    d,u,r
}
```

El código anterior, define un PC Windows que utilizará el comando `check_host_alive` para comprobar si el PC está encendido. Esta comprobación se realizará cada cinco minutos y tras cinco comprobaciones fallidas se asumirá que el PC está apagado. Si está apagado, se enviará una notificación cada treinta minutos.

En la Tabla 10 del *Anexo I* muestra las palabras clave que se pueden utilizar para describir hosts.

Una directiva importante para el desarrollo del proyecto es la directiva `parents` y se usa para definir la topología de la red. Normalmente esta directiva apunta a un switch, router u otro dispositivo responsable del reenvío de paquetes por la red. Un host se considera inalcanzable si su dispositivo padre se encuentra apagado. El dispositivo padre de todos los equipos de la *Figura 3* es el elemento de interconexión entre todos ellos, el switch.

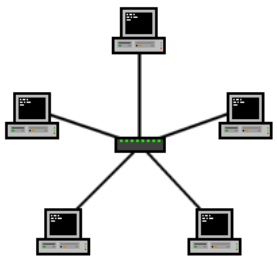


Figura 3 - Topología en estrella

2.1.2.3 Grupos de hosts

Nagios permite agrupar múltiples hosts para monitorizarlos de manera más efectiva. Para ello se utilizar la directiva `hostgroup` que agrupa a uno o más hosts. Además, un host puede ser miembro de uno o más grupos. Normalmente la agrupación se hace por tipo de máquinas, localización o el papel que juega el equipo. Cada grupo tiene un nombre corto y único que lo identifica, un nombre descriptivo del grupo, y sus miembros.

El siguiente código muestra un ejemplo de definiciones de grupos de hosts:

```
define hostgroup{
        hostgroup_name          routers ; The name of the hostgroup
        alias                   Routers ; Long name of the group
        members                 cisco2500,cisco2600
}
define hostgroup{
        hostgroup_name          hosts
        alias                   PCs
        members                 remoto1,local1,remoto2,local2
}
define hostgroup{
        hostgroup_name          unix-servers
        alias                   UNIX servers
        hostgroup_members       hosts, routers
}
```

La *Tabla 1* muestra las directivas que se pueden utilizar para definir grupos de hosts:

Opción	Descripción
`hostgroup_name`	Nombre corto y único del grupos de hosts
`alias`	El nombre descriptivo del grupo de hosts
`members`	Lista de todos los miembros que pertenecen al grupo; separados por comas
`hostgroup_members`	Lista de todos los otros grupos de hosts cuyos miembros también lo son de este grupo; separados por comas

Tabla 1 - Directivas de definición de grupos de hosts [2]

2.1.2.4 Servicios

Los servicios son objetos que describen una funcionalidad que host en particular ofrece. Esto se refiere a prácticamente cualquier información o recurso (desde servidores que ofrecen SSH, recursos como espacio en disco, o carga de CPU).

Un servicio siempre está ligado al host en el que está funcionando. También se identifica por su descripción y debe ser única en el entorno del equipo.

Un servicio define además cómo y cuándo Nagios debe comprobar si funciona correctamente e informar a la persona o personas responsables de ese servicio en caso de que alguna anomalía ocurra.

A continuación se muestra un ejemplo que define un servicio ICMP que en un equipo remoto

```
define service{
       hostgroup_name                  hosts
       service_description             PING
       check_command                   check_ping!200.0,20%!600.0,60%
       normal_check_interval           5
       retry_check_interval            1
       check_period                    24x7
       max_check_attempts              5
       notification_interval           60
       notification_period             24x7
       contact_groups                  admins
}
```

La *Tabla 11* en el *Anexo II* muestra las posibles directivas para definir un servicio:

Nagios requiere que se defina al menos un servicio para cada host. Para crear un nuevo servicio o incluir uno descargado es necesario añadirlo en la ruta `/etc/nagios-plugins/config/` con la extensión `.cfg`.

2.1.2.5 Comandos

La definición de comandos describe cómo se realizan las comprobaciones de un host o servicio. Pueden definir cómo deben funcionan las notificaciones o la gestión de eventos.

Los comandos definidos en Nagios ofrecen información de cómo realizan las comprobaciones. Qué comandos ejecutar para comprobar si una base de datos está funcionando correctamente, cómo comprobar si el servidor WEB, SSH, o FTP funciona correctamente, o si el servidor DHCP está asignando direcciones IP correctamente. La ejecución de comandos sirve para que los usuarios conozcan la información de la red, o para tratar de recuperar un problema de forma automática.

Nagios no hace ninguna distinción entre los comandos proporcionados por el proyecto Nagios-Plugins y los comandos personalizados, ya sea creado por un tercero o por escrito por uno mismo, y ya que su interfaz es muy sencilla, es fácil crear comandos propios controles.

Los comandos se definen de una manera similar a otros objetos en Nagios. La definición de un comando tiene dos parámetros: el nombre y la línea de comandos. El primer parámetro es un nombre que posteriormente se utiliza para definir los controles y notificaciones. El segundo parámetro es el comando real que se ejecutará junto con todos los atributos.

Los comandos son utilizados por los hosts y los servicios. Definen qué comando del sistema se ejecuta a la hora de asegurarse de que un host o servicio funciona correctamente. Un comando se identifica por un nombre único.

Cuando se usa con otras definiciones de objetos, también puede tener argumentos adicionales y se utiliza signo de exclamación como delimitador. Un comando con parámetros tiene la siguiente sintaxis: `nombre_del_comando [!ARG1] [!ARG2] [!ARG3] [...].`

El nombre del comando suele ser el mismo que el del plugin que se ejecuta, pero puede ser diferente. La línea de comandos incluye definiciones de macro (como `$HOSTADDRESS$`). Los comandos también utilizan macros `$ARG1$, $ARG2$, ... $ARGX$` si un comando para un host o servicio pasa argumentos adicionales.

Se muestra a continuación un ejemplo que define un comando ping a un host para asegurarse de que funciona correctamente, no utiliza ningún argumento.

```
define command{
       command_name      check-host-alive
       command_line      $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80%
-c 5000.0,100% -p 5
}
```

Una breve definición de un host que utiliza el comando anterior:

```
define host{
       host_name         comtrend
       address           10.0.0.1
       check_command     check-host-alive
}
```

Dicha verificación se hace generalmente como parte de la comprobación de hosts. Esto permite a Nagios asegurarse de que una máquina está funcionando correctamente si responde a las peticiones ICMP. Los comandos permiten pasar argumentos, y así se ofrece una forma más flexible de definir comprobaciones.

El siguiente fragmento de código muestra cómo se define un comando para que admita parámetros:

```
define command{
       command_name      check-host-alive-limits
       command_line      $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c
$ARG2$ -p 5
}
```

Y la correspondiente definición del host:

```
define host{
       host_name         tplink
       address           10.0.0.2
       check_command     check-host-alive-limits!3000.0,80%!5000.0,100%
```

2.2 Monitorización SNMP

Simple Network Management Protocol (SNMP) está diseñado para monitorizar y administrar dispositivos conectados a una red. Su objetivo principal es crear una forma estandarizada para obtener y establecer los parámetros, independientemente del hardware subyacente. El protocolo permite la recuperación de información desde un dispositivo y el establecimiento de opciones, y cubre los medios para un dispositivo para notificar a otras máquinas sobre un fracaso.

En este capítulo, se expondrá lo que SNMP es y cómo funciona. También se explica cómo configurar SNMP en varios tipos de máquinas y cómo recuperar la información usando plugins de Nagios.

2.2.1 Protocolo SNMP

SNMP es un estándar de la industria y de apoyan los principales proveedores de hardware y software. Todos los sistemas operativos de uso común pueden proporcionar información a través de SNMP. Microsoft ofrece SNMP para su plataforma de Windows. Los sistemas UNIX tienen "demonios" (servicios) SNMP que reciben solicitudes de otras máquinas.

SNMP también ofrece una forma estandarizada y jerárquica de agrupar y acceder a la información, se llama *Management Information Base* (MIB), que define los atributos a los que se puede acceder y los tipos de datos asociados con ellos. Esto permite la creación de atributos que todos los dispositivos deben utilizar para proporcionar información sobre los parámetros estándar, como la configuración de la red, el uso, etc. También permite que los parámetros personalizados que se creen no interfieran con los datos de otros dispositivos.

La mayoría de los sistemas operativos vienen con varias utilidades que permiten la comunicación con otros dispositivos a través de SNMP. Estas utilidades se pueden utilizar para verificar que los atributos están disponibles en los dispositivos específicos y cuáles son sus valores en ese momento.

SNMP está diseñado de modo que sea fácil de implementar y puede proporcionar una manera uniforme para acceder a información en varias máquinas. Está diseñado de manera que el uso de los servicios SNMP sea mínimo. Esto permite que los dispositivos con almacenamiento y memoria muy limitados puedan utilizar el protocolo. SNMP utiliza *User Datagram Protocol* (UDP) que requiere menos recursos que TCP. Utiliza un solo paquete para realizar una operación de petición o respuesta, por lo que el protocolo utiliza pocos recursos de red.

Cada máquina que es administrada por SNMP tiene un servicio que responde a las peticiones de los equipos locales y remotos. Dicha aplicación se llama *agente*. Para los sistemas UNIX, por lo general es un *demonio* (servicio) que se ejecuta en segundo plano. Los dispositivos de red suelen tener un agente SNMP y uno o varios subagentes. El agente SNMP se comunica con el gestor de la red y el subagente recupera y actualiza los datos y se los da al agente para que éste se los comunique al solicitante.

Muchos dispositivos con sistemas embebidos tienen soporte SNMP nativo. En todos estos casos, un dispositivo necesita escuchar solicitudes SNMP y responder en consecuencia.

Todos los agentes suelen ser gestionados por una o más máquinas llamadas *gestores*. Este es un equipo que consulta agentes para obtener información y puede establecer sus atributos. Por lo general, se trata de una aplicación que se ejecuta en segundo plano, se comunica a través de SNMP y almacena la información.

De forma predeterminada, SNMP utiliza el puerto UDP 161 para comunicarse con el agente y el puerto 162 para enviar información desde el agente al gestor. Para utilizar SNMP, estos puertos necesitan ser reenviados correctamente por todos los routers de la red y no deben ser filtrados por los cortafuegos.

Hay dos tipos de comunicación utilizados por SNMP: la primera es cuando un gestor envía peticiones a un agente. Estos pueden ser peticiones `GET` en las que el gestor quiere obtener información de un agente. Si se desea modificar la información debe utilizarse una solicitud `SET`.

Otro tipo de comunicación es cuando un agente quiere notificar a un gestor un problema. En tales casos, se envía una *trap* SNMP. Un agente necesita saber la dirección IP del gestor a la que debe enviar la información. Un gestor tiene que escuchar de traps SNMP y debe reaccionar ante el problema. La *Figura 4* ilustra los posibles tipos de comunicación SNMP.

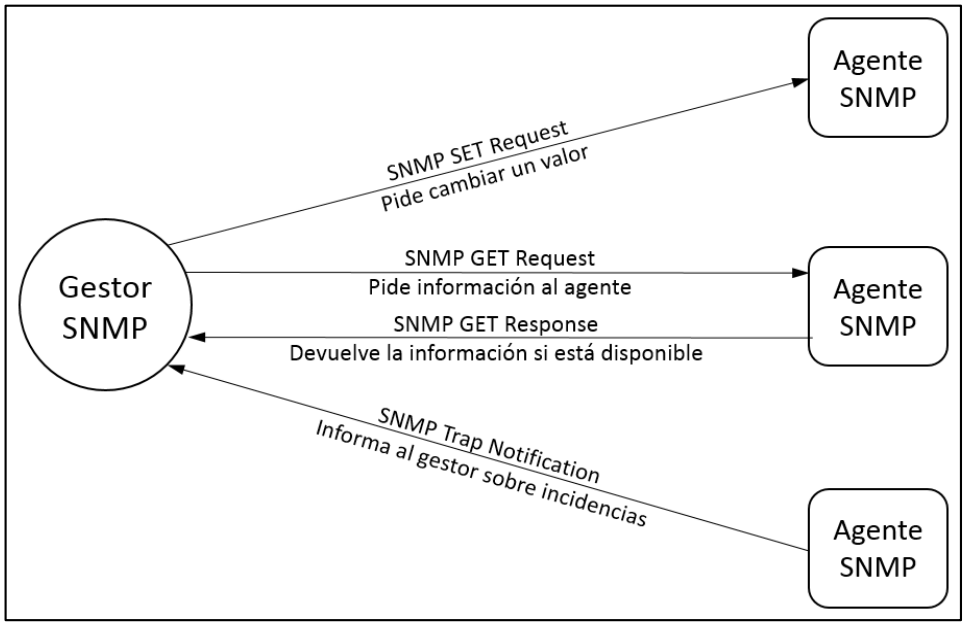


Figura 4 - Tipos de comunicación SNMP

SNMP tiene varias versiones a través de las cuales un agente puede comunicarse. SNMPv1 fue la primera versión del protocolo que incluyó operaciones `GET`, `SET` y `TRAP`. El estándar define los datos como objetos escalares, un solo valor, así como objetos tabulares, que forman una tabla de objetos. También definió la operación `GETNEXT` que permite iterar sobre las tablas de objetos.

El modelo de seguridad relacionada con SNMPv1 es relativamente poco sofisticado. Un `GET`, `SET`, o `GETNEXT` se autentifica en base a la dirección IP del gestor y la comunidad que utiliza. Todos los dispositivos SNMP que se comunican a través de SNMPv1 utilizan la comunidad para verificar que la solicitud puede ser realizada. Por defecto, la comunidad privada permite la lectura y escritura de información, mientras que comunidad pública sólo permite la lectura.

SNMPv2 introdujo mejoras en términos de rendimiento y seguridad. En lugar de utilizar `GET` y `GETNEXT`, se utiliza una operación `GETBULK` que permite la recuperación de todas las entradas de una tabla en una sola operación. También introdujo un paquete de notificación; se trata de una *trap* que requiere reconocimiento (ACK) del gestor. Esto aborda el problema de que un paquete UDP pueda perderse, evitando así una *traps* no recibidas por el gestor. Esta versión también introdujo un nuevo modelo de seguridad que no ganó una amplia aceptación debido a su complejidad.

La implementación más común de la versión 2 es *Simple Network Management Protocol* basado en comunidades (SNMPv2c). Utiliza las características de la versión 2, sin la implementación del nuevo modelo de seguridad, sino que utiliza el mecanismo de comunidades que se introdujo en SNMPv1.

SNMPv3 introduce un nuevo modelo de seguridad que incluye la autentificación, privacidad y control de acceso. Esta norma está ganando más atención que SNMPv2c, ya que ofrece una mayor seguridad que SNMPv2c.

La mayoría de las implementaciones del servidor SNMP que se integran con los sistemas operativos son compatibles con SNMPv1, SNMPv2c y SNMPv3. Algunos dispositivos sólo admiten SNMPv1, mientras que otros también ofrecen SNMPv2c. Los paquetes de diferentes versiones de SNMP son incompatibles, por lo que un dispositivo que soporte sólo SNMPv1 no reconocerá un paquete SNMPv2c.

2.2.2 Identificadores de objeto (OIDs)

SNMP utiliza *Object Identifiers* (OIDs) para identificar los datos de los objetos que se refiere. Los OIDs definen un objeto único para un agente SNMP específico. El objeto se identifica utilizando una definición jerárquica. Los identificadores de objeto son una serie de números separados por puntos. Cada número representa una parte del árbol. Un ejemplo de un OID es 1.3.6.1.2.1.1.5.0, que apunta al nombre de sistema de un equipo.

Como puede resultar difícil de memorizar, leer y comparar OIDs escritos como una serie de números, hay un estándar para nombrar y describir el árbol MIB. El estándar se llama *Management Information Base* (MIB). En él se describe cómo están definidos los diversos parámetros, cómo se llaman y qué tipos de valores pueden devolver estos objetos. Cada definición de MIB es un archivo de texto escrito en un subconjunto de la notación *Abstract Syntax Notation One* (ASN.1*)*. Un archivo puede describir un subconjunto pequeño o grande de los árboles MIB. Actualmente, el estándar es la MIB SMIv2 que define todos los atributos utilizados comúnmente junto con información adicional.

Las MIB describen los objetos que se pueden utilizar en SNMP, definen nodos padre en la jerarquía, el identificador numérico y el tipo de datos al que está asociado cada objeto.

SNMP utiliza los siguientes tipos de datos básicos:

- *String*: Una cadena, escrito como bytes, que puede tener 0 a 65535 bytes
- *Integer* / *Integer32*: Un valor entero de 32 bits con signo
- *Counters32* / *Counter64*: Enteros no negativos que se incrementan y se restablecen a 0 después de alcanzar el valor máximo
- *Gauges*: Estos son enteros no negativos que pueden aumentar y disminuir dentro de un rango definido
- *Timetick*: Tiempo: Esto define un lapso de tiempo, donde el valor de 100 representa un segundo
- *IP address*: Representa una dirección de la familia de protocolos IP; SNMPv1 sólo es compatible con IPv4, mientras que la versión 2 y 3 de apoyo IPv4 e IPv6

Un ejemplo es el OID `1.3.6.1.2.1.1.5.0`. La *Tabla 2* describe cada elemento, como número y como texto.

Identificador	Descripción
1	**iso**:Este es el árbol estándar ISO
3	**org**: *Organizations*; este nodo es un marcador de posición para todas las organizaciones nacionales e internacionales

Identificador	Descripción
6	**dod**: *Department of Defense*; este es el nodo para el Departamento de Defensa de EE.UU.
1	**internet**: subnodo para Internet; ya que Internet fue originalmente un proyecto para la defensa militar de Estados Unidos, su OID está bajo el subárbol dod
2	**mgmt**: Este es el nodo de administración de sistemas
1	**mib-2**: Este es el nodo raíz la versión 2 de la Base de Información de Gestión
1	**system**: Esta es la información del sistema operativo
5	**sysName**: Este es el nombre de la máquina
0	Este es un índice de los elementos; en este caso, siempre es 0

Tabla 2 - Descripción de identificador de objeto

La representación en texto del OID anterior es `iso.org.dod.internet.mgmt.mib-2.system.sysName.0`. También se representa como `SNMPv2-MIB::sysName.0`.

La parte del OID `1.3.6.1.2.1` define los elementos raíz para todos los parámetros estandarizados de la MIB-2. Todos los parámetros SNMP estandarizados que utilizan los dispositivos están bajo este OID o sus descendientes. Este nodo también se llama el espacio de nombres `SNMPv2MIB`; de ahí que el mismo OID se pueda representar también como `SNMPv2-MIB::sysName.0`.

El árbol MIB tiene unos pocos nodos principales que son las bases para muchos otros subárboles que pueden ser importantes para usted bajo varias circunstancias, que son los siguientes:

- `1.3.6.1.2.1`: Esto significa que `iso.org.dod.internet.mgmt.mib-2`, es la base de todos los atributos que están disponibles en la mayoría de los dispositivos SNMP.
- `1.3.6.1.4.1`: Esto significa que `iso.org.dod.internet.private.enterprise`, es el nodo raíz para todas las corporaciones y empresas que utilizan objetos privados.

El nodo más importante es `1.3.6.1.2.1`, que es utilizado por todos los dispositivos SNMP. Esta parte del árbol MIB es el nodo raíz para la mayoría de los objetos estándar. También es obligatorio para todos los dispositivos SNMP proporcionar al menos la parte básica de la información en este subárbol.

Por ejemplo, información de contacto, ubicación, nombre del sistema y tipo, deben ser proporcionadas por todos los dispositivos SNMP.

SNMP se puede utilizar para recuperar diferentes tipos de información. Esta información suele agruparse en diversas categorías. Todas las categorías también tienen sus correspondientes alias con los que se hace referencia para evitar poner toda la estructura en cada definición de OID o el nombre de la MIB. Todas las aplicaciones que ofrecen la comunicación a través de SNMP permiten la especificación de atributos utilizando tanto OID como nombres MIB.

La información contenida en IF-MIB, IP-MIB, IPv6-MIB, RFC1213-MIB, IP-FORWARD-MIB, TCP-MIB, y UDP-MIB describe la conectividad de los interfaces de red, configuración IP, enrutamiento, *forwarding*, y los protocolos TCP y UDP. Permiten la consulta de la configuración actual, así como los sockets activos y a la escucha.

Los datos contenidos en SNMPv2-MIB y HOST-RESOURCES-MIB describen la información del sistema. Esto puede incluir información sobre el almacenamiento en disco, los procesos actuales, aplicaciones instaladas, y el hardware con el que está funcionando el equipo.

2.2.3 Net-SNMP

Los diferentes sistemas operativos pueden venir con diferentes aplicaciones SNMP. Para la monitorización SNMP con Nagios se va a utilizar el paquete de Net-SNMP (http://netsnmp.sourceforge.net/). Este paquete está incluido en todas las distribuciones de Linux y funciona con casi todos los sistemas operativos UNIX. El paquete Net-SNMP está sujeto a varias licencias (`CMU/UCD, Networks Associates Technology, Inc, Cambridge Broadband Ltd., Sun Microsystems, Inc`., etc) que permiten, todas ellas, usar, copiar, modificar y distribuir el código fuente o los archivos binarios, siempre que se mantenga el aviso de *copyright*.

El primer comando importante de Net-SNMP es `snmpget`. Permite la consulta de uno o varios atributos de SNMP. La sintaxis del comando es la siguiente:

```
snmpget [OPTIONS] AGENT OID [OID]...
```

Los comandos Net-SNMP aceptan un gran número de parámetros. La *Tabla 3* muestra los más comunes:

Opción	Descripción
`-h`	Muestra la ayuda
`-V`	Imprime la versión de Net-SNMP
`-c`	Especifica el nombre de comunidad que se utilizará
`-v`	Especifica la versión de SNMP que se utilizará: 1, 2c, o 3
`-m`	Carga la lista de MIBs dada (`All` carga todas)
`-r`	Especifica el número de reintentos
`-t`	Esto indica el tiempo de espera en segundos
`-O`	Esto denota opciones de salida y debe ser uno o más de los siguientes: n: Imprime OID como valores numéricos sin ampliar desde MIB e: Imprime enumeración y campos OID como números en lugar de cadena valora v: Imprime solamente los valores, en lugar del nombre = formato de valores f: Imprime nombres OID completos y no permite atajos como SNMPv2-MIB

Tabla 3 - Opciones de la instrucción snmpget [3]

Net-SNMP también ofrece un comando para recorrer todo el árbol MIB o sólo una parte de él. El comando `snmpwalk` acepta las mismas opciones que las mostradas anteriormente. El comando snmpwalk de Net-SNMP no requiere un OID para funcionar.

El siguiente comando enumerará todo el árbol MIB de un agente SNMPv1:

```
snmpwalk -v 2c -c public 10.0.0.254
```

Dependiendo del sistema operativo subyacente y el agente de SNMP en sí, los datos pueden ser diferentes.

Para recuperar solamente una parte del árbol de la MIB hay que pasarle el OID prefijo correspondiente a la parte en cuestión. El siguiente comando muestra un ejemplo:

```
snmpwalk -v 1 -c public 10.0.0.254 1.3.6.1.2.1.1
```

El comando anterior limita la consulta al nodo `iso.org.dod.internet.mgmt.mib-2.system` y sus hijos. Esta consulta se completará mucho más rápido que la de todo el árbol.

Otra herramienta útil es el comando `snmptable`. Permite el listado de diversas tablas SNMP y las muestra en un formato legible. La sintaxis es la siguiente:

```
snmptable [OPTIONS] AGENT TABLE-OID
```

Por ejemplo, para listar todas las conexiones TCP / IP, el siguiente comando se puede utilizar:

```
root@ubuntu:/# snmptable -v 1 -c public 10.0.0.110 TCP-MIB::tcpConnTable

SNMP table: TCP-MIB::tcpConnTable

 tcpConnState tcpConnLocalAddress tcpConnLocalPort tcpConnRemAddress
       listen             0.0.0.0               22           0.0.0.0
       listen             0.0.0.0               53           0.0.0.0
       listen             0.0.0.0               80           0.0.0.0
       listen             0.0.0.0               81           0.0.0.0
       listen             0.0.0.0              139           0.0.0.0
       listen             0.0.0.0              445           0.0.0.0
       listen             0.0.0.0            55555           0.0.0.0
```

Net-SNMP también permite la creación de nuevos valores de los objetos que se pueden utilizar para reconfigurar varios dispositivos, que se pueden realizar con el comando `snmpset`. La sintaxis de este comando es la siguiente:

```
snmpset [OPTIONS] AGENT OID TYPE VALUE [OID TYPE VALUE]...
```

Este comando acepta las mismas opciones que el comando `snmpget`. Con una sola instrucción `snmpset` es posible establecer valores en diferentes variables. Cada operación SET debe especificar el nuevo valor junto con el tipo de datos que se debe establecer. La *Tabla 4* muestra el parámetro necesario para cada tipo de valor.

Los tipos más comunes son String, Integer, y OID. Los dos primeros requieren la aprobación de un número o texto que el valor del objeto se debe establecer. Configuración de un tipo OID del objeto requiere proporcionar ya sea un identificador OID completa o una cadena que puede ser igualada por las definiciones MIB.

Opción	Descripción
`i`	Entero
`u`	Entero sin signo
`s`	Cadena de texto (String)
`x`	String hexadecimal - cada letra se representa como 2 dígitos hexadecimales
`d`	String decimal - cada letra se representa como 1-2 dígitos
`n`	Objeto nulo (NULL)
`o`	OID
`t`	Timeticks
`a`	Dirección IP
`B`	Serie de bits

Tabla 4 - Tipos de datos para la instrucción snmpset [3]

Un ejemplo de cómo configurar el nombre de contacto y el nombre de host de un sistema es el siguiente:

```
root@ubuntu:/# snmpset -v 2c -c private 10.0.0.1 SNMPv2-MIB::sysContact.0 s
administrador SNMPv2-MIB::sysName.0 s ADSL
SNMPv2-MIB::sysContact.0 = STRING: administrador
SNMPv2-MIB::sysName.0 = STRING: ADSL
```

En las definiciones de las MIBs se especifica qué atributos son de lectura y cuáles soportan la escritura. El uso de una herramienta gráfica para saber qué atributos se pueden modificar facilitará la configuración de dispositivos a través del protocolo SNMP.

2.2.3.1 SNMPtrapd

SNMPtrapd es un programa del paquete Net-SNMP que escucha, típicamente, en el puerto 162 TCP esperando mensajes *Trap* o *Inform* del protocolo SNMP. Estos mensajes son los que el agente envía al gestor SNMP cuando determinados eventos ocurren en el dispositivo. Es necesaria la instalación de los módulos Perl de Net-SNMP para su correcto funcionamiento.

2.2.4 SNMPTT

SNMPTT (*SNMP Trap Traductor*) es un controlador de captura de traps SNMP multiplataforma escrito en Perl. Es un software gratuito bajo la licencia GPLv2.

Además puede guardar la información en cualquiera de los siguientes destinos: un fichero de texto, el log del sistema, el registro de eventos NT o una base de datos SQL. La trap traducida puede enviarse a programas externos [4].

SNMPTT se integra con `snmptrapd` y permite manipular las traps con mayor flexibilidad. El este caso será utilizado para traducir las traps y enviárselas a Nagios, quien las interpretará como servicios.

2.2.5 MG-SOFT

MG-SOFT es una empresa enfocada en la monitorización y desarrollo SNMP que vende su software bajo licencia comercial. Existen diferentes configuraciones de los paquetes de software. Se ha utilizado el paquete MIB Browser que incluye herramientas para el desarrollo y compilación de una MIB (MIB Builder y MIB Compiler respectivamente) y el SNMP Master Agent que sustituye el agente SNMP de Windows ofreciendo una interfaz gráfica más amigable para la configuración.

2.2.5.1 MIB Builder

MIB Builder es una herramienta que permite diseñar una MIB de forma gráfica añadiendo nodos y objetos a la MIB de manera sencilla. Soporta las versiones de SMIv1 y SMIv2 de la estructura de gestión de la información (*Structure of Management Information*). La estructura de la MIB la describe automáticamente añadiendo los nodos y objetos en ASN1 (Abstract Syntax Notation One). Una vez descrita permite exportarla para que pueda ser compilada. El fichero exportado escrito en ASN1 tiene la extensión `.my`. La *Figura 5* muestra la interfaz de MIB Builder.

2.2.5.2 MIB Compiler

El MIB Compiler de MG-SOFT permite, a partir de una MIB descrita en código ASN.1, generar una MIB con el formato `.smidb`, y la hace compatible con otros programas del paquete software. Este software se utilizará para compilar la MIB creada por el MIB Builder para poder identificar los objetos del subagente gestionado por el SNMP Master Agent. La *Figura 6* muestra la interfaz de MIB Builder.

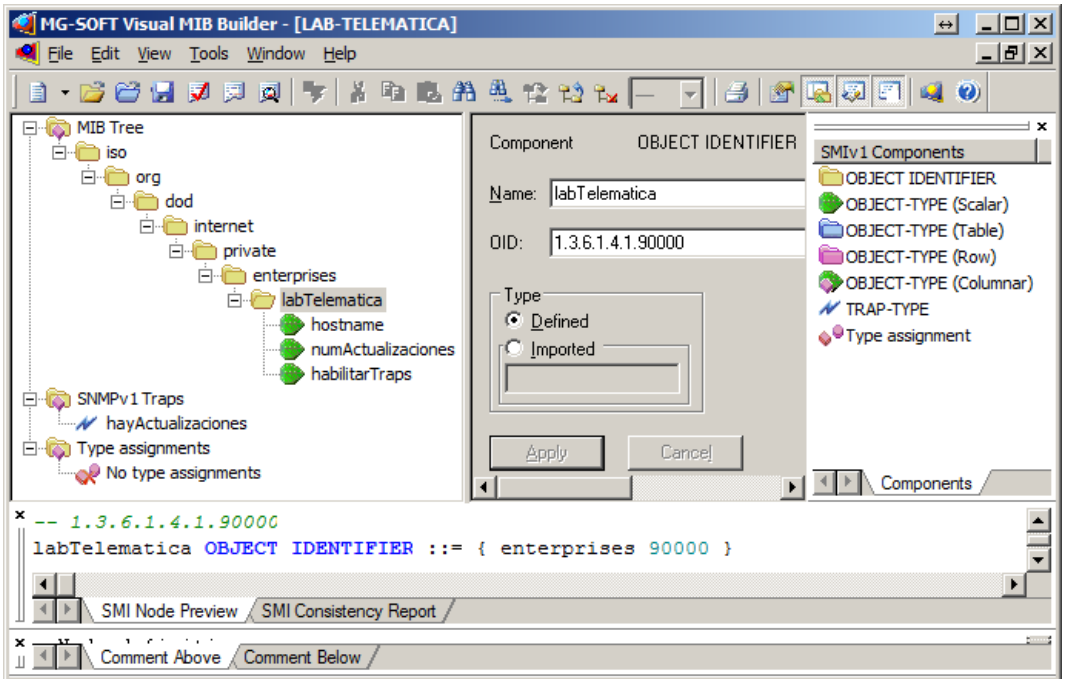


Figura 5 - Interfaz de MG-SOFT MIB Builder

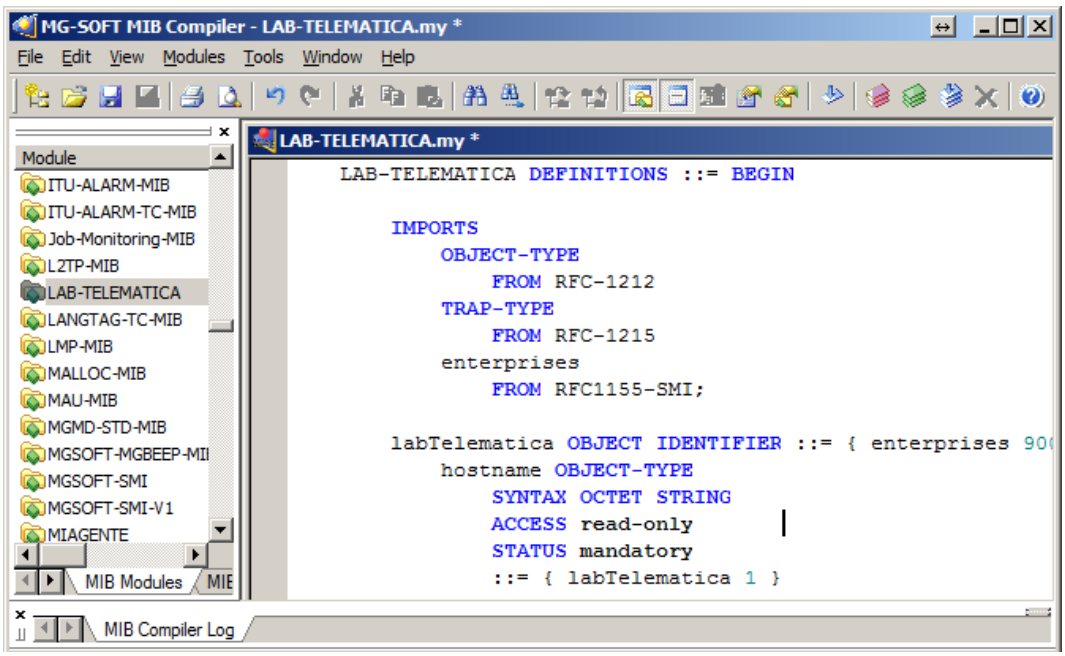


Figura 6 - Interfaz de MG-SOFT MIB Compiler

2.2.5.3 SNMP Master Agent

El agente maestro SNMP de MG-SOFT es un software que reemplaza de forma transparente al servicio SNMP que se ejecuta en los sistemas operativos Microsoft Windows.

Este agente SNMP implementa un interfaz API compatible con la API SNMP de Microsoft, lo que significa que todos los subagentes SNMP diseñados e implementados para ejecutarse en el servicio SNMP de Microsoft continuarán funcionando también bajo este agente maestro, sin la necesidad de modificarlos o recompilalos. [5]

Posee una interfaz mínima que permite la configuración de las diferentes versiones del protocolo así como añadir MIBs para poder identificar los objetos de un nuevo subagente. En la *Figura 7* se muestra la interfaz de configuración.

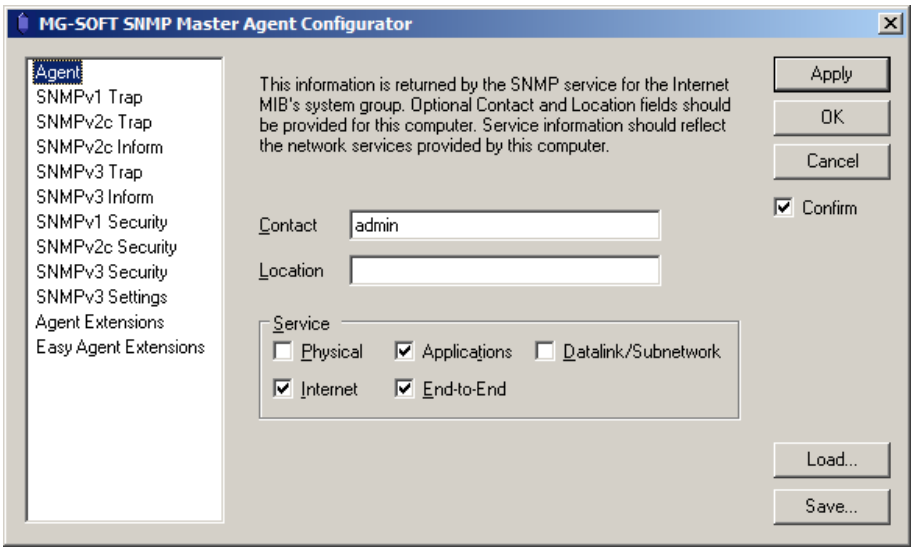


Figura 7 - Interfaz de configuración de MG-SOFT SNMP Master Agent

2.3 Módulos adicionales de Nagios

2.3.1 PNP4Nagios

PNP es un complemento para Nagios que permite mostrar gráficos utilizando los valores que recogen los plugins de Nagios. Es una herramienta muy útil para visualizar históricos de cierta información.

Este *addon* analiza los datos de rendimiento obtenidos por los plugins y los almacena automáticamente en bases de datos RDD (*Round Robin Databases*).

PNP requiere de forma obligatoria datos de rendimiento válidos de los plugins de nagios. La salida de los plugins hasta la versión 2.x de Nagios estaba limitada a una línea. Cuando el plugin genera datos de rendimiento, éstos se dividen en 2 partes. El símbolo de la barra vertical (“ | ”) se usa como delimitador.

Ejemplo de salida del comando `check_icmp` : [6]

```
OK - 127.0.0.1: rta 2.687ms, lost 0% | rta=2.687ms;3000.000;5000.000;0;
pl=0%;80;100;;
```

Devuelve el texto a la izquierda del símbolo delimitador:

```
OK - 127.0.0.1: rta 2.687ms, lost 0%
```

Y los datos de rendimiento:

```
rta=2.687ms;3000.000;5000.000;0; pl=0%;80;100;;
```

Las bases de datos RRD son aquellas en la que se almacena la información con el algoritmo Round Robin sobrescribiendo el contenido más antiguo. De esta forma el tamaño de la base de datos siempre permanece constante.

PNP4Nagios utilizar las librerías GD de PHP para poder crear y manipular ficheros de imágenes en una variedad de diferentes formatos de imagen, incluyendo GIF, PNG, JPEG, WBMP y XPM. Este módulo permite además transferir flujos de imagen directamente al navegador. [7]

2.3.2 NagVis

NagVis es un plugin para Nagios que permite visualizar gráficamente todos sus objetos de monitorización (hosts, servicios, etc). Se puede utilizar además para visualizar los datos que recoge Nagios.

Es un sistema basado en plantillas o mapas personalizables con los que se puede ver toda la infraestructura de red y/o sus servicios de un solo vistazo. Utilizando los datos suministrados por el motor, actualizará los objetos colocados en los mapas en intervalos de tiempo para reflejar el estado actual. Estos mapas permiten organizar los objetos para mostrarlos en diferentes diseños:

Permite distintas plantillas para la personalización de los mapas, existe un método manual para la definición del mapa y otro automático "AutoMap", en el que él mismo decide teniendo en cuenta la directiva `parents` de la definición de los objetos la posición en el mapa.

Sus principales características son: [8]

- Visualización de los hosts y servicios individuales
- Muestra un resumen del estado de un host y todos sus servicios
- Mostrar sólo los problemas reales
- La visualización completa de los procesos de TI usando gráficos autodibujados
- Configuración web de los mapas

Para poder utilizar NagVis es necesaria la instalación de un "backend". Es un proceso que corre en la máquina y es capaz de comunicarse y utilizar los datos que Nagios almacena. Este *backend* se llama Livestatus o MK Livestatus haciendo mención a las iniciales de su desarrollador, Mathias Kettner.

La forma clásica de acceder a los datos de Nagios era a través del fichero `status.dat` que refleja el estado de todos los hosts y servicios. Este método se descartó por un alto consumo de recursos dependiendo del valor de actualización de Nagios.

Otro método consistía en actualizar bases de datos con la ayuda del módulo NEB (*Nagios Event Broker*). Con esta solución no es necesario parsear los datos del fichero `status.dat` ya que la información se envía por sockets UNIX y se almacena en bases de datos. Sus desventajas radican en la dificultad de implementación, en el rápido crecimiento de las bases de datos y, por consiguiente, en un elevado consumo de recursos cuando existen muchos objetos en Nagios.

La solución que propone Livestatus es hacer uso de la API de *Nagios Event Broker* y cargar un módulo binario en su proceso de Nagios. Livestatus no escribe activamente datos. En lugar de ello, abre un socket por el cual los datos pueden ser recuperados bajo demanda

Algunas de las ventajas son: [9]

- Utilizar Livestatus no supone ninguna carga mesurable de CPU. Sólo se necesita una cantidad muy pequeña de CPU cuando se producen las consultas, pero no es suficiente para bloquear Nagios.
- Livestatus no hace uso del disco duro cuando pregunta datos de estado.
- El acceso a los datos es mucho más rápido que el análisis de `status.dat` o una base de datos.
- Livestatus maneja bien grandes instalaciones, incluso más de 50.000 servicios.

3. INSTALACIÓN DEL SOFTWARE

A continuación se explica detalladamente el proceso de instalación de las herramientas utilizadas para la monitorización del entorno de trabajo.

3.1 Nagios

Nagios se puede instalar de dos modos. El primero es instalar Nagios de los paquetes binarios de la distribución de Linux que se esté ejecutando. La parte positiva es que los paquetes estarán actualizados en términos de seguridad por la distribución de Linux, la parte negativa es que puede que no esté disponible para todas las distribuciones.

El segundo método, y más recomendado, es compilar e instalar Nagios manualmente. Este tipo de instalación permite una mayor capacidad de configuración del sistema.

En la realización de este proyecto se ha optado por la segunda opción. Como se describió anteriormente se procederá a la instalación de Nagios en un equipo con Ubuntu. Esta guía servirá para otras distribuciones de la rama Debian.

3.1.1 Prerrequisitos de instalación

Compilar Nagios desde el código fuente requiere tener instalado un compilador del lenguaje C, la librería estándar de desarrollo en C. Además, serán necesarios los ficheros de desarrollo de OpenSSL para que los plugins basados en red sean capaces de comunicarse a través de una capa SSL. Los paquetes de desarrollo de MySQL y PostgreSQL también deberán instalarse para que las comprobaciones de la base de datos funcionen.

Como el objetivo de este apartado es compilar Nagios desde los ficheros fuente, es necesario además un compilador con diferentes herramientas. Éstos son `gcc`, `make`, `cpp`, y `binutils`. También se necesitan la librería de desarrollo de ficheros en C. Estas herramientas sueles estar instaladas, pero son indispensables para poder compilar Nagios.

Nagios por sí sólo no necesita una gran cantidad de archivos para realizar una instalación básica, pero se pueden añadir funcionalidades con software adicional.

Es necesario un servidor web que soporte scripts CGI (“Common Gateway Interface”) para poder utilizar la interfaz de Nagios. Apache es el que se recomienda en la página de la fundación (www.nagios.org) y además es uno de los más populares para una instalación en Linux. De todas formas, Nagios funciona con cualquier servidor web que soporte scripts CGI y PHP. En la realización de este proyecto se ha utilizado Apache.

Muchos de los plugins de Nagios están escritos en Perl, por ello es necesaria su instalación. Algunos de ellos requieren, además, el paquete `Net::Snmp` de Perl para comunicarse mediante el protocolo SNMP.

Se instalará del servidor web Apache 2 y Perl desde los paquetes del repositorio de Ubuntu. La mayoría de las instrucciones que se va a utilizar necesitan privilegios de `root` en la cuenta.

```
apt-get install gcc make binutils cpp libpq-dev libmysqlclient-dev
libssl1.0.0 libssl-dev pkg-config libgd2-xpm-dev libgd-tools perl libperl-
dev libnet-snmp-perl snmp snmpd apache2 apache2-utils php5 libapache2-mod-
php5 build-essential php5-mysql
```

Los nombres de los paquetes pueden ser diferentes para otras versiones del operativo.

Para comprobar que el servidor web Apache se ha instalado correctamente es necesario modificar la página de prueba que ofrece Apache introduciendo un pequeño bloque de código.

Se debe modificar el fichero `/var/www/html/testphp.php` con cualquier herramienta que Ubuntu ofrece (`gedit`, `nano`, `vi`, etc…) como muestra la *Figura 8*.

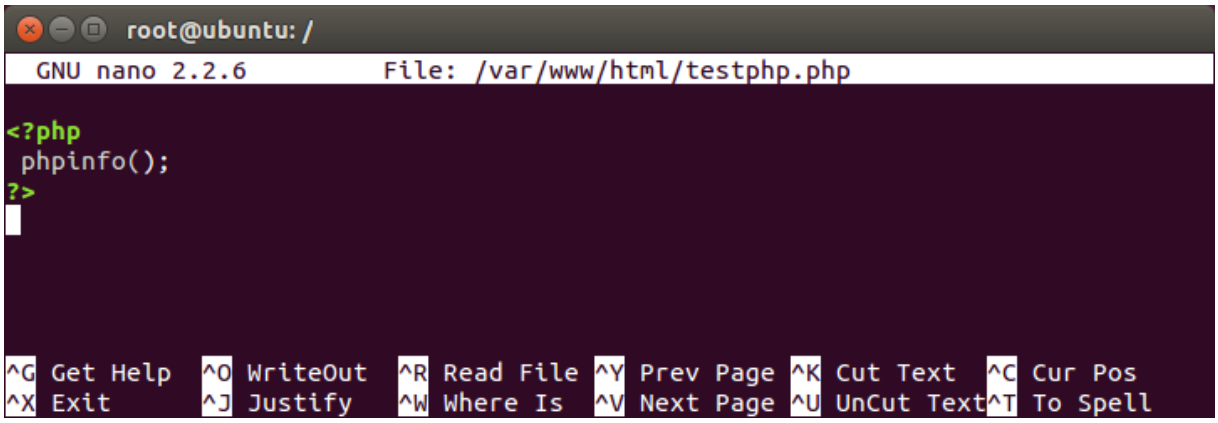


Figura 8 - Modificar el fichero testphp.php

Se guarda el fichero, y se abre un navegador web que apunte a nuestro bucle local, `http://localhost/testphp.php` o `http://127.0.0.1/testphp.php` como se muestra en la *Figura 9*.

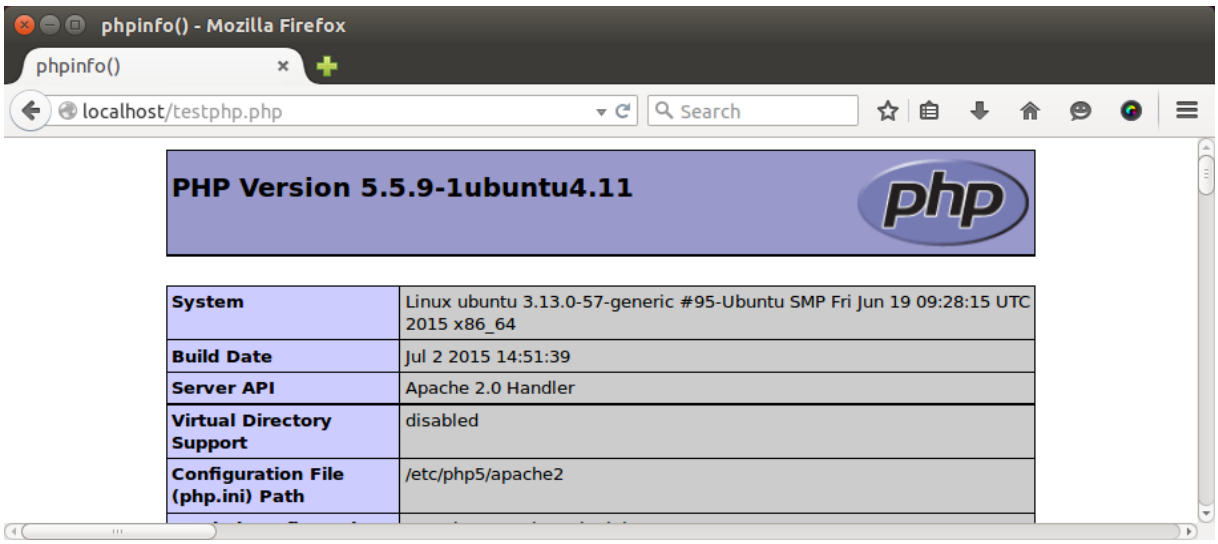


Figura 9 - Comprobar que Apache 2 funciona

3.1.2 Obtener el código fuente de Nagios

Nagios es un software de código abierto, lo que significa que el código fuente de todos los componentes de Nagios se encuentran de forma gratuita en su página web. Nagios se distribuye bajo la licencia GNU GPL ("General Plublic License") Versión 2 (para más información visite: `http://www.gnu.org/licenses/old-licenses/gpl-2.0.html`). Esto quiere decir que el código fuente de Nagios puede modificarse y redistribuirse de manera gratuita siempre que se distribuya como código fuente.

Nagios tiene un conjunto de plugins estándar, llamados Nagios plugins, que han sido desarrollados de forma independiente como un proyecto de SourceForge disponible en

`http://sourceforge.net/projects/nagiosplug/` o en su página `https://nagios-plugins.org/` y que se distribuyen bajo la licencia GPL Versión 3, que puede encontrarse en `http://www.gnu.org/licenses/gpl.html`.

Para obtener el código fuente de Nagios, hay que dirigirse a la sección de descargas de su página oficial `http://www.nagios.org/download/`.

Lo primero es descargar el núcleo de Nagios en la sección "Get Nagios Core" (véase *Figura 10*).

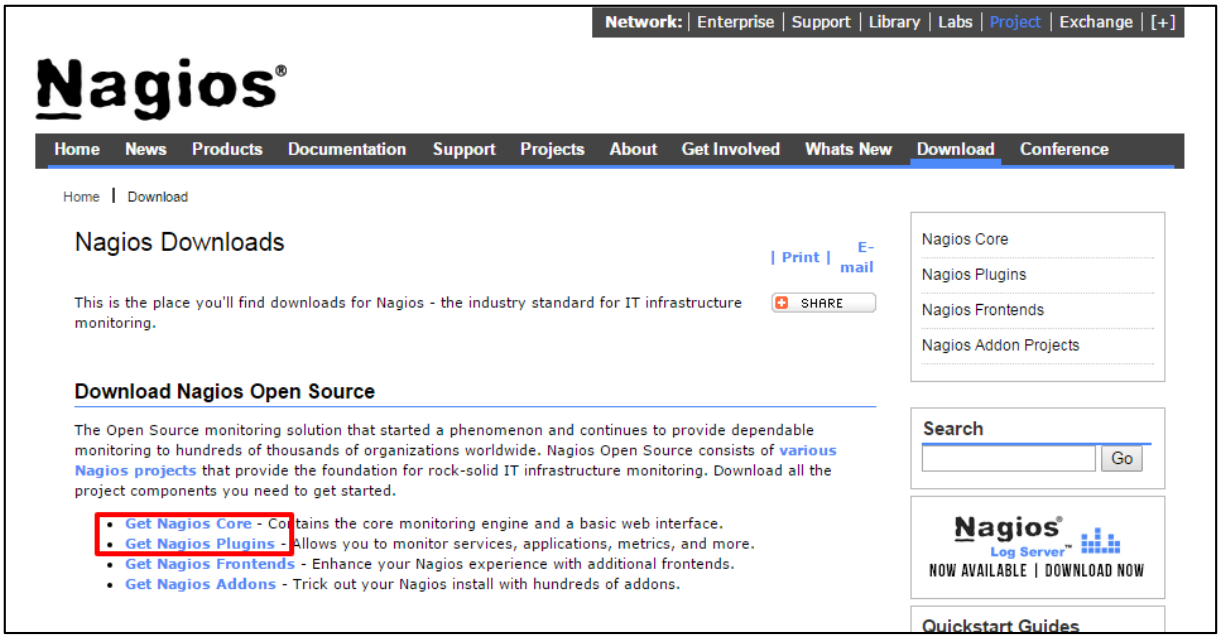


Figura 10 - Sección de descargas en la web de Nagios

Cuando pregunte la edición del software, elegir la versión gratuita de Nagios (véase *Figura 11*). El nombre del archivo TAR indica la versión del software. En este documento se utilizó la versión `nagios-4.0.8.tar.gz`.

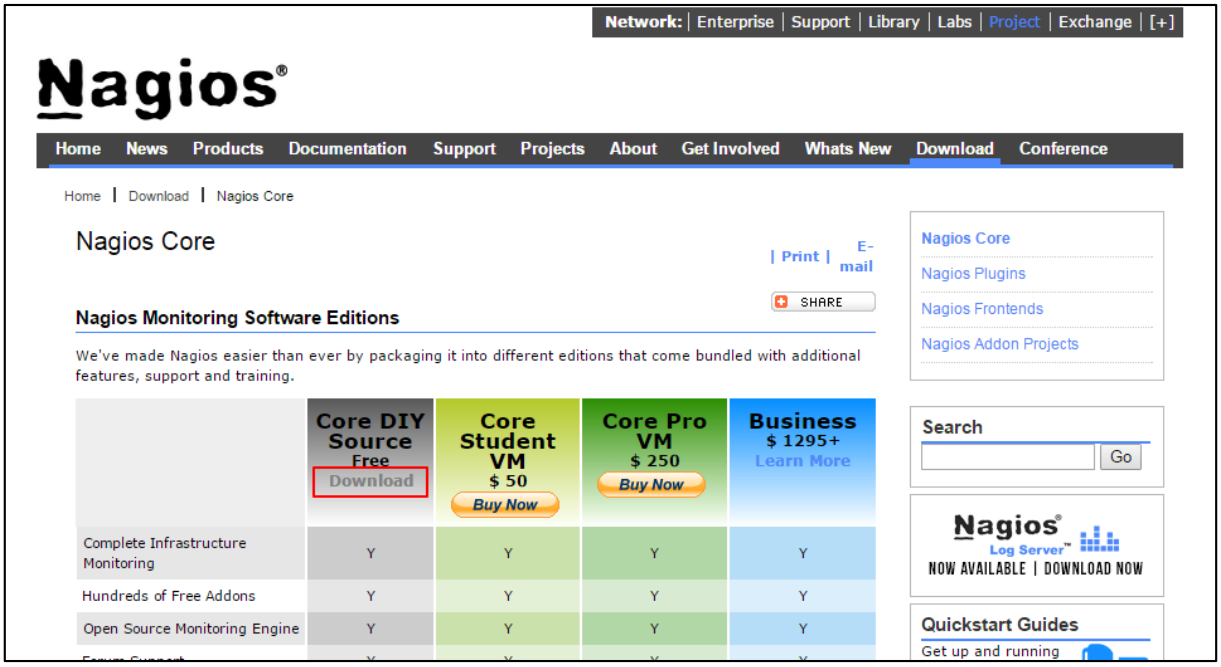


Figura 11 - Sección de descargas en la web de Nagios (II)

De igual forma se deben descargar el archivo TAR correspondiente a los plugins desde la misma página, están disponibles de en el enlace correspondiente a "Get Nagios Plugins". El nombre del archivo TAR indica la versión de los plugins. Durante la realización de este proyecto se utilizó la última versión estable de los mismos: `nagios-plugins-2.0.3.tar.gz`.

Para desempaquetar ambos ficheros, es recomendable crear un directorio; como referencia en este documento se utilizará uno perteneciente al Escritorio en Ubuntu: `/home/usuario/Desktop/nagios-source`. Para crearlo y desempaquetar los ficheros se utilizan las siguientes instrucciones.

```
mkdir /home/usuario/Desktop/nagios-source
cd /home/usuario/Desktop/nagios-source
tar -xzf nagios-4.0.8.tar.gz
tar -xzf nagios-plugins-2.0.3.tar.gz
```

3.1.3 Establecer usuarios y grupos

Por comodidad se recomienda establecer el usuario como "root" con la siguiente instrucción:

```
sudo –s
```

Y a continuación, introducir la contraseña del usuario con privilegios para poder realizar las siguientes operaciones. Nótese que mientras se escribe la contraseña no se visualizarán los caracteres en la pantalla.

Es necesario crear los usuarios y grupos para los datos de Nagios. También se debe crear un usuario de sistema y un grupo llamados `nagios` y `nagcmd` respectivamente, que se utilizarán como servicio del sistema ("daemon"). Los siguientes comandos crearán el usuario y el grupo: El comando `passwd` establece una contraseña en la cuenta recién creada.

```
useradd -m nagios
passwd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios
```

El motivo de la creación de usuarios de sistema es que Nagios utiliza un usuario separado. Lo que aumenta la seguridad y permite una mayor flexibilidad en la configuración. Nagios además se comunica con componentes externos a través de un socket Unix. Este socket funciona de forma parecida a un fichero en el sistema operativo. Entonces es necesario que los procesos tengan acceso al socket para que puedan realizar actualizaciones en Nagios. Un uso típico es que la interfaz web tiene que ser capaz de enviar comandos al proceso de monitorización de Nagios.

Si se quiere utilizar la interfaz web, como es el caso, será necesario añadir el usuario que utiliza el servidor web, en este caso Apache 2, al grupo anteriormente creado, `nagcmd`.

Para conocer cuál es el usuario que utiliza el servidor web es posible hacerlo con la siguiente instrucción:

```
root@ubuntu:~# grep -r ^User /etc/apache* /etc/httpd*
/etc/apache2/apache2.conf:User www-data
```

En el ejemplo se obtiene que el usuario que utiliza Apache es `www-data`.

En ocasiones en Ubuntu, el resultado es algo diferente:

```
root@ubuntu:~# grep -r ^User /etc/apache* /etc/httpd*
/etc/apache2/apache2.conf:User ${APACHE_RUN_USER}
```

En ese caso se muestra una referencia cuyo valor está definido en el fichero `/etc/apache2/envvars`.

```
root@ubuntu:~# grep APACHE_RUN_USER /etc/apache2/envvars
/etc/apache2/envvars:export APACHE_RUN_USER=www-data
```

En este caso el usuario también es `www-data`.

Ahora se añade este usuario al grupo `nagcmd`.

```
usermod -a -G nagcmd www-data
```

3.1.4 Compilar e instalar Nagios

Es posible establecer los directorios de los ficheros de instalación y configuración de Nagios, aunque en este caso se van a utilizar los que el script de instalación define por defecto.

Como en el apartado *3.1.2* ya se descargaron y desempaquetaron los ficheros fuente de Nagios y de sus plugins en una carpeta, se continúa con el proceso de compilación.

Tras haber desempaquetado los ficheros, se habrá creado una carpeta dentro de ésta, con el nombre `nagios-4.0.8` o similar, dependiendo de la versión descargada de la página de Nagios.

```
cd /home/usuario/Desktop/nagios-source/nagios-4.0.8
```

A continuación se configurará Nagios antes de compilarlo. El script de configuración admite los parámetros reflejados en la *Tabla 5*:

Opción	Descripción
`--prefix=<dir>`	Especifica el directorio principal en el cuál se instalarán los ficheros binario de Nagios, su valor por defecto es: `/usr/local/nagios`
`--sysconfdir=<dir>`	Especifica el directorio donde se almacenarán todos los ficheros de configuración de Nagios, su valor por defecto es `[PREFIX]/etc`
`--localstatedir=<dir>`	Especifica el directorio donde se almacenarán todos los ficheros de estado y otra información de Nagios, su valor por defecto es `[PREFIX]/var`
`--with-nagios-user=<user>`	Especifica el usuario de Unix que se usará el daemon de Nagios, su valor por defecto es nagios
`--with-nagios-group=<grp>`	Especifica el grupo de Unix que se usará el daemon de Nagios, su valor por defecto es nagios
`--with-mail=<path>`	Especifica la ruta que apunta al programa de correo electrónico a usar para enviar emails
`--with-httpdconf=<path>`	Especifica la ruta del directorio de configuración de Apache, puede usarse para generar los ficheros de configuración de Apache
`--with-initdir=<path>`	Especifica el directorio donde deben localizarse los scripts requeridos para configurar un servicio del sistema, su valor por defecto es /etc/rc.d/init.d

Tabla 5 - Opciones de compilación de Nagios [2]

Para configurar la compilación se utilizará la siguiente instrucción:

```
./configure --with-command-group=nagcmd
```

En el resto de opciones se utilizarán sus valores por defecto.

Este script puede que tarde un poco en completarse ya que recopilará información del sistema operativo y verificará como construir Nagios.

Si el script `configure` falla, lo más probable es que uno o más requisitos no estén presentes en el sistema. Si es el caso, se debe analizar cuál es el problema y solventarlo instalando o configurando paquetes adicionales. La mayoría de las veces la información que ofrece la salida es bastante clara y será sencillo identificar el problema.

Asumiendo que le script funcionó, mostrará un resumen de la configuración elegida:

```
[...]
Creating sample config files in sample-config/ ...

*** Configuration summary for nagios 4.0.8 08-12-2014 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  ${prefix}/var/nagios.lock
   Check result directory:  ${prefix}/var/spool/checkresults
           Init directory:  /etc/init.d
  Apache conf.d directory:  /etc/httpd/conf.d
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
[...]
```

El proceso de compilación utiliza el comando make, parecido a casi todos los programas en Unix. Las instrucciones de la *Tabla 6*, pueden usarse para compilar e instalar Nagios.

Opción	Descripción
`make all`	Compila Nagios; esto debe ser los primero que debe realizarse
`make install`	Instala el programa principal, el CGI y los ficheros HTML
`make install-commandmode`	Instala y configura el fichero externo de comandos
`make install-config`	Instalar las plantillas de los ficheros de configuración, esto sólo debería utilizarse para instalaciones limpias
`make install-init`	Instala los scripts para establecer Nagios como un servicio del sistema

Tabla 6 - Instrucciones para compilar Nagios [2]

Lo primero es compilar cada módulo de Nagios, para hacer esto simplemente ejecutar la instrucción:

```
make all
```

Si se produce algún error, es probable que sea debido a que falte algún fichero de cabecera o paquete de desarrollo no instalado. El siguiente código representa la salida de una compilación de Nagios correcta. Finaliza con un mensaje amigable que indica que se ha completado con éxito la operación.

```
cd ./base && make
make[1]: Entering directory `/home/usuario/Desktop/nagios-source/nagios-
[...]
*** Compile finished ***
```

Ahora, se procederá con la instalación de Nagios ejecutando los siguientes comandos:

```
root@ubuntu:~/nagios-4.0.8# make install
[...]
*** Main program, CGIs and HTML files installed ***
```

```
root@ubuntu:~/nagios-4.0.8# make install-init
[...]
*** Init script installed ***
```

```
root@ubuntu:~/nagios-4.0.8# make install-commandmode
[...]
*** External command directory configured ***
```

Para una instalación limpia, se recomienda instalar los ficheros de configuración de ejemplo que se usarán posteriormente para la configuración de Nagios.

```
root@ubuntu:~/nagios-4.0.8# make install-config
[...]
*** Config files installed ***
```

En este punto Nagios ya está instalado con la configuración escogida.

3.1.5 Compilar e instalar los plugins de Nagios

El siguiente paso que debe llevarse a cabo es compilar el paquete de los plugins de Nagios.

Es posible instalarlo desde los repositorios, al igual que Nagios, pero se ha escogido instalarlo desde el código fuente para tener siempre la última versión.

Para instalar los plugins manualmente es necesario posicionarse en la carpeta que contiene el código fuente desempaquetado.

```
cd /home/usuario/Desktop/nagios-source/nagios-plugins-2.0.3
```

La ruta dependerá de la versión que se haya descargado. Existen diferentes opciones de configuración para los plugins, se reflejan en la *Tabla 7*.

Opción	Descripción
`--prefix=<dir>`	Especifica el directorio principal en el cuál se instalarán los ficheros binario de Nagios, su valor por defecto es: `/usr/local/nagios`
`--sysconfdir=<dir>`	Especifica el directorio donde se almacenarán todos los ficheros de configuración de Nagios, su valor por defecto es `[PREFIX]/etc`
`--localstatedir=<dir>`	Especifica el directorio donde se almacenarán todos los ficheros de estado y otra información de Nagios, su valor por defecto es `[PREFIX]/var`
`--enable-perl-modules`	Instala el paquete `Nagios::Plugin` con todos los paquetes dependientes
`--with-nagios-user=<user>`	Especifica el usuario de Unix que se usará el daemon de Nagios, su valor por defecto es nagios
`--with-nagios-group=<grp>`	Especifica el grupo de Unix que se usará el daemon de Nagios, su valor por defecto es nagios
`--with-pgsql=<path>`	Especifica la ruta a la instalación de PostgreSQL; necesario para plugins que requieran PostgreSQL

Opción	Descripción
`--with-mysql=<path>`	Especifica la ruta a la instalación de MySQL; necesario para plugins que requieran MySQL
`--with-openssl=<path>`	Especifica la ruta a la instalación de OpenSSL; se puede especificar si la instalación no se realizó en el directorio estándar
`--with-perl=<path>`	Especifica la ruta a la instalación de Perl; se puede especificar si la instalación no se realizó en el directorio estándar

Tabla 7 - Opciones de compilación de los plugins de Nagios [2]

Para configurar la compilación de los plugins debe ejecutarse la siguiente instrucción:

```
./configure --with-nagios-user=nagios --with-nagios-group=nagcmd --enable-perl-modules
```

El script tardará un tiempo en realizar todas las comprobaciones y se completará con éxito si todos los prerrequisitos fueron correctamente instalados.

El proceso de compilación utiliza el mismo comando que el software de Nagios, el comando `make`. En este caso solamente se utilizaran las opciones `all` e `install`. De esto modo habrá que ejecutar los siguientes comandos:

```
make all
make install
```

Si la instalación ha sido correcta, ya se puede decir que se posee una instalación completa de Nagios.

El siguiente paso es comprobar que Nagios funciona correctamente después de su instalación. Para realizarlo es necesario ejecutar Nagios con la plantilla de configuración que creó el comando `make install-config`.

Se debe ejecutar Nagios como el usuario de sistema que se creó anteriormente, `nagios`, ya que el programa se va a ejecutar con ese usuario. Con la siguiente instrucción se comprueba su funcionamiento:

```
su -c '/usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg'

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Nagios 4.0.8 starting... (PID=42345)
Local time is Mon Aug 17 16:21:56 CEST 2015
nerd: Channel hostchecks registered successfully
nerd: Channel servicechecks registered successfully
nerd: Channel opathchecks registered successfully
nerd: Fully initialized and ready to rock!
wproc: Successfully registered manager as @wproc with query handler
wproc: Registry request: name=Core Worker 42347;pid=42347
wproc: Registry request: name=Core Worker 42348;pid=42348
wproc: Registry request: name=Core Worker 42349;pid=42349
wproc: Registry request: name=Core Worker 42346;pid=42346
Successfully launched command file worker with pid 42350
```

Este mensaje indica que el proceso se arrancó correctamente. Después de comprobar que Nagios se inició correctamente es necesario presionar la combinación de teclas `Ctrl + C` para detener el proceso.

3.1.6 Interfaz web

La interfaz web de Nagios es parte del código fuente de Nagios Core. Por lo tanto, al instalar Nagios, también se instalan los archivos de la interfaz web. Lo único que se necesita es un servidor web con CGI y soporte para PHP; en este caso será Apache 2. La Interfaz web de Nagios utiliza CGI (*Common Gateway Interface*), un estándar para generar sitios web dinámicos.

La interfaz web de Nagios también utiliza archivos adicionales, como páginas HTML estáticas, CSS e imágenes. A partir de Nagios 4, PHP (*PHP: Hypertext Preprocessor*), un lenguaje de programación para el desarrollo web, se utiliza para ayudar en la configuración de las páginas HTML de la interfaz web.

Por defecto, todos los ficheros HTML de Nagios y otros archivos estáticos utilizados por la interfaz web se copian en el subdirectorio `share` de la instalación de Nagios y todos los binarios CGI entran en el subdirectorio `sbin`. Suponiendo que Nagios se ha configurado utilizando los directorios predeterminados utilizados en el apartado *3.1.4*, `/usr/local/nagios`, las rutas de acceso de archivos serían `/usr/local/nagios/share` y `/usr/local/nagios/sbin` respectivamente.

Nagios utiliza archivos PHP para páginas estáticas. PHP se utiliza principalmente para dar formato a las páginas HTML y Nagios utiliza PHP para permitir la configuración de ubicaciones de ficheros y scripts, como las rutas del fichero de configuración de Nagios, el archivo de estado, el archivo de configuración CGI y la URL a los archivos CGI.

Con el fin de ejecutar la interfaz web, deben habilitarse los módulos CGI y de escritura en Apache con las siguientes instrucciones:

```
a2enmod rewrite
a2enmod cgi
```

Para poder acceder a la interfaz web debe configurarse Apache para utilizar la URL adecuada y crear un usuario válido que será capaz de acceder a Nagios.

Las siguientes instrucciones, suponiendo que la configuración de Apache está en `/etc/apache2` y que servidor web leerá todos los archivos de configuración en `/etc/apache2/sites-enabled`.

En los ficheros fuente de Nagios existe una plantilla para habilitar la interfaz web en Apache, para utilizarla ejecutar la siguiente instrucción en la carpeta que contiene el código fuente.

```
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-
enabled/nagios.conf
```

También es posible crear el fichero `nagios.conf` manualmente con un editor de texto en la ruta `/etc/apache2/sites-enabled/` y copiando el siguiente código:

```
ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">
#SSLRequireSSL
Options ExecCGI
AllowOverride None
Order allow,deny
Allow from all
#Order deny,allow
#Deny from all
#Allow from 127.0.0.1
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
```

```
</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">
#SSLRequireSSL
Options None
AllowOverride None
Order allow,deny
Allow from all
#Order deny,allow
#Deny from all
#Allow from 127.0.0.1
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
</Directory>
```

Para limitar los clientes que puedan acceder a la interfaz en función de la dirección IP, es necesario cambiar las directivas `Order` y `Allow` en ambas definiciones `Directory` de la siguiente forma:

```
Order Deny,Allow
Deny From All
Allow From 192.168.110.0/24
```

De esta forma sólo podrán acceder las direcciones IP que empiecen por 192.168.110.xxx

El último paso es crear el usuario administrador, que por defecto en Nagios es `nagiosadmin`, con el que se tendrá acceso a todas las funciones de la interfaz web de Nagios.

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

Tras ejecutar esta instrucción, pide que se introduzca una contraseña dos veces en las que no se verá la contraseña mientras se teclea.

En el PC que se ha instalado Nagios para la monitorización de los laboratorios se ha introducido la siguiente contraseña: `nagiOS*admin`

Tras reiniciar Apache y Nagios será posible acceder a la interfaz web.

```
service apache2 restart && service nagios restart
```

Con un navegador web que soporte CGI (Firefox, Chrome, Internet Explorer, etc) se debe acceder a la dirección IP del equipo, en local `http://127.0.0.1/nagios`, o en remoto, para el host con el que se está trabajando en este caso, `http://192.168.110.5/nagios` y se podrá visualizar la pantalla de inicio como se muestra en la *Figura 12*.

Nagios ofrece un panel que muestra un estado general de todos los hosts, servicios y otras características. Se puede acceder haciendo clic en el enlace de *Tactical Overview* en el menú de la izquierda. Es posible evaluar fácilmente la importancia de los problemas, visualizando el número de hosts y servicios que fallan, los que cambian de estado constantemente (*flapping*) y los que se encuentran en espera de comprobación como se muestra en la *Figura 13*. También muestra cuántos hosts son inalcanzables debido a que sus hosts "padres" no se encuentra activos. Como todavía no se han configurado ninguno de ellos, sólo aparece el host local (*localhost*) y los servicios que Nagios monitoriza en él por defecto.

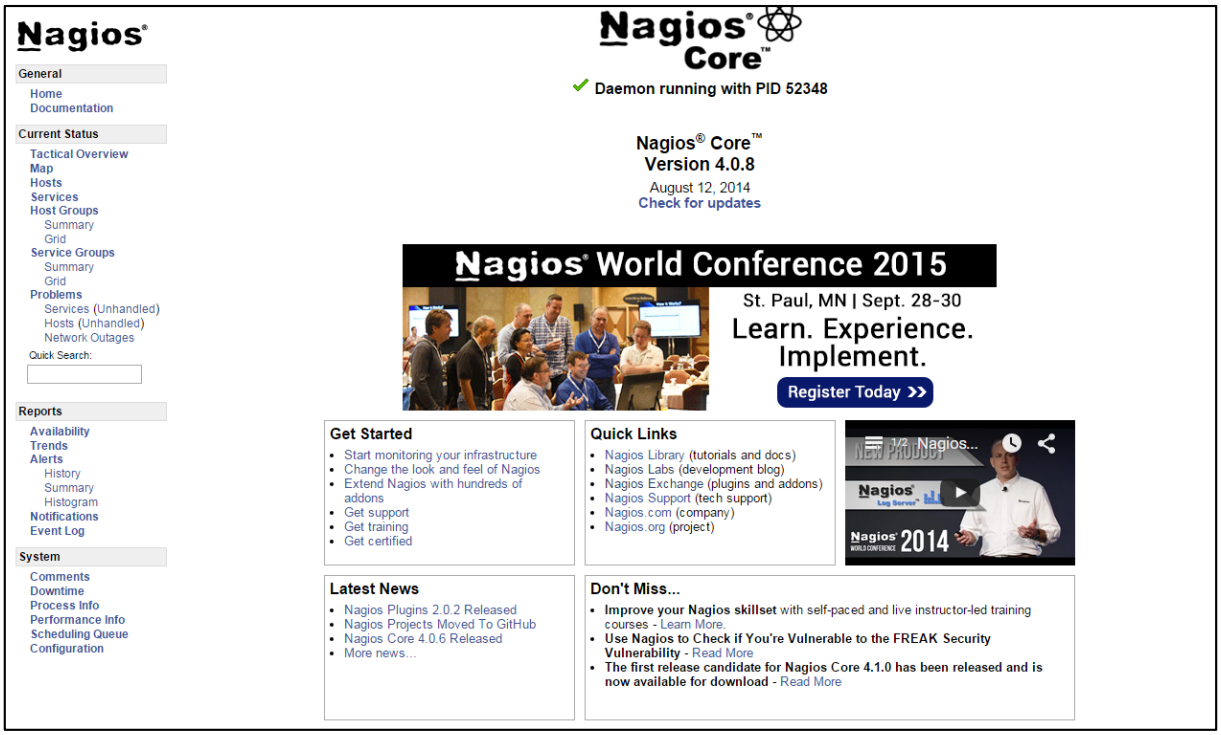


Figura 12 - Página principal interfaz web de Nagios

En el *Tactical Overview* se puede obtener información general sobre Nagios y monitorización. La página informa sobre los estados de los hosts y servicios. También muestra si los hosts y servicios tienen sus comprobaciones, notificaciones, o controladores de eventos deshabilitados.

La esquina superior derecha muestra la información de rendimiento. Muestra detalles sobre las comprobaciones que se han realizado, los informes latencias de las mismas y el tiempo medio que se tarda en realizar comprobaciones. Estos valores son muy importante porque si hay demasiadas comprobaciones programadas, Nagios puede no ser capaz de realizar algunas de ellas. Por lo general, debe ajustarse la configuración de Nagios en los casos en que la latencia sea mayor de dos segundos.

Debajo de la información de rendimiento se encuentra un gráfico que muestra la salud de los hosts y servicios. Contiene barras que muestran el número de hosts y servicios que se encuentran en el estado OK. Dado que todos los servicios están trabajando correctamente, la barra se extiende a todo lo ancho y es de color verde. Si algún host o servicio no está funcionando, el color de la barra cambia a amarillo o rojo en consecuencia.

El *Tactical Overview* también se puede utilizar para ver los hosts o listas servicios filtrados por criterios específicos. *Network Outages* muestras los hosts que no son accesibles porque un dispositivo "padre" está desconectado.

Nagios permite mostrar un mapa gráfico de host mediante las relaciones entre padres e hijos, junto con los estados. Se puede acceder haciendo clic en el enlace de *Map* en el menú lateral de la parte izquierda (véase *Figura 12*). Esto se puede utilizar para realizar un seguimiento de los hosts junto con sus estados de modo que puede apreciarse cómo un host fuera de línea causa que otras partes de la red sean inalcanzables. La *Figura 14* muestra el mapa de Nagios.

La página del mapa puede mostrarse de varias maneras. La captura de pantalla anterior muestra un árbol circular con todos los hosts definidos. Es posible mostrar también un árbol de arriba hacia abajo de todos los hosts.

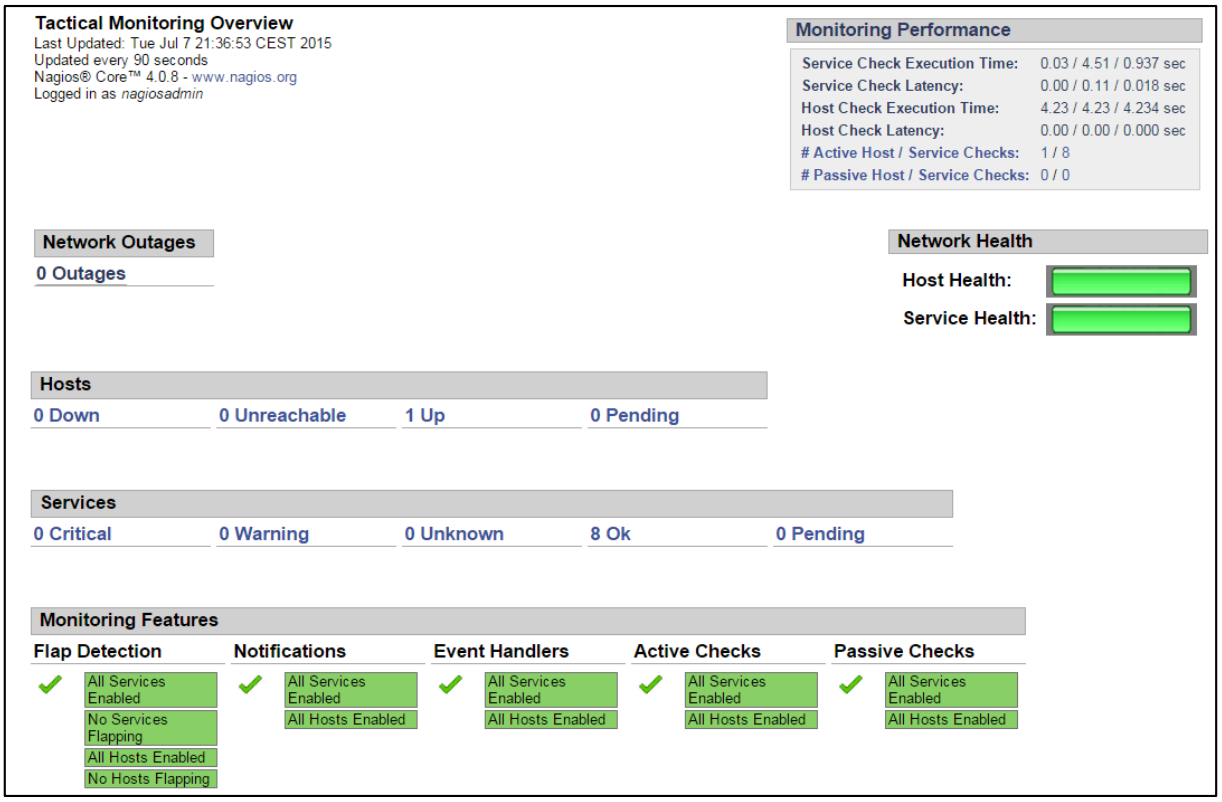


Figura 13 - Página Tactical Overview en la interfaz web de Nagios

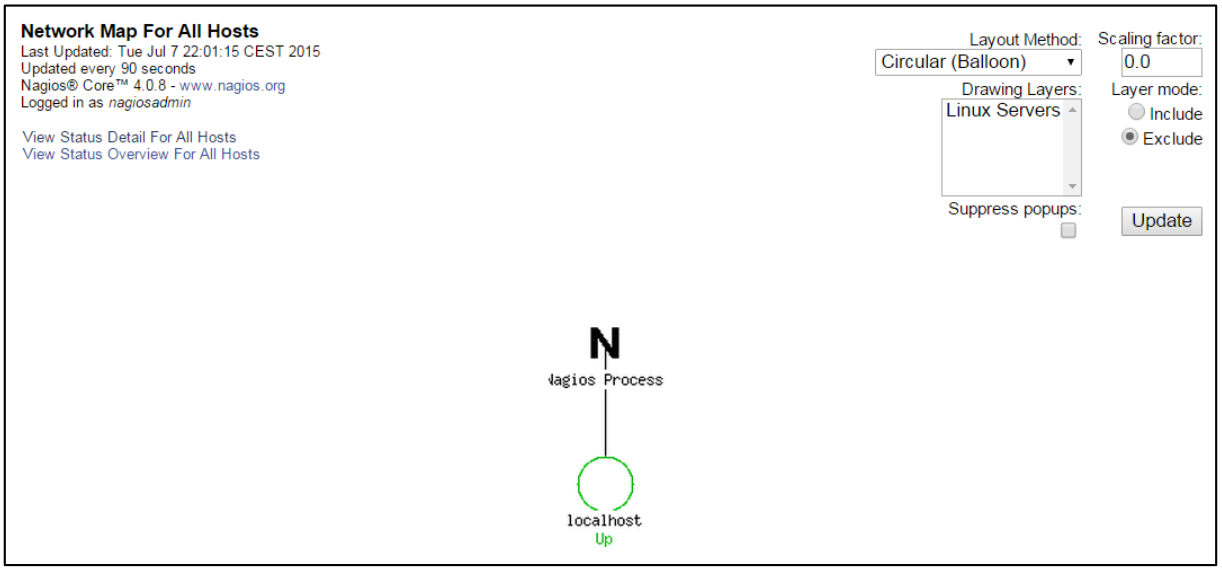


Figura 14 - Página Map de la interfaz web de Nagios

Se utilizará un módulo adicional, NagVis (véase *2.3.2*), para obtener una visualización más completa y personalizable de la representación de la topología de red.

Nagios contiene varias páginas que se pueden utilizar para ver y modificar la información de host. La interfaz web de Nagios ofrece una vista de todos los equipos definidos, sus estados, e información básica. Las vistas de los grupos de hosts también muestran los estados para los servicios asignados a los hosts. Las páginas de información de hosts también ofrecen la modificación de diversos parámetros relacionados con la configuración del host.

Nagios ofrece un panel que muestra todos los hosts junto con sus estados (véase *Figura 15*). Se puede acceder haciendo clic en el enlace *Hosts* del menú lateral de la parte izquierda (véase *Figura 12*).

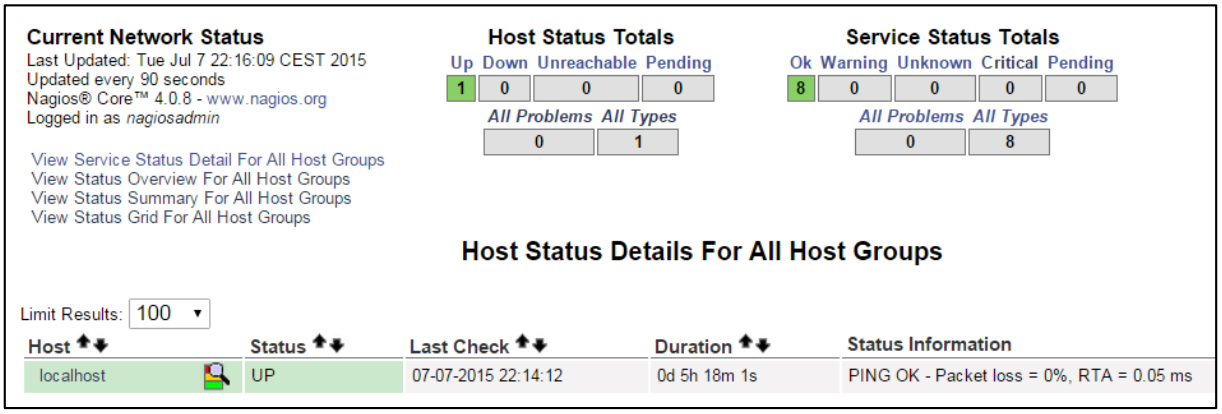


Figura 15 - Página Hosts de la interfaz web de Nagios

La página muestra una lista de todos los hosts, sus estados, y la información básica, como cuándo fue la última vez que se realizó la comprobación y la última vez que cambió su estado. También muestra respuesta del plugin de comprobación. El orden de cómo se muestra la tabla se puede cambiar con los botones de flecha junto al encabezado de cada columna.

Al igual que con la página *Tactical Overview*, los totales de la parte superior de la página se pueden utilizar para filtrar los hosts o los servicios para mostrar sólo aquellos con estados específicos. Después de hacer clic en cualquier tipo de estado en la tabla de totales de los hosts, la lista de los equipos se filtra para sólo mostrar los que actualmente tienen esa condición. Del mismo modo ocurre con tabla de totales de los servicios.

La interfaz web de Nagios también ofrece tres vistas que muestran los estados de todos los grupos de hosts. Una de estas vistas es la vista *Grid* (rejilla), que muestra grupos de hosts con los servicios de cada host y junto con el estado de los mismos como se puede apreciar en la *Figura 16*.

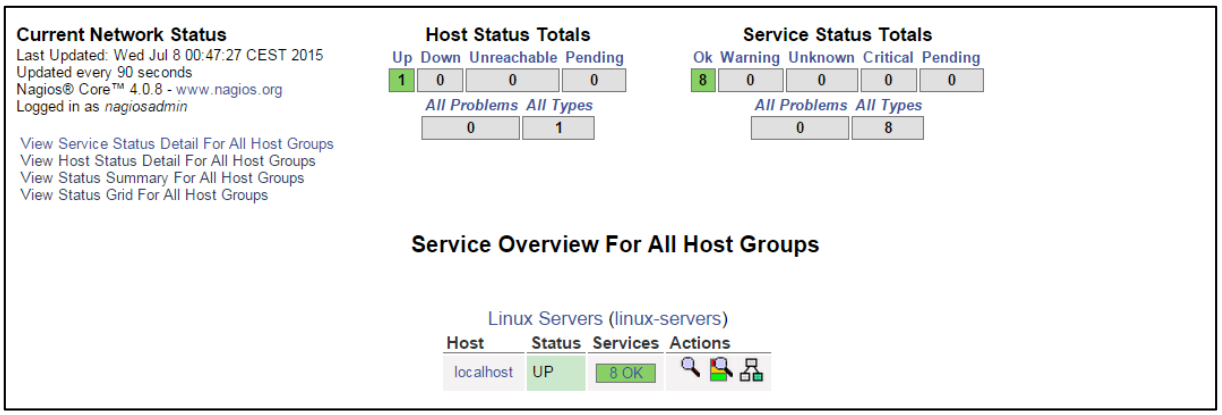


Figura 16 - Vista en rejilla de los grupos de hosts

Al hacer clic en un host en cualquier vista de la interfaz web, lleva a la página de información del equipo. En la *Figura 17* se muestra la página de información del host local.

Esta página contiene información detallada sobre el host seleccionado. Muestra las comprobaciones de estado y de acogida actuales que han sido o serán realizadas. También contiene información sobre qué funcionalidad está activada o desactivada para el host especificado, y si el host está cambiando de estado continuamente junto con los valores umbral.

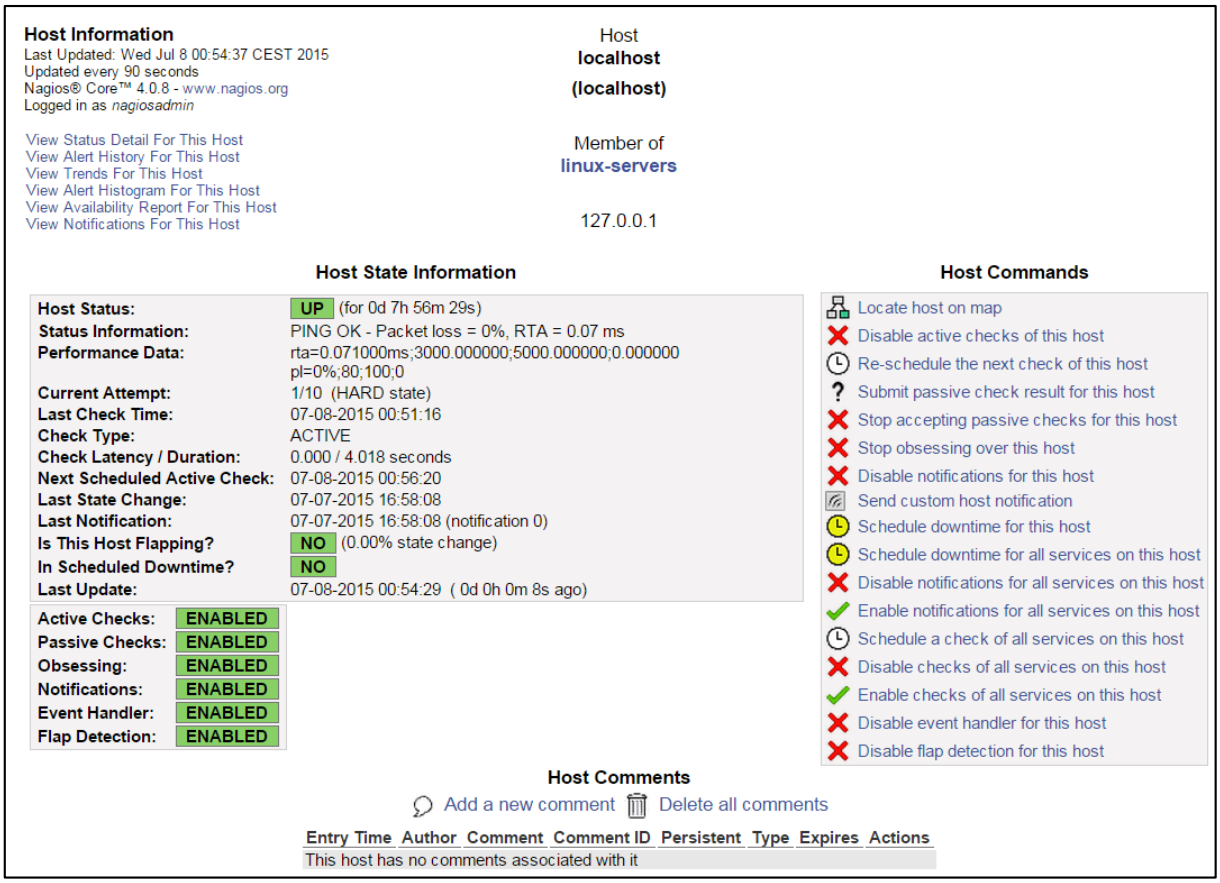


Figura 17 - Página de información del host local

El menú de la derecha, *Host Commands*, se puede utilizar para llevar a cabo operaciones relacionadas con este host. Permite configurar si deben habilitarse las comprobaciones activas, si Nagios debe aceptar resultados de una comprobación pasiva, y si se detecta un *flapping*. También puede configurar si Nagios debe "preocuparse" del host o enviar notificaciones y eventos. Además hay una opción para programar los controles para el host o todos los servicios con destino a este host. También puede enviar resultados de una comprobación pasiva a través de la interfaz web.

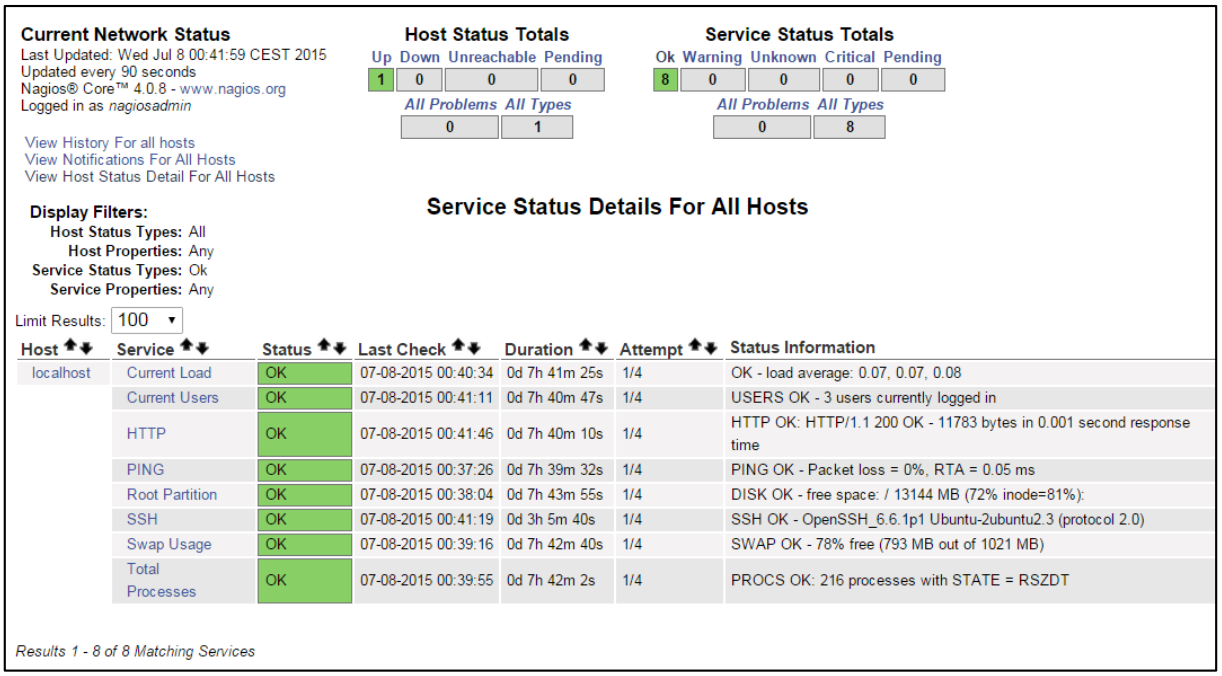
Debe tenerse en cuenta que en este menú, es el texto quien define la acción que se realizará al hacer clic. Los iconos del aspa roja y el "tic" verde pueden llevar a confusión sobre si la característica está habilitada o no.

La página de información también permite ver y modificar todos los comentarios relacionados con este host. Todos los comentarios actuales se muestran en la sección *Host Comments*.

La interfaz web de Nagios ofrece una vista de todos los servicios definidos, sus estados, e información básica, como muestra la *Figura 18*. Se puede acceder haciendo clic en *Services* en el menú lateral de la parte izquierda (véase *Figura 12*).

La parte principal de la página es la tabla que muestra todos los servicios, junto con sus estados y los detalles de la información devuelta por los plugins. Es posible ordenar la tabla según sus necesidades haciendo clic en las flechas de las cabeceras para cualquier columna de la tabla.

En la parte superior de la *Figura 18*, se encuentran los valores totales para cada estado de host y servicio. También puede filtrarse la tabla de servicios usando estos valores, sólo para mostrar estados de servicios específicos o para los hosts con un estado específico.



Current Network Status
Last Updated: Wed Jul 8 00:41:59 CEST 2015
Updated every 90 seconds
Nagios® Core™ 4.0.8 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	All Types
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	0	0	0

All Problems	All Types
0	8

Service Status Details For All Hosts

Display Filters:
Host Status Types: All
Host Properties: Any
Service Status Types: Ok
Service Properties: Any

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	07-08-2015 00:40:34	0d 7h 41m 25s	1/4	OK - load average: 0.07, 0.07, 0.08
	Current Users	OK	07-08-2015 00:41:11	0d 7h 40m 47s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	07-08-2015 00:41:46	0d 7h 40m 10s	1/4	HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.001 second response time
	PING	OK	07-08-2015 00:37:26	0d 7h 39m 32s	1/4	PING OK - Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	07-08-2015 00:38:04	0d 7h 43m 55s	1/4	DISK OK - free space: / 13144 MB (72% inode=81%):
	SSH	OK	07-08-2015 00:41:19	0d 3h 5m 40s	1/4	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.3 (protocol 2.0)
	Swap Usage	OK	07-08-2015 00:39:16	0d 7h 42m 40s	1/4	SWAP OK - 78% free (793 MB out of 1021 MB)
	Total Processes	OK	07-08-2015 00:39:55	0d 7h 42m 2s	1/4	PROCS OK: 216 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

Figura 18 - Página Services de la interfaz web de Nagios

La página también contiene un menú rápido que permite seleccionar una vista entre las más comunes. Permite ir al histórico y al registro de notificaciones, así como a una lista de todos los hosts con sus estados detallados.

Al hacer clic en cualquier host, le llevará a una página de información de host para el objeto seleccionado. Del mismo modo, al hacer clic en cualquier servicio se mostrará una página de información detallada para ese objeto.

Al hacer clic en un servicio en cualquier vista de la interfaz web le llevará a la página de información de servicio. Contiene detalles sobre el estado actual del servicio, una lista de los comentarios, y un panel de comandos que permite modificar la configuración del servicio, la programación de los controles, o el envío de notificaciones personalizadas. La *Figura 19* muestra el estado del servicio SSH.

La tabla principal de la izquierda, presenta información detallada sobre el servicio, como el estado actual, los datos de rendimiento, así como información detallada sobre la última y la próxima comprobación planificada. La página también muestra si el servicio está cambiando constantemente de estado junto con el umbral de cambio, y cuando se envió la última notificación.

El menú de la derecha, *Service Commands*, permite modificar si se debe realizar la comprobación, si las notificaciones han de enviarse, y si Nagios debe "obsesionarse" con este servicio. También hay una opción para programar cuándo se va a realizar la siguiente comprobación.

Al igual que el menú *Host Commands* de la *Figura 17*, debe tenerse en cuenta que es el texto quien define la acción que se realizará al hacer clic. Los iconos del aspa roja y el "tic" verde pueden llevar a confusión sobre si la característica está habilitada o no.

Además de todo esto existen secciones para planificar mantenimientos, ver el estado del proceso de Nagios, comprobar la información de rendimiento y generar registros de información.

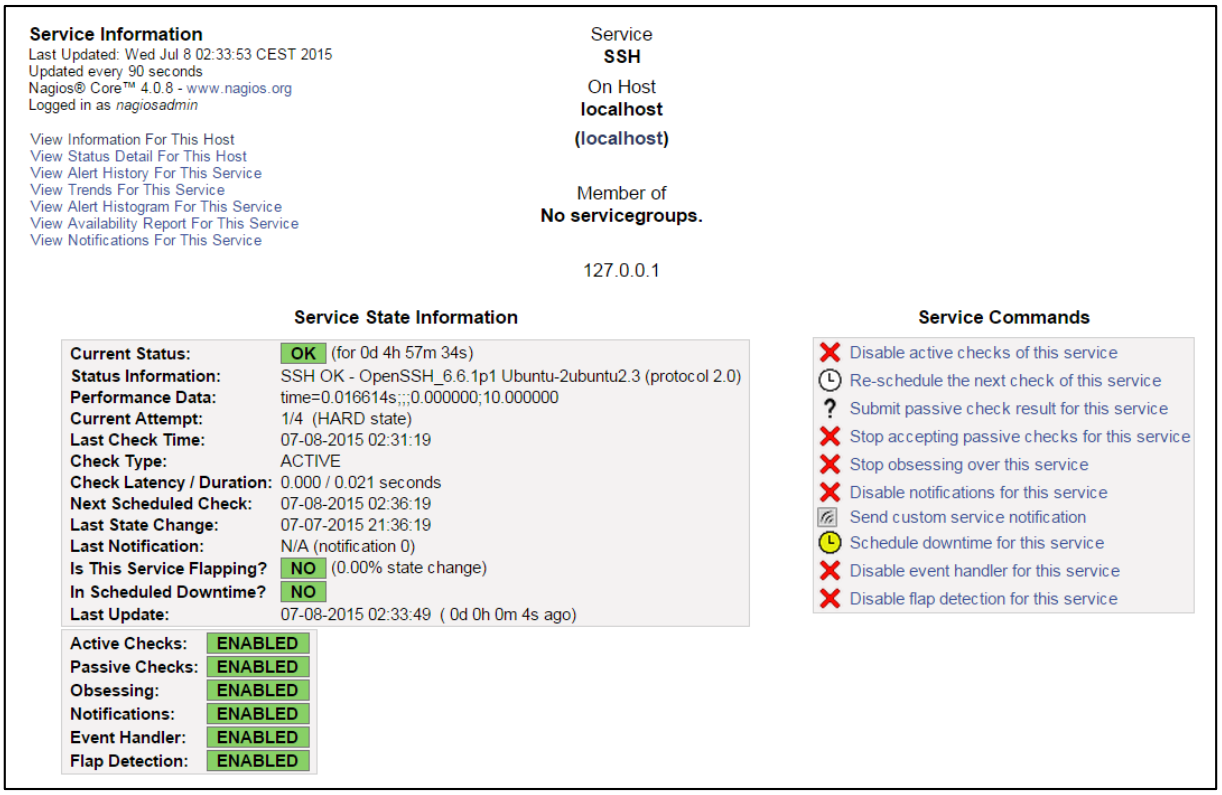


Figura 19 - Información del servicio SSH

3.2 Net-SNMP

Es recomendable la instalación de este paquete en Ubuntu desde el código fuente para habilitar el soporte de Perl para la recepción de traps. Es posible obtener el fichero que contiene el código fuente desde la propia web `http://www.net-snmp.org/download.html` o desde línea de comandos con la siguiente instrucción:

```
wget http://sourceforge.net/projects/net-snmp/files/net-snmp/5.7.2/net-snmp-5.7.2.tar.gz
```

Una vez descargado, se desempaqueta y se procede con la configuración, compilación e instalación. Es necesaria la librería de desarrollo de Perl para su funcionamiento:

```
apt-get install libperl-dev
tar -zxvf net-snmp-5.7.2.tar.gz
cd net-snmp-5.7.2
./configure --with-perl-modules
make
make install
```

Al finalizar la instalación del paquete Net-SNMP es necesario instalar los módulos de Perl. Éstos añaden más funcionalidades como, por ejemplo, el servicio de captura de traps. En el mismo directorio ejecutar las siguientes instrucciones:

```
cd perl
perl Makefile.PL
make
make test
make install
```

Después de una instalación exitosa, es posible ejecutar cualquier comando relacionado con SNMP, como `snmpget`, y comprobar la versión de Net-SNMP utilizando el siguiente comando:

```
root@ubuntu:/# snmpget -V
NET-SNMP version: 5.7.2
```

Suponiendo un host con el agente SNMP configurado es posible consultar su nombre utilizando el siguiente comando:

```
root@ubuntu:/# snmpget -v 1 -c public 192.168.110.1 sysName.0
sysName.0: Unknown Object Identifier (Sub-id not found: (top) -> sysName)
```

El error anterior se debe a que no encuentra una MIB en la que ese OID esté definido. Por ello, para obtener una mejor experiencia es recomendable descargar las MIBs estándar de IANA e IETF con el siguiente comando:

```
apt-get install snmp-mibs-downloader
```

Tras la instalación comenzará la descarga automáticamente y, al finalizar, ya será posible realizar peticiones de OIDs en formato textual.

Es posible que existan incompatibilidades si anteriormente se instaló Net:SNMP desde el repositorio.

```
/usr/sbin/snmpd: symbol lookup error: undefined symbol: smux_listen_sd
```

Este error se produce es por un problema de relocalización, se soluciona con borrar los ficheros adicionales que se instalan con el código fuente:

```
rm /usr/local/lib/libnetsnmp*
```

3.3 SNMPTT

SNMPTT permite el manejo de traps desde su recepción, pero son necesarios determinados paquetes para que la traducción de las traps sea posible. Para la instalación de los prerrequisitos es necesaria la instalación de los módulos Perl de Net-SNMP.

3.3.1 Prerrequisitos de instalación

A continuación se muestran las instrucciones necesarias para instalar los paquetes requeridos por SNMPTT [4]:

```
cpan YAML
cpan Getopt::Long
cpan File::Basename
cpan Time::HiRes
cpan Sys::Hostname
cpan Text::Balanced
cpan Text::ParseWords
cpan Sys::Syslog
cpan Crypt::DES
cpan Digest::MD5
cpan Digest::SHA1
cpan Digest::HMAC
cpan List::Util
cpan Config::IniFiles
cpan SNMP
cpan Net::SNMP
```

3.3.2 Obtener el código fuente de SNMPTT

La instalación de SNMPTT solo se puede realizar con los ficheros fuente. El código fuente se puede obtener a través de su web oficial `http://snmptt.sourceforge.net/downloads.shtml` o a través de consola con la siguiente instrucción:

```
wget
http://downloads.sourceforge.net/project/snmptt/snmptt/snmptt_1.4/snmptt_1.4.tgz
```

Una vez descargado, se desempaqueta y se procede con la configuración e instalación [10]:

```
tar -zxvf snmptt_1.4.tgz
cd snmptt_1.4
cp snmptthandler /usr/sbin
cp snmptt /usr/sbin
cp snmpttconvert /usr/sbin
cp snmpttconvertmib /usr/sbin
```

Añadir las siguientes líneas al fichero `/etc/snmp/snmptrapd.conf` con un editor de texto:

```
traphandle default /usr/sbin/snmptt
disableAuthorization yes
donotlogtraps yes
```

Cambiar los valores de `TRAPDRUN` y `TRAPDOPTS` en `/etc/default/snmpd` y `/etc/init.d/snmpd`:

```
export MIBS=ALL
TRAPDRUN=yes
TRAPDOPTS='-On -Lsd -p /var/run/snmptrapd.pid'
```

Copiar el fichero de configuración de ejemplo del código fuente de SNMPTT:

```
cp snmptt.ini /etc/snmp
```

Cambiar los siguientes valores en `/etc/snmp/snmptt.ini`:

```
net_snmp_perl_enable = 1
dns_enable=1
mibs_environment = ALL
```

Ahora se configuran los ficheros de log. Añadir al final de `/etc/logrotate.conf`:

```
/var/log/snmp/snmptt.log /var/log/snmp/snmpttunknown.log
{missingok}
```

Y crear el directorio y los ficheros donde se guardarán los logs de SNMPTT

```
mkdir /var/log/snmptt
touch /var/log/snmptt/snmptt.log
touch /var/log/snmptt/snmpttunknown.log
```

3.4 PNP4Nagios

3.4.1 Prerrequisitos de instalación

PNP4Nagios utiliza las librerías GD de PHP. Para instalarlas es necesario ejecutar los siguientes comandos:

```
apt-get install php5-gd librrds-perl rrdtool
```

3.4.2 Obtener el código fuente de PNP4Nagios

PNP4Nagios es un software gratuito que se distribuye bajo la licencia GPL en su versión 2. Para obtener el código fuente hay que dirigirse a la web oficial (véase *Figura 20*) donde se encuentran los enlaces a los ficheros, `https://docs.pnp4nagios.org/pnp-0.6/dwnld`.

Figura 20 - Sección de descargas en la web de PNP4Nagios

Al pinchar en el enlace la web redirige al usuario a la web donde se alojan los ficheros del proyecto, en *Sourceforce*, `http://sourceforge.net/projects/pnp4nagios/`.

Se ha utilizado la última versión de este software, se descargará el código fuente y se compilará con algunas modificaciones:

```
wget http://kent.dl.sourceforge.net/project/pnp4nagios/PNP-0.6/pnp4nagios-
     0.6.25.tar.gz
tar zxfv pnp4nagios-0.6.25.tar.gz
cd pnp4nagios-0.6.25
```

3.4.3 Compilar e instalar PNP4Nagios

Para compilar PNP4Nagios hay que pasarle al script de configuración unos parámetros adicionales. La ruta de instalación, la activación del sitio web en Apache y la dirección base a la que se accederá para poder visualizar los gráficos de PNP en el navegador web. Tras la configuración, sólo es necesario compilar el código fuente e instalarlo.

```
./configure --with-httpd-conf=/etc/apache2/sites-enabled --
     sysconfdir=/etc/pnp4nagios --with-base-url=/pnp4nagios
make all
make fullinstall
```

Al terminar se muestra un resumen de las opciones de instalación

```
*** Configuration summary for pnp4nagios-0.6.25 03-01-2015 ***
  General Options:
  -------------------------        -------------------
  Nagios user/group:               nagios nagcmd
  Install directory:               /usr/local/pnp4nagios
  HTML Dir:                        /usr/local/pnp4nagios/share
  Config Dir:                      /etc/pnp4nagios
  Location of rrdtool binary:      /usr/bin/rrdtool Version 1.4.7
```

```
RRDs Perl Modules:                FOUND (Version 1.4007)
RRD Files stored in:              /usr/local/pnp4nagios/var/perfdata
process_perfdata.pl Logfile:      /usr/local/pnp4nagios/var/perfdata.log
Perfdata files (NPCD) stored in:  /usr/local/pnp4nagios/var/spool
Web Interface Options:
------------------------          -------------------
HTML URL:                      http://localhost/pnp4nagios
Apache Config File:            /etc/apache2/sites-enabled/pnp4nagios.conf
```

Si la instalación se realizó correctamente mostrará el siguiente mensaje:

```
*** Main program, Scripts and HTML files installed ***
```

3.4.4 Interfaz web

Tras la instalación es posible comprobar el funcionamiento de PNP4Nagios. Es necesario reiniciar apache para que establezca la nueva configuración con la siguiente instrucción

```
service apache2 restart
```

Ahora ya será accesible la URL de PNP4Nagios. Se utilizarán las mismas credenciales que para acceder a la URL de Nagios, `http://localhost/pnp4nagios`.

La primera vez que se acceda esta URL se comprobará que la herramienta ha realizado unos test para comprobar que el entorno es correcto y que todos los requisitos están disponibles. La página de comprobación se muestra en la *Figura 21*. También avisa de que se renombre la página de comprobación para poder acceder al entorno de PNP4Nagios.

```
mv /usr/local/pnp4nagios/share/install.php
     /usr/local/pnp4nagios/share/install.php.backup
```

Si se vuelve a cargar la página y se obtiene el siguiente error:

```
Please check the documentation for information about the following error.
perfdata directory "/usr/local/pnp4nagios/var/perfdata/C" is empty. Please
check your Nagios config. Read FAQ online
```

Este mensaje es normal mientras no se realice la integración con Nagios, debido a que no puede procesar los datos que los plugins de Nagios recogen.

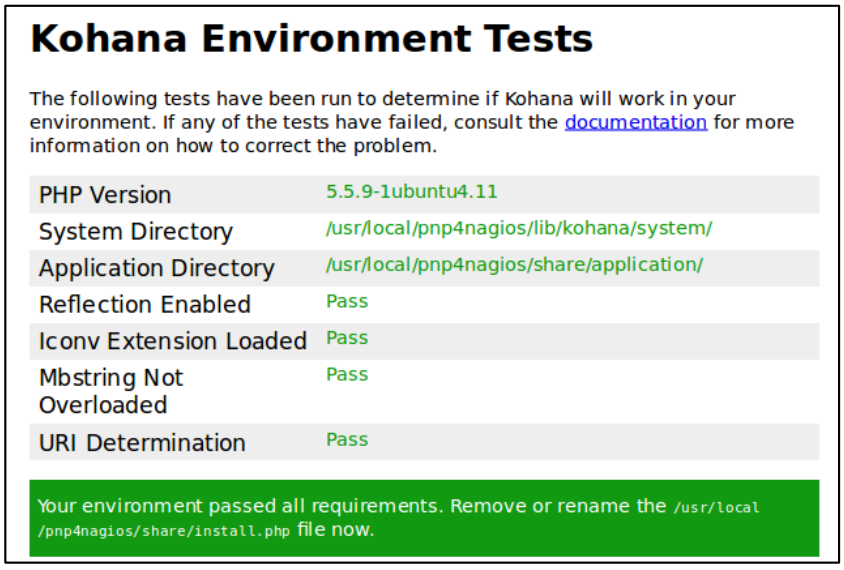


Figura 21 - Página de comprobación de PNP4Nagios

3.4.5 Integración con Nagios

Ahora hay que realizar la configuración de PNP4Nagios. Éste permite varias opciones para configurar la integración con Nagios Core. Se ha escogido la opción denominada "Modo Masivo con NPCD" (*Bulk Mode with NPCD*). Visto desde el punto de vista de Nagios, es el mejor método de procesado, puesto que éste nunca llegaría a bloquearse. Nagios utiliza un fichero temporal para almacenar los datos y ejecuta un comando después de un cierto tiempo predefinido. En lugar de procesar inmediatamente los datos mediante `process_perfdata.pl`, el fichero se mueve a un directorio de `spool` (sondeo). El mover un archivo en el sistema de ficheros del operativo, prácticamente no conlleva tiempo, por lo que Nagios es capaz de ejecutar su trabajo de forma inmediata.

NPCD (*Nagios Performance C Daemon*) es un servicio que monitoriza el directorio `spool` en busca de nuevos ficheros y le pasa los nombres a `process_perfdata.pl`. Implica que el procesado de los datos está desacoplado de Nagios. Además, NPCD es capaz de lanzar múltiples hilos de ejecución para el procesado de datos. [6] La *Figura 22* muestra un esquema del funcionamiento de este método.

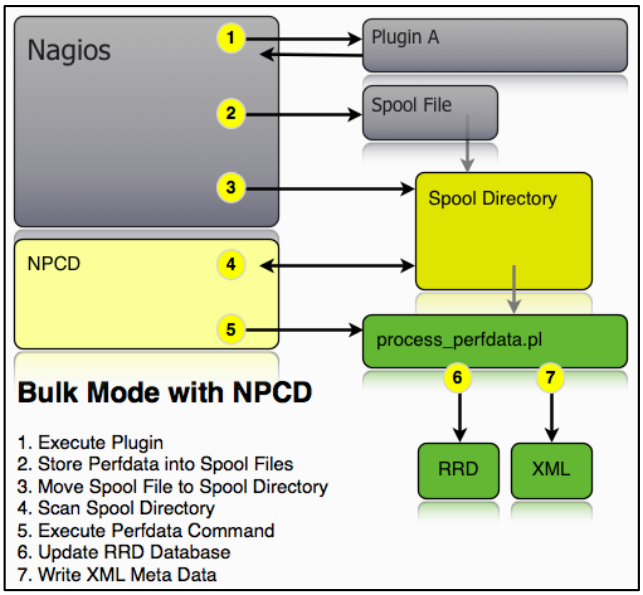


Figura 22 - Descripción del Modo Masivo con NPCD de PNP4Nagios [6]

Para configurar este método, hay que copiar parte del código del fichero de ejemplo que PNP4Nagios instalala en el equipo, `/etc/pnp4nagios/nagios.cfg-sample`, al fichero de configuración de Nagios `/usr/local/nagios/etc/nagios.cfg`.

```
# Bulk / NPCD mode
#
process_performance_data=1
#
service_perfdata_file=/usr/local/pnp4nagios/var/service-perfdata
service_perfdata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::$TIMET$\t
HOSTNAME::$HOSTNAME$\tSERVICEDESC::$SERVICEDESC$\tSERVICEPERFDATA::$SERVIC
EPERFDATA$\tSERVICECHECKCOMMAND::$SERVICECHECKCOMMAND$\tHOSTSTATE::$HOSTST
ATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$\tSERVICESTATE::$SERVICESTATE$\tSERVIC
ESTATETYPE::$SERVICESTATETYPE$
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=15
service_perfdata_file_processing_command=process-service-perfdata-file
#
# *** the template definition differs from the one in the original
#
```

```
host_perfdata_file=/usr/local/pnp4nagios/var/host-perfdata
host_perfdata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNA
ME::$HOSTNAME$\tHOSTPERFDATA::$HOSTPERFDATA$\tHOSTCHECKCOMMAND::$HOSTCHECK
COMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$
host_perfdata_file_mode=a
host_perfdata_file_processing_interval=15
host_perfdata_file_processing_command=process-host-perfdata-file
```

A continuación, es necesario copiar el siguiente fragmento de código del fichero de ejemplo, `/etc/pnp4nagios/misccommands.cfg-sample`, al final del fichero de definición de comandos, `/usr/local/nagios/etc/objects/commands.cfg`:

```
# Bulk with NPCD mode
#
define command {
       command_name      process-service-perfdata-file
       command_line      /bin/mv /usr/local/pnp4nagios/var/service-perfdata
}
define command {
       command_name      process-host-perfdata-file
       command_line      /bin/mv /usr/local/pnp4nagios/var/host-perfdata
}
```

Como funcionalidad adicional es posible integrar las gráficas de PNP4Nagios en el núcleo de Nagios. De esta forma se podrá acceder directamente desde el host o servicio en Nagios a las gráficas correspondientes de éste sin tener que ir al interfaz de PNP4Nagios para localizarlo.

Siguiendo las instrucciones de la documentación, `https://docs.pnp4nagios.org/pnp-0.6/webfe`, se configurará este servicio para que muestre "popups" (ventanas emergentes) al pasar el ratón por encima. Es necesario crear plantillas para aplicarlas luego a los objetos.

Al final del fichero `/usr/local/nagios/etc/objects/templates.cfg` se añade:

```
define host{
       name host-pnp
       action_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=_HOST_'
       class='tips'
       register 0
}
define service{
       name srv-pnp
       action_url
       register 0
}
```

Se modifican todos los objetos host y servicios para que hereden dicha plantilla. Por ejemplo:

```
define host{
       use linux-server,host-pnp
       host_name localhost
       alias localhost
       address 127.0.0.1
}
define service{
       use local-service,srv-pnp
       host_name localhost
       service_description PING
       check_command check_ping!100.0,20%!500.0,60%
}
```

Por último, es necesario copiar un fichero a la configuración web de Nagios y las rutas de la documentación de PNP4Nagios ya que no coinciden con las que se han instalado. Se copia el fichero `status-header.ssi` que se encuentra en la carpeta del código fuente, anteriormente descargado, de PNP4Nagios:

```
cp contrib/ssi/status-header.ssi /usr/local/nagios/share/ssi/
```

Si la integración se ha realizado correctamente aparecerán los iconos para acceder pinchando directamente a las gráficas del host o servicio y además, si se pasa el ratón por encima del icono muestra un *popup* con la gráfica.

Se configura el inicio automático del demonio y se inicia el servicio a la vez que se reinicia el servicio de Nagios.

```
ln -s /etc/init.d/npcd /etc/rcS.d/S99npcd
service npcd start && service nagios restart
```

Por último es posible recargar la URL de Nagios y comprobar cómo las gráficas de PNP4Nagios se integran con el Core de Nagios mediante *popups* como muestra la *Figura 23*.

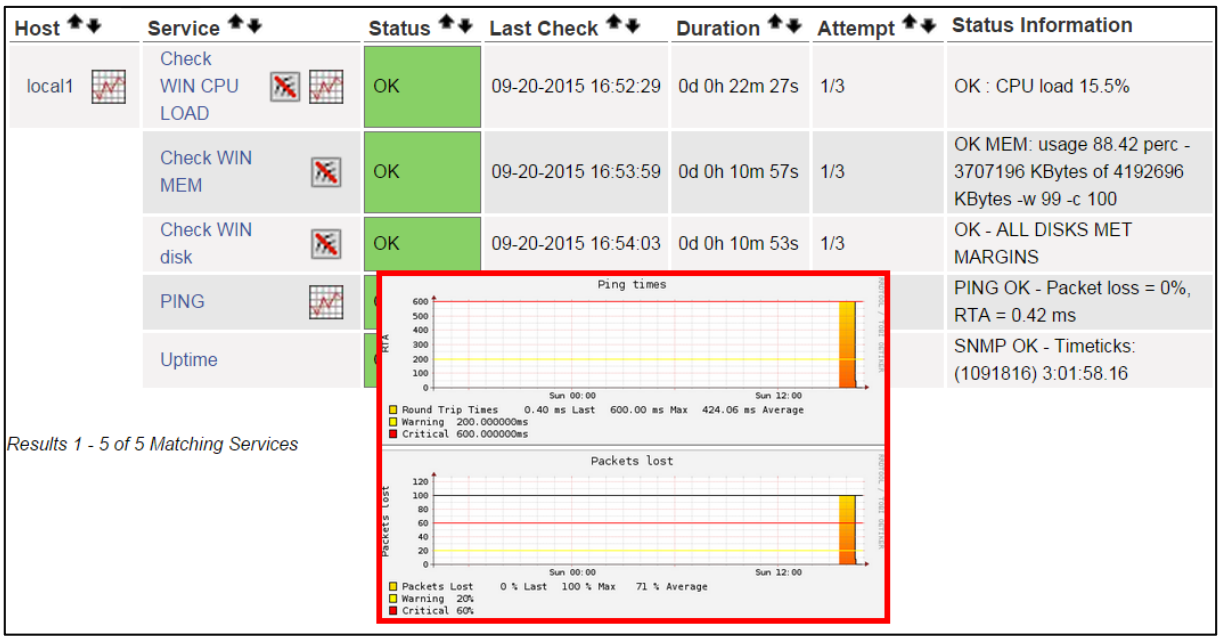


Figura 23 - Popups de PNP4Nagios integradas en Nagios Core

Puede ocurrir que tarde un tiempo en actualizarse la información de Nagios durante el cual PNP4Nagios no estará disponible, dependerá del tiempo de refresco de Nagios.

En caso de acceder a la URL de PNP4Nagios, `http://localhost/pnp4nagios`, será posible visualizar todas las gráficas de un determinado host para el cuál se han recogido y representado los valores de rendimiento obtenidos por los plugins de Nagios. En la *Figura 24* se muestran algunas de las gráficas que representan los servicios de host local con diferentes intervalos de tiempo.

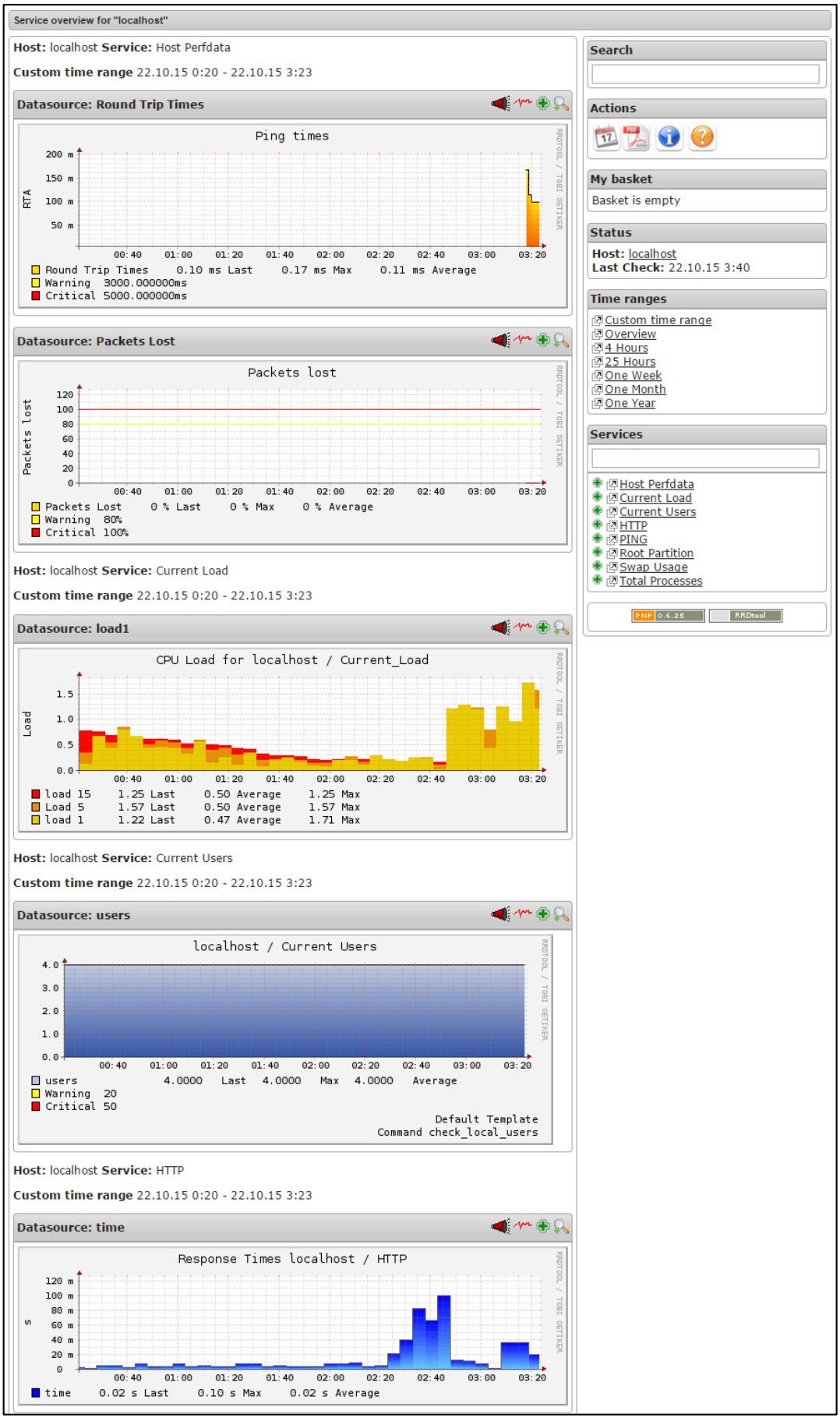


Figura 24 - Gráficas de PNP4Nagios

3.5 NagVis

NagVis es un plugin para Nagios que permite visualizar gráficamente todos sus objetos de

La instalación de NagVis se realizará a través del código fuente. Es necesario compilarlo con los parámetros adecuados y proceder con la instalación. La página oficial de NagVis, `http://www.nagvis.org/`, muestra la sección de descargas, así como la documentación de ayuda para poder instalar el software.

3.5.1 Prerrequisitos de instalación

Para poder realizar una compilación del código con éxito es necesario tener instalados los siguientes paquetes:

```
apt-get install gawk g++ make libc6-dev
```

3.5.2 Obtener el código fuente de Livestatus

Recordar que Livestatus es un *backend*, necesario para que NagVis pueda comunicarse con Nagios (véase *2.3.2*). Éste es un software gratuito que se distribuye bajo la licencia GPL en su versión 2.

Para obtener el código fuente hay que dirigirse a la web oficial del desarrollador de Livestatus, `http://mathias-kettner.com/check_mk_download_source.html` (*Figura 25*).

Figura 25 - Sección de descargas en la web de Livestatus

Se ha utilizado la última versión estable de este software, se descargará el código fuente y se compilará con algunas modificaciones:

```
wget http://mathias-kettner.com/download/mk-livestatus-1.2.6p10.tar.gz
tar zxfv mk-livestatus-1.2.6p10.tar.gz
cd mk-livestatus-1.2.6p10
```

3.5.3 Compilar e instalar Livestatus

Tras desempaquetar el "tarball" (fichero con extensión `.tar`), y situar el prompt en la nueva carpeta que contiene el código fuente se deben ejecutar las siguientes instrucciones para configurar Livestatus para la versión 4 de Nagios y posteriormente compilar e instalar.

```
./configure --with-nagios4
make
make install
```

Si no se mostró ningún error en la compilación, se puede asumir que la instalación Livestatus se ha completado correctamente, aunque más tarde se realizarán las comprobaciones necesarias para cerciorarse. Para completar la integración con Nagios es necesario añadir un fragmento de código en el fichero de configuración de Nagios, `nagios.cfg`. Según los parámetros establecidos en la instalación de Nagios, la ruta en la que se encuentra este fichero es `/usr/local/nagios/etc/`.

Con un editor de texto se añaden las siguientes líneas:

```
broker_module=/usr/local/lib/mk-livestatus/livestatus.o
/usr/local/nagios/var/rw/live
event_broker_options=-1
```

Tras la instalación de Livestatus, es necesario reiniciar Nagios

```
service nagios restart
```

Y en el log de Nagios, `/usr/local/nagios/var/nagios.log`, comprobar que Livestatus se ha inicializado correctamente:

```
[1441071601] livestatus: Livestatus 1.2.6p10 by Mathias Kettner. Socket:
'/usr/local/nagios/var/rw/live'
[...]
[1441071602] livestatus: Finished initialization. Further log messages go
to /usr/local/nagios/var/livestatus.log
[1441071602] Event broker module '/usr/local/lib/mk-
livestatus/livestatus.o' initialized successfully.
```

3.5.4 Obtener el código fuente de NagVis

NagVis es un software gratuito que se distribuye bajo la licencia GPL en su versión 2. Para obtener el código fuente hay que dirigirse a la web oficial de NagVis donde se alojan los ficheros del código fuente en la web del desarrollador, `http://www.nagvis.org/downloads`.

Es posible descargarlo directamente de la web (*Figura 26*) o a través de la línea de comandos:

```
wget http://www.nagvis.org/share/nagvis-1.8.5.tar.gz
tar zxfv nagvis-1.8.5.tar.gz
cd nagvis-1.8.5
```

Se ha utilizado la última versión estable de este software, se descargará el código fuente y se compilará con algunas modificaciones:

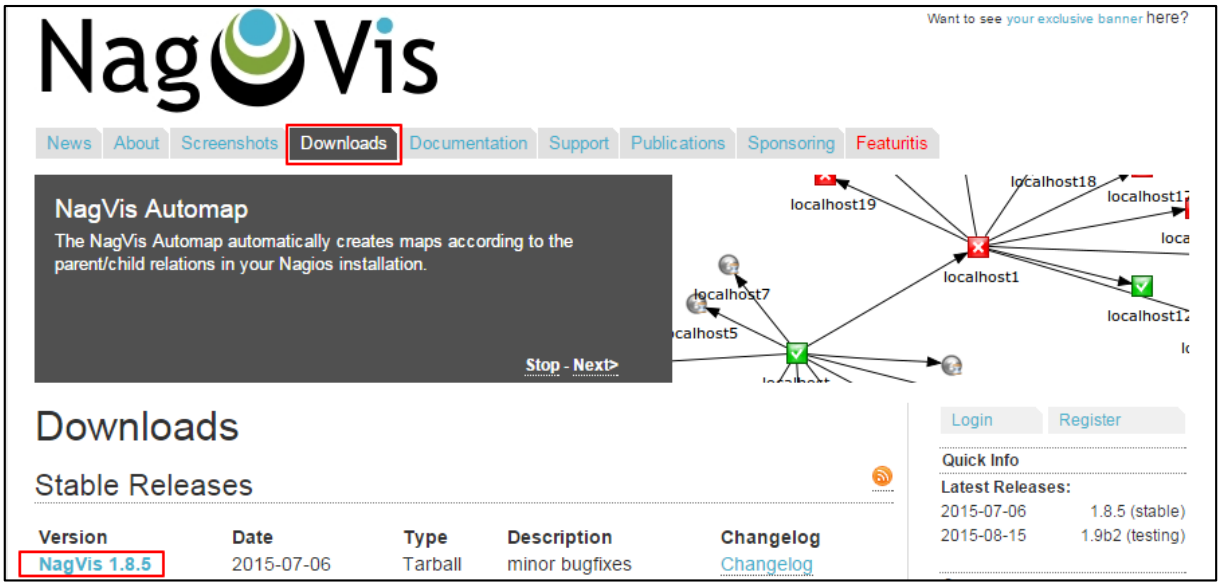


Figura 26 - Sección de descargas en la web de NagVis

3.5.5 Compilar e instalar NagVis

Para que NagVis funcione correctamente son necesarios algunos módulos adicionales de PHP, como los gráficos, bases de datos y Graphviz. Éste último es necesario para poder utilizar la función "Automap" en NagVis. [8]

```
apt-get install php5-gd php5-json php-net-socket graphviz rsync php5-
common libapache2-mod-php5 php5-cli php5-cgi sqlite sqlite3 php5-sqlite
libjson-xs-perl php-pear php-apc php5-curl
```

Tras instalar los módulos necesarios se procederá a instalar NagVis. Con la ayuda de un script que se encuentra en la carpeta desempaquetada del dichero .tar, `install.sh`. Solamente es necesario pasarle dos argumentos y responder a todas las preguntas que haga afirmativamente. Así se establecerán los valores por defecto en NagVis. En caso de que las rutas de instalación no coincidan o se hayan cambiado algunas opciones, es necesario establecerlas adecuadamente.

```
root@ubuntu:~/Desktop/nagvis-1.8.5# ./install.sh -n /usr/local/nagios -p
/usr/local/nagvis -l "unix:/usr/local/nagios/var/rw/live" -b mklivestatus -u www-
data -g nagcmd -w /etc/apache2/conf-available -a y
+------------------------------------------------------------------------------+
| Welcome to NagVis Installer 1.8.5                                            |
+------------------------------------------------------------------------------+
[...]
+------------------------------------------------------------------------------+
| Do you want to proceed? [y]:
+------------------------------------------------------------------------------+
[...]
+------------------------------------------------------------------------------+
| Installation complete                                                        |
|                                                                              |
| You can safely remove this source directory.                                 |
|                                                                              |
| For later update/upgrade you may use this command to have a faster update:   |
| ./install.sh -n /usr/local/nagios -p /usr/local/nagvis -l
|                                                                              |
| What to do next?                                                             |
| - Read the documentation                                                     |
| - Maybe you want to edit the main configuration file?                        |
|   Its location is: /usr/local/nagvis/etc/nagvis.ini.php                      |
| - Configure NagVis via browser                                               |
|   <http://localhost/nagvis/config.php>                                       |
| - Initial admin credentials:                                                 |
|     Username: admin                                                          |
|     Password: admin                                                          |
+------------------------------------------------------------------------------+
```

3.5.6 Interfaz web

Al final del proceso de instalación de NagVis se informa al usuario de cuáles son las credenciales de administrador. Para simplificar y utilizar las mismas que en Nagios es necesario añadir unas líneas de código en el fichero de configuración de NagVis.

Este fichero se encuentra en la ruta `/usr/local/nagvis/etc/nagvis.ini.php`, como se muestra en el resumen de la instalación. Las siguientes líneas deben descomentarse (quitar el símbolo " ; ") en la sección `[global]` del fichero de configuración:

```
logonmodule="LogonMixed"
logonenvvar="REMOTE_USER"
logonenvcreateuser="1"
```

También debe editarse el fichero de configuración creado en el directorio de Apache, `/etc/apache2/conf-available/nagvis.conf`, quitándo el símbolo # al inicio de las líneas, de tal forma que quede como el texto a continuación:

```
AuthName "NagVis Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
```

Tras la instalación es necesario habilitar el sitio web en Apache. Para ello se debe crear un enlace simbólico ("symlink") del fichero en la carpeta de Apache `sites-enabled`.

```
ln -s /etc/apache2/conf-available/nagvis.conf /etc/apache2/sites-
enabled/nagvis.conf
```

El último paso es ejecutar un script que establecerá al usuario `nagiosadmin` como administrador. El fichero se encuentra en la carpeta desempaquetada con el código fuente de NagVis. Para establecer los permisos del usuario ejecutar las siguientes instrucciones:

```
cp nagvis-make-admin /usr/local/nagvis/etc/
cd /usr/local/nagvis/etc/
./nagvis-make-admin nagiosadmin
service apache2 restart && service nagios restart && service npcd restart
```

Tras reiniciar Apache, Nagios y PNP, la interfaz web de NagVis ya será accesible. Con un navegador web dirigirse a `http://localhost/nagvis` y se podrá visualizar la pantalla principal como se muestra en la *Figura 27*.

Se recomienda borrar el usuario por defecto `admin` desde la pestaña *Users Menu -> Manage Users*.

En la interfaz se muestran mapas de ejemplo con los que se puede apreciar el potencial de NagVis. Al hacer clic en cada uno de ellos se abrirá el mapa muestra.

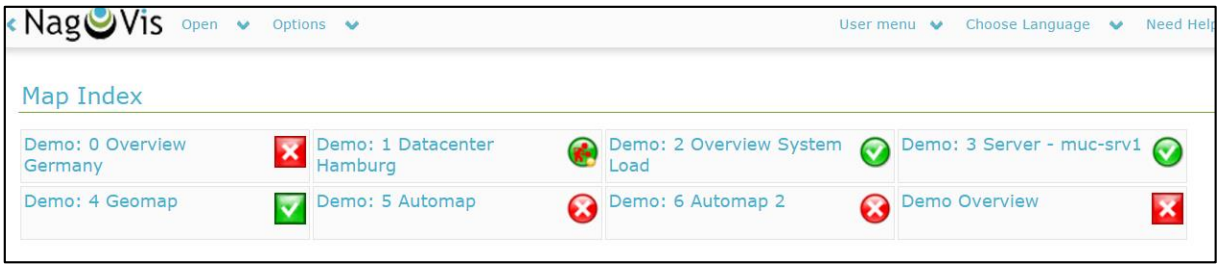


Figura 27 - Interfaz web de NagVis

4. IMPLEMENTACIÓN EN EL LABORATORIO

El objetivo de este trabajo es establecer un sistema de monitorización de la red y los sistemas que pertenecen al Laboratorio de Telemática y al Laboratorio de Aplicaciones Telemáticas de la Universidad de Cantabria.

En este apartado se procederá a la configuración de todo el entorno de trabajo para la monitorización de los equipos. La *Figura 28* muestra un esquema de la red de ambos laboratorios. Conforme a esa figura se han de configurar los ficheros en los que se definen los hosts y los servicios que utilizarán.

4.1 Descripción de la topología de los laboratorios

Ambos laboratorios están interconectados a través de un router Cisco 2600 que, a partir de ahora, será considerado en nodo "padre" del cual todos dependen. El Cisco 2600 dispone de un agente SNMP propietario de Cisco, que puede utilizarse para obtener, por ejemplo, el tráfico de red que pasa por él.

Este router está conectado a un switch SMC de 24 puertos al que están conectados los equipos denominados "locales". Son diez equipos que se definirán en Nagios desde el `local1` al `local10`. Éstos son los PCs más nuevos, tienen sistema operativo Windows 7 x64 y son los equipos donde se instalará el subagente SNMP. Poseen direcciones privadas de la subred 192.168.110.0/24.

Los equipos locales de definirán en Nagios utilizando la directiva `parents` como `cisco2600` y con el `hostgroup` definido como `hosts`.

El Cisco 2600 está conectado además a una red X.25 con una rango de direcciones 192.168.200.0/24 que, a efectos de monitorización será completamente transparente.

Se podrá asumir entonces que el router Cisco 2500, conectado a la red X.25, será "hijo" directo del router Cisco 2600. El router Cisco 2500 está conectado a una red de diez ordenadores denominados "remotos". Éstos están conectados mediante clave coaxial y pertenecen a la subred 192.168.100.0/24 y en Nagios se definirán desde `remoto1` a `remoto10`. Son equipos antiguos que no cuentan con un agente SNMP instalado.

Los equipos locales de definirán en Nagios utilizando la directiva `parents` como `cisco2500` y con el el mismo grupo de hosts que los locales, `hostgroup hosts`.

La salida a Internet de este laboratorio se realiza a través de un servidor llamado Atlas. Este servidor está conectado al Cisco 2600 y realiza funciones de firewall y NAT (*Network Address Translation*) entre la red privada e Internet. El servidor con nombre de dominio `leda.tlmat.unican.es` está conectado directamente a Internet con una dirección IP pública. Ambos servidores se definirán en Nagios en el grupo `servers`

En el laboratorio de Aplicaciones Telemáticas existen seis nodos ATM. Estos nodos administran los usuarios de las diferentes secciones de la red ATM de la Universidad de Cantabria. Los usuarios son agrupados en redes locales virtuales (VLANs) según su localización.

El nodo ATM del Servicio de Informática es hijo directo del Cisco 2600 y están conectados con el direccionamiento 192.168.0.20/30. Este nodo está directamente conectado con los otros cinco nodos ATM. Cada enlace posee su propio direccionamiento y cada nodo sus propias VLANs.

Los seis nodos ATM pertenecerán en Nagios al grupo `ATMnodes`

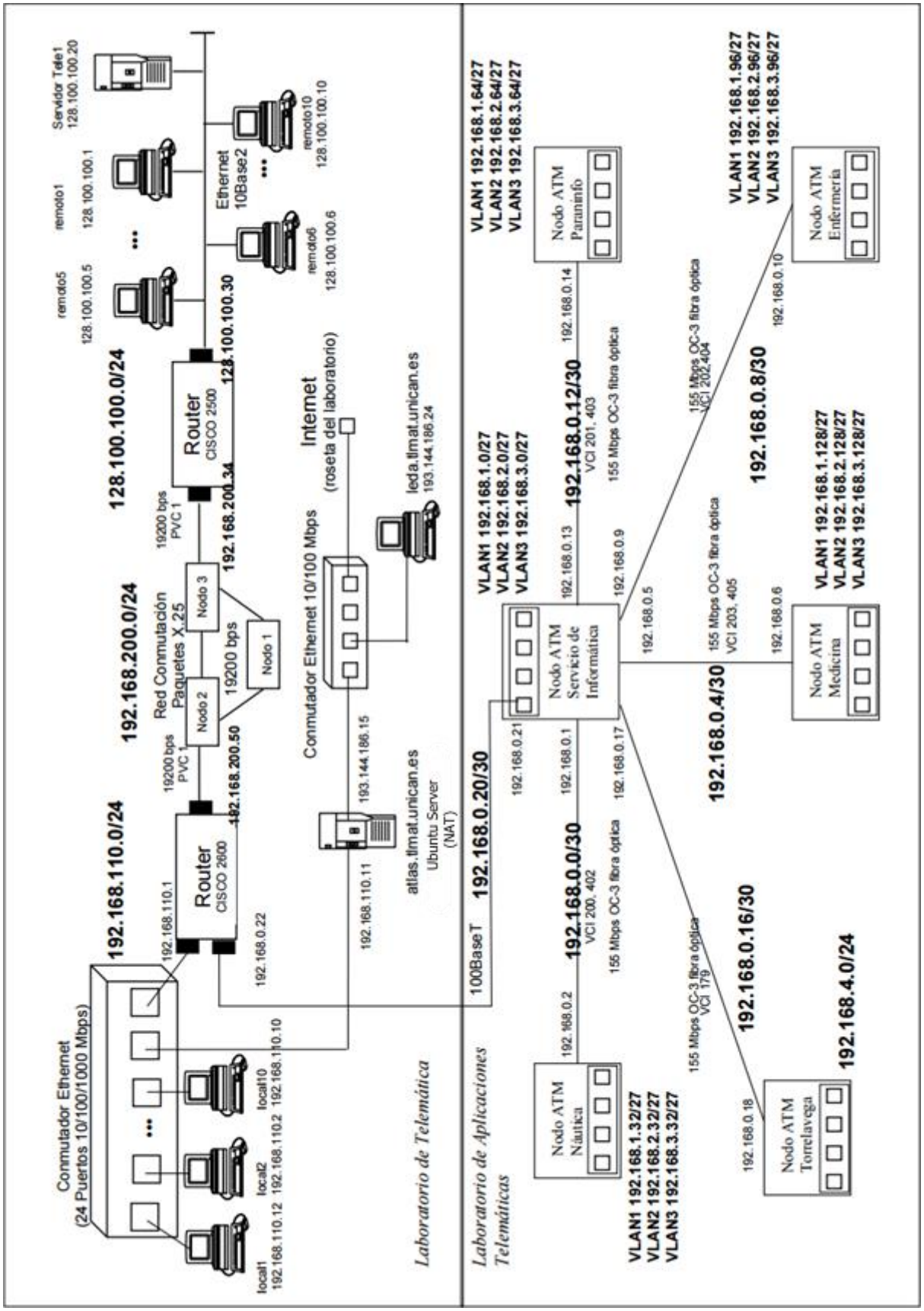


Figura 28 - Esquema de red de los laboratorios

4.2 Definición de objetos en Nagios

Tras haber estudiado el funcionamiento de la definición de objetos en Nagios se procederá a continuación, a establecer los ficheros donde los equipos serán definidos.

El conjunto de los equipos a monitorizar en los laboratorios, se ha dividido en cuatro tipos. Esta división atiende al tipo de máquina y/o servicio que ofrece, PCs, servidores, routers y nodos ATM.

Se creará una carpeta en la ruta `/usr/local/nagios/etc`, que contendrá todos los ficheros que definirán las objetos del laboratorio.

```
mkdir /usr/local/nagios/etc/lab
```

Cada fichero de configuración, con la extensión `.cfg`, contendrá la información necesaria para que Nagios sea capaz de monitorizar cada objeto. Para crear un fichero de texto se pueden utilizar cualquier editor de texto plano en Ubuntu (`gedit`, `nano`, `vi`, etc…). Con la siguiente instrucción se creará el fichero `hosts.cfg` en la ruta adecuada:

```
gedit /usr/local/nagios/etc/lab/hosts.cfg
```

A continuación se muestra un fragmento del fichero `hosts.cfg` que corresponde a la definición del grupo de hosts y a la de un PC con sistema operativo Windows.

```
define hostgroup{
       hostgroup_name          hosts
       alias                   PCs
}
define host{
       use                     windows-server,host-pnp
       host_name               local1
       alias                   PC local1
       address                 192.168.110.12
       parents                 Cisco 2600
       hostgroups              hosts
}
```

En este fichero deben definirse todos los equipos que pertenecientes al `hostgroup hosts`, como se detalla en el apartado *4.1*. Siguiendo la misma metodología deberán definirse el resto de ficheros. Todos los ficheros de definición de objetos se encuentran en el *Anexo I*.

Los servicios también pueden definirse en un fichero a parte o en varios, el único requisito es añadir la ruta de cada fichero de definición de objetos al fichero de configuración principal de Nagios.

En este caso el fichero que contiene los servicios se ha establecido en la misma ruta del fichero `hosts.cfg` con el nombre `services.cfg`.

Para la monitorización de los PCs del laboratorio se hará uso de los plugins de Nagios. Uno de los servicios propuesto es `check_ping` que realiza un petición `ECHO` al equipo y espera su respuesta. Al comando se le pasan parámetros adicionales que definirán, en función de la respuesta, el estado del servicio en Nagios.

Los plugins de Nagios muestran información de funcionamiento si se ejecutan con el parámetro `-h`:

```
root@ubuntu:/usr/local/nagios/libexec# ./check_ping -h
check_ping v2.0.3 (nagios-plugins 2.0.3)
[...]
Usage:
check_ping -H <host_address> -w <wrta>,<wpl>% -c <crta>,<cpl>%
 [-p packets] [-t timeout] [-4|-6]
```

Este plugin proporciona además la latencia del ping como información de rendimiento. Y gracias a PNP4Nagios es posible comprobar estas latencias representadas gráficamente y con históricos. Solo es necesario añadir la plantilla de servicio (`srv-pnp`) para que empiece a recoger y procesar la información de rendimiento devuelta por el plugin.

Otro de los plugins utilizado es `check_snmp` que proporciona soporte para realizar cualquier petición SNMP para obtener información de equipos remotos. Utilizando este plugin de Nagios se realiza una petición SNMP para obtener el tiempo que el equipo lleva encendido, desde el último reinicio, con el OID `sysUpTime.0` definido en la MIB RFC1213.

Gracias a la flexibilidad de Nagios y a una comunidad de usuarios activa, se han incluido varios plugins que realizan diversas comprobaciones mediante el protocolo SNMP. Estos scripts están escritos en Perl y se han obtenido de Nagios Exchange (https://exchange.nagios.org). Cada plugin tiene su propia licencia establecida por su autor. Todos los plugins utilizados para realizar la monitorización en el laboratorio son gratuitos y con licencia de libre de uso, distribución y modificación.

El script `check_win_snmp_disk`, alojado en `/usr/local/nagios/libexec/`, comprobará el espacio total de cada disco duro o partición reconocida por Windows, así como su espacio utilizado y libre.

Uno de los requisitos para utilizar estos scritps, es su definición en el fichero de comandos de Nagios. Este fichero se encuentra en la ruta `/usr/local/nagios/etc/object/commands.cfg`.

Aquí se define el nombre del servicio que se invocará en el fichero `services.cfg` y la instrucción que se ejecutará en la línea de comandos. Es posible utilizar macros definidas o genéricas (véase *2.1.2.1*) en este fichero para ajustarse a los requerimientos de cada plugin.

```
# 'check_win_snmp_disk' command definition
define command{
       command_name      check_win_snmp_disk
       command_line      /usr/local/nagios/libexec/check_win_snmp_disk
$ARG1$ $ARG2$ $ARG3$ $ARG4$
}
```

Una vez establecida la definición del comando, se debe proceder con la definición del servicio en el fichero `/usr/local/nagios/etc/lab/services.cfg`. Se especifica la descripción del servicio y el comando de comprobación al que se le pasarán los argumentos necesarios para la ejecución del script. Este plugin, por su diseño, necesita la dirección del host al que se realizará la petición, el nombre de la comunidad SNMP, y los márgenes de WARNING y CRITICAL para la información en Nagios.

```
define service{
       use                     generic-service
       hosts                   local1,local2,local3,local4,local6,
                               local7,local8,local9,local10
       service_description     Check WIN DISK
       check_command     check_win_snmp_disk!$HOSTADDRESS!public!90!100
       notifications_enabled   0
}
```

Otro script para monitorizar sistemas Windows es `check_win_cpuload`. Comprobará el porcentaje de utilización del procesador. Si éste tiene más de un núcleo, devuelve la media de utilización de todos ellos. Además se comprobará la memoria RAM disponible y utilizada por los PCs con `check_winmem`.

Para la correcta ejecución de estos scripts es necesario incluir, como argumentos, los porcentajes de utilización de los estados de aviso y crítico.

Los comandos y la difinición del servicio de los scripts `check_win_cpuload` y `check_winmem` deben realizarse del mismo modo que `check_win_snmp_disk`. Utilizan las mismas macros, lo único que cambiaría es el nombre del script.

El último plugin adicional que se utilizará en la monitorización de los host es `check_iftraffic64`. A este script debe pasársele como argumento la comunidad SNMP y recorrerá la MIB-2 en busca de las interfaces de red disponibles. Obtiene los datos de utilización de dicha interfaz y devuelve los valores medios de entrada y salida de datos y los valores acumulados desde el último reinicio del equipo.

Sus definiciones como comando y servicio son, respectivamente:

```
define command{
       command_name      check_iftraffic64
       command_line      /usr/local/nagios/libexec/check_iftraffic64.pl -H
                         $HOSTADDRESS$ -C $ARG1$ -b 100 -u m
}

define service{
       use                       generic-service,srv-pnp
       hosts                     local1,local2,local3,local4,local6,
                                 local7,local8,local9,local10
       service_description       Trafico
       check_command             check_iftraffic64!public
       notifications_enabled     0
}
```

Tras reiniciar Nagios, si la configuración es correcta, será posible visualizar en la página de los host los servicios establecidos. La *Figura 29* muestra los servicios activos del host `local1`.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
local1	Check WIN CPU LOAD	OK	10-22-2015 00:08:33	0d 1h 58m 34s	1/3	OK : CPU load 53%
	Check WIN MEM	OK	10-22-2015 00:09:01	0d 0h 42m 20s	1/3	OK MEM: usage 70.86 perc - 2971068 KBytes of 4192696 KBytes -w 90 -c 100
	Check WIN disk	OK	10-22-2015 00:10:33	0d 0h 27m 50s	1/3	OK - ALL DISKS MET MARGINS
	PING	OK	10-22-2015 00:14:32	0d 0h 28m 51s	1/3	PING OK - Packet loss = 0%, RTA = 1.56 ms
	Trafico	OK	10-22-2015 00:10:20	0d 0h 29m 2s	1/3	OK - Average IN: 101.55KB (0.81%), Average OUT: 2.44KB (0.02%)Total RX: 1.25GBytes, Total TX: 87.56MBytes
	Uptime	OK	10-22-2015 00:09:32	0d 1h 58m 26s	1/3	SNMP OK - Timeticks: (3344053) 9:17:20.53

Figura 29 - Servicios del host local1 en Nagios

La *Figura 30* muestra las gráficas generadas con PNP4Nagios al situar el cursor sobre el icono de integración, remarcado en la figura, del servicio "Tráfico".

Los scripts `check_win_cpuload` y `check_winmem` se han adaptado, modificando el código en Perl, para que ofrezcan información de rendimiento. Esto supondrá que PNP4Nagios será capaz de recibir esa información y representarla gráficamente. La *Figura 31* muestra una captura de la representación gráfica del uso de la CPU en el host `local1`.

Recordar que PNP4Nagios es capaz de almacenar datos de rendimiento para realizar gráficos históricos de la información de rendimiento obtenida de los plugins. Puede visualizarse con diferentes escalas temporales en la misma página.

El servidor Atlas es un equipo sensible y por ello tiene el acceso restringido a parte de la MIB-2. No es posible monitorizar el uso de las interfaces de red que posee. Solamente se monitoriza si el equipo está activo y, en caso de estarlo, cuánto tiempo ha pasado desde el último reinicio.

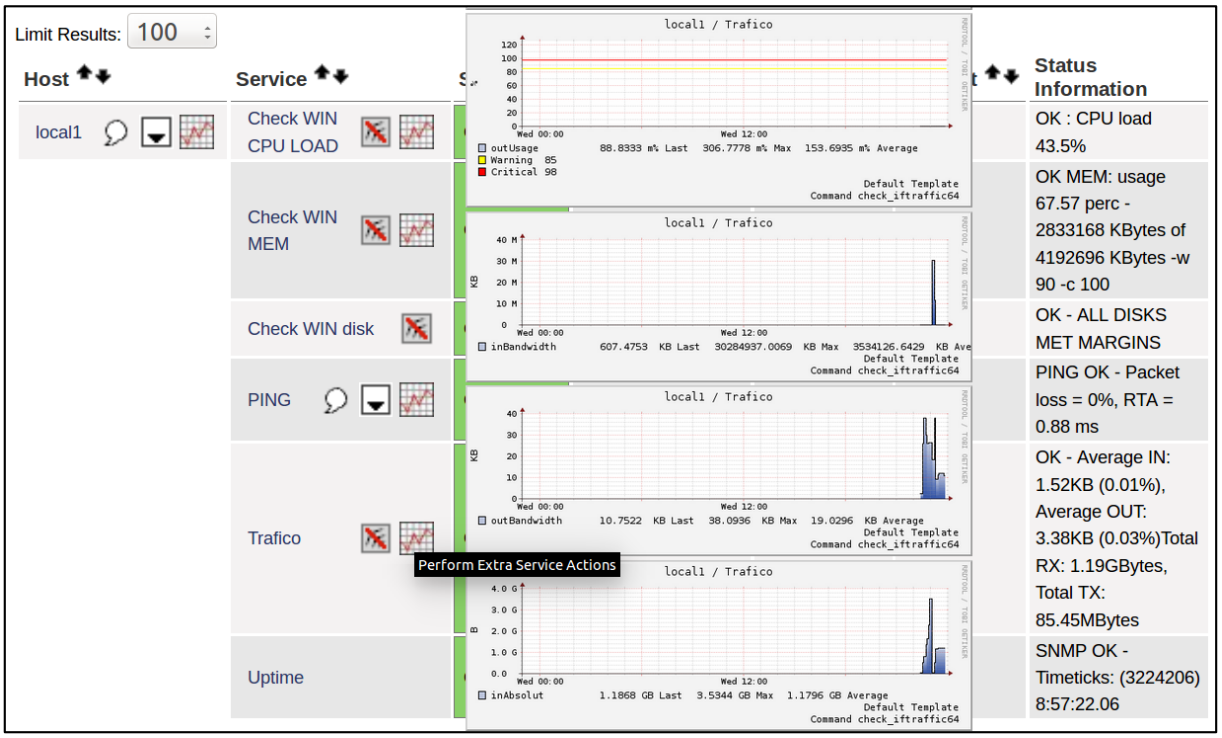


Figura 30 - Representación del tráfico de red del host local1

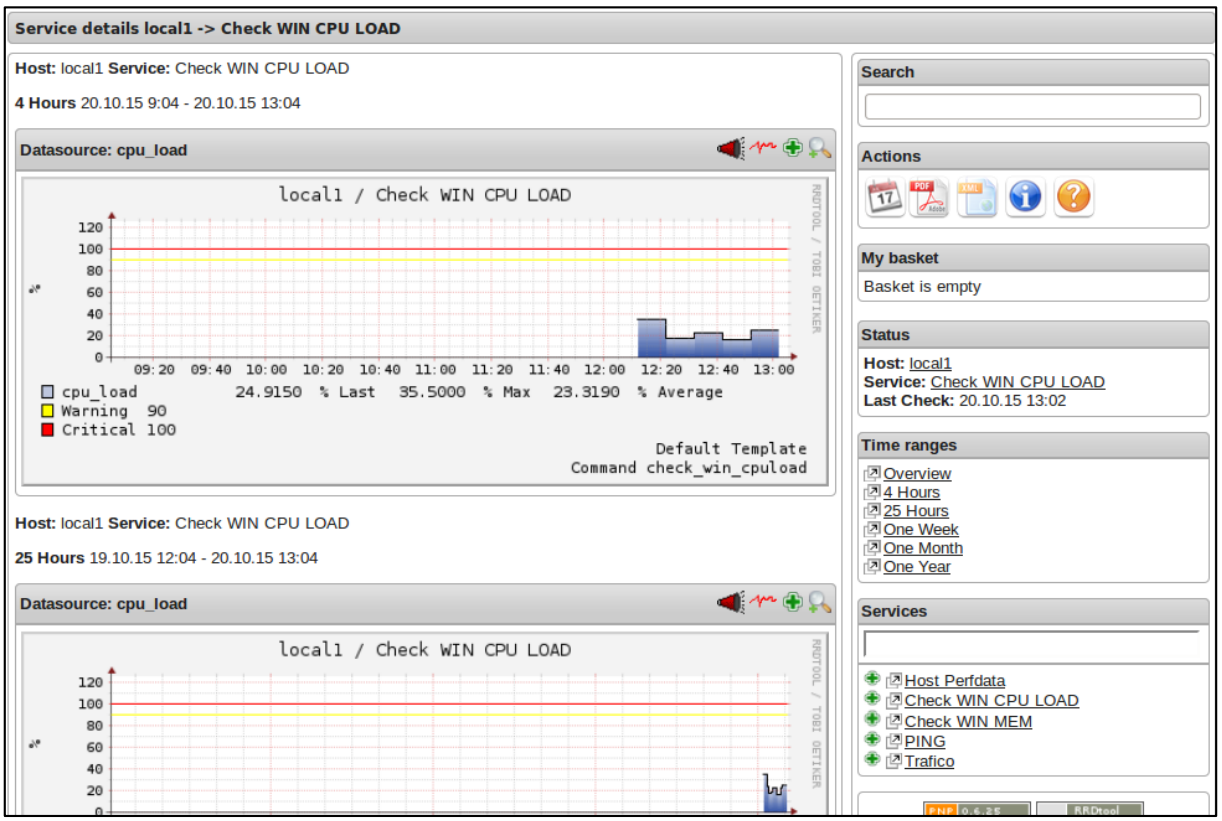


Figura 31 - Representación del uso de la CPU en el host local1

Para los routers, se ha utilizado también el plugin `check_iftraffic`. Así es capaz de mostrar el tráfico de entrada, salida y acumulado de cada interfaz. Sólo se han definido los servicios necesarios para monitorizar las interfaces que están siendo utilizadas, ya que las interfaces desconectadas son detectadas como errores con estado crítico.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
switchSMC24p	PING	OK	10-20-2015 23:48:17	0d 0h 34m 31s	1/3	PING OK - Packet loss = 0%, RTA = 2.67 ms
	Trafico Interfaz 01	OK	10-20-2015 23:49:50	0d 0h 32m 58s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 107.39B (0.00%)Total RX: 923.31KBytes, Total TX: 15.39MBytes
	Trafico Interfaz 02	OK	10-20-2015 23:49:31	0d 1h 18m 30s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 107.39B (0.00%)Total RX: 4.12MBytes, Total TX: 73.95MBytes
	Trafico Interfaz 03	OK	10-20-2015 23:50:16	0d 1h 12m 31s	1/3	OK - Average IN: 31.94B (0.00%), Average OUT: 136.12B (0.00%)Total RX: 2.82GBytes, Total TX: 204.38MBytes
	Trafico Interfaz 04	OK	10-20-2015 23:52:01	0d 1h 10m 46s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 106.95B (0.00%)Total RX: 23.72MBytes, Total TX: 292.38MBytes
	Trafico Interfaz 05	OK	10-20-2015 23:45:05	0d 1h 17m 42s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 108.01B (0.00%)Total RX: 15.35MBytes, Total TX: 249.77MBytes
	Trafico Interfaz 06	OK	10-20-2015 23:46:50	0d 1h 15m 57s	1/3	OK - Average IN: 34.92KB (0.28%), Average OUT: 1.13KB (0.01%)Total RX: 358.74MBytes, Total TX: 3.12GBytes
	Trafico Interfaz 07	OK	10-20-2015 23:48:35	0d 1h 14m 12s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 107.58B (0.00%)Total RX: 17.50MBytes, Total TX: 541.01MBytes
	Trafico Interfaz 08	OK	10-20-2015 23:50:20	0d 1h 12m 27s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 107.29B (0.00%)Total RX: 13.18MBytes, Total TX: 363.79MBytes
	Trafico Interfaz 09	OK	10-20-2015 23:52:05	0d 1h 10m 42s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 107.06B (0.00%)Total RX: 4.84MBytes, Total TX: 229.21MBytes
	Trafico Interfaz 10	OK	10-20-2015 23:45:09	0d 1h 17m 38s	1/3	OK - Average IN: 0.00B (0.00%), Average OUT: 106.68B (0.00%)Total RX: 910.11KBytes, Total TX: 15.34MBytes
	Trafico Interfaz 11	OK	10-20-2015 23:50:05	0d 1h 12m 42s	1/3	OK - Average IN: 1.57KB (0.01%), Average OUT: 21.63KB (0.17%)Total RX: 2.20GBytes, Total TX: 502.90MBytes
	Trafico Interfaz 15	OK	10-20-2015 23:48:24	0d 1h 14m 23s	1/3	OK - Average IN: 74.12B (0.00%), Average OUT: 107.00B (0.00%)Total RX: 9.77MBytes, Total TX: 14.94MBytes
	Trafico Interfaz 16	OK	10-20-2015 23:50:09	0d 1h 12m 38s	1/3	OK - Average IN: 667.25B (0.01%), Average OUT: 816.42B (0.01%)Total RX: 76.26MBytes, Total TX: 93.07MBytes
	Uptime	OK	10-20-2015 23:44:16	0d 0h 28m 32s	1/3	SNMP OK - Timeticks: (13382330) 1 day, 13:10:23.30

Figura 32 – Servicios del switch SMC de 24 puertos

En los nodos ATM se ha utilizado el plugin `check_snmp_iproute` que comprueba que las rutas hacia las VLANs son correctas. Desde el nodo ATM del Servicio de Informática (véase *Figura 28*) se comprueba la ruta hacia las VLANs de todos los nodos, en el resto sólo las propias.

Para comprobar que las rutas son correctas el plugin comprueba los objetos de la MIB IP-FORWARD. Concretamente la tabla `ipCidrRouteTable`, de la cual se pueden obtener las direcciones IP de las VLANs, sus máscaras de subred y sus puertas de enlace. La *Figura 33* muestra los servicios definidos para el nodo ATM del Servicio de Informática. Sus definiciones como comando y servicio son, respectivamente:

```
define command{
       command_name check_snmp_iproute
       command_line /usr/local/nagios/libexec/check_snmp_iproute -H
                    $HOSTADDRESS$ -C $ARG1$ -r $ARG2$ -m $ARG3$ -g $ARG4$
}
define service{
       use                     generic-service
       host_name               ATM_Informatica
       service_description     VLAN 192.168.1.32/27
       check_command           check_snmp_iproute!solomira
}
```

Por último, la configuración del equipo `local5`, en el que Nagios está corriendo, hace uso de los plugins locales incorporados en el paquete de plugins de Nagios. Son específicos para monitorizar el sistema Linux local. Estos, proporcionan información sobre el uso de la CPU (*Current Load*), el espacio disponible en la partición del disco duro (*Root Partition*), el uso de la partición del disco duro para la memoria virtual (*Swap Usage*) y el número de usuarios identificados en el sistema (*Current Users*). La *Figura 34* muestra los servicios descritos para el equipo `local5`, el host local para Nagios

ATM_Informatica	Enlace con Enfermeria	OK	10-21-2015 01:47:40	0d 0h 19m 56s	1/3	OK - 192.168.0.8 (Router ID 192.168.0.9) found
	Enlace con Laboratorio Telematica	OK	10-21-2015 01:49:26	0d 0h 18m 10s	1/3	OK - 192.168.0.20 (Router ID 192.168.0.21) found
	Enlace con Medicina	OK	10-21-2015 01:56:42	0d 0h 20m 54s	1/3	OK - 192.168.0.4 (Router ID 192.168.0.5) found
	Enlace con Nautica	OK	10-21-2015 01:48:28	0d 0h 19m 8s	1/3	OK - 192.168.0.0 (Router ID 192.168.0.1) found
	Enlace con Paraninfo	OK	10-21-2015 01:50:15	0d 0h 17m 21s	1/3	OK - 192.168.0.12 (Router ID 192.168.0.13) found
	Enlace con Torrelavega	OK	10-21-2015 01:52:01	0d 0h 15m 35s	1/3	OK - 192.168.0.16 (Router ID 192.168.0.17) found
	PING	OK	10-21-2015 01:54:59	1d 6h 11m 15s	1/3	PING OK - Packet loss = 0%, RTA = 1.67 ms
	Uptime	OK	10-21-2015 01:50:56	0d 5h 26m 37s	1/3	SNMP OK - Timeticks: (10833098) 1 day, 6:05:30.98
	VLAN 192.168.1.0/27	OK	10-21-2015 01:53:47	0d 0h 13m 49s	1/3	OK - 192.168.1.0 (Router ID 192.168.1.1) found
	VLAN 192.168.1.128/27	OK	10-21-2015 01:56:45	0d 0h 20m 51s	1/3	OK - 192.168.1.128 (Router ID 192.168.0.10) found
	VLAN 192.168.1.32/27	OK	10-21-2015 01:48:32	0d 0h 19m 4s	1/3	OK - 192.168.1.32 (Router ID 192.168.0.2) found
	VLAN 192.168.1.64/27	OK	10-21-2015 01:50:18	0d 0h 17m 18s	1/3	OK - 192.168.1.64 (Router ID 192.168.0.14) found
	VLAN 192.168.1.96/27	OK	10-21-2015 01:52:04	0d 0h 15m 32s	1/3	OK - 192.168.1.96 (Router ID 192.168.0.6) found
	VLAN 192.168.2.0/27	OK	10-21-2015 01:53:51	0d 0h 13m 45s	1/3	OK - 192.168.2.0 (Router ID 192.168.2.1) found
	VLAN 192.168.2.128/27	OK	10-21-2015 01:56:49	0d 0h 20m 47s	1/3	OK - 192.168.2.128 (Router ID 192.168.0.10) found
	VLAN 192.168.2.32/27	OK	10-21-2015 01:48:35	0d 0h 19m 1s	1/3	OK - 192.168.2.32 (Router ID 192.168.0.2) found
	VLAN 192.168.2.64/27	OK	10-21-2015 01:50:21	0d 0h 17m 15s	1/3	OK - 192.168.2.64 (Router ID 192.168.0.14) found
	VLAN 192.168.2.96/27	OK	10-21-2015 01:52:08	0d 0h 15m 28s	1/3	OK - 192.168.2.96 (Router ID 192.168.0.6) found
	VLAN 192.168.3.0/27	OK	10-21-2015 01:53:54	0d 0h 13m 42s	1/3	OK - 192.168.3.0 (Router ID 192.168.3.1) found
	VLAN 192.168.3.128/27	OK	10-21-2015 01:56:52	0d 0h 20m 44s	1/3	OK - 192.168.3.128 (Router ID 192.168.0.10) found
	VLAN 192.168.3.32/27	OK	10-21-2015 01:48:38	0d 0h 18m 58s	1/3	OK - 192.168.3.32 (Router ID 192.168.0.2) found
	VLAN 192.168.3.64/27	OK	10-21-2015 01:50:25	0d 0h 17m 11s	1/3	OK - 192.168.3.64 (Router ID 192.168.0.14) found
	VLAN 192.168.3.96/27	OK	10-21-2015 01:52:11	0d 0h 15m 25s	1/3	OK - 192.168.3.96 (Router ID 192.168.0.6) found
	VLAN 192.168.4.0/27	OK	10-21-2015 01:53:57	0d 0h 13m 39s	1/3	OK - 192.168.4.0 (Router ID 192.168.4.1) found

Figura 33 - Servicios del nodo ATM del Servicio de Informática

local5	Current Load	OK	10-21-2015 02:08:04	1d 9h 59m 12s	1/4	OK - load average: 0.14, 0.21, 0.22
	Current Users	OK	10-21-2015 02:08:48	1d 9h 59m 28s	1/4	USERS OK - 4 users currently logged in
	HTTP	OK	10-21-2015 02:07:43	1d 9h 59m 32s	1/4	HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.001 second response time
	PING	OK	10-21-2015 02:08:22	5d 11h 52m 58s	1/3	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-21-2015 02:09:56	1d 10h 1m 15s	1/4	DISK OK - free space: / 171594 MB (92% inode=97%):
	Swap Usage	OK	10-21-2015 02:10:14	1d 9h 59m 8s	1/4	SWAP OK - 100% free (8071 MB out of 8071 MB)

Figura 34 - Servicios del host local5

4.3 Gestión de mapas en NagVis

NagVis incorpora por defecto un conjunto de mapas de demostración para que el usuario tenga una visión de lo que esta herramienta permite realizar. Estos mapas se muestran en la pantalla inicial de NagVis pero es posible borrarlos. Tras identificarse con un usuario con privilegios de administrador, es posible borrar los mapas de ejemplo desde la pestaña *Options* y después seleccionar *Manage Maps*. La *Figura 35* muestra la sección correspondiente para borrar los mapas mencionados.

En la parte superior el mismo menú se pueden crear mapas nuevos. El sistema de creación de un mapa objeto a objeto en NagVis, puede resultar un proceso tedioso para entorno grandes. La opción "Automap" crea el mapa a partir de las relaciones parentales entre objetos, establecidas en los ficheros de definición de objetos de Nagios. La *Figura 36* muestra cómo crear un mapa automático.

Tras pulsar el botón *Create* se mostrará el mapa renderizado de la forma que NagVis escoja. Las opciones de renderizado modifican la visualización del mapa en cuanto a la forma que tiene NagVis de organizar los objetos. Para este caso se recomienda la opción *Undirected* que puede seleccionarse en la pestaña *Edit Map* y *Map Options* y cambiar el parámetro `render_mode`. También es posible añadir una barra de zoom y un valor de zoom por defecto como muestra la *Figura 37*.

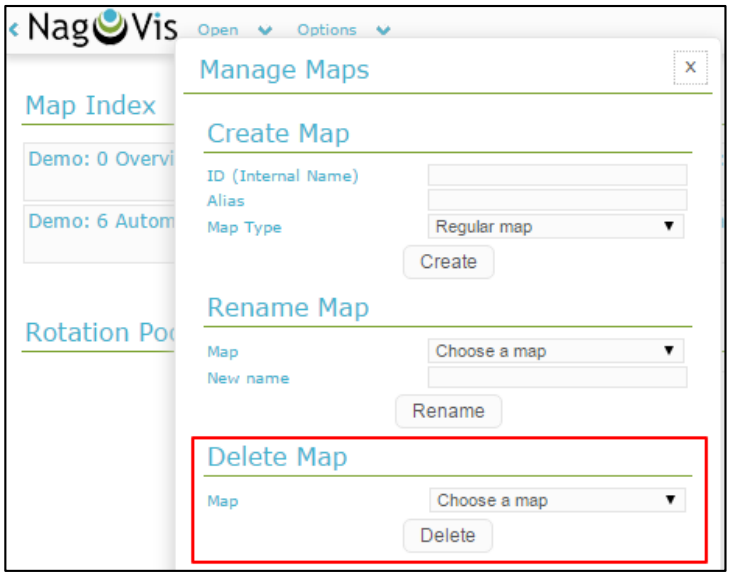


Figura 35 - Ventana de gestión de mapas en NagVis

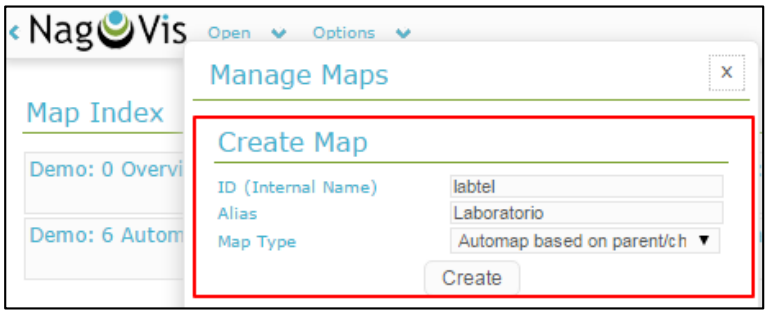


Figura 36 - Ventana de gestión de mapas en NagVis (II)

Figura 37 - Ventana de edición de mapas en NagVis

La *Figura 38* muestra el mapa automático creado por NagVis en base a las relaciones parentales definidas en los objetos. Al pasar el cursor sobre cualquier nodo, se muestra un popup en el que aparecen los servicios definidos y su estado.

Si se desea más información sobre los servicios de un nodo, es posible pinchar sobre el icono del nodo. Esta acción abrirá la página de Nagios en la que se muestra la información del estado de los servicios del host determinado.

Los iconos reflejan el estado del equipo y sus servicios. El icono morado con aspa blanca, representa un host inalcanzable. El rojo con aspa blanca, hosts caídos o con errores críticos. El círculo gris con interrogación blanca, indica que no se ha comprobado el estado del host. El círculo amarillo con un rayo blanco significa que algún servicio está en estado WARNING y el icono verde que el host está activo y todos sus servicios en estado OK.

Una vez creado el mapa automático será posible editarlo para ajustarlo a las necesidades. Para ello, visualizando el mapa, en el menú *Actions* pinchar en *Export to static Map*. Se creará un mapa nuevo que será editable a través del menú *Edit Map*.

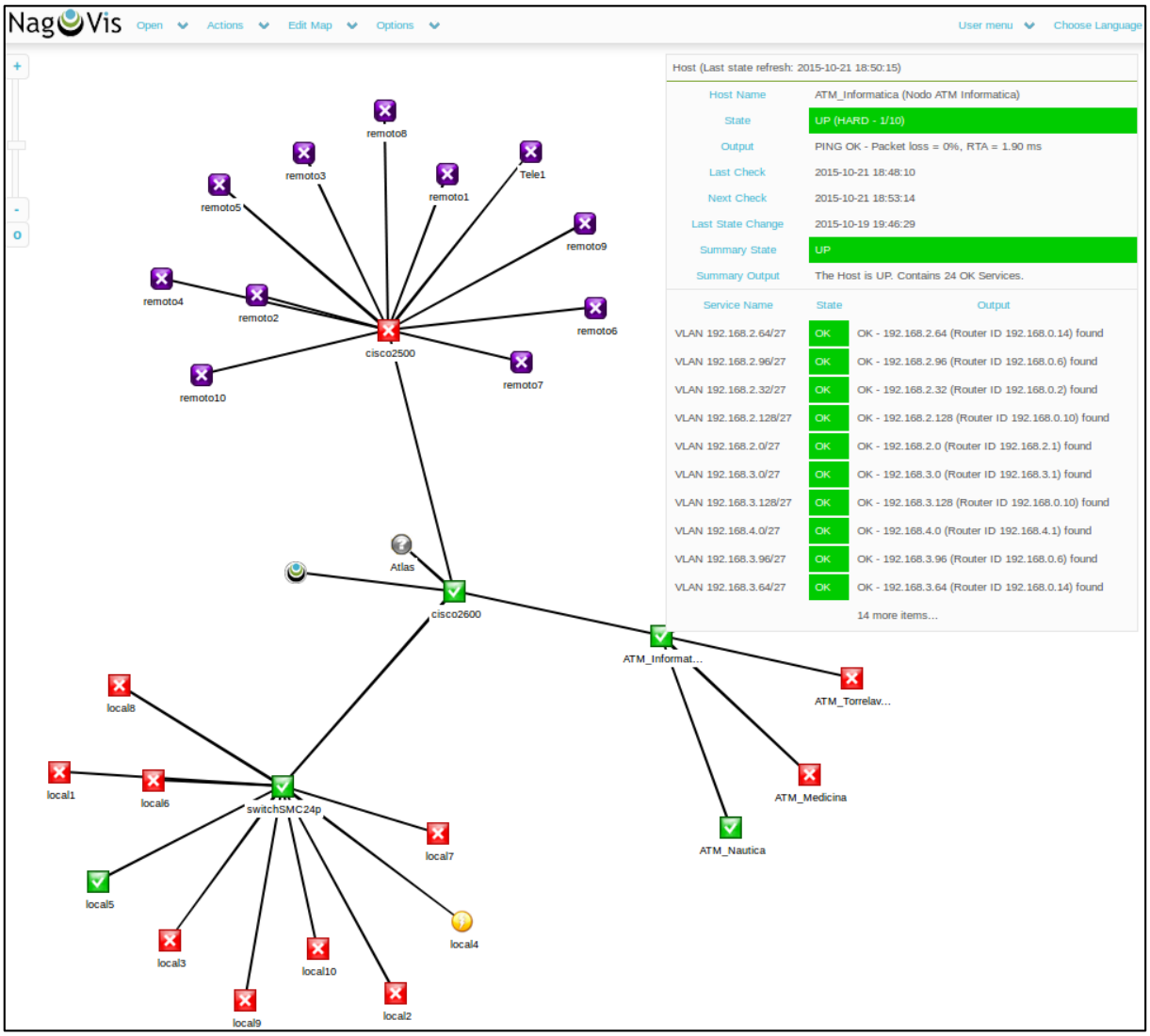


Figura 38 - Mapa de la topología de red de los laboratorios en NagVis

4.4 Agente SNMP para Windows

Para el desarrollo de un subagente SNMP, se han utilizado códigos de ejemplo con fines didácticos. Se ha utilizado principalmente el código de CodeProject [11] para su desarrollo.

Los subagentes que soporta Windows tienen formato DLL (Dynamic-Link Library), y que se cargan como un servicio más en el sistema bajo el agente SNMP maestro.

Este subagente se ha desarrollado con Visual Studio 2015 en lenguaje C++ y ha sido compilado para ejecutarse en máquinas de 64 bits.

En el diseño del subagente se ha incorporado una rutina que comprobará las actualizaciones de Windows pendientes de instalar, cuando se cargue el agente en el arranque del sistema. Se hace uso de la API de Windows WUA (*Windows Update Agent*) para obtener el número de actualizaciones pendientes.

Si existiese alguna, el agente enviaría una trap al gestor con el nombre del equipo y el número de ellas.

En caso de que se cumplan las condiciones necesarias para enviar traps, éstas serán enviadas con una frecuencia de un día. Como en ocasiones, esto puede llegar a resultar molesto para el administrador, se ha definido un objeto de la MIB con el que podrá activarse o desactivarse el envío de traps.

Los objetos de la MIB LAB-TELEMATICA (véase *Anexo III*) cuelgan del nodo `enterprises` de la MIB-II (véase *2.2.2*). Se ha elegido el OID 90000 para que no interfiera con ningún otro registrado por la organización que regula la asignación de números corporativos, IANA (*Internet Assigned Numbers Authority*) [12].

Para instalar el subagente de Windows es necesario copiar el fichero `.dll` en un directorio del PC, por ejemplo, `C:\Windows\System32`. Una vez copiado es necesario crear las claves en el registro para que el SNMP Master Agent pueda detectar el subagente.

Las claves se pueden crear manualmente en el editor del registro de Windows (`regedit.exe`) o importando un fichero de registro con las claves. El fichero debe tener la extensión `.reg` y debe contener las claves definidas. Un posible fichero `.reg` contendría lo siguiente:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\LabTelematica\subagenteSNMP\CurrentVersion]
"Pathname"="C:\\Windows\\System32\\LabTelematica.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SNMP\Parameters\Exten
sionAgents]
"LabTelematica"="SOFTWARE\\LabTelematica\\subagenteSNMP\\CurrentVersion"
```

La *Figura 39* ilustra cómo deben quedar las claves en el editor del registro de Windows.

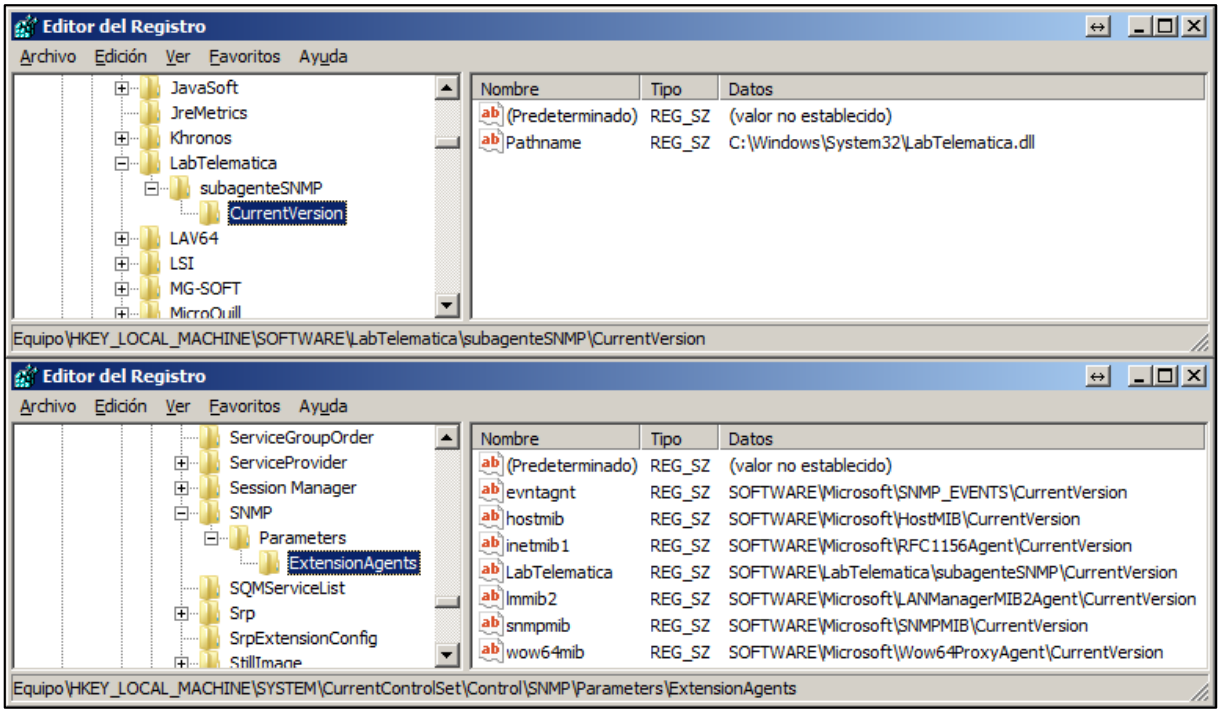


Figura 39 - Claves del registro de Windows necesarias para cargar el subagente SNMP

Tras establecer las claves, se puede comprobar si el SNMP Master Agent Configurator ha detectado el nuevo subagente (véase *Figura 40*).

Ahora se debe añadir la MIB compilada por el MIB Compiler en formato `.smidb` pulsado en botón *Add* en la pestaña *Easy Agent Extensions* (véase *Figura 41*).

El último paso es configurar la comunidad y dirección IP a la que deben mandarse las traps, en este caso Nagios se ha instalado en el equipo `local5` y su IP es 192.168.110.5 como muestra la *Figura 42*.

Tras aceptar los cambios, el programa pregunta si se desea reiniciar el servicio del SNMP Master Agent. Una vez reiniciado, el subagente estará funcionando. Téngase en cuenta que el subagente comprueba las actualizaciones pendientes de instalar en Windows, y es un proceso que lleva dos o tres minutos dependiendo del número de ellas. Por ello, el agente no responderá a ninguna petición hasta que no detecte si hay o no actualizaciones pendientes.

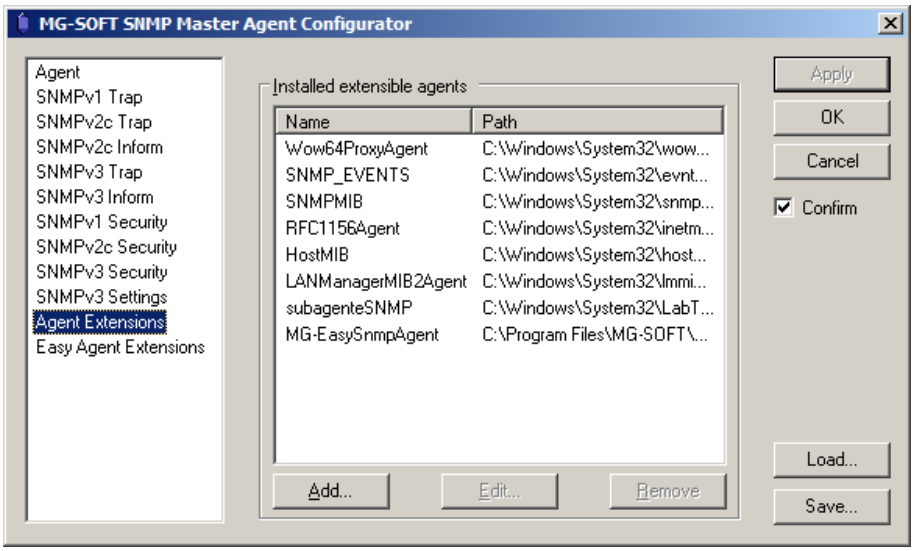


Figura 40 - Comprobar la inclusión del subagente al SNMP Master Agent

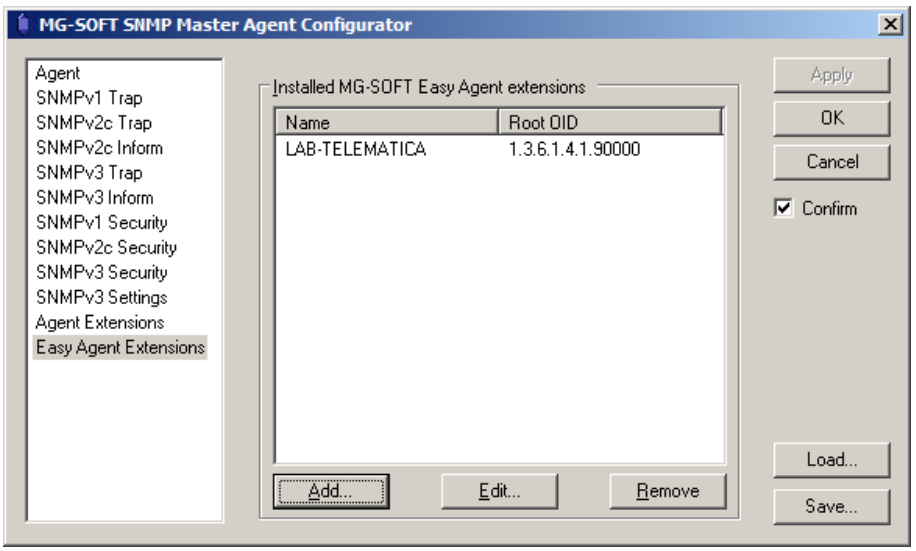


Figura 41 - Añadir MIB al SNMP Master Agent

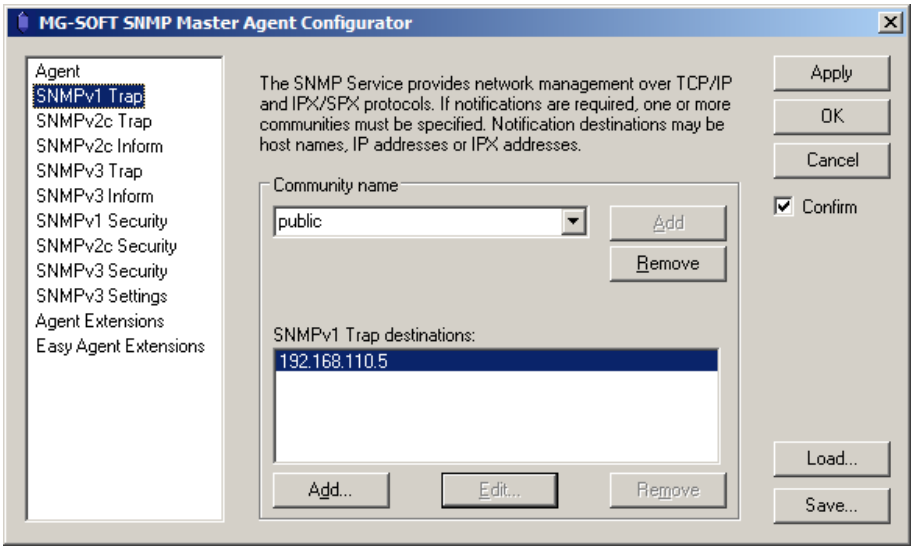


Figura 42 - Configurar el envío de traps en SNMP Master Agent

4.5 Integración de SNMPTT con Nagios

Las traps enviadas por el agente SNMP de Windows son capturadas por el servicio `snmptrapd` de Net-SNMP y se traducen utilizando SNMPTT para que Nagios obtenga la información de ellas en el formato correcto.

El primer paso es copiar la MIB creada con MIB-Builder en la siguiente ruta:

```
cp LAB-TELEMATICA.my /usr/share/snmp/mibs/
```

Ahora se debe convertir dicha MIB utilizando la herramienta `snmpttconvertmib` incluida en el paquete SNMPTT. El parámetro de entrada es MIB, con ella genera un fichero con la información de la trap definida en la MIB (véase *Anexo III*) y por último se hace uso de un plugin de Nagios, `submit_check_result`, que envía la información al propio servicio de Nagios.

```
snmpttconvertmib --in=/usr/share/snmp/mibs/LAB-TELEMATICA.my --
     out=/etc/snmp/snmptt.conf.updates --
     exec='/usr/local/nagios/libexec/eventhandlers/submit_check_result $r
     TRAP_Updates 1 "El equipo $1 tiene $2 actualizaciones pendientes"'
```

Con la instrucción `exec` se indica el nombre del host con la macro `$r`, el nombre del servicio a configurar en Nagios, `TRAP_Updates`, el estado de la alerta (`0=OK, 1=WARNING, 2=CRITICAL`) y el texto de información que devuelve la trap, dónde `$1` es el nombre del equipo y `$2` el número de actualizaciones pendientes.

El fichero `/etc/snmp/snmptt.conf.updates` se genera de manera automática y debe modificarse para que se ajuste a lo necesario. Debe coincidir con la siguiente definición:

```
EVENT hayActualizaciones .1.3.6.1.4.1.90000.0.0.* "Status Events" Normal
FORMAT $*
EXEC /usr/local/nagios/libexec/eventhandlers/submit_check_result $r
TRAP_Updates 1 "El equipo $1 tiene $2 actualizaciones pendientes"
SDESC
Variables:
  1: hostname
  2: numActualizaciones
EDESC
```

El siguiente paso es añadir al final del fichero `/etc/snmp/snmptt.ini` la ruta del fichero creado con `snmpttconvertmib` del siguiente modo:

```
[TrapFiles]
snmptt_conf_files = <<END
/etc/snmp/snmptt.conf
/etc/snmp/snmptt.conf.updates
END
```

Ahora hay que añadir la definición del servicio de la trap, como una plantilla, en el fichero `/usr/local/nagios/etc/objects/templates.cfg`.

```
define service{
       name trap-service
       use generic-service
       register 0
       service_description TRAP_Updates
       is_volatile 1
       check_command check-host-alive
       max_check_attempts 1
       normal_check_interval 1
       retry_check_interval 1
       active_checks_enabled 0
       passive_checks_enabled 1
       check_period none
       notification_interval 0
       contact_groups admins
}
```

El servicio asociado a cada equipo que envíe las traps deberá estar asociado al servicio TRAP_Updates definido en Nagios:

```
define service{
       use                     trap-service
       hosts                   local1,local2,local3,local4, local6,
                               local7,local8,local9,local10
       service_description     TRAP_Updates
}
```

El plugin `submit_check_result` que gestionará las traps en Nagios debe crearse en la ruta específica y con los permisos correspodientes:

```
mkdir /usr/local/nagios/libexec/eventhandlers/
cd /usr/local/nagios/libexec/eventhandlers/
touch submit_check_result
```

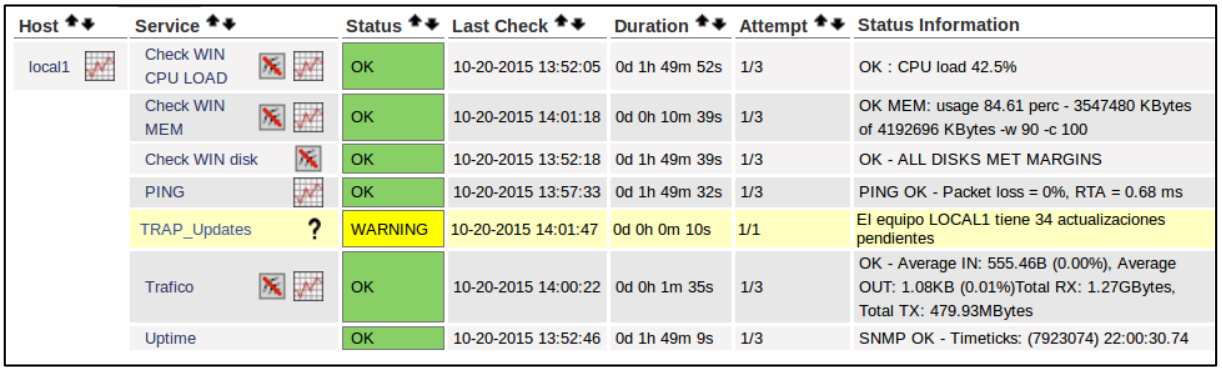
Añadir al fichero `submit_check_result` el siguiente código:

```
#!/bin/sh
echocmd="/bin/echo"
CommandFile="/usr/local/nagios/var/rw/nagios.cmd"
datetime=`date +%s`
cmdline="[$datetime] PROCESS_SERVICE_CHECK_RESULT;$1;$2;$3;$4"
`$echocmd $cmdline >> $CommandFile`
```

Es necesario darle permisos de ejecución y cambiar el propietario con las instrucciones siguientes:

```
chown nagios:nagcmd
     /usr/local/nagios/libexec/eventhandlers/submit_check_result
chmod 755 /usr/local/nagios/libexec/eventhandlers/submit_check_result
service snmpd restart
```

Tras reiniciar el servicio `snmpd` ya se pasarán las traps a Nagios. La *Figura 43* muestra un ejemplo de la recepción de traps en Nagios.



Host	Service	Status	Last Check	Duration	Attempt	Status Information
local1	Check WIN CPU LOAD	OK	10-20-2015 13:52:05	0d 1h 49m 52s	1/3	OK : CPU load 42.5%
	Check WIN MEM	OK	10-20-2015 14:01:18	0d 0h 10m 39s	1/3	OK MEM: usage 84.61 perc - 3547480 KBytes of 4192696 KBytes -w 90 -c 100
	Check WIN disk	OK	10-20-2015 13:52:18	0d 1h 49m 39s	1/3	OK - ALL DISKS MET MARGINS
	PING	OK	10-20-2015 13:57:33	0d 1h 49m 32s	1/3	PING OK - Packet loss = 0%, RTA = 0.68 ms
	TRAP_Updates	WARNING	10-20-2015 14:01:47	0d 0h 0m 10s	1/1	El equipo LOCAL1 tiene 34 actualizaciones pendientes
	Trafico	OK	10-20-2015 14:00:22	0d 0h 1m 35s	1/3	OK - Average IN: 555.46B (0.00%), Average OUT: 1.08KB (0.01%)Total RX: 1.27GBytes, Total TX: 479.93MBytes
	Uptime	OK	10-20-2015 13:52:46	0d 1h 49m 9s	1/3	SNMP OK - Timeticks: (7923074) 22:00:30.74

Figura 43 - Recepción de trap del host local1 en Nagios

4.6 Verificar la configuración de Nagios

En este punto, los archivo de configuración deben estarán preparados para su uso. Ahora se puede verificar que todas las instrucciones de configuración son correctas y que Nagios arranca correctamente con dicha configuración. Es posible comprobarlo mediante la ejecución del siguiente comando:

```
root@ubuntu:/# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
 Checked 120 services.
 Checked 32 hosts.
 Checked 6 host groups.
 Checked 1 contacts.
 Checked 1 contact groups.
 Checked 30 commands.
 Checked 5 time periods.
Checking for circular paths...
 Checked 32 hosts
 Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight
```

El mensaje anterior muestra un resumen de los objetos definidos y que no existen errores en la configuración.

5. CONCLUSIONES Y LÍNEAS FUTURAS

En este trabajo fin de grado se ha implementado un sistema de monitorización de la red de comunicaciones instalada en del Laboratorio docente de Telemática de la Escuela de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria.

La memoria de este TFG se ha desarrollado siguiendo tres fases diferenciadas:

Por un lado se han analizado las principales características de Nagios y de algunas de las herramientas más empleadas en el campo de la monitorización SNMP, como Net-SNMP.

La segunda parte detalla el proceso de instalación del sistema distribuido de monitorización. Se ha intentado que esta sección sea genérica, es decir, que pueda emplearse en cualquier otro entorno, independientemente de la topología de red que se desee monitorizar.

En cuanto a la tercera parte, más específica, muestra con detalle la personalización del entorno de monitorización para el Laboratorio docente en cuestión.

Respecto a las conclusiones del TFG, se pueden resumir en unas pocas ideas.

La elección del enfoque de monitorización a emplear debe siempre partir del objetivo que se persigue con el mismo, en este caso ha sido el de medir el rendimiento y contabilizar el uso de la red. Se ha tenido en cuenta que una monitorización activa agrega tráfico a la red, pero en el entorno de trabajo no supone una desventaja ya que son pocos los dispositivos a monitorizar.

Uno de los beneficios de la monitorización activa es el ahorro energético, ya que puede detectarse si un equipo se ha dejado encendido fuera del horario docente sin motivo aparente.

La monitorización pasiva viene de la mano del subagente SNMP y de las traps. Estas notificaciones informan al gestor de la red si existen actualizaciones pendientes de instalar de los PCs con Windows. Gracias a las traps se pueden realizar tareas de mantenimiento en los equipos que lo necesiten garantizando así, que estén actualizados.

Nagios es una herramienta de monitorización muy completa y altamente configurable. Quizás, el único pero que se le puede poner, es la gestión de traps. Éste no ofrece un método nativo, fácilmente configurable, que permita la recepción de traps o notificaciones SNMP. Aunque gracias a SNMPTT, se ha conseguido que cumpliese este cometido.

Una ventaja de Nagios frente a otras plataformas de monitorización, es que es un sistema de monitorización multiusuario, al que varios administradores pueden acceder simultáneamente gracias a la interfaz web.

Podría incluso utilizarse como material docente complementario para que los alumnos visualicen cierta información y comprendan mejor el concepto de monitorización. Para dicho cometido, se podría definir en Nagios un usuario sin privilegios de administrador.

En cuanto a líneas futuras, una de las mejoras posibles, sería la implementación de un subagente SNMP que gestione tablas. De esta forma sería posible devolver el nombre de todas las actualizaciones pendientes de instalar y poder así visualizarlo en Nagios.

MGSOFT provee, en uno de sus módulos, un SDK propio para el desarrollo de agentes SNMP. Además incluye un código de ejemplo que utiliza tablas, pero funciona como aplicación de consola y no se integra en el sistema operativo. Podría utilizarse dicho SDK para crear un subagente SNMP en formato DLL para Windows.

Otra mejora a implementar en un futuro sería la edición de un mapa automático creado con NagVis, así como la definición de líneas que representen los enlaces entre equipos y que muestren el tráfico bidireccional que pasa por ellos. De este modo se visualizaría el uso del ancho de banda instantáneo en cada enlace.

BIBLIOGRAFÍA

Durante desarrollo del proyecto se ha utilizado la siguiente documentación:

[1] J. Á. Irastorza Teja, «Introducción a la Gestión de Redes,» [En línea]. Disponible en: http://www.tlmat.unican.es/siteadmin/submaterials/1551.pdf.

[2] Nagios, «Nagios Core Documentation,» 2001. [En línea]. Disponible en: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/toc.html.

[3] Net-SNMP, «Net-SNMP,» [En línea]. Disponible en: http://www.net-snmp.org/.

[4] A. Burger, «SNMPTT,» [En línea]. Disponible en: http://www.snmptt.org/docs/snmptt.shtml#Requirements.

[5] M.-S. Corporation., «MG-SOFT,» [En línea]. Disponible en: www.mg-soft.com.

[6] PNP4Nagios, «PNP4Nagios,» [En línea]. Disponible en: http://docs.pnp4nagios.org/.

[7] The PHP Group, «PHP Introducción Manual,» [En línea]. Disponible en: http://php.net/manual/es/intro.image.php.

[8] NagVis Project, «NagVis,» [En línea]. Disponible en: http://www.nagvis.org/.

[9] M. Kettner, «MK Livestatus,» [En línea]. Disponible en: http://mathias-kettner.de/checkmk_livestatus.html.

[10] S. W. Stephan, «SANJAY WILLIE'S Human Language. Asterisk | Nagios | OpenSource | Microsoft | Security,» 2009. [En línea]. Disponible en: http://highsecurity.blogspot.com.es/2009/11/nagios-receive-traps-with-snmptt_08.html.

[11] Ramanan.T, «How to develop a SNMP extension agent DLL,» CodeProject, [En línea]. Disponible en: http://www.codeproject.com/Articles/9024/How-to-develop-a-SNMP-extension-agent-DLL.

[12] IANA, «Entreprise Numbers,» [En línea]. Disponible en: http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers.

ANEXOS

Anexo I Información adicional de Nagios

Opción	Descripción
log_file	Especifica el fichero de log a utilizar; por defecto [localstatedir]/nagios.log
cfg_file	Especifica el fichero de configuración del que se leerán las definiciones de los objetos; pueden especificarse varios
cfg_dir	Especifica el directorio de configuración donde todos los ficheros serán tratados como definiciones de objetos; pueden especificarse varios.
resource_file	Fichero que almacena información adicional de definiciones de macros; [sysconfdir]/resource.cfg
temp_file	Directorio en el que se alojará un fichero con información temporal; [localstatedir]/nagios.tmp
lock_file	DIrectorio en el que se alojará un fichero que será utilizado para sincronización; por defecto [localstatedir]/nagios.lock
temp_path	Directorio en el que Nagios podrá crear ficheros temporales; por defecto /tmp
status_file	Directorio en el que se alojará un fichero que guarda el estado actual de todos los hosts y servicios; por defecto [localstatedir]/status.dat
status_update_interval	Especifica la frecuencia (en segundos) con la que el status_file se actualizará; por defecto 10 (segundos)
nagios_user	Usuario con el que se ejecutará el servicio de Nagios
nagios_group	Grupo con el que se ejecutará Nagios
command_file	Especifica la ruta a la línea de commandos externa que otro proceso utilizará para controlar el servicio de Nagios; por defecto [localstatedir]/rw/nagios.cmd
use_syslog	Define si Nagios debe reportar mensajes al log del Sistema así como al propio log de Nagios; por defecto 1 (activado)
state_retention_file	Ruta al fichero que aloja la información de estado cuando se apaga el equipo en el que Nagios está ejecutándose; por defecto [localstatedir]/ retention.dat
retention_update_ interval	Frecuencia (en segundos) con la que el fichero retention file debe actualizarse; por defecto 60 (segundos)
service_check_timeout	Especifica después de cuántos segundos se debe asumir que la comprobación de un servicio ha fallado; por defecto 60 (segundos)
host_check_timeout	Especifica después de cuántos segundos se debe asumir que la comprobación de un host ha fallado; por defecto es 30 (segundos)
event_handler_timeout	Especifica después de cuántos segundos debe finalizarse un controlador de eventos; por defecto 30 (segundos)
notification_timeout	Especifica después de cuántos segundos se debe suponer que un intento de notificación ha fallado; por defecto 30 (segundos)
enable_environment_ macros	Especifica si Nagios debe pasar todas las macros a los plugins como variables de entorno; por defecto es 1 (activado)
interval_length	Especifica el número de segundos del "intervalo unidad"; el valor predeterminado es 60, lo que significa que un intervalo es de un minuto; no se recomienda cambiar esta opción, ya que podría producir un comportamiento no deseado

Tabla 8 - Parámetros del fichero de configuración de Nagios [2]

Opción	Descripción
HOSTNAME	Nombre corto y único del host, apunta a la directiva host_name en la definición del host
HOSTADDRESS	La dirección IP o el nombre de dominio del host, apunta a la directiva address en la definición del host
HOSTDISPLAYNAME	Nombre descriptivo del host, apunta a la directiva alias en la definición del host
HOSTSTATE	Estado actual del host (UP, DOWN, o UNREACHABLE)
HOSTGROUPNAMES	Nombres cortos de todos los grupos a los que pertenece un host; separados por comas
LASTHOSTCHECK	La fecha y hora de la última comprobación del host; en Unix el "Timestamp" (número de segundos desde el 01-01-1970)
LASTHOSTSTATE	El ultimo estado conocido del host (UP, DOWN, o UNREACHABLE)
SERVICEDESC	Descripción del servicio; apunta a la directiva description en la definición del servicio
SERVICESTATE	Estado actual del servicio (OK, WARNING, o UNKNOWN)
SERVICEGROUPNAMES	Nombres cortos de todos los grupos a los que pertenece un servicio; separados por comas
CONTACTNAME	Nombre corto y único del contacto, apunta a la directiva contact_name en la definición del contacto
CONTACTALIAS	Nombre descriptivo del contacto, apunta a la directiva alias en la definición del contacto
CONTACTEMAIL	Dirección e-mail del contacto, apunta a la directiva email en la definición del contacto
CONTACTGROUPNAMES	Nombres cortos de todos los grupos a los que pertenece un contacto; separados por comas

Tabla 9 - Macros soportadas por Nagios [2]

Opción	Descripción
host_name	Nombre corto y único del host
alias	Nombre descriptivo del host
address	Una dirección IP o un dominio FQDN del host
parents	Lista de todos los hosts padres del que éste depende; separados por comas
hostgroups	Lista de todos los grupos de host a los que pertenece; separados por comas
check_command	Nombre corto del comando que se utiliza para comprobar si el host está activo, si el comando devuelve "OK", se supone que el host está activo si no se supone apagado
check_interval	Especifica cada cuanto debe realizarse la comprobación; en minutos
retry_interval	Especifica el tiempo de espera en minutos para la siguiente comprobación si el host está activo
max_check_attempts	Especifica cuantas veces se informará de que el host está fuera de línea antes de que Nagios lo considere apagado
check_period	Especifica el nombre del período de tiempo que se debe utilizar para determinar los tiempos durante los cuales deben realizarse pruebas si el host está activo
contacts	Lista de contactos que deben recibir notificaciones relacionadas con los cambios de estado del host; separados por comas; debe especificarse al menos un contacto o grupos de contacto para cada host

Opción	Descripción
notification_interval	Especifica el tiempo en minutos antes de enviar la siguiente notificación de que el host está fuera de línea
notification_period	Especifica el periodo de tiempo durante los que las notificaciones relacionadas con el estado del host deben enviarse
notification_options	Especifica qué tipo de notificaciones de estado del host deben enviarse; separados por comas; debe ser uno o más de los siguientes valores: d: el host está fuera de línea (DOWN) u: el host es inalcanzable (UNREACHABLE) r: recuperación de host (en línea) f: el host cambia de estados continuamente (FLAPPING) s: notificar cuando comienza o termina el tiempo de inactividad programado

Tabla 10 - Directivas de definición de hosts [2]

Opción	Descripción
host_name	Nombre corto y único del host
hostgroup_name	Nombre corto de los grupos de hosts en los que el servicio se está ejecutando; separados por comas
service_description	Descripción del servicio para identificarlo unívocamente en un host
check_command	Nombre corto del comando que se utiliza para comprobar si el host está activo, si el comando devuelve "OK", se supone que el host está activo si no se supone apagado
check_interval	Especifica cada cuanto debe realizarse la comprobación; en minutos
retry_interval	Especifica el tiempo de espera en minutos para la siguiente comprobación si el host está activo
max_check_attempts	Especifica cuantas veces se informará de que el host está fuera de línea antes de que Nagios lo considere apagado
check_period	Especifica el nombre del período de tiempo que se debe utilizar para determinar los tiempos durante los cuales deben realizarse pruebas si el host está activo
contacts	Lista de contactos que deben recibir notificaciones relacionadas con los cambios de estado del host; separados por comas; debe especificarse al menos un contacto o grupos de contacto para cada host
first_notification_delay	Especifica el tiempo en minutos antes de enviar la primera notificación de que el host está fuera de línea
notification_interval	Especifica el tiempo en minutos antes de enviar la siguiente notificación de que el host está fuera de línea
notification_period	Especifica el periodo de tiempo durante los que las notificaciones relacionadas con el estado del host deben enviarse
notification_options	Especifica qué tipo de notificaciones de estado del host deben enviarse; separados por comas; debe ser uno o más de los siguientes valores: w: estado en alerta (WARNING) u: estado desconocido (UNKNOWN) c: estado crítico (CRITICAL) r: recuperación de servico (en línea) f: el host cambia de estados continuamente (FLAPPING) s: notificar cuando comienza o termina el tiempo de inactividad programado

Tabla 11 - Directivas de definición de servicios [2]

Anexo II Ficheros de definición de objetos en Nagios

hosts.cfg

```
define hostgroup{
        hostgroup_name          hosts
        alias                   PCs
}
define host{
        use                     windows-server,host-pnp
        host_name               local1
        alias                   PC local1
        address                 10.0.0.12
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local2
        alias                   PC local2
        address                 192.168.110.2
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local3
        alias                   PC local3
        address                 192.168.110.3
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local4
        alias                   PC local4
        address                 192.168.110.4
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local5
        alias                   PC local5
        address                 192.168.110.5
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local6
        alias                   PC local6
        address                 192.168.110.6
        parents                 cisco2600
        hostgroups              hosts
}
```

```
define host{
        use                     windows-server,host-pnp
        host_name               local7
        alias                   PC local7
        address                 192.168.110.7
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local8
        alias                   PC local8
        address                 192.168.110.8
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local9
        alias                   PC local9
        address                 192.168.110.9
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               local10
        alias                   PC local10
        address                 192.168.110.10
        parents                 cisco2600
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto1
        alias                   PC remoto1
        address                 128.100.100.1
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto2
        alias                   PC remoto2
        address                 128.100.100.2
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto3
        alias                   PC remoto3
        address                 128.100.100.3
        parents                 cisco2500
        hostgroups              hosts
}
```

```
define host{
        use                     windows-server,host-pnp
        host_name               remoto4
        alias                   PC remoto4
        address                 128.100.100.4
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto5
        alias                   PC remoto5
        address                 128.100.100.5
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto6
        alias                   PC remoto6
        address                 128.100.100.6
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto7
        alias                   PC remoto7
        address                 128.100.100.7
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto8
        alias                   PC remoto8
        address                 128.100.100.8
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto9
        alias                   PC remoto9
        address                 128.100.100.9
        parents                 cisco2500
        hostgroups              hosts
}
define host{
        use                     windows-server,host-pnp
        host_name               remoto10
        alias                   PC remoto10
        address                 128.100.100.10
        parents                 cisco2500
        hostgroups              hosts
}
```

routers.cfg

```
define hostgroup{
        hostgroup_name          routers
        alias                   Routers
}
define host{
        use                     generic- switch,host-pnp
        host_name               cisco2600
        alias                   Cisco 2600
        address                 192.168.110.1
        hostgroups              routers
}
define host{
        use                     generic- switch,host-pnp
        host_name               cisco2500
        alias                   Cisco 2500
        address                 192.168.200.34 128.100.100.30
        parents                 cisco2600
        hostgroups              routers
}
define host{
        use                     generic-switch,host-pnp
        host_name               switchSMC24p
        alias                   Switch SMC 24p
        address                 192.168.110.20
        parents                 cisco2600
        hostgroups              switches
}
```

servers.cfg

```
define hostgroup{
        hostgroup_name          servers
        alias                   Servidores
}
define host{
        use                     linux-server,host-pnp
        host_name               Atlas
        alias                   Servidor Atlas
        address                 192.168.110.11 193.144.186.15 atlas.tlmat.unican.es
        parents                 cisco2600
        hostgroups              servers
}
define host{
        use                     windows-server,host-pnp
        host_name               Tele1
        alias                   Servidor Tele1
        address                 192.100.100.20
        parents                 cisco2500
        hostgroups              servers
}
```

ATMNodes.cfg

```
define hostgroup{
        hostgroup_name          ATMnodes
        alias                   Nodos ATM
}
```

```
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Informatica
        alias                   Nodo ATM Informatica
        address                 192.168.0.21 192.168.0.1 192.168.0.17
                                192.168.0.5 192.168.0.9 192.168.0.13
        parents                 cisco2600
        hostgroups              ATMnodes
}
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Nautica
        alias                   Nodo ATM Nautica
        address                 192.168.0.2
        parents                 ATM_Informatica
        hostgroups              ATMnodes
}
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Torrelavega
        alias                   Nodo ATM Torrelavega
        address                 192.168.0.18
        parents                 ATM_Informatica
        hostgroups              ATMnodes
}
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Medicina
        alias                   Nodo ATM Medicina
        address                 192.168.0.6
        parents                 ATM_Informatica
        hostgroups              ATMnodes
}
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Enfermeria
        alias                   Nodo ATM Enfermeria
        address                 192.168.0.10
        parents                 ATM_Informatica
        hostgroups              ATMnodes
}
define host{
        use                     generic-router,host-pnp
        host_name               ATM_Paraninfo
        alias                   Nodo ATM Paraninfo
        address                 192.168.0.14
        parents                 ATM_Informatica
        hostgroups              ATMnodes
}
```

services.cfg

```
define service{
        use                     generic-service,srv-pnp
        hostgroup_name          hosts,ATMnodes,servers,routers
        service_description     PING
        check_command           check_ping!200.0,20%!600.0,60%
        normal_check_interval 5
        retry_check_interval  1
}
```

```
define service{
        use                     generic-service
        hostgroup_name          hosts,servers,routers
        service_description     Uptime
        check_command           check_snmp!-C public -o sysUpTime.0 -m RFC1213-MIB
}
define service{
        use                     generic-service
        hostgroup_name          ATMnodes
        service_description     Uptime
        check_command           check_snmp!-C solomira -o sysUpTime.0 -m RFC1213-MIB
}
define service{
        use                     generic-service
        hosts                   local1,local2,local3,local4,local6,
                                local7,local8,local9,local10
        service_description     Check WIN disk
        check_command           check_win_snmp_disk!$HOSTADDRESS!public!90!100
}
define service{
        use                     generic-service,srv-pnp
        hosts                   local1,local2,local3,local4,local6,
                                local7,local8,local9,local10
        service_description     Check WIN CPU LOAD
        check_command           check_win_cpuload!$HOSTADDRESS!public!90!100
}
define service{
        use                     generic-service,srv-pnp
        hosts                   local1,local2,local3,local4,local6,
                                local7,local8,local9,local10
        service_description     Check WIN MEM
        check_command           check_winmem!$HOSTADDRESS!public!90!100
}
define service{
        use                     generic-service,srv-pnp
        hostgroup_name          servers
        service_description     SSH
        check_command           check_ssh
}
define service{
        use                     trap-service
        hosts                   local1,local2,local3,local4,local6,
                                local7,local8,local9,local10
        service_description     TRAP_Updates
}
define service{
        use                     generic-service,srv-pnp
        hosts                   local1,local2,local3,local4,local6,
                                local7,local8,local9,local10
        service_description     Trafico
        check_command           check_iftraffic64!public!1
}
define service{
        use                     generic-service,srv-pnp
        hosts                   cisco2500,cisco2600
        service_description     Trafico_cisco
        check_command           check_snmp_cisco_traffic!public!1
}
```

```
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 01
        check_command           check_iftraffic64!public!1
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 02
        check_command           check_iftraffic64!public!2
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 03
        check_command           check_iftraffic64!public!3
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 04
        check_command           check_iftraffic64!public!4
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 05
        check_command           check_iftraffic64!public!5
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 06
        check_command           check_iftraffic64!public!6
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 07
        check_command           check_iftraffic64!public!7
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 08
        check_command           check_iftraffic64!public!8
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 09
        check_command           check_iftraffic64!public!9
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 10
        check_command           check_iftraffic64!public!10
}
```

```
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 11
        check_command           check_iftraffic64!public!11
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 15
        check_command           check_iftraffic64!public!15
}
define service{
        use                     generic-service,srv-pnp
        hosts                   switchSMC24p
        service_description     Trafico Interfaz 16
        check_command           check_iftraffic64!public!16
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     Root Partition
        check_command           check_local_disk!20%!10%!/
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     Current Users
        check_command           check_local_users!20!50
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     Total Processes
        check_command           check_local_procs!250!400!RSZDT
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     Current Load
        check_command           check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     Swap Usage
        check_command           check_local_swap!20!10
}
define service{
        use                     local-service,srv-pnp
        host_name               local5
        service_description     HTTP
        check_command           check_http
}
define service{
        use                     generic-service
        service_description     Enlace con Nautica
        host_name               ATM_Informatica
        check_command           check_snmp iproute!solomira
                                !192.168.0.0!255.255.255.252!192.168.0.1
}
```

```
define service{
        use                     generic-service
        service_description     Enlace con Medicina
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.0.4!255.255.255.252!192.168.0.5
}
define service{
        use                     generic-service
        service_description     Enlace con Enfermeria
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.0.8!255.255.255.252!192.168.0.9
}
define service{
        use                     generic-service
        service_description     Enlace con Paraninfo
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.0.12!255.255.255.252!192.168.0.13
}
define service{
        use                     generic-service
        service_description     Enlace con Torrelavega
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.0.16!255.255.255.252!192.168.0.17
}
define service{
        use                     generic-service
        service_description     Enlace con Laboratorio Telematica
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.0.20!255.255.255.252!192.168.0.21
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.0/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.0!255.255.255.224!192.168.1.1
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.32/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.32!255.255.255.224!192.168.0.2
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.64/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.64!255.255.255.224!192.168.0.14
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.96/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.96!255.255.255.224!192.168.0.6
}
```

```
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.128/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.128!255.255.255.224!192.168.0.10
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.0/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.0!255.255.255.224!192.168.2.1
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.32/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.32!255.255.255.224!192.168.0.2
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.64/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.64!255.255.255.224!192.168.0.14
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.96/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.96!255.255.255.224!192.168.0.6
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.128/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.128!255.255.255.224!192.168.0.10
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.0/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.3.0!255.255.255.224!192.168.3.1
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.32/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.3.32!255.255.255.224!192.168.0.2
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.64/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.3.64!255.255.255.224!192.168.0.14
}
```

```
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.96/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.3.96!255.255.255.224!192.168.0.6
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.128/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.4.0/27
        host_name               ATM_Informatica
        check_command           check_snmp_iproute!solomira
                                !192.168.4.0!255.255.255.0!192.168.4.1
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.32/27
        host_name               ATM_Nautica
        check_command           check_snmp_iproute!solomira
                                !192.168.1.32!255.255.255.224!.192.168.1.33
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.32/27
        host_name               ATM_Nautica
        check_command           check_snmp_iproute!solomira
                                !192.168.2.32!255.255.255.224!192.168.2.33
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.32/27
        host_name               ATM_Nautica
        check_command           check_snmp_iproute!solomira
                                !192.168.3.32!255.255.255.224!192.168.3.33
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.64/27
        host_name               ATM_Paraninfo
        check_command           check_snmp_iproute!solomira
                                !192.168.1.64!255.255.255.224!192.168.1.65
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.64/27
        host_name               ATM_Paraninfo
        check_command           check_snmp_iproute!solomira
                                !192.168.2.64!255.255.255.224!192.168.2.65
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.64/27
        host_name               ATM_Paraninfo
        check_command           check_snmp_iproute!solomira
                                !192.168.3.64!255.255.255.224!192.168.3.65
}
```

```
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.96/27
        host_name               ATM_Enfermeria
        check_command           check_snmp_iproute!solomira
                                !192.168.1.96!255.255.255.224!192.168.1.97
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.96/27
        host_name               ATM_Enfermeria
        check_command           check_snmp_iproute!solomira
                                !192.168.2.96!255.255.255.224!192.168.2.97
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.96/27
        host_name               ATM_Enfermeria
        check_command           check_snmp_iproute!solomira
                                !192.168.3.96!255.255.255.224!192.168.3.97
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.1.128/27
        host_name               ATM_Medicina
        check_command           check_snmp_iproute!solomira
                                !192.168.1.128!255.255.255.224!192.168.1.129
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.2.128/27
        host_name               ATM_Medicina
        check_command           check_snmp_iproute!solomira
                                !192.168.2.128!255.255.255.224!192.168.2.129
}
define service{
        use                     generic-service
        service_description     VLAN 192.168.3.128/27
        host_name               ATM_Medicina
        check_command           check_snmp_iproute!solomira
                                !192.168.3.128!255.255.255.224!192.168.3.129
}
define service{
        use                     generic-service
        service_description     Subred 192.168.4.0/27
        host_name               ATM_Torrelavega
        check_command           check_snmp_iproute!solomira
                                !192.168.4.0!255.255.255.0!192.168.4.1
}
```

Anexo III Definición de la MIB

LAB-TELEMATICA.my

```
LAB-TELEMATICA DEFINITIONS ::= BEGIN

      IMPORTS
            OBJECT-TYPE
                  FROM RFC-1212
            TRAP-TYPE
                  FROM RFC-1215
            enterprises
                  FROM RFC1155-SMI;

      labTelematica OBJECT IDENTIFIER ::= { enterprises 90000 }

      hostname OBJECT-TYPE
            SYNTAX OCTET STRING
            ACCESS read-only
            STATUS mandatory
      ::= { labTelematica 1 }

      numActualizaciones OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-only
            STATUS mandatory
      ::= { labTelematica 2 }

      habilitarTraps OBJECT-TYPE
            SYNTAX INTEGER (0..1)
            ACCESS read-write
            STATUS mandatory
      ::= { labTelematica 3 }

      hayActualizaciones TRAP-TYPE
            ENTERPRISE labTelematica
            VARIABLES { hostname, numActualizaciones }
      ::=  0

END
```