Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE CARRERA

TÍTULO DEL TFC: Monitorización de la infraestructura técnica de un Centro de Datos real

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

AUTOR: Fadi Ahmad Taki

DIRECTOR EXTERNO: Miquel López Luque

DIRECTOR: Anna Agustí Torra

FECHA: 27 de Junio de 2011

Título: Monitorización de la infraestructura técnica de un Centro de Datos real
Autor: Fadi Ahmad Taki
Director externo: Miquel López Luque
Director: Anna Agustí Torra
Fecha: 27 de Junio de 2011

Resumen

Los avances tecnológicos han propiciado la aparición de las Tecnologías de la Información y la Comunicación (TIC), que se han integrado en la mayoría de los campos laborales (como el científico, el educativo y el industrial).

La integración de las TIC en el entorno laboral ha permitido que empresas que disponen de un sistema informático y un centro de proceso de datos común puedan tener sedes físicamente separadas pero trabajando conjuntamente.

Sin embargo, el elevado grado de dependencia de la tecnología también supone que muchas empresas vean condicionada la continuidad de sus procesos de negocio al correcto funcionamiento de los equipos de comunicación o de soporte informático.

Mediante la monitorización se puede detectar cuando se produce una incidencia o degradación en el servicio y notificar a la persona responsable, con un mensaje de notificación en la consola, vía correo electrónico o a través de un mensaje SMS al teléfono móvil. Esta notificación permite tomar las acciones correctivas oportunas y evitar en muchos casos problemas mayores.

Las herramientas de monitorización permiten controlar una gran variedad de parámetros y procesos como, por ejemplo, el estado de los servicios de red, parámetros geofísicos (como la temperatura del CPD) o voltajes de entrada y salida de los SAI.

En este trabajo final de carrera se propone y desarrolla un sistema de monitorización del CPD de la Fundación ESADE.

Palabras Clave

Monitorización, control y gestión de alarmas, supervisión del rendimiento de sistemas, protocolo de actuación en caso de caída o saturación, software libre “Nagios”, Round Robin Database, Cacti, Nmap.

Title: Monitoring of the technical infrastructure of a real data center
Author: Fadi Ahmad Taki
External Director: Miquel López Luque
Director: Anna Agustí Torra
Date: 27th of June 2011

Overview

The technological advances have led to the emergence of the information and communication technologies, the called ICT, which were incorporated in most business fields, including the scientific, educational and industrial.

The integration of the ICT´s in the work/job environments allowed the companies that have a computer system and a common data processing center to have headquarters physically separated but working together.

However, the bigger dependence on technology means that many companies have it business continuity depending on the proper functioning of the communication and media equipments and devices.

By monitoring we can detect the services degradations and incidents and notify the responsible person, with a notification message on the console, through an e-mail or a short message to the mobile phone. This notification will allow him to react and in many cases avoid/prevent further problems.

The monitoring tools allow us to control a large variety of parameters and process, for example, the networks services state, geophysical parameters (such as the data center temperature) or the UPS input and output voltages.

In this final thesis we propose and develop the monitoring system of the Data Processing Center of the ESADE Foundation.

Keywords

Monitoring, control and management of alarms, monitoring systems' performance, response protocol in case of a fall or saturation, open software "Nagios", Round Robin Database, Cacti, Nmap.

ÍNDICE

INTRODUCCIÓN

Hoy en día, la dependencia de la tecnología en ámbitos como el educativo, el científico y el industrial, es muy elevada, y un fallo en los sistemas de información o en las comunicaciones puede suponer un riesgo elevado para la continuidad de la actividad.

Para garantizar el funcionamiento de los servicios se han creado Centros de Procesamiento de Datos (CPDs) en los que se concentran físicamente equipos electrónicos (como, por ejemplo, servidores de bases de datos o servidores de correo). La función principal del CPD es garantizar la seguridad y el buen funcionamiento de estos equipos, permitiendo así que la empresa disponga de los recursos necesarios para el procesado de la información que requieren sus operaciones.

En este trabajo final de carrera se propone y desarrolla un sistema de monitorización del CPD de la Fundación ESADE.

ESADE es una institución académica internacional con más de cincuenta años de historia que desarrolla su actividad académica en los Campus de Barcelona, Madrid y Buenos Aires. Cada Campus tiene su propio centro técnico de atención al usuario pero la institución dispone de un Centro de Proceso de Datos situado en el campus de Barcelona, en la facultad de derecho. En el CPD se procesan datos académicos, correos electrónicos e incluso datos financieros mediante equipos informáticos como servidores, equipos de comunicación, etc. Cualquier fallo de alguno de los equipos del CPD puede afectar de manera importante al funcionamiento de la institución. Por lo tanto, es necesario conocer el estado de los equipos para anticipar cualquier incidencia o degradación en el servicio.

Este trabajo final de carrera se ha realizado en el departamento de Tecnologías de la Información y Comunicación (TIC) de Fundación ESADE, en el área del departamento de Arquitectura y Sistemas.

Objetivos

El objetivo del trabajo es la monitorización de la infraestructura técnica del Centro de Procesamiento de Datos (CPD) de ESADE considerando los siguientes ámbitos:

- Equipos informáticos.
- Equipos de comunicaciones.
- Centro de datos: climatización, continuidad eléctrica, etc.

Los aspectos a cubrir son los siguientes:

- Instalación del software libre "*Nagios*" (www.nagios.org), configuración y desarrollo de los módulos necesarios para capturar y mostrar los datos de funcionamiento de los equipos del CPD.

- Preparación de las máquinas con las herramientas específicas para su monitorización. La monitorización deberá incluir elementos fundamentales (CPU, memoria, placa del sistema, temperatura y sistema de ventilación y fuentes de alimentación), elementos de conectividad (control y detección de pérdidas de conectividad) y almacenamiento interno o externo (control y detección del estado de los medios de almacenamiento tanto internos como externos).

- Identificación de los equipos del CPD a monitorizar y definición de su criticidad y del protocolo de actuación y procedimiento a seguir en caso de caída o saturación (utilizando una estrategia común de control y gestión de alarmas).

- Monitorización de los componentes reflejando los parámetros de supervisión y control sobre una consola única centralizada que permita localizar de forma inmediata el origen de las incidencias que puedan producirse sobre cualquiera de los equipos implicados.

- Captura de los datos y procesado de la información para su posterior publicación como gráficos en una interfaz web.

La estructura de la memoria del trabajo es la siguiente. En el capítulo 1 se presenta el trabajo y se identifica el contenido, desde sistema operativo y software hasta herramientas y plugins. En el capítulo 2 se explica porque se ha elegido a Nagios como software para implementar el sistema de monitorización, en qué consiste y el modo de funcionamiento de este software, detallando la instalación y configuración de los software y herramientas empleados a lo largo del trabajo. El capítulo 3 consiste en explicar la monitorización de los equipos y la presentación grafica de su comportamiento en una escala de tiempos mediante el software Cacti y en él también se explica el funcionamiento de la herramienta de autodescubrimiento. En el capítulo 4 se encuentran las posibles futuras implementaciones y cambios que se pueden

aplicar al sistema que se ha creado y las conclusiones obtenidas sobre el trabajo realizado.

CAPÍTULO 1. Presentación y contenido

En este trabajo se propone un sistema de monitorización utilizando el programa Nagios en Linux. Para la gestión y presentación de resultados se utilizan las herramientas RRDTool (Round Robin Database Tool), N2RRD (Nagios to Round Robin Database) y Cacti. Finalmente, para el autodescubrimiento de equipos de la red se utiliza el software Nmap y el plugin Nmap2Nagios. En este capítulo se realiza una breve explicación sobre todos estos programas.

1.1. Linux

Linux hace referencia a la familia de sistemas operativos UNIX que usan el núcleo o "*Kernel*" Linux. Se trata de un sistema operativo de libre distribución (es decir, no hace falta adquirir ninguna licencia para poder usarlo y además el sistema viene acompañado del código fuente).

El sistema, formado por el núcleo y un gran número de aplicaciones y librerías que hacen posible su utilización, ha sido diseñado, actualizado y mejorado por multitud de programadores alrededor del mundo.

Linux puede ser instalado en una amplia variedad de equipos informáticos, desde teléfonos móviles, ordenadores y consolas de videojuegos, hasta supercomputadores. Es el más popular entre los sistemas operativos para servidores y actualmente se utiliza en algunos de los supercomputadores más rápidos del mundo.

También es uno de los principales sistemas operativos para equipos de sobremesa, gracias a la gran seguridad y estabilidad que ofrece, a su velocidad, y a la ausencia de problemas de fragmentación.

Día a día, son más y mejores los programas y las aplicaciones para este sistema, y la presencia de Linux en empresas aumenta cada vez más gracias a la simplificación de su manejo y a la excelente relación calidad-precio que ofrece.

1.2. Nagios

N.A.G.I.O.S.[1] es un acrónimo recursivo: "*Nagios Ain't Gonna Insist On Sainthood*". Es una referencia al nombre original, "*Netsaint*", que tuvo que cambiarse por su supuesta similitud con el nombre comercial "*Agios*" que significa "*santo*" en griego.

[1] Pagina Web Oficial de Nagios: http://www.nagios.org/

Nagios es un sistema de monitorización de código abierto para la supervisión de equipos y servicios de la red. Permite a una empresa identificar problemas en su infraestructura antes de que afecten de manera drástica las funciones del negocio.

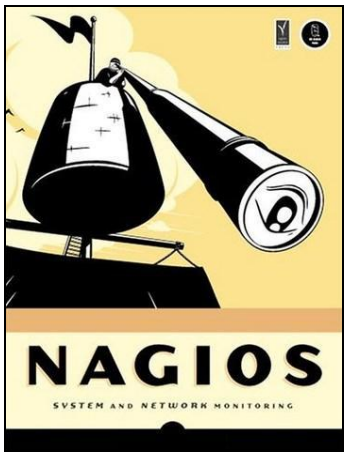


Fig. 1.1 Logo de N.A.G.I.O.S.

1.3. RRDTool

RRDtool[2] es el acrónimo de "*Round Robin Database tool*". Se trata de una herramienta para generar gráficas y obtener datos estadísticos directamente de una base de datos.

El método Round Robin permite explorar una lista ordenadamente y de forma circular (volviendo al primer elemento después de analizar el último).

Fig. 1.2 Logo de Round Robin

En este trabajo, la herramienta RRDtool se utiliza para generar gráficas en tiempo real accesibles desde una interfaz web, permitiendo así el seguimiento visual del estado de los equipos monitorizados.

[2] Pagina Web de RRDTool: http://www.mrtg.org/rrdtool/

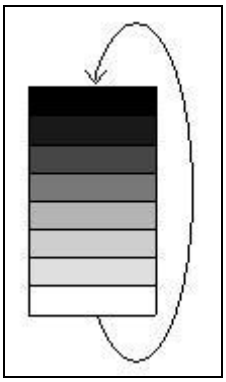


Fig. 1.3 Funcionamiento de la base de datos Round Robin

En este trabajo, la herramienta RRDtool se utiliza para generar gráficas en tiempo real accesibles desde una interfaz web, permitiendo así el seguimiento visual del estado de los equipos monitorizados.

1.4. N2RRD

N2RRD[3] es un acrónimo de "*Nagios to Round Robin Database*". Se trata de una herramienta que almacena datos generados por los plugins de Nagios en bases de datos Round Robin. N2RRD incluye la herramienta de visualización rrd2graph, aunque los archivos que genera se pueden visualizar utilizando cualquier otra herramienta de visualización. En este trabajo, la herramienta utilizada es Cacti.

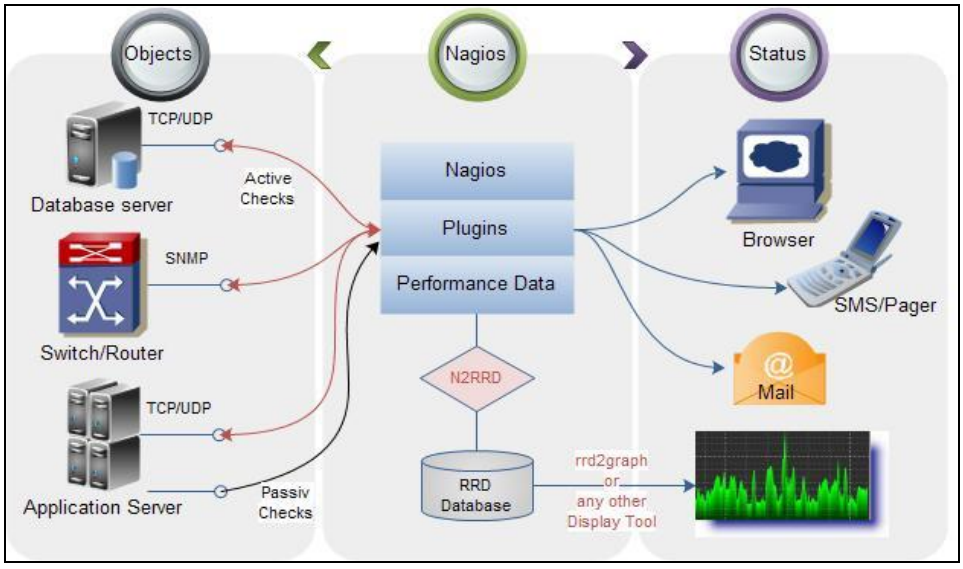


Fig. 1.4[4] Funcionamiento de N2RRD

[3] Pagina Web de N2RRD http://n2rrd.diglinks.com/cgi-bin/trac.fcgi
[4] Fig. 1.4: (Autor: Badri Pillai - http://n2rrd.diglinks.com/cgi-bin/trac.fcgi)

1.5. Cacti

Cacti[5] es una herramienta gráfica para la monitorización de redes usando RRDtool y SNMP[6], desarrollada en php bajo licencia GNU y disponible en sistemas Linux y Windows.

Cacti posee una completa lista de utilidades típicas de una herramienta de este tipo: monitor de alertas, gráficas de estado para CPU, memoria, carga, interfaces de red, etc. Además, tiene la opción de "*Weathermap*" que permite monitorizar mapas con toda la infraestructura de la red pudiendo así verificar el estado de los enlaces entre los diferentes dispositivos.

Tiene una interfaz de usuario fácil de usar, que resulta conveniente para instalaciones del tamaño de una red de área local (LAN), así como también para redes complejas con cientos de dispositivos.

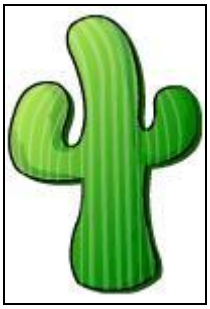


Fig. 1.5 Logo de Cacti

1.6. Nmap

Nmap[7] ("*Network Mapper*" o mapeador de redes) es una herramienta de código abierto para la exploración de redes. Utiliza paquetes IP para analizar rápidamente grandes redes (y también equipos individuales) y determinar qué equipos están disponibles, mostrar qué servicios operativos ejecutan, qué servicios ofrecen y otras características.

1.7. Nmap2Nagios

Nmap2Nagios es un script escrito en lenguaje Perl para convertir los archivos de salida de Nmap (en formato XML) a archivos de configuración de objetos de Nagios (con extensión *.cfg). Se puede adaptar el archivo de configuración de nmap2nagios para automatizar la generación de los archivos de configuración de Nagios de modo que se muestren las características deseadas del equipo descubierto (sistema operativo que utiliza, servicios que ofrece, etc.).

[5] Pagina Web Oficial de Cacti: http://www.cacti.net
[6] SNMP (Simple Network Management Protocol) es un protocolo que facilita el intercambio de información de administración entre dispositivos de red.
[7] Pagina Web Oficial de Nmap: http://nmap.org/

CAPÍTULO 2. Preparación del entorno

2.1. Nagios

2.1.1. Sistemas y software de Monitorización

Durante la fase de planificación de este trabajo se evaluaron distintos sistemas y programas de monitorización de redes: Nagios, Pandora, Zabbix y Zenoss. En este apartado se mencionan las principales características de cada uno de ellos.

De entre todas las opciones, se eligió Nagios por ser el más completo, el que cumple todos los requisitos deseados y el que cubre todas las necesidades de monitorización de la red. Además, Nagios es actualmente el software de monitorización de código libre más utilizado y dispone de gran cantidad de documentación e información de soporte.

2.1.1.1. Nagios

El software Nagios monitoriza la infraestructura de la red detectando incidencias o degradaciones en el servicio y notificándolas a la persona responsable (con un mensaje en la consola, vía correo electrónico o a través de un SMS) para poder resolverlas antes de que afecten al funcionamiento de la empresa y/o a los clientes o usuarios finales.

Las funciones principales de Nagios son:

- Monitorizar los recursos de los equipos: carga de CPU, uso de los discos y de la memoria, logs del sistema, etc.
- Dar al administrador de la red la opción de diseñar plugins que le permitan realizar sus propias comprobaciones de los servicios en función de sus necesidades.
- Comprobar el estado de los servicios de un sistema.
- Notificar al administrador de la red los problemas detectados en servicios o equipos con un mensaje de notificación en la consola, vía correo electrónico o a través de un mensaje SMS al teléfono móvil.
- Permitir la rotación automática del archivo de log.

- Visualizar el estado de la red en tiempo real a través de una interfaz web, generando informes y gráficas del comportamiento de los sistemas y servicio monitorizados.

- Visualizar el listado de notificaciones enviadas, del historial de problemas y de los archivos de registros.

Para el desarrollo de este trabajo se ha elegido este programa porque, comparado con otros, ofrece mejores cualidades técnicas para la monitorización de la red.

2.1.1.2. Pandora

Pandora[8] es una herramienta de software libre que permite monitorizar, utilizando un navegador, el rendimiento y estado de parámetros de diferentes sistemas operativos, servidores, aplicaciones y sistemas hardware tales como bases de datos, servidores web, etc.

Las principales funcionalidades que ofrece son:

- Monitorización multiplataforma. Permite monitorizar equipos con diferentes sistemas operativos, como Solaris, Windows, GNU/Linux, etc.

- Monitorización de servicios, aplicaciones, puertos, procesos, archivos logs.

- Monitorización remota.

- Gestión de alertas y notificaciones.

- Gestión web.

- Alta capacidad de procesamiento.

2.1.1.3. Zabbix

Zabbix[9] es un sistema de código abierto para la monitorización de red. Ha sido diseñado para supervisar y para seguir el estado de varios servicios de red, servidores y elementos físicos de la red.

Este sistema cuenta con una interfaz de administración vía web y envía avisos por correo electrónico cuando hay algún problema en alguno de los equipos o en algún servicio.

[8] Pagina Web Oficial de Pandora: http://pandorafms.org/
[9] Pagina Web Oficial de Zabbix: http://www.zabbix.com/

Zabbix concentra toda la configuración de forma centralizada en un servidor y permite tener toda la información de monitorización en tiempo real. También permite el descubrimiento de nodos mediante agentes SNMP, soportando avisos y alarmas (traps) SNMP en las tres versiones del protocolo.

2.1.1.4. Zenoss

Zenoss[10] es una plataforma de código abierto para la gestión de la red, los servicios y los equipos que la forman. Las funcionalidades más destacadas de esta aplicación son:

- Visualización de la disponibilidad de los equipos utilizando SNMP.
- Monitorización de los servicios de red.
- Monitorización de los recursos y del rendimiento de los equipos de la red (carga de la CPU, utilización del disco, etc.)
- Gestión de alertas y notificación al administrador de la red de los problemas detectados.
- Interfaz web para la monitorización de los sistemas.
- Soporte para el formato de plugins de Nagios.

2.1.2. Estructura del sistema

La estructura interna de Nagios se compone de cuatro módulos:

- Núcleo o *kernel*. El *kernel* contiene el software necesario para realizar la monitorización, el control de los procesos y la gestión de los servicios y de las máquinas de la red. Utiliza diversos componentes ya incluidos con la aplicación y también permite utilizar componentes realizados por terceros.
- Extensiones o *plugins*. Los *plugins* son secuencias de comandos o scripts que se ejecutan para comprobar el estado de una máquina o servicio.
- Interfaz web. La web reside en el mismo servidor Nagios y utiliza el software web Apache 2 para la publicación. Está programada en HTML y CSS y utiliza scripts CGI. A través de la interfaz web se puede observar el resultado de la monitorización de los equipos y de los servicios, permitiendo al administrador de la infraestructura tener un control visual y gráfico de su comportamiento.

[10] Pagina Web Oficial de Zenoss: http://www.zenoss.com/

- Bases de datos. La información de configuración y la información del histórico se guardan en archivos de texto permitiendo su análisis posterior.

2.1.3. Funcionamiento

Nagios es un software muy potente y flexible, pero conseguir que funcione correctamente puede resultar complejo. Para poder realizar la instalación y la correcta configuración del programa, es necesario conocer su funcionamiento interno.

2.1.3.1. Plugins

Nagios no incluye ningún mecanismo interno para comprobar el estado de los equipos o servicios. Se basa en programas externos llamados *plugins* y utiliza el resultado de la ejecución de cada *plugin* para determinar el estado de un equipo o servicio y tomar las medidas necesarias, por ejemplo notificar al administrador de la red de la situación.

Hay una distribución oficial de *plugins* para Nagios para controlar los recursos básicos, pero también se pueden diseñar y escribir *plugins* según las necesidades de gestión de cada red. Pueden ser escritos en lenguajes de programación Bash, C++, Perl, Ruby, Python, PHP, C#, etc.

Hay que mencionar que Nagios no entiende los detalles de lo qué se está monitorizando, sólo registra los cambios en el estado de los recursos. Los *plugins* son los que saben realmente qué es lo que se está monitorizando y toman las acciones necesarias (como activar los procesos para solucionar un evento o enviar notificaciones). El texto que generan antes de que acabe su ejecución es el resultado de la acción principal de cada *plugin* y es el estado de la operación. Este estado se muestra en la interfaz web de Nagios y también se guarda en un archivo de registros o Log.

2.1.3.2. Estado de equipos y servicios

Nagios verifica primero el estado de los servicios de un equipo. Si un servicio muestra un estado erróneo, Nagios verifica el estado del equipo mediante el comando ping.

- Si el equipo responde al ping, Nagios interpreta que es el servicio el que no funciona correctamente y procede a realizar las notificaciones correspondientes y/o a tomar las medidas pertinentes.

- Si el equipo no responde al ping, Nagios interpreta que hay un problema con el equipo, cancela las notificaciones sobre el estado de sus servicios (ya que si un equipo no responde, no se pueden monitorizar sus servicios) y notifica el estado del equipo al administrador de la red.

2.1.3.3. Tipo de estados

El estado real de los servicios y de los equipos monitorizados se determina por dos componentes:

- El estado del servicio o del equipo. Se trata del valor real retornado por los *plugins* en la consulta que realizan a los equipos o servicios. Puede ser:
  - OK
  - WARNING
  - CRITICAL
  - UP: Host o servicio alcanzables.
  - DOWN: Host o servicio caído.
  - UNKNOWN: estado desconocido.
- El tipo de estado virtual en el que se encuentra el servicio o el equipo.

Los estados son cruciales para la lógica de monitorización y también se utilizan para determinar cuándo se ejecutan los controladores de eventos y cuándo se envían las notificaciones.

Hay dos tipos de estados en Nagios: SOFT (ligeros) y HARD (graves).

Estados SOFT (Ligeros)

Ocurren en las siguientes situaciones:

- Cuando el resultado de la comprobación de un equipo o servicio es un estado no-OK o no-UP y no se ha comprobado el servicio el número máximo de veces especificado por la variable "*max_check_attempts*" fijada en la definición del equipo o servicio. Esta situación se denomina error SOFT o ligero.
- Cuando un equipo o servicio se recupera de un error ligero.

Cuando un equipo o servicio experimenta un cambio de estado ligero Nagios activa los siguientes procesos:

- Se registra el error y la recuperación del servicio en un archivo de registros (Log). Este servicio debe habilitarse en el archivo de configuración principal.
- Se ejecutan aquellos comandos previamente configurados para intentar solucionar el problema de forma proactiva antes que el estado se convierta en HARD (grave).

Estados HARD (Graves)

Ocurren en las siguientes situaciones:

- Cuando el resultado de la comprobación de un equipo o servicio es un estado no-OK o no-UP y se ha comprobado el servicio el número de veces especificado por la variable "*max_check_attempts*" fijada en la definición del equipo o servicio. Esta situación se denomina error HARD o grave.

- Cuando un equipo o servicio pasa de un estado de error HARD a otro estado de error. Por ejemplo al pasar de un aviso (WARNING) a un estado crítico (CRITICAL).

- Cuando el resultado de comprobación de un equipo o servicio es un estado no-OK y el correspondiente equipo está caído/inactivo o inalcanzable.

- Cuando un equipo o servicio se recupera de un error grave.

Cuando un equipo o servicio experimenta un cambio de estado grave Nagios activa los siguientes procesos:

- Se registra el error y la recuperación en un archivo de registros (Log).

- Se ejecutan los comandos configurados para tratar el problema.

- Se notifican los problemas en el equipo o en el servicio y/o su recuperación al contacto definido en la configuración.

2.1.3.4. Interrupciones en la red

Nagios tiene la capacidad de determinar si los equipos que están siendo monitorizados están en estado caído/inactivo (DOWN) o si están en estado inalcanzable (UNREACHABLE). Hay una gran diferencia entre estos dos estados y conocerlos permite localizar con precisión los problemas de la red.

Para que Nagios pueda distinguir entre los estados DOWN y UNREACHABLE, se tiene que definir la conexión entre los equipos desde el punto de vista del servidor Nagios y para ello se define la relación padre/hijo entre los equipos. Así, cuando Nagios descubre que hay un equipo padre en estado caído/inactivo (DOWN) considera que todos sus equipos hijos están inalcanzables (UNREACHABLE) porque el equipo padre está bloqueando el camino hacia ellos.

2.1.3.5. Notificaciones

La decisión de enviar notificaciones se define en la lógica de comprobación de los equipos y servicios. Se notifica:

- Cuando la respuesta a la comprobación es un estado de error HARD.

- Cuando el equipo o servicio se mantiene en un estado no-OK, pasado el tiempo especificado en la variable "*notification_interval*" desde el envío de la última notificación.

Nagios envía las notificaciones a todos los miembros del grupo de contactos especificados en la variable "*contact_group*" que se encuentra en la definición de cada equipo y servicio. Si un contacto se encuentra en más de un grupo, sólo se le envía la notificación una vez, para evitar duplicaciones.

Para evitar el envío de muchas notificaciones innecesarias se utiliza filtros:

- Filtro de programa. Es el primer filtro que hay que pasar y consiste en comprobar que el servicio de notificación esta activo verificando el valor de la directiva "*enable_notifications*" del archivo de configuración principal. Se puede desactivar individualmente para cualquier equipo o servicio.

- Filtros de equipo y servicio. Se componen de un conjunto de filtros:

    - El primer filtro consiste en verificar si el equipo o el servicio se encuentra en el periodo de apagado programado. Nagios permite programar un periodo en el que se puede dejar de monitorizar un equipo o servicio para mantenimiento o cualquier otra acción. Si el equipo o el servicio se encuentra en este periodo, no se notifica a nadie. En caso contrario, se pasa al segundo filtro.

    - El segundo filtro consiste en comprobar si el equipo o servicio se encuentra en un estado inestable. Si es así, no se notifica a ningún contacto. En caso contrario, se pasa al siguiente filtro.

    - El tercer filtro son las opciones de notificación definidas para cada equipo o servicio y que determinan si debe notificarse una respuesta de WARNING a una comprobación o si debe notificarse cuando el equipo o servicio está caído/inactivo o inalcanzable.

    - El cuarto filtro consiste en verificar el periodo de notificación (*notification_period*). Si el periodo no se encuentra en un intervalo de tiempo válido, no se envían notificaciones. Si este filtro no se pasa, Nagios reprograma la notificación para un intervalo de tiempo válido del periodo de notificación.

    - El último filtro se condiciona a: 1) que la notificación ya haya sido enviada y 2) que el equipo o servicio se hayan mantenido en un

estado no-OK después del envío de la última notificación. Si no se cumplen las dos condiciones, se envía la notificación. Si se cumplen las dos condiciones, se verifica que el tiempo que ha pasado desde el envío de la última notificación no haya excedido el intervalo de notificación. Si es así, no se envían notificaciones. En caso contrario, se avanza hacia los filtros de contactos.

- Filtros de contacto. Cada contacto tiene sus propios filtros por los cuales las notificaciones han de pasar antes de que el contacto las reciba. Se especifican dos tipos de filtros que coinciden con el tercer y cuarto filtro del conjunto de filtros de equipo y servicio.

Nagios es capaz de enviar notificaciones por distintas vías (correo electrónico, SMS, etc.). Por defecto, las notificaciones se envían por correo electrónico pero puede modificarse el método instalando los *plugins* necesarios y configurando los comandos de notificación en los archivos de configuración.

2.1.4. Configuración

Con la instalación de Nagios, se instalan unos archivos de configuración de ejemplo en el directorio "*/usr/local/nagios/etc/*". Estos archivos de configuración se pueden utilizar directamente eliminando la palabra "*sample*" con algunas modificaciones para adaptar Nagios a las necesidades particulares de la red, los equipos y los servicios que se desea monitorizar en cada caso.

Hay cuatro tipos de archivos de configuración:

- Archivo de configuración principal *(Main Config File)*
- Archivo de recursos *(Resource File)*
- Archivo de definición de objetos *(Object Definition File)*
- Archivo de configuración de CGI *(CGI Config File)*

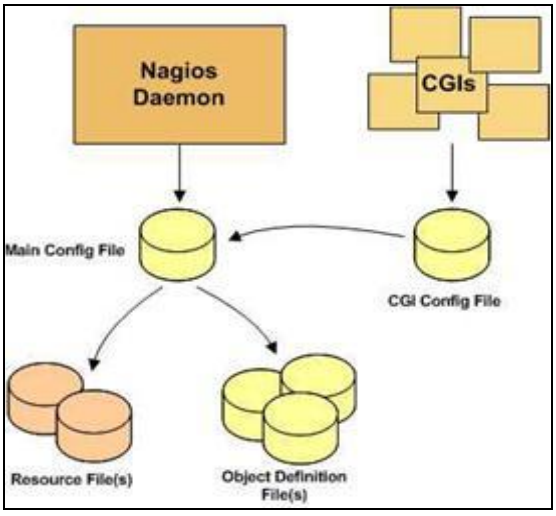


Fig. 2.1 Archivos de configuración de Nagios

A continuación se describen las funciones principales de cada archivo y se especifica la ruta del CD adjunto a esta memoria donde se pueden consultar los archivos modificados para el caso particular de este trabajo junto con una explicación detallada de la función de cada comando.

2.1.4.1. Archivo de configuración principal

El archivo de configuración principal es "*nagios.cfg*" y contiene las directivas que definen el funcionamiento de Nagios. En este archivo se define el "*path*" de los otros archivos de configuración (archivo de log, archivo de recursos, archivos de definición de objetos, etc.) También incluye la definición de otros parámetros como la fecha y la hora, el número máximo de procesos en paralelo que se pueden generar, el nombre del usuario administrador de Nagios y del grupo de usuarios al que pertenece, etc.

```
# LOG FILE
# This is the main log file where service and host events are logged #
for historical purposes.  This should be the first option specified #
in the config file.

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

```
# NAGIOS USER
# This determines the effective user that Nagios should run as.
# You can either supply a username or a UID.

nagios_user=nagios
```

```
# NAGIOS GROUP
# This determines the effective group that Nagios should run as.
# You can either supply a group name or a GID.

nagios_group=nagios
```

- Se puede acceder a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Configuration files\nagios.cfg*".

2.1.4.2. Archivo de recursos

El archivo de recursos define las macros, es decir, variables internas que Nagios utiliza durante la ejecución. Un ejemplo de una macro fundamental es la macro "*$USER1$*" que contiene la ruta hacia el directorio donde se encuentran los *plugins*. De este modo no es necesario escribir la ruta entera en los servicios, basta con invocar la macro y el nombre del *plugin* a ejecutar.

```
# Sets $USER1$ to be the path to the plugins
 $USER1$=/usr/local/nagios/libexec
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Configuration files\resource.cfg*".

2.1.4.3. Definición y configuración de objetos

Un objeto es todo aquél elemento involucrado en el proceso de monitorización y notificación. Los objetos se definen utilizando plantillas de formato flexible en los ficheros especificados por la directiva "*cfg_file*" y/o en los directorios especificados por la directiva "*cfg_dir*" en el archivo de configuración principal.

Los diferentes tipos de objetos que se pueden definir son:

- *Plantillas (templates):* Definen conjuntos de propiedades que pueden ser heredadas por otros objetos. Se definen en el archivo "*templates.cfg*" y pueden ser contactos, equipos y servicios. Se definen los periodos e intervalos de notificación, algunos comandos y otros parámetros.

  El objetivo de estas plantillas es hacer más fácil la configuración y modificación de las opciones de los objetos, ya que, modificando los parámetros en este archivo, queda modificado en los objetos que los heredan.

  - Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Configuration files\nagios.cfg*".

- *Hosts y grupos de hosts.* Los hosts son los equipos monitorizados. Por defecto, se definen en el archivo "*hosts.cfg*". Sin embargo, en este TFC se ha generado un archivo para cada equipo monitorizado.

  La definición de grupos de hosts permite agrupar un conjunto de equipos (por ejemplo, un grupo de conmutadores) de modo que sean mostrados como grupo en la interfaz gráfica.

- *Servicios y grupos de servicios.* En el archivo "*services.cfg*" se definen los servicios que se desea monitorizar de cada equipo (desde atributos del equipo, como la carga de la CPU o la ocupación en disco, hasta procesos que corren en el equipo, como FTP, HTTP, etc.

  Los servicios se pueden encontrar en un estado u otro y vienen presentados por defecto en la interfaz web bajo diferentes colores.

  - OK (Servicio en correcto funcionamiento): Color verde
  - WARNINNG (Alerta sobre un funcionamiento que no es el deseado): Color amarillo.
  - UNKNOWN (Estado desconocido): Color naranja.
  - CRITICAL (Alerta sobre un funcionamiento crítico del servicio): Color rojo.

La creación de grupos de servicios facilita la visualización de los servicios relacionados (pertenecientes a un mismo grupo) en la interfaz web.

- *Contactos y grupos de contactos.* Un contacto es la persona implicada en el proceso de notificación y es especificado en el archivo "*contactos.cfg*". Normalmente es el administrador de la red o la persona que debe ser notificada en caso producirse una incidencia en la monitorización de los equipos y servicios. Junto a los contactos se indican las circunstancias bajo las cuales se produce una notificación cuando hay un cambio de estado.

  En este fichero se indica también el método de notificación al/los contacto/s definido/s (que puede ser mediante correo electrónico, mensaje cortos a dispositivos móviles, etc.)

  El grupo de contactos, al igual que el grupo de equipos o servicios, sirve para agrupar un conjunto de contactos haciendo más fácil la definición de las personas que han de ser notificadas cuando se produzca algún error en un equipo o servicio.

  - Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects \contacts.cfg*".

- *Comandos.* Se definen en el archivo "*commands.cfg*" y son los comandos que indican a Nagios qué programas, scripts, etc. se deben ejecutar para monitorizar los equipos y servicios.

  - Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects \commands.cfg*".

- *Periodos de tiempo:* Se definen en el archivo "*timeperiods.cfg*" y se utilizan para controlar cuando deben efectuarse la monitorización de los equipos y servicios y cuando se deben notificar las incidencias a los contactos.

  - Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects \timeperiods.cfg*".

2.1.4.4. Archivo de configuración de CGI

El archivo "*cgi.cfg*" contiene la configuración de las directivas que afectan al funcionamiento de las CGIs[11] que Nagios utiliza para implementar las peticiones por parte del usuario hacia el servidor web.

[11] CGI (*Common Gateway Interface)*: Es un mecanismo de comunicación entre el servidor web y una aplicación externa. CGI especifica un estándar para transferir datos entre el cliente y el programa. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Un ejemplo es la generación de listas de servicios y equipos que se encuentran en la misma condición (por ejemplo, en estado crítico). También contiene información relacionada con la ruta hacia el archivo de configuración principal, hacia la localización de los archivos HTML de Nagios y hacia otros archivos relevantes.

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Configuration files \cgi.cfg*".

2.1.4.5. Verificación de la configuración y arranque de Nagios

Siempre que se modifican los archivos de configuración, hay que verificar que los archivos modificados no contienen errores antes de iniciar o reiniciar Nagios. Para ello, se utiliza el comando:

```
[root@nagios ~]# /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
```

Si hay errores, Nagios indica el nombre del fichero y la línea dónde se ha detectado el problema.

Para arrancar, parar o reiniciar el servicio se puede utilizar cualquiera de los dos comandos siguientes:

```
[root@nagios ~]# service nagios start /stop/restart
[root@nagios ~]# /etc/rc.d/init.d/nagios start /stop/restart
```

2.2. RRDTool

La herramienta RRDTool permite tratar datos temporales y series de datos (temperaturas, cargas del procesador, etc.)

Para instalarla, se pueden descargar los paquetes de la web "*http://packages.sw.be/rrdtool/*" e instalarlos mediante los comandos siguientes:

```
[root@nagios downloads]# rpm –Uvh rrdtool-1.4.2-1.el5.rf.i386.rpm
[root@nagios downloads]# rpm –Uvh rrdtool-devel-1.4.2-
1.el5.rf.i386.rpm
```

2.3. N2RRD

RRDTool necesita ficheros de entrada con extensión "**.rrd*" de los que leer los datos. La herramienta N2RRD (Nagios to Round Robin Database) permite

convertir los datos generados por los plugins de Nagios a archivos con extensión "**.rrd*".

Para instalar N2RRD se deben seguir los pasos siguientes:

1. Descargar el paquete de instalación y descomprimirlo:

```
http://n2rrd.diglinks.com/download/n2rrd-1.4.2.tar.gz
[root@nagios downloads]# tar xzf n2rrd-1.4.2.tar.gz
```

2. Crear el directorio "*/etc/n2rrd*" y copiar todos los archivos obtenidos al descomprimir el paquete de instalación.

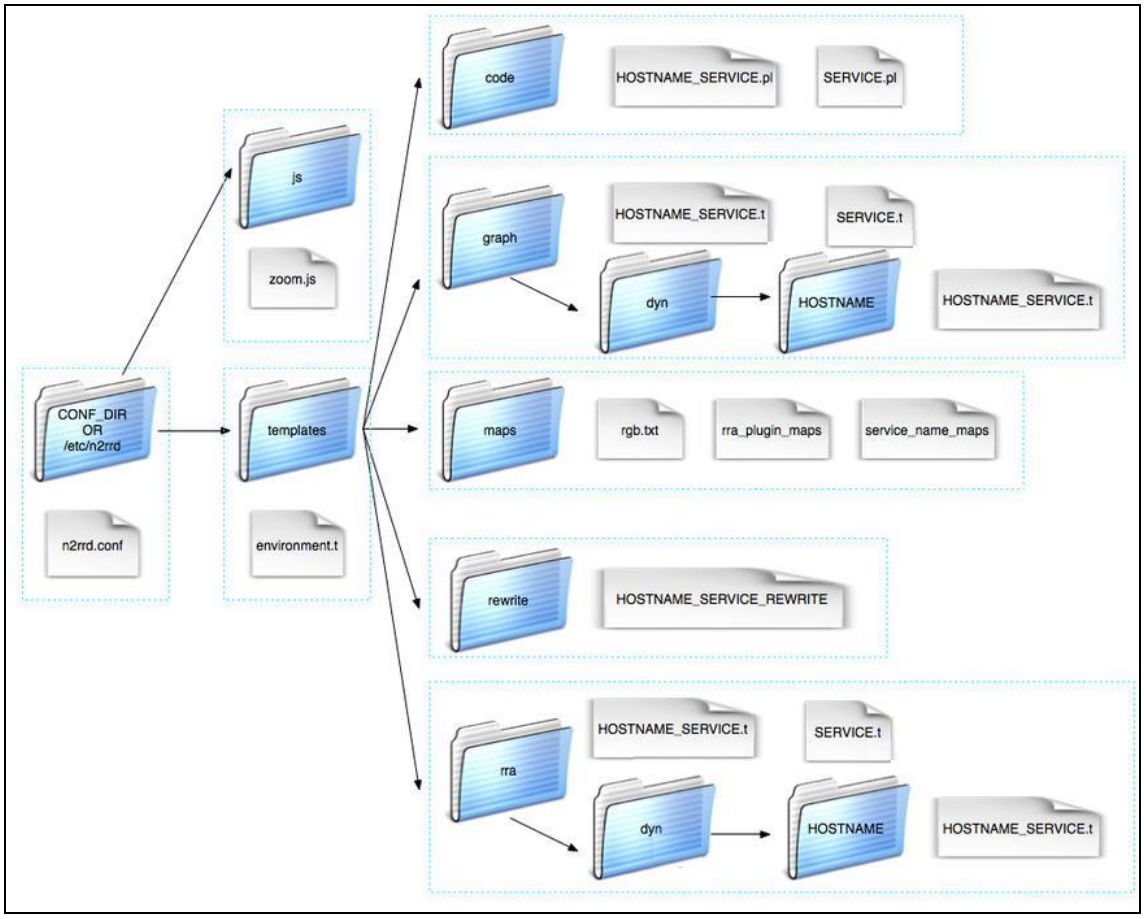


Fig. 2.2 Distribución de los archivos en el directorio de configuración de N2RRD

3. Crear el directorio "*/etc/n2rrd*" y copiar todos los archivos obtenidos al descomprimir el paquete de instalación.

   - Cambiar el nombre del fichero "*/etc/n2rrd/dist-n2rrd.conf*" por el de "*n2rrd.conf*".

- Cambiar el nombre del fichero "*/etc/n2rrdtemplates/maps/dist-rra_plugin_maps*" por el de "*rra_plugin_maps*". Además, hay que verificar que en este archivo hay la definición correcta del tipo de datos de origen (DST):

```
check_ping=rta:GAUGE,pl:GAUGE
check_netstat=active:DERIVE,passive:DERIVE,failed:DERIVE,resets:DERIVE
,established:GAUGE
```

- Cambiar el nombre del archivo "*/etc/n2rrdtemplates/maps/dist-rgb.txt*" por el de "*rgb.txt*". En este archivo es dónde se definen los colores utilizados para la representación gráfica.

- Además de los dos archivos mencionados, hay que renombrar más archivos. Esto se puede hacer ejecutando el comando.

```
[root@nagios tempaltes]# for fdist in dist-*; do fnew=`echo $fdist |
sed 's/dist-//'`; mv $fdist $fnew; done
```

4. Instalar el programa:

```
[root@nagios ~]# cd /etc/n2rrd
[root@nagios n2rrd]# ./install.sh
```

2.3.1. Configuración y adaptación de Nagios

Hay que configurar Nagios para integrarlo con la herramienta N2RRD.

1. Acceder al archivo de configuración de Nagios, "*/usr/local/nagios/etc/nagios.cfg*" y fijar el valor de "*process_performance_data*" a "*1*". Esta variable determina que Nagios debe procesar los datos de rendimiento devueltos de la comprobación (*check*) de los equipos con el comando "*host_perfdata_command*" y con el comando "*service_perfdata_command*" en el caso de los servicios.

```
process_performance_data=1
host_perfdata_command=process-host-perfdata
service_perfdata_command=process-service-perfdata
```

2. Definir los comandos "*process-host-perfdata*" y "*process-service-perfdata*" en el archivo "*commands.cfg*":

```
#'process-host-perfdata' command definition
define command{
     command_name    process-host-perfdata
     command_line /usr/local/bin/n2rrd.pl -d -D "HOST" -N
"/usr/local/nagios/var/status.dat" -C '$HOSTCHECKCOMMAND$' -c
/etc/n2rrd/n2rrd.conf -T $LASTHOSTCHECK$ -H $HOSTNAME$ -s "check_icmp"
-o "$HOSTPERFDATA$"
    }
```

```
# 'process-service-perfdata' command definition
define command{
     command_name    process-service-perfdata
     command_line    /usr/local/bin/n2rrd.pl -d -N
"/usr/local/nagios/var/status.dat" -C '$SERVICECHECKCOMMAND$' -c
/etc/n2rrd/n2rrd.conf -T $LASTSERVICECHECK$ -H $HOSTNAME$ -s
"$SERVICEDESC$" -o "$SERVICEPERFDATA$"
     }
```

En la línea de comandos "*command_line*" se especifican los argumentos que Nagios debe ejecutar:

- -c <path>/config-file-name. Ruta al fichero de configuración n2rrd.conf
- -C check_command. Comando usado para hacer la comprobación. Los valores posibles son $HOSTCHECKCOMMAND$ | $SERVICECHECKCOMMAND$"
- -d debug. Modo *debug* del comando.
- -N <path>/status-filenagios $STATUSDATAFILE$. Ruta al fichero de estado de Nagios. Permite acceder la información que Nagios está generando.
- -T time $LASTHOSTCHECK$. Fecha y hora actuales.
- -H hostname $HOSTNAME$. Nombre del host.
- -G hostgroup $HOSTGROUPNAME$. Grupo de Nagios donde está el equipo.
- -s servicename "$SERVICEDESC$". Nombre del servicio.
- -o service_check_output "$SERVICEPERFDATA$". Datos de salida.

Para los comandos de comprobación definidos para cada equipo y servicio, se crea dinámicamente una plantilla de archivo round robin (RRA – Round Robin Archive). Cada RRA contiene un número limitado de datos consolidados (selección lógica de las muestras recogidas). Cada base de datos contiene 3 archivos RRA: el primero almacena los valores obtenidos durante los 5 primeros minutos, el segundo almacena los valores obtenidos durante 1 hora y el último archivo almacena los datos de 1 día.

El proceso de almacenamiento de los datos es el siguiente: Se recogen los valores de los últimos 5 minutos (a razón de 1 valor cada minuto). Se calcula el promedio de los 5 valores y se almacena el resultado en el fichero siguiente. Y así sucesivamente. La figura siguiente muestra el proceso de forma gráfica:

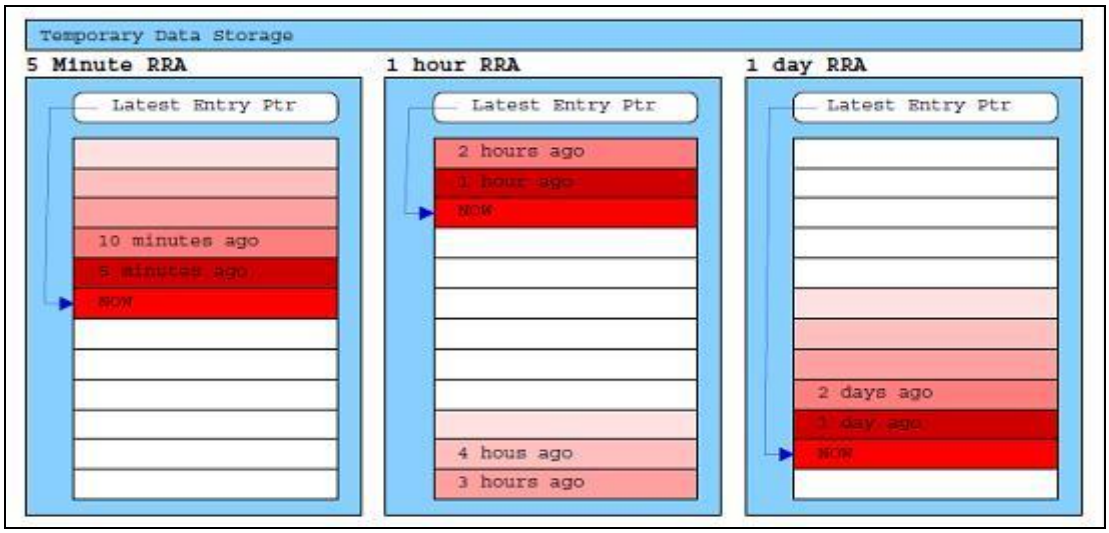


Fig. 2.3 Proceso de almacenaje de datos en los archivos RRA

Se puede ver un ejemplo de estos archivos tipo RRA:

- Accediendo al archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Archivo rra\ localhost_ping.env*".

Después de terminar las modificaciones y las adaptaciones necesarias en los archivos de configuración, hay que comprobar que no hay ningún error en los ficheros de configuración y se reinicia Nagios:

```
[root@nagios ~]# /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
[root@nagios ~]# service nagios restart
```

2.4. Cacti

Cacti es la herramienta elegida en este trabajo para mostrar las gráficas de los datos obtenidos de los equipos en proceso de monitorización.

2.4.1. Prerrequisitos

La instalación de Cacti requiere la presencia de los paquetes siguientes:

- httpd
- php
- php-mysql
- php-snmp
- mysql
- mysql-server

- net-snmp

En el caso de no tenerlos, se pueden instalar utilizando el sistema de administración e instalación de paquetes RPM.

2.4.2. Configuración de php

Php necesita los módulos siguientes:

- MySQL
- SNMP
- XML
- Session
- Socket

Para verificar que están instalados, se puede utilizar el comando:

```
[root@nagios ~]# php –m
```

Si falta algún módulo en la lista proporcionada por el comando anterior, se debe instalar antes de proceder a la configuración de php mediante los pasos siguientes:

1. Acceder al archivo "*php.ini*" y modificar la línea de código donde se define la ruta hacia el directorio de extensiones. En la mayoría de distribuciones de Linux, este directorio es "*/etc/php.d*" pero en este trabajo se utiliza el directorio "*/etc*".

```
; Directory in which the loadable extensions (modules) reside.
extension_dir = /usr/lib/php/modules/
```

2. Activar la extensión MySQL accediendo a "*/etc/php.d/mysql.ini*" y escribiendo la siguiente línea de código:

```
; Enable mysql extension module
extension=mysql.so
```

3. Activar la extensión SNMP accediendo a "*/etc/php.d/snmp.ini*" y escribiendo la siguiente línea de código:

```
; Enable snmp extension module
extension=snmp.so
```

4. En el caso de utilizar la versión php 4.3.5 o una anterior, se debe añadir la línea de código que se indica a continuación en el fichero "/etc/php.ini". En el caso de utilizar una versión equivalente a la 4.3.6 o superior, no hay que incluir la línea.

```
session.save_path = "/tmp"
```

5. Por último, se activa la directiva que permite cargar archivos HTTP en el fichero “/etc/php.ini”:

```
; Whether to allow HTTP file uploads.
file_uploads = On
```

2.4.3. Configuración del servidor web

Para permitir que los ficheros con extensión “*.php” sean procesados por el intérprete de php5 hay que añadir algunas líneas de código en el archivo de configuración “*php.conf*” que se encuentra en el directorio “*/etc/httpd/conf.d*”.

```
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
```

Las primeras dos líneas son para que el intérprete de php pueda manipular archivos con extensión “*.*php*”. La tercera línea permite añadir el archivo “*index.php*” a la lista de archivos que sirven como índices del directorio.

Además, hay que modificar el archivo de configuración “*httpd.conf*” del servidor web Apache, en el directorio “*/etc/httpd/conf*”, para indicarle al servidor que ha de cargar los archivos de configuración que se encuentran en el directorio *"/etc/httpd/conf.d":*

```
# Load config files from the config directory "/etc/httpd/conf.d".
#
Include conf.d/*.conf
```

2.4.4. Configuración del servicio MySQL

Cacti también necesita que MySQL esté en ejecución. Para arrancar el servicio manualmente se puede utilizar el comando:

```
[root@nagios ~]# /etc/init.d/mysqld start
```

Si se quiere añadir MySQL a la lista de servicios que se arrancan automáticamente cuando se inicia el sistema, se pueden utilizar los comandos:

```
[root@nagios ~]# chkconfig --add mysqld
[root@nagios ~]# chkconfig mysqld on
```

2.4.5. Descarga, instalación y configuración de Cacti

Una vez configurados los módulos necesarios, se procede a la instalación del programa Cacti siguiendo los pasos siguientes:

1. Descargar el paquete de http://www.cacti.net/downloads/cacti-0.8.7g.tar.gz y descomprimirlo:

```
[root@nagios downloads]# tar xzvf cacti-0.8.7g.tar.gz
```

2. Acceder al directorio de cacti, "*/var/www/html/cacti*" y crear una base de datos del tipo MySQL como usuario "root". El nombre de esta base de datos será "cacti":

```
[root@nagios cacti]# mysqladmin --user=root create cacti
```

3. Importar la base de datos que venía por defecto con el paquete de instalación a la base de datos creada:

```
[root@nagios cacti]# mysql cacti < cacti.sql
```

4. Crear un nombre de usuario y contraseña MySQL para Cacti con todos los permisos y privilegios sobre Cacti.

```
[root@nagios cacti]# mysql --user=root mysql
mysql> GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY
'cactiuser';
mysql> flush privileges;
```

5. Modificar el archivo de configuración "*config.php*" del directorio "*/var/www/html/cacti/include*" para especificar el tipo y nombre de la base de datos que se utiliza, el nombre de la máquina donde se instala el programa y el nombre de usuario y contraseña de acceso a Cacti.

```
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "cactiuser";
```

6. Asignar los permisos necesarios sobre los directorios de Cacti para la generación de graficas (/var/www/html/cacti/rra) y archivos de registro (/var/www/html/cacti/log). Es necesario estar dentro del directorio donde se encuentran los archivos de Cacti para cambiar los permisos:

```
[root@nagios ~]# cd /var/www/html/cacti
[root@nagios cacti]# chown -R cactiuser rra/ log/
```

7. Acceder al archivo "*crontab*" que se encuentra en el directorio "*/etc*" y añadir la siguiente línea de código:

```
*/5 * * * * cactiuser php /var/www/html/cacti/poller.php > /dev/null
2>&1
```

De esta manera se indica al administrador de procesos (demonio) "*cron*" que cada 5 minutos, como usuario "cactiuser" debe ejecutar el comando "*php*

/var/www/html/cacti/poller.php", que permite recoger los datos para los equipos y servicios definidos, y que debe ignorar los datos de salida del comando.

8. Acceder a Cacti a través de un navegador web introduciendo la URL de Cacti.

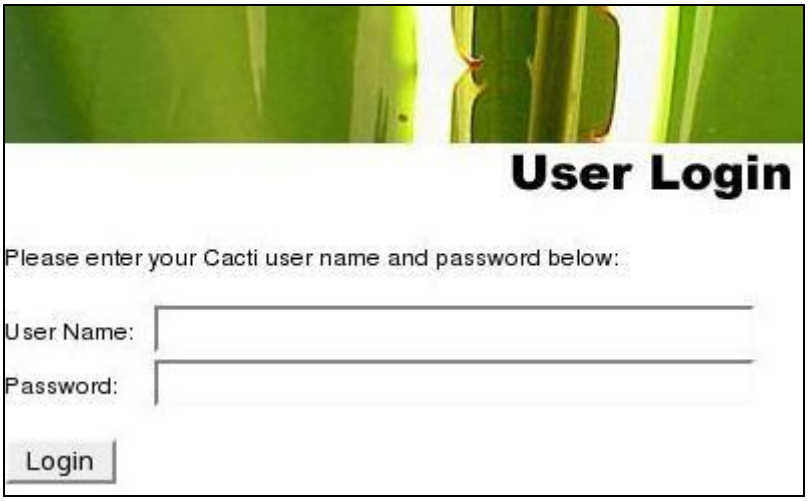


Fig. 2.4 Acceso a Cacti

Se solicitará un nombre de usuario y contraseña.

Una vez instalado y configurado el programa, se pueden empezar a mostrar gráficas.

Para más información, se pueden consultar los manuales que se encuentran en la página oficial de Cacti, "*www.cacti.net*".

2.5. Nmap

Nmap es una herramienta de exploración de redes y su función es la de descubrir nuevos equipos que se añadan a la red. Nmap usa paquetes IP para determinar qué equipos están conectados y mostrando sus servicios, su sistema operativo y otras características.

Una vez se ejecuta el comando de Nmap, se genera un archivo en formato XML en el que se guardan los datos recibidos como respuesta al comando (por ejemplo, la IP del equipo o los servicios que se están ejecutando). Para más información sobre el uso de Nmap se pueden consultar los enlaces siguientes:

"*http://nmap.org/man/es/man-briefoptions.html*"

"*http://nmap.org/man/es/man-host-discovery.html*"

2.5.1. Descarga e Instalación

Para descargar e instalar Nmap y Zenmap se pueden usar los comandos siguientes:

```
[root@nagios~]# rpm -vhU http://nmap.org/dist/ nmap-5.51-1.i386.rpm
[root@nagios~]# rpm -vhU http://nmap.org/dist/zenmap-5.51-1.noarch.rpm
```

Es importante instalar antes Nmap que Zenmap (ya que Zenmap depende de Nmap). Para saber cuál es la versión del software, se puede accede a la dirección "*http://nmap.org/download.html*"

El descubrimiento de equipos se puede realizar mediante líneas de comandos o usando la interfaz gráfica Zenmap.

2.5.2. Nmap2Nagios

El *plugin* Nmap2Nagios es el encargado de que, a partir de los archivos XML de salida de Nmap, se generen los archivos con extensión "*.cfg" donde se describen los nuevos equipos descubiertos, sus servicios y sus características.

Estos archivos "*.cfg" pueden ser leídos por Nagios y mostrados en la interfaz web.

1. Descargar el paquete con el plugin y los archivos de configuración del enlace "*http://exchange.nagios.org/components/com_mtree/attachment.php?link_id=12&cf_id=24*" y descomprimirlo.

```
[root@nagios ~]# cd  /downloads
[root@nagios downloads]# tar xvf nmap2nagios-ng.tar
```

2. Acceder a la carpeta "*nmap2nagios-ng*" y copiar los archivos "*nmap2nagios-ng.pl*" y "*nmap2nagios-ng_3x.conf*" a la carpeta donde se encuentran los otros *plugins* de Nagios, "*/usr/local/nagios/libexec*".

CAPÍTULO 3. MONITORIZACIÓN

El Centro de Procesado de Datos (CDP) que se quiere monitorizar en este trabajo contiene una gran variedad de equipos: sondas de control de temperatura y detección de humedad de la sala, equipos de alimentación ininterrumpida (SAI), servidores de todo tipo, equipos de comunicaciones y el propio servidor de Nagios.

A parte de los equipos del CPD, también se desea monitorizar los conmutadores de los repartidores de planta y las antenas WIFI.

También es parte de este trabajo, la instalación de una herramienta de auto descubrimiento y registro de equipos que permite descubrir todo tipo de equipos conectados a la red, desde servidores hasta impresoras o simples PCs, solamente indicando la dirección IP.

3.1. Protocolo SNMP

Para el proceso de monitorización de los equipos, la mayoría de los *plugins* utilizan el protocolo SNMP (*Simple Network Management Protocol*) para obtener la información necesaria.

El protocolo SNMP es parte de la familia de protocolos TCP/IP. Facilita el intercambio de información de administración entre dispositivos de red y permite supervisar el funcionamiento de los equipos y buscar y resolver posibles problemas. Se han definido tres versiones del protocolo. Las versiones 1 y 2 son las más utilizadas y las que se usan en este trabajo.

SNMP utiliza un servicio no orientado a conexión (UDP) para enviar mensajes entre administradores y agentes. Los agentes son módulos software de administración de red que residen en los dispositivos administrados.

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos: lectura, escritura, notificación y operaciones transversales.

- Comando de lectura. Permite supervisar los elementos de la red examinando las diferentes variables que son mantenidas por los dispositivos administrados.

- Comando de escritura. Permite controlar los elementos de la red y cambiar los valores de las variables almacenadas dentro de los dispositivos administrados.

- Comando de notificación. Permite que los dispositivos administrados reporten los eventos de forma asíncrona. Cuando cierto tipo de evento ocurre, un dispositivo administrado envía una notificación al *Network Management System* (NMS) o sistema de administración de red.

- Operaciones transversales. Se utilizan para determinar qué variables soporta un dispositivo administrado y para recoger la información en tablas de variables.

Las tablas de variables son bases de datos de información de administración o MIBs (*Management Information Bases*) organizadas jerárquicamente. La raíz del árbol es anónima y las ramas o niveles son asignados por diferentes organizaciones de forma estandarizada. Cada objeto administrado en la jerarquía MIB tiene su propio OID (identificador de objeto) que le identifica. Estos objetos son principalmente información de hardware, rendimiento de la CPU y parámetros de configuración.

La Fig. 3.1 muestra un ejemplo de árbol MIB.

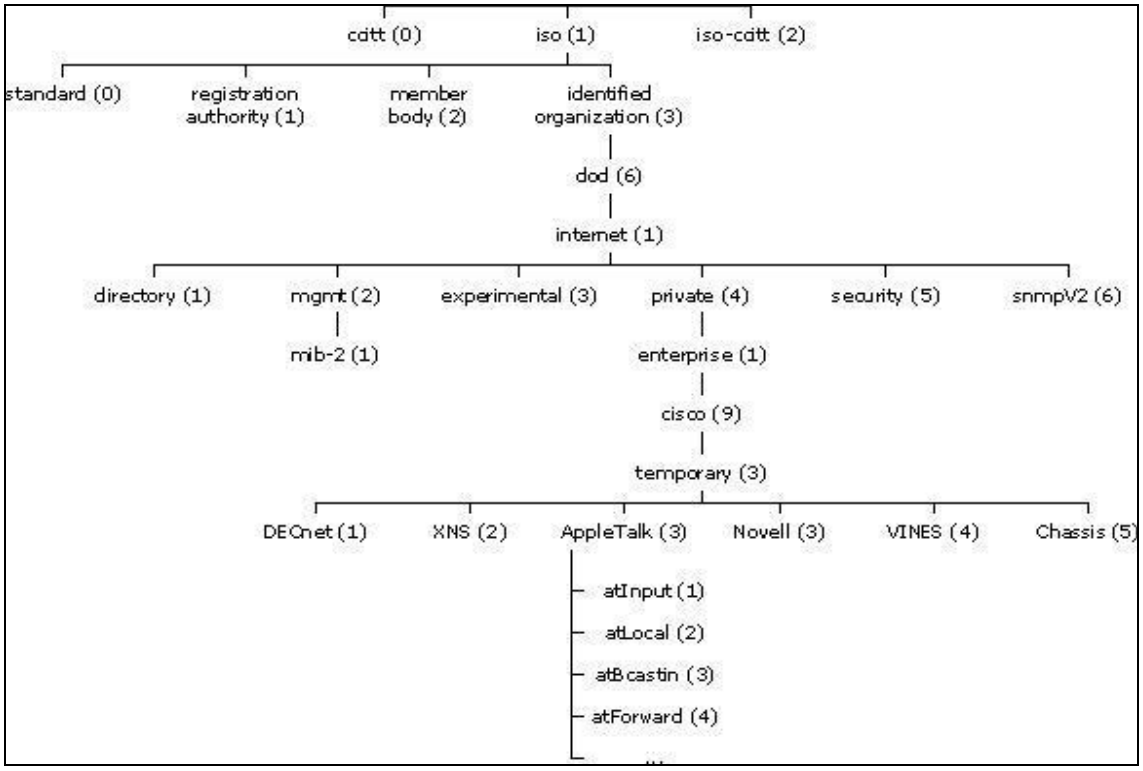


Fig. 3.1 Árbol MIB

3.2. Generación de gráficas

Mediante la herramienta RRDTool y el *plugin* Nagios2RRD, Nagios guarda los datos en bases de datos circulares. Monitorizando las 24 horas del día todos los días de la semana, se genera una gran cantidad de datos. Con la

herramienta RRDTool las bases de datos son circulares y de tamaño definido, de modo que los datos se sobrescriben siguiendo un proceso circular.

Con los datos recogidos se generan gráficas mediante el software Cacti que permite crear gráficas en tiempo de real de los servicios y equipos monitorizados. Para que Cacti pueda crear las gráficas hay que configurarlo para tener acceso a las herramientas y archivos necesarios.

3.2.1. Configuración

Para modificar las opciones de configuración de Cacti hay que acceder a la pestaña "*settings*" en el lateral izquierdo de la página principal de configuración. De entre todas las opciones configurables es necesario modificar al menos dos:

- *General*. En la pestaña "*General*" se pueden modificar algunos parámetros como la versión de las herramientas que se utilizan y la comunidad SNMP. La Fig. 3.4 muestra los valores elegidos en este trabajo.

- *Path***.** En la pestaña **"***Path***"** se pueden definir las rutas hacia los directorios donde hay las herramientas necesarias para la generación de gráficas y los archivos necesarios. La Fig. 3.5 muestra los valores de los campos "*Path*" utilizados en este trabajo

3.2.2. Creación de un dispositivo

Una vez fijados los parámetros básicos de la configuración, hay que crear un dispositivo para cada equipo de la red del que se quieren obtener gráficos. Para ello hay que seleccionar la primera opción en la página principal de Cacti:

Fig. 3.2 Opciones para creación de dispositivos y gráficos

Para cada equipo debe especificarse el nombre del equipo, los parámetros del protocolo SNMP y el tipo de equipo. Como ejemplo, la Fig. 3.3 muestra la definición de un router Cisco denominado "*NeoSky*".

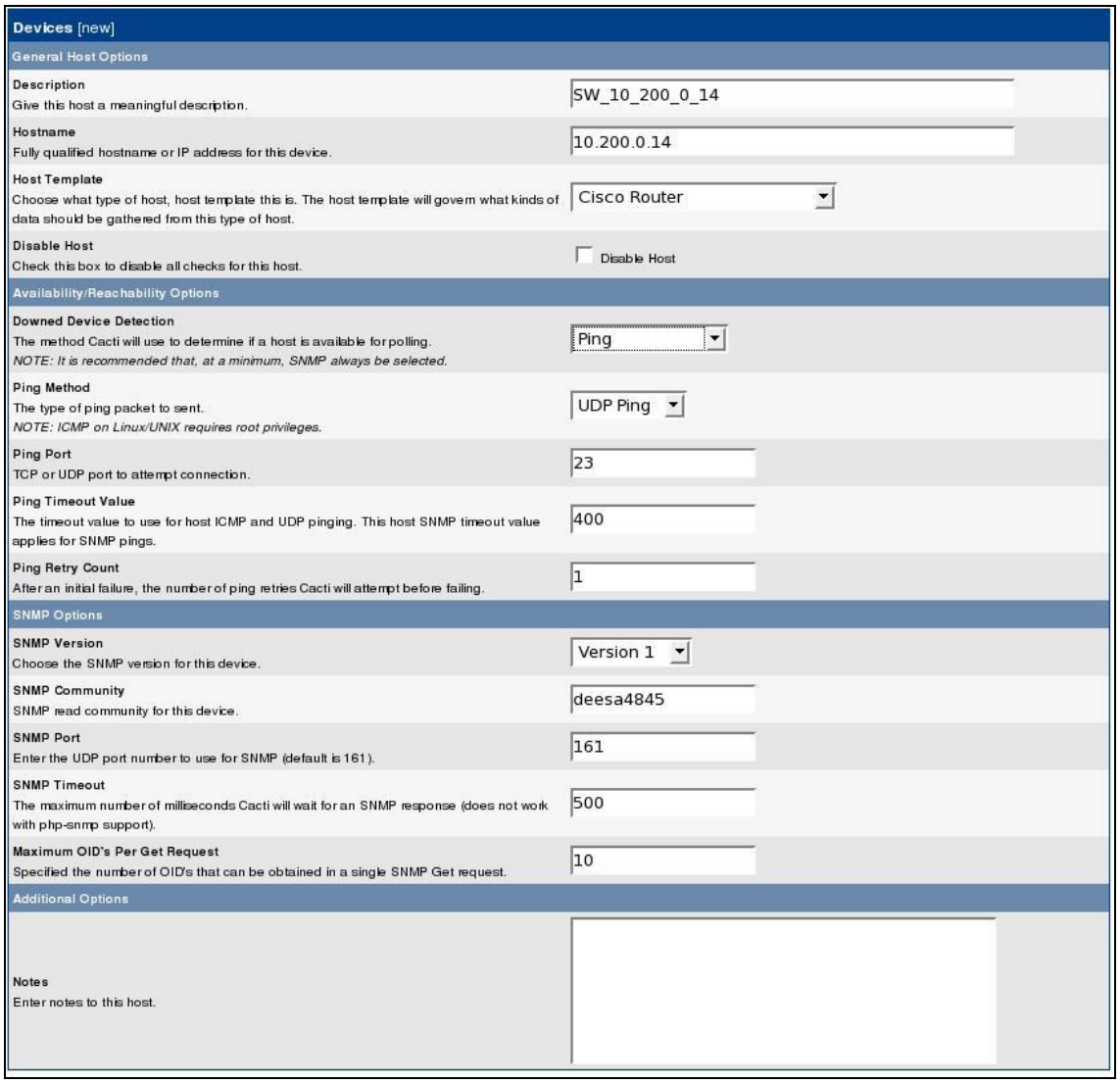


Fig. 3.3 Creación de dispositivo Neosky

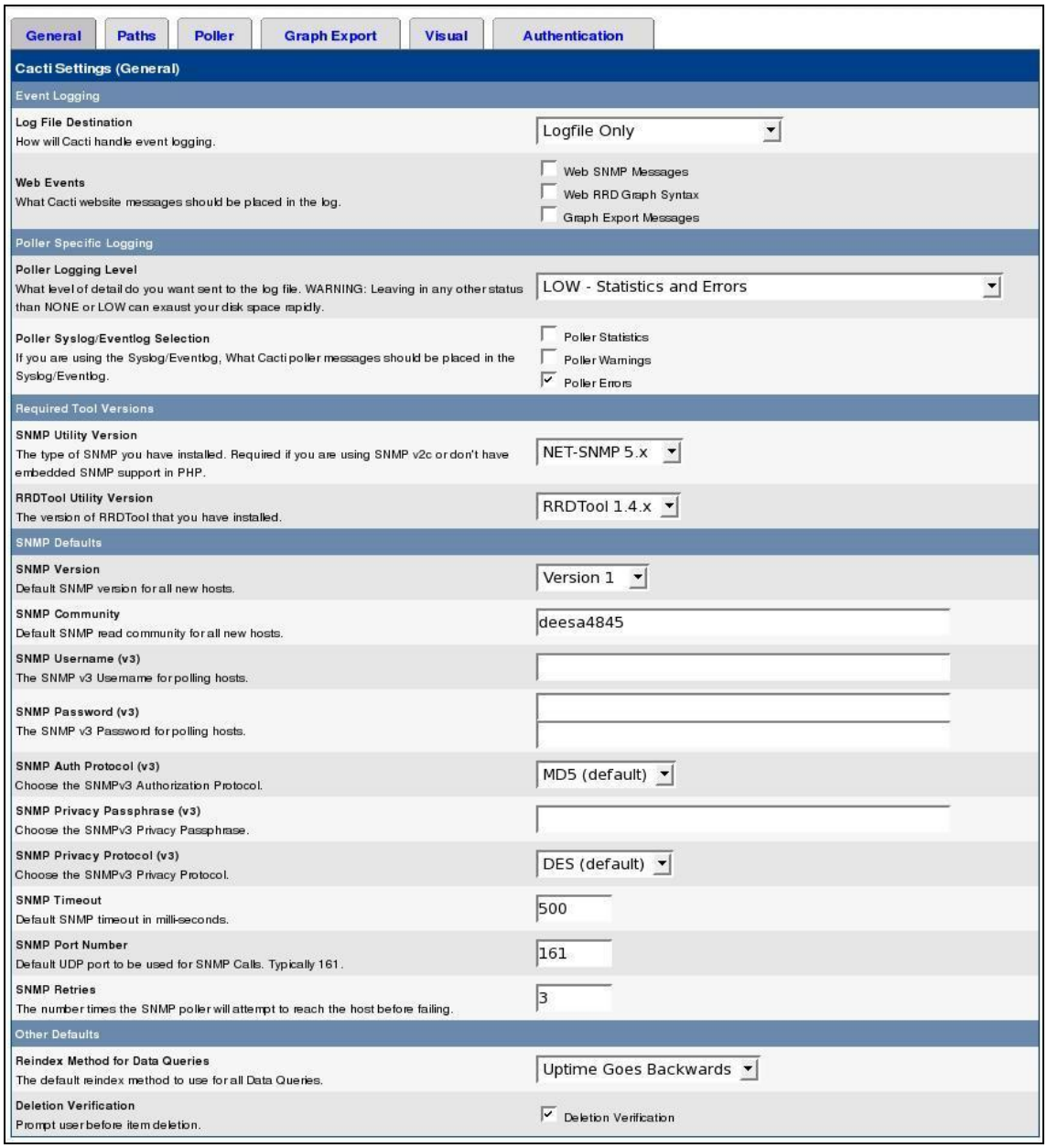


Fig. 3.4 Configuración general de Cacti

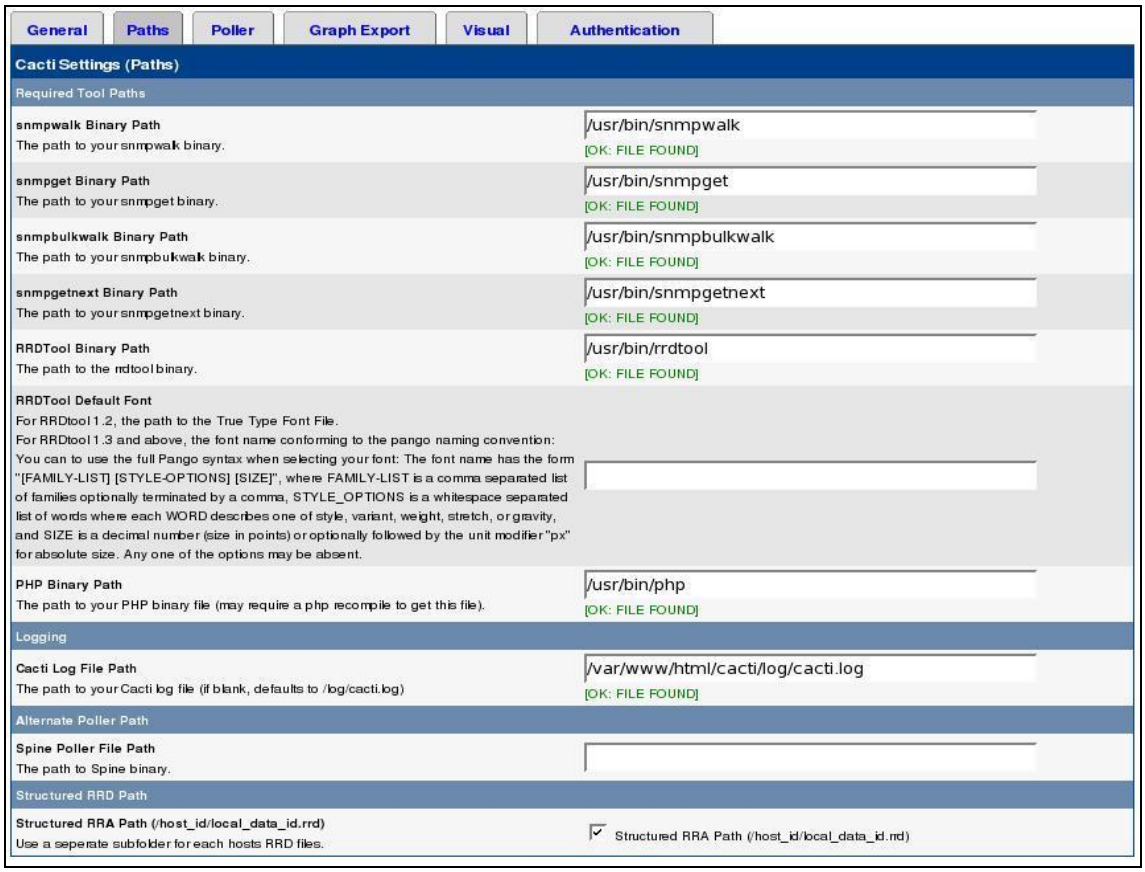


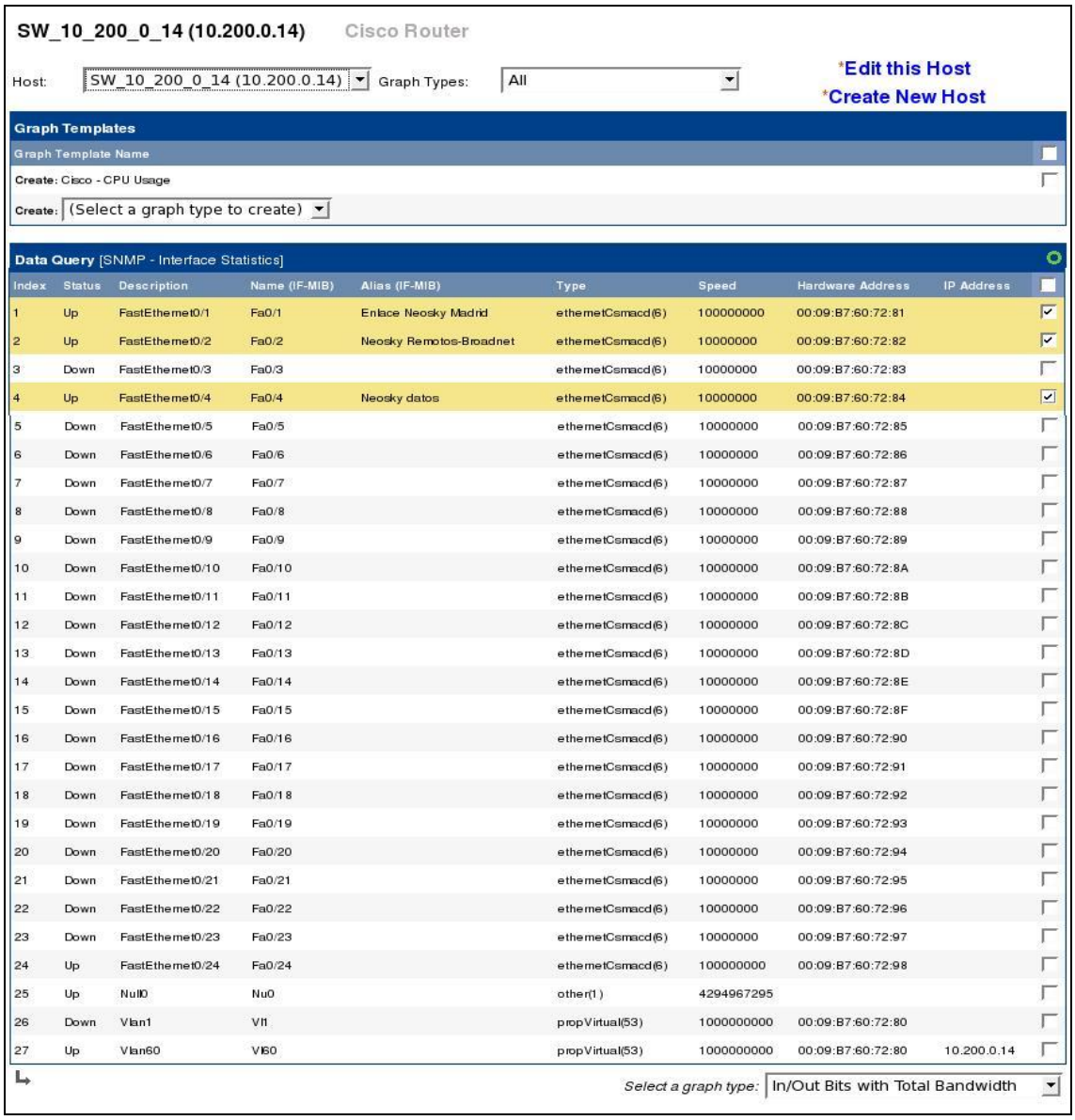
Fig. 3.5 Configuración de la ruta hacia los directorios

3.2.3. Creación de un grafico

Una vez creados los equipos, se pueden definir los gráficos. Para ello, hay que seleccionar la opción de generación de gráfico en la página principal de Cacti y definir el gráfico a crear mediante los pasos siguientes:

1. Seleccionar el dispositivo del cual se quieren obtener gráficas y definir las gráficas a generar.

   Al crear cada dispositivo se le pueden asociar algunas plantillas y datos a consultar (Data Queries) que pueden seleccionarse para mostrar los gráficos de su estado.



SW_10_200_0_14 (10.200.0.14) Cisco Router

Host: SW_10_200_0_14 (10.200.0.14) Graph Types: All

*Edit this Host
*Create New Host

Graph Templates

Graph Template Name	
Create: Cisco - CPU Usage	☐

Create: (Select a graph type to create)

Data Query [SNMP - Interface Statistics]

Index	Status	Description	Name (IF-MIB)	Alias (IF-MIB)	Type	Speed	Hardware Address	IP Address	
1	Up	FastEthernet0/1	Fa0/1	Enlace Neosky Madrid	ethernetCsmacd(6)	100000000	00:09:B7:60:72:81		☑
2	Up	FastEthernet0/2	Fa0/2	Neosky Remotos-Broadnet	ethernetCsmacd(6)	10000000	00:09:B7:60:72:82		☑
3	Down	FastEthernet0/3	Fa0/3		ethernetCsmacd(6)	10000000	00:09:B7:60:72:83		☐
4	Up	FastEthernet0/4	Fa0/4	Neosky datos	ethernetCsmacd(6)	10000000	00:09:B7:60:72:84		☑
5	Down	FastEthernet0/5	Fa0/5		ethernetCsmacd(6)	10000000	00:09:B7:60:72:85		☐
6	Down	FastEthernet0/6	Fa0/6		ethernetCsmacd(6)	10000000	00:09:B7:60:72:86		☐
7	Down	FastEthernet0/7	Fa0/7		ethernetCsmacd(6)	10000000	00:09:B7:60:72:87		☐
8	Down	FastEthernet0/8	Fa0/8		ethernetCsmacd(6)	10000000	00:09:B7:60:72:88		☐
9	Down	FastEthernet0/9	Fa0/9		ethernetCsmacd(6)	10000000	00:09:B7:60:72:89		☐
10	Down	FastEthernet0/10	Fa0/10		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8A		☐
11	Down	FastEthernet0/11	Fa0/11		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8B		☐
12	Down	FastEthernet0/12	Fa0/12		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8C		☐
13	Down	FastEthernet0/13	Fa0/13		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8D		☐
14	Down	FastEthernet0/14	Fa0/14		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8E		☐
15	Down	FastEthernet0/15	Fa0/15		ethernetCsmacd(6)	10000000	00:09:B7:60:72:8F		☐
16	Down	FastEthernet0/16	Fa0/16		ethernetCsmacd(6)	10000000	00:09:B7:60:72:90		☐
17	Down	FastEthernet0/17	Fa0/17		ethernetCsmacd(6)	10000000	00:09:B7:60:72:91		☐
18	Down	FastEthernet0/18	Fa0/18		ethernetCsmacd(6)	10000000	00:09:B7:60:72:92		☐
19	Down	FastEthernet0/19	Fa0/19		ethernetCsmacd(6)	10000000	00:09:B7:60:72:93		☐
20	Down	FastEthernet0/20	Fa0/20		ethernetCsmacd(6)	10000000	00:09:B7:60:72:94		☐
21	Down	FastEthernet0/21	Fa0/21		ethernetCsmacd(6)	10000000	00:09:B7:60:72:95		☐
22	Down	FastEthernet0/22	Fa0/22		ethernetCsmacd(6)	10000000	00:09:B7:60:72:96		☐
23	Down	FastEthernet0/23	Fa0/23		ethernetCsmacd(6)	10000000	00:09:B7:60:72:97		☐
24	Up	FastEthernet0/24	Fa0/24		ethernetCsmacd(6)	100000000	00:09:B7:60:72:98		☐
25	Up	Null0	Nu0		other(1)	4294967295			☐
26	Down	Vlan1	Vl1		propVirtual(53)	1000000000	00:09:B7:60:72:80		☐
27	Up	Vlan60	Vl60		propVirtual(53)	1000000000	00:09:B7:60:72:80	10.200.0.14	☐

Select a graph type: In/Out Bits with Total Bandwidth

Fig. 3.6 Selección de interfaces de las cuales generar graficas del dispositivo Neosky

En la Fig. 3.6 se muestran las interfaces del router Cisco configurado en el apartado anterior, que pueden ser seleccionadas para la generación de gráficos. Para este dispositivo también se pueden generar gráficos de los datos entrantes y salientes, como muestra la Fig. 3.7:

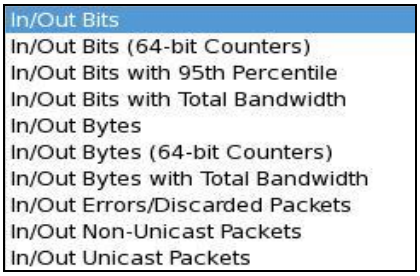


Fig. 3.7 Selección tipo de grafico del dispositivo Neosky

2. Para visualizar los gráficos hay que pulsar el botón "*Graphs*" que se muestra en la parte superior de la ventana y seleccionar el dispositivo del que se quieren obtenerse los gráficos.

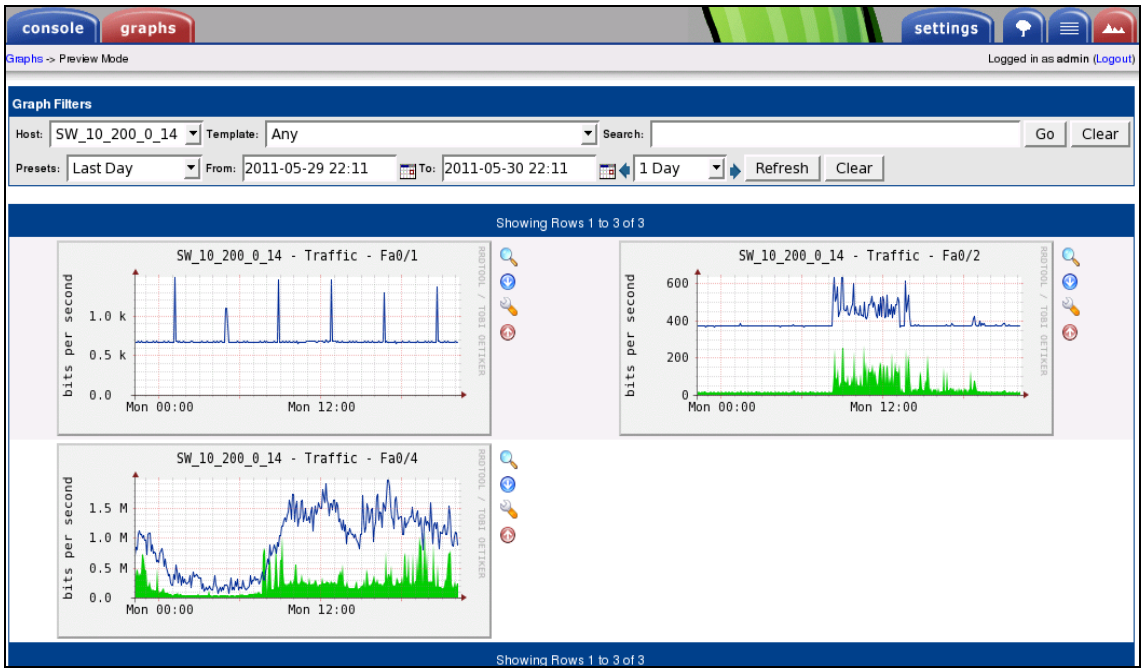


Fig. 3.8 Graficas del dispositivo Neosky

Pulsando sobre cada una de las gráficas se pueden visualizar los resultados a escala anual, mensual, semanal o diaria.

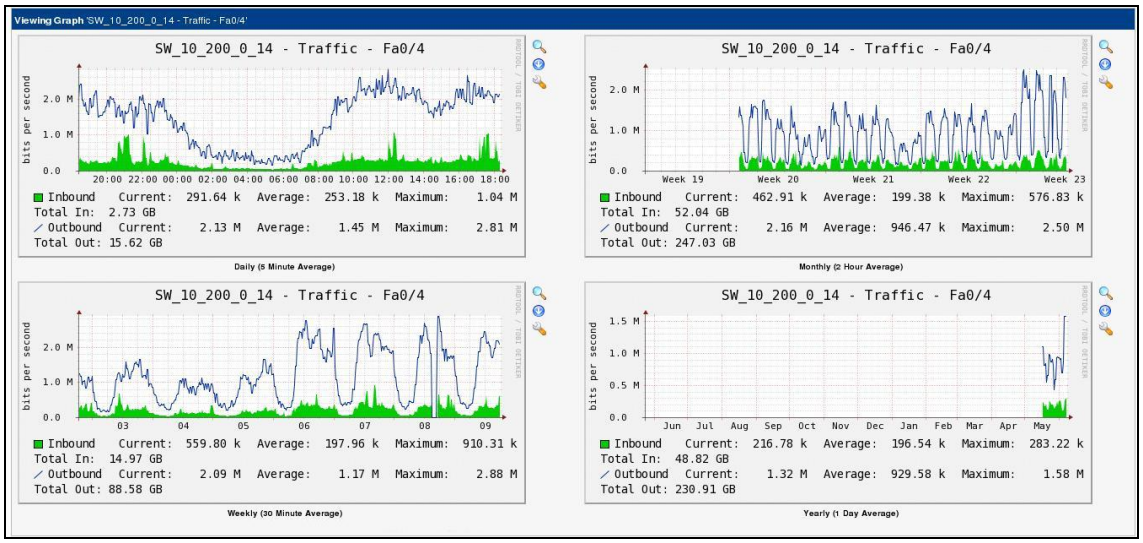


Fig. 3.9 Graficas sobre el avance diario, semanal, mensual y anual del dispositivo Neosky

Hay que repetir estos mismos pasos para todos los dispositivos de los que se quiere visualizar alguna gráfica.

3.2.3.1. Datos a tener en cuenta

Con la instalación de Cacti vienen integradas por defecto algunas plantillas básicas de equipos, gráficos y datos. En el caso de no disponer de plantillas para el dispositivo a monitorizar, hay que crearlos.
También existe la posibilidad de encontrar plantillas en internet, que se pueden descargar e importar a Cacti.

Para realizar estas dos tareas, se puede seguir el manual de Cacti:
"*http://www.cacti.net/downloads/docs/html/*"

3.3. Monitorización de equipos

En este apartado se definen los pasos a seguir para monitorizar los equipos y los servicios. Como ejemplo, se detalla la monitorización del equipo Nagios. La monitorización de los otros equipos, mencionados a continuación, se puede consultar en el anexo: Monitorización de equipos (Anexo C).

- SAI
- Netbotz
- Equipos Dell
- Antena WiFi
- Switch y Router
- Equipos de Conexión al exterior

3.3.1. Definición del equipo y servicios a monitorizar en Nagios

- Definición de un equipo genérico y un servicio genérico:

  Para monitorizar cada equipo y cada servicio, se parte de una plantilla genérica que permite fijar algunos parámetros que heredarán los equipos y servicios que utilicen esta plantilla en su definición.

  Estas plantillas genéricas se definen en el archivo de plantillas, "*/usr/local/nagios/etc/objects/templates.cfg*".

```
# Generic host definition template

define host{
        name                            host        ; The name of this host template
        notifications_enabled           1           ; Host notifications are enabled
        event_handler_enabled           1           ; Host event handler is enabled
        flap_detection_enabled          1           ; Flap detection is enabled
        failure_prediction_enabled      1           ; Failure prediction is enabled
        process_perf_data               1           ; Process performance data
        retain_status_information       1           ; Retain status information across
                                                    program restarts
        retain_nonstatus_information    1           ; Retain non-status information
                                                    across program restarts
```

```
        notification_period             24x7        ; Send host notifications at any
                                                    time
        register                        0           ; DONT REGISTER THIS DEFINITION
        }
```

```
# Generic service definition template

define service{
        name                            service     ; The 'name' of this service template
        active checks enabled           1           ; Active service checks are enabled
        passive_checks_enabled          1           ; Passive service checks are
                                                    enabled-accepted
        parallelize_check               1           ; Active service checks should be
                                                    parallelized (disabling this can lead
                                                    to major performance problems)
        obsess_over_service             1           ; We should obsess over this service
                                                    (if necessary)
        check_freshness                 0           ; Default is to NOT check service
                                                    'freshness'
        notifications enabled           0           ; Service notifications are enabled
        event handler enabled           1           ; Service event handler is enabled
        flap detection enabled          1           ; Flap detection is enabled
        failure_prediction_enabled      1           ; Failure prediction is enabled
        process_perf_data               1           ; Process performance data
        retain_status_information       1           ; Retain status information across
                                                    program restarts
        retain_nonstatus_information    1           ; Retain non-status information
                                                    across program restarts
        is_volatile                     0           ; The service is not volatile
        check_period                    24x7        ; The service can be checked at any
                                                    time of the day
        max_check_attempts              3           ; Re-check the service up to 3 times
                                                    in order to determine its final
                                                    (hard) state
        normal_check_interval           10          ; Check the service every 10 minutes
                                                    under normal conditions
        retry_check_interval            2           ; Re-check the service every two
                                                    minutes until a hard state can be
                                                    determined
        contact_groups                  admins      ; Notifications get sent out to
                                                    everyone in the 'admins' group
                                                    notification_options
                                                    w,u,c,r         ; Send notifications
                                                    about warning, unknown, critical, and
                                                    recovery events
        notification_interval           60          ; Re-notify about service problems
                                                    every hour
        notification_period             24x7        ; Notifications can be sent out at
                                                    any time
        register                        0           ; DONT REGISTER THIS DEFINITION
        }
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\templates.cfg*".

– Definición de equipos tipo "linux-server":

El equipo que se desea monitorizar en este ejemplo tiene sistema operativo Linux. Como habrá más equipos a monitorizar con Linux, se define un host tipo Linux en el mismo fichero "templates.cfg" que servirá de plantilla común a todos ellos.

```
# Linux host definition template

define host{
    name                    linux-server            ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
```

```
    check interval        5                     ; Actively check the host every 5
                                                 minutes
    retry interval        1                     ; Schedule host check retries at 1
                                                 minute intervals
    max_check_attempts    10                    ; Check each Linux host 10 times (max)
    check_command         check-host-alive      ; Default command to check Linux hosts
    notification_period   workhours             ; we only notify during the day
                                                ; Note that the notification period
                                                 variable is being overridden from
                                                   ; the value that is inherited from
                                                    the generic-host template.
    notification_interval   120                   ; Resend notifications every 2 hours
    notification options    d,u,r                 ; Only send notifications for
                                                  specific host states
    contact groups          admins                ; Notifications get sent to the
                                                  admins by default
    register                0                     ; DONT REGISTER THIS DEFINITION
    }
```

– Definición del equipo “localhost” de tipo “linux-server”:

Ahora se debe crear una carpeta “*Linux*” en “*/usr/local/nagios/etc/objects*” y un archivo “*localhost.cfg*” en el que se defina el host a monitorizar, en este caso Nagios bajo el nombre “localhost”.

```
###############################################################################
#
# HOST DEFINITION
#
###############################################################################

# Define a host for the local machine

define host{
        use         linux-server            ; Name of host template to use
                                            ; This host definition will inherit all
                                            variables that are defined
                                            ; In (or inherited by) the linux-server host
                                            template definition.
        host name   localhost
        alias       localhost
        address     127.0.0.1
}
```

o Acceso a este archivo mediante la siguiente ruta del CD adjunto: “*TFC_Fadi Taki:\Objects\Linux\localhost.cfg*”.

– Definición de un grupo de equipos de tipo “linux”:

En la carpeta creada antes, “*/usr/local/nagios/etc/objects/Linux*”, se crea otro archivo, “*linux.cfg*”, en el que se define un grupo de equipos y se indican los equipos que son miembros de este grupo.

```
###############################################################################
#
# HOST GROUP DEFINITION
#
###############################################################################
define hostgroup {
       hostgroup_name        linux
       alias                 Linux Servers
       members               srv161.esade.es,srv162.esade.es,localhost
}
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\Linux\linux.cfg*".

– Definición de un servicio "local-service"

En el archivo de plantillas, "*/usr/local/nagios/etc/objects/templates.cfg*", se define un servicio que hereda los parámetros del servicio genérico definido anteriormente y en el que se especifican algunos parámetros de monitorización como el periodo de muestreo, la habilitación o no de las notificaciones, etc.

```
# Local service definition template

define service{
        name                    local-service ; The name of this service template use
                                              service
                                              ; Inherit default values from the generic-
                                              service definition
        max_check_attempts      4             ; Re-check the service up to 4 times in
                                              order to determine its final (hard) state
        normal_check_interval   5             ; Check the service every 5 minutes under
                                              normal conditions
        retry_check_interval    1             ; Re-check the service every minute until
                                              a hard state can be determined
        register                0             ; DONT REGISTER THIS DEFINITION
}
```

– Definición de los servicios a monitorizar

En el archivo de comandos, "*usr/local/nagios/etc/objects/commands.cfg*", se definen los comandos que cualquier equipo puede utilizar para monitorizar los servicios deseados.

Para utilizar un comando, se debe acceder al fichero de configuración del equipo (usr/local/nagios/etc/objects/Linux/localhost.cfg) e invocar los comandos con el valor deseado.

Por ejemplo:

- Definición del comando "*check_ping*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
# 'check ping' command definition
define command{
        command name    check ping
        command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
}
```

La directiva "command_line" especifica los argumentos de la instrucción. En este caso, se usa la macro "$USER1$" (que es la ruta hacia el directorio donde se encuentra la herramienta), la macro "$HOSTADDRESS$" (que es la dirección IP del equipo), y las macros "$ARG1$" y "$ARG2$" (que son argumentos para pasar parámetros).

- Definición del servicio "*PING*" en el archivo "*usr/local/nagios/etc/objects/Linux/localhost.cfg*":

```
# Define a service to "ping" the local machine

define service{
        use                     local-service      ; Name of service template to use
        host name               localhost
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
}
```

El valor del argumento “$ARG1$=100.0,20%” indica que se generará una alerta del tipo “WARNING” en el caso de que el tiempo de respuesta al ping sea mayor que 100 ms o en el caso de que la pérdida de paquetes sea mayor del 20%. El argumento “$ARG2$=500.0,60%”indica que se generará una alerta del tipo “CRITICAL” en el caso de que el tiempo de respuesta al ping sea mayor que 500 ms o en el caso de que la pérdida de paquetes sea mayor que el 60%. Si no se produce ninguno de estos casos, la alerta generada es del tipo “OK”.

– Visualización en la interfaz web

Una vez definido el equipo y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios, tal y como muestra la Fig. 3.9.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-05-2011 11:55:53	130d 19h 12m 48s	1/4	OK - load average: 0.00, 0.04, 0.08
	Current Users	OK	10-05-2011 11:56:31	130d 19h 12m 10s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	10-05-2011 11:54:12	123d 23h 7m 0s	1/4	HTTP OK: HTTP/1.1 200 OK - 261 bytes in 0.000 second response time
	PING	OK	10-05-2011 11:52:27	4d 14h 55m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.01 ms
	Root Partition	OK	10-05-2011 11:56:10	130d 19h 10m 18s	1/4	DISK OK - free space: / 56945 MB (91% inode=99%):
	SSH	OK	10-05-2011 11:56:08	11d 12h 16m 23s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Swap Usage	OK	10-05-2011 11:53:26	130d 19h 9m 3s	1/4	SWAP OK - 100% free (1983 MB out of 1983 MB)
	Total Processes	OK	10-05-2011 11:55:25	130d 19h 8m 25s	1/4	PROCS OK: 62 processes with STATE = RSZDT
	check_icmp	OK	10-05-2011 11:52:35	105d 18h 55m 40s	1/4	OK - 127.0.0.1: rta 0.021ms, lost 0%

Fig. 3.9 Visualización de los servicios monitorizados del host Nagios en la interfaz web

3.3.2. Visualización de gráficas en Cacti

Para el equipo Nagios se han definido las gráficas que permiten visualizar la carga de la CPU, el número de usuarios activos en la maquina y la latencia del Ping. El proceso para definir las gráficas es el que se explica en el apartado 3.2.3.

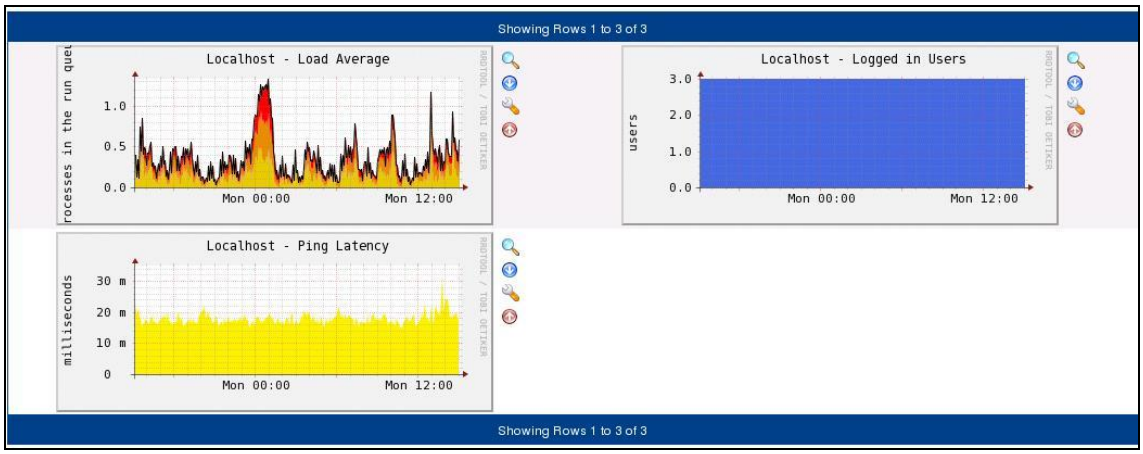


Fig. 3.10 Gráficas del equipo Nagios en Cacti

3.4. Autodescubrimiento de equipos

El autodescubrimiento de equipos es un servicio que se ha definido en este trabajo. Este servicio permite descubrir automáticamente los equipos de la red y registrarlos automáticamente en función del tipo de dispositivo (cosa que ahorra mucho tiempo y trabajo al administrador de la red).

El servicio de autodescubrimiento de equipos se ha definido como un servicio más del equipo Nagios.

Para definirlo, hay que seguir los pasos siguientes:

1. Adaptar el archivo de configuración "*nmap2nagios-ng_3x.conf"* del directorio "*/usr/local/nagios/libexec/"* a la necesidades del sistema de monitorización para recoger los datos de cada equipo y registrarlos en el archivo de configuración de los equipos. Se definen los parámetros y las opciones de creación de cada equipo, grupo de equipos y servicios una vez descubierto un nuevo equipo.

   – Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\ Autodescubrimiento\ nmap2nagios-ng_3x.conf*".

Lo que se ha hecho es añadir el servicio al host Nagios y se ha indicado que se ejecute cada 5 minutos. Es muy importante definir este valor correctamente ya que según la red que se esté descubriendo, ejecutando este comando cada 5 minutos puede provocar la saturación de la red. En este trabajo se ha puesto que sean 5 minutos porque solamente se ha descubierto una red de 15 dispositivos como ejemplo.

Lo más normal es que se cree una línea de comandos en el "*crontab*" indicado que se ejecute el servicio de descubrimiento una vez cada día, y normalmente por la noche, cuando la red está siendo menos utilizada.

3.4.1. Archivo de configuración

Acceder al archivo de configuración en el que se definen los parámetros y la opción de creación de cada host, grupo de host y servicios una vez descubierto un nuevo equipo. Este archivo es: "*/usr/local/nagios/libexec/nmap2nagios-ng_3x.conf*".

Definición de un host genérico

```
  <host>
        <host_name>default</host_name>
        <alias>alias</alias>
        <address></address>
<template>default-template</template>
        <check_command>check-host-alive</check_command>
        <checks_enabled>1</checks_enabled>
        <max_check_attempts>3</max_check_attempts>
        <notification_interval>5</notification_interval>
        <notification_period>24x7</notification_period>
        <notification_options>d,u,r</notification_options>
        <notifications_enabled>1</notifications_enabled>
        <event_handler_enabled>1</event_handler_enabled>
        <flap_detection_enabled>1</flap_detection_enabled>
        <flap_detection_options>o,d,u</flap_detection_options>
        <initial_state>o</initial_state>
        <process_perf_data>1</process_perf_data>
        <retain_status_information>1</retain_status_information>
        <retain_nonstatus_information>1</retain_nonstatus_information>
        <contact_groups>admins</contact_groups>
        <register>0</register>

  </host>
```

Definición de un grupo de hosts. En este caso se muestra la definición del grupo de equipos tipo Linux. Los demás grupos se definen de la misma manera.

```
  <hostgroup>
    <group_name>linux</group_name>
    <group_alias>Linux Servers</group_alias>
    <match>
      <field>osclass</field>
      <data>linux</data>
    </match>
  </hostgroup>
```

Definición de un servicio genérico. Se define un servicio genérico en el que se fijan parámetros que heredarán los demás servicios que lo incluyan en la definición.

```
<service>
        <name>default</name>
        <template>default-service-template</template>
        <service_description>description</service_description>
        <check_command></check_command>
        <max_check_attempts>3</max_check_attempts>
        <normal_check_interval>5</normal_check_interval>
        <retry_check_interval>1</retry_check_interval>
        <check_period>24x7</check_period>
        <notification_interval>5</notification_interval>
        <notification_period>24x7</notification_period>
        <notification_options>c,r</notification_options>
```

```
        <notifications_enabled>1</notifications_enabled>
        <contact_groups>admins</contact_groups>
        <active_checks_enabled>1</active_checks_enabled>
        <passive_checks_enabled>0</passive_checks_enabled>
        <parallelize_check>1</parallelize_check>
        <obsess_over_service>0</obsess_over_service>
        <check_freshness>0</check_freshness>
        <event_handler_enabled>1</event_handler_enabled>
        <flap_detection_enabled>1</flap_detection_enabled >
        <flap_detection_options>w,u,c,o</flap_detection_options>
        <process_perf_data>1</process_perf_data>
        <retain_status_information>1</retain_status_information>
        <retain_nonstatus_information>1</retain_nonstatus_information>
        <initial_state>o</initial_state>
        <register>0</register>
  </service>
```

Definición de un servicio. Como ejemplo se muestra la definición del servicio http.

```
  <service>
    <name>http</name>
    <service_description>HTTP</service_description>
    <check_command>check_http</check_command>
  </service>
```

En los siguientes apartados se muestra cómo se heredaran los parámetros definidos en los hosts, grupos de hosts y servicios indicando a cada opción qué parámetro heredar y como se registrarán en el archivo de nuevos hosts.

3.4.2. Grupo de hosts y host

```
]]></header>
<hostgroup_header><![CDATA[
]]></hostgroup_header>

<hostgroup_entry><![CDATA[
define hostgroup {
        hostgroup_name {--group_name--}
        alias {--group_alias--}
        members {--hosts--}
}

]]></hostgroup_entry>
<hostgroup_footer><![CDATA[
]]></hostgroup_footer>

<host_header><![CDATA[
]]></host_header>

<host_template_entry><![CDATA[
define host{
    name                            {--template--}
    notifications_enabled           {--notifications_enabled--}
    event_handler_enabled           {--event_handler_enabled--}
    flap_detection_enabled          {--flap_detection_enabled--}
    flap_detection_options           {--flap_detection_options--}
    process_perf_data               {--process_perf_data--}
    retain_status_information       {--retain_status_information--}
    retain_nonstatus_information    {--retain_nonstatus_information--}
    register                        {--register--}
}

]]></host_template_entry>

<host_entry><![CDATA[
define host {
        use                             {--template--}
```

```
        host_name                       {--host_name--}
        alias                           {--alias--}
        address                         {--address--}
        check_command                   {--check_command--}
        max_check_attempts              {--max_check_attempts--}
        checks_enabled                  {--checks_enabled--}
        notification_interval           {--notification_interval--}
        initial_state                   {--initial_state--}
        contact_groups                  {--contact_groups--}
}
```

3.4.3. Servicio

Acceder al archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*" y definir el comando "*check_find_new_hosts*".

```
]]></host_footer>
<service_header/>
<service_template_entry><![CDATA[

define service{
    name                            {--template--}
    active_checks_enabled           {--active_checks_enabled--}
    passive_checks_enabled          {--passive_checks_enabled--}
    parallelize_check               {--parallelize_check--}
    obsess_over_service             {--obsess_over_service--}
    check_freshness                 {--check_freshness--}
    notifications_enabled           {--notifications_enabled--}
    event_handler_enabled           {--event_handler_enabled--}
    flap_detection_enabled          {--flap_detection_enabled--}
    flap_detection_options          {--flap_detection_options--}
    process_perf_data               {--process_perf_data--}
    retain_status_information       {--retain_status_information--}
    retain_nonstatus_information    {--retain_nonstatus_information--}
    register                        {--register--}
}

]]></service_template_entry>
<service_entry><![CDATA[
define service {
        use                                     {--template--}
        host_name                               {--host_name--}
        service_description                     {--service_description--}
        check_command                           {--check_command--}
        max_check_attempts                      {--max_check_attempts--}
        normal_check_interval                   {--normal_check_interval--}
        retry_check_interval                    {--retry_check_interval--}
        check_period                            {--check_period--}
        notification_interval                   {--notification_interval--}
        notification_period                     {--notification_period--}
        notification_options                    {--notification_options--}
        contact_groups                          {--contact_groups--}
        initial_state                           {--initial_state--}
}

]]></service_entry>
<service_footer/>
</template>
</n2n>
```

3.4.4. Ejecución del Servicio

Acceder al archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*" y definir el comando "*check_find_new_hosts*".

```
#check_find_new_host
###############
define command{

       command name   check find new hosts
       command_line   /usr/local/nagios/libexec/check_find_new_hosts - v
/usr/local/nagios/etc/objects/hosts_hostgroups_and_services 192.168.1.1 28 admins
       }
```

En la línea de comandos "*command_line*" se indica que la red que se quiere descubrir es la "*192.168.1.1*" con una máscara "*28*" y autorización del grupo de administradores "*admins*". Como se indicó en el apartado 2.5, puede variar la lista de datos dependiendo de las opciones del comando.

Se define el servicio "*check_find_new_hosts*" en el archivo de definición de host y servicios a monitorizar del dispositivo, "*usr/local/nagios/etc/objects/localhost/localhost.cfg*":

```
# Check_find_new_hosts

define service{
        use                     service              ; Name of service template to use
        host name               localhost
        service_description     FIND_NEW_HOSTS       ; STRING can be anything.
        is volatile             0
        check_period            24x7
        max_check_attempts      3                    ; Re-check the service up to 4 times
                                                     in order to determine its final
                                                     (hard) state
        normal_check_interval   360                  ; Check the service every 5 minutes
                                                     under normal conditions
        retry_check_interval    1                    ; Re-check the service every minute
                                                     until a hard state can be determined
        register                0                    ; DONT REGISTER THIS DEFINITION
        check_command           check_find_new_hosts
        }
```

Esto hace que se cree un archivo en el directorio "*/usr/local/nagios/etc/objects/hosts_hostgroups_and_services*" en cual se guardan los nuevos hosts descubiertos.

A continuación se muestra un ejemplo de un nuevo host registrado en este archivo:

```
define hostgroup {
       hostgroup name linux
       alias          Linux Servers
       members        srv161.esade.es
}

define host{
    name                            default-template
    notifications_enabled           1
    event_handler_enabled           1
    flap_detection_enabled          1
    flap_detection_options          o,d,u
    process perf data               1
    retain status information       1
    retain_nonstatus_information    1
    register                        0
}

define host {
       use                          default-template
       host_name                    srv161.esade.es
```

```
        alias                           alias
        address                         192.168.1.7
        check_command                   check-host-alive
        max_check_attempts              3
        checks_enabled                  1
        notification_interval           5
        initial state                   o
        contact_groups                  admins
}

define service{
    name                                default-service-template
    active checks enabled               1
    passive checks enabled              0
    parallelize check                   1
    obsess_over_service                 0
    check_freshness                     0
    notifications enabled               1
    event handler enabled               1
    flap detection enabled              1
    flap_detection_options              w,u,c,o
    process_perf_data                   1
    retain_status_information           1
    retain nonstatus information        1
    register                            0
}

define service {
        use                                     default-service-template
        host_name                               srv161.esade.es
        service description                     Connectivity
        check command                           check_icmp
        max_check_attempts                      3
        normal_check_interval                   5
        retry check interval                    1
        check_period                            24x7
        notification interval                   5
        notification_period                     24x7
        notification_options                    c,r
        contact_groups                          admins
        initial_state                           o
}
```

CAPÍTULO 4. CONCLUSIONES Y FUTURAS IMPLEMENTACIONES

4.1. Conclusiones

El sistema implementado en este proyecto ha permitido configurar un sistema de monitorización de la red y del centro de procesamiento de datos de la Fundación ESADE, considerada una de las escuelas más importantes a nivel internacional. Con este sistema se puede realizar la monitorización de todos los equipos y servicios desde una aplicación web sin tener que acceder manualmente a cada equipo para revisar su estado.

El sistema integra un sistema de notificación encargado de enviar avisos cuando se produce cualquier incidencia. Así, el administrador de la red no tiene que estar pendiente de comprobar constantemente el estado de los equipos y de los servicios, ya que es informado automáticamente de cualquier problema.
El diseño del sistema de monitorización se ha hecho lo suficientemente genérico para poder adaptarlo a la monitorización de cualquier sistema. Una vez planteado el diseño general, se ha particularizado la implementación a las necesidades de monitorización de la red de ESADE.

Una de las mayores aportaciones del proyecto es la creación del servicio de autodescubrimiento de equipos. Este servicio permite reducir de meses a semanas el trabajo de creación de un sistema de monitorización de red, permitiendo un ahorro considerable en términos temporales y económicos.

4.2. Futuras Implementaciones

Las mejoras del sistema desarrollado que se proponen para un trabajo futuro incluyen los puntos siguientes:

- Depuración de la herramienta de autodescubrimiento para registrar los equipos descubiertos en archivos separados y ordenados en directorios según el tipo de equipo.

- Monitorización de todos los equipos informáticos conectados a la red: PCs de las aulas de informática y de los despachos, impresoras, servidores, conmutadores y antenas WiFi de todos los edificios que pertenecen a la Fundación.

- Monitorización de los equipos de aire acondicionado que se encuentran en el centro de procesamiento de datos. Actualmente solamente se

puede ver el estado de estos equipos accediendo al CPD y revisándolos en situ. Pero pronto se instalará un conversor de puerto serie-Ethernet RS232 o RS 485 y se hará un estudio sobre los datos a monitorizar recogiendo los arboles MIB y el valor de los OID mediante el protocolo SNMP.

BIBLIOGRAFÍA

[1] Documentación y manuales de Linux: http://www.linux-es.org

[2] Documentación y manuales de Red_Hat_Enterprise_Linux: http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/index.html

[3] Documentación y manuales de Nagios: http://www.nagios.org/

[4] Foros de ayuda y soporte de Nagios: http://wiki.nagios.org/index.php/Forums

[5] Documentación y Plugins de Nagios: http://exchange.nagios.org/

[6] Plugins de Nagios: http://nagios.manubulon.com/

[7] Documentación y manuales de Pandora: http://pandorafms.org/

[8] Documentación y manuales de Zabbix: http://www.zabbix.com/

[9] Documentación y manuales de Zenoss: http://www.zenoss.com/

[10] Información web general sobre RRD: http://es.wikipedia.org/wiki/RRDtool

[11] Documentación y manuales de RRD: http://www.mrtg.org/rrdtool/

[12] Información web general sobre RRD: http://www.loriotpro.com/Products/On-line_Documentation_V5/LoriotProDoc_EN/V22-RRD_Collector_RRD_Manager/V22-A1_Introduction_RRD_EN.htm

[13] Información web general sobre N2RRD: http://en.wikipedia.org/wiki/N2rrd

[14] Documentación y manuales de N2RRD: http://n2rrd.diglinks.com/cgi-bin/trac.fcgi

[15] Documentación y manuales de Nmap: http://nmap.org/

[16] Documentación y manuales de Cacti: http://www.cacti.net

[17] Información web general sobre Cacti: http://es.wikipedia.org/wiki/Cacti

[18] Foros de ayuda y soporte de Cacti: http://forums.cacti.net/

[19] Documentación de Cacti: http://docs.cacti.net/

[20] Información web general sobre Nmap: http://es.wikipedia.org/wiki/Nmap

[21] Información web general sobre SNMP:
http://es.wikipedia.org/wiki/Simple_Network_Management_Protocol

[22] Información web general sobre SNMP:
http://es.kioskea.net/contents/internet/snmp.php3

[22] Información web general sobre sistemas de alimentación ininterrumpida - SAI:
http://es.wikipedia.org/wiki/Sistema_de_alimentaci%C3%B3n_ininterrumpida

[23] Documentación y manuales de Netbotz: http://www.netbotz.com/

[24] Información web general sobre Dell Open Manage:
http://folk.uio.no/trondham/software/check_openmanage.html

[25] Manuales de soporte de Dell Open Manage:
http://support.dell.com/support/edocs/software/svradmin/

[26] Arboles MIB de equipos del fabricante Cisco:
http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

ANEXOS

TÍTULO DEL TFC: Monitorización de la infraestructura técnica de un Centro de Datos real

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

AUTOR: Fadi Ahmad Taki

DIRECTOR EXTERNO: Miquel López Luque

DIRECTOR: Anna Agustí Torra

FECHA: 27 de Junio de 2011

ANEXO A. Linux como sistema operativo

A.1 Introducción

Linux hace referencia a la familia de sistemas operativos UNIX de computadores que usan el núcleo o "*Kernel*" Linux. Se trata de un sistema operativo de libre distribución, lo que quiere decir es que no hace falta adquirir ninguna licencia para poder usarlo y además el sistema viene acompañado del código fuente.

El sistema ha sido diseñado y programado por multitud de programadores alrededor del mundo. Lo forman el núcleo del sistema (kernel) más un gran número de aplicaciones y librerías que hacen posible su utilización.

Linux puede ser instalado en una amplia variedad de equipos informáticos, desde teléfonos móviles, ordenadores, consolas de videojuegos, hasta supercomputadores. Es líder entre sistemas operativos para servidores y actualmente lo utilizan algunos de los supercomputadores más rápidos del mundo.

También es uno de los principales en el mercado de sistemas operativos de equipos de sobremesa, gracias a la extremada seguridad y estabilidad que ofrece, a su velocidad, y su falta de problemas de fragmentación.

Día a día, hay mas disponibilidad de programas y aplicaciones para este sistema, y la calidad de los mismos sigue aumentando. La presencia de Linux en empresas aumenta constantemente por la simplicidad en su manejo y la excelente relación calidad-precio que se consigue.

A.2 Características

A continuación se detallan las características más importantes de GNU/LINUX:

- Multitarea: se trata de la capacidad de ejecutar varios procesos al mismo tiempo. Existen dos tipos de multitareas, la multitarea preventiva, utilizada en Linux, en la cual el sistema operativo es el encargado de ceder tiempo de microprocesador a cada programa asegurando que, todos los programas que estén siendo utilizados, sean ejecutados. La otra multitarea es la cooperativa, utilizada en Windows, donde los programas se ejecutan hasta que deciden que sea otro el que ocupe el procesador.

- Multiusuario: Varios usuarios pueden usar la maquina al mismo tiempo

- Multiplataforma: Linux puede ser utilizado en varias plataformas, tales como Intel, Amiga, Atari, etc.…

- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.

- Escalabilidad: Linux es un sistema operativo que se adapta sobre son los muchos hardwares que le utilizan.

- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente.

- Consolas virtuales múltiples: varias sesiones de "*login*" a través de la consola. Se crean dinámicamente y se puede tener hasta 64.

- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.

- Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.

- Diversos *protocolos de red* incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

A.2.1 Programación

Las distribuciones de Linux soportan gran cantidad de lenguajes de programación. La colección de lenguajes más común es la incluida en la colección de compiladores de GNU (GCC) y en el sistema de compilación de GNU. Los más representativos/utilizados encontramos:

- C/C++: Es el lenguaje por defecto del sistema operativo GNU/Linux. Esto se debe principalmente a la portabilidad de C. El compilador C en este sistema operativo se encuentra bajo el nombre de "*gcc*" (GNU C compiler)

- Java: Las herramientas de desarrollo de Java están disponibles para Linux, y es más, se incluye un soporte en el núcleo de Linux para reconocer de forma automática a las clases de Java y ejecutarlas. Inicialmente las herramientas de desarrollo JDK de SUN fue portado para Linux, ya que se disponía de todo el código fuente para reconstruir JDK en cualquier plataforma con un sistema operativo y unas librerías adecuadas pero de esta forma, se dependía totalmente de SUN, ya que se utilizaba su código fuente. Por ello, se creó el proyecto JOLT, apoyado desde RedHat, y cuyo objetivo es lograr un conjunto de herramientas de libre distribución que permitan el desarrollo de aplicaciones Java.

- PHP: es un lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas.

- Perl: es un lenguaje desarrollado para la manipulación de texto y actualmente es utilizado para un amplio rango de tareas entre las cuales la administración de sistemas, desarrollo web, programación en red y más.

- Perl está instalado por defecto en las distribuciones más populares de GNU/Linux.

- Bash: Es el intérprete de comandos por defecto y predeterminado en la mayoría de las distribuciones de Linux. Se utiliza en la Shell para la administración de sistemas.

A.2.2 RedHat

RedHat es actualmente el proveedor líder mundial de soluciones de código abierto para la empresa. Es una plataforma abierta que ofrece flexibilidad, eficiencia y control y es idónea para una amplia gama de aplicaciones de la infraestructura de las Tecnologías de Infraestructura (TI).

Las infraestructuras de TI actuales son sistemas multiproveedor y multiplataforma y RedHat se ha diseñado de forma que ofrece las innovaciones en hardware más recientes de numerosos fabricantes.

La plataforma RedHat Enterprise Linux es un entorno ideal para el desarrollo de servicios, procedimientos y políticas de centros de datos. Su arquitectura modular, flexible y sólida, además de las herramientas de gestión, ofrecen un mayor control y escalabilidad.

Con tecnología diseñada específicamente para la monitorización, la gestión y la protección de las aplicaciones, RedHat Enterprise Linux se convierte en un host ideal en cualquiera de las principales plataformas de virtualización.

A.3 Instalación del sistema operativo

A.3.1 Introducción

RedHat Enterprise Linux también conocido por sus siglas RHEL es una distribución comercial de Linux desarrollada por RedHat. RedHat Enterprise Linux proporciona a los responsables de servicios de información y directores de informática de una empresa los medios mejorar la flexibilidad operativa en toda su infraestructura informática. Existen varias versiones de RHEL, desde la

versión 1 hasta la 6. Para la elaboración de este trabajo se utilizó la versión 5 (RHEL5). A continuación se detallan los pasos seguidos para realizar la instalación y la configuración del sistema operativo con el entorno de escritorio e infraestructura de desarrollo GNOME.

A.3.2 Instalación

Al tener los archivos de instalación en un Virtual CD-ROM Drive, se configura la BIOS del equipo para que este arranque desde este dispositivo. Una vez hecho esto, se reinicia la maquina y al arrancar se inicia el proceso de instalación.
Al arrancarse el equipo aparece la siguiente pantalla de bienvenida:

Fig. A.1 Pantalla de inicial para la selección del tipo de instalación

Se ofrecen dos opciones:

- Instalación o actualización en modo de interfaz grafica de usuario (pulsar <Intro>).

- Instalación o actualización de la Interfaz de usuario en modo texto (escribir: "*Linux text*" y pulsar <Intro>).

Hay que elegir una de estas dos opciones. En este trabajo se utilizó la instalación en modo grafico.
Una vez seleccionada la opción, se muestra la siguiente pantalla:

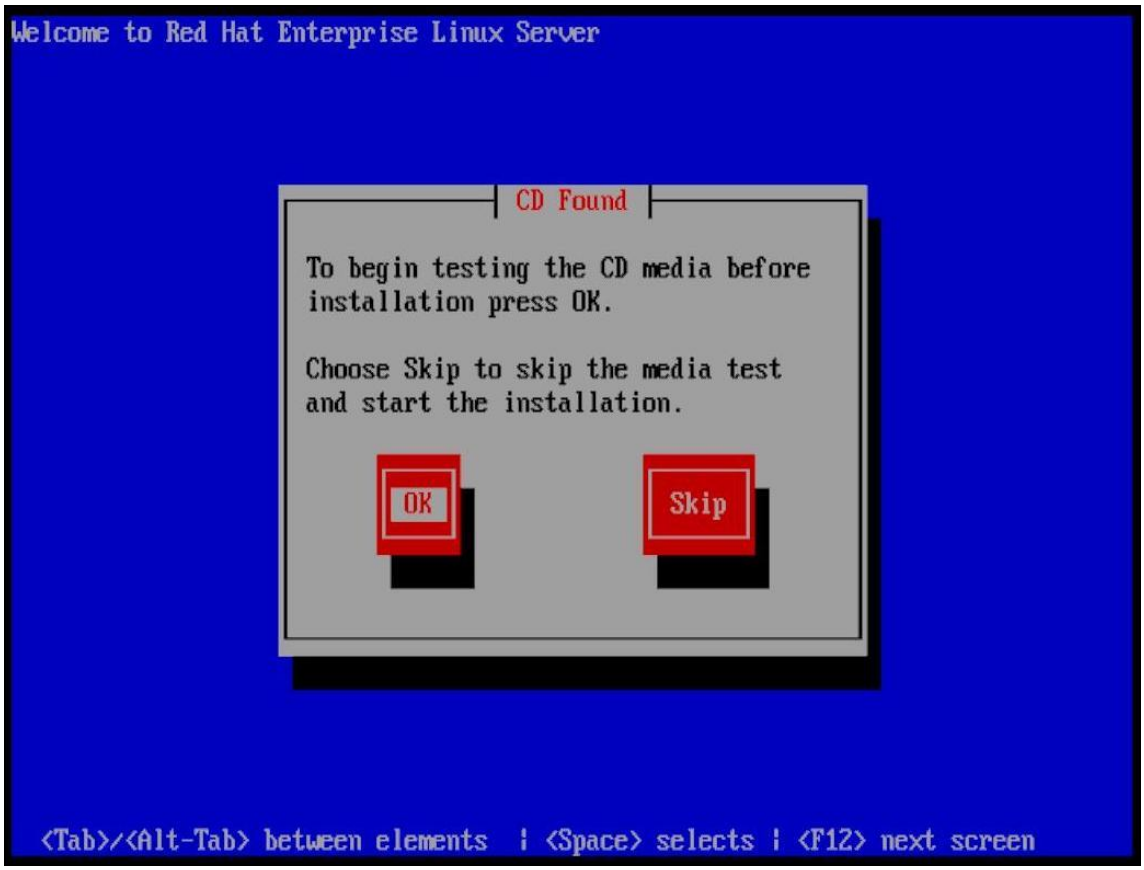


Fig. A.2 Pantalla en la que elegimos si analizar o no el CD de instalación

Esta pantalla indica que se ha encontrado el CD de instalación y ofrece dos opciones:

- Empezar a testear el CD de instalación antes de realizar la instalación (pulsar "*OK*" si queremos elegir esta opción)
- Saltarse el paso anterior e iniciar la instalación (pulsar "*Skip*" si queremos elegir esta opción)

A.3.2.1 Bienvenida a RedHat Linux Enterprise

La pantalla de Bienvenida no pide ninguna información.

Se pulsa el botón "*Next*" (Siguiente) para seguir con la instalación.

Fig. A.3 Pantalla de bienvenida a RedHat Linux Enterprise

A.3.2.2 Selección de idioma

Escoger el idioma predeterminado para el sistema operativo. Se seleccionó el inglés como idioma predeterminado.

Pulsar el botón “*Next*” (Siguiente) para seguir con la instalación.

Fig. A.4 Selección del idioma

A.3.2.3 Configuración de teclado

Seleccionar el tipo de teclado predeterminado para el sistema y que se quiere utilizar durante el proceso de instalación.

Se selecciona el inglés de Estado Unidos como tipo de teclado.

Pulsar el botón “*Next*” (Siguiente) para seguir con la instalación.

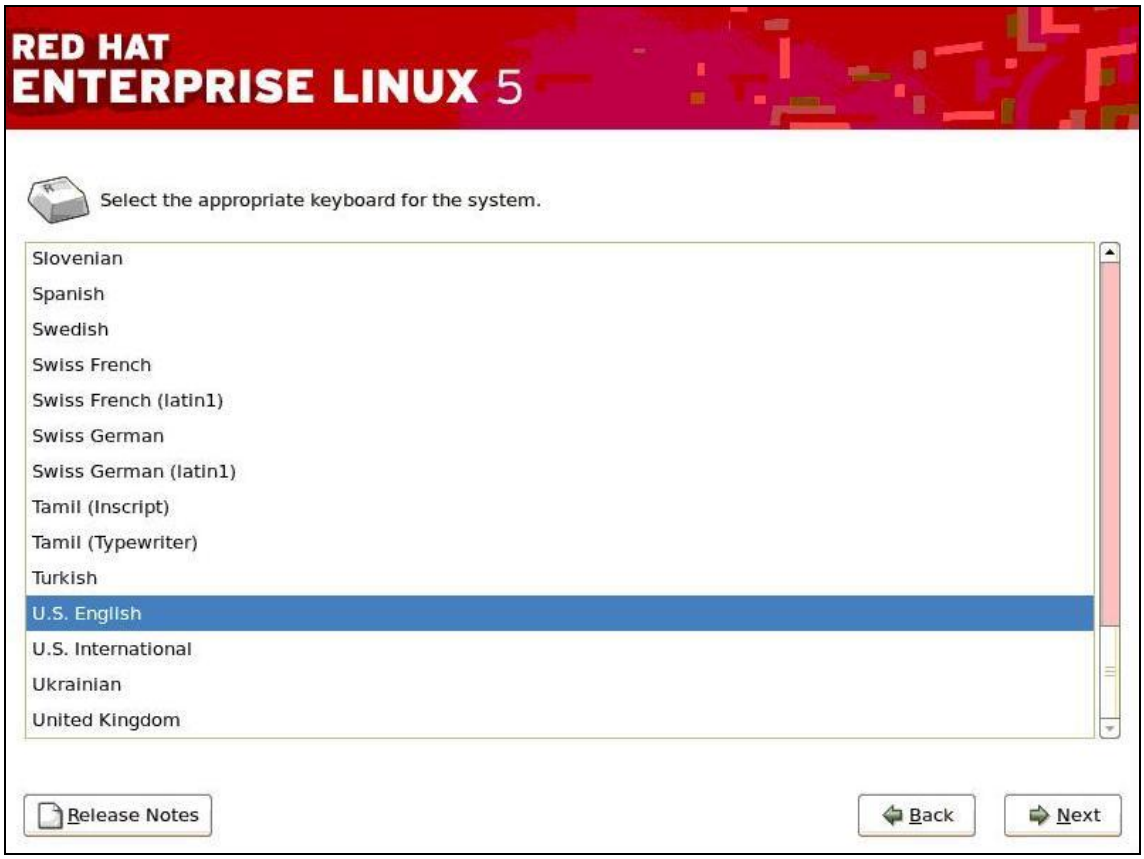


Fig. A.5 Configuración del teclado

Se muestra un mensaje dando la opción de instalar el conjunto completo de paquetes de apoyo incluidos en la suscripción.

Este número determinará el conjunto de la selección de paquetes que está disponible para instalar.

En este trabajo se seleccionó la opción de omitir la introducción del número de serie instalación (Skip entering installation number) de modo que se selecciona la instalación de los paquetes básicos.

Pulsar el botón “*OK*” para seguir con la instalación.

Fig. A.6 Solicitud del número de serie de la instalación

Aparece un mensaje de aviso/ayuda dando información sobre qué conlleva omitir la introducción del número de serie de la instalación:

- Si no se puede localizar el número de la instalación, consultar: http://www.redhat.com/apps/support/in.html.

- Si se omite la introducción del numero de instalación:

  - No se podrá tener acceso a la totalidad de los paquetes incluidos en la suscripción.

  - Se dará lugar a una instalación sin soporte / no certificada por RedHat Linux Enterprise.

  - No se recibirán las actualizaciones de software y seguridad de los paquetes no incluidos en la suscripción.

Pulsar el botón de omitir (Skip) y después el botón “Next” para seguir con la instalación.

Fig. A.7 Información sobre que conlleva saltarse el paso de introducir el número de serie de la instalación

A.3.2.4 Configuración de la partición de disco

La partición consiste en dividir el disco en secciones diferentes, donde cada sección se comporta como si fuera ella misma un disco duro diferente. Se suelen crear particiones en el disco cuando se quiere tener múltiplos sistemas operativos.

En la siguiente pantalla se indica que se va a proceder a hacer la partición del disco y que por defecto se escogerá una capa de partición valida. También existe la posibilidad de crear otra partición.

Se puede elegir entre tres opciones para proceder con la partición:

- Eliminar todas las particiones en dispositivos seleccionados (Incluidas las particiones creadas por otros sistemas operativos, por ejemplo Windows) y crear la disposición predeterminada.

- Eliminar particiones de Linux en los dispositivos seleccionados y crear configuración predeterminada. seleccionando esta opción para remover las particiones de Linux únicamente (esta opción permite eliminar las particiones de Linux únicamente, las creadas por instalaciones de Linux previas).

- Usar el espacio disponible en dispositivos seleccionados y crear la configuración predeterminada. Hay que tener en cuenta que tiene que haber el espacio suficiente en el disco duro para realizar la instalación.

Seleccionar la primera opción (Seleccionando la unidad que se usará para hacer esta instalación.

Fig. A.8 Configuración de la partición de disco duro

Pulsar el botón "*Next*" para seguir con la instalación. Se muestra un mensaje de aviso como el que se ve en la figura a continuación.

Se trata de un aviso indicando que se ha elegido la opción de eliminar todas las particiones del disco duro seleccionado.

Indicar que sí (Yes).

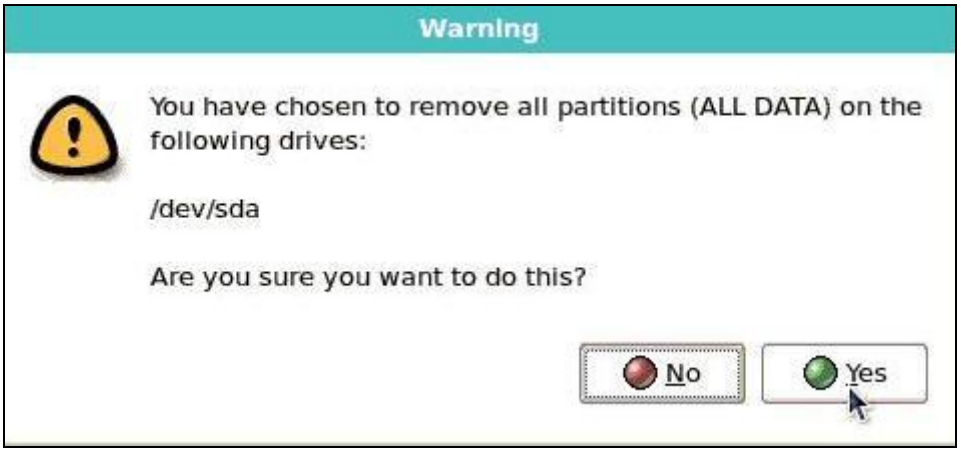


Fig. A.9 Aviso sobre la eliminación de todas las particiones de disco existentes

A.3.2.5 Configuración de red

El programa de instalación detecta automáticamente las tarjetas/interfaces de red que tiene la maquina.

Una vez seleccionada la interface, se selecciona "*Modificar*" para configurar manualmente la dirección IP y la Máscara de Red o utilizar la configuración de IP dinámica (si se tiene un cliente DHCP en red y está configurado para asignar una dirección IP y Mascara de red determinada). Se ha seleccionó la segunda opción.

En relación al nombre del Host, seleccionar la opción de configurarlo de forma automática a través de DHCP.

Fig. A.10 Configuración de las interfaces de red

A.3.2.6 Configuración del huso horario

Se puede seleccionar el huso horario de dos maneras:

- Una seleccionando una ciudad en el mapa que se muestra en la pantalla a continuación.

- Otra opción es elegir una de las ciudades de la lista.

Marcar la opción "*El reloj del sistema usa UTC*" si el sistema está configurado para utilizar UTC. En este trabajo lo está configurado de esta manera.

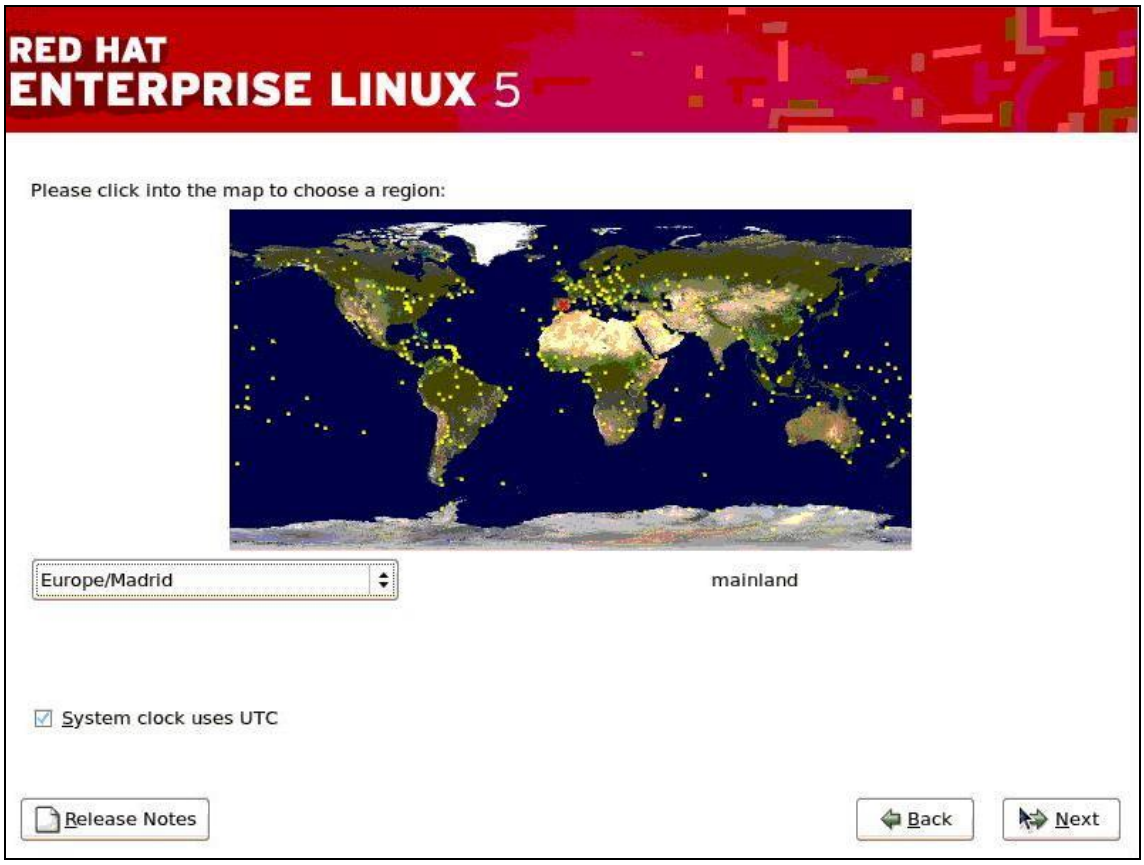


Fig. A.11 Configuración del huso horario

Pulsar el botón "*Next*" para seguir.

A.3.2.7 Configuración de la contraseña de Root

Se trata de la configuración de la cuenta de administrador del sistema. El usuario Root tiene acceso y control completo sobre sistema. Se utiliza para realizar tareas de mantenimiento y administración del sistema. La contraseña de Root es la contraseña administrativa para el sistema.

Los cambios realizados utilizando esta cuenta tienen implicaciones sobre todo el sistema.

Se introduce una contraseña. Puede ser una combinación de letras y números, y que es sensible a las minúsculas y mayúsculas. La contraseña de Root debe tener al menos seis caracteres y no aparecerá en la pantalla cuando se escriba.

La contraseña de Root elegida es: "*calvin*".

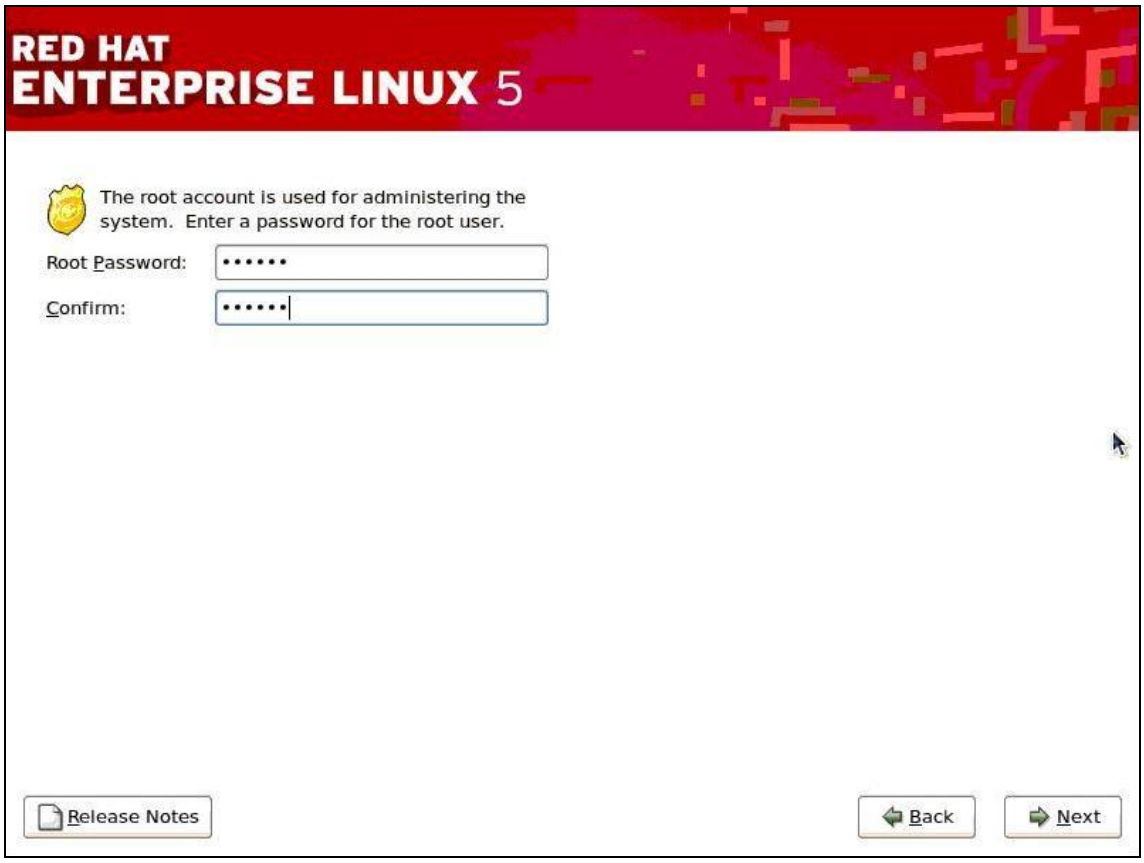


Fig. A.12 Configuración de la contraseña de Root

A.3.2.8 Instalación de paquetes

En este apartado se detalla la instalación de paquetes. En primer lugar aparece una pantalla en la que se muestran los paquetes predeterminados seleccionados para la instalación.

Se hará una instalación personalizada de los paquetes del sistema.

Fig. A.13 Instalación personalizada de paquetes

.3.2.8.1. Entornos de escritorio

Existen dos entornos gráficos entre los cuales se puede elegir. Son interfaces gráficas de usuario que incluyen un panel, un escritorio, iconos de sistema y un gestor de archivos gráficos.

- GNOME
- KDE (K Desktop Environment)

Se ha seleccionado GNOME.

Fig. A.14 Selección del entorno de escritorio

.3.2.8.2. *Aplicaciones*

- Creación y edición: estas herramientas permiten crear documentos en formato Dockbook y convertirlos a HTML, PDF, PostScript y texto.

- Editores: son los conocidos como editores de texto. Estos son programas que nos permiten crear y editar archivos.
  Estos incluyen Emacs y VI.

- Ingeniería y ciencia: este grupo incluye paquetes para realizar cálculos matemáticos y científicos y también trazado, así como conversión de unidades.

- Juegos y entretenimiento: Varias maneras de relajarse y pasar el tiempo libre.

- Internet Grafico: este grupo incluye e-mail, Web y clientes de chat gráficos.

- Gráficos: este grupo incluye paquetes de ayuda a la manipulación y el escaneo de imágenes.

- Oficina/productividad: la aplicación incluye el “Office Suites”, visualizadores de PDF y más.

- Audio y video: desde grabación de CD hasta reproducción de CDs de audio y archivos de multimedia. Este grupo de paquetes nos permite trabajar con audio y video.

- Internet basado en texto: este grupo incluye e-mail, Webs y clientes de chat basados en texto. Estas aplicaciones no requieren en sistema de ventanas X (protocolo que permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la red para éste).

.3.2.8.3. Aplicaciones

- Librerías de desarrollo: los paquetes en este grupo son las librerías básicas necesarias para desarrollar aplicaciones.

- Herramientas de desarrollo: estas herramientas incluyes herramientas de desarrollo básicas tales como automake, gcc, Perl, python y compiladores

- Desarrollo de software de GNOME: instalar estos paquetes con el fin de desarrollar las aplicaciones graficas de GTK+ y GNOME.

- Desarrollo de JAVA: soporte para el desarrollo de programa en el lenguaje de programación JAVA.

- Desarrollo de software KDE: instalar los paquetes para desarrollar las aplicaciones graficas de QT y KDE.

- Desarrollo de software de patrimonio: los paquetes proporcionan soporte de compatibilidad con versiones anteriores.

- Ruby: soporte básico para el lenguaje de programación Ruby.

- Desarrollo de software X: estos paquetes te permiten desarrollar aplicaciones para el sistema de ventanas X.

.3.2.8.4. Servidor

- Servidores de nombres DNS: este grupo de paquetes nos permite ejecutar un servidor DNS (BIND) en el sistema.

- Servidor FTP: estas herramientas nos permiten ejecutar un servidor FTP en el sistema.

- Legado de servidor de red: estos paquetes incluyen servidores para protocolos de red antiguos tales como rsh y telnet.

- Servidor de correo: este grupo de paquetes nos permite configurar un servidor de correo IMAP o SMTP.

- Base de datos MySQL: este grupo de paquetes contiene paquetes útiles para su uso con MySQL.

- Servidores de red: estos paquetes incluyen servidores basados en la red, tales como DHCP, Kerberos y NIS.

- Servidor de noticias: este grupo nos permite configurar el sistema como servidor de noticias.

- Base de datos PostgreSQL: este grupo de paquetes es útil para la utilización con PostgreSQL.

- Soporte para impresión: se han de instalar estas herramientas para habilitar que el sistema pueda imprimir o comportarse como servidor de correo.

- Herramientas de configuración de servidores: este grupo contiene herramientas de configuración para personalizar un servidor Redhat.

- Servidor Web: estas herramientas nos permiten ejecutar un servidor web en el sistema.

- Servidor de archivos Windows: este grupo de paquetes nos permite compartir archivos entre sistemas Linux y sistemas Microsoft Windows.

.3.2.8.5. Sistema Raíz

- Herramientas de administración: este grupo es una colección de herramientas de administración grafica para el sistema, tal como administrar cuentas de usuarios y configurar el hardware del sistema.

- Raíz: este grupo incluye conjunto mínimo de paquetes. Es útil para la creación de cajas pequeñas de router / cortafuegos, por ejemplo.

- Soporte para redes de acceso telefónico.

- Java: soporte para la ejecución de programas escritos en lenguaje de programación Java.

- Legado de soporte de software

- Herramientas de sistema: este grupo es una colección de diversas herramientas para el sistema, tales como el cliente de conexión para comparticiones SMB y herramienta para diversos controladores de tráfico de red.

- Sistema de ventanas X: instalar este grupo de paquetes para utilizar la interfaz grafica de usuario básica (X).

.3.2.8.6. Idiomas

- Español

- Ingles

Una vez finalizada la selección de paquetes que se quieren instalar, se inicia la instalación. Aparece la siguiente pantalla:

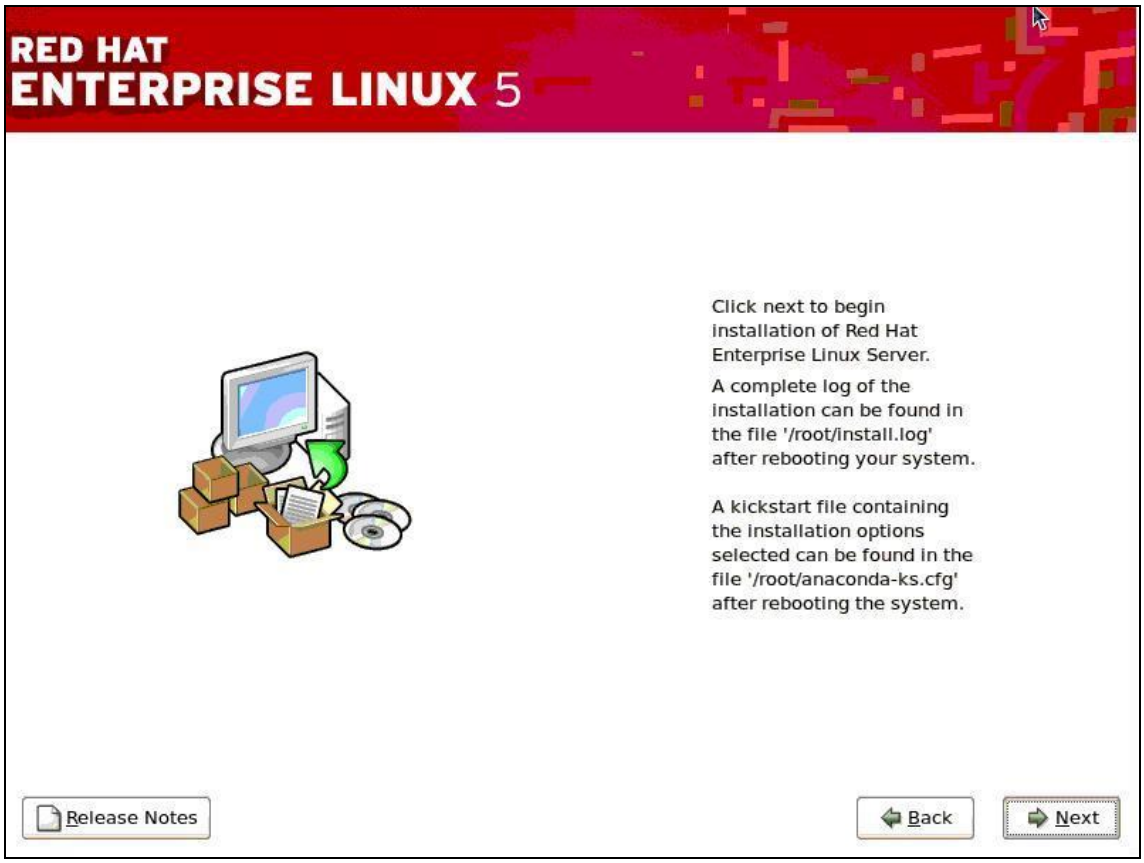


Fig. A.15 Inicio de la instalación del Sistema Operativo con los paquetes que se han seleccionado

Pulsar "*Next*" para continuar. Empieza la instalación según muestra la pantalla a continuación:

Fig. A.16 Evolución de la instalación

Una vez finalizado el proceso, aparece una pantalla con el mensaje de que ha finalizado el proceso de instalación satisfactoriamente y que se debe remover el medio de instalación y pulsar el botón de "*Reboot*" para reiniciar el sistema.

Fig. A.17 Finalización del proceso de instalación

Pulsar "*Reboot*" para re arrancar el sistema y seleccionar el dispositivo de arranque. Una vez hecho este paso, se inicia el sistema y aparece el agente de configuración inicial.

- La primera pantalla es una pantalla de bienvenida que indica que hay algunos pasos que seguir antes de que el sistema esté listo para ser usado.

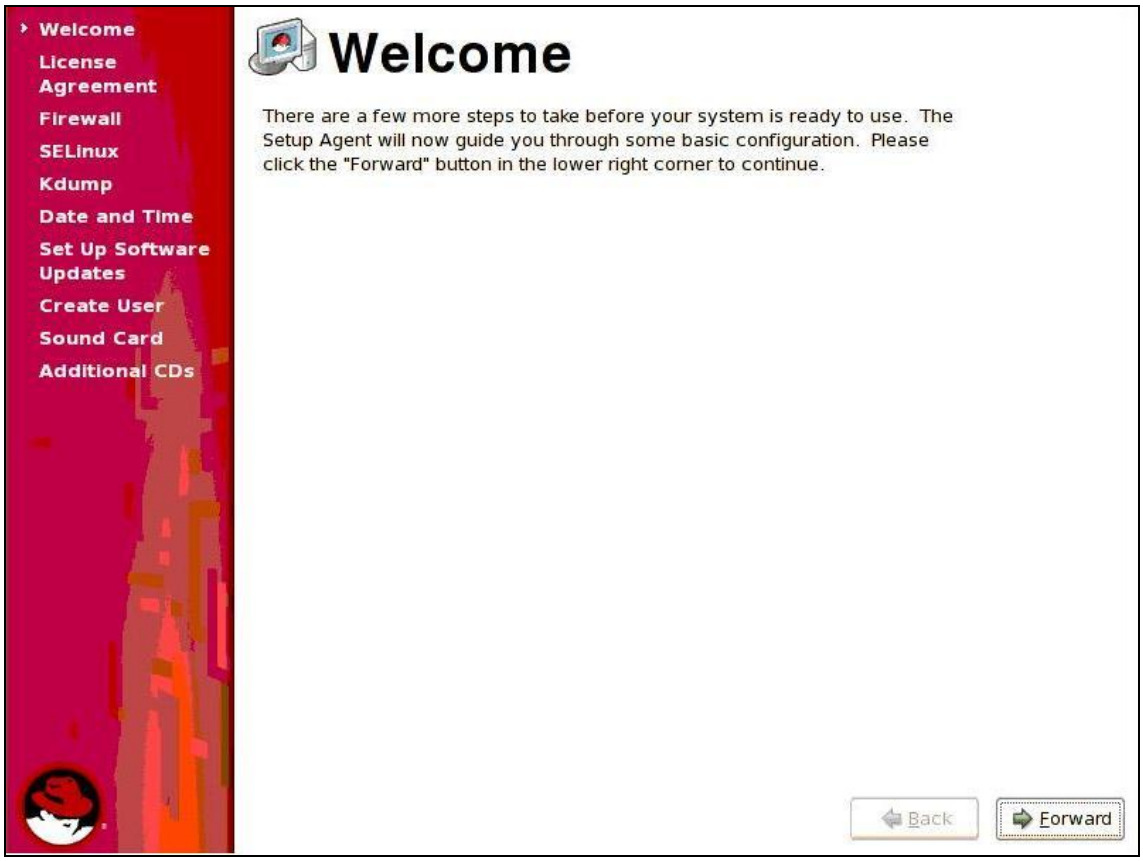


Fig. A.18 Pantalla de bienvenida al sistema

Pulsar “Forward” para seguir.

- El primer paso es aceptar el acuerdo de licencia. Marcar la pestaña “*Yes*” y luego pulsar “*Forward*”.

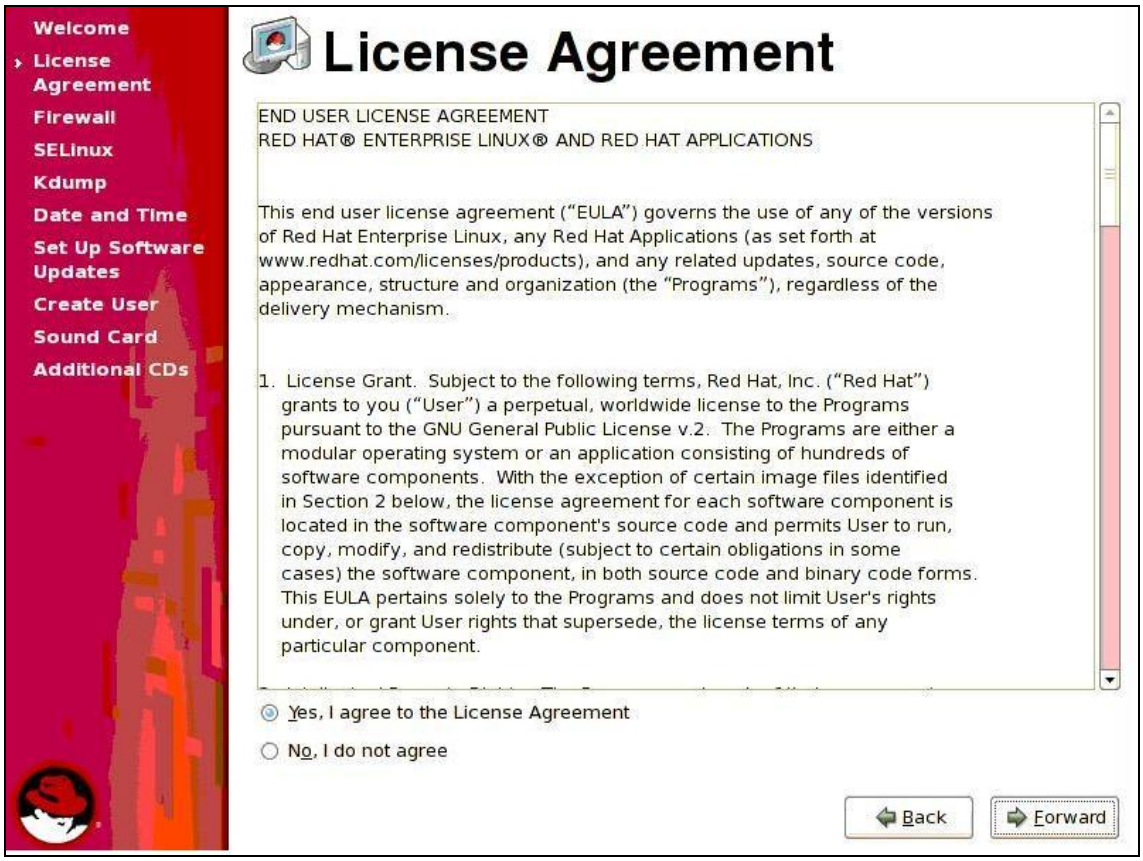


Fig. A.19 Contrato de licencia de RedHat Enterprise Linux

- El siguiente paso es la configuración de un cortafuego. En este trabajo se deshabilitó ya que la red ya tiene configurado un cortafuegos interno.

Fig. A.20 Deshabilitar Cortafuegos

Pulsar “Forward” para seguir.

- Esta opción permite habilitar la opción de utilizar SELinux (Security-Enhanced Linux, en castellano Seguridad Mejorada de Linux). Se trata de una característica de seguridad de Linux que provee una variedad de políticas de seguridad a través del uso de módulos de seguridad en el núcleo Linux. Se puede elegir entre tres opciones seguridad forzada, permisiva, o deshabilitar este formato de seguridad. No se hará servir en este trabajo, por lo tanto se deshabilita esta opción y se pulsa “*Forward*” para seguir adelante.

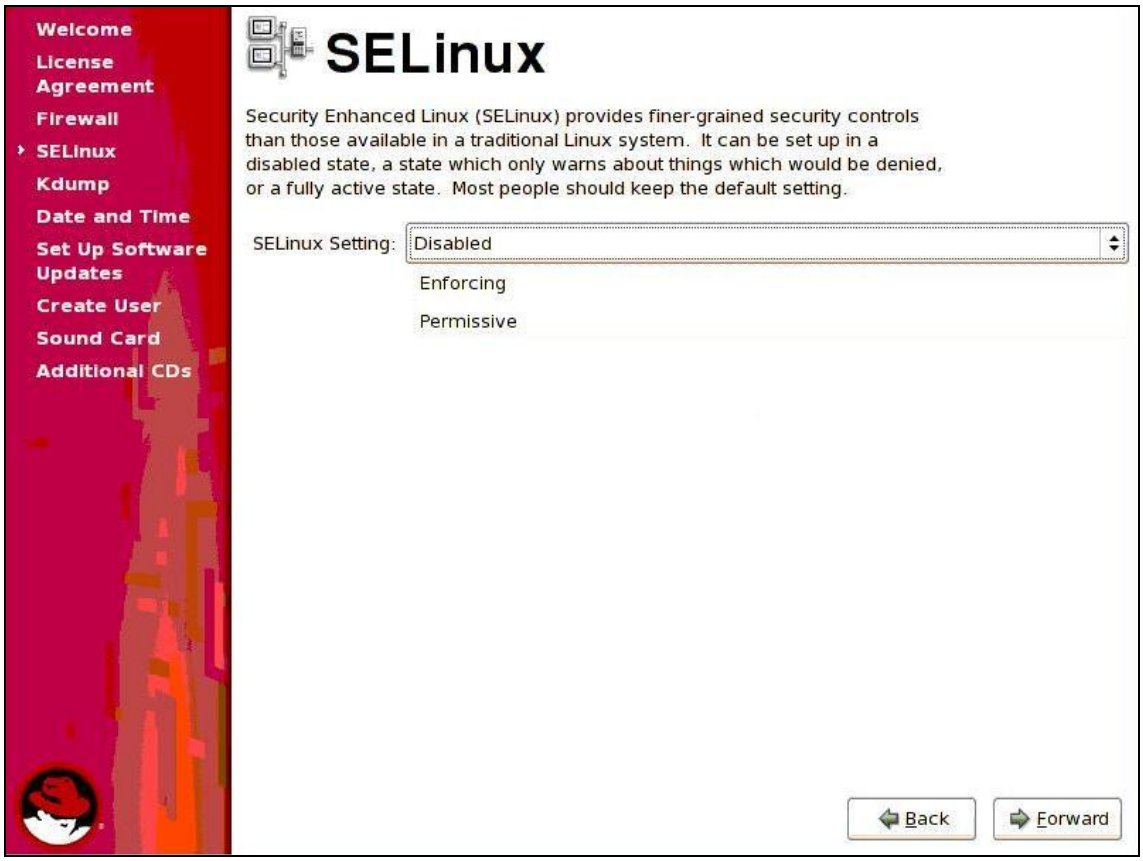


Fig. A.21 Deshabilitar la opción de seguridad mejorada SELinux

- La siguiente opción es el mecanismo Kdump. Se trata de un mecanismo de volcar los datos en caso de que haya problema en el núcleo del sistema operativo.
  No se usará esta opción, por lo tanto no se selecciona “Enable Kdump” y se pulsa “Forward” para avanzar.

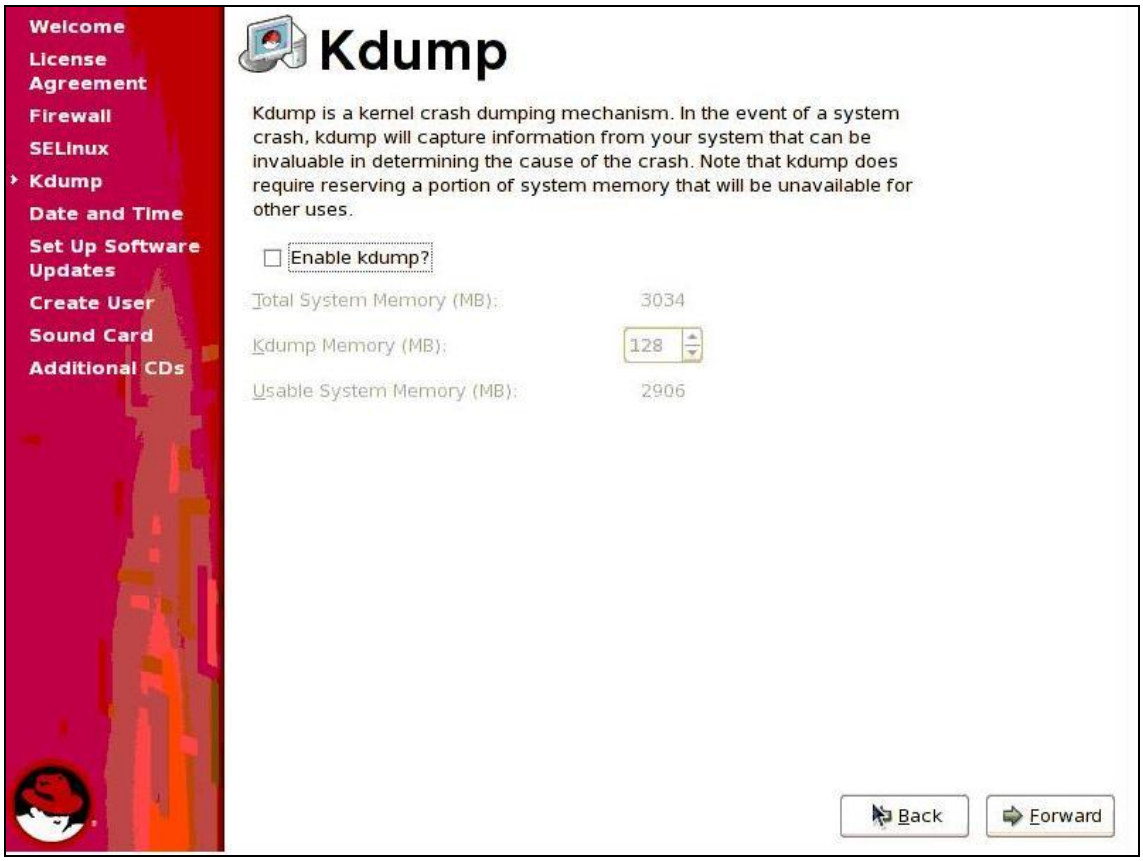


Fig. A.22 Opción del mecanismo Kdump

Se llega a la configuración de fecha y hora del sistema. En este paso existen dos opciones:

- Configuración de la fecha y hora manualmente:

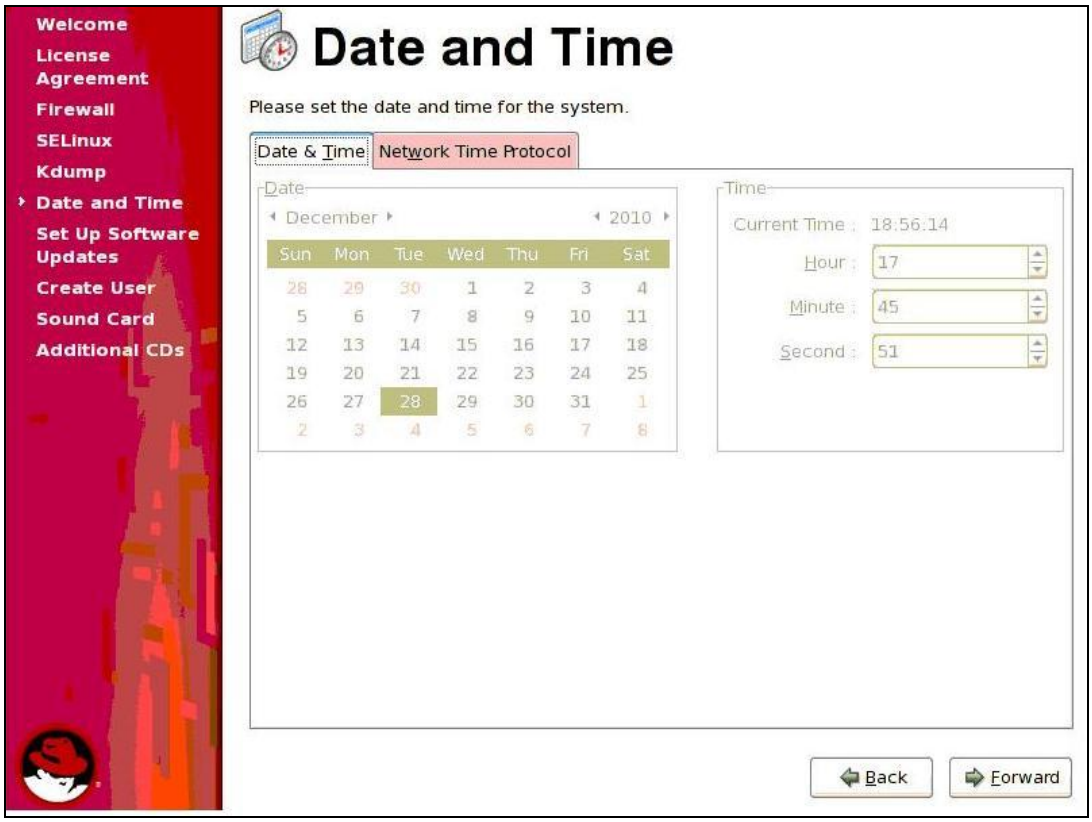


Fig. A.23 Configuración manual de la fecha y hora

- Configuración de fecha y hora automáticamente usando un servidor NTP (Network Time Protocol). Se utilizó esta opción ya que hay un servidor NTP para tener el tiempo del equipo sincronizado con el servidor: "*ntp.esade.edu*".

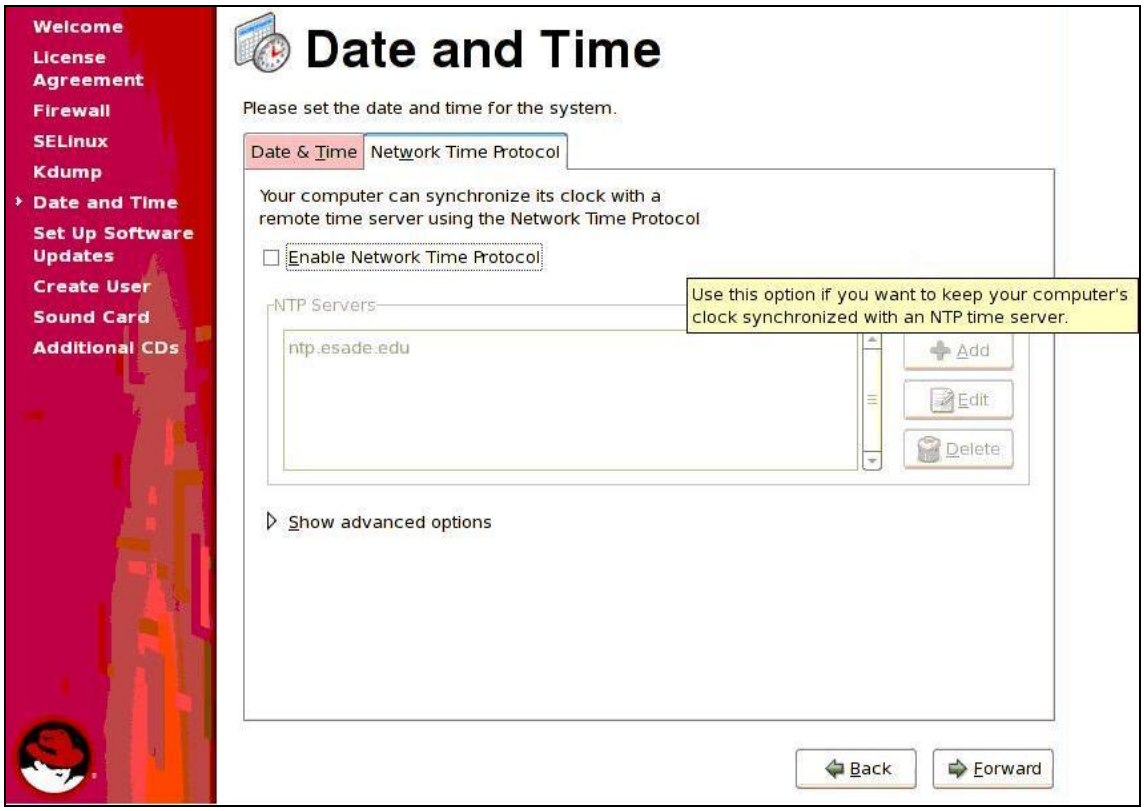


Fig. A.24 Configuración automática de la fecha y hora

- Este paso consiste en establecer las actualizaciones de software. Como se puede ver en la pantalla a continuación, aparece un mensaje de aviso de que no se puede realizar esta tarea ya que el equipo no se encuentra conectado a la red en ese momento, y eso es debido a que aun no se ha configurado la interfaz de la tarjeta de red para que el equipo tenga conexión. Esta configuración se hará más adelante, por lo tanto se salta este paso en este momento pulsando "*Forward*" para seguir con la configuración inicial.

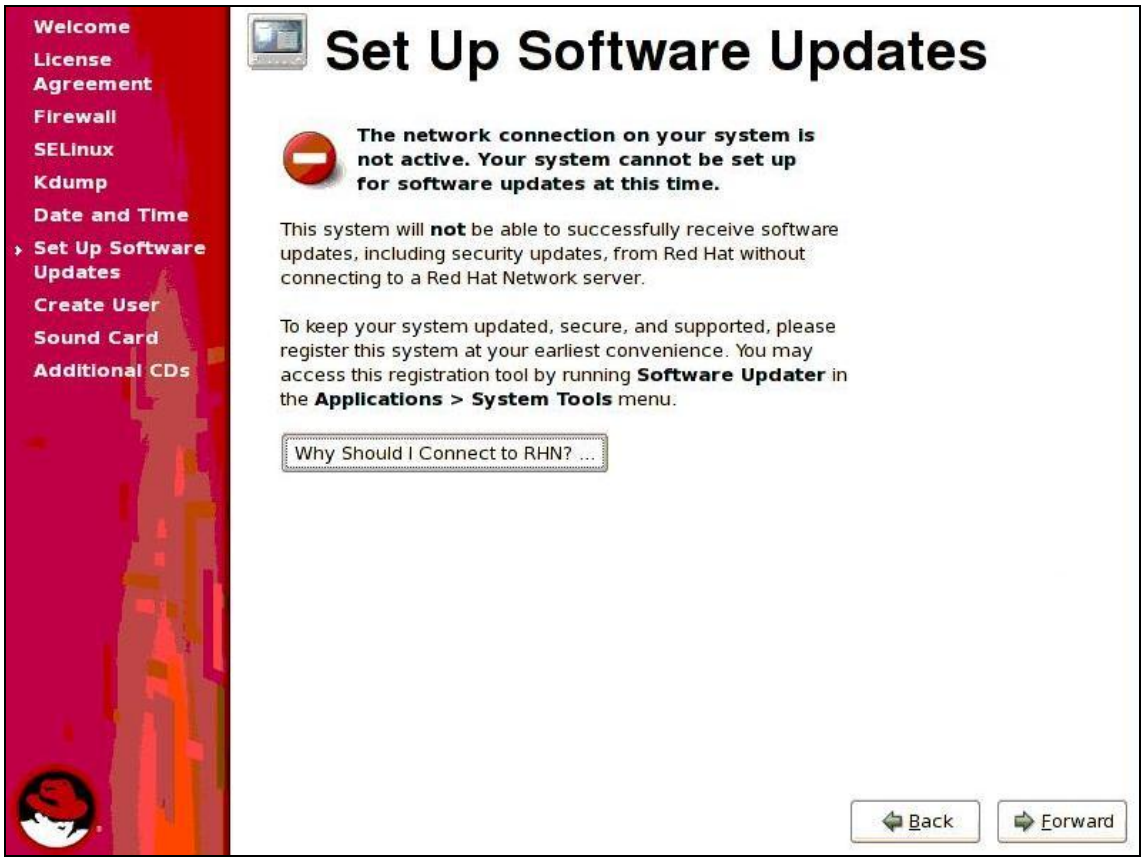


Fig. A.25 Actualización de software

- Se llega a la opción de crear usuario. Es un usuario no administrativo para el uso regular del sistema. Se introducimos el nombre y contraseña que se desea. El nombre de usuario es: "*esade*" y la contraseña: "*esade.edu*"

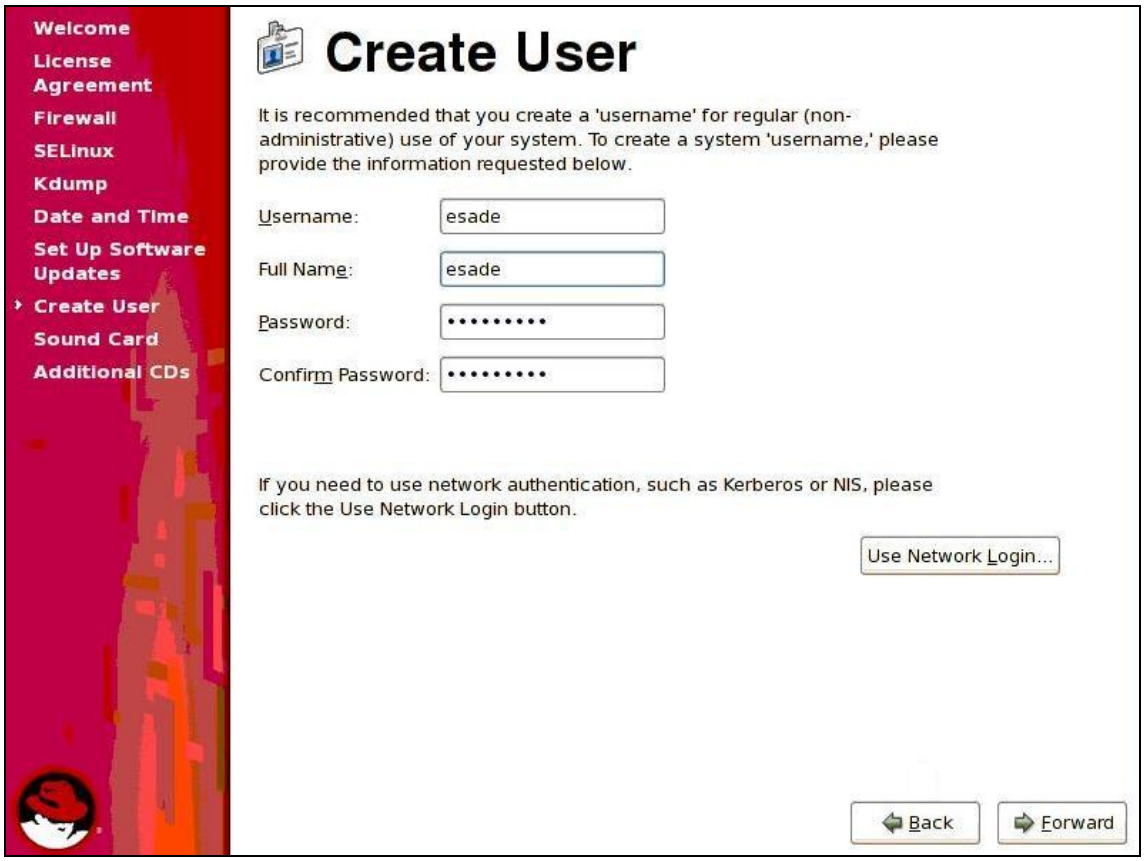


Fig. A.26 Creación de usuario

Pulsar "*Forward*" para avanzar.

- Este paso consiste en configurar la tarjeta de sonido. Indica los pasos a seguir para comprobar el correcto funcionamiento de este dispositivo.

  Como se puede ver en la pantalla a continuación, el sistema no ha detectado ninguna tarjeta de sonido en el sistema y efectivamente, no hay ninguna tarjeta de sonido instalada ya que no hace falta en un servidor de monitorización.

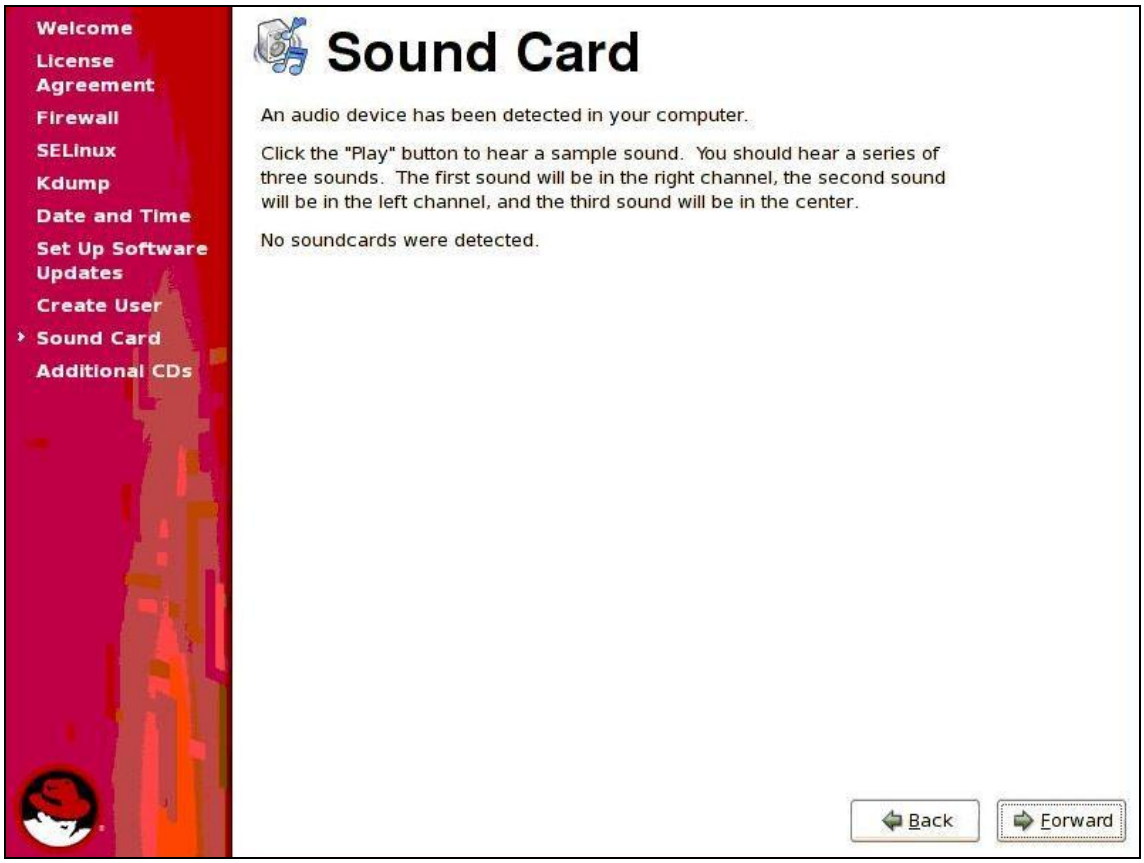


Fig. A.27 Configuración de la tarjeta de sonido

- El último paso es la instalación de software adicional introduciendo los CD que incluyan dichos software. No se instala ningún software adicional.

  Pulsar "*Finish*" para finalizar la configuración inicial del sistema.

Fig. A.28 Instalación de software adicional mediante CDs

A.4 Configuración de red

Este proceso es muy importante, ya que si el servidor no está conectado, como mínimo, a la red local, no se puede monitorizar ningún equipo ni servicio de la red. Normalmente, durante el proceso de instalación, la interfaz de la tarjeta de red se configura automáticamente. En este trabajo no ha sido posible, por lo tanto se configura manualmente una vez la instalación del sistema operativo está hecha y se ha accedido al sistema.

A.4.1 Configuración manual de la red

El nombre de la interfaz Ethernet de nuestro equipo es "*eth0*". Para realizar la configuración se abre una consola de comandos de Linux e se introduce la línea de comandos que se verá a continuación para acceder al directorio donde se encuentran los archivos de configuración de las interfaces Ethernet del equipo.

- Línea de comando: cd /etc/sysconfig/network-scripts.

```
[root@nagios ~]# cd /etc/sysconfig/network-scripts
```

Una vez se está en este directorio, se accede al archivo de configuración de nuestra interfaz:

- Línea de comando: vi ifcfg-eth0

```
# Broadcom Corporation NetXtreme BCM5721 Gigabit Ethernet PCI Express
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.230
NETMASK=255.255.255.0
NETWORK=192.168.0.0
HWADDR=00:1C:23:E1:6B:16
ONBOOT=yes
```

Como se puede ver en la figura anterior, en el archivo se indica:

- El nombre del dispositivo: en este caso "*eth0*".

- Protocolo de asignación de IP: podría ser dinámica o estática, dependiendo si hay un servidor DHCP que asigna direcciones IP a las interfaces de los dispositivos. En la red de ESADE se dota de un servidor de este tipo pero se ha preferido utilizar una asignación estática para fijar el valor de la dirección IP y evitar cualquier problema de asignación que pueda surgir en el futuro.

- Dirección de broadcast: 192.168.0.255

- Dirección IP: 192.168.0.230. Esta dirección IP se ha asignado de manera estática. Lo que se hizo también es reservar esta dirección de IP en el servidor DHCP de modo que el servidor no asigne esta IP a ningún otro equipo. Esto se ha hecho relacionando la dirección IP con la dirección MAC de la tarjeta de red del equipo.

- Mascara de red: 255.255.255.0. Es una dirección que tiene la mismas características que una dirección IP, expresada por medio de valores decimales separados por puntos. Esta máscara tiene un valor de 32 bit y por eso se anota igualmente en forma de cuatro cifras de tres dígitos cada una. La máscara es necesaria para indicar que un equipo pertenece a una red determinada y de este modo permite a las maquinas con la misma mascara de red conectarse directamente entre ellos sin la necesidad de router o un Gateway.

- Dirección de pasarela de Gateway: 192.168.0.0. Es la dirección del equipo informático configurado para dar a los equipos de la red local conectados a él acceso hacia una red exterior.

- Dirección MAC de la tarjeta de red: 00:1C:23:E1:6B:16. Se trata de la dirección física en cuanto a identificar dispositivos de redes. Es un valor que viene definido por fabricante y no puede ser modificado.

Cuando se acaba de introducir estos datos en el archivo de configuración de la interfaz Ethernet, se guarda el archivo y se configura la que sería la puerta de enlace del equipo en la red local con la red exterior. Para hacerlo se utiliza la siguiente comanda:

```
[root@nagios ~]# route add default gw 192.168.0.1 eth0
```

Una vez acabada esta configuración, se reinicia el servicio de red para que se establezcan los valores fijados:

```
[root@localhost network-scripts]# /etc/init.d/network reload
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
```

Fig. A.29 Reiniciar servicio de red

A.4.1.1 Configuración de DNS

Se trata de la configuración del sistema de nombres del dominio que se encarga de traducir identificadores binarios o direcciones IP a un nombre de dominio.
Se ha asociado la dirección IP del equipo, 192.168.0.230, al siguiente nombre: Nagios.esade.edu

A.5 Configuración del servicio VNC

Al haber instalado un sistema operativo con interfaz grafica, se utiliza este servicio para la conexión remota al equipo y poder trabajar. El primer paso es revisar si el servicio VNC está instalado en el equipo. Para ello se accede a la carpeta "*/etc/init.d*" para visualizar todos los servicios que hay instalados en la maquina ejecutando el comando "*ls*". Con la distribución de sistema operativo que se ha instalado, RHEL5, este servicio ya venía integrado en la instalación. En caso de que no hubiera venido instalado directamente, hay que descargarlo e instalarlo.

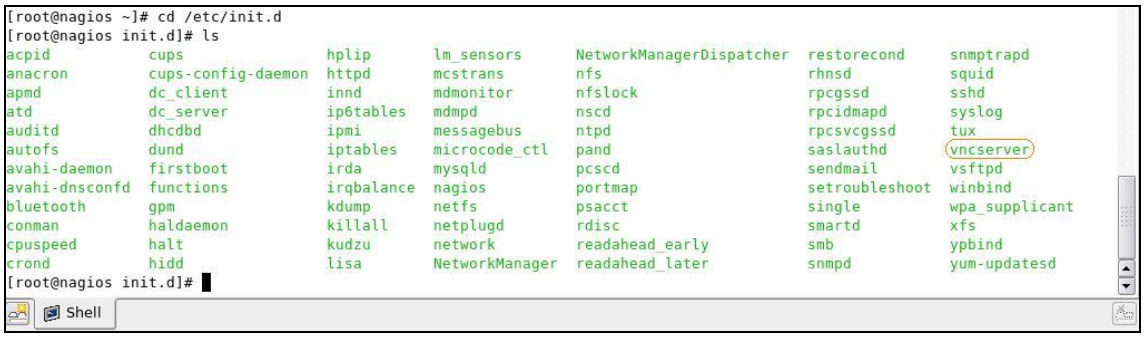


Fig. A.30 Servicios instalados de los cuales dota el equipo, en especial VncServer

Una vez asegurada la presencia de este servicio en la maquina, se pone en marcha con comandarle el siguiente comando:

```
[root@nagios ~]# cd /etc/init.d
[root@nagios init.d]# service vncserver start
```

Al iniciar el servicio por primera vez, pedirá introducir una contraseña de acceso. Se introduce la contraseña que se quiere utilizar para poder acceder a la maquina remotamente. La contraseña de acceso utilizada es: "*esade.edu*". Hay que introducir esta contraseña dos veces para verificarla.

Otra configuración que hay que realizar es la adaptación del archivo de configuración "*vncservers*" para hacer que el servicio "*VncServer*" se ponga en marcha automáticamente al iniciar el equipo.

Se accede al archivo "*/etc/sysconfig/vncservers*" y se descomentan las últimas dos líneas indicando en ellas el usuario bajo el nombre del cual queremos arrancar el servicio (root) y los parámetros geométricos (amplitud y altitud) de la ventana del control remoto.

```
# The VNCSERVERS variable is a list of display:user pairs.
#
# Uncomment the lines below to start a VNC server on display :2
# as my 'myusername' (adjust this to your own).  You will also
# need to set a VNC password; run 'man vncpasswd' to see how
# to do that.
#
# DO NOT RUN THIS SERVICE if your local area network is
# untrusted!  For a secure way of using VNC, see
# <URL:http://www.uk.research.att.com/archive/vnc/sshvnc.html>.

# Use "-nolisten tcp" to prevent X connections to your VNC server via TCP.

# Use "-nohttpd" to prevent web-based VNC clients connecting.

# Use "-localhost" to prevent remote VNC clients connecting except when
# doing so through a secure tunnel.  See the "-via" option in the
# `man vncviewer' manual page.

VNCSERVERS="1:root"
VNCSERVERARGS[1]="-geometry 1024x768"
```

Fig. A.31 Configuración de los parámetros geométricos de la ventana del control remoto.

A.6 Configuración del Proxy

En este apartado se configura el navegador para que se pueda acceder a internet.

Abrir una nueva ventana del navegador y abrir las preferencias del navegador:

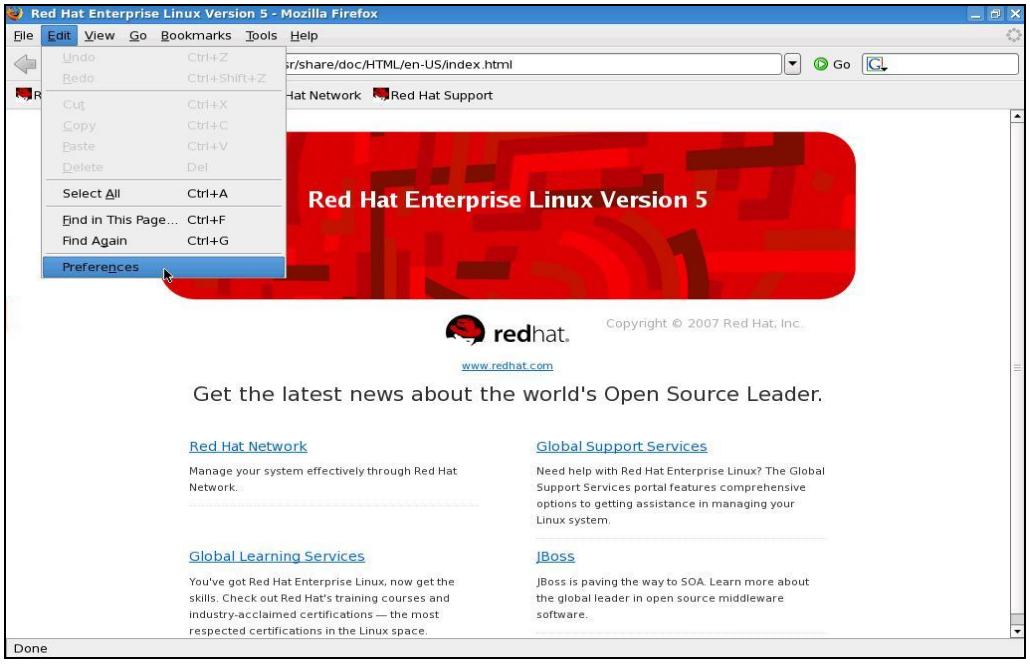


Fig. A.32 Preferencias del navegador web.

Entrar a la configuración de conexiones en las preferencias generales e introducir el nombre del proxy seleccionando la configuración manual del proxy.

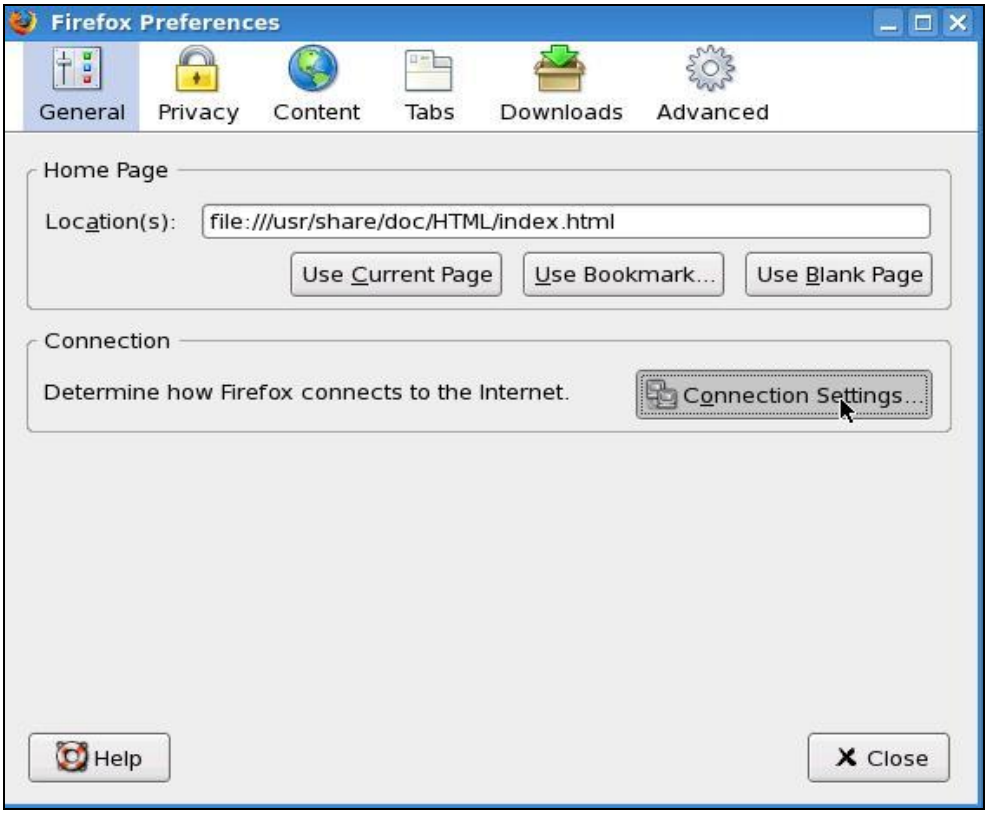


Fig. A.33 Configuración de las conexiones en el navegador.

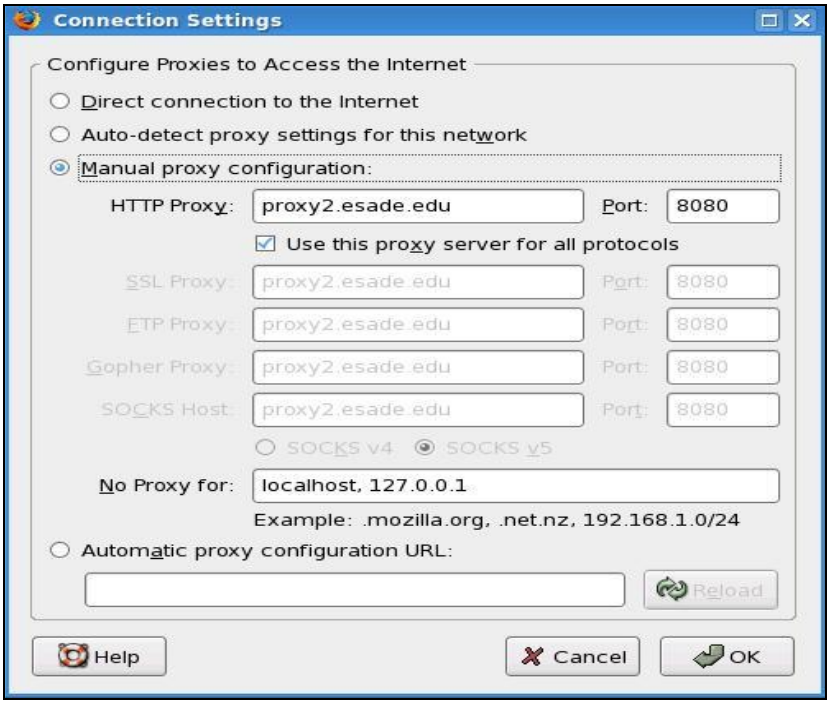


Fig. A.34 Configuración manual del Proxy.

ANEXO B. Instalación de Nagios

B.1 Prerrequisitos

Para realizar este proyecto final de carrera se ha realizado la instalación de Nagios en una máquina en la cual se ha instalado también un sistema operativo Unix, en este caso, RedHat Enterprise Linux 5 bajo la licencia GNU (General Public License).

Como servidor web, se ha optado por utilizar el Apache 2 con el intérprete PHP.

Ya que para el desarrollo de diferentes partes del proyecto se requiere un compilador C, se ha optado por instalar el conjunto de compiladores estándares para Linux, GCC. Además, se ha instalado el conjunto de librerías de desarrollo GD para la generación dinámica de imágenes.

B.2 Instalación de los requisitos

Todos los programas indicados anteriormente se pueden instalar utilizando la herramienta "*yum*" de RedHat Linux

- Instalación del servidor e intérprete php:

```
[root@nagios ~]# yum install httpd php
```

- Instalación del conjunto de compiladores:

```
[root@nagios ~]# yum install gcc glibc glibc-common
```

- Instalación de librerías de desarrollo GD:

```
 [root@nagios ~]# yum install gd gd-devel
```

B.3 Creación de un usuario

Para la instalación y posterior configuración de Nagios es necesario crear un usuario "*nagios*" que pertenezca al grupo "*nagcmd*".

Para crear el usuario y asignarle una contraseña se utilizan los comandos siguientes:

```
[root@nagios ~]# /usr/sbin/useradd -m nagios
[root@nagios ~]# passwd nagios
```

Para generar el grupo (cuyos usuarios tengan acceso completo a la máquina) y añadirle el usuario "*nagios*", hay que ejecutar los siguientes comandos:

```
[root@nagios ~]# /usr/sbin/groupadd nagcmd
[root@nagios ~]# /usr/sbin/usermod -a -G nagcmd nagios
```

B.4 Descarga e instalación del núcleo

Para descargar e instalar el núcleo de Nagios hay que realizar los pasos siguientes:

1. Crear una carpeta donde descargar los archivos de instalación y acceder a ella:

```
[root@nagios ~]# mkdir  /downloads
[root@nagios ~]# cd  /downloads
```

2. Descargar los paquetes fuente de Nagios y descomprimirlos:

```
[root@nagios downloads]# wget
http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.0.tar.gz

[root@nagios downloads]# tar xzf nagios-3.2.0.tar.gz
```

3. Acceder a la carpeta nagios-3.2.3 y ejecutar el script que contiene la configuración de Nagios (indicando el nombre del grupo anteriormente creado):

```
[root@nagios downloads]# cd nagios-3.2.3
[root@localHost nagios-3.2.3]# ./configure --with-command-group=nagcmd
```

   Por defecto Nagios se instala en la ruta es "*/usr/local/Nagios*".

4. Una vez configuradas correctamente las fuentes, se han de compilar usando la herramienta "make" que viene incorporada con la instalación básica de Linux.

```
[root@localHost nagios-3.2.3]# make all
```

5. Si la compilación del programa principal y de las librerías ha finalizado sin ningún error, se puede proceder con la instalación del programa ejecutando la secuencia de comandos siguientes.

   - Instalación del programa principal, Librerías CGI y archivos HTML

```
[root@localHost nagios-3.2.3]# make install
```

   - Instalación del script "init" en el directorio "/etc/rc.d/init.d"

```
[root@localHost nagios-3.2.3]# make install-init
```

- Instalación y configuración de los permisos sobre el directorio que contiene el archivo de comandos externos.

```
[root@localHost nagios-3.2.3]# make install-commandmode
```

- Instalación de los archivos de configuración de muestra en el directorio "*/usr/local/nagios/etc*". (Estos archivos deben ser modificados antes de utilizar Nagios).

```
[root@localHost nagios-3.2.3]# make install-config
```

- Instalación del archivo de configuración de Apache para la interfaz web.

```
[root@localHost nagios-3.2.3]# make install-webconf
```

Con esto quedará instalado el núcleo del sistema.

B.5 Descarga e instalación de plugins

Para descargar e instalar los plugins hay que realizar los pasos siguientes:

1. Acceder a la carpeta "*/downloads*", descargar el fichero de instalación de los plugins y descomprimirlo.

```
[root@nagios ~]# cd  /downloads

[root@nagios downloads]# wget
http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-
1.4.11.tar.gz

[root@nagios downloads]# tar xzf nagios-plugins-1.4.11.tar.gz
```

2. Acceder a la carpeta que se ha creado como resultado de la descompresión y compilar e instalar los plugins:

```
[root@nagios downloads]# cd nagios-plugins-1.4.11
[root@localHost nagios-plugins-1.4.11]# ./configure --with-nagios-
user=nagios --with-nagios-group=Nagios
[root@localHost nagios-plugins-1.4.11]# make
[root@localHost nagios-plugins-1.4.11]# make install
```

3. Añadir Nagios a la lista de servicios que se ejecutan automáticamente al iniciar el sistema:

```
[root@nagios ~]# chkconfig --add nagios
[root@nagios ~]# chkconfig nagios on
```

B.6 Interfaz Web

Para poder acceder a Nagios a través de la interfaz web hay que instalar el archivo de configuración de Nagios para Apache y crear una cuenta de usuario para acceder a la interfaz web:

```
[root@nagios ~]# cd /etc/httpd/conf.d
[root@conf.d]# make install-webconf
[root@nagios ~]# htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

En este caso el usuario creado es "*nagiosadmin*" y la contraseña (que se solicita al ejecutar el comando para la creación del usuario) es "*nagios*".

También es necesario configurar el navegador para poder acceder a la interfaz web. Para el caso del navegador Mozilla Firefox, los pasos a seguir son los siguientes:

1. Abrir un navegador y seleccionar la opción "Edit → Preferencias" en la barra de navegación.

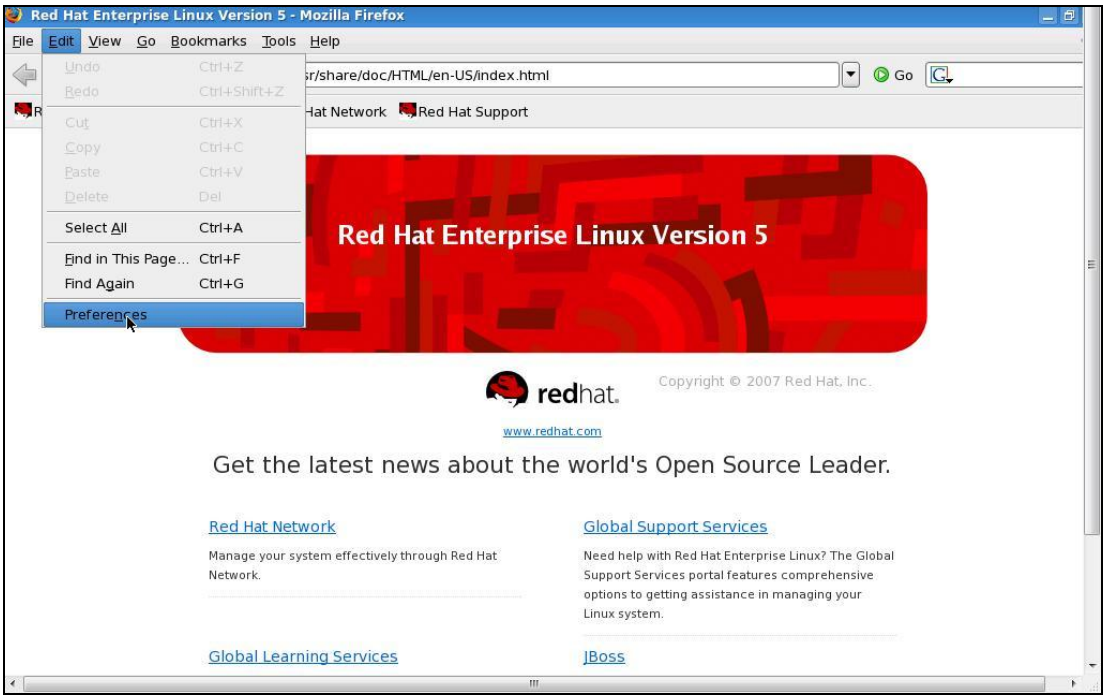


Fig. B.1 Edición de las preferencias del navegador

2. En la ventana que se abre, acceder a la opción de "configuración de conexiones.

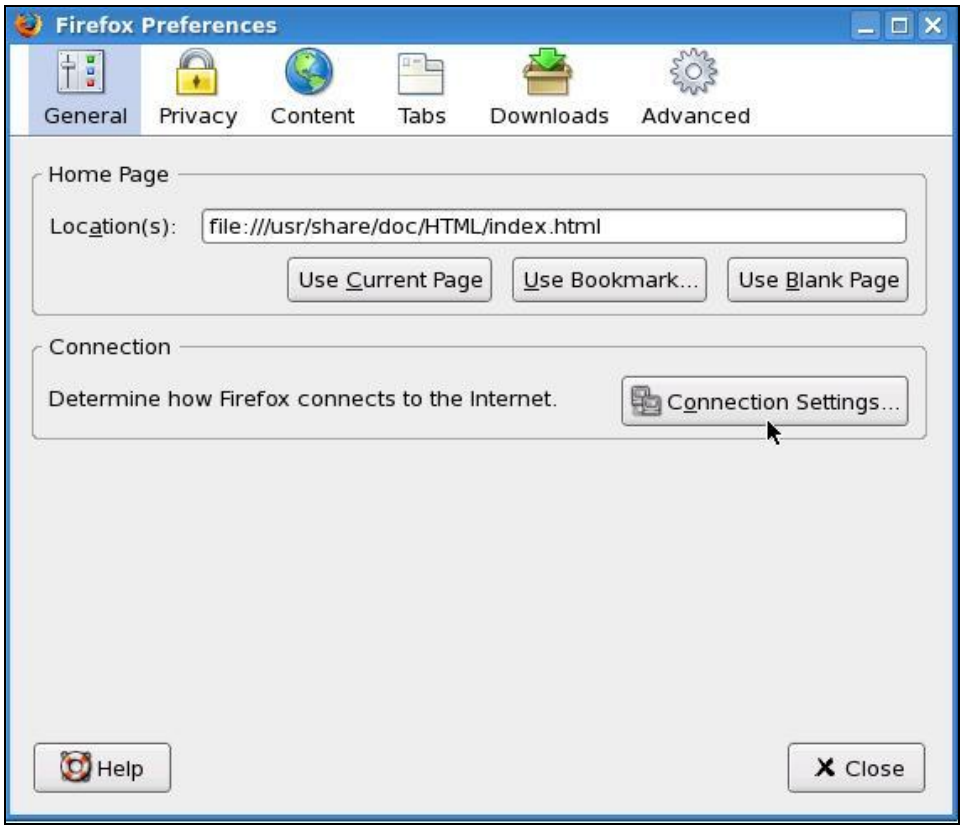


Fig. B.2 Edición de las preferencias del navegador

3. En la siguiente ventana hay que definir el servidor proxy y el puerto por el que se accede.

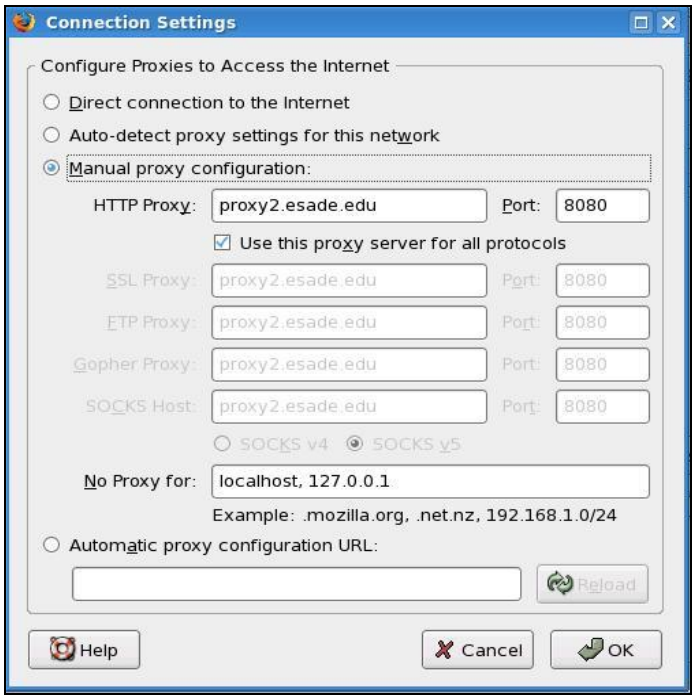


Fig. B.3 Configuración del proxy

Ahora, para acceder a Nagios sólo hay que introducir la dirección URL de la maquina donde está instalado Nagios e introducir el usuario (nagiosadmin) y la contraseña (nagios).

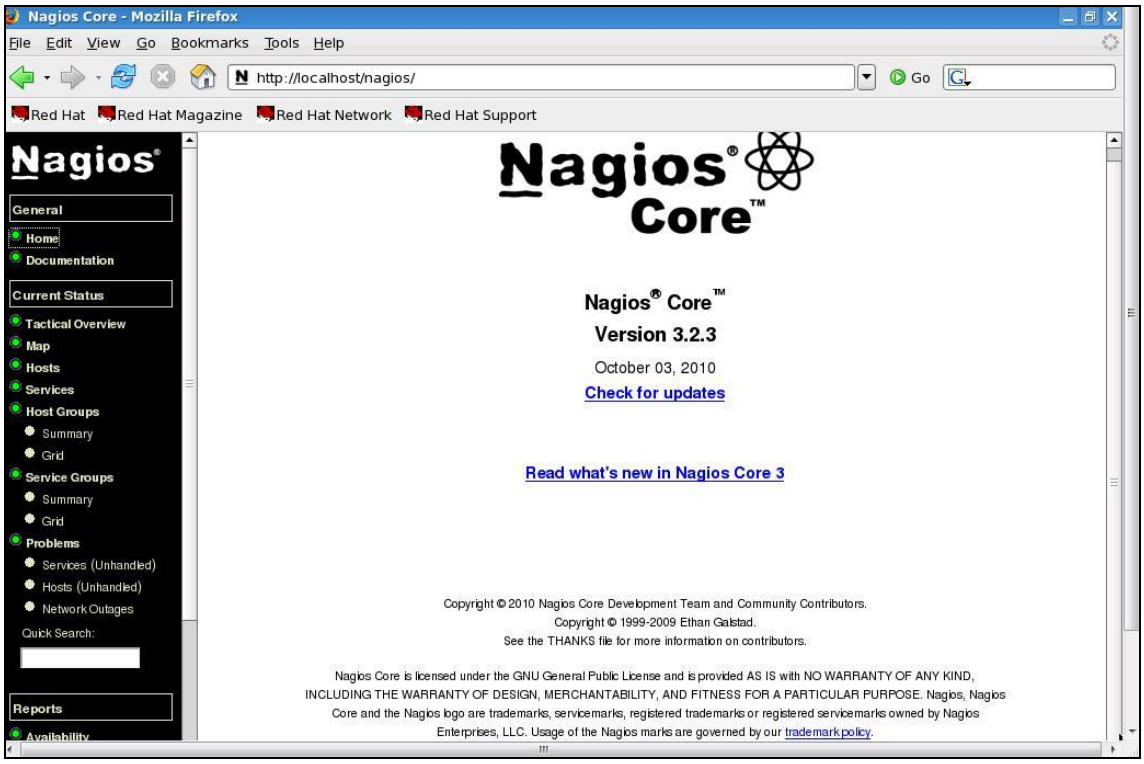


Fig. B.4 Interfaz web de Nagios

ANEXO C. Monitorización de equipos

C.1 SAI

El SAI es el acrónimo de un "*Sistema de Alimentación Ininterrumpida*" que su equivalente en ingles es "*Uninterruptible Power Supply, UPS*". Se trata de un dispositivo que proporciona energía eléctrica tras un fallo de suministro a todos los dispositivos que tenga conectados. Suelen dar energía a equipos llamados cargas críticas, como pueden ser aparatos médicos, industriales y en nuestro caso equipos informáticos que necesitan alimentación continua y de calidad.

Este dispositivo también ejerce otras funcionalidades no menos importantes que la que hemos citado antes, entre ellas la corrección de fallos en el suministro como:

- La sobretensión: la tensión supera el 110% del valor nominal.
- La caída de tensión: la tensión es inferior al 85% del valor nominal.
- Corrección de los picos de tensión.
- Eliminación del ruido eléctrico.
- Corrección de la inestabilidad en la frecuencia.

En el CPD de la Fundación ESADE hay 3 equipos de este tipo. Se ha hecho un estudio completo sobre el árbol MIB de ellos. Una vez descubiertos todos los datos que se pueden recoger de dicho árbol, se ha hecho y se ha presentado al director del proyecto una propuesta sobre los objetos más importantes a mostrar y seguir en la monitorización. Una vez aceptada y seleccionados los objetos que más les interesa monitorizar, se ha procedido a monitorizar los equipos.

C.1.1 Configuración en Nagios

- Definición del equipo

  Se define y se fija el valor de los parámetros de los equipos tipo SAI en el archivo de plantillas "*templates.cfg*".

```
# Define a template for the SAI that we can reuse
define host{
    name                    SAI                     ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
```

```
    check_interval        5                     ; Actively check the host every 5
                                                 minutes
    retry_interval        1                     ; Schedule host check retries at 1
                                                 minute intervals
    max_check_attempts    10                    ; Check each Linux host 10 times (max)
    check_command         check-host-alive      ; Default command to check Linux hosts
    notification_period   24x7                  ; Send notifications at any time
                                                ; Note that the notification_period
                                                 variable is being overridden from
                                                   ; the value that is inherited from
                                                   the generic-host template.
    notification_interval   30                    ; Resend notifications every 30
                                                   minutes
    notification_options    d,r                   ; Only send notifications for
                                                   specific host states
    contact_groups          admins                ; Notifications get sent to the
                                                   admins by default
    register                0                     ; DONT REGISTER THIS DEFINITION
    }
```

Ahora se debe crear una carpeta "*SAI*" en "*/usr/local/nagios/etc/objects*"; y en ella un archivo por cada SAI y en él se define el equipo a monitorizar.

```
# Define a host for the UPS machine we'll be monitoring

define host{
       use             SAI                    ; Inherit default values from a template
       host_name       SAI_Derecha            ; The name we're giving to this host
       alias           SAI_Derecha            ; A longer name associated with the host
       address         192.168.1.136          ; IP address of the host
       hostgroups      SAI
}
```

- Acceso a esta carpeta mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\SAI*".

– Definición de un grupo de equipos de tipo "SAI"

En la carpeta "*/usr/local/nagios/etc/objects/SAI*", se crea otro archivo, "*SAI.cfg*", en el que se define un grupo de hosts al que pertenecerían todos los equipos de este tipo.

```
# Define a hostgroup SAI machines
# All hosts that use the SAI template will automatically be a member of this group

define hostgroup{
       hostgroup_name        SAI     ; The name of the hostgroup
       alias                 SAI     ; Long name of the group
}
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\SAI\SAI.cfg*".

– Definición de un servicio

Al haber definido el servicio genérico en el archivo de plantillas, "*/usr/local/nagios/etc/objects/templates.cfg*", ya no hace falta definir mas servicios ya que los servicios a monitorizar harán uso de este.

- Definición de los servicios a monitorizar

  En el archivo de comandos, "*usr/local/nagios/etc/objects/commands.cfg*", se definen los comandos que cualquier equipo puede utilizar para monitorizar los servicios deseados..Para utilizar un comando, se debe acceder al fichero de configuración del equipo (Al tener varios hosts de este tipo, se ha accedido a uno de ellos, "*usr/local/nagios/etc/objects/SAI/SAI_Derecha.cfg*") e invocar los comandos con el valor deseado.

  Por ejemplo:

  - Definición del comando "*check_ups*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
###############
#check_ups
###############
define command{

        command_name    check_ups
        command_line    snmpget -v1 -c public $HOSTADDRESS$ -Osv $ARG1$
}
```

En la línea de comandos "*command_line*" están los argumentos que Nagios ejecutará: En este caso se usará el comando "*snmpget*" del protocolo SNMP con los parámetros:

- -v1: indica que se trata de SNMP versión 1
- -c public: definición la comunidad SNMP
- La macro "$HOSTADDRESS$" es la dirección IP del host definido;
- -Osv: Se indica que en la respuesta a este comando se muestre solamente el último elemento simbólico del OID y que no se muestre el OID sino solamente el valor de este.
- La macro "$ARG1$" es un argumento mediante el cual se pasan parámetros al comando definido.

A continuación se muestran los servicios monitorizados y se expone un ejemplo:

- Descripción del SAI (Marca y modelo) – "UPS Description"
- Estado de la Batería – "Battery Status"
- Cuanto lleva encendido el equipo – "System Up Time"
- Tiempo que queda en la bacteria – "Ups Seconds On Battery"

- Estimación del tiempo que falta para la recarga completa de la bacteria – “ups Estimated Charge of Battery Remaining”
- Voltaje de la bacteria – “ups Battery Voltage”
- Corriente de la bacteria – “ups Battery Current”
- Frecuencia de entrada – “ups Input Frequency”
- Voltaje de entrada – “ups Input Voltage”
- Corriente de entrada – “ups Input Current”
- Potencia real de entrada – “ups Input True Power”
- Voltaje de salida – “ups Output Voltage”
- Corriente de salida – “ups Output Current”
- Potencia de salida – “ups Output Power”
- Porcentaje de carga de salida – “ups Output Percent Load”
- Frecuencia derivada – “ups Bypass Frequency”
- Corriente derivada – “ups Bypass Current”
- Potencia derivada – “ups Bypass Power”

o Definición del servicio “UPS Description” en el archivo de definición de equipos y servicios a monitorizar del SAI, “*usr/local/nagios/etc/objects/SAI/SAI_Derecha.cfg*”:

```
# Create a service for monitoring the SAI

define service{
       use                     service
       host_name               SAI_Derecha
       service_description     UPS Description
       check_command           check_ups! 1.3.6.1.2.1.1.1.0
}
```

El único parámetro que se ha de pasar a este comando es el OID correspondiente al objeto del cual se quiere obtener el valor, que es equivalente al servicio a monitorizar, en este caso la descripción del SAI. Se trata del valor del argumento “$ARG1$=1.3.6.1.2.1.1.1.0”.

o Acceso a este archivo mediante la siguiente ruta del CD adjunto: “*TFC_Fadi Taki:\Objects\SAI\SAI_Derecha.cfg*”.

– Visualización en la interfaz web

Una vez definido el equipo y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios, tal y como muestra la Fig. C.1.

Service Status Details For Host Group 'SAI'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
SAI_Derecha	Battery Status --- Unknown-1- --- batteryNormal-2- --- batteryLow-3- --- batteryDepleted-4-	OK	11-05-2011 12:29:36	25d 20h 0m 23s	1/3	INTEGER: 2
	PING	OK	11-05-2011 12:36:56	7d 8h 35m 31s	1/3	PING OK - Packet loss = 0%, RTA = 0.97 ms
	System Up Time	OK	11-05-2011 12:34:07	26d 18h 54m 40s	1/3	Timeticks: (2911824932) 337 days, 0:24:09.32
	UPS Description	OK	11-05-2011 12:34:01	25d 20h 3m 26s	1/3	STRING: GALAXY 5000 40 kVA
	Ups Seconds On Battery	OK	11-05-2011 12:32:12	26d 18h 55m 15s	1/3	INTEGER: 0
	ups Battery Current --- The value is in -0.1 Amp DC -	OK	11-05-2011 12:32:47	26d 18h 17m 26s	1/3	INTEGER: 0
	ups Battery Temperature --- The value is in - degrees Centigrade -	OK	11-05-2011 12:31:26	26d 18h 16m 1s	1/3	INTEGER: 21
	ups Battery Voltage --- The value is in -0.1 Volt DC -	OK	11-05-2011 12:31:46	26d 18h 27m 5s	1/3	INTEGER: 4070
	ups Bypass Current --- The value is in - 0.1 RMS Amp -	OK	11-05-2011 12:30:52	26d 18h 0m 26s	1/3	INTEGER: 213
	ups Bypass Frequency --- The value is in - 0.1 Hert -	OK	11-05-2011 12:31:11	26d 18h 0m 16s	1/3	INTEGER: 499
	ups Bypass Percent Load --- The value is in - Percent -	OK	11-05-2011 12:27:32	26d 17h 59m 56s	1/3	INTEGER: 0
	ups Bypass Power --- The value is in - Watts -	OK	11-05-2011 12:27:46	26d 17h 59m 41s	1/3	INTEGER: 1304
	ups Bypass Voltage --- The value is in - RMS Volts -	OK	11-05-2011 12:27:55	26d 17h 59m 33s	1/3	INTEGER: 409
	ups Estimated Charge of Battery Remaining - in Percent -	OK	11-05-2011 12:32:05	26d 18h 25m 22s	1/3	INTEGER: 100
	ups Estimated Minutes Remaining On Battery	OK	11-05-2011 12:33:49	26d 18h 23m 38s	1/3	INTEGER: 27
	ups Input Current--- The value is in - 0.1 RMS Amp -	OK	11-05-2011 12:32:52	26d 18h 14m 35s	1/3	INTEGER: 213
	ups Input Frequency --- The value is in - 0.1 Hert -	OK	11-05-2011 12:34:17	26d 18h 13m 10s	1/3	INTEGER: 499
	ups Input True Power --- The value is in - Watts -	OK	11-05-2011 12:35:23	26d 18h 4m 43s	1/3	INTEGER: 1215
	ups Input Voltage --- The value is in - RMS Volts -	OK	11-05-2011 12:35:43	26d 18h 11m 44s	1/3	INTEGER: 407
	ups Output Current --- The value is in - 0.1 RMS Amp -	OK	11-05-2011 12:34:10	26d 18h 3m 17s	1/3	INTEGER: 196
	ups Output Frequency --- The value is in - 0.1 Hert -	OK	11-05-2011 12:35:36	26d 18h 1m 51s	1/3	INTEGER: 499
	ups Output Percent Load --- The value is in - Percent -	OK	11-05-2011 12:32:09	26d 18h 0m 25s	1/3	INTEGER: 32
	ups Output Power --- The value is in - Watts -	OK	11-05-2011 12:30:38	26d 17h 59m 24s	1/3	INTEGER: 4
	ups Output Voltage --- The value is in - RMS Volts -	OK	11-05-2011 12:28:18	26d 17h 59m 17s	1/3	INTEGER: 401
	Battery Status --- Unknown-1- --- batteryNormal-2- --- batteryLow-3- --- batteryDepleted-4-	OK	11-05-2011 12:28:33	26d 17h 48m 54s	1/3	INTEGER: 2

Fig. C.1 Visualización de los servicios monitorizados de un host SAI en la interfaz web

C.2 Netbotz

El Netbotz es un dispositivo que supervisa las variables ambientales críticas del CPD como la como la temperatura, la humedad, apertura de puertas, detección de fluidos, detección de corrientes de aire, detección de ruidos, de movimiento y grabación de video. Contiene un módulo de cámara y otro de sensores integrados que permiten detectar anomalías en el ambiente. Estos módulos pueden contener hasta 4 cámaras y 17 sensores.

Se ha hecho un estudio sobre el árbol MIB de este dispositivo e igual que en el caso del SAI, se ha presentado una propuesta al director del proyecto sobre los servicios a monitorizar y se han seleccionado los objetos más interesantes.

C.2.1 Configuración en Nagios

– Definición del equipo

Se define y se fija el valor de los parámetros de los equipos tipo Netbotz en el archivo de plantillas "*templates.cfg*".

```
# Define a template for Netbotz that we can reuse
define host{
    name                    Netbotz                 ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
    check_interval          5                       ; Actively check the host every 5
                                                     minutes
    retry_interval          1                       ; Schedule host check retries at 1
                                                     minute intervals
    max_check_attempts      10                      ; Check each Linux host 10 times (max)
    check_command           check-host-alive        ; Default command to check Linux hosts
    notification_period     24x7                    ; Send notifications at any time
                                                    ; Note that the notification_period
                                                    variable is being overridden from
                                                      ; the value that is inherited from
                                                      the generic-host template.
    notification_interval   30                      ; Resend notifications every 30
                                                      minutes
    notification_options    d,r                     ; Only send notifications for
                                                      specific host states
    contact_groups          admins                  ; Notifications get sent to the
                                                      admins by default
    register                0                       ; DONT REGISTER THIS DEFINITION
    }
```

Luego se debe crear una carpeta "*Netbotz*" en "*/usr/local/nagios/etc/objects*"; y en ella un archivo "*Netbotz.cfg*", en el que se define el host a monitorizar.

```
# Define a host for the Netbotz we'll be monitoring

define host{
       use             Netbotz              ; Inherit default values from a template
       host_name       Netbotz              ; The name we're giving to this host
       alias           Netbotz              ; A longer name associated with the host
       address         192.168.1.137        ; IP address of the host
       hostgroups      Netbotz
}
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\Netbotz\Netbotz.cfg*".

– Definición de un grupo de equipos

En este archivo creado antes ("*/usr/local/nagios/etc/objects/Netbotz/Netbotz.cfg"*) se define un grupo de hosts al que pertenecerían todos los equipos de este tipo.

```
# Define a hostgroup for Netbotz
# All hosts that use the Netbotz template will automatically be a member of this group

define hostgroup{
       hostgroup_name        Netbotz        ; The name of the hostgroup
       alias                 Netbotz        ; Long name of the group
}
```

– Servicios a monitorizar

En el archivo de comandos, "*usr/local/nagios/etc/objects/commands.cfg*", se definen los comandos que cualquier equipo puede utilizar para monitorizar los servicios deseados. Para utilizar un comando, se debe acceder al fichero de configuración del equipo, "*usr/local/nagios/etc/objects/Netbotz/Netbotz.cfg*", e invocar los comandos con el valor deseado.

Por ejemplo:

- Definición del comando "*check_netbotz*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
###############
#check netbotz
###############
define command{

               command_name   check_netbotz
               command_line   snmpget -v1 -c public $HOSTADDRESS$ -Ov $ARG1$
       }
```

En la línea de comandos "*command_line*" están los argumentos que Nagios ejecutará: En este caso se usará el comando "*snmpget*" del protocolo SNMP con los parámetros:

- -v1: Indica que se trata de SNMP versión 1.
- -c public: definición de la comunidad SNMP.
- La macro "$HOSTADDRESS$" es la dirección IP del host definido;
- -Ov: Indica que en la respuesta a este comando no se muestre el OID sino solamente el valor de este.

- La macro “$ARG1$” es un argumento mediante el cual se pasan parámetros al comando definido.

A continuación se muestran los servicios monitorizados y se expone un ejemplo:

- Descripción del Netbotz (Marca y modelo) – “Netbotz Description”
- Cuanto lleva encendido el equipo – “System Up Time”
- Temperatura del CPD – “CPD Temperature”
- Humedad en el CPD – “CPD Humidity”
- Rocío en el CPD – “CPD Dew Point”
- Audio en el CPD – “CPD Audio”
- Flujo de aire en el CPD – “CPD AirFlow”
- interruptor de puerta del CPD – “CPD Door Switch”
- Nivel del Pozo – “Nivel Pozo 2”
- Liquido en el suelo – “Liquido Suelo”
- Corriente de entrada – “Corriente Entrada”
- Detección de movimiento por la camera – “Camera Motion”
- Estado del enlace Ethernet – “Ethernet Link status”

- Definición del servicio “*CPD Temperature*” en el archivo de definición de host y servicios a monitorizar del Netbotz, “*usr/local/nagios/etc/objects//Netbotz/Netbotz.cfg*”:

```
define service{
       use                     service
       host_name               Netbotz
       service_description     CPD Temperature -In Celsius-
       check_command           check_netbotz!1.3.6.1.4.1.5528.100.4.1.1.1.7.1095346743
}
```

El único parámetro que se ha de pasar a este comando es el OID correspondiente al objeto del cual se quiere obtener el valor, que es equivalente al servicio a monitorizar, en este caso la temperatura del CPD detectada por el sensor de temperatura del Netbotz. Se trata del valor del argumento “*$ARG1$=1.3.6.1.4.1.5528.100.4.1.1.1.7.1095346743*”.

– Visualización en la interfaz web

Una vez definido el equipo y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios, tal y como muestra la Fig. C.2.

Service Status Details For Host Group 'Netbotz'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Netbotz	CPD AirFlow -In ft/min-	OK	12-05-2011 14:40:37	26d 20h 36m 43s	1/3	STRING: "7.619998"
	CPD Audio -In Celcius-	OK	12-05-2011 14:40:47	26d 20h 36m 33s	1/3	STRING: "10.000000"
	CPD Dew Point -In Celcius-	OK	12-05-2011 14:47:09	26d 20h 30m 11s	1/3	STRING: "11.100000"
	CPD Door Switch	OK	12-05-2011 14:40:54	26d 20h 36m 26s	1/3	STRING: "Closed"
	CPD Humidity -In Percent-	OK	12-05-2011 14:45:36	26d 20h 31m 44s	1/3	STRING: "46.000000"
	CPD Temperature -In Celcius-	OK	12-05-2011 14:42:39	26d 20h 34m 41s	1/3	STRING: "23.000000"
	Camera Motion	OK	12-05-2011 14:41:48	26d 20h 35m 32s	1/3	STRING: "No Motion"
	Corriente Entrada	OK	12-05-2011 14:41:28	26d 20h 35m 52s	1/3	STRING: "220"
	Ethernet Link status	OK	12-05-2011 14:42:23	26d 20h 34m 57s	1/3	STRING: "Up"
	Fluid Detector	OK	12-05-2011 14:41:54	26d 20h 35m 26s	1/3	STRING: "220"
	Liquido Suelo	OK	12-05-2011 14:42:16	26d 20h 35m 4s	1/3	STRING: "Seco"
	Netbotz Description	OK	12-05-2011 14:42:32	26d 20h 34m 48s	1/3	STRING: Linux netbotz02F2EC 2.4.26 #1 Tue Feb 13 12:38:20 CST 2007 ppc
	Nivel Pozo 2	OK	12-05-2011 14:42:09	26d 20h 35m 11s	1/3	STRING: "vacio"
	PING	OK	12-05-2011 14:42:20	0d 4h 20m 0s	1/3	PING OK - Packet loss = 0%, RTA = 0.37 ms
	System Up Time	OK	12-05-2011 14:41:31	26d 20h 35m 49s	1/3	Timeticks: (2744373212) 317 days, 15:15:32.12

Fig. C.2 Visualización de los servicios monitorizados de un host Netbotz en la interfaz web

C.3 Equipos Dell – Dell OpenManage

La mayoría de los servidores que hay en el CPD, para no decir todos, son del fabricante Dell. Este fabricante ha desarrollado una herramienta, o mejor dicho, una aplicación para la gestión de sistemas. Sus ventajas son las siguientes:

- Simplificar la monitorización de un servidor mediante líneas de comandos o también mediante una interfaz grafica de usuario vía web.
- Obtener información sobre la configuración del sistema, la salud y el rendimiento.
- También se puede obtener información sobre los servicios corriendo en tiempo real, para visualizar si alguno tiene problemas o si requieren operaciones de recuperación remotamente.
- Utilizar un diagnostico en line para ayudar al aislamiento de problemas, o reiniciar y apagar el equipo.

Se Ha instalado un plugin de Dell OpenManage en Nagios que da la capacidad de supervisar el estado general de los equipos con Hardware de Dell mediante

la interfaz web de Nagios y permite obtener información sobre la temperatura del equipo, carga de la CPU, estado de la ventilación, etc.…

C.3.1 Proceso de descarga e instalación

1. Descargar y descomprimir el paquete que contiene el plugin necesario de la siguiente URL:

http://exchange.nagios.org/components/com_mtree/attachment.php?link_id=323&cf_id=24

```
[root@nagios downloads]# tar xzf check_openmanage-3.6.7.tar.gz
```

2. Acceder a la carpeta que se ha creado como resultado de la descompresión:

```
[root@nagios downloads]# cd  / check_openmanage-3.6.7
```

3. Existen varios ficheros en esta carpeta. La forma más sencilla para realizar la instalación es la copia del plugin "*check_openmanage*" a la carpeta de plugins de Nagios, "*/usr/local/nagios/libexec/*"

```
[root@nagios downloads]# cp check_openmanage
/usr/local/nagios/libexec/
```

C.3.2 Configuración en Nagios

– Definición del equipo

Se define y se fija el valor de los parámetros de los equipos tipo Dell en el archivo de plantillas "*templates.cfg*".

```
# Define a template for Dell Machines that we can reuse
define host{
    name                    Dell                    ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
    check_interval          5                       ; Actively check the host every 5
                                                     minutes
    retry_interval          1                       ; Schedule host check retries at 1
                                                     minute intervals
    max_check_attempts      10                      ; Check each Linux host 10 times (max)
    check_command           check-host-alive        ; Default command to check Linux hosts
    notification_period     24x7                    ; Send notifications at any time
                                                    ; Note that the notification_period
                                                     variable is being overridden from
                                                       ; the value that is inherited from
                                                       the generic-host template.
    notification_interval    30                       ; Resend notifications every 30
                                                       minutes
    notification_options     d,r                      ; Only send notifications for
                                                       specific host states
```

```
    contact_groups          admins                  ; Notifications get sent to the
                                                    admins by default
    register                0                       ; DONT REGISTER THIS DEFINITION
    }
```

Luego se debe crear una carpeta "*dell_server*" en "*/usr/local/nagios/etc/objects*", y en ella un archivo en el que se define el equipo a monitorizar.

```
# Define a host for the Dell Server machine we'll be monitoring

define host{
       use             dell_server           ; Inherit default values from a template
       host_name       dell_server_1         ; The name we're giving to this host
       alias           dell_server_1         ; A longer name associated with the host
       address         192.168.1.86          ; IP address of the host
       hostgroups      dell_servers
}
```

- Definición de un grupo de equipos

  En este archivo creado antes ("*/usr/local/nagios/etc/objects/dell_server/dell_server_1.cfg*) se define un grupo de hosts al que pertenecerían todos los equipos de este tipo

```
# Define a hostgroup for Dell Server machines
# All hosts that use the Dell template will automatically be a member of this group

define hostgroup{
       hostgroup_name        dell_servers          ; The name of the hostgroup
       alias                 Dell Servers          ; Long name of the group
}
```

- Definición de los servicios a monitorizar

  En el archivo de comandos, "*usr/local/nagios/etc/objects/commands.cfg*", se definen los comandos que cualquier equipo puede utilizar para monitorizar los servicios deseados. Para utilizar un comando, se debe acceder al fichero de configuración del equipo "*usr/local/nagios/etc/objects/dell_server/dell_server_1.cfg*" e invocar los comandos con el valor deseado.

  Por ejemplo:

  - Definición del comando "*check_openamanage*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
###############
#check openamange
###############
define command{

              command_name check_openmanage
              command_line $USER1$/check_openmanage -H $HOSTADDRESS$ $USER7$ $ARG1$
       }
```

  En la línea de comandos "*command_line*" están los argumentos que Nagios ejecutará:

- La macro “$USER1$” es la ruta hacia el directorio donde se encuentra el plugin.
- La macro “$HOSTADDRESS$” es la dirección IP del host definido.
- La macro $USER7$ es la comunidad (-C <community>).
- La macro “$ARG1$” es un argumento mediante el cual se pasan parámetros a la herramienta definida.

A continuación se muestran los servicios monitorizados y se expone un ejemplo:

- Estado del servidor – “Dell OpenManage Status”
- Temperatura del servidor – “Dell OpenManage Temperature”
- Voltaje del equipo y sus sensores – “Dell OpenManage Voltage”
- Estado de los ventiladores – “Dell OpenManage Fans”
- Carga de la CPU – “Dell OpenManage CPU”
- Estado de la potencia – “Dell OpenManage Power”
- Estado de la memoria – “Dell OpenManage Memory”
- Analisis general del hardware del equipo – “Dell OpenManage Chassis”

- Definición del servicio “*Dell OpenManage Temperature*” en el archivo de definición de equipos y servicios a monitorizar del servidor Dell, “*usr/local/nagios/etc/objects/dell_server/dell_server_1.cfg*”:

```
define service{
       use                     service
       host name               dell server 1
       service_description     Dell OpenManage Temperature
       check_command           check_openmanage! -C deesa4845 --only temp -d
       }
```

Se indica la comunidad SNMP, en este caso “*deesa4845*” y que solamente se revise la temperatura con la opción “*’’only temp*”. Con la opción “’*-d*” se muestran mensajes de salida sobre los componentes comprobados, en este caso la temperatura, con sus respectivas alertas de estado.

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\dell_server\dell_server_1.cfg*".

– Visualización en la interfaz web

Una vez definido el equipo y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios, tal y como muestra la Fig. C.3.

Service Status Details For Host Group 'Netbotz'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Netbotz	CPD AirFlow -In ft/min-	OK	12-05-2011 14:40:37	26d 20h 36m 43s	1/3	STRING: "7.619998"
	CPD Audio -In Celcius-	OK	12-05-2011 14:40:47	26d 20h 36m 33s	1/3	STRING: "10.000000"
	CPD Dew Point -In Celcius-	OK	12-05-2011 14:47:09	26d 20h 30m 11s	1/3	STRING: "11.100000"
	CPD Door Switch	OK	12-05-2011 14:40:54	26d 20h 36m 26s	1/3	STRING: "Closed"
	CPD Humidity -In Percent-	OK	12-05-2011 14:45:36	26d 20h 31m 44s	1/3	STRING: "46.000000"
	CPD Temperature -In Celcius-	OK	12-05-2011 14:42:39	26d 20h 34m 41s	1/3	STRING: "23.000000"
	Camera Motion	OK	12-05-2011 14:41:48	26d 20h 35m 32s	1/3	STRING: "No Motion"
	Corriente Entrada	OK	12-05-2011 14:41:28	26d 20h 35m 52s	1/3	STRING: "220"
	Ethernet Link status	OK	12-05-2011 14:42:23	26d 20h 34m 57s	1/3	STRING: "Up"
	Fluid Detector	OK	12-05-2011 14:41:54	26d 20h 35m 26s	1/3	STRING: "220"
	Liquido Suelo	OK	12-05-2011 14:42:16	26d 20h 35m 4s	1/3	STRING: "Seco"
	Netbotz Description	OK	12-05-2011 14:42:32	26d 20h 34m 48s	1/3	STRING: Linux netbotz02F2EC 2.4.26 #1 Tue Feb 13 12:38:20 CST 2007 ppc
	Nivel Pozo 2	OK	12-05-2011 14:42:09	26d 20h 35m 11s	1/3	STRING: "vacio"
	PING	OK	12-05-2011 14:42:20	0d 4h 20m 0s	1/3	PING OK - Packet loss = 0%, RTA = 0.37 ms
	System Up Time	OK	12-05-2011 14:41:31	26d 20h 35m 49s	1/3	Timeticks: (2744373212) 317 days, 15:15:32.12

Fig. C.3 Visualización de los servicios monitorizados de un host Dell en la interfaz web

C.3.3 Graficas en Cacti

Para este dispositivo se visualizan las graficas de la temperatura del equipo y el estado de los ventiladores.

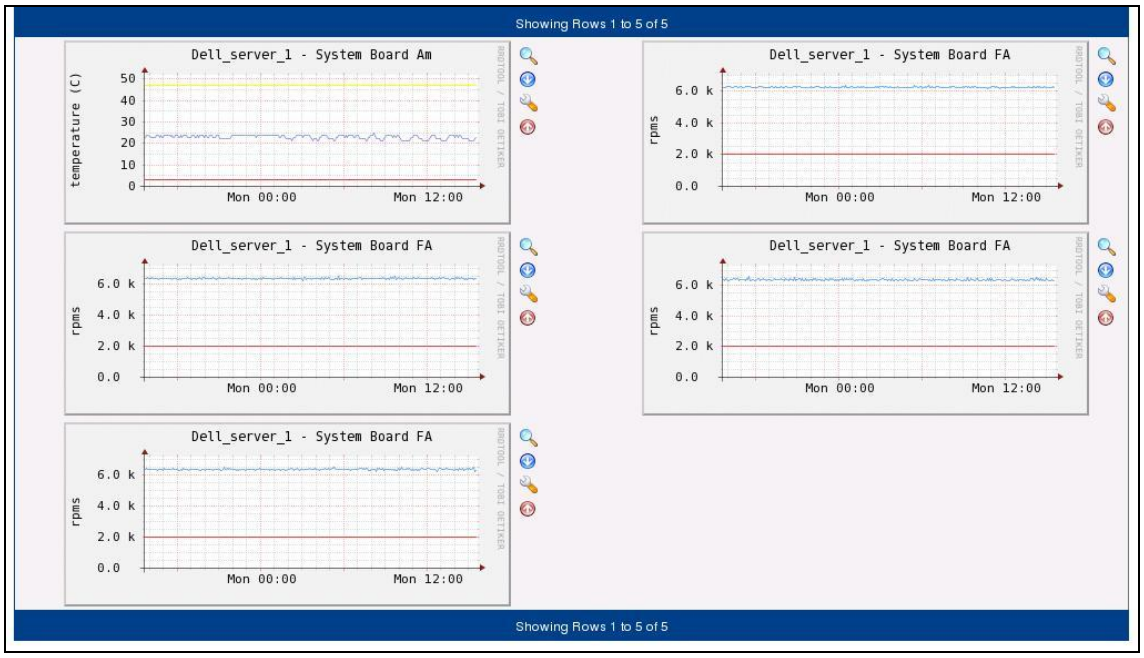


Fig. C.4 Graficas del host Dell en Cacti

C.4 Antena Wi-Fi

Hay distintas antenas repartidas en el edificio para que los usuarios puedan dotar de conexión inalámbrica a internet. A continuación se verá cómo se ha configurado Nagios para monitorizar una de las antenas.

C.4.1 Configuración en Nagios

– Definición del equipo

Se define y se fija el valor de los parámetros de los equipos tipo Antena en el archivo de plantillas "*templates.cfg*".

```
# Define a template for Antenas that we can reuse
define host{
    name                    antena                  ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
    check_interval          5                       ; Actively check the host every 5
                                                     minutes
    retry_interval          1                       ; Schedule host check retries at 1
                                                     minute intervals
    max_check_attempts      10                      ; Check each Linux host 10 times (max)
    check_command           check-host-alive        ; Default command to check Linux hosts
    notification_period     24x7                    ; Send notifications at any time
                                                     ; Note that the notification_period
                                                     variable is being overridden from
                                                       ; the value that is inherited from
                                                       the generic-host template.
    notification_interval   30                        ; Resend notifications every 30
                                                       minutes
```

```
    notification_options    d,r                     ; Only send notifications for
                                                    specific host states
    contact_groups          admins                  ; Notifications get sent to the
                                                    admins by default
    register                0                       ; DONT REGISTER THIS DEFINITION
    }
```

Se crea una carpeta "*wifi_antena*" en "*/usr/local/nagios/etc/objects*"; y en ella un archivo en el que se define el equipo a monitorizar.

```
# Define a host for the Wifi Antena we'll be monitoring

define host{
       use             antena                  ; Inherit default values from a template
       host_name       wifi_antena_1           ; The name we're giving to this host
       alias           wifi_antena _1          ; A longer name associated with the host
       address         172.17.128.11           ; IP address of the host
       hostgroups      wifi_antena
}
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects\ wifi_antena \Antena_1.cfg*".

– Definición de un grupo de equipos

Se siguen los mismos pasos seguidos en los dispositivos definidos anteriormente:

```
# Define a hostgroup for Wifi Antenas
# All hosts that use the Wifi Antenas template will automatically be a member of this
group

define hostgroup{
       hostgroup_name        wifi_antenas          ; The name of the hostgroup
       alias                 Wifi Antenas          ; Long name of the group
}
```

– Definición de los servicios a monitorizar

En este dispositivo se monitorizará solamente un servicio y es el número de usuarios conectados a la antena en tiempo real. Es un dato muy importante ya que si el número de usuarios es muy elevado la antena se puede saturar.

- Definición del comando "*check_wifi_users*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
###############
#check_wifi_users
###############
define command{

command name   check wifi users
command_line   snmpget -v1 -c deesa4845 $HOSTADDRESS$ .1.3.6.1.4.1.9.9.273.1.1.2.1.1.1
       }
```

En la línea de comandos "*command_line*" están los argumentos que Nagios ejecutará: En este caso se usará el comando "*snmpget*" del protocolo SNMP con los parámetros:

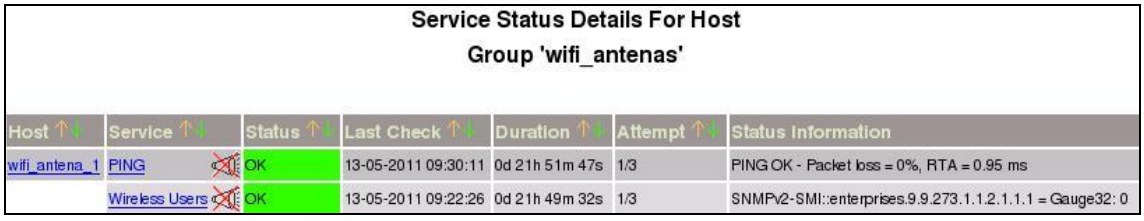
- -v1: Indica que se trata de SNMP versión 1.
- -c deesa4845: Definición de la comunidad SNMP.
- La macro “$HOSTADDRESS$” es la dirección IP del host definido;
- .1.3.6.1.4.1.9.9.273.1.1.2.1.1.1: es el OID correspondiente al objeto del cual se quiere obtener el valor, que es equivalente al servicio a monitorizar, en este caso el número de usuarios conectados a la antena.

o Definición del servicio “*Wirless Users*” en el archivo de definición de equipos y servicios a monitorizar del dispositivo, “*usr/local/nagios/etc/objects/wifi_antena/Antena_1.cfg*”:

```
# Create a service for monitoring the Wifi Antenna

define service{
       use                     service
       host_name               wifi_antena_1
       service_description     Wireless Users
       check_command           check_wifi_users!
       }
```

– Visualización en la interfaz web

Una vez definido el equipo y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios, tal y como muestra la Fig. C.5.



Service Status Details For Host Group 'wifi_antenas'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
wifi_antena_1	PING	OK	13-05-2011 09:30:11	0d 21h 51m 47s	1/3	PING OK - Packet loss = 0%, RTA = 0.95 ms
	Wireless Users	OK	13-05-2011 09:22:26	0d 21h 49m 32s	1/3	SNMPv2-SMI::enterprises.9.9.273.1.1.2.1.1.1 = Gauge32: 0

Fig. C.5 Visualización de los servicios monitorizados de una antena wifi en la interfaz web

C.5 Switches y routers

Hay gran cantidad de switches y routers en el edificio, a parte de los que se encuentra en el CPD. Se han separado los switches en grupos según la planta en la que se encuentran. Hay 4 plantas y en cada una hay un repartidor que contiene entre 3 y 4 switches. Los equipos son del fabricante Cisco, cosa que la configuración básica de Nagios para monitorizarlos es igual para todos, dependiendo de los servicios que se quieren monitorizar de cada uno.

C.5.1 Configuración en Nagios

– Definición del equipo

Se define y se fija el valor de los parámetros de los equipos tipo switch en el archivo de plantillas "*templates.cfg*".

```
# Define a template for Antenas that we can reuse
define host{
    name                    switch                  ; The name of this host template
    use                     host                    ; This template inherits other values
                                                     from the generic-host template
    check_period            24x7                    ; By default, Linux hosts are checked
                                                     round the clock
    check_interval          5                       ; Actively check the host every 5
                                                     minutes
    retry_interval          1                       ; Schedule host check retries at 1
                                                     minute intervals
    max_check_attempts      10                      ; Check each Linux host 10 times (max)
    check_command           check-host-alive        ; Default command to check Linux hosts
    notification_period     24x7                    ; Send notifications at any time
                                                    ; Note that the notification_period
                                                    variable is being overridden from
                                                      ; the value that is inherited from
                                                      the generic-host template.
    notification_interval   30                      ; Resend notifications every 30
                                                      minutes
    notification_options    d,r                     ; Only send notifications for
                                                      specific host states
    contact_groups          admins                  ; Notifications get sent to the
                                                      admins by default
    register                0                       ; DONT REGISTER THIS DEFINITION
    }
```

Luego se crear una carpeta "*switch*" en el directorio: "*/usr/local/nagios/etc/objects*", y en ella un archivo en el que se define cada host a monitorizar. A continuación se puede ver la definición de un switch de la planta 0.

```
# Define the switch that we'll be monitoring

###############################################################################

define host{
       use             switch                  ; Inherit default values from a template
       host_name       SW_1_P_0                ; The name we're giving to this switch
       alias           Cisco_SW_1_P_0          ; A longer name associated with the switch
       address         10.200.0.201            ; IP address of the switch
       hostgroups      Switches_P_0            ; Host groups this switch is associated with
       }
```

– Definición de un grupo de equipos de tipo switch

Se Ha creado un grupo por cada planta para la mejor organización de los equipos. Se muestra a continuación la definición del grupo de hosts de la planta 0.

```
# Create a new hostgroup for switches

define hostgroup{
       hostgroup_name        Switches_P_0                ; The name of the hostgroup
       alias                 Network Switches_P_0        ; Long name of the group
       }
```

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects*\switch*switch.cfg*".

– Definición de los servicios a monitorizar

Hay gran cantidad de servicios que se pueden monitorizar en estos dispositivos, desde la carga de la CPU hasta el tráfico de entrada y salida por las interfaces. En la mayoría de los hosts se ha procedido a monitorizar los siguientes servicios.

- Comprobar la conexión al host – "PING"
- Cuanto tiempo lleva encendido el equipo – "UpTime"
- Estado del puerto de conexión al exterior – "Port 1 Link Status"
- Estado de la ventilación – "Cisco_env"
- Estado de la memoria – "Cisco_mem"

En algunos otros dispositivos se ha monitorizado el servicio de comprobar el tráfico de entrada y salida por las interfaces.

A continuación se puede ver un ejemplo de la monitorización de uno de los servicios.

- Definición del comando "*Cisco_env*" en el archivo de comandos "*usr/local/nagios/etc/objects/commands.cfg*":

```
# 'check_snmp_env.pl' command definition that checks cisco fans and power supply status

define command{
  command_name       check_snmp_env
  command_line       $USER1$/check_snmp_env.pl -H $HOSTADDRESS$ $USER7$ -T $ARG1$ $ARG2$
}
```

En la línea de comandos "*command_line*" están los argumentos que Nagios ejecutará:

- La macro "$USER1$" es ruta hacia el directorio donde se encuentra la herramienta.
- La macro "$HOSTADDRESS$" es la dirección IP del host definido.
- En SNMP versión 1, la macro $USER7$=-C <community>.
- El argumento ARG1 es el tipo de host o marca (stand,netsc,netsl,as400,cisco,cata,nsc,fg,bc,nokia,hp,lp,hpux).

- El argumento ARG2 es un argumento adicional por si necesita en algún momento.

- Definición del servicio "*Cisco_env*" en el archivo de definición de equipo y servicios a monitorizar del dispositivo, "*usr/local/nagios/etc/objects/switch/SW_1_P_0.cfg*":

```
# Monitor Cisco fans and power supply status via SNMP

define service {
       use                     service        ; Inherit values from a template
       host name               SW 1 P 0
       service description     Cisco env
       check_command           check_snmp_env!-C deesa4845 -T cisco
       }
```

La comunidad SNMP es "deesa4845" y el tipo de host es Cisco.

- Acceso a este archivo mediante la siguiente ruta del CD adjunto: "*TFC_Fadi Taki:\Objects*\switch*SW_1_P_0.cfg*".

– Visualización en la interfaz web

Una vez definidos los equipos y los servicios a monitorizar, se pueden visualizar en la interfaz web de Nagios. La Fig. C.6. muestra los servicios monitorizados de los switches de la planta 0 del edificio.

Service Status Details For Host Group 'Switches_P_0'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
SW_1_P_0	Cisco_env	OK	13-05-2011 09:26:17	10d 22h 38m 39s	1/3	1 Fan OK, 1 ps OK : OK
	Cisco_mem	OK	13-05-2011 09:31:15	2d 1h 33m 41s	1/3	Processor:27%,I/O:72% : 32% : : OK
	PING	OK	13-05-2011 09:31:33	2d 1h 38m 23s	1/3	PING OK - Packet loss = 0%, RTA = 1.63 ms
	Port 1 Link Status	OK	13-05-2011 09:28:26	30d 1h 27m 46s	1/3	SNMP OK - 2
	Uptime	OK	13-05-2011 09:27:05	10d 22h 37m 51s	1/3	SNMP OK - Timeticks: (673758736) 77 days, 23:33:07.36
SW_2_P_0	Cisco_env	OK	13-05-2011 09:27:41	10d 22h 37m 15s	1/3	1 Fan OK, 1 ps OK : OK
	Cisco_mem	OK	13-05-2011 09:27:52	10d 22h 37m 4s	1/3	Processor:15%,I/O:67% : 20% : : OK
	PING	OK	13-05-2011 09:33:11	4d 3h 36m 45s	1/3	PING OK - Packet loss = 0%, RTA = 0.88 ms
	Port 1 Link Status	OK	13-05-2011 09:28:02	10d 22h 46m 54s	1/3	SNMP OK - 2
	Uptime	OK	13-05-2011 09:27:29	10d 22h 37m 27s	1/3	SNMP OK - Timeticks: (673635023) 77 days, 23:12:30.23
SW_3_P_0	Cisco_env	OK	13-05-2011 09:27:20	10d 22h 37m 36s	1/3	1 Fan OK, 1 ps OK : OK
	Cisco_mem	OK	13-05-2011 09:33:45	2d 1h 41m 11s	1/3	Driver text:0%,Processor:15%,I/O:43% : 17% : : OK
	PING	OK	13-05-2011 09:32:31	2d 1h 37m 25s	1/3	PING OK - Packet loss = 0%, RTA = 2.78 ms
	Port 1 Link Status	OK	13-05-2011 09:27:55	10d 22h 37m 1s	1/3	SNMP OK - 2
	Uptime	OK	13-05-2011 09:28:23	30d 1h 36m 33s	1/3	SNMP OK - Timeticks: (455112444) 52 days, 16:12:04.44

Fig. C.6 Visualización de los servicios monitorizados de los switches de la planta 0 en la interfaz web

C.6 Conexiones al exterior – Neosky y RedIris

Se trata de los equipos a través de los cuales se conecta la fundación con el exterior. Se definen el host y el grupo de hosts como se han definido para un switch y el servicio más importante a monitorizar de estos equipos es el tráfico de entrada y salida por la interfaz Ethernet de esta conexión.

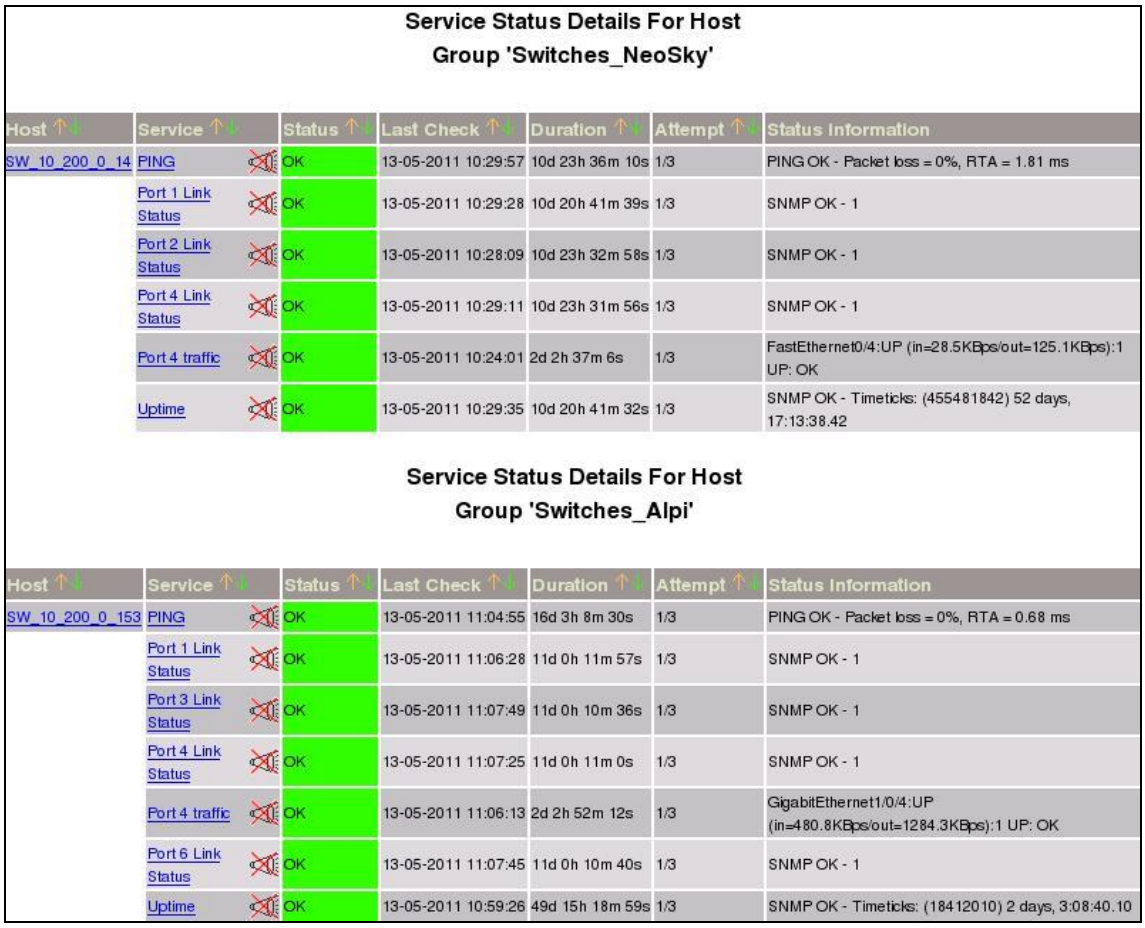
```
# Monitor Port 4 status via SNMP

define service{
     use                     service           ; Inherit values from a template
     host_name               SW 10 200 0 14
     service_description     Port 4 Link Status
     check_command           check_snmp!-C deesa4845 -o ifOperStatus.4 -r 1 -m RFC1213-MIB
}
```

En la línea de comando se puede ver:

- La comunidad SNMP es “deesa4845”.
- La interfaz que se monitoriza es la 4, “–n 0/4”.
- La opción “-f” activa el rendimiento de salida.
- La opción “-k” activa el uso de características estándar.
- La opción “-S” incluye la velocidad en el rendimiento de salida en bits/s.
- Las opciones -w y –c indican que se generan alertas del tipo “WARNING” y “CRITICAL” según los parámetros que les hemos pasado.
- La opción “--label” pone etiquetas “in” y “out” antes de las velocidades de entrada y salida.

– Visualización en la interfaz web

A continuación se muestran los servicios monitorizados de los switches NeoSky y RedIris.



Service Status Details For Host Group 'Switches_NeoSky'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
SW_10_200_0_14	PING	OK	13-05-2011 10:29:57	10d 23h 36m 10s	1/3	PING OK - Packet loss = 0%, RTA = 1.81 ms
	Port 1 Link Status	OK	13-05-2011 10:29:28	10d 20h 41m 39s	1/3	SNMP OK - 1
	Port 2 Link Status	OK	13-05-2011 10:28:09	10d 23h 32m 58s	1/3	SNMP OK - 1
	Port 4 Link Status	OK	13-05-2011 10:29:11	10d 23h 31m 56s	1/3	SNMP OK - 1
	Port 4 traffic	OK	13-05-2011 10:24:01	2d 2h 37m 6s	1/3	FastEthernet0/4:UP (in=28.5KBps/out=125.1KBps):1 UP: OK
	Uptime	OK	13-05-2011 10:29:35	10d 20h 41m 32s	1/3	SNMP OK - Timeticks: (455481842) 52 days, 17:13:38.42

Service Status Details For Host Group 'Switches_Alpi'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
SW_10_200_0_153	PING	OK	13-05-2011 11:04:55	16d 3h 8m 30s	1/3	PING OK - Packet loss = 0%, RTA = 0.68 ms
	Port 1 Link Status	OK	13-05-2011 11:06:28	11d 0h 11m 57s	1/3	SNMP OK - 1
	Port 3 Link Status	OK	13-05-2011 11:07:49	11d 0h 10m 36s	1/3	SNMP OK - 1
	Port 4 Link Status	OK	13-05-2011 11:07:25	11d 0h 11m 0s	1/3	SNMP OK - 1
	Port 4 traffic	OK	13-05-2011 11:06:13	2d 2h 52m 12s	1/3	GigabitEthernet1/0/4:UP (in=480.8KBps/out=1284.3KBps):1 UP: OK
	Port 6 Link Status	OK	13-05-2011 11:07:45	11d 0h 10m 40s	1/3	SNMP OK - 1
	Uptime	OK	13-05-2011 10:59:26	49d 15h 18m 59s	1/3	SNMP OK - Timeticks: (18412010) 2 days, 3:08:40.10

Fig. C.7 Visualización de los servicios monitorizados de los switches Neosky y RedIris

C.6.1 Graficas en Cacti

Solamente se ha visualizado el tráfico del switch-router Neosky. Las graficas se pueden ver en el apartado 3.2.3