



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Implementación de un servidor Nagios para la
monitorización de servicios en red y la actividad de sus
usuarios

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e
Imagen

AUTOR/A: Saez Agud, Joan Angel

Tutor/a: Sendra Compte, Sandra

Cotutor/a: Lloret Mauri, Jaime

CURSO ACADÉMICO: 2023/2024

Índice de contenido

Índice de ilustraciones	3
1. Introducción.....	9
1.1. Motivación del proyecto.....	10
1.2. Objetivos del proyecto	10
1.3. Estructura del trabajo	11
1.4. Planificación temporal.....	11
2. Estado del arte	12
3. Diseño del proyecto	13
3.1. Qué es Nagios	13
3.2. Cómo funciona Nagios	14
3.3. Herramientas utilizadas.....	16
4. Creación del Laboratorio.....	17
4.1. Estructura de Directorios Nagios.....	17
4.2. Servidor Nagios	18
4.2.1. Instalación Nagios	18
4.2.2. Instalación <i>plugins</i> esenciales Nagios	21
4.2.3. Instalación módulo NRPE	22
4.3. Cliente Linux	23
4.3.1. Instalación cliente Linux	23
4.3.2. Instalación <i>plugins</i> esenciales Nagios	24
4.3.3. Instalación y configuración módulo NRPE	24
4.4. Cliente Windows	26

4.4.1.	Instalación cliente Windows	26
4.4.2.	Instalación y configuración módulo NSClient	28
4.5.	Configuración Nagios	29
4.5.1.	Agregar nuevos <i>hosts</i>	29
4.5.2.	Agregar nuevos servicios a monitorear	30
4.6.	Monitoreo cliente Linux	35
4.7.	Monitoreo servidor Windows.....	36
5.	Posibles casos de uso.....	38
5.1.	Caso 1: Creación <i>plugin</i> personalizado de control de memoria RAM.....	38
5.2.	Caso 2: Monitoreo puerto específico	41
6.	Presupuesto	44
7.	Conclusiones	45
	Bibliografía	47

Indice de ilustraciones

Ilustración 1: Esquema de red	10
Ilustración 2: Diagrama Temporal.....	12
Ilustración 3: Funcionamiento Nagios	14
Ilustración 4: Flujo NRPE.....	15
Ilustración 5: Estructura Directorios Nagios	17
Ilustración 6: Subdirectorios/etc	18
Ilustración 7: Habilitación Puertos de Forma Permanente.....	19
Ilustración 8: Instalación Dependencias Nagios	20
Ilustración 9: Instalación Dependencias Nagios	20
Ilustración 10: Descarga Archivos Fuente de Nagios	20
Ilustración 11: Compilación Nagios	20
Ilustración 12: Instalación Nagios.....	20
Ilustración 13: Autenticación Web Nagios	21
Ilustración 14: Interfaz Web Nagios	21
Ilustración 15: Archivos Plugins Nagios	21
Ilustración 16: Plugins Instalados.....	22
Ilustración 17: Descarga Archivos Fuente NRPE	22
Ilustración 18: Datos NRPE	22
Ilustración 19: Archivo de Comandos.....	23
Ilustración 20: Comando NRPE Creado.....	23

Ilustración 21: Habilitación Puerto 5666.....	24
Ilustración 22: Instalación Dependencias.....	24
Ilustración 23: Descarga Plugins.....	24
Ilustración 24: Descarga NRPE.....	24
Ilustración 25: Datos NRPE.....	25
Ilustración 26: Archivo Configuración NRPE.....	25
Ilustración 27: Habilitación Inicio del Sistema con NRPE.....	25
Ilustración 28: Inicio NRPE.....	25
Ilustración 29: Comprobación de Conexión Cliente-Servidor.....	26
Ilustración 30: Regla Firewall.....	26
Ilustración 31: Regla Firewall ICMP.....	27
Ilustración 32: Servicio de NSClient Instalado y Habilitado.....	27
Ilustración 33: Instalación NSClient.....	28
Ilustración 34: Indicar la IP Servidor Nagios.....	28
Ilustración 35: Creación Archivo Configuración Windows.....	29
Ilustración 36: Creación Archivo Configuración Linux.....	29
Ilustración 37: Archivo Configuración Linux.....	29
Ilustración 38: Archivo Configuración Windows.....	29
Ilustración 39: Servicio de Disco Duro.....	30
Ilustración 40: Servicio de Carga del Host.....	30
Ilustración 41: Servicio de Carga de la Partición Swap.....	30

Ilustración 42: Servicio de Comprobación de Ping	30
Ilustración 43: Servicio de Comprobación de Usuarios Activos	31
Ilustración 44: Ruta Archivo Configuración Cliente Linux	31
Ilustración 45: Configuración Archivo NRPE	31
Ilustración 46: Comprobación Archivos de Configuración Cliente Linux	32
Ilustración 47: Servicio Comprobar Versión NSClient.....	32
Ilustración 48: Servicio Comprobar Tiempo Activo	33
Ilustración 49: Servicio Comprobar Carga CPU	33
Ilustración 50: Servicio Comprobar Uso de Memoria	33
Ilustración 51: Servicio Comprobar Espacio en el Disco C:.....	33
Ilustración 52: Ruta Archivo Configuración Cliente Windows	33
Ilustración 53: Archivo Configuración NSClient	34
Ilustración 54: Comprobación Archivos Configuración Cliente Windows	34
Ilustración 55: Listado de Host.....	35
Ilustración 56: Listado de Servicios.....	35
Ilustración 57: Mapa de Red	35
Ilustración 58: Listado de Host.....	36
Ilustración 59: Listado de Servicios.....	36
Ilustración 60: Monitorización Servicios de los 3 Hosts	37
Ilustración 61: Mapa de Red	37
Ilustración 62: Plugins Oficiales	38

Ilustración 63: Creación Archivo Perl	38
Ilustración 64: Fragmento de Código Plugin en Perl Personalizado	39
Ilustración 65: Permisos Archivo Perl.....	39
Ilustración 66: Opciones Sintaxis Plugin.....	40
Ilustración 67: Ejemplo Sintaxis y Prueba Plugin	40
Ilustración 68: Archivo Configuración NRPE	40
Ilustración 69: Definición Servicio a Monitorear.....	41
Ilustración 70: Plugin en Funcionamiento.....	41
Ilustración 71: Plugin que se utiliza	41
Ilustración 72: Ejemplo de uso del plugin.....	41
Ilustración 73: Se define el comando	42
Ilustración 74: Se define el servicio.....	42
Ilustración 75: Interfaz web con el plugin funcionando	43

Resumen

Nagios es una herramienta de código abierto que permite monitorear infraestructuras de red y alertar sobre posibles problemas que puedan afectar a sus usuarios. Nagios permite monitorear servidores, estaciones de trabajo y otros dispositivos de red, así como servicios como HTTP, SMTP, SSH entre otros. Los servidores Nagios recopilan datos de los clientes y generan alertas en tiempo real cuando se detectan problemas, lo que ayuda a mantener la red y sus servicios funcionando de manera eficiente.

En este proyecto se propone la implementación de un servidor Nagios, los plugins y las dependencias que sean necesarias para llevar a cabo la tarea de monitorización de una red virtualizada, con el fin de poder detectar los posibles problemas que puedan surgir en ella para reducir su impacto.

Palabras clave

Nagios, supervisión, *software*, red, telemática, servidor, cliente, monitorización, virtualización.

Abstract

Nagios is an open source tool that allows you to monitor network infrastructures and alert about possible problems that may affect its users. Nagios allows you to monitor servers, workstations and other network devices, as well as services such as HTTP, SMTP, SSH and others. Nagios servers collect data from clients and generate real-time alerts when problems are detected, which helps to keep the network and its services running efficiently.

This project proposes the implementation of a Nagios server, plugins and dependencies that are necessary to carry out the task of monitoring a virtualized network, in order to detect potential problems that may arise in it to reduce its impact.

Key words

Nagios, monitoring, software, network, telematics, server, client, virtualization.

1. Introducción

El presente trabajo de fin de grado (TFG) se centra en la implantación de Nagios con todos los requisitos que ello conlleva.

Nagios es una herramienta de monitorización de sistemas de código abierto. En este trabajo se explorarán sus características principales, su arquitectura y su implementación en diferentes escenarios.

Teniendo en cuenta que, en el mundo actual, se puede considerar que las redes informáticas son el pilar fundamental de cualquier organización, la monitorización de estas redes se ha convertido en una necesidad absoluta. En las infraestructuras de TI se encuentran aplicaciones críticas y servicios esenciales, por lo que su rendimiento y disponibilidad son cruciales para estas empresas. Aquí es donde interviene la monitorización de redes, ya que es un proceso que permite a las organizaciones supervisar continuamente el estado y el rendimiento de sus sistemas, identificar problemas antes de que afecten críticamente y asegurar la integridad de la red.

Para llevar a cabo una monitorización eficaz, es indispensable contar con sistemas operativos que ofrezcan funciones avanzadas de supervisión. Estos sistemas operativos proporcionan la base necesaria para implementar herramientas de monitorización que puedan detectar fallos, gestionar recursos y mantener la integridad de los datos. Una de las herramientas más destacadas en este ámbito es Nagios, un supervisor de redes que ofrece una solución completa para la monitorización de sistemas, redes e infraestructuras.

Nagios está considerado como una primera opción para muchas organizaciones debido a su capacidad para proporcionar una visión integral y en tiempo real del estado de la red. Este *software* permite a los administradores de sistemas detectar problemas antes de que estos se conviertan en fallos críticos, optimizar el rendimiento de la red y garantizar un tiempo de actividad máximo. En este trabajo, se explorarán las características y ventajas de Nagios, así como su importancia en la gestión y monitorización de redes modernas. A través de un análisis detallado, se demostrará cómo Nagios puede ser una herramienta esencial para cualquier organización que busque mantener su infraestructura de TI robusta y eficiente.

1.1. Motivación del proyecto

Se ha elegido este proyecto principalmente por dos razones:

- La más relevante es el interés que suscita en el sector TI este tipo de herramientas, ya que están muy extendidas en el ámbito de la seguridad empresarial. Además, previamente realicé un proyecto con una herramienta de monitorización similar llamada PandoraFMS, por ello, este proyecto me da la posibilidad de estudiar y probar Nagios. Esto puede proporcionarme un espectro más amplio de opinión sobre los supervisores que se pueden encontrar en el mercado.
- Por otra parte, este tipo de herramientas pueden optimizar la gestión de sistemas, mejorar la eficiencia operativa y asegurar la continuidad de las infraestructuras TI.

1.2. Objetivos del proyecto

En este proyecto, se pretende implementar Nagios para monitorizar una red, que será virtualizada y desplegada con VirtualBox, cuya intención será crear una red local con diferentes máquinas. Esta red estará formada por un Windows Server 2022 y dos máquinas CentOS 7, a las que se le asignarán diferentes opciones de monitorización (*plugins* locales, *plugins* remotos, alertas, etc.) para posteriormente poder tener una visión en tiempo real del estado de la red y sus métricas.

En la ilustración 1, se puede observar un esquema de la estructura de la red.

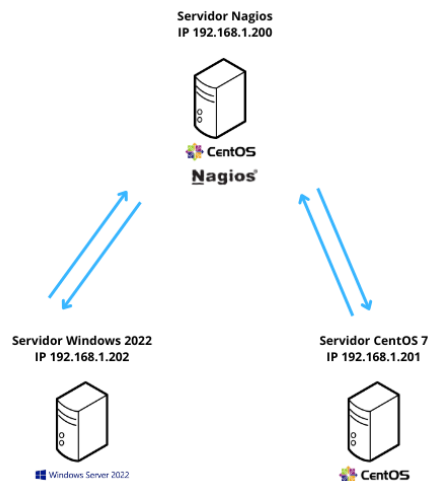


Ilustración 1: Esquema de red

1.3. Estructura del trabajo

El contenido de este trabajo está formado por 7 puntos principales, en cada uno de estos puntos se explica o implementa una parte de Nagios. A continuación, se va a hacer una breve introducción de lo que se puede encontrar en cada punto.

- **Apartado 1:** El primer apartado de este proyecto está formado por la introducción, los objetivos, la estructura y la planificación temporal que se llevará a cabo para la realización del proyecto.
- **Apartado 2:** En el segundo apartado se expondrán diferentes trabajos, artículos científicos y una tabla comparativa con distintas herramientas de monitorización.
- **Apartado 3:** En el tercer apartado se encontrará una explicación de qué es Nagios, cómo funciona y las herramientas que se van a utilizar para su implementación en este trabajo.
- **Apartado 4:** En el cuarto apartado se detallará la creación del laboratorio que estará compuesta por la instalación y configuración tanto del servidor Nagios como de sus clientes.
- **Apartado 5:** En el quinto apartado de este trabajo se verán posibles casos de uso.
- **Apartado 6:** En el sexto apartado se expondrá la simulación de un presupuesto para la implementación de este proyecto en una pequeña empresa.
- **Apartado 7:** En el séptimo apartado se realizarán unas conclusiones y futuras ampliaciones de este trabajo.

1.4. Planificación temporal

Para el desarrollo de este proyecto se han planteado 5 fases, las cuales se detallan a continuación:

- **Fase 1:** Es la fase de inicio, que consiste en la identificación de la idea sobre la que tratará el proyecto.
- **Fase 2:** Es la fase de investigación, se indaga en lo qué consiste el trabajo a realizar y cómo se llevará a cabo.
- **Fase 3:** Es la fase de implantación. Los conocimientos adquiridos en la fase anterior se podrán aplicar para el desarrollo del proyecto.
- **Fase 4:** Es la fase de redacción, en la que se elaborará la memoria del TFG.
- **Fase 5:** Es la fase de cierre. La tutora corregirá la memoria y esta se ultimaré.

A continuación, en la ilustración 2, se observa el diagrama temporal del proyecto dividido en las 5 fases comentadas.

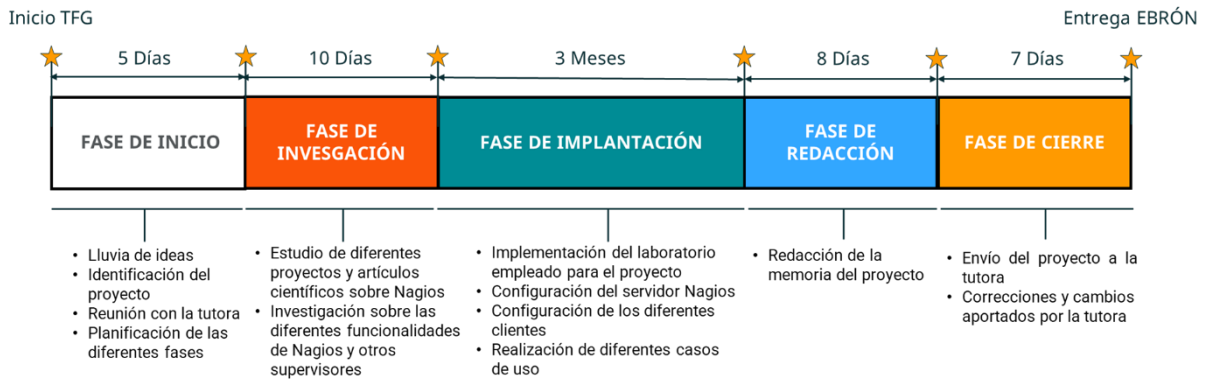


Ilustración 2: Diagrama Temporal

2. Estado del arte

En este apartado del documento, en primer lugar, se mencionan algunos TFGs/TFMs, artículos científicos, etc., relacionados con herramientas de supervisión.

- *Building a Service-Oriented Monitoring Framework with REST and Nagios.* (2011b, julio 1). IEEE Conference Publication | IEEE Xplore.
https://ieeexplore.ieee.org/abstract/document/6009290?casa_token=f5dvxEA9cL0AAAAA:ZFZfBnyC1IhyMA-r9KqI1y4stn_GPuIdY6mpfKN855wMotOmH6lpLVC1mkrqCFK4qUiVBQWsQ
- *Using Nagios as a groundwork for developing a better network monitoring system.* (2012b, julio 1). IEEE Conference Publication | IEEE Xplore.
https://ieeexplore.ieee.org/abstract/document/6304293?casa_token=ogOBa1R15Z8AAAAA:otbDXyU7dpKyoc3IO4ozQpUQDKGU67EDbNHbUU4hfVd_shyhP0OQRnv4HNFNc_NsNXGZY5qd2w
- *Nagios Core Administration Cookbook.* (s. f.). Google Books.
https://books.google.es/books?hl=es&lr=&id=WG1LDAAAQBAJ&oi=fnd&pg=PP1&dq=nagios&ots=_evpOSbGlz&sig=GH5RvPX7e9Mm72euaVv6FitnJV4#v=onepage&q=nagios&f=false
- *Learning Nagios.* (s. f.). Google Books.
<https://books.google.es/books?hl=es&lr=&id=dqnWDQAAQBAJ&oi=fnd&pg=PP1&dq=nagios&ots=sxt8cRd4j4&sig=8qQXAnKjjOyZN2TLMHlxzUcC9KI#v=onepage&q=nagios&f=false>

En segundo lugar, se presenta una tabla comparativa de diversas herramientas de monitorización. Además, se explica por qué se ha seleccionado Nagios entre ellas.

Característica	Nagios	Pandora FMS	Paessler PRTG
Tipo de herramienta	Monitoreo de red y sistemas	Monitoreo de red y sistemas	Monitoreo de red y sistemas
Modelo de licenciamiento	Open Source y Comercial	Open Source y Comercial	Comercial
Plataformas soportadas	Windows, Linux y Unix	Windows, Linux y Unix	Windows
Facilidad de uso	Moderada	Moderada	Alta
Escalabilidad	Alta	Alta	Media
Soporte y comunidad	Amplia comunidad y soporte comercial	Amplia comunidad y soporte comercial	Soporte comercial y comunidad activa
Alertas y notificaciones	Avanzadas	Avanzadas	Avanzadas
Informes y dashboards	Personalizables	Personalizables	Personalizables
Integraciones	Múltiples integraciones	Múltiples integraciones	Múltiples integraciones
Costo	Gratis (<i>open source</i>) y versiones pagas	Gratis (<i>open source</i>) y versiones pagas	Paga (con versión de prueba)
Funcionalidades especiales	Plugins extensivos y escalabilidad	Monitoreo distribuido y soporte IoT	Mapas de red automáticos y monitoreo móvil
Requisitos de hardware	Moderados	Moderados	Moderados

Se ha escogido Nagios sobre las otras herramientas porque, aunque Pandora FMS y Paessler PRTG también son opciones muy competentes y fáciles de utilizar, el modelo de costos de Nagios lo convierte en una opción preferible para muchas organizaciones que buscan una solución de monitoreo robusta y adaptable.

3. Diseño del proyecto

En este tercer apartado se detalla el funcionamiento de Nagios, sus diferentes componentes y los flujos que se encuentran en estos. También se explican las herramientas que se utilizan para su correcta implementación en este trabajo.

3.1. Qué es Nagios

Nagios es una herramienta de código abierto cuya función es monitorizar infraestructuras TI. A este tipo de herramientas también se les puede identificar como supervisores.

Nagios puede proporcionar un monitoreo en tiempo real de la infraestructura TI y reportar tanto alertas como notificaciones. También se caracteriza por su extensibilidad, ya que su arquitectura modular y la disposición de *plugins* le permiten adaptarse a cualquier entorno. Nagios está dotado de una interfaz web, la que facilita su manejo y la visualización del estado de los sistemas. Esta combinación simplifica el registro y seguimiento de incidencias que puedan suceder en la red.

3.2. Cómo funciona Nagios

Para comprender el funcionamiento de Nagios y sus componentes, se puede observar la siguiente ilustración:

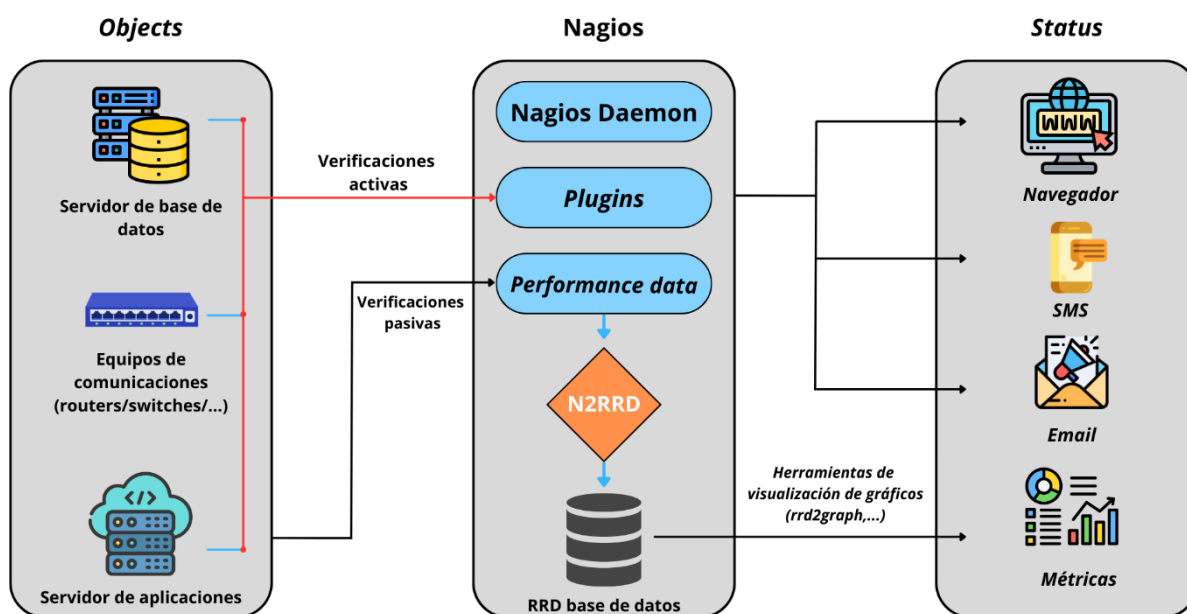


Ilustración 3: Funcionamiento Nagios

En la sección central de la ilustración 3, “Nagios”, se encuentra la estructura por la que está compuesta Nagios. Esta estructura se divide en las siguientes partes:

- **Nagios Daemon:** Es considerado como el componente central de Nagios. Se encarga de ejecutarse en segundo plano, en el servidor donde está instalado Nagios y realizar la mayor parte del trabajo de monitoreo. Realiza comprobaciones periódicas del estado de los clientes y servicios configurados (CPU, RAM, HTTP, SMTP, etc.). Cuando Nagios *Daemon* detecta un problema es capaz de generar alertas y notificaciones. Además, mantiene un registro detallado de todos los eventos de monitoreo, pudiendo ser utilizados para generar informes y analizar tendencias de rendimiento.
- **Plugins:** Se pueden integrar a Nagios para ampliar sus funcionalidades. Estos son esenciales, ya que ejecutan las tareas específicas de monitoreo y devuelven los resultados al núcleo de Nagios. Son programas independientes, debido a que no están integrados directamente en el núcleo de Nagios. Los *plugins* pueden ser oficiales o desarrollados por los propios usuarios en lenguajes como Bash, Perl, Python, etc.

- **Performance data:** Se encarga de acumular datos sobre los servicios y los hosts monitoreados, para posteriormente proporcionar gráficas de rendimiento mediante N2RRD.

En la sección izquierda de la ilustración 3, “Objects”, se encuentran los equipos que Nagios es capaz de monitorizar. Estos son los siguientes:

- **Servidores/Host:** Para la monitorización de los clientes, Nagios hace uso de los protocolos TCP/IP y SSL, este último se encarga de garantizar la privacidad e integridad de los datos transmitidos. Para la monitorización, se ejecuta un programa agente en los clientes, en el caso de máquinas con sistema operativo Windows es el NSClient y en el caso de máquinas basadas en Linux se usa el NRPE, que operan en el puerto 5666. Nagios se comunica con este programa agente con el propósito de consultar el estado de los clientes. El programa agente hace uso de los *plugins* configurados con el fin de saber que instrucciones ejecutar en cada cliente. Estos *plugins* deben coincidir tanto en el servidor Nagios como en el cliente, ya que Nagios se encarga de conectarse al cliente y ejecutar el *plugin* desde este. En la siguiente ilustración se muestra el diagrama de flujo de datos ente los clientes y el servidor Nagios.

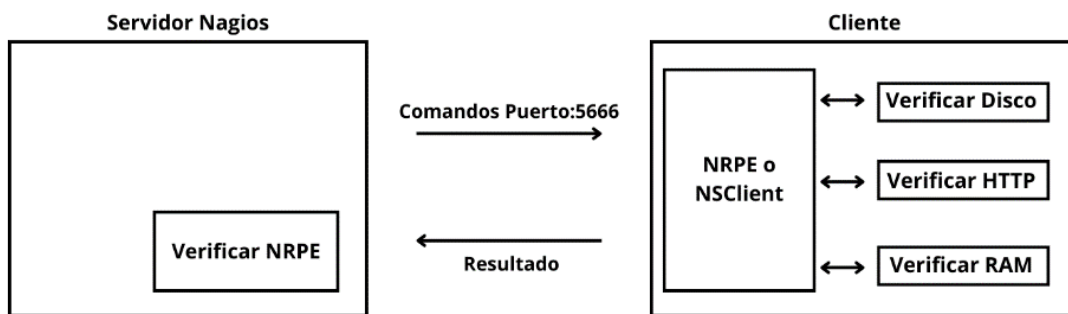


Ilustración 4: Flujo NRPE

- **Equipos de comunicaciones:** Por lo general, al no poder ejecutar el programa agente ni los *plugins* en los equipos de comunicaciones, se hace uso del protocolo SNMP, que se encargara de comunicar el estado de los equipos al servidor Nagios.

En resumen, el cliente no se encarga de mandar la información periódicamente a Nagios. Es Nagios el que tiene los tiempos de consulta programados y mediante su NRPE, cada cierto tiempo consulta a los clientes su estado. Mediante los agentes instalados, los clientes devuelven al servidor Nagios su estado. Nagios *Daemon* se encarga de procesar el estado y así poder consultarlo como se indica en la sección derecha de la ilustración 3, “Status”.

Entre las secciones “Nagios” y “Objects”, se puede encontrar:

- **Verificaciones activas:** Estos son los chequeos que realiza Nagios de manera programada.
- **Verificaciones pasivas:** Estos son los chequeos que no son disparados por Nagios, son disparados por aplicaciones de terceros desde los clientes.

En la sección derecha de la ilustración 3, "Status", se encuentran las maneras en las que Nagios reportará la información. Estas son las siguientes:

- **Navegador:** Se puede consultar la información recopilada por Nagios accediendo a su interfaz web mediante un navegador introduciendo la dirección IP de este.
- **SMS:** Se puede consultar la información mediante un teléfono móvil.
- **EMAIL:** Se pueden configurar diferentes *plugins* y programas para que se puedan recibir alertas mediante Email.
- **Métricas:** Nagios es capaz de generar gráficas con la información recopilada.

3.3. Herramientas utilizadas

Para la implementación de este proyecto se han utilizado diferentes herramientas. Como pilar fundamental se ha utilizado el programa VirtualBox para la creación de la red en la que será implementado este proyecto. Este virtualizador se ha elegido porque es gratuito, fácil de usar y cumple con los requisitos necesarios para el proyecto.

Los dos sistemas operativos utilizados tanto en los clientes como en los servidores son los que se indican a continuación:

- Por una parte, se ha hecho uso del sistema operativo Linux, específicamente la distribución CentoOS 7, tanto para el servidor Nagios como para uno de los clientes a monitorizar. Se ha escogido esta distribución respecto a otras por su amplia documentación, gran estabilidad y bajo consumo de recursos, entre otros motivos.
- Por otra parte, se ha utilizado el sistema operativo Windows Server 2022 como servidor a monitorizar, debido a su extenso uso en el ámbito empresarial.

Como herramienta principal se ha empleado Nagios, concretamente Nagios Core, ya que es la versión gratuita.

Todos los clientes disponen de un agente *software* para la recopilación de la información. En el caso de los clientes cuya distribución de sistema operativo está basada en Linux, se hace uso del NRPE, mientras que para los clientes cuya distribución de sistema operativo está basada en Windows, se emplea NSClient, ambos mencionados anteriormente.

4. Creación del Laboratorio

Para la correcta implementación del proyecto, se hará uso de un laboratorio de pruebas, el cual está compuesto por las herramientas citadas en el punto anterior. Para el correcto funcionamiento de este laboratorio, las máquinas y la red deberán ser instaladas y configuradas correctamente.

4.1. Estructura de Directorios Nagios

Con el fin de un mejor entendimiento del funcionamiento de Nagios y su estructura, ya que se hará uso de esta durante todo el trabajo, a continuación, se exponen los tipos de configuraciones que se pueden encontrar dentro de cada directorio.

En la siguiente ilustración se puede observar el directorio principal de Nagios, donde se encuentra toda su estructura de carpetas.

```
[root@nagios nagios]# tree /usr/local/nagios/ -L 2 -d
/usr/local/nagios/
├── bin
├── etc
│   └── objects
├── libexec
├── sbin
├── share
│   ├── angularjs
│   ├── bootstrap-3.3.7
│   ├── contexthelp
│   ├── d3
│   ├── docs
│   ├── images
│   ├── includes
│   ├── js
│   ├── locale
│   ├── media
│   ├── spin
│   ├── ssi
│   └── stylesheets
├── var
│   ├── archives
│   ├── rw
│   └── spool
23 directories
```

Ilustración 5: Estructura Directorios Nagios

A continuación, se nombran los directorios más relevantes:

- **bin:** Directorio donde se encuentran los ejecutables propios del motor de Nagios, no sus *plugins*.
- **etc:** Directorio donde se encuentra el archivo principal de configuración de Nagios. Seguidamente, en la ilustración 6 se muestra este directorio.

```
[root@nagios etc]# ls
cgi.cfg  htpasswd.users  nagios.cfg  objects  resource.cfg
```

Ilustración 6: Subdirectorios/etc

- **cgi.cfg:** Archivo de configuración de la página web.
- **nagios.cfg:** Archivo de configuración *core* de Nagios.
- **resource.cfg:** Archivo donde se encuentran las referencias al directorio donde están los *plugins*.
- **htpasswd.users:** Archivo donde se sitúan los usuarios que pueden iniciar sesión en la página de Nagios.
- **objects:** Directorio donde se encuentran todos los archivos de configuración de los servidores que se quiere monitorear: comandos, *templates*, etc.
- **libexec:** Directorio donde están todos los *plugins* que Nagios va a emplear para el monitoreo.
- **Share:** Directorio donde se encuentran todos los archivos referentes a la interfaz web del servidor Nagios a la que se accede para visualizar el estado de los servidores y servicios.
- **Var:** Directorio donde permanecen los archivos temporales guardados por Nagios y los archivos de cola de los monitoreos.

4.2. Servidor Nagios

En este subapartado se desarrolla la instalación del servidor Nagios, sus *plugins* y el NRPE. También se tratan las diferentes configuraciones tanto para el *firewall* como para las dependencias y el NRPE.

4.2.1. Instalación Nagios

Para la instalación de la máquina virtual donde estará instalado Nagios, en primer lugar, es necesario configurar la tarjeta de red en modo puente, ya que de esta manera la máquina virtual se comporta como si estuviera en la misma red. Además, es necesario configurar la IP en modo estático, ya que esto permite que los clientes puedan identificar al servidor Nagios sin dificultades.

4.2.1.1. Configuración Firewall

Debido a que Nagios dispone de una interfaz web, se tienen que habilitar algunos de los puertos para que esto sea posible. Los puertos que se van a habilitar son el puerto 80 y el 443, los cuales se detallan seguidamente.

Puerto 80: Se utiliza de forma predeterminada en HTTP por los servidores web para escuchar y recibir solicitudes de navegación de clientes, como navegadores web.

Puerto 443: Este puerto se habilita en el caso de emplear HTTPS y se desee añadir un certificado.

A continuación, en la ilustración 7, se muestra cómo se habilitan permanentemente los puertos mencionados.

```
[root@nagios selinux]# firewall-cmd --permanent --add-port=80/tcp
success
[root@nagios selinux]# firewall-cmd --permanent --add-port=443/tcp
success
[root@nagios selinux]# firewall-cmd --reload
success
[root@nagios selinux]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports: 80/tcp 443/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Ilustración 7: Habilitación Puertos de Forma Permanente

4.2.1.2. Instalación de dependencias

Con el objetivo de realizar la compilación del motor de Nagios es necesario instalar algunas dependencias, para que este funcione correctamente.

En el primer comando se instalan las siguientes dependencias:

- **gettext:** Utilidades de internacionalización y localización.
- **wget:** Herramienta de línea de comandos para descargar archivos desde la web.
- **net-snmp:** Implementación del protocolo SNMP (*Simple Network Management Protocol*).
- **net-snmp-utils:** Utilidades adicionales para net-snmp.
- **openssl-devel:** Bibliotecas de desarrollo para aplicaciones que utilizan OpenSSL.
- **glibc-common:** Archivos comunes para la biblioteca C estándar de GNU.
- **unzip:** Herramienta para descomprimir archivos ZIP.
- **perl:** Lenguaje de programación Perl.
- **epel-release:** Paquete que habilita el repositorio EPEL (*Extra Packages for Enterprise Linux*).
- **gcc:** Compilador de C y C++.
- **php:** Lenguaje de programación PHP.
- **gd:** Biblioteca gráfica para la manipulación de imágenes.
- **automake:** Herramienta para generar Makefiles compatibles con GNU.
- **autoconf:** Herramienta para generar *scripts* de configuración automática.
- **httpd:** Servidor web Apache.

- **make:** Herramienta de construcción para administrar la compilación de proyectos.
- **glibc:** Biblioteca C estándar de GNU.
- **gd-devel:** Archivos de desarrollo para la biblioteca gd.

Este primer comando con las dependencias comentadas a instalar se muestra en la siguiente ilustración.

```
[root@nagios ~]# yum install -y gettext wget net-snmp-utils openssl-devel glibc-common unzip perl epel-release gcc php gd automake autoconf httpd make glibc gd-devel net-snmp
```

Ilustración 8: Instalación Dependencias Nagios

En el segundo comando se instala la siguiente dependencia:

- **perl-Net-SNMP:** Permite a los *scripts* de Perl realizar operaciones SNMP, tales como la gestión y monitoreo de dispositivos de red.

Este segundo comando con la dependencia mencionada a instalar se muestra en la siguiente ilustración.

```
[root@nagios ~]# yum install perl-Net-SNMP
```

Ilustración 9: Instalación Dependencias Nagios

4.2.1.3. Instalación binarios Nagios

Con el fin de realizar la compilación de Nagios es necesario descargar sus archivos binarios. A continuación, en las ilustraciones 10, 11 y 12 se indican algunos de los comandos utilizados para este fin, que son la descarga, compilación e instalación de Nagios, respectivamente.

```
[root@nagios ~]# wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.1/nagios-4.5.1.tar.gz
```

Ilustración 10: Descarga Archivos Fuente de Nagios

```
[root@nagios nagios-4.5.1]# make all
```

Ilustración 11: Compilación Nagios

```
[root@nagios nagios-4.5.1]# make install
```

Ilustración 12: Instalación Nagios

Una vez descargados todos los archivos fuente de Nagios, instalados todas las dependencias necesarias y creados sus usuarios, se puede acceder a la interfaz web de Nagios, para poder visualizar todos los datos recopilados.

En las ilustraciones 13 y 14 se puede visualizar la autenticación web y la página inicial, respectivamente.

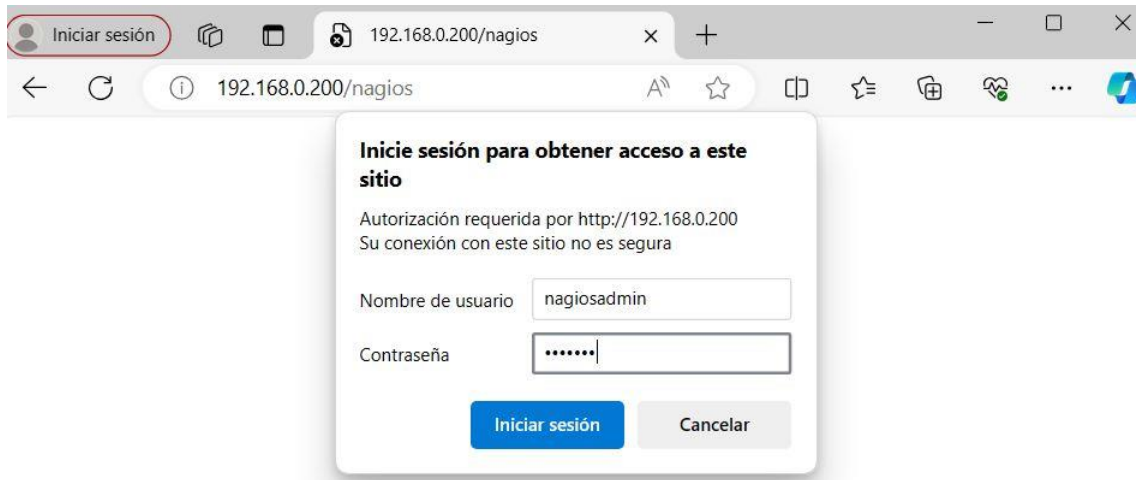


Ilustración 13: Autenticación Web Nagios

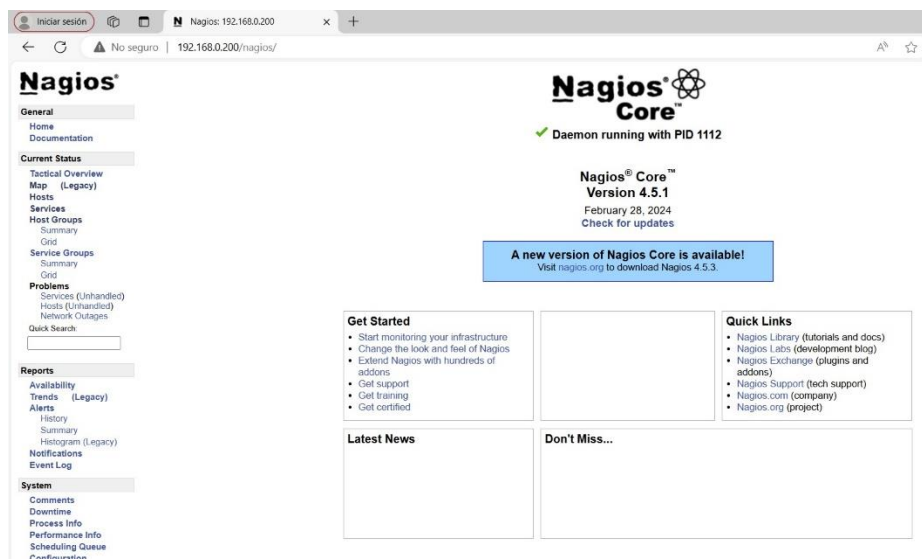


Ilustración 14: Interfaz Web Nagios

4.2.2. Instalación *plugins* esenciales Nagios

Para que Nagios pueda monitorizar los servicios específicos de cada *host* es necesario descargar y compilar los *plugins*. En este primer momento se instalan los *plugins* de Nagios, aunque también es posible crear unos propios.

En la ilustración 15, se muestra cómo se descargan e instalan los archivos fuente de los *plugins* de Nagios.

```
[root@nagios ~]# wget https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.9/nagios-plugins-2.4.9.tar.gz
```

Ilustración 15: Archivos *Plugins* Nagios

Posteriormente, en la ilustración 16, se pueden comprobar los *plugins* instalados. Entre estos podemos encontrar para la verificación de discos, memoria, protocolos, procesos, entre otros.

```
[root@nagios nagios-plugins-2.4.9]# cd /usr/local/nagios/libexec/
[root@nagios libexec]# ls
check_apt      check_ftp      check_mrtg     check_pop      check_tcp
check_breeze  check_hpjd     check_mrtgtraf check_procs    check_time
check_by_ssh   check_http     check_nagios   check_real     check_udp
check_clamd    check_icmp     check_nntp     check_rpc      check_ups
check_cluster  check_ide_smart check_nntp     check_sensors  check_uptime
check_dhcp     check_ifoperstatus check_nt       check_simap    check_users
check_dig      check_ifstatus  check_ntp     check_smtp     check_wave
check_disk     check_imap     check_ntp_peer check_snmp     negate
check_disk_smb check_ircd     check_ntp_time check_spop     remove_perfdata
check_dns      check_jabber   check_nwstat  check_ssh      urlize
check_dummy    check_load     check_oracle  check_ssl_validity utils.pm
check_file_age check_log      check_overcr  check_ssmtp    utils.sh
check_flexlm   check_mailq   check_ping    check_swap
```

Ilustración 16: Plugins Instalados

4.2.3. Instalación módulo NRPE

El agente *software* NRPE (*Nagios Remote Plugin Executor*) es una utilidad para ejecutar *plugins* de Nagios en máquinas remotas. Es especialmente útil para monitorear recursos locales como la utilización de CPU, memoria, espacio en disco, etc. La comunicación del NRPE está protegida por SSL, para mayor seguridad y opera en el puerto 5666. Este puerto permite conectar el servidor Nagios a los clientes que se pretenden monitorear. Mediante el NRPE se ejecutan los *plugins* y se obtienen las respuestas de los estados de los servicios que se desean monitorear.

En la ilustración 17, se visualiza cómo se instala, compila y configura el NRPE en el servidor Nagios.

```
[root@nagios ~]# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.1.0/nrpe-4.1.0.tar.gz
```

Ilustración 17: Descarga Archivos Fuente NRPE

Se comprueban los datos del NRPE en la siguiente ilustración:

```
*** Configuration summary for nrpe 4.1.0 2022-07-18 ***:
General Options:
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios
```

Ilustración 18: Datos NRPE

Se debe crear un comando para que Nagios pueda usar el NRPE. Este comando se crea en el archivo `commands.cfg`.

En la siguiente ilustración, se presenta cómo se crea el archivo.

```
[root@nagios nrpe-4.1.0]# cd /usr/local/nagios/etc/objects/
[root@nagios objects]# ls
commands.cfg  localhost.cfg  switch.cfg    timeperiods.cfg
contacts.cfg  printer.cfg    templates.cfg windows.cfg
[root@nagios objects]# nano commands.cfg
```

Ilustración 19: Archivo de Comandos

A continuación, se detalla la estructura del comando mencionado anteriormente:

- **define command:** Se define el comando.
- **command_name check_nrpe:** Se indica el nombre del comando con el que se invocará.
- **command_line:** Línea de ejecución.
 - **\$USER1\$:** Significa `/usr/local/nagios/libexec`, es una abreviación.
 - **check_nrpe -H:** Es un *plugin* que se utiliza para enviar comandos y consultar el estado de los servicios y los recursos en un *host* remoto que tiene instalado y configurado el servidor NRPE.
 - **\$HOSTADDRESS\$ -c:** Dirección IP del servidor que se está monitoreando, esta IP es utilizada por el *plugin* `check_nrpe -H`.
 - **\$ARG1\$:** Indica el *plugin* que se quiere chequear.

En la siguiente ilustración, se visualiza el comando recientemente detallado.

```
define command {
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Ilustración 20: Comando NRPE Creado

4.3. Cliente Linux

En este subapartado se trata la instalación del cliente Linux, sus *plugins* y el NRPE. También se desarrollan las diferentes configuraciones tanto para el *firewall* como para las dependencias y el NRPE.

4.3.1. Instalación cliente Linux

A fin de realizar la instalación del cliente Linux, se debe realizar la configuración de la tarjeta de red en modo puente y configurar la IP con una IP estática, como se ha hecho en el caso del servidor Nagios.

4.3.1.1. Configuración Firewall

Como se ha mencionado en puntos anteriores, el NRPE funciona sobre el puerto 5666, por tanto, este puerto se tiene que habilitar de forma permanente para que este el NRPE funcione.

En la ilustración 21, se habilita el puerto permanentemente.

```
[root@cliente ~]# firewall-cmd --permanent --add-port=5666/tcp  
success
```

Ilustración 21: Habilitación Puerto 5666

4.3.1.2. Instalación de dependencias

Para el correcto funcionamiento de Nagios en el cliente, es necesario instalar algunas dependencias.

En el siguiente comando, que se muestra en la ilustración 22, se instalan algunas de las dependencias descritas anteriormente en la instalación del servidor Nagios.

```
[root@cliente ~]# yum install -y gcc glibc glibc-common openssl openssl-devel perl wget
```

Ilustración 22: Instalación Dependencias

4.3.2. Instalación *plugins* esenciales Nagios

En la siguiente ilustración, se muestra cómo se descargan e instalan los *plugins* como se hizo con el servidor Nagios.

```
[root@cliente ~]# wget https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.9/  
nagios-plugins-2.4.9.tar.gz
```

Ilustración 23: Descarga Plugins

4.3.3. Instalación y configuración módulo NRPE

En la ilustración 24, se visualiza como se descarga e instala el NRPE como se hizo con el servidor Nagios.

```
[root@cliente ~]# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.1.0/nrpe-4.1.0  
.tar.gz
```

Ilustración 24: Descarga NRPE

En la siguiente ilustración, se comprueban los datos del NRPE.

```

*** Configuration summary for nrpe 4.1.0 2022-07-18 ***:

General Options:
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios

```

Ilustración 25: Datos NRPE

A partir de este punto del documento, la configuración del cliente tiene una diferencia notable respecto a la configuración del servidor Nagios, esto es debido a que se edita el archivo nrpe.cfg de configuración del NRPE.

En el archivo de configuración se busca la línea `allowed_host`, la cual siempre se tiene que editar en los clientes que quieran ser monitoreados por Nagios. En esta línea se añade la dirección IP del servidor Nagios.

En la siguiente ilustración se introduce la IP del servidor Nagios.

```

GNU nano 2.3.1                               Fichero: nrpe.cfg
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,::1,192.168.1.200

```

Ilustración 26: Archivo Configuración NRPE

En las ilustraciones 27, 28 y 29, se habilita el servicio NRPE para que siempre se inicie con el sistema y se comprueba desde el servidor Nagios si está funcionando correctamente.

```

[root@cliente etc]# systemctl enable nrpe
Created symlink from /etc/systemd/system/multi-user.target.wants/nrpe.service to /usr/lib/systemd/system/nrpe.service.

```

Ilustración 27: Habilitación Inicio del Sistema con NRPE

```

[root@cliente etc]# systemctl start nrpe

```

Ilustración 28: Inicio NRPE

```
[root@nagios libexec]# ./check_nrpe -H 192.168.1.201
NRPE v4.1.0
```

Ilustración 29: Comprobación de Conexión Cliente-Servidor

4.4. Cliente Windows

En este subapartado se desarrolla la instalación del servidor Windows, sus *plugins* y el NRPE. También se tratan las diferentes configuraciones tanto para el *firewall* como para las dependencias y el NRPE.

4.4.1. Instalación cliente Windows

Con el fin de realizar la instalación del cliente Windows, se tiene que realizar una configuración de la tarjeta de red en modo puente y configurar la IP con una IP estática, como se ha realizado en las configuraciones de red del resto de las máquinas.

4.4.1.1. Configuración Firewall

En el *firewall* de Windows se crea una regla de entrada en la que se permite la interacción del servidor Nagios con los diferentes puertos y protocolos del cliente. De esta manera puede aceptar conexiones y peticiones.

En la siguiente ilustración, se muestra cómo se crea la regla del *firewall*.

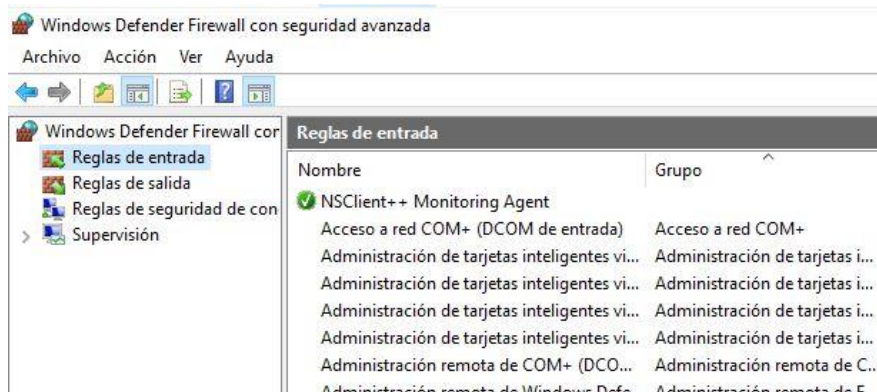


Ilustración 30: Regla Firewall

La regla del *firewall* del servidor Windows se configura de manera que permita que el ICMP pueda comunicarse de forma adecuada con el servidor Nagios.

A continuación, en la ilustración 31, se visualiza la configuración de la regla del *firewall*.

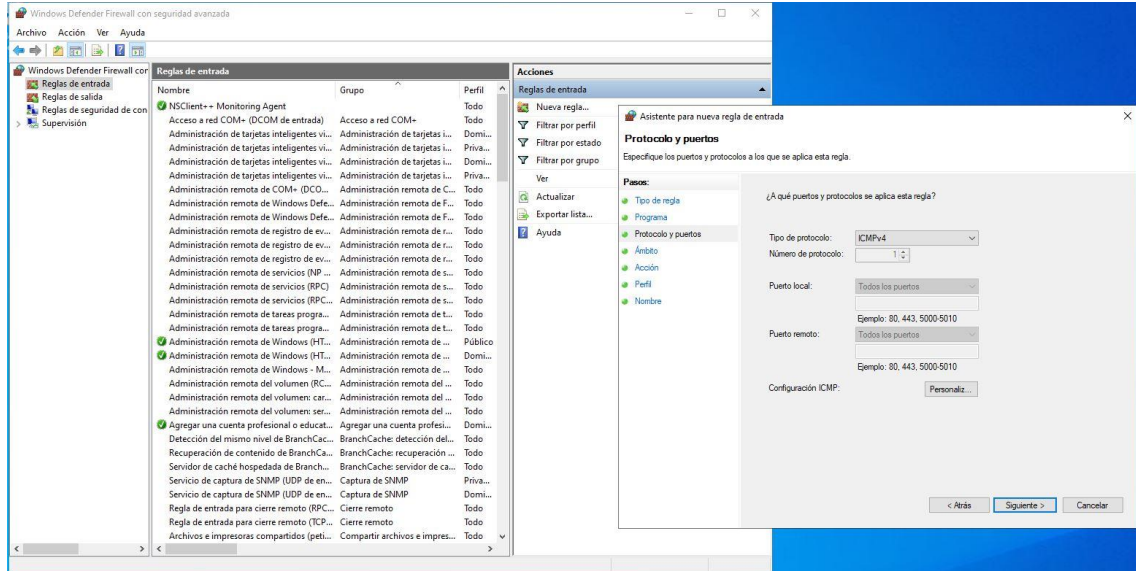


Ilustración 31: Regla Firewall ICMP

En la siguiente ilustración, se asegura que el servicio de NSClient está instalado y habilitado.

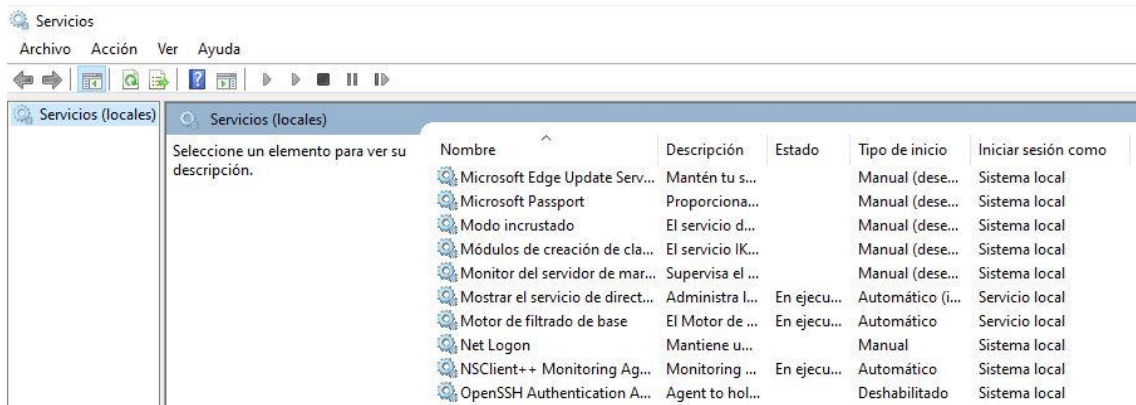


Ilustración 32: Servicio de NSClient Instalado y Habilitado

4.4.2. Instalación y configuración módulo NSClient

La herramienta NSClient es la versión del NRPE para Windows. El NRPE del servidor Nagios se comunica con el NSClient en el cliente Windows para pedirle al NSClient que ejecute los *plugins* de monitorización desde el propio cliente.

En la siguiente ilustración, se descarga la última versión de NSClient.

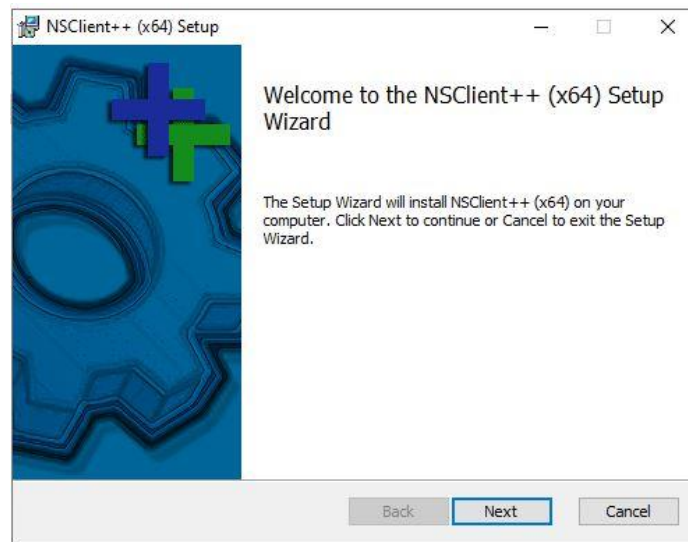


Ilustración 33: Instalación NSClient

En la ilustración 34, se configura el NSClient, indicando la IP del servidor Nagios para que este ejecute los *plugins* sobre el cliente.

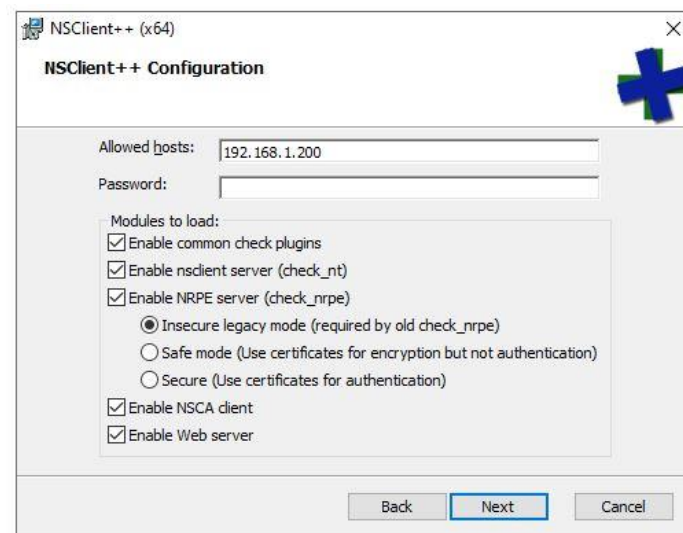


Ilustración 34: Indicar la IP Servidor Nagios

4.5. Configuración Nagios

En este apartado se realizan las configuraciones necesarias, tanto en el servidor Nagios como en los clientes, para que estos y sus servicios sean monitorizados correctamente.

4.5.1. Agregar nuevos *hosts*

En las ilustraciones 35 y 36, se agregan en el servidor Nagios los nuevos *hosts* a monitorear. Para ello, en el directorio “objects”, se crea un archivo de configuración tanto para el cliente Windows como el cliente Linux.

```
[root@nagios ~]# cd /usr/local/nagios/etc/objects/  
[root@nagios objects]# nano windows.cfg
```

Ilustración 35: Creación Archivo Configuración Windows

```
[root@nagios objects]# nano linux.cfg_
```

Ilustración 36: Creación Archivo Configuración Linux

Seguidamente se detalla el archivo de configuración de los clientes Linux y Windows:

- **use:** Define el *template* que utiliza.
- **host_name:** Define el nombre del cliente a monitorizar.
- **alias:** Alias del cliente.
- **check_interval:** Intervalo que tarda en realizar cada comprobación del *host*. En este caso es 1 minuto.
- **address:** IP del cliente a monitorizar.

En las siguientes ilustraciones, 37 y 38, se definen los nuevos *hosts*.

```
#### Definimos host linux a monitorear ####  
  
define host{  
    use                linux-server  
    host_name          cliente_linux  
    alias              Linux Centos  
    check_interval    1  
    address            192.168.1.201  
}
```

Ilustración 37: Archivo Configuración Linux

```
define host {  
  
    use                windows-server          ; Inherit default values from a template  
    host_name          cliente_windows        ; The name we're giving to this host  
    alias              servidor windows      ; A longer name associated with the host  
    address            192.168.1.202        ; IP address of the host  
}
```

Ilustración 38: Archivo Configuración Windows

4.5.2. Agregar nuevos servicios a monitorear

En los archivos mencionados en el apartado anterior y a continuación de definir los clientes a monitorizar, se ponen los servicios a monitorizar de cada uno de ellos.

4.5.2.1. Servicios cliente Linux

Se puede observar cómo se definen en el servidor Nagios los servicios que se quieren monitorizar del cliente Linux.

- **use:** Define qué servicio se quiere utilizar. Esta utiliza algunos de sus parámetros del archivo *templates*.
- **host_name:** Nombre del *host* a monitorizar definido anteriormente.
- **service_description:** Breve descripción del servicio.
- **check_command:** Comando del NRPE que se debe ejecutar en el cliente para obtener la información requerida.

En las ilustraciones de la 39 a la 43, se definen los servicios a monitorear.

```
define service{
    use                generic-service
    host_name          cliente_linux
    service_description Monitorear Disco Duro
    check_command      check_nrpe!check_hda1
}
```

Ilustración 39: Servicio de Disco Duro

```
define service{
    use                generic-service
    host_name          cliente_linux
    service_description Carga del host
    check_command      check_nrpe!check_load
}
```

Ilustración 40: Servicio de Carga del Host

```
define service{
    use                generic-service
    host_name          cliente_linux
    service_description Swap
    check_command      check_nrpe!check_swap
}
```

Ilustración 41: Servicio de Carga de la Partición Swap

```
define service{
    use                generic-service
    host_name          cliente_linux
    service_description Ping
    check_command      check_ping!500.0,20%!800.0,60%
}
```

Ilustración 42: Servicio de Comprobación de Ping

```

define service{
    use                generic-service
    host_name          cliente_linux
    service_description Usuarios activos
    check_command      check_nrpe!check_users
}

```

Ilustración 43: Servicio de Comprobación de Usuarios Activos

Además, es necesario indicar al servidor Nagios que tenga en cuenta el nuevo archivo creado "linux.cfg", para su procesamiento. Con este fin se edita el archivo primario de configuración "nagios.cfg", donde se indica la ruta de este nuevo archivo.

En la ilustración 44, se indica la ruta del archivo "linux.cfg".

```

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

```

Ilustración 44: Ruta Archivo Configuración Cliente Linux

En la ilustración 45, se muestra cómo en la parte del cliente Linux es necesario configurar el archivo del NRPE con los servicios que se desean monitorizar.

```

GNU nano 2.3.1          Fichero: /usr/local/nagios/etc/nrpe.cfg
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -r -w .15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/mapper/centos_cliente-root
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200

command[check_swap]=/usr/local/nagios/libexec/check_swap -w 20% -c 10%
command[check_uptime]=/usr/local/nagios/libexec/check_uptime

# The following examples allow user-supplied arguments and can
# only be used if the NRPE daemon was compiled with support for
# command arguments *AND* the dont_blame_nrpe directive in this
# config file is set to '1'. This poses a potential security risk, so
# make sure you read the SECURITY file before doing this.

```

Ilustración 45: Configuración Archivo NRPE

Por último, en la ilustración 46, se comprueba que los archivos de configuración son correctos.


```

[root@nagios ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.1
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-02-28
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 14 services.
  Checked 2 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

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

```

Ilustración 46: Comprobación Archivos de Configuración Cliente Linux

4.5.2.2. Servicios cliente Windows

En las ilustraciones de la 47 a la 51, se observa cómo se definen en el servidor Nagios los servicios que se quieren monitorizar del cliente Windows.

```

define service {
    use                generic-service
    host_name          cliente_windows
    service_description NSClient++ Version
    check_command      check_nt!CLIENTVERSION
}

```

Ilustración 47: Servicio Comprobar Versión NSClient

```
define service {
    use                generic-service
    host_name          cliente_windows
    service_description Uptime
    check_command      check_nt!UPTIME
}
```

Ilustración 48: Servicio Comprobar Tiempo Activo

```
define service {
    use                generic-service
    host_name          cliente_windows
    service_description CPU Load
    check_command      check_nt!CPULOAD!-l 5,80,90
}
```

Ilustración 49: Servicio Comprobar Carga CPU

```
define service {
    use                generic-service
    host_name          cliente_windows
    service_description Memory Usage
    check_command      check_nt!MEMUSE!-w 80 -c 90
}
```

Ilustración 50: Servicio Comprobar Uso de Memoria

```
define service {
    use                generic-service
    host_name          cliente_windows
    service_description C:\ Drive Space
    check_command      check_nt!USEDISKSPACE!-l c -w 80 -c 90
}
```

Ilustración 51: Servicio Comprobar Espacio en el Disco C:.

En la ilustración 52, se visualiza cómo se edita el archivo primario de configuración “nagios.cfg”, donde se indica la ruta de este nuevo archivo.

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

Ilustración 52: Ruta Archivo Configuración Cliente Windows

En la ilustración 53, se muestra cómo es necesario modificar el archivo NSClient en el cliente Windows y agregar los servicios que se desea monitorizar.

```
nsclient: Bloc de notas
Archivo: Edición Formato Vbr Ayuda
# If you want to fill this file with all available options run the following command:
# nscp settings --generate --add-defaults --load-all
# If you want to activate a module and bring in all its options use:
# nscp settings --activate-module <MODULE NAME> --add-defaults
# For details run: nscp settings --help

; Undocumented section
[modules]

; CheckDisk - CheckDisk can check various file and disk related things. The current version has commands to check Size of hard drives and directories.
CheckDisk = enabled

; Event Log Checker - Check for errors and warnings in the event log. This is only supported through NRPE so if you plan to use only NSClient this wont help you at all.
CheckEventlog = enabled

; Check External Scripts - A simple wrapper to run external scripts and batch files.
CheckExternalScripts = enabled

; Helper function - Various helper function to extend other checks. This is also only supported through NRPE.
CheckHelpers = enabled

; Check NSCP - Checks the state of the agent
CheckNSCP = enabled

; CheckSystem - Various system related checks, such as CPU load, process state, service state memory usage and PDI counters.
CheckSystem = enabled

; CheckDMI - CheckDMI can check various file and disk related things. The current version has commands to check Size of hard drives and directories.
CheckDMI = 1

; NRPE server - A simple server that listens for incoming NRPE connection and handles them.
NRPEServer = enabled

; NSClient server - A simple server that listens for incoming NSClient (check_nt) connection and handles them. Although NRPE is the preferred method NSClient is fully supported and can be used for simplicity or for compatibility.
NSClientServer = enabled

; Undocumented section
[settings/default]

; ALLOWED HOSTS - A comaseparated list of allowed hosts. You can use netmasks (/ syntax) or * to create ranges.
allowed hosts = 192.168.1.200
```

Ilustración 53: Archivo Configuración NSClient

Por último, en la siguiente ilustración, se presenta cómo se comprueba que los archivos de configuración del servidor Nagios son correctos.

```
[root@nagios objects]# nagioscheck

Nagios Core 4.5.1
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-02-28
License: GPL

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

Running pre-flight check on configuration data..

Checking objects..
  Checked 20 services.
  Checked 3 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths..
  Checked 3 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers..
Checking obsessive compulsive processor commands..
Checking misc settings..

Total Warnings: 0
Total Errors: 0

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

Ilustración 54: Comprobación Archivos Configuración Cliente Windows

4.6. Monitoreo cliente Linux

Posteriormente a la realización de las configuraciones pertinentes en los puntos anteriores, al acceder a la interfaz web del servidor Nagios, se puede observar cómo está conectado con el cliente Linux.

A continuación, en la siguiente ilustración, se observa el cliente Linux conectado.

The screenshot shows the Nagios web interface. On the left is a navigation menu with options like 'General', 'Home', 'Documentation', 'Current Status', 'Tactical Overview', 'Map (Legacy)', 'Hosts', 'Services', 'Host Groups', 'Service Groups', 'Problems', and 'Quick Search'. The main content area is titled 'Host Status Details For All Host Groups'. It includes summary statistics for 'Current Network Status', 'Host Status Totals', and 'Service Status Totals'. Below these is a table listing hosts:

Host	Status	Last Check	Duration	Status Information
cliente_linux	UP	05-06-2024 17:43:55	0d 0h 0m 34s	PING OK - Packet loss = 0%, RTA = 0.49 ms
localhost	UP	05-06-2024 17:39:34	7d 22h 7m 44s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Ilustración 55: Listado de Host

En la ilustración 56, se observa cómo se está monitorizando los servicios de manera correcta.

The screenshot shows the Nagios web interface for a specific host. The main content area is titled 'Service Status Details For Host 'cliente_linux''. It includes summary statistics for 'Current Network Status', 'Host Status Totals', and 'Service Status Totals'. Below these is a table listing services:

Service	Status	Last Check	Duration	Attempt	Status Information
Carga del host	OK	05-06-2024 18:33:56	0d 0h 3m 15s	1/3	OK - load average: 0.00, 0.02, 0.05
Monitorrear Disco Duro	OK	05-06-2024 18:33:23	0d 0h 30m 19s	1/3	DISK OK - free space: /var/tmp 15575 MiB (89.54% inode=99%):
Ping	OK	05-06-2024 18:33:42	0d 0h 49m 18s	1/3	PING OK - Packet loss = 0%, RTA = 0.45 ms
Swap	OK	05-06-2024 18:33:54	0d 0h 0m 17s	1/3	SWAP OK - 100% free (2047 MB out of 2047 MB)
Tiempo que lleva host encendido	OK	05-06-2024 18:33:27	0d 0h 23m 15s	1/3	Uptime OK. 0 day(s) 0 hour(s) 50 minute(s)
Usuarios activos	OK	05-06-2024 18:33:56	0d 0h 46m 14s	1/3	USERS OK - 2 users currently logged in

Ilustración 56: Listado de Servicios

En la ilustración 57, se visualiza un mapa de red con el *host* Linux agregado.



Ilustración 57: Mapa de Red

4.7. Monitoreo servidor Windows

Posteriormente a la realización de las configuraciones pertinentes en los puntos anteriores, al acceder a la interfaz web del servidor Nagios, se puede observar cómo están conectados los clientes Linux y Windows.

A continuación, en la siguiente ilustración, se observan los clientes Linux y Windows conectados.

The screenshot shows the Nagios web interface. At the top, there are three summary boxes: 'Current Network Status' (Last Updated: Wed May 8 11:58:22 CEST 2024), 'Host Status Totals' (Up: 3, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 8, Warning: 1, Unknown: 5, Critical: 6, Pending: 0). Below these is a table titled 'Host Status Details For All Host Groups' with the following data:

Host	Status	Last Check	Duration	Status Information
cliente_linux	UP	05-08-2024 11:58:11	0d 0h 0m 11s	PING OK - Packet loss = 0%, RTA = 0.45 ms
cliente_windows	UP	05-08-2024 11:58:05	0d 0h 5m 17s	PING OK - Packet loss = 0%, RTA = 0.40 ms
localhost	UP	05-08-2024 11:57:05	9d 16h 21m 37s	PING OK - Packet loss = 0%, RTA = 0.05 ms

Ilustración 58: Listado de Host

En la ilustración 59, se puede observar cómo se están monitorizando los servicios de manera correcta.

The screenshot shows the Nagios web interface with the 'Service Status Details For Host 'cliente_windows'' table. The summary boxes show 'Current Network Status' (Last Updated: Wed May 8 12:22:09 CEST 2024), 'Host Status Totals' (Up: 1, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 5, Warning: 0, Unknown: 0, Critical: 0, Pending: 0). The table below shows the following data:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
cliente_windows	C:\ Drive Space	OK	05-08-2024 12:20:12	0d 0h 1m 57s	1/3	c - total: 49.34 Gb - used: 11.31 Gb (23%) - free 38.03 Gb (77%)
	CPU Load	OK	05-08-2024 12:15:42	0d 0h 6m 27s	1/3	CPU Load 0% (5 min average)
	Memory Usage	OK	05-08-2024 12:17:10	0d 0h 4m 51s	1/3	Memory usage: total 3195.58 MB - used: 1008.56 MB (32%) - free: 2191.02 MB (68%)
	NSClient++ Version	OK	05-08-2024 12:18:05	0d 23h 6m 32s	1/3	NSClient++ 0.5.2.35.2018-01-28
	Uptime	OK	05-08-2024 12:18:53	0d 0h 3m 16s	1/3	System Uptime - 0 day(s) 0 hour(s) 27 minute(s)

Ilustración 59: Listado de Servicios

En la ilustración 60, se puede percibir a los tres *hosts* con los servicios siendo monitorizados.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
cliente_linux	Carga del host	OK	05-08-2024 12:28:26	0d 0h 29m 33s	1/3	OK - load average: 0.00, 0.01, 0.04
	Monitorear Disco Duro	OK	05-08-2024 12:28:37	0d 0h 30m 22s	1/3	DISK OK - free space: /var/tmp 15575 MB (89.54% inode=99%):
	Ping	OK	05-08-2024 12:28:26	0d 0h 30m 33s	1/3	PING OK - Packet loss = 0%, RTA = 0.41 ms
	Swap	OK	05-08-2024 12:28:23	0d 0h 30m 36s	1/3	SWAP OK - 100% free (2047 MB out of 2047 MB)
	Tiempo que lleva host encendido	OK	05-08-2024 12:28:26	0d 0h 30m 33s	1/3	Uptime OK - 0 day(s) 0 hour(s) 30 minute(s)
	Usuarios activos	OK	05-08-2024 12:28:20	0d 0h 30m 39s	1/3	USERS OK - 1 users currently logged in
cliente_windows	C:\ Drive Space	OK	05-08-2024 12:20:12	0d 0h 8m 47s	1/3	c: - total: 49.34 Gb - used: 11.31 Gb (23%) - free 38.03 Gb (77%)
	CPU Load	OK	05-08-2024 12:25:42	0d 0h 13m 17s	1/3	CPU Load 3% (5 min average)
	Memory Usage	OK	05-08-2024 12:27:18	0d 0h 11m 41s	1/3	Memory usage: total 3199.58 MB - used: 1027.89 MB (32%) - free: 2171.69 MB (68%)
	NSClient++ Version	OK	05-08-2024 12:28:05	0d 23h 13m 22s	1/3	NSClient++ 0.5.2.35 2018-01-28
localhost	Uptime	OK	05-08-2024 12:28:53	0d 0h 10m 6s	1/3	System Uptime - 0 day(s) 0 hour(s) 37 minute(s)
	Current Load	OK	05-08-2024 12:26:53	9d 16h 49m 44s	1/4	OK - load average: 0.00, 0.01, 0.05
	Current Users	OK	05-08-2024 12:27:38	9d 16h 49m 6s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	05-08-2024 12:25:58	9d 16h 48m 29s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.000 second response time
	PING	OK	05-08-2024 12:26:46	9d 16h 47m 49s	1/4	PING OK - Packet loss = 0%, RTA = 0.06 ms
	Root Partition	OK	05-08-2024 12:27:33	9d 16h 52m 14s	1/4	DISK OK - free space: / 15374 MB (88.39% inode=99%):
	SSH	OK	05-08-2024 12:28:21	9d 16h 51m 36s	1/4	SSH OK - OpenSSH_7.4 (protocol 2.0)
	Swap Usage	OK	05-08-2024 12:24:09	9d 16h 50m 59s	1/4	SWAP OK - 100% free (2047 MB out of 2047 MB)
	Total Processes	OK	05-08-2024 12:28:04	9d 16h 50m 20s	1/4	PROCS OK. 41 processes with STATE = RSZDT

Ilustración 60: Monitorización Servicios de los 3 Hosts

En la ilustración 61, se visualiza un mapa de red con los *hosts* Windows y Linux agregados.

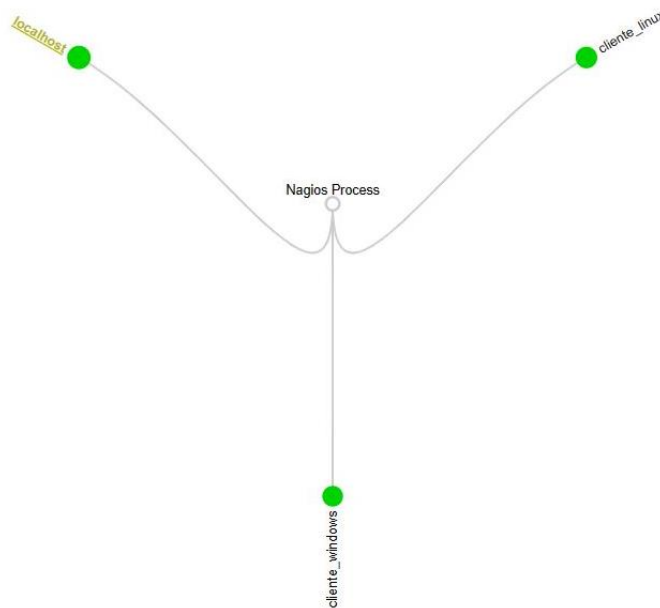


Ilustración 61: Mapa de Red

5. Posibles casos de uso

En puntos anteriores como el 4.5.2, se ha comentado cómo se agregan servicios de monitoreo de Nagios. En el presente punto se verán otros casos de uso que pueden surgir en Nagios.

5.1. Caso 1: Creación *plugin* personalizado de control de memoria RAM

A diferencia de los *plugins* utilizados anteriormente, que son oficiales de Nagios, los *plugins* personalizados son creados desde cero para un fin en específico. En este caso se crea un *plugin* personalizado para el monitoreo de la memoria RAM de un cliente Linux.

Como se puede observar en la ilustración 62, en el directorio donde están los *plugins* no se encuentra ninguno para el monitoreo de la RAM en un cliente Linux.

```
[root@nagios libexec]# ls
check_apt      check_dummy    check_imap     check_nntp     check_pop      check_ssl_validity  negate
check_breeze  check_file_age check_ircd     check_nrpe    check_procs    check_ssmt         remove_perfddata
check_by_ssh  check_flexlm   check_ircd     check_nt       check_real     check_swap         urlize
check_clamd   check_ftp      check_load    check_ntp      check_rpc      check_tcp          utils.pm
check_cluster check_hpjd     check_log     check_ntp_peer check_sensors  check_time         utils.sh
check_dhcp    check_http    check_mailq   check_ntp_time check_simap    check_udp
check_dig     check_icmp    check_mrtg    check_nwstat  check_smtp     check_ups
check_disk    check_ide_smart check_mrtgtraf check_oracle  check_snmp     check_uptime
check_disk_smb check_ifoperstatus check_nagios  check_ovncr   check_spop     check_users
check_dns     check_ifstatus check_nntp    check_ping    check_ssh      check_wave
```

Ilustración 62: Plugins Oficiales

En la siguiente ilustración, se visualiza cómo se crea el archivo donde está el código del *plugin*. Este archivo es “.pl” ya que se trata de un archivo programado en perl.

```
[root@nagios libexec]# nano check_mem.pl
```

Ilustración 63: Creación Archivo Perl

En la ilustración 64, se puede observar parte del código que compone el *plugin*.

```
sub tell_nagios {
    my ($used,$free,$caches,$hugepages) = @_ ;

    # Calculate Total Memory
    my $total = $free + $used;
    print "$total Total\n" if ($opt_v);

    my $perf_warn;
    my $perf_crit;
    if ( $opt_u ) {
        $perf_warn = int($total) * $opt_w / 100;
        $perf_crit = int($total) * $opt_c / 100;
    } else {
        $perf_warn = int($total) * ( 100 - $opt_w ) / 100;
        $perf_crit = int($total) * ( 100 - $opt_c ) / 100;
    }

    my $perfdata = "|TOTAL=${total}KB;;; USED=${used}KB;${perf_warn};${perf_crit};; FREE=${free}KB;;; CACHES=${caches}KB;;;";
    $perfdata .= " HUGE_PAGES=${hugepages}KB;;; if ($opt_h);

    if ($opt_f) {
        my $percent = sprintf "%.1f", ($free / $total * 100);
        if ($percent <= $opt_c) {
            finish("CRITICAL - $percent% ($free kB) free!$perfdata", $exit_codes{'CRITICAL'});
        }
        elsif ($percent <= $opt_w) {
            finish("WARNING - $percent% ($free kB) free!$perfdata", $exit_codes{'WARNING'});
        }
        else {
            finish("OK - $percent% ($free kB) free.$perfdata", $exit_codes{'OK'});
        }
    }
    elsif ($opt_u) {
        my $percent = sprintf "%.1f", ($used / $total * 100);
        if ($percent >= $opt_c) {
            finish("CRITICAL - $percent% ($used kB) used!$perfdata", $exit_codes{'CRITICAL'});
        }
        elsif ($percent >= $opt_w) {
            finish("WARNING - $percent% ($used kB) used!$perfdata", $exit_codes{'WARNING'});
        }
        else {
            finish("OK - $percent% ($used kB) used.$perfdata", $exit_codes{'OK'});
        }
    }
}
```

Ilustración 64: Fragmento de Código Plugin en Perl Personalizado

A fin de realizar la ejecución del archivo mencionado, es necesario otorgar permisos de ejecución, para ello se usa el comando que se encuentra a continuación:

- **chmod:** Comando utilizado para cambiar los permisos de archivos y directorios en sistemas Unix y Linux.
- **a+x:** Especifica los permisos que se van a cambiar. En este caso:
 - **a:** Significa "all" (todos). Se refiere a todos los usuarios: el propietario del archivo, el grupo del archivo y otros usuarios.
 - **+x:** Agrega el permiso de ejecución al archivo. El permiso de ejecución permite que el archivo sea ejecutado como un programa.
- **check_mem.pl:** Nombre del archivo al que se le están cambiando los permisos.

En la siguiente ilustración, se otorgan los permisos al archivo.

```
[root@nagios libexec]# chmod a+x check_mem.pl
```

Ilustración 65: Permisos Archivo Perl

Si se ejecuta el archivo en perl creado anteriormente, este devuelve la sintaxis para saber cómo emplearlo.

- **-<f|u>**: Si se desea comprobar por la memoria libre o por la memoria utilizada.
- **-w y -c**: Los umbrales por los que se quiere comprobar.
- **-f**: Verifica la memoria libre en porcentaje.
- **-u**: Verifica la memoria utilizada en porcentaje.
- **-C**: Comprueba la memoria cache libre.
- **-h**: Comprueba la memoria libre utilizada.

En la siguiente ilustración, se visualiza la sintaxis del *plugin*.

```
[root@nagios libexec]# perl check_mem.pl

check_mem.pl v1.0 - Nagios Plugin

usage:
  check_mem.pl -<f|u> -w <warnlevel> -c <critlevel>

options:
  -f          Check FREE memory
  -u          Check USED memory
  -C          Count OS caches as FREE memory
  -h          Remove hugepages from the total memory count
  -w PERCENT  Percent free/used when to warn
  -c PERCENT  Percent free/used when critical

Copyright (C) 2000 Dan Larsson <dl@tyfon.net>
check_mem.pl comes with absolutely NO WARRANTY either implied or explicit
This program is licensed under the terms of the
MIT License (check source code for details)
```

Ilustración 66: Opciones Sintaxis Plugin

En la siguiente ilustración, se puede observar un ejemplo de utilización del *plugin*, en este caso devuelve la memoria libre utilizada en porcentaje.

```
[root@nagios libexec]# perl check_mem.pl -f -w 50 -c 30
OK - 75.1% (1412688 kB) free. |TOTAL=1882084KB;;; USED=469396KB;941042;1317458;; FREE=1412688KB;;; CACHES=202440KB;;;
```

Ilustración 67: Ejemplo Sintaxis y Prueba Plugin

En la ilustración 68, se agrega esta línea al archivo NRPE de Nagios para indicar el comando que ejecutará.

```
command[check_mem]= perl /usr/local/nagios/libexec/check_mem.pl -f -C -w 50 -c 30
```

Ilustración 68: Archivo Configuración NRPE

En la siguiente ilustración, se muestra cómo se define el servicio que se va a monitorizar.

```
define service{
    use                generic-service
    host_name          Nagios_Server
    check_interval     1
    service_description Carga memori RAM
    check_command      check_nrpe!check_mem
}
```

Ilustración 69: Definición Servicio a Monitorrear

En la ilustración 70, se observa la interfaz web. Se puede visualizar el *plugin* ya agregado y en funcionamiento.

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
Nagios_Server	Carga memori RAM	OK	07-06-2024 15:40:03	0d 0h 1m 32s	1/3	OK - 84.7% (1594600 kB) free.
	ping	OK	07-06-2024 15:39:26	0d 0h 40m 10s	1/3	PING OK - Packet loss = 0%, RTA = 0.04 ms

Ilustración 70: Plugin en Funcionamiento

5.2. Caso 2: Monitoreo puerto específico

En este caso de uso se va a tratar cómo monitorizar un puerto específico en un *host* determinado. Esto es muy útil, ya que el monitoreo de un puerto específico es utilizado si un servicio determinado lo emplea o si un *host* externo no hace *ping*. También se pueden consultar los puertos abiertos para hacer otro tipo de consulta.

En la ilustración 71, se consulta el *plugin* que se va a utilizar y las opciones de sintaxis con las que se puede usar.

```
[root@nagios libexec]# ./check_tcp
check_tcp: No arguments found
Usage:
check_tcp -H host -p port [-w <warning time>] [-c <critical time>] [-s <send string>]
[-e <expect string>] [-q <quit string>] [-m <maximum bytes>] [-d <delay>]
[-t <timeout seconds>] [-r <refuse state>] [-M <mismatch state>] [-v] [-4|-6] [-j]
[-D <warn days cert expire>[,<crit days cert expire>]] [-S <use SSL>] [-E]
[-N <server name indication>]
[root@nagios libexec]#
```

Ilustración 71: Plugin que se utiliza

En la ilustración 72, se visualiza un caso de uso del *plugin*.

- **./check_tcp**: Indica el *plugin* que se utiliza.
- **-H**: Determina la IP del *host* al que se le hace la consulta.
- **-p**: Señala el puerto al que se le va a hacer la consulta.

```
[root@nagios libexec]# ./check_tcp -H 192.168.0.201 -p 22
TCP OK - 0,001 second response time on 192.168.0.201 port 22|time=0,001086s;;;0,000000;10,000000
```

Ilustración 72: Ejemplo de uso del plugin

En la ilustración 73, se observa cómo se define el comando en el archivo “commands.cfg”.

- **define command:** Define el comando.
- **command_name:** Indica el nombre del comando con el que se invoca.
- **command_line:** Línea de ejecución.
 - **\$USER1\$:** Significa /usr/local/nagios/libexec, es una abreviación.
 - **check_tcp -H:** Es un *plugin* que se utiliza para enviar comandos y consultar el estado de los servicios y los recursos en un *host* remoto que tiene instalado y configurado el servidor NRPE.
 - **\$HOSTADDRESS\$ -c:** Dirección IP del servidor que se está monitoreando, esta IP es utilizada por el *plugin* check_nrpe -H.
 - **\$ARG1\$:** Indica el *plugin* que se quiere chequear.

```
define command {  
  
    command_name    check_tcp  
    command_line    $USER1$/check_tcp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$  
}
```

Ilustración 73: Se define el comando

En la ilustración 74, se percibe como en el archivo “linux.cfg” se crea el servicio. Este *plugin* no hace uso del NRPE, ya que se ejecuta desde el servidor Nagios y este observa el puerto del *host* al que se quiere monitorizar.

- **use:** Define qué servicio se quiere utilizar. Esta utiliza algunos de sus parámetros del archivo *templates*.
- **host_name:** Nombre del *host* a monitorizar definido anteriormente.
- **service_description:** Breve descripción del servicio.
- **check_command:** Comando que se ejecuta con el puerto indicado.

```
define service{  
    use                generic-service  
    host_name          cliente_linux  
    check_interval     1  
    service_description Monitoreo puerto especifico  
    check_command      check_tcp!22  
}
```

Ilustración 74: Se define el servicio

En la siguiente ilustración, se observa cómo se está monitorizando el puerto 22 y este está abierto.

Limit Results:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
cliente_linux	Carga del host	OK	07-06-2024 13:02:51	0d 0h 34m 45s	1/3	OK - load average: 0.01, 0.02, 0.05
	Monitorear Disco Duro	OK	07-06-2024 13:03:10	0d 0h 37m 26s	1/3	DISK OK - free space: /var/tmp 15574 MIB (89.54% inode=99%)
	Monitoreo puerto especifico Puerto 22	OK	07-06-2024 13:02:44	0d 0h 5m 52s	1/3	TCP OK - 0.001 second response time on 192.168.0.201 port 22
	Ping	OK	07-06-2024 13:03:05	0d 0h 37m 31s	1/3	PING OK - Packet loss = 0%, RTA = 0.44 ms
	Swap	OK	07-06-2024 13:02:34	0d 0h 38m 2s	1/3	SWAP OK - 100% free (2047 MB out of 2047 MB)
	Tiempo que lleva host encendido	OK	07-06-2024 13:03:26	0d 0h 37m 10s	1/3	Uptime OK: 0 day(s) 0 hour(s) 38 minute(s)
	Usuarios activos	OK	07-06-2024 13:02:57	0d 0h 37m 39s	1/3	USERS OK - 2 users currently logged in

Ilustración 75: Interfaz web con el plugin funcionando

6. Presupuesto

Para la presupuestación de este proyecto se supone que su implantación es en un servidor físico dedicado, de una empresa pequeña con unos 50 equipos aproximadamente. Teniendo en cuenta lo comentado anteriormente, el presupuesto se desglosa en los siguientes *ítems*:

- **Nagios Core – 0 €:** Teniendo en cuenta que se hace uso de esta versión gratuita, no tiene costes.
- **Servidor físico – 540 €:** Para la implementación de Nagios se utiliza un servidor físico dedicado, ateniéndose a sus requisitos de funcionamiento.
- **Mano de obra – 3750 €:** Para la implementación de Nagios y su correcta configuración, se tiene en cuenta que esto lo realiza un Ingeniero/Técnico senior con conocimientos previos, lo que implica un coste de 75 €/hora. Haciendo estimaciones serían necesarias unas 50 horas de trabajo para una implementación completa y customizada.

El coste total de este proyecto se estima en unos **4.290,00€**.

7. Conclusiones

A lo largo del desarrollo de este proyecto, se han obtenido las siguientes conclusiones:

1. Eficiencia en la monitorización de sistemas

A lo largo del proyecto, Nagios ha demostrado ser una solución eficiente y eficaz para la monitorización de sistemas y redes. Gracias a su capacidad para detectar problemas y alertar a los administradores del sistema, permite una respuesta rápida a las incidencias.

2. Flexibilidad y Extensibilidad

Al tener Nagios una arquitectura modular, junto con un amplio soporte de *plugins*, este permite una gran flexibilidad y adaptabilidad a diversos entornos de TI. Esto facilita la integración de Nagios en infraestructuras, permitiendo la monitorización de una amplia gama de dispositivos y aplicaciones.

3. Facilidad de uso y configuración

La curva inicial de Nagios es elevada, pero su configuración y uso se vuelven más manejables con el tiempo. También se ha podido observar a lo largo del desarrollo de este proyecto, que gracias a su amplia documentación y comunidad permiten un soporte considerable para la resolución de problemas.

4. Presupuesto económico

A la hora de la elección de este proyecto se tuvo en cuenta que Nagios es una herramienta más económica que su competencia. Posteriormente de la finalización del proyecto, el estudio de la herramienta y parte de su competencia se puede corroborar que esto es cierto.

Se puede concluir que Nagios es una herramienta integral y versátil, que bien implementada y gestionada puede proporcionar una monitorización exhaustiva y precisa de infraestructuras de TI complejas. Este TFG ha mostrado que, con la formación y práctica adecuadas, los profesionales de TI pueden aprovechar al máximo las capacidades de Nagios para mejorar significativamente la gestión y operación de sus sistemas.

Con la finalización de este proyecto, mi objetivo a futuro para la continuación de este trabajo sería implementar algunas de las siguientes funcionalidades:

- **Integración con Herramientas de Automatización:**

Ansible, Puppet o Chef: Investigar la integración de Nagios Core con herramientas de automatización para facilitar la configuración y el despliegue masivo de agentes de monitorización.

- **Monitorización en Entornos de Contenedores:**

Kubernetes y Docker: Implementar y evaluar la efectividad de Nagios Core en la monitorización de entornos de contenedores, incluyendo la creación de *plugins* específicos para Kubernetes y Docker.

- **Análisis de Datos y *Machine Learning*:**

Análisis Predictivo: Utilizar técnicas de *machine learning* para predecir posibles fallos y optimizar el rendimiento de la infraestructura basada en los datos históricos recogidos por Nagios Core.

Bibliografía

- [1] P.E. Esquivel León, M.A. Astudillo Pesántez, "Monitoreo de red implementando Nagios", (trabajo académico). Universidad del Azuay (Ecuador), 2006.
<https://dspace.uazuay.edu.ec/handle/datos/2152>
- [2] *Nagios Core / Nagios Open Source*. (s. f.). Nagios Open Source.
<https://www.nagios.org/projects/nagios-core/> (20/05/2024).
- [3] *Nagios, 2nd Edition*. (s. f.). Google Books.
https://books.google.es/books?hl=es&lr=&id=QgYvDwAAQBAJ&oi=fnd&pg=PA5&dq=nagios&ots=hVq_afDPwb&sig=PDqsVPKfII4EMMA3nUSSNkfxd0k#v=onepage&q&f=false
- [4] *Learning Nagios 4*. (s. f.). Google Books.
<https://books.google.es/books?hl=es&lr=&id=pdgiAwAAQBAJ&oi=fnd&pg=PT9&dq=nagios&ots=Z5Dp9FTCLf&sig=IKWv6ulMUgB7ecqkVDDy-cnHQGg#v=onepage&q=nagios&f=false>
- [5] *Network's server monitoring and analysis using Nagios*. (2017, 1 marzo). IEEE Conference Publication | IEEE Xplore.
<https://ieeexplore.ieee.org/abstract/document/8300092>
- [6] *Building a Service-Oriented Monitoring Framework with REST and Nagios*. (2011, 1 julio). IEEE Conference Publication | IEEE Xplore.
<https://ieeexplore.ieee.org/abstract/document/6009290>
- [7] *Using Nagios as a groundwork for developing a better network monitoring system*. (2012, 1 julio). IEEE Conference Publication | IEEE Xplore.
<https://ieeexplore.ieee.org/abstract/document/6304293>