

Facultat d'Informàtica de Barcelona  
Departament d'Arquitectura de Computadors  
Centres de Processament de Dades

# Activitat AiP

Monitorització

Orellana Celis, Nicolás Ignaci  
Rodríguez González, Isis

Tema 15

Data: 08/04/2021

# **ÍNDICE**

<b>Introducción</b>	<b>2</b>
¿Qué son las herramientas de monitorización?	2
Historia	3
<b>Nagios</b>	<b>7</b>
Descripción del software	7
Componentes de Nagios	7
Despliegue de Nagios	8
Instalación Nagios	10
Configurar los servicios para conectarse con Nagios	14
Configuración fichero servidor HTTP	16
Configuración fichero servidor FTP	16
Configuración fichero servidor SSH	17
Ventajas de Nagios	18
Desventajas de Nagios	18
<b>Ganglia</b>	<b>19</b>
Descripción del software	19
Componentes de Ganglia	19
Despliegue de Ganglia	19
Instalación de Apache2 HTTP Server en el Servidor	21
Instalación de PHP 7.2 y módulos relacionados en el Servidor	22
Instalación del paquete de Ganglia	25
Configuración del servicio de Ganglia	26
Acceso al portal web de Ganglia	27
Instalar Ganglia en el cliente	28
Comprobar la monitorización de Ganglia	29
Ventajas de Ganglia	30
Desventajas de Ganglia	30
<b>Conclusiones</b>	<b>31</b>

# 1. Introducción

## 1.1. ¿Qué son las herramientas de monitorización?

La monitorización de sistemas informáticos comprende una amplia clase de productos diseñados para permitir que los ingenieros informáticos determinen si los equipos de la empresa están encendidos y funcionan a los niveles de servicio esperados (SLA). Además también pueden servir para resolver problemas detectados en estos sistemas de monitorización.

Estas herramientas de monitorización van desde comprobaciones básicas (comprobar que un servicio está corriendo) hasta herramientas más avanzadas, que pueden examinar de forma exhaustiva el rendimiento de un servicio/servidor.

## 1.2. Monitorización de disponibilidad

La monitorización de disponibilidad, también conocida como monitorización del sistema, nace de la idea de que el tiempo de inactividad de nuestros sistemas informáticos provoca pérdidas, muchas veces de miles de dólares, a la empresa. La monitorización de disponibilidad evita este tipo de situaciones al verificar la actividad de los componentes de la infraestructura de nuestros sistemas informáticos, como servidores o aplicaciones. En caso de haber alguna caída, el sistema notificará al administrador de sistemas del problema con el equipo, evitando el retraso en la detección del error, y dando una gran ventaja al darle al ingeniero un tiempo para poder maniobrar.

## 1.3. Monitorización de rendimiento web

Se trata de un subconjunto de la monitorización de disponibilidad, la monitorización de rendimiento web está diseñado para monitorizar la disponibilidad de un servidor o servicio web, pero también agrega detalles más detallados al sistema.

Estas herramientas recogen información como el tiempo de carga de la página, la ubicación de los errores que se generan y los tiempos de carga individuales de varios elementos web, lo que ayuda a las personas encargadas de analizar la web a ajustar su sitio web.

**1.4. Gestión de aplicaciones / gestión del rendimiento de las aplicaciones (APM por sus siglas en inglés)**

Las herramientas APM son similares a las herramientas de monitorización del rendimiento web, pero están diseñadas teniendo en cuenta las aplicaciones orientadas al cliente, lo que permite a los encargados del sitio web realizar un seguimiento del rendimiento de una aplicación y detectar cualquier problema antes de que se vuelva demasiado grave para los usuarios. Las herramientas de APM más modernas pueden incluir rutinas automatizadas para solucionar estos problemas sin la intervención de un desarrollador.

**1.5. Monitorización de API**

Las empresas que ofrecen API's a desarrolladores externos encontrarán crucial garantizar el tiempo de actividad de estos servicios. Las herramientas de monitorización de API y el software de monitorización nos dan información sobre si una API funciona correctamente, lo que garantiza un tiempo de inactividad mínimo.

**1.6. Monitorización de usuarios reales (RUM por sus siglas en inglés)**

La monitorización de usuarios reales está diseñado para registrar las interacciones reales del usuario final con un sitio web o una aplicación. El monitorizar los tiempos de carga y el comportamiento del usuario, puede identificar problemas basados en desafíos de experiencia de usuario “reales”, a diferencia de hacerlo mediante simulaciones. Este tipo de monitorización está diseñado para ser retrospectivo, no predictivo, lo que permite a los encargados de la monitorización detectar problemas solo después de que ocurran.

**1.7. Monitorización de seguridad**

La monitorización de seguridad es un tipo de monitorización de IT altamente específico, diseñado para observar una red en busca de intentos de pentesting u otra actividad inusual. La monitorización de seguridad es una categoría amplia y de alto nivel que incluye numerosos subconjuntos de herramientas de análisis de seguridad.

**1.8. Historia**

Para empezar antes de los años 90 podemos concluir que en la época de los primeros miniordenadores(PC), había una total ausencia de monitorización. Los sistemas operativos de vez en cuando disponían de ciertos mecanismos de monitorización interna como para poder administrar cosas como la memoria virtual. Las herramientas de monitorización como las conocemos hoy en día eran primitivas y sus resultados eran poco más que volcados de memoria y registros.

Esto no es de extrañar debido a que los sistemas eran orientados en batch(sistemas por lotes), de forma que se disponía de muy poca entrada o salida en tiempo real. Lo normal hace muchos años era tener expertos que podían interpretar las luces de las grandes máquinas en el panel de control.

Después del nacimiento de los PC y el nacimiento de Unix, hay que decir que fue fundamental cuando se trataba de realizar la transición de los sistemas operativos orientados a batch a los sistemas interactivos/en tiempo real.

Con Unix nacieron muchos de los primeros comandos y herramientas de monitorización básicas( por ejemplo vmstat, syslog, fuser o top). Desde la década de 1990, estos componentes fundamentales de supervisión se convirtieron en una parte estándar tanto de Linux como de Unix

También fue durante los años 90 cuando las herramientas de monitorización interactivas en tiempo real se convirtieron en un estándar de la mayoría de los sistemas operativos de sobremesa. El Performance Monitor / System Monitor que nació con Windows NT 3.1 se convirtió en una parte estándar de Windows tanto de 32 bits como de 64 bits. A finales de los 90, las herramientas de monitorización gráficas también se incluyeron en la mayoría de los entornos de escritorio Linux/Unix.

Volviendo un poco hacia atrás, también cabe destacar que en los años 80, aunque más en los años 90, también se vio el desarrollo de herramientas de monitorización de red como nmon o Big Brother

Si bien las herramientas de monitoreo de escritorio generalmente podían permitirse enfocarse en un solo sistema y un solo usuario, las herramientas de monitoreo de red enfrentaron un desafío más amplio. Tenían que realizar un seguimiento no solo del rendimiento del hardware de la red y del software de gestión de la red, sino también de las actividades de varios usuarios.

Esto significó monitorizar el estado y el rendimiento de múltiples interfaces, así como el hardware del servidor y los recursos del sistema, mientras que al mismo tiempo se proporcionaba el tipo de datos de tráfico que permitirían al sistema administrar adecuadamente la carga de usuarios.

A finales de la década de los noventa, la mayoría de las herramientas de monitoreo en uso se habían desarrollado bajo el supuesto de que iban a usarse para monitorear una red de área local o su equivalente, con un número relativamente limitado de usuarios en un entorno administrado de cerca.

Sin embargo, a principios del siglo XXI, se hizo evidente que las necesidades de supervisión de los sitios web y los servicios basados en Internet no eran las mismas que las de una LAN de oficina típica. Esto condujo inicialmente al desarrollo de una generación de herramientas de monitoreo (como Cacti, Nagios y Zabbix) que admitían protocolos de Internet estándar, podían usarse en múltiples plataformas, a menudo eran bastante escalables y, por lo general, tenían interfaces basadas en web.

Sin embargo, estas herramientas todavía se enfocan generalmente en métricas funcionales y de desempeño, con un fuerte énfasis en el hardware de servidores y comunicaciones y problemas relacionados. Ampliaron el alcance de las herramientas de supervisión de redes más antiguas, pero conservaron gran parte de la naturaleza básica de esas herramientas. La primera década del siglo XXI veía la creciente necesidad de un nuevo tipo de herramienta de seguimiento.

El desafío básico de principios del siglo XXI era el siguiente: para cada vez más organizaciones, Internet ya no era una salida alternativa u opcional para hacer negocios; ahora era su plataforma principal (y a veces la única).

Junto con todos los problemas estándar de funcionamiento, surgió la necesidad de monitorizar métricas relacionadas con el negocio. Era tan importante conocer el flujo de tráfico de una página (o un elemento dentro de una página) a la siguiente, el patrón de tráfico a lo largo del tiempo y la fuente geográfica de ese tráfico como saber si el servidor estaba manejando el tráfico adecuadamente.

Incluso con el seguimiento funcional, las prioridades estaban cambiando. El tráfico anómalo en el carrito de la compra o en las páginas de autorización podría ser una señal de un error potencialmente catastrófico o de un robo. A medida que los sitios web de negocios se convirtieron en tiendas, debían observarse de la misma manera que se veía una tienda física, debido a muchos de los mismos problemas potenciales.

Tanto la naturaleza como el volumen de los datos de monitoreo cambiaron a medida que más negocios se trasladaron a Internet. Más clientes en línea y más clientes significaban más datos de clientes, y esa creciente cantidad de datos tenía que ser analizada, si quería ser de alguna utilidad. La monitorización se estaba convirtiendo no solo en monitorización, sino en monitorización más análisis orientado al mercado.

El siguiente paso, por supuesto, fue que el comercio en línea se trasladara a la nube, y ahí es donde nos encontramos hoy. El movimiento mayorista hacia la nube ha transformado la naturaleza de las herramientas de monitorización. En una implementación basada en la nube, por ejemplo, no hay necesidad de monitorizar los problemas relacionados con el hardware (a menos que, por supuesto, seamos el proveedor de servicios en la nube).

El rendimiento sigue siendo importante, pero cuando se supervisa el rendimiento en la nube, necesariamente debemos hacerlo en el contexto del software y la infraestructura virtualizada.

## 2. Nagios



### 2.1. Descripción del software

Nagios es un sistema de monitorización de código abierto para todo tipo de equipos, incluidos los CPDs. Fue diseñado para ejecutarse Linux y puede monitorizar dispositivos que corran sobre sistemas operativos diversos como pueden ser Linux, Windows y Unix. El software de Nagios realiza comprobaciones periódicas sobre ciertas características críticas de las aplicaciones, sobre la red y sobre cualquier tipo de recurso del sistema.

Por ejemplo, Nagios puede monitorear el uso de la memoria, el uso del disco, la carga del microprocesador, la cantidad de procesos en ejecución y archivos de registro. Nagios también puede monitorear servicios, como servidores de correo (SMTP), boxes de correo (POP3), HTTP, entre otros.

### 2.2. Componentes de Nagios

#### Nagios Core

Nagios Core es lo que se podría llamar el cerebro de Nagios, ya que es el centro del software. Es la aplicación de monitorización de red basada en código abierto. Es el encargado de monitorizar los hosts y los servicios que queramos monitorizar e indicar cuando pasa algo relativo que requiera nuestra atención.

Nagios core se diseño para ejecutarse en sistemas Linux mayoritariamente, pero debería funcionar sin problemas en la mayoría de sistemas Unix.



En cuanto las funcionalidades principales de Nagios Core podemos observar las siguientes (no son todas):

- Monitorización de servicios de red (SMTP, POP3, HTTP, ICMP, etc)
- Supervisión de los recursos de los hosts como puede ser la carga del procesador o el uso del disco
- Notificaciones al administrador cuando se produzcan problemas en algún servicio o en algún host (podría ser por correo electrónico, aunque también hay otros métodos).
- Interfaz web (no obligatoria) para poder ver el estado actual de la red, los logs, archivos de registro, etc...

Este software se puede instalar en cualquier sistema Linux o variante de Unix y que disponga de un compilador de C (en caso de instalar por código fuente).

Adicionalmente, en caso de querer disponer de la opción de poder acceder a Nagios mediante la interfaz web se necesitaría un Apache2 y la librería gd en su versión 1.6.3 o superior.

Nagios XI  
Nagios Log Server  
Nagios Network Analyzer  
Nagios Fusion

### **2.3. Despliegue de Nagios**

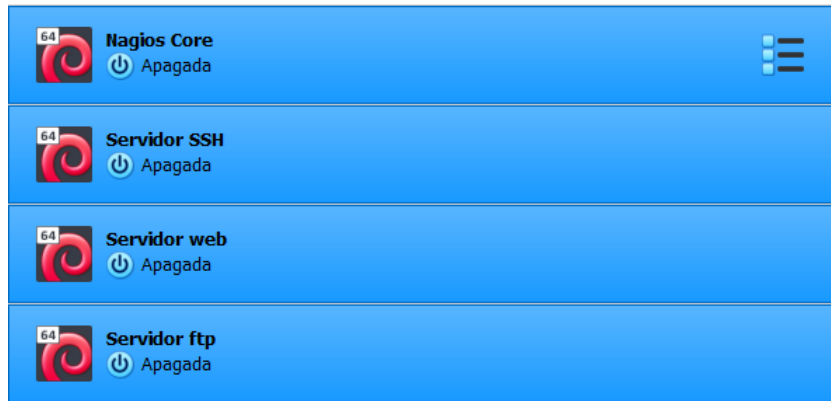
Para empezar con el despliegue de Nagios, empezaremos describiendo el entorno que hemos escogido y los servicios que hemos elegido monitorizar.

En este caso hemos escogido como sistema operativo Debian debido a la familiaridad que tenemos con él y la facilidad para poder instalar Nagios dentro de este sistema operativo.










En cuanto a la versión del sistema operativo hemos escogido Debian 10 buster.

También añadir que a la hora de realizar la prueba de monitorización hemos elegido monitorizar un servidor FTP, un servidor SSH y un servidor web (HTTP).

A continuación tenemos una imagen de las máquinas virtuales que hemos creado.



A continuación las especificaciones de la máquina utilizado para clonar el resto:

	<b>General</b>
Nombre:	debian_10_limpio
Sistema operativo:	Debian (64-bit)
	<b>Sistema</b>
Memoria base:	2048 MB
Orden de arranque:	Disquete, Óptica, Disco duro
Aceleración:	VT-x/AMD-V, Paginación anidada, Paravirtualización KVM
	<b>Pantalla</b>
Memoria de vídeo:	16 MB
Controlador gráfico:	VMSVGA
Servidor de escritorio remoto:	Inhabilitado
Grabación:	Inhabilitado
	<b>Almacenamiento</b>
Controlador:	IDE
IDE secundario maestro:	[Unidad óptica] Vacío
Controlador:	SATA
Puerto SATA 0:	debian_10_limpio.vdi (Normal, 12,00 GB)
	<b>Audio</b>
Controlador de anfitrión:	Windows DirectSound
Controlador:	ICH AC97
	<b>Red</b>
Adaptador 1:	Intel PRO/1000 MT Desktop (NAT)
	<b>USB</b>
Controlador USB:	OHCI, EHCI
Filtros de dispositivos:	0 (0 activo)
	<b>Carpetas compartidas</b>
	<b>Descripción</b>
	Ninguno

## Instalación Nagios

Una vez estemos seguros que todas las máquinas arranquen, nos aseguraremos que estén actualizados los repositorios y los paquetes estén su última versión con el siguiente comando:

```
debian@debian:~$ sudo apt update && sudo apt -y upgrade
[sudo] password for debian:
Obj:1 http://deb.debian.org/debian buster InRelease
Obj:2 http://deb.debian.org/debian buster-updates InRelease
```

Acto seguido tenemos que instalar los paquetes gcc, make, wget y unzip (si no se tiene alguno de estos ya previamente instalados, aunque probablemente se tengan algunos)

```
debian@debian:~$ sudo apt install gcc make unzip wget
[sudo] password for debian:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
unzip ya está en su versión más reciente (6.0-23+deb10u2).
fijado unzip como instalado manualmente.
wget ya está en su versión más reciente (1.20.1-1.1).
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu gcc-8 libasan5 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libgcc-8-dev libitm1 liblsan0 libmpx2 libtsan0
  libubsan1 linux-libc-dev manpages-dev
```

Lo primero para realizar la instalación de Nagios es instalar los plugins propios de Nagios.

Esto será necesario hacerlo en todas las máquinas, es decir, tanto en el servidor como en las máquinas sobre las que realizaremos la monitorización.

Estos plugins sirven para poder realizar la monitorización correctamente mediante los comandos propios de Nagios.

```
debian@debian:~$ wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
--2021-03-28 04:12:29-- https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
Resolviendo nagios-plugins.org (nagios-plugins.org)... 72.14.186.43
Conectando con nagios-plugins.org (nagios-plugins.org)[72.14.186.43]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2782610 (2,7M) [application/x-gzip]
Grabando a: "nagios-plugins-2.3.3.tar.gz"

nagios-plugins-2.3. 37%[=====>] 1,01M 414KB/s
```

```
debian@debian:~$ tar xvf nagios-plugins-2.3.3.tar.gz
nagios-plugins-2.3.3/
```

```

debian@debian:~/nagios-plugins-2.3.3$ sudo apt install -y postgresql-server-dev-all libdbi-dev libldap2-dev
ault-libmysqclient-dev libssl-dev dnsutils smbclient qstat fpings smbclient qstat fping
[sudo] password for debian:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 binfmt-support clang-7 dctrl-tools lib32gcc1 lib32stdc++6 libc6-i386 libclang-common-7-dev libclang1-7
 libdbi1 libffi-dev libgmp-dev libgmpxx4ldbl libgnutls-dane0 libgnutls-openssl27 libgnutls28-dev
 libgnutlsxx28 libidn2-dev liblrs161 libmariadb-dev libmariadb-dev-compat libncurses-dev libobjc-8-dev
 libomp-7-dev libomp5-7 libp11-kit-dev libpq-dev libstdc++-8-dev libtasn1-6-dev libtasn1-doc libtinfo-dev
 libunbound8 llvm-7 llvm-7-runtime nettle-dev postgresql-client-common postgresql-common
 postgresql-server-dev-11 python-crypto python-gpg python-ldb python-samba python-tdb samba-common
 samba-common-bin samba-dsdb-modules zlib1g-dev

```

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

```
debian@debian:~/nagios-plugins-2.3.3$ /usr/local/nagios/libexec/check_disk -w 20% -c %5 -p /
DISK OK - free space: / 4129 MiB (39,37% inode=71%);| /6357MiB;8854;11068;0;11068
debian@debian:~/nagios-plugins-2.3.3$
```

```
debian@debian:~/nagios-plugins-2.3.3$ wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.4.6/nagios-4.4.6.tar.gz
--2021-03-28 04:35:08-- https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.4.6/nagios-4.4.6.tar.gz
Resolviendo github.com (github.com)... 140.82.121.3
Conectando con github.com (github.com)[140.82.121.3]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización: https://github-releases.githubusercontent.com/16119670/6195cc80-8969-11ea-9b68-e3a8c5b15c0e?%3F=nagios-4.4.6.tar.gz
debian@debian:~$ tar xf nagios-4.4.6.tar.gz
debian@debian:~$ cd nagios-4.4.6/
```

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

11

*sudo apt install -y libgd-dev*

```
*** Configuration summary for nagios 4.4.6 2020-04-28 ***:

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

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/bin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.
```

Hacemos el make para crear el usuario/grupo e instalamos los archivos de Nagios Core.

```
debian@debian:~/nagios-4.4.6$ sudo make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
```

```
debian@debian:~/nagios-4.4.6$ sudo make install
cd ./base && make install
make[1]: se entra en el directorio '/home/debian/nagios-4.4.6/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
```

El siguiente make añadirá al sistema los ficheros de configuración necesarios para nuestro servidor web de Nagios.

```
debian@debian:~/nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

debian@debian:~/nagios-4.4.6$
```

Instalamos luego los ficheros de configuración necesarios para el inicio de Nagios

```
debian@debian:~/nagios-4.4.6$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/templatedata/templatedata.cfg /usr/local/nagios/etc/objects/templatedata.cfg

debian@debian:~/nagios-4.4.6$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
```

El siguiente comando sirve para indicar que se arranque en cada boot del sistema Nagios.

```
debian@debian:~/nagios-4.4.6$ sudo make install-daemoninit
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service.

*** Init script installed ***

debian@debian:~/nagios-4.4.6$
```

```
debian@debian:~/nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

Creamos un usuario para administrar Nagios

```
debian@debian:~/nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Activamos el módulo CGI de Apache necesario para iniciar Nagios

```
debian@debian:~/nagios-4.4.6$ sudo a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
systemctl restart apache2
debian@debian:~/nagios-4.4.6$ sudo systemctl restart apache2
debian@debian:~/nagios-4.4.6$
```

Accedemos al servidor web para confirmar que la instalación se haya realizado correctamente



En el apartado Hosts podremos ver nuestra propia máquina servidor

Host **	Status **	Last Check **	Duration **	Status Information
localhost	UP	03-28-2021 05:37:37	0d 0h 4m 18s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Luego en el apartado servicios podremos ver los servicios que se monitorizan del propio servidor por defecto

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
localhost	Current Load	OK	03-28-2021 05:34:29	0d 0h 4m 48s+	1/4	OK - load average: 0.50, 0.16, 0.06
	Current Users	OK	03-28-2021 05:35:07	0d 0h 4m 48s+	1/4	USERS OK - 0 users currently logged in
	HTTP	OK	03-28-2021 05:35:44	0d 0h 4m 48s+	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.199 second response time
	PING	OK	03-28-2021 05:36:22	0d 0h 4m 48s+	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	03-28-2021 05:36:59	0d 0h 4m 48s+	1/4	DISK OK - free space: / 3937 MiB (37.54% inode=70%):
	SSH	CRITICAL	03-28-2021 05:37:37	0d 0h 1m 3s	1/4	connect to address 127.0.0.1 and port 22: Conexión rehusada
	Swap Usage	OK	03-28-2021 05:38:14	0d 0h 4m 48s+	1/4	SWAP OK - 100% free (974 MB out of 974 MB)
	Total Processes	PENDING	N/A	0d 0h 4m 48s+	1/4	Service check scheduled for Sun Mar 28 05:38:52 CEST 2021

## Configurar los servicios para conectarse con Nagios

Para poder sincronizar las máquinas tendremos que instalar el plugin de nagios NRPE (Nagios Remote Plugin Executor). Este plugin nos permitirá ejecutar los comandos de monitorización de Nagios de forma remota desde el servidor para así poder obtener información.

Para realizar esta configuración tuvimos que instalar NRPE en todas las máquinas. El proceso de instalación fue idéntico a cuando instalamos Nagios Core. Primero descargamos el tar, lo descomprimos e instalamos mediante autotools.

```
debian@debian:~$ wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.0.3/nrpe-4.0.3.tar.gz
--2021-03-28 15:39:38-- https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.0.3/nrpe-4.0.3.tar.gz
Resolviendo github.com (github.com)... 140.82.121.4
Conectando con github.com (github.com)[140.82.121.4]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
```

```
*** Configuration summary for nrpe 4.0.3 2020-04-28 ***:
```

```
General Options:
```

```
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios
```

```
debian@debian:~/nrpe-4.0.3$ systemctl status nrpe
● nrpe.service - Nagios Remote Plugin Executor
   Loaded: loaded (/lib/systemd/system/nrpe.service; disabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-03-28 15:46:15 CEST; 12s ago
     Docs: http://www.nagios.org/documentation
    Main PID: 3458 (nrpe)
      Tasks: 1 (limit: 2347)
     Memory: 736.0K
    CGroup: /system.slice/nrpe.service
           └─3458 /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -f
debian@debian:~/nrpe-4.0.3$
```

Como hemos mencionado antes, este proceso de instalación de NRPE tendremos que repetirlo en las máquinas que vayamos a monitorizar, la diferencia será que en la máquina que tenga Nagios Core tenemos que compilar el comando `check_nrpe` y no todo el programario.

Comprobación de la instalación en la máquina con Nagios Core:

```
debian@debian-nagios:~$ /usr/local/nagios/libexec/check_nrpe -H 192.168.2.229
NRPE v4.0.3
```

Una vez instalado todo correctamente tendremos que crear el comando que utilizaremos para poder realizar la monitorización mediante el plugin.

```
define command {
    command_name    check_nrpe
    command_line    $USER$1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Cuando hayamos creado el comando, ya podremos pasar a la parte de configuración. En este caso lo que haremos será modificar el fichero nagios.cfg en el servidor y descomentaremos la siguiente línea, que será el directorio donde guardaremos los ficheros de configuración de los servicios que monitorizaremos. Una vez hecho esto, podremos pasar a crear el directorio que se nos indica y posteriormente pasaremos a crear los ficheros de configuración para las máquinas que queramos monitorizar.

(Por ejemplo: /usr/local/nagios/etc/servers/debian-http.config)

```
# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

cfg_dir=/usr/local/nagios/etc/servers
```

Para realizar la configuración de los servicios tendremos que modificar los ficheros nrpe.cfg de las máquinas clientes y el fichero <nombre\_servicio>.cfg del servidor. En el fichero nrpe.cfg tendremos que añadir el comando que utilizaremos para poder realizar la monitorización y añadir la ip/hostname del servidor en allowed\_hosts para que la máquina cliente pueda conectarse a la máquina servidor. Por parte del servidor tendremos que añadir las entradas correspondientes para que la máquina servidor tenga conexión con las máquinas cliente.

Las primeras capturas corresponden al fichero nrpe.cfg y las segundas al fichero de configuración del servidor



## Configuración fichero servidor HTTP

```
define host {
    use                linux-server
    host_name          debian-http
    alias              Servidor Web
    address            192.168.2.229
    max_check_attempts 5
    check_period       24x7
    notification_interval 30
    notification_period 24x7
}

define service{
    use                generic-service
    host_name          debian-http
    service_description Comprobar_web_1
    check_command       check_nrpe!check_http
}
```

```
debian@debian-http:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-03-29 00:09:12 CEST; 27min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 416 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 470 (apache2)
    Tasks: 55 (limit: 2347)
   Memory: 17.5M
    CGroup: /system.slice/apache2.service
            └─470 /usr/sbin/apache2 -k start
              └─472 /usr/sbin/apache2 -k start
                └─473 /usr/sbin/apache2 -k start
debian@debian-http:~$
```

```
command[check_http]=/usr/local/nagios/libexec/check_http localhost
```

## Configuración fichero servidor FTP

```
define host {
    use                linux-server
    host_name          debian-ftp
    alias              Servidor_FTP
    address            192.168.2.249
    max_check_attempts 5
    check_period       24x7
    notification_interval 30
    notification_period 24x7
}

define service{
    use                generic-service
    host_name          debian-ftp
    service_description Comprobar_ftp_1
    check_command       check_nrpe!check_ftp
}
```

```

debian@debian-ftp:~/nrpe-4.0.3$ systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-03-29 01:07:57 CEST; 22min ago
     Main PID: 1382 (vsftpd)
        Tasks: 1 (limit: 2347)
       Memory: 592.0K
      CGroup: /system.slice/vsftpd.service
              └─1382 /usr/sbin/vsftpd /etc/vsftpd.conf

```

```
command[check_ftp]=/usr/local/nagios/libexec/check_ftp -H localhost
```

## Configuración fichero servidor SSH

```

define host {
    use                linux-server
    host_name          debian-ssh
    alias              Servidor_SSH
    address            192.168.2.116
    max_check_attempts 5
    check_period       24x7
    notification_interval 30
    notification_period 24x7
}

```

```

define service{
    use                generic-service
    host_name          debian-ssh
    service_description Comprobar_SSH_1
    check_command       check_nrpe!check_ssh
}

```

```

debian@debian-ssh:~/nrpe-4.0.3$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-03-29 01:29:54 CEST; 3s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 31189 (sshd)
         Tasks: 1 (limit: 2347)
        Memory: 1.1M
       CGroup: /system.slice/ssh.service
               └─31189 /usr/sbin/sshd -D

```

```
command[check_total_procs]=/usr/local/nagios/libexec/check_ssh localhost
```

Después de configurar todas las máquinas, podremos comprobar en la web de Nagios que están todas activas.

Host ↕	Status ↕	Last Check ↕	Duration ↕	Status Information
debian-ftp 	UP	03-29-2021 01:42:51	0d 0h 5m 45s	PING OK - Packet loss = 0%, RTA = 1.0 ms
debian-http 	UP	03-29-2021 01:44:01	0d 0h 40m 16s	PING OK - Packet loss = 0%, RTA = 0.8 ms
debian-ssh 	UP	03-29-2021 01:40:02	0d 0h 5m 45s+	PING OK - Packet loss = 0%, RTA = 0.8 ms
localhost 	UP	03-29-2021 01:39:34	0d 9h 21m 53s	PING OK - Packet loss = 0%, RTA = 0.0 ms

#### **2.4. Ventajas de Nagios**

- Nagios es un software de código abierto. Es gratis de usar y editar.
- Tiene una configuración abierta por lo cual es fácil añadir scripts personalizados para ampliar los servicios disponibles.
- Hay muchos dispositivos que el sistema Nagios puede monitorear. El único requisito que necesita ese dispositivo es disponer del protocolo SNMP.
- Avisar, notificar o comentar el estado del sistema. Tiene una variedad de herramientas de alerta.
- Tiene muchos complementos integrados y complementos que se pueden descargar y desarrollar de forma gratuita.

#### **2.5. Desventajas de Nagios**

- Muchas funciones no están disponibles en la versión gratuita de Nagios. En Nagios XI se encuentran disponibles funciones como asistentes o un dashboard, pero esa versión es de pago.
- Hay muchos archivos de configuración y esto puede llegar a causar mucha confusión si no se tiene claro qué archivo se quiere modificar y con qué propósito.
- Nagios Core tiene una interfaz confusa.
- Nagios no puede administrar la red, solo monitorea la red.
- Nagios no puede monitorear el rendimiento de la red (ancho de banda, por ejemplo)

### 3. Ganglia



#### 3.1. Descripción del software

Ganglia es un sistema de monitorización de código abierto escalable para sistemas informáticos de alto rendimiento como clusters y grids. Este software viene incluido en distribuciones linux de nivel empresarial.

Ganglia se basa en un protocolo escucha/anuncio en broadcast que permite monitorizar el estado dentro de los clusters, y utiliza un árbol de conexiones punto a punto entre nodos del clúster representativos para federar los clusters (añadir más de un nodo) y agregar su estado. Utiliza tecnologías como XML para la representación de los datos, XDR para el transporte de datos compacto, y RRDtool para el almacenamiento y visualización de los datos.

#### 3.2. Componentes de Ganglia

- **gmond (daemon de monitoreo de Ganglia):** servicio encargado de recopilar información sobre un nodo. Se instala en todos los servidores (o hosts) que se quieran monitorear.
- **gmetad (Ganglia meta daemon):** servicio en el nodo maestro (servidor) que recopila datos de todos los gmond
- **herramienta RRD (base de datos Round Robin):** herramienta que se encuentra en el nodo servidor que sirve para almacenar datos y visualizaciones de ganglia en series de tiempo.
- **interfaz web PHP:** interfaz web en el nodo servidor que muestra gráficos y métricas a partir de los datos de RRD

#### 3.3. Despliegue de Ganglia

Para empezar con el despliegue de Ganglia, indicaremos el entorno que hemos escogido.

Como sistema operativo hemos escogido Ubuntu 18.04 debido a que es un sistema operativo con el que tenemos familiaridad.

Hemos creado dos máquinas, una para el servidor y otra para el cliente.



Las especificaciones que hemos dado a estas máquinas son las siguientes:

#### **General**

Nombre: Ganglia Server  
Sistema operativo: Ubuntu (64-bit)

#### **Sistema**

Memoria base: 4096 MB  
Orden de arranque: Disquete, Óptica, Disco duro  
Aceleración: VT-x/AMD-V, Paginación anidada, Paravirtualización KVM

#### **Pantalla**

Memoria de vídeo: 16 MB  
Controlador gráfico: VBoxVGA  
Servidor de escritorio remoto: Inhabilitado  
Grabación: Inhabilitado

#### **Almacenamiento**

Controlador: IDE  
IDE secundario maestro: [Unidad óptica] Vacío  
Controlador: SATA  
Puerto SATA 0: Ganglia Server.vdi (Normal, 10,00 GB)

#### **Audio**

Controlador de anfitrión: Windows DirectSound  
Controlador: ICH AC97

#### **Red**

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)

#### **USB**

Controlador USB: OHCI  
Filtros de dispositivos: 0 (0 activo)

#### **Carpetas compartidas**

Ninguno

#### **Descripción**

Ninguno

## Instalación de Apache2 HTTP Server en el Servidor

Instalaremos Apache2 HTTP server ya que Ganglia necesita tener un servidor web y este es de los más populares.

Para instalarlo utilizaremos los siguientes comandos en la máquina servidor:

```
gangliaserver@gangliaserver-VirtualBox:~$ sudo apt update
[sudo] contraseña para gangliaserver:
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:5 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [48,9 kB]
Des:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [60,4 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [295 kB]
Des:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2.464 B]
Des:9 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [290 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2.468 B]
Des:11 http://es.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [9.292 B]
Descargados 960 kB en 1s (675 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 239 paquetes. Ejecute «apt list --upgradable» para verlos.
gangliaserver@gangliaserver-VirtualBox:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0
gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
```

*sudo apt update*

*sudo apt install apache2*

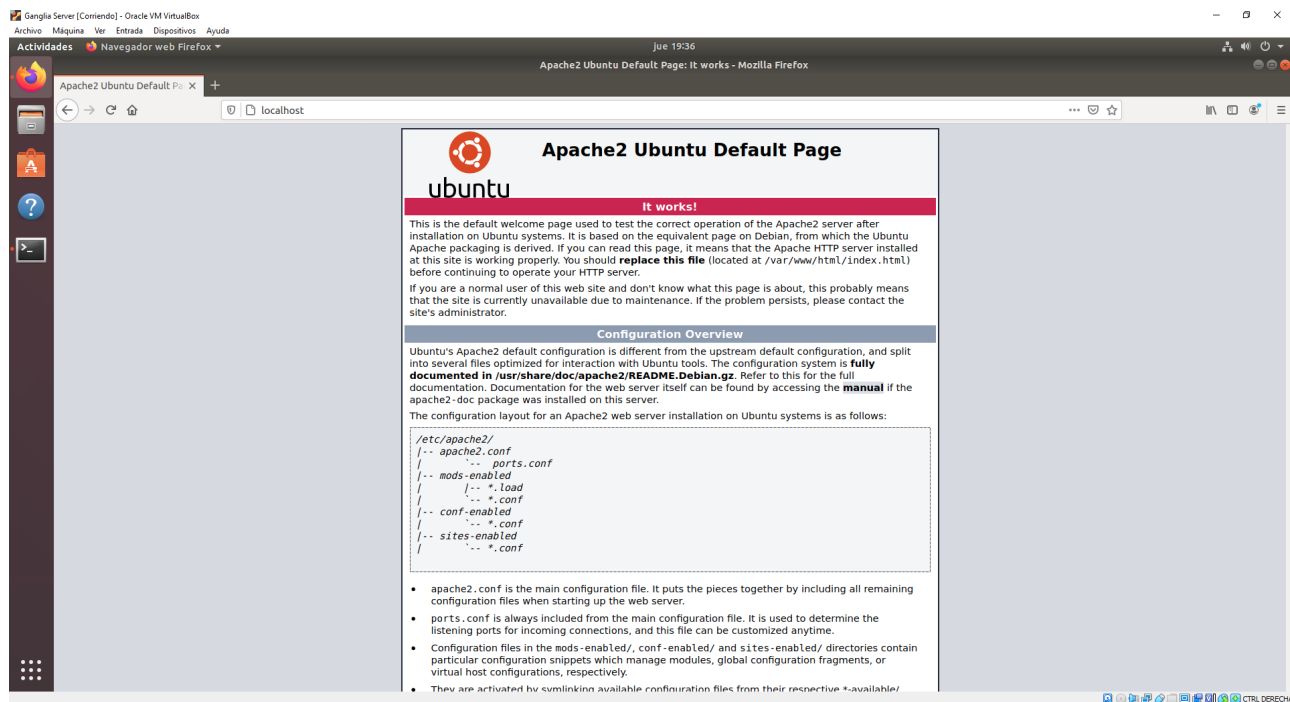
Para iniciar, parar, y activar Apache2 siempre que el server arranque podemos utilizar los siguientes comandos:

*sudo systemctl start apache2.service*

*sudo systemctl stop apache2.service*

*sudo systemctl enable apache2.service*

Para comprobar que Apache2 está instalado y operativo abriremos el navegador y en la barra de búsqueda pondremos *localhost* o su dirección IP. Sabremos que funciona correctamente si la página resultante es esta:



## Instalación de PHP 7.2 y módulos relacionados en el Servidor

Algunas versiones de Ubuntu no tienen PHP 7.2 en sus repositorios por defecto, por lo que si no lo tenemos tendremos que instalarlo.

Para comprobar si lo tenemos y, en caso de tenerlo, qué versión tenemos introduciremos el siguiente comando en el terminal:

```
gangliaserver@gangliaserver-VirtualBox:~$ php -v

No se ha encontrado la orden «php», pero se puede instalar con:

sudo apt install php7.2-cli
sudo apt install hhvm
```

*php -v*

Como podemos ver, en nuestro caso no tenemos PHP por lo que tendremos que instalarlo.

Aquí tenemos dos opciones:

- Instalar PHP 7.2 con HHVM  
*sudo apt install php7.2-cli*  
*sudo apt install hhvm*
- Instalar PPA para versiones soportadas de PHP con extensiones PECL  
*sudo apt-get install software-properties-common*  
*sudo add-apt-repository ppa:ondrej/php*

En este caso, nos hemos inclinado por instalar PHP usando la segunda opción ya que nos interesan los módulos que estos comandos nos aportan.

Queremos utilizar PPA ya que es el repositorio dónde se encontrará PHP. Y utilizando el segundo comando nos instalará este PPA, y además añadiremos el repositorio que nos aporta los módulos necesarios para poder continuar con la instalación.

```
gangliaser@gangliaser-VirtualBox:~$ sudo apt-get install software-properties-common
[sudo] contraseña para gangliaser:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.

gangliaser@gangliaser-VirtualBox:~$ sudo add-apt-repository ppa:ondrej/php
Co-installable PHP versions: PHP 5.6, PHP 7.x and most requested extensions are included. Only Supported Versions of PHP (http://php.net/supported-versions.php) for Supported Ubuntu Releases (https://wiki.ubuntu.com/Releases) are provided. Don't ask for end-of-life PHP versions or Ubuntu release, they won't be provided.
```

A continuación haremos un update con:

*sudo apt update*

Y continuaremos instalando PHP con sus módulos con:

*sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common*  
*php7.2-gmp php7.2-curl php7.2-intl php7.2-mbstring*  
*php7.2-xmllrpc php7.2-mysql php7.2-gd php7.2-xml php7.2-cli*  
*php7.2-zip*

```
gangliaser@gangliaser-VirtualBox:~$ sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-gmp php7.2-curl php7.2-intl php7.2-mbstring php7.2-xmllrpc php7.2-mysql php7.2-gd php7.2-xml php7.2-cli php7.2-zip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
```

Para comprobar que PHP7.2 se ha instalado correctamente introduciremos de nuevo el comando

*php -v*

```
gangliaser@gangliaser-VirtualBox:~$ php -v
PHP 7.2.34-18+ubuntu18.04.1+deb.sury.org+1 (cli) (built: Feb 23 2021 15:08:03) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.34-18+ubuntu18.04.1+deb.sury.org+1, Copyright (c) 1999-2018, by Zend Technologies
```



Una vez instalado, abriremos el fichero de configuración por defecto para Apache2:

```
sudo nano /etc/php/7.2/apache2/php.ini
```

En caso de que la configuración no traiga estos valores por defecto, los cambiaremos tal que:

file_uploads	On
allow_url_fopen	On
short_open_tag	On
memory_limit	256M
upload_max_filesize	100M
max_execution_time	360
max_input_vars	1500
date.timezone	Europe/Madrid

Con nuestra configuración por defecto, nos ha sido necesario cambiar todos los valores excepto los dos primeros, pero es necesario que se comprueben todos.

Una vez cambiados estos valores, guardaremos la configuración y cerraremos el fichero.

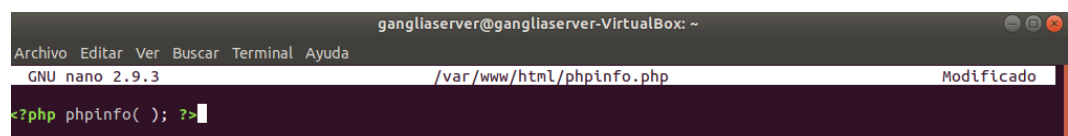
Para recargar la configuración de PHP reiniciamos Apache2 con el siguiente comando:

```
sudo systemctl restart apache2.service
```

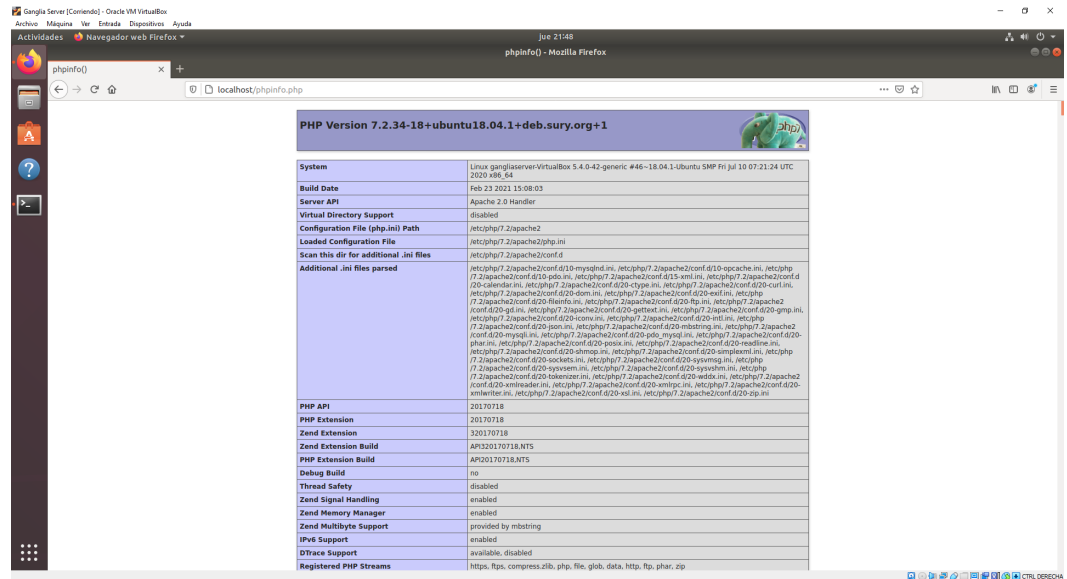
Para verificar que PHP7.2 está instalado y encontrar las configuraciones por defecto de Apache2, creamos un fichero phpinfo.php en el directorio raíz:

```
sudo nano /var/www/html/phpinfo.php
```

Una vez creado, añadiremos la siguiente línea, guardaremos y cerraremos el fichero



A continuación, iremos a localhost/phpinfo.php en nuestro navegador por tal de ver la información PHP.



## Instalación del paquete de Ganglia

Una vez hayamos instalado Apache2 y PHP procederemos a instalar Ganglia.

```
sudo apt update
sudo apt install ganglia-monitor rrdtool gmetad
ganglia-webfrontend
```

```
ganglia-server@ganglia-server-VirtualBox:~$ sudo apt install ganglia-monitor rrdtool gmetad ganglia-webfrontend
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0
gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1
libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1
libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14
libfreerdp-client2-2 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6
libgpgod-common libgpgod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
libmsh-0.1-1 libodfgen-0.1-1 libqpdf2v5 libraw16 libvenge-0.0-0 libsgutils2-2 libssh-4
libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian30 libxmlsec1-nss lp-solve
media-player-info python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
usb-creator-common
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
libconfuse-common libconfuse2 libdbi1 libganglia1 librrd8
```

Cuando la instalación haya finalizado, podemos utilizar estos comandos para iniciar, parar y activar Ganglia siempre que el server arranque:

```
sudo systemctl start ganglia-monitor.service
sudo systemctl stop ganglia-monitor.service
sudo systemctl enable ganglia-monitor.service
```

## Configuración del servicio de Ganglia

Los ficheros de configuración de Ganglia los podremos encontrar en el directorio `/etc/ganglia`. Para que la configuración se adapte al entorno tendremos que hacer algunos cambios en el fichero de configuración principal, que es *gmetad* y se encuentra en el directorio mencionado previamente.

Comenzaremos comprobando si Ganglia está instalado:

*systemctl status ganglia-monitor.service*

```
gangliaserver@gangliaserver-VirtualBox:~$ systemctl status ganglia-monitor.service
● ganglia-monitor.service
   Loaded: loaded (/etc/init.d/ganglia-monitor; generated)
   Active: active (running) since Thu 2021-04-01 21:55:48 CEST; 44min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 2 (limit: 4667)
   CGroup: /system.slice/ganglia-monitor.service
           └─19623 /usr/sbin/gmond --pid-file /var/run/gmond.pid

abr 01 21:55:48 gangliaserver-VirtualBox systemd[1]: Starting ganglia-monitor.service...
abr 01 21:55:48 gangliaserver-VirtualBox ganglia-monitor[19621]: Starting Ganglia Monitor Daemon: gmond.
abr 01 21:55:48 gangliaserver-VirtualBox systemd[1]: Started ganglia-monitor.service.
```

Tenemos que definir el nodo master en el fichero *gmetad*, por lo que tendremos que modificarlo:

*sudo nano /etc/ganglia/gmetad.conf*

Cambiaremos la línea de *data\_source* “my cluster” localhost por *data\_source* “my cluster” 50 [ip]:8649 :

```
GNU nano 2.9.3 /etc/ganglia/gmetad.conf

# note that the web frontend determines a host as down if its TN value is less
# than 4 * TMAX (20sec by default). Therefore, if you set the polling interval
# to something around or greater than 80sec, this will cause the frontend to
# incorrectly display hosts as down even though they are not.
#
# A list of machines which service the data source follows, in the
# format ip:port, or name:port. If a port is not specified then 8649
# (the default gmond port) is assumed.
# default: There is no default value
#
# data_source "my cluster" 10 localhost my.machine.edu:8649 1.2.3.5:8655
# data_source "my grid" 50 1.3.4.7:8655 grid.org:8651 grid-backup.org:8651
# data_source "another source" 1.3.4.7:8655 1.3.4.8
#data_source "my cluster" localhost
data_source "my cluster" 50 192.168.1.60:8649
```

Después de hacer estos cambios, abriremos el fichero de configuración de gmond *gmond.conf* que se encuentra en `/etc/ganglia`

*sudo nano /etc/ganglia/gmond.conf*

En este fichero comentaremos *mcast\_join* y *bind*, y en lugar de *mvast\_join* pondremos *host* con nuestra IP en *udp\_send\_channel*:

```
cluster {
  name = "cluster cpd"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}

/* The host section describes attributes of the host, like the location */
host {
  location = "unspecified"
}

/* Feel free to specify as many udp_send_channels as you like.  Gmond
   used to only support having a single channel */
udp_send_channel {
  /*mcast_join = 192.168.1.60*/
  host = 192.168.1.60
  port = 8649
  ttl = 1
}

/* You can specify as many udp_rcv_channels as you like as well. */
udp_rcv_channel {
  /*mcast_join = 192.168.1.60*/
  port = 8649
  /* bind = 192.168.1.60*/
}
```

Por último, copiamos el fichero de configuración de Ganglia al directorio del host virtual de Apache.

```
sudo cp /etc/ganglia-webfrontend/apache.conf
/etc/apache2/sites-enabled/ganglia.conf
```

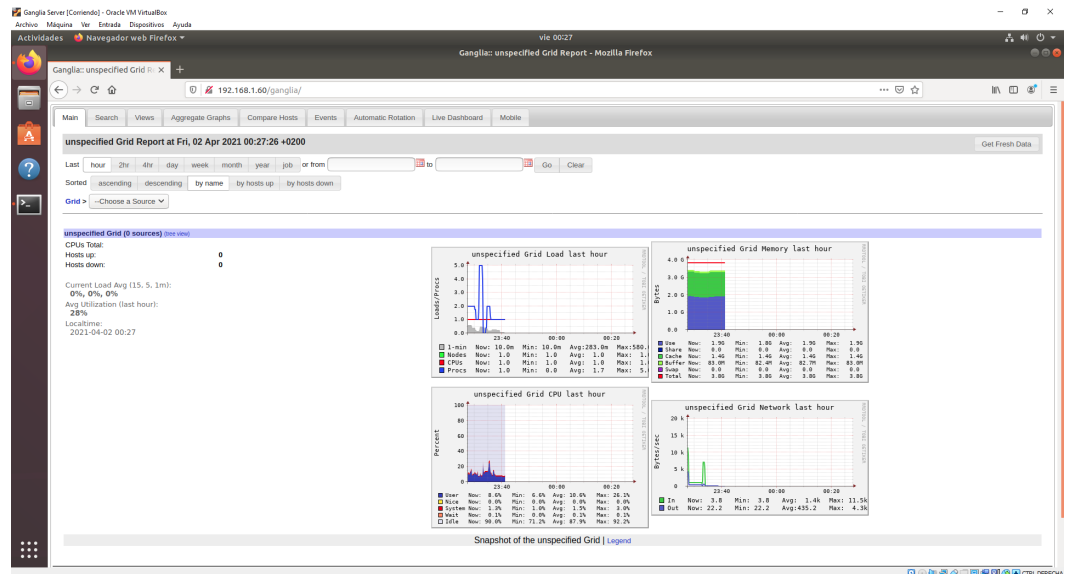
Podemos reiniciar el monitor de Ganglia, gmetad y Apache services con los siguientes comandos:

```
sudo systemctl restart ganglia-monitor
sudo systemctl restart gmetad
sudo systemctl restart apache2
```

### Acceso al portal web de Ganglia

Para acceder al portal de Ganglia lo podemos hacer accediendo a este enlace con el navegador:

*<http://192.168.1.60/ganglia> (donde 192.168.1.60 es nuestra IP)*



## Instalar Ganglia en el cliente

Una vez hayamos instalado Ganglia en la máquina server y esté funcionando, instalaremos Ganglia en las máquinas cliente.

Para instalarlo, simplemente usaremos el comando:

```
sudo apt install ganglia-monitor
```

```
Ganglia Cliente [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal
jue 23:52
gangliaserver@gangliaserver-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
gangliaserver@gangliaserver-VirtualBox:~$ sudo apt install ganglia-monitor
[sudo] contraseña para gangliaserver:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
```

Una vez instalado, hace falta definir el nodo master cambiando *mcast\_join* por *host* en *udp\_send\_channel* por la IP del nodo master.

```
sudo nano /etc/ganglia/gmond.conf
```

```
/* Feel free to specify as many udp_send_channels as you like. Gmond
   used to only support having a single channel */
udp_send_channel {
  /*mcast_join = 192.168.1.60*/
  host = 192.168.1.60
  port = 8649
  ttl = 1
}
```

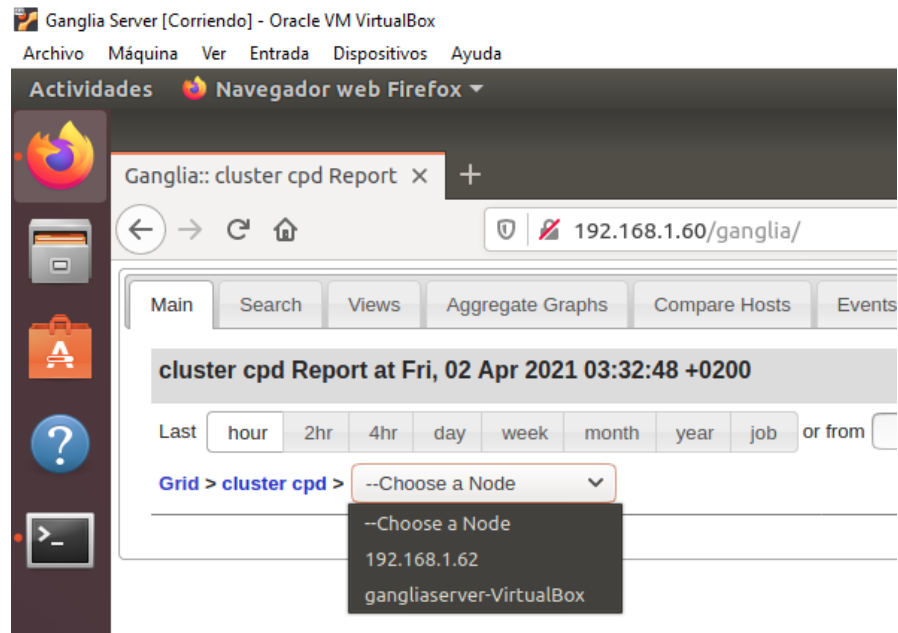
Para finalizar, levantaremos Ganglia con el siguiente comando:

```
systemctl start ganglia-monitor
```

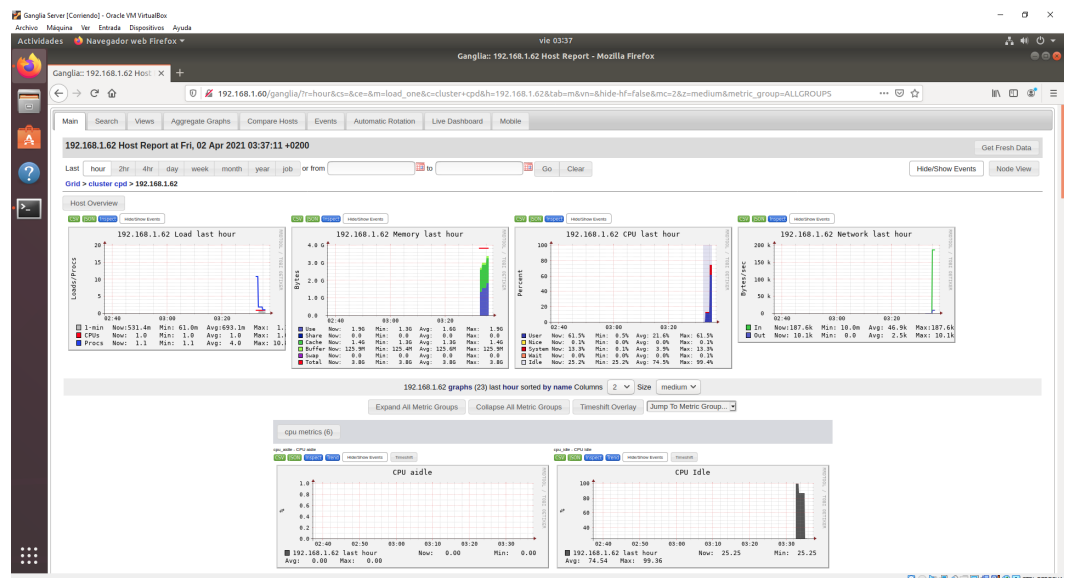
## Comprobar la monitorización de Ganglia

Para comprobar el correcto funcionamiento de Ganglia, iremos a la máquina servidor y en el navegador volveremos/refrescaremos la página <http://192.168.1.60/ganglia> (poniendo nuestra IP).

Como vemos en esta imagen, nos tiene que dar opción a visualizar la monitorización de los clientes y el servidor en el cual está corriendo Ganglia.



En la captura de abajo, podemos ver la monitorización del cliente (192.168.1.62). Como no hemos tenido el cliente mucho rato activo no hay mucho tráfico, pero podemos observar el flujo que tiene estando activo.



### **3.4. Ventajas de Ganglia**

- Código abierto
- Escalable
- No hace falta instalar ningún plugin
- Instalación sencilla, sobretodo para los clientes
- Envío de métricas mediante UDP (más seguros)

### **3.5. Desventajas de Ganglia**

- Enfocado únicamente para sistemas informáticos de alto rendimiento
- El uso de multicast puede llegar a ser contraproducente ya que no es bueno para las transferencias en bulk
- La transferencia mediante UDP no es confiable

## 4. Conclusiones

Una vez vistos los dos softwares de monitorización, sus ventajas y sus desventajas, creemos que los softwares no son comparables. Vemos que Nagios se utiliza principalmente para monitorizar redes, mientras que Ganglia se utiliza más para monitorizar el rendimiento de los hosts.

Nagios necesita de plugins como NRPE, el cual se tiene que instalar en cada máquina y nos da información del estado de la máquina, si está levantada o no y el servicio. Este plugin no nos proporciona una vista gráfica del rendimiento, y requiere de una configuración y mantenimiento.

Unos de los problemas más importantes es la carga y los problemas de seguridad que este plugin aporta a la red, ya que el servidor tiene que tener conexiones TCP con todos los hosts de la red y puede dar pie a vectores de ataque.

En cambio, Ganglia envía sus métricas utilizando UDP, evitando así que los hosts que están siendo monitoreados eviten tener que aceptar conexiones TCP y las implicaciones de seguridad que eso conlleva.

Hemos visto que también es posible utilizar ambos servicios con ganglia-nagios-bridge. Esto puede ser cómodo ya que solo tendremos un solo fichero de configuración en el cual podemos especificar qué métricas queremos usar de Ganglia y mapearlas a servicios de Nagios, por ejemplo.

En conclusión, creemos que al ser softwares que están enfocados a tareas diferentes es difícil compararlos. Podemos ver que Nagios tiene problemas de seguridad y de carga de red causados por NRPE, por lo que creemos que en ese aspecto podemos decir que Ganglia nos parece más seguro ya que usa UDP evitando así esos problemas. También nos ha parecido interesante que Nagios y Ganglia se puedan combinar ya que esto nos permite tener un fichero de configuración personalizable, en vez de un por defecto, y nos permite extraer métricas de Ganglia XML y Nagios analiza las máquinas mediante el plugin NRPE.